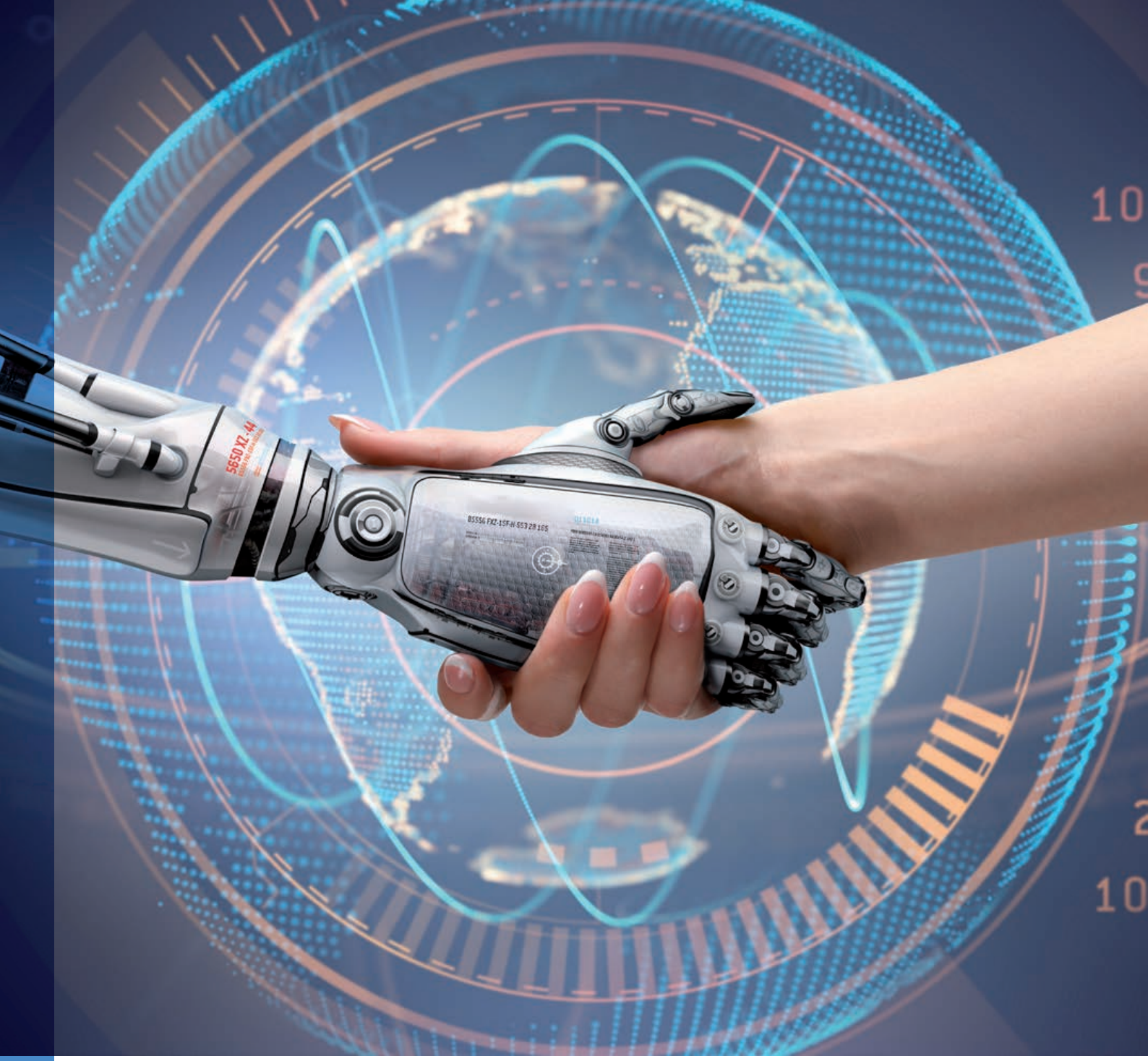


Unleashing the potential of the Internet of Things

1956 2016
CCITT / ITU-T

ITU


UIT ITU

ABOUT ITU-T

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

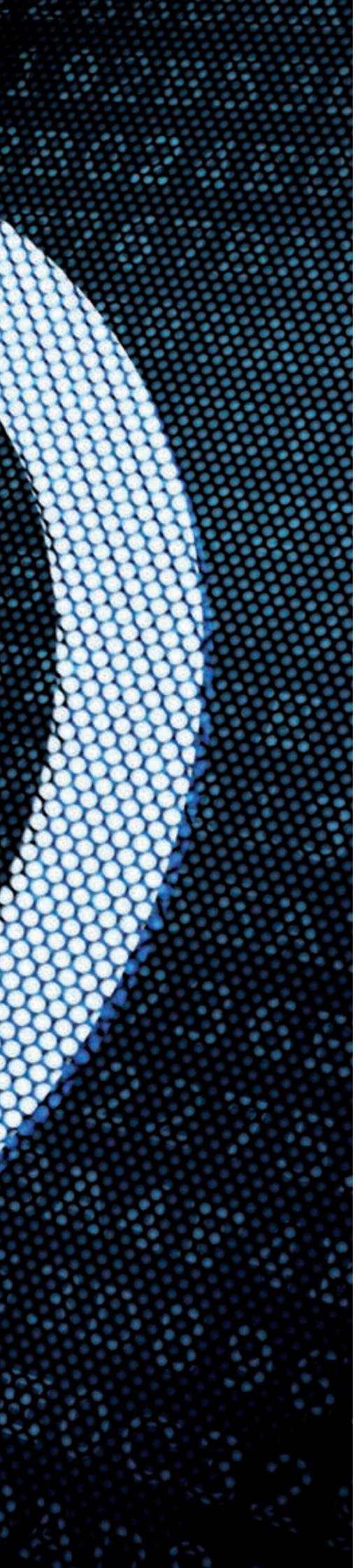
In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE ON THESE RECOMMENDATIONS

In the Recommendations that constitute this compendium, the expression “Administration” is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with a Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability), and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words “shall” or some other obligatory language such as “must” and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

Individual ITU-T Recommendations can downloaded from
http://www.itu.int/en/ITU-T/publications



INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of any of these Recommendations may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of the Recommendations that constitute this compendium, ITU had received notice of intellectual property, protected by patents, which may be required to implement one or more of the Recommendations. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at http://www.itu.int/ITU-T/ipr/.


IOT
22.02.35.2
90.50.3.2
Innovation
Branding
Solution
Marketing
Analysis
Ideas
Success
Management
PLATFORM
APPLICATIONS

FOREWORD

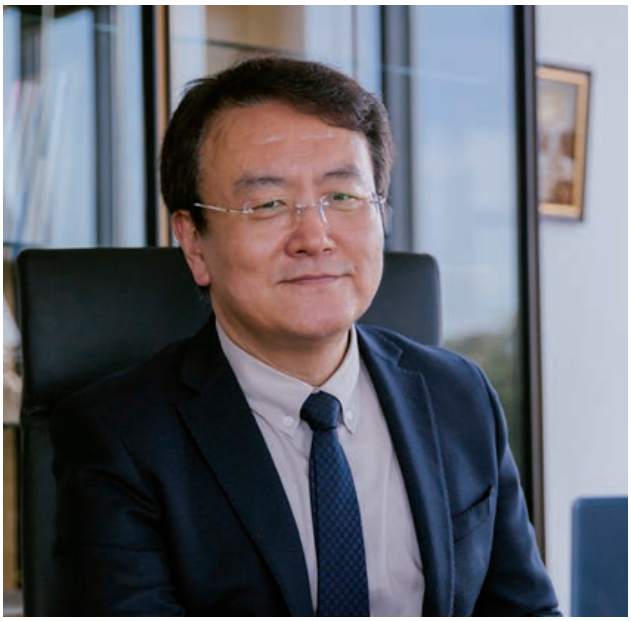
The years approaching 2020 will see Internet of Things (IoT) technologies enabling the interconnection of billions of devices, things and objects to achieve the efficiencies borne of innovations such as intelligent buildings and transportation systems, and smart energy and water networks.

IoT is contributing to the convergence of industry sectors, with utilities, healthcare and transportation among the many sectors with a stake in the future of IoT. The new ITU-T Study Group 20 established in June 2015 provides the specialized IoT standardization platform necessary for this convergence to rest on a cohesive set of international standards.

Today we are faced with the challenge of addressing the standardization requirements of the many vertical industries applying information and communication technologies (ICTs) as enabling technologies. This is particularly evident in the field of IoT, where IoT platforms are being developed independently, according to the specific needs of each sector. This divergence in IoT development and deployment has led to an urgent need for stakeholders to come together to mitigate the risk of data "silos" emerging in different industry sectors.

ITU-T Study Group 20 has taken up this challenge, providing government, industry and academia with a unique global platform to collaborate in the development of international IoT standards. One of the group's primary objectives is to support the creation of an inclusive, interoperable IoT ecosystem capable of making full use of the data generated by IoT-enabled systems.

The Study Group is building on over ten years of ITU-T experience in IoT standardization, developing international standards to enable the coordinated development of IoT technologies, including radio-frequency identification, ubiquitous sensor networks and machine-to-machine communications. A central part of this study is the standardization of end-to-end architectures



for IoT, and mechanisms for the interoperability of IoT applications and datasets employed by various vertical industries. An important aspect of the group's work is the development of standards that leverage IoT technologies to address urban-development challenges.

This flipbook presents a compendium of the first set of ITU international standards for IoT, providing a resource of great value to standards experts interested in contributing to the work of ITU-T Study Group 20. This compendium is also expected to assist the wide variety of stakeholders interested in implementing these IoT standards or calling for adherence to standards in policy and regulatory frameworks relevant to IoT. This compendium will be updated continuously, according to the progress of IoT developments in ITU.

Chaesub Lee
Director, ITU Telecommunication
Standardization Bureau

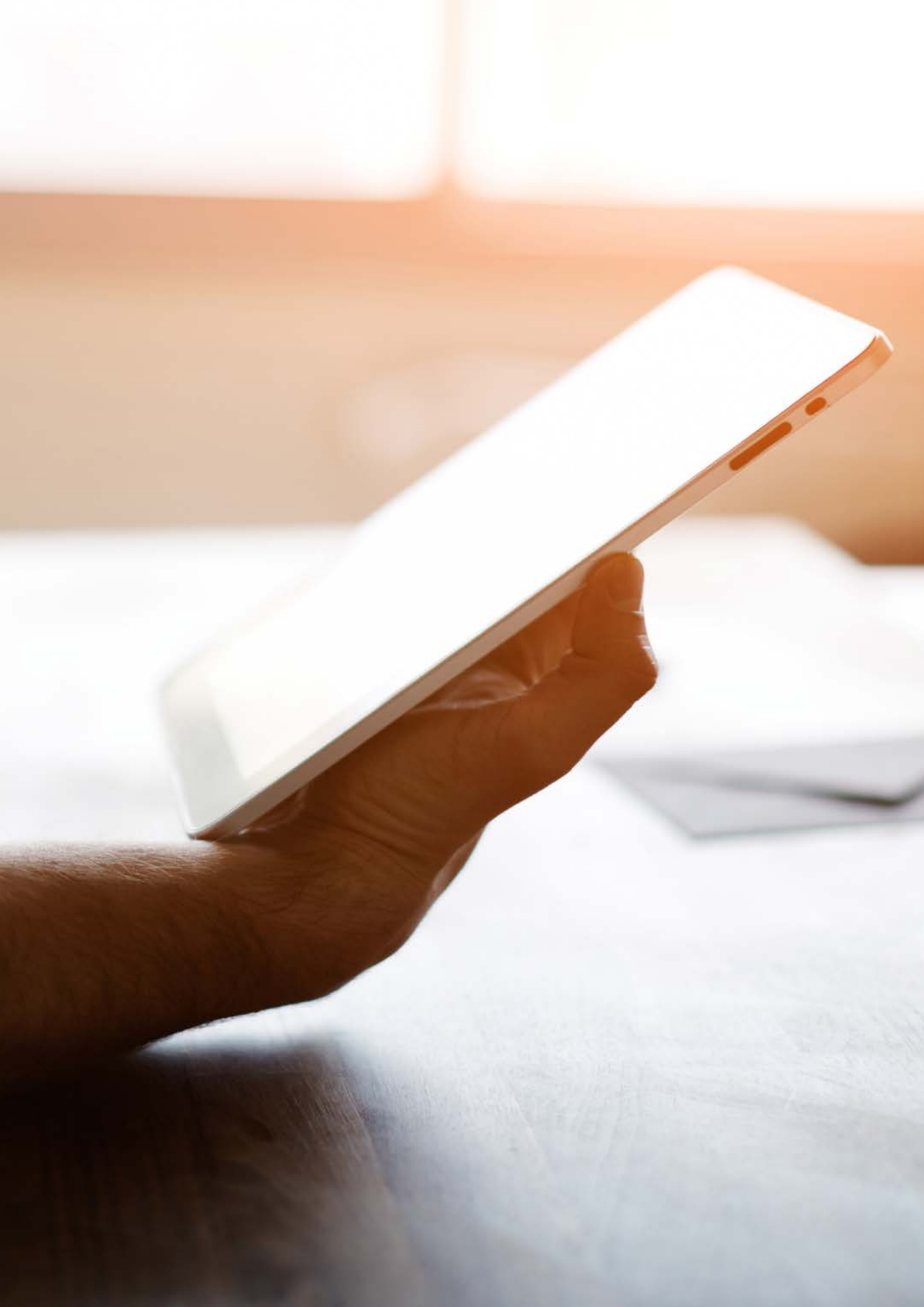


TABLE OF CONTENTS

TABLE OF CONTENTS


Aa

General 1


31
1

Y.4000/Y.2060

Overview of the Internet of things

Overview of the Internet of things

Summary

Recommendation ITU-T Y.2060 provides an overview of the Internet of things (IoT). It clarifies the concept and scope of the IoT, identifies the fundamental characteristics and high-level requirements of the IoT and describes the IoT reference model. The ecosystem and business models are also provided in an informative appendix.

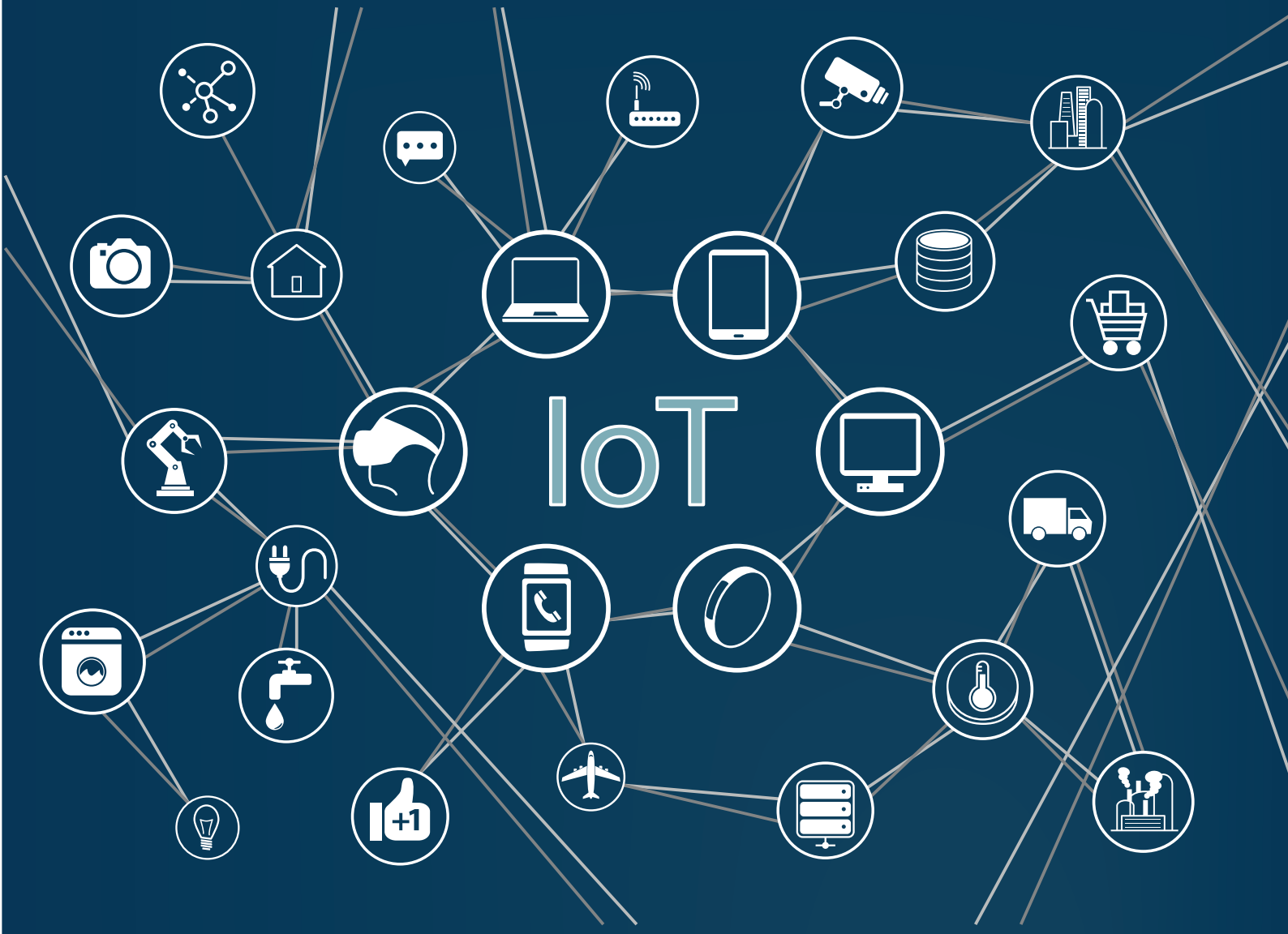
History

Edition	Recommendation	Approval	Study Group
1.0	ITU-T Y.2060	2012-06-15	13

Keywords

Device, Internet of things, physical thing, reference model, thing, virtual thing.

Table of Contents


IoT
+1

Recommendation ITU-T Y.4000/Y.2060

Overview of the Internet of things

1 Scope

This Recommendation provides an overview of the Internet of things (IoT) with the main objective of highlighting this important area for future standardization.

More specifically, this Recommendation covers the following:

– IoT-related terms and definitions

– concept and scope of the IoT

– characteristics of the IoT

– high-level requirements of the IoT

– IoT reference models.

IoT ecosystem and business models-related information is provided in Appendix I.

2 References

None.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following term defined elsewhere:

3.1.1 next generation network (NGN) [b-ITU-T Y.2001]: A packet-based network which is able to provide telecommunication services and able to make use of multiple broadband, QoS-enabled transport technologies and in which service-related functions are independent from underlying transport-related technologies. It enables unfettered access for users to networks and to competing service providers and/or services of their choice. It supports generalized mobility which will allow consistent and ubiquitous provision of services to users.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 device: With regard to the Internet of things, this is a piece of equipment with the mandatory capabilities of communication and the optional capabilities of sensing, actuation, data capture, data storage and data processing.

3.2.2 Internet of things (IoT): A global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies.

NOTE 1 – Through the exploitation of identification, data capture, processing and communication capabilities, the IoT makes full use of things to offer services to all kinds of applications, whilst ensuring that security and privacy requirements are fulfilled.

NOTE 2 – From a broader perspective, the IoT can be perceived as a vision with technological and societal implications.

3.2.3 thing: With regard to the Internet of things, this is an object of the physical world (physical things) or the information world (virtual things), which is capable of being identified and integrated into communication networks.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

2G Second Generation

3G Third Generation

AAA Authentication, Authorization and Accounting

CAN Controller Area Network

DSL Digital Subscriber Line

FCAPS Fault, Configuration, Accounting, Performance, Security

ICT Information and Communication Technology

IoT Internet of Things

ITS Intelligent Transport Systems

LTE Long Term Evolution

NGN Next Generation Network

PSTN Public Switched Telephone Network

TCP/IP Transmission Control Protocol/Internet Protocol

5 Conventions

None.

6 Introduction of the IoT

6.1 Concept of the IoT

The Internet of things (IoT) can be perceived as a far-reaching vision with technological and societal implications.

From the perspective of technical standardization, the IoT can be viewed as a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies (ICT).

Through the exploitation of identification, data capture, processing and communication capabilities, the IoT makes full use of "things" to offer services to all kinds of applications, whilst ensuring that security and privacy requirements are fulfilled.

NOTE – The IoT is expected to greatly integrate leading technologies, such as technologies related to advanced machine-to-machine communication, autonomic networking, data mining and decision-making, security and privacy protection and cloud computing, with technologies for advanced sensing and actuation.

As shown in Figure 1, the IoT adds the dimension "Any THING communication" to the information and communication technologies (ICTs) which already provide "any TIME" and "any PLACE" communication.

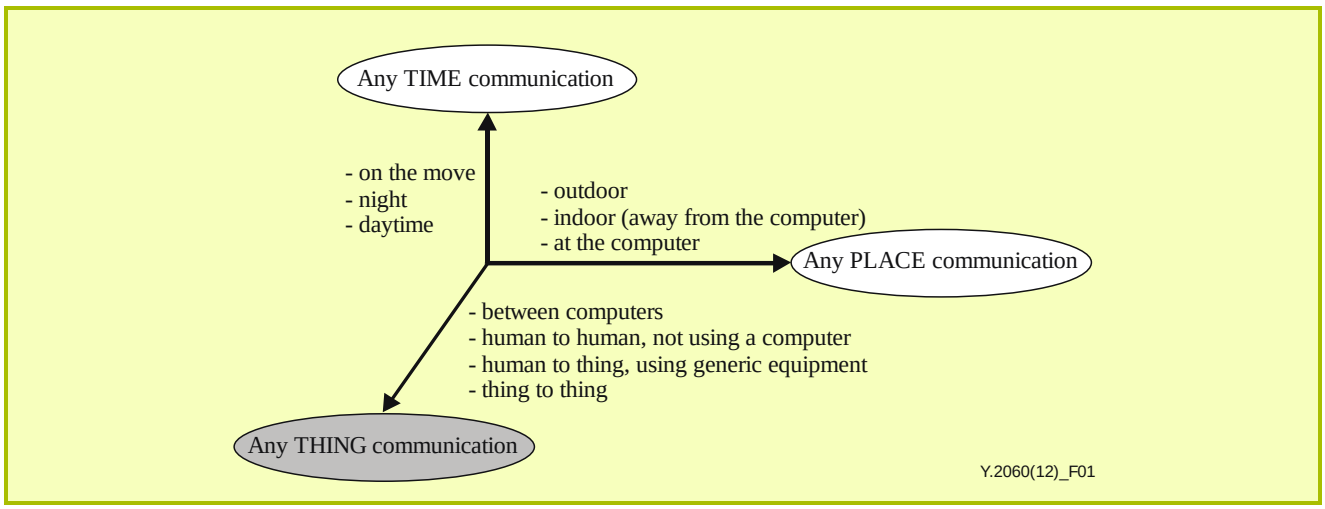


Figure 1 – The new dimension introduced in the Internet of things [b-ITU Report]

Regarding the IoT, things are objects of the physical world (physical things) or of the information world (virtual world) which are capable of being identified and integrated into communication networks. Things have associated information, which can be static and dynamic.

Physical things exist in the physical world and are capable of being sensed, actuated and connected. Examples of physical things include the surrounding environment, industrial robots, goods and electrical equipment.

Virtual things exist in the information world and are capable of being stored, processed and accessed. Examples of virtual things include multimedia content and application software.

6.2 Technical overview of the IoT

Figure 2 shows the technical overview of the IoT.

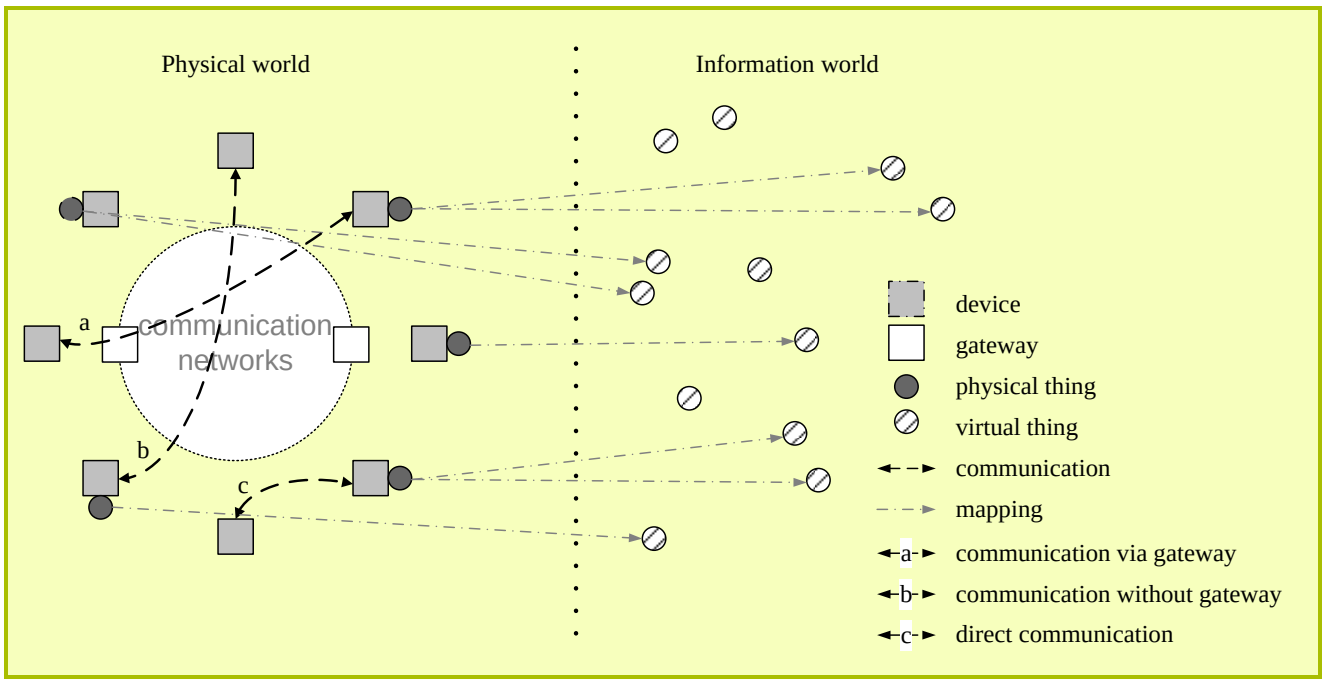


Figure 2 – Technical overview of the IoT

A physical thing may be represented in the information world via one or more virtual things (mapping), but a virtual thing can also exist without any associated physical thing.

A device is a piece of equipment with the mandatory capabilities of communication and optional capabilities of sensing, actuation, data capture, data storage and data processing. The devices collect various kinds of information and provide it to the information and communication networks for further processing. Some devices also execute operations based on information received from the information and communication networks.

Devices communicate with other devices: they communicate through the communication network via a gateway (case a), through the communication network without a gateway (case b) or directly, that is without using the communication network (case c). Also, combinations of cases a and c, and cases b and c are possible; for example, devices can communicate with other devices using direct communication through a local network (i.e., a network providing local connectivity between devices and between devices and a gateway, such as an ad-hoc network) (case c) and then communication through the communication network via a local network gateway (case a).

NOTE 1 – Although Figure 2 shows only interactions taking place in the physical world (communications between devices), interactions also take place in the information world (exchanges between virtual things) and between the physical world and the information world (exchanges between physical things and virtual things).

The IoT applications include various kinds of applications, e.g., "intelligent transportation systems", "smart grid", "e-health" or "smart home". The applications can be based on proprietary application platforms, but can also be built upon common service/application support platform(s) providing generic enabling capabilities, such as authentication, device management, charging and accounting.

The communication networks transfer data captured by devices to applications and other devices, as well as instructions from applications to devices. The communication networks provide capabilities for reliable and efficient data transfer. The IoT network infrastructure may be realized via existing networks, such as conventional TCP/IP-based networks, and/or evolving networks, such as next generation networks (NGN) [b-ITU-T Y.2001].

Figure 3 shows the different types of devices and the relationship between devices and physical things.

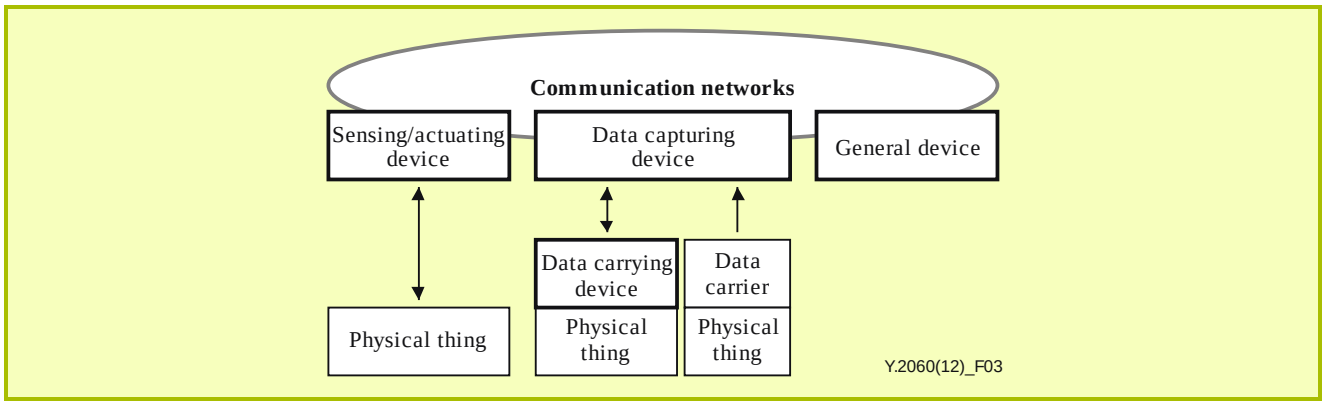


Figure 3 – Types of devices and their relationship with physical things

NOTE 2 – A "general device" is also a (set of) physical thing(s).

The minimum requirement of the devices in the IoT is their support of communication capabilities. Devices are categorized into data-carrying devices, data-capturing devices, sensing and actuating devices and general devices as described as follows:

– Data-carrying device: A data-carrying device is attached to a physical thing to indirectly connect the physical thing with the communication networks.

– Data-capturing device: A data-capturing device refers to a reader/writer device with the capability to interact with physical things. The interaction can happen indirectly via data-carrying devices, or directly via data carriers attached to the physical things. In the first case, the data-capturing device reads information on a data-carrying device and can optionally also write information given by the communication networks on the data-carrying device.

NOTE 3 – Technologies used for interaction between data-capturing devices and data-carrying devices or data carriers include radio frequency, infrared, optical and galvanic driving.

– Sensing and actuating device: A sensing and actuating device may detect or measure information related to the surrounding environment and convert it into digital electronic signals. It may also convert digital electronic signals from the information networks into operations. Generally, sensing and actuating devices form local networks communicate with each other using wired or wireless communication technologies and use gateways to connect to the communication networks.

– General device: A general device has embedded processing and communication capabilities and may communicate with the communication networks via wired or wireless technologies. General devices include equipment and appliances for different IoT application domains, such as industrial machines, home electrical appliances and smart phones.

7 Fundamental characteristics and high-level requirements of the IoT

7.1 Fundamental characteristics

The fundamental characteristics of the IoT are as follows:

– Interconnectivity: With regard to the IoT, anything can be interconnected with the global information and communication infrastructure.

– Things-related services: The IoT is capable of providing thing-related services within the constraints of things, such as privacy protection and semantic consistency between physical things and their associated virtual things. In order to provide thing-related services within the constraints of things, both the technologies in physical world and information world will change.

– Heterogeneity: The devices in the IoT are heterogeneous as based on different hardware platforms and networks. They can interact with other devices or service platforms through different networks.

– Dynamic changes: The state of devices change dynamically, e.g., sleeping and waking up, connected and/or disconnected as well as the context of devices including location and speed. Moreover, the number of devices can change dynamically.

– Enormous scale: The number of devices that need to be managed and that communicate with each other will be at least an order of magnitude larger than the devices connected to the current Internet. The ratio of communication triggered by devices as compared to communication triggered by humans will noticeably shift towards device-triggered communication. Even more critical will be the management of the data generated and their interpretation for application purposes. This relates to semantics of data, as well as efficient data handling.

7.2 High-level requirements

The following provide high-level requirements which are relevant for the IoT:

– Identification-based connectivity: The IoT needs to support that the connectivity between a thing and the IoT is established based on the thing's identifier. Also, this includes that possibly heterogeneous identifiers of the different things are processed in a unified way.

– Interoperability: Interoperability needs to be ensured among heterogeneous and distributed systems for provision and consumption of a variety of information and services.

– Autonomic networking: Autonomic networking (including self-management, self-configuring, self-healing, self-optimizing and self-protecting techniques and/or mechanisms) needs to be supported in the networking control functions of the IoT, in order to adapt to different application domains, different communication environments and large numbers and types of devices.

– Autonomic services provisioning: The services need to be able to be provided by capturing, communicating and processing automatically the data of things based on the rules configured by operators or customized by subscribers. Autonomic services may depend on the techniques of automatic data fusion and data mining.

– Location-based capabilities: Location-based capabilities need to be supported in the IoT. Something-related communications and services will depend on the location information of things and/or users. It is needed to sense and track the location information automatically. Location-based communications and services may be constrained by laws and regulations, and should comply with security requirements.

– Security: In the IoT, every 'thing' is connected which results in significant security threats, such as threats towards confidentiality, authenticity and integrity of both data and services. A critical example of security requirements is the need to integrate different security policies and techniques related to the variety of devices and user networks in the IoT.

– Privacy protection: Privacy protection needs to be supported in the IoT. Many things have their owners and users. Sensed data of things may contain private information concerning their owners or users. The IoT needs to support privacy protection during data transmission, aggregation, storage, mining and processing. Privacy protection should not set a barrier to data source authentication.

– High quality and highly secure human body related services: High quality and highly secure human body related services needs to be supported in the IoT. Different countries have different laws and regulations on these services.

NOTE – Human body related services refer to the services provided by capturing, communicating and processing the data related to human static features and dynamic behaviour with or without human intervention.

– Plug and play: Plug and play capability needs to be supported in the IoT in order to enable on-the-fly generation, composition or the acquiring of semantic-based configurations for seamless integration and cooperation of interconnected things with applications, and responsiveness to application requirements.

– Manageability: Manageability needs to be supported in the IoT in order to ensure normal network operations. IoT applications usually work automatically without the participation of people, but their whole operation process should be manageable by the relevant parties.

8 IoT reference model

Figure 4 shows the IoT reference model. It is composed of four layers as well as management capabilities and security capabilities which are associated with the four layers.

The four layers are as follows:

– application layer
– service support and application support layer
– network layer
– device layer.

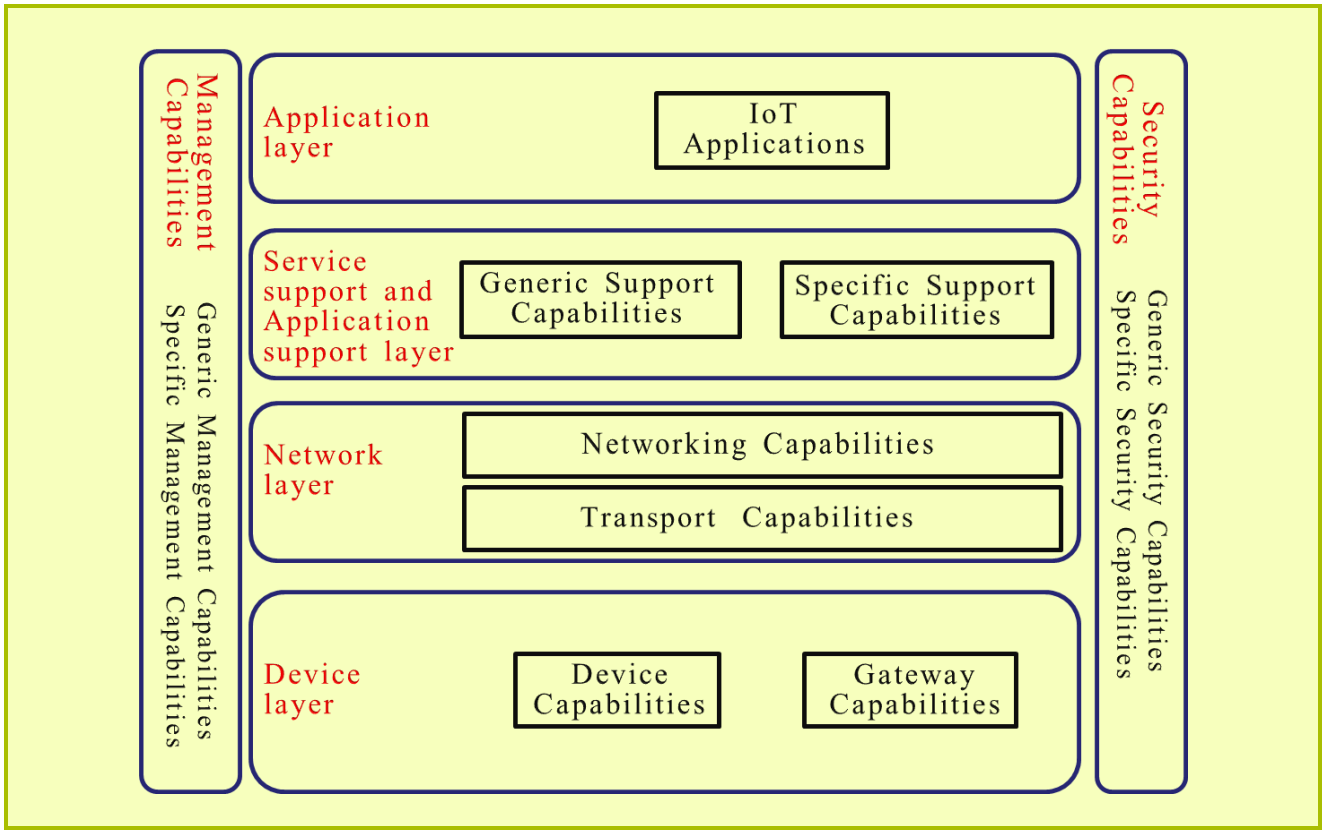


Figure 4 – IoT reference model

8.1 Application layer

The application layer contains IoT applications.

8.2 Service support and application support layer

The service support and application support layer consists of the following two capability groupings:

– Generic support capabilities: The generic support capabilities are common capabilities which can be used by different IoT applications, such as data processing or data storage. These capabilities may be also invoked by specific support capabilities, e.g., to build other specific support capabilities.
– Specific support capabilities: The specific support capabilities are particular capabilities which cater for the requirements of diversified applications. In fact, they may consist of various detailed capability groupings, in order to provide different support functions to different IoT applications.

8.3 Network layer

This consists of the following two types of capabilities:

– Networking capabilities: provide relevant control functions of network connectivity, such as access and transport resource control functions, mobility management or authentication, authorization and accounting (AAA).

– Transport capabilities: focus on providing connectivity for the transport of IoT service and application specific data information, as well as the transport of IoT-related control and management information.

8.4 Device layer

Device layer capabilities can be logically categorized into two kinds of capabilities:

– **Device capabilities**:

The device capabilities include but are not limited to:

Direct interaction with the communication network: Devices are able to gather and upload information directly (i.e., without using gateway capabilities) to the communication network and can directly receive information (e.g., commands) from the communication network.

Indirect interaction with the communication network: Devices are able to gather and upload information to the communication network indirectly, i.e., through gateway capabilities. On the other side, devices can indirectly receive information (e.g., commands) from the communication network.

Ad-hoc networking: Devices may be able to construct networks in an ad-hoc manner in some scenarios which need increased scalability and quick deployment.

Sleeping and waking-up: Device capabilities may support "sleeping" and "waking-up" mechanisms to save energy.

NOTE – The support in a single device of both capabilities of direct interaction with the communication network and indirect interaction with the communication network is not mandatory.

– **Gateway capabilities**:

The gateway capabilities include but are not limited to:

Multiple interfaces support: At the device layer, the gateway capabilities support devices connected through different kinds of wired or wireless technologies, such as a controller area network (CAN) bus, ZigBee, Bluetooth or Wi-Fi. At the network layer, the gateway capabilities may communicate through various technologies, such as the public switched telephone network (PSTN), second generation or third generation (2G or 3G) networks, long-term evolution networks (LTE), Ethernet or digital subscriber lines (DSL).

Protocol conversion: There are two situations where gateway capabilities are needed. One situation is when communications at the device layer use different device layer protocols, e.g., ZigBee technology protocols and Bluetooth technology protocols, the other one is when communications involving both the device layer and network layer use different protocols e.g., a ZigBee technology protocol at the device layer and a 3G technology protocol at the network layer.

8.5 Management capabilities

In a similar way to traditional communication networks, IoT management capabilities cover the traditional fault, configuration, accounting, performance and security (FCAPS) classes, i.e., fault management, configuration management, accounting management, performance management and security management.

The IoT management capabilities can be categorized into generic management capabilities and specific management capabilities.

Essential generic management capabilities in the IoT include:

– device management, such as remote device activation and de-activation, diagnostics, firmware and/or software updating, device working status management;

– local network topology management;

– traffic and congestion management, such as the detection of network overflow conditions and the implementation of resource reservation for time-critical and/or life-critical data flows.

Specific management capabilities are closely coupled with application-specific requirements, e.g., smart grid power transmission line monitoring requirements.

8.6 Security capabilities

There are two kinds of security capabilities: generic security capabilities and specific security capabilities. Generic security capabilities are independent of applications. They include:

– at the application layer: authorization, authentication, application data confidentiality and integrity protection, privacy protection, security audit and anti-virus;

– at the network layer: authorization, authentication, use data and signalling data confidentiality, and signalling integrity protection;

– at the device layer: authentication, authorization, device integrity validation, access control, data confidentiality and integrity protection.

Specific security capabilities are closely coupled with application-specific requirements, e.g., mobile payment, security requirements.

Appendix I

IoT ecosystem and business models

(This appendix does not form an integral part of this Recommendation.)

I.1 Business roles

The IoT ecosystem is composed of a variety of business players. Each business player plays at least one business role, but more roles are possible. The identified IoT business roles are shown in Figure I.1.

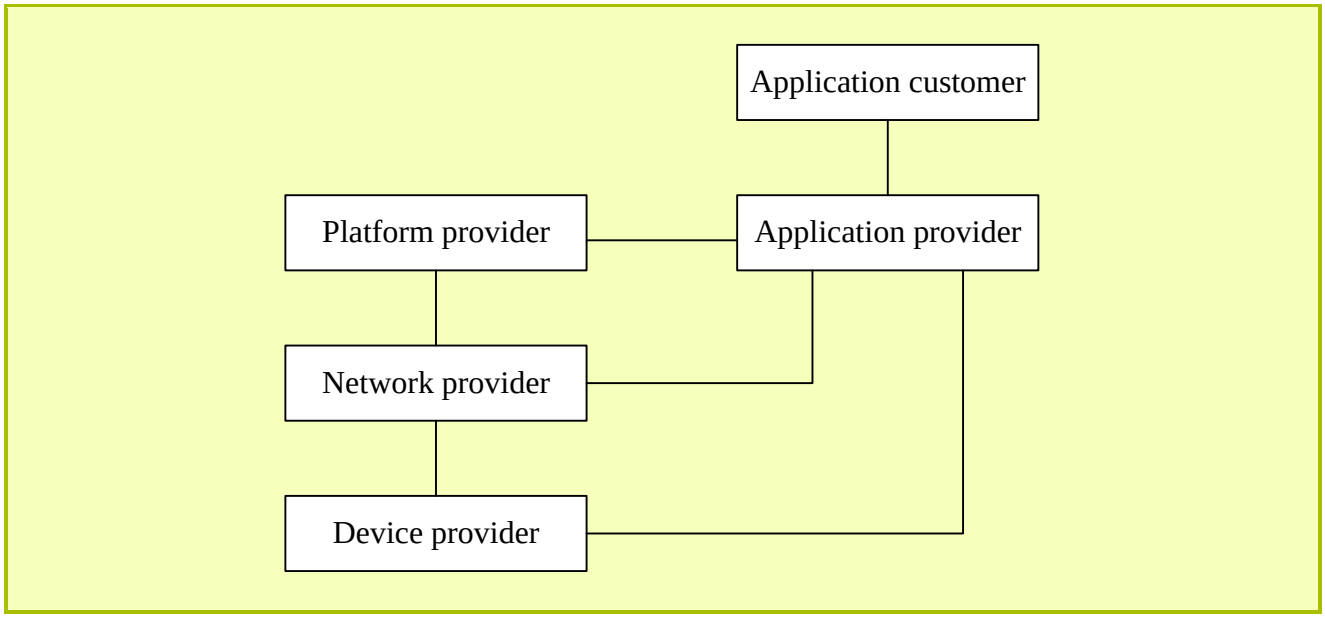


Figure I.1 – IoT ecosystem

NOTE – The identified business roles and their relationships as described in the IoT ecosystem do not represent all possible relevant roles and relationships which can be found across IoT business deployments.

I.1.1 Device provider

The device provider is responsible for devices providing raw data and/or content to the network provider and application provider according to the service logic.

I.1.2 Network provider

The network provider plays a central role in the IoT ecosystem. In particular, the network provider performs the following main functions:

– access and integration of resources provided by other providers;

– support and control of the IoT capabilities infrastructure;

– offering of IoT capabilities, including network capabilities and resource exposure to other providers.

I.1.3 Platform provider

The platform provider provides integration capabilities and open interfaces. Different platforms can provide different capabilities to application providers. Platform capabilities include typical integration capabilities, as well as data storage, data processing or device management. Support for different types of IoT applications is also possible.

I.1.4 Application provider

The application provider utilizes capabilities or resources provided by the network provider, device provider and platform provider, in order to provide IoT applications to application customers.

I.1.5 Application customer

The application customer is the user of IoT application(s) provided by the application provider.

NOTE – An application customer may represent multiple applications users.

I.2 Business models

The IoT ecosystem players may have a variety of relationships in real deployments.

The motivations for this variety of relationships are based on different possible business models. This appendix examines only some IoT business models from the perspective of telecom service and network operators. From this perspective, five business models are described below.

I.2.1 Model 1

In model 1, player A operates the device, network, platform and applications and serves the application customer directly, as shown in Figure I.2.

In general, telecom operators and some vertically integrated businesses (such as smart grid and intelligent transport systems (ITS) businesses) act as player A in model 1.

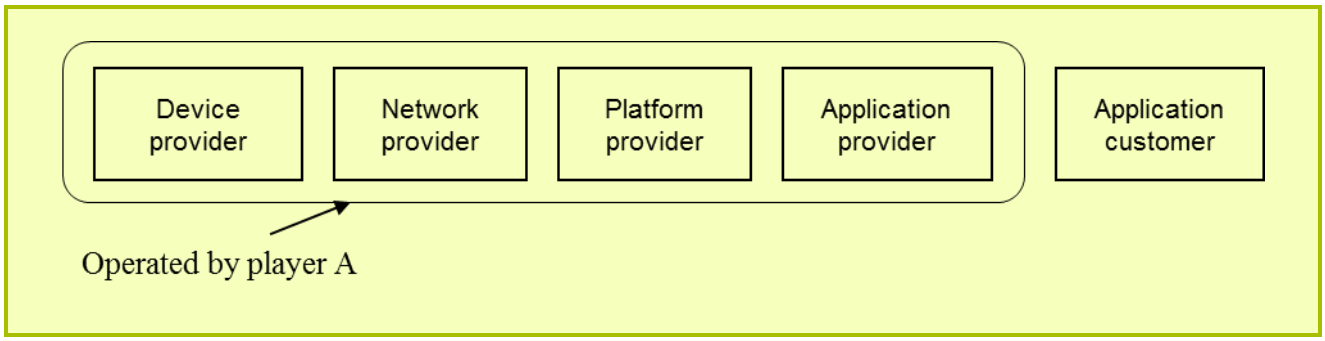


Figure I.2 – Model 1

I.2.2 Model 2

In model 2, player A operates the device, network, and platform, and player B operates the application and serves the application customers, as shown in Figure I.3.

In general, telecom operators act as player A, other service providers as player B in model 2.

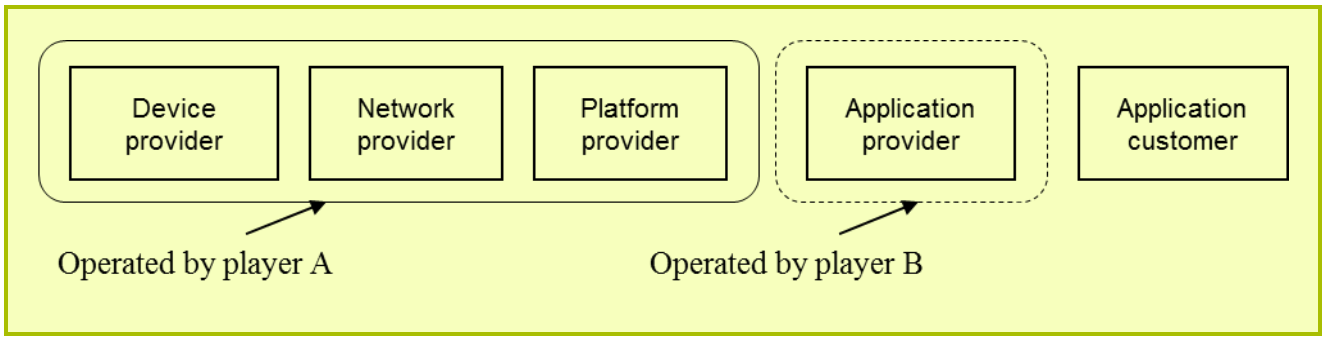


Figure I.3 – Model 2

I.2.3 Model 3

In model 3, player A operates the network and platform, player B operates the device and applications and serves the application customers, as shown in Figure I.4.

In general, telecom operators act as player A and other service providers act as player B.

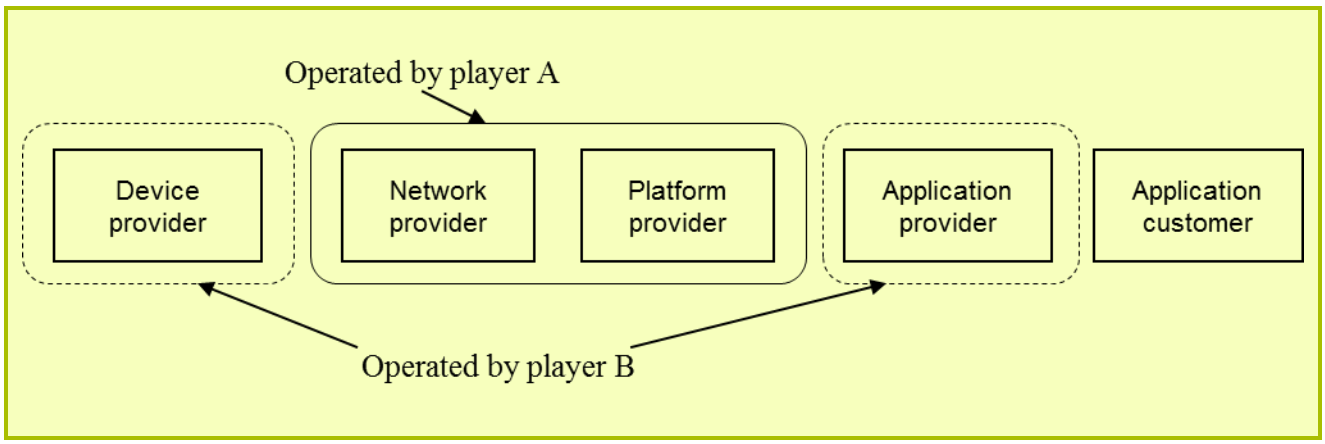


Figure I.4 – Model 3

I.2.4 Model 4

In model 4, player A only operates the network and player B operates the device and platform, providing applications to the application customers, as shown in Figure I.5.

In general, telecom operators act as player A, other service providers and vertically integrated businesses act as player B in model 4.

NOTE – A variation of this model does not include a platform provider and associated platform functionalities (player B only provides applications).

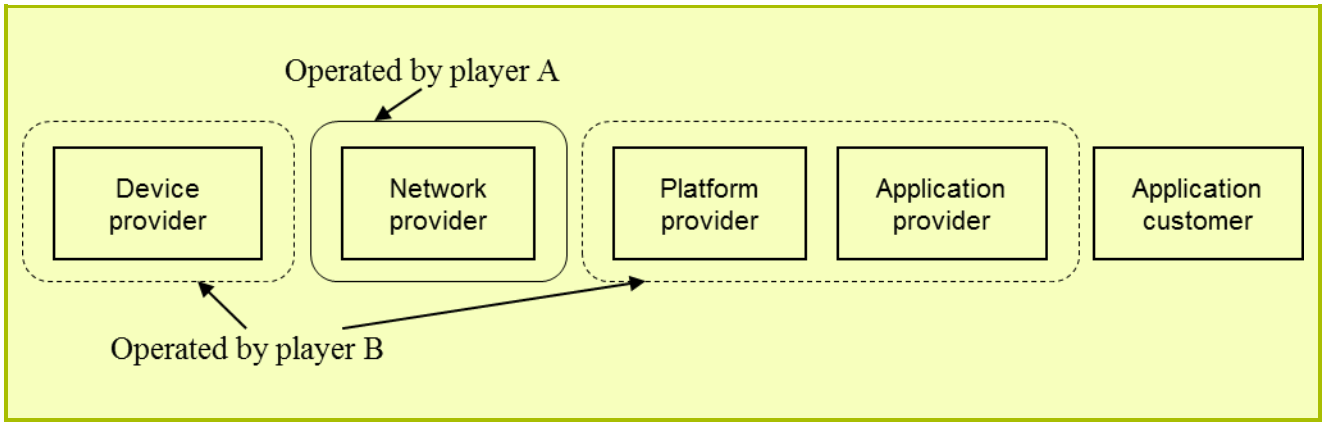


Figure I.5 – Model 4

I.2.5 Model 5

In model 5, player A only operates the network, player B operates the platform, and player C operates devices and provides applications to the application customers, as shown in Figure I.6.

In general, telecom operators act as player A, other service providers act as player B, and vertically integrated businesses act as player C in model 5.

NOTE – A variation of this model does not include a platform provider and associated platform functionalities (player B only provides applications).

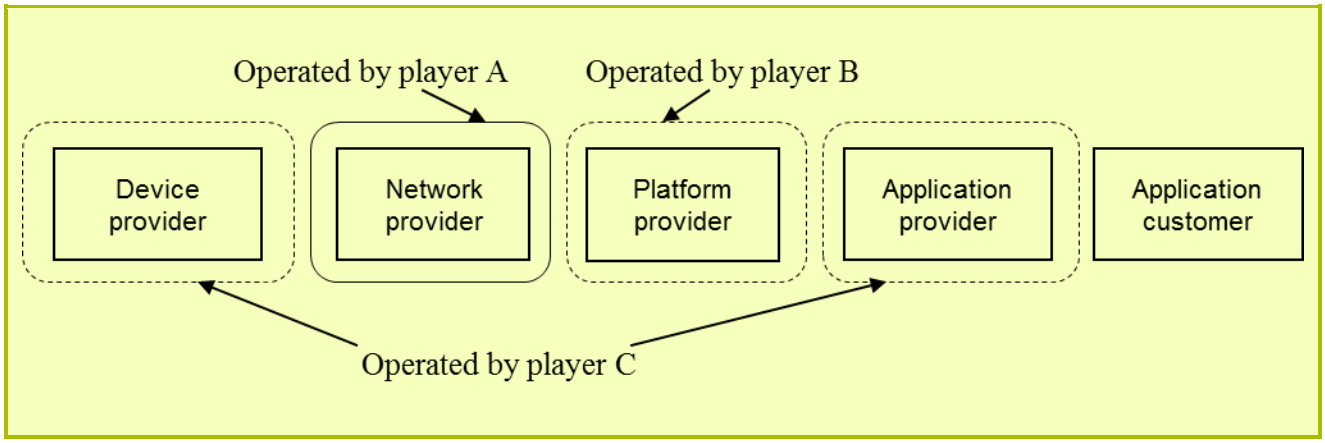
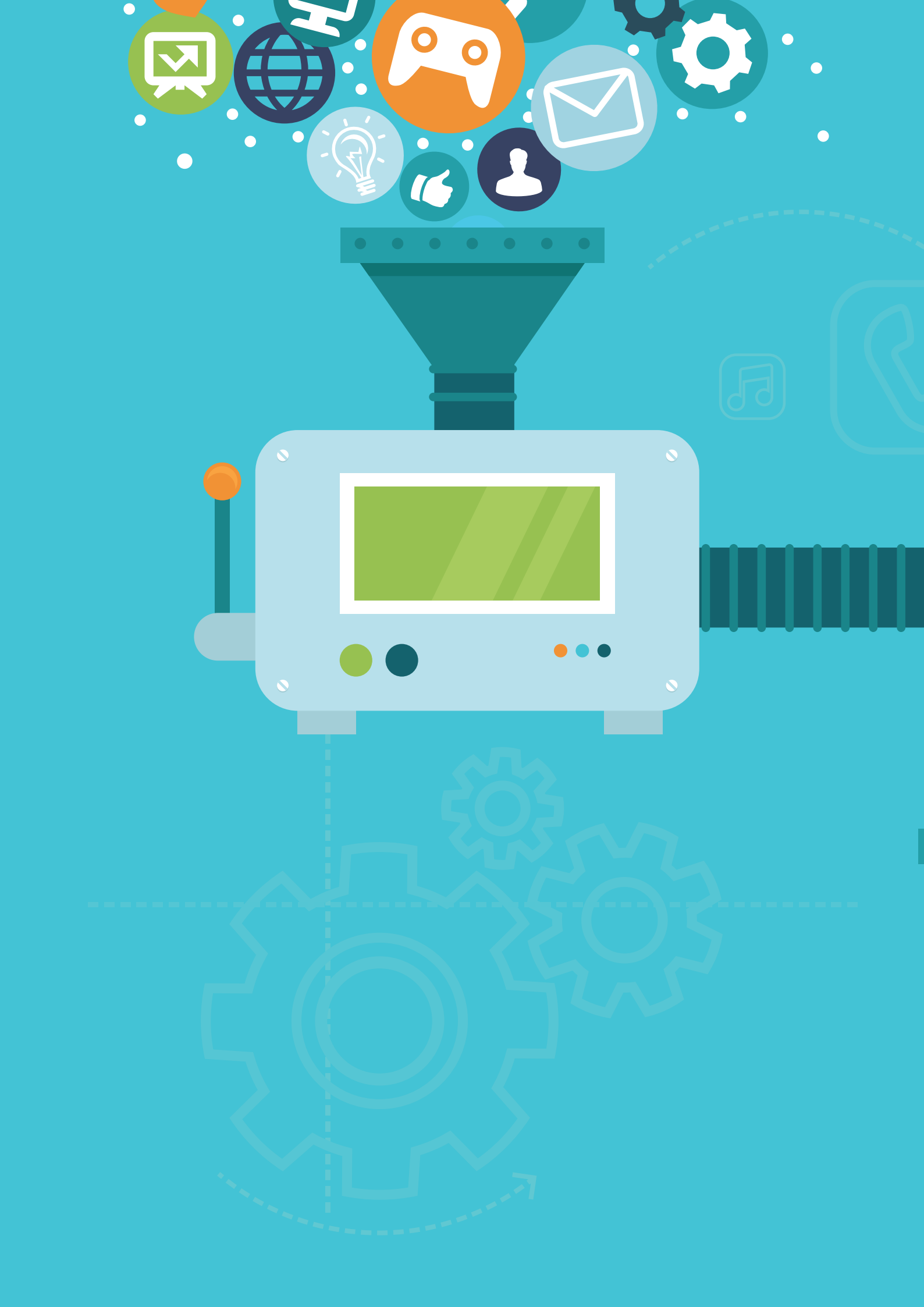


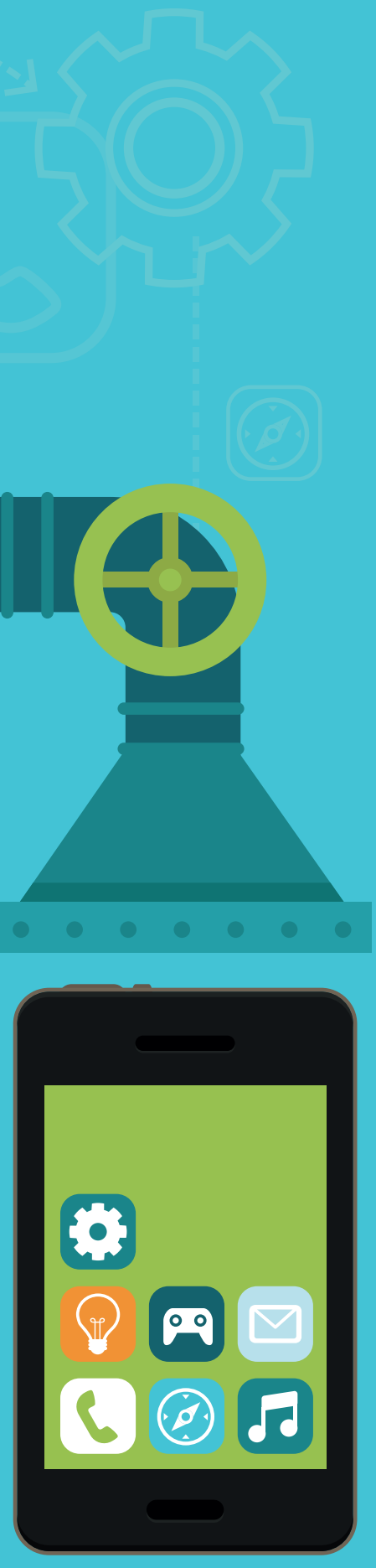
Figure I.6 – Model 5

Bibliography

[b-ITU Report] ITU Internet Reports (2005), *The Internet of Things*.

[b-ITU-T Y.2001] Recommendation ITU-T Y.2001 (2004), *General overview of NGN*.





Y.4001/F.748.2

Machine socialization: Overview and reference model

Machine socialization: Overview and reference model

Summary

Recommendation ITU-T Y.4001/F.748.2 describes machine socialization, which enables machines to cooperate with one another via their relations with other machines. In machine socialization, machines can be identified, can communicate and can capture data using machine identifiers, features of machine capabilities and machine owners, etc. Machines can be socialized with the information of identified machines through the establishment of relations. This Recommendation provides an overview, requirements and a reference model for machine socialization.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T Y.4001/F.748.2	2015-11-29	16	11.1002/1000/12621

Keywords

Internet of things (IoT), machine socialization, social Internet of things, socialization, social web of things.

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

Table of Contents

Introduction

Social relations existed between people before the appearance of social network services as known today. However, these social relations were constrained by time, location, space, etc.

A social networking service is a platform that enables the building of social networks or social relations among people who share interests, activities, backgrounds or real-life connections. Unlike traditional social networks or social relations, social network services make it possible to connect people who share interests and activities across political, economic, and geographic borders, etc. In addition, social network services make it easy to create, maintain, strengthen and extend social networks or social relations.

The most important factor in the use of social network services is the possibility of being able to cooperate with other people including crowd activities by sharing and exchanging information.

According to the definition of the Internet of things (IoT), things or machines collect data (either environmental or non-environmental) and transfer this data to the information world through communication networks. Though things or machines are interconnected with one another, the important point of the IoT is in providing the capability for communication and data (either environmental or non-environmental) capture to things or machines. Without collaboration or cooperation between things or machines, they may remain isolated and constrained from a capability point of view.

Because humans have an always-on networking capability, a social network service becomes a great way to share and exchange information. Using this capability, it is easy for humans to acquire information on the experience, knowledge and capability of other humans without the barriers associated with time, space, etc.

Consequently, it can be easily understood that all networked things or machines will:

– produce numerous items of meaningful information or more specifically, captured data, occasionally pre-processed by things or machines;

– evolve intellectually and then converse with one another, in other words, they will be socialized.

To enable things to communicate what they do or need, follow one another, discuss with one another, collaborate, create events and do things together demands the socialization of machines to a level corresponding to that of social relations among humans.

Recommendation ITU-T Y.4001/F.748.2

Machine socialization: Overview and reference model

1 Scope

This Recommendation specifies machine socialization which enables machines to cooperate with one another using their relations with other machines. This Recommendation covers the following:

– overview of machine socialization;

– requirements for machine socialization; and

– reference models of machine socialization including a service model and functional model.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T Y.4000] Recommendation ITU-T Y.4000/Y.2060 (2012), *Overview of the Internet of things*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 Internet of things (IoT) [ITU-T Y.4000]: A global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies.

NOTE 1 – Through the exploitation of identification, data capture, processing and communication capabilities, the IoT makes full use of things to offer services to all kinds of applications, whilst ensuring that security and privacy requirements are fulfilled.

NOTE 2 – From a broader perspective, the IoT can be perceived as a vision with technological and societal implications.

3.1.2 thing [ITU-T Y.4000]: In the Internet of things, this is an object of the physical world (physical things) or of the information world (virtual things), which is capable of being identified and integrated into communication networks.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 machine: An object of the physical world which is capable of being identified and of communicating, computing and processing data.

3.2.2 machine socialization: Enabling things or machines to communicate what they do or what they need, as well as to follow one another, discuss with one another and collaborate with one another.

3.2.3 relation: An association between or among machines or things enabling machines or things to share or to provide the capability to achieve a task in collaboration. This includes scheduling of processes between or among machines or things to perform a task.

3.2.4 sociality: The tendency of things or machines to be in the state of socialization.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

E-R Entity Relationship

IoT Internet of Things

M2M Machine to Machine

QoS Quality of Service

RFID Radio Frequency Identification

XML Extensible Markup Language

5 Conventions

The keywords "is required to" indicate a requirement which must be strictly followed and from which no deviation is permitted if conformance to this Recommendation is to be claimed.

The keywords "is recommended" indicate a requirement which is recommended but which is not absolutely required. Thus this requirement need not be present to claim conformance.

6 Overview of machine socialization

6.1 General overview of machine socialization

The Internet of things (IoT) is defined as a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies. Through the exploitation of identification, data capture, processing and communication capabilities, the IoT makes full use of things to offer services to all kinds of applications, while ensuring that security and privacy requirements are fulfilled. From a broader perspective, the IoT can be perceived as a vision with technological and societal implications [ITU-T Y.4000]. Other definitions on the IoT can be found, however they do not have any significant differences.

According to the existing definitions of the IoT, things or machines collect data (either environmental or non-environmental) and transfer it to the information world through communication networks. In other words, current understating of the IoT is reduced to merely a collection of world-wide sensor networks and radio frequency identification (RFID) systems and global machine-to-machine (M2M) systems. Though things or machines are interconnected with one another, the point of interest of the IoT is in providing capability for communication and data capture to things or machines. However, expectations for the IoT go beyond sensor networks, RFID and M2M, etc., as these are just some of the enablers for the IoT.

Like the human experience of using social network services to obtain information on the knowledge and capabilities of other people, unrestricted by barriers of time and place, etc., machines can communicate and say what they do or what they need, they can follow one another, discuss, collaborate, create events and do things together. This involves the socialization of machines to a level corresponding to that of the social relations of humans.

Figure 1 depicts a conceptual model of machine socialization. In machine socialization, machines are capable of basic communication and computing. For machine socialization, machines should at least be able to discover other machines and obtain information about the properties of other machines such as capability (service that the machine can provide) and interface.

In Figure 1, M3 locates in the home and office whereas M2 locates in public and the home. When M3 locates in the office, M3 has M4 and M5 as its neighbourhood. M4 and M5 have different properties from the properties of M1 and M2 which locate in M3's home. If M3 is socialized with M4 and M5, M3 is able to collaborate with M4 and M5.

M3 can do different jobs when M3 is socialized with M4 and M5 compared to the socialization with M1 and M2 because M4 and M5 provide different services. M2 also has M1, M3 and M6 and M7 as its neighbourhood in home and public. From the socialization with both neighbourhoods in home and public, M2 can do different jobs when it locates in home and in public respectively.

The capability of machines can be extended through socializations supporting collaboration, and machines can be socialized with many other types of machines. This means that machines can extend their capability in different ways using various socializations.

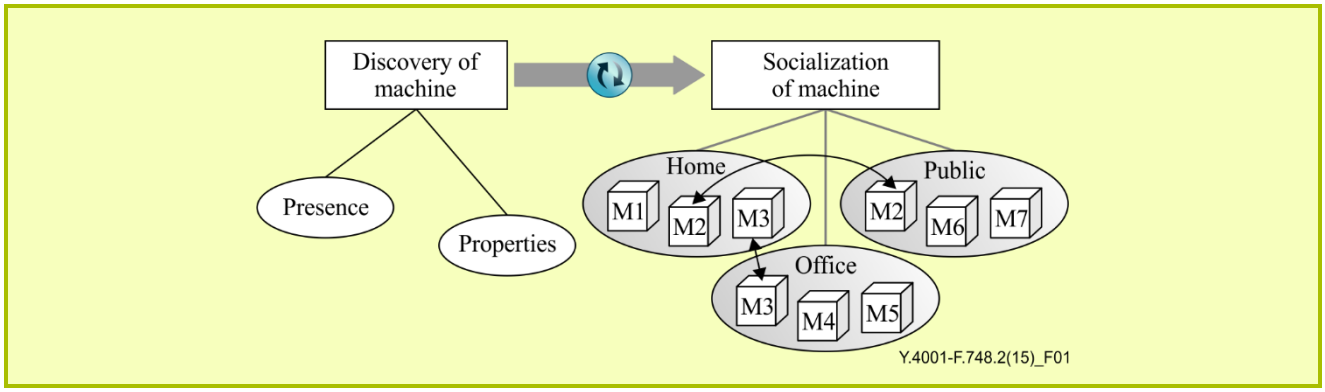


Figure 1 – Conceptual model of machine socialization

6.2 Relations for socialization

As defined in clause 3, a relation is an association between or among machines to share or provide capability. A relation also specifies the schedule of processes between or among machines while performing the task in collaboration.

Establishing a relation enables machines to collaborate with other machines in a form of capabilities sharing.

Figure 2 depicts an entity relationship (E-R) diagram of socialization which associates two machines by a relation.

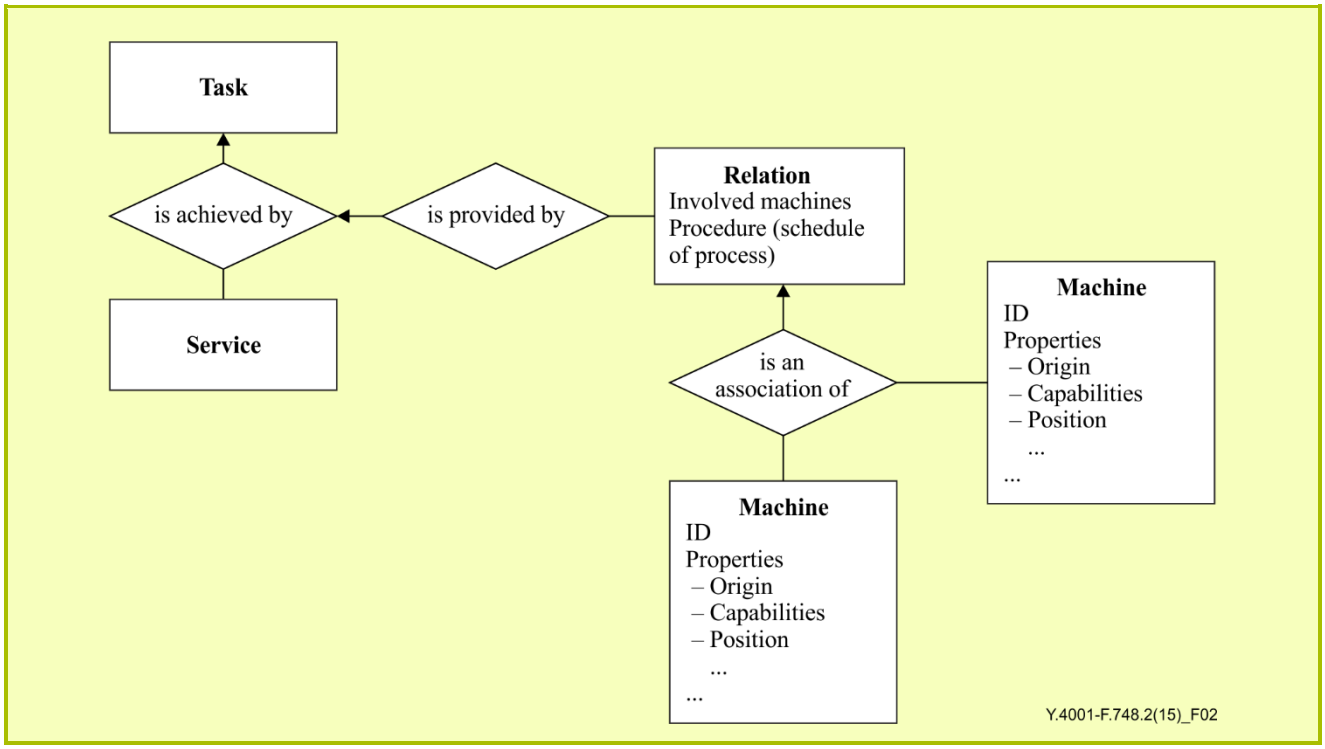


Figure 2 – E-R diagram of socialization

As shown in Figure 2, socialization can be established by establishing a relation or relations. A relation is an association between or among machines to enable a machine to expose its capabilities to other machines for collaboration. Once a machine is associated with other machine(s) as a relation, their capabilities can be exchanged to achieve a given task.

Figure 3 presents an E-R diagram of a relation as an example. In this example, three machines are associated as a relation. Each machine has different properties and capabilities. This relation includes machine information of machines that are involved in socialization and also includes procedures to be carried out in each machine to achieve a given task. This procedure defines sequential actions for each machine and the relationship between or among capabilities of the machines.

In a relation, output from one machine can be transferred to another machine as an input. Display mirroring is an example of this property. Some vehicles can be associated with a smart phone for mirroring a smart phone's display. In this case, navigation information can be displayed in a vehicle with the aid of a smart phone, even if the vehicle does not have a navigation system.

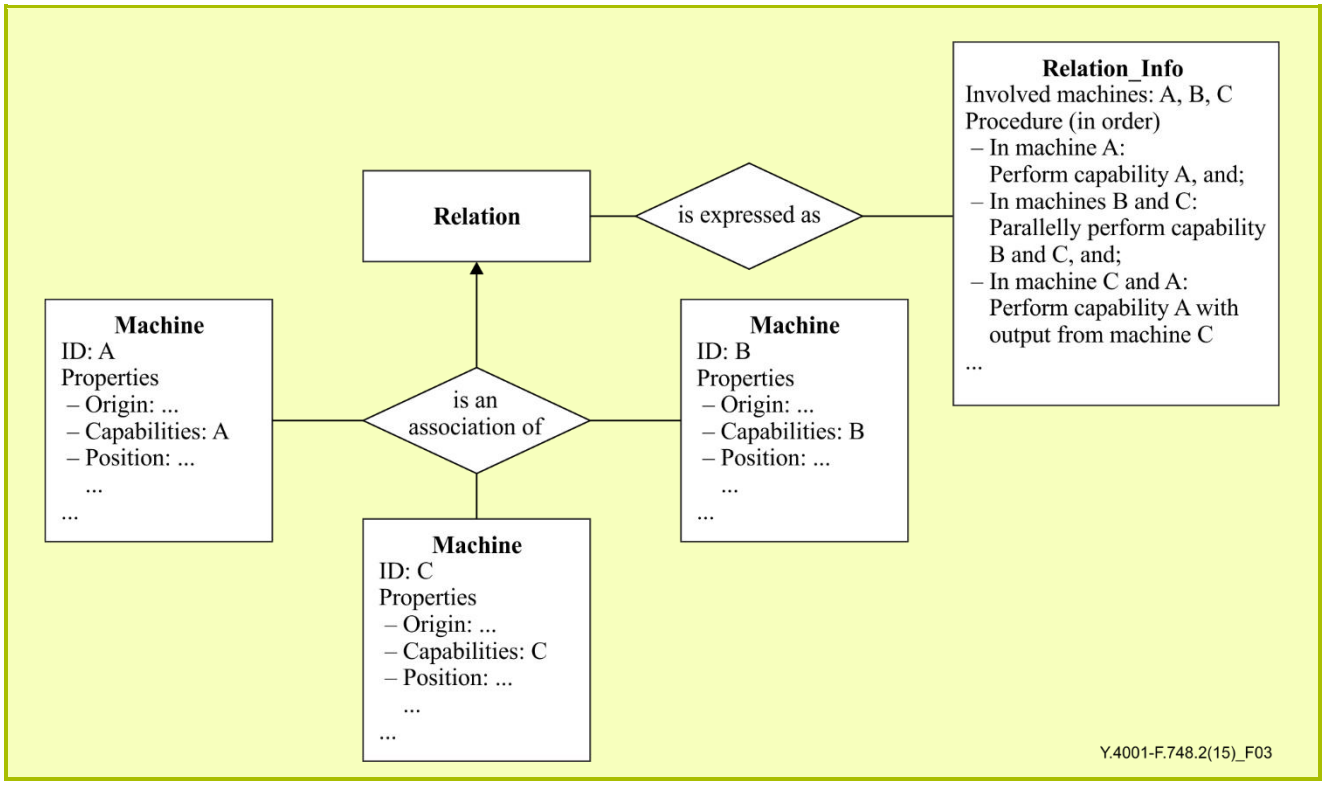


Figure 3 – E-R diagram of relation (example)

6.3 Socialization under the same ownership of machines

Typically, machine socialization is established between or among machines which are under the same ownership. In the case of machine socialization under the same ownership, particular authentication and authorization of access to a machine is not necessary.

6.4 Socialization under different ownerships of machines

A user of a machine can configure his/her machine to expose its capability to other machines which are under different owners, or vice versa. When machine socialization is necessary between or among machines which are under different ownerships, particular authentications and authorizations are needed with respect to machine socialization under the same ownership.

6.5 General procedures of machine socialization

Figure 4 shows socialization procedures.

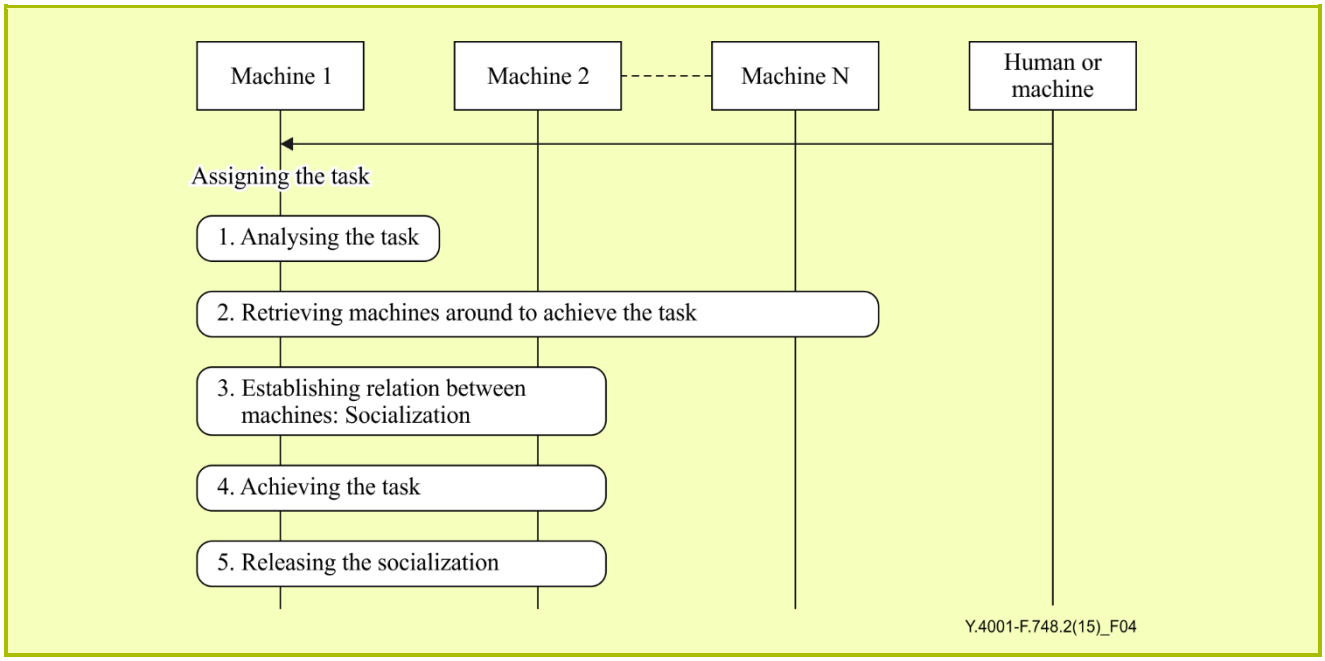


Figure 4 – Socialization procedures

A user can be a human or a machine. When a user assigns a task to a machine, the machine analyses the task. Through an analysis of the task, the machine obtains information about the capabilities needed to achieve the task. If capabilities that are needed are not supported by the machine, the machine starts to retrieve machines to provide those capabilities. Once the machine finds another machine to provide the capability, the machine tries to establish a relation with that machine and achieve the task through that relation. After achieving the task, the relation is released and socialization is also released.

7 Requirements for machine socialization

This clause describes requirements for machine socialization from an application point of view; therefore, communication specific requirements are not covered.

7.1 Standardized description of a machine

In a machine socialization, a machine has to find other machines from the perspective of their capability with which to be associated by a relation. To do this, the machine is required to present its machine capability(s) in a standardized way. Machine presentation is used to perform service discovery.

7.2 Service discovery

For a machine to find other machines with the necessary capabilities, service discovery is required. Through service discovery, a machine can find other machines to be associated with.

7.3 Standardized expression of relation

It is required to express relation information in a standardized form, for example as an extensible markup language (XML) schema. Relation information encompasses the machines involved, their association information with other machines and task information which is given to the machines, etc.

7.4 Dynamic update of relation

Once a relation is established among machines under a given task, it is required to update a relation in runtime. This includes an update of the association status (leaving or joining the association of a machine) and an update of a given task.

7.5 Multiple ways of establishing a relation

When a task is given to a particular machine, that machine is required to be capable of establishing relations with other machines in various ways. These may include that a separate object (server) analyses the task to determine machines with which it should be associated, or these procedures may be carried out by the machine itself.

7.6 Caching of relation information

A device may have patterns to establish a relation in a specific area such as a home or an office where neighbouring devices are seldom changed. In this case, it is recommended to maintain or cache relation information in a device after accomplishing the task for rapid re-establishment of the relation.

7.7 Fault recovery for a relation

When a fault occurs in a device performing a task in a form of machine socialization, it is recommended to recover the relation information after fault recovery in the machine.

7.8 Resilience of relation

When a fault occurs in a relation, it is required to isolate the faulty device from a relation to keep a relation unaffected by the fault. The task performed by the faulty device can be taken over by another machine if available.

7.9 Negotiation of QoS

In machine socialization, tasks are allocated to each machine in a relation according to capability. However, this does not mean that a machine can satisfy the full level of quality of service (QoS) for the given task. Therefore, it is required to be able to negotiate QoS when establishing a relation.

7.10 Verification of ownership of a machine

A relation can be established under both the same ownership of machines and under different ownerships of machines. For this reason, it is required to verify the ownership of a machine.

8 Reference models of machine socialization

The objective of machine socialization is to enable things to communicate what they do or what they need, follow one another, discuss with one another, collaborate, create events and do things together. This clause describes reference models of machine socialization including a service model and functional model.

8.1 Service model of machine socialization

Feasible services by machine socialization may be varied and numerous from simple services such as display mirroring to complex services in which different functionalities are utilised by multiple socialized participants.

However, a service model of machine socialization can be considered as one providing any services through relations between different machines.

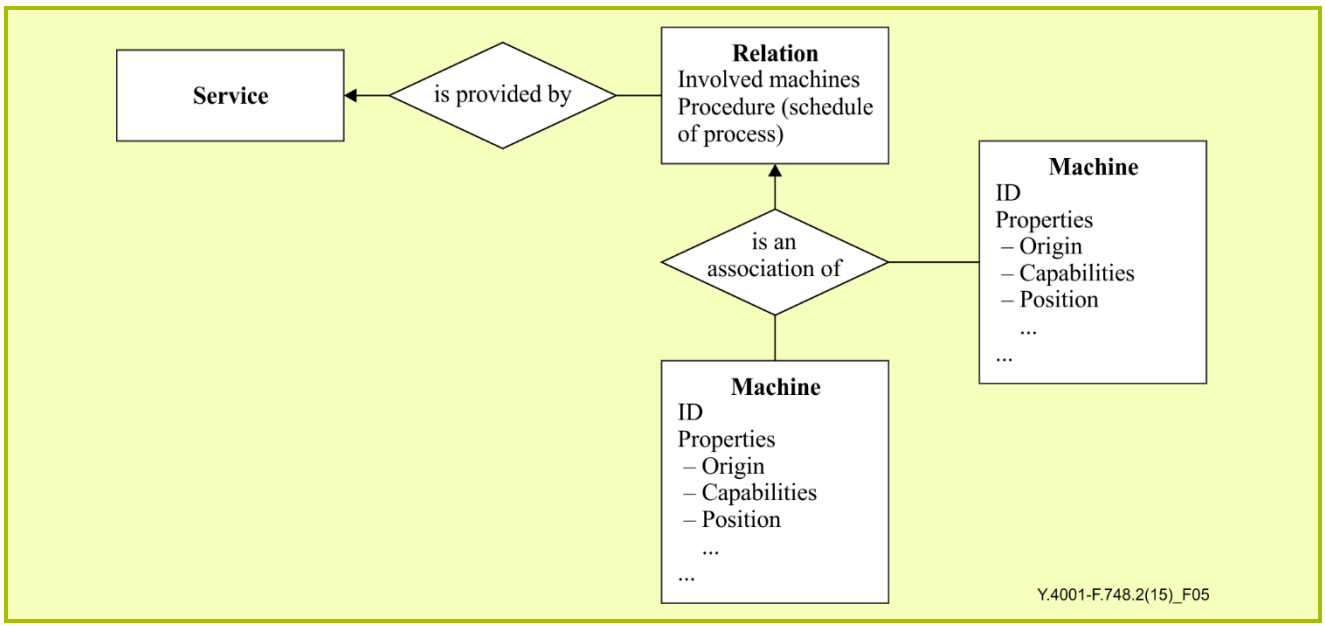


Figure 5 – Abstract service model

Figure 5 depicts a service model of machine socialization. Machine socialization is a procedure of establishing relations among different machines to make machines communicate with one another what they do or what they need, follow one another, discuss with one another and collaborate with one another.

Characteristics of a relation depend on the characteristics of services to be provided. For example, display mirroring in a vehicle between a smart phone and display unit of a vehicle is provided by a simple relation of display capability. In the case of a complex service, relations may be complex where various capabilities of different machines are associated.

8.2 Functional model of machine socialization

From a functional viewpoint, machine socialization is a process of establishing relations as explained above.

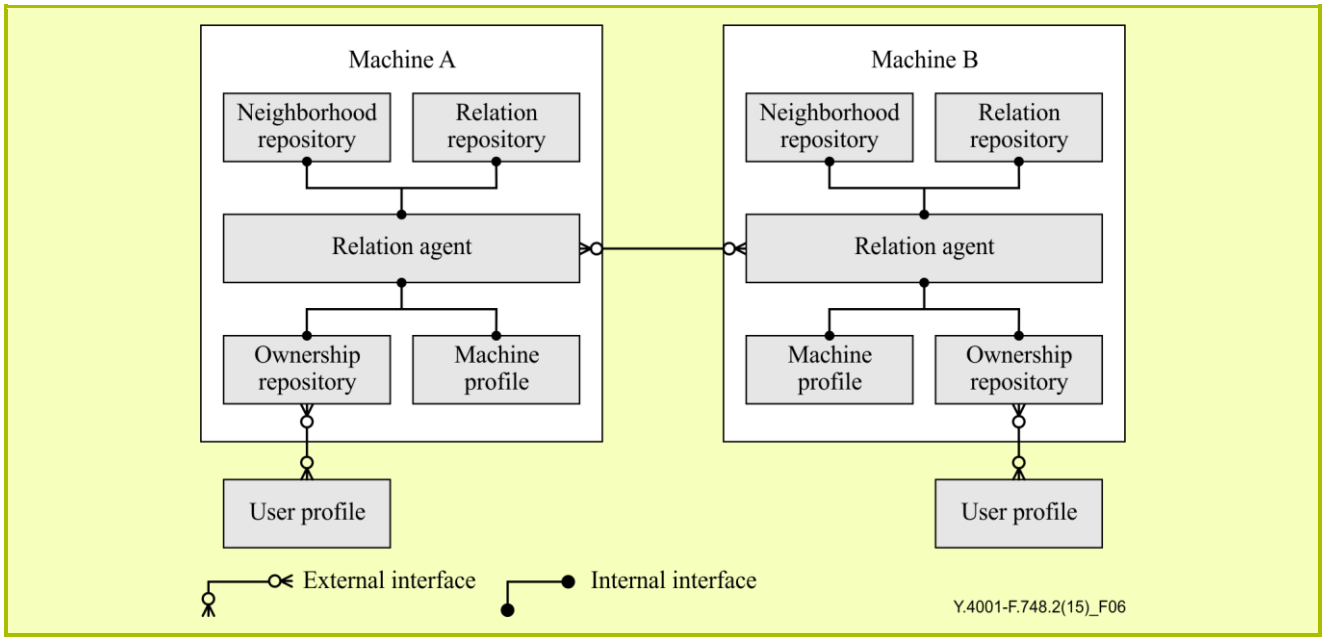


Figure 6 – Functional model

Figure 6 depicts a functional model of machine socialization. Each machine has internal function blocks and internal interfaces as well as outgoing interfaces with external entities such as user profile or other machines.

8.2.1 Machine profile

Machine profile maintains capabilities with a standardized description of a machine as defined in clause 7.1. Machine profile is used to negotiate QoS as described in clause 7.9.

8.2.2 Neighbourhood repository

A neighbourhood repository stores information of machines with which associations are needed. Once a relation agent discovers the machines with the necessary capabilities which are needed for accomplishing the given task, as described in clause 7.2, the information of those machines is stored in a neighbourhood repository.

8.2.3 Relation repository

When a relation is established, the relation is represented as a standardized expression as described in clause 7.3. A relation repository maintains standardized expressions of established relations. Dynamic update of a relation as described in clause 7.4, caching of relation information as described in clause 7.6, fault recovery and resilience of relation as described in clauses 7.7 and 7.8, respectively are carried out on this relation repository.

8.2.4 Relation agent

A relation agent performs service discovery as described in clause 7.2, relation establishment and management are carried out as described in clauses 7.3, 7.4, 7.5, 7.6, 7.7, 7.8 and 7.9. A relation agent may perform limited functions in the case where preparation of a relation is carried out by a separate object (server) as described in clause 7.5.

8.2.5 Ownership repository

An ownership repository maintains the ownership information of a machine. When a relation is established, the ownership of a machine should be verified as described in clause 7.10. The ownership repository is involved in verification of ownership. The ownership repository may also interface with outside user profiles to check permissions for the establishment of a relation with the different ownerships of machines. In this case, a user profile outside the machine maintains the user's permission information for a relation.


INTERNET
OF THINGS

Y.4002/F.748.3

Machine socialization: Relation management models and descriptions

Machine socialization: Relation management models and descriptions

Summary

Recommendation ITU-T Y.4002/F.748.3 specifies the relation management models and descriptions for machine socialization which enables machines to cooperate to achieve a given task using their relations with other machines. This Recommendation also includes use cases of relation management models and presents schemas to describe a relation.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T Y.4002/F.748.3	2015-11-29	16	11.1002/1000/12622

Keywords

Machine socialization, relation management model.

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

Table of Contents


W
W
W
90
ON
OF

Recommendation ITU-T Y.4002/F.748.3

Machine socialization: Relation management models and descriptions

1 Scope

This Recommendation specifies the relation management models and descriptions for machine socialization which enables machines to cooperate to achieve a given task using their relations with other machines. This Recommendation covers the following:

– relation management models for machine socialization;

– relation descriptions for machine socialization; and

– use cases for relation management models.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T Y.4001] Recommendation ITU-T Y.4001/F.748.2 (2015), *Machine socialization: Overview and reference model*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 Internet of things (IoT) [b-ITU-T Y.4000]: A global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies.

NOTE 1 – Through the exploitation of identification, data capture, processing and communication capabilities, the IoT makes full use of things to offer services to all kinds of applications, whilst ensuring that security and privacy requirements are fulfilled.

NOTE 2 – From a broader perspective, the IoT can be perceived as a vision with technological and societal implications.

3.1.2 machine socialization [ITU-T Y.4001]: Enabling things or machines to communicate what they do or what they need to each other as well as to follow each other, discuss with each other and collaborate with each other.

3.1.3 relation [ITU-T Y.4001]: An association between or among machines or things enabling machines or things to share or to provide the capability to achieve a task in collaboration. This includes scheduling of processes between or among machines or things to perform a task.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms.

3.2.1 capability parameter: Information on a capability of a machine to achieve a given task such as an input parameter, an output parameter, processing time, etc.

3.2.2 capability set: A set of capabilities that machines can provide to process commands from users.

3.2.3 machine profile schema: A schema to describe features of a machine.

3.2.4 relation module: A module that acts as a relation server in the distributed relation management model. A relation module resides in every machine in the distributed relation management model.

3.2.5 relation profile schema: A schema to describe a relation to achieve a given task. It contains information on a group of machines, processing schedules, workgroup IDs, work descriptions and a capability set.

3.2.6 relation server: A server to establish a relation and operate machines according to the relations.

3.2.7 status parameter: The current status of a machine such as active, standby, expected termination time of the current operation, current execution task, current execution function and expected termination time of the current execution function.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

XML Extensible Markup Language

5 Conventions

None.

6 Overview of relation for machine socialization

Like the human experience of using social network services to obtain information on the knowledge and capabilities of other people, unrestricted by barriers of time and place, etc., machines can communicate what they do or what they need; they can follow each other, discuss, collaborate, create events and do things together. This involves the socialization of machines to a level corresponding to that of the social relations of humans [ITU-T Y.4001].

Socialization may have many meanings, but one of the most important features of socialization is working together. Machine socialization implicates that machines work together to achieve given tasks. To achieve the task in collaboration, machines have to establish associations with other machines by sharing and providing capabilities, these associations are called relations in [ITU-T Y.4001] and in this Recommendation.

In other words, machines should exchange data with other machines and understand the meaning of the exchanged data to establish a relation.

Humans can exchange their ideas with other people using languages or gestures, however a machine cannot exchange information with other machines using natural languages. Therefore, a machine should exchange information and respond to other machines by analysing the information with other machines that is expressed in a mutually understandable form, such as extensible markup language (XML).

This Recommendation defines a schema for a mutually understandable form of machine socialization.

If machines can understand and exchange data with other machines, a relation can be established through relation management.

Relation management establishes a relation, operates machines using the established relation and releases the relation as specified by relation management models.

Procedures to establish a relation vary according to relation management models.

7 Relation management models

For machine socialization, it is necessary to manage relations among machines. Relation management varies depending on the relation models. Relation management models may require user intervention in cases where the results of tasks are critical to the user.

In this Recommendation, relation management models define creation and release of relations in three different ways based on the machine socialization framework defined in [ITU-T Y.4001].

7.1 Centralized relation management model

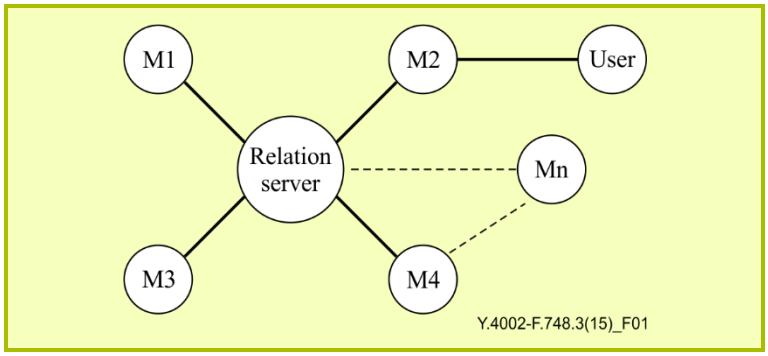


Figure 1 – Centralized relation management model

In the centralized relation management model shown in Figure 1, a relation server manages relations among machines and executes a task following the steps listed below:

1) Machines register machine profile parameters to the relation server.
2) A machine receives a user's command to execute a task and forwards the task to the relation server.
3) The relation server analyses the task.
4) The relation server establishes a capability set to execute the task.
5) The relation server establishes a group of useable machines to execute the task based on the capability set, capability parameters and status parameters of the machines that are already registered on the server or extractable from the target machines.
6) The relation server establishes a relation for the group based on the capability set, the capability parameters and the status parameters of the member machines.
7) The relation includes the group of machines and the schedules of the processes necessary to perform the task.
8) The relation server establishes a relation profile based on the relation.
9) If the results of the processes in the relation profile are critical to the user, the relation server should request a user intervention and update the relation profile:
   a) The relation server requests the user to approve the performance of the task or to select processes by listing processes executed by the machines.
   b) The user can either approve the performance of the task or select the processes to permit the performance of the task and then notifies the relation server.

c) The relation server may establish a new group of machines and a new relation based on the user's decision.

d) The relation server establishes a new relation profile.

10) The relation server commands the machines to execute the processes following the schedule of processes included in the relation profile.

11) After completion of the task, the relation server releases the machines from the relation.

7.2 Distributed relation management model

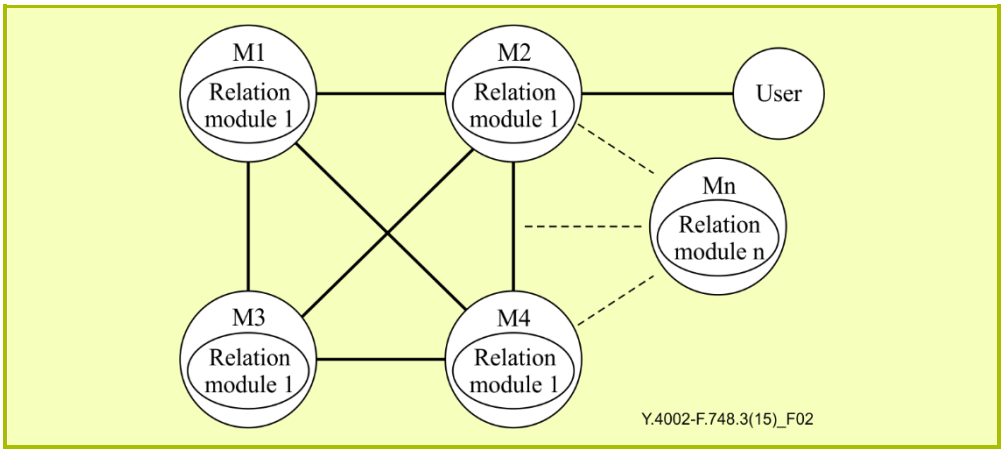


Figure 2 – Distributed relation management model

In the distributed relation management model shown in Figure 2, a relation is established by a relation module that resides in each individual machine. The machine which receives a user's command to perform a task forwards the task to other machines to create a relation. Each machine should determine whether to join the relation or not to execute the task. The machine that received the user's command acts as the coordination machine to create a relation based on the capability sets, capability parameters and status parameters of the group of machines in the relation.

The relation module creates and manages the relation and executes a task following the steps listed below:

1) A machine receives a user's command to execute a task from the coordinating machine and forwards the task to the other machines.

2) The relation module in each individual machine analyses the task and makes its own capability set to execute the task.

3) Each relation module decides whether to join the group of machines to execute the task, based on its capability set, capability parameter and status parameter.

4) If a relation module decides to join the group, the relation module shall notify the coordination machine and send its capability set, capability parameters and status parameters to the coordinating machine.

5) The coordinating machine's relation module creates a relation based on the received capability sets, capability parameters and status parameters of the group of machines.

6) The relation includes the group of machines and the schedule of processes necessary to perform the task.

7) The coordinating relation module creates the relation profile and forwards the relation and the relation profile to the member machines.

8) If the results of processes in the relation profile are critical to the user, the machine that received the user's command should request a user intervention and update the relation profile:
   a) The coordinating machine requests the user to approve performance of the task or to select processes by listing processes executed by the machines.
   b) The user can either approve performance of the task or select the processes and then notifies the coordinating machine.
   c) The coordinating machine notifies the user's decision to other machines.
   d) The relation modules in individual machines make new decisions to join the relation and forward necessary information to the coordinating machine.
   e) The coordinating relation module creates a new group of machines and a new relation based on the user's decision.
   f) The coordinating relation module establishes a new relation profile.
9) Grouped machines should operate by following the schedule of processes included in the relation profile.
10) After completion of its own task, each machine releases itself from the relation.

7.3 Nested-centralized relation management model

The nested-centralized relation management model extends the centralized relation management model.

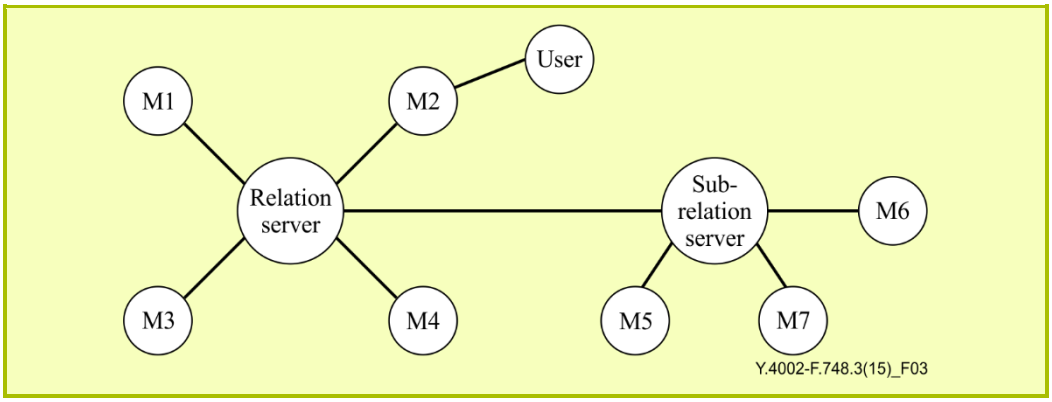


Figure 3 – Nested-centralized relation management model

Figure 3 shows an example of the nested-centralized relation management model where seven machines and two relation servers are involved.

The model manages a relation and executes a task following the steps listed below:

1) A machine registers machine profile parameters to either the relation server or to the sub-relation server.
2) A machine receives a user's command to execute a task and forwards the task to the relation server.
3) The relation server forwards the task to the sub-relation server.
4) The relation server and the sub-relation server analyse the task.
5) The relation server and the sub-relation server establish a capability set to execute the task.

6) The relation server and the sub-relation server create a group of useable machines based on the capability set, capability parameters and status parameters of the member machines that are already registered on the server or extractable from the target machines.

7) The sub-relation server establishes a relation of a group of machines and forwards it to the relation server.

8) The relation server establishes a relation using its group of machines and the relation received from the sub-relation server.

9) The relation includes information of the group of machines and the schedule of processes.

10) The relation server establishes a relation profile according to the relation.

11) If the results of processes in the relation profile are critical to the user, the relation server should request a user intervention and update the relation profile:

   a) The relation server requests the user to approve performance of the task or to select processes by listing processes executed by the machines.

   b) The user can either approve the performance of the task or select the processes and then notifies the machine.

   c) The relation server may establish a new group of machines and a new relation based on the user's decision.

   d) The relation server establishes a new relation profile.

12) The relation server forwards the relation profile to the sub-relation server.

13) The relation server and the sub-relation server command a group of machines to execute processes following the schedule of processes included in the relation profile.

14) After the completion of the task, the relation server and the sub-relation server release the machines from the relation.

8 Relation descriptions

The machine profile schema enables the relation server or relation module to mutually understand the features of machines to establish a relation and the relation profile schema stores the relation established.

8.1 Machine profile schema

The machine profile schema is a template to describe machines in terms of status, capabilities, IDs, interfaces, etc., which are given in Table 1. A relation is established by a relation server or a relation module with the information inscribed using this profile.

Table 1 – Machine profile schema

Machine profile parameter	Sub-parameter
Status	Active
	Standby
	Current process
	Expected termination time of the current process
	Current function
	Expected termination time of the current execution function
Capability	Name of process
	Input parameter

Table 1 – Machine profile schema

Machine profile parameter	Sub-parameter
	Output parameter
	Processing time (duration)
	Processing condition
	Function 1
	Function 2
	…….
	Function n
	End of capability
Machine ID	
User ID	
Group ID	
Operating system	
Machine interface	Interface protocol between machines
	Interface parameter between machines
	Interface protocol between machine and relation server
	Interface parameter between machine and relation server
End of machine profile parameter	

Machine profile parameters are described below:

– Status: Status of a machine indicating active, standby, the expected termination time of the current process, the current execution process, the current execution function and the expected termination time of the current execution function.

– Capability: Capability of a machine indicating the name of the task, input parameter, output parameter, processing time, processing condition and functions.

– Machine ID: Alphanumeric identifier of a machine which distinguishes a machine from another machine.

– Group ID: Alphanumeric identifier of a group of machines in a relation.

– Operating system: Operating system running on a machine.

– Machine interface: Communication protocols that a machine can support.

– End of machine profile parameter: Indicator to specify the end of the machine profile.

8.2 Relation profile schema

The relation profile schema shown in Table 2, is a template to describe a relation in terms of capability set, grouped machines, workgroup ID, task description and task processing schedule, etc. After a relation is established, relation information is stored in a relation profile. Tasks to be executed by a relation in machine socialization are expressed in a relation profile and are achieved according to the schedule of processes defined in the relation.

Table 2 – Relation profile schema

Relation profile parameter	Sub-parameter
Capability set	Capability 1
	…
	Capability n
	End of capability set
Grouped machines	Machine ID 1
	….
	Machine ID n
	End of machine ID
Workgroup ID	
Task description	
Task processing schedule	Process 1
	Process start time
	Process start condition
	Allotted machine ID
	Start time of function 1
	Functions 1
	….
	Start time of function n
	Function n
	End of functions
	Expected time of process termination
	Interface parameters
	Termination condition
	End of process 1
	Process 2
	Process start time
	Process start condition
	Allotted machine ID
	Start time of function 1
	Functions 1
	….
	Start time of function n
	Function n
	End of functions
	Expected time of process termination
	Interface parameters
	Ending condition
	End of process 2
	…..

Table 2 – Relation profile schema

Relation profile parameter	Sub-parameter
	Process n
	Process start time
	Process start condition
	Allotted machine ID
	Start time of function 1
	Functions 1
	….
	Start time of function n
	Function n
	End of functions
	Expected time of process termination
	Interface parameters
	Termination condition
	End of process n
	End of task processing schedule
End of relation profile parameter	

Relation profile parameters are described below:

– Capability set: The set of capabilities needed to complete the user's command to perform a task

– Grouped machines: The group of machines which support the capability set.

– Workgroup ID: Alphanumeric identifier of the workgroup.

– Task description: Human readable description of the task implying a user's command.

– Task processing schedule: Sequence of processes to complete the task.

– End of relation profile parameter: Indicator to specify the end of relation profile parameter.

Appendix I

Use cases of relation management models

(This appendix does not form an integral part of this Recommendation.)

Three use cases of relation management models are introduced.

I.1 Booking of a movie ticket based on a centralized relation management model

The use case shown in Figure I.1 illustrates the booking of a movie ticket using a centralized relation management model of machine socialization.

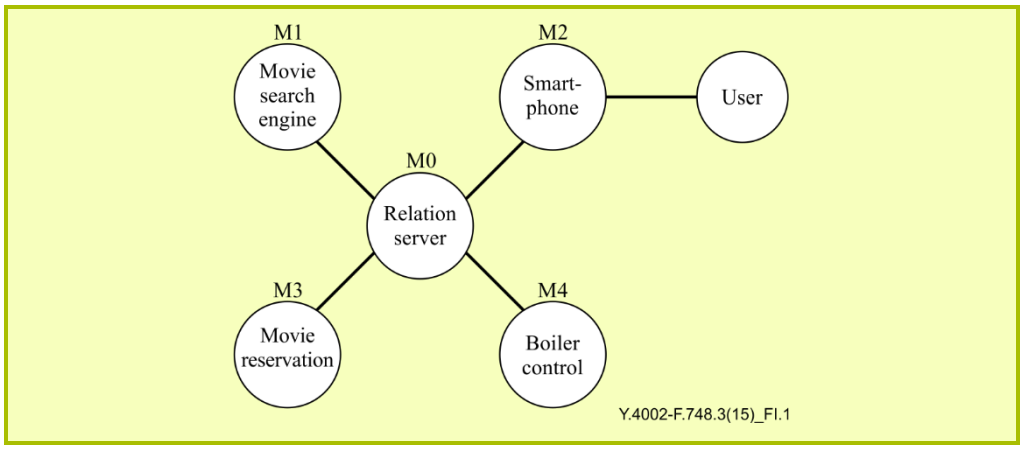


Figure I.1 – Centralized relation management model

Each machine can perform the following processes:

– M0: M0 manages relations among machines and executes a task.

– M1: M1 searches movies.

– M2: M2 receives the user's command to perform a task, booking a movie ticket. It forwards the task to the relation server.

– M3: M3 books movie tickets.

– M4: M4 controls the temperature of the house.

A relation is established among M0, M1, M2, M3 and M4 according to the following procedures:

1) M1, M3 and M4 have registered their machine profiles including searching capability, booking capability and controlling temperature capability to the relation server respectively.

2) M2 receives a user's command to perform a task and forwards the task to M0.

3) M0 analyses the task and determines a capability set which requires searching for a movie, booking the movie and controlling temperature. Controlling temperature capability will be used to ensure a comfortable environment for a user after watching a movie.

4) M0 creates a group of M1, M3 and M4 including M2 which forwarded the user's commands to M0.

5) With the assumption that the results of this task are not critical, M0 establishes a relation based on the capability set, the capability and the status in machine profiles of M1, M3 and M4 according to the following schedule.

6) Activation of M0, M1, M2, M3 and M4 by using the relation is as follows:

   a) M0 sends a command to M1 to search for the specified movie;

b) M1 returns the result to M0 and M0 forwards the result to M2;

c) The user selects a movie and the selection is forwarded to M3;

d) M3 books a movie ticket with the confirmation of the user;

e) M0 activates M4 at the time according to the schedule of the process;

f) The relation is released.

I.2 Booking of a movie ticket based on a distributed relation management model

The use case shown in Figure I.2 illustrates the booking of a movie ticket based on a distributed relation management model for machine socialization.

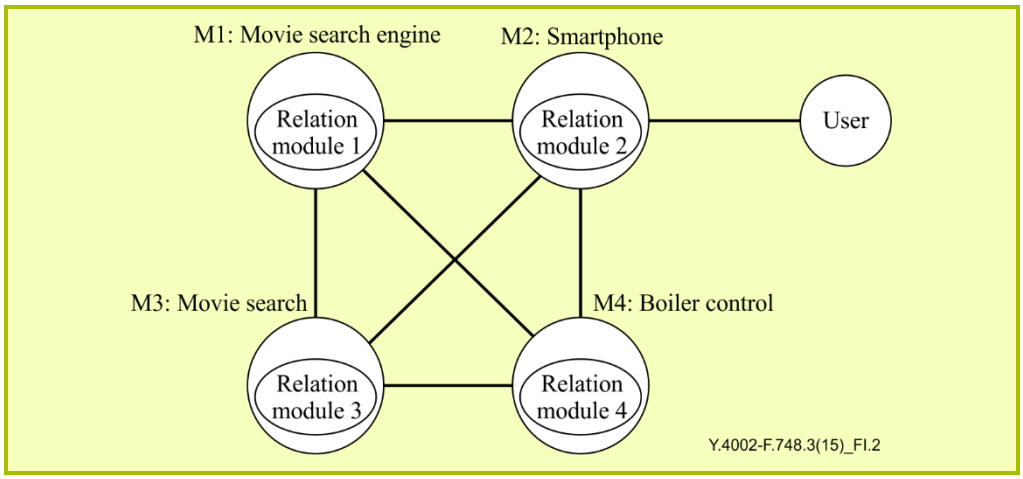


Figure I.2 – Distributed relation model

Each machine can perform the following processes:

– M1: M1 searches for a movie.

– M2: M2 is a coordinating machine and receives the user's command to perform a task, booking a movie ticket. It forwards the task to the other machines.

– M3: M3 books movie tickets.

– M4: M4 controls the temperature of the house.

A relation is established among M0, M1, M2, M3 and M4 according to the following procedure:

1) M2 receives a user's command to perform a task and forwards the task to the M1, M3 and M4.

2) The relation modules in the individual machines analyse the task and establish a capability set to execute the task.

3) The relation modules decide whether to join the group of machines for executing the task coming from the user based on the capability set and its own machine capability parameter and machine status parameter.

4) If the relation module decides to join the relation, it notifies M2 and forwards its capability set, the capability parameters and the status parameters.

5) M2 creates a group of machines, a relation and a relation profile.

6) The relation includes a group of machines and the schedule of processes to perform the task.

7) With the assumption that the results of this task are not critical, the created relation and the relation profile are forwarded to M1, M3 and M4.

Activation of M1, M2, M3 and M4 by using the relation is as follows:

a) M1 searches for the specified movie according to the relation profile;

b) M1 returns the result to M2 and M2 forwards the result to the user;

c) The user selects a movie and the selection is forwarded to M3;

d) M3 books a movie ticket with the confirmation of the user;

e) Relation module 4 in M4 activates M4 at the time according to the schedule of processes;

f) The relation is released.

I.3 Booking of a movie ticket based on a nested-centralized relation management model

The use case shown in Figure I.3 illustrates the booking of a movie ticket based on the nested-centralized relation management model for machine socialization.

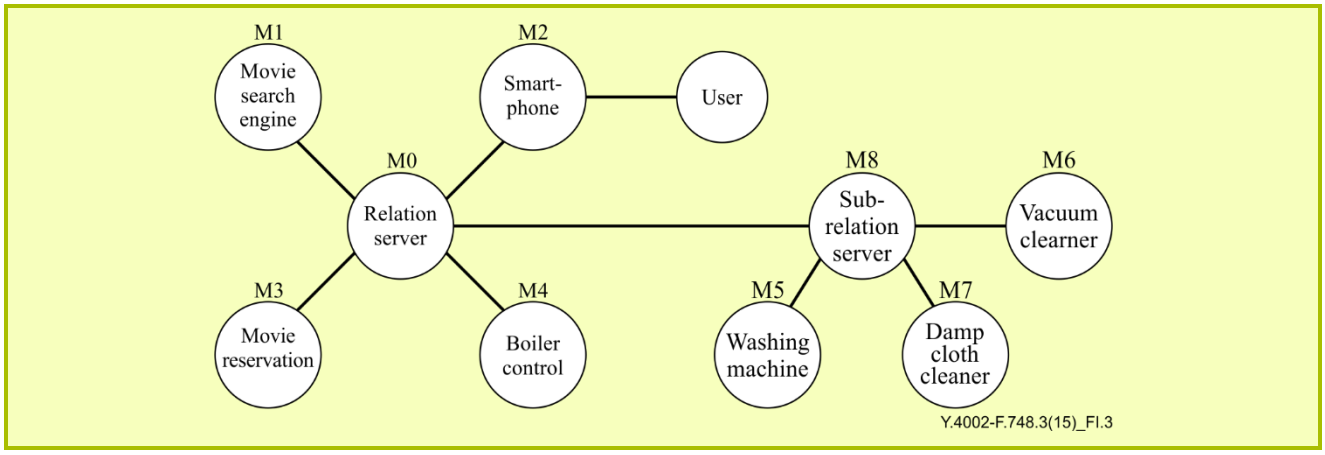


Figure I.3 – Nested-centralized relation management model

Each machine can perform the following processes:

– M0: M0 manages relations among machines and executes a task.

– M1: M1 searches for movies.

– M2: M2 receives the user's command to perform a task, booking a movie. It forwards the task to the relation server.

– M3: M3 books movie tickets.

– M4: M4 controls the temperature of the house.

– M5: M5 washes clothes.

– M6: M6 performs vacuum cleaning.

– M7: M7 performs damp cloth cleaning.

– M8: M8 manages relations among submachines and executes a subtask.

A relation is established among M0, M1, M2, M3, M4, M5, M6, M7 and M8 according to the following procedure:

1) M1, M2, M3, M4, M5, M6, M7 and M8 respectively have registered their machine profiles including searching capability, booking capability and controlling temperature capability to M0.

2) M2 receives the user's command to perform a task and forwards the task to M0 and M8.

3) M8 server analyses the received task and establishes a relation.

4) M0 analyses the received task and grouping machines which can be used to execute the command by using machine profiles and the generated capability set.

5) With the assumption that the results of this task are not critical, a new relation is created in M0 based on the capability set, the capability, the status in other machine profiles and a relation which is generated by a sub-relation server.

6) M0 forwards the new relation profile to M8 server.

7) Activation of M0, M1, M2, M3, M4, M5 and M8 by using the new relation is as follows:

   a) M0 sends a command to M1 to search for the specified movie;

   b) M1 returns the result to M0 and M0 forwards the result to M2;

   c) The user selects a movie and the selection is forwarded to M3;

   d) M3 books a movie ticket with the confirmation of the user;

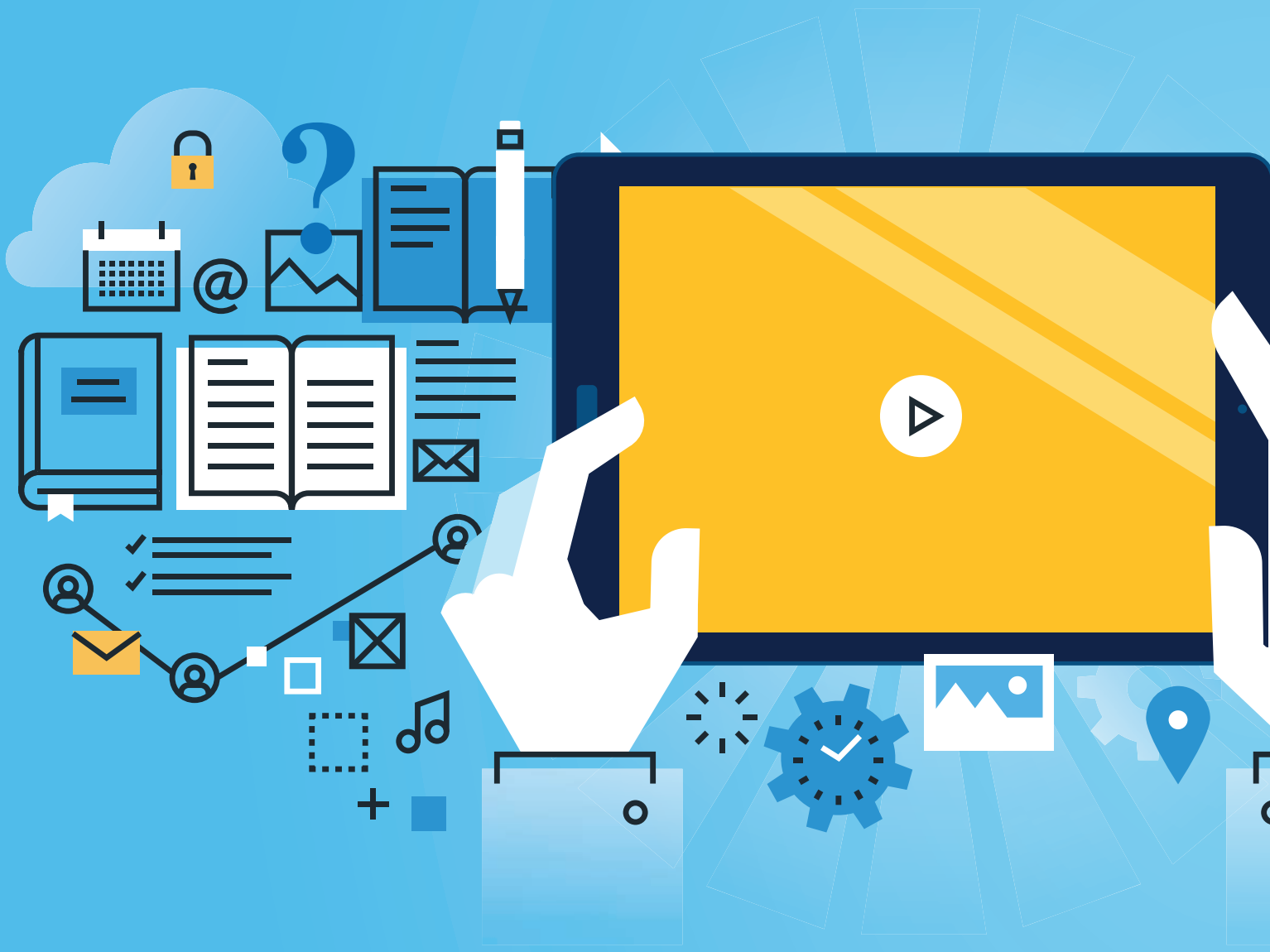
   e) M8 activates M5 at the time according to the schedule of the process;

   f) The relation is released.

Bibliography

[b-ITU-T Y.4000] Recommendation ITU-T Y.4000/Y.2060 (2012), *Overview of the Internet of Things*.

2 Definitions and terminologies



Y.4050/Y.2069

Terms and definitions for the Internet of things

Terms and definitions for the Internet of things

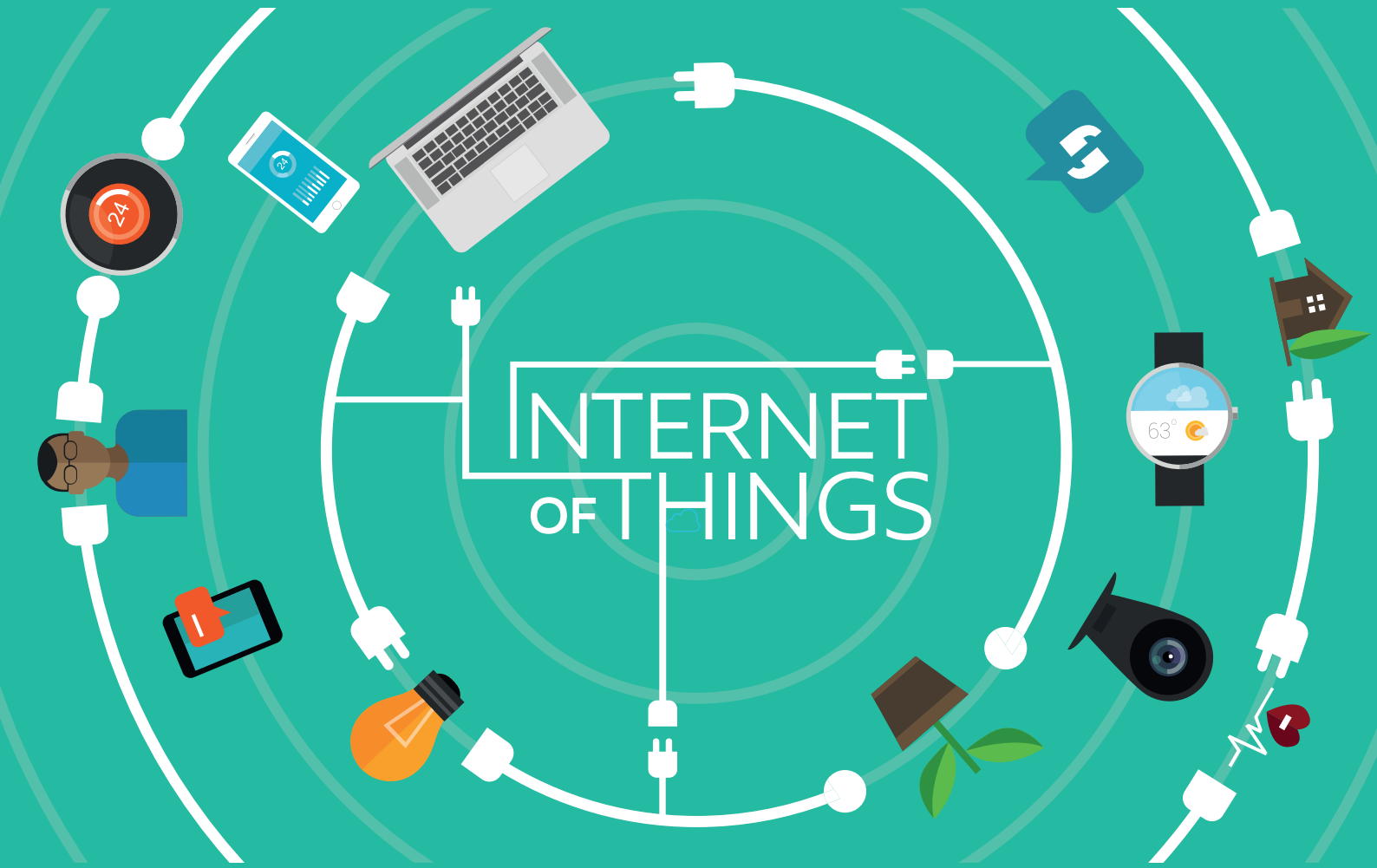
Summary

Recommendation ITU-T Y.2069 specifies the terms and definitions relevant to the Internet of things (IoT) from an ITU-T perspective, in order to clarify the Internet of things and IoT-related activities.

History

Edition	Recommendation	Approval	Study Group
1.0	ITU-T Y.2069	2012-07-29	13

Table of Contents


INTERNET
OF THINGS

Recommendation ITU-T Y.4050/Y.2069

Terms and definitions for the Internet of things

1 Scope

This Recommendation specifies the terms and definitions relevant to the Internet of things (IoT) from an ITU-T perspective, in order to clarify the Internet of things and IoT-related activities.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T F.744] Recommendation ITU-T F.744 (2009), *Service description and requirements for ubiquitous sensor network middleware*.

[ITU-T F.771] Recommendation ITU-T F.771 (2008), *Service description and requirements for multimedia information access triggered by tag-based identification*.

[ITU-T Q.1300] Recommendation ITU-T Q.1300 (1995), *Telecommunication applications for switches and computers (TASC) – General overview*.

[ITU-T Y.2002] Recommendation ITU-T Y.2002 (2009), *Overview of ubiquitous networking and of its support in NGN*.

[ITU-T Y.2060] Recommendation ITU-T Y.2060 (2012), *Overview of the Internet of things*.

[ITU-T Y.2061] Recommendation ITU-T Y.2061 (2012), *Requirements for the support of machine-oriented communication applications in the next generation network environment*.

[ITU-T Y.2063] Recommendation ITU-T Y.2063 (2012), *Framework of the web of things*.

[ITU-T Y.2091] Recommendation ITU-T Y.2091 (2011), *Terms and definitions for Next Generation Networks*.

[ITU-T Y.2213] Recommendation ITU-T Y.2213 (2008), *NGN service requirements and capabilities for network aspects of applications and services using tag-based identification*.

[ITU-T Y.2221] Recommendation ITU-T Y.2221 (2010), *Requirements for support of ubiquitous sensor network (USN) applications and services in the NGN environment*.

[ITU-T Y.2240] Recommendation ITU-T Y.2240 (2011), *Requirements and capabilities for next generation network service integration and delivery environment*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 actuator [ITU-T Y.2061]: A device that triggers a physical action following stimulation by an input signal.

NOTE (from [ITU-T Y.2061]) – As examples, an actuator might act on the flow of a gas or liquid, or electricity, through a mechanical operation. Dimmers and relays are examples of actuators. The decision to activate the actuator may come from an MOC application, human or MOC devices and gateway.

3.1.2 context [ITU-T Y.2002]: The information that can be used to characterize the environment of a user.

NOTE (from [ITU-T Y.2002]) – Context information may include where the user is, what resources (devices, access points, noise level, bandwidth, etc.) are near the user, at what time the user is moving, interaction history between person and objects, etc. According to specific applications, context information can be updated.

3.1.3 device [ITU-T Y.2060]: In the Internet of things, a piece of equipment with the mandatory capabilities of communication and the optional capabilities of sensing, actuation, data capture, data storage and data processing.

3.1.4 ID tag [ITU-T Y.2213]: A physical object which stores one or more identifiers and optionally application data such as name, title, price, address, etc.

NOTE 1 (from [ITU-T Y.2213]) – Depending on its implementation, it may have a communication capability with an ID terminal.

NOTE 2 – The same term is also defined in [ITU-T F.771].

3.1.5 ID terminal [ITU-T Y.2213]: A device with a data reading and optional writing capability which reads (and optionally writes) identifier(s) and optionally application data from/into an ID tag.

NOTE 1 (from [ITU-T Y.2213]) – The data reading (and optionally writing) capability depends on its implementation.

NOTE 2 – The same term is also defined in [ITU-T F.771].

3.1.6 identifier [ITU-T Y.2091]: An identifier is a series of digits, characters and symbols or any other form of data used to identify subscriber(s), user(s), network element(s), function(s), network entity(ies) providing services/applications, or other entities (e.g., physical or logical objects). Identifiers can be used for registration or authorization. They can be either public to all networks, shared between a limited number of networks or private to a specific network (private IDs are normally not disclosed to third parties).

NOTE – The same term is also defined in [ITU-T F.771].

3.1.7 identifier resolution [ITU-T Y.2213]: A function to resolve an identifier into associated information (see "Forward identifier resolution") and vice versa (see "Reverse identifier resolution").

NOTE – A similar term "ID resolution" is defined in [ITU-T F.771].

3.1.8 identifier scheme [ITU-T Y.2213]: It is a numbering scheme that specifies the format and structure of the identifiers used within that scheme.

3.1.9 Internet of things (IoT) [ITU-T Y.2060]: A global infrastructure for the information society enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving, interoperable information and communication technologies.

NOTE 1 (from [ITU-T Y.2060]) – From a broad perspective, the IoT can be perceived as a vision with technological and societal implications.

NOTE 2 (from [ITU-T Y.2060]) – Through the exploitation of identification, data capture, processing and communication capabilities, the IoT makes full use of things to offer services to all kinds of applications, whilst ensuring that security and privacy requirements are fulfilled.

3.1.10 machine-oriented communication (MOC) [ITU-T Y.2061]: A form of data communication between two or more entities in which at least one entity does not necessarily require human interaction or intervention in the process of communication.

3.1.11 machine-to-machine applications [ITU-T Y.2240]: Applications enabled by the communication between two or more machines that need limited or no direct human intervention.

NOTE – The major subject of [ITU-T Y.2240] is the NGN service integration and delivery environment in which M2M is considered as one of the possible use cases. At the time of approval, M2M and its definition were under study in ITU-T. The definitions of M2M may be modified according to the study.

3.1.12 meter [ITU-T Y.2061]: A device that measures and optionally records the quantity, degree, or rate of something, e.g., the amount of electricity, gas, or water used.

NOTE – (from [ITU-T Y.2061]): A meter is responsible for measuring the total amount of something consumed in a given period.

3.1.13 multimedia information [ITU-T F.771]: Multimedia information is digital information that uses multiple forms of information content and information processing, such as text, pictures, audio, video, three-dimensional panoramic pictures and digital maps to inform or entertain users.

3.1.14 multimedia information delivery function [ITU-T F.771]: A multimedia information delivery function is a function for delivering multimedia information to an ID terminal which is triggered by tag-based identification.

3.1.15 object [ITU-T Q.1300]: An intrinsic representation of an entity that is described at an appropriate level of abstraction in terms of its attributes and functions.

NOTE 1 (from [ITU-T Y.2002]) – An object is characterized by its behaviour. An object is distinct from any other object. An object interacts with its environment including other objects at its interaction points. An object is informally said to perform functions and offer services (an object which makes a function available is said to offer a service). For modelling purposes, these functions and services are specified in terms of the behaviour of the object and of its interfaces. An object can perform more than one function. A function can be performed by the cooperation of several objects.

NOTE 2 (from [ITU-T Y.2002]) – Objects include terminal devices (e.g., used by a person to access the network such as mobile phones, personal computers, etc.), remote monitoring devices (e.g., cameras, sensors, etc.), information devices (e.g., a content delivery server), products, contents, and resources.

3.1.16 open application interface [ITU-T F.744]: An interface used by USN applications to access USN middleware.

NOTE – This definition is associated with USNs, but it can be applied to the interfaces between the application layer and the service support/application support layer.

3.1.17 processed data [ITU-T F.744]: Data that is processed from raw sensed data by sensor network or USN middleware.

NOTE – This definition is associated with USNs, but it can be applied to other use cases of the IoT.

3.1.18 real-world entity [ITU-T F.771]: A real-world entity is a physical and logical entity which mainly acts or is used in the real world, such as a physical object, logical object, place or person. Examples of *physical objects* include a water bottle, book, desk, wall, chair, tree, animal, cloth, food, television, light and so on. Examples of *logical objects* include digital content such as a video, movie, music or story. Examples of *places* include a room, corridor, road, gate, garden and so on. The real-world entity concept includes both networked entities and non-networked entities.

3.1.19 sensed data [ITU-T F.744]: Data sensed by a sensor that is attached to a specific sensor node.

3.1.20 sensor [ITU-T Y.2221]: An electronic device that senses a physical condition or chemical compound and delivers an electronic signal proportional to the observed characteristic.

3.1.21 sensor network [ITU-T Y.2221]: A network comprised of interconnected sensor nodes exchanging sensed data by wired or wireless communication.

3.1.22 sensor network common interface [ITU-T F.744]: An interface used between USN middleware and a sensor network/radio frequency identification (RFID) reader.

3.1.23 sensor network metadata [ITU-T F.744]: Information about a sensor network, such as a description of the sensor network, sensor node identifier, supported sensor type, the number of attached sensors for each sensor node, and the number of sensor nodes connected to the specific sensor network, etc.

3.1.24 sensor network metadata directory service [ITU-T F.744]: A directory service that provides sensor network metadata.

3.1.25 sensor node [ITU-T Y.2221]: A device consisting of sensor(s) and optional actuator(s) with capabilities of sensed data processing and networking.

3.1.26 smart grid [b-Smart-O-30Rev.6]: The "Smart Grid" is a two way electric power delivery network connected to an information and control network through sensors and control devices. This supports the intelligent and efficient optimization of the power network.

3.1.27 tag-based identification [ITU-T Y.2213]: The process of specifically identifying a physical or logical object from other physical or logical objects by using identifiers stored on an ID tag.

NOTE – The same term is also defined in [ITU-T F.771].

3.1.28 thing [ITU-T Y.2060]: In the Internet of things, this is an object of the physical world (physical things) or of the information world (virtual things), which is capable of being identified and integrated into communication networks.

3.1.29 ubiquitous networking [ITU-T Y.2002]: The ability for a person and/or device to access services and communicate while minimizing technical restrictions regarding where, when and how these services are accessed, in the context of the service(s) subscribed to.

NOTE (from [ITU-T Y.2002]) – Although technical restrictions to access services and communicate may be minimized, other constraints such as regulatory, national, provider and environmental constraints may impose further restrictions.

3.1.30 ubiquitous sensor network (USN) [ITU-T Y.2221]: A conceptual network built over existing physical networks which makes use of sensed data and provides knowledge services to anyone, anywhere and at any time, and where the information is generated by using context awareness.

3.1.31 web of things [ITU-T Y.2063]: A concept which refers to making use of the IoT in order for (physical and virtual) things to be connected and controlled via the world wide web.

NOTE (from ITU-T Y.2063) – This Recommendation intends using and accessing several kinds of physical devices on the web whether the devices are accessible on the web itself or not.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

IoT Internet of Things

M2M Machine to Machine

MOC Machine-Oriented Communication

USN Ubiquitous Sensor Network

Bibliography

[b-Smart-O-30 Rev.6] Smart-O-30 Rev.6 (2011), Focus Group on Smart Grid (FG Smart), *Deliverable on Smart Grid Terminology*. <http://www.itu.int/en/ITU-T/focusgroups/smart/Pages/Default.aspx>


Internet
of Thing
CONNEC
THE

3 Requirements and Use of Cases


IOT

Y.4100/Y.2066

Common requirements of the Internet of Things

Common requirements of the Internet of things

Summary

Recommendation ITU-T Y.2066 provides the common requirements of the Internet of things (IoT). These requirements are based on general use cases of the IoT and IoT actors, which are built from the definition of IoT contained in Recommendation ITU-T Y.2060. The common requirements of the IoT are independent of any specific application domain, which refer to the areas of knowledge or activity applied for one specific economic, commercial, social or administrative scope, such as transport application domain and health application domain.

This Recommendation builds on the overview of IoT (Recommendation ITU-T Y.2060), developing the common requirements based on general use cases of the IoT and the IoT actors and taking into account important areas of consideration from a requirement perspective. Some representative use cases of the IoT, which are abstracted from application domains, are also provided. The common requirements of the IoT specified in this Recommendation are classified into the categories of non-functional requirements, application support requirements, service requirements, communication requirements, device requirements, data management requirements and security and privacy protection requirements.

History

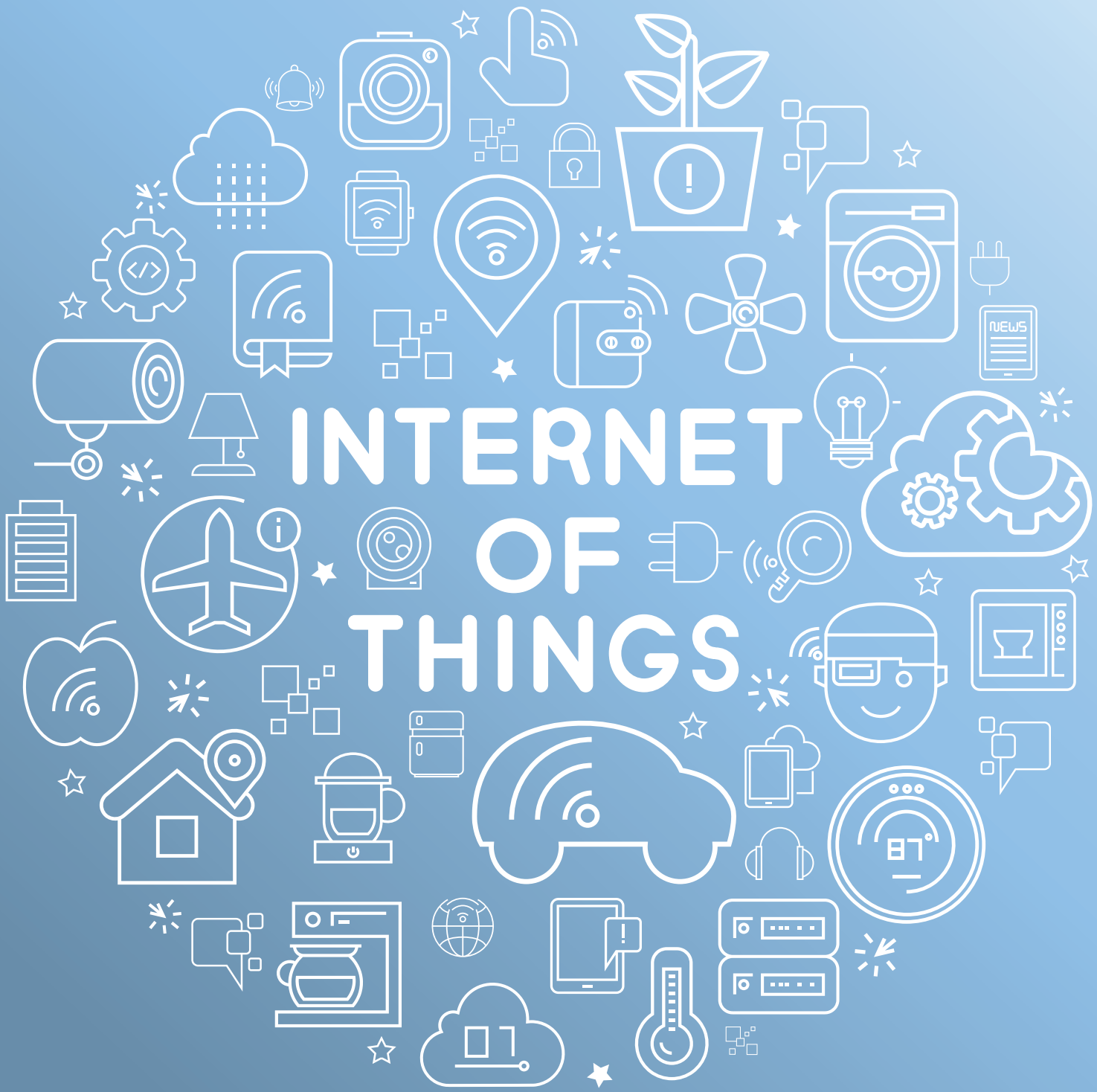
Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T Y.2066	2014-06-22	13	11.1002/1000/12169

Keywords

Common requirements, functional requirements, Internet of things (IoT), non-functional requirements, use cases.

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

Table of Contents


INTERNET
OF
THINGS
NEWS

Recommendation ITU-T Y.4100/Y.2066

Common requirements of the Internet of things

1 Scope

This Recommendation provides the common requirements of the Internet of things (IoT). These requirements are based on general use cases of the IoT and IoT actors, which are built from the definition of IoT contained in [ITU-T Y.2060]. The common requirements of the IoT are independent of any specific application domain, which refer to the areas of knowledge or activity applied for one specific economic, commercial, social or administrative scope, such as transport application domain and health application domain.

This Recommendation builds on the overview of IoT [ITU-T Y.2060], developing the common requirements based on general use cases of the IoT and IoT actors and taking into account important areas of consideration from a requirement perspective. The common requirements of the IoT specified in this Recommendation are classified into the categories of non-functional requirements, application support requirements, service requirements, communication requirements, device requirements, data management requirements and security and privacy protection requirements.

The scope of this Recommendation includes:

- general use cases of the IoT
- IoT actors
- important areas of consideration from a requirement perspective
- common requirements of the IoT.

The common requirements of the IoT are summarized and numbered in Annex A.

Some representative use cases of the IoT, which are abstracted from application domains, are provided in Appendix I.

NOTE – Regulatory, legal and business aspects are outside the scope of this Recommendation. Protocol and interface related requirements (e.g., for the control and management aspects of IoT) are also outside the scope of this Recommendation.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T Y.2060] Recommendation ITU-T Y.2060 (2012), *Overview of Internet of things*.

[ITU-T Y.2091] Recommendation ITU-T Y.2091 (2011), *Terms and definitions for next generation networks*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 application [ITU-T Y.2091]: A structured set of capabilities, which provide value-added functionality supported by one or more services, which may be supported by an API interface.

3.1.2 customer [ITU-T Y.2091]: The customer buys products and services from the enterprise or receives free offers or services. A customer may be a person or a business.

NOTE – There can be many users per customer.

3.1.3 device [ITU-T Y.2060]: With regard to the Internet of things, this is a piece of equipment with the mandatory capabilities of communication and optional capabilities of sensing, actuation, data capture, data storage and data processing.

3.1.4 Internet of things (IoT) [ITU-T Y.2060]: A global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving, interoperable information and communication technologies.

NOTE 1 – Through the exploitation of identification, data capture, processing and communication capabilities, the IoT makes full use of things to offer services to all kinds of applications, whilst ensuring that security and privacy requirements are fulfilled.

NOTE 2 – In a broad perspective, the IoT can be perceived as a vision with technological and societal implications.

3.1.5 service [ITU-T Y.2091]: A set of functions and facilities offered to a user by a provider.

3.1.6 thing [ITU-T Y.2060]: With regard to the Internet of things, this is an object of the physical world (physical things) or the information world (virtual things), which is capable of being identified and integrated into communication networks.

3.2 Terms defined in this Recommendation

This Recommendation defines the following term:

3.2.1 application domain: An area of knowledge or activity applied for one specific economic, commercial, social or administrative scope.

NOTE – Transport application domain, health application domain and government application domain are examples of application domains.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

2G	Second Generation
3G	Third Generation
API	Application Programming Interface
CAN	Controller Area Network
DSL	Digital Subscriber Line
IoT	Internet of Things
ITS	Intelligent Transport Systems
LTE	Long Term Evolution
M2M	Machine-to-Machine

MOC Machine Oriented Communication

SDP Service Delivery Platform

SLA Service Level Agreement

UML Unified Modelling Language

WiFi Wireless Fidelity

5 Conventions

In this Recommendation:

The keywords "**is required to**" indicate a requirement which must be strictly followed and from which no deviation is permitted if conformance to this document is to be claimed.

The keywords "**is recommended**" indicate a requirement which is recommended but which is not absolutely required. Thus this requirement need not be present to claim conformance.

The keywords "**can optionally**" and "**may**" indicate an optional requirement which is permissible, without implying any sense of being recommended. These terms are not intended to imply that the vendor's implementation must provide the option and the feature can be optionally enabled by the network operator/service provider. Rather, it means the vendor may optionally provide the feature and still claim conformance with the specification.

6 General use cases of the IoT and IoT actors

This clause describes general use cases of the IoT and IoT actors and the relations among general use cases and IoT actors. An IoT actor specified in this Recommendation refers to an entity that is external to the IoT and that interacts with the IoT.

6.1 General use cases

The general use cases are built from the definition of IoT contained in [ITU-T Y.2060].

According to the definition of IoT, given in [ITU-T Y.2060], IoT enables "advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies". This implies that the IoT interconnects things to sense or actuate things and to provide advanced services, so the general use cases of "IoT sensing or actuating" and "IoT service provision" can be derived.

According to the definition of IoT, as given in [ITU-T Y.2060], "Through the exploitation of identification, data capture, processing and communication capabilities, the IoT makes full use of things to offer services to all kinds of applications, whilst ensuring that security and privacy requirements are fulfilled". This implies that data capture and processing capabilities can be grouped as data management capabilities and privacy protection should be guaranteed. So the general use cases of "IoT data management" and "IoT privacy protection" can be derived.

Figure 6-1 shows the general use case model of the IoT, which is described via unified modelling language (UML), for more information see [b-UML]. This model consists of four general use cases: IoT sensing or actuating, IoT data management, IoT service provision and IoT privacy protection.

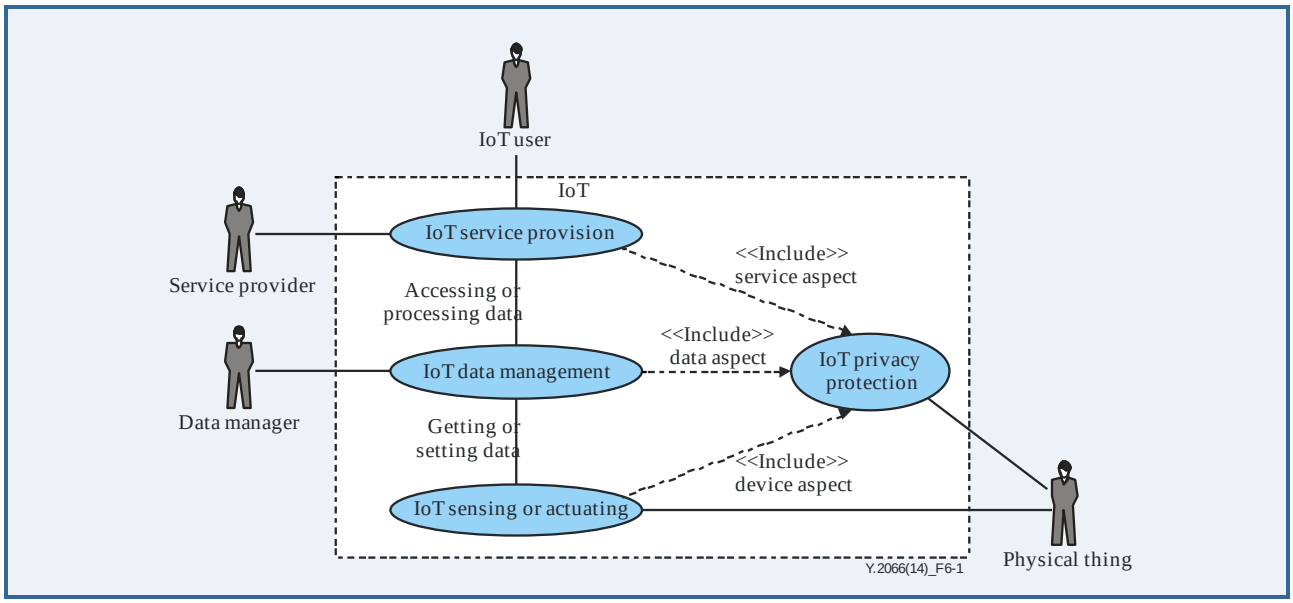


Figure 6-1 – The general use case model of the IoT

NOTE 1 – In [b-UML], a use case is defined as a single unit of meaningful work in a system. It may provide a view of behaviour observable by entities outside the system. The use cases can be used to capture the requirements of a system. A use case model (combination of single units of work) can show the interaction between the system and entities external to the system. These external entities are referred to as "actors" in UML. In this perspective, the IoT becomes the system being modelled with UML while an "IoT actor" is an entity that is outside the IoT, and interacts with the IoT.

NOTE 2 – Some use cases abstracted from IoT applications (the representative use cases described in Appendix I) can be decomposed into the general use cases described in clauses 6.1.1 to 6.1.4, to facilitate the generation of the functional requirements corresponding to the IoT actors. For example, the "video surveillance" use case described in Appendix I.1 can be decomposed into video capturing (IoT sensing or actuating), video transmission and storage (IoT data management) and video replay and analysis (IoT service provision) use cases. These use cases can be used to generate the functional requirements from different actors of video surveillance, such as time synchronization to support real-time video transmission and virtual storage to support storage of a large amount of video continuously generated by video cameras.

6.1.1 "IoT sensing or actuating" use case

"IoT sensing or actuating" use case is a general use case that can be applicable to multiple application domains. This use case involves the activities of connecting with physical things, sensing the states of physical things or actuating the physical things.

6.1.2 "IoT data management" use case

"IoT data management" use case is a general use case that can be applicable to multiple application domains. This use case involves the activities of capturing, transferring, storing and processing the data of physical things.

6.1.3 "IoT service provision" use case

"IoT service provision" use case is a general use case that can be applicable to multiple application domains. This use case involves the activities of providing services by the service provider and using services by the IoT user.

6.1.4 "IoT privacy protection" use case

"IoT privacy protection" use case is a general use case that can be applicable to multiple application domains. This use case involves the activities of securing and hiding the private information of the physical things.

6.1.5 The relationships among the general use cases

The relationships among the identified general use cases are shown in Figure 6-1. The "IoT data management" use case is related to both the "IoT sensing or actuating" use case and the "IoT service provision" use case. The "IoT privacy protection" use case is related to all other use cases.

6.2 The IoT actors

The use cases are used to capture the requirements of a system (see [b-UML]). Each use case includes the functional requirements of the actors involved in the use case.

According to the general use case model of IoT illustrated in Figure 6-1, there are four IoT actors: the "Physical thing" actor, the "Data manager" actor, the "Service provider" actor and the "IoT user" actor. These four IoT actors, described in this clause, are entities defined outside the IoT and specified from a requirement viewpoint. They are different from the business roles described in Appendix I of [ITU-T Y.2060], which are specified from a business viewpoint.

NOTE 1 – The "Physical thing" actor described in this Recommendation corresponds to physical thing as described in [ITU-T Y.2060]. According to the general use case model of IoT, the actor that would correspond to virtual thing as described in [ITU-T Y.2060] is not considered in this Recommendation since a virtual thing is an entity of the IoT itself.

NOTE 2 – The following provides the applicable mappings between the IoT actors described in this Recommendation and the roles described in Appendix I of [ITU-T Y.2060]:

- The "IoT user" actor corresponds to the application customer role.
- The "Service provider" actor corresponds to application provider, platform provider and network provider roles.
- The "Data manager" actor corresponds to the application provider role in the case where the provided applications involve some data management functionalities and it may also correspond to the device provider role in the case where the provided devices involve some data management functionalities.

6.2.1 "Physical thing" actor

The "Physical thing" actor is an IoT actor that has a unique identifier in the physical world. "Physical thing" interacts with the IoT via sensing or actuating activities.

NOTE – The "Physical thing" actor can be further instantiated into "artificial thing" and "natural thing". An artificial thing is a physical thing that is produced by mankind and can be identified by a product serial number. A natural thing is a physical thing that is generated in nature and can be identified, for example, by generated time, location and its category. Sensing natural things can constitute a challenge in the development of IoT.

Note that, in the following clauses of this Recommendation, the term "thing" refers to "physical thing".

6.2.2 "Data manager" actor

The "Data manager" actor is an IoT actor that is responsible for management of capturing, storing, transferring and processing IoT data to satisfy the IoT service provision requirements.

NOTE – The "Data manager" actor can be further instantiated into human "Data manager" and machine "Data manager" actors. A human "Data manager" actor performs the data management of IoT manually while a machine "Data manager" actor performs this in an automatic manner. These two instantiations of the "Data manager" actor are associated with different use cases of IoT data management.

6.2.3 "Service provider" actor

The "Service provider" actor is an IoT actor that provides all possible services related with things, such as monitoring, location tracking and service discovery.

NOTE – The "Service provider" actor can be further instantiated into a common "Service provider" actor that provides services which are independent of specific application domains and into an application "Service provider" actor that provides applications based on specific application domains.

6.2.4 "IoT user" actor

The "IoT user" actor is an IoT actor that uses all possible services related with things, such as monitoring, location tracking and service discovery.

7 Important areas for consideration from a requirement perspective

There are several important areas that need to be focused on for the specification of requirements of the IoT. Based on the IoT characteristics and high level requirements contained in [ITU-T Y.2060] as well as results of IoT related public and academic research (e.g., [b-IoT-A D6.2]), the following clauses describe important areas for consideration from a requirement perspective.

7.1 Implementation and operational aspects

The implementation and operational aspects of IoT are an important area to be addressed, e.g., in order to achieve interoperability among heterogeneous IoT implementations and to obtain the adequate scalability for the support of a large amount of connected devices and high availability for the support of automatic operations in IoT.

7.2 Ubiquitous connectivity

In order to realize connectivity between things and IoT, ubiquitous connectivity is required to be considered. Connectivity capabilities need to be independent of specific application domains and integration of heterogeneous communication technologies needs to be supported.

7.3 End-to-end intelligence

End-to-end intelligence is required to be considered, in particular with regard to the "intelligence of communications" and the "intelligence of services", e.g., in order to provide services without human intervention. This includes consideration of location based communications and context based communications (which may be regarded as intelligent communications), content-aware services and context-aware services (which may be regarded as intelligent services), as well as self-configuration, self-healing, self-optimization and self-protection services (which may be regarded as other intelligent services termed as a whole as autonomic services [ITU-T Y.2060]).

7.4 Time synchronization

In order to keep time synchronicity among the actions of interconnected things when using communication and service capabilities, time synchronization is required to be considered.

7.5 Human body connectivity

In order to provide communication capabilities related with the human body in compliance with regulation and laws, the requirements of human body connectivity are required to be carefully considered. Special quality of service (QoS) is required to be specified, reliability is required to be quantified, and privacy protection is required.

7.6 A large amount of data from things

As there will be a large number of devices connected with the IoT, there will be a large amount of data – the term "big data" is popularly used to signify large volume, variety and velocity of data – transmitted from things to the IoT. In order to classify, transfer, store, process, validate and query the big data within the time as required by IoT users or applications, resource scalability, such as communication bandwidth, storage and processing capacity, should be considered.

7.7 Privacy protection related with things

The data from things may contain private information related to the owners or users of things. The data may be used to locate or trace the owners or users of the things violating their privacy. Privacy protection during capturing, transferring, storing, validating and processing data of things should be considered. Privacy protection should not be used to hinder the validation of data of things.

8 Common requirements of the IoT

The common requirements of IoT specified in this Recommendation are technical requirements and are independent of any specific application domain. Protocol and interface related requirements (e.g., for the control and management aspects of IoT) are outside the scope of this Recommendation.

8.1 Categories of IoT common requirements

In this Recommendation, the IoT common requirements are divided into non-functional requirements and functional requirements.

The IoT non-functional requirements refer to the requirements related to the implementation and operation of the IoT itself.

The IoT functional requirements refer to the requirements related to the IoT actors, i.e., entities which are external to the IoT and that interact with the IoT. The IoT functional requirements specified in this Recommendation are categorized as follows:

– application support requirements
– service requirements
– communication requirements
– device requirements
– data management requirements
– security and privacy protection requirements.

All the requirements described in the following clauses are listed and numbered in Annex A. In the following clauses, the requirement numbers, as shown in Annex A, are put between square brackets "[ ]" and inserted at the end of each paragraph describing the corresponding requirement(s).

8.2 Non-functional requirements

The requirements of this category are not related to any IoT actors as they are not derived from the general use cases of IoT described in clause 6.

8.2.1 Interoperability

Interoperability is required to be ensured among heterogeneous IoT implementations [N1].

NOTE – In order to support interoperability in IoT, there is a need to standardize an architecture reference model of IoT.

8.2.2 Scalability

Scalability is required to be supported in IoT in order to handle a large number of devices, applications and users [N2].

NOTE 1 – The requirement in scalability for handling a large number of devices implies the requirement of handling a large amount of data (big data) in IoT.

NOTE 2 – The requirement in scalability for handling a large number of applications and users implies the requirement of having a large amount of processing and storage resources. Such a requirement may be supported via the integration of cloud computing technologies in IoT.

NOTE 3 – Fairness in handling a large number of devices, applications and users should be considered.

8.2.3 Reliability

Reliability in capabilities of IoT, such as reliability in communication, service and data management capabilities of IoT, is required [N3].

NOTE – Consideration should be given to resilience for support of reliability

8.2.4 High availability

High availability is required in service provisioning, data management, communication, sensing and actuating things of IoT [N4].

8.2.5 Adaptability

Adaptability to the new technologies emerging in the future is required in IoT [N5].

NOTE – The technical standards used in IoT should impose minimum constraints concerning the adaptability to new technologies.

8.2.6 Manageability

Manageability is required to be supported in IoT in order to ensure normal operations. IoT operations are usually performed automatically without people's intervention, but the operation process should be manageable [N6].

NOTE 1 – Consideration should be given to device management in IoT, e.g., device state management, device connectivity management, energy consumption management, etc. The constraints in device resources, such as energy, memory and bandwidth, should be considered in device management.

NOTE 2 – Consideration should be given to automatic fault management in IoT, e.g., proactive fault reporting, fault diagnosis, fault recovery, etc.

NOTE 3 – Consideration should be given to automatic configuration management in IoT, e.g., automatic configuration of device parameters.

8.3 Application support requirements

Application support requirements refer to the functional requirements from the development of IoT applications in different application domains. These requirements are only related to the "service provider" actor.

8.3.1 Programmable interfaces

Standardized programmable interfaces are required in order to provide open access to application support capabilities [A1].

NOTE – Programmable interfaces allow the support of IoT applications in a programmable way.

8.3.2 Group management

Group management, including display, creation, modification, deletion of IoT groups and display, addition, modification and deletion of IoT group members, is required to be supported in IoT [A2].

NOTE – An IoT group may contain IoT users and/or devices.

8.3.3 Time synchronization

Reliable time synchronization is required, in order to support global time stamping in IoT [A3].

NOTE – Time stamping allows the provision of secure and trusted time critical services.

8.3.4 Collaboration

Collaboration is required among services or among devices accessing, with the same goal, IoT applications, so that the IoT can enable autonomous goal-driven collaboration among such services or devices [A4].

NOTE – Collaboration among devices accessing IoT applications is expected to be activated by the devices themselves, so that the IoT can support scalable collaboration with distributed control among such devices.

8.3.5 User management

User management is required, including creation, authentication, authorization and accounting of IoT users [A5].

8.3.6 Resource usage accounting

Accounting of IoT resource usage is required on a per application basis [A6].

8.4 Service requirements

These requirements are related to the service provider, IoT user and thing actors.

NOTE – According to the general definition of "service" as a set of functions and facilities offered to a user by a provider [ITU-T Y.2091], the service requirements are related to both the IoT user and service provider actors. This does not exclude the case of a service offered directly to the thing actor.

8.4.1 Service prioritization

Prioritization of services is required to satisfy the different service requirements of different groups of IoT users [S1].

NOTE – Differentiated services are expected to be supported, so that the IoT can provide different service level agreements (SLAs).

8.4.2 Semantic based services

Semantic based services are required in IoT to support autonomic service provisioning. The mechanisms for implementing semantic based services include service semantic annotation, service semantic access and semantic exchange among services [S2].

NOTE – Service semantic annotation can allow the semantic description of services. Service semantic access can be used to access services through semantic interfaces. Semantic exchange among services can enable the provision and exchange of semantics between services in order to support automatic creation of new services.

8.4.3 Service composition

Service composition is required to support flexible service creation in IoT [S3].

NOTE 1 – The primary services are a set of basic operations that cannot directly satisfy some requirements of IoT applications. Service composition is one of the service creation methods that can be used to automatically create more complex services based on primary services in order to satisfy all of the various requirements of IoT applications.

NOTE 2 – Existing flexible service provisioning technologies, such as service delivery platform (SDP), can support, among others, the requirements of service composition.

8.4.4 Mobility services

Mobility services are required, so that the IoT can support service mobility, user mobility and device mobility in the service provisioning perspective, e.g., service provisioning is not constrained by the service access location when service mobility is supported [S4].

8.4.5 Reliable and secure human body connectivity services

High reliability and security are required when human body connectivity services are provided [S5].

NOTE – Different countries may have different legal and regulatory requirements on these services.

8.4.6 Autonomic services

Autonomic services are required, so that the IoT can enable automatic capture, communication and processing of data of things based on rules configured by service providers or customized by IoT users [S6].

NOTE – Support of both centralized and decentralized control of autonomic services is expected, so that the IoT can enable centralized or decentralized automated activities.

Location based and context-aware services are required, so that the IoT can enable flexible, user-customized and autonomic services based on the location information and related context of things and/or users. [S7].

8.4.7 Service management

Service management is required so that service provisioning can be supported in a highly available and reliable way [S8].

NOTE – Service management includes, among others, service lifecycle management and service integrity checking. Service lifecycle management can help to increase service availability and service integrity checking can help to increase service reliability.

8.4.8 Discovery services

Discovery services are required, so that the IoT users, services, devices and data of things can be discovered by service providers or IoT users [S9].

NOTE – The service provider or IoT user can discover specific IoT users, services, devices and data of things according to different criteria, such as geographic location information, type of device, etc.

8.4.9 Service subscription support

Service subscription support is required, so that the IoT can provide a means to allow the IoT user to subscribe to the needed services and associated data of things [S10].

8.4.10 Naming and addressing

Standardized naming and addressing of things and services is required [S11].

8.4.11 Virtual storage and processing

Virtual storage and processing capabilities are required in order to store and process a large amount of data (big data) [S12].

8.5 Communication requirements

Communication requirements refer to the functional requirements related to message exchange among the IoT user, service provider, data manager and thing actors. These requirements are related to all the IoT actors.

8.5.1 Communication modes

Event-based, periodic and automatic communication modes between devices or between IoT users are required to be supported [C1].

The support of the unicast communication mode is required (e.g., for communications between IoT users or devices). The support of the multicast, broadcast and anycast communication modes is required, so that the IoT can provide various communication services within a group of IoT users or devices (e.g., to support the collaboration among IoT users or devices) [C2].

NOTE – It is recommended to support event-based, periodic and automatic communication modes between devices or between IoT users, while preserving network performance by the support of mechanisms for avoiding the possibility of traffic congestion.

The support of device initiated communications is required so as to satisfy the requirements of automatic communications [C3].

8.5.2 Communication control

Error control for communications is required to be supported, so that the IoT is able, for example, to cope with interferences between devices [C4].

Time-critical communications are required to be supported, so that the IoT can provide time-critical message handling and delivery [C5].

8.5.3 Intelligent communication

The requirements of intelligent communication include requirements of autonomic networking [ITU-T Y.2060], content-aware communication and location based communication.

Autonomic networking is required in IoT to support self-configuring, self-healing, self-optimizing and self-protecting capabilities at the networking level, in order to adapt to different application domains, different communication environments and large numbers and varied types of devices [C6].

Content-aware communication is required, so that, for example, the IoT can provide a support for path selection and routing of communications based on content [C7].

Location based communication is required, so that the IoT can support location based interactions among IoT actors [C8].

NOTE – Location information is expected to be captured and traced automatically.

8.5.4 Heterogeneous communication support

Communications can take place in the device layer (see [ITU-T Y.2060]) through various kinds of wired or wireless technologies, such as controller area network (CAN) bus, ZigBee, Bluetooth, WiFi, etc. Support for heterogeneous device related communication technologies is required [C9].

Communications can take place in the network layer (see [ITU-T Y.2060]) through various kinds of technologies, such as second generation/third generation (2G/3G), long term evolution (LTE), Ethernet, digital subscriber line (DSL), etc.

Support for heterogeneous network related communication technologies is required [C10].

8.6 Device requirements

Device requirements refer to the functional requirements from the piece of equipment connected with things. These requirements are related to the IoT user and thing actors.

8.6.1 Connectivity of things

The IoT is required to support the establishment of the connectivity between a thing and the IoT based on the identifier of the thing [D1].

NOTE – Heterogeneous identifiers of things are expected to be processed in a unified way, (see [ITU-T Y.2060]).

8.6.2 Device control and configuration

Support of remote monitoring, control and configuration of devices is required so that device manageability in IoT is increased [D2].

Plug and play capability is required to be supported in IoT in order to enable on-the-fly generation, composition or acquisition of semantic-based configurations for seamless integration and cooperation of things with applications and responsiveness to application requirements, (see [ITU-T Y.2060]) [D3].

8.6.3 Monitoring of things

Automatic notification of the status of things and its changes is required in order to monitor things in timely manner [D4].

8.6.4 Device mobility

Device mobility is required, so that the IoT can support mobility of things connected with devices [D5].

8.6.5 Device integrity checking

Device integrity checking is required, in order help to support high availability of devices [D6].

8.7 Data management requirements

Data management requirements refer to the functional requirements from storing, aggregating, transferring and processing the data of the IoT. These requirements are related to the data manager and IoT user actors.

8.7.1 Data storage

Storing data of things based on predefined rules and policies is required to be supported [DM1].

8.7.2 Data processing

Data fusion and mining based on predefined rules and policies are required to be supported [DM2].

8.7.3 Data query

Querying stored historical data of things is required to be supported, so that the IoT can provide historical information about things [DM3].

8.7.4 Data access control

Access control of data by their owner is required to be supported in IoT, so that IoT users can have the ability to control how their data are exposed to other IoT users [DM4].

8.7.5 Data exchange

Data exchange with entities outside the IoT is required to be supported, so that the IoT is able to provide access to external data sources, e.g., health databases outside the IoT [DM5].

8.7.6 Data validation

Integrity checking and life cycle management of data of things are required to be supported, so that the IoT is able to provide high availability and reliability of data of things [DM6].

8.7.7 Semantic annotation and access to data of things

Semantic annotation of data of things is required. Semantic access to data of things is required, so that automatic querying of things can be supported [DM7].

8.7.8 Semantic storage, transfer and aggregation of data of things

Semantic storage, transfer and aggregation of data of things are required, so that storage, transfer and aggregation of data of things can be performed automatically according to the requirements of IoT users or applications [DM8].

8.8 Security and privacy protection requirements

Security and privacy protection requirements refer to the functional requirements during capturing, storing, transferring, aggregating and processing the data of things, as well as to the provision of services which involve things. These requirements are related to all the IoT actors.

8.8.1 Communication security

Secure, trusted and privacy protected communication capability is required, so that unauthorized access to the content of data can be prohibited, integrity of data can be guaranteed and privacy-related content of data can be protected during data transmission or transfer in IoT [SP1].

8.8.2 Data management security

Secure, trusted and privacy protected data management capability is required, so that unauthorized access to the content of data can be prohibited, integrity of data can be guaranteed and privacy-related content of data can be protected when storing or processing data in IoT [SP2].

8.8.3 Service provision security

Secure, trusted and privacy protected service provision capability is required, so that unauthorized access to service and fraudulent service provision can be prohibited and privacy information related to IoT users can be protected [SP3].

8.8.4 Integration of security policies and techniques

The ability to integrate different security policies and techniques is required, so as to ensure a consistent security control over the variety of devices and user networks in IoT [SP4].

8.8.5 Mutual authentication and authorization

Before a device (or an IoT user) can access the IoT, mutual authentication and authorization between the device (or the IoT user) and IoT is required to be performed according to predefined security policies [SP5].

8.8.6 Security audit

Security audit is required to be supported in IoT. Any data access or attempt to access IoT applications are required to be fully transparent, traceable and reproducible according to appropriate regulation and laws. In particular, IoT is required to support security audit for data transmission, storage, processing and application access [SP6].

Annex A

The IoT common requirements list

(This annex forms an integral part of this Recommendation.)

The following table lists and numbers the requirements described in clause 8 "Common requirements of the IoT".

Requirement number	Requirement category	Requirement description	Summary of the requirement	
N1	Non-functional	Interoperability is required to be ensured among heterogeneous IoT implementations.	Interoperability is required.	
N2	Non-functional	Scalability is required to be supported in IoT in order to handle a large amount of devices, applications and users.	Scalability is required.	
N3	Non-functional	Reliability in capabilities of IoT, such as reliability in communication, service and data management capabilities of IoT, is required	Reliability is required.	
N4	Non-functional	The IoT is required to provide high availability in service provisioning, data management, communication, sensing and actuating things.	High availability is required.	
N5	Non-functional	Adaptability to the new technologies emerging in the future is required in IoT	Adaptability is required.	
N6	Non-functional	Manageability is required to be supported in IoT in order to ensure normal operations.	Manageability is required.	
A1	Application support	Programmable interfaces are required to be standardized to provide open access to application support capabilities.	Standardized programmable interfaces are required.	
A2	Application Support	Group management, including display, creation, modification, deletion of IoT groups and display, addition, modification, deletion of IoT group members, is required to be supported in IoT.	Group management is required.	
A3	Application Support	In order to support global time stamping in IoT, reliable time synchronization is required.	Reliable time synchronization is required.	
A4	Application Support	Collaboration among services or among devices with the same goal accessing IoT applications is required.	Collaboration is required.	

Requirement number	Requirement category	Requirement description	Summary of the requirement	
A5	Application support	User management is required, including creation, authentication, authorization and accounting of IoT users.	User management is required.	
A6	Application support	Accounting of IoT resource usage is required on a per application basis.	Resource usage accounting is required.	
S1	Service	Prioritization of services is required to satisfy the different service requirements of different groups of IoT users.	Prioritization of services is required.	
S2	Service	Semantic based services are required in IoT to support autonomic service provisioning.	Semantic based services are required.	
S3	Service	Service composition is required to support flexible service creation in IoT.	Service composition is required.	
S4	Service	Mobility services are required, so that the IoT can support service mobility, user mobility and device mobility.	Mobility services are required.	
S5	Service	High reliability and security are required when human body connectivity services are provided.	Highly reliable and secure human body connectivity services are required.	
S6	Service	Autonomic services are required, so that the IoT can enable automatic capture, communication and processing of data of things based on rules configured by service providers or customized by IoT users.	Autonomic services are required.	
S7	Service	Location based and context-aware services are required, so that the IoT can enable flexible, user-customized and autonomic services based on the location information and related context of things and/or users.	Location based and context-aware services are required.	
S8	Service	Service management is required so that service provisioning can be supported in highly available and reliable way.	Service management is required.	
S9	Service	Discovery services are required, so that the IoT users, services, devices and data of things can be discovered by service providers or IoT users.	Discovery services are required.	

Requirement number	Requirement category	Requirement description	Summary of the requirement	
S10	Service	Service subscription support is required, so that the IoT can provide a means to allow the IoT user to subscribe to the needed services and associated data of things.	Service subscription support is required.	
S11	Service	Standardized naming and addressing of services and things is required.	Standardized naming and addressing is required.	
S12	Service	In order to store and process a large amount of data (big data), virtual storage and processing capabilities are required.	Virtual storage and processing capabilities are required.	
C1	Communication	The IoT is required to support event-based, periodic and automatic communications between devices or between IoT users.	Event-based, periodic, and automatic communication modes are required to be supported.	
C2	Communication	The support of the unicast communication mode is required (e.g., for communications between IoT users or devices). The support of the multicast, broadcast and anycast communication modes is required, so that the IoT can provide various communication services within a group of IoT users or devices (e.g., to support the collaboration among IoT users or devices).	The support of the unicast, multicast, broadcast and anycast communication modes is required.	
C3	Communication	The support of device initiated communications is required so as to satisfy the requirements of automatic communications.	The support of device initiated communications is required.	
C4	Communication	Error control for communications is required, so that the IoT is able, for example, to cope with interferences between devices.	Error control for communications is required to be supported.	
C5	Communication	The IoT is required to provide time-critical message handling and delivery.	Time-critical communications are required to be supported.	
C6	Communication	Self-configuring, self-healing, self-optimizing and self-protecting capabilities at the networking level are required in IoT.	Autonomic networking is required.	

Requirement number	Requirement category	Requirement description	Summary of the requirement	
C7	Communication	Content-aware communication is required, so that, for example, the IoT can provide a support for path selection / and routing of communications based on content.	Content-aware communication is required.	
C8	Communication	The IoT is required to support location based interactions among IoT actors.	Location based communication is required.	
C9	Communication	Communications can take place in the device layer [ITU-T Y.2060] through various kinds of wired or wireless technologies, such as controller area network (CAN) bus, ZigBee, Bluetooth, WiFi, etc.	Support for heterogeneous device related communication technologies is required.	
C10	Communication	Communications can take place in the network layer [ITU-T Y.2060] through various kinds of technologies, such as second generation /third generation (2G/3G), long term evolution (LTE), Ethernet, digital subscriber line (DSL), etc.	Support for heterogeneous network related communication technologies is required.	
D1	Device	The IoT is required to support the establishment of the connectivity between a thing and the IoT based on the identifier of the thing.	Identification-based connectivity between a thing and the IoT is required.	
D2	Device	Support of remote monitoring, control and configuration of devices is required so that device manageability in IoT is increased.	Remote monitoring, control and configuration of devices are required.	
D3	Device	Plug and play capability is required to be supported in IoT in order to enable on-the-fly semantic-based configurations of devices.	Plug and play capability is required.	
D4	Device	Automatic notification of the status of things and its changes is required in order to monitor things in a timely manner.	Monitoring things in a timely manner is required.	
D5	Device	The IoT is required to support mobility of things.	Device mobility is required.	
D6	Device	Device integrity checking is required, in order to support high availability of devices.	Device integrity checking is required.	

Requirement number	Requirement category	Requirement description	Summary of the requirement	
DM1	Data management	The IoT is required to support storing data of things based on predefined rules and policies.	Storing data of things is required to be supported.	
DM2	Data management	Data fusion and mining based on predefined rules and policies are required to be supported.	Processing data of things is required to be supported.	
DM3	Data management	The IoT is required to provide historical information about things	Querying historical data of things is required to be supported.	
DM4	Data management	Access control of data by their owner is required to be supported in IoT, so that IoT users can have the ability to control how their data are exposed to other IoT users	Data access control by the data owners is required.	
DM5	Data management	The IoT is required to provide access to external data sources, e.g., health databases outside the IoT.	Data exchange with entities outside the IoT is required to be supported.	
DM6	Data management	The IoT is required to provide integrity checking and life cycle management of data of things, so that the IoT is able to provide high availability and reliability of data of things.	Integrity checking and life cycle management of data of things is required.	
DM7	Data management	Semantic annotation of data of things is required. Semantic access to data of things is required, so that automatic querying of things can be supported.	Semantic annotation and semantic access to data of things are required.	
DM8	Data management	Storage, transfer and aggregation of data of things are required to be performed automatically according to the requirements of IoT users or applications.	Semantic storage, transfer and aggregation of data of things are required.	
SP1	Security and privacy protection	The IoT is required to support secure, trusted and privacy protected communication capability.	Communication security is required.	
SP2	Security and privacy protection	The IoT is required to provide secure, trusted and privacy protected data management capability.	Data management security is required.	
SP3	Security and privacy protection	The IoT is required to provide secure, trusted and privacy protected service provision capability.	Service provision security is required.	

Requirement number	Requirement category	Requirement description	Summary of the requirement	
SP4	Security and privacy protection	Integration of different security policies and techniques related to the variety of devices and user networks in IoT is required.	Integration of different security policies and techniques is required.	
SP5	Security and privacy protection	Before a device (or an IoT user) can access the IoT, mutual authentication and authorization is required according to predefined security policies.	Mutual authentication and authorization is required.	
SP6	Security and privacy protection	Any data access or attempt to access IoT applications are required to be fully transparent, traceable and reproducible according to appropriate regulation and laws.	Security audit is required to be supported in IoT.	

Appendix I

Representative use cases of the IoT

(This appendix does not form an integral part of this Recommendation.)

This appendix describes some representative use cases of the IoT, which are abstracted and classified based on application use cases within application domains or across multiple application domains.

I.1 Video surveillance

Video surveillance is a typical class of use cases present in numerous IoT applications. For example, in smart city applications, video cameras are used to watch people's movements for city safety purposes. In pollution supervision, video surveillance is used to watch whether polluted water flows out of a factory. Hospitals use video surveillance to watch the status of a patient remotely.

Video surveillance typically requires a large number of resources, such as high communication bandwidth for transferring video, a large amount of storage resources for keeping copies of video and powerful processors for searching and processing video.

I.2 Emergency alerting

Emergency alerting is abstracted from a large number of use cases, such as rescue message transmission when a patient heart disease occurs, alerting message transmission before a vehicle fails to work normally or after that, when a traffic accident happens, and alerting message transmission when blood pressure exceeds a threshold value.

Such use cases require high priority and reliable data transport with minimized time delay and also require device-initiated communication capabilities.

I.3 Data acquisition

This class of use cases includes a number of use cases, such as gas metering, water metering and quality monitoring, electricity metering, bus ticket terminal data uploading, etc. In these use cases, data communications happen at regular time intervals.

These use cases require mechanisms for periodical data transmission. The transmission task may be activated automatically under a given policy. Normally, in most of these use cases, the throughput of data transmission is low.

I.4 Remote control

This class of use cases includes use cases within a number of application domains, such as home automation, manufacturing and intelligent transport systems (ITS). In these use cases, the IoT application requires the capability for the user to control devices remotely.

For this class of use cases, data communications for controlling remote devices are not continuous and do not necessarily happen at regular time intervals. These use cases require mechanisms to establish connectivity between controllers and remote devices initiated by controllers or devices only when data transmission is required.

I.5 Transfer of events across different application domains

In many IoT applications such as smart city and emergency management applications, events that happen in one application domain are transferred to other relevant application domains. Based on events transferred across different application domains, different applications can work collaboratively so that more functions and services can be provided than those specific to a single application domain. Examples of such use cases include events transferred between road and bridge maintenance applications, between traffic management and driving applications, between weather forecast and flood prevention applications, etc.

Such use cases need that the events be described in a standardized format so the different IoT applications can understand them. Furthermore, the events should be transferred reliably and securely.

I.6 Data sharing across different application domains

Some data are of importance not only in the IoT application where such data are collected, but also in other IoT applications. Such data includes geographic position data, road traffic data, etc. In accordance with appropriate regulation and laws, data may also be shared across different application domains thus allowing for more functions and services to be provided. For example, data related to the geographic position of mobile phones might be used for calculating the road traffic.

Such use cases need standardized data formats among the different IoT application domains so that data can be shared across different application domains.

I.7 Integrated operating centre for smart city

Smart cities developed based on IoT infrastructure are becoming a new trend in city development all over the world. In the future, cities will need to have an intelligent "brain" system to analyse different kinds of data collected by IoT devices and to act upon their analysis and other related actions. Such a city brain system may be referred to as an "integrated operating centre for a smart city".

This integrated operating centre basically requires data sharing, aggregation and processing across multiple application domains. For example, implementations of such an integrated operating centre usually require the integration of urban real-time operational status information with event monitoring, data analysis, intelligent early warning and information dissemination, intelligent decision-making and integrated command and dispatching.

I.8 One detailed use case: traffic accident information collection

An ITS-station (ITS-S) inside a vehicle that is directly involved in or is passing by an accident detects that a crash has happened and starts an accident reporting process automatically. It tries to connect to the IoT and then sends the accident report to it. The IoT receives and verifies the accident report and the analysis result is pushed to the service subscribers, i.e., a Police Station and a Rescue Centre.

The service subscribers can ask the IoT to collect more information about the accident. The IoT receives these service requests and then asks the ITS-Ss to collect some more information according to the requests of the subscribers. The ITS-Ss in proximity to the accident site receive, verify, parse and execute the received commands, i.e., take pictures, get current travel status, generate reports, sign the reports and upload signed reports to the IoT. The IoT accumulates and verifies the reports uploaded by the ITS-Ss and then generates a report containing visual information about the accident scene for the Rescue Centre and a report about the traffic situation near the accident site. These reports are again pushed to the Rescue Centre and Police Station respectively.

The rescue centre analyses the report about the accident scene and then formulates a specific rescue plan. The police station analyses the report about traffic situation and formulates a specific traffic control plan.

This use case requires device-initiated communication capability, secure and trust communication capability and event-driven collaboration among different applications.

Bibliography

[b-ITU-T Y.2061] Recommendation ITU-T Y.2061 (2012), *Requirements for the support of machine-oriented communication applications in the next generation network environment.*

[b-IoT-A D6.2] The Internet of Things Architecture – IoT-A (2011), *Project Deliverable D6.2 – Updated Requirements List*.
http://www.iot-a.eu/public/public-documents/documents-1

[b-UML] ISO/IEC 19505-2:2012, *Information technology – Object Management Group Unified Modeling Language (OMG UML) – Part 2: Superstructure*.
http://www.iso.org/iso/catalogue_detail.htm?csnumber=52854

Y.4101/Y.2067

Common requirements and capabilities of a gateway for Internet of Things applications

Common requirements and capabilities of a gateway for Internet of things applications

Summary

Recommendation ITU-T Y.2067 provides the common requirements and capabilities of a gateway for Internet of things (IoT) applications. The provided common requirements and capabilities are intended to be generally applicable in gateway application scenarios.

NOTE – This Recommendation focuses on the gateway as equipment interconnecting devices with communication networks.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T Y.2067	2014-06-06	13	11.1002/1000/12170

Keywords

Capabilities, common requirements, gateway, IoT, IoT applications.

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

Table of Contents

Recommendation ITU-T Y.4101/Y.2067

Common requirements and capabilities of a gateway for Internet of things applications

1 Scope

This Recommendation provides the common requirements and capabilities of a gateway for Internet of things (IoT) applications. The provided common requirements and capabilities are intended to be generally applicable in gateway application scenarios.

The scope of this Recommendation includes:

- General characteristics of a gateway for IoT applications
- Common requirements of a gateway for IoT applications
- Common capabilities of a gateway for IoT applications

Use cases of a gateway for IoT applications are provided in appendixes.

NOTE – This Recommendation focuses on the gateway as equipment interconnecting devices with communication networks.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T Y.2060] Recommendation ITU-T Y.2060 (2012), *Overview of Internet of things.*

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 device [ITU-T Y.2060]: With regard to the Internet of things, this is a piece of equipment with the mandatory capabilities of communication and the optional capabilities of sensing, actuation, data capture, data storage and data processing.

3.1.2 Internet of things (IoT) [ITU-T Y.2060]: A global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies.

NOTE 1 – Through the exploitation of identification, data capture, processing and communication capabilities, the IoT makes full use of things to offer services to all kinds of applications, whilst ensuring that security and privacy requirements are fulfilled.

NOTE 2 – From a broader perspective, the IoT can be perceived as a vision with technological and societal implications.

3.2 Terms defined in this Recommendation

This Recommendation defines or uses the following term:

3.2.1 gateway: A unit in the Internet of things which interconnects the devices with the communication networks. It performs the necessary translation between the protocols used in the communication networks and those used by devices.

4 Abbreviations and acronyms

This Recommendation defines or uses the following terms:

3G	Third Generation
4G	Fourth Generation
CAN	Controller Area Network
CRM	Customer Relationship Management
ECU	Electronic Control Unit
GPRS	General Packet Radio Service
GPS	Global Positioning System
IoT	Internet of Things
IP	Internet Protocol
LTE	Long Term Evolution
MAC	Media Access Control
MSISDN	Mobile Subscriber International ISDN/PSTN number
NGN	Next Generation Network
PHY	Physical layer
QoS	Quality of Service
SMS	Short Message Service
TCP	Transmission Control Protocol
URI	Uniform Resource Identifier
WCDMA	Wideband Code Division Multiple Access
Wi-Fi	Wireless Fidelity
xPON	x Passive Optical Network

5 Conventions

In this Recommendation:

The keywords "**is required to**" indicate a requirement which must be strictly followed and from which no deviation is permitted if conformance to this document is to be claimed.

The keywords "**is recommended**" indicate a requirement which is recommended but which is not absolutely required. Thus this requirement need not be present to claim conformance.

The keywords "**can optionally**" and "**may**" indicate an optional requirement which is permissible, without implying any sense of being recommended. These terms are not intended to imply that the vendor's implementation must provide the option and the feature can be optionally enabled by the

network operator/service provider. Rather, it means the vendor may optionally provide the feature and still claim conformance with the specification.

6 Introduction to gateways for IoT applications

In IoT applications, information in either the physical or information world is collected by devices and then sent to the IoT applications through communication networks. Some devices cannot connect to the communication networks directly. The gateways support the interconnection of such devices with the communication networks.

Figure 1 shows the typical deployment scenario of gateways for IoT applications.

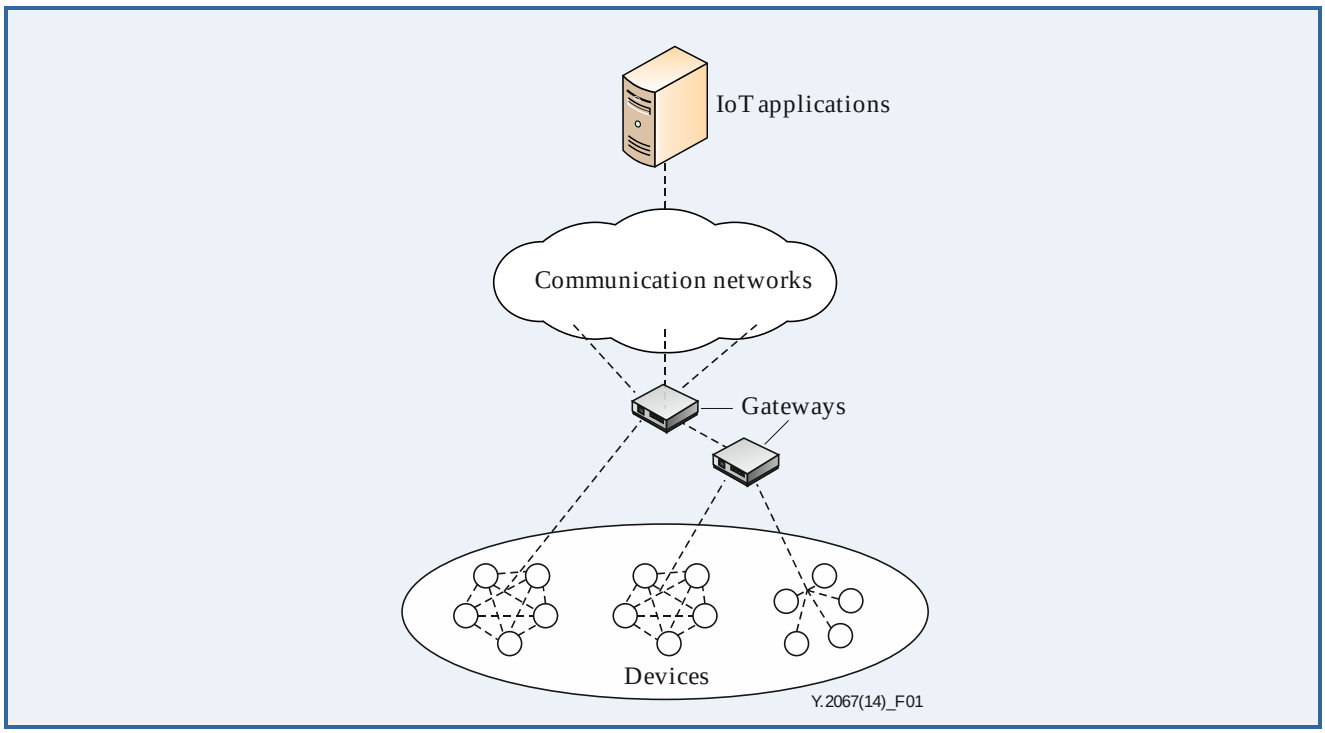


Figure 1 – Typical deployment scenario of gateways for IoT applications

As shown in Figure 1, different kinds of devices can connect to the communication networks through one or multiple gateways. The connectivity between devices and gateway(s) can be based on different kinds of wired or wireless technologies, such as a controller area network (CAN) bus, ZigBee, Bluetooth or Wi-Fi.

The communication networks can be realized via existing networks, such as conventional TCP/IP-based networks and/or evolving networks, such as next generation networks (NGN) [b-ITU-T Y.2001]. A gateway that connects to these networks should support the appropriate communication technologies.

The IoT applications implement application logic according to the application requirements. The applications can be based on proprietary application platforms, but can also be built upon common service/application support platforms providing generic enabling capabilities, such as authentication, device management, charging and accounting [ITU-T Y.2060].

The gateway connects to the IoT applications through the communication networks.

7 General characteristics of a gateway for IoT applications

7.1 Connection to the communication networks

The gateway has the general characteristic of connecting to the communication networks. Devices can connect to the communication networks through such a gateway. In some cases, for example in configurations with multiple gateways, one or more gateways are connected to other gateways (as shown in Figure 1) and not directly to the communication networks.

The gateway supports different kinds of communication technologies to connect to different communication networks.

7.2 Device access

The gateway has the general characteristic of supporting the access of devices. The devices can connect to each other or to the communication networks by accessing gateways. The gateway supports different kinds of device access technologies.

7.3 Protocol translation

The gateway has the general characteristic of protocol translation. The gateway supports the protocol translation between the devices and the communication networks. In some cases, a gateway translates the protocols among different devices which are connected to the gateway itself.

7.4 Interaction with applications

The gateway has the general characteristic to support the interaction with applications, including common application logic interaction.

7.5 Adaptability

The gateway has the general characteristic of adaptability. It is expected that the gateway has standardized interfaces. The gateway can be deployed in different application environments by adapting according to functional components and related protocols.

7.6 Management functions support

The gateway has the general characteristic to support management functions, including device management, network management and protocol management.

7.7 Security functions support

The gateway has the general characteristic to support security functions. The gateway provides security mechanisms to support the security requirements of applications.

NOTE – Common security mechanisms used in a gateway include those for device authentication, data encryption, privacy protection, etc.

8 Common requirements and recommendations of a gateway for IoT applications

8.1 General gateway requirements and recommendations

– **Scalability**

  There may be a huge number of devices accessing a gateway. The gateway is required to be scalable in terms of the number of connected devices and to support interconnection with other gateways to increase the global scalability of the gateways.

– **Addressing**

The gateway is required to support various addressing schemes, e.g., IP and non-IP addressing schemes, including public and private addressing for IP schemes.

– **Openness to functional extensions**

The gateway is required to provide standard interfaces to support functional extensions of the gateway, e.g., for deployment in diversified application environments.

– **Quality of service**

The gateway usually plays a key role in the IoT application scenarios where quality of service (QoS) support is essential.

The QoS related requirements of the gateway are as follows:

1) The gateway is required to support traffic control policy and QoS differentiation according to the categories of traffic.
2) The gateway is required to provide mechanisms for performance measurement and management.

– **Communication aspects**

The gateway is deployed between devices and communication networks and can use different communication technologies (e.g., 3G, 4G, xPON, ZigBee, Wi-Fi and Ethernet) to transfer data.

The communication related requirements of the gateway are as follows:

1) The gateway is required to support communication bridging between devices and communication networks.
2) The gateway is required to support communications with at least one application.
3) The gateway is recommended to support multiple communication technologies to interact with communication networks and devices and be able to enhance the capabilities of the communication interfaces in case that the support of additional communication technologies is required. In such case, the gateway is required to be able to select the communication technologies according to the specific service requirements.

8.2 Adaptation related requirements and recommendations

– **Protocol diversity support**

The gateway needs to communicate with devices and applications that may support different protocols. The gateway should be able to load new protocols according to the communication requirements.

The protocol related requirements of the gateway are as follows:

1) The gateway is required to support protocol translation between different protocols as necessary when communicating with devices and applications.
2) The gateway is recommended to support dynamic protocol loading.

– **Uniformity of interactions**

The gateway is recommended to support uniform interaction with different devices and applications in order to cope with their heterogeneity.

The requirements of the gateway related to uniformity of interactions are as follows:

1) The gateway is recommended to support uniform operations through standardized protocols on devices which use different communication technologies.

2) The gateway is recommended to support uniform interaction through standardized protocols with different applications.

8.3 Support capabilities related requirements and recommendations

– **Device and service discovery**

When devices are connected to a gateway, the gateway discovers them. In addition, the gateway discovers new services which are published by applications.

The device and service discovery requirements of the gateway are as follows:

1) The gateway is required to support mechanisms for device discovery when a device connects to the gateway for the first time or in the case of gateway restart.

2) The gateway is required to support mechanisms for service discovery when new services are published by applications.

– **Device management**

There are a great number of devices that are connected to a gateway and most of them have capability constraints. The gateway manages devices based on policies or instructions received from applications.

The device management requirements of the gateway are as follows:

1) The gateway is required to support management of device related information, e.g., device identification, device configuration, etc.

2) The gateway is required to support monitoring of device status for usage by applications or itself.

3) The gateway is required to support firmware and software update of devices.

4) The gateway is required to support device management on behalf of applications upon request.

5) The gateway is recommended to support fault management of devices based on policies.

6) The gateway is recommended to support performance management of devices based on policies.

– **Device identifier management**

Multiple types of device identifiers may be used in IoT applications, e.g., IP address, MSISDN, URI, data elements, etc. A device may have single or multiple identifiers which are managed by the gateway.

The device identifier requirements of the gateway are as follows:

1) The gateway is required to support identifier mapping capability between different types of device identifiers.

2) The gateway is recommended to support identifier combination capability, e.g., the combination of device identifier with gateway identifier.

NOTE – The combined identifiers may be provided to applications as globally unique identifiers, while the gateway resolves the combined identifiers in order to address the different devices.

3) The gateway is recommended to support the assignment of temporary communication identifiers to the devices connected to the gateway itself.

– **Storage**

A gateway has two methods to store data. The first is temporary storage, in this case the data which are temporarily stored need to be removed according to pre-defined policies, e.g., service logic, maximum data storage volume. The second data storage method is permanent storage, in this case the data which are permanently stored are important for successful service operations and for correct gateway and device operations.

For data safety and security, the data stored in gateways and applications should be kept consistent.

The storage requirements and recommendations of the gateway are as follows:

1) The gateway is required to support local storage, including temporary and permanent storage.
2) The gateway is recommended to support capabilities for ensuring data consistency between the gateway and applications.

NOTE – Applications are expected to support capabilities for ensuring data consistency with gateways.

– **Device grouping**

Devices may be grouped by type, location, etc. For example, all devices in the same room can constitute a group. Likewise the devices of the same type behind a gateway can constitute a group. A gateway can operate devices efficiently based on groups. The gateway is required to support group operations for devices, including operations to create, update, read and delete groups of devices.

– **Data capture and aggregation**

A gateway captures data from devices and transfers the data to applications. A gateway may have multiple modes of capturing and aggregating data based on policies.

The data capture and aggregation requirements of the gateway are as follows:

1) The gateway is required to support data capture from devices based on policies, e.g., real time collection or time schedule-based collection.
2) The gateway is recommended to support aggregation of data from devices.

– **Data dispatching and delivery**

For a large number of devices behind a gateway, the gateway can efficiently dispatch and transfer data between devices and applications based on policies.

The data dispatching and delivery requirements and recommendations of the gateway are as follows:

1) The gateway is required to support mechanisms to dispatch data based on policies.
2) The gateway is recommended to support mechanisms to pre-process data based on policies before dispatching them.
3) The gateway is required to support data delivery based on QoS requirements of applications.
4) The gateway is required to support data delivery based on devices' group identification if devices are grouped.

8.4 Application related requirements

– **Application logic integration**

The gateway is recommended to support application logic integration.

NOTE – By supporting application logic integration, the gateway can process application related functions locally and independently from remote facilities. For example, in some cases, the gateway can perform some processing and analysis of the data captured from the connected devices before transferring the data to applications.

8.5 Security and management related requirements

– **Security and privacy**

For the security of applications, a gateway must control the access to devices and to itself and must protect data security and privacy for the gateway and devices.

The security and privacy requirements of the gateway are as follows:

1) The gateway is required to support identification of each access to the connected devices.
2) The gateway is required to support authentication with devices. Based on application requirements and device capabilities, it is required to support mutual or one-way authentication with devices.
3) The gateway is required to support mutual authentication with applications.
4) The gateway is required to support the security of the data which are stored in devices and the gateway, or transferred between the gateway and devices, or transferred between the gateway and applications. The gateway is required to support the security of these data based on security levels.
5) The gateway is required to support mechanisms to protect privacy for devices and the gateway.

– **Self-management and remote maintenance**

The gateway is required to support self-management and remote maintenance.

The self-management and remote maintenance requirements of the gateway are as follows:

1) The gateway is required to support self-diagnosis and self-repair as well as remote maintenance.
2) The gateway is required to support firmware and software update.
3) The gateway is required to support auto configuration or configuration by applications. The gateway is required to support multiple configuration modes, e.g., remote and local configuration, automatic and manual configuration and dynamic configuration based on policies.

9 Common capabilities of a gateway for IoT applications

9.1 Reference technical framework and typical high-level flows of a gateway for IoT applications

9.1.1 Reference technical framework

The reference technical framework of a gateway for IoT applications is composed of the following capability groups:

- Applications group
- Support capabilities group
- Adaptation capabilities group
- Security and management capabilities group

The applications group provides support for interacting with remote applications and for local processing of application logic. It supports the deployment of multiple IoT applications of different

kinds and used in different domains (e.g., power metering in smart home domain, elder people monitoring in e-health domain, etc.). This group may utilize the capabilities provided by the support capabilities group.

The support capabilities group provides common capabilities for the gateway to interact with devices and applications. This group includes the following capabilities:

– Device management, which provides capabilities for managing devices and communicates device profiles to the gateway itself and to applications.

– Communication management, which provides capabilities for establishing and managing communication with devices and applications. It includes capabilities for the support of the communication QoS requirements (e.g., communication delay, packet loss, etc.).

– Data storage, which provides capabilities for permanent and temporary storage of data, including data collected from devices, gateway configuration data, data from applications, etc.

– Data processing, which provides capabilities for processing data, including analyzing data, transforming data formats, encapsulating data based on application protocols and aggregating data from devices.

– Data dispatching, which provides capabilities for pre-processing data from applications based on policies and optimizing data dispatching.

The adaptation capabilities group provides capabilities for communicating with devices and applications and hiding the differences between devices and applications. This group includes the following capabilities:

– Interface abstraction, which provides an abstract interface supporting basic operations (such as reading data from a device) to interact with devices and applications and also provides mapping capability from an abstract interface to specific interfaces supported by devices and applications.

– Device adaptation, which provides connectivity adaptation for the different types of devices or other gateways that connect to the gateway.

– Network adaptation, which provides adaptation to different network technologies, including PHY/MAC layer adaptation between the gateway and the (access portion of the) communication networks.

The security and management capabilities group provides capabilites for supporting security and management of the gateway itself.

Figure 2 shows the reference technical framework of a gateway for IoT applications.

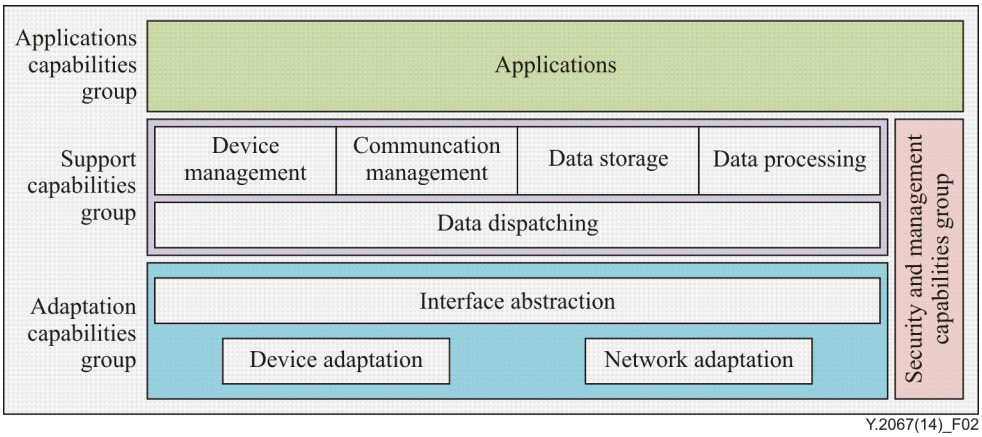


Figure 2 – Reference technical framework of a gateway for IoT applications

9.1.2 Typical high-level flows

In IoT applications, a gateway can receive data from IoT applications and then send the data to devices and it can receive data from devices and then send the data to IoT applications. In this regard, the typical high-level flows with respect to the capability groups identified in the gateway reference technical framework are as follows:

– Data are received from IoT applications and sent to devices: the gateway receives data from IoT applications through the adaptation capabilities group which provides network adaptation and interface abstraction. The gateway makes the necessary application logic process ing via the applications capabilities group, and sends data to devices through the adaptation capabilities group which provides interface abstraction and device adaptation. These processes are accomplished in collaboration with the support capabilities group and the security and management capabilities group.

– Data are received from devices and sent toIoT applications: the gateway receives data from devices through the adaptation capabilities group which provides device adaptation and interface abstraction. The gateway makes the necessary application logic processing via the applications capabilities group, and sends data to IoT applications through the adaptation capabilities group which provides interface abstraction and network adaptation. These processes are accomplished in collaboration with the support capabilities group and the security and management capabilities group.

9.2 Details on common capabilities of a gateway for IoT applications

9.2.1 Applications group

The functionalities of the applications group are as follows:

– Support deployment of specific IoT application logic in the gateway via standard open interface. Via such application logic, the gateway can process some IoT application related functions locally.

– Support resource openness with proper access control, so that the resources of the gateway which are usable for the creation of new IoT applications, can be discovered and accessed. The gateway is required to support functions for resource openness, including resource abstraction, resource identifier management, resource registration and deregistration etc.

9.2.2 Support capabilities group

9.2.2.1 Data dispatching

The functionalities of data dispatching are as follows:

– Support capability of dispatching data to devices according to the sequential order of the devices' data.

– Support capability of dispatching data from devices to applications as appropriate.

– Support capability of adjusting the sequential order of the devices' data based on policies.

9.2.2.2 Device management

The functionalities of device management are as follows:

– Support capability of providing collection and monitoring of device status.

– Support capability of providing device related information to applications.

– Support capability for device firmware and software update.

– Support device configuration, according to configuration profiles (downloaded from applications, or stored in the gateway) or configuration commands (received from applications).

– Support device diagnosis and automatic reparation.
– Support capability of creating, updating, deleting and retrieving device identifiers and managing identifier mapping.
– Support device discovery.
– Support capability of grouping devices based on device attributes (such as device type, device location, etc.).

9.2.2.3 Data processing

The functionalities of data processing are as follows:

– Support capability of data format transformation between different data formats as required by devices and applications.
– Support capability of aggregating data from devices and applications.

9.2.2.4 Data storage

The functionalities of data storage are as follows:

– Support access rights (e.g., read, write) to data that are stored in the gateway for security and privacy purposes.
– Support capability of data caching for data from devices and applications.
– Support data synchronization between the gateway and applications, e.g., upload of collected data from devices to applications, download of configuration management data from applications to the gateway.

9.2.2.5 Communication management

The functionalities of communication management are as follows:

– Support capability of establishing and managing communications between the gateway and applications.
– Support selection of the access network (to connect with the communication networks) according to the communication technologies supported by the gateway (e.g., GPRS, WCDMA, LTE, etc).
– Support capability of data transferring from applications and devices based on QoS enabled policies, e.g., priority of data transferring from devices in different network environments.
– Support capability of communication based on device grouping.

9.2.3 Adaptation capabilities group

9.2.3.1 Interface abstraction

The functionalities of interface abstraction are as follows:

– Support interface mapping from abstract interface to specific interfaces supported by devices and applications. This includes interface mapping for new device interfaces when new types of devices connect to the gateway.

9.2.3.2 Device adaptation

The functionalities of device adaptation are as follows:

– Support capability of connectivity adaptation for the different types of devices or other gateways that connect to the gateway.

9.2.3.3 Network adaptation

The functionalities of network adaptation are as follows:

– Support capability for connecting to various types of communication networks according to the appropriate communication technologies, including for PHY/MAC layer adaptation between the gateway and the access portion of the communication networks.
– Support capability for dynamic loading of communication protocols.

9.2.4 Security and management capabilities group

The functionalities of security and management capabilities group are as follows:

– Support mutual authentication between the gateway and applications.
– Support mutual or one-way authentication between the gateway and devices.
– Support security policies according to different security levels.
– Support key lifecycle management including key generation, key distribution, key update, key destruction, etc.
– Support data encryption and decryption based on security policies.
– Support privacy protection of the data of the gateway and devices.
– Support automatic discovery of services.
– Support gateway self-management and remote maintenance.
– Support gateway firmware and software update.
– Support gateway configuration according to multiple configuration modes, e.g., remote and local configuration, automatic and manual configuration and dynamic configuration based on policies.

Appendix I

Use cases of a gateway for IoT applications

(This appendix does not form an integral part of this Recommendation.)

This appendix describes some use cases of a gateway for IoT applications.

I.1 Gateway in home services

A gateway in home services can connect to electrical equipment and safety equipment through local networks and can connect to remote application servers through the communication networks. The electrical equipment and safety equipment can be controlled remotely by the gateway. Figure I.1 shows a use case of a gateway in home services.

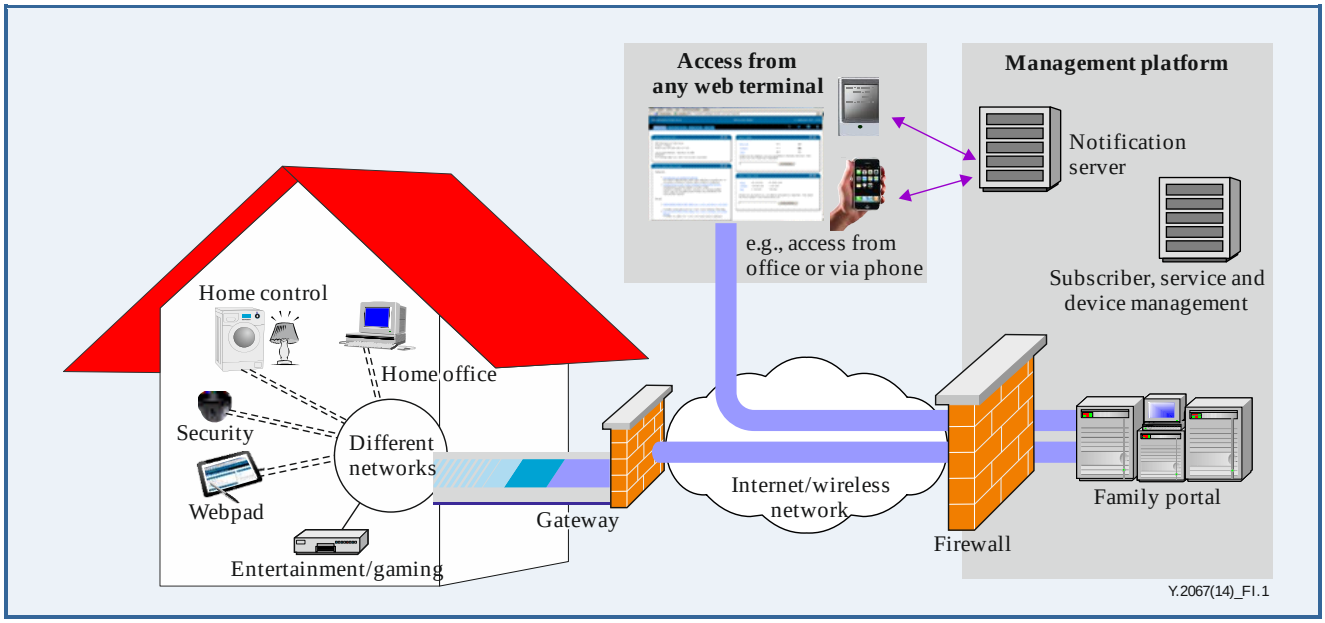


Figure I.1 – Use case of a gateway in home services

Home monitoring and management applications include:

– Monitoring of home security remotely (i.e., remote monitoring through web cameras via the TV, laptop or smartphone)

– Control of appliances (e.g., turn on/off lights, sprinklers, garage door, security alarm, thermostat, pool heater) remotely via a device with a web browser

– Scheduling of appliances (e;g. scheduling of lights, water heater, alarm system, heat, and more) via automatically created profiles

In these scenarios, as shown in Figure I.1, the gateway has a very important role.

The home owner can configure the gateway to control each of the connected devices. Control functions may be implemented through pre-set rules (time-of-day, threshold or alarm driven, etc.) or implemented through commands delivered via a SMS message.

The gateway can aggregate the information collected from multiple sensors and permit the information to be combined in order to provide more advanced services.

For example, in home security scenarios, the gateway usually integrates the inputs coming from different sensors and provides the home owner with a user interface to configure the home security system.

I.2 Gateway in automotive telematics

Automotive telematics deals with wireless communications of information and applications between a vehicle and/or its occupants and external entities. Such communications allow authorized entities such as automakers, emergency services and service centres, to interact with a vehicle and its driver, enabling enhanced safety and support services. In its most advanced modes, automotive telematics also allows motorists to safely expand mobile computing capabilities directly into their vehicles and benefit from Internet-based services.

The applications of automotive telematics can be divided into four categories:

– Driver safety and security applications
– Customer relationship management (CRM) applications for automakers and dealers
– Personal applications and services
– Business applications and services

Figure I.2 shows a typical use case of a gateway in automotive telematics.

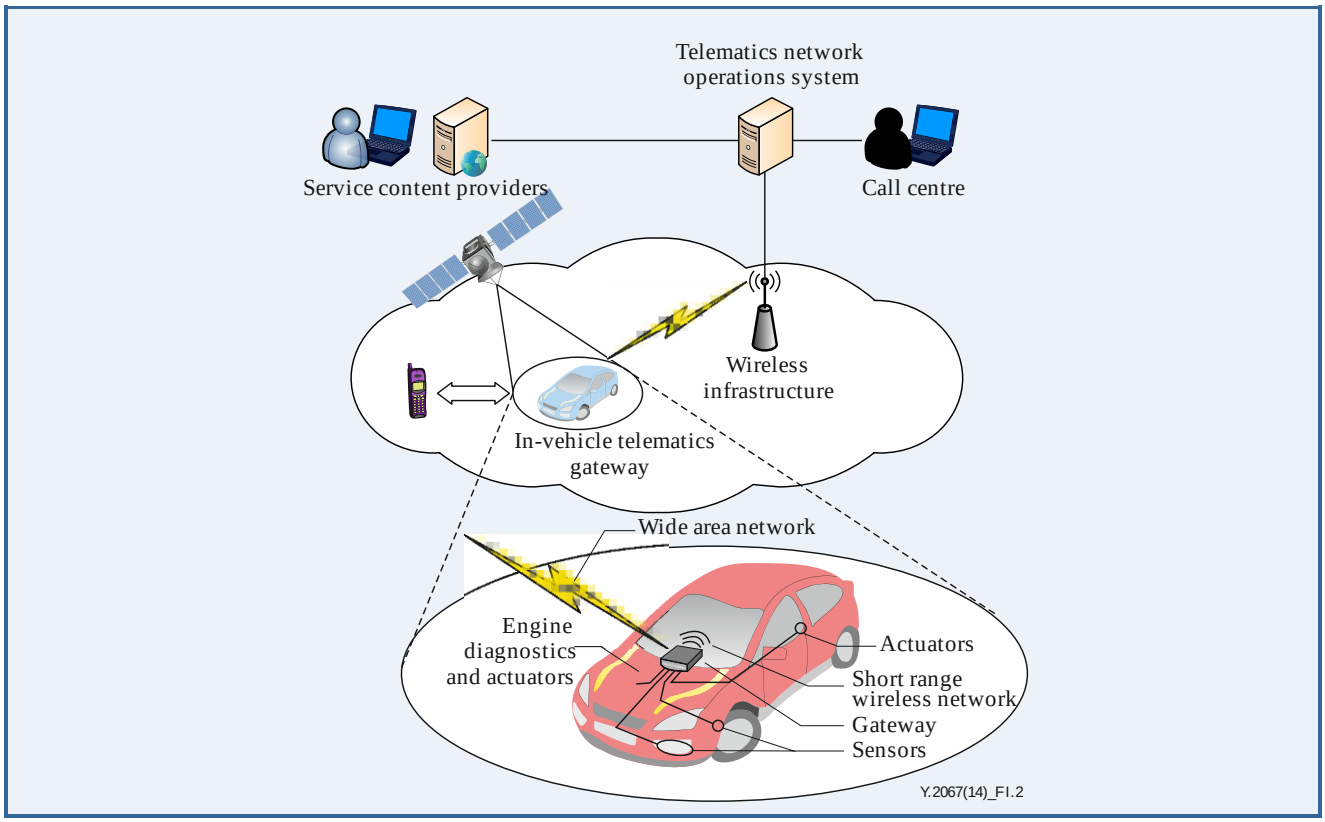


Figure I.2 – Use case of a gateway in automotive telematics

In automotive telematics, the gateway is the key entity. It is the embedded in-vehicle gateway that communicates with the automobile electronic control units (ECUs) and the global positioning system (GPS) satellite and accesses the telematics services over the wireless infrastructure.

In driver safety and security applications the gateway can monitor the various crash sensors in the vehicle and in the event of a crash, sends the details of the crash (e.g., intensity and location information) to the service centre if the crash notification service is provided. For the stolen vehicle tracking, anti-theft alarm notification and remote door service, the in-vehicle gateway can be triggered to periodically send precise location information to the service centre or can be triggered automatically by the anti-theft sensors in the vehicle. In this way, the service centre can track the vehicle.

In diagnostics services, the gateway in the vehicle can perform a detailed diagnostic scan when triggered remotely or when some key thresholds are crossed (e.g., distance travelled or time elapsed since last diagnostic scan).

I.3 Gateway in online collaborative whiteboard

Online collaborative whiteboard is an application for web-based visual collaboration.

The online collaborative whiteboard application allows distributed project participants to collaborate on developing and managing software projects. For example, online collaborative whiteboard allows participants, via the network, to share web documents (e.g., web pages) and spread sheets, exchange ideas, write and edit annotations, ask questions, post tasks and web applications and other collaboration tasks with other participants.

The data (e.g., web pages, web applications, spread sheets, etc.) transferred through the network by different devices (e.g., pad, mobile phone, laptop, etc.) are handled by the gateway for display in online collaborative whiteboard. The gateway in online collaborative whiteboard acts as the data aggregation point for real-time management and visualization. The data can be regarded as a resource for collaborative work services, such as brainstorming, virtual meeting, remote learning and remote training. Figure I.3 shows a use case of a gateway in online collaborative whiteboard.

Via the gateway in online collaborative whiteboard, the participants of the distributed project, who use different devices, can for example, upload background images and web documents and draw on top of them. All participants connected to the whiteboard can see the various changes in real-time.

The gateway in online collaborative whiteboard represents a typical use case of integration of application functionalities into the gateway. In these use cases, the gateway can process some application functions locally without communicating with remote application servers.

The features provided by the local application functionalities of the gateway in online collaborative whiteboard include:

– Providing a fast web document viewer
– Being a browser-based application
– Automatically synchronizing between project participants
– Recording and displaying of edited web documents
– Writing, inserting and replacing annotations
– Deleting web documents and web applications
– Connecting the gateway with the participants of the distributed project via the network

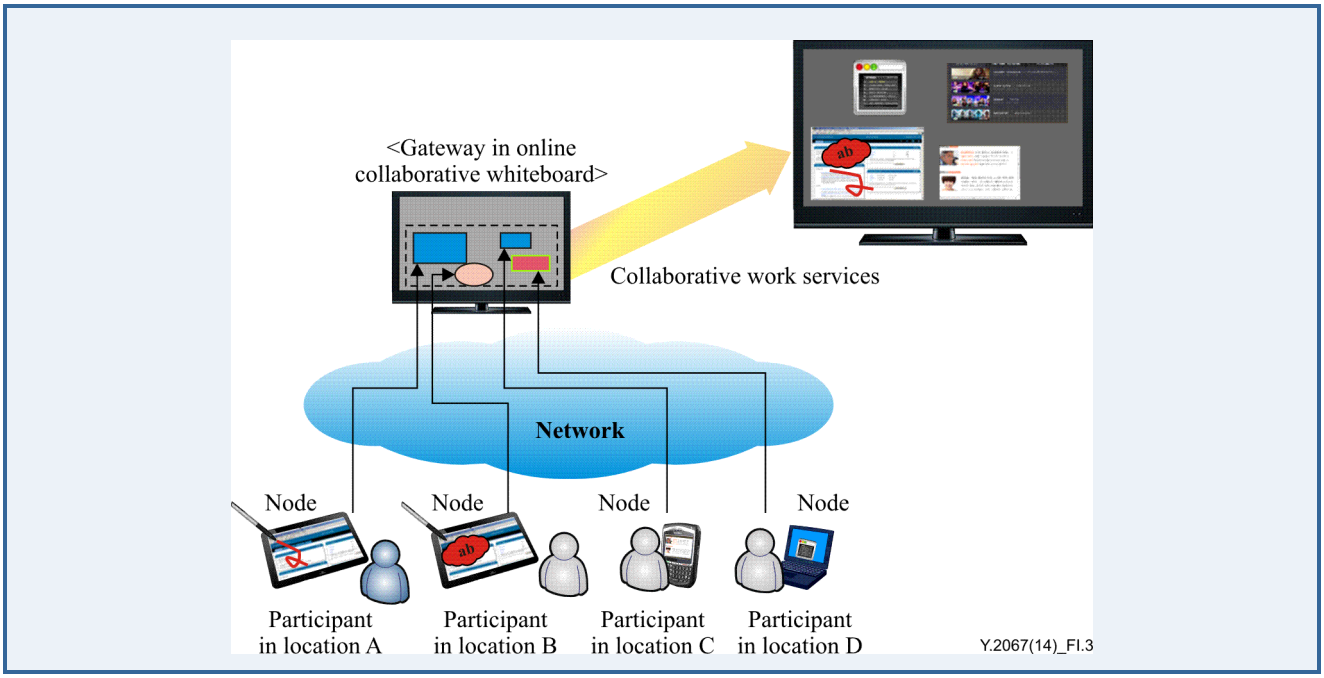


Figure I.3 – Use case of a gateway in online collaborative whiteboard

Bibliography

[b-ITU-T Y.2001] Recommendation ITU-T Y.2001 (2004), *General overview of NGN*.












Y.4102/Y.2074

Requirements for Internet of things devices and operation of Internet of things applications during disasters

Requirements for Internet of things devices and operation of Internet of things applications during disaster

Summary

Recommendation ITU-T Y.2074 provides requirements for Internet of things (IoT) devices used for operation of IoT applications in the context of disaster in addition to the common requirements of IoT in ITU-T Y.2066. It also provides requirements for the operation of IoT applications during disaster.

It is necessary to specify these requirements in order to use IoT devices and IoT applications during disaster for evacuation and rescue processes.

Appendix I describes methods concerning the assurance of integrity and reliability of data produced by IoT devices during disaster.

This Recommendation is relevant for IoT application developers and IoT service providers as well as emergency service providers.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T Y.2074	2015-01-13	13	11.1002/1000/12421

Keywords

Disaster, Internet of things (IoT), IoT application, IoT device, requirements, safety systems.

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

Table of Contents

Introduction

Every new information and communication technology (ICT) aims to be helpful and useful for users. This means that, even during disaster, ICT should aim to provide support for the rescue of users in dangerous situations. In fact, users sometimes have no time to wait for a rescue team or external help. In these cases, the only way is for users to act by themselves and try to leave the disaster area as soon as possible. It is necessary therefore to develop requirements for Internet of things (IoT) devices, as well as requirements for operation of IoT applications during disaster despite the normal operation of these applications. In fact, IoT applications usually become practically useless during a disaster when the imperative aim of IoT users is to be saved. Since the IoT infrastructure is already widely deployed, its technical resources could be very useful in saving human lives.

From a practical point of view, it is extremely difficult to develop and successfully implement a new emergency safety system, due to the complex standardization and certification procedures required for disaster management. However, it is rather easy to enhance the functionalities of existing safety systems with enhanced capabilities for support of IoT applications during disaster. Also, IoT based services could be combined with existing safety systems and be used by the safety systems during disaster.

It is important to understand that new IoT intelligence systems will never replace the existing tested and certified safety systems proven over many years; however, new IoT intelligence systems may support the capability of interaction with existing safety systems. It would still be technically possible to manage IoT applications from the administration centre of the existing safety systems during disaster.

It is expected that the interaction of these enhanced IoT applications with existing safety systems will be useful for rescue procedures during disaster, such as alerting and evacuation.

Recommendation ITU-T Y.4102/Y.2074

Requirements for Internet of things devices and operation of Internet of things applications during disaster

1 Scope

This Recommendation provides requirements for IoT devices that can be used for operation of IoT applications in the context of disaster, in addition to the common requirements of IoT [ITU-T Y.2066]. It also provides special requirements for the operation of IoT applications during disaster.

The scope of this Recommendation includes requirements for:

- IoT devices in the context of disaster;
- operation of IoT applications during disaster (for each of the three identified operating strategies).

Appendix I describes methods concerning assurance of integrity and reliability of the data produced by IoT devices during disaster.

This Recommendation is relevant for IoT application developers and IoT service providers as well as emergency service providers.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision. Users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T X.1303] Recommendation ITU-T X.1303 (2007), *Common alerting protocol (CAP 1.1)*.

[ITU-T Y.1271] Recommendation ITU-T Y.1271 (2004), *Framework(s) on network requirements and capabilities to support emergency telecommunications over evolving circuit-switched and packet-switched networks*.

[ITU-T Y.2066] Recommendation ITU-T Y.2066 (2014), *Common requirements of Internet of things*.

[ITU-T Y.2205] Recommendation ITU-T Y.2205 (2011), *Next Generation Networks – Emergency telecommunications – Technical considerations*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 alert [b-ITU-T X.674]: A warning or alarm message concerning an impending danger or problem.

3.1.2 device [b-ITU-T Y.2060]: With regard to the Internet of things, a piece of equipment with the mandatory capabilities of communication and the optional capabilities of sensing, actuation, data capture, data storage and data processing.

3.1.3 emergency telecommunications (ET) [ITU-T Y.2205]: Any emergency-related service that requires special handling from the next generation network (NGN) relative to other services. This includes government authorized emergency services and public safety services.

3.1.4 Internet of things (IoT) [b-ITU-T Y.2060]: A global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on, existing and evolving, interoperable information and communication technologies.

NOTE 1 – Through the exploitation of identification, data capture, processing and communication capabilities, the IoT makes full use of things to offer services to all kinds of applications, whilst ensuring that security and privacy requirements are fulfilled.

NOTE 2 – From a broader perspective, the IoT can be perceived as a vision with technological and societal implications.

3.1.5 next generation network (NGN) [b-ITU-T Y.2001]: A packet-based network which is able to provide telecommunication services and able to make use of multiple broadband, QoS-enabled transport technologies in which service-related functions are independent from underlying transport-related technologies. It enables unfettered access for users to networks and to competing service providers and/or services of their choice. It supports generalized mobility which will allow consistent and ubiquitous provision of services to users.

3.2 Terms defined in this Recommendation

This Recommendation defines or uses the following terms:

None.

4 Abbreviations and acronyms

This Recommendation defines or uses the following terms:

CAP Common Alerting Protocol

ET Emergency Telecommunications

ICT Information and Communication technology

IoT Internet of Things

NGN Next Generation Network

5 Conventions

In this Recommendation:

The keywords "is required to" indicate a requirement which must be strictly followed and from which no deviation is permitted if conformance to this document is to be claimed.

The keywords "is recommended" indicate a requirement which is recommended but which is not absolutely required. Thus this requirement need not be present to claim conformance.

The keywords "can optionally" and "may" indicate an optional requirement which is permissible, without implying any sense of being recommended. These terms are not intended to imply that the vendor's implementation must provide the option and the feature can be optionally enabled by the network operator/service provider. Rather, it means the vendor may optionally provide the feature and still claim conformance with the specification.

The keyword "disaster" indicates any kind of critical situation or emergency with natural or man-made origins.

The keywords "IoT device" indicate a device in IoT environment.

6 Requirements for IoT devices in the context of disaster

6.1 General requirements concerning disaster

The following Recommendations deal with telecommunications concerning disaster:

– [ITU-T Y.1271] provides network requirements and capabilities for emergency telecommunications (ET).

– [ITU-T Y.2205] specifies technical considerations that can optionally be applied within the next generation network (NGN) to enable ET. In addition, this Recommendation also outlines the underlying technical principles involved in supporting ET.

These Recommendations deal with requirements and technical aspects for emergency telecommunications. Assuming that the IoT applications will use the NGN during a disaster as a telecommunication infrastructure, these requirements are fully applicable to them.

According to [ITU-T Y.2205], it is recommended to use the common alerting protocol (CAP) defined in [ITU-T X.1303] in order to provide information interaction between alerting systems.

6.2 Requirements for IoT devices

All manufactured IoT devices are required to pass testing procedures.

These procedures should include testing of IoT devices under conditions beyond the operating range (e.g., temperature, pressure, radiation) in order to verify their safety for the environment and for humans during disaster. IoT devices must not cause complications or occurrences of emergencies of other types.

Test conditions should be selected based on the characteristics of possible emergencies in the area of deployment.

The test results and potential hazards caused by devices outside the operating range are required to be introduced in the technical characteristics of the devices.

New IoT devices are recommended to be developed with an extended range of operating characteristics (e.g., operating temperature, humidity, pressure). The requirement for IoT devices to extend the range of operating characteristics is essential for IoT applications which could potentially fail, due to the uncertainty of the environment behaviour and its impact on the IoT devices during disaster.

Dissemination of this practice is recommended on widely used IoT device types. The operation of IoT devices providing measurements during disaster might provide a database of environmental parameter measurements during disasters of different natures. Such measurements would help to make important conclusions about the stages of disaster occurrence and allow for taking them into account in the IoT device design phase.

7 Requirements for operation of IoT applications during disaster

This clause describes requirements for IoT applications concerning their operation during disaster. In particular, clauses 7.1 to 7.3 describe requirements for each of the three identified operating strategies for IoT applications related to disaster, and clause 7.4 describes switching between two or more operation strategies during disaster.

To improve the efficiency of the infrastructure resources associated with the operation of IoT applications, it is recommended that IoT applications implement one or more of the following operation strategies related to disaster.

All strategies assume that the IoT applications do not continue normal operation during a disaster, but instead perform only tasks aimed at rescuing people.

False emergency alerts are possible: a state of emergency may be cancelled (for example, in case of false emergency detection) and, in this case, the IoT application switches back to its normal operation. The time period required to decide on a false alert (continuation of current operation or switch back to normal operation) has a different duration for each particular implementation, depending on its complexity.

7.1 IoT applications with dedicated operation mode

If an IoT application has a dedicated operation mode, which can be activated in case of emergencies, it can be used without any further action or external control. Figure 1 shows the operation mode change of IoT applications following this strategy.

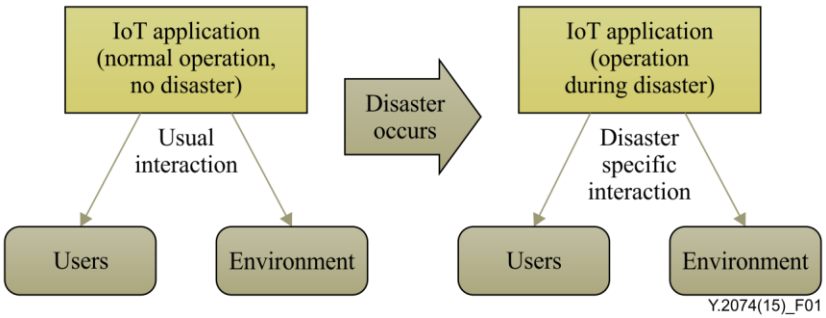


Figure 1 – Operation mode change for IoT applications with dedicated operation mode activated during disaster

Sensor network based applications designed for positioning users in a building and having dedicated operation modes activated during disaster can be extremely effective for self-evacuation from the building in case of fires, earthquakes or other disasters.

Another example of this operational strategy is that one of the IoT applications can act as a safety system.

NOTE – There are prototypes of such safety systems based on wireless sensor technologies (e.g., as described in [b-ITU-T Y.2222]), but they are not widely used because of long and complex standardization and certification procedures for the safety system equipment.

IoT applications with dedicated operation mode activated during disaster are required to comply with all appropriate regulatory rules.

7.2 IoT applications temporally providing resources to external safety systems

Normally, IoT applications have specific purposes and, for the most part, are not intended to assist or help users during disaster. Consequently, the resources of IoT applications should be assisted by external safety systems in order to improve the efficiency of the disaster management process. Figure 2 shows the operation mode change of IoT applications following this strategy.

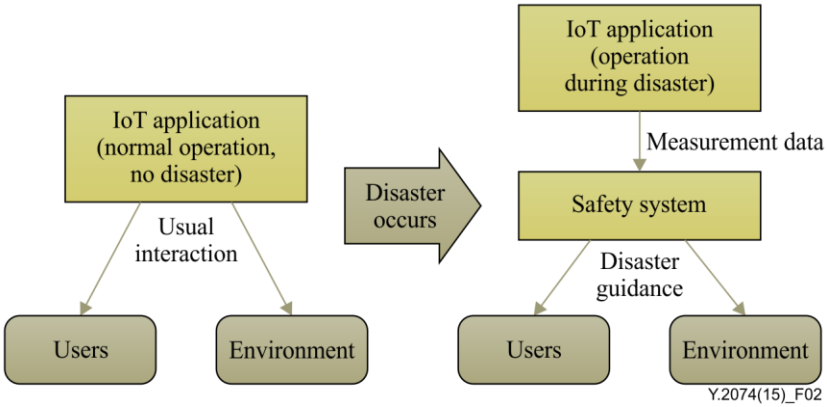


Figure 2 – Operation mode change for IoT applications temporally providing resources to external safety systems during disaster

IoT applications designed for users within a building or other environment equipped with a safety system should temporally (during disaster) share both the IoT application control capability and all kinds of measurement data with the safety system. These resources might be useful for the safety system operation, for example, data from the various sensors such as temperature and humidity in case of fire.

To simplify the integration of IoT applications with external safety systems, it is recommended to use CAP [ITU-T X.1303] for the interaction between IoT applications and external safety systems. CAP is a two-way communication protocol that can enable both the transmission of data from IoT applications to safety systems and the transmission of alert messages from safety systems to IoT applications.

The main disadvantage of this operation strategy is the possibility of failures of functional components of the IoT infrastructure if these are not designed to operate correctly during disaster. Such failures, in the case of functional components that are needed during disaster management processes, may cause negative consequences. These failures are possible due to the fact that there are no special certification procedures for the functional components of the IoT infrastructure to ensure correct operation during disaster, in contrast to the certified procedures of safety systems.

7.3 IoT applications with external control of operation during disaster

The third operation strategy for IoT applications during disaster involves a complete transfer of control capabilities and measurement data from IoT applications to external safety systems or external control centres.

NOTE 1 – The complete transfer of control capabilities implies the termination of the resource management process by the IoT application itself.

NOTE 2 –An external control centre may be, for example, an organization or a functional unit of an organization that carries the full legal and administrative responsibility for correct disaster management in a given area.

Figure 3 shows the operation mode change of IoT applications following this operation strategy.

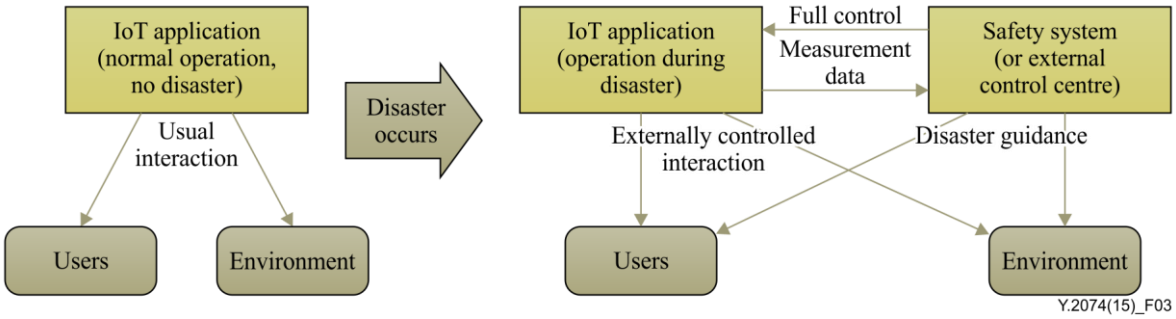


Figure 3 – Operation mode change for IoT applications with external control of operation during disaster

In this operation strategy, with respect to IoT applications following the operation mode described in clause 7.1, the users' behaviour during disaster is fully controlled by external safety systems and alerts.

The main purpose of this operation strategy is to ensure that the most effective use of all available resources of IoT applications, via proper resource management, is performed by safety systems or external control centres.

Appendix I describes methods concerning assurance of integrity and reliability of the data produced by IoT devices. A monitoring and control centre for IoT devices, as described in Appendix I, may serve as an external control centre for IoT applications for this operation strategy.

Similarly to the operation strategy described in clause 7.2, it is recommended to use CAP [ITU-T X.1303] for the interaction between IoT applications and external safety systems or external control centres in this operation strategy.

7.4 Switching between two or more operation strategies during disaster

Depending on the purpose of the IoT application and its capabilities, a combination of one or more operation strategies can be implemented in the IoT application. This involves the IoT application capability to switch between operation strategies in case of the appearance of certain external conditions, such as reception of control signals, excess of a prescribed degree in sensor readings, etc.

As an example, the operation of the IoT application can be realized as follows:

Consider an IoT application (within a geographical area) equipped with a safety system (external with respect to the IoT application). If the monitoring of the IoT device data shows an emergency occurring during normal operation, the IoT application automatically switches to dedicated operation mode for operation during disaster and implements the strategy described in clause 7.1.

At the end of the false alert decision time, the IoT application continues operation in dedicated operation mode or switches back to normal operation mode (in case of a false alert). If operation in dedicated operation mode continues, before the catastrophic phase of disaster, the IoT application generates customized information for each person, involved in the disaster, to manage his or her rescue.

Upon the occurrence of the catastrophic phase, when the IoT application is unable to manage rescues because of reduced capabilities, the IoT application switches to the operation strategy described in clause 7.2 (monitoring and transmission of gathered data to the external safety system). This may help save lives during the subsequent emergency rescue phase and will monitor the development of the disaster.

Appendix I

Methods concerning assurance of integrity and reliability of the data produced by IoT devices during disaster

(This Appendix does not form an integral part of this Recommendation.)

Ubiquitous IoT devices may play a significant role in people's everyday life, influencing their decisions and actions. Hence, people may depend on their IoT devices, in particular on their information and sensor readings, as well as on the derived actions that impact the environment. Therefore, the integrity and reliability of data produced by IoT devices are very significant issues for the IoT in general.

The problem with the integrity and reliability of data produced by IoT devices becomes especially relevant during both natural and man-made disasters, where the integrity of the IoT devices themselves may not be guaranteed.

To preserve the integrity and reliability of data produced by IoT devices, it is necessary to establish a trusted environment for the IoT devices' operation. For this purpose, it is important to determine the scope of liability for the IoT devices' behaviour in general, e.g., for any incorrect sensor readings. There are two methods to achieve this goal:

1. the manufacturer of IoT devices is fully responsible for any malfunction of the produced IoT device and guarantees appropriate IoT device behaviour;
2. an independent authorized centre is fully responsible for any malfunction of an IoT device under its control (under its jurisdiction), and guarantees appropriate IoT device behaviour.

The first method is less effective than the second due to the complicated interaction between users and the manufacturers responsible for the user's IoT devices, because of the possible variety of IoT devices from different manufacturers used within the same deployment area. This problem becomes especially relevant during disaster, when the integrity and reliability of the data produced by IoT devices becomes a matter of protecting human lives. During disaster, neither users nor rescue services, or IoT devices will be able to make contact with the manufacturer of each particular IoT device to confirm the integrity and reliability of its data.

The second method is much more concrete in that it consists of the establishment of monitoring and control centres for IoT devices. These centres will be responsible for the correct operation of the IoT devices under their jurisdiction.

I.1 General overview of a monitoring and control centre for IoT devices

A monitoring and control centre (the Centre) for IoT devices is an organization, or functional unit of an organization, which carries full legal and administrative responsibility for the correct operation of the IoT devices under its jurisdiction. It also monitors the IoT devices and stores information about operations during disaster. The main goal of a monitoring and control centre for IoT devices is to check the integrity and reliability of information provided by the IoT devices under its jurisdiction. In addition, the Centre is responsible for prompt notification to users and/or owners of the IoT devices if malfunctions of any IoT device are identified.

In case of threat of disaster or during disaster, the Centre is responsible for:

- monitoring the status of the IoT devices under its jurisdiction and their output data (e.g., sensors' readings);
- identifying improperly operating IoT devices and promptly notifying users and/or owners about the malfunctions;

- determining the disaster area and the nature and parameters of the disaster, taking into account the information obtained from the IoT devices under its jurisdiction and external sources of information (e.g., emergency agencies);
- managing the IoT devices under its jurisdiction in order to safely evacuate people from the disaster area;
- recording and storing information obtained during disaster and the history of operations during disaster.

I.2 The distribution of the monitoring and control centre's responsibilities to local centres

Ubiquitous IoT devices are present in large quantities in apartments, houses, organizations, streets, public places, etc.

In the case of a monitoring and control centre for IoT devices, it is possible for all IoT devices in a given house or building to be under the jurisdiction of one local centre. Similarly, all IoT devices in other areas, for example, on the same street, could be managed by other local centres. All these local centres could be integrated into the infrastructure of the root Centre.

The infrastructure of the root Centre can be organized as a multi-level hierarchy containing monitoring and control nodes of several levels responsible for IoT devices in different: buildings (local centres), cities (municipal centres), regional (regional centres) and countries (federal centres).

Additionally, the responsibility of local centres may be distributed on an IoT device purpose basis. For example, the Centre may manage several local centres, one being responsible for IoT devices for household purposes, another for IoT devices for traffic management purposes, a third one for IoT devices for security system purposes, etc.

The following clauses describe possible working scenarios of the monitoring and control centre.

I.3 The monitoring and control centre's working scenarios

The main goal of the Centre is to check the integrity and reliability of the information provided by the IoT devices under its jurisdiction. This goal can be achieved in the following ways:

1. comparing the sensors' readings of the IoT devices under the Centre's jurisdiction, with the readings of autonomous (duplicated) sensor networks;
2. intelligent monitoring of the sensors' readings, under the Centre's jurisdiction, consisting of data collection and mathematical analysis (data mining) of the obtained information, thus allowing the identification of IoT device malfunctions.

Both methods may be implemented and used in combination in the appropriate proportion.

The above methods are described in more detail in clauses I.3.1 and I.3.2.

I.3.1 Autonomous sensor network

The Centre deploys autonomous sensor networks containing sensors of various physical parameters, which duplicate the sensors of the IoT devices under the Centre's jurisdiction.

The autonomous sensor network is required to cover the entire area under the Centre's jurisdiction. For instance, a local indoor centre should deploy a sensor network which covers the indoor area that contains IoT devices under the Centre's control.

The sensors of this autonomous sensor network are considered reference sensors, i.e., their readings are taken as reference values of physical parameters in this area. It is expected that the reference sensors are certified by a trusted and properly certified organization.

The Centre collects data from the IoT devices under its jurisdiction, and compares them with the reference values. On this comparison basis, the Centre makes decisions about integrity and reliability of the data produced by the IoT devices.

The advantage of this method is the potentially high-reliability of reference sensors, independent from the IoT devices. Hence, the malfunctions of IoT devices are identified with high accuracy.

The disadvantages of this method are the cost and complexity of deploying autonomous sensor networks and the possible failures of the reference sensors during disaster.

I.3.2 Intelligent monitoring

Intelligent monitoring concerns the collection of device information and sensor readings obtained from the IoT devices under the Centre's jurisdiction, and the mathematical analysis of this information. This includes, but is not limited to, the methods of statistical analysis and correlation signal processing.

Intelligent monitoring allows the identification of out of order IoT devices or their sensors within a group of similar devices.

The advantage of this method is complete independence from external parameters of the environment, allowing operation in every situation during disaster.

The disadvantage of this method is the need to have a group of similar IoT devices for more reliable determination of malfunctions.

I.4 Use of the stored data

The Centre implements monitoring, recording and storing of device information and sensor readings obtained from the IoT devices under its jurisdiction, including those obtained immediately before and during disaster.

This functionality allows the Centre to operate as a "black box" in emergencies. The Centre is assumed to help identify the causes of emergencies, in a similar way to what is done by the black box in aircraft.

Historical data collected in the Centre's data store may be used to improve the methods of intelligent monitoring and to develop IoT device management and control methods under the threat of disaster or during disaster in order to achieve the greatest possible evacuation of people, safely from the disaster area.

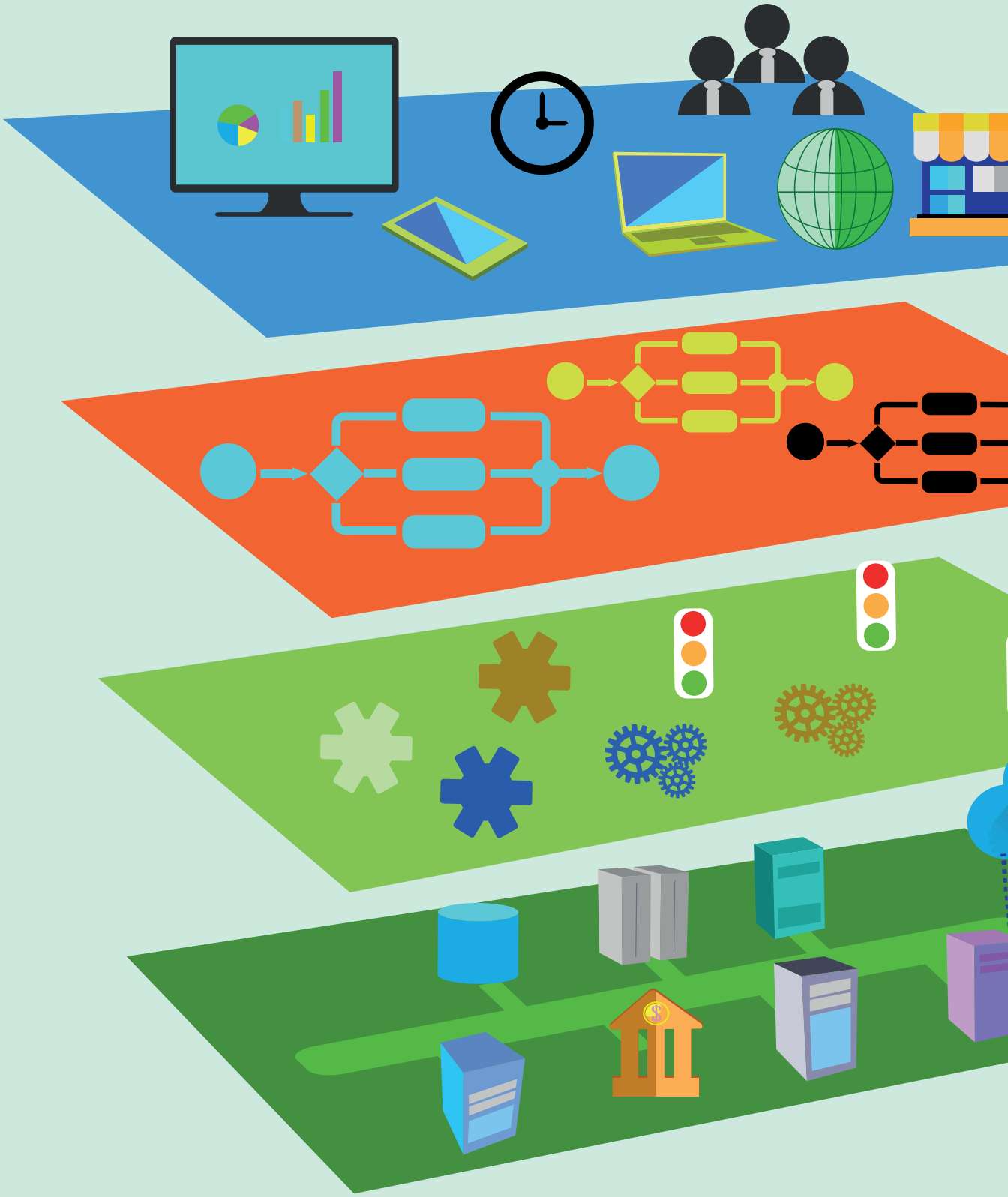
Bibliography

[b-ITU-T X.674] Recommendation ITU-T X.674 (2011), *Procedures for the registration of arcs under the Alerting object identifier arc*.

[b-ITU-T Y.2001] Recommendation ITU-T Y.2001 (2004), *General overview of NGN*.

[b-ITU-T Y.2060] Recommendation ITU-T Y.2060 (2012), *Overview of Internet of things.*

[b-ITU-T Y.2222] Recommendation ITU-T Y.2222 (2013), *Sensor control networks and related applications in a next generation network environment*.



Y.4103/F.748.0

Common requirements for Internet of things (IoT) applications

Common requirements for Internet of things (IoT) applications

Summary

Recommendation ITU-T F.748.0 includes the common requirements for Internet of things (IoT) applications enabling advanced services by interconnecting (physical and virtual) things based on, existing and evolving, interoperable information and communication technologies.

The requirements defined in this Recommendation are general requirements, and can therefore be applied to many kinds of IoT applications regardless of their types and characteristics.

This Recommendation is based on the high-level requirements and the reference model defined in Recommendation ITU-T Y.2060.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T F.748.0	2014-10-14	16	11.1002/1000/12228

Keywords

Internet of things, IoT, things, ubiquitous computing.

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

Table of Contents

Recommendation ITU-T Y.4103/F.748.0

Common requirements for Internet of things (IoT) applications

1 Scope

This Recommendation defines the common requirements for Internet of things (IoT) applications based on [ITU-T Y.2060].

This Recommendation covers the following from the application point of view:

– overview of the IoT applications;

– characteristics of the IoT applications;

– common requirements for the IoT applications.

NOTE – This Recommendation mainly focuses on the viewpoint of the IoT applications. The network layer aspect of the IoT is out of scope of this Recommendation.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T Y.2060] Recommendation ITU-T Y.2060 (2012), *Overview of the Internet of things.*

[ISO/IEC 29182-1] ISO/IEC 29182-1 (2013), *Sensor networks: Sensor Network Reference Architecture (SNRA) – Part 1: General overview and requirements*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 data fusion [ISO/IEC 29182-1]: Deriving information by processing data from various sources.

3.1.2 Internet of things (IoT) [ITU-T Y.2060]: A global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies.

NOTE 1 – Through the exploitation of identification, data capture, processing and communication capabilities, the IoT makes full use of things to offer services to all kinds of applications, whilst ensuring that security and privacy requirements are fulfilled.

NOTE 2 – From a broad perspective, the IoT can be perceived as a vision with technological and societal implications.

3.1.3 thing [ITU-T Y.2060]: With regard to the Internet of things, this is an object of the physical world (physical things) or of the information world (virtual things), which is capable of being identified and integrated into communication networks.

3.2 Terms defined in this Recommendation

None.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

IoT Internet of Things

M2M Machine-to-Machine

MOC Machine Oriented Communication

MTC Machine-Type Communication

QoS Quality of Service

RFID Radio Frequency Identification

SOA Service Oriented Architecture

USN Ubiquitous Sensor Network

5 Conventions

In this Recommendation:

– The keywords "is required to" indicate a requirement which must be strictly followed and from which no deviation is permitted if conformance to this document is to be claimed.

– The keywords "is recommended" indicate a requirement which is recommended but which is not absolutely required. Thus this requirement need not be present to claim conformance.

– The keywords "can optionally" and "may" indicate an optional requirement which is permissible, without implying any sense of being recommended. These terms are not intended to imply that the vendor's implementation must provide the option and the feature can be optionally enabled by the network operator/service provider. Rather, it means the vendor may optionally provide the feature and still claim conformance with the specification.

6 Characteristics of things in the IoT

[ITU-T Y.2060] explains the concept of the Internet of things (IoT) as a vision with technological and social implications. In addition, the IoT can be viewed as a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on, existing and evolving, interoperable information and communication technologies.

Things in the IoT can be characterized using five criteria: fundamental, common, social, autonomy and capability of self-replication or control [b-CERP-IoT].

6.1 Fundamental characteristics

Fundamentally, things have characteristics as follows:

– Things can be "real world entities" or "virtual entities";

– Things have identity and there are means for identifying (automatically or manually) them, for example barcode and radio frequency identification (RFID);

– Things and their associated information (their virtual representations) respect the privacy, security and safety of other things or people with which they interact;

– Association (or relation) among things (both physical and virtual) and the related information is as important as the things and the information in IoT application; and

– Things communicate with each other via the infrastructure or the infrastructure-less communications means.

6.2 Common characteristics

There are common characteristics in things as follows:

– Things can use services that act as interfaces to things;
– Things could be competing with other things for resources, services and subject to selective pressures;
– Things may have embedded or attached sensors (and/or actuators), thus they can interact with their environment;
– Things use protocols to communicate with each other and the infrastructure; and
– Things are environmentally safe, where things are devices for identification, sense or communication, etc.

6.3 Social characteristics

Things have the following social characteristics towards other things or people:

– Things can communicate with other things, computing devices and with people;
– Things can collaborate to create groups or networks;
– Things can initiate communication without human intervention;
– Things can create, manage and destroy other things; and
– Things can respect the privacy, security and safety of other things or people with which they interact.

6.4 Autonomy of things

Autonomy is an important feature of the IoT. The followings are characteristics of autonomous things:

– Things can do many tasks autonomously;
– Things can negotiate, understand and adapt to their environment;
– Things can extract patterns from the environment or to learn from other things;
– Things can take decisions through their reasoning capabilities; and
– Things can selectively transform or evolve and propagate information.

6.5 Capability of self-replication or control

Autonomous things tend to have a capability of self-replication or control under specific conditions.

– Things can create, manage and destroy other things.

7 Characteristics of IoT applications

NOTE – Characteristics given in clauses 7.1 to 7.5 refer to [ITU-T Y.2060] and characteristics given in clauses 7.6 to 7.10 refer to [ISO/IEC 29182-1].

7.1 Interconnectivity

In the IoT, anything will be inter-connected with the global information and communication infrastructure.

7.2 Things-related services

The IoT is capable of providing thing-related services within the constraints of things, such as privacy protection and semantic consistency between physical things and their associated virtual things. In order to provide thing-related services within the constraints of things, both the technologies in the physical world and information world will change.

7.3 Heterogeneity

The devices in the IoT are heterogeneous and based on different hardware platforms and networks. They can interact with other devices or service platforms through different networks.

7.4 Dynamic changes

The state of devices change dynamically, e.g., sleeping and waking up, connected and/or disconnected as well as the context of devices including location and speed. Moreover, the number of devices involved can change dynamically.

7.5 Enormous scale

The number of devices that need to be managed and that communicate with each other will be at least an order of magnitude larger than the devices connected to the current Internet. The ratio of communication triggered by devices as compared to communication triggered by humans will noticeably shift towards device-triggered communication. Even more critical will be the management of the data generated and their interpretation for application purposes. This relates to semantics of data, as well as efficient data handling.

7.6 Data gathering and processing by things

IoT devices gather data from the real world and pre-process the data. Then IoT services are provided to the user, either directly from IoT devices or via a service provider.

NOTE – As seen in the definition of the IoT, sensor network technology is one of the key enablers for IoT services. [ISO/IEC 29182-1] describes the characteristics and defines the requirements of sensor networks. This clause rephrases the characteristics described in [ISO/IEC 29182-1] for reflecting the characteristics of the IoT.

7.7 Collaborative data processing

In IoT applications, IoT devices may collaborate to solve complex sensing problems such as the detection, classification and tracking of objects in the physical world. The data from an IoT device may be pre-processed and refined at the IoT device acquiring the data or at another IoT device.

Depending on the application, intermediate data, such as features or estimated parameters, may be extracted from the captured data during pre-processing. The results from this pre-processing may be shared among IoT devices. Once shared, the intermediate data from the multiple IoT devices can be transformed into context data and situation information by data fusion.

NOTE – As seen in the definition of the IoT, sensor network technology is one of the key enablers for IoT services. [ISO/IEC 29182-1] describes the characteristics and defines the requirements of sensor networks. This clause rephrases the characteristics described in [ISO/IEC 29182-1] for reflecting the characteristics of the IoT.

7.8 Maintenance-free operation

IoT devices may have to operate for long periods of time without maintenance or technical support to resolve problems. Provision of remote diagnostics and resolution may be required.

NOTE – As seen in the definition of the IoT, sensor network technology is one of the key enablers for IoT services. [ISO/IEC 29182-1] describes the characteristics and defines the requirements of sensor networks.

This clause rephrases the characteristics described in [ISO/IEC 29182-1] for reflecting the characteristics of the IoT.

7.9 Self-adaptation

IoT devices may self-adapt to accommodate changing operating conditions, to support robustness and reliability and to optimize resource management and functionality.

NOTE – As seen in the definition of the IoT, sensor network technology is one of the key enablers for IoT services. [ISO/IEC 29182-1] describes the characteristics and defines the requirements of sensor networks. This clause rephrases the characteristics described in [ISO/IEC 29182-1] for reflecting the characteristics of the IoT.

7.10 Energy efficiency and operating lifetime

Energy management is important in many IoT devices where the IoT device is battery-operated and it is desirable for the device to be operational for as long as possible. Energy harvesting technologies may help with energy management and extending the device lifetime.

NOTE – As seen in the definition of the IoT, sensor network technology is one of the key enablers for IoT services. [ISO/IEC 29182-1] describes the characteristics and defines the requirements of sensor networks. This clause rephrases the characteristics described in [ISO/IEC 29182-1] for reflecting the characteristics of the IoT.

7.11 Embedded intelligence

Embedded intelligence can be defined as the capability of things to collect information of the surroundings and analyse it to learn the state of the real world, possibly interacting with other widely deployed things. Smart things (or intelligent objects [b-SPRINGER-TRON]) are things with embedded intelligence that can interoperate with each other and can act independently if necessary.

Embedded intelligence (sometimes called ambient intelligence) and autonomous control will be integrated into IoT devices. The IoT is a large non-deterministic and open network in which auto-organized or intelligent entities (web services, service oriented architecture (SOA) components), and virtual objects will interoperate with each other, and shall be able to act independently depending on the context, circumstances or environments.

7.12 Location considerations

The precise geographic location of a thing and its precise geometrical dimensions will be critical (i.e., some things in the IoT will be sensor nodes in sensor networks. Sensor node location is important for many applications.)

It is desirable to provide the location context to the things and, if appropriate, to IoT applications in order to take full of advantage of the IoT.

7.13 Auto-configuring reliable information transmission over ubiquitous networks

According to the diversity of IoT services, the services information categories become much richer and differentiation of quality of service (QoS) in each category becomes more complicated than in existing networks. Information service, rather than connection service, will be a basic operation feature of the networks used in the IoT. As an infrastructure and support environment for a ubiquitous information society, ubiquitous networks will be an important feature in the IoT service environment. Reliable transmission technologies that are easy to set up or are auto-configuring are required in existing and/or evolving networks to provide ubiquitous and intelligent services and provide people with rich real-world information.

7.14 Security

In the IoT, all things are connected which results in significant security threats, such as threats towards confidentiality, authenticity and integrity of both data and services. A critical example of security requirements is the need to integrate different security policies and techniques related to the variety of devices and user networks in the IoT.

7.15 Privacy

Many things have their owners and users. Sensed data of things may contain private information concerning their owners or users. Unlike ordinary desktop and other legacy applications, in the IoT data may be collected by a ubiquitous sensor network (USN) without human users being aware of such collection.

7.16 Infrastructure-less versus infrastructure-based application

Some IoT applications, such as machine-to-machine (M2M), machine-type communications (MTC), machine oriented communications (MOC) or USN-based applications, require network infrastructure (for example, Internet or mobile telecommunication networks as a delivery/backbone network). In contrast, some applications used in smart home or smart office may not require network infrastructure. These two types of application will have different requirements. Still, it is required that these two types of applications be able to talk to each other through proper gateways.

7.17 Observation and/or actuation vs. data exchanges

Typically, things with embedded sensors observe physical environments and acquire information about surroundings. Based on this information, some devices are actuated (actuators) and physical surrounding can be controlled. Some applications, such as RFID applications for example, use data exchanges between things. In this type of application, data that the thing acquires from outside and/or holds inside are essential to provide the IoT services.

7.18 Application domains

IoT applications can be deployed in many domains. Table 7-1 lists typical application domains. This list is not exhaustive.

In the IoT, inter-domain applications will also be very common.

For example, a pre-planning of an outdoor outing by a family or a group of friends can use the following services.

– Information provided about transportation: train timetable and its operation status, expressway congestion, etc.
– Weather services of regions to be visited.
– Information about the environmental conditions of natural habitats such as mountains, rivers, lakes, marshes, etc. of the area to be visited.
– If the outing is overnight, information related to reservations (for example, hotel, camping sites or restaurants).
– If someone in the group is physically challenged, information on accessibility.

Table 7-1 – Example of IoT application domains

Domains	Description	Examples
Industry	Activities involving financial or commercial transactions among companies, organizations and other entities: These include business to business (B2B) and business to customers (B2Cs)	Manufacturing, logistics, service sector, banking, financial governmental authorities, intermediaries, etc.
Environment	Activities regarding the protection, monitoring and development of all natural resources	Agriculture and breeding, recycling, environmental management services, energy management, etc.
Society	Activities/initiatives regarding the development and inclusion of societies, cities and people	Governmental services towards citizens and other society structures (e-participation), e-inclusion (e.g., elderly, disabled people), public transportation, etc.
Home	Activities concerning individual and family members	Health monitoring for oneself (weight, sleeping hours, etc.), nutrition care by monitoring of diet taken by family members using Cloud database.

8 Common requirements for IoT applications

NOTE – Requirements given in clauses 8.2 to 8.10 refer to [ITU-T Y.2060]. The requirements of the IoT as a whole and the requirement for a single IoT application need to be considered separately. The requirements for IoT applications are defined here.

8.1 Identification

For communication between things, unique identification of the thing to communicate is required before communication. Many identification schemes may be used, depending on the application (e.g., RFID applications, sensor network applications and M2M applications).

8.2 Identification-based connectivity

IoT applications are required to support the establishment of connectivity between a thing and the IoT based on the thing's identifier.

A common approach is required for handling the possible assignment of heterogeneous identifiers to different types of things (see clause 7.16).

8.3 Interoperability

Interoperability is required to be ensured among heterogeneous and distributed systems for provision and consumption of a variety of information and services (see clauses 7.1 and 7.3). Proper gateways are required to be provided if infrastructure-less and infrastructure-based applications are mixed.

8.4 Autonomic networking

Autonomic networking (such as self-management, self-configuring, self-healing, self-optimizing, and self-protecting techniques and/or mechanisms; see clause 7.13) may be supported in networking control functions of the IoT in order to adapt to different application domains (see clause 7.18), different communication environments (see clauses 7.1, 7.3 and 7.16) and large numbers and types of devices (see clause 7.5).

8.5 Autonomic services provisioning

The services may be provided by capturing, communicating and processing automatically the data of things based on the rules configured by operators or customized by subscribers. Autonomic services may depend on the techniques of automatic data fusion and data mining. Some things may be equipped with actuators to act on the surrounding environment.

8.6 Location-based capabilities

The IoT as a whole supports location-based services. Location-based capabilities may be optionally supported by IoT applications. Certain types of communications and services will depend on the location information of things and/or users. It is required to sense and track the location information automatically, unless security and/or privacy concerns dictate otherwise, when location information is necessary for an IoT application.

8.7 Security

Generally accepted measures for providing confidentiality, authenticity and integrity of data are required to be provided to the things and servers after a proper threat-analysis is performed. The proper threat-analysis is required to pay attention the characteristics of an IoT application in particular (see clause 7.14)

8.8 Privacy protection

Privacy protection is required to be supported in the IoT. IoT applications are required to support privacy protection during data transmission, aggregation, storage, mining and processing. Privacy protection is recommended to strike a balance and not to impose an undue a barrier to data source authentication provided by the authentication requirement.

8.9 Plug and play

Plug and play capability is an important feature to be supported in the IoT in order to enable on-the-fly generation for seamless integration and cooperation of interconnected things with applications, and for improving responsiveness of things to application requirements. IoT applications are recommended to support plug and play features (see clauses 7.8 and 7.13).

8.10 Manageability

Manageability is required to be supported in the IoT in order to ensure normal network operations. IoT applications usually work automatically without people's participation, but their whole operation process should be manageable by the relevant parties.

8.11 Compliance with laws and regulations

Communications and services may be constrained by laws and regulations. Such constraints are often found in location-based services (see clause 8.6), and services related to human body.

Often security and privacy requirements are imposed by laws and regulations (see clauses 8.7 and 8.8). These are required to be obeyed in a global manner and IoT applications must meet local requirements as well.

8.12 Awareness of service

Even though IoT services are generally available without human intervention, humans (the users of IoT services) may need to be aware of IoT services surrounding them. When IoT services are provided to a user, it is recommended that the user be able to notice (discover) their presence. This has implication for security and privacy protection (for example surveillance, see clauses 8.7 and 8.8) [b-EC-PRIVACY].

8.13 Mobility support

IoT devices can be either mobile or static. When an IoT device moves from place to place, it is necessary to support mobility at the application level (such as service mobility between different service providers) as well as the network level. Therefore, IoT applications are recommended to support mobility of IoT devices.

8.14 Scalability support

As stated in clause 7.5, the scale of the network of IoT devices may be huge. IoT applications are recommended to support scalability, including the number of devices, the volume of data traffic that needs to be communicated, etc.

8.15 Robustness against dynamic changes

Clause 7.4 describes dynamic change of status of an IoT device. Therefore, IoT applications are recommended to provide robustness, e.g., seamless continuity and sustainability, against dynamic transformation and change of IoT devices.

8.16 Self-organization (re-organization) and self-healing

IoT devices may provide maintenance-free operation and may be self-adaptable as described in clauses 7.8 and 7.9. For coping with these characteristics, IoT applications are recommended to support self-organization (re-organization) and self-healing of the application and the network on the IoT device to recover from failure or mal-function. This requirement is related to robustness against dynamic transformation and changes in clause 8.15.

8.17 Energy efficient operation

IoT applications are recommended to operate IoT thing devices in a way that minimizes the necessary energy for operation. This will ensure longer battery life, if the devices are battery-operated (see clause 7.10), and longer maintenance-free operation (see clause 7.8). This will also help reduction of carbon gas emissions.

8.18 Common data format for collaborative data processing

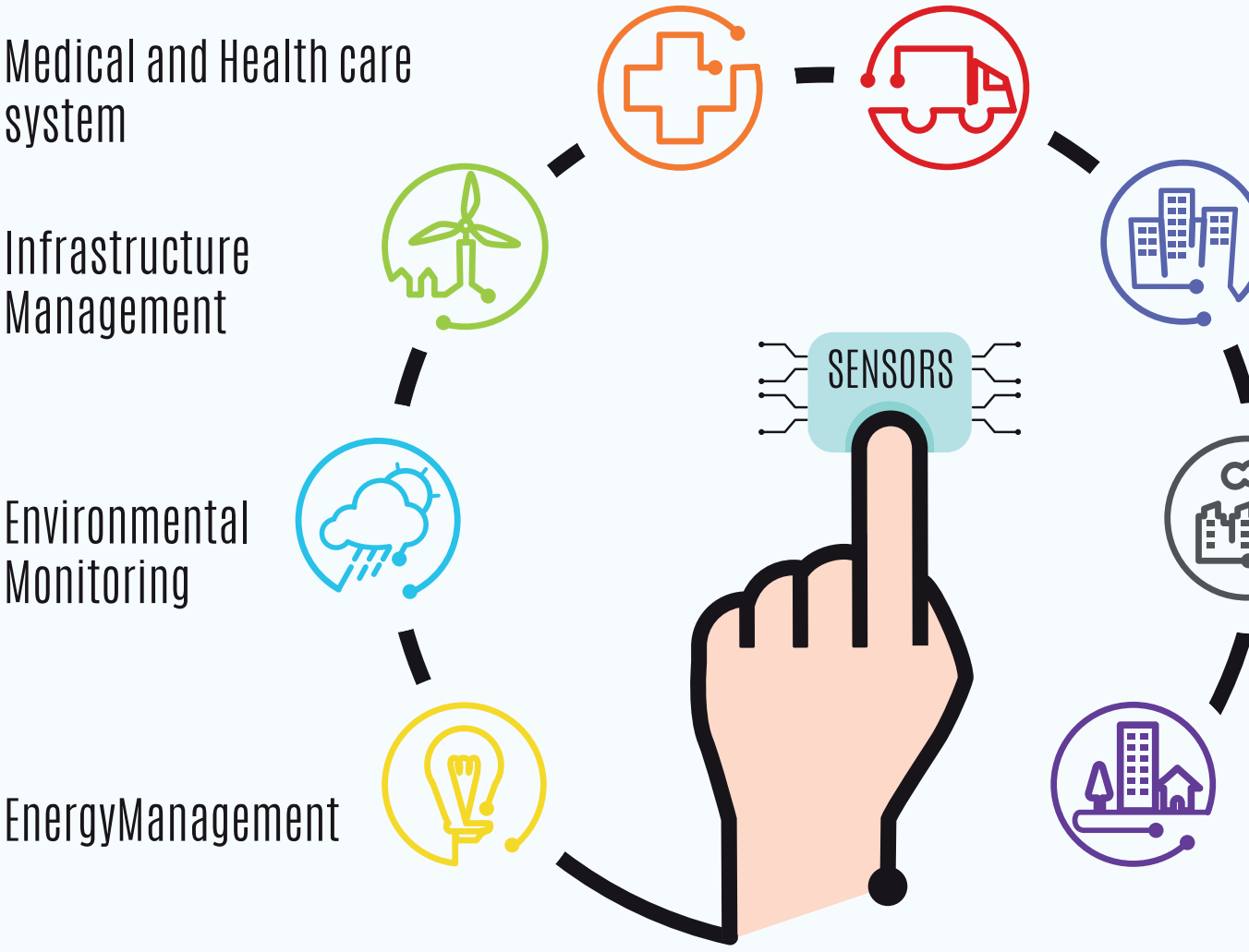
IoT applications are recommended to adopt common data formats (see clause 7.7). This is to facilitate the mixing and mashing of data gathered by many IoT applications (which adds value to the collected data as a whole) as well as to facilitate data exchange.

Bibliography

[b-CERP-IoT] *Vision and Challenges for Realizing the Internet of Things*, CERP-IoT (Cluster of European Research Projects on the Internet of things), Publication Office of The European Union, March 2010, ISBN 978-92-79-15088-3. Also available online at http://bookshop.europa.eu/en/vision-and-challenges-for-realising-the-internet-of-things-pbKK3110323/

[b-EC-PRIVACY] *Privacy and Data Protection Impact Assessment Framework for RFID Applications*, 2011. Available online at http://cordis.europa.eu/fp7/ict/enet/documents/rfid-pia-framework-final.pdf

[b-SPRINGER-TRON] *TRON Project 1987 Open Architecture Computer Systems*, Proceedings of the Third TRON Project Symposium, Springer Verlag, 1987, ISBN 978-4431700272.



Internet of Things (IoT)

Everythings will be connected

Transportation Management

Home and Building Automation

Manufacturing Management

Large Scale Development

Y.4104/F.744

Service description and requirements for ubiquitous sensor network middleware

Service description and requirements for ubiquitous sensor network middleware

Summary

The purpose of Recommendation ITU-T F.744 is to describe ubiquitous sensor network (USN) services and requirements for USN middleware. To provide various USN services easily and effectively, it is desirable to define an intermediate entity such as USN middleware for providing functions commonly required by various USN services. This Recommendation covers USN service description, USN middleware description, use cases of USN services using USN middleware, the functional model for USN middleware and requirements for USN middleware.

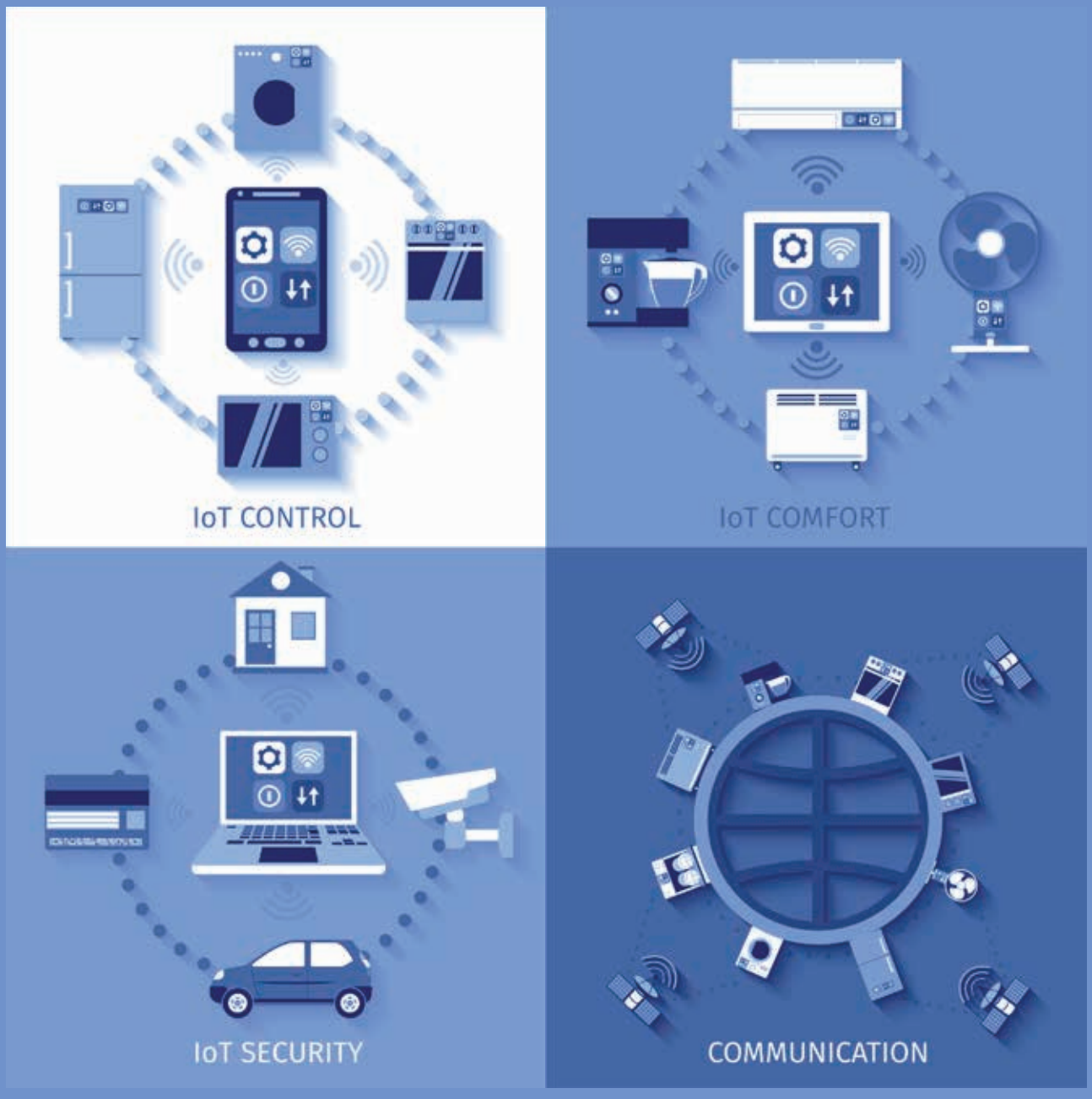
History

Edition	Recommendation	Approval	Study Group
1.0	ITU-T F.744	2009-12-14	16

Keywords

Functional model, requirement, sensor network, USN middleware, USN services.

Table of Contents


IoT CONTROL
IoT COMFORT
IoT SECURITY
COMMUNICATION

Recommendation ITU-T Y.4104/F.744

Service description and requirements for ubiquitous sensor network middleware

1 Scope

This Recommendation describes USN services and requirements for ubiquitous sensor network (USN) middleware. This Recommendation covers:

– description of the USN services;

– description of the USN middleware;

– use cases of USN services that use USN middleware;

– functional model of USN middleware;

– requirements for USN middleware to support functions commonly required by USN services.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T Y.2201] Recommendation ITU-T Y.2201 (2009), *Requirements and capabilities for ITU-T NGN*.

[ITU-T Y.2221] Recommendation ITU-T Y.2221 (2010), *Requirements for support of ubiquitous sensor network (USN) applications and services in the NGN environment*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 context awareness [ITU-T Y.2201]: Context awareness is a capability to determine or influence a next action in telecommunication or process by referring to the status of relevant entities, which form a coherent environment as a context.

NOTE – This Recommendation also uses context-aware with same meaning of context awareness.

3.1.2 sensor [ITU-T Y.2221]: An electronic device that senses a physical condition or chemical compound and delivers an electronic signal proportional to the observed characteristic.

3.1.3 sensor network [ITU-T Y.2221]: A network comprised of inter-connected sensor nodes exchanging sensed data by wired or wireless communication.

3.1.4 sensor node [ITU-T Y.2221]: A device consisting of sensor(s) and optional actuator(s) with capabilities of sensed data processing and networking.

3.1.5 ubiquitous sensor network (USN) [ITU-T Y.2221]: A conceptual network built over existing physical networks which make use of sensed data and provide knowledge services to anyone, anywhere and at anytime, and where the information is generated by using context awareness.

3.1.6 USN middleware [ITU-T Y.2221]: A set of logical functions to support USN applications and services.

NOTE – The functionalities of USN middleware include sensor network management and connectivity, event processing, sensor data mining, etc.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 open application interface: An interface used by USN applications to access USN middleware.

3.2.2 processed data: Data that are processed from raw sensed data by sensor network or USN middleware.

3.2.3 sensed data: Data sensed by a sensor that is attached to a specific sensor node.

3.2.4 sensor network common interface: An interface used between USN middleware and a sensor network/radio frequency identification (RFID) reader.

3.2.5 sensor network metadata: Information about a sensor network, such as description of the sensor network, sensor node identifier, supported sensor type, the number of attached sensors for each sensor node, and the number of sensor nodes connected to the specific sensor network, etc.

3.2.6 sensor network metadata directory service: A directory service that provides sensor network metadata.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

ECG Electrocardiogram

RFID Radio Frequency Identification

USN Ubiquitous Sensor Network

WSN Wireless Sensor Network

5 Conventions

None.

6 Description of USN services, USN middleware and use cases

6.1 Description of USN services

USN is a conceptual network built over existing physical networks that make use of sensed data and provide knowledge services to anyone, anywhere and at anytime, and where the information is generated by using context awareness. USN utilizes wireline sensor networks and/or wireless sensor networks (WSNs). See [ITU-T Y.2221].

USN applications and services can be used in many civilian application areas such as, industrial automation, home automation, agricultural monitoring, healthcare, environment, pollution and disaster surveillance; in homeland security, military field, etc., see [ITU-T Y.2221].

A USN service is a type of service that uses various sensors and/or actuators. In the USN services framework, communications take place between USN applications and sensor networks directly or via some intermediate entity.

Some USN applications and services use basic data processing to obtain the necessary data and others may use advanced data processing such as data mining, context-aware processing, and event processing. In addition, authentication of sensor network and confidentiality of sensed data are very important to protect the USN services from fraudulent data.

The functions of various USN applications and services can be summarized as follows:

– finding appropriate sensor networks to obtain sensed data;
– requesting raw sensed data and/or processed data;
– processing received sensed data;
– activating actuators;
– monitoring sensor network status;
– controlling sensor networks;
– authenticating sensor networks;
– providing appropriate services to users.

These functions are commonly required by many types of USN applications and services. Concerning complexity, scalability and cost-effectiveness, it would be beneficial to support functions by a separate entity rather than by each USN application and service.

6.2 Description of USN middleware

USN middleware is an intermediate entity that provides functions commonly required by different types of USN applications and services. USN middleware receives requests from USN applications and delivers those requests to appropriate sensor networks. Similarly, USN middleware receives sensed data or processed data from sensor networks and delivers them to appropriate USN applications. USN middleware can provide information processing functions such as query processing, context-aware processing, event processing, sensor network monitoring and so on.

6.3 Use cases of USN services

USN services use only sensor nodes or both sensor nodes and RFID readers. In some cases, USN services can activate actuators after processing the sensed data. Some other USN services monitor and/or control sensor networks.

USN services can be categorized into three groups, based on the above observations:

– using only sensed data including RFID tag data (e.g., healthcare applications);
– activating one or more actuators, based on the sensed data, including RFID tag data (e.g., cold chain management applications);
– monitoring and/or controlling sensor networks, including RFID readers (e.g., sensor network monitoring applications).

Use cases in this Recommendation show how USN services and USN middleware work together.

6.3.1 Healthcare applications

A healthcare application continuously monitors the location and the health status of the persons within the range of a sensor network in buildings, in order to handle possible emergencies. See Figure 1. Every resident wears a sensor node on his/her wrist, which looks like a wristwatch. The sensor node senses body temperature, pulse, momentum, and electrocardiogram (ECG) of the resident and then periodically transmits the sensed data to the USN application. A healthcare application displays the current location and health condition of the resident based on the sensed data.

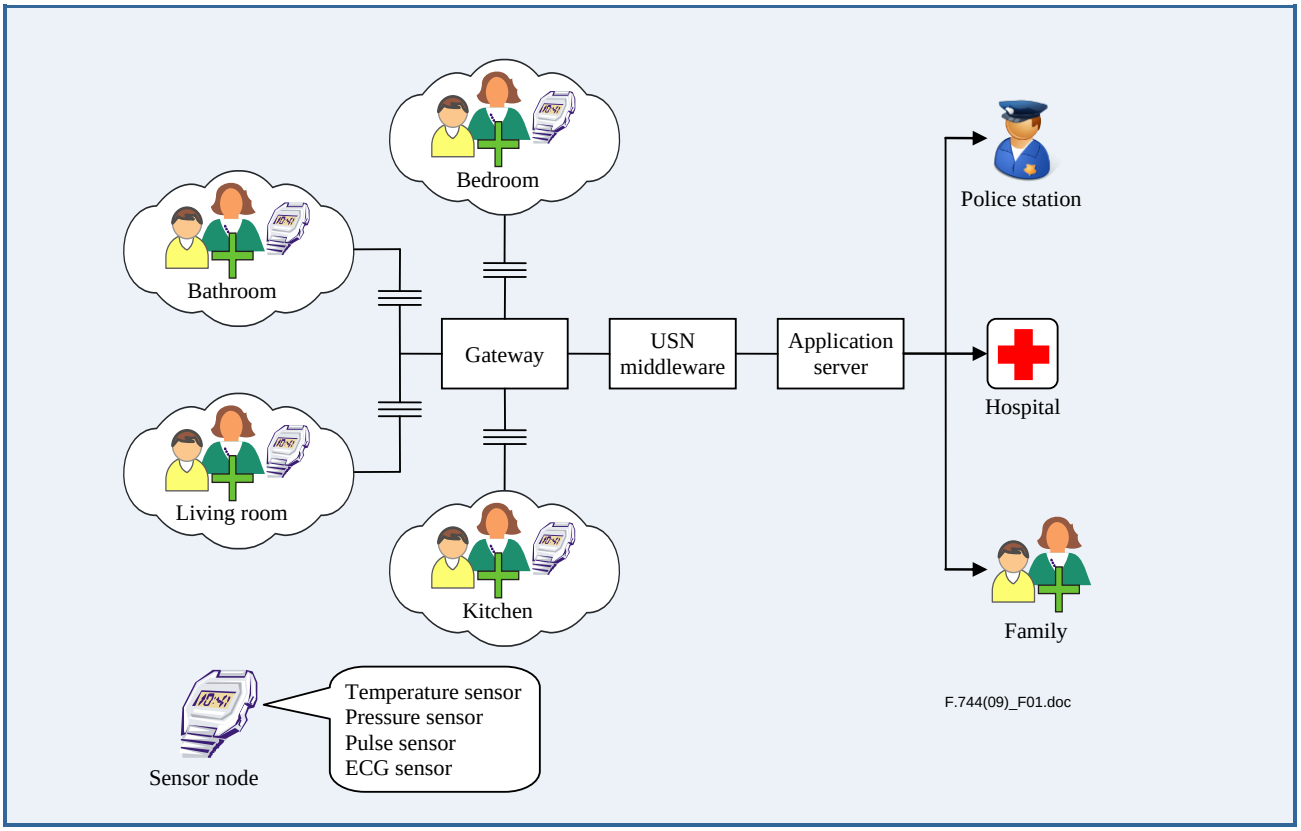


Figure 1 – Use case of a healthcare service

A healthcare application needs residents' medical histories and medical background to provide context-aware service for each resident. Based on the medical background, context-aware rules can be created. Context-aware rules take sensed data and residents' medical histories as inputs and issue emergency notifications when certain emergency conditions are met. Emergency notifications are delivered to the related authorities such as a hospital, a police station and the relatives or family to handle the situation appropriately. The steps are as follows:

Step 1: Managers or operators of a healthcare application should generate appropriate rules based on the medical background to process context-aware information. For services tailored to the individual, the rules need to take residents' medical histories as inputs.

Step 2: Each resident wears a sensor node on his/her wrist. After being turned on, each sensor node senses body temperature, pulse, momentum, and ECG and then periodically sends that sensed data to USN middleware.

Step 3: When a sensor network tries to connect to USN middleware, the USN middleware authenticates the connecting sensor network to protect itself against deceptive sensor networks.

Step 4: If a healthcare application tries to connect to USN middleware, the USN middleware needs to authenticate the connecting application.

Step 5: A healthcare application can utilize a sensor network metadata directory service to obtain the target sensor network metadata.

Step 6: A healthcare application registers appropriate rules to the USN middleware to obtain emergency notifications based on the rules and sensed data.

Step 7: USN middleware collects sensed data from the appropriate sensor networks, based on the requests of USN applications. USN middleware receives sensed data from sensor networks without any requests, if they periodically send sensed data.

Step 8: USN middleware processes sensed data based on the rules for context awareness and simultaneously provides sensed data to a healthcare application.

Step 9: A healthcare application displays current locations and the medical status of the residents on the screen using processed data and/or raw sensed data from USN middleware. The user can select which of the target residents to monitor in detail, then the healthcare application shows the detailed values on the screen.

Step 10: USN middleware generates an event to notify the application of an emergency if certain abnormal condition is detected. The application then alerts related parties such as a hospital and family.

Step 11: When a healthcare application is about to stop its service, it may request the USN middleware to no longer collect data from the sensor networks.

6.3.2 Cold chain management application

A cold chain management application uses RFID tag data and sensed data to monitor the condition of a delivery system. RFID tags are attached to each palette containing products to identify the objects on the palette. Sensor nodes and RFID readers are installed in delivery vehicles and storage buildings of distribution centres. Sensor nodes sense temperature, and send the data to a cold chain management application to report the current status of the delivery environment. If unusual conditions are detected, then a cold chain management application alerts operators to such unusual conditions. The steps are as follows:

Step 1: A cold chain management application generates appropriate rules based on each product management information to determine and then react to the abnormal conditions.

Step 2: Each sensor node equipped with sensors and RFID readers are attached to the delivery vehicles and storage buildings to sense temperature and to recognize the delivered items.

Step 3: Sensor networks/RFID readers are connected to USN middleware to provide sensed data/RFID tag data. When connected, USN middleware is required to authenticate the connecting sensor networks/RFID readers.

Step 4: When the application requires connection to USN middleware, the USN middleware authenticates the connecting application to protect itself from unauthorized application.

Step 5: A cold chain management application registers the context-aware rules in the USN middleware.

Step 6: A cold chain management application can utilize a sensor network metadata directory service to obtain the target sensor network metadata.

Step 7: A cold chain management application requests sensed data/RFID tag data to USN middleware. USN middleware in turn sends the requests to the target sensor networks and RFID readers to collect data.

Step 8: Sensor networks sense temperature and send the sensed data to USN middleware. At the same time, RFID readers collect RFID tag data and then send them as well.

Step 9: USN middleware collects and integrates the sensed data and RFID tag data to provide pairs of {RFID tag data, sensed data} to a cold chain management application. USN middleware filters redundant duplicate tag data to reduce the redundant RFID tag data. Moreover, sensed data aggregation may be performed.

Step 10: USN middleware tells the sensor networks to lower the temperature, if the rules are specified. Then the refrigerator (actuator within the sensor network) can be activated to decrease the temperature. Simultaneously, the event can be delivered to the application to notify the operators of the abnormal situation.

Step 11: A cold chain management application displays the current status of environment on the screen. RFID tag data may be converted into the product information and linked to other sensed data.

Step 12: A cold chain management application may request USN middleware not to collect data any more when a cold chain management application is about to stop its service.

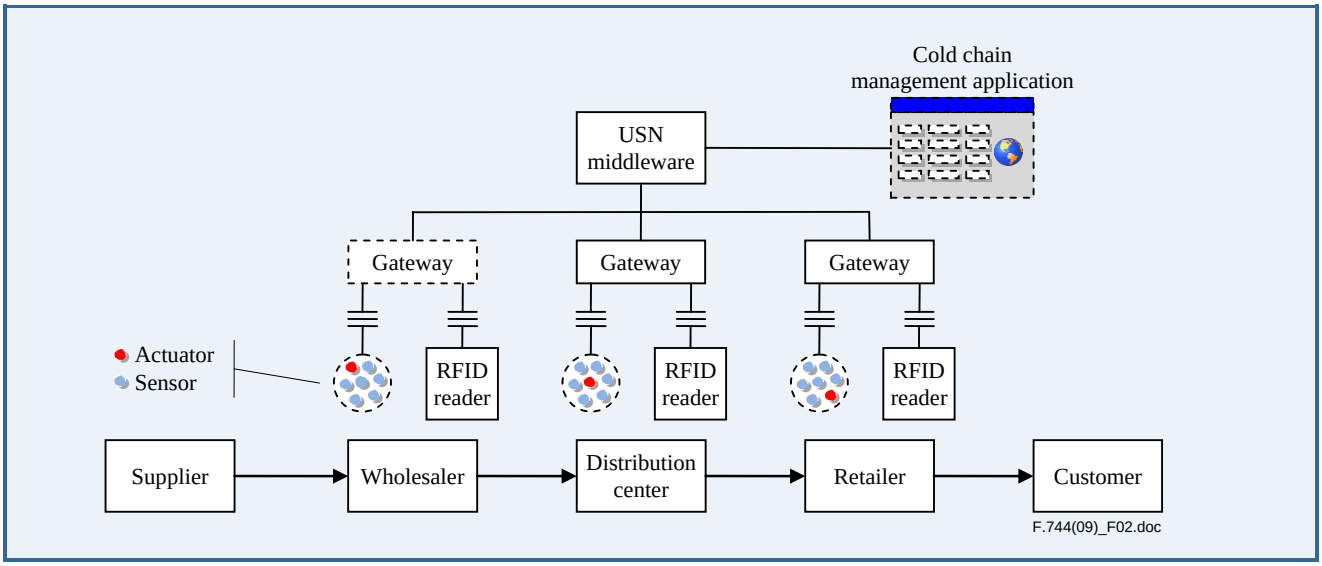


Figure 2 – Use case of a cold chain management service

6.3.3 Sensor network monitoring application

A sensor network monitoring application monitors the various sensor networks. The purpose of a sensor network monitoring application is to check and to control current state of sensor networks. See Figure 3.

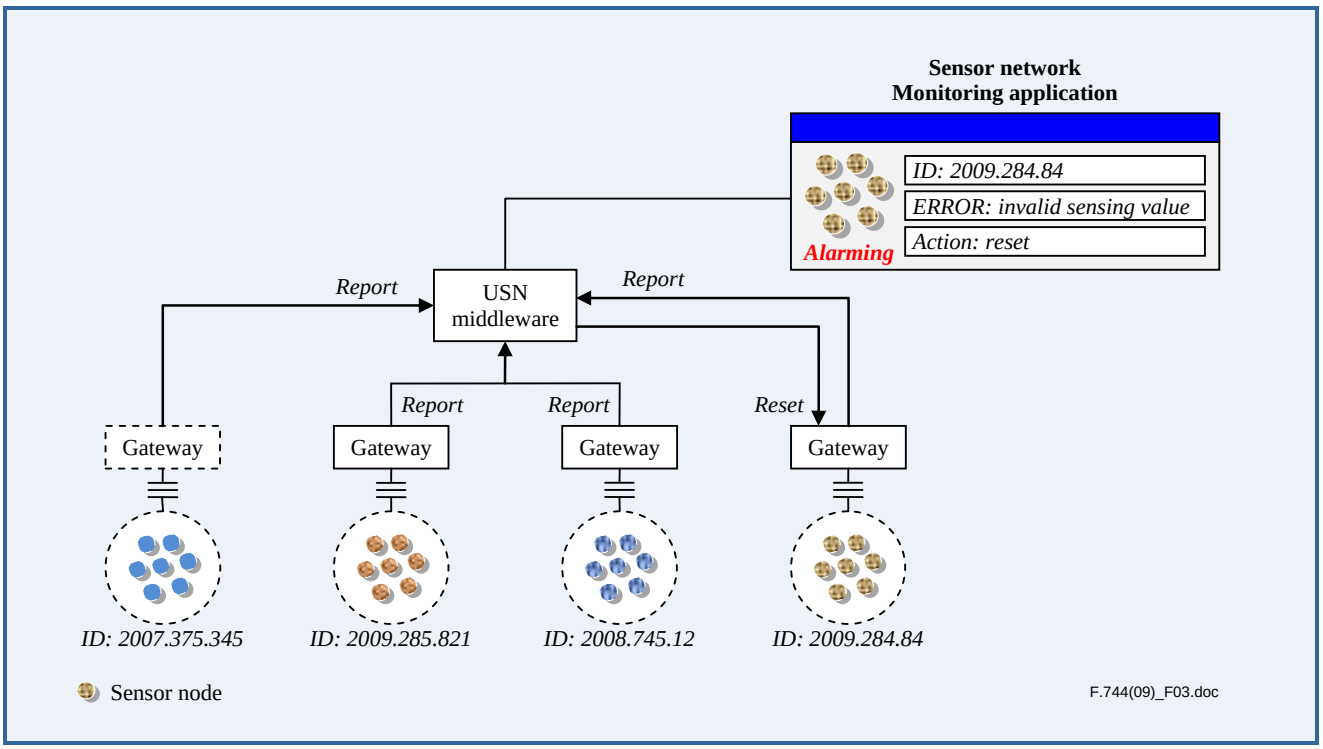


Figure 3 – Use case of a sensor network monitoring service

The sensor network should be associated with the appropriate authorization mechanism because sensor networks are shared by different applications. Therefore, only authorized applications or administrators should be permitted to control the sensor networks. The sensor network monitoring service operates as follows:

Step 1: A sensor network connects to USN middleware. The USN middleware is required to authenticate the connecting sensor network.

Step 2: A sensor network monitoring application connects to the USN middleware to monitor the sensor networks. The USN middleware is required to authenticate the connecting application.

Step 3: A sensor network monitoring application may monitor all the sensor networks that are connected to USN middleware at the same time, or may monitor specific sensor networks. A sensor network monitoring application may refer to the sensor network metadata directory service to determine target sensor networks.

Step 4: A sensor network monitoring application requests USN middleware to collect monitoring information from target sensor networks.

Step 5: USN middleware sends monitoring requests to the sensor networks for collecting monitoring information.

Step 6: Sensor networks send current information to USN middleware for monitoring.

Step 7: USN middleware collects monitoring information from the sensor networks and sends them to a sensor network monitoring application.

Step 8: A sensor network monitoring application displays the current status of target sensor networks on the screen.

Step 9: If a sensor network monitoring application detects abnormal conditions, it may request USN middleware to reset the sensor network.

Step 10: USN middleware sends the reset request to the target sensor network.

Step 11: USN middleware receives the reset result and sends the result to a sensor network monitoring application.

7 Functional model of USN middleware

In the USN service environment where the USN middleware works there are three main elements: the USN application, the USN middleware and the sensor network. In this environment, the USN application utilizes the sensed data and/or activates some actuators, and the sensor network produces sensed data and control actuators. The USN middleware provides functions commonly required by different USN applications and services over the shared sensor networks.

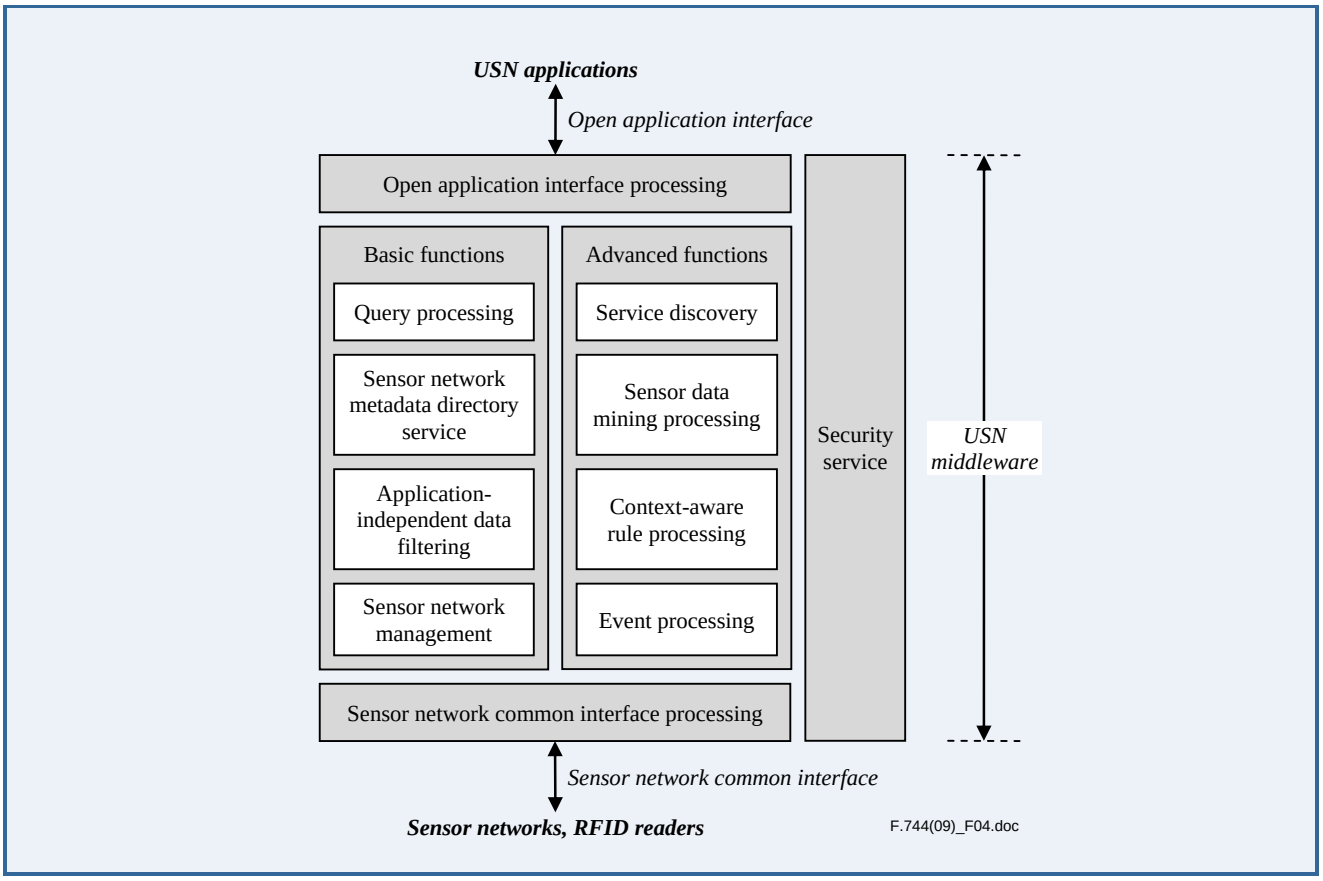


Figure 4 – Functional model of USN middleware

As shown in Figure 4, USN middleware provides functions for USN applications such as data query processing, sensor data mining processing, event processing, sensor network metadata directory service, data filtering, context-aware rule processing and sensor network management. In addition, USN middleware may provide a service discovery function for USN applications. USN applications refer to service discovery to retrieve functions provided by USN middleware and other USN services. In summary, USN middleware may provide five functions: open application interface processing, basic functions, advanced functions, sensor network common interface processing and security service.

7.1 Open application interface processing

An open application interface processing provides the following functions for USN applications and services:

– application interface for applications to access USN middleware;

– authentication, authorization and channel protection functions by cooperating with a security service.

7.2 Basic functions

USN middleware provides basic functions that can be used by most USN applications, as follows:

– Sensor network metadata directory service:
  - registration and retrieval of USN metadata.

– Application-independent data filtering:
  - sensed data validation regarding associated measurement units, data types and value ranges;
  - RFID tag data filtering (duplicate reduction).

– Sensor network management:
  - management of sensor networks including sensor network gateways and RFID readers;
  - software upgrade of sensor node;
  - topology (connectivity) management.

  NOTE – Software upgrade of sensor node and topology management can be provided optionally by sensor network management.

– Query processing:
  - query scheduling for multiple USN applications and multiple sensor networks;
  - query routing to designated sensor nodes;
  - application-dependent RFID tag data filtering;
  - application-dependent sensed data filtering;
  - sensed data aggregation and integration based on an application policy.

7.3 Advanced functions

Advanced functions are the functions that provide service discovery and processed information to applications by using basic functions if necessary. Advanced functions provide the following functions for USN applications and services:

– Sensor data mining processing:
  - detecting outlier, analysing patterns and predicting some events.

– Event processing:
  - generation of events based on raw sensed data or context-aware rule processing;
  - processing of events such as alerting applications and necessary authorities.

– Context-aware rule processing:
  - processing application-dependent context-aware rules on the collected sensed data.

– Service discovery:
  - registration and discovery of USN middleware services;
  - registration and discovery of USN services.

7.4 Sensor network common interface processing

A sensor network common interface processing provides the following functions for USN applications and services:

– sensor network common interface processing;

– authentication and channel protection functions by cooperating with a security service.

7.5 Security service

The security service provides the following functions for protecting USN middleware:

– access control for protecting USN middleware from malicious attacks;

– secure channel for protecting information exchanged between USN middleware and applications/sensor networks;

– secure channel for protecting information exchanged within USN middleware if USN middleware functions are distributed over the networks.

8 Requirements for USN middleware

According to the functional model, the requirements for USN middleware are defined in terms of interfaces, functions and security.

8.1 Interface requirements

– A standardized interface between the USN middleware and heterogeneous sensor networks is required to be provided.

– A standardized interface for applications to access USN middleware is required to be provided.

8.2 Functional requirements

– It is required to support sensor network metadata management.

– It is required to provide sensor network metadata directory service for obtaining sensor network information.

– It is required to provide appropriate data filtering functions such as redundant RFID tag data filtering and sensed data validation for removing unnecessary or erroneous data.

– It is required to provide sensor network monitoring and control function to effectively utilize sensor network.

– It is required to support explicit request-reply query processing mode (pull-mode) and implicit request-reply query processing mode (push-mode). An implicit request-reply query processing means providing sensed data to USN application without any explicit request.

– It is recommended to provide a sensor data mining function.

– It is recommended to provide event generating and processing functions.

– It is recommended to provide a context-aware rule processing function for supporting decision making of USN applications.

– It is recommended to provide service registration and discovery functions including both, USN services and USN middleware services.

8.3 Security requirements

USN middleware is required to provide the appropriate access control mechanisms to protect the sensed data from malicious applications and corrupted sensor networks. Before access to USN middleware, each application is required to register its profile to the USN middleware. Based on the registered application profiles, each application is authenticated and authorized to use USN middleware functions and sensor networks. The application profile may include application description, security information, policy, etc. USN middleware is required to provide security on sensed data and to control data against malicious attacks. In addition, USN middleware is required to authenticate the connecting sensor networks to prevent corruption of sensed data. It is recommended to provide a secure channel to protect the sensed data between the USN middleware and sensor networks.

NOTE – Detailed security requirements for USN middleware are out of scope of this Recommendation.

Bibliography

[b-ITU-T F.700] Recommendation ITU-T F.700 (2000), *Framework Recommendation for multimedia services.*

[b-ITU-T F.701] Recommendation ITU-T F.701 (2000), *Guideline Recommendation for identifying multimedia service requirements.*

[b-ITU-T F.741] Recommendation ITU-T F.741 (2005), *Service description and requirements for audiovisual on-demand services.*

[b-ITU-T F.742] Recommendation ITU-T F.742 (2005), *Service description and requirements for distance learning services.*

Y.4105/Y.2221

Requirements for support of ubiquitous sensor network (USN) applications and services in the NGN environment

Requirements for support of ubiquitous sensor network (USN) applications and services in the NGN environment

Summary

Recommendation ITU-T Y.2221 provides a description and general characteristics of ubiquitous sensor network (USN) and USN applications and services. It also analyses the service requirements of USN applications and services, and specifies the extended or new NGN capability requirements based on the service requirements.

History

Edition	Recommendation	Approval	Study Group
1.0	ITU-T Y.2221	2010-01-13	13

Keywords

NGN, sensor networks, ubiquitous sensor network (USN), USN applications and services, wireless sensor networks (WSNs).

Table of Contents

Recommendation ITU-T Y.4105/Y.2221

Requirements for support of ubiquitous sensor network (USN) applications and services in the NGN environment

1 Scope

This Recommendation, based on [ITU-T Y.2201], covers extensions and additions to NGN capabilities in order to support ubiquitous sensor network (USN) applications and services [b-ITU-T Y.Sup.7] in the NGN environment.

The scope of this Recommendation includes:

- Description and general characteristics of USN and USN applications and services;
- Service requirements to support USN applications and services;
- Requirements of extended or new NGN capabilities based on the service requirements.

The NGN functional architecture extensions for support of the identified extended or new NGN capabilities are out of scope of this Recommendation.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T Q.1703] Recommendation ITU-T Q.1703 (2004), *Service and network capabilities framework of network aspects for systems beyond IMT-2000*.

[ITU-T Q.1706] Recommendation ITU-T Q.1706/Y.2801 (2006), *Mobility management requirements for NGN*.

[ITU-T Y.2012] Recommendation ITU-T Y.2012 (2006), *Functional requirements and architecture of the NGN release 1*.

[ITU-T Y.2201] Recommendation ITU-T Y.2201 (2009), *Requirements and capabilities for ITU-T NGN*.

[ITU-T Y.2233] Recommendation ITU-T Y.2233 (2008), *Requirements and framework allowing accounting and charging capabilities in NGN*.

[ITU-T Y.2234] Recommendation ITU-T Y.2234 (2008), *Open service environment capabilities for NGN*.

[ITU-T Y.2701] Recommendation ITU-T Y.2701 (2007), *Security requirements for NGN release 1*.

[ITU-T Y.2702] Recommendation ITU-T Y.2702 (2008), *Authentication and authorization requirements for NGN release 1*.

[ITU-T Z.100] Recommendation ITU-T Z.100 (1999), *Specification and Description Language (SDL)*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 context awareness [ITU-T Y.2201]: A capability to determine or influence a next action in telecommunication or process by referring to the status of relevant entities, which form a coherent environment as a context.

3.1.2 network mobility [ITU-T Q.1703]: The ability of a network, where a set of fixed or mobile nodes are networked to each other, to change, as a unit, its point of attachment to the corresponding network upon the network's movement itself.

3.1.3 open service environment capabilities [ITU-T Y.2234]: Capabilities provided by open service environment to enable enhanced and flexible service creation and provisioning based on the use of standards interfaces.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 sensor: An electronic device that senses a physical condition or chemical compound and delivers an electronic signal proportional to the observed characteristic.

3.2.2 sensor network: A network comprised of interconnected sensor nodes exchanging sensed data by wired or wireless communication.

3.2.3 sensor node: A device consisting of sensor(s) and optional actuator(s) with capabilities of sensed data processing and networking.

3.2.4 service: A set of functions and facilities offered to a user by a provider.

3.2.5 service description language: A language for the specification of event-driven systems, in particular telecommunication systems, and an object-oriented formal language intended for the specification of complex, event-driven, real-time, and interactive applications involving many concurrent activities that communicate using discrete signals.

3.2.6 ubiquitous sensor network (USN): A conceptual network built over existing physical networks which makes use of sensed data and provides knowledge services to anyone, anywhere and at anytime, and where the information is generated by using context awareness.

3.2.7 USN end-user: An entity that uses the sensed data provided by USN applications and services. This end-user may be a system or a human.

3.2.8 USN gateway: A node which interconnects sensor networks with other networks.

3.2.9 USN middleware: A set of logical functions to support USN applications and services.

NOTE 1 – The functionalities of USN middleware include sensor network management and connectivity, event processing, sensor data mining, etc.

NOTE 2 – In the NGN environment, functions of the USN middleware may be provided by the NGN open service environment (OSE) capabilities [ITU-T Y.2234] and/or by other NGN capabilities. However, some of the USN middleware functions (e.g., those for supporting interface to sensor networks) may not be provided by the NGN OSE capabilities or other NGN capabilities.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

CDMA Code Division Multiple Access

IP Internet Protocol

ITS	Intelligent Transportation System
MAC	Media Access Control
MAN	Metropolitan Area Network
NGN	Next Generation Network
OSE	Open Service Environment
PHY	PHYsical layer
QoS	Quality of Service
USN	Ubiquitous Sensor Network
WCDMA	Wideband CDMA
WiMAX	Worldwide Interoperability for Microwave Access
WMN	Wireless Mesh Network
WPAN	Wireless Personal Area Network
WSN	Wireless Sensor Network

5 Conventions

In this Recommendation:

The keywords "is required to" indicate a requirement which must be strictly followed and from which no deviation is permitted if conformance to this Recommendation is to be claimed.

The keywords "is recommended" indicate a requirement which is recommended but which is not absolutely required. Thus, this requirement need not be present to claim conformance.

The keywords "can optionally" and "may" indicate an optional requirement which is permissible, without implying any sense of being recommended. These terms are not intended to imply that the vendor's implementation must provide the option and the feature can be optionally enabled by the network operator/service provider. Rather, it means the vendor may optionally provide the feature and still claim conformance with the specification.

6 USN description and characteristics

Ubiquitous sensor network (USN), as defined in clause 3.2.6, is a conceptual network built over existing physical networks which makes use of sensed data and provides knowledge services to anyone, anywhere and at anytime, and where the information is generated by using context awareness.

USN utilizes wireline sensor networks and/or wireless sensor networks (WSNs). WSNs are wireless networks consisting of interconnected and spatially distributed autonomous devices using sensors to cooperatively monitor physical or environmental conditions (e.g., temperature, sound, vibration, pressure, motion or pollutants) at different locations. Up to now, WSNs were generally implemented as isolated networks. Simple design of applications and services based on isolated sensor networks is made by the capture and transmission of collected sensed data to designated application systems.

Such isolated, simple applications and services have been evolving over the years through deployment of networks, based upon advanced hardware and software technologies that provide network and service integration, data processing schemes enhanced by business logic and by data mining rules, context-awareness schemes, etc. These technical developments enable the ability to build an intelligent information infrastructure of sensor networks connected to the existing network

infrastructure. This information infrastructure has been called ubiquitous sensor network (USN) opening wide possibilities for applications and services based on sensor networks to various customers such as human consumers, public organizations, enterprises and government.

USN applications and services are created via the integration of sensor network applications and services into the network infrastructure. They are applied to everyday life in an invisible way as everything is virtually linked by pervasive networking between USN end-users (including machines and humans) and sensor networks, relayed through intermediate networking entities such as application servers, middleware entities, access network entities, and USN gateways. USN applications and services can be used in many civilian application areas such as industrial automation, home automation, agricultural monitoring, healthcare, environment, pollution and disaster surveillance, homeland security or military field.

Support of USN applications and services may require some extensions and/or additions to core network architectures in order to cover the functional capability requirements extracted from USN applications and services. USN applications and services are provided through many network assisted functionalities such as context modelling and processing, orchestration of sensed information, data filtering, content management, open interface functions, network and software management, sensor profile management and directory services.

Figure 1 shows an overview of USN with related technical areas including physical sensor networks, NGN, USN middleware and USN applications and services.

NOTE 1 – The details of physical sensor networks and USN middleware are out of scope of this Recommendation.

NOTE 2 – Figure 1 does not represent a functional architecture. The positioning of USN applications and services, USN middleware, NGN and sensor networks in this figure does not correspond to functional layering.

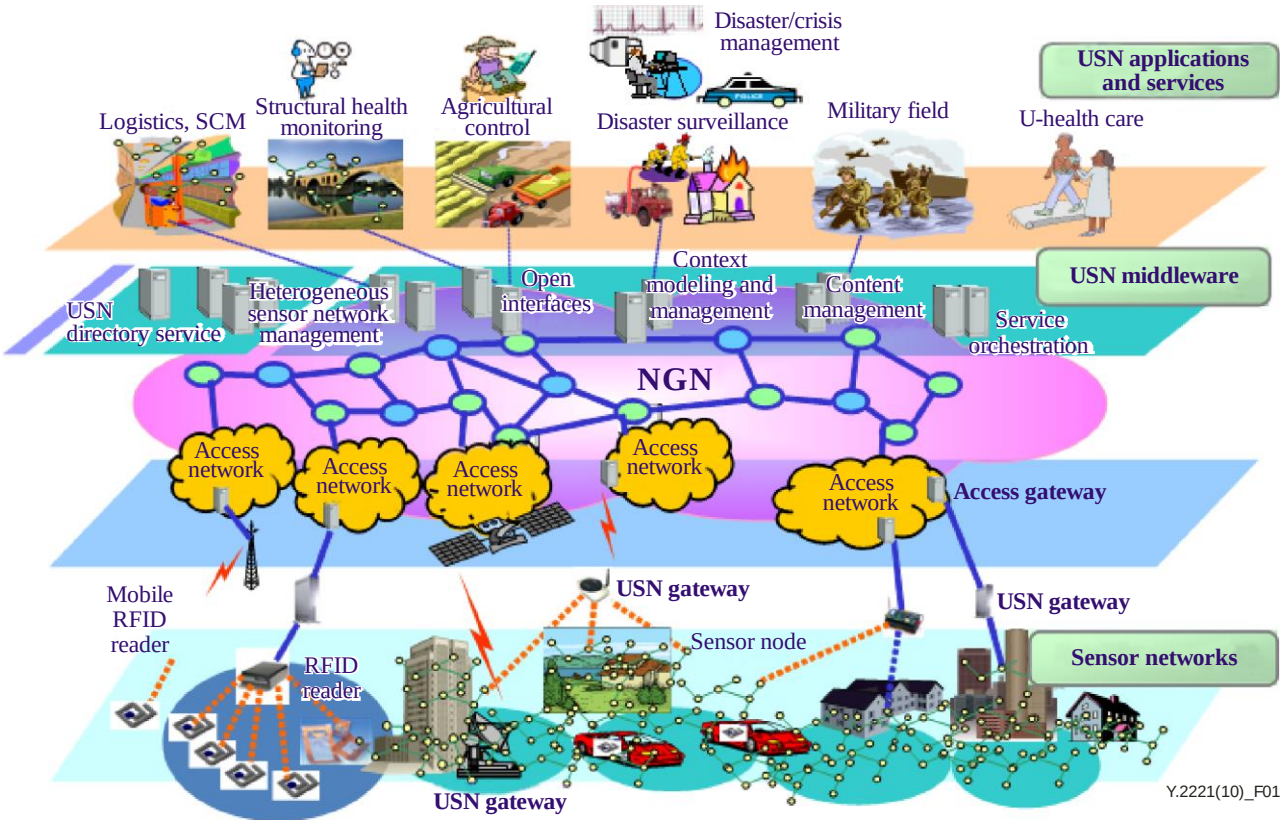


Figure 1 – An overview of USN with related technical areas

For the support of USN applications and services, network and service functions have to be carefully designed to support the unique characteristics of sensor networks and their applications and services, including:

– Limited capabilities of sensor nodes;

NOTE 3 – Sensor nodes generally have limited bandwidth, low processing power, and small memory size such as 32K.

– Limited power that sensor nodes can harvest or store;

– Harsh and dynamic environmental conditions which cause high possibility of node and link failure;

– Mobility support of sensor nodes, sensor networks and services;

NOTE 4 – Due to limited hardware capability, mobility capabilities may not be fully supported by a sensor node or a sensor network.

– Dynamic network topology;

NOTE 5 – Sensor networks often dynamically change the topologies due to the association and de-association of sensor nodes.

– High possibility of communication failures (e.g., due to low bandwidth or link failure);

– Heterogeneity of nodes;

NOTE 6 – A USN application or service may be built using more than one sensor network, where sensor nodes use different PHY/MAC (e.g., [b-IEEE 802.15.4], [b-IEEE 802.15.3]) layers or operate differently in IP-based or non-IP based networks.

– Large scale of deployment;

NOTE 7 – A USN application or service can be deployed on a wide geographical scale to monitor environmental conditions, for example of a river or a seashore.

These characteristics impact many technical areas of USN applications and services in NGN environment, as described in clause 7.

7 Service requirements of USN applications and services

The following are service requirements of USN applications and services which impact the NGN capabilities. These requirements are used to fetch the required extensions to the set of NGN capabilities.

NOTE – Appendix II provides requirements which do not directly impact the NGN capabilities. They are provided for informative purposes.

7.1 Sensor network management

IP-based sensor networks and non-IP-based sensor networks using various types of wired and/or wireless connection can coexist in USN applications and services. Therefore, it is required to manage diverse types of sensor networks. Non-IP-based sensor networks are often managed through their gateway. IP-based sensor networks include the case of a single sensor node directly connected to NGN, while sensor networks are often managed as a set.

Configuration and reconfiguration of sensor networks may require different mechanisms than traditional network management, as sensor networks are normally a group of nodes. A sensor network must not lose its connectivity or its functionality despite the loss of a connection to a single node in the network due to link or hardware failure, which has a high probability of occurrence in sensor networks. Configuration and reconfiguration of a sensor network are used to support assurance of connectivity and lifetime management.

Thus, USN applications and services have the following requirements in order to be supported by various types of sensor networks:

1) It is required to manage IP-based sensor networks including the case of a single node directly connected to NGN.

2) It is required to manage non-IP-based sensor networks.

3) It is required to support configuration and reconfiguration of sensor networks for the assurance of connectivity and lifetime management.

7.2 Profile management

7.2.1 Service profile

In USN environments, a sensor network and its sensed data are utilized by several different applications and services, so sensed data are manipulated as different service data according to the different needs of applications and services. User demands also vary application-by-application and service-by-service.

USN service profile is a way to support the various characteristics and demands of sensed data usage. USN service profiles are composed by information sets of USN applications and services and may include service identifier, data types, service provider, and location information. Thus, USN applications and services have the following requirement:

1) It is recommended to use a standard set of USN service profiles to register and discover USN services.

7.2.2 Device profile

In USN applications and services, a device profile consisting of the information of sensor networks and/or sensor nodes can be optionally provided in conjunction with USN service profile. Unlike traditional networks, only a group of sensor nodes provide meaningful data for general USN applications and services, while data from a single node are also meaningful in some other types of USN applications and services. As there are various types of sensors, sensor nodes and sensor networks, device profiles would help to manage the large number of heterogeneous nodes and networks. The information of device profiles may include sensor network identifier, device identifier, device types, capabilities and location. Thus, USN applications and services have the following requirement:

1) It is optional to use device profiles containing sensor network related information.

7.3 Open service environment

7.3.1 Service registration and discovery

In order to discover USN applications and services, USN services should be registered beforehand. The association of an identifier of a sensor network and sensed data should be registered to service directories. As USN applications and services are very diverse, efficiency of registration and discovery may be increased by a standard set of service profiles as described in clause 7.2. USN end-users and applications should be able to discover the registered services by specifying one or more attributes.

For some USN applications and services, devices in sensor networks may need to be registered and discovered as well as USN services. If a device owner does not want to allow the device to be accessible by others, the device does not need to be registered or discovered. In order to provide device discovery, devices need to be registered with various attributes. USN end-users and applications may be able to discover the registered devices by specifying one or more attributes.

In addition, a USN service description language is required to be provided to support service registration and discovery.

Thus, USN applications and services have the following requirements on service registration and discovery:

1) It is required to support at least one USN service description language and its associated execution framework.

2) It is recommended to register and discover USN services based on a standard set of USN service profiles.

3) Registration and discovery of sensor network devices may be supported.

4) Context-awareness can be optionally supported in the service discovery for USN applications and services.

7.3.2 Service composition and coordination

It is useful to enable easy service creation by the reuse of existing resources and composition of services. Thus, USN applications and services have the following requirement to be supported on service composition and coordination:

1) It is recommended to support service composition and coordination for the creation of USN applications and services.

7.3.3 Interworking with service creation environments

New USN applications and services can be provided via integration with other services (e.g., integration with messaging service, or integration with other USN services). In order to support integration of USN applications and services with features of other service creation environments, interworking with service creation environments is recommended to be supported. Thus, USN applications and services have the following requirement on interworking with service creation environments:

1) It is recommended to support interworking with other service creation environments for the creation of USN applications and services.

7.4 Quality of service (QoS) support

7.4.1 Differentiated QoS and data prioritization

USN mission-critical applications and services should be carefully managed. QoS may be a key technical issue in some scenarios. For example, emergency notification of a fire incident must be delivered in a timely and reliable way to the appropriate national disaster monitoring systems. As USN applications and services are supported over the existing network infrastructure, the emergency data are often carried over the network infrastructure to provide alarm notification. Thus, USN applications and services have the following requirement:

1) It is recommended to provide differentiated quality of service and data prioritization according to the specific USN service requirements.

7.4.2 Application traffic control

Besides the prioritization of certain types of data, efficient traffic and resource management for the sensed data may increase the QoS of USN applications and services, as in general the application transaction volume due to USN applications and services is usually very high. The following requirements are placed on both infrastructure network and application/service provider's resources:

1) It is required to manage the transaction volume generated by USN applications and services.

2) It is recommended to be able to avoid access concentration to a single resource.

7.5 Connectivity

In IP-based sensor networks, sensor nodes are networked using the IP. Although the underlying wired and/or wireless media access control manages the connectivity, connections between USN end-users and sensor networks are through the IP. In this type of sensor networks, it may be possible that a single sensor node is directly connected to the infrastructure networks without a USN gateway; however, USN gateways are normally used to interconnect sensor networks with infrastructure networks.

In non-IP-based sensor networks, sensor nodes do not have IP addresses, and the connections between USN end-users and sensor networks are through the USN gateways.

The different types of sensor networks have to able to connect to the infrastructure networks, so the following requirement applies:

1) It is required to support connectivity between sensor networks and infrastructure networks, regardless of the sensor network type, i.e., IP-based or non-IP-based and using various types of wired and/or wireless media connections. This includes the case in IP-based sensor networks of a single sensor node directly connected to the infrastructure networks.

7.6 Location-based service support

Location of sensor networks and/or individual sensor nodes needs to be maintained and managed in order to support context awareness with location information for USN applications and services. In addition, service and device discovery can be facilitated by the usage of the location information. Thus, USN applications and services have the following requirements:

1) Location information of sensor networks is recommended to be registered for USN applications and services. Registration can be static or dynamic.
2) Location information of individual sensor node can be optionally registered for USN applications and services when the location information of a single sensor node is useful.
3) Location information is recommended to be trustworthy and so location discovery and management is recommended to be secure.

7.7 Mobility support

The challenge of achieving mobility in USN applications and services depends on the technologies used in the sensor networks. Existing IP mobility technologies can be adapted for IP-based sensor networks. However, to port heavy IP mobile mechanisms into very low-power, low-rate sensor networks pose various challenging issues.

A typical USN application and service scenario illustrating mobility requirements can be found in the healthcare application domain. For instance, medical check-up data of a patient may be monitored via a sensor network. Several sensors may be attached to the patient, resulting in a body area sensor network. The sensors periodically gather the medical check-up data and send them to patient's doctor via a home-gateway when the patient is at home; while moving, the data can be sent via an access gateway in a network-enabled car, bus, train, or subway. Various cases of mobility may occur in such an application scenario.

The mobility scenarios for USN applications and services can be classified into three cases:

– A sensor node moving within a sensor network, namely intra-sensor network mobility;
– A sensor node moving across multiple sensor networks, namely inter-sensor network mobility;
– A sensor network moving across infrastructure networks (e.g., across NGN and non-NGN), namely network mobility.

The first two cases can be managed by sensor network technologies which do not have an impact on the infrastructure networks, unless there is a need for location tracking of a single sensor node. The last case requires support of existing mobility technologies of infrastructure networks. Thus, USN applications and services have the following requirements on mobility:

1) It is required to support network mobility when a sensor network moves across infrastructure networks.

2) Infrastructure networks are required to support intra-sensor network mobility and inter-sensor network mobility when location information of a moving sensor node is required to be traced.

7.8 Security

In general, USN applications and services require strong security, due to very sensitive sensed data. It has to be considered that tiny sensor nodes cannot provide all security features because they have lots of system limitations. Thus, the sensed data carried over infrastructure networks may not have strong encryption or security protection. Thus, USN applications and services have the following security requirements:

1) It is required to support key management schemes for USN applications and services.

2) It is recommended to support scalable key management schemes for USN applications and services operating with sensor networks of large size.

3) It is recommended to provide security for the aggregated data when sensed data from two or more applications and services are integrated in infrastructure networks for the creation of new services.

4) The security approaches for the support of USN applications and services are recommended to be consistent with the general approach for the security in NGN.

5) In addition to data security, the USN communication infrastructure is recommended to provide information transport security for protection against well-known passive and active attacks. Protocols for information transport are required to be resilient to attacks.

6) Depending on the specific USN application security requirements, means for intrusion detection are required.

7.9 Identification, authentication and authorization

Network providers and USN service providers must verify the identification of users to access USN applications and services. There are various issues to be considered, such as protection against unauthorized use of network resources and unauthorized access to information flows and applications, authentication of users which try to access the NGN registration and discovery service for sensed data.

In USN applications and services, data can have different levels of authentication requirements. For example, in military systems, raw sensed data are as important as service data which are derived from raw sensed data by processing and manipulation from service providers or applications, while this may not be the case for other systems (e.g., hospital systems). Thus, USN service providers or NGN network providers should support authentication and authorization to use either raw or manipulated service data based on the service requirements. Thus, USN applications and services have the following requirements:

1) It is required to support identification, authentication and authorization of users to access USN applications and services based on the security level of service data.

2) It is required to support different levels of authentication for different types of data based on the requirements of USN applications and services.

3) The USN end-users can optionally identify and authenticate network providers and USN service providers.

7.10 Privacy

USNs allow for the remote collection of a huge volume of sensed data which in many cases are time-stamped and geo-located. The high volume from one hand and the possibility for remote collection from the other increase the potential damage that can be caused by unauthorized parties. Furthermore, the use of multi-hop based infrastructure may require the use of source, location and time for routing purposes, making thus this sensitive information available to intermediate relaying nodes.

In addition, knowing when and where events within a USN occur may compromise the security of the USN itself as well as the security of USN end-users (e.g., in building/home automation USN applications). For this reason, such information needs to be kept "private", i.e., can only be shared between trusted parties. Furthermore, in cases where the USN infrastructure is shared by different USN applications, there is the need for clearly keeping data "private" to each application (especially in operated USNs where a telecom operator may offer commercial services to business clients with conflicting interests).

Thus, USN applications and services have the following requirements:

1) There should be an option for privacy enhanced multi-hop routing mechanisms (information on originating node id, time and location should not be revealed – at least totally – to intermediate nodes).

2) There should be an operating option to de-correlate sensor activity patterns (revealing sensitive context information) from the ensuing communication traffic patterns.

7.11 Accounting and charging

There may be a number of sensor networks deployed inside a given geographical area. Some of them may be built within a single enterprise domain and some may be directly connected to access networks of a service provider domain. Different accounting and charging requirements might have to be addressed depending on the scenarios of USN applications and services. As an example, there are USN applications and services whose sensed data do not have to be continuously transmitted to the application systems, but it is sufficient if they are transmitted, at least once, within a certain period of time. In these scenarios, the network connections may stay in an idle state for a long time. On the contrary, some other USN applications and services may continuously generate and transmit streaming data. These applications and services may require different accounting and charging policies. Thus, USN applications and services have the following requirement:

1) It is required to support different accounting and charging policies according to different data transaction types of USN applications and services.

8 NGN capability requirements for support of USN applications and services

USN applications and services use NGN capabilities [ITU-T Y.2201] but require some extended and/or new capabilities. The capability requirements in this clause are provided from a high level perspective and are not intended to constitute precise functional requirements for the NGN entities.

8.1 Requirements for extensions or additions to NGN capabilities

Based on the service requirements described in clause 7, this clause specifies the requirements for extensions or additions to NGN capabilities.

8.1.1 Network management

Based on the service requirements in clause 7.1, the following additional NGN management capabilities are placed on NGN:

1) NGN is required to manage IP-based sensor networks including the case of a single node directly connected to the NGN.
2) NGN is required to manage non-IP based sensor networks.
3) NGN is required to support configuration and reconfiguration of sensor networks.

8.1.2 Profile management

[ITU-T Y.2201] provides requirements for user profile and device profile management in NGN. The following are additional requirements for the support of USN applications and services.

8.1.2.1 Service profile

Based on the service requirement in clause 7.2.1, the following requirement is placed on NGN:

1) NGN is recommended to support a standard set of USN service profiles.

8.1.2.2 Device profile

Based on the service requirement in clause 7.2.2, the following requirement is placed on NGN:

1) NGN may support device profiles which contain sensor network-related information sets.

8.1.3 Open service environment

[ITU-T Y.2234] defines the open service environment (OSE) capabilities for NGN. The following are additional requirements for the support of USN applications and services.

8.1.3.1 Service registration and discovery

Based on the service requirements in clause 7.3.1, the following requirements are placed on NGN:

1) NGN open service environment (OSE) is required to support at least one standard USN service description language and its associated execution framework.
2) NGN is recommended to register and discover USN applications and services based on a standard set of USN service profiles.
3) NGN may support registration and discovery of sensor network devices (e.g., actuator, gateway) for USN applications and services.

8.1.3.2 Interworking with service creation environments

Based on the service requirement in clause 7.3.3, the following requirement is placed on NGN:

1) NGN OSE is required to support interworking of USN service creation capabilities with capabilities of other service creation environments, as described in [ITU-T Y.2234].

8.1.4 Quality of service

In addition to NGN QoS capabilities, the following requirements list those needed for the support of USN applications and services.

8.1.4.1 Application traffic control

Based on the transaction and traffic-related requirements in clause 7.4.2, the following additional requirements are placed on NGN:

1) NGN is required to support QoS capabilities to sustain the transaction volume caused by USN applications and services.
2) NGN is recommended to support QoS capabilities preventing from access concentration to a single resource (e.g., USN data repositories).

8.1.5 Privacy

Based on the privacy requirements in clause 7.10, the following additional requirements are placed on NGN:

1) NGN is required to provide protection of privacy-related information on relaying control and data packets of USN applications and services.

2) NGN is required to provide an optional operation to de-correlate activity patterns of sensor node and networks from the ensuing USN communication traffic patterns.

8.2 Requirements supported by existing NGN capabilities

Based on the service requirements in clause 7, this clause specifies requirements supported by existing NGN capabilities.

8.2.1 Open service environment

8.2.1.1 Service composition and coordination

NGN provides service composition and coordination capabilities. The service composition and coordination requirement specified in clause 7.3.2 is supported by the existing capabilities [ITU-T Y.2234].

8.2.2 Quality of service

8.2.2.1 Differentiated quality of service and data prioritization

NGN provides QoS supporting capabilities in terms of differentiated quality of service and data prioritization. The differentiated quality of service and data prioritization requirement specified in clause 7.4.1 is supported by the existing capabilities of NGN [ITU-T Y.2201].

8.2.3 Connectivity

NGN provides connectivity capability. The connectivity requirement specified in clause 7.5 is supported by the existing connectivity capabilities of NGN [ITU-T Y.2201].

8.2.4 Location management

NGN provides location management capability which determines and reports information regarding the location of users and devices within the NGN. The location management requirements specified in clause 7.6 are supported by the existing location management capabilities of NGN [ITU-T Y.2201].

8.2.5 Mobility

NGN provides mobility support for the NGN. The mobility requirements specified in clause 7.7 are supported by the existing capabilities of the NGN Release 1 [ITU-T Q.1706].

8.2.6 Security

NGN provides security capabilities. The service requirements specified in clause 7.8 are supported by the existing security capabilities of NGN [ITU-T Y.2201] and [ITU-T Y.2701].

8.2.7 Identification, authentication and authorization

NGN provides identification, authentication and authorization capabilities. The service requirements specified in clause 7.9 are supported by the existing capabilities of NGN [ITU-T Y.2201] and [ITU-T Y.2702].

8.2.8 Accounting and charging

NGN provides accounting and charging capabilities. The service requirement specified in clause 7.11 is supported by the existing capabilities of NGN [ITU-T Y.2233].

9 Reference diagram of NGN capabilities for support of USN applications and services

The reference diagram of NGN capabilities for the support of USN applications and services is shown in Figure 2, based on the service requirements of USN applications and services described in clause 7, and the NGN capability requirements for the support of USN applications and services described in clause 8. The functional capabilities in the figure show extended or new NGN capabilities as well as existing NGN capabilities to support USN applications and services. The related NGN architecture details are out of scope of this Recommendation.

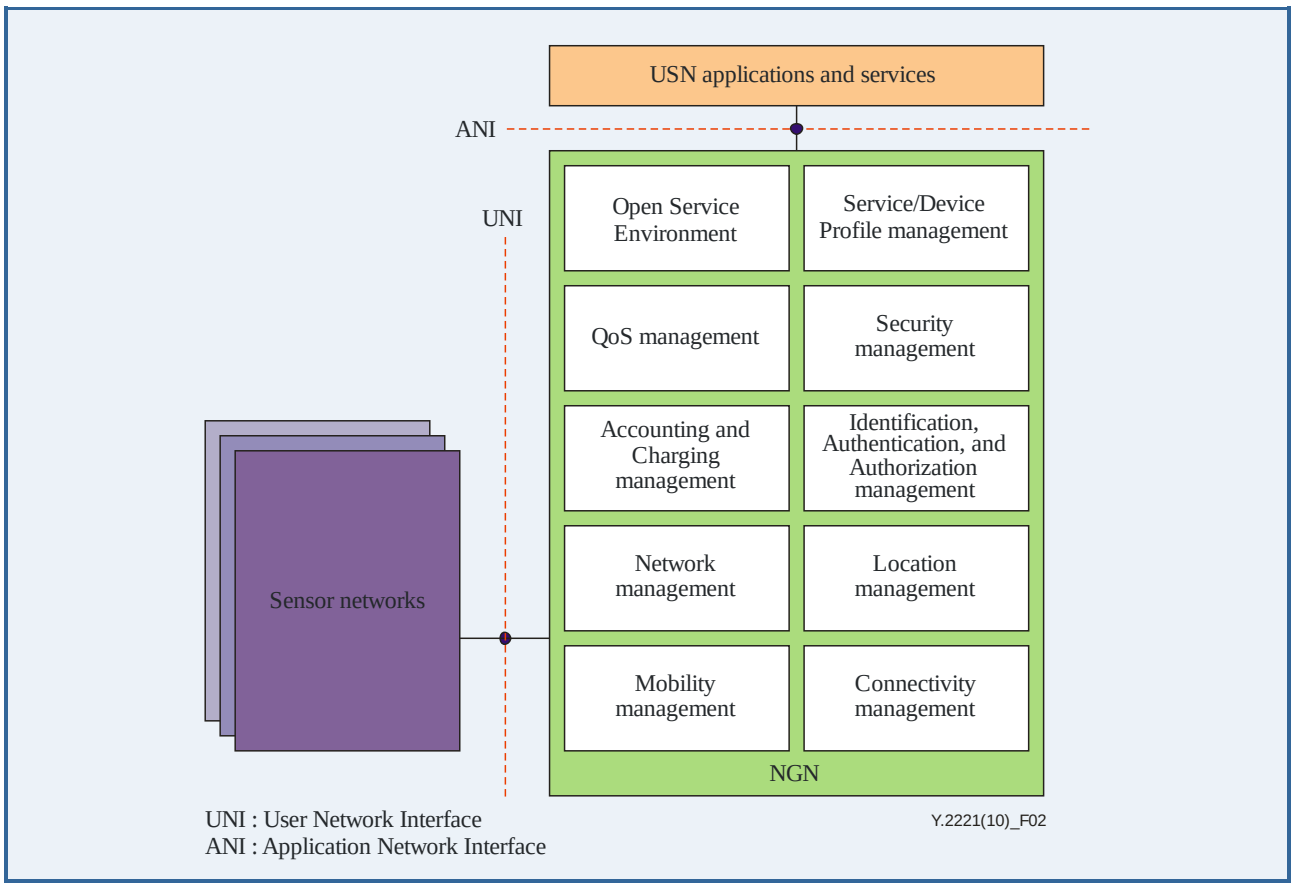


Figure 2 – Reference diagram of NGN capabilities for support of USN applications and services

10 Security considerations

Security is an important issue for USN applications and services. Different USN applications and services have different security requirements. USN applications and services require stringent security for the protection of sensed data. Service requirements on security and identification, authentication, and authority are described in clauses 7.8 and 7.9. NGN capability requirements on security are covered in [ITU-T Y.2201], [ITU-T Y.2701] and [ITU-T Y.2702], as stated in clauses 8.2.6 and 8.2.7.

Appendix I

Use-cases of USN applications and services

(This appendix does not form an integral part of this Recommendation)

Detailed analysis of USN applications and services is out of scope of this Recommendation, but some use-cases are listed in this appendix because they imply market needs and technical issues.

The USN applications and services can be grouped from the perspective of the market they serve, as follows:

– Automation, monitoring and control of manufacturing and industrial applications;
– Home automation;
– Agricultural monitoring;
– Monitoring and management of building and utility;
– Health care and medical research;
– Environment, pollution and disaster surveillance;
– Chemical, biological, radiological and nuclear (CBRN) sensor-based applications;
– Security;
– Military;
– Intelligent transportation management;
– Vehicle communication;
– Smart utility networks (e.g., smart metering for water, electricity, or gas); and
– Urban resource management (e.g., lightning, watering, or parking).

The list is not exhaustive, as USN applications and services are emerging markets and the applications and services constantly evolve.

USN applications and services are numerous, and it is necessary to classify them according to varying business and technical factors. The following three examples show some use-cases of USN applications and services over the NGN.

I.1 Weather information service

One example USN use-case associated with the NGN is that of weather measuring sensors, installed at seashore, river, and local weather measuring points, gathering meteorological data such as temperature changes, humidity changes, and precipitation. Figure I.1 shows this example. The sensor networks and necessary entities for USN applications and services, such as directory servers, can be installed by third-party USN service providers, or directly by the national weather centre.

The sensor nodes, gateways, or separate data gathering entities send the collected information to the servers of the service provider or the weather forecast centre that are connected to the NGN. The sensed data are periodically transferred and/or triggered by meteorological events. The servers of the centre estimate, integrate, and process the information.

The following provide examples of USN applications and services:

1) A fisherman in a seashore area wants to get the on-demand and alarm information of the wave conditions through his cell phone. He will subscribe to a USN service accessible through his cell phone.

2) A tourist who will go mountain hiking for a week wants to get the periodic and alarm information of the weather conditions of the mountain for the week. He or she will subscribe to a temporary weather service in the region.

3) A national disaster centre, which does not own sensor networks in a particular area, will subscribe to an on-demand USN service of a USN service provider, use the information to observe the natural phenomena of the area, and foresee an emergency situation.

USN service providers may manipulate the collected sensed data to suit the request of the USN end-users. The service provider uses NGN functions for support of USN applications and services that perform the data mining and event processing of the sensed data, the USN directory service, etc.

The sensed data are provided to the users in the following manner:

1) A user subscribes to a USN weather information service of a service provider.

2) The sensed data are provided either on demand from the user, or on an event-by-event basis (alarm case).

3) When the user requests the sensed data, the request goes to the USN service provider. When the USN service provider owns the functions for USN applications and services and the corresponding sensor networks, it will process the service data and deliver them to the user. If the USN service provider is a third-party service provider which does not own functions for USN applications and services and the corresponding sensor networks, it will request the necessary information to the service provider who owns the necessary functions, as shown in Figure I.2. The data are delivered via cellular networks, Mobile WiMAX networks, or other access networks.

4) When the service provider detects an emergency case, it will send an alarm notification to the USN end-users without request, as shown in Figure I.3.

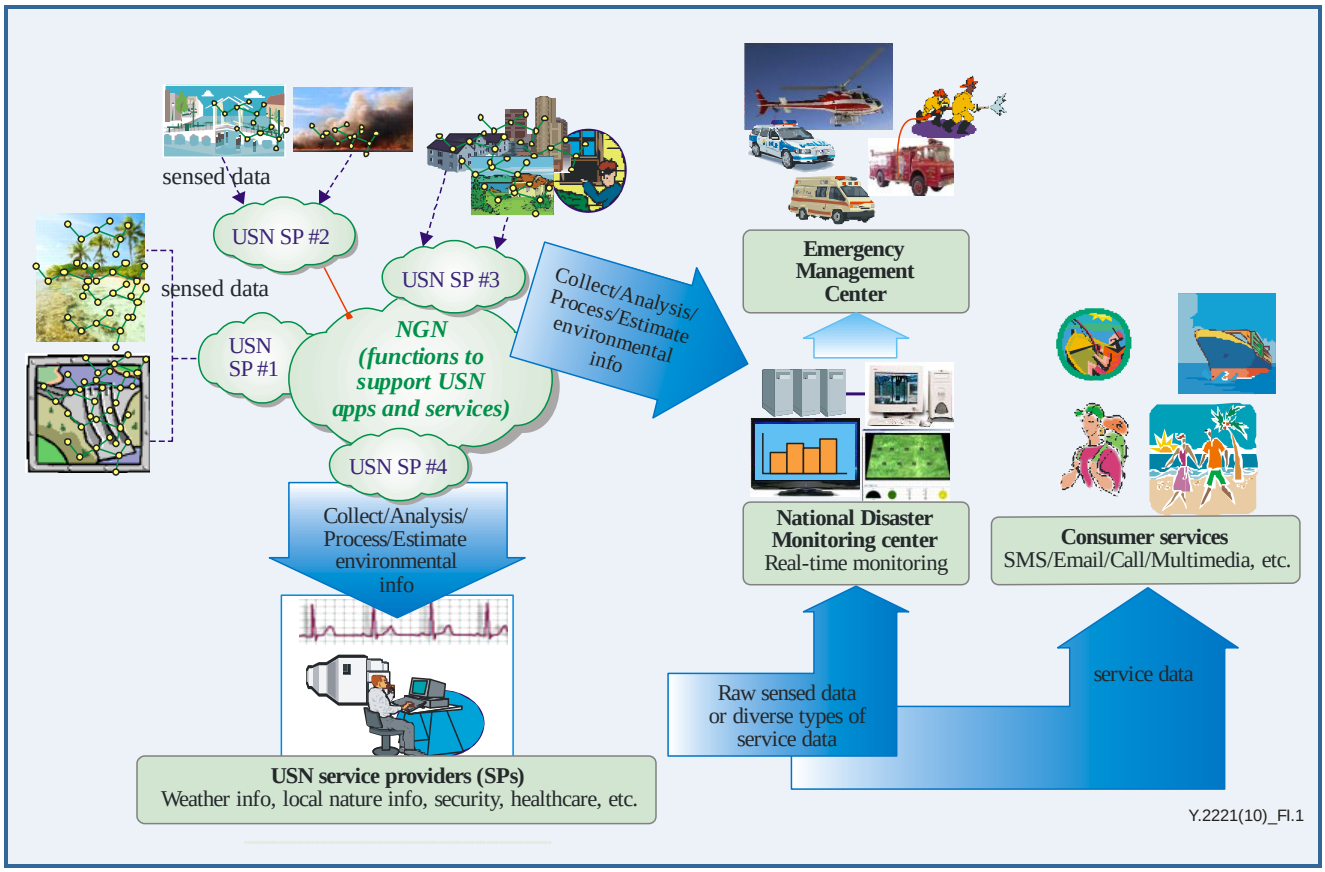


Figure I.1 – USN weather information service

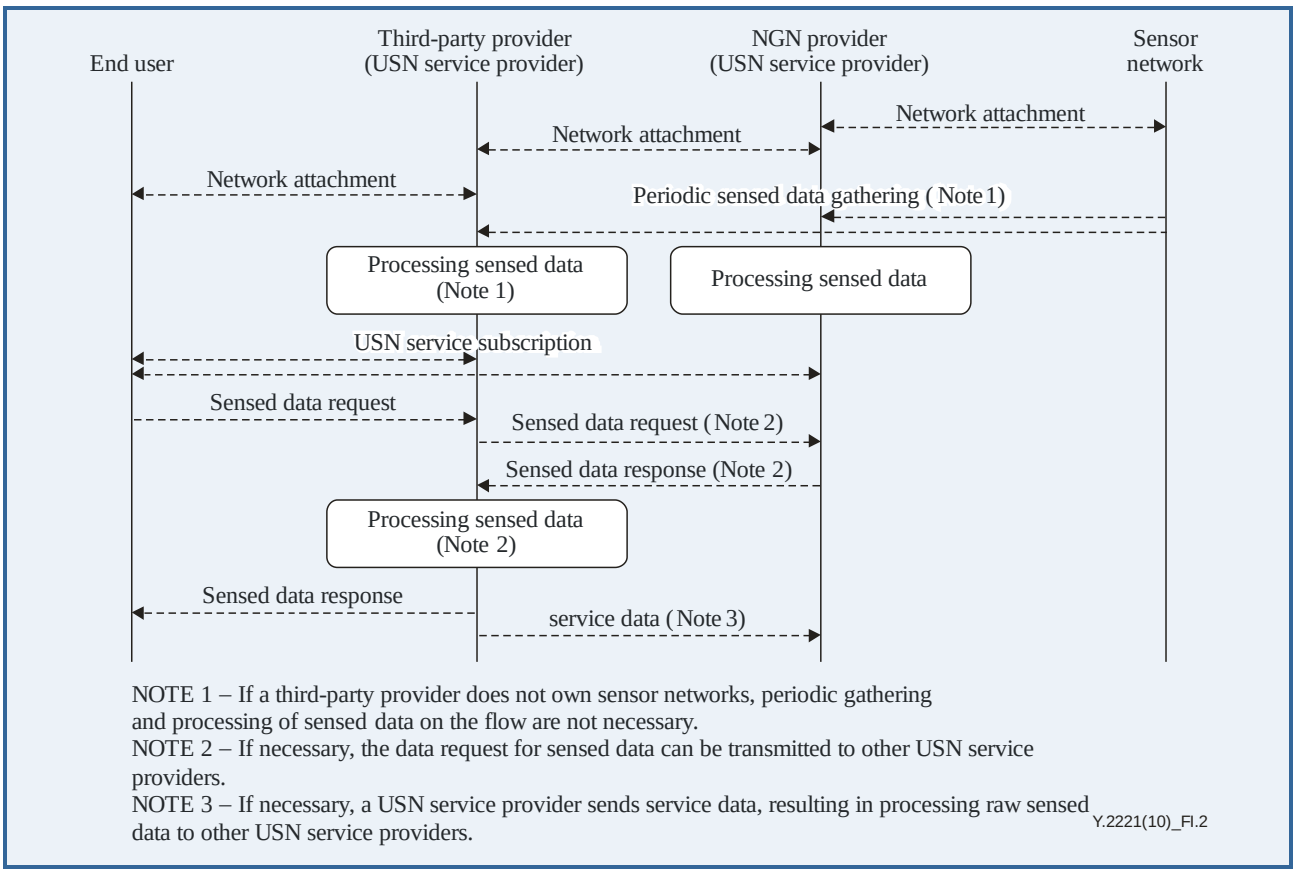


Figure I.2 – Information flow of on-demand USN service

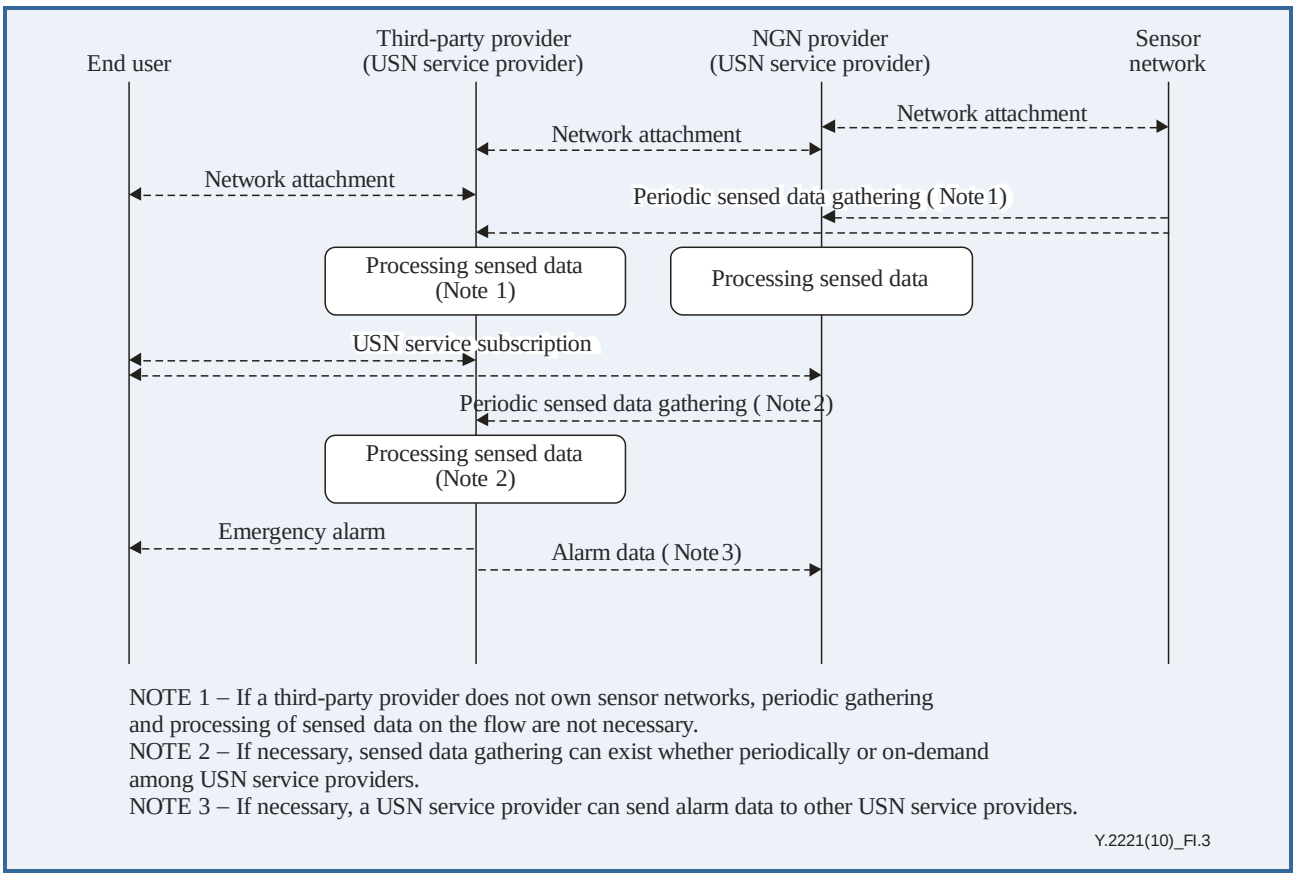


Figure I.3 – Information flow of USN alarm service

I.2 Healthcare service

Another application scenario is that of a patient wearing medical equipment such as watches with attached pulse-measuring sensors, or glasses with attached temperature-measuring sensors, etc. Home network providers may provide a USN-service-enabled home gateway and be USN service providers. The sensors periodically gather medical data and send them to the USN service provider(s).

As Figure I.4 depicts, sensed data may be provided in the following examples:

1) A hospital can establish a business relationship with the USN service provider. The hospital system gets the sensed data either directly from the home gateway or through the service provider.

2) The family of the patient can subscribe to the service to get periodic status information of the patient. The service includes alarm notification in emergency cases.

3) The service will directly call the ambulance when it is necessary.

In an advanced scenario, the sensed data can be transferred even while the patient is moving. The data can be sent via an access gateway in a network-enabled car, bus, train, or subway, which may be connected via different types of access networks, e.g., WLAN, Mobile WiMAX, or cellular networks. The doctor obtains the information in the same way, using available communication networks.

Figures I.2 and I.3 cover the information flow of the USN healthcare services. Sensed data are sent to the service provider, and delivered to different USN end-users as diverse service data resulting from processing the raw sensed data.

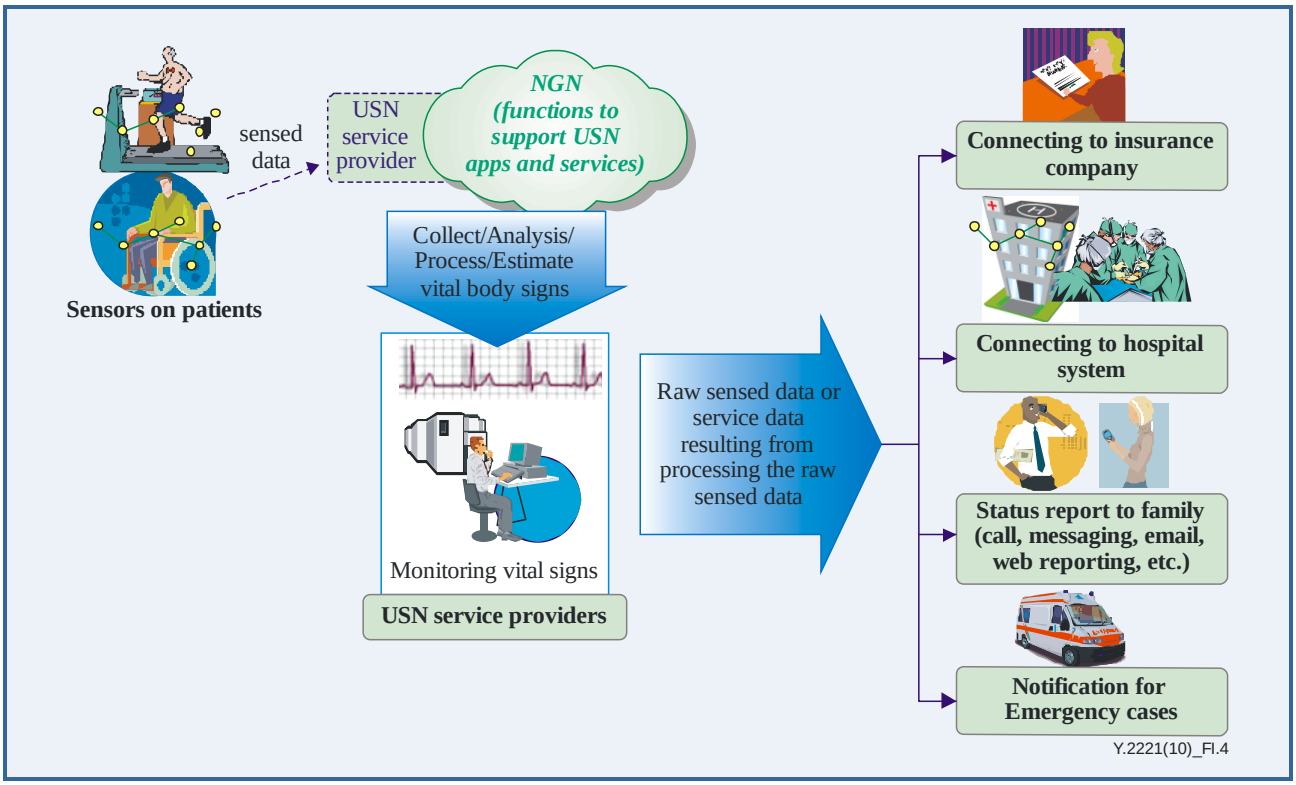


Figure I.4 – USN healthcare service

I.3 Environmental and situational information service using public transportation

Environmental and situational information is a useful source for specific mission-critical services and everyday value-added services. Moreover, information can be more valuable if it is provided regularly to a wide coverage area. Since it is not efficient to deploy static sensor networks for the city-wide target area, it is worthwhile to consider the adoption of a mobile solution (mobile sensor networks).

Environmental sensor nodes, video sensor nodes and location sensor nodes can form sensor networks on public transportation vehicles like buses or cabs. Environmental sensors include those which measure temperature, humidity, particulate air pollution, ozone, illumination, ultraviolet, etc., and video sensors include video cameras that collect video data on street or traffic situation. Environmental and video data can be collected together with location data using location sensor nodes that include GPS information. A gateway can be located in the vehicle where sensor nodes are installed and be connected to the NGN through various types of wireless access networks. As the vehicle moves, environmental and situational information is gathered along the route of the vehicle. Depending on the services, data collection rates can vary. Utilizing the information collected, a variety of services can be provided, as shown below:

– Traffic surveillance services for the operators of the intelligent transportation system (ITS).

– Environmental monitoring service for the administrators of the city environment.

– Traffic accident or crime inspection service.

Web-based environmental and situational information services can be provided for Internet users using mobile handheld devices, IPTV terminals or PCs. People who live or work near the route of the vehicle may be mostly interested in those services.

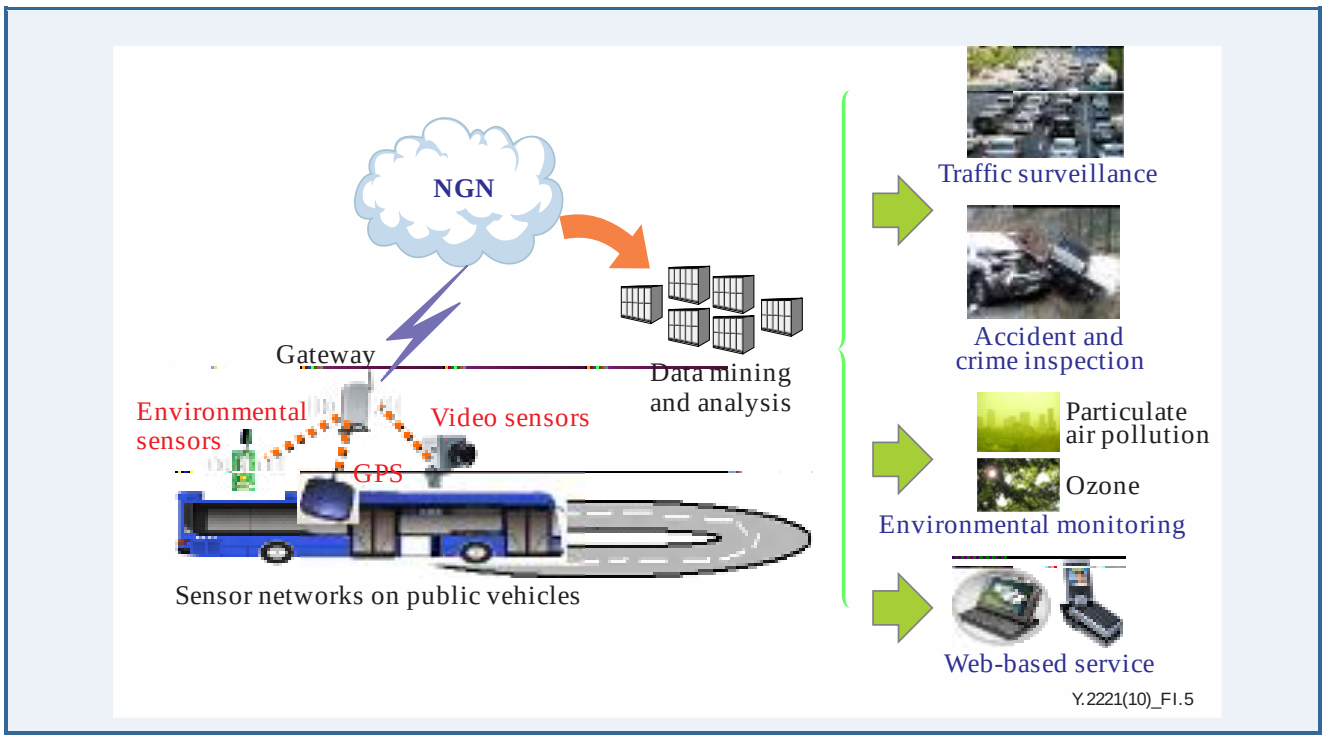


Figure I.5 – Mobile USN service of environmental and situational information monitoring

The above services can be provided in both passive and proactive ways. In a proactive service, the processing of the sensed data and recognition of some critical events is done by data mining and analysis systems that can notify the relevant USN end-users when an emergency situation happens. A passive service is just used by USN end-users for the monitoring of environmental and situational information: in this service, USN end-users detect critical events by themselves.

The following technical challenges need to be tackled:

– Sensor nodes should sense environmental data when the vehicles networked by sensor nodes are moving fast. Thus, the accuracy of the sensed data should consider the speed of the vehicle. Technologies for sensed data diagnostics can be adopted.

– Strong video compression technologies are highly recommended because the video data volume can be huge due to continuous monitoring data.

– Networking between the sensor networks on the vehicle and the NGN should be reliable although the sensor networks move fast. Mobility support for the sensor networks must be provided.

Appendix II

Capability requirements for support of USN applications and services not directly affecting the NGN

(This appendix does not form an integral part of this Recommendation)

The following requirements do not directly affect functional capabilities of the NGN but USN applications and services. The following are on sensor network areas, not on access or core networks.

II.1 Power conservation (sensors node)

In sensor networks, some devices are powered by mains power lines, but most are battery-operated (e.g., AA battery or IEC designated LR6 (alkaline), R6 (carbon-zinc), KR157/51 (nickel-cadmium), HR6 (nickel-metal-hydride), or FR6 (lithium-iron-disulfide)). In addition, sensor nodes have the characteristics of small devices, limited memory sizes, low processors, low bandwidth, high loss rates, etc. These characteristics lead to the following requirements:

1) It is required to provide small code size of network and transport layer protocols, application protocols and data.

2) Low protocol state is required to be supported; low memory usage, low protocol overhead, etc.

3) It is highly recommended to provide robust and energy efficient protocols to handle dynamic loss from battery deficit or mainly sleeping nodes.

II.2 Network formation: auto-configuration and self-healing (sensor networks)

An important trait of sensor devices is their unreliability due to their limited system capabilities. It is predicted that user interaction and maintenance become impractical in such conditions, and auto-configuration and self-healing capabilities are useful to provide robustness of sensor networks. Thus, sensor networks have the following requirement:

1) Auto-configuration and self-healing are recommended to be supported for dynamically adaptive topologies.

II.3 Addressing mechanisms

Some USN applications and services such as nature monitoring system, sensor networks will be comprised of significantly higher numbers of devices than counted in current networks. In addition, USN applications and services have point-to-multipoint (P2MP) or MP to P traffic patterns, more than point-to-point (P2P) traffic. To support USN applications and services, the following addressing requirements are placed on sensor networks:

1) Address mechanisms are recommended to support high scalability. IP addressing can be used as a global address mechanism for IP-based sensor networks, while local address mechanisms can be used within the stub networks in non-IP based sensor networks. When no global addresses are used in the sensor networks, it should be guaranteed that a local gateway can provide the connectivity to the sensor networks.

2) Efficient P2MP or MP2P communication is required to be supported. It can be provided either with a special address for multipoint or by efficient transport mechanisms.

II.4 ID design

As sensor networks are generally deployed as a stub network in many services, IDs for sensor nodes in the network may be allocated by a coordinator in the sensor network considering the applications and service types. In other words, they could have a global address such as an IP address, but have a special naming mechanism for the services. USN applications and services have the following ID design requirements:

1) In some applications and services, a data-aware ID or naming mechanism is recommended. (e.g., temp_etri_x36y30, wind_etri_x36y30). Application functions should support to decode the ID with local or global addresses of the sensor nodes.

2) In some applications and services, a geographical ID or a naming mechanism is recommended. (e.g., temp_etri_x36y30, wind_etri_x36y30). Application functions should support to decode the ID with local or global addresses of the sensor nodes.

II.5 Sensor nodes mobility support

Sensor networks are likely to have a certain degree of mobility. Due to the low performance characteristics of sensor nodes, the following requirement is placed on sensor networks:

1) Inter- and intra-mobility are required to be provided without extra protocol overhead in sensor nodes.

II.6 Secure control messages

Security threats within sensor networks may be different from existing threat models in other networks, e.g., bootstrapping and neighbor discovery may be susceptible to threats. The following requirements are placed on sensor networks:

1) Control messages within sensor networks are required to be secure, in the way that security mechanism should not be overhead of low-powered sensor networks.

2) Design for power conservation should not compromise security, especially in USN applications with strong security requirements.

II.7 Lightweight routing

As sensor networks have special requirements on energy saving and data-oriented communication, the following requirements are placed on sensor networks:

1) Energy efficient routing schemes are required to be supported.

   NOTE – Energy efficiency should not be considered in absolute terms (e.g., support of multi-path routing in case of USN application specific security and resilience requirements).

2) It is required to support routing schemes for sensor nodes in sleeping mode most of the time.

3) It can optionally support data-aware routing schemes.

4) It is recommended to support efficient routing schemes for diverse data traffic patterns; MP2P, P2MP, and P2P.

Some USN applications and services are based on large scale sensor networks. To support high scalability, the following requirement is placed on sensor networks:

5) Scalable routing schemes (e.g., with reduced routing state) are recommended to be supported for a large size of sensor networks.

II.8 Connectivity

Sensor networks, regardless of sensor network types, are required to support connectivity to other networks (e.g., NGN or IP network). To support connectivity, the following requirements are placed on sensor networks:

1) IP-based sensor networks can be connected to other IP-based networks through IP routers. Protocol conversion or tunnelling capability is required to be supported when the IP versions of the connected network and the sensor network are different.

2) Non-IP based sensor networks are required to be connected to other networks using gateways that support protocol conversion.

3) Scalability issues are recommended to be taken into account to support large scale sensor networks.

Bibliography

[b-ITU-T Y.2001] Recommendation ITU-T Y.2001 (2004), *General overview of NGN*.

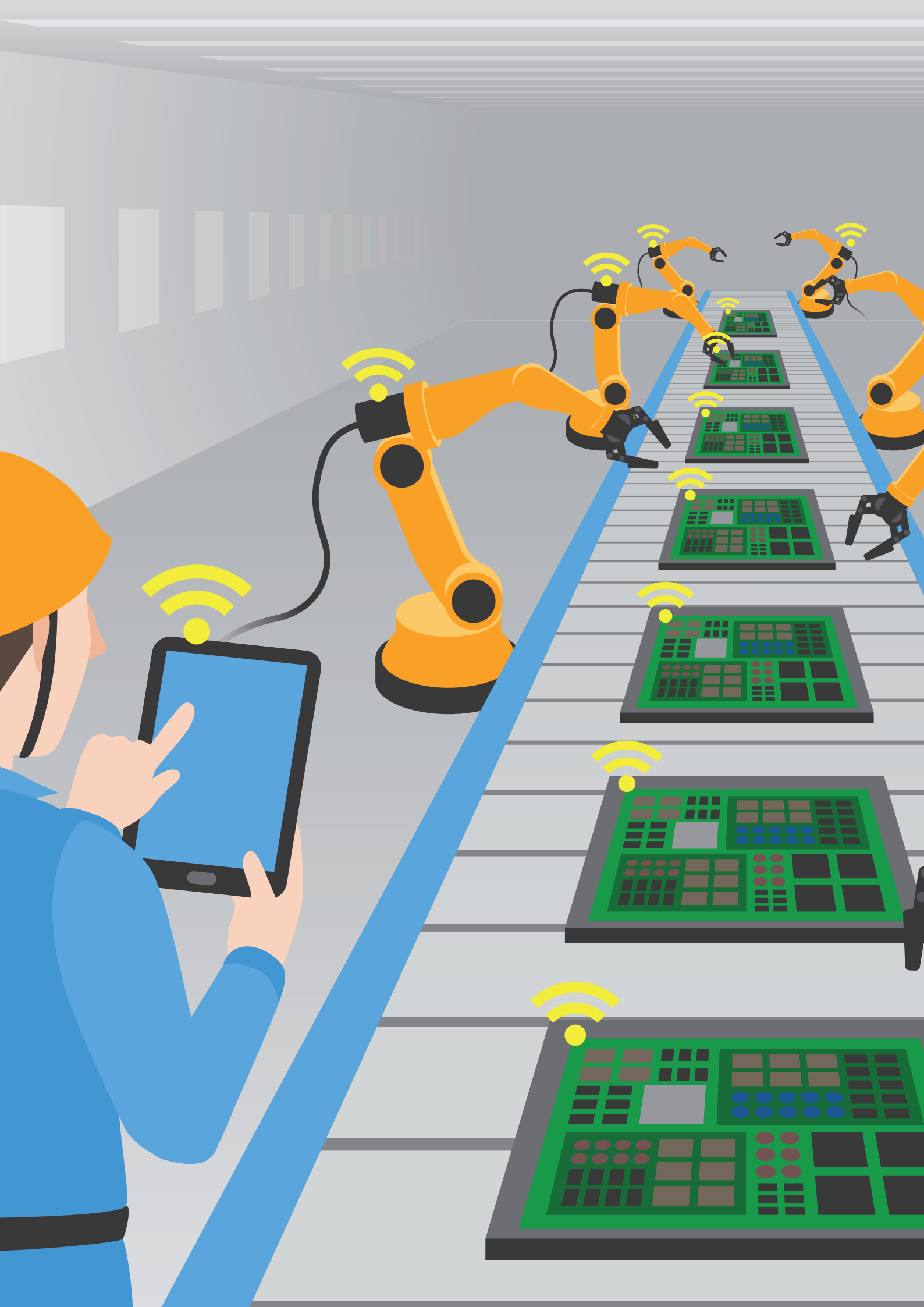
[b-ITU-T Y.2011] Recommendation ITU-T Y.2011 (2004), *General principles and general reference model for Next Generation Networks*.

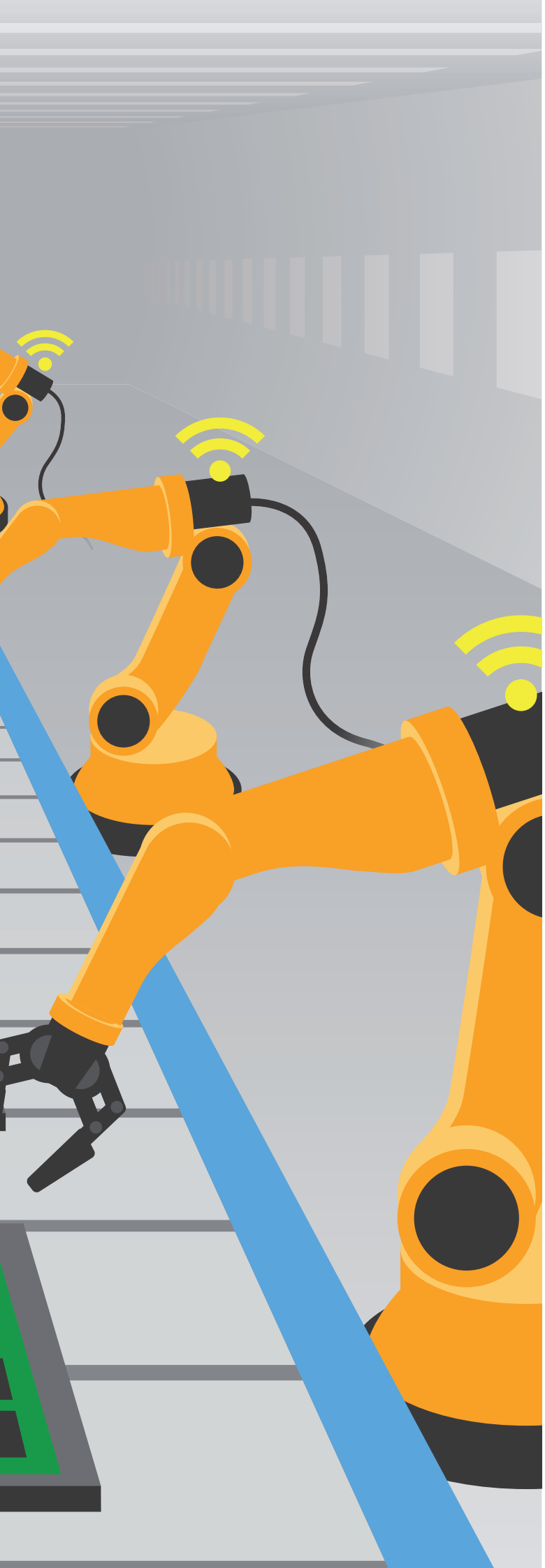
[b-ITU-T Y.Sup.7] ITU-T Y-series Recommendations – Supplement 7 (2008), ITU-T Y.2000-series – *Supplement on NGN release 2 scope*.

[b-IEEE 802.15.3] IEEE 802.15.3 (2003), *Wireless medium access control (MAC) and physical layer (PHY) specifications for high rate wireless personal area networks (WPANs)*.

[b-IEEE 802.15.4] IEEE 802.15.4 (2006), *Wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (WPANs)*.


01010
0 1 0 1 0 1 0 1
01
10100101





Y.4106/F.747.3

Requirements and functional model for a ubiquitous network robot platform that supports ubiquitous sensor network applications and services

Requirements and functional model for a ubiquitous network robot platform that supports ubiquitous sensor network applications and services

Summary

Recommendation ITU-T F.747.3 describes the concept, use cases, requirements and functional model of a ubiquitous network robot platform that supports ubiquitous sensor network (USN) applications and services.

History

Edition	Recommendation	Approval	Study Group
1.0	ITU-T F.747.3	2013-03-16	16

Keywords

Middleware, network robot system, ubiquitous network robot platform, USN.

Table of Contents


SCAN PROCESS
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
18.5%
33%
9%
35
100%

Recommendation ITU-T Y.4106/F.747.3

Requirements and functional model for a ubiquitous network robot platform that supports ubiquitous sensor network applications and services

1 Scope

The objective of this Recommendation is to define a ubiquitous network robot platform, and to identify its requirements and functional model. The use of standard interfaces for the ubiquitous network robot platform will ensure network robot service reusability, portability across several network robot services, and network accessibility and interoperability by the ubiquitous sensor network (USN).

The scope of this Recommendation includes:

– the concept of ubiquitous network robot platform;

– requirements of the ubiquitous network robot platform;

– functional model of the ubiquitous network robot platform.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T F.744] Recommendation ITU-T F.744 (2009), *Service description and requirements for ubiquitous sensor network middleware*.

[ITU-T Y.2221] Recommendation ITU-T Y.2221 (2010), *Requirements for support of ubiquitous sensor network (USN) applications and services in the NGN environment*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 sensor [ITU-T Y.2221]: An electronic device that senses a physical condition or chemical compound and delivers an electronic signal proportional to the observed characteristic.

3.1.2 sensor network [ITU-T Y.2221]: A network comprised of inter-connected sensor nodes exchanging sensed data by wired or wireless communication.

3.1.3 sensor node [ITU-T Y.2221]: A device consisting of sensor(s) and optional actuator(s) with capabilities of sensed data processing and networking.

3.1.4 service [ITU-T Y.2221]: A set of functions and facilities offered to a user by a provider.

3.1.5 ubiquitous sensor network (USN) [ITU-T Y.2221]: A conceptual network built over existing physical networks which makes use of sensed data and provides knowledge services to anyone, anywhere and at any time, and where the information is generated by using context awareness.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 robot: A device with a processing unit often accompanied by sensors and actuators that can work with real-world phenomena and entities. Robots can be roughly classified into the following three types: visible-type robot, virtual-type robot and unconscious-type robot.

3.2.2 ubiquitous network robot platform: Middleware that enables applications to perform services continuously by combining multiple robotic devices effectively across multiple areas.

3.2.3 unconscious-type robot: A type of robot that mainly senses real-world phenomena and processes measurement results into high-level abstractions. An example of this type of robot is a camera equipped with a processing unit used for detecting people. Unconscious-type robots are often embedded in environments such as roads, towns or rooms, or are hidden in clothes or accessories.

3.2.4 virtual-type robot: A type of robot that mainly processes and utilizes information via a network. Smartphones are examples of this type of robot. Virtual-type robots typically interact with people through audio and visual modalities.

3.2.5 visible-type robot: A type of robot that can be seen and which can take one of several forms such as a humanoid, a pet or a stuffed animal. Visible-type robots interact with people in a combination of verbal and nonverbal modalities such as through a conversation augmented by gestures.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

FDML	Field Data Mark-up Language
FE	Functional Entity
GPF	Global Platform
ID	Identifier
LOD	Linked Open Data
LPF	Local Platform
NGN	Next Generation Network
NRS	Network Robot System
PF	Platform
SOA	Service-Oriented Architecture
UNR	Ubiquitous Network Robot
UNR-PF	Ubiquitous Network Robot Platform
USN	Ubiquitous Sensor Network

5 Conventions

None.

6 Overview of UNR-PF in terms of USN applications and services

6.1 General overview of UNR-PF

Demands for assistance robots are quickly emerging with the increase in elderly population. In recent decades, researchers have been focusing on constructing robots that can interactively support daily human activities. As the structure and functionality of robots grow large and complicated, constructing robots now requires more time, cost and effort. As with traditional software engineering, developers have started to seek modularity and reusability of basic functional components, which has led to research and development of common libraries and middleware. Reusable and modular components allow developers to utilize existing functional modules in combination with their own software and to rapidly develop working robots. This modularized development process has accelerated the development of stand-alone robots as well as individual functional components. At the same time, however, variations in both hardware and software have decreased the reusability of robotic applications.

Another approach to enhance capabilities of robots is the concept called network robot system (NRS) or networked robots [b-Sanfeliu2008]. The main goal of NRS was to integrate various types of devices such as robots, sensor networks and smartphones, so that the whole system could act as an integrated system with enhanced capabilities that cannot be attained by a single robot or multiple uniform robots. Since its introduction in 2002, many research projects have been carried out accompanied with real-world field experiments. These have successfully shown that the concept is effective ([b-Jung2007], [b-Nakamura2008], [b-Tezuka2006], [b-Sanfeliu2010], [b-Shiomi2011] and [b-Salvini2011]).

However, customers need robots to support them in a much wider range of daily activities. Robots for individual household tasks such as cleaning floors or folding laundry are not sufficient. Support for the elderly and disabled in a variety of daily activities is in great demand. Such support requires robots to accompany people to many different places (e.g., homes, shopping malls, hospitals) and to assist people for various activities (e.g., checking health, showing routes, carrying luggage) in ways that differ depending on place and the physical demands of an individual (e.g., wheelchair user). At the same time, these sequences of activities shall be well integrated to provide comfortable and seamless support throughout our daily lives.

Existing robotic systems cannot yet provide such continuous support in various aspects of our daily lives. Although they have had success in enriching capabilities of individual robotic applications, a general framework is missing for adapting behaviours of robots and composing them to form an integrated sequence of applications. Similar concepts can be seen in computer systems, such as service oriented architecture (SOA). SOA provides design principles for constructing large-scale information systems and has been adopted in various middleware stacks, especially in web service systems. In combination with the concepts of grid or cloud computing systems, many commercial services with high dependability are available today. However, these concepts are not sufficient for robotic systems.

The major difference between information systems and robotic systems is where the applications reside. Information system applications reside in cyberspace using information terminals such as personal computers or mobile phones. This is especially clear for web-based systems where applications run through web browsers. Based on the information shown on the screen, users perform actions such as business procedures, travel and cooking. Thus, the actual work in the physical world is left to the customers. This is where the robotic systems with sensor networks start. Robotic systems need to consider various factors in real-world environments, including physical abilities and limitations of both customers and devices. Moreover, as stated previously, a variety of different devices need to be integrated to realize continuous robotic support in our daily lives.

As such, one needs to focus on different kinds of "ubiquity", not just for location but also for various applications and ways of providing each service on the basis of customer attributes, as well as inter-application and inter-location continuity. As the complexity of robotic devices is much higher than that of traditional information systems, abstract, common access methods need to be provided for application programmers and service providers to make the development process easy and highly reusable. A new framework is now required that can bridge the gap between systems that realize the above ubiquity and those that improve and extend traditional robot systems which satisfy such demands. Based on such considerations, an extended infrastructure based on NRS is defined in this Recommendation, the ubiquitous network robot platform (UNR-PF) [b-Sato2011b] and [b-Kamei2012]. UNR-PF combines multiple robots in multiple areas for the support of NGN, USN and their middleware that provide access to sensor networks and portable devices. This concept offers the possibility of providing innovative services promptly and at low lost.

6.2 Relationships between USN and UNR-PF

UNR-PF utilizes USN and USN middleware in two ways: a) as an infrastructure with advanced functionalities such as intelligent routing of messages, and b) as a representative of various sensor networks/nodes. USN may provide connectivity not only to sensors, or unconscious-type robots, but also to robots of the visible and virtual types. UNR-PF can be seen as a middleware for USN applications that provides detailed information and control over USN, such as detailed device profiles, common representation of measurements and an abstract command interface. UNR-PF can also be considered as a higher layer router over USN. UNR-PF selects and allocates appropriate devices to applications. This allocation may change dynamically over time, while the applications do not care or notice the actual changes.

Figure 1 shows the relationship between the open USN service platform and UNR-PF defined in this Recommendation.

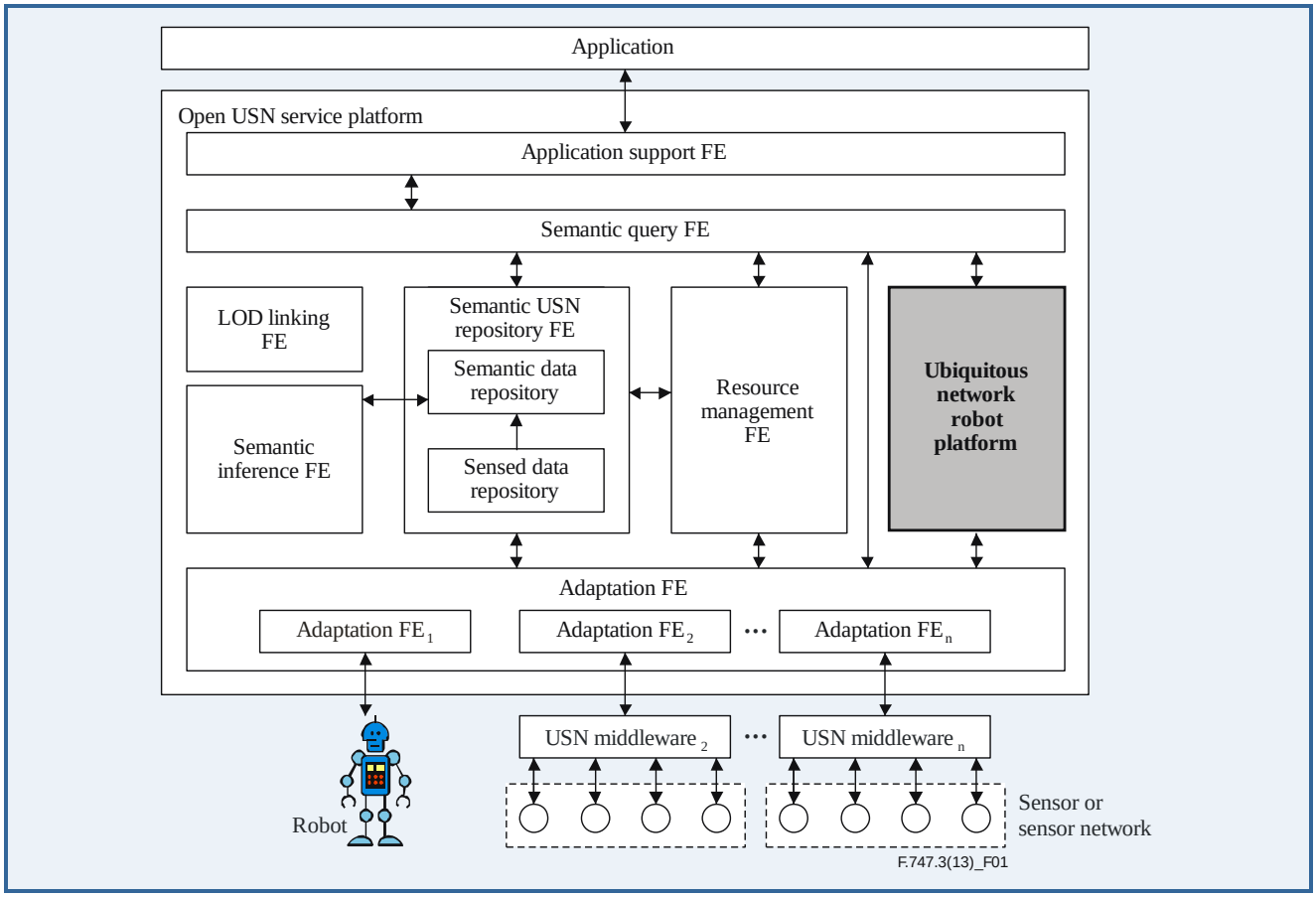


Figure 1 – Relation between open USN service platform and UNR-PF

7 Use cases of ubiquitous network robot platform

7.1 Health support service

A health support service using two different platforms is shown in Figure 2.

In the figure, '1' is an electronic tag attestation platform. A user is registered in an entrance area as the first step. At the same time, disclosure level of user information is set here. The attestation platform discloses customer information to the user according to the user's demand. Next, the user sets up the alarm clock of get-up time in an alarm clock support area. The user can then exercise in the physical exercise area where the user's health condition is checked. A teacher robot gives instructions for the exercise and the assistance robot instructs the user according to the actions performed. The user's health condition is checked by the action recognition unit embedded in the robots. Finally, the user gets the health condition results in the health check area.

This example shows an implementation of two or more cooperative services and one independent service (alarm clock support area) on the network robot platform.

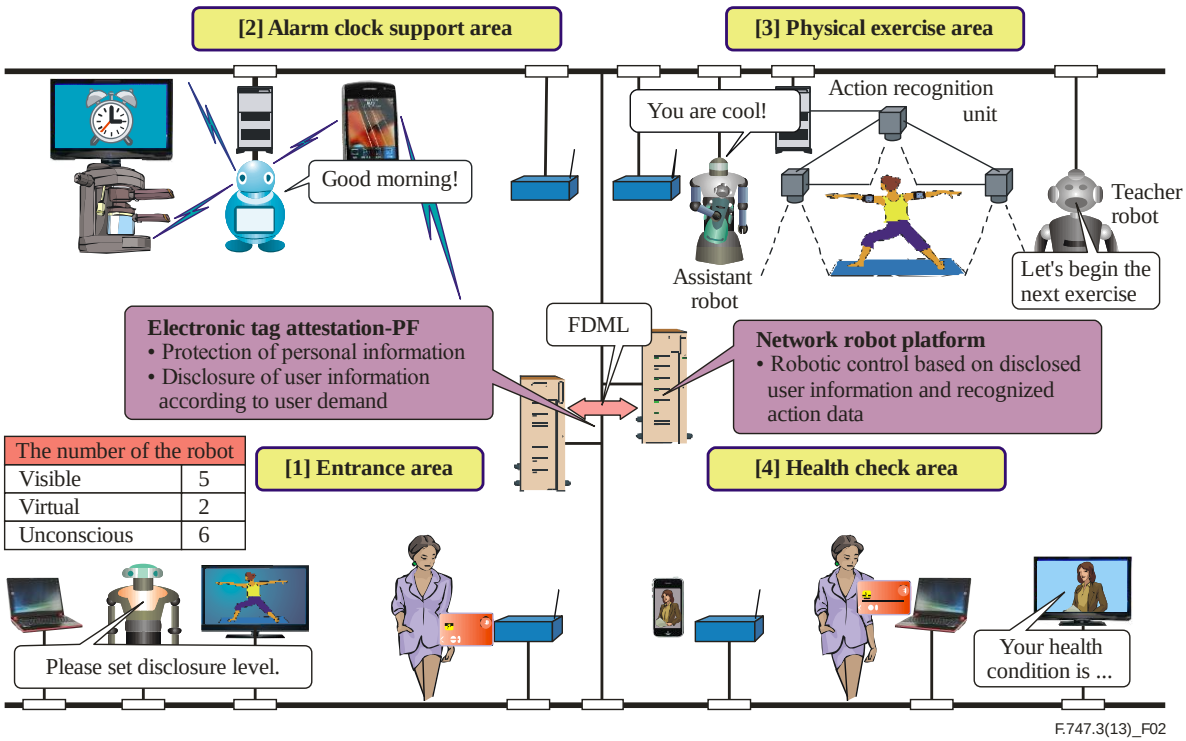


Figure 2 – Health support service

7.2 Shopping support service

The second use case of UNR-PF is shopping support ([b-Nishio2010], [b-OMG2012] and [b-Kamei2012]). Figure 3 shows the overview of the scenario. A typical service scenario is performed across three areas: customer's home (Area 1 in Figure 3), shopping mall (Area 2) and support centre (Area 3).

When a customer feels like going for shopping, he/she first makes reservation of the service using a smartphone. Here, with help from a remote operator, the customer makes a rough plan on what to buy. The operator or the smartphone application may provide the customer with recommendations on things to buy ("fish on sale today") or on where to go shopping ("many customers gave store X a high ranking"). After arriving at the mall, the customer finds that a robot is already waiting for him/her at the entrance. In this example, as the customer has a walking disability, a wheelchair robot is prepared. The customer sits on the wheelchair robot and is guided in the mall so that he/she can purchase the planned items.

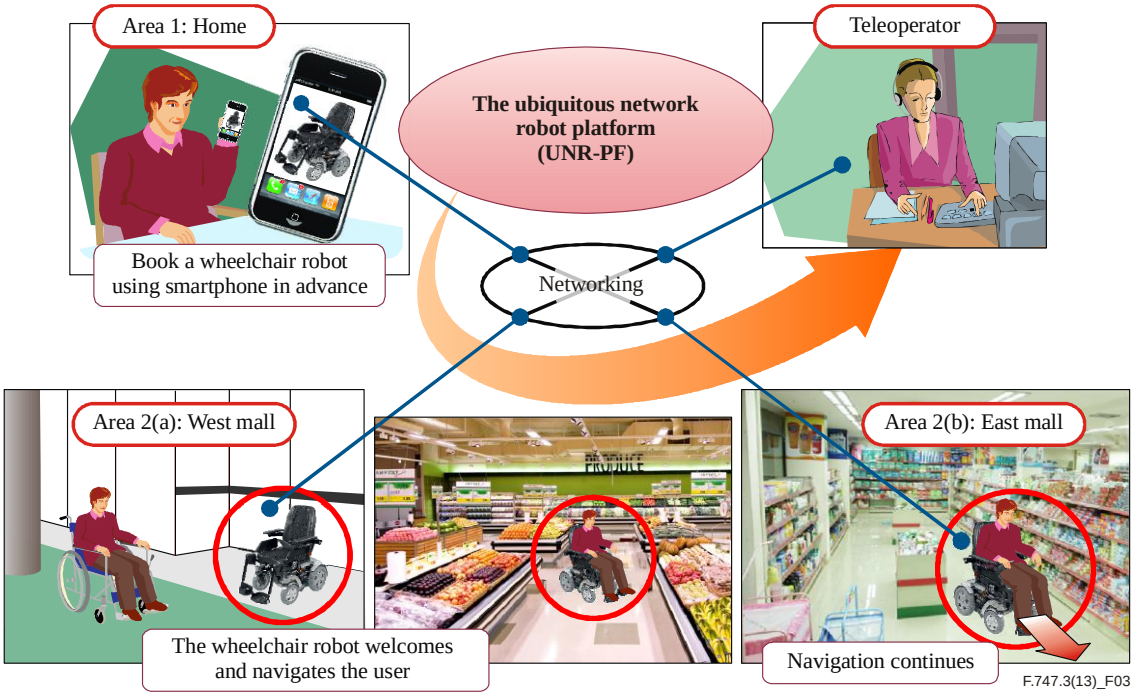


Figure 3 – Overview of the shopping support service

Figure 4 shows the structure and interaction sequence of the service. The customer first makes a reservation via the virtual-type robot on the mobile device at home. The virtual-type robot is connected to the local platform (LPF) installed in the user's home. LPF notifies the global platform (GPF) of the reservation information. The shopping support service application is connected to the GPF and receives the reservation request. After receiving the reservation request, the service application registers its service ID and a trigger condition (the user's arrival at the shopping mall, in the example) to the service queue on the GPF. The GPF refers to its map registry to confirm the LPF of the shopping mall and then registers the service and its starting condition to the LPF's service queue. When the user approaches the shopping mall, the virtual-type robot on the user's mobile device connects to the LPF of the shopping mall and notifies it of the user arrival. Then, the LPF of the shopping mall determines that the state meets the starting condition for the service and notifies the service application of the start via the GPF. Next, the service application requests the resource manager to reserve the robot in the shopping mall and the operator in the support centre via the GPF. To reserve the robot, the resource manager refers to the user registry and selects a suitable robot for the user. In the shopping support service, the resource manager selects a wheelchair robot if the user has difficulty in walking. Otherwise, it selects another type of mobile robot. After the allocation of the resources, the service is executed in accordance with the service flow defined in the service application. The service application refers to the map registry in the LPF and instructs the robot to navigate around the shopping mall.

In the case of a large shopping mall, the shopping support service is provided across several areas managed by different LPFs. When the robot comes close to the boundary between the two areas, the robot notifies GPF's state manager via the state manager of the LPF of the current area. Then, GPF's state manager refers to its map registry to find the LPF of the next area and then registers the service and its trigger condition, i.e., the customer's arrival at the next area or the service queue of the LPF. When the user arrives at the next area, the robot disconnects from the LPF of the first area and connects to the next LPF, if the robot is able to continue working in the next area. Otherwise, another robot in the next area comes to pick up the customer. Thus, the service is executed continuously. In this way, the customer can receive the service seamlessly in a wide area regardless of the area segmentation for the robotic system. When an operator is required, the service application requests the assignment of an operator to the LPF through GPF. The assigned operator will assist the user by remotely taking the user's detailed request and adapting it for the platform.

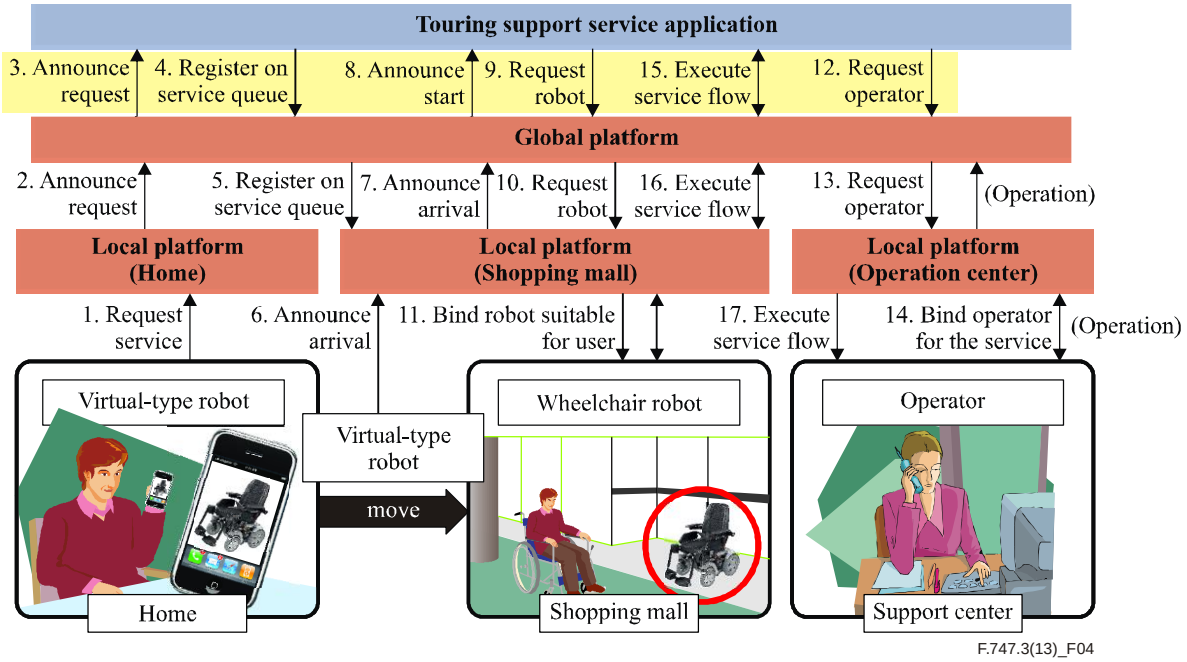


Figure 4 – Overview of the shopping support service sequence

8 Requirements for UNR-PF

8.1 Abstraction of functionality

It is required that UNR-PF provides a standardized and abstracted interface for controlling as well as receiving results from USN nodes. That is, from application point of view, UNR-PF provides access to a set of abstracted functionalities instead of raw USN nodes.

One of the most simple use cases of UNR-PF is collaboration among sensor networks and robots. A sensor network measures phenomena in the real world, notifies them to one or more applications, and then the application commands actuator devices, such as robots and smartphones, to interact with people. Such collaboration may be performed by USN, without UNR-PF, when treating robots as one of the "sensor nodes" with actuator facility. However, by the abstraction of functionality provided by UNR-PF, this can be realized in a much simpler way.

Consider the example shown in Figure 5. Here, based on the sensing result, a person is approaching a dangerous construction area, a robot warns the person. In order to warn the person, a) a mobile robot can approach the person and give a warning, or b) a nearby loudspeaker can warn the person. From the viewpoint of the effect, i.e., to let the person know that he/she is approaching a dangerous area, these two devices perform equivalent services. Dangerous spots change every day. If the application is coded to use a navigation robot and if there are no robots available nearby, the customer would not receive a warning. However, if the application is programmed to utilize the "alert" functionality, then the system would search for a device that can provide such functionality and choose it based on resource availability. This is also useful from the point that applications are not bound to a certain device at the time when it was coded. The application can then continue to provide service without change even though the devices have changed, if they provide the same kind of functionality.

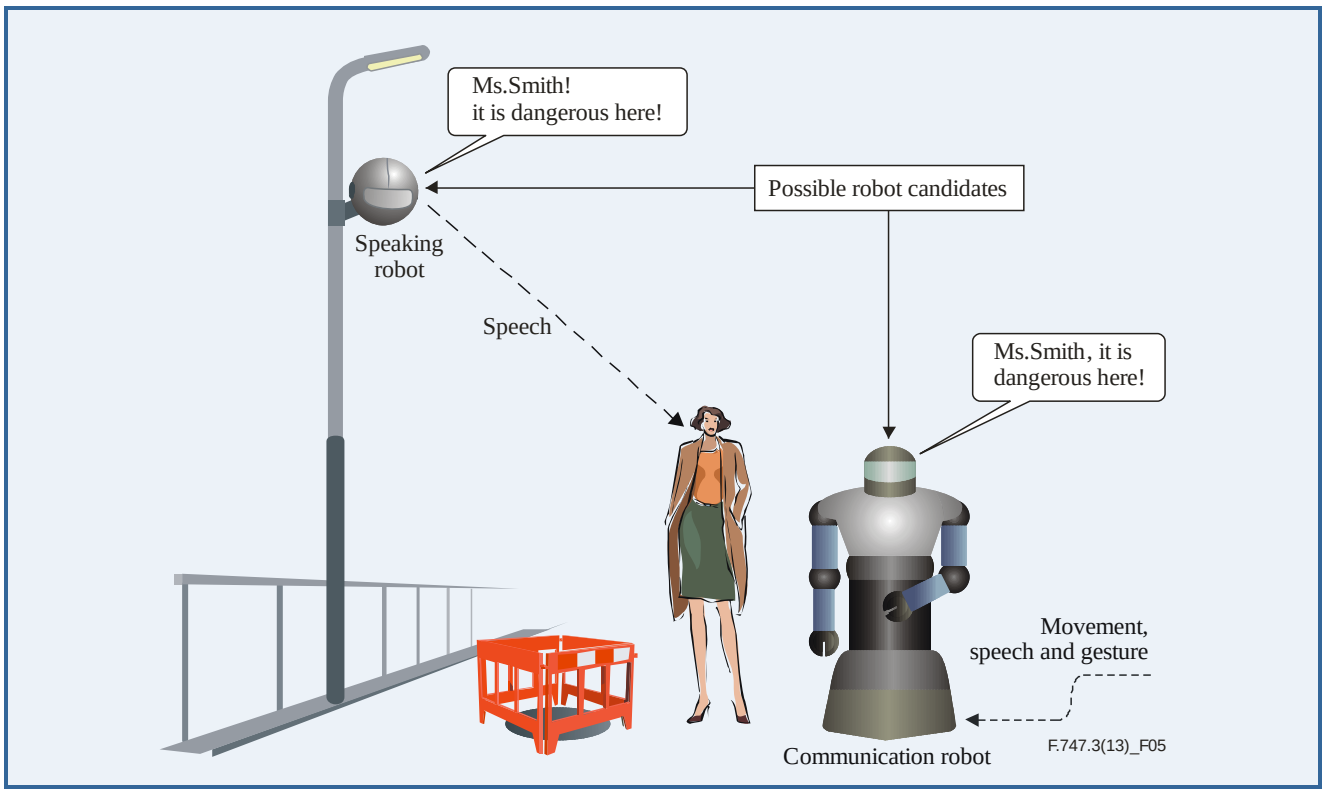


Figure 5 – Service execution by abstracted functionalities

Although the above case focuses on the actuation part, the same concept can be applied for the sensing part. When the sensing functionality is also abstracted, applications can request abstracted sensing functionalities, such as detection of dangerous areas, to the infrastructure, while the actual sensing equipment or sensor network in use may change.

8.2 Inter-service collaboration

It is required that UNR-PF provides functionalities to allow collaboration between different USN applications for realizing compositional applications.

Consider the photo printing service shown in Figure 6(a). This service has two tasks: taking a picture and then printing it out. This represents a simple functional dependency. Robot A possesses the photo taking function while Robot B possesses the photo printing function; they are spatially separated, but linked via a network. To realize this service, the photo printing task can be performed only after the photo taking task. Therefore, the robots need to interact with the customer based on the customer's history.

Another example is a service that guides exhibits in a museum, exhibition hall, etc., as shown in Figure 6(b). The tasks are to explain the exhibit and to guide the visitor to the next exhibit. Given that the visitor is free to visit the exhibits in any order, each robot needs to modify its responses according to the visitor's service history.

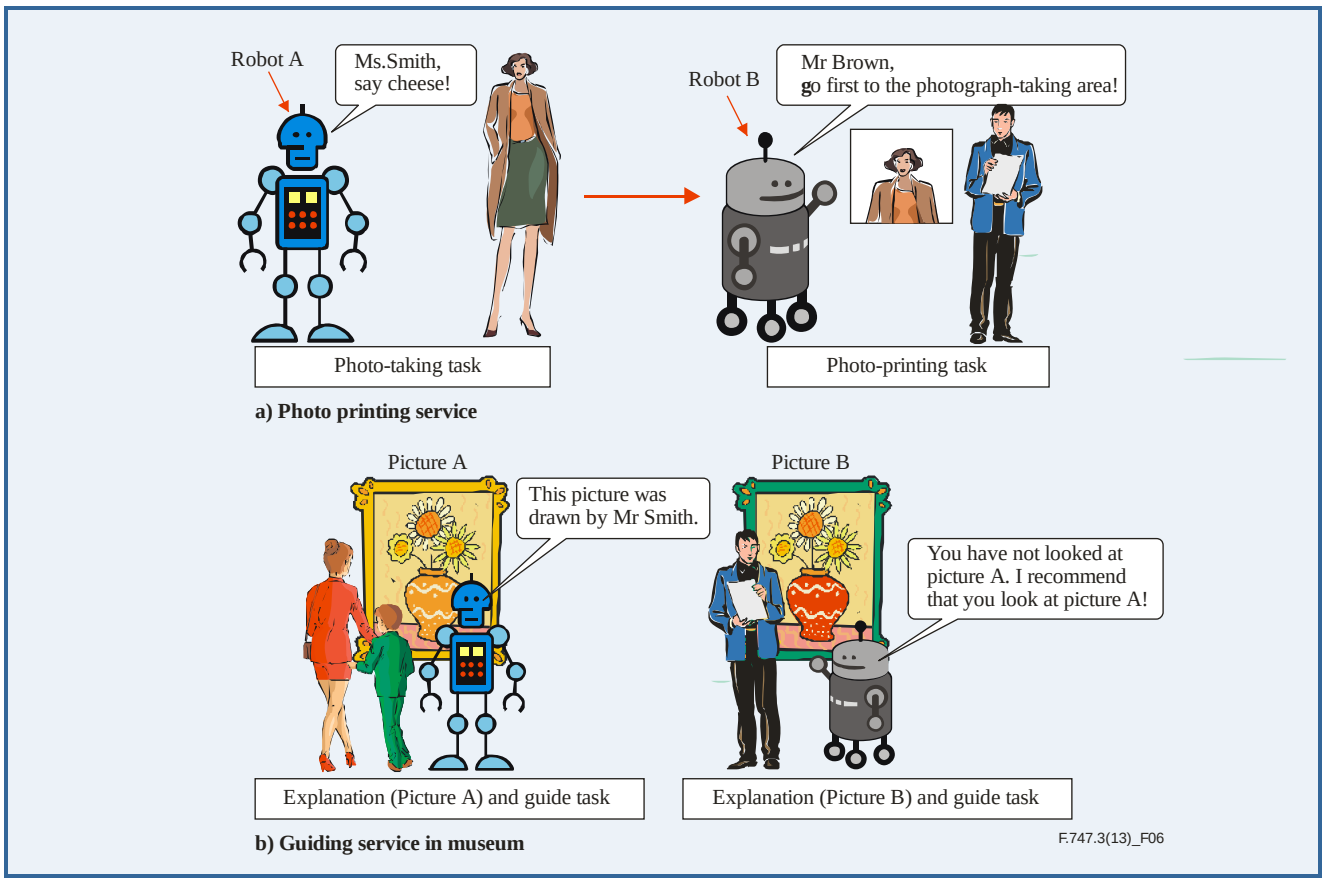


Figure 6 – Inter-service collaboration

As can be seen from these examples, there are many cases where multiple tasks or applications need to collaborate with each other to provide a consistent service to customers. In some cases, applications from different service providers need to collaborate. In the photo printing service example, photo taking and printing may be performed by different companies. In both situations of Figure 6, the important thing is that the applications share certain aspects of customer history. In the photo printing service case, only photograph data may be exchanged. However, in the museum guidance service, robots may share more detailed information, such as in which point the customer was interested, what kind of questions did the customer ask, or how long the customer spent in front of a Van Gogh painting. In some cases, such information sharing can be performed within frameworks of traditional information system. However, a common infrastructure is needed for sharing information required for affecting human-robot interaction and reflecting it on the behaviour of the robot.

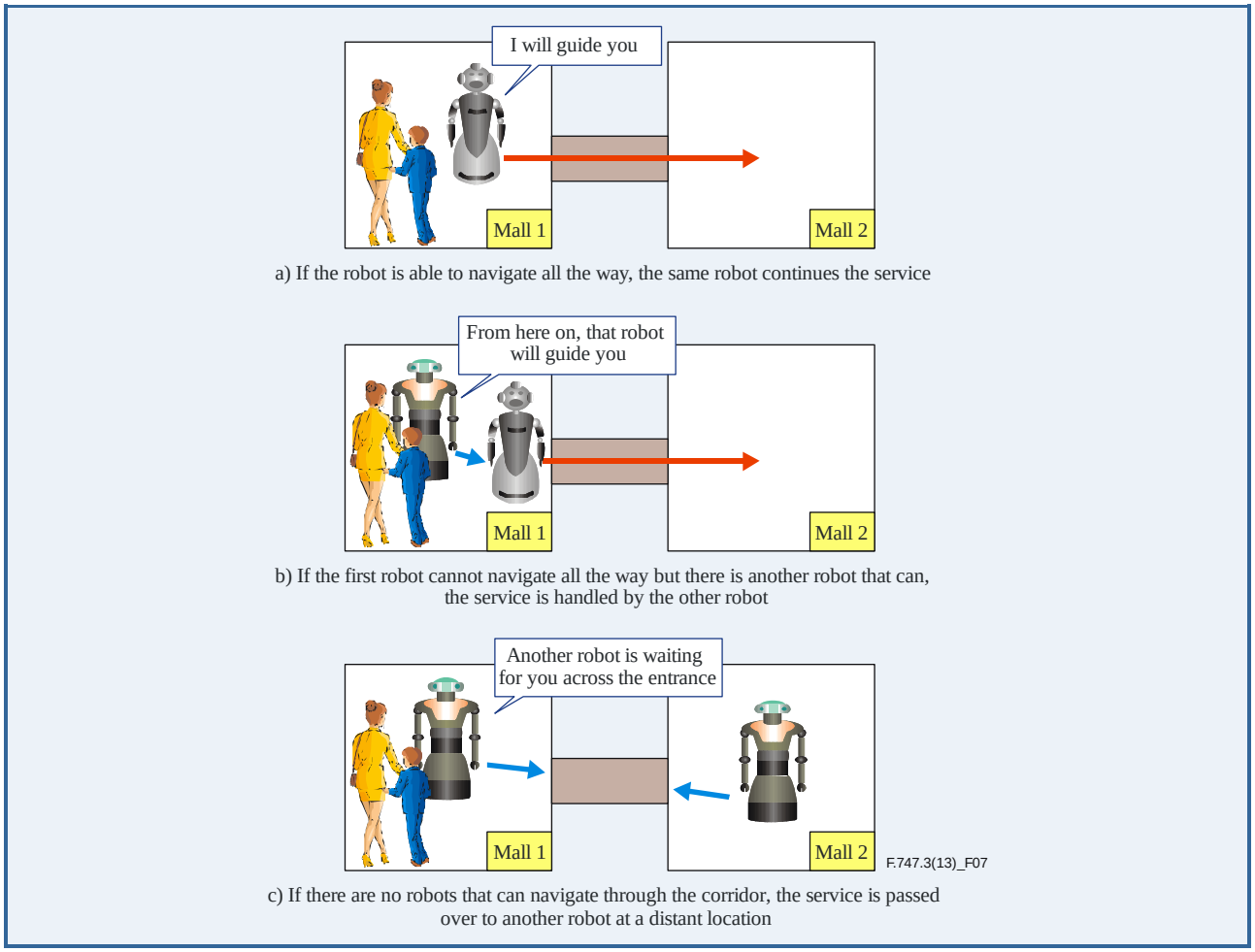
8.3 Service among multiple areas

It is required that UNR-PF provides functionalities to allow applications to seamlessly utilize USN nodes in multiple areas.

One of the key aspects in multiple-device collaboration is resource allocation in the spatiotemporal domain. Robots, as well as sensor networks, can only be effective in limited areas due to their physical nature. Sensors can only sense phenomena happening in a limited area, and robots can only serve within a certain area. For example, robots nowadays have a limited navigation capability and suffer interference from obstacles, floor materials and bumps. Battery life is another concern; after a while, robots need to be recharged. Due to these limits, in order to serve people that have much wider area of activity, multiple robots and sensor networks need to collaborate with each other while applications run continuously. For example, consider a navigation service where a robot guides a customer around a shopping mall. Depending on its ability, the robot may not be able to

navigate in certain areas such as outdoor corridor between different mall buildings. In such cases, however, the service must continue. Therefore, UNR-PF will perform handover of the navigation act that is necessary for the guidance service (Figure 7). Such handover may occur in different patterns based on the ability of the robot and the resource availability. A similar situation is likely to happen for sensing equipment, such as for localizing or identifying people throughout the mall. When a customer reaches the end of sensing area of one device, UNR-PF searches for other devices that can continue to sense the customer so that the application can continue to receive and utilize sensing results.

In order to realize a smooth handover between various devices, UNR-PF shall provide rich support so that applications do not need to be concerned about the actual resource availability or the device ability. UNR-PF shall manage various devices not only spatially but also in temporal space in order to effectively allocate them according to application requirements. At the same time, in order to perform effective handovers among devices of various natures and in varieties of areas, UNR-PF needs to manage spatial information as well as specifications of devices. As shown in Figure 8 ([b-Kolbe2011] and [b-OGC2011]), the navigation service is one typical example that clearly shows necessity for such information. In many indoor and outdoor areas, there are places not suitable for robots to move around. This limitation depends on the specifications of robots and of the kind of service the robot needs to perform. Moreover, the spatial nature may change with time. For example, think of a case where a robot guides the user around a station. In mornings and evenings of weekdays there are huge numbers of people passing through during the rush-hour, and thus the corridors become not suitable for robots to move around. But at other hours, the station is less crowded and the robot can easily perform its service. As there are many cases where such dynamic changes occur, the spatiotemporal nature of areas shall be managed and considered by UNR-PF.



a) If the robot is able to navigate all the way, the same robot continues the service

b) If the first robot cannot navigate all the way but there is another robot that can, the service is handled by the other robot

c) If there are no robots that can navigate through the corridor, the service is passed over to another robot at a distant location

Figure 7 – Example of service handover between robots

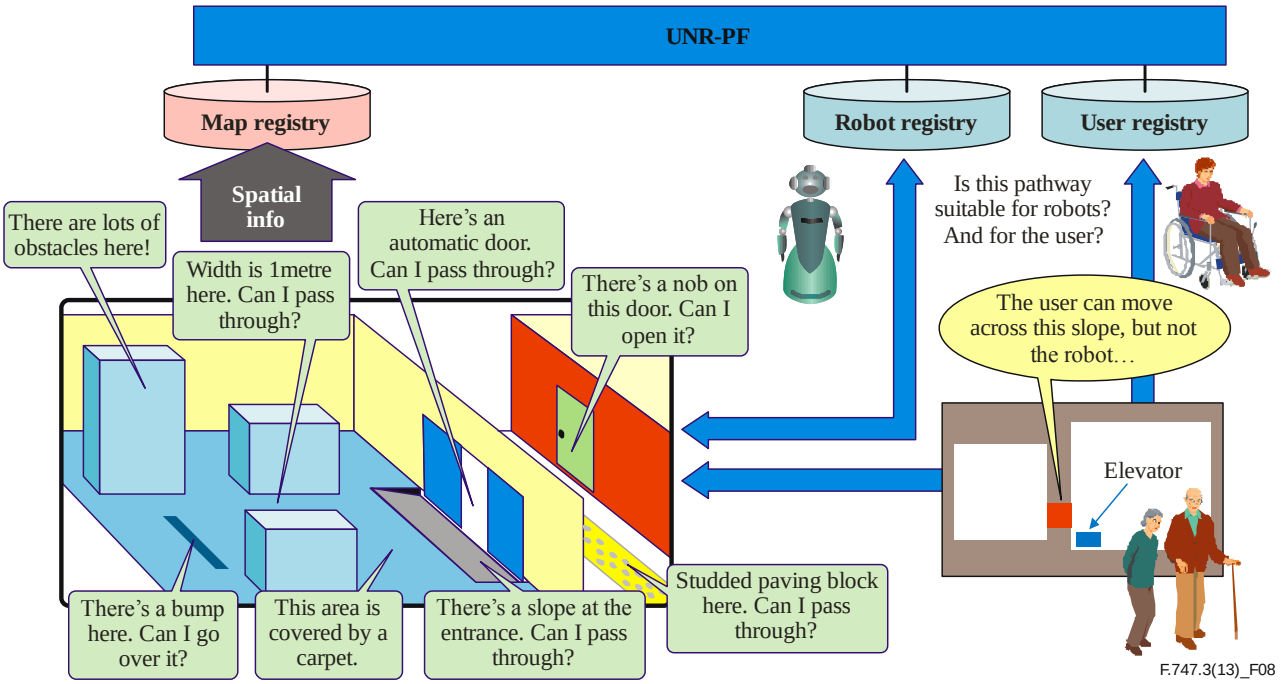


Figure 8 – Spatial information for robot and customer activities

8.4 Service execution based on customer attributes

It is required that UNR-PF provides functionalities for selecting appropriate USN nodes based on customer attributes.

An important requirement for UNR-PF is the selection of devices based on customer attributes. Similar to selecting combination of robots based on spatiotemporal availability and robot ability, UNR-PF also needs to consider customer attributes for its robot allocation planning. As for utilizing general customer tendencies, many data mining algorithms and applications have been built and are now actively in use, such as online stores. In such sites, the application servers "remember" what a certain customer chose in the past and would recommend products that are likely to suit the customers. However, for real-world applications using robots, this is a more serious issue. One case that the UNR-PF handles relates to physical disabilities as shown in Figure 8. If a customer can only walk slowly, the robots also need to slow down. If a customer is troubled by steep inclines, the chosen route should consist of flat paths. When a customer requires a wheelchair to move around, a wheelchair robot or a robot that can push wheelchairs shall be used for application execution. As such, walking ability is one clear example to show why customer attributes are important in planning device allocation for service execution. There are, however, a number of such factors, including visual disabilities or impairment in audition. Based on descriptions for each customer, UNR-PF needs to carefully select appropriate devices to make the customer experience as good as possible.

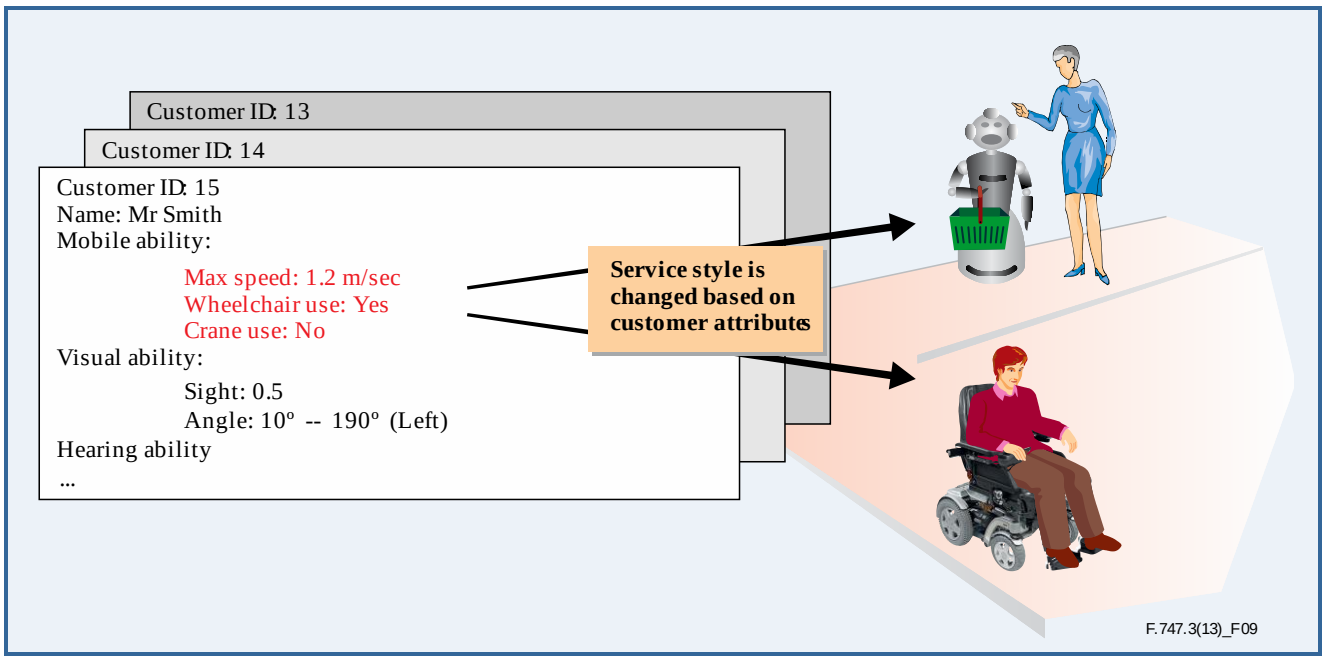


Figure 9 – Service execution based on customer ability

9 Functional model for UNR-PF

A general network robot system consists of three parts as shown in Figure 10:

1) Robot service applications

2) UNR-PF

3) Robotic functions

The first robotic function consists of robot service applications that define flow of the service contents, and are typically held and maintained by service providers, using the common interface that UNR-PF provides. The third robotic function consists of various robots that are dedicated to specific capabilities and registered to UNR-PF through the common interface.

The second robotic function is UNR-PF, which is the subject of this Recommendation, which establishes the interaction between the service applications and the robots. The functional model of UNR-PF is shown in the dotted part of Figure 10. This UNR-PF has two layers: the lower layer consists of reusable functional modules that belong to individual robots, and the higher layer manages logic for service contents.

The UNR-PF should be independent of and serve common functions to the other two parts. Once such a common platform is developed and is stable, it will separate robotic functions and services so that the whole robotic system can be developed at lower cost and become more dependable in terms of reusability, scalability and availability.

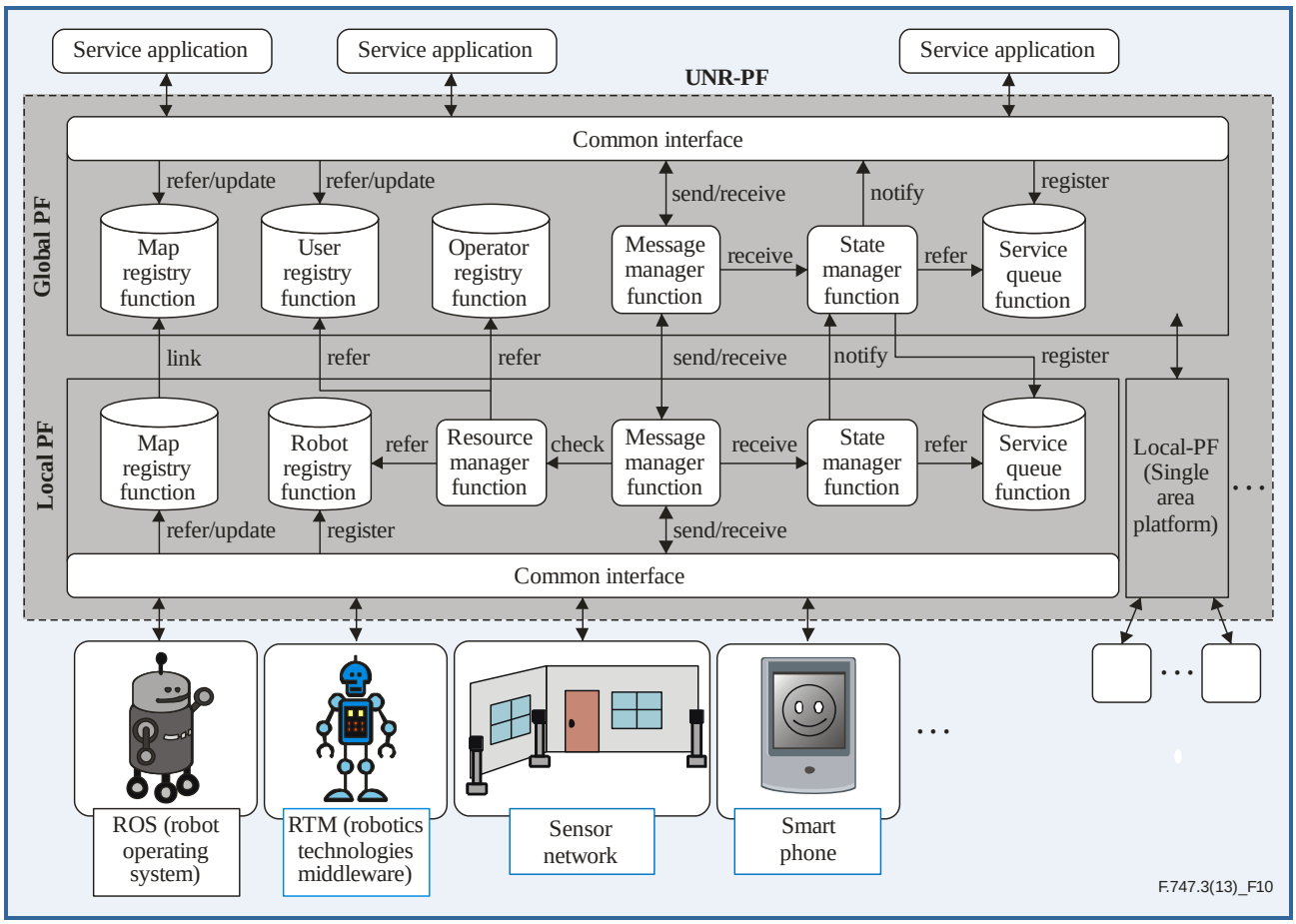


Figure 10 – Functional model of UNR-PF

As such, UNR-PF serves as middleware between services and robotic devices and is composed of two layers of network robot platforms: a local platform (LPF) and a global platform (GPF). LPF is a platform for configuring the robotic system in a single area. GPF is a platform for configuring the robotic system in a wider range of areas that includes a number of LPFs. These platforms serve as a middle-layer between robot service applications and robotic functions. The platform has five database functions and three management functions. The database functions consist of robot registry, operator registry, user registry, map registry and service queue. The management functions consist of a state manager, resource manager and message manager. Each function is detailed below. Refer to Figure 4 in clause 7.2 about concrete message flows among three parts.

UNR-PF contains several registries (database) for managing attributes for robots, operators and robotic service users. These are used for selecting appropriate robots and/or operators on the basis of each user's demands.

9.1 Robot registry function

The robot registry contains information about the robots available in each area, such as their shapes and capabilities as well as their statuses. In addition, functionalities that each robot provides, such as navigation, people detection or face recognition, are stored in the registry as functional profiles. These pieces of information show what kind of service a certain robot can perform, what kind of route the robot can navigate, and whether the robot is available now or in the near future. From this information, UNR-PF can decide what device to allocate for a certain application.

9.2 Operator registry function

Operators for tele-operation of robots are managed in the operator registry. Such operators are often necessary to compensate for the artificial intelligence limitations in the robot. In UNR-PF, operators are treated as one kind of robot. The same selection mechanism works for the operators on respective service application demands based on the information stored in the operator registry.

9.3 User registry function

The user registry holds attributes on each customer who wishes to receive robotic support, as well as a history of services that the customer has used. From these pieces of information, UNR-PF can allocate the robots and operators necessary to provide certain services.

9.4 Map registry function

The map registries in both LPF and GPF are used to improve the linkage between different areas. The map registry of the LPF contains spatial information of the service execution environment, such as the properties of the floor and information about movable zones and keep-out zones. The map registry of the GPF contains the positional relationship among single areas.

9.5 Service queue function

The service queue is a function implemented in both LPF and GPF. This function is used to manage the start of the service. This database contains IDs of the services and their initiation conditions. At first, the service application registers its ID and initiation condition to the service queue in the GPF and the state manager in the GPF registers the ID and condition to the service queue in the appropriate LPF in accordance with the state notification from the LPFs.

9.6 State manager function

The state manager is implemented in both LPF and GPF. This function subscribes to the message manager for the state notifications that are registered in the service queue. When receiving the state notification, the manager determines if the state complies with the start conditions in the service queue. If the state complies with the start condition, the manager sends a message to the service application to start the service.

9.7 Resource manager function

The resource manager is implemented in LPF. When receiving the command message for executing a service application, the resource manager refers to the robot registry, the user registry and the operator registry and reserves the robot suitable for the customer and the operator who can operate the service depending on the situation.

9.8 Message manager function

The message manager is implemented in both LPF and GPF. This function manages the messages exchanged between the service applications and the robotic functions through the common interface. The robotic functions provide the profile of available messages to the message manager.

When receiving the message from the service applications, the message manager refers to the profiles and selects the robotic functions that fit the service application requirements. In LPF, the message manager requires the resource manager to reserve the necessary resources. When receiving the message, i.e., the state notification from the robotic function, the message manager checks the delivery addresses, which are given at the time of the state notification subscription, and forwards the message to the appropriate state manager and/or the service applications.

Bibliography

[b-Jung2007] Jung, H.-S. et al., *Unified Ubiquitous Middleware for U-City*, in Proc. ICCIT2007, pp. 2374-2379, 2007.

[b-Kamei2012] Kamei, K. *et al.*, *Cloud Network Robotics*, IEEE Network, Vol. 26, No. 3, pp. 28-34, 2012.

[b-Kolbe2011] Kolbe, T.H., *Indoor Localization and Tracking supported by 3D Models*, OGC Standards for the Internet of Things – a Workshop, 2011.

[b-Nakamura2008] Nakamura, Y. *et al.*, *Framework and service allocation for network robot platform and execution of interdependent services*, Robotics and Autonomous Systems, Vol. 56, pp.831-843, 2008.

[b-Nishio2010] Nishio, S. *et al.*, *Robotic Localization Service Standard for Ubiquitous Network Robots*, in Abdellatif, H. ed., Robotics 2010 Current and Future Challenges, pp. 381-400, In-Tech: Vukovar, Croatia, 2010.

[b-OGC2011] Open Geospatial Consortium, *OpenGIS City Geography Markup Language (CityGML) Encoding Standard*, version 1.1.0, 2011.

[b-OMG2012] *Object Management Group, Robotic Localization Service*, version 1.1, formal/2012-08-01, http://www.omg.org/spec/RLS/, 2012.

[b-Shiomi2011] Shiomi, M. *et al.*, *Field Trial of a Networked Robot at a Train Station*, Int. J. Social Robotics, Vol. 3, No. 1, pp. 27-40, 2011.

[b-Salvini2011] Salvini, P. *et al.*, *Do Service Robots Need a Driving License?*, IEEE Robotics and Automation Magazine, Vol. 18, No. 2, pp. 12-13, 2011.

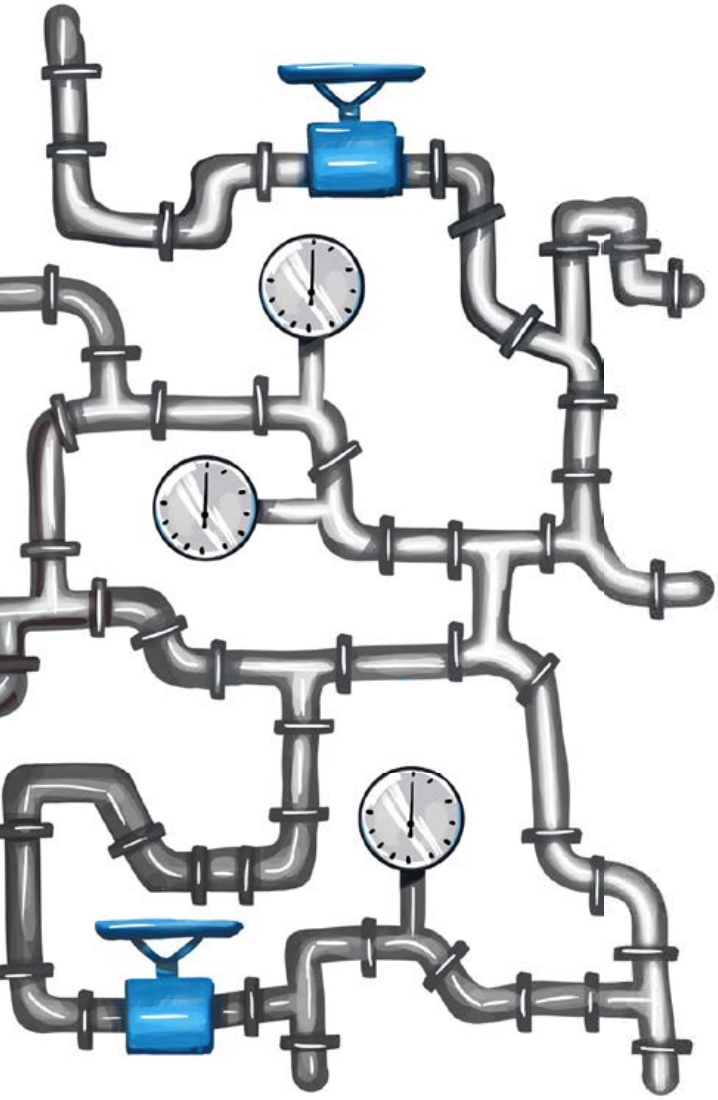
[b-Sanfeliu2008] Sanfeliu, A. et al., *Network Robot Systems*, Robotics and Autonomous Systems, Vol. 56, pp. 793-797, 2008.

[b-Sanfeliu2010] Sanfeliu, A. *et al.*, *Decentralized Sensor Fusion for Ubiquitous Networking Robotics in Urban Areas*, Proc. Sensors, pp. 2274-2314, 2010.

[b-Sato2011a] Sato, M. *et al.*, *Standardizing Framework for Robotic Services and Functions*, Proc. ICRA Workshop on Robotics Modular Architecture Design and Standardization, 2011.

[b-Sato2011b] Sato, M. *et al.*, *The Ubiquitous Network Robot Platform: Common Platform for Continuous Daily Robotic Services*, in Proc. 2011 IEEE/SICE International Symposium on System Integration, pp. 318-323, 2011.

[b-Tezuka2006] Tezuka, H. *et al.*, *Robot platform architecture for information sharing and collaboration among multiple networked robots*, Journal of Robotics and Mechatronics, Vol. 18, No. 3, pp. 325-332, 2006.



Y.4107/F.747.6

Requirements for water quality assessment services using ubiquitous sensor networks (USNs)

Requirements for water quality assessment services using ubiquitous sensor networks (USN)

Summary

To make a safe and ecologically healthy water environment, assessment of changes in water quality and water quality monitoring are required in rivers, lakes and other bodies of water. Recommendation ITU-T F.747.6 describes scenarios for the applications of water quality assessment and the sensor network technology that is the most suitable method to fulfil it.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T F.747.6	2014-10-14	16	11.1002/1000/12226

Keywords

Ubiquitous sensor network (USN), water quality assessment (WQA).

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

Table of Contents


10:08 AM
10:08 AM
Saturday, Jul 12

Recommendation ITU-T Y.4107/F.747.6

Requirements for water quality assessment services using ubiquitous sensor networks (USNs)

1 Scope

This Recommendation identifies requirements and scenarios for water quality assessment (WQA) services using ubiquitous sensor networks (USNs). The scope of this Recommendation covers the following:

– Overview of WQA;

– WQA scenarios;

– Requirements for WQA services;

– USN capabilities for supporting the requirements of WQA services.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T F.744] Recommendation ITU-T F.744 (2009), *Service description and requirements for ubiquitous sensor network middleware*.

[ITU-T Y.2221] Recommendation ITU-T Y.2221 (2010), *Requirements for support of ubiquitous sensor network (USN) applications and services in the NGN environment*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 processed data [ITU-T F.744]: Data that are processed from raw sensed data by the sensor network or USN middleware.

3.1.2 sensed data [ITU-T F.744]: Data sensed by a sensor that is attached to a specific sensor node.

3.1.3 sensor [ITU-T Y.2221]: An electronic device that senses a physical condition or chemical compound and delivers an electronic signal proportional to the observed characteristic.

3.1.4 sensor network [ITU-T Y.2221]: A network comprised of inter-connected sensor nodes exchanging sensed data by wired or wireless communication.

3.1.5 sensor node [ITU-T Y.2221]: A device consisting of sensor(s) and optional actuator(s) with capabilities of sensed data processing and networking.

NOTE– In WQA environment, these sensor nodes have sensing and networking capabilities except sensed data processing.

3.1.6 ubiquitous sensor network (USN) [ITU-T Y.2221]: A conceptual network built over existing physical networks which makes use of sensed data and provides knowledge services to anyone, anywhere and at any time, and where the information is generated by using context awareness.

3.1.7 USN end-user [ITU-T Y.2221]: An entity that uses the sensed data provided by USN applications and services. This end-user may be a system or a human.

NOTE – In WQA environment, a WQA user is a kind of USN end-user. This may be a WQA application or a human.

3.1.8 USN gateway [ITU-T Y.2221]: A node which interconnects sensor networks with other networks.

NOTE – In WQA environment, the USN gateway has the sensed data processing capabilities.

3.1.9 USN middleware [ITU-T Y.2221]: A set of logical functions to support USN applications and services.

NOTE – In WQA environment, a WQA server is a kind of USN middleware. The main functionalities of it are sensor network management and sensor data mining and processing.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 measured data: The sensing data by a sensor that is attached to a specific sensor node.

3.2.2 software sensor: Software that gets the processed and predicted data from measured real-time sensed data.

NOTE – WQA server has the software estimating the processed data (e.g., total nitrogen (TN) and total phosphorus (TP) values) using the water quality parameters (e.g., potential of hydrogen (pH), dissolved oxygen (DO), electrical conductivity (EC)) aggregated from sensors in sensor networks in real-time.

3.2.3 water quality assessment (WQA) node: A device measuring water quality and capable of sensing, processing, networking and optionally actuating.

3.2.4 WQA system: The devices consisting of sensor nodes with sensors, a USN gateway and a WQA server in order to support the water quality assessment.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

DO	Dissolved Oxygen
EC	Electrical Conductivity
IP	Internet Protocol
IPv4	Internet Protocol version four
IPv6	Internet Protocol version six
ORP	Oxidation-Reduction Potential
pH	Potential of Hydrogen
TN	Total Nitrogen
TP	Total Phosphorus
USN	Ubiquitous Sensor Network
WQA	Water Quality Assessment
ZIP	Zone Improvement Plan

5 Conventions

None.

6 Overview of water quality assessment

The water quality assessment (WQA) monitors dispersion of water pollution, tracking of a water pollutant source and predicts water quality change using the values measured from the measurement devices covering a specified area. It plays an important role to improve water quality through its real-time management.

WQA is divided into water quality management for wide areas and for middle and small-sized areas such as rivers or lakes. In the former case, it is easy to monitor a large water pollutant accident, while the latter is used to prevent the spread of water pollution and to monitor water pollution before the water pollutant, actually generated at middle and small-sized rivers, is diluted.

Applications for the WQA include the smart farm (for example, the horticultural and livestock industries) and smart leisure (for example, fishing where the angler is interested in information related to water quality, or the different opinions about a water pollutant source among local communities). These applications use sensor network technologies to assess water quality. The WQA nodes with water quality sensors deliver sensing data in real-time to a WQA server via wireless or mobile networks. The sensed data are used to monitor the water quality and track the water pollutant source in real-time. Furthermore, the large-scale deployment of a sensor network enhances the density of the WQA node thus realizing reliable water quality assessment. Besides, unmanned long-term operation of the WQA system is possible through network management technologies together with low power consumption and automated control of sensors.

Figure 1 shows the overall conceptual diagram for the WQA. The device with the water quality sensor, flow sensor, water level sensor, etc., by the rivers and lakes periodically measures the value of the water quality parameters (e.g., potential of hydrogen (pH), dissolved oxygen (DO)), the flow velocity, the water level, etc. The sensed data are delivered to the WQA server located in infrastructure network. WQA server estimates the WQA information based on the sensed data. WQA server provides the information to WQA users in real-time.

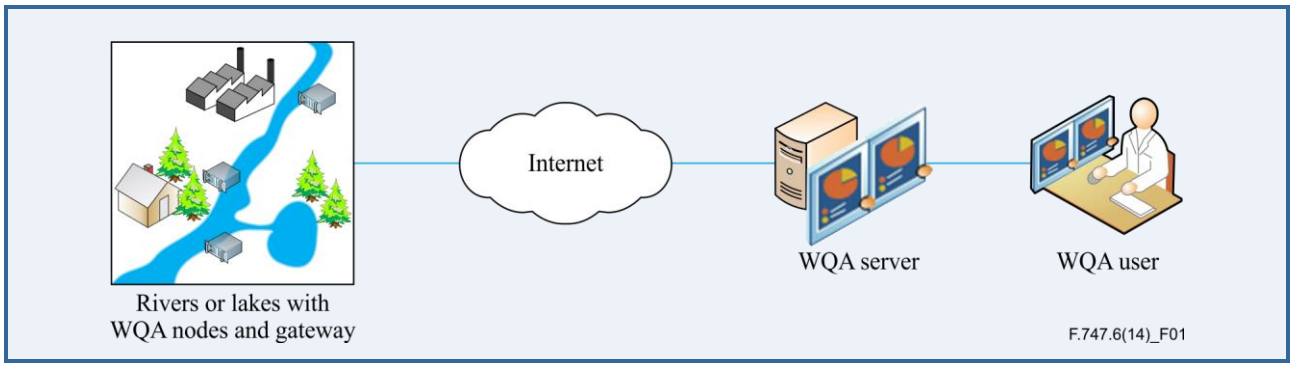


Figure 1 – Overall conceptual diagram for WQA

7 Scenarios for WQA services

The scenarios for the WQA include the following entities: the WQA nodes, server and users and are done through the interaction among them.

7.1 Scenario I: Real-time water quality data aggregation

Scenario I describes procedures where the measured data for the WQA are delivered periodically to the WQA server and, subsequently, the water quality information is provided to users in real-time.

1) The WQA server initially sets the data-sensing period of the WQA node.
2) The WQA nodes obtain the measured data periodically from the rivers, lakes, etc.
3) The data measured by the WQA nodes and gateway are delivered to the WQA server.
4) Steps 2 and 3 above are repeated after waiting for the data-sensing period. The WQA server estimates the water quality to provide the distribution of each water parameter from the delivered measured data.

   NOTE – The procedures from step 1 to step 3 are obtaining periodic measured data. In the scenario II and III, below, the same procedures are used.
5) The WQA server derives the water quality distribution map of each water quality item by applying the WQA model.
6) The WQA server provides the information on the water quality distribution to the WQA users.

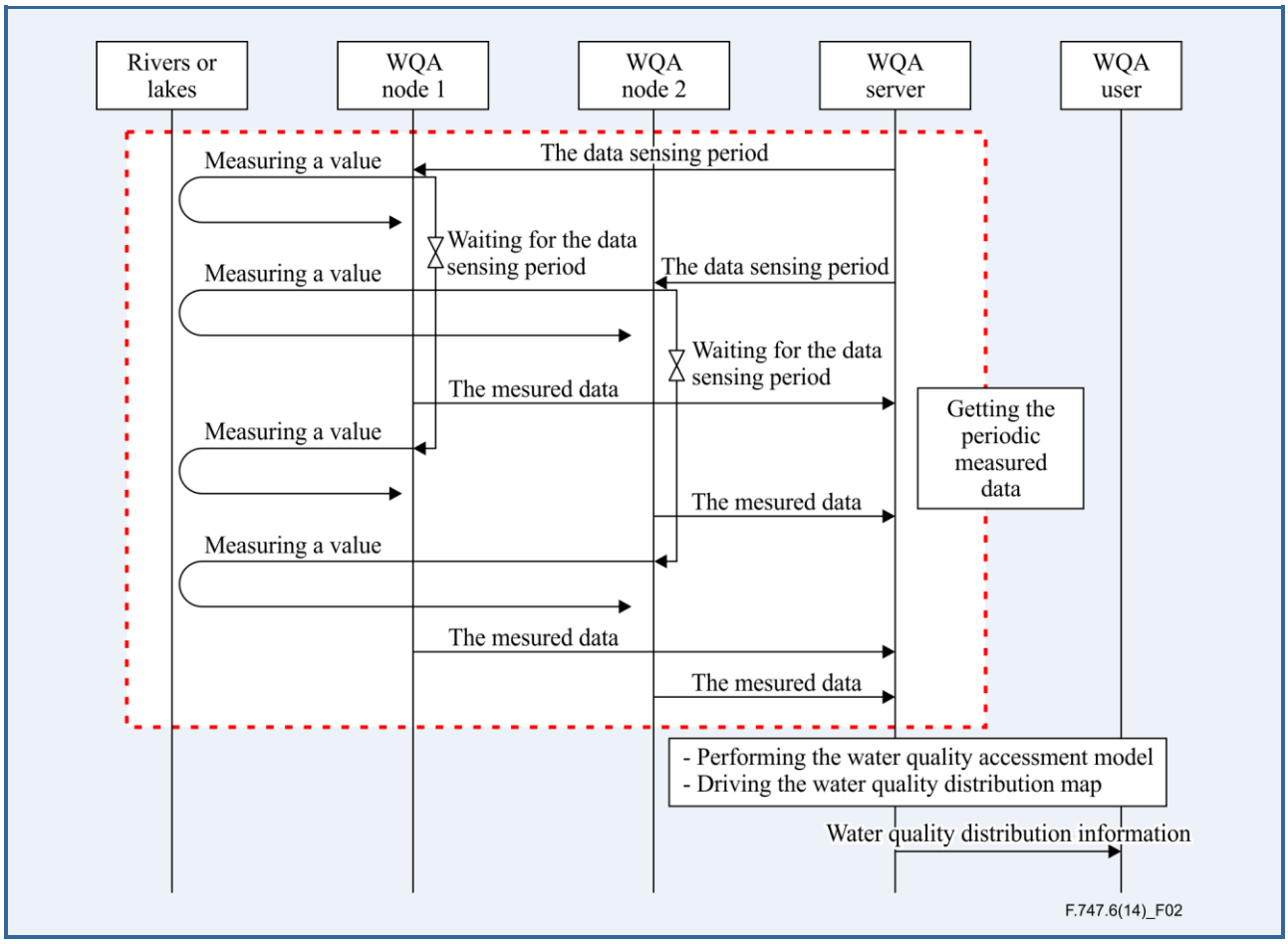


Figure 2 – Real-time water quality data aggregation

7.2 Scenario II: Automatic WQA node control

The WQA server monitors the water quality parameters measured from the WQA nodes and then filters any faults to improve the accuracy of the WQA. It also distinguishes sensing faults from the aggregated sensed data. For example, the WQA server can operate the sensor wiper of WQA node to prevent bio-fouling.

Scenario II describes procedures for long-term unmanned operation of the WQA where the WQA server recognizes the changes, or faults, of sensor values and automatically controls the data sensing period and the operation of the sensor wiper.

1) When the WQA server analyses the measured data, and if a change to the data sensing period is required, it requests the WQA nodes to change the period.

2) The WQA nodes obtain sensing data with the new data sensing period from the rivers, lakes, etc. The sensed data by the WQA node are delivered to the WQA server.

NOTE – The same procedure for obtaining the periodic measured data is used as in Figure 2.

3) The WQA server examines the changes or faults of sensor values.

4) When the WQA server recognizes changes, or faults, of sensor values, it automatically controls the data sensing period and the operation of the sensor wiper.

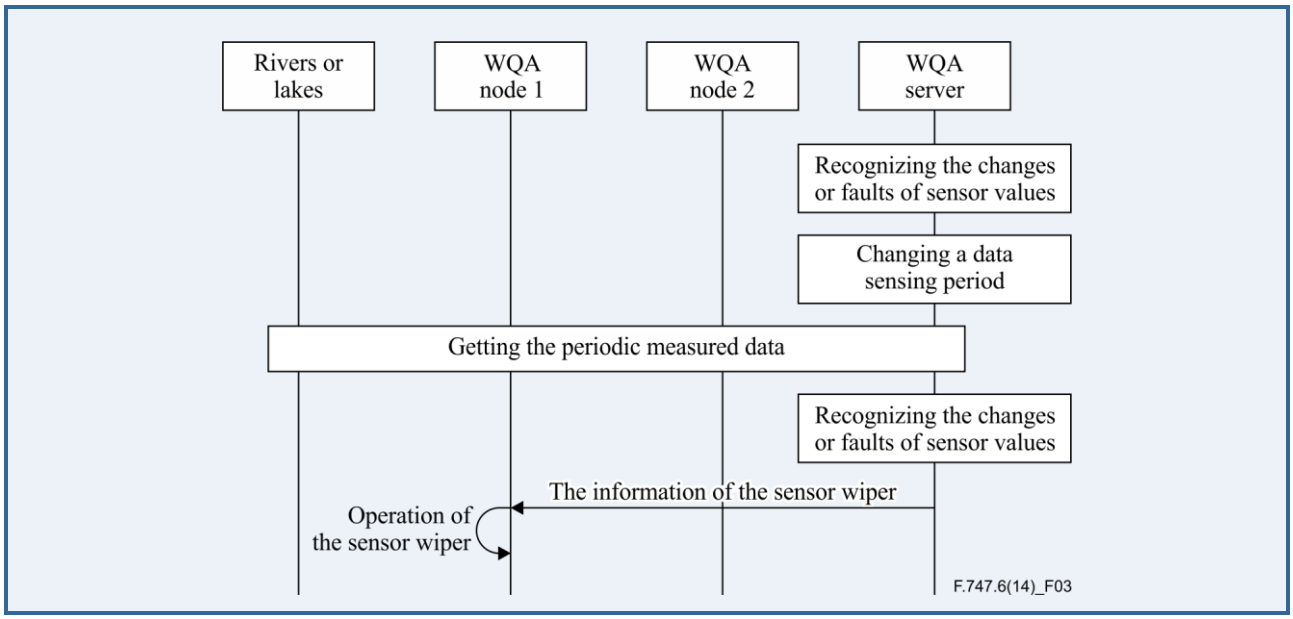


Figure 3 – Automatic WQA node control

7.3 Scenario III: WQA node surveillance and logging

Scenario III describes the procedure where the WQA server observes the WQA nodes and then detects, separates and diagnoses faults of the devices. If the communication between the WQA server and the WQA nodes is interrupted, the WQA node must log the measured data until it returns to a normal communication state. Thus, fault monitoring among the WQA nodes is important.

1) The WQA nodes obtain the measured data in every period from the rivers, lakes, etc. The measured data are delivered to the WQA server.

NOTE – The same as the procedure for obtaining the periodic measured data is used as in Figure 2.

2) The WQA server periodically performs surveillance of the WQA nodes.

3) The device receiving a surveillance request delivers the result to the WQA server. On the other hand, when a node detects a fault, it is delivered to the server.

4) When the communication between the WQA server and the WQA nodes is interrupted, the WQA node logs the measured data.

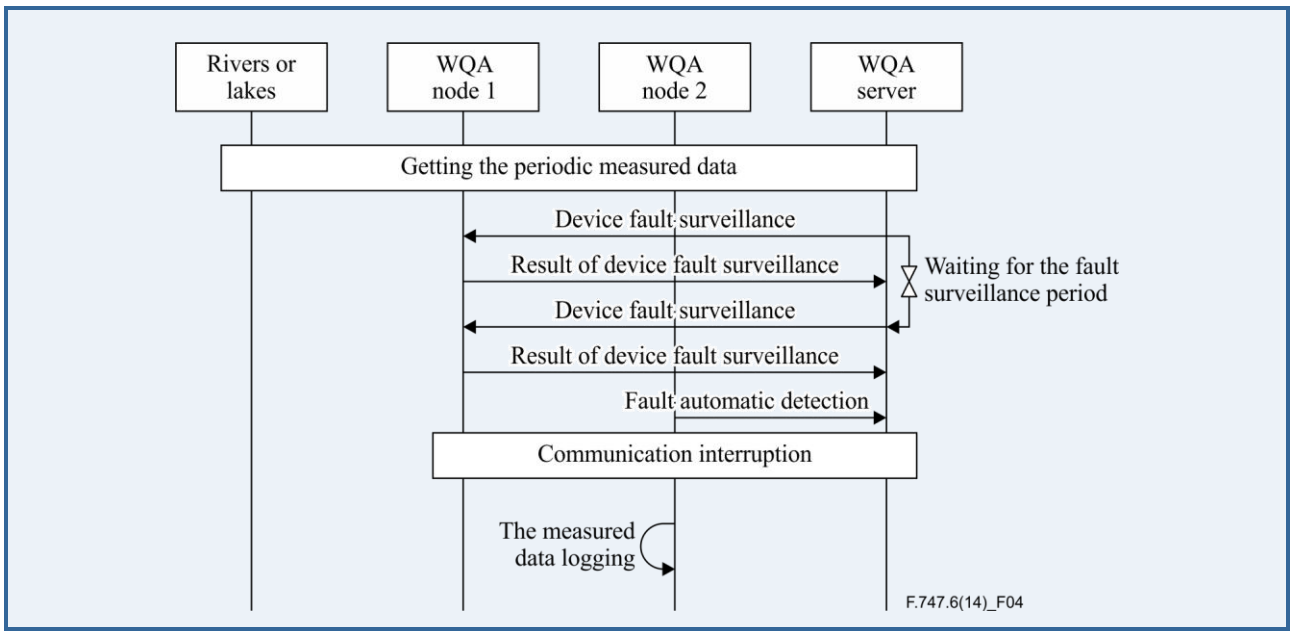


Figure 4 – WQA node surveillance and logging

7.4 Scenario IV: Water quality prediction through software sensors

Scenario IV describes procedures that predict values of water quality parameters through software sensors. In general, WQA nodes measure the values of water quality parameters (e.g., pH, DO, electrical conductivity (EC)) in real-time using water quality sensors. However, total nitrogen (TN) and total phosphorus (TP) values, which are important criterion parameters for judgment of water pollution, are not measured via the WQA nodes in real-time due to its measurement method. Hence, the values of these parameters are predicted through the software sensor in real-time. TN and TP values obtained from the software sensor are offered as the input values for the analysis of water quality distribution, tracking of a water pollutant source and the prediction of the water quality change.

1) The WQA nodes obtain the measured data in every period from the rivers, lakes, etc.

2) The measured data are delivered to the WQA server.

3) The WQA server updates the estimation function of the software sensor with the measured data. Here, initial TN and TP values are time series data measured in advance (measured by getting the water samples and testing its quality at the laboratory or by underwater pumps at the monitoring stations that are built beside rivers). The rest of the values (e.g., pH, DO, EC) are time series data measured in real-time from WQA nodes.

4) The WQA server performs the estimation function of the software sensor. The software sensor, based on the values measured from WQA nodes, estimates TN and TP values in real-time.

5) The WQA server delivers the values from the software sensor module to the WQA module in order to analyse the water quality distribution, the water quality prediction and the total amount of water pollution.

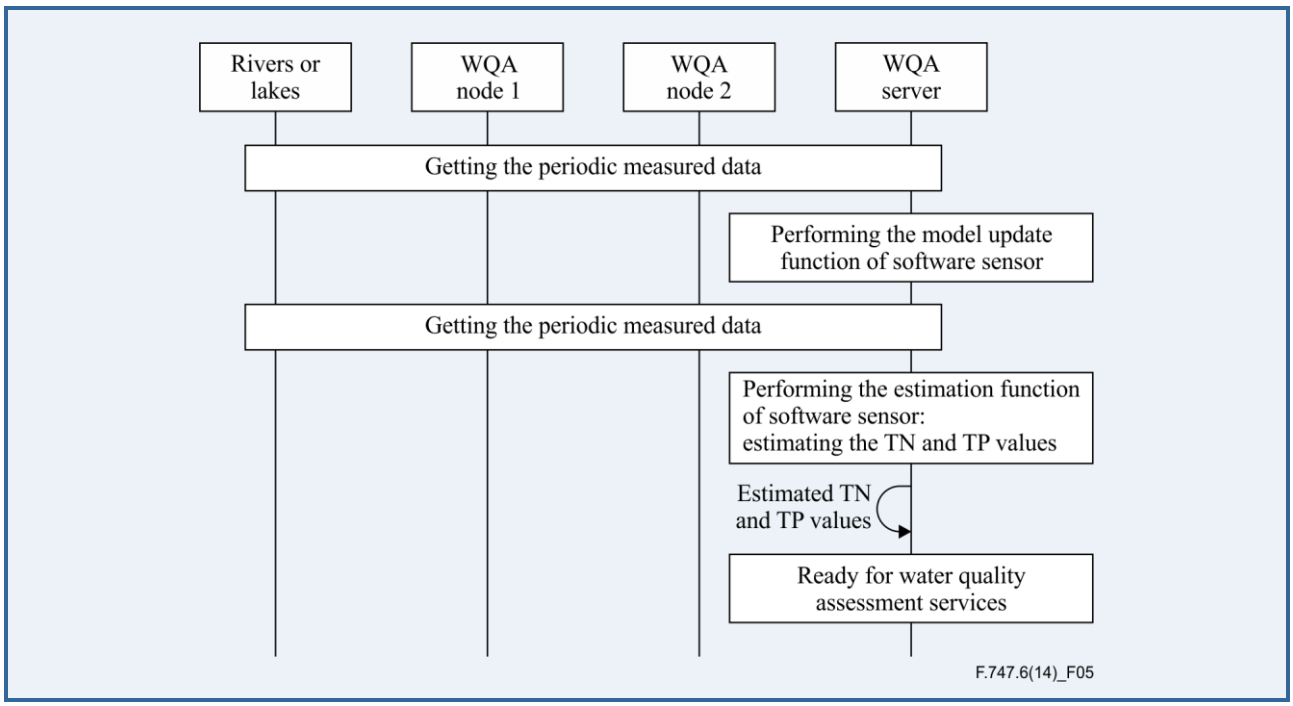


Figure 5 – Water quality prediction through software sensors

8 Requirements of WQA services

8.1 Reliable data transfer

Because the WQA services such as dispersion of the water pollution, tracking of the water pollutant source and the prediction of the water quality change are using the values measured from the WQA nodes, reliable data transfer without any data loss from WQA node to WQA server is required.

8.2 Real-time water quality information transfer

The WQA information including sensing data, aggregated data, control data, the results of water quality analysis, etc., is required to deliver to the WQA server in real-time. It allows the public authorities to monitor the information and to handle the water pollutant accident in real-time. It also satisfies the user's desires to receive the information in prompt.

8.3 Bidirectional communication

While the WQA node offers information to the WQA server periodically, the WQA server delivers control information to the WQA node. Thus, the bidirectional communication is recommended to support the smooth operation of the WQA system.

8.4 Security

Security services are required, for example, to protect the integrity, delivery and confidentiality of water quality data, in order to provide accurate WQA.

8.5 Water assessment modelling

Precise water pollution dispersion, prediction and water pollutant source tracking by using the measured data are required. To achieve this and to provide WQA services accurate modelling of WQA is essential.

9 USN-based WQA services

Ubiquitous sensor network (USN)-based WQA services are provided by the analysis of the data measured from sensors within the sensor network. A WQA server, using sensing data, assesses changes in water quality and performs water quality monitoring through the aggregated sensing data in real-time.

The sensor network consists of WQA nodes with sensors and a gateway. Sensors measure the value of the water quality parameters (e.g., pH, DO, EC, oxidation-reduction potential (ORP)), the flow velocity, the water level, etc. The WQA nodes aggregate the sensing data from various sensors and send it to the USN gateway. The USN gateway also sends the aggregated data to the WQA server where the WQA is performed. Therefore, the WQA system provides following WQA services.

9.1 Water quality distribution service

The water quality distribution service provides water quality distribution map of rivers, lakes, ponds, etc. in real-time where WQA nodes are installed. For the service, the WQA server calculates estimated values of TN and TP in real-time using a software sensor. The WQA server offers water quality distribution in real-time for water temperature, turbidity, pH, TN or TP of middle and small-sized rivers.

9.2 Water quality prediction service

The water quality prediction service provides water quality prediction values of rivers, lakes, ponds, etc. in real-time where WQA nodes are installed. For the service, the WQA server calculates estimated values of TN and TP in real-time using a software sensor. The WQA server offers prediction of water quality change in real-time for TN or TP of medium and small-sized rivers.

9.3 Service for total amount of polluted water

The total amount of polluted water service provides the total amount of the polluted water of rivers, lakes, ponds, etc. in real-time where WQA nodes are installed. For the service, the WQA server calculates estimated values of TN and TP in real-time using software sensor. The WQA server offers the total amount of water pollution in real-time for flow velocity, water level, TN or TP of medium and small-sized rivers.

10 USN capabilities for WQA services

10.1 Reliable communication link in sensor networks

The sensor networks are deployed in a large area in an outdoor water environment. Even if they have poor radio environment, reliable delivery of sensing data is required. That is, the WQA system offers an optimal communication link considering its operation and the characteristics of the sensing information. It also has to guarantee high end-to-end success rate of the data transmission.

10.2 Transmission delay guarantee to the WQA server

The measured data from sensors is delivered in real-time to the WQA server. For that, the transmission delay (for example, data processing time and transmission time in the sensor networks) is guaranteed. The transmission delay over the Internet is also guaranteed.

10.3 Low power consumption in sensor networks

The sensor network needs to provide uninterruptible power for long-term unmanned operation in a broad area. It must have sensors or sensor nodes that consume low electric power. It also has to use low power mechanisms or its own power supply.

10.4 Bidirectional communication between WQA nodes and servers

In general, the data in sensor networks are transferred in an upward direction. However, the control data, which are transferred in a downward direction, are needed to control the sensors and USN gateways for the unmanned long-term operation in WQA system. For example, Internet protocol (IP) infrastructure is a good candidate to support the bidirectional communication.

10.5 Multi-hop data transfer in sensor networks

The sensor network is constructed with an almost linear topology and so the proper multi-hop networking protocol must be considered. WQA nodes must be able to deliver the measured data to the USN gateway through multi-hop paths.

10.6 IP infrastructure compatibility

For the interworking between WQA nodes and WQA servers or between WQA servers and users, IP based networking is considered with the support of Internet protocol version four/Internet protocol version six (IPv4/IPv6) translation.

Along with IP networking, the network management protocol is considered to monitor and operate the WQA nodes in real-time.

10.7 Long distance transmission support in sensor networks

A sensor network may require long distance data transmission (for example, several kilometres) between WQA nodes and USN gateways. Proper communication distance in large area (for example, dozens of kilometres) outside must be guaranteed.

10.8 Security services

The security services are required to protect the delivery of water quality information; to protect data confidentiality and integrity among WQA nodes; to provide authentication between WQA nodes and USN gateways; to provide data confidentiality between the USN gateway and WQA server; and to preform authorization, verification, etc.

10.9 Data logging

The gateway keeps a record of the measured data to prevent data loss due to communication interruptions between the USN gateway and WQA server, server faults, the flooding of WQA nodes, etc.

10.10 Maintainability of sensor networks

The network components in sensor network must be observed and faults must be detected automatically during unmanned long-term operation. Furthermore, the current network status can be optionally reported to the WQA server through the USN gateway. For the prevention of bio-fouling, the sensor control, including changing the data sensing period and operation of the sensor wiper, may be optionally controlled automatically.

10.11 Naming and addressing in sensor networks

The WQA system has two key features: real-time support and direct control of sensors. To support these features, each WQA node with sensors is required to be uniquely distinguished by the naming and addressing method. Here, the naming is recommended to have the relation to river name, zone improvement plan (ZIP) code and others.

Y.4108/Y.2213

NGN service requirements and capabilities for network aspects of applications and services using tag-based identification

NGN service requirements and capabilities for network aspects of applications and services using tag-based identification

Summary

Recommendation ITU-T Y.2213 describes high-level service requirements and NGN capability requirements needed to support applications and services using tag-based identification. Several examples of applications and services using tag-based identification are also described with scenarios. The scope of this Recommendation is limited to applications and services using tag-based identification and they are distinguished by the following three mandatory elements: ID tag, ID terminal and identifier.

Source

Recommendation ITU-T Y.2213 was approved on 12 September 2008 by ITU-T Study Group 13 (2005-2008) under Recommendation ITU-T A.8 procedure.

Keywords

Bar code, capabilities, capability requirements, ID tag, ID terminal, identifier, identifier resolution, RFID, service requirements and tag-based identification.

Table of Contents

Introduction

Numerous applications have already incorporated identifiers to support differentiated services, where identifiers are mainly used for logical identification. Emerging applications, whose examples are described in Appendices II and III, require also that logical identification be associated with physical objects. This Recommendation expects that a relationship between physical and logical objects is maintained and assumes that applications and services using tag-based identification are based on such a relationship. This relationship makes a wide variety of business opportunities available: for example, passport numbers may be applied to physical objects for identification and each physical object, i.e., passport, may be associated with a logical object such as a text content, application program, mobile executable code, or data record.

Applications and services using tag-based identification adopted and deployed widely in various business fields have used identifiers as a means for identifying physical and logical objects. Consequently, their market volume is being expanded dramatically. In addition to use cases of identifier in business fields, physical and logical objects with an identifier can enable applications and services using tag-based identification in consumer fields as well.

Existing applications and services using tag-based identification have been exploited in business-to-business (B2B) fields, but currently they are expanding to business-to-consumer (B2C) and business-to-business-to-consumer (B2B2C) fields as shown in clause II.3. Both B2B and B2C/B2B2C tag-based identification applications and services basically have similar service requirements. Additionally, B2C/B2B2C-specific service requirements and related challenges for NGN are also examined in this Recommendation, such as privacy management, identifier resolution, management, etc.

Identifier information in such applications and services using tag-based identification works as the key information used to retrieve related contents, execute certain application programs, point to a specific information resource, make an association with a specific object, and so on. The identification feature supporting hyperlink-like access will enable NGN applications and services to be performed and provided to consumers in easier and more convenient ways. For example, consumers do not have to type a URL in a phone user interface because the tag-based identification supports that automatically.

Since a number of applications and services using tag-based identification are expected to be deployed widely in NGN, the NGN architecture framework and the NGN functional entities have to support them.


N

Recommendation ITU-T Y.4108/Y.2213

NGN service requirements and capabilities for network aspects of applications and services using tag-based identification

1 Scope

This Recommendation covers extensions to NGN release 1 capabilities in order to support tag-based identification applications and services in the NGN environment and builds upon [b-ITU-T Y.Sup1] and [ITU-T Y.2201].

The scope of this Recommendation is limited to tag-based identification applications and services distinguished by the following three mandatory elements: ID tag, ID terminal and Identifier. These applications and/or services use identifiers only read from ID tags by ID terminals.

This Recommendation covers:

- description and scope of tag-based identification applications and services with some example scenarios;
- high-level service requirements of tag-based identification applications and services; and
- extended or new NGN capabilities based on the high-level service requirements.

Functional requirements and related NGN architecture extensions for support of the described capabilities are out of scope of this Recommendation.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T X.668] Recommendation ITU-T X.668 (2008) | ISO/IEC 9834-9:2008, *Information technology – Open Systems Interconnection – Procedures for the operation of OSI Registration Authorities: Registration of object identifier arcs for applications and services using tag-based identification*.

[ITU-T Y.2201] Recommendation ITU-T Y.2201 (2007), *NGN release 1 requirements*.

[ITU-T Y.2701] Recommendation ITU-T Y.2701 (2007), *Security requirements for NGN release 1*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 application network interface (ANI) [b-ITU-T Y.2012]: Interface which provides a channel for interactions and exchanges between applications and NGN elements. The ANI offers capabilities and resources needed for realization of applications.

3.1.2 end user [b-ITU-T M.3050.1]: The actual user of the products or services offered by the enterprise. The end user consumes the product or service.

3.1.3 identifier [b-ITU-T Y.2091]: A series of digits, characters and symbols or any other form of data used to identify subscriber(s), user(s), network element(s), function(s), network entity(ies) providing services/applications, or other entities (e.g., physical or logical objects). Identifiers can be used for registration or authorization.

NOTE – Identifiers can be either public to all networks, shared between a limited number of networks or private to a specific network (private identifiers are normally not disclosed to third parties).

3.1.4 name [b-ITU-T Y.2091]: The identifier of an entity (e.g., subscriber, network element) that may be resolved/translated into an address.

3.1.5 service [b-ITU-T Z.100]: A set of functions and facilities offered to a user by a provider.

3.1.6 user [b-ITU-T Y.2091]: It can be an end user, a person, a subscriber, a system, an equipment, a terminal (e.g., FAX, PC), a (functional) entity, a process, an application, a provider, or a corporate network.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 associated information: The information which is associated with an identifier.

NOTE – Example associated information instances are URL, URN, IP address, E.164 number, etc.

3.2.2 B2B tag-based identification applications and services: This term refers to tag-based identification applications and services based on business relationships which involve exchanges of identification information between business partners.

3.2.3 B2B2C tag-based identification applications and services: This term refers to tag-based identification applications and services based on integrated business relationships of B2B and B2C which involve exchanges of identification information.

3.2.4 B2C tag-based identification applications and services: This term refers to tag-based identification applications and services based on business relationships which involve exchanges of identification information between business and consumer.

3.2.5 C2C tag-based identification applications and services: This term refers to tag-based identification applications and services based on business relationships which involve exchanges of identification information between consumers.

3.2.6 forward identifier resolution: A function to resolve an identifier into an associated information.

3.2.7 identifier resolution: A function to resolve an identifier into associated information (see "Forward identifier resolution") and vice versa (see "Reverse identifier resolution").

3.2.8 identifier scheme: It is a numbering scheme that specifies the format and structure of the identifiers used within that scheme.

3.2.9 ID tag: A physical object which stores one or more identifiers and optionally application data such as name, title, price, address, etc.

NOTE – It may have a communication capability with an ID terminal depending on implementations.

3.2.10 ID terminal: A device with a data reading and optional writing capability which reads (and optionally writes) identifier(s) and optionally application data from/into an ID tag.

NOTE – The data reading (and optionally writing) capability depends on implementations.

3.2.11 reverse identifier resolution (or backward identifier resolution): A function to resolve an associated information into a corresponding identifier. It is the reverse operation of the forward identifier resolution.

3.2.12 tag-based identification: The process of specifically identifying a physical or logical object from other physical or logical objects by using identifiers stored on an ID tag.

3.2.13 tag-based identification applications and services: Applications and services which use tag-based identification.

3.2.14 tag-terminal interface: A communication interface between ID tag and ID terminal. The ID terminal reads identifier(s) and optionally application data from the ID tag and/or writes them into the ID tag.

NOTE – The interface medium may be infrared, RF, camera, optical scanner, and galvanic current. Communication techniques depend on the interface medium.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

1D	1-Dimension
2D	2-Dimension
ANI	Application Network Interface
B2B	Business-to-Business
B2B2C	Business-to-Business-to-Consumer
B2C	Business-to-Consumer
C2C	Consumer-to-Consumer
DB	Database
DNS	Domain Name System
ENUM	TElephone NUmber Mapping
FAX	Facsimile
GPS	Global Positioning System
ID	Identification
IP	Internet Protocol
IrDA	Infrared Direct Access
ISBN	International Standard Book Number
LAN	Local Area Network
MAC	Medium Access Control
NGN	Next Generation Network
OECD	Organisation for Economic Co-operation and Development
OID	Object Identifier
PC	Personal Computer
PII	Personally Identifiable Information
POS	Point Of Sale
QoS	Quality of Service
RF	Radio Frequency
RFID	Radio Frequency Identification

SCM	Supply Chain Management
TCP	Transmission Control Protocol
TTI	Tag Terminal Interface
UNI	User Network Interface
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
URN	Uniform Resource Name
WAP	Wireless Application Protocol

5 Conventions

None.

6 Tag-based identification applications and services description and high-level reference service architecture

There are many applications and services which use identifiers over various operation layers. Here are examples: MAC address is a data link layer identifier; IP is a network layer identifier; port number is a transport layer identifier; cookie identifiers associated with web applications are session layer identifiers; and passport numbers are application layer identifiers.

This Recommendation does not cover all the applications and services using identifiers but is limited to applications and services using tag-based identification. Clause 6.1 describes the basic characteristics of tag-based identification applications and services covered in this Recommendation. Clause 6.2 describes some expected impacts on NGN. Clause 6.3 describes a high-level reference service architecture model.

6.1 Basic characteristics of tag-based identification applications and services

Use of identifiers in this Recommendation involves consideration of the following four elements:

- identifier;
- ID tag;
- ID terminal; and
- associated information.

Every tag-based identification application and service is described using the following three mandatory elements: Identifier; ID tag; and ID terminal. Applications and services which are not based on these three elements are out of scope of this Recommendation. The associated information element is optional.

Using this approach, RFID is a wireless communication technology used to transmit the identifiers stored in an RFID tag to an RFID terminal. Where the RFID tag is an ID tag, the RFID terminal is an ID terminal, the identifier stored in the RFID tag is just an identifier, and the information associated with the identifier is associated information.

6.1.1 Identifier

The identifier is described in clause 3.

6.1.2 ID tag

The identifier is required to be stored on the ID tag. The ID tag, which acts as a storage medium, is required to be read by the ID terminal. Examples of ID tags are 1D or 2D barcode tags, smartcards and RFID tags.

6.1.3 ID terminal

The ID terminal is required to read the identifier stored on the ID tag and the identifier is required to be transmitted to the ID terminal. The ID terminal is required to have a data reader. Examples of data reading techniques are RF, camera, optical scanner, IrDA, galvanic wire-line, manual key-in, etc.

Additionally, the ID terminal may have a capability to write identifier(s) and optionally application data on ID tags.

6.1.4 Associated information

The identifier may be associated with application/service-related information. Such associations may be maintained by a directory service. An identifier may have multiple associations.

6.2 Impact of tag-based identification applications and services on the network

According to the assumption of growth of telecommunications data traffic, transactions are expected to increase significantly when wide adoption of tag-based identification applications and services are achieved. On the other hand, even though total traffic will show dramatic growth during the course of dissemination of broadband services, traffic generated by tag-based identification applications and services will make up only small part of the total traffic. That is, tag-based identification applications and services will show a tremendous number of transactions but with a relatively small amount of traffic. Such transaction increases have various impacts on NGN.

Network resource management is necessary to cope with such transaction increases generated by tag-based identification applications and services, and to avoid access concentration on specific resources, in addition to the appropriate traffic management of NGN.

6.3 Reference service architecture model

Tag-based identification applications and services are provided through NGN as shown in Figure 1. Three interfaces called TTI, UNI and ANI are involved; however, the TTI is out of scope of NGN. Tag-based identification applications and services are provided to end users through the following three operations with the above interfaces: identifier reading, forward identifier resolution and information access from ID terminal's point of view.

An ID terminal reads identifier(s) from an ID tag via TTI. It requests forward identifier resolution(s) and receives corresponding associated information. If the associated information received is a URL, the ID terminal accesses the information content pointed by the URL. Thus, the forward identifier resolution is followed by a content retrieval operation through NGN.

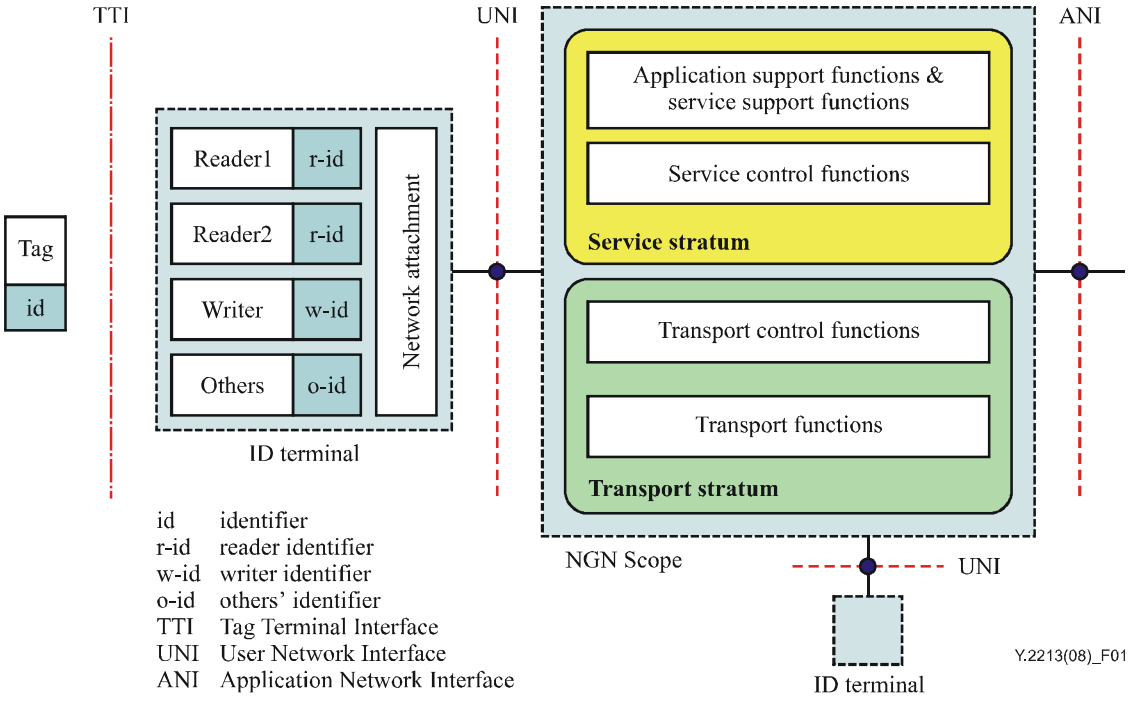


Figure 1 – Reference service architecture model

7 Service requirements of tag-based identification applications and services

The following are high-level service requirements for tag-based identification applications and services. These requirements identify the required extensions to the set of NGN release 1 capabilities. Requirements in Appendix I do not impact NGN capabilities and they are provided for informative purposes in order to develop tag-based identification applications and services over the NGN environment.

7.1 Multi-identifier interpretation requirements

An identifier constituted by sub-identifier elements is required to be interpreted into sub-identifiers by its structure information. An ID terminal is required to contain the structure information of the supported identifiers.

Tag-based identification applications and services have the following requirement to support multi-identifier interpretation:

- identifier structure information may be provided to an ID terminal.

7.2 Identifier resolution

Tag-based identification applications and services have the following identifier resolution requirements:

1) it is recommended to provide forward identifier resolution from an identifier to associated information;

2) support of a reverse identifier resolution from associated information to a corresponding identifier may be provided (see Note 1);

NOTE 1 – The reverse identifier resolution provides a means to find a specific identifier of an object and its location in the physical world, from the associated information of the object. This is like searching for a book and its location in a library by using the library catalogue.

3) it is recommended to support one-to-one association between an identifier and associated information;

4) support of one-to-many associations between an identifier and associated information instances may be provided (see Note 2);

NOTE 2 – The one-to-many associations enable different usages of an identifier among users of the identifier. For example, manufacturers may use identifiers for production planning and material/parts inventory control while retailers may use the same identifiers for store inventory management, out of stock alerts, anti-theft, etc., and consumers may use the same identifiers for product information retrieval.

5) different associated information for an identifier may be resolved according to usage context of the identifier, for example, who/when/where/why, which is called context-dependent forward identifier resolution (see Note 3).

NOTE 3 – Even though the same identifiers may be read by different users, each user is expecting different services and/or different information. Different services and different information are usually managed by different resources in different network locations. Therefore, the context-dependent forward identifier resolution is necessary to support this scenario. This forward identifier resolution will respond with different network addresses according to the context even though the same identifier is read.

7.3 ID terminal and ID tag management

Tag-based identification applications and services have the following requirements for remote management of ID terminals and ID tags:

1) it is recommended to manage ID terminals so as to read identifiers and relevant information from ID tags;

2) it is recommended to manage some types of ID tags' operation mode (e.g., active RFID tags may be booted up and/or put in sleeping mode remotely);

3) it is recommended to manage ID terminal and ID tag aspects such as:
   - radio operations;
   - networking operations;
   - software update;
   - time synchronization;
   - device identifiers (e.g., r-id, w-id and o-id in Figure 1);
   - identifier structure information;
   - filtering rules update; and
   - location registration and management;

4) when ID terminals and ID tags are managed locally, ID terminals and ID tags may report management information such as that related to the aspects listed above.

7.4 Content distribution control

Tag-based identification applications and services have the following requirement for content distribution control:

- it is recommended that facilities for control of information content distribution be supported (to accommodate possible commercial, regulatory and privacy requirements).

7.5 Privacy management

Some of the NGN services may have privacy impact for the users. Privacy invasions caused by usage of identifiers, ID tags and ID terminals may be summarized as follows: association threats, location threats, and information leakage threats. As an example, RFID technology may facilitate access to information pertaining to the merchandise that individuals wear and/or carry, and may

create an opportunity for abuse of this information such as tracking an individual's location or invading his/her privacy.

Tag-based identification applications and services have the following requirements:

1) it is recommended to meet the guidelines described in [b-OECD-Privacy-Protection];

NOTE 1 – National situations may add other guidelines and obligations.

2) it is recommended to provide a privacy protection capability as specified in [ITU-T Y.2201];

3) it is recommended to manage PII protection policies in consistency with the general approach for security in NGN [ITU-T Y.2701]; and

4) the security approach for support of tag-based identification applications and services is recommended to be consistent with the general approach for security in NGN [ITU-T Y.2701].

NOTE 2 – As the variety of NGN services is very wide, it is not possible to define good practices for all the possible situations. Furthermore, privacy impacts of a particular service may not be obvious. So, it is recommended to do a privacy impact assessment to identify privacy risks and take measures to mitigate them.

7.6 Location-based services support

Tag-based identification applications and services have the following requirements:

1) location information of ID terminal and/or ID tag is recommended to be registered (either statically or dynamically);

2) location information of ID terminal and/or ID tag may be provided if requested by tag-based applications and services.

7.7 Service quality control

There are many tag-based identification applications and services for business and consumer purposes. Some of them can be combined and interworked via various business agreements and partnerships. Such tag-based identification applications and services may have different service quality requirements.

Tag-based identification applications and services have the following requirement on service quality control:

- it is recommended to provide different service qualities according to service quality requirements.

7.8 Application transaction and traffic requirements

Tag-based identification applications and services place the following requirements on both NGN and application/service provider's resources:

1) it is required to manage the transaction volume generated by tag-based identification applications and services; and

2) it is recommended to be able to avoid access concentration to single resources.

8 NGN capabilities for tag-based identification applications and services

Tag-based identification applications and services use NGN release 1 capabilities but require some extended or new capabilities. The requirements given below are provided from a high-level perspective and are not intended to constitute precise functional requirements for NGN entities.

8.1 Requirements for extensions or additions to NGN release 1 capabilities

Based on the high-level service requirements described in clause 7, this clause specifies requirements for extensions or additions to NGN release 1 capabilities.

8.1.1 Multi-identifier interpretation

Based on the service requirements in clause 7.1, the following requirements are required to support the multi-identifier interpretation:

1) NGN is required to support the OID-based identification scheme specified by [ITU-T X.668] in order to distinguish identifier schemes unambiguously for tag-based identification applications and services; and

2) NGN may provide structure information of identifier schemes.

8.1.2 Identifier resolution

Based on the service requirements in clause 7.2, the following requirements are recommended to support identifier resolution capabilities of NGN:

1) NGN is recommended to provide forward identifier resolution;

2) NGN may provide reverse identifier resolution;

3) NGN may provide identifier resolution of multiple associations between an identifier and multiple associated information instances;

4) NGN may provide context-dependent forward identifier resolution, depending on usage context of identifiers;

5) NGN identifier resolution is recommended to scale in order to handle increased demand for identifier resolutions; and

6) NGN identifier resolution is prohibited from being affected by a single point of failure.

8.1.3 Privacy management

Based on the service requirements in clause 7.5, tag-based identification applications and services place the following requirement on NGN:

- NGN is recommended to adopt good privacy practices and provide a PII protection capability.

This capability is recommended to be consistent with the general approach for security in NGN [ITU-T Y.2701].

8.1.4 Content distribution control

Based on the service requirements in clause 7.4, tag-based identification applications and services place the following requirement on NGN for control of content distribution:

- NGN is recommended to support capability for content distribution control.

8.1.5 Device management

Based on the service requirements in clause 7.3, following requirements are recommended to support remote device management of ID terminals and ID tags:

1) NGN is recommended to support remote device management of ID terminal and ID tag aspects such as those listed in clause 7.3; and

2) NGN may receive updated management information of ID terminal and ID tag aspects such as those listed in clause 7.3.

8.1.6 Profile management

8.1.6.1 User profile

Based on the service requirements in clause 7.5, tag-based identification applications and services may use NGN release 1 user profile management capability with some extensions of user profile attributes as follows:

- NGN is recommended to support user profile management satisfying privacy management requirements.

8.1.6.2 Device profile

Based on the service requirements in clause 7.3, tag-based identification applications and services may use NGN release 1 device profile management capability with some extensions of device profile attributes as follows:

- NGN is recommended to support device profile management enabling remote management of ID terminal and ID tag aspects such as those listed in clause 7.3.

8.1.7 Quality of service

The transaction and traffic-related requirements in clause 7.8 place these additional requirements on NGN QoS capabilities:

1) NGN is required to support QoS capabilities to sustain the transaction volume caused by tag-based identification applications and services; and
2) NGN is recommended to support QoS capabilities to be able to avoid access concentration to single resources (e.g., identifier resolution).

8.2 Requirements supported by existing NGN release 1 capabilities

Based on the high-level service requirements described in clause 7, this clause specifies requirements supported by existing NGN release 1 capabilities.

8.2.1 Service quality control

The service quality control requirement specified in clause 7.7 is supported by existing NGN release 1 QoS capabilities.

8.2.2 Location management

NGN release 1 provides location management capability which determines and reports information regarding the location of users and devices within NGN.

The requirements in clause 7.6 are supported by existing NGN release 1 location management capability.

Appendix I

Non-NGN high-level service requirements

(This appendix does not form an integral part of this Recommendation)

This appendix describes high-level service requirements which enable new application/service provider functions and which will not affect the functional capabilities of the NGN. These requirements apply to tag-based identification applications and services. [b-ITU-T F.771] includes some service requirements from the point of view of multimedia information access.

I.1 General requirements for identifiers

An identifier is a fundamental component of tag-based identification applications and services. Applications and services may be triggered by identifiers. Identifiers have the following requirements:

1) identifiers are required to be unique, i.e., the identifier scheme is required to allow for uniqueness of identifiers;
2) identifiers may have a life time;

   NOTE – The lifetime requirement depends on applications and services.
3) identifiers may be managed by an identifier lifecycle management capability;
4) identifiers may be validated. For example, life time or revocation of identifiers may require validation of the identifiers;
5) identifiers may be used for registration or authorization [b-ITU-T Y.2091];
6) identifiers may be either "public to all networks", "shared between a limited number of networks", or "private to a specific network" [b-ITU-T Y.2091]; and
7) an identifier may be used by more than one application/service.

I.2 Requirements for identification of identifier schemes

Identifiers may be assigned using different identifier schemes and the identifiers created are required to allow for unique identification of the scheme used to create the identifier.

ID terminals of tag-based identification applications and services have the following requirement:

- identifiers assigned by a certain identifier scheme are required to be distinguishable from identifiers assigned by other identifier schemes, because NGN may handle various identifier schemes.

I.3 Requirements related to application data encoding

Identifier information may be encoded optionally with other application data like title, name, price, etc., into an ID tag. The following requirement allows for application data processing:

- application data is required to be encoded in a standardized way on ID tags that are used by tag-based identification applications and services.

For example, tag-based identification service providers can be retail shops, pubs, restaurants, taxi drivers associations, museums, galleries, movie production companies, and so on. They may want to provide enhanced and differentiated services to their consumers, which might require varying amounts of information to be captured as application data items, e.g., price, name, title, or location to be stored locally on the ID tag. To make this possible, application data needs to be stored in a standardized format (e.g., [b-ISO/IEC 15961] and [b-ISO/IEC 15962]).

I.4 Requirements for identification service interworking

Various tag-based identification applications and services have already been deployed with their own service network and without cooperative relationships because an identifier has been implemented in various fashions (e.g., ID tag types such as 1D or 2D barcode, RFID tags, contact or contactless smart cards) and service networks have been established separately. The following requirements are intended to facilitate interworking among these service networks:

- an interworking capability is recommended to be provided to allow interworking of different tag-based identification applications and services.

Following example application cases are possible:

- B2C tag-based identification applications and services can be combined with B2B into B2B2C application and service models.
- Bar code-triggered and RFID-triggered tag-based identification applications and services might be established independently but they may correspond to the same object. Integration requires interworking between these two tag-based identification applications and services.

I.5 Requirements for location information management

Location-based application models need location information. In particular, they need location information with regard to a device. For example, applications may require ID terminal location information or ID tag location information, and will know who is to use this location information.

With regard to use of location information, the ID terminal and ID tag locations can be usually assumed as being the same, because the user has to read ID tags usually within a few metres at a maximum. However, how to get the location information for the ID terminal and ID tag is different because:

1) user location, that is an ID terminal location, can be provided with an accuracy to about 10 metres by using GPS-based solutions; to a few hundreds of metres by using cell information from cellular networks; and variable accuracy depending on access network technologies used, as specified in [ITU-T Y.2201]; and
2) ID tag location may be included in the ID tag itself, or retrieved from a service provider via an identifier.

I.6 Requirements related to management of application mobility

Tag-based identification applications and services have the following management of application mobility requirement:

- application mobility is recommended to be provided among different tag-based identification applications and services, because an ID tag might be moved among them with a requirement that their communication associations be retained.

Application mobility means a communication association is handed over to other applications. A typical example is given for a transportation application. A sophisticated transportation system may charge for a single fare for the route from A to C via B where two transportation fare applications are associated from A to B and B to C. So, the association of a fare application between A and B has to be handed over to the other fare application between B and C in order to support the single fare association.

I.7 Requirements related to traceability

Traceability relates to object traceability and usage traceability. The object traceability can be seen as syntactic tracking of an object so every read point during the life time of the object can be traced regardless of application-perspective usages.

The usage traceability can be seen as semantic tracking of an object so how the object has been treated and what the object has been used for could be traced.

Tag-based identification applications and services have the following traceability-related requirements:

- it is recommended to provide the information on what ID terminals have read an ID tag for an object; and
- it is recommended to provide the information on what tag-based identification applications and services have read an ID tag for an object.

I.8 Requirements related to identifier filtering

Tag-based identification applications and services have the following filtering requirements:

- users, applications modules, middleware functions, or lower-layer read functions do not have to process unnecessary ID tags or identifier schemes. Proper filtering is recommended to be provided.

Appendix II

Classification of tag-based identification applications and services

(This appendix does not form an integral part of this Recommendation)

This appendix describes an application and service trend toward the social infrastructure, classification of relevant applications and services and some typical applications and service examples. It also describes how to identify tag-based identification applications and services that are covered by this Recommendation, by evaluating them with respect to the key elements identified in clause 6.1.

II.1 Overview of tag-based identification applications and services

Tag-based identification applications and services have already been widely adopted as business applications since its innovation. Recently, B2B tag-based identification applications and services have rapidly evolved.

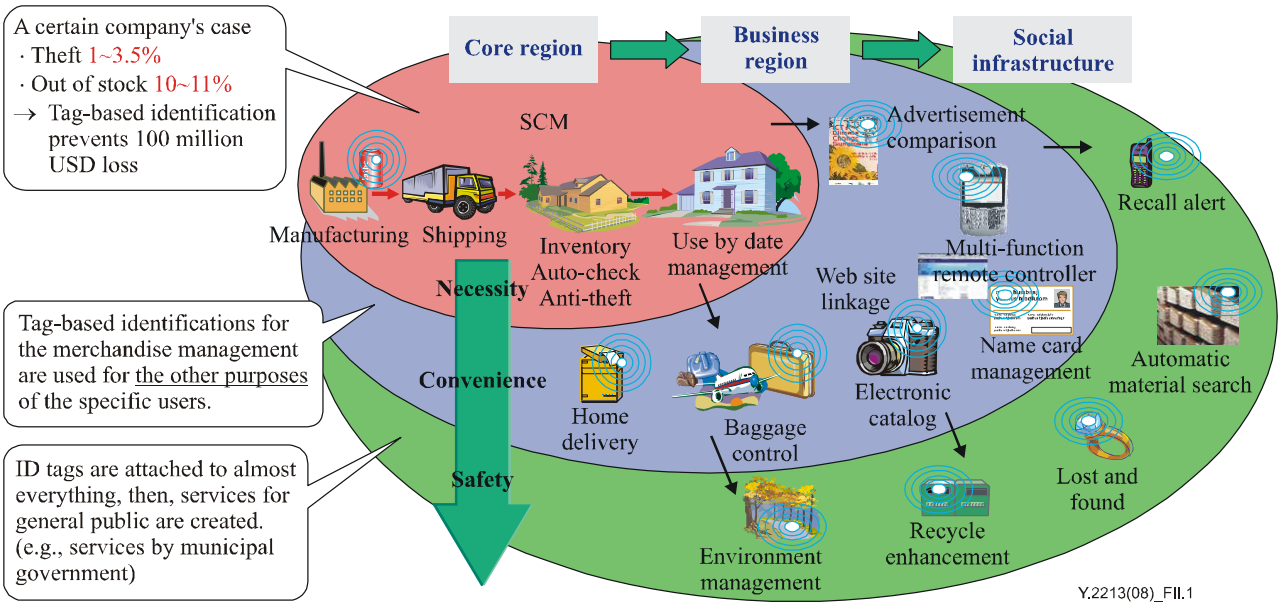


Figure II.1 – Tag-based identification application development toward social infrastructure

Figure II.1 shows a development model of tag-based identification applications and services from an application perspective. B2B applications are realized at first because they may be expected to justify the large investment of the initial implementations. An industrial entity (in many cases, the manufacturer of an object) starts the assignment of identifier to an object and the attachment of an ID tag to that object. For example, many entities are going to use the same identifier and ID tag for different purposes throughout the supply chain management (SCM). Manufacturers use these identifiers and ID tags for production control; transport companies for shipping control; wholesalers for inventory control; and retailers for inventory control; auto check-out; and anti-theft measures, respectively. This "core region" of tag-based identification applications adopted by a relevant industry is facilitated by the business necessity.

Once ID tags for objects are widely adopted by industries and resolving functions from identifiers to services and relevant databases are commonly used, many new derivatives of the "core region" applications are created by third parties for B2C/B2B2C applications; these are identified as "Business Region" in Figure II.1. The same identifiers and ID tags applied by the industrial entities

(i.e., manufacturers) are not only used for the primary industrial purposes (e.g., SCM), but also for the benefits of consumers (e.g., electronic information (instruction manual and/or catalogue) delivery of the product triggered by an identifier, home delivery management of the physical object, and so on). The convenience for consumers becomes the driving force for these new B2C/B2B2C applications.

In the further adoption of tag-based identification applications, it works for the improvement of social benefit. For example, information on the hazardous material used in consumer products is easily obtained by knowledge of the identifiers, which promotes enhancement of recycling and reduction of the environmental burden by the proper disposal by consumers. Tag-based identification applications are also quite effective for recall alerts and collection of the recalled products. In addition, ID tag-equipped "smart medicine" enables automatic alarms, when any ill effects occur with simultaneous usage of two medicines. In this stage, tag-based identification applications are incorporated into the social infrastructure.

Throughout the development to the wider usage of the tag-based identification applications and services in Figure II.1, interoperability among B2B and B2C/B2B2C tag-based identification applications and services is an essential issue and the same or interoperable technical standards are utilized by multiple entities for multiple purposes.

II.2 Classification of tag-based identification applications and services

Tag-based identification applications and services can be classified according to two viewpoints: business relationships and usage types.

From the business relationship viewpoint, tag-based identification applications and services can be classified into closed domain, B2B, B2C, B2B2C and C2C categories.

From the usage type viewpoint, tag-based identification applications and services can be classified into reader-based and tag-based categories. The tag-based category does not apply to B2B applications.

The reader-based type refers to the case where a user terminal is equipped with an identifier reader which is used to read an identifier from an ID tag. But the tag-based type refers to the case where a user terminal has an attached ID tag and other readers read an identifier from the terminal's ID tag.

II.3 Examples of tag-based identification applications and services

The feature of identification has been used in a barcode form for a long time for business purposes in retail shops, logistics, automotive and aviation, pharmaceutical industries, inventory management, manufacturing and processing, supply chain management, transportation, etc.

Such identification is used:

- to identify products at retail shops; and
- to identify and route products, monitor delivery paths of products, exchange product-related information between business partners, and so on.

The former type corresponds to a single enterprise scope without exchanging real-time business information between business partners. So, it can be just called a tag-based identification business application. Typical examples include:

- inventory management;
- security and access control;
- agriculture;
- library;
- parking management.

The latter type corresponds to a business relationship among multiple business partners. So, it can be called a B2B tag-based identification application because identification information is exchanged in a B2B manner. Typical examples include:

- supply chain management;
- food chain management;
- manufacturing and processing;
- transport and logistics;
- pharmaceutical industry.

Additionally identifier users may be human consumers. Information services and value-added services may be provided to consumers when the identifier information is processed. Such tag-based identification applications are called B2C tag-based identification applications. Examples of these include:

- personal welfare and safety;
- sports and leisure;
- bus and subway route search;
- advertisement and detail information retrieval;
- on-line shopping;
- payment;
- drug safety;
- location-based information service.

Additionally a business partnership may integrate B2B with B2C applications so that B2B2C business associations may be developed. Examples of B2B2C tag-based identification applications include:

- food chain information service;
- home delivery service;
- product origin check.

Moreover C2C applications are also possible. A typical example is the case where a cell phone user uses his cell phone, which contains both an ID terminal and an ID tag, to read an identifier and application data from other cell phones. For example, one of two movie tickets, which a man bought and of which associated data was stored in his cell phone, could be transferred to a cellular phone of his girl friend. In this example, a movie ticket transfer can be performed directly between two terminals without the use of any network communication. Additionally, a movie ticket exchanged and displayed on a receiver's screen can have embedded advertisement information content which was provided via a network operation. Resultant applications will depend on implementations. So, C2C interactions may also be involved in, and require network functions.

Example scenarios of tag-based identification applications and services are described in Appendix III.

II.4 Evaluation of tag-based identification applications and services

This clause provides examples of how to evaluate applications and services in the context of the key elements, so as to determine if they are covered by this Recommendation.

Table II.1 – Evaluation of tag-based identification applications and services

Application or service	Evaluation in the context of the key elements
General telecommunication service in which an E.164 identifier for the called party is used.	Identifier: E.164 ID tag: Not applicable: the identifier is stored in human brain, a phone book, an electronic form, or otherwise. ID terminal: Applicable: it is the telephone. The identifier is transmitted via manual key-in or dialing at the telephone. Associated information: Applicable: an E.164 identifier can be mapped to a set of information resources via the ENUM technique so that an identifier resolution process may be followed. Conclusion: [Invalid] As all of the three mandatory features are not present, it is not a tag-based service.
In the above service scenario an old father uses an identifier reader-equipped phone or terminal; an ID tag embedding an E.164 phone number for his son is affixed to a photograph for the son; he aims or touches his phone or ID terminal at/to the photograph; and then he gets an automated calling to his son.	Identifier: E.164 ID tag: Applicable: an ID tag containing the identifier is affixed to the photograph. ID terminal: Applicable: it is the phone or the ID terminal. The identifier is transmitted via RF, camera, or other identifier reading techniques to the phone or terminal. Associated information: Not applicable. Conclusion: [Valid] As all of the three mandatory elements are present, it is a tag-based service.
Internet network equipment use the IP addressing scheme to identify a termination point of datagram routing, resulting in distinguishing an IP network node over the global Internet.	Identifier: IP address ID tag: Not applicable: the identifier is stored at network equipment. ID terminal: Not applicable: the identifier is transmitted on-line between network equipment, not between ID tag and ID terminal. Associated information: Applicable: IP addresses can be mapped to domain names and DNS provides resolution of their mapping relationships. Conclusion: [Invalid] As both an ID tag and an ID terminal are not involved, it is not a tag-based application.
TCP/IP applications use the IP and port addressing scheme to identify a transport end point.	Identifier: A pair of IP/port of source and destination nodes ID tag: Not applicable: the identifiers are stored in running application programs in computer systems. ID terminal: Not applicable: the identifiers are transmitted on-line between the application programs, not between ID tag and ID terminal. Associated information: Not applicable. Conclusion: [Invalid] As both an ID tag and an ID terminal are not involved, it is not a tag-based application.

Table II.1 – Evaluation of tag-based identification applications and services

Application or service	Evaluation in the context of the key elements
Web browsers use cookie identifiers to identify a semantic association over TCP connections in networked computer systems.	Identifier: Cookie number ID tag: Not applicable: the identifier is stored in running application programs in computer systems. ID terminal: Not applicable: the identifier is transmitted on-line between the application programs, not between ID tag and ID terminal. Associated information: Not applicable. Conclusion: As both an ID tag and an ID terminal are not involved, it is not a tag-based application.
Book applications use ISBN (International Standard Book Number) to identify books.	Identifier: ISBN ID tag: Applicable: the identifier is stored mostly at a cover page of book in a barcode form. ID terminal: Applicable: the identifier is transmitted via an optical scanner, that is, ID terminal. Associated information: Applicable: the identifier is related to an information resource. An identifier resolution between them is required via a query process with a database system or via a communication protocol with a remote directory service. Conclusion: [Valid] As all of the mandatory elements are present, it is a tag-based application.
Many applications use URI (Uniform Resource Identifier) to identify a network resource with describing access protocol, transport point, access authority, network and file system location, and query and fragment information.	Identifier: URI: URI itself can be used as an identifier. ID tag: Not applicable: the identifier is stored in running application programs in computer systems. ID terminal: Not applicable: the identifier is transmitted on-line between the application programs. Associated information: Not applicable: the identifier is self-descriptive by the URI specification. Conclusion: [Invalid] As all of the mandatory elements are not present, it is not a tag-based application.
The government uses an identifier scheme to identify their people.	Identifier: Passport number, for example. ID tag: Applicable: the identifier is stored on a cover page in the passport. ID terminal: Applicable: the identifier is transmitted via an optical scanner Associated information: Applicable: an identifier resolution is required via a query process with a database system or via a communication protocol with a remote directory service. Conclusion: [Valid] As all of the mandatory elements and other optional elements as well are present, it is a tag-based application.

Table II.1 – Evaluation of tag-based identification applications and services

Application or service	Evaluation in the context of the key elements
Every network node uses a MAC address to identify network nodes within a LAN scope.	Identifier: MAC address ID tag: Not applicable: the identifier is stored in node systems. ID terminal: Not applicable: the identifier is transmitted on-line between the node systems. Associated information: Not applicable. Conclusion: [Invalid] As all of the mandatory elements are not present, it is not a tag-based application.
NOTE – "[Valid]" means the application or service is considered as a tag-based identification application or service by the evaluation result, and "[Invalid]" means the opposite case.	

Appendix III

Example scenarios of tag-based identification applications and services

(This appendix does not form an integral part of this Recommendation)

III.1 Closed-domain tag-based identification applications and services

The following tag-based identification applications operate within an enterprise scope which is not limited geographically. So they can work in a closed domain of a nation-wide or global enterprise network. The enterprise scope means identification operations are not associated with outer parties as shown in Figure III.1.

A company installs multiple ID terminals within its factory. A proprietary inventory management system within a small work domain, for example, a library, can use this network configuration.

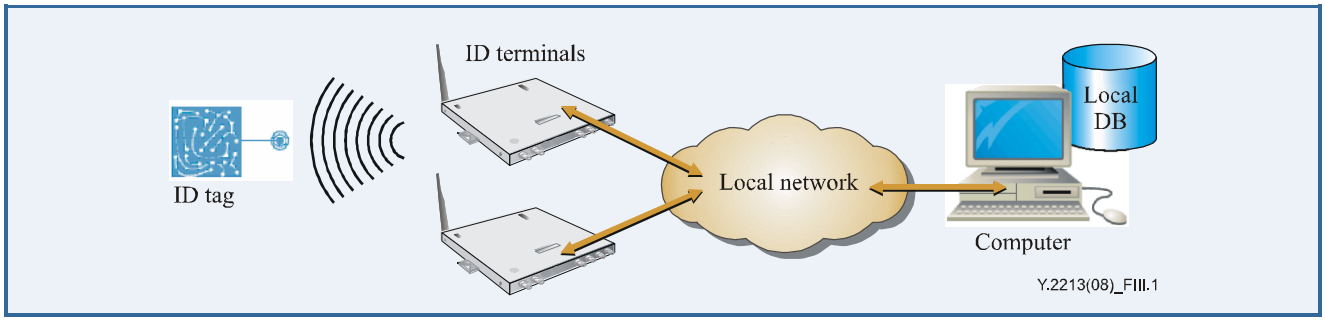


Figure III.1 – Small scale enterprise-scope configuration

An enterprise network can be expanded nationwide in a closed domain as shown in Figure III.2.

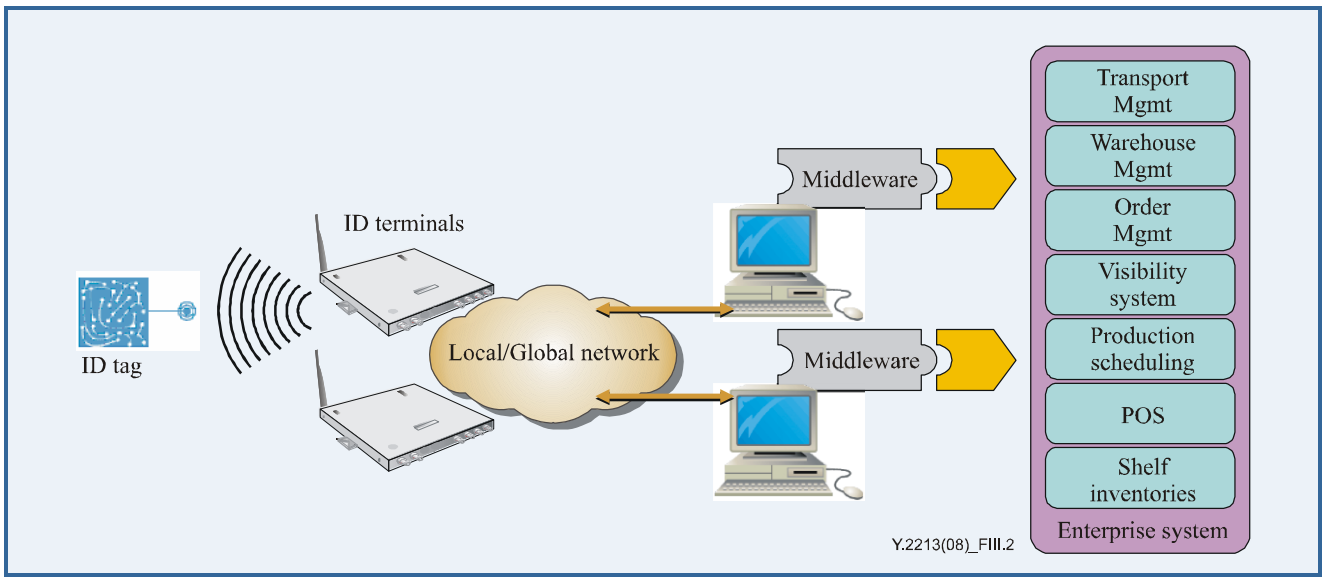


Figure III.2 – Large scale enterprise-scope configuration

III.1.1 Inventory management

A simple inventory management system is installed in a warehouse stocked with materials for manufacturing or products for sale. ID tags can be attached to item-level end products, packaged boxes, transport packages, and freight containers. Such managed objects have a transport route made by conveyer belts, transport robots, forklifts, etc. ID terminals are installed at control and monitoring points in transport routes and read identifiers from managed objects. This system allows

every incoming and outgoing object to be tracked automatically and shortages of materials can be prevented.

Usually such inventory management does not operate solely within the warehouse but typically interworks with other business application systems such as procurement, sales, manufacturing, transportation, etc., as shown in Figure III.2. Using these applications, a manufacturer can enjoy better performance of its business operations.

III.1.2 Parking management

A parking lot may adopt a tag-based parking management for parking fee, parking permit, etc. ID tags are attached to cars and an ID terminal at a gate reads identifier information from ID tags. Then the cars can be identified and parking fees can be calculated according to parking times. Entrance control also is possible with the tag-based parking management application.

A parking business company may have to handle multiple parking lots distributed in a city or nation where network functions will be needed.

III.2 B2B tag-based identification applications and services

Most existing tag-based identification applications and services correspond to the B2B type and have used identification information in a barcode form. But RF-based identification, called RFID, has been evolved dramatically with low cost solutions and is being adopted for various business applications.

A typical B2B network configuration can be depicted as shown in Figure III.3. In this case, the large scale enterprise-scope configuration is expanded to other business partners that may have their own distributed work forces. Thus, multiple networked enterprise networks are combined and then a globally networked B2B business partnership is made where identifiers are required to be globally unique.

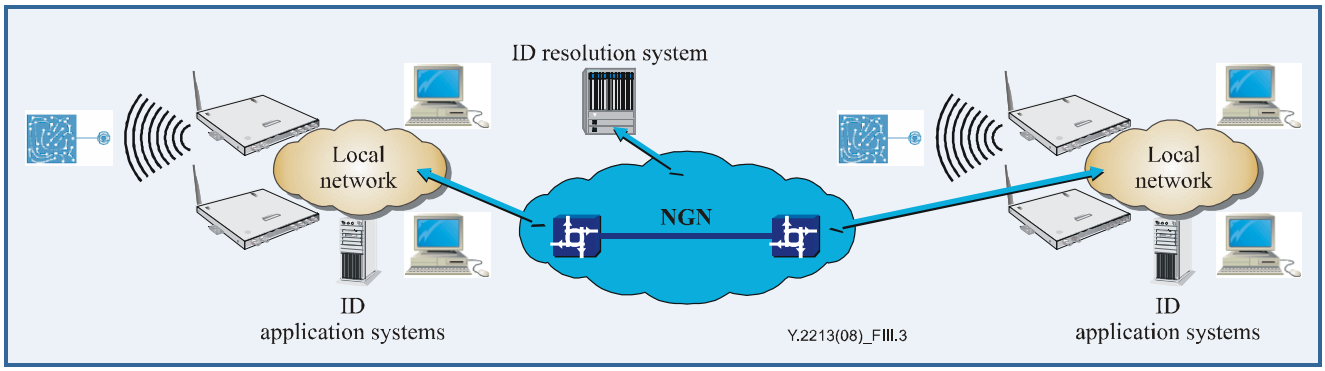


Figure III.3 – Globally networked B2B configuration model

III.2.1 Supply chain management

The supply chain management has a huge business flow with various business partners. So, application scenarios could be realized in a variety of implementation cases. The following chain is a simple case.

A company manufactures home appliances to which ID tags are attached and transports them to its warehouse in which inventory management is performed. A wholesale company orders a set of appliance goods. The manufacturer transports them from its warehouse to a warehouse of the wholesale company and then sends the wholesale company a message containing the following: the shipment notice, identifiers for the shipped appliances (read by ID terminals at the warehouse gate of the manufacturer), the volume shipped, the driver's information, etc. Such transportation is usually provided by a transport business company between the manufacturer and the wholesale company. At an entrance gate of the warehouse of the wholesale company, ID terminals read ID

tags from the appliance goods off loaded from a freight container and an SCM application of the wholesale company gathers all identifiers and analyses them to determine whether all the goods have arrived or not. Then the wholesale company responds with a reception notice to the transport company and the manufacturer.

In this scenario, identification information is exchanged between the manufacturer and transporter, the transporter and wholesaler, and the manufacturer and wholesaler in a mesh relationship. The more business partners involved, the more such chained relationships are created.

III.2.2 Manufacturing and production management

A manufacturer needs various materials to make goods at a factory and the materials are usually supplied by business partners. ID tags are affixed to materials and boxes of packaged materials in a transport unit. A supplier's ID terminals read identifiers from the materials and boxes at the originating shipping location. The supplier sends the identifiers to the manufacturer and the manufacturer receives them. ID terminals of the manufacturer read identifiers from the materials and/or transport boxes off loaded from a freight container and the manufacturer analyses the identifiers to determine whether all items were received or not. Received materials will go into a warehouse and sometime later will go out to manufacturing lines consisting of a set of conveyer belts. Materials should be routed to a proper conveyer belt. ID terminals at routing points read identifiers from the materials and provide appropriate signals to routing devices.

III.3 B2C tag-based identification applications and services

Figure III.4 shows a B2C network configuration model. Consumer terminals are connected to the service provider network which maintains and provides information contents to consumers. The information contents may be made by the service provider or content providers which are established into a logically single business domain. So the B2C model is realized as shown in Figure III.4.

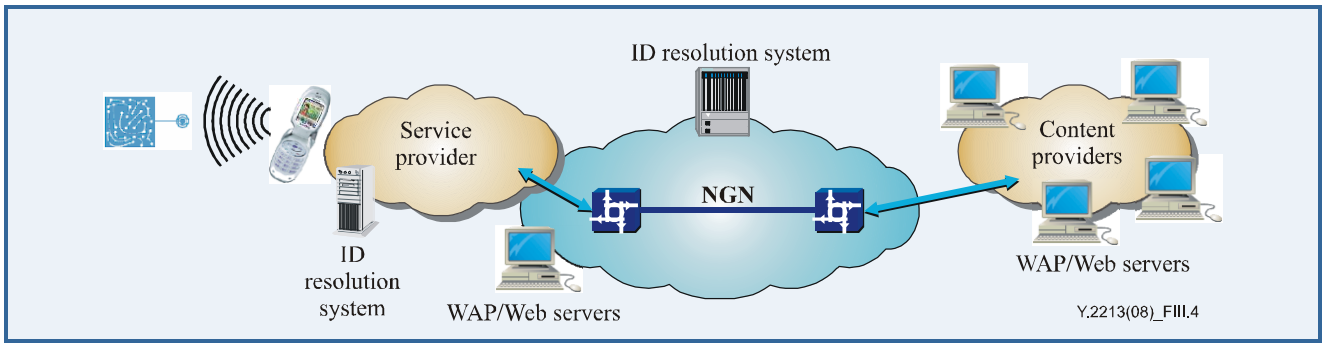


Figure III.4 – Globally networked B2C configuration model

In the B2C model, an ID terminal may be equipped additionally with an ID tag with which different types of tag-based identification applications and services can be provided as in clause III.3.3. User property is dynamic for consumer applications because service targets are human end users, but it is static for business applications because service targets are business logics of the business applications. Such different user properties produce different service characteristics as follows:

- Within a B2B environment, application information is exchanged between application systems in standardized procedures, methods, presentation styles, data structures and formats, etc., which keep used for a long time. Then an automated work process can be provided. Such service property requires a formatted data syntax which should not be changed frequently.
- Within a B2C environment, information is exchanged between business application and human. Since human users are arbitrary and may have different feelings and requirements,

an agreement on service procedures, methods, presentation styles, data structures and formats, etc., is almost impossible. So a service provider chooses a way to adapt them and provides resulting contents to end users. The contents, however, might be changed frequently by trend, culture, events, accidents, news, etc.

III.3.1 Bus and subway route search

A bus passenger reads ID tags attached to bus stop signs, gets aware of his location from the tags, inserts his destination at an input interface, and then acquires the optimum route via subways and/or buses to the destination. If he is at an inappropriate location to take a bus or subway, he gets directions on how to go to the proper location as well as the optimum route.

III.3.2 Mobile shopping

A consumer looks around a shopping mall to buy something. His cell phone equipped with an ID terminal reads ID tags attached to a shelf or items for buying and presses "order" using a shopping application user interface. In the end, he pays for chosen items as in on-line shopping. The mall packs and delivers the ordered items to his pre-registered postal address.

III.3.3 Tag-based payment

This application is a use case of payment based on ID tag, ID terminal and identifier.

When a passenger gets on a bus, he places his ID tag-attached terminal near a bus fare device and pays for the bus fare. The fare system may be connected to the network or work in the batch mode for further work flow processing. A detail payment process depends on implementations but an identifier of the passenger is transmitted to the fare device all the time.

III.4 B2B2C tag-based identification applications and services

This type of applications is a combined case between B2B and B2C applications as shown in Figure III.5. A B2B domain in the right side of Figure III.5 is connected to the B2C domain in the left side. Information resources made by the B2B domain are transmitted to the B2C domain by which they may be processed and/or upgraded for an enhanced service and then provided to consumers. The broker/gateway could be inserted into the B2C domain for intermediary proxy services such as identifier resolution, media transformation, content translation, filtering, etc.

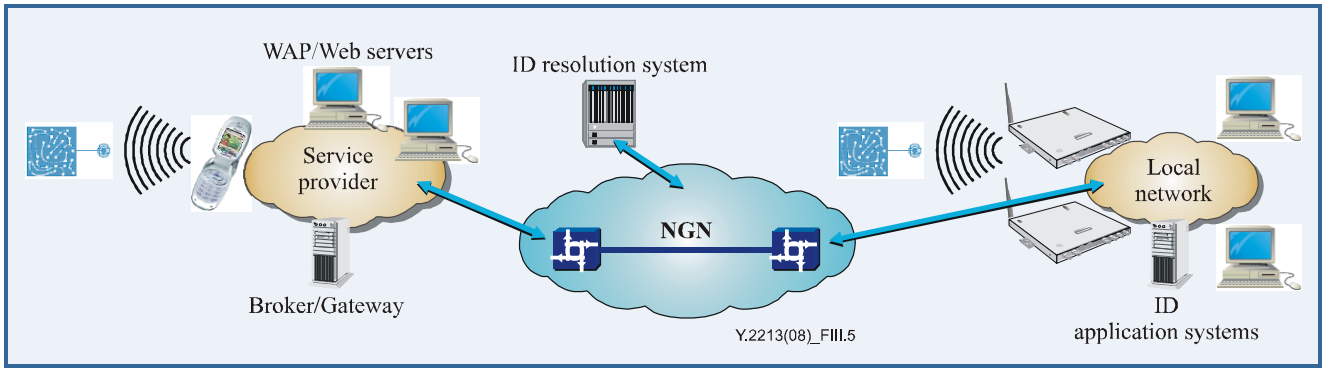


Figure III.5 – Globally networked B2B2C configuration model

III.4.1 Food chain information service

In case of vegetable goods, a tag-based distribution application scenario involving vegetable farms to wholesale markets or vegetable stores is very similar to the previous B2B cases. As before, business-oriented information including identifiers read from ID tags is exchanged between and used by business partners.

In B2B2C models, a single business corporation can care for both B2B and B2C applications or another business company provides B2C services by inter-operation with B2B applications by a business partnership contract.

At a vegetable store, a consumer browses vegetables to find better quality goods and his cell phone equipped with an ID terminal reads an identifier from a chosen vegetable to check the place of origin, harvest date, quality certification, etc. Then he can decide to buy better vegetables with such information which is prepared by a contracted business operation between B2B and B2C domains.

III.4.2 Home delivery service

This is a kind of postal service. A consumer asks a post office to send a postal package to someone. The office affixes an ID tag for delivery control in various transportation networks where many transportation agents may be involved with reading and using identifiers for routing decisions for proper delivery. That is, identification information is used in a B2B domain.

The sender reads an identifier from a receipt issued by the post office and checks delivery status for the postal package. A post man delivers the package to a receiver and his information terminal equipped with an ID terminal reads the identifier from the delivered package and sends a delivery confirmation message to the sender.

III.5 C2C tag-based identification applications and services

III.5.1 Business card exchange

The consumer-to-consumer relationship can be realized by an intervention of business domains or a direct connection between consumers. It depends on implementations.

A user terminal may have ID terminal as well as ID tag. That is, it works sometimes as a terminal and sometimes as a tag. When a user meets someone for the first time, he aims his terminal at or touches it to the other person's terminal to read identification information consisting of identifier, name, contact address, etc. Using this process business card information can be exchanged.

Bibliography

[b-ITU-T F.771] Recommendation ITU-T F.771 (2008), *Service description and requirements for multimedia information access triggered by tag-based identification*.

[b-ITU-T M.3050.1] Recommendation ITU-T M.3050.1 (2004), *Enhanced Telecom Operations Map – The business process framework*.

[b-ITU-T X.501] Recommendation ITU-T X.501 (2005) | ISO/IEC 9594-2:2005, *Information technology – Open Systems Interconnection – The Directory: Models*.

[b-ITU-T Y.Sup1] ITU-T Y-series Recommendations – Supplement 1 (2006), *ITU-T Y.2000 series – Supplement on NGN release 1 scope*.

[b-ITU-T Y.2012] Recommendation ITU-T Y.2012 (2006), *Functional requirements and architecture of the NGN release 1*.

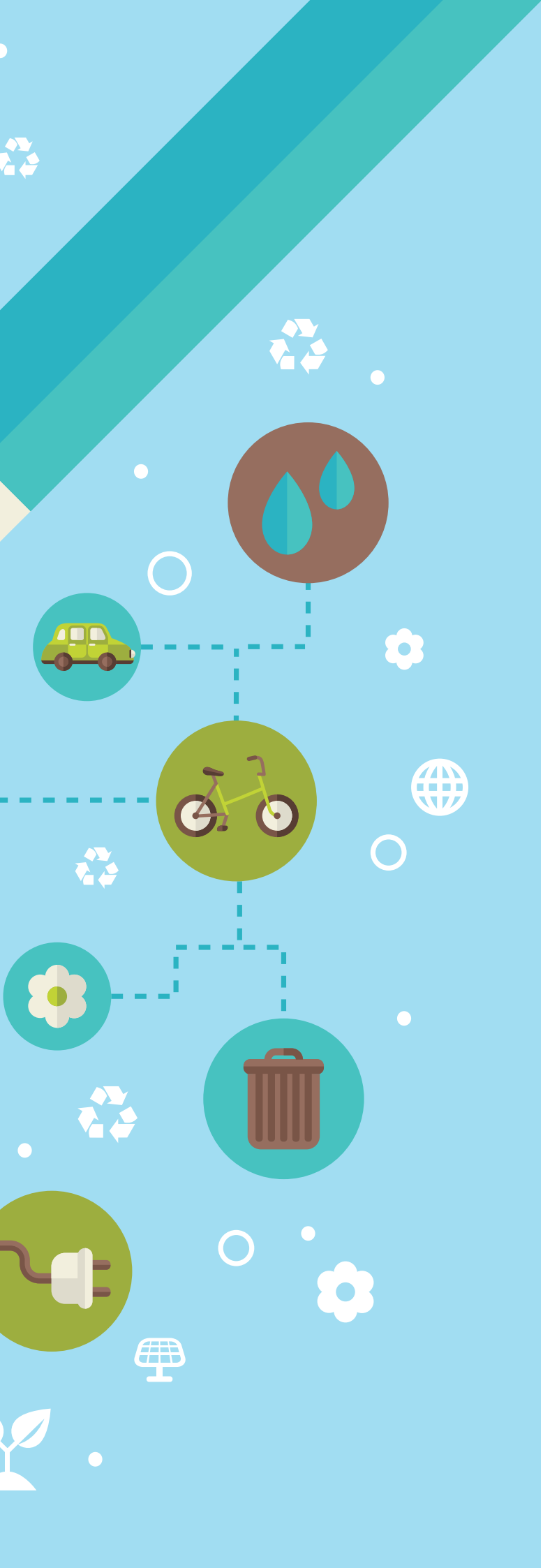
[b-ITU-T Y.2091] Recommendation ITU-T Y.2091 (2007), *Terms and definitions for Next Generation Networks*.

[b-ITU-T Z.100] Recommendation ITU-T Z.100 (2002), *Specification and Description Language (SDL)*.

[b-ISO/IEC 15961] ISO/IEC 15961 (2004), *Information technology – Radio frequency identification (RFID) for item management – Data protocol: application interface*.
<http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=30528>

[b-ISO/IEC 15962] ISO/IEC 15962 (2004), *Information technology – Radio frequency identification (RFID) for item management – Data protocol: data encoding rules and logical memory functions*.
<http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=30529>

[b-OECD-Privacy-Protection] OECD (1980), OECD *Guidelines on the Protection of Privacy and Transborder Flows of Personal Data*.
(http://www.oecd.org/document/18/0,2340,en_2649_34255_1815186_1_1_1_1,00.html)



Y.4109/Y.2061

Requirements for the support of machine-oriented communication applications in the next generation network environment

Requirements for the support of machine-oriented communication applications in the next generation network environment

Summary

Recommendation ITU-T Y.2061 provides an overview of machine-oriented communication (MOC) applications in the next generation network (NGN) environment. This includes the description of an MOC ecosystem, the characteristics of MOC and some relevant use cases. By analysing the service requirements of MOC applications, it specifies the requirements for NGN capabilities and the requirements of MOC-device domain capabilities based on these service requirements. Furthermore, this Recommendation provides a reference framework for MOC capabilities.

History

Edition	Recommendation	Approval	Study Group
1.0	ITU-T Y.2061	2012-06-15	13

Keywords

Machine-oriented communication (MOC), MOC applications, MOC capabilities, MOC device, MOC-device domain, MOC gateway, MOC-service domain, NGN, NGN capabilities, NGN domain, service requirements.

Table of Contents

Recommendation ITU-T Y.4109/Y.2061

Requirements for the support of machine-oriented communication applications in the next generation network environment

1 Scope

This Recommendation covers extensions and additions to next generation networks as well as device capabilities in order to support machine-oriented communication (MOC) applications in the NGN environment. Although this Recommendation deals with the support of MOC applications in the NGN environment, these capabilities can conceptually be applicable to other networks.

The scope of this Recommendation includes:

- network overview; description of an MOC ecosystem and the characteristics of MOC
- service requirements for the support of MOC applications
- requirements of NGN capabilities based on MOC service requirements
- requirements of MOC-device domain capabilities based on MOC service requirements
- reference framework for MOC capabilities.

NOTE – Appendix I provides details of actors and roles in an MOC ecosystem and Appendix II provides relevant use cases of MOC applications in the NGN environment.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T Q.1706] Recommendation ITU-T Q.1706/Y.2801 (2006), *Mobility management requirements for NGN*.

[ITU-T Y.2012] Recommendation ITU-T Y.2012 (2010), *Functional requirements and architecture of next generation networks*.

[ITU-T Y.2060] Recommendation ITU-T Y.2060 (2012), *Overview of the Internet of things*.

[ITU-T Y.2201] Recommendation ITU-T Y.2201 (2009), *Requirements and capabilities for ITU-T NGN*.

[ITU-T Y.2221] Recommendation ITU-T Y.2221 (2010), *Requirements for support of ubiquitous sensor network (USN) applications and services in the NGN environment*.

[ITU-T Y.2233] Recommendation ITU-T Y.2233 (2008), *Requirements and framework allowing accounting and charging capabilities in NGN*.

[ITU-T Y.2240] Recommendation ITU-T Y.2240 (2011), *Requirements and capabilities for next generation network service integration and delivery environment*.

[ITU-T Y.2701] Recommendation ITU-T Y.2701 (2007), *Security requirements for NGN release 1*.

[ITU-T Y.2702] Recommendation ITU-T Y.2702 (2008), *Authentication and authorization requirements for NGN release 1*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 device [ITU-T Y.2060]: With regard to the Internet of things, this is a piece of equipment with the mandatory capabilities of communication and the optional capabilities of sensing, actuation, data capture, data storage and data processing.

3.1.2 gateway [ITU-T Y.2091]: A unit which interconnects different networks and performs the necessary translation between the protocols used in these networks.

3.1.3 ID terminal [b-ITU-T Y.2213]: A device with a data reading and optional writing capability which reads (and optionally writes) identifier(s) and optionally application data from/into an ID tag.

3.1.4 network mobility [ITU-T Q.1703]: The ability of a network, where a set of fixed or mobile nodes are networked to each other, to change, as a unit, its point of attachment to the corresponding network upon the network's movement itself.

3.1.5 NGN service integration and delivery environment (NGN-SIDE) [ITU-T Y.2240]: An open environment in NGN integrating resources from different domains and delivering integrated services to applications over NGN.

NOTE – These domains include, but are not limited to, a telecommunication domain (e.g., fixed and mobile networks), Internet domain, broadcasting domain and content provider domain.

3.1.6 open service environment capabilities [ITU-T Y.2234]: Capabilities provided by an open service environment to enable enhanced and flexible service creation and provisioning based on the use of standards interfaces.

3.1.7 sensor [ITU-T Y.2221]: An electronic device that senses a physical condition or chemical compound and delivers an electronic signal proportional to the observed characteristic.

3.1.8 universal IC card (UICC) [b-ITU-T Q.1741.7]: A physically secure device, an IC card (or 'smart card'), that can be inserted and removed from the terminal. It may contain one or more applications. One of the applications may be a USIM.

3.2 Terms defined in this Recommendation

This Recommendation defines or uses the following terms:

3.2.1 actuator: A device performing physical actions caused by an input signal.

NOTE – As examples, an actuator might act on the flow of a gas or liquid, on electricity distribution, or through a mechanical operation. Dimmers and relays are examples of actuators. The decision to activate the actuator may come from an MOC application, a human or MOC devices and gateways.

3.2.2 machine-oriented communication (MOC): A form of data communication between two or more entities in which at least one entity does not necessarily require human interaction or intervention in the communication process.

3.2.3 machine-oriented communication (MOC) capabilities: A set of functions for the support and management of MOC applications, shared by different MOC applications and accessed through a set of standard interfaces.

NOTE 1 – When MOC capabilities are supported by NGN, they provide standard interfaces for MOC applications to MOC devices and gateways for data collection, management and operation. They also reuse or interact with NGN capabilities [ITU-T Y.2201] [ITU-T Y.2240], IT capabilities or Internet capabilities to provide MOC applications.

NOTE 2 – When MOC capabilities are supported by MOC devices and gateways, they interact with NGN functionalities and MOC applications through a set of standard interfaces.

3.2.4 machine-oriented communication (MOC) device: A device involved in the support of MOC applications.

NOTE – In the NGN environment, an MOC device connects with NGN directly or indirectly through an MOC gateway.

3.2.5 machine-oriented communication (MOC) end user: An end user of MOC applications.

NOTE – This end user may be a system (e.g., MOC application server, other network equipment, other applications, MOC device, MOC gateway), or a human (e.g., NGN end user).

3.2.6 machine-oriented communication (MOC) gateway: A gateway which interconnects and provides interoperability between MOC local networks and the network, and where applicable, interoperability at the MOC application level.

NOTE – In the NGN environment, an MOC gateway acts as a proxy or data aggregator to ensure interoperability and interconnection of MOC devices with the NGN.

3.2.7 machine-oriented communication (MOC) group: A list of MOC devices and/or gateways grouped according to one or multiple criteria.

NOTE – Criteria may include the MOC application subscriber, MOC device manufacturer, MOC application, or location.

3.2.8 machine-oriented communication (MOC) local network: A network which provides connectivity between MOC devices without the intermediation of an MOC gateway, and between MOC devices and gateways.

NOTE – An MOC local network may provide IP based and/or non-IP based connectivity.

3.2.9 meter: A device that measures and optionally records the quantity, degree, or rate of something, especially the amount of electricity, gas or water used.

NOTE – A meter is responsible for measuring the total amount of something consumed in a period.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

ACI Application to Capability Interface

ANI Application to Network Interface

API Application Programming Interface

B2C Business to Customer

CDR Charging Data Record

GNSS Global Navigation Satellite Systems

GPS Global Positioning System

IC Integrated Circuit

ID Identification

IP Internet Protocol

IT Information Technology

MOC Machine-Oriented Communication

NGN Next Generation Network

NNI Network to Network Interface

OSE Open Service Environment

QoS Quality of Service

SIDE	Service Integration and Delivery Environment
SIP	Session Initiation Protocol
SLA	Service Level Agreement
SMS	Short Message Service
SNS	Social Network Services
UICC	Universal Integrated Circuit Card
UNI	User to Network Interface

5 Conventions

In this Recommendation:

The keywords "is required to" indicate a requirement which must be strictly followed and from which no deviation is permitted if conformance to this document is to be claimed.

The keywords "is recommended" indicate a requirement which is recommended but which is not absolutely required. Thus this requirement needs not be present to claim conformance.

The keywords "can optionally" and "may" indicate an optional requirement which is permissible, without implying any sense of being recommended. These terms are not intended to imply that the vendor's implementation must provide the option and the feature can be optionally enabled by the network operator/service provider. Rather, it means the vendor may optionally provide the feature and still claim conformance with the specification.

6 Introduction

6.1 Network overview

Machine-oriented communications (MOC) are a form of data communications between two or more entities in which at least one entity does not necessarily require human interaction or intervention in the process of communication.

MOC include communications with remote MOC devices for the support of procedures covering aspects such as registration, authentication, authorization, monitoring, maintenance, provisioning and troubleshooting. MOC applications intend to automate decision and communication processes.

Figure 6-1 shows the network overview for the support of MOC applications in the NGN environment.

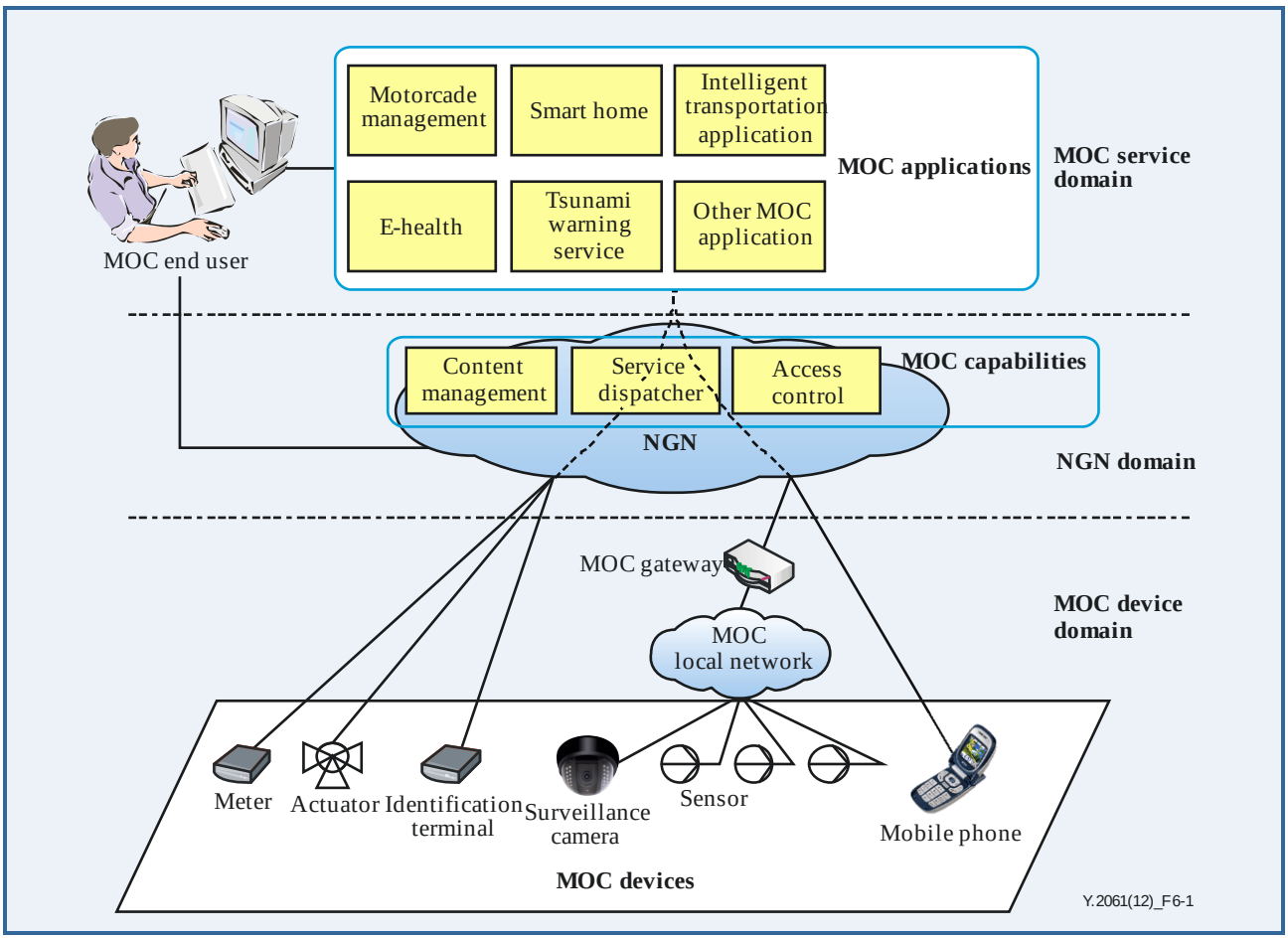


Figure 6-1 – Network overview for the support of MOC applications in the NGN environment

The MOC device domain includes MOC devices and MOC gateways. MOC devices include various types of devices as shown in Figure 6-1.

NOTE 1 – Proprietary devices (see Figure 11-1) are not shown in the above figure.

MOC devices can be categorized into general devices, data capturing devices, data carrying devices and sensing or actuating devices [ITU-T Y.2060]. Examples for the different categories include:

– sensing or actuating devices: sensor, surveillance camera, meter, actuator with remote control;

– data capturing devices and data carrying devices: identification terminal (ID terminal);

– general devices: mobile phone, personal computer, networked television.

MOC devices may access the NGN directly or via an MOC local network and the MOC gateways attached to it. MOC devices or gateways may access the NGN by use of wire-line or wireless connectivity.

MOC devices and gateways may access NGN via multiple access networks, for example to ensure reliable communications.

The NGN domain not only provides access, data transportation, network control and interconnection (with other networks) functions, but also provides MOC capabilities to support multiple MOC applications.

MOC capabilities reuse or interact with NGN capabilities [ITU-T Y.2201], expose functionalities to MOC applications through a set of standard interfaces, provide support to facilitate application development and deployment through hiding network specificities to MOC applications. MOC

capabilities include capabilities for content management, the service dispatcher and access control. Details can be found in clause 11.

NOTE 2 – Although not shown in Figure 6-1, MOC capabilities in NGN can also interact with other applications outside the MOC service domain, such as social network services (SNS) or blog applications, which can make MOC related information available in accordance with customer or application requirements.

The MOC service domain includes MOC applications. MOC applications run the application logic and use MOC capabilities accessible via standard interfaces.

NOTE 3 – Although not shown in Figure 6-1, MOC capabilities and MOC applications can also exist in the MOC device domain.

6.2 Types of machine-oriented communications

MOC covers communications among MOC devices and humans, specifically:

– communications among different MOC devices and among MOC devices and MOC applications;

– communications among MOC devices and other devices controlled by humans.

The first type of communications deals with data collection, device management, device operations and other communication functions with remote equipment. These communications are used in many scenarios, e.g., that of MOC applications getting the relevant information provided by sensors.

The second type of communications may be initiated by remote MOC devices to timely inform humans about relevant information detected in MOC devices, or may be initiated by humans to get relevant information from remote MOC devices. These communications involve many scenarios, e.g., a human connecting to a surveillance camera in his house by using a mobile phone.

In the case of an MOC device interacting with MOC capabilities in a network domain or with MOC applications, the execution of an MOC application in an NGN environment may be divided into the following phases:

- Data collection: the MOC device detects, measures and records data (e.g., data related to physical properties, multimedia data, etc.). When the MOC device meets a trigger threshold or receives an instruction from MOC capabilities in the network domain or MOC applications, the MOC device requests the NGN to transfer the data to the MOC application.

  NOTE 1 – The MOC device follows the pre-configured policy, which can be decided by the MOC capabilities or the MOC applications. According to the policy, the MOC device detects data, executes the logic, and initiates the communication to the MOC applications or human-controlled MOC devices to report the relevant information.

- Data transportation: the NGN establishes a data path between the MOC device and the MOC capabilities. The MOC application can communicate with the MOC device directly (without gateway) under the authorization of the MOC capabilities in NGN: in order to manage the MOC device, the MOC application gets the authorization information from the MOC capabilities which are used for the secure communication's authorization and session key negotiation.

  NOTE 2 – The MOC device may also initiate such a process.

- Data analysis: the MOC application analyses data received from the MOC device. MOC capabilities in NGN can also analyse the data based on rules defined by MOC end users.

- Service delivery: the MOC application executes the service logic and decides how to publish the information to MOC end users (including MOC devices, humans or other applications). Information delivery can be "active" meaning that the MOC application

forwards information to MOC end users automatically. The MOC application also supports forwarding the information based on demand from MOC end users ("passive" information delivery).

NOTE 3 – Not all of the above phases are necessary for the execution of all MOC applications. For example, an MOC end user can send a request to a vehicle-related MOC application to turn on the klaxon of the vehicle when he/she wants to find out his/her vehicle in a large parking area, and the vehicle-related MOC application will turn on the vehicle klaxon after receiving the request. In this example, data collection and data transportation from the MOC device (the vehicle) to the MOC application are not executed.

6.3 MOC ecosystem

Figure 6-2 depicts business roles which are relevant in an MOC ecosystem, and their relationships.

The business roles identified in Figure 6-2 and their relationships are based on the IoT ecosystem [ITU-T Y.2060] and adapted to the context of MOC.

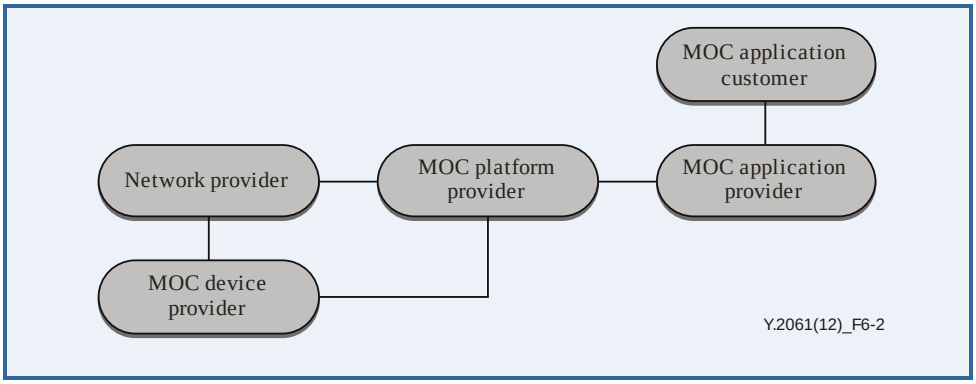


Figure 6-2 – Business roles in an MOC ecosystem

Five key roles are identified: MOC application customer, network provider, MOC device provider, MOC platform provider and MOC application provider.

- Network provider

In the context of this Recommendation, the network provider is the role that offers the NGN capabilities as described in [ITU-T Y.2201].

The network provider has a business relationship with the MOC platform provider and the MOC device providers.

NOTE 1 – An actor playing the role of network provider can also play the roles of MOC platform provider, MOC device provider and MOC application provider.

- MOC application provider

The MOC application provider is the role providing functions in the MOC service domain. It utilizes capabilities provided by the MOC platform provider in order to provide services to the MOC application customer.

The MOC application provider has a business relationship with the MOC application customer and the MOC platform provider.

NOTE 2 – An actor playing the role of MOC application provider can also play the role of MOC platform provider.

- MOC device provider

The MOC device provider is responsible for providing functions in the MOC device domain. It provides raw data or other necessary resources to the network provider and the MOC platform provider according to the service logic.

The MOC device provider has a business relationship with the MOC platform provider and the network provider.

NOTE 3 – An actor playing the role of MOC device provider can also play the role of MOC application provider and the MOC platform provider.

- MOC platform provider

The MOC platform provider is the role responsible for providing the following functions in the NGN domain:

– access to and integration of resources provided by MOC device providers and the network provider;

– support and control of the service integration and delivery functionalities;

– offering to the MOC application provider of capabilities (including resource exposure) for support of MOC applications.

The MOC platform provider has a business relationship with the MOC device provider, the MOC application provider and the network provider.

NOTE 4 – An actor playing the role of MOC platform provider can also play the role of MOC application provider and MOC device provider.

- MOC application customer

The MOC application customer may be a human or a device. The MOC application customer consumes applications offered by the MOC application provider. Organizations or persons such as enterprise, family or individuals are examples of MOC application customers.

The MOC application customer has a business relationship with the MOC application provider. The MOC application customer is a subscriber of the MOC application provider.

NOTE 5 – A given MOC application customer can represent multiple MOC end users in the MOC-service domain.

Appendix I provides details of actors and roles in the MOC ecosystem.

7 Characteristics of MOC

This clause provides the characteristics of MOC including those for applications, devices and gateways. Consideration should be given to the fact that these characteristics may vary across MOC applications, devices and gateways. These characteristics include, but are not limited to, the following:

1) Variety of MOC device types and capability levels

There are various types of MOC devices for different uses; some of them have low performance and limited functionality (e.g., low processing capability, small memory, limited security capabilities), while others have powerful embedded capabilities (e.g., bilateral authentication and authorization capabilities with the network and MOC applications).

2) MOC applications support of heterogeneous MOC devices

MOC applications may communicate with more than one type of MOC devices. In such cases, MOC applications need to cope with this heterogeneity.

3) Grouping of MOC devices

In some use cases, groups of MOC devices are deployed for services. Usually, the MOC devices of a specific group have the same characteristics, functions, performance or policies.

4) Variety of mobility levels of MOC devices and gateway

Some MOC devices and gateways are mobile and can be used everywhere. Some devices and gateway do not move. Some MOC devices and gateways move only within a certain area. Some MOC devices and gateways should not be moved once installed (their movement would mean theft of these devices has taken place).

5) Remote management of MOC devices in varied and large deployments

Massively deployed MOC devices cover large areas, they exist for a long time and they may be "moving", thus making it difficult for operators or MOC end users to manage all devices in the field.

The remote management functionality of MOC devices (e.g., firmware updates) is particularly important in such deployments.

6) Increased security threats from limited manual intervention

Some MOC devices and gateways are required to be managed remotely rather than operated manually in the field. This increases the security threat level, such as physical tampering, hacking, and unauthorized monitoring and so on. So adequate security measures should be provided to detect or resist possible attacks.

7) Variety of data communication characteristics

Most MOC applications depend on data communications driven by machines without human interventions. The characteristics of such data communications are much different from human driven communications.

The variety of data communications is also caused by the variability of other factors, such as packet size (small or large packets), data transmission periods and frequency or the MOC device communication role (initiating party or terminating party).

8) Large amounts of data transmitted to the network

With more and more intelligence embedded in devices and the large number of devices connected to the network, large amounts of data are transmitted to the network.

8 Service requirements of MOC applications

8.1 Mobility levels

Different types of MOC applications may require different levels of MOC device and/or gateway mobility. This includes no mobility for devices and gateway that do not move, limited mobility for devices and gateway that move infrequently (e.g., medical devices on a patient may not move frequently), or low geographical mobility for devices and gateway that move only within a certain region (e.g., bus equipment devices such as a camera may move only within a city). It is needed to provide mobility management for different mobility levels in order to reduce resource usage (e.g., the timer of periodic location update should be reduced for the MOC devices which have infrequent movement).

Requirements of mobility levels are as follows:

1) MOC applications are required to be supported with optimized mobility management according to the required levels of mobility.

8.2 Time controlled network communications

In order to minimize costs and optimize network efficiency, MOC devices or MOC gateways can locally cache the collected data and transmit them to the network during the time slots allowed or pre-configured by the network operator.

Requirements for time controlled network communications are as follows:

1) MOC applications are recommended to support time control for MOC device communications with the network based on service criteria (e.g., daily network traffic load, MOC device location).

8.3 Resource usage

MOC applications are expected to engage low resource usage in the case where MOC devices send or receive data infrequently (i.e., with a long time period between two data transmissions). To improve the operational efficiency of an MOC application and decrease the MOC application's operational costs and the MOC device's energy consumption, the resource usage of both MOC devices and networks need to be optimized.

MOC applications requirements for optimized resource usage are as follows:

1) MOC applications are recommended to optimize the usage of resources for both MOC devices and networks.

NOTE – This is particularly relevant in the case of infrequent data transmissions.

8.4 Interoperability with proprietary devices

Since a lot of proprietary devices (e.g., devices with proprietary standards for inter-working with network entities) have already been deployed, MOC applications should be able to support interoperability with these proprietary devices.

Requirements for the support of proprietary devices are as follows:

1) It is recommended that MOC applications be interoperable with proprietary devices through appropriate means, e.g., MOC gateways.

2) MOC applications are recommended to support the effective hiding of proprietary devices' operations.

8.5 Application collaboration

For some MOC application cases, there may be multiple MOC application providers providing different MOC applications which need to collaborate with each other.

As an example, business to customer (B2C) companies deliver products to logistics companies for further delivery to the customers who have ordered them. Logistics companies track the products in transit, while customers may check the whereabouts of the purchased products by visiting the application systems of the B2C companies which collaborate with the application systems of the logistics companies.

MOC applications requirements for application collaboration are as follows:

1) MOC applications are recommended to support application collaboration with other MOC applications via the intermediation of the MOC capabilities.

2) MOC applications of a given MOC application provider are recommended to support application collaboration with MOC applications of other MOC application providers via the intermediation of the MOC capabilities.

8.6 Support of service integration and delivery environment

MOC applications would benefit from support by the network of a service integration and delivery environment. In addition to the support of common purpose capabilities for different types of applications, interfaces to such an environment would enable the support of specific capabilities for MOC applications.

MOC application requirements for the support of service integration and a delivery environment are as follows:

1) MOC applications are recommended to be able to access service integration and delivery environment capabilities provided by the network.

NOTE – Access to such an environment also provides access to application development and testing capabilities.

8.7 Load balancing and robustness

In some application cases, e.g., in a river control application with the water level of the river reaching the alert threshold, a large number of monitor points may send alert information to the MOC capabilities in the NGN domain, including the possibility to send real-time video. Such emergency scenarios require robustness of the network and MOC capabilities in the NGN domain.

The distribution of MOC devices across territories and the density of MOC devices in a certain area may be high or low. This may cause an imbalance in the network and MOC capabilities in the NGN domain for both signalling and data traffic.

In order to support network and MOC platform load balancing and robustness, the requirements are as follows:

1) MOC applications require mechanisms in the network and MOC capabilities in the NGN domain for load balancing.

2) MOC applications require a robust network and MOC capabilities in the NGN domain, whilst also ensuring a sufficient level of QoS under given circumstances, e.g., emergency scenarios.

8.8 Accounting and charging

Different charging and accounting requirements need to be addressed depending on the scenarios of the MOC applications. For example, there are MOC applications with frequency data transmission and small amounts of data, in this case charging and accounting may be based on the number of communications. Some other MOC applications may rarely connect to the network but generate large amounts of data for each communication, in this case charging and accounting may be based on the amount of data. In other cases, charging and accounting may be based on the duration of communications.

As an MOC application may use multiple devices for a single customer, charging per device by MOC application providers or network providers would generate a lot of charging data records (CDR) that impose a heavy load on some functions, e.g., the charging function. In these scenarios, group-based accounting and charging instead of per device accounting and charging may be more appropriate.

MOC applications have the following requirements:

1) MOC applications are required to support different charging and accounting methods, such as charging based on the duration of communications, number of communications, amount of transmitted data, etc.

2) MOC applications are recommended to support unified charging for customers.

3) When group-based support is enabled (see clause 8.12), MOC applications are required to support online and offline accounting and charging based on groupings.

8.9 Management

8.9.1 Device management

MOC devices cover a large area, exist for a long time and could be "on the move", so it may be difficult for operators or subscribers to manage these devices manually. Thus, MOC devices and gateways should be managed and monitored remotely (for example, updating the firmware to correct faults).

MOC devices with a universal integrated circuit card (UICC) may be deployed outdoors without human supervision and it might happen that a given UICC is put into another device without permission of the UICC owner. Thus, in order to avoid such issues, the change of association between an MOC device and UICC should be accessible by MOC applications interacting with those MOC devices.

When MOC devices and gateways provide service logic, the MOC devices and gateways provide support capabilities for both the customer and the service. MOC devices and gateways are required to be managed in terms of both network and service management.

The requirements of MOC device management are as follows:

1) MOC applications are required to support mechanisms for managing gateways acting as traffic aggregators (a gateway aggregates traffic and acts as a channel).
2) MOC applications are required to monitor the state of various aspects of MOC devices and gateways including:
   a) abnormal behaviour of MOC devices and gateways, such as an active service not being aligned with the subscribed feature;
   b) the association between the MOC devices and gateways and the UICC;
   c) the attachment information of MOC devices and gateways, such as attachment location;
   d) the connectivity of MOC devices and gateways.
3) MOC applications are required to support mechanisms to perform simple and scalable pre-provisioning of MOC devices and gateways, enable and disable features, report errors from devices, and query device status.
4) MOC applications are required to support mechanisms to perform software upgrades (e.g., provisioning of new service logic and/or bug fixes to be loaded on devices and/or gateways, including applications and system software).
5) MOC applications are required to manage low capability MOC devices using lightweight mechanisms.

8.9.2 Service profile management

The service profile of a specific MOC application is composed by a set of information specific to that MOC application. It may include, but it is not limited to, the MOC application identifier, MOC application provider identifier and application data types.

MOC applications have the following requirements:

1) MOC applications are recommended to use standard service profiles for registration and discovery.
2) MOC applications are required to support mechanisms to perform service profile updates.

8.9.3 Device profile management

The MOC device profile is a set of information related to MOC devices and MOC gateways. As there are various types of MOC devices and MOC gateways, the device profiles are helpful in the management of large numbers of heterogeneous devices and gateways.

NOTE 1 – The MOC device profile information may include an MOC device identifier, MOC device type, MOC device capabilities and MOC device location.

MOC applications have the following requirements:

1) MOC applications are recommended to use and manage standard device profiles for MOC devices and gateways, including their registration and discovery.

NOTE 2 – Device profile management includes the creation of associations between MOC devices (gateways) and service profiles.

8.10 Addressing and identification

There are two types of connection methods between MOC devices and MOC applications. MOC devices may connect to MOC applications directly or via MOC gateways based on IP connectivity. Different MOC devices may communicate with different MOC applications via a single MOC gateway or via multiple gateways.

NOTE – The multiple gateways scenario allows the reduction of an MOC gateways' load and the growth of the network access reliability.

MOC devices may support public or private IP addresses and may also support non-IP addresses when they connect to the network via MOC gateways. MOC application servers and MOC devices which are using public IP addresses should be able to communicate with other MOC devices which are using private IP addresses.

The requirements of addressing and identification are as follows:

1) MOC applications are required to be able to operate with different types of MOC device addressing schemes, e.g., IP addressing and non-IP addressing schemes.
2) MOC applications require the support of unique identification of MOC devices.
3) MOC applications require the support of unique identification of MOC groups (see clause 8.12 for description of an MOC group).
4) MOC applications require the support of addressing mechanisms enabling communication with MOC devices behind an MOC gateway.
5) MOC applications require the support of addressing mechanisms enabling communication with MOC gateways.
6) MOC applications require the support of unique identification of MOC gateways.

8.11 Location-based support

Location data may be collected by MOC applications from MOC devices and gateways or from the network. Types of location information include global navigation satellite systems (GNSS), latitude or longitude data or cell identifier (CellID).

MOC applications have the following requirements:

1) MOC applications are required to be aware of the location of MOC devices. For example, based on the location information of MOC devices, the MOC application could initiate a service trigger to upgrade the firmware on the MOC devices within a certain area by the broadcast or multicast method.
2) MOC applications are recommended to maintain and manage location information of both a single MOC device and a set of MOC devices behind an MOC gateway.
3) MOC applications are recommended to maintain and manage different types of location information.

8.12 Group-based support

MOC groups may be used in many MOC applications. For example, a vehicle company owner could manage the company vehicles in groups, e.g., track the locations of all the vehicles in the group and send notification messages to all the vehicles in the group. An electricity company could collect the metering data of all the MOC devices in a certain area at a certain time. A consumer could query the different meters at his home when he is on a business trip.

Different MOC applications may have different MOC groups.

MOC applications may have static MOC groups which are pre-configured, for example the consumer could pre-configure the MOC devices installed at his home into an MOC group. MOC applications may also have dynamic groups which are grouped according to some on-demand criteria, for example a vehicle company owner may request to communicate with all the vehicles in a certain area when needed.

The MOC devices inside a group may directly connect to the network or indirectly connect to the network.

The requirements for group-based support are the following:

1) MOC applications require the support of static and dynamic MOC groups.
2) MOC applications require the support of data transmission to/from one or all members in an MOC group using group identifier.
3) MOC applications require the support of the group based QoS policy.
4) MOC applications require the support of group based traffic parameters.
5) MOC applications require the support of MOC group management, including display/creation/modification/deletion of MOC groups and associated attributes and display/addition/modification/deletion of MOC group members.
6) MOC applications are recommended to be able to send data per MOC group and apply data prioritization according to member's data prioritization in the MOC group.

8.13 Quality of service

8.13.1 Application traffic control

The application traffic is not only generated by MOC devices, but also generated by MOC applications.

MOC applications often cover a large number of MOC devices and gateways. In such scenarios, from the viewpoint of applications, their QoS may be impacted by high application traffic.

From the viewpoint of the network, the QoS of MOC applications may be improved if the application traffic is well managed.

MOC applications have the following requirements:

1) MOC applications require mechanisms for application traffic management, e.g., to limit the maximum number of application transactions per second.
2) MOC applications require that access concentration into a single resource is avoided.

8.13.2 Data prioritization

The MOC mission-critical applications should be carefully managed. For example, emergency notification of a fire incident must be delivered in a timely and reliable way to the appropriate national disaster monitoring systems. In order to provide alarm notifications, the emergency data are carried over the network.

MOC applications have the following requirements:

1) MOC applications are recommended to be able to set the prioritization of data (within a single application or among different applications).
2) MOC applications are recommended to be able to manage different data according to their prioritization.
3) MOC applications are recommended to be able to apply data prioritization to MOC devices and gateways according to the related service level agreements (SLA) between MOC application customers and MOC application providers.

8.14 Security

8.14.1 Authentication and authorization

The MOC end users accessing the MOC applications need to be authenticated and authorized. Access to applications has to align with the relevant security levels.

The MOC devices involved in the MOC applications and directly connected need to be authenticated and authorized.

The MOC devices involved in the MOC applications and connected via an MOC gateway should generally be authenticated and authorized.

Requirements of authentication and authorization are as follows:

1) MOC applications are required to support the authentication and authorization of MOC end users to access MOC applications and related data according to the related security levels.
2) MOC applications are required to support a mechanism for authentication and authorization of directly connected MOC devices associated with the MOC applications themselves.
3) MOC applications are recommended to support a mechanism for authentication and authorization of MOC devices which are in an MOC local network (connected via an MOC gateway) and which are associated with the MOC applications themselves.
4) MOC applications are required to support a mechanism for the registration of directly connected MOC devices associated with the MOC applications themselves.

8.14.2 Security of data

In general, MOC applications require strong security, due to very sensitive data. It has to be considered that MOC devices cannot provide all security features because they may have system limitations. For example, sensed data carried over the network may not be sufficiently protected from the security viewpoint.

MOC applications have the following requirements:

1) MOC applications are required to provide security for the connectivity between MOC applications and MOC devices even when the MOC devices roam from one network domain to another network domain.
2) MOC applications are required to support the integrity and confidentiality of the data exchanged during the application operations.
3) MOC applications are recommended to provide mechanisms of data encryption in order to also support MOC devices with limited capabilities.

8.14.3 Security of MOC device access

All data produced by MOC devices are required to be unknown to unauthorized entities. For example, private or sensitive data of an MOC device should not be sent to an unauthenticated end user if this end user initiates a communication with that MOC device.

Due to the limited capabilities of the MOC devices with no support of authentication and authorization functionalities, MOC applications have the following requirements:

1) Before MOC device resources can be used by MOC end users and MOC applications, MOC applications are required to support mechanisms for the authentication and authorization of MOC end users and MOC applications for their access to MOC device resources.

8.15 Device association and interaction with multiple applications

In some application scenarios, a single MOC device may be required to communicate with different MOC applications simultaneously. For example, when a traffic accident happens, the damaged vehicle may be required to provide information to both the health service centre and the department for traffic control.

MOC applications have the following requirements:

1) MOC applications are required to not prevent the delivery of information to other MOC applications by an MOC device or gateway associated with that MOC application.

2) MOC applications are required to not prevent an MOC device or gateway associated with that MOC application to receive information from other MOC applications.

NOTE – In this Recommendation, it is assumed that the capability of an MOC device or gateway to communicate with multiple applications is controlled by the network (i.e., NGN). Requirements of an MOC device or gateway with "network control independent capability" to communicate with multiple applications are for further study.

8.16 Communication with sleeping device

In the case of offline status for a given period of time, MOC devices enter or stay in sleep mode in order to:

– save power, especially for devices using a battery

– save network resources, especially for devices with wireless network access.

NOTE – Sleep mode is an energy-saving mode. It normally refers to an MOC device in a situation when, traffic is not being generated for a period of time, device sessions and related traffic channels are released to save resources, and all unnecessary components are shut down. According to certain criteria, an MOC device in offline status can enter sleep mode.

MOC applications have the following requirements:

1) MOC applications are recommended to be able to send instructions to a sleeping MOC device to wake it up.

2) MOC applications are required to support network-initiated communications towards a sleeping MOC device.

8.17 Differentiation and handling of collected data

With the large variety of data collected by MOC devices being transmitted in the network, it is expected that the network be able to differentiate particular collected data from other data, and then trigger the relevant processes based on their category. For example, the network may cache and only later forward data which are collected in non-network performance sensitive applications. On the other hand, the network is required to immediately transmit high priority data which are collected in network performance sensitive applications.

MOC applications requirements for collected data differentiation and related handling are as follows:

1) MOC applications are required to enable the identification and categorization of the data collected by MOC devices according to relevant policies.

2) If the network can handle data collected by MOC devices according to relevant data categorization, MOC applications are required to manage these data accordingly.

9 Requirements of NGN capabilities

9.1 Requirements for extensions or additions to NGN capabilities

This clause identifies extensions or additions to NGN capabilities defined in [ITU-T Y.2201] for the support of MOC applications.

9.1.1 Numbering, naming and addressing

NGN provides addressing and identification capabilities. The service requirements specified in clause 8.10 are supported by the existing capabilities of NGN [ITU-T Y.2201] [ITU-T Y.2702].

Based on the service requirements in clause 8.12, NGN is required to support the following additional numbering, naming and addressing requirements.

1) An NGN is required to provide group based addressing mechanisms for the support of MOC applications according to the NGN provider's policy.

2) An NGN is required to support a static MOC grouping capability.

NOTE 1 – A static MOC group contains the members of the MOC devices and gateway which are pre-configured.

3) An NGN is required to support a dynamic MOC grouping capability.

NOTE 2 – A dynamic MOC group may be generated upon request using specific criteria, such as location, status of MOC devices and gateway, etc.

4) An NGN is required to support the MOC grouping capability in both cases of groups constituted by MOC devices and gateway directly or indirectly connected to NGNs.

5) An NGN is required to map the MOC group identifier to network addresses of MOC devices and gateways of a static MOC group.

6) An NGN is required to identify the list of MOC devices and gateways and their network addresses matching the specified criteria for a dynamic MOC group.

9.1.2 Quality of service

The differentiated quality of service and data prioritization requirements, specified in clause 8.13 are supported by the existing capabilities of NGN [ITU-T Y.2201] [ITU-T Y.2221]. The following subclauses identify the requirements for extensions to NGNs.

9.1.2.1 Per group QoS policy

Based on the service requirements in clause 8.12, the following additional requirements are placed on NGNs:

1) An NGN is required to support a per group level QoS policy, in parallel with, or instead of, a per device level QoS policy.

NOTE – Per group QoS policy parameters include, but are not limited to:

– packet transfer delay

– packet delay variation

– packet loss ratio

– packet error ratio.

9.1.2.2 Traffic control

NGNs provide processing and traffic management capabilities. The service requirements specified in clause 8.7 are supported by the existing capabilities of NGN [ITU-T Y.2201].

Based on the time controlled network communication requirements defined in clause 8.2, the following additional requirements are placed on NGNs:

1) An NGN is required to allow MOC end users' access (e.g., attachment to the network or establishment of a data connection) during a defined granted network communication access time interval.

2) An NGN is required to reject MOC end users' access (e.g., attachment to the network or establishment of a data connection), or allow it with different charging parameters, during a defined forbidden network communication access time interval.

3) An NGN is required to allow the modification of granted network communication access time intervals based on service criteria (e.g., daily network traffic load, MOC device location).

4) An NGN is required to communicate granted network communication access time schedules and durations to MOC devices and gateways.

5) An NGN is required to terminate MOC end users' access (e.g., detachment from the network or release of a data connection) when a network communication access time duration has ended.

6) An NGN can optionally support the communication of granted network communication access time schedules and durations to other MOC end users than the MOC devices and gateway (e.g., MOC application server).

Based on the resource usage requirements in clauses 8.3 and 8.16, the following additional requirements are placed on NGNs:

7) An NGN is required to page the target MOC devices and gateways before service interaction when the network needs to initiate a service.

8) An NGN is required to establish communication resources only when data transmission is required.

Based on the requirements in clause 8.12, in addition to the existing NGN traffic control capabilities, an NGN is required to support:

9) optimized handling of group communications in order to save network resources and to prevent network congestion;

10) per group level traffic control in parallel with, or instead of, per device level traffic control.

NOTE – Per group level traffic parameters include, but are not limited to:

– maximum allowed packet size

– data rate and the bucket size

– peak rate and peak bucket size

– sustainable rate and sustainable bucket size.

9.1.3 Accounting and charging

Based on the service requirements in clause 8.8, the following additional requirements are placed on NGNs [ITU-T Y.2233]:

1) An NGN is required to support group-based accounting and charging for either or both online charging and offline charging in parallel with, or instead of, per device level charging.

9.1.4 Mobility

NGNs provide mobility support for end users, MOC devices and gateways [ITU-T Y.2201] [ITU-T Q.1706].

Based on the mobility levels in clause 8.1, the following additional requirements are placed on NGNs:

1) An NGN is required to support pre-defined mobility levels for MOC devices and gateways via the device profile management.

2) An NGN is required to support different mobility level management according to the mobility requirements of MOC devices and gateways, such as reducing the frequency of the mobility management procedures for MOC devices and MOC gateways with low mobility.

3) An NGN is recommended to support dynamical instruction of the MOC devices and gateway in order to set the mobility level (implying, for example, the adjustment of the frequency of mobility management procedures).

9.1.5 Profile management

9.1.5.1 User profile management

Based on the service requirements in clause 8.9.2, the following requirement is placed on NGNs:

1) An NGN is recommended to support standard service profiles with enhancements for MOC applications' specific information.

9.1.5.2 Device profile management

Based on the service requirements in clause 8.9.3, the following requirement is placed on NGNs:

1) An NGN is recommended to support standard device profiles with enhancements for MOC devices and gateway's specific information.

9.1.6 Device management

Based on the service requirements in clause 8.9.1, NGN is required to support the following additional device management requirements:

1) An NGN is required to be able to manage and control MOC devices and gateways, including:
   a) monitoring MOC devices and gateways' operations;
   b) where applicable, monitoring changes, and related actions, in the associations between MOC devices or the gateway and UICCs;
   c) monitoring changes, and related actions, related to the network attachment points of MOC devices and gateways;
   d) monitoring MOC devices and gateways' network connectivity.

9.1.7 Data differentiation and handling

Based on the requirements of differentiation and handling of collected data in clause 8.17, the following additional requirements are placed on NGNs:

1) An NGN is recommended to be able to identify data according to relevant categories.

2) An NGN is recommended to apply different data handling (e.g., caching and/or forwarding) based on data identification.

9.1.8 Application collaboration and environment for service integration and delivery

Based on the service requirements specified in clauses 8.5 and 8.6, the following additional requirements are placed on NGNs:

1) An NGN is required to provide capabilities for application collaboration and for a service integration and delivery environment.

   NOTE – [ITU-T Y.2240] capabilities may be used for the support of such requirements.

9.2 Requirements supported by existing NGN capabilities

Based on the service requirements in clause 8, this clause specifies requirements supported by existing NGN capabilities for the support of MOC applications.

9.2.1 Group management

The group management requirements specified in clause 8.12 are supported by the existing capabilities of NGN [ITU-T Y.2201].

9.2.2 Location management

NGN provides the location management capability which determines and reports information regarding the location of users and devices within the NGN. The location management requirements specified in clause 8.11 are supported by the existing capabilities of NGNs [ITU-T Y.2201].

9.2.3 Security

An NGN provides security capabilities. The service requirements specified in clause 8.13 are supported by the existing capabilities of NGNs [ITU-T Y.2201] [ITU-T Y.2701].

9.2.4 Group related communication modes

Based on the service requirements in clause 8.12, an NGN is required to support the following communication modes for MOC groups (with MOC devices and gateways directly or indirectly connected to NGNs):

– any cast
– multicast
– broadcast.

These requirements are supported by the existing capabilities of NGNs [ITU-T Y.2201].

10 Capability requirements of an MOC device domain

This clause identifies the capability requirements of an MOC device domain for the support of MOC applications.

10.1 Application enablement

Based on the requirements in clause 8.4, the following requirements apply to the MOC gateway and MOC device's capabilities:

1) MOC devices are recommended to support a set of abstracted operations.
2) MOC devices are recommended to support the implementation of service logic to provide MOC capabilities.
3) MOC gateways can optionally support a set of abstracted operations on MOC devices.
4) MOC gateways can optionally support the implementation of service logic to provide MOC capabilities.

10.2 Mobility

Based on the requirements in clause 8.1, the following requirements apply to the MOC gateway and device's capabilities:

1) MOC gateways and MOC devices are required to support enhanced mobility management capabilities in order to support different levels of mobility.

10.3 Communication

Based on the requirements in clauses 8.2, 8.3 and 8.4, the following requirements apply to the MOC gateway and MOC device's capabilities:

1) MOC gateways and MOC devices are required to be able to establish, maintain or release communication resources according to the data communication needs.

2) MOC gateways are required to be able to select the proper routing paths between the traffic originating endpoint (MOC devices or MOC application server) and the traffic receiving endpoint (MOC devices or MOC application server) according to the application the MOC device is associated with or vice versa.

3) MOC gateways are required to allow the setting and modification of granted/forbidden network communication access time schedules and durations.

4) MOC gateways are required to support the following communication modes according to the service requirements:
   - any cast
   - multicast
   - broadcast.

5) MOC gateways are recommended to support communication with proprietary devices (e.g., devices with proprietary interfaces for inter-working with network entities).

6) MOC devices are required to go offline when no data transmission is required and then to go into sleep mode according to the necessary policies.

   NOTE – The transition to sleep mode can also be controlled by the network.

7) MOC devices can optionally support communication with the network according to the following criteria: daily network traffic load, MOC device location, access time schedules and durations.

10.4 QoS

Based on the requirements defined in clauses 8.2 and 8.13, the following requirements apply to the MOC gateway and MOC device's capabilities:

1) MOC gateways and MOC devices are required to support the traffic control policy which defines granted network communication access time schedules and durations.

2) MOC gateways and MOC devices are required to support QoS differentiation according to different categories of traffic.

3) MOC gateways and MOC devices are required to provide performance measurement and management.

4) MOC gateways and MOC devices are recommended to support application prioritization.

10.5 Remote management

Based on the requirements in clause 8.9.1, the following requirements apply to the MOC gateway and MOC device's capabilities:

1) MOC gateways are required to act as a management proxy for MOC devices of the connected MOC local network. This includes support of management requests from NGN and local (to the MOC local network) firmware and software management.

2) MOC devices are required to support software and firmware management.

3) MOC gateways and MOC devices are required to support configuration management.

4) MOC gateways are required to support fault and performance data collection and storage.

5) MOC devices are recommended to support fault and performance data collection and storage.

10.6 Device addressing and identification

Based on the requirements in clauses 8.10 and 8.12, the following requirements apply to the MOC gateway and MOC device's capabilities:

1) MOC gateways are required to support mapping between the identification of an MOC device and one or more MOC local network addresses.
2) MOC gateways are required to support mapping between the identification of an MOC device group and one or more MOC local network addresses for each MOC device within the group.
3) MOC devices are required to support unique identification within the context of a single MOC application.
4) An MOC gateway can optionally use temporary identifiers for MOC devices connecting and disconnecting to the network dynamically.
5) An MOC gateway can optionally re-assign dynamically released temporary identifiers to other MOC devices.

10.7 Security

Based on the requirements in clause 8.14, the following requirements apply to the MOC gateway and MOC device's capabilities:

1) MOC gateways are required to identify and authenticate MOC applications, other MOC devices and MOC end users.
2) MOC gateways are required to support mechanisms for secure transport, such as encryption and integrity protection.
3) MOC devices are recommended to identify and authenticate MOC applications, other MOC devices and MOC end users.
4) MOC devices are recommended to support mechanisms for secure transport, such as encryption and integrity protection.

10.8 Accounting and charging

Based on the requirements in clause 8.8, the following requirements apply to the MOC gateway capabilities:

1) MOC gateways are recommended to support different accounting and charging methods for the connected MOC devices.

10.9 Data identification

Based on the requirements of collected data differentiation and related data handling in clause 8.17, the following requirements apply to the MOC device and MOC gateway's capabilities:

1) MOC gateways are recommended to make data identifiable according to relevant policies.
2) MOC devices can optionally make data identifiable according to relevant policies.

11 Reference framework for MOC capabilities

11.1 High-level view

Figure 11-1 provides a high-level view of the reference framework for MOC capabilities.

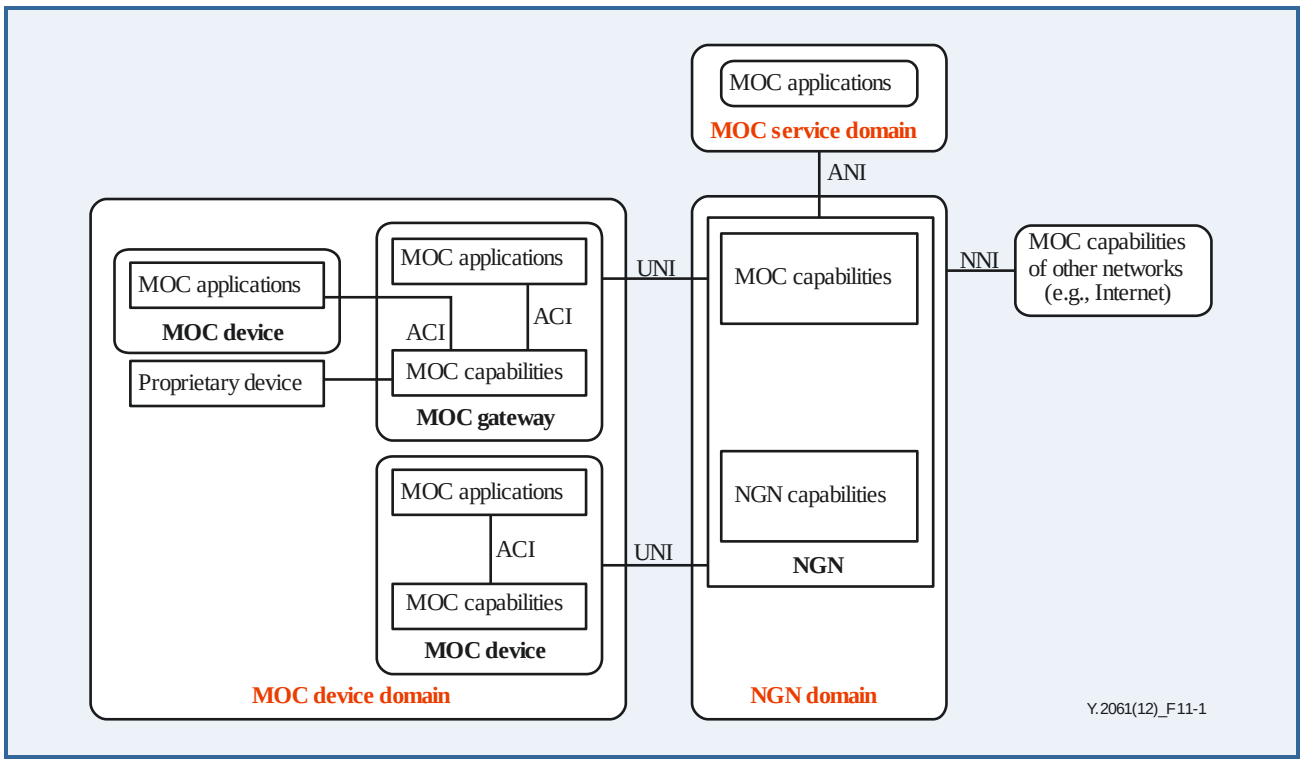


Figure 11-1 – High-level view of the reference framework for MOC capabilities

The MOC device domain is composed of MOC devices, proprietary devices and MOC gateways. MOC capabilities of the MOC device domain collaborate with MOC capabilities and NGN capabilities of the NGN domain to support MOC applications.

The interface between MOC devices or gateways and the NGN is the user to network interface (UNI).

The interface between MOC capabilities and MOC applications residing in the MOC device domain is the application to capability interface (ACI).

The NGN domain is composed of:

– NGN capabilities (modified and extended as necessary for the support of MOC applications as per clause 9);

– MOC capabilities.

The interface between the NGN and other networks is the network to network interface (NNI).

The MOC service domain is composed of MOC applications.

The interface between the NGN and MOC applications residing in the MOC service domain is the application to network interface (ANI).

In order to support MOC applications, service interfaces are provided across ACI, ANI, NNI and UNI. The requirements of these service interfaces are given in clause 11.4.

11.2 MOC capabilities in the NGN domain

The main MOC capabilities in the NGN domain are shown in Figure 11-2. These capabilities provide standard interfaces for MOC applications to MOC devices and gateway for data collection, management and operations.

The MOC capabilities shown in Figure 11-2 are [ITU-T Y.2240] capabilities adapted as necessary to the context of MOC.

In line with [ITU-T Y.2240], the MOC capabilities also interact with NGN capabilities [ITU-T Y.2201], IT capabilities or Internet capabilities.

NOTE 1 – The MOC capabilities in the NGN domain are positioned inside the NGN service stratum [ITU-T Y.2012].

NOTE 2 – Other [ITU-T Y.2240] capabilities are not precluded to be part of, or adapted as, MOC capabilities in the NGN domain as shown in Figure 11-2.

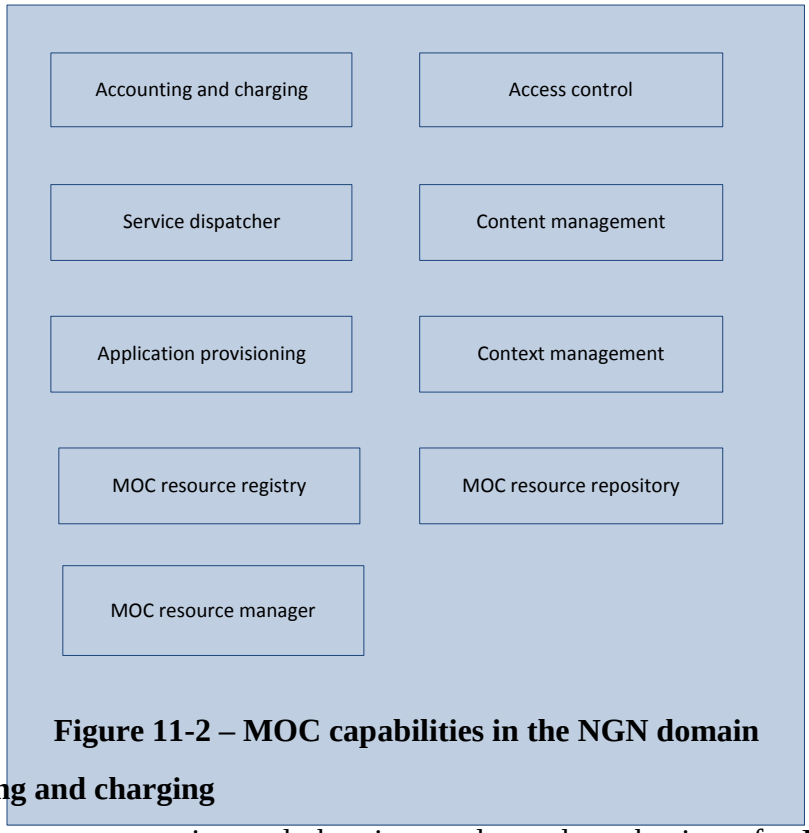


Figure 11-2 – MOC capabilities in the NGN domain

11.2.1 Accounting and charging

This capability supports accounting and charging modes and mechanisms for MOC applications, including:

– support of revenue sharing among the various actors involved in the MOC ecosystem

– support of event-based online or offline charging in the MOC ecosystem.

For further details on this capability see [ITU-T Y.2240].

11.2.2 Access control

This capability performs authentication and authorization of MOC applications before allowing them to access a specific set of capabilities. The access control capability provides translation of application programming interfaces (APIs) and protocols across different service interfaces as well as access from applications to functionalities exposed by MOC capabilities.

For further details on this capability see [ITU-T Y.2240].

11.2.3 Service dispatcher

This capability provides unified message routing and message exchange mechanisms among the MOC capabilities in the NGN domain.

The service dispatcher also provides API and protocol transformation from MOC applications to common message structure and business event handling and vice versa.

For further details on this capability see [ITU-T Y.2240].

11.2.4 Content management

Content can be provided as resources to an MOC application or end users by different MOC device providers (e.g., content providers and end users).

The content management capability provides the extraction of appropriate information (including size, type, location) from content, enabling the MOC capabilities in the NGN domain to ensure the integrity of the content itself.

The content management capability provides profiling of content as appropriate to enable its delivery to different MOC applications, such as content for specific MOC applications or content for a specific MOC end user.

The content management capability provides dispatching of content in order to expose content to applications.

For further details on this capability see [ITU-T Y.2240].

11.2.5 Application provisioning

This capability is used for the deployment of applications in a secure way by the MOC application provider when applications are available for deployment. This capability provides application packaging, publishing, deployment, lifecycle management and monitoring functions.

For further details on this capability see [ITU-T Y.2240].

11.2.6 Context management

This capability collects, aggregates and manages context information related to different context sources, exposing context information, including to other MOC capabilities, according to the MOC application provider's policies.

For further details on this capability see [ITU-T Y.2240].

11.2.7 MOC resource registry

This capability provides the functionalities related to the registration, de-registration, discovery and governance of resources offered by MOC device providers. Registration-oriented resource descriptions including unique resource identification and resource addressing are published in the MOC resource registry.

This capability defines a mechanism for a resource of an MOC device provider to be registered within the MOC capabilities in the NGN domain, so that this resource can be located and accessed by applications.

This capability corresponds functionally to the resource registry capability in [ITU-T Y.2240] when adapted to the context of MOC.

11.2.8 MOC resource repository

This capability provides functionalities for the storage of information related to the registered MOC resources. Being stored in the MOC resource repository, the MOC resource information can be accessed by the authenticated and authorized MOC applications. Information related to the registered resources includes various suitable packaging tools for application developers.

This capability corresponds functionally to the resource registry capability in [ITU-T Y.2240] when adapted to the context of MOC.

11.2.9 MOC resource manager

This capability performs the control functions for resources provided by NGN and MOC devices in order to satisfy the MOC applications' requirements, including the management of the dynamic

information concerning the reachability status of MOC devices (e.g., online/offline, forwarding information).

This capability corresponds functionally to the resource registry capability in [ITU-T Y.2240] when adapted to the context of MOC.

11.3 MOC capabilities in the MOC device domain

The MOC capabilities in the MOC device domain are shown in Figure 11-3.

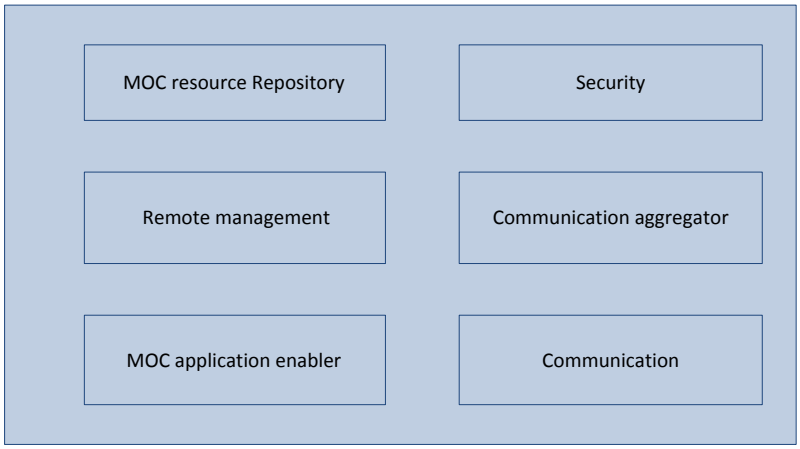


Figure 11-3 – MOC capabilities in the MOC device domain

11.3.1 Communication aggregator

This capability, residing in MOC gateways, performs proxy and traffic aggregator functions.

Based on the requirements identified in clause 10.3, the communication aggregator capability supports the relaying of traffic between MOC devices and NGNs, QoS mechanisms based on the MOC application policy and the recording of charging information for MOC devices within the MOC local network.

11.3.2 MOC application enabler

Based on the requirements identified in clause 10.1, this capability performs the exposure to MOC applications of MOC capabilities residing in the MOC gateway and MOC devices.

11.3.3 MOC resource repository

Based on the requirements identified in clause 10.1, this capability, residing in MOC gateways, provides functionalities for the storage of information related to the registered MOC devices. Being stored in the MOC device repository, the MOC devices' data can be read by the authenticated and authorized MOC applications.

11.3.4 Remote management

Based on the requirements identified in clause 10.5, this capability supports configuration management, fault management, performance management, software and firmware upgrade of the MOC gateway and MOC devices in the MOC local network.

NOTE – The functions above can also be realized by local management capabilities of the MOC devices and gateway.

11.3.5 Security

Based on the requirements identified in clause 10.7, this capability performs registration and mutual authentication with MOC applications.

The security capability provides encryption and integrity protection on data exchanged with the NGN and MOC applications to ensure secure delivery.

The security capability performs key management based on the service keys generated in MOC devices, and protects from unauthorized applications' access to MOC devices.

11.3.6 Communication

Based on the requirements identified in clauses 10.3, 10.6, 10.8 and 10.9, this capability performs the generic communication functions in the MOC devices and gateway.

The communication capability provides application data transport functions, delivers and receives data to/from MOC applications in accordance with the service criteria (e.g., daily network traffic load, MOC device location, access time schedules and durations), and handles these data.

11.4 MOC service interfaces

The MOC service interfaces support:

– MOC applications hosted in MOC devices and gateways, which access NGNs via UNI;

– MOC applications hosted in MOC devices and gateways, which access the MOC capabilities hosted in the MOC gateway via ACI;

– MOC applications hosted in MOC devices, which access the MOC capabilities hosted in those same MOC devices via ACI;

– MOC applications hosted in the MOC service domain, which access the NGN via ANI.

– MOC applications hosted in the MOC service domain, which access the MOC capabilities of other networks via NNI.

MOC applications hosted in MOC devices and gateways can be invoked by an MOC end user and other MOC applications, e.g., MOC applications hosted in the MOC service domain.

MOC service interfaces are recommended to implement standardized APIs, protocols and technologies to realize the service exposure towards MOC applications.

11.4.1 Service interface requirements across ACI

The service interface across the ACI is used to provide interaction in the MOC device domain between the MOC capabilities of devices and gateways and MOC applications of devices and gateways. The service interface across the ACI allows an application residing in an MOC device to access MOC capabilities in the same MOC device or in an MOC gateway. It also allows an application residing in an MOC gateway to access MOC capabilities in the same MOC gateway.

The ACI is required to enable the following functions:

– registration of the MOC devices and gateway to the MOC capabilities in the MOC device and gateway (e.g., registration of a sensor or GPS in a car to the gateway in the car);

– MOC applications' execution requests of an MOC device-specific tasks by an MOC device and gateway or group of MOC devices and gateways;

– subscription and notification to specific events (e.g., mutual subscription and notification to specific events (e.g., connectivity of the MOC devices and gateway) between MOC capabilities and applications);

– the MOC devices and gateway's requests of group creation, deletion and members' listing.

11.4.2 Service interface requirements across UNI

The service interface across the UNI is used to provide interaction between the MOC capabilities of MOC devices and gateways and the MOC capabilities of the NGN. The UNI is recommended to

support standardized APIs for exposing resources provided by the MOC device domain to the MOC capabilities in the NGN domain.

The UNI is required to enable the following functions:

– registration of MOC capabilities in the MOC device domain to the MOC capabilities in the NGN domain;
– request from MOC devices of the execution of a specific task to be performed by an MOC application;
– subscription and notification of specific events from/to the MOC device domain;
– requests of group creation, deletion and members' listing.

11.4.3 Service interface requirements across ANI

The service requests initiated by MOC applications are sent to the MOC capabilities in the NGN domain via the ANI. The service interface across the ANI is required to provide the interaction between MOC applications in the MOC service domain and MOC capabilities in the NGN domain.

The ANI is recommended to support standardized APIs for exposing resources provided by the NGN domain to MOC applications.

The ANI is required to enable the following functions:

– registration of MOC applications to the MOC capabilities in the NGN domain;
– request from MOC applications of the execution of a specific task to be performed by an MOC device and gateway or group of MOC devices and gateways;
– subscription and notification of specific events from/to MOC applications;
– requests of group creation, deletion and members' listing.

11.4.4 Service interface requirements across NNI

The service interface across the NNI is used to provide interaction with MOC capabilities of other networks. The following service interface across the NNI is relevant for the interaction of NGNs with MOC capabilities of other networks:

– service interface between NGNs and other networks (NGNs or non-NGNs), both of which have capabilities for supporting MOC applications.

NOTE – The scenario of NNI interaction between an NGN and other networks which have no capabilities for the support of MOC applications is not relevant as it implies only transport level interaction.

12 Security considerations

Security requirements for MOC applications are described in clause 8.14.

Security requirements for the NGN domain are provided in clause 9.2.5.

Security requirements for the MOC device domain are provided in clause 10.7.

Appendix I

Actors and related roles in the MOC ecosystem

(This appendix does not form an integral part of this Recommendation.)

The following items identify different actors of the MOC ecosystem and the business roles (see clause 6.3) that they can play:

– **The "carrier" actor**. The "carrier" actor plays the role of network provider. Depending on the business scenario, the "carrier" actor may also play the role of MOC application provider, MOC platform provider and MOC device provider.
– **The "third party application provider" actor**. The "third party application provider" actor plays the role of MOC application provider. Examples of third party application providers include web-based application providers. The "third party application provider" actor may also play (but is not limited to) the role of MOC platform provider.
– **The "third party device provider" actor**. The "third party device provider" actor plays the role of MOC device provider. Examples of third party device providers include device operators, end users. The "third party device provider" actor may also play (but is not limited to) the roles of MOC platform provider, MOC application provider and MOC application customer.

Appendix II

MOC use cases

(This appendix does not form an integral part of this Recommendation.)

II.1 e-Health

e-Health is a relatively recent term used to designate healthcare practices supported by electronic processes and communications.

Figure II.1 shows an example of e-Health service configuration.

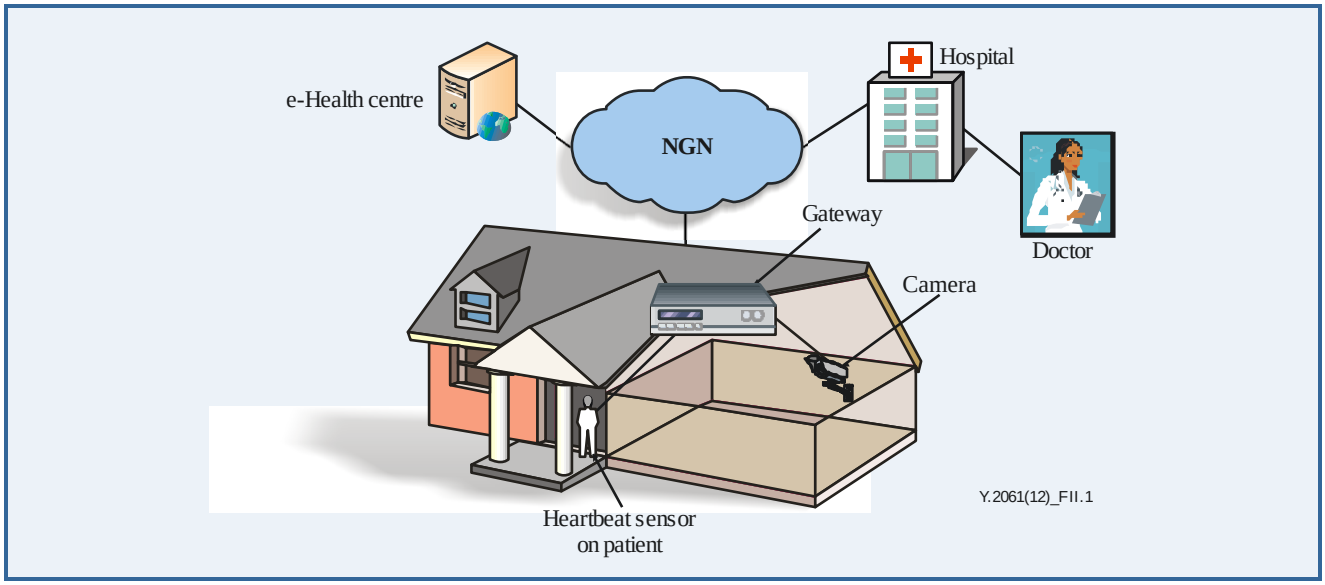


Figure II.1 – e-Health service configuration

Various types of devices are involved in the provisioning of e-Health services. Some of these devices only collect data and interact with the network (e.g., heartbeat sensors), others can interact bidirectionally (e.g., cameras), some devices usually generate small amounts of data (e.g., thermometers), while others may deal with multimedia streaming (e.g., cameras) or, deal with call session control (e.g., SIP terminals supporting video calls). Some devices may even work as both gateway and sensor-like service platforms.

The e-Health devices gather data and send them to the relevant parties, such as the e-Health centre in Figure II.1. Hospitals, doctors and families can subscribe to the service to get raw or processed data.

The devices associated with patients can access the network directly or via a gateway(s) (e.g., home gateway or a gateway worn on the body):

1) When the patients stay in an indoor environment, the devices can access the network via a single static home gateway or via multiple dynamic home gateways (in this second case the patients can move and access the network via different gateways).

2) When the patients are outdoors, the devices can access the network directly via a mobile network or indirectly via the gateway worn on the body.

The following technical challenges need to be considered for e-Health:

– Grouping should be supported. This is useful, for instance, for multiple patients with the same type of disease, or in the case of a single patient, to manage a set of devices which can be managed in group mode.

– Optimized traffic control should be supported. For example, the detected data may be very small and need to be reported to the network every hour: in such a case, it is a waste of resources to be permanently connected to the network. The network should be optimized in terms of traffic control, and, in such a case, traffic could be delivered, for example via user plane signalling without a data-dedicated IP bearer. Additionally, devices on a patient might stay in sleep mode and wake up when the doctor needs to diagnose the patient remotely.
– Different mobility levels should be supported. For instance, in the case of patients with poor mobility (moving infrequently and not very far), it is a waste of resources to activate full mobility management capabilities.
– Remote device activation and management should be supported. For example, devices in sleep mode would be woken up only when the doctor needs to diagnose the patient remotely.
– Time control should be supported. For instance, devices on patients may collect a lot of data but do not always need to report them at every collection as the data may be not very critical, e.g., only for routine examination. In these scenarios, the network can allocate specific time slots for the devices' data to be reported (the devices cannot report data during other time periods or are charged at higher rates in those periods).
– Device profiles should be supported. Patient may buy new devices and connect them to the network dynamically: device related information should be included in the device profile and be updated dynamically to enable the network authentication and control of the newly-added devices and also their removal.
– Devices behind a gateway should be able to be identified by the network. The gateway might provide only a bearer channel and act as a data aggregator for the devices connected to it or might provide service control for the devices connected to it. In the first case, the devices connected to the gateway should be controlled by the network, or by both the network and gateway.
– Proprietary devices should be supported. There are plenty of proprietary devices and gateways running in networks: adaptation to existing proprietary devices and gateways should be supported.
– Service profile should be supported. Patients are usually not very familiar with the services offered by different hospitals, they can usually just logon to the e-Health centre's portal and access services, whereas the e-Health centre is usually familiar and can determine the target hospitals based on their professional knowledge. There might be one or multiple hospitals providing medical services to a patient jointly. In other words, when the devices on a patient report data to the e-Health centre, the centre can intelligently help the patient to select the best target hospitals and route the data to those hospitals for joint diagnosis. Multimedia call control sessions might be needed in this scenario, including audio, video, text messaging, etc. The devices should also interact with multiple applications.

When doctors diagnose and provide healthcare services remotely, they usually also need the existing internal disease diagnostics system or database system of the hospital for assistance: data reported from devices may be input to the existing hospital internal system. In this case, the devices should be interoperable with existing systems (e.g., data format, service capabilities invocation, etc.), that is the e-Health system should be able to collaborate and inter-work with the existing application systems which are usually heterogeneous.

– Traffic load balancing should be taken into consideration in order to cope with particular situations. For instance, due to the number of patients in areas varying quite considerably, there may be relatively high rates of patients in geriatric wards, communities of elderly people and certain cities, as compared with other cities. The network should be able to handle accordingly the system imbalance in case of specific situations of high traffic or

service load, especially for video-like services (for example, when a lot of patients use remote video diagnosis).

II.2 Tsunami warning service

The tsunami warning system is used to detect tsunamis and issue warnings to prevent loss of life and property.

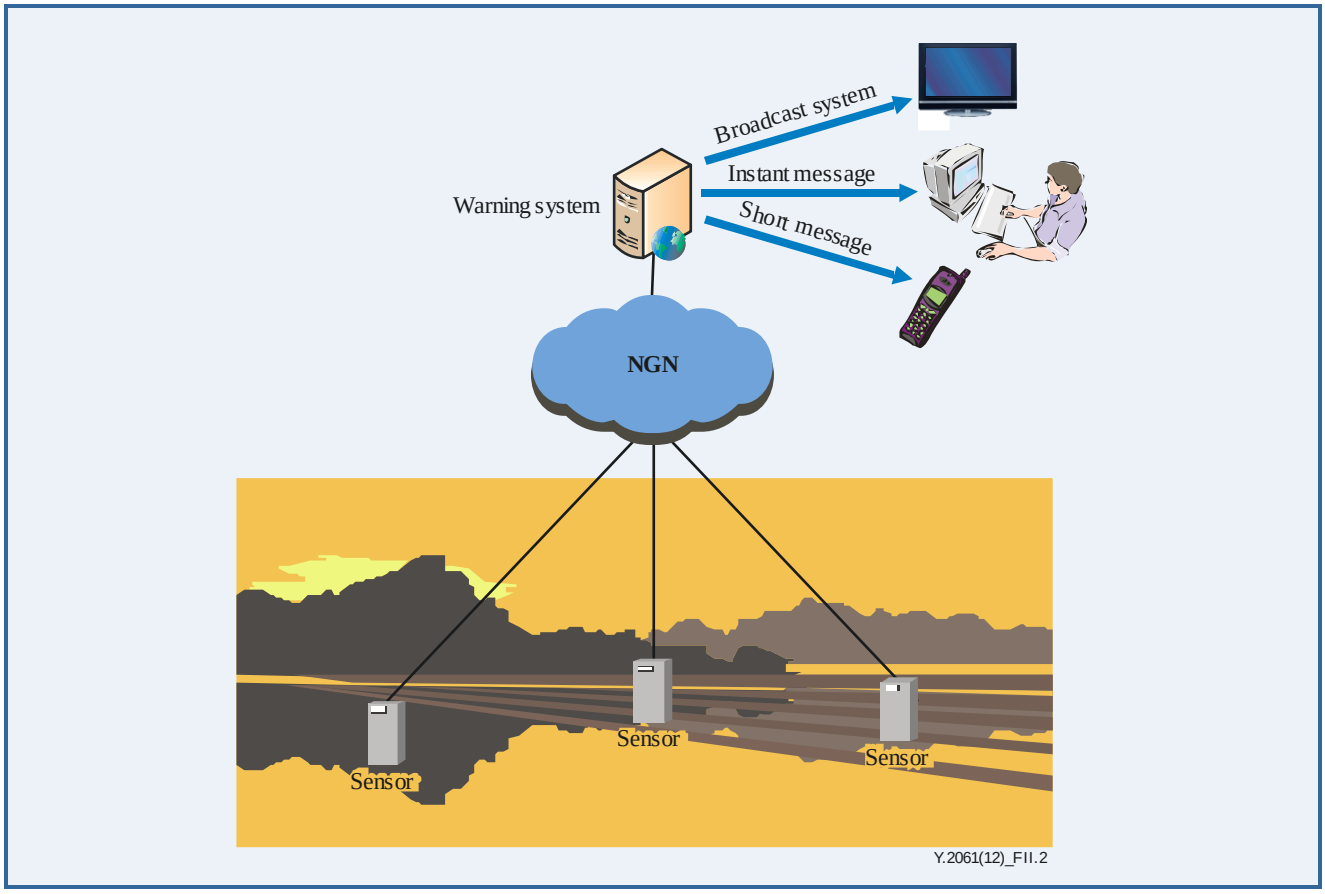


Figure II.2 – Tsunami warning service configuration

As shown in Figure II.2, it consists of two equally important components: a network of sensors to detect tsunamis and a communications infrastructure to issue timely alarms to help evacuation of coastal areas. Detection and prediction of tsunamis is only half the work of the system. The other equal importance is the ability to warn the populations of the areas that will be affected. To save lives more certainly, proper guidance for escape according to their situation in danger (e.g., time, place, and occasion) should be considered. For a visitor who comes to an unfamiliar area at night, a simple alarm is not enough to escape to a safe place. All tsunami warning systems feature multiple lines of communications (such as SMS, e-mail, fax, radio, text and telex, often using hardened dedicated systems) enabling emergency messages to be sent to the emergency services and armed forces, as well to population alerting systems (e.g., sirens). In this use case, the service is required to support:

– inter-working with heterogeneous network, including: mass media networks (e.g., radio network, television network) and dedicated communication systems (e.g., sirens);

– delivery of emergency information, including both the primary information generated by the detector and the secondary information transferred to the target population;

– delivery of emergency information over multiple networks, including reliable and unreliable bearers (e.g., communication through satellite systems) to maximize the probability of the delivery;

NOTE – Information integrity which might be compromised by reliable and unreliable bearers needs further study.

– system robustness, i.e., the system should support information bursting within a short while due to a large number of machines (e.g., emergency detectors or sensors) within an area;

– prioritized delivery of emergency information, i.e., emergency message for an earthquake, should be prioritized compared with other service messages.

II.3 Motorcade management

Figure II.3 shows a typical service configuration for motorcade management.

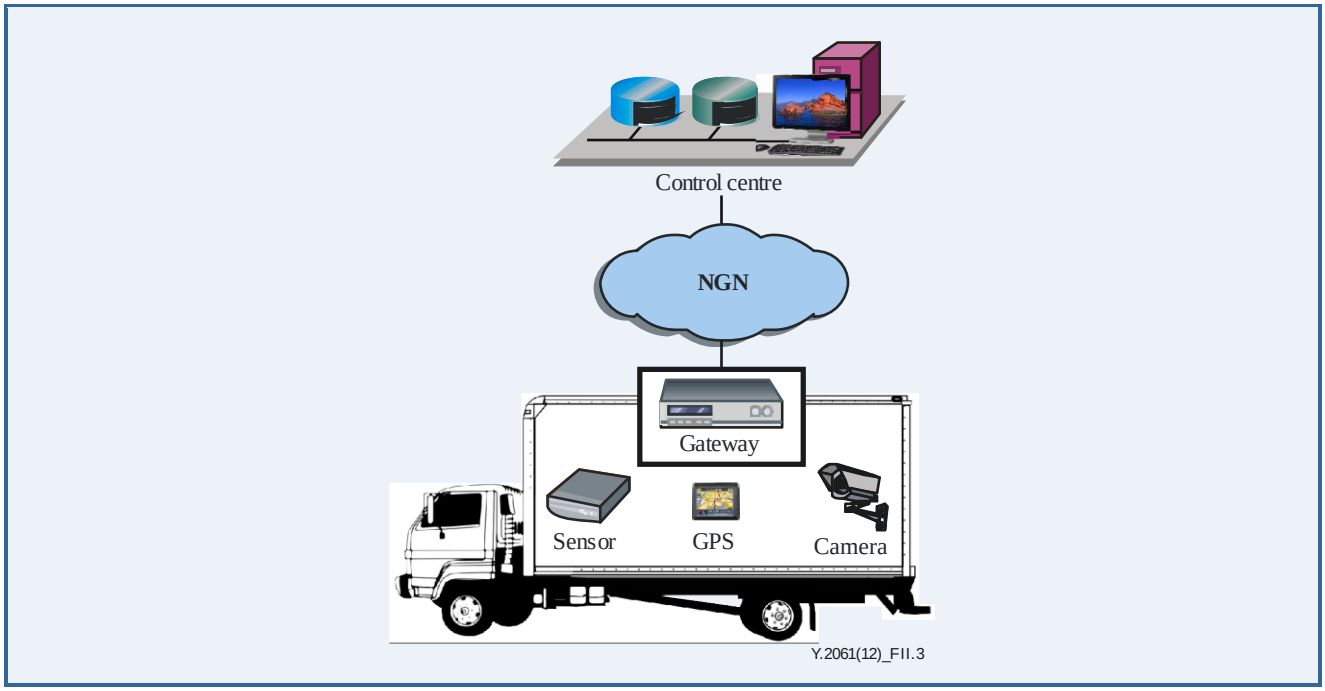


Figure II.3 – Typical motorcade management service configuration

Every bus is equipped with devices and gateways which have the same characteristics. The control centre gathers data related to location, speed and the situation given from the sensors, global positioning system (GPS) terminal and cameras of the bus. Data aggregated through a gateway located on the bus are transmitted to the NGN using wireless access.

The dynamic timetable can be forwarded to the monitor screen on the bus stop by the control centre according to the location information collected from the bus.

When a sensor on the bus detects an abnormal situation, such as the smell of gasoline, an alarm indication is sent to the control centre.

The bus always has a fixed route which means it should not move out of the pre-defined roads. When a bus moves out of a particular area, an application should be triggered. For example, a call may be made to the bus driver, or an alert indication may be made to the bus administrator while the bus moves out of the area.

In this use case, the service is required to support:

– location based service: an application should be triggered when devices are in or out of a particular area;

– prioritized service level, for example, alarm indication should be prioritized compared with other data;
– group management for devices with the same characteristics.

II.4 Smart home

Smart home usually involves a mix of different devices and applications, such as real-time or near real-time sensors, power outage notification and power quality monitoring.

Figure II.4 shows a typical "smart home" configuration.

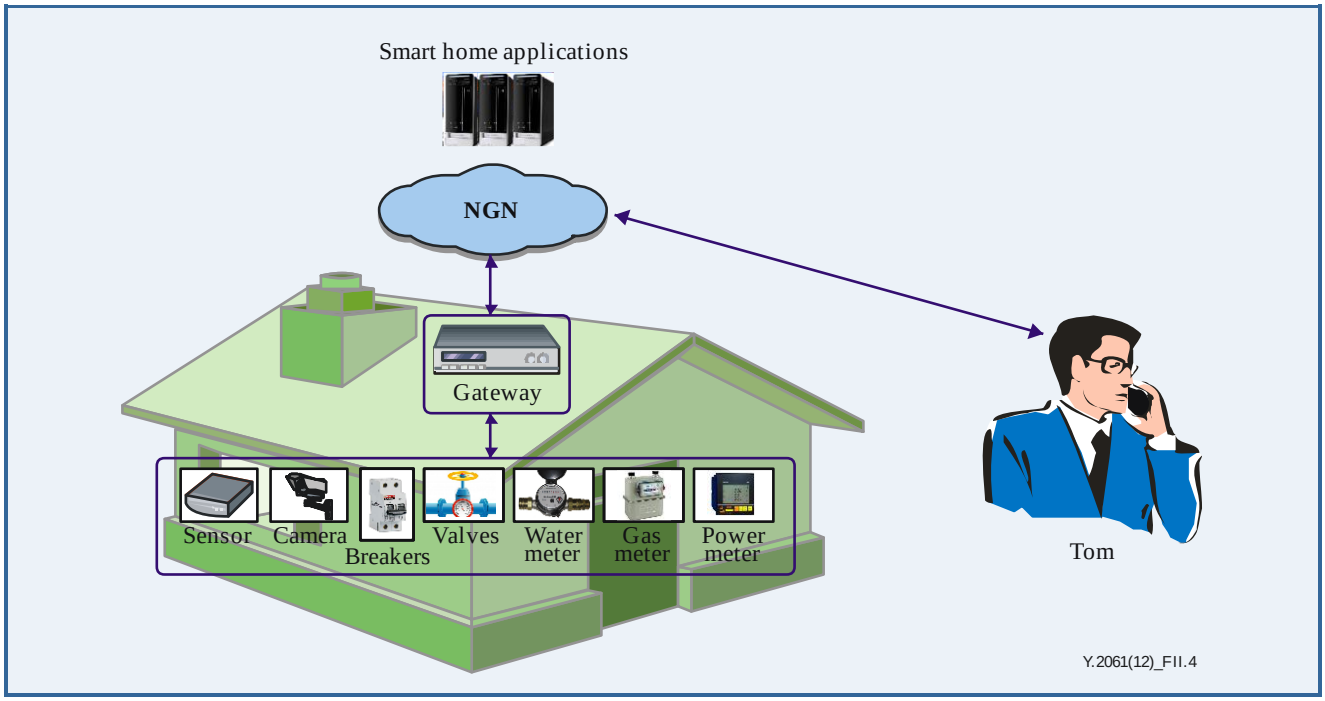


Figure II.4 – Typical smart home service configuration

As shown in Figure II.4, a "smart home" scenario often refers to devices (e.g., smoke sensor, electricity meters, gas meters, etc.) which are connected to a smart home application platform via a gateway located in the smart home. The data centre collects data from the "smart home" devices and is able to control these devices remotely via the gateway. In this scenario, Tom's house information related to power, gas and water consumption can be collected and reported to the smart home applications platform. At the same time, Tom can manage the application related policy of his home using the smart home applications and the application related policy can be sent to MOC devices in order to be executed according to Tom's requirements.

For example, Tom defines the application related policy as follows:

1) If a fire sensor detects signs of a fire in Tom's house, then the fire sensor shall send a short message alarm to Tom's mobile phone.
2) If an alarm is triggered by a door break-out sensor then a video communication is initiated allowing Tom to see in real time what is happening inside his house.

Let us assume that a thief breaks open a door of Tom's house. When detecting this event, the MOC device (i.e., the door break-out sensor) initiates a video communication between Tom and a visual surveillance camera located in Tom's home. Tom watches and records the video on his mobile, (a record which may be used as evidence of the crime).

Let us now consider that Tom is out of his house while a fire occurs in his kitchen where his son is cooking. When detecting this event, the MOC device (i.e., the smoke sensor) sends an alarm

message to Tom directly. Upon receipt of this information, Tom initiates a video communication with the camera to check the status of the kitchen, and to tell his son how to use the fire extinguisher or to exit. For privacy and security reasons, the camera is only connected and controlled by members of Tom's family.

In this use case, the service is required to support:

– enhanced video/audio based capabilities, such as concurrent video streaming and local-breakout;

– group management for MOC devices with the same characteristics, for example, power meters in different smart homes;

– message broadcasting and multicasting based on specific characteristics, such as group and location, to support functions such as firmware upgrading.

II.5 Integration with Internet services

There are many attractive services emerging on the Internet, such as social network services (SNS). MOC applications should be able to work with those Internet services to ensure that customers can use the MOC applications with existing popular Internet services. Integrated with Internet services, MOC applications themselves will extend their value chains and attract more customers.

In some integration scenarios, MOC capabilities should be able to apply data detection rules (i.e., setting rules) to the MOC devices and gateway. Once detected, the data should be transferred to the MOC capability in a defined format. The formatted data facilitates the MOC capability to communicate smoothly with the Internet services that provide publishing services.

Figure II.5 shows an example use case of the integration of MOC application and Internet service.

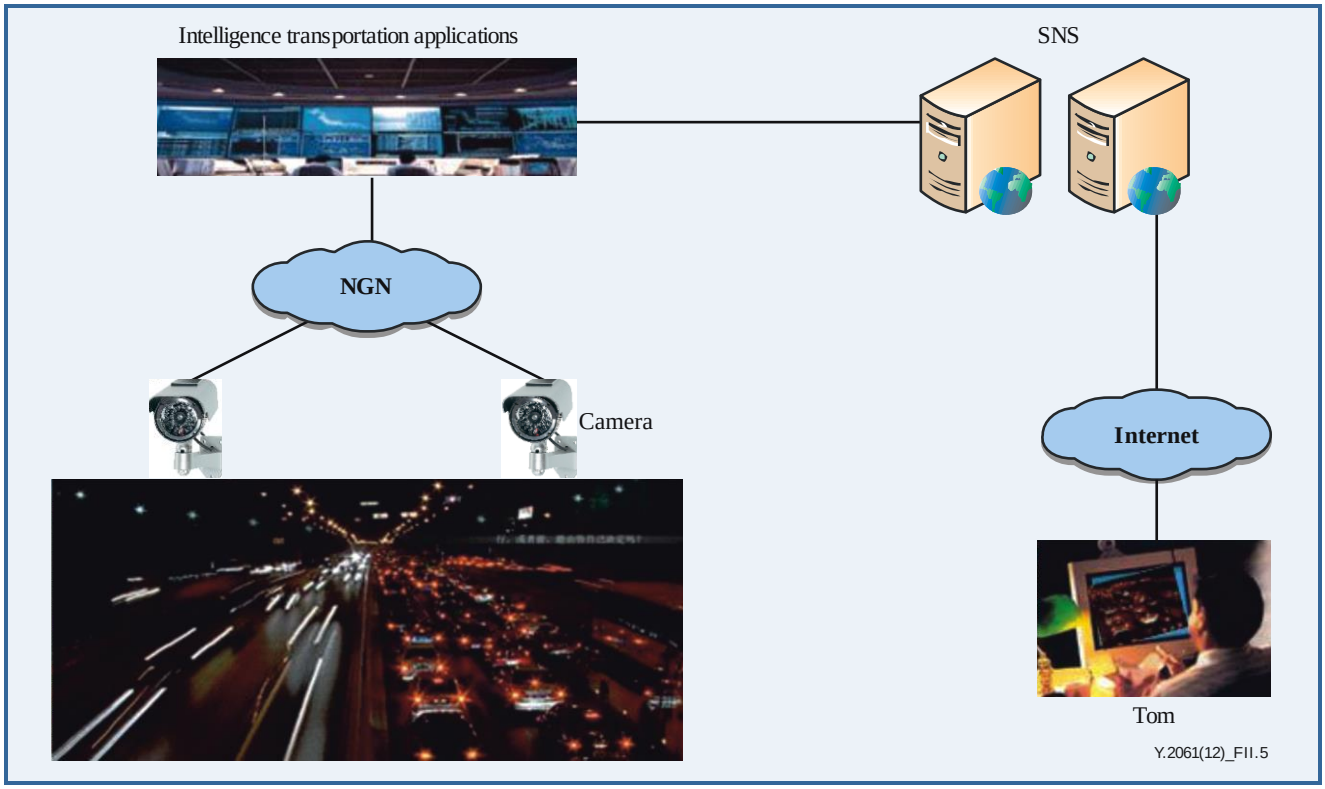


Figure II.5 – Typical internet service configuration

AN MOC application provider offers intelligence transportation service for customers. The service allows the customers to access their interested content via SNS with their preferred rules (e.g., publishing times). The provider collects the content by an MOC application and provides it for the customers via SNS according to the customers' rules.

Tom finds that the intelligence transportation service provides real time pictures and videos of city traffic captured by the cameras on the streets and that information is provided via SNS. Tom subscribes to this service (e.g., regularly updated every five minutes) and receives the instantaneous traffic information on his SNS.

The service updates the information via SNS according to Tom's preference. Tom can find out the highway traffic information on his way home.

In this use case, the service is required to support:

– integration with Internet services using the MOC capabilities;
– setting rules to detect the MOC devices and gateway's data and transfer the data with a defined format to the capability which is used to communicate with the Internet services for publishing;
– enabling customers to access relevant MOC content via the Internet services with defined rules;
– detecting the relevant MOC content and provide it to the Internet services based on the rules;
– communicating with the Internet services to exchange the information.

Bibliography

[b-ITU-T Y.2001] Recommendation ITU-T Y.2001 (2004), *General overview of NGN*.

[b-ITU-T Y.2011] Recommendation ITU-T Y.2011 (2004), *General principles and general reference model for Next Generation Networks*.

[b-ITU-T Y.2213] Recommendation ITU-T Y.2213 (2008), *NGN service requirements and capabilities for network aspects of applications and services using tag-based identification*.

[b-ITU-T Y-Sup.7] ITU-T Y-series Recommendations – Supplement 7 (2008), *Supplement on NGN release 2 scope*.

[b-ITU-T Q.1741.7] Recommendation ITU-T Q.1741.7 (2011), *IMT-2000 references to Release 9 of GSM-evolved UMTS core network*.

[b-ETSI TS 102 689] ETSI TS 102 689 V1.1.1 (2010), *Machine-to-Machine communications (M2M); M2M service requirements*.

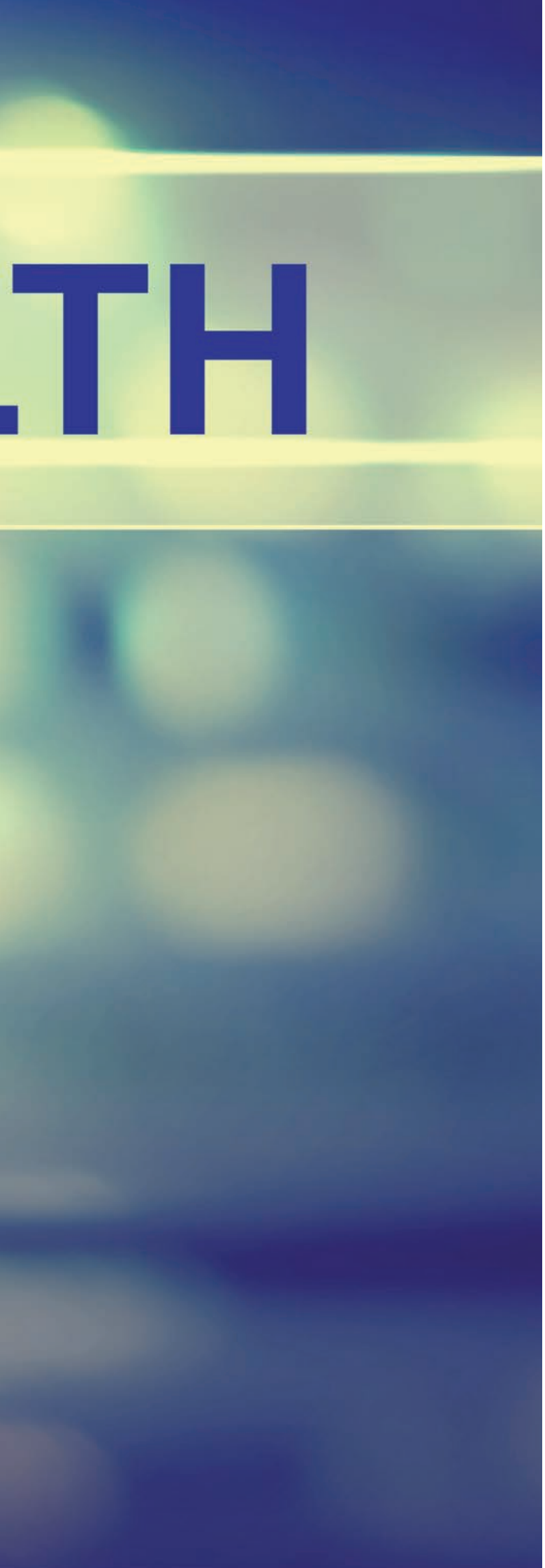
[b-ETSI TS 102 690] ETSI TS 102 690 V1.1.1 (2011) *Machine-to-Machine communications (M2M); Functional architecture*.

[b-3GPP TS 22.368] 3GPP TS 22.368 V 11.3.0 (2011), *Service requirements for Machine-Type Communications (MTC)*.

[b-3GPP2-S.R0141-0] 3GPP2 S.R0141-0 V.1.0 (2010), Study for *Machine-to-Machine (M2M) Communication for cdma2000 Networks*.


INTERNET
OF THINGS
31
SALE


E-HEAL



Y.4110/Y.2065

Service and capability requirements for e-health monitoring services

Service and capability requirements for e-health monitoring services

Summary

Recommendation ITU-T Y.2065 provides service and capability requirements for e-health monitoring services.

Three classes of e-health monitoring services, including their general and specific characteristics, are described. Service requirements for the support of e-health monitoring services are also described, and based on the identified service requirements, the capability requirements are specified.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T Y.2065	2014-03-22	13	11.1002/1000/12072

Keywords

Capability requirements, e-health monitoring services, service requirements.

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

Table of Contents

Recommendation ITU-T Y.4110/Y.2065

Service and capability requirements for e-health monitoring services

1 Scope

This Recommendation describes the service requirements for the support of e-health monitoring services, and it specifies the corresponding capability requirements.

The scope of this Recommendation includes:

– classification of e-health monitoring services;

– description of characteristics of e-health monitoring services;

– service requirements for supporting e-health monitoring services;

– capability requirements for supporting e-health monitoring services.

Relevant service scenarios of e-health monitoring services are provided in Appendix I.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T Y.2060] Recommendation ITU-T Y.2060 (2012), *Overview of the Internet of things*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 device [ITU-T Y.2060]: With regard to the Internet of things, this is a piece of equipment with the mandatory capabilities of communication and the optional capabilities of sensing, actuation, data capture, data storage and data processing.

3.1.2 Internet of things (IoT) [ITU-T Y.2060]: A global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies.

NOTE 1 – Through the exploitation of identification, data capture, processing and communication capabilities, the IoT makes full use of things to offer services to all kinds of applications, whilst ensuring that security and privacy requirements are fulfilled.

NOTE 2 – From a broader perspective, the IoT can be perceived as a vision with technological and societal implications.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 e-health monitoring (EHM) service: A service which consists of observing and recording information based on a customer's physiological data, environmental data and other data, with the aim of monitoring the customer's state of health through the use of information and communication technologies.

3.2.2 e-health monitoring healthcare (EHMH) service: A class of EHM services providing the customer with health monitoring services for a 'healthy' state.

3.2.3 e-health monitoring rehabilitation (EHMR) service: A class of EHM services providing the customer with health monitoring services for a 'not fully healthy' or 'in recovery' state of health.

3.2.4 e-health monitoring treatment (EHMT) service: A class of EHM services providing the customer with health monitoring services for an 'illness' state of health.

3.2.5 EHM system: A set of hardware and software components which constitute as a whole the technical chain of e-health monitoring (EHM) service provisioning.

NOTE – EHM systems include EHM devices, gateways, networks, service support platforms and EHM applications.

3.2.6 EHM device: A device, as defined in [ITU-T Y.2060], which has sufficient qualification for e-health monitoring (EHM) service provisioning.

NOTE – Examples include EHM devices for EHMH (i.e., EHM devices which have sufficient qualification for EHMH), EHM devices for EHMT and EHM devices for EHMR.

3.2.7 EHM terminal: An e-health monitoring (EHM) device directly connected to the communication network.

3.2.8 EHM end point: An e-health monitoring (EHM) device connected to the communication network through gateway(s).

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

CT	Computed Tomography
ECG	Electrocardiogram
EHM	e-health Monitoring
EHMH	e-health Monitoring Healthcare
EHMR	e-health Monitoring Rehabilitation
EHMT	e-health Monitoring Treatment
EMR	Electronic Medical Record
EMSS	Emergency Medical Service System
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSM	Global System for Mobile communications
ICT	Information and Communication Technology
IP	Internet Protocol
IoT	Internet of Things

MRI	Magnetic Resonance Imaging
PDA	Personal Digital Assistant
PEMS	Pre-hospital Emergency Medical Service
QoS	Quality of Service
RFID	Radio Frequency Identification
SSAS	Service Support and Application Support
UMTS	Universal Mobile Telecommunications System
WAN	Wide Area Network
WSN	Wireless Sensor Network

5 Conventions

In this Recommendation:

The keywords "is required to" indicate a requirement which must be strictly followed and from which no deviation is permitted if conformance to this Recommendation is to be claimed.

The keywords "is recommended" indicate a requirement which is recommended but which is not absolutely required. Thus, this requirement need not be present to claim conformance.

The keywords "can optionally" and "may" indicate an optional requirement which is permissible, without implying any sense of being recommended. These terms are not intended to imply that the vendor's implementation must provide the option and the feature can be optionally enabled by the network operator/service provider. Rather, it means the vendor may optionally provide the feature and still claim conformance with the specification.

6 Classification of e-health monitoring services

This clause introduces a classification of e-health monitoring (EHM) services. The main purpose of this classification is to simplify the analysis of service network requirements and capability requirements for the support of EHM services.

For this classification of EHM services, two factors are considered: completeness and independency. Completeness means that the identified classes of EHM services cover all possible EHM services. Independency means that the identified classes of EHM services do not overlap with each other; in other words, each class has unique features specific to the EHM services of that class.

In this classification, human health is seen in one of four possible states: healthy, in recovery, not fully healthy, and illness. Each state has some service requirements which are unique to that state. These four states can be mapped into three EHM service classes which meet the two factors of completeness and independency: EHM healthcare, EHM rehabilitation and EHM treatment. Figure 6-1 shows these four human states of health and the corresponding EHM service classes.

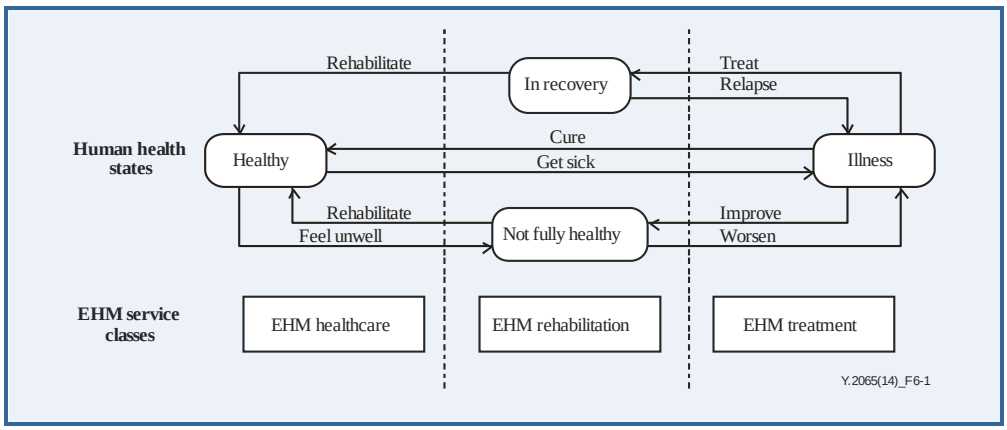


Figure 6-1 – Human states of health and corresponding EHM service classes

NOTE 1 – These EHM service classes have different characteristics, e.g., in terms of the number and type of target customers, target customers' mobility and the timing of service feedback to customers. Different service requirements are identified for each class.

NOTE 2 – EHM service classification does not sufficiently address health emergency situations. In such situations, there are a large number of requirements to be satisfied which are beyond the specific scope of e-health monitoring services.

6.1 EHM healthcare (EHMH) services

The target people of EHMH services are those in good health (healthy) but who pay close attention to their health status or those who still require some attention in that they are potentially at risk of getting diseases.

NOTE – EHMH services are usually provided by social and commercial organizations offering daily health-care services to people without on-site care.

6.2 EHM rehabilitation (EHMR) services

The target people of EHMR services include people with chronic diseases (not fully healthy state of health), and others who need on-site care (in recovery state of health).

NOTE – EHMR services may be provided by qualified organizations, such as rehabilitation centres, physical examination agencies, community medical stations and so on.

6.3 EHM treatment (EHMT) services

The target people of EHMT services include those who are hospitalized (illness state of health) and need medical services.

NOTE – EHMT services may be provided by qualified professional organizations, e.g., hospitals, medical emergency centres and so on.

7 Characteristics of e-health monitoring services

7.1 General characteristics

7.1.1 A class of services exploiting the capabilities of the IoT

EHM services exploit the identification, data capture, data processing and communication capabilities of the IoT [ITU-T Y.2060] to monitor customers' health, whilst maintaining the required privacy.

EHM services involve capabilities at all layers of the IoT reference model [ITU-T Y.2060], i.e., at the device layer, network layer, service support and application support layer and the application layer, whilst having some unique service requirements and capability requirements with respect to other classes of services which are exploiting the capabilities of the IoT.

7.1.2 Support of data sharing

The data generated by EHM services can be shared among different EHM services according to regulations, laws and other requirements.

7.1.3 Enhanced value via service support and application support layer capabilities

The service support and application support layer [ITU-T Y.2060] is key to the infrastructure of the IoT. Based on the capabilities of the service support and application support layer, the capabilities of the EHM services, e.g., data sharing and data communication, are enhanced in terms of efficiency, reliability and safety.

7.1.4 Enhanced value via network layer capabilities

In order to support customer access to EHM services remotely and locally, the network acts as a data transmission channel.

Based on the network layer capabilities, e.g., policy-based communication, network-based locating and network resource provisioning, the capabilities of EHM services are enhanced, e.g., in terms of network intelligence.

7.1.5 Combination of health-related technology and ICT

EHM services make use of both health monitoring-related technologies and information and communication technologies (ICTs); this implies that EHM services have to comply not only with ICT technical specifications but also with health-related specifications.

7.1.6 Multiple EHM devices serving one single user

Multiple EHM devices can serve one single user in a collaborative way.

Many EHM devices have a single function. For example, a blood pressure monitor measures blood pressure but it does not collect other physical health signals, such as ECG information, blood-oxygen levels, information on posture and so on. This implies that multiple EHM devices may be associated with one single user in a collaborative way to gather health information.

7.1.7 Users with different accessibility needs

Since EHM services address people with different accessibility needs, they have to be capable of meeting those needs accordingly.

7.1.8 Regulated services

Various EHM service aspects, including device, application and other aspects, are regulated by specific entities according to regulation and laws. Different types of EHM services may need to follow different regulation policies.

7.2 Specific characteristics of EHM services

7.2.1 Characteristics of EHM healthcare services

1) Service and network scalability

Compared to EHMT and EHMR services, the number of service providers and customers involved with EHMH services may be very large, as there are less professional and administrative constraints associated with these services. Consequently, service and network scalability is a key concern.

2) Wide service coverage

The EHMH users may access the services from a wide range of locations including home, school, office, train, vehicles and so on.

3) Data transmission with high reliability requirements and weak latency constraints

EHMH services need data transmission with high reliability but which also allows high latency.

– Data of EHMH services are transmitted without faults.

– EHMH services have weaker latency constraints than EHMT and EHMR services.

4) Unguaranteed support of clinical intervention

The EHMH services do not guarantee support of clinical intervention for customers.

7.2.2 Characteristics of EHM rehabilitation services

1) Access to data produced by EHMT and EHMH services

EHMR services may benefit from accessing data which have been produced by EHMH and EHMT services.

2) Restricted service coverage

EHMR services may be provided to users in qualified locations.

NOTE 1 – Inside qualified service buildings, users can usually obtain EHMR services with full capabilities. In other locations, users may access EHMR services with partial capabilities.

3) Support of clinical intervention

EHMR services provide support for clinical intervention to users.

4) Data transmission with high reliability requirements and medium latency constraints

EHMR services need data transmission with high reliability and which allows medium latency.

– Data of EHMR services are transmitted without faults.

– EHMR services have a stricter latency requirement than EHMH services, but a looser latency requirement than EHMT services.

7.2.3 Characteristics of EHM treatment services

1) Centralized management

EHMT services have usually centralized management inside organizations providing these services.

2) Medical imaging

Medical imaging devices used in EHMT services, such as CT, MRI, ultrasonic devices and so on, usually generate big data streams.

– Big data streams are generated among departments inside a hospital or among hospitals, as well as between hospital and emergency cars, and between disaster sites and hospital or emergency cars.

3) Data transmission with high reliability and low latency requirements

EHMT services need data transmission with high reliability and low latency requirements.

– EHMT services have the highest requirements of latency compared to EHMH and EHMR services.

8 Service requirements for support of e-health monitoring services

8.1 EHM roles

The roles participating in EHM services include EHM customer, EHM device provider, network provider, platform provider and EHM application provider.

These EHM roles can be mapped to the IoT business roles introduced in Appendix I of [ITU-T Y.2060], as shown in Figure 8-1.

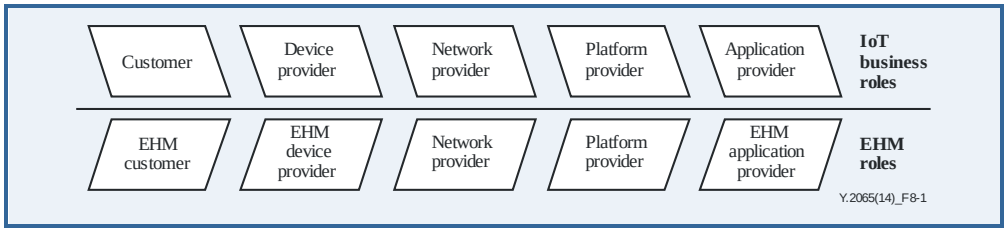


Figure 8-1 – Mapping between EHM roles and IoT business roles

The EHM customer is the end user of EHM services.

NOTE – For the purposes of this Recommendation, a healthy person, an in recovery or not fully healthy person, and a hospitalized person are the three actors which are considered as playing the role of an EHM customer.

The EHM device provider manages the EHM devices.

The network provider provides network access and connectivity for EHM devices, and provides network connections for the service support platform and for EHM applications.

The platform provider provides general service capabilities and EHM-dedicated service capabilities.

The EHM application provider provides EHM applications.

8.2 Service requirements of EHM customers

The following EHM customer requirements are essential for the support of EHM services.

8.2.1 Service requirements of a healthy person

A healthy person who is willing to use EHMH services to monitor his/her health condition is the target user of EHMH services.

1) A healthy person needs support to connect to the EHMH applications of EHM devices for EHMH in a convenient way; this includes meeting any accessibility needs. For usability, EHM services should be understandable in layman's terms.
2) A healthy person needs support for EHMH service access regardless of his/her location.

   NOTE 1 – Healthy people can use EHMH services continuously whether they work in a local city or travel or settle down in another city or country.
3) A healthy person needs support for information sharing.

   NOTE 2 – For example, the data generated by EHMR and EHMT services can be accessed by EHMH services as reference data.
4) A healthy person needs support in receiving one single bill regardless of the number of devices used.
5) A healthy person needs support for his/her location to be tracked.

   NOTE 3 – Based on the location information, EHMH services can send out messages for help if needed.
6) A healthy person needs support for fault recovery of used devices as soon as possible.
7) A healthy person needs support for personal information protection.

8.2.2 Service requirements of an in recovery or not fully healthy person

An in recovery or not fully healthy person is the target user of EHMR services.

1) An in recovery or not fully healthy person needs support to connect to the EHMR applications of EHM devices for EHMR in a convenient way; this includes meeting any accessibility needs. An in recovery or not fully healthy person needs support for EHMR connectivity.

2) An in recovery or not fully healthy person needs support for EHMR service access regardless of his/her location.

   NOTE 1 – An in recovery or not fully healthy person can use EHMR services continuously whether he/she works in a local city or travels or settles down in another city or country. He/she wishes to have the same service experience throughout their use of the services.

3) An in recovery or not fully healthy person needs support for information sharing.

   NOTE 2 – For example, the data generated by EHMH and EHMT services can be accessed by EHMR services as reference data.

4) An in recovery or not fully healthy person needs support in receiving one single bill regardless of the number of devices used.

5) An in recovery or not fully healthy person needs support for his/her location to be tracked.

   NOTE 3 – Based on the location information, an in recovery or not fully healthy person can receive first aid in an emergency situation.

6) An in recovery or not fully healthy person needs support for fault recovery of used devices as soon as possible.

7) An in recovery or not fully healthy person needs support for personal information protection.

8.2.3 Service requirements of a hospitalized person

A hospitalized person who is under treatment in medical facilities such as hospitals, medical emergency centres or ambulances, is the target user of EHMT services.

1) A hospitalized person needs support to connect to the EHMT applications of EHM devices for EHMT in a convenient way; this includes meeting any accessibility needs.

2) A hospitalized person needs support for obtaining reliable EHMT services.

3) A hospitalized person needs support for information sharing.

   NOTE 1 – For example, the data generated by EHMH and EHMR services can be accessed by EHMT services as reference data.

4) When a hospitalized person uses multiple EHM devices for EHMT at the same time, time synchronization among EHM devices is needed.

   NOTE 2 – Parameters gathered by multiple EHM devices for EHMT need to be synchronized to reflect the value of the different physiological parameters at the same time.

5) A hospitalized person needs support for his/her location to be tracked so that, e.g., he/she can get first aid in an emergency situation.

6) A hospitalized person needs support for fault recovery of used devices as soon as possible.

7) A hospitalized person needs support for personal information protection.

8) A hospitalized person needs support for the availability of EHM devices.

8.3 Service requirements of an EHM device provider

The following EHM device provider requirements are essential for the support of EHM services.

1) In order to reduce EHM device costs and to provide support for interoperability with service support platforms, EHM applications and other EHM devices, the EHM device provider needs support for EHM devices which reuse common purpose capabilities as much as possible.
2) When EHM device updates of software or firmware take place, the EHM device provider needs support for notifying the EHM application provider and EHM customer.
3) The EHM device provider needs support for EHM device reliability and security according to technical standards requirements.
4) The EHM device provider needs support for open interfaces to EHM device capabilities in order to enable EHM device capabilities' access by EHM applications, service support platforms, networks and other devices.
5) The EHM device provider needs support for the collection of fault information from devices, networks, service support platforms and application to give verdict on whether the root of an accident comes from the devices.
6) The EHM device provider needs support for acquiring the information related to device initialization and registration from the application provider, platform provider and network provider.
7) The EHM device provider needs support for the time calibration of EHM devices.

8.4 Service requirements of a network provider

8.4.1 Network provider essential requirements

The following network provider requirements are essential for the support of EHM services.

1) The network provider needs support for distinguishing which EHM service is in use (i.e., EHMH, EHMR and EHMT). This is for example, to guarantee the EHM service's QoS and EHM customer's QoE.

8.4.2 Network provider's essential but not EHM specific requirements

The following network provider requirements are essential for the support of EHM services but not specific to EHM services.

1) The network provider needs support for providing access to EHM applications as fast as possible upon service request.
2) The network provider needs support for obtaining the customer's EHM service-related information in order to allocate or configure for the EHM customer the appropriate network resources, such as IP address, network bandwidth, QoS policy, and so on.
3) The network provider needs support for flexible accounting for an EHM application provider and EHM customer.
4) The network provider needs support for the collection of fault information from devices, networks, service support platforms and applications to give verdict on whether the root of an accident comes from the network.
5) The network provider needs support for the remote update of an EHM customer's network subscription information residing in the EHM customer's device.

8.5 Service requirements of a platform provider

The following platform provider requirements are essential for the support of EHM services.

1) In addition to IoT common service capabilities, the platform provider needs to provide EHM dedicated service capabilities for EHM services.
2) The platform provider needs support for EHM service information sharing.
3) The platform provider needs support for data storage of EHM service information, e.g., to ensure EHM service information is not lost or inconsistent.
4) The platform provider needs support for the collection of fault information from devices, networks, service support platforms and applications to give verdict on whether the root of an accident comes from the service support platform.
5) The platform provider needs support for time synchronization for EHM devices, service support platforms and application servers.

8.6 Service requirements of an EHM application provider

8.6.1 EHM application provider's essential requirements

The following EHM application provider requirements are essential for the support of EHM services.

1) The EHM application provider needs support for EHM service information sharing.
2) The EHM application provider needs support for the collection of fault information from devices, networks, service support platforms and applications to give verdict on whether the root of an accident comes from the application.
3) The EHM application provider needs support for protecting the EHM customer's personal information.
4) The EHM application provider needs support for registration management of an EHM customer's devices.
5) The EHM application provider needs support for distinguishing the accuracy of the EHM data collected by the EHM devices.
6) The EHM application provider needs support for time synchronization of the EHM data provided to EHM applications by EHM devices.

8.6.2 EHM application provider's essential but not EHM specific requirements

The following EHM application provider requirements are essential for the support of EHM services but are not specific to EHM services.

1) The EHM application provider needs support for the upgrade of software/firmware hosted in EHM devices.
2) The EHM application provider needs support for flexible accounting from the network provider and/or platform provider.
3) The EHM application provider needs support for EHM service access which is independent of the EHM application's location, i.e., EHM applications need to be accessed by EHM customers continuously no matter where the EHM applications are located.
4) The EHM application provider needs support for network switching mechanisms in order to be able to change the network provider to which applications can subscribe.
5) The EHM application provider needs support for getting the location information of EHM customers.

9 Capability requirements for support of e-health monitoring services

9.1 Introduction to the EHM capabilities

The following subclauses describe the EHM capability requirements according to the IoT reference model [ITU-T Y.2060].

The EHM reference model, shown in Figure 9-1, exhibits two types of capabilities, EHM essential IoT capabilities derived from the EHM service requirements and EHM not essential IoT capabilities. They are located at the various layers of the IoT reference model [ITU-T Y.2060].

NOTE 1 – The EHM reference model excludes on purpose the IoT capabilities which are not related to the specific support of EHM services. Consequently, this clause does not cover other IoT common capabilities which are still necessary to support EHM services.

NOTE 2 – The distinction between EHM essential IoT capabilities and EHM not essential IoT capabilities concerning the capabilities described in each of the following subclauses is beyond the scope of this Recommendation.

In Figure 9-1, rounded rectangles represent layers (i.e., application layer, service support and application support (SSAS) layer, network layer, device layer) according to the IoT reference model; rectangles represent capabilities provided by the various layers of the IoT reference model, as well as security and management capabilities.

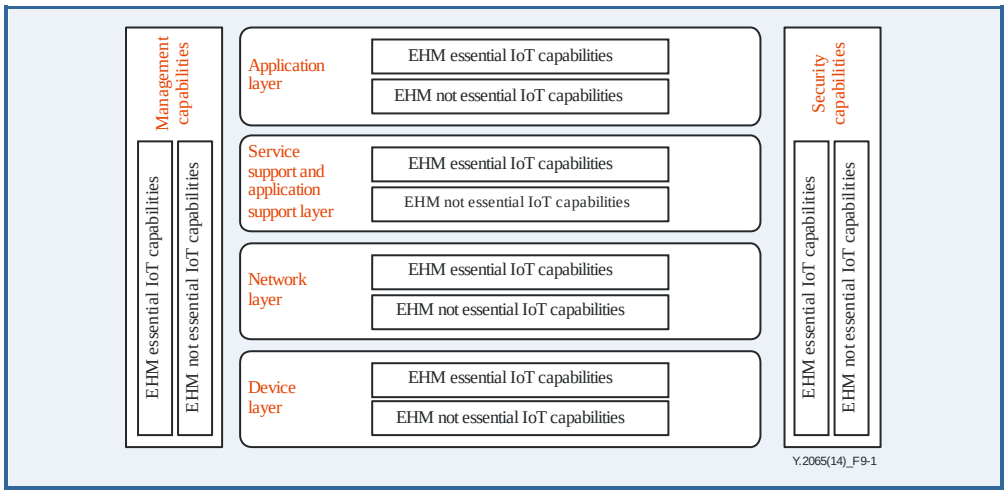


Figure 9-1 – EHM reference model

9.2 Capabilities of the application layer

9.2.1 Information sharing

Information sharing is one of the basic capability requirements for EHM. According to the service requirements 8.6.1(1), 8.2.1(3), 8.2.2(3), 8.2.3(3), the application layer is required to provide standard interfaces and policy-based mechanisms to enable the sharing of EHM information among different EHM services. Examples of policy rules for policy-based mechanisms include, but are not limited to, government rules, privacy rules, commercial agreements between application providers and so on.

9.2.2 Accounting related information provision

According to service requirements 8.6.2(2), 8.2.1(4), 8.2.2(4), the application layer is recommended to report accounting related information to the SSAS layer. The accounting related information includes, but is not limited to, application type (EHMH, EHMR and EHMT), the number of times that the application is used, the time during which the application is used, etc.

9.2.3 QoS information provision

According to service requirements 8.4.2(2), 8.4.1(1), 8.2.3(2), the QoS information for EHM services is required to be configured by the application layer and provided to the other layers, so that the other layers can ensure the QoS of EHM services according to the QoS information provided by the application layer.

The following QoS related parameters are recommended to be indicated in the provided QoS information:

1) Required response time

   As different types of EHM services need to be dealt with in different time periods, response time is an important criterion in reckoning the requirements of an EHM service.

2) Allowed dispose time

   Dispose time is the time period from the moment when data arrive at the server to the moment when doctors or application providers execute an appropriate reaction. Generally speaking, dispose time includes the time to analyse data, the time to store data in the storage area, the time to send an alarm to doctors when unusual results are deduced, and so on. Dispose time as part of the response time is very important in reckoning the EHM application capabilities.

3) Instantaneity level

   Instantaneity level indicates the level of priority of the EHM application when the EHM application related data are transmitted, processed and queued.

4) Minimum transmission rate

   In some EHM scenarios (i.e., in an emergency car or disaster rescue), the voice, video or dynamic monitoring data need be transmitted to the remote server for diagnosis and treatment in real time. To ensure real-time data transmission, a minimum transmission rate is required to be indicated.

5) Maximum transmission time

   Maximum transmission time as part of the response time is used to limit the transmission time. For some non-real time EHM applications (i.e., routine physical examination), although there is no minimum transmission rate requirement, there is an allowed maximum transmission time restriction.

9.3 Capabilities of the SSAS layer

9.3.1 Service accounting and charging

Service accounting is responsible for gathering data about the usage of EHM services and for charging the service usage to the user. Different policies may be considered for service accounting and charging, e.g., the number of times the service is used, the amount of time the service is used or the volume of service data used. According to service requirements 8.6.2(2), 8.2.1(4), 8.2.2(4), the service accounting and charging capability supported in the SSAS layer has the following requirements:

1) It is required to provide service accounting and charging to EHM service users.
2) It is recommended to provide service accounting and charging according to the quality of service of EHM services.
3) It is recommended to provide service accounting and charging also in support to roaming scenarios among networks owned by different network providers.
4) It is recommended to provide service accounting and charging according to the frequency of access to EHM services.

5) As a user may use several EHM devices at the same time, it is recommended to support unified service charging per user, not per end point.

9.3.2 Message conversion

According to service requirement 8.5(2), the SSAS layer is required to provide message conversion for EHM applications and EHM devices. Structured information sharing among EHM applications is realized via messages which are composed of predefined syntax and semantics. The messages transmitted between EHM applications and EHM devices are often not uniform. EHM applications and EHM devices may use messages with different syntax or semantics, which are possibly not compatible with each other. So the SSAS layer is required to provide message conversion for EHM applications and EHM devices.

9.3.3 Data storage

According to service requirement 8.5(3), the SSAS layer is required to provide data storage for EHM applications and EHM devices.

NOTE – The exponential growth in electronically stored EHM data and the simultaneous storage of huge amounts of data are putting pressure on this capability. Large data centres are of increased relevance for support of this capability.

The data storage capability requirements include the following:

1) Standard format

   The data stored in the SSAS layer are recommended to be stored in standard format so that the information can be easily exchanged among different EHM applications.

2) Object orientation

   The data storage in the SSAS layer is recommended to adopt the object-oriented access technique for layer separation and independence, so that the information of each EHM customer and each EHM device can be modelled as objects and mapped into the storage area.

3) Time stamping

   The EHM application data stored in the SSAS layer are required to be marked with collection time, since health conditions can vary over time. Using time stamping, the EHM applications can obtain useful information according to the health history.

9.3.4 Time synchronization

According to service requirement 8.2.3(4), the time synchronization capability is required to be supported in the SSAS layer, which includes:

1) Time retrieval

   The SSAS layer is required to retrieve time parameters from authoritative time servers or via other ways according to the application requirements.

2) Time announcement

   The SSAS layer is required to publish the time parameters according to the application requests of EHM applications and devices. It is recommended that the SSAS layer publishes time parameters periodically for the time calibration of EHM devices and applications.

9.3.5 Location provisioning

According to service requirements 8.2.1(5), 8.2.2(5), 8.2.3(5), 8.6.2(5), the location provisioning capability is required to be supported in the SSAS layer to provide to EHM applications the position of EHM customers, according to regulations and laws.

The location provision capability supported in the SSAS layer includes:

1) Location information collection

   The SSAS layer is required to collect the location information from the network layer or device layer according to the collection strategy such as event triggered collection or periodic collection.

2) Location information tracking

   The SSAS layer is recommended to track the position of EHM customers via frequent collection of the location information of EHM customers.

3) Location information reporting

   The SSAS layer is required to report the location information required by the application layer in standard format.

9.4 Capabilities of the network layer

9.4.1 Policy-based communication

According to service requirement 8.4.1(1), the network layer is required to provide policy-based communication for EHM applications and EHM devices. Policy is a set of rules whose variables include, but are not limited to, time, bandwidth, data throughput, network type, traffic priority, and so on. By means of policy-based communication, EHM applications and EHM devices can obtain the desired QoS.

The policy-based communication capability provided by the network layer is required to set the network policy to support the QoS of EHM services according to their QoS requirements.

9.4.2 Network-based locating

According to service requirements 8.2.1(5), 8.2.2(5), 8.2.3(5), the network layer is recommended to provide the location related information from the network layer (e.g., IP address, access point location, and so on) for locating the position of EHM devices.

Event triggered location information notification is recommended to be supported. For example, when the EHM customer has moved out of the preconfigured network area, a network location information notification may be triggered by the event.

9.4.3 Network resource provision

According to service requirements 8.4.2(1), 8.4.2(2), 8.2.1(1), 8.2.2(1), 8.2.3(1), the network layer is required to provide the network resource provision capability for EHM applications and EHM devices. Example of network resources include, but are not limited to, a network address for an EHM device, network bandwidth for an EHM application, and so on.

Depending on the specific deployment of EHM applications and EHM devices, EHM applications and EHM devices may automatically use these provided network resources and configure themselves to connect to the network directly. In this way, EHM customers can use the EHM services directly, without the need to configure the EHM devices.

9.5 Capabilities of the device layer

9.5.1 Device identification

According to service requirement 8.4.1(1), the device layer is required to support device profiles to identify the intended use of EHM devices, such as the supporting of EHMH and/or EHMR and/or EHMT services.

NOTE – The EHM devices are different from ordinary customer electronic devices. In EHM services, the EHM devices collect physical signals directly and/or indirectly from the human body. The EHM devices have high demands for security, safety and reliability.

9.5.2 Gateway

According to service requirement 8.3(1), the device layer is required to provide gateway capabilities for EHM devices and EHM applications. A gateway can serve multiple EHM end points and it provides gateway capabilities by acting on behalf of the EHM end points (e.g., the gateway can provide data processing when the connected EHM end points cannot process the raw data by themselves).

9.5.3 Data sensing and processing

According to service requirement 8.6.1(5), the device layer is required to support the data sensing and processing capability for obtaining EHM data.

The data sensing and processing capability required to be supported in the device layer includes:

1) Data sensing

   Data sensing is used to obtain the raw EHM data and is required to respect corresponding regulations and laws. It is recommended to support the sensing of multiple EHM parameters in a single EHM device.

2) Data processing

   Data processing is used to process raw EHM data, such as filtering, aggregating, computing, etc., in order to obtain the desired EHM data.

NOTE – EHM devices can utilize this capability to derive the desired EHM data according to different policies, including at fixed time intervals, upon application request and so on.

9.5.4 Data collection time provision

According to service requirements 8.2.3(4), 8.3(7), the data collection time provision capability is recommended to be supported in the device layer, so that the collected EHM data can be marked with the collection time.

The collection time of EHM data is recommended to be known with precision by the EHM application server. It is required to mark the EHM data with the collection time in EHM devices or gateways instead of the EHM application server, since the network transmission time and dispose time affect the precision of the collection time.

The data collection time provision capability recommended to be supported in the device layer includes:

1) Time calibration

   The time calibration capability is used to obtain the time parameters from the SSAS layer and calibrate the built-in time clock of EHM devices.

2) Time provision

   The time provision capability is used to provide the calibrated collection time along with the collected EHM data for time stamping.

9.5.5 Device based locating

According to service requirements 8.2.1(5), 8.2.2(5), 8.2.3(5), 8.6.2(5), the locating capability is recommended to be supported in the device layer to get the position of EHM devices.

The EHM devices or gateways can utilize different techniques (e.g., GPS, gyroscope and motion state sensor) to implement the locating capability.

Different levels of location accuracy are allowed according to the application requirements. It is recommended to indicate the location accuracy when the location information is sent from the device layer to other layers.

9.5.6 Device redundancy

According to service requirements 8.2.3(8), the device redundancy capability is recommended to be supported in the device layer to guarantee increased reliability and availability for EHMT services.

9.6 Management capabilities

9.6.1 General

According to service requirements 8.3(5), 8.4.2(4), 8.5(4), 8.6.1(2), , 8.2.1(6), 8.2.2(6), 8.2.3(6), 8.2.3(1), 8.6.2(1), 8.6.1(4), the EHM system, composed of entities in the application layer, SSAS layer, network layer and device layer, is required to support the following management capabilities:

– fault management capability;
– configuration management capability;
– initialization and registration management capability.

9.6.2 Fault management

According to service requirements 8.3(5), 8.4.2(4), 8.5(4), 8.6.1(2), 8.2.1(6), 8.2.2(6), 8.2.3(6), the EHM system is required to recognize, isolate, correct and log faults that occur in the EHM system.

– It is required to enable service logging reports to the various parties involved in an EHM service.
– It is required to enable the collection and storage of fault management data.

9.6.3 Configuration management

According to service requirements 8.2.3(1), 8.6.2(1), the EHM system is required to provide the configuration management capability for EHM applications and EHM devices. Examples of provisioning actions include hardware and programming (configurations) changes, including the addition of new devices and programs, modification of the existing EHM system and removal of obsolete EHM systems and programs.

The different layers of the EHM system are required to support different configuration capability requirements.

1) The application layer and the SSAS layer are required to support the following capabilities:

- connection configuration management;
- software and firmware configuration management;
- EHM application configuration management, such as lifecycle management;
- service configuration management, e.g., service configuration, service profile setting and so on.

2) The device layer is required to support the following capabilities:

- fault management and connection management;
- software and firmware configuration management;
- proxy management, which includes but is not limited to the following capabilities:
  - acting as a management client to perform the management functionalities for the EHM gateway itself;

- acting as a management proxy for EHM devices:
  - accepting and processing management requests, targeted at one or multiple EHM devices, from the application and SSAS layers;
  - accepting and processing management requests from one or multiple EHM devices and/or further interacting with the application and SSAS layers on behalf of the EHM devices (e.g., in the case of fault detection and reporting);
  - triggering the application and SSAS layers to start performing device management tasks (e.g., firmware/software update, fault diagnostics) with one or multiple devices;
  - scheduling of remote management tasks for sleeping devices.

9.6.4 Initialization and registration management

According to service requirements 8.2.3(1), 8.6.1(4), the initialization and registration management capability is required to be supported in the EHM system. When EHM devices access the EHM system for the first time, the initialization and registration management capability can help the EHM devices to complete the device initialization set-up, and write the device and user information into the related database.

The initialization and registration management capability needs the following support at the different layers:

1) Application layer and SSAS layer

   The application layer and the SSAS layer are required to be able to write the device or user information into the related application layer or SSAS layer database and to provide to the EHM devices the required configuration information for the initialization set-up of the EHM devices.

2) Network layer

   The network layer is required to provide the network resources for EHM devices to access the network, e.g., network address allocation.

3) Device layer

   The device layer is required to support the capability of initialization set-up. The EHM device can complete the initialization set-up by itself or with the help of the EHM gateway according to the provided configuration information from the application layer or the SSAS layer.

9.7 Security capabilities

According to service requirements 8.2.1(7), 8.2.2(7), 8.2.3(7), 8.3(3), 8.6.1(3), the EHM system is required to support the following security capabilities:

1) Authentication and authorization

   The EHM system is required to support authentication and authorization mechanisms.

2) Secure communications

   According to service requirements 8.2.1(2), 8.2.2(2), the information carried by the EHM services may be delivered across different administrative domains (e.g., countries, operators). The EHM system supports secure communications between different domains. The information exchanged between different domains must be protected from random errors, as well as snooping or hacking attacks.

3) Confidentiality

Whenever information is exchanged, stored or processed, the confidentiality of the data must be enforced and safeguarded by the EHM system. All exchanges of data between e-health partners, for example EHM device provider, EHM application provider, network provider and platform provider, must be performed in a way that prohibits any unwanted disclosure of data, e.g., to third parties.

4) Integrity

The integrity of the transmitted information must be guaranteed: transmitted data from the sender should be received without any alteration. It must be identified that the transmitted data have not been damaged, reduced or altered. Any loss of integrity of the transmitted data must be recognizable by the recipient.

5) Access control

It should be ensured that only authorized persons and EHM system entities (e.g., applications, devices) are able to access protected data.

6) Audit trail

Any access or attempt to access medical data through EHM services must be fully transparent, traceable and reproducible.

7) Data storage security

It is recommended to support data storage security strategies including, but not limited to, data backup, anti-hacker data protection, uninterruptible power of data storage, data integrity validation and data recovery. In addition, data access control is required to be supported for privacy.

Appendix I

e-health monitoring service scenarios

(This appendix does not form an integral part of this Recommendation.)

I.1 Individual/family (indoor and outdoor)

The EHM services described in this appendix are examples of EHMH services.

In the individual/family scenarios, by means of communication and diagnostic tools, EHM customers can sample their own physiological parameters at anytime and anywhere, and send them to health-care institutions in a timely and accurate way. The staff of health-care institutions can provide guidance to EHM customers based on both the past and current data received regarding their conditions.

The individual/family scenarios include both indoor and outdoor ones. In indoor scenarios, the sampled physiological parameters can be transmitted in both wired and wireless ways, while in the outdoor scenarios sampled physiological parameters are generally transmitted in a wireless way.

In individual/family scenarios, e-health monitoring devices should have the basic medical monitoring capability, as well as the features of miniaturization, portability, easy operation and the capability of short-distance communication.

An example of an indoor EHM service scenario is shown below in Figure I.1:

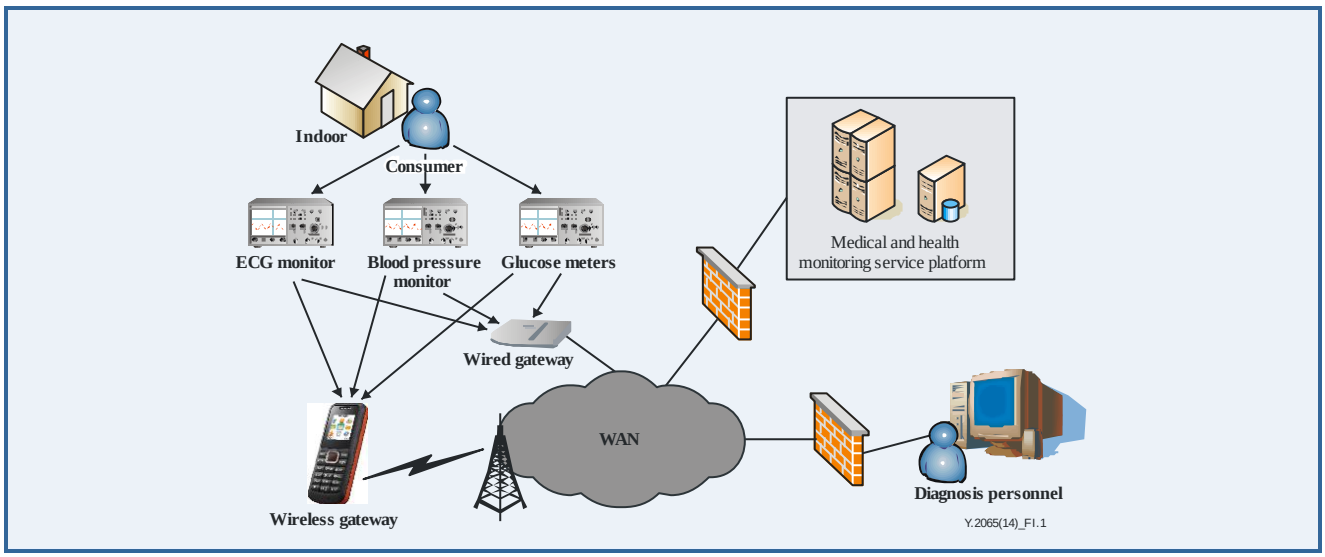


Figure I.1 – Indoor scenario

An example of an outdoor service scenario is shown below in Figure I.2:

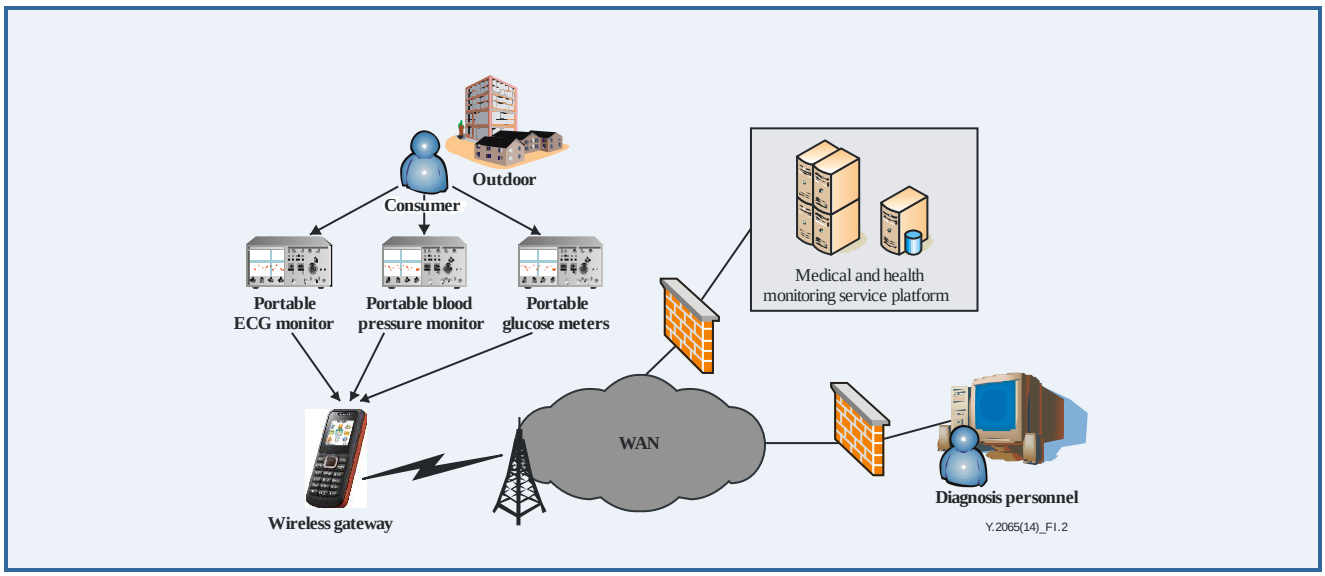


Figure I.2 – Outdoor scenario

Customers can detect their sampled data through portable ECG monitors, portable blood pressure monitors, portable glucose meters and other portable equipment, then they can preprocess the monitored data and forward the data to the medical and health monitoring service platform via a wired gateway or a wireless gateway (as wired network connectivity is not convenient outdoors, one must use a wireless gateway, e.g., a smart phone).

The diagnosis personnel can access in real time the monitored data through the service platform, determine the customer's conditions according to their basic information and past medical history, and give health guidance to them.

I.2 Physical examination

A user is assumed to have physical examinations or disease check-ups regularly or to have had them in the past. The physical examination includes routine checks such as height, weight, blood pressure, eyesight, chest X-ray, etc., and where required, specific disease check-ups. The user chooses to send the monitored data via a wired or wireless gateway to the ubiquitous e-health monitoring server in health-care institutions or have the data written into the user's e-health records (including a user's basic information and past health records, which are stored in the system). Then the medical staff analyse and determine the user's health conditions according to both current and past data, and gives health guidance to the user.

In the physical examination scenario, the e-health monitoring devices should have the basic medical monitoring capability, as well as the communication capability to transmit the monitored data and receive data from the e-health application servers.

An example of a physical examination scenario is shown below in Figure I.3.

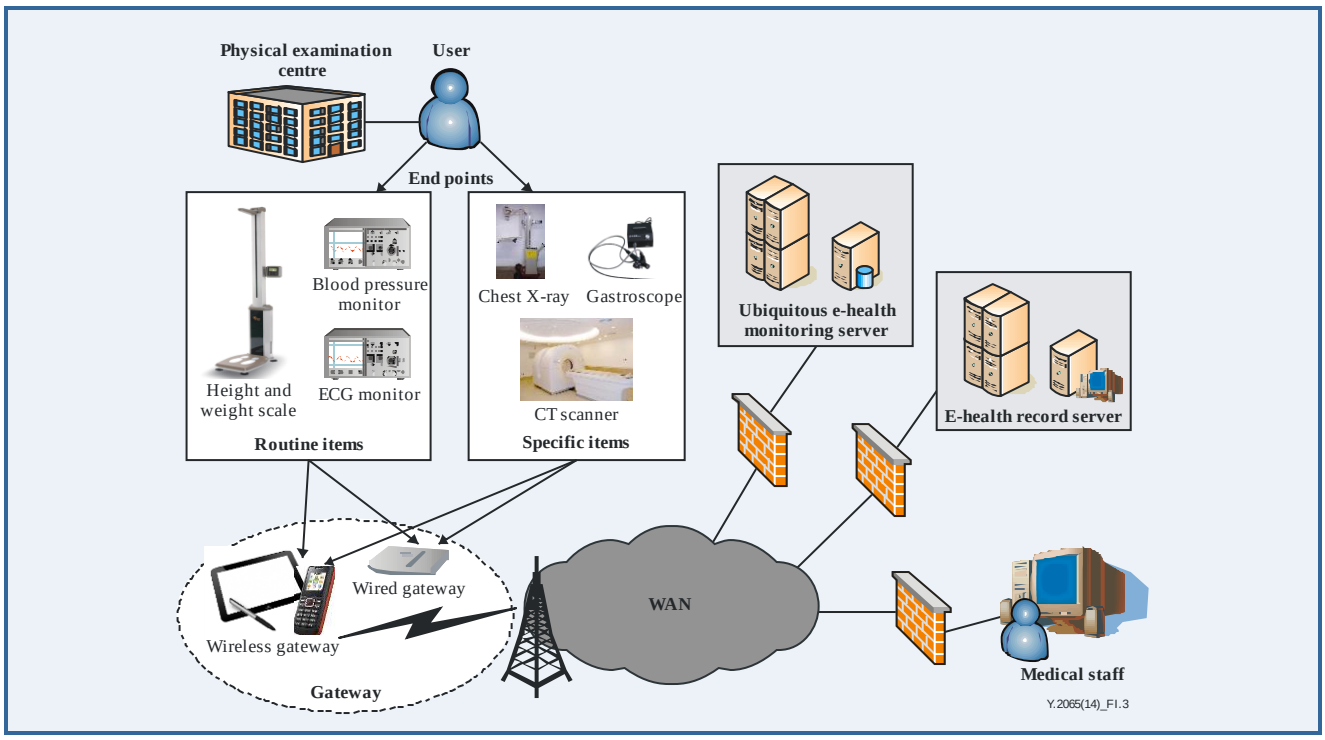


Figure I.3 – Physical examination scenario

The features of a physical examination service include, but are not limited to, the following concerning:

– Existing resources:
  - various kinds of advanced medical devices embedded with sensors, such as height and weight scales, ECG monitors, blood pressure monitors, etc.
  - advanced communication and information processing technologies, including the technologies of IoT, wireless sensor networks, context awareness, etc.
  - e-health monitoring service platform and e-health record applications.

– Required capabilities:
  - Device: The devices used in a physical examination service for physiological parameter collection should have high accuracy and stability to ensure reliable measurements.
  - Gateway: The gateway is necessary: a) in special areas of the physical examination centre (gateway collecting different data and transmitting them); b) for possible service extension to home environments (this scenario is not described here). The gateway should convert the information received from each device into the data (and associated formats) transmitted over the WAN. A high signalling processing capability is required with a larger number of subordinate end points.
  - Network: A private network may be applied to ensure secure and reliable connectivity between the gateway and the e-health monitoring and e-health record servers. For the possible service extension to home environments, a public network is used. However, special attention should be paid to data and network security in this case.

– Security requirements:
  - Authentication and authorization: the e-health monitoring server and the e-health record server provide authentication and authorization for gateways and devices. The authentication and authorization of each end point can be done by the gateway it is subordinated to, or by the e-health monitoring server and the e-health record server.

- Data storage: devices should be able to store the acquired data for a certain period of time (for example, 24 hours, 7 days, etc.). The gateway should at least be able to store the routing and topology related information of the subordinate end points, and the physiological parameters. When a gateway is the authentication point of end points, the gateway should be also able to store the authentication and authorization information of the subordinate end points.
- Electrical safety: The devices should be able to resist electromagnetic interference and meet the limitation requirements of electromagnetic interference. Radiation levels should meet certain standards.

I.3 Disaster rescue

In disaster rescue scenarios, by means of advanced communication and diagnostic tools, the sampled physiological parameters of injured people can be obtained at anytime, anywhere and in a timely and accurate way by the medical staff located both inside and outside of the disaster area. Then the medical staff can determine the conditions of those injured according to sampled physiological parameters, and they can give first-aid guidance to them. The location information of those injured is acquired and recorded by means of the wireless sensor network, so that those who are injured can be easily found by the medical staff.

The disaster rescue scenarios include those inside the disaster area and those outside of the disaster area. For those inside the disaster area, the sampled physiological parameters are transmitted in a wireless way, while for those outside of the disaster area, the sampled physiological parameters can be transmitted in wired or wireless ways.

In disaster rescue scenarios, the e-health monitoring devices should have the basic medical monitoring capability, as well as the capabilities of short-distance and long-distance communication in order to acquire and transmit data to the wireless gateway and remote monitoring centre.

An example of a disaster rescue scenario is shown below in Figure I.4.

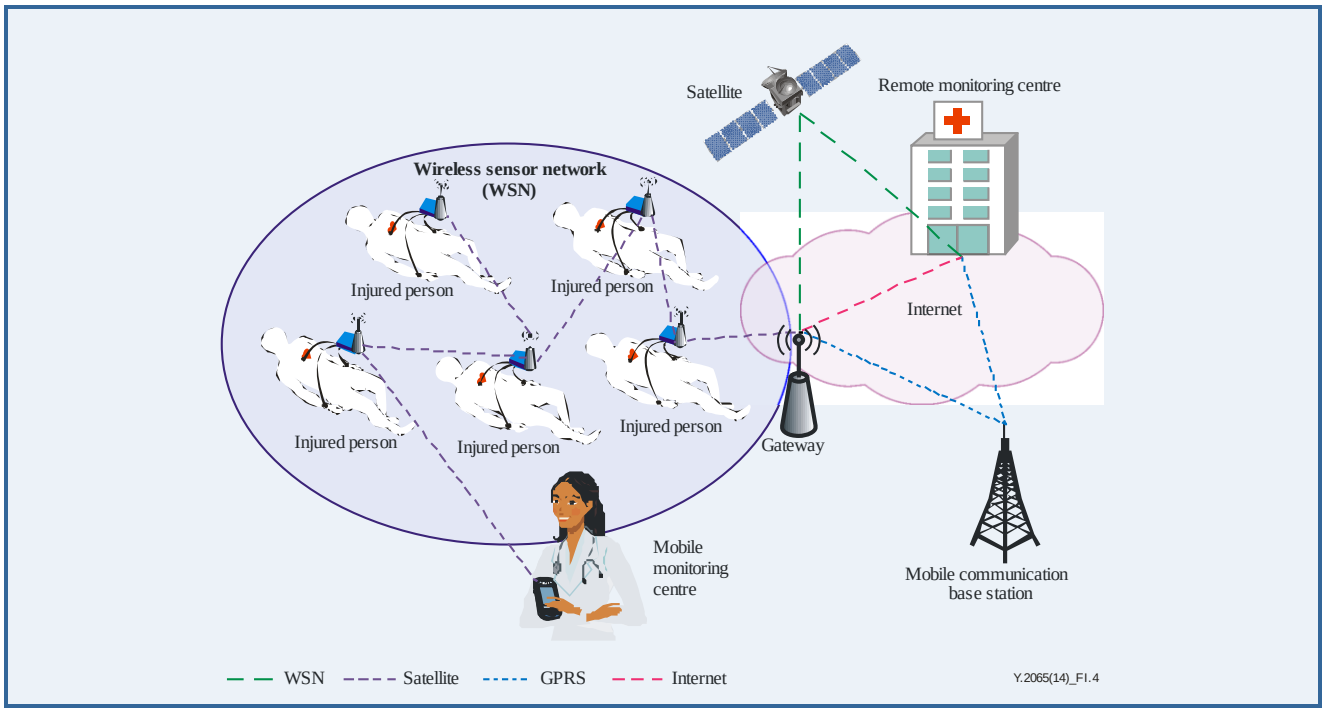


Figure I.4 – Disaster rescue scenario

The network of e-health monitoring services in disaster rescue scenarios can be divided into two parts: the wireless sensor network and the long-distance network.

Where there is a complicated geographical environment in the disaster area, wireless networks can be established more easily and flexibly than wired networks. So networks based on wireless technologies such as wireless sensor networks are usually established in the disaster area.

The long-distance network is built outside the disaster area: through it the sampled physiological parameters can be transmitted in both wired and wireless ways, e.g., via Internet, GPRS or satellite.

In the wireless sensor network, each injured person wears a wireless end point. The end point includes two parts: portable multi-parameter sensors and a wireless transceiver. The injured person's physiological parameters, such as ECG, blood pressure, heart rate and temperature, are measured in a timely and accurate way by portable multi-parameter sensors without medical staff on site. Then, the physiological parameters from all the injured people are transmitted to the mobile monitoring centre and wireless gateway by wireless transceivers and a wireless network. At the same time, the location information of those injured will be acquired and recorded by the wireless sensor network, so that they can be easily found by the medical staff.

The mobile monitoring centre can be a panel computer or personal digital assistant (PDA) carried by the medical staff in the disaster area. The physiological parameters collected from the injured people are shown on the computer/PDA so that the medical staff can supervise them in a timely way when moving within the disaster area.

The wireless gateway is the gateway of the wireless sensor network. It has three main functions: configuring the wireless sensor network, gathering physiological parameters of all the injured people from the wireless sensor network and communicating with the remote monitoring centre via the long-distance network (e.g., transmitting the physiological parameters to the remote monitoring centre and transmitting instructions from the remote monitoring centre to the wireless sensor network).

The remote monitoring centre can be a hospital with rich medical resources. The doctors can monitor in real time the critically injured people's conditions according to the received physiological parameters and they can comprehensively determine their illness. Then the doctors send first-aid guidance to the medical staff in the disaster area via the long-distance network and the wireless sensor network, so that those who are critically injured can get a timely and accurate diagnosis, as well as appropriate emergency treatment.

The features of a disaster rescue service include, but are not limited to, the following concerning:

– Providers of a disaster rescue service

In a disaster area, medical staff include doctors and nurses; the nurses take care of those who are slightly injured, while both nurses and doctors take care of the seriously injured people. Both nurses and doctors are required to have basic medical treatment training before undertaking the disaster rescue tasks.

Outside of the disaster area, some medical staff are located in the remote monitoring centre. They monitor the seriously injured people and are required to have a high level of professional medical treatment experience before undertaking these tasks.

– Users of disaster rescue service

The identification of the injured person is realized via a wristlet which has an RFID (radio frequency identification) module embedded in it and which is worn by each injured person. The wristlet is the one and only way of identifying an injured person during treatment. The physiological parameters of the injured people are bound with their own ID number, and all this information is sent to the medical staff, including those inside and those outside of the disaster area.

The activation of useful related information concerning the injured person: the medical staff record the injured person's information, such as name, age, gender, family relationship, etc. in the equipment constituting the mobile monitoring centre. With this information, the medical staff can activate the useful related information (such as drug history, family history of disease) concerning the injured person. The physiological parameters collected from the injured person are also shown on the mobile monitoring centre computer so that the medical staff can supervise the injured person in a timely way when moving within the disaster area.

– Unique features of service

The wireless end points: the wireless end points are portable medical multi-parameter devices. The sensors of medical parameters, such as ECG, blood pressure, heart rate and temperature, are integrated into the wireless end points to reduce the number of required devices and simplify the complexity of the wireless sensor network. At the same time, the wireless end points replace the medical staff's manual way of collecting the injured person's physiological parameters.

The network inside the disaster area: considering the complicated geographical environment of the disaster area, a wireless sensor network is built inside the disaster area. In the wireless sensor network, each injured person wears a wireless end point. The injured people's physiological parameters are collected by the wireless end point and transmitted via the wireless sensor network to the mobile monitoring centre and then to the remote monitoring centre.

The location of the injured people: in the disaster area, the location of those injured varies. In some disaster cases, such as earthquakes or floods, the global system for mobile communications/ universal mobile telecommunications system (GSM/UMTS) network is not available; in these cases, the injured person can be located by the wireless sensor network, so that he/she can be found by the medical staff. On the other hand, if the GSM/UMTS network and the injured person's mobile phone are available, the injured person can use the mobile phone to report his/her location.

Data storage: end points should be able to store the physiological parameters of the injured people. The gateway should be able to store the locations, routings and topologies of the end points in the wireless sensor network, and store the data when needed. The remote monitoring centre should store the acquired data if needed and also for future treatment.

– Common features of service

The gateway: the gateway has the three capabilities of configuring the wireless sensor network, gathering the physiological parameters from the wireless end points and communicating with the remote monitoring centre via the long-distance network. A high signalling processing capability of the gateway is required to ensure both wireless sensor network and long-distance network reliability.

The network outside the disaster area: the long-distance network built outside the disaster area and through which data can be transmitted in both wired and wireless ways, such as Internet, GPRS and satellite, ensures data are received by the remote monitoring centre.

– Security requirements

Electrical safety: the wireless end points should be able to resist electromagnetic interference and meet the limitation requirements of electromagnetic interference. Radiation levels should meet the related standards.

I.4 Pre-hospital emergency medical service

I.4.1 Overview of pre-hospital emergency medical service

The pre-hospital emergency medical service (PEMS) which is usually offered outside of the hospitals, can be defined as an emergency medical treatment for patients injured by accidents or life-threatening diseases, and who are treated during transportation from an on-site location to the hospital; it can also reduce the time and costs of patient transportation significantly. The PEMS system is an important component of the emergency medical service system (EMSS), which is a precondition for successful rescue, and plays a significant role in modern life.

The summary of PEMS operational steps (as shown in Figure I.5) is as follows:

Step 1: An emergency call is made from the patient's side to the receptionist at the PEMS platform.

Step 2: Information on patient location computed by GPS navigation system is sent to the hospital tele-management, which is responsible for the initial evaluation of the patient, triage decisions and pre-transfer arrangements.

Step 3: Urgency is initially evaluated according to the information provided by the patient's call. Based on the evaluation result, a triage decision is being made and then ambulance assignment is allocated by the hospital tele-management.

Step 4: On-site rescue where supervision and consultation for primary care treatments is not available i.e. there is no physician on site.

Step 5: History, physical examination findings and available test data exchange takes place between the ambulance and hospital. Based on this information, which hospital the patient will be taken to and what medical resources (e.g., physician, surgical instruments) should be prepared by the hospital for the patient are arranged by the pre-transfer management.

Step 6: The patient is taken to the hospital by ambulance.

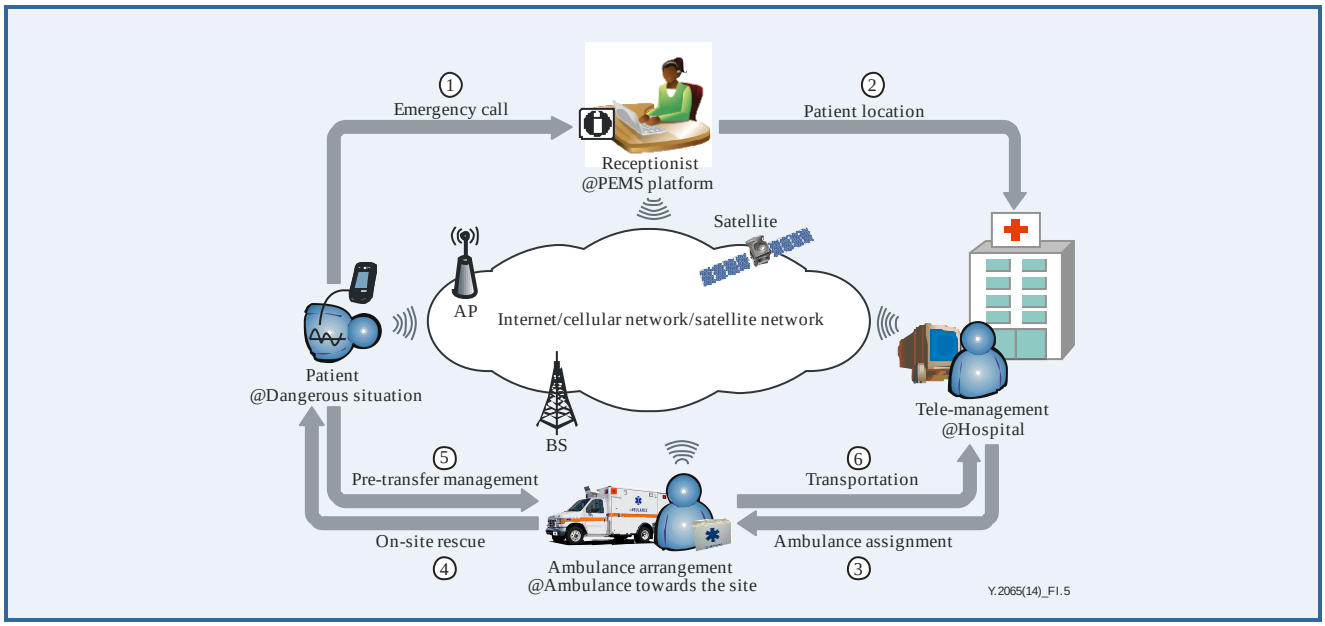


Figure I.5 – Pre-hospital emergency medical service operational flow

From a functional perspective, the PEMS system is made up of three main parts: a navigation system, physiological parameter monitoring system, and a remote medical treatment-assistant system (as shown below in Figure I.6).

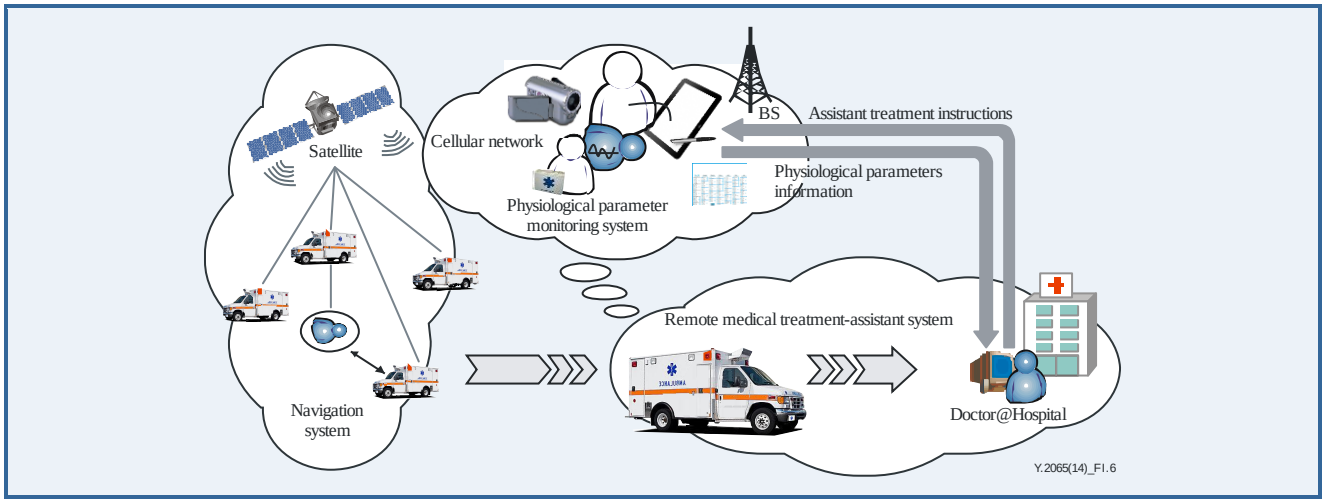


Figure I.6 – Main parts of the PEMS system

1) Ambulances install a navigation system with a positioning system, e.g., GPS, and wireless communication network capabilities, e.g., GPRS. By means of the GPS satellite positioning system, the emergency medical service centre can locate the patient and any available ambulances; the closest ambulance can be quickly sent. At the same time, the navigation system can provide the ambulance team with the most effective route to the hospital.

2) The physiological parameter monitoring system includes medical terminals and a mobile network and it provides the emergency medical service doctor with the real-time physiological parameters of remote patients, such as ECG, heart rate, oxygen saturation, blood pressure, respiratory rate, etc. Despite the unstable environment of a moving ambulance, the physiological parameters must be transmitted by a mobile network in a guaranteed manner so that the doctor can collect high quality physiological parameters. Also, the medical terminals on the ambulance must resist the fast fading when physiological parameters are transmitted to the hospitals via mobile networks.

3) A remote medical treatment-assistant system makes it possible for patients in an ambulance who require specialist medical care to have face-to-face consultations with specialists that are situated in the hospital or at another distant medical institution. In other words, it enables the emergency medical doctor to send medical data (including sounds, images and video), captured using medical peripherals, to a doctor in a hospital for generating patient diagnostics.

I.4.2 Special requirements for a pre-hospital emergency medical service

Pre-hospital emergency medical treatment is different from hospital treatment. As well as the fight against time, the emergency vehicle is moving at high speed. Thus, the following special requirements for PEMS should be seriously considered:

1) Accuracy

The real-time medical data of patients, such as ECG, heart rate, oxygen saturation, blood pressure, respiratory rate, etc., are the basis for emergency medical treatment which requires accuracy in data collection. The physiological parameter monitoring system should have a real-time data processing capability, including real-time dynamic signal filtering, fast detection and recognition for medical characteristic waveforms, self-learning and adaptive algorithms.

2) Mobility

Since the ambulance used in emergency medical treatment is moving at high speed and the pre-hospital emergency medical service centre communicates with it in a special fast fading channel, the mobile network needs to ensure high reliability of the transmissions. For reliable transmissions, mobile network switching and routing technologies should be adopted.

3) High QoS

It is essential in a critical medical environment that the PEMS performs with high precision; otherwise, the outcome could be fatal for patients. For this to be possible, it is necessary that the physiological parameters reach the end location with a high degree of reliability and predictability. PEMS systems are said to have stringent real-time QoS constraints, which if not respected can lead to disaster; for example, the unbounded delay and jitter in the control system of a remote medical treatment-assistant can lead to mission failure. Lastly, sufficient availability of network resources is imperative for achieving correct analysis results, because the generated traffic may be crucial for a patient's health and life.

I.5 Smart ward service

I.5.1 Overview of the smart ward service

A smart ward service inside a hospital provides efficient health care to patients, minimizes the nursing workload and facilitates a doctor's diagnosis. Patients, doctors, nurses and medical assets are connected together as shown below in Figure I.7. This makes a ward smart. The patient can move freely around the hospital and wears only a few wearable devices. The wearable devices can detect the patient's physiological parameters and location. The physiological parameters are uploaded directly to the electronic medical record (EMR) system. Doctors can access the patient information anywhere. The connection between nurses and patients creates a safer and more efficient care environment.

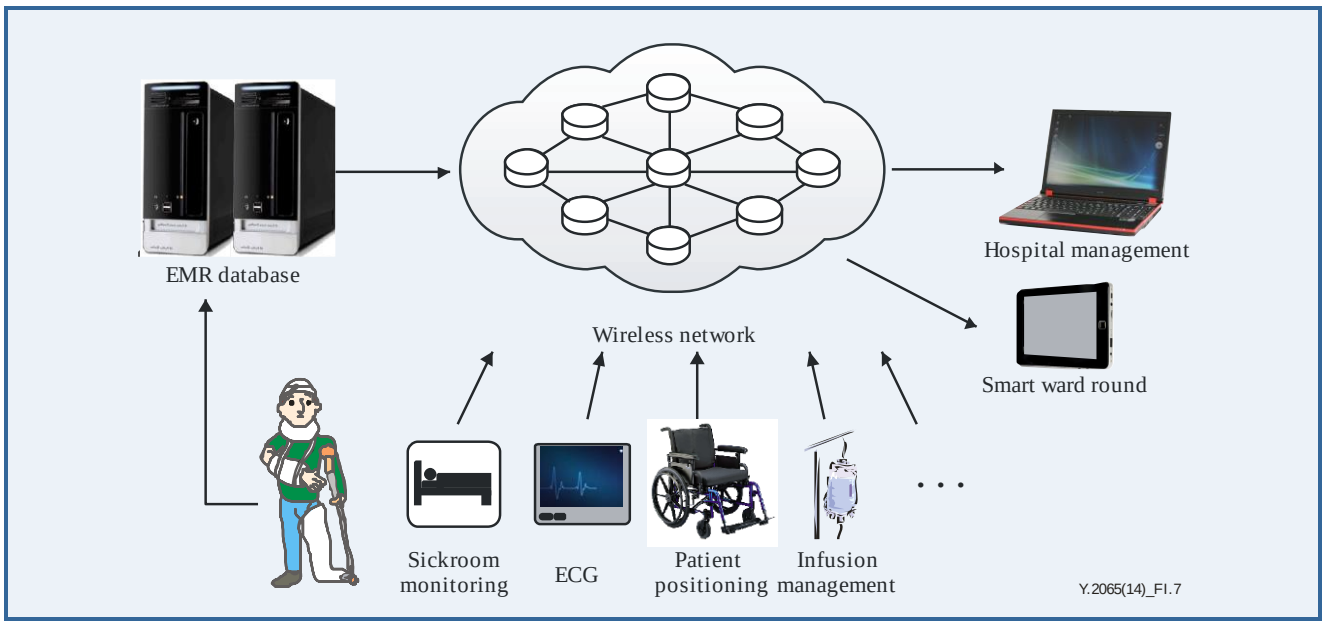


Figure I.7 – Smart ward network

The effects of nursing rounds can be improved through the smart ward service. Diagnostic results and electronic medical records can be displayed to patients anywhere. Tracking of patients is critical for the clinical risk management process, particularly for a hospital ward where patients need intensive care. When a patient's condition suddenly deteriorates, the smart ward service can

identify and locate the patient. The patient care flow is often delayed when a medical asset cannot be found, and the smart ward service provides medical asset management to reduce the delay associated with asset search. By reducing the search time, nurses have more time to treat the patients.

The smart ward service can be broken into the three main components of physiological parameters monitoring, indoor patient tracking and medical asset management:

1) the movement physiological parameters monitoring involves the acquisition of physiological parameters in movement and then the analysis of the data;

2) the indoor patient tracking is used to locate the patients inside a building;

3) the medical asset management system can locate the desired medical asset.

I.5.2 The requirements of smart ward services

1) Time critical service

In health-care environments, delayed or lost information may be vital. Therefore, reliable transmission must be guaranteed. Immediate action has to be generally undertaken as a response to the received data. For example, if a patient falls down, the patient's location should be reported to hospital staff immediately.

2) Simplicity

Service operation should be convenient for users, who may not be experts in the wireless network field.

3) Low power radiation

The wireless network is used in proximity to a human body. As a result, the radiation of the wireless network should not pose a health risk.

4) Low power consumption

The power budgets of wearable devices are constrained, requiring low power communication solutions. The wireless network should support low power mechanisms.

I.6 Chronic disease care

In the chronic disease care scenarios of e-health applications, there is a body area network concept of e-health; this is the collection of physiological parameters like blood pressure, blood oxygen, pulse rate, ECG, body temperature, blood sugar and others by computers, mobile phones, PDA or other gateway devices via sensors worn around the human body. The sensors can get the physiological parameters and transmit them by wireless means to the data centre. The data centre gets the data, analyses them and then sends the patients their results. Based on this, the patient can achieve real-time detection, and the doctor can provide each patient with health guidance.

The core of the chronic disease care service is the delivery and sharing of patient information, including information between different departments in the hospital, between hospitals, even between hospitals and the community, health insurance and government departments. This requires the devices to combine the sensor capability, computation capability and network connectivity capability. The sensor capability of the device collects the patient's physiological parameters in real-time. The computation capability of the device preprocesses the collected physiological parameters. Through the network connectivity capability of the device, the preprocessed physiological parameters are sent to the data centre. The medical staff obtain the patient's processed physiological parameters and other related information from the data centre and then, based on this information they make appropriate decisions which will be eventually sent back to the patients. Figure I.8 below illustrates the general architecture of a chronic disease care service.

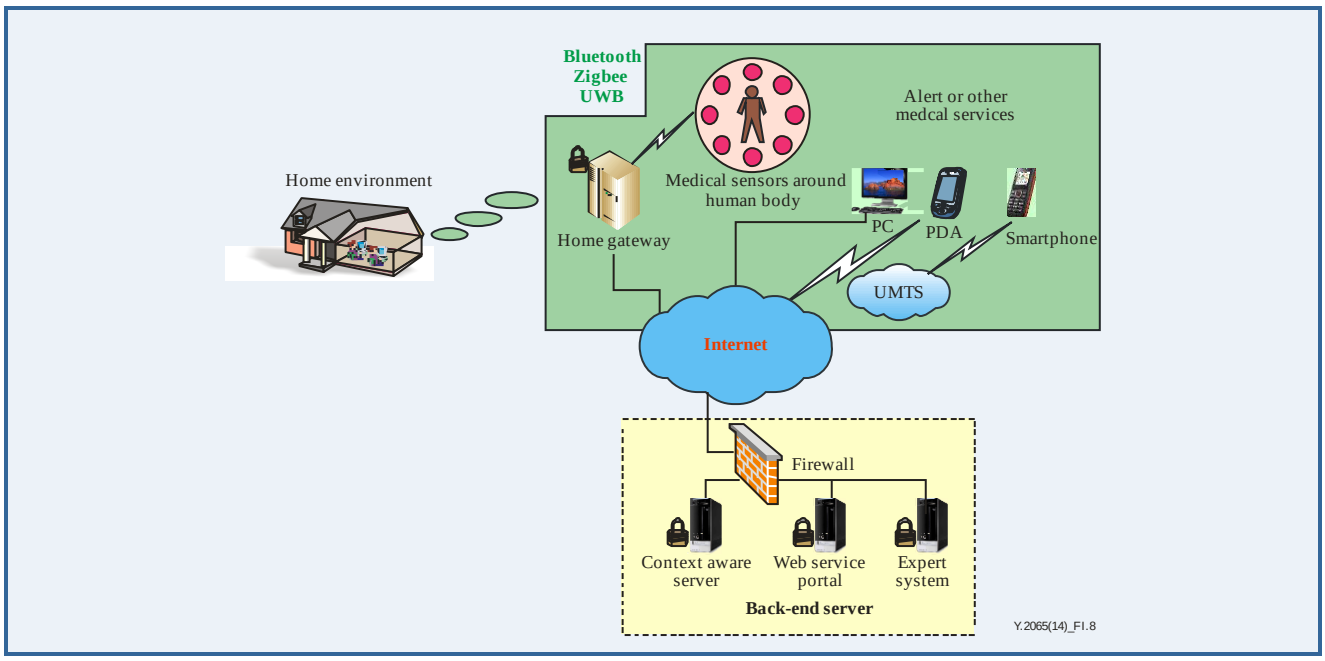


Figure I.8 – Chronic disease care service scenario

In Figure I.8, for the home environment, a variety of wireless access technologies and networks are shown. Physiological parameter sensors (such as blood pressure sensors, heart rate sensors, etc.), or other sensors (such as motion detection sensors) are worn if needed for the collection of a patient's monitoring parameters. Data collected through short-range wireless technologies (Bluetooth, Zigbee, UWB, etc.) are transmitted to a gateway (the gateway may be embedded in the family's ADSL box, personal computers, mobile phones, PDAs, etc.). Through the gateway, the patient's daily data are sent to the hospital in order to achieve real-time monitoring and expert guidance. A variety of health-care services in the home environment need to be supported by a back-end server. The care of chronic diseases (such as diabetes, heart disease, etc.) can be typically done by using a monitoring application.

The features of a chronic disease care service include, but are not limited to, the following concerning:

– Providers of a chronic disease care service:
  - Doctors handle the abnormal results. If a diagnosis from this data gives an abnormal result which might imply a risk of disease for the patient, an associated doctor is informed of the result. Effective action is then taken by the doctor.
  - A data centre is the core of the whole system. It deals with all the data including user information, doctor information, device information and physiological parameters. Large storage and high speed processing requirements must be satisfied. The algorithms to process the data are another key factor which determines the effectiveness of the whole system.
  - Devices can be rented or sold to the users. They can measure the user's physiological parameters automatically and can send the data to the data centre by wired/wireless communication.

– Users of chronic disease care service
  - In the chronic disease care service, the elderly are the prime users. As life expectancy is increasing more and more in many countries, the number of users who need the chronic disease care service will increase.

- More and more people will have health issues and will be classed as 'not fully healthy'. Those in this category may become users of the chronic disease care service.
- Users of the system want their health to be monitored automatically without the need to go to hospital every day. In this way, some hidden health risks for the user should be detected in time.

– Device requirements

Patient demand for monitoring devices might vary, for example, some patients only need a few parameters to be monitored, while others only require monitoring for specific time periods. The flexibility of device configuration in order to meet the needs of different users should be considered.

– Network requirements

As well as bandwidth and transmission speed, user mobility requirements should be considered. The heterogeneous coverage of wireless mobile networks can ensure access to a wide range of applications at anytime and anywhere.

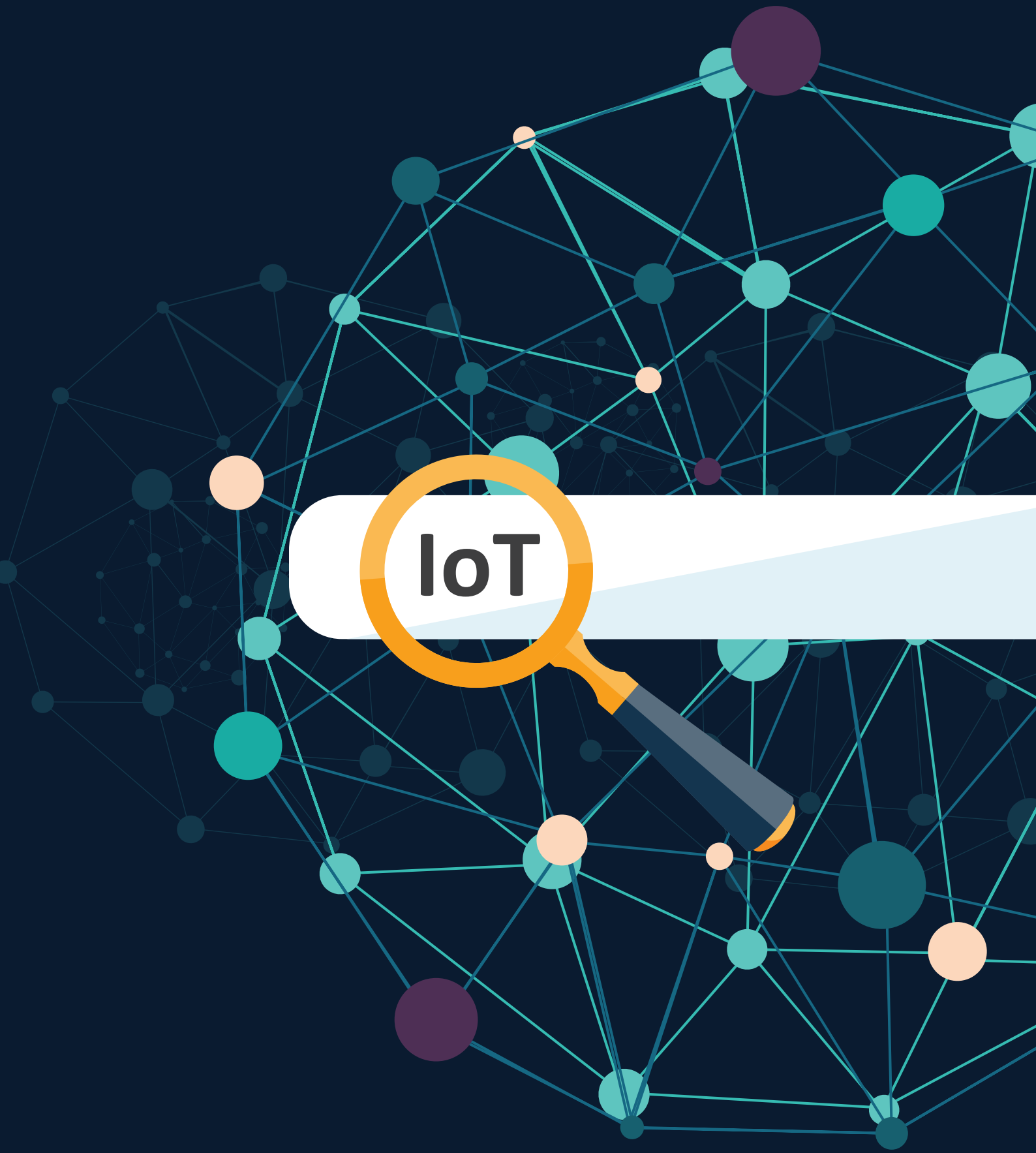
– System availability requirements

The chronic disease care service must be available all the time. The user may need to measure his/her physiological parameters at any time of the day.

– System precision requirements

Precision must be achieved. Only precise data can guarantee users with appropriate services. Otherwise, inaccurate data or inaccurate diagnosis results might lead to errors or severe incidents.

Internet of Things (IOT)


IoT

Y.4111/Y.2076

Semantics based requirements and framework of the Internet of Things

Semantics based requirements and framework of the Internet of things

Summary

Recommendation ITU-T Y.4111/Y.2076 specifies the semantics based requirements and framework of the Internet of things (IoT) as a basis for further IoT semantics based standardization work, including semantic aspects for IoT services in different business domains, semantically enhanced IoT capabilities and others.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T Y.4111/Y.2076	2016-02-13	13	11.1002/1000/12705

Keywords

Internet of things, semantics based capability framework, semantics based requirements.

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

Table of Contents


www.
8:20
8:20
Friday, March 30

Recommendation ITU-T Y.4111/Y.2076

Semantics based requirements and framework of the Internet of things

1 Scope

This Recommendation specifies the semantics based requirements and framework of the Internet of things (IoT).

Taking into consideration the IoT reference model [ITU-T Y.4000] and building on the common requirements of IoT [ITU-T Y.4100], semantics based requirements are specified, including those related to the four layers (application, service support and application support (SSAS), network, and device layer) and the management and security capabilities of the IoT reference model, as well as semantics based requirements across layers.

Based on the identified IoT semantics based requirements and existing semantic technologies, the semantics based capability framework of the IoT is specified.

The scope of this Recommendation includes:

– introduction to semantic technologies for the IoT;

– semantics based use cases for IoT actors;

– semantics based requirements of the IoT;

– semantics based capability framework of the IoT.

Appendix I provides IoT applications scenarios highlighting the value of semantic technologies in the IoT.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T Y.4000] Recommendation ITU-T Y.4000/Y.2060 (2012), *Overview of Internet of things*.

[ITU-T Y.4100] Recommendation ITU-T Y.4100/Y.2066 (2014), *Common requirements of the Internet of things*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 Internet of things (IoT) [ITU-T Y.4000]: A global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on, existing and evolving, interoperable information and communication technologies.

NOTE 1 – Through the exploitation of identification, data capture, processing and communication capabilities, the IoT makes full use of things to offer services to all kinds of applications, whilst ensuring that security and privacy requirements are fulfilled.

NOTE 2 – In a broad perspective, the IoT can be perceived as a vision with technological and societal implications.

3.1.2 ontology [b-ITU-T X.1570]: An explicit specification of a conceptualization.

3.1.3 thing [ITU-T Y.4000]: With regard to the Internet of things, this is an object of the physical world (physical things) or the information world (virtual things), that is capable of being identified and integrated into communication networks.

3.1.4 semantics [b-ITU-T Z.341]: The rules and conventions governing the interpretation and assignment of meaning to constructions in a language.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 data model: A representation structure for data that can organize data as elements in the structure and standardize the meaning of data elements and their relationships.

NOTE – Data models usually use vocabularies to describe their data elements and data elements' relationships. A semantic data model uses vocabularies complying with ontologies. For more information, refer to http://www.w3.org/standards/semanticweb/.

3.2.2 data set: A collection of data that conforms to a particular data model.

NOTE – A semantic data set conforms to a semantic data model (it can be a collection of native semantic data or a collection of semantically annotated data). For more information, refer to http://www.w3.org/standards/semanticweb/.

3.2.3 IoT ontology: An ontology for the IoT that includes the union of ontologies for the different components of the IoT, including the relationships between these ontologies.

NOTE 1 – An important part of the IoT ontology concerns the ontologies for IoT devices and things.

NOTE 2 – The development of the IoT ontology is an evolving process that is expected to take into account new concepts as far as needed along its development.

3.2.4 query: Technology that can programmatically retrieve information from data sets.

NOTE – A semantic query uses semantic technologies to retrieve information from semantic data sets. For more information, refer to http://www.w3.org/standards/semanticweb/ .

3.2.5 semantic description language: Language used to formally model and describe ontologies. For more information, refer to http://www.w3.org/standards/semanticweb/.

3.2.6 vocabulary: The set of terms defined, classified and used to describe concepts and relationships of a particular area of concern.

NOTE – The word "ontology" is used for a more complex and quite formal collection of terms, whereas "vocabulary" is used when such strict formalism is not necessary. For more information, refer to http://www.w3.org/standards/semanticweb/.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

AL Application Layer

DL Device Layer

DM Device Management

IoT Internet of Things

NL Network Layer

OWL Web Ontology Language

RDF	Resource Description Framework
RIF	Rule Interchange Format
SMS	Semantic Management Support
SMSC	Semantic Management Support Capabilities
SPARQL	SPARQL Protocol and RDF Query Language
SSAS	Service Support and Application Support
SSASL	Service Support and Application Support Layer
SSSC	Semantic Security Support Capabilities
UML	Unified Modelling Language
XACML	Extensible Access Control Markup Language

5 Conventions

The keywords "is required to" indicate a requirement which must be strictly followed and from which no deviation is permitted if conformance to this Recommendation is to be claimed.

The keywords "is recommended" indicate a requirement which is recommended but which is not absolutely required. Thus this requirement needs not be present to claim conformance.

The keywords "can optionally" and "may" indicate an optional requirement which is permissible, without implying any sense of being recommended. These terms are not intended to imply that the vendor's implementation must provide the option and the feature can be optionally enabled by the network operator/service provider. Rather, it means the vendor may optionally provide the feature and still claim conformance with the specification.

6 Introduction to semantic technologies for the IoT

Due to the growing number of interconnected things and related communication connections, as well as the variety of IoT devices and related connectivity, the volume and types of data generated by the things as well as the number and type of services provided by the IoT infrastructure are increasing more and more quickly. As a result of these phenomena, requirements for automatic operations, consistency, interoperability and reusability of the IoT infrastructure are becoming more and more urgent.

Semantic technologies (i.e., technologies based on semantics) are promising candidates to meet the foresaid requirements for the IoT infrastructure. Semantic technologies for IoT enable the efficient description of data (e.g., collected data and virtual representation of physical things) and services, so that machines and humans can have a common understanding of the exchanged data and processed services in the IoT infrastructure in order to benefit automatic operations, analysis and processing activities. In addition, semantic technologies can enhance representation, annotation, discovery, analytics, interoperability, reusability and composability of data and services.

The semantic technologies are applicable to the different layers of the IoT reference model [ITU-T Y.4000]. For example: in the application layer (AL), semantic technologies can help to provide users with a smart human-machine interface; in the service support and application support layer (SSASL), semantic technologies can help capabilities and resources deployed in distributed nodes (e.g., devices, gateways, servers) to be discovered and interoperated in an automatic way; in the network layer (NL), semantic technologies can simplify and help to automate network configuration; in the device layer (DL), semantic technologies can help the IoT infrastructure to understand different device properties such as computation and storage capacity and sensor types, etc.

Concerning security capabilities of the IoT reference model, semantic technologies can benefit the security related decision making (e.g., based on semantic technologies, resources access rights can be deduced). Furthermore, semantic technologies can enhance the conventional description of security policies, thus helping the security negotiation process between different IoT components (e.g., device, IoT application server, platform and network).

Concerning management capabilities of the IoT reference model, semantic technologies can facilitate the understanding of service logging reports by both machines and humans and also facilitate automatic configuration of IoT components.

In summary, semantic technologies reveal outstanding features for applicability to the IoT, including, but not limited to, the following:

- consistency: Through semantic technologies, data and services can refer to the same meaning across time, location and IoT components;
- scalability: Through semantic technologies, data and services can be managed locally to IoT components (e.g., semantic annotation can help local interpretation of data reducing the need to involve other IoT components). This increases the IoT technical component independence (components become loosely coupled) and decentralizes the management, leading to enhanced scalability. Moreover, the service reachability can be more easily expanded to reach more users and the functional evolution of services can be rationalized;
- re-usability: Through semantic technologies, data and services can be reused and composed to construct new data and services;
- analytics and actionable knowledge: Through semantic technologies, merging, correlation and analysis of diverse data generated by the IoT, together with data from external sources such as social media, events and news, can be facilitated in order to produce actionable knowledge;
- interoperability: Based on semantic technologies, the interoperability level [b-SI] of data and services of the IoT within one application domain and/or among different application domains can be improved;
- human-machine interaction: On one hand, since semantic technologies are based on natural human concepts, data and services become easier for humans to understand. On the other hand, since semantic technologies are formal ways to express concepts, data and services can also be understood by machines. This can improve the interaction between humans and the IoT.

7 Semantics based use cases for IoT actors

The IoT actors considered in these use cases are from those defined in [ITU-T Y.4100].

Actors and use case diagrams are modelled in unified modelling language (UML) [b-UML].

In Figure 7-1, some semantics based use cases for IoT actors are identified.

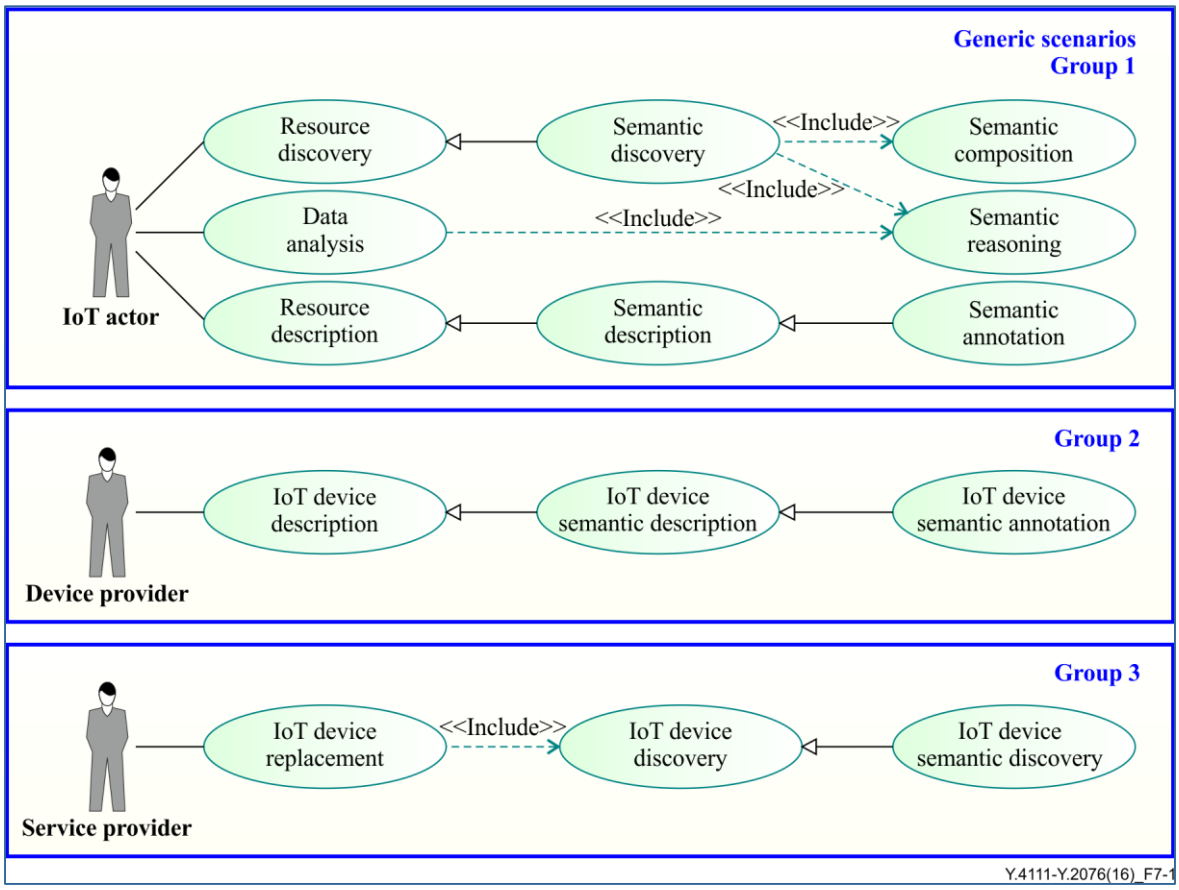


Figure 7-1 – Semantics based use cases for IoT actors

These groups of use cases shown in Figure 7-1 describe a non-exhaustive set of scenarios from which corresponding requirements can be derived.

The first group of use cases (Group 1) deals with generic scenarios where semantics is used by IoT actors, including for resource description, resource discovery and data analysis.

The second group of use cases (Group 2) deals with the description of IoT devices by the "device provider" actor that uses the semantic description of IoT devices.

The third group of use cases (Group 3) deals with the replacement of an IoT device. A "service provider" actor semantically discovers new IoT devices that have replaced original IoT devices.

8 Semantics based requirements of the IoT

8.1 General semantics based requirements for IoT

NOTE – Appendix I describes some IoT application scenarios using semantic technologies which address requirements identified in this clause.

8.1.1 IoT ontology

As defined in clause 3, the IoT ontology is an ontology for the IoT that includes the union of ontologies for the different components of the IoT, including the relationships between these ontologies.

IoT ontology is required to be the foundation of the IoT for semantic annotation, semantic interoperability, semantic discovery, semantic reasoning and semantic composition so that consistent meanings and relationships can be setup across different IoT components.

Existing ontologies related to the IoT such as for example semantic sensor network ontology [b-SSNO], are recommended to be integrated into a complete IoT ontology ecosystem, which covers all the components of the IoT.

NOTE – How to expand and reuse existing ontologies related to the IoT in the IoT ecosystem is outside of the scope of this Recommendation.

8.1.2 Semantic annotation

To realize the added value of semantics, the data exchanged in the IoT need to be described via semantic descriptions.

Semantic annotation, as a non-intrusive technique, is generally used for the semantic description of IoT data in order to describe characteristics of IoT resources (e.g., collected data, IoT devices and IoT applications) and to indicate relationships between IoT resources in a consistent and maintainable way.

It is required to use semantic annotation to implement semantic description.

The semantic annotation is required to be based on IoT ontology and appropriate data models, as well as predefined description languages.

It is required to support interoperability between different semantic description languages used for semantic annotation.

NOTE – One possible way to achieve interoperability is via language translation.

8.1.3 Semantic interoperability

Semantic interoperability addresses interoperability at the semantic level [b-SI], based on the meaning of exchanged data between IoT components rather than only on the representation of exchanged data. The meaning of exchanged data can be described via semantic annotation.

In practice, semantic technologies can provide support for an increased level of interoperability among different IoT components. These technologies may be implemented in different ways in the different IoT components, according to the different parties and the different application domains.

Semantic interoperability depends on the semantic annotation of the exchanged data and the interfaces of the interacting IoT components.

Semantic interoperability is recommended in the IoT so that the data transferred across different IoT components can be easily understood by each IoT technical component.

8.1.4 Semantic discovery

Semantic discovery enables the discovery of IoT resources via the meaning of query requests (semantic query) rather than the query requests' data sets.

Semantic discovery depends on IoT ontology, semantic annotation of IoT resources and semantic query.

Semantic discovery is recommended in the IoT so that IoT resources can be discovered according to the meaning of query requests.

8.1.5 Semantic reasoning

Semantic reasoning enables reasoning based on semantic annotation of IoT resources, IoT ontology and semantic rules.

Based on IoT ontology and semantic rules, semantic reasoning analyses semantic annotation to obtain implicit meanings and relationships concerning IoT resources. For example, in the case of semantic annotation describing a device as "daylight lamp", semantic reasoning can infer "daylight lamp" is also a kind of "lamp".

Semantic reasoning is recommended in the IoT so that implicit meanings and relationships concerning IoT resources can be deduced from semantically annotated information.

Semantic rules are required to be based on IoT ontology so that the meaning of semantic rules can be consistent across IoT components.

8.1.6 Semantic composition

Based on IoT ontology and semantic annotation of IoT resources, semantic composition can compose appropriate IoT resources to create new (semantically annotated) IoT resources. As a concrete example, via semantic composition, the data for "average temperature" of a room can be created by composing the "temperature" data from several individual sensors in the room.

The rules of semantic composition are based on IoT ontology.

Semantic composition can be used when straightforward semantic discovery fails to satisfy a particular semantic query. In such a case, the semantic composition process can start with an adequate query decomposition process, followed by the semantic discovery processes launched by those partial queries. The returned results for partial queries can then be adequately composed in order to satisfy the original semantic query.

Semantic composition is recommended in the IoT so that new (semantically annotated) IoT resources can be created based on existing ones.

NOTE – The newly created resources are then automatically annotated.

8.2 Semantics based requirements for IoT with respect to the IoT reference model

The following clauses describe semantics based requirements for IoT with respect to the different layers and cross-layer capabilities of the IoT reference model [ITU-T Y.4000].

8.2.1 Semantics based requirements for the device layer

1) Semantic annotation

The device layer (DL) is required to be empowered by semantic annotation for the description of IoT devices and their collected data:

– semantic annotation for IoT devices

Semantic annotation is required to be supported for the description of IoT devices.

Semantic annotation is required to be based on ontology and the semantic data model of the IoT as well as predefined semantic description languages. In this way, the data sets related to IoT devices, e.g., the type and functions of an IoT device, the operations supported by the IoT device and other information, can be correctly understood by other IoT components.

– semantic annotation for the collected data

Semantic annotation is required to be supported for the description of the data collected by IoT devices.

Semantic annotation is required to be based on ontology and the semantic data model of the IoT as well as predefined semantic description languages. In this way, the information related to the collected data, e.g., the meaning, origin, standard value and reliability of the collected data as well as other information can be correctly understood by other IoT components.

The data collected by IoT devices are recommended to be transformed into a semantic format, e.g., resource description framework (RDF) triples [b-RDF11].

The data collected by IoT devices are required to be linked to predefined semantic data models complying with the IoT ontology.

The data collected by IoT devices are recommended to be associated with provenance information.

NOTE – Provenance information is useful to provide correct contextualisation and traceability of the collected data.

8.2.2 Semantics based requirements for the SSAS layer

1) Semantic annotation

The SSAS layer is required to support semantic annotation for the description of IoT resources.

NOTE – The IoT resources semantically annotated in the SSAS layer can belong to different layers.

All semantically annotated information is required to be organized according to predefined data models.

2) Semantic discovery

The SSAS layer is recommended to support semantic discovery of IoT resources.

3) Semantic interoperability

The SSAS layer is required to support semantic interoperability, based on the semantically annotated information of IoT resources. In this way, IoT resources can be accessed, understood and exchanged by different IoT components.

4) Semantic reasoning

The SSAS layer is recommended to support semantic reasoning for IoT resources. With semantic reasoning, the semantically annotated information of IoT resources can be further enriched.

5) Semantic composition

The SSAS layer is recommended to support semantic composition of IoT resources. With semantic composition, the SSAS layer can compose IoT resources to create new resources based on the semantically annotated information of IoT resources.

8.2.3 Semantics based requirements for the network layer

1) Semantic annotation

The network layer (NL) is required to be empowered by semantic annotation for the description of IoT resources of the network layer.

Semantic annotation is recommended to be supported to facilitate the network configuration based on ontology and the semantic data model of the IoT. In this way, the information related to the network operations, e.g., the status of the network, the type of the network (e.g., fixed network, mobile network, etc.), the capabilities which can be exposed by the network and other information, can be correctly understood by other IoT components.

2) Semantic discovery

The network layer is recommended to be empowered by semantic discovery for the discovery of IoT resources of the network layer.

Semantic discovery is recommended to be supported in order to find relevant IoT resources, e.g., network interfaces or network connections, based on semantic query and ontology.

8.2.4 Semantics based requirements for the application layer

1) Semantic annotation

The application layer (AL) is required to be empowered by semantic annotation for describing IoT resources of the application layer and in support to semantic discovery of IoT resources.

2) Semantic discovery

The application layer is recommended to be empowered by semantic discovery for discovering the desired IoT resources.

3) Semantic reasoning and semantic composition

The application layer is recommended to be empowered by semantic reasoning.

The application layer is recommended to be empowered by semantic composition.

8.2.5 Semantics based requirements for the management capabilities

The requirements for the management capabilities are as follows:

1) Semantic annotation

The management capabilities are required to be empowered by semantic annotation for the description of management functions and related parameters. The management functions of IoT are generally distributed across different IoT components. With the help of semantic annotation, the information related to the management functions, e.g., device remote triggers and software/firmware updates, etc., can be correctly understood by all IoT components.

2) Semantic interoperability

The management capabilities are required to be empowered by semantic interoperability to support interworking among different types of management operations (e.g., among management operations based on TR069 protocols [b-TR069] and management operations based on Open Mobile Alliance (OMA) device management (DM) protocols [b-OMADM]). In such a way, the management functions, whose implementation can be based on various technologies or standards, can be triggered across different IoT components.

8.2.6 Semantics based requirements for the security capabilities

The requirements for the security capabilities are as follows:

1) Semantic annotation

The IoT uses a plethora of heterogeneous protocols and technologies for security, which may originate interoperability issues.

Security capabilities are recommended to be empowered by semantic annotation for security policies and mechanisms.

2) Security policy management

The IoT is required to support security policy management capabilities in order to provide control rules pertaining to access control, privacy protection, trust and authentication, etc.

Security policies are recommended to be specified using semantic description languages, such as RDF [b-RDF] and web ontology language (OWL) [b-OWL], or using a dedicated ontology.

3) Access control

The IoT is required to support access control capabilities in order to provide protection of information by restricting access to and/or modification of IoT resources only to those entities authorized to do so.

In order to efficiently do so, access control rules are required to be provided (e.g., by policy information points [b-RFC 2904]) and be comprehensible.

NOTE – Access control rules can be specified declaratively using specific access policy description languages, such as extensible access control markup language (XACML) [b-XACML], or using semantic description languages, such as RDF and OWL, or using a dedicated ontology.

9 Semantics based capability framework of the IoT

9.1 Overview

In this clause, the IoT semantic capabilities are derived based on the requirements in clause 8.

9.1.1 The distribution of semantic capabilities in the IoT reference model

Figure 9-1 positions the semantic capabilities at the various layers as well as the cross-layer of the IoT reference model [ITU-T Y.4000]: application layer, SSAS layer, network layer, device layer, management capabilities and security capabilities. It also shows the high-level relationships among the semantic capabilities positioned at these various layers and the cross-layer.

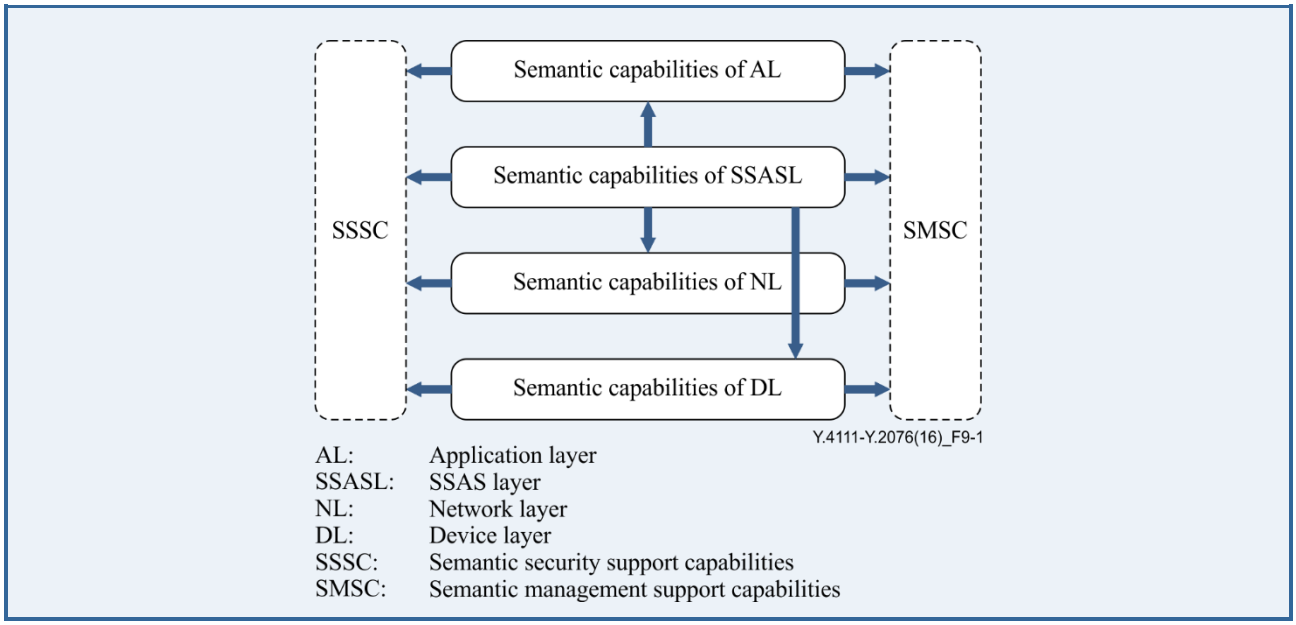


Figure 9-1 – Semantic capabilities in the IoT reference model and their relationships

Concerning the four horizontal layers, the starting points of the directional arrows shown in Figure 9-1 are the semantic capabilities of the SSAS layer and the end points of these directional arrows are the semantic capabilities of the application layer, network layer and device layer. The intention is to indicate that the semantic capabilities of the SSAS layer can be invoked by the semantic capabilities of the application layer, network layer and device layer.

Concerning the cross-layer capabilities, the directional arrows start from the application layer, SSAS layer, network layer and device layer and end at semantic security support capabilities (SSSC) or semantic management support capabilities (SMSC). The intention is to indicate that the SMSC and SSSC of the IoT can invoke the semantic capabilities in the application layer, the SSAS layer, the network layer and the device layer.

Semantic capabilities of the IoT can operate on exposed IoT resources.

9.1.2 Global view of the IoT semantics based capability framework

Figure 9-2 provides a global view of the IoT semantics based capability framework based on the IoT reference model [ITU-T Y.4000].

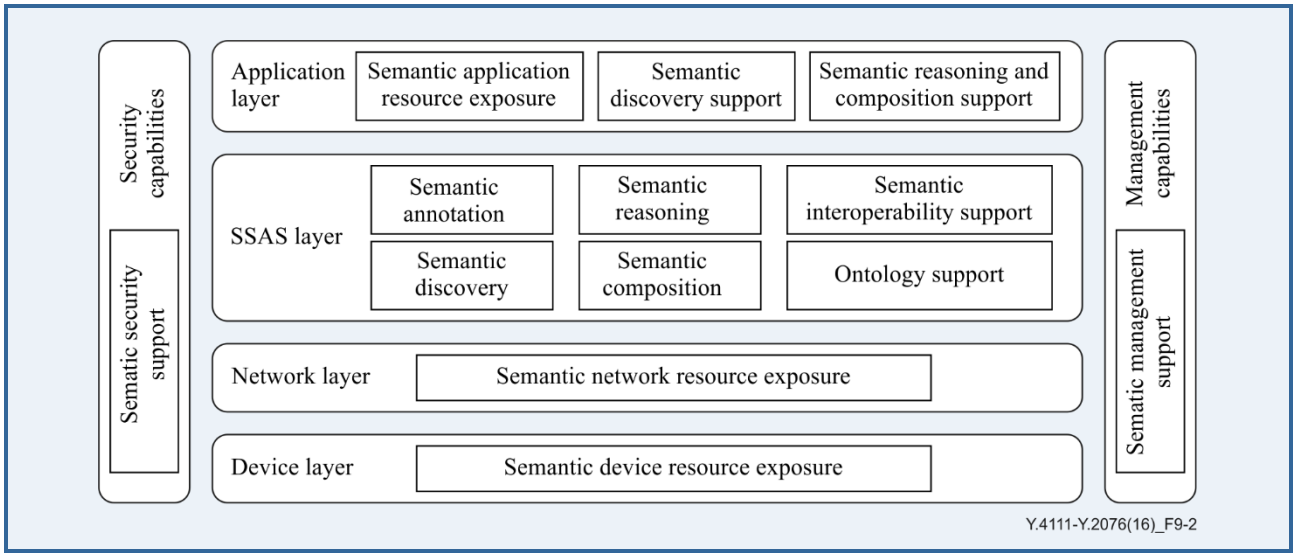


Figure 9-2 – Global view of the IoT semantics based capability framework

The SSAS layer supports semantic capabilities, including semantic annotation, semantic discovery, semantic reasoning, semantic composition and semantic interoperability support capabilities, in order to meet the requirements described in clause 8.2.2. The SSAS layer also provides the ontology support capability for semantic enablement at all layers.

The network layer supports the network resource semantic exposure capability in order to expose the resources in the network layer to the SSAS layer for semantic annotation and semantic discovery. The exposure capability conforms to the requirements in clause 8.2.3 which state that the network layer is recommended to expose its resources to the SSAS layer for semantic annotation and discovery.

The device layer supports the device resource semantic exposure capability in order to expose the resources in the device layer to the SSAS layer for semantic annotation and semantic discovery. The exposure capability conforms to the requirements in clause 8.2.1 which state that the device layer is required to expose its resources to the SSAS layer for semantic annotation and discovery.

The application layer supports the application resource semantic exposure capability in order to expose the resources in the application layer to the SSAS layer for semantic annotation and semantic discovery. The exposure capability conforms to the requirements in clause 8.2.4 which state that the application layer is required to expose its resources to the SSAS layer for semantic annotation and discovery. The application layer also supports the semantic discovery support capability and the semantic reasoning and composition support capability in order to enable the usage by IoT applications of semantic discovery, semantic reasoning and semantic composition capabilities in the SSAS layer according to the requirements in clause 8.2.4.

Security capabilities and management capabilities support, respectively, the semantic security support capability (SSSC) and the SMSC, in order to semantically enhance the security capabilities and the management capabilities.

9.1.3 The exposure of IoT resources

There are IoT resources at the different layers of the IoT reference model.

NOTE 1 – The IoT resources associated with the security and management capabilities of the IoT reference model can be seen as resources distributed in the four layers of the IoT reference model.

Figure 9-3 describes the relationship among IoT resources and the various semantic resource exposure capabilities.

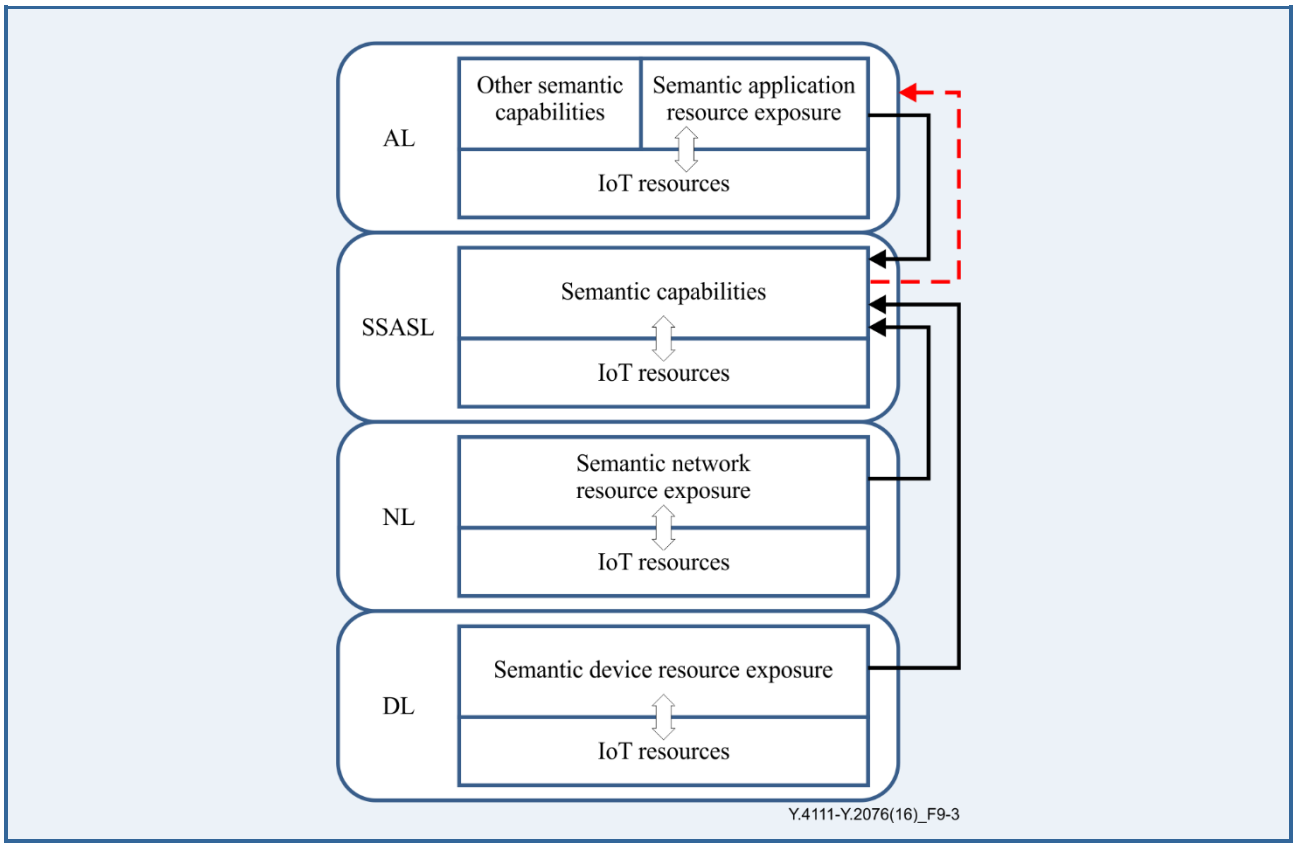


Figure 9-3 – Relationship among IoT resources and the various semantic resource exposure capabilities

As shown in Figure 9-3, the IoT resources in the device layer, the network layer and the application layer can be exposed to the SSAS layer via, respectively, the semantic device resource exposure capability, the semantic network resource exposure capability and the semantic application resource exposure capability. The SSAS layer can expose IoT resources of each layer to the application layer.

NOTE 2 – In Figure 9-3, the semantic capabilities in the SSAS layer include: semantic annotation, semantic discovery, semantic reasoning, semantic composition and semantic interoperability support capabilities.

NOTE 3 – The other semantic capabilities in the application layer indicated in Figure 9-3 include: semantic discovery support capability and the semantic reasoning and composition support capability.

The details about these resource exposure capabilities, as well as other semantic capabilities at each layer, are specified in the following clauses.

9.2 Application layer

9.2.1 Semantic application resource exposure

The semantic application resource exposure capability is used to expose the IoT resources of the application layer to the SSAS layer for semantic annotation.

The semantic application resource exposure capability is required in the application layer in order to expose the IoT resources in the application layer based on a predefined data model. The exposed IoT resources can be described in a predefined format via possible support of a standard semantic description language.

9.2.2 Semantic discovery support

The semantic discovery support capability is used to support applications in the application layer for the invocation of the semantic discovery capability in the SSAS layer.

The semantic discovery support capability is required to use a predefined format and standard semantic query language (e.g., SPARQL [b-SPARQL]) in order to invoke the semantic discovery capability in the SSAS layer according to the applications' requests.

9.2.3 Semantic reasoning and composition support

The semantic reasoning and composition support capability is used to support the transfer of the semantic rules defined by applications from the application layer to the SSAS layer so that semantic reasoning and composition can be executed according to the applications' needs.

The semantic reasoning and composition support capability is recommended in the application layer.

If the semantic reasoning and composition support capability is enabled, it is required to use a predefined format and standard semantic languages such as rule interchange format (RIF) [b-RIF], to transfer the semantic rules according to the applications' needs.

9.3 SSAS layer

9.3.1 Semantic annotation

The semantic annotation capability is used to semantically annotate IoT resources exposed to the SSAS layer.

The semantic annotation capability is required in the SSAS layer to semantically annotate IoT resources based on IoT ontology in a standard semantic language. When the exposed IoT resource is not described in a semantic way, the semantic annotation capability needs to translate it into a standard semantic description, e.g., based on the data model used by the exposed IoT resource.

9.3.2 Semantic discovery

The semantic discovery capability enables the discovery of IoT resources via semantic queries.

The semantic discovery capability is required to support discovery filters described via standard semantic query languages (e.g., SPARQL).

The semantic discovery capability is required to be able to trigger semantic composition capabilities in case it fails to satisfy a particular semantic query.

9.3.3 Semantic reasoning

The semantic reasoning capability is used to analyse the explicit semantically annotated IoT resources in order to obtain some implicit information.

The semantic reasoning capability is recommended in the SSAS layer.

If the semantic reasoning capability is enabled, it is required to run semantic reasoning based on IoT ontology.

NOTE – The information derived via semantic reasoning can be added to the semantic description of the related IoT resources via semantic annotation, for example with the goal to benefit further semantic discovery operations.

9.3.4 Semantic composition capability

The semantic composition capability composes IoT resources in order to create new (semantically annotated) resources.

The semantic composition capability is required to be based on IoT ontology for the configuration of its composition rules (for composition and de-composition processes).

The semantic composition capability is required to support requests by the semantic discovery capability in the case where the semantic discovery capability fails to satisfy a particular semantic query.

9.3.5 Semantic interoperability support

The semantic interoperability support capability is used to support the exchange of semantic information among different IoT components for the purpose of semantic level interoperability.

The semantic interoperability support capability is required to use a predefined format to exchange semantic information.

NOTE – Examples of exchanged semantic information include semantically annotated data sets related to exchanged data as well as to interfaces of the interacting IoT components.

9.3.6 Ontology support capability

The ontology support capability provides IoT ontology for semantic capabilities.

The ontology support capability is required to provide IoT ontology for the following semantic capabilities:

- the semantic annotation capability in order to annotate IoT resources;
- the semantic discovery capability in order to resolve the meaning of queries;
- the semantic composition capability in order to configure composition rules;
- the semantic reasoning capability in order to analyse the semantically annotated information of IoT resources for obtaining implicit information;
- the semantic interoperability support capability in order to semantically annotate exchanged data and interfaces of the interacting IoT components.

The ontology support capability is required to be able to integrate within the IoT ontology existing ontologies as well as newly created ontologies.

9.4 Network layer

9.4.1 Semantic network resource exposure

The semantic network resource exposure capability is used to expose the IoT resources in the network layer to the SSAS layer for semantic annotation.

The semantic network resource exposure capability is recommended in the network layer to expose the IoT resources in the network layer based on a predefined format via possible support of a standard semantic description language.

9.5 Device layer

9.5.1 Semantic device resource exposure

The semantic application resource exposure capability is used to expose the IoT resources in the device layer to the SSAS layer for semantic annotation.

The semantic network resource exposure capability is required in the device layer to expose IoT resources in the device layer based on a predefined format via possible support of a standard semantic description language.

9.6 Management capabilities

9.6.1 Semantic management support capability

The SMSC enhances the management capabilities of the IoT via the support of semantically annotated data sets.

The semantic management support capability is recommended to support semantic annotation of management functions and semantic level interoperability of management functions using semantic annotation capability and semantic interoperability capability in the SSAS layer.

9.7 Security capabilities

9.7.1 Semantic security support capability

The semantic security support capability (SSSC) enhances the security capabilities of the IoT via the support of semantically annotated data sets.

The semantic security support capability is recommended to support semantic annotation of security policies (e.g., access control rules) and semantic level interoperability of security mechanisms using semantic annotation capability and semantic interoperability capability in the SSAS layer.

Appendix I

IoT application scenarios using semantic technologies

(This appendix does not form an integral part of this Recommendation.)

I.1 Semantics-enabled home automation

In this IoT application scenario, as shown in Figure I.1, a home gateway is deployed for the support of home automation applications.

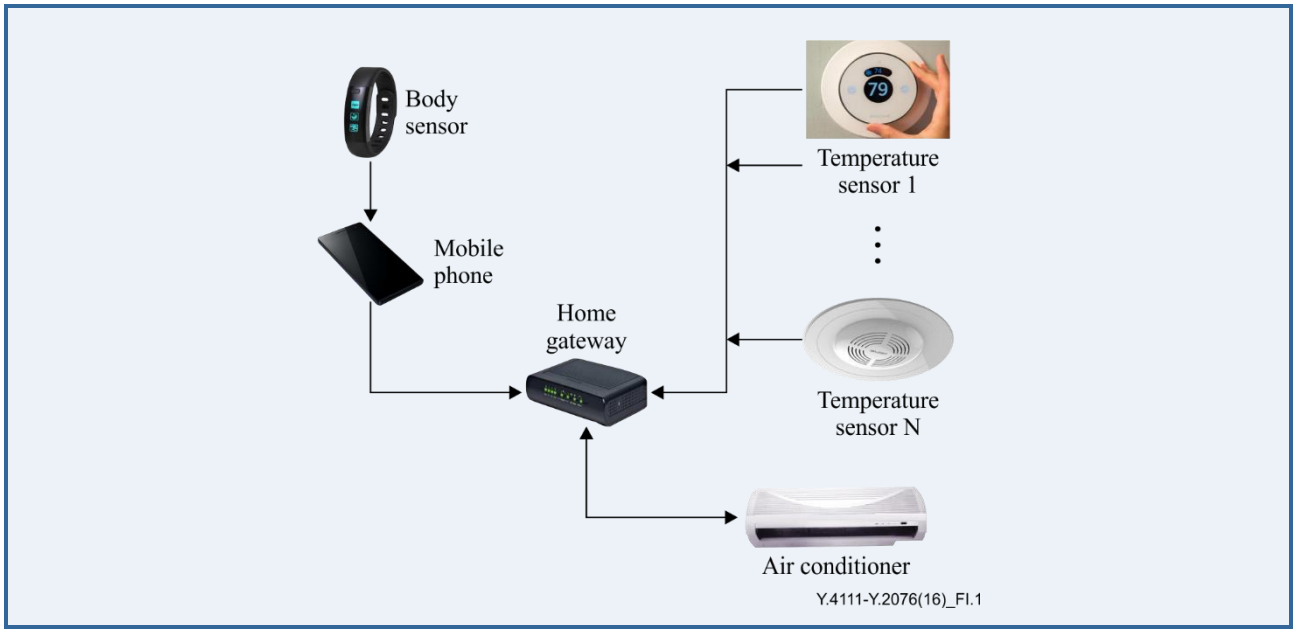


Figure I.1 – Semantics-enabled home automation

The body sensors of the home owner (e.g., embedded in the watch, sports band) collect the body monitoring data (e.g., skin temperature, heart rate, skin humidity) and send them to the mobile phone of the home owner.

When the home owner enters the house, the mobile phone automatically connects to the home gateway and sends the body monitoring data to the home gateway. The body monitoring data sent by the mobile phone are semantically annotated. Based on the semantic annotation, the home gateway can understand the meaning of the received body monitoring data and get the body status of the home owner.

In the case where the home gateway finds that the home owner's skin temperature is high, with the help of semantic reasoning, the home gateway can infer that the home owner may need a cooler environment. The home automation application in the home gateway can then initiate the following process:

1) The home automation application queries the average temperature of the room. With the help of semantic discovery and semantic composition, the home gateway automatically composes all the temperature sensors in the room to form a virtual object that combines all the measurements in order to generate an approximate average temperature. The formed virtual object directly provides the approximate average temperature of the room and exposes it to the home automation application in the home gateway.

2) The home automation application in the home gateway checks whether the approximate average temperature of the room is higher than a comfortably cool level or not. If the approximate average temperature of the room is higher, the home automation application will reduce the temperature.

3) The home automation application queries the device in the room that can reduce the temperature. With the help of semantic discovery, the air conditioner and its operations are exposed to the home automation application. The home automation application controls the air conditioner in order to reduce the room temperature to a cooler level until the relevant body monitoring data sent from the mobile phone become normal.

I.2 Semantics enabled location-based service

A semantics-enabled location-based service can provide users with a spatial inquiring service. IoT semantic technologies are used for the implementation of a semantics enabled location-based service.

As an example of this service, this clause describes the process related to a specific spatial inquiry: finding all gas stations within 10 miles offering a gas price lower than a given amount.

One possible way to offer a semantics-enabled location-based service is shown in Figure I.2.

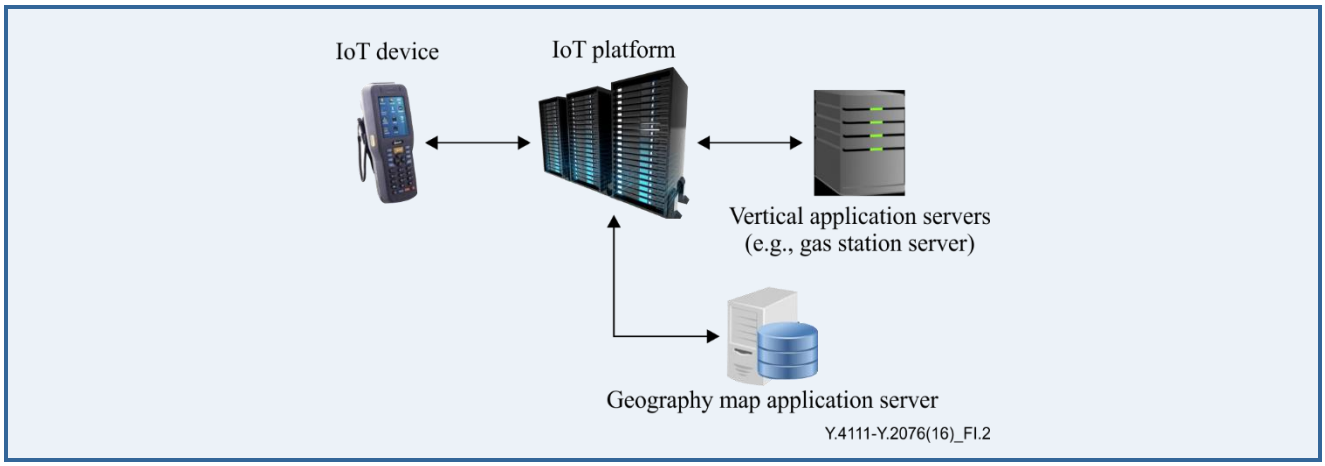


Figure I.2 – Semantics enabled location-based service

The application layer of the IoT device provides semantic discovery support functionality, which can send out query requests in a semantic way. For the specific inquiry above, i.e., finding all the gas stations within 10 miles with a gas price lower than a given amount, a query request is sent out to the IoT platform according to the specific IoT ontology used by the IoT device and in corresponding semantic description language. At the same time, the current location information of the IoT device (e.g., latitude and longitude) is sent out to the IoT platform.

The IoT platform retrieves the query request and triggers the semantic discovery functionality. The semantic discovery functionality fails because it cannot find an IoT resource that meets the query request directly. Then a semantic composition functionality is triggered which composes feedback from a geographical map application server and feedback from multiple gas station application servers. The geographical map application server returns the list of gas stations that are within 10 miles of the IoT device. Gas station application servers feedback corresponding gas stations' location information. The semantic composition functionality of the IoT platform composes these feedbacks and produces the answer for the query request of the IoT device.

Bibliography

[b-ITU-T X.1570] Recommendation ITU-T X.1570 (2011), *Discovery mechanisms in the exchange of cybersecurity information*.

[b-ITU-T Z.341] Recommendation ITU-T Z.341 (1988), *Glossary of terms*.

[b-ETSI-TR 101 584] ETSI TR 101 584 V2.1.1 (2013), *Machine-to-Machine Communications (M2M); Study on Semantic support for M2M Data*.
<http://www.etsi.org/deliver/etsi_tr/101500_101599/101584/02.01.01_60/tr_101584v020101p.pdf>

[b-RFC 2904] IETF RFC 2904 (2000), *AAA Authorization Framework*.
<https://tools.ietf.org/html/rfc2904>

[b-XACML] OASIS Standard (2013), *eXtensible Access Control Markup Language (XACML) Version 3.0*.
<http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.pdf>

[b-OMADM] Open Mobile Alliance (2012), *OMA Device Management V2.0*.
<http://technical.openmobilealliance.org/Technical/technical-information/release-program/current-releases/dm-v2-0>

[b-OWL] OWL (2012), *Web Ontology Language (OWL)*.
<http://www.w3.org/OWL/>

[b-RDF] W3C Specification (2004), *Resource Description Framework (RDF)*.
<http://www.w3.org/RDF/>

[b-RDF11] W3C Recommendation (2014), *Resource Description Framework (RDF). Concepts and Abstract Syntax*.
<http://www.w3.org/TR/rdf11-concepts/>

[b-RIF] W3C specification, *RIF Overview (Second Edition)*.
<http://www.w3.org/TR/rif-overview/>

[b-SemanticWeb] W3C Semantic Web, *Semantic Web overview*.
<http://www.w3.org/standards/semanticweb>

[b-SSNO] W3C Semantic Sensor Network Incubator Group (2005), *Semantic Sensor Network Ontology*.
<http://purl.oclc.org/NET/ssnx/ssn>

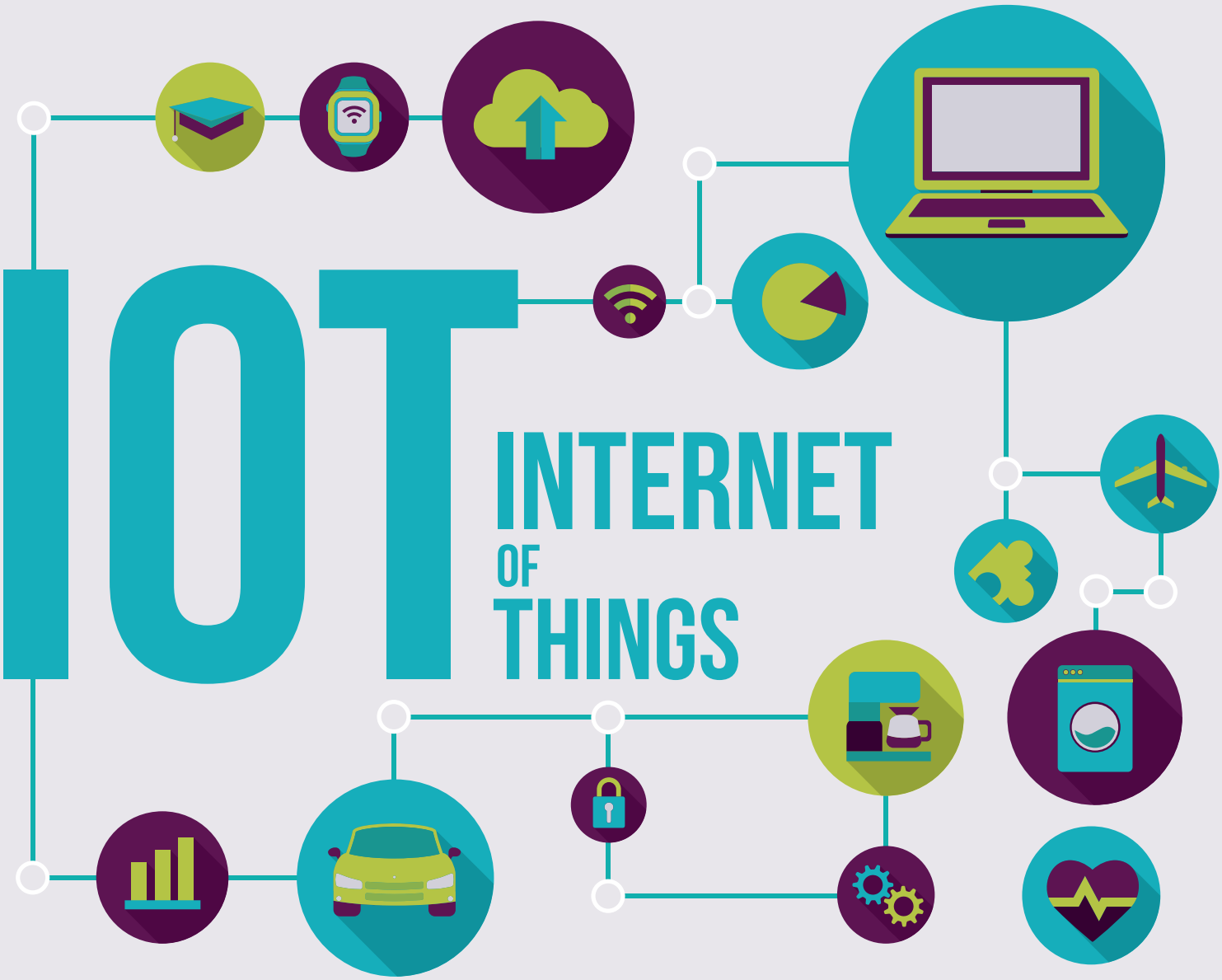
[b-SI] European Research Cluster on the Internet of Things, *IoT Semantic Interoperability: Research Challenges, Best Practices, Solutions and Next Steps, IERC AC4 Manifesto "Present and Future"*.
<http://www.probe-it.eu/wp-content/uploads/2013/10/IERC-AC4-SemanticInteroperabilityManifesto-V1_130830-Final1.pdf>

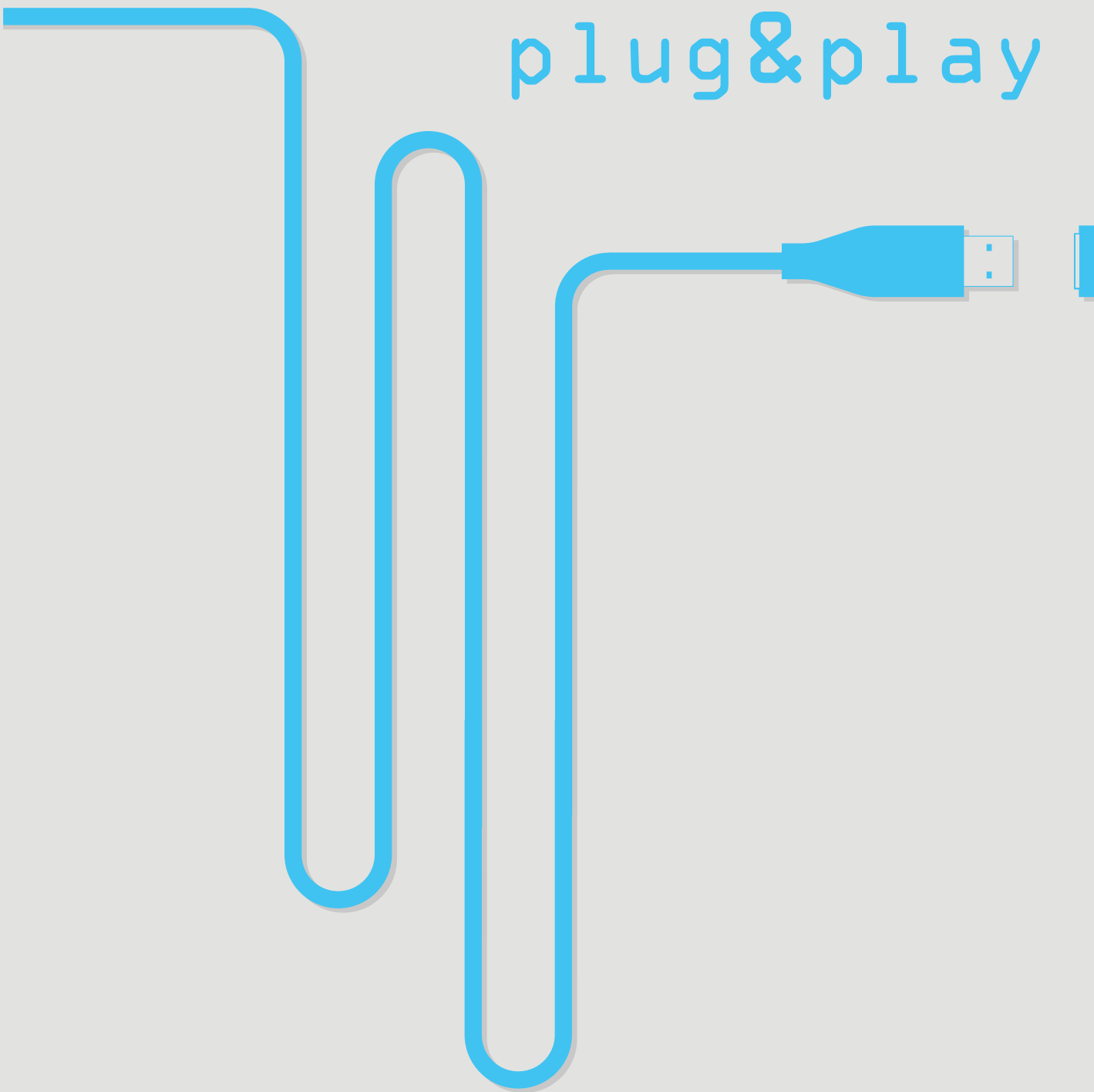
[b-SPARQL] K. Kyzirakos, M. Karpathiotakis, and M. Koubarakis (2010), *Developing Registries for the Semantic Sensor Web Using stRDF and stSPARQL*, Int'l Workshop Semantic Sensor Networks.
<http://people.csail.mit.edu/pcm/tempISWC/workshops/SSN2010/paper8.pdf>

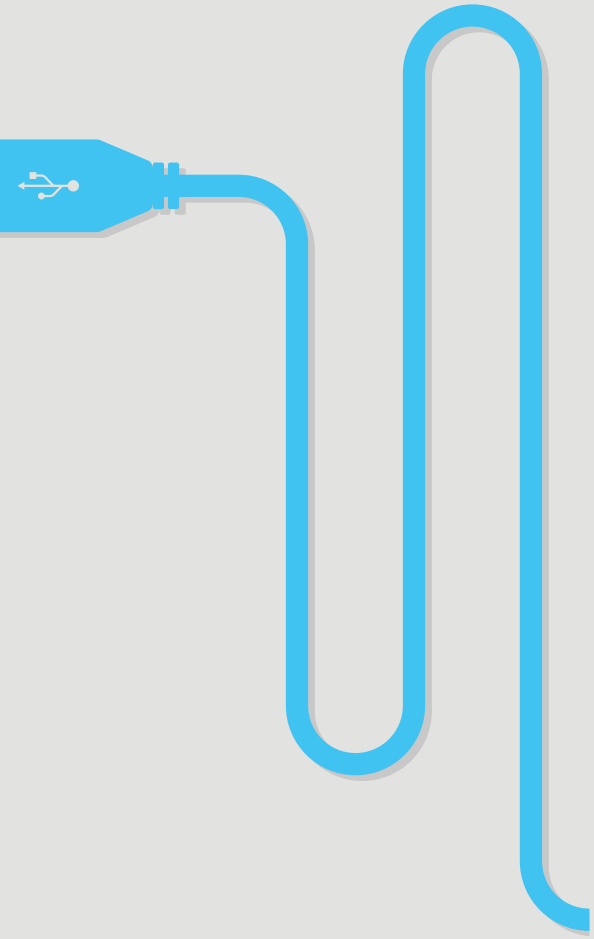
[b-TR069] Broadband Forum TR-069 Amendment 5 (2013), *CPE WAN Management Protocol*.
<http://www.broadband-forum.org/technical/download/TR-069_Amendment-5.pdf>

[b-UML] Object Management Group (OMG), (2014), *Unified Modeling Language (UML®) Resource Page*.
<http://www.uml.org/>

[b-AIOTI-WG3] AIOTI WG03 IoT Standardization (2015), *Semantic Interoperability*, Release 2.0.
<https://docbox.etsi.org/SmartM2M/Open/AIOTI/!!20151014Deliverables/AIOTI_WG3_SemanticInterop_Release_2_0a.pdf>


IOT
INTERNET
OF
THINGS


plug&play



Y.4112/Y.2077

Requirements of the Plug and Play capability of the Internet of Things

Requirements of the plug and play capability of the Internet of things

Summary

Recommendation ITU-T Y.4112/Y.2077 specifies the requirements of the plug and play capability of the Internet of things (IoT), as a basis for further standardization work related to the plug and play aspects in the IoT.

This Recommendation first describes the concept and the purpose of the plug and play capability of the IoT, and it then provides the components of this capability as well as its requirements.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T Y.4112/Y.2077	2016-02-13	13	11.1002/1000/12706

Keywords

Internet of things, plug and play, plug and play capability.

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

Table of Contents

Recommendation ITU-T Y.4112/Y.2077

Requirements of the plug and play capability of the Internet of things

1 Scope

This Recommendation specifies the requirements of the plug and play (PnP) capability of the Internet of things (IoT). More specifically, this Recommendation covers the following:

- concept and purpose of the PnP capability of the IoT
- components of the PnP capability of the IoT
- requirements of the PnP capability of the IoT.

Use cases of the PnP capability are provided in Appendix I.

This Recommendation can be seen as complementary to the common requirements of IoT identified in [ITU-T Y.4100].

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T Y.4000] Recommendation ITU-T Y.4000/Y.2060 (2012), *Overview of the Internet of things*.

[ITU-T Y.4100] Recommendation ITU-T Y.4100/Y.2066 (2014), *Common requirements of the Internet of things*.

[ITU-T Y.4101] Recommendation ITU-T Y.4101/Y.2067 (2014), *Common requirements and capabilities of a gateway for Internet of things applications*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 device [ITU-T Y.4000]: With regard to the Internet of things, a piece of equipment with the mandatory capabilities of communication and the optional capabilities of sensing, actuation, data capture, data storage and data processing.

3.1.2 Internet of things [ITU-T Y.4000]: A global infrastructure for the information society enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving, interoperable information and communication technologies.

NOTE 1 – Through the exploitation of identification, data capture, processing and communication capabilities, the IoT makes full use of things to offer services to all kinds of applications, whilst ensuring that security and privacy requirements are fulfilled.

NOTE 2 – In a broad perspective, the IoT can be perceived as a vision with technological and societal implications.

3.1.3 gateway [ITU-T Y.4101]: A unit in the Internet of things which interconnects the devices with the communication networks. It performs the necessary translation between the protocols used in the communication networks and those used by devices.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 plug and play (PnP) (capability): With regard to the IoT, a capability which enables automatic generation or acquisition of configurations for a device when it is connected to the communication network, in order for the device to satisfy the requirements of related IoT application(s).

NOTE – For the purpose of this Recommendation, the PnP capability can be considered as composed of the PnP management capability, PnP security capability, device PnP capability and gateway PnP capability.

3.2.2 PnP management capability: For the purpose of this Recommendation, this is the component of the PnP capability providing configuration management, fault management and activation/deactivation of PnP.

3.2.3 PnP security capability: For the purpose of this Recommendation, this is the component of the PnP capability providing PnP authorization and access control of both devices and applications, as well as the confidentiality and integrity protection of data generated by the PnP procedure.

3.2.4 device PnP capability: For the purpose of this Recommendation, this is the component of the PnP capability enabling a device to respond to PnP management capability requests for obtaining a device's properties.

3.2.5 gateway PnP capability: For the purpose of this Recommendation, this is the component of the PnP capability enabling a gateway to respond to PnP management capability requests on behalf of devices.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

IoT Internet of Things

PnP Plug and Play

XML Extensible Markup Language

5 Conventions

In this Recommendation:

– The keywords "is required to" indicate a requirement which must be strictly followed and from which no deviation is permitted if conformance to this document is to be claimed.

– The keywords "is recommended" indicate a requirement which is recommended but which is not absolutely required. Thus this requirement need not be present to claim conformance.

– The keywords "can optionally" and "may" indicate an optional requirement which is permissible, without implying any sense of being recommended. These terms are not intended to imply that the vendor's implementation must provide the option and the feature can be optionally enabled by the network operator/service provider. Rather, it means the vendor may optionally provide the feature and still claim conformance with the specification.

6 Overview of the plug and play capability

6.1 Introduction

IoT is defined as a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies [ITU-T Y.4000]. Among the fundamental characteristics of the IoT, the plug and play (PnP) capability is recommended in order to enable fast generation, composition or the acquisition of configurations for seamless integration and cooperation of interconnected devices with applications, and for a responsiveness to application requirements [ITU-T Y.4000].

NOTE – The PnP capability is not mandatory to support IoT applications. For example, some IoT applications have extra requirements for devices or need a highly secure operating environment; under such circumstances, the PnP capability might be disabled. Additionally, the service provider and/or user may have the permission to activate/deactivate the PnP capability.

The PnP capability of the IoT is responsible for triggering the configuration procedure automatically as soon as a device is connected to the network, without impacting security and privacy.

This Recommendation describes requirements for the PnP capability as a framework to enable functionalities such as:

– PnP capability discovery

– automatic generation of device configuration

– automatic fault recovery of the PnP procedure

– PnP security protection.

6.2 The components of the plug and play capability

For the purpose of this Recommendation, the PnP capability can be considered as composed of the PnP management capability, PnP security capability, device PnP capability and gateway PnP capability.

Figure 1 shows the IoT reference model [ITU-T Y.4000] with the positioning of the different PnP capability components.

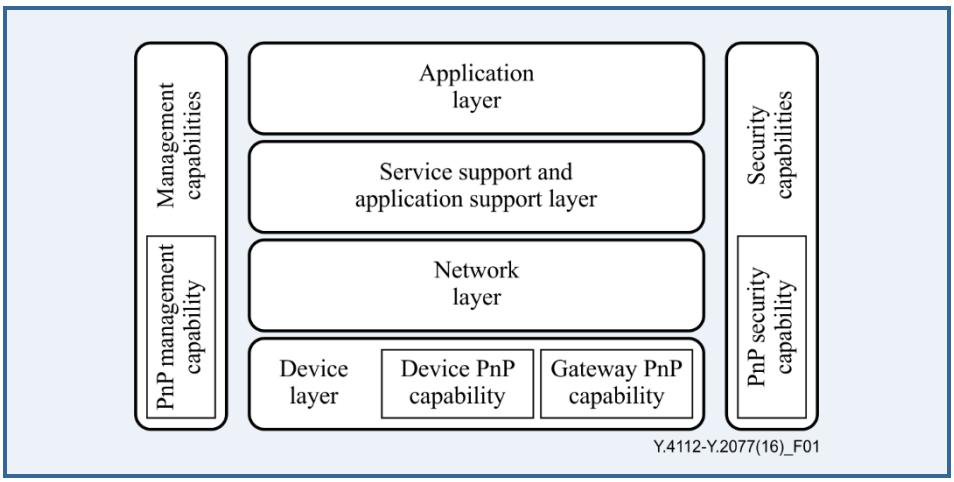


Figure 1 – The IoT reference model with the components of the PnP capability

6.2.1 PnP management capability

The PnP management capability belongs to the management capabilities of the IoT reference model [ITU-T Y.4000] and is the core part of the PnP capability. The PnP management capability covers configuration management, fault management and activation/deactivation of the PnP procedure.

The PnP management capability does not increase the freedom of the device configuration; on the contrary it increases the level of device configuration automation by restricting the configuration procedure. When a device connects to the network and is discovered by the PnP management capability, the PnP management capability tries to obtain the properties of the device, including manufacturer, model, application level protocol, memory space, average response time, etc. The configuration file is generated by the PnP management capability by taking into consideration the device properties and the requirements of related IoT application(s). As the application(s) can interact with the device based on the configuration file, the device can work automatically.

In all scenarios when it is required or recommended to deactivate the PnP procedure, the PnP management capability takes charge of its deactivation.

If any error happens during the configuration procedure or the device cannot fulfil the requirements of the IoT application after the configuration file is generated, the PnP management capability acts according to the specific policies, e.g., it repeats the configuration procedure or produces an error notification to guide an external intervention.

6.2.2 PnP security capability

After a device has connected to the network, mutual authentication and authorization between the device and IoT are required [ITU-T Y.4100]. This is the task of the IoT basic security capability.

After a device has connected to the network, the configuration procedure of the device is executed. Manual configuration is considered as a safe mechanism, which does not need extra security protection. However, if the IoT has PnP capability and the configuration procedure is automatic, some basic security functionalities may be skipped and this may increase vulnerability to network attacks. The PnP security capability is necessary in this situation.

The PnP security capability is part of the security capabilities of the IoT and, with respect to the IoT reference model [ITU-T Y.4000], includes:

– at the device layer, PnP authorization and access control of the device, device data confidentiality and integrity protection;

– at the service support and application support layer, PnP authorization and access control of the application and application data protection.

6.2.3 Device PnP capability

There are many different kinds of devices in the IoT. If a device cannot support the PnP capability, e.g., a basic sensor directly accessing the network, the PnP procedure does not work even if the network supports the PnP capability. In order to start the PnP procedure, the device has to be capable of responding to the PnP management capability's requests.

The device PnP capability refers to the ability of the device to respond to PnP requests with the device's properties.

As described in clause 6.2.4, a gateway with gateway PnP capability can enable the PnP procedure for devices connected through it to the network which do not support device PnP capability.

6.2.4 Gateway PnP capability

Some devices may connect to the network through a gateway. A gateway is required to support the management of device related information, e.g., device identification, device configuration, etc. [ITU-T Y.4101].

If the devices connected to the network through a gateway have no device PnP capability, the gateway can respond to PnP management capability requests with devices' properties on behalf of them.

7 Requirements of the PnP capability

In addition to the IoT common requirements [ITU-T Y.4100], which constitute the basic support for the PnP capability, the following subclauses describe the specific requirements of the PnP capability.

7.1 PnP management capability related requirements

7.1.1 PnP discovery

In addition to the common requirements of IoT for discovery services [ITU-T Y.4100], PnP discovery is necessary for the support of the PnP management capability. This functionality is used to identify whether a device or a gateway has PnP capability. Without identifying this, potential network problems may arise after the device is connected to the network.

The following are the PnP discovery related requirements:

– The IoT is required to support PnP discovery.

7.1.2 PnP configuration management

PnP configuration management is responsible for generating the device configuration. It first sends a request to the device or gateway to obtain the properties of the device, a configuration is then generated by taking into consideration the device properties and IoT application's requirements.

The following are the PnP configuration management related requirements:

– The IoT is required to have the capability of sending PnP configuration requests to devices or gateways.

– The IoT is required to have the capability of processing the device properties replied by devices or gateways.

– The IoT is required to have the capability of disabling unnecessary device capabilities according to the IoT application's requirements.

– The IoT is required to have the capability of generating device configurations built using predefined syntax and semantics.

– The IoT is required to support the manual modification of configurations generated by the PnP procedure.

– The IoT is required to store device configurations, e.g., for possible future usage by the same application and same device.

– The IoT is required to have the capability to store different configurations for the same device, e.g., according to the associated application.

7.1.3 PnP fault management

If any error happens during the configuration procedure or the device cannot fulfil the requirements of the IoT application after the configuration file is generated, the PnP fault management will operate according to the specific policies, e.g., it will repeat the configuration procedure or produce an error notification to guide an external intervention.

The following are the PnP fault management related requirements:

– The IoT is required to recognize, isolate and correct faults that occur during the PnP procedure.

– The IoT is required to have the capability of restarting the PnP procedure.

– The IoT is required to have the capability of interrupting or terminating the PnP procedure.

– The IoT is required to have the capability of setting the duration time of the PnP procedure.

– The IoT is required to log the PnP related activities.

7.1.4 PnP activation/deactivation

As described in clause 6.1, under certain circumstances, the PnP capability might be deactivated.

The following are the PnP activation/deactivation related requirements:

– The IoT is required to support the capability of activating/deactivating the PnP capability.

7.2 PnP security capability related requirements

Mutual authentication between a device and IoT makes sure that it is impossible for a third party to masquerade as a device by spoofing its identity. Mutual authentication between an application and IoT makes sure that it is impossible for a third party to masquerade as an application by spoofing its identity [ITU-T Y.4100]. However, authenticated entities still need to be authorized for any PnP related operation.

The PnP procedure, including the generated configurations, is controlled by access rules and protected by firewall capabilities.

7.2.1 PnP authorization

The following are the PnP authorization related requirements:

– Any PnP related operation on the device is required to be authorized.

– An application is required to be authorized to perform any PnP related operation.

– If the PnP procedure involves user information, or the user needs to manually modify the PnP configuration, the user is required to be authorized.

– If a device without PnP capability connects to the IoT through a gateway, the gateway is required to be authorized to perform any PnP related operation on the device.

7.2.2 PnP access control

The following are the PnP access control related requirements:

– The IoT is required to only run the PnP procedure if the necessary device configuration does not exist, e.g., when the device connects to the network for the first time or a new application wants to access the connected device.

7.2.3 Firewall protection

The following are the firewall protection related requirements:

– The IoT is recommended to support firewall protection between devices and the IoT infrastructure (including gateways).

– Firewall protection is required to have the capability to stop the PnP procedure if the properties provided by the device are suspicious (e.g., the data transmission redundancy of a device is too high).

– Firewall protection is required to have the capability to stop the PnP procedure if the request made by the application is suspicious (e.g., the data upload frequency required by the application is too high).

– Firewall protection is required to have the capability to protect the PnP procedure from any illegal access or attack.

7.2.4 Device and application data security

The following are the device and application data security related requirements:

– The IoT is required to provide integrity protection to all data generated during the PnP procedure.

– The IoT is required to provide confidentiality protection to all data generated during the PnP procedure.

– The IoT is required to store the device configurations generated by the PnP procedure in a safe manner.

7.3 Device PnP capability related requirements

The following requirements are related to devices which support the PnP capability.

The following are the device PnP capability related requirements:

– The device is required to support PnP discovery triggered by the IoT.

– The device is required to be able to communicate to the IoT, on demand, all its relevant properties.

– The device is required to have the capability of refusing PnP configuration requests from the IoT, e.g., the device provides sensitive data but the current network access connectivity is unsecure.

7.4 Gateway PnP capability related requirements

The following requirements are related to gateways which support the PnP capability.

NOTE – In case of deployment scenarios where both gateway and devices which are connected through it to the network support the PnP capability, the gateway is transparent to the PnP procedure.

– The gateway is required to support PnP discovery triggered by the IoT.

– The gateway is required to be able to communicate to the IoT, on demand, all properties of the connected devices.

– The gateway is required to have the capability of refusing PnP configuration requests from the IoT.

Appendix I

Use cases of the PnP capability

(This appendix does not form an integral part of this Recommendation.)

I.1 Large scale sensor deployment: greenhouse example

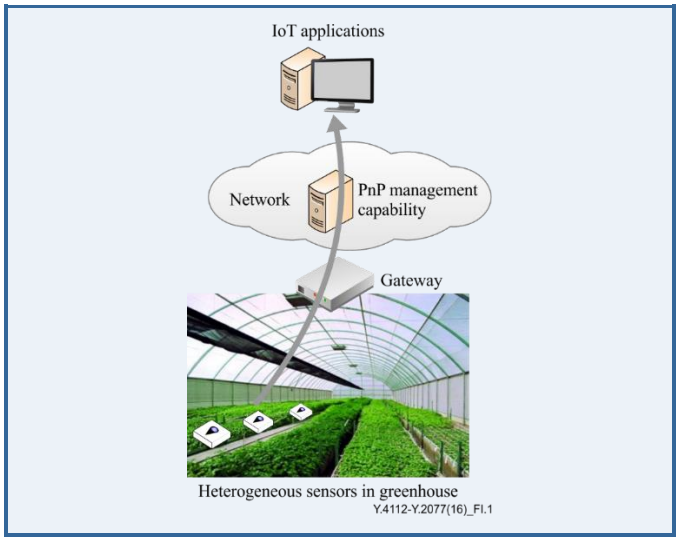


Figure I.1 – Use case of PnP capability in greenhouse sensor deployment

Deploying environmental sensors in a greenhouse is a typical scenario of smart agriculture applications. There are a large number and different types of sensors used in greenhouses, which can collect data such as temperature, humidity, illumination, CO_2 etc. Users who subscribe to this greenhouse monitoring service are normally farmers who lack network and communication knowledge. Following deployment, sensors need to work for a long time without any manual maintenance.

The PnP management capability is important in this situation. Figure I.1 shows a use case of PnP capability in greenhouse sensor deployment. All the sensors only need to be deployed in the appropriate position, without any manual configuration. The PnP management capability will automatically generate or acquire the corresponding configuration for every kind of sensor, according to their properties and related IoT application. All sensors will work automatically in the end.

If a sensor is broken, the user only needs to change it with an identical one (or updated version), and the new sensor will work in the same way as before.

I.2 Security protection from counterfeit device

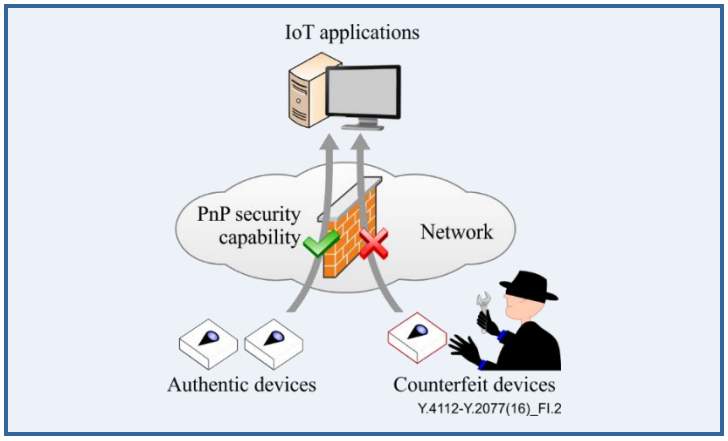


Figure I.2 – Use case of PnP security capability for protection from counterfeit devices

Unlike mobile phones, a large number of IoT devices will be placed in locations with difficult accessibility, without any manual maintenance. In this situation, the PnP management capability will reduce the maintenance issue. However, this situation also provides hackers with an opportunity to hack into the IoT from the device side. If a hacker connects a counterfeit device to the IoT supporting the PnP capability, the device configuration procedure will run automatically to make the device work. As a result, the counterfeit device will provide misleading data to IoT application(s), or block the transmission channel using redundant data.

The PnP security capability is important in this situation. Figure I.2 shows a use case of PnP security protection from counterfeit devices. The counterfeit devices will pass the authentication procedure of IoT by simulating the authentic devices. However, the configuration parameters provided by the counterfeit devices will differ from those of the authentic ones. The PnP security capability will detect the potential risk and terminate the configuration procedure.

I.3 Enablement of customized configuration of IoT device

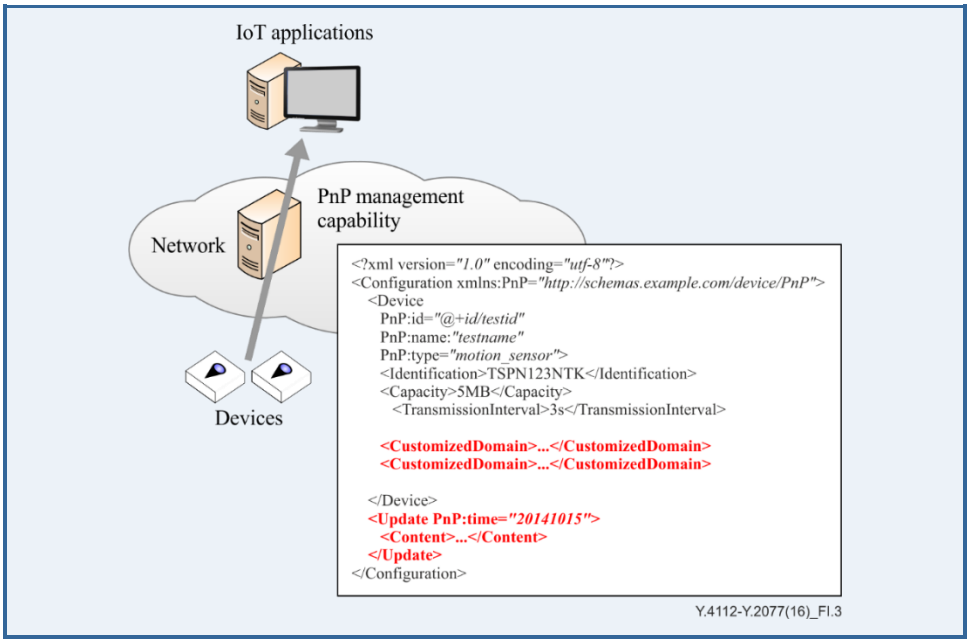
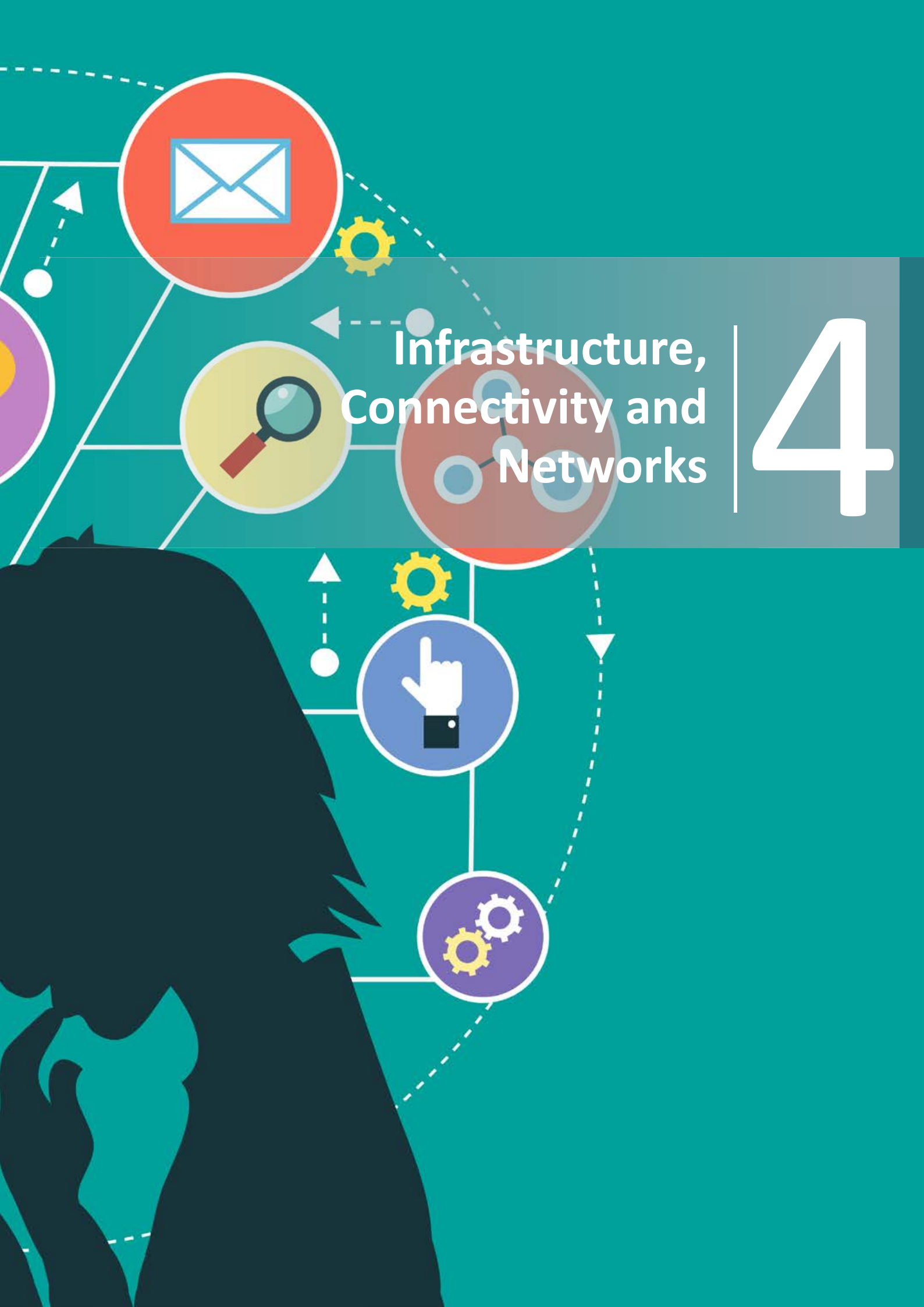


Figure I.3 – Use case of customized configuration of IoT device

There are many different kinds of devices in IoT, and the requirements of IoT applications are diverse. Unless highly customized for a certain kind of IoT application, it is possible that a device does not work immediately after connecting to the network. The most important mission of the PnP management capability is to automatically generate the configuration for the device. However, the automatic generation of the configuration does not mean that a user cannot change the configuration. Figure I.3 shows a use case of customized configuration enablement of an IoT device. The user can interrupt the PnP procedure by updating the configuration file manually, and the device works according to the new configuration. The new configuration is recorded and re-generated automatically for the same device and IoT application.



Infrastructure, Connectivity and Networks

4


SCANNING
57%
32452345
5466572345
3452345
234562346
13452345

Y.4250/Y.2222

Sensor control networks and related applications in a next generation network environment

Sensor control networks and related applications in a next generation network environment

Summary

Recommendation ITU-T Y.2222 provides an introduction to sensor control networks (SCNs) and related applications in a next generation network (NGN) environment. More specifically, it provides an overview of SCNs, configurations for SCN applications and service requirements of SCN applications for support in a NGN environment.

History

Edition	Recommendation	Approval	Study Group
1.0	ITU-T Y.2222	2013-04-13	13

Keywords

Actuator, configurations for SCN applications, decision-making process, emergency management, gate, mote, NGN, SCN, SCN applications, SCN controller, sensor control networks, sensor networks, verification application.

Table of Contents

Recommendation ITU-T T.4250/Y.2222

Sensor control networks and related applications in a next generation network environment

1 Scope

This Recommendation provides an introduction to sensor control networks (SCNs) and related applications in a next generation network (NGN) environment. More specifically, this Recommendation provides:

– definitions of SCN and SCN related terms;

– overview of SCNs;

– description of possible configurations for SCN applications and the decision-making process for these configurations;

– service requirements of SCN applications for support in NGN environment.

Moreover, two important use cases of SCNs are described in the appendices: SCNs for verification and SCNs for emergency management.

Business models and charging issues are outside of the scope of this Recommendation.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T Y.2020] Recommendation ITU-T Y.2020 (2011), *Open service environment functional architecture for next generation networks*.

[ITU-T Y.2061] Recommendation ITU-T Y.2061 (2012), *Requirements for the support of machine-oriented communication applications in the next generation network environment*.

[ITU-T Y.2234] Recommendation ITU-T Y.2234 (2008), *Open service environment capabilities for NGN*.

[ITU-T Y.2240] Recommendation ITU-T Y.2240 (2011), *Requirements and capabilities for next generation network service integration and delivery environment*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 actuator [ITU-T Y.2061]: A device performing physical actions caused by an input signal.

NOTE 1 – As examples, an actuator might act on the flow of a gas or liquid, on electricity distribution, or through a mechanical operation. Dimmers and relays are examples of actuators. The decision to activate the actuator may come from an MOC application, a human or MOC devices and gateways.

NOTE 2 (added by ITU-T Y.2222) – There are three types of actuators: information actuators, which are intended to provide visual, audio, sensory interaction with the human user; gateway actuators, which are intended to forward control commands given by SCN to other networks; machine actuators, which are electromechanical devices intended for physical interaction with the external environment.

3.1.2 application [b-ITU-T Y.101]: A structured set of capabilities, which provide value-added functionality supported by one or more services.

3.1.3 context awareness [b-ITU-T Y.2201]: A capability to determine or influence a next action in telecommunication or process by referring to the status of relevant entities, which form a coherent environment as a context.

3.1.4 machine oriented communication (MOC) [ITU-T Y.2061]: A form of data communication between two or more entities in which at least one entity does not necessarily require human interaction or intervention in the communication process.

3.1.5 next generation network (NGN) [b-ITU-T Y.2001]: A packet-based network able to provide telecommunication1 services and able to make use of multiple broadband, QoS-enabled transport technologies and in which service-related functions are independent from underlying transport-related technologies. It enables unfettered access for users to networks and to competing service providers and/or services of their choice. It supports generalized mobility which will allow consistent and ubiquitous provision of services to users.

3.1.6 NGN service integration and delivery environment (NGN-SIDE) [ITU-T Y.2240]: An open environment in NGN integrating resources from different domains and delivering integrated services to applications over NGN.

NOTE – These domains include, but are not limited to, telecommunication domain (e.g., fixed and mobile networks), Internet domain, broadcasting domain and content provider domain.

3.1.7 nomadism [b-ITU-T Q.1706]: The ability of the user to change their network access point on moving. When changing the network access point, the user's service session is completely stopped and then started again, i.e., there is no service continuity or hand-over used. It is assumed that normal usage pattern is that users shut down their service session before attaching to a different access point.

3.1.8 open service environment capabilities [ITU-T Y.2234]: Capabilities provided by open service environment to enable enhanced and flexible service creation and provisioning based on the use of standards interfaces.

3.1.9 seamless handover [b-ITU-T Q.1706]: It is a special case of mobility with service continuity since it preserves the ability to provide services without any impact on their service level agreements to a moving object during and after movement.

3.1.10 sensed data [b-ITU-T F.744]: Data sensed by a sensor that is attached to a specific sensor node.

3.1.11 sensor [b-ITU-T Y.2221]: An electronic device that senses a physical condition or chemical compound and delivers an electronic signal proportional to the observed characteristic.

3.1.12 service [b-ITU-T Y.101]: A structure set of capabilities intended to support applications.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 aggregate value: A value calculated by combining the sensed data of several spatially distributed motes, reference values and other data and intended to represent the environmental conditions of a given geographical area.

3.2.2 central communication channel: A communication channel or a set of communication channels for transmitting data between SCN controllers and actuators without utilizing motes.

3.2.3 gate: Intermediate device that is used to provide a communication channel between an actuator and one or several nearby motes.

3.2.4 mote: A miniature computing device equipped with sensors and signal transceivers operating in a given radio band, and used for transmitting sensed data.

NOTE 1 – It is a kind of sensor node and it is characterized by a mandatory minimal level of computing resources.

NOTE 2 – Depending on application, a mote has one or more of the following capabilities: data manipulation, intelligent commutation and connectivity with actuators, retrieving data on the read-outs of near-site sensors.

3.2.5 mote group: A set of motes connected to each other without making usage of external networks.

3.2.6 non-SCN-enabled actuator: An actuator that is not able to communicate with motes directly, and uses networks to communicate with motes.

NOTE – In this Recommendation, a non-SCN-enabled actuator uses NGN to communicate with motes.

3.2.7 reference value: A value calculated by combining the sensed data of one or several closely situated motes and intended to represent the environmental conditions of some specific location.

3.2.8 sensor control network (SCN): A sensor network consisting of motes which is intended for controlling one or more actuators.

3.2.9 SCN application: An application that uses SCNs for controlling actuators.

3.2.10 SCN controller: A hardware/software system designed to collect and process data from motes and to transmit to actuators the necessary information for their control.

3.2.11 SCN-enabled actuator: An actuator that is able to communicate with motes directly.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

3G	Third Generation
GPRS	General Packet Radio Service
MOC	Machine Oriented Communication
NGN	Next Generation Network
NGN-SIDE	NGN Service Integration and Delivery Environment
OSE	Open Service Environment
PDA	Personal Digital Assistant
QoS	Quality of Service
SCN	Sensor Control Network
Wi-Fi	Wireless Fidelity
WiMAX	Worldwide Interoperability for Microwave Access

5 Conventions

In this Recommendation:

The keywords "is required to" indicate a requirement which must be strictly followed and from which no deviation is permitted if conformance to this document is to be claimed.

The keywords "is recommended" indicate a requirement which is recommended but which is not absolutely required. Thus this requirement need not be present to claim conformance.

The keywords "can optionally" and "may" indicate an optional requirement which is permissible, without implying any sense of being recommended. These terms are not intended to imply that the vendor's implementation must provide the option and the feature can be optionally enabled by the network operator/service provider. Rather, it means the vendor may optionally provide the feature and still claim conformance with the specification.

6 Overview of SCNs

Sensor networks are becoming more and more utilized and are under active development for different applications. These networks are frequently deployed with some control: sensor control networks (SCNs) are used to allow control by applications of the "actuators" which can be found in many sensor network deployments. The control is realized on a real-time basis and depending on environmental parameters.

In SCNs, for the measurement of environmental parameters, motes are deployed and they can be seen as an evolutive version of sensors. In comparison with a sensor in a sensor network, a mote, besides physical conditions monitoring, can also have capabilities of data manipulation, intelligent commutation and connectivity to actuators. Motes are connected to NGN, by the use of a gateway or directly, and may also act as an access network to the SCN applications for actuators.

NOTE 1 – The details of sensors and actuators are outside of the scope of this Recommendation.

The goal of any application using SCNs ("SCN application") is to run a decision-making process and finally provide all involved actuators with relevant control commands. This process includes various activities of data acquisition, data transmission and data manipulation. A range of configurations (see details in clause 7) may be employed for SCN application's decision-making process depending on the capabilities of actuators and motes. For example, an actuator may only use commands provided by the SCN application to itself, or may customize them in order to meet its own capabilities and user requirements or may perform decision making completely by itself based on the data provided by the SCN application.

Application field examples of SCNs include:

– everyday life: navigation, excursions, sports;
– medicine: body control, e-health;
– enterprise: logistics, stock management;
– industrial field: fabrication automation, production process control;
– military field: combat assistance, remote piloting;
– emergency and disaster management: early warning, evacuation, emergency orchestration of civilians and rescuers, automatic danger elimination.

The decisions of SCN applications may be targeted for both human users and control of machines (e.g., gears, vehicles, robots). The latter case also includes machine oriented communication (MOC) devices [ITU-T Y.2061] implying communications between two or more entities in which at least one entity does not necessarily require human interaction or intervention in the process of communication.

Since decisions of SCN applications directly involve the operations of actuators, tight integration between SCN infrastructure and actuators' capabilities must be achieved. Although the following matter is outside of the scope of this Recommendation, it is expected that standardization of the interfaces between SCNs and actuators will be critical.

In addition, although the following matter is also outside of the scope of this Recommendation and for further study, it is anticipated that some enhancements to the NGN capabilities [b-ITU-T Y.2201] will be required in order to support the service requirements of SCN applications identified in this Recommendation.

Figure 6-1 shows an overview of SCNs, including SCN applications and the supporting role of NGN.

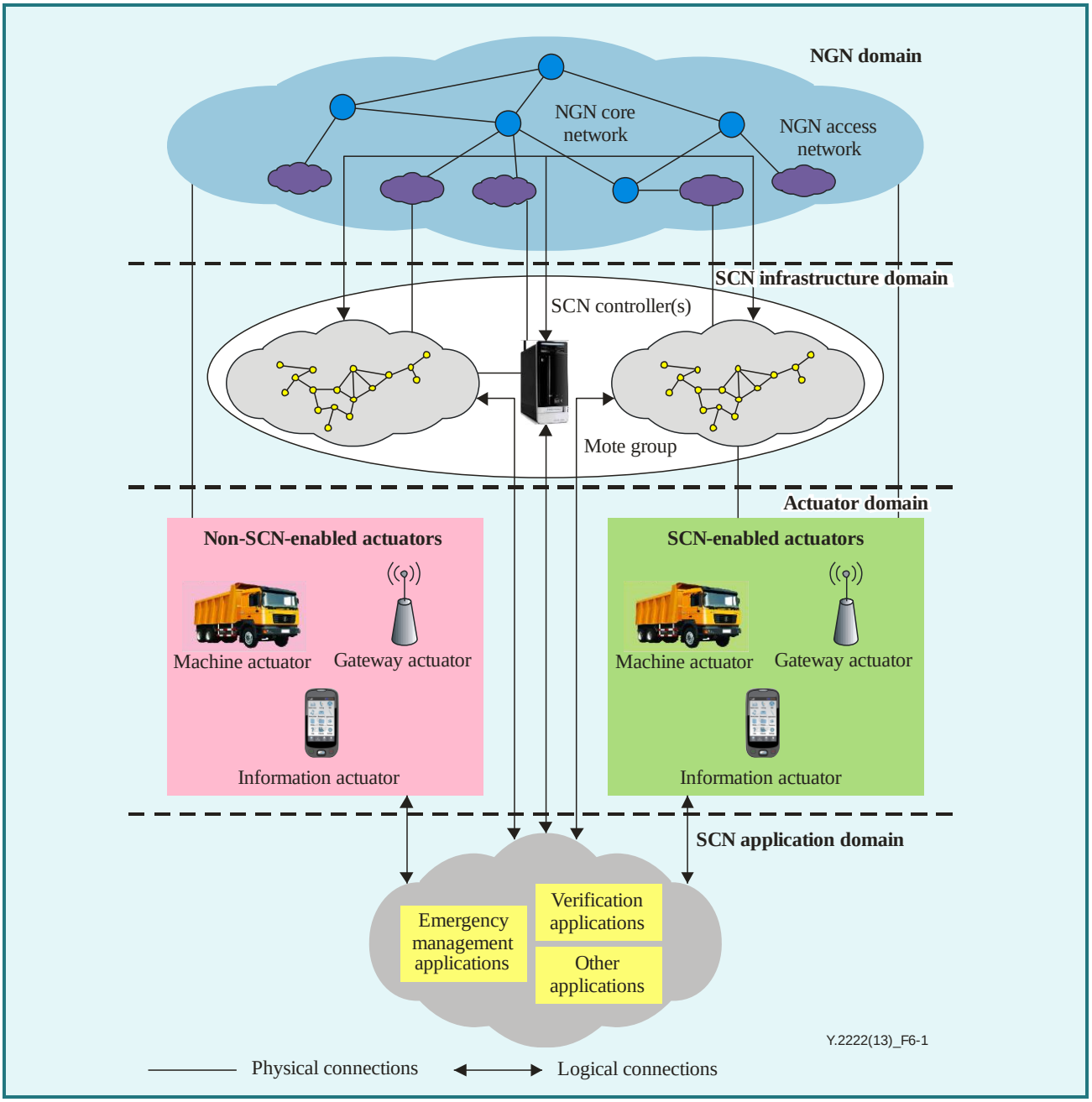


Figure 6-1 – Overview of SCNs

Figure 6-1 depicts four domains:

1) NGN domain: the connectivity via NGN fulfils two objectives. First, NGN provides access to SCN applications for both non-SCN-enabled and SCN-enabled actuators when direct communication of actuators with motes is not possible or desirable (e.g., when an actuator is a mobile phone and its owner does not want his/her location to be exposed due to privacy reasons). Second, NGN is used to unite spatially distributed mote groups and the SCN controllers into a single network.

   NOTE 2 – NGN is expected to provide the same capabilities (e.g., load balancing, fault tolerance) as a single mote group directly connected to the SCN controllers.

2) SCN infrastructure domain: the SCN infrastructure includes one or several SCN controllers and mote groups. They may be spatially distributed: in that case, NGN is used to unite them into a single network. Authorized personnel may use the SCN controllers for SCN monitoring and administration. Motes can allow direct access to SCN applications of SCN-enabled actuators, while direct access via motes to SCN applications of non-SCN enabled actuators is not possible. The SCN controllers are connected to NGN directly.

3) Actuator domain: the actuators can be of three different types: machine actuators (e.g., car, water sprinkler, door lock), information actuators (e.g., screen, loudspeaker, mobile phone, PDA, notebook) and gateway actuators (e.g., computer with telephone private branch exchange software).

   NOTE 3 – The term "SCN objects" is used in the following clauses to cover motes and SCN controllers (which constitute the SCN infrastructure), as well as actuators.

4) SCN application domain: consists of SCN applications, e.g., verification applications (see Appendix I), emergency management applications (see Appendix II) and others.

   NOTE 4 – Different parts of SCN applications can reside in different SCN objects according to the specific application requirements.

7 Configurations for SCN applications

7.1 Basic operations for SCN applications

Four types of operations are identified for SCN applications:

1) Fetching of sensed data (shown as Sensed data in Figures 7-1 to 7-5 and I.1).

2) Calculation of reference values by combining (e.g., averaging) the sensed data of one or several closely situated motes (shown as Reference values in Figures 7-1 to 7-5 and I.1). The aim of this process can be for example:
   - comparison of sensed data readings with thresholds for the purpose of filtering sensed data and taking them into account during calculations of aggregate values and/or decision making,
   - auxiliary pre-calculations for the purpose of quicker calculation of aggregate values and/or decision making,
   - synchronous analysis of multiple sensed data readings.

3) Calculation of aggregate values by combining (e.g., averaging) the sensed data of several spatially distributed motes, reference values and other data (shown as Aggregate values in Figures 7-1 to 7-5 and I.1).

4) Decision making (shown as Decision making in Figures 7-1 to 7-5 and Figure I.1). During this process a specific control command for the actuator is formed. It can use fetched aggregate values.

In SCNs, data can be transmitted via the SCN infrastructure (i.e., using motes and SCN controllers as intermediate nodes) and via the central communication channel (e.g., using GPRS/3G, Wi-Fi and WiMAX technologies).

The above-described operations and the associated data transmissions can be represented in a flow chart. The rows of such a chart represent the above listed operations and the columns represent elements participating in the decision-making process. Data transmission flows are depicted as horizontal arrows whose endings correspond to the sending and receiving elements of the actual transmission stage, while data computational flows are depicted as vertical arrows corresponding to the above described operations. Figure 7-1 shows an example of such a flow chart.

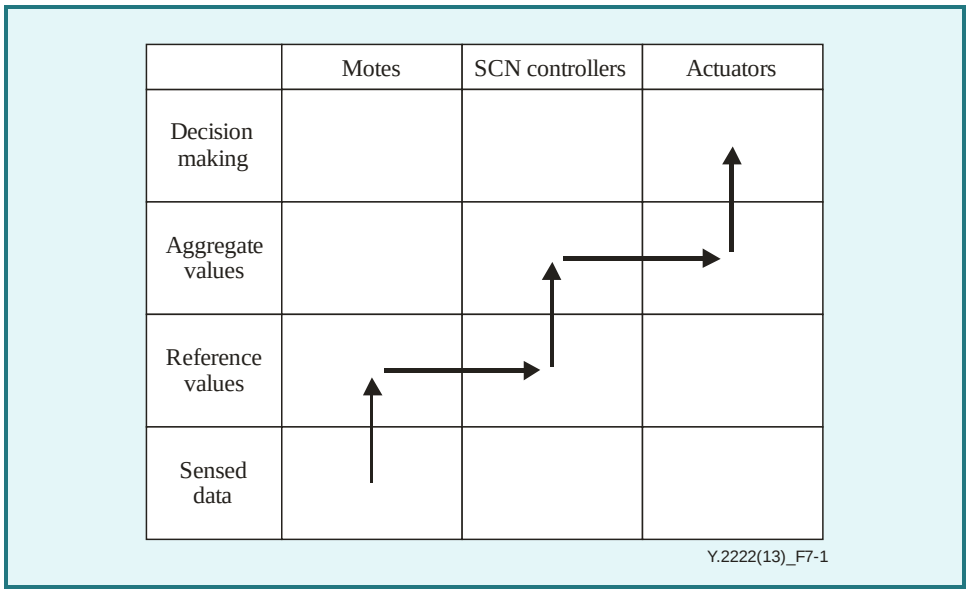


Figure 7-1 – Flow chart example of basic operations for a SCN application

7.2 Decentralized configuration for SCN applications

7.2.1 Introduction to decentralized configuration for SCN applications

The decentralized configuration is the most universal configuration for SCN applications in terms of flexibility, expansibility and reliability. It is so called because it makes minimal demand to the central communication channel and the SCN controllers. This provides the possibility of ubiquitous usage of such configurations in a wide range of applications, including emergency management applications (due to the high risk of failure related to centralized entities in case of disaster or emergency).

7.2.2 Role distribution in decentralized configuration for SCN applications

- **SCN controller:**

– It receives from the actuators via the central communication channel requests about aggregate values, which are necessary for making decision but cannot be calculated by the actuators themselves.

– It requests transmission of sensed data and reference values from the appropriate motes via the SCN infrastructure and regularly calculates the necessary aggregate values.

– It transmits to each actuator via the central communication channel the aggregate values requested by that actuator.

– It interoperates with external systems (e.g., a different application server) and the authorized personnel administrating the SCN.

- **Actuator:**

– It requests the necessary sensed data and reference values from the motes via the SCN infrastructure.

– It requests from the SCN controllers via the central communication channel the aggregate values which are necessary for decision making but cannot be calculated by the actuator itself.

– It receives from the motes via the SCN infrastructure the requested sensed data and reference values and calculates the other necessary reference values and aggregate values.

– It receives from the SCN controllers via the central communication channel the requested aggregate values.

– It forms the appropriate control commands.

– It transmits to the SCN controllers information about its own status via the central communication channel.

- **Mote:**

– It receives requests from the SCN controllers and the actuators via the SCN infrastructure about sensed data or reference values.

– It transmits the requested data to the SCN controllers and the actuators via the SCN infrastructure.

7.2.3 Decision-making process

The decision-making process follows the following procedure:

1) The necessary sensed data, reference values and aggregate values are kept in the SCN controllers' memory and regularly updated.

2) Each actuator sends requests for sensed data and reference values to the motes, and then stores the received ones in memory. The data requests can be of different types, such as broadcast requests (all motes send data on demand to actuators via the SCN infrastructure), or threshold-exceeding requests (only motes whose sensed data exceed some thresholds send data), etc.

3) Some other reference values can be computed as needed by the actuators based on received sensed data and reference values.

4) Each actuator needs to have the up-to-date aggregate values necessary to make a decision. These aggregate values can be computed by the actuator itself or fetched from the SCN controllers.

5) Each actuator forms a control command depending on the aggregate values.

Two examples of flow charts for decentralized configuration are shown in Figures 7-2 and 7-3.

In the first example, actuators use aggregate values received from the SCN controllers (data flow 2) and aggregate values calculated using reference values received from motes (data flow 1).

In the second example, actuators use only aggregate values calculated using reference values received from motes (data flow 1). There is no influence of the SCN controllers on the decision-making process. The SCN controllers only calculate (data flow 2) and store in memory aggregate values for the purpose of interoperation with external systems and the authorized personnel administrating the SCN.

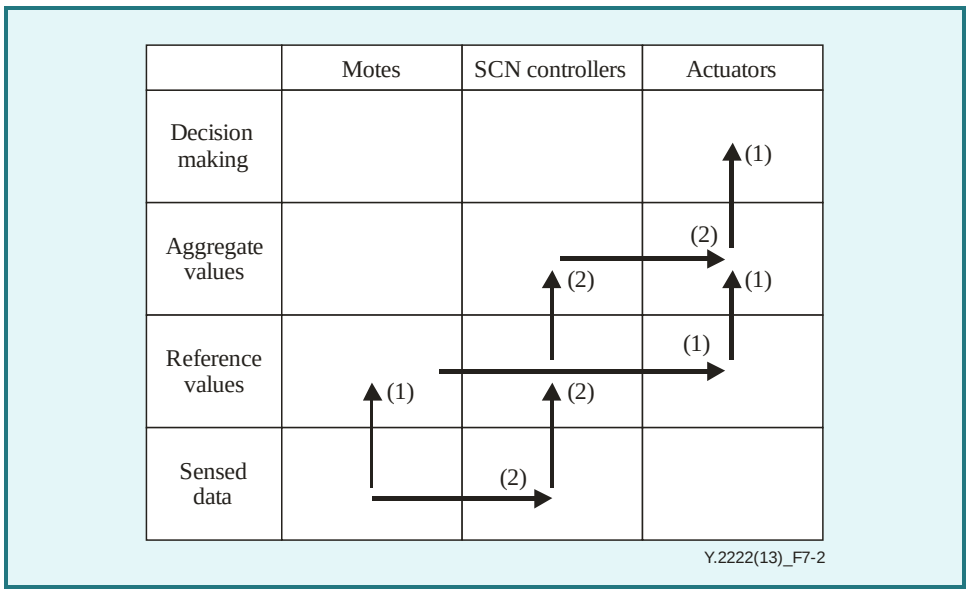


Figure 7-2 – A first example of a flow chart for decentralized configuration for SCN applications

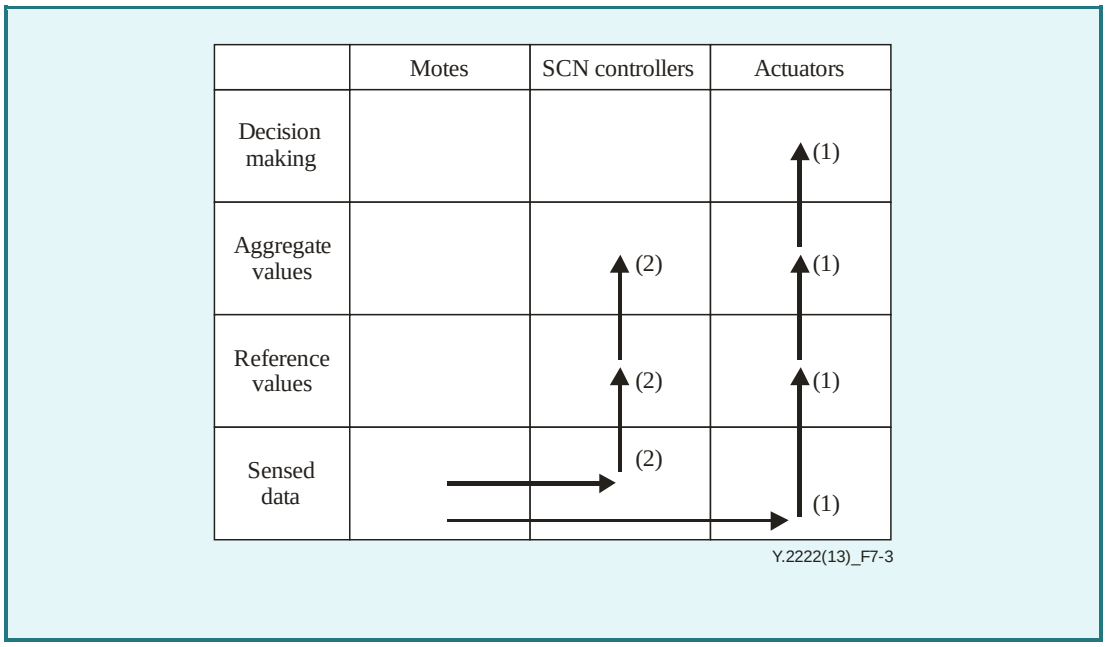


Figure 7-3 – A second example of a flow chart for decentralized configuration for SCN applications

7.3 Transitional configurations for SCN applications

Nowadays most mass mobile user terminals, such as mobile phones, PDAs and netbooks, have no technical possibility of direct data exchange with existing infrastructures of motes because of differences in transceiver types and transmission standards. Therefore, transitional configurations are needed to provide a possibility of using SCNs with mass mobile user terminals.

7.3.1 Centralized configuration for SCN applications

7.3.1.1 Introduction to centralized configuration for SCN applications

This configuration is so called because the data for every decision made by SCN are transferred through the SCN controllers and are delivered to the actuators via a central communication channel. It should be employed when actuators can only communicate via the central communication channel and/or it is not desirable to change the existing infrastructure of motes and actuators to enable SCN applications.

7.3.1.2 Role distribution in a centralized configuration for SCN applications

- **SCN controller:**

– It receives from the actuators requests via the central communication channel about aggregate values.

– It requests transmission of sensed data and reference values from the appropriate motes via the SCN infrastructure and regularly calculates the necessary aggregate values.

– It transmits to each actuator via the central communication channel the aggregate values requested by that actuator.

– It interoperates with external systems (e.g., a different application server) and the authorized personnel administrating the SCN.

- **Actuator:**

– It requests from the SCN controllers via the central communication channel aggregate values, which are necessary for making a decision.

– It receives from the SCN controllers via the central communication channel the requested aggregate values.

– It forms the appropriate control commands.

– It transmits information about its own status to the SCN controllers via the central communication channel.

- **Mote:**

– It receives requests from the SCN controllers via the SCN infrastructure about sensed data or reference values.

– It transmits to the SCN controllers the requested data via the SCN infrastructure.

7.3.1.3 Decision-making process

The decision-making process follows the following procedure:

1) The necessary sensed data, reference values and aggregate values are kept in the SCN controllers' memory and regularly updated.

2) Each actuator needs to have the up-to-date aggregate values necessary to make a decision. These aggregate values are fetched from the SCN controllers.

3) Each actuator forms a control command depending on the aggregate values.

An example of a flow chart for centralized configuration is shown in Figure 7-4.

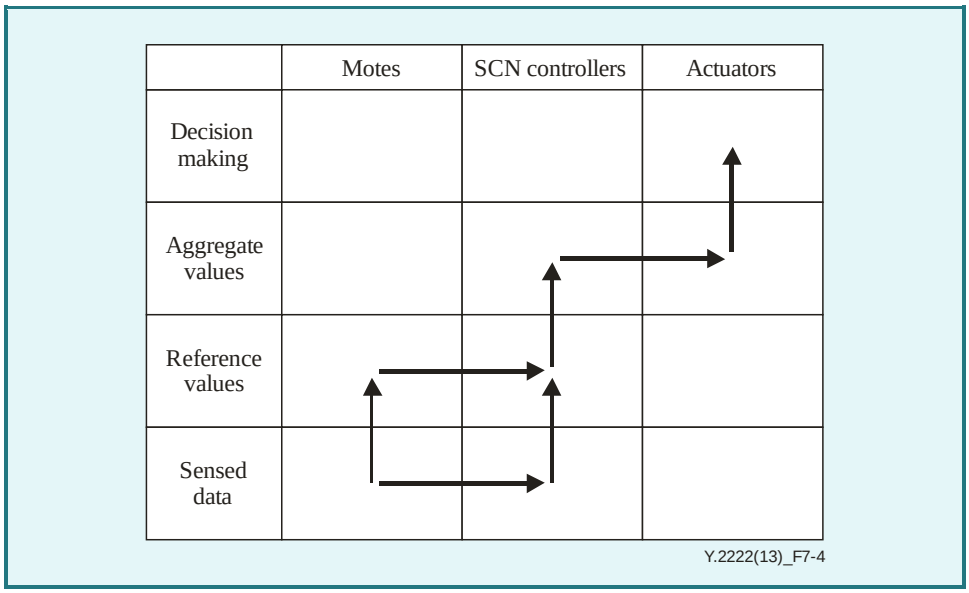


Figure 7-4 – Example of flow chart for centralized configuration for SCN applications

7.3.2 Ad-hoc configuration for SCN applications

7.3.2.1 Introduction to ad-hoc configuration for SCN applications

This configuration is so called because it utilizes ad-hoc networks (e.g., based on Bluetooth or Wi-Fi technologies) to deliver data to actuators. It should be employed when there is the possibility to expand the existing SCN infrastructure and the actuators have some ad-hoc wireless network capabilities. Some intermediate devices called gates are used to provide a communication channel between actuators and one or several nearby motes.

7.3.2.2 Role distribution in ad-hoc configuration for SCN applications

- **SCN controller:**

– It receives from the actuators via the central communication channel requests about aggregate values, which are necessary for making a decision but cannot be calculated by the actuators themselves.

– It requests transmission of sensed data and reference values from the appropriate motes and regularly calculates the necessary aggregate values.

– It transmits to each actuator via the central communication channel the aggregate values requested by that actuator.

– It interoperates with external systems (for example, a different application server) and the authorized personnel administrating the SCN.

- **Gate:**

– It receives requests from the actuators via the ad-hoc network about sensed data and reference values and forwards them to the motes via the SCN infrastructure.

– It transmits the requested data to the actuators via the ad-hoc network.

- **Actuator:**

– It requests the necessary sensed data and reference values from the gates via the ad-hoc network.

– It requests from the SCN controllers via the central communication channel aggregate values, which are necessary for a decision but cannot be calculated by the actuator itself.

– It receives from the gates via the ad-hoc network the requested sensed data and reference values and calculates the other necessary reference values and aggregate values.

– It receives from the SCN controllers via the central communication channel the requested aggregate values.

– It forms the appropriate control commands.

– It transmits to the SCN controllers information about its own status via the central communication channel.

- **Mote:**

– It receives requests from the SCN controllers and the gates via the SCN infrastructure about sensed data or reference values.

– It transmits the requested data to the SCN controllers and the gates via the SCN infrastructure.

7.3.2.3 Decision-making process

The decision-making process follows the following procedure:

1) The necessary sensed data, reference values and aggregated values are kept in the SCN controllers' memory and regularly updated.

2) Each gate forwards sensed data and reference values from the motes to the actuators.

3) Each actuator sends requests for sensed data and reference values to the gates, and then stores the received ones in memory. The data requests can be of different types, such as broadcast request (the gates send data of all motes on demand to the actuator), threshold-exceeding request (gates send data only of the motes whose sensed data exceed some thresholds), etc.

4) Some other reference values can be regularly computed as needed by the actuators based on received sensed data and reference values.

5) Each actuator needs to have the up-to-date aggregate values necessary to make a decision. These aggregate values can be computed by the actuator itself or fetched from the SCN controllers.

6) Each actuator forms a control command depending on the aggregate values.

An example of flow chart for ad-hoc configuration is shown in Figure 7-5.

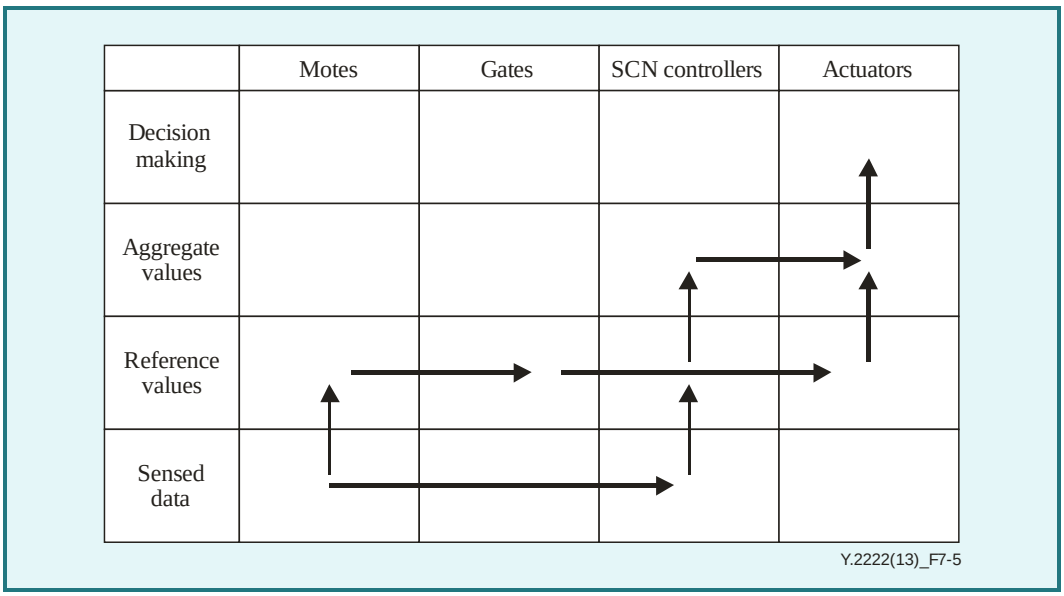


Figure 7-5 – Example of flow chart for ad-hoc configuration for SCN applications

8 Service requirements of SCN applications

The followings are high-level service requirements of SCN applications.

8.1 Connectivity

1) SCN applications are required to support three types of communications:
   - SCN controller-actuator communication: actuators communicate with the SCN controllers;
   - Infrastructure communication (including one-to-one, one-to-many, many-to-one and many-to-many): motes from different mote groups communicate with each other and with the SCN controllers;
   - SCN controller-NGN communication: NGN elements communicate with the SCN controllers.
2) SCN applications are recommended to support the following type of communications:
   - Mote-actuator communication (including one-to-one, one-to-many and optionally many-to-one and many-to-many): actuators communicate with motes.
3) SCN applications can optionally provide these types of communications:
   - Inter-actuator communication: actuators communicate with each other;
   - Actuator-NGN communication: NGN elements communicate with actuators;
   - Mote-NGN communication: NGN elements communicate with motes.

8.2 Mobility support

Actuators as well as motes can be classified as follows:

– Network-specific actuators and motes, designed to operate in specific SCN application fields (e.g., complex industrial mechanisms);

– Mission-specific actuators and motes, designed to be deployed after preliminary setup (e.g., mass-produced switches);
– Generic actuators and motes, designed to be instantly deployed and used in a wide range of SCN applications (e.g., PDAs).

SCN applications have the following general requirements concerning mobility:

1) SCN applications are required to support nomadism for mission-specific and generic actuators and motes.
2) SCN applications are recommended to support seamless handover for generic actuators and motes.

8.3 Context awareness

Context information can have a great effect on the decision-making process. Examples of context information elements include the following:

– Capabilities of SCN objects;
– Location of actuators and motes;
– Presence and operational status of actuators and motes;
– Traffic load and computational load of SCN objects;
– Information about faults of SCN objects.

SCN applications have the following general requirements concerning context-awareness:

1) Context information is required to be collected and distributed to any interested SCN object.
2) Delays in context information updates are required to be minimized so that the reliability of the decision making is not decreased significantly.

NOTE – Requirements specific to some context-related aspects are given in other specific clauses.

8.4 Location awareness

Location information needs to be maintained and managed in order to support context awareness with location information for SCN applications. Management can be in static or dynamic conditions of the located entities. In addition, SCN application service and device discovery can be facilitated by the usage of the location information. Thus, SCN applications have the following requirements:

1) Location information of mote groups is recommended to be managed for SCN applications.
2) Location information of actuators is recommended to be managed for SCN applications.
3) Location information of individual mote can be optionally managed for SCN applications when the location information of a single mote is useful.

8.5 Presence awareness

Presence information includes the network-related part of presence information (e.g., concerning connectivity and willingness to communicate) and the operation-related part of presence information (e.g., concerning workability and currently performing operation). SCN applications have the following requirements:

1) The network-related part of presence information is required to be managed for SCN applications.
2) The operation-related part of presence information is recommended to be managed for SCN applications.

8.6 Traffic and load awareness

To realize traffic and load optimization, SCN applications have the following requirements:

1) Information related to traffic and computational capabilities of SCN objects is recommended to be registered for SCN applications.

2) Information about current traffic and computational load is recommended to be managed for SCN applications.

8.7 Fault awareness

A SCN needs to react to the failure of any SCN objects in order to provide reliability and availability.

1) Information about SCN object faults is required to be managed for SCN applications.

8.8 Routing

1) SCN applications are required to support routing using distributed mechanisms, such as those based upon peer-to-peer (P2P) techniques.

2) SCN applications are recommended to identify the preferred path between any pair of SCN objects. The path selection can be based on historical data or on real-time data to reflect the traffic congestion situation between those SCN objects.

8.9 Load balancing

1) SCN applications are required to dynamically balance the traffic load of SCN objects, based on the status and/or capabilities of SCN objects, traffic balancing policy, etc.

8.10 Scalability

1) SCN applications are required to offer scalability by using P2P and/or other distributed mechanisms, so that the capacity of the SCN infrastructure to provide services to users is proportional, or nearly proportional, to the number of the motes and actuators.

8.11 Fault tolerance

1) SCN applications are required to ensure reliability and availability of the SCN infrastructure in order to handle a single mote failure and a mote group failure.

2) SCN applications are recommended to ensure reliability and availability of the SCN infrastructure in the case of failure of the SCN controllers.

NOTE – In the case of these failures, the capabilities of the failed SCN object(s) can be dynamically replaced by those of other SCN objects to provide consistent service to end users.

8.12 Quality of service (QoS)

Different SCN applications may have different QoS requirements. For example, data transmission in verification applications may require much lower delays than in other SCN applications.

1) SCN applications are recommended to support QoS differentiation according to the required service level quality.

2) It is required that the traffic volume generated by SCN applications be managed.

3) It is recommended that SCN applications avoid access concentration in a single SCN controller or a single mote.

4) It is recommended that specific QoS support for emergency applications be provided.

NOTE 1 – Clause 8.18 provides further information and requirements about emergency applications in SCNs.

5) It is recommended that specific QoS applications intended for pledging of security of decisions be provided.

NOTE 2 – Clause 8.14 provides further information and requirements about pledging of security of decisions.

8.13 Management

1) SCN applications are required to allow the user to enable and disable the provided services.
2) SCN applications are required to allow the user to apply different policies concerning allowing and denying specific commands to actuators.
3) SCN applications are recommended to provide the user with the ability to personalize the services.

8.14 Pledging of security of decisions

The decision-making process in SCN applications includes different activities on different SCN objects and can be very complicated. As a result, there are a number of sources of errors in decisions including erroneous, outdated, incomplete data and object synchronization errors. Some erroneous decisions of SCN applications can entail considerable negative consequences.

1) SCN applications are required to provide measures to avoid considerable negative consequences of their decisions on condition that all the actuators carry out commands given by the SCN applications exactly.
2) SCN applications are required to provide all the necessary measures to identify the party responsible for erroneous decision operations entailing considerable negative consequences.
3) SCN applications are required to provide operational logging sufficient to determine the source of errors entailing considerable negative consequences.

8.15 Open service environment (OSE) support

SCN applications can optionally support open service environment (OSE) capabilities as described in [ITU-T Y.2020] and [ITU-T Y.2234].

In case of SCN applications' support of OSE capabilities, SCN applications, services, actuators, motes and mote groups are recommended to be registered beforehand in order to enable the ability to be discovered (by specifying one or more related attributes).

It may be desirable for the user to use the same application in different SCN infrastructures. As the user changes his location and moves to another SCN infrastructure, service discovery is automatically started to check if that SCN infrastructure provides the required services. If these services are not registered, the SCN application may try to use a service composition procedure to create the required services from other existing services based on the capabilities of the SCN infrastructure. A service description language and its associated execution framework are recommended to be provided to support service registration, discovery and composition.

The following requirements are identified in the case of SCN applications' support of OSE capabilities:

1) It is recommended to support registration and discovery of SCN applications, services, actuators, motes and mote groups.
2) It is recommended to support at least one service description language and its associated execution framework.
3) Automatic service discovery and service composition can be optionally supported.

8.16 NGN service integration and delivery environment (NGN-SIDE) support

SCN applications can optionally support next generation network service integration and delivery environment (NGN-SIDE) capabilities [ITU-T Y.2240].

The SCN objects can be integrated with resources from different domains (e.g., telecommunication domain (fixed and mobile networks), broadcasting domain, Internet domain or content provider domain) over NGN with the use of NGN-SIDE.

From this point of view, the SCN objects can be considered as resources, and NGN-SIDE acts as a mediator between these resources and SCN applications. More specifically, the NGN-SIDE adaptation layer adapts the resources offered by the SCN objects in order to provide uniformly adapted resources (e.g., control and media format) for interaction with the NGN-SIDE integration layer as described in [ITU-T Y.2240]. The NGN-SIDE adaptation layer provides adaptation capabilities, called adaptors, hiding the details of the resources offered by the SCN objects.

The following requirement is identified in the case of SCN applications' support of NGN-SIDE capabilities:

1) SCN applications are required to access SCN objects through adaptors.

8.17 Mass mobile user terminal support

Most of the mass mobile user terminals have no technical possibility of direct data exchange with an existing infrastructure of motes. However, it is generally desirable to offer SCN applications to the users of these terminals because of their prevalence and considerable communication and computing capabilities. SCN applications may provide connectivity with such terminals using available communication technologies (e.g., Bluetooth, GPRS/3G, Wi-Fi, WiMAX).

1) SCN applications can optionally support the mass mobile user terminals including the terminals that are not specifically intended for SCN applications.

8.18 Emergency management applications

Some SCN applications provide support for early-warning emergency enhanced by recommendations about the escape from emergency situations based on the features of user location awareness and service personalization according to users' medical peculiarities or duties.

1) SCN applications for emergency management are recommended to be supported by the SCN objects.

9 Security considerations

SCN applications are required to support the integrity and confidentiality of the data exchanged during the application operations.

SCN applications are required to provide security for exchanged data against malicious attacks.

SCN applications are required to authenticate motes and mote groups to prevent compromising of sensed data.

It is recommended to provide a secure channel to protect the sensed data among SCN objects.

NOTE – Detailed security requirements for SCN applications are outside of the scope of this Recommendation.

Appendix I

Use case of SCN for verification

(This appendix does not form an integral part of this Recommendation.)

I.1 Errors in decisions

The decision-making process in SCN applications includes different activities and can be very complicated. As a result, there is a whole series of sources of errors in decisions:

– Unreliability of communication channels. Normally, SCN applications intend to make wide usage of wireless communications which are more error-prone in comparison with wired communications.

– Distributed calculation model. This model makes it difficult to synchronize all the activities and to provide all the SCN objects with actual data.

– Not very predictable duration of the decision-making process, due to different delays in calculations and data transmissions.

– Mobility of actuators and possibility of their usage for different SCN applications. When this is accompanied by the presence of several software and hardware vendors of actuators, this feature deprives the system designer of the possibility to test the system thoroughly in various conditions. Furthermore, when the interaction of the actuator with the SCN infrastructure is not on a systematic basis, making critical decisions cannot be entrusted entirely.

– Hardware and software errors.

As a result, control commands given by SCN applications should be analysed in order to ensure that possible errors do not result in considerable negative consequences. When a control command is intended for an information actuator, it can be analysed by the human who receives the control command. When a control command is forwarded to another network by a gateway actuator, the duty of analysis shifts onto the other network. But when a control command should be carried out without any direct human intervention to a machine actuator, special measures should be taken to counteract errors in decisions made by SCN applications.

I.2 Verification

For a machine actuator, there is a set of critical operations which can lead to considerable negative consequences when carried out in an improper system state. To avoid this, for each critical operation, a set of rules should be defined, which must be checked before this operation and/or while the operation is in progress. These rules are called "verification rules". To check the verification rules, a number of values of different types must be determined:

– Aggregate values, reference values, sensed data obtained in SCN application as part of normal flow of decision making.

– Aggregate values, reference values, sensed data obtained in SCN application which are only intended to support verification.

– Sensed data obtained from sensors associated with machine actuators.

– Values obtained upon request from SCN controllers or NGN entities.

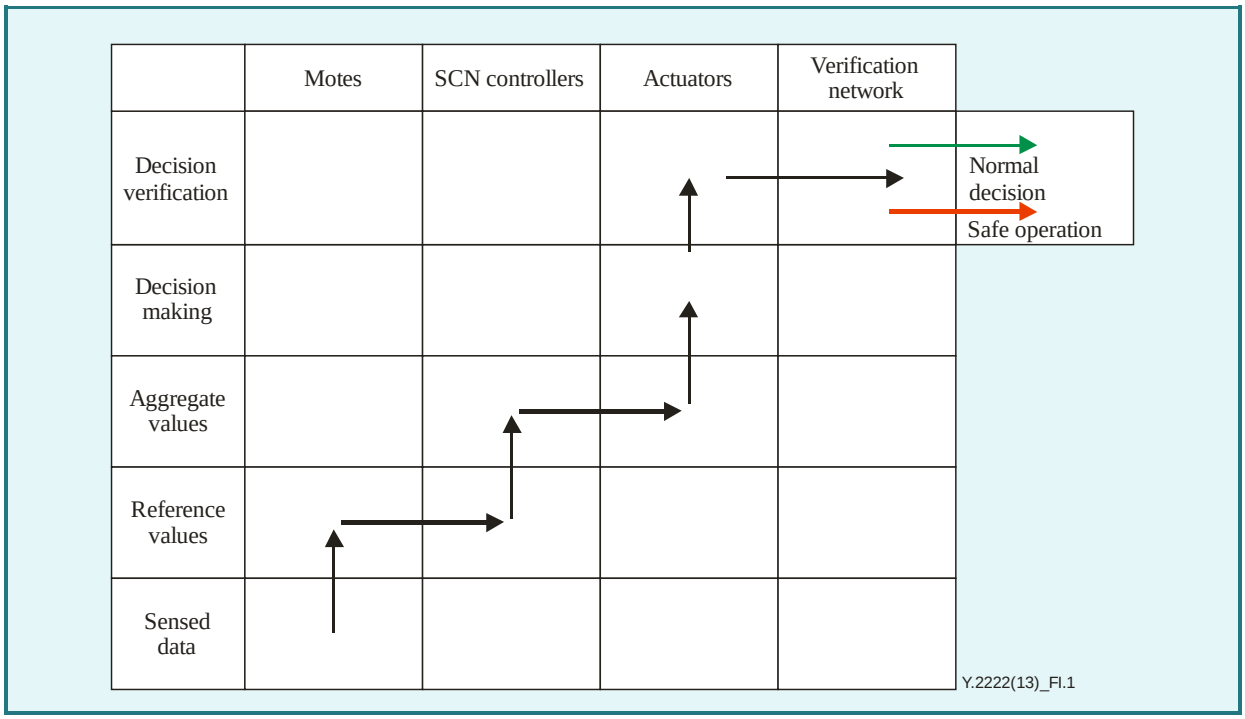
In order to provide verification applications, verification networks are used. A verification network consists of devices and communication channels which are used to fetch and process the above described values. A verification network can have some elements which are also SCN objects or that constitute a part of the SCN infrastructure. A verification network may have much more strict requirements concerning reliability, security and performance compared to other SCN applications. Data processing and transmission for the purpose of verification may have higher priority in QoS in comparison with other operations of SCN applications.

Examples of operations which require verification, verification rules and verification network elements are given below.

The operations in a verification network can be executed once before critical operations of SCN applications, or at the time of these operations or periodically. For each possible machine actuator state and detected verification rule failure, some operations should be defined to be performed instead of those given by the SCN application when one of the verification rules is mismatched. Such operations can be, for example:

– Immediate machine actuator stop.
– State transition of the machine actuator to some safe state.
– Alarm notification to authorized personnel administrating the SCN.
– Generation of a log entry.
– No action.

An example of flow chart for the decision-making process in verification networks can be depicted as in Figure I.1.



F Figure I.1 – Decision-making process in verification networks

The figure depicts a normal decision-making flow, but because the decision involves a machine actuator, the verification process is initiated by the verification network.

If some of the checks of the verification process fail, some safe operation (or no action) is performed instead of normal decision.

I.3 Examples of verification applications

I.3.1 Fire safety system

A fire safety system is deployed in a building. If one of its sensors detects ignition or smoke, it activates machine actuators that lock doors and windows to prevent air circulation. The considerable negative consequences include people being locked inside the rooms and people being crushed by automatic doors.

Therefore, the verification network should consist of:

– sensors of movement in rooms;

– light sensors between door wings, and strain sensors on the motors of door wings to halt the motors if there is a person between the door wings.

I.3.2 On-road speed control

Sensors are used to monitor road conditions. Vehicles are equipped with actuators able to communicate with sensors which can automatically decrease the vehicle speed if the road conditions are dangerous.

A possible relevant negative consequence is that if the speed decrease is too fast, chances of collision with vehicles behind can happen. To prevent this result, the verification network should use a rear parking sensor that can monitor the free space behind the vehicle and slow the rate of speed decrease if there is another vehicle nearby.

I.3.2 Rescue robots

After a plane crash, rescuers use autonomous robots to retrieve survivors from under the wreckage. These robots pick up wreckage fragments and move them to a safe place.

If a robot runs out of energy after it has picked up a heavy fragment, it could drop it on the injured human or rescuers, causing therefore relevant negative consequences. The verification network should be based on energy sensors and on robot's CPU. The latter should be able to estimate the weight of the fragment to be picked up and the amount of energy required. If the robot's energy level is not sufficient, the operation must be blocked.

Appendix II

Use case of SCN for emergency management

(This appendix does not form an integral part of this Recommendation.)

An emergency management system [b-ITU News] uses motes to observe the physical conditions of a building (temperature, smoke, etc.). At the entrance to the building, a mobile user terminal (e.g., phone, PDAs or tablet PC) automatically connects to the SCN infrastructure and obtains data from the motes.

An emergency management system automatically detects emergency situations. In this case, the user equipment launches software for guidance in emergency cases. It gives instructions on how to leave the building in the safest way, for example:

– evacuation plans or maps;
– step-by-step sound commands and visual hints (e.g., interior photos with overlaid arrows towards the exit);
– videos showing how to use safety equipment.

The content of these instructions depends on various factors, for example:

– state of building detected by motes like accessibility and hazard level of rooms and escape routes;
– position of the user determined by the nearest network node or using GPS;
– user's state of health determined by e-health equipment.

In addition, special information containing both needs and duties is taken into account by the system. It may be limitations of motion and senses for disabled people that influence the route choice. At the same time, special personnel of the building may need specific instructions concerning their service duties (for example, emergency case specialists at the time of an accident at a nuclear power plant).

In-building actuators (e.g., door openers, emergency lightning and sprinklers) also get commands from the system and start working.

The emergency operation centre and/or emergency services also get information about the emergency including its type and place of origin.

If the motes detect no emergency situations, energy consumption of the system should be minimized. The system should be optimized for low traffic. Motes and mobile user terminals may be sleeping most of the time or be used for other applications. However, as an emergency situation is detected, the system must switch to special mode in order to rescue people as soon as possible (15 minutes or less), e.g.,:

– all the motes and mobile user terminals should be awakened from sleep;
– traffic not related to rescue should be discarded to provide low latency and high transmission rate;
– software applications running on mobile user terminals and not required for rescue (e.g., games, media players) should be suspended to decrease hardware resources consumption (CPU, memory, etc.) and switch user's attention entirely to the rescue tasks.

Figure II.1 shows an emergency management use case.

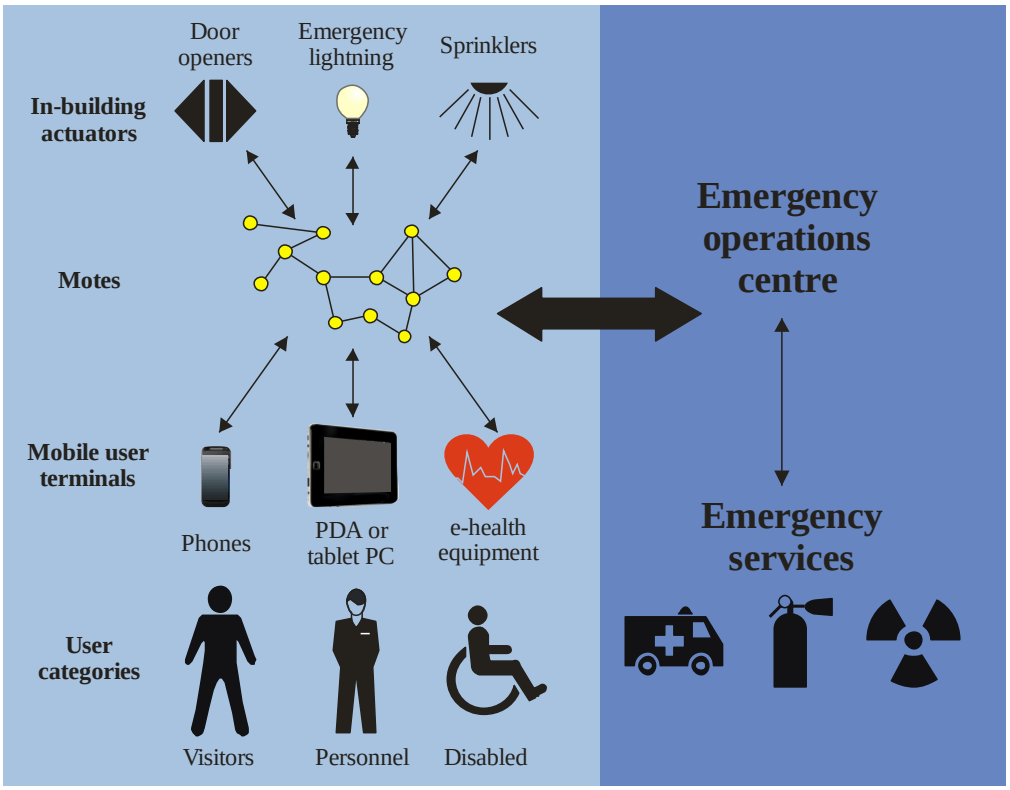


Figure II.1 – Emergency management use case

Bibliography

[b-ITU-T F.744] Recommendation ITU-T F.744 (2009), *Service description and requirements for ubiquitous sensor network middleware*.

[b-ITU-T Q.1706] Recommendation ITU-T Q.1706/Y.2801 (2006), *Mobility management requirements for NGN*.

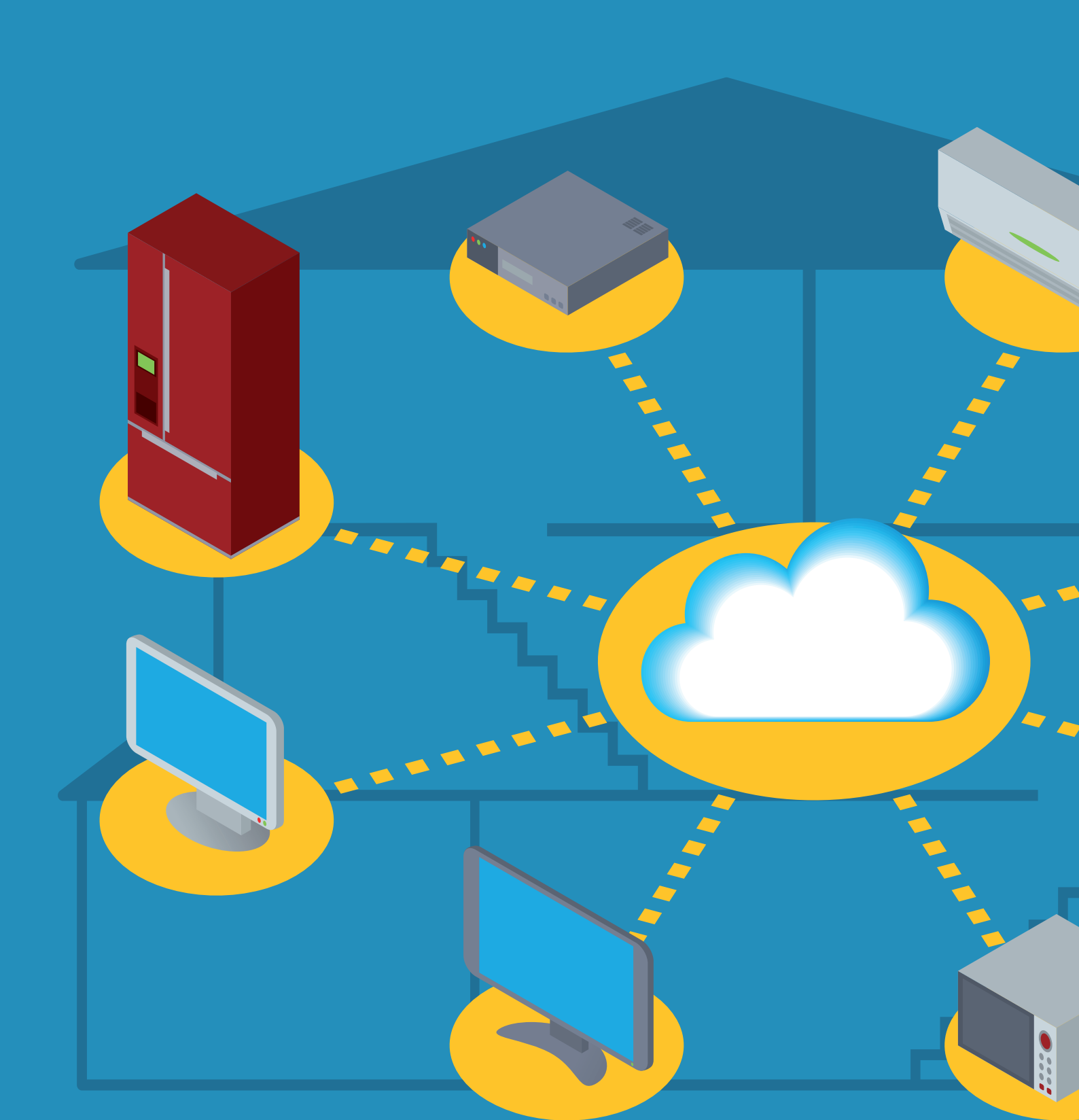
[b-ITU-T Y.101] Recommendation ITU-T Y.101 (2000), *Global Information Infrastructure terminology: Terms and definitions*.

[b-ITU-T Y.2001] Recommendation ITU-T Y.2001 (2004), *General overview of NGN*.

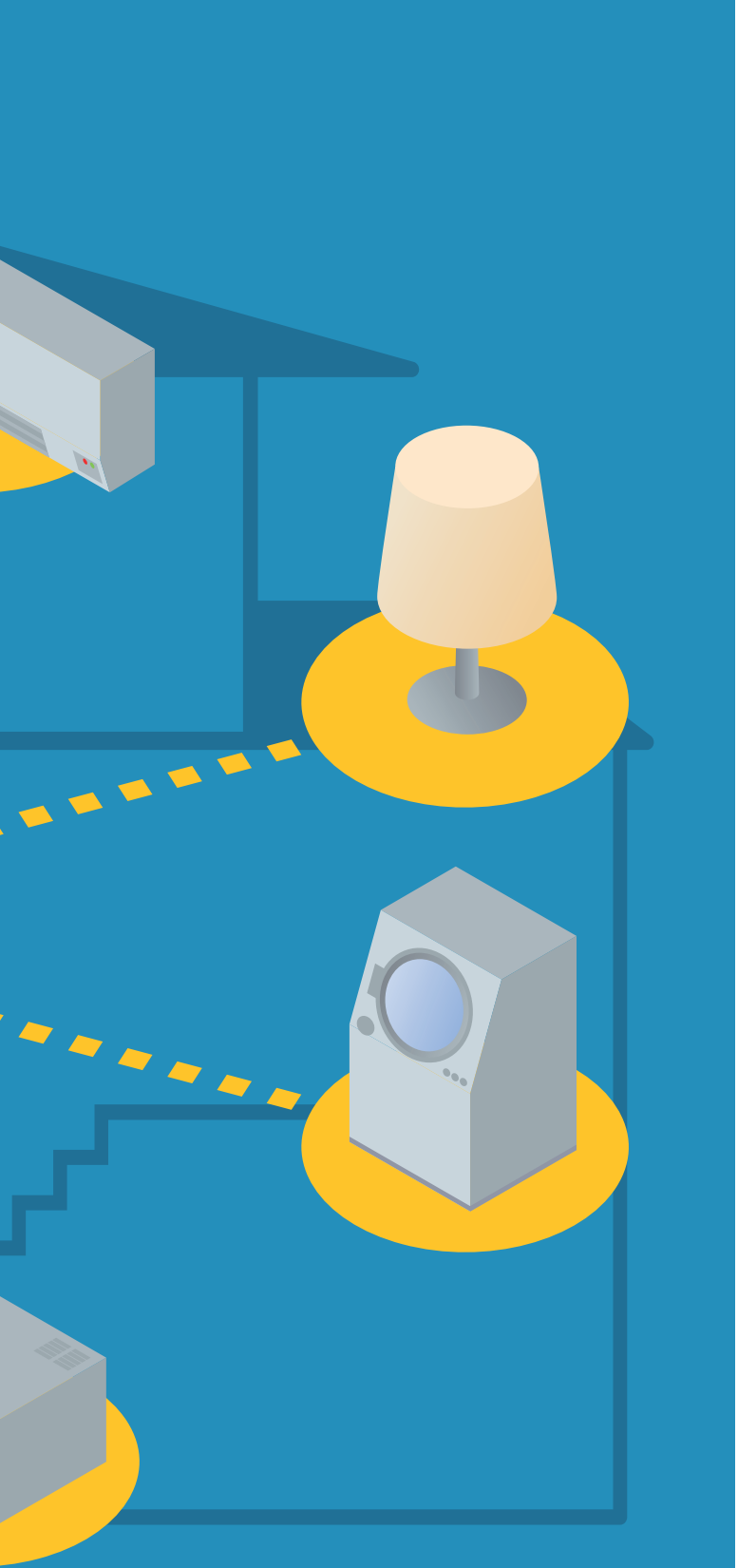
[b-ITU-T Y.2201] Recommendation ITU-T Y.2201 (2009), *Requirements and capabilities for ITU-T NGN*.

[b-ITU-T Y.2221] Recommendation ITU-T Y.2221 (2010), *Requirements for support of ubiquitous sensor network (USN) applications and services in the NGN environment*.

[b-ITU News] ITU News No. 3 (April 2012), *Personal safety in emergencies – Innovative application for mobile phones*.
<https://itunews.itu.int/En/2475-Personal-safety-in-emergencies.note.aspx>



Internet of T



hings

Y.4251/F.747.1

Capabilities of ubiquitous sensor networks for supporting the requirements of smart metering services

Capabilities of ubiquitous sensor networks for supporting the requirements of smart metering services

Summary

Recommendation ITU-T F.747.1 identifies the capabilities of ubiquitous sensor networks (USNs) for supporting the requirements of smart metering services. To this end, an overview of smart metering is described, with a clarification between smart grids and smart metering provided. This Recommendation takes into account a few typical use case scenarios of smart metering and identifies the general requirements and USN-based smart metering services to support these use cases. Finally this Recommendation defines USN capabilities based on identified requirements for providing smart metering services.

History

Edition	Recommendation	Approval	Study Group
1.0	ITU-T F.747.1	2012-06-29	16

Keywords

Smart grid, smart metering, USN.

Table of Contents

Recommendation ITU-T Y.4251/F.747.1

Capabilities of ubiquitous sensor networks for supporting the requirements of smart metering services

1 Scope

The main purpose of this Recommendation is to identify the capabilities of ubiquitous sensor networks (USNs) which support the requirements of smart metering services. The scope of this Recommendation covers the following:

– overview of smart metering

– smart metering use case scenarios

– requirements of smart metering services

– USN capabilities for supporting the requirements of smart metering services.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T F.744] Recommendation ITU-T F.744 (2009), *Service description and requirements for ubiquitous sensor network middleware*.

[ITU-T Y.2221] Recommendation ITU-T Y.2221 (2010), *Requirements for support of ubiquitous sensor network (USN) applications and services in the NGN environment*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 sensor [ITU-T Y.2221]: An electronic device that senses a physical condition or chemical compound and delivers an electronic signal proportional to the observed characteristic.

3.1.2 sensor network [ITU-T Y.2221]: A network comprised of inter-connected sensor nodes exchanging sensed data by wired or wireless communication.

3.1.3 sensor node [ITU-T Y.2221]: A device consisting of sensor(s) and optional actuator(s) with the capabilities of sensed data processing and networking.

3.1.4 ubiquitous sensor network (USN) [ITU-T Y.2221]: A conceptual network built over existing physical networks which makes use of sensed data and provides knowledge services to anyone, anywhere and at any time, and where the information is generated by using context awareness.

3.1.5 USN gateway [ITU-T Y.2221]: A node which interconnects sensor networks with other networks.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 demand response: A smart metering feature that allows consumers to reduce or change their use patterns of electricity, gas and water during peak demand usually in exchange for a financial incentive.

3.2.2 sensor network gateway: A sensor network element that connects a sensor network to another network with different architecture or protocols, permitting information exchange between them. See also USN gateway.

NOTE – Sensor network gateway functionalities may include either address or protocol translation or both.

3.2.3 smart grid: An electricity network that can intelligently integrate the actions of all users connected to it – generators, consumers and those that do both – in order to efficiently deliver sustainable, economic and secure electricity supplies.

3.2.4 smart meter: A device in a user's premises for monitoring and controlling electrical power, gas and water usage of home appliances based on demand response information from home appliances.

3.2.5 smart metering: An operation to provide information to consumers and smart metering operators about energy consumption. The information includes how much energy the consumers are using or generating and how much it costs.

3.2.6 smart metering gateway: See USN gateway.

3.2.7 utility: An entity providing services such electricity, gas, water and heating to the general public and/or to industrial and commercial entities.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

USN Ubiquitous Sensor Network

5 Conventions

None.

6 Overview of smart metering

Smart meters are utility meters, for example electricity, gas, water and other meters, which may bring about the end of estimated bills and meter readings, and provide customers and energy distributors and suppliers with accurate information on the amount of a utility that is being used.

Smart metering provides:

– customers with the information they require to become energy savvy and make smarter decisions about their energy usage;

– energy suppliers with the means to better understand and service their customers;

– distributors with an effective tool to better monitor and manage their networks.

In addition, smart metering enables those customers who choose to generate their own electricity (micro-generators) to be financially rewarded for their contribution to the national grid, and for distributors to better manage this contribution [b-ETSI TR 102 691].

Smart metering may be regarded as one of the key technologies for smart grid systems.

6.1 Smart grids and smart metering

A smart grid is a type of electrical grid that attempts to predict and intelligently respond to the behaviour and actions of all electric power users connected to it – suppliers, consumers and those that do both – in order to efficiently deliver reliable, economic and sustainable electricity services including:

– enhancement of reliability
– reducing peak demand
– shifting usage to off-peak hours
– lower total energy consumption
– actively managing electricity charging
– actively managing other usage to respond to other renewable resources.

Smart grid technologies have already been used in other applications, such as manufacturing and telecommunications. In general, smart grid technology may be divided into seven areas: integrated communications, sensing and measurement, smart metering, advanced components, advanced control, improved interfaces and decision support, and smart power generation. In other words, smart meters are key components of smart grids and consequently, smart metering is one of the crucial features for smart grids.

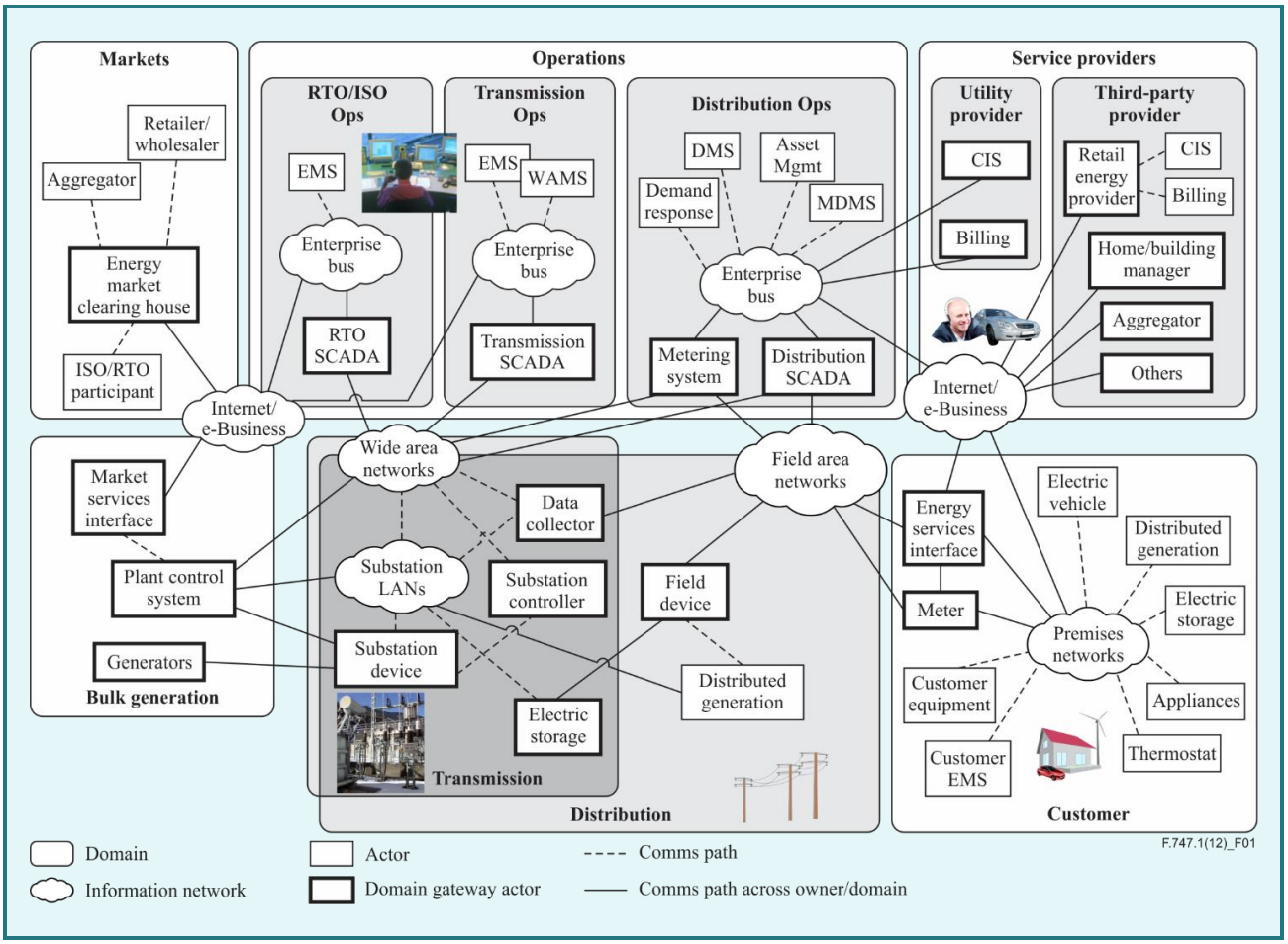


Figure 1 – Smart grid architecture [b-NIST]

6.2 Technical overview of smart metering

It is not only governments and utility companies, such as electricity, gas and water suppliers, but also researchers, that have been interested in automatic meter reading based on communication systems. Examples of smart metering benefits to customers, governments and utility companies are:

– lower metering cost
– energy savings for residential consumers
– reliability of supply
– various pricing schemes to attract new costumers
– easier detection of fraud and of outages
– automated billing.

Smart metering comprises metering and exchange of meter information between smart meters and utility companies. Various technologies can be used for metering and exchanging meter information. For example, power-line communications have been used for delivering electricity power to consumers and for transmitting gas and water measurements to utility providers. Alternatively, mobile networks can be used for exchanging messages in an automatic meter reading system.

Sensor network technologies may be used for metering and collecting information of utility usage, and communication networks can be used for exchanging the information. Figure 2 depicts an overall diagram of smart metering systems. The meter information, obtained from home appliances by sensor nodes is collected and delivered to operators of utility companies.

The operators manage the collected information and inform consumers of variable pricing information or enforced load control messages. Such operators' actions may lead to consumers' reducing energy consumption.

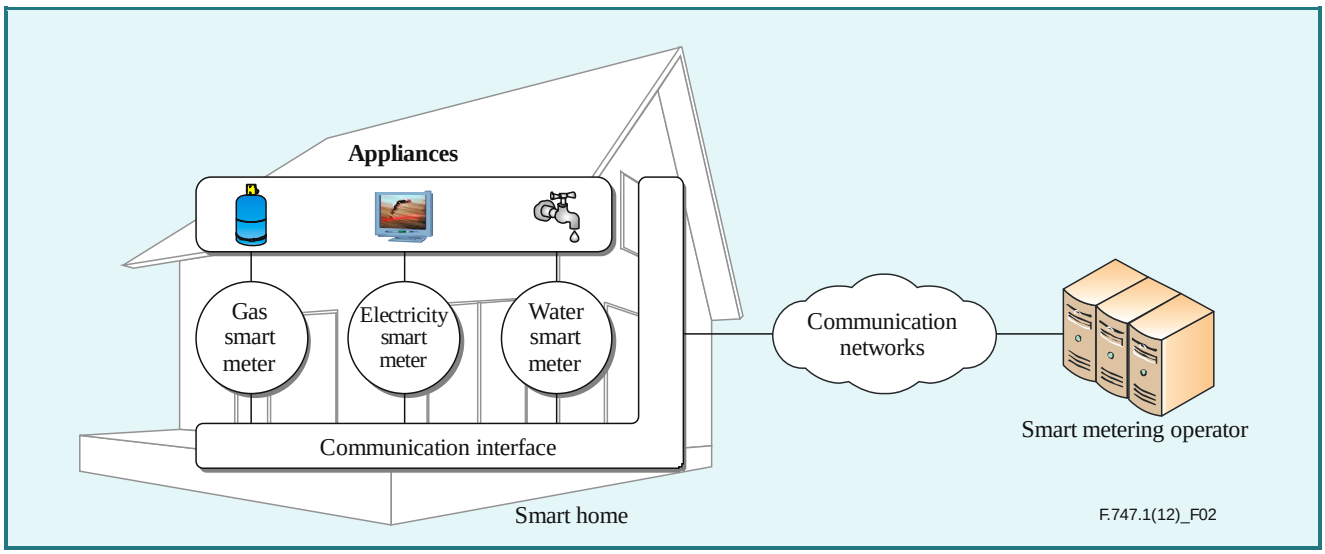


Figure 2 – Overview of smart metering

6.3 USN-based smart metering services

The smart metering services shown in Figure 3 require technological facilities to support metering, and to exchange metering information between smart meters and metering operators in utility control centres.

The basic characteristic of USN applications and services are gathering data by sensor networks and data transmission to a remote server through communication networks. Also, control information is transmitted to sensor networks from the remote server. Furthermore, if USN middleware is deployed, the middleware can provide USN applications or sensor networks with various functions such as data query, data mining, event processing, sensor network metadata directory service, data filtering, context-aware rule processing and sensor network management.

These USN features can be applied to smart metering services. Sensor nodes in sensor networks have the capabilities of metering and delivering metered information to a server at a utility control centre. USN middleware can provide functions which are required by an on-demand remote meter reading scenario and a demand response meter reading scenario.

A smart metering system can be implemented by sensor nodes and USN middleware as follows:

– Smart metering sensor nodes: Each sensor node may have smart metering capabilities, therefore, sensor nodes can act as smart meters supporting the processing of smart metering information for smart metering procedures, such as meter reading collection and tariff setting.

– Smart metering gateway: A sensor network gateway or a USN gateway is capable of connecting a sensor network to another network with different architecture or protocols, permitting information exchange between them. Therefore a sensor network gateway or a USN gateway can be used as a smart metering gateway. A smart metering gateway is only responsible for delivering information between the sensor nodes (smart meters) and metering operators. In cases where a gateway has smart metering capabilities, it collects meter reading data and then delivers it to the metering operator(s).

– USN middleware [ITU-T F.744]: USN middleware provides data filtering, data query, data mining and context-aware rule processing, etc. These features of USN middleware satisfy meter reading data processing requirements and can be applied to an on-demand remote meter reading scenario and a demand response meter reading scenario.

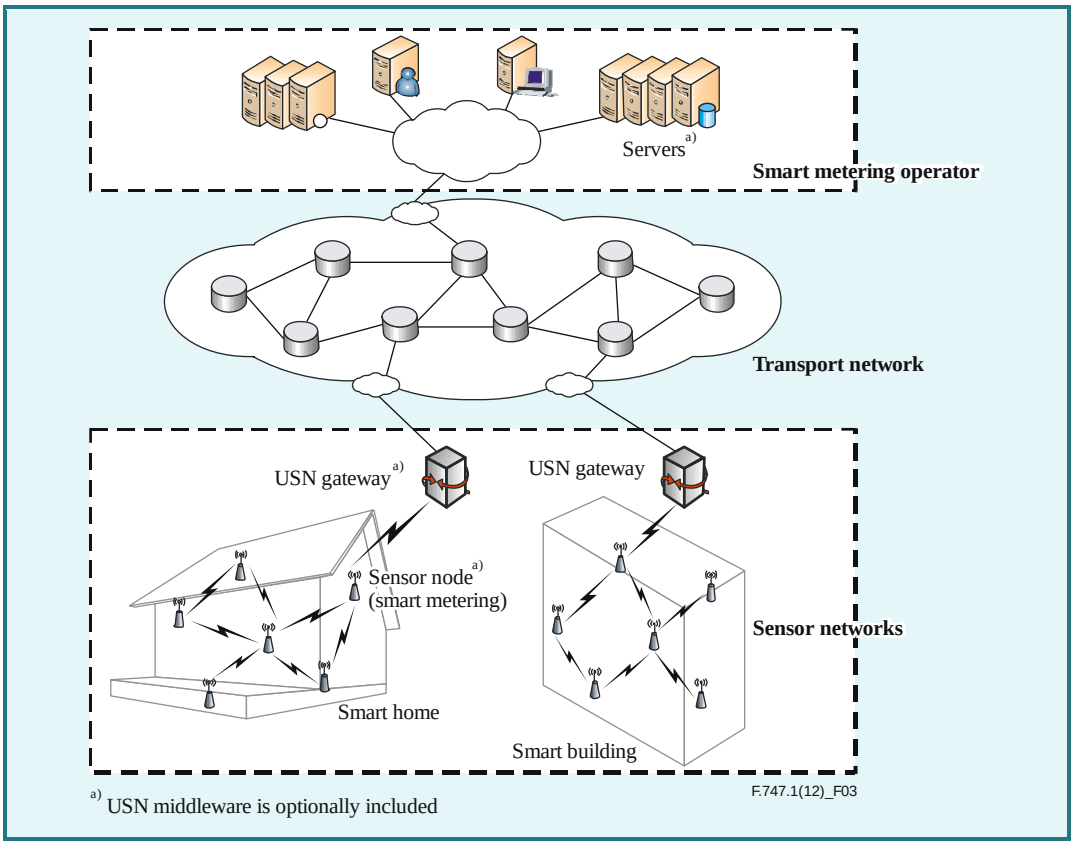


Figure 3 – Smart metering services based on USN

7 Smart metering service scenarios

The following scenarios illustrate the use of smart metering services.

7.1 Scenario I: Regularly scheduled remote meter reading

Scenario I in Figure 4 describes procedures where meter reading data are delivered to smart metering operators at regularly scheduled intervals.

1) A smart metering operator sends a message including schedule information to smart meters.
2) Smart meters are configured with the schedule for measurement and data transfer.
3) According to the schedule, smart meters measure energy consumption from home appliances and obtain measurement data.
4) The smart meters deliver the data to the smart metering operator.

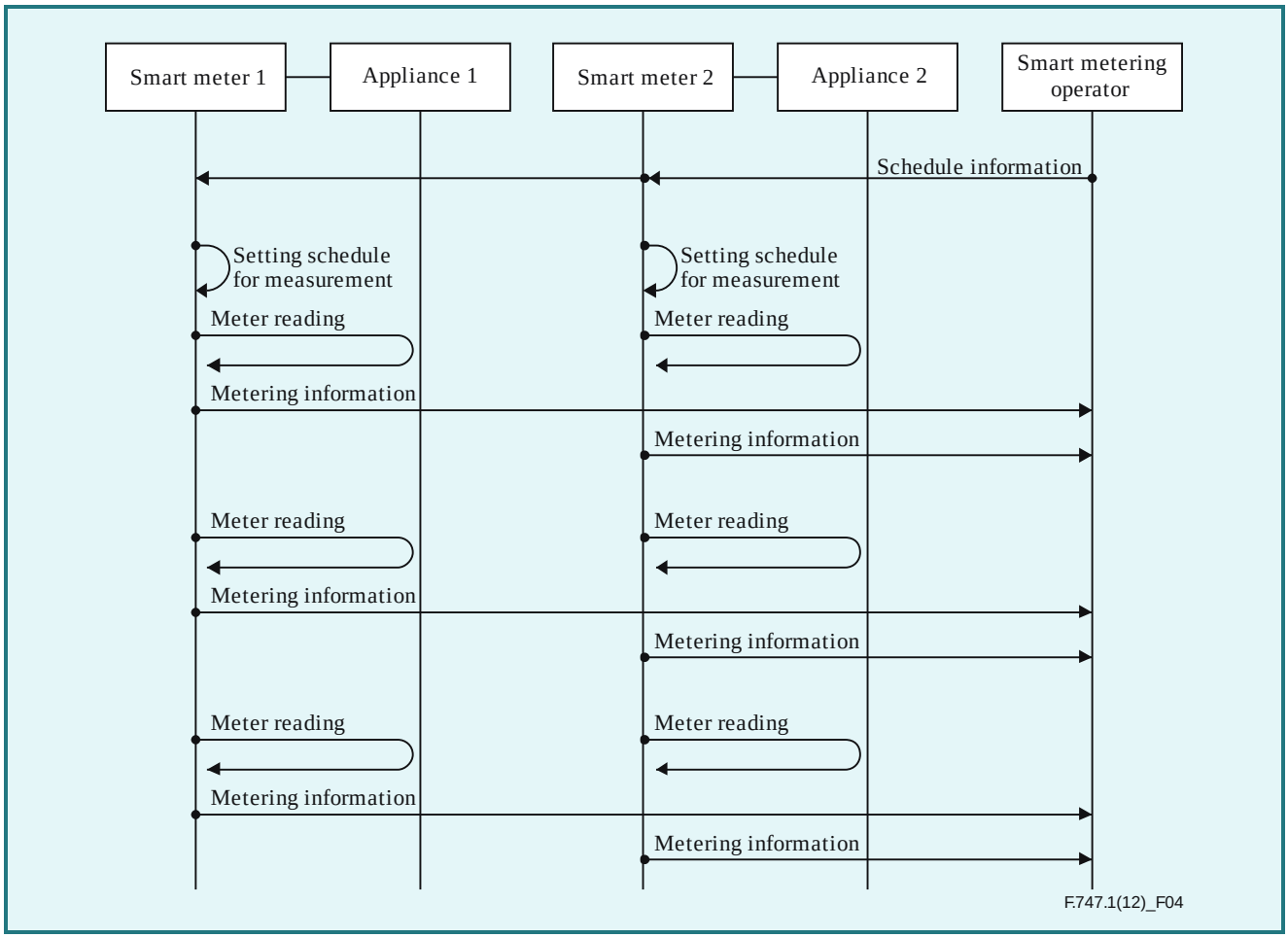


Figure 4 – Scenario I: Regularly scheduled remote metering

7.2 Scenario II: On-demand remote meter reading

Scenario II in Figure 5 describes the procedures for when smart meters measure energy consumption and deliver the results to a smart metering operator on demand.

1) When a smart metering operator collects measurement data from the smart meters, the operator sends a task message for smart meters to collect measurement data.
2) Smart meters verify the operator's message and, if the message is validated, they measure energy consumption.

3) The resulting data are delivered to the smart metering operator.

NOTE – Figure 5 does not include the flow for the message verification, as this can be performed in a number of ways.

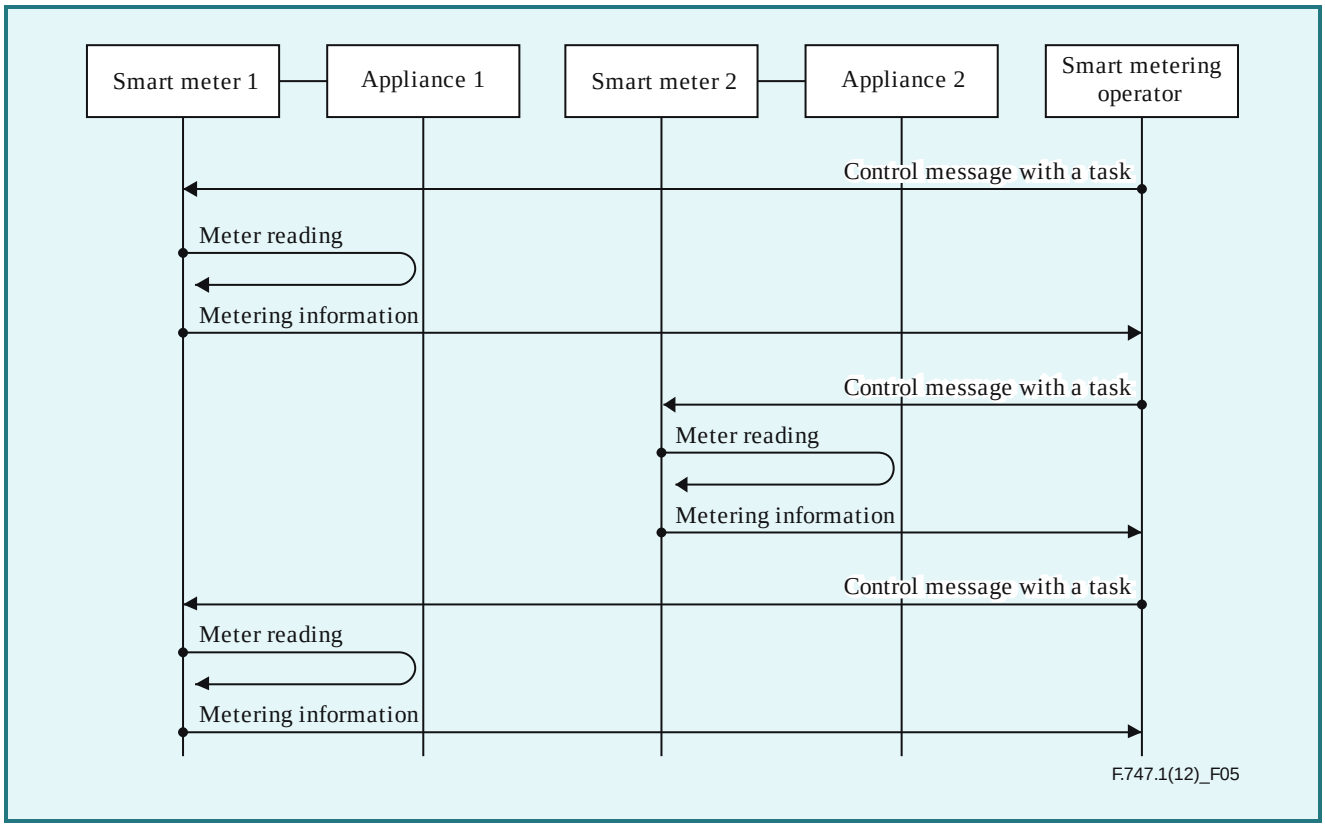


Figure 5 – Scenario II: On-demand remote meter reading (excluding the verification step)

7.3 Scenario III: Demand response

Scenario III illustrated in Figure 6 describes customers responding to the metering operator's demands.

1) A smart metering operator sends a message including schedule information to smart meters.

2) Smart meters are configured with the schedule for measurement and data transfer.

3) When the number of home appliances turned on simultaneously is on the increase, the total amount of energy consumption also steeply increases.

4) Information about the increasing amount of energy consumption is being reported to the smart metering operator at scheduled intervals, and the price of energy may also change according to the tariff policies (e.g., depending on supply and demand).

5) (a) The smart metering operator sends a message to inform consumers of the price change.

   (b) The smart metering operator sends load control messages to enforce the reduction of energy consumption or to force shut off due to a management policy.

6) (a) Smart meters display the message to the consumers, and the consumers may decide to reduce energy consumption, or the home appliances may be switched off by the consumer.

   (b) When receiving the load control messages, the smart meter displays the message and proceeds to shutting off the connected home appliance.

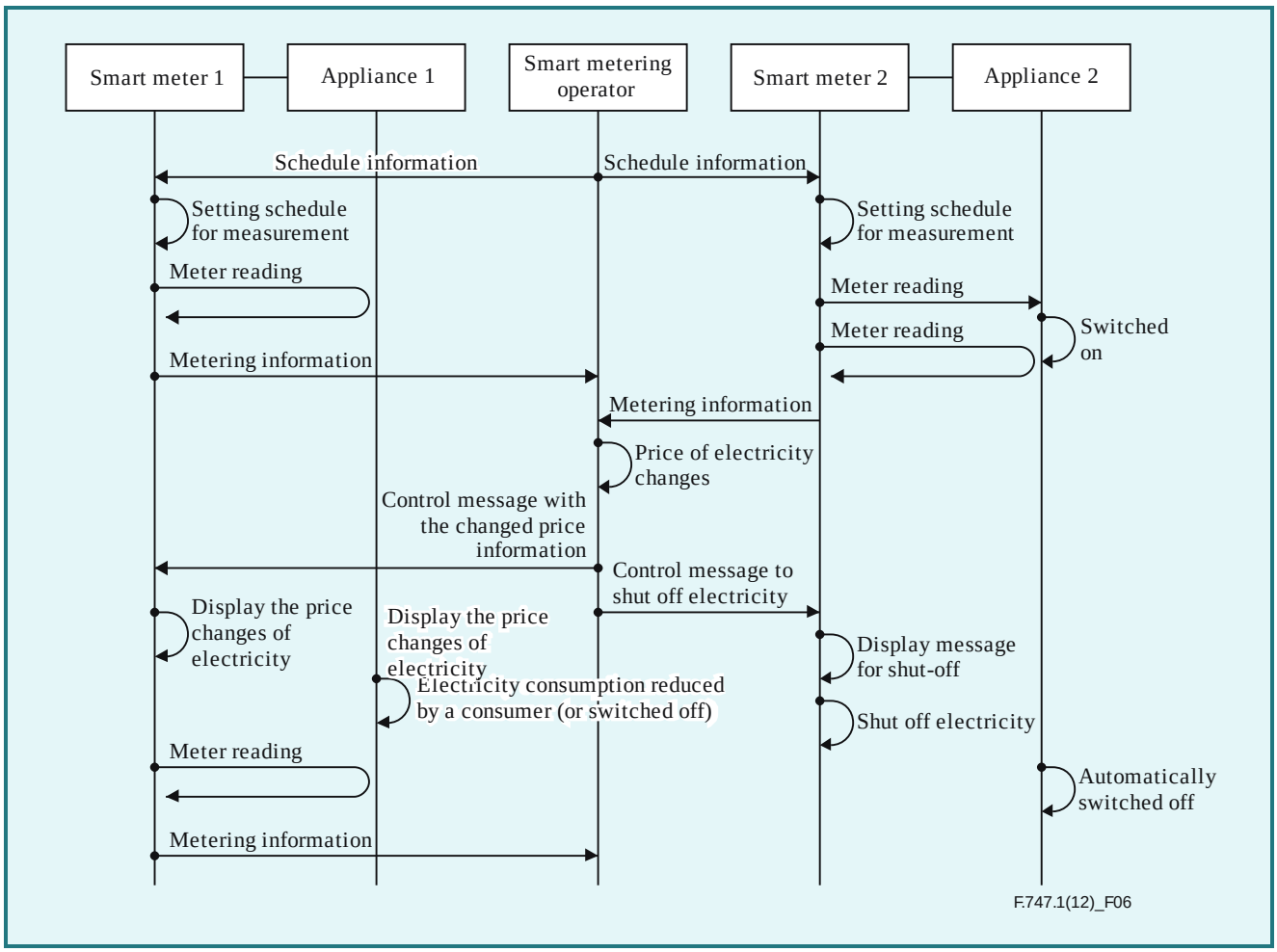


Figure 6 – Scenario III: Demand response

7.4 Scenario IV: Tariff configuration

Scenario IV, illustrated in Figure 7 describes how the price of energy consumption is decided and configured within the meters.

1) A smart metering operator sends a message including schedule information to smart meters.

2) Smart meters are set up with the schedule for measurement and data transfer.

3) The smart meters regularly deliver meter reading data to the metering operator according to regularly scheduled intervals.

4) The price of electricity may be dynamically changed by pricing policies (e.g., depending on supply and demand).

5) When the price changes, a message including changed price information is delivered to all the smart meters concerned.

6) The changed cost information may be shown on their displays, and the new cost is applied.

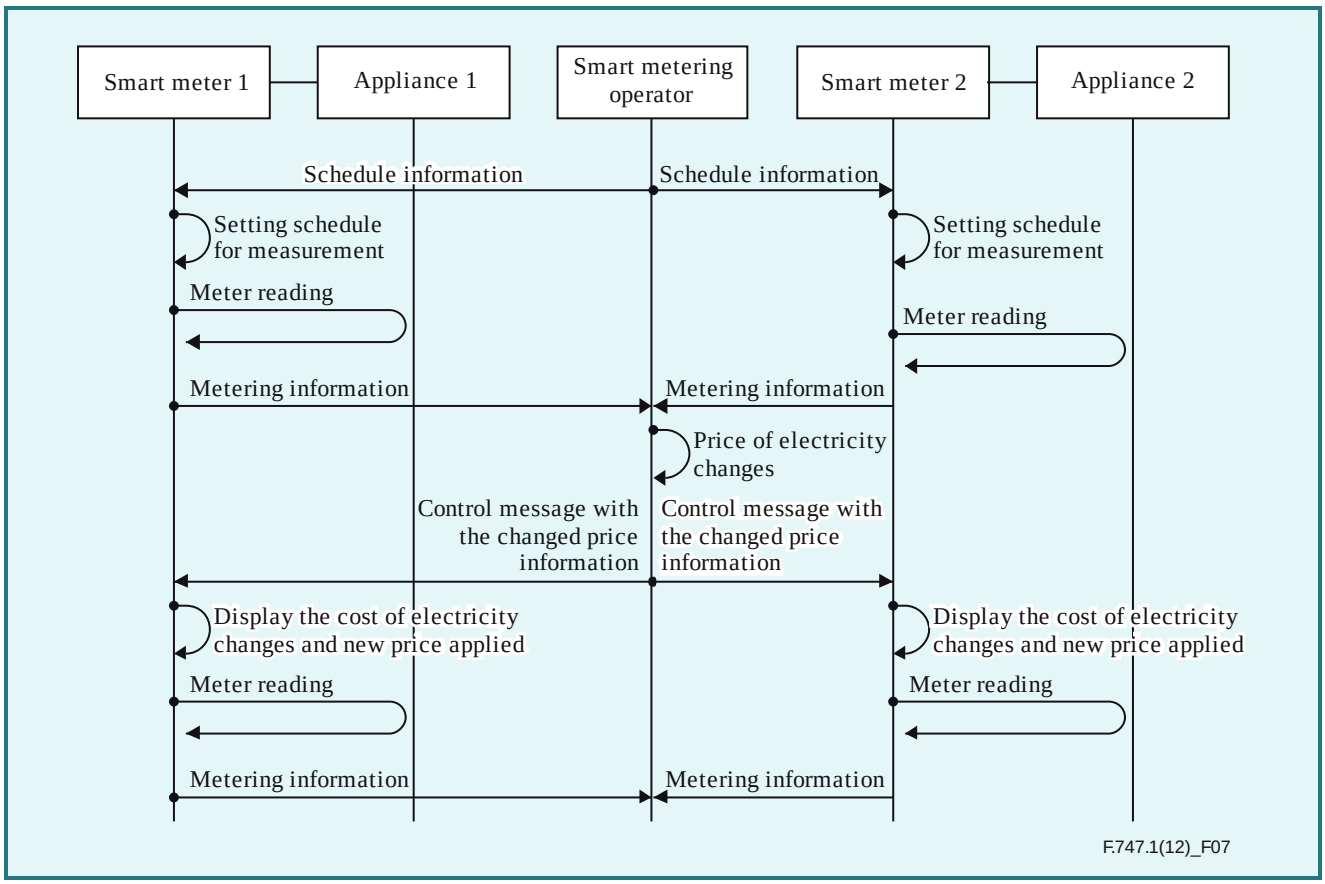


Figure 7 – Scenario IV: Tariff configuration

7.5 Scenario V: Meter reading data aggregation

This scenario describes the procedures of how meter reading data, obtained at scheduled intervals by smart meters, are collected at an aggregating smart meter (or at an aggregating smart metering gateway) and then delivered to metering operators.

1) A smart metering operator sends the smart meters a message to request aggregated metering data.

2) Smart meters are configured to obtain meter reading data.

3) According to the schedule, smart meters measure energy consumption and obtain meter reading data from home appliances.

4) The obtained meter reading data are collected in an aggregating smart meter (or aggregating smart metering gateway). This is represented by Smart meter 2 in Figure 8.

5) Finally, meter reading data are transferred to a smart metering operator.

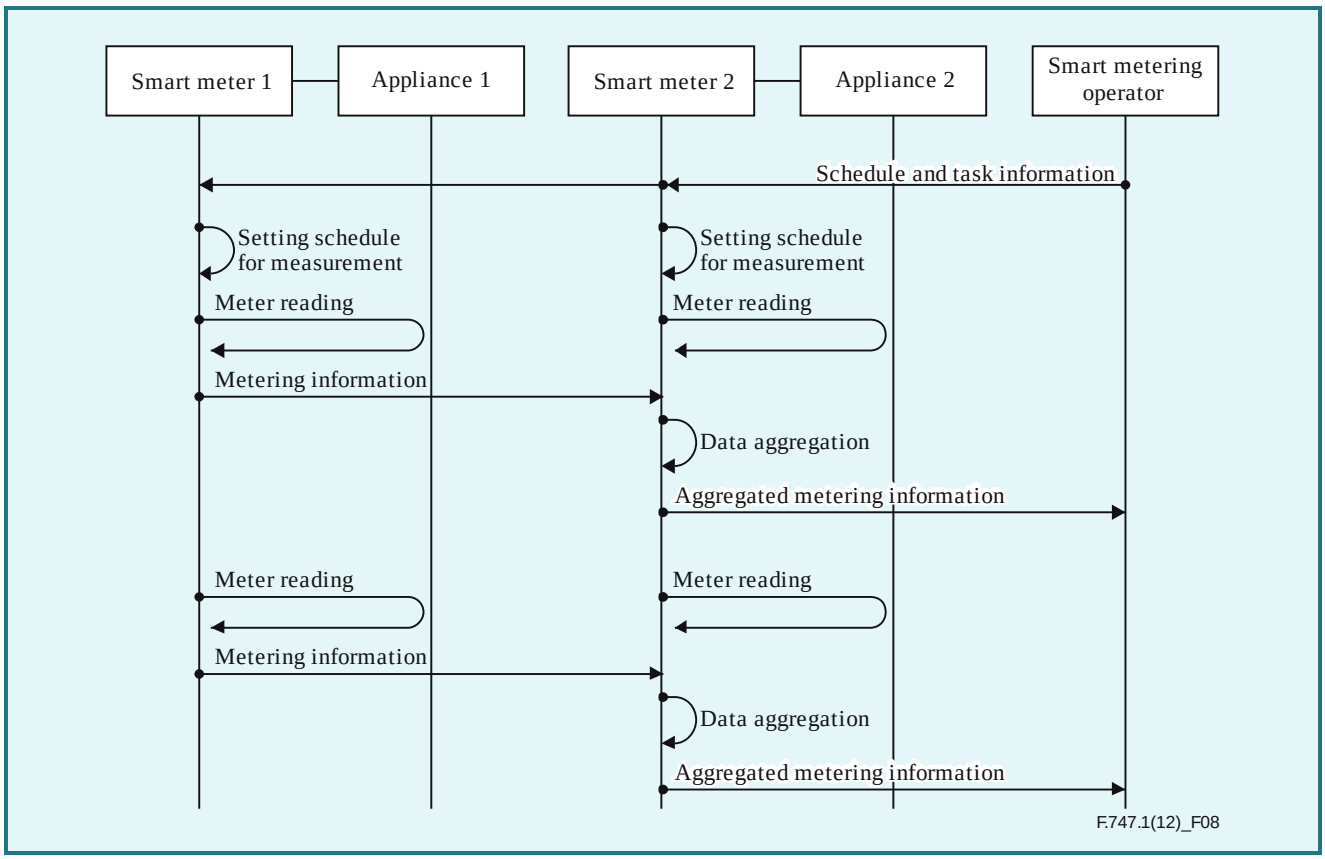


Figure 8 – Scenario V: Meter reading aggregation

8 Network and USN requirements for smart metering services

This clause specifies the sub-set of smart metering service requirements that relate to sensor networks and USNs.

8.1 Time synchronization

In smart metering services, exchanged data uses time stamps between smart meters and the systems of smart metering operators. Therefore, accurate and secure time synchronization should be supported in smart metering systems.

8.2 Reliable information delivery

In smart metering services, meter reading data can be frequently transferred from distributed meters, and also operators can send many control messages to smart meters. If delivery has failed, the operators cannot collect all of the meter reading data from smart meters. Therefore, smart metering services are required to provide reliable information delivery, such as meter reading data and control messages described in the scenarios of clause 7.

8.3 Minimal time delay

In the case of on-demand remote meter reading, a meter reading request and response should be delivered to the requesting entity with a minimal time delay.

8.4 Real-time delivery of meter reading data

The demands of smart metering operators require real-time responses of interconnected consumer premises equipment. Additionally, meter reading data are required to arrive at the metering operators in real-time (e.g., step 3 of scenarios I and II). Smart metering services are required to guarantee real-time data exchange.

8.5 Bidirectional communication between meters and operators

Meter reading data are delivered from smart meters to smart metering operators as shown in step 3 of scenarios I and II, and step 2 of Scenario III. Control messages are also sent to and from the operators and the smart meters (e.g., in step 5 of Scenario III and step 3 of Scenario IV). In order to achieve this, smart metering services are required to support bidirectional communication between meters and operators.

8.6 Security support including the authorization of operator and data confidentiality

Operators need to control meters so that the meters take action in accordance with their instructions. In this case, the operators should be authorized by smart meters to take actions to secure against unauthorized access (step 2 of Scenario II).

As metering information may include personal information, it should not be revealed to unauthenticated third parties. Therefore, it is required to secure metering information from access by unauthenticated third parties as well as to support the integrity check of the data exchanged.

8.7 Authentication of smart meters

Smart meters should be authenticated by smart metering applications. Authentication of smart metering devices can be achieved directly by smart metering applications, or by the authenticated smart metering gateways.

8.8 Meter reading data processing

Both operators and smart meters should be able to perform data processing on meter reading, such as filtering, validation and aggregation. In some cases, data mining processing is necessary, for example, analysing patterns and predicting some events.

8.9 Monitoring and management of smart meters

Smart meters or smart metering gateways should be monitored proactively in order to attempt to prevent and to correct errors. In addition, the following management capabilities should be supported from the network side:

– secure software and firmware provisioning

– configuration management

– auto-configuration functions for smart meter area networks

During the operation of smart metering services, smart metering applications should be able to specify a regular reporting schedule (Scenario I) for specific parameters and specific metering devices. Also, smart metering applications should be able to modify the value of the requested time period. Smart metering applications should be able to change the tariff configuration of smart meters when needed.

9 USN capabilities for smart metering services

9.1 Time synchronization

USN is required to support accurate and secure time synchronization among sensor nodes, sensor network gateways and smart metering applications.

9.2 Reliable transmission

Smart metering requires reliability of metering information delivery to ensure correct results. Therefore, a USN is recommended to guarantee the reliable transmission of measurement data and message delivery.

9.3 Scalability

New smart meters (sensor nodes) can be added to, or one of the existing meters (sensor nodes) can be removed from an existing smart metering group (sensor networks). Such a change should not degrade the performance of the smart metering service. Therefore, sensor networks in USNs are required to support scalability.

9.4 Mobility support

In some cases, a consumer can change the location of home appliances. This change may require changes in a meters' position in the intra-sensor network or inter-sensor networks. Therefore, sensor networks in USNs are recommended to support the mobility of smart meters (sensor nodes).

9.5 Delivery latency

Meter reading data should be delivered with a minimal time delay or within a pre-set time. Therefore, a USN is recommended to guarantee data delivery with a minimal time delay or within a pre-set time.

9.6 Fault detection and recovery

Link failure between wireless nodes is possible due to the characteristics of wireless transmission. Such failure can present negative effects on the reliability and delivery latency of smart metering services. A USN is recommended to detect link failures and recover from such failures.

9.7 Security supporting confidentiality, integrity check, authorization and authentication

Meter reading data and messages may be transferred to/from operators and meters by hop-by-hop transmission amongst sensor nodes. Therefore, hop-by-hop security among sensor nodes is recommended to be implemented and a USN is recommended to provide an integrity check of the data exchanged. In addition, the USN is recommended to provide smart meters with authorization capabilities for access to smart meters of smart metering operators.

Smart meters should be authenticated by smart metering applications directly, or through a smart metering gateway. Therefore, a USN is recommended to provide an authentication capability to smart metering applications for the authentication of smart meters (sensor nodes).

9.8 Connectivity

Meter reading data, obtained from each sensor node, are collected by a gateway, and then the collected meter reading data are delivered to smart metering operators through transport networks. In addition, control messages of the operators are also delivered to each sensor node. In order to achieve this, it is recommended that connectivity be guaranteed amongst sensor nodes, between sensor nodes and the gateway, and between the gateway and outer networks.

9.9 Unicasting and multicasting

A unicast networking service is required to be supported amongst sensor nodes so that meter reading data are delivered to smart metering operators. Furthermore, control messages of the metering operators should be delivered to all sensor nodes. Therefore, multicast networking support is also required.

9.10 Data aggregation

Meter reading data from smart meters is collected and aggregated at a smart meter (sensor node) or at smart metering gateways, prior to data transfer to smart metering operators. Therefore, USN sensor nodes and gateways are required to support data aggregation.

9.11 Distributed processing

If there are a number of smart meters (sensor nodes) in a smart home or building, a large number of meter reading data sets may be sent at the same time to a smart metering operator. To prevent node and service failure because of burst data centralization, distributed processing is required.

9.12 Monitoring and management of sensor nodes

Smart meters (sensor nodes) or smart metering gateways (sensor network gateways) are recommended to be monitored and managed proactively in order to attempt to prevent and correct errors including secure software and firmware provisioning and configuration management, as well as, auto-configuration functions for sensor networks.

Smart metering applications may also require a change of parameters in the smart meter for regularly scheduled reporting, as well as for changing tariff configuration. For satisfying this requirement, a USN is recommended to support the remote configuration setting and re-setting of sensor nodes acting as smart meters.

Bibliography

[b-ETSI TR 102 691] ETSI TR 102 691 v1.1.1 (2010), *Machine-to-Machine communications (M2M); Smart Metering Use Cases*.

[b-Khalifa] T. Khalifa, Naik, K. and Nayak, A. (2011), *A Survey of Communication Protocols for Automatic Meter Reading Applications*, Communications Surveys and Tutorials, Vol. 13, No. 2; pp.168-182, IEEE.

[b-NIST] NIST SP-1108 (2010), *Framework and Roadmap for Smart Grid Interoperability Standards*, Release 1.0, NIST. <http://www.nist.gov/public_affairs/releases/upload/smartgrid_interoperability_final.pdf>


-00.04
+10.05
+18.01
+27.15
+32.05
21°C
zone 1
zone 2
zone 3
A
B
C
D
E
F
G

Y.4252/Y.2064

Energy saving using smart objects in home networks

Energy saving using smart objects in home networks

Summary

Recommendation ITU-T Y.2064 describes requirements and capabilities for saving energy by using smart objects in home networks. It also presents the functional architecture of key components for saving energy through home/building automation.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T Y.2064	2014-01-13	13	11.1002/1000/12071-en

Keywords

Energy saving, home network, smart object.

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

Table of Contents


Home Control

Recommendation ITU-T Y.4252/Y.2064

Energy saving using smart objects in home networks

1 Scope

This Recommendation describes requirements and capabilities for saving energy by using smart objects in home networks. It develops the functional architecture of key components for saving energy through home/building automation. This Recommendation covers the following:

- general overview for saving energy by using smart objects in home networks
- requirements and capabilities for saving energy by using smart objects in home networks
- functional architecture for saving energy by using smart objects in home networks.

This Recommendation considers the fixed home environment such as residential buildings, and it also considers aspects of the mobile environment relating to the home such as networked electric vehicles (EVs) which support ubiquitous networking among smart objects.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T X.1111] Recommendation ITU-T X.1111 (2007), *Framework of security technologies for home network.*

[ITU-T Y.2002] Recommendation ITU-T Y.2002 (2009), *Overview of ubiquitous networking and of its support in NGN*.

[ITU-T Y.2060] Recommendation ITU-T Y.2060 (2012), *Overview of the Internet of things*.

[ITU-T Y.2062] Recommendation ITU-T Y.2062 (2012), *Framework of object-to-object communication for ubiquitous networking in next generation networks*.

[ITU-T Y.2281] Recommendation ITU-T Y.2281 (2011), *Framework of networked vehicle services and applications using NGN*.

[ITU-T Y.2291] Recommendation ITU-T Y.2291 (2011), *Architectural overview of next generation home networks*.

[ITU-T Y.2701] Recommendation ITU-T Y.2701 (2007), *Security requirements for NGN release 1*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 context [ITU-T Y.2002]: The information that can be used to characterize the environment of a user.

NOTE – Context information may include where the user is, what resources (devices, access points, noise level, bandwidth, etc.) are near the user, at what time the user is moving, interaction history between person and objects, etc. According to specific applications, context information can be updated.

3.1.2 object [ITU-T Y.2002]: An intrinsic representation of an entity that is described at an appropriate level of abstraction in terms of its attributes and functions.

NOTE 1 – An object is characterized by its behaviour. An object is distinct from any other object. An object interacts with its environment including other objects at its interaction points. An object is informally said to perform functions and offer services (an object which makes a function available is said to offer a service). For modelling purposes, these functions and services are specified in terms of the behaviour of the object and of its interfaces. An object can perform more than one function. A function can be performed by the cooperation of several objects.

NOTE 2 – Objects include terminal devices (e.g., used by a person to access the network such as mobile phones, Personal computers, etc.), remote monitoring devices (e.g., cameras, sensors, etc.), information devices (e.g., content delivery server), products, contents, and resources.

3.1.3 ubiquitous networking [ITU-T Y.2002]: The ability for persons and/or devices to access services and communicate while minimizing technical restrictions regarding where, when and how these services are accessed, in the context of the service(s) subscribed to.

NOTE – Although technical restrictions to access services and communicate may be minimized, other constraints such as regulatory, national, provider and environmental constraints may impose further restrictions.

3.2 Terms defined in this Recommendation

This Recommendation defines the following term:

3.2.1 smart object: An object which is aware of its characteristics, context and situation. It shares and processes information, such as its identity, current location, physical properties and the information it senses from its surroundings, while performing object-to-object communications.

NOTE – [ITU-T Y.2002] and [ITU-T Y.2062] provide details of object-to-object communication.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

BAN	Building Area Network
BAS	Building Automation System
BEMS	Building Energy Management System
DR	Demand Response
EMS	Energy Management System
ESI	Energy Service Interface
EV	Electric Vehicle
FMS	Facility Management System
GHG	Greenhouse Gas
H2G	Home to Grid
HAN	Home Area Network
HEG	Home Energy Gateway
HVAC	Heating, Ventilating and Air Conditioning
ICT	Information and Communication Technology

IoT	Internet of Things
IP	Internet Protocol
IT	Information Technology
ITS	Intelligent Transport System
QoS	Quality of Service
V2G	Vehicle to Grid
V2I	Vehicle to Infrastructure
V2V	Vehicle to Vehicle

5 Conventions

In this Recommendation:

The keywords "is required to" indicate a requirement which must be strictly followed and from which no deviation is permitted, if conformance to this Recommendation is to be claimed.

The keywords "is prohibited from" indicate a requirement which must be strictly followed and from which no deviation is permitted, if conformance to this Recommendation is to be claimed.

The keywords "is recommended" indicate a requirement which is recommended but which is not absolutely required. Thus, this requirement need not be present to claim conformance.

The keywords "is not recommended" indicate a requirement which is not recommended but which is not specifically prohibited. Thus, conformance with this Recommendation can still be claimed even if this requirement is present.

The keywords "can optionally" indicate an optional requirement which is permissible, without implying any sense of being recommended. This term is not intended to imply that the vendor's implementation must provide the option, and the feature can be optionally enabled by the network operator/service provider. Rather, it means the vendor may optionally provide the feature and still claim conformance with this Recommendation.

6 Overview of energy saving using smart objects in home networks

Home networks [ITU-T Y.2291] provide packet-based transfers (in particular support of Internet protocol (IP)) and user access to a wide range of services and applications in a seamless environment, using multiple broadband and quality of service (QoS)-enabled transport technologies. They support automatic discovery and management of fixed and mobile terminals to the home network.

Energy issues such as low-power consumption are central to the ubiquitous network environment [ITU-T Y.2002] of the Internet of things (IoT) [ITU-T Y.2060]. For devices which constrain and minimize energy consumption, there are significant efforts being made to implement autonomous smart objects and to develop their applications which relate to energy issues. Home networks play an important role in interconnecting smart objects with the Internet.

In the ubiquitous network environment, everything is becoming connected. Therefore, a network has evolved from being primarily a source of information to being the platform for all types of applications. Smart and connected communities using the "connecting to anything" capability of ubiquitous networking [ITU-T Y.2002] play an important role in applications and services. Therefore, one of the core applications for interdisciplinary fusion services that this Recommendation focuses on is the requirements and key functionalities for saving energy using smart objects through combining information technology (IT) and other technologies (e.g., from energy related industries) in home networks.

Figure 1 depicts energy saving using smart objects; it shows a diagram of home networks with examples of smart objects. In the fixed home environment (e.g., residential buildings), objects such as energy saving systems, smart meters and home automation controllers, are used for energy management. In the mobile environment (e.g., networked EVs), objects such as devices for navigation and/or safety are used for saving energy.

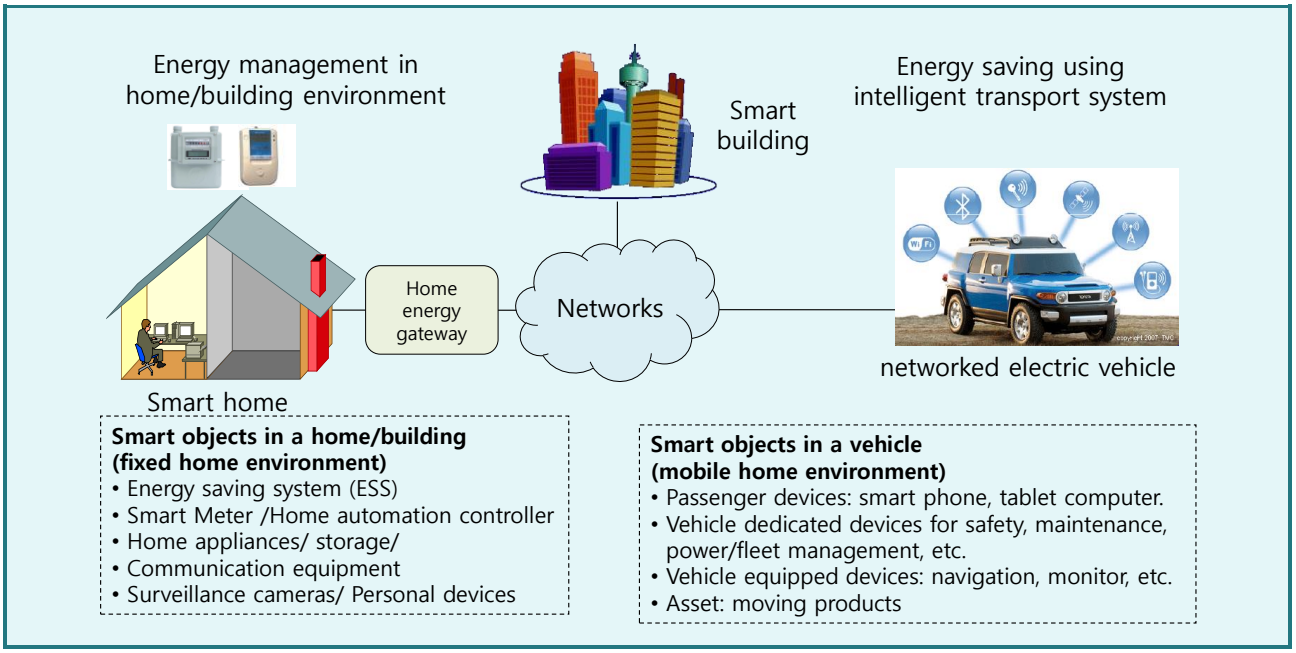


Figure 1 –Energy saving using smart objects in home networks

"Smart homes/buildings" represent a suite of technologies used to make the design, construction and operation of buildings more efficient; they are applicable to both existing and new-build properties. Building automation systems (BAS) including building management systems are important components which run heating and cooling systems. Data from these systems can be used to identify additional opportunities for efficiency improvements. As information and communication technology (ICT) applications become more sophisticated, the range of building automation system functions will expand.

Building system engineering supported by intelligent and networked room and building controllers (lighting, sun protection, heating, ventilation and air conditioning (HVAC), as well as other building engineering systems) contribute significantly to conservative and requirement-based energy use.

Various concepts and approaches are possible in the optimization of energy efficiency in homes/buildings; in this context, the use of intelligent building control provides a proven and interesting alternative or addition that is clearly set apart by its convincing cost-benefit ratio.

Automotive transport represents one of the main sources of greenhouse gas (GHG) emissions, but the wide-spread availability of ultra-high-speed broadband access, with ubiquitous provision of services, would enable multiple tasks to be achieved simultaneously with minimum power consumption. ICTs can be applied to transport through the development of intelligent transport systems (ITS). Although the main focus of ITS is on the safety, management and efficiency of transport systems, ITS can also be used to reduce their energy consumption.

7 Requirements and capabilities for saving energy using smart objects

7.1 High-level requirements

In home networks, there are the following high-level requirements for saving energy using smart objects:

- Home networks are required to support networking using various communication protocols and interfaces.
- Home networks are required to deliver energy consumption information and other information for customers, on demand.
- Home networks are required to store the most recent energy-consumption readings.
- Upon the request of authorized persons/organizations, home networks are required to provide periodic energy monitoring information.
- Home networks are required to support energy-related details/changes when they are being managed remotely (e.g., meter status, activation/de-activation capability, error messaging, fraud detection).
- Home networks are required to support any tariff changes that have been made remotely through interaction with the utility company.

7.2 Requirements of key components in home networks

7.2.1 Requirements for home energy gateway

The home energy gateway (HEG) offers features for collecting power consumption data over home networks with multiple access interfaces from appliances, controlling power activation, and communicating with utility networks, as well as communication network operators.

NOTE – HEG is a set of functions consisting of gateway functions and functions required for Smart Grid applications to control and manage Smart Grid services in customer premises.

The requirements for the HEG are as follows:

- An HEG is required to support various kinds of communication interfaces:
  - multiple communication access interfaces (e.g., WiFi, ZigBee) in a home;
  - a communication interface for remote control/readout outside a home;
  - the interconnection between home and utility companies/ communication network operators.
- The HEG is required to support authentication and authorization of all equipment in a home, including an electric vehicle (EV).
- The HEG is required to support the monitoring of energy consumption of home appliances, the analysing of information and the learning of usage patterns:
  - collecting power consumption data from various appliances;
  - the management of electric power load in the house, including EV energy consumption control based on the EV charging status.
- The HEG is required to detect and raise the alarm of an abnormal/emergency situation.
- The HEG is required to support charging/billing control.
- The HEG is required to support display control.

- The HEG is recommended to support device control and management:
  - control menus to control appliances
  - activation and deactivation of appliances.

7.2.2 Requirements for electric vehicles

For saving energy, the requirements for EVs are as follows:

- The networks for EVs are required to support vehicle communications [ITU-T Y.2281] including vehicle to vehicle (V2V), vehicle to infrastructure (V2I), vehicle to grid (V2G) in mobile environments.
- The networks for EVs are required to support charging/billing:
  - Smart charging: the vehicle charges its battery at a variable rate. It always tries to balance the vehicle owner's needs, the battery's demands and the grid operator's wishes.
  - Smart billing: it makes paying for the energy used convenient to the car owner and it enables advanced services and new business models.
- The networks for EVs are required to support interworking with several grid interfaces, e.g., V2G and home to grid (H2G).
- The networks for EVs are required to support the remote management of battery charging.
- The networks for EVs are required to support mobile networking and services concerning a networked EV.
- The networks for EVs are required to support new applications' combined telecom services and energy-related services.

7.3 Required capabilities for saving energy

The following are required capabilities for saving energy:

- energy-consumption monitoring and control
- intelligent power consumption monitoring application
- device virtualization environment
- web-based information processing
- personalized service creation
- secure and privileged access
- identity-based user management
- IP connectivity.

The following describes essential capabilities for customers in their home/building:

- home/building energy automation;
- energy management (including control and logging);
- renewable energy management (store, manage and integration);
- support electric vehicles charging;
- manage home appliances;
- home area networking and management;
- supports load control: a load control device (e.g., smart appliance, pool pump controller, energy management system (EMS), etc.) has the capability to reduce the peak power consumption of the equipment under its control.

8 Functional architecture for energy saving using smart objects

8.1 Configuration of home networks

There are important components of home networks. A residential building is comprised of an HEG to control the electricity, home appliances with energy-measuring sensors, smart meters (e.g., electricity meter, gas meter, etc.), display and a set-top box. A non-residential building is comprised of a building energy management system (BEMS) for managing building facility and component operation, focused especially on energy, BAS, HVAC, facility management system (FMS), and in-building infrastructure with sensing, metering and controlling components. Additionally, an EV, charging system (i.e., storage battery) and a parking system are considered for both homes and buildings.

Figure 2 shows the physical configuration of home/building networks with outdoor networks. Specifically, an energy service interface (ESI) in home/building networks refers to the interface between the home/building domain and outdoor networks. This interface can be a simple logical device within an HEG in a home/building network.

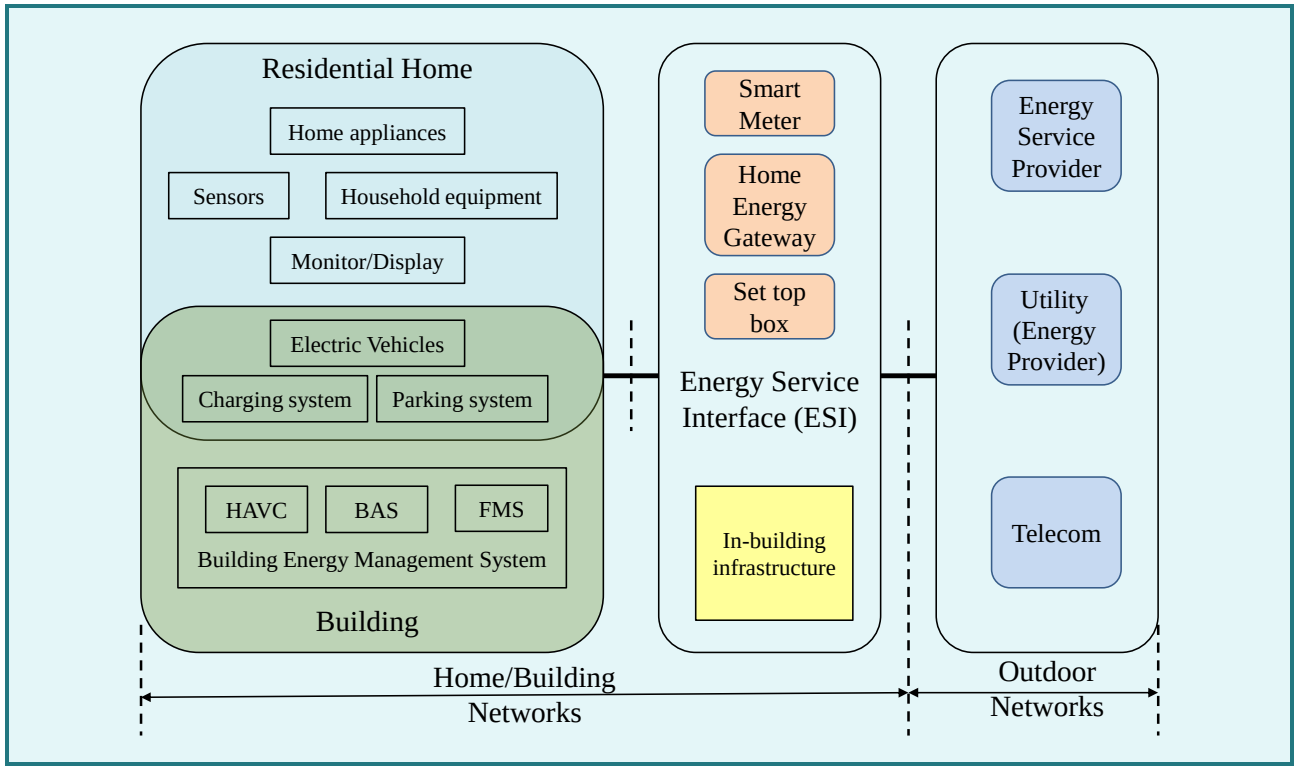


Figure 2 – Physical configuration of home networks (residential building and non-residential building) with outdoor networks

An HEG has the following functionalities:

- support of multiple communication interfaces (e.g., WiFi, ZigBee, etc.);
- interconnecting between home and utility companies/communication network operators;
- authentication and authorization of all home equipment including an electric vehicle;
- monitoring of energy consumption of home appliances, analysing information and learning usage patterns;

- detecting and raising the alarm of abnormal/emergency situations;
- charging/billing control;
- display control.

8.2 Functional architecture in home/building for saving energy

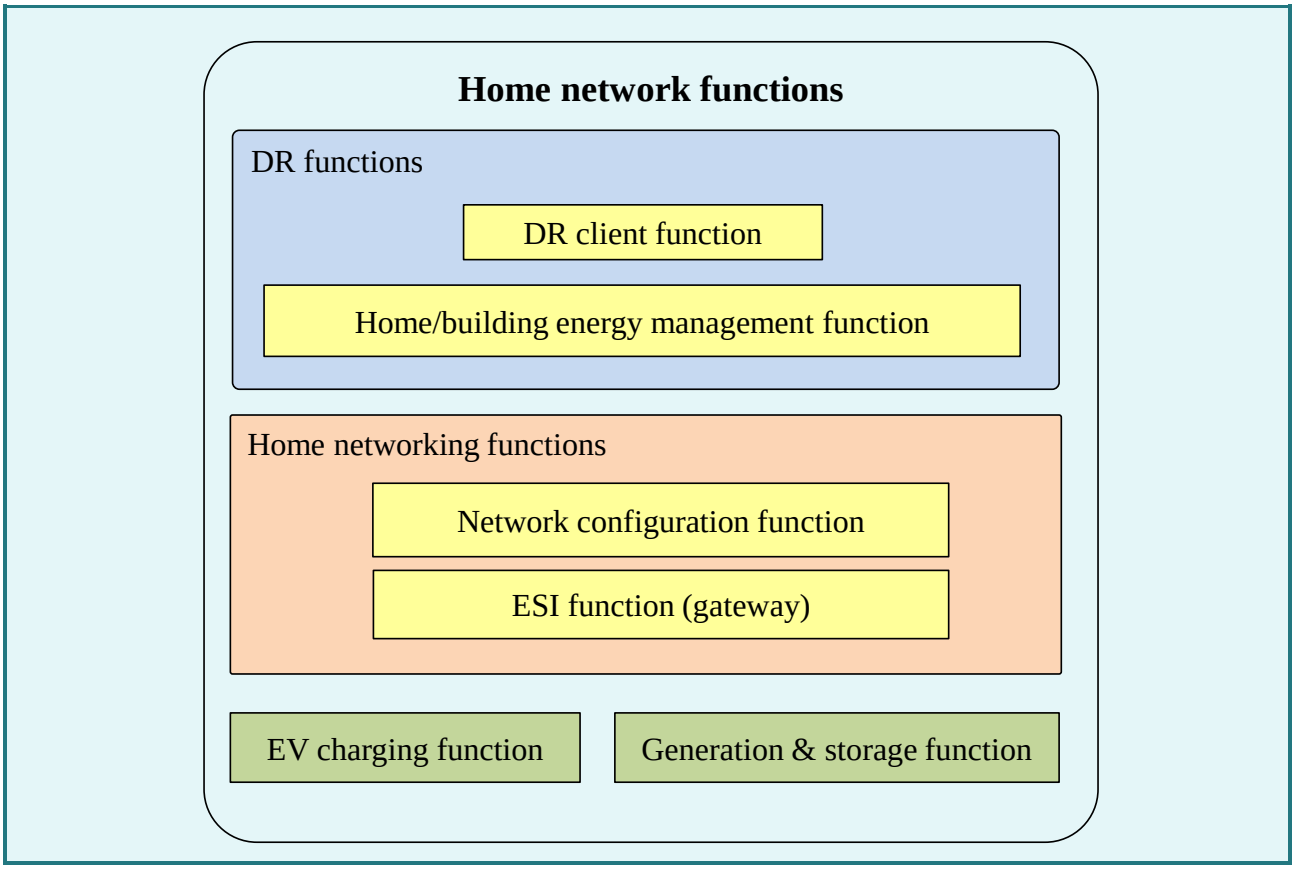


Figure 3 – Functional architecture for home network functions to support energy saving

A home network function group consists of the following functions: energy demand response, home/building energy management and automation, local energy generation and storage, and EV charging. It interacts with a DR application for dynamic pricing information, it controls energy usage of home appliances and in-building equipment.

NOTE – A home network function group can also interact with energy control functions which are located in outdoor networks for distribution capacity management and two-way energy transmission. It is out of the scope of this Recommendation.

- **DR functions***:* This function group covers all operations in the home/building domain for energy saving applications where the users interact with the service provider domain connected with outdoor networks.

DR client function: This function supports subscription to the service and dynamic pricing information and enables the management of energy consumption per users' needs while supporting metering information retrieval.

Home/building energy management function: This function monitors the energy consumption of appliances and dynamic pricing information, in order to control appliances, and generation and storage devices, through the interactions with users in the home/building domain. This function also provides notification of power outage to utility companies, and responds to mitigation and recovery signals during a scheduled or unscheduled energy outage.

- **Home networking functions***:* This function group provides a communications function in the home/building through a home area network (HAN) and building area network (BAN), respectively. These networks interconnect all appliances and equipment, EMS, EV charging stations, generation and storage facilities, and meters.
  - Network configuration function: This function manages equipment that joins and leaves the network. It supports by authenticating the members, authorizing the operations they could perform, and the information they could send and receive, and maintains encryption key information.
  - ESI functions: This function gates information in/out of HAN/BAN like a firewall and performs other functions which provide bi-directional logical interface that supports the communication of information between energy automation and other entities located in outdoor networks.
- **EV charging function***:* This supports managing the charging rate and billing information of an EV to support energy saving applications.
- **Generation and storage function**: This function manages the facilities for local energy sources. It interacts with the EMS for the switching of power for consumption in the home/building domain. It also interacts with the home/building energy management function to efficiently manage energy usage.

NOTE – This Recommendation does not consider all functions which support energy savings for various home environments. In particular, any additional functions for networked EVs can be added.

9 Security considerations

Security considerations for home networks should be in accordance with the security requirements given in [ITU-T X.1111] and [ITU-T Y.2701].

Appendix I

Use cases for saving energy through home automation and building energy management

(This appendix does not form an integral part of this Recommendation.)

This appendix describes various use cases for saving energy through home automation and building energy management.

I.1 Energy saving through home automation

For saving energy, the more detailed cases in home networking environments are considered. The following shows more specific cases for saving energy through home automation [b-AIM project].

- **Case 1 for residential users**

  Local users can interact with the system to perform a set of operations for intelligent power management services for autonomous energy preservation. The operations include the monitoring of energy use, the personalization of energy use and gateway maintenance.
- **Case 2 for utility companies**

  For a metering service for energy planning;, some components allow the metering system to exchange information with the power distribution network operator.
- **Case 3 for communications network operators**

  For remote monitoring and management, the users can monitor and control the power consumption of their homes remotely while moving outdoors.

  NOTE – In this Recommendation, two cases (Case 1 and Case 3) are considered for saving energy using smart objects.

I.2 Energy saving through building energy management

BEMS is a system technology for managing the building facility and component operation focused especially on energy. To improve energy efficiency, it is anticipated that various energy saving services are provided in the building domain [b-ITU-T FG-Smart]. The following is a list of use cases for saving energy using BEMS.

- **Dynamic pricing and metering information transfer**: To enhance the efficiency of electrical power usage and provide detailed energy usage in a building. The BEMS monitors and manages electric usage for building operation and maintenance based on the input dynamic pricing information and the usage information provided.
- **Demand response message transfer**: After receiving the message to reduce energy demand by the consumer when reaching peak demand, the BEMS is able to control electricity usage in the building based on a BEMS energy management algorithm and policy.
- **EV information transfer and EV's electric charge and discharge**: Based on the information of an EV's state such as its storage state or operating schedule, the BEMS is able to control an EV's electric charge and discharge in order to optimize energy usage by the EV.

Bibliography

[b-ITU-T FG-Smart] Smart-O-31Rev.7 (2011), *"Use cases for Smart Grid"*.

[b-AIM project] AIM Deliverable 4.1.1 (2009), *Use-cases design report*.


HOME CONTROL
SECURITY
MULTIMEDIA
LIGHTS
CLIMATE
ELECTRONICS
ROOMS

5 Frameworks, Architectures and Protocols

Y.4400/Y.2063

Framework of the web of things

Framework of the web of things

Summary

Recommendation ITU-T Y.2063 provides a framework of the web of things (WoT). As the use of various devices has become so widespread, it is difficult to access data on these devices in a unified way. The WoT allows physical devices to be accessed as resources of both the web and services/applications based upon a web-based service environment, as well as through legacy telecommunications.

This Recommendation describes the overview of the WoT and identifies the requirements to support the WoT. In addition, this Recommendation specifies the functional architecture including a deployment model for the WoT.

History

Edition	Recommendation	Approval	Study Group
1.0	ITU-T Y.2063	2012-07-29	13

Keywords

Web, web of things, WoT.

Table of Contents


00101
11001

Recommendation ITU-T Y.4400/Y.2063

Framework of the web of things

1 Scope

This Recommendation provides a framework of the web of things (WoT). The Recommendation covers the followings:

- overview of the WoT
- requirements to support the WoT
- deployment models of the WoT
- functional architecture for the WoT.

This Recommendation demonstrates how physical devices can interact with web resources. This Recommendation also includes WoT use cases in Appendix I and information flows in Appendix II. The detailed web technology including the semantics and ontology is beyond the scope of this Recommendation.

NOTE – This Recommendation only addresses physical devices within a broad scope of "things" [ITU-T Y.2060].

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T Y.2002] Recommendation ITU-T Y.2002 (2009), *Overview of ubiquitous networking and of its support in NGN*.

[ITU-T Y.2060] Recommendation ITU-T Y.2060 (2012), *Overview of the Internet of things.*

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 client [b-W3C WACterms]: The role adopted by an application when it is retrieving and/or rendering resources or resource manifestations.

3.1.2 device [b-W3C dig loss]: An apparatus through which a user can perceive and interact with the web.

NOTE – In the IoT, a piece of equipment with the mandatory capabilities of communication and the optional capabilities of sensing, actuation, data capture, data storage and data processing [ITU-T Y.2060].

3.1.3 Internet of things (IoT) [ITU-T Y.2060]: A global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies.

NOTE 1 – Through the exploitation of identification, data capture, processing and communication capabilities, the IoT makes full use of things to offer services to all kinds of applications, whilst ensuring that security and privacy requirements are fulfilled.

NOTE 2 – From a broad perspective, the IoT can be perceived as a vision with technological and societal implications.

3.1.4 resource [b-IETF RFC 3986]: The term "resource" is used in a general sense for whatever might be identified by a URI.

NOTE – Familiar examples include an electronic document, an image, a source of information with a consistent purpose (e.g., "today's weather report for Los Angeles"), a service (e.g., an HTTP-to-SMS gateway), and a collection of other resources. A resource is not necessarily accessible via the Internet; e.g., human beings, corporations, and bound books in a library can also be resources. Likewise, abstract concepts can be resources, such as the operators and operands of a mathematical equation, the types of a relationship (e.g., "parent" or "employee"), or numeric values (e.g., zero, one, and infinity).

3.1.5 server [b-W3C WACterms]: The role adopted by an application when it is supplying resources or resource manifestations.

3.1.6 the World Wide Web (WWW, or simply the web) [b-W3C web arch]: An information space in which the items of interest, referred to as resources, are identified by global identifiers called Uniform Resource Identifiers (URI).

3.1.7 thing [ITU-T Y.2060]: With regard to the Internet of things, this is an object of the physical world (physical things) or the information world (virtual things), which is capable of being identified and integrated into communication networks.

3.1.8 URI [b-IETF RFC 3986]: A simple and extensible means for identifying a resource.

3.1.9 user agent [b-W3C dig loss]: A client within a device that performs rendering. Browsers are examples of user agents, as are web robots that automatically traverse the web collecting information.

3.1.10 web resource [b-W3C WACterms]: A resource, identified by a URI, that is a member of the web core.

3.2 Term defined in this Recommendation

This Recommendation defines the following term:

3.2.1 Web of things (WoT): A way to realize the IoT where (physical and virtual) things are connected and controlled through the World Wide Web.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

FE	Functional Entity
HTTP	Hyper Text Transport Protocol
ID	Identifier
IoT	Internet of Things
JINI	Java Intelligent Network Infrastructure
NGN	Next Generation Network
REST	Representational State Transfer

UPnP	Universal Plug and Play
URI	Unique Resource Identifiers
WoT	Web of Things
WWW	World Wide Web

5 Conventions

In this Recommendation,

- The keywords "is required to" indicate a requirement which must be strictly followed and from which no deviation is permitted if conformance to this Recommendation is to be claimed.
- The keywords "can optionally" indicate an optional requirement which is permissible, without implying any sense of being recommended. This term is not intended to imply that the vendor's implementation must provide the option and the feature can be optionally enabled by the network operator/service provider. Rather, it means the vendor may optionally provide the feature and still claim conformance with the specification.

6 Overview of the web of things

From the perspective of the creation of applications, the development of applications that run on top of physical devices is a difficult process that requires expert knowledge and time. In this context, many efforts are being veered towards networking to devices. There are a number of solutions to expose the functionality of devices upon which to build applications; for example, JINI and UPnP are a set of open protocols for allowing devices to collaborate in a peer-to-peer fashion. However physical devices are still dedicated to particular systems/applications. They cannot be controlled and managed without using dedicated protocol and proprietary interfaces due to the following reasons:

- a lack of interoperability across open and proprietary platforms: there are many hardware platforms, operating systems, databases, middleware and applications.
- Many heterogeneous networks: they cannot exchange content and information easily.
- Different data type: All systems worldwide have their own data representation formats and it is difficult to ensure compatibility between them.

The Internet of things (IoT) tries to find a way interconnecting things based on interoperable information and communication technologies. Through the exploitation of identification, data capture, processing and communication capabilities, the IoT makes full use of things to offer services to all kinds of applications, whilst maintaining the required privacy [ITU-T Y.2060]. Although the WoT has a similar viewpoint to the IoT, the WoT is intended so that physical devices can be accessed as resources of the web and services/applications can be provided based upon a web-based service environment as well as legacy telecommunications.

The World Wide Web (WWW) is used as a platform to deliver services to an end-user, the web enables business entities and applications to intercommunicate openly with each other over a network. The web has program language independent properties, uses message driven communications and easily bounds to different transport protocols. As a result, web technology allows the exposure of physical devices as resources on the web using a WoT approach. Therefore, users can interact with the devices using web interfaces. The WoT can provide capabilities of device reusability, portability across several heterogeneous networks and accessibility based on web with web standards.

Figure 6-1 shows the general concept of WoT. The physical devices are mapping the services into the web and those are considered as web resources so that service developers and/or service providers can easily create web applications for the physical devices.

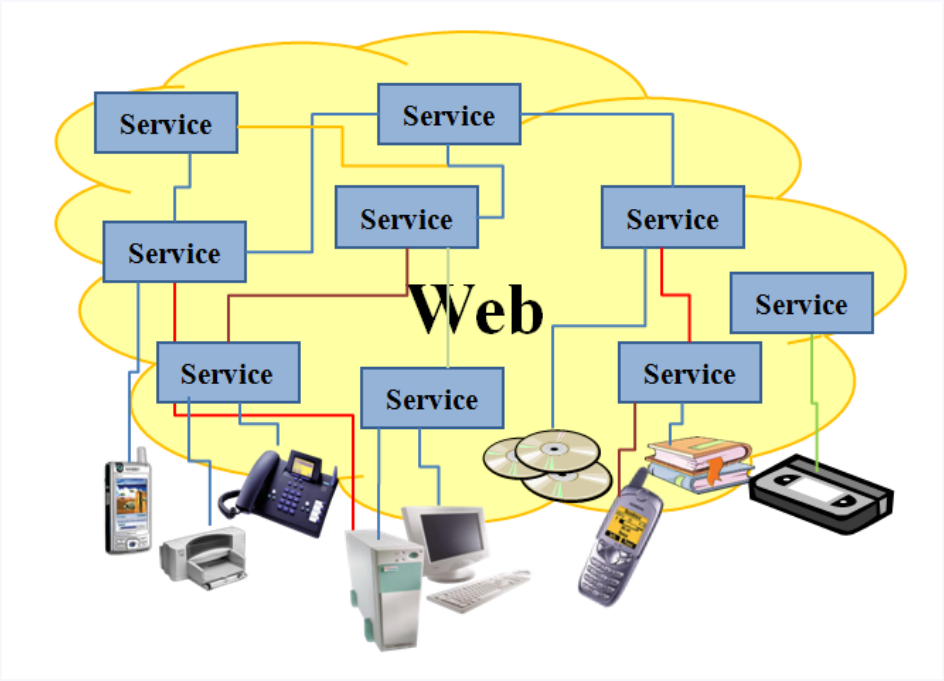


Figure 6-1 – General concept of the web of things

7 Requirements for the web of things

This clause specifies the requirements to support the WoT. The requirements are needed to consider general aspects and functional aspects.

7.1 General requirements for the WoT

The followings are the requirements in the general aspects:

- WoT is required to access physical devices.
  - ✓ The WoT user (e.g., service developer, service provider, applications and user agent, etc.) can access device capabilities on the web.
  - ✓ The service developer and the service provider can create new web services using web technologies. They do not need to know the technical details of a physical device such as a physical device's interfaces and their protocols. They do not waste time and human resources for the physical device application development.
- WoT is required to provide the means that make physical devices accessible to web resources.
- WoT is required to support interoperability among different networks and operating systems.
  - ✓ The user can use WoT services on the web regardless of networks and operating systems.
  - ✓ The service developer can create services across heterogeneous networks with different types of devices and different service providers.

- ✓ Services can be created regardless of the operating system and programming languages for devices.

- WoT is required to support location transparency.
  - ✓ Devices can be accessed by a user from anywhere on the network without knowing where the devices are located.
- WoT is required to support the compatibility between different data representation formats.

7.2 Functional requirements for the WoT

The following are the requirements for the functional aspects:

- The WoT is required to support service profile management to discover and register services through a web interface.
- The WoT is required to support service control for executing and managing WoT services.
- The WoT is required to support service composition for creating new WoT services.
- The WoT is required to support service access control for protection against unauthorized request/user/access.
- The WoT is required to support an agent that can make physical devices accessible to the web.
- The WoT is required to support resource management for supporting agent control, agent registration/deregistration and agent profile management.
- The WoT is required to support resource ID management for supporting mapping information between devices and agents.

8 Conceptual and deployment models of the web of things

8.1 Conceptual model

Figure 8-1 shows the conceptual model of the WoT. Applications can access and use physical devices on the web through a WoT broker or directly. A WoT broker has several agents which have adaptation capabilities to make a physical device's interface work with a web interface. Each agent is dedicated to a specific interface of the subnetwork (e.g., Wi-Fi, Zigbee and Bluetooth).

In a WoT environment, three services can be used. The characteristics of these three services in the WoT are as follows:

- WoT service: a service which has 1:1 mapping with services and/or functions in a physical device through the adaptor.

  NOTE – If WoT services are integrated with each other in a WoT broker, it is a composite WoT service.
- Mash-up service: combined services which integrate WoT services in a WoT broker with web services outside of the WoT broker.
- Web service: services that can be accessed on the web.

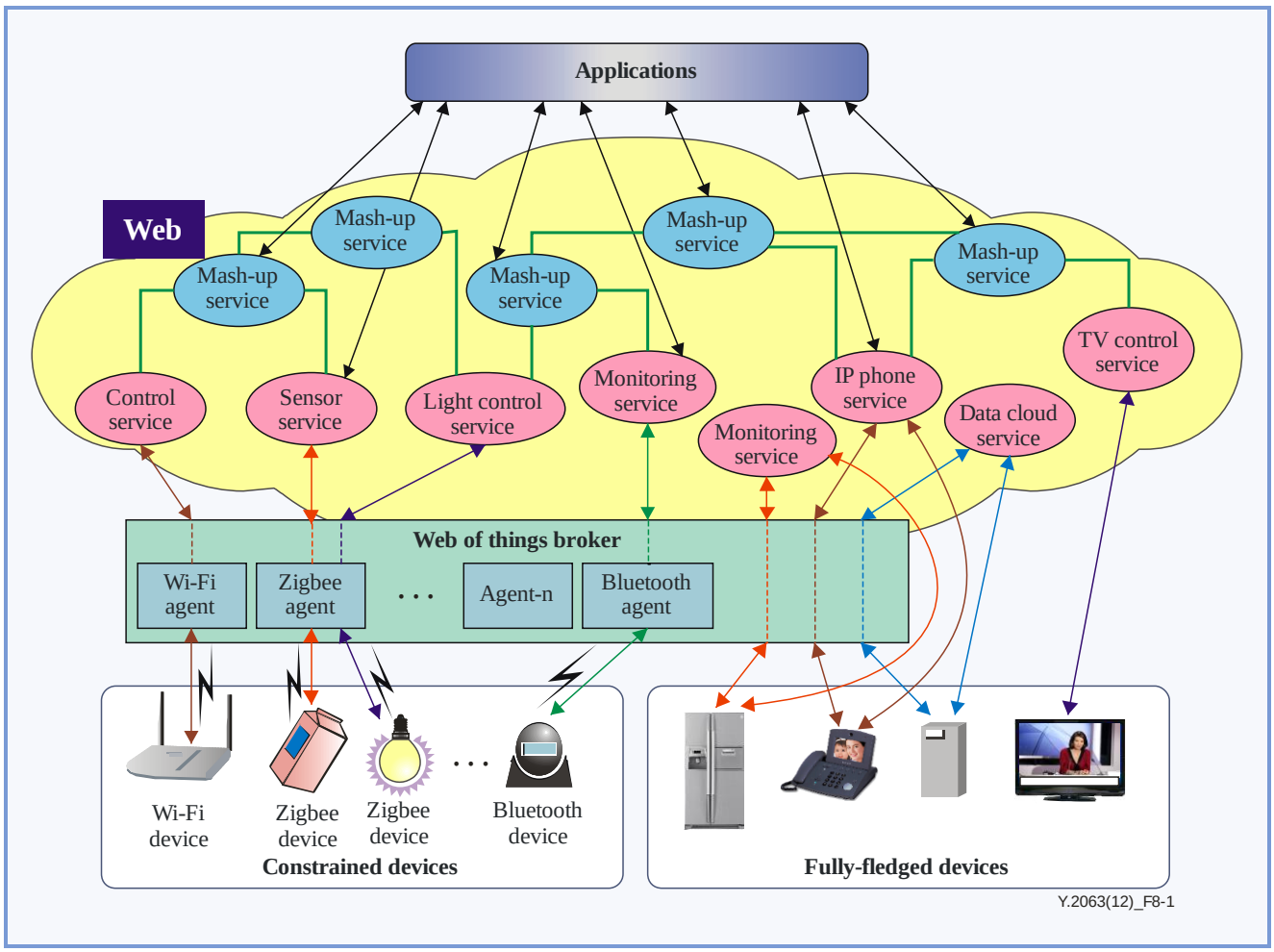


Figure 8-1 – Conceptual model of the web of things

8.1.1 Two types of devices

There are several kinds of physical devices in the physical world (e.g., Wi-Fi, Zigbee device, Bluetooth device, TV, phone, data server). Among them, some devices can neither connect to the Internet nor fully respect the web (or REST architectural style). Regarding the physical devices on the WoT, they are divided into two categories: constrained devices and fully-fledged devices.

Constrained device: A constrained device cannot connect to the Internet and it has no functionality of the web. The device interacts with an agent of the WoT broker.

Fully-fledged device: A fully-fledged device has the functionalities of the web. The device can interact not only with the WoT broker but also with the services on the web.

8.1.2 WoT Broker

The WoT broker has a role for integrating and exposing the devices to the web. This broker has responsibility for communicating between the user of the WoT (e.g., web clients, applications) and fully-fledged devices as well as constrained devices. In the case where a device is communicated with by dedicated software and proprietary interfaces, the device cannot be exposed and integrated on the web directly. The WoT broker enables the seamless integration of the device onto the web. The agent of the WoT broker has a role to control and communicate with physical devices. If a request is received from an application to access physical devices, the WoT broker adapts the request to a proprietary interface of the physical device through an agent.

8.2 Deployment models

8.2.1 WoT deployment model for fully-fledged devices

This deployment model represents the pure essence of physical devices accessing and consuming services in the web. In this model, each physical device can have a web server. Therefore each physical device and its capabilities can be linked to the web and it can be discovered by user agents or applications without any other external element (e.g., WoT broker). Additionally those devices can be linked to the web through the WoT broker as necessary. For example, some capabilities of a fully-fledged device can be needed to composite with other physical devices within the WoT broker and/or a fully-fledged device needs to help some functionality of the WoT broker.

In Figure 8-2, there are four parts according to their communicating methods. Basically each part can communicate using the HTTP protocol. However the WoT broker can optionally communicate with devices using each proprietary interface.

The WoT broker has the role of intermediation between physical devices and applications. The WoT broker can create a new service using the services of each physical device. From the perspective of applications, applications can access the physical devices on the web directly or through the WoT broker. In addition, applications can access mash-up services which are combined or aggregated with existing services on physical devices.

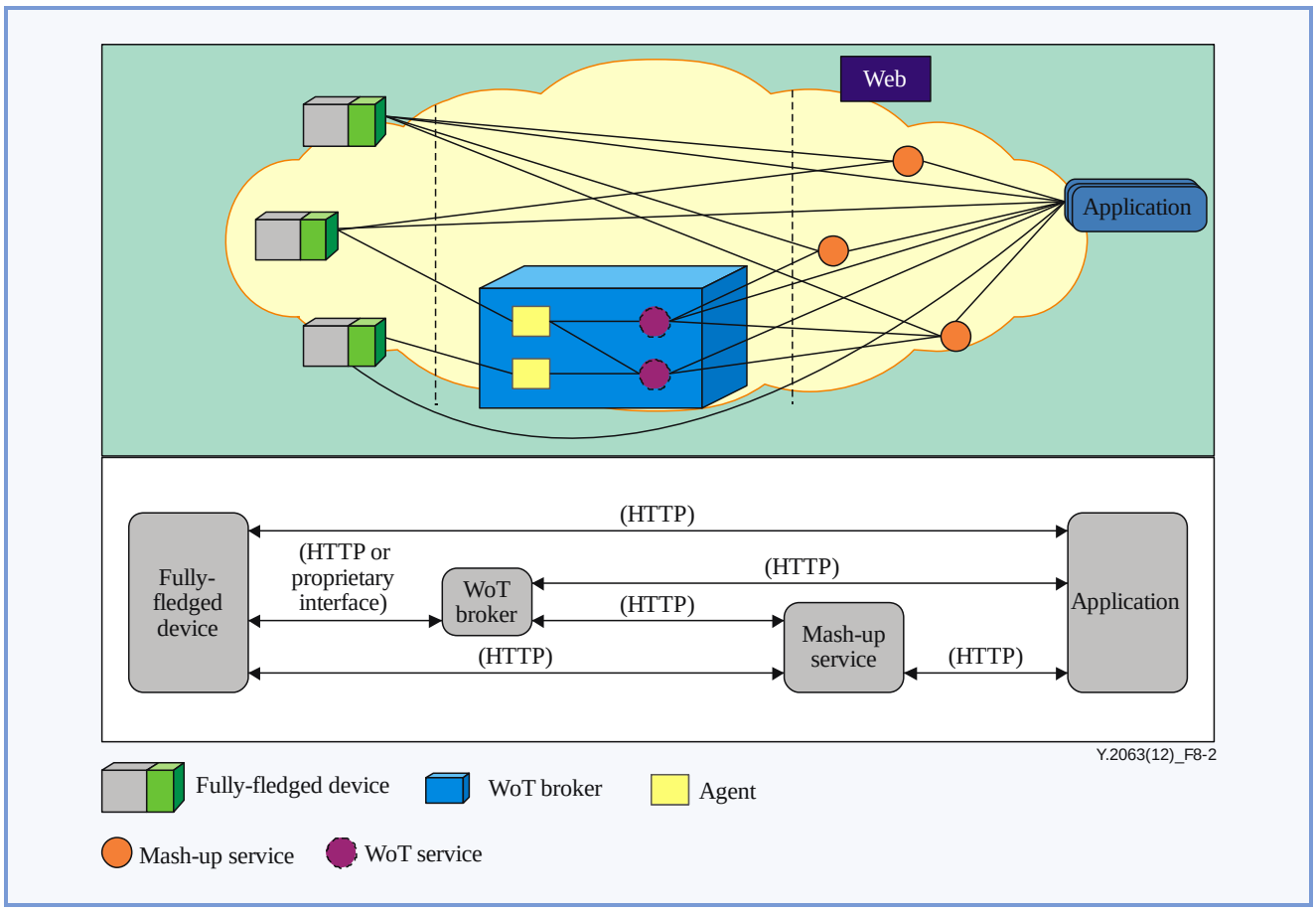


Figure 8-2 – WoT deployment model for fully-fledged devices

8.2.2 WoT deployment model for constrained devices

The key factor of this model is how applications can use constrained devices. This deployment model is implemented by the WoT broker, which performs as an intermediation between physical devices and applications. The resources of devices can be exposed on the web through the WoT broker.

In Figure 8-3, there are also four parts according to their communicating methods like the model for fully-fledged devices. However each device communicates with the WoT broker using their proprietary interface. Applications can access mash-up services in a similar way to the model for fully-fledged devices.

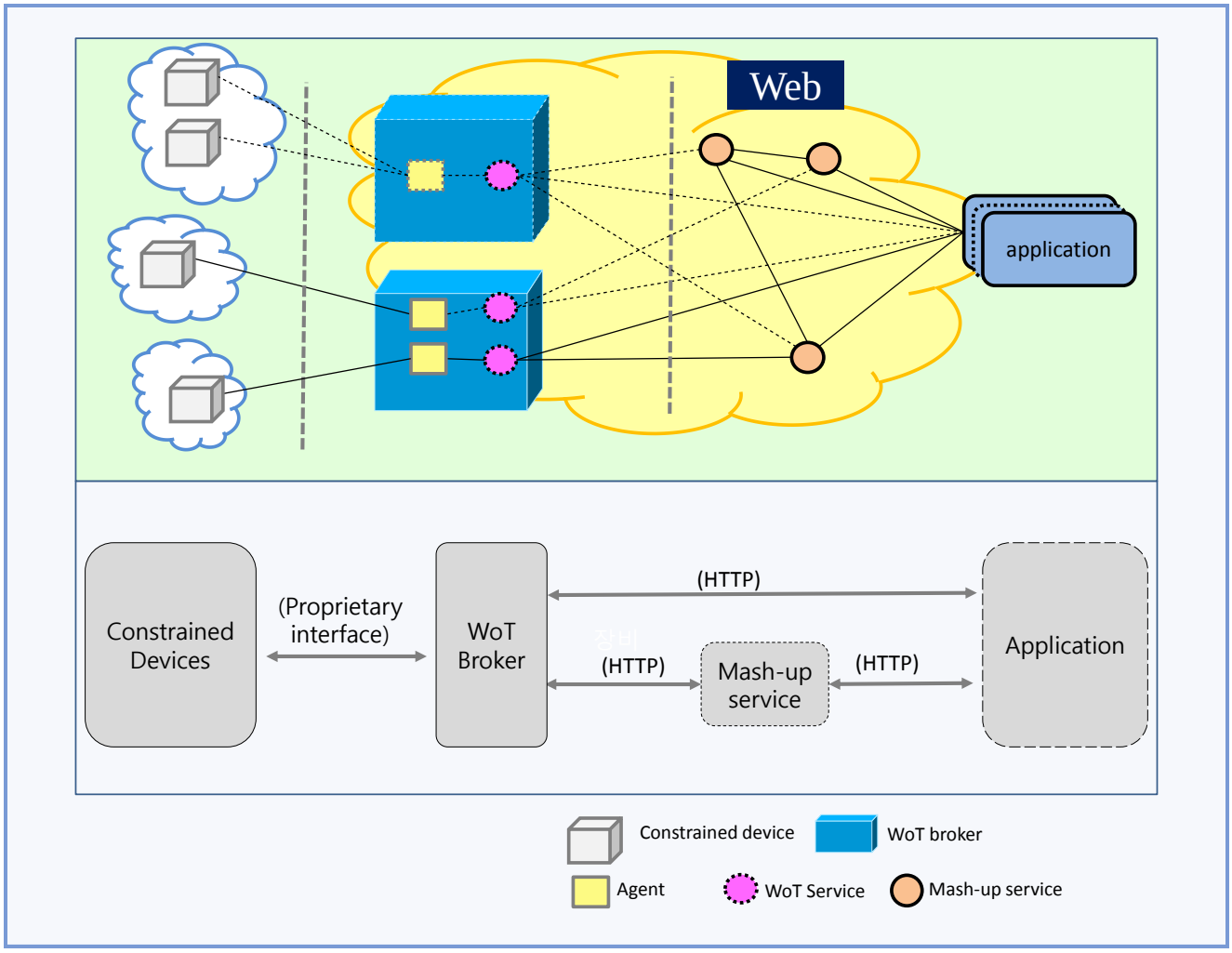


Figure 8-3 – WoT deployment model for constrained devices

9 Functional architecture for the web of things

9.1 Overview of the WoT architecture

The WoT architecture is divided into three layers: service layer, adaptation layer and physical layer.

- Service layer
  - ✓ The service layer provides a common function for service capabilities. It is the entity responsible for making a service available and for managing it.

- Adaption layer
  - ✓ The adaption layer is where agents reside. In this layer, each agent interacts with physical devices for the translation of different protocols and message formats. According to the type of physical device (e.g., Bluetooth, Zigbee) the correspondence agent in the adaptation layer can be connected. Also, all the resource management on the agent is conducted in this layer.
- Physical layer
  - ✓ The physical devices are in the physical layer. All constrained devices can be accessed by an agent in the adaptation layer. Also, it is possible that a fully-fledged device in the physical layer can be directly accessed by the mash-up service or applications.

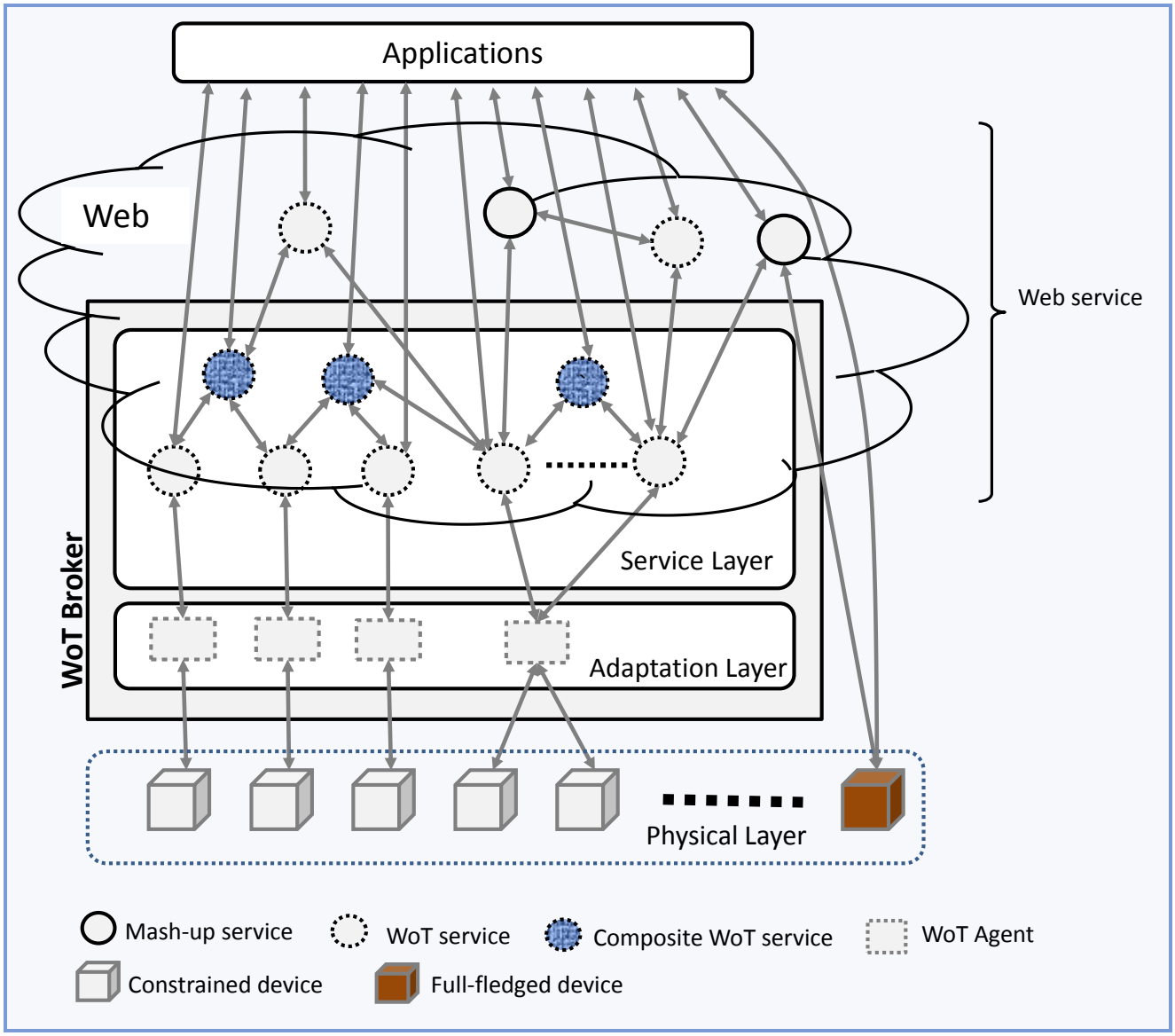


Figure 9-1 – Overview of the web of things architecture

9.2 Functional architecture of the WoT broker

Figure 9-2 shows the functional architecture of the WoT broker. The WoT broker functional architecture is divided into the service layer and adaptation layer and it consists of six functional entities (FEs) and several WoT agents. The service layer consists of the service profile management FE, service control FE, service composition FE and service access control FE. The adaptation layer consists of the resource ID management FE, resource management FE and WoT agents.

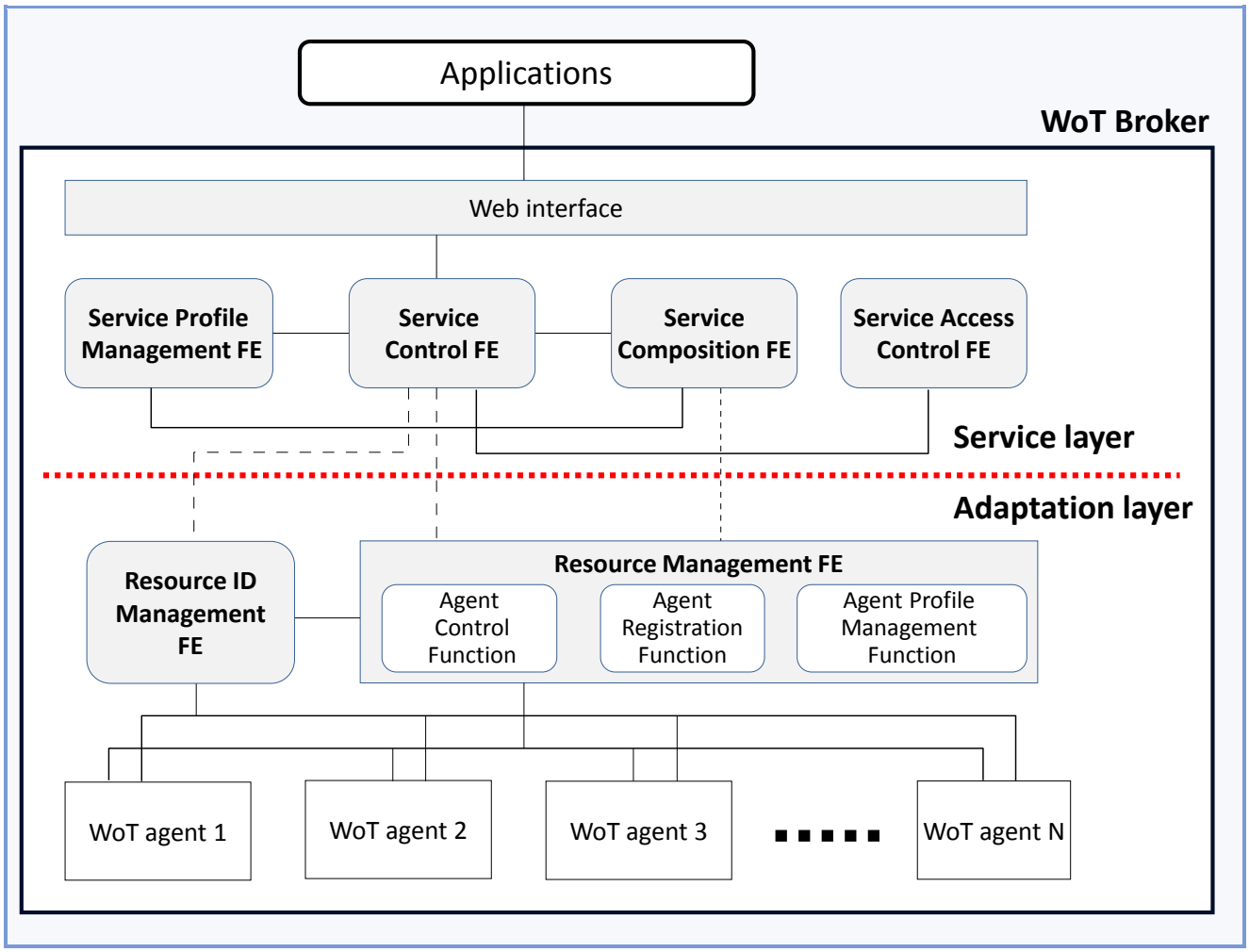


Figure 9-2– Functional architecture of the WoT broker

Based on the functional architecture of the WoT broker, it is possible to describe five service information flows: service discovery, service registration, service execution, service composition and agent registration. These information flows are described in Appendix II.

9.2.1 Service profile management functional entity

The service profile management FE contains service information which is supported by the WoT broker and this FE is responsible for registering WoT services. This FE interworks with the service control FE for the discovery of services. It also interworks with the service composition FE which has the ability to composite new services using registered services.

This FE has the following information:

- type of service (e.g., power control service, monitoring service, sensing service, printing service);

- service category name (e.g., personal WoT service, public WoT service, company WoT service);
- service name;
- service provider information.

The service profile is updated by requests from the service control FE and service composition FE.

9.2.2 Service control functional entity

The service control FE is responsible for the access, execution and management between resources and applications. The service control FE interworks with the service profile management FE to discover registered services and it interworks with a service access control FE to check whether the service requester has the right authentication and authorization or not.

In addition, the service control FE has responsibility for service registration/deregistration with the service profile management FE. In order to provide a service, the service should be registered with the service profile management FE. When a new service is created or deleted, the service control FE sends a service registration/deregistration request to the service profile management FE. The service control FE also helps the service composition FE to find services which are using new composite services.

9.2.3 Service composition functional entity

The service composition FE provides the capabilities to compose existing services to create new services. Service composition is accomplished by service providers. New composition services are registered at the service profile management FE.

NOTE – The service logic of a composite service is not defined in this Recommendation because it depends on the implementation.

9.2.4 Service access control functional entity

The service access control FE is responsible for access control of the user (e.g., application provider, service provider). It checks and controls user authentication, authorization, accounting and user-related information. For example, if an unauthorized user requests to access/use services, this FE rejects to access the services. This FE interworks with the service control FE to support access control.

9.2.5 Resource management functional entity

There are many agents in the WoT broker to support physical devices belonging to heterogeneous networks. The resource management FE is able to control each agent and it can register/deregister agents and maintain the information of each agent.

1) Agent control function

This function is responsible for controlling each agent. It can identify resources (device/service) and execute the requested service by interworking with the resource ID management FE.

2) Agent registration function

The agent registration function is responsible for the registration and deregistration of agents. When a new agent appears, the service provider can register the new agent using the agent registration function. Information related to the new agent is registered through the agent profile management function.

3) Agent profile management function

The agent profile management function is responsible for checking and storing agents with agent-related information. It checks the agent's authentication, authorization and types of agent.

The agent profile management function has the following information:

- types of agent (e.g., Wi-Fi agent, Bluetooth agent)
- location of agent.

9.2.6 Resource ID management functional entity

The resource ID management FE is responsible for storing the identifiers of resources (e.g., resource ID, agent ID) and maintaining mapping information between the resource ID and agent ID. This FE can optionally maintain mapping information using the mapping table. When a service requester wants to use a resource, this FE can provide related information (e.g., agent ID, resource ID). The resource ID management FE is also recommended to maintain the latest mapping information. This FE has the following additional information:

- physical information related to the service (e.g., subnetwork ID, subnetwork type and service location).

9.2.7 WoT agent

A WoT agent is a bridge between the WoT service and the physical devices residing in a subnetwork. The WoT agent has the role of communicating and translating between the adaptation layer of the WoT broker and the physical devices in a subnetwork. If a WoT agent receives a request from a resource management FE in the WoT adaptation layer, the WoT agent translates the request for the proprietary interface of the device and sends the translated request to a device in the subnetwork using the communication protocol so that the device can recognize the request and execute it. If the WoT agent receives results from the device in the subnetwork using its communication protocol, the WoT agent also translates the results in order to understand on the web and send the translated results using the web interface and web protocol.

NOTE – There are different subnetworks. Therefore, the WoT broker may have many agents to support those subnetworks because an agent is dedicated only one subnetwork.

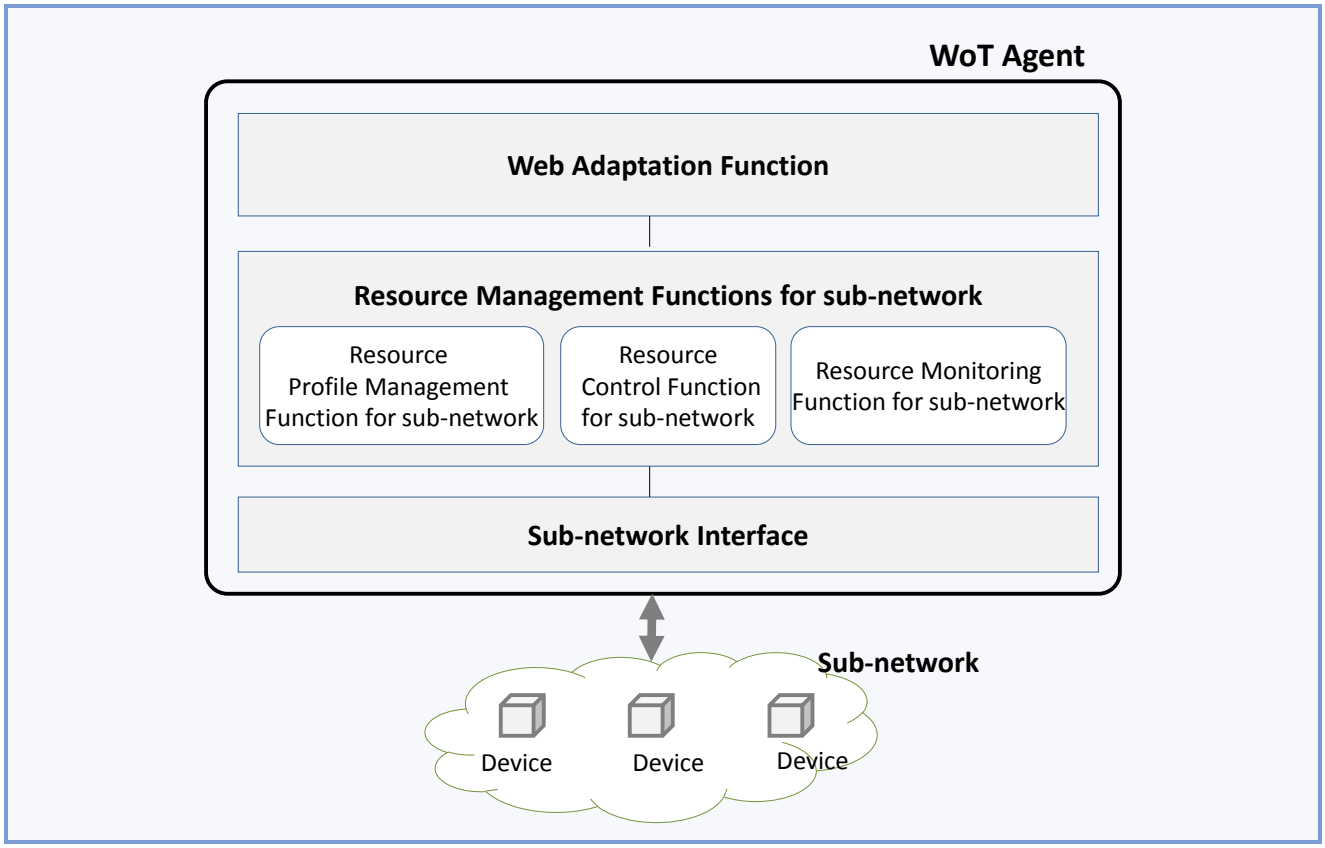


Figure 9-3 – Functional architecture of WoT agent

9.2.7.1 Web adaptation function

The web adaptation function is responsible for adapting the dedicated device interface to the web interface which can be accessed and used by the WoT service. It has two main capabilities as follows:

- adaptation of a subnetwork protocol to the web protocol in order to communicate between the physical device and the WoT service;
- supporting the description method and status of resources status based on the web.

9.2.7.2 Resource management functions for the subnetwork

Resource management functions for the subnetwork have responsibility for controlling and managing physical devices. The functions consist of the resource profile management function, resource control function and resource monitoring function for the subnetwork.

1) Resource profile management function for the subnetwork

The resource profile management function for the subnetwork keeps and maintains information of the physical devices which are located in the subnetwork. This function interworks with the resource monitoring function for the subnetwork to maintain the latest status of the physical devices.

The resource profile management function for the subnetwork has the following information:

- physical devices status (e.g., availability, capability)
- sub-network ID
- physical devices ID.

2) Resource control function for the subnetwork

This function is responsible for controlling the physical devices in the subnetwork. It can directly control and manage the physical devices in the subnetwork and provide registration/deregistration of devices in the subnetwork through interworking with the resource profile management function.

3) Resource monitoring function for the subnetwork

The resource monitoring function for the subnetwork checks and monitors the status of the physical devices (e.g., physical devices' availability, response time). If the status of the physical devices has changed, this function informs the resource profile management function for the subnetwork about it so that it can update the information of the physical device.

9.2.7.3 Subnetwork interface

Each subnetwork has a dedicated communication interface which is used for the interconnection of network elements (e.g., physical devices, agents).

10 Security considerations

Security is an important issue for WoT services because WoT services are built upon various kinds of physical devices. Some physical devices can provide strong security features themselves but others cannot provide all security features because they have many limitations (e.g., bandwidth, computing power). Thus, the WoT service provider should support the security of the physical devices, especially constrained devices (e.g., sensor). Additionally WoT service providers should verify the identification of users which access physical devices and WoT services, to protect against the unauthorized use of WoT services/physical devices and unauthorized access to applications.

Appendix I

Use cases and scenarios of the web of things

(This appendix does not form an integral part of this Recommendation.)

I.1 Home control services using WoT

This appendix describes a use case of how an owner of a home can control devices in his home on the web using the WoT. The scenario is outlined in Figure I.1. In this scenario, we can see seven physical devices (a rice cooker, an air-conditioner, a robot cleaner, a heater, a light controller, a temperature sensor, a TV) and we can see nine services which can be accessed by the web user. These services are classified as follows:

- WoT services: cooking service, home-cleaning service, home-light control service, home-heating service, home-cooling service, home-temperature-monitoring service;
- pure web service: TV control service;
- mash-up services: home temperature control service, home automation service.

Most of the devices (e.g., a cooker, an air-conditioner, a robot cleaner, a heater and a light controller) can be accessed and used on the web through the WoT broker. However the TV contains an embedded web server. Therefore it can be exposed and used on the web directly without the help of a WoT broker. In this scenario the WoT broker has three agents (i.e., a WoT agent for Bluetooth, WoT agent for Wi-Fi, WoT agent for Zigbee). Each agent can communicate with each device using its own dedicated interface and it performs adaptation roles to make each device accessible to web services. The WoT services can be used to make new composite services, i.e., mash-up services. In this scenario two mash-up services are shown. The home temperature control service is composed of a home-temperature-monitoring service, a home-heating service, and a home-cooling service. The owner of the home can easily maintain the desired temperature using this service on the web. The home automation service is composed of a home-cleaning service, a cooking service, a home-light control service and a TV control service.

If the owner of the home has devices connected to the web with a web agent (e.g., web browser), he can control and manage devices which are located in his home remotely. Based on these capabilities, we can consider the following scenarios:

- Scenario 1: If the owner wants to clean his home when he goes out, he simply commands the robot cleaner located at home to clean his home using a web browser. He can do it from his office or even while walking on the street using devices with web capabilities.
- Scenario 2: When an owner is about to leave his office on a very hot summer's day, he wants to return to a cool house. He can set the temperature through the web before he leaves his office.

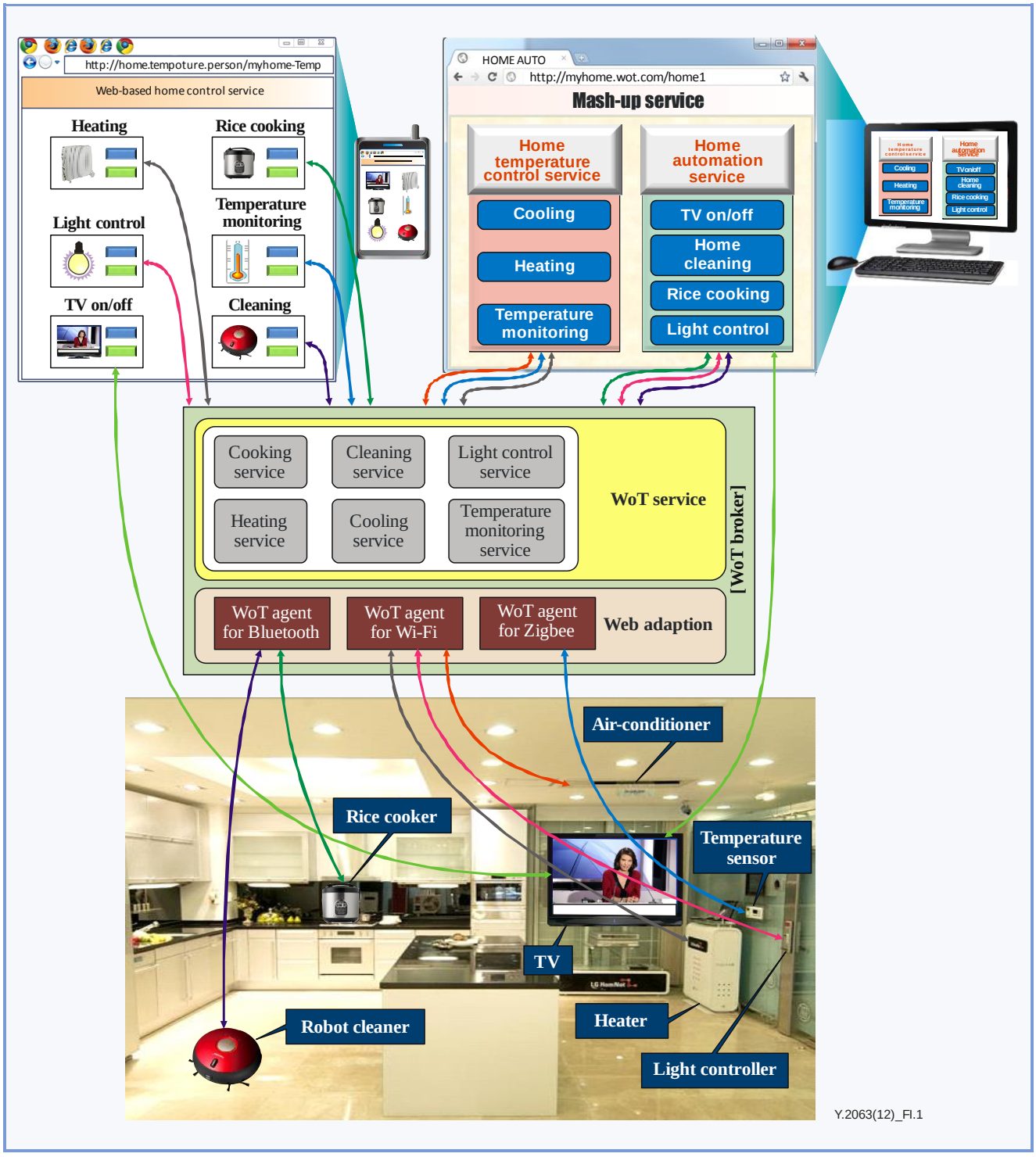


Figure I.1 – Home control service using the WoT

Appendix II

WoT broker service information flows

(This appendix does not form an integral part of this Recommendation.)

This appendix describes information flows related to the operation of a WoT broker which includes service discovery, service registration, service execution, service composition and agent registration. This appendix is helpful for understanding how applications can use WoT services.

II.1 Service discovery

Figure II.1 shows the information flows describing how the applications can discover WoT services.

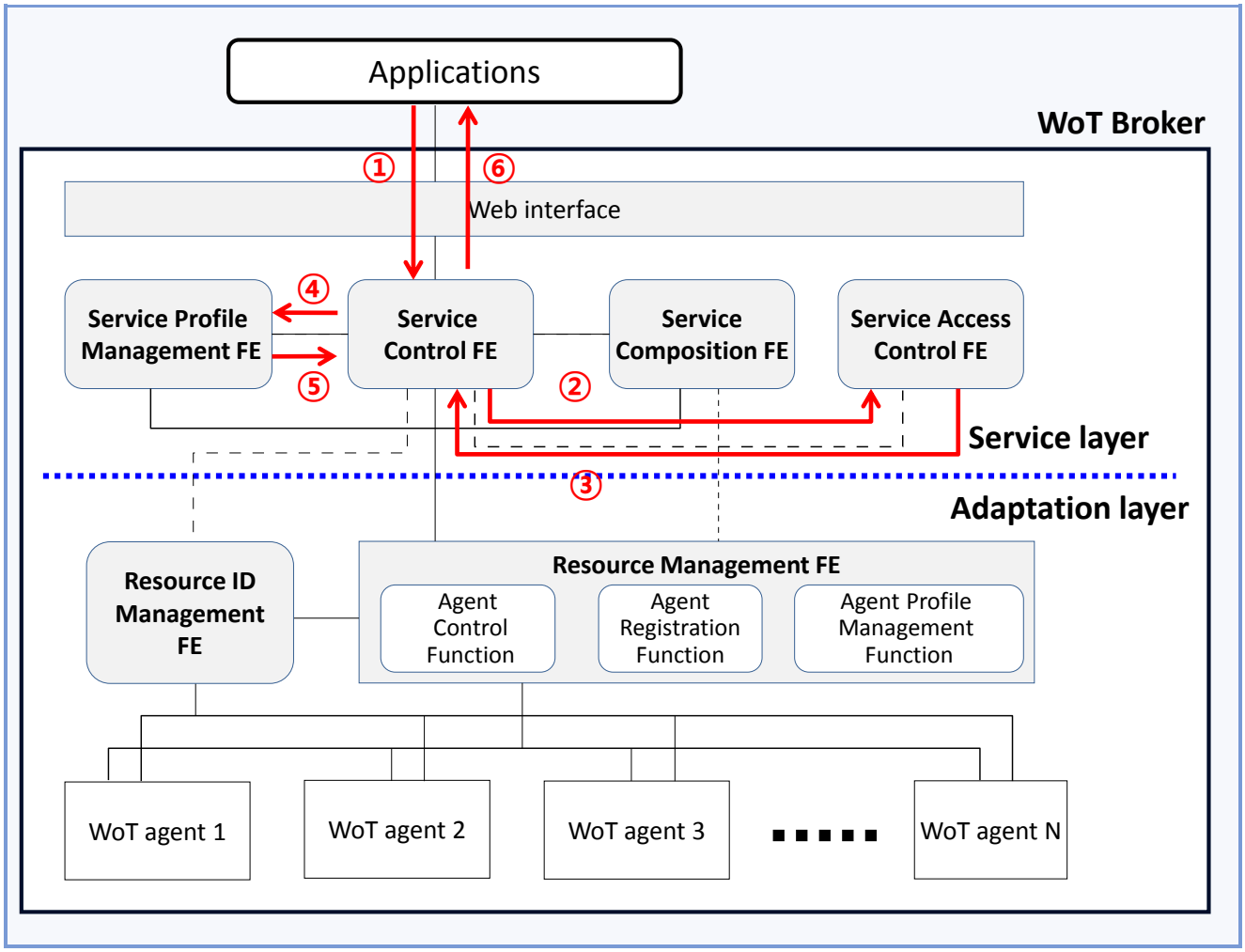


Figure II.1 – Information flow of service discovery in a WoT broker

(1) The application requests a WoT service using a web interface to a service control FE.

(2) The service control FE sends a request message to a service access control FE to check whether the application has authentication and authorization for the requested service.

(3) The service access control FE checks the authentication and authorization for the application. It sends a result to the service control FE.

(4) If the application has the right authentication and authorization, the service control FE sends a discovery message to a service profile management FE.

(5) The service profile management FE returns the search results of the service to the service control FE.

(6) The service control FE returns the information about the service to the application.

II.2 Service execution

Figure II.2 shows the information flows describing how to execute services in the WoT.

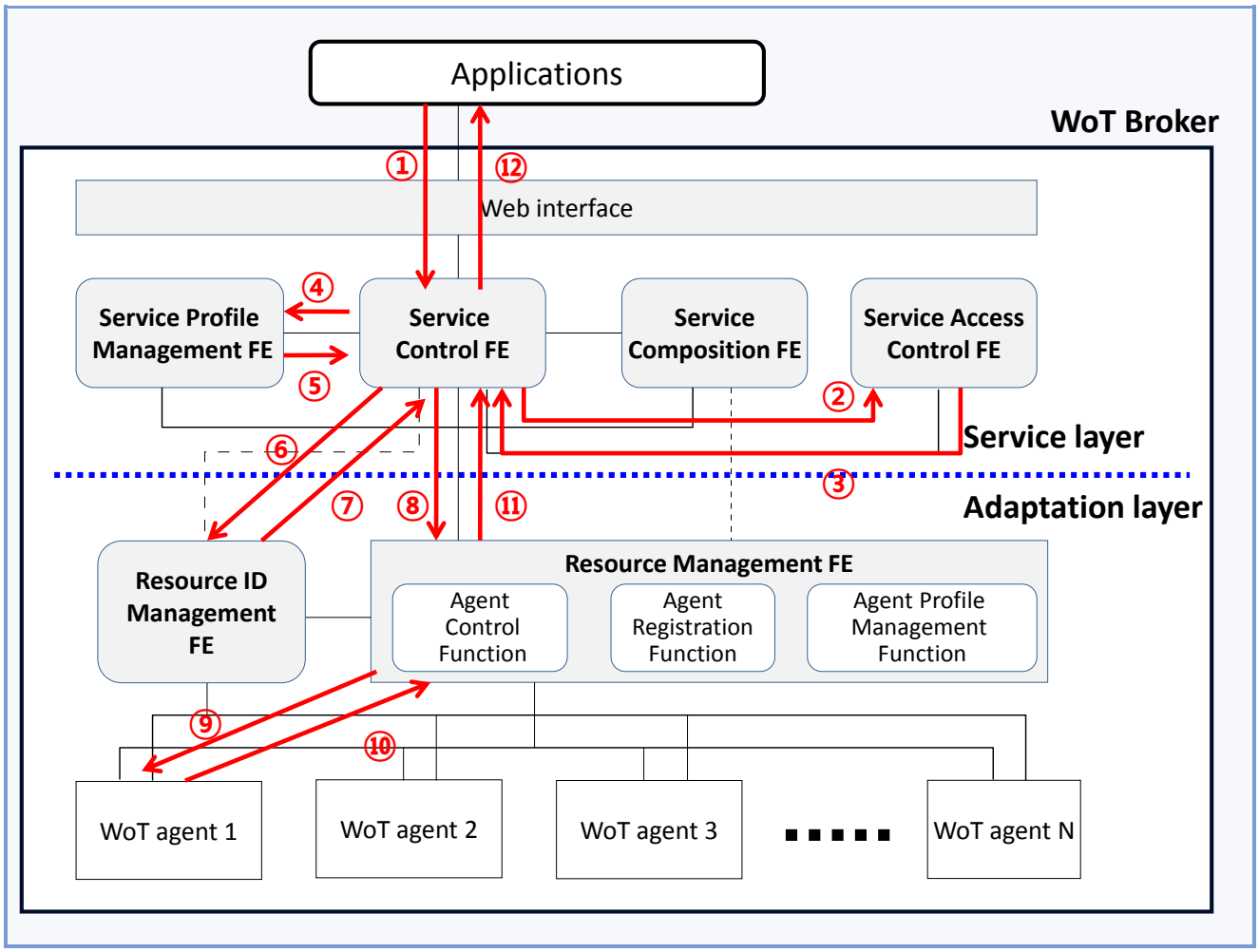


Figure II.2 – Information flow of service execution in a WoT broker

(1) The application requests a WoT service through a web interface to a service control FE.

(2) The service control FE sends a request message to a service access control FE to check whether the application has authentication and authorization for the requested service.

(3) The service access control FE checks authentication and authorization for the application. It sends the result to the service control FE.

(4) If the application has the right authentication and authorization, the service control FE sends a discovery message to a service profile management FE.

(5) The service profile management FE returns the search result of the service to the service control FE.

(6) The service control FE sends a message to the resource ID management FE to find which agent manages and controls the requested services.

(7) The resource ID management FE returns the search result of the service to the service control FE.

(8) The service control FE requests to execute the service to the resource management FE with a resource ID and an agent ID.

(9) The resource management FE checks the requested service and command the agent to execute the services.

(10) The agent returns the results of the service execution to the resource management FE.

(11) The resource management FE sends the results to the service control FE.

(12) The service control FE sends the results to the application.

II.3 Service composition

Figure II.3 shows the information flows describing how to composite services in the WoT broker.

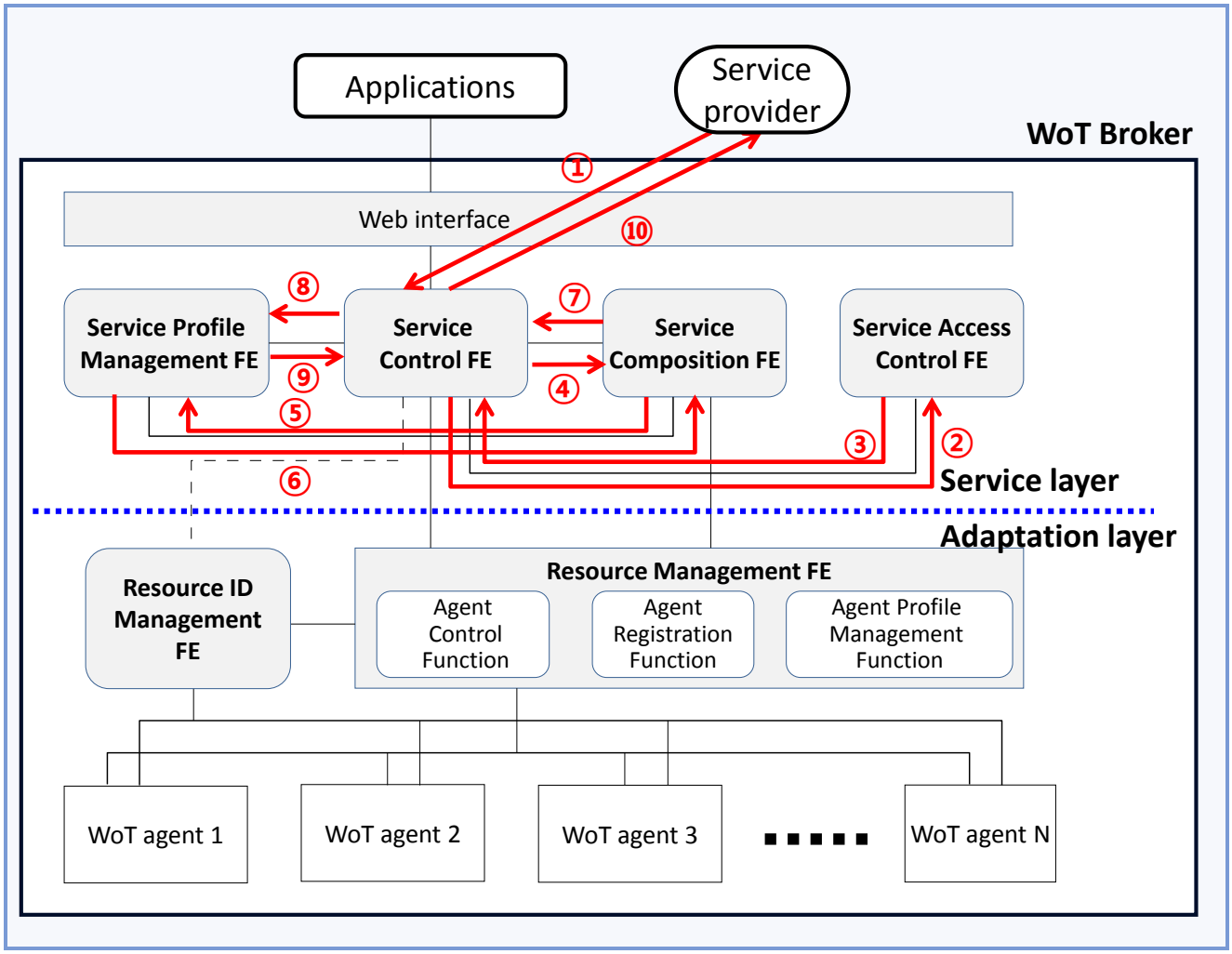


Figure II.3 – Information flow of service composition in a WoT broker

(1) A service provider requests service composition to a service control FE.

(2) The service control FE sends a request to a service access control FE regarding whether the service provider has authentication and authorization for the request.

(3) The service access control FE checks authentication and authorization for the application. It sends the result to the service control FE.

(4) The service control FE requests service composition to a service composition FE.

(5) The service composition FE sends a discovery message to the service profile management FE to find related services which will use the service composition process according to the request.

(6) The service profile management FE returns the results to the service composition FE.

(7) The service composition FE performs a composition process and it sends the result to the service control FE.

(8) The service control FE requests to register the new service to the service profile management FE.

(9) The service profile management FE returns the result to the service control FE.

(10) The service control FE sends the result to the service provider.

II.4 Agent registration

When there is a new agent, the new agent should be registered to the resource management FE to access and be used through the WoT broker. Figure II.4 shows the process of how to register an agent.

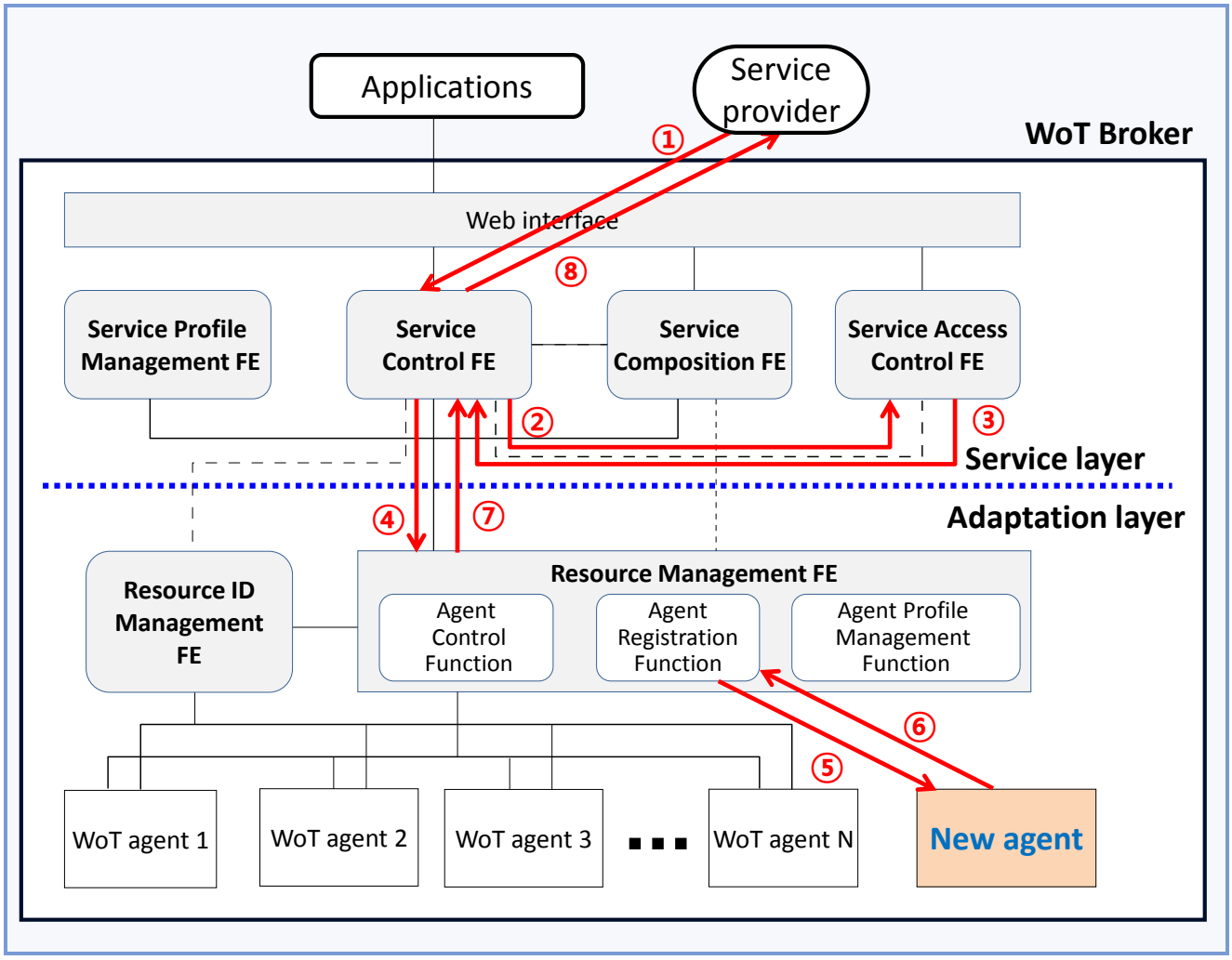


Figure II.4 – Information flow of agent registration in a WoT broker

(1) A service provider is responsible for registering an agent. The service provider requests that the service control FE registers a new agent.

(2) The service control FE sends a request to a service access control FE regarding whether the service provider and the new agent have authentication and authorization for the request.

(3) The service access control FE checks authentication and authorization about the service provider and the new agent. It sends the result of the request to the service control FE.

(4) The service control FE instructs the resource management FE to register the new agent.

(5) The resource management FE interworks with the new agent to register the agent. It requests information related to the agent (e.g., service category, network characteristics, number of services belonging to the new agent).

(6) The new agent sends the requested information.

(7) The resource management FE registers the new agent and sends the result to the service control FE.

(8) The service control FE informs the service provider of the result.

II.5 Service registration

A new service is registered through an agent that the service belongs to. An agent recognizes a new service in the subnetwork. The agent tries to register the service in the WoT broker.

Figure II.5 shows the process for how to register a service.

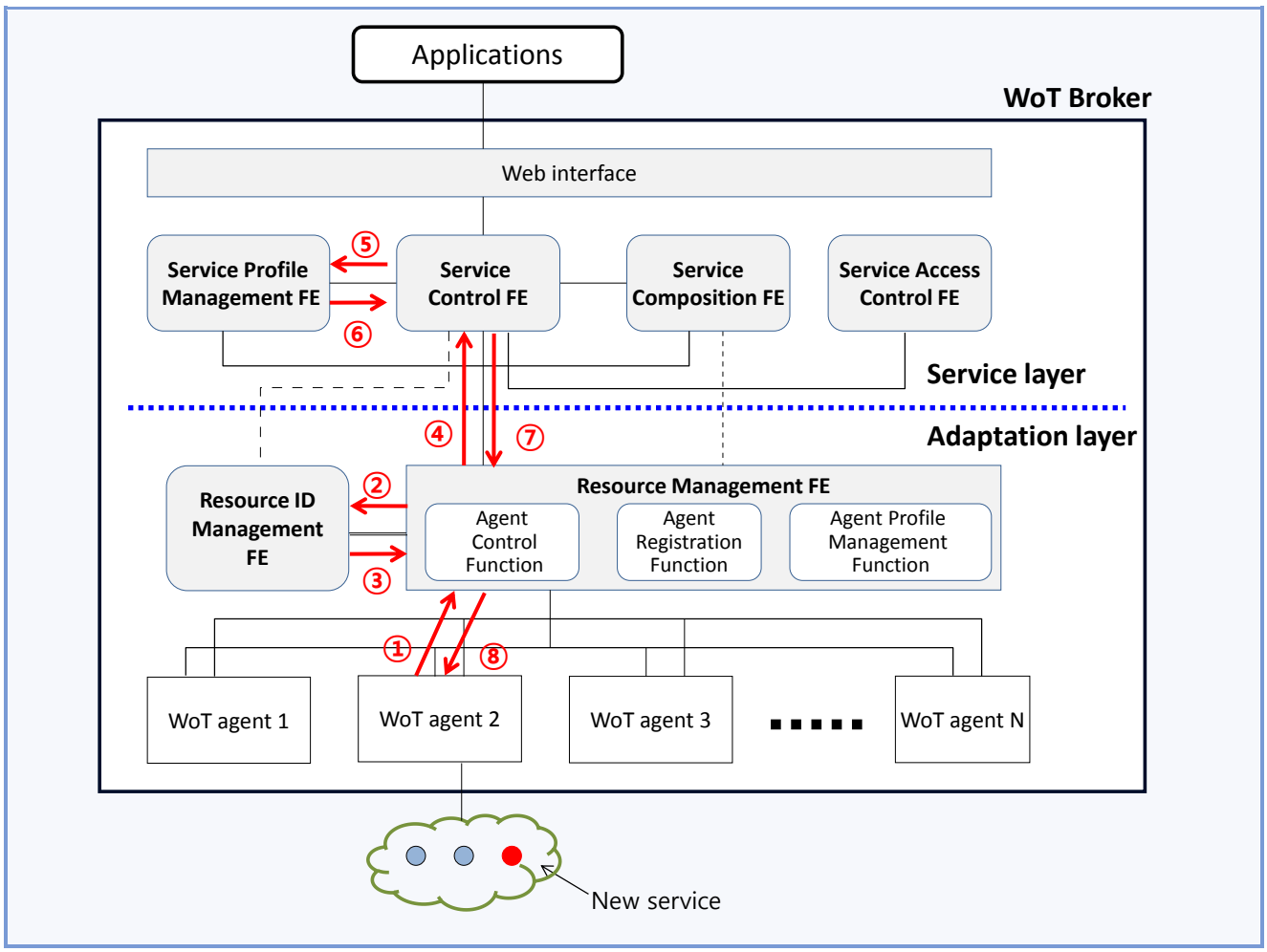


Figure II.5 – Information flow of service registration in the WoT broker

(1) An agent requests that the resource management FE registers a new service.

(2) The resource management FE checks the request and it sends a message to register the new service with information to the resource ID management FE.

(3) The resource ID management FE registers the new service with the resource ID and agent ID, and it returns the result to the resource management FE.

(4) The resource management FE tries to register the new service to the upper layer (service layer). It sends a message to register the new service to the resource control FE.

(5) The service control FE checks and requests to register the new service to the service profile management FE.

(6) The service profile management FE registers the new service and it returns the result to the service control FE.

(7) The service control FE returns the result to the resource management FE.

(8) The resource management FE informs the result to the agent.

Bibliography

[b-W3C dig loss] W3C (2005), *Glossary of Terms for Device Independence* <http://www.w3.org/TR/di-gloss/>

[b-W3C WACterms] W3C (1999), *Web Characterization Terminology & Definitions Sheet* <http://www.w3.org/1999/05/WCA-terms/>

[b-IETF RFC 3986] IETF RFC 3986 (2005), *IETF Uniform Resource Identifiers (URI): Generic Syntax* <http://datatracker.ietf.org/doc/rfc3986/?include_text=1>

[b-W3C web arch] W3C (2004), *Architecture of the World Wide Web, Volume One* <http://www.w3.org/TR/webarch/>

INTERNET OF THINGS

Y.4401/Y.2068

Functional framework and capabilities of the Internet of Things

Functional framework and capabilities of the Internet of things

Summary

Recommendation ITU-T Y.2068 provides a description of the basic capabilities of the Internet of things (IoT), based on the functional view, the implementation view and the deployment view of the IoT functional framework described in this Recommendation, in order to fulfil the IoT common requirements specified in Recommendation ITU-T Y.2066.

This Recommendation also describes additional capabilities of the IoT for the integration of cloud computing and big data technologies with the IoT.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T Y.2068	2015-03-22	13	11.1002/1000/12419

Keywords

Capability, deployment view, functional component, functional entity, functional framework, functional group, functional view, implementation view, Internet of things, IoT, IoT basic capabilities.

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

Table of Contents

Recommendation ITU-T Y.4401/Y.2068

Functional framework and capabilities of the Internet of things

1 Scope

This Recommendation describes the functional framework of the Internet of things (IoT) in three different views, the IoT basic capabilities, and additional capabilities for the integration of cloud computing and big data technologies with the IoT.

The scope of this Recommendation includes:

- concepts of the IoT functional framework;
- the functional view, the implementation view and the deployment view of the IoT functional framework;
- the IoT basic capabilities fulfilling the common requirements of the IoT specified in [ITU-T Y.2066];
- additional IoT capabilities for the integration of cloud computing and big data technologies with the IoT.

All capabilities of the IoT specified in this Recommendation are numbered and summarized in Annex A.

Appendix I provides an analysis of all capabilities of the IoT specified in this Recommendation in terms of matching with the common requirements of the IoT specified in [ITU-T Y.2066].

NOTE – The detailed specification of the capabilities identified in this Recommendation is outside the scope of this Recommendation.

2 References

The following ITU-T Recommendations and other references contain provisions which, through references in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T Y.2012] Recommendation ITU-T Y.2012 (2010), *Functional requirements and architecture of next generation networks*.

[ITU-T Y.2060] Recommendation ITU-T Y.2060 (2012), *Overview of the Internet of things*.

[ITU-T Y.2066] Recommendation ITU-T Y.2066 (2014), *Common requirements of the Internet of things*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 cloud computing [b-ITU-T Y.3500]: Paradigm for enabling network access to a scalable and elastic pool of shareable physical or virtual resources with self-service provisioning and administration on-demand.

NOTE – Examples of resources include servers, operating systems, networks, software, applications, and storage equipment.

3.1.2 device [ITU-T Y.2060]: With regard to the Internet of things, this is a piece of equipment with the mandatory capabilities of communication and the optional capabilities of sensing, actuation, data capture, data storage and data processing.

3.1.3 functional entity [ITU-T Y.2012]: An entity that comprises an indivisible set of specific functions. Functional entities are logical concepts, while groupings of functional entities are used to describe practical, physical implementations.

3.1.4 Internet of things (IoT) [ITU-T Y.2060]: A global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies.

NOTE 1 – Through the exploitation of identification, data capture, processing and communication capabilities, the IoT makes full use of things to offer services to all kinds of applications, whilst ensuring that security and privacy requirements are fulfilled.

NOTE 2 – From a broader perspective, the IoT can be perceived as a vision with technological and societal implications.

3.1.5 next generation network (NGN) [b-ITU-T Y.2001]: A packet-based network able to provide telecommunication services and able to make use of multiple broadband, QoS-enabled transport technologies and in which service-related functions are independent from underlying transport-related technologies. It enables unfettered access for users to networks and to competing service providers and/or services of their choice. It supports generalized mobility which will allow consistent and ubiquitous provision of services to users.

3.1.6 thing [ITU-T Y.2060]: With regard to the Internet of things, this is an object of the physical world (physical things) or the information world (virtual things), which is capable of being identified and integrated into communication networks.

3.2 Terms defined in this Recommendation

None.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

IoT	Internet of Things
MMCF	Mobility Management and Control Functions
NACF	Network Attachment Control Functions
NGN	Next Generation Network
QoS	Quality of Service
RACF	Resource and Admission Control Functions
TaaS	Things as a Service

5 Conventions

In this Recommendation:

The keywords "**is required to**" indicate a requirement which must be strictly followed and from which no deviation is permitted if conformance to this document is to be claimed.

The keywords "**is recommended**" indicate a requirement which is recommended but which is not absolutely required. Thus this requirement need not be present to claim conformance.

The keywords "**can optionally**" and "**may**" indicate an optional requirement which is permissible, without implying any sense of being recommended. These terms are not intended to imply that the vendor's implementation must provide the option and the feature can be optionally enabled by the network operator/service provider. Rather, it means the vendor may optionally provide the feature and still claim conformance with the specification.

6 Concepts of the IoT functional framework

6.1 Openness and interoperability of the IoT capabilities

The openness of the IoT capabilities refers to opening the networking and service-provisioning functionalities of the IoT to stimulate the innovation ability for IoT technologies and applications development. Through an open, shared, collaborative approach, IoT applications can be developed effectively for business, industry, and social community.

The openness of IoT capabilities can be realized by encapsulating IoT capabilities into service-provisioning interfaces.

The "open IoT capabilities" refer to the set of the IoT capabilities that are required to be opened to IoT applications or users. These IoT capabilities should have open interfaces that can be accessed by IoT applications or users.

The interoperability of the IoT capabilities can be realized by specifying the service-provisioning interfaces in a standardized way.

The "interoperable IoT capabilities" refer to the set of the IoT capabilities that are required to interact between different IoT functional elements, especially when deployed by different service providers. These IoT capabilities are distributed across different functional elements, and collaboration between these different functional elements fulfils their functionalities.

6.2 Completeness, implementability and applicability of the IoT capabilities

The completeness of the IoT capabilities refers to the fact that the whole set of IoT capabilities can fulfil all the common requirements of the IoT [ITU-T Y.2066].

NOTE – There may not be one-to-one mapping between IoT common requirements and IoT capabilities (i.e., one common requirement may involve multiple capabilities).

The implementability of the IoT capabilities refers to the set of the IoT capabilities that can be implemented in the functional elements described in, or reasonably derived from, specifications of existing networks.

The applicability of the IoT capabilities refers to the set of the IoT capabilities that can be deployed in the functional elements of the IoT implementations.

The IoT capabilities specified in this Recommendation should fulfil the requirements of completeness, implementability and applicability. These characteristics of the IoT capabilities specified in this Recommendation are validated by the IoT functional framework.

6.3 The different views of the IoT functional framework

The IoT functional framework consists of the IoT functional elements and their relations. In this Recommendation, the IoT functional framework can be described via three distinct views, i.e., the functional view, the implementation view and the deployment view.

NOTE 1 – The three views reflect three different phases of development of the IoT, namely the design phase, implementation phase, and deployment phase. Each view describes IoT capabilities aiming to fulfil the requirements encountered in different phases of development of the IoT.

NOTE 2 – The IoT functional elements in the functional view are named "functional groups". The IoT functional elements in the implementation view are named "functional entities". The IoT functional elements in the deployment view are named "functional components".

The functional view identifies functional groupings of IoT capabilities. The functional view of the IoT functional framework consists of the IoT "functional groups", and their relations. The functional view of the IoT functional framework is used to describe the completeness of the IoT capabilities by establishing the relations of the IoT capabilities with the common requirements of the IoT.

NOTE 3 – Functional groupings help to simplify the specification and analysis of the IoT capabilities.

The implementation view identifies capabilities of the IoT when implementation of functional groupings is realized. The implementation view of the IoT functional framework consists of the IoT "functional entities", and their relations. The implementation view of the IoT functional framework is used to describe the implementability of the IoT capabilities by establishing the relations of the IoT capabilities with the functional entities described in, or reasonably derived from, specifications of existing networks.

The deployment view identifies capabilities of the IoT when deployment of functional entities is realized. The deployment view of the IoT functional framework consists of the IoT "functional components" (such as gateway for IoT as specified in [b-ITU-T Y.2067]) and their relations. The deployment view of the IoT functional framework is used to describe the applicability of the IoT capabilities by establishing the relations of the IoT capabilities with the functional components deployed in concrete IoT implementations.

The capabilities identified via the three views are the "basic IoT capabilities" which fulfil the common requirements of the IoT [ITU-T Y.2066] (see clause 8). Additional capabilities which fulfil some common requirements of the IoT [ITU-T Y.2066] are identified for the integration of cloud computing and big data technologies with the IoT (see clause 9).

7 The IoT functional framework

7.1 The IoT functional framework in functional view

The IoT functional framework in functional view is to describe the IoT capabilities at the functional level in order to guarantee that the IoT capabilities can fulfil all common requirements of the IoT specified in [ITU-T Y.2066]. A practical way is to describe the IoT capabilities in groups corresponding to all categories of common requirements of the IoT as specified in [ITU-T Y.2066]. The IoT functional framework in functional view consists of groups of the IoT capabilities and their relationships.

In this Recommendation, the groups of the IoT capabilities are named "IoT functional groups". The classification of the IoT functional groups is based on the following requirement categories specified in [ITU-T Y.2066]: application support requirements, service requirements, data management requirements, device requirements, communication requirements, security and privacy protection requirements, and non-functional requirements.

IoT functional group names correspond to those of the requirement categories as follows: application support group, service provision group, data management group, connectivity group, communication group, security and privacy protection group and management group.

7.1.1 The IoT functional groups

The application support group is defined as a group of the IoT capabilities that can fulfil the requirements specified in the category of application support requirements [ITU-T Y.2066].

NOTE 1 – Based on the specifications of the category of application support requirements in [ITU-T Y.2066], this group of capabilities cannot be used directly by IoT users, but can be used by service providers.

NOTE 2 – IoT user, service provider, data manager and thing are the four IoT actors as described in clause 6 of [ITU-T Y.2066]. In this Recommendation, the term "thing" refers to "physical thing" as noted in clause 6.2.1 of [ITU-T Y.2066].

The service provision group is defined as a group of the IoT capabilities that can fulfil the requirements specified in the category of service requirements [ITU-T Y.2066].

NOTE 3 – Based on the specifications of the category of service requirements in [ITU-T Y.2066], this group of capabilities can be used by IoT users, service providers and things.

The data management group is defined as a group of the IoT capabilities that can fulfil the requirements specified in the category of data management requirements [ITU-T Y.2066].

NOTE 4 – Based on the specifications of the category of data management requirements in [ITU-T Y.2066], this group of capabilities can be used by data managers.

The connectivity group is defined as a group of the IoT capabilities that can fulfil the requirements specified in the category of device requirements [ITU-T Y.2066].

NOTE 5 – Based on the specifications of the category of device requirements in [ITU-T Y.2066], this group of capabilities can be used by data managers and things.

The communication group is defined as a group of the IoT capabilities that can fulfil the requirements specified in the category of communication requirements [ITU-T Y.2066].

NOTE 6 – Based on the specifications of the category of communication requirements in [ITU-T Y.2066], this group of capabilities can be used by IoT users, service providers and things.

The security and privacy protection group is defined as a group of the IoT capabilities that can fulfil the requirements specified in the category of security and privacy protection requirements [ITU-T Y.2066].

NOTE 7 – Based on the specifications of the category of security and privacy protection requirements in [ITU-T Y.2066], this group of capabilities can be used by IoT users, things, service providers and data managers.

The management group refers to a group of the IoT capabilities that can fulfil some non-functional requirements, such as manageability, reliability, high availability. The management group includes the capabilities for managing the operations related to application support, service provision, data management, connectivity, and communication of the IoT.

NOTE 8 – Based on the specifications of the category of security and privacy protection requirements in [ITU-T Y.2066], this group of capabilities can be used by IoT users, service providers or data managers.

7.1.2 Relations among the IoT functional groups

Figure 7-1 describes the IoT functional framework in functional view constituted by the IoT functional groups and the relations among these groups. The connectivity group is within the device layer defined in [ITU-T Y.2060], the communication, data management, service, and application support groups are within the network layer and the service support and application support layer defined in [ITU-T Y.2060].

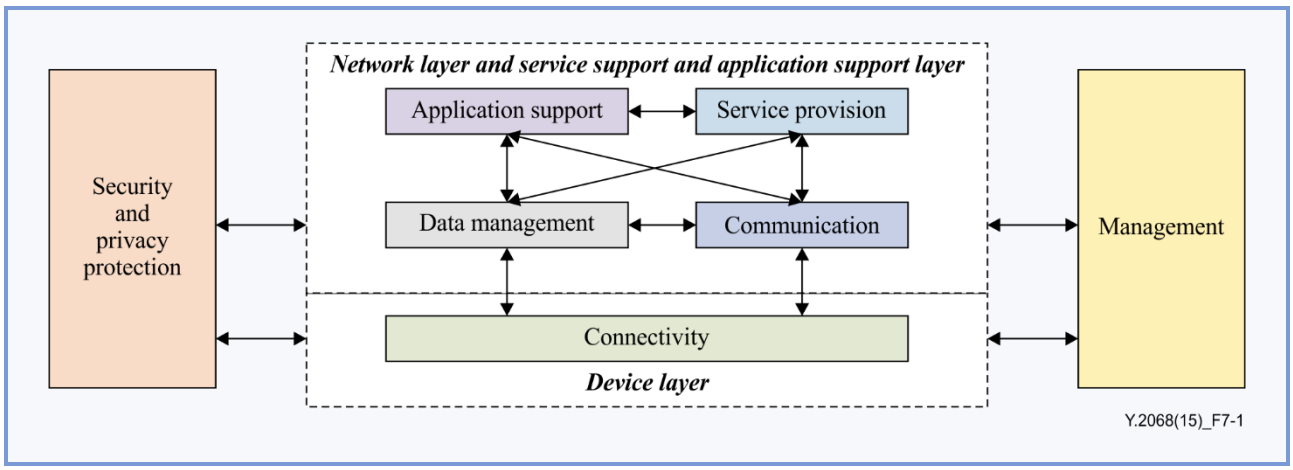


Figure 7-1 – The IoT functional framework in functional view

The connectivity group provides services to the data management group and communication group. The connectivity group can provide services to the communication group and data management group triggered by requests. The security and privacy protection group configures and manages the security and privacy protection aspects of connectivity capabilities, and the management group configures and manages the other aspects of connectivity capabilities.

The communication group provides communication services to the other functional group. The other functional groups use the communication services. The management group configures and manages the communication capabilities. The security and privacy protection group configures and manages the security and privacy protection aspects of communication capabilities.

The data management group provides services to the other functional groups. The other functional groups request and configure the data management services. The management group configures and manages the data management capabilities. The security and privacy protection group configures and manages the security and privacy protection aspects of data management capabilities.

The application support group requests services from the data management group and communication group, and these two groups can provide services to the application support group. The management group configures and manages the application support capabilities. The security and privacy protection group configures and manages the security and privacy protection aspects of application support capabilities.

The service provision group requests services from the data management group and communication group, and these two groups can provide services to the service provision group. The management group configures and manages the service provision capabilities. The security and privacy protection group configures and manages the security and privacy protection aspects of the service provision capabilities.

The security and privacy protection group configures and manages the security and privacy protection aspects of the capabilities in other functional groups.

The management group configures and manages the capabilities, except the security and privacy protection aspects of these capabilities, in other functional groups.

7.2 The IoT functional framework in implementation view

In this Recommendation, the functional entities of the implementation view are only described by their capabilities without mentioning their detailed relationships.

NOTE – There may be different implementation views based on different implementation approaches of the IoT. In this Recommendation, only one implementation view of the IoT functional framework is presented in

order to describe and analyse the capabilities of the IoT. It is anticipated that there is no need to cover all possible implementation views of the IoT functional framework: the implementation view of the IoT is in fact only used for showing the implementability of the IoT capabilities, so one implementation view of the IoT is sufficient to show this possibility.

7.2.1 Structure of an implementation view

An implementation view of the IoT functional framework consists of the functional entities of the IoT, and their high level relations. Figure 7-2 illustrates an implementation view of the IoT functional framework based on the IoT reference model specified in [ITU-T Y.2060] and the IoT common requirements specified in [ITU-T Y.2066], and building over functional entities described in the NGN functional architecture [ITU-T Y.2012].

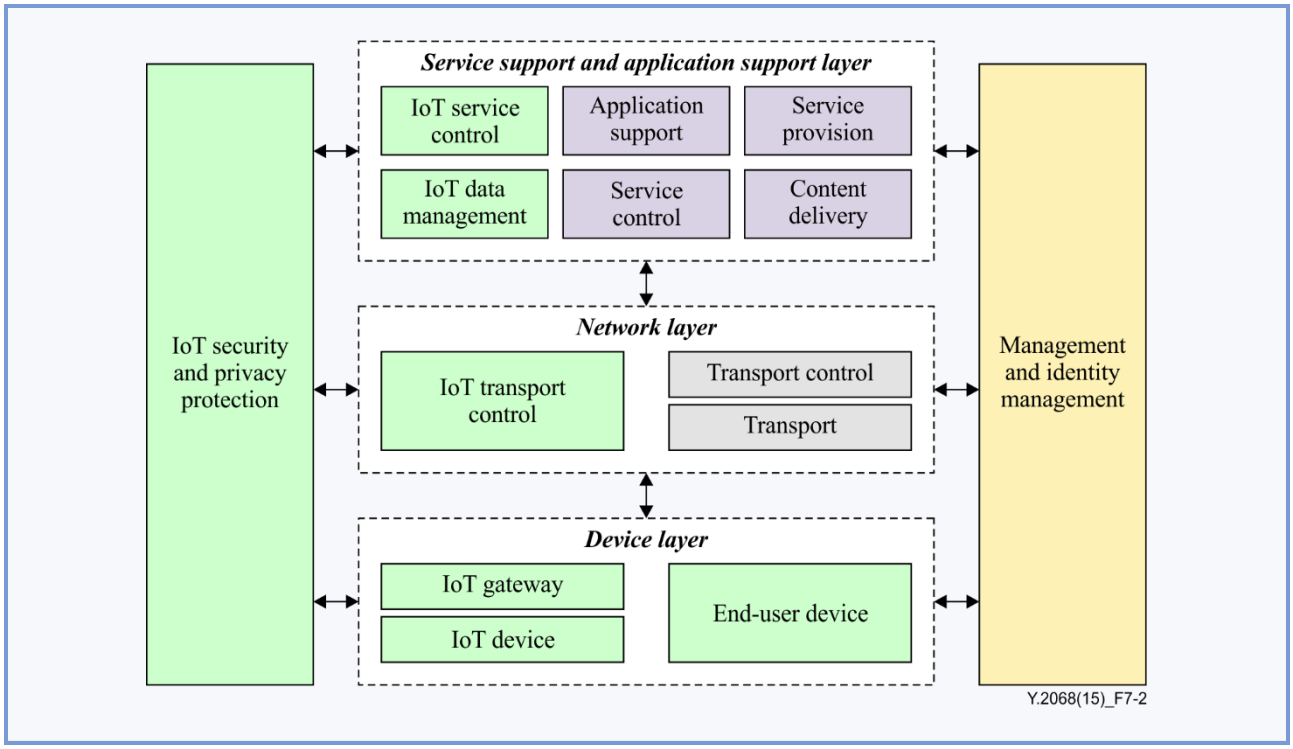


Figure 7-2 – Implementation view of the IoT functional framework building over the NGN functional architecture

There are two classes of functional entities in this implementation view of the IoT functional framework, one is for the functional entities already specified for the NGN [ITU-T Y.2012], and another is for the functional entities specific to the IoT.

The functional entities that are illustrated by green boxes in Figure 7-2 are the functional entities specific to the IoT (to be specified in this Recommendation), while the functional entities illustrated by differently coloured boxes are the functional entities described in [ITU-T Y.2012]. Among the functional entities described in [ITU-T Y.2012], the functional entities illustrated by the same colour belong to a single functional layer except the Management and Identity Management functional entity that crosses all functional layers of the IoT reference model [ITU-T Y.2060].

Even if some end-user functions are already mentioned in NGN Recommendations, these Recommendations only cover specifications on interactions between end-user functions and other NGN functions. There is no specification of end-user functions. In the implementation view, the end-user functions are needed to be described in order to cover the possibility that the IoT device capabilities are implemented in end-user functional entities. The "End-User" functional entity of NGN Recommendations enhanced with some IoT device capabilities is named as "End-User Device" functional entity in this Recommendation.

NOTE 1 – A smart phone configured with sensors and associated application software is an implementation of the End-User Device functional entity.

With respect to the functional entities already specified for the NGN, the Transport and Transport Control functional entities are in the network layer, and the Application Support, Service Provision, Service Control and Content Delivery functional entities are in the service support and application support layer. The Management and Identity Management functional entity crosses all functional layers.

With respect to the functional entities specific to the IoT, the IoT Device, the IoT Gateway and the End-User Device functional entities are in the device layer, the IoT Transport Control functional entity in the network layer, the IoT Data Management and the IoT Service Control functional entities in the service support and application support layer. The IoT Security and Privacy Protection functional entity crosses all functional layers.

NOTE 2 – The functional entities described in this Recommendation located in the service support and application support layer are only related with the generic support capabilities specified in [ITU-T Y.2060].

7.2.2 Functional entities of an implementation view

The Transport functional entity illustrated in Figure 7-2 includes the access network functions, edge functions, core transport functions, gateway functions, and media handling functions as specified in [ITU-T Y.2012].

The Transport Control functional entity illustrated in Figure 7-2 includes resource and admission control functions (RACF), network attachment control functions (NACF), and mobility management and control functions (MMCF) as specified in [ITU-T Y.2012].

The Service Provision functional entity and the Application Support functional entity illustrated in Figure 7-2 include functions such as the gateway, registration, authentication and authorization functions at the application level as specified in [ITU-T Y.2012].

The Service Control functional entity illustrated in Figure 7-2 includes resource control, registration, and authentication and authorization functions at the service level for both mediated and non-mediated services as specified in [ITU-T Y.2012].

The Content Delivery functional entity illustrated in Figure 7-2 receives content from the Application Support functional entity and Service Provision functional entity, stores, processes, and delivers it to the End-User Device functional entity using the capabilities of the Transport functional entity, under control of the Service Control functional entity as specified in [ITU-T Y.2012].

The Management and Identity Management functional entity illustrated in Figure 7-2 includes management functions and identity management functions as specified in [ITU-T Y.2012].

The IoT Device functional entity contains the capabilities of connecting and monitoring things, or controlling things that fulfil the device requirements of the IoT specified in [ITU-T Y.2066].

The IoT Gateway functional entity contains the capabilities of interconnecting devices with networks, buffering and transferring data, and configuring and monitoring devices that fulfil some device requirements and some data management requirements of the IoT specified in [ITU-T Y.2066] and [b-ITU-T Y.2067].

The End-User Device functional entity contains the capabilities of time synchronization, collaboration among services or among devices, reliable and secure human body connectivity, automatic service, intelligent communication, and device mobility to fulfil some application support requirements, service requirements, communication requirements, and device requirements of the IoT specified in [ITU-T Y.2066].

NOTE – As the above capabilities can be distributed in different functional entities, the capabilities contained in the End-User Device functional entity are named by prefixing them with the term "end-user" in order to distinguish them from capabilities residing in other functional entities.

The IoT Data Management functional entity contains the capabilities of semantic annotating, aggregating, storing, and transporting data of things that fulfil the data management requirements of the IoT specified in [ITU-T Y.2066].

The IoT Transport Control functional entity contains the capabilities of configuring and monitoring communication modes, autonomic networking, content-aware communication, and location-based communication that fulfil some communication requirements specified in [ITU-T Y.2066].

The IoT Service Control functional entity contains the capabilities of group management, time synchronization, collaboration among services, configuring and monitoring the semantic based services, autonomic services, location-based and context-aware services that fulfil some service requirements of the IoT specified in [ITU-T Y.2066].

The IoT Security and Privacy Protection functional entity contains the capabilities of performing the operations of security and privacy protection in communication, data management, and service provisioning. These capabilities fulfil some security and privacy protection requirements of the IoT specified in [ITU-T Y.2066].

7.3 The IoT functional framework in deployment view

In this Recommendation, the functional components specified in the deployment view of the IoT functional framework are only described by their capabilities without mentioning their detailed relationships.

NOTE – There may be different deployment views based on different deployment approaches of the IoT. In this Recommendation, only one deployment view of the IoT functional framework is presented in order to describe and analyse the capabilities of the IoT. It is anticipated that there is no need to cover all possible deployment views of the IoT functional framework in the Recommendation: the deployment view of the IoT is in fact only used for showing the applicability of the IoT capabilities, so one deployment view of the IoT is enough to show this possibility.

7.3.1 Structure of a deployment view

A deployment view of the IoT functional framework consists of its functional components and their high level relations. Figure 7-3 illustrates a deployment view of the IoT functional framework based on the IoT reference model specified in [ITU-T Y.2060], the IoT common requirements specified in [ITU-T Y.2066], and the NGN components described in the NGN functional architecture [ITU-T Y.2012].

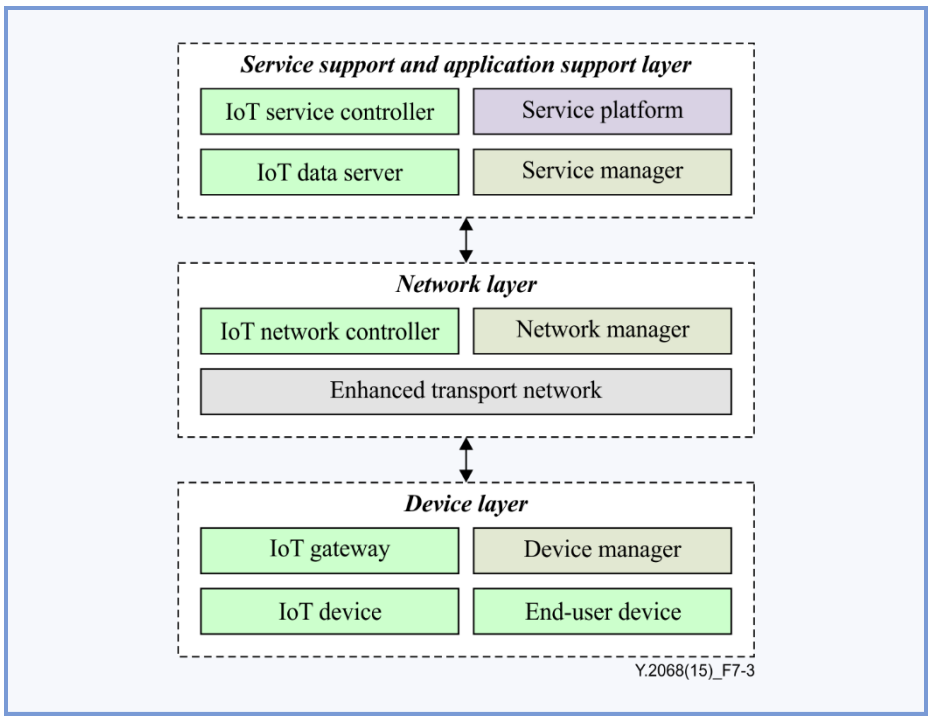


Figure 7-3 – Deployment view of the IoT functional framework building over the NGN components

The functional components that are illustrated by green boxes in Figure 7-3 are the functional components specific to the IoT (to be specified in this Recommendation), while the functional components illustrated by differently coloured boxes are the functional components described or partially described in [ITU-T Y.2012].

The deployment view of the IoT functional framework is only a logical approach for deploying IoT capabilities, and the functional components described in the deployment view can be mapped to physical components of some practical IoT deployments.

The functional components of this deployment view are classified respectively into device layer, network layer, and service support and application support layer as specified in [ITU-T Y.2060]. The cross-layer capabilities specified in [ITU-T Y.2060] are assigned to different functional components (such as Device Manager, Network Manager and Service Manager), distributed in each functional layer, in order to simplify the description and analysis of the IoT capabilities.

The IoT Device, IoT Gateway, End-User Device, and Device Manager functional components belong to the device layer. The Enhanced Transport Network, IoT Network Controller, and Network Manager functional components belong to the network layer. The IoT Data Server, IoT Service Controller, Service Platform, and Service Manager functional components belong to the service support and application support layer.

NOTE – The functional components described in this Recommendation in the service support and application support layer are solely related to generic support capabilities specified in [ITU-T Y.2060].

7.3.2 Functional components of a deployment view

The IoT Device functional component includes the capabilities of the IoT Device functional entity, capabilities of autonomic management and energy management, and capabilities of security and privacy protection.

The IoT Gateway functional component includes capabilities of interconnecting IoT Devices with Enhanced Transport Network, capabilities of aggregating and transferring data of things as well as capabilities of the IoT Device functional component.

The End-User Device functional component includes capabilities of existing networking terminal, and capabilities of the IoT Device functional components.

The Device Manager functional component includes capabilities of identifying and managing devices within a defined domain, and capabilities of autonomic management.

The Enhanced Transport Network functional component includes capabilities of transport and transport control as specified in [ITU-T Y.2012], and enhanced capabilities to fulfil some communication requirements specified in [ITU-T Y.2066].

The IoT Transport Controller functional component includes capabilities of configuring, monitoring, and controlling functionalities of the IoT related communication performed in the Enhanced Transport Network to fulfil communication requirements specified in [ITU-T Y.2066].

The Network Manager functional component includes capabilities of managing the Enhanced Transport Network, and capabilities of security and privacy protection in the Enhanced Transport Network.

The IoT Data Server functional component includes capabilities of storing, querying and managing data of things, and contains database and data management related with data of things.

The IoT Service Controller functional component includes capabilities of configuring, monitoring, and controlling functionalities of IoT application support and service provision performed in Service Platform to fulfil some application support requirements and service requirements of the IoT specified in [ITU-T Y.2066].

The Service Platform functional component includes capabilities of Application Support, Service Provision, Content Delivery, Service Control, and other enhanced capabilities to fulfil some application support requirements and service requirements specified in [ITU-T Y.2066].

The Service Manager functional component includes capabilities of managing both Service Platform and the IoT Service Controller, and capabilities of Security and Privacy Protection for Service Platform and for the IoT Service Controller.

8 The IoT basic capabilities

The IoT basic capabilities in this Recommendation refer to the capabilities that fulfil the common requirements of the IoT as specified in [ITU-T Y.2066].

Clauses 8.1 to 8.7 describe the IoT basic capabilities. These same capabilities are numbered and summarized in Annex A.

NOTE – In clauses 8.1 to 8.7, the capability numbers, as shown in Annex A, appear between square brackets at the end of the description of the corresponding capability.

8.1 Service provision capabilities

Service provision capabilities include service prioritization, semantic based service, service composition, mobility service, autonomic service, location-based and context-aware service, service management, service discovery, service subscription, naming and addressing, virtual storage and processing capabilities, adaptable service provision, and service provision acknowledgement.

- Service prioritization capability involves the abilities of providing services in different priorities, such as querying data or transferring data in different priorities [C-1-1].
- Semantic based service capability involves the abilities of semantically annotating data or service, semantically querying data or semantically requesting services [C-1-2].

NOTE – Semantic based service capability enables the description and exchange of semantics between services in order to support, for example, automatic service customization.

- Service composition capability involves the abilities of creating new services or customized services based on existing capabilities and user specific requirements [C-1-3].
- Mobility service capability involves the abilities of remote access to the IoT, and remote authentication of users [C-1-4].
- Autonomic service capability involves the abilities of automatic capturing, transferring, and analysing data of things, and automatic service provisioning based on predefined rules or policies [C-1-5].
- Location-based and context-aware service capability involves the abilities of automatically provisioning services based on location and context information, and predefined rules or policies [C-1-6].
- Service discovery capability involves the abilities of discovering IoT users, services, devices and things [C-1-7].
- Service subscription capability involves the abilities of subscribing the needed services and associated data of things by IoT users [C-1-8].
- Naming and addressing capability involves the abilities of creating, updating, deleting, querying names and addresses of users, devices and things [C-1-9].
- Virtual storage and processing capability involves the abilities of providing storage and processing resources in a scalable way [C-1-10].
- The capability of adaptable service provision involves the abilities of extending service configurations to provide new services as required by applications or users of the IoT in order to be adaptable to different applications or users of the IoT [C-1-11].
- The capability of service provision acknowledgement involves the abilities of acknowledging the correct service provision requested by applications or users of the IoT in order to support reliable service provision in the IoT [C-1-12].

8.2 Communication capabilities

The communication group includes event-based communication, periodic communication, self-configuring for networking, self-healing for networking, self-optimizing for networking, self-protection for networking, multicast communication, unicast communication, broadcast communication, anycast communication, error control for communication, Quality of Service enabling communication, content-aware communication, location-based communication, transport acknowledgement and adaptable networking capabilities.

- Event-based communication capability enables IoT devices and service provider to initiate communication based on predefined events [C-2-1].
- Periodic communication capability enables IoT devices and service provider to periodically initiate communication based on predefined rules [C-2-2].

NOTE 1 – In the perspective of network performance, it is required that the usage of event-based or periodic communication capabilities be avoided, unless there is a specific reason to communicate using these capabilities.

- Unicast communication capability enables the IoT to transfer messages from the source entity to single destination entity [C-2-3].
- Multicast communication capability enables the IoT to transfer messages from the source entity to a group of destination entities simultaneously [C-2-4].
- Broadcast communication capability enables the IoT to transfer messages to all destination entities of a given domain [C-2-5].
- Anycast communication capability enables the IoT to transfer messages to any of the destination entities of a given domain [C-2-6].

- The capability of error control for communications involves the abilities of ensuring correct message transfer from source entity to destination entity [C-2-7].
- Quality of Service enabling communication capability provides mechanisms to enable support of Quality of Service for message transfer from source entity to destination entity [C-2-8].
- The capability of self-configuring for networking involves the abilities of automatically configuring networking parameters based on discovered network interfaces and predefined rules [C-2-9].
- The capability of self-healing for networking involves the abilities of automatically recovering from fault status of networking based on monitoring and predefined rules [C-2-10].
- The capability of self-optimizing for networking involves the abilities of automatically optimizing networking operations based on monitoring and predefined rules [C-2-11].
- The capability of self-protecting for networking involves the abilities of automatically protecting networking entities from harmful operations based on predefined rules [C-2-12].
- Content-aware communication capability involves the abilities of selecting path and routing of messages based on content and predefined rules [C-2-13].

NOTE 2 – This capability can be used to block messages based on the specified content and predefined rules.

- The capability of location-based communication involves the abilities of identifying locations and initiating communication control based on identified locations and predefined rules [C-2-14].
- The capability of transport acknowledgement involves the abilities of acknowledging the correct message delivery to support reliable communications as required by IoT applications [C-2-15].
- The capability of adaptable networking involves the abilities of extending networking configurations for connecting to emerging communication networks of the IoT [ITU-T Y.2060] in order to be adaptable to different networking technologies [C-2-16].

8.3 Application support capabilities

The application support group includes programmable interface provision, group management, time synchronization, orchestration, user management, and application operation acknowledgement capabilities.

- The capability of programmable interface provision involves the abilities of supporting new services or customized services based on existing capabilities and application specific requirements [C-3-1].
- The capability of group management involves the abilities of creating, modifying, deleting, and querying IoT groups, and adding, modifying, deleting and querying IoT group members [C-3-2].
- The capability of time synchronization involves the abilities of synchronizing the time among related functional components in a reliable way, in order to support global or local time stamping for applications [C-3-3].
- Orchestration capability involves the abilities of automatic arrangement and coordination of service provisioning or device operations in order to fulfil application specific requirements [C-3-4].
- User management capability involves the abilities of creating, querying, updating and deleting IoT user profiles, and authenticating, authorizing, registering and auditing IoT users [C-3-5].

- The capability of application support operation acknowledgement involves the abilities of acknowledging the correct operations requested by applications in order to support reliable application operations in the IoT [C-3-6].

8.4 Data management capabilities

The data management group includes data storage, data processing, data querying, data access control, open information exchange, semantic data operation and autonomic data operation capabilities.

- The capability of data storage involves the ability of storing data of things based on predefined rules and policies [C-4-1].
- The capability of data processing involves the ability of data fusion and mining based on predefined rules and policies [C-4-2].

NOTE 1 – Data processing refers to a set of data operations in order to fulfil the application requirements. Data processing in the IoT includes collecting, representing, fusing, mining, and interpreting the data of things. From an application perspective, data processing can be regarded as data analysis that consists of data fusing and data mining. From an implementation perspective, the operation of data fusing includes data collection and data representation, and the operation of data mining includes data interpretation.

- The capability of data querying involves the ability of querying information about things connected to the IoT [C-4-3].
- The capability of data access control involves the abilities of controlling and monitoring data access operations by the owners of the data [C-4-4].
- The capability of open information exchange involves the abilities of sending data to or receiving data from external data sources, e.g., data centres and data servers outside the IoT [C-4-5].
- The capability of semantic data operation involves the abilities of semantic annotating, semantic discovering, semantic storing, and semantic composing data of things to fulfil the requirements of IoT users or applications [C-4-6].
- The capability of autonomic data operation involves the abilities of automatically collecting, aggregating, transferring, storing, analysing data of things, and automatically managing these data operations for support of operating data of things in a scalable way [C-4-7].

NOTE 2 – This capability can be used to face the impact of big data in the IoT.

8.5 Management capabilities

The management group includes capabilities fulfilling the IoT interoperability requirements, capabilities fulfilling the IoT scalability requirements, capabilities fulfilling the IoT reliability requirements, capabilities fulfilling the IoT high availability requirements, and capabilities fulfilling the IoT manageability requirements.

NOTE – The abilities involved in the management capabilities specified in this Recommendation may be operated in a remote way. Remote operation can be disabled based on security or other policy considerations.

8.5.1 Capabilities fulfilling IoT interoperability requirements

The capabilities fulfilling the IoT interoperability requirements specified in [ITU-T Y.2066] include managing data models for exchanging data of things, managing service description, managing network configuration, managing device configuration, managing security policy, and managing privacy protection policy capabilities.

- The capability of managing data models for exchanging data of things involves the abilities of creating, querying and updating data models for support of interoperability among IoT applications. This capability also includes the abilities of creating and updating data models for support of semantic interoperability among IoT applications [C-5-1].

- The capability of managing service description involves the abilities of creating, querying and updating service description for support of service interoperability [C-5-2].
- The capability of managing network configuration involves the abilities of creating, querying and updating network configuration for support of network interoperability [C-5-3].
- The capability of managing device configuration involves the abilities of creating, querying and updating network configuration for support of device interoperability [C-5-4].
- The capability of managing security policy involves the abilities of creating, querying and updating security policy for support of interoperability between different implementations of security policy [C-5-5].
- The capability of managing privacy protection policy involves the abilities of creating, querying and updating privacy protection policy for support of interoperability between different implementations of privacy protection policy [C-5-6].

8.5.2 Capabilities fulfilling the IoT scalability requirements

The capabilities fulfilling the IoT scalability requirements specified in [ITU-T Y.2066] include managing distributed processing and managing multiple domains.

- The capability of managing distributed processing involves the abilities of managing IoT functional components in a distributed way for support of IoT scalability [C-5-7].
- The capability of managing multiple domains involves the abilities of managing IoT functional components in multiple domains for support of IoT scalability [C-5-8].

8.5.3 Capabilities fulfilling the IoT reliability requirements

The capabilities fulfilling the IoT reliability requirements specified in [ITU-T Y.2066] include redundant deployment enablement capability.

- The capability of redundant deployment enablement involves the abilities of enabling deployment of redundant functional components of the IoT to guarantee reliability required in communication, service provision and data management [C-5-9].

8.5.4 Capabilities fulfilling the IoT high availability requirements

The capabilities fulfilling the IoT high availability requirements specified in [ITU-T Y.2066] include service integrity check, data integrity check, device integrity check, security integrity check and user integrity check capabilities.

- The capability of service integrity check involves the abilities of checking the service lifetime, the available resources required to provide the service in order to guarantee the high availability of service provisioning [C-5-10].
- The capability of data integrity check involves the abilities of checking the data lifetime, the available attributes of the data, and the consistency of data in order to guarantee the high availability of data management [C-5-11].
- The capability of device integrity check involves the abilities of checking the status of all functions of device to guarantee the high availability of IoT devices [C-5-12].
- The capability of security integrity check involves the abilities of checking the consistency of security policies deployed in all functional components of the IoT to guarantee the high availability of security in the IoT [C-5-13].
- The capability of user profile integrity check involves the abilities of checking the lifetime, subscription, privacy protection and availability of services subscribed to by users to guarantee the high availability of service provisioning and privacy protection for users [C-5-14].

8.5.5 Capabilities fulfilling the IoT manageability requirements

The capabilities fulfilling the IoT manageability requirements specified in [ITU-T Y.2066] include managing devices, managing networks, managing services, managing data operations, managing security operations, managing privacy protection, managing user operations, and plug and play capabilities.

- The capability of managing devices involves the abilities of configuring, monitoring, diagnosing and recovering devices of the IoT, and updating device software to enhance capabilities of devices of the IoT [C-5-15].
- The capability of managing networks involves the abilities of configuring, monitoring, accounting and charging, optimizing, diagnosing and recovering networks of the IoT [ITU-T Y.2060], and updating network software to enhance capabilities of networks of the IoT [C-5-16].
- The capability of managing services involves the abilities of describing, configuring, monitoring, accounting and charging, optimizing, recovering, and updating services of the IoT [C-5-17].
- The capability of managing data operations involves the abilities of configuring, monitoring, accounting and charging, optimizing and recovering data operations, and updating software related with data operations to enhance capabilities of the IoT [C-5-18].
- The capability of managing security operations involves the abilities of configuring, monitoring, auditing, diagnosing and recovering security operations, and updating software related with security operations to enhance capabilities of the IoT [C-5-19].
- The capability of managing privacy protection involves the abilities of configuring, monitoring, auditing and recovering privacy protection, and updating software related with privacy protection to enhance capabilities of the IoT [C-5-20].
- The capability of managing user operations involves the abilities of configuring, monitoring, accounting and charging, optimizing, diagnosing and recovering operations of IoT users, and updating software related with operations of IoT users to enhance capabilities of IoT [C-5-21].
- Plug and play capability involves the abilities of automatic configuring, connecting and activating devices of the IoT to enable on-the-fly semantic-based configuration and activation of IoT devices [C-5-22].

8.6 Connectivity capabilities

The connectivity group includes identification-based connectivity, things' status notification device mobility capability, and adaptable connectivity capabilities.

- The capability of identification-based connectivity involves the abilities of establishing the connectivity based on the identification of things [C-6-1].
- The capability of things' status notification involves the abilities of automatic notification of the status of things and its changes based on predefined rules [C-6-2].
- The capability of device mobility involves the abilities of keeping the connectivity with the IoT when a device moves [C-6-3].
- The capability of adaptable connectivity involves the abilities of extending connectivity configurations to enable connectivity of new types of devices to the IoT in order to be adaptable to different device technologies [C-6-4].

8.7 Security and privacy protection capabilities

The security and privacy protection group includes communication security capability, data management security capability, service provision security capability, security integration capability, mutual authentication and authorization capability, and security audit capability.

- Communication security capability involves the abilities of supporting secure, trusted and privacy-protected communication [C-7-1].
- Data management security capability involves the abilities of providing secure, trusted and privacy-protected data management [C-7-2].
- Service provision security capability involves the abilities of providing secure, trusted and privacy-protected service provision [C-7-3].
- Security integration capability involves the abilities of integrating different security policies and techniques related to the variety of IoT functional components [C-7-4].
- Mutual authentication and authorization capability involves the abilities of authenticating and authorizing each other before a device accesses the IoT based on predefined security policies [C-7-5].
- Security audit capability involves the abilities of monitoring any data access or attempt to access IoT applications in a fully transparent, traceable and reproducible way based on appropriate regulation and laws [C-7-6].

NOTE – These security and privacy protection capabilities include also the ability of coping with the security and privacy protection issues for operations across different domains.

9 IoT capabilities for integration of key emerging technologies

The following clauses describe the IoT capabilities for integration of some key emerging technologies, in alignment with the IoT capabilities list provided in Annex A. In the following clauses, the capability numbers, as shown in Annex A, are put between square brackets "[ ]" and inserted at the end of each paragraph describing the corresponding capability.

Clauses 9.1 and 9.2 describe the additional IoT capabilities for integration of cloud computing technologies and big data technologies.

NOTE – This Recommendation does not prevent more additional capabilities for integration with the IoT of other emerging technologies, such as network function virtualization and software-defined networking, to be considered further.

9.1 Capabilities for integration of cloud computing technologies

Owing to the high scalability, energy efficiency and deployment efficiency requirements of the IoT, there are some great challenges in the deployment of the IoT. Some key features of cloud computing technologies, such as virtualization and resource sharing, can help to improve scalability, energy efficiency (i.e., reduce the energy consumption) and deployment efficiency (e.g., reduce the memory and bandwidth usage) for the IoT. Additional capabilities for the integration of cloud computing technologies with the IoT are required.

With the integration of cloud capabilities of the infrastructure capabilities type [b-ITU-T Y.3500] into the IoT, the IoT infrastructure can be deployed utilizing these cloud capabilities. In this way, the IoT infrastructure can increase its scalability for computing, data storage and other aspects, and also increase energy efficiency. The capability of accessing virtual processing resources and the capability of accessing virtual storage resources are required in order to integrate with cloud capabilities of the infrastructure capabilities type.

- The capability of accessing virtual processing resources involves the abilities of adopting the capabilities specified in infrastructure capabilities type of cloud to use processing resource deployed in cloud [C-8-1].
- The capability of accessing virtual storage resources involves the abilities of adopting the capabilities specified in infrastructure capabilities type of cloud to use storage resource deployed in cloud [C-8-2].

With the integration of cloud capabilities of the platform capabilities type [b-ITU-T Y.3500] into the IoT, the IoT can deploy its platform according to this type of cloud capability. In such a way, the IoT can provide flexibility for the usage of IoT application support and service support capabilities, e.g., IoT application providers can more easily deploy IoT applications based on the platform of the IoT.

With the integration of cloud capabilities of the application capabilities type [b-ITU-T Y.3500] into the IoT, the IoT can deploy IoT applications according to this type of capability. In such a way, IoT applications can be used more flexibly.

Things as a Service (TaaS) can be considered as a cloud service category whose things-related services (e.g., accessing, subscription and notification of things-related data, and management and control of things-related devices) are provided to cloud service customers. TaaS offers cloud capabilities of the platform capabilities type and/or application capabilities type based on the IoT infrastructure. Via TaaS, IoT applications and/or IoT users can easily use the desired things-related services (e.g., get the desired things-related data, and control the desired things-related devices).

In order to integrate cloud computing technologies for implementation of TaaS, the additional capabilities of publishing things as services, summarizing data of things, and synchronizing data with things are required.

- The capability of publishing things as services involves the abilities of adopting the capabilities specified in the platform capabilities type of cloud to deploy things-related services [C-8-3].
- The capability of summarizing data of things involves the abilities of collecting and aggregating in the service support and application support layer of the IoT reference model [ITU-T Y.2060] the data of things related with the required things-related services for their provision to cloud users [C-8-4].
- The capability of synchronizing data with things involves the abilities of creating, deleting, and updating the data of things just in time with respect to application requirements, based on the updated status of sensed things in order to guarantee the quality of TaaS [C-8-5].

NOTE – The above identified capabilities of publishing things as services, summarizing data of things and synchronizing data with things are additional capabilities that are required for the integration of cloud computing technologies with the IoT in order to provide TaaS, they are not part of the IoT basic capabilities specified in clause 8 of this Recommendation.

9.2 Capabilities for integration of big data technologies

The development of the IoT causes rapid growth of IoT data. As there will be a large number of devices connected with the IoT and a large number of IoT services will flourish, there will be a large amount of IoT data created and used in the IoT.

Major characteristics of IoT data relevant for the integration of big data technologies into the IoT are:

a) Massive quantity (volume): the IoT infrastructure connects countless physical things and virtual things. These interactions and data aggregations produce huge data in the information world [ITU-T Y.2060], including not only the data collected by sensors, but also other data of the IoT infrastructure, e.g., other data concerning devices and data concerning networks, platforms and applications.

b) Heterogeneity (variety): IoT data are heterogeneous (data types and sources). For instance, IoT data related to healthcare applications usually differ from those related to transportation applications, not only from a structure (i.e., format) point of view but also from a semantic point of view.

c) Coexistence of structured and un-structured data (variety): structured data and un-structured data coexist in the IoT. Structured data are generally more efficient than un-structured data for data management.

d) High speed of data generation and processing (velocity): the collection and aggregation at the IoT device layer of data from a huge number of data sources form constantly large and high speed flows; the support of IoT application requirements may require processing of the data of things at high speed, e.g., for applications requiring real-time decision making.

e) Frequent update or change of the values of data (volatility): the values of data of things are changed or updated frequently over a period of time as they have to reflect the status of things, and update the services related with things, just in time as required by IoT applications.

f) Contextualization: a lot of IoT data are meaningful only if they are collected and integrated with related contextual data in order to provide context-aware services.

Data management capability of the IoT can be enhanced by big data technologies for transferring, storing, processing, validating and querying IoT data more efficiently, as well as for extracting information and actionable knowledge from IoT data. Besides, additional capabilities for the integration of big data technologies are also required.

The required additional capabilities for the integration of big data technologies are the capabilities of adopting big data collection, adopting big data aggregation, adopting big data storage, adopting big data integration, adopting big data query, and adopting big data analysis.

- The capability of adopting big data collection involves the abilities of adopting the technologies of big data collection [C-9-1].
- The capability of adopting big data aggregation involves the abilities of adopting the technologies of big data aggregating from different sources in the device layer of the IoT reference model as specified in [ITU-T Y.2060] [C-9-2].
- The capability of adopting big data storage involves the abilities of adopting the technologies of big data storing [C-9-3].
- The capability of adopting big data integration involves the abilities of adopting the technologies of big data summarizing from different sources in the service support and application support layer of the IoT reference model as specified in [ITU-T Y.2060] [C-9-4].
- The capability of adopting big data query involves the abilities of adopting the technologies of big data query [C-9-5].
- The capability of adopting big data analysis involves the abilities of adopting the technologies of big data analysis [C-9-6].

10 Security considerations

Security is a fundamental aspect to be considered in IoT technical specifications. The security issues in IoT can be divided into two groups: one is about the usual security threats, and another is about privacy protection that is particularly significant in IoT. This Recommendation considers the security issues from both the IoT functional framework perspective and the IoT basic capabilities perspective.

In the functional view of the IoT functional framework described in clause 7.1, the security and privacy protection functional group is specified. In the implementation view of the IoT functional framework described in clause 7.2, the IoT security and privacy protection functional entity is specified. In the deployment view of the IoT functional framework described in clause 7.3, the functional components of device manager, network manager, and service manager are specified to contain the capabilities of security and privacy protection.

Among the IoT basic capabilities described in clause 8, the capabilities of security and privacy protection are described in clause 8.7, and the management capabilities related with security and privacy protection are described in clause 8.5. These capabilities fulfil the IoT common requirements on security and privacy protection specified in [ITU-T Y.2066].

Annex A

The IoT capabilities list

(This annex forms an integral part of this Recommendation.)

Tables A.1 to A.9 list and number the capabilities identified in this Recommendation.

Tables A.1 to A.9 have similar formats.

The first column is headed "capability number" and assigns a number to each IoT capability. The numbering rule for each IoT capability is as follows: C-<the sub-clause number of clause 8 or 7+sub-clause number of clause 9>-<the sequence number of each IoT capability in each sub-clause>. For example, the first IoT capability described in clause 8.1 is numbered C-1-1.

The second column is headed "capability name" and gives the name of each IoT capability.

The third column is headed "capability summary" and briefly describes what the capability does.

The fourth column is headed as "related requirement(s)" and describes the common requirement(s) specified in [ITU-T Y.2066] to be fulfilled by the capability.

NOTE – One IoT capability may fulfil several requirements, and several IoT capabilities may fulfil the same single requirement.

The fifth column is headed "associated component(s)" and lists the functional components (from clause 7.3) associated with the IoT capability. This column can be used to validate that the IoT capability can be implemented and deployed.

Table A.1 – List of service provision capabilities

Capability number	Capability name	Capability summary	Related requirement(s)	Associated component(s)
C-1-1	Service prioritization	Service prioritization involves the abilities of providing services in different priorities, such as querying data or transferring data in different priorities.	Prioritization of services is required to fulfil the different service requirements of different groups of IoT users.	Service platform, IoT service controller
C-1-2	Semantic based service	Semantic based service involves the abilities of semantic annotating data or service, semantic querying data, or semantic requesting services.	Semantic based services are required to support autonomic service provisioning.	Service platform, IoT data server
C-1-3	Service composition	Service composition involves the abilities of creating customized services based on existing capabilities.	Service composition is required to support flexible service creation.	Service platform, service manager

Table A.1 – List of service provision capabilities

Capability number	Capability name	Capability summary	Related requirement(s)	Associated component(s)
C-1-4	Mobility service	Mobility service involves the abilities of remote accessing service platform, remote authenticating users, and remote requesting services.	Mobility services are required to support service mobility, user mobility and device mobility.	Service platform, service manager
C-1-5	Autonomic service	Autonomic service involves the abilities of automatic capturing, transferring, and analysing data of things, and automatic providing services based on predefined rules or policies.	Autonomic services are required to enable automatic capture, communication and processing of data of things.	Service platform, service manager, IoT service controller
C-1-6	Location-based and context-aware service	Location-based and context-aware service involves the abilities of automatic providing services based on the location information and related context and predefined rules or policies.	Location-based and context-aware services are required to enable flexible, user customized and autonomic services based on the location information and/or related context.	Service platform, service manager, IoT service controller
C-1-7	Service discovery	Service discovery involves the abilities of discovering IoT users, services, devices and data of things.	Discovery services are required to support discover IoT users, services, devices and data of things.	Service manager, IoT data server, device manager
C-1-8	Service subscription	Service subscription involves the abilities of subscribing the needed services and associated data of things by IoT users.	Service subscription support is required to allow the IoT user to subscribe the needed services and associated data of things.	Service manager, IoT data server
C-1-9	Standardized naming and addressing	Standardized naming and addressing involves the abilities of creating, updating, deleting, querying names and addresses of users, devices and things.	Standardized naming and addressing is required to support interoperability among different domains.	Service manager, network manager, device manager

Table A.1 – List of service provision capabilities

Capability number	Capability name	Capability summary	Related requirement(s)	Associated component(s)
C-1-10	Virtual storage and processing	Virtual storage and processing involves the abilities of providing storage and processing resources in a scalable way.	Virtual storage and processing capabilities are required to support storing and processing a large amount of data (big data).	Service platform, IoT data server
C-1-11	Adaptable service provision	The capability of adaptable service provision involves the abilities of extending service configurations to provide new services as required by applications or users of the IoT in order to be adaptable to different applications or users of the IoT.	Adaptability to the new technologies emerging in the future is required in the IoT.	Service platform, service manager
C-1-12	Service provision acknowledgement	The capability of service provision acknowledgement involves the abilities of acknowledging the correct service provision requested by applications or users of the IoT to support reliable service provision in the IoT.	Reliability in capabilities of the IoT, such as reliability in communication, service and data management capabilities of the IoT, is required.	Service platform

Table A.2 – List of communication capabilities

Capability number	Capability name	Capability summary	Related requirement(s)	Associated component(s)
C-2-1	Event-based communication	Event-based communication capability enables IoT devices and service provider to initiate communication based on predefined events.	Event-based communication between devices or between IoT users is required to be supported.	IoT device, IoT gateway, end-user device, service platform
C-2-2	Periodic communication	Periodic communication capability enables IoT devices and service provider to periodically initiate communication based on predefined events.	Periodic communication between devices or between IoT users is required to be supported.	IoT device, IoT gateway, end-user device, service platform

Table A.2 – List of communication capabilities

Capability number	Capability name	Capability summary	Related requirement(s)	Associated component(s)
C-2-3	Unicast communication	Unicast communication capability enables transport network to transfer messages from the source entity to single destination entity.	The support of the unicast communication mode between IoT users or devices is required.	Enhanced transport network
C-2-4	Multicast communication	Multicast communication capability enables transport network to transfer messages from the source entity to a group of destination entities simultaneously	The support of the multicast communication modes is required to provide communication services within a group of IoT users or devices.	Enhanced transport network
C-2-5	Broadcast communication	Broadcast communication capability enables transport network to transfer messages to all nodes of a given network area	The support of the broadcast communication modes is required to support the collaboration among IoT users or devices.	Enhanced transport network
C-2-6	Anycast communication	Anycast communication capability enables transport network to transfer messages to any one node of a given network area.	The support of the anycast communication modes is required to support the collaboration among IoT users or devices.	Enhanced transport network
C-2-7	Error control for communications	Error control for communications capability involves the abilities of ensuring to transfer data correctly from end to end.	Error control for communications is required to be supported.	IoT device, IoT gateway, end-user device, IoT network controller
C-2-8	Quality of Service enabling communication	Quality of Service enabling communication capability provides some mechanisms to guarantee the delivery and processing of time-critical messages	Time-critical communications are required to be supported.	IoT device, IoT gateway, end-user device, IoT network controller

Table A.2 – List of communication capabilities

Capability number	Capability name	Capability summary	Related requirement(s)	Associated component(s)
C-2-9	Self-configuring for networking	The capability of self-configuring for networking involves the abilities of automatically configuring networking parameters based on discovered networking interfaces and predefined rules.	Autonomic networking is required.	IoT device, IoT gateway, end-user device, network manager
C-2-10	Self-healing for networking	The capability of self-healing for networking involves the abilities of automatically recovering from fault status of networking based on monitoring results and predefined rules.	Autonomic networking is required.	IoT device, IoT gateway, end-user device, network manager
C-2-11	Self-optimizing for networking	The capability of self-optimizing for networking involves the abilities of automatically optimizing networking operations based on monitoring results and predefined rules.	Autonomic networking is required.	IoT device, IoT gateway, end-user device, network manager
C-2-12	Self-protecting for networking	The capability of self-protecting for networking involves the abilities of automatically protecting entities of networking from harmful operations based on predefined rules.	Autonomic networking is required.	IoT device, IoT gateway, end-user device, network manager
C-2-13	Content-aware communication	Content-aware communication capability involves the abilities of identifying content and selecting path and routing messages based on content, or blocking messages based on content.	Content-aware communication is required for support of path selection and routing of communications based on content.	IoT device, IoT gateway, end-user device, IoT network controller
C-2-14	Location-based communication	The capability of location-based communication involves the abilities of identifying locations and initiating communication based on predefined rules.	Location-based communication is required.	IoT device, IoT gateway, end-user device, IoT network controller

Table A.2 – List of communication capabilities

Capability number	Capability name	Capability summary	Related requirement(s)	Associated component(s)
C-2-15	Transport acknowledgement	The capability of transport acknowledgement involves the abilities of acknowledging correctly received messages to support reliable communications as required by IoT applications.	Reliability in capabilities of IoT, such as reliability in communication, service and data management capabilities of the IoT, is required.	Enhanced transport network, IoT device, IoT gateway, end-user device
C-2-16	Adaptable networking	The capability of adaptable networking involves the abilities of extending networking configurations to connect with emerging IoT to be adaptable to different networking technologies.	Adaptability to the new technologies emerging in the future is required in the IoT.	Enhanced transport network, network manager, IoT device, IoT gateway, end-user device, device manager

Table A.3 – List of application support capabilities

Capability number	Capability name	Capability summary	Related requirement(s)	Associated component(s)
C-3-1	Programmable interface provision	The capability of programmable interface provision involves the abilities of supporting new services or customized services based on existing capabilities and application specific requirements.	Programmable interfaces are required to be standardized to provide open access to application support capabilities.	Service platform
C-3-2	Group management	The capability of group management involves the abilities of creating, modifying, deleting, and querying IoT groups, and adding, modifying, deleting and querying IoT group members.	Group management is required.	Service platform

Table A.3 – List of application support capabilities

Capability number	Capability name	Capability summary	Related requirement(s)	Associated component(s)
C-3-3	Time synchronization	The capability of time synchronization involves the abilities of synchronizing the time among related functional components in reliable way, in order to support global or local time stamping for applications.	Reliable time synchronization is required.	Service platform
C-3-4	Orchestration	Orchestration capability involves the abilities of automatic arrangement and coordination of service provisioning or device operations to fulfil application specific requirements.	Collaboration is required.	IoT device, IoT gateway, end-user device, service platform
C-3-5	User management	User management capability involves the abilities of creating, querying, updating and deleting IoT user profiles, and authenticating, authorizing, registering and auditing IoT users.	User management is required.	Service platform
C-3-6	Application support operation acknowledgement	The capability of application support operation acknowledgement involves the abilities of acknowledging correct operations requested by applications to support reliable application operations in the IoT.	Reliability in capabilities of the IoT, such as reliability in communication, service and data management capabilities of the IoT, is required.	IoT data server, IoT gateway

Table A.4 – List of data management capabilities

Capability number	Capability name	Capability summary	Related requirement(s)	Associated component(s)
C-4-1	Data storage	The capability of data storage involves the ability of storing data of things based on predefined rules and policies.	Storing data of things is required to be supported.	IoT data server, IoT gateway
C-4-2	Data processing	The capability of data processing involves the ability of data fusion and mining based on predefined rules and policies.	Processing data of things is required to be supported.	IoT data server
C-4-3	Data querying	The capability of data querying involves the ability of querying historical information about things	Querying historical data of things is required to be supported.	IoT data server
C-4-4	Data access control	The capability of data access control involves the abilities of controlling and monitoring the data access operations by the owners of data.	Data access control by the data owners is required.	IoT data server
C-4-5	Open information exchange	The capability of open information exchange involves the abilities of sending data to or receiving data from external data sources, e.g., data centres and data servers outside the IoT.	Data exchange with entities outside the IoT is required to be supported.	IoT data server
C-4-6	Semantic data operation	The capability of semantic data operation involves the abilities of semantic annotating, semantic discovering, semantic storing and semantic composing data of things to fulfil the requirements of IoT users or applications	Semantic annotation and semantic access to data of things are required. Semantic storage, transfer and aggregation of data of things are required.	IoT data server, IoT gateway
C-4-7	Autonomic data operation	The capability of autonomic data operation involves the abilities of automatically collecting, aggregating, transferring, storing, analysing data of things, and automatically managing these data operations for support of operating data of things in a scalable way.	Scalability is required to be supported in the IoT in order to handle a large amount of devices, applications and users.	IoT device, IoT gateway, end-user device, IoT data server

Table A.5 – List of management capabilities

Capability number	Capability name	Capability summary	Related requirement(s)	Associated component(s)
C-5-1	Managing data models for exchanging data of things	The capability of managing data models for exchanging data of things involves the abilities of creating, querying and updating data models for support of interoperability among IoT applications. This capability also includes the abilities of creating and updating data models for support of semantic interoperability among IoT applications.	Interoperability is required to be ensured among heterogeneous IoT implementations.	IoT data server
C-5-2	Managing service description	The capability of managing service description involves the abilities of creating, querying and updating service description for support of service interoperability.	Interoperability is required to be ensured among heterogeneous IoT implementations.	Service manager, service platform
C-5-3	Managing network configuration	The capability of managing network configuration involves the abilities of creating, querying and updating network configuration for support of network interoperability.	Interoperability is required to be ensured among heterogeneous IoT implementations.	Network manager, enhanced transport network
C-5-4	Managing device configuration	The capability of managing device configuration involves the abilities of creating, querying and updating network configuration for support of device interoperability.	Support for heterogeneous device related communication technologies is required.	Device manager, IoT device, IoT gateway, end-user device
C-5-5	Managing security policy	The capability of managing security policy involves the abilities of creating, querying and updating security policy for support of interoperability between different implementations of security policy.	Interoperability is required to be ensured among heterogeneous IoT implementations.	Service manager, network manager, device manager

Table A.5 – List of management capabilities

Capability number	Capability name	Capability summary	Related requirement(s)	Associated component(s)
C-5-6	Managing privacy protection policy	The capability of managing privacy protection policy involves the abilities of creating, querying and updating privacy protection policy to support interoperability between different implementations of privacy protection policy.	Interoperability is required to be ensured among heterogeneous IoT implementations.	Service manager, network manager, device manager
C-5-7	Managing distributed processing	The capability of managing distributed processing involves the abilities of managing IoT functional components in a distributed way to support IoT scalability.	Scalability is required to be supported in IoT in order to handle a large number of devices, applications and users.	IoT data server, IoT service controller, service platform, service manager, IoT network controller, network manager, enhanced transport network, IoT gateway, device manager
C-5-8	Managing multiple domains	The capability of managing multiple domains involves the abilities of managing IoT functional components in multiple administrative domains to support of IoT scalability.	Scalability is required to be supported in IoT in order to handle a large number of devices, applications and users.	IoT data server, IoT service controller, service platform, service manager, IoT network controller, network manager, enhanced transport network, device manager
C-5-9	Redundant deployment enablement	The capability of redundant deployment enablement involves the abilities of enabling deployment of redundant functional components of the IoT to guarantee reliability required in communication, service provision and data management.	Reliability in capabilities of IoT, such as reliability in communication, service and data management capabilities of IoT, is required.	IoT data server, IoT service controller, service platform, service manager, IoT network controller, network manager, IoT gateway, device manager

Table A.5 – List of management capabilities

Capability number	Capability name	Capability summary	Related requirement(s)	Associated component(s)
C-5-10	Service integrity check	The capability of service integrity check involves the abilities of checking the service lifetime, the available resources required to provide the service in order to guarantee the high availability of service provisioning.	The IoT is required to provide high availability in service provisioning, data management, communication, sensing and actuating things.	Service manager
C-5-11	Data integrity check	The capability of data integrity check involves the abilities of checking the data lifetime, the available attributes of the data, and the consistency of data in order to guarantee the high availability of data management.	The IoT is required to provide high availability in service provisioning, data management, communication, sensing and actuating things.	IoT data server
C-5-12	Device integrity check	The capability of device integrity check involves the abilities of checking the status of all functions of device to guarantee the high availability of IoT devices.	The IoT is required to provide high availability in service provisioning, data management, communication, sensing and actuating things.	Device manager, IoT device, IoT gateway, end-user device
C-5-13	Security integrity check	The capability of security integrity check involves the abilities of checking the consistency of security policies deployed in all functional components of the IoT to guarantee the high availability of security in the IoT.	The IoT is required to provide high availability in service provisioning, data management, communication, sensing and actuating things.	Service manager, network manager, device manager, IoT device, IoT gateway, end-user device
C-5-14	User profile integrity check	The capability of user profile integrity check involves the abilities of checking the lifetime, subscription, privacy protection, and availability of services subscribed by users to guarantee the high availability of service provisioning and privacy protection for users.	The IoT is required to provide high availability in service provisioning, data management, communication, sensing and actuating things.	Service manager, network manager, device manager, IoT device, IoT gateway, end-user device

Table A.5 – List of management capabilities

Capability number	Capability name	Capability summary	Related requirement(s)	Associated component(s)
C-5-15	Managing devices	The capability of managing devices involves the abilities of configuring, monitoring, diagnosing, and recovering devices of the IoT, and updating software to enhance capabilities of devices of the IoT.	Manageability is required to be supported in IoT in order to ensure normal operations.	Device manager
C-5-16	Managing networks	The capability of managing networks involves the abilities of configuring, monitoring, accounting and charging, optimizing, diagnosing, and recovering networks of the IoT, and updating network software to enhance capabilities of networks of the IoT.	Manageability is required to be supported in IoT in order to ensure normal operations.	Network manager
C-5-17	Managing services	The capability of managing services involves the abilities of describing, configuring, monitoring, accounting and charging, optimizing, recovering, and updating services of the IoT.	Manageability is required to be supported in IoT in order to ensure normal operations.	Service manager
C-5-18	Managing data operations	The capability of managing data operations involves the abilities of configuring, monitoring, accounting and charging, optimizing, and recovering data operations, and updating software related with data operations to enhance capabilities of the IoT.	Manageability is required to be supported in IoT in order to ensure normal operations.	IoT data server
C-5-19	Managing security operations	The capability of managing security operations involves the abilities of configuring, monitoring, auditing, diagnosing, and recovering security operations, and updating software related with security operations to enhance capabilities of the IoT.	Manageability is required to be supported in IoT in order to ensure normal operations.	Service manager, network manager, device manager

Table A.5 – List of management capabilities

Capability number	Capability name	Capability summary	Related requirement(s)	Associated component(s)
C-5-20	Managing privacy protection	The capability of managing privacy protection involves the abilities of configuring, monitoring, auditing, and recovering privacy protection, and updating software related with privacy protection to enhance capabilities of the IoT.	Manageability is required to be supported in IoT in order to ensure normal operations.	Service manager, network manager, device manager
C-5-21	Managing user operations	The capability of managing user operations involves the abilities of configuring, monitoring, accounting and charging, optimizing, diagnosing, recovering operations of IoT users, and updating software related with operations of IoT users to enhance capabilities of the IoT.	Manageability is required to be supported in IoT in order to ensure normal operations.	Service manager, network manager, device manager
C-5-22	Plug and play capability	Plug and play capability involves the abilities of automatic configuring, connecting and activating devices of the IoT to enable on-the-fly semantic-based configuration and activation of IoT devices.	Plug and play capability is required.	IoT device, IoT gateway, end-user device, network manager, service manager

Table A.6 – List of connectivity capabilities

Capability number	Capability name	Capability summary	Related requirement(s)	Associated component(s)
C-6-1	Identification-based connectivity	The capability of identification-based connectivity involves the abilities of establishing the connectivity based on the identification of things.	Identification-based connectivity between a thing and the IoT is required.	IoT device, IoT gateway, end-user device, network manager
C-6-2	Things' status notification	The capability of things' status notification involves the abilities of automatic notification of the status of things and its changes based on predefined rules.	Monitoring things in timely manner is required.	IoT device, IoT gateway, end-user device

Table A.6 – List of connectivity capabilities

Capability number	Capability name	Capability summary	Related requirement(s)	Associated component(s)
C-6-3	Device mobility	The capability of device mobility involves the abilities of keeping the connectivity with the IoT when a device moves.	Device mobility is required.	IoT device, IoT gateway, end-user device, network manager
C-6-4	Adaptable connectivity	The capability of adaptable connectivity involves the abilities of extending connectivity configurations to connect with new types of devices of the IoT in order to be adaptable to different technologies in devices of IoT.	Adaptability to the new technologies emerging in the future is required in IoT.	IoT device, IoT gateway, end-user device, device manager

Table A.7 – List of security and privacy protection capabilities

Capability number	Capability name	Capability summary	Related requirement(s)	Associated component(s)
C-7-1	Communication security	The capability of communication security involves the abilities of supporting secure, trusted and privacy-protected communication.	Communication security is required.	IoT device, IoT gateway, end-user device, device manager, network manager, enhanced transport network
C-7-2	Data management security	The capability of data management security involves the abilities of providing secure, trusted and privacy-protected data management.	Data management security is required.	IoT data server, IoT gateway
C-7-3	Service provision security	The capability of service provision security involves the abilities of providing secure, trusted and privacy-protected service provision.	Service provision security is required.	Service platform, service manager
C-7-4	Security integration	The capability of security integration involves the abilities of integrating different security policies and techniques related to the variety of IoT functional components.	Integration of different security policies and techniques is required.	Device manager, network manager, service manager

Table A.7 – List of security and privacy protection capabilities

Capability number	Capability name	Capability summary	Related requirement(s)	Associated component(s)
C-7-5	Mutual authentication and authorization	The capability of mutual authentication and authorization involves the abilities of authenticating and authorizing each other before a device accesses IoT based on predefined security policies.	Mutual authentication and authorization is required.	Device manager, network manager
C-7-6	Security audit	The capability of security audit involves the abilities of monitoring any data access or attempt to access IoT applications in a fully transparent, traceable and reproducible way based on appropriate regulation and laws.	Security audit is required to be supported in IoT.	Device manager, network manager, service manager, IoT data server

Table A.8 – List of capabilities for integration of cloud computing technologies

Capability number	Capability name	Capability summary	Related requirement(s)	Associated component(s)
C-8-1	Accessing virtual processing resources	The capability of accessing virtual processing resources involves the abilities of adopting the capabilities specified in infrastructure capabilities type of cloud to use processing resource deployed in cloud.	Integration with cloud computing technologies is required.	IoT data server, IoT gateway
C-8-2	Accessing virtual storage resources	The capability of accessing virtual storage resources involves the abilities of adopting the capabilities specified in infrastructure capabilities type of cloud to use storage resource deployed in cloud.	Integration with cloud computing technologies is required.	IoT data server, IoT gateway
C-8-3	Publishing things as services	The capability of publishing things as services involves the abilities of adopting the capabilities specified in platform capabilities type of cloud to deploy the services of things.	Integration with cloud computing technologies is required.	IoT data server, service platform

Table A.8 – List of capabilities for integration of cloud computing technologies

Capability number	Capability name	Capability summary	Related requirement(s)	Associated component(s)
C-8-4	Summarizing data of things	The capability of summarizing data of things involves the abilities of collecting and aggregating the data of things related with the required services to provide the service of things to cloud users.	Integration with cloud computing technologies is required.	IoT data server, service platform, IoT device, IoT gateway, end-user device
C-8-5	Synchronizing data with things	The capability of synchronizing data with things involves the abilities of creating, deleting, and updating the data of things just in time with respect to application requirements, based on the updated status of sensed things in order to guarantee the quality of TaaS.	Integration with cloud computing technologies is required.	IoT data server, service platform, IoT device, IoT gateway, end-user device

Table A.9 – List of capabilities for integration of big data technologies

Capability number	Capability name	Capability summary	Related requirement(s)	Associated component(s)
C-9-1	Adopting big data collection	The capability of adopting big data collection involves the abilities of adopting the technologies of big data collection.	Integration with big data technologies is required.	IoT data server, IoT gateway
C-9-2	Adopting big data aggregation	The capability of adopting big data aggregation involves the abilities of adopting the technologies of summarizing big data from different sources in device layer.	Integration with big data technologies is required.	IoT device, IoT gateway, end-user device
C-9-3	Adopting big data storage	The capability of adopting big data storage involves the abilities of adopting the technologies of storing big data.	Integration with big data technologies is required.	IoT data server

Table A.9 – List of capabilities for integration of big data technologies

Capability number	Capability name	Capability summary	Related requirement(s)	Associated component(s)
C-9-4	Adopting big data integration	The capability of adopting big data integration involves the abilities of adopting the technologies of summarizing big data from different sources in server support and application support layer.	Integration with big data technologies is required.	IoT data server, IoT gateway
C-9-5	Adopting big data query	The capability of adopting big data query involves the abilities of adopting the technologies of big data query.	Integration with big data technologies is required.	IoT data server
C-9-6	Adopting big data analysis	The capability of adopting big data analysis involves the abilities of adopting the technologies of big data analysis.	Integration with big data technologies is required.	IoT data server

Appendix I

Matching analysis between requirements and capabilities of the IoT

(This Appendix does not form an integral part of this Recommendation.)

Tables I.1 to I.7 provide matching analyses between requirements and capabilities of the IoT. The following provides a legend for the structure of these tables.

The two columns headed "Requirement number" and "Requirement summary" are copied from the corresponding columns in the table in Annex A of [ITU-T Y.2066].

The column headed "Capability number" contains one or multiple capability numbers provided in Annex A whose corresponding capabilities support the requirement listed in the same row.

The column headed "Capability name" contains the name of the capabilities associated with the "Capability number" provided in the same row. These capabilities are described in clause 8 and support the requirement listed in the same row.

I.1 Matching analysis of non-functional requirements of the IoT

Matching analysis results between non-functional requirements of the IoT and the supported capabilities of the IoT are shown in Table I.1. Results show that all non-functional requirements specified in [ITU-T Y.2066] are fulfilled.

NOTE – There are multiple capabilities associated with each row of Table I.1. These capabilities act together to support the requirement in the same row.

Table I.1 – List of matching analysis of non-functional requirements of the IoT

Requirement number	Requirement summary	Capability number	Capability name
N1	Interoperability is required.	C-5-1, C-5-2, C-5-3, C-5-4, C-5-5, and C-5-6.	Managing data models for exchanging data of things, managing service description, managing network configuration, managing device configuration, managing security policy, and managing privacy protection policy.
N2	Scalability is required.	C-4-7, C-5-7, and C-5-8.	Autonomic data operation, managing distributed processing, and managing multiple domains.
N3	Reliability is required.	C-1-12, C-2-15, C-3-6, and C-5-9.	Service provision acknowledgement, transport acknowledgement, application support operation acknowledgement, and redundant deployment enablement.

Table I.1 – List of matching analysis of non-functional requirements of the IoT

Requirement number	Requirement summary	Capability number	Capability name
N4	High availability is required.	C-5-10, C-5-11, C-5-12, C-5-13, and C-5-14.	Service integrity check, data integrity check, device integrity check, security integrity check, and user profile integrity check.
N5	Adaptability is required.	C-1-11, C-2-16, and C-6-4.	Adaptable service provision, adaptable networking, and adaptable connectivity.
N6	Manageability is required.	C-5-15, C-5-16, C-5-17, C-5-18, C-5-19, C-5-20, and C-5-21.	Managing devices, managing networks, managing services, managing data operations, managing security operations, managing privacy protection, and managing user operations.

I.2 Matching analysis of application support requirements of the IoT

Matching analysis results between application support requirements of the IoT and the capabilities of the IoT that can fulfil those requirements are shown in Table I.2. Results show that all application support requirements specified in [ITU-T Y.2066] are fulfilled.

NOTE – There may be multiple capabilities associated with a single row of Table I.2. Where multiple capabilities are listed, they act together to support the requirement in the same row.

Table I.2 – List of matching analysis of application support requirements of the IoT

Requirement number	Requirement summary	Capability number	Capability name
A1	Standardized programmable interfaces are required.	C-3-1	Programmable interface provision
A2	Group management is required.	C-3-2	Group management
A3	Reliable time synchronization is required.	C-3-3	Time synchronization
A4	Collaboration is required.	C-3-4	Orchestration
A5	User management is required.	C-3-5	User management
A6	Resource usage accounting is required.	C-5-16, C-5-17, and C-5-18	Managing networks, managing services, and managing data operations.

I.3 Matching analysis of service requirements of the IoT

Matching analysis results between service requirements of the IoT and the supported capabilities of the IoT are shown in Table I.3. Results show that all service requirements specified in [ITU-T Y.2066] are fulfilled.

NOTE – There may be multiple capabilities associated with a single row of Table I.3. Where multiple capabilities are listed, they act together to support the requirement in the same row.

Table I.3 – List of matching analysis of service requirements of the IoT

Requirement number	Requirement summary	Capability number	Capability name
S1	Prioritization of services is required.	C-1-1	Service prioritization
S2	Semantic based services are required.	C-1-2	Semantic based service
S3	Service composition is required.	C-1-3	Service composition
S4	Mobility services are required.	C-1-4	Mobility services
S5	Highly reliable and secure human body connectivity services are required.	C-1-12, C-2-15, C-3-6, C-7-1, C-7-2, and C-7-3.	Service provision acknowledgement, transport acknowledgement, application support operation acknowledgement, communication security, data management security, and service provision security.
S6	Autonomic services are required.	C-1-5	Autonomic service
S7	Location-based and context-aware services are required.	C-1-6	Location-based and context-aware service
S8	Service management is required.	C-5-17	Managing services
S9	Discovery services are required.	C-1-7	Service discovery
S10	Service subscription support is required.	C-1-8	Service subscription
S11	Standardized naming and addressing is required.	C-1-9	Standardized naming and addressing
S12	Virtual storage and processing capabilities are required.	C-1-10	Virtual storage and processing

I.4 Matching analysis of communication requirements of the IoT

Matching analysis results between communication requirements of the IoT and the supported capabilities of the IoT are shown in Table I.4. Results show that all communication requirements specified in [ITU-T Y.2066] are fulfilled.

NOTE – There may be multiple capabilities associated with a single row of Table I.4. Where multiple capabilities are listed, they act together to support the requirement in the same row. In the case of the row identified by Requirement number "C3", each of the two identified capabilities can fulfil the requirement.

Table I.4 –List of matching analysis of communication requirements of IoT

Requirement number	Requirement summary	Capability number	Capability name
C1	Event-based, periodic, and automatic communication modes are required to be supported.	C-2-1 and C-2-2.	Event-based communication and periodic communication.
C2	The support of the unicast, multicast, broadcast and anycast communication modes is required.	C-2-3, C-2-4, C-2-5 and C-2-6.	Unicast communication, multicast communication, broadcast communication, and anycast communication.
C3	The support of device initiated communications is required.	C-2-1 or C-2-2.	Event-based communication or periodic communication.
C4	Error control for communications is required to be supported.	C-2-7	Error control for communications
C5	Time-critical communications are required to be supported.	C-2-8	Time-critical communications
C6	Autonomic networking is required.	C-2-9, C-2-10, C-2-11 and C-2-12.	Self-configuring for networking, self-healing for networking, self-optimizing for networking, and self-protecting for networking.
C7	Content-aware communication is required.	C-2-13	Content-aware communication
C8	Location-based communication is required.	C-2-14	Location-based communication
C9	Support for heterogeneous device related communication technologies is required.	C-5-4	Managing device configuration
C10	Support for heterogeneous network related communication technologies is required.	C-5-3	Managing network configuration

I.5 Matching analysis of device requirements of the IoT

Matching analysis results between device requirements of IoT and the supported capabilities of the IoT are shown in Table I.5. Results show that all device requirements specified in [ITU-T Y.2066] are fulfilled.

Table I.5 – List of matching analysis of device requirements of the IoT

Requirement number	Requirement summary	Capability number	Capability name
D1	Identification-based connectivity between a thing and the IoT is required.	C-6-1	Identification-based connectivity
D2	Remote monitoring, control and configuration of devices are required.	C-5-15	Managing devices
D3	Plug and play capability is required.	C-5-22	Plug and play capability
D4	Monitoring things in a timely manner is required.	C-6-2	Things' status notification
D5	Device mobility is required.	C-6-3	Device mobility
D6	Device integrity checking is required.	C-5-12	Device integrity check

I.6 Matching analysis of data management requirements of the IoT

Matching analysis results between data management requirements of the IoT and the supported capabilities of the IoT are shown in Table I.6. Results show that all data management requirements specified in [ITU-T Y.2066] are fulfilled.

Table I.6 – List of matching analysis of data management requirements of the IoT

Requirement number	Requirement summary	Capability number	Capability name
DM1	Storing data of things is required to be supported.	C-4-1	Data storage
DM2	Processing data of things is required to be supported.	C-4-2	Data processing
DM3	Querying historical data of things is required to be supported.	C-4-3	Data querying
DM4	Data access control by the data owners is required.	C-4-4	Data access control
DM5	Data exchange with entities outside the IoT is required to be supported.	C-4-5	Open information exchange
DM6	Integrity checking and life cycle management of data of things is required.	C-5-11	Data integrity check
DM7	Semantic annotation and semantic access to data of things are required.	C-4-6	Semantic data operation
DM8	Semantic storage, transfer and aggregation of data of things are required.	C-4-6	Semantic data operation

I.7 Matching analysis of security and privacy protection requirements of the IoT

Matching analysis results between security and privacy protection requirements of the IoT and the supported capabilities of the IoT are shown in Table I.7. Results show that all security and privacy protection requirements specified in [ITU-T Y.2066] are fulfilled.

Table I.7 – List of matching analysis of security and privacy protection requirements of the IoT

Requirement number	Requirement summary	Capability number	Capability name
SP1	Communication security is required.	C-7-1	Communication security
SP2	Data management security is required.	C-7-2	Data management security
SP3	Service provision security is required.	C-7-3	Service provision security
SP4	Integration of different security policies and techniques is required.	C-7-4	Security integration
SP5	Mutual authentication and authorization is required.	C-7-5	Mutual authentication and authorization
SP6	Security audit is required to be supported in the IoT.	C-7-6	Security audit

Bibliography

[b-ITU-T Y.2001] Recommendation ITU-T Y.2001 (2004), *General overview of NGN*.

[b-ITU-T Y.2061] Recommendation ITU-T Y.2061 (2012), *Requirements for support of machine-oriented communication applications in the next generation network environment*.

[b-ITU-T Y.2067] Recommendation ITU-T Y.2067 (2014), *Common requirements and capabilities of a gateway for Internet of things applications*.

[b-ITU-T Y.3500] Recommendation ITU-T Y.3500 (2014), *Information technology – Cloud computing – Overview and vocabulary*.

[b-ETSI TS 102 690] ETSI TS 102 690 v1.2.1 (2013), *Machine-to-Machine communications (M2M); Functional architecture*, http://www.etsi.org/standards

[b-IoT-A D1.4] The Internet of things architecture (IoT-A, 2012), *Project Deliverable D1.4 – Converged architectural reference model for the IoT v2.0*, http://www.iot-a.eu/public/public-documents/documents-1

Y.4402/F.747.4

Requirements and functional architecture for the open ubiquitous sensor network service platform

Requirements and functional architecture for the open ubiquitous sensor network service platform

Summary

Recommendation ITU-T F.747.4 defines the requirements and functional architecture for the open ubiquitous sensor network (USN) service platform.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T F.747.4	2013-12-14	16	11.1002/1000/12051-en

Keywords

Open USN service, semantic data, USN middleware, USN resource.

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

Table of Contents

Introduction

There is a large number of middleware available for sensor networks and different kinds of ubiquitous sensor network (USN) middleware that may be deployed. Furthermore, USN services may utilize widely distributed sensors or sensor networks through different USN middleware. In a widely distributed environment, USN applications need to know of the various USN middleware, sensors and sensor networks used. For example, if an application wants to get the current temperature in Geneva, the application should have information specifying which USN middleware and which sensors or sensor networks can provide the data requested and how they can provide the data.

The open USN service platform aims to provide unified access to USN resources and sensed data/semantic data through heterogeneous USN middleware, thereby enabling USN applications to take full advantage of the USN capabilities. It allows providers, users and application developers to provide USN resources, use USN services, or develop USN applications without needing to have specific knowledge about the USN middleware and sensors, or how to access specific sensor networks. The main purpose of the open USN service platform defined in this Recommendation is to provide:

– easy access to and use of the global USN resources and sensed data/semantic data;
– easy connection of USN resources; and
– easy development and distribution of various applications.

Recommendation ITU-T Y.4402/F.747.4

Requirements and functional architecture for the open ubiquitous sensor network service platform

1 Scope

The objective of this Recommendation is to define the open ubiquitous sensor network (USN) service platform and provide requirements and functional architecture for the open USN service platform.

The scope of this Recommendation includes:

– concept of the open USN service platform;

– requirements for the open USN service platform;

– functional architecture of the open USN service platform;

– functional entities of the open USN service platform.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T F.744] Recommendation ITU-T F.744 (2009), *Service description and requirements for ubiquitous sensor network middleware*.

[ITU-T Y.2201] Recommendation ITU-T Y.2201 (2009), *Requirements and capabilities for ITU-T NGN*.

[ITU-T Y.2221] Recommendation ITU-T Y.2221 (2010), *Requirements for support of ubiquitous sensor network (USN) applications and services in the NGN environment*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 context awareness [ITU-T Y.2201]: A capability to determine or influence a next action in telecommunication or process by referring to the status of relevant entities, which form a coherent environment as a context.

3.1.2 sensor [ITU-T Y.2221]: An electronic device that senses a physical condition or chemical compound and delivers an electronic signal proportional to the observed characteristic.

3.1.3 sensor network [ITU-T Y.2221]: A network comprised of interconnected sensor nodes exchanging sensed data by wired or wireless communication.

3.1.4 sensor node [ITU-T Y.2221]: A device consisting of sensor(s) and optional actuator(s) with capabilities of sensed data processing and networking.

3.1.5 sensed data [ITU-T F.744]: Data sensed by a sensor that is attached to a specific sensor node.

NOTE – The sensed data are collected from USN resources via USN middleware and stored in a sensed data repository.

3.1.6 ubiquitous sensor network (USN) [ITU-T Y.2221]: A conceptual network built over existing physical networks which makes use of sensed data and provides knowledge services to anyone, anywhere and at anytime, and where the information is generated by using context awareness.

3.1.7 USN middleware [ITU-T Y.2221]: A set of logical functions to support USN applications and services.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 metadata: Description of USN resources which include types of operations supported, attributes for those operations, etc.

3.2.2 open USN service: USN service which provides unified access to USN resources and sensed data/semantic data through heterogeneous USN middleware.

3.2.3 semantic data: Data translated into resource description framework (RDF) [b-RDF] form from metadata of USN resources and sensed data, and data processed by the Semantic inference functional entity (FE) from data represented in RDF form. These data are stored in a Semantic data repository.

3.2.4 USN resource: An entity that provides a USN service including sensor, actuator, sensor node, sensor network and gateway.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

API	Application Programming Interface
FE	Functional Entity
LOD	Linked Open Data
RDF	Resource Description Framework
REST	Representational State Transfer
SPARQL	SPARQL Protocol and RDF Query Language
URI	Uniform Resource Identifier
USN	Ubiquitous Sensor Network

5 Conventions

None.

6 Open USN service description and characteristics

USN services require USN applications to have knowledge of USN middleware and sensors or sensor networks in order to access USN resources. For example, heterogeneous USN middleware is not easily accessed by applications since each USN middleware may have proprietary application programming interfaces (APIs) which may hinder access to various USN resources attached to the USN middleware.

Even when applications have access to multiple USN middleware entities, the applications must search, collect, analyse and process the sensed data themselves.

These limitations can be overcome by providing a unified access method for USN resources and sensed data/semantic data via heterogeneous USN middleware.

Figure 6-1 shows the traditional USN service framework and the open USN service framework.

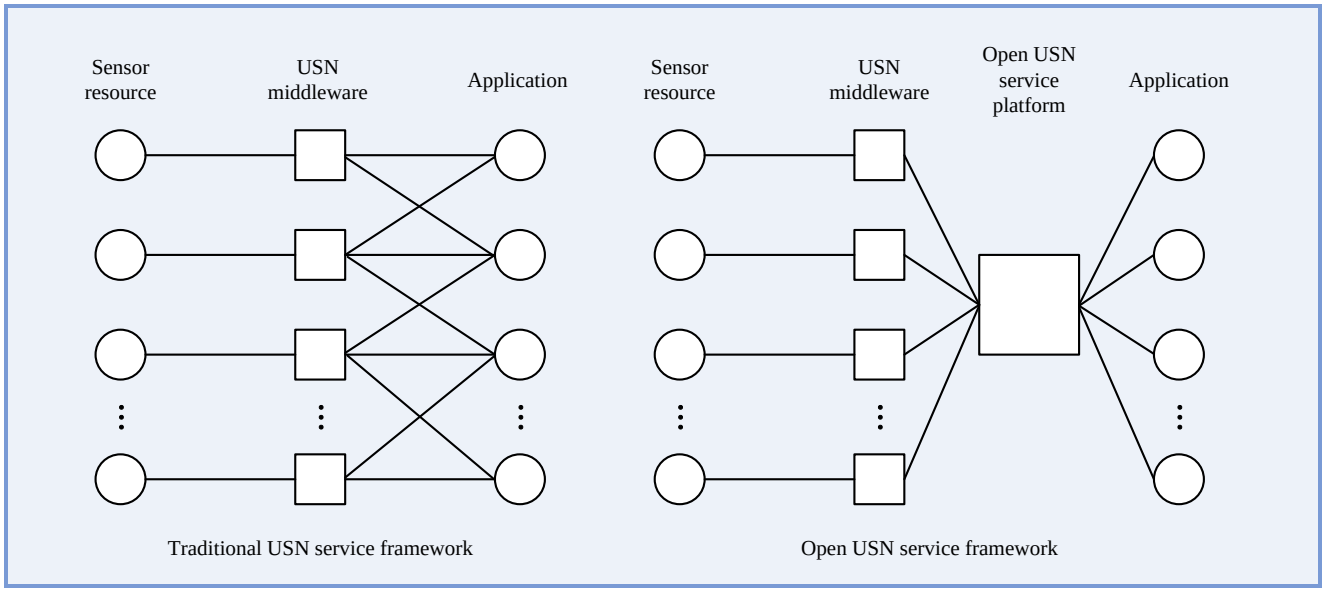


Figure 6-1 – Traditional USN service framework and open USN service framework

In the traditional USN service framework, each application needs to know how to access heterogeneous USN middleware and which USN resources should be accessed. In the open USN service framework, each application does not need to know how to access heterogeneous USN middleware nor which USN resources should be accessed.

Figure 6-2 shows heterogeneous USN middleware access provided by the open USN service platform.

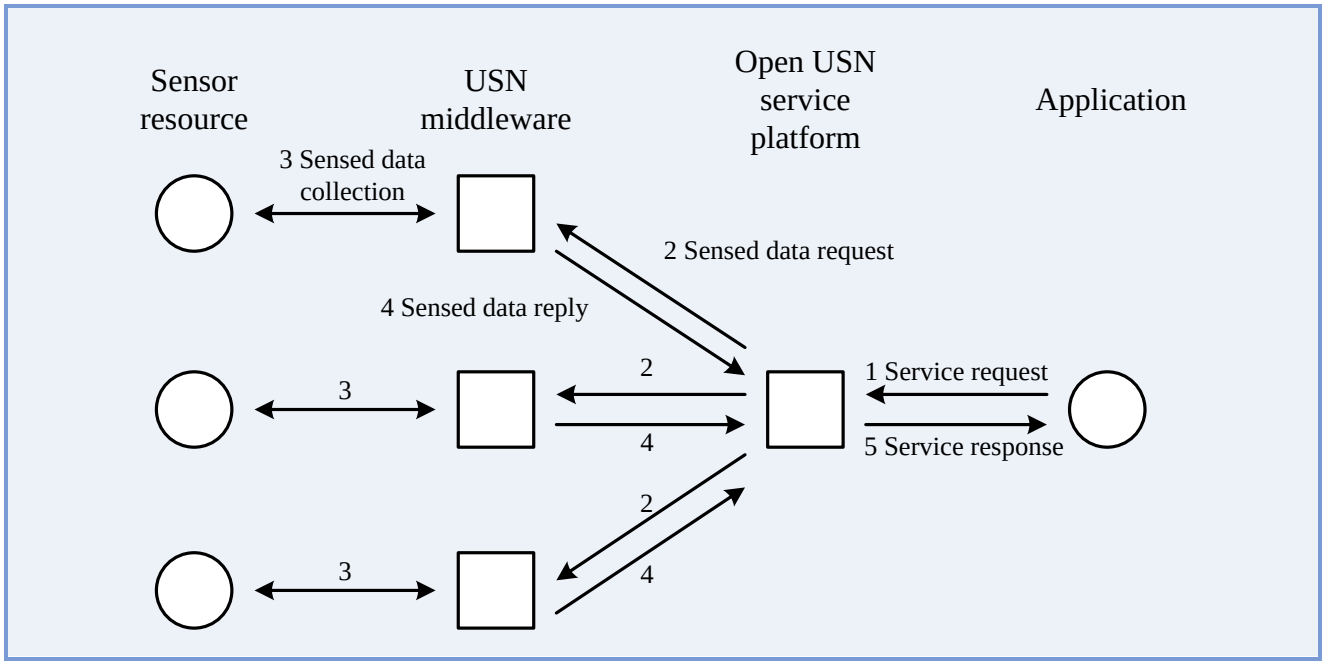


Figure 6-2 – Heterogeneous USN middleware access through the open USN service platform

In the open USN service framework, a USN application only needs to send requests to the open USN service platform; the remaining processing is done by the platform itself. The open USN service platform converts a request from each USN application into a specific request for different USN middleware.

The ultimate goal of the open USN service platform is to provide the application with the following services:

– easy access to and use of global USN resources and sensed data/semantic data;
– easy connection of USN resources;
– easy development and distribution of various applications.

7 Requirements for the open USN service platform

The following are the requirements for the open USN service platform.

7.1 Requirements to communicate with heterogeneous USN middleware

– It is required to provide an open interface for heterogeneous USN middleware to provide the sensed data and metadata received from USN resources.
– USN resources and semantic data are required to be identified by a universal resource identifier (URI).
– URIs for USN resources are required to be dynamically assigned when the USN resources are registered to the open USN service platform.
– It is required to provide open interface for accessing heterogeneous USN middleware.

7.2 Requirements of the open USN service platform

– USN resource management is required according to proper management policies on authentication, authorization and access rights.
– The characteristics and status of USN resources are required to be managed.
– It is required to provide functionality of inheritance and binding of USN middleware management policy.
– It is recommended to manage a logical group of USN resources according to application service requests.
– It is recommended to provide inference functions to derive the context data by user rules.

7.3 Requirements for linking the LOD

– It is required to be accessed by external linked open data (LOD) [b-LOD] by assigning a unique URI to each USN resource and semantic data.
– It is required to access the external LOD via the web.

7.4 Requirements for applications

– It is required to provide an open protocol, such as representational state transfer (REST), for applications.
– It is required to provide an open interface to services and applications for the combination of existing applications.

7.5 Requirements for USN resources and sensed data/semantic data

– Metadata of USN resources and semantic data are required to be represented in RDF [b-RDF] format.

– It is required to provide functions to store, query, modify and delete data in RDF form in the repository.
– It is required to provide functions to search USN resources and sensed data/semantic data by analysing the intention of service requests from an application.
– It is required to support a standard query language such as SPARQL Protocol and RDF Query Language (SPARQL) [b-SPARQL].
– It is recommended to manage access for USN resources and sensed data/semantic data according to proper access rights.

8 Functional architecture of the open USN service framework

8.1 Functional architecture

Figure 8-1 shows the functional architecture of the open USN service framework.

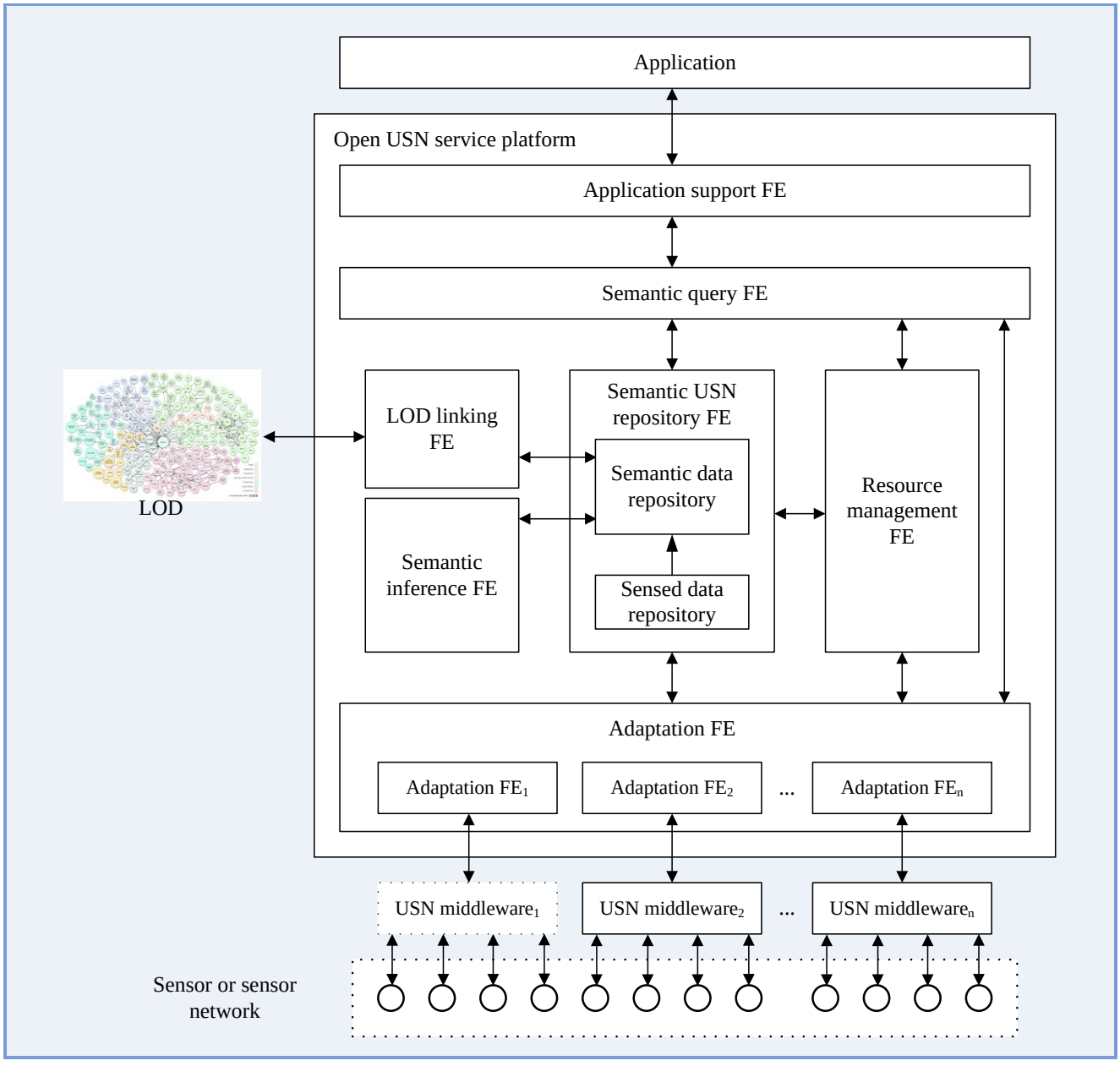


Figure 8-1 – Functional architecture of the open USN service framework

The functional architecture of the open USN service framework consists of the open USN service platform and heterogeneous USN middleware. The open USN service platform consists of seven functional entities (FEs): Application support FE, LOD linking FE, Semantic inference FE, Resource management FE, Semantic USN repository FE, Semantic query FE and Adaptation FE.

The heterogeneous USN middleware entities are integrated into the open USN service platform through the Adaptation FEs; furthermore, the metadata of USN resources and semantic data are shared with the other services through LOD linking FE.

8.2 Functional entities

8.2.1 Application support FE

The Application support FE provides the functions which enable USN applications to obtain open USN services and/or the sensed data/semantic data from the open USN service platform.

The Application support FE also supports the functions that allow the establishment or maintenance of connections or disconnections according to the type of data request, and access control to handle access rights for user authentication and the use of services.

8.2.2 LOD linking FE

The LOD linking FE provides the functions that enable users to access the metadata of USN resources and semantic data in the open USN service platform via the web. It allows linking external LOD with the metadata of USN resources and semantic data in the open USN service platform.

The LOD linking FE also supports the interface for querying the metadata of USN resources and semantic data from the LOD, and the functions which allow the application and management of policies that include criteria about selection and publication of data for the LOD.

8.2.3 Semantic inference FE (optional)

The Semantic inference FE provides the inference functions based on the information described in the ontology schema and user rules by using the data stored in the Semantic USN repository FE.

Through the inference functions, the original sensed data in the sensed data repository are processed into semantic data, such as context data, and stored in the semantic data repository. The semantic data repository is updated with the inferred data either periodically or on-demand. Furthermore, it provides the functions to compose different kinds of patterns and levels for inference.

8.2.4 Resource management FE

The Resource management FE provides the functions that issue and manage the URIs of USN resources and semantic data. It also provides the functions that manage the mapping relations with the address of the USN resource. Further, the Resource management FE supports the functions that enable USN resources to be automatically registered in the open USN service platform when a USN resource is connected to a network such as the Internet, and enables applications to obtain and utilize information about USN resources.

The Resource management FE provides the functions that enable USN resources to actively register their status and connection information. By using this information, the open USN service platform will support network connection and mobility of USN resources.

Therefore, the Resource management FE can support plug and play functions which enable the open USN service platform to dynamically use USN resources which can automatically connect to the open USN service platform and register their own status and property information.

The Resource management FE provides the functions needed to search URIs of USN resources for performing queries that can provide necessary information for requests from applications.

In some cases, the Resource management FE can provide the functions necessary to configure and manage a logical group on USN resources for satisfying application service requests.

The Resource management FE may perform the functions to create a resource group according to application service requests and to manage lists of USN resources that belong to the resource group. Also, it supports the functions needed to create, maintain and manage information such as the resource group purpose, makers, control with rights, etc. It provides the functions necessary to manage the lifecycle of each resource group according to the duration of service.

8.2.5 Semantic USN repository FE

The Semantic USN repository FE includes the functions for converting metadata of USN resources and sensed data into RDF form. The Semantic USN repository FE includes two different repositories: Semantic data repository and Sensed data repository.

The Semantic USN repository FE stores the metadata of USN resources and semantic data in the Semantic data repository in RDF form. Also, the Semantic USN repository FE stores sensed data collected from USN middleware in the Sensed data repository.

It also provides query functions for searching, modifying and deleting stored data, as well as for inserting new data.

8.2.6 Semantic query FE

The Semantic query FE performs the functions that handle queries to USN middleware, Semantic USN repository FE and Resource management FE for providing responses to application information requests. It consists of a query analyser function, middleware query function, SPARQL query function and URI request query function.

The query analyser function creates queries by analysing the intentions of requests made by the applications, translates the results of each query process according to application message specifications, and delivers the translated data to the applications through the Application support FE. It classifies the requests from the applications into a query to the USN middleware, a query to the Semantic USN repository FE and a query to the Resource management FE. The query to the USN middleware, which requests the sensed data to the USN resources through the USN middleware, is created from metadata of USN resources according to types and attributes of operations supported. The query to the Semantic USN repository FE, which requests the metadata of USN resources and semantic data to the Semantic USN repository FE, is created by translating the queries that the applications request into SPARQL. The query to the Resource management FE, which requests the URIs of corresponding USN resources to the Resource management FE, is created for performing queries to the USN middleware or the Semantic USN repository FE that can provide the necessary information for satisfying requests from applications.

The middleware query function performs the functions to send queries to the USN middleware, and to collect the resulting data from the USN middleware through the Adaptation FE: it also manages the query status of each query to the USN middleware created from the query analyser function. The data, received temporarily or periodically from the USN middleware, are stored in the Semantic USN repository FE by the Adaptation FE. However, in some cases, such as for a real-time sensed data request, the sensed data can be directly sent to the query analyser function.

The SPARQL query function performs the functions to simultaneously handle many SPARQL queries created by the query analyser function, to produce the outcome of each query from the Semantic USN repository FE and to deliver these to the query analyser function.

The URI request query function performs the functions to send URI request queries to the Resource management FE, receive the URIs of corresponding USN resources from the Resource management FE and deliver them to the query analyser function.

8.2.7 Adaptation FE

Adaptation FE provides the functions to handle the protocols and messages for setting up connections with USN middleware and delivering queries and commands. It works as an interface between the open USN service platform and heterogeneous USN middleware for processing the corresponding protocols and messages for the respective USN middleware.

It supports the message translation function to translate the data from/to heterogeneous USN middleware according to proper message specifications to deal with in the open USN service platform and the respective USN middleware. It also provides the message routing function to deliver the translated data to corresponding FEs (Semantic USN repository FE, Resource management FE and Semantic query FE) of the open USN service platform in order to process requests.

Appendix I

Information flow in the open USN service framework

(This appendix does not form an integral part of this Recommendation.)

This appendix describes the information flow related to the operation of the open USN service framework that includes USN resource registration, sensed data/semantic data access from Semantic USN repository FE and USN resources.

I.1 USN resource registration

Figure I.1 shows the information flow describing how to register a USN resource into the open USN service platform.

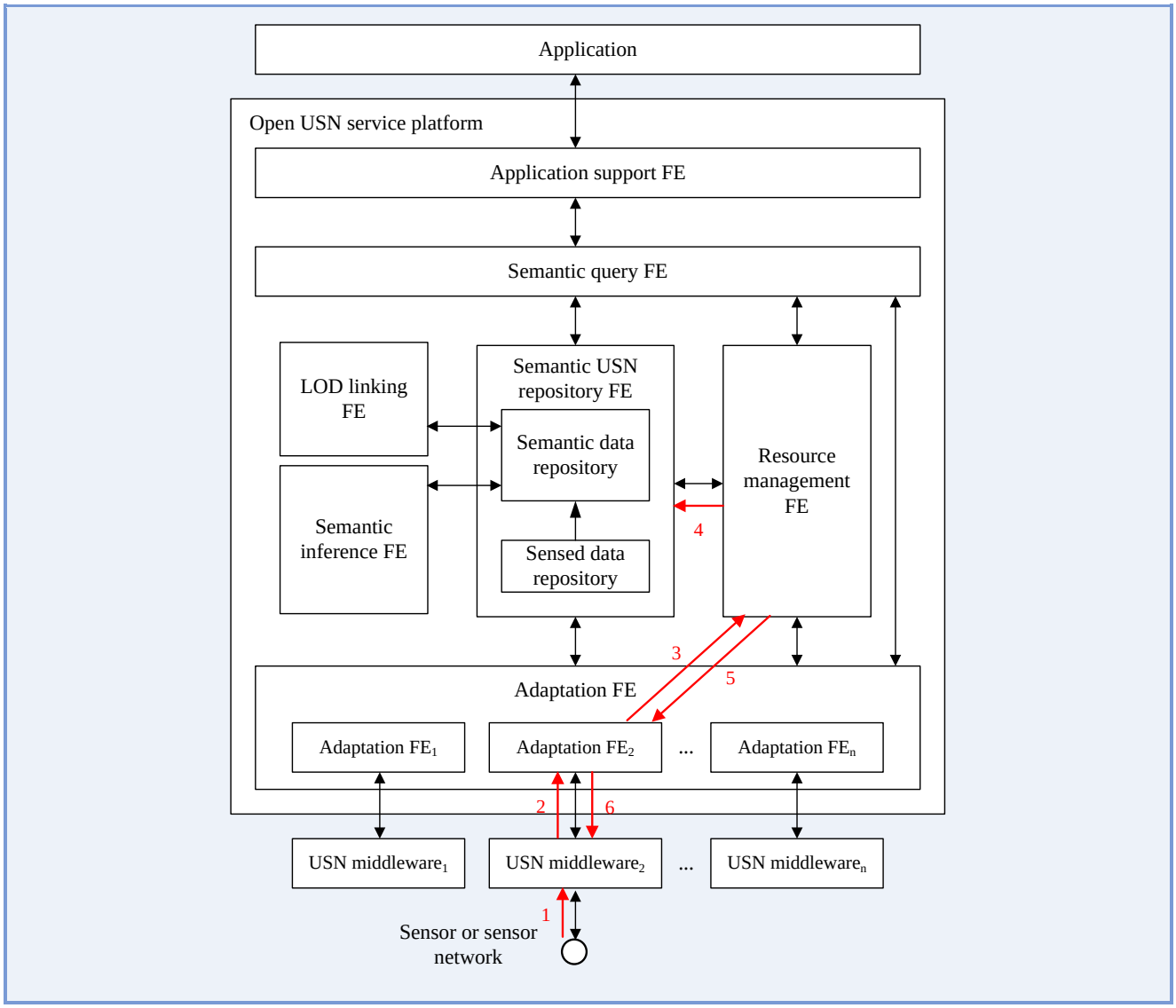


Figure I.1 – Information flow of USN resource registration

(1) The USN resource requests the USN middleware to register its information to the open USN service platform.

(2) The USN middleware sends a request to the open USN service platform through the Adaptation FE.

(3) The Adaptation FE sends the request message to the Resource management FE.

(4) The Resource management FE issues the URI of the USN resource and requests the Semantic USN repository FE to store the URI and metadata of the USN resource.

(5) The Resource management FE returns the result of the registration to the Adaptation FE.

(6) The Adaptation FE returns the result to the USN middleware.

I.2 Sensed data/semantic data access from Semantic USN repository FE

Figure I.2 shows the information flows describing how to access sensed data/semantic data stored in the Semantic USN repository FE.

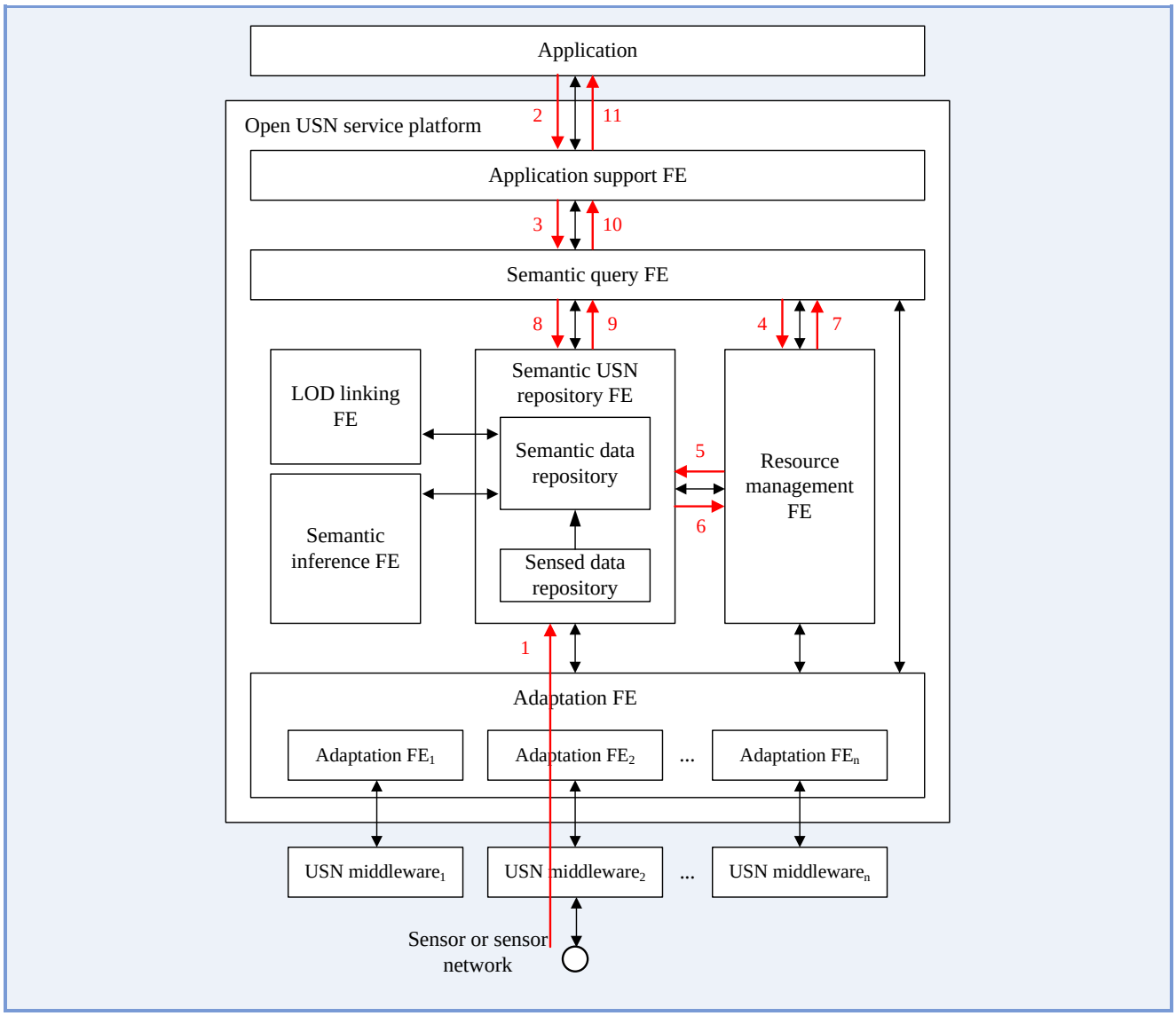


Figure I.2 – Information flow of accessing sensed data/ semantic data from Semantic USN repository FE

(1) The sensed data via the USN middleware from USN resources are periodically or continuously stored in the Semantic USN repository FE. The Semantic inference FE converts sensed data into semantic data periodically or on-demand.

(2) The Application requests the sensed data/semantic data from the open USN service platform through the Application support FE.

(3) The Application support FE sends a request message to the Semantic query FE.

(4) The Semantic query FE requests from the Resource management FE the URIs of USN resources that can be used for performing queries that will provide the necessary information to satisfy the Application request.

(5) The Resource management FE sends this request message to the Semantic USN repository FE.

(6) The Resource management FE receives the URIs of corresponding USN resources from the Semantic USN repository FE.

(7) The Resource management FE returns the URIs of corresponding USN resources to the Semantic query FE.

(8) The Semantic query FE queries the Semantic USN repository FE for the sensed data/semantic data related to the returned URIs.

(9) The Semantic query FE receives the sensed data/semantic data related to the returned URIs in the Semantic USN repository FE.

(10) The Semantic query FE sends the sensed data/semantic data to the Application support FE.

(11) The Application support FE sends the sensed data/semantic data to the Application.

I.3 Sensed data access from USN resources

Figure I.3 shows the information flows describing how to access sensed data directly from USN resources in real-time.

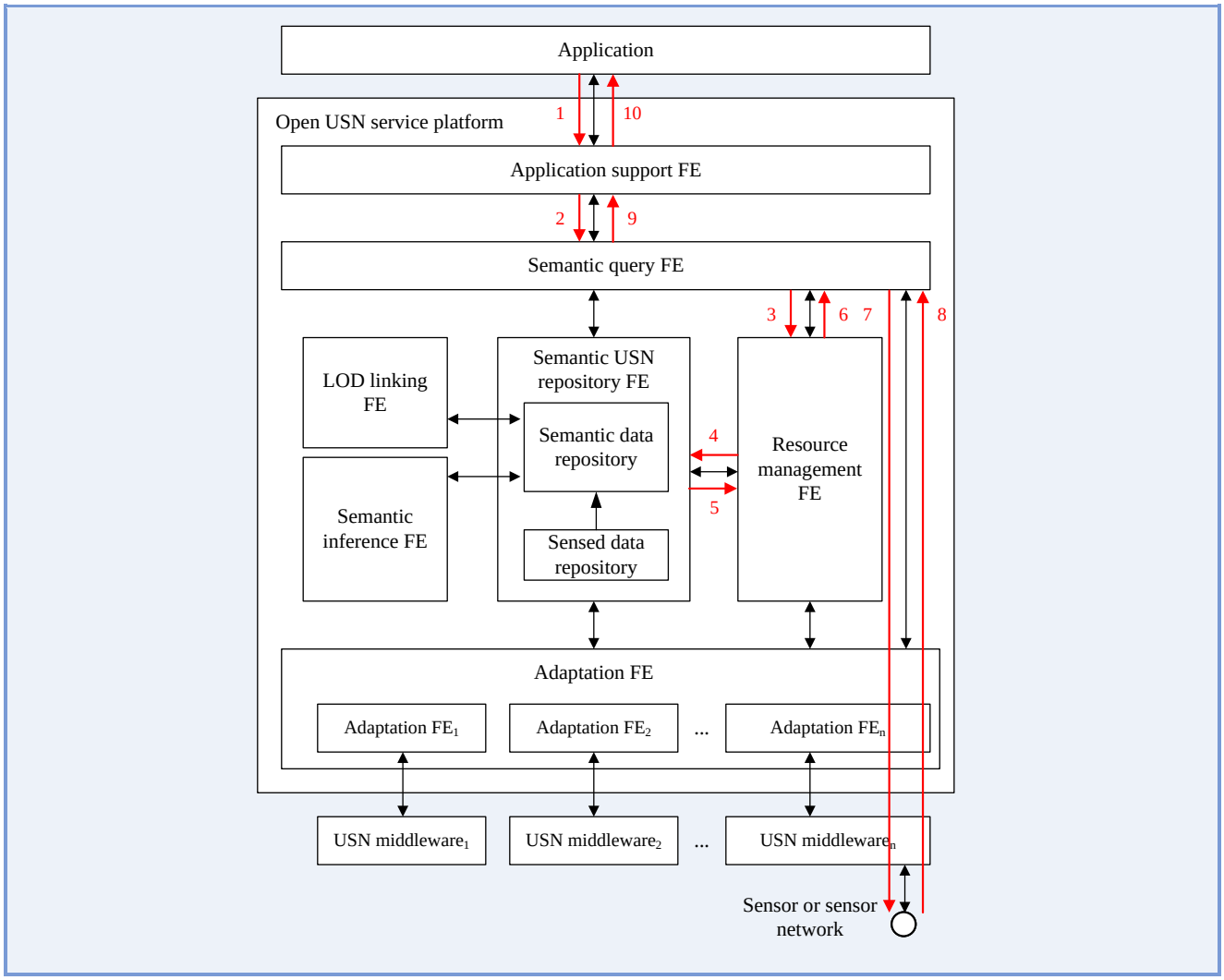


Figure I.3 – Information flow of accessing sensed data from USN resources

(1) The Application requests the sensed data from the open USN service platform through the Application support FE.

(2) The Application support FE sends a request message to the Semantic query FE.

(3) The Semantic query FE requests from the Resource management FE the URIs of USN resources that can be used for performing queries that will provide the necessary information to satisfy the request from the Application.

(4) The Resource management FE sends this request message to the Semantic USN repository FE.

(5) The Resource management FE receives the URIs of corresponding USN resources from the Semantic USN repository FE.

(6) The Resource management FE returns the URIs of corresponding USN resources to the Semantic query FE.

(7) The Semantic query FE queries via the USN middleware the sensed data of the corresponding USN resources using the returned URIs.

(8) The Semantic query FE receives the sensed data from corresponding USN resources via the USN middleware.

(9) The Semantic query FE sends the sensed data to the Application support FE.

(10) The Application support FE sends the sensed data to the Application.

Appendix II

Use cases of the open USN service platform

(This appendix does not form an integral part of this Recommendation.)

This appendix describes a use case for providing traffic information service using the open USN service platform.

II.1 Traffic information service using the open USN service platform

Most traffic information services provide information related to the driving of vehicles, such as traffic reports, road conditions and route guides (navigation) based on distance data. If various data related to the driving of vehicles are obtained in real-time, more valuable traffic information services could be provided. For example, recommended routes could be served more usefully by utilizing various data in real-time such as vehicle speed and condition data, weather data, traffic light data, personal schedule, and so on. Figure II.1 shows a use case that illustrates providing recommended routes in the traffic information service.

For this service, the open USN service platform could collect the following data:

– vehicle speed and condition (e.g., parts, tires, fuel) data from several sensors installed within the vehicle,

– weather data from weather sensors installed nationwide,

– traffic light data from sensor nodes that collect the state of traffic lights,

– personal scheduling data stored in a smart phone that can be used as a sensor node or gateway which delivers data in a smart phone to destination.

These data can be collected by the methods used to handle the respective messages and protocols through heterogeneous USN middleware, such as the open geospatial consortium (OGC) sensor web enablement (SWE) [b-SWE] or USN middleware described in [ITU-T F.744]. In the open USN service platform, the data collected from heterogeneous USN middleware are translated into proper message specifications by the message translation function of the Adaptation FE. The translated data are delivered to the corresponding FEs (Semantic USN repository FE, Resource management FE and Semantic query FE) of the open USN service platform in order to process requests by the message routing function of the Adaptation FE. The above data are stored in the Semantic USN repository FE in RDF form.

Using the Semantic inference FE, data with semantics such as the following context data are created and provided to users:

– recommended routes with good weather conditions and without traffic congestion, based on weather and speed data collected from vehicles on route to the destination;

– recommended routes to reach a meeting place in time for appointments according to personal schedule and vehicle fuel-level data based on traffic light and speed data collected from vehicles on route to the destination

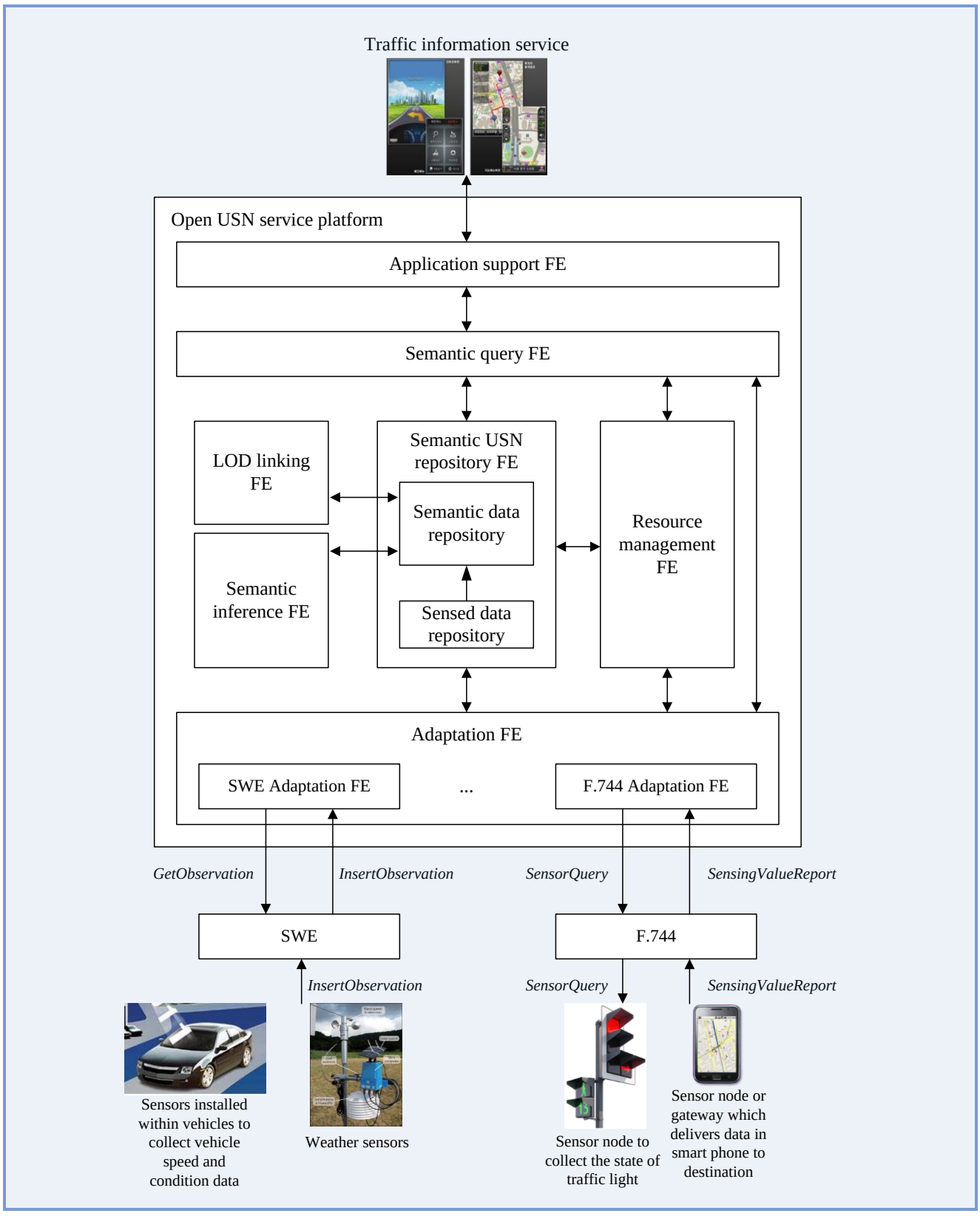


Figure II.1 – Traffic information service using the open USN service platform

Based on these data, the following scenario can be considered:

– The application requests information from the open USN service platform for finding the optimal route – with good weather conditions without traffic jams – to a destination.

– The open USN service platform collects relevant data from various sensors through various USN middleware including weather sensors and sensors installed within vehicles.

– The open USN service platform provides the information which is needed by the application to determine the optimal route to the destination.

– The application generates the recommended route to the destination based on the information provided from the open USN service platform.

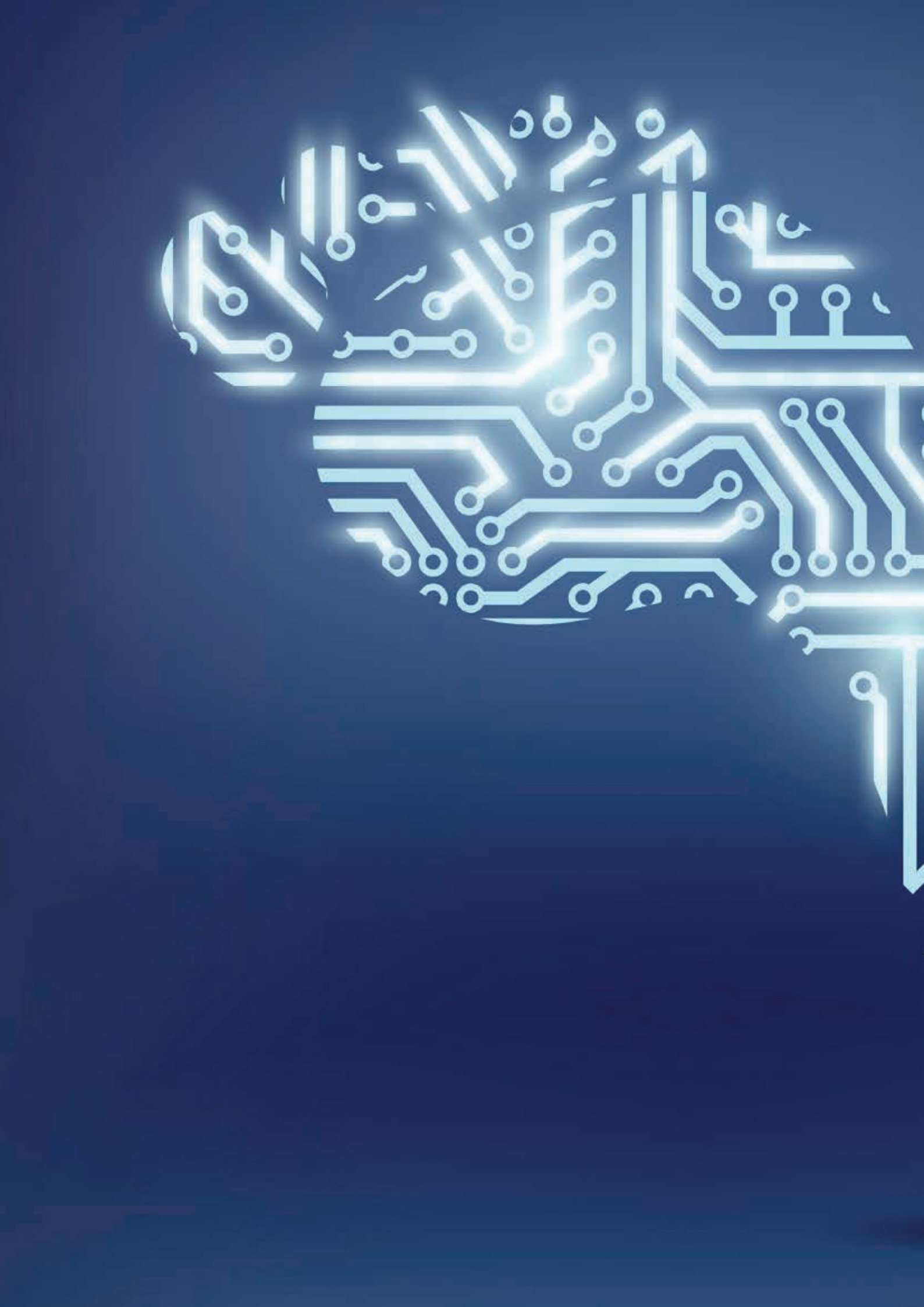
Bibliography

[b-LOD] W3C Working Group Note 27 June 2013, Linked Data Glossary, *Linked Open Data (LOD)*, <http://www.w3.org/TR/ld-glossary/#linked-open-data>

[b-RDF] W3C RDF Working Group (2004-02-10), *Resource Description Framework (RDF)*, <http://www.w3.org/2001/sw/wiki/RDF>

[b-SPARQL] W3C Recommendation 15 January 2008, *SPARQL Query Language for RDF*, <http://www.w3.org/TR/rdf-sparql-query>

[b-SWE] Open Geospatial Consortium (OGC), Sensor Web Enablement Domain Working Group (DWG), *Sensor Web Enablement*, <http://www.opengeospatial.org/projects/groups/sensorwebdwg>



Y.4403/Y.2026

Functional requirements and architecture of the next generation network for support of ubiquitous sensor network applications and services

Functional requirements and architecture of the next generation network for support of ubiquitous sensor network applications and services

Summary

Recommendation ITU-T Y.2026 includes functional requirements and architecture of the next generation network (NGN) for the support of ubiquitous sensor network (USN) applications and services. This Recommendation is based on the capabilities defined in Recommendation ITU-T Y.2221.

History

Edition	Recommendation	Approval	Study Group
1.0	ITU-T Y.2026	2012-07-29	13

Keywords

Frameworks, functional architecture, sensor(s), sensor network(s), USN.

Table of Contents

Recommendation ITU-T Y.4403/Y.2026

Functional requirements and architecture of the next generation network for support of ubiquitous sensor network applications and services

1 Scope

This Recommendation, which is based on [ITU-T Y.2012], covers extended features of the next generation network (NGN) for the support of ubiquitous sensor network (USN) applications and services. This Recommendation describes functional requirements, a functional architecture and functional entities in order to support the NGN service requirements and capabilities defined in [ITU-T Y.2221].

This Recommendation covers:

- Functional requirements and functions to support the NGN capabilities defined in [ITU-T Y.2221]
- A functional architecture and entities of the NGN to support USN applications and services

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T Y.2012] Recommendation ITU-T Y.2012 (2010), *Functional requirements and architecture of next generation networks*.

[ITU-T Y.2221] Recommendation ITU-T Y.2221 (2010), *Requirements for support of ubiquitous sensor network (USN) applications and services in the NGN environment*.

[ITU-T Y.2701] Recommendation ITU-T Y.2701 (2007), *Security requirements for NGN release 1*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 sensor [ITU-T Y.2221]: An electronic device that senses a physical condition or chemical compound and delivers an electronic signal proportional to the observed characteristic.

3.1.2 sensor network [ITU-T Y.2221]: A network comprised of interconnected sensor nodes exchanging sensed data by wired or wireless communication.

3.1.3 sensor node [ITU-T Y.2221]: A device consisting of sensor(s) and optional actuator(s) with capabilities of sensed data processing and networking.

3.1.4 ubiquitous sensor network (USN) [ITU-T Y.2221]: A conceptual network built over existing physical networks which makes use of sensed data and provides knowledge services to anyone, anywhere and at anytime, and where the information is generated by using context awareness.

3.1.5 USN end-user [ITU-T Y.2221]: An entity that uses the sensed data provided by USN applications and services. This end-user may be a system or a human.

3.1.6 USN gateway [ITU-T Y.2221]: A node which interconnects sensor networks with other networks.

3.2 Terms defined in this Recommendation

None.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

ABG-FE Access Border Gateway Functional Entity
AN-FE Access Node Functional Entity
ASF&SSF Application Support Functions and Service Support Functions
AS-FE Application Support Functional Entity
CAF Charging and Accounting Functions
EN-FE Edge Node Functional Entity
IP Internet Protocol
MLM-FE Mobile Location Management Functional Entity
MMCF Mobility Management and Control Functions
NACF Network Attachment Control Functions
NAC-FE Network Access Configuration Functional Entity
NGN Next Generation Network
OSE Open Service Environment
QoS Quality of Service
RACF Resource and Admission Control Functions
SAA-FE Service Authentication and Authorization Functional Entity
SC&CDF Service Control and Content Delivery Functions
SCF Service Control Functions
SCP-FE Service and Content Protection Functional Entity
SUP-FE Service User Profile Functional Entity
TAA-FE Transport Authentication and Authorization Functional Entity
TLM-FE Transport Location Management Functional Entity
TRC-FE Transport Resource Control Functional Entity
TUP-FE Transport User Profile Functional Entity
USN Ubiquitous Sensor Network

5 Conventions

None.

6 Functional requirements and functions of the NGN for USN applications and services

6.1 NGN functional requirements

[ITU-T Y.2221] describes the NGN service requirements and capabilities to support USN applications and services, and clause 8.1 of [ITU-T Y.2221] specifically states the following requirements for extensions or additions to NGN capabilities to support USN applications and services:

- Network management
- Profile management
  - Service profile

    NOTE 1 – Service profile requirement can be supported by existing NGN capabilities. See clause I.1.
  - Device profile
- Open service environment (OSE)
  - Service registration and discovery
  - Inter-working with service creation environments

    NOTE 2 – This Recommendation does not take these requirements into consideration. See clause I.1.
- Quality of service (QoS)
  - Application traffic control

    NOTE 3 – Application traffic control can be supported by existing NGN capabilities. See clause I.1.
- Privacy

  NOTE 4 – This Recommendation does not take this requirement into consideration. See clause I.1.

In addition to these capabilities, clause 8.2 of [ITU-T Y.2221] explains that the following capabilities are supported by the existing NGN capabilities to support USN applications and services:

- Open service environment
  - Service composition and coordination

    NOTE 5 – This Recommendation does not take this requirement into consideration. See clause I.1.
- Quality of service
  - Differentiated QoS and data prioritization
- Connectivity
- Location management
- Mobility
- Security
- Identification, authentication and authorization
- Accounting and charging

Appendix I describes requirements for the support of USN applications and services in NGN and allocates the corresponding functions of NGN to satisfy the requirements.

6.2 Functional architecture model

Figure 1 depicts the overall diagram of USN architecture, which uses NGN as a backbone network.

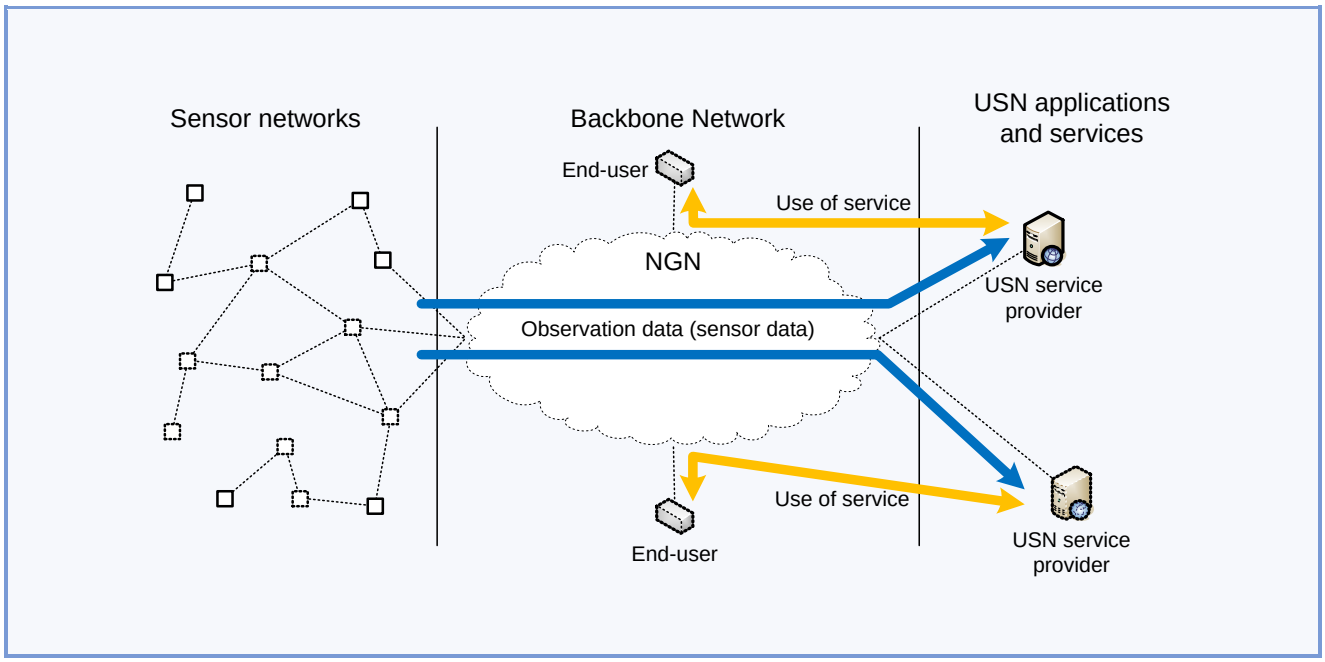


Figure 1 – Overall USN architecture

The sensor networks gather information about their physical surroundings and deliver this information to USN service providers through NGN. The USN service providers create USN services using this information and provide them to USN end-users via NGN. The USN end-users utilize USN services through NGN.

As a backbone network, NGN provides the capabilities for the support of USN applications and services in the transport stratum functions, the service stratum functions, the management functions and the end-user functions.

Figure 2 shows the overall functional architecture model of the NGN to support USN applications and services. More specifically, the figure shows the functions required for the support of USN applications and services. Appendix I describes requirements for the support of USN applications and services in NGN and allocates the corresponding functions of NGN to satisfy the requirements.

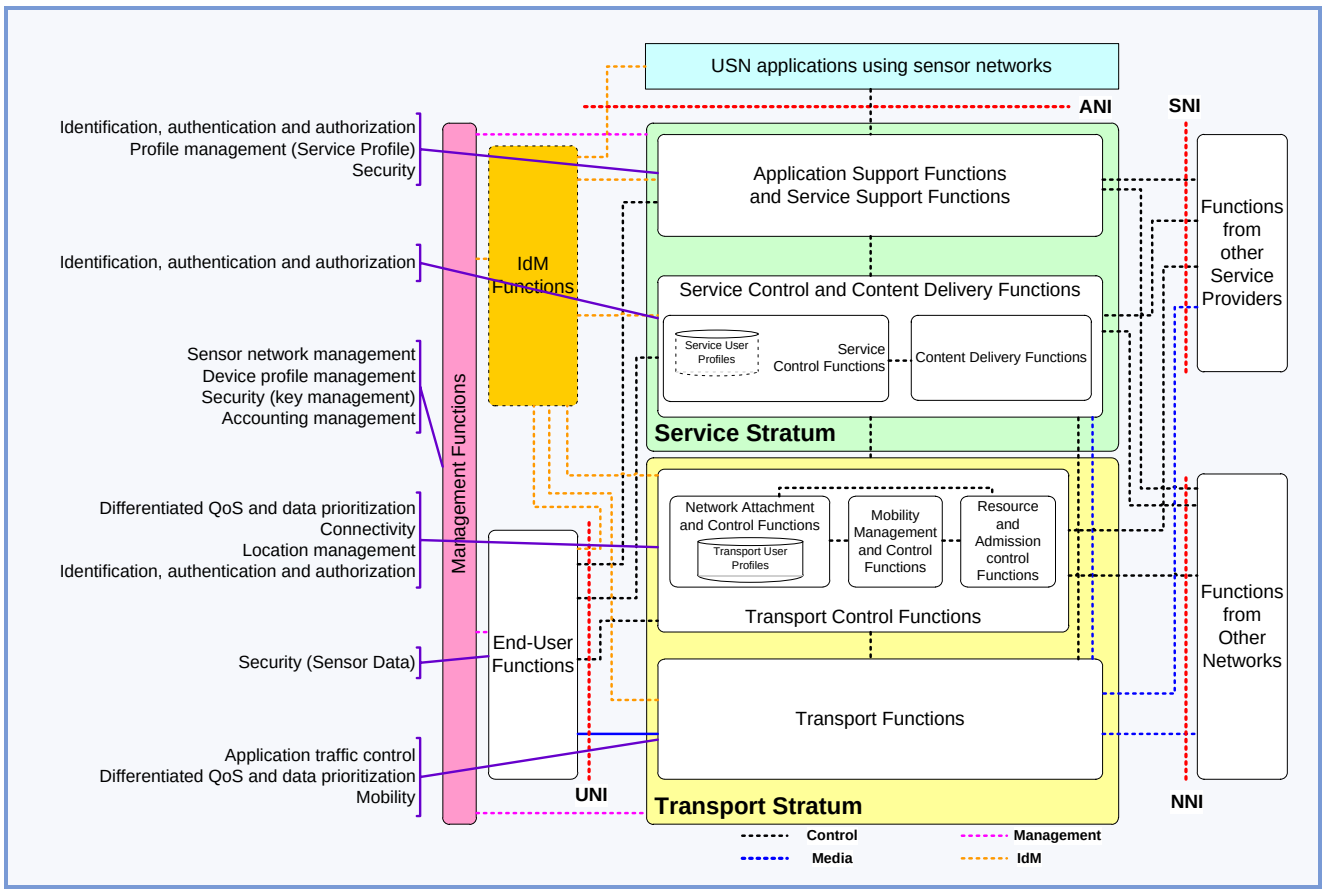


Figure 2 – Overall functional architecture model

As seen in Figure 2, there are no additional functions beyond [ITU-T Y.2012] to support requirements for USN applications and services. However, some NGN functions should be extended to support requirements in terms of functionalities. Clause 6.3 describes NGN functions to support USN applications and services.

6.3 Functions to support USN applications and services

This clause describes how the functional requirements identified in clause 6.1 are supported in the functional architecture model shown in clause 6.2. This clause focuses on the NGN capabilities which need the extensions of NGN functions to support USN applications and services.

The extension of NGN functions means that the NGN functions defined in [ITU-T Y.2012] should support additional capabilities for the requirements of USN applications and services. The extension of NGN functions can be achieved by implementation of additional capabilities of each stratum function.

6.3.1 Transport stratum functions

The transport stratum functions are required to be extended to support requirements of USN applications and services. The extensions of the transport stratum functions are given in the following clauses.

6.3.1.1 Transport functions

The following functional requirements are supported by the existing capabilities of the transport functions [ITU-T Y.2012]:

- Application traffic control requirement:

  The access network functions, edge functions and core transport functions are used for satisfying an application traffic control requirement which requests the management of transaction volume generated by the USN end-users.

- Differentiated QoS and data prioritization requirement:

  USN mission-critical applications and services should be carefully managed. For example, emergency notification of a fire case must be delivered by a time-critical and reliable way to appropriate national disaster monitoring systems. The access network functions, edge functions and core transport functions provide differentiated QoS and data prioritization capabilities.

6.3.1.2 Transport control functions

There are no extended capabilities in the transport control functions to support USN applications and services. The following functional requirements are supported by the transport control functions [ITU-T Y.2012]:

- Differentiated QoS and data prioritization requirement:

  As described in clause 6.3.1.1, USN applications and services require differentiated QoS and data prioritization. The resource and admission control functions (RACF) provides this capabilities in cooperation with the transport functions.

- Connectivity requirement:

  Both IP-based and non IP-based sensor networks can connect to NGN. When a non IP-based sensor network connects to NGN, an IP-capable gateway is used. The network attachment control function (NACF) provides connectivity to IP-based and non-IP based sensor networks through gateways.

- Location management requirement:

  The NACF provides a location management capability at the IP layer. If a sensor network uses IP addresses directly or a USN gateway provides IP-based connectivity, the location information management of sensor networks at the IP layer is supported by the NACF.

- Mobility requirement:

  The access network functions provide a mobility capability of a sensor network as well as mobility of a sensor node within a sensor network or across sensor networks. If a sensor network is based on IP technologies, the mobility management and control functions (MMCF) provide functions for the support of IP-based mobility of a sensor network as well as a sensor node.

- Identification, authentication and authorization:

  The NACF provides authentication of the USN end-users and authorization of network access.

6.3.2 Service stratum functions

The service stratum functions are required to be extended to support requirements of USN applications and services. The extensions of the service stratum functions are given in the following clauses.

6.3.2.1 Service control and content delivery functions

The following functional requirements are supported by existing service control and content delivery functions [ITU-T Y.2012]:

- Identification, authentication and authorization requirement:

  The service control functions (SCF) support authentication and authorization functions for the USN end-users at the service level.

6.3.2.2 Application support functions and service support functions

There are no extended capabilities in the application support functions and service support functions to support USN applications and services.

The following functional requirements are supported by existing application support functions and service support functions [ITU-T Y.2012]:

- Identification, authentication and authorization requirement:

  The application support functions and service support functions (ASF&SSF) provide authentication and authorization to access the services at the application level.

- Profile management requirement for registration and discovery of service:

  [ITU-T Y.2221] specifies that the service profile for service registration and discovery requires the extension of NGN capabilities. However, [ITU-T Y.2012] defines that the ASF&SSF, especially the application support functional entity (AS-FE), provides generic application server functions such as service selection and service discovery.

- Security:

  Application support functions and service support functions provide the protection of content (sensor data).

6.3.3 Management functions

The management functions are required to be extended to support requirements of USN applications and services. The extensions of the management functions are as follows:

- Sensor network management requirement:

  IP-based and non-IP-based sensor networks using various types of wired and/or wireless connection can coexist in USN applications and services. Non-IP-based sensor networks are often managed through their gateway. IP-based sensor networks include the case of a single sensor node directly connected to NGN, although sensor networks are often managed as a set. The management functions are required to manage IP-based sensor networks as well as non IP-sensor networks.

- Profile management (device profile) requirement:

  In USN applications and services, a device profile consisting of the information of sensor networks and/or sensor nodes should be provided and managed. As there are various types of sensors, sensor nodes and sensor networks, device profiles would help to manage a large number of heterogeneous nodes and networks. The information of device profiles may include sensor network identifier, device identifier, device types, capabilities and location. The management functions may support device profile management.

However, the following functional requirements are supported by existing management functions [ITU-T Y.2012]:

- Security requirement:

  Security management of USN applications and services including key management is supported by the management functions.

- Accounting and charging requirement:

  Different accounting and charging requirements might have to be addressed depending on the scenarios of USN applications and services. The management functions are required to support different accounting and charging policies according to different data transaction types of USN applications and services.

6.3.4 End-user functions

Existing end-user functions are required to support the following functions:

- Security requirement:

  Sensor data security within a sensor network is provided by the end-user functions.

7 Functional architecture of the NGN for USN applications and services

The extensions and additions of the functional entities defined in [ITU-T Y.2012] are required for the support of USN applications and services. Based on the functions described in clause 6.3, this clause identifies the extensions and additions of the functional entities of the NGN functional architecture illustrated in Figures 9-2, 9-3 and 9-4 of [ITU-T Y.2012].

7.1 Transport processing functional entities

7.1.1 T-2: Access node functional entity (AN-FE)

- Application traffic control requirement:

  The AN-FE supports an application traffic control requirement which requests the management of transaction volume generated by the USN end-users.

- Differentiated QoS and data prioritization requirement:

  The AN-FE supports different QoS and also supports data prioritization capabilities.

7.1.2 T-3: Edge node functional entity (EN-FE)

- Application traffic control requirement:

  The EN-FE supports an application traffic control requirement which requests the management of transaction volume generated by the USN end-users.

- Differentiated QoS and data prioritization requirement:

  The EN-FE supports different QoS and also supports data prioritization capabilities.

7.1.3 T-5: Access border gateway functional entity (ABG-FE)

- Differentiated QoS and data prioritization requirement:

  The ABG-FE supports different QoS and also supports data prioritization capabilities.

7.2 Transport control functional entities

7.2.1 T-10: Network access configuration functional entity (NAC-FE)

- Connectivity requirement:

  The NAC-FE provides connectivity to IP-based and non-IP based sensor networks through gateways.

7.2.2 T-11: Transport authentication and authorization functional entity (TAA-FE)

- Identification, authentication and authorization:

  The TAA-FE provides authentication of the USN end-users and authorization of network access.

7.2.3 T-12: Transport user profile functional entity (TUP-FE)

- Identification, authentication and authorization:

  The TAA-FE provides authentication of the USN end-users and authorization of network access.

7.2.4 T-13: Transport location management functional entity (TLM-FE)

- Location management requirement:

  The TLM-FE provides location management capability at the IP layer. If a sensor network uses IP addresses directly or a USN gateway provides IP-based connectivity, the location information management of sensor networks at the IP layer is supported by the TLM-FE.

7.2.5 T-17: Transport resource control functional entity (TRC-FE)

- Differentiated QoS and data prioritization requirement:

  The TRC-FE provides different QoS and also supports data prioritization capabilities.

7.2.6 T-18: Mobile location management functional entity (MLM-FE)

- Mobility requirement:

  The MLM-FE provides functions for the support of IP-based mobility of a sensor network as well as a sensor node.

7.3 Service control functional entities

7.3.1 S-5: Service user profile functional entity (SUP-FE)

- Identification, authentication and authorization:

  The SUP-FE is responsible for storing user profiles, subscriber-related location data and presence status data in the service stratum. A user profile is required to be provided in support of authentication, authorization and so on.

7.3.2 S-6: Service authentication and authorization functional entity (SAA-FE)

- Identification, authentication and authorization:

  The SAA-FE provides authentication and authorization in the service stratum.

7.4 Application support functions and service support functions

7.4.1 A-1: Application support functional entity (AS-FE)

- Profile management requirement for service registration and discovery:

  The AS-FE supports generic application server functions including hosting and executing services such as service selection servers and service discovery servers.

7.4.2 A-8: Service and content protection functional entity (SCP-FE)

- Identification, authentication and authorization:

  The SCP-FE provides authentication and authorization in the service stratum.

- Security:

  The SCP-FE provides the protection of the services and content (sensor data).

8 Security considerations

Security considerations regarding the functional requirements and architecture of the NGN are addressed in [ITU-T Y.2701].

Appendix I

Analysis of service requirements and network capabilities defined in Recommendation ITU-T Y.2221

(This appendix does not form an integral part of this Recommendation.)

I.1 Requirements for extensions to NGN capabilities

[ITU-T Y.2221] defines the following requirements to be supported by extended NGN capabilities.

Requirements	Explanation
Network management	• It is required to manage IP-based sensor networks including the case of a single node directly connected to NGN. • It is required to manage non-IP-based sensor networks. • It is required to support configuration and reconfiguration of sensor networks for assurance of connectivity and lifetime management.
Profile management for service registration and discovery	• It is recommended to use a standard set of USN service profiles to register and discover USN services. NOTE – [ITU-T Y.2221] specifies that the service profile for service registration and discovery requires the extension of NGN capabilities. However, [ITU-T Y.2012] defines that the ASF&SSF, especially the application support functional entity (AS-FE), provides generic application server functions such as service selection and service discovery. Therefore, a service profile requirement can be supported by existing NGN capabilities.
Profile management: Device profile	• It is optional to use device profiles containing sensor network related information.
Open service environment: Service registration and discovery	• It is required to support at least one USN service description language and its associated execution framework. • It is recommended to register and discover USN services based on a standard set of USN service profiles. • Registration and discovery of sensor network devices may be supported. • Context-awareness can be optionally supported in service discovery for USN applications and services. NOTE – [ITU-T Y.2221] specifies that service registration and discovery using the service profile under OSE require the extension of NGN capabilities. However, [ITU-T Y.2012] defines that the ASF&SSF, especially the application support functional entity (AS-FE), provides generic application server functions such as service selection and service discovery. Therefore, OSE service registration and discovery requirement can be supported by the existing NGN capability (i.e., ASF&SSF) instead of OSE.

Requirements	Explanation
Open service environment: Inter-working with service creation environments	• It is recommended to support inter-working with other service creation environments for the creation of USN applications and services NOTE – From the viewpoint of USN applications and services, NGN may be regarded as an intermediate network providing connectivity, data delivery and management. Service creation under OSE provides NGN service creation, not creation of USN applications and services. USN service creation is provided by other service providers and not the NGN. Therefore this Recommendation does not take this requirement into consideration.
Quality of service: Application traffic control	• It is required to manage the transaction volume generated by USN applications and services. • It is recommended to be able to avoid access concentration to a single resource. NOTE – [ITU-T Y.2221] specifies that QoS-application traffic control requires the extension of NGN capabilities. However, [ITU-T Y.2012] defines that access network functions and edge functions provide QoS and traffic control. Therefore QoS-application traffic control requirement can be supported by existing NGN capabilities.
Privacy	• There should be an option for privacy-enhanced multi-hop routing mechanisms (information on originating node identifier (ID), time and location should not be revealed – at least not totally – to intermediate nodes). • There should be an operating option to de-correlate sensor activity patterns (revealing sensitive context information) from the ensuing communication traffic patterns. NOTE – [ITU-T Y.2221] specifies that privacy protection should be supported by NGN. However, this requirement needs an end-to-end (from sensor node/sensor network to application) security mechanism. The main role of NGN for supporting USN applications and services is delivering sensor data to USN applications or delivering control data to sensor networks. From the USN point of view, NGN is an intermediate network and the components of NGN can also be regarded as intermediate nodes. Therefore privacy protection that requires an end-to-end security mechanism cannot be supported by NGN functions and this Recommendation does not take this requirement into consideration.

I.2 Requirements supported by existing NGN capabilities

[ITU-T Y.2221] defines the following requirements to be supported by existing NGN capabilities.

Requirements	Explanation
Open service environment: Service composition and coordination	• It is recommended to support service composition and coordination for creation of USN applications and services. NOTE – From the viewpoint of USN applications and services, NGN may be regarded as an intermediate network providing connectivity, data delivery and management. Service composition and coordination under OSE provides NGN service composition and coordination, not USN applications and services. USN service composition and coordination is provided by other service providers than NGN. Therefore this Recommendation does not take this requirement into consideration.
Quality of service: Differentiated QoS and data prioritization	• It is recommended to provide differentiated QoS and data prioritization according to the specific USN service requirements.
Connectivity	• It is required to support connectivity between sensor networks and infrastructure networks, regardless of the sensor network type, i.e., IP-based or non-IP-based and using various types of wired and/or wireless media connections. This includes the case in IP-based sensor networks of a single sensor node directly connected to the infrastructure networks.
Location management	• Location information of sensor networks is recommended to be registered for USN applications and services. Registration can be static or dynamic. • Location information of an individual sensor node can be optionally registered for USN applications and services when the location information of a single sensor node is useful. • Location information is recommended to be trustworthy, hence location discovery and management is recommended to be secure.
Mobility	• It is required to support network mobility when a sensor network moves across infrastructure networks. • Infrastructure networks are required to support intra-sensor network mobility and inter-sensor network mobility when location information of a moving sensor node is required to be traced.

Requirements	Explanation
Security	• It is required to support key management schemes for USN applications and services. • It is recommended to support scalable key management schemes for USN applications and services operating with large sensor networks. • It is recommended to provide security for the aggregated data when sensed data from two or more applications and services are integrated in infrastructure networks for the creation of new services. • The security approaches for the support of USN applications and services are recommended to be consistent with the general approach for security in NGN. • In addition to data security, the USN communication infrastructure is recommended to provide information transport security for protection against well-known passive and active attacks. Protocols for information transport are required to be resilient to attacks. • Depending on the specific USN application security requirements, a means for intrusion detection is required.
Identification, authentication and authorization	• It is required to support identification, authentication and authorization for users to access USN applications and services based on the security level of service data. • It is required to support different levels of authentication for different types of data based on the requirements of USN applications and services. • The USN end-users can optionally identify and authenticate network providers and USN service providers.
Accounting and charging	• It is required to support different accounting and charging policies according to different data transaction types of USN applications and services.

I.3 Mapping table of the requirements and the extended NGN functions

Clause I.1 defines the NGN capabilities to be extended to satisfy the requirements of [ITU-T Y.2221]. This clause allocates proper functions of NGN which are defined in [ITU-T Y.2012] to support the capabilities.

Requirements	Corresponding functions
Network management	• Management functions
Profile management: Device profile	• Management functions

I.4 Mapping table of the requirements and the existing NGN functions

Clause I.2 defines the existing NGN capabilities to satisfy the requirements of [ITU-T Y.2221]. This clause allocates proper functions of NGN which are defined in [ITU-T Y.2012] to support the capabilities.

Requirements	Corresponding functions
Profile management for service registration and discovery	• Application support functions and service support functions (ASF&SSF)
Quality of service: Application traffic control	• Transport functions – Access network functions – Edge functions • Core transport functions
Quality of service: Differentiated QoS and data prioritization	• Transport functions – Access network functions – Edge functions – Core transport functions • Transport control functions – Resource and admission control functions (RACF)
Connectivity	• Transport control functions – Network attachment control functions (NACF)
Location management	• Transport control functions – Network attachment control functions (NACF)
Mobility	• Transport functions – Access network functions – Mobility management and control functions (MMCF)
Security	• Management functions (security key management) • End-user functions (data security) • Application support functions and service support functions (ASF&SSF)
Identification, authentication and authorization	• Transport control functions – Network attachment control functions (NACF) • Service control and content delivery functions (SC&CDF) – Service control functions (SCF) • Application support functions and service support functions (ASF&SSF)
Accounting and charging	• Management functions (accounting management) – Charging and accounting functions (CAF)

Appendix II

USN middleware functions provided by NGN

(This appendix does not form an integral part of this Recommendation.)

[b-ITU-T F.744] describes USN middleware as follows:

- USN middleware is an intermediate entity which provides functions commonly required by different types of USN applications and services. USN middleware receives requests from USN applications and delivers those requests to appropriate sensor networks. Similarly, USN middleware receives sensed data or processed data from sensor networks and delivers them to appropriate USN applications. USN middleware can provide information processing functions such as query processing, context-aware processing, event processing, sensor network monitoring and so on.

NOTE – Detailed descriptions and requirements related to USN middleware are given in [b-ITU-T F.744].

From the description above, it is understood that USN middleware provides accessibility to sensor networks and data between sensor networks and USN applications and services. Also, USN middleware encompasses information processing functions which may be commonly required by various types of USN applications and services.

As described in Appendix I, NGN may be regarded as an intermediate network providing connectivity, data delivery and management. From these facts, USN middleware functions can be implemented by NGN functions.

Figure II.1 depicts the functional model of USN middleware given in [b-ITU-T F.744]. NGN functions may provide basic functions and advanced functions of USN middleware in each stratum of NGN.

However, this Recommendation does not intend to clarify NGN functional entities for providing USN middleware functions. This is out of scope of this Recommendation.

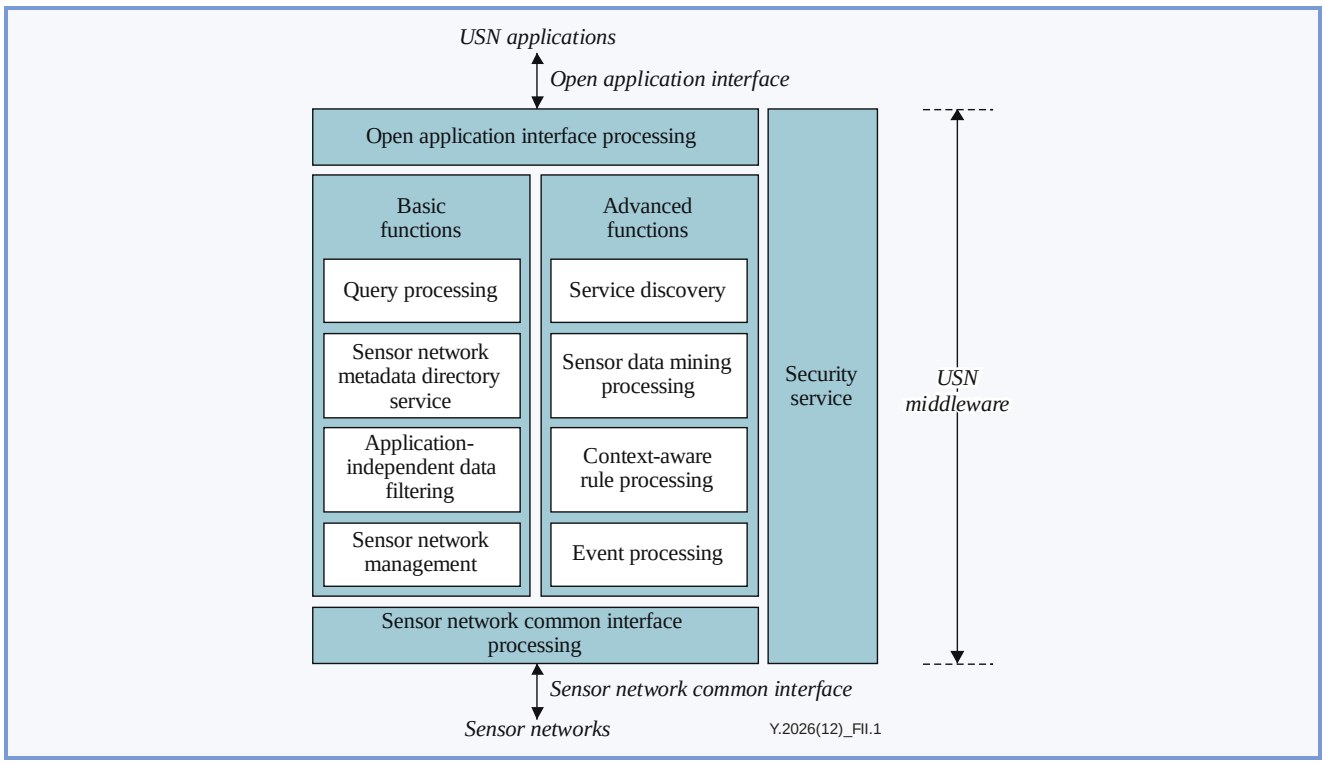
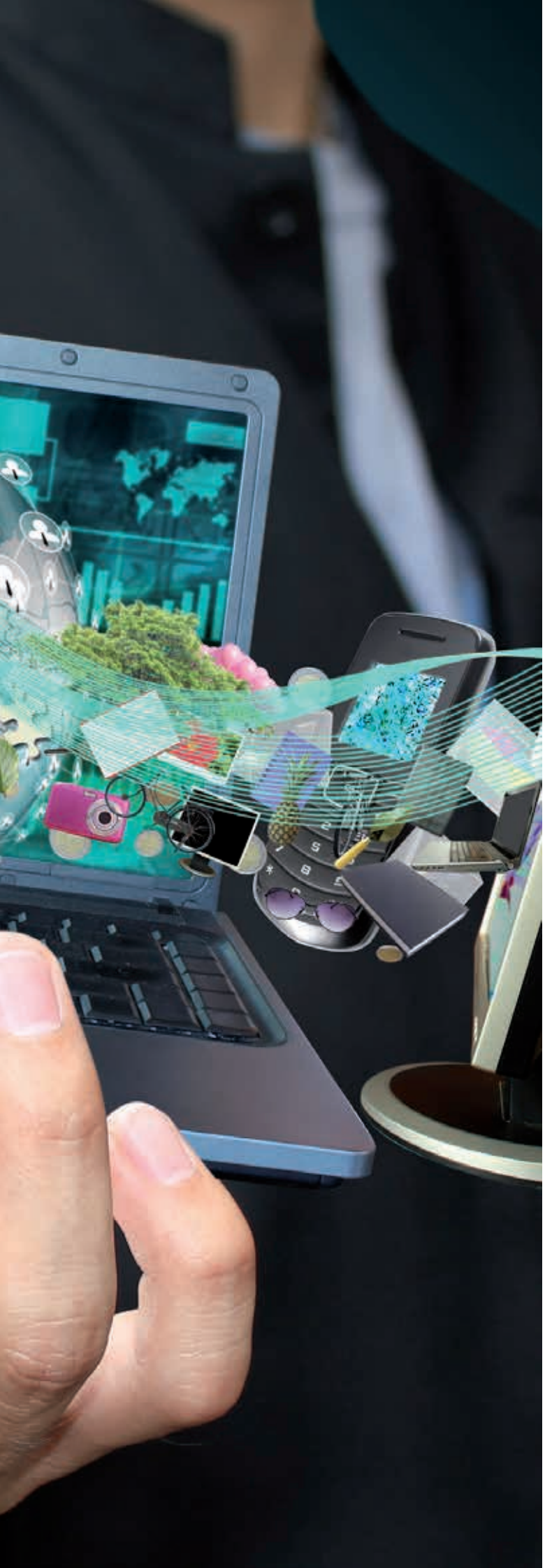


Figure II.1 – Functional model of USN middleware ([b-ITU-T F.744])

Bibliography

[b-ITU-T F.744] Recommendation ITU-T F.744 (2009), *Service description and requirements for ubiquitous sensor network middleware*.



Y.4404/Y.2062

Framework of object-to-object communication for ubiquitous networking in next generation networks

Framework of object-to-object communication for ubiquitous networking in next generation networks

Summary

Recommendation ITU-T Y.2062 describes the concept and high-level architectural model of object-to-object communication for ubiquitous networking in next generation networks (NGNs). It also presents requirements and a mechanism for identification of all objects and for providing connectivity to them.

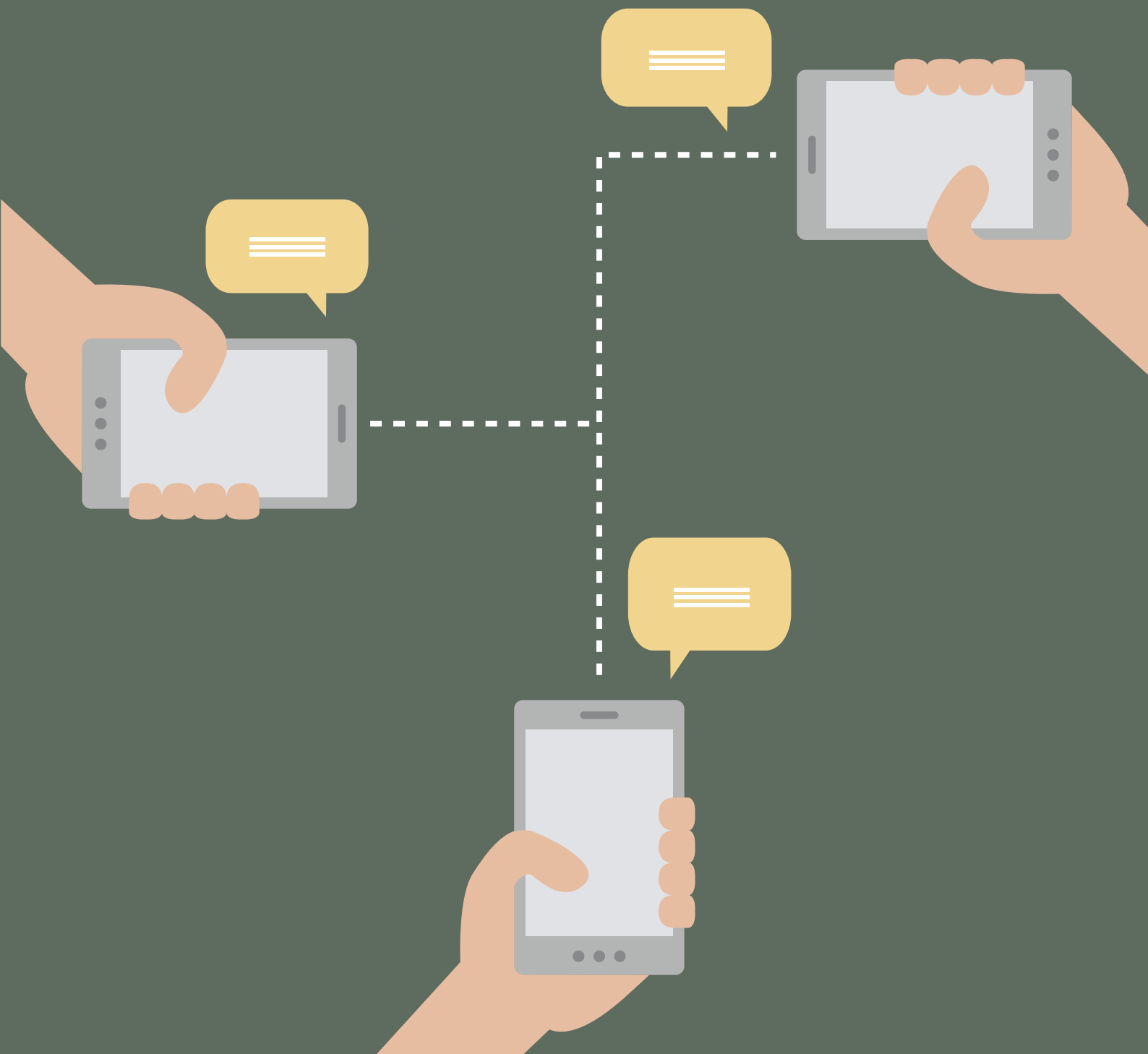
History

Edition	Recommendation	Approval	Study Group
1.0	ITU-T Y.2062	2012-03-29	13

Keywords

NGN, object, ubiquitous networking.

Table of Contents



Recommendation ITU-T Y.4404/Y.2062

Framework of object-to-object communication for ubiquitous networking in next generation networks

1 Scope

This Recommendation describes the concept and high-level architectural model of object-to-object communication for ubiquitous networking in next generation networks (NGNs). It also presents requirements and a mechanism for identification of all objects and for providing connectivity to them. This Recommendation covers the following items:

- General overview of ubiquitous networking in NGNs from the end-user perspective
- Basic concept and high-level architectural model for object-to-object communication using NGNs
- Requirements and technical considerations of object-to-object communication for ubiquitous networking
- A mechanism for object-to-object communication.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T Y.2001] Recommendation ITU-T Y.2001 (2004), *General overview of NGN.*

[ITU-T Y.2002] Recommendation ITU-T Y.2002 (2009), *Overview of ubiquitous networking and of its support in NGN.*

[ITU-T Y.2291] Recommendation ITU-T Y.2291 (2011), *Architectural overview of next generation home networks.*

[ITU-T Y.2701] Recommendation ITU-T Y.2701 (2007), *Security requirements for NGN release 1.*

[ITU-T Y.2702] Recommendation ITU-T Y.2702 (2008), *Authentication and authorization requirements for NGN release 1.*

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 context [ITU-T Y.2002]: The information that can be used to characterize the environment of a user.

NOTE – Context information may include where the user is, what resources (devices, access points, noise level, bandwidth, etc.) are near the user, at what time the user is moving, interaction history between person and objects, etc. According to specific applications, context information can be updated.

3.1.2 object [ITU-T Y.2002]: An intrinsic representation of an entity that is described at an appropriate level of abstraction in terms of its attributes and functions.

NOTE 1 – An object is characterized by its behaviour. An object is distinct from any other object. An object interacts with its environment including other objects at its interaction points. An object is informally said to perform functions and offer services (an object which makes a function available is said to offer a service). For modelling purposes, these functions and services are specified in terms of the behaviour of the object and of its interfaces. An object can perform more than one function. A function can be performed by the cooperation of several objects.

NOTE 2 – Objects include terminal devices (e.g., used by a person to access the network such as mobile phones, Personal computers, etc.), remote monitoring devices (e.g., cameras, sensors, etc.), information devices (e.g., content delivery server), products, contents, and resources.

3.1.3 ubiquitous networking [ITU-T Y.2002]: The ability for persons and/or devices to access services and communicate while minimizing technical restrictions regarding where, when and how these services are accessed, in the context of the service(s) subscribed to.

NOTE – Although technical restrictions to access services and communicate may be minimized, other constraints such as regulatory, national, provider and environmental constraints may impose further restrictions.

3.2 Term defined in this Recommendation

None.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

ANI	Application to Network Interface
API	Application Programming Interface
BT	Bio Technology
CT	Content Technology
ID	Identifier
IdM	Identity Management
IP	Internet Protocol
IT	Information Technology
ITS	Intelligent Transportation System
LAN	Local Area Network
LTE	Long Term Evolution
NGN	Next Generation Network
NT	Nano Technology
PC	Personal Computer
PDA	Personal Digital Assistant
QoE	Quality of Experience
QoS	Quality of Service
RFID	Radio Frequency Identifier
UNI	User to Network Interface

URI Uniform Resource Identifier

URL Uniform Resource Locator

WiMAX Worldwide Interoperability for Microwave Access

WLAN Wireless Local Area Network

xDSL Various types of Digital Subscriber Lines

5 Conventions

In this Recommendation:

The keywords "is required to" indicate a requirement which must be strictly followed and from which no deviation is permitted, if conformance to this Recommendation is to be claimed.

The keywords "is prohibited from" indicate a requirement which must be strictly followed and from which no deviation is permitted, if conformance to this Recommendation is to be claimed.

The keywords "is recommended" indicate a requirement which is recommended but which is not absolutely required. Thus, this requirement need not be present to claim conformance.

The keywords "is not recommended" indicate a requirement which is not recommended but which is not specifically prohibited. Thus, conformance with this Recommendation can still be claimed even if this requirement is present.

The keywords "can optionally" indicate an optional requirement which is permissible, without implying any sense of being recommended. This term is not intended to imply that the vendor's implementation must provide the option, and the feature can be optionally enabled by the network operator/service provider. Rather, it means the vendor may optionally provide the feature and still claim conformance with this Recommendation.

6 Ubiquitous networking in NGNs

6.1 Overview of ubiquitous networking

The term "ubiquitous networking" as defined in clause 3.1.3 is used for networking capabilities to support various classes of applications and services which require the "Any services, any time, any where and any objects" operation using NGN-enabled capabilities. This networking capability should support person-to-person, person-to-object (e.g., device and/or machine), and object-to-object communications.

For object-to-object communication, an object delivers information (e.g., sensor-related information) to another object, with or without the involvement of persons.

Figure 1 shows a general network configuration for ubiquitous networking. Objects around us are connected to the network and communicate through the establishment of end-to-end connectivity between them. Objects which are not moving are called fixed objects. Objects which move from one place to another are called mobile objects. Logical objects (e.g., contents in a server, resources, etc.) are considered as entities to be connected. These objects are connected to an NGN via wired or wireless interfaces in a fixed environment (e.g., home, building, etc.) or mobile environment (e.g., vehicle). In particular, some physical objects (i.e., fixed objects and/or mobile objects) are connected as logical objects through their virtual representation to be identifiable. A gateway can be used as an intermediate node between object(s) and a network. Depending on communication environment, a home network can reside in the position of gateway.

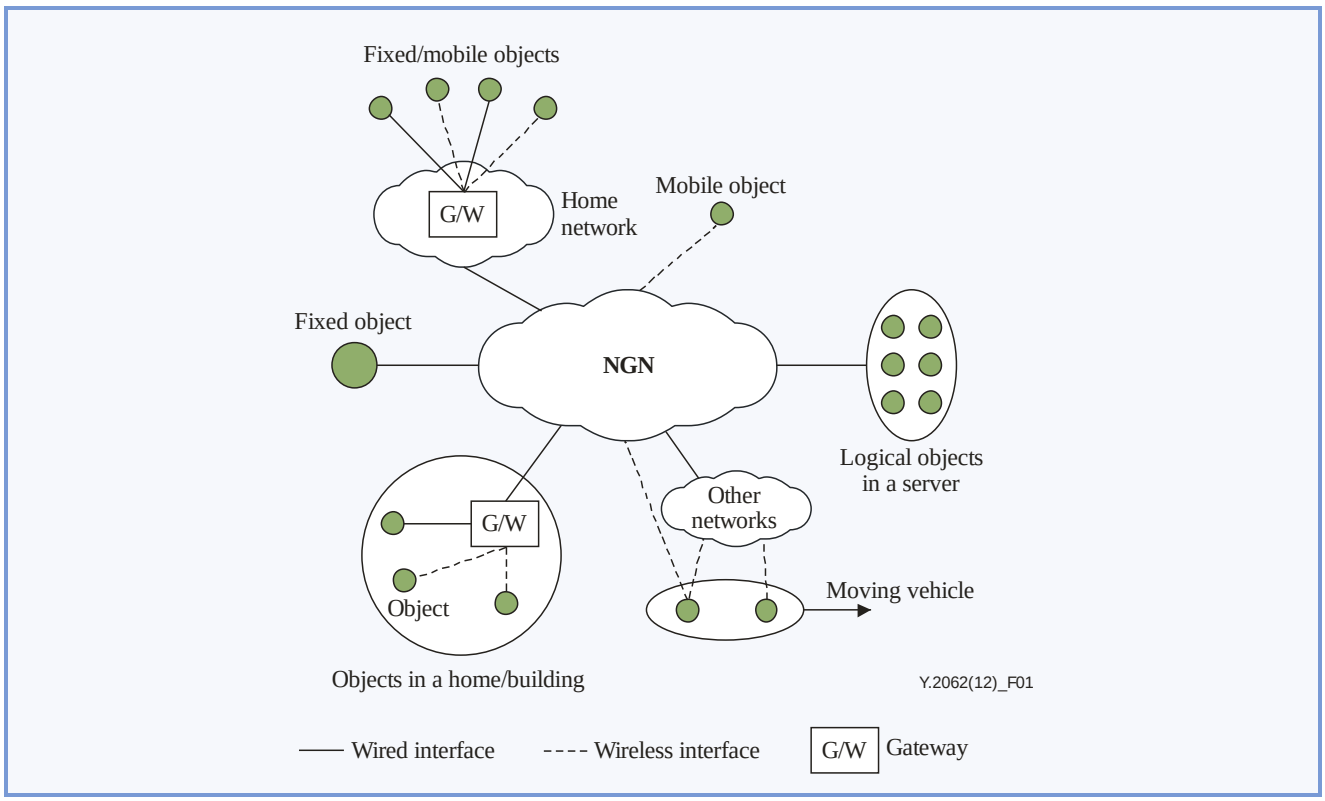


Figure 1 – General network configuration for ubiquitous networking

6.2 Architectural model for ubiquitous networking

From the architectural model for ubiquitous networking in Figure 2 of [ITU-T Y.2002] (reproduced here as Figure 2), enhanced capabilities for ubiquitous networking in NGNs include:

- Connecting-to-anything capabilities
- Open web-based service environment capabilities
- Context-awareness and seamless capabilities
- Multi-networking capabilities
- End-to-end connectivity over interconnected networks.

Among the specified capabilities, "connecting-to-anything" is tightly related to functionalities on the end-user side of NGNs. This Recommendation focuses mainly on object-to-object communication to support the connecting-to-anything capability for ubiquitous networking on the end-user side.

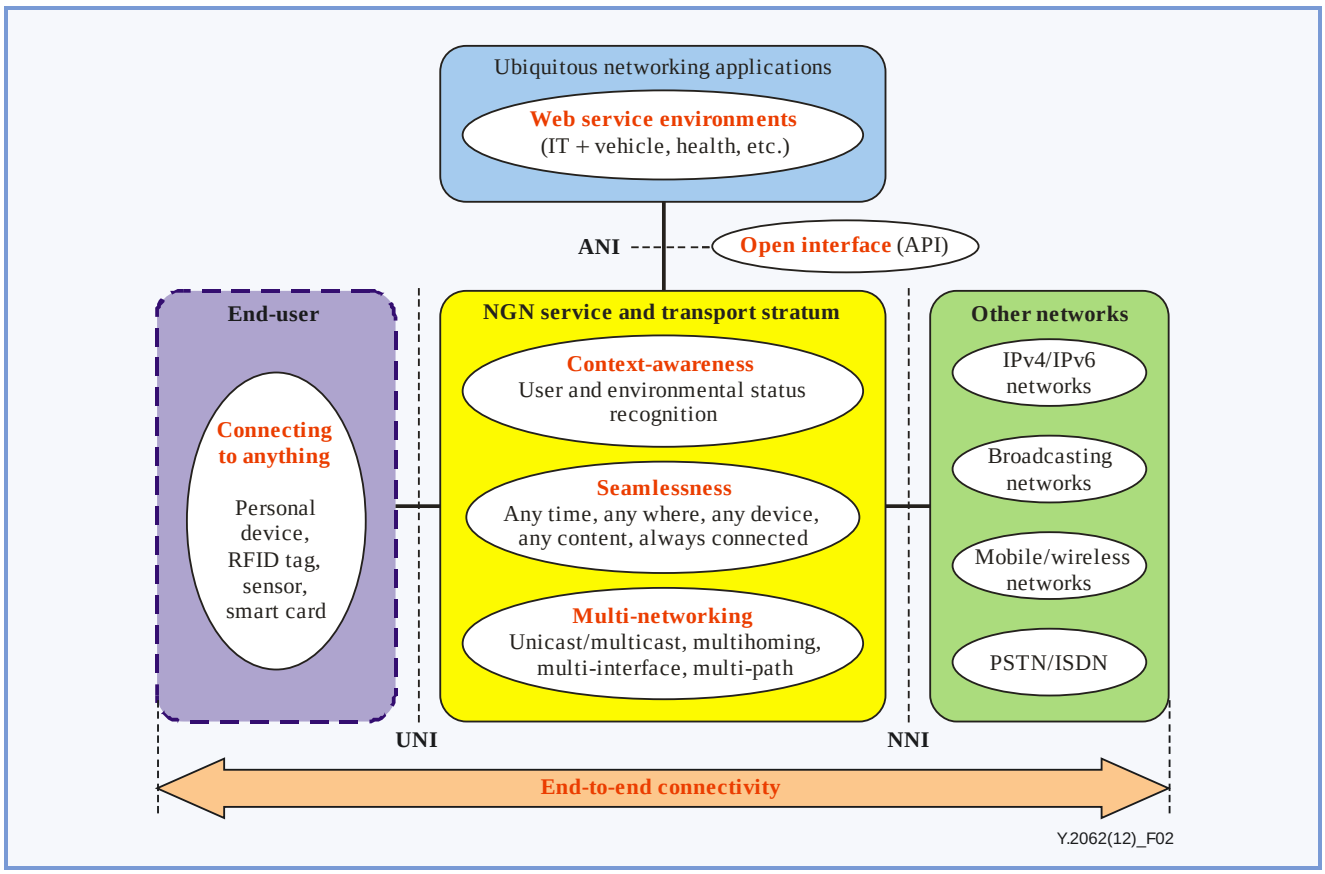
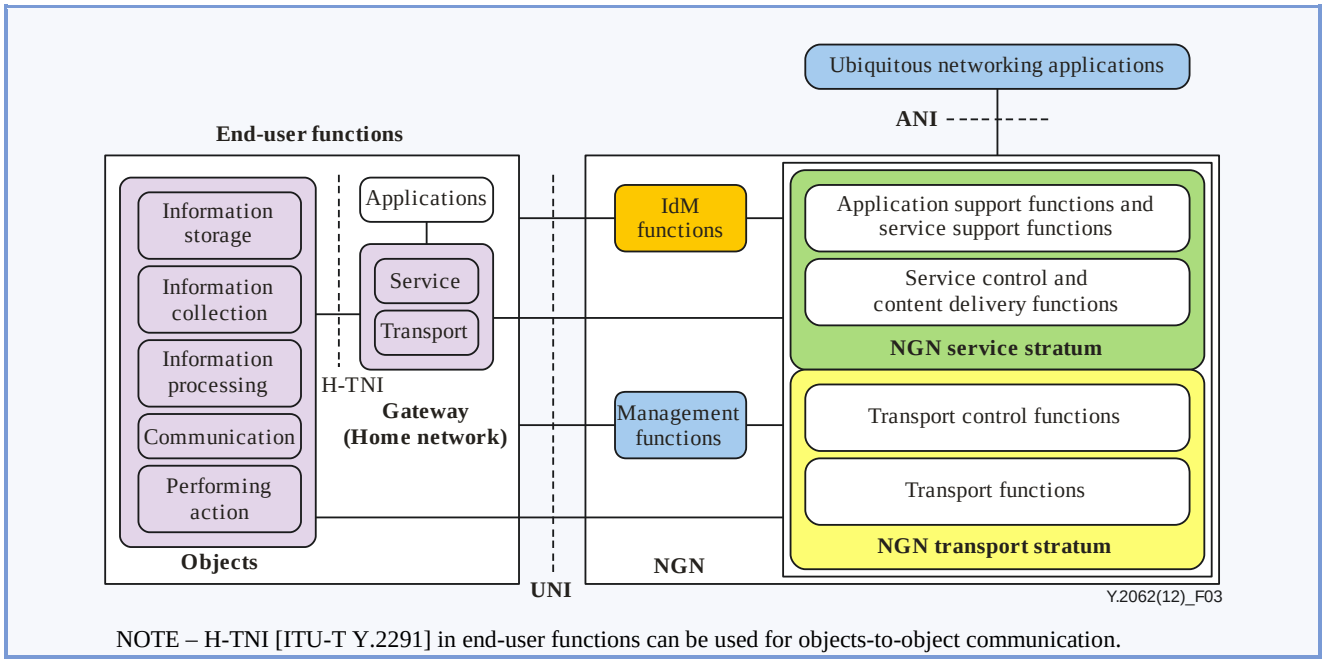


Figure 2 – Connecting-to-anything in the architectural model for ubiquitous networking in an NGN

Figure 3 shows a high-level architectural model for object-to-object communication with an NGN based on architectural model for ubiquitous networking in Figure 2.



NOTE – H-TNI [ITU-T Y.2291] in end-user functions can be used for objects-to-object communication.

Figure 3 – High-level architectural model for object-to-object communication with an NGN

Capabilities for ubiquitous networking are highly distributed and interact with infrastructure (e.g., an NGN) which interconnects with various types of objects. The infrastructure aims to continuously capture, manage and provide data about objects for applications. The following functions are required for object–to-object communication within an NGN.

End-user functions

End-user functions provide a set of functionalities with or without gateway (or home network) for connecting and collaborating objects on the end-user side. To support ubiquitous networking applications, objects support functions for information storage, information collection, information processing, communication, and performing actions.

Depending on the type of objects (see clause 7.2 and Appendix I), objects which have limited functionalities collaborate among other objects to provide additional functionality. Furthermore, objects need unique identifiers or names that can be used as a link in order to find and manage data to support ubiquitous networking applications.

NGN transport stratum

Due to the heterogeneities of objects, (e.g., different types of interfaces without supporting IP), there are restrictions to supporting a direct communication and data exchange between the objects. One of the functions of the transport stratum is to provide a bridge across this technological gap with end-user functions.

The transport stratum supports the exchange of messages among objects and the given applications. The transport stratum also maintains the list of objects and implements address mapping accordingly.

NGN service stratum

The usage context and the situation of objects are changed during the lifespan of the physical objects. For instance, as objects move along the supply chain, change owner and location, and are faced with changing environmental and regulatory conditions, the respective objects have to support different, often unpredictable, application scenarios. In close interaction with the transport stratum, the service stratum provides the required software repositories, as well as monitoring functionality to capture the current situation.

The function to establish highly available, scalable, and secure information management enables to automatically decide which portion of data is relevant in a given usage scenario and context.

The data that have been captured or processed at the transport stratum have also to be filtered or aggregated carefully depending on the data density and accuracy required by the respective applications. Filtering and aggregation of data can be applied at multiple semantic levels.

To find the requested information that may reside in different data storage systems distributed, locally managed data repositories, as well as a naming service, are supported.

Ubiquitous networking applications

Ubiquitous networking applications utilize and enrich the information that the underlying infrastructure provides in many ways. Data that have been previously collected by objects and persisted in repositories are used to support various applications among various stakeholders.

NOTE – IdM functions and management functions are common functionalities to be considered both on the end-user side and in NGN service/transport strata.

7 Basic concept of object-to-object communication

7.1 Objects in the ubiquitous networking environment

The object means the user or other entity that are connected to the network. It includes almost everything around us, such as remote monitoring and information devices, machine, or content.

As shown in Figure 4, the types of objects on the end-user side include the following:

- Personal devices
- Information devices
- RFID or sensors
- Contents
- Appliances
- Vehicles, trains, airplanes or any means of transportation.

These objects associated with humans are connected to an NGN through the user to network interface (UNI) with heterogamous networking environments in terms of network/access protocols and physical mediums. Various types of gateways and/or ad-hoc networks can be used to support connecting-to-anything capabilities with an NGN.

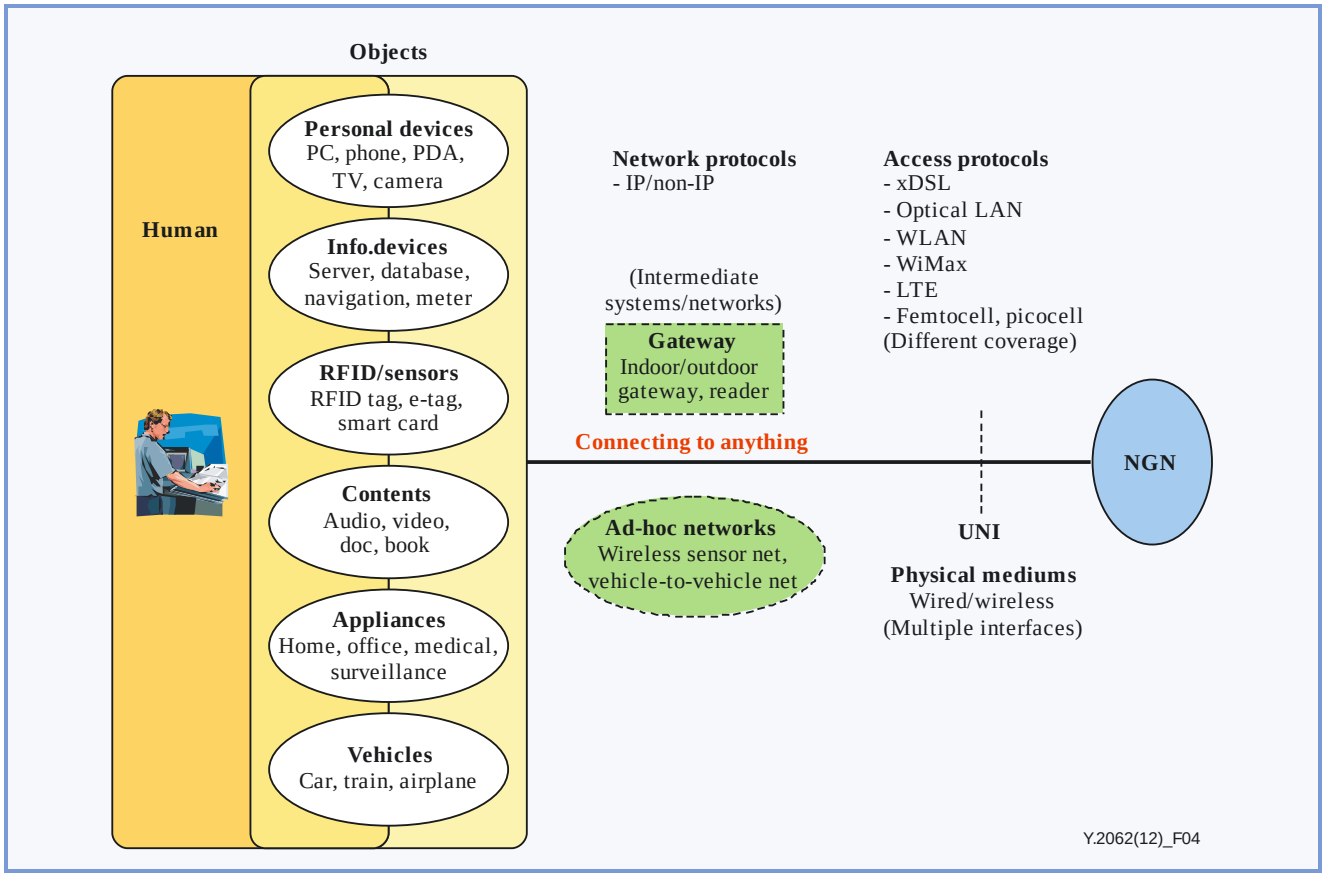


Figure 4 – Conceptual diagram for "connecting-to-anything" with an NGN

7.2 Characteristics of objects

Objects can be classified into several types as follows.

- Physical object vs. logical object (e.g., content and resource)
- Mobility: fixed object vs. mobile object
- Tag: Active RFID vs. Passive RFID

- Size: common (normal) devices vs. tiny (small) devices
- Power (energy): power supplied vs. power limited (for emergency)
- Manageability: managed by human vs. managed by device human intervened or not
- Different networking capabilities: IP vs. non-IP
  - In most cases, an unattended (constrained) device communicating with others objects in a potentially very large scale environment

Objects in an ubiquitous networking environment have the following characteristics:

- Heterogeneous access interfaces
- Lightweight protocol for low power consumption
- Different amount of information transactions.

Appendix I provides characteristics and examples of each type of object according to the classification of objects in an ubiquitous networking environment.

8 Requirements of "connecting-to-anything" capability for ubiquitous networking

8.1 General requirements for object-to-object communication

The following are general requirements for communications between objects in an NGN.

- For connecting an object, it is required to identify each object to be connected to the network.
- In the case of small-sized objects with limited power, the capabilities of the communication objects are less compared to high processing computing devices. To cope with such constraints on objects, it is required to use lightweight protocols which remove unnecessary loads.
- For configuring objects automatically, it is recommended to provide a self-configuration functionality.
- Auto-discovery is required to connect any objects which are in the range of communication.
- Objects can be moved from one place to another and may be attached to another network with different technology. Object mobility management is required to provide seamless communication among mobile objects.
- Network size is increasing as a lot of objects are connected into the network. Scalable solutions are required in order to cope with the increase of traffic and routing table size and the shortage of IP addresses.
- To support end-to-end connectivity, each object is recommended to have a separate, unique IP address. Adequate address space enables the connection of large numbers of objects to the network. Otherwise, it is recommended that each object provide the direct connectivity to host or gateway with a unique IP address.
- It is required to provide QoS and QoE of required level. Important objects need to handle on time and with some level of accuracy without communication errors for reliable services.
- Security and privacy are required to be managed in the proper way as connection to many sophisticated objects might cause huge damage if security is breached.

8.2 Technical considerations for object-to-object communication

Technical considerations for object-to-object communication of ubiquitous networking are as follows:

8.2.1 Identification

As there are various kinds of objects with different identifiers, it is required to support identification of each object and provide seamless communication through association with the network as well as the tracking of the object without restrictions of location.

8.2.2 Scalability

Scalability regarding addressing can be taken as an example. Object-to-object communication needs a huge number of IP addresses in order to uniquely identify each object. As a scalable solution, IPv6, which can accommodate as many objects as required to include in ubiquitous networking, can optionally be used.

8.2.3 Interoperability

Objects have different communication, information and processing capabilities. Each object is also subjected to very different conditions, such as power availability and a communication bandwidth requirement. The interoperability solution is required to be maintained to provide seamless interaction among objects. Otherwise, additional networking capabilities are required to be provided to support islands of objects in heterogeneous networks.

8.2.4 Service discovery

Suitable services for objects must be automatically identified. It is required to support an appropriate semantic means of describing their functionality. Self-configuration is required for each object to configure itself without manual/human intervention. For this, context information has critical roles to support context aware networking for changes of communication environments and to support semantic as the virtual representation of physical objects.

8.2.5 Data traffic

From the network perspective, it is difficult to handle a bulk amount of data if a large number of objects produce huge data depending on their applications/services. To solve this problem, it is required to develop solutions such as periodic communication between objects, data compression, and optimized traffic engineering.

8.2.6 Energy efficiency

As objects move around, it is difficult to connect to a power supply all the time and consequently they need to operate with a self-sufficient energy source. It is required to develop energy-efficient protocols to minimize power consumption and eliminate unnecessary communication procedure among objects.

8.2.7 Fault-tolerance

To maintain a robust, trustworthy and dynamic ubiquitous networking environment, it is required to support redundancy at several levels and ability in order to automatically adapt to abnormal conditions.

8.2.8 Security and privacy

Confidentiality, authenticity, and trustworthiness of communication partners are required to be maintained. Users may want to give objects limited service access so that they are not allowed to communicate in an uncontrolled manner.

8.2.9 Intelligence

An object is required to cooperate intelligently with its environment. Sensing the current environment, and acting intelligently according to the situation, is required to support services using object-to-object communication. Objects act according to their predetermined set of actions or they collaborate with each other based on the current context.

9 A mechanism for object-to-object communication: identity processing for connecting to anything

Instead of existing network terminals, new types of devices, such as RFID, sensors or smart cards, lead the change for the ubiquitous networking which enables devices to communicate among themselves. Identification, naming, and addressing capabilities are essential for supporting 'connecting-to-anything' in the end-user domain.

To support connecting-to-anything, there are specific technical considerations that take into account the following points:

- identification of object(s)
- finding/tracking the location of object(s)
- providing the connectivity to the NGN in cooperation with naming and addressing.

In ubiquitous networking, for object-to-object communications, information for several kinds of object on the top of end points should be identified in the network. A service is an entity that is either an instance of a specific application service or a specific data object. The identity of the object persists over time and is not tied to the end system hosting the service or data.

Identification of all objects for providing end-to-end connectivity in the ubiquitous networking environment is crucial. Identifier(s) in ubiquitous networking is/are capable enough of identifying all relevant objects and facilitating object-to-object communications. In particular, the globally unique identifier(s) enable a great many applications, including tracking, access control and protection of objects.

As shown in Figure 5, the layered architecture of an NGN requires specific processing capabilities at each layer. Each user and/or object in applications identifies by identity, like a name with a set of attributes of an entity. An attribute can be thought of as metadata that belongs to a specific entity in a specific context, some of which might be highly private or sensitive. The identity should be associated with object IDs (RFID, content ID, telephone number, URI, or URL) through identification and authorization. Each object ID should also be associated with communication IDs (session/protocol ID, IP address or MAC address) through mapping/binding.

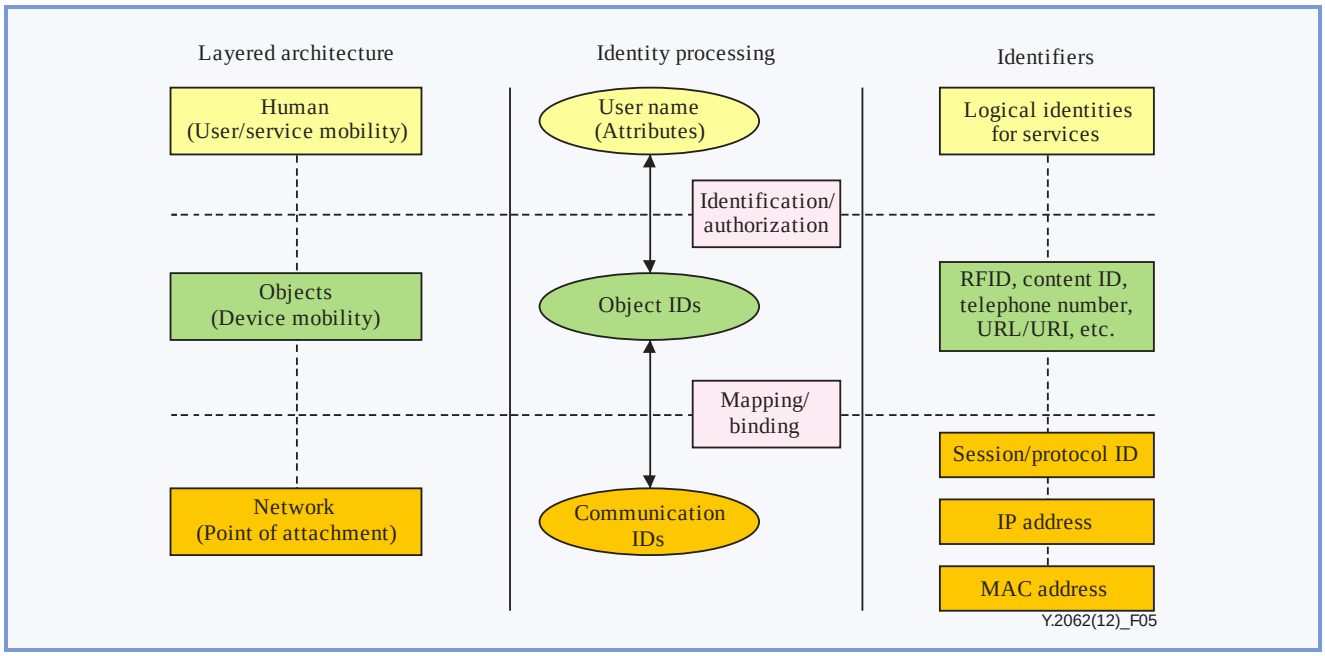


Figure 5 – Identity processing for connecting to anything

All objects (i.e., devices and contents) should be reachable by the other users/objects. Since managing a large number of different identification codes to use IP network infrastructure becomes vital, a mechanism is required to use both location information of the IP address and uniqueness of identification codes.

For connecting to anything using object identification, Figure 6 shows object mapping/binding with the IP address for IP connectivity to all objects on the end-user side. It provides the global connectivity with an NGN to objects through the association (e.g., mapping/binding) between the identifier of the object and the IP address.

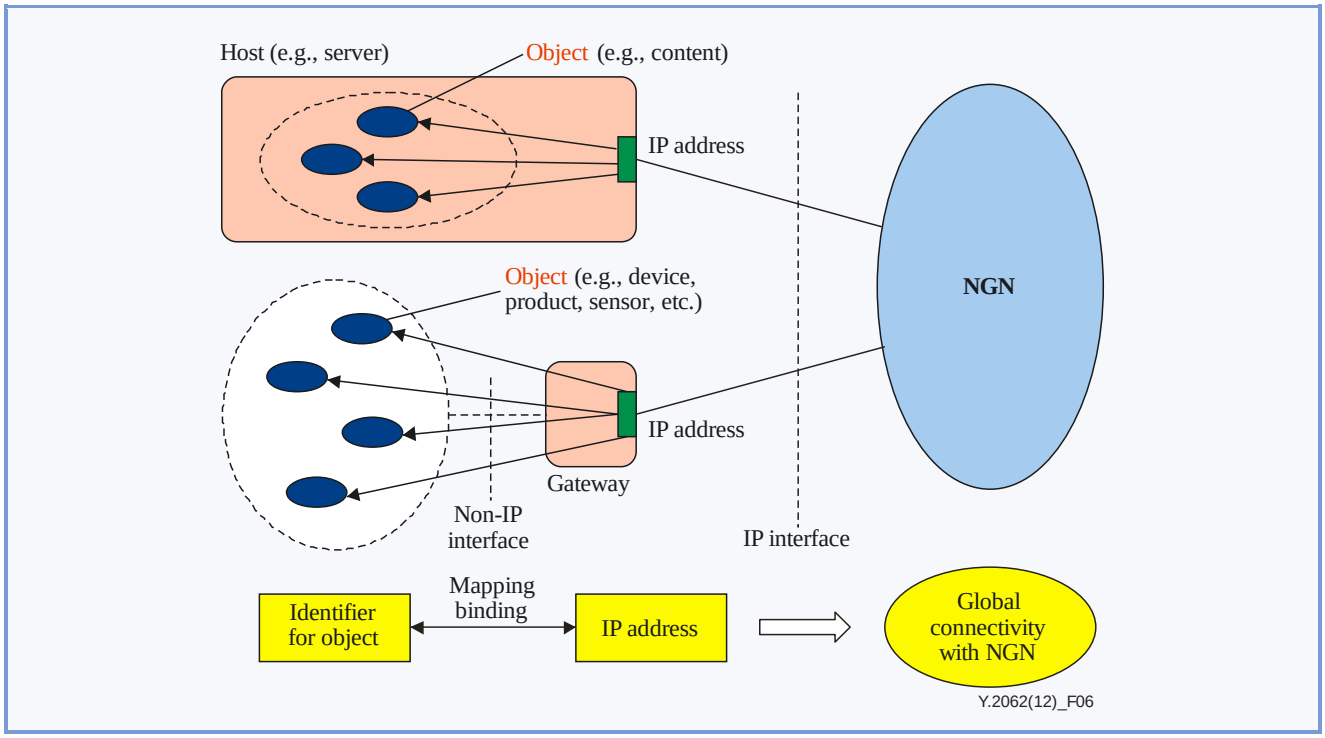


Figure 6 – Object mapping/binding with an IP address for connecting to anything

10 Security considerations

Basic considerations on security architecture for NGNs are addressed in [ITU-T Y.2001], while security requirements of the NGNs are described in [ITU-T Y.2701]. Concerning the specifics of ubiquitous networking, the various kinds of terminals, devices and contents that can be involved will have to conform to the security requirements of the network they are willing to attach. When attaching to the NGN, corresponding authentication and authorization requirements as described in [ITU-T Y.2702].

In this Recommendation, objects involved in NGNs have their own identities and are interconnected involving more interactions throughout a dynamic and heterogeneous environment. Accordingly, security is very crucial, including the design of the security architecture for a secure information discovery and delivery to users, including persons and objects.

Appendix I

Characteristics and examples of objects in the ubiquitous networking environment

(This appendix does not form an integral part of this Recommendation.)

Table I.1 shows characteristics and examples of each type of objects according to their classification. These objects are characterized by the following heterogeneities: order(s) of a magnitude bigger than the Internet, no computers or humans at endpoint, inherently mobile, disconnected, unattended, communication style and so on.

Table I.1 – Characteristics and examples of objects in the ubiquitous networking environment

Types		Characteristics	Examples
Size	Small objects	Small in size, short communication range.	Sensor, tiny devices.
	Normal objects	No constraint in size.	Home appliances.
Mobility	Mobile objects	Moveable, continuous change in context information.	Car, bus, and train.
	Fixed objects	Do not move normally and can be connected to power grid.	Traffic light, building, bridge.
Power	Objects without power supply	Do not have continuous power, battery with fixed period of uses.	Sensor in outdoor, RFID.
	Objects with power supply	Connected with power supply, no need to be worried about energy consumption.	Home appliances.
Connectivity	Objects connected to physical world	Objects are connected to physical world to provide data and information about some real time phenomenon. Objects are not only able to sense physical information but also able to react according to need.	Environment sensors (measuring temperature, pressure, humidity, rain). Actuators, robots, automatic application triggering depend on context information.
	Intermittent connectivity	Objects communicate and collaborate intermittently (periodically or based on some contextual condition).	Sensor which sends data in every pre-defined time interval. Actuator or robot which act according to its surrounding: increasing, lowering temperature, calling security, fire or emergency services based on environmental conditions.
Ability	Ability to sense and actuate	Object sense the environment where it is subjected. Object can also react based on sensed information.	Normal sensors, actuators (senses and react dynamically). (Note)

Table I.1 – Characteristics and examples of objects in the ubiquitous networking environment

Types		Characteristics	Examples
People involvement	Object of interest of people	People can augment communication and computation properties on the physical objects.	Tagged food item, electric lamp with light sensor, video content with automatically pause and play capabilities, cup with thermometer.
	Objects managed by devices not people	These objects are managed by other devices rather than people themselves.	Smart meter managing light sensors around home, Home automation system managing automatic door and windows system.
Physical/ logical	Physical objects	All physical objects related to real time activities falls in this category.	Different sensors attached with physical objects (lamp, environment, tree).
	Logical objects	Can be identified as a resource or a virtual object by using a unique identifier.	Contents and resources (e.g., software, computing power, storage).
Object with tag	Object with active tag	Object can be tagged with active RFID tag.	Products attached with active RFID.
	Object with passive tag	Passive tag can be attached to objects which need to be uniquely identified.	Items tagged with passive RFID in shipping company, supermarket.
IP/ Non IP	IP enabled object	IP enabled objects are capable of having end to end connectivity.	Refrigerator in home which has own IP address, TV, electric lap with processing devices.
	Non IP enabled object	Non IP enabled objects participate in network with the help of some gateway or middleware which acts on behalf of the objects.	Non IP objects: tiny devices, products with active or passive RFID tag.
NOTE – Classification according to different roles/names can be considered as follows: – A sensor: device that measures a physical quantity and converts it to analogue or digital signal (e.g., power consumption and quality, vibration of an engine, pollution, temperature, motion detection). – An actuator: device that controls a set of equipment (e.g., control and/or modulates the flow of a gas or liquid, control electricity distribution, perform a mechanical operation).			

Appendix II

Ubiquitous networking applications and examples using object-to-object communication

(This appendix does not form an integral part of this Recommendation.)

In the ubiquitous networking environment, applications can be newly created through the integration and combination of technologies such as bio technology (BT), nano technology (NT) and content technology (CT). Therefore, it is necessary to combine BT, NT and CT as well as information technology (IT) using ubiquitous networking capabilities.

Communication networks have been mainly supporting the evolution of information processing and service capabilities within IT industries. However, the capabilities of networks benefiting from ubiquitous networking should impact other industries such as the medical industry, education industry, finance industry, transportation/distribution industry, etc., resulting in new requirements for specific services taking IT into consideration.

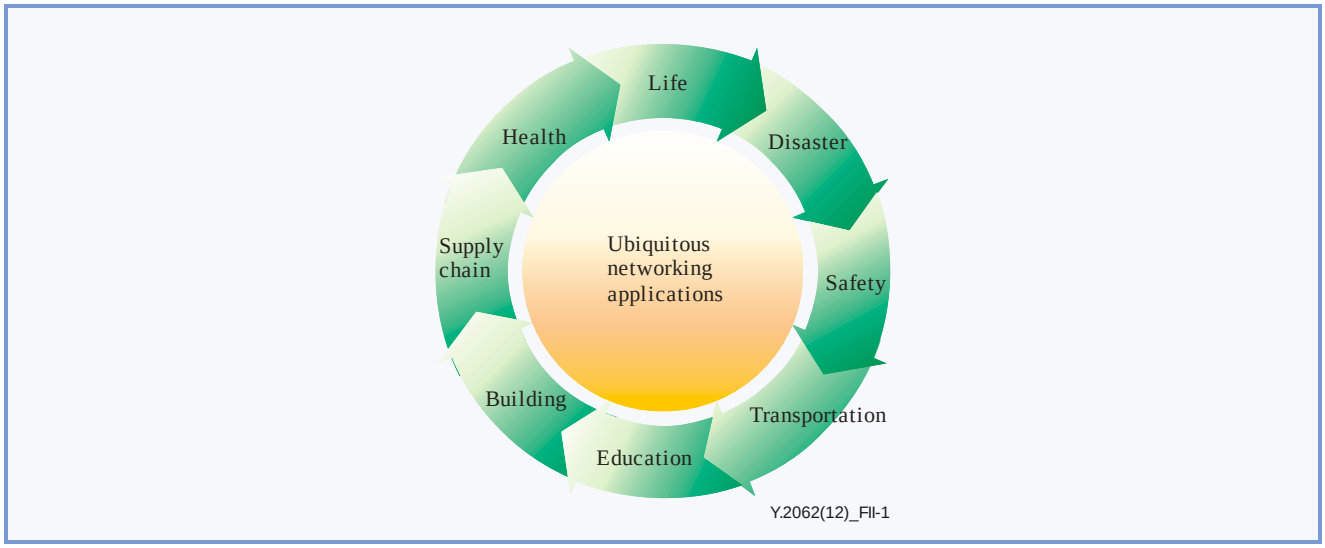


Figure II.1 – Examples of ubiquitous networking applications

As shown in Figure II.1, the technologies using connecting-to-anything capabilities can be used for the following applications/services in the convergence environment with IT:

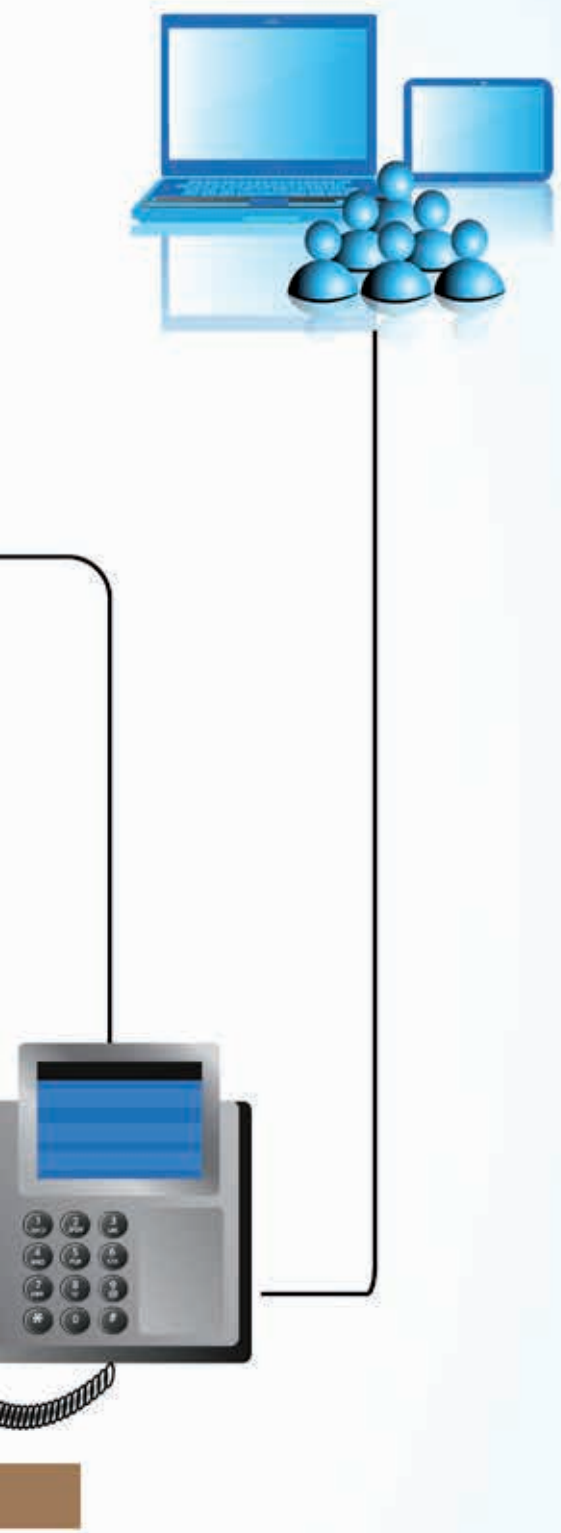
- IT + transportation: ITS, networked vehicle and telematics, navigation IT + education: online cyber learning system
- IT + building: IBS, home networking, etc.
- IT + supply chain: supply chain management, distribution system
- IT + health: remote diagnosis and medical experimentation
- IT + life: environment management, equipment management
- IT + disaster: disaster management, emergency alarming system
- IT + safety: finance, commerce, public peace.

For supporting ubiquitous networking applications, NGNs shall solve technical considerations which are specified in clause 8.

Bibliography

[b-ITU-T Y.2011] Recommendation ITU-T Y.2011 (2004), *General principles and general reference model for Next Generation Networks*.

[b-ITU-T Y.2201] Recommendation ITU-T Y.2201 (2009), *Requirements and capabilities for ITU-T NGN*.



Y.4405/H.621

Architecture of a system for multimedia information access triggered by tag-based identification

Architecture of a system for multimedia information access triggered by tag-based identification

Summary

Recommendation ITU-T H.621 defines the system architecture for the multimedia information access triggered by tag-based identification on the basis of Recommendation ITU-T F.771, and serves as a technical introduction to subsequent definition of detailed system components and protocols. The services treated by this Recommendation provide the users with a new method to refer to the multimedia content without typing its address on a keyboard or inputting the name of objects about which relevant information is to be retrieved. This is one of the major communication services using identification (ID) tags such as radio frequency identifications (RFIDs), smart cards and barcodes. International standardization of these services will give a big impact to international multimedia information services using ID tags. It contains the functional model, its constituent components as well as its workflow. An appendix describes how this architecture realizes typical services.

Editorial note – This document includes in clean text the changes introduced by Amd. 1 (2014).

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T H.621	2008-08-06	16	11.1002/1000/9490
1.1	ITU-T H.621 (2008) Amd.1	2014-10-14	16	11.1002/1000/12246

Keywords

Multimedia information access, tag-based identification.

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

Table of Contents

Introduction

This Recommendation defines the system architecture for multimedia information access triggered by tag-based identification and serves as a technical introduction to subsequent specifications of detailed system components and protocols. It contains the functional model, its constituent components as well as its workflow. An appendix describes how this architecture realizes typical services.

Recommendation ITU-T Y.4405/H.621

Architecture of a system for multimedia information access triggered by tag-based identification

1 Scope

This Recommendation defines the following issues to cover multimedia information access services triggered by tag-based identification as defined in [ITU-T F.771]:

– a functional architecture reference model with descriptions of corresponding elements;

– interface protocols between communication elements; and

– a generic work flow to support multimedia information access triggered by tag-based identification.

Moreover, this Recommendation describes implementation examples with work flows.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T F.771] Recommendation ITU-T F.771 (2008), *Service description and requirements for multimedia information access triggered by tag-based identification*, including its Amendment 1 (2014).
<http://www.itu.int/rec/T-REC-F.771>

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 ID resolution: [ITU-T F.771].

3.1.2 ID tag: [ITU-T F.771].

3.1.3 ID terminal: [ITU-T F.771].

3.1.4 identifier: [ITU-T F.771].

3.1.5 multimedia information: [ITU-T F.771].

3.1.6 multimedia information delivery function: [ITU-T F.771].

3.1.7 real-world entity: [ITU-T F.771].

3.1.8 tag-based identification: [ITU-T F.771].

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

2D	Two Dimensional
3D	Three Dimensional
3G	Third Generation wireless systems
CD	Compact Disk
DNS	Domain Name Server
DVD	Digital Versatile Disk
GW	GateWay
HTTP	HyperText Transfer Protocol
ID	Identification
IDR	Identification Resolver
IDT	Identification Terminal
IP	Internet Protocol
IR	Infrared
IRS	Identification Resolution Server
MIDF	Multimedia Information Discovery Function
MIDS	Multimedia Information Delivery Server
MIHF	Multimedia Information Handling Function
MIM	Multimedia Information Manager
MMS	Multimedia Messaging Service
NFC	Near Field Communication
NGN	Next Generation Network
ORM	Optically Readable Media
P2P	Peer to Peer
PDA	Personal Digital Assistant
RF	Radio Frequency
RFID	Radio Frequency Identification
R/W	Reader/Writer
SB	Service Broker
SIM	Subscriber Identity Module
SMS	Short Message Service
URL	Uniform Resource Locator
WAP	Wireless Application Protocol
Wi-Fi	Wireless Fidelity

5 Conventions

In this Recommendation:

– The expression "**is required to**" indicates a requirement which must be strictly followed and from which no deviation is permitted if conformance to this Recommendation is to be claimed.

– The expression "**is recommended**" indicates a requirement which is recommended but which is not absolutely required. Thus this requirement need not be present to claim conformance.

– The expression "**can optionally**" indicates an optional requirement which is permissible, without implying any sense of being recommended. This term is not intended to imply that the vendor's implementation must provide the option and the feature can be optionally enabled by the network operator/service provider. Rather, it means the vendor may optionally provide the feature and still claim conformance with the specification.

6 System functional architecture

This clause defines the functional architecture of multimedia information access systems in which the multimedia information access is triggered by tag-based identification. This architecture is based on the system components described in clause 6 of [ITU-T F.771] and shown in Figure 1. Compared with the high-level functional architecture, this system architecture decomposes each component into more detailed functional components.

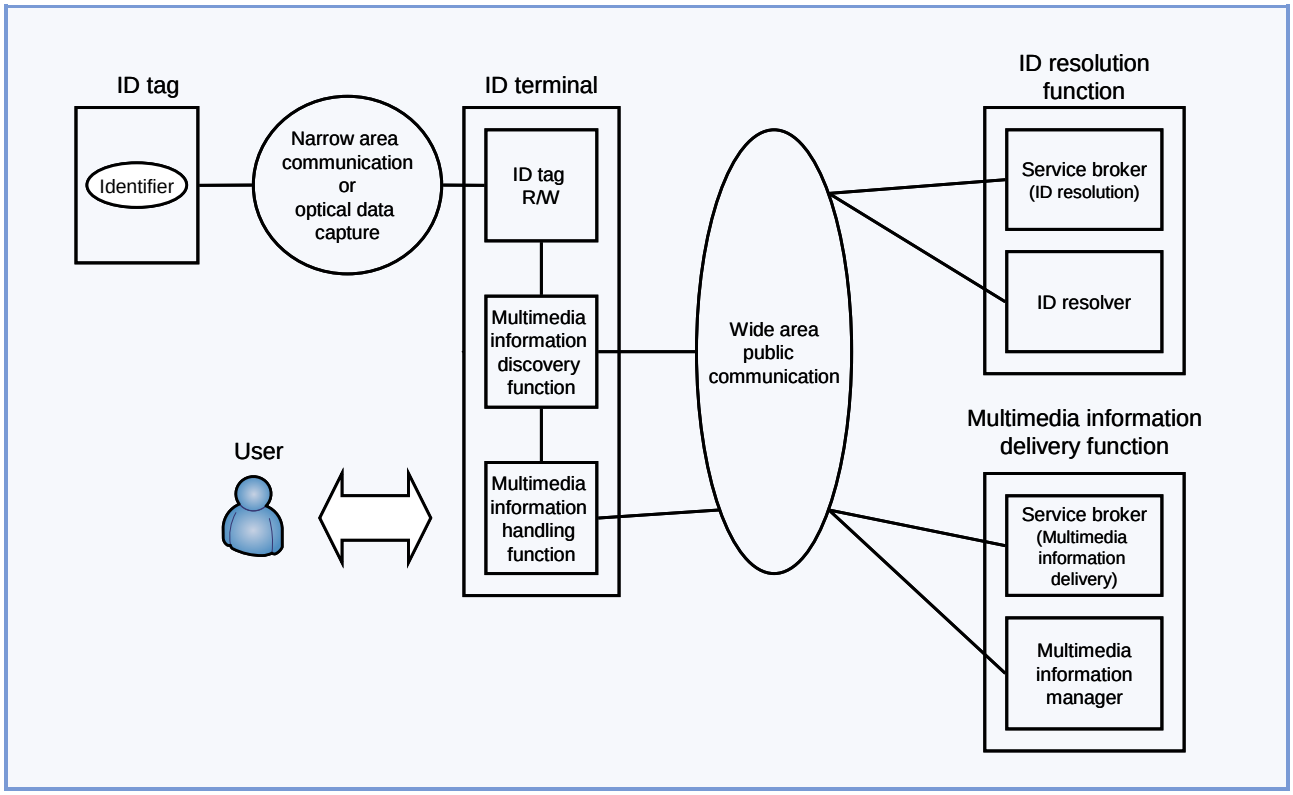


Figure 1 – Functional architecture

Figure 1 shows the logical functional architecture. This does not show the physical implementation of each high-level functional component. Examples of corresponding physical level architecture and their implementation are described in Appendix I.

6.1 Functional components

The system functional architecture for multimedia access triggered by tag-based identification is required to include the following components: ID tag, ID tag reader/writer (ID tag R/W, in short), multimedia information discovery function (MIDF), multimedia information handling function (MIHF), service broker (SB), ID resolver (IDR), and multimedia information manager (MIM). ID tag R/W, MIDF, and MIHF are sub-components included in an ID terminal. The IDR and SB are sub-components included in an ID resolution function. The MIM and SB are sub-components included in a multimedia information delivery function. Refer to clause 6 of [ITU-T F.771] regarding narrow area communication and wide area public communication.

6.1.1 ID tag

An ID tag is required to store identifier(s) which can be read by an ID tag R/W in an ID terminal via narrow area communication. It can optionally store multimedia information and/or other data that is used in ID resolution and/or multimedia information presentation. Typical examples of ID tags are RFID, smartcard, infrared tag, barcode, 2D barcode, NFC listening device, etc.

6.1.2 ID terminal

An ID terminal is required to be composed of three sub-components: 1) ID tag R/W; 2) multimedia information discovery function (MIDF); and 3) multimedia information handling function (MIHF). It can optionally contain multimedia information and/or other data. This data, such as a user's profile, can be used in ID resolution and/or multimedia information presentation.

6.1.2.1 ID tag R/W

An ID tag R/W is required to provide communication interfaces to an ID tag, and read a single or multiple identifier(s) as well as application data from the ID tag. After reading the identifiers, it sends their information to the MIDF. An ID terminal can optionally contain multiple ID tag R/Ws where a selection function of frequency bands is required to choose a proper ID tag R/W interface against multiple RF types of ID tags such as HF-type ID tags and UHF-type ID tags. The selection function may be provided via a manual selection user interface, an automatic scanning function or other ways which are an implementation issue. Similarly, selection function of optically readable media (ORM) (1-dimentional, vs 2-dimentional, different type of 2-dimentional codes, etc.) is an optional feature which can be provided via a manual selection user interface, an automatic scanning function or other manners which are an implementation issue.

6.1.2.2 Multimedia information discovery function (MIDF)

A multimedia information discovery function (MIDF) is required to obtain the identifier from an ID tag R/W, and issues queries to the IDR or optionally the SB, depending on implementations, via wide area public communication. It uses the identifier as a query key in both cases. The ID resolver returns pointer information (e.g., URL) to access the MIM providing the multimedia information delivery services. After obtaining the pointer information, it sends the information to the MIHF.

6.1.2.3 Multimedia information handling function (MIHF)

A multimedia information handling function (MIHF) is required to provide a function to download multimedia information from the MIM, and presents the information to the user. It can optionally upload information to the MIM.

6.1.3 Service broker (SB)

A service broker can optionally provide ID resolution services with the help of an ID resolver. When an ID terminal sends an identifier to the SB, it consults the ID resolver for resolution of the identifier, discovers the multimedia information access information and responds by sending the corresponding resolution result to the ID terminal. That is, the SB works as a proxy for the ID resolver.

A SB can optionally provide multimedia information handling functions as well as the ID resolution proxy functions. That is, the SB can work as a multimedia information delivery proxy as well. Existence of the SB and its features depend on implementations.

6.1.4 ID resolver (IDR)

An ID resolver is required to preserve the relationship between an identifier and its pointer information, such as URL, IP address and phone number, to access the multimedia information delivery function. It is required to provide the MIDF and SB with a translation service from the identifier into the pointer information.

6.1.5 Multimedia information manager (MIM)

A multimedia information manager is required to receive a request from the MIHF in the ID terminal, and delivers multimedia information to it. It can optionally receive uploaded multimedia information from the MIHF.

6.2 Protocols

This architecture is required to be supported by the following standard protocols on the interfaces among the functional components described in clause 6.1. Figure 2 shows those interfaces.

6.2.1 ID tag communication protocol

The ID tag communication protocol is used by the ID terminal and ID tag for their data exchanges and allows the ID terminal to obtain an identifier from the ID tag.

6.2.2 ID resolution protocol

The ID resolution protocol is used by the MIDF and IDR for ID resolution services. The SB is required to use this protocol to interwork with the IDR.

6.2.3 Service broker protocol

The service broker protocol is a communication protocol used between the MIDF and SB, and also between the MIHF and SB.

6.2.4 Multimedia information delivery protocol

The multimedia information delivery protocol is a communication protocol used between the MIHF and MIM.

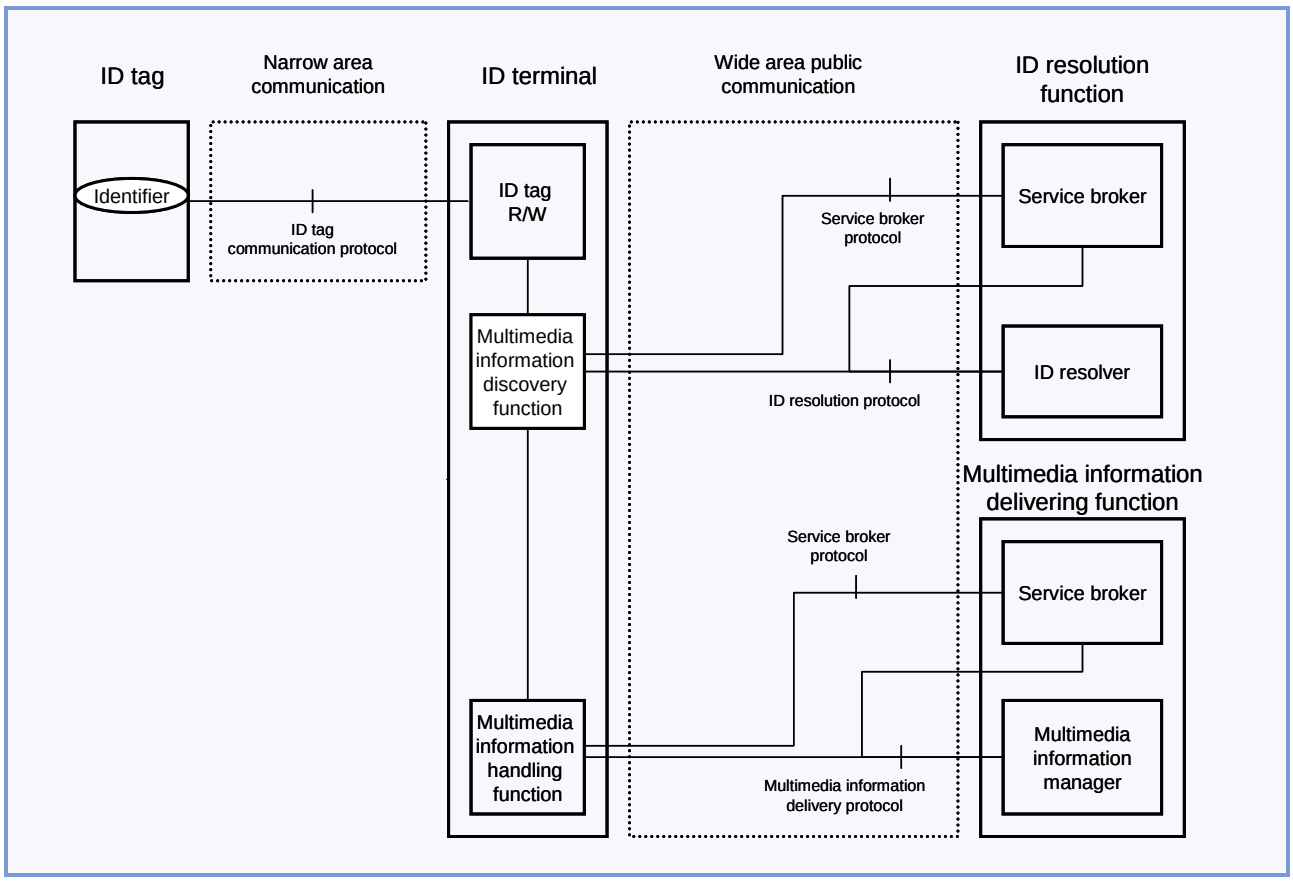


Figure 2 – Interfaces between components in functional architecture

6.3 General workflow

This clause describes the high-level workflow that realizes multimedia access triggered by tag-based identification. This architecture is recommended to work according to the following workflow (see Figure 3).

1) Identifier in the ID tag is obtained by the ID tag R/W in the ID terminal.

2) ID tag R/W sends the identifier to the MIDF.

3) MIDF sends the identifier to the IDR to discover pointer information of the multimedia information delivery function related to the identifier.

   This communication can optionally be mediated by the SB. In this case, the MIDF requests the SB to make ID resolution (3-1), then the SB consults the IDR and retrieves the pointer information of the multimedia information delivery function (3-2).

4) IDR resolves the identifier, finds the pointer information of the multimedia information delivery function related to the identifier, and then returns it to the MIDF.

   This communication can also optionally be mediated by the SB. In this case, the IDR first sends a reply, including the pointer information to the SB (4-1), and then the SB forwards the reply to the MIDF (4-2).

5) MIDF invokes the MIHF by forwarding the pointer information.

6) MIHF sends the request of retrieving the multimedia information service to the MIM in the multimedia information delivery function.

   This communication can optionally be mediated by SB. In this case, the MIHF requests to the SB (6-1), then the SB forwards the request to the MIM (6-2).

7) MIM provides the multimedia information service to the MIHF.

   This communication can also optionally be mediated by the SB. In this case, the MIM first provides the service to the SB (7-1), and then the SB mediates the service to the MIHF (7-2).

8) MIHF plays the information and shows it to the user, or it uploads multimedia information to the MIM.

Examples of this workflow are described in Appendix II.

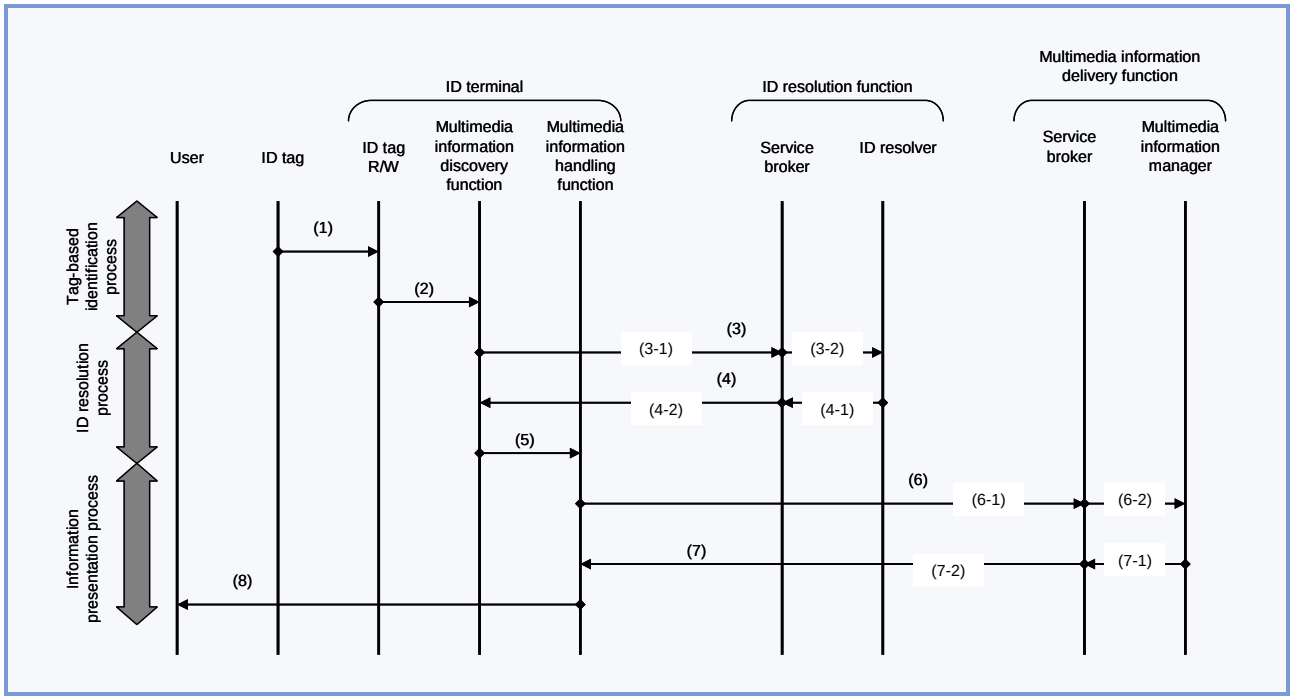


Figure 3 – General workflow of tag-based identification triggered multimedia information access

Appendix I

Examples of physical level architecture

(This appendix does not form an integral part of this Recommendation)

I.1 Configuration example of physical level architecture

This appendix describes examples of physical level architecture based on the functional system architecture defined in this Recommendation. In the example shown in Figure I.1, the architecture consists of five types of physical components: 1) ID tags; 2) ID terminals (IDTs); 3) service broker (SB); 4) ID resolution servers (IRSs); and 5) multimedia information delivery servers (MIDSs). A wide area public network provides only end-to-end connection among IDTs, SBs, IRSs, and MIDSs, and it is divided into an IP network and other networks, such as mobile networks, which are interconnected by gateways. IDTs can be connected to other networks, and use the multimedia access triggered by tag-based identification (non-IP IDT).

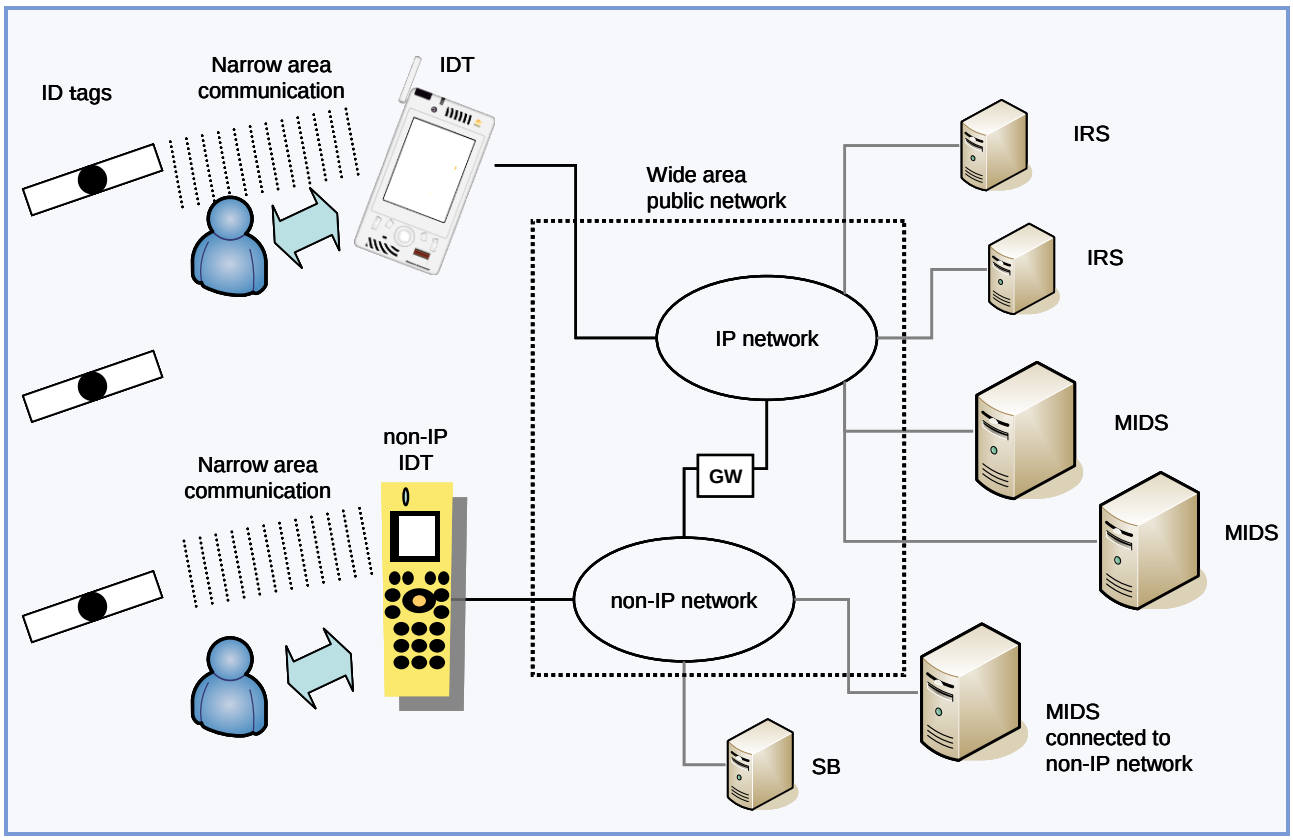


Figure I.1 – An example of physical level architecture

I.2 Components

ID tag

The ID tag contains identifier(s) of object, person and location. It may be RFID, RF/IR tag, barcode or 2D barcode. In usual cases, a single ID tag contains an identifier. However, a single ID tag may contain multiple identifiers. Alternatively, some ID tags are equipped with anti-collision communication or multiplexed communication mechanisms, which enable the tag reader/writer to communicate with multiple ID tags simultaneously.

ID terminal (IDT)

The ID terminal (IDT) implements the three functional components: ID tag reader/writer, multimedia information discovery function (MIDF), and multimedia information handling function (MIHF). Some IDTs, such as a PDA with a Wi-Fi facility and IP protocol stacks, may connect to an IP network directly, and also some other IDTs may connect to a non-IP network such as a mobile network which is interconnected to the IP network by a gateway. Additionally, Internet browsers are a popular implementation of an MIHF.

Service broker (SB)

The service broker (SB) works as a gateway or proxy of the ID terminal. It also works as a proxy of the MIDF.

ID resolution server (IRS)

The ID resolution server (IRS) realizes the function of ID resolver (IDR). The number space of identifiers is managed by multiple distributed ID resolution servers, which are connected to the IP network and cooperate with each other. To resolve an identifier into the pointer for the multimedia information delivery server, the resolution query is sent to multiple ID resolution servers (Figure I.4).

Multimedia information delivery server (MIDS)

The multimedia information delivery server (MIDS) realizes the function of multimedia information manager (MIM). Generally speaking, a single MIDS can provide multiple services. In this architecture, there may be a huge number of MIDSs, which are connected to an IP network or non-IP networks as peer nodes. Typical examples of the MIDSs are web servers, video/audio streaming servers, etc.

Narrow area communication

The narrow area communication connects ID tags and the ID tag R/W in the ID terminal. It has various types depending on the kinds of ID tag (see clause I.3 for examples). In most cases, the communication range of this network is less than a few metres.

Wide area public network

The wide area public network connects IDTs, IRSs, SBs and MIDSs. This architecture requires only end-to-end reliable connections among these components to the underlying wide area public network. In this architecture, the IP network is supposed to be the primary network, and other networks, such as mobile networks, will be interconnected by gateways. Some IDTs may be connected to the IP network directly, and some IDTs (for example, cellular phones) may be connected to other networks. In the same way, MIDSs may be connected to either IP or non-IP networks.

I.3 Implementation examples of narrow area communication between ID tag and ID terminal

I.3.1 Variations of narrow area communication between ID tag and ID tag R/W

Narrow area communication between ID tag and ID terminal is implemented by various communication technologies, mainly depending on the kinds of ID tag (Table I.1).

Table I.1 – Variations of narrow area communication between ID tag and ID tag R/W

ID tag	ID tag R/W in ID terminal	Narrow area communication
Passive RFID	RFID R/W	Contactless communication of RFID such as [b-ISO/IEC 18000-x]
Contactless smart card	Smart card R/W	Contactless communication of smart card such as [b-ISO/IEC 14443-x], and NFC reader/writer mode
Contact smart card	Smart card R/W or smart card socket	Contact communication of smart card such as [b-ISO/IEC 7816-x]
Barcode, 2D barcode	Camera with image recognition or laser scanner	Image acquisition
Infrared tag	Infrared transceiver	Infrared communication
Active RFID	Base station	Narrow area wireless communication such as Bluetooth, ZigBee, Wi-Fi and [b-ISO/IEC 18000-4]
NFC listening device	NFC polling device	NFC reader/writer mode, NFC peer mode

I.3.2 Narrow area communication implementation using NFC

For example, if we adopt near field communication (NFC) for the narrow area wireless communication network, the ID tag and ID terminal can take several novel forms. Figure I.2a takes the normal form of ID tag and ID terminal in a mobile phone. In Figure I.2b, the ID tag function is implemented by the reader/writer device in a mobile phone, and the identifier is read by an ID terminal implemented as a desktop PC. It is also possible to implement an ID tag and ID terminal in one device together.

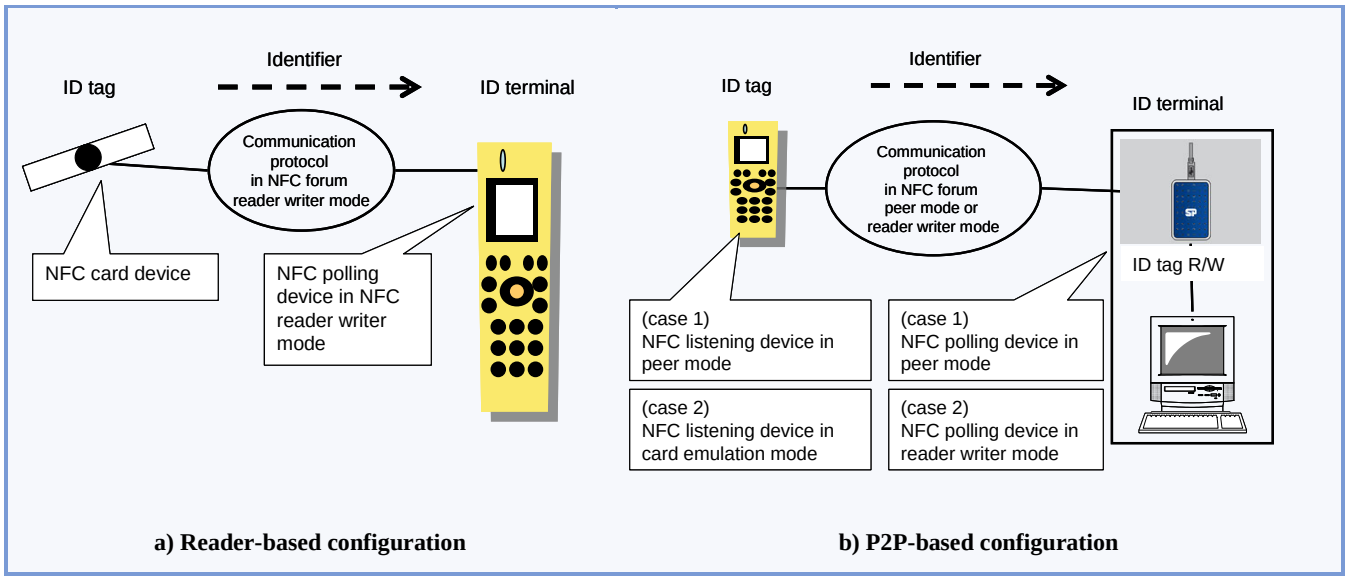


Figure I.2 – Examples of narrow area communication using NFC

I.3.3 Wired narrow area communication

Narrow area communication includes wired or contact communication of smart card tags such as those described in [b-ISO/IEC 7816-x]. For example, if the [b-ISO/IEC 7816-x] reader/writer is implemented as an external device for an ID terminal, it takes the form illustrated in Figure I.3a. It is also possible to implement the reader/writer as an internal device for an ID terminal. Figure I.3b illustrates this configuration. Many 3G mobile terminals include a SIM socket and a small smart card is embedded into the socket. The system architecture in this Recommendation covers this type of system configurations. In this configuration, an ID terminal can obtain the identifier and use the multimedia information access at any time because it always carries ID tag(s) within it. On the other hand, it cannot change the identifier for the multimedia information access because it is fixed inside the ID terminal. If the user wants to change the identifier, he/she has to exchange the smart card manually.

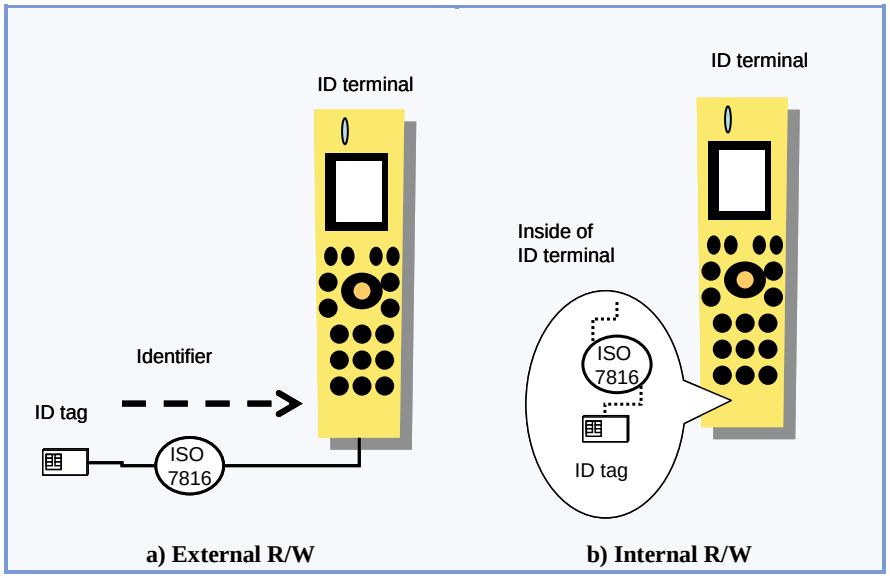


Figure I.3 – Examples of wired narrow area communication

I.4 Distributed implementation of ID resolution server

The total number of identifiers used in multimedia information access is expected to be very large. From the point of view of ID resolution query performance, and from the point of view of identifier space management, distributed implementation of IRSs is necessary. Figure I.4 illustrates the configuration of distributed IRSs in a tree structure and the ID resolution process on the basis of this configuration. In this example, identifiers are managed by multiple distributed IRSs, which are connected to the IP network and cooperate with each other. To resolve an identifier into a pointer to MIDS, the resolution query is sent to multiple IRSs. In this example, a query is processed as for domain name servers (DNSs) on the Internet. IRSs near to the root are responsible for resolving upper bits of identifiers, and IRSs near to leaf are responsible for resolving the lower bits. In Figure I.4, the grey part of identifier represents the bits managed by each IRS.

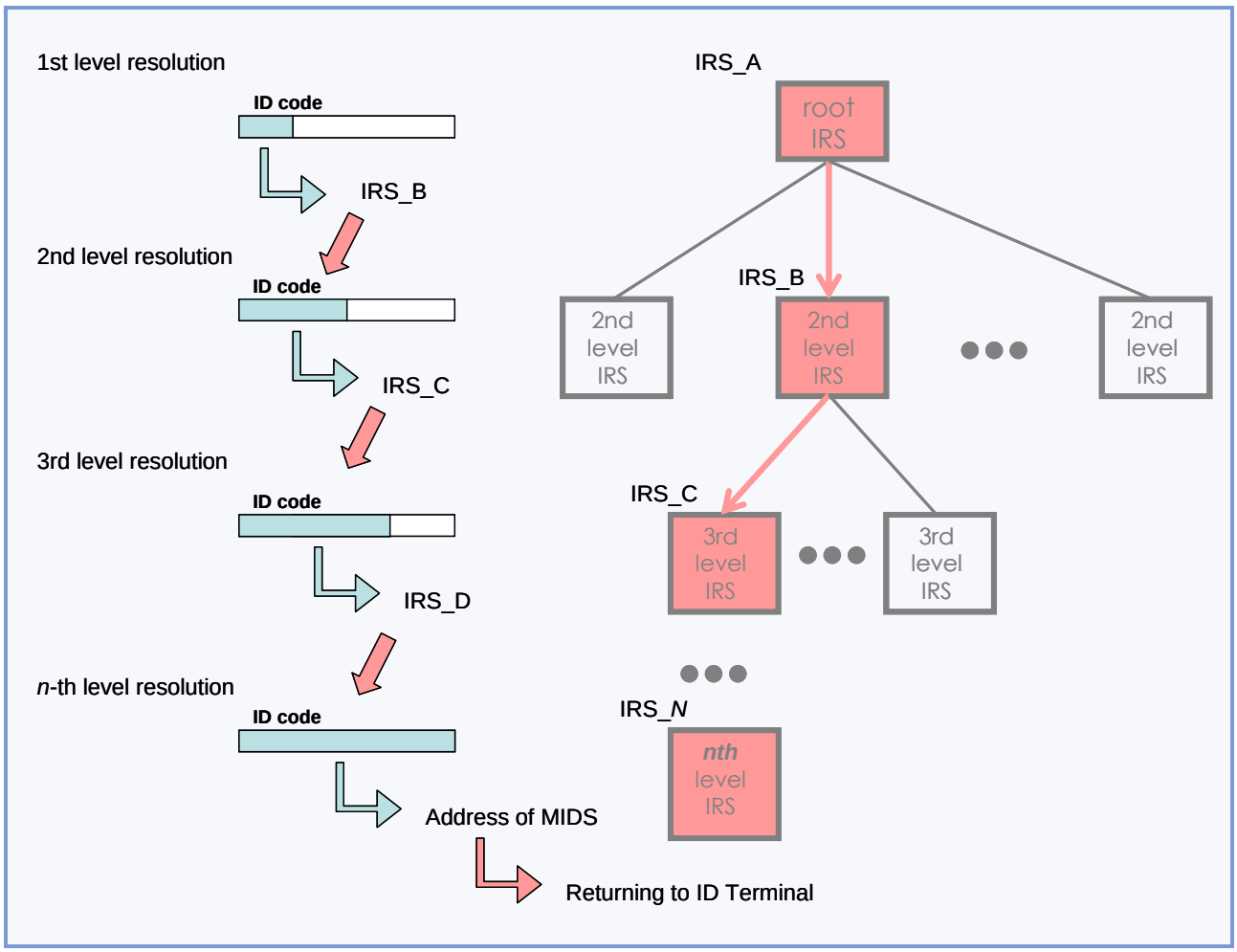


Figure I.4 – Cascade search for ID resolution

Appendix II

Workflow examples for multimedia information access triggered by tag-based identification

(This appendix does not form an integral part of this Recommendation)

This appendix presents seven typical examples of multimedia information access triggered by tag-based identification included in [ITU-T F.771]. For each example, its application scenario is described, and then the functional architecture and associated workflow are presented. Here, to make the description simple, the service broker is not used in the architecture and workflow.

II.1 Location-aware multimedia information service

II.1.1 Application scenario

Location-aware information services are among the most important applications of RFID. They provide location-aware information once the RFID and the active tag (beacon), which are attached and installed to the physical infrastructure (e.g., road), are read by passers-by and vehicles that try to access location-specific information.

An identifier is assigned to the business information of a shop. The same content may be accessed by many methods, such as short message service (SMS), multimedia messaging service (MMS), wireless application protocol (WAP) and hypertext transfer protocol (HTTP), in several media types, such as text, image and video. The most appropriate method can be selected according to the capabilities of the ID terminal.

II.1.2 System architecture

The architecture in Figure II.1 implements the above scenario.

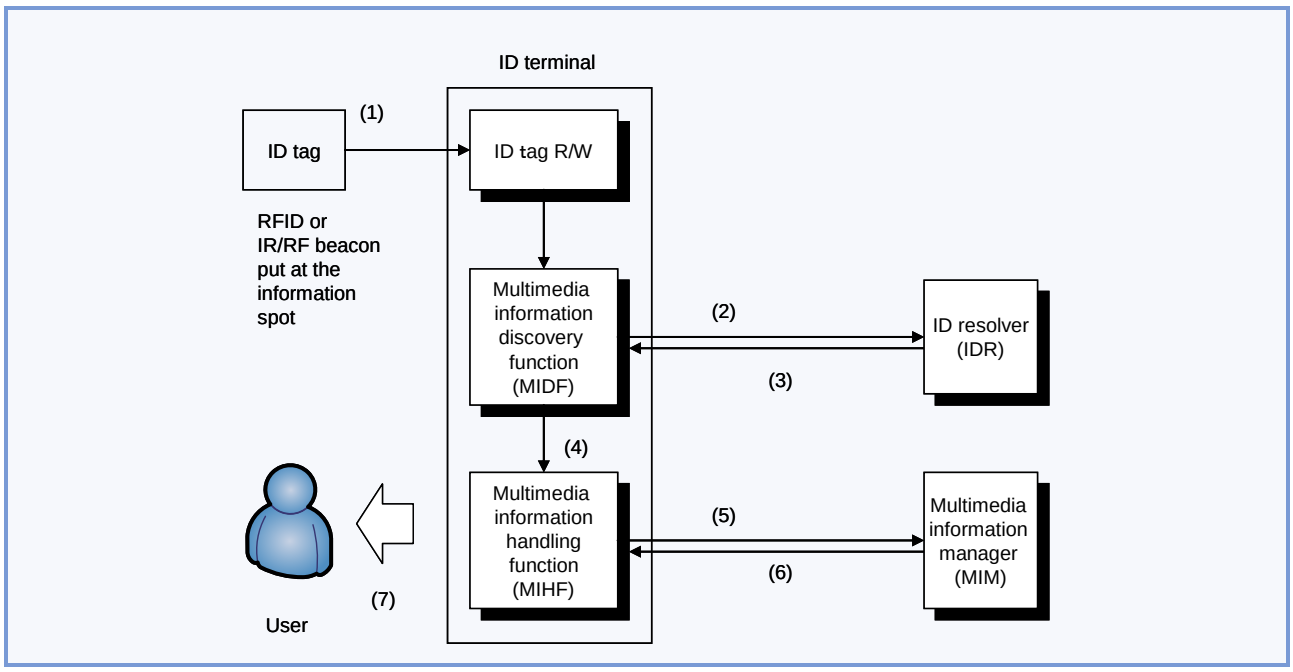


Figure II.1 – Implementation architecture of location-aware information delivery services

II.1.3 Workflow

1) ID tag R/W obtains the identifier of the physical location (location identifier) from RFID or IR/RF beacon.
2) Multimedia information discovery function (MIDF) sends the location identifier to ID resolver (IDR) to find the pointer and transfer protocol information of the associated multimedia information managers (MIMs).
3) The pointer and transfer protocol of the MIMs are provided to the MIDF.
4) MIDF sends the information of pointer and transfer protocol to the multimedia information handling function (MIHF).
5) MIHF sends a multimedia information delivery service request to the MIMs which contain detailed information associated with the location identifier.
6) The multimedia information associated with the location identifier in the tag is delivered to the MIHF in the ID terminal.
7) User watches the multimedia information displayed by the ID terminal.

II.2 Multimedia information download via posters service

II.2.1 Application scenario

An RFID tag containing a movie identifier is attached to an advertisement poster for the movie. Multimedia information may be associated with this identifier, such as images, audio/music, movie segments, news or a portal web page for booking a ticket. If the user touches his/her mobile phone with an RFID reader on the RFID in the poster, he/she receives a list of the candidate services from the network. Then the user can pick up the desired information service by operating the mobile phone.

II.2.2 System architecture

The architecture in Figure II.2 implements this scenario.

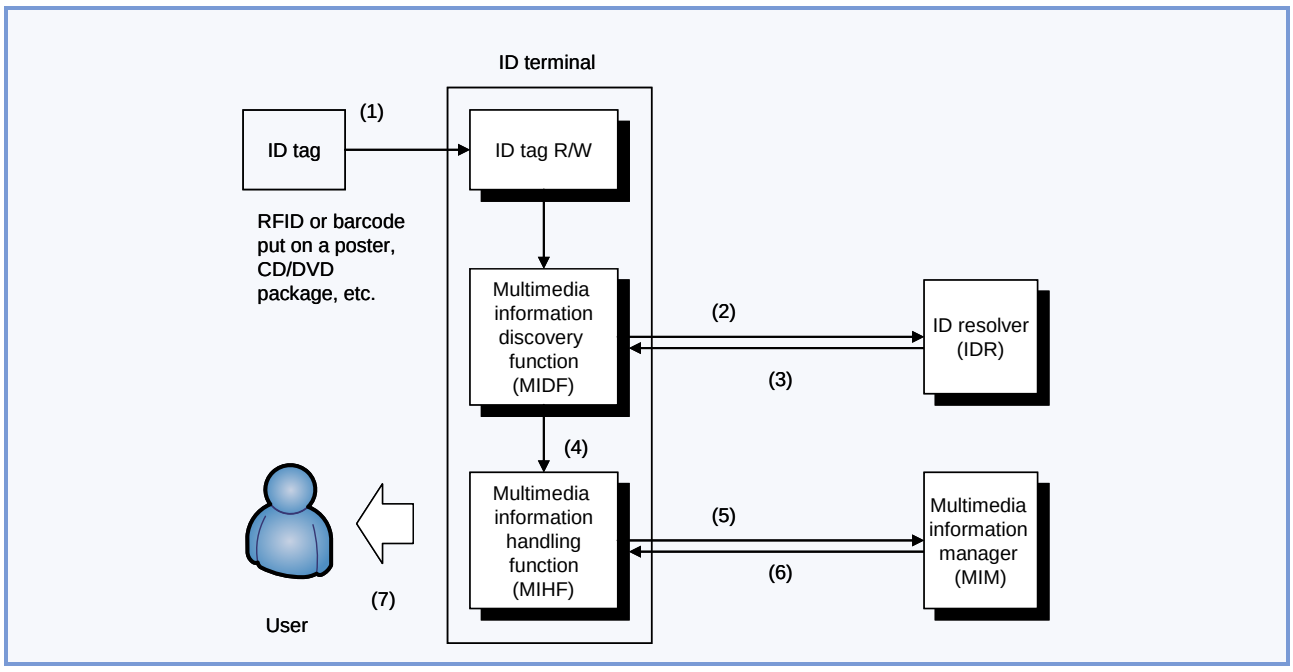


Figure II.2 – Implementation architecture and workflow of digital content delivery services using posters

II.2.3 Workflow

1) ID tag R/W obtains the identifier in the poster or CD/DVD package from RFID or barcode.
2) MIDF sends the identifier to the IDR to find the pointer and transfer protocol information of the associated MIMs.
3) The pointer and transfer protocol information on MIMs is provided to the MIDF.
4) MIDF sends the information, which includes the service identifier and the protocol of the MIM-*i* to the MIHF.
5) The MIHF calls the multimedia information delivery service of the MIM-*i* containing the detailed information associated with the ID code.
6) The multimedia information associated with the ID code is delivered to the MIHF in the ID terminal.
7) User watches the multimedia information via the ID terminal.

II.3 u-Museum

II.3.1 Application scenario

u-Museum (ubiquitous museum) provides a multimedia information service for visitors, such as guidance for exhibited art pieces, navigation in the gallery, and advertisement information for museum shops. This service is implemented by RFID tags, active infrared tags, mobile terminals with an RFID reader and infrared receiver, multimedia database of exhibits, wired/wireless networks, and so on. In the u-Museum, an active infrared tag is put at the entrance gate of an exhibition room, and sends the identifier of the room. When a visitor with a mobile terminal walks through the gate, the terminal receives the identifier, retrieves the information of the exhibition in this room, and shows the information to the visitor. The exhibition room shows several pieces of fine art, and a tiny RFID tag is embedded in the explanation plate of each exhibit. The user can get precise information on the exhibits by touching the mobile terminal on the plate. When the visitor wants to go to the next exhibit, the system navigates the route according to the art tour route. If the visitor takes a wrong turn, the ID terminal receive an unexpected location identifier from an infrared tag. Then the ID terminal gives a warning to the visitor.

II.3.2 System architecture

The architecture in Figure II.3 implements this scenario.

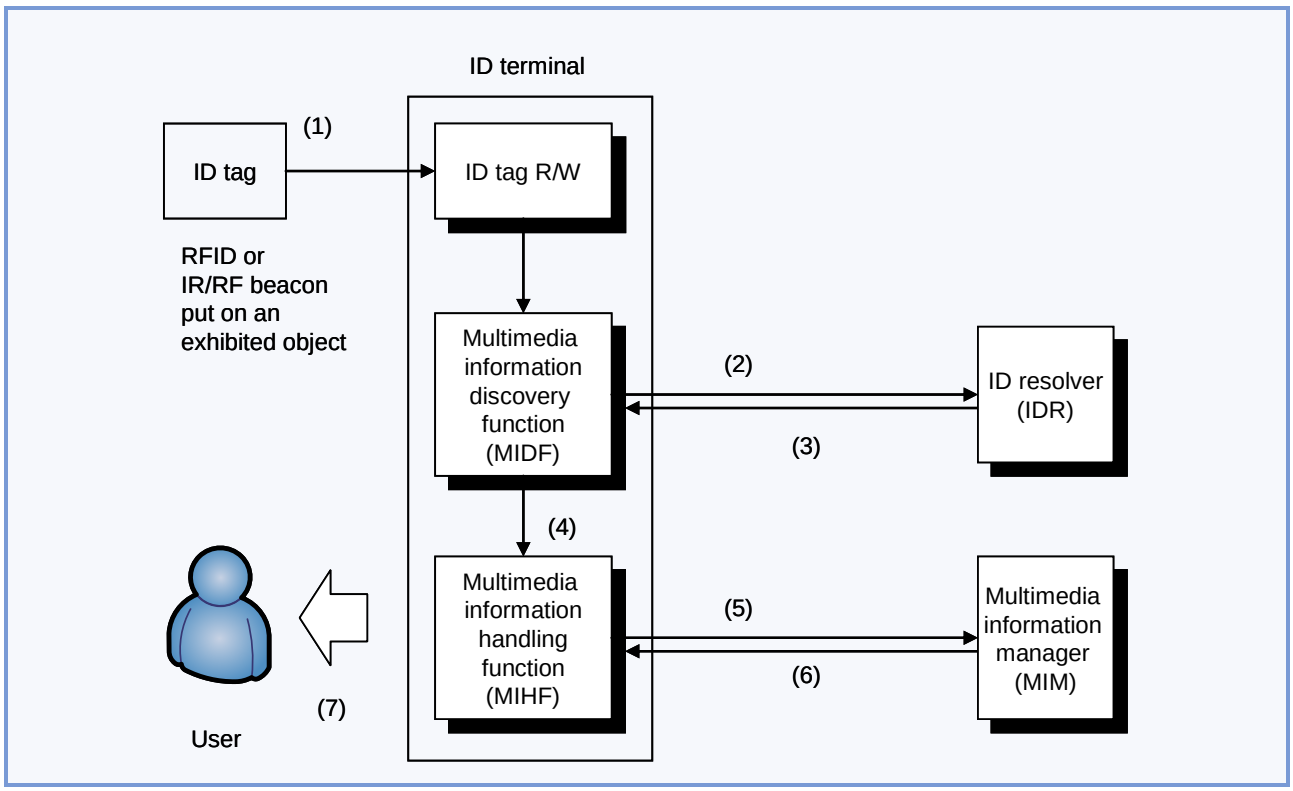


Figure II.3 – Implementation architecture and workflow of u-Museum services

II.3.3 Workflow

1) ID tag R/W obtains the identifier of the exhibited item in a gallery of u-Museum from the RFID or IR/RF beacon.
2) MIDF sends the identifier to the IDR to find the pointer and transfer protocol information of the associated MIMs.
3) The pointer and transfer protocol information of the MIMs is provided to the MIDF.
4) MIDF sends the information to the MIHF.
5) MIHF sends a multimedia information delivery service request to the MIMs.
6) Multimedia information associated with the identifier is delivered to the MIHF in the ID terminal.
7) User watches the multimedia information displayed by ID terminal.

II.4 Business card with personal identifier

II.4.1 Application scenario

Suppose that an identifier of a businessman is written on a business card. The identifier is associated to the latest contact address data record, including telephone number, fax number and e-mail address. His/her business client could get all the latest information from this identifier even after he/she has moved to another office or company.

II.4.2 System architecture

The architecture in Figure II.4 implements this scenario.

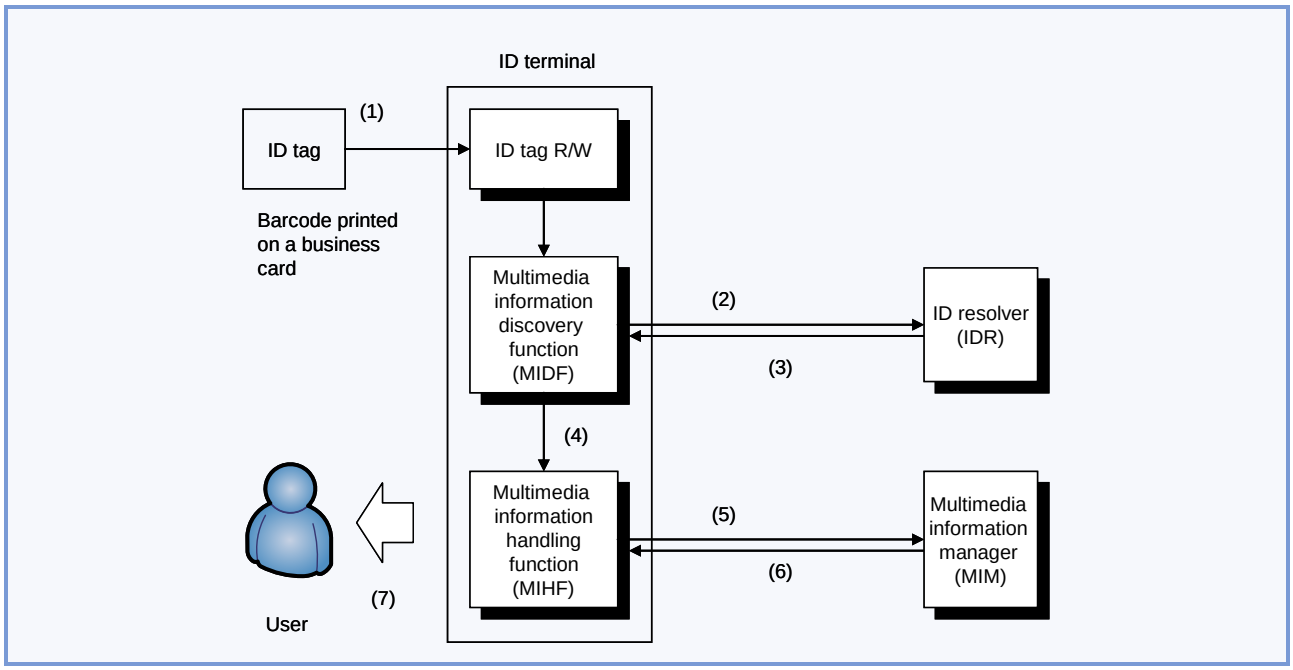


Figure II.4 – Implementation architecture and workflow of business card services

II.4.3 Workflow

1) ID tag R/W obtains the identifier of a person from the barcode printed on a business card (personal identifier).
2) MIDF sends the personal identifier to the IDR to find an associated personal information service.
3) Pointer and transfer protocol information of the MIMs are provided to MIDF.
4) MIDF sends the information to the MIHF.
5) MIHF requests the personal information delivery service of the MIMs with the personal identifier. The MIHF also sends the login name and password information for user authentication.
6) The personal information is delivered to the ID terminal according to the authentication result. If the user is authenticated as a valid business partner, the server will provide full contact information. If not, it will only provide an e-mail address.
7) User receives the personal information displayed by the ID terminal.

II.5 Presence service with multimedia information

II.5.1 Application scenario

Imagine a theatre in which every visitor has a ticket with RFID, and every seat in the theatre contains an RFID reader. When the visitor enters the theatre and takes a seat, he/she puts the ticket on the RFID reader located in the arm of the seat. The reader reads the visitor identifier and automatically notifies the theatre office of the visitor status through the theatre management application.

II.5.2 System architecture

In this scenario, the configuration of the system looks somewhat different from that of the other scenarios (Figure II.5). However, this architecture is a simple variation of the general system architecture in Figure 1. ID terminal consists of two physical components: ID tag R/W and personal computer equipped with a presence management application. In this case, a visitor's ticket is the real-world entity identified by the ID tag, and the theatre manager in the theatre office is the user of the multimedia information access. The theatre management server manages the presence information in the theatre, and delivers, for example, theatre seat status information in 2D map format to the ID terminal.

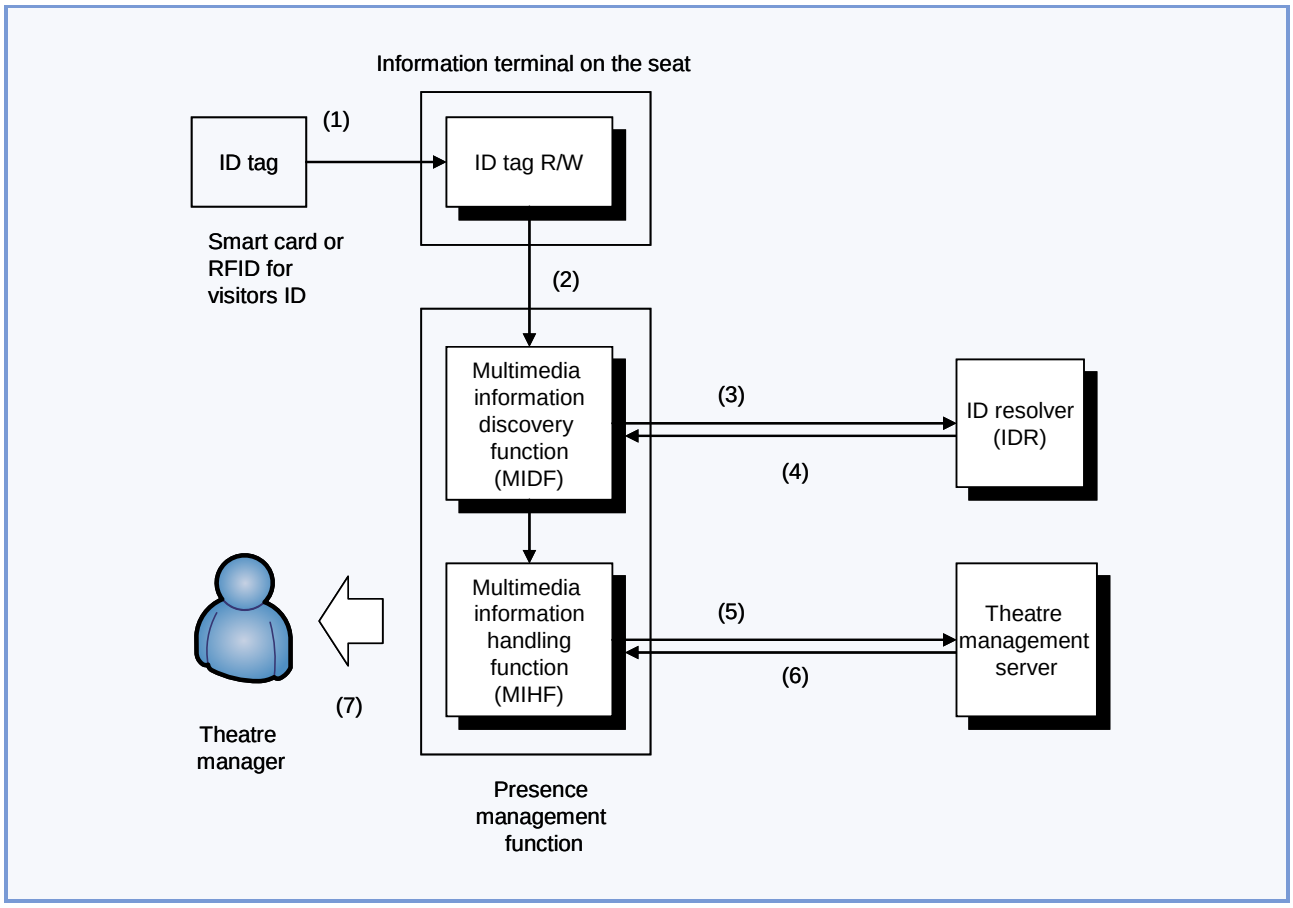


Figure II.5 – Implementation architecture and workflow of the presence service in theatres

II.5.3 Workflow

1) ID tag R/W of the theatre seat obtains the visitor identifier from the visitor's ticket.
2) ID tag R/W sends the visitor identifier to the MIDF in the presence management application.
3) MIDF sends the visitor identifier to the IDR to find the theatre management servers associated with the visitor identifier.
4) IDR informs the pointer and transfer protocol of the theatre management servers to the MIDF.
5) MIDF sends the information to the MIHF, which, in turn, requests the theatre management servers to obtain the presence information.
6) The server replies with the presence information.
7) Multimedia information browser updates the presence information according to the received information, and shows it to the theatre manager.

II.6 Food safety check and purchase

II.6.1 Application scenario

This application scenario is associated with the use case scenario of I.4, "Food traceability", in Appendix I of [ITU-T F.771].

A consumer wants to check with his/her smartphone the origin of production, the route of supply and the current status of quality for a food item before the purchase. The food item has been attached a UHF-type RFID tag. The smartphone has a dual-band RFID reader/writer to communicate with both HF and UHF ID tags, where the HF communication feature of the dual band reader/writer is provided by the NFC technology.

The user instructs the smartphone to do a food safety check and gets a response containing safety and other information from the food trace information server. If the user is satisfied with the information, he/she proceeds to buy the food item.

The user enables a payment feature of the smartphone and approaches the smartphone to a payment terminal with an HF-type (NFC-type) ID R/W, for the payment. A payment receipt is returned to the user after the payment succeeds.

II.6.2 System architecture

In this scenario, the configuration of the system is illustrated with the following representation relationships:

– Smartphone: ID terminal
– UHF ID tag: UHF-type RFID tag working as an ID tag
– HF ID tag: HF-type RFID tag working as an ID tag by NFC
– ID tag R/W a): UHF-type RFID R/W
– ID tag R/W b): HF-type RFID R/W

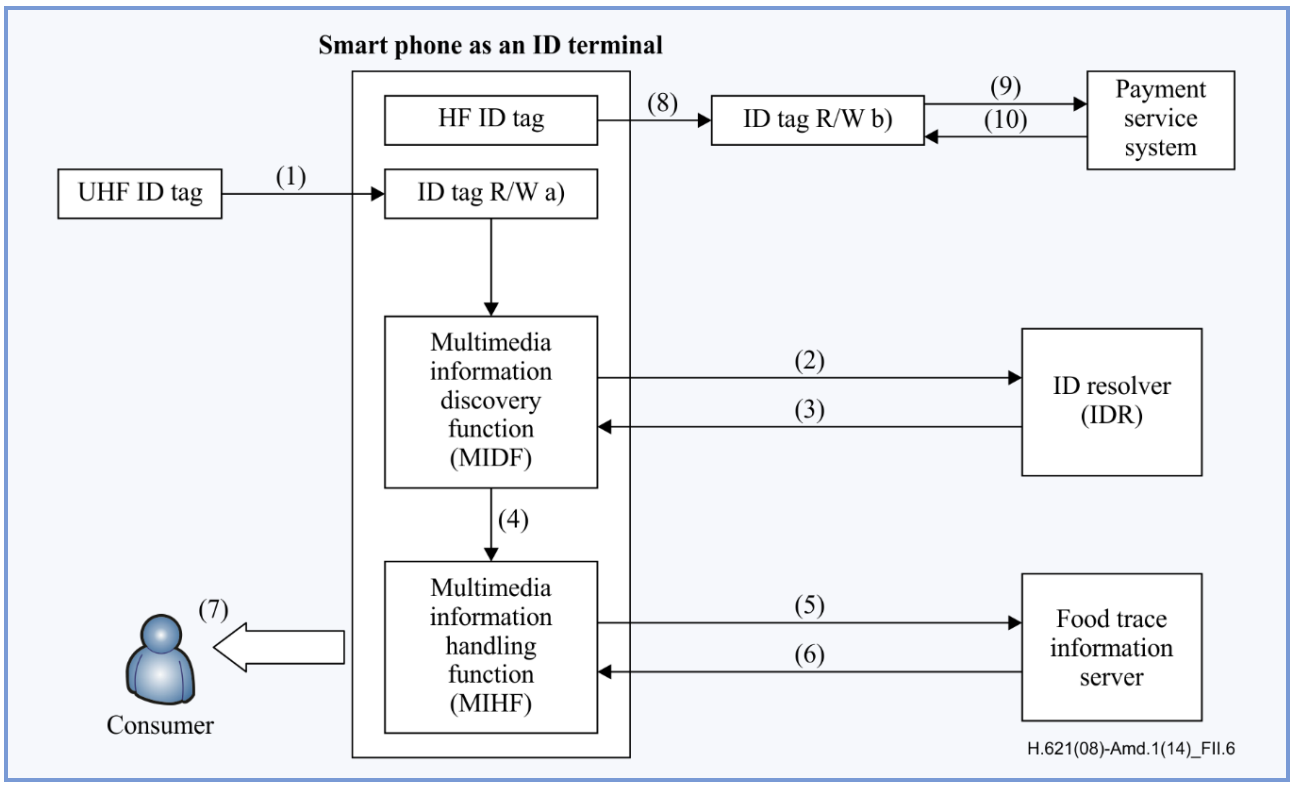


Figure II.6 – Implementation architecture and workflow of the food safety check and purchase case

The dual band RFID reader in the ID terminal can work sometimes as a UHF RFID reader, "ID tag R/W a)", sometimes as an NFC reader and sometimes as an NFC tag, "HF ID tag". The "ID tag R/W b)" and the "Payment Service System" correspond to the conventional smart card payment service.

II.6.3 Workflow

1) ID tag R/W a) in the smartphone reads the food identifier via UHF.

2) MIDF sends the food identifier to the IDR to find the food trace information server associated with food identifier.

3) IDR informs the location and transfer protocol of the food trace information server to the MIDF.

4) MIDF sends the location and transfer protocol information to the MIHF.

5) MIHF requests the food trace information server to obtain the food safety information.

6) The food trace information server replies with the food safety information.

7) The consumer reads the food safety information.

8) In case of buying the food, the consumer enables the payment feature of his/her smart phone and puts his/her smart phone, working as an HF ID tag herein, to ID tag R/W b).

9) ID tag R/W b) sends payment information consisting of a credit card number, for example, to the payment service system.

10) The payment service system sends a payment operation result, i.e. success or failure, back to the ID tag R/W b) which, then, prints out a proper notice for receipt or failure.

II.7 Visitor identification and guidance service with multimedia information

II.7.1 Application scenario

In this scenario, ID terminal is a kind of smartphone which has four components of interest: a UHF ID tag R/W, an HF ID tag, an HF ID tag R/W and a human presence management application. In this scenario, a visitor is given a visitor ID tag and the visitor reads that tag by the HF ID tag R/W and writes that information to the HF ID tag in his/her smartphone. The visitor is guided by UHF ID tags and the presence management system to the final destination in the building. The security building has an HF ID reader at the entrance gate and UHF ID tags are installed on the walls of corridors. The HF ID reader identifies a visitor and the UHF ID R/W of the smartphone reads the UHF ID tags on walls, and the smartphone displays the direction and route to the final destination. Therefore, a visitor who has the smartphone described above can enter the security building by obtaining the admission credentials by using the HF ID tag, and will be guided to the final destination in the building by using the UHF ID tag R/W.

II.7.2 System architecture

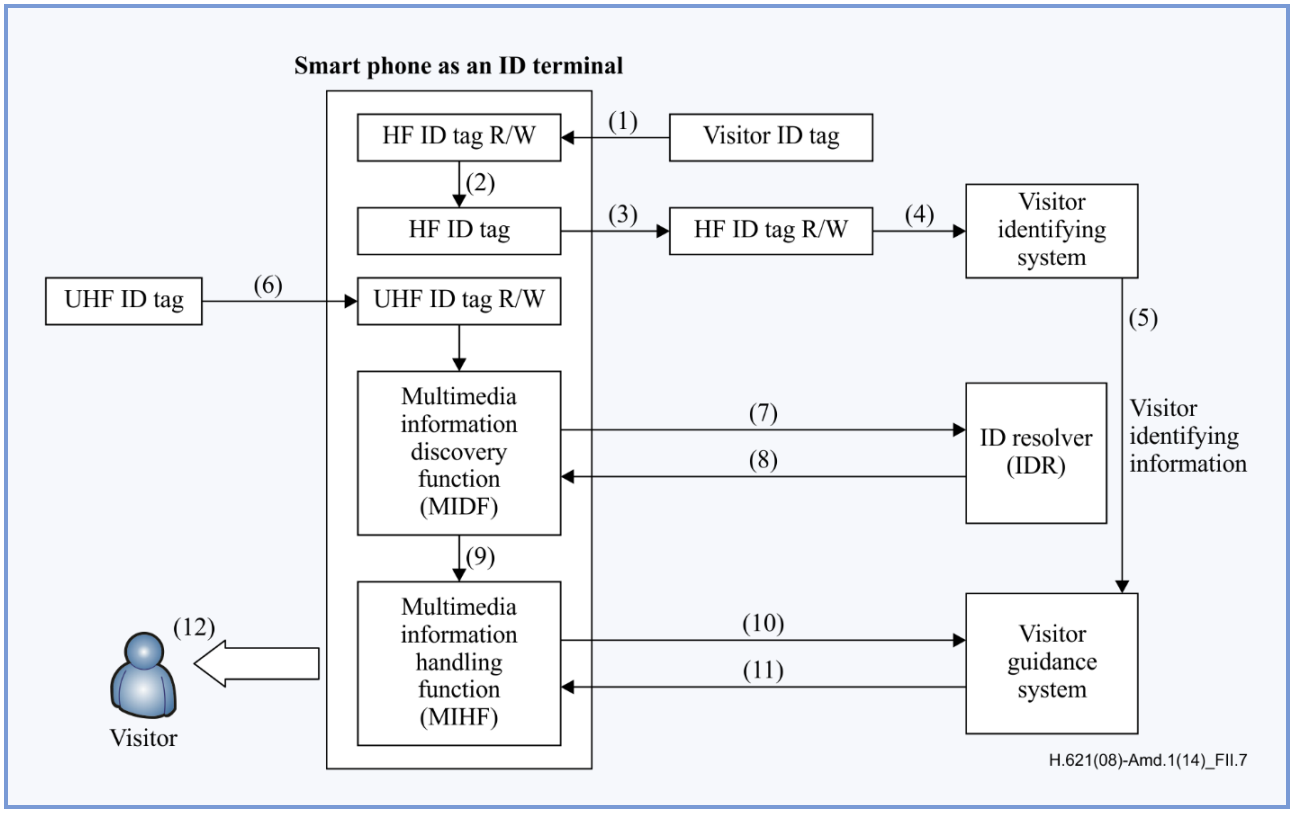


Figure II.7 – Implementation architecture and workflow of a visitor identification and guidance service with multimedia information

II.7.3 Workflow

1) Visitor reads visitor card (HF ID Tag) which is given by a guide of a building by using HF ID tag R/W of his/her smartphone.

2) HF ID tag R/W writes that information to HF ID tag of his/her smartphone.

3) HF ID tag R/W of the visitor identification system reads the HF ID tag which is located in the smartphone of the visitor and obtains the visitor identifier information from that.

4) Identifier of the visitor is sent to the visitor identification system.

5) Visitor identification system sends that information to the visitor guidance system.

6) UHF ID tag reader reads a UHF ID tag which is installed on the wall of corridor.

7) MIDF sends the ID to the IDR.

8) IDR informs the direction and route instructions of the visitor guidance server to the MIDF.

9) MIDF sends the information to the MIHF, which, in turn, requests the visitor management servers to obtain the presence information.

10) MIHF sends presence information to the visitor guidance system.

11) The visitor guidance system updates the presence information according to the received information, and calculates the route to the final destination.

12) Smart phone of the visitor shows the current location and route to the final destination on the screen of the smart phone.

13) The visitor can open the door of the final destination by using the HF ID tag in his/her smartphone.

Bibliography

[b-ISO/IEC 18000-x] ISO/IEC 18000-x-series (2008), *Information technology – Radio frequency identification for item management*.
<http://www.iec.ch/searchpub/cur_fut.htm>

[b-ISO/IEC 18000-4] ISO/IEC 18000-4 (2008), *Information technology – Radio frequency identification for item management – Part 4: Parameters for air interface communications at 2,45 GHz*.
<http://webstore.iec.ch/webstore/webstore.nsf/artnum/041837>

[b-ISO/IEC 14443-x] ISO/IEC 14443-x-series (in force), *Identification cards – Contactless integrated circuit cards – Proximity cards*.
<http://webstore.iec.ch/webstore/webstore.nsf/artnum/039489>
<http://webstore.iec.ch/webstore/webstore.nsf/artnum/040281>
<http://webstore.iec.ch/webstore/webstore.nsf/artnum/040282>
<http://webstore.iec.ch/webstore/webstore.nsf/artnum/041675>

[b-ISO/IEC 7816-x] ISO/IEC 7816-x-series (in force), *Identification cards – Integrated circuit cards*.
<http://www.iec.ch/searchpub/cur_fut.htm>

Y.4406/Y.2016

Functional requirements and architecture of the NGN for applications and services using tag-based identification

Functional requirements and architecture of the NGN for applications and services using tag-based identification

Summary

Recommendation ITU-T Y.2016 includes functional requirements and architecture of the NGN for the support of applications and services using tag-based identification. This Recommendation is based on the capabilities defined in Recommendation ITU-T Y.2213.

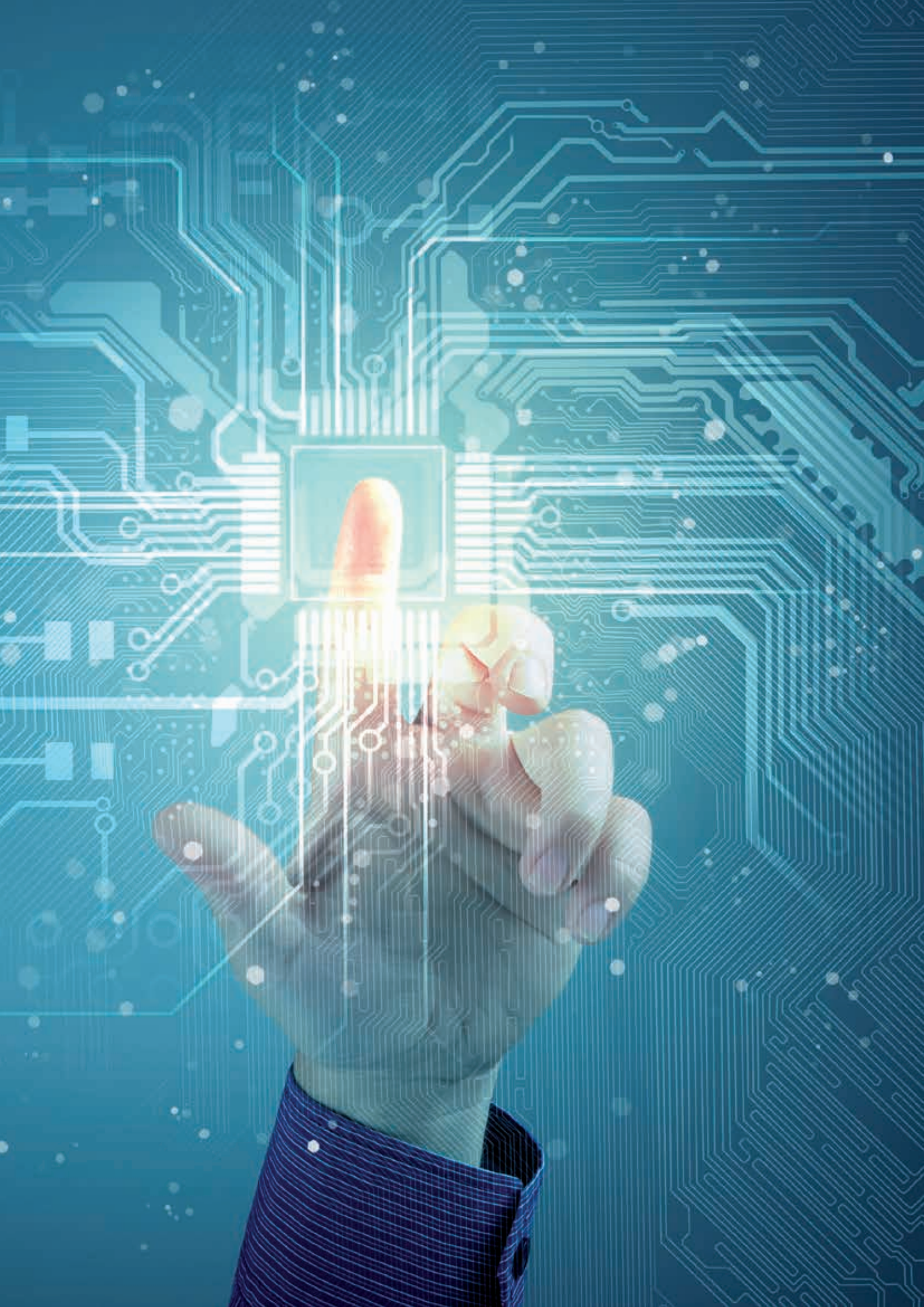
Source

Recommendation ITU-T Y.2016 was approved on 22 August 2009 by ITU-T Study Group 13 (2009-2012) under Recommendation ITU-T A.8 procedures.

Keywords

ID tag, ID terminal, frameworks, functional architecture, identifier, RFID, tag-based identification.

Table of Contents



Recommendation ITU-T Y.4406/Y.2016

Functional requirements and architecture of the NGN for applications and services using tag-based identification

1 Scope

This Recommendation based on [ITU-T Y.2012] covers extended features in order to support applications and services using tag-based identification in NGN.

This Recommendation describes functional requirements, functional architecture and functional entities in order to support the NGN service requirements and capabilities defined in [ITU-T Y.2213].

This Recommendation covers:

- Support of capabilities defined in [ITU-T Y.2213] from an architectural viewpoint;
- Functional requirements of the NGN architecture to support applications and services using tag-based identification;
- Functional architecture and entities extensions for applications and services using tag-based identification in NGN.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T X.668] Recommendation ITU-T X.668 (2008) | ISO/IEC 9834-9:2008, *Information technology – Open Systems Interconnection – Procedures for the operation of OSI Registration Authorities: Registration of object identifier arcs for applications and services using tag-based identification*.

[ITU-T Y.2012] Recommendation ITU-T Y.2012 (2006), *Functional requirements and architecture of the NGN release 1*.

[ITU-T Y.2213] Recommendation ITU-T Y.2213 (2008), *NGN service requirements and capabilities for network aspects of applications and services using tag-based identification.*

[ITU-T Y.2701] Recommendation ITU-T Y.2701 (2007), *Security requirements for NGN release 1*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 associated information [ITU-T Y.2213]: The information which is associated with an identifier.

NOTE – Example associated information instances are URL, URN, IP address, E.164 number, etc.

3.1.2 forward identifier resolution [ITU-T Y.2213]: A function to resolve an identifier into an associated information.

3.1.3 ID tag [ITU-T Y.2213]: A physical object which stores one or more identifiers and optionally application data such as name, title, price, address, etc.

NOTE – It may have a communication capability with an ID terminal depending on implementations.

3.1.4 ID terminal [ITU-T Y.2213]: A device with a data reading and optional writing capability which reads (and optionally writes) identifier(s) and optionally application data from/into an ID tag.

NOTE – The data reading (and optionally writing) capability depends on implementations.

3.1.5 identifier resolution [ITU-T Y.2213]: A function to resolve an identifier into associated information (see "Forward identifier resolution") and vice versa (see "Reverse identifier resolution").

3.1.6 reference point [ITU-T Y.2012]: A conceptual point at the conjunction of two non-overlapping functional entities that can be used to identify the type of information passing between these functional entities.

NOTE – A reference point may or may not correspond to one or more physical interfaces between pieces of equipment.

3.1.7 reverse identifier resolution (or backward identifier resolution) [ITU-T Y.2213]: A function to resolve an associated information into a corresponding identifier. It is the reverse operation of the forward identifier resolution.

3.1.8 tag-based identification [ITU-T Y.2213]: The process of specifically identifying a physical or logical object from other physical or logical objects by using identifiers stored on an ID tag.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 functional architecture: A set of functional entities used to describe the structure of an NGN. These functional entities are separated by reference points, and thus, they define the distribution of functions. The functional entities can be used to describe a set of reference configurations. These reference configurations identify which reference points are visible at the boundaries of equipment implementations and between administrative domains.

3.2.2 functional entity: An entity that comprises a specific set of functions at a given location. Functional entities are logical concepts, while groupings of functional entities are used to describe practical and physical implementations.

3.2.3 identifier: An identifier is a series of digits, characters and symbols or any other form of data used to identify subscriber(s), user(s), network element(s), function(s), network entity(ies) providing services/applications, or other entities (e.g., physical or logical objects). Identifiers can be used for registration or authorization.

NOTE – Identifier can be either public to all networks, shared between a limited number of networks or private to a specific network (private identifiers are normally not disclosed to third parties).

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations:

ABG-FE Access Border Gateway Functional Entity

AN-FE Access Node Functional Entity

ANI Application Network Interface

APL-SCM-FE	Application Service Coordination Manager Functional Entity
AS-FE	Application Support Functional Entity
B2B	Business-to-Business
B2C	Business-to-Customer
DB	Data Base
DNS	Domain Name System
EN-FE	Edge Node Functional Entity
FE	Functional Entity
IBG-FE	Interconnection Border Gateway Functional Entity
ID	Identification
IP	Internet Protocol
IRI	Internationalized Resource Identifier
LDAP	Lightweight Directory Access Protocol
NACF	Network Attachment Control Functions
NGN	Next Generation Network
NNI	Network Node Interface
OID	Object Identifier
PD-FE	Policy Decision Functional Entity
PII	Personally Identifiable Information
RACF	Resource and Admission Control Functions
RFID	Radio Frequency Identification
SAA-FE	Service Authentication and Authorization Functional Entity
SUP-FE	Service User Profile Functional Entity
TRC-FE	Transport Resource Control Functional Entity
TTI	Tag to Terminal Interface
UNI	User Network Interface
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
URN	Uniform Resource Name
XNI	Any Network Interface

5 Conventions

None.

6 Functional requirements and functions of the NGN for applications and services using tag-based identification

6.1 NGN functional requirements

[ITU-T Y.2012] describes the NGN architectural functions to support NGN Release 1 services. In order to support applications and services using tag-based identification, NGN functions are required to be extended.

Clause 8.1 of [ITU-T Y.2213] requires that the following capabilities be supported by extension of the existing capabilities of NGN:

- Multi-identifier interpretation;
- Identifier resolution;
- Privacy management;
- Content distribution control;
- Device management;
- Profile management:
  - User profile;
  - Device profile.
- Quality of Service.

In addition, the following capabilities are supported by the existing capabilities of NGN (see clause 8.2 of [ITU-T Y.2213]):

- Service quality control;
- Location management.

Appendix I of [ITU-T Y.2213] also identifies non-NGN high-level service requirements which will not affect the existing functional capabilities of the NGN.

Based upon the analysis of [ITU-T Y.2213] service requirements and network capabilities (see Appendix I for further details), the following list can be supported by NGN functions and end-user functions.

- General requirements for identifiers:
  - Identifiers validation supported by NGN functions during identifier resolution.
- Identification of identifier schemes:
  - Supported by end-user functions in ID terminal.
- Application data encoding:
  - Encoded application data must be decoded in the ID terminal. Therefore, end-user functions in the ID terminal support application data decoding. The definition of this feature is outside the scope of this Recommendation.
- Identification service interworking:
  - Supported by NGN functions.
- Location information management:
  - Supported by NGN functions.
- Management of application mobility:
  - Supported by NGN functions.

- Traceability:
  - Supported by NGN functions. In many applications, traceability will affect PII and therefore these functions are required to be compliant with the legislation and regulation relevant to privacy and protection of PII.
- Identifier filtering:
  - Supported by end-user functions in the ID terminal.

6.2 Functional architecture model

NOTE – High-level reference architecture of ID-based application and services is given in Appendix II.

The capabilities for the support of applications and services using tag-based identification are supported by transport stratum functions, service stratum functions, end-user functions and management functions.

Figure 1 shows the overall functional architecture model of NGN to support applications and services using tag-based identification. More specifically, the figure shows the functions required for the support of applications and services using tag-based identification. Appendix III describes a use-case example based on functional architecture model.

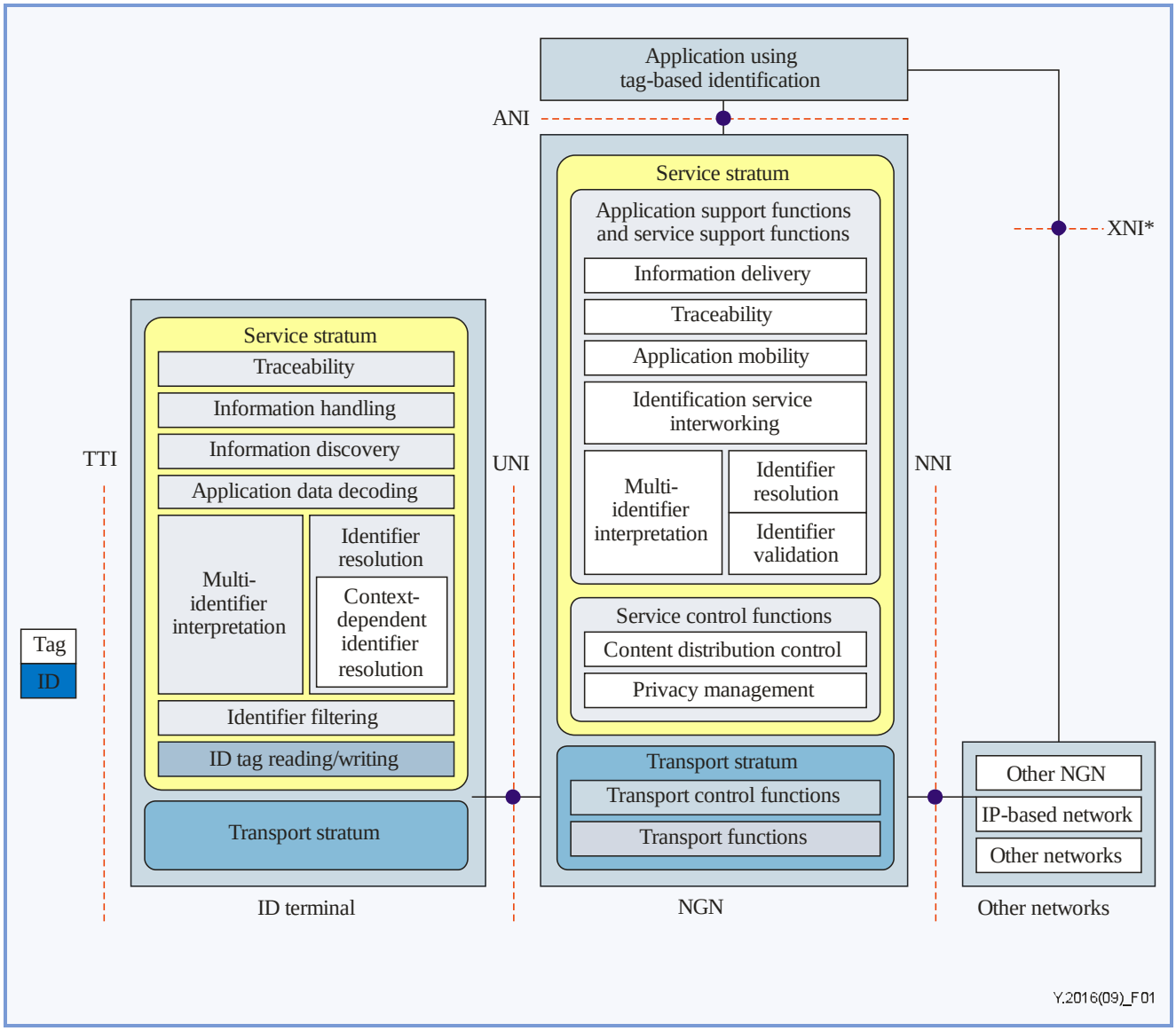


Figure 1 – Overall functional architecture model

6.3 Functions to support applications and services using tag-based identification

This clause describes how functional requirements identified in clause 6.1 are supported in the functional architecture model of clause 6.2. More specifically, this clause identifies how [ITU-T Y.2012] functions need to be extended to support applications and services using tag-based identification.

6.3.1 Transport stratum functions

Transport stratum functions are required to be extended to sustain the transaction volume caused by tag-based identification applications and services.

6.3.1.1 Transport functions

There are no extended capabilities on transport functions to support applications and services using tag-based identification. Access network functions, edge functions and core transport functions [ITU-T Y.2012] support different QoSs, according to the service quality requirements.

6.3.1.2 Transport control functions

The following functional requirements are supported by transport control functions [ITU-T Y.2012]:

- Service quality control:

  Resource and admission control functions (RACF) provide different QoS, according to service quality requirements.

- Location management:

  Network attachment control functions (NACF) support location information management of ID terminal and/or ID tag.

In addition, transport control functions are needed to be extended to support the following capability:

- Quality of Service (Application transaction and traffic requirements):

  Data and signalling traffic caused by tag-based identification applications and services will be increased and access concentration to single resources (e.g., identifier resolution) may happen. Resource and admission control functions (RACF) are required to provide the capability to sustain transaction volume caused by tag-based identification applications and services. In addition, resource and admission control functions (RACF) are recommended to provide traffic distribution using traffic overload control mechanisms.

6.3.2 Service stratum functions

Service stratum functions are required to be extended to support requirements of applications and services using tag-based identification. Extensions of service stratum functions are given in the following subclauses.

6.3.2.1 Service control functions

- Privacy management handling:

  The service user profile is recommended to contain the privacy management profile as a part of the whole profile information. Therefore, service control functions are recommended to provide the capability to handle the privacy management.

- Content distribution control:

  [ITU-T Y.2213] indicates that facilities for control of information content distribution are recommended to be supported (to accommodate possible commercial, regulatory and privacy requirements). Service control functions are recommended to provide content distribution control functions on the basis of the service user profile.

6.3.2.2 Application support functions and service support functions

Application support functions and service support functions are extended with the following functions:

- Multi-identifier interpretation:

  Multi-identifier interpretation is provided by using OID-based identifier interpretation. To support the OID-based identifier interpretation, NGN is required to support OID-based identification scheme, such as [ITU-T X.668], and can optionally provide structure information of identifier schemes based on OID. Application support functions and service support functions are required to support multi-identifier interpretation and can optionally provide structure information of identifier schemes to end-user terminal.

- Identifier resolution:

  Identifier resolution is a procedure that allows finding associated information resources with a given identifier. The application support functions and service support functions are recommended to provide identifier resolution function.

- Identifier validation:

  Lifetime requirement of identifier depends on applications and services. Identifiers are validated according to identifier lifetime. The application support functions and service support functions can optionally provide identifier validation function. This identifier validation capability can be provided during identifier resolution process.

- Identification service interworking:

  The application support functions and service support functions can optionally support identification service interworking which allows interworking of different tag-based identification applications and services.

- Application mobility:

  A communication association can be handed over to other applications by application mobility. A typical example is given for a transportation application. In route from A to C via B, the association of a fare application between A and B has to be handed over to the other fare application between B and C in order to support the single fare association. Application mobility can be optionally provided among different tag-based identification applications and services by the application support functions and service support functions.

- Traceability:

  The application support functions and service support functions support traceability. Traceability provides ID terminal's history of reading an ID tag for an object or tag-based identification applications and services' history of reading an ID tag for an object. Any distribution of information in support of traceability shall meet national legal and regulatory requirements on data privacy and protection.

  NOTE – Traceability mechanism and referential information flows are given in Appendix IV.

- Information delivery:

  The information delivery functions receive content from the application functions, store, process, and deliver it to the end-user functions using the capabilities of the transport functions, under control of the service control functions.

6.3.3 End-user functions

End-user functions are required to support requirements of applications and services using tag-based identification. The following functions are provided by end-user functions:

- Multi-identifier interpretation:

  End-user functions can optionally support multi-identifier interpretation using OID-based identifier interpretation. Application support functions can optionally provide structure information of identifier schemes based on OID to end-user functions. End-user functions use this structure information to support multi-identifier interpretation.

- Identifier resolution:

  End-user functions are required to support identifier resolution. End-user functions send identifier resolution request to NGN and application support functions and service support functions in order to perform identifier resolution.

- Context-dependent identifier resolution:

  End-user functions can optionally support usage context of the identifier capability for context-dependent identifier resolution under one-to-many associations between an identifier and associated information instances. End-user application functions can send usage context of the identifier to the appropriate identifier resolution serving function in the NGN.

- Application data decoding:

  Identifier information may be encoded optionally with other application data like title, name, price, etc., into an ID tag in a standardized way. Therefore, end-user functions support application data decoding in a standardized way. The definition of this feature is outside the scope of this Recommendation.

- Traceability:

  When ID terminal reads ID tag, the ID terminal is required to send location information to NGN, with location information and identifier gained from ID tag. End-user functions are required to support the sending of the location information and usage information of identifiers to the NGN. Any distribution of information in support of traceability shall meet national legal and regulatory requirements on data privacy and protection.

  NOTE – Traceability mechanism and referential information flows are given in Appendix IV.

- ID tag reading and writing:

  These functions provide communication interfaces to ID tag. These functions include reading and writing of a single or multiple identifiers as well as application data from and into the ID tag. The information resulting from ID tag reading is sent to the identifier resolution function for further processing. ID terminal can optionally contain multiple ID tag reading and/or writing functions.

- Identifier filtering:

  End-user functions support identifier filtering in order not to process unsolicited ID tags or identifier schemes.

- Information discovery:

  This function obtains identifier from ID tag, and issues queries to identifier resolution functions in NGN. It uses the identifier as a query key in both cases. The identifier resolution function returns pointer information (e.g., URL) for the information delivery services. After obtaining the pointer information, the information discovery function sends the information to the information handling function.

- Information handling:

  This function provides a capability to download relevant information from information delivery services, and to present the downloaded information to the user.

6.3.4 Management functions

Management functions are required to be extended to support requirements of applications and services using tag-based identification. Extensions of management functions are as follows:

- Device management and device profile management (ID terminal and ID tag management):

  Applications using tag-based identification require a number of ID terminals and ID tags to capture identifier and relevant information from ID tags. ID terminals and ID tags consist of a number of technical features in aspects of radio operations, network operations, software upgrade, time synchronization, device identifier, identifier structure information, and filtering rules which need monitoring and maintenance. Management functions are recommended to support ID terminal and ID tag management functions with device profile management.

7 Functional architecture of the NGN for applications and services using tag-based identification

Functional entities defined in [ITU-T Y.2012] need to include additional functions in order to be used for the support of applications and services using tag-based identification.

Based on the functions provided in clause 6.3, this clause identifies potential extensions required to the functional entities of the generalized NGN functional architecture shown in Figure 3 of [ITU-T Y.2012].

NOTE – This Recommendation uses the naming conventions of NGN FEs as defined in [ITU-T Y.2012].

7.1 Transport processing functional entities

7.1.1 T-2 Access node functional entity (AN-FE)

- Service quality control:

  The AN-FE supports different QoSs, according to the service quality requirements.

7.1.2 T-3 Edge node functional entity (EN-FE)

- Service quality control:

  The EN-FE supports different QoSs, according to the service quality requirements.

7.1.3 T-5 Access border gateway functional entity (ABG-FE)

- Service quality control:

  The ABG-FE supports different QoSs, according to the service quality requirements.

7.1.4 T-6 Interconnection border gateway functional entity (IBG-FE)

- Service quality control:

  The IBG-FE supports different QoSs, according to the service quality requirements.

7.2 Transport control functional entities

7.2.1 T-16 Policy decision functional entity (PD-FE)

- Quality of Service (Application transaction and traffic requirements):

  The PD-FE is required to support QoS capabilities to sustain the transaction volume caused by tag-based identification applications and services. Also, the PD-FE is recommended to support QoS capabilities to be able to avoid access concentration to single resources (e.g., identifier resolution).

7.2.2 T-17 Transport resource control functional entity (TRC-FE)

- Quality of Service (Application transaction and traffic requirements):

  The TRC-FE is required to handle transaction volume of applications using tag-based identification and to provide access distribution to single resource.

7.3 Service control functional entities

7.3.1 S-5 Service user profile functional entity (SUP-FE)

- Privacy management:

  The SUP-FE is recommended to support privacy management.

- Content distribution control:

  The SUP-FE is recommended to provide information for content distribution control using the service user profile.

7.3.2 S-6 Service authentication and authorization functional entity (SAA-FE)

- Privacy management:

  The SAA-FE is recommended to provide privacy management when consulting the SUP-FE.

- Content distribution control:

  The SAA-FE is recommended to provide information for content distribution control.

7.4 Application support functions and service support functions

7.4.1 A-1 Application support functional entity (AS-FE)

The following text does not mandate that the same instance of AS-FE has to support the functions listed in this clause but allows for the case where these functions are supported by different instances of the AS-FE (e.g., identifier resolution and traceability can be supported in two different instances of AS-FE).

- Multi-identifier interpretation:

  The AS-FE is required to support the OID-based identification scheme specified by [ITU-T X.668]. Also, the AS-FE can optionally provide structure information of identifier schemes to the end-user terminal.

- Identifier resolution:

  The AS-FE is recommended to provide identifier resolution function. Identifier resolution function may be a part of AS-FE or an external entity of the NGN. In case of external entity, the AS-FE interacts with the external entity, via the ANI, to perform the identifier resolution process.

- Context-dependent identifier resolution:

  The AS-FE can optionally support context-dependent identifier resolution under one-to-many associations between an identifier and associated information instances using usage context of the identifier received from the end-user application.

- Identifier validation:

  The AS-FE can optionally provide identifier validation capability that can be performed during the identifier resolution process.

- Traceability:

  The AS-FE can optionally support traceability. In applications and services using tag-based identification, service providers maintain specific systems for traceability like database system. The AS-FE can optionally support traceability functions in cooperation with these specific systems. In case that this database system is maintained in NGN, AS-FE is required to support intercommunicating with end-user functions and receive location information and usage information about the identifier.

- Information delivery:

  The AS-FE supports information delivery. The AS-FE delivers contents to the end-user functions using the capabilities of the transport functions.

7.4.2 A-3 Application service coordination manager functional entity (APL-SCM-FE)

- Identification service interworking:

  The APL-SCM-FE can optionally support identification service interworking between B2C and B2B or some other cooperative business relationships as well as different application and service infrastructures using tag-based identification.

- Application mobility:

  The APL-SCM-FE can optionally support application mobility among different tag-based identification applications and services.

8 Security

Security considerations regarding the functional requirements and architecture of the NGN are addressed in [ITU-T Y.2701].

Appendix I

Analysis of service requirements and network capabilities defined in [ITU-T Y.2213]

(This appendix does not form an integral part of this Recommendation)

I.1 NGN service requirements of tag-based identification applications and services

[ITU-T Y.2213] defines the following NGN service requirements for applications and services using tag-based identification.

Requirement	Explanation
Multi-identifier interpretation requirements	An identifier constituted by sub-identifier elements is required to be interpreted into sub-identifiers by its structure information. To interpret multi-identifier, identifier structure information may be provided to ID terminal.
Identifier resolution	Identifier resolution is a procedure to find information resources associated with the identifier. Association between identifier and information can be one-to-one or one-to-many. Different associated information for an identifier may be resolved according to usage context of the identifier, for example, who/when/where/why, which is called context-dependent identifier resolution.
ID terminal and ID tag management	The NGN may monitor and manage ID terminal and ID tag in aspects of radio operations, network operations, software upgrade, time synchronization, device identifier, identifier structure information, filtering rules update and location registration and management.
Content distribution control	It is recommended that facilities for control of information content distribution should be supported (to accommodate possible commercial, regulatory and privacy requirements).
Privacy management	There are many privacy threats in applications and services using tag-based identification. Therefore privacy protection must be prepared.
Location-based services support	Location information of ID terminal and/or ID tag is recommended to be registered (either statically or dynamically) and may be provided if requested by tag-based applications and services.
Service quality control	Some tag-based identification applications and services may have different service quality requirements. It is recommended to provide different service qualities according to service quality requirements.
Application transaction and traffic requirements	It is required to manage the transaction volume generated by tag-based identification applications and services, and to be able to avoid access concentration to single resources.

I.2 Non-NGN high level service requirements of tag-based identification applications and services

[ITU-T Y.2213] defines the following non-NGN high level service requirements of applications and services using tag-based identification.

Requirement	Explanation
General requirements for identifiers	Identifier satisfies the requirements such as uniqueness of identifier, lifetime, Identifier validation, registration or authorization of identifier, and sharing identifier between several applications/services.
Identification of identifier schemes	Identifiers assigned by a certain identifier scheme are required to be distinguishable from identifiers assigned by other identifier schemes, because NGN may handle various identifier schemes.
Application data encoding	Identifier information may be encoded optionally with other application data like title, name, price, etc., into an ID tag. Therefore, application data is required to be encoded in a standardized way on ID tags that are used by tag-based identification applications and services.
Identification service interworking	Various ID-based services and applications are required to be interworked. For example, B2C tag-based identification applications and services can be combined with B2B into B2B2C service model and barcode-based services and RFID-based services should be integrated.
Location information management	Applications may require ID terminal location information or ID tag location information, and will know who is to use this location information. User location, that is an ID terminal location, can be provided using GPS or cell information from cellular networks and so on. ID tag location may be included in the ID tag itself, or retrieved from a service provider via an identifier.
Management of application mobility	Application mobility is recommended to be provided among different tag-based identification applications and services, because an ID tag might be moved among them with a requirement that their communication associations should be retained.
Traceability	Traceability relates to object traceability and usage traceability. For some use cases, it is recommended to provide the information on what an ID terminal has read an ID tag for an object and to provide the information on what tag-based identification applications and services have read an ID tag for an object.
Identifier filtering	Users, applications modules, middleware functions, or lower-layer read functions do not have to process unsolicited ID tags or identifier schemes. Proper filtering is recommended to be provided.

I.3 NGN requirements supported by extensions or additions of NGN Release 1 capabilities

[ITU-T Y.2213] suggests that the following NGN capabilities are required to be extended or added. This clause allocates proper functions of NGN to satisfy the capabilities.

Requirement	Functions supporting requirement
Multi-identifier interpretation requirements	• End-user functions • Application support functions and service support functions
Identifier resolution	• Application support functions and service support functions • End-user functions
Device management (ID terminal and ID tag management in Appendix III)	• Management functions
Content distribution control	• Service control functions with service user profile
Privacy management	• Service control functions with service user profile
User profile (To satisfy privacy management)	• Service control functions
Device profile (To satisfy device management)	• Management functions
Quality of Service (Application transaction and traffic requirements in Appendix III)	• Resource and admission control functions (RACF)

I.4 NGN requirements supported by existing NGN Release 1 capabilities

[ITU-T Y.2213] suggests that the following NGN capabilities are supported by existing NGN Release 1 capabilities.

Requirement	Functions supporting requirement
Service quality control	• Access network functions • Edge functions • Core transport functions • Resource and admission control functions (RACF)
Location management	• Network attachment control functions (NACF)

I.5 Non-NGN high level service requirements supported by NGN Release 1 capabilities

[ITU-T Y.2213] defines the following non-NGN high level service requirements, but some requirements are needed to be supported by NGN functions.

Requirement	Functions supporting requirement
General requirements for identifiers – Identifier validation	• Application support functions and service support functions through identifier resolution
Identification of identifier schemes	• Not applicable
Application data encoding	• End-user functions – End-user functions shall support application data decoding, but this is not in the scope of this Recommendation.
Identification service interworking	• Application support functions and service support functions
Location information management	• Network attachment control functions (NACF)
Management of application mobility	• Application support functions and service support functions
Traceability	• Application support functions and service support functions • End-user functions
Identifier filtering	• End-user functions support identifier filtering

Appendix II

High-level reference architecture of ID-based applications and services in the NGN

(This appendix does not form an integral part of this Recommendation)

Figure II.1 depicts high-level reference architecture for the support of ID-based applications in the NGN.

User terminal (ID terminal equipment), service providers and the NGN domains are major components. ID terminal is a device that reads the ID tag and/or writes data to the ID tag. An ID terminal can be connected to the NGN via different types of access networks. ID-based services which are related to the identifier(s) stored in the ID tag are provided to end-users by service providers.

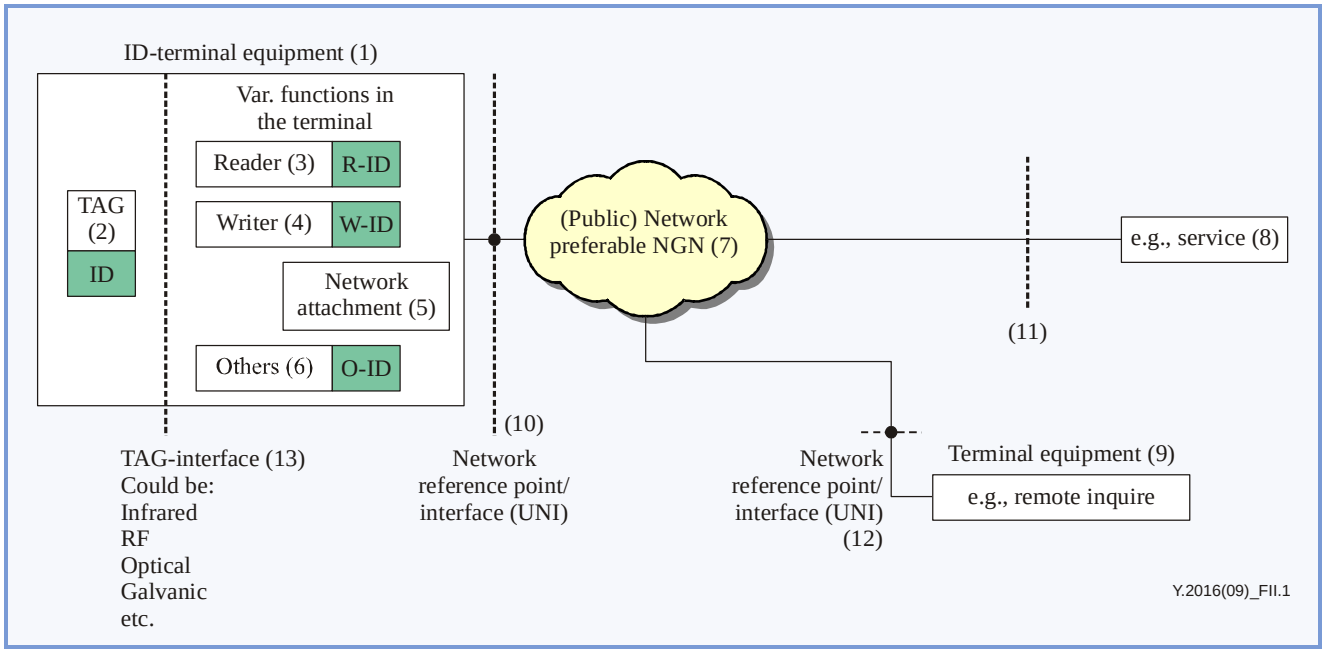


Figure II.1 – High-level reference architecture

Figure II.1 shows very generic and simplified high-level reference architecture. It illustrates the main objects, reference points as well as functional blocks for the support of ID-based applications and services in the NGN. In this figure, the "public network" corresponds to the NGN. Reference point (11) relates to service support and control functions for ID-based applications and services.

Refer to Appendix III of [ITU-T Y.2213] for example scenarios of tag-based identification applications and services, and more specifically to typical B2B and B2C network configuration models.

Appendix III

Use-case example of applications using tag-based identification in the NGN

(This appendix does not form an integral part of this Recommendation)

Typical ID-based applications and services can generally be provided with four steps as follows:

1) Reading identifier from ID tags: End-user reads identifier from ID tag with ID terminal (ID tag reading/writing functions).

2) Identifier resolution process: Identifier itself has no meaning for an end-user. Identifier resolution is a procedure to find the location of the information resources (service/content) which are associated with the identifier. The location of the information can be presented as URI, URL, IRI, etc. that provide how to access to and where to locate the final target content. Identifier resolution process is performed in the NGN. This means that some functional entities (FEs) are required to support identifier resolution capability. An NGN FE can perform identifier resolution process by itself or commit identifier resolution to external identifier resolution server. Identifier resolution protocol can be implemented with a directory service protocol such as X.500, LDAP or DNS.

3) Selection of content/service: The end-user obtains the result of identifier resolution request. The result will be in a form of a content/service list. Then the end-user chooses the preferred content/service from the list. Choosing the service in the ID terminal leads to finding the service provider for the service with its locator such as URI, URL or IRI.

4) Service delivery: After the end-user has chosen the service, the end-user is connected to the corresponding service provider through the NGN. The chosen service is delivered to the end-user by the service provider via the NGN.

Figure III.1 illustrates an example use-case where the end-user reads the identifier stored in ID tags attached to a movie poster. The end-user reads the identifier with his/her ID terminal, then the ID terminal sends an identifier resolution request to the NGN. The identifier resolution is processed by the NGN itself or with the aid of the NGN. The end-user obtains a service list including a trailer service as a result of the identifier resolution request. A trailer service is provided by a service provider. This example assumes the following:

- A movie poster has media for storing an identifier.
- A service provider creates a trailer service for that movie and associates that service with the identifier.
- The end user is provided an ID terminal (e.g., containing a RFID reading function, ID tag reading/writing functions).
- The reading identifier by the ID terminal involves an identifier resolution with which a user can acquire a service list associated with the identifier.

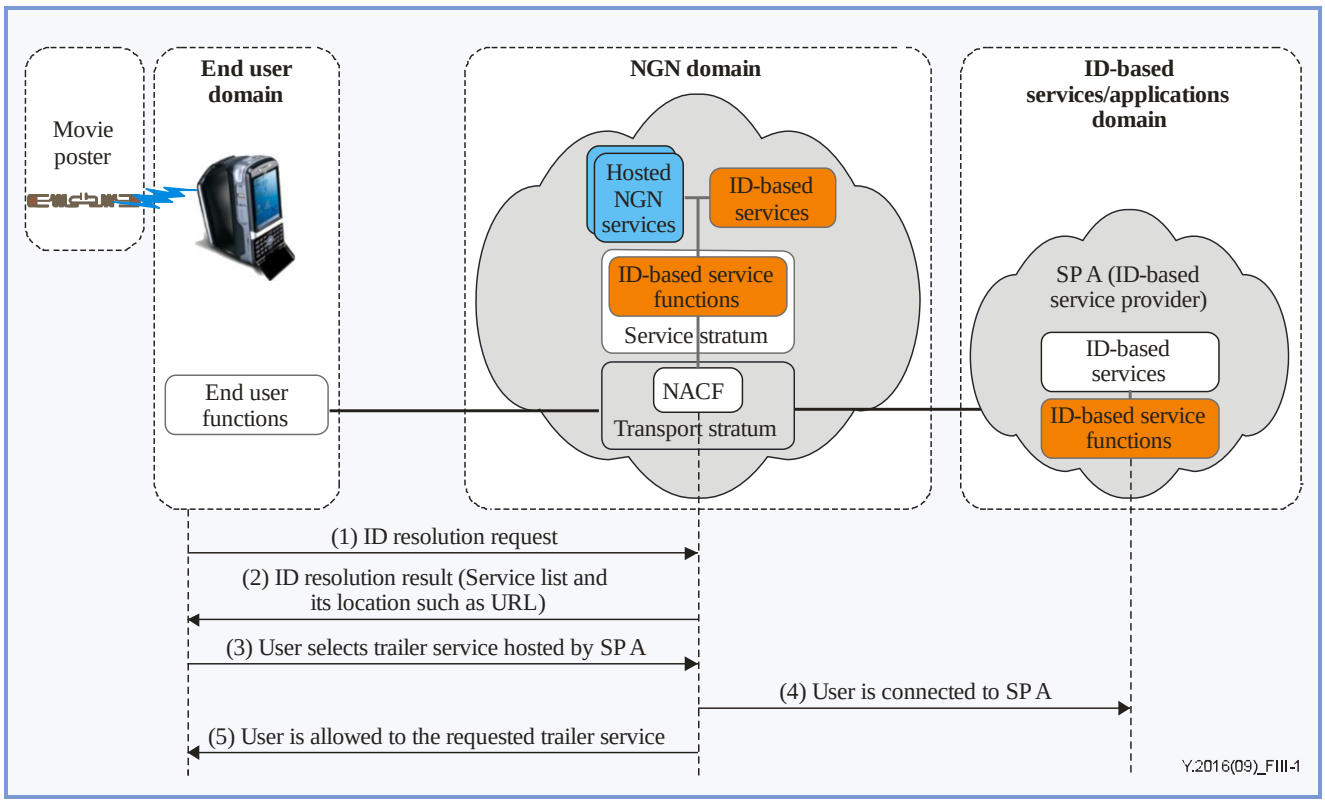


Figure III.1 – Typical ID-based application and services information flow

The information flows are summarized as follows:

(1) The end-user device reads the identifier from the movie poster. For the identifier resolution, the end-user device attaches to the NGN service provider and requests identifier resolution.

(2) The NGN service provider performs identifier resolution using the identifier resolution protocol such as DNS, X.500 or specific protocols. The result of the identifier resolution process includes a service list and its location.

For example, service list may have "1. Preview, 2. Ticketing, 3. Review" services and its location may be "1. http://www.xxx.com/preview/themovie.html",
2. http://www.xxx.com/ticket/themovie.html",
3. http://www.xxx.com/review/themovie.html".

(3) The end-user selects "Preview" service hosted by service provider A.

(4) The end-user is connected to service provider A through the NGN service provider.

(5) "Preview" service is delivered to the user through the NGN service provider.

Appendix IV

Traceability mechanism and referential information flows

(This appendix does not form an integral part of this Recommendation)

Due to the mobility of the objects that contain an ID tag (called "ID tagged objects"), it is useful to get the current ID tag's position, and to know how the ID tagged object has been treated and what the ID tagged object has been used for. This information can be preserved in a database for traceability, called "track information database" in this appendix.

Because of mobility, it is difficult to configure this traceability information statically, and therefore a dynamic mechanism needs to be considered.

Any distribution of information in support of traceability shall meet national legal and regulatory requirements on data privacy and protection.

IV.1 Traceability mechanism

The traceability mechanism is as follows:

1) ID tagged object traceability

   It is unreasonable for the ID tagged object to transfer its location information to the network, since an ID tagged object has generally limited power. Given that the communication distance between the ID tag reading function and the ID tag is usually very short (i.e., within a few metres), and given that end-user functions containing the ID tag reading function usually have more intelligence, the track information DB stores the location information of the ID tag reading function as the location of ID tag.

   When a given ID tag reading function in the end-user functions identifies a new ID tag, the end-user functions send location information and the identifier of the ID tag to the track information DB.

2) Usage traceability

   The operation of writing/reading (to) ID tag(s), and the handling of identifier(s) stored in ID tag(s) are performed by the end-user functions. In order for the track information DB to get usage-related information, end-user functions can be required to send usage-related information to the track information DB.

IV.2 Information flows

Figure IV.1 provides an example of information flows for object traceability and usage traceability.

Detailed flows are as follows.

(1) End-user functions read an identifier from the ID tag. As a result, the corresponding end-user functions request ID resolution.

(2) ID resolution function provides the result of resolution to the end-user functions, including the URL of the track information DB and other relevant information.

(3) End-user functions notify the location information to the track information DB. The location information can be the location information of the ID reader or more globally of the end-user functions.

(4) As a result, the track information DB provides a confirmation notification.

End-user functions may perform some operations on the ID tag such as writing data or invoking some services. Such operations may be reported to the track information DB as follows:

(5) End-user functions notify the track information DB about the usage description.

(6) As a result, the track information DB returns a confirmation notification.

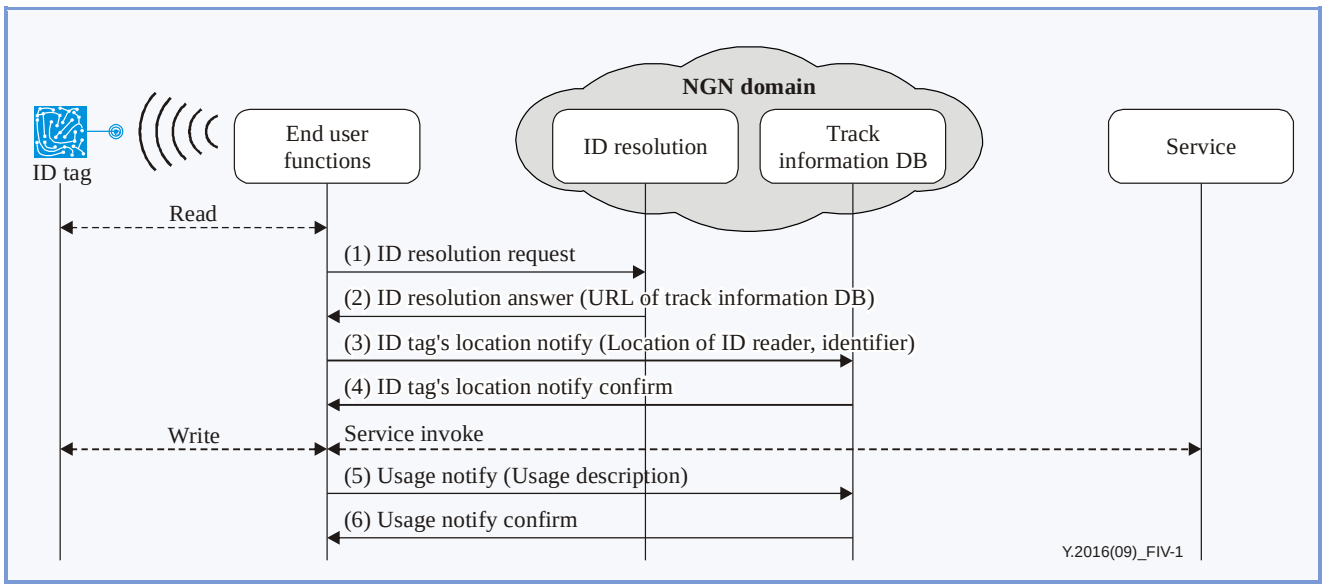


Figure IV.1 – Example of information flows for object traceability and usage traceability

Bibliography

[b-ITU-T Y.2001] Recommendation ITU-T Y.2001 (2004), *General overview of NGN*.

[b-ITU-T Y.2011] Recommendation ITU-T Y.2011 (2004), *General principles and general reference model for Next Generation Networks*.

[b-ITU-T Y.2201] Recommendation ITU-T Y.2201 (2007), *NGN release 1 requirements*.


WORLD

Y.4407/Y.2281

Framework of networked vehicle services and applications using NGN

Framework of networked vehicle services and applications using NGN

Summary

Recommendation ITU-T Y.2281 describes the framework of networked vehicle services and applications in the context of next generation networks (NGN). This Recommendation identifies the relationship between NGN and a networked vehicle as well as requirements taking into consideration the necessity of supporting networked vehicle services and applications using NGN. In addition, a framework architecture of NGN-capable networked vehicle and intelligent transport systems (ITS) infrastructure is described to support the communication features of an NGN harmonized with the networked vehicle.

History

Edition	Recommendation	Approval	Study Group
1.0	ITU-T Y.2281	2011-01-28	13

Keywords

ITS, networked vehicle, NGN, vehicle communication.

Table of Contents

Introduction

A vehicle is one of the important components utilizing network capabilities in terms of vehicle to infrastructure (V2I), vehicle to vehicle (V2V) and vehicle to home (V2H) communications. In that context, a networked vehicle can cooperate with next generation networks (NGNs) to support more advanced services and applications such as road safety applications, road traffic related applications, multimedia services and location-based implementation of these services.

Recommendation ITU-T Y.4407/Y.2281

Framework of networked vehicle services and applications using NGN

1 Scope

This Recommendation describes a framework for networked vehicle services and applications assuming operation in an area where telecommunication services are provided by NGN. This Recommendation identifies features and requirements for the provision of networked vehicle services and applications using NGN. This Recommendation also provides a framework architecture to support these features and requirements.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T Y.2012] Recommendation ITU-T Y.2012 (2010), *Functional requirements and architecture of next generation networks*.

[ITU-T Y.2201] Recommendation ITU-T Y.2201 (2009), *Requirements and capabilities for ITU-T NGN*.

[ITU-T Y.2291] Recommendation ITU-T Y.2291 (2011), *Architectural overview of next generation home networks*.

[ITU-T Y.2701] Recommendation ITU-T Y.2701 (2007), *Security requirements for NGN release 1*.

[ITU-T Y.2702] Recommendation ITU-T Y.2702 (2008), *Authentication and authorization requirements for NGN release 1*.

[ITU-R M.1453-2] Recommendation ITU-R M.1453-2 (2005), *Intelligent transport systems – Dedicated short range communications at 5.8 GHz*.

[ITU-R M.1457-9] Recommendation ITU-R M.1457-9 (2010), *Detailed specifications of the terrestrial radio interfaces of International Mobile Telecommunications-2000 (IMT-2000)*.

[ETSI EN 302 665] ETSI EN 302 665 V1.1.1 (2010), *Intelligent Transport Systems (ITS); Communications Architecture*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 functional entity [ITU-T Y.2012]: An entity that comprises an indivisible set of specific functions. Functional entities are logical concepts, while groupings of functional entities are used to describe practical, physical implementations.

3.1.2 intelligent transport systems (ITS) [b-ITU-R Handbook]: ITS is defined as systems utilizing the combination of computers, communications, positioning and automation technologies to improve the safety, management and efficiency of terrestrial transport systems.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 infrastructure: The basic facilities and systems comprised of network nodes (i.e., switches and/or routers) and the means to connect them (i.e., wired (cable or fibre) or wireless) for the purpose of communication between two end-points.

3.2.2 networked vehicle: A vehicle capable of providing communication between entities within the vehicle as well as between entities within the vehicle and ITS infrastructure or other communication networks such as NGN using various access technologies.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

ABS	Automatic Break System
AM	Amplitude Modulation
ANI	Application to Network Interface
API	Application Programming Interface
CALM	Communication Access to Land Mobile
DMB	Digital Multimedia Broadcasting
DRM	Digital Rights Management
DSRC	Dedicated Short Range Communications
DVB-T	Digital Video Broadcasting – Terrestrial
DVD	Digital Versatile Disc
ECM	Engine Control Module
ECU	Electronic Control Unit
FM	Frequency Modulation
FMC	Fixed Mobile Convergence
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
H2V	Home to Vehicle
ICT	Information and Communication Technologies
IMT	International Mobile Telecommunications
ISDN	Integrated Services Digital Network
ITS	Intelligent Transport Systems
IVN	In-Vehicle Network
MAC	Media Access Control
MM	Multifunction Mobile

MP3	Moving Picture 3 (digital audio format)
NAPT	Network Address Port Translation
NAT	Network Address Translation
NGN	Next Generation Networks
NNI	Network to Network Interface
OBD	On Board Diagnostics
OBU	On Board Unit
OEM	Original Equipment Manufacturer
PDA	Personal Digital Assistant
PSTN	Public Switched Telephone Network
QoE	Quality of Experience
QoS	Quality of Service
RFID	Radio Frequency Identification
RTK	Real-Time Kinematic
SMS	Short Message Service
SNI	Server to Network Interface
SNMP	Simple Network Management Protocol
SP	Service Platform
SPI	Service Programming Interface
SRC	Seamless Radio Connectivity
TCM	Transmission Control Module
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
UNI	User to Network Interface
USB	Universal Serial Bus
UWB	Ultra-Wideband
V2G	Vehicle to Grid
V2H	Vehicle to Home
V2I	Vehicle to Infrastructure
V2I2V	Vehicle to Infrastructure to Vehicle
V2V	Vehicle to Vehicle
VGP	Vehicle Gateway Platform
VoD	Video on Demand
VPN	Virtual Private Network
WLAN	Wireless Local Area Network

5 Conventions

In this Recommendation, the "I" in the keywords "V2I" and "V2I2V" stands for ITS infrastructure.

In this Recommendation:

The keywords "is required to" indicate a requirement which must be strictly followed and from which no deviation is permitted, if conformance to this Recommendation is to be claimed.

The keywords "is prohibited from" indicate a requirement which must be strictly followed and from which no deviation is permitted, if conformance to this Recommendation is to be claimed.

The keywords "is recommended" indicate a requirement which is recommended but which is not absolutely required. Thus, this requirement need not be present to claim conformance.

The keywords "is not recommended" indicate a requirement which is not recommended but which is not specifically prohibited. Thus, conformance with this Recommendation can still be claimed even if this requirement is present.

The keywords "can optionally" indicate an optional requirement which is permissible, without implying any sense of being recommended. This term is not intended to imply that the vendor's implementation must provide the option, and the feature can be optionally enabled by the network operator/service provider. Rather, it means the vendor may optionally provide the feature and still claim conformance with this Recommendation.

6 High-level view of a networked vehicle

Intelligent transport systems (ITS) provide capabilities, using information and communication technologies (ICT), to support safe and efficient use of transport infrastructure and transport means (e.g., car, train, plane or ship) for transportation of goods and humans.

A vehicle (i.e., bus, car, lorry (truck) or van), which is one of the most dominant means of transportation, is either moving or stationary.

A moving vehicle requires various capabilities including the support of networked vehicle services and applications such as road safety, traffic status, automatic toll road billing and emergency information. The support of such networked vehicle services and applications normally requires the use of communication capabilities supported by telecommunication infrastructures such as NGN, dedicated short range communication (DSRC) [ITU-R M.1453-2] or IMT-2000 [ITU-R M.1457-9]. Considering the importance of such communications to/from a networked vehicle, QoS/QoE, security and mobility, together with other factors, should be supported by the above-mentioned networking infrastructures.

A stationary vehicle requires, from time-to-time, static operation such as maintenance and/or upgrade of devices. While in a stationary mode, networked vehicle communication generally needs support from residential home networks (e.g., that of the vehicle owner) and/or other networks (e.g., that of a repair workshop).

Thus, key features of NGN such as fixed mobile convergence (FMC) support providing QoS and security play an important role. Figure 6-1 shows a high-level conceptual view of the relationship between a networked vehicle and external networks using NGN capabilities. In this figure, a networked vehicle is composed of personal devices and on-board equipments, etc. Roadside station and central station (e.g., traffic information centre) are examples of elements within ITS infrastructure.

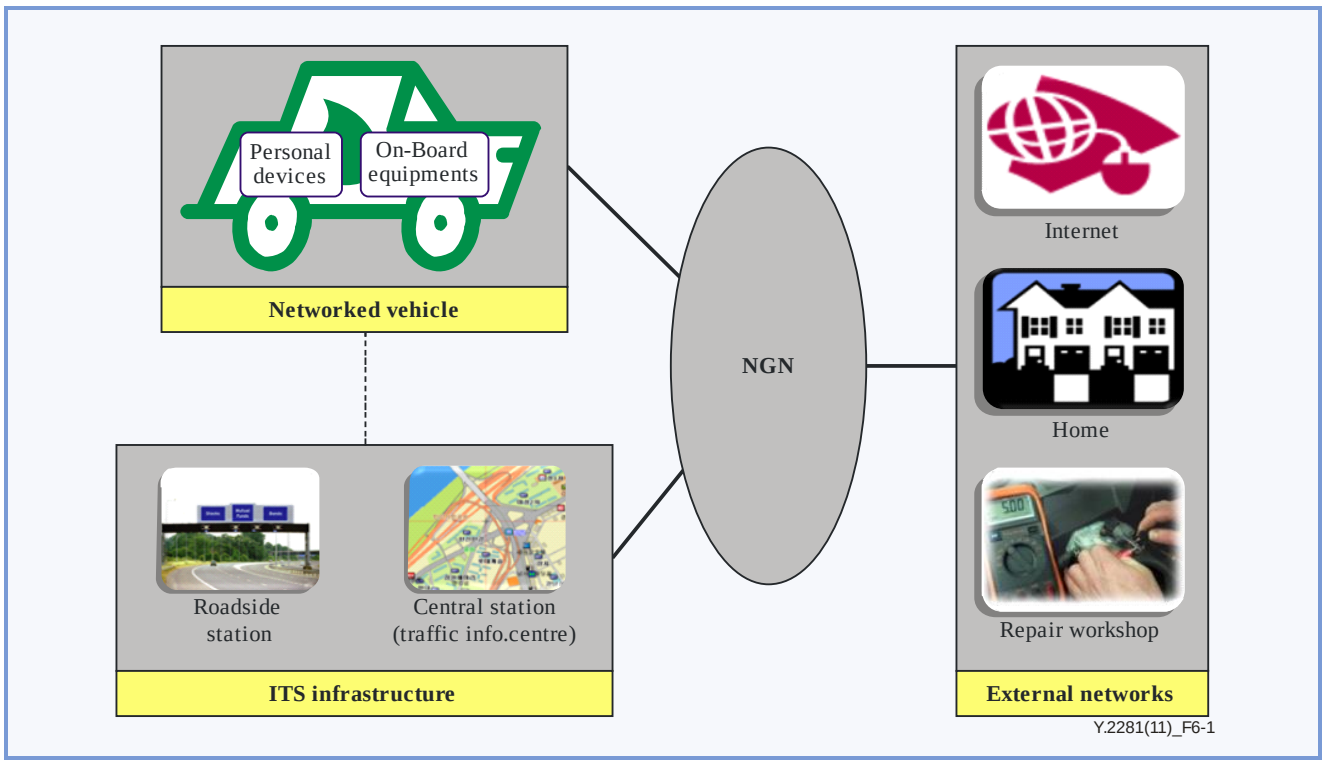


Figure 6-1 – High-level view of a networked vehicle using NGN capabilities

7 Services and communication features of networked vehicles

7.1 Networked vehicle services and applications

From the service and application aspects related to a networked vehicle, the following perspective can be considered:

1) **Vehicle maintenance-oriented services/applications (SA-1)**: The primary objective of these services/applications is the management of the vehicle itself. Possible vehicle management-oriented services/applications include:
   - Remote vehicle diagnosis: This service/application supports the remote monitoring of the vehicle status based on the status of the relevant vehicle devices and management of internally controlled and monitored diagnostics. Security and privacy protection of remotely monitored vehicle information is recommended to be ensured.
   - Vehicle data/software provisioning and update

2) **Road safety services/applications (SA-2)**: The primary objective of these services/applications is the improvement of road safety. However, it is recognized that in improving road safety these services/applications may offer secondary benefits which are not directly associated with road safety. These services/applications are built upon the reporting of road-related situations such as vehicle accidents or vehicle malfunctioning. Possible co-operative road safety services/applications include:
   - "Driving assistance – Co-operative awareness" covers use cases such as 'Emergency vehicle warning', 'Slow car warning', 'Intersection collision warning' [b-ETSI TR 102 638];
   - "Driving assistance – Road Hazard Warning" covers use cases such as 'Collision risk warning', 'Wrong way driving warning', 'Vehicle accident', 'Traffic condition warning', 'Road adhesion', 'Road visibility', etc. [b-ETSI TR 102 638] .

3) **Passenger-oriented services/applications (SA-3)**: The primary objective of these services/applications is to provide passengers in the vehicle with telecommunication-oriented services/applications such as interpersonal conversational services, audiovisual services (e.g., IPTV services), informational services (e.g., about the presence of locally based services or/and points of interest), access to Internet.

   – These services/applications are provided using devices either already built in the vehicle and/or temporarily operated within a vehicle through communication inside the vehicle or to/from outside the vehicle. The vehicle owner can download or upload multimedia data through a head unit (i.e., built-in on-board unit (OBU)) with networking capability or personal stations such as a smart phone.

4) **Traffic efficiency services/applications (SA-4)**: The primary objective of these services/applications is the improvement of traffic flow. However, it is recognized that improving traffic flow may offer secondary benefits not directly associated with traffic management. These services/applications provide for collecting and delivering of real-time traffic information by exchanging vehicle probe data periodically among vehicles and traffic information centres. The time stamp, the vehicle speed and location are some of the core elements of vehicle probe data, and should be transmitted to the centre in predetermined time periods. Possible applications include:

   – "Speed management" covering use cases such as 'Regulatory/contextual speed limits' and 'Traffic light optimal speed advisory' [b-ETSI TR 102 638];

   – "Co-operative navigation" covering use cases such as 'Traffic information and recommended itinerary', 'Enhanced route guidance and navigation' and 'Limited access warning and detour notification' [b-ETSI TR 102 638].

5) **Vehicle-oriented services/applications (SA-5)**: The primary objective of these services/applications is the support of vehicle logistics and freight-oriented applications. These applications include applications such as fleet management and vehicle parking management applications.

7.2 Communications features

Two types of communications are considered, within a networked vehicle and to/from outside a networked vehicle:

1) **Communication inside a networked vehicle**: This is a case in which communications are made inside a networked vehicle. The following types of communication can be distinguished:

   • **Communication among objects which are equipped components of a networked vehicle**: This is the case where objects of a networked vehicle (such as engine control module (ECM), transmission control module (TCM), automatic break system (ABS)) communicate among themselves and/or communicate with a central vehicle control/monitoring platform.

   • **Communication with devices temporarily operating within a networked vehicle**: This is the case of any devices temporarily operating in a networked vehicle (typically devices belonging to passengers such as a smart phone, an MP3 player or a personal digital assistant (PDA)). These devices may communicate among themselves and/or with equipment of the networked vehicle (such as a car radio, or a TV set).

2) **Communication from/to outside of a networked vehicle**: This is the case where communication is established between a given networked vehicle and other objects located outside of the networked vehicle. The following types of communication can be distinguished:

- **Vehicle-to-vehicle (V2V) communication**: This is relevant both to ad hoc communication and communication through various mobile access networks. This communication occurs between a networked vehicle and another networked vehicle to propagate safety critical information within a short period of time while the networked vehicle is moving. V2V communication is classified into two categories. One is direct communication between moving networked vehicles by using point-to-point, point-to-multipoint, or multi-hop routing. The other is vehicle to infrastructure to vehicle (V2I2V) multi-hop communication having more than two-hops by way of either a roadside station or a mobile network infrastructure. V2V direct communication mode is out of scope of this Recommendation.
- **Vehicle-to-infrastructure (V2I) communications**: This relates to communication between a networked vehicle and an ITS infrastructure through networks such as NGN, international mobile telecommunications (IMT)-2000, wireless local area network (WLAN), or dedicated short range communications (DSRC).
- **Vehicle-to-home (V2H) communications**: This relates to communications between a networked vehicle and a residential home network through networks such as NGN. When the networked vehicle is parked in a home, the networked vehicle could be treated as an element of the home network. There are two cases: one is for handling it as the owner's home device (i.e., belonging to the same owner as the home), and the other is to treat it as a visiting device (i.e., the one that does not belong to the owner of the home, but which is allowed as a visitor).
- **Vehicle-to-grid (V2G) communications**: This relates to communications between a networked vehicle and utility grids for smart charging. Ethernet over power-line adapter or IMT-2000/DSRC can be used for this type of communication.

7.3 Relationship between networked vehicle services and applications with communication features

A key aspect of a networked vehicle is its ability to communicate while transporting persons or goods. Therefore, at a given time, a networked vehicle can be considered as a moving equipment and/or device. In addition, a networked vehicle may also stay in a specific place such as in a car park or garage for a certain period of time. In this situation, a networked vehicle can also be considered as a fixed equipment and/or device.

Based on whether the network vehicle is a moving vehicle or a stationary vehicle (as described in clause 6), Table 7-1 provides the summary of the relationship between networked vehicle services and the applications described in clause 7.1 with communication features identified in clause 7.2.

Table 7-1 – Relationship between networked vehicle services and applications with communication features

Situation of networked vehicle	Communication aspects				
	Inside vehicle		Outside vehicle		
	Between on-board equipments	Between a vehicle and temporary devices	V2V	V2I	V2H/V2G
Moving vehicle	SA-1, SA-2	SA-1, SA-3	SA-2	SA-1, SA-2, SA-3, SA-4, SA-5	SA-1, SA-3
Stationary vehicle	SA-1, SA-3	SA-3	N/A	SA-1, SA-3, SA-5	SA-1, SA-3
N/A = Not Applicable					

From these key communication aspects, a networked vehicle can be considered as follows:

- a fast moving 'terminal and/or residential home network' interacting with several end devices;
- a service platform which monitors/controls the networked vehicle operation (including remote operation) and also supports multimedia services and applications by passengers such as video on demand (VoD), music within a networked vehicle;
- a communication platform for V2V, V2I and V2H communications.

Those features are the basic input for the requirements identified in clause 8.

8 Requirements for networked vehicle services and applications using NGN

A networked vehicle is a vehicle with various devices, including those belonging to the vehicle occupants. It should be possible to connect to a networked vehicle through various networks, including NGN. This clause identifies requirements of networked vehicle services and applications using NGN.

8.1 Requirements for a networked vehicle communicating with ITS infrastructure

8.1.1 General requirements

This clause identifies requirements for a networked vehicle in terms of general aspects of telecommunication.

- **Maintenance of positioning information**

  A networked vehicle is required to maintain positioning information obtained from the global navigation satellite system (GNSS) or other sources. Requirements include:

  i) collection and management of the networked vehicle's positioning information calculated from various sources such as a GNSS receiver (i.e., real-time kinematic (RTK)), a wireless network (e.g., WLAN, ZigBee, ultra-wideband (UWB)) and sensors (e.g., Gyro, radio frequency identification (RFID), accelerometer);

  ii) periodic reception of positioning information, and transmission of information either periodically, based on position triggers or as a response to a server application request.

- **Ad hoc network connectivity**

  Networked vehicle occupant devices connect smoothly to the networked vehicle.

  i) a networked vehicle is required to support ad hoc network connectivity with networked vehicle occupant devices such as an MP3 player, a PDA, a mobile phone, etc., for various value-added services (e.g., user's operation via the mobile phone, multimedia sharing, the user's phone address book);

  ii) a networked vehicle is recommended to support multiple communication interfaces such as Bluetooth, ZigBee, and WLAN.

- **IPv6 support**

  i) A networked vehicle is required to support IPv4 and is recommended to support IPv6 packet delivery over ITS communication.

- **Network management**

  i) A networked vehicle is required to have network management functionality in order to differentiate services according to the networked vehicle's priority in terms of services.

- **Operational requirements**

  i) A networked vehicle is required to handle multiple clients for the on-board diagnostics (OBD) system, to minimize the risk of collisions (i.e., send multiple requests at the same time).

8.1.2 Service and application aspects requirements

- **Use of networked vehicle data and ITS infrastructure**

  Devices can receive the input from in-vehicle controls and restrict access to vehicle/nomadic devices, depending on authorization.

  i) A networked vehicle is required to collect in-vehicle electronic control unit (ECU) and sensor data as well as to translate it into a standardized format for the purposes of networked vehicle services and applications.

8.1.3 Communication aspects requirements

- **Seamless access to networks**

  Since a communication link is continuously changing due to the mobility of a networked vehicle, enabling seamless access to networks requires autonomous switching between the "best" available communication systems at the current time and location. Therefore, the following measures are required to support seamless access:

  i) a networked vehicle is required to have interfaces to one or more wide area networks such as NGN, and one or more ad hoc networks for networked vehicles including short range radio communication;

  ii) since NGN supports multiple access technologies and mobility as basic functionality, a networked vehicle is required to provide the capability for initiating end-user functions to access the NGN services and the management of IP connectivity.

- **Security**

  Security is an issue because of complex technologies and sharing of sensitive information of a networked vehicle. Information of a networked vehicle should be protected from any malicious use such as invoking of malfunction, abuse of private information, etc. Therefore, the following steps are identified as requirements to support security:

  i) A networked vehicle is required to keep security through protocols and cryptographic mechanisms deployed in the components of the vehicle.

- **Latency**

  Vehicle safety applications allow a degree of latency depending on safety services requirements and on the network characteristics. Safety applications as 'pre-crash warning' or 'lane changing assistance' require real-time communication processing with accompanying vehicles. To exchange the emergency information and invoke safety critical controls for the networked vehicle or alert the driver, low latency communication should be supported. On the contrary, safety services as a moving road work and an approaching emergency networked vehicle can tolerate seconds of latency.

  i) A networked vehicle supporting stringent safety applications is required to support low latency communication to exchange emergency warning information within a certain period of time (e.g., end-to-end latency ≤ 100 ms) to assure the reliability of communication with nearby moving vehicles.

- **Beaconing of vehicle information**

  A networked vehicle providing safety applications could gather its own vehicle information and transmit the information to its neighbouring networked vehicles through beacons. Vehicle information includes globally unique networked vehicle's identifier (e.g., MAC address), location, speed, etc. The beacon protocol is applied between networked vehicles and also between a networked vehicle and a roadside station to exchange positions of nearby moving vehicles.

  i) a networked vehicle is required to transmit vehicle information periodically to its neighbouring networked vehicles;

  ii) a networked vehicle is required to be able to receive vehicle information periodically from its neighbouring networked vehicles. It is also required to maintain its neighbouring networked vehicles' information. The roadside station is recommended to be able to receive/transmit networked vehicles' beacon information.

- **Multi-hop communication**

  In case of beaconing of vehicle information, it is necessary to extend the original emergency messages' propagation range. To do so, it is recommended to devise a multi-hop communication technique. By means of the multi-hop communication, a networked vehicle can communicate with another networked vehicle which is outside of its propagation range.

  i) a networked vehicle and a roadside station exchange probe data collected from each networked vehicle and roadside sensors. The traffic and road status information is propagated to networked vehicles by using multi-hop communication;

  ii) a networked vehicle is required to support multi-hop communication with hop count parameters depending on the distance from an "event" or "time" since the event may be just as relevant.

- **Vehicle ad hoc routing**

  To support multi-hop communication, a message should be forwarded from a networked vehicle to other networked vehicles based on the routing schemes. A number of ad hoc routing algorithms can be applied application by application.

  i) A networked vehicle is required to support ad hoc routing for the following purposes:

     - to exchange the networked vehicle's positioning information with the neighbouring networked vehicles;
     - to deliver advanced traffic information to the networked vehicles that follow;
     - to receive traffic/roadside information from the infrastructure.

- **Network address translation**

  NGN is required not to preclude solutions for access of a networked vehicle to an NGN with network address translation (NAT)/network address port translation (NAPT) and firewalls in the user environment where the assignment of IP addresses to networked vehicle can optionally be done by the user network. These addresses need not be routable in the NGN.

  A networked vehicle's access to the NGN is required to support the following configurations:

  i) direct connectivity and interaction between the networked vehicle and the NGN;

  ii) indirect connectivity and interaction between individual devices in a networked vehicle and the NGN.

- **Vehicle-to-home (V2H) communication**

  The requirements for a networked vehicle's access to the home network are as follows:

  i) A networked vehicle's access to the home network is recommended to be enabled by access services provided within the home network (locally and/or through interconnected NGN).

  ii) A networked vehicle's access to the home network is recommended to support:

  - security, management and QoS for interoperability with home networks;
  - device provisioning and service configuration including remote access.

8.1.4 Accessibility requirements

Accessibility is required to ensure that the specified services and features are also usable as much as possible by people with disabilities. This clause describes accessibility requirements for networked vehicle services and features by applying the checklist defined in [b-ITU-T Technical Paper].

- Control of devices through a user interface

  i) A networked vehicle is recommended to have a multi-modal interface controlling the ITS infrastructure for driving assistance.

- Control of services

  i) A networked vehicle is recommended to have alternative ways to control networked vehicle services and applications. For example, real-time traffic information obtained from a roadside station can change the driving route guided from ITS infrastructure under a co-operative system.

- Media transport

  i) a networked vehicle is recommended to support text transport properties to present various safety/infotainment messages and alarms to drivers;

  ii) a networked vehicle is recommended to support audio transport properties to present various safety messages and alarms to drivers;

  iii) a networked vehicle is recommended to support video transport properties to present multimedia traffic information and record the scene of accidents.

- Media entry by the user

  i) Video entry properties are recommended to be selected so that it is possible to present video with network supported quality for multimedia traffic information. This property is recommended to be selected so that it is possible to record video pre/post scene of accidents and incidents automatically.

- Media presentation to the user

  i) a networked vehicle is recommended to support various ways of presenting information according to the user (or passenger) situation such as text and graphic (including multimedia) presentations for a person with hearing difficulties, and sound and vocal presentation of information for a person with visual difficulties;

  ii) a networked vehicle, when provided with the above media presentation requirement, is required to support emergency information following the above media presentation to the user (or passenger) in the networked vehicle.

- User and device profile management

  i) A networked vehicle is recommended to support the user and device profile management. The profiles should themselves be configurable, or be configured through exterior sources such as NGN or personal devices to assist disabled people.

- Video resolution for sign language and lip reading

  i) A networked vehicle is recommended to support a high quality video resolution for sign language and lip reading good visual reproduction of movements [b-ITU-T H-Sup.1].

8.2 NGN requirements for networked vehicle services and applications

NGN is recommended to allow the simultaneous use of multiple types of access technologies by a single networked vehicle. Therefore, the coordination of the multiple connections is required for networked vehicle services and applications from the network point of view.

It should be noted that it is not intended to preclude the attachment of terminal equipment which could enable interface adaptation to varying user requirements, including the needs of people with disabilities, using commonly provided user interface devices.

- **Maintenance of positioning information**

  Since positioning information is essential for networked vehicle services and applications, the networked vehicle is recommended to maintain the positioning information obtained from GNSS or other sources.

  i) NGN is recommended to support network-based positioning when a networked vehicle requests its own positioning information to a location server in the core network.

- **IPv6 support**

  i) NGN is recommended to support IPv6 network access for the roadside station and the OBU in a networked vehicle.

  NOTE 1 – NGN domains may support user equipment using IPv4 only, IPv6 only, or both at the user-network interface.

  NOTE 2 – In NGN, it is assumed that IPv6-based user equipment can also support IPv4 at the user-network interface.

- **Network management**

  i) NGN is recommended to support network management (e.g., simple network management protocol (SNMP)) with roadside station for exchange of network-attached devices conditions that need administrative attention.

- **Use of networked vehicle data and ITS infrastructure**

  The OBU located within a networked vehicle can receive input from in-vehicle controls and restrict access to personal station, depending on authorization.

  i) NGN is required to support secure transmission of networked vehicle data by using advanced security mechanisms such as confidentiality, anonymity, and traceability.

- **Priority**

  A networked vehicle is required to support a priority scheme by the type of service and also the type of client for the networked vehicle.

  For this purpose, the following steps are identified as NGN requirements:

  i) NGN is required to support "Prioritization" of networked vehicle transmission requests in two different levels: 'Concurrent channel access requests' within a single device' and 'Concurrent medium access requests among different devices';
     - Concurrent channel access: Prioritization among networked vehicle applications which request channel access concurrently;
     - Concurrent medium access: Prioritization among networked vehicle stations which try to access the physical communication channel simultaneously.

  ii) NGN is required to provide negotiation mechanisms between the networked vehicle and NGN to ensure the end-to-end networked vehicle service/application with different priorities.

  NOTE 3 – As an example, signalling an airbag deployment in a traffic accident should have a higher priority than downloading an entertainment video.

  NOTE 4 – Service/application priority should be identified by means of QoS and security control.

- **QoS**

  QoS can be a key criterion for choosing an available network by the user policy according to charging, network environments such as different access interfaces. Therefore, the following steps are identified as requirements:

  i) NGN is required to support per-flow, per-session, and per-service-class QoS control granularity for the networked vehicle.

  ii) NGN is required to support mechanisms prioritizing the delivery of emergency telecommunications of the networked vehicle and vehicle control information.

- **Location-based services**

  NGN is required to be able to receive location information, GNSS-based or network-based, of a networked vehicle in real time, as well as to manipulate this information dynamically in order to implement location-based services. Location information of the networked vehicle can be optionally provided upon request by an authorized service provider.

- **Network address translation**

  NGN is required not to preclude solutions for access of a networked vehicle to an NGN with NAT/NAPT and firewalls in the user environment where the assignment of IP addresses to the networked vehicle can optionally be done by the user network. These addresses need not be routable in the NGN.

- **Accessibility**

  i) Control of devices through a user interface:
     - NGN is recommended to support bidirectional connectivity to the networked vehicle to support the multi-modal commands for driving assistance. For example, the name of the driving destination can be input to the navigation system through the communication network with the help of an assistant in the service centre while driving.

  ii) Control of services:
     - NGN is recommended to support bidirectional connectivity to the networked vehicle to support the control of services for driving assistance.

iii) NGN requirements on accessibility [ITU-T Y.2201]:

Users with disabilities have a general need to be provided with means to control and use terminals and services in alternative ways and modes, suiting varied capabilities and preferences. Such requirements are best met by the inclusive design of the general provision of terminals and services.

- NGN is required to provide the means needed for the invocation of relay services. Relay services translate between various modes of telecommunication that are of interest for people with disabilities (e.g., sign language, lip reading, text, voice). Invocation of relay services can optionally be based on user preferences, address resolution or user commands.
- NGN is required to have the capability to invoke relay services by either party in an emergency telecommunication.

NOTE 5 – Other needs for users with disabilities to use emergency telecommunication services are handled in clause 20.4 of [ITU-T Y.2201].

9 Framework architecture of the networked vehicle and ITS infrastructure

This clause specifies an overall framework architecture of the networked vehicle.

Figure 9-1 shows a configuration model involving a networked vehicle including its communication types. The figure shows how networked vehicles relate to the ITS infrastructure and also to external networks such as a residential home network and an utility grid network for power distribution and transmission.

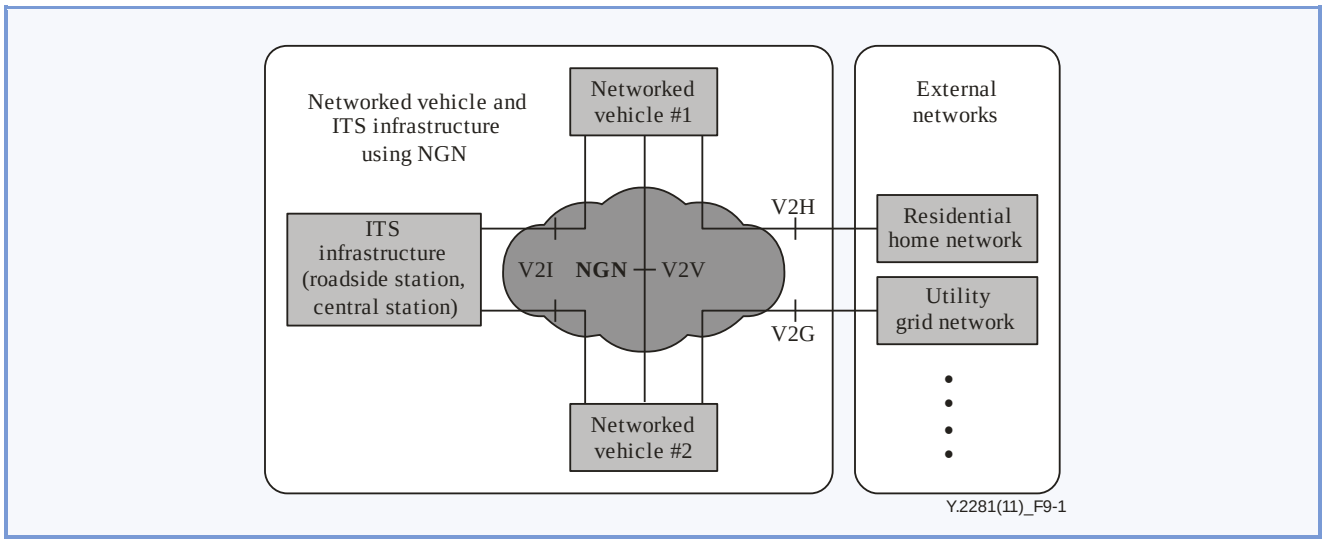


Figure 9-1 – Overall configuration model of networked vehicle and ITS infrastructure

While [ETSI EN 302 665] shows a peer-to-peer communication model using ITS-specific systems such as an ITS station router and ITS station gateways, it does not show how to use public communication networks such as NGN. Hence, this Recommendation identifies the use of NGN, thereby minimizing interoperability problems between peer-to-peer and public network scenarios. These interoperability features are especially important in the support of QoS, mobility and security with various multimedia services.

9.1 Reference architecture of the NGN-capable networked vehicle and ITS infrastructure

Based on the overall configuration model in Figure 9-1, it is possible to derive the reference architecture of the networked vehicle and ITS infrastructure. Figure 9-2 is a reference architecture model of the NGN-capable networked vehicle and ITS infrastructure based on the ITS station reference architecture model defined by [ETSI EN 302 665].

Functions for the support of networked vehicle services and applications are mapped into the ITS reference architecture which is defined in [ETSI EN 302 665]. The ITS reference architecture explains the generic functionalities that can be adopted in ITS stations. Figure II.1 shows the ITS station reference architecture defined in [ETSI EN 302 665].

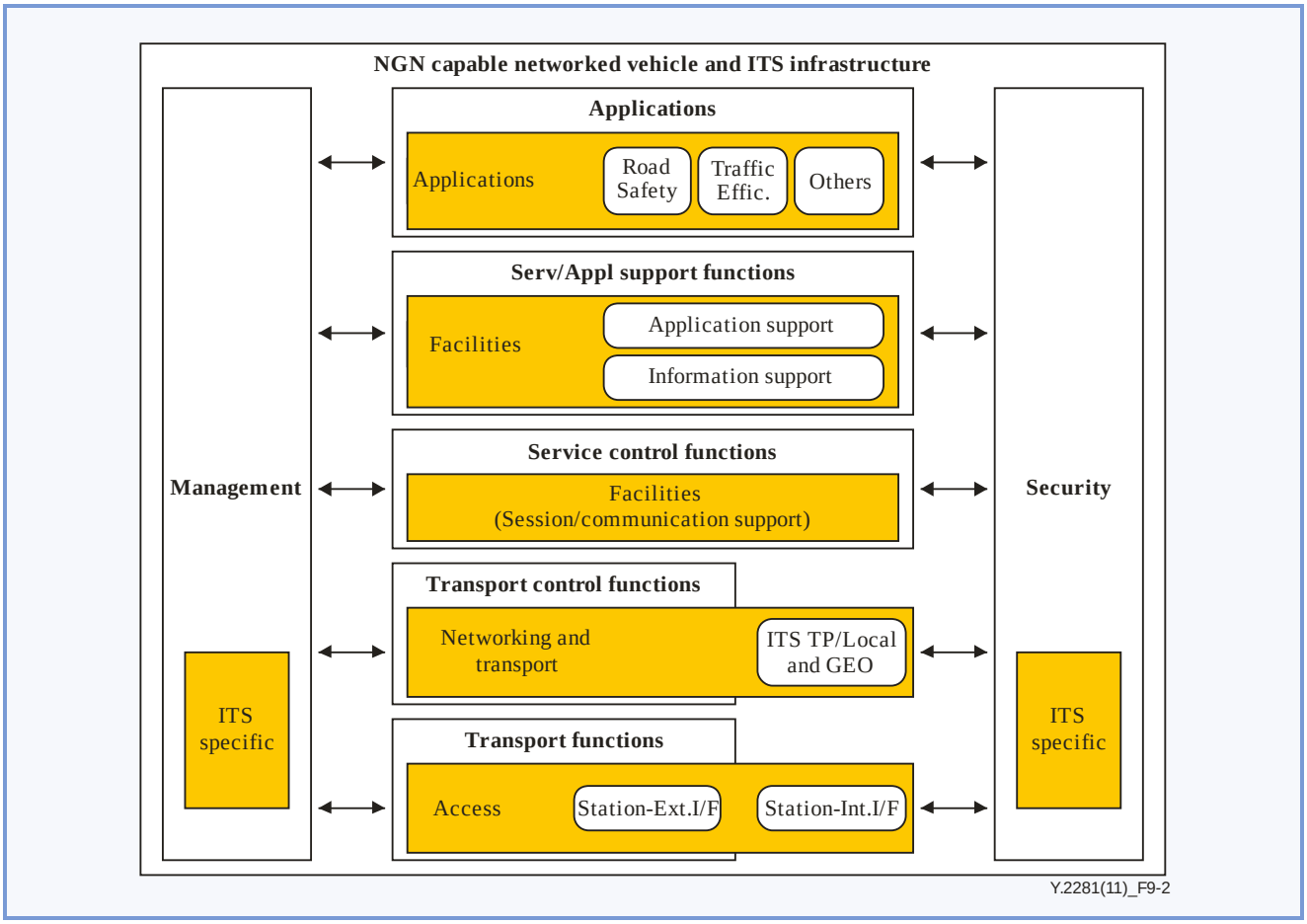


Figure 9-2 – Reference architecture of NGN-capable networked vehicle and ITS infrastructure

The key features of the functions used in Figure 9-2, based on the ITS reference architecture in [ETSI EN 302 665], are as follows:

- Applications: These functions manage networked vehicle services and applications with a function of classification, prioritization and channel assignment in the context of ITS communications.
- Facilities: These functions provide support to networked vehicle services and applications which can share generic functions and data such as session support.

- Networking and transport: These functions provide networking protocols such as GeoNetworking [b-ETSI TS 102 636], IPv6 networking and communication access to land mobile financial information exchange adapted for streaming (CALM FAST) and also provide TCP/UDP and dedicated ITS communication protocols.
- Access: These functions define the various access technologies which can be supported by the networked vehicle and ITS infrastructure.
- Management: These functions support management functions for congestion control, cross-interface, networking and application/service support.
- Security: These functions provide security functions for ITS communication protocol and applications.

9.2 Overview architecture of the NGN-capable networked vehicle and ITS infrastructure

[ITU-T Y.2012] specifies the NGN functional architecture and related functional entities. From the NGN's aspects, the NGN overview architecture consists of "End user functions", "Service stratum", "Transport stratum", "Management functions" and "NGN-based applications". The functions of the NGN-capable networked vehicle and ITS infrastructure, which is defined in clause 9.1, is located in the overview architecture, as shown in Figure 9-3.

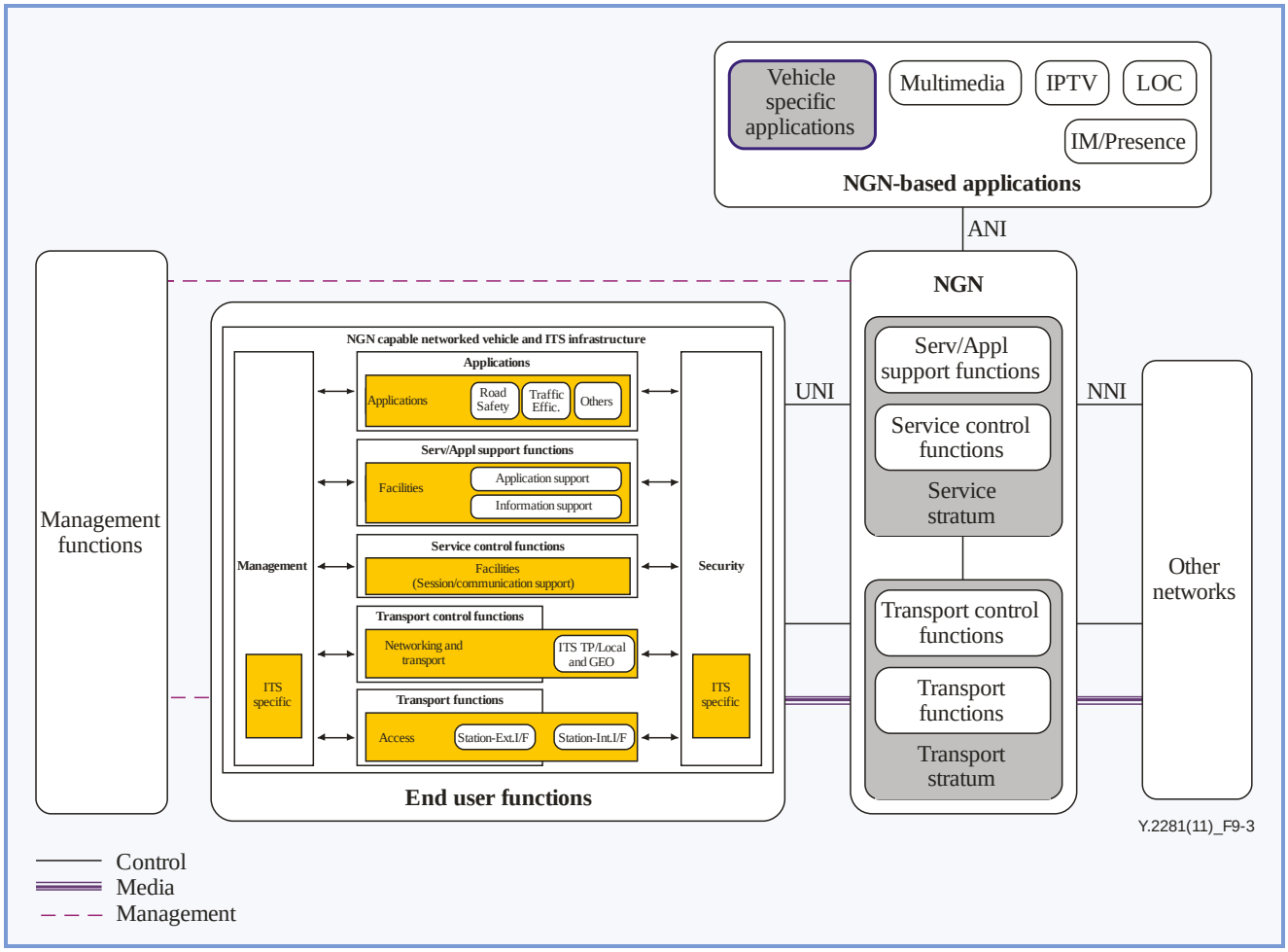


Figure 9-3 – Overview architecture of NGN-capable networked vehicle and ITS infrastructure in coorporation with NGN

In Figure 9-3, the networked vehicle related to the support of NGN capabilities is located in the end user functions. Vehicle-specific applications in the NGN-based applications are supported through the NGN for situations such as an emergency call or a public telecommunication.

NOTE – The NGN-capable networked vehicle and ITS infrastructure are the end user function in the NGN perspective [b-ITU-T Y.2291], the residential home network is located in the 'Other networks' box in Figure 9-3.

10 Security considerations

Security is an important issue for a networked vehicle. Different levels of security requirements need to be applied according to the environment where the networked vehicle is located, for example, when it is moving or stationary in a home network environment. Security requirements identified in [ITU-T Y.2201], [ITU-T Y.2701] and [ITU-T Y.2702] are applicable whenever the networked vehicle operates in NGN environments. Other cases of security requirements are out of scope of this Recommendation.

Appendix I

Use cases of networked vehicle services and applications using NGN

(This appendix does not form an integral part of this Recommendation.)

For the development of NGN capabilities and functional model to support networked vehicle services and applications, it is useful to look at usage flows of networked vehicle services and applications as well as to identify relevant mechanisms. In this regard, this appendix introduces "Use cases of the networked vehicle" using the service configuration model, as shown in Figure I.1.

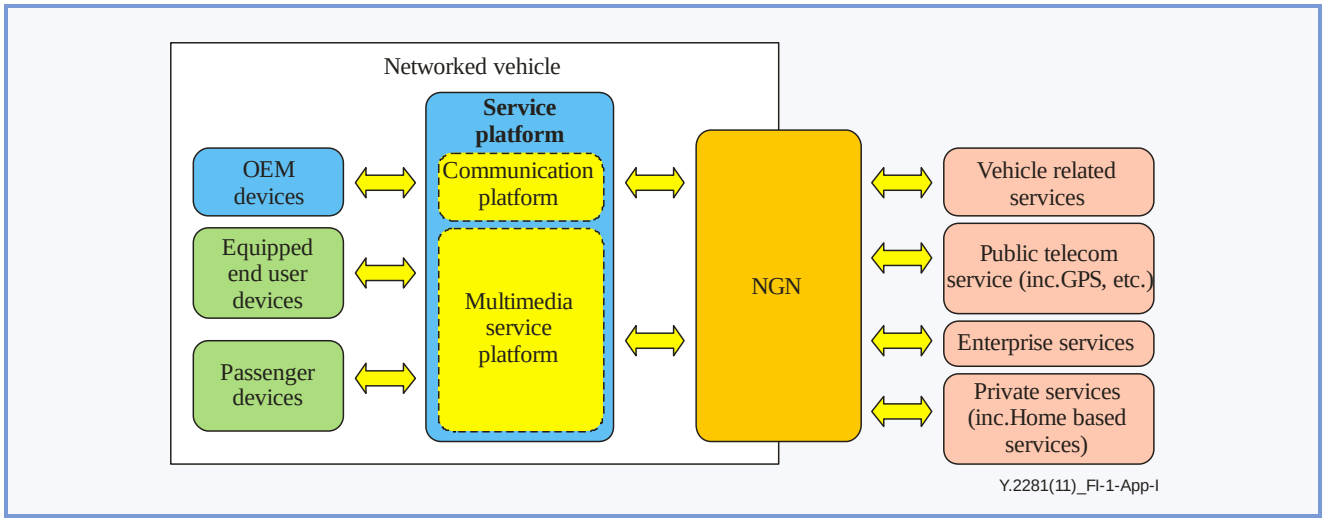


Figure I.1 – Service configuration model of a networked vehicle

Taking into account the communication features and above service configuration model, following classification of use cases can be derived as shown in Figure I.2.

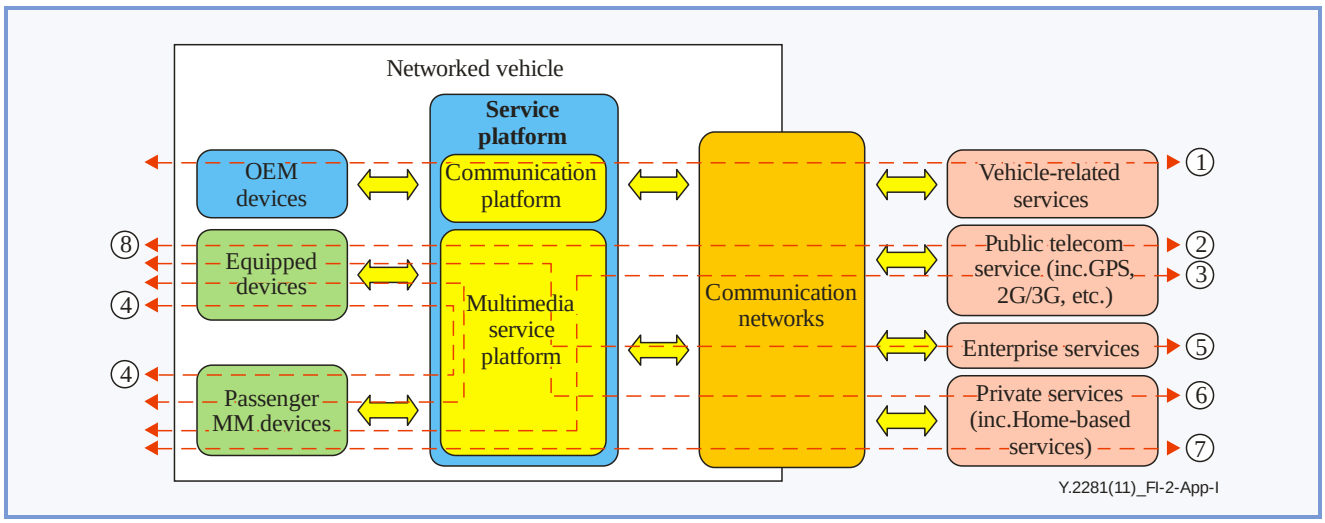


Figure I.2 – Classification of use cases for communication of a networked vehicle

Each use case is explained in Table I.1.

Table I.1 – Use cases of networked vehicle

Use Case	Objectives	Examples	NGN/Networked vehicle supports
1	Monitoring, operation and maintenance of networked vehicle	• Monitoring of networked vehicle status by communication with OEM devices • Optimization of networked vehicle operation and maintenance • Upgrade of relevant drivers for OEM devices and networked vehicle support system (may be VGP)	• Broadband with managed capability • Security • Mobility management
2	Use of public telecom services by using equipped terminal devices	• Radio programmes over FM/AM • GPS services • Telephone voice services • Internet access services • TV over mobile (e.g., DMB/DVB-T etc.) • Other public services, as appropriate	• Broadband • QoS • Security • Mobility management
3	Use of public telecom services by using passenger-owned terminal devices		
4	Sharing of resources and capabilities between equipped MM devices and passenger-owned MM devices	• Sharing music/video stored in passenger device(s) to vehicle music player (including the other direction) • Distributing video to the rear seat viewers and vice versa • Network gaming inside the networked vehicle (using vehicle equipped devices and passenger devices)	• Mounting devices into VGP with open I/F • Content protection (e.g., DRM) • Content distribution
5	Extension of business life while seated in a networked vehicle	• Use of company presence services (similar as Messenger) • Use of electronic signature to approve enterprise administration	• VPN • Security • QoS
6	Extension of home life while seated in a networked vehicle by using equipped devices	• Seamless use of personal/family scheduling • Seamless use of information from a networked vehicle to information stored in a home and vice versa	• VPN • Security • Connectivity to home network
7	Extension of home life while seated in a networked vehicle by using portable devices		
8	Use of public telecommunication access network for less stringent safety services	• Change of road conditions	• Security • Mobility management

Appendix II

Comparison between ITS station reference architecture and NGN functional architecture

(This appendix does not form an integral part of this Recommendation.)

II.1 Features and detailed functions of ITS station reference architecture

[ETSI EN 302 665] defines the ITS communication architecture using the following configuration and features of the ITS station reference architecture:

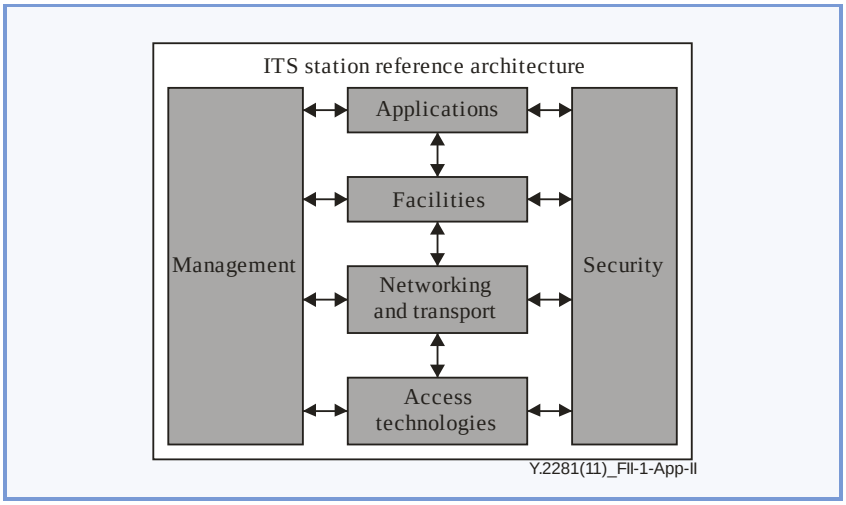
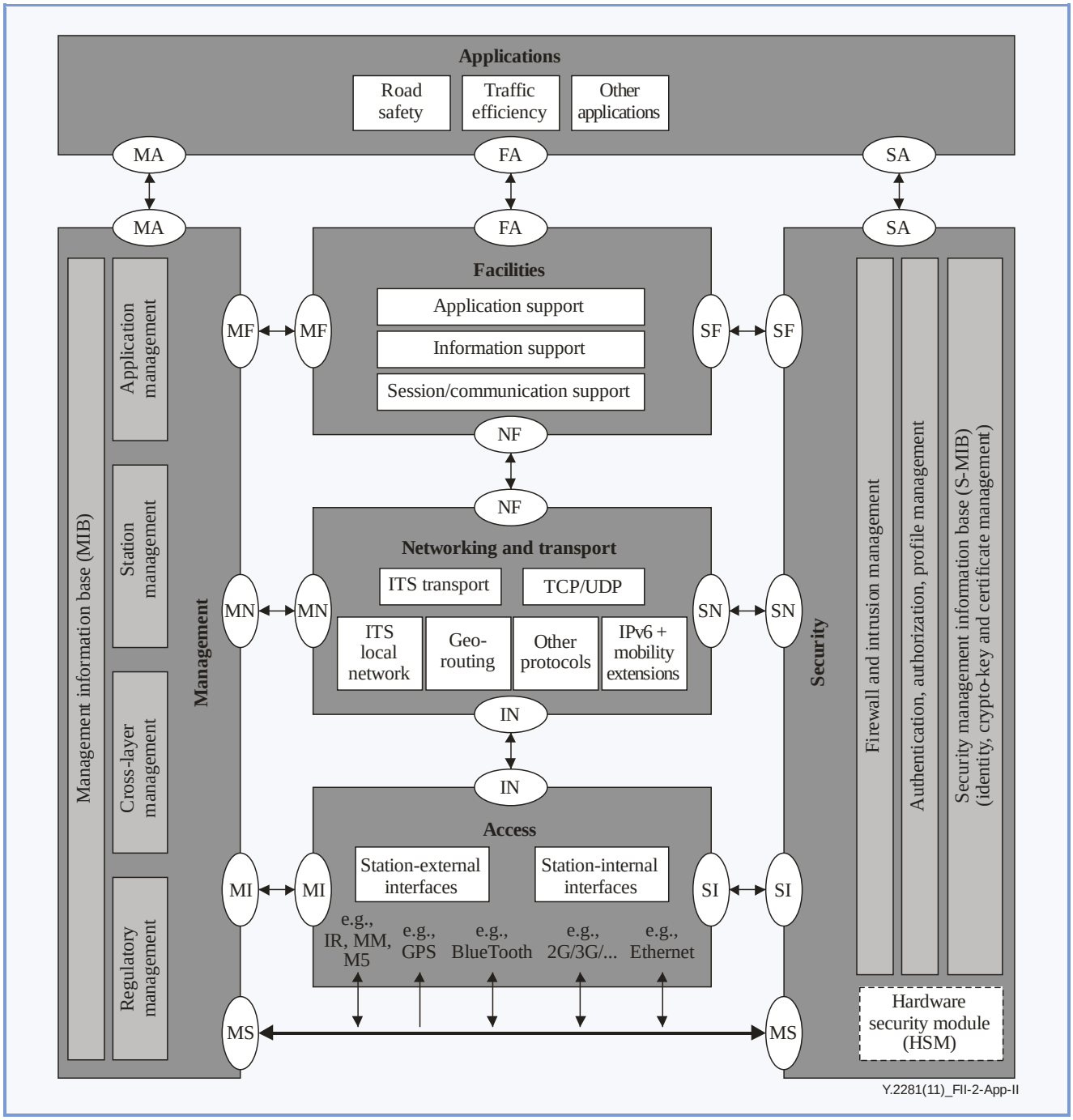


Figure II.1 – ITS station reference architecture

ITS station reference architecture is the functionality contained in ITS stations which are part of ITS sub-systems such as personal ITS sub-system (e.g., in hand-held devices), central ITS sub-system (part of an ITS central system), vehicle ITS sub-system (in cars, trucks, etc., in motion or parked) and roadside ITS sub-system (on gantries, poles, etc.). ITS station is the functional entity specified by the ITS station reference architecture, and it is composed of six entities such as 'Access' representing OSI layers 1 and 2, 'Networking and Transport' representing layers 3 and 4, 'Facilities' representing layers 5, 6 and 7, 'Applications', 'Security' and 'Management'.

Detailed functions in each layer have been also identified in [ETSI EN 302 665], as shown in Figure II.2.



F Figure II.2 – ITS station reference architecture with detailed functions

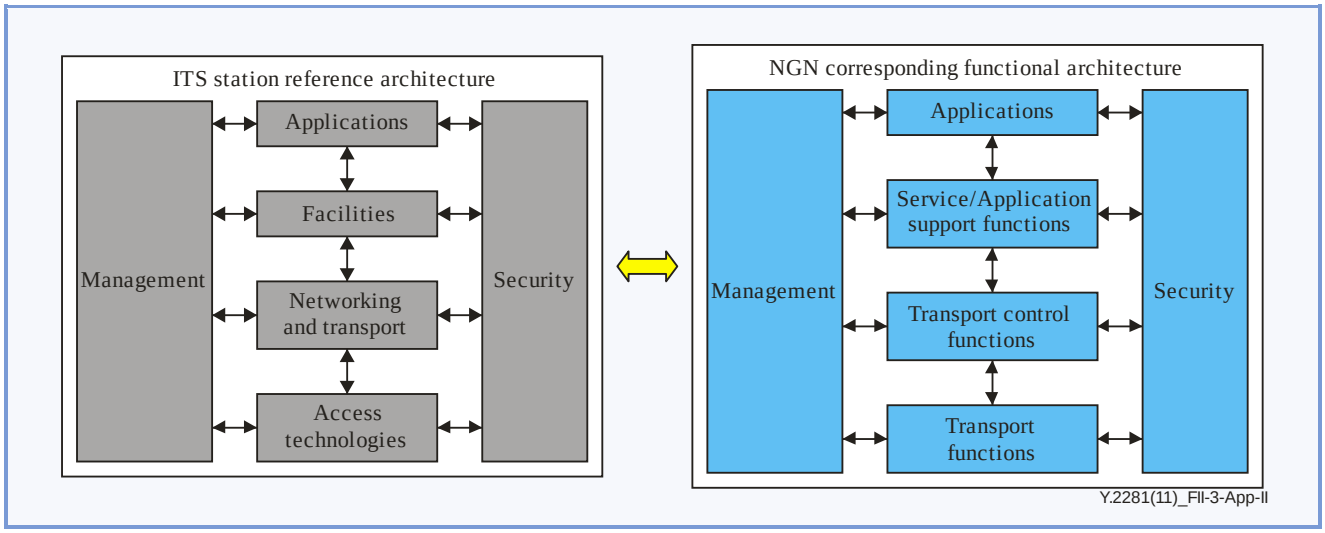
II.2 Analysis between two architectural models

An analysis of features and detailed functions, shown in clause II.1, reveals that the ITS station reference architecture has a structure quite similar to the functional architecture model of NGN, although some of the terminology ("Facilities", "Transport" and "Access technologies") in the ITS station reference architecture is used in a different way in NGN. Considering this different usage of terminology, this appendix shows a way to harmonize the ITS station reference architecture model into the functional architecture model of the networked vehicle over NGN, based on the analysis of features and detailed functions. Table II.1 shows a summary of the analysis between the ITS station reference architecture and the NGN functional architecture defined in [ITU-T Y.2012].

Table II.1 – Features and comparison between ITS station reference architecture and NGN functional architecture

ITS station reference architecture	Corresponding NGN functional architecture	Rationale
Access technologies	Transport functions	Even called "Access technologies" but this part generally represents the transport function itself. NGN applies at "Station external IF" but not at "Station internal IF".
Networking and transport	Transport control functions	This generally handles networking protocols such as IPv6, TCP/UDP, etc., so it should correspond to "Transport control functions".
Facilities	Service control and service/application support functions	"Facilities" is the most difficult part to map because of the different usage of naming, but it is mainly involved in "session support" and "ITS application". Thus, this should correspond to "Service/application support functions" in NGN.
Applications	Applications	This part mainly involves providing ITS applications, so it corresponds to "Application functions" in NGN.
Management	Management	Mostly the same but need to consider ITS specific requirements should be considered.
Security	Security	Mostly the same but need to consider ITS specific requirements should be considered.

Based on Table II.1, a possible mapping of the ITS station reference architecture into the NGN functional architecture can be depicted, as shown in Figure II.3.



F Figure II.3 – Correspondence between the ITS station reference architecture and the functions of NGN functional architecture

One of the fundamental considerations is to address the support of ITS specific aspects driven by service requirements, operation, administration and management of ITS. When ITS specific services need NGN capabilities, NGN functions should be incorporated into ITS station features, as shown in Figure II.4. Most of the detailed functions in the ITS station reference architecture are well mapped with the relevant functions of the NGN functional architecture, except for "Information support", the role of which is not clear, and the difference between "Application support" and "Communication support".

II.3 Features and functions for a networked vehicle

A networked vehicle should be operated as a part of ITS as well as a part of communication objects, used for public and private communication services and applications. This means that a networked vehicle should have enough capabilities to support ITS specific features and general communication features. Therefore, it is highly anticipated that combined (or harmonized) functions to support both features would be incorporated into a networked vehicle.

In this regard, the ITS station reference architecture covers the ITS specific service features but not enough to support communication features, specifically the need to adopt capabilities from the communication functions to support mobility, security, management capability and connectivity. It is anticipated to use capabilities provided by NGN for such features rather than develop different functions because a networked vehicle is also a part of communication objects, whether it is moving or stationary. Therefore, functions for a networked vehicle, especially over the NGN environment, are harmonized using both the ITS station reference architecture and the NGN functional architecture, as shown in Figure II.4.

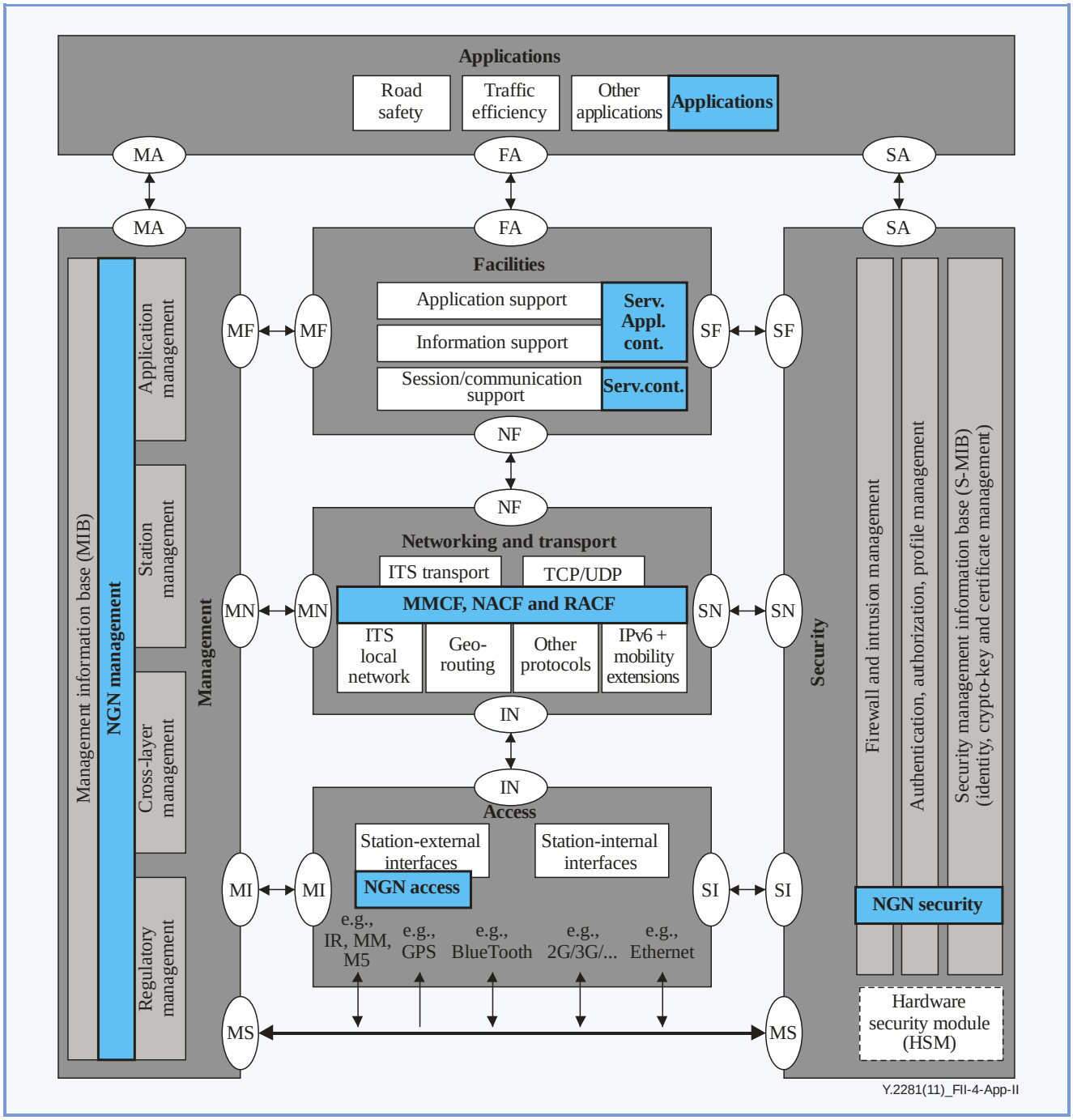


Figure II.4 – Arrangement of the relevant NGN functions into the ITS station reference architecture

Bibliography

[b-ITU-T H-Sup.1] ITU-T H-series Recommendations – Supplement 1 (2009), *Application profile – Sign language and lip-reading real-time conversation using low bit rate video communication*.

[b-ITU-T Technical Paper] ITU-T Technical Paper (2006), *Telecommunications Accessibility Checklist*.

[b-ITU-T Y.2291] Recommendation ITU-T Y.2291 (2011), *Architectural overview of next generation home networks*.

[b-ITU-R Handbook] ITU-R Handbook (2006), *Land Mobile (including Wireless Access) – Volume 4: Intelligent Transport Systems*, 109 p, http://www.itu.int/pub/R-HDB-49-2006/en.

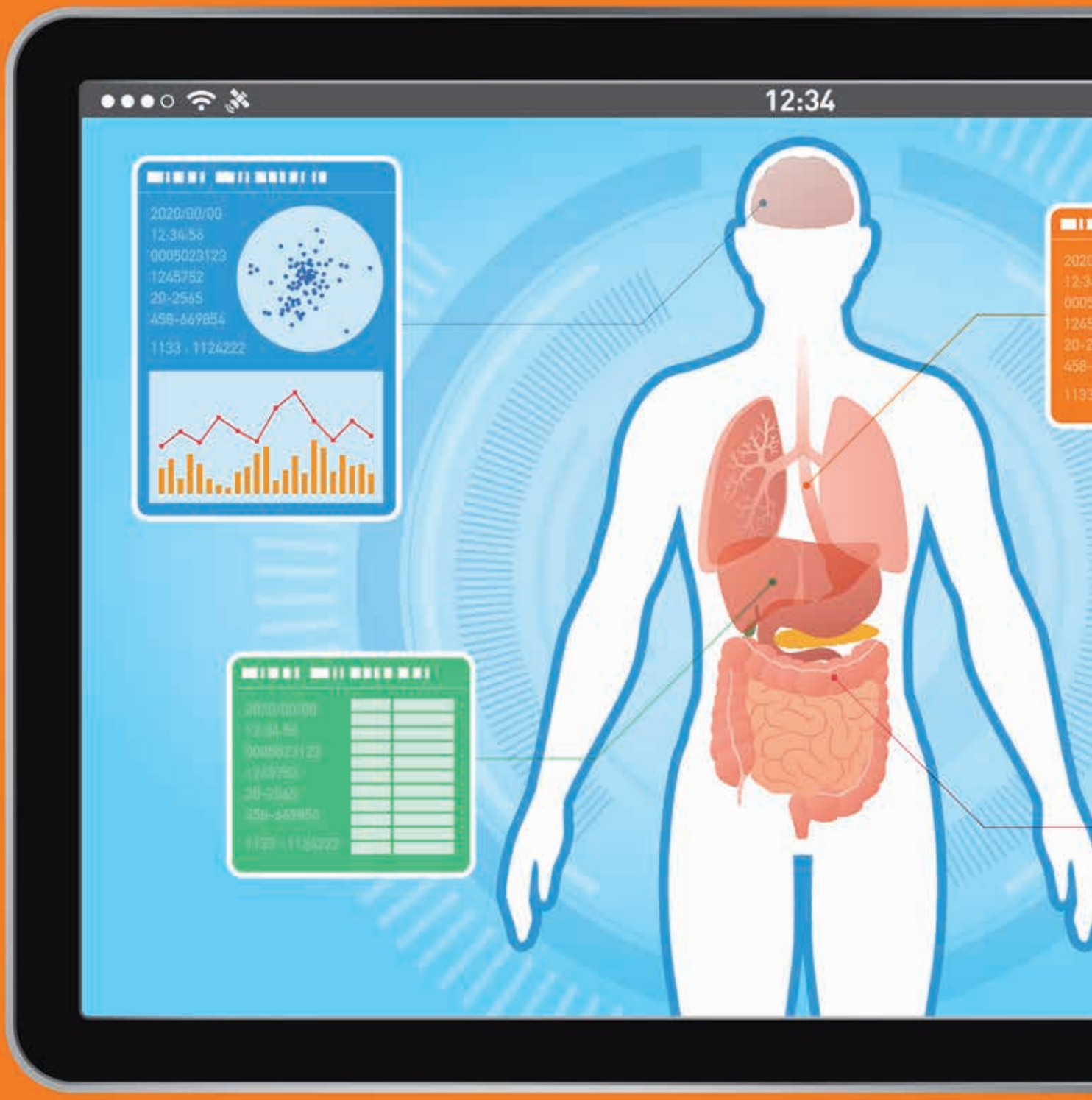
[b-ITU-R M.1452-1] Recommendation ITU-R M.1452-1 (2009), *Millimetre wave radiocommunication systems for intelligent transport system applications*.

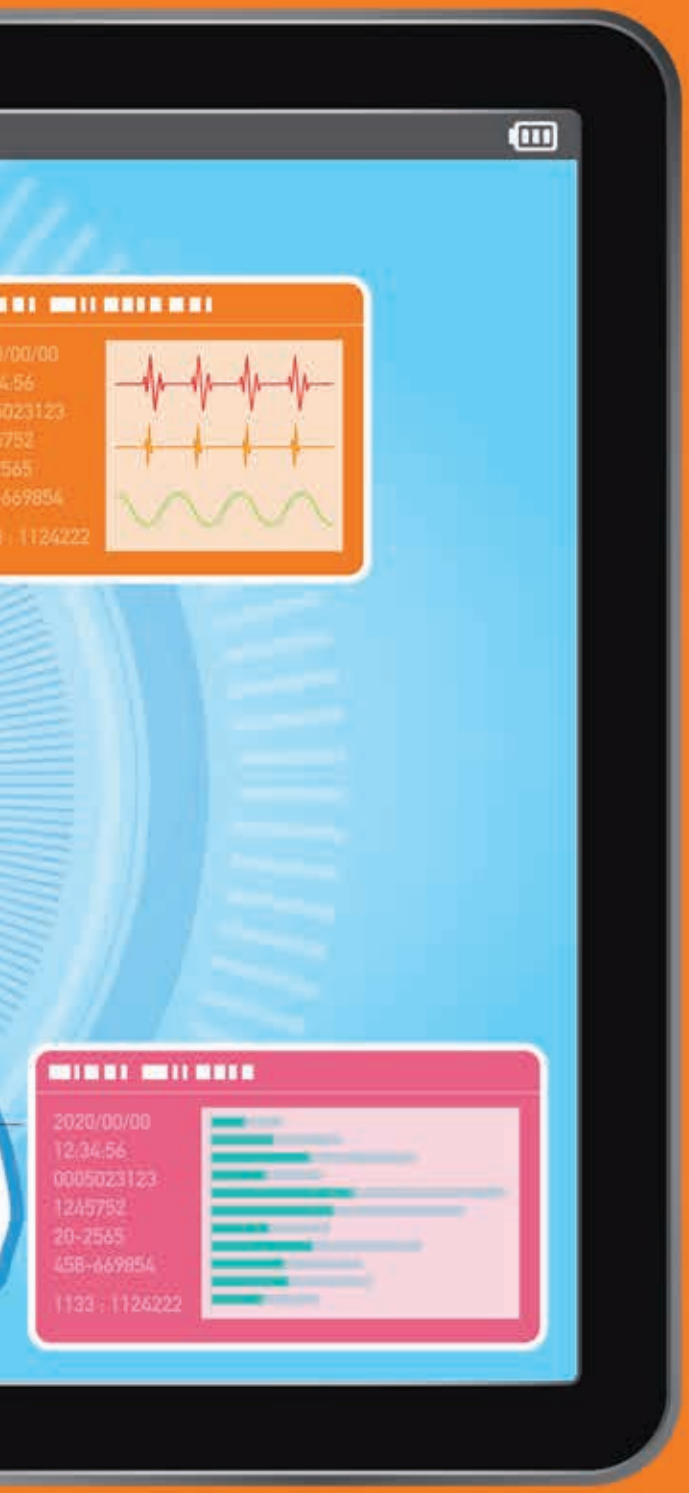
[b-ETSI TR 102 638] ETSI TR 102 638 V1.1.1. (2009), *Intelligent Transport Systems (ITS); Vehicular communications; Basic Set of Applications; Definitions*.

[b-ETSI TS 102 636] ETSI TS 102 636 (in force), *Intelligent Transport Systems (ITS); Vehicular communications; Geonetworking*.

[b-ISO 21217] ISO 21217:2010, *Intelligent transport systems – Communications access for land mobiles (CALM) – Architecture*.

[b-Intelligent] *Intelligent Transport Systems Standards* (2008), Artech House.


12:34



Y.4408/Y.2075

Capability framework for e-health monitoring services

Capability framework for e-health monitoring services

Summary

Recommendation ITU-T Y.2075 specifies the capability framework for support of the requirements of e-health monitoring (EHM) services (Recommendation ITU-T Y.2065).

To facilitate the identification of the capabilities in support of EHM services, an EHM conceptual framework is provided making usage of five components (i.e., EHM terminal, EHM end point, EHM gateway, Internet of things (IoT) platform and EHM application server) and the relationships among these components.

Based on the EHM conceptual framework and the requirements specified in Recommendation ITU-T Y.2065, the EHM capability framework is provided including the per-layer and cross-layer distribution of EHM capabilities in the five EHM components.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T Y.2075	2015-09-29	13	11.1002/1000/12582

Keywords

Capability framework, capability requirements, e-health monitoring, EHM, Internet of Things, IoT.

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

Table of Contents

CLINICAL CARE

MEDICAL DIAGNOSIS

PREVENTION

TELEMEDICINE

Recommendation ITU-T Y.4408/Y.2075

Capability framework for e-health monitoring services

1 Scope

This Recommendation specifies the capability framework for support of the requirements of e-health monitoring (EHM) services [ITU-T Y.2065].

The scope of this Recommendation includes:

– EHM conceptual framework

– EHM capability framework

An overview of the EHM capabilities in the various EHM components is provided in Annex A.

Two EHM service deployment technical scenarios are described in Appendix I.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T Y.2060] Recommendation ITU-T Y.2060 (2012), *Overview of the Internet of things*.

[ITU-T Y.2065] Recommendation ITU-T Y.2065 (2014), *Service and capability requirements for e-health monitoring services*.

[ITU-T Y.2067] Recommendation ITU-T Y.2067 (2014), *Common requirements and capabilities of a gateway for Internet of things applications*.

[ITU-T Y.2068] Recommendation ITU-T Y.2068 (2015), *Functional framework and capabilities of the Internet of things*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 e-health monitoring (EHM) service [ITU-T Y.2065]: A service which consists of observing and recording information based on [the collection of] a customer's physiological data, environmental data and other data, with the aim of monitoring the customer's state of health through the use of information and communication technologies.

3.1.2 EHM device [ITU-T Y.2065]: A device, as defined in [ITU-T Y.2060], which has sufficient qualification for e-health monitoring (EHM) service provisioning.

3.1.3 EHM end point [ITU-T Y.2065]: An e-health monitoring (EHM) device connected to the communication network through gateway(s).

3.1.4 EHM terminal [ITU-T Y.2065]: An e-health monitoring (EHM) device directly connected to the communication network.

3.1.5 Internet of things (IoT) [ITU-T Y.2060]: A global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies.

NOTE 1 – Through the exploitation of identification, data capture, processing and communication capabilities, the IoT makes full use of things to offer services to all kinds of applications, whilst ensuring that security and privacy requirements are fulfilled.

NOTE 2 – From a broader perspective, the IoT can be perceived as a vision with technological and societal implications.

3.2 Terms defined in this Recommendation

None.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

EHM E-Health Monitoring

EHMH E-Health Monitoring Healthcare

EHMR E-Health Monitoring Rehabilitation

EHMT E-Health Monitoring Treatment

ICT Information and Communication Technology

IoT Internet of Things

QoS Quality of Service

SSAS Service Support and Application Support

5 Conventions

In this Recommendation:

The keywords "is required to" indicate a requirement that must be strictly followed and from which no deviation is permitted if conformance to this document is to be claimed.

The keywords "is recommended" indicate a requirement that is recommended, but which is not absolutely required. Thus this requirement need not be present to claim conformance.

The keywords "can optionally" and "may" indicate an optional requirement that is permissible, without implying any sense of being recommended. These terms are not intended to imply that the vendor's implementation must provide the option and the feature can be optionally enabled by the network operator or service provider. Rather, it means the vendor may optionally provide the feature and still claim conformance with the specification.

6 Introduction

E-health monitoring (EHM) services include three types of services, i.e., EHM treatment (EHMT), EHM rehabilitation (EHMR) and EHM healthcare (EHMH), covering the person in healthy state, sub-healthy state and illness state respectively [ITU-T Y.2065]. It is expected that moving from one of these three types of EHM service to another is transparent from the customer's point of view, i.e., the customer can benefit from the EHM service required according to their change in state of health while minimizing the impact on the customer from the information and communication technology (ICT) point of view.

In practice, the ICT solutions adopted by EHM services vary among different scenarios and use cases. For example, as shown in Appendix I, an EHM service in community scenarios generally uses fixed devices and wired network technologies for network access, while in mobile scenarios, an EHM service uses mobile devices and wireless network technologies for network access. Actually, for real deployment scenarios, the situation is possibly even more complex than that described in Appendix I. Standardization is one way to reduce the negative impact of ICT heterogeneity on customers.

High-level standardization approaches with respect to EHM services, technical details related to specific devices, gateways, platforms and application servers used in the large variety of EHM service deployments lie outside the scope of this Recommendation. In this perspective, this Recommendation uses the concept of "component" on behalf of this large variety of specific EHM devices, gateways, platforms and application servers, and focuses on general characteristics of EHM devices, gateways, platforms and application servers.

Five types of component are identified in this Recommendation: EHM terminal, EHM end point, EHM gateway, Internet of things (IoT) platform and EHM application server.

For each component, "layers" as specified by the IoT reference model [ITU-T Y.2060] are introduced as an abstraction to illuminate the functionalities of the component.

The capabilities of a layer represent at high level the layer's functions.

NOTE – These functions are represented as functional entities at a detailed functional level.

A given component includes at least the capabilities of one of the layers as shown in Figure 1.

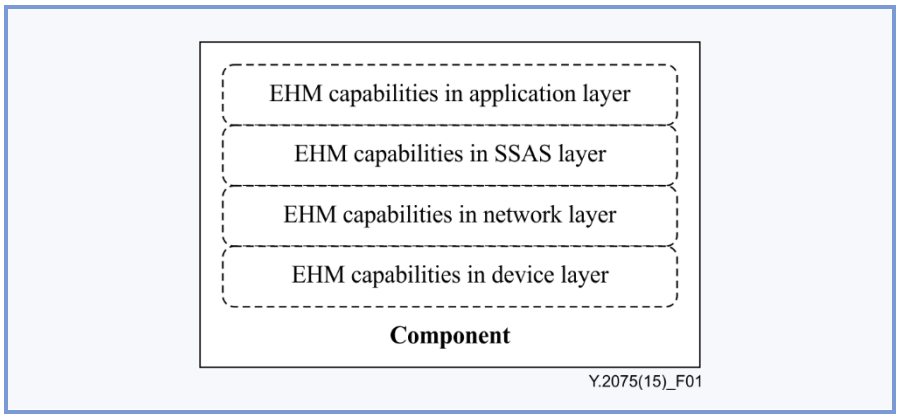


Figure 1 – Capabilities and layers in a component

Clauses 7 and 8 introduce the EHM conceptual framework and the EHM capability framework, respectively.

7 EHM conceptual framework

The EHM conceptual framework is used to identify functions and relationships of the different EHM components in order to support the EHM capability framework.

7.1 EHM conceptual framework overview

The EHM conceptual framework, shown in Figure 2, exhibits the components for the deployment of EHM services: EHM end point [ITU-T Y.2065], EHM terminal [ITU-T Y.2065], EHM gateway, IoT platform and EHM application server.

The EHM end point is an EHM device connected with the communication network through gateway(s).

The EHM terminal is an EHM device directly connected with the communication network.

The EHM gateway is a kind of gateway [ITU-T Y.2067] that provides support to the EHM end points to access the communication network.

The IoT platform is a technical infrastructure that provides generic support capabilities and specific support capabilities for EHM devices and EHM application server(s).

The EHM application server is a kind of server that runs EHM applications in order to provide EHM services.

NOTE – In Figure 2, the EHM components are represented as rectangles.

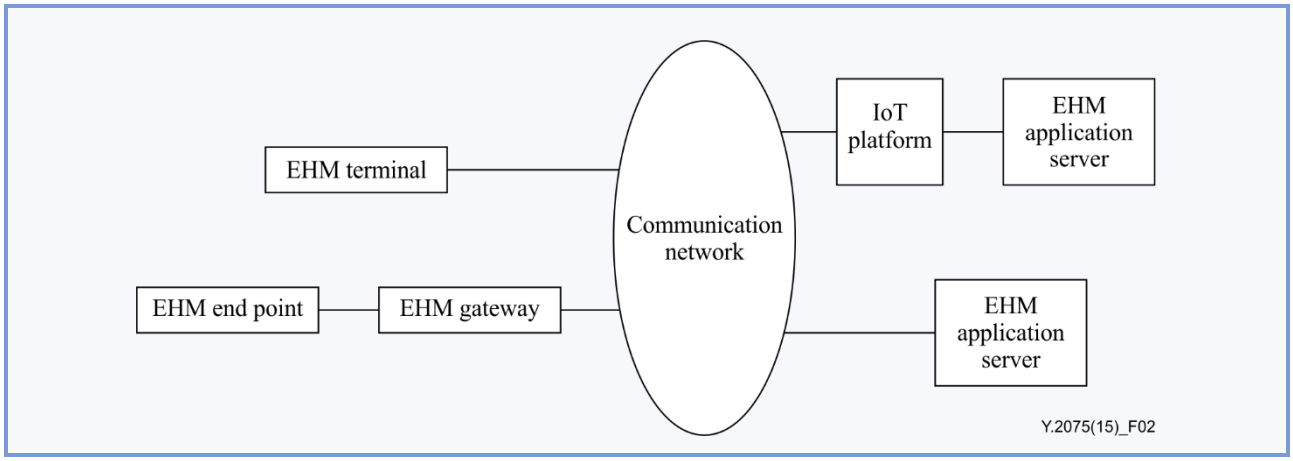


Figure 2 – EHM conceptual framework

7.2 EHM components

7.2.1 EHM terminal

The EHM terminal is defined as an EHM device directly connected to the communication network [ITU-T Y.2065]. The characteristics of an EHM terminal are given according to the following aspects:

7.2.1.1 Functions

The functions of the EHM terminal include:

(F-ET-1) EHM data sensing and collection:

The EHM terminal can sense and collect the EHM data from the bodies of EHM customers or their EHM environment.

(F-ET-2) EHM application running and management support

EHM applications can run on the EHM terminal, and the EHM terminal can provide support for the management of applications by both EHM customers and EHM application server.

(F-ET-3) Connection to the communication network

The EHM terminal can connect to and communicate with other components via the communication network.

(E-ET-4) Interworking with IoT platform and EHM application servers

The EHM terminal can interwork with the IoT platform and EHM application server, and handle operations and data exchanges with the IoT platform and EHM application server.

7.2.1.2 Relationships with other components

The EHM terminal can access the EHM application server and IoT platform via the communication network. It can access the EHM application server directly or via the help of the IoT platform.

7.2.2 EHM end point

The EHM end point is defined as an EHM device connected with the communication network via gateway(s) [ITU-T Y.2065]. The characteristics of an EHM end point are given according to the following aspects:

7.2.2.1 Functions

The functions of the EHM end point include:

(F-EEP-1) EHM data sensing and collection:

The EHM end point can sense and collect EHM data from EHM customers or related environment.

(F-EEP-2) EHM application running

The EHM end point can support running of EHM applications according to the EHM end point's usage purpose.

(F-EEP-3) Connection to EHM gateway

The EHM end point can connect to the EHM gateway via local wireless or wired communication technologies.

7.2.2.2 Relationships with other components

The EHM end point can directly connect to and access the communication network via the EHM gateway. The EHM end point can also access the IoT platform and EHM application server via the EHM gateway and the communication network.

7.2.3 EHM gateway

The characteristics of an EHM gateway are given according to the following aspects:

7.2.3.1 Functions

The functions of the EHM gateway include:

(F-EG-1) Gateway application running

Gateway applications can run on the EHM gateway.

(F-EG-2) Connection to the communication network

The EHM gateway can connect to and can communicate with other components via the communication network.

(F-EG-3) Connection to EHM end points

The EHM gateway can connect to EHM end points via local wireless or wired communication technologies.

(F-EG-4) EHM end point connection support

The EHM gateway can provide support for the connection of EHM end points to the communication network.

(F-EG-5) EHM service support

The EHM gateway can support the interworking of EHM end points with the IoT platform and EHM application server. Also, it can handle operations and data exchanges with the IoT platform and EHM application server.

7.2.3.2 Relationships with other components

The EHM gateway can locally connect to the EHM end points, and also can access the IoT platform and EHM application server via the communication network.

7.2.4 IoT platform

The IoT platform is a platform that can provide service support and application support capabilities for EHM devices and the EHM application server. The characteristics of the IoT platform are given according to the following aspects:

7.2.4.1 Functions

The functions of the IoT platform include:

(F-MP-1) Application running

Applications can run on the IoT platform.

(F-MP-2) Connection to the communication network

The IoT platform can connect to and communicate with other components via the communication network.

(F-MP-3) Service support and application support

The IoT platform can provide service support and application support for EHM devices and EHM applications using the service support and application support capabilities.

7.2.4.2 Relationships with other components

The IoT platform can access the EHM application server, EHM gateway and EHM terminals via the communication network.

7.2.5 EHM application server

The EHM application server is a kind of server that runs EHM applications. The characteristics of the EHM application server are given according to the following aspects:

7.2.5.1 Functions

The functions of the EHM application server include:

(F-EAS-1) EHM application running and management support

EHM applications can run on the EHM application server, and the EHM application server can provide support for their management.

(F-EAS-2) Connection to the communication network

The EHM application server can connect to and communicate with other components via the communication network.

(F-EAS-3) EHM service support

The EHM application server can interwork with the IoT platform, EHM gateway and EHM terminals. It can also handle operations and data exchanges with the IoT platform, EHM gateway and EHM terminals.

7.2.5.2 Relationships with other components

The EHM application server can connect to and access the IoT platform, EHM gateway and EHM terminals via the communication network.

8 EHM capability framework

The EHM capability framework maps the layered EHM capabilities identified in [ITU-T Y.2065] to each EHM component as specified in the EHM conceptual framework.

8.1 Distribution map of the EHM capabilities

The EHM essential IoT capabilities, based on the requirements identified in [ITU-T Y.2065], are specified in this Recommendation.

NOTE – IoT basic capabilities and IoT capabilities for integration of key emerging technologies, as specified in [ITU-T Y.2068], are also applicable to the support of EHM services.

According to the EHM capability requirements [ITU-T Y.2065] and based on the components of the EHM conceptual framework, the distribution of the EHM capabilities in the four layers of the IoT reference model [ITU-T Y.2060] is shown in Figure 3. As all EHM components have cross-layer EHM management and security capabilities, these cross-layer capabilities are not shown in Figure 3, the purpose of Figure 3 being to show the differences from a capability viewpoint among the EHM components.

In Figure 3, the boxes represent the components specified in the EHM conceptual framework, the dashed rounded rectangles represent the four layers of [ITU-T Y.2060] and the circles represent the presence of EHM capabilities at a given layer in a given component.

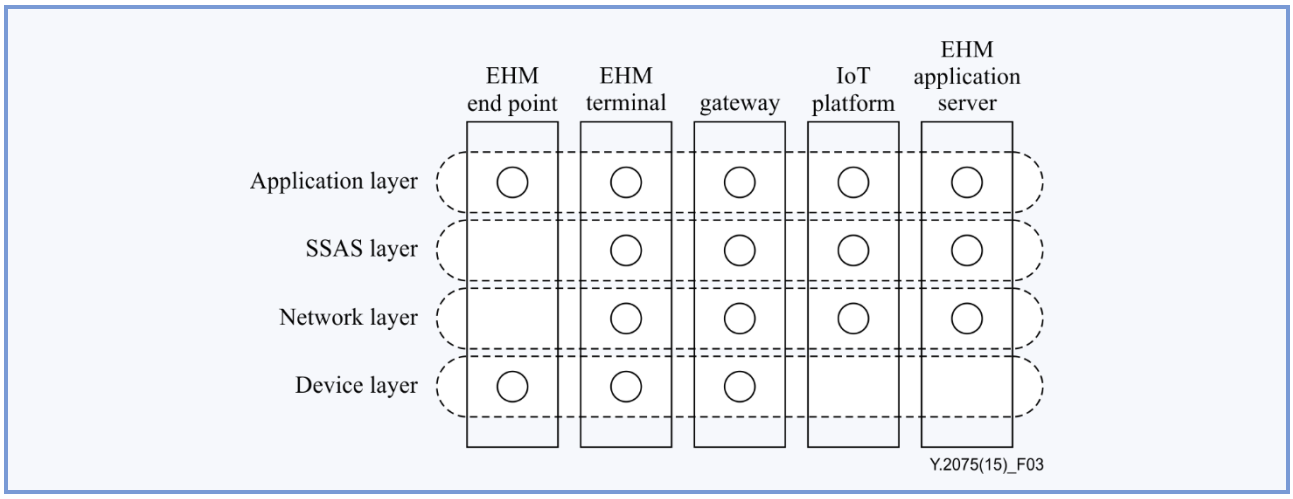


Figure 3 – Distribution map of EHM capabilities

In the EHM end point, the EHM capabilities are distributed in the device layer and application layer, as well as cross-layer EHM management and security capabilities.

In the EHM terminal, the EHM capabilities are distributed in the device layer, network layer, service support and application support layer and application layer, as well as cross-layer EHM management and security capabilities.

In the gateway, the EHM capabilities are distributed in the device layer, network layer, service support and application support layer, application layer, as well as cross-layer EHM management and security capabilities.

On the IoT platform, the EHM capabilities are distributed in the network layer, service support and application support layer, application layer, as well as cross-layer EHM management and security capabilities.

In the EHM application server, the EHM capabilities are distributed in a network layer, service support and application support layer, application layer, as well as cross-layer EHM management and security capabilities.

Clauses 8.2 to 8.7 provide details of this distribution map.

Annex A provides an overview of EHM capabilities, summarizing their distribution in the different EHM components.

8.2 Application layer capabilities of EHM components

8.2.1 Information sharing

According to the capability requirements described in clause 9.2.1 of [ITU-T Y.2065], the application layer of the components is recommended for support of standard interfaces and policy-based mechanisms [ITU-T Y.2065] to enable the sharing of EHM information among EHM components.

Table 1 shows the EHM components involved in the information sharing capability of application layer: the first column indicates the EHM components providing the EHM information, the first row indicates the EHM components receiving the EHM information.

Table 1 – EHM components involved in the information sharing capability of application layer

Provider	Receiver				
	EHM end point	EHM terminal	EHM gateway	IoT platform	EHM application server
EHM end point	✓		✓		
EHM terminal		✓	✓	✓	✓
EHM gateway		✓	✓	✓	✓
IoT platform		✓	✓	✓	✓
EHM application server		✓	✓	✓	✓

8.2.2 Accounting-related information provision

According to the capability requirements described in clause 9.2.2 of [ITU-T Y.2065], some EHM components are recommended for provision of the capability for accounting-related information provision.

The specific requirements of the accounting-related information provision capability of application layer for each EHM component are described in Table 2.

Table 2 – EHM components and accounting-related information provision capability of the application layer

EHM components	Accounting-related information provision
EHM end point	The application layer of the EHM end point is recommended to report accounting-related information, including, but not limited to, service class (i.e., EHMH or EHMR or EHMT), number of times and duration time of EHM service usage.
EHM terminal	The application layer of the EHM terminal is recommended for the reporting of accounting-related information to the service support and application support (SSAS) layer of the EHM terminal. This information includes, but is not limited to, service class (i.e., EHMH or EHMR or EHMT), number of times and duration time of EHM service usage.
EHM gateway	The application layer of the EHM gateway is recommended to report accounting-related information about the connected EHM end points to the SSAS layer of the EHM gateway. This information includes, but is not limited to, service type (EHMH, EHMR and EHMT), number of duration times and duration time of application usage.
IoT platform	None.
EHM application server	The application layer of the EHM application server is recommended to report accounting-related information to the SSAS layer of the EHM application server. This information includes, but is not limited to, number of times and duration time of EHM service usage.

8.2.3 QoS related information provision

According to the capability requirements described in clause 9.2.3 of [ITU-T Y.2065], the application layer of the EHM end point, EHM terminal, EHM gateway and EHM application server is recommended to provide Quality of Service (QoS) information to the SSAS layer or network layer for QoS configuration purposes.

NOTE – Refer to clause 9.2.3 of [ITU-T Y.2065] for the recommended QoS parameters.

8.3 SSAS layer capabilities of EHM components

8.3.1 Message conversion

According to the capability requirements described in clause 9.3.2 of [ITU-T Y.2065], the SSAS layer of the EHM terminal, EHM gateway, IoT platform and EHM application server is required to provide message conversion capability at the syntactic or semantic level.

NOTE – "Message" is intended here as the carrier of information transmitted between EHM technical components.

8.3.2 Data storage

According to the capability requirements described in clause 9.3.3 of [ITU-T Y.2065], the SSAS layer of the EHM terminal, EHM gateway, IoT platform and EHM application server is required to provide data storage.

The data stored in the SSAS layer are recommended to be stored in standard format so that the information can be easily exchanged among different EHM application servers, gateways and EHM terminals.

The specific requirements for data storage capability of the SSAS layer for each EHM component are described in Table 3.

Table 3 – EHM components and data storage capability of SSAS layer

EHM components	Data storage
EHM end point	None
EHM terminal	The EHM application data stored in the SSAS layer of the EHM terminal and EHM gateway are required to be marked with collection time.
EHM gateway	
IoT platform	The EHM application data stored in the SSAS layer of the IoT platform and EHM application server are required to be associated with time information, e.g., collection time and expiration time.
EHM application server	

8.3.3 Time synchronization

According to the capability requirements described in clause 9.3.4 of [ITU-T Y.2065], the SSAS layer of the EHM terminal, EHM gateway, IoT platform and EHM application server is required to be able to retrieve time parameters from authoritative time servers or via other ways.

The SSAS layer of the IoT platform is required to be able to publish the time parameters to the EHM gateway, EHM terminals and EHM application servers. It is recommended that the time parameters be published periodically.

8.3.4 Location provisioning

According to the capability requirements described in clause 9.3.5 of [ITU-T Y.2065], the SSAS layer of the EHM terminal and EHM gateway is required to collect the location information from the network layer or device layer according to the collection strategy, such as event-triggered collection or periodic collection.

According to the capability requirements described in clause 9.3.5 of [ITU-T Y.2065], the SSAS layer of the EHM terminal and EHM gateway is required to report the location information required by the application layer in standard format.

According to the capability requirements described in clause 9.3.5 of [ITU-T Y.2065], the SSAS layer of the IoT platform and EHM application server is required to be able to collect the position of EHM terminals and gateways according to the collection strategy, such as event-triggered collection.

8.3.5 Service accounting and charging

According to the capability requirements described in clause 9.3.1 of [ITU-T Y.2065], the SSAS layer of the IoT platform is required to gather data about the usage of EHM services for charging purposes. Different policies may be considered for service accounting and charging, e.g., the number of times the service is used, the amount of time the service is used or the volume of used service data.

The service accounting and charging capability in the SSAS layer of the IoT platform has the following requirements:

1) It is required that service accounting and charging be provided to EHM service users.
2) It is recommended that service accounting and charging be provided according to the QoS of EHM services.
3) It is recommended that service accounting and charging also be provided in support of roaming scenarios for EHM services among networks owned by different network providers.
4) It is recommended that service accounting and charging be provided according to the frequency of access to EHM services.
5) As a user may use several EHM devices at the same time, it is recommended that unified service charging per user be supported.

8.4 Network layer capabilities of EHM components

8.4.1 Policy-based communication

According to the capabilities requirements described in clause 9.4.1 of [ITU-T Y.2065], the network layer of the EHM terminal, EHM gateway, IoT platform and EHM application server is required to support policy-based communication capability and is required to be able to set the network policy in order to support the QoS of EHM services.

8.4.2 Network-based locating

According to the capability requirements described in clause 9.4.2 of [ITU-T Y.2065], the EHM capabilities on the network layer are recommended to provide the location-related information from the network layer (e.g., IP address, access point location, and so on) to locate the position of EHM terminals and EHM gateways.

Specifically, the network layer of the EHM terminal and EHM gateway is recommended to provide the location-related information. In addition, the network layer of the EHM terminal and EHM gateway is recommended to support event-triggered location information notification. In this way, when the EHM terminal or EHM gateway moves out of a preconfigured network area, a network location information notification may be triggered by the event.

8.4.3 Network resource provision

According to the capability requirements described in clause 9.4.3 of [ITU-T Y.2065] , the EHM capabilities on the network layer are required to provide the network resources (e.g., network address, network bandwidth) to establish network connectivity.

Specifically, the network layer of the EHM terminal, EHM gateway, IoT platform and EHM application server is required to be able to provide the network resources (e.g., network address, network bandwidth) to establish network connectivity. The network layer of each of these components is recommended to inform the other EHM components about the provided network resources (e.g., network address, network bandwidth).

8.5 Device layer capabilities of EHM components

8.5.1 Device identification

According to the capability requirements described in clause 9.5.1 of [ITU-T Y.2065], the EHM capabilities on the device layer of the EHM terminal and EHM end point are required to provide device profiles in order to identify the intended use of EHM devices, such as for support of EHMH, EHMR or EHMT services.

8.5.2 Data sensing and processing

According to the capability requirements described in clause 9.5.3 of [ITU-T Y.2065], the EHM capabilities on the device layer of the EHM terminal and EHM end point are required to support data sensing and processing capability to obtain the EHM data.

This capability includes:

1) Data sensing

   Data sensing is used to obtain the raw EHM data according to regulation and laws.

2) Data processing

   Data processing is used to process the raw EHM data, such as filtering, aggregating and compressing, in order to improve the quality and usage efficiency of EHM data.

The device layer of the EHM terminal is recommended to support data sensing for multiple EHM parameters in a single EHM terminal.

8.5.3 Data collection time provision

According to the capability requirements described in clause 9.5.4 of [ITU-T Y.2065] , the EHM capabilities on the device layer of the EHM terminal and EHM gateway are recommended to support data collection time provision, so that the collected EHM data can be marked with the collection time.

This capability includes:

1) Time calibration

   The time calibration capability is used to obtain time parameters from the SSAS layer and calibrate the built-in time clock of EHM devices.

2) Time provision

   The time provision capability is used to provide calibrated collection time along with the collected EHM data.

8.5.4 Device-based locating

According to the capability requirements described in clause 9.5.5 of [ITU-T Y.2065], the EHM capabilities on the device layer of the EHM terminal and EHM gateway are recommended to support device-based locating capability.

NOTE – Different techniques (e.g., GPS, gyroscope and motion state sensor) can be used to get the position of EHM terminals or EHM gateways.

It is recommended that location accuracy be indicated along with the location information.

8.5.5 Gateway

According to the capability requirements described in clause 9.5.2 of [ITU-T Y.2065], the device layer of the EHM gateway is required to provide gateway capabilities [ITU-T Y.2067] for connected EHM end points or EHM terminals, e.g., network adaptation capability and raw data processing capability.

8.6 Management capabilities of EHM components

According to the capability requirements described in clause 9.6 of [ITU-T Y.2065], the EHM components are required to support the following management capabilities:

1) fault management

   The EHM gateway and IoT platform are required to recognize, isolate, correct and log faults. The EHM end point, EHM terminal and EHM application server are required to recognize, correct and log faults.

2) configuration management

   The EHM terminal and EHM end point are required to be able to be configured remotely. The IoT platform and EHM application server are required to support remote device configuration. The EHM gateway is required to be able to configure the connected EHM end points according to configuration requests from the IoT platform or EHM application server.

3) initialization and registration management

   The EHM terminal is required to support initialization set-up and registration capability. The EHM end point is required to complete the initialization set-up and registration procedure by itself or with the help of the EHM gateway. The EHM gateway is required to support the initialization set-up and registration capability to help EHM end points, where necessary, in order to complete the initialization set-up and registration.

8.7 Secure capabilities of EHM components

According to the capability requirements described in clause 9.7 of [ITU-T Y.2065], the EHM components are required to support the following security capabilities:

1) Authentication and authorization

The EHM gateway, EHM terminal, IoT platform and EHM application server are required to support authentication and authorization mechanisms.

The EHM gateway and IoT platform are required to support authentication and authorization for EHM devices and EHM application servers.

The EHM devices and EHM application servers are recommended to support a mutual authentication and authorization when accessing the EHM gateway or IoT platform.

The authentication and authorization mechanisms among the EHM gateway, IoT platform, EHM devices and EHM application servers can be based on network level authentication mechanisms (e.g., IP based or SIM card based) or application level authentication mechanisms (e.g., certificate based or account and password based).

The authentication and authorization mechanisms between EHM devices and the EHM gateway can be additionally based on authentication mechanisms (e.g., Bluetooth based, wired connection based) of the local network among EHM devices and gateways.

The EHM terminal and EHM application server are required to support authentication and authorization for EHM customer access.

NOTE – EHM gateways that support EHM customer access are also required to provide authentication and authorization for accessing users.

2) Secure communications

All EHM components are required to support secure communications. The EHM application, EHM terminal and IoT platform are required to support secure communications through the communication network. The EHM end point is required to support secure communications through the local network among EHM devices and gateways. The EHM gateway is required to support secure communications through both the communication network and the local network among EHM devices and EHM gateways.

3) Confidentiality

All EHM components are required to enforce the confidentiality of the data whenever the data are exchanged, stored or processed.

4) Integrity

All EHM components are required to guarantee the integrity of data when the data are transmitted. Any loss of integrity of the transmitted data must be recognizable by the receiving components.

5) Access control

The EHM gateway, EHM terminal, IoT platform and EHM application server are required to ensure that only authorized EHM components are able to access protected data and only authorized users can access the EHM components.

6) Audit trail

The EHM gateway, EHM terminal, IoT platform and EHM application server are required to trace and record any access or attempt to access EHM data.

7) Data storage security

All EHM components that support data storage are required to support data integrity validation and data privacy protection. In addition, the IoT platform and EHM application server are required to support data backup, anti-hacker data protection, uninterruptible power of data storage and data recovery.

Annex A

Overview of EHM component capabilities

(This annex forms an integral part of this Recommendation.)

Table A.1 summarizes the capabilities of the different EHM components and assigns the following indicators:

- "MA" means this capability is required in the EHM component.
- "REC" means this capability is recommended in the EHM component.
- "NA" means this capability is not supported by the EHM component.

Table A.1 – Overview of EHM component capabilities

Capabilities		EHM component				
		EHM end point	EHM terminal	EHM gateway	IoT platform	EHM application server
Application layer capabilities	Information sharing	MA	MA	MA	MA	MA
	Accounting-related information provision	REC	REC	REC	NA	REC
	QoS information provision	REC	REC	REC	NA	REC
SSAS layer capabilities	Message conversion	NA	MA	MA	MA	MA
	Data storage	NA	MA	MA	MA	MA
	Time synchronization	NA	MA	MA	MA	MA
	Location provisioning	NA	MA	MA	MA	MA
	Service accounting and charging	NA	NA	NA	MA	NA
Network layer capabilities	Policy-based communication	NA	MA	MA	MA	MA
	Network-based locating	NA	REC	REC	NA	NA
	Network resource provision	NA	REC	REC	REC	REC
Device layer capabilities	Device identification	MA	MA	NA	NA	NA
	Data sensing and processing	MA	MA	NA	NA	NA
	Data collection time provision	NA	REC	REC	NA	NA
	Device-based locating	NA	REC	REC	NA	NA
	Gateway	NA	NA	MA	NA	NA

Table A.1 – Overview of EHM component capabilities

Capabilities		EHM component				
		EHM end point	EHM terminal	EHM gateway	IoT platform	EHM application server
Management capabilities	Fault management	NA	MA	MA	MA	MA
	Configuration management	MA	MA	MA	MA	MA
	Initialization and registration management	MA	MA	MA	NA	NA
Secure capabilities	Authentication and authorization	NA	MA	MA	MA	MA
	Security communications	MA	MA	MA	MA	MA
	Confidentiality	MA	MA	MA	MA	MA
	Integrity	MA	MA	MA	MA	MA
	Access control	NA	MA	MA	MA	MA
	Audit trail	NA	MA	MA	MA	MA
	Data storage security	NA	MA	MA	MA	MA

Appendix I

EHM service deployment technical scenarios

(This appendix does not form an integral part of this Recommendation.)

This appendix gives examples of EHM service deployment technical scenarios involving EHM components.

I.1 Technical scenario for community EHM services

In community environment, EHM services are provided by qualified facilities and serve all inhabitants of a community. Figure I.1 shows a deployment case for community EHM service.

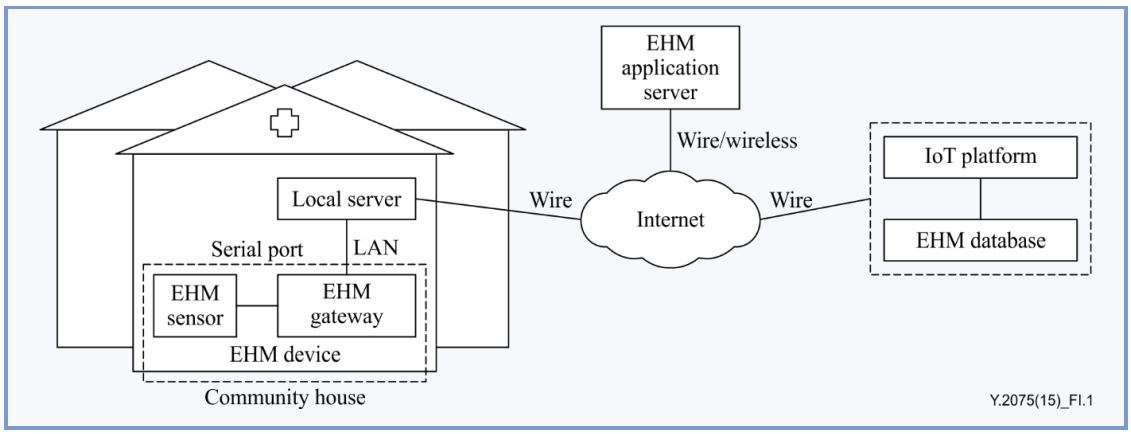


Figure I.1 – EHM services deployed in community environment

As shown in Figure I.1, the EHM devices, including EHM sensors and EHM gateways, are deployed in the community house. Usually, there are several EHM sensors, such as ECG, blood pressure meter, weighing scales, thermometer and densitometer, with which general practitioners can check the health states of inhabitants. In the current market, most EHM sensors just have limited communication capabilities, e.g., a lot of EHM sensors just have serial port interfaces in order to reduce sensor cost.

To obtain extended communication capabilities, EHM sensors connect to EHM gateways to communicate and interwork with the local server(s), as shown in Figure I.1. The EHM gateways provide communication protocol translation between the EHM sensors and local server. The southbound ports of the EHM gateways usually use point-to-point short-distance communication protocols (e.g., serial port and USB); the northbound ports of the EHM gateways usually use the Ethernet protocol. Multiple EHM gateways can be connected to the local server via star topology.

The local server, as shown in Figure I.1, has four main functions: firstly, it manages the EHM gateways and EHM sensors in the community house; secondly, it analyses and stores the EHM data sent by the EHM sensors; thirdly, it reports the local EHM data to the EHM application server and IoT platform; fourth, it receives notification from the IoT platform to update the local software.

The EHM application server, as shown in Figure I.1, has two main functions: firstly, it analyses and storages the EHM data sent by the local server in the community house; secondly, it can share EHM data, analysis results and computing capability with other EHM application servers.

The IoT platform, as shown in Figure I.1, provides capabilities to facilitate reliability, security, efficient interworking between the EHM application server and the local server in the community house. Usually the IoT platform connects with an EHM database which stores and manages EHM data.

I.2 Technical scenario for mobile EHM services

In the mobile EHM services scenario, users usually use a handset (e.g., mobile phone or pad) to access the EHM services via a wireless network (e.g., cellular network or carrier-WiFi network).

The technical deployment for this scenario shown in Figure I.2 involves such entities as sensor, handset, network, EHM application server, IoT platform and EHM database. Solid boxes represent mandatory entities in this scenario. The dashed boxes represent an optional entity, and the dot dashed boxes mean that the included entities can be integrated as an entity.

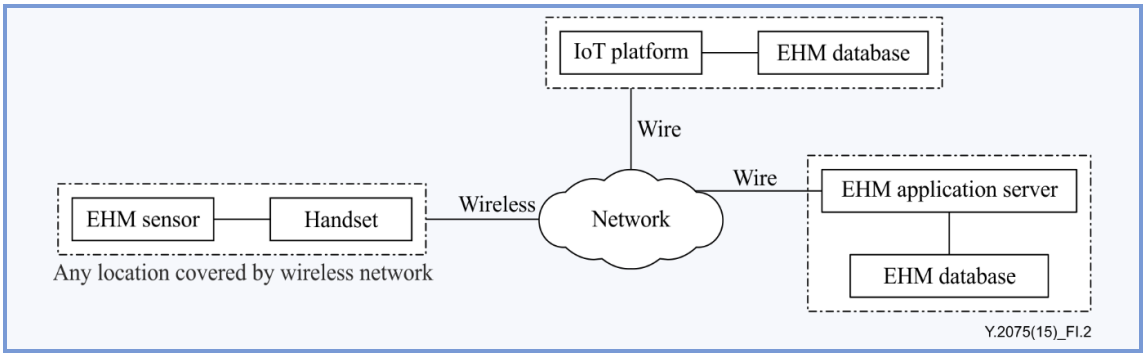


Figure I.2 – Mobile EHM services

1) EHM sensor
The EHM sensor can be seen as a type of EHM end point, which can sense EHM-related data from the body of a user, but cannot access the communication network directly. There are two methods for the deployment of a sensor. One is to deploy the EHM sensor inside the handset, which can use the communication capability of the handset via an internal interface. The other is to deploy the EHM sensor as a single device outside the handset. The sensor can access the handset via a local communication protocol (e.g., Bluetooth or USB).

2) Handset
The handset is a type of communication terminal, which can access the communication network via wireless communication. In the mobile EHM service deployment scenario, it can play the role of communication gateway and provide the EHM data collected by the EHM sensors to a remote EHM application server and IoT platform. Besides the communication capability, the handset can also process and manage the EHM data collected by EHM sensors or feedback from the EHM application server and IoT platform based on the EHM application running on them and show the results to users.

3) Network
The network in the mobile EHM service deployment scenario is a type of communication network, which can provide wireless access to the handset and also provide wired access to the EHM application server and IoT platform. It supports interconnection among the handset, EHM application server and IoT platform.

4) EHM application server
The EHM application server provides capabilities for analysing and managing the EHM data provided by the handset or IoT platform and feeds the corresponding diagnosis results back to the handset or IoT platform.
The IoT platform connects the handset with the EHM application server. It provides the service support and application support capabilities for the handset and the EHM application server. In practical deployment scenarios, multiple IoT platforms with different capability sets may be deployed to form a virtual IoT platform.

5) EHM database
The EHM database is used for EHM data storage. Both the handset and EHM application server can access the EHM database by authorization. There are three possible methods for the deployment of the EHM database. The first consists of deploying the EHM database in the IoT platform and opening secure access to the EHM database for the EHM application server and the handset. The second consists of deploying the EHM database on the EHM application server. The last is a hybrid method, consisting of deploying the EHM database both in the IoT platform and the EHM application server.

Y.4409/Y.2070

Requirements and architecture of the home energy management system and home network services

Requirements and architecture of the home energy management system and home network services

Summary

Recommendation ITU-T Y.2070 provides the requirements and architecture of the home energy management system (HEMS) and home network (HN) services. The HEMS supports energy efficiency and reduction of energy consumption by monitoring and controlling devices such as home appliances, storage batteries and sensors connected to the HN from the HEMS application.

While the algorithm for the energy efficiency and reduction of energy consumption runs in the HEMS application, the development of a platform (PF) is desired which provides common functions to enable the application to access the devices and to support the efficient development of applications. This is not only applies for the HEMS, but also for other HN services such as home security and healthcare. This Recommendation provides common requirements for the HN services to support the HEMS as the widely known HEMS is mainly considered one of the HN services. It also describes the reference architecture and the functional architecture including the functional relationship for the HEMS and the other HN services.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T Y.2070	2015-01-13	13	11.1002/1000/12420

Keywords

Home energy management system, home network, home network service.

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

Table of Contents


Consumption
5.00 KW
75%
ECO mode
ON
OFF

Recommendation ITU-T Y.4409/Y.2070

Requirements and architecture of the home energy management system and home network services

1 Scope

This Recommendation provides the requirements and architecture of the home energy management system (HEMS) and home network (HN) services. The HEMS supports energy efficiency and reduction of energy consumption by monitoring and controlling devices such as home appliances, storage batteries and sensors connected to the HN from the HEMS application with the HN service architecture. The HEMS is one of the HN services. The other HN services, such as home security and healthcare, are provided with the same architecture as the HEMS and by monitoring and controlling the devices from the application specific to the service. In this Recommendation, the requirements, the reference architecture and the functional architecture including functional relationship are described to support the HEMS and the other HN services.

This Recommendation covers the followings:

- overview of the HN service architecture for the HEMS and other HN services;
- requirements for the device, home gateway (HGW) and management platform (PF) in the HN service architecture as well as the security required for the architecture;
- reference architecture with four ways to connect to the devices from the HGW according to the device type: basic device (IP based and non-IP based) and non-basic device (connecting to the HGW directly or through the adapter);
- functional architecture with the entities: device, HGW, management PF and application;
- functional relationship with three functional categories in the functional architecture: device operation, application execution and management;
- security model and functions for the HN services mainly describing the HEMS.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T X.1111] Recommendation ITU-T X.1111 (2007), *Framework of security technologies for home network.*

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 demand response [b-FG-Smart Terminology]: A smart grid feature that allows consumers to reduce or change their electrical use patterns during peak demand, usually in exchange for a financial incentive. Mechanisms and incentives for utilities, business, industrial, and residential

customers to cut energy use during times of peak demand or when power reliability is at risk. Demand response is necessary for optimizing the balance of power supply and demand.

NOTE – Smart grid [b-FG-Smart Terminology]: A two way electric power delivery network connected to an information and control network through sensors and control devices. This supports the intelligent and efficient optimization of the power network.

3.1.2 device object [b-ECHONET Lite]: A logical model of the information held by equipment devices or home electrical appliances such as sensors, air conditioners and refrigerators, or of control items that can be remotely controlled. The interface form for remote control is standardized. The information and control target of each device is specified as property, and the operating method (setting and browsing) is specified as a service.

3.1.3 home network [b-ITU-T J.190]: A short-range communications system designed for the residential environment, in which two or more devices exchange information under some sort of standard control.

3.1.4 presence [b-ITU-T Y.2720]: A set of attributes that characterize an entity relating to the current status.

3.1.5 smart meter [b-FG-Smart Terminology]: Smart meter is a premise device to monitor and control of electrical power usage of home devices based on "demand response information" from home devices. But, it is not recommended that the smart meter controls directly per each premise appliances because of the private security policy. To control and manage the each premise appliances, it is required for home management system such as home gateway and home server to support the control and management.

3.1.6 web of things [b-ITU-T Y.2063]: A way to realize the IoT where (physical and virtual) things are connected and controlled through the world wide web.

3.1.7 web resource [b-W3C WCterms]: A resource, identified by a URI, that is a member of the web core.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 adapter: An entity used to connect a non-basic device to the home gateway by converting the dedicated communications protocol to the IP based protocol and the dedicated data model to the abstract data model.

3.2.2 device management: A variety of functionalities to manage a collection of the devices including primary capabilities: auto-configuration and dynamic service provisioning, software/firmware image management, software module management, status and performance monitoring, and diagnostics.

3.2.3 fault diagnosis: An example of a maintenance action, which is a sequence of elementary maintenance activities carried out for a given purpose.

3.2.4 home controller: A small computer for the application for the home energy management system to monitor and control the home equipment such as home appliances and storage batteries to reduce energy consumption.

3.2.5 home energy management system: A computer system comprising a software platform providing basic support services and a set of applications providing the functionality needed for the effective operation of home equipment, such as home appliances and storage batteries, so as to assure adequate security of energy supply at minimum cost.

3.2.6 home gateway: An always on, always connected device which acts as the central point connecting the devices on the home network to the applications on the wide area network, and monitors and performs actions on data flows within the home network as well as on bi-directional communication flows between the home network and the wide area network.

3.2.7 home network resource: A device (e.g., home appliance, storage battery, sensor), a network device (e.g., hub and access point in the home network) and network capacity for data transmission among them for the home network services.

3.2.8 in-home display: A user screen device to present home energy consumption information. Users can optionally control their home devices with its user interfaces.

3.2.9 managed agent: A software program running on the device to set the configuration information and to collect the information of the device. The managed agent gets the information from the resource management function on the management platform for the configuration of the device and sends the collection of the internal status of the device to it for various home network services including remote management and fault diagnosis.

3.2.10 management platform: A platform which has common functions providing the interface and the management for the home network applications, and the virtual device management and the resource management for the home gateway and the devices.

3.2.11 wide area network: An IP based communication network that covers a wide geographical area including the Internet and accommodates devices and local area networks.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

API Application Programming Interface

CPU Central Processing Unit

DB Database

DR Demand Response

EV Electric Vehicle

HEMS Home Energy Management System

HGW Home Gateway

HN Home Network

HTTP Hypertext Transfer Protocol

IHD In-Home Display

IP Internet Protocol

L2 Layer 2

LAN Local Area Network

MAC Message Authentication Code

NAT Network Address Translation

PF Platform

SOAP Simple Object Access Protocol

WAN Wide Area Network

WoT Web of Things

XML Extensible Markup Language

XMPP Extensible Messaging and Presence Protocol

5 Conventions

In this Recommendation,

- The keywords "is required to" indicate a requirement which must be strictly followed and from which no deviation is permitted if conformance to this Recommendation is to be claimed.
- The keywords "can optionally" indicate an optional requirement which is permissible, without implying any sense of being recommended. This term is not intended to imply that the vendor's implementation must provide the option and the feature can be optionally enabled by the network operator/service provider. Rather, it means the vendor may optionally provide the feature and still claim conformance with the specification.

6 Overview

This overview clause describes the HN service architecture. The HEMS is one of the HN services. There are other HN services, such as the home security service, for example, which detects a suspicious person with the human aware sensors. The HEMS and other HN services are provided with the HN service architecture. Clause 6.1 describes the HN service architecture. Clause 6.2 describes how the HEMS is provided by the HN service architecture by replacing the HN applications with the HEMS application.

With the more widespread deployment of HN services and the increase of the devices connected to the HN, it is more complicated and more difficult for application developers to develop applications for the HN; a deep knowledge about the HN devices and communications protocols is required. Therefore, the development of an architecture for the HN services to support the application developers is important and forms the background of this Recommendation.

NOTE –The term “Internet” is used in the description in clause 6 to support a clear understanding by the reader. However, it should be understood as the wide area network (WAN), which includes the Internet. The term WAN is used from clause 7 onward in this Recommendation.

6.1 HN service architecture

HN applications have been developed to run on dedicated home controllers, which are located in the home. As shown in Figure 6-1(a) individual access, every home controller connects to one or more devices (home equipment) such as home appliances and storage batteries, each of which has its own dedicated communication interface at the device interface. For this reason, each application needs to be developed to meet with the device interface of the connecting device in order to monitor and control the device.

On the other hand, as the communications protocols are standardized, devices are connecting with the standardized protocol to the common PF, which works on the home controller as shown in Figure 6-1(b). In this, common access, diagram, it is possible to abstract the device interface by the common PF and the devices can be accessed from HN applications which are also connected to the common PF at the application interface.

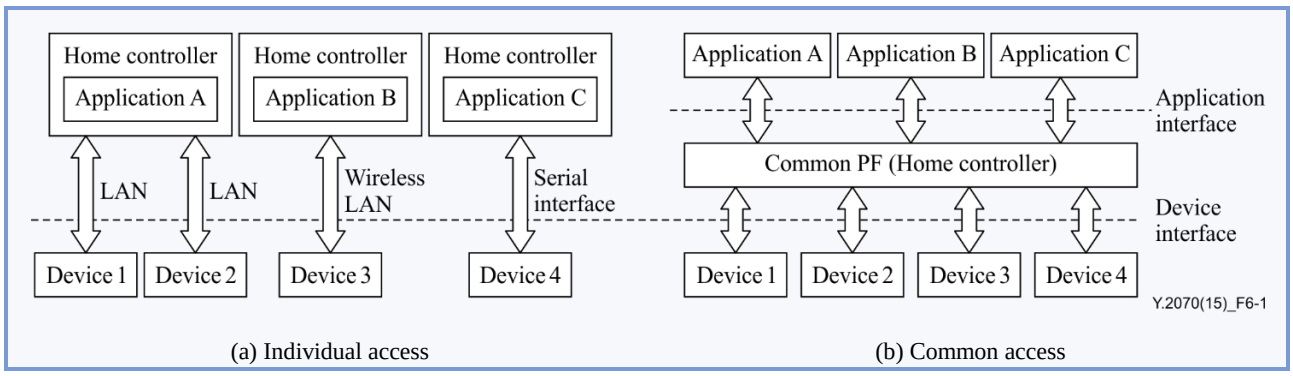


Figure 6-1 – Two access types for HN services

Figure 6-2 below shows an HN service architecture that is composed of two types of architecture.

Figure 6-2(a) shows an architecture in which all of the devices and the home controller are placed inside the home, and the applications and the common PF work on the home controller. This is referred to as the aggregate type architecture in this Recommendation. Figure 6-2(b) shows an architecture in which the devices are located inside the home, but the applications can be placed on the Internet. The functions of the common PF are separately distributed to the HGW inside the home and the management PF on the Internet, instead of on the home controller. This architecture enables the applications to access the devices from the Internet. This architecture is referred to as the distribute type architecture in this Recommendation.

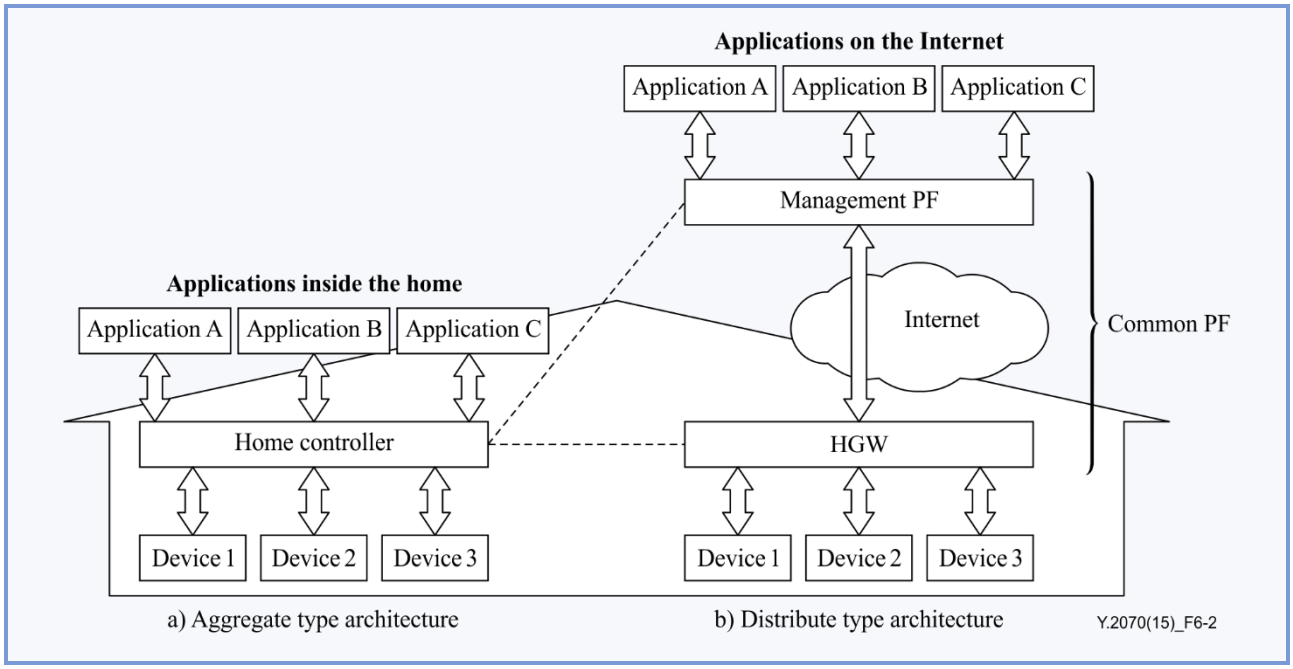


Figure 6-2 – The HN service architecture

Both types of the architecture are within the scope of this Recommendation since both architectures have the same functions of the common PF. In the following clauses, however, the distribute type architecture is mainly described.

6.2 HEMS based on HN service architecture

The HEMS, based on the HN service architecture, is described in this clause, clarifying the features of the architecture.

6.2.1 HEMS and HN service architecture

The HEMS is generally considered to provide the following services:

- Visualization of the energy consumption by the entire house, or by selected devices such as home appliances, storage batteries with power sensors and the smart meter.
- Realization of energy-efficiency and/or cutting energy usage during peak demand by monitoring and controlling the devices.

The HEMS application is one of the HN applications; therefore, the architecture for the HEMS can be the same as that for other HN services which are within the HN service architecture.

The HN service architecture applied for the HEMS is shown in Figure 6-3. This is the distribute type architecture shown in Figure 6-2(b), where by the HEMS application replaces the HN applications.

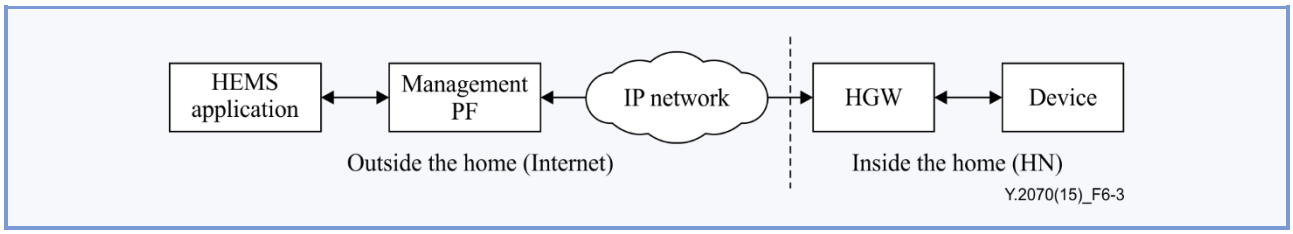


Figure 6-3 – HEMS based on HN service architecture

In Figure 6-3, everything to the left of the HGW (on the left side of the dotted line) is considered the Internet (outside the home) and everything to right hand side is considered the HN (inside the home).

Devices such as home appliances, storage batteries and power sensors connected to the HN are monitored and controlled from the HEMS application on the Internet in this architecture, and in order to do so, the HGW bridges the Internet and the HN. The HGW converts the various types of communications protocols used for communication with the devices to the protocol, which is used on the Internet for communication with the management PF. The management PF is placed on the Internet and provides a web-based application interface. The HEMS application runs through this interface.

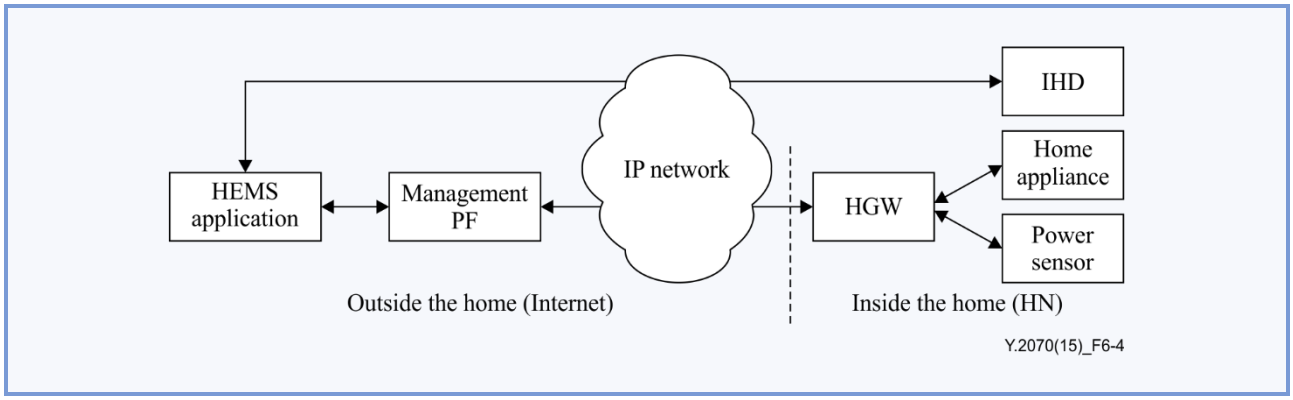
With the HGW and the management PF, it is possible for the HEMS application to discover and identify devices connected to the HN and to access them using their identifiers. In this way, the HEMS application monitors and controls the individual devices and enables the HEMS.

By making use of a standardized communication protocol between the HGW and the management PF, the HEMS application does not need to take into consideration the interfaces of the devices or the communications protocols used between the HGW and the devices. The devices are represented as web resources by the management PF. Therefore this architecture supports application developers by allowing them to develop applications without deep knowledge about multiple device interfaces and communications protocols.

6.2.2 HEMS examples

This clause describes two HEMS examples based on the HN service architecture; these show the features of the architecture.

In Figure 6-4, devices such as a home appliance, e.g., an air conditioner, and power sensors are connected to an HGW in the home, using various networks and protocols such as standardized communications protocols and dedicated communications protocols depending on the interface of each device. A HEMS application runs on the Internet and connects to the devices through the HGW and the management PF. In this architecture, the HEMS application collects the electronic power consumption data of the home appliance from the power sensors through the HGW and the management PF, and sends the data to visualize the energy consumption in the web text-based format to the web browser on the in-home display (IHD). It may be also possible to make the system send the data to web browser on a smart phone so that the end users can refer to the energy consumption from outside the home.



F Figure 6-4 – Visualization of energy consumption with IHD

This service may provide the end user with a visualization of the energy consumption in a graphical representation, e.g., for the past week. For this service, the management PF stores the data received from the power sensors and provides them to the HEMS application when required.

In Figure 6-5, a HEMS application makes use of a utility company's service, such as the demand response (DR) service. This architecture enables new services by combining Internet-based services. This is the main feature of this architecture in which the application is placed on the Internet.

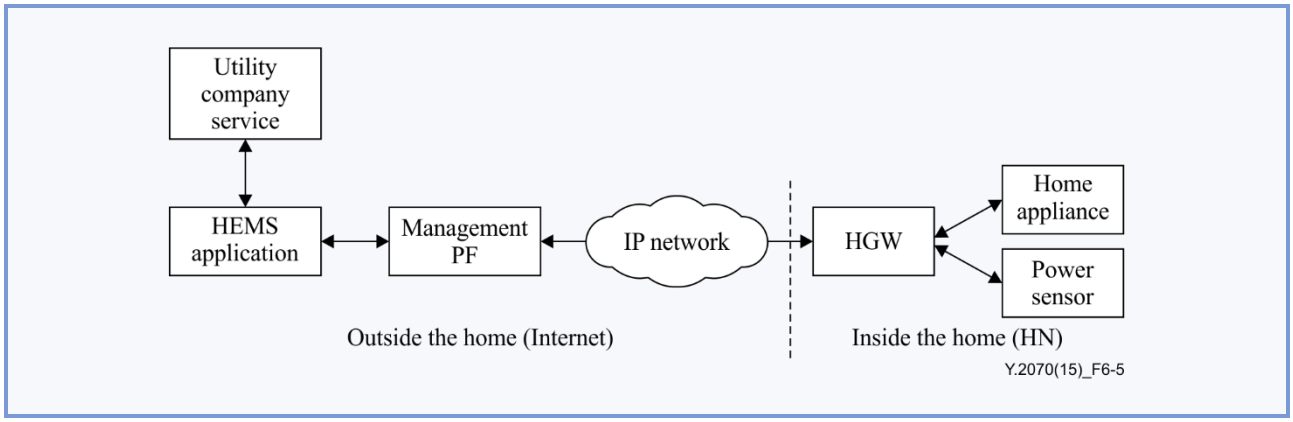


Figure 6-5 – Energy consumption control with DR

6.3 Merits of HN service architecture

The previous clauses showed the HN service architecture and how to apply the architecture to the HEMS. Although some of the merits of the architecture are described in the previous clauses, they are also summarized below.

As described in clause 6.1, the HN service architecture is composed of the aggregate type architecture (Figure 6-2(a)) and the distribute type architecture (Figure 6-2(b)), and both architectures have the same functions of the common PF. The merits of the HN service architecture (both the aggregate type architecture and the distribute type architecture) are described below in items 1) through 3).

1) It enables application developers to develop applications with the application interface on the common PF (i.e., the home controller in the aggregate type architecture and the management PF in the distribute type architecture) to provide various services.

2) It enables easier and lower-cost introduction of services for end users who can install by themselves devices that support plug-and-play functionality. It is not necessary for the end users to be concerned about the connections to the devices.

3) It enables the common PF to maintain the entire system and the HN resources remotely. The common PF provides functions for auto-configuration of devices and for detection of faults occurring on the HN. These functions make the system more stable and provide services at a lower cost.

The distribute type architecture has a function on the management PF to cooperate with devices and applications working on networks that have different policies. Thus, there are some additional merits to the distribute type architecture; these are described below in items 4) through 6).

4) It avoids increasing hardware resources, such as central processing unit (CPU) or memory on the home controller which would lead to an increase in service costs when providing other HN services in addition to the HEMS. It enables modifying/adding applications on the Internet without hardware restrictions.

5) It enables the applications to easily monitor and control the devices on the HN with the management PF, which provides the application programming interface (API) needed to access the devices. This access is enabled even in the case where a firewall and network address translation (NAT) protects the devices from illegal access from the Internet.

6) It enables easy development of security-conscious applications by providing functions for the authentication and the authorization for the devices and the HGW, and the encryption for the HN.

7 Requirements

This clause describes requirements for the device, the HGW and the management PF in the HN service architecture, as well as for the security required for the architecture. Since the functions of the HEMS are specified in clause 6.2, the following requirements are extracted for the HEMS. As the HN service architecture is applied not only for the HEMS but also for other HN services, the following are also requirements for these other HN services as well. The requirements for the applications are out of the scope of this Recommendation.

7.1 Requirements for the device

The following are requirements for the device:

1) requirement for device operation

- device object

  It is required to have a device object which is an abstract data model representing the functions of the device.

NOTE – For a device that does not have a device object, the adapter or the HGW connecting to the device directly is required to have a device object.

2) requirements for management:

- managed agent

  It is required to respond to the resource information collector function of the HGW.

  It is required to check the status of the device itself for fault diagnosis.

  It is required to set the configuration of the device and the network device such as the hub and the access point in the HN.

7.2 Requirements for HGW

The followings are requirements for the HGW:

1) requirement for device operation

- data format and protocol (hypertext transfer protocol (HTTP)/Internet protocol (IP)) conversion

  It is required to convert the format of the device object to that of the virtual device and Internet protocol (IP) to HTTP as the protocol which delivers the format on the WAN with the secure communication to the management PF.

2) requirements for management

- resource information collector

  It is required to discover the devices that are newly connected to the HN, identify each of them and manage their status.

  It is required to collect the internal status of each device and other HN resources, and the traffic status of the HN in order to determine the cause of any fault when the HN service is not working well.

3) requirements for application execution

- application for disconnect

  It can optionally work autonomously, continuing to control devices and store data with a backup purpose application in case of network disconnection from the management PF.

  It can optionally take some tasks with the backup purpose application and work with the management PF.

7.3 Requirements for management PF

The followings are requirements for the management PF:

1) requirements for device operation

- virtual device

  It is required to provide a web-friendly representation corresponding to the device object.

  It is required to monitor and control the virtual device.

  It can optionally be enabled to take the plural functions in one physical device as plural virtual devices and the plural physical devices as one virtual device.

NOTE – The physical device represents the device connected to the HN. The term physical device is used to distinguish it from a virtual device.

2) requirements for management

- resource management

  It is required to discover, activate, monitor and control the devices connected to the HGWs.

It is required to identify the devices globally unique from the applications.

It is required to register the HGW's profile, give the HGW an identifier, register the end user who owns the HGW and identify the HGW with the authenticated user information.

3) requirements for application execution

- application management

  It is required to have the capability of authenticating and authorizing applications with acceptance by the end user to connect to the devices.

  It can optionally store the data received from the devices through the HGW.

- application interface

  It is required to have a web-based application interface for the HEMS and other HN services.

7.4 Requirements for security

The following are requirements for security:

- secure communication

  It is required to securely communicate between the devices and the application through the WAN.

NOTE – HEMS security requirements to support secure communication are listed in Table III.2.

- device authentication

  It is required for the management PF to have the capability to authenticate the devices.

8 Reference architecture

This clause describes the reference architecture for the HEMS. This architecture can also be applied to other HN services.

Figure 8-1 shows the distribute type reference architecture and the reference points of the architecture.

In Figure 8-1, the left side of the HGW is the WAN (i.e., outside the home) and right side is the HN. The HGW is IP based and it bridges the WAN and the HN.

This reference architecture is shown with the categorized devices; basic device and non-basic device. The basic device has a device object which is an abstract data model representing functions of the device. The interface of the basic device is provided to the HGW with the abstract data model and is represented in the management PF as virtual devices, which the applications monitor and control.

There are some standardized specifications for the interface of the basic device. Some of them support IP based communications protocols and the others support non-IP based protocols. Therefore, the two types of basic devices are those having the interface for the IP based protocol and those having the interface for the non-IP based protocols.

The non-basic device does not have the device object. Since it supports its dedicated interface, an adapter is required to connect a non-basic device to the HGW. If the HGW provides the function of the adapter, the non-basic device can be connected directly to the HGW.

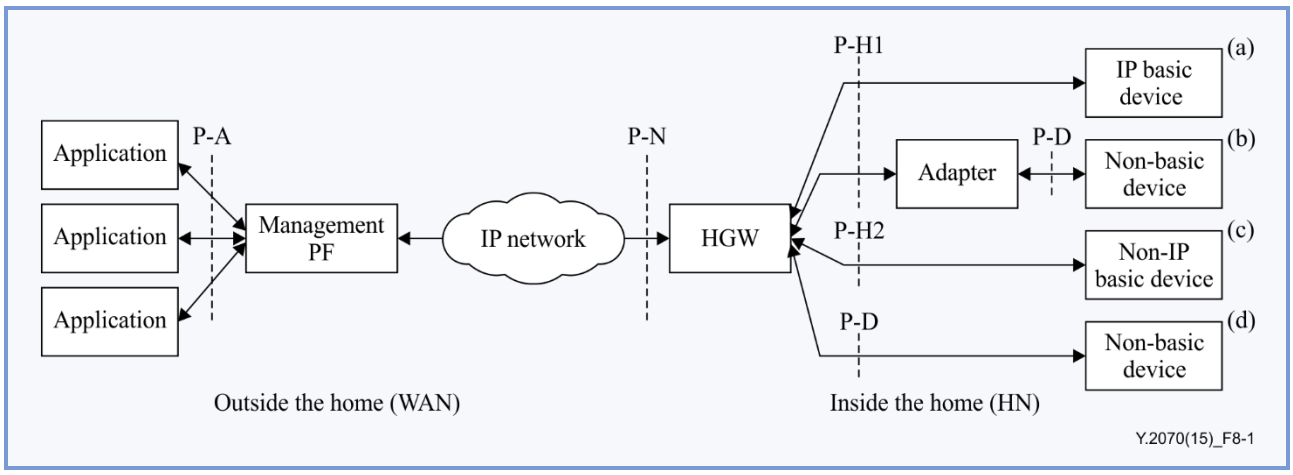


Figure 8-1 – Reference architecture and reference points

In Figure 8-1, the following reference points are defined. The application interface in Figure 6-1 corresponds to the P-A reference point in Figure 8-1. The device interface shown in Figure 6-1 corresponds to the P-H1, the P-H2 and the P-D reference points in Figure 8-1.

1) P-A reference point

The P-A reference point allows the applications to access the management PF through web-based application interfaces and to monitor and control the physical devices connected to the HN as logical devices of web resources. This reference point enables the applications to create and delete the logical devices, and to read and update the properties for them.

2) P-N reference point

The P-N reference point allows the management PF to access the HGW, placed in the home, through the WAN. This reference point enables the management PF to activate the devices, to get their status, and control them by specifying the property's value as a function of the resource management.

3) P-H1 reference point

The P-H1 reference point allows the HGW to access the basic device with the IP based communications protocol (IP based basic device), and the adapter which converts the non-basic device connecting it to the basic device. This reference point enables the HGW to activate the devices, get the status of, and control them by specifying the property's value.

4) P-H2 reference point

The P-H2 reference point allows the HGW to access the basic device with the non-IP based communications protocol (non-IP based basic device). This reference point enables the HGW to activate the devices, get the status of, and control them by specifying the property's value.

5) P-D reference point

The P-D reference point allows the adapter and the function of the adapter equipped in the HGW to access the non-basic device with the dedicated communications protocol, which connects to the device interface. This reference point enables the HGW to activate the devices, get the status of, and control them by specifying the property's value.

The devices in the home are connected to the HGW through the HN in one of four ways. The following is a description of the devices from (a) to (d) shown in Figure 8-1.

Device (a) is an IP based basic device that connects directly to the HGW at the P-H1 reference point. It makes use of the IP based communications protocol between the device and the HGW as the device has an interface to connect to the protocol. [b-ECHONET Lite] is one of such communications protocols.

Device (b) is a non-basic device and supports its dedicated interface. To connect to the HGW, this device requires an adapter which converts the dedicated communications protocol implemented by the interface of the device to the IP based protocol, and which converts the dedicated data model to the abstract data model. Therefore, device (b) will be recognized as a basic device by the adapter. A battery charger for electric vehicles (EVs) connecting to the HN with a serial interface is an example of this type of device.

Device (c) is a basic device, but supports the non-IP based communications protocol only because the non-IP based communications protocol is used for communication directly with the HGW.

Device (d) is a non-basic device and it is the same as device (b). In Figure 8-1, the device connects to the HGW directly since the HGW has functions of the adapter for device (d).

The HGW has a function to discover the newly-installed devices automatically. The HGW receives notifications of the installation of the devices and alarms when malfunctions occur. This raises the system's reliability. It also allows the end users the ability to easily install devices and to get the HEMS and other HN services started. The HGW converts the various types of the communications protocols, used for communicating with the devices, to the protocols which are used on the WAN for communication with the management PF at the P-N reference point. For example, the communications protocol used for communication with the devices will be converted into HTTP.

The communication between the HGW and the device (through the adapter) may not be encrypted regardless of whether the IP or non-IP based communications protocol is used. However, the communication between the HGW and the management PF through the WAN is encrypted or is on a secured communications protocol such as HTTP to ensure secure communication. HTTP can be utilized on the standardized device management protocol such as [b-BBF TR-069].

The management PF is a server that provides the web-based application interface on the WAN. The applications run through the interface at the P-A reference point. By making use of the standardized communications protocol between the HGW and the management PF, the applications do not need to take into consideration the interface of the devices or the communications protocols used on the HN. The devices are represented as web resources by the management PF. In this way the application developers can develop applications without deep knowledge about the devices.

NOTE – A deployment model with the web of things (WoT) specified in [b-ITU-T Y.2063] is shown in Appendix I.

9 Functional architecture

This clause describes the functional architecture for the HEMS. This architecture can also be applied to other HN services.

Figure 9-1 shows the distribute type functional architecture for the IP based basic device. The functions in this architecture are composed of three categories: device operation, application execution and management. The details for each category are described in clause 10. Although the device is shown as an IP based basic device in Figure 9-1, this architecture can also be applied to other types of devices with the appropriate deployment of the functions between the HGW and the device. The architectures for each type of device are shown in clause 10.1.1 to 10.1.4. The architecture between the management PF and the application is the same for all types of devices as shown in Figure 9-1.

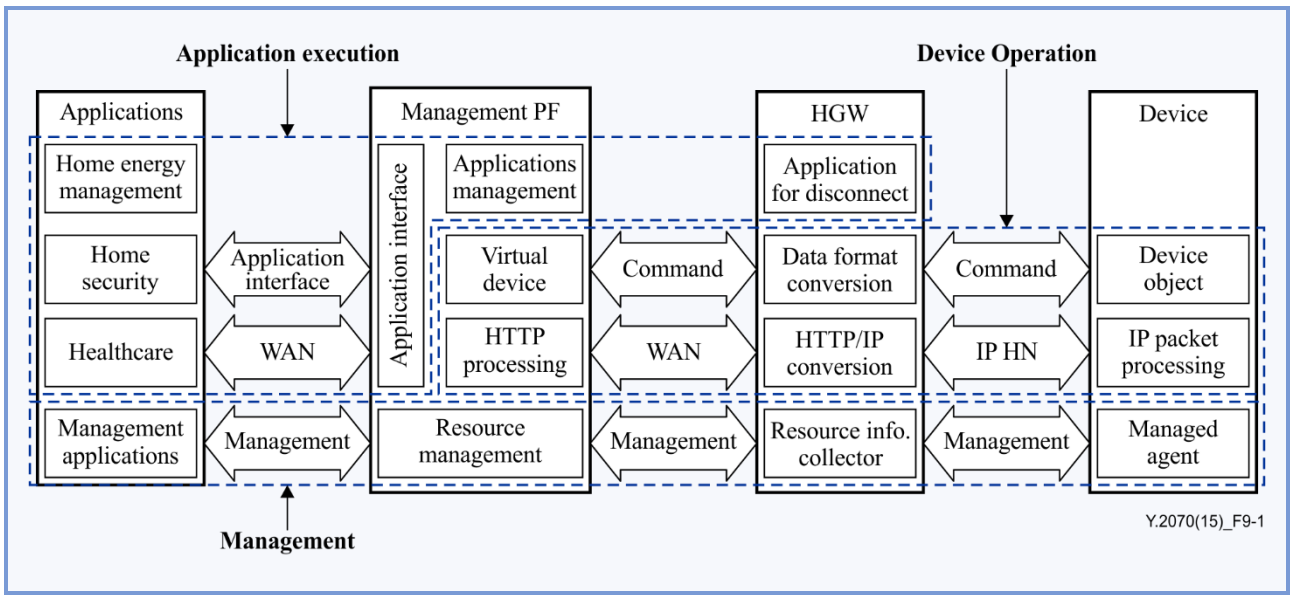


Figure 9-1 – Functional architecture for IP based basic device

The devices have their own proprietary functions. The functions are defined as profiles and are transmitted with communications protocols to the application via the HGW and the management PF. This architecture enables the application to monitor and control the devices.

Although Figure 9-1 shows the IP based communications protocol for the IP based basic device, the IP based, the non-IP based and the dedicated communications protocols can be used for the communication between the HGW and the devices as described in clause 8. The commands transmitted between the HGW and the devices provide control methods for the HGW to the devices such as "GET" to get their status, "SET" to specify their properties and/or set the value and "INFORM" to request notification about their status and the events that have occurred in them. The HGW converts the communications protocol into HTTP and communicates to the management PF through the WAN.

The communications protocol for device management such as [b-BBF TR-069] can be used between the HGW and the management PF. [b-BBF TR-069] refers to the generalized device data model with extensible markup language (XML) in [b-BBF TR-181] and this data model is communicated between them with commands such as "GET", "SET" and "INFORM". The XML format is specified for each communications protocol used between the HGW and the devices.

The management PF stores device status and configuration data transmitted via the HGW. The management PF manages virtual devices and provides them to the application through a web-based application interface so that the application developers can develop applications to control the physical devices as web resources.

The following clauses describe each entity. The functions of the device and the HGW are described for the IP based basic device. The functions for other types of devices are shown in clause 10.1.2 to 10.1.4. The functions for the management PF and the applications are common to all types of devices.

NOTE – Examples of HN applications are described in Appendix II.

9.1 Device

The IP based basic device provides the following functions:

1) functions for device operation
   - device object;
   - IP packets processing.
2) function for management
   - managed agent.

Device object and managed agent are described in the following clauses. IP packet processing is not described, as it does not require specific operations.

9.1.1 Device object

The basic device has the device object. It is composed of properties that specify the device functions which are independent of the implementation of the manufacturers. The properties are logical internal items to get the device status and to control the device functions, which can be remotely accessed and controlled from the application. The data form for the remote control is specified as the tuple of <property, value>. Since the device object is specified for each type of device (e.g., home appliance, storage battery), existing home appliances made by different manufacturers would be remotely controlled in exactly the same way.

For example, air conditioners have properties of operating status, temperature setting and operation mode, which are defined as the property configurations in [b-ECHONET Lite]. [b-SEP 2.0] and [b-ISO/IEC 14543-3-x] also define similar property configurations. The HGW specifies the property to get the data (value) from them. To configure or control them, it specifies the property and sets the appropriate value. For example, to set the targeted temperature of an air conditioner, it specifies the property, which is appointed for the targeted temperature and sets the appropriate value (e.g., 25 (degrees)).

9.1.2 Managed agent

The managed agent is a function for the management to keep the HN stable. It is important to get the information about all of the HN resources because lack of this information could cause failure of fault detection thus, making determination of the causes of the fault difficult. The managed agent holds the internal status of the device and transfers it on demand to the resource management function of the management PF via the resource information collector of the HGW. This detail is described in clause 10.3.

9.2 HGW

The HGW bridges the WAN and the HN. It provides the following functions when it connects to the IP based basic device:

1) function for device operation
   - data format and protocol (HTTP/IP) conversion.
2) function for management
   - resource information collector.
3) function for application execution
   - application for disconnect.

9.2.1 Data format and protocol (HTTP/IP) conversion

This is a function of the device operation to convert the communications protocol. On the WAN, HTTP is usually utilized as the communications protocol. Thus, the HGW converts the communications protocol used on the HN to HTTP.

The tuple of <property, value> of the physical devices is communicated to the HGW through the HN and put into HTTP at the HGW for further communication with the management PF. It is widely known that [b-BBF TR-069] specifies the simple object access protocol (SOAP) based communications protocol; thus it is one of the candidate communications protocols for the device management between the HGW and the management PF. The extensible messaging and presence communications protocol (XMPP) is also a candidate.

9.2.2 Resource information collector

This is a management function used to collect information about the HN resources, for each of the HGW devices, and to deliver it to the resource management function on the management PF. This function also discovers the devices newly connected to the HGW, and sets the configuration for these devices. The HGW gives a unique identifier to each of the devices for management.

9.2.3 Application for disconnect

The application for disconnect function provides a backup purpose application to keep the devices connected to the HN working when the WAN is disconnected for any reason. If the WAN disconnects, this backup purpose application sets the proper configuration for the situation instead of the application running on the WAN. The application interface for this backup application is based on HTTP and provides the API, which manages the device object using the converted data format.

9.3 Management PF

The management PF manages the physical devices as virtual devices and provides them as web resources to the application through the web-based application interface. The management PF provides the following functions:

1) functions for device operation
   - virtual device;
   - HTTP processing.
2) function for management
   - resource management.
3) functions for application execution
   - applications management;
   - application interface.

These functions are described in the following clauses, except for HTTP processing: it is not described because it does not require specific operations.

9.3.1 Virtual device

The virtual device is the device representation corresponding to the device object of the basic device connected to the HN. The properties of the device are represented in XML format in order to be easily handled by the web applications. The function is described in clause 10.1.

The virtual device provides two functions for the device abstraction. The first function provides abstraction of the devices' properties and the communication protocols. For example, if vendor A and vendor B give different properties to similar functions of their air conditioners, the management PF creates a virtual device by converting the property of the devices (e.g., air conditioners) to the same properties.

The second function provides virtual separation to the plural virtual devices which are made from the plural functions in one physical device. For example, an air conditioner controlling its power based on its motion sensor has the property based on the data detected by the motion sensor. Thus, the air conditioner has two functions: air conditioning and motion sensing. Two virtual devices (i.e., air conditioning device and a motion sensing device) will be created from one physical device (i.e., air conditioner).

9.3.2 Resource management

The resource management on the management PF provides the function to gather information of the HN resources which the managed agent collects. It also manages the internal status of the device, the network device and the network capacity for each HGW to detect fault and to provide the fault processing. The details are described in clause 10.3.

9.3.3 Applications management

The applications management registers the information of the application and holds the relationships between the applications and the devices. This function delivers data from the devices to the appropriate applications. In addition, it has the historical data management function which stores the data received from the devices instead of the applications. The function provides the data of the devices, for example for the past 24 hours, in responding to the requirement of the application.

9.3.4 Application interface

The application interface is a web-based interface used to monitor and control the devices through the HGW from the application by accessing the virtual devices on the management PF as web resources. This makes it possible, therefore, for application developers to develop applications with this interface.

This interface does not support the management as shown in Figure 9-1.

9.4 Application

Although the architecture in this Recommendation does not require common functions for the applications, three functions in the applications which provide the following three operations that make use of the functions of the management PF are described in this clause. These three functions are: the device management function, the device operation function and the fault diagnosis function as shown in Figure 9-2. Every application in the application entity shown in Figure 9-1 can have these functions.

The three functions are described below with their corresponding operation.

1) function: device management, operation: registration

The applications that access the devices are registered to the applications management function in the management PF. This operation is performed by the device management function in the application and enables exclusive control to the devices and specifies the application to which the data of the devices is delivered automatically.

2) function: device operation, operation: monitor & control

The device operation function in the application monitors and controls the devices by getting and setting the configurations. ON / OFF control to the devices is one example. This operation is performed by monitoring and controlling the virtual devices in the management PF which is described in clause 10.2. The operation from the virtual devices to the physical devices on the HN is described in clause 10.1.

3) function: fault diagnosis, operation: fault detection

The fault diagnosis function in the application gets the status and configuration data of the HN resources. This function connects to the resource management in the management PF and gets the HN related information detected by the resource management and provide it to those applications which need the information.

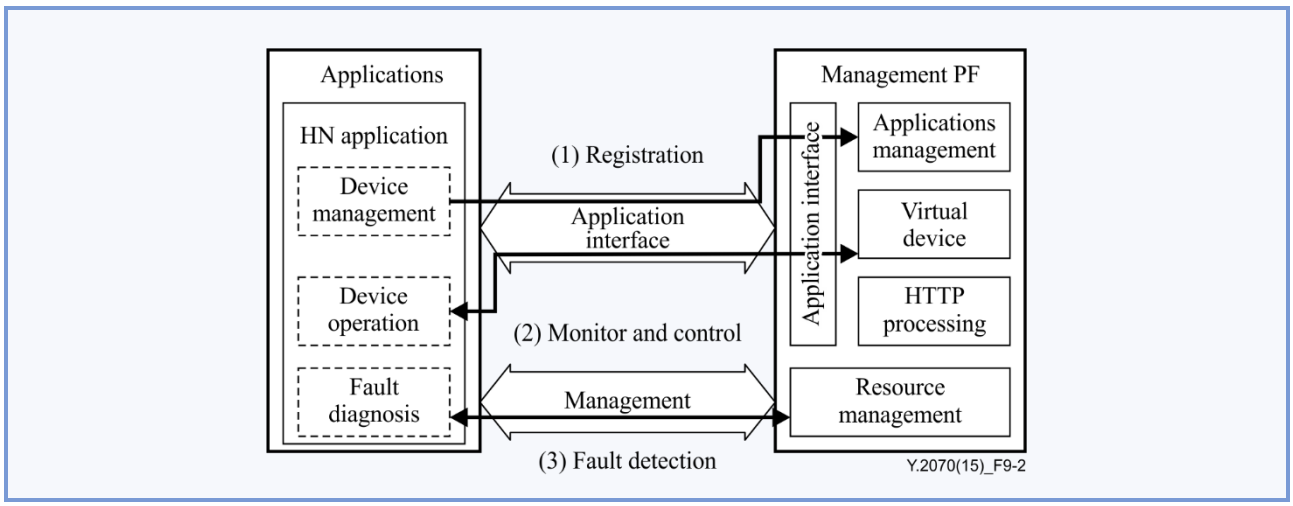


Figure 9-2 – Three operations for the HN applications

10 Functional relationship

In this clause, details of the three functional categories: device operation, application execution and management shown in Figure 9-1 are described to clarify the relationship between the entities.

10.1 Device operation

The device operation provides a function to monitor and control the devices from the management PF. Since there are four ways to connect to the devices from the HGW according to the device types shown in Figure 8-1, four operations for each device type are described in the following clauses.

10.1.1 Operation for IP based basic device

Figure 10-1 shows the functional architecture for the IP based basic device operation (i.e., device (a) in Figure 8-1). The IP based basic device has two functions; device object and IP packets processing. The virtual device on the management PF is the device representation corresponding to the device object of the basic device. The applications remotely monitor and control the devices by specifying the properties of the virtual devices through the application interface.

The two-tuple of <property, value> is the data form used to control the device. The device command is transferred to the HGW on the IP based communications protocol through the HN and to the management PF on the HTTP based protocol through the WAN. The HGW converts the device command between the HN and the WAN since the form of the tuples on the HN is different from that on the WAN. Therefore, the HGW has two functions: data format conversion that converts the form of the tuple, and protocol (HTTP/IP) conversion that converts the communications protocol on the HN to HTTP.

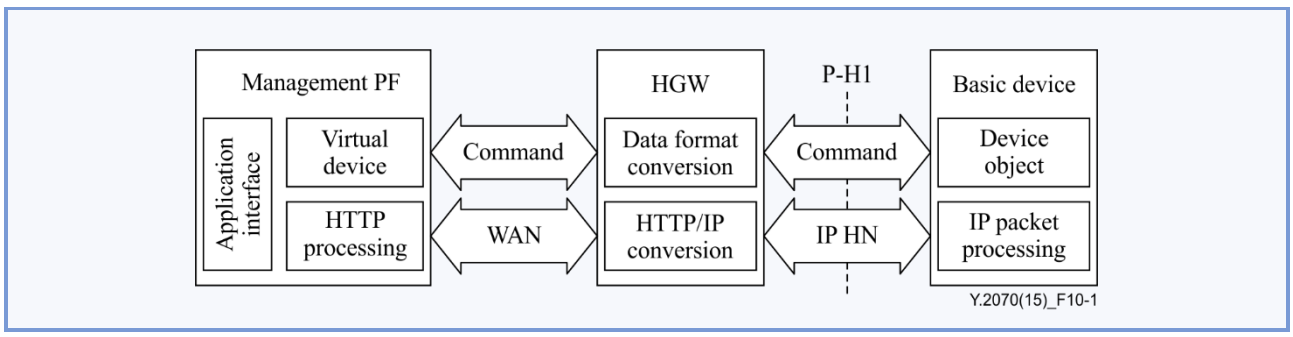


Figure 10-1 – Functional architecture for IP based basic device operation

10.1.2 Operation for non-IP based basic device

Figure 10-2 shows the functional architecture for the basic device used to connect to the HGW with the non-IP based communications protocol (i.e., device (c) in Figure 8-1). In this case, the device command is transferred at the P-H2 reference point. The layer 2 (L2) frame processing function can be used to convert the non-IP based communications protocol, which the device supports, to the IP based protocol in the HGW. The communications protocol between the HGW and the management PF can be the same as the IP based basic device shown in Figure 10-1.

The device interface of [b-SEP 2.0] supports the P-H2 reference point.

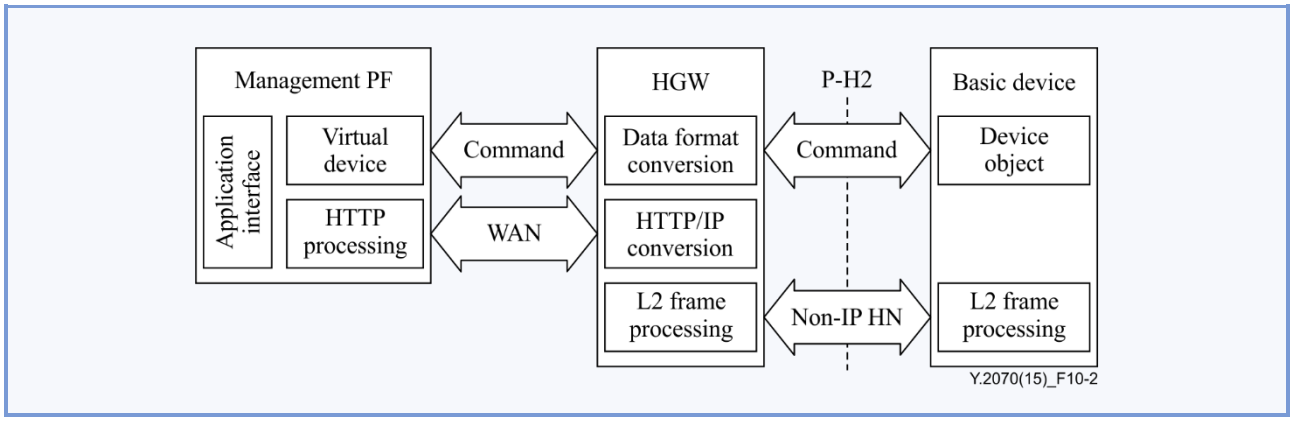


Figure 10-2 – Functional architecture for non-IP based basic device operation

10.1.3 Operation for non-basic device with adapter

Figure 10-3 shows the functional architecture for the non-basic device operation with the adapter (i.e., device (b) in Figure 8-1). This Recommendation defines the non-basic device for the existing devices that do not have the device object. The non-basic device does not have the device object, but has the dedicated interface, for example the serial interface, at the reference point P-D. The adapter has the device object and provides the same interface as the basic device. The adapter, placed between the device and the HGW, works to convert the non-basic device to be recognized as a basic device at the reference point P-H1. In this way, the adapter converts the dedicated protocol to the IP based protocol such as [b-SEP 2.0], [b-ECHONET Lite], [b-ISO/IEC 14543-3-x] and [b-BACnet]. The device object is provided in the device abstraction function. The device abstraction function also provides the device interface conversion function that converts the property of the device to the operational procedure to the device at the reference point P-D.

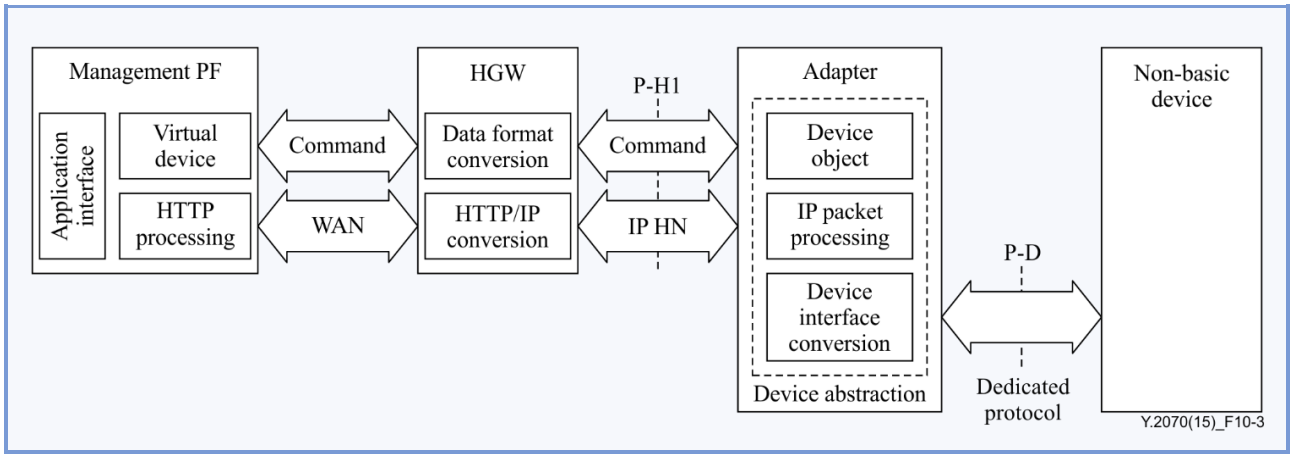


Figure 10-3 – Functional architecture for non-basic device operation with adapter

10.1.4 Operation for non-basic device with adapter function in HGW

Figure 10-4 shows the functional architecture for the non-basic device operation connecting directly to the HGW (i.e., device (d) in Figure 8-1). For the non-basic device, the HGW equips the function of the adapter (the device abstraction function) instead of placing the adapter to connect to the device directly. The device abstraction function inside the HGW provides the same interface as the basic device. The device abstraction function also supports the operational procedure at the reference point P-D and the non-basic device connects to the HGW directly as shown in Figure 10-4.

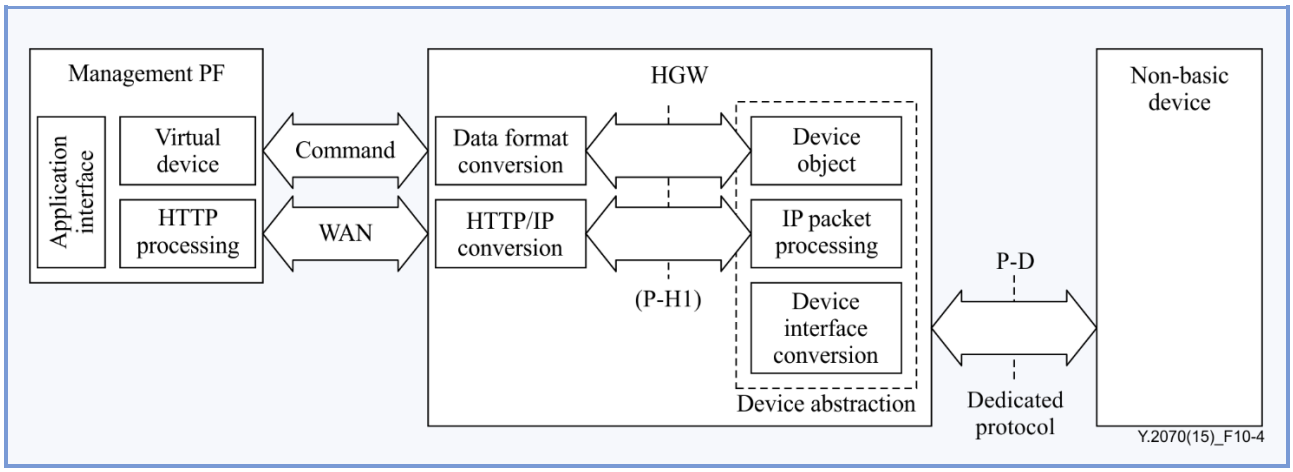


Figure 10-4 – Functional architecture for non-basic device operation with adapter function in HGW

10.2 Application execution

Setting and getting the value of the property of the virtual device on the management PF by the application results in monitoring and controlling the physical devices connected to the HN. As shown in Figure 10-5, the application interface on the management PF converts the virtual device to the web resource, enabling the application to monitor and control the physical devices with the HTTP protocol.

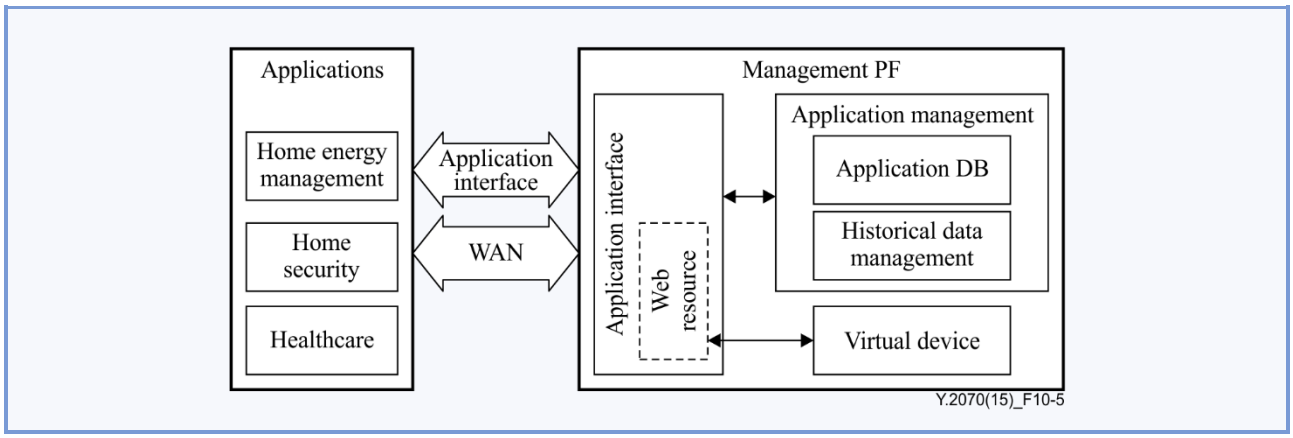


Figure 10-5 – Functional architecture for application execution

The applications management is composed of two functions: application database (DB) and historical data management.

The application DB maintains a list of the applications connected to the management PF through the WAN. The application DB is used to deliver data to the targeted application from the HGWs and the other applications.

The historical data management function is described in clause 9.3.3.

10.3 Management

The HN is sometimes very complex; many different types of technologies may co-exist in the HN. The devices connected to the HN are used in a variety of fields. The HN could have a complicated topology composed of various HN resources (e.g., devices and/or access points). It could be difficult to manage and maintain the HN for end users, since there is no administrator or technician in the home. Therefore, resource management is provided to support a variety of fault determination processes and fault recovery processes, including easy configuration with no administrators and with remote administrators.

Figure 10-6 shows the functional architecture of the management for the HN with the basic devices. For the non-basic devices, which do not have the managed agent, the adapter or the HGW provides the managed agent.

There is the resource management function in the management PF, which holds the information and configuration data of the HN resources. The HGW has the resource information collector function. It gets status, performance and configuration data of the HN resources, detects faults and provides fault processing. It also provides easy configuration procedures of the HN resources for end users. It configures the HN resources with minimum settings required for their operation. The managed agent on the device executes configuring and gathering the home environment information by the instruction from the resource information collector function on the HGW. The management application is the application for use by remote administrators such as call centers and customer support centers. It provides a function to display the entire resource information for faults diagnosis and to set the specified properties for recovery operation from such faults.

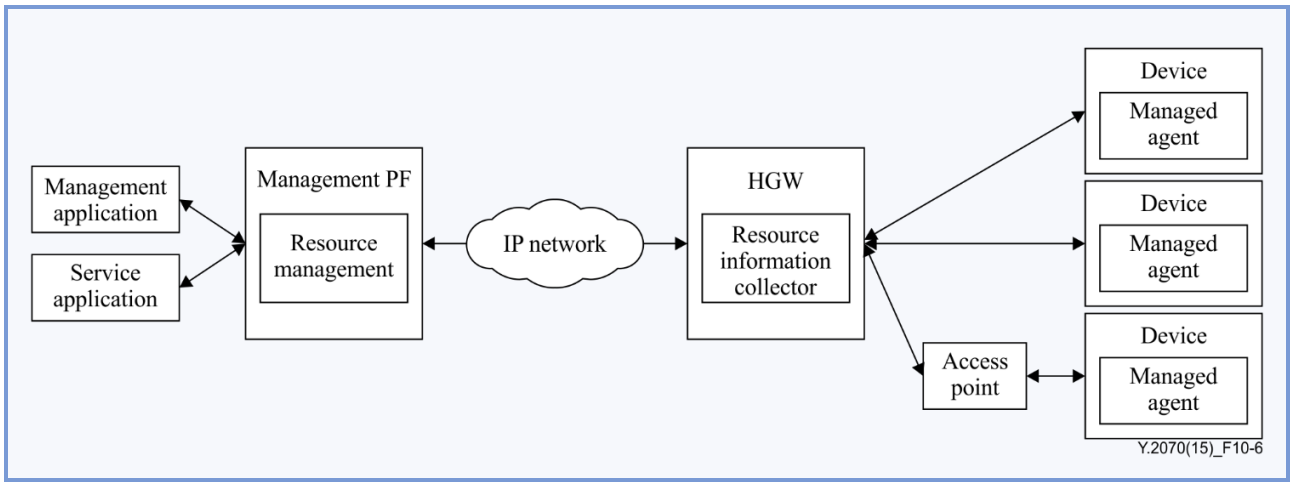


Figure 10-6 – Functional architecture for management for basic device

11 Security support

This clause describes the security model and functions for the HN services, especially the HEMS. The general security requirements and technologies for the HN are described in [ITU-T X.1111]. This Recommendation applies the technologies in [ITU-T X.1111] to the HN service architecture to establish secure communications between the device and the application through the WAN. A HEMS model for security is shown in clause 11.1. As the result of the security considerations on the HEMS model in Appendix III, the security functional architecture is shown in clause 11.2.

11.1 HEMS model for security

This clause describes a HEMS model for security shown in Figure 11-1 which is based on the distribute type architecture shown in Figure 6-4. Since the HEMS is just one of the HN services, this model can also be applied to the general HN model.

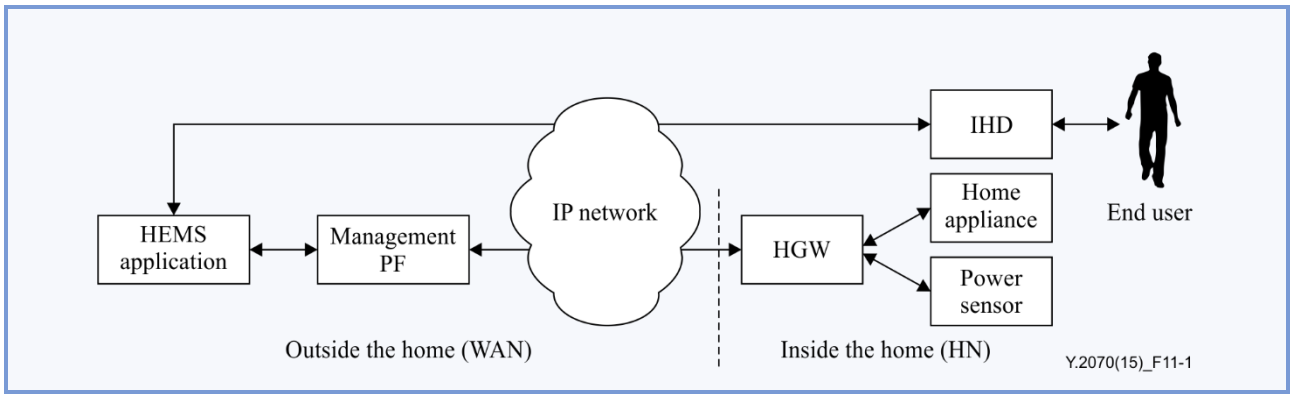


Figure 11-1 – HEMS model for security

In Figure 11-1 the end user uses the web browser on an IHD to connect to the HEMS application with a secure connection. The web browser on the IHD connects to the HEMS application through a broadband router in the home. The HEMS application collects data of the devices (e.g., home appliance properties and power sensor properties) and controls them in response to instructions from the end user.

The HN security is specified in [ITU-T X.1111] and this Recommendation refers to [ITU-T X.1111] as a framework for security technologies in the HN.

The entities and the relationships in this model are required to be specified since they are different from those in [ITU-T X.1111]. There are six entities in this model: end user, IHD, HEMS application, management PF, HGW and device such as home appliance or power sensor. There are five relationships in this model: between end user and IHD, IHD and HEMS application, HEMS application and management PF, management PF and HGW, and HGW and device.

In Appendix III, the security considerations of this model are described based on the process in [ITU-T X.1111] and the relationship between the security functions and this model is shown in Table III.3.

11.2 Security functions

The devices are expected to have some of the security functions specified in [ITU-T X.1111]. Figure 11-2 shows the security functional architecture derived from the result of the considerations in Appendix III based on the technologies in [ITU-T X.1111]. In Figure 11-2, the security functions shown with a solid line or dotted line represent required functions or optional functions respectively. Although nine security functions are specified in Table III.3 of Appendix III, the security functions in Figure 11-2 are simplified into three functions (i.e., anti-availability, message authentication and entity authentication) taking [b-ISO/IEC 27000] into account. The anti-availability function provides protection from attacks by an unauthorized entity. The message authentication function protects information being transmitted from modification, to maintain accuracy and completeness. The entity authentication function identifies the device to protect it from making information of the entity available to unauthorized entities.

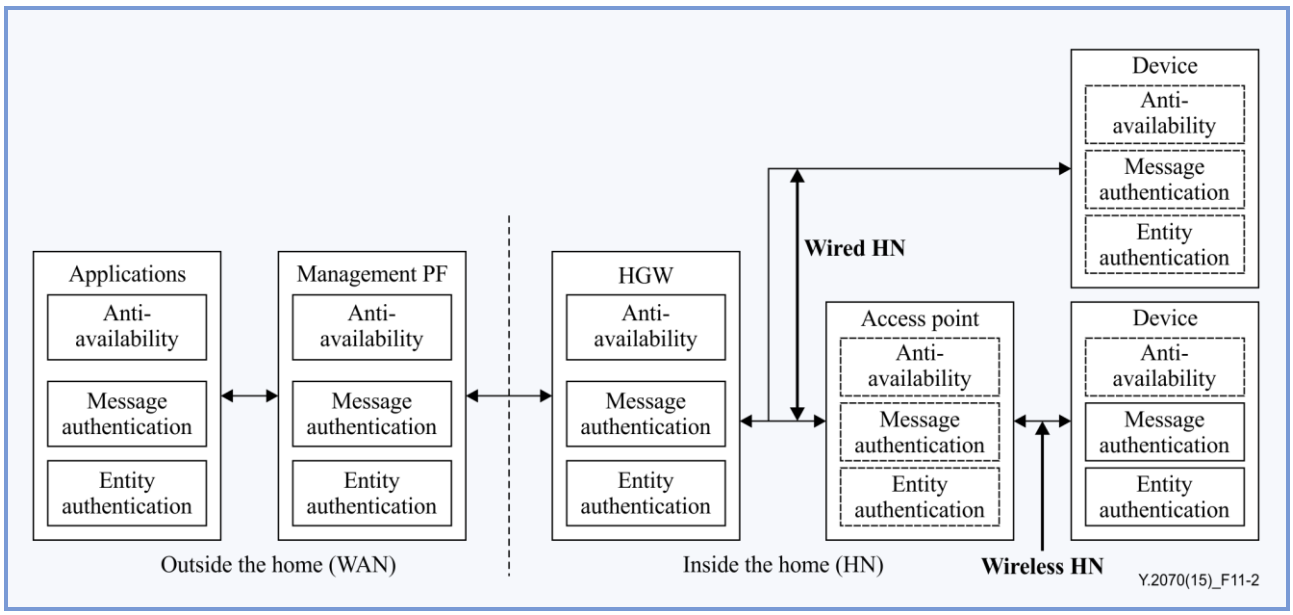


Figure 11-2 – Security functional architecture

Since actual devices such as home appliances and sensors do not have capabilities of the full security functions, in many cases because of its low performance, two solutions to compensate for the limitation of the device's security functions are provided as described below.

First, the management PF and the HGW support security functions as shown in Figure 11-2. In addition, the broadband router (not shown in Figure 11-2), which is usually placed between the WAN and the HN, generally has firewall functions. In this way, the devices connecting to the HGW with the wired HN are usually allowed to have no security functions and the devices connecting with the wireless HN are allowed to have only the least security functions required for the wireless connection. Therefore, the message authentication function and the entity authentication function between the device and the access point are required for the secure wireless connection as shown in Figure 11-2.

NOTE – The security functions on the access point shown in Figure 11-2 are those which are required from the HGW through the wired HN and are optional functions.

Another solution is to provide the following entity authentication functions for the HGW and the device by the resource management in the management PF described in clause 10.3. When the HN is maintained to be secure, this is a simple and effective method to provide the entity authentication.

1) HGW authentication

The authentication information of the HGW is pre-registered in the management PF until the HGW connects to the management PF for the first time. When the first connection to the HGW is established, the resource management compares it with the pre-registered information. The information is managed in the resource management function.

2) device authentication

The information identifying the devices is pre-registered in the management PF until they connect to the HGW to accommodate the case where devices do not have the capability of the security functions. The device authentication is provided as a function and is used to determine consistency between their pre-registered information in the management PF and their data when they connect to the HGW.

3) device access control

The entity authentication function to the physical devices is provided as a function of access control to the virtual devices in the management PF.

Appendix I

Deployment model with WoT

(This appendix does not form an integral part of this Recommendation.)

In [b-ITU-T Y.2063] the physical device on the WoT is divided into two categories: constrained device and fully-fledged device.

- constrained device: A constrained device cannot connect to the Internet and has no functionality of the web. The device interacts with an agent of the WoT broker.
- fully-fledged device: A fully-fledged device has the functionalities of the web. The device can interact, not only with the WoT broker, but also with the services on the web.

The constrained device and the fully-fledged device do not have the device object and hence they correspond to the non-basic device described in this Recommendation. The constrained device communicates with the HGW through the adapter then further communicates with the management PF. The fully-fledged device can also communicate with the management PF by way of the HGW.

In this Recommendation, the HGW, which has the function of the resource information collector, manages all of the devices connected to it through the HN including constrained devices and fully-fledged devices.

[b-ITU-T Y.2063] defines the WoT broker and the physical devices can be accessed as web resources through it from the application. Its functional architecture is divided into the service layer and the adaptation layer where the service layer corresponds to the management PF and the adaptation layer corresponds to the HGW in this Recommendation.

The web adaption function of the WoT broker supports only the adaptation of the communications protocol to the web protocol for communication between the physical devices and the WoT services. Therefore the web broker supports the P-D reference point only in this Recommendation. Figure I.1 shows the deployment model in which the HGW is connected to a constrained device and a fully-fledged device.

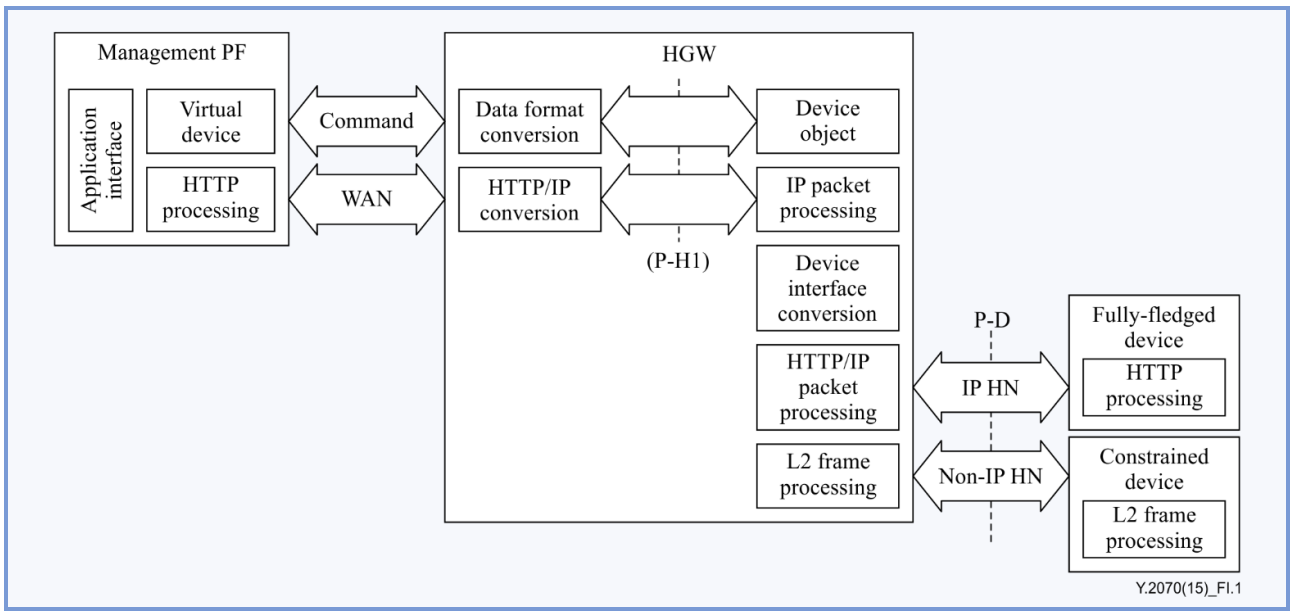


Figure I.1 – Deployment model with WoT

The management PF provides the virtual devices to the application developers to be treated as web resources through the web-based application interface so that they can develop applications for the mash-up service defined in [b-ITU-T Y.2063], which is a combined service integrating WoT services in a WoT broker with web services outside of the WoT broker.

In this way the architecture in this Recommendation provides the HEMS and the other HN services with the WoT.

Appendix II

Examples of HN applications

(This appendix does not form an integral part of this Recommendation.)

The distribute type architecture can be applied to various HN services. The following are examples of HN applications that support these services.

II.1 Home security

A home security application detects threats using sensors installed in the home and alerts the security company to dispatch security guards. The sensors, such as human-aware sensors which detect suspicious persons, or fire sensors which detect a fire, are connected to the home security application through the HGW and the management PF.

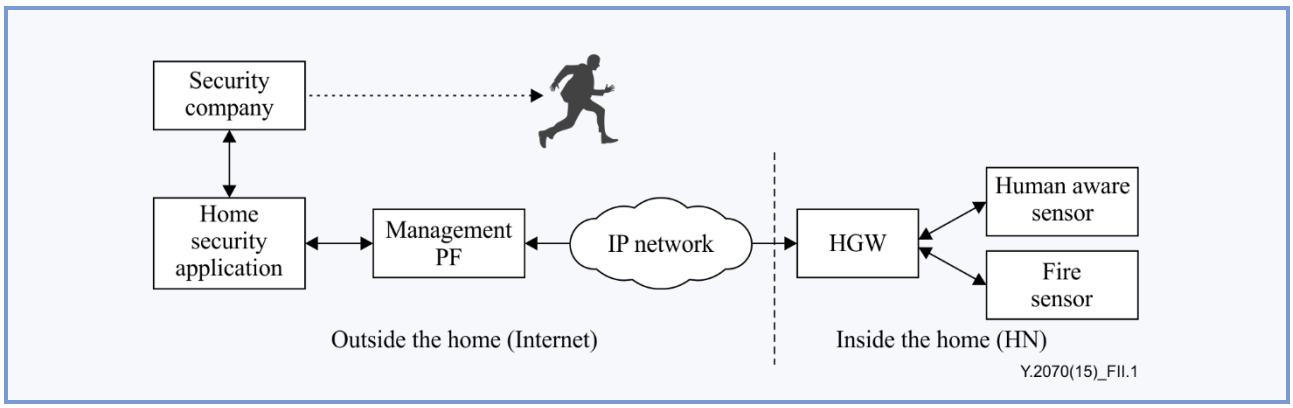


Figure II.1 – Architecture with home security application

II.2 Customer support with controlling access right to device

Customer support is an important service as the HN is getting more complex with the variety of devices being connected. When the service does not work well, it is difficult to determine why and where the fault happens. The example shown in Figure II.2 assumes that two different customer support services are separately provided through the same management PF for each of the supporting devices (appliances) produced by companies A and B respectively.

When the devices produced by more than two different companies are installed on the HN, the challenge for each device company is that each company could get fault information about the devices produced by the other company. Most of device companies prevent other companies from getting this information.

Figure II.2 shows three services running on the management PF. In this case, the HN service provider's application is able to get diagnostic information, such as network disconnections through the management PF. The customer support service gets the detailed diagnostic information about the supporting device (e.g., the customer support service for company A gets the detailed fault information about the supporting device produced by company A). This service is provided because the management PF has a function for distributing information to the proper services.

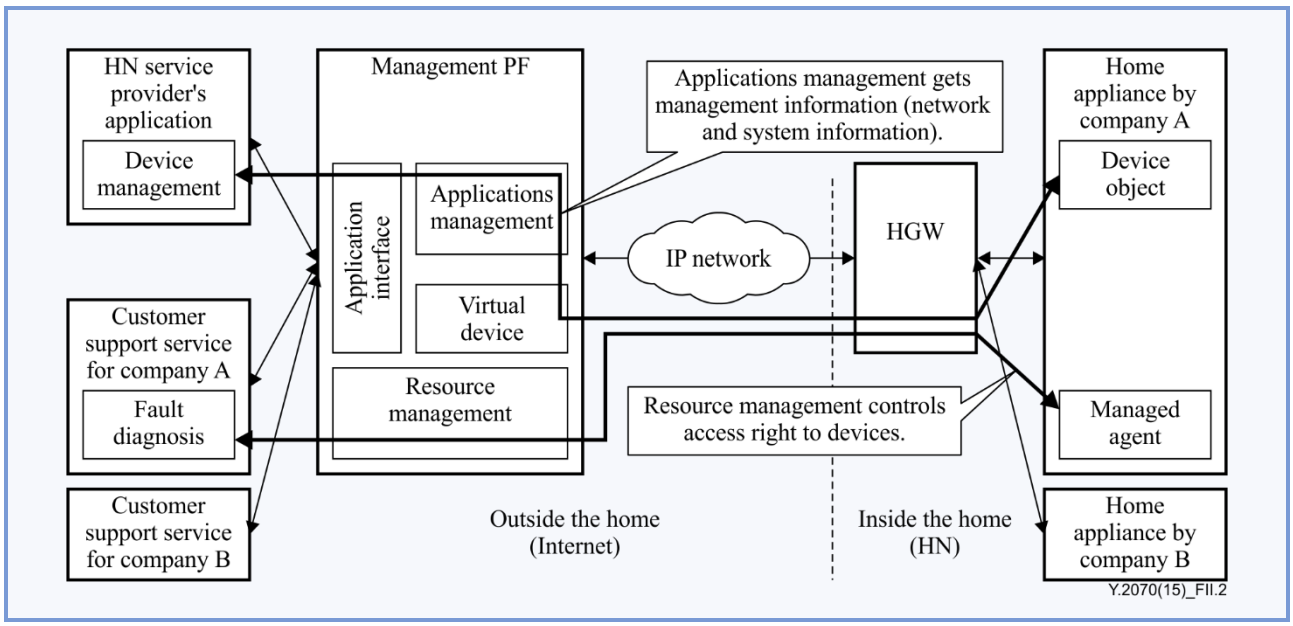


Figure II.2 – Architecture with customer support application

II.3 Room facility coordination for better sleep

The room facility coordination service for better sleep maintains a suitable environment for sleep by adjusting room temperature, humidity and illumination by controlling the air conditioner and the lighting fixture. To achieve the condition for better sleep, sleep sensors monitor the sleep pattern, heart rate, breathing rate and snoring of the user to calculate the condition for better sleep.

As the sleep sensor information is highly sensitive, the management PF strictly controls and delivers it to the proper service. In this case, the sleep monitoring service gets the data delivered from the sleep sensor exclusively. It provides the information about the suitable environment for better sleep to the room facility coordination service, as metadata, through an external interface. Then the room facility coordination service controls the air conditioner and the lighting fixture.

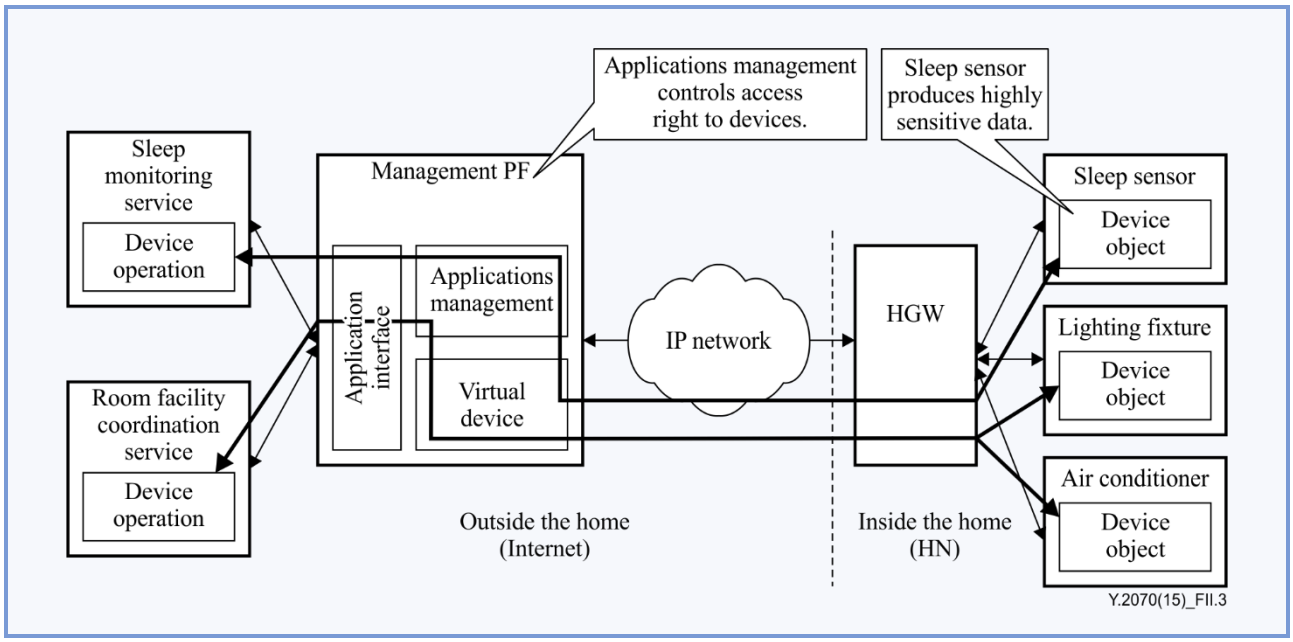


Figure II.3 – Architecture with room facility coordination application for better sleep

Appendix III

Security considerations based on [ITU-T X.1111]

(This appendix does not form an integral part of this Recommendation.)

[ITU-T X.1111] defines entities, relationship between the entities, security threats and security requirements, and describes security functions based on them. In Figure 11-1, there are six entities from the end user to the device and five relationships between them. Table III.1 shows the relationships of the security threats to the HEMS model.

Table III.1 – Relationship of security threats to HEMS model

Entity or relations	General security threats					
	Disclosure/ Eavesdropping	Interrup-tion	Modification/ Injection	Unauthorized access	Repudiation	Packet abnormal-forwarding
Device	Y	Y	Y	Y		
HGW	Y	Y	Y	Y		Y
MPF	Y	Y	Y	Y		
Application	Y	Y	Y	Y		
IHD	Y	Y	Y	Y		
User/IHD				Y		
IHD/Application	Y	Y	Y	Y		
Application/MPF	Y	Y	Y	Y		
MPF/HGW	Y	Y	Y	Y	Y	
HGW/Device	Y	Y	Y	Y	Y	
NOTE – MPF and user stand for the management PF and the end user respectively. The notation "xxx/yyy" in the column of "Entity or relations" means the relation between xxx and yyy. The letter "Y" in a cell designates that a particular threat exists for a specific entity or relation.						

Although in [ITU-T X.1111] there are descriptions about mobile-oriented security threats, these are out of the scope of this Recommendation; it focuses on the security from the HN application to the devices. Table III.2 shows the relationships between the HEMS security requirements, the threats and the functions based on the security requirements listed in [ITU-T X.1111].

Table III.2 – Relationship of HEMS security requirements, threats and functions

Security requirement	General security threats	Security functions
Data confidentiality	Disclosure/Eavesdropping Unauthorized access	Encryption Access control Key management
Data integrity	Modification/Injection Packet abnormal-forwarding	Integrity Message authentication code (MAC) Digital signature Notarization Key management
Authentication	Disclosure/Eavesdropping Interruption Modification/Injection Unauthorized access Repudiation	MAC Digital signature Notarization Key management
Non-repudiation	Repudiation	Digital signature Notarization Key management
Access control or authorization	Disclosure/Eavesdropping Interruption Modification/Injection Unauthorized access	Encryption MAC Entity authentication Digital signature Access control Key management
Availability	Interruption	MAC Entity authentication Digital signature Access control Key management Anti-availability
Privacy security	Disclosure/Eavesdropping	Encryption MAC Entity authentication Digital signature Access control Key management
Communication flow security	Packet abnormal-forwarding	Integrity MAC Entity authentication Access control Key management

The relationship between the security functions and the HEMS model for security (see Figure 11-1) are sorted in Table III.3 based on the security functions listed in Table III.2.

Table III.3 – Relationship between security functions and model

Entity or relations		Security Function								
		Encryption	Integrity	MAC	Entity authentication	Digital signature	Notarization	Access control	Key management	Anti-availability
Stored data	Device	Y	Y	Y	Y	Y	–	–	–	Y
	HGW	Y	Y	Y	Y	Y	Y	Y	Y	Y
	MPF	Y	Y	Y	Y	Y	Y	Y	Y	Y
	Application	Y	Y	Y	Y	Y	Y	Y	Y	Y
	IHD	Y	Y	Y	Y	Y	Y	Y	Y	Y
Communication data	User/IHD	Y	–	–	Y	Y	–	–	Y	Y
	IHD/Application	Y	Y	Y	Y	Y	–	Y	Y	Y
	Application/MPF	Y	Y	Y	Y	Y	Y	Y	Y	Y
	MPF/HGW	Y	Y	Y	Y	Y	–	Y	Y	Y
	HGW/Device	Y	Y	Y	–	–	–	–	–	Y

NOTE – MPF and user stand for the management PF and the end user respectively. The notation "xxx/yyy" in the column of "Entity or relations" means the relation between xxx and yyy. The letter "Y" in a cell designates that a particular security service can be provided by a corresponding security function.

When taking the relationship between the security functions and the model in Table III.3 into consideration, the security functions required for the entities in the architecture provided in this Recommendation are shown in Figure 11-2.

The security functions are simplified into three functions shown in Figure 11-2, taking [b-ISO/IEC 27000] into account. They are the anti-availability, message authentication and entity authentication function. The common functions such as encryption and key management are omitted. The other functions are merged into the three functions. These functions correspond to the requirements of the information security: availability, integrity and confidentiality specified in [b-ISO/IEC 27000].

NOTE – [b-ISO/IEC 27000] provides the information security management system. The requirements are defined as follows:

- **availability**: property of being accessible and usable upon demand by an authorized entity;
- **integrity**: property of accuracy and completeness;
- **confidentiality**: property that information is not made available or disclosed to unauthorized individuals, entities or processes.

Bibliography

[b-ITU-T J.190] Recommendation ITU-T J.190 (2002), *Architecture of MediaHomeNet that supports cable-based services*.

[b-ITU-T Y.2063] Recommendation ITU-T Y.2063 (2012), *Framework of the web of things*.

[b-ITU-T Y.2720] Recommendation ITU-T Y.2720 (2009), *NGN identity management framework.*

[b-ISO/IEC 14543-3-x] ISO/IEC 14543-3-x:2006/2007, *Information technology – Home Electronic System (HES) architecture – Part 3-x*.

[b-ISO/IEC 27000] ISO/IEC 27000:2014, *Information technology – Security techniques – Information security management systems – Overview and vocabulary*.

[b-BACnet] ANSI/ASHRAE, *Standard 135-2004*.

[b-BBF TR-069] Broadband Forum (2011), *TR-069 Amendment, 4CPE WAN Management Protocol.*

[b-BBF TR-181] Broadband Forum (2012), *TR-181 Issue 2 Amendment 6, Device Data Model for TR-069.*

[b-ECHONET Lite] ECHONET Consortium, *ECHONET Lite Specification Version 1.10.*

[b-FG-Smart Terminology] ITU-T FG Smart Deliverable (2011), *Smart Grid Terminology*.

[b-SEP 2.0] ZigBee Alliance, *Smart Energy Profile 2.0 Application Protocol.*

[b-W3C WCterms] W3C (1999), *Web Characterization Terminology & Definitions Sheet.*

SMART HOME

security lights garage thermostat locks
You can remotely monitor your...

Y.4410/Y.2291

Architectural overview of next generation home networks

Architectural overview of next generation home networks

Summary

Recommendation ITU-T Y.2291 provides an architectural overview of next generation home networks (NGHN). In line with Recommendation ITU-T Y.2011 and Recommendation ITU-T Y.2012 principles, an implementation-independent approach is adopted.

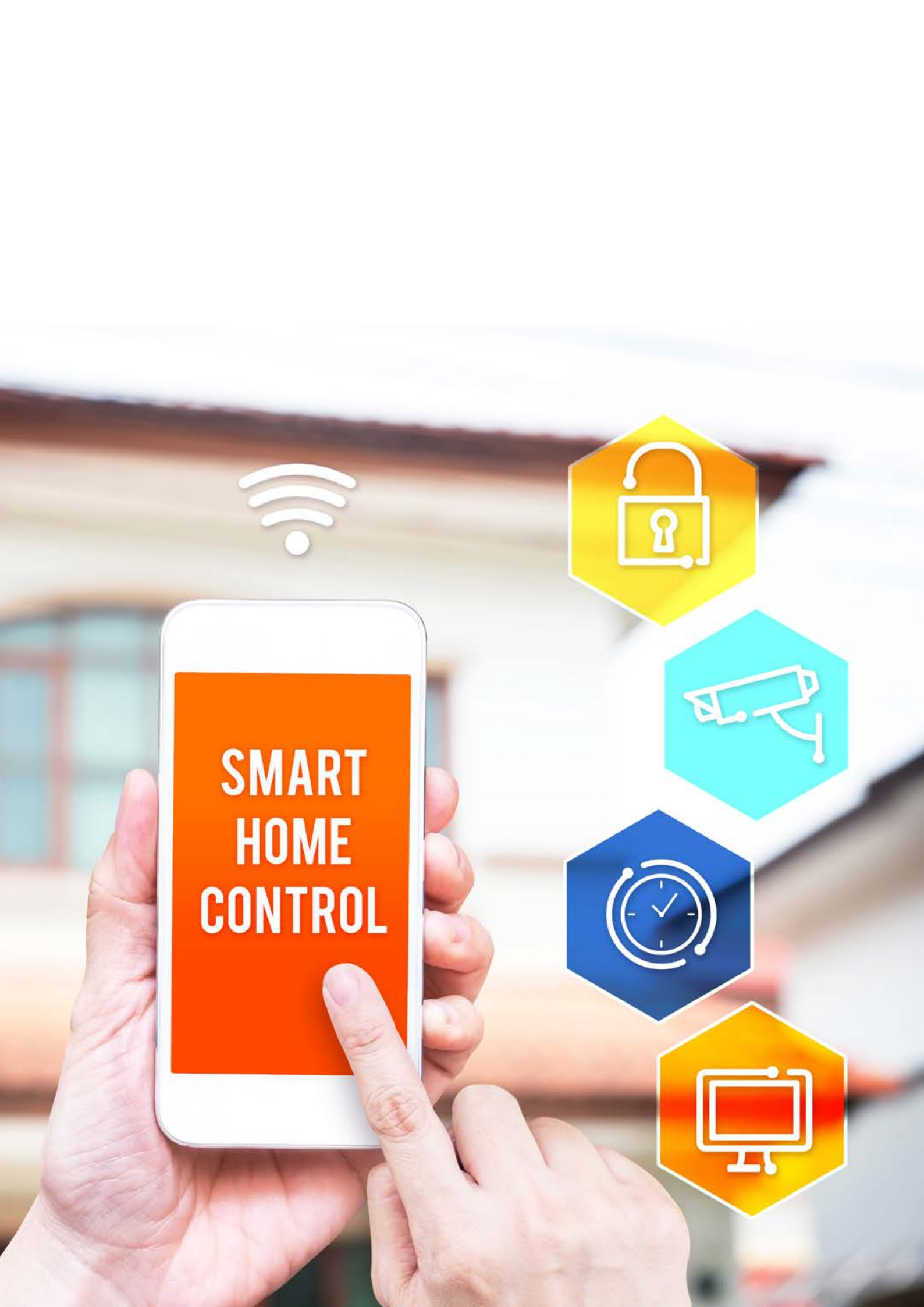
History

Edition	Recommendation	Approval	Study Group
1.0	ITU-T Y.2291	2011-01-28	13

Keywords

Home network, next generation home network (NGHN), next generation network (NGN).

Table of Contents


SMART
HOME
CONTROL

Recommendation ITU-T Y.4410/Y.2291

Architectural overview of next generation home networks

1 Scope

The objective of this Recommendation is to provide the architectural overview of next generation home networks (NGHN) identifying overall features and functions of home network using an implementation independent approach using [ITU-T Y.2011] and [ITU-T Y.2012] principles.

NGHN are intended to support NGN capabilities as per [ITU-T Y.2201].

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T H.622] Recommendation ITU-T H.622 (2008), *A generic home network architecture with support for multimedia services*.

[ITU-T X.1111] Recommendation ITU-T X.1111 (2007), *Framework of security technologies for home network*.

[ITU-T Y.2011] Recommendation ITU-T Y.2011 (2004), *General principles and general reference model for Next Generation Networks*.

[ITU-T Y.2012] Recommendation ITU-T Y.2012 (2010), *Functional requirements and architecture of next generation networks*.

[ITU-T Y.2111] Recommendation ITU-T Y.2111 (2008), *Resource and admission control functions in next generation networks*.

[ITU-T Y.2201] Recommendation ITU-T Y.2201 (2009), *Requirements and capabilities for ITU-T NGN*.

[ITU-T Y.2701] Recommendation ITU-T Y.2701 (2007), *Security requirements for NGN release 1*.

[ITU-T Y.2720] Recommendation ITU-T Y.2720 (2009), *NGN identity management framework*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 federation [ITU-T Y.2720]: Establishing a relationship between two or more entities or an association comprising any number of service providers and identity providers.

3.1.2 home network [ITU-T H.622]: A home network is the collection of elements that process, manage, transport and store information, thus enabling the connection and integration of multiple computing, control, monitoring, communication and entertainment devices in the home.

NOTE – In this Recommendation, entities are end users, terminals and services.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 home network application network interface: Interface between home network applications and the next generation home network (NGHN).

3.2.2 home network terminal network interface: Interface between terminal equipment and the next generation home network (NGHN).

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

ANI	Application Network Interface
ASF&SSF	Application Support Function and Service Support Functions
CDF	Content Delivery Functions
H-ANI	NGHN ANI
H-ASF&SSF	NGHN ASF&SSF
H-CDF	NGHN CDF
H-IdM	NGHN IdM
H-MF	NGHN Management Functions
H-MMCF	NGHN Mobility Management Control Functions
HN	Home Network
H-NACF	NGHN NACF
H-RACF	NGHN RACF
H-SCF	NGHN SCF
H-TCF	NGHN Transport Control Functions
H-TrF	NGHN Transport Functions
H-TNI	Home network Terminal Network Interface
IdM	Identity Management
IP	Internet Protocol
ISDN	Integrated Services Digital Network
NACF	Network Attachment Control Functions
NAT	Network Address Translation
NGHN	Next Generation Home Network
NGN	Next Generation Network
PSTN	Public Switched Telephone Network
QoS	Quality of Service
RACF	Resource and Admission and Control Functions
SCF	Service Control Functions
SLA	Service Level Agreement
TF	Terminal Functions
UNI	User Network Interface

5 Conventions

None.

6 Overview of next generation home network (NGHN)

6.1 General characteristics of next generation home network (NGHN)

Next generation home network (NGHN) aims at providing the following characteristics:

- Packet-based transfer, in particular support of Internet protocol (IP) as the protocol used at layer 3 in NGHN;
- NGHN can be viewed as an IP-based home network;
- User access to a wide range of services and applications (including real time/non-real time and multimedia services);
- Seamless environment for acquiring, sharing, storing and accessing digital media and content within the home network;
- Use of multiple broadband (wired and/or wireless), QoS-enabled transport technologies;
- Support of fixed and mobile terminals, including support of legacy terminals (e.g., PSTN/ISDN terminals);
- Automatic discovery and management of terminals attached to the home network.

Thus, features in NGHN architecture are to enhance home network capabilities described in [ITU-T H.622].

6.2 Connectivity to the NGHN

Figure 6-1 shows the different connectivity provided by a next generation home network (NGHN).

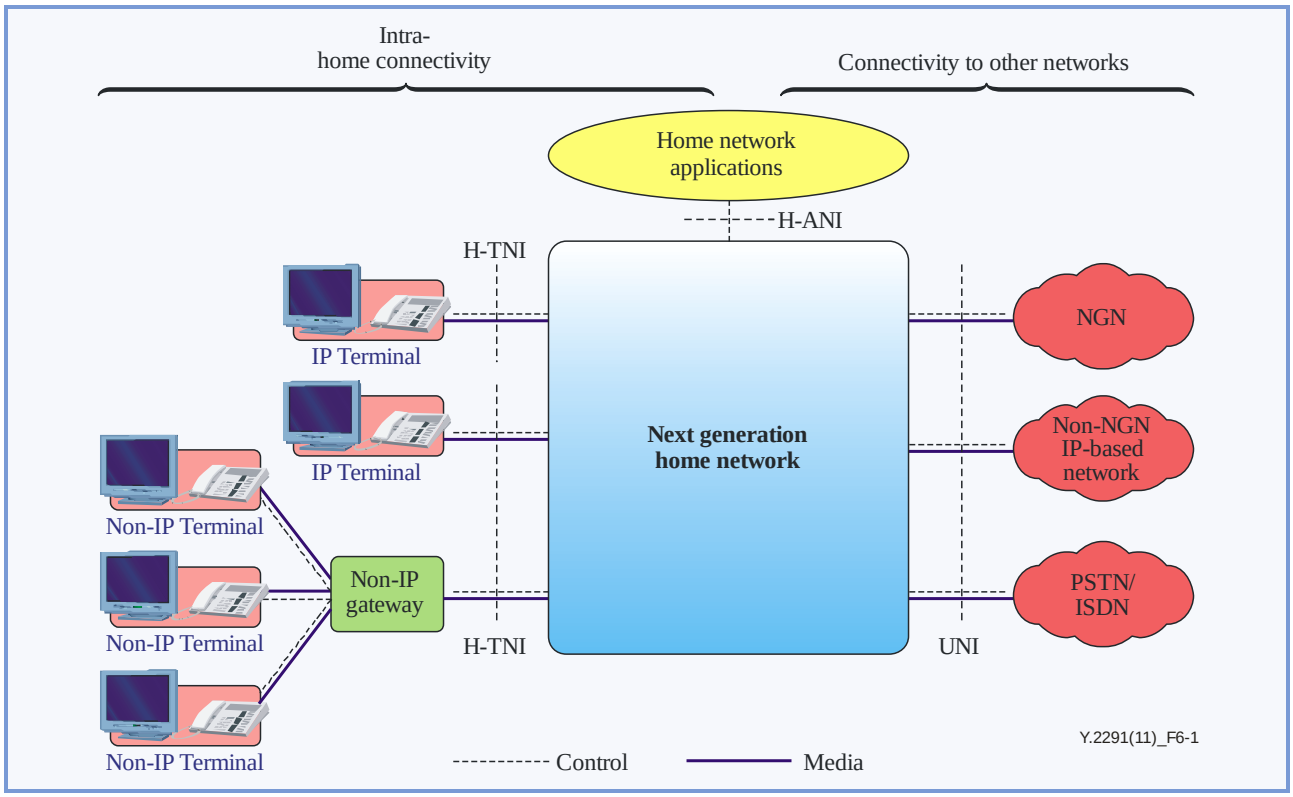


Figure 6-1 – Connectivity to NGHN

NGHN provides two types of connectivity:

- **Intra-home connectivity** covers the connectivity of terminals within the home network. This includes connectivity of IP terminals at the home network terminal network interface (H-TNI) and connectivity of non-IP terminals via a non-IP gateway at the H-TNI;
- **Connectivity to other networks** covers the connectivity of NGHN at the UNI to other external networks such as NGN, non-NGN IP-based networks or PSTN/ISDN.

Considering these two types of connectivity, there are two major roles for the home network, i.e., extending the other networks (such as NGN) and their access network as well as interconnecting terminals in the NGHN itself. The portion working as an extension of the access networks must be aligned with the technical requirements of the network provider.

Requirements regarding connection and access to the NGN are captured in particular in clauses 17.1 and 17.2 of [ITU-T Y.2201].

7 Overview of the NGHN architecture

7.1 Overview of functional framework

Figure 7-1 shows an overview of the NGHN architecture. The NGHN architecture follows a similar decomposition as the one defined for the NGN architecture as defined in [ITU-T Y.2012].

The NGHN functional architecture supports the UNI, H-TNI and H-ANI reference points shown in Figure 6-1.

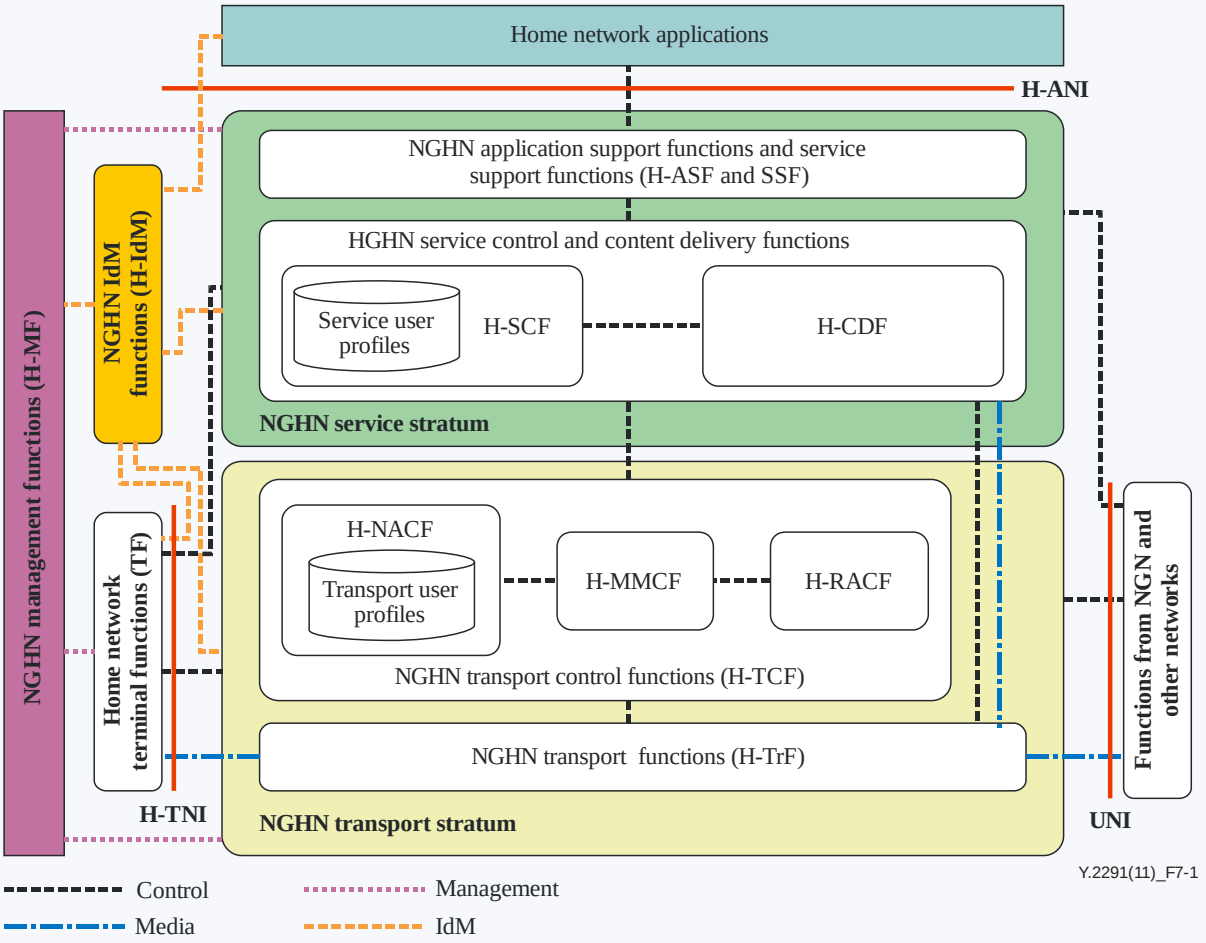
The NGHN functions are divided into service stratum functions and transport stratum functions according to the principles described for NGN in [ITU-T Y.2011].

The NGHN service stratum provides the user functions that transfer service-related data and the functions that control and manage service resources and network services to enable user services and applications.

The NGHN transport stratum provides the user functions that transfer data and the functions that control and manage transport resources to carry such data between terminating entities.

The delivery of services/applications to the end user is provided by utilizing the application support functions and service support functions and related control functions.

The transport stratum provides IP connectivity services to NGHN users under the control of transport control functions within the NGHN, including the network attachment control functions (H-NACF), resource and admission control functions (H-RACF) and mobility management and control functions (H-MMCF).



NOTE 1 – The user network interface (UNI), NGHN application network interface (H-ANI) and home network terminal network interface (H-TNI) are to be understood as general NGHN reference points that can be mapped to specific physical interfaces depending on the particular physical implementations.

NOTE 2 – Boxes in Figure 7-1 identify high level functional groups, for which overall descriptions are given later in this clause.

NOTE 3 – The control links between the functional groups represent high-level logical interactions.

NOTE 4 – User profiles in both the service stratum and the transport stratum are shown as separate functional databases. Depending on the implementation in place, these two functional databases can optionally be co-located.

NOTE 5 – The NGN-UNI line shows the functional aspect only and should not make any pre-decision about an ownership domain.

NOTE 6 – It is possible for NGHN identity management functions (H-IdM) to reside in different planes (e.g., user, control and management) and different strata of the distributed architecture (e.g., service stratum and transport stratum). Although IdM functions are shown in a standalone group of functions, this is not intended to impose any implementation design and restrictions for IdM.

NOTE 7 – Although IdM functions are shown on the left side of Figure 7-1, this does not mean that IdM functions are located on the H-TNI side or part of the terminal functions.

NOTE 8 – Although not shown in Figure 7-1, IdM functions can be connected to functions from other networks using the UNI reference point.

Figure 7-1 – NGHN architectural overview

7.2 NGHN functions at transport stratum

The transport stratum functions include transport functions and transport control functions in [ITU-T Y.2011].

7.2.1 NGHN transport functions (H-TrF)

The H-TrF provide the connectivity for all components and physically separated functions within the NGHN. These functions provide the support for unicast and/or multicast transfer of media information, as well as the transfer of control and management information.

The H-TrF also provide capabilities to interwork with terminals and/or other networks (such as NGN, non-NGN IP based networks, PSTN/ISDN).

7.2.2 NGHN transport control functions (H-TCF)

7.2.2.1 NGHN attachment control functions (H-NACF)

The H-NACF provide transport stratum level identification/authentication, manage the IP address space of the NGHN, and authenticate access sessions. These functions may also announce the contact point of NGHN functions in the service stratum to the terminal. The H-NACF provides the following functionalities:

- dynamic provisioning of IP addresses and other user equipment configuration parameters;
- provisioning of non-IP gateway connecting to non-IP terminals;
- by endorsement of end-user, auto-discovery of user terminal capabilities and other parameters;
- authentication of end-user/terminal and home network at the IP layer (and possibly other layers).

The H-NACF include the transport user profile which takes the form of a functional database representing the combination of a user/terminal's information and other control data.

7.2.2.2 NGHN resource and admission control function (H-RACF)

Within the NGHN architecture, the resource and admission control functions (RACF) act as the arbitrator between service control functions and transport functions for QoS-related transport resource control. The decision is based on transport subscription information, service level agreements (SLAs), network policy rules, service priority, and transport resource status and utilization information.

The RACF provides an abstract view of transport network infrastructure to service control functions (SCF) and makes service stratum functions agnostic to the details of transport facilities such as network topology, connectivity, resource utilization and QoS mechanisms/technology.

The H-RACF provides real-time application-driven and policy-based transport resource management for a wide range of services and for a variety of transport technologies in the NGHN.

7.2.2.3 NGHN mobility management and control functions (H-MMCF)

The H-MMCF provide functions for the support of IP-based/non-IP mobility in the transport stratum. These functions allow the support of mobility of a terminal. H-MMCF provide mechanisms to achieve seamless mobility among various terminals with heterogeneous interfaces and different coverage.

7.3 NGHN functions at service stratum

7.3.1 NGHN service control functions (H-SCF)

The H-SCF include resource control, registration, and authentication and authorization functions at the service level for both mediated and non-mediated services. They can also include functions for controlling media resources, i.e., specialized resources and gateways at the service-signalling level.

The H-SCF accommodate service user profiles which represent the combination of user information and other control data into user profile function in the service stratum, in a form of functional databases. These functional databases may be specified and implemented as a set of cooperating databases with functionalities residing in any part of the NGHN.

7.3.2 NGHN content delivery functions (H-CDF)

The H-CDF store, process, and deliver contents to the terminal functions under control of the H-SCF.

7.3.3 NGHN application support functions and service support functions (H-ASF&SSF)

The H-ASF&SSF include functions such as registration, authentication and authorization functions at application level within the NGHN. These functions are available to the "home network applications" and "terminal" functional groups in NGHN. The H-ASF&SSF work in conjunction with the H-SCF to provide end users and applications with the services within NGHN.

Through the H-TNI, the H-ASF&SSF provide reference points to the terminal functions. Application interactions with the H-ASF&SSF are handled through the H-ANI reference point.

7.4 NGHN management functions (H-MF)

Support for management is fundamental to the operation of the NGHN. These functions provide the capabilities to manage the NGHN in order to provide services with the expected quality, security, and reliability.

Management functions apply to the NGHN service and transport strata. For each of these strata, they cover: QoS management, security management, performance monitoring and diagnostics and troubleshooting, terminal management and accounting management.

7.4.1 QoS management function

The QoS management function supports:

- QoS-related transport resources management within the home network with incorporation of RACF in NGN;
- application-driven QoS management for the home network;
- per-flow, per-session, per-service-class QoS control granularity.

7.4.2 Security management function

For protection of unauthorized access into the home network and provision of the privacy of data, security management function provides manageable security to enhance the confidence of the end user through the firewall and network address translation (NAT) capabilities. Optionally, it provides the ability to hide terminals from the service provider so as not to have full visibility of the home network.

7.4.3 Performance monitoring, diagnostics and troubleshooting function

System level fault (e.g., hardware, operating system and software related) can optionally be detected and communicated to the service or network provider. Performance monitoring, diagnostics and troubleshooting supports:

- remote diagnostic tests to check the state of the different components of the home network;

  NOTE – These tests are either scheduled periodically or launched by system-operator request.
- performance monitoring to see statistics at network level;
- generation of event to detect a possible fault within the system.

7.4.4 Terminal management function

Terminal management function provides capabilities to manage and control terminals in the NGHN. Terminal management capabilities are used for:

- configuration management, such as terminal hardware information, media capabilities, software version;
- local performance monitoring and maintenance;
- remote fault diagnosis;
- remote identification of manageable terminal.

7.4.5 Accounting management function

Accounting management function identifies who is using the resources of the NGHN and to what extent, and allocates cost to those users on the basis of their usage. It supports the users of the NGHN resources to absorb the cost thereof in an agreed-upon manner.

7.5 NGHN identity management functions (H-IdM)

The H-IdM are related to service features to be provided. The H-IdM include the following functions in accordance with [ITU-T Y.2720]:

- identification of federations for home network services;
- identification of customer-oriented service for fairly new service concept and model;
- provision of abstract identity framework for customer-oriented dynamic identification;
- provision of identity for alliance between home network users to cooperate with;
- provision of home network trust with identity to guarantee privacy.

7.6 Home network terminal functions (TF)

The TF are related to various types of terminals including IP terminals/non-IP terminals. These terminals have heterogeneous interfaces, including fixed/mobile for accessing NGHN. These functions support capabilities to provide connectivity with NGHN and support various services through H-TNI.

8 Security considerations

Major security requirements for NGHN are:

- authentication of the communication entities for policy requests in NGHN;
- data confidentiality and integrity among NGHN users;
- availability and accessibility in NGHN, upon demand by an authorized entity;
- availability of mechanisms of non-repudiation for preventing one of the entities or parties in communication from denying participation in the whole or part of the communication falsely.

The security considerations in the NGHN should be in accordance with the security requirements in [ITU-T X.1111] and [ITU-T Y.2701].

Appendix I

Federation of NGHNs

(This appendix does not form an integral part of this Recommendation.)

Figure I.1 shows a configuration of a federation of NGHNs. The home network domain comprises heterogeneous home environments with NGHN functions. The NGHN functions support multiple different capabilities in accordance with needs of home network users and a federated configuration among multiple entities in the home networks.

The federation between/among multiple entities in the NGHNs creates physical, logical group environments or service/user community.

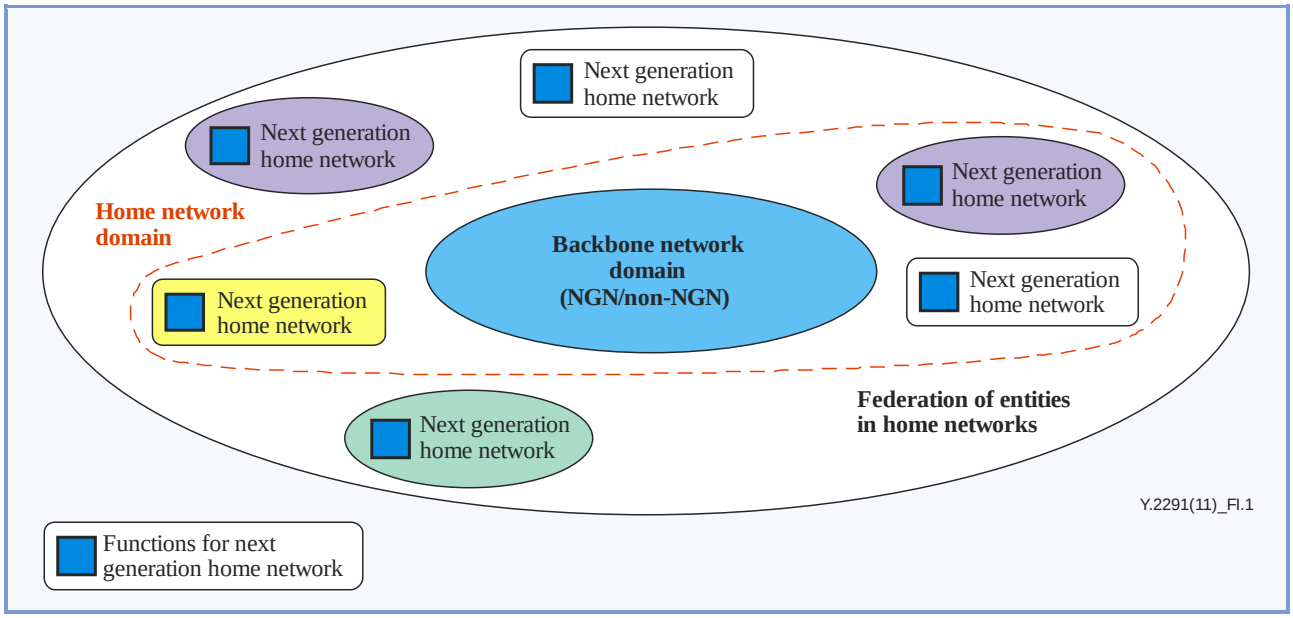


Figure I.1 – Configuration of a federation of NGHNs

NOTE – The home network domain includes several NGHNs. Home network domain and NGN/non-NGN backbone network domain are involved to create a federation of entities in home networks.

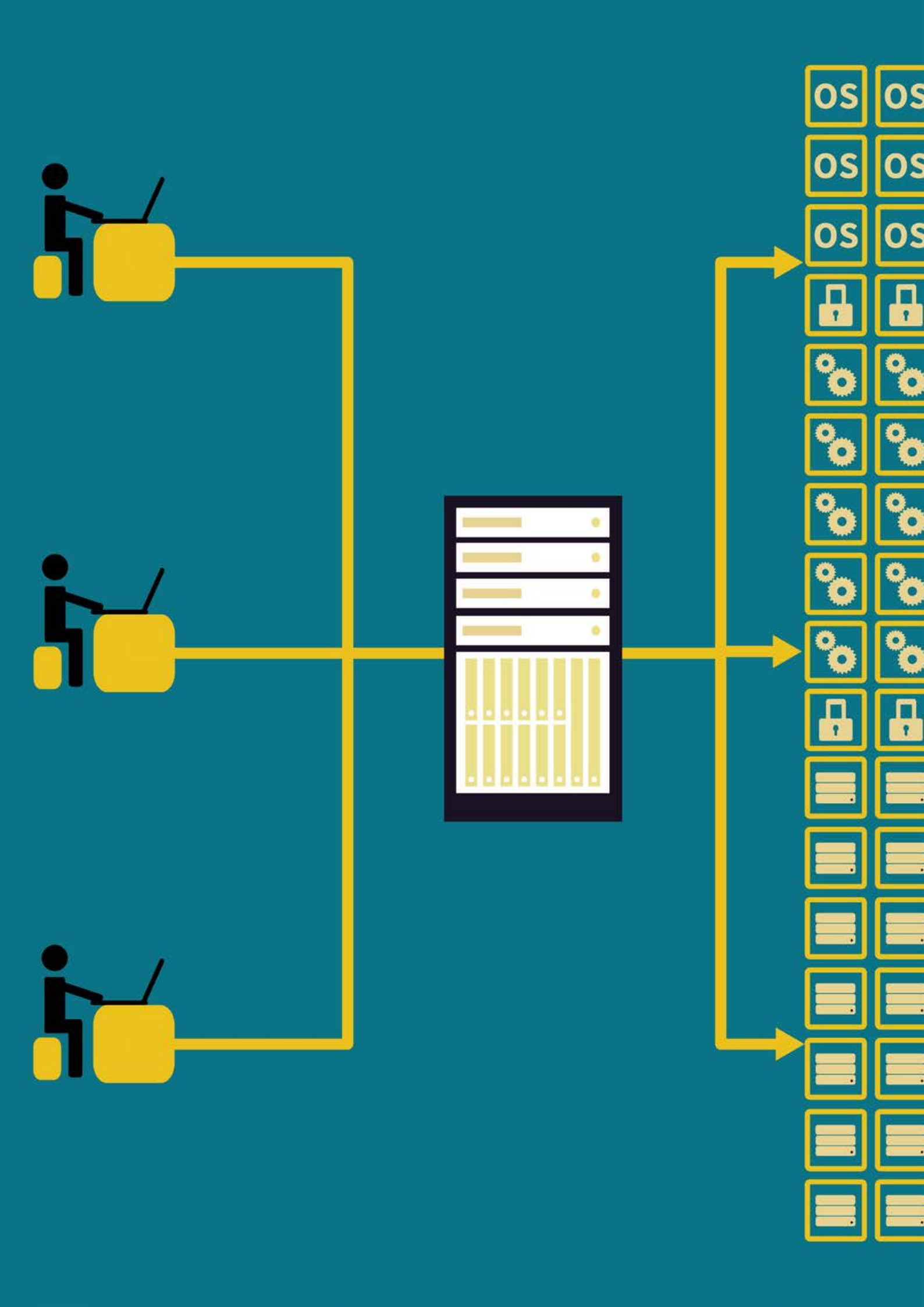
Bibliography

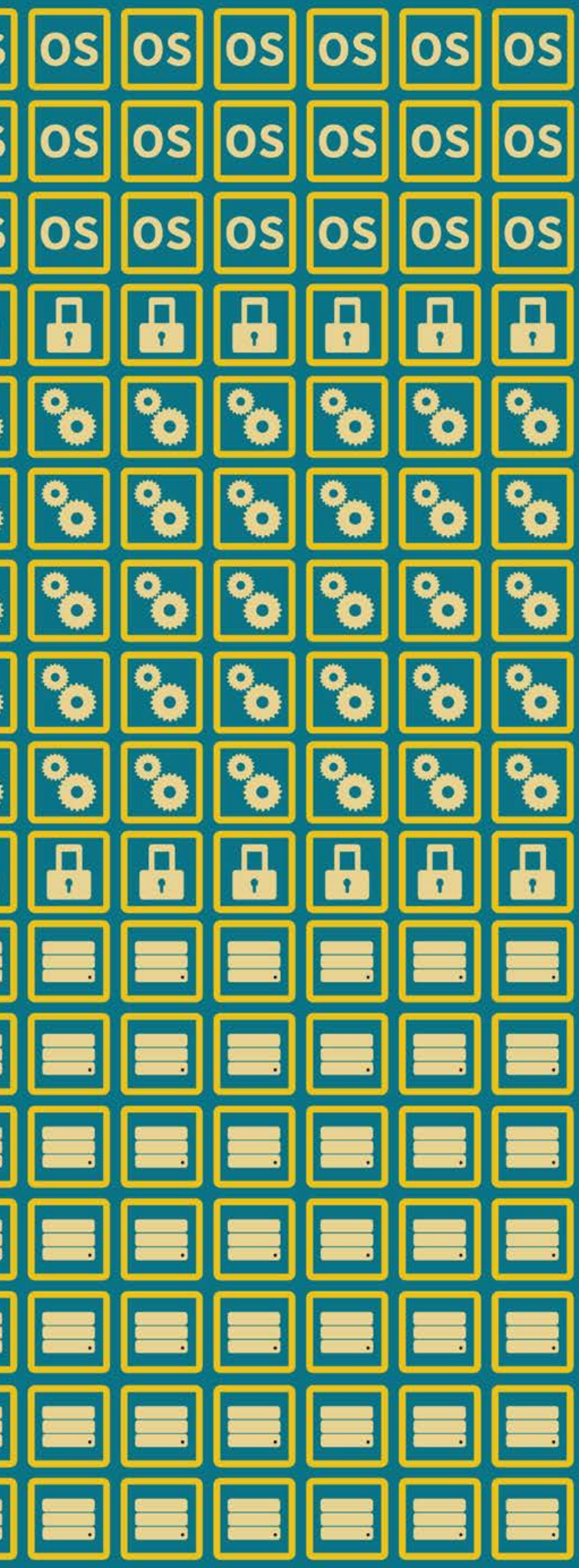
[b-ITU-T G.9970] Recommendation ITU-T G.9970 (2009), *Generic home network transport architecture.*

[b-HGI] *Home Gateway Technical Requirements Residential Profile, Version 1.0.1,* (2008).

[b-TR-069] BroadBand Forum TR-069 Amendment 3 (2010), *CPE WAN Management Protocol*.

[b-TR-094] BroadBand Forum TR-094 (2004), *Multi-Service Delivery Framework for Home Networks*.


OS
OS
OS
OS
OS
OS



Y.4411/Q.3052

Overview of application programming interfaces and protocols for the machine-to-machine service layer

Overview of application programming interfaces and protocols for the machine-to-machine service layer

Summary

Recommendation ITU-T Y.4411/Q.3052 describes an overview of application programming interfaces (APIs) and protocols for the machine-to-machine (M2M) service layer and the related API and protocol requirements. It describes the component-based M2M reference model, including the reference points of the M2M service layer. APIs and protocols for M2M are introduced, including existing APIs and protocols for M2M service layer and M2M protocol structure and stacks. Finally, general requirements of APIs and protocols with respect to the M2M service layer are described.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T Y.4411/Q.3052	2016-02-13	11	11.1002/1000/12698

Keywords

API, application programming interface, M2M, machine-to-machine, protocol.

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

Table of Contents


IDEA
RANK
Social
evolution
idea
FIND MORE
DISCOVER
APP DEVELOPMENT
Lorem ipsum dolor sit amet consectetur adipiscing elit
Lorem ipsum dolor sit amet consectetur adipiscing elit
USB
LAPTOP
STORAGE
PC
CLOUD
RANK
DEVICE
IT
Lorem ipsum dolor sit amet consectetur adipiscing elit
Lorem ipsum dolor sit amet consectetur adipiscing elit
Lorem ipsum dolor sit amet consectetur adipiscing elit

Recommendation ITU-T Y.4411/Q.3052

Overview of application programming interfaces and protocols for the machine-to-machine service layer

1 Scope

This Recommendation provides an overview of APIs and protocols for the M2M service layer and the related API and protocol requirements. First, it describes the component-based M2M reference model, including the reference points of the M2M service layer. Then, APIs and protocols for M2M are introduced, including existing APIs and protocols for M2M service layer and M2M protocol structure and stacks. Finally, API and protocol general requirements with respect to the M2M service layer are described.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T Y.4000] Recommendation ITU-T Y.4000/Y.2060 (2012), *Overview of the Internet of things*.

[IETF RFC 2045] IETF RFC 2045 (1996), *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies*.

[IETF RFC 2616] IETF RFC 2616 (1999), *Hypertext Transfer Protocol – HTTP/1.1*.

[IETF RFC 6455] IETF RFC 6455 (2011), *The WebSocket Protocol*.

[IETF RFC 6749] IETF RFC 6749 (2010), *The OAuth 2.0 Authorization Framework*.

[IETF RFC 6886] IETF RFC 6886 (2013), *NAT Port Mapping Protocol (NAT-PMP)*.

[IEEE 11073-20601] Health informatics – Personal health device communication – Part 20601 (2010), *Application profile – Optimized exchange protocol*.

[DPWS] OASIS standard (2009), *Devices Profile for Web Services (DPWS)*.

[SOAP] W3C Recommendation SOAP V.1.2 (2007), *Simple Object Access Protocol*.

[XML 1.1] W3C Recommendation XML 1.1 (2006), *Extensible Markup Language (XML) 1.1 (Second Edition)*.

3 Definitions

3.1 Terms defined elsewhere

None.

3.2 Terms defined in this Recommendation

This Recommendation defines the following term:

3.2.1 application programming interface (API): A particular set of rules and specifications that a software program can follow to access and make use of the services and resources provided by another software program or set of resources.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

API	Application Programming Interface
CSP	Community Service Provider
CoAP	Constrained Application Protocol
DA	Device Application
DPWS	Devices Profile for Web Services
GA	Gateway Application
HTML	Hyper Text Markup Language
HTTP	Hypertext Transfer Protocol
HVAC	Heating, Ventilation, and Air Conditioning
IGD	Internet Gateway Device
IoT	Internet of Things
LAN	Local Area Network
M2M	Machine-to-Machine
M2M SL	Machine-to-Machine Service Layer
MIME	Multipurpose Internet Mail Extensions
MTU	Master Terminal Unit
NA	Network Application
NAT	Network Address Translation
NAT-PMP	NAT Port Mapping Protocol
PKI	Public Key Infrastructure
RDF	Resource Description Framework
REST	Representational State Transfer
RFID	Radio Frequency Identification
ROA	Resource-Oriented Architecture
RTU	Remote Terminal Unit
SAO	Service-Oriented Architecture
SCADA	Supervisory Control and Data Acquisition
SLA	Service Level Agreement
SOAP	Simple Object Access Protocol
TCP	Transmission Control Protocol

UDP	User Datagram Protocol
UPnP	Universal Plug and Play
URI	Uniform Resource Identifier
WAN	Wide Area Network
WSDL	Web Services Description Language
XML	Extensible Markup Language
6LoWPAN	IPv6 over Low power Wireless Personal Area Networks

5 Conventions

None.

6 General introduction

This decade is widely predicted to see the rise of machine-to-machine (M2M) communications over various networks. The M2M opportunity is diverse, dynamic and is rapidly expanding. It provides connectivity for a huge variety of different devices and machines including utility meters, vehicles, sensors point of sale terminals, security devices, healthcare devices and many more. Every day, objects are becoming machines that can be addressed, recognized, localized and controlled via communication networks and platforms.

An application programming interface (API) is a set of rules ('code') and specifications that software programs can follow to communicate with each other. It serves as an interface between different software programs and facilitates their interaction, similar to the way a user interface facilitates interaction between humans and computers.

In the M2M world, APIs provide the level of abstraction necessary to implement interactions uniformly. The data exchanged are inherently related to the protocol stack used in the communication.

A protocol is the special set of rules that end points in a communication network use when they communicate with each other. It is expected that M2M applications utilize standardized protocols in order to be widely deployable. Depending on the needs of the M2M application, different protocols may be utilized.

7 Component-based M2M reference model and its relationship with M2M service layer

7.1 Component based M2M reference model

The component-based M2M reference model is shown in Figure 1. It can be decomposed in five main components and one super-component:

- **device**: The component that hosts the device applications. It can connect directly, or via the gateway, to the network. It may host M2M service layer (M2M SL) capabilities. When it does not contain the M2M SL capabilities, it is considered as a legacy device.
- **gateway**: A component that may host M2M SL capabilities and gateway applications (GA) and acts as an intermediary between the network and a legacy device.
- **network**: A component, which does not host M2M SL capabilities, and that connects device, gateway and network application server with each other.
- **M2M platform**: A component that hosts the M2M SL capabilities, and that can be used by one or more application servers. The M2M platform is part of the network application server.

  NOTE 1 – Clauses 7.2 and 7.3 provide some details about M2M platforms, as they play a relevant role from a protocol/API point of view.

- **application server**: A component that hosts the network applications. The application server is also part of the network application server.
- **network application server**: A super-component including both the application server and the M2M platform.

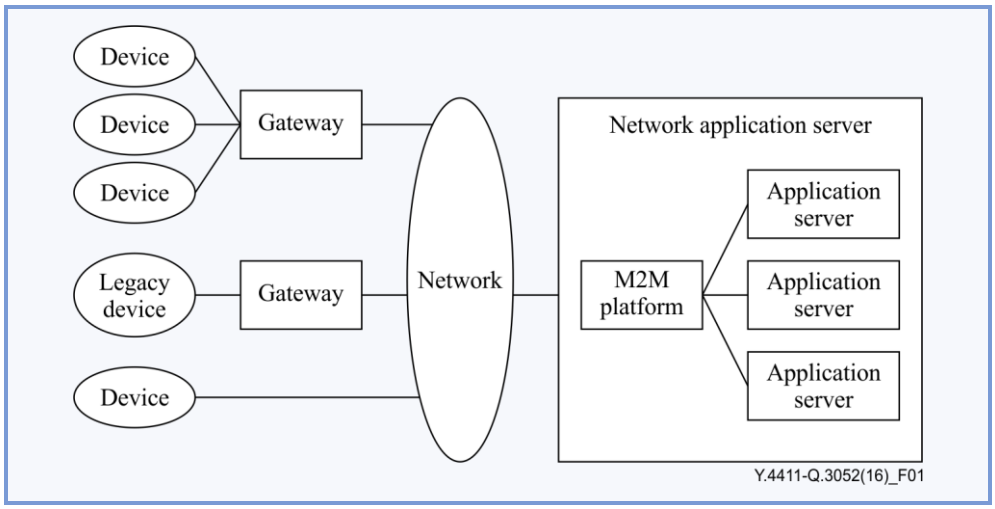


Figure 1 – Component based M2M reference model

The different elements of the component-based M2M reference model can communicate among themselves using capabilities of multiple layers.

NOTE 2 – [ITU-T Y.4000] defines the Internet of things (IoT) reference model as having four layers (application layer, service support and application support layer, network layer, and device layer).

7.2 M2M platforms

Traditionally, M2M solutions have been conceived and deployed as 'stovepipe' (or standalone) solutions with the aim of improving (or enabling) a specific process, but without consideration of how these solutions might one day be integrated into a wider business context.

Today, M2M solutions are doing more than just monitoring the status of remote assets and equipment. They are gathering real-time data from millions of connected machines, for example, tractors, medical devices, vending machines, or storage tanks, and translating them into meaningful information for quick decisions, automated actions and strategic analytics. The primary driver for M2M solutions is now enabling new services, rather than just improving operational efficiency/cost saving.

A platform is considered to be a group of technologies that are used as a base upon which applications, processes or other technologies are developed/delivered.

Creating a platform is usually a complex and delicate task, and needs to serve multiple purposes, but the primary purpose is to support and simplify the work of those who will be using/consuming this platform. M2M platforms have transformed the M2M market by making device data more accessible to application developers, and also by offering well-defined software interfaces and making APIs available so that application developers can readily integrate information sources and control parameters into their applications.

7.3 Types of M2M platforms

Over the past decade, the M2M platform space has developed rapidly, and now includes the following broad platform functions:

- connectivity support: It encompasses all of the most fundamental tasks that must be undertaken to configure and support a machine-to-machine connection. In a mobile environment, such tasks include connection provisioning, usage monitoring and some level of support for fault resolution.
- service enablement: It has extensive capabilities in terms of solution support, reporting and provision of a software environment and APIs to facilitate solution development. Together, connectivity support and service enablement functions represent the 'horizontal' elements of the M2M platforms industry.
- device management: It has typically been aligned to single device manufacturers and potentially supports devices of multiple types and vendors connected through multiple networks. Device management platforms essentially exist to facilitate sales of devices (and device-centric solutions) where those devices typically require some form of non-standard systems support (reporting, management, etc.).
- application support: It is characterized by the provision of tailored solutions, encompassing connected devices potentially of multiple types, connected with multiple technologies, and connected to the networks of multiple communication service providers (CSPs).
- solution provider: It should be regarded typically as an enabler for a large system development initiative, rather than as a standalone offering. These M2M platforms are generally used by systems integrators to support turnkey and client-specific solutions.

While many of these platform type implementations have some overlapping functionality (further complicating the M2M delivery ecosystem while simultaneously attempting to streamline it), the end goal is similar, i.e. mainly to make sure that the data collected from all of these machines and sensors are actually used to improve the business of the company investing in the collection.

7.4 Reference points of the M2M service layer in the component-based M2M reference model

The purpose of this Recommendation is to provide overview of APIs and protocols handled by M2M service layer. It is necessary to make clear the reference points used by APIs and protocols. It is possible to clarify the reference points by referring to [ITU-T Y.4000].

Figure 2 highlights the reference points that are in the scope of this Recommendation based on [ITU-T Y.4000]. Figure 2 focuses on reference points of the M2M service layer from a functional point of view.

There are three types of M2M applications on top of the M2M service layer: device application (DA), gateway application (GA) and network application (NA). DA, GA and NA reside, respectively, in device, gateway and network application server. All these applications use capabilities provided by the M2M service layer. These three types of applications are shown in the application layer in Figure 2.

Four reference points are identified for the ITU-T M2M service layer: D-SL, G-SL, A-SL and SL-SL.

- D-SL: reference point between DA and M2M service layer
- G-SL: reference point between GA and M2M service layer
- A-SL: reference point between NA and M2M service layer
- SL-SL: reference point between different M2M service layers

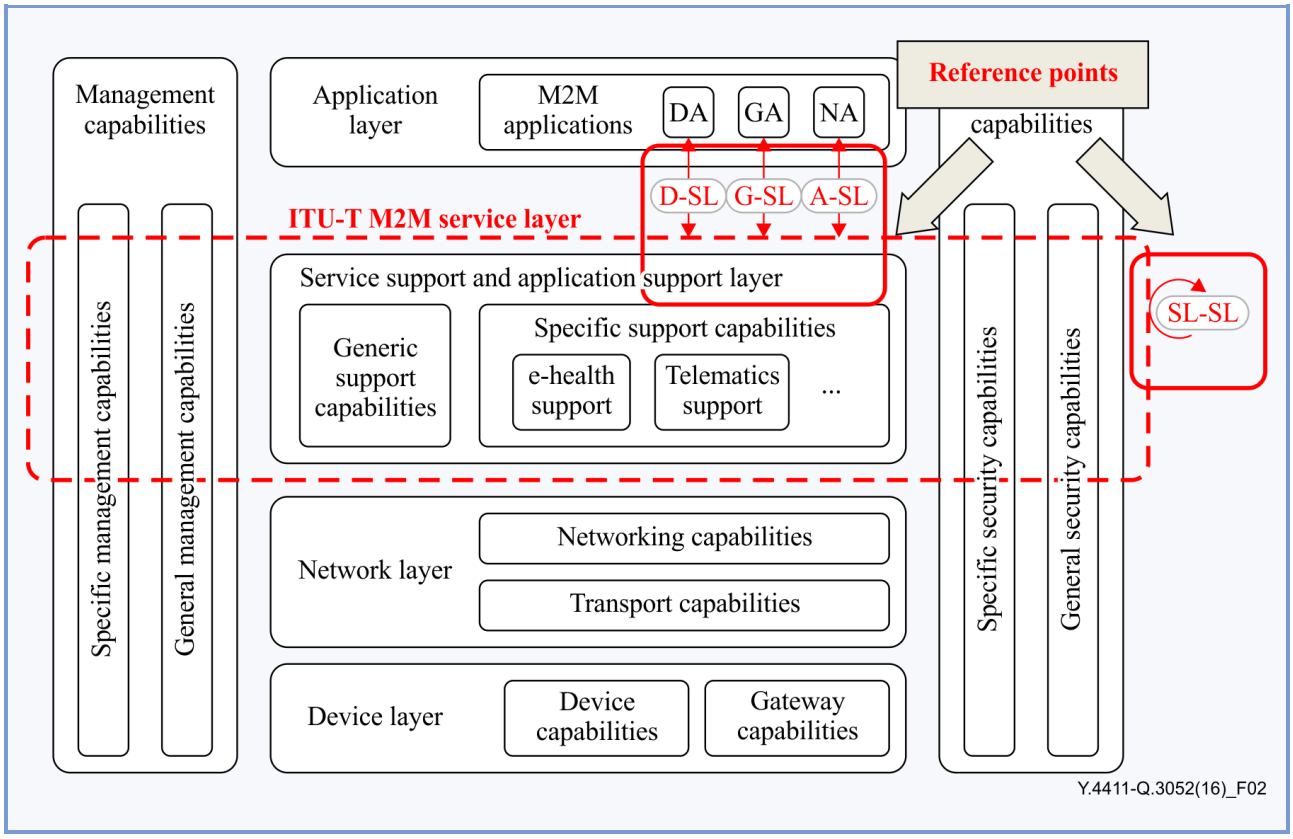


Figure 2 – Reference points of the ITU-T M2M service layer

The four references points are shown in Figure 3 with respect to the component-based M2M reference model.

NOTE – Figure 3 shows the general case of devices providing also M2M SL capabilities (simply called "the SL function" in the following part of this Recommendation), a similar figure can be described for the case of legacy devices.

DA and the SL function are included in the device, the GA and the SL functions are included in the gateway, and the NA and the SL functions are included in the Network application server.

D-SL, G-SL, A-SL, SL-SL, as shown in Figure 3, are established as reference points.

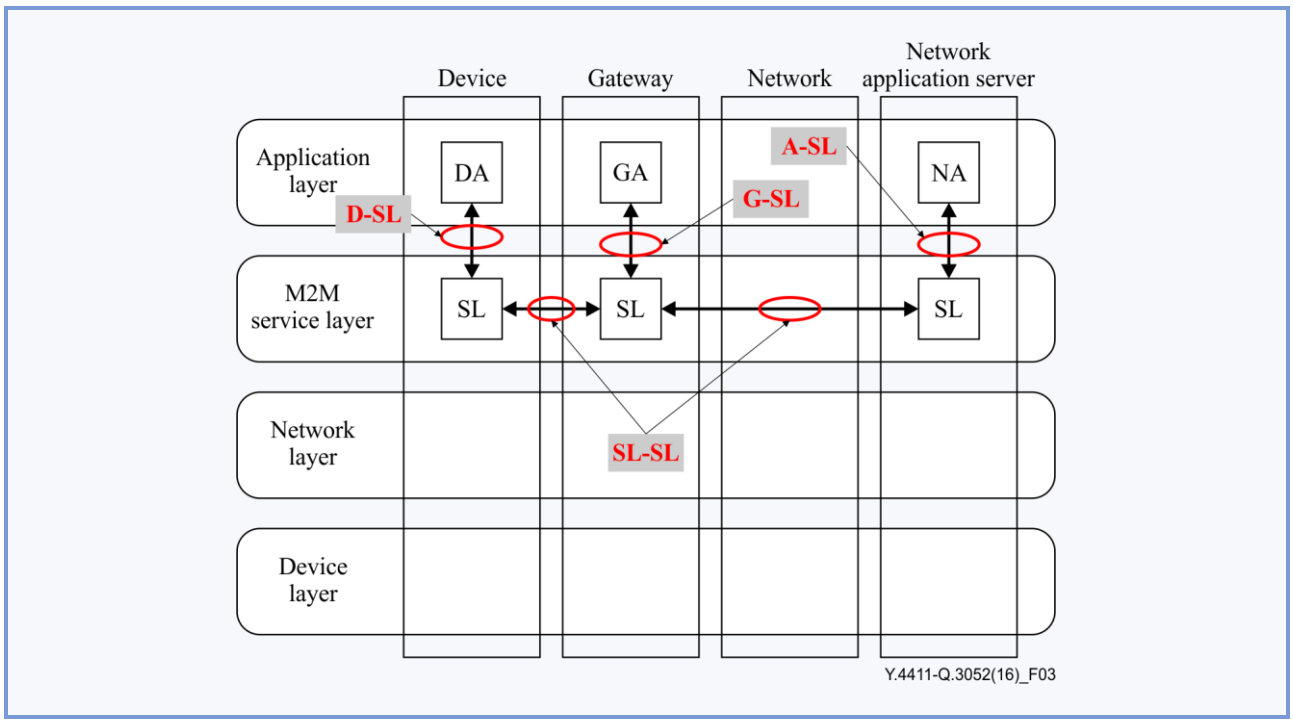


Figure 3 – Reference points in the component based M2M reference model (case with no legacy device)

It is necessary to clarify requirements of each reference point which can be used by both generic support capabilities and specific support capabilities, which reside in the M2M service layer, in order to identify protocols and APIs to be used across these reference points.

8 APIs and protocols for M2M

8.1 API overview

API stands for "Application Programming Interface". An API is a set of commands, routines, functions, tools and protocols that programmers can use when building software applications for a specific operating system. APIs allow programmers to use predefined functions to interact with the operating system, instead of writing the functions from scratch. APIs can be implemented by an application to allow other applications to interact more easily and/or effectively with it.

An API shields applications from the underlying resources, and reduces efforts involved in service development. Services are intended to be replicated and ported between different execution environments and hardware platforms. At the same time, services and technology platforms are allowed to evolve independently. A true value of an M2M enabled infrastructure comes from collecting, integrating and analysing the information from all devices in order to achieve specific business purposes. A quick and effective way to accomplish this practical goal is to connect this M2M enabled infrastructure with the core systems and business processes of the infrastructure. These assets can be made accessible to the M2M enabled infrastructure via APIs. Standardized APIs aim to ensure service interoperability and allow ubiquitous end-to-end service provisioning. Standardized APIs can provide efficiency in scale, service production and service development.

The M2M movement represents a significant shift for a number of industries. Connecting and digitizing data from disparate devices is usually disruptive to these industries. However, the foundational elements for the integration of M2M data and application services can be linked. Many companies have already begun to expose APIs that can be used in the context of M2M enabled applications.

NOTE – These APIs may need to be tuned further in order to minimize the data payloads, adapt the data formats to fit the peculiarities of the connected devices, or include security policies that fit the profile of the data being exchanged. It makes sense then for the communication link between the M2M enabled infrastructure and the enterprise assets to be based on standardized APIs.

There are a number of distinct advantages to this approach:

- APIs allow for real-time integration of the M2M enabled infrastructure with the e-health information related storage area, removing costs associated with erroneous stored data and respecting regulatory boundaries of data storage.
- APIs provide a consistent approach for integrating the enterprise's services, as well as those from the M2M enabled infrastructure providers, making skills and tools readily available.
- APIs are web-based, and they also enable the performance, scalability and security needed for high scale M2M deployments.

At a high level, APIs are ideal for the integration of an M2M enabled infrastructure and indeed many readily available APIs can be used for this purpose. However, APIs have generally evolved in the context of real-time human interactions. There are some characteristics of M2M enabled interactions that differ and must be considered when architecting M2M enabled applications:

- **access control and security** – Much of the security and access control that is implemented for APIs assumes a human end-user with specific permissions. A device-oriented security model is required to ensure appropriate control of the data flow. Different solutions may also be required depending on the access control requirements (e.g., API key model versus OAuth [IETF RFC 6749]).
- **synchronicity** – Many real-time APIs are synchronous. Many devices in an M2M enabled infrastructure require asynchronous communications for technical and business reasons. Existing APIs may need changes to handle these requirements.
- **scale and bandwidth** – M2M enabled communications demand simultaneously high scalability to handle the proliferation of devices, while being constrained on bandwidth based on geographical deployment in locations atypical for IT. This requires flexibility in service level agreements (SLAs) and optimization of APIs.

8.2 Design approach for M2M service layer APIs

There are mainly two design approaches for M2M service layer APIs.

Service-oriented architecture (SOA)

Service-oriented architecture (SOA) is a software design and software architecture design pattern based on discrete pieces of software providing application functionality as services to other applications.

A service is a self-contained unit of functionality. Services can be combined by other software applications to provide the complete functionality of a large software application. SOA makes it easy for computers connected over a network to cooperate. Every computer can run an arbitrary number of services, and each service is built in a way that ensures that the service can exchange information with any other service in the network without human interaction and without the need to make changes to the underlying program itself.

Resource-oriented architecture (ROA)

Resource-oriented architecture (ROA) is a style of software architecture and programming paradigm for designing and developing software in the form of resources with "RESTful" interfaces. These resources are software components (discrete pieces of code and/or data structures) which can be reused for different purposes.

Representational state transfer (REST) is an approach for getting information content from a website by reading a designated web page that contains an extensible markup language (XML) [XML 1.1] file that describes and includes the desired content. The REST approach is basically based on web technologies such as transfer protocols like hypertext transfer protocol (HTTP) [IETF RFC 2616], identification formats such as universal resource locator (URI), representation formats such as XML or (hyper text markup language (HTML) and content identifiers such as multipurpose iInternet mail extensions (MIME) [IETF RFC 2045] types. REST is neither a product nor a tool: it describes how a distributed software system can be architected. The design of a distributed system receives the stamp of approval of experts in REST, if the design meets a number of constraints: then the system design is called RESTful. REST is consistent with an information publishing approach that a number of web log sites use to describe some aspects of their site content, called RSS (RDF site summary). RSS uses resource description framework (RDF), a standard way to describe a website or other Internet resources.

8.3 Existing APIs and protocols for M2M service layer

A protocol is a uniform set of rules that enable two devices to connect and transmit data to one another. Protocols determine how data are transmitted between computing devices and over networks. Key capabilities of a protocol include type of error checking to be used, data compression method (if any), how the sending device will indicate that it has finished a message, and how the receiving device will indicate that it has received the message.

The following is a non-exhaustive list of existing APIs and protocols that can be considered for M2M service layer:

NAT-PMP

NAT port mapping protocol (NAT-PMP) [IETF RFC 6886] is a protocol for automating the process of creating network address translation (NAT) port mappings. NAT-PMP allows a computer in a private network (behind a NAT router) to automatically configure the router to allow parties outside the private network to contact it. NAT-PMP runs over user datagram protocol (UDP). It essentially automates the process of port forwarding. Included in the protocol is a method for retrieving the public IP address of a NAT gateway, thus allowing a client to make this public IP address and port number known to peers that may wish to communicate with it.

DPWS

Devices profile for web services (DPWS) [DPWS] defines a minimal set of implementation constraints to enable secure web service messaging, discovery, description and eventing on resource-constrained devices. DPWS is aligned with web services technology and includes numerous extension points allowing for seamless integration of device-provided services in enterprise-wide application scenarios. DPWS builds on the following core Web Services standards: Web services description language (WSDL) 1.1, XML Schema, SOAP 1.2, WS-Addressing, and further comprises WS-MetadataExchange, WS-transfer, WS-policy, WS-security, WS-discovery and WS-eventing.

SOAP

Simple object access protocol (SOAP) [SOAP] is a lightweight protocol intended for exchanging structured information in a decentralized, distributed environment. It uses XML technologies to define an extensible messaging framework providing a message construct that can be exchanged over a variety of underlying protocols. The framework has been designed to be independent of any particular programming model and other implementation specific semantics.

OAuth

The OAuth 2.0 authorization framework [IETF RFC 6749] enables a third-party application to obtain limited access to an HTTP service, either on behalf of a resource owner by orchestrating an approval interaction between the resource owner and the HTTP service, or by allowing the third-party application to obtain access on its own behalf.

WebSocket

The WebSocket protocol [IETF RFC 6455] enables two-way communication between clients running untrusted code in a controlled environment to a remote host that has opted-in to communications from that code. The security model used for this is the origin-based security model commonly used by web browsers. The protocol consists of an opening handshake followed by basic message framing, layered over transmission control protocol (TCP). The goal of this technology is to provide a mechanism for browser-based applications that need two-way communication with servers that does not rely on opening multiple HTTP connections (e.g., using XMLHttpRequest or <iframe>s and long polling).

8.4 M2M protocol structure and stacks

In order to introduce the M2M service layer related protocol layering and stacks, an example of protocol stacks in the component-based M2M reference model is shown in Figure 4. The components of the component-based M2M reference model (device, gateway, M2M platform and application server) are all involved in the application related operations.

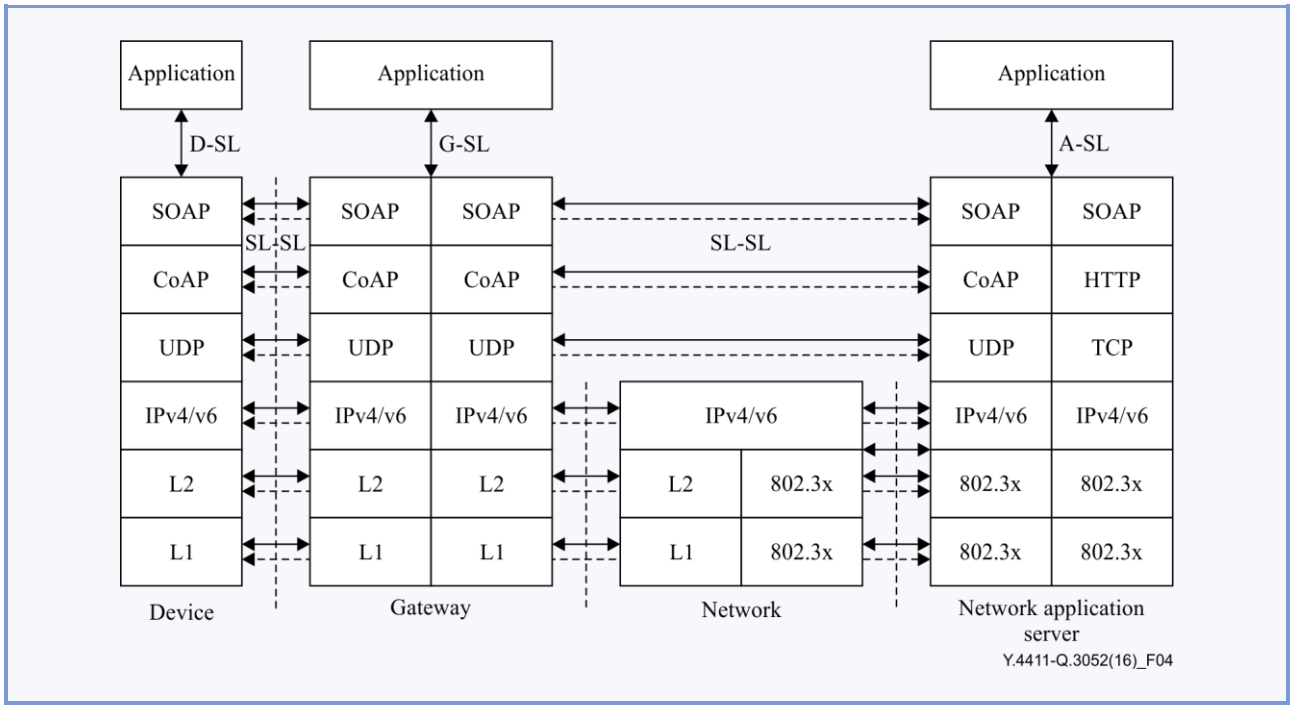


Figure 4 – Example of protocol stacks in the component-based M2M reference model

There are two cases of connection types between devices and M2M platform. The first case is when a device communicates with M2M platform via a gateway(s), shown in Figure 4 with continuous lines, and the second case is when a device communicates with M2M platform directly without a gateway(s), shown in Figure 4 with dashed lines.

Figure 5 shows a concrete application instance that utilizes the component-based M2M reference model for an e-health application. This concrete instance provides a specific example of M2M protocol stacks where, for example, a scale or a sphygmomanometer is used as device connected to a gateway, and a PHR server is used as application server. This example shows that IEEE11073/IEEE20601 [IEEE 11073-20601] is used for the SL-SL reference point between device and gateway, HL7v2/SOAP/HTTP [SOAP] [IETF RFC 2616] is used for the G-SL, A-SL and gateway-platform SL-SL reference points, and HL7v3/SOAP/HTTP [SOAP] [IETF RFC 2616] is used for the A-SL reference point related to NA.

NOTE 1 – These protocol stacks are just one example; the appropriate protocols are required according to the devices and applications in use.

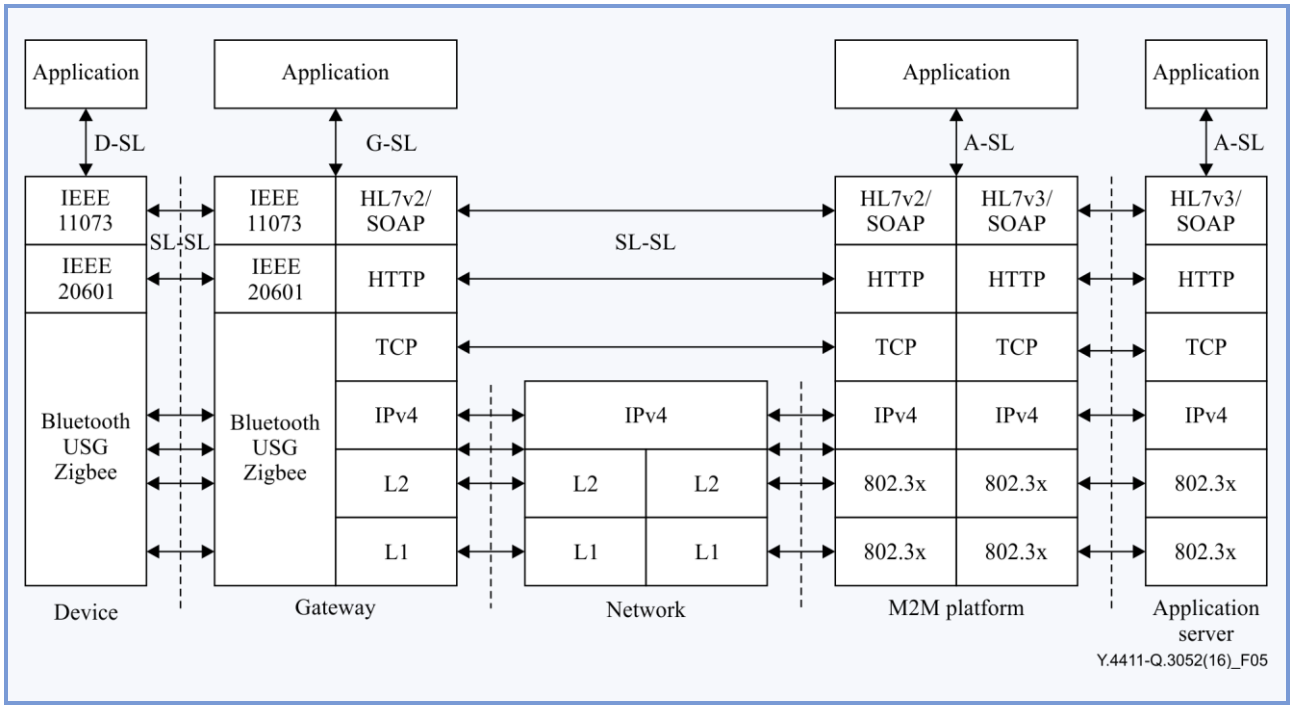


Figure 5 – Example of M2M protocol stacks for e-health application (using gateway)

Another concrete application instance in Figure 6 shows a specific example of M2M protocol stacks where, for example, a smart phone is used as device connected with the network without a gateway and a PHR server is used as application server. This example shows the usage of SOAP/CoAP [SOAP] for the D-SL and device-M2M platform SL-SL reference points, and SOAP/HTTP [SOAP] [IETF RFC 2616] for the A-SL reference point related to NA.

NOTE 2 – These protocol stacks are just one example; the appropriate protocols are required according to the devices and applications in use.

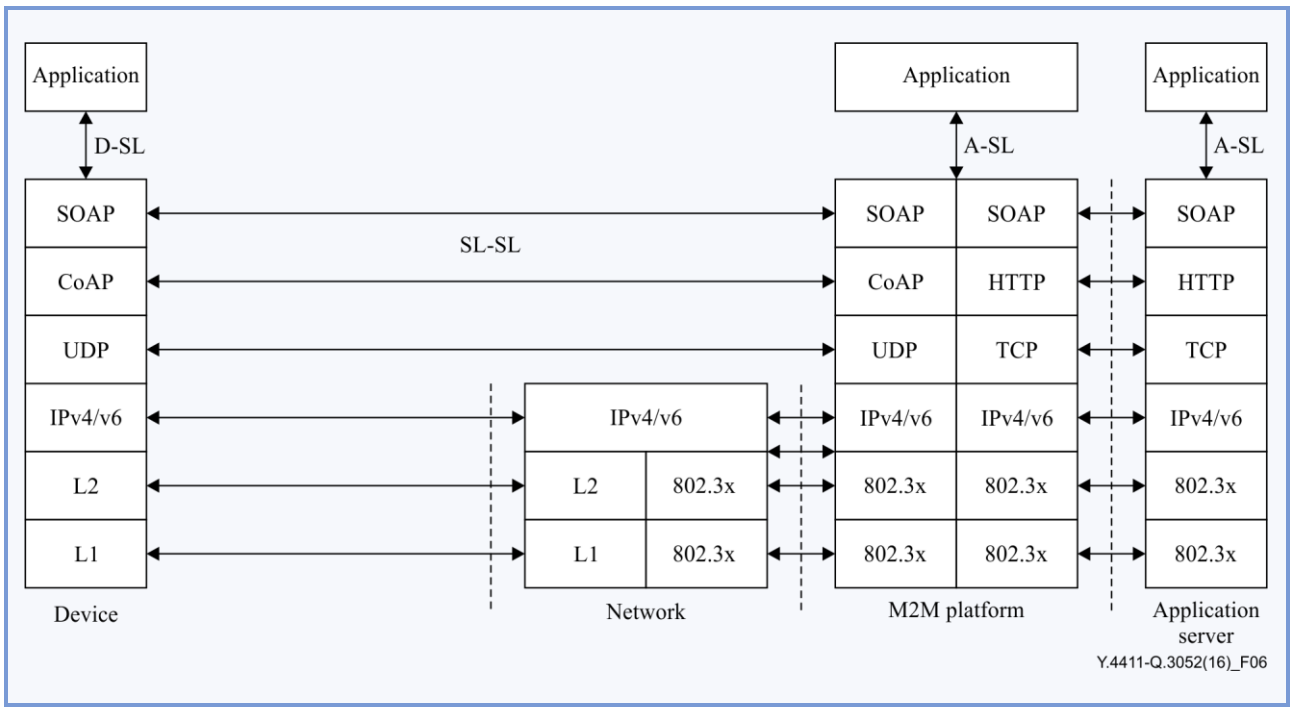


Figure 6 – Example of M2M protocol stacks for e-health application (without gateway)

NOTE 3 – The present document considers specific cases for e-health application scenarios, but it will be necessary to consider other application scenarios (implying different types of devices, networks and security levels, etc.) in the future.

9 General requirements of APIs and protocols with respect to the M2M service layer

9.1 Extensibility

M2M protocols are designed to allow continued development and to facilitate changes by means of standardized extensions.

The impact of extensibility on the existing M2M protocol functions shall be minimized.

Extensibility can be related to one or more of the following aspects:

- Handling a wide range of transport protocols as well as a different number of devices.
- Adding, removing or modifying protocol functionality.
- New protocol routines.

9.2 Scalability

For provisioning scalability as a requirement in the design of M2M protocols, one or several of the following mechanisms are used:

- Ensuring direct addressability to the M2M platform hosting target resources, to minimize network hops.
- Asynchrony in terms of data processing, with the objective of minimizing the number of discarded packets.
- Caching mechanisms that allow all the received packets to be processed.
- Efficient load distribution to avoid bottlenecks and data loss.
- Data compression and/or aggregation, in order to reduce the amount of data sent through the network.

9.3 Fault tolerance and robustness

One or more of the following mechanisms in terms of link availability can be exploited in the design of M2M protocols to account for a variety of exception conditions:

- To provide reliable transmission of data packets, packet recovery will be dealt with by using mechanisms appropriate for the operating environment.
- When M2M protocols are employed over unreliable links, multiple data dissemination paths can be provided and maintained.

9.4 Efficiency

M2M protocols are designed with consideration of efficiency for networking involved resource-constrained devices.

- As energy consumption directly affects the overall system performance, M2M protocols should consider energy efficiency, especially in resource constrained environments with battery-powered M2M devices.
- Energy efficient M2M protocols aims at reducing the overall energy consumption while maintaining the performance required by the M2M Applications.

9.5 Interoperability

API interoperability between different protocol stacks is expected. M2M API over HTTP/TCP/IP needs to interoperate with CoAP/UDP in a local network using M2M API. M2M protocols are specified to provision the API interoperability.

9.6 Self-operation and self-management

Devices employing the M2M API interwork with established management protocols (e.g., security, discovery, bootstrapping, etc.). The interworking with legacy management protocols via the M2M API shall be carried out in self-operation methods.

Appendix I

Examples of attributes for APIs and protocols

(This appendix does not form an integral part of this Recommendation.)

Clause 8.3 lists existing APIs and protocols related to the M2M service layer. Because of the large number of potential APIs and protocols, their classification and analysis is useful information for developers when selecting suitable protocols for M2M service layer. The following provides some examples of attributes to be considered for APIs and protocols with respect to the various interfaces:

- Interface for device – gateway, device – network application server and device – device (e.g.) protocol load (information volume, connectionless/connection-oriented), routing capability, IP based/non IP based
- M2M platform interface (e.g.) communication security, scalability, real time capability, multitask capability, stateful/stateless
- Application server interface (e.g.) Internet compatibility (with other Internet services)
- Attributes common to all above indicated interfaces (e.g.) expandability, usability, openness (including open source projects).

Appendix II

Reference of other existing APIs and protocols for M2M service layer

(This appendix does not form an integral part of this Recommendation.)

CoAP

Constrained application protocol (CoAP) [b-IETF CoAP] is a specialized web transfer protocol for use with constrained networks and nodes for machine-to-machine applications such as smart energy and building automation. CoAP provides a method/response interaction model between application end-points, supports built-in resource discovery, and includes key web concepts such as URIs and content-types. CoAP easily translates to HTTP for integration with the Web while meeting specialized requirements such as multicast support, low overhead and simplicity for constrained environments.

Modbus

Modbus [b-Modbus] is an application layer messaging protocol, positioned at level 7 of the OSI model, which provides client/server communication between devices connected on different types of buses or networks. Modbus is a serial communication protocol extensively used in supervisory control and data acquisition (SCADA) systems to establish a communication between remote terminal unit (RTU) and devices.

UPnP

Universal plug and play (UPnP) technology defines an architecture for pervasive peer-to-peer network connectivity of intelligent appliances, wireless devices and PCs of all form factors. It is designed to bring easy-to-use, flexible, standards-based connectivity to ad-hoc or unmanaged networks whether in the home, in a small business, public spaces or attached to the Internet. UPnP technology provides a distributed, open networking architecture that leverages TCP/IP and web technologies to enable seamless proximity networking in addition to control and data transfer among networked devices.

IGD

Internet gateway device (IGD) [b-IGD] standardized device control protocol is an "edge" interconnect device between a residential local area network (LAN) and the wide area network (WAN) providing connectivity to the Internet. It is supported by some NAT routers. It is a common method of automatically configuring port forwarding.

BiTXML

The BiTXml [b-BiTXml] communication protocol has been designed to implement a presentation level of the (OSI-based) communication stack reference, with the main goal of standardizing the way commands and control information are exchanged for the specific target of M2M communication demands (i.e., communication with generic devices with or without processing power on board – like sensors, actuators, as well as air conditioning systems, lifts, etc. or a combination of them).

Bibliography

[b-ITU-T FG M2M D2.1] ITU-T FG M2M Deliverable D2.1, *M2M service layer: Requirements and architectural framework*. https://www.itu.int/dms_pub/itu-t/opb/fg/T-FG-M2M-2014-D2.1-PDF-E.pdf

[b-BiTXml] BiTXml (12007), *M2M communications Protocol*.

[b-ETSI TS 118 104] ETSI TS 118 104 V1.0.0 (2015-02), *Service Layer Core Protocol Specification*.

[b-HL7v2] HL7 Version 2, *HL 7 Standard version 2*.

[b-HL7v3] HL7 Version 3, *HL 7 Standard version 3*.

[b-IETF CoAP] IETF draft-ietf-core-coap (2013), *Constrained Application Protocol (CoAP)*.

[b-IGD] UPnP Forum –IGD (2010), *Internet Gateway Device*.

[b-Modbus] Modbus (2012), *Application protocol specification*.


CARD

Y.4412/F.747.8

Requirements and reference architecture for audience-selectable media service framework in the IoT environment

Requirements and reference architecture for audience-selectable media service framework in the IoT environment

Summary

Recommendation ITU-T Y.4412/F.747.8 describes requirements and reference architecture for the audience-selectable media (ASM) service framework. The ASM service provides audiences with media selection options according to their interests and preferences. Examples of media selection options include the selection of an object of interest, and the selection of an interested camera object among multi-camera objects providing different views.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T Y.4412/F.747.8	2015-11-29	16	11.1002/1000/12620

Keywords

ASM, audience-selectable, Internet of things, media.

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

Table of Contents

Introduction

Recently, demands for personalized media and interactivity in the media service are increasing and media service platforms are needed to support the fast development and deployment of new multimedia services/applications by reconfiguration of the existing service components. Due to the expansion of the Internet of things (IoT), objects related to media services can be identified and recognized by people or other objects. Also, the IoT enables person-to-object and object-to-object communications as well as person-to-person communication.

Audience-selectable media (ASM) service enables viewers to be interactive and select preferred media in the running time by providing media selection options.

Recommendation ITU-T Y.4412/F.747.8

Requirements and reference architecture for audience-selectable media service framework in the IoT environment

1 Scope

This Recommendation defines requirements and reference architecture for audience-selectable media (ASM) service in the Internet of things (IoT). The scope of this Recommendation includes:

– concept of ASM service framework;

– requirements of ASM service framework;

– reference architecture of ASM service framework; and

– functional entities of ASM service framework.

2 References

None.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 device [b-ITU-T Y.4400]: An apparatus through which a user can perceive and interact with the web.

3.1.2 Internet of things [b-ITU-T Y.4000]: A global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies.

NOTE 1 – Through the exploitation of identification, data capture, processing and communication capabilities, the IoT makes full use of things to offer services to all kinds of applications, whilst ensuring that security and privacy requirements are fulfilled.

NOTE 2 – From a broader perspective, the IoT can be perceived as a vision with technological and societal implications.

3.1.3 object [b-ITU-T Y.2002]: An intrinsic representation of an entity that is described at an appropriate level of abstraction in terms of its attributes and functions.

3.1.4 resources [b-ITU-T Y.4400]: The term "resource" is whatever might be identified by a URI.

NOTE – Familiar examples include an electronic document, an image, a source of information with a consistent purpose (e.g., "today's weather report for Los Angeles"), a service (e.g., an HTTP-to-SMS gateway), and a collection of other resources. A resource is not necessarily accessible via the Internet; e.g., human beings, corporations, and bound books in a library can also be resources. Likewise, abstract concepts can be resources, such as the operators and operands of a mathematical equation, the types of a relationship (e.g., "parent" or "employee"), or numeric values (e.g., zero, one, and infinity).

3.1.5 thing [b-ITU-T Y.4000]: With regard to the Internet of things, this is an object of the physical world (physical things) or the information world (virtual things), which is capable of being identified and integrated into communication networks.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 camera control server: A server that selects and updates camera objects, camera object groups, and the main camera object. It also controls camera objects.

3.2.2 camera object: A device that captures and tracks objects of interest. It can also transmit and receive the tracked position of each object of interest to both other camera devices and a camera control server through the Internet. Its pan-tilt-zoom may be controlled by the camera control server for tracking objects of interest.

3.2.3 camera object group: A group of camera objects that capture the same object of interest whose occupied region size in the frame is greater than the pre-defined value.

3.2.4 main camera object: A camera object that captures a main broadcasting video. The initial main camera object is selected by audiences or the broadcaster. The main camera object can be changed to the sub-camera object during the service.

3.2.5 object of interest: A moving person or a moving thing in the video of which the service audience may have interest.

3.2.6 sub-camera object: A camera object that is not selected as a main camera object. An initial sub-camera object is usually selected by the broadcaster. The sub-camera object can be changed to the main camera object by audiences during the service.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

ASM	Audience-Selectable Media
FE	Functional Entity
ID	Identifier
IoT	Internet of Things
URI	Uniform Resource Identifier
Wi-Fi	Wireless Fidelity

5 Conventions

None.

6 Overview of audience-selectable media (ASM) service

Internet of things (IoT) and web of things (WoT) enable physical devices to be accessed as resources and services/applications of the web.

As new multimedia services/applications are emerging, media service platforms need to support fast development and deployment of new multimedia services/applications by reconfiguration of the existing service components. Also, media service devices, such as cameras, are beginning to provide wireless fidelity (Wi-Fi) technology.

IoT-applied media service enables objects in media service to be identified and recognized by people or other objects; IoT also enables person-to-object and object-to-object communications as well as person-to-person. This realizes the smart media service where objects can be automatically reconfigured to create and deliver media content according to the audience's demands and preferences.

In the current broadcasting service, a suitable camera object is selected among available camera objects, video from the selected camera object is encoded, and the encoded video is broadcasted to viewers. Attributes of the content and service region are considered for broadcasting.

This traditional broadcasting service does not provide any options to audiences to select various views from different camera objects in the same scene. The lack of this option causes audiences to remain passive.

The ASM service enables audiences to be interactive and select preferred media in the running time by providing media selection options. Examples of media selection options include the selection of an object of interest and a selection of an interested camera object among multi-camera objects providing different views. Figure 1 illustrates the concept of the ASM service.

As described in Figure 1, there are three components in the ASM service: Broadcaster, audience, and camera management server. These components generate and update object of interest, camera group for each object of interest, and a camera view-video that the audience is interested in, including the following:

(1) **broadcaster**: The broadcaster shows object of interest candidates to the audience.

(2) **audience**: The audience selects one object from the object of interest candidates, selects one of the camera objects in the camera group for the selected object of interest, and watches the selected camera view-video.

(3) **camera management server**: The camera management server updates the cameras in each camera group for each object of interest, gets the selection option inputs from the audience, and broadcasts the selected camera-view video to the audience.

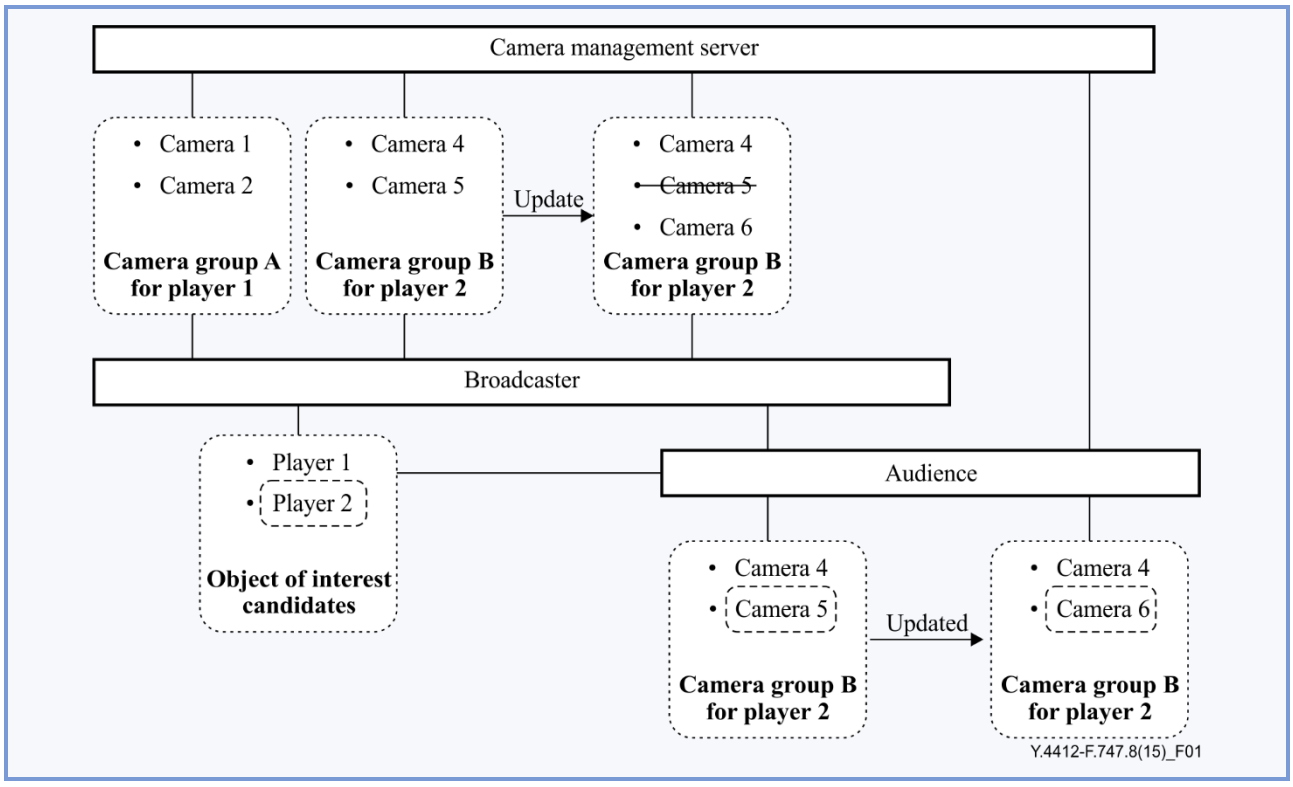


Figure 1 – Concept of ASM service

The concept of the ASM service described in Figure 1 can be explained as follows:

– broadcasters first specify the object of interest candidates in the scene (e.g., player 1 and player 2) and then show them to the audience for selection;

– the camera management server constitutes an initial corresponding camera group for each object of interest candidate using all available cameras based on the positions of each object of interest, and each camera;

– an audience first selects one object from the object of interest candidates (e.g., player 2), and also selects one of the cameras in the selected camera group (e.g., camera 5 in camera group B);

– the camera management server receives the audience's selections for the camera and object of interest, and broadcasts a selected camera-view video (e.g., camera 5 in camera group B);

– as an object of interest (e.g., player 2) moves in the scene during the running time, the camera management server updates each member of each camera group based on the updated position of each object of interest (e.g., In Figure 1, camera 5 in the initial camera group B is deleted from the updated camera group B);

– with updated camera members of each group, the camera management server shows each updated selectable camera list for each object of interest;

– then, an audience may change which camera view to watch (e.g., from camera 5 in initial camera group B to camera 6 in updated camera group B).

7 Requirements of ASM service framework

7.1 Service requirements

7.1.1 Media selection option

The ASM service is required to provide audiences with:

– media selection option to select media according to their interests and preferences;

– video selection option to select one camera object among multi-camera objects;

– switching option to change the current camera object to another camera object among multi-camera objects.

7.1.2 Object of interest selection option

The ASM service is required to:

– provide audiences with a selection option of object of interest to select an object they wish to watch;

– support audiences with changing option of object of interest to in the running time;

– support identification of each object of interest with an appropriate identifier (ID).

7.1.3 Main camera object selection option

– the ASM service is required to provide audiences with a main camera object selection option among multi-camera objects.

7.1.4 Camera object group selection option

– the ASM service is required to provide audiences with the choice of a camera object group selection that captures each object of interest among multi-camera object groups according to their preferences.

7.2 Functional requirements

7.2.1 Functional requirements of a camera object

Each camera object is required to:

– be able to obtain the position or the tracking result of an audience-selected object of interest in each captured frame;

– receive the positions or the tracking results of an audience-selected object of interest in each frame of the main camera object as a sub-camera object;

– convert the tracked position of an audience-selected object of interest in the main camera object to the position of each camera object;

– receive audience's video selection information from the video playback terminal through the camera control server;

– share an ID of each tracked object of interest with other camera objects within the same camera object group.

Each camera object selected as a main camera object is required to:

– transmit the positions or the tracking result of an audience-selected object of interest in each frame to the other sub-camera objects and the camera control server.

7.2.2 Functional requirements of camera control server

A camera control server is required to:

– select camera objects that should be shown to the audience for selection;

– update members of each camera object group based on the tracking results of the audience-selected object of interest within the appropriate time interval;

– update the main camera object candidates in each camera object group based on the tracking results of the audience-selected object of interest within the appropriate time interval;

– recommend the main camera object candidates in each camera object group to the audiences based on the tracking results of the object of interest;

– notify audiences of updates of members in each camera object group;

– receive audience's selection information such as an object of interest selection, camera object group selection, and main camera object selection;

– notify audience's selection information to each audience-selected camera object for video transmission to the audience's video playback terminal;

– receive the positions or the tracking results of the audience-selected object of interest in each frame of all camera objects within the appropriate time interval;

– manage each camera object member within each camera object group according to the movement of each object of interest.

A camera control server is recommended to:

– control the pan-tilt-zoom of each sub-camera object according to the tracked position and the size of the object of interest in the frame of the main camera object.

8 Reference architecture of ASM service framework

8.1 Reference architecture

Figure 2 shows a reference architecture of the ASM service framework which consists of ASM service framework functions, functional entities (FEs) of each ASM service framework function, reference points between each ASM service framework function.

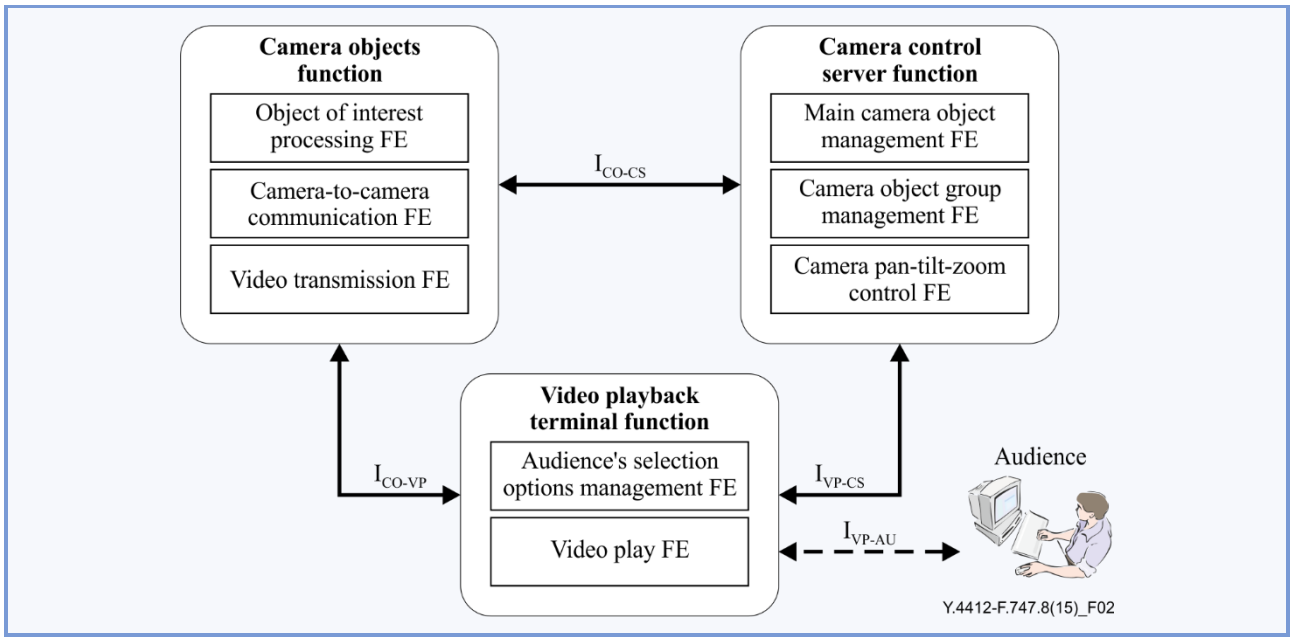


Figure 2 – Reference architecture of ASM service framework

As shown in Figure 2, there are three ASM service framework functions: camera objects function, camera control server function and video playback terminal function. Each function is described as follows in terms of supported functions, supported information, and supported FEs:

– **camera objects function**: A camera objects function tracks an object of interest, communicates the updated status of the object of interest to the camera control server function, and transmits the audience-selected video to the video playback terminal function. Based on these functional descriptions, the camera objects function includes three FEs of object of interest processing, communication, and video transmission;

– **camera control server function**: The camera control server function manages the main camera object based on the audience's selection, updates the camera object group based on the tracking result of the object of interest, and controls each camera object using the calculated pan, tilt, and, zoom values. Camera control server function contains three FEs of main camera object management, camera object group management, and camera pan-tilt-zoom control;

– **video playback terminal function**: Video playback terminal function receives the audience's selection options, transmits the audience's selection information to the camera objects function and camera control sever function, and plays the audience-selected video. The video playback terminal function consists of two FEs of the audience's selection options management and video play.

As illustrated in Figure 2, there are four reference points: I_{CO-CS}, I_{CO-VP}, I_{VP-CS}, and I_{VP-AU} as follows:

– **I_{CO-CS}**: I_{CO-CS} is a reference point between the camera objects function and the camera control server function. The position of the object of interest is transferred from the camera objects function to the camera control server function. On the other hand, updated information, including object of interest selection, main camera object selection, camera object group selection, and camera control (e.g., pan, tilt, and zoom values of each camera object) information is transferred from the camera control server function to camera objects function;

– **I_{CO-VP}**: I_{CO-VP} is a reference point between the camera object function and video playback terminal function. The camera objects function sends the audience-selected video to the video playback terminal function. Also, the playback terminal function sends the audience's selection information and playing options to the camera objects function;

– **I_{VP-CS}**: I_{VP-CS} is a reference point between the video playback terminal function and camera control server function. Through this reference point, video playback terminal transfers the audience's selection information and playing options to the camera control server function. Also, the video playback terminal function transmits the updated information, including the main camera object selection information (for each camera object group), camera object group selection information (for each object of interest), and camera control information for each camera object to the camera control server function;

– **I_{VP-AU}**: I_{VP-AU} is a reference point between the video playback terminal function and the audience. The video playback terminal function shows a list of the audience's selection options available to the audience. The video playback terminal function then receives the audience's selection information and playing options from the audience. Through this reference point, the video playback terminal function shows the current status of each camera object group, current camera control status of each camera object, currently-selected main camera object, position of each object of interest, multi-camera video thumbnails, and audience-selected video with selected playing options during the service.

8.2 Camera objects function

The camera objects function:

– consists of object of interest processing, communication, and video transmission FEs.

8.2.1 Object of interest processing FE

The object of interest processing FE:

– obtains the ID and the position or the tracking result of the audience-selected object of interest in each captured frame through object of interest extraction and tracking;

– identifies an object of interest in order to provide positional information of each object of interest that is used for generating and updating its corresponding camera object group in camera control server function;

– converts the tracked position of the audience-selected object of interest in the main camera object to the position in each frame captured by its sub-camera object using the camera position and pose information.

8.2.2 Communication FE

The communications FE:

– transmits the ID and the positions or the tracking results of an audience-selected object of interest in each captured frame to the other sub-camera objects and the camera control server function when one of the camera object functions is selected as a main camera object;

– receives the positions or the tracking result of the audience-selected object of interest in each frame of the main camera object when one of the camera objects is selected as a sub-camera object;
– receives the audience's selection information such as an object of interest selection, a main camera object selection in a camera object group, a selection of an interested camera object among multi-camera objects, video playback terminal selection, and camera object group selection from the video playback terminal function;
– receives the camera control information such as pan-tilt-zoom values from camera pan-tilt-zoom control FE of the camera control server function.

8.2.3 Video transmission FE

The video transmission FE:

– transmits a video stream of the main camera object selected by audience to the video playback terminal function.

8.3 Camera control server function

The camera control server function:

– consists of main camera object management, camera object group management, and optional camera pan-tilt-zoom control FEs.

8.3.1 Main camera object management FE

The main camera object management FE:

– receives the audience's main camera object selection information from the video playback terminal function;
– updates the main camera object candidates in each camera object group for updating based on the tracking results of the audience-selected object of interest transmitted from the object of interest processing FE within appropriate intervals;
– recommends the main camera object candidates in each camera object group to the audiences based on the tracking results of the object of interest when the main camera object needs to be updated;
– notifies the audience's selection to the newly audience-selected main camera object in order to track the selected object of interest and transmit an audience-selected video when the audience's main camera object selection information is updated from the video playback terminal function.

8.3.2 Camera object group management FE

The camera object group management FE:

– initially generates camera object groups according to each identified object of interest;
– classifies each camera object into its corresponding camera object group according to the existence of identified objects of interest in camera objects;
– receives the position or the tracking result of the audience-selected object of interest in each frame from all the camera objects in the group within appropriate time interval;
– updates the members of each camera object group based on the tracking result of each audience-selected object of interest;
– notifies member updates in each camera object group to the audiences.

8.3.3 Camera pan-tilt-zoom control FE

The camera pan-tile-zoom control FE:

– calculates pan, tilt, and zoom values of each sub-camera object according to the position and the size of the object of interest in the frame of the main camera object in order to make camera objects within the same camera object group look at the same object of interest;

– transmits the calculated pan, tilt, and zoom values of sub-camera objects to each corresponding sub-camera object within the same camera object group in order to control the sub-camera objects, respectively.

8.4 Video playback terminal function

The video playback terminal function:

– consists of the audience's selection options management and video play FEs.

8.4.1 Audience's selection options management FE

The audience's selection options management FE:

– shows a list of available audience selection options to the audience;

– receives audience's selection information from the audience;

– transmits audience's selection information to the camera control server function;

– transmits audience's selection information to each camera object function;

– includes options of an object of interest selection, a main camera object selection, and a selection of interested camera objects among multi-camera objects providing different views;

– includes the audience's choices among available audience selection options.

8.4.2 Video play FE

The video play FE:

– receives an audience-selected video from the camera object function;

– shows a current status of each camera object group, a current camera control status of each camera object, a currently-selected main camera object, the position of each object of interest, multi-camera video thumbnails, and an audience-selected video with audience-selected playing options;

– plays an audience-selected video;

– controls a playing status of the current audience-selected video with audience-selected playing options.

Appendix I

Use cases for ASM service

(This appendix does not form an integral part of this Recommendation.)

I.1 Use case for an ASM broadcasting service using multi-camera objects

Figure I.1 shows a use case for an ASM broadcasting service using multi-camera objects. This use case provides audiences with a selection option of choosing a video media from one camera object amongst multi-camera objects that capture a person (an object of interest) whom an audience is interested in. It also provides a selection option of switching a video media from one camera object to another camera object.

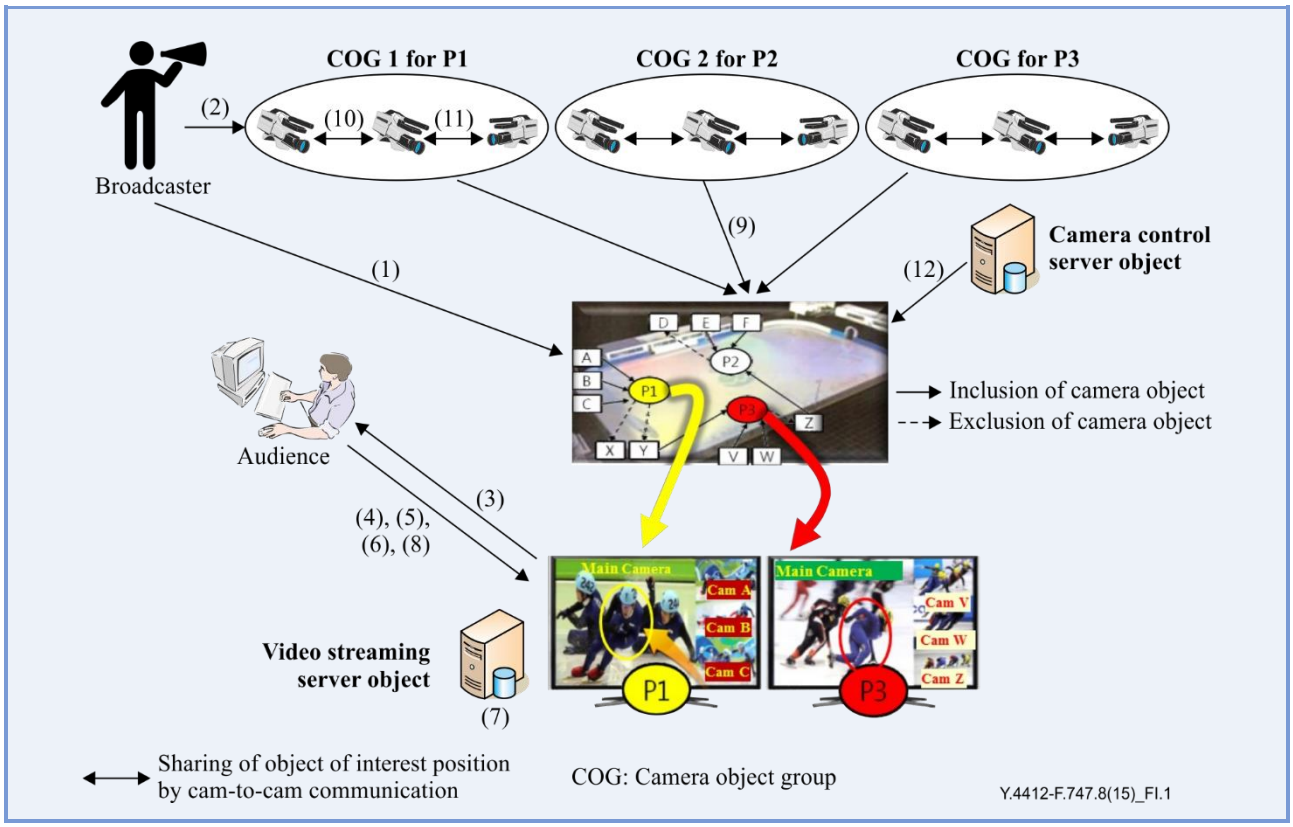


Figure I.1 – Use case for ASM broadcasting service using multi-camera objects

In this use case, there is the assumption that each audience has its own object of interest (e.g., a person such as a sports player or famous actress or actor) in the broadcasted video and each object of interest appears and disappears in some camera objects and frames.

(1) broadcasters predefine object of interest candidates (e.g., sports players) that will appear in the scene that they are going to broadcast;

(2) broadcasters generate camera object groups by combining suitable multi-camera objects that capture candidate objects of interest among available camera objects located at the site where broadcasters want to broadcast;

(3) broadcasters show videos or thumbnail images captured by each camera object group to the audiences;

(4) an audience selects one camera object group that captures an object of interest that he or she is interested in among multi-camera object groups;

(5) the audience also selects a main camera object among multi-camera objects in the selected camera object group;

(6) the audience also selects a region of interest that includes an object of interest in the main camera object image;

(7) video streaming server object streams and shows a video of the main camera object with sub-images (e.g., thumbnails) of other camera objects to the audience;

(8) during the streaming, the audiences may switch the main camera object to another camera object if he or she wants to watch the video with a different camera angle;

(9) after the object of interest selection, a video processing server object that is linked with the main camera object starts the video processing, such as automatic tracking of a selected object of interest in the main camera object frame by frame;

(10) during the tracking process, the main camera object transmits a tracked position of the selected object of interest to the other camera objects by camera-to-camera transmission;

(11) the other camera objects receive and calculate the position of the selected object of interest from the received tracked position in the image of the main camera object;

(12) based on each calculated position of an object of interest in each camera object, the camera object control server object updates the component camera objects in each camera object group;

(13) the process repeats step 4) to step 13) until the audiences or the broadcasters quit the current media service.

Finally, each audience can enjoy personalized video media content that includes an object of interest and has a camera view that each audience wants to watch.

I.2 Use case for an open screen service using multi-camera objects

Open screen service can be a use case of the ASM service. Open screen service is a service based on open platform to control various types of screens and to show content by sensing states (e.g., age and gender) and movement of a human being. Representative use cases of open screen service include digital signage and emergency service. For example, if a person is near a screen using open platform of open screen service, the screen can show an image or video media advertisement or an evacuation route to the person.

In order to sense states and movement of human beings, multi-camera objects and camera control servers are mandatory for the open screen service. In the perspective of open screen service, the audience can be an advertiser or emergency service provider, and the object of interest can be a human being who is a customer of an advertiser.

Figure I.2 explains use case for an open screen service using multi-camera objects:

(1) an object of interest (e.g., customer) moves to the vicinity of a screen (e.g., screen 1) equipped with a camera object (e.g., camera object 1).

(2) The camera object (e.g., camera object 1) captures video data of the object of interest and then sends the captured video data to camera control server.

(3) The camera control server sends information for tracking the object of interest to the camera object (e.g., camera object 1).

(4) The camera control server sends the position information and captured video data to media server being owned by the audience (e.g., advertiser) and providing image or video media (e.g., advertisement) that the audience (e.g., advertiser) wants to provide for the object of interest (e.g., customer).

(5) The media server determines images or video media (e.g., advertisements) that the audience wants to provide for the object of interest (e.g., customer), and then provides the media to the screen (e.g., screen 1). Interaction between camera objects can include information for supporting the camera's tracking the object of interest. The information for tracking the object of interest can include position information for where the object of interest is located.

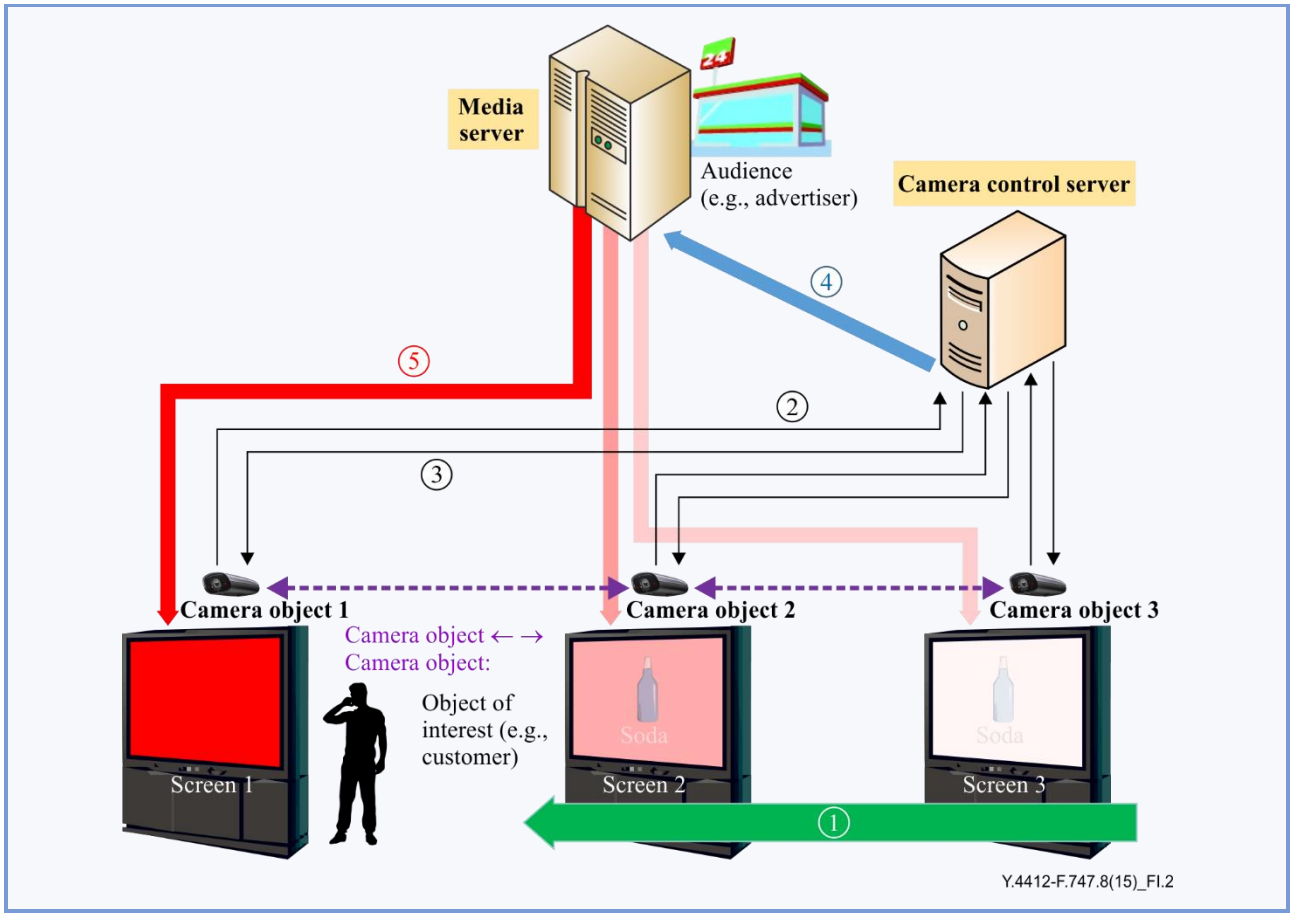


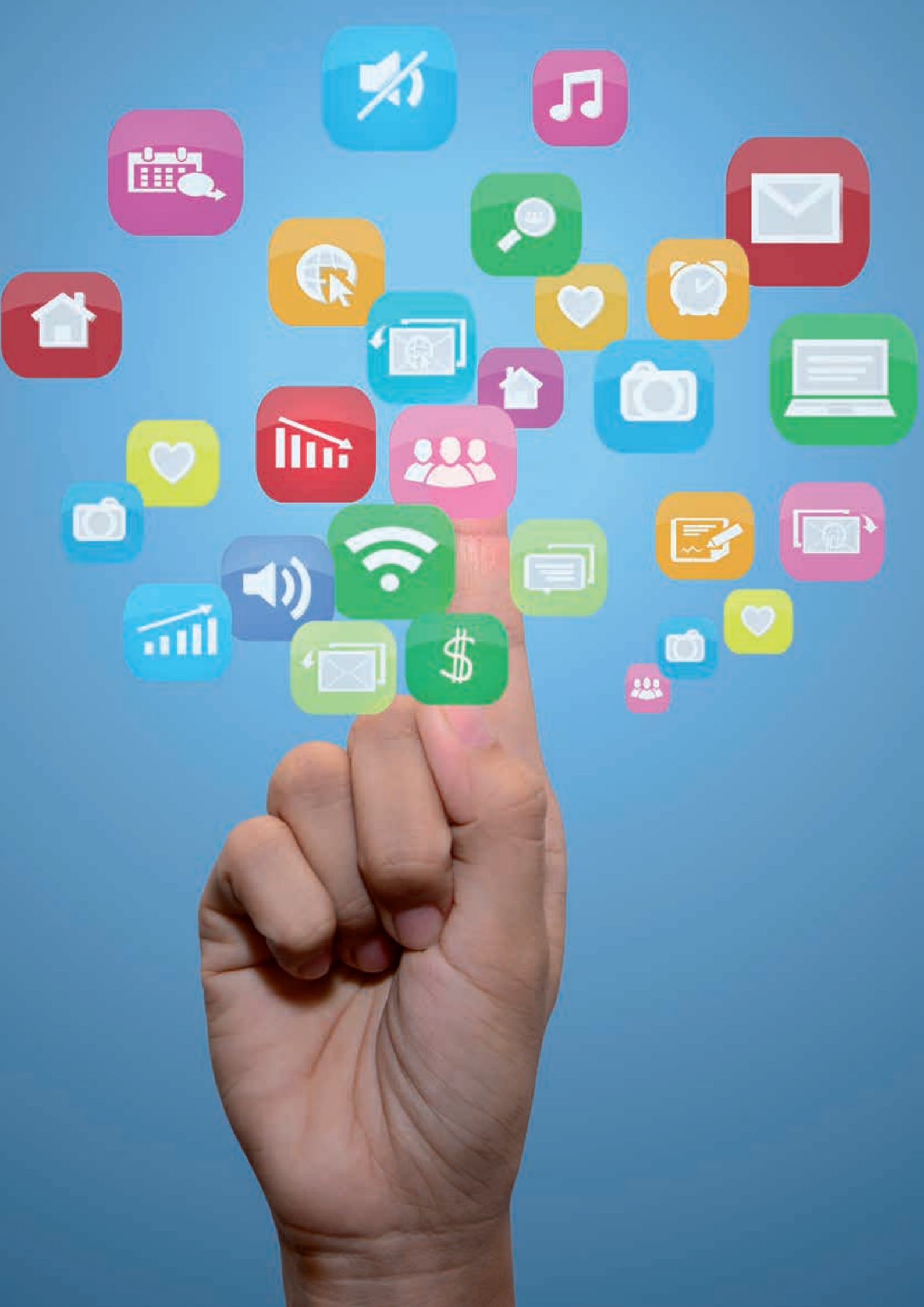
Figure I.2 – Use case for an open screen service using multi-camera objects, camera control server and media server

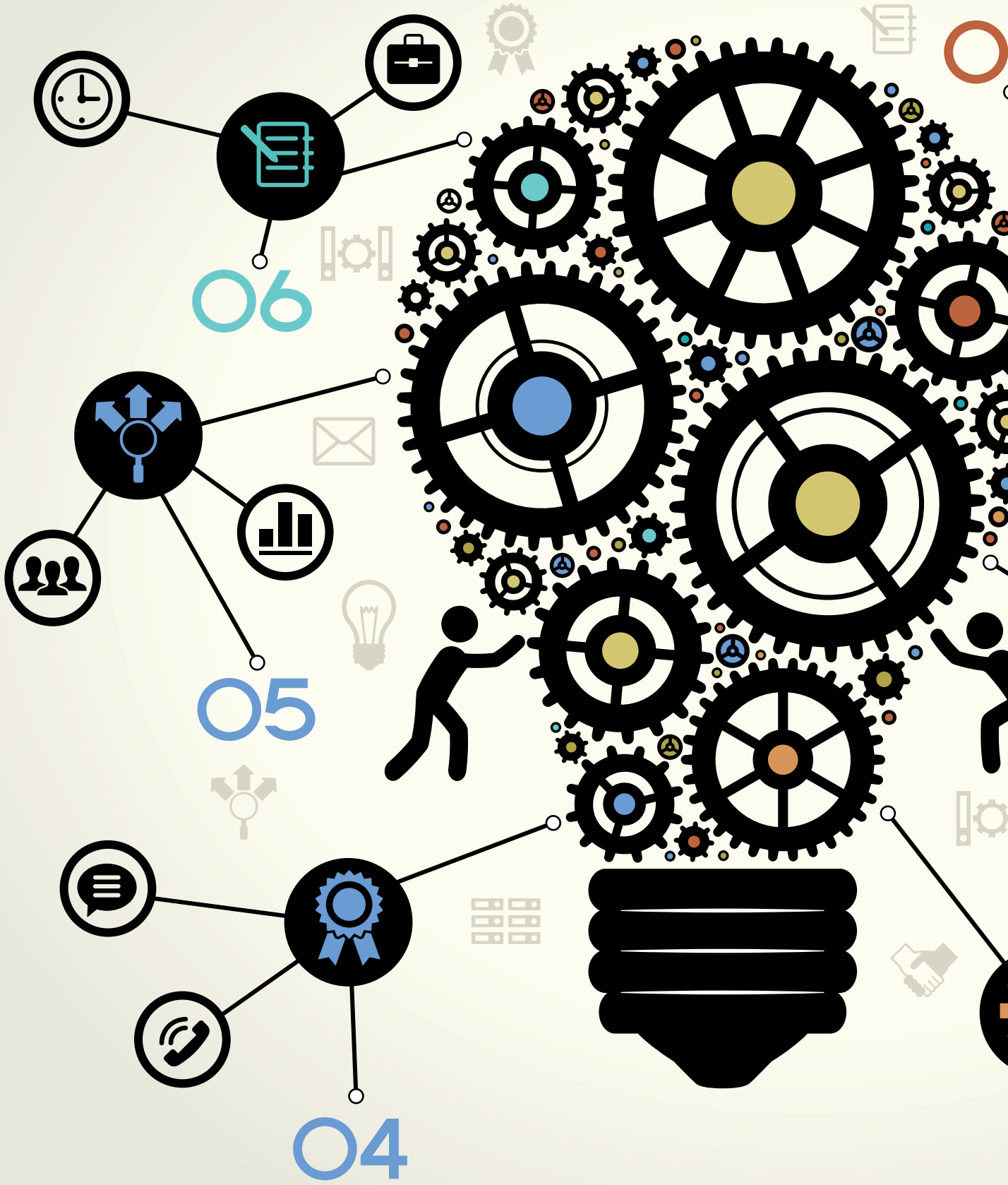
Bibliography

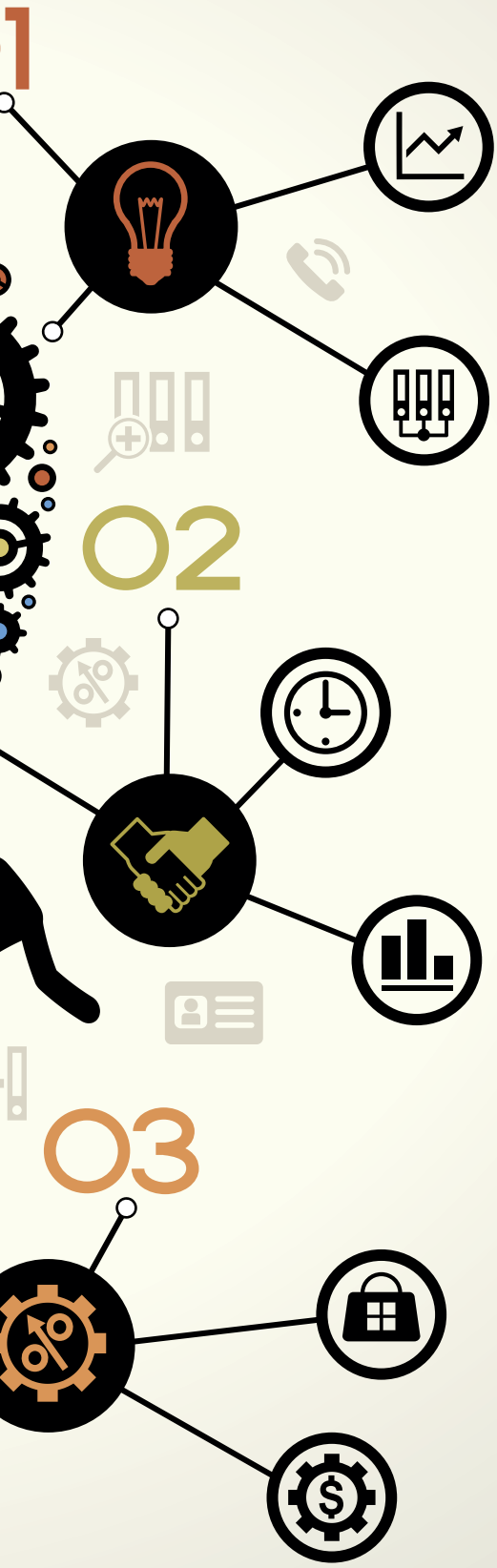
[b-ITU-T Y.2002] Recommendation ITU-T Y.2002 (2009), *Overview of ubiquitous networking and of its support in NGN*.

[b-ITU-T Y.4000] Recommendation ITU-T Y.4000/Y.2060 (2012), *Overview of the Internet of Things*.

[b-ITU-T Y.4400] Recommendation ITU-T Y.4400/Y.2063 (2012), *Framework of Web of Things*.




06
05
04



Y.4413/F.748.5

Requirements and reference architecture of the machine-to-machine service layer

Requirements and reference architecture of the machine-to-machine service layer

Summary

Recommendation ITU-T Y.4413/F.748.5 identifies requirements of the machine-to-machine (M2M) service layer, which are common to all M2M verticals or specific to e-health application support, and provides an architectural framework of the M2M service layer.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T Y.4413/F.748.5	2015-11-29	16	11.1002/1000/12623

Keywords

Internet of things, IoT, M2M, machine-to-machine, service layer.

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

Table of Contents

Recommendation ITU-T Y.4413/F.748.5

Requirements and reference architecture of the machine-to-machine service layer

1 Scope

This Recommendation identifies requirements of the machine-to-machine (M2M) service layer, which are common to all M2M verticals or specific to e-health application support, and to provide an architectural framework of the M2M service layer.

In particular, the scope of this Recommendation includes:

– definition of the M2M service layer;

– requirements of the M2M service layer;

– architectural framework of the M2M service layer;

– reference points of the M2M service layer.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T Y.4000] Recommendation ITU-T Y.4000/Y.2060 (2012), *Overview of the Internet of things*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 application dedicated node [b-oneM2M-TS-0011]: A node that contains at least one application entity and does not contain a common services entity. There may be zero or more application dedicated nodes (ADNs) in the field domain of the oneM2M system.

3.1.2 application entity [b-oneM2M-TS-0011]: Represents an instantiation of application logic for end-to-end M2M solutions.

3.1.3 application service node [b-oneM2M-TS-0011]: A node that contains one common services entity and contains at least one application entity. There may be zero or more ASNs in the field domain of the oneM2M system.

3.1.4 common service entity [b-oneM2M-TS-0011]: Represents an instantiation of a set of common service functions of the M2M environments. Such service functions are exposed to other entities through reference points.

3.1.5 infrastructure node [b-oneM2M-TS-0011]: A node that contains one common services entity and contains zero or more application entities. There is exactly one infrastructure node in the infrastructure domain per oneM2M service provider.

3.1.6 IoT [ITU-T Y.4000]: A global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on, existing and evolving, interoperable information and communication technologies.

NOTE 1 – Through the exploitation of identification, data capture, processing and communication capabilities, the IoT makes full use of things to offer services to all kinds of applications, whilst ensuring that security and privacy requirements are fulfilled.

NOTE 2 – In a broad perspective, the IoT can be perceived as a vision with technological and societal implications.

3.1.7 middle node [b-oneM2M-TS-0011]: A node that contains one common services entity and contains zero or more application entities. There may be zero or more middle nodes in the field domain of the oneM2M system.

3.1.8 node [b-oneM2M-TS-0011]: Functional entity containing one of the following: one or more M2M applications; one CSE and zero or more M2M applications.

3.2 Terms defined in this Recommendation

None.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

ADN	Application Dedicated Node
AE	Application Entity
ASN	Application Service Node
BAN	Body Area Network
CSE	Common Service Entity
CSF	Common Services Function
DA	Device Application
GA	Gateway Application
IN	Infrastructure Node
IoT	Internet of Things
M2M	Machine-to-Machine
MN	Middle Node
NA	Network Application
NSE	Network Service Entity
SCL	Service Capabilities Layer
SL	Service Layer

5 Conventions

None.

6 Definition of the ITU-T M2M service layer

6.1 The ITU-T M2M service layer and its relationship with the IoT reference model

From the ITU-T perspective, the machine-to-machine (M2M) technologies are a key enabler of the Internet of things (IoT), [ITU-T Y.4000].

The M2M service layer in the ITU-T scope, the "ITU-T M2M service layer", includes a set of generic and specific functions for the support of a variety of applications enabled by the M2M technologies. These include management functions and security functions, as well as service support and application support functions. The capabilities of the ITU-T M2M service layer are a subset of the entire set of capabilities of the IoT.

Figure 1 shows the ITU-T M2M service layer and its position in the IoT reference model [ITU-T Y.4000].

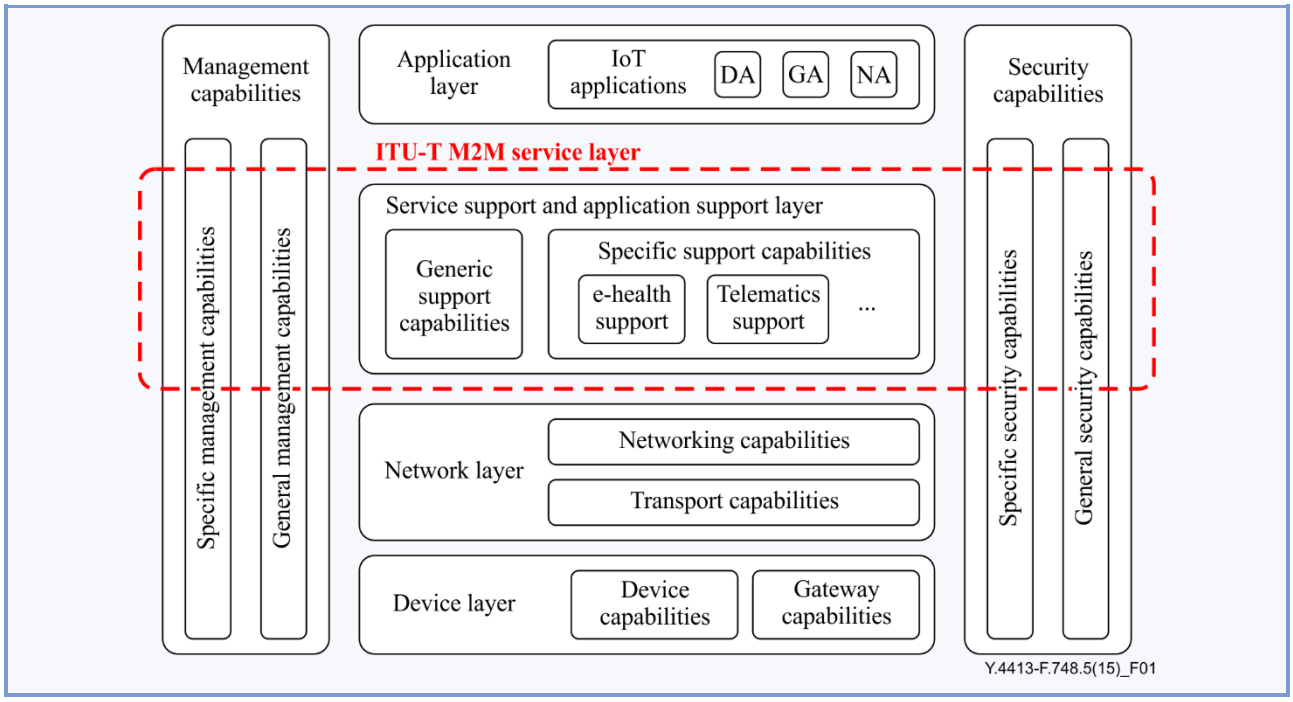


Figure 1 – The ITU-T M2M service layer in the IoT reference model

The layered architectural approach, as illustrated in Figure 1, reduces the implementation complexity while providing interoperability between the various applications enabled by the M2M technologies.

NOTE – Other architectural approaches are out of scope of this Recommendation. It is recognized that cross layer architectural approaches can show higher performances, but at the expense of higher implementation complexity.

The specific support capabilities in the service support and application support layer include application specific support capabilities (e.g., e-health support, telematics support) as shown in Figure 1).

Three types of applications are identified on top of the ITU-T M2M service layer (application layer): device application (DA), gateway application (GA) and network application (NA) servers. DA, GA and NA reside, respectively, in a device, gateway and network application server. All these applications can use capabilities provided by the ITU-T M2M service layer.

6.2 The ETSI M2M service capabilities layer and its relationship with the ITU-T M2M service layer

The ETSI M2M service capabilities layer (SCL) [b-ETSI 102 690] provides functions that are shared by different applications enabled by the M2M technologies, and can be positioned with respect to the IoT reference model described in [ITU-T Y.4000] as shown in Figure 2.

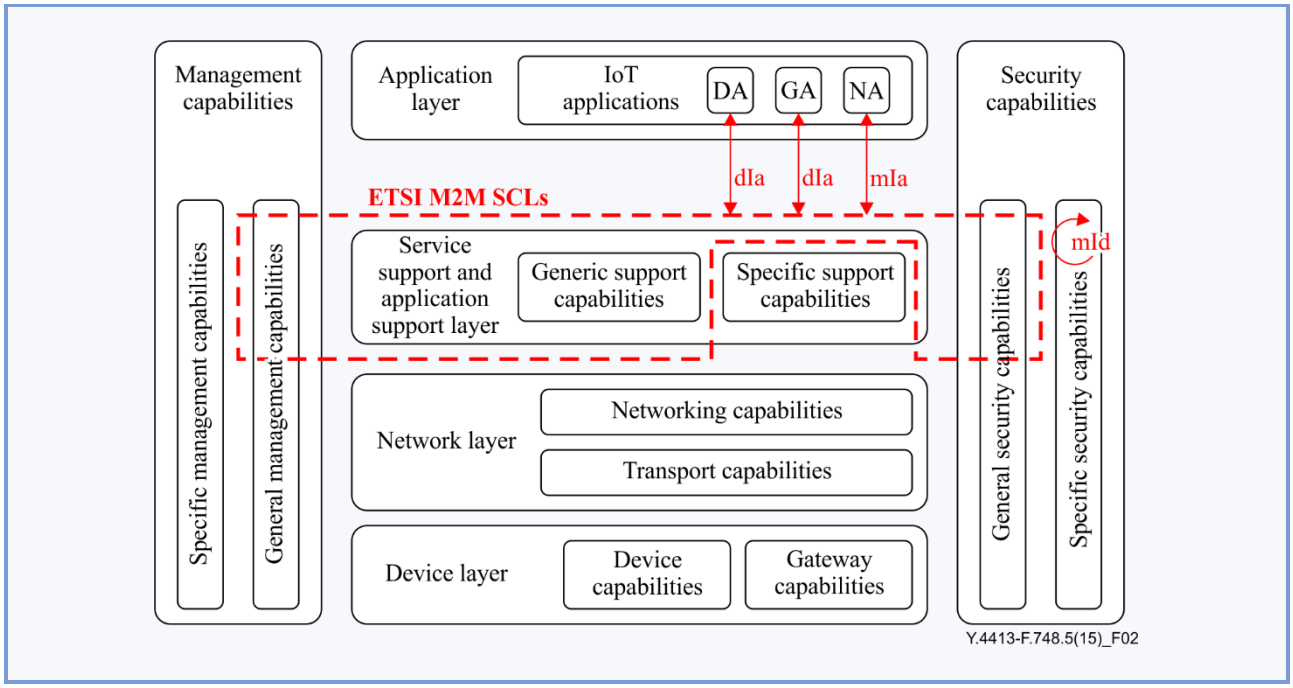


Figure 2 – ETSI M2M SCL in the IoT reference model

In Figure 2, dIa and mIa from [b-ETSI 102 690] can be considered as reference points between IoT applications and the service support and application support layer with inclusion of the general management capabilities and general security capabilities. mId from [b-ETSI 102 690] can be considered as the reference point between the service support and application support layer of different devices.

As shown in Figure 2, the ETSI M2M SCL includes only general functions of service support and application support layer, general management capabilities and general security capabilities.

Compared to the ETSI M2M SCL, the ITU-T M2M service layer includes specific support capabilities in the service support and application support layer, specific management capabilities and specific security capabilities as shown in Figure 1.

It is anticipated that dIa, mIa and mId from [b-ETSI 102 690] may need extension to include the support of the specific support capabilities in the service support and application support layer, the specific management capabilities and the specific security capabilities.

6.3 The oneM2M common service entity and its relationship with the ITU-T M2M service layer

The oneM2M common service entity (CSE) [b-oneM2M-TS-0001] means a group of the twelve common service functions (CSFs) that are shared by different applications enabled by the M2M application service provider and/or user, and can be positioned with respect to the IoT reference model described in [ITU-T Y.4000], as shown in Figure 3.

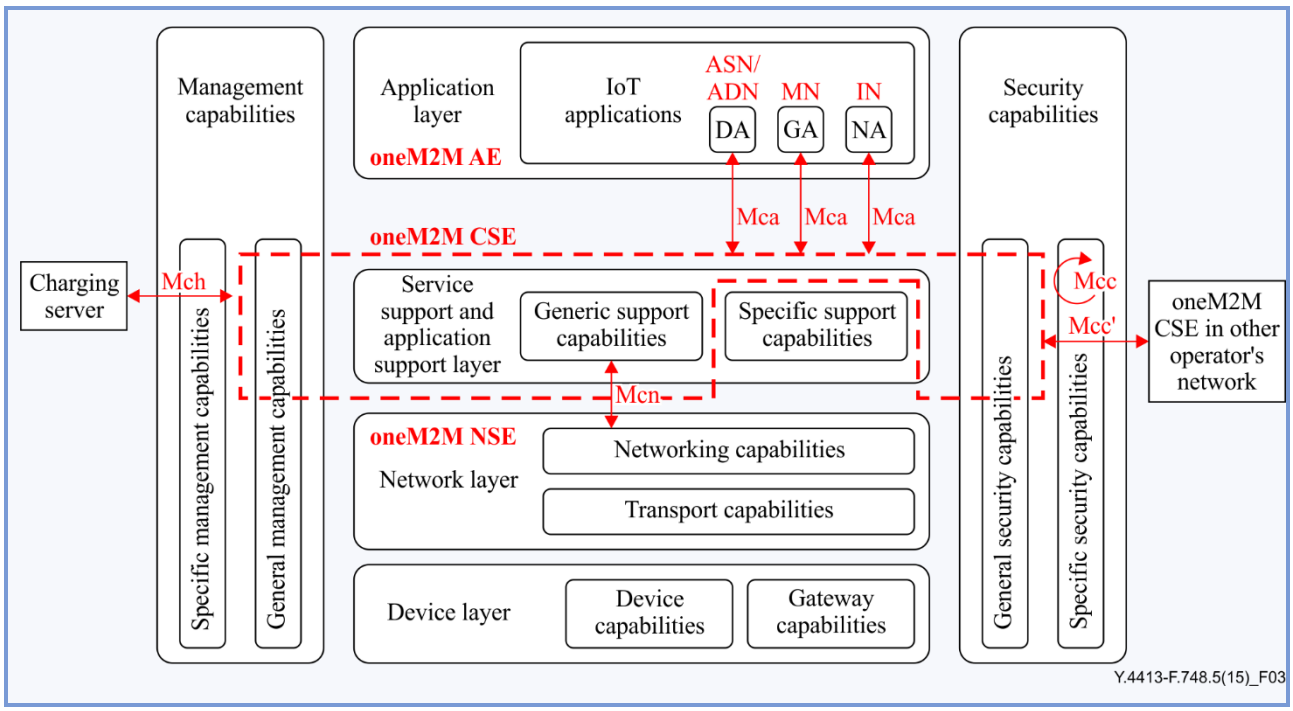


Figure 3 – oneM2M functional architecture in the IoT reference model

In Figure 3, Mca can be considered as reference points between IoT applications and the service support and application support layer with inclusion of the general management capabilities and general security capabilities. Mch can be considered as reference points between charging server and the service support and application support layer and with inclusion of the general management capabilities and general security capabilities. Mcc and Mcc' can be considered as the reference point between the service support and application support layer of different nodes. Here, Mcc' can be considered to connect to the CSE of other service providers.

As shown in Figure 3, the oneM2M CSE, like the ETSI M2M SCL, includes only general functions of service support and application support layer, general management capabilities and general security capabilities.

Compared to the oneM2M CSE, the ITU-T M2M service layer includes specific support capabilities in the service support and application support layer, specific management capabilities and specific security capabilities as shown in Figure 1.

7 Requirements of the ITU-T M2M service layer

7.1 Common requirements

7.1.1 Communication management

– Message scheduling:

  The ITU-T M2M service layer is required to support various priorities of messages.

– Various types of communications:

  The ITU-T M2M service layer is required to support various types of communication (e.g., on-demand, continuous) requested by applications. It should also support notification of communication failure.

– Various underlying network technologies support:

  The ITU-T M2M service layer is required to support underlying network technologies.

7.1.2 Application management

– Multiple applications support:

The ITU-T M2M service layer is required to support multiple applications concurrently.

7.1.3 Service and device discovery and registration

– Service and device discovery and registration:

The ITU-T M2M service layer is required to support service and device discovery and registration.

7.1.4 Service accounting and charging

– Service accounting and charging:

The ITU-T M2M service layer is required to support service accounting and charging.

7.1.5 Device management

– Auto configuration:

The ITU-T M2M service layer is required to support auto configuration and configuration management of devices and upgrading of software on the devices in a secure way.

– Management of multiple devices and various types of devices:

The ITU-T M2M service layer is required to support management of multiple devices and various types of devices.

7.1.6 Data processing

– Data storage and notification:

The ITU-T M2M service layer is recommended to provide capability of data storage for applications. Once data are updated, the ITU-T M2M service layer should inform subscribed applications.

– Data formatting and translation:

The ITU-T M2M service layer is recommended to provide capability of data formatting and translation to facilitate semantic interoperation between applications.

– Data collection and reporting:

The ITU-T M2M service layer is required to support both on-demand and periodic reporting as requested by applications.

7.1.7 Diagnostics and fault recovery

– Diagnostics and fault recovery:

The ITU-T M2M service layer is required to support diagnostic mechanisms for applications and devices. In addition, it should support fault recovery and fault management to recognize, isolate, correct and log faults that occur.

7.1.8 Identification, naming and addressing

– Reachability of devices by identification:

The ITU-T M2M service layer is required to support reachability of devices based on device identification.

7.1.9 Security

– Authentication:

The ITU-T M2M service layer is required to provide authentication mechanisms for applications and devices and prevent unauthorized use of the devices.

– Privacy:

The ITU-T M2M service layer is required to support privacy protection capabilities, such as anonymity of identity and location, according to regulation and laws.

– Confidentiality:

The ITU-T M2M service layer is required to support data transfer confidentiality.

– Integrity:

The ITU-T M2M service layer is required to support data integrity protection.

– Support of security for service scenarios involving multiple actors:

The ITU-T M2M service layer is required to support security capabilities, such as supporting user access control of protected data, for M2M service scenarios involving multiple actors inside a single administrative domain and across different administrative domains (e.g., countries, operators).

– Availability:

The ITU-T M2M service layer is required to support data availability. Typically, data, and their related-functions or services must be available whenever they are needed.

7.1.10 Location provisioning

– Location information:

The ITU-T M2M service layer is recommended to support collection, tracking and reporting of location information according to different collection, tracking and reporting strategies.

7.1.11 Group management

– Group management:

The ITU-T M2M service layer is required to support a mechanism to create and manage virtual group of devices.

7.2 e-health specific requirements

– Security for personal health information:

The ITU-T M2M service layer is required to provide security capabilities in compliance with regulation and laws regarding personal health information (personal data and medical data).

– Privacy protection:

The ITU-T M2M service layer is required to provide privacy protection capabilities for personal health information in compliance with regulation and laws (personal data and medical data when they are associated with a person's identification).

– e-health device profile support:

The ITU-T M2M service layer is required to support e-health device profile according to international standards (e.g., medical body area network (BAN) [b-IEEE 802.15.6], Bluetooth [b-Bluetooth]).

– Time synchronization and timestamping:

The ITU-T M2M service layer is required to support timestamping since health conditions vary over time. With timestamping, e-health applications can obtain useful information according to the health condition history. For support of timestamping, the ITU-T M2M service layer is also required to retrieve time parameters from authoritative time servers and publish time parameters according to the requests from e-health applications.

– Audit trail support:

The ITU-T M2M service layer is required to support audit trails ensuring that any access or attempt to access personal health information is fully transparent, traceable and reproducible.

8 Architectural framework of the ITU-T M2M service layer

8.1 Overview of the architectural framework of the ITU-T M2M service layer

As described in clause 6, the ITU-T M2M service layer is positioned between the application layer and the network layer, and provides various types of capabilities, including generic support capabilities, specific support capabilities, general and specific management capabilities, general and specific security capabilities.

Figure 4 shows the architectural framework of the ITU-T M2M service layer.

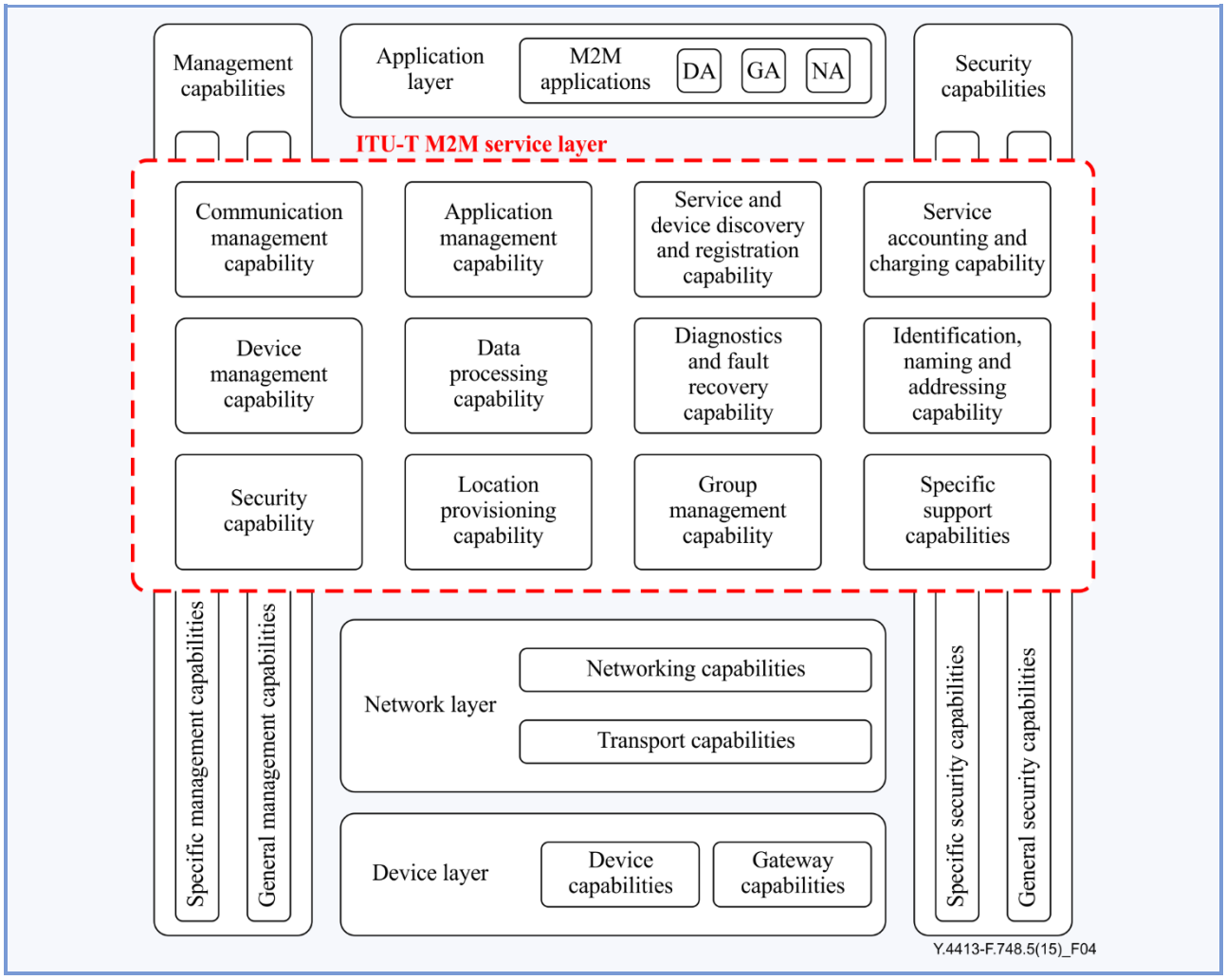


Figure 4 – The ITU-T M2M service layer architectural framework

8.2 The capabilities of the ITU-T M2M service layer

NOTE – The capabilities in clause 8.2 (i.e., 8.2.1 through 8.2.11) correspond to the requirements in clause 7.1 (i.e., 7.1.1 through 7.1.11), respectively.

8.2.1 Communication management

This capability supports message scheduling, various types of communications and various underlying network technologies.

8.2.2 Application management

This capability supports multiple applications.

8.2.3 Service and device discovery and registration

This capability supports service and device discovery and registration.

8.2.4 Service accounting and charging

This capability supports accounting and different charging models, including both online and offline charging.

8.2.5 Device management

This capability supports auto configuration, management of multiple devices and various types of devices.

8.2.6 Data processing

This capability supports data storage, notification, formatting, translation, collection and reporting.

8.2.7 Diagnostics and fault recovery

This capability supports recognition, isolation, correction and logging of the faults that occur in the application layer and the ITU-T M2M service layer.

8.2.8 Identification, naming and addressing

This capability supports reachability of devices based on device identification, naming and addressing.

8.2.9 Security

This capability supports authentication, privacy protection, confidentiality, integrity and support of security for service scenarios involving multiple actors.

8.2.10 Location provisioning

This capability supports the acquisition and management of location information based on the requests from applications.

8.2.11 Group management

This capability supports mechanisms to create and manage virtual group of devices.

8.2.12 Specific support

These capabilities are support capabilities that apply to specific applications. These capabilities are out of scope of this Recommendation.

9 Reference points of the ITU-T M2M service layer

9.1 Overview of the reference points

Figure 5 shows the reference points of the ITU-T M2M service layer.

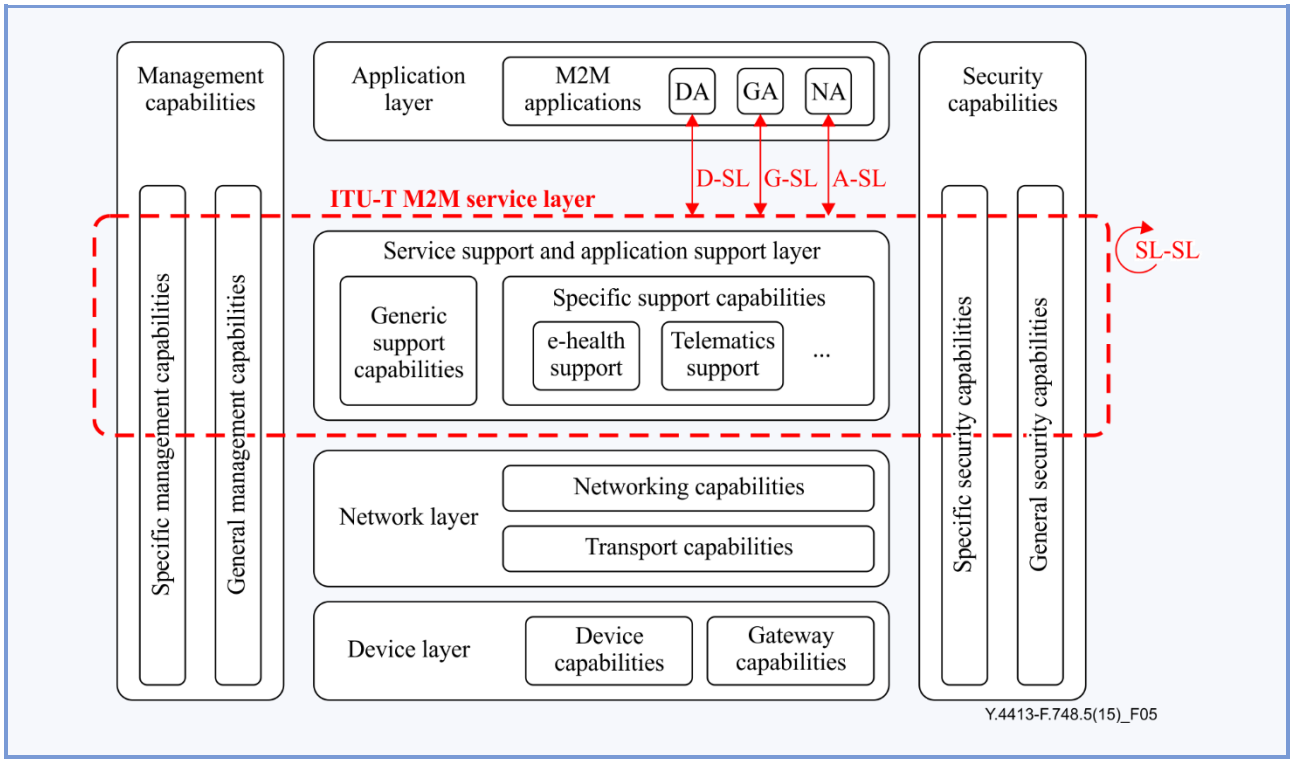


Figure 5 – Reference points of the ITU-T M2M service layer

As described in clause 6.1, three types of applications are identified on top of the ITU-T M2M service layer (application layer): DA, GA and NA servers. DA, GA and NA reside, respectively, in a device, gateway and network application server. All these applications can use capabilities provided by the ITU-T M2M service layer.

Four different reference points are identified for the ITU-T M2M service layer: D-SL, G-SL, A-SL and SL-SL. D-SL is the reference point between the DA and the ITU-T M2M service layer, G-SL is the reference point between the GA and the ITU-T M2M service layer, A-SL is the reference point between NA and the ITU-T M2M service layer, and SL-SL is the reference point between the ITU-T M2M service layers residing, respectively, in device, gateway and network application servers.

9.2 Details on the reference points

Figure 6 provides a detailed illustration of the reference points described in Figure 5.

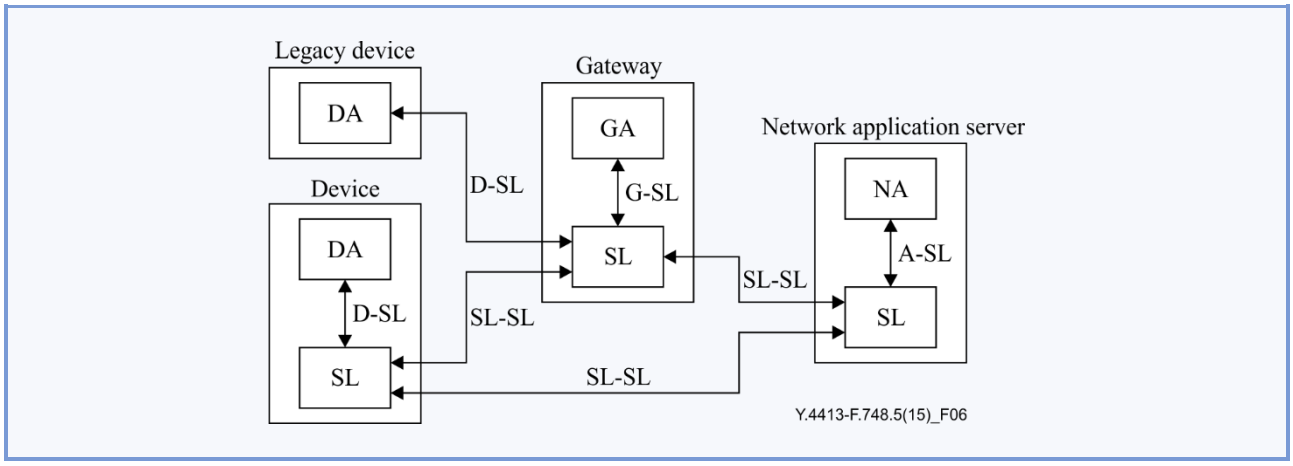


Figure 6 – Reference points between device, gateway and network application server

The D-SL reference point allows a device application in a device to access the ITU-T M2M service layer in the same device or in the gateway. The D-SL reference point between device application and service layer in a gateway is for legacy devices which do not have ITU-T M2M service layer capabilities.

The G-SL reference point allows a gateway application in a gateway to access the ITU-T M2M service layer in the same gateway.

The A-SL reference point allows a network application server to access the ITU-T M2M service layer in the same network application server.

The SL-SL reference point allows the ITU-T M2M service layer in a device, gateway or network application server to access the ITU-T M2M service layer in a different device, gateway or network application server.

Appendix I

Comparison between the capabilities of the ITU-T M2M service layer and common services functions of oneM2M

(This appendix does not form an integral part of this Recommendation.)

This appendix provides comparison between the capabilities of the ITU-T M2M service layer and CSFs of oneM2M. oneM2M defines CSE as Figure I.1 in [b-oneM2M-TS-0001].

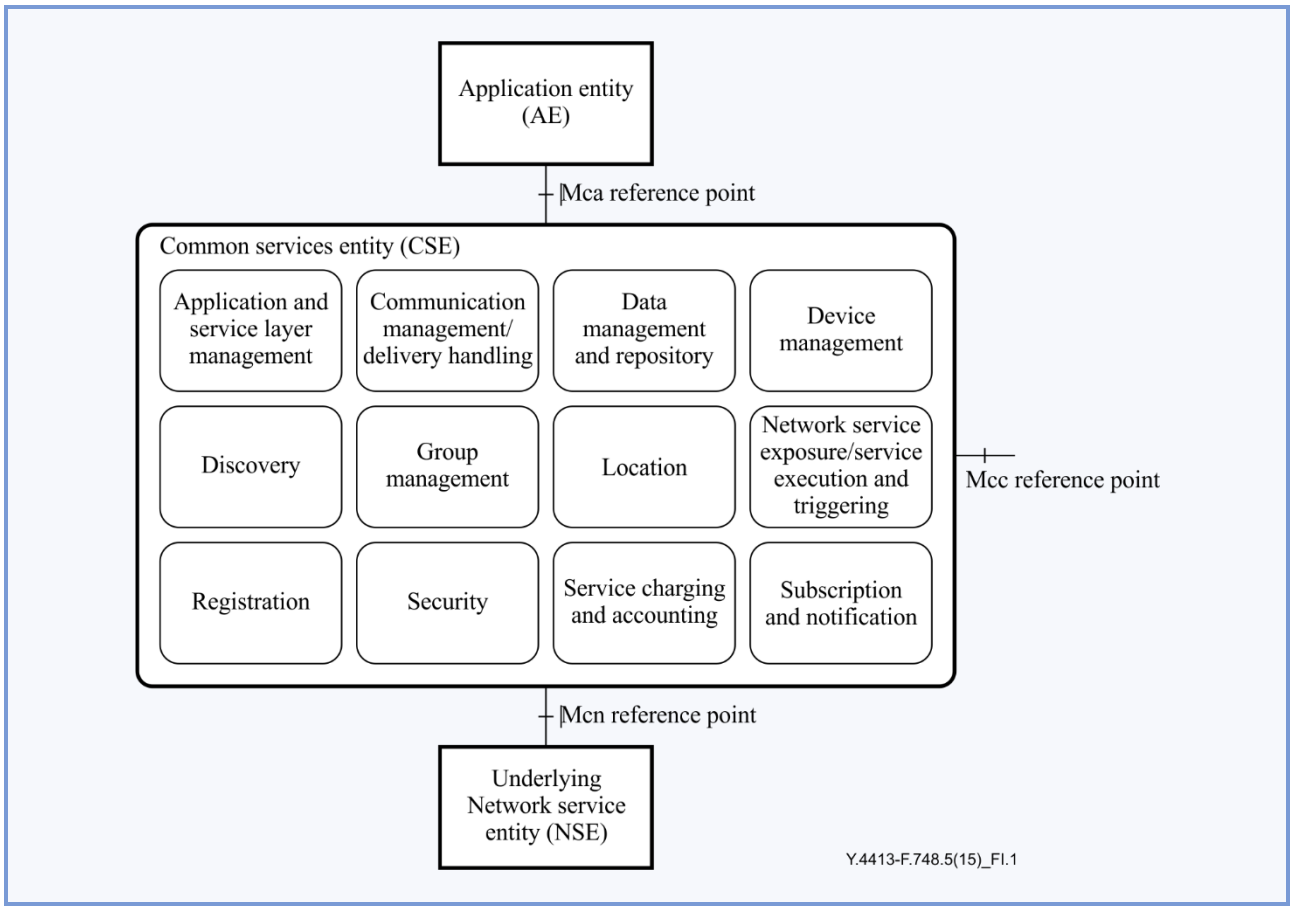


Figure I.1 – oneM2M common services functions

Figure I.2 shows the summary of comparison between the capabilities of the ITU-T M2M service layer and the CSFs of oneM2M.

For example, some functions of communication management/delivery handling and device management in CSFs of oneM2M is covered by communication management capability of the ITU-T M2M service layer.

The main differences between them are that oneM2M CSF does not dealing with the application-specific support and ITU-T M2M service layer does not dealing with the subscription and notification.

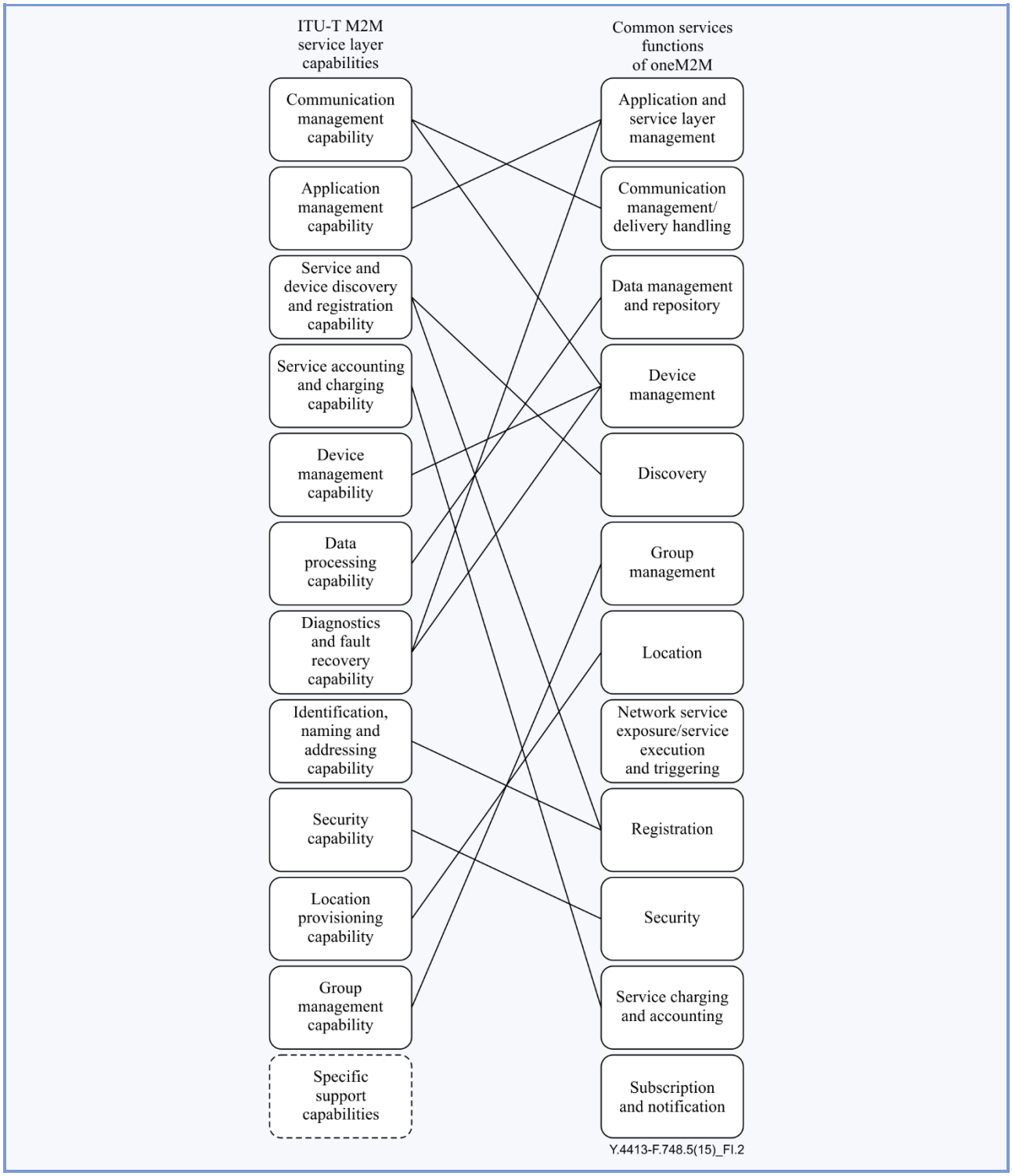


Figure I.2 – Comparison between the capabilities of the ITU-T M2M service layer and the CSFs of oneM2M

Appendix II

Comparison of reference points between the ITU-T M2M service layer and common services entity of oneM2M

(This appendix does not form an integral part of this Recommendation.)

Figure II.1 shows comparisons of reference points between the ITU-T M2M service layer and the CSE of oneM2M.

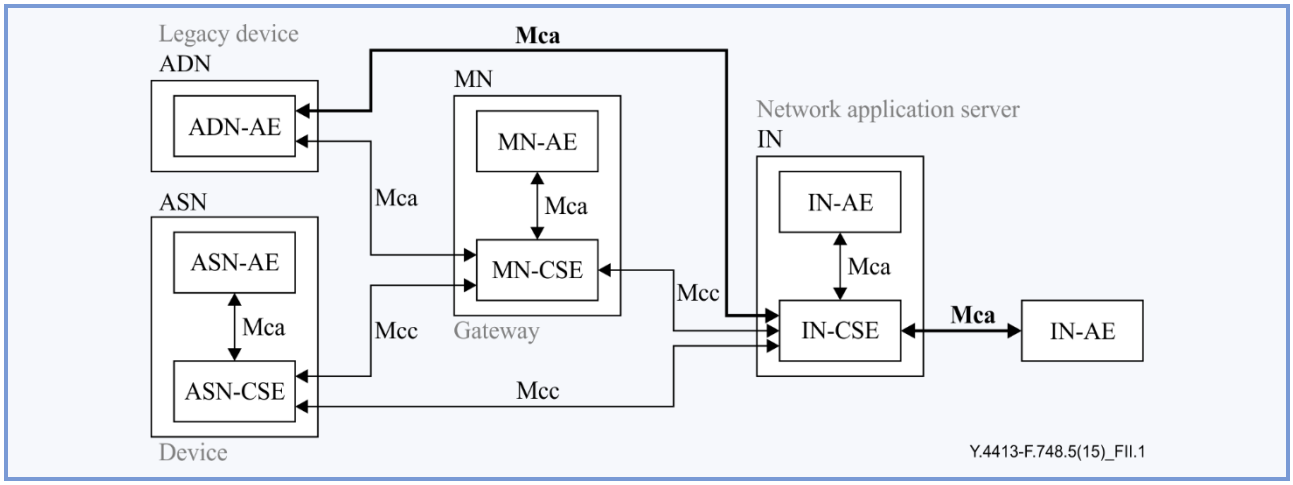


Figure II.1 – Comparison of reference points between the ITU-T M2M service layer and the CSEs of oneM2M

Compared to the ITU-T M2M service layer, oneM2M CSE has additional reference points between ASN-AE and IN-CSE and between IN-CSE and IN-AE.

Bibliography

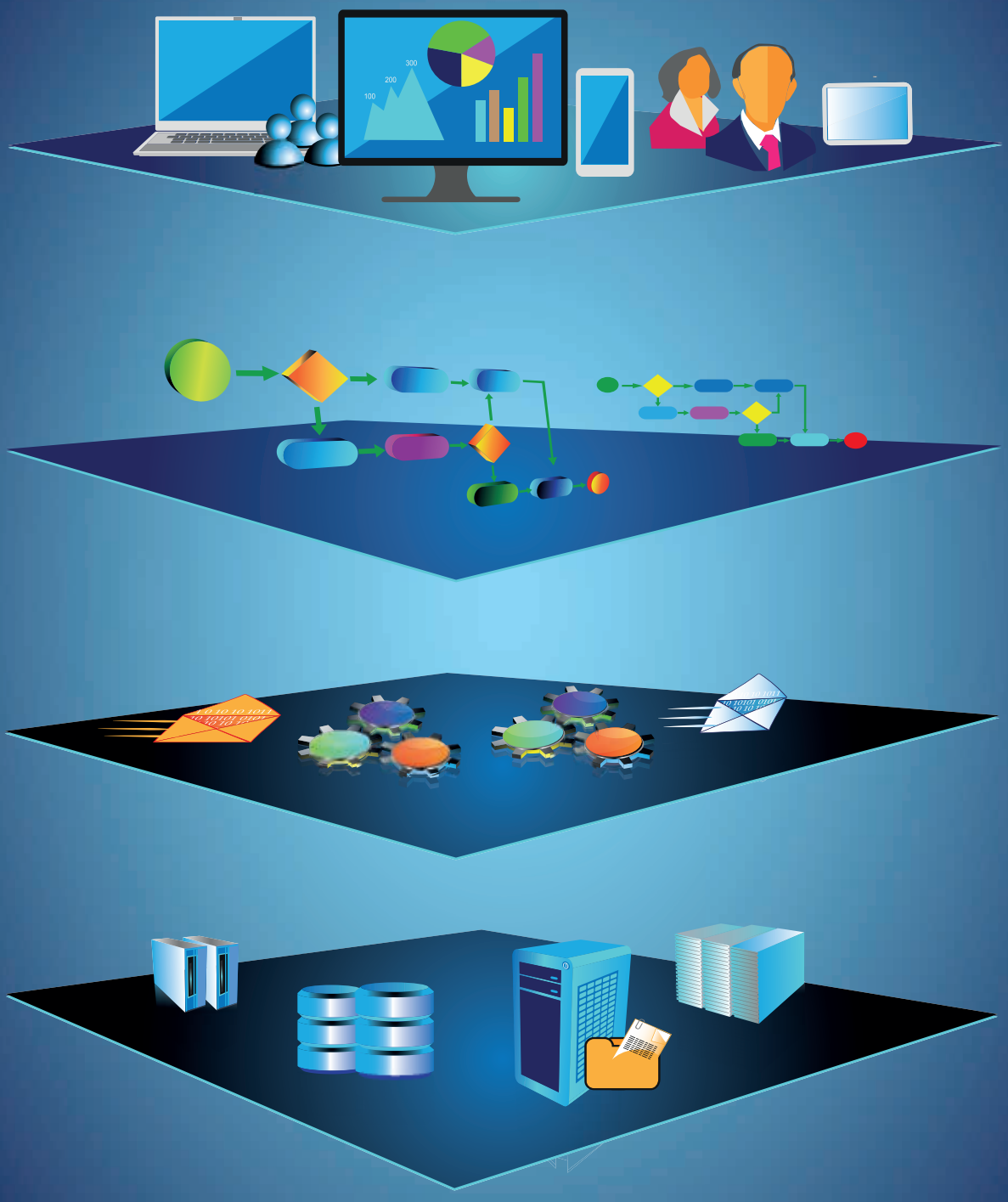
[b-Bluetooth] Bluetooth, *adopted specifications*. <https://www.bluetooth.org/en-us/specification/adopted-specifications>

[b-ETSI 102 690] ETSI TS 102 690 v2.1.1 (2013), *Machine-to-Machine communications (M2M): Functional architecture*.

[b-IEEE 802.15.6] IEEE 802.15.6 (2012), *IEEE Standard for Local and metropolitan area networks - Part 15.6: Wireless Body Area Networks*. <http://standards.ieee.org/findstds/standard/802.15.6-2012.html>, accessed on 2014-06-10.

[b-oneM2M-TS-0001] oneM2M-TS-0001-V-2014-08 (2014), *oneM2M Functional Architecture Baseline Draft*.

[b-oneM2M-TS-0011] oneM2M-TS-0011-V-2014-08 (2014), *oneM2M Definitions and Acronyms*.


300
200
100


XML

Y.4414/H.623

Web of things service architecture

Web of things service architecture

Summary

Recommendation ITU-T Y.4414/H.623 defines a web of things (WoT) service architecture that can encompass service discovery, accessibility, sharing and mash-up for IoT devices and services with web technologies. It includes an overview of WoT services, the functional architecture of WoT services and WoT service/resource functions.

The WoT service architecture supports accessibility and reusability across IoT resources, and supports portability across heterogeneous network environments. Therefore, this Recommendation is applicable to seamless and interoperable IoT services with information interaction and exchange over physical IoT devices.

History

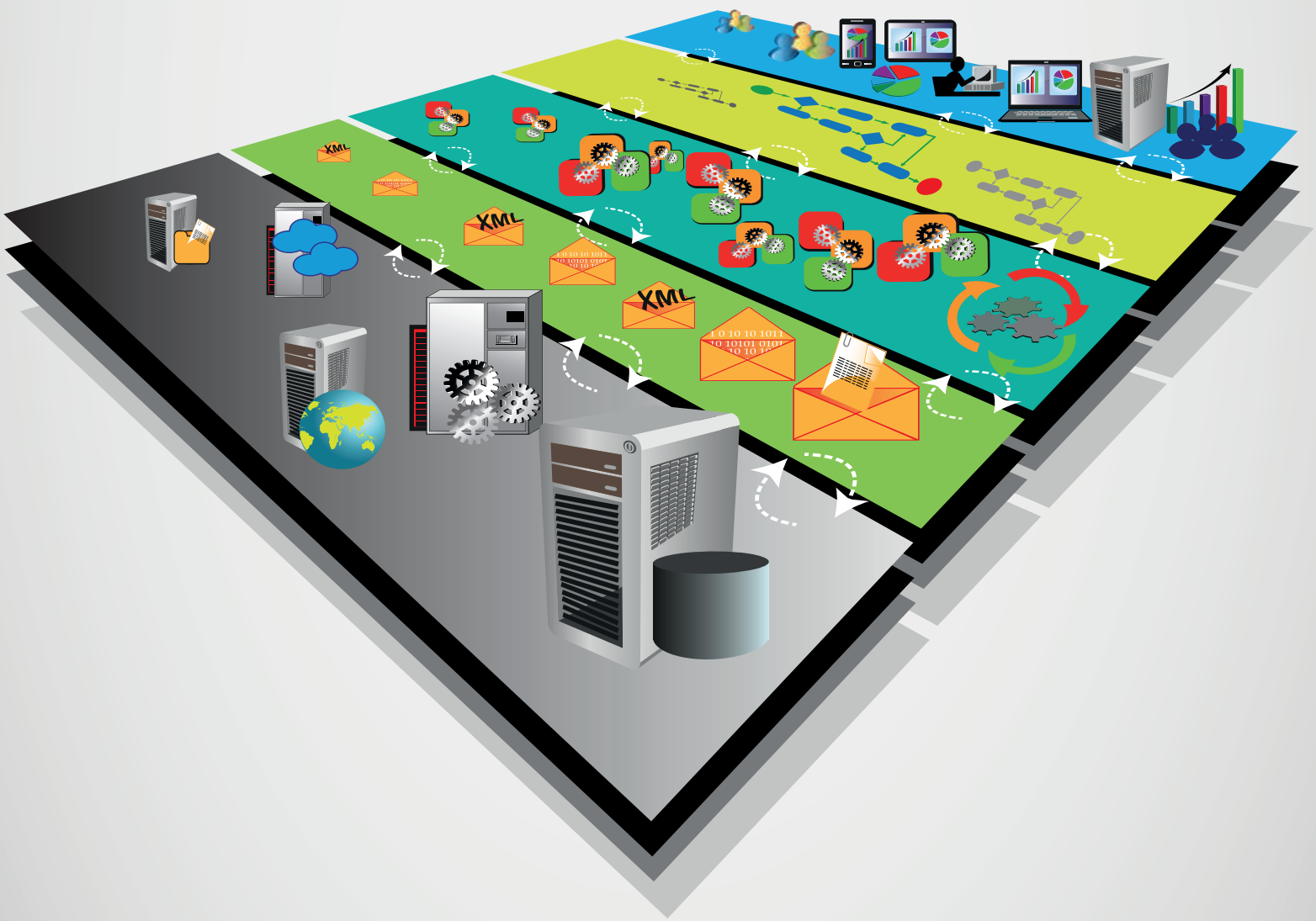
Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T Y.4414/H.623	2015-11-29	16	11.1002/1000/12647

Keywords

Web of things, WoT, WoT service architecture.

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

Table of Contents


XML
XML
XML

Recommendation ITU-T Y.4414/H.623

Web of things service architecture

1 Scope

The objective of this Recommendation is to define a WoT service architecture. The scope of this Recommendation covers the following:

– overview of the WoT applications

– a functional architecture for the WoT service

– functional entities for the WoT service

2 References

The following ITU-T Recommendations and other references contain provisions, which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T M.3030] Recommendation ITU-T M.3030 (2002), *Telecommunications Markup Language (tML) framework*.

[ITU-T Y.2002] Recommendation ITU-T Y.2002 (2009), *Overview of ubiquitous networking and of its support in NGN*.

[ITU-T Y.2232] Recommendation ITU-T Y.2232 (2008), *NGN convergence service model and scenario using web services*.

[ITU-T Y.4000] Recommendation ITU-T Y.4000/Y.2060 (2012), *Overview of the Internet of things*.

[ITU-T Y.4400] Recommendation ITU-T Y.4400/Y.2063 (2012), Framework of the w*eb of things*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 client [b-W3C WACterms]: The role adopted by an application when it is retrieving and/or rendering resources. A program establishes connections for the purpose of sending requests.

3.1.2 device [b-W3C digloss]: An apparatus through which a user can perceive and interact with the Web.

3.1.3 Internet of things (IoT) [ITU-T Y.4000]: A global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on, existing and evolving, interoperable information and communication technologies.

NOTE 1 – Through the exploitation of identification, data capture, processing and communication capabilities, the IoT makes full use of things to offer services to all kinds of applications, whilst ensuring that security and privacy requirements are fulfilled.

NOTE 2 – In a broad perspective, the IoT can be perceived as a vision with technological and societal implications.

3.1.4 metadata [ITU-T M.3030]: Data that describes other data.

3.1.5 resource [b-IETF RFC 3986]: This specification does not limit the scope of what might be a resource; rather, the term "resource" is used in a general sense for whatever might be identified by a URI. Familiar examples include an electronic document, an image, a source of information with a consistent purpose (e.g., "today's weather report for Los Angeles"), a service (e.g., an HTTP-to-SMS gateway), and a collection of other resources. A resource is not necessarily accessible via the Internet; e.g., human beings, corporations, and bound books in a library can also be resources. Likewise, abstract concepts can be resources, such as the operators and operands of a mathematical equation, the types of a relationship (e.g., "parent" or "employee"), or numeric values (e.g., zero, one, and infinity).

3.1.6 server [b-W3C WACterms]: The role adopted by an application when it is supplying resources.

3.1.7 service provider [ITU-T Y.2232]: An entity that provides services.

3.1.8 SOAP [b-W3C SOAP 1]: The formal set of conventions governing the format and processing rules of a SOAP message. These conventions include the interactions among SOAP nodes generating and accepting SOAP messages for the purpose of exchanging information along a SOAP message path.

3.1.9 SOAP intermediary [b-W3C SOAP 1]: A SOAP intermediary is both a SOAP receiver and a SOAP sender and is targetable from within a SOAP message. It processes the SOAP header blocks targeted at it and acts to forward a SOAP message towards an ultimate SOAP receiver.

3.1.10 the World Wide Web (WWW, or simply web) [b-W3C webarch]: An information space in which the items of interest, referred to as resources, are identified by global identifiers called Uniform Resource Identifiers (URI).

3.1.11 thing [ITU-T Y.4000]: In the Internet of things, object of the physical world (physical things) or of the information world (virtual things), which is capable of being identified and integrated into the communication networks.

3.1.12 URI [b-IETF RFC 3986]: A URI is an identifier consisting of a sequence of characters matching the syntax rule named <URI> in Section 3 of [b-IETF RFC 3986]. It enables uniform identification of resources via a separately defined extensible set of naming schemes (Section 3.1 of [b-IETF RFC 3986]). How that identification is accomplished, assigned, or enabled is delegated to each scheme specification.

3.1.13 web of things [ITU-T Y.4400]: A way of realization of the IoT where (physical and virtual) things are connected and controlled through the World Wide Web.

3.1.14 web resource [b-W3C WACterms]: A resource, identified by a URI.

3.1.15 web services [ITU-T Y.2232]: Web services is a service provided using web services systems.

3.1.16 web services gateway (WSG) [ITU-T Y.2232]: A gateway which handles the web services message between the WSP and WSR.

3.1.17 web services provider (WSP) [ITU-T Y.2232**]**: A service provider that exposes a capability for use to create web services.

3.1.18 web services registry [ITU-T Y.2232]: An entity that stores web services information (e.g., WSDL).

3.1.19 Web services requester (WSR) [ITU-T Y.2232]: Client software that makes use of the services provided by a WSP.

3.1.20 web services system [ITU-T Y.2232]: A web services system is a software system designed to support interoperable machine-to-machine interaction over a network using web services standards.

NOTE – It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the web services in a manner prescribed by its description using SOAP-messages, typically conveyed using HTTP with an XML serialization in conjunction with other web-related standards.

3.1.21 WSDL [b-W3C WSDL 1]: Web services description language Version 2.0 (WSDL 2.0) provides a model and an XML format for describing web services. WSDL 2.0 enables one to separate the description of the abstract functionality offered by a service from concrete details of a service description such as "how" and "where" that functionality is offered.

3.2 Terms defined in this Recommendation

None.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

API	Application Programming Interface
CPU	Central Processing Unit
DPWS	Device Profile for Web Services
GPU	Graphics Processing Unit
HTML	Hypertext Markup Language
HTTP	Hypertext Transport Protocol
ID	Identification
IoT	Internet of Things
IP	Internet Protocol
JSON	JavaScript Object Notation
NFC	Near Field Communication
QoS	Quality of Service
RDFa	Resource Description Framework in attributes
REST	Representational State Transfer
SOAP	Simple Object Access Protocol
TCP	Transmission Control Protocol
UI	User Interface
URI	Uniform Resource Identifiers
Web-RM FE	Web Resource Management Functional Entity
WoT	Web of Things
WoT-BM FE	WoT Broker Management Functional Entity
WoT-DM FE	WoT Description Management Functional Entity
WoT-EN FE	WoT Enrolment Functional Entity
WoT RMF	WoT Resource Management Function
WoT-S CF	WoT Service Control Function
WoT-S SF	WoT Service Support Function
WoT-S-CR FE	WoT Service Creation Functional Entity

WoT-S-DIS FE	WoT Service Discovery Functional Entity
WoT-S-EX FE	WoT Service Execution Functional Entity
WoT-S-MA FE	WoT Service Management Functional Entity
WoT-S-MON FE	WoT Service Monitoring Functional Entity
WoT-S-PM FE	WoT Service Policy Management Functional Entity
WoT-S-PRM FE	WoT Service Profile Management Functional Entity
WoT-S-RE FE	WoT Service Registration Functional Entity
WSDL	Web Service Description Language
XML	extensible Markup Language

5 Conventions

None.

6 Overview of the WoT service

The web is used as a global standard platform to deliver services to an end user and applications to intercommunicate with each other over a network. [ITU-T Y.4400] describes that the web has program language independent properties, uses message driven communications and easily bounds to different transport protocols. Web technology allows the exposure of physical devices as well as several kinds of content as web resources such as URI and HTTP. Therefore, users and service developers can interact with the devices using web interfaces. The WoT can provide capabilities such as device reusability, portability across several heterogeneous networks and accessibility based on web standards technologies.

Figure 1 shows a general concept of the WoT service. The physical devices as well as content are integrated into the web and are viewed as the sort of services provided by the web environment. In addition, each service has a mash-up capability to create a new service by composition of the existing services in the web environments. According to [ITU-T Y.4400], the web technologies can support to create applications of networked devices and services.

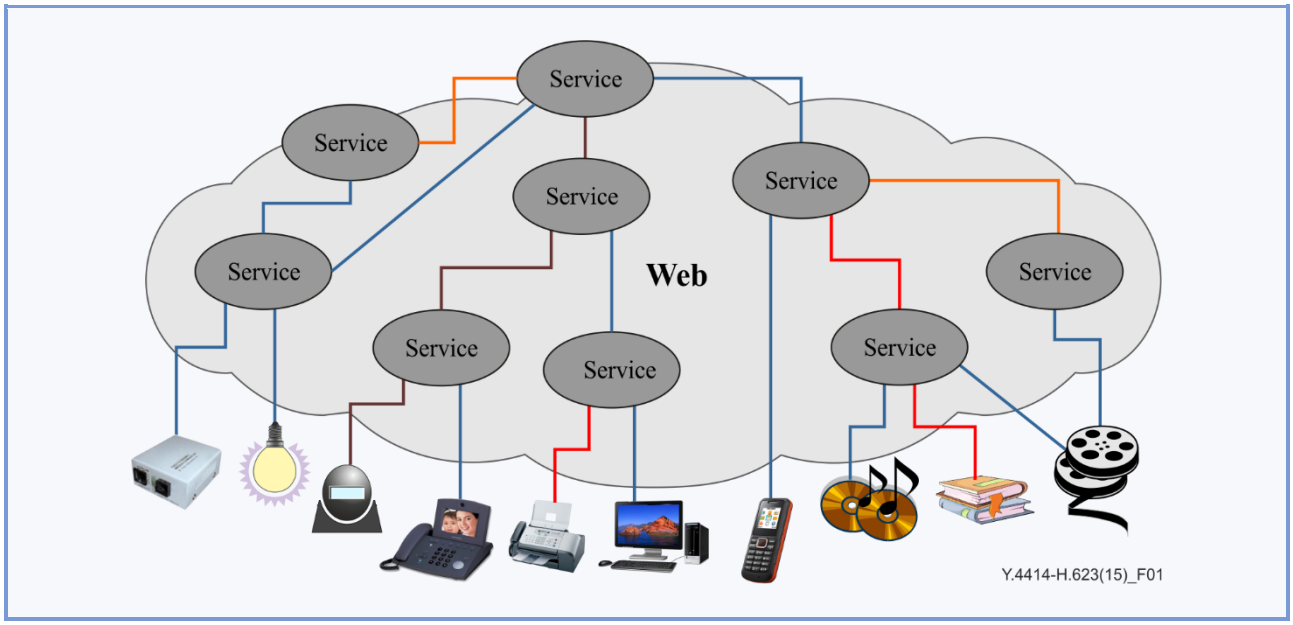


Figure 1 – General concept of a web of things service

7 Functional architecture of the WoT service

Figure 2 shows an overview of the functional architecture of a WoT service.

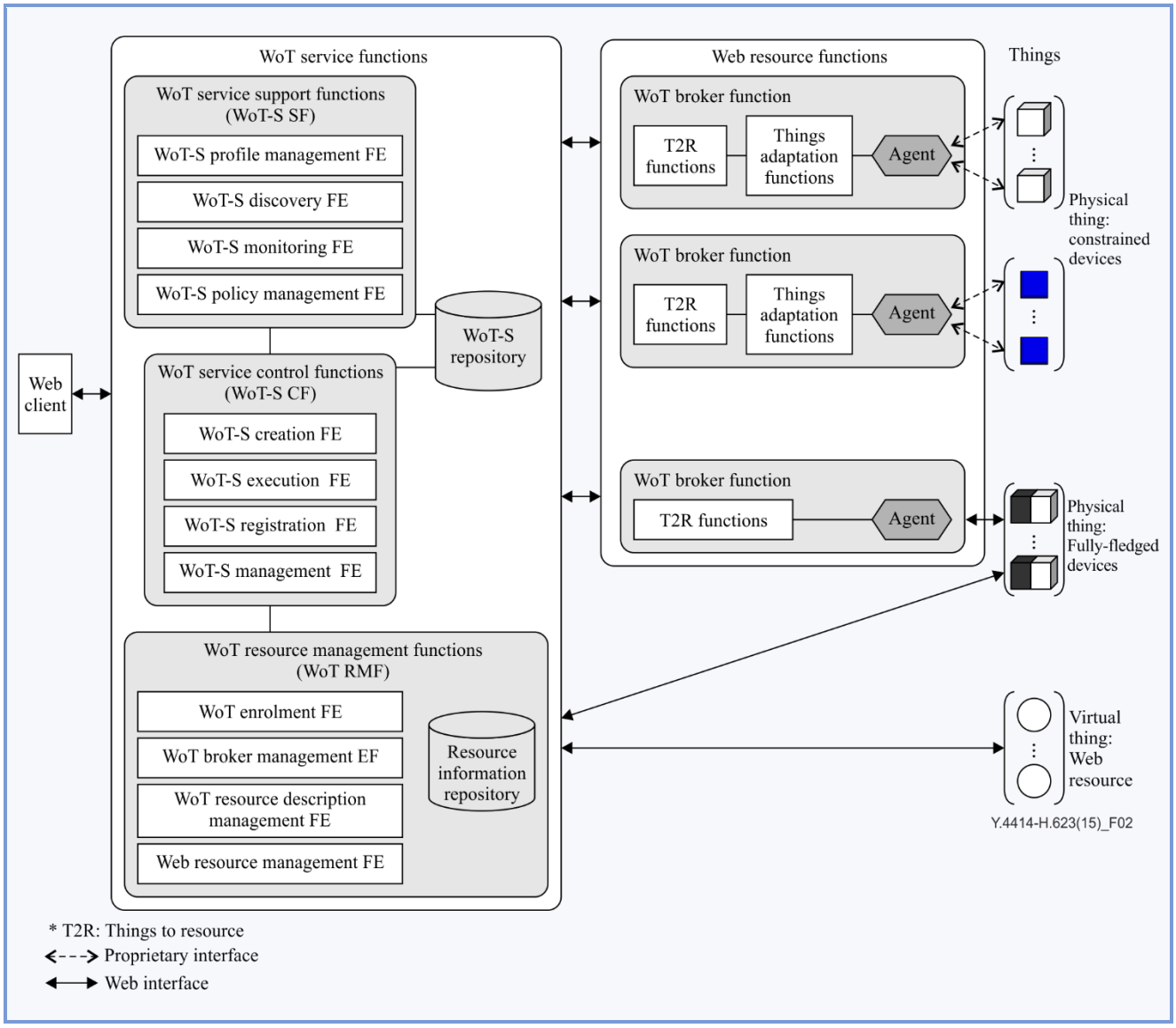


Figure 2 – Functional architecture of WoT service

NOTE – Virtual things exist in the information world and are capable of being stored, processed and accessed. Examples of virtual things include multimedia content and application software [ITU-T Y.4000]. In the case of a WoT service architecture, virtual things exist as the web resources on the web environments.

7.1 WoT service functions

The WoT service functions provide WoT services to WoT clients. They manage overall behaviours related to the WoT service. If a WoT client requests a WoT service, the WoT service functions analyse the request then discover and provide the services for a WoT client. The WoT service functions have three sub-functions as follows:

– **WoT service support functions (WoT-S SF)**: The WoT-S SF manages overall behaviours of the WoT service. The WoT-S SF are responsible for providing service profile management, service discovery, service monitoring, QoS management, access control and policy management of WoT service. And it is also responsible for containing and updating the WoT service information.

– **WoT service control functions (WoT-S CF)**: They control the WoT service which are registered in the WoT service providers. It is responsible for executing and creating the WoT service and mash-up services.
– **WoT resource management functions (WoT RMF)**: They contain resource information (things information) which is used by and registered at the WoT service functions.

7.2 Web resource functions

The web resource functions have responsibility for accessing the things by enabling web technologies. If there are no relevant web technologies to access the things (devices) directly, the WoT broker function should be used which provides access to the devices on the web. Also it is responsible for integrating the web resources (e.g., fully-fledged device and constrained device) to the web [ITU-T Y.4400]. If it is a virtual thing, it can be accessed directly by the WoT resource management function. The T2R function has a role for integrating and exposing things to the web resources. In [ITU-T Y.4400], the service layer of the WoT broker is responsible for the T2R role. And, the things adaptation functions are responsible for adaptation.

7.3 Web client

With the web client, an end user can use WoT services provided by an accessed physical device or virtual thing. The web browser and web applications allow the web client to use WoT services. The web client can control devices by using a web browser and web technologies such as HTTP, RESTful API etc.

7.4 Things

In the WoT service, the things can be classified into two types; physical things and virtual things. The physical things can be classified into two types of devices, constrained devices and fully-fledged devices. The things can be accessed and exposed on the web as a WoT service.

7.4.1 Physical things

7.4.1.1 Constrained device

The constrained device is a non-capable device to connect to the Internet or web alone. And it has no functionality of web. It has just its own proprietary interfaces and protocols such as ZigBee, Bluetooth, NFC, etc. These interfaces and protocols do not have interoperability with web technologies. In this case, the devices need the WoT broker. By exposing on web through the WoT broker, the devices can communication with the WoT broker using their proprietary interface. And the WoT broker has a role of agent for these devices in order to communicate with the web user.

7.4.1.2 Fully-fledged device

The fully-fledged device is a device to support direct web connectivity functionalities, therefore there is no need to translate HTTP requests from the web clients into the appropriate protocol.

However, to enable this capability, the web server is built in the devices itself.

The fully-fledged device is necessary to satisfy the following two capabilities:

– support of TCP/IP protocol suite
– web server.

These capabilities enable devices seamless integrating to the web and their functionality through the web interface.

7.4.2 Virtual things

The virtual things are information world objects such as media, image, video and documents, etc. These objects are web resources exposing through the web service.

8 WoT service functions

8.1 Functional entities of the WoT service support functions (WoT-S SF)

The WoT-S SF manages overall behaviours of the WoT service. The WoT-S SF are responsible for providing service profile management, service discovery, service monitoring, QoS management, access control and policy management of a WoT service. The WoT-S SF consists of four functional entities (FEs); the WoT service profile management FE (WoT-S-PRM FE), the WoT service discovery FE (WoT-S-DIS FE), the WoT service policy management FE (WoT-S-PM FE) and the WoT service monitoring FE (WoT-S MON FE). The WoT-S SF also interworks with the WoT-S repository to register and discover WoT services, etc.

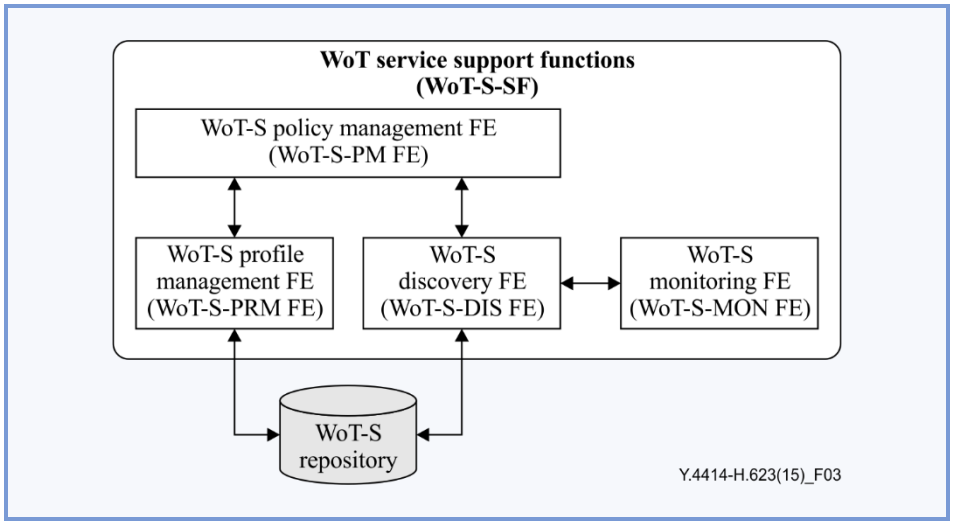


Figure 3 – Functional entities and interfaces of WoT-S SF

8.1.1 WoT service profile management functional entity (WoT-S-PRM FE)

The WoT-S profile management FE (WoT-S-PRM FE) is responsible for managing the profile of the registered services. The service profile is a record that contains information related to the WoT service such as version of service, service life cycle, service classification etc. If a service version is changed, this information is reflected in the registered WoT-S repository. The WoT-S-PRM FE interworks with the WoT-S repository to maintain the latest information of the WoT service.

A WoT user requests a service through the WoT-S-PRM FE and the WoT-S-PRM FE forwards the request to the WoT-S-DIS FE. The WoT-S-DIS FE analyses the request and searches the services in the WoT-S repository or on the web. The WoT-S-DIS FE returns the result to the WoT-S-PRM FE. And the WoT-S-PRM FE provides the service to the WoT user.

8.1.2 WoT service discovery functional entity (WoT-S-DIS FE)

The WoT service discovery FE (WoT-S-DIS FE) provides a discovery capability in collaboration with web resource functions. The WoT-S-DIS FE interworks with the WoT broker functions to discover the WoT service composed of physical devices.

In the WoT service environments, lots of the WoT services will be presented and provided on the web and many users and service developers want to search and find them on the web. In the context of the WoT, the simplest way of enabling search for the WoT service (things) is using the web search engines like the web documents. However, a search for things as the WoT service is more complicated than a search for documents on the web due to its contextual information such as their absolute location or their relative location.

8.1.3 WoT service monitoring functional entity (WoT-S-MON FE)

The WoT service monitoring FE (WoT-S-MON FE) is responsible for monitoring the registered WoT service. The web client or other FEs (WoT-S-DIS FE, WoT-S-PRM FE) utilize the WoT-S-MON FE to find out the status of the WoT service. The status of the WoT service is recommended to include service availability or predict response time, static/dynamic status information and resource information, e.g., CPU/GPU power.

The WoT-S-MON FE is recommended to provide static and dynamic information in order to guarantee a stable quality of the WoT service. Static information consists of quality requirements, e.g., response time, min performance value, max performance value and normal performance value for the WoT service. Dynamic information consists of a real-time updated status for the WoT system and service. For example, system status includes application ID, network port, the amount of instant data and average data, network bandwidth, and so on. Service status includes execution time, fps, screen size, and so on.

The WoT-S-MON FE is required to monitor system resource information of processor power consumption, e.g., CPU and GPU, in order to predict system power consumption of the WoT service. Therefore, the system resource is necessary to provide the QoS guaranteed WoT service such as a high performance with lower power consumption. A GPU resource is measured by using a power coefficient corresponding to each GPU frequency, GPU utilization and GPU static power.

8.1.4 WoT service policy management functional entity (WoT-S-PM FE)

The WoT service policy management FE (WoT-S-PM FE) provides three capabilities for the WoT service as follows:

– QoS information management: The WoT-S-PM FE is responsible for managing QoS information about the registered WoT service. It checks accessibility, performance, reliability of the WoT service. The WoT-S-PM FE is also responsible for providing QoS information to the web client.

– Service access control: The WoT-S-PM FE provides service access control capabilities to control the accessibility of a specific service by an application. It provides authentication and authorization actions required to ensure that the web client has appropriate access rights to the requested service.

– Security and safety management: The WoT-S-PM FE provides control capabilities regarding security and safety in order to support a trustworthy WoT service. It provides methods to guarantee the safe execution of service in the WoT system, which are comprised of actions by detecting an execution request and controlling execution of the service command.

8.2 Functional entities of the WoT service control functions (WoT-S CF)

The WoT-S CF is responsible for controlling the WoT service creation, service execution, service registration and service management. The WoT-S-CF consists of four functional entities (FEs); the WoT-S management FE (WoT-S-MA FE), the WoT service execution FE (WoT-S-EX FE), the WoT-S creation FE (WoT-S-CR FE), the WoT service registration FE (WoT-S-RE FE) with WoT-S repository.

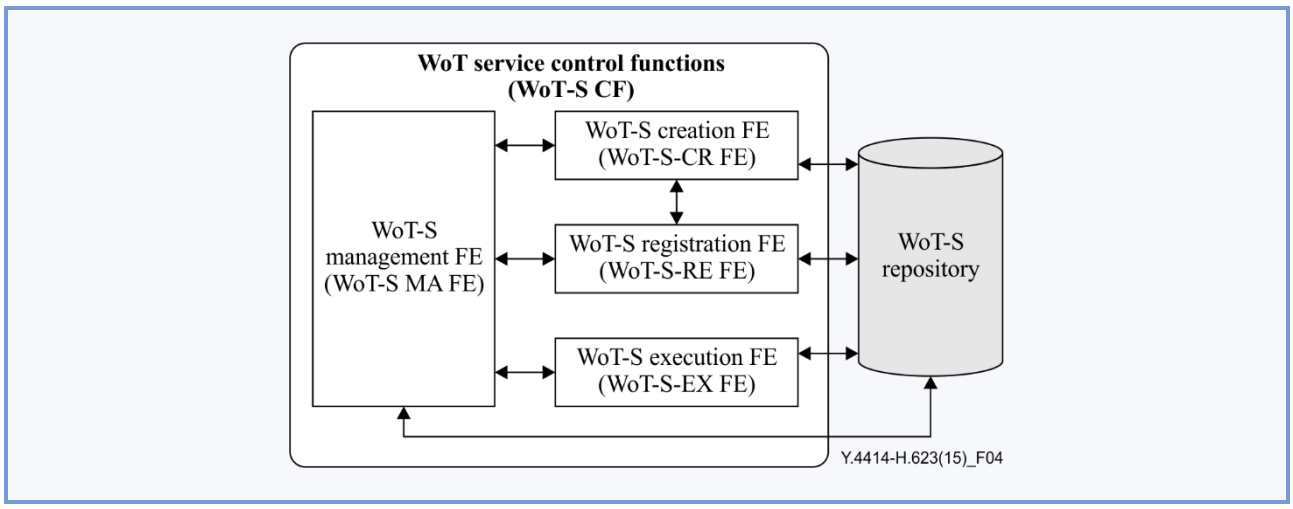


Figure 4 – Functional entities and interfaces of WoT-S CF

8.2.1 WoT service execution functional entity (WoT-S-EX FE)

The WoT service execution FE (WoT-S-EX FE) is responsible for executing the WoT service. The WoT-S-EX FE enables the WoT service user to access and execute the WoT service. When an end user (web client) wants to use the WoT service, the WoT-S-EX FE interacts with the WoT-S-PM FE located in the WoT-S SF to check the authentication and the authorization of the requester. Additionally, the WoT-S-PRM FE also checks availability of the service at that time.

The WoT-S-EX FE interworks with the WoT-S repository which includes logics to execute a service. The WoT execution logics have a method for how to operate services.

The basic functionalities of the WoT-S-EX FE on managing logic are:

– storing and management logics describing an event in the WoT-S repository, a condition to be satisfied by the event, and an action to be executed when the conditions are satisfied;

  NOTE – When a service requester requests to register the logic, the logic is stored if and only if the logic does not generate a static conflict in which the logic describes different actions upon the same event and the same conditions for the action with other logics.

– retrieving a logic describing the event storage when an event occurs;

– inspecting the conditions whether or not the event satisfies a condition described in the retrieved logic;

– executing an action described in the retrieved logic when the event satisfies the condition.

8.2.2 WoT service creation functional entity (WoT-S-CR FE)

The WoT service creation FE (WoT-S-CR FE) allows the WoT service to be developed by simple composition of the existing WoT services on top of web-based things.

The WoT-S-CR FE is responsible for managing and executing mash-up WoT services. The WoT-S-CR FE provides two functions: mash-up logic management and mash-up engine.

A WoT user is able to search, create or update service execution logic through mash-up logic management via the web access. The service execution logic is stored and retrieved using the WoT-S repository. A WoT user can also request a mash-up engine to operate the service execution logic for the mash-up WoT service. Each unit of the WoT service for the mash-up service is executed by the WoT-S-EX FE.

NOTE – The web mash-up is defined as "web applications generated by combining content, presentation or application functionality from the disparate web sources". The aim of mash-up is to create new useful applications or services by combining them in a value-adding manner. The WoT allows developers to access and search for the web-enabled things, and also allows owners of the web-enabled things to have a simple and scalable mechanism on sharing them.

8.2.3 WoT service management functional entity (WoT-S-MA FE)

The WoT service management FE (WoT-S-MA FE) is responsible for managing the WoT service. It provides capabilities to manage service updating, service tracking, update management, auditing, version control, logging and service coordination.

The WoT-S-MA FE supports the collection and storage of the WoT service logs. It also provides capabilities to replace or substitute the WoT service. If the WoT-S-MA FE finds better services, it recommends to change the old service to the new service. The WoT-S-MA has the ability to detect failures of services as well as recovery from failures.

8.2.4 WoT service registration functional entity (WoT-S-RE FE)

The WoT service registration FE (WoT-S-RE FE) allows registration and deregistration of the WoT service in the WoT service repository. The WoT-S-RE FE has the ability to analyse services for registration in terms of their characteristics (e.g., service category, service provider and information about service charging).

The WoT-S-RE FE interworks with the WoT service creation FE (WoT-S-CR FE). The WoT-S-RE FE registers the new service after creating a service by the WoT-S-CR FE. The WoT-S-RE FE investigates the service and extracts information of the service.

8.2.5 WoT service repository (WoT-S repository)

The WoT service repository is responsible for storing WoT services and related information such as service characteristics, service type, service location, service version, service status, etc. The WoT service repository also contains the service execution logic, which comprises the service execution description of each WoT service. Mash-up services also have the service execution description which contains service information, service rule and service profiles like other WoT services:

– service information is related to the linked information with the WoT-S repository or in case of mash-up services, it contains key features for WoT mash-up services such as classification, QoS, owner, etc.;

– service rule is related to the method on how to operate services, or in case of mash-up services, it contains one or mixed rules of sequential execution, conditional execution and operational execution;

– service profiles are related to the parameters to provide customised services according to location, preference or other clients' characteristics.

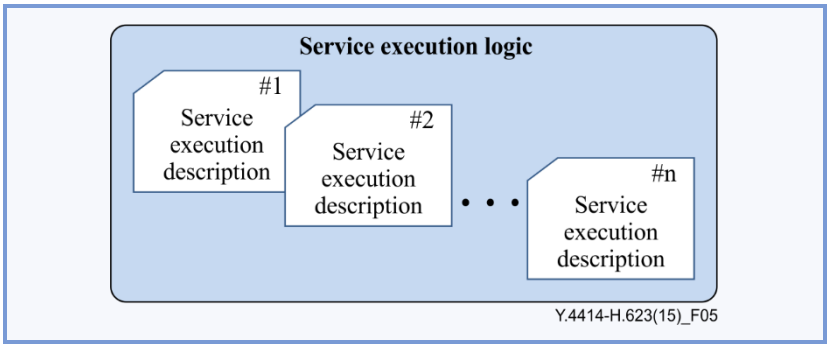


Figure 5 – Service execution logic in a WoT-S repository

8.3 Functional entities of the WoT resource management functions (WoT RMF)

The WoT RMF manage resources that are used for the WoT service. The WoT RMF can register resources into a resource information repository and can delete them. The resources are provided by the WoT broker functions and the web. The WoT RMF consists of four functional entities; the WoT enrolment FE (WoT-EN FE), the WoT broker management FE (WoT-BM FE), the web resource management FE (Web-RM FE) and the WoT description management FE (WoT-DM FE).

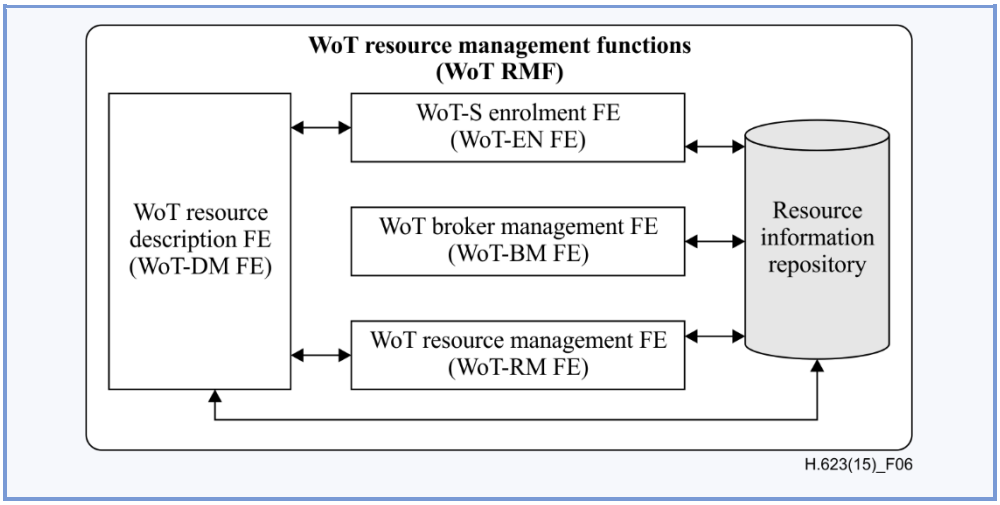


Figure 6 – Functional entities and interfaces of WoT RMF

8.3.1 WoT broker management functional entity (WoT-BM FE)

The WoT-BM FE is responsible for registering information of each WoT broker function with related information such as location, owner, their services, etc.

The WoT broker functions are responsible for providing physical devices as the web resources. The WoT broker functions reside in the WoT resource support functions area. However, there can be a large number of brokers and each WoT broker provides a number of different web resources. Therefore, in order to provide efficient WoT service management of the WoT a broker is necessary.

8.3.2 Web resource management functional entity (Web-RM FE)

The Web-RM FE is responsible for managing the web resource with the resource information repository.

There are many types of content on the web such as image, video, music, documents, weather information, and text as a virtual things. These virtual things are ready to be exposed on the web and the WoT service provider uses them to create new WoT services. The Web-RM FE interworks with the resource information repository to manage these virtual things.

8.3.3 WoT enrolment functional entity (WoT-EN FE)

The WoT-EN FE provides a capability to support the WoT service enrolment, e.g., submission and registration. The web resource functions submit information of the web-enabled things to the WoT-EN FE. The WoT-EN FE checks and filters them using the enrolment policy.

The WoT-EN FE also manages the enrolment policy. The WoT-EN FE updates and deletes the enrolment policy according to the changes on the information and context of the WoT service. The following are functions of the WoT-EN FE:

– Submission handling: The WoT-EN FE receives the enrolment request from the WoT enabled things. For the request, these WoT services can submit their information such as IP, descriptions, name, URI, capability, related services, etc. Then the WoT-EN FE allows or blocks the request based on the enrolment policy.

– Enrolment policy: The enrolment policy contains the rules of allowance for the web-enabled things. The WoT-EN FE manages the enrolment policy according to the changes of related entities such as the WoT service, service providers, users, etc.
– Enrolment management: The WoT-EN FE has the capability for registration and management of the submitted web-enabled things to the resource information repository. The changes on the information and the context of the registered web-enabled things induce change of information of the resource information repository. The WoT-EN FE re-examines the changed web-enabled things with enrolment policy rules. It means that the WoT-EN FE can remove the registered web-enabled things beforehand from the resource information repository. The WoT-EN FE can allow the blocked WoT service to register in the repository if those are acceptable based on the changed enrolment policy.

8.3.4 WoT description management functional entity (WoT-DM FE)

The WoT-DM FE provides a description method and it manages all description information related to things. Also, it provides capabilities for translating descriptions among the WoT services.

There are many kinds of things that exist and all things need a mechanism to describe themselves and their services to be (automatically) discovered and used on the web. There are many description methods to utilize describing things such as web service description language (WSDL), resource description framework in attributes (RDFa), microformat, device profile for web services (DPWS) metadata, SensorML, JSON, etc. The WoT-DM FE provides the capability of description translation to cooperate among the many description methods and models supported by the different service providers.

9 Web resource functions

9.1 WoT broker functions

The WoT broker functions have a role for integrating and exposing the devices to the web. To expose and use physical devices on the web, a web server supporting the HTTP is implemented on the devices. However, every device does not satisfy these conditions because of constraints.

The WoT broker functions support communication between users of the WoT (e.g., the web clients, applications) and fully-fledged devices as well as constrained devices. The WoT broker functions enable the seamless integration of a device onto the web. The agent of the WoT broker has a role to control and communicate with physical devices.

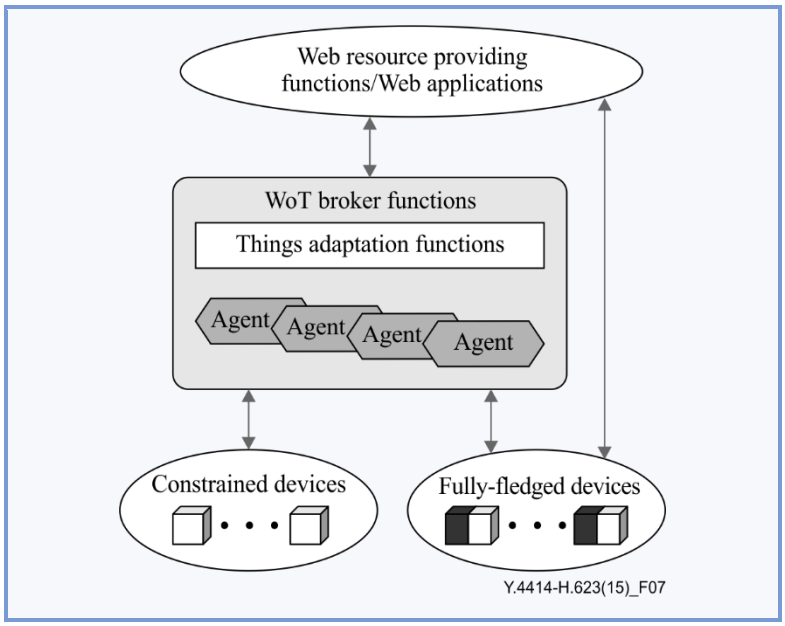


Figure 7 – Overview of WoT broker function architecture

NOTE – Detailed functions and architecture of the WoT broker function are given in [ITU-T Y.4400].

There are several ways to interact between a WoT broker and devices such as http-based connection and web services:

– Web services: enable business entities and applications to intercommunicate openly with each other over a network. Web service systems have program-language-independent properties, use message-driven communication, and are easily bound to different transport protocols [ITU-T Y.2232]. The basic web services for WoT define an interaction [b-W3C SOAP 0] [b-W3C SOAP 1] between WoT broker and constraint/fully-fledged device as an exchange of messages.

– In the web services, a service and services descriptions are described [b-W3C WSDL 0]. A service is enabled by a software module deployed on network-accessible platforms provided by the service provider. The service description contains the details of the interface and implementation of the service including the data types, operations, binding information and network location. The service description is supported by a web services provider.

– Also, in order for an application to take advantage of web services, three behaviours must take place: the publication of service descriptions, the finding and retrieval of service descriptions, and the binding or invoking of services based on the service description. These behaviours can occur singly or iteratively, with any cardinality between the roles. In detail, these operations are the execution (binding or invoking) of services based on the service description, finding and retrieval of service descriptions and publication of service descriptions.

The WoT broker is a web services provider who provides a capability for use to create web services. The web client and web service application function are the web services requester who use the services provided by a WSP.

9.2 RESTful web services

The RESTful web services have some efficient properties such as high scalability achieved by the result of a loosely coupled design and facility for deployment since it builds up on the web infrastructure and standards.

The REST includes the primacy of resources, identified using URIs, and a uniform interface generally implemented by the HTTP protocol. Resources are manipulated through representations portrayed according to a media type (e.g., HTML, Atom, etc.) and some metadata.

The WoT uses HTTP as an application protocol rather than using it as a transport protocol. The WoT exposes its interface following the REST principle because these technical characteristics enable services and applications on the WoT to be built as loosely coupled systems which expose the application programming interface (API) in a uniform manner.

Appendix I

WoT description model

(This appendix does not form an integral part of this Recommendation.)

This appendix provides the following WoT description models.

I.1 Physical characteristics in WoT description

I.1.1 Device description

The device description provides information about things itself and its manufacturers. As a result, browsers, search engines and applications discovery by browsing the product description will be able to render its UI and visualization in an enhanced manner. Device description may have the following general components:

– unique ID
– name
– brand
– manufacturer
– description
– device picture
– tags and
– authoritative information URL.

I.1.2 Location description

The location description provides the physical location information of things. The user and application get the current latitude and longitude of the things. Also they get address, postal code, country name, street information etc.

I.1.3 Owner description

The owner description provides the current owner of things.

I.2 Service characteristics in WoT description

I.2.1 Quality of service description

The quality of service description covers parameters such as bandwidth, up-time, average response time it helps taking the right decision. In a WoT populated by billions of things quality of service information can be of great help to choose the right things for the right application. These data can be based on monitoring services or provided by the manufacturer of device and service provider.

I.2.2 Types of services description

The type of services description covers basic information required to describe things on the web. It contains a description of services that a thing offers (e.g., sensor information, multimedia service, etc.).

Appendix II

An example of information flows

(This appendix does not form an integral part of this Recommendation.)

This appendix describes information flows for providing WoT services. The information flow shows an example of a use case of how to provide a WoT service or how to access things.

II.1 Information flow between web client and things

Figure II.1 shows one of the information flows relating to using WoT services. In this case, the web client wants to get a WoT service which consists of several kinds of things. At first, the web client sends a request for the WoT service to the WoT service functions using web technology (e.g., HTTP). And then the WoT service functions can access things through interworking with WoT broker functions. The detailed operation with information flow is followed as below.

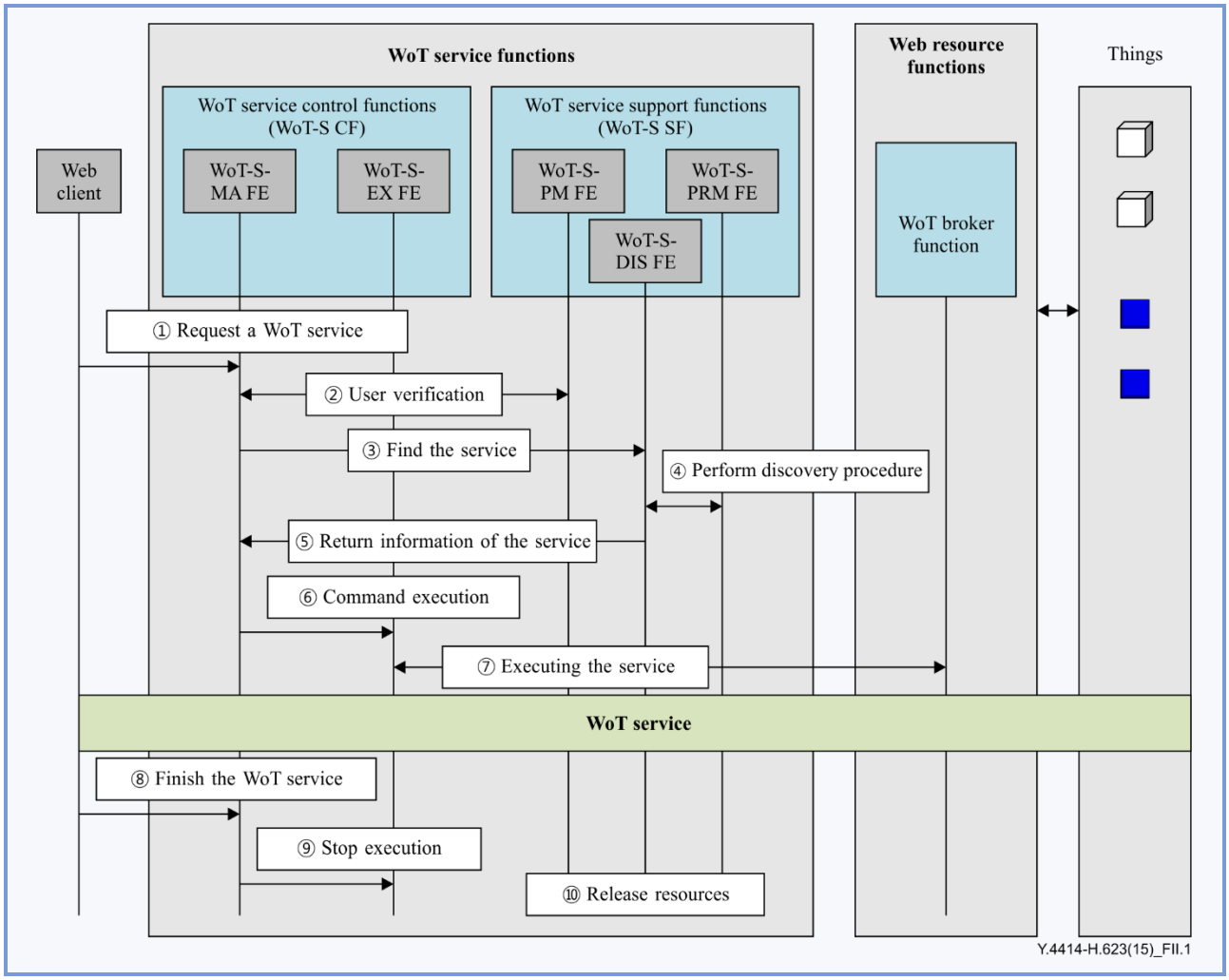


Figure II.1 – Information flow between end user and things

1) Web client requests a WoT service to the WoT-S-MA FE.

2) The WoT-S-MA FE (WoT service management FE) performs user verification with the WoT-S-PM FE (WoT service policy management FE). The WoT-S-MA FE sends a message to the WoT-S-PM FE to verify the user. The WoT-S-PM FE checks user validation and authentication to see whether the user is authorized to access the service. If the user does not have authorization, the WoT-S-PM FE returns a failure message.

3) When the authentication procedure is finished successfully, the WoT-S-MA FE sends a message to the WoT-S-DIS FE to discover the service.

4) The WoT-S-DIS FE performs a search procedure with the WoT-S-PRM FE (WoT service profile management FE). The WoT-S-PRM FE searches for the WoT service and retrieves information from the WoT-S repository.

5) When the search procedure is finished successfully, the WoT-S-DIS FE returns information of the WoT service to the WoT-S-MA FE.

6) The WoT-S-MA FE requests executing the WoT service to the WoT-S-EX FE (WoT service execution FE).

7) The WoT-S-EX FE executes the service, interworking between the WoT-S-EX FE and the WoT broker function, to use/access things.

8) The web client informs the WoT-S-MA FE to finish the WoT service.

9) The WoT-S-MA FE commands the WoT-S-EX FE to stop executing the WoT service.

10) The WoT-S-EX FE releases resources.

Bibliography

[b-IETF RFC 3986] IETF RFC 3986 (2005), *Uniform Resource Identifiers (URI): Generic Syntax.*
https://tools.ietf.org/html/rfc3986

[b-W3C digloss] Glossary of Terms for Device Independence (2005)
http://www.w3.org/TR/di-gloss/

[b-W3C SOAP 0] W3C Recommendation (2007), *SOAP Version 1.2 Part 0: Primer (Second Edition).*
http://www.w3.org/TR/soap12-part0/

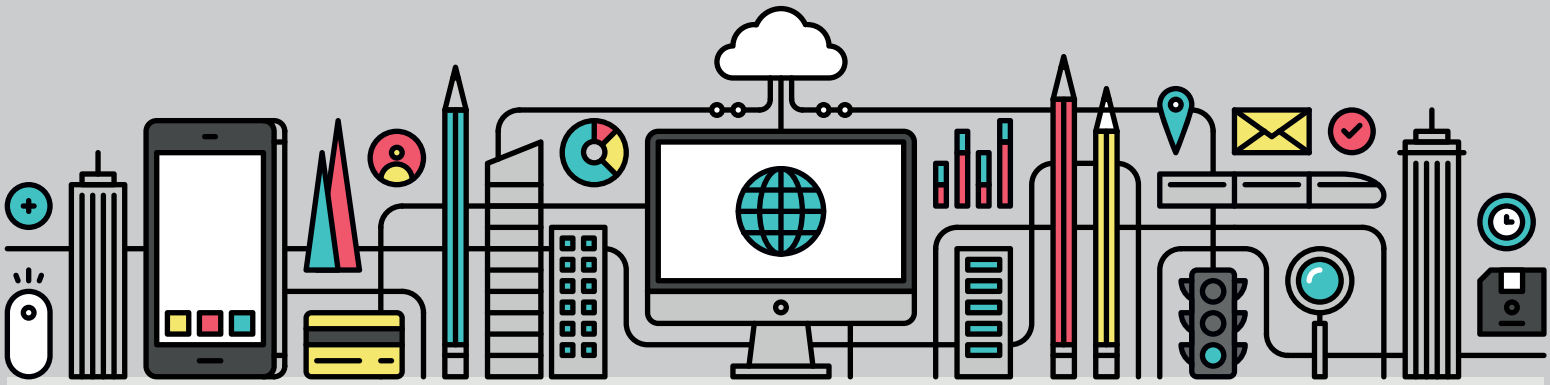
[b-W3C SOAP 1] W3C Recommendation (2007), *SOAP Version 1.2 Part 1: Messaging Framework (Second Edition).*
http://www.w3.org/TR/soap12-part1/

[b-W3C WACterms] W3C et (1999) – *Web Characterization Terminology & Definitions Sheet*.
http://www.w3.org/1999/05/WCA-terms/

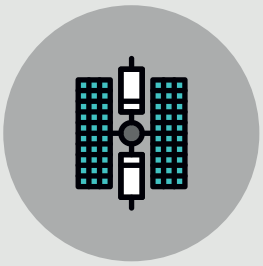
[b-W3C webarch] Jacobs, I & Walsh N (2004), *Architecture of the World Wide Web*, Volume One, 2004.
http://www.w3.org/TR/webarch/

[b-W3C WSDL 1] W3C Recommendation (2007), *Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language.*
http://www.w3.org/TR/wsdl20/

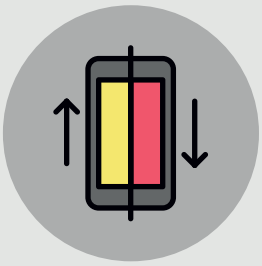
Internet Of Things



Big Data



Communication



Sync

Data Flow


12:00 PM
wellness
100%
FARMING
Wellness # Eat Healthy
Stay Fit # Take Care of Yourself

Y.4450/Y.2238
Overview of Smart Farming based on networks

Overview of Smart Farming based on networks

Summary

Recommendation ITU-T Y.2238 considers the actualized convergence service for agriculture, namely Smart Farming, as a solution to cope with various problems caused by severe conditions or the gap of viewpoints between people engaged in farming and IT engineers. In particular, this Recommendation defines service capabilities for Smart Farming, provides a reference model for Smart Farming, and identifies network capabilities required to produce an infrastructure which supports Smart Farming.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T Y.2238	2015-06-13	13	11.1002/1000/12520

Keywords

Convergence service for agriculture, Smart Farming.

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

Table of Contents

Introduction

An actualized convergence service for agriculture is expected to bring more efficiency and quality improvement in production, distribution and consumption of agricultural products with the aid of IT information processing and autonomous control technologies.

However, there exist many difficulties to establish services and systems to actualize the convergence service in the agricultural field to cope with various problems such as time-varying weather changes, growth condition of farm products, and continual diseases or technical problems such as battery life and sensor malfunctions due to severe conditions. In addition, the gap of viewpoints between people engaged in farming and IT engineers may make it difficult to accomplish this mission.

Therefore, it is appropriate to consider an actualized convergence service for agriculture, namely Smart Farming, as a solution to cope with anticipated problems. In addition, the aspect of network capabilities to support this convergence service, where various types of networks, such as NGN, Future Networks and legacy networks, could be applied should be considered. This Recommendation develops a reference model, defines service capabilities and identifies network capabilities required to support such services.

Recommendation ITU-T Y.4450/Y.2238

Overview of Smart Farming based on networks

1 Scope

This Recommendation provides an overview of Smart Farming based on networks.

The scope of this Recommendation includes:

– Smart Farming reference model.

– Service capabilities required by Smart Farming.

– Network capabilities required by Smart Farming.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T Y.2002] Recommendation ITU-T Y.2002 (2009), *Overview of ubiquitous networking and its support in NGN*.

[ITU-T Y.2060] Recommendation ITU-T Y.2060 (2012), *Overview of the Internet of things*.

[ITU-T Y.2701] Recommendation ITU-T Y.2701 (2007), *Security requirements for NGN release 1*.

[ITU-T Y.3041] Recommendation ITU-T Y.3041 (2013), *Smart ubiquitous networks – Overview*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 context [ITU-T Y.2002]: The information that can be used to characterize the environment of a user.

NOTE – Context information may include where the user is, what resources (devices, access points, noise level, bandwidth, etc.) are near the user, at what time the user is moving, interaction history between person and objects, etc. According to specific applications, context information can be updated.

3.1.2 object [ITU-T Y.2002]: An intrinsic representation of an entity that is described at an appropriate level of abstraction in terms of its attributes and functions.

NOTE 1 – An object is characterized by its behaviour. An object is distinct from any other object. An object interacts with its environment including other objects at its interaction points. An object is informally said to perform functions and offer services (an object which makes a function available is said to offer a service). For modelling purposes, these functions and services are specified in terms of the behaviour of the object and of its interfaces. An object can perform more than one function. A function can be performed by the cooperation of several objects.

NOTE 2 – Objects include terminal devices (e.g., used by a person to access the network such as mobile phones, personal computers, etc.), remote monitoring devices (e.g., cameras, sensors), information devices (e.g., content delivery server), products, contents, and resources.

3.1.3 ubiquitous networking [ITU-T Y.2002]: The ability for persons and/or devices to access services and communicate while minimizing technical restrictions regarding where, when and how these services are accessed, in the context of the service(s) subscribed to.

NOTE – Although technical restrictions to access services and communicate may be minimized, other constraints such as regulatory, national, provider and environmental constraints may impose further restrictions.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 agricultural consumer: The service role that ultimately purchases the final agricultural products from distributors or agricultural producers.

3.2.2 agricultural distributor: The service role that distributes agricultural products supplied from agricultural producers through the distribution channel.

3.2.3 agricultural producer: The service role that actually produces agricultural products to be supplied to distributors or consumers.

3.2.4 Smart Farming based on networks: A service that uses networks to actualize a convergence service in the agricultural field to attain more efficiency and quality improvement and to cope with various problems.

NOTE – Problems may include such items as time-varying weather changes, growth condition of farm products, plant diseases, and technical problems, such as battery life and sensor malfunctions due to severe conditions. The service may overcome such problems with the aid of IT information processing and autonomous control technologies.

3.2.5 Smart Farming service provider: The service role that provides the requested Smart Farming services, such as providing a portal or consulting based on data gathered from agricultural fields, to requesting users.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

AAA Authentication, Authorization and Accounting
API Application Programming Interface
CCTV Closed-Circuit Television
DC Distribution Centre
GC Gathering Centre
IoT Internet of Things
IPv6 Internet Protocol, version 6
ISDN Integrated Services Digital Network
IT Information Technology
NGN Next Generation Network
PSTN Public Switched Telephone Network
QoS Quality of Service
RFID Radio Frequency Identification

5 Conventions

None.

6 Introduction of Smart Farming based on networks

6.1 Concept

Smart Farming is a service that uses networks to actualize a convergence service in the agricultural field to cope with various problems, e.g., time-varying weather changes, growth condition of farm products, and continual diseases or technical problems, such as battery life, sensor malfunctions due to severe conditions, with the aid of information processing and autonomous control technologies of the information technology (IT) area.

Smart Farming, based on networks, needs to be considered on the basis of interactions between the entities that are tightly related to the agricultural field, i.e., agricultural producers, service providers, logistics agents, market distributors, customers and the telecommunications network that interconnects these, as shown in Figure 1.

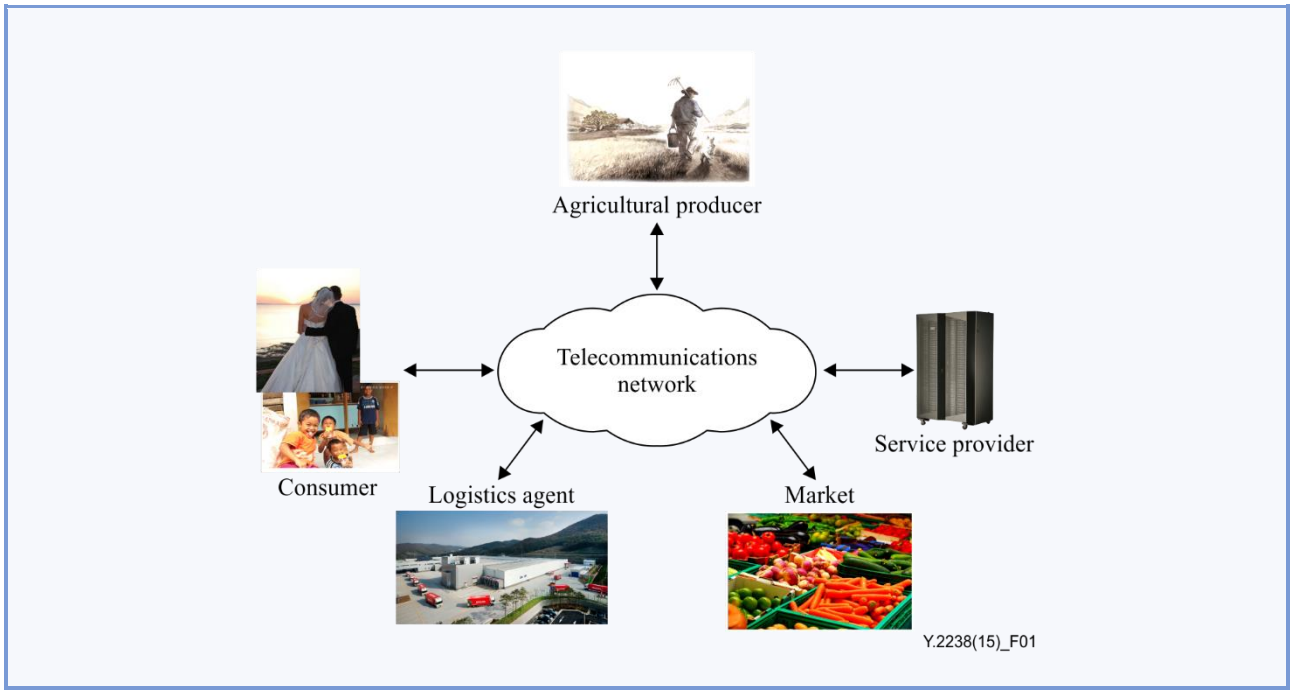


Figure 1 – Conceptual diagram of Smart Farming based on networks

6.2 General overview

Smart Farming can run autonomously without human intervention with the cyclic procedures of Appendix I considering environments as shown in Appendix II when advanced technologies such as sensors, computers or control systems are used. However, Smart Farming must also be capable of coping with unexpected events, such as product theft and reduction in revenues due to excess production. Hence, there is a need for capabilities to address these issues. Services, such as the following, which use networks, could be developed to address such problems in the future environment:

– Farm products protection: a service that prevents the theft of agricultural products in farmlands, greenhouses or warehouses by humans and animals through surveillance, such as closed-circuit television (CCTV), infrared sensors or other sensors that are connected via networks to agricultural producers.

– Farm products traceability: a service that provides traceability information about farm products in the market to customers, including identity information about agricultural producers, food safety certification, etc.
– Remote farm management: a service that enables agricultural producers to monitor and control farm conditions on remote sites through devices such as smartphones or other terminals connected to networks.
– Farm production regulation: a service that provides information to aid agricultural producers in deciding which farm products to sow, when to sow, when to harvest, and applicable farm production regulations.

7 Reference model of Smart Farming based on networks

7.1 Reference architecture

To apply Smart Farming based on networks shown in Figure 1, a reference architecture showing the service roles, consumers, distributors, agricultural producers and service providers, is presented in Figure 2.

Service roles shown in Figure 2 (consumers, agricultural producers, distributors, service providers and network providers) play major roles in Smart Farming based on networks as follows:

– Consumer: the service role that ultimately purchases the final agricultural product from distributors, agricultural producers or direct sellers and also provides a farm product traceability service to the service provider.
– Agricultural producer: the service role that actually produces agricultural products to be supplied to distributors or consumers that is provided with Smart Farming services, such as a farm product protection service or remote farm management service.
– Distributor: the service role that distributes agricultural products supplied from agricultural producers through the distribution network and provides a farm production regulation service, which is required to identify and maintain the break-even point.
– Service provider: the service role that provides the requested Smart Farming services to requesting users such as consumers, agricultural producers and distributors.
– Network provider: the service role that provides the infrastructure that conveys information related to Smart Farming based on networks and interconnects the other service roles.

These service roles play their part according to the stage at which they are positioned, i.e., pre-production stage, production stage or post-production stage. In the pre-production and production stages, the agricultural producer can be advised on what to produce, when to produce, what to seed and other matters by the service provider via plan/production consulting. Distributors and consumers can also be advised by the service provider about market demands, food traceability and prices. The network provider can provide the telecommunication network to support information transfer between the various service roles.

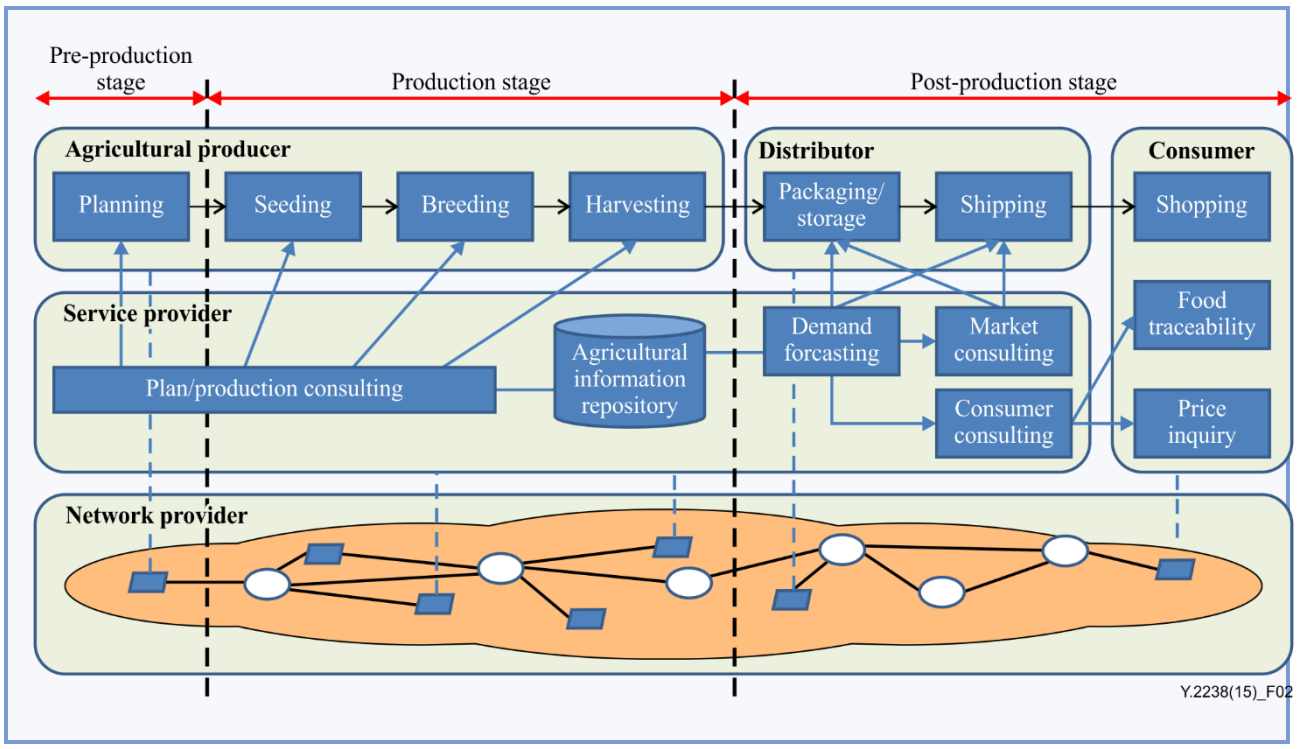


Figure 2 – Reference model of Smart Farming based on networks

7.2 Service roles

There can be various types of service roles, each with differing capabilities. Categorization of each of the identified service roles facilitates identification of appropriate service capabilities in this Recommendation. For this purpose, the service roles in the Smart Farming service are categorized in Table 1.

Table 1 – Service roles

Domain	Actor	Role
Agricultural producer	Outdoors producer	Agricultural producer who grows crops outdoors.
	Greenhouse producer	Agricultural producer who grows crops in greenhouses.
	Plant factory operator	Agricultural producer who operates a plant factory and grows crops in it.
Distributor	Direct seller	Seller who grows and sells crops.
	Wholesale/retail distributor	Business operator who gathers, selects, packages and/or processes agricultural products to sell wholesale/retail.
	On-line seller	Business operator who runs an on-line market site and trades and sells agricultural products to those who access the on-line site.
Service provider	Service business operator	Business operator who makes a profit by providing a portal to consumers through the network.
	Content business operator	Information provider, e.g., agriculture consultant, who makes a profit by providing plant growth aid information based on data, gathered from sensors and videos in greenhouses or plant factories.

Table 1 – Service roles

Domain	Actor	Role
Network provider	Network business operator	Business operator who makes a profit by providing networks to other service roles for agricultural products information delivery.
Consumer	General consumer	Consumer who buys agricultural products for personal use.
	Business consumer	Consumer who buys agricultural products for business purposes, such as restaurants and hotels.
	Group consumer	Consumer who buys agricultural products for inter-group members' purposes.

The service roles could interact for the purpose of conducting business. Actors may assume more than one role and through the use of services may attain profits. It is also possible to access dedicated services through these interactions.

8 Service capabilities required to support Smart Farming

There are a number of tasks that must be accomplished during the various implementation stages associated with Smart Farming in Figure 2. In the pre-production stage, the agricultural producer performs planning based on the service provider's plan/production consulting using connections provided by the network provider. In the production stage, the agricultural producer performs seeding, breeding and harvesting based on the service provider's production consulting through the connections provided by the network provider. The agricultural information repository is required to support these consulting tasks of the service provider. In the post-production stage, the distributor performs packaging/storage and shipping based on the service provider's market consulting and demand forecasting through the connections provided by the network provider. The consumer also performs shopping with food traceability and price inquiry based on consulting through the connections provided by the network provider.

These tasks could be defined in the context of service capabilities. The detailed service capability information is provided in Appendix III. It is recognized that, while Appendix III may not provide an all-inclusive list of the many tasks that must be undertaken during the various stages of implementing Smart Farming, it does allow for sufficient information to be obtained to develop a set of required network capabilities that are identified in clause 9.

9 Network capabilities

The high-level network capabilities for the support of Smart Farming are as follows:

– Connecting to anything capabilities: Theses capabilities refer to the support of the different ubiquitous networking communication types (person-to-person communication, person-to-object communication and object-to-object communication) and include the support of tag-based devices (e.g., radio frequency identification, RFID) and sensor devices. Identification, naming and addressing capabilities are essential for supporting "connecting to anything".

– Open web-based service environment capabilities: Emerging ubiquitous services/applications will be provided based upon an open web-based service environment, as well as on legacy telecommunication and broadcasting services. In particular, application programming interfaces (APIs) and web with dynamics and interactivities will be supported. Such a web-based service environment will allow not only the creation of retail community-type services, but also the building of an open service platform environment which third-party application developers can access to launch their own applications. Using interactive, collaborative and customizable features, the web can provide rich user experiences and new business opportunities for the provision of ubiquitous networking services and applications.
– Context-awareness and seamlessness capabilities: Context-aware means the ability to detect changes in the status of objects. Intelligence systems associated with this capability can help to provide the best service, which meets the situation using user and environmental status recognition. Seamlessness is a key capability for "5 Any" (anytime, anywhere, any-service, any-network and any-object).
– Multi-networking capabilities: A transport stratum needs multi-networking capabilities in order to simultaneously support unicast/multicast, multi-homing and multi-path. Because of high traffic volume and the number of receivers, ubiquitous networking requires multicast transport capability for resource efficiency. Multi-homing enables the device to be always best connected using multiple network interfaces including different fixed/mobile access technologies. These capabilities can improve network reliability and guarantee continuous connectivity with desirable quality of service (QoS) through redundancy and fault tolerance.
– End-to-end connectivity over interconnected networks: For Smart Farming, it is critical to develop a solution to provide end-to-end connectivity to all objects over interconnected heterogeneous networks such as next generation networks (NGNs), other IP-based networks, broadcasting networks, mobile/wireless networks and public switched telephone network/integrated services digital networks (PSTN/ISDNs). Internet Protocol, version 6 (IPv6), with its large address space, can be considered a good candidate for providing globally unique addresses to objects [ITU-T Y.3041].
– Networking capabilities: Provide relevant control functions of network connectivity, such as the Internet of Things (IoT) and transport resource control functions, mobility management or authentication, authorization and accounting (AAA) [b-ITU-T Y Suppl. 3].
– The device capabilities include, but are not limited to: Direct interaction with the communication network: Devices are able to gather and upload information directly (i.e., without using gateway capabilities) to the communication network and can directly receive information (e.g., commands) from the communication network [ITU-T Y.2060].

10 Security considerations

This Recommendation is recognized as an enhancement of IP-based networks. Thus, it is assumed that security considerations in general are based on the security of IP-based networks and thus it is required to follow the security considerations identified by clauses 7 and 8 of [ITU-T Y.2701].

Appendix I

The cyclic procedures of a convergence service for agriculture

(This appendix does not form an integral part of this Recommendation.)

An actualized convergence service for agriculture is expected to bring more efficiency and quality improvement in the cyclic procedures for production, distribution and consumption of agricultural products with the aid of information processing and autonomous control technologies of the IT area as shown in Figure I.1.

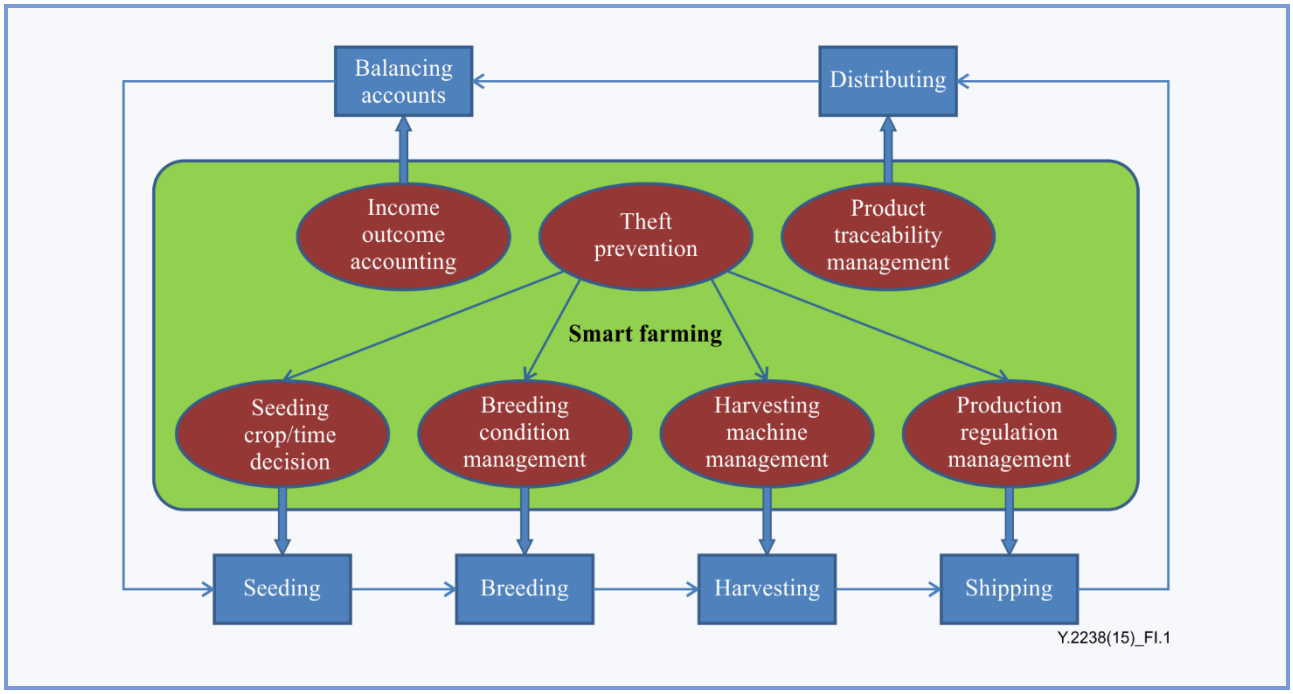


Figure I.1 – The cyclic procedures of a convergence service for agriculture

Appendix II

Environments and deployments of a convergence service for agriculture

(This appendix does not form an integral part of this Recommendation.)

Considering the deployments of a convergence service for outdoor farming, there could exist two types of environment, i.e., real-time facilities environments and crop growth environments. The former is related to administration networks mainly consisting of indoor actuators connected with telecommunication networks, while the latter is related to monitoring networks mainly consisting of sensors connected with telecommunication networks. Figure II.1 shows the aspect of environments and deployments of the convergence service for outdoor farming.

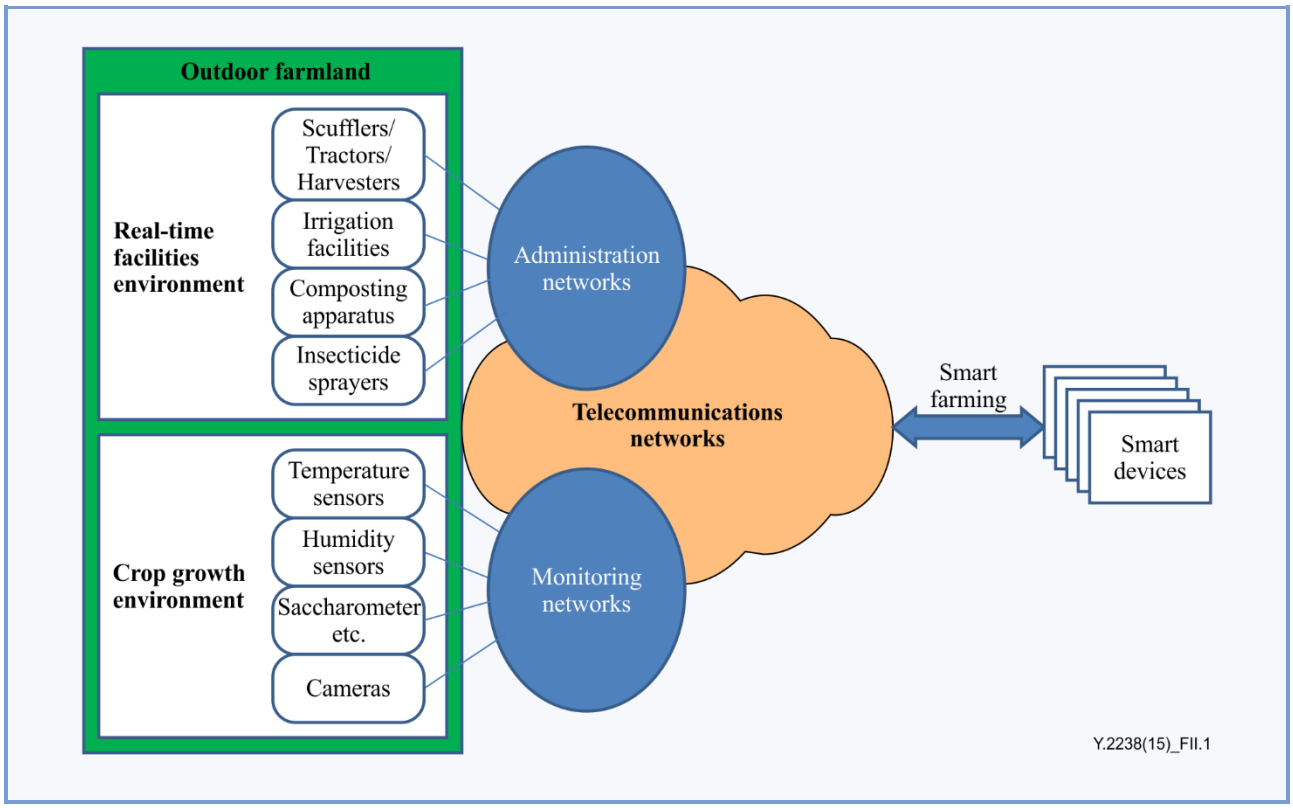


Figure II.1 – Environments and deployments of a convergence service for outdoor farming

Considering the deployments of a convergence service for greenhouse/plant factory farming, there could exist two types of environment, i.e., real-time facilities environments and crop growth environments. The former is related to administration networks mainly consisting of indoor actuators connected with telecommunication networks, while the latter is related to monitoring networks mainly consisting of sensors connected with telecommunication networks. Figure II.2 shows the aspect of environments and deployments of the convergence service for greenhouse/plant factory farming.

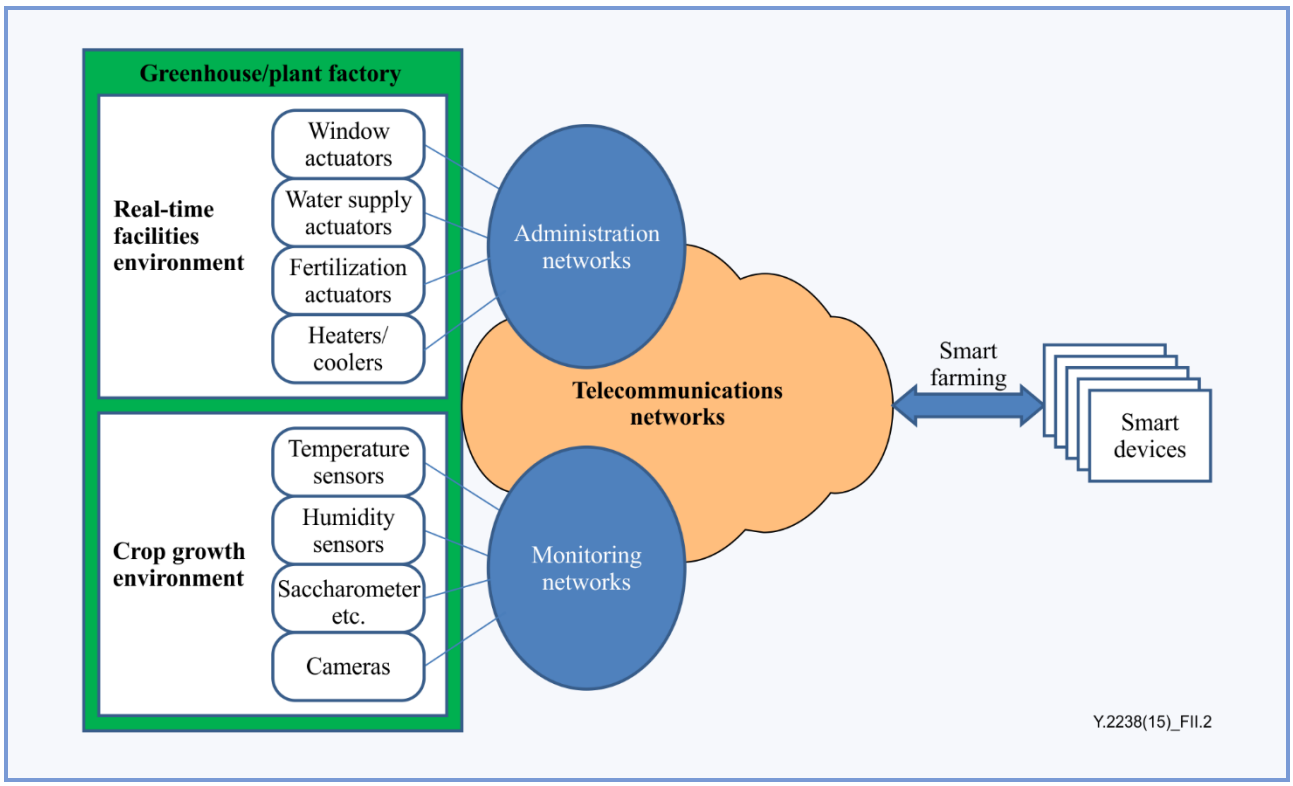


Figure II.2 – Environments and deployments of a convergence service for greenhouse/plant factory farming

Appendix III

Service capabilities

(This appendix does not form an integral part of this Recommendation.)

In this appendix, attention focuses on tasks that must be accomplished during the various implementation stages associated with Smart Farming. It is recognized that this is not an all-inclusive list of the many tasks which must be undertaken during the various stages of implementing Smart Farming.

III.1 Service capabilities for the pre-production stage

In the pre-production stage, the following service capabilities are required to facilitate a convergence service for agriculture:

– Making a business plan: A business plan should consider sales/production planning, generate profit and loss calculations of the overall business production enterprise, address marketing and management issues, and understand business interactions with personal communication communities;

– Operational review: While an operational review is required to ensure stable revenues, it is also required to provide a global review on licenses, permits, regulations, patents, trademarks, risk management, environmental issues, production quality, timeline, etc.;

– Role of IT in Smart Farming: The role of IT in Smart Farming should be determined, e.g., how to apply IT for such items as sensing and monitoring systems, network infrastructure, controlling systems, information data bases and farming management servers to support Smart Farming;

– Business review: For the stable management of the Smart Farming business, it is appropriate to consider: the size, type and quality of facilities and facility equipment; real estate; farming equipment; ownership structure; etc.

– Financial estimate: For maintaining the Smart Farming business and attaining more profits, it is very important to estimate projected cash flow, income statement and balance sheet. It is also necessary to estimate a projected statement of expenses and profitability;

– Decision making: This is the process used to make decisions on produce and product selection and amounts, development of a rationalized price policy and marketing policy at all levels of marketing, etc.;

– Marketing plan: Before the production phase, it would be appropriate to consider market trends, customer service, marketing contracts, strategic partners, pricing, promotion, distribution, target markets, competitive advantage, etc. in a marketing plan to produce a better decision;

– Understanding customers: For better understanding, it is required to identify customer class for primary products and anticipating customer's changing characteristics.

III.2 Service capabilities for the production stage

Service capabilities for the production stage need to be described for Smart Farming to attain effective crop production. For this purpose, optimal environmental control methods that take into consideration energy use and automation efficiency are required for more effective crop production. The optimal growth condition is different as it depends on crop species, types and varieties of plants that are grown in greenhouses and plant factories.

The control system at the production stage could define the major components in applying IT technologies in Smart Farming and specify the requirements and the architecture for technological

issues. The system could collect information for the growth management of crops and control the facilities promoting optimal growth environments in greenhouses. Such a system covers the growth environment management, the growth environment controls, etc. This is specified to make an artificial cultural environment for crops with culture fluid, carbon dioxide concentration, temperature, humidity, wind and light. Smart farming enables generation of such a system, which provides an automatic and continuous production of agricultural produce regardless of the place and the season, and control over the greenhouse and plant factory (indoor) or outdoor fields by management of sensor networks.

It is also required to have defined interfaces for Sensor Nodes and Control Gateways as the control system for such operations between the Sensor Nodes. A Smart Farming. Control Gateway needs to interact with inside and outside sensors. The technical requirements for sensor nodes and gateways must allow for operation in appropriate environments, e.g., it must cope with external weather conditions, internal air quality and various soil environments. The following should be defined and characterized:

– Operating environment: Sensors and sensor node configuration, communications, environments and operation management;
– Message passing model: Message passing models between a Smart Farming control gateway and sensor nodes should be developed;
– Message format: General frame structure and configuration of each field;
– Message flow: Initialization for sensor nodes, sensor data monitoring (passive, active and event-based), the sensor node status and reporting of information obtained.

It is also necessary to define the relationship among environmental variables, e.g., temperature, relative humidity, controlling actuators and application technologies for Smart Farming. It should contribute to stability and high efficiency of plant production in plant factories and greenhouses by providing indicators for environmental control. The convergence service for greenhouse and plant factory farming can be divided into several phases as follows:

– Monitoring changes of facility environments in real-time;
– Offering the most suitable information on the quality of crops by analysing synthetically the observation of crop status, growth conditions and environmental information;
– Organic controlling facility environments according to the growth level of crops and any changes of environments.

Considering these phases, Smart Farming can be provided to users based on environments related to real-time facilities and crop growth monitoring.

Required features in greenhouses and plant factories, which consist of devices related to energy, harvesting devices, light sources, environmental controls for a greenhouse and a plant factory interior, and a wide area of interfaces for the related control information are as follows:

– Interface among sensors, controllers and energy components;
– Information and communication interfaces among plant factories and greenhouses;
– Component monitoring specification for sensing in plant factories and greenhouses;
– Control and monitoring specification for actuators in plant factories and greenhouses;
– Energy monitoring and control specification in plant factories and greenhouses;
– Service structure and service specification standards in plant factories and greenhouses.

III.3 Service capabilities for the post-production stage

Service at the post-production stage is required for product distribution flow from producers to consumers. In the product flow, farmers, vendors/agents, wholesalers, rural retailers and suppliers, and logistics are involved. At all levels, information flow and produce management is essential to maintain

the product's quality throughout the product distribution flow as shown in Figure III.1. The flow of input products from producers to consumers should be described in detail and the constraints in each sub-process should be identified to develop appropriate solutions for the post-production stage.

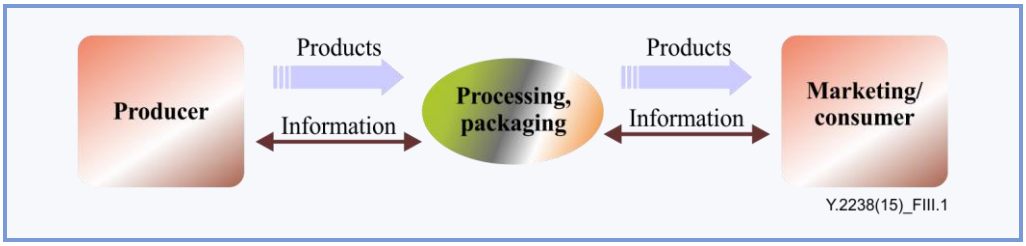


Figure III.1 – Product flow between producers and consumers

It is important to note that lack of packaging facilities may be one of the constraints in the product flow control of small-scale farmers during the transition from subsistence to commercial farming. Significant post-harvest losses occur when especially vulnerable crops and fruits are subjected to mechanical damage. Therefore, management of packaging and distribution should be taken into consideration in the development of product flow control systems during the post-production stage.

III.3.1 Product distribution flow at the post-production stage

Integrated post-production networks can be developed by forming clusters of producers and determining the optimum gathering centres (GCs) linking goods producers, distributors, and consumers/retailers enabling coordinated distribution of local goods products and facilitating the integration of food distribution in the local products supply systems into large scale product distribution channels as shown in Figure III.1. Detailed gathering and distribution routes can be analysed using product flow control functions. It can be noted that coordinating and integrating the product flow control activities of local product delivery systems may reduce the number of routes, transport distances and transport times. Such post-production network integration can produce positive improvements towards potential markets, product flow efficiency, environmental issues and traceability of product quality and product origins.

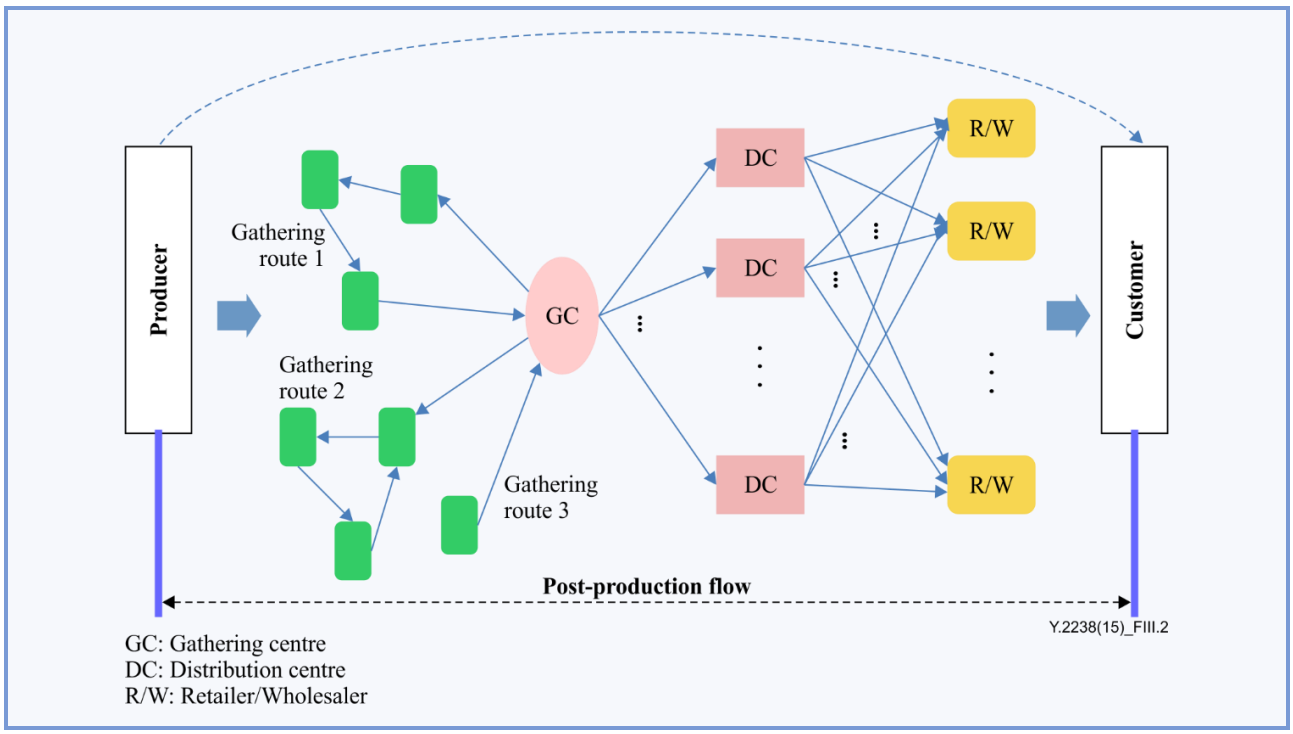


Figure III.2 – Product distribution flow at the post-production stage

The dotted line in Figure III.2 indicates the case of direct delivery from producer to customers. Coordination and network integration in the local product supply chain increases distribution efficiency, potential markets and access to information while reducing environmental impact. Forming the best gathering and distribution centres (DCs) for locally produced product can be very important. Such location decisions can be supported technically, since the location decisions may have dynamic results over time. Therefore, in the process of developing improved product flow control in the local product supply chain, it can prove essential to try detailed location analysis to map and cluster producers, to determine the optimum location of gathering and distribution centres, to analyse routes to optimize them for product gathering and distribution, and to simulate route distance and delivery time.

III.3.2 Product flow control at the post-production stage

Generally, product flow control at the post-production stage consists of the components shown in Figure III.3, to work collaboratively to meet the consumers' demand. The utilization of product flow control at the post-production stage can provide a meeting point among availability, facilitate factors which can be realized through products. In setting up product flow control, all factors involved in the supply chain process of a farming business can be used as a reference. In other words, a product flow control system can translate a simple object to a very complex system in the operational decisions. Policies on product flow control system utilization at the post-production stage are expected to provide benefits in the farming management commodity supply chain, from production, storage and distribution, to the wholesaler and finally consumer level.

In principle, the structure of a product flow control system can accommodate two important decisions from both the producer's and the consumer's point of view. From the producer's side, it is important to consider how products can be made available and well distributed, and from the consumer's side, how, where and when they can obtain a good quality product.

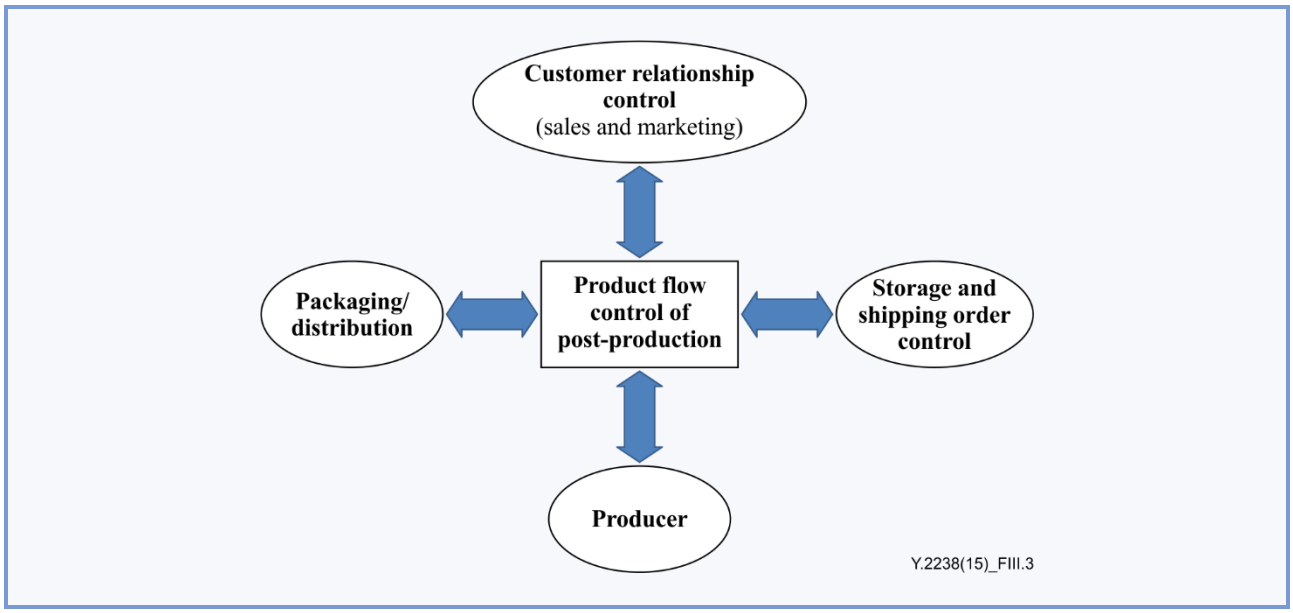


Figure III.3 – Conceptual product flow control for post-production stage

Conceptual product flows in Figure III.3 describe interaction at the post-production stage and how it works. Efficient management of the product flow is required for production planning, physical gathering of primary products from fields, processing and storage at various levels, handling, packaging and distribution of final products. In the product distribution flow, various types of customers such as farmers, vendor/agents, wholesalers, rural retailers and suppliers and transporters are involved.

The activities required for product flow control function can consist of four parts: producers' activity for production quality, markets' activity for sales and marketing, customers' activity for purchasing and product flow control for coordination. The physical infrastructure can consist of a series of IT network infrastructures to connect each location, i.e., producers, wholesalers, retailers, markets and consumers. Communication networks connect central offices with geographically separated branch offices of customers or markets. From the perspective of effective product flow control, an integrated approach for farming is required for the effective control of product hazards that is a shared responsibility of producers, packers, processors, distributors, retailers, food service operators and consumers. Therefore, tracking products from seed sowing to harvesting and shipment is becoming an area of focus.

In the product distribution flow, there is potential for control related to improvements in terms of reducing transport routes, distances and times; reducing emissions from vehicles; improving the packaging of food products; and improving transport services for deliveries from wholesalers/retailers to consumers.

Bibliography

[b-ITU-T Y Suppl. 3] ITU-T Y-series Recommendations – Supplement 3 (2008), *ITU-T Y.2000 series – Supplement on service scenarios for convergence services in a multiple network and application service provider environment*.

6 Services, Applications, Computation and Data Processing

Y.4551/F.771

Service description and requirements for multimedia information access triggered by tag-based identification

Service description and requirements for multimedia information access triggered by tag-based identification

Summary

Recommendation ITU-T F.771 specifies a high-level functional model, a service description and requirements for multimedia information access triggered by tag-based identification. The scope of this Recommendation is limited to those applications and services that have both multimedia and tag-based characteristics.

Editorial note – This document includes in clean text the changes introduced by Amd. 1 (2014).

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T F.771	2008-08-06	16	11.1002/1000/9465
1.1	ITU-T F.771 (2008) Amd. 1	2014-10-14	16	11.1002/1000/12230

Keywords

Multimedia information access, requirements, tag-based identification.

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

Table of Contents

Introduction

Multimedia information access triggered by tag-based identification is a class of the generic multimedia services identified in Recommendation ITU-T F.700. Following the methodology given in Recommendation ITU-T F.701, this Recommendation specifies the functional model, service description and requirements for multimedia information access triggered by tag-based identification.

When a user's device obtains an identifier from an identification (ID) tag wherever it is attached, the device tries to find the location of associated multimedia information automatically. Each identifier is stored in an ID tag such as a barcode, a passive/active radio frequency identification (RFID), or a smart card, so as to be recognized automatically by various types of tag readers. This feature enables the user to refer to the multimedia content without typing its address on a keyboard or inputting the name of objects about which relevant information is to be retrieved. This information is stored in databases somewhere in the network and provided by several service providers.

Recommendation ITU-T Y.4554/F.771

Service description and requirements for multimedia information access triggered by tag-based identification

1 Scope

This Recommendation specifies the service description and the requirements for multimedia information access triggered by tag-based identification. This service enables users to access multimedia information through users' electronic devices equipped with ID tag readers and communication functions.

The multimedia information is comprised of voice, sound, text, graphic, video and other media which have various applications such as digital maps for route-finding and interactive three-dimensional panoramic pictures. Users will receive its delivery via communication networks such as fixed and mobile networks according to their network access capabilities.

These applications and services are characterized by the use of the following:

1) Identifier: An identifier is assigned to each real-world entity such as a physical/logical object, a place, or a person.

2) ID tag: An ID tag is a tag, such as a barcode, a passive/active RFID, a smartcard, or an infrared tag, used to store the identifier.

3) ID terminal: An ID terminal is a device equipped with an ID tag reader/writer used to capture the identifier. Capturing the identifier triggers access to multimedia information via a network.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T F.700] Recommendation ITU-T F.700 (2000), *Framework Recommendation for multimedia services*.

[ITU-T F.701] Recommendation ITU-T F.701 (2000), *Guideline Recommendation for identifying multimedia service requirements*.

3 Definitions

3.1 Terms defined elsewhere

None.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms.

3.2.1 ID resolution: ID resolution is a function to resolve an identifier into associated information. In this Recommendation, it is specifically a function to resolve an identifier on/in an ID tag into necessary access information such as uniform resource locator (URL), Internet Protocol

(IP) address, and phone number, which may indicate a protocol and/or a pointer to access multimedia information services provided by multimedia information delivery functions.

NOTE – Since the ID resolution function handles mapping relationships between identifier and access information, it may have a database for the efficient management of mapping record.

3.2.2 ID tag: An ID tag is a tiny physical object which stores a small amount of information which is an identifier or includes an identifier with other additional application data such as name, title, price and address. An ID tag is attached to or associated with a real-world entity to carry the information or attributes about the entity. In this Recommendation, an ID tag is used to store an identifier of the real-world entity with optional application data. Examples are radio frequency identification (RFIDs), barcodes, 2D barcodes, infrared tags, active radio frequency (RF) tags, etc.

3.2.3 ID terminal: An ID terminal is a device with a capability to capture data from ID tags and write data into ID tags, and other capabilities such as communication capability and multimedia information presentation capability. The data capture capability may include a function to obtain an identifier from ID tags even with no communication capability such as barcodes and two-dimensional barcodes. Examples of equipment that use data capture techniques are digital cameras, optical scanners, radio frequency (RF) transponders, infrared data association (IrDA), galvanic wire-line, etc. Sometimes, an ID terminal is called a user terminal. An ID terminal can optionally have multiple capture devices. An ID terminal can optionally have a capability to communicate with multiple RF types of ID tags. An ID terminal should have a frequency band selector in case of multiple band reader/writer.

NOTE – As described in clause 6.1.2.1 of ITU-T H.621 Amd.1, the ID terminal may be composed of multiple ID tag Readers/Writers. For example, a camera for 1-dimentional and/or 2-dimentional barcode reading and an RFID reader/writer for RFID tag reading and writing may be equipped together in an ID terminal. But also, multiple RFID readers/writers may be equipped to support different frequency bands such as HF and UHF. Otherwise, a single RFID reader/writer may support HF and UHF bands and such an RFID reader/writer is called a dual band RFID reader/writer, or dual band RFID reader, for short.

3.2.4 identifier: An identifier is a series of digits, characters and symbols or any other form of data used to identify a real-world entity. It is used to represent the relationship between the real-world entity and its information/attributes in computers. This relationship enables users to access the information/attributes of the entity stored in computers via users' ID terminals.

3.2.5 multimedia information: Multimedia information is digital information that uses multiple forms of information content and information processing, such as text, pictures, audio, video, three-dimensional panoramic pictures and digital maps, which informs or entertains users.

3.2.6 multimedia information delivery function: A multimedia information delivery function is a function to deliver multimedia information to an ID terminal which is triggered by the tag-based identification.

3.2.7 real-world entity: A real-world entity is a physical and logical entity which mainly acts or is used in the real world, such as a physical object, logical object, place and person. Examples of *physical objects* include water bottle, book, desk, wall, chair, tree, animal, cloth, food, television, light and so on. Examples of *logical objects* include digital content such as video, movie, music and story. Examples of *places* include room, corridor, road, gate, garden and so on. The real-world entity concept includes both networked entities and non-networked entities.

3.2.8 tag-based identification: Tag-based identification is the process of specifically identifying a real-world entity by capturing its identifier from an ID tag storing the identifier. This identification process consists of two steps. The first step is to read an identifier from an ID tag. The second step is to resolve the identifier into associated information such as a uniform resource locator (URL), an Internet Protocol (IP) address, a telephone number, an e-mail address, and/or a

content address for audio/video data. For tag-based identification, the identifier, ID tag and ID terminal are mandatory elements.

NOTE – Resolved results indicate the identification of a real-world entity. Thus, the identification process selects proper information from multiple associated information with the identifier according to the condition of applications, services and system implementations. For example, in the identification for telephone applications, a person's identifier may be resolved into a telephone number. On the other hand, in the identification for multimedia messaging applications, the identifier may be resolved into an e-mail address.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

2D	Two Dimensional
3D	Three Dimensional
B2B	Business to Business
B2C	Business to Consumer
DVD	Digital Versatile Disk
G2C	Government to Consumer
HF	High Frequency
ID	Identification
IP	Internet Protocol
IR	Infrared
IrDA	Infrared Data Association
MAC	Media Access Control
NGN	Next Generation Network
PDA	Personal Digital Assistant
QoS	Quality of Service
RF	Radio Frequency
UHF	Ultra High Frequency
RFID	Radio Frequency Identification
URL	Uniform Resource Locator
Wi-Fi	Wireless Fidelity

5 Conventions

In this Recommendation:

– The expression "**is required to**" indicates a requirement which must be strictly followed and from which no deviation is permitted if conformance to this Recommendation is to be claimed.

– The expression "**is recommended**" indicates a requirement which is recommended but which is not absolutely required. Thus this requirement need not be present to claim conformance.

6 High-level functional model and generic service description

The objective of this clause is to describe the high-level functional model of the multimedia information access triggered by tag-based identification. First, this clause describes the high-level functional model consisting of multiple elementary functional components and the relationships between them. Next, a generic service is described with the work process of this model.

6.1 High-level functional model

Figure 1 shows the high-level functional model consisting of high-level functional components and the relationships between them.

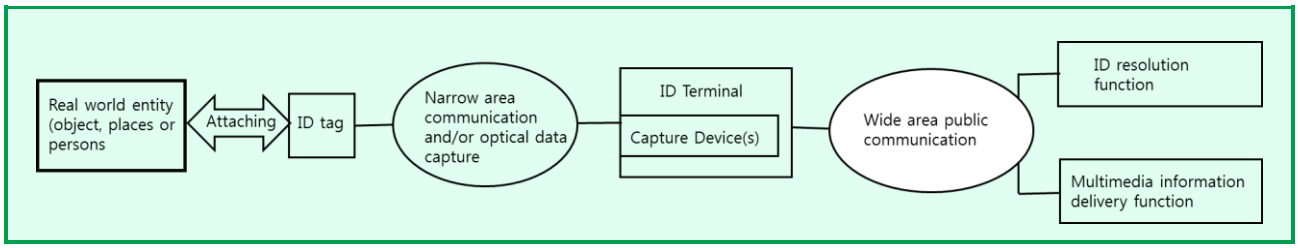


Figure 1 – High-level functional model of the multimedia information access triggered by tag-based identification

6.1.1 ID tag

ID tag is defined in clause 3.2.2.

6.1.2 ID terminal

ID terminal is defined in clause 3.2.3.

6.1.3 ID resolution function

ID resolution function provides the function of ID resolution defined in clause 3.2.1.

6.1.4 Multimedia information delivery function

Multimedia information delivery function provides the function of multimedia information delivery defined in clause 3.2.6.

6.1.5 Narrow area communication

Narrow area communication means that an ID terminal and an ID tag have to be located close together, within a few metres, and the ID terminal captures an identifier written in the ID tag in a wireless manner. In case of RFID, this is the contactless communication between the RFID tag and a reader/writer in the ID terminal. In case of infrared tag, it is infrared communication between the infrared tag and the ID terminal.

6.1.6 Optical data capture

When the ID tag is a printed tag, such as barcode or two-dimensional barcode, optical data capture is used to obtain an identifier from the ID tag.

6.1.7 Wide area public communication

Wide area public communication means public communication to support data exchanges among the ID terminal, multimedia information delivery function and ID resolution function. Examples are the Internet, a mobile telephone network and NGN.

6.2 Generic service description

The service defined in this Recommendation is to enable users to access multimedia information relating to real-world entities such as objects, places and persons. Appendix I describes eight examples of the multimedia information access service triggered by tag-based identification. This service is realized by the following three operations:

1) Tag-based identifier-reading process;

2) ID resolution process; and

3) Information presentation process.

6.2.1 Tag-based identifier-reading process

The tag-based identifier-reading process enables the ID terminal to start identifying real-world entities by using the identifier stored in the ID tag. When this identifier is read, it is transferred to the ID terminal via narrow area communication. Hence, this identifier-reading process works only when the ID tag and ID terminal are in a narrow area together.

6.2.2 ID resolution process

In an ID resolution process, an ID terminal asks an ID resolution function (see clause 3.2.1) to resolve an identifier and receives a result from the ID resolution function. This process enables the ID terminal to find necessary information for accessing the multimedia information delivery service via wide area public communications. Examples of this information are URLs for a web-based information service, a content ID for a digital multimedia retrieval service, a phone number for a voice service, etc.

The simplest relationship between the identifier and multimedia information access may be a static "one-to-one" relationship. However, this relationship may generally be a "one-to-many" type, and a proper service is selected according to the context of users and environments.

For example, an RFID tag is attached to a signboard of a restaurant, and then a user puts the ID terminal on the RFID tag. The menu of the restaurant is shown in the user's preferred language, which is pre-registered in the user profile in the ID terminal. In this example, multiple information content is associated with a single identifier, and the ID resolution function selects the most suitable content among them according to the user's profile.

6.2.3 Information presentation process

The information presentation process enables the ID terminal to access a multimedia information function via wide area public communications, such as voice guidance at a museum, video clip delivery for advertisement, pedestrian navigation using a digital map, restaurant menu, and pedestrian navigation for persons with visual disability. This information presentation process is classified into the following three process types.

6.2.3.1 Information download process

The information download process enables the ID terminal to retrieve multimedia information from multimedia information delivery functions. In this process type, the information may be stored in a server as: fixed data such as in video-on-demand systems; dynamically generated data from elements in the server; or real-time streaming data such as from broadcasting or a networked camera.

6.2.3.2 Information upload process

The information upload process enables the ID terminal to upload information into multimedia information delivery functions. In this process type, information uploaded may be temporarily stored in the ID terminal, manually input by the user, or generated in real-time, such as live data from a video camera in the ID terminal.

6.2.3.3 Bidirectional information process (both download and upload)

The bidirectional information process is a combination of the above two types of multimedia information presentation processes. The ID terminal and the multimedia information delivery function exchange multimedia information with each other. A simple example is a video-conference system. In this case, the ID terminal and the video-conference server exchange audio/video information with each other, after the ID terminal has read all participants' identifiers and resolved the video conference service.

7 Requirements

This clause provides a list of basic requirements for multimedia information access triggered by tag-based identification. The multimedia information access service triggered by tag-based identification has the following requirements in terms of user, service, identifier, ID tag, ID terminal, ID resolution, multimedia information delivery, wide area public communication, security and quality of services (QoS).

7.1 User requirement

USR-001: This service is required to be usable by a wide range of people including children, the elderly and the physically challenged.

7.2 Service requirements

SVC-001: This service is required to deliver multimedia information to the ID terminal, and is recommended to collect multimedia information from the ID terminal such as user profiles and operation logs.

SVC-002: This service is required to deliver multimedia information stored in prepared multimedia data files, and is recommended to deliver multimedia information encoded in real-time (real-time streaming option).

SVC-003: It is required that both push- and pull-type delivery services be supported.

Push-type delivery is useful for advertisement services, warning message delivery, etc. Pull-type delivery is useful for services where the user initiates the exchange of information. The former service will be realized by IR tags and RF tags, and the latter by passive RFIDs, barcodes, etc.

7.3 Identifier requirements

ID-001: Identifier is recommended to be used by different applications.

If an identifier is assigned to a product, the identifier could be used for production management in a factory, supply chain management, and customer service for end users of the product. If a location identifier is assigned to a room in a building, the identifier could be used for a room reservation system, maintenance activity for the room, or a guidance system in case of emergencies. As described in these examples, an identifier, if required, could be used by different applications.

ID-002: Identifier is required to be assigned for real-world entities such as physical/logical objects, persons and places.

Identifier is required to identify products, foods, drugs, digital content, locations, persons, etc. This requirement has an impact on the design of the identifier scheme because the total number of these entities is estimated to be very large. Hence, the total number of identifiers used by the multimedia information access triggered by tag-based identification is also estimated to be much larger than the number of identifiers for communication entities such as IP addresses, phone numbers and MAC addresses. However, today, it is difficult to estimate accurately the total number of identifiers used in multimedia information access triggered by tag-based identification.

ID-003: Identifier is required to be issuable by any organization, such as businesses, non-profit organizations, governments and individual users.

ID-004: Identifier is required to be globally unique so that the multimedia information access triggered by the identifier is to be globally available.

ID-005: Multiple identifier schemes are required to be supported. It is important to provide a function to use multiple identifier schemes in this service because there are already many existing identifier schemes and new identifier schemes for this service could be designed in the future.

7.4 ID tag requirement

TAG-001: Several types of ID tags are required to be used.

ID tags should involve several kinds of tags such as RFIDs, smart contact-less cards, active tags, IR tags, RF tags, printed tags such as barcodes and two-dimensional barcodes. This is because, technically, there is no tag today that satisfies all uses. For example, RFIDs are of high performance for reading identifiers but are rather expensive. Barcodes are of very low-cost, but are very easy to forge. Therefore, the best tag differs according to the condition of applications and users.

7.5 ID terminal requirements

TRM-001: ID terminal is required to be equipped with at least one reader of ID tags, such as an RFID reader, IrDA receiver, or high resolution camera for barcode and two-dimensional barcode recognition. An ID terminal is recommended to be equipped with a writer of ID tags.

TRM-002: ID terminal is required to be equipped with a wide area public communication interface such as a mobile communication interface or an IP network interface over a Wi-Fi connection.

TRM-003: ID terminal is required to be equipped with a multimedia information input/output function such as multimedia information browsing and web input function.

TRM-004: When multiple ID tag readers are provided, ID terminal is required to provide the selection function of the ID tag readers.

TRM-005: ID terminal can optionally be equipped with a multiple band RFID reader which supports interrogation of ID tags that use different frequency bands.

TRM-006: ID terminal is required to provide a frequency band selection function if it is equipped with a multiple band RFID reader.

7.6 ID resolution function requirements

RSL-001: ID resolution function is required to be able to resolve an identifier into the necessary address information to access the multimedia information related to the identifier, such as URL, IP address and telephone number.

RSL-002: ID resolution functions are recommended to be managed by different organizations such as businesses, non-profit organizations, governments and individual users. They may be distributed among multiple different servers.

7.7 Multimedia information delivery function requirements

MID-001: Multimedia information delivery function is required to reliably request and receive information.

MID-002: Multimedia information delivery function is required to be able to deliver the multimedia information according to the address information, such as the URL returned by the ID resolution function to the ID terminal.

MID-003: Multimedia information delivery function is recommended to be established and maintained by various types of organization, such as businesses, non-profit organizations, governments and individual users. It is recommended to be open for every organization and individual.

7.8 Wide area public communication requirement

WAN-001: Wide area public communication is required to mediate the communication among information terminals, multimedia information delivery functions and ID resolution functions. For example, the Internet, mobile network or NGN can satisfy this requirement.

7.9 Security requirements

SEC-001: Tag-based identification is required to protect privacy.

Some consumers are concerned about privacy threats incurred by ID tags. Especially RFID reader-equipped mobile phones or secret RFID reader/writers may threaten privacy because consumers cannot be aware of exposing their private information to RFIDs. The exposed information may include brand names, manufacturer, price, etc., of goods that the consumer possesses.

SEC-002: ID resolution functions and multimedia information delivery functions are required to be aware of validation.

Some information associated with objects, places and persons may be available only to a limited number of valid users. In this case, some access control mechanism with a validation check is necessary at the ID resolution function or the multimedia information delivery function.

7.10 Quality of service (QoS) requirement

QOS-001: Proper QoS function is required to be provided by each functional component and communication function for each service and application triggered by tag-based identification.

Tag-based identification triggers several types of multimedia service such as web page reference services, video delivery services and video conference services. These services have their own different QoS requirements. So, each functional component and communication function should provide satisfactory QoS for each service.

Appendix I

Service description in applications

(This appendix does not form an integral part of this Recommendation)

The multimedia information access triggered by tag-based identification is useful in various fields such as medical applications in hospitals and drug stores, manufacture, agriculture, library management, personal safety, welfare, shopping, leisure (such as sight-seeing), logistics and supply chain management. Table I.1 summarizes these application fields. This appendix describes multimedia information access services triggered by tag-based identification in eight typical example applications; however, application is not restricted to these services. It is also noted that some other scenarios of tag-based identification applications and services are given in Appendix III to [b-ITU-T Y.2213].

I.1 u-Museum

u-Museum (ubiquitous museum) provides a multimedia information retrieval service for visitors, such as guidance of exhibited art pieces, navigation in the gallery, and advertisement information for museum shops. This service is implemented by RFID tags, active infrared tags, mobile terminals with an RFID reader and infrared receiver, multimedia database of exhibits, wired/wireless networks, and so on. In the u-Museum, an active infrared tag is put at the entrance gate of an exhibition room, and sends the identifier of the room. When a visitor with a mobile terminal walks through the gate, the terminal receives the identifier, retrieves the information of the exhibition in this room, and shows the information to the visitor. The exhibition room shows several pieces of fine art and a tiny RFID tag is embedded in the explanation plate of each exhibit. The user can get precise information on the exhibits by touching the mobile terminal on the plate. When the visitor wants to go to the next exhibit, the system navigates the route according to the art tour route. If the visitor takes a wrong turn, the mobile terminal receives an unexpected location identifier from an infrared tag. Then the mobile terminal gives a warning to the visitor.

I.2 Multimedia information download via posters

Multimedia information may be assigned to an RFID tag attached to a movie advertisement poster; this information could include images, audio/music, movie segments, news information, or the portal site for booking a ticket. If the user touches his/her mobile phone with an RFID reader on the RFID in the poster, he/she receives a list of the candidate services from the network. Then the user can pick up the desired information service by operating the mobile phone.

I.3 Operating manual for a product

Recently, several classes of electronic equipment have been released that have very complicated operation sequences. Typical examples of such equipment are DVD players, hard-disk video recorders, video cameras, digital televisions, facsimile machines, etc. These machines are often used for a long time, and the user sometimes loses the operating manuals. To assist in this situation, each of these machines can bear an RFID or 2D barcode which contains an identifier of the machine. This identifier is associated with the details of the version and options of the product. Using this identifier, the user can select a proper operating manual from a large manual database, and refer to it via a network with his/her hand-held terminal.

I.4 Food traceability

The confidence in food wavers due to problems such as bovine spongiform encephalopathy (BSE) and pesticide residue found in vegetables. Consequently, food safety has become a global issue with a central focus of the consumer.

Traceability of the food chain enables tracing and tracking of food and the information at each stage of the food chain, including production, processing, distribution and sales. When a food accident occurs, consumers can search for information and confirm the safety of the food in their refrigerators.

An identifier of the food is attached to the product using RFID or a small barcode. Food chain information is associated with the identifier so that the consumer can retrieve the food information from multimedia information delivery functions via available networks.

There may be a use case of dual band communication. The food chain management is a subset of the supply chain management where UHF-type RFID tags are popularly deployed. ID tags for the food traceability may be UHF-type RFID tags. An ID terminal like a cell phone needs to be equipped with a UHF-type RFID reader to capture data from a UHF-type RFID tag attached to a food item. After a safety check of a food item, and due to security reasons, an end user may want to buy the food using his/her ID terminal as a mobile commerce terminal that deploys the HF communication technology. For example, NFC is an HF communication technology and an NFC-equipped ID terminal can support payment for the food. From an implementation perspective, installation of two RFID readers/writers in an ID terminal is not preferable. But a dual band RFID reader/writer that supports not only a single air interface but also multiple frequency bands (e.g., HF and UHF) solves the implementation difficulty. In this scenario, the ID terminal does a safety check of a food item using UHF, and does payment via HF, resulting in a dual band communication.

I.5 Business card with personal identifier

Suppose that an identifier of a businessman is written on a business card. The identifier is associated with the latest contact address data record, including telephone number, fax number and e-mail address. His/her business client could get all the latest information from this identifier even after he/she has moved to another office or company.

I.6 Presence service with multimedia information

Imagine a theatre in which every visitor had a ticket with RFID, and every seat in the theatre contained an RFID reader. When the visitor enters the theatre and takes a seat, he/she puts the ticket on the RFID reader located in the arm of the seat. The reader reads the ticker identifier and automatically notifies the theatre office of the visitor status through the theatre management application.

I.7 Location-aware information delivery for commercial advertisement

Suppose that an ID tag is embedded in front of the doors of a department store or shop. In this case, if the ID tag is an active RFID or an infrared tag so as to be able to send an identifier to ID terminals, users can obtain the identifier automatically without explicit operations. An advertisement movie clip or speech message is associated to the location identifier. In this scenario, when a visitor walks in front of the entrance of the store or shop, he/she is automatically notified of the advertisement message.

I.8 Sightseeing information delivery

Suppose that each sightseeing spot is equipped with RFIDs, infrared tags, active RF tags, or barcodes including the identifier of the location. At the sightseeing spot, a visitor with a handheld terminal automatically reads the location identifier and receives sight-seeing video clips or speech explanation with pictures retrieved via available networks.

I.9 Visitor identification and guidance service with multimedia information

In this scenario, ID terminal is a kind of smart phone which has four components of interest: a UHF ID tag reader/writer, an HF ID tag, an HF ID tag reader/writer and a human presence management application. In this scenario, a visitor is given a visitor ID tag and the visitor reads that tag by the HF ID tag reader/writer and writes that information to the HF ID tag in his/her smartphone. The visitor is guided by UHF ID tags and the presence management system to the final destination in the building. The security building has an HF ID reader at the entrance gate and UHF ID tags are installed on the walls of corridors. The HF ID reader identifies a visitor and the UHF ID reader/writer of the smart phone reads the UHF ID tags on walls and displays the direction and route to the final destination. Therefore, a visitor who has a smartphone as described above can enter the security building by obtaining the admission credential by using the HF ID tag and will be guided to the final destination in the building by using the UHF ID tag reader/writer.

The visitor can open all the doors to the destination by using his/her smartphone.

Table I.1 – Applications sorted by fields

Field	Application	Business types	Function	Purpose	Tag mobility (Note)
Supply chain	Food traceability (see clause I.4)	B2B, B2C	Information retrieval, information monitoring	Increasing food safety, increasing total visibility of food chain	Mobile tag
Medical	Confirmation of drugs	B2B, B2C	Information retrieval, information monitoring	Reducing human error, anti-counterfeit	Mobile tag
Museum	u-Museum (see clause I.1)	B2C	Information retrieval	Value-added service	Mobile tag
Office	Business card with personal identifier (see clause I.5)	B2B	Information retrieval	Value-added service	Mobile tag
	Automatic telephone calling using business card with personal identifier	B2B	Bidirectional information exchange	Value-added service	Mobile tag
Family safety	Child monitoring	B2C	Information monitoring	Increasing safety	Mobile tag
Shopping	Advertisement, shopping guidance (see clause I.7)	B2C	Information retrieval	Value-added service	Fixed tag
Advertisement	Music information download from advertisement poster (see clause I.2)	B2C	Information retrieval	Effective advertisement	Fixed tag

Table I.1 – Applications sorted by fields

Field	Application	Business types	Function	Purpose	Tag mobility (Note)
Customer support	Operating manual download (see clause I.3)	B2C	Information retrieval	Value-added service	Mobile tag
Leisure	Sight-seeing information, navigation (see clause I.8)	B2C G2C	Information retrieval	Value-added service	Fixed tag
	Presence service of audience (see clause I.6)	B2B	Information monitoring	Saving cost for audience management	Fixed tag
Welfare	Location-aware information	G2C, B2C	Information retrieval	Value-added service, increasing safety	Fixed tag
Security and convenience	Identification and guidance	B2C	Information retrieval and monitoring	Increasing security and convenience	Fixed and mobile tag
NOTE – If the tag is attached to an object that is fixed in location, it is called a "fixed tag", otherwise it is called a "mobile tag".					

Bibliography

[b-ITU-T Y.2213] Recommendation ITU-T Y.2213 (2008), *NGN service requirements and capabilities for network aspects of applications and services using tag-based identification.*


Sold
Sales
Total
Benefits
0 0 € 1 € -1 €
30 1.560 € 13.591 € -12.031 €
60 3.120 € 27.181 € -24.061 €
90 4.680 € 40.771 € -36.091 €
120 6.240 € 54.361 € -48.121 €
150 7.800 € 67.951 € -60.151 €
180 9.360 € 81.541 € -72.181 €
210 10.920 € 95.131 € -84.211 €
240 12.480 € 108.721 € -96.241 €
270 14.040 € 122.311 € -108.271 €
300 15.600 € 135.901 € -120.301 €
330 17.160 € 149.491 € -132.331 €
360 18.720 € 163.081 € -144.361 €
390 20.280 € 176.671 € -156.391 €
420 21.840 € 190.261 € -168.421 €
450 23.400 € 203.851 € -180.451 €
480 24.960 € 217.441 € -192.481 €
510 26.520 € 231.031 € -204.511 €
540 28.080 € 244.621 € -216.541 €
570 29.640 € 258.211 € -228.571 €
600 31.200 € 271.801 € -240.601 €
Benefits -0
300.000 €
250.000 €
200.000 €
150.000 €
100.000 €
50.000 €
0 €
0 60 120 180 240 300 360 420 480 540 600
Sales

Y.4552/Y.2078

Application support models of the Internet of Things

Application support models of the Internet of Things

Summary

Recommendation ITU-T Y.4552/Y.2078 provides application support models of the Internet of things (IoT). This Recommendation describes the basis of IoT application support models: the configurable application support model, the adaptable application support model and the reliable application support model. These three application support models are described in functional view, implementation view and deployment view, in order to identify, respectively, the configurable capabilities, the adaptable capabilities and the reliable capabilities for support of IoT applications having some characteristic requirements.

This Recommendation describes the IoT configurable capabilities that extend the IoT basic capabilities specified in Recommendation ITU-T Y.4401/Y.2068 in order to enable the IoT applications to configure the IoT capabilities based on their characteristic requirements.

This Recommendation describes the IoT adaptable capabilities that extend the IoT basic capabilities specified in Recommendation ITU-T Y.4401/Y.2068 in order to enable the IoT applications to adapt to the IoT capabilities based on their characteristic requirements.

This Recommendation describes the IoT reliable capabilities that extend the IoT basic capabilities specified in Recommendation ITU-T Y.4401/Y.2068 in order to support the IoT applications by the IoT capabilities with required degrees of reliability for fulfilling their characteristic requirements.

Use cases from the smart home environment provide examples about the usage of the IoT application support models.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T Y.4552/Y.2078	2016-02-13	13	11.1002/1000/12707

Keywords

Adaptable capability, application support model, configurable capability, deployment view, functional view, implementation view, Internet of things (IoT), reliable capability, requirements.

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

Table of Contents

Recommendation ITU-T Y.4552/Y.2078

Application support models of the Internet of Things

1 Scope

This Recommendation describes basis of application support models of the Internet of things (IoT) and specifies three application support models of the IoT: the configurable application support model, the adaptable application support model and the reliable application support model.

The three application support models are specified in functional view, implementation view and deployment view respectively.

The configurable capabilities, adaptable capabilities and reliable capabilities related, respectively, to each of the three models, are also identified and described.

The scope of this Recommendation includes:

- The basis of application support models;
- The functional view, the implementation view and the deployment view of the configurable application support model and related configurable capabilities that extend the IoT basic capabilities specified in [ITU-T Y.4401] to enable the IoT applications to configure the IoT capabilities based on their characteristic requirements;
- The functional view, the implementation view and the deployment view of the adaptable application support model and related adaptable capabilities that extend the IoT basic capabilities specified in [ITU-T Y.4401] to enable the IoT applications to adapt to the IoT capabilities based on their characteristic requirements;
- The functional view, the implementation view and the deployment view of the reliable application support model and related reliable capabilities that extend the IoT basic capabilities specified in [ITU-T Y.4401] to support the IoT applications by the IoT capabilities with required degrees of reliability for fulfilling their characteristic requirements.

All capabilities identified and specified in this Recommendation are numbered and summarized in the annexes.

Appendix I shows three use cases of the IoT application support models from the smart home environment.

NOTE – Only three IoT application support models are described and specified in this Recommendation. The specification of other application support models is outside the scope of this Recommendation and for further consideration.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T Y.4000] Recommendation ITU-T Y.4000/Y.2060 (2012), *Overview of the Internet of things*.

[ITU-T Y.4100] Recommendation ITU-T Y.4100/Y.2066 (2014), *Common requirements of the Internet of things.*

[ITU-T Y.4401] Recommendation ITU-T Y.4401/Y.2068 (2015), *Functional framework and capabilities of the Internet of things.*

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 application [b-ITU-T Y.2091]: A structured set of capabilities, which provide value-added functionality supported by one or more services, which may be supported by an API interface.

3.1.2 application domain [ITU-T Y.4100]: An area of knowledge or activity applied for one specific economic, commercial, social or administrative scope.

NOTE – Transport application domain, health application domain and government application domain are examples of application domains.

3.1.3 device [ITU-T Y.4000]: With regard to the Internet of things, this is a piece of equipment with the mandatory capabilities of communication and the optional capabilities of sensing, actuation, data capture, data storage and data processing.

3.1.4 functional entity [b-ITU-T Y.2012]: An entity that comprises an indivisible set of specific functions. Functional entities are logical concepts, while groupings of functional entities are used to describe practical, physical implementations.

3.1.5 Internet of things (IoT) [ITU-T Y.4000]: A global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies.

NOTE 1 – Through the exploitation of identification, data capture, processing and communication capabilities, the IoT makes full use of things to offer services to all kinds of applications, whilst ensuring that security and privacy requirements are fulfilled.

NOTE 2 – From a broader perspective, the IoT can be perceived as a vision with technological and societal implications.

3.1.6 thing [ITU-T Y.4000]: With regard to the Internet of things, this is an object of the physical world (physical things) or the information world (virtual things), which is capable of being identified and integrated into communication networks.

3.2 Terms defined in this Recommendation

None.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

IoT Internet of Things

M2M Machine-to-Machine

QoS Quality of Service

5 Conventions

In this Recommendation:

The keywords "is required to" indicate a requirement which must be strictly followed and from which no deviation is permitted if conformance to this document is to be claimed.

The keywords "is recommended" indicate a requirement which is recommended but which is not absolutely required. Thus this requirement need not be present to claim conformance.

The keywords "can optionally" and "may" indicate an optional requirement which is permissible, without implying any sense of being recommended. These terms are not intended to imply that the vendor's implementation must provide the option and the feature can be optionally enabled by the network operator/service provider. Rather, it means the vendor may optionally provide the feature and still claim conformance with the specification.

6 Basis of IoT application support models

6.1 Concepts and purpose of IoT application support models

The IoT application support models refer to different sets of the IoT capabilities, including their relations, which can support IoT applications with some characteristic requirements, such as application adaptability, reliability and manageability.

NOTE 1 – The application characteristic requirements, as named in this Recommendation, are part of the common requirements as specified in [ITU-T Y.4100]. The application characteristic requirements refer to requirements related with some characteristics of IoT applications, such as the adaptability of M2M applications, the reliability of e-health applications and the configurability of smart city applications.

The IoT application support models are used to guide the design, implementation and deployment of the IoT capabilities to fulfil application characteristic requirements, in order to establish a common service platform [ITU-T Y.4401] for support of IoT applications across different application domains.

NOTE 2 – The service platform established by implementing and deploying capabilities of the IoT application support models may be used to shorten the time period and reduce the cost of developing the IoT applications with characteristic requirements, such as configurable, adaptable, or reliable requirements, by making usage of the capabilities of the service platform. Appendix I describes some use cases from the smart home environment showing examples about the usage of the IoT application support models.

In particular, the purposes of the IoT application support models are as follows: the first one is to specify groups of IoT capabilities in order to facilitate the selection of IoT capabilities [ITU-T Y.4401] for the support of IoT applications with some characteristic requirements; the second one is to derive, based on the selected IoT capabilities, other IoT capabilities, not explicitly identified in [ITU-T Y.4401], as necessary in order to facilitate the design, implementation and deployment of the IoT capabilities for support of IoT applications with some characteristic requirements.

In this Recommendation, the framework of the IoT application support models and three specific application support models: the adaptable application support model, the reliable application support model and the configurable application support model, are specified. Clause 6.2 provides the rationale for this classification of the IoT application support models.

NOTE 3 –This Recommendation specifies only three application support models. Because of different IoT application characteristic requirements, different IoT application support models could be specified. Other application support models are for further consideration with respect to other relevant characteristic requirements of IoT applications.

The configurable application support model refers to the set of IoT capabilities, including their relations, to support the IoT applications with the characteristic requirement of configurability. The configurable application support model includes the IoT capabilities that can be configured by IoT applications, such as some service capabilities and communication capabilities that are related with the IoT applications.

The reliable application support model refers to the set of IoT capabilities, including their relations, to support the IoT applications with the required degrees of reliability. The reliable application support model includes the IoT capabilities that can enhance the reliability of IoT applications, such as reliable data communication capability.

The adaptable application support model refers to the set of IoT capabilities, including their relations, to support the IoT applications with the characteristic requirement of adaptability. The adaptable application support model includes the IoT capabilities that are adaptable to different application contexts, such as content awareness capability and context awareness capability.

6.2 Rationale for the selection of the IoT applications support models

Regarding the possible diverse classifications that can be considered for IoT applications, different classes for a given classification may require different application support models.

One possible classification of IoT applications is based on the characteristics of things, characteristics of IoT users and other functional characteristics of IoT.

The characteristics of things may include mobility, intelligent ability, etc. The characteristics of IoT users may include mobility, non-human operated, etc. Other functional characteristics of IoT may include content awareness, context awareness, etc.

NOTE 1 – For example, the IoT applications in support of things with mobility belong to the category of mobile thing applications of IoT; the IoT applications in support of things with intelligent ability belong to the category of smart thing applications of IoT; the IoT applications in support of non-human operated users of IoT belong to the category of IoT applications with non-human operators.

This classification of IoT applications may be too diverse to derive common application support models usable across different application domains. So this classification of IoT applications is not suitable as the basis to describe the IoT application support models.

Another classification of IoT applications is based on the non-functional requirements of IoT applications as specified in [ITU-T Y.4100], such as reliability, availability, manageability and adaptability. Based on this classification, IoT applications can be classified into reliable applications, manageable applications, adaptable applications, etc. Even if there are some differences among these non-functional requirements across different application domains, these differences consist in the absence of certain requirements in given application domain(s), or in the different strengths of certain requirements to be satisfied at the implementation and deployment level. So the application support models derived from this IoT application classification can be used across different application domains. This classification of IoT applications is suitable as the basis to describe the three IoT application support models specified in this Recommendation.

NOTE 2 – The three IoT application support models are related with several practical IoT applications, such as M2M applications, e-health applications and smart city applications. The configurable application support model specified in this Recommendation may be used to support smart city applications because these applications may address different application domains and require different configurations in these different domains. The adaptable application support model specified in this Recommendation may be used to support M2M applications because these applications usually require being able to adapt to different networking or application environments. The reliable application support model specified in this Recommendation may be used to support e-health applications because the e-health applications usually require high reliable networking and service provisioning.

6.3 The three views of IoT application support models

The three views of an IoT application support model consists of the functional view, the implementation view and the deployment view for support of the identified IoT applications with some characteristic requirements.

The functional view consists of the functional groups, including their relations, which support the identified applications.

NOTE 1 – This functional view is based on the functional view of the IoT framework that is specified in [ITU-T Y.4401].

The implementation view consists of the functional entities, including their relations, which support the identified applications.

NOTE 2 – This implementation view is based on the implementation view of the IoT framework that is specified in [ITU-T Y.4401].

The deployment view consists of the functional components, including their relations, which support the identified applications.

NOTE 3 – This deployment view is based on the deployment view of the IoT framework that is specified in [ITU-T Y.4401].

NOTE 4 – Based on the three views of the IoT application support models, some IoT application support capabilities can be derived for support of the IoT applications with some characteristic requirements.

NOTE 5 – The IoT application support capabilities derived from the three views of the IoT application support models are aligned with and extend the IoT capabilities specified in [ITU-T Y.4401] in order to fulfill some characteristic requirements of the IoT applications. The three views of the IoT application support models can identify the IoT application support capabilities required in the stages of designing, implementing and deploying the IoT applications.

7 The configurable application support model

The configurable application support model consists of the functional view, implementation view and deployment view and related capabilities.

NOTE – The three views of application support model can be used to derive and validate the capabilities for support of configurable applications of the IoT.

7.1 The description of the configurable application support model

7.1.1 The functional view of the configurable application support model

The functional view of the configurable application support model consists of a configurable management group, a configurable security and privacy protection group, a configurable data management group, a configurable service provision, a configurable communication group, a configurable connectivity group and a configurable application support group and the interactions among these groups as illustrated in Figure 7-1. Each functional group contains related capabilities for support of the IoT configurable applications.

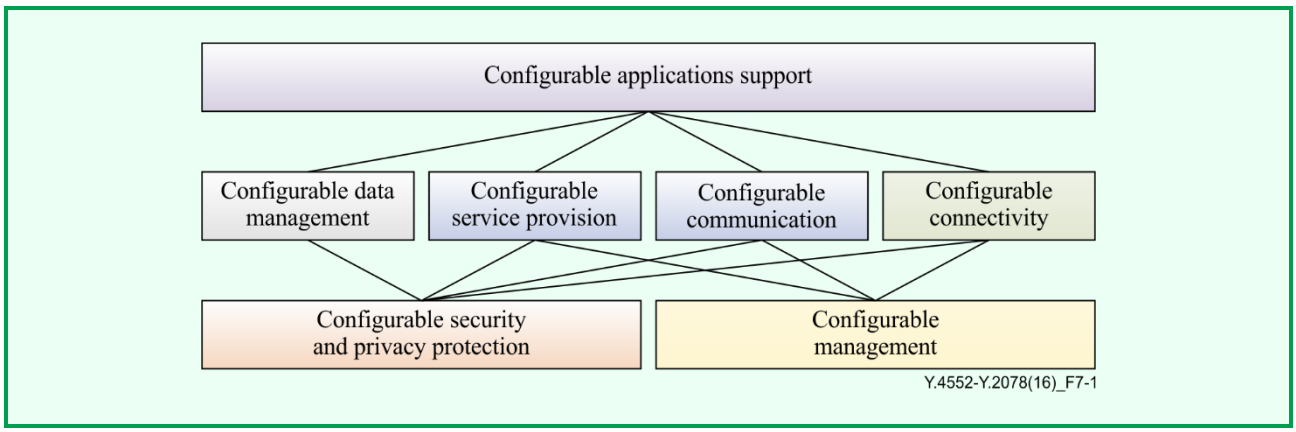


Figure 7-1 – The functional view of the configurable application support model

The configurable security and privacy protection group is related to the configurable data management group, the configurable service provision group, the configurable communication group and the configurable connectivity group, which refer to the fact that the other functional groups rely on the security and privacy protection capabilities specified in this functional group to protect their configurable capabilities for support of IoT applications.

The configurable management group is related to the configurable service provision group, the configurable communication group and the configurable connectivity group to provide required management capabilities.

The data management group has its own management capabilities, because configurable data management capabilities depend on data models. By ensuring the management of the data models by the data management group's own management capabilities, configuration management can be simplified.

The security and privacy protection group also has its own configurable management capabilities in order to prevent any possible intrusion or attack from external configuration management.

The configurable application support group is related to the configurable data management group, the configurable service provision group, the configurable communication group and the configurable connectivity group to allow the exposure of the configurable capabilities contained in these functional groups to IoT applications.

NOTE 1 – The functional view of the configurable application support model can be used to identify the functional groups related to the configurable capabilities for support of IoT applications.

NOTE 2 – In the functional view of the configurable application support model, there is no interaction among the configurable data management group, the configurable service provision group, the configurable communication group and the configurable connectivity group, because each of these functional groups does not need to interact with others to provide configurable capabilities.

7.1.2 The implementation view of the configurable application support model

The implementation view of the configurable application support model consists of a configurable management and identity management entity, a configurable IoT security and privacy protection entity, a configurable IoT gateway entity, a configurable end-user device entity, a configurable transport control entity, a configurable IoT transport control entity, a configurable service control entity, a configurable IoT service control entity, a configurable IoT data management entity and a configurable application support entity and the interactions among these entities as illustrated in Figure 7-2.

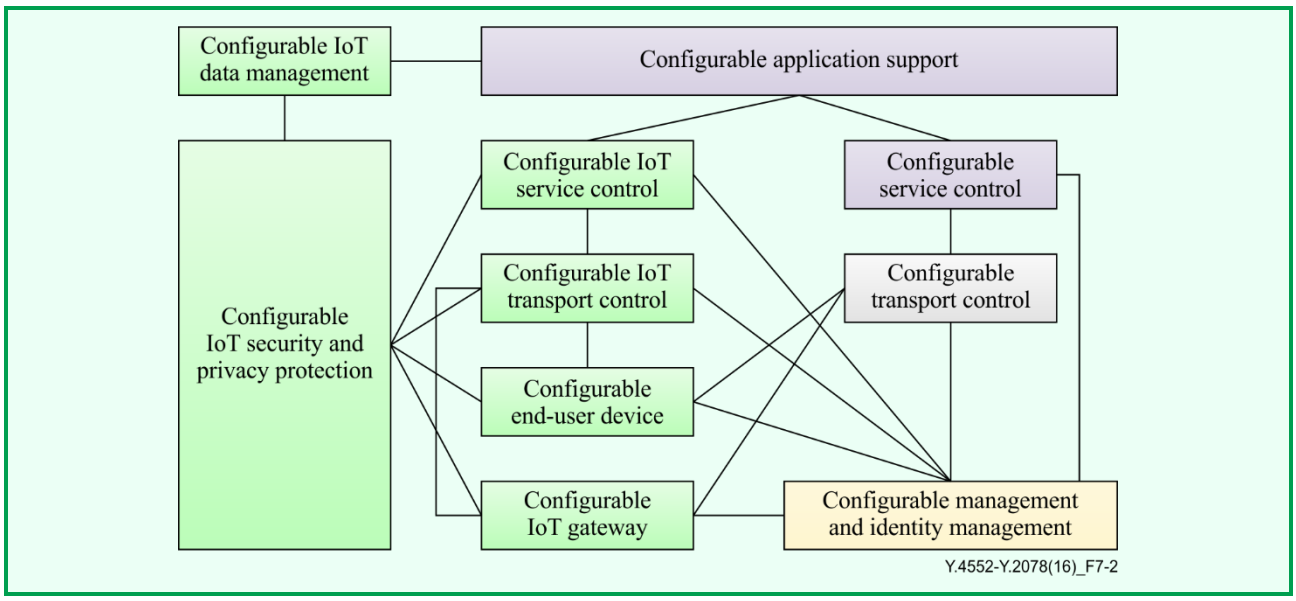


Figure 7-2 – The implementation view of the configurable application support model

The configurable management and identity management entity is related to the configurable IoT gateway entity, the configurable end-user device entity, the configurable transport control entity, the configurable IoT transport control entity, the configurable service control entity and the configurable IoT service control entity, in order to provide the required management capabilities for these functional entities.

The IoT security and privacy protection entity is related to the configurable IoT gateway entity, the configurable end-user device entity, the configurable IoT transport control entity, the configurable IoT service control entity and the configurable IoT data management entity, in order to provide the required IoT related security and privacy protection capabilities for these functional entities.

The configurable transport control entity is related to the configurable IoT gateway entity and the configurable end-user device entity in order to provide configurable transport capabilities for these functional entities. The configurable IoT transport control entity is related to the configurable IoT gateway entity and the configurable end-user device entity in order to provide configurable IoT related transport capabilities for these functional entities.

The configurable service control entity is related to the configurable transport control entity in order to expose the configurable transport control capabilities in the configurable service control entity. The configurable IoT service control entity is related to the configurable IoT transport control entity in order to allow the exposure of the configurable IoT related transport control capabilities in the configurable IoT service control entity.

The configurable application support entity is related to the configurable IoT data management entity, the configurable IoT service control entity and the configurable service control entity, in order to allow the exposure of all configurable capabilities specified in clause 7.2 by the application support entity specified in [ITU-T Y.4401].

7.1.3 The deployment view of the configurable application support model

The deployment view of the configurable application support model consists of the configurable device manager component, the configurable IoT gateway component, the configurable end-user device component, the configurable network manager component, the configurable IoT network controller component, the configurable service manager component, the configurable IoT service controller component, the configurable IoT data server component and the configurable service platform component and the interactions among these components as illustrated in Figure 7-3.

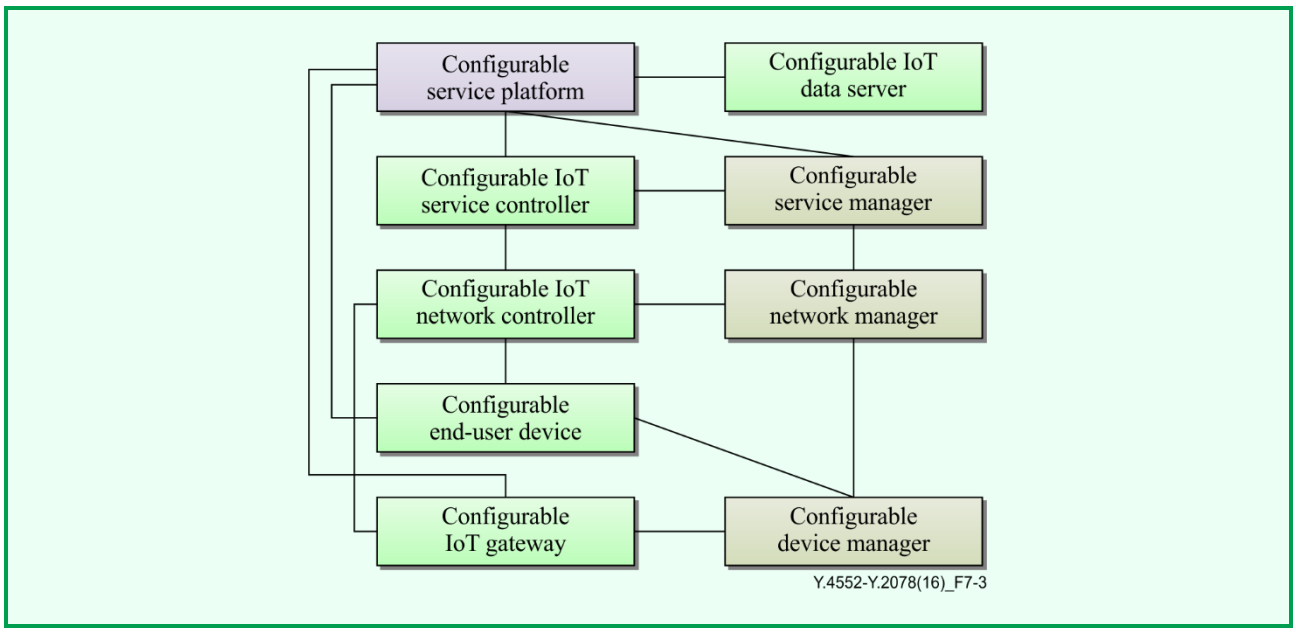


Figure 7-3 – The deployment view of the configurable application support model

The configurable service platform component is related to the configurable IoT data server component to enable the configurable capabilities for support of the IoT applications to be configured with different requirements on data of things. The configurable service platform component interacts with the configurable service manager component and the configurable IoT service controller component to enable the configurable capabilities for support of the IoT applications with different requirements of service provisioning, such as service creation or service customization.

The configurable service platform component is related to the configurable end-user device to enable the configurable capabilities for support of the IoT applications to be configured with different requirements of end-users. The configurable service platform component is related to the configurable IoT gateway component to enable the configurable capabilities for support of the IoT applications to be configured with different requirements of IoT devices, such as different ways of capturing and transferring data of things.

The configurable IoT network controller component is related to the configurable end-user device component and the configurable IoT gateway component to enable the configurable capabilities for support of the IoT applications with different network requirements.

The configurable network manager component is related to the configurable service manager component and the configurable device manager component to enable the configurable capabilities for support of the IoT applications to be configured across all functional layers of IoT, such capturing, buffering, transferring and analysing the data of things.

7.2 The capabilities of the configurable application support model

Based on the categories of the IoT basic capabilities specified in [ITU-T Y.4401] and the description of the configurable application support model specified in clause 7.1, the capabilities of the configurable application support model can be classified into the following groups: the configurable service provision capabilities, the configurable communication capabilities, the configurable data management capabilities, the configurable connectivity capabilities, the configurable management capabilities, the configurable application support capabilities and the configurable security and privacy protection capabilities.

The capabilities of the configurable application support model are specified from the perspective of configurable application support components as described in the deployment view of the configurable application support model in clause 7.1.3 because these capabilities are implemented, deployed and used in these configurable application support components.

The following clauses describe, respectively, these capabilities of the configurable application support model. These same capabilities are numbered and summarized in Annex A.

NOTE – In the following clauses, the capability numbers, as shown in Annex A, are put between square brackets "[ ]" and inserted at the end of the description of the corresponding capability.

7.2.1 Configurable service provision capabilities

The configurable service provision capabilities extend the service provision capabilities of the IoT basic capabilities specified in [ITU-T Y.4401], in order to enable IoT applications to configure the IoT service provision capabilities based on their requirements.

The configurable service provision capabilities include the configurable service prioritization capability, the configurable service composition capability and the configurable location based and context-aware service capability.

The configurable service prioritization capability enables the IoT applications to configure services with different priorities, in order to provide differentiated services based on their requirements [A-1-1].

The configurable service composition capability enables the IoT applications to configure service creation or service customization based on their requirements [A-1-2].

The configurable location based and context-aware service capability enables the IoT applications to configure services that are provided both on the location information and related context and on the predefined rules or policies, in order to fulfil their requirements [A-1-3].

7.2.2 Configurable communication capabilities

The configurable communication capabilities extend the communication capabilities of the IoT basic capabilities specified in [ITU-T Y.4401], in order to enable IoT applications to configure the IoT communication capabilities based on their requirements.

The configurable communication capabilities include the configurable event-based communication capability, the configurable periodic communication capability, the configurable communication mode capability, the configurable quality of service (QoS) communication capability, the configurable content-aware communication capability and the configurable location based communication capability.

The configurable event-based communication capability enables the IoT applications to configure different events in order to initiate communication based on the requirements of the IoT applications [A-2-1].

The configurable periodic communication capability enables the IoT applications to configure the rules in order to periodically initiate communication based on the requirements of the IoT applications [A-2-2].

The configurable communication mode capability enables the IoT applications to configure different modes of communications in the transport network in order to transfer data from the source(s) to the destination(s) based on the requirements of the IoT applications [A-2-3].

The configurable quality of service communication capability enables the IoT applications to configure the related mechanisms in order to guarantee the QoS required for the delivery and processing of data (e.g., time-sensible data) based on the requirements of the IoT applications [A-2-4].

The configurable content-aware communication capability enables the IoT applications to configure the parameters related to the content and selected path for routing or blocking data transfer based on the requirements of the IoT applications [A-2-5].

The configurable location based communication capability enables the IoT applications to configure the parameters related to the locations and predefined rules in order to initiate communication based on the requirements of the IoT applications [A-2-6].

7.2.3 Configurable data management capabilities

The configurable data management capabilities extend configurable capabilities to the data management capabilities of the IoT basic capabilities specified in [ITU-T Y.4401], in order to enable IoT applications to configure the IoT data management capabilities based on their requirements.

The configurable data management capabilities include the configurable data storage capability, the configurable data processing capability, the configurable information exchange capability, the configurable semantic data operation capability and the configurable autonomic data operation capability.

The configurable data storage capability enables the IoT applications to configure the rules or the policies for storing data based on the requirements of the IoT applications [A-3-1].

The configurable data processing capability enables the IoT applications to configure the rules or the policies for processing data based on the requirements of the IoT applications [A-3-2].

The configurable information exchange capability enables the IoT applications to configure the parameters for sending data to or receiving data from external data sources, e.g., data centres and data servers outside the IoT based on the requirements of the IoT applications [A-3-3].

The configurable semantic data operation capability enables the IoT applications to configure the parameters for semantic annotating, semantic discovering, semantic storing and semantic composition of data of things based on the requirements of the IoT applications [A-3-4].

The configurable autonomic data operation capability enables IoT applications to configure the parameters for automatically collecting, aggregating, transferring, storing, analyzing data of things, as well as automatically managing these data operations based on the requirements of the IoT applications [A-3-5].

7.2.4 Configurable connectivity capabilities

The configurable connectivity capabilities extend the connectivity capabilities of the IoT basic capabilities specified in [ITU-T Y.4401], in order to enable IoT applications to configure the IoT communication capabilities based on their requirements.

The configurable connectivity capabilities include the configurable identification based connectivity capability, the configurable things' status notification capability, the configurable device mobility capability and the configurable and adaptable connectivity capability.

The configurable identification based connectivity capability enables the IoT applications to configure the parameters for connectivity establishment based on the identification of things and the requirements of the IoT applications [A-4-1].

The configurable things' status notification capability enables the IoT applications to configure the rules of automatic notification of the status of things and its changes based on the requirements of the IoT applications [A-4-2].

The configurable device mobility capability enables the IoT applications to configure the parameters for maintaining the connectivity with the IoT when end-user devices or IoT gateways are moving, based on the requirements of the IoT applications [A-4-3].

The configurable and adaptable connectivity capability enables the IoT applications to configure the parameters for extending connectivity configurations in order to connect with different types of devices of the IoT based on the requirements of the IoT applications, in order to be adaptable to different technologies in devices of IoT [A-4-4].

7.2.5 Configurable security and privacy protection capabilities

The configurable security and privacy protection capabilities extend the security and privacy protection capabilities of the IoT basic capabilities specified in [ITU-T Y.4401], in order to enable IoT applications to configure the IoT security and privacy protection capabilities based on their requirements.

The configurable security and privacy protection capabilities include the configurable communication security capability, the configurable data management security capability, the configurable service provision security capability, the configurable security integration capability and the configurable mutual authentication and authorization capability.

The configurable communication security capability enables IoT applications to configure the rules and policies for supporting secure, trusted and privacy protected communication based on the requirements of IoT applications [A-5-1].

The configurable data management security capability enables the IoT applications to configure the rules and policies for providing secure, trusted and privacy protected data management based on the requirements of the IoT applications [A-5-2].

The configurable service provision security capability enables the IoT applications to configure the rules and policies for providing secure, trusted and privacy protected service provision based on the requirements of the IoT applications [A-5-3].

The configurable security integration capability enables the IoT applications to configure the rules and policies for enabling integration of different security policies and techniques related to the IoT functional components based on the requirements of the IoT applications [A-5-4].

The configurable mutual authentication and authorization capability enables the IoT applications to configure the rules and policies for authenticating and authorizing IoT applications and devices before a device accesses IoT based on the requirements of the IoT applications [A-5-5].

7.2.6 Configurable application support capabilities

The configurable application support capabilities extend the application support capabilities of the IoT basic capabilities specified in [ITU-T Y.4401], in order to enable IoT applications to configure the IoT application support capabilities based on their requirements.

The configurable application support capabilities include the configurable group management capability, the configurable time synchronization capability, the configurable orchestration capability and the configurable application support operation acknowledgement capability.

The configurable group management capability enables the IoT applications to configure the parameters for creating, modifying, deleting and querying IoT groups, as well as adding, modifying, deleting and querying IoT group members, based on the requirements of the IoT applications [A-6-1].

The configurable time synchronization capability enables the IoT applications to configure the parameters for synchronizing the time among related functional components with different degrees of reliability, in order to support global or local time stamping for applications based on the different QoS requirements of the IoT applications [A-6-2].

The configurable orchestration capability enables the IoT applications to configure the parameters for automatic coordination of service provisioning or device operations based on the requirements of the IoT applications [A-6-3].

The configurable application support operation acknowledgement capability enables the IoT applications to configure the parameters for acknowledging the correct operations requested by applications in order to support reliable application operations in the IoT, based on the requirements of the IoT applications [A-6-4].

7.2.7 Configurable management capabilities

The configurable management capabilities enhance the management capabilities of the IoT basic capabilities specified in [ITU-T Y.4401], in order to enable IoT applications to configure the IoT management capabilities based on their requirements.

The configurable management capabilities include the configurable redundant deployment enablement capability, the configurable service integrity check capability, the configurable data integrity check capability, the configurable device integrity check capability, the configurable security integrity check capability and the configurable user profile integrity check capability.

The configurable redundant deployment enablement capability enables the IoT applications to configure deployment of redundant functional components of the IoT in order to provide different degrees of reliability required in communication, service provision and data management, based on the requirements of the IoT applications [A-7-1].

The configurable service integrity check capability enables the IoT applications to configure the parameters for checking the service lifetime, the available resources required to provide the service in order to provide different degrees of availability in service provisioning, based on the requirements of the IoT applications [A-7-2].

The configurable data integrity check capability enables the IoT applications to configure the parameters for checking the data lifetime, the available attributes of the data and the consistency of data in order to provide different degrees of availability in data management, based on the requirements of the IoT applications [A-7-3].

The configurable device integrity check capability enables the IoT applications to configure the parameters for checking the status of all device functions in order to provide different degrees of availability of IoT devices, based on the requirements of the IoT applications [A-7-4].

The configurable security integrity check capability enables the IoT applications to configure the parameters for checking the consistency of security policies deployed in all functional components of the IoT in order to provide different degrees of availability in security and privacy protection provisioning, based on the requirements of the IoT applications [A-7-5].

The configurable user profile integrity check capability enables the IoT applications to configure the parameters for checking the lifetime, subscription, privacy protection and availability of services subscribed by users in order to provide different degrees of availability in service provisioning and privacy protection for users, based on the requirements of the IoT applications [A-7-6].

8 The adaptable application support model

The adaptable application support model consists of the functional view, implementation view and deployment view and related capabilities.

NOTE 1 – The three views of the application support model can be used to derive and validate the capabilities for support of adaptable applications of the IoT.

NOTE 2 – "Adaptable" capabilities in this Recommendation refer to capabilities that can adjust themselves to make them suitable to their operating environment, including requirements of the IoT applications.

8.1 The description of the adaptable application support model

8.1.1 The functional view of the adaptable application support model

The functional view of the configurable application support model consists of the adaptable application support group, the adaptable data management group, the adaptable service provision group, the adaptable communication group, the adaptable connectivity group and the interactions among these groups as illustrated in Figure 8-1. Each functional group contains related capabilities for support of the IoT adaptable applications. Each functional group contains related capabilities for support of the IoT adaptable applications.

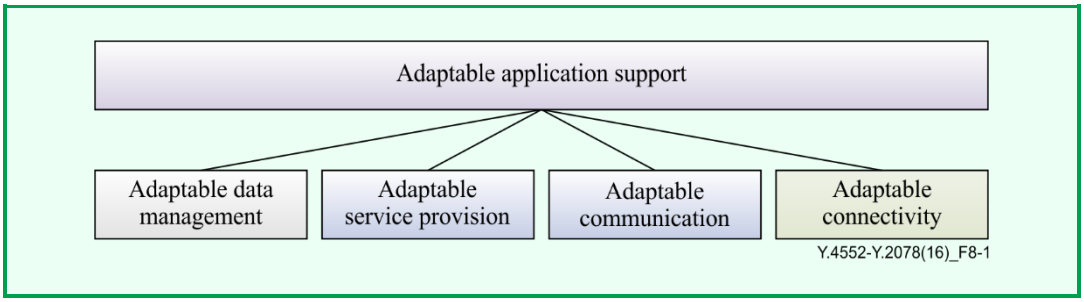


Figure 8-1 – The functional view of the adaptable application support model

The adaptable application support group is related to the adaptable data management group, the adaptable service provision group, the adaptable communication group and the adaptable connectivity group to expose their adaptable capabilities to the IoT applications.

NOTE – Neither the adaptable management group nor the adaptable security and privacy protection group are specified in the functional view of the adaptable application support model, because the functions both in management group and in security and privacy protection group are not adaptable. The management group contains capabilities that can manage the capabilities of the adaptable application support model based on policies or rules predefined by human operators, in order to make these adaptable capabilities controllable by humans, the adaptable application support model does not contain capabilities of the management group.

The security and privacy protection group contains capabilities that can secure the capabilities of the adaptable application support model and protect privacy in these adaptable capabilities based on the policies or rules predefined by human operators, in order to make these adaptable capabilities secured and privacy-protected strictly by humans, the adaptable application support model does not contain capabilities of the security and privacy protection group.

8.1.2 The implementation view of the adaptable application support model

The implementation view of the adaptable application support model consists of the adaptable IoT gateway entity, the adaptable end-user device entity, the adaptable transport control entity, the adaptable IoT transport control entity, the adaptable service control entity, the adaptable IoT service control entity, the adaptable IoT data management entity, the adaptable applications support entity and the interactions among these entities as illustrated in Figure 8-2.

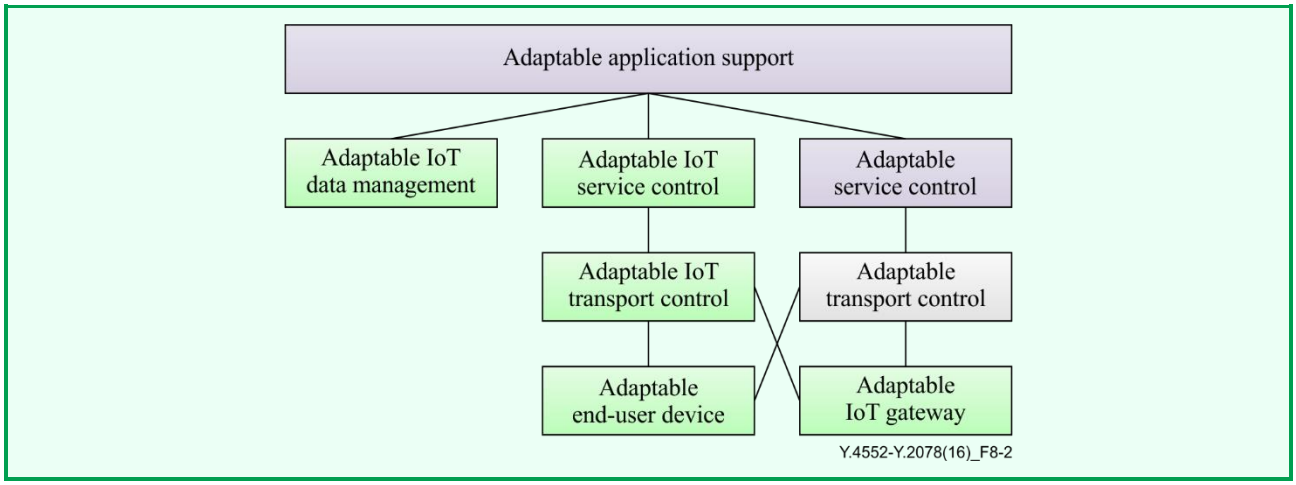


Figure 8-2 – The implementation view of the adaptable application support model

The adaptable application support entity is related to the adaptable IoT data management entity, the adaptable IoT service control entity and the adaptable service control entity, in order to allow exposure of their adaptable capabilities that can be accessed by IoT applications.

The adaptable IoT transport control entity is related to the adaptable end-user device entity and the adaptable IoT gateway entity in order to provide capabilities of adaptable communication and adaptable connectivity to fulfil adaptable requirements of IoT, such as adaptable event-based communication and adaptable identification-based connectivity.

The adaptable transport control entity is related to the adaptable end-user device entity and the adaptable IoT gateway entity in order to provide capabilities of adaptable communication and adaptable connectivity to fulfil general adaptable requirements, such as adaptable QoS enabling communication and adaptable device mobility.

The adaptable IoT transport control entity is related to the adaptable IoT service control entity in order to fulfil adaptable communication or connectivity requirements of IoT. The adaptable transport control entity is related to the adaptable service control entity in order to support IoT-independent adaptable communication or connectivity capabilities.

8.1.3 The deployment view of the adaptable application support model

The deployment view of the adaptable application support model consists of the adaptable IoT gateway component, the adaptable end-user device component, the adaptable IoT network controller component, the adaptable IoT service controller component, the adaptable IoT data server component, the adaptable service platform component and the interactions among these components as illustrated in Figure 8-3.

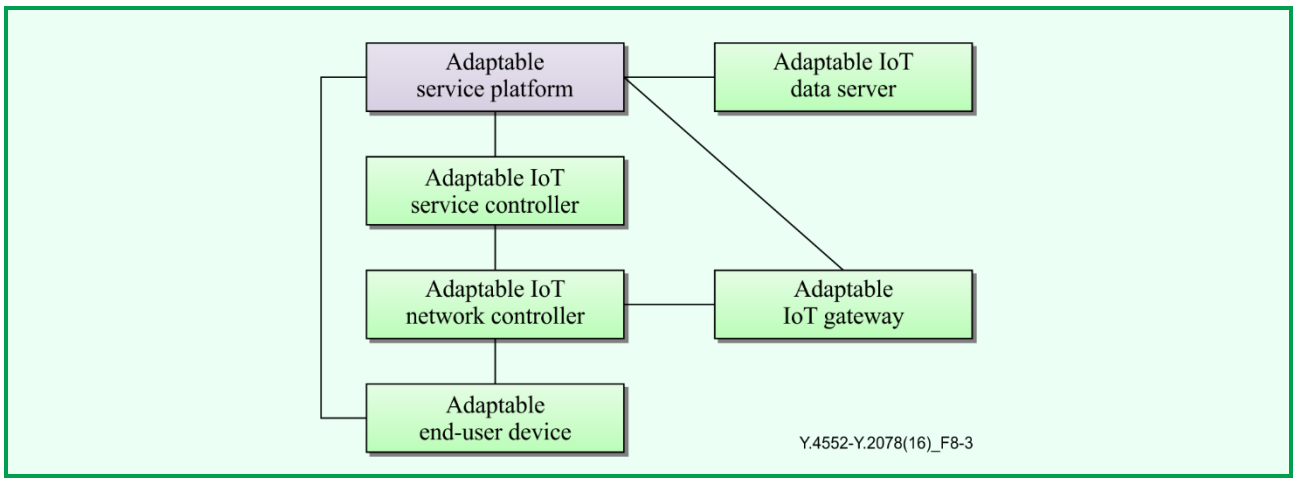


Figure 8-3 – The deployment view of the adaptable application support model

The adaptable service platform component is related to the adaptable end-user device component and the adaptable IoT gateway component in order to provide IoT-dependent adaptable application support capabilities, such as adaptable group management and adaptable orchestration capabilities.

NOTE – It is assumed that some capabilities contained in an adaptable service platform are deployed both in the adaptable end-user device component and in the adaptable IoT gateway component.

The adaptable service platform component is related to the adaptable IoT data server component in order to provide adaptable data management capabilities to IoT applications, such as the adaptable data processing capability.

The adaptable service platform component is related to the adaptable IoT service controller component in order to provide adaptable service provision capabilities to IoT applications, such as the adaptable service prioritization capability.

8.2 The capabilities of the adaptable application support model

Based on the categories of the IoT basic capabilities specified in [ITU-T Y.4401] and the functional view of the adaptable application support model specified in clause 8.1, the capabilities of the adaptable application support model can be classified into the following functional groups: adaptable service provision capabilities, adaptable communication capabilities, adaptable application support capabilities, adaptable data management capabilities and adaptable connectivity capabilities.

The capabilities of the adaptable application support model are specified from the perspective of the adaptable application support components as described in the deployment view of the adaptable application support model in clause 8.1.3, because these capabilities are implemented, deployed and used in these adaptable application support components.

NOTE 1 – IoT semantic capability is included in the capabilities of the adaptable application support model. IoT semantic capability facilitates the adaptable application support model's understanding of the meaning of IoT applications' service requests based on semantics.

NOTE 2 – The capability exposure capability is included in the capabilities of the adaptable application support model, specifically in the adaptable application support group. The capability exposure capability enables capabilities of the adaptable application support model to be discovered by IoT applications.

The following clauses describe, respectively, these capabilities of the adaptable application support model. These same capabilities are numbered and summarized in Annex B.

NOTE 3 – In the following clauses, the capability numbers, as shown in Annex B, are put between square brackets "[ ]" and inserted at the end of the description of the corresponding capability.

8.2.1 Adaptable service provision capabilities

The following IoT basic capabilities specified in [ITU-T Y.4401] provide adaptable service support abilities to IoT applications. They are part of the adaptable service provision capabilities of the IoT adaptable application support model. These capabilities include:

- Semantic based service capability, numbered as C-1-2 in [ITU-T Y.4401];
- Autonomic service capability, numbered as C-1-5 in [ITU-T Y.4401];
- Location based and context-aware service capability, numbered as C-1-6 in [ITU-T Y.4401]; and
- Adaptable service provision capability, numbered as C-1-11 in [ITU-T Y.4401].

In addition to the above capabilities, the following capabilities are part of the adaptable service provision capabilities.

The adaptable service prioritization capability enables the adaptable service platform and the adaptable IoT service controller to adjust services priorities, in order to adapt to differentiated services requirements from the IoT applications based on predefined rules [B-1-1].

The adaptable service composition capability enables the adaptable service platform and the adaptable service manager to adjust service creation or service customization based on the requirements of the IoT applications and predefined rules [B-1-2].

The adaptable mobility service capability enables the adaptable service platform and the adaptable service manager to adjust the mechanisms of remote service access, remote user authentication and remote service execution based on predefined rules [B-1-3].

8.2.2 Adaptable communication capabilities

The following IoT basic capabilities specified in [ITU-T Y.4401] provide adaptable communication abilities to IoT applications. They are part of the adaptable communication capabilities of the IoT adaptable application support model. These capabilities include:

- Content-aware communication, numbered as C-2-13 in [ITU-T Y.4401];
- Location-based communication, numbered as C-2-14 in [ITU-T Y.4401]; and
- Adaptable networking, numbered as C-2-16 in [ITU-T Y.4401].

In addition to the above capabilities, the following capabilities are part of the adaptable communication capabilities.

The adaptable event-based communication capability enables the adaptable service platform, the adaptable end-user devices and the adaptable IoT gateways to adjust the events for initiating communication based on predefined rules [B-2-1].

The adaptable quality of service enabling communication capability enables the adaptable network controller, the adaptable end-user devices and the adaptable IoT gateways to adjust the mechanisms according to current network status and predefined rules in order to guarantee the QoS required for the delivery and processing of data (e.g., time-sensible data) [B-2-2].

8.2.3 Adaptable application support capabilities

In addition to exposing the capabilities to IoT applications from other functional groups of the adaptable application support model, the following capabilities should be added in the adaptable application support capabilities.

The adaptable group management capability enables the adaptable service platform to create, modify, delete and query IoT groups, as well as to add, modify, delete and query IoT group members based on the requirements of the IoT applications and predefined rules [B-3-1].

The adaptable orchestration capability enables the adaptable service platform, the adaptable end-user devices and the adaptable IoT gateways to dynamically coordinate service provisioning based on the requirements of the IoT applications and predefined rules [B-3-2].

8.2.4 Adaptable data management capabilities

The following IoT basic capabilities specified in [ITU-T Y.4401] provide adaptable data management abilities to IoT applications. They are part of the adaptable data management capabilities of the IoT adaptable application support model. These capabilities include:

− Semantic data operation, numbered as C-4-6 in [ITU-T Y.4401]; and

− Autonomic data operation, numbered as C-4-7 in [ITU-T Y.4401].

In addition to the above capabilities, the following capabilities are part of the adaptable data management capabilities.

The adaptable data processing capability enables the adaptable IoT data server to adjust methods of data fusion and mining based on the IoT application requirements and predefined rules [B-4-1].

The adaptable information exchange capability enables the adaptable IoT data server to send data to or receive data from external data sources, e.g., data centres and data servers outside the IoT, based on the IoT application requirements and predefined rules [B-4-2].

8.2.5 Adaptable connectivity capabilities

The following IoT basic capability specified in [ITU-T Y.4401] provides adaptable connectivity abilities to IoT applications. The following capability is part of the adaptable connectivity capabilities of the IoT adaptable application support model:

− Adaptable connectivity, numbered as C-6-4 in [ITU-T Y.4401].

In addition to the above capability, the following capabilities are part of the adaptable connectivity capabilities:

The adaptable identification based connectivity capability enables the adaptable network manager, the adaptable end-user devices and the adaptable IoT gateways to dynamically choose the mechanisms for establishing the connectivity based on the identification of things and predefined rules [B-5-1].

The adaptable device mobility capability enables the adaptable network manager, the adaptable end-user devices and the adaptable IoT gateways to dynamically negotiate the mechanisms for keeping connectivity when the adaptable end-user devices or the adaptable IoT gateways are moving based on predefined rules [B-5-2].

9 The reliable application support model

The reliable application support model consists of the functional view, implementation view and deployment view of descriptions on the reliable application support model and related capabilities.

NOTE 1 – The three views of application support model can be used to derive and validate the capabilities for support of reliable applications of the IoT.

NOTE 2 – The degrees of reliability that may be realized in an IoT implementation will depend on application requirements and resource management. The definitions and specifications of the degrees of reliability are out of the scope of this Recommendation.

9.1 The description of the high reliable application support model

9.1.1 The functional view of the reliable application support model

The functional view of the reliable application support model consists of the reliable management group, the reliable data management group, the reliable service provision group, the reliable communication group, the reliable connectivity group, the reliable application support group and the interactions among these groups as illustrated in Figure 9-1. Each functional group contains related capabilities for support of the IoT reliable applications.

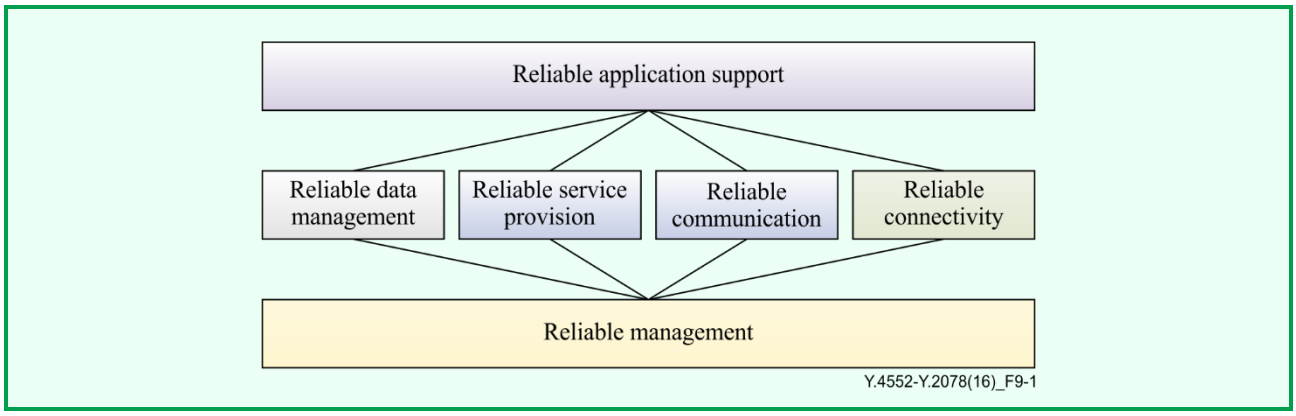


Figure 9-1 – The functional view of the reliable application support model

The reliable application support group is related to the reliable data management group, the reliable service provision group, the reliable communication group and the reliable connectivity group, in order to allow exposure of their reliable capabilities to IoT applications.

The reliable management group is related to the reliable data management group, the reliable service provision group, the reliable communication group and the reliable connectivity group, in order to provide management capabilities to support additional reliability requirements of IoT applications, such as reliable service integrity check capability and reliable data integrity check capability.

NOTE – The concepts of security and privacy protection are related to the concepts of reliability. In order to ensure that security and privacy protection are realized, some reliable support mechanisms are required for the implementation and deployment of the security and privacy protection capabilities. Based on the requirements that the security and privacy protection capability be isolated from other capabilities during implementation and deployment, these reliable support mechanisms are required to be implemented and deployed in self-sustained functional components. According to these considerations, the reliable application support model does not contain capabilities of the security and privacy protection group.

9.1.2 The implementation view of the reliable application support model

The implementation view of the reliable application support model consists of the reliable management and identity management entity, the reliable IoT gateway entity, the reliable end-user device entity, the reliable transport control entity, the reliable IoT transport control entity, the reliable service control entity, the reliable IoT service control entity, the reliable IoT data management entity, the reliable application support entity and the interactions among these entities as illustrated in Figure 9-2.

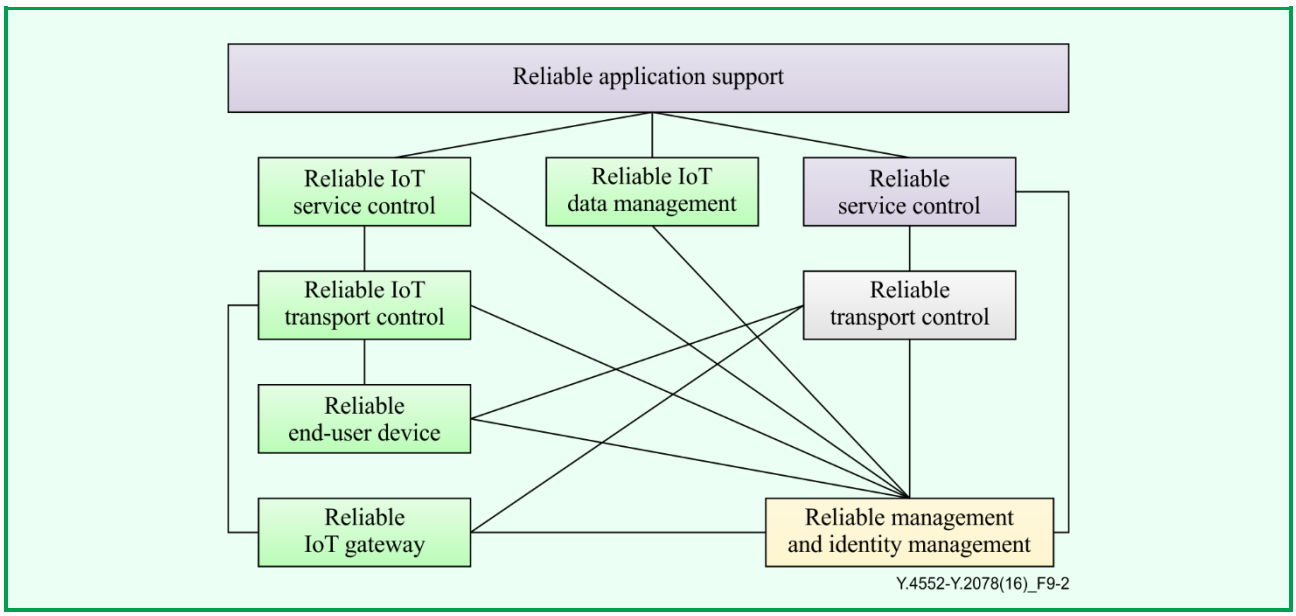


Figure 9-2 – The implementation view of the reliable application support model

The reliable application support entity is related to the reliable IoT data management entity, the reliable IoT service control entity and the reliable service control entity, in order to allow exposure of their reliable capabilities so that can be accessed by IoT applications, such as reliable programming interface capability.

The reliable management and identity management entity is related to the reliable IoT service control entity, the reliable IoT transport control entity, the reliable end-user device entity, the reliable IoT gateway entity, the reliable service control entity and the reliable transport control entity, in order to implement additional reliability features by management capabilities, such as the reliable distributed processing capability.

The reliable end-user device entity and reliable IoT gateway entity are related both to the reliable transport control entity and to the reliable IoT transport control entity, in order to implement reliable communication and connectivity capabilities, such as the reliable periodic communication capability and the reliable identification-based connectivity capability.

9.1.3 The deployment view of the reliable application support model

The deployment view of the reliable application support model consists of the reliable device manager component, the reliable IoT gateway component, the reliable end-user device component, the reliable network manager component, the reliable IoT network controller component, the reliable service manager component, the reliable IoT service controller component, the reliable IoT data server component, the reliable service platform component and the interactions among these components as illustrated in Figure 9-3.

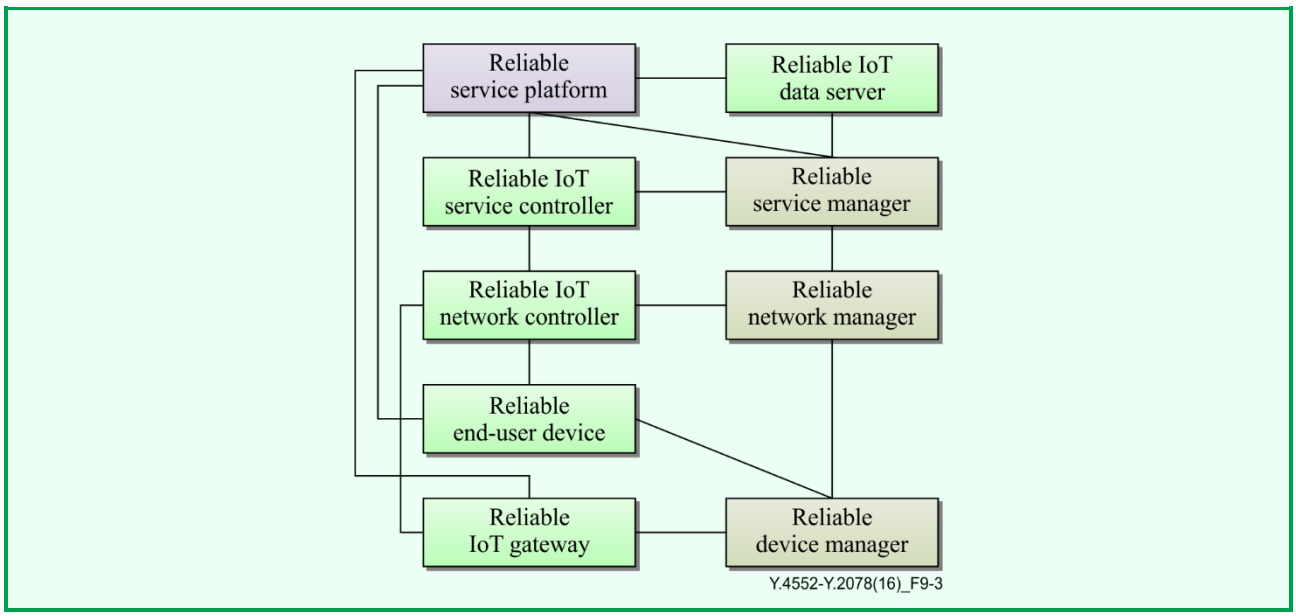


Figure 9-3 – The deployment view of the reliable application support model

The reliable service platform component is related to the reliable end-user device component and the reliable IoT gateway component to provide reliable application support capabilities, such as the reliable group management capability.

NOTE – It is assumed that IoT applications are required to be executed in the reliable end-user device component and the reliable IoT gateway component.

The reliable IoT data server component is related to the reliable service platform component in order to provide reliable data management capabilities, such as the reliable information exchange capability.

The reliable service platform component is related to the reliable service manager component, the reliable service manager component is related to the reliable network manager component and the reliable network manager component is related to the reliable device manager component, in order to provide integrated reliable management capabilities, such as the reliable management capability for multiple domains as defined in [ITU-T Y.4401].

9.2 The capabilities of the reliable application support model

Based on the categories of the IoT basic capabilities specified in [ITU-T Y.4401] and the functional view of the reliable application support model specified in clause 9.1, the capabilities of the reliable application support model can be classified into the following functional groups: reliable service provision capabilities, reliable communication capabilities, reliable application support capabilities, reliable data management capabilities, reliable management capabilities and reliable connectivity capabilities.

The capabilities of the reliable application support model are specified from the perspective of reliable application support components as described in the deployment view of the reliable application support model in clause 9.1.3, because these capabilities are implemented, deployed and used in these reliable application support components.

The following clauses describe, respectively, these capabilities of the reliable application support model. These same capabilities are numbered and summarized in Annex C.

NOTE – In the following clauses, the capability numbers, as shown in Annex C, are put between square brackets "[ ]" and inserted at the end of the description of the corresponding capability.

9.2.1 Reliable service provision capabilities

The reliable service provision capabilities extend the service provision capabilities of the IoT basic capabilities specified in [ITU-T Y.4401], in order to support reliable service provisioning to the IoT applications.

The reliable service provision capabilities include the reliable service prioritization capability, the reliable service composition capability, the reliable mobility service capability, the reliable autonomic service capability and the reliable naming and addressing capability.

The reliable service prioritization capability enables the reliable service platform and the reliable IoT service controller to guarantee services priorities, in order to provide reliable differentiated services to the IoT applications [C-1-1].

The reliable service composition capability enables the reliable service platform and the reliable service manager to guarantee correct service creation or service customization based on the requirements of the IoT applications [C-1-2].

The reliable mobility service capability enables the reliable service platform and the reliable service manager to guarantee correct remote service access, remote user authentication and remote service execution based on the IoT application requirements [C-1-3].

The reliable autonomic service capability enables the reliable service platform, the reliable service manager and the reliable IoT service controller to guarantee automatic capturing, transferring and analyzing data of things and automatic provision of services in correct ways based on predefined rules [C-1-4].

The reliable naming and addressing capability enables the reliable service manager, the reliable network manager and the reliable device manager to guarantee creating, updating, deleting, querying names and addresses of users, devices and things in correct ways based on predefined rules [C-1-5].

In addition to the above capabilities, the service provision acknowledgement capability numbered as C-1-12 in [ITU-T Y.4401] is also part of reliable service provision capabilities.

9.2.2 Reliable communication capabilities

The reliable communication capabilities extend the communication capabilities of the IoT basic capabilities specified in [ITU-T Y.4401], in order to support reliable communication for the IoT applications.

The reliable communication capabilities include the reliable event-based communication capability, the reliable periodic communication capability and the reliable quality of service enabling communication capability.

The reliable event-based communication capability enables the reliable service platform, the reliable end-user devices and the reliable IoT gateways to guarantee correct ways of initiating communication based on predefined events and rules [C-2-1].

The reliable periodic communication capability enables the reliable service platform, the reliable end-user devices and the reliable IoT gateways to guarantee correct ways of periodically initiating communication based on predefined rules [C-2-2].

The reliable quality of service enabling communication capability enables the reliable network controller, the reliable end-user devices and the reliable IoT gateways to guarantee the QoS required for the delivery and processing of data (e.g., time-sensible data) in correct ways [C-2-3].

In addition to the above capabilities, the transport acknowledgement capability numbered as C-2-15 in [ITU-T Y.4401] is also part of reliable communication capabilities.

9.2.3 Reliable application support capabilities

The reliable application support capabilities extend the application support capabilities of the IoT basic capabilities specified in [ITU-T Y.4401], in order to provide reliable application support to the IoT applications.

The reliable application support capabilities include the reliable programmable interface provision capability, the reliable group management capability, the reliable time synchronization capability and the reliable user management capability.

The reliable programmable interface provision capability enables the reliable service platform to guarantee correct ways of providing or customizing services making use of existing capabilities based on the IoT application requirements [C-3-1].

The reliable group management capability enables the reliable service platform to guarantee correct ways of creating, modifying, deleting and querying IoT groups and of adding, modifying, deleting and querying IoT group members based on the requirements of the IoT applications and predefined rules [C-3-2].

The reliable time synchronization capability enables the reliable service platform to guarantee correct ways of synchronizing the time among related functional components, in order to support global or local time stamping for the IoT applications [C-3-3].

The reliable orchestration capability enables the reliable service platform, the reliable end-user devices and the reliable IoT gateways to guarantee correct ways of coordinating service provisioning based on the requirements of the IoT applications and predefined rules [C-3-4].

The reliable user management capability enables the reliable service platform to guarantee correct ways of creating, querying, updating and deleting IoT user profiles and of authenticating, authorizing, registering and auditing IoT users based on predefined rules [C-3-5].

In addition to the above capabilities, the application support operation acknowledgement capability numbered as C-3-6 in [ITU-T Y.4401], is also part of reliable application support capabilities.

9.2.4 Reliable data management capabilities

The reliable data management capabilities extend the data management capabilities of the IoT basic capabilities specified in [ITU-T Y.4401], in order to support reliable data management to the IoT applications.

The reliable data management capabilities include the reliable data processing capability, the reliable information exchange capability and the reliable autonomic data operation capability.

The reliable data processing capability enables the reliable IoT data server to guarantee trustable results of data fusion and mining based on the IoT application requirements and predefined rules [C-4-1].

The reliable information exchange capability enables the reliable IoT data server to guarantee correct ways of sending data to or receiving data from external data sources, e.g., data centres and data servers outside the IoT, based on the IoT application requirements and predefined rules [C-4-2].

The reliable autonomic data operation capability enables the reliable IoT data server, the reliable end-user devices and the reliable IoT gateways, to guarantee correct ways of automatically collecting, aggregating, transferring, storing, analyzing data of things, as well as automatically managing these data operations based on the IoT application requirements and predefined rules [C-4-3].

9.2.5 Reliable management capabilities

The reliable management capabilities extend the management capabilities of the IoT basic capabilities specified in [ITU-T Y.4401], in order to support reliable management to the IoT applications.

The reliable management capabilities include the reliable distributed processing management capability, the reliable multi-domain management capability, the reliable service integrity check capability, the reliable data integrity check capability, the reliable device integrity check capability, the reliable security integrity check capability and the reliable user profile integrity check capability.

The reliable distributed processing management capability enables the reliable IoT data server, the reliable service manager, the reliable network manager and the reliable device manager to guarantee correct ways of managing IoT functional components in a distributed way based on predefined rules [C-5-1].

The reliable multi-domain management capability enables the reliable IoT data server, the reliable service manager, the reliable network manager and the reliable device manager to guarantee correct ways of managing IoT functional components in multiple administrative domains [ITU-T Y.4401] based on predefined rules [C-5-2].

The reliable service integrity check capability enables the reliable service manager to guarantee trustable ways of checking service lifetime and available resources required to provide the service, in order to guarantee a certain degree of service provision availability based on the IoT application requirements [C-5-3].

The reliable data integrity check capability enables the reliable IoT data server to guarantee trustable ways of checking data lifetime, available attributes of data and consistency of data in order to guarantee a certain degree of availability of data management based on the IoT application requirements [C-5-4].

The reliable device integrity check capability enables the reliable device manager, the reliable IoT gateway and the reliable end-user device to guarantee trustable ways of checking the status of all device functionalities in order to guarantee a certain degree of device availability based on the IoT application requirements [C-5-5].

The reliable security integrity check capability enables the reliable service manager, the reliable network manager, the reliable device manager and the reliable IoT data server to guarantee trustable ways of checking the consistency of security policies deployed in all functional components of the IoT, in order to guarantee a certain degree of security availability in the IoT based on the IoT application requirements [C-5-6].

The reliable user profile integrity check capability enables the reliable service manager, the reliable network manager, the reliable device manager and the reliable IoT data server to guarantee trustable ways of checking lifetime, subscription, privacy protection and availability of services subscribed by users, in order to guarantee a certain degree of availability of service provisioning and privacy protection for users based on the IoT application requirements [C-5-7].

In addition to the above capabilities, the redundant deployment enablement capability numbered by C-5-9 in [ITU-T Y.4401] is also part of reliable management capabilities.

9.2.6 Reliable connectivity capabilities

The reliable connectivity capabilities extend the connectivity capabilities of the IoT basic capabilities specified in [ITU-T Y.4401], in order to support reliable connectivity to the IoT applications.

The reliable connectivity capabilities include reliable identification based connectivity capability and reliable device mobility capability.

The reliable identification based connectivity capability enables the reliable network manager, the reliable end-user devices and the reliable IoT gateways to guarantee correct ways of establishing the connectivity based on the identification of things and predefined rules [C-6-1].

The reliable device mobility capability enables the reliable network manager, the reliable end-user devices and the reliable IoT gateways to guarantee correct ways of keeping the connectivity when the reliable end-user devices or the reliable IoT gateways are moving based on predefined rules [C-6-2].

10 Security considerations

Security is one of the fundamental aspects to be considered in the IoT application support models. This Recommendation considers the issues of security and privacy protection both from the perspective of the IoT application support models' description and of the capabilities of the IoT application support models' capabilities.

The issues of security and privacy protection from the perspective of the IoT application support models' description are considered, respectively, in clause 7.1 for the IoT configurable application support model, in clause 8.1 for the IoT adaptable application support model and in clause 9.1 for the IoT reliable application support model.

The issues of security and privacy protection from the perspective of the IoT configurable application support model capabilities are considered in clause 7.2.5. Concerning security and privacy protection with respect to the adaptable application support model and the reliable application support model, these issues are considered, respectively, in clauses 8.1.1 and 9.1.1.

Annex A

The list of configurable capabilities for support of IoT applications

(This annex forms an integral part of this Recommendation.)

The tables in this annex list and number the configurable capabilities identified in this Recommendation for support of IoT applications.

All tables in this annex have the following format:

– The first column of these tables is named as "capability number" and assigns a number to each IoT capability. The numbering rule for each IoT capability is as follows: A-<the sub-clause number of clause 7.2>-<the sequence number of each configurable capability in each sub-clause>. For example, the first configurable capability described in clause 7.2.1 is numbered as A-1-1.

– The second column of these tables is named as "capability name" and gives the name of each configurable capability.

– The third column of these tables is named as "capability summary" and briefly describes what the capability does.

– The fourth column of these tables is named as "related basic capabilities" and describes the IoT basic capabilities specified in [ITU-T Y.4401] that are related to the configurable capability.

NOTE 1 – One configurable capability may be related to one or several IoT basic capabilities.

The fifth column of these tables is named as "associated components" and lists the functional components of the deployment view of the configurable application support model described in clause 7.1 that are associated with the configurable capability. This column can be used to validate that the configurable capability can be implemented and deployed.

NOTE 2 – For the purpose of simplification, the prefixed "configurable" is omitted for the associated components naming in the tables.

Table A.1 shows the list of configurable service provision capabilities.

Table A.1 – List of configurable service provision capabilities

Capability number	Capability name	Capability summary	Related basic capabilities	Associated components
A-1-1	Configurable service prioritization	The configurable service prioritization capability enables the IoT applications to configure services in different priorities, in order to provide differentiated services based on their requirements.	Service prioritization numbered as C-1-1 in [ITU-T Y.4401]	Service platform, IoT service controller

Table A.1 – List of configurable service provision capabilities

Capability number	Capability name	Capability summary	Related basic capabilities	Associated components
A-1-2	Configurable service composition	The configurable service composition capability enables the IoT applications to configure service creation or service customization based on their requirements.	Service composition numbered as C-1-3 in [ITU-T Y.4401].	Service platform, service manager
A-1-3	Configurable location based and context aware service	The configurable location based and context-aware service capability enables the IoT applications to configure services that are provided both on the location information and related context, and on the predefined rules or policies, in order to fulfil their requirements.	Location based and context-aware service numbered as C-1-6 in [ITU-T Y.4401].	Service platform, service manager, IoT service controller

Table A.2 shows the list of configurable communication capabilities.

Table A.2 – List of configurable communication capabilities

Capability number	Capability name	Capability summary	Related basic capabilities	Associated components
A-2-1	Configurable event-based communication	The configurable event-based communication capability enables the IoT applications to configure different events in order to initiate communication based on the requirements of the IoT applications.	Event-based communication numbered as C-2-1 in [ITU-T Y.4401].	IoT gateway, end-user device, service platform
A-2-2	Configurable periodic communication	The configurable periodic communication capability enables the IoT applications to configure the rules in order to periodically initiate communication based on the requirements of the IoT applications.	Periodic communication numbered as C-2-2 in [ITU-T Y.4401].	IoT gateway, end-user device, service platform

Table A.2 – List of configurable communication capabilities

Capability number	Capability name	Capability summary	Related basic capabilities	Associated components
A-2-3	Configurable communication mode	The configurable communication mode capability enables the IoT applications to configure different modes of communications in transport network in order to transfer data from the source(s) to the destination(s) based on the requirements of the IoT applications.	Unicast communication numbered as C-2-3 in [ITU-T Y.4401]. Multicast communication numbered as C-2-4 in [ITU-T Y.4401]. Broadcast communication numbered as C-2-5 in [ITU-T Y.4401]. Anycast communication numbered as C-2-6 in [ITU-T Y.4401].	Network manager
A-2-4	Configurable Quality of Service communication	The configurable quality of service communication capability enables the IoT applications to configure the related mechanisms in order to guarantee the delivery and process the time-sensible data based on the requirements of the IoT applications.	Quality of service enabling communication numbered as C-2-8 in [ITU-T Y.4401].	IoT gateway, end-user device, IoT network controller
A-2-5	Configurable content-aware communication	The configurable content-aware communication capability enables the IoT applications to configure the parameters related with content and selected path for routing or blocking data transfer based on the requirements of the IoT applications.	Content-aware communication numbered as C-2-13 in [ITU-T Y.4401].	IoT gateway, end-user device, IoT network controller
A-2-6	Configurable location based communication	The configurable location based communication capability enables the IoT applications to configure the parameters related with locations and predefined rules in order to initiate communication based on the requirements of the IoT applications.	Location based communication numbered as C-2-14 in [ITU-T Y.4401].	IoT gateway, end-user device, IoT network controller

Table A.3 shows the list of configurable data management capabilities.

Table A.3 – List of configurable data management capabilities

Capability number	Capability name	Capability summary	Related basic capabilities	Associated components
A-3-1	Configurable data storage	The configurable data storage capability enables the IoT applications to configure rules or policies for storing data based on the requirements of the IoT applications.	Data storage numbered as C-4-1 in [ITU-T Y.4401].	IoT data server, IoT gateway
A-3-2	Configurable data processing	The configurable data processing capability enables the IoT applications to configure the rules or the policies for processing data based on the requirements of the IoT applications.	Data processing numbered as C-4-2 in [ITU-T Y.4401].	IoT data server
A-3-3	Configurable information exchange	The configurable information exchange capability enables the IoT applications to configure the parameters for sending data to or receiving data from external data sources, e.g., data centres and data servers outside the IoT based on the requirements of the IoT applications.	Open information exchange numbered as C-4-5 in [ITU-T Y.4401].	IoT data server
A-3-4	Configurable semantic data operation	The configurable semantic data operation capability enables the IoT applications to configure the parameters for semantic annotating, semantic discovering, semantic storing and semantic composition of data of things based on the requirements of the IoT applications.	Semantic data operation numbered as C-4-6 in [ITU-T Y.4401].	IoT data server, IoT gateway
A-3-5	Configurable autonomic data operation	The configurable autonomic data operation capability enables the IoT applications to configure the parameters for automatically collecting, aggregating, transferring, storing, analyzing data of things, as well as automatically managing these data operations based on the requirements of the IoT applications.	Autonomic data operation numbered as C-4-7 in [ITU-T Y.4401].	IoT gateway, end-user device, IoT data server

Table A.4 shows the list of configurable connectivity capabilities.

Table A.4 – List of configurable connectivity capabilities

Capability number	Capability name	Capability summary	Related basic capabilities	Associated components
A-4-1	Configurable identification based connectivity	The configurable identification based connectivity capability enables the IoT applications to configure the parameters for connectivity establishment based on the identification of things and the requirements of the IoT applications.	Identification based connectivity numbered as C-6-1 in [ITU-T Y.4401].	IoT gateway, end-user device, network manager
A-4-2	Configurable things' status notification	The configurable things' status notification capability enables the IoT applications to configure the rules of automatic notification of the status of things and its changes based on the requirements of the IoT applications.	Things' status notification numbered as C-6-2 in [ITU-T Y.4401].	IoT gateway, end-user device
A-4-3	Configurable device mobility	The configurable device mobility capability enables the IoT applications to configure the parameters for maintaining the connectivity with the IoT when end-user devices or IoT gateways are moving, based on the requirements of the IoT applications.	Device mobility numbered as C-6-3 in [ITU-T Y.4401].	IoT gateway, end-user device, network manager
A-4-4	Configurable and adaptable connectivity	The configurable and adaptable connectivity capability enables the IoT applications to configure the parameters for extending connectivity configurations to connect with different types of devices of the IoT based on the requirements of the IoT applications, in order to be adaptable to different technologies in devices of IoT.	Adaptable connectivity numbered as C-6-4 in [ITU-T Y.4401].	IoT gateway, end-user device, device manager

Table A.5 shows the list of configurable communication capabilities.

Table A.5 – List of configurable communication capabilities

Capability number	Capability name	Capability summary	Related basic capabilities	Associated components
A-5-1	Configurable communication security	The configurable communication security capability enables the IoT applications to configure the rules and policies for supporting secure, trusted and privacy protected communication based on the requirements of the IoT applications.	Communication security numbered as C-7-1 in [ITU-T Y.4401].	IoT gateway, end-user device, device manager, network manager, enhanced transport network
A-5-2	Configurable data management security	The configurable data management security capability enables the IoT applications to configure the rules and policies for providing secure, trusted and privacy protected data management based on the requirements of the IoT applications.	Data management security numbered as C-7-2 in [ITU-T Y.4401].	IoT data server, IoT gateway
A-5-3	Configurable service provision security	The configurable service provision security capability enables the IoT applications to configure the rules and policies for providing secure, trusted and privacy protected service provision based on the requirements of the IoT applications.	Service provision security numbered as C-7-3 in [ITU-T Y.4401].	Service platform, service manger
A-5-4	Configurable security integration	The configurable security integration capability enables the IoT applications to configure the rules and policies for enabling integration of different security policies and techniques related to IoT functional components based on the requirements of the IoT applications.	Security integration numbered as C-7-4 in [ITU-T Y.4401].	Device manager, network manager, service manager

Table A.5 – List of configurable communication capabilities

Capability number	Capability name	Capability summary	Related basic capabilities	Associated components
A-5-5	Configurable mutual authentication and authorization	The configurable mutual authentication and authorization capability enables the IoT applications to configure the rules and policies for authenticating and authorizing IoT applications and devices before a device accesses IoT based on the requirements of the IoT applications.	Mutual authentication and authorization numbered as C-7-5 in [ITU-T Y.4401].	Device manager, network manager

Table A.6 shows the list of configurable application support capabilities.

Table A.6 – List of configurable application support capabilities

Capability number	Capability name	Capability summary	Related basic capabilities	Associated components
A-6-1	Configurable group management	The configurable group management capability enables the IoT applications to configure the parameters for creating, modifying, deleting, and querying IoT groups, as well as adding, modifying, deleting and querying IoT group members, based on the requirements of the IoT applications.	Group management numbered as C-3-2 in [ITU-T Y.4401].	Service platform
A-6-2	Configurable time synchronization	The configurable time synchronization capability enables the IoT applications to configure the parameters for synchronizing the time among related functional components with different degrees of reliability, in order to support global or local time stamping for applications based on the different Quality of Service requirements of the IoT applications.	Time synchronization numbered as C-3-3 in [ITU-T Y.4401].	Service platform

Table A.6 – List of configurable application support capabilities

Capability number	Capability name	Capability summary	Related basic capabilities	Associated components
A-6-3	Configurable orchestration	The configurable orchestration capability enables the IoT applications to configure the parameters for automatic coordination of service provisioning or device operations based on the requirements of the IoT applications.	Orchestration numbered as C-3-4 in [ITU-T Y.4401].	IoT gateway, end-user device, service platform
A-6-4	Configurable application support operation acknowledgement	The configurable application support operation acknowledgement capability enables the IoT applications to configure the parameters for acknowledging the correct operations requested by applications in order to support reliable application operations in the IoT, based on the requirements of the IoT applications.	Application support operation acknowledgement numbered as C-3-6 in [ITU-T Y.4401].	IoT data server, IoT gateway

Table A.7 shows the list of configurable management capabilities.

Table A.7 – List of configurable management capabilities

Capability number	Capability name	Capability summary	Related basic capabilities	Associated components
A-7-1	Configurable redundant deployment enablement	The configurable redundant deployment enablement capability enables the IoT applications to configure deployment of redundant functional components of the IoT in order to provide different degrees of reliability required in communication, service provision and data management, based on the requirements of the IoT applications.	Redundant deployment enablement numbered as C-5-9 in [ITU-T Y.4401].	IoT data server, IoT service controller, service platform, service manager, IoT network controller, network manager, IoT gateway, device manager

Table A.7 – List of configurable management capabilities

Capability number	Capability name	Capability summary	Related basic capabilities	Associated components
A-7-2	Configurable service integrity check	The configurable service integrity check capability enables the IoT applications to configure the parameters for checking the service lifetime, the available resources required to provide the service in order to provide different degrees of availability in service provisioning, based on the requirements of the IoT applications.	Service integrity check numbered as C-5-10 in [ITU-T Y.4401].	Service manager
A-7-3	Configurable data integrity check	The configurable data integrity check capability enables the IoT applications to configure the parameters for checking the data lifetime, the available attributes of the data, and the consistency of data in order to provide different degrees of availability in data management, based on the requirements of the IoT applications.	Data integrity check numbered as C-5-11 in [ITU-T Y.4401].	IoT data server
A-7-4	Configurable device integrity check	The configurable device integrity check capability enables the IoT applications to configure the parameters for checking the status of all device functions in order to provide different degrees of availability in IoT devices, based on the requirements of the IoT applications.	Device integrity check as C-5-12 in [ITU-T Y.4401].	Device manager, IoT device, IoT gateway, end-user device

Table A.7 – List of configurable management capabilities

Capability number	Capability name	Capability summary	Related basic capabilities	Associated components
A-7-5	Configurable security integrity check	The configurable security integrity check capability enables the IoT applications to configure the parameters for checking the consistency of security policies deployed in all functional components of the IoT in order to provide different degrees of availability in security and privacy protection provisioning, based on the requirements of the IoT applications.	Security integrity check numbered as C-5-13 in [ITU-T Y.4401].	Service manager, network manager, device manager, IoT device, IoT gateway, end-user device
A-7-6	Configurable user profile integrity check	The configurable user profile integrity check capability enables the IoT applications to configure the parameters for checking the lifetime, subscription, privacy protection, and availability of services subscribed by users in order to provide different degrees of availability in service provisioning and privacy protection for users, based on the requirements of the IoT applications.	User profile integrity check numbered as C-5-14 in [ITU-T Y.4401].	Service manager, network manager, device manager, IoT device, IoT gateway, end-user device

Annex B

The list of adaptable capabilities for support of IoT applications

(This annex forms an integral part of this Recommendation.)

The following table lists and numbers the adaptable capabilities identified in this Recommendation for support of IoT applications.

The table in this annex has the following format:

– The first column of the table is named as "capability number" and assigns a number to each IoT capability. The numbering rule for each IoT capability is as follows: B-<the sub-clause number of clause 8.2>-<the sequence number of each adaptable capability in each sub-clause>. For example, the first adaptable capability described in clause 8.2.1 is numbered as B-1-1.

– The second column of the table is named as "capability name" and gives the name of each adaptable capability.

– The third column of the table is named as "capability summary" and shortly describes what the capability does.

– The fourth column of the table is named as "related basic capabilities" and describes the IoT basic capabilities specified in [ITU-T Y.4401] that are related with the adaptable capability.

– The fifth column of the table is named as "associated components" and lists the functional components of the deployment view of the adaptable application support model described in clause 8.1 that are associated with the adaptable capability. This column can be used to validate that the adaptable capability can be implemented and deployed.

NOTE – For simplification purpose, the prefixed "adaptable" is omitted for the associated components naming in the tables.

Table B.1 shows the list of adaptable application support capabilities.

Table B.1 – List of adaptable application support capabilities

Capability number	Capability name	Capability summary	Related basic capabilities	Associated components
B-1-1	Adaptable service prioritization	The adaptable service prioritization capability enables the service platform and the IoT service controller to adjust services priorities, in order to adapt to differentiated services requirements from the IoT applications.	Service prioritization numbered as C-1-1 in [ITU-T Y.4401]	Service platform, IoT service controller

Table B.1 – List of adaptable application support capabilities

Capability number	Capability name	Capability summary	Related basic capabilities	Associated components
B-1-2	Adaptable service composition	The adaptable service composition capability enables the service platform and the service manager to adjust service creation or service customization based on the requirements of the IoT applications.	Service composition numbered as C-1-3 in [ITU-T Y.4401].	Service platform, service manager
B-1-3	Adaptable mobility service	The adaptable mobility service capability enables the service platform to adjust the mechanism of remote service access, remote user authentication, and remote service execution.	Mobility service numbered as C-1-4 in [ITU-T Y.4401].	Service platform, service manager
B-2-1	Adaptable event-based Communication	The adaptable event-based communication capability enables the service platform, the end-user devices, and the IoT gateways to adjust the events for initiating communication based on predefined rules.	Event-based communication numbered as C-2-1 in [ITU-T Y.4401]	IoT device, IoT gateway, end-user device, service platform
B-2-2	Adaptable Quality of Service enabling communication	The adaptable Quality of Service enabling communication capability enables the network controller, the end-user devices, and the IoT gateways to adjust the mechanisms according to current network status and predefined rules to guarantee the Quality of Service required for the delivery and processing of data (e.g., time-sensible data).	Quality of service enabling communication numbered as C-2-8 in [ITU-T Y.4401]	IoT device, IoT gateway, end-user device, IoT network controller

Table B.1 – List of adaptable application support capabilities

Capability number	Capability name	Capability summary	Related basic capabilities	Associated components
B-3-1	Adaptable group management	The adaptable group management capability enables the service platform to create, modify, delete, and query IoT groups, as well as to add, modify, delete and query IoT group members based on the requirements of the IoT applications and predefined rules.	Group management numbered as C-3-2 in [ITU-T Y.4401]	Service platform
B-3-2	Adaptable orchestration	The adaptable orchestration capability enables the service platform, the end-user devices, and the IoT gateways to dynamic coordinate service provisioning or device operations based on the requirements of the IoT applications and predefined rules.	Orchestration numbered as C-3-4 in [ITU-T Y.4401]	IoT device, IoT gateway, end-user device, service platform
B-4-1	Adaptable data processing	The adaptable data processing capability enables the IoT data server to adjust methods of data fusion and mining based on the IoT application requirements and predefined rules.	Data processing numbered as C-4-2 in [ITU-T Y.4401]	IoT data server
B-4-2	Adaptable information exchange	The adaptable information exchange capability enables the IoT data server to autonomously send data to or receive data from external data sources, e.g., data centres and data servers outside the IoT, based on the IoT application requirements and predefined rules.	Open information exchange numbered as C-4-5 in [ITU-T Y.4401]	IoT data server

Table B.1 – List of adaptable application support capabilities

Capability number	Capability name	Capability summary	Related basic capabilities	Associated components
B-5-1	Adaptable identification based connectivity	The adaptable identification based connectivity capability enables the network manager, the end-user devices, and the IoT gateways to dynamic choose the ways of establishing the connectivity based on the identification of things and predefined rules.	Identification based connectivity numbered as C-6-1 in [ITU-T Y.4401]	IoT device, IoT gateway, end-user device, network manager
B-5-2	Adaptable device mobility	The adaptable device mobility capability enables the network manager, the end-user devices, and the IoT gateways to dynamic negotiate the ways of keeping the connectivity when the IoT devices or the IoT gateways are moving based on predefined rules.	Device mobility numbered as C-6-3 in [ITU-T Y.4401]	IoT device, IoT gateway, end-user device, network manager

Annex C

The list of reliable capabilities for support of IoT applications

(This annex forms an integral part of this Recommendation.)

The following table lists and numbers the reliable capabilities identified in this Recommendation for support of IoT applications.

The table in this annex has the following format:

– The first column of the table is named as "capability number" and assigns a number to each IoT capability. The numbering rule for each IoT capability is as follows: C-<the sub-clause number of clause 9.2>-<the sequence number of each reliable capability in each sub-clause>. For example, the first reliable capability described in clause 9.2.1 is numbered as C-1-1.

– The second column of the table is named as "capability name" and gives the name of each reliable capability.

– The third column of the table is named as "capability summary" and shortly describes what the capability does.

– The fourth column of the table is named as "related basic capabilities" and describes the IoT basic capabilities specified in [ITU-T Y.4401] that are related with the reliable capability.

– The fifth column of the table is named as "associated components" and lists the functional components of the deployment view of the reliable application support model described in clause 9.1 that are associated with the reliable capability. This column can be used to validate that the reliable capability can be implemented and deployed.

NOTE – For simplification purpose, the prefixed "reliable" is omitted for the associated components naming in the tables.

Table C.1 shows the list of reliable application support capabilities.

Table C.1 – List of reliable application support capabilities

Capability number	Capability name	Capability summary	Related basic capabilities	Associated components
C-1-1	Reliable service prioritization	The reliable service prioritization capability enables service platform and IoT service controller to guarantee services priorities, in order to provide reliable differentiated services to the IoT applications.	Service prioritization numbered as C-1-1 in [ITU-T Y.4401]	Service platform, IoT service controller

Table C.1 – List of reliable application support capabilities

Capability number	Capability name	Capability summary	Related basic capabilities	Associated components
C-1-2	Reliable service composition	The reliable service composition capability enables service platform and service manager to guarantee correct service creation or service customization based on the requirements of the IoT applications.	Service composition numbered as C-1-3 in [ITU-T Y.4401].	Service platform, service manager
C-1-3	Reliable mobility service	The reliable mobility service capability enables the service platform and the service manager to guarantee correct remote service access, remote user authentication, and remote service execution based on the IoT application requirements.	Mobility service numbered as C-1-4 in [ITU-T Y.4401].	Service platform, service manager
C-1-4	Reliable autonomic service	The reliable autonomic service capability enables the service platform, the service manager, and the IoT service controller to guarantee automatic capturing, transferring, and analyzing data of things, and automatic provision of services in correct ways based on predefined rules.	Autonomic service numbered as C-1-5 in [ITU-T Y.4401]	Service platform, service manager, IoT service controller
C-1-5	Reliable naming and addressing	The reliable naming and addressing capability enables the service manager, the network manager, and the device manager to guarantee creating, updating, deleting, querying names and addresses of users, devices and things in correct ways based on predefined rules.	Standardized naming and addressing numbered as C-1-9 in [ITU-T Y.4401]	Service manager, network manager, device manager
C-2-1	Reliable event-based Communication	The reliable event-based communication capability enables the service platform, the end-user devices, and the IoT gateways to guarantee correct ways of initiating communication based on predefined events and rules.	Event-based communication numbered as C-2-1 in [ITU-T Y.4401]	IoT gateway, end-user device, service platform

Table C.1 – List of reliable application support capabilities

Capability number	Capability name	Capability summary	Related basic capabilities	Associated components
C-2-2	Reliable periodic communication	The reliable periodic communication capability enables the service platform, the end-user devices, and the IoT gateways to guarantee correct ways of periodically initiating communication based on predefined rules.	Periodic communication numbered as C-2-2 in [ITU-T Y.4401]	IoT gateway, end-user device, service platform
C-2-3	Reliable Quality of Service enabling communication	The reliable Quality of Service enabling communication capability enables the network controller, the end-user devices, and the IoT gateways to guarantee the Quality of Service required for the delivery and processing of data (e.g., time-sensible data) in correct ways.	Quality of service enabling communication numbered as C-2-8 in [ITU-T Y.4401]	IoT gateway, end-user device, IoT network controller
C-3-1	Reliable programmable interface provision	The reliable programmable interface provision capability enables the service platform to guarantee correct ways of providing services or customizing services from existing capabilities based on the IoT application requirements.	Programmable interface provision numbered as C-3-1 in [ITU-T Y.4401]	Service platform
C-3-2	Reliable group management	The reliable group management capability enables the service platform to guarantee correct ways of creating, modifying, deleting, and querying IoT groups, as well as of adding, modifying, deleting and querying IoT group members based on the requirements of the IoT applications and predefined rules.	Group management numbered as C-3-2 in [ITU-T Y.4401]	Service platform

Table C.1 – List of reliable application support capabilities

Capability number	Capability name	Capability summary	Related basic capabilities	Associated components
C-3-3	Reliable time synchronization	The reliable time synchronization capability enables the service platform to guarantee correct ways of synchronizing the time among related functional components, in order to support global or local time stamping for the IoT applications.	Time synchronization numbered as C-3-3 in [ITU-T Y.4401]	Service platform
C-3-4	Reliable orchestration	The reliable orchestration capability enables the service platform, the end-user devices, and the IoT gateways to guarantee correct ways of coordinating service provisioning based on the requirements of the IoT applications and predefined rules.	Orchestration numbered as C-3-4 in [ITU-T Y.4401]	IoT gateway, end-user device, service platform
C-3-5	Reliable user management	The reliable user management capability enables the service platform to guarantee correct ways of creating, querying, updating and deleting IoT user profiles, and authenticating, authorizing, registering and auditing IoT users based on predefined rules.	User management numbered as C-3-5 in [ITU-T Y.4401]	Service platform
C-4-1	Reliable data processing	The reliable data processing capability enables the IoT data server to guarantee trustable results of data fusion and mining based on the IoT application requirements and predefined rules.	Data processing numbered as C-4-2 in [ITU-T Y.4401]	IoT data server
C-4-2	Reliable information exchange	The reliable information exchange capability enables the IoT data server to guarantee correct ways of sending data to or receiving data from external data sources, e.g., data centres and data servers outside the IoT, based on the IoT application requirements and predefined rules.	Open information exchange numbered as C-4-5 in [ITU-T Y.4401]	IoT data server

Table C.1 – List of reliable application support capabilities

Capability number	Capability name	Capability summary	Related basic capabilities	Associated components
C-4-3	Reliable autonomic data operation	The reliable autonomic data operation capability enables the IoT data server, the end-user devices, and the IoT gateways to guarantee correct ways of automatically collecting, aggregating, transferring, storing, analyzing data of things, as well as automatically managing these data operations based on the IoT application requirements and predefined rules.	Autonomic data operation numbered as C-4-7 in [ITU-T Y.4401]	IoT gateway, end-user device, IoT data server
C-5-1	Reliable distributed processing management	The reliable distributed processing management capability enables the IoT data server, the service manager, the network manager, and the device manager to guarantee correct ways of managing IoT functional components in a distributed way based on predefined rules.	Managing distributed processing numbered as C-5-7 in [ITU-T Y.4401]	IoT data server, service manager, network manager, device manager
C-5-2	Reliable multi-domain management	The reliable multi-domain management capability enables the IoT data server, service manager, the network manager, and the device manager to guarantee correct ways of managing IoT functional components in multiple administrative domains based on predefined rules.	Managing multiple domains numbered as C-5-8 in [ITU-T Y.4401]	IoT data server, service manager, network manager, device manager
C-5-3	Reliable service integrity check	The reliable service integrity check capability enables the service manager to guarantee trustable ways of checking service lifetime and available resources required to provide the service in order to guarantee certain degree of service provision availability based on the IoT application requirements.	Service integrity check numbered as C-5-10 in [ITU-T Y.4401]	Service manager

Table C.1 – List of reliable application support capabilities

Capability number	Capability name	Capability summary	Related basic capabilities	Associated components
C-5-4	Reliable data integrity check	The reliable data integrity check capability enables the IoT data server to guarantee trustable ways of checking data lifetime, available attributes of data, and consistency of data in order to guarantee certain degree of availability of data management based on the IoT application requirements.	Data integrity check numbered as C-5-11 in [ITU-T Y.4401]	IoT data server
C-5-5	Reliable device integrity check	The reliable device integrity check capability enables the device manager, the IoT gateway, and the end-user device to guarantee trustable ways of checking the status of all device functionalities in order to guarantee certain degree of device availability of IoT devices based on the IoT application requirements.	Device integrity check numbered as C-5-12 in [ITU-T Y.4401]	Device manager, end-user device, IoT gateway, end-user device
C-5-6	Reliable security integrity check	The reliable security integrity check capability enables the service manager, the network manager, the device manager, and the IoT data server to guarantee trustable ways of checking the consistency of security policies deployed in all functional components of the IoT, in order to guarantee certain degree of security availability in the IoT based on the IoT application requirements.	Security integrity check numbered as C-5-13 in [ITU-T Y.4401]	Service manager, network manager, device manager, IoT data server

Table C.1 – List of reliable application support capabilities

Capability number	Capability name	Capability summary	Related basic capabilities	Associated components
C-5-7	Reliable user profile integrity check	The reliable user profile integrity check capability enables the service manager, the network manager, the device manager, and the IoT data server to guarantee trustable ways of checking the lifetime, subscription, privacy protection, and availability of services subscribed by users, in order to guarantee certain degree of availability of service provisioning and privacy protection for users based on the IoT application requirements.	User profile integrity check numbered as C-5-14 in [ITU-T Y.4401]	Service manager, network manager, device manager, IoT data server
C-6-1	Reliable identification based connectivity	The reliable identification based connectivity capability enables the network manager, the end-user devices, and the IoT gateways to guarantee correct ways of establishing the connectivity based on the identification of things and predefined rules.	Identification based connectivity numbered as C-6-1 in [ITU-T Y.4401]	IoT gateway, end-user device, network manager
C-6-2	Reliable device mobility	The reliable device mobility capability enables the network manager, the end-user devices, and the IoT gateways to guarantee correct ways of keeping the connectivity when the end-user devices or the IoT gateways are moving based on predefined rules.	Device mobility numbered as C-6-3 in [ITU-T Y.4401]	IoT gateway, end-user device, network manager

Appendix I

Use cases for the IoT applications support models from the smart home environment

(This appendix does not form an integral part of this Recommendation.)

Smart home is one of the IoT applications that can be used to make the home environment comfortable and fully automated by connecting home appliances and other electronic devices and sensors through specific wired/wireless connectivity technologies, supporting networking, service provisioning and data collecting and processing functionalities and providing smart home applications, such as home energy management, home security and safety, remote monitoring and control, etc.

A smart home can be implemented and deployed based on its specific (vertical) protocol stack without depending on an IoT service platform. However, a smart home implemented and deployed based on an IoT service platform can make easier and more cost-effective development and deployment of configurable, adaptable and reliable applications by using the capabilities of the service platform.

The following use cases give some examples for smart home applications based on an IoT service platform providing application support model capabilities as specified in this Recommendation.

NOTE – An IoT service platform is identified as the "service platform" functional component of the IoT functional framework specified in [ITU-T Y.4401]

I.1 Use case 1: Configurable remote monitoring in a smart home

One of the procedures for configurable remote monitoring in a smart home is illustrated in Figure I.1.

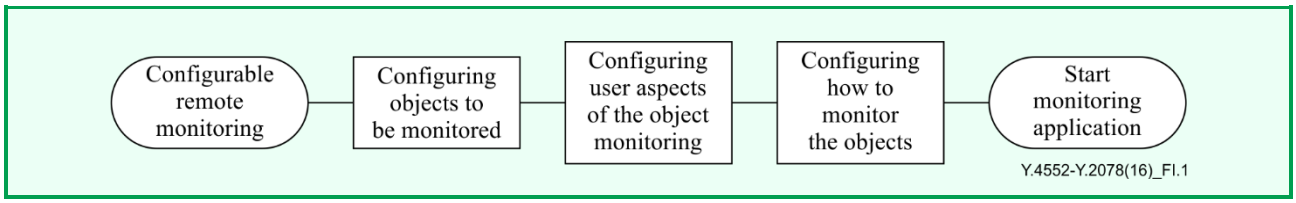


Figure I.1 – A procedure for configurable remote monitoring in a smart home

Configuring objects to be monitored involves activities for configuring the related devices that can sense objects to be monitored. These activities can be realized by making use of the configurable things' status notification capability numbered as A-4-2 and the configurable identification based connectivity capability numbered as A-4-1 and specified in clause 7.2.

Configuring the user aspects of the object monitoring involves activities for configuring data storage methods, mode of the communication and grouping mechanisms related to the end-user devices for monitoring the objects. These activities can be realized by making use of the configurable data storage capability numbered as A-3-1, the configurable communication mode capability numbered as A-2-3 and the configurable group management capability numbered as A-6-1 and specified in clause 7.2.

Configuring how to monitor the objects involves activities for configuring mechanisms for processing monitoring data, the time period for monitoring the objects and the quality of service for transferring monitoring data. These activities can be realized by making use of the configurable data processing capability numbered as A-3-2, the configurable event-based communication capability numbered as A-2-1, the configurable periodic communication capability numbered as A-2-2, the configurable quality of service communication capability numbered as A-2-4 and the configurable content-aware communication capability numbered as A-2-5 specified in clause 7.2.

After finishing these configurations, the application of remote monitoring in a smart home can be started.

I.2 Use case 2: Adaptable home energy management

One of the procedures for adaptable home energy management is illustrated in Figure I.2.

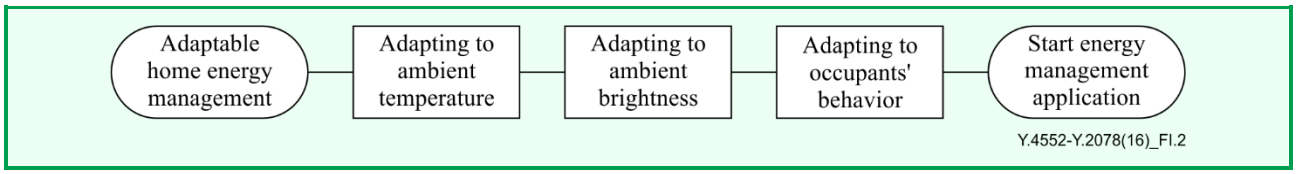


Figure I.2 – A procedure for adaptable home energy management

In this use case, only three types of energy management functions are considered, these are, adapting to ambient temperature, adapting to ambient brightness and adapting to occupants' behavior functions.

Adapting to ambient temperature involves activities for collecting and processing data of ambient temperature based on adaptable data models and adaptable rules of knowledge, adaptable grouping of the home functional components related to monitoring and controlling ambient temperature, such as temperature sensor, smart air conditioners, smart home controller, etc., and adaptable orchestrating of the actions in these home functional components. These activities can be realized by making use of the adaptable identification based connectivity capability numbered as B-5-1, the adaptable data processing capability numbered as B-4-1, the adaptable group management capability numbered as B-3-1 and the adaptable orchestration capability numbered as B-3-2 and specified in clause 8.2.

Adapting to ambient brightness involves activities for collecting and processing data of ambient brightness based on adaptable data models and adaptable rules of knowledge, adaptable grouping the home functional components related to monitoring and controlling ambient brightness, such as brightness sensor, smart lighting switches, smart home controller, etc., and adaptable orchestrating of the actions in these home functional components. These activities can be realized by making use of the adaptable identification based connectivity capability numbered as B-5-1, the adaptable data processing capability numbered as B-4-1, the adaptable group management capability numbered as B-3-1 and the adaptable orchestration capability numbered as B-3-2 and specified in clause 8.2.

Adapting to occupants' behavior involves activities for collecting and processing data of occupants moving and other activities at home based on adaptable data models and adaptable rules of knowledge, adaptable grouping the home functional components related with monitoring, controlling and processing occupants behavior, such as home activity sensor, smart phone, smart home controller, etc., and adaptable orchestrating of the actions in these home functional components. These activities can be realized by making use of the adaptable identification based connectivity capability numbered as B-5-1, the adaptable device mobility capability numbered as B-5-2, the adaptable data processing capability numbered as B-4-1, the adaptable group management capability numbered as B-3-1 and the adaptable orchestration capability numbered as B-3-2 and specified in clause 8.2.

Via the support of these activities, the application of adaptable home energy management can be started.

I.3 Use case 3: Reliable health monitoring at home

One of the procedures for reliable home health monitoring is illustrated in Figure I.3.

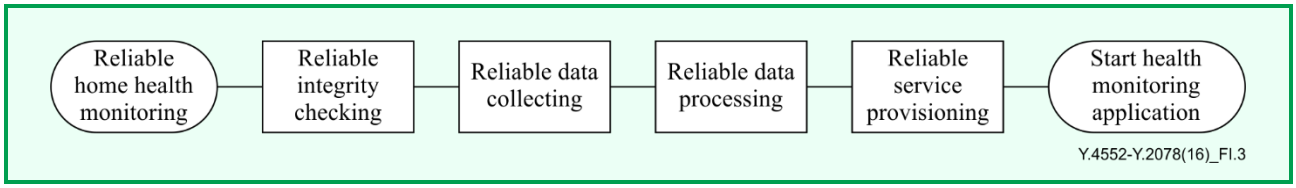


Figure I.3 – One of the procedures for reliable health monitoring at home

Reliable integrity checking involves activities for checking all functions to avoid any possible default. These activities can be realized by making use of the reliable service integrity check capability numbered as C-5-3, the reliable data integrity check capability numbered as C-5-4, the reliable device integrity check capability numbered as C-5-5, the reliable security integrity check capability numbered as C-5-6 and the reliable user profile integrity check capability numbered as C-5-7 and specified in clause 9.2 of this Recommendation.

Reliable data collection involves activities for gathering, transferring and storing data of health monitoring at home in a reliable way. These activities can be realized by making use of the reliable identification based connectivity capability numbered as C-6-1, the reliable device mobility capability numbered as C-6-2, the reliable autonomic data operation capability numbered as C-4-3, the reliable event-based communication capability numbered as C-2-1, the reliable periodic communication capability numbered as C-2-2 and the reliable quality of service enabling communication capability numbered as C-2-3 and specified in clause 9.2.

Reliable data processing involves activities for processing data of health monitoring at home locally or remotely in a reliable way. These activities can be realized by making use of the reliable data processing capability numbered as C-4-1 and the reliable distributed processing management capability numbered as C-5-1 and specified in clause 9.2.

Reliable service provisioning involves activities for providing reliable service interfaces, managing users, groups and services in a reliable way and providing reliable autonomic services. These activities can be realized by making use of the reliable user management capability numbered as C-3-5, the reliable group management capability numbered as C-3-2, the reliable programmable interface provision capability numbered as C-3-1, the reliable mobility service capability numbered as C-1-3 and the reliable autonomic service capability numbered as C-1-4 and specified in clause 9.2.

Via the support of these activities, the application of reliable health monitoring at home can be started.

Bibliography

[b-ITU-T Y.2012] Recommendation ITU-T Y.2012 (2010), *Functional requirements and architecture of next generation networks*.

[b-ITU-T Y.2091] Recommendation ITU-T Y.2091 (2011), *Terms and definitions for next generation networks*

Y.4553

Requirements of smartphone as sink node for IoT applications and services

Requirements of smartphone as sink node for IoT applications and services

Summary

Recommendation ITU-T Y.4553 provides common requirements of a smartphone working as a sink node (SPSN) for Internet of things (IoT) applications and services. Recommendation ITU-T Y.4553 clarifies the concept of a sink node in the IoT domain, and identifies the characteristics, work modes and the high level functional requirements of the SPSN. Cases of use are provided in an appendix.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T Y.4553	2016-03-15	20	11.1002/1000/12779

Keywords

Mobile network, sensor network, sink node, smart phone.

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

Table of Contents

Introduction

With the enormous growth in numbers of mobile phone subscribers, smartphones are being deployed as display terminals for many Internet of things (IoT) applications and services. The use of a smartphone as a sink node to collect various pieces of information from the IoT is becoming more and more popular. For example, a near-field communication- (NFC)-enabled smartphone can be paired with NFC tags or stickers that can be programmed by NFC APPs to automate tasks. In E-commerce, smartphones are used to obtain credit card information and operate as point of sale (POS) terminals. With the powerful computing, communication and storage capacities of these mobile terminals, it is anticipated that the smartphone will act as one of the key devices in the IoT system. The objective of this Recommendation is to develop common requirements for the smartphone as a sink node for IoT applications and services.

Recommendation ITU-T Y.4553

Requirements of smartphone as sink node for IoT applications and services

1 Scope

This Recommendation specifies common requirements for a smartphone acting as a sink node as well as an end user terminal and a mobile gateway for IoT applications and services. More specifically, this Recommendation covers:

– the concept and characteristics of a sink node of the IoT system;

– work modes of a smartphone as a sink node (SPSN) for IoT applications and services;

– requirements for the use of a smartphone as a sink node for IoT applications and services.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T Y.4000] Recommendation ITU-T Y.4000/Y.2060 (2012), *Overview of the Internet of things*.

[ITU-T Y.4101] Recommendation ITU-T Y.4101/Y.2067 (2014), *Common requirements and capabilities of a gateway for Internet of things applications*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 device [ITU-T Y.4000]: With regard to the Internet of things, this is a piece of equipment with the mandatory capabilities of communication and the optional capabilities of sensing, actuation, date capture, data storage and data processing.

3.1.2 gateway [ITU-T Y.4101]: A unit in the Internet of things which interconnects the devices with the communication networks. It performs the necessary translation between the protocols used in the communication networks and those used by devices.

3.1.3 Internet of things (IoT) [ITU-T Y.4000]: A global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies.

NOTE 1 – Through the exploitation of identification, data capture, processing and communication capabilities, the IoT makes full use of things to offer services to all kinds of applications, whilst ensuring that security and privacy requirements are fulfilled.

NOTE 2 – From a broader perspective, the IoT can be perceived as a vision with technological and societal implications.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 sink node: A node in IoT which collects and/or transfers information from/to a group of Internet of things devices (e.g., wearable sensors) in an end user network.

3.2.2 smartphone as a sink node: A smart phone that supports the functionalities of a sink node.

NOTE 1 – In local service mode, a smartphone as a sink node can process collected information locally (e.g., data sorting, format changing, and forwarding), and is the last unit in the flow of information processing (i.e., the smartphone consumes the collected information and does not forward it to a control centre or external entities).

NOTE 2 – In remote service mode, a smartphone as a sink node can forward collected information to remote IoT applications and services through communication networks (i.e., the smartphone does not consume the collected information, but forwards it to other entities over an communication network).

NOTE 3 – A smartphone as a sink node can support local and remote service modes simultaneously.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

2G	second Generation
3G	third Generation
4G	fourth Generation
API	Application Programming Interface
DM	Device Management
IoT	Internet of Things
IP	Internet Protocol
MAC	Media Access Control
NFC	Near-Field Communication
OS	Operating System
PHY	Physical
POS	Point Of Sale
REST	Representational State Transfer
SPSN	Smartphone as a Sink Node
UI	User Interface
Wi-Fi	Wireless Fidelity

5 Conventions

The following conventions are used in this Recommendation:

– The keywords "is required to" indicate a requirement which must be strictly followed and from which no deviation is permitted, if conformance to this Recommendation is to be claimed.

– The keywords "is recommended" indicate a requirement which is recommended but which is not absolutely required. Thus, this requirement need not be present to claim conformance.

6 Description and characteristics of SPSN

6.1 Descriptions

Typically, full-pledged IoT devices (e.g., smartphones) collect the information in either the physical or information world, and then send the information to IoT applications and services through communication networks. However, many kinds of constrained IoT devices in end user networks (e.g., wearable devices, devices in the home or vehicle) cannot connect to communication networks directly. In this case, a smartphone can act as a sink node to provide those types of connectivity capabilities for IoT devices.

Figure 6-1 shows a typical deployment scenario of an SPSN for IoT applications and services, which lies in the device layer of the IoT reference model [ITU-T Y.4000], along with a gateway with sink node functionality. The SPSN performs as a "mobile" sink node to detect and access various types of sensing nodes in an end user network with different types of communication technologies, such as Bluetooth, Wi-Fi and NFC.

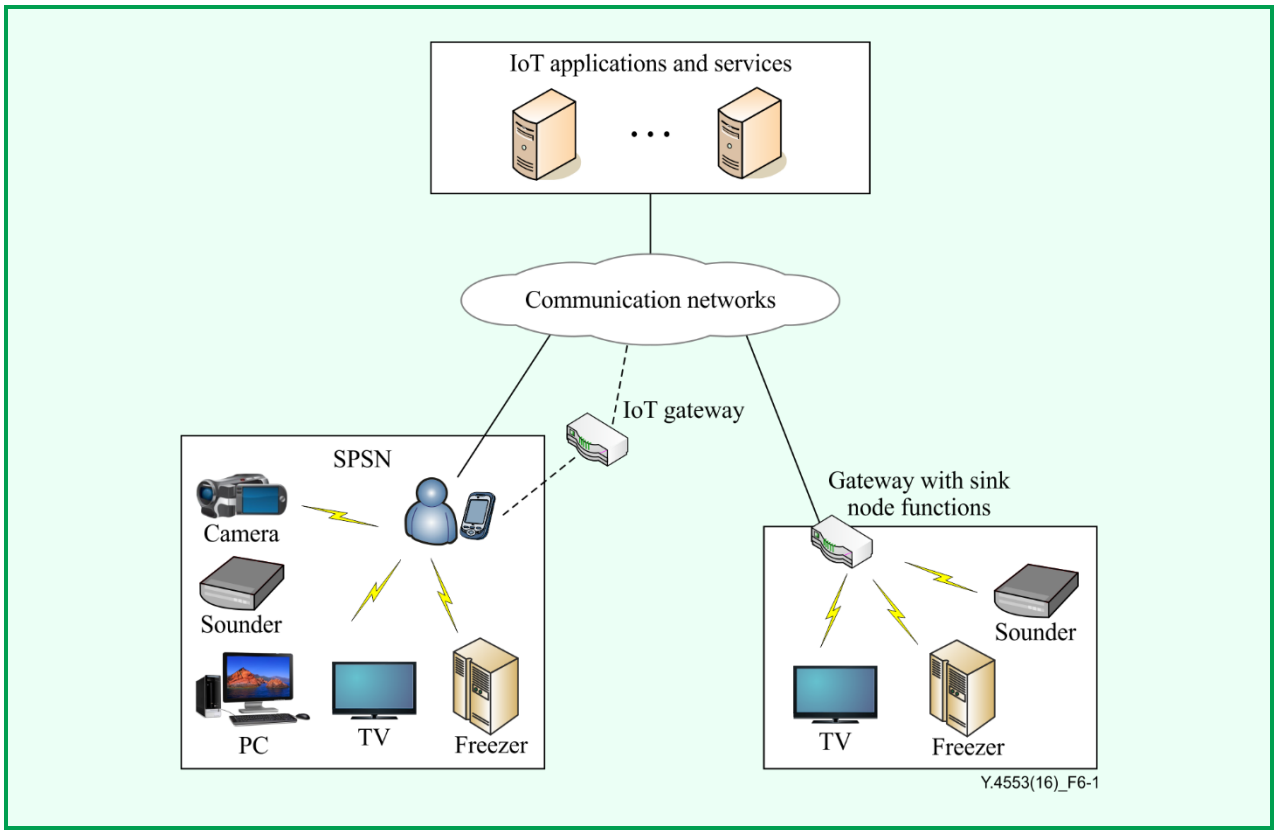


Figure 6-1 – Typical deployment scenario of SPSN

The SPSN can host various IoT applications, which can collect and process the information from IoT devices locally. For instance, the owner of an SPSN can use the IoT applications hosted in the SPSN to manage home IoT devices (e.g., sounder, TV, camera, and freezer).

IoT devices in an end user network can interact, via the SPSN, with remote IoT applications and services through communication networks. Users can remotely manage their IoT devices via their SPSN in an end user network (e.g., at home).

6.2 General characteristics

This clause provides characteristics of the SPSN for IoT applications and services. These characteristics include, but are not limited to, those described in 6.2.1 to 6.2.8.

6.2.1 Access to communication networks

An SPSN accesses communication networks directly or via the IoT gateway as described in [ITU-T Y.4101].

6.2.2 Interaction with IoT applications and services

An SPSN interacts with IoT applications and services through communication networks.

6.2.3 Access to the IoT device

An SPSN supports access to IoT devices in an end user network based on different communication technologies, such as Bluetooth, Wi-Fi and NFC.

6.2.4 Protocol translation

In an end user network, an SPSN supports data and protocol translations between IoT devices and IoT applications and services.

6.2.5 Device management characteristic

An SPSN supports the exposure of IoT devices in an end user network to allow an IoT device management (DM) service to manage the exposed IoT devices remotely.

6.2.6 Security

An SPSN provides security supports, such as IoT device authentication, data encryption and privacy protection. It is essential for the SPSN to retain security and privacy protection for its mobile characteristics.

6.2.7 Portability

An SPSN provides portability. It is expected that an SPSN provide standardized internal and external interfaces with which IoT applications can easily interact with IoT devices exposed by the SPSN.

6.2.8 Process of accessing data

An SPSN supports the process of collecting information from IoT devices in order to either display it on the SPSN or forward it to other elements in the communication network or IoT devices in the end user network.

7 Work modes of the SPSN

7.1 Local service mode

In local service mode, an SPSN can: host local IoT applications; collect IoT data from IoT devices in an end user network; and transfer IoT data from local IoT applications to IoT devices in the end user network. Local IoT applications on the SPSN can manage (e.g., store, process and demonstrate) collected IoT data and can forward it in processed form to IoT devices in the end user network. In this service mode, an SPSN does not need to transfer IoT data to remote IoT applications and services through communication networks. See Figure 7-1.

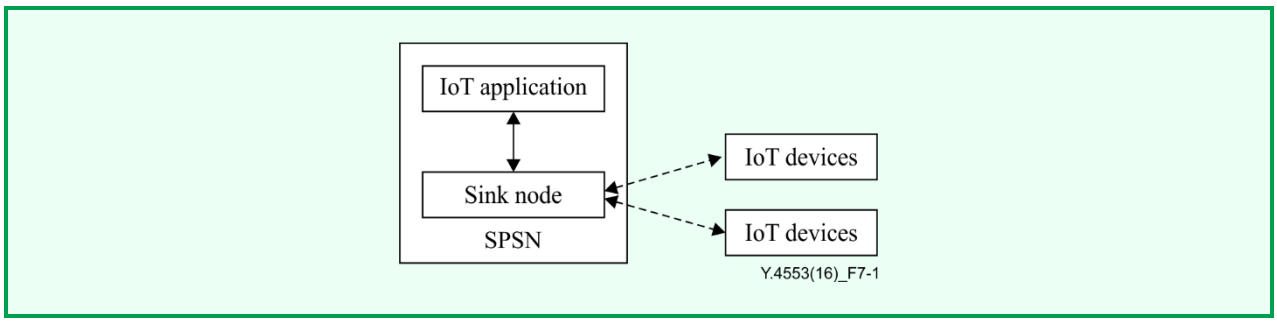


Figure 7-1 – Local service mode of the SPSN

7.2 Remote service mode

In remote service mode, an SPSN accesses IoT devices in an end user network and transfers IoT data from the SPSN to remote IoT applications and services through communication networks. Transferred IoT data can be stored and processed in such remote IoT applications and services. The users can browse and access IoT data using their terminals (e.g., smartphone, laptop, and PC) via the Internet. Additionally, in this service mode, an SPSN can forward IoT data provided by remote IoT applications and services to the IoT devices in an end user network. See Figure 7-2.

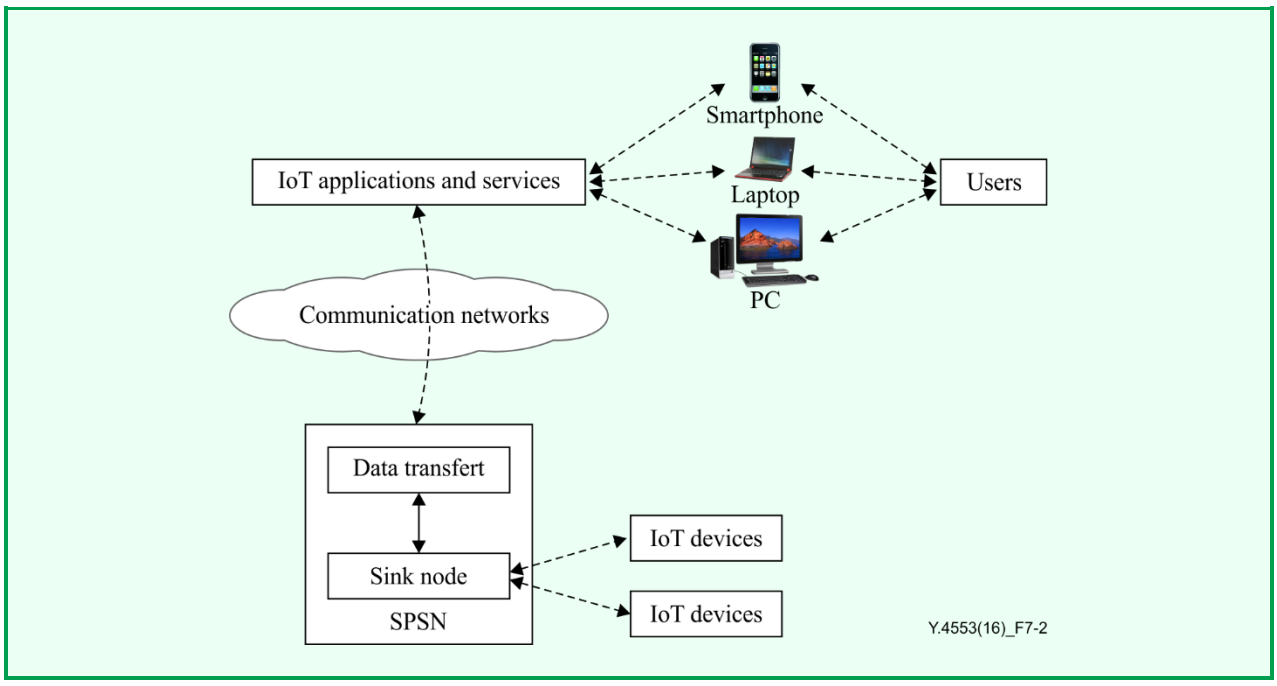


Figure 7-2 – Remote service mode of the SPSN

An SPSN may support local and remote service modes simultaneously. An SPSN can process collected IoT data from IoT devices in an end user network in order to either display it through local IoT applications on the SPSN or transfer the processed IoT data to remote IoT applications and services.

8 Requirements of the SPSN

8.1 General requirements

The general requirements of the SPSN are as follows.

1) The SPSN is required to support multiple connections of IoT devices and collaborations with other SPSNs.
2) It is recommended that the SPSN support various addressing schemes, e.g., Internet protocol (IP) and non-IP addressing schemes, to interact with the connecting or connected IoT devices.
3) The SPSN is required to provide standardized open interfaces for easy capability extensions, e.g., for integration with third party IoT applications and services.

8.2 Network connectivity

The SPSN provides logical connectivity between IoT devices in end user networks and communication networks.

The network connectivity requirements of the SPSN are as follows.

1) The SPSN is required to be able to connect to communication networks through communication technologies (e.g., 2G/3G/4G/Wi-Fi).
2) The SPSN is required to support data exchange between IoT devices and IoT applications and services.
3) It is recommended that the SPSN translate data and protocols between IoT devices and IoT applications and services.

8.3 Local information processing

The SPSN can support information processing locally. With IoT applications in the SPSN, IoT data can be collected, stored, processed and demonstrated in the SPSN directly.

The local information processing requirements of the SPSN are as follows.

1) The SPSN is required to install the IoT applications locally.
2) It is recommended that the SPSN store data locally.
3) It is recommended that the SPSN compute IoT data locally (e.g., data averaging).
4) It is recommended that the SPSN support the user interface (UI) for connected IoT devices.
5) It is recommended that the SPSN support standardized Web-based interfaces [e.g., representational state transfer- (REST)-ful application programming interfaces (APIs)] with which remote IoT applications and services interact with connected IoT devices.

8.4 Devices connectivity

When the SPSN joins an end user network, it can detect the IoT devices in that end user network, and the IoT devices can also actively detect and connect to the SPSN.

The devices connectivity requirements of the SPSN are as follows.

1) The SPSN is required to be able to discover and connect to the IoT devices in an end user network, subject to the user's request.
2) The SPSN is required to allow access to the connected IoT devices and to allow access from the connected IoT devices, according to pre-defined polices.
3) The SPSN is required to connect and disconnect the IoT devices actively, if necessary.

8.5 Data exchanging

When an SPSN establishes connections with IoT devices in an end user network, it can exchange data with those devices.

The data exchange requirements of the SPSN are as follows.

1) The SPSN is required to collect data from IoT devices.
2) The SPSN is required to send or dispatch data to IoT devices.
3) It is recommended that the SPSN adapt policy to data collection or data transfer according to users' requests.

8.6 Support of multiple communication protocols

IoT devices in an end user network can support various communication protocols. An SPSN can provide protocol adoption functions to support interactions with IoT devices.

The related requirements of the SPSN are as follows.

1) The SPSN is required to support multiple communication protocols, such as Bluetooth, Wi-Fi and NFC, in order for smartphones to connect to IoT devices.
2) It is recommended that the SPSN dynamically support various protocols.

8.7 DM requirements

The DM server can remotely manage an SPSN and connected IoT devices.

The DM requirements of the SPSN are as follows.

1) It is recommended that the SPSN support remote DM service to manage the IoT devices exposed by the SPSN.
2) It is recommended that the SPSN support remote DM service to manage the SPSN.

8.8 Security and privacy

An SPSN can participate in or leave an end user network casually, therefore it is essential to guarantee the security and privacy of the SPSN and connected IoT devices.

The security and privacy requirements of the SPSN are as follows.

1) An SPSN is required to support mutual or one-way authentication with IoT devices.
2) An SPSN is required to support mutual authentication with communication networks.
3) An SPSN is required to support mutual authentication with IoT applications and services.
4) An SPSN is required to securely store data or transfer the its own data and that of connected IoT devices.
5) An SPSN is required to protect its own data privacy and that of connected IoT devices.

Appendix I

Use cases of the SPSN for IoT applications and services

(This appendix does not form an integral part of this Recommendation.)

This appendix describes typical SPSN scenarios for IoT applications and services.

I.1 SPSN for commercial merchant service

In this scenario, an SPSN can be used to manage supplier service information. In commercial service, suppliers need to collect information about products, order data and the purchasers' account information required to complete transactions. On one hand, an SPSN has sensors for gathering information including identity data, transaction sensitive data and geographical position data that is acquired through various technologies, such as NFC, sounder and Bluetooth. On the other hand, an SPSN has a big screen to display real information for both suppliers and purchasers. Therefore, it is natural that suppliers use SPSNs to gather information. See Figure I.1.

Suppliers or retailers can use SPSNs to access and control each IoT device using related IoT applications installed on them. The SPSN performs identity recognition and information acquisition during the transaction process.

Subsequently, information is transferred to a data processing centre, through a special network, such as the Internet, cable network or wireless network.

Finally, information processing centres deployed in banks and third parties undertake information analysis, decision-making, sharing, publishing and provision to users of an instant intelligence acquisition service.

For instance, when a customer wants to buy products and pay for them, the merchant could use the SPSN to scan tags to collect the products' information, swipe the contactless bank card to get the account information or even swipe the customer's mobile phone if he or she has a mobile wallet. As a result, the SPSN collects all the data together just as the sink node does. It sends the data to a processing centre through the network and also transmits feedback to the merchant and customers.

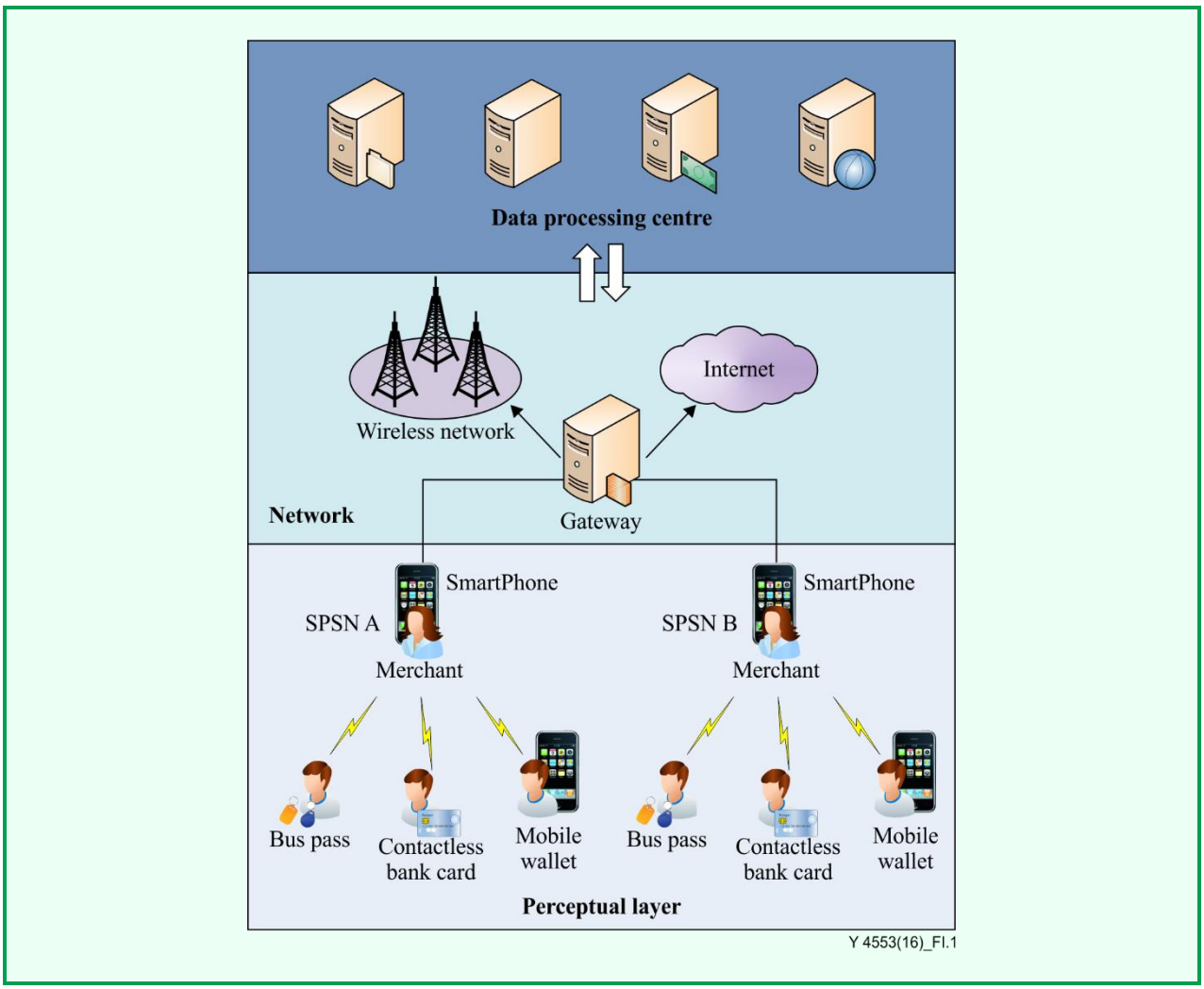


Figure I.1 – Scenario of the SPSN used for commercial merchant

I.2 SPSN for home services

The SPSN can be used at home, where occupiers can connect and manage IoT devices (e.g., sounder, freezer, and camera). With the support of an occupier's SPSN, IoT devices in home network can communicate with remote IoT applications and services through a communication network. Additionally, IoT devices can interact with local IoT applications hosted on the SPSN. Figure I.2 shows the scenario for SPSN use in the home.

When an occupier returns home carrying his or her SPSN, the device can automatically (or mutually) discover and connect to domestic IoT devices. Then the occupier can control these devices through the IoT application hosted on the SPSN. Furthermore, an occupier can leave the SPSN at home and remotely control his or her domestic IoT devices through the SPSN via communication networks, for example from an office.

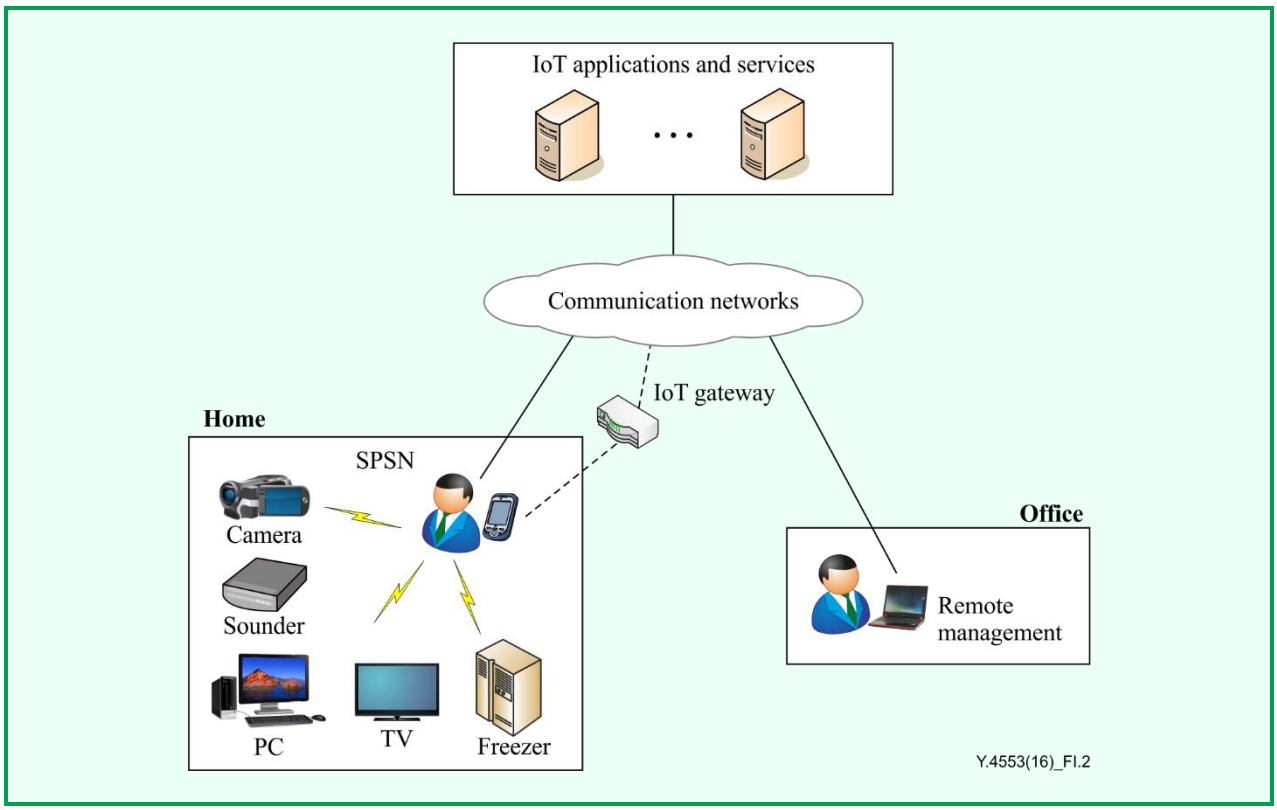


Figure I.2 – Scenario for SPSN use in the home

I.3 SPSN for environment-monitoring services

The SPSN can be used to monitor the environment. Environment-monitoring staff can collect sensing data with the SPSN and control the local sensor network in the outdoor environment. In this scenario, the SPSN is used as a processor as well as an information aggregator.

Figure I.3 illustrates this scenario. A temperature sensor, moisture sensor, hydro sensor and outdoor watering equipment are located at specific places to monitor temperature, humidity and hydration of the ground. All these sensors are equipped with a Bluetooth (or Wi-Fi) module.

A specific application installed on the SPSN can detect these environment-monitoring devices through Bluetooth and recognize them via the media access control (MAC) addresses of the devices. It can also transform raw data from the sensor to a readable form for the user to interpret easily.

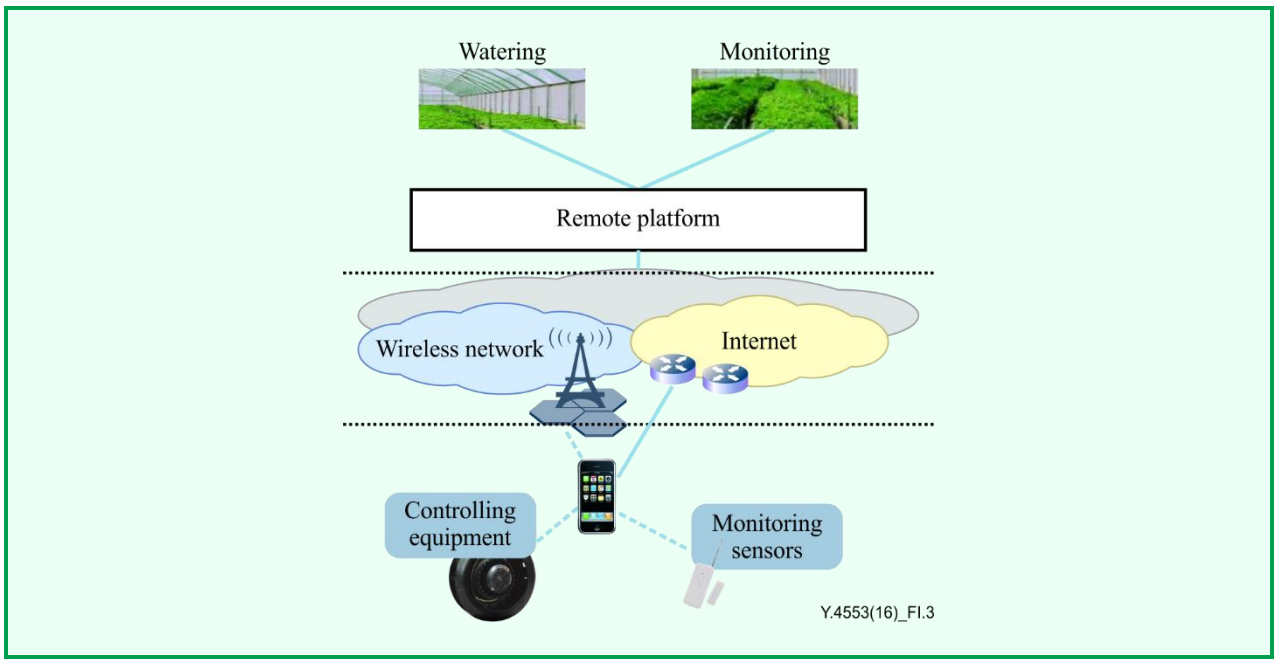


Figure I.3 – Scenario for SPSN use for environment monitoring

When environment-monitoring staff wish to check the status of a specific place, they turn on the monitoring application and the SPSN discovers sensors close to the SPSN with a Bluetooth-equipped sensor, connects to them and collects data automatically. The SPSN then asks for the temperature, moisture and water sensor data and displays them as programmed.

In this scenario, the SPSN collects the information from various sensors, processes the data locally to the required form, and finally uploads all the information to the monitoring centre.

I.4 SPSN for wearable smart devices

Wearable smart devices (e.g., watches, glasses, headbands and belts) are becoming more and more popular. Generally, wearable smart devices allow the owner to access information in real-time or non-real-time.

Due to the high mobility of smartphones, SPSN can be the most common tool for owners to access and manage their wearable smart devices.

An SPSN can be used to connect and manage users' constrained wearable smart devices through personal communication technologies, such as ZigBee, NFC, Bluetooth, Wi-Fi or USB. Figure I.4 shows the scenario for SPSN use for wearable smart devices.

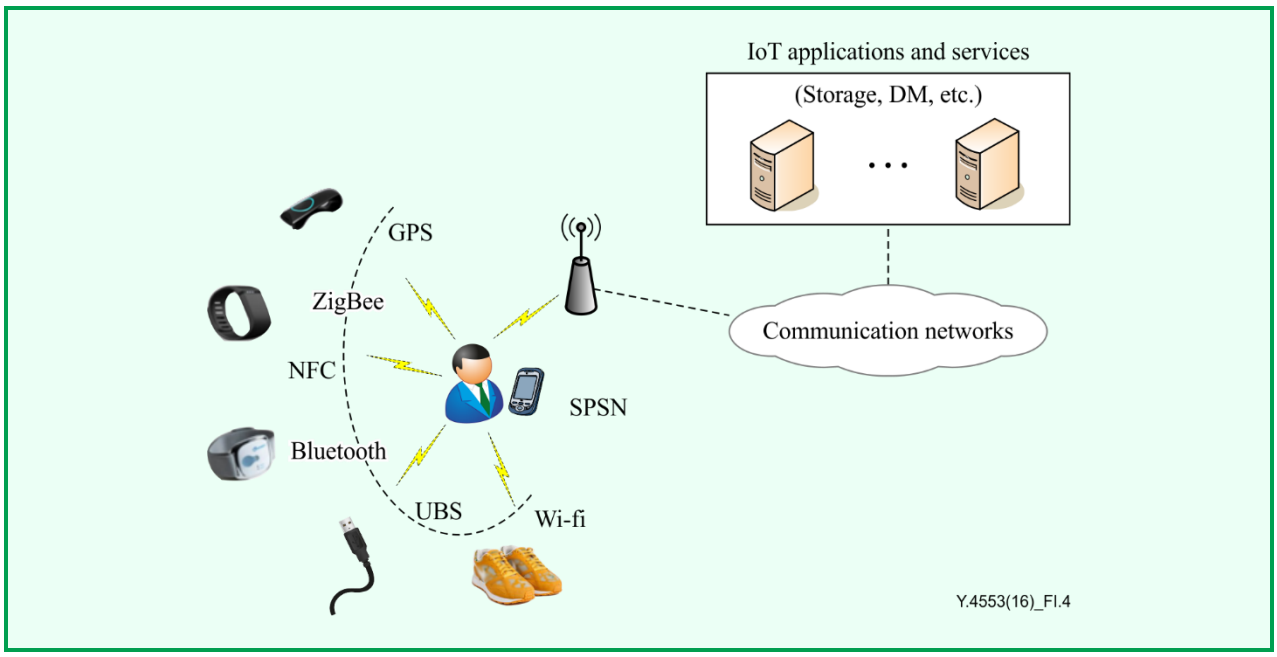


Figure I.4 – Scenario for SPSN use for wearable smart devices

In Figure I.4, the user has several wearable smart devices, such as a hairpin, a bracelet, a watch, a pair of glasses and a pair of shoes. These wearable smart devices connect to the user's SPSN through personal communication technologies.

Users can use their SPSNs to detect and connect to their wearable smart devices, through relevant IoT applications installed on the SPSN.

The SPSN collects and synchronizes information (including device capabilities) of the connected wearable smart devices with local stores or network repositories (such as cloud stores). The data collection can be real-time or non-real-time, depending the communication technology used and the device's capabilities.

Furthermore, the SPSN can process the devices' information locally with local relevant IoT applications.

At any time, the owner can use the SPSN to configure any of the wearable smart devices connected to it.

Appendix II

Example of sink node related functions of an SPSN

(This appendix does not form an integral part of this Recommendation.)

NOTE – This appendix takes the Android operating system (OS) platform as an example to illustrate the sink node-related functions of an SPSN. Note that each type of OS platform for SPSNs may have their own relevant implementation mechanisms.

The sink node-related functions of an SPSN include four logical function groups generally: adaptation functions, supporting functions, applications, and security and management functions. Those functions are part of the smartphone. See Figure II.1.

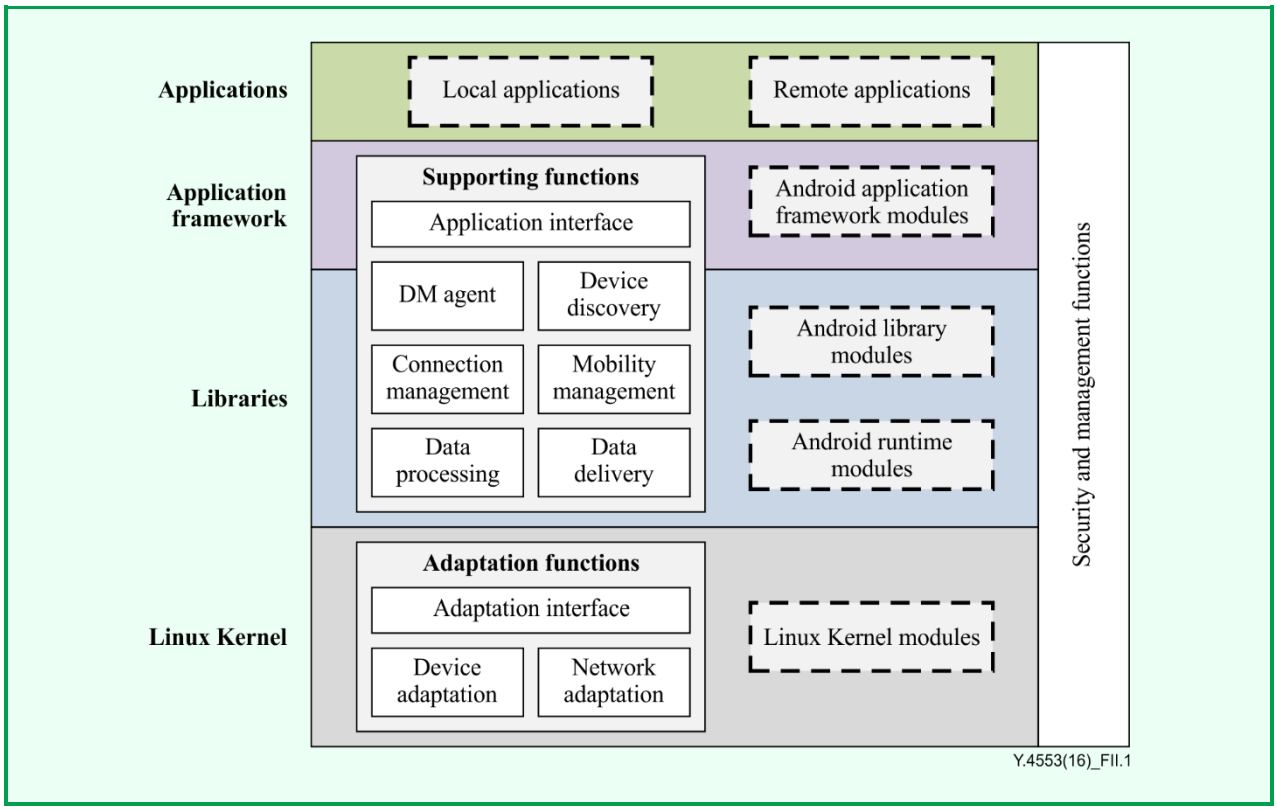


Figure II.1 – Example of sink node-related functions of an SPSN

The adaptation functions group in the Linux kernel layer of the Android OS includes communication-related functions of the SPSN to interact with the IoT devices (via device adaptation), and to communicate with the IoT applications and services (via network adaptation). This functional group includes at least the following functions.

– Device adaptation, which provides implementation of connecting IoT devices in the end user network and maps the device information according to the abstract adaptation interface.

– Network adaptation, which provides PHY/MAC layer adaptations for the SPSN to interact with the IoT devices in an end user network or the IoT applications and services via the communication networks, respectively.

– Adaptation interface, which provides an abstract interface to support the upper supporting functions and applications to access IoT devices, or the IoT applications and services, respectively.

The supporting functions group, across the libraries layer and application framework layer of the Android OS, provides functionalities including DM and discovery, connection management, mobility management, data processing and data delivery. This functions group includes at least the following functions.

– A DM agent, which supports local or remote IoT device management applications to manage the IoT devices exposed by the SPSN.
– Device discovery, which discovers the IoT devices in an end user network actively and processes active connection requests from the IoT devices.
– Connection management, which establishes and manages the connections, including the connections between IoT devices and the SPSN, and the connections between the SPSN and IoT applications and service.
– Authentication management, which manages the mobility of the SPSN. When the SPSN joins or leaves an end user network the SPSN manages the authentications and authorities to IoT devices and end user networks.
– Data processing, which collects and processes IoT data from IoT devices locally.
– Data delivery, which delivers IoT data (including locally processed IoT data) to other entities in communication networks or back to IoT devices.
– Application interface, which provides an abstract interface to support local and remote IoT applications and services to access the IoT devices exposed by the SPSN, with standard and uniform logical methods (e.g., web-based interfaces).

The local or remote IoT applications and service use the application interfaces and adaptation interfaces mentioned above to discover, access and manage IoT devices in the end user network.

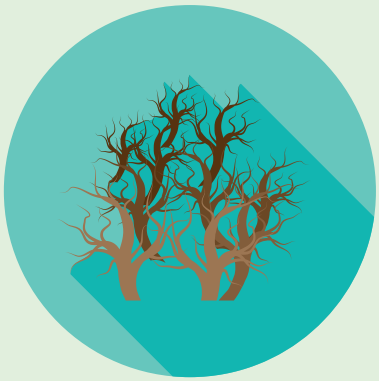
The security and management functions group, cooperating closely with all the layers of the Android OS, provides capabilities for supporting security of data and communications.


TV
Refrigerator
Car
Light
Washing machine
IOT
INTERNET
OF THINGS

Management, Control and Performance

7



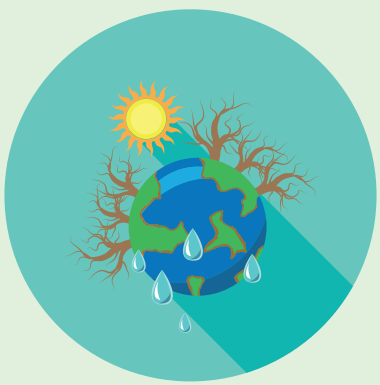










Y.4700/F.747.2

Deployment guidelines for ubiquitous sensor network applications and services for mitigating climate change

Deployment guidelines for ubiquitous sensor network applications and services for mitigating climate change

Summary

Recommendation ITU-T F.747.2 provides deployment guidelines for ubiquitous sensor network (USN) applications and services for mitigating climate change.

History

Edition	Recommendation	Approval	Study Group
1.0	ITU-T F.747.2	2012-06-29	16

Keywords

CC, climate change, GHG, greenhouse gas, USN, ubiquitous sensor network.

Table of Contents

Recommendation ITU-T Y.4700/F.747.2

Deployment guidelines for ubiquitous sensor network applications and services for mitigating climate change

1 Scope

This Recommendation provides deployment guidelines for ubiquitous sensor network (USN) applications and services for mitigating climate change. The scope of this Recommendation includes:

– an overview of climate change monitoring;

– analysis of environmental impact by USN applications and services; and

– the requirements for deployment of USN applications and services for mitigating climate change.

Monitoring climate change covers monitoring the status of greenhouse gas (GHG) emissions, as well as monitoring climate change by tracing temporal changes of GHG emissions.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T Y.2221] Recommendation ITU-T Y.2221 (2010), *Requirements for support of ubiquitous sensor network (USN) applications and services in the NGN environment*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 climate change [b-IPCC]: Climate change refers to a change in the state of the climate that can be identified (e.g., by using statistical tests) by changes in the mean and/or the variability of its properties, and that persists for an extended period, typically decades or longer. Climate change may be due to natural internal processes or external forcings, or to persistent anthropogenic changes in the composition of the atmosphere or in land use. Note that the Framework Convention on Climate Change (UNFCCC), in its Article 1, defines climate change as: 'a change of climate which is attributed directly or indirectly to human activity that alters the composition of the global atmosphere and which is in addition to natural climate variability observed over comparable time periods'. The UNFCCC thus makes a distinction between climate change attributable to human activities altering the atmospheric composition, and climate variability attributable to natural causes.

3.1.2 greenhouse gas [b-ISO 14064-1]: Gaseous constituent of the atmosphere, both natural and anthropogenic, that absorbs and emits radiation at specific wavelengths within the spectrum of infrared radiation emitted by the Earth's surface, the atmosphere and clouds.

3.1.3 sensor [ITU-T Y.2221]: An electronic device that senses a physical condition or chemical compound and delivers an electronic signal proportional to the observed characteristic.

3.1.4 sensor network [ITU-T Y.2221]: A network comprised of interconnected sensor nodes exchanging sensed data by wired or wireless communication.

3.1.5 sensor node [ITU-T Y.2221]: A device consisting of sensor(s) and optional actuator(s) with capabilities of sensed data processing and networking.

3.1.6 ubiquitous sensor network [ITU-T Y.2221]: A conceptual network built over existing physical networks which make use of sensed data and provide knowledge services to anyone, anywhere and at any time, and where the information is generated by using context awareness.

3.1.7 USN middleware [ITU-T Y.2221]: A set of logical functions to support USN applications and services.

3.2 Terms defined in this Recommendation

None.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

API	Application Program Interface
CPU	Central Processing Unit
GAW	Global Atmosphere Watch
GHG	Greenhouse Gas
IPCC	Intergovernmental Panel on Climate Change
RX	Receiver
TX	Transmitter
UNFCCC	United Nations Framework Convention on Climate Change
USN	Ubiquitous Sensor Network

5 Conventions

None.

6 Overview of climate change monitoring

6.1 Global greenhouse gas monitoring network

Monitoring greenhouse gas (GHG) emissions as well as climate change requires that GHG sensors, sensor nodes and sensor networks are installed nationally and/or globally. A nationwide GHG monitoring network may interwork with a global GHG monitoring network, for example, the one illustrated in Figure 1, which is maintained by the Global Atmosphere Watch (GAW) programme of the World Meteorological Organization (WMO) [b-GAW programme].

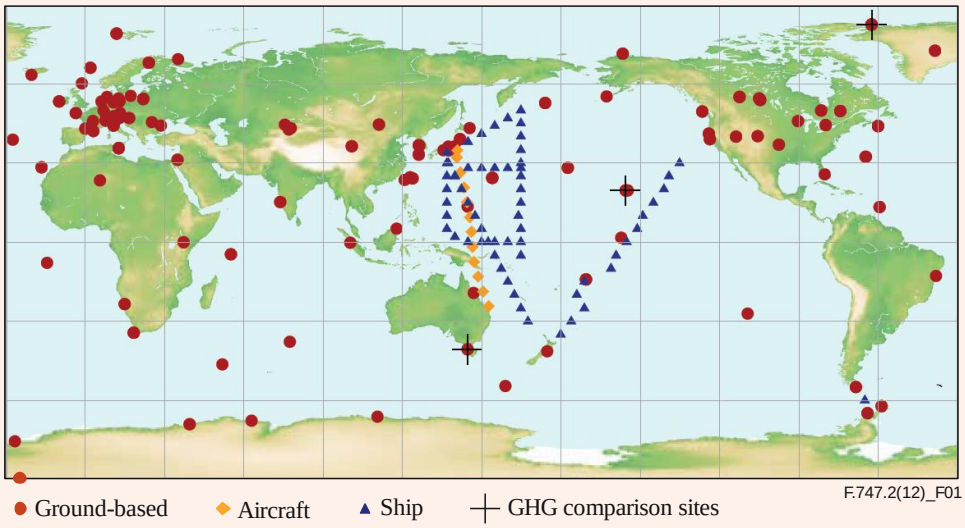


Figure 1 – The WMO-GAW global network for GHG

6.2 Local GHG monitoring network

[b-IPCC Guidelines] defines three tiers for estimating GHG emissions from fossil fuel combustion:

– The tier 1 method is fuel-based, since emissions from all sources of combustion can be estimated on the basis of the quantities of fuel combusted (usually from national energy statistics) and average emission factors. Tier 1 emission factors are available for all relevant direct greenhouse gases.

– The tier 2 method is estimated from similar fuel statistics as used in tier 1, but country-specific emission factors are used in place of the tier 1 defaults. This is because different specific fuels, combustion technologies or even individual plants may produce different country-specific emission factors.

– The tier 3 method uses either detailed emission models or measurements and data at an individual plant level where appropriate. Properly applied, these models and measurements should provide better estimates, primarily for non-CO2 greenhouse gases, though at the cost of more detailed information and effort.

The tier 3 method allows an enterprise to measure real GHG emissions to avoid overestimation that may happen due to the conservativeness principle. In the latter, conservative assumptions, values and procedures are used when data and assumptions are uncertain and the cost of measures to reduce uncertainty is not worth the increase in accuracy. Conservative accounting results for GHG emissions are more likely to be overestimated than underestimated.

Enterprises may install a local GHG monitoring network at their plant level.

7 Analysis of environmental impact by USN applications and services

7.1 Deployment elements of USN

[ITU-T Y.2221] defines USN as a conceptual network and an information infrastructure that delivers sensed information and knowledge services to anyone, anywhere and at any time. In USNs, information and knowledge are developed by using context-aware techniques.

USN applications and services are established by integration of sensor network services into a network infrastructure. They can be applied to everyday life in an invisible way as everything is virtually linked by pervasive networking between users (including machine and human) and sensor nodes, and relayed through intermediate networking entities such as application servers,

middleware entities, access network entities, and USN gateways. Integration of the hardware, software, USN applications and USN services can be used in many civilian application areas such as industrial automation, home automation, agricultural monitoring, healthcare, environment, pollution and disaster surveillance, and security.

Figure 2 shows elements of deploying USN applications and services to mitigate climate change. They may cause both positive and negative impacts on the environment.

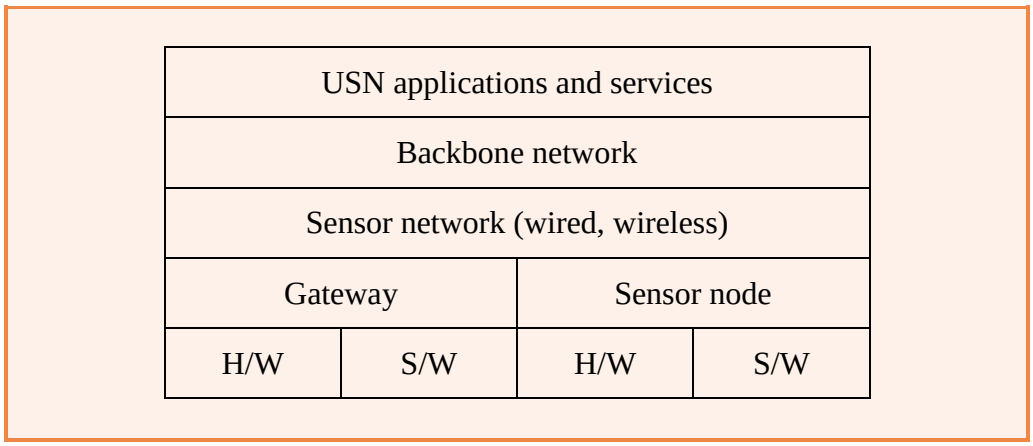


Figure 2 – Deployment elements of USN applications and services

7.2 Positive environmental impacts

USN is a key technology to mitigate climate change by monitoring diverse environmental data and enabling energy consuming sources to be controlled according to the environmental data.

Sensor nodes can measure and deliver different types of environmental data, such as, pressure, humidity, temperature, light, chemicals, strain and tilt, speed and acceleration, magnetic fields, vibrations, motion, metal detection and sound.

The sensing parameters are used to trace climate change and to understand climate phenomena. The issues are how to deliver the sensed data and how to manage, present and exploit the data to derive value-added information for countering climate change. This clause briefly introduces examples of how USNs are applied to mitigate climate change.

7.2.1 Direct climate change monitoring

USN applications and services provide direct monitoring for the acquisition of climate data. For example, marine environment monitoring and glacier status monitoring help trace continuous environmental changes.

In order to help counter climate change, it is important to monitor the climate to verify if changes to the environment are caused by human influence or natural phenomena. The use of sensor networks to monitor the climate has been researched for decades; this has allowed the development of viable technology and techniques for monitoring climate change. It has been proved with many experiments, that USN-based monitoring systems give valuable data.

Marine environment monitoring shown in Figure 3 is an example of direct environment monitoring. The data of the sensor nodes used to monitor the real-time status of the marine and glacier environment are transmitted to the local monitoring and management system.

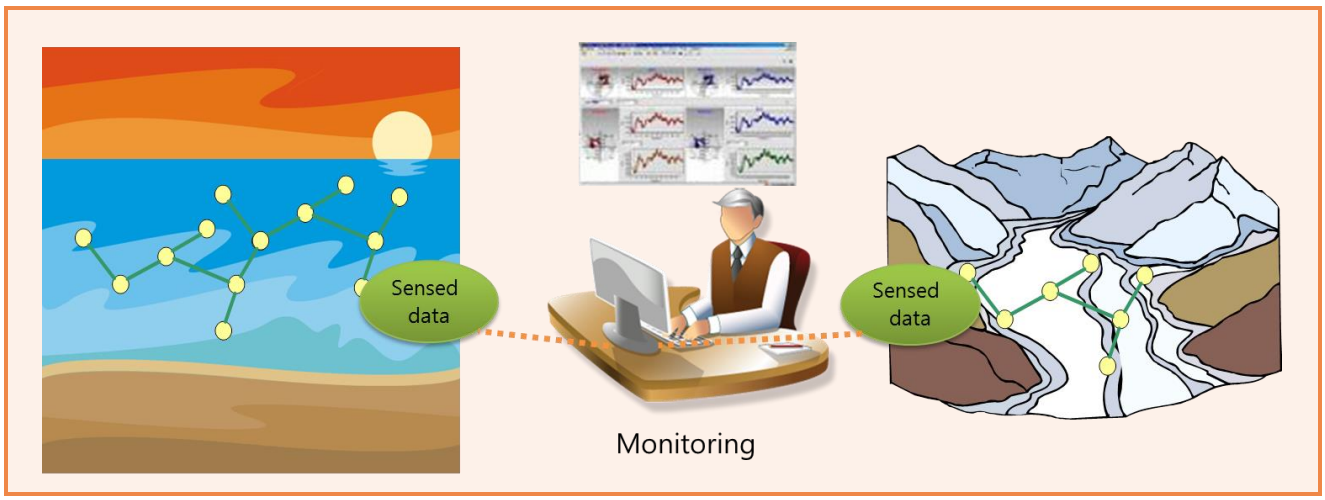


Figure 3 – Example of marine and glacier environment monitoring

Upper air current and atmospheric state monitoring is presented as another direct climate monitoring example, as shown in Figure 4. It includes slightly different features than the general area of climate monitoring. Altitude changes, temperature, humidity and atmospheric flow are the key information for understanding climate change in a certain region.

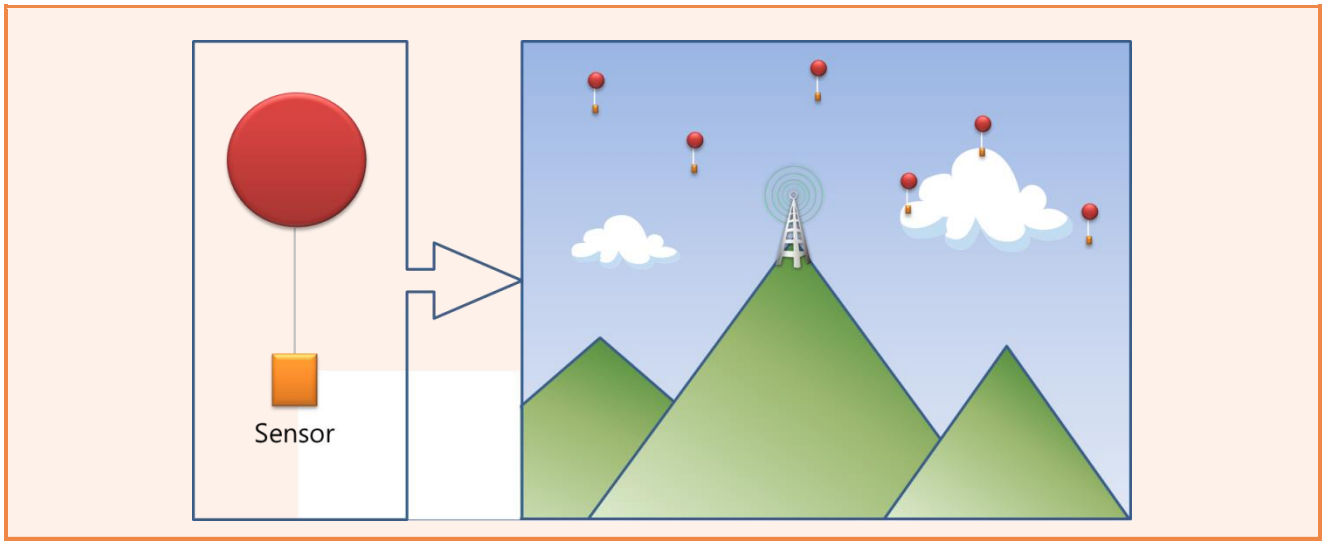


Figure 4 – Upper air current and upper atmosphere state monitoring example

7.2.2 Monitoring and control of GHG emissions

USN applications can be developed to automatically monitor and limit power consumption levels. Various USN applications have the ability to monitor electric power consumption and air pollution to alert users when their systems exceed established thresholds.

A good example of this category of USN applications is a management system of various components in urban infrastructure, such as roads, sewerage, water and gas lines. When USN systems sense a defect, the systems will activate the corresponding maintenance systems to correct the malfunction. For example, a road management system captures road conditions and provides this information to drivers with additional weather information. In addition, USN applications can help reduce GHG emissions caused by stop-and-go traffic through rerouting traffic to less congested routes.

Figure 5 – Management of city facilities example

Another example of monitoring energy and GHG emission is home and commercial building automation. Light bulbs can automatically control the appropriate brightness based on information from motion sensors and ambient light. Home appliances and other electronic gadgets can enter energy-saving modes when not in use. Controlling power consumption levels and GHG emissions of commercial buildings is more complex than controlling the consumption and emissions levels of single-family homes. However, the same type of automation equipment or similar system concepts can be used in commercial buildings. Home and building monitoring servers are able to show the monitored power consumption levels to allow owners to adjust usage levels appropriately. These monitoring and control systems are known to reduce GHG emissions on average by about 10%.

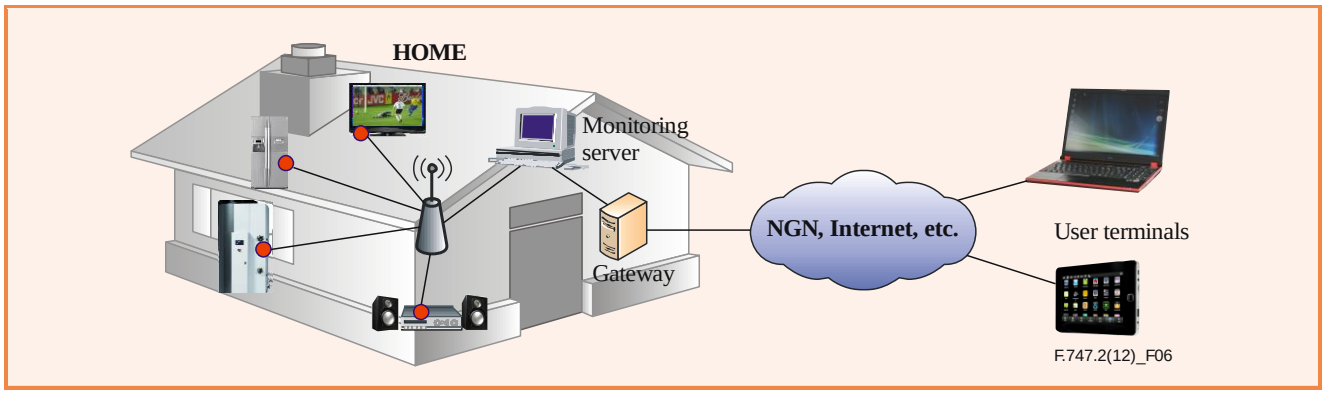


Figure 6 – Home GHG monitoring example

7.2.3 Indirect monitoring to learn climate features

There are many USN applications that allow indirect monitoring for the acquisition of climate data. This type of USN application is essential in allowing researchers to analyse and understand climate change. Understanding climate change is the first step in developing strategies to deal with impending crises that could threaten global supplies for drinking water, sanitation and irrigation.

USN applications can be extensively deployed to monitor any environmental changes and help understand the cause of the change. The results of the collected data can be used to predict future changes.

A hydro watch USN application builds wireless sensor networks to more closely examine the water cycle and can be used to understand climate phenomena. Sensor nodes may be installed in greenhouses and in open fields, and a sensor network application monitors the agricultural environment and learns about the plants' habitat, in order to help manage optimal plant growing conditions.

7.3 Negative environmental impacts

As global awareness on climate change rises in the ICT sector, there is also increased awareness of the environmental impact of electric and electronic products, the restriction on the use of hazardous substances and the use of eco-designs. Furthermore, GHG emission by-products are generated in the life cycle of all raw materials including material processing, manufacturing, distribution, use, repair and maintenance, and the disposal or recycling of products. Although USNs are not an exception in this aspect, they can be used in many areas and cause a positive net environmental impact.

7.3.1 Use of hazardous materials

The elements of USN contain physical equipment such as gateways, sensor nodes, sensors and batteries. This includes small sensor nodes mostly powered by batteries. Batteries contain heavy metals such as mercury, lead, cadmium and nickel, which can contaminate the environment if batteries are improperly disposed. If the used sensor node cannot be collected, the electronics waste generated from USN physical equipment and certain metals can release hazardous elements in the ash produced by the combustion process. Therefore, the recovery from environmental pollution by electronic waste causes further GHG emission.

7.3.2 Indirect GHG emissions

USN applications and services will cause an environmental load in each product life cycle phase. However, most of the environmental load is caused by using electric power in the use phase. Consuming electric power causes indirect GHG emission from power plants (e.g., thermoelectric power plants, etc.) where the GHG were produced during electric power generation.

8 Requirements for deployment of USN applications and services for mitigating climate change

Even though USN applications and services have a greater positive impact on mitigating climate change in various areas, they are not free from GHG emission as described in clauses 7.2 and 7.3. Therefore, it is important to deploy and utilize USN applications and services in an environmentally-friendly manner. In addition, eco-design and eco-operations must be considered in sensor network gateways and other dedicated servers, as well as sensor nodes.

8.1 Environmentally friendly resources

Sensor nodes are designed and manufactured in small sizes with small memory and low processing power, and run on very limited power supplied by non-rechargeable batteries. This basic design principle of sensor nodes with small sizes and low processing power makes USN applications and services a good solution to pursue low carbon emissions. However, there are many areas still to be considered, such as the materials of the elements, batteries, resource recycling, etc. In particular, the use of solar batteries or other alternative environment-friendly energy sources must be taken into account.

8.1.1 Materials for elements

In the case of using sensors to monitor the weather or collect environmental information, the environmental load due to sensors has to be minimized. Generally, products emit GHG during all of their life cycles, from raw material acquisition to the final disposal of the products. In addition, environmental load is also caused by the use of harmful raw materials. Therefore, if decommissioned sensor nodes can be collected, GHG emissions can be reduced by reusing or recycling them. In case decommissioned sensor nodes cannot be collected, the environmental load can be minimized if the sensor node is made of environment-friendly materials. Thus, the following should be considered for USN elements:

– using environment-friendly materials for sensor node and related equipment;
– using recyclable materials for sensor node and/or reusable sensor node;
– avoiding hazardous materials for sensor node and related equipment;
– managing the location information of sensor nodes for collection.

8.1.2 Batteries

Batteries and energy resources containing hazardous materials have a serious impact not only on GHG emissions but also on the environment. The fact that sensor nodes are often used in the mobile situation and require frequent battery maintenance, they can cause an unnecessarily large GHG footprint. On the other hand, energy saving and harvesting using environment-friendly resources (such as solar energy) can minimize the GHG footprint of sensor nodes. Thus, one should consider the following concerning the use of batteries in sensor nodes:

– using environment-friendly or rechargeable batteries;
– using high capacity batteries for the reduction of electronic waste;
– using environment-friendly energy sources (e.g., solar energy, electromagnetic energy, thermal energy).

8.2 Energy efficiency

Figure 7 illustrates the typical energy consumption of a sensor node. The total energy used in the sensor node is calculated as the sum of the energy used in each part of the sensor node. Particularly, the energy is most used for communications. Energy used in calculation and other tasks is relatively small. Figure 7 shows that the overall energy efficiency can be drastically increased by designing for low energy consumption communications and by using an energy-efficient operation of communications.

This clause classifies three categories of energy efficiency which should be considered in not only wireless sensor networks but also wired sensor networks.

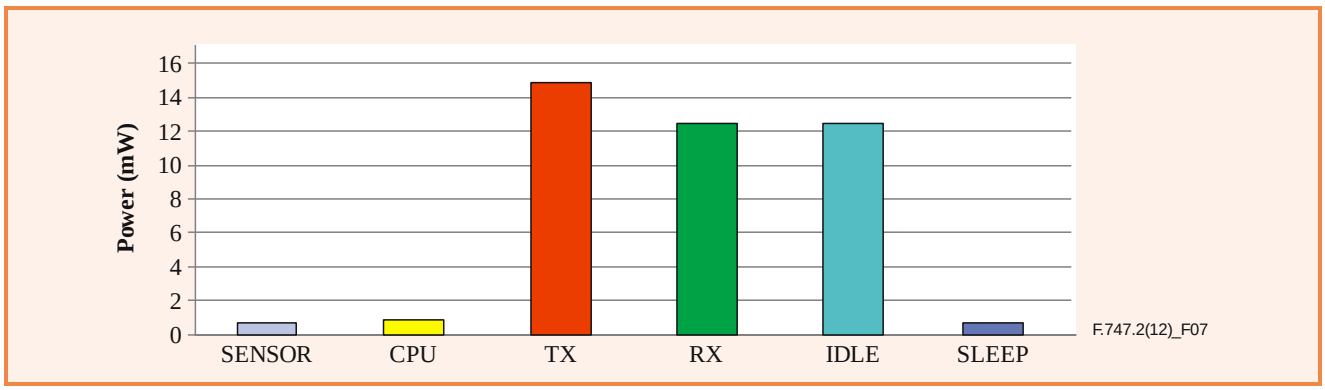


Figure 7 – Energy consumption for a typical sensor reported in [b-IEEE VTC]

8.2.1 Energy efficient hardware setting

Sensor nodes rely mainly on small batteries for their lifetime, all programs must be implemented with a small code size and require minimal power consumption. Power-aware networking and data delivery are a pivotal feature for the longevity of nodes and their batteries, and to reduce electronic waste. Thus, the following considerations apply to hardware settings for sensor nodes:

– To deploy a sufficient number of devices: density and network radius vary amongst different applications and services. Redundant communication may cause the unnecessary consumption of power, and too scarce deployment may cause an unnecessary increase in retransmissions, thus wasting energy.

– To consider radio power and interference, especially obstacles to radio transmissions in an indoor environment.

8.2.2 Energy efficient protocols

For efficient energy consumption, diverse modes (e.g., sleep, idle and hibernate operation modes) and their efficient operation must be supported. The sampling rate for gathering data may be different for each application and service. Thus, protocols for sensor nodes and sensor networks need to consider the following:

– support of diverse modes (e.g., sleep, idle, and hibernate operation modes);

– implementation codes should be as small as possible;

– minimize sensing, calculation and communication;

– equally consume energy on sensor nodes at the same networks;

– support self-recovery, tolerant networks and remote management (movement prevention for frequent maintenance).

8.2.3 Energy efficient applications and services

USN applications and services for different purposes are being developed such as climate monitoring systems and home or building automation systems. Existing USN applications and services create other new services by convergence and they can be used to mitigate climate change. Thus, applications and services for USN have the following considerations:

– to reduce the operations of sensor node;

– to perform the processing load on server;

– to reuse already deployed USN applications and services (when applicable);

– to develop USN applications and services for multi-use of the sensed data (e.g., database schema, API, USN middleware);

– to include the sensor network management function in USN applications and services for automatic checking and the remote reset of USN element malfunction;

– to analyse the application of USN applications and services for energy saving (e.g., control of electric lighting, ventilation, air conditioning and heating).

NOTE – Energy saving should be carefully considered when USN applications and services apply to the facilities which directly relate to people's lives (e.g., surgery room, intensive care unit, emergency room, incubator).

8.3 Operation conditions of GHG sensors

A national GHG monitoring sensor network may have to be established by national regulations, domestic standards, or international standards. They may contain a set of specifications prescribing conditions for geographic locations, target GHGs, sensing frequencies, standard reference GHGs, calculation formulas, meter configurations, device positions, etc. Practitioners should check them before deploying USN applications and services.

Bibliography

[b-GAW programme] WMO Global Atmosphere Watch home page <http://www.wmo.int/pages/prog/arep/gaw/gaw_home_en.html>

[b-IEEE VTC] Ding, M., Cheng, X., and Xue, G. (2003), *Aggregation tree construction in sensor networks*, Vehicular Technology Conference, 2003. Vol.4, No., pp. 2168- 2172, IEEE.

[b-ISO 14064-1] ISO 14064-1 (2006), Greenhouse gases – Part 1: *Specification with guidance at the organization level for quantification and reporting of greenhouse gas emissions and removals*.

[b-IPCC] IPCC Working Group 1 Report (2007), *Glossary of Terms used in the IPCC Fourth Assessment Report*.

[b-IPCC Guidelines] IPCC Guidelines for National Greenhouse Gas Inventories (2006).

Y.4701/H.641

SNMP-based sensor network management framework

SNMP-based sensor network management framework

Summary

Recommendation ITU-T H.641 describes a sensor network management framework intended to provide integrated management functionalities for heterogeneous sensor networks using the simple network management protocol (SNMP).

History

Edition	Recommendation	Approval	Study Group
1.0	ITU-T H.641	2012-02-13	16

Keywords

Sensor network management framework, SNMP.

Table of Contents

Introduction

Ubiquitous sensor network (USN) is a conceptual network built over existing physical networks that makes use of sensed data and provides knowledge services to anyone, anywhere and at any time, and where the information is generated by using context awareness [ITU-T Y.2221]. One of the basic infrastructures of USN is wireless sensor networks that monitor physical or environmental conditions. These sensor networks may use different MAC/PHY protocols and different transport protocols, they may have different sensor node identification schemes and may use different management protocols.

From the viewpoint of a sensor network management protocol, each sensor network may use different sensor network management protocols that are optimized for its MAC/PHY characteristics, transport layer characteristics and for each sensor network management information base (MIB). For example, an IEEE 802.15.4 based sensor network can deliver a 128 bytes frame at one time including 22 bytes of the IEEE 802.15.4 header. It means that a sensor network management protocol for an IEEE 802.15.4-based sensor network should be designed considering this frame size. The ZigBee sensor network based on IEEE 802.15.4 uses its own addressing scheme and defines its own MIB.

Currently, many new sensor networking technologies are under development and it is inevitable that optimized sensor networking technologies for a particular purpose will be deployed. This means that there will be many heterogeneous sensor networks.

From the viewpoint of network management, managing each heterogeneous sensor network with heterogeneous management protocols is impractical and an integrated management protocol for all heterogeneous sensor networks is needed.

The common management information protocol (CMIP) [ITU-T X.711] is widely used in network management systems. However, most TCP/IP devices only support SNMP. SNMP is favoured and strongly supported by vendors, and it has been successfully adapted to manage wired and wireless networks.

Due to the limited computing and communication power of sensor networks, the use of standard SNMP in sensor networks is either impractical or impossible.

Recommendation ITU-T Y.4701/H.641

SNMP-based sensor network management framework

1 Scope

This Recommendation provides an SNMP-based sensor network management framework. The primary purpose of this Recommendation is to describe the framework of integrated sensor network management which can be used to manage heterogeneous sensor networks. The scope of this Recommendation includes:

– overall architecture of framework
– functional entities of framework
– object identifier allocation for MIB
– object identifier translation between SNMP and sensor network management protocol.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T Y.2221] Recommendation ITU-T Y.2221 (2010), *Requirements for support of Ubiquitous Sensor Network (USN) applications and services in the NGN environment*.

[ITU-T Y.2201] Recommendation ITU-T Y.2201 (2009), *Requirements and capabilities for ITU-T NGN*.

[IETF RFC 3411] IETF RFC 3411 (2002), *An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 context awareness [ITU-T Y.2201]: Context awareness is a capability to determine or influence a next action in telecommunication or process by referring to the status of relevant entities, which form a coherent environment as a context.

3.1.2 management information base [IETF RFC 3411]: A collection of managed objects, residing in a virtual information store.

3.1.3 sensor [ITU-T Y.2221]: An electronic device that senses a physical condition or chemical compound and delivers an electronic signal proportional to the observed characteristic.

3.1.4 sensor network [ITU-T Y.2221]: A network comprised of interconnected sensor nodes exchanging sensed data by wired or wireless communication.

3.1.5 sensor node [ITU-T Y.2221]: A device consisting of sensor(s) and optional actuator(s) with capabilities of sensed data processing and networking.

3.1.6 SNMP agent [IETF RFC 3411]: An SNMP entity containing one or more command responder and/or notification originator applications (along with their associated SNMP engine).

3.1.7 SNMP manager [IETF RFC 3411]: An SNMP entity containing one or more command generator and/or notification receiver applications (along with their associated SNMP engine).

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 sensor network management protocol agent: A sensor network management protocol entity containing one or more command responder and/or notification originator applications (along with their associated sensor network management protocol engine).

3.2.2 sensor network management protocol manager: A sensor network management protocol entity containing one or more command generator and/or notification receiver applications (along with their associated sensor network management protocol engine).

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

6LoWPAN	IPv6 over Low power Wireless Personal Area Networks
AIB	Application support Information Base
ALG	Application Level Gateway
CMIP	Common Management Information Protocol
DST	Destination
ID	Identifier
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
MAC	Media Access Control
MIB	Management Information Base
NIB	Network Information Base
OID	Object Identifier
PAN	Personal Area Network
PDU	Protocol Data Unit
PHY	Physical layer
PIB	PAN Information Base
SDO	Standards Development Organization
SNMP	Simple Network Management Protocol
SRC	Source
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
USN	Ubiquitous Sensor Network

5 Conventions

None.

6 Architecture of an SNMP-based sensor network management framework

Figure 1 illustrates the architecture of the SNMP-based sensor network management framework.

The architecture comprises a sensor network manager, sensor network gateway and sensor node. The sensor network manager resides in the TCP/IP network and manages sensor nodes or sensor networks through sensor network gateways.

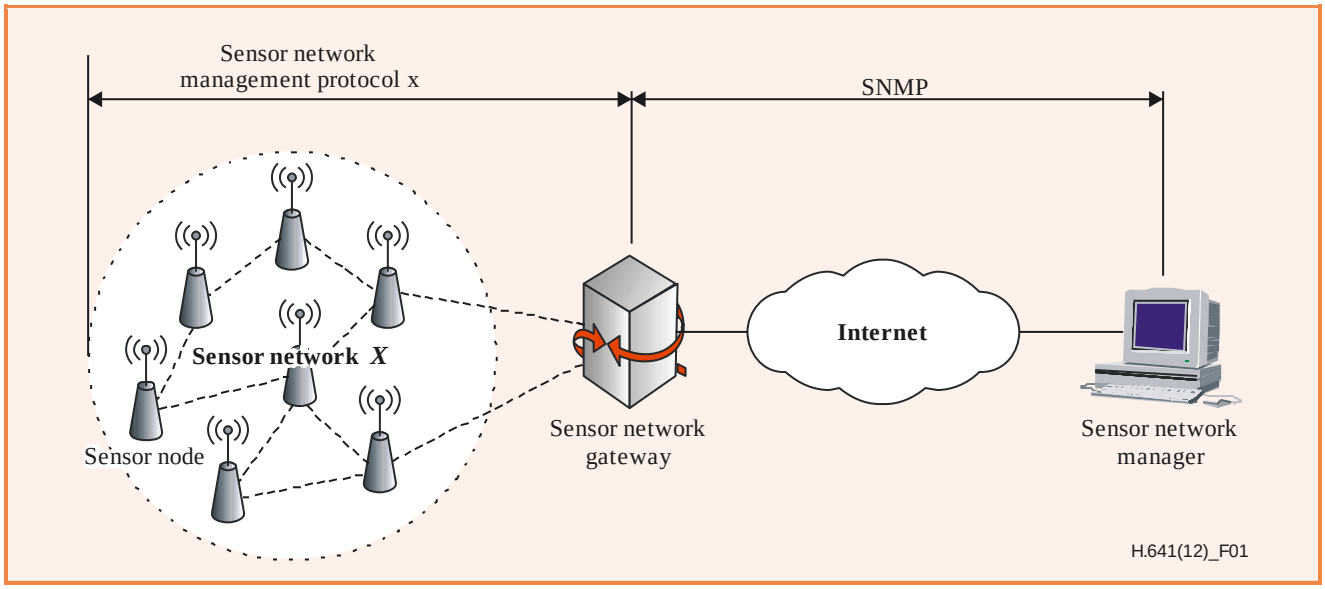


Figure 1 – Architecture of an SNMP-based sensor network management framework

A sensor network manager communicates with the sensor network gateway using standard SNMP over the IPv4 or IPv6.

A sensor network gateway communicates with sensor nodes using a sensor network specific management protocol. Specific sensor network management protocols for specific sensor networks are out of the scope of this Recommendation.

A sensor network gateway has dual network interfaces and performs MAC/PHY protocol translation, transport protocol translation, and management protocol translation between TCP/IP network and specific sensor networks if necessary. The translation of management protocol means translation between SNMP and sensor network specific management protocols. Protocol translations, other than management protocol translation, are out of the scope of this Recommendation.

7 Functional entities of SNMP-based sensor network management framework

Figure 2 illustrates the functional entities of an SNMP-based sensor network management framework.

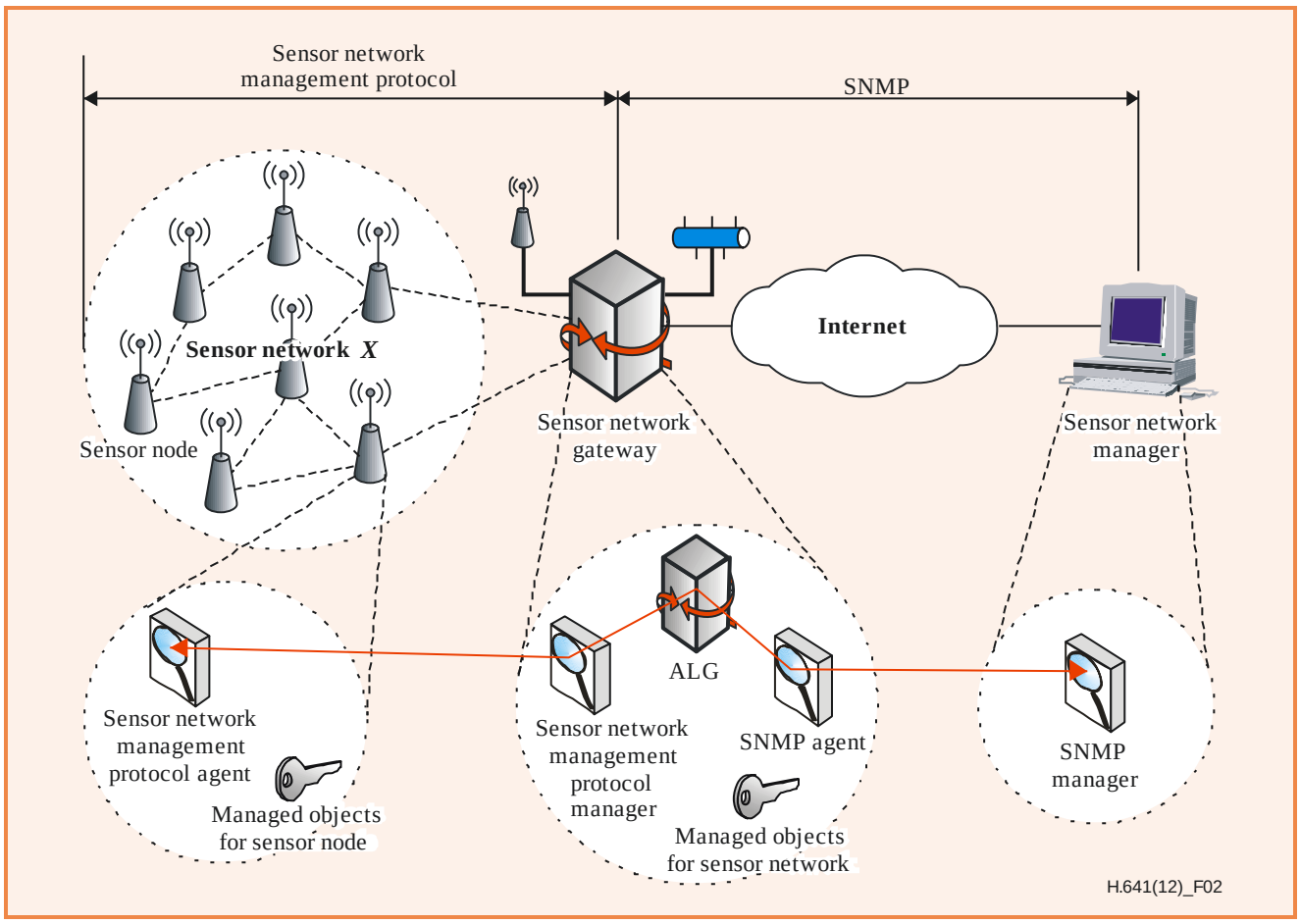


Figure 2 – Functional entities of an SNMP-based sensor network management framework

7.1 SNMP manager

An SNMP manager is a set of management applications that monitor and control network elements such as the sensor network gateway and sensor network elements such as sensor nodes. An SNMP manager uses standard SNMP. Because the sensor network gateway is a normal TCP/IP device, an SNMP manager can manage the sensor network gateway using standard SNMP.

7.2 SNMP agent

An SNMP agent is a set of software modules that reside in network elements. It collects and stores management information such as the number of error packets received by a network element. In this framework, the SNMP agent has two roles, a role as the standard SNMP agent for managing the sensor network gateway and another role of delivering SNMP commands to the application level gateway (ALG) and SNMP responses to the SNMP manager.

7.3 Sensor network management protocol manager

A sensor network management protocol manager is a set of management applications that monitor and control sensor network elements. It receives commands from an ALG and delivers these commands to the sensor network management protocol agent and receives responses from the sensor network management protocol agent and delivers these responses to the ALG.

7.4 Sensor network management protocol agent

A sensor network management protocol agent is a set of software modules that reside in sensor network elements. It collects and stores management information such as the number of error packets received by a sensor network element.

7.5 Application level gateway

An application level gateway is a set of software modules that perform protocol translation between the SNMP and the sensor network management protocol. A sensor network management protocol manager, a sensor network management protocol agent and an ALG may be implemented in one software module, or in separate ones.

7.6 Managed object for sensor network

A managed object for a sensor network is a characteristic of a sensor network that can be managed. The characteristic of a sensor network is mainly related to the whole sensor network (not just for each sensor node) such as the number of sensor nodes in a sensor network. This information is managed by the SNMP agent in a sensor network gateway (see clause 6).

7.7 Managed object for sensor node

A managed object for a sensor node is a characteristic of a sensor node that can be managed. This information is managed by a sensor network management protocol agent in each sensor node. The managed object for a sensor node should be defined by other SDOs developing standards for specific sensor network technology. For example, if the sensor network is a ZigBee sensor network, the managed objects for the sensor nodes could be a ZigBee network information base (NIB) and a ZigBee application support information base (AIB), which are described in [b-ZigBee Specification].

8 Operation of an SNMP-based sensor network management framework

8.1 Overview

Figure 3 illustrates the overall operation flow of an SNMP-based sensor network management framework.

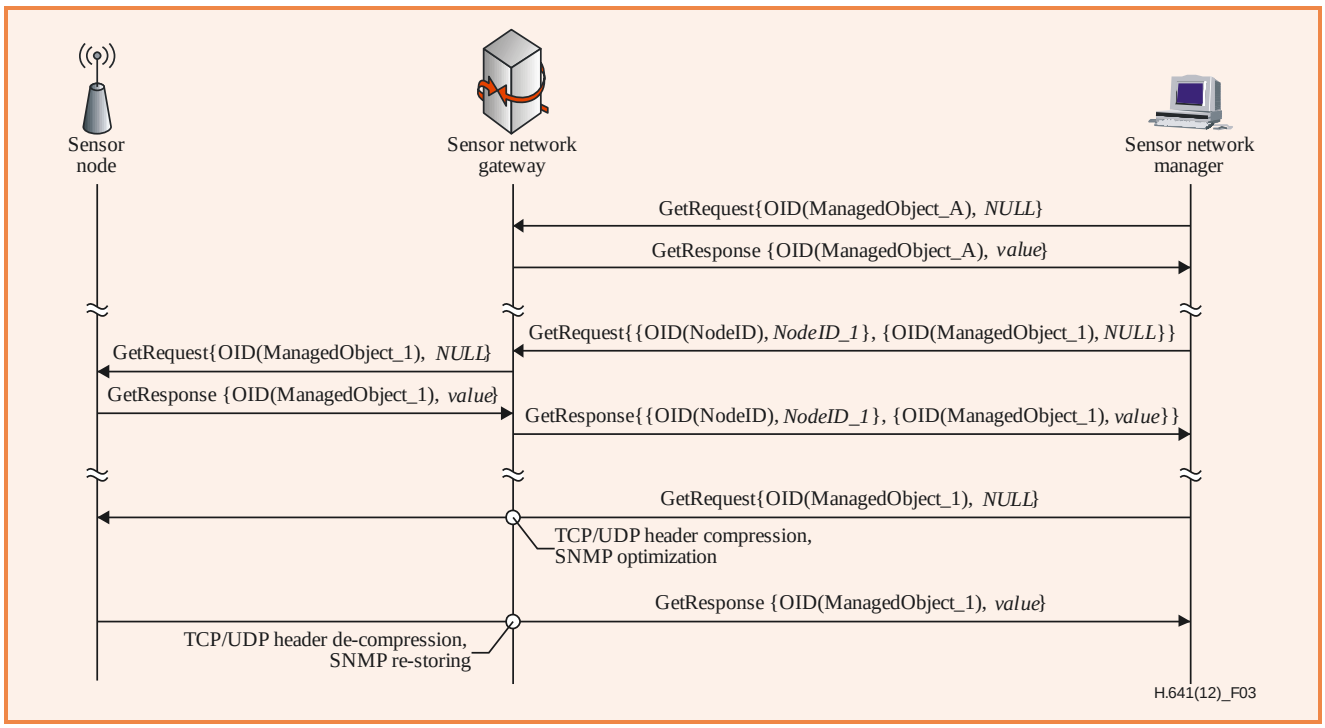


Figure 3 – Operation of an SNMP-based sensor network management framework

To manage a sensor node, a sensor network manager needs to send SNMP requests to a sensor network gateway.

When the SNMP agent receives an SNMP message from the sensor network manager, it should check the OID values in the first *VarBind*. If this OID does not begin with `{itu-t(0) recommendation(0) h(8) h641(641)}` (see Annex A), then it just performs a normal SNMP agent role as specified in [IETF RFC 3411].

If this OID begins with `{itu-t(0) recommendation(0) h(8) h641(641)}`, it passes the SNMP message to the ALG. In this message, the sensor node ID should be included as a *VarBind*. The ALG in the sensor network gateway converts this SNMP message to a sensor network management protocol message and sends it to a sensor node which is identified by a sensor node ID in *VarBind*. In this conversion process, the ALG removes the first *VarBind* from the *VarBindList*. The value of the first *VarBind* (sensor node ID) is used for the destination address in the sensor network protocol.

A sensor node management protocol agent replies back to the sensor network gateway using the sensor network management protocol. When the ALG in the sensor network gateway receives this message, the ALG converts this sensor network management protocol message to an SNMP message. In this conversion process, the sensor network gateway adds *VarBind* to the *VarBindList*. The value of this *VarBind* can be acquired from the source address of the received sensor network management protocol message or from a mapping table managed by the ALG in the sensor network gateway.

The ALG in the sensor network gateway also performs object identifiers (OIDs) conversion between the sensor network management protocol messages and SNMP messages (see clause 9 and Appendix I).

8.2 Application level gateway database

The sensor network gateway should maintain an ALG database. It is constructed with the information listed below for every SNMP message from the SNMP manager in order to perform protocol and OID translation between the SNMP and the sensor network management protocol.

– Request ID sequence number (generated by ALG)
– IP header (from SNMP message)
– UDP header (from SNMP message)
– Common SNMP header (from SNMP message)
– Get/set header (from SNMP message)
– Variable binding list (from SNMP message)

The sequence number shall be used as a primary key in this ALG database.

8.3 Translation from an SNMP to a sensor network management protocol message

Figure 4 shows the operational procedure of the sensor network gateway when it receives an SNMP message from an SNMP manager.

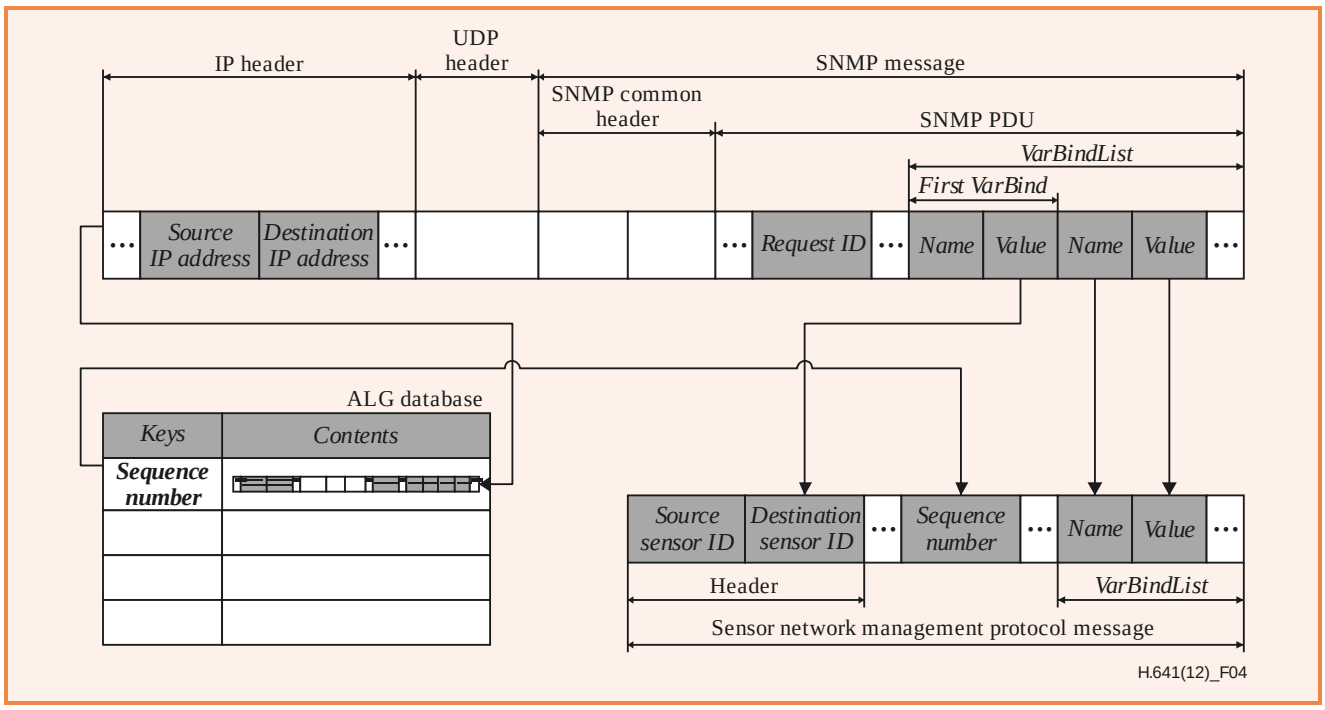


Figure 4 – Translation from an SNMP message to a sensor network management protocol message

When the sensor network gateway receives an SNMP message from an SNMP manager, it generates a sequence number to be used as the request ID in the sensor network management protocol message and records this sequence number and the whole message into an ALG database. Then it creates a sensor network management protocol message. It copies the sequence number in the ALG database into the sequence number field of the sensor network management protocol message. It copies the value of the first *VarBind* into the destination sensor ID field of the sensor network management protocol message. Then it copies the *VarBindList* except for the first one into the sensor network management protocol message. When it copies object identifiers in the *VarBindList*, base OID should be removed (see clause 9 and Appendix I).

The sensor network management protocol message should include a request ID field. If this field does not use the same format with an SNMP message, then the sensor network gateway also maintains mapping information between the request ID from the SNMP message and request ID from the sensor network management protocol message.

8.4 Translation from a sensor network management protocol to an SNMP message

Figure 5 shows the operational procedure of the sensor network gateway when it receives a sensor network management protocol message from a sensor network management protocol agent.

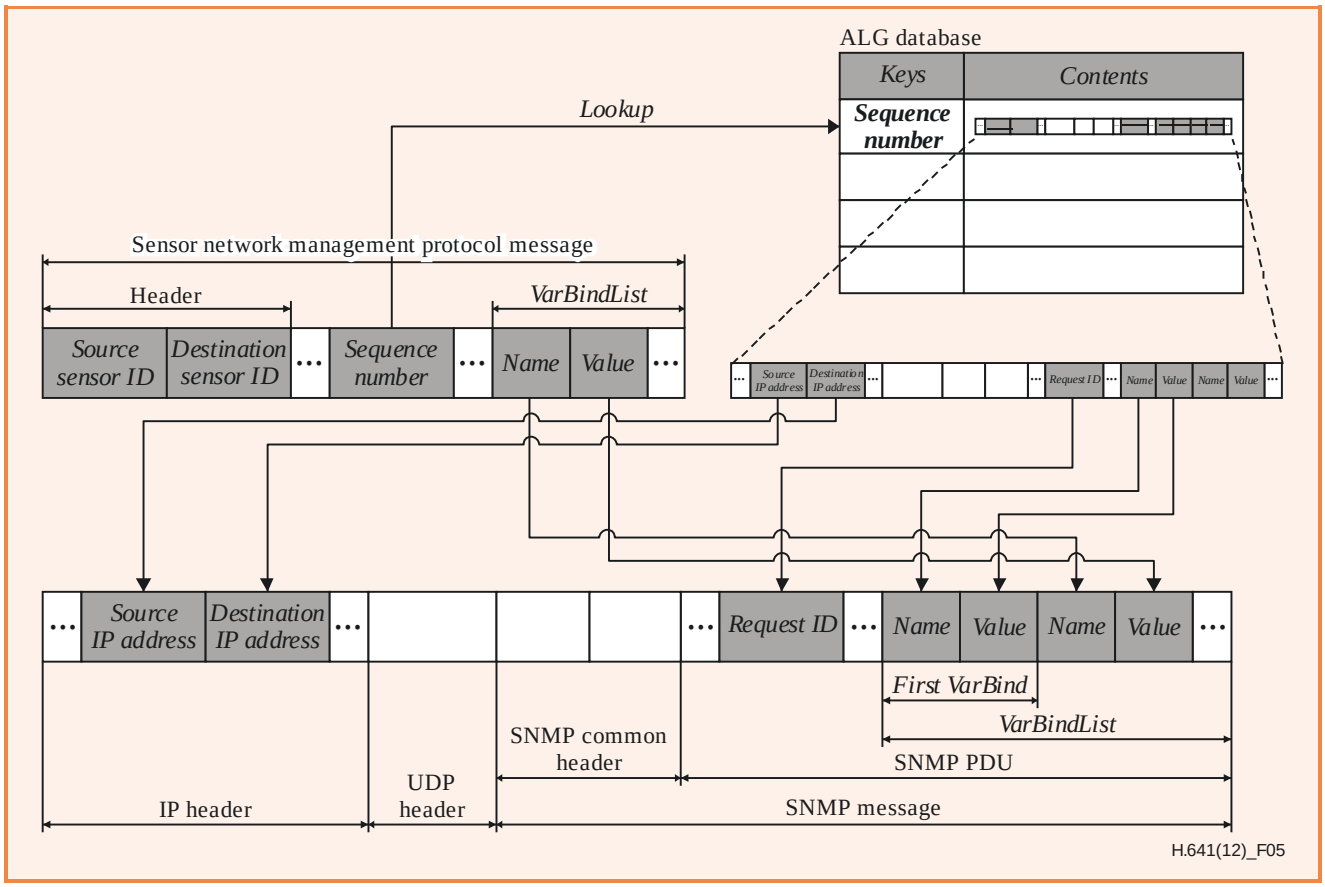


Figure 5 – Translation from a sensor network management protocol message to an SNMP message

When the sensor network gateway receives a sensor network management protocol message from a sensor node, it extracts the sequence number and looks up the original SNMP message from the ALG database. Then it creates a new SNMP message. It copies the destination IP address of the original SNMP message into the source IP address field. It then copies the source IP address of the original SNMP message into the destination IP address field. Then it copies the request ID and the first *VarBind*. It copies the *VarBindList* from the sensor network management protocol message into the *VarBindList* of the new SNMP message. When it copies the OID in the *VarBindList*, base OID should be added (see clause 9 and Appendix I).

8.5 Consideration of the sensor network gateway for supporting IP-based sensor networks

In case an IP-based sensor network is deployed (e.g. 6LoWPAN), the sensor network gateway (6LowPAN gateway) can be expected to inspect the data packets traversing the sensor network gateway. If the destination IP address of an IP header of an SNMP message is the IP address of the sensor node that resides under the sensor network gateway, the conversion of an SNMP message described in clause 8.3 is not needed. However, the sensor network gateway can perform TCP/UDP header compression and SNMP optimizations.

When the sensor network gateway receives an SNMP message that is bound for the sensor network manager, the sensor network gateway may perform TCP/UDP header de-compression and SNMP message re-storing if TCP/UDP header compression and SNMP optimizations are performed on the received SNMP message.

9 Object identifier allocation for MIB and object identifier translation between SNMP and sensor network management protocols

In SNMP, managed objects are identified by OIDs [b-ITU-T X.660] that are relatively long byte strings to be transferred on a sensor network. Considering the low data rate of sensor networks, relative OIDs are used in sensor network management protocols so as to identify managed objects relative to the base OID `{itu-t(0) recommendation(0) h(8) h641(641) sensor-network-mgt(2) n}` (see Annex A).

When a sensor network gateway translates an SNMP message into a sensor network management protocol message, the sensor network gateway can remove the base OID from the OID for MIB in *VarBind*. When a sensor network gateway translates a sensor network management protocol message into an SNMP message, the sensor network gateway inserts the base OID in front of the OID of the MIB in *VarBind* (see Appendix I).

Annex A

Object identifier assignments

(This annex forms an integral part of this Recommendation.)

Table A.1 lists the assignment of OIDs defined for use by this Recommendation.

Table A.1 – Object identifier assignments

Object Identifier Value	Description
`{itu-t(0) recommendation(0) h(8) h641(641) sensor-network-mgt(2) ieee-802-15-4(1))}`	This OID is used to indicate that the sensor network conforms to [b-IEEE 802.15.4]. Subsequent arcs of this node are identical to the identifiers of PHY and MAC PIB attributes defined in [b-IEEE 802.15.4].
`{itu-t(0) recommendation(0) h(8) h641(641) sensor-network-mgt(2) zigbee(2)}`	This OID is used to indicate that the sensor network conforms to [b-ZigBee Specification]. Subsequent arcs of this node are identical to the identifiers of the network layer information base, application layer information base and security-related application layer information base attributes defined in [b-ZigBee Specification].

Appendix I

Example of object identifier translation

(This annex forms an integral part of this Recommendation.)

Figure I.1 shows the translation of an OID in a sensor network gateway.

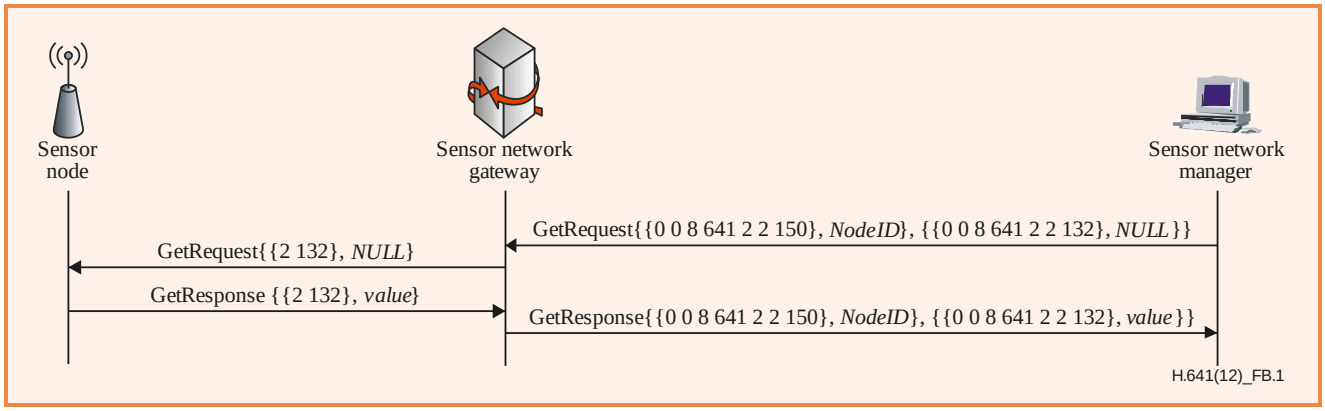


Figure I.1 – Example of object identifier translation

In Figure I.1, the sensor network is a ZigBee sensor network. When the sensor network manager wants to get the `nwkMaxChildren` of a ZigBee node, it sends a *GetRequest* message to the sensor network gateway with OID `{itu-t(0) recommendation(0) h(8) h641(641) sensor-network-mgt(2) zigbee(2) nwkMaxChildren(132)}` in a *VarBind*. When the sensor network gateway receives an SNMP message from the sensor network manager, it removes the base OID so as to create the relative OID `{zigbee(2) nwkMaxChildren(132)}` for the sensor network management protocol message.

NOTE – The OID `{itu-t(0) recommendation(0) h(8) h641(641) sensor-network-mgt(2) zigbee(2) nwkNetworkAddress(150)}` is allocated for nwkNetworkAddress of the ZigBee node and this is a 16-bit address used in a ZigBee network.

When the sensor network gateway receives a sensor network management protocol message from a sensor node, it inserts the base OID in front of the relative OID `{zigbee(2) nwkNetwork Address(150)}` so as to create the full OID `{itu-t(0) recommendation(0) h(8) h641(641) sensor-network-mgt(2) zigbee(2) nwkMaxChildren(132)}` for the sensor network management protocol message.

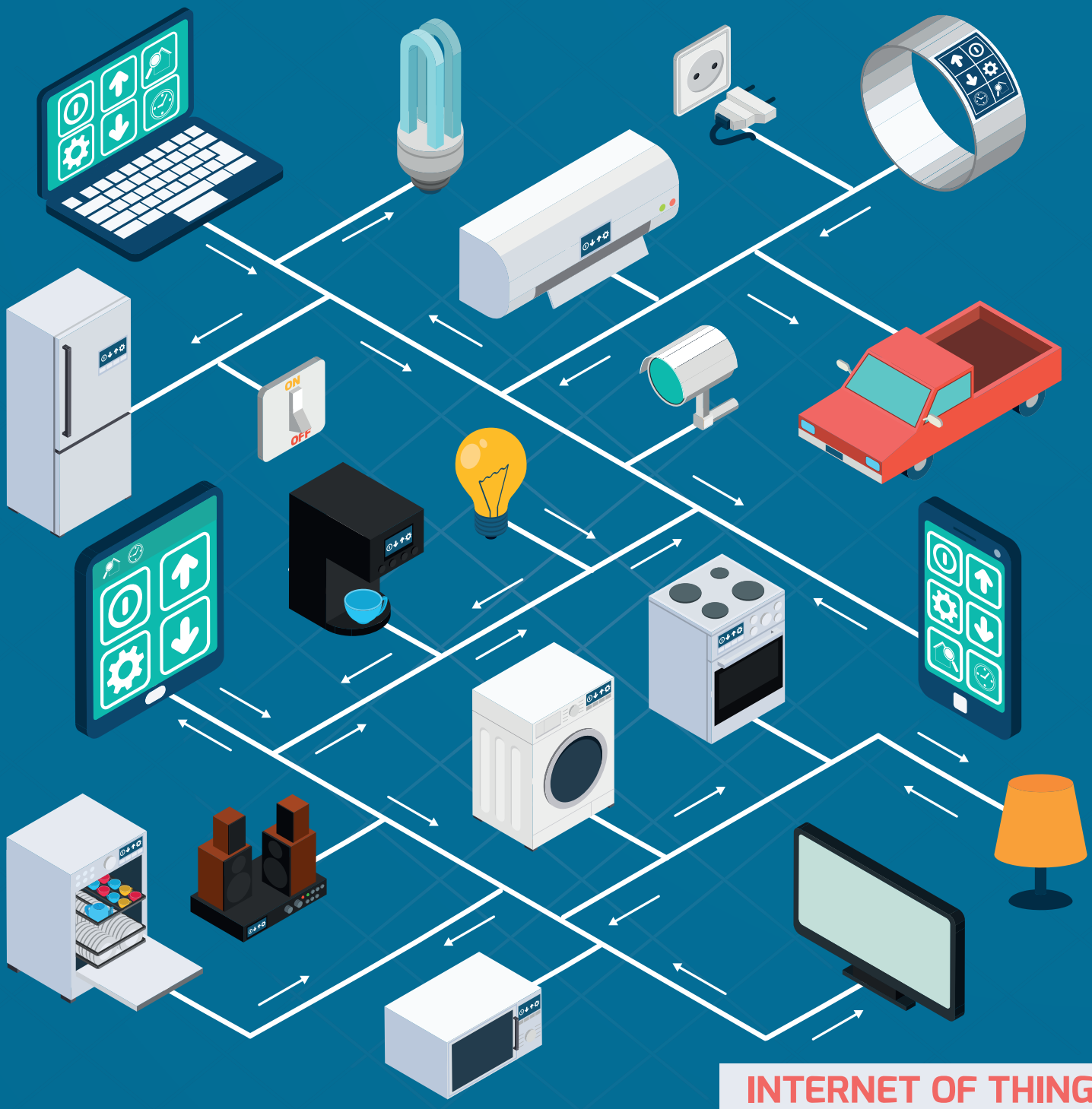
Bibliography

[b-ITU-T X.660] Recommendation ITU-T X.660 (2011) | ISO/IEC 9834-1:2012, *Information technology – Procedures for the operation of object identifier registration authorities: General procedures and top arcs of the international object identifier tree.*

[b-ITU-T X.711] Recommendation ITU-T X.711 (1997) | ISO/IEC 9596-1:1997, *Information technology – Open Systems Interconnection – Common Management Information Protocol: Specification*.

[b-IEEE 802.15.4] IEEE 802.15.4 (2006), *IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements, Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs).*

[b-ZigBee Specification] ZigBee Document 053474r17 (2008), *ZigBee Specification.*



INTERNET OF THINGS


Internet
Of Things

Y.4702

Common requirements and capabilities of device management in the Internet of things

Common requirements and capabilities of device management in the Internet of things

Summary

Recommendation ITU-T Y.4702 provides the common requirements and capabilities of device management (DM) in the Internet of things (IoT).

The provided common requirements and capabilities are intended to be generally applicable in device management application scenarios.

History

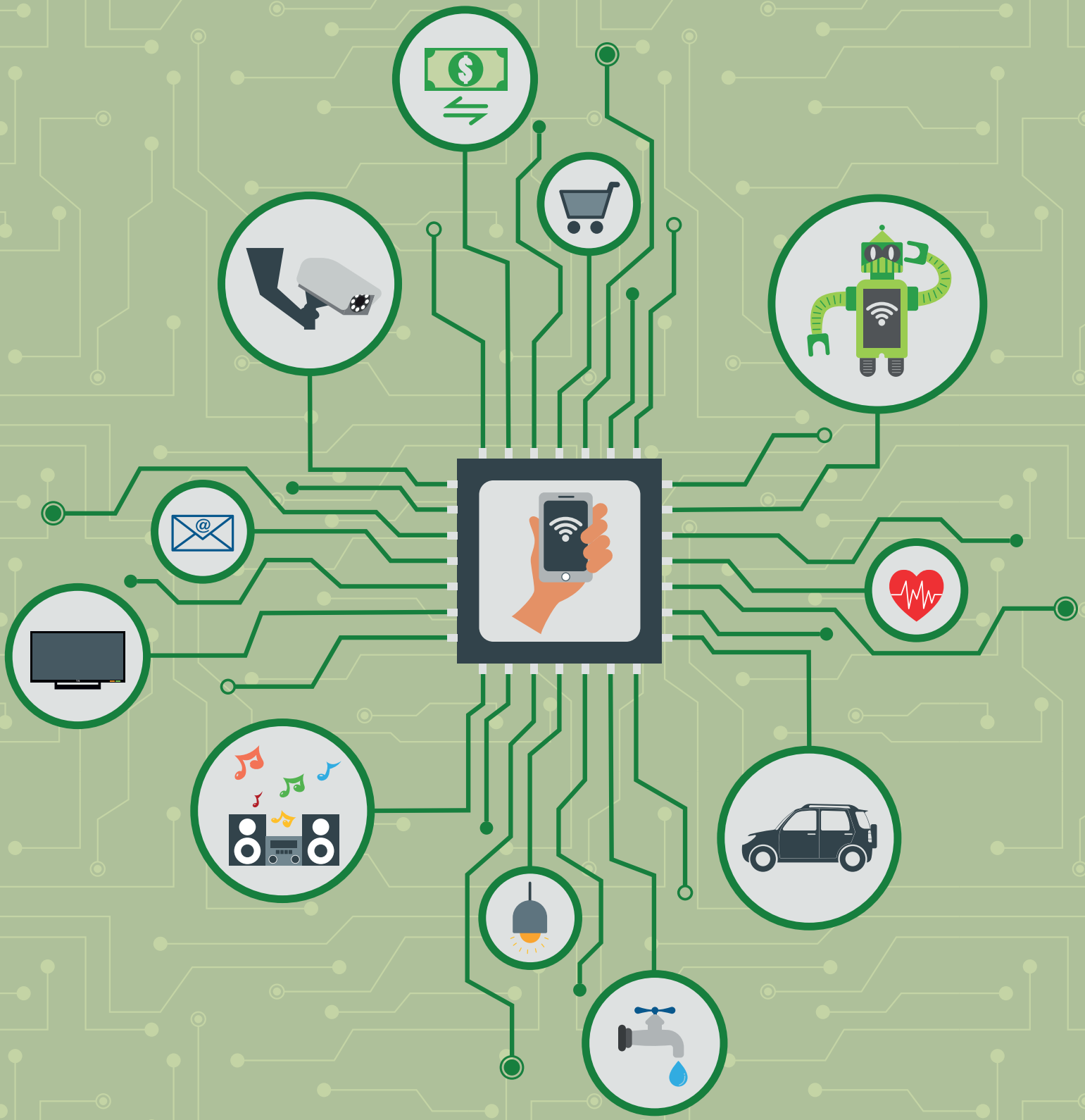
Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T Y.4702	2016-03-15	20	11.1002/1000/12780

Keywords

Common requirements, common capabilities, device management, DM, Internet of things (IoT).

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

Table of Contents



Recommendation ITU-T Y.4072

Common requirements and capabilities of device management in the Internet of things

1 Scope

This Recommendation provides the common requirements and capabilities of device management (DM) in the Internet of things (IoT).

The provided common requirements and capabilities are intended to be generally applicable in device management application scenarios.

The scope of this Recommendation includes:

1) Common requirements of device management in the IoT.

2) Common capabilities of device management in the IoT.

NOTE – This Recommendation focuses on the requirements of DM for the interaction between devices and the various DM functional components. The DM client, a functional component optionally present in some IoT applications, and its specific requirements are outside of the scope of this Recommendation.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T Y.4000] Recommendation ITU-T Y.4000/Y.2060 (2012), *Overview of Internet of things.*

[ITU-T Y.2061] Recommendation ITU-T Y.2061 (2012), *Requirements for the support of machine-oriented communication applications in the next generation network environment.*

[ITU-T Y.2066] Recommendation ITU-T Y.4100/Y.2066 (2014), *Common requirements of the Internet of things.*

[ITU-T Y.2067] Recommendation ITU-T Y.4101/Y.2067 (2014), *Common requirements and capabilities of a gateway for Internet of things applications*.

[OMA-RD-LightweightM2M] OMA-RD-LightweightM2M (2013), *Lightweight Machine to Machine Requirements.*

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 device [ITU-T Y.4000]: With regard to the Internet of things, this is a piece of equipment with the mandatory capabilities of communication and the optional capabilities of sensing, actuation, data capture, data storage and data processing.

3.1.2 gateway [ITU-T Y.2067]: A unit in the Internet of things which interconnects the devices with the communication networks. It performs the necessary translation between the protocols used in the communication networks and those used by devices.

3.1.3 Internet of things (IoT) [ITU-T Y.4000]: A global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies.

NOTE 1 – Through the exploitation of identification, data capture, processing and communication capabilities, the IoT makes full use of things to offer services to all kinds of applications, whilst ensuring that security and privacy requirements are fulfilled.

NOTE 2 – From a broader perspective, the IoT can be perceived as a vision with technological and societal implications.

3.2 Terms defined in this Recommendation

This Recommendation defines or uses the following terms:

3.2.1 Device management (DM) agent: The DM functional component responsible for collecting DM-related information from devices and gateways, reporting device related information to the DM manager or DM GW manager and analysing the commands from the DM manager or DM GW manager in order to execute DM-related tasks.

3.2.2 Device management (DM) client: The DM functional component optionally present in some IoT applications and interacting with the DM manager implemented by specific capabilities of the IoT SSAS capability set [ITU-T Y.4000]. It provides access to DM capabilities in order to enable device management functionalities in IoT applications.

3.2.3 Device management (DM) gateway (GW) manager: The DM functional component responsible for managing devices connected to a given gateway (GW).

3.2.4 Device management (DM) manager: With regard to device management in IoT, the DM functional component, responsible for managing devices and gateways.
NOTE – The device management manager interacts with other DM functional components to get DM-related information of devices and gateways and to send commands for execution of DM-related tasks. According to the IoT application deployment scenarios, it may be implemented by DM capabilities of the IoT service support and application support (SSAS) capability set [ITU-T Y.4000] or by the IoT applications themselves.

4 Abbreviations and acronyms

3G Third Generation

4G Fourth Generation

ADSL Asymmetric Digital Subscriber Line

API Application Programming Interface

CPU Central Processing Unit

DM Device Management

GPRS General Packet Radio Service

GW Gateway

ID Identifier

IoT Internet of Things

IP Internet Protocol

LAN Local Area Network

OS Operating System

OSS Operation Support System

QoS Quality of Service

SMS Short Message Service

SSAS Service Support and Application Support

WiFi Wireless Fidelity

5 Conventions

In this Recommendation:

– The keywords "is required to" indicate a requirement which must be strictly followed and from which no deviation is permitted if conformance to this document is to be claimed.

– The keywords "is recommended" indicate a requirement which is recommended but which is not absolutely required. Thus this requirement need not be present to claim conformance.

– The keywords "can optionally" and "may" indicate an optional requirement which is permissible, without implying any sense of being recommended. These terms are not intended to imply that the vendor's implementation must provide the option and the feature can be optionally enabled by the network operator/service provider. Rather, it means the vendor may optionally provide the feature and still claim conformance with the specification.

– The term "IoT applications" is used to identify applications which are operated over the IoT infrastructure.

6 Introduction

Device management (DM) is an essential set of management capabilities in the Internet of things (IoT), providing support for, but not being limited to, devices' remote activation and de-activation, diagnostics, firmware/software updating and sensor node working status management [ITU-T Y.4000].

DM provides a set of capabilities through which the users of these capabilities can exercise various management tasks for devices, locally and remotely. Via DM, the devices in IoT can be correctly configured and can correctly and efficiently operate through standardized interfaces and procedures. In addition, the users of these capabilities can know the current status of devices and get notification if there is something wrong with the devices. The ultimate aim of DM is to make sure that the applications running on a given device operate well.

Traditionally, IoT applications interact directly with devices (and gateways) to realize DM functionalities. Nowadays, with the advent of service support and application support (SSAS) capabilities [ITU-T Y.4000] in the IoT infrastructure, the DM functionalities can be provided as a common service to IoT applications. IoT applications may directly use the DM service provided by DM capabilities of the SSAS capability set in order to realize DM functionalities without interacting with devices (and gateways) directly. This is a simpler and more efficient way to realize DM functionalities for IoT applications.

In the IoT, the DM functional components can be deployed in devices, gateways, components which provide SSAS capabilities and IoT applications.

Four DM functional components are identified: DM manager, DM agent, DM gateway (GW) manager and DM client. Figure 1 shows the positioning of these components in the IoT from the layering perspective of the IoT reference model [ITU-T Y.4000].

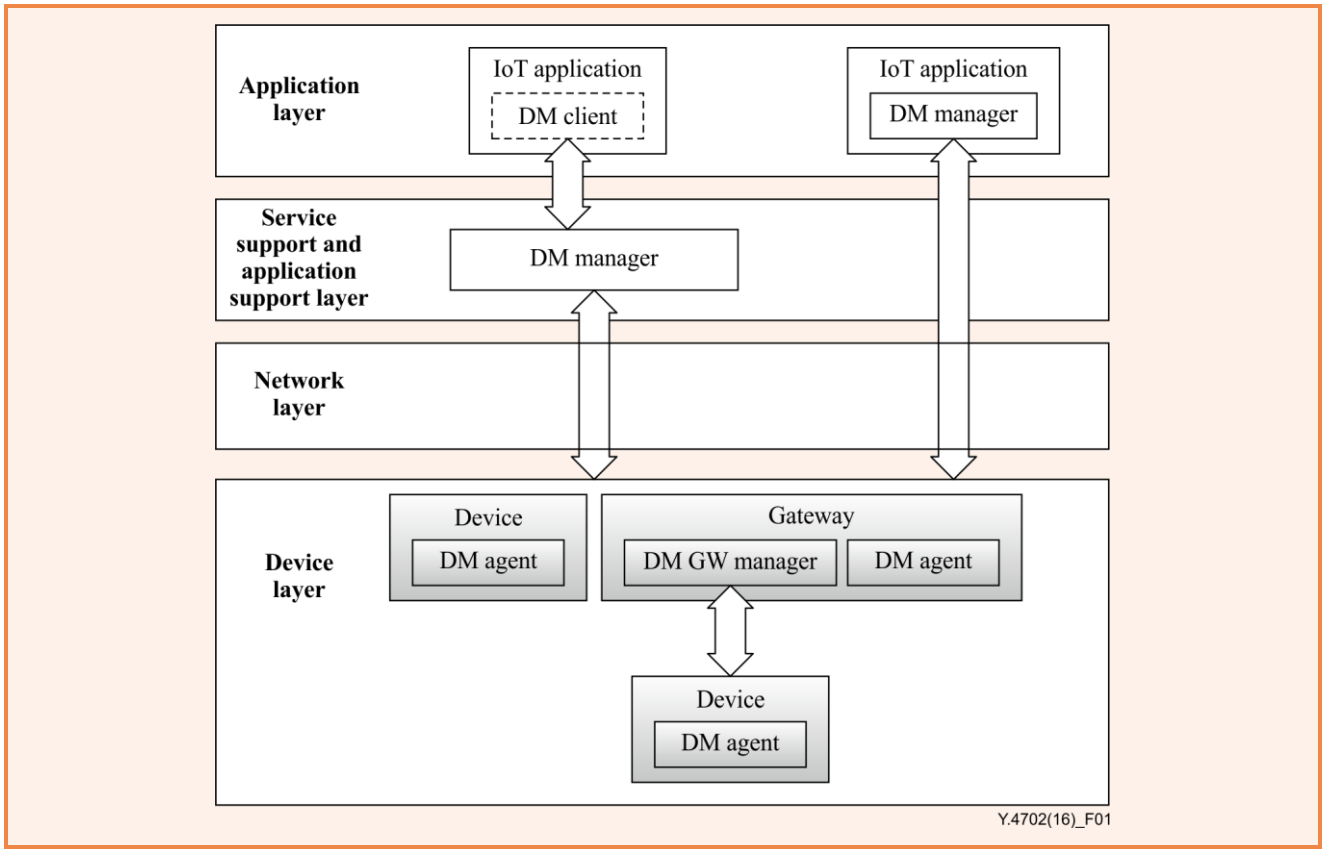


Figure 1 – The DM functional components in the IoT from a layering perspective

The DM manager is a functional component, responsible for managing devices and gateways. The DM manager interacts with DM agents and DM GW managers to get DM-related information of devices and gateways and sends them commands for execution of DM-related tasks. According to the IoT application deployment scenarios, a DM manager may be implemented by DM capabilities of the IoT SSAS capability set [ITU-T Y.4000] or by the IoT applications themselves.

NOTE 1 – When implemented by DM capabilities of the IoT SSAS capability set, the DM Manager provides DM capabilities to IoT applications as a common service, enabling an indirect way of interaction between IoT applications and devices (and gateways).

The DM agent is responsible for collecting DM-related information local to devices and gateways, reporting status and fault information local to devices and gateways and analysing the commands from the DM manager or DM GW manager in order to execute DM-related tasks.

The DM GW manager is responsible for managing devices connected to a given gateway. The DM GW manager acts as a proxy between devices and the DM manager, so that the DM manager and DM agent interact with each other through the DM GW manager.

The DM client is the functional component optionally present in some IoT applications and interacting with the DM manager implemented by specific capabilities of the IoT SSAS capability set. It provides access to DM capabilities in order to enable device management functionalities in IoT applications.

NOTE 2 – This Recommendation focuses on the requirements of DM for the interaction between devices and the various DM functional components. The DM client and its specific requirements are outside of the scope of this Recommendation.

As far as deployment is concerned, the various DM functional components are deployed as follows:

– The DM agents are deployed in devices and gateways.

– The DM GW managers are deployed in gateways.

According to the IoT application deployment scenarios, the DM manager may be deployed in IoT applications or in components which provide SSAS capabilities.

7 Requirements of device management in the IoT

7.1 Characteristics specific to device management in the IoT

Device management in IoT has some specific characteristics (implying corresponding requirements to be met).

The following specific characteristics are identified:

1) In some IoT application scenarios it is necessary to manage a large number of devices [ITU-T Y.2066].

2) It is frequent to have IoT application scenarios where a lot of devices are connected to the same gateway, with their DM agents interacting with the DM GW manager located in that gateway. In some of these cases, these devices are managed by the DM GW manager directly, in other ones they are managed by the DM manager through the DM GW manager.

   NOTE 1 – A gateway manages devices based on gateway policies or instructions received from IoT applications.

3) Different devices may have different capabilites. Resource constrained devices have limited computing capabilities (e.g., small central processing unit (CPU), small memory size, limited battery) or constrained communication capabilities (such as general packet radio service (GPRS)): these devices are not often able to support full DM functionalities. Other devices have rich computing capabilities (e.g., large CPU, large memory size) or rich communication capabilities (e.g., Third Generation (3G), Fourth Generation (4G), wireline) and are able to support full DM functionalities.

4) It is frequent to have IoT application scenarios where a lot of devices are powered by battery. These devices often run in power-saving modes, such as sleeping mode [ITU-T Y.2061].

5) Different devices may use different communication technologies, such as wireless access networks (e.g., GPRS, 3G, 4G, wireless fidelity (WiFi)) and wireline access networks (e.g., asymmetric digital subscriber line (ADSL), local area network (LAN) and power line) [ITU-T Y.2066]. For devices characterized by a small amount of transmitted data, bandwidth constrained wireless networks are often used, such as GPRS, etc. In these cases, DM protocols should be simple and concise.

6) There are IoT application scenarios where it is not only necessary to communicate with devices to retrieve device related information, but where it is also necessary to communicate with the network to get connectivity status information.

   NOTE 2 – Device related information includes information about operating status, configuration, fault and performance of device hardware, peripherals, operating system (OS), communication service, DM services, applications running on the device, etc.

7) There are IoT application scenarios with not only multi-purpose devices, but also dedicated devices. Multi-purpose devices may interact with multiple IoT applications via the SSAS capabilities of the IoT infrastructure. Dedicated devices, from their initial activation, support only a single IoT application, without supporting other applications in the future.

7.2 Common requirements of device management in the IoT

The common requirements of device management in the IoT are as follows:

1) Each device managed by the DM manager is required to be identified or addressable in order to be recognized and managed.

2) The scope of the device identification scheme is recommended to be large enough to support scenarios with a huge number of devices to be managed.

3) The DM manager and DM GW manager are required to manage devices independently of the devices' capabilities, including devices having different DM capabilities.

4) It is required to enable open access to the DM capabilities.

   NOTE 1 – The open access to the DM capabilities allows some IoT applications to directly use the DM services provided by the DM capabilities of the SSAS capability set in order to realize DM functionalities and administrators of such IoT applications to also use the DM services through a web portal.

5) In case of open access to the DM capabilities, it is required to provide open application programming interfaces (APIs) for access by IoT applications.

6) It is recommended that the devices be manageable by groups and that different ways to group devices be supported, including by device identifier (ID), location, software version and application type [ITU-T Y.2061].

7) The DM manager and DM GW manager are recommended to be robust enough to support a large number of devices accessing the DM manager simultaneously.

8) It is recommended that different devices have different DM service access levels.

   NOTE 2 – Devices with low service access levels may be denied access to DM services by the DM manager or DM GW manager when a large number of devices access the service at the same time.

9) It is recommended that the DM manager and DM GW manager use scheduling mechanisms in order to avoid communication congestion issues.

10) It is required to support mechanisms for device discovery when devices connect to the DM GW manager or DM manager for the first time.

11) It is required to support mechanisms for device capability discovery [OMA-RD-LightweightM2M].

12) It is required to support mechanisms for service discovery when new services are published by applications in devices [ITU-T Y.2067].

13) It is required to support mechanisms for device registration and different registration modes, including active registration by devices, manual registration by administrators, etc.

14) It is recommended to support mechanisms for device software/firmware image management, including image inventory management, update results reporting, integrity check of image before update process, update process monitoring and fallback when update fails [ITU-T Y.2061], [ITU-T Y.2067].

15) It is recommended to support manager initiated mode and/or device initiated mode for device software/firmware image fallback mechanism.

   NOTE 3 – For the manager initiated mode, the DM manager sends a request to devices to go back to the former software/firmware image version when it discovers problems concerning the devices. For the device initiated mode, the devices request restoration of the former software/firmware image version when they discover that the software/firmware update has failed.

16) It is recommended to enable activation and de-activation of the reporting of device related information from devices to the DM manager or DM GW manager.

17) It is recommended that some configuration parameters of a device with its DM agent directly interacting with the DM manager can be set locally and remotely by the DM manager.

18) It is recommended that the DM manager and DM GW manager be able to upload and download DM-related files to/from devices.

NOTE 4 – Examples of DM-related files are the device configuration file and the device log file.

19) It is required to support factory reset of devices [OMA-RD-LightweightM2M].

20) It is required that the DM manager and DM GW manager be able to get device related information.

NOTE 5 – Such information can be reported by devices and/or retrieved by the DM manager and DM GW manager from devices [ITU-T Y.4000].

21) It is recommended to support plug and play mechanisms for initialization of devices [ITU-T Y.4000], [b-ITU-T Y.4112].

22) It is recommended to enable the provisioning of not only the current device related information for a specified range of devices, but also historical device related information.

23) It is recommended to enable device reporting of DM-related events, and different reporting mechanisms be supported (e.g., configurable time-based mechanism, configurable threshold-based mechanism).

NOTE 6 – Examples of DM-related events include some key configuration parameters being changed locally, or CPU load exceeding the threshold.

24) If supported by devices, the DM manager and DM GW manager are required to support device reporting's policy setting, including what kind of device related information should be reported and when to report.

25) It is required that devices, any service running on devices and the device peripherals can be activated and de-activated locally and remotely by the DM manager [ITU-T Y.4000], [ITU-T Y.2066], [OMA-RD-LightweightM2M].

26) It is required that the DM manager be able to remotely restart devices, any service running on devices and device peripherals [OMA-RD-LightweightM2M].

27) It is recommended that the DM manager be able to support a mechanism to trigger the device establishment of an application level connection to the DM manager.

28) It is recommended that the device power status be reported at a certain frequency to the DM manager or DM GW manager if a device is powered by battery.

29) It is recommended that the DM manager and DM GW manager be able to request to set the power saving mode for a device if the device supports such a mode.

30) It is recommended that the DM manager be able to obtain location information of devices.

NOTE 7 – Device location information can be obtained from devices or from the network.

31) It is recommended that the DM manager and DM GW manager be able to receive and process results of self-diagnostics reported by devices.

32) It is recommended that the DM manager and DM GW manager provide diagnostic analysis results.

33) DM protocols are recommended to be simple and concise, if bandwidth constrained wireless networks are used and/or devices are resource constrained [ITU-T Y.2061].

34) DM protocols are recommended to support compression mechanisms if bandwidth constrained wireless networks are used [ITU-T Y.2061].

35) If some abnormal condition occurs, it is recommended for the DM manager and DM GW manager to execute a diagnostic device fault location, isolation and restoration procedure.

36) It is recommended that trouble tickets be generated by proactive failure detection of devices and be reported to the DM capabilities' users.

37) It is required to be able to get network connectivity information from network(s) and/or device connectivity from devices.

NOTE 8 – Network connectivity information includes whether the device has connected to the network, identification of the radio cell in case of cellular network connectivity, etc.

38) It is required for the DM manager and DM GW manager to support a mechanism to prohibit the connection of IoT devices to the network for a certain duration [OMA-RD-LightweightM2M].

39) It is required to support a mechanism to retrieve the connection log information from devices [OMA-RD-LightweightM2M].

40) It is recommended for the DM manager and DM GW manager to support mechanisms to control the device access to the network based on time and/or location.

41) In the case of open access to the DM capabilities, it is recommended to implement accounting management based on access time of DM services, use of DM services and number of managed devices.

42) It is recommended for the DM manager and DM GW manager to support accounting mechanisms to collect DM service usage data, to calculate accounting information.

43) It is recommended that the ability to log DM-related operations be supported.

44) It is recommended that the configuration parameters of a device with its DM agent indirectly interacting with the DM manager via the DM GW manager be obtained and/or set by the DM GW manager or the DM manager through the DM GW manager.

45) It is recommended that for a device with its DM agent indirectly interacting with the DM manager via the DM GW manager, any service running on that device and that device's peripherals be activated and de-activated locally and remotely by the DM GW manager or DM manager through the DM GW manager.

46) It is required that the DM GW manager be able to remotely restart a device with its DM agent indirectly interacting with the DM manager via the DM GW manager, any service running on that device and that device's peripherals [OMA-RD-LightweightM2M].

47) It is recommended to support mechanisms to remotely lock or erase contents of IoT devices (e.g., in order to protect sensitive personal information from the possibility of loss/theft of a device).

48) It is required to support different levels of security according to the IoT applications' requirements.

NOTE 9 – Low resource-consumption security mechanisms should be used for resource constrained devices.

49) It is required to support device integrity checking [ITU-T Y.2066].

50) It is recommended to support mutual authentication between DM functional components.

51) It is recommended to support non-repudiation and to support countermeasures against replay attacks to the communication between DM functional components.

52) It is recommended to support encryption of DM-related communications.

53) It is required that credentials of devices have appropriate protection mechanisms.

54) In the case of open access to the DM capabilities, it is required that IoT applications and administrators of IoT applications be allowed to manage only authorized devices.

55) It is required to support DM functional components' protection mechanisms against threats such as denial of service attacks.

8 Common capabilities of device management in the IoT

The following clauses describe common capabilities of device management in the IoT.

NOTE – Not all the capabilities listed here are required to be implemented by all IoT systems or applications.

8.1 Configuration management capability

The configuration management capability provides functions to identify, collect and exercise control over configuration data from devices and to provide configuration data to devices [b-ITU-T M.3400].

The configuration management capability supports the following functions:

1) Discovery, provisioning and registration

   (i) Discovery is the process to allow devices, capabilities of the devices and applications running on devices to be found and identified by the DM manager and/or DM GW manager.

   (ii) Provisioning consists of procedures which are necessary to bring a device into service, including the bootstrap procedure, installing parameters and/or applications on a device to establish given services, such as DM services, applications, etc.

   (iii) Registration is the process of recording the information of the device in the DM manager and/or DM GW manager the first time the device accesses the DM manager and/or DM GW manager if the registration is successful, enabling then the DM manager and/or DM GW manager to interact with the device for DM services.

   NOTE – If a registration attempt fails, the DM manager and/or DM GW manager should support the logging of the registration attempt. In addition, the device will not be able to get DM services provided by the DM manager and/or DM GW manager.

2) Firmware/Software image management

   Firmware/Software image management consists of image inventory management, update results reporting, integrity check of image before update process, update process monitoring, fallback mechanism when update fails, etc.

3) Configuration status monitoring

   Configuration status monitoring is the process to get the current status of device configuration parameters and device components. It can take place periodically, on request, or triggered by events.

4) Configuration control

   Configuration control provides the ability to control on demand certain aspects of a device, including setting the device configuration parameters, changing the service state of the device or components of the device, activating and de-activating the device, etc.

8.2 Performance management capability

The performance management capability provides functions to evaluate and report upon the behaviour of a device. Its role is to gather and analyse statistical data for the purpose of monitoring and correcting behaviour and effectiveness of a device and to aid in planning, provisioning, maintenance and measurement of quality [b-ITU-T M.3400].

The performance management capability supports the following functions:

1) Performance monitoring

   Performance monitoring involves the collection of data concerning the performance of devices and the measurement of the overall quality in order to detect service degradation, including performance monitoring policy setting, performance data collection and

processing, performance alarm rule setting, performance alarm collection and processing, performance status reporting, etc.

2) Performance control

Performance control supports the management of schedules, thresholds and other attributes for performance management.

3) Performance analysis

Performance analysis involves additional processing and analysis on the collected data from devices in order to evaluate the performance level of devices, such as performance summary, performance forecasting, performance exception analysis, etc.

8.3 Fault management capability

The fault management capability provides functions enabling the detection, isolation and correction of abnormal operation of devices [b-ITU-T M.3400].

The fault management capability supports the following functions:

1) Alarm surveillance

Alarm surveillance provides the ability to monitor device failures in time, including alarm policy setting, alarm reporting, alarm summary, alarm correlation and filtering, failure event detection and reporting, etc.

2) Fault localization and diagnosis

Where the initial failure information is insufficient for fault localization, it has to be augmented with information obtained by additional failure localization routines. Fault localization, or diagnosis, includes running of diagnostic functions in devices, getting network connection information from the underlying network, summarizing all information from different sources and providing diagnostic reporting.

3) Fault correction

Fault correction deals with repairing a device fault if the device supports fault correction or restoration functions, such as using redundant units, isolating a faulty unit, etc.

4) Trouble administration

Trouble administration deals with investigating and clearing fault reports originated by end users and trouble tickets originated by proactive failure detection, including fault reporting, fault information query, trouble ticket management, etc.

8.4 Security management capability

The security management capability provides the following functions:

1) Security management for communications

The security management capability provides security management mechanisms for communications such as authentication, access control, data confidentiality, data integrity and non-repudiation, which may be exercised in the course of any DM-related communications between devices and the DM manager and/or DM GW manager.

2) Security event detection and reporting

The security management capability provides mechanisms of security event detection and reporting of related results concerning any activity that may be construed as a security violation, such as unauthorized user access, physical tampering with devices, etc.

3) Device security assurance

The security management capability provides mechanisms of device security assurance in order to make sure the device security is not damaged or, at least, to make sure the device has not been intruded by device integrity checking.

4) Device security control

The security management capability provides mechanisms to control device security settings, such as remotely locking or erasing contents of a device if the device has been stolen.

8.5 Connectivity management capability

For device management in the IoT, connectivity is crucial. The connectivity management capability is a necessity for a device's working status monitoring and fault location. The connectivity management capability deals with the communication bearer between device and DM manager (dealing with it as a whole and not dealing with the management of the different network elements and communication links between them, concerned by the communication bearer).

The connectivity management capability supports the following functions:

1) Device connectivity status monitoring

Device connectivity status monitoring provides the ability to get device connectivity status information through direct communication with a device. It is often implemented by the heartbeat mechanism.

2) Device connectivity configuration management

Device connectivity configuration management provides the ability to get and set parameters related to the device connectivity configuration.

3) Network connectivity status monitoring

Network connectivity status monitoring provides the ability to get the status information of the communication bearer between a device and the DM manager. Such information is collected from the network, instead of devices.

4) Network connectivity control

Network connectivity control provides the ability for the DM manager to prohibit the connection of devices to the network for a certain period of time if and as needed.

8.6 DM protocol engine capability

The DM protocol engine capability provides the protocol engine to process DM protocol messages.

The DM protocol engine capability supports the following functions:

1) DM protocol message encapsulation and de-capsulation

2) DM protocol flow control

3) DM protocol adaptation (between two or more DM protocols)

4) DM protocol statistics collection and reporting

8.7 Accounting management capability

In the case of open access to the DM capabilities, the accounting management capability is necessary to enable measurement of DM services' usage and determination of related accounting information.

The accounting management capability includes the following functions:

1) Usage measurement

   Usage measurement provides the ability to collect DM service usage data based on access time of DM services, use of DM services and number of managed devices.

2) Accounting

   Accounting involves the processes responsible for calculating metrics related to DM service usage data.

8.8 Service exposure – web portal capability

In the case of open access to the DM capabilities, the service exposure – web portal capability is a necessity. Via such capability, any DM service provided to administrators of IoT applications, such as configuration management, performance management, fault management, etc., can be accessed through a web portal.

8.9 Service exposure – API capability

In the case of open access to the DM capabilities, the service exposure – API capability is a necessity. Via such capability, any DM service provided to IoT applications, such as configuration management, performance management, fault management, etc., can be accessed through open APIs by IoT applications.

Bibliography

[b-ITU-T M.3400] Recommendation ITU-T M.3400 (2000), *TMN management functions*.

[b-ITU-T Y.4112] Recommendation ITU-T Y.4112/Y.2077 (2016), *Requirements of the plug and play capability of the Internet of things.*

[b-ITU-T Y.4401] Recommendation ITU-T Y.4401/Y.2068 (2015), *Functional framework and capabilities of the Internet of things.*

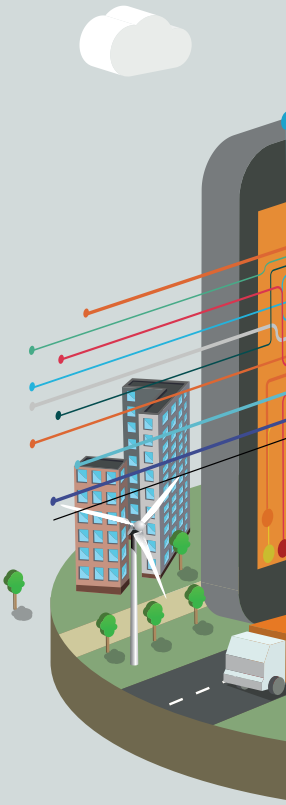
[b-OMA-RD-DM] Recommendation OMA-RD-DM (2013), *Device Management Requirements*.

[b-OneM2M TS-0002] Recommendation OneM2M TS-0002 (2015), *OneM2M Technical Specification Requirements*.

[b-TR-069] Technical Report DSL Forum TR-069 (2004), *CPE WAN Management Protocol*.

8 Identification and Security


SHOP



Y.4800/F.747.5

Requirements and functional architecture of an automatic location identification system for ubiquitous sensor network (USN) applications and services

Requirements and functional architecture of an automatic location identification system for ubiquitous sensor network (USN) applications and services

Summary

Recommendation ITU-T F.747.5 defines the functional requirements, architecture and entities of automatic location identification in ubiquitous sensor networks (USNs).

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T F.747.5	2014-01-13	16	11.1002/1000/12052-en

Keywords

Automatic positioning capability, location-based services, sensor location information.

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

Table of Contents

Introduction

With the help of numerous sensors, radio frequency identification (RFID) tags and other end-node devices, the ubiquitous sensor network (USN) can provide information exchange without any human intervention, for example, it is possible to monitor the weather in one particular area of China from the ITU office in Geneva by means of humidity and temperature sensors, and it is also possible to display traffic flow information for London via the velocity sensors on mobile phones. These trillions of new USN applications and services are built using three basic elements – data, time and location.

With the increase in USN applications and services, new positioning methods are being created and automatic location identification (ALI) capabilities need to be added to the open USN service platform which provide unified accessibility to USN resources and which allows the data of USN applications to take full advantage of the USN capabilities.

Recommendation ITU-T Y.4800/F.747.5

Requirements and functional architecture of an automatic location identification system for ubiquitous sensor network (USN) applications and services

1 Scope

The automatic location identification (ALI) capability enables a device to discover its own location. The ALI system can be deployed along with network equipment or independently integrated with end-node devices. It can be used in various networks such as a mobile network, the Internet, or a low power wireless network.

The scope of this Recommendation includes:

– requirements of the ALI system

– the functional architecture of the ALI system within an open USN service platform

– Specific scenarios of the ALI system.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T F.744] Recommendation ITU-T F.744 (2009), *Service description and requirements for ubiquitous sensor network middleware*.

[ITU-T F.747.4] Recommendation ITU-T F.747.4 (2014), *Requirements and functional architecture for the open ubiquitous sensor network service platform*.

[ITU-T Y.2221] Recommendation ITU-T Y.2221 (2010), *Requirements for support of ubiquitous sensor network (USN) applications and services in the NGN environment*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 open USN service [ITU-T F.747.4]: USN service which provides unified access to USN resources and sensed data/semantic data through heterogeneous USN middleware.

3.1.2 sensor [ITU-T Y.2221]: An electronic device that senses a physical condition or chemical compound and delivers an electronic signal proportional to the observed characteristic.

3.1.3 sensor network [ITU-T Y.2221]: A network comprised of interconnected sensor nodes exchanging sensed data by wired or wireless communication.

3.1.4 sensor node [ITU-T Y.2221]: A device consisting of sensor(s) and optional actuator(s) with capabilities of sensed data processing and networking.

3.1.5 ubiquitous sensor network (USN) [ITU-T Y.2221]: A conceptual network built over existing physical networks which makes use of sensed data and provides knowledge services to anyone, anywhere and at any time, and where the information is generated by using context awareness.

3.1.6 USN middleware [ITU-T Y.2221]: A set of logical functions to support USN applications and services.

3.1.7 USN resource [ITU-T F.747.4]: An entity that provides a USN service including sensor, actuator, sensor node, sensor network and gateway.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 ALI service: A service which provides the best accessible positioning results to USN applications through heterogeneous USN middleware and open USN service platform provisioning based on the use of standard interfaces.

3.2.2 ALI system: An ALI system is a set of interacting or interdependent components forming an integrated whole or a set of elements to offer an ALI service.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

ALI Automatic Location Identification

API Application Programming Interface

CDMA Code Division Multiple Access

FE Functional Entity

GPS Global Positioning System

GSM Global System for Mobile

RFID Radio Frequency Identification

USN Ubiquitous Sensor Network

UWB Ultra-Wideband

5 Conventions

No specific conventions have been used in this Recommendation.

6 Requirements for ALI systems

6.1 General requirements

The following are the general requirements of the ALI system which address the minimum requirements of the location identification system for USN applications and services:

– ALI systems should coexist with existing location-related standards defined by other SDOs such as 3GPP2 [b-3GPP2 IP-BLS] and [b-3GPP2 b-MAPLS], 3GPP [b-3GPP LCS], and OMA [b-OMA SUPL and b-OMA MLS]. It is recommended that the introduction of an ALI system does not have any negative impact on the operation and performance of existing standards.

– The functions introduced by the ALI system are recommended to be either hosted in existing function elements of the USN or in completely new physical entities. The ALI system is required to not impose any modifications on the architecture or functionality of underlying network technology.
– It is recommended that all ALI location data be time-stamped, and the USN application and service are required to use the most recent data available.
– The ALI system is required to provide mechanisms to prevent denial of service attacks.
– The ALI system is required to ensure that data is protected in all transactions, in accordance with the user's privacy preferences, except for when this information is required for emergency or lawful purposes depending on local/regional regulations.
– The architecture is recommended to support the storage of location information for a sensor node so that it may be available later, if required.
– ALI capability is required in order to allow support for location requests, regardless of a user's privacy preferences, when associated with emergency services and applicable by local regulations. The ALI system is required to support sensor node-initiated and network-initiated positioning for emergency location requests. It is required that emergency services location information requests are given a higher priority over other location information requests, based on local regulatory requirements.

6.2 High-level requirements

The following are the high-level requirements of the ALI system, which specify the user's needs for automatic location identification for the ubiquitous sensor network.

– Supporting seamless positioning techniques:

  The ALI system is recommended to support seamless positioning techniques. This could be accomplished by automatically changing to other suitable techniques when the currently operating positioning technique cannot determine the location, or the positioning result is not acceptable.

– Supporting hybrid positioning techniques:

  The ALI system is recommended to support location identification by combining more than one positioning technique when a single positioning technique cannot specify the exact location of the device. It is also recommended to support the process of choosing the best-fit positioning technique from the many available techniques.

– Supporting USN location identification techniques:

  The device utilizes USN techniques to determine its location, such as an image processing technique, RFID or a gyroscope technique. The ALI system is recommended to support these techniques.

– Positioning data compression and conflict control management:

  When portable devices periodically update their location, network performance could be affected if a large number of USN devices are working in an inadequate network environment. It is recommended that ALI systems support conflict control and compression of the positioning data to reduce this problem.

– Conversion of coordinates:

  Location information is shared to support many USN applications which use different coordinates; as a result, the ALI system is recommended to support the uniformity of coordinates and the possible conversion of coordinates.

7 Functional architecture of the ALI system for USN applications and services

7.1 Functional architecture

The functional architecture of the ALI system for USN applications and services may be varied according to diverse USN architectures. Figure 1 shows the functional architecture of the ALI system within the open USN service platform, as a typical example. The functional entities defined in clause 7.2 and the information flow defined in Appendix II, have been applied to this architecture.

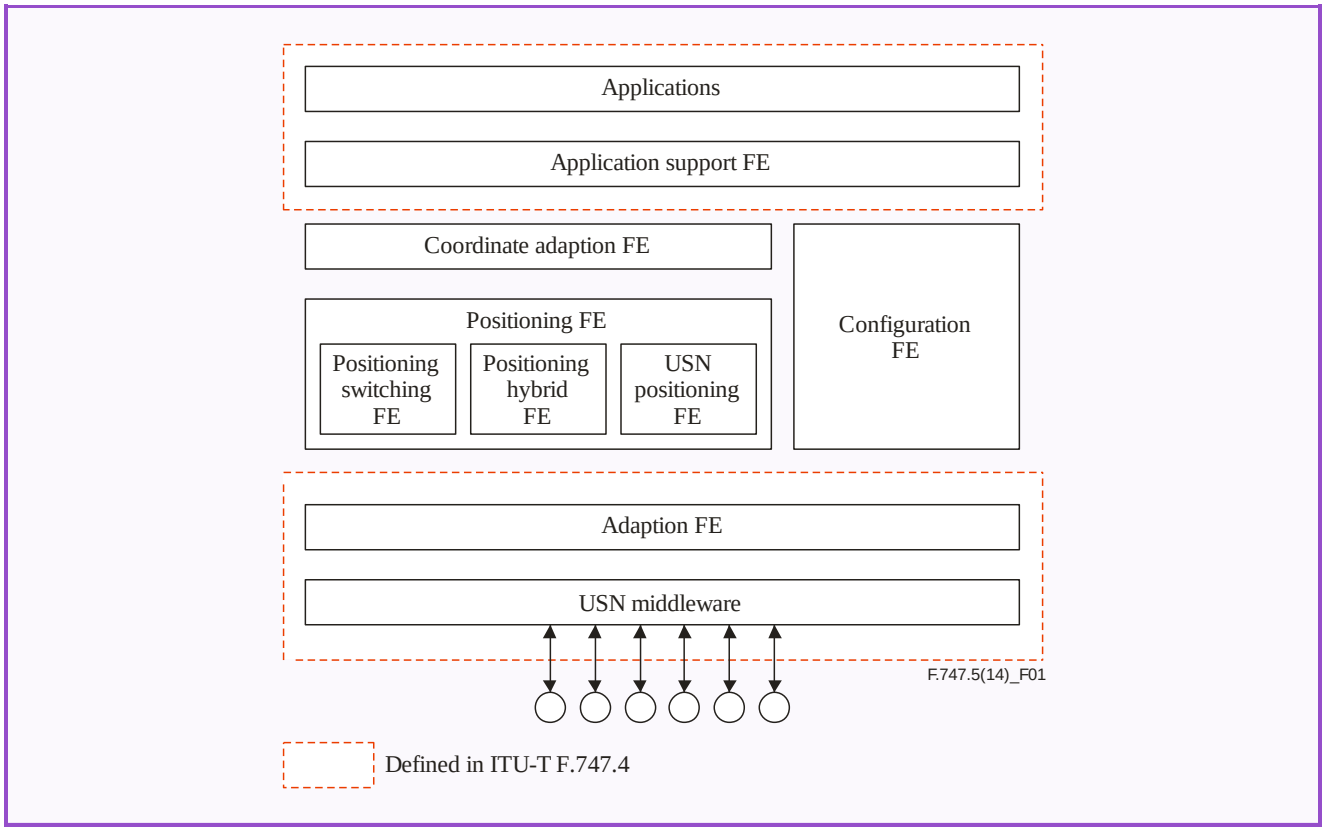


Figure 1 – Functional architecture of the ALI system

7.2 Functional entities

7.2.1 Application support FE

The application support FE provides the functions which enable open USN services to obtain ALI services and/or the positioning data from the ALI system.

Also, it supports the functions which allow the establishment or maintenance of the connection or disconnection according to the type of a request, and the access control to handle the access privileges for users and services.

7.2.2 Coordinate adaptation FE

The coordinates adaptation FE provides the functions which translate the positioning results into the standard and required coordinate, and this is provided for related open USN services.

7.2.3 Remote configuration FE

The remote configuration FE provides the function which enables applications to modify the configurations of the positioning services through an ALI system.

7.2.4 Adaptation FE

The adaption FE provides the functions which handle the protocol and message for setting the connection with USN middleware, and which deliver queries and commands as an interface for processing several types of positioning data that come from USN middleware.

Also, it supports the function to translate generated data from USN middleware to proper message specifications that are dealt with in the ALI system.

The adaptation FE provides a similar function to that of the adaptation FE which is defined in [ITU-T F.747.4]. However, the adaptation FE defined here also directly provides services to sensor nodes which require positioning results and it does not share this with other open application programming interfaces (APIs).

7.2.5 Positioning FE

7.2.5.1 Positioning switching FE

The positioning switching FE provides the functions which support the smooth transition to a valid positioning technique, and which estimate the best available positioning result when the operating positioning technique cannot determine the location or the positioning result is not acceptable.

7.2.5.2 Positioning hybrid FE

The positioning hybrid FE provides the functions which support position calculation by combining various positioning techniques when a single technique cannot specify the exact location of the device. It also supports positioning result optimization by choosing the best-fit positioning technique from the many available techniques, according to the requirements of the applications, such as accuracy first or time first.

7.2.5.3 USN positioning FE

The USN positioning FE provides the functions which support USN techniques for the determination of a sensor node location.

Appendix I

Relationship between open USN service platform and ALI capabilities

(This appendix does not form an integral part of this Recommendation.)

Figure I.1 shows the relationship between an open USN service platform and ALI capabilities. ALI capabilities are important capabilities of an open USN service platform, which are to be provided to various USN applications and services.

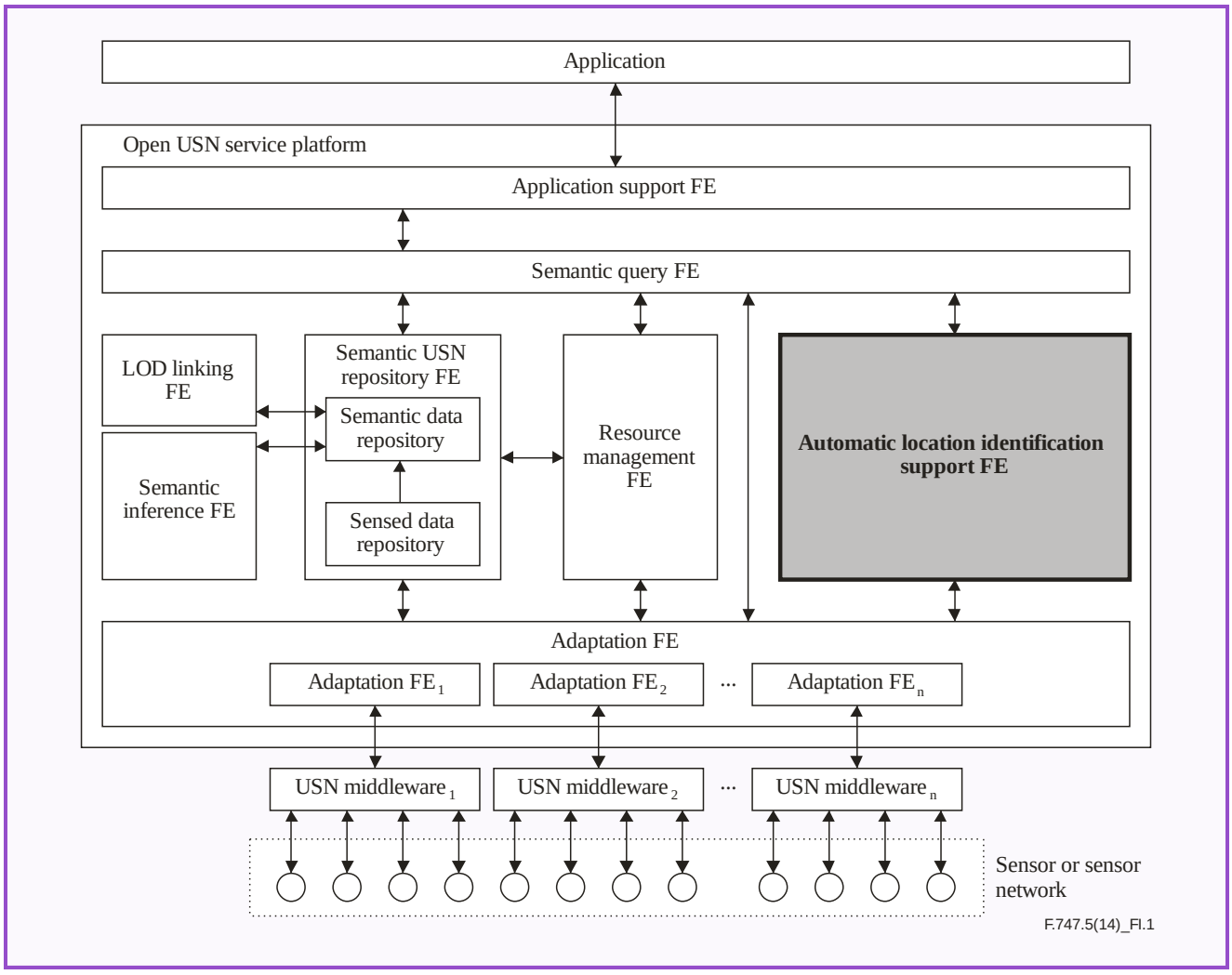


Figure I.1 – Relationship between open USN service platform and ALI capabilities

Appendix II

Information flow of ALI services

(This appendix does not form an integral part of this Recommendation.)

This appendix shows the information flow of ALI systems.

II.1 Resource registration

Figure II.1 shows the information flow of sensor node registration for the use of ALI services.

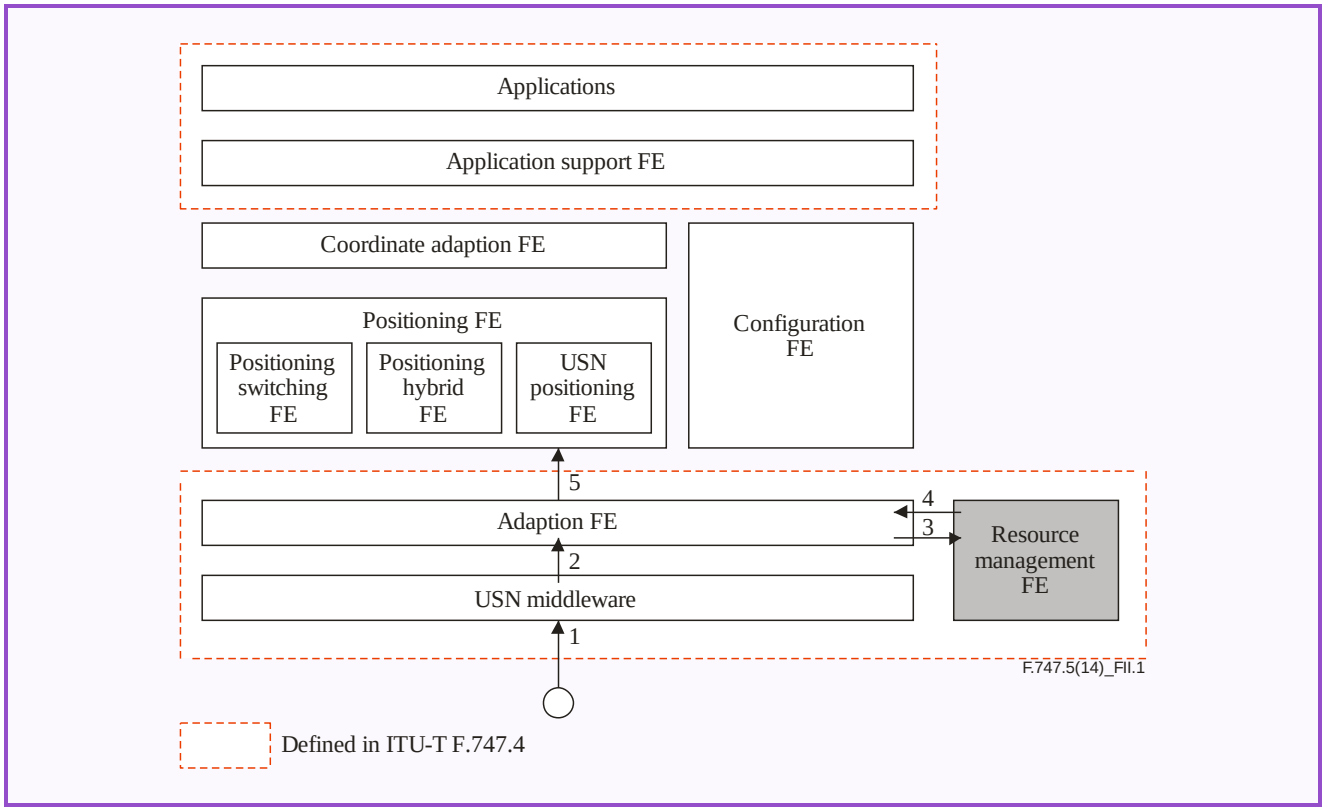


Figure II.1 – Information flow of sensor node registration for the use of ALI services

(1) After receiving an application request, the sensors send the registration requests to the ALI system with the location-related data via USN middleware.

(2) USN middleware sends a request to an open USN service platform through the adaptation FE to ensure that the sensor/application is registered.

(3) The adaptation FE sends the message to the resource management FE.

(4) The resource management FE returns the result of the registration to the adaptation FE and ensures that the sensor supports the ALI function.

(5) The adaptation FE returns the result to the ALI service; the valid requests are sent to the ALI service platform.

II.2 Positioning process

Figure II.2 shows the information flow of the positioning process of an ALI system.

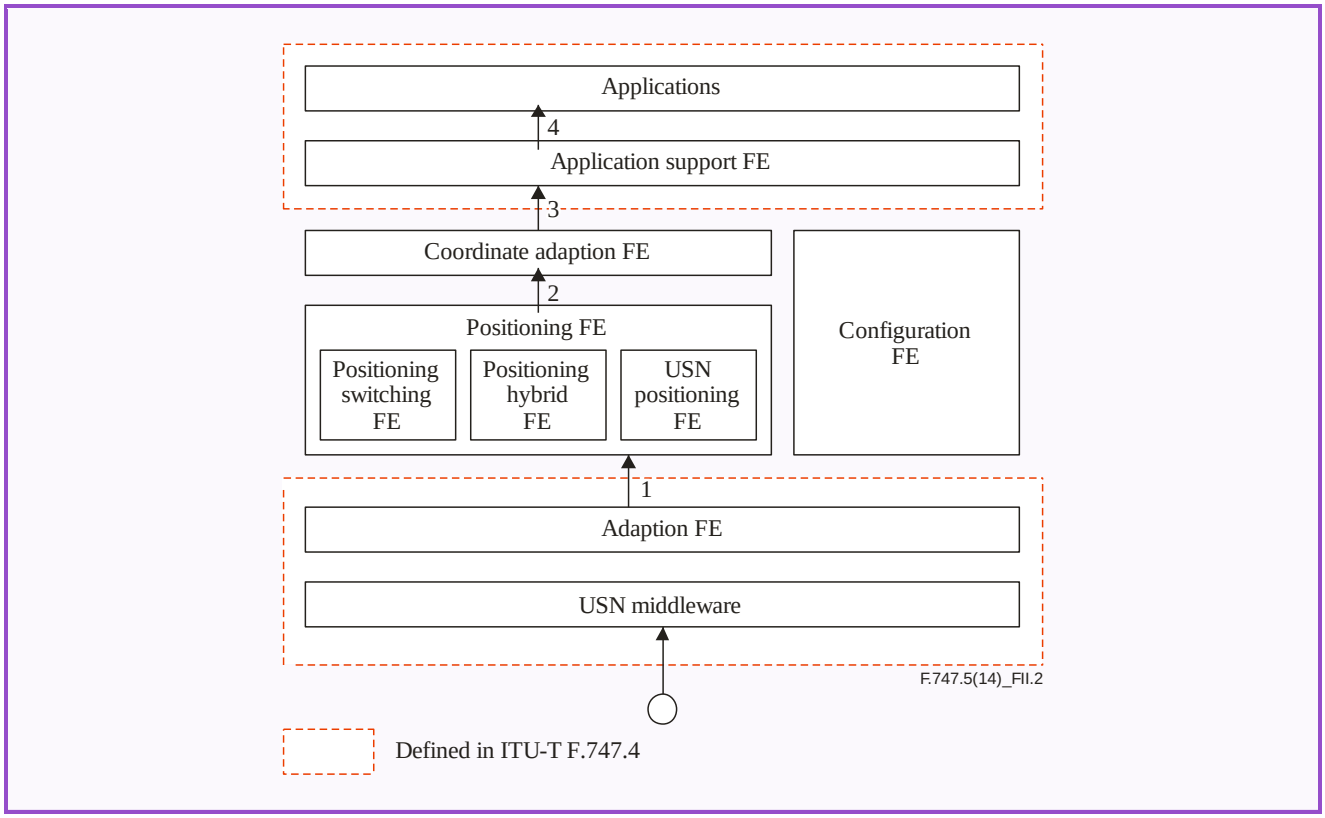


Figure II.2 – Information flow of the positioning process of an ALI system

(1) The valid requests are sent to the ALI service platform.

(2) The positioning FE will then decide, according to the positioning request and any requirements, on the best positioning method, such as positioning switching, positioning hybrid and USN positioning. Positioning switching, positioning hybrid and the USN positioning functions in the positioning FE calculate the position of the sensor.

(3) According to the positioning request, the positioning results are translated into the requested data via the coordinates adaption FE. The coordinates adaption FE sends the data to the application support FE.

(4) The application support FE sends the data to the application.

II.3 Configuration process

Figure II.3 shows the information flow of the configuration process of the ALI system.

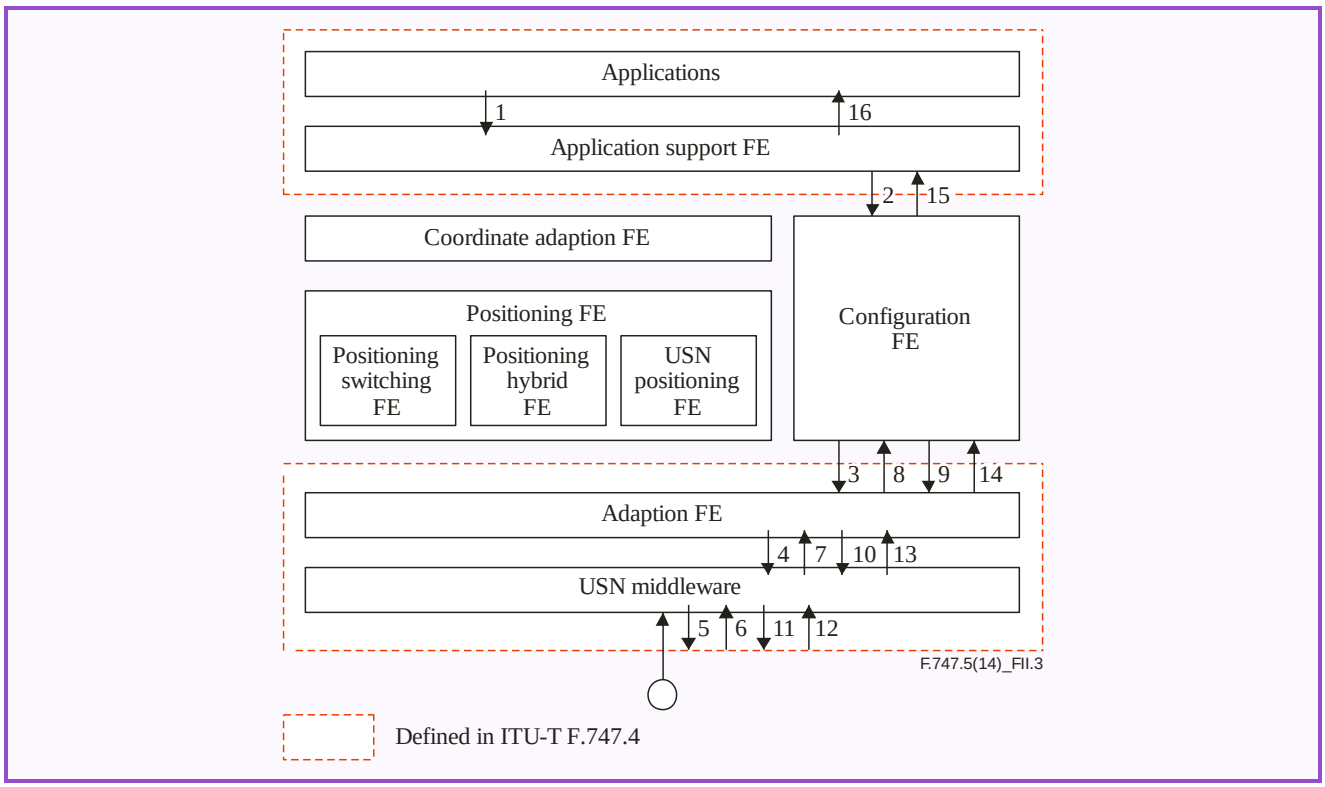


Figure II.3 – The information flow of the configuration process of the ALI system

(1) The application sends a configuration change request to the application support FE.

(2) The application support FE sends the request to the configuration FE. The configuration FE checks the request, sees if it is a valid process and accepts the rules according to the privacy policy and other rules which exist in the configuration FE.

(3) If it is acceptable, the configuration FE sends the configuration change request message to the adaptation FE. Otherwise, it will return the unsuccessful information back to the application.

(4) The adaption FE sends the information to the USN middleware.

(5) The USN middleware sends the request to the sensor.

(6) The sensor receives the configuration change message, and decides whether or not to download the new configuration. When the sensor accepts the request, it sends a request for downloading the new configuration to the USN middleware; otherwise it returns the error code and goes to step 12.

(7) The USN middleware sends the request to the adaptation FE.

(8) The adaptation FE sends the request to the configuration FE.

(9) The configuration FE sends the configuration parameters to the sensor via the adaptation FE.

(10) The adaptation FE sends the new parameters to the USN middleware.

(11) The USN middleware sends the new parameters to the sensor.

(12-14) After successfully receiving the new configuration, the sensor sends the success feedback message to the configuration FE via the USN middleware, and the adaptation FE; otherwise, the sensor sends the error code to the configuration FE via the USN middleware, and the adaptation FE.

(15-16) The configuration FE sends the information back to the application via the application support FE.

Appendix III

ALI scenarios for USN applications and services

(This appendix does not form an integral part of this Recommendation.)

Possible use cases in the ALI system are described in this appendix.

III.1 Location identification for sensor nodes working in complex environments

Compared with regular mobile positioning capabilities, sensor nodes sometimes work in complex environments. One example is that the capability of one single technique, such as a GSM/CDMA cell tower, Wi-Fi real-time locating systems or a global positioning system (GPS), is restricted and the location of a mobile unit cannot be determined, or sometimes it receives multiple location results. Therefore, in complex environments, the sensing node either supports the capability to combine multiple positioning techniques to provide a suitable positioning result or it supports the capability of choosing the best positioning method from more than one approach. The following use cases demonstrate this scenario.

III.1.1 Not receiving an acceptable result from a single positioning technique

This scenario covers an example that requires hybrid positioning techniques. The positioning for indoor sensors during a fire emergency is a typical case which requires the use of various positioning techniques.

The fire zone includes indoor and outdoor sites and many sensors are shot into the burning area. These sensors sense the temperature or capture the motion around them to detect survivors. All the sensors are communicating with wireless / near-field communication techniques, such as Wi-Fi, ZigBee, UWB, etc. With the help of one or more sink nodes, the sensor network connects to transmission networks to report the specific situation of the alerting zone. Some of the sensors contain GPS modules. These sensors are regarded as the reference points, or marks. The position of other sensors could not be determined without the help of these GPS references. By calculating the distance using the power attenuation from the reference points to the sensors, the positions of these sensors could be determined by hybrid positioning techniques, i.e., GPS and received signal strength indication. Figure III.1 illustrates this use case.

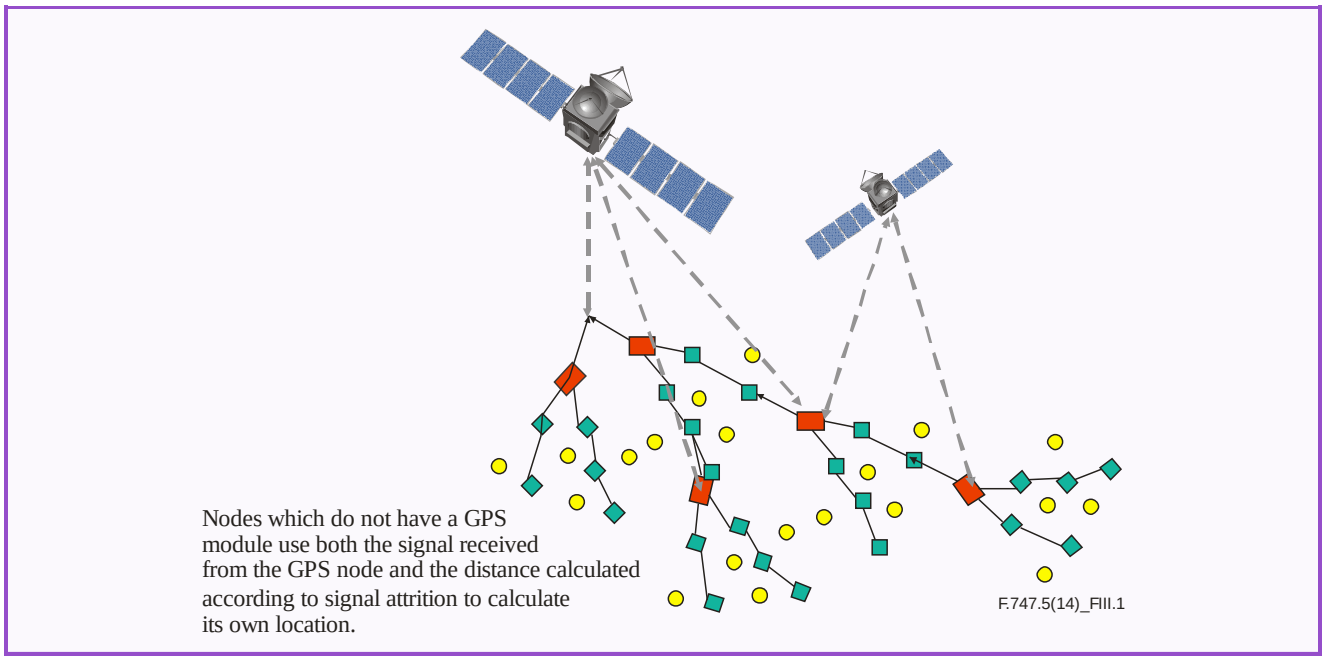


Figure III.1 – Fire emergency use case

III.1.2 Receiving more than one location identification result

With the development of positioning techniques, devices sometimes support more than one methodology of obtaining location information. Several positioning results may be attained; these are estimated via multiple procedures. It is necessary to choose the most appropriate procedure according to the requirements of the application (time-first or accurate-first) to present the most applicable outcome.

A medical emergency can be used to illustrate this. The service is highly sensitive to the amount of time used. When an emergency service requests location information, several modules (IP, cell-tower, GPS, and others) start the positioning processes. The system first informs the nearby ambulances according to the IP address, and requests that they head to the location area consistent with the result from the cell-tower. Yet, the result is not precise enough. While on the road, GPS sends the accurate position of the patient to the ambulance. The optimization of the multiple positioning methods helps the ambulance reach the site of the incident as soon as possible.

III.2 Continuing location identification for sensor nodes in a changing physical environment

Some USN applications ask for an integrated positioning solution that can provide seamless location identification in a variety of environments. This entails a positioning solution which is required to switch automatically from one positioning method to another, as soon as the node realizes the current positioning result is not acceptable to provide the required services with guaranteed quality, or the current positioning procedure cannot be supported. This scenario provides an example where nodes require seamless location identification. The following use cases describe this scenario.

III.2.1 Continuing location identification within the same system of coordinates

Traditionally, positioning results are obtained through geographical coordinates, i.e., latitude and longitude. When switching between traditional techniques, positioning results are expected to be in a uniform format and updated to applications seamlessly.

The tracking of a heart disease patient is a typical use case for the switching between traditional techniques use case. The sensors attached to a patient's body, monitors the patient's heart rate and other vital signs. They also report latitude and longitude to a health manager centre which is tracking the patient's location. When the patient walks along the street, the patient's location is obtained by the GPS module; if the patient drives into a tunnel and the GPS signal is lost, the location can be attained through GSM/CDMA cell-towers; as soon as the patient walks into a house, the device switches automatically to the Wi-Fi positioning mode to address the patient's precise location; and the hospital also provides RFID indicators to specify a more accurate position. Figure III.2 illustrates the heart disease patient use case for a multiple-location identification scenario.

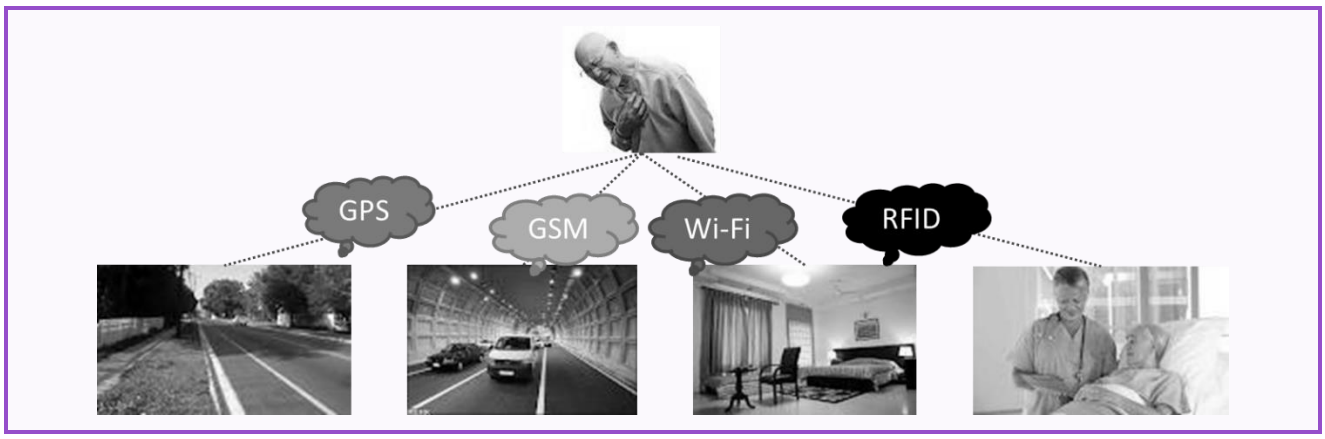


Figure III.2 – Use case of heart disease patient tracking

III.2.2 Continuing location identification across different systems of coordinates

Instead of traditional geographical coordinates, the location information obtained by most new techniques is presented in other formats. When switching between a traditional positioning technique and these new positioning techniques, applications should either get the result in the same format or be informed of the change in format of the coordinates.

Shopping centre navigation is an illustration of this use case. Applications direct customers to target places. Outdoor navigation uses GPS (resulting in geographical coordinates) to help customers drive to the shopping centre; after entering the shopping centre, the indoor guiding system directs customers to their preferred shops, using Wi-Fi (resulting in indoor coordinates) or RFID (represented by a special identification string ID). The indoor coordinate system uses offset x, y, z to indicate the customer's location, while identification string ID uses strings to represent a particular spot. The change in the format of the coordinates should be notified, so that applications understand the meaning of the positioning results and give users the correct directions.

III.3 Location identification for sensor nodes using sensing and actuating techniques

Various sensing and actuating techniques are introduced in the USN. Other than the traditional cell-tower or GPS, these USN sensing and actuating techniques could also be used to determine the location. It opens the door to numerous innovative positioning ways, which also brings challenges for ALI systems to be compatible with these techniques. There are several ways of delivering these state-of-the-art techniques for location identification; these include but are not limited to the following use cases.

III.3.1 Location identification using images

With the new image processing ability, the devices with cameras or other sensors can recognize landmarks or particular patterns, thus determining the location.

The example below of robot positioning shows the image processing technique used in location identification. The robots, which are equipped with cameras/mark-readers, follow particular patterns on the floor. Each mark indicates a certain spot. The robot reads the mark and identifies the position. A supervisor thus tracks the footpaths that the robots have reported, and also gives them appropriate commands. Figure III.3 illustrates the example that the robots identify the locations with the help of an image processing technique.

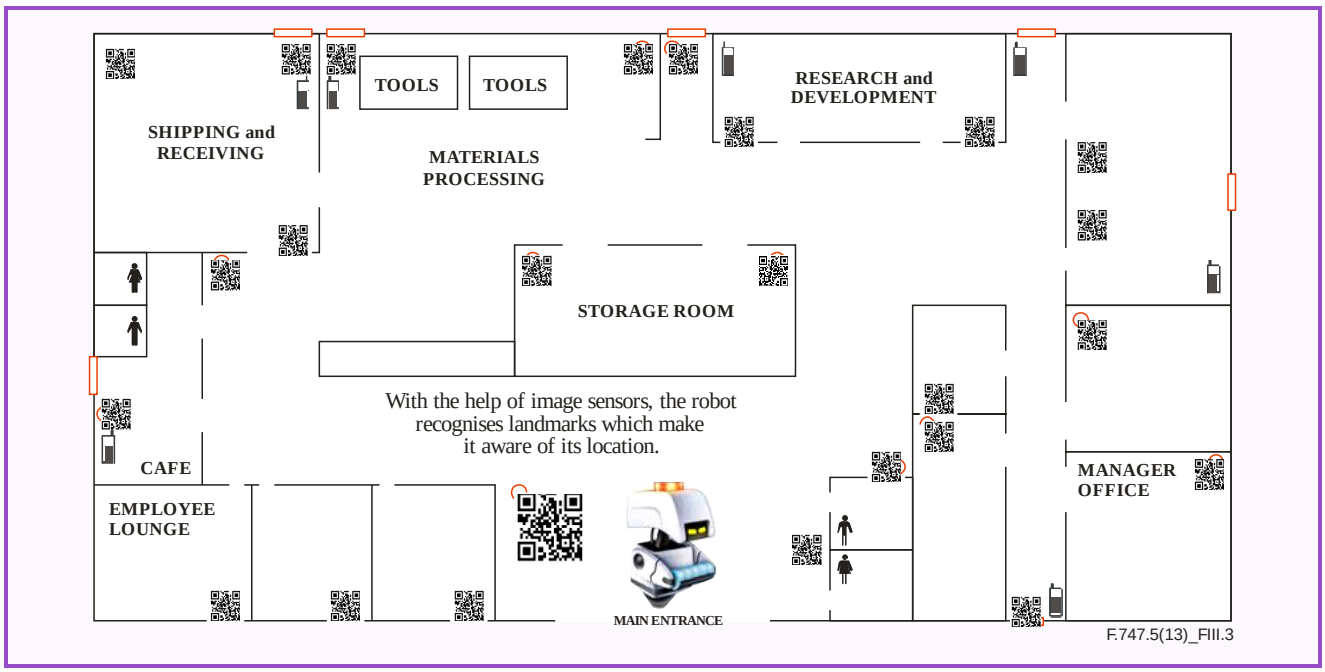


Figure III.3 – Use case of robot positioning with image processing capabilities

III.3.2 Location identification using an RFID tag

When a radio signal from a specific RFID tag is collected by devices at a certain point, the identification string of the tag can be decoded, and the corresponding location can be determined.

III.3.3 Location identification using a gyroscope

A device with a gyroscope can quantify movement in the x, y, z directions, and thus it can calculate its position from the original point.

III.4 Configuration for sensor nodes with limited processing capabilities

Sometimes, the configuration parameters of the sensor need to be modified during the positioning process according to its environment or the requirements from applications. However, the processing capabilities of the USN end nodes are limited. The ALI end node cannot adjust itself spontaneously and it needs the help of the platform. The configuration parameters may include but are not limited to the time of positioning periods, the alarming condition, etc. When sensors are not able to complete these actions dynamically, they connect to the management platform and download the configurations and process as required.

The sensor placed in a briefcase can be a typical example. It checks its position periodically. The positioning period and the sound of the alarm can be altered remotely by the management platform, according to the distance between the briefcase and the owner. If the distance between the briefcase and its owner exceeds the distance limit, the platform can send a notification to the owner.

III.5 Location identification for sensor nodes in a resource-limited USN environment

Resource-limited places are common in daily life and USN systems may be deployed in such places. Moreover, a USN may utilize resource-limited devices or terminals to facilitate its deployment. The limitation may be the power of the battery or the bandwidth of the link. This scenario demonstrates that location identification works in resource-limited environments, and it describes use cases which optimize the ALI system in such situations.

III.5.1 Power-limited environment

A power-limited environment refers to cases where, when the ALI system is used, the power of the device can be exhausted after a period of time. A typical power-limited environment is when devices adopt batteries as their power supply. Decreasing the power consumption can prolong the usage time. For instance, the ZigBee [b-ZigBee] terminal providing location identification can switch to power saving mode to extend its usage time (when batteries are used).

III.5.2 Network-limited environment

The network-limited environment includes two cases, low bandwidth and unstable connections. For example, ZigBee or some UWB technologies have very low bandwidths. The connection of GSM/CDMA is sometimes not stable, due to strong interference. In this case, the ALI system needs to increase its transmission efficiency, for instance, by compressing the payload of location information in packets.

Bibliography

[b-3GPP LCS] 3GPP (2013), *Functional stage 2 description of Location Services (LCS)* <http://www.3gpp.org/ftp/Specs/html-info/23271.htm> (accessed 2013-05-20)

[b-3GPP2 IP-BLS] 3GPP2 (2005), *IP-Based Location Services Revision: 0 – 3GPP2.* <http://www.3gpp2.org/public_html/specs/X.S0024-0_v1.0_051102.pdf>(accessed 2013-05-20)

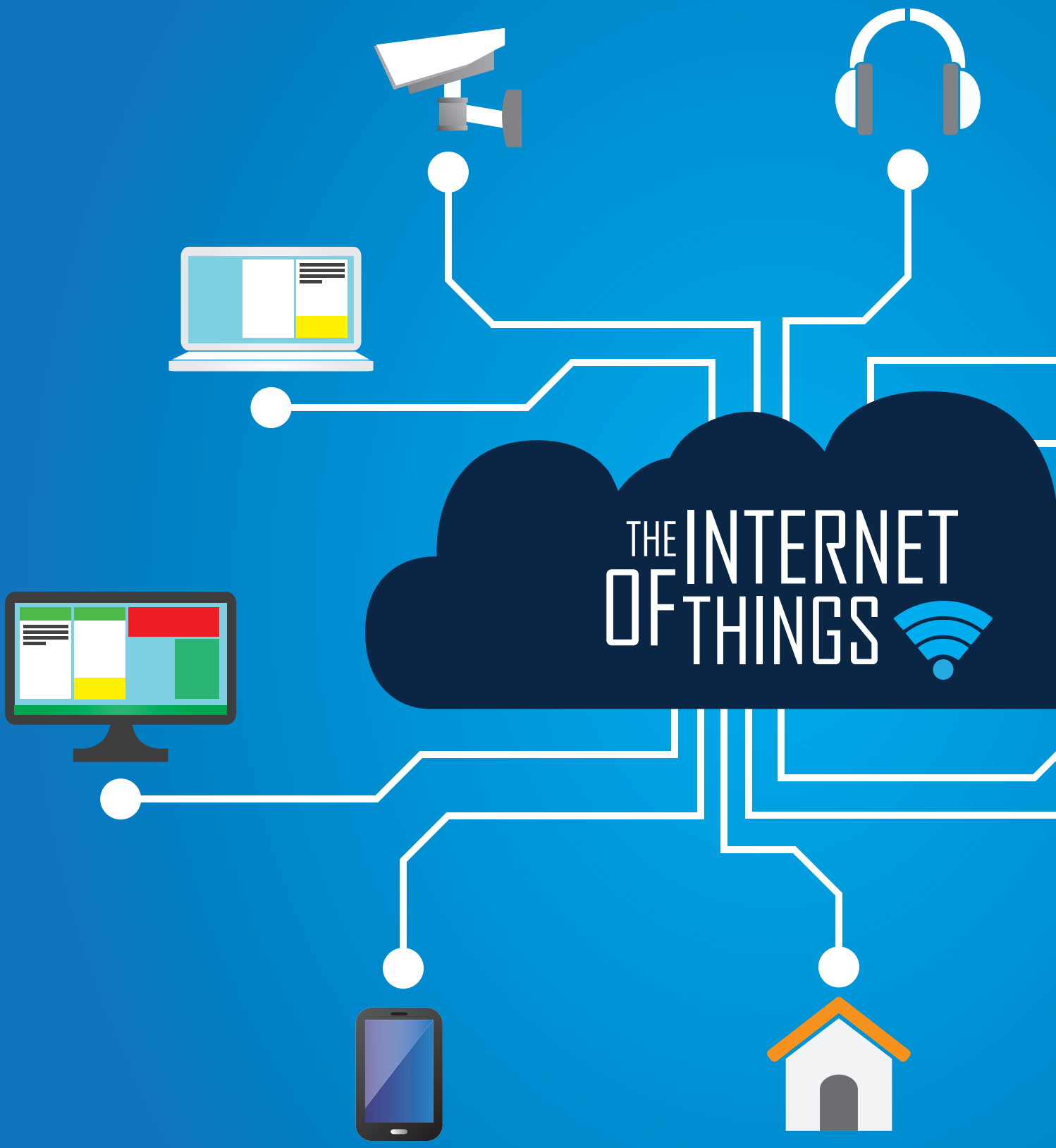
[b-3GPP2 MAP-LS] 3GPP2 (2006), *MAP Location Services Enhancements.* <http://www.3gpp2.org/public_html/specs/X.S0002-0_v2.0_060531.pdf> (accessed 2013-05-20)

[b-OMA] OMA (2011), *Mobile Location Service V1.2*. <http://technical.openmobilealliance.org/Technical/release_program/mls_v1_2.aspx> (accessed 2013-05-20)

[b-OMA SUPL] OMA (2013), *OMA Secure User Plane Location V2.0.* <http://technical.openmobilealliance.org/technical/release_program/supl_v2_0.aspx> (accessed 2013-05-20)

[b-ZigBee] ZigBee Alliance (2007), *ZigBee Specification*. <http://www.zigbee.org/Specifications/ZigBee/download.aspx> (accessed 2013-12-12)


BUS STOP


THE INTERNET OF THINGS

Y.4801/F.748.1

Requirements and common characteristics of the IoT identifier for the IoT service

Requirements and common characteristics of the IoT identifier for the IoT service

Summary

Recommendation ITU-T F.748.1 describes the requirements and common characteristics of the Internet of things (IoT) identifier for the IoT service.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T F.748.1	2014-10-14	16	11.1002/1000/12229

Keywords

Identifier, IoT.

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

Table of Contents

Introduction

ITU perceives the Internet of things (IoT) as a vision with technological and societal implications that can be viewed as a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies. As a common understanding, the IoT involves various kinds of technologies such as: identification, architectural work, communications, networking, discovery and search engines, power and energy storage and security and privacy. Among these technologies, it is widely agreed that radio frequency identification (RFID), ubiquitous sensor network (USN) or sensor networks and machine-to-machine (M2M) or machine oriented communication (MOC) will be enablers for the IoT.

In RFID environments, tag-based identification [b-ITU-T F.771] enables users to access multimedia information through users' electronic devices equipped with ID tag readers and communication functions. In the IoT environments, it is required to study not only tag-based identification, but also general-purpose identification schemes. Because the IoT involves various enablers such as RFID, USN or sensor networks, and M2M or MOC, the identification scheme of the IoT is required to be applied to these various enablers.

To realize the IoT services, existing information and communication technologies should evolve to support the characteristics of the IoT. Third generation partnership project (3GPP) systems, Internet, wireless local area network (WLAN), wireless personal area network (WPAN), and next generation networks (NGNs) are evolving to provide IoT services based on existing technologies. Due to economic and technical challenges, it is rare to implement completely new technologies. As existing technologies are evolving respectively and independently, the existing identification scheme for each technology will be used continually.

There may be two options for the creation of a future IoT identifier. One is the evolution from existing identifiers, and the other is the creation of a new identifier. Given that information and communication technologies are typically developed with the guarantee of interoperability using existing technologies, a future IoT identifier looks likely to evolve from existing identifiers.

Recommendation ITU-T Y.4801/F.748.1

Requirements and common characteristics of the IoT identifier for the IoT services

1 Scope

The objective of this Recommendation is to analyse identifiers in existing technologies for the Internet of things (IoT) services, and describe the requirements of the IoT identifier, common characteristics of the IoT identifier, and the reference model of the IoT identifier.

This Recommendation describes the requirements and common characteristics of the IoT identifier for the IoT services. The scope of this Recommendation includes:

– analysis of identifiers in existing technologies,

– common characteristics of the IoT identifier,

– requirements of the IoT identifier,

– reference model of the IoT identifier.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T Y.2060] Recommendation ITU-T Y.2060 (2012), *Overview of the Internet of things*.

[ITU-T Y.2063] Recommendation ITU-T Y.2063 (2012), *Framework of the web of things*.

[ITU-T Y.2091] Recommendation ITU-T Y.2091 (2011), *Terms and definitions for next generation networks*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 address [ITU-T Y.2091]: An address is the identifier for a specific termination point and is used for routing to this termination point.

3.1.2 identifier [ITU-T Y.2091]: An identifier is a series of digits, characters and symbols or any other form of data used to identify subscriber(s), user(s), network element(s), function(s), network entity(ies) providing services/applications, or other entities (e.g., physical or logical objects). Identifiers can be used for registration or authorization. They can be either public to all networks, shared between a limited number of networks or private to a specific network (private IDs are normally not disclosed to third parties).

3.1.3 Internet of things [ITU-T Y.2060]: A global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on, existing and evolving, interoperable information and communication technologies.

NOTE 1 – Through the exploitation of identification, data capture, processing and communication capabilities, the IoT makes full use of things to offer services to all kinds of applications, whilst ensuring that security and privacy requirements are fulfilled.

NOTE 2 – In a broad perspective, the IoT can be perceived as a vision with technological and societal implications.

3.2 Terms defined in this Recommendation

None.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

3GPP	Third Generation Partnership Project
CID	Caller ID
DNS	Domain Name System
ETSI	European Telecommunications Standards Institute
FQDN	Fully Qualified Domain Name
HTTP	Hypertext Transfer Protocol
ICTs	Information and Communication Technologies
IEEE	Institute of Electrical and Electronics Engineers
IMEI	International Mobile Equipment Identity
IMSI	International Mobile Subscriber Identity
IoT	Internet of Things
Ipv4	Internet Protocol version 4
Ipv6	Internet Protocol version 6
ITS	Intelligent Transport Systems
M2M	Machine-to-Machine
MAC	Medium Access Control
MOC	Machine Oriented Communication
MSISDN	Mobile Subscriber ISDN Number
MTC	Machine Type Communication
NGN	Next Generation Network
PDA	Personal Digital Assistant
RFID	Radio Frequency Identification
TCP/IP	Transmission Control Protocol/Internet Protocol
URI	Uniform Resource Identifier
USN	Ubiquitous Sensor Network
WLAN	Wireless Local Area Network
WPAN	Wireless Personal Area Network

5 Conventions

None.

6 Introduction of the IoT identifier

Information and communication technologies (ICTs) have been used for and by humans, but this situation is evolving. ICTs are being used not only by humans, but also by machines (e.g., things, devices, objects). Specifically, technological environments composed of fixed computers, laptops and high performance machines are evolving to be composed of smartphones, personal digital assistants (PDAs), portable multimedia devices, lightweight devices and sensors. Moreover, all things will be connected to networks as time goes by. This change will be one of the features of IoT.

ICT services are evolving to support the IoT services; as a result, various services such as intelligent transport systems (ITS), smart home services, u-Health, and smart-metering services will converge into the IoT services. Communication and networking technologies such as third generation partnership project (3GPP) systems, Internet, wireless local area network (WLAN), wireless personal area network (WPAN) and next generation network (NGN) are also evolving to provide IoT services based on existing technologies. However, to provide the IoT services, it is necessary to implement completely new services and technologies.

As the existing ICTs services are evolving respectively and independently, the identifier for the existing technologies will be used continually. In the same manner, the future IoT identifier may evolve from the existing identifiers. In 3GPP system-based networks, the mobile subscriber ISDN number (MSISDN), international mobile equipment identity (IMEI), and international mobile subscriber identity (IMSI) are used to provide the IoT services. In the Institute of Electrical and Electronics Engineers (IEEE) 802.11 WLAN-based networks, the medium access control (MAC) address is used as an identifier. In IEEE 802.16 WiMAX-based networks, caller IDs (CID) or MAC addresses are used as an identifier. Each identifier can be used in the same networking technologies, but it is impossible to interoperate with other identifiers used in different networking technologies. If it is required that different identifiers are interoperable in the same manner as the conventional Internet, upper layer identifiers such as Internet protocol version 4 (IPv4), Internet protocol version 6 (IPv6) and uniform resource identifiers (URI) can be used.

If the use of IoT services increases exponentially and globally, new access technologies and new communication networks can be deployed for the IoT services. In this case, a new IoT identifier should be designed and implemented.

Since ICTs are developed with the guarantee of interoperability using existing technologies, then the future IoT identifier looks likely to evolve from the existing identifiers. For interconnection among things (e.g., machines, devices, objects) that are connected to different networks where different identifiers are used, the existing approach of the conventional Internet (interconnection at upper layer using IPv4, IPv6, and URI) or a new approach considering the characteristics of the IoT can be used. For example, a new identifier can be considered to associate the IoT services with devices and vice versa.

In the IoT environment, various ICTs as well as various services and applications should provide for interworking. The existing ICT services and applications have been developed for their own purpose and there is no requirement for them to interconnect with other services and applications. As many things are interconnected and connected to the networks, various services and applications have become interworked. To interwork different services and applications, web-based services and applications have been acknowledged to support this requirement [ITU-T Y.2063]. Whatever the future features of services and applications are in the IoT, the IoT identifier in services and applications will be developed based on the existing identifiers with interoperability.

This Recommendation analyses identifiers in the existing technologies for the IoT services, and describes the requirements of the IoT identifier, common characteristics of the IoT identifier, and the reference model of the IoT identifier.

7 Analysis of identifiers in the existing technologies

This clause describes the existing identifiers in 3G networks and the Internet that can be used in a network for the IoT services.

Both MSISDN and IMSI are used as identifiers in 3G networks; on the other hand, IPv4 and IPv6 addresses, URIs, and fully qualified domain names (FQDNs) are used as identifiers in the Internet.

Each identifier can be used in the same network, but it is impossible to be interoperable with other identifiers used in different networks. If it is required that different identifiers are interoperable, in the same manner as those in the conventional Internet, upper layer identifier such as IPv4 and IPv6 can be used. Considering the characteristics of the IoT, it is certain that the approach used in the conventional Internet is not suitable for the IoT.

In the 3GPP network, identifiers are categorized as follows [b-3GPP 23.003]:

– MSISDN with existing length,
– MSISDN with max length of 15 digits,
– IMSI

Although the maximum length of the MSISDN is 15 digits, the currently used length is 12 to 13 digits. The MSISDN with existing length is a globally unique identifier in the 3GPP network and it does not impact the current billing system and authentication scheme in the 3G network. In addition, the MSISDN scheme can represent location information and service information of a device. The MSISDN with existing length can work very well as an identifier scheme in the 3G network. However, the length of the MSISDN is limited; therefore, the MSISDN can lead to a lack of identifiers holding numerous numbers of devices connected to the 3G network [b-3GPP 22.988].

The MSISDN with a maximum length of 15 digits can be used as another identifier in the 3G networks and is a globally unique identifier. In addition, this scheme does not require new standards in 3G networks and can provide a large number of additional MSISDNs. However, if this scheme is used as an identifier of a device in the IoT, it may require changes to existing devices in 3G networks and can have an impact on billing systems.

The IMSI is supported widely in 3G networks and serves as an effective identification scheme to uniquely identify each device. However, it is not suitable for session/call routing identifiers, but rather its primary purpose is for authenticating devices. In addition, the IMSI is not generally accessible when a device is connected through the Internet. Even if the IMSI is used as an identifier of a device in the IoT, the domain name systems (DNS) have to translate the IMSI into an IP address. The IMSI cannot be used as an identifier for billing systems. Thus, modifications to existing billing systems may be required. The IMSI is geographically distributed, but used within a dedicated region. Although the available number of IMSIs is much larger than the number of MSISDNs, the number of devices in the IoT may increase dramatically and consequently lead to insufficient numbers for the IMSI to be used as an identifier.

In the Internet, identifiers are categorized as follows:

– uniform resource identifier,
– fully qualified domain names,
– IPv4 address, IPv6 address.

URIs are widely used as generic identifiers. They can be resolved to an IP address by DNS and enable interaction between resources over the Internet using specific protocols (e.g., hypertext transfer protocol (HTTP)). Thus, if the IoT network infrastructure is realized by existing network technologies, such as conventional transmission control protocol/Internet protocol (TCP/IP) based networks, URIs can be used as a generic identifier of the IoT.

In the conventional Internet environment, the FQDN is a basic domain name scheme and works very well with current IP networks. However, if the FQDN is used as a device identifier in the IoT, the FQDN scheme can be a huge burden on DNS servers. Mapping of every FQDN to an IP address requires an entry in the DNS system. Thus, using the FQDN as an identifier of the IoT devices would increase entries in the DNS system. In addition, this scheme requires additional system updates for authentication and billing in 3G networks. The FQDN scheme has virtually unlimited name space, thus it does not have a limit to the number of identifiers.

IPv4/IPv6 addresses are not suitable as public identifiers because they are used as routing identifiers in the Internet. Although IPv4/IPv6 addresses are generic identifiers, there are not enough of them to accommodate numerous devices in the IoT (i.e., the exhaustion of identifiers can occur).

8 Common characteristics of the IoT identifier

Common characteristics of the existing identifiers can be applied to the IoT identifier. Generally, an identifier can be used to identify subscriber, user, network element, function, network entity, or other entity (e.g., physical or logical objects) [ITU-T Y.2091].

In addition to common characteristics of the existing identifiers, the IoT identifier has the following additional characteristics:

– Identifying things:

  An identifier can identify various kinds of things. The existing identifiers identify *specific* things, but the IoT identifiers identify *various* kinds of things. To identify various kinds of things and interconnect things, it is acknowledged that the Internet and the web-based identifiers are more suitable.

– Communicating with other things:

  After being identified, various types of things can communicate with computers, humans, and other things. With communication between things, things can be interconnected with each other.

– Connecting to networks:

  Although different communication technologies and services/applications are used, things are connected to networks directly/indirectly and always/periodically/on-demand with identifiers.

– Huge accommodation:

  Existing identifiers have a limited range of available allocation. The number of the IoT identifiers should be large enough (may be limitless) to cover an enormous number of things.

– Interconnection:

  Identifiers of things in different networks can be interconnected. Identifiers of physical things in the physical world and identifiers of virtual things in the information world will be interconnected.

– Diversity:

  Identifiers in the physical world and the information world are different. In the physical world, the identifiers of physical things of the IoT devices may be different according to applied technologies.

– Multiplicity:

  Multiple identifiers may be allocated to a physical thing, a device and a virtual thing. For a specific service and a specific operation, a particular identifier will be selected and utilized.

– Permanent or limited lifetime:

  Some identifiers that are assigned to things will be either permanent or temporary (i.e., vanish after their time-to-live).

– Identification of the IoT services:

  Some identifiers can be used to identify the IoT services. When doing this, it is efficient to manage resource and devices in the application layer.

9 Requirements of the IoT identifier

In the IoT environment, a thing is understood as a physical object in the physical world and a virtual object in the information world that is capable of being identified and connected to information and communication networks. The role of an identifier of a thing in the IoT environment is to uniquely identify a physical and a virtual object. An IoT identifier has the requirements listed in the following clauses.

9.1 Identifying anything

REQ-001: It is required that identification schemes be able to identify anything.

According to the fundamental characteristics of interconnectivity in [ITU-T Y.2060], anything (physical or virtual, devices or non-devices (do not need communication capabilities)) will be interconnected with the global information and communication infrastructure.

9.2 Communication between things

REQ-002: It is required to support the connectivity between things and to provide communication based on identifiers of things.

REQ-003: It is recommended to provide a harmonized way to integrate identifiers of devices that need communication capability with identifiers of things that do not need communication capability.

In IoT environments, a thing can be a physical and a virtual object, but some things do not need communication capability.

9.3 Association between physical objects and virtual objects

REQ-004: It is required to provide association between a physical object and a virtual object using an identifier.

An identifier in conventional information and communication networks is mainly used to provide connectivity and communication between physical objects. However, an identifier in an IoT environment can be used to provide connectivity and association between physical objects and/or virtual objects.

9.4 Networking technology independency

REQ-005: As IoT devices may use different network technologies, identifiers for these IoT devices are recommended to be independent of underlying networking technologies.

In IoT environments, numerous devices will be connected to each other, and those devices may belong to different networks that use different networking technologies. Generally, devices in a network using specific networking technologies can be identified by a specific identifier and/or address used in those networking technologies.

9.5 Mapping identifiers to objects

The IoT reference model in [ITU-T Y.2060] is composed of four layers as well as management capabilities and security capabilities associated with the four layers. The four layers are: application layer, service/application support layer, network layer and device layer. From the point of view of an identifier, an identifier in each layer is used to map specific objects (resources) in each layer or one universal identifier can be used to map specific objects (resources) in all layers. Figure 9-1 (a) shows the case where an identifier in each layer is used to map to specific objects in each layer and Figure 9-1 (b) shows the case where one universal identifier is used to map to specific objects in all layers.

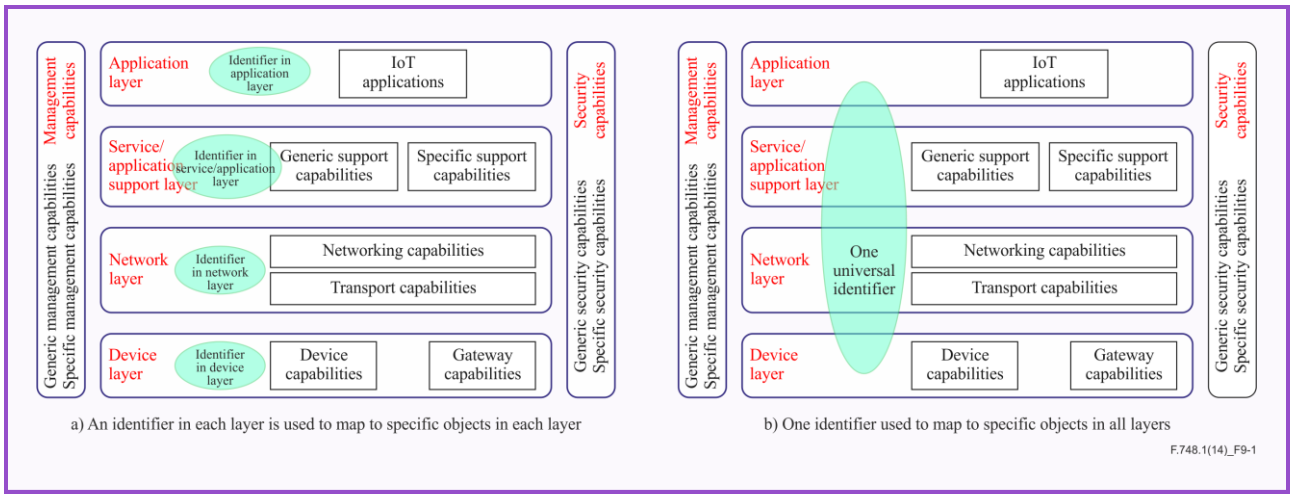


Figure 9-1 – Mapping identifiers to objects

REQ-006: In the case where an identifier in each layer is used to map to specific objects in each layer, it is required that each identifier, in each layer, does not directly impact other identifiers in adjacent layers.

When an identifier in each layer is used to map to specific objects in each layer, it is similar to the usage of identifiers in the conventional Internet. For example, in the application layer URIs and FQDNs are used. In the service/application support layer, port numbers and session numbers are used. In the network layer, IPv4 addresses or IPv6 addresses are used. In the device layer, IEEE 802.11 MAC addresses can be used.

REQ-007: In the case where one universal identifier is used to map to specific objects in all layers, it is recommended that this one universal identifier efficiently maps to specific objects in each layer and is integrated in a harmonized way.

The case where one universal identifier is used to map to specific objects in all layers is somewhat different from the conventional Internet. One universal identifier may cover all four layers, provide interconnection capabilities between physical objects and virtual objects, and provide simplicity. A universal identifier can be a new identifier with or without interoperability with existing identifiers. Alternatively, one identifier in each layer can be chosen and used as the one universal identifier.

9.6 Relation between characteristics and requirements

NOTE – This clause does not define requirements.

In clause 8, common characteristics of IoT identifiers are described. Table 9-1 shows the relation between common characteristics and requirements of the IoT identifiers.

Table 9-1 – Relation between common characteristics and requirements of IoT identifiers

Common characteristics	Requirements
Identifying things	REQ-001
Communicating with other things	REQ-002, REQ-003
Connecting to networks	REQ-002, REQ-003
Huge accommodation	REQ-001
Inter-connection	REQ-002, REQ-003, REQ-004, REQ-005
Diversity	REQ-004, REQ-005
Multiplicity	REQ-006
Permanent or limited life-time	
Identification of the IoT services	

10 New capability for IoT identifiers

As described above, the IoT reference model in [ITU-T Y.2060] is composed of four layers (application, service/application support, network, and device) and two capabilities (management and security). When only considering the IoT identifier in the physical world, the reference model of identification in the IoT is not different from the reference model in [ITU-T Y.2060]. The fact that the IoT identifier should identify devices and physical things in the physical world and virtual things in the information world necessitates a different reference model of identification in the IoT.

In [ITU-T Y.2060] a device, a physical thing and a virtual thing are described as follows:

– A device is a piece of equipment with the mandatory capabilities of communication and the optional capabilities of sensing, actuation, data capture, data storage and data processing.

– A physical thing is an object of the physical world that is capable of being identified and integrated into the communication networks.

– A virtual thing is an object of the information world that is capable of being identified and integrated into the communication network.

A physical thing may be represented in the information world via one or more virtual things (mapping), but a virtual thing can exist without an associated physical thing. A device and a physical thing are well identified in the [ITU-T Y.2060] reference model, but it is difficult to identify a virtual thing in that reference model. Therefore, it is required to associate identifiers in the physical world with identifiers in the information world and vice versa.

– Mapping capability: Mapping capability is a particular capability that can be used by two different worlds. Due to the nature of different characteristics of the physical world and the information world, identifiers in each world may be different. This capability associates identifiers in the physical world with identifiers in the information world and vice versa.

11 Reference model of identification in the IoT

Figure 11-1 shows a reference model of identification in the IoT where an identifier in each layer is used to identify specific objects in each layer (see clause 9.5). As shown in Figure 11-1, identifiers in each layer in the physical world may be associated with identifiers in the information world. For some specific cases, only related identifiers in the physical world may be associated with identifiers in the information world. In this scenario, mapping capability associates (all or some specific) identifiers in the physical world with the identifier in the information world and vice versa.

Figure 11-2 shows a reference model of identification where one identifier is used to identify specific objects in all layers (see in clause 9.5). As shown in Figure 11-2, one identifier in the physical world may be associated with identifiers in the information world. In this scenario, mapping capability associates one identifier in the physical world with identifiers in the information world and vice versa.

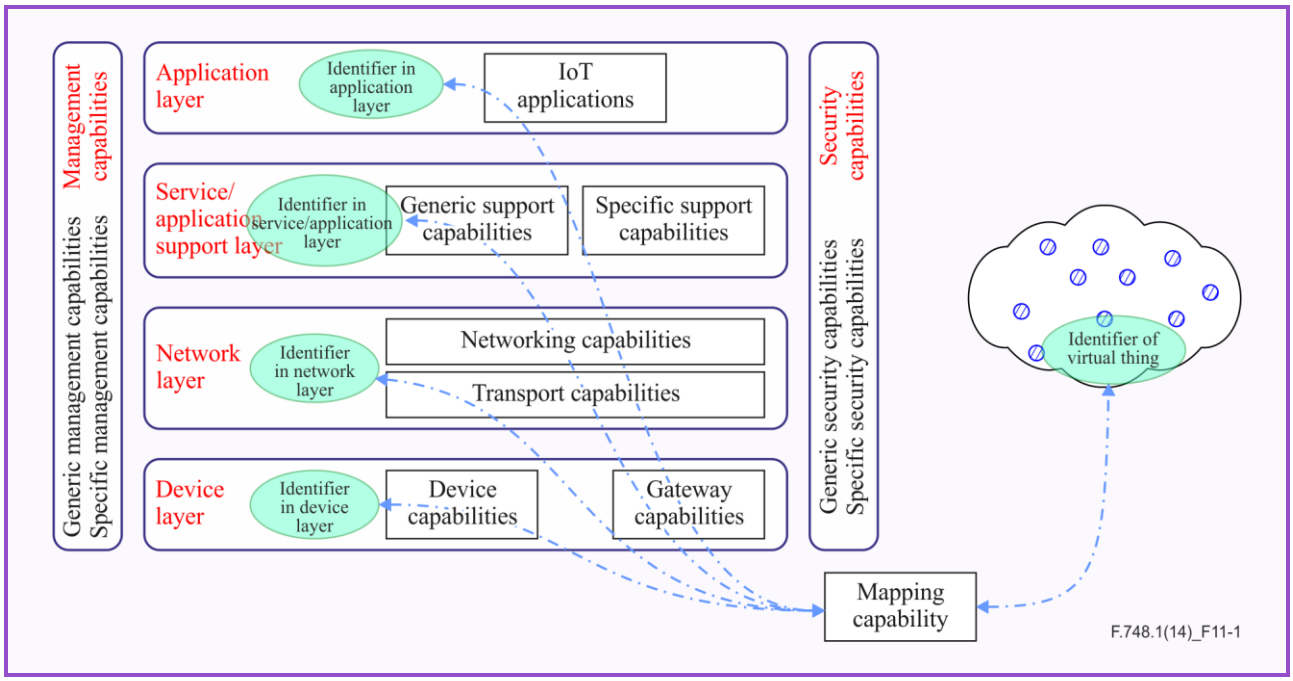


Figure 11-1 – Reference model of identification (case (a) in Figure 9-1)

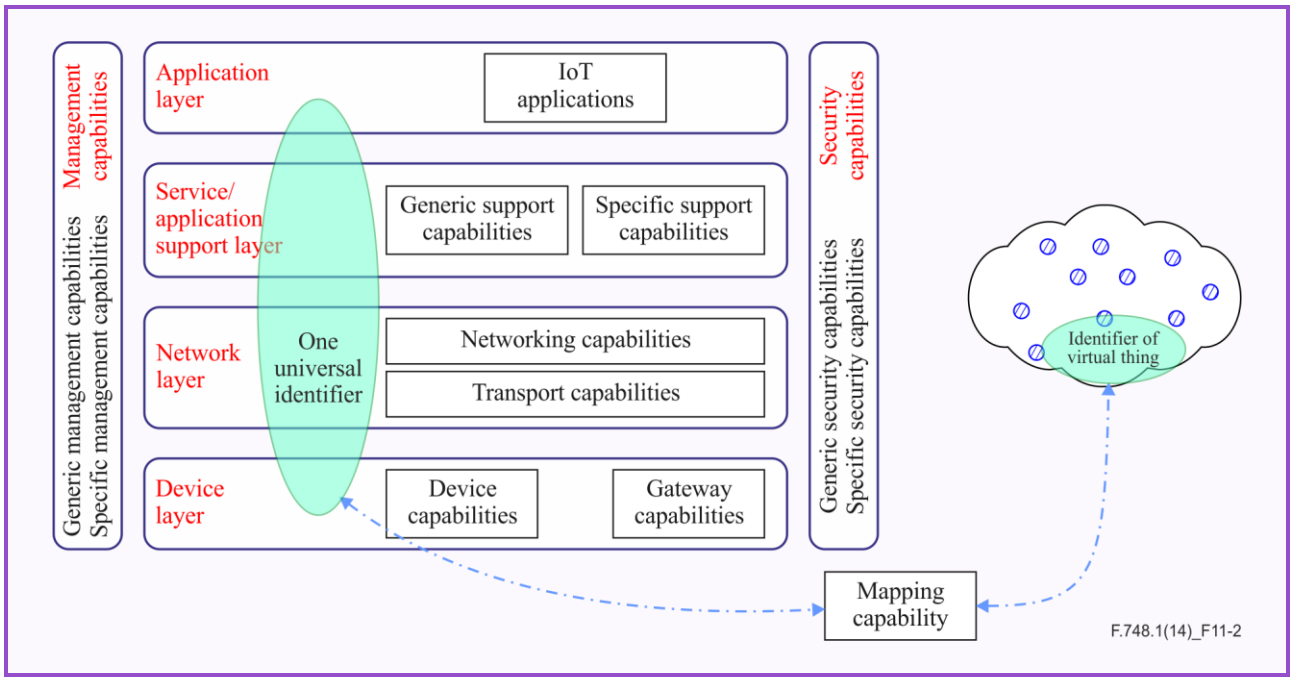


Figure 11-2 – Reference model of identification (case (b) in Figure 9-1)

Bibliography

[b-ITU-T F.771] Recommendation ITU-T F.771 (2008), *Service description and requirements for multimedia information access triggered by tag-based identification*; Amendment 1 (2014).

[b-3GPP 22.988] 3GPP TR 22.988 v12.2.0 (2012-09), *Study on alternatives to E.164 for Machine-Type Communications (MTC).*

[b-3GPP 23.003] 3GPP TS 23.003 v11.0.0 (2011-12), *Numbering, addressing and identification*.


Internet
of Things
Everything
will be
connected.


MULTIMED
FUN!!

Y.4802/H.642.2

Multimedia information access triggered by tag-based identification – Registration procedures for identifiers

Multimedia information access triggered by tag-based identification – Registration procedures for identifiers

Summary

Recommendation ITU-T H.642.2 describes registration procedures for the identifier scheme defined in Recommendation ITU-T H.642.1. It also designates the Registration Authority (RA) in charge of implementing these procedures.

History

Edition	Recommendation	Approval	Study Group
1.0	ITU-T H.642.2	2012-06-29	16

Keywords

Identification scheme, identifier, registration authority.

Table of Contents

Recommendation ITU-T Y.4802/H.642.2

Multimedia information access triggered by tag-based identification – Registration procedures for identifiers

1 Scope

This Recommendation defines registration procedures of the identification scheme defined by [ITU-T H.642.1] for the first level code (1LC) value of 0001_2. The identification scheme consists of a first level code (1LC), a second level code (2LC), class and elements such as a third level code (3LC), and a fourth level code (4LC). A 2LC is pre-assigned to ITU Member States, and then a 3LC is allocated by the registrant of the 2LC, which is called the second level registration authority (RA). The mechanism is meant for the distributed RA hierarchy.

Therefore, this Recommendation specifies as follows:

– registration procedures for the 2LC and 3LC;

– responsibilities of the second level RA; and

– procedures for the operation of the second level RA.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T F.771] Recommendation ITU-T F.771 (2008), *Service description and requirements for multimedia information access triggered by tag-based identification*.

[ITU-T H.621] Recommendation ITU-T H.621 (2008), *Architecture of a system for multimedia information access triggered by tag-based identification*.

[ITU-T H.642.1] Recommendation ITU-T H.642.1 (2012), *Multimedia information access triggered by tag-based identification – Identification scheme*.

[ITU-T H.642.3] Recommendation ITU-T H.642.3 | ISO/IEC 29177 (2012), *Information technology – Automatic identification and data capture technique – Identifier resolution protocol for multimedia information access triggered by tag-based identification*.

[ISO 3166-1] ISO 3166-1:2006 *Codes for the representation of names of countries and their subdivisions – Part 1: Country codes*, plus its Cor.1 (2007).

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 class [ITU-T H.642.1]: Part of an identifier that defines the layout and interpretation of the following bit string inside the identifier, especially the length of the third level code (3LC) and fourth level code (4LC).

3.1.2 country name [ISO 3166-1]: Name of a country, dependency, or other area of particular geopolitical interest.

3.1.3 first level code (1LC) [ITU-T H.642.1]: Part of the identifier that represents the identifier sub-blocks.

3.1.4 fourth level code (4LC) [ITU-T H.642.1]: Part of the identifier that serializes individual multimedia information and services.

3.1.5 ITU-T H.642 identification scheme [ITU-T H.642.1]: Name given to the identification scheme defined in [ITU-T H.642.1].

3.1.6 second level code (2LC) [ITU-T H.642.1]: Part of the identifier assigned to ITU Member States.

3.1.7 third level code (3LC) [ITU-T H.642.1]: Part of the identifier assigned to a registration authority (RA) that handles the allocation of the subspace to other organizations.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 numeric-3 code element: The numeric code found in column 6 of the table in clause 9 of [ISO 3166-1].

3.2.2 second level RA: The organization to which a second level code (2LC) is assigned and that manages the allocation of third level codes (3LCs) under class from 1 to 14.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

1LC First Level Code

2LC Second Level Code

3LC Third Level Code

4LC Fourth Level Code

RA Registration Authority

5 Conventions

None.

6 Registration authority for a second level code

The second level code (2LC) is assigned to ITU Member States and its value is taken from the values assigned to countries in [ISO 3166-1]. Each ITU Member State receives four 2LCs using the formula:

C*4, C*4+1, C*4+2, C*4+3,

where C is the numeric-3 code (without leading zeroes) element defined in [ISO 3166-1] for the specific country name.

Therefore, a registration authority (RA) is not needed for a 2LC.

6.1 Selection

The selection of the party to run the second level RA at a given 2LC shall be managed by the administration of each Member State, or a delegated organization from the administration.

6.2 Announcement

When an organization is selected to run a second level RA at a pre-assigned 2LC (see clause 6.1), the following information shall be sent to the Director of TSB (Telecommunication Standardization Bureau), so that interoperability between different second level RAs can be maintained:

a) the assigned 2LC value from the pre-assigned 2LCs;

b) name of the initial organization to operate the second level RA;

c) address of the initial organization;

d) date of the initial assignment;

e) date of the last transfer of assignment, if allowed (updatable);

f) name of the current organization (updatable);

g) address of the current organization (updatable);

h) the name, title, postal/e-mail address, telephone/facsimile number of a contact person within the organization (updatable);

i) date of the last update (updatable).

6.3 Change of information

The information provided in clause 6.2 for organization identified by a 2LC may change from time to time. The Director of TSB shall be notified of all such changes in a timely manner (within 30 days of such change).

6.4 Publication

The Director of TSB shall post the information concerning the selection of second level RAs at a pre-assigned 2LC (clause 6.2) on the ITU website and publish regular updates in the Operational Bulletin of the International Telecommunication Union.

7 Second level RA

7.1 Responsibilities

The responsibilities of the second level RA are as follows:

a) to receive applications for third level code (3LC) (the required content of the application is specified in clause 7.3.1);

b) to process applications within 30 days of receipt of the application form;

c) if the application is accepted according to the criteria of clause 7.2, to allocate the 3LC and to send a registration announcement to the applicant;

d) if the application is not accepted, to send a notice of rejection;

e) to maintain a publicly available register of allocated 3LCs (see clause 7.3.4);

f) to provide the necessary resources to operate a resolution server, based on Recommendation ITU-T H.642.3 and future protocols that are derived from it, and new protocols that may emerge.

The permitted fee structure shall be on a cost-recovery basis.

7.2 Criteria for acceptance

7.2.1 Time-scale

The application for 3LC shall identify the time-scale within which the 3LC is to be used for multimedia information access, triggered by tag-based identification. The application shall be rejected if the time-scale exceeds 12 months, and can be voided if it is not in use within that time-scale.

7.2.2 Size

Based on the request and estimated usage of codes, the second level RA can suggest that the applicant apply for a subspace under a particular class.

7.2.3 Nature of service

The service for which the allocation of a 3LC is requested shall be services which require interchange between multiple applications in an open environment in the long run.

7.3 Detailed procedures for the operation of the second level RA

7.3.1 Registration application for a 3LC

The application shall include at least the following information:

a) name of the organization submitting the application;

b) name, postal mail address, and e-mail address; optionally, telephone and fax numbers for the contact point within the requesting organization;

c) full identification of the person submitting the application (including their role in the organization);

d) time-scale for application of the allocated 3LC;

e) estimated usage (number of codes) in code subspace under 3LC.

7.3.2 Registration announcement

The second-level RA shall send a registration announcement to an applicant when the assignment of a 3LC has been agreed. The registration announcement shall include at least the following information:

a) the name of the organization submitting the application;

b) the name, postal/electronic mail address and telephone/facsimile number for the contact point within the requesting organization;

c) the full identification of the person submitting the application (including their role in the organization);

d) the 3LC value and the class value assigned.

7.3.3 Notice for rejection

Any applications for a 3LC shall be rejected by the second level RA when it contains incomplete or incomprehensible information, or if the conditions of criteria of acceptance are not satisfied.

The second level RA shall send a notice of rejection to an applicant when the assignment of a 3LC has been rejected. The notice of rejection shall include at least the following information:

a) the name of the organization submitting the application;

b) the name, postal/electronic mail address and telephone/facsimile number for the contact point within the requesting organization;

c) full identification of the person submitting the application (including their role in the organization);

d) the reason for rejection, which *inter alia* may be:

- the absence of a proper fee;
- incomplete or incomprehensible information in application;
- the justification for inclusion in the register (as defined in this standard) is not adequate.

7.3.4 Content of the register

At a minimum, the register shall contain:

a) the assigned 3LC and the class under which the 3LC is used;

b) name of the initial applicant;

c) address of the initial applicant;

d) date of the original assignment;

e) date of the last transfer of assignment, if allowed (updatable);

f) name of the current owner (updatable);

g) address of the current owner (updatable);

h) if the owner is an organization, the name, title, postal/e-mail address (with the email address protected against robot harvesting), and telephone/facsimile number of a contact person within the organization (updatable);

i) date of last update (updatable).

7.3.5 Change of registration information

The registered organization identified by a 3LC shall not significantly change from the original application, but supporting information, such as the information provided in clause 7.3.1, may change from time to time. The second level RA shall be notified of all such changes, and shall update the register, maintaining an audit trail of earlier information.

7.3.6 Appeals process

In response to a notice of rejection of a 3LC application, the applicant can submit to a second level RA a supplement to its original application that responds to the reason(s) for rejection.

7.4 Transfer of register entries held by the second level RA

The register entries held by the second level RA shall be made available to any subsequently appointed second level RA.

Identification

Y.4804/H.642.1

Multimedia information access triggered by tag-based identification – Identification scheme

Multimedia information access triggered by tag-based identification – Identification scheme

Summary

Recommendation ITU-T H.642.1 defines an identification scheme for multimedia information access triggered by tag-based identification. Multimedia information associated with such an identifier can be retrieved using a resolution process.

This Recommendation provides a framework for accommodating the legacy existing identification schemes together with the new identification scheme defined here using a pair of object identifiers for the identification scheme and identifier itself.

History

Edition	Recommendation	Approval	Study Group
1.0	ITU-T H.642.1	2012-06-29	16

Keywords

Identification scheme, identifier.

Table of Contents

Introduction

The identification scheme defined in this Recommendation provides a new method for accessing multimedia content without typing its address on a keyboard or inputting the name of objects and/or places of relevant information. This is a major communication service that uses an identifier in data carriers such as radio frequency identifications (RFIDs), smart cards and barcodes.

The purpose of this Recommendation is to specify a new identification scheme to be used in applications and services where the existing identification schemes and their combinations cannot be used (see Appendix I for some examples).

From the standpoint of compatibility and reusability of existing identifier schemes, this Recommendation provides a framework to support multiple identification schemes. In this framework, it is possible that multiple existing identification schemes together with the proposed new identification scheme can be used for identification where appropriate. Each identification scheme is identified by an object identifier (OID) according to [ITU-T X.660].

Clause 6 introduces the concept of multiple identification schemes for multimedia information access triggered by tag-based identification.

Clause 7 specifies a new generic identification scheme for multimedia information access triggered by tag-based identification.

Appendix I provides additional information which explains why a new generic identification scheme is needed.

Recommendation ITU-T Y.4804/H.642.1

Multimedia information access triggered by tag-based identification – Identification scheme

1 Scope

This Recommendation defines an identification scheme for multimedia information access triggered by tag-based identification. This identification scheme is mainly used in the multimedia information system architecture defined in [ITU-T H.621]. It also satisfies the requirements defined in [ITU-T F.771].

This Recommendation does not define encoding rules to store the identifier value into data carriers such as barcode tags and RFID tags. When stored in a data carrier, the OID and identifier value shall be encoded according to the relevant international standards if such standards exist for the type of data carrier.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T F.771] Recommendation ITU-T F.771 (2008), *Service description and requirements for multimedia information access triggered by tag-based identification*.

[ITU-T H.621] Recommendation ITU-T H.621 (2008), *Architecture of a system for multimedia information access triggered by tag-based identification*.

[ITU-T H.642.2] Recommendation ITU-T H.642.2 (2012), *Multimedia information access triggered by tag-based identification – Registration procedures for identifiers*.

[ITU-T X.660] Recommendation ITU-T X.660 (2011) | ISO/IEC 9834-1:2012, *Information technology – Procedures for the operation of object identifier registration authorities: General procedures and top arcs of the international object identifier tree*.

[ITU-T X.668] Recommendation ITU-T X.668 (2008) | ISO/IEC 9834-9:2008, *Information technology – Open Systems Interconnection – Procedures for the operation of OSI Registration Authorities: Registration of object identifier arcs for applications and services using tag-based identification*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 ID resolution [ITU-T F.771]: A function to resolve an identifier into associated information.

3.1.2 multimedia information [ITU-T F.771]: Digital information that uses multiple forms of information content and information processing, such as text, pictures, audio, video, three-dimensional panoramic pictures and digital maps, which informs or entertains users.

3.1.3 object identifier [b-ITU-T X.680]: A globally unique value associated with an object to unambiguously identify it.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 class: Part of an identifier that defines the layout and interpretation of the following bit string inside the identifier, especially the length of the third level code (3LC) and fourth level code (4LC).

3.2.2 first level code (1LC): Part of the identifier that represents the identifier sub-blocks.

3.2.3 fourth level code (4LC): Part of the identifier which serializes individual multimedia information and services.

3.2.4 ITU-T H.642 identification scheme: Name given to the identification scheme defined in this Recommendation.

3.2.5 identification scheme: Definition and description of the structure of identifiers.

3.2.6 identifier value: String of characters that represents the value of the identifier.

3.2.7 second level code (2LC): Part of the identifier assigned to ITU Member States.

3.2.8 third level code (3LC): Part of the identifier assigned to an RA that handles the allocation of the subspace to other organizations.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

1LC	First Level Code
2LC	Second Level Code
3LC	Third Level Code
4LC	Fourth Level Code
EPC	Electronic Product Code
GIAI	Global Individual Asset Identifier
GRAI	Global Returnable Asset Identifier
ID	Identifier
IP	Internet Protocol
ISBN	International Standard Book Number
ISSN	International Standard Serial Number
OID	Object Identifier
RA	Registration Authority
RFID	Radio Frequency Identification
SGLN	Serialized Global Location Number
SGTIN	Serialized Global Trade Item Number

SSCC Serial Shipping Container Code

URL Uniform Resource Locator

URN Uniform Resource Name

UUID Universally Unique Identifier

5 Conventions

In this Recommendation, xx_2 denotes that number xx is expressed as a binary (base-2) number.

6 Concept of an identification scheme for multimedia information access triggered by tag-based identification

6.1 Overview

An identification scheme in this Recommendation is designed to discriminate multimedia information and services associated with "objects" or "places" of the real world in the architecture defined by [ITU-T H.621].

A data carrier such as an RFID tag which triggers access can be placed anywhere and its whereabouts is out of the scope of this Recommendation.

6.2 Uniqueness of ID value

Figure 1 gives an overview of the general approach to accommodate multiple identification schemes. Many different identification schemes can be used for multimedia information access triggered by tag-based identification. Each identification scheme should be identified by an object identifier (OID) under the arc {joint-iso-itu-t(2) tag-based(27)} as specified in [ITU-T X.668]. For example, in Figure 1, the top row indicates that Identification scheme 1 is identified by OID arch {joint-iso-itu-t(2) tag-based(27) i_1}.

Identifier schemes which are widely used and already have an OID allocated to them (not under arc {joint-iso-itu-t(2) tag-based (27)}) can continue to use that OID. In Figure 1, "Existing identification scheme 1", that has an OID value not under arc {joint-iso-itu-t(2) tag-based(27)} is identified by OID {x_1 y_1 z_1}.

For a given identifier scheme (identified by an OID), the uniqueness of identifier values is required to be managed by the organization which is allocating those values. That is, the uniqueness of identifier values should be managed by the registration authority (RA) for that identifier scheme with the cooperation of those who store identifiers into data carriers.

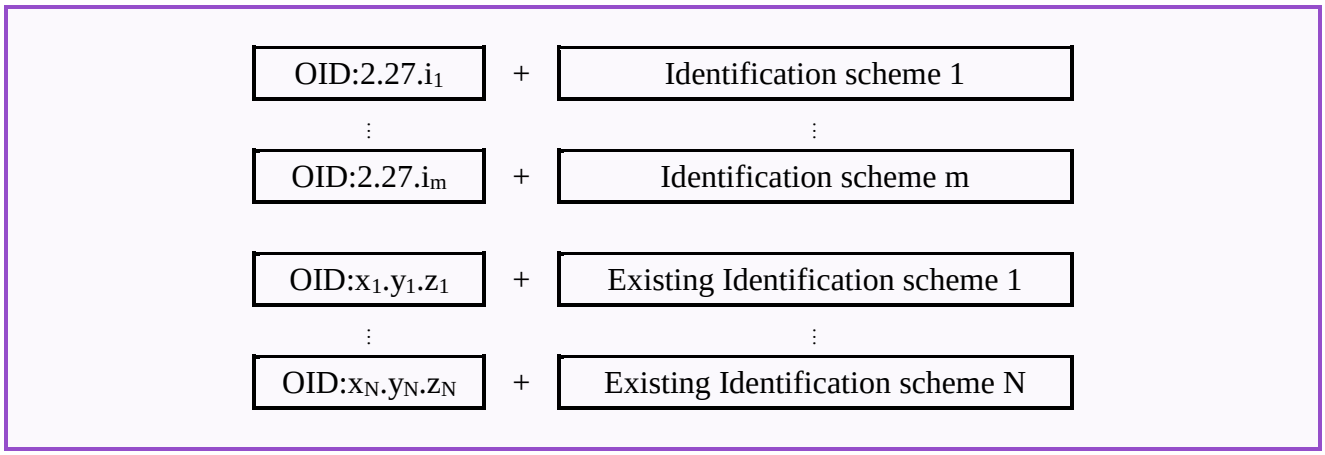


Figure 1 – Multiple identification schemes

7 ITU-T H.642 identification scheme for multimedia information access triggered by tag-based identification

7.1 Structure of an ITU-T H.642 identification scheme

The identification scheme defined in this Recommendation is called the ITU-T H.642 identification scheme. The ITU-T H.642 identification scheme consists of the fields: 'First Level Code' (1LC), 'Second Level Code' (2LC) and 'Class', followed by some of the elements ('Third Level Code' (3LC) and 'Fourth Level Code' (4LC)) as shown in Figure 2.

<table>
<tr><td>1LC</td><td>2LC</td><td>CLASS</td><td colspan="8">3LC / 4LC</td></tr>
<tr><td>4 bits</td><td>16 bits</td><td>4 bits</td><td>8 bits</td><td>16 bits</td><td>16</td><td>16</td><td>16</td><td>16</td><td>16</td></tr>
<tr><td rowspan="16">0001_2</td><td rowspan="16">16 bits</td><td>0000_2</td><td colspan="8">Reserved</td></tr>
<tr><td>0001_2</td><td colspan="8">Reserved</td></tr>
<tr><td>0010_2</td><td colspan="8">Reserved</td></tr>
<tr><td>0011_2</td><td colspan="8">Reserved</td></tr>
<tr><td>0100_2</td><td colspan="8">Reserved</td></tr>
<tr><td>0101_2</td><td colspan="8">Reserved</td></tr>
<tr><td>0110_2</td><td colspan="8">Reserved</td></tr>
<tr><td>0111_2</td><td colspan="8">Reserved</td></tr>
<tr><td>1000_2</td><td colspan="8">Reserved</td></tr>
<tr><td>1001_2</td><td colspan="2">3LC</td><td colspan="6">4LC</td></tr>
<tr><td>1010_2</td><td colspan="3">3LC</td><td colspan="5">4LC</td></tr>
<tr><td>1011_2</td><td colspan="4">3LC</td><td colspan="4">4LC</td></tr>
<tr><td>1100_2</td><td colspan="5">3LC</td><td colspan="3">4LC</td></tr>
<tr><td>1101_2</td><td colspan="6">3LC</td><td colspan="2">4LC</td></tr>
<tr><td>1110_2</td><td colspan="7">3LC</td><td>4LC</td></tr>
<tr><td>1111_2</td><td colspan="8">Reserved</td></tr>
<tr><td>0000_2</td><td colspan="9">Reserved for backward compatibility (see clause 7.2)</td></tr>
<tr><td>0010_2 ~ 1111_2</td><td colspan="9">Reserved for future use</td></tr>
</table>

Figure 2 – Structure of an identifier

Table 1 – Field name and length

Field name	Length
1LC	4 bits
2LC	16 bits
CLASS	4 bits
3LC, 4LC	Variable length, but the sum of these is 104 bits

Value of class	Length of 3LC	Length of 4LC
9	8 bits	96 bits
10	24 bits	80 bits
11	40 bits	64 bits
12	56 bits	48 bits
13	72 bits	32 bits
14	88 bits	16 bits

7.2 First level code

1LC is 4 bits long and represents the identifier of sub-blocks. It serves as version information.

1LC from 0010_2 to 1111_2 are reserved for future use.

Part of the identifier sub-block with 1LC of 0000_2 shall be reserved to avoid conflict with the 'ucode' identification scheme [b-uID].

7.3 Second level code (2LC)

2LC is 16 bits long and is assigned to ITU Member States as specified in [ITU-T H.642.2].

7.4 Class

Class indicates the layout of a particular identifier format. There are different formats that correspond to Class 9, 10…, 14 as shown on Figure 2 (Class 0, 1, 2…, 8, 15 are reserved for future use).

7.5 Third level code (3LC)

The 3LC assignment is done by an entity that is assigned the 2LC of that subspace (see clause 7.3). A 3LC is assigned to an RA that handles the allocation of the subspace to other organizations. The subspace controlled by this RA is called the 3LC space. It has six different sizes from 16 bits to 96 bits by steps of 16 bits.

3LC spaces are assumed to be managed by entities such as companies, associations, universities, schools, local governments, etc.

7.6 Fourth level code (4LC)

4LC is a serialization number to identify individual multimedia information and services.

Appendix I

Survey on identification schemes

(This appendix does not form an integral part of this Recommendation.)

I.1 Introduction

The purpose of this appendix is to survey whether existing identification schemes (most of them being internationally standardized) satisfy the requirements found in [ITU-T F.771].

The currently existing schemes may not satisfy all the needs identified by [ITU-T F.771] that came later than the schemes themselves. The ITU-T H.642 identification scheme in this Recommendation is a complementary scheme to existing identification schemes. It is meant to fill the gaps left by existing schemes and identified by [ITU-T F.771] for particular classes of applications. It is complementary to existing schemes in this sense, and is not a competitor. This Recommendation outlines the framework to make these identification schemes co-exist in future applications to take advantage of all the existing schemes including the ITU-T H.642 identification scheme.

I.2 Need for a generic identification scheme for multimedia information access triggered by tag-based identification

A generic identification scheme in this Recommendation is applicable to multimedia information and services associated with any kind of objects and places. Any systems for multimedia information access triggered by tag-based identification should implement the generic identification scheme defined in this Recommendation.

I.3 Requirements on the identifier for multimedia information access triggered by tag-based identification

[ITU-T F.771] defines in clause 7.3 five requirements for identifiers for multimedia information access triggered by tag-based identification. The following gives clarification of each identifier requirement to stress that the identification scheme defined in this Recommendation satisfies these identifier requirements.

– **ID-001: identifier is recommended to be used by different applications**.

  Multimedia information access triggered by tag-based identification specifies a mechanism of information processing and communication, but no applications. The mechanism is proposed and recommended to be used by any application. In other words, it is for "general purpose" use. Also, the identifiers should be for general purposes too. This means, it is recommended to be used by different applications. Most identification schemes can be used for many applications, so almost every identification scheme satisfies this requirement.

– **ID-002: identifier is required to be assigned for real-world entities such as physical/logical objects, persons and places**.

  "Tag-based identification" deals with every type of real-world entity e.g. tangible objects (items), persons and places because all of these can be targets of tag-based identification. (Although "people" is mentioned here, this Recommendation does not attempt to define an identifier for persons). In some applications, the information service is triggered by some tangible objects (items), persons or places. Thus, identifiers must be assignable to tangible objects, logical objects, people and places. It is important to note that many application target areas do not have an applicable international standard identification scheme. For example, locations with meanings, vegetables, fish, meat, houses, roads, buildings, bridges, boats, art items, posters and so on, do not have proper identification schemes based on an

international standard. This Recommendation is meant for such wider areas which lack any previous standard ID scheme.

– **ID-003: identifier is required to be issuable by any organization such as companies, non-profit organizations, governments and individual users**.

– ID-003 is a requirement for the governance of identification schemes. To satisfy this requirement, for example, even an individual without any corporate affiliation should be able to obtain identifiers, and identifiers must be allocated at a low enough cost. For example, if an identification scheme has a short company code bit field, and it is too short to be used by all the companies in the country or in the world, it does not satisfy this requirement. If the company code cannot be assigned to non-company organizations such as governments, NPOs, small businesses, schools and individuals, it does not satisfy this requirement. The governance of this Recommendation should satisfy this requirement.

– **ID-004: identifier is required to be globally unique so that the multimedia information access triggered by the identifier is globally available**.

Essentially, the identification scheme proposed must be a globally unique numbering system. When the same identifier is assigned to two different entities, confusion arises. However, some identification schemes do not follow this principle entirely. For example, an IP address system has a private address subspace in its numbering space, such as "192.168.1.1". Everyone can assign the private address to the network interfaces of his/her own networked machines. Some supply chain management (SCM) identification schemes have an "in-house ID code area", which can be used by shops or factories freely, thus the code area is not globally unique. In [b-ITU-T X.667], identifiers are generated automatically by a standard algorithm and procedure, so each identifier is assured to be almost globally unique. However, in theory, there is a small possibility of collisions when random number generators are used. Additionally, there may be bugs in the identifier generating software, and some unauthorised people may intentionally issue duplicate identifiers. Therefore, [b-ITU-T X.667] does not fully satisfy this requirement. The governance of this Recommendation is meant to satisfy this requirement.

– **ID-005: multiple identifier schemes are required to be supported**.

This is an important requirement to accommodate both the existing and the yet-to-appear future identification schemes.

Among the requirements outlined above, ID-005 is not a requirement for the identification scheme itself, but for the system that utilizes identifiers. Therefore, this requirement is excluded from the use case analysis. ID-001 is also excluded from the use case analysis because all the target identification schemes listed below satisfy this requirement.

I.4 Survey on existing identification schemes

I.4.1 Existing identification schemes

To make this use case analysis comprehensive, widely-used existing identification schemes from a wide range of application areas were studied.

I.4.1.1 EPC

The electronic product code (EPC) is a family of identification schemes proposed by EPCglobal. It is currently one of the most widely used identification schemes in the field of supply chain management, even though it is not a *de jure* international standard. It is designed to be used in a low-cost way for tracking goods using RFID technology, to meet the needs of various industries, to identify each item manufactured (unlike barcode systems), and to guarantee uniqueness for all EPC-compliant tags. This implies that the initial intention of EPC is to identify products manufactured by using RFID-based tags.

It contains a serialized global trade item number (SGTIN), a serial shipping container code (SSCC), a serialized global location number (SGLN), a global returnable asset identifier (GRAI) and a global individual asset identifier (GIAI).

I.4.1.2 ISO/IEC 15459-x

[b-ISO/IEC 15459-x] is an identification scheme designed for supply chain management because the original intended application of this identification scheme is for a supply chain management system.

I.4.1.3 ISO/IEC 6709

[b-ISO/IEC 6709] is an international standard representation of latitude, longitude and altitude for (fixed) geographic point locations. It is used by many GIS applications. But other kinds of ID-based applications such as a supply chain management system, which is intended for the identification of moving objects, cannot use this identification scheme. (It is impractical to replace identifiers whenever objects change their locations. We need a separation of the identifier value and the location information. The multiple identification schemes in [ITU-T H.621] would be effective here).

I.4.1.4 ISO 2108 International standard book number

The international standard book number (ISBN) is a numbering scheme for the identification of books.

I.4.1.5 ISO 3297 International standard serial number

The international standard serial number (ISSN) is a numbering scheme for the identification of serial publications such as newspapers, magazines, journals and annually published books globally.

I.4.1.6 Universally unique identifier

The universally unique identifier (UUID) in [b-ITU-T X.667] is an identification scheme designed for identifying information or components in the distributed systems without the assumption of any management systems of identifiers. One of the most famous implementation of UUID is Microsoft's GUID.

I.4.1.7 ISO/IEC 15963

[b-ISO/IEC 15963] is an identification scheme designed for RFID tags and mainly used for supply chain management. Since this identification scheme is designed only for RFID tags, it cannot be used for applications that require other kinds of ID tags such as a printing code.

I.4.1.8 ISO/IEC 11784

[b-ISO/IEC 11784] is an identification scheme designed for identifying animals.

NOTE – All of the above identification schemes other than EPC are *de jure* international standards, and also note that EPC is widely used.

I.4.2 Analysis

Surveys on existing identification schemes are in Table I.1. For comparison, the ITU-T H.642 identification scheme is also included in this analysis.

Table I.1 – Analysis of existing identification schemes

ID schemes	ID-002	ID-003	ID-004
EPC	Most identifiers are mainly assigned to objects, but not to logical objects. SSCC can be assigned for shipping containers and SGLN can be assigned only for locations.	Registered issuing agencies may issue identifiers and a hierarchical structure of issuing agencies is provided.	Each identifier is assured to be globally unique.
ISO/IEC 15459-x	Identifiers are mainly assigned for objects, but not places. Since this identification scheme is designed for ID tags, it is impossible to assign the identifier to any logical object.	Registered issuing agencies may issue identifiers and a hierarchical structure of issuing agencies is provided.	Each identifier is assured to be globally unique.
ISO 6709	Identifiers are assigned for geographic point locations. This identifier cannot be assigned to any objects that move around.	It is a representation for identifying geographic point locations, thus no one can issue the identifiers.	Each identifier is assured to be globally unique.
ISBN	ISBN numbers are assigned only for books and physical publications. ISBN numbers are not assignable for any logical objects and other physical objects.	Only affiliated organizations (sometimes national organizations) that manage books in the region or country may issue the ISBN numbers.	Identifiers are assured to be globally unique.
ISSN	ISSN numbers are assigned only for serial publications. ISSN numbers are not assignable for any other logical or physical objects.	Only affiliated organizations (sometimes national organizations) that manage serial publications in the region or country may issue the ISSN numbers.	Identifiers are assured to be globally unique.
UUID	Identifiers can be assigned for everything that can be identified.	Individual users or programs may issue identifiers. Since central coordination is difficult in this identification scheme, it is not suitable for organizations that want to control issuance of identifiers from the fixed numbering space that can freely be managed by the organization to hierarchically manage products.	Each identifier is almost assured to be globally unique, but there is small possibility of collisions (except if the UUID is registered as an OID).

Table I.1 – Analysis of existing identification schemes

ID schemes	ID-002	ID-003	ID-004
ISO/IEC 15963	Identifiers are mainly assigned for objects, but not places. Since this identification scheme is designed for RFID tags, it is impossible to assign the identifier to any logical objects.	This identifier is always written into RFID tags when a tag chip manufacturer produces an RFID tag. Only tag chip manufacturers can issues this identifier.	Each identifier is assured to be globally unique.
ISO/IEC 11784	Identifiers are assigned only for animals.	Registered issuing agencies may issue identifiers and a hierarchical structure of issuing agencies is provided.	Each identifier is assured to be globally unique.
ITU-T H.642 identification scheme	An ITU-T H.642 identifier can be assigned for multimedia information and services associated with any object or place.	Registered issuing agencies may issue identifiers and a hierarchical structure of issuing agencies is provided.	Each identifier is assured to be globally unique.

I.4.3 Summarized analysis

Table I.2 summarizes the information given in the preceding clauses. According to this investigation, it is considered that no existing identification scheme other than the ITU-T H.642 identification scheme satisfies all the requirements described in [ITU-T F.771].

Table I.2 – Summary of analysis

	ID-002	ID-003	ID-004
EPC	U	S	S
ISO/IEC 15459-x	U	S	S
ISO 6709	U	U	S
ISBN	U	S	S
ISSN	U	S	S
UUID	S	U	P
ISO/IEC 15963	U	P	S
ISO/IEC 11784	U	S	S
ITU-T H.642 identification scheme	S	S	S
S: satisfied, U: unsatisfied, P: Not strictly satisfied			

Bibliography

[b-ITU-T X.667] Recommendation ITU-T X.667 (2008) | ISO/IEC 9834-8:2008, *Information technology – Open Systems Interconnection – Procedures for the operation of OSI Registration Authorities: Generation and registration of Universally Unique Identifiers (UUIDs) and their use as ASN.1 object identifier components*.

[b-ITU-T X.680] Recommendation ITU-T X.680 (2002) | ISO/IEC 8824-1:2002, *Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation*.

[b-ISO 2108] ISO 2108 (2005), *Information and documentation – International standard book number (ISBN)*.

[b-ISO 3297] ISO 3297 (2007), *Information and documentation – International standard serial number (ISSN)*.

[b-ISO 6709] ISO 6709 (2008), *Standard representation of geographic point location by coordinates*.

[b-ISO 11784] ISO/IEC 11784 (1996), *Radio frequency identification of animals – Code structure*.

[b-ISO/IEC 15459-x] ISO/IEC 15459-x, *Information technology – Unique identifiers*.

[b-ISO/IEC 15963] ISO/IEC 15963 (2009), *Information technology – Radio frequency identification for item management – Unique identification for RF tags*.

[b-uID] uID (2009), *Ubiquitous Code: ucode*, uID center, Version 1.A0.10.

Other related publications:

United for Smart Sustainable Cities

Shaping Smarter and more Sustainable cities: Striving for sustainable development goals

International
Telecommunication
Union
Place des Nations
CH-1211 Geneva 20
Switzerland

ISBN: 978-92-61-17981-6

9 789261 179816

July 2016

About ITU-T and IoT and its applications including smart cities and communities (SC&C):
http://www.itu.int/en/ITU-T/ssc/

E-mail: **tsbsg20@itu.int**

Printed in Switzerland
Geneva, 2016
Photo credits: Shutterstock