ISO/IEC CD 30141:20160910(E)

ISO/IEC JTC 1/WG 10

Secretariat: ANSI

Information technology – Internet of Things Reference Architecture (IoT RA)

CD stage

Contents

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see the following URL: www.iso.org/iso/foreword.html.

The committee responsible for this document is Technical Committee ISO/IEC JTC 1, [Information technology].

Introduction

Internet of Things (IoT) has a broad use in industry and society today and it will continue to develop for many years to come. Various IoT applications and services have adopted IoT techniques to provide capabilities that were not possible a few years ago. IoT is one of the most dynamic and exciting area of the IT. It involves the connecting of physical entities ("things") with IT systems through networks. Foundational to IoT are the electronic devices that interact with physical world. Sensors get the information f physical world, while actuators can act on it. Both sensors and actuators can be in many forms such as thermometer, accelerometers, video cameras, microphones, relays, heathers or industrial equipment for manufacturing or process controlling. Mobile technology, cloud computing, big data and deep analytics (predictive, cognitive, real-time and contextual) play important roles by gathering and processing data to achieve the final result of controlling physical entities.

IoT uses much of existing technology and combines this for improving operations and lowering costs, or for creating new products and business models, or for driving engagement and customer experiences etc. IoT covers a very wide spectrum of applications and represents the integration of systems from different vertical sectors (enterprise, consumer, government, industries etc.).

Several forecasts indicate that IoT will connect 50 billion devices worldwide by the year 2020. There are a number of possible application areas such as: smart city, smart grid, smart home/building, digital agriculture, smart manufacturing, intelligent transport system, e-Health. IoT is an enabling technology that consists of many supporting technologies, for example, different types of communication networking technologies, information technologies, sensing and control technologies, software technologies, device/hardware technologies. This international standard is based on widely used enabling technologies that are defined in standards from several organizations such as ISO, IEC, ITU, IETF, IEEE, ETSI, 3GPP, W3C, etc.

This document provides a standardized IoT reference architecture using a common vocabulary, reusable designs and industry best practices. It uses a top down approach, beginning with collecting the most important characteristics of IoT, abstracting those into a generic IoT conceptual model, deriving from the conceptual model to a high level system based reference model and then breaking down from reference model to the five architecture views (functional view, system view, user view, information view and communication view) from different perspectives.

This document can be served as a base on which to develop specific IoT applications. Therefor the target readers are engineers and technical managers who are going to develop or design IoT applications.

Information technology – Internet of Things Reference Architecture (IoT RA)

1 Scope

This document specifies the general IoT reference architecture in terms of defining system characteristics, a conceptual model, a reference model and architecture views for IoT.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 20924, *Internet of Things — Definition and Vocabulary*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 20924.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at http://www.iso.org/obp

— IEC Electropedia: available at http://www.electropedia.org/

[NOTE] When this document is "stable" review content, add needed terms that not are covered by ISO/IEC 20924.

4 Symbols and abbreviated terms

5Vs	Volume, Velocity, Veracity, Variability, and Variety
6LoWPAN	IPv6 over Low Power Wireless Personal Area Network
API	Application Programming Interface
ASD	Application Service Domain
CM	Conceptual Model
DHCP	Dynamic Host Configuration Protocol
FQDNs	Fully Qualified Domain Names
HTTP	Hypertext Transfer Protocol
IoT	Internet of Things
IoT RA	Internet of Things Reference Architecture

LAN	Local Area Network
LOB	Line of Business
OMD	Operation & Management Domain
PAN	Personal Area Network
PED	Physical Entity Domain
PII	Personally Identifiable Information
QoS	Quality of Service
RA	Reference Architecture
RID	Resource & Interchange Domain
RM	Reference Model
SAP	Session Announcement Protocol
SCD	Sensing & Controlling Domain
TCP/IP	Transmission Control Protocol/Internet Protocol
UML	Universal Modelling Language
UD	User Domain
URI	Uniform Resource Identifier
VPN	Virtual Private Network
WAN	Wide Area Network
WLAN	Wireless Local Area Network

5 IoT RA goals and objectives

5.1 General

IoT is defined as an infrastructure of interconnected physical entities, systems and information resources together with the intelligent services which can process and react information of both the physical world and the virtual world and can influence activities in the physical world.

The IoT Reference Architecture (IoT RA) described in this document provides the conceptual model (CM), reference model (RM) and reference architectures (RA) from different architectural views. The IoT RA not only outlines "what" the overall structured approach for the construction of IoT systems by means of the architectural structure description, but also indicates "how" the architecture and its domains or entities will operate. In short, the IoT RA provides rules and guidance for developing IoT system architecture.

The IoT RA serves the following goals:

1) to describe the characteristics of IoT systems;
2) to define the domains of the IoT system;
3) to describe CM, RM of IoT systems; IoT architecture views; and
4) to describe interoperability of IoT system's entities.

Each IoT system has specific system requirements that should be met, and the specific system requirements can vary from one IoT system to another per user group and/or domain. The IoT RA provides the generic parts as a starting point which can be used to create a system specific architecture.

The IoT RA supports the following important standardization objectives:

1) to enable the production of a coherent set of international standards for IoT;
2) to provide a technology-neutral reference point for defining standards for IoT; and
3) to encourage openness and transparency in the development of a target IoT system architecture and in the implementation of the IoT system.

The IoT RA is also intended to:

1) facilitate the understanding of the overall structure of IoT systems;
2) illustrate and provide understanding of IoT RA from different architectural views;
3) provide a technical reference to enable the international community to understand, discuss, categorize and compare IoT systems; and
4) facilitate the analysis of candidate use cases/applications including data/information flows.

5.2 Structural overview

The IoT RA described in this document provides:

1) A CM containing common entities and their relations, and
2) A RM and different architecture views

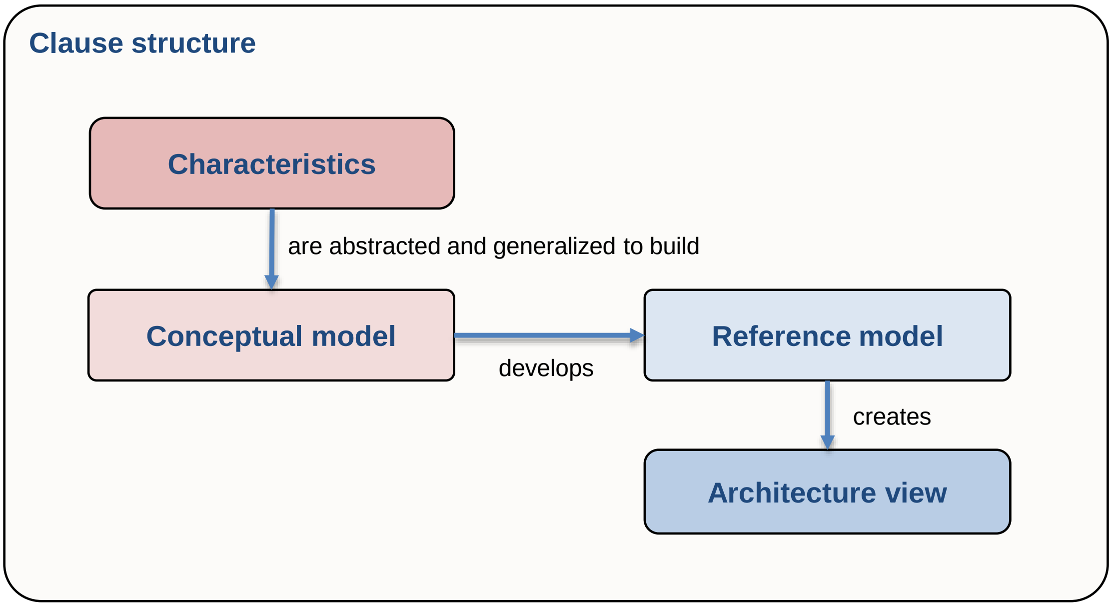

Figure 1 – IoT RA structure

5.2.1 CM

CM contains the following elements:

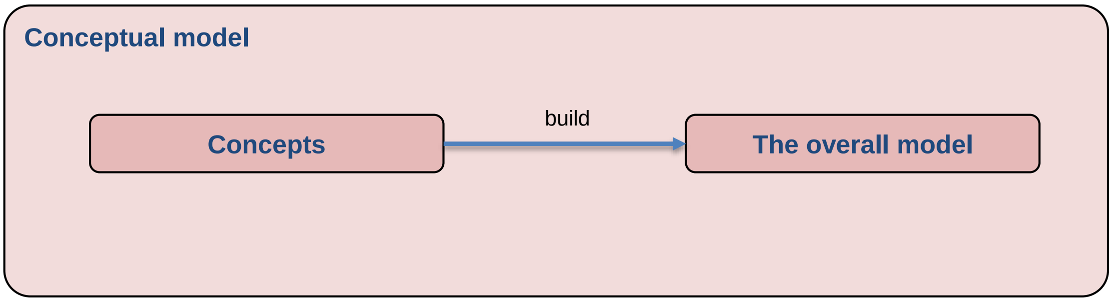

Figure 2 – CM structure

5.2.2 RM and architecture views

CM is described in Clause 7 and RM contains the following parts:

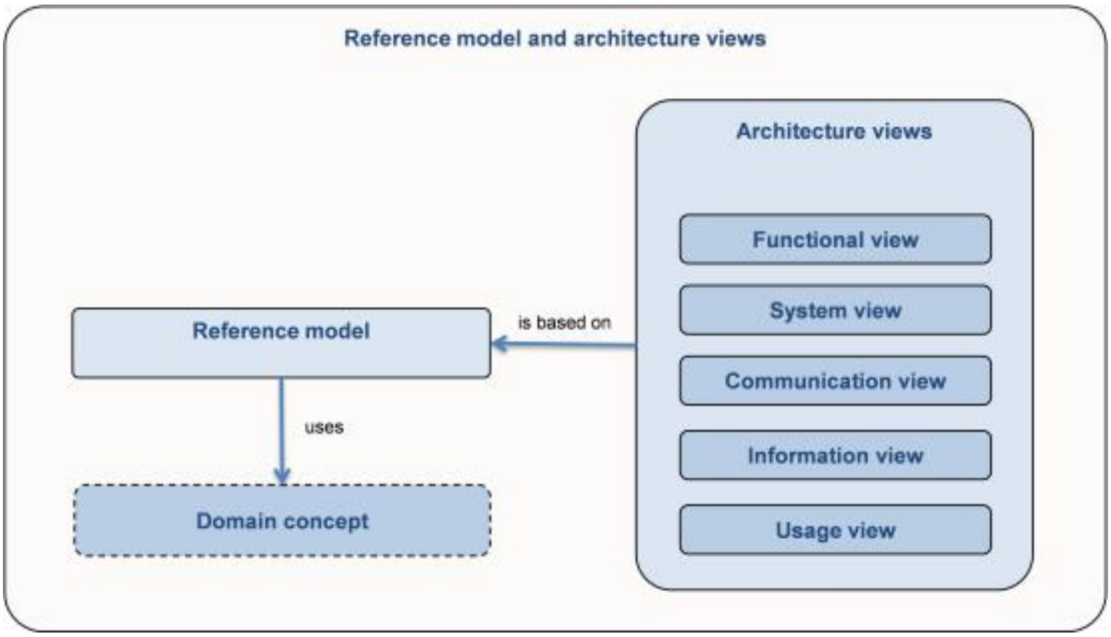

Figure 3 – RM and architecture views

The respective views are described in clause 8.

6 Main characteristics of IoT systems

6.1 Introduction

This clause provides characteristics of IoT systems. Functions based on all or a part of these characteristics can be implemented in IoT systems according to services and operations. Characteristics are sorted in alphabetical order.

Table 1 – Main characteristics of IoT systems

<table>
<tr><td>Grouping</td><td>1st Level</td></tr>
<tr><td rowspan="8">6.1 IoT System Characteristics</td><td>6.1.1 Auto-configuration</td></tr>
<tr><td>6.1.2 Function and management capabilities separation</td></tr>
<tr><td>6.1.3 Highly distributed systems</td></tr>
<tr><td>6.1.4 Network communication</td></tr>
<tr><td>6.1.5 Network management and operation</td></tr>
<tr><td>6.1.6 Real-time capability</td></tr>
<tr><td>6.1.7 Self-description</td></tr>
<tr><td>6.1.8 Service subscription</td></tr>
</table>

6.2 IoT Service Characteristics	6.2.1 Content-Awareness
	6.2.2 Context-Awareness
	6.2.3 Timeliness
6.3 IoT Component Characteristics	6.3.1 Composability
	6.3.2 Discoverability
	6.3.3 Modularity
	6.3.4 Network connectivity
	6.3.5 Shareability
	6.3.6 Unique identification
6.4 Compatibility	6.4.1 Legacy support
	6.4.2 Well defined components
6.5 Usability	6.5.1 Flexibility
	6.5.2 Manageability
6.6 Robustness	6.6.1 Accuracy
	6.6.2 Reliability
	6.6.3 Resilience
6.7 Security	6.7.1 Availability
	6.7.2 Confidentiality
	6.7.3 Integrity
	6.7.4 Safety
6.8 Protection of Personally Identifiable Information	
6.9 Other Characteristics	6.9.1 Data- Volume, Velocity, Veracity, Variability and Variety
	6.9.2 Heterogeneity
	6.9.3 Regulation compliance

	6.9.4 Scalability
	6.9.5 Trustworthiness

6.2 IoT system characteristics

6.2.1 Auto-configuration

6.2.1.1 Description

Auto-configuration is the automatic configuration of devices based on the interworking of predefined rules (associated algorithms based on data inputs). Auto-configuration includes automatic networking, automatic service provisioning and plug & play. Auto-configuration allows an IoT system to react on conditions and the addition and removal of components such as devices and networks. Auto-configuration needs security and authentication mechanisms to make sure that only authorised components can be auto-configured into the system. Security and authentication mechanism need to be organized appropriately for each market segment.

6.2.1.2 Relevance to IoT systems

Auto-configuration is useful for IoT systems where there are many and varied components that can change over time and it benefits those users who expect robust systems because auto-configuration can allow automatic elimination of faulty components and maintenance of a working system.

6.2.1.3 Examples

Examples of auto-configuring devices and protocols include DHCP, Zero Configuration Networking (Zeroconf), Bonjour, UPnP etc.

6.2.2 Function and management capability separation

6.2.2.1 Description

Separation of functional and management capabilities means that the functional interfaces and capabilities of an IoT component, such as an IoT device, are cleanly separated from the management interfaces and capabilities of that component. This typically means that the management interface is on a different endpoint from that of the functional interface and the management capabilities are handled by different software components than the functional interfaces.

6.2.2.2 Relevance to IoT systems

Management capabilities and functional capabilities have logically distinctively different

- purposes (execution/action vs information/description),
- user roles (control and modify behaviour vs transfer or consume facts and information),
- classification and types of data (technical or system specific vs personal/sensitive/public),
- access (e.g. an operator may access system configuration, but not gathered personal data; while the user can access the personal data but not access and modify system configuration)

- protocols, formats and lifecycle (e.g. support multiple control protocols vs metadata/structure of the transferred information, which is particularly important considering interoperability and co-existence of multiple versions and variants of management capabilities)

Usually, the differences have associated specific risks and require special security (and other) controls, e.g. retention policy is applicable while dealing with functional data, but might not apply to management data; access control may be weaker for a user and stronger for an administrator).

Ubiquitous penetration of IoT into virtually all areas of life increases the attack surface, multiplying the number of potential attack targets and often making ineffective measures such as physical security controls. The key value of IoT – the connection of numerous edge components to each other and to IoT service components – increases the security concerns, since adding a weak link makes whole chain weak. Applications and systems previously running in well-protected data centers may become exposed to additional threats via connected IoT components.

Separation of management from functional capabilities enables or strengthens the ability to apply different authorization, authentication and protection mechanisms or constraints to management as opposed to functional capabilities. Broad sharing of data from an IoT system might be useful or desirable, and yet there are many circumstances where it is necessary to limit control of an IoT system or component to only a subset of the entities with which the data from that IoT system is shared.

6.2.2.3 Examples

If an IoT system is used to provide sensors and data for HVAC or other building management systems, it might be desirable to share data with other inter-related systems (alarms, access control, power management or auxiliary power, etc.), while still retaining management of the system to ensure system constraints are respected.

6.2.3 Highly distributed systems

6.2.3.1 Description

Distributed systems are the systems which, while being functionally integrated, consists of sub-systems which may be physically separated and remotely located from one another. These sub-systems are normally connected by a communication link (e.g. data bus). (ISO 3511-4)

6.2.3.2 Relevance to IoT systems

IoT systems can span whole buildings, span whole cities, and even span the globe. Wide distribution can also apply to data – which can be stored at the edge of the network or stored centrally. Distribution can also apply to processing – some processing takes place centrally (in cloud services), but processing can take place at the edge of the network, either in the IoT gateways or even within (more capable types of) sensors and actuators. Today there are officially more mobile devices than people in the world. Mobile devices and networks are one of the best known IoT devices and networks.

6.2.3.3 Examples

For industry 4.0, productions can be done using smart manufactory systems which have highly distributed assembly lines across the factories and closely integrated with 3rd party suppliers, logistics companies, market providers and customers etc.

6.2.4 Network communication

6.2.4.1 Description

IoT systems depend on network communications of a number of different types. There are often limited range, low power networks collectively termed proximity networks that form the local connections for IoT devices. There are the wide area networks that connect the proximity networks to the internet, which can take wired and wireless forms and which may be dedicated to the IoT system or which may be shared general purpose networks.

Communication protocols used can vary between the different network types. It is common for proximity networks to use specialized protocols suited to the specialized nature of these networks. IP is more typically used for the wide area networks, although the higher levels in the protocol stack can vary, with HTTP being used in some cases, and messaging protocols being used in other cases. Some networks are deliberately intermittent in nature and the protocols used for such networks reflect the intermittent transmission pattern.

6.2.4.2 Relevance to IoT systems

IoT systems rely on the ability to exchange information units in a structured manner based upon different but interoperable kinds of network types. Devices need to both transmit and receive data and need to communicate with software services that may be located nearby or in a remote location.

Gateways may be employed to connect networks of different types, typically between the proximity networks and the wide area. Network structure may need to be dynamic and needs to consider properties such as QoS, resilience, security and management capabilities.

6.2.4.3 Examples

In a proximity network, IoT devices can be connected by wireless technology, e.g., IEEE 802.15.4 and IEEE 802.11 in communication protocols on physical and data link layers. Data may be transported by 6LowPAN which is IoT specific IP and UDP. The IoT devices are then connected to a dedicated or general purpose wide area network via area a Gateway which routes data between the proximity network and the wide area network as necessary.

6.2.5 Network management and operation

6.2.5.1 Description

IoT systems require network management. The form of network management and operation depend on the network type and network ownership and the type of communication taking place over the network. Management is required during the setting up of a network, including the handling of device identity and addresses, profiles for the usage of the network and the inclusion of dynamic management capabilities. Management of the networks in-service involves control over QoS, dynamic extension of the networks (for new or updated IoT devices), fault handling and security control.

6.2.5.2 Relevance to IoT systems

Some networks are managed as part of the IoT system – particularly the proximity networks connecting the IoT devices. Other networks, particularly the wide area networks, may not be managed as part of the IoT system, since they are general purpose networks often run by other organizations (e.g. mobile phone networks).

IoT network management has to span both kinds of networks and assemble them into a coherent system that can serve the purposes of the IoT system. Where IoT systems make use of third party general purpose communication networks, their management and operational interfaces can be used, where available.

6.2.5.3 Examples

Energy monitoring by smart meters is an example of a context where strict operation and management will be likely, since there is a commercial interest in such an IoT system being free of unauthorized activity. In such a context, all of the IoT devices, communication networks and information processing platforms are managed.

On the other hand, in case of home energy management, it is not necessary that the individual device be managed strictly. The management of the networks and information processing platforms of the vendor's support infrastructure will be done more as a means of selling more devices than as a profit-generating service in itself.

6.2.6 Real time capability

6.2.6.1 Description

Real time capability is pertaining to a system or mode of operation in which computation is performed during the actual time that an external process occurs, in order that the computation results can be used to control, monitor, or respond in a timely manner to the external process. (ISO/IEC/IEEE 24765)

6.2.6.2 Relevance to IoT systems

IoT systems often function in real time; data flows in continually about events in progress and there can be a need to produce timely responses to that stream of events. This may involve stream processing; acting on the event data as it arrives, comparing it against previous events and also against static data in order to react in the most appropriate way.

6.2.6.3 Examples

In process control, process parameters like temperature, flow, or pressure or status of a device are continuously monitored by sensors and instant actions are initiated.

6.2.7 Self-description

6.2.7.1 Description

Self-description is process by which components of an IoT system describe their capabilities in order to inform other IoT components or other IoT systems for the purposes of composition and interoperability. Self-description includes interface specification, the capabilities of the IoT component, what types of devices can be connected to an IoT system, what kinds of service are made available by the IoT system, and the current state of the IoT system.

6.2.7.2 Relevance to IoT systems

Self-description is needed for composability and interoperability for IoT systems and IoT devices. Self-description is of most benefit for those use cases where an IoT system needs to be interconnected with other IoT systems or those use cases where an IoT system benefits from being extended by the addition of new IoT devices.

6.2.7.3 Examples

Example of self-description for an IoT system and protocols: A system which uses Bluetooth in its proximity networks provides device name and supported service list to each other when connecting.

Access points broadcast the SSID. Wi-Fi devices send passwords and MAC addresses to an access point when connecting to it.

6.2.8 Service subscription

6.2.8.1 Description

It is often the case that IoT users subscribe to IoT services made available by IoT service providers. In this case, the IoT service providers make available a subscription process by which the IoT users can subscribe to a particular IoT service. The subscription process can include payments, plus a clear statement of any pre-requisites that apply to the IoT user. It can be the case that the IoT service involves the installation of IoT devices and the installation and configuration of software components – these are typically provided or specified by the IoT service provider.

In some alternative cases, the IoT user can establish their own IoT service, but in this case the IoT user has the burden of acquiring the necessary equipment and software and has the subsequent responsibilities for operating and maintaining the IoT service.

6.2.8.2 Relevance to IoT systems

Some IoT systems are established on the basis of a subscription model where the IoT users pay for their use of the IoT system – in these cases, the IoT service provider must establish clear mechanisms for establishing and maintaining the subscriptions.

6.2.8.3 Examples

An example of a subscription-oriented IoT service is the provision of personal fitness monitoring, where the IoT user must purchase a wearable IoT device that is then connected to an IoT service that monitors their activity and provides analysis and advice on how their activity is helping the user achieve life goals.

6.3 IoT service characteristics

6.3.1 Content-Awareness

6.3.1.1 Description

Content-Awareness is the property of being aware of the information in an IoT component and its associated metadata. Content-Awareness devices and services are able to adapt interfaces, abstract application data, improve information retrieval precision, discover services, and enable appropriate user interactions.

6.3.1.2 Relevance to IoT systems

Content-Awareness facilitates appropriate functional operations, such as data routing, speed of delivery, security capabilities such as encryption, based on factors such as location, quality of service requirements and sensitivity of data.

6.3.1.3 Examples

This capability can be essential in many applications including health services, broadcasting, surveillance systems and emergency services where some types of information or data flows have specific requirements with respect to timeliness, security and privacy.

6.3.2 Context-Awareness (location awareness, time awareness)

6.3.2.1 Description

Context-Awareness is the property of an IoT device, service or system being able to monitor its own operating environment and events within that environment to determine information such as when (time awareness), where (location awareness), or in what order (awareness of sequence of events) one or more observations occurred in the physical world.

6.3.2.2 Relevance to IoT systems

Context-Awareness enables flexible, user-customized and autonomic services based on the related context of IoT components and/or users. Context information is used as the basis for taking actions in response to observations, possibly through the use of sensor information and actuators. To fully utilize an observation and effect an action, the understanding of context is often critical.

6.3.2.3 Examples

An example of location-based services is a system which different services according to the location of a user.

In cases of an emergency like a fire, the arrival of the fire service requires that the doors to a building shall be unlocked. The security policy that governs the door's access can be enhanced with context. The context here is that an emergency situation is currently happening and that the emergency services are in the vicinity. Based on these two contextual inputs the policy could enable the system to unlock the door automatically and provide access without the need for further authorisation.

6.3.3 Timeliness

6.3.3.1 Description

Timeliness is the property of performing an action, function, or service within a specified period of time.

6.3.3.2 Relevance to IoT systems

Because IoT systems act on the physical world, some actions need to occur at certain times. To achieve this, the actions, functions, and services that lead to such events need to happen within specific time constraints. Timeliness in IoT includes not only latency related issues, but other aspects such as jitter, frequency/sampling rate, and phase.

6.3.3.3 Examples

An IoT system for smart meters needs to collect energy consumption data within specific time constraints in order for the grid system to react to demand.

In an industrial manufacturing process, an example is where some sensors are monitoring the quality of items flowing down an automated production line. Any items which are considered below the required quality must be removed from the line. The removal is performed by some actuators that divert the relevant items off the line. To achieve this, there is a strict time limit on commanding the actuators to perform the diversion – all the processing of sensor information and other relevant data must be completed within the time limit. Where IoT entities are part of any kind of control loop, overall processing time for the loop is critical.

6.4 IoT component characteristics

6.4.1 Composability

6.4.1.1 Description

Composability is the ability to combine the discrete IoT components into an IoT system to achieve a set of goals and objectives.

6.4.1.2 Relevance to IoT systems

System integration, interoperability and composability deal with how the functional components are assembled to form a complete IoT system and how the functional components connect to each other and the binding mechanisms which are used (e.g. dynamic or static, agent-based or peer-to-peer). Interoperability and composability are important topics in both the cyber and physical spaces. Composability imposes a stronger requirement than interoperability in that it requires components not only compatible in their interfaces but exchangeable with other components of the same kind that share similar characteristics such as timing behaviours, performance, scalability and security. When a component is replaced by another of the same kind that is composable, the overall system functions and characteristics are unchanged.

6.4.1.3 Examples

An example of composability might be the ability to swap out sensor components from one vendor and replace them with sensor components produced by a different vendor. In this example, there might be two levels of composability.

First would be complete interchangeability of “commodity” functionality, such as an IoT device from Vendor A being fully replaceable with one from Vendor B.

A second level of composability (or possibly interoperability) might be an IoT control that is vendor-specific at the interface between the IoT component and a physical process device being controlled (a valve, motor, switch, pump or fan, for example), but is still fully interchangeable at the interface between the IoT device and the rest of the IoT system. In this sort of example, the IoT device would serve as a kind of “middleware” between the vendor-agnostic IoT infrastructure, and the vendor-specific physical devices or mechanisms being controlled.

6.4.2 Discoverability

6.4.2.1 Description

Discoverability allows users, services, and other devices, to find not only devices on the network but also the capabilities and services they offer at any particular time. Discovery services allow IoT users, services, devices and data from devices to be located, identified, and accessed according to different criteria, such as geographic location, security, safety and privacy.

6.4.2.2 Relevance to IoT systems

Services connected with an IoT system can indicate what information can be found by a Discovery/Lookup service in accordance with predefined rules for each market segment. Discovery/Lookup services allow IoT systems to locate other devices, services or systems based on parameters such as geographical location, capabilities, interfaces, accessibility, ownership, security policy, operational configuration, data provided, data consumed, or other relevant factors.

6.4.2.3 Example

IoT systems which support dynamic configuration, such as the addition of new devices and services to the IoT system, have a requirement for some form of discoverability, since there is a need to identify and characterize new components added to the system. So the addition of a new temperature sensor in a building monitoring IoT system is an example, where it is necessary to bring the new sensor into the existing system with minimum effort. Various protocols and software solutions exist to provide discovery in IoT systems, with a variety of architectures, some server based others being peer-to-peer. Examples include Hypercat, Alljoyn and Consul.

6.4.3 Modularity

6.4.3.1 Description

Modularity is when a component is a distinct unit that can be combined with other components.

6.4.3.2 Relevance to IoT systems

Modularity allows components to be combined in different configurations to form systems as needed. By focusing on standardized interfaces and not specifying the internal workings of each component, implementers have flexibility in the design of components and IoT systems.

6.4.3.3 Examples

An example of Modularity in an IoT system might be a smart thermostat. Because the interface to an HVAC system and the interface to a larger IoT infrastructure could both be defined in compliance with open interface standards, there is nothing to prevent a thermostat from Vendor A being replaced by one from Vendor B. Furthermore, it is not important how the functionality of the device is implemented. Vendor A might provide the capability in the form of an ASIC-based state machine, while Vendor B's design might be based on a microcontroller. As long as both devices perform the same functions in response to the same inputs, and they are both compliant with open standard interfaces without imposing any proprietary constraints, there is nothing to prevent one from being replaced by the other.

6.4.4 Network connectivity

6.4.4.1 Description

In IoT systems, components communicate with each other across network links. The connections between components are established using either wired or wireless media. Networked IoT devices that originate, route and terminate communications are described as (network) nodes. Endpoint network devices are the source or destination of any kind of information. Any IoT related networking communications protocol is layered onto more specific or more general communications protocols, down to the physical layer that directly deals with the transmission media at every network node.

6.4.4.2 Relevance to IoT systems

IoT systems rely on the ability to exchange information in a structured manner based upon multiple different but interworkable network topologies – all within a physical, wired or wireless network. IoT devices are called "networked" when one device is able to exchange information with other devices whether or not they have a direct connection to each other. IoT network structure can be static or dynamic and may have capabilities such as QoS, resilience, encryption, authentication and authorisation.

6.4.4.3 Examples

The Scale of an IoT network can vary substantially, from local proximity networks connecting a handful of devices over a limited distance, to global scale networks operating at Internet scale and connecting very large numbers of devices and service components.

It is typical for the networks in IoT systems to be heterogeneous and connected to each other via gateways or equivalent components.

6.4.5 Shareability

6.4.5.1 Description

Shareability is the ability to share the use of an individual component between multiple interconnected systems.

6.4.5.2 Relevance to IoT systems

Many IoT components are underutilized since a single system often uses only a fraction of a component's capabilities. Resources can be used more efficiently if the functionalities of components can be shared among multiple systems.

6.4.5.3 Examples

The motion detection capabilities of a lighting control system could be leveraged by the security system to increase the security systems capability.

Temperature sensing for heating control could be used by the security system for fire detection.

6.4.6 Unique identification

6.4.6.1 Description

Unique identification is the characteristic of an IoT system to enable the entities to be identifiable and traceable. These entities include the components of the IoT system itself, such as the software components, the sensors and actuators and the network components.

6.4.6.2 Relevance to IoT systems

It is essential that the entities in an IoT system can be distinguished from each other. This enables interoperability and global services across heterogeneous IoT systems. It is important for entities to be uniquely identifiable so that IoT systems can monitor and communicate with specific entities. A variety of identification schemes may be supported in specific implementations of IoT systems to meet the application requirements.

6.4.6.3 Examples

IPv4, IPv6, URI, and Fully Qualified Domain Names (FQDNs) are used as unique, unambiguous identification of network endpoints in internet applications. Individual hardware devices, software etc. may have unique manufacturer's IDs, OIDs, UUIDs or other identifiers, which can be used to tag data from those entities or direct commands to them.

Physical entities are often given unique identifiers in the form of RFID tags, barcodes and equivalent labelling technologies. For humans, biometric information can be used to provide unique identification.

6.5 Compatibility

6.5.1 Legacy support

6.5.1.1 Description

Legacy support is the concept that an IoT system might need to incorporate existing installed components even where these components embody technologies that are no longer standard or approved. A service, a protocol, a device, system, component, technology, or standard that is outdated but which is still in current use, may need to be incorporated into an IoT system.

6.5.1.2 Relevance to IoT systems

Support of legacy component integration and migration can be important, although when supporting legacy components, it is also important to ensure that the design of new components and systems does not unnecessarily limit future system evolution. To prevent prematurely stranding legacy investment, a plan for adaptation and migration of legacy systems is important. Care ought to be taken when integrating legacy components to ensure that security and other essential performance and functional requirements are met. Legacy components may increase risk and vulnerabilities. Since current technology becomes legacy technology in the future it is important to have a process in place for managing legacy aspects of IoT. The different lifecycles of physical systems and information systems also creates additional challenges for managing legacy aspects in IoT.

6.5.1.3 Examples

One example of transition from legacy to future compatibility is the current slow rollover from IPv4 compliance to IPv6 compliance. The limits of the IPv4 address space and of the IPv4 protocol are known, and the transition to IPv6 is clearly the way of the future, but the varying pace of the transition, depending on the context, makes it a topic which can be very complex.

Many existing standards and application environments still assume and depend on IPv4, and yet it's clear that continuing to use IPv4 forever is not a viable strategy. Deciding how and when to make the transition, however, is a topic that nobody has a universal answer to.

The end result is that different market segments, vendors and communities of interest are each pursuing their own strategies for the v4 to v6 transition, and anybody whose activities straddles several of these different transition strategy enclaves has an additional layer of complexity draped over the individual transition strategies.

6.5.2 Well-defined components

6.5.2.1 Description

IoT entities are deemed to be well-defined when an accurate description of their capabilities and characteristics is available, including any associated uncertainties. Capability information includes not only information about the specific component functionality, but configuration, communication, security, reliability and other relevant information.

6.5.2.2 Relevance to IoT systems

Many components are used to assemble an IoT system. They are typically discovered through an information system interface and information about the component may not be available. Without understanding the capabilities of each component that will be used within a system it is difficult to understand whether the system meets its design goals.

6.5.2.3 Examples

An example of an implementation of a well-defined component is: A particular IoT component is available with varying amounts of memory or support for various RF frequencies, waveforms and protocols. Such a device has a baseline information interface which all the variants make use of to inform other IoT components of the list of capabilities possessed by the device. Once the devices' respective configurations have been exchanged, each device's software or applications can then self-adjust to take into account the capabilities of the other devices.

6.6 Usability

6.6.1 Flexibility

6.6.1.1 Description

Flexibility is the capability of an IoT system, service, device or other component to provide a varied range of functionality, depending on need or context.

6.6.1.2 Relevance to IoT systems

History and experience tell us that while there are exceptions, the economic and functional sweet spot for flexibility is usually somewhere in the middle, between the extremes of a dedicated single purpose component on one end of the spectrum, and a massively capable, programmable, extensible, "all things to all people" general purpose component.

It is possible to break down the general concept of flexibility into different sub-categories or dimensions.

One dimension of flexibility is the distinction between IoT capabilities hosted on a platform powered by a general purpose computing core and a similar capability implemented in the form of state machines implemented using discrete components, programmable FPGAs, or a purpose-specific ASIC. The state machine versions tend to be smaller, faster, more power efficient, and potentially more secure (due to a more limited range of capability). The general purpose version trades off speed, size, power draw and other traits to gain more generalized capabilities, and a greater ability to adapt to meet unanticipated future requirements.

A second dimension of flexibility is illustrated by the distinction between the following kinds of device:

1) A device which has fixed, nonprogrammable, non-extensible functionality – "hard wired, single purpose".

2) A device which has fixed H/W capability, but which provides some amount of configurability within the single available format.

3) A device which is both programmable and expandable in the hardware domain – such as adding memory, adding more computational capability or adding RF channel capability.

4) A family of devices, each of which might fall into categories 1-3, from which an integrator can select the one(s) which are appropriate for a given context.

5) A family of devices such as in 4, where some of the options provide different amounts of composability or modularity, at different levels of abstraction.

A third dimension of flexibility might involve the range of standards, protocols, formats, and interfaces which an IoT component is designed to support, where that support might then be designed and implemented taking the factors above into account.

Aside from the IoT component, there is another dimension of flexibility that involves the overall design of the IoT system. As in other domains, there will likely be open IoT ecosystems, and proprietary IoT ecosystems, with varying amounts of overlap between the two.

6.6.1.3 Examples

An example of differences flexibility relating to a sensor device is such as a thermostat. The simplest devices may only offer simple temperature control and reporting of temperature. More sophisticated and flexible thermostats allow for remote control via smartphone, can be connected to other IoT devices in the building to detect occupancy, to gain information about the weather and so on – and these more capable devices typically have software components that can themselves be upgraded to offer newer capabilities.

6.6.2 Manageability

6.6.2.1 Description

Manageability addresses aspects of IoT systems such as device management, network management, system management, and interface maintenance and alerts. Manageability is important to meet IoT system requirements. Components capable of monitoring the system and changing configurations are needed for manageability of the IoT device, network and system.

6.6.2.2 Relevance to IoT systems

Many IoT devices, networks, and systems are unmanned and run automatically. Special care must be taken to ensure that such systems remain manageable even when parts of the system malfunction, become unstable or mis-calibrated in the course of operation. Even in circumstances where individual IoT entities are accessible, the potentially large scale and geographic span of IoT systems argues for the ability to manage IoT entities remotely to the greatest extent possible, to increase both convenience and operational effectiveness.

6.6.2.3 Examples

IoT devices such as smoke sensors are deployed in various locations in buildings. These devices are often hard to maintain because of their locations. Any type of malfunction could cause undesirable events and consequences. Thus, remote manageability should be a system design consideration and goal from the beginning of specification, and throughout the development, and deployment, and operational lifecycle of the IoT system.

Additionally, software updates are necessary to ensure that devices and systems maintain functionality and the latest security vulnerabilities are patched. The manageability capabilities of an IoT entity might include device state monitoring capability, the link monitoring, calibration, etc.

6.7 Robustness

6.7.1 Accuracy

6.7.1.1 Description

In the context of reliability, accuracy is the capability of an IoT device, service or system to provide calculations or actions within the expected range of acceptable precision.

6.7.1.2 Relevance to IoT systems

An appropriate level of accuracy is essential to some IoT system deployments and applications. Depending on the context, differing degrees of accuracy might be required.

6.7.1.3 Examples

In a medical or manufacturing context, it might be critical for an IoT Device, application or system providing temperature information or control to be accurate to within a tenth of a degree Fahrenheit, while in a home HVAC context, accuracy to plus or minus two degrees might be adequate.

6.7.2 Reliability

6.7.2.1 Description

Reliability is the consistent intended behaviour of a system. An appropriate level of reliability in capabilities such as communication, service and data management capabilities is important to meet system requirements.

6.7.2.2 Relevance to IoT systems

An appropriate level of reliability is essential in diverse IoT system deployments and applications. Reliability can be highly critical in some applications, e.g. for specific health related applications, industrial manufacturing operations and time-critical applications.

6.7.2.3 Examples

Reliability of data is of great importance for the decision-making processes of many IoT systems. The absence of data or data corruption can lead to incorrect decisions or the failure to make decisions. Reliability of communications networks is important for ensuring the availability and correct operation of IoT systems, particularly in mission-critical use cases.

Medical devices are one potential IoT application area where the specifications for mean time between failure might be quite stringent, due to the possibility of injury or death if an IoT device, application or system providing medical capability were to fail while a patient is being treated.

6.7.3 Resilience

6.7.3.1 Description

Resilience is the ability of an IoT system or its components to continue to perform their required function in the presence of faults and failures.

6.7.3.2 Relevance to IoT systems

Communication, device or software component failures are to be expected in IoT systems and without appropriate design, they can escalate quickly causing the global failure of the system. IoT systems need to be designed for resilience, incorporating self-monitoring and self-healing techniques to improve the system resilience.

6.7.3.3 Examples

An IoT system has to be resilient to gateway failures to ensure continuing communications paths between software components and IoT devices.

One approach to resiliency is to adopt a master-slave design where if the master unit fails then a redundant device is available to assume the master role.

For networks, a mesh network design is resilient to the failure of one link or one node - data can still flow from source to sink through an alternative route.

6.8 Security

6.8.1 Availability

6.8.1.1 Description

Availability is the ability of a system to be accessible and usable on demand by an authorized entity. IoT systems can include both human users and service components as "authorized entities".

6.8.1.2 Relevance to IoT systems

In IoT systems, availability can be seen in terms of devices, data and services. Availability of a device is related both to its inherent properties of operating correctly over time and to the network connectivity of the device. Availability of data is related to the ability of the system to get the requested data from a system component. Availability of services is related to the ability of the system to provide the requested service to users with a pre-defined QoS.

6.8.1.3 Examples

In some critical applications, e.g. health monitoring or intrusion detection, devices and data have to be always available so that alarms can be sent to the system immediately when raised. In these cases, system design must take into account potential failure modes and provide means of continuing operations, such as power supply backups, redundant devices, multiple instances of a service.

6.8.2 Confidentiality

6.8.2.1 Description

Confidentiality is the property, that information is not made available or disclosed to unauthorized individuals, entities, or processes.

6.8.2.2 Relevance to IoT systems

In an IoT system the confidentiality protection is responsible for prohibiting people or systems from reading data or control messages when they are not authorized to do so.

Confidentiality is a pre-requisite for a secure operation especially when the data to be transmitted contains secret tokens, e.g. for access control. Confidentiality is also required to protect sensitive data, which may include financial information or personal data (see the clause on Privacy).

6.8.2.3 Examples

Many items of data flowing an IoT system need to be treated as confidential – in the hands of the wrong recipient the data could be used for criminal acts or represent inappropriate use of personal data. For example, IoT motion detection sensors could reveal whether a property is occupied or not – which could be used by thieves to target the property.

Similar concerns relate to IoT smart meters – where even the frequency of messages transmitted should not depend on the rate of electricity use, since this could reveal whether a property is occupied or not.

6.8.3 Integrity

6.8.3.1 Description

Data integrity is the property that data has not been altered or destroyed in an unauthorized and undetected manner. [ISO_19790:2012, 3.58] Given that data is the basis on which IoT systems operate, tampering or destruction of data flowing or stored in the system could compromise the operation of the system and lead to highly undesirable outcomes.

6.8.3.2 Relevance to IoT systems

Data integrity is vital for IoT systems to ensure that the data used for decision-making processes in the system and executable software has not been altered by faulty or unauthorized devices or by malicious actors. The protection of the integrity of the data is a key requirement to ensure the security of the IoT system.

6.8.3.3 Examples

In IoT deployments that comprise of multi-hop wireless sensor networks there is a risk that intermediate nodes may alter the data and this can have impact on the functioning of the system. For example, an intermediate node may increase the value of the temperature of a room but this should not cause the air-conditioning system to increase the amount of cooling.

6.8.4 Safety

6.8.4.1 Description

Safety is the freedom from risk which is not tolerable. Risk is the combination of probability of occurrence of harm and the severity of that harm. Harm includes injury or damage to the health of people, or damage to property or the environment. Harm can be due to malfunction, failure, or accident. While prior traits describe the desired behaviour of the system when operating correctly, Safety includes the consideration of failure modes with the intent of preventing, reducing or mitigating the potential for undesired outcomes; specifically, damage, harm or loss.

6.8.4.2 Relevance to IoT systems

Many IoT systems are deployed in contexts or operational environments where damage, loss, injury or death might result if failure modes are not adequately addressed. In many operational contexts, approval to operate or approval to connect will not be granted if safety requirements are not met.

Even in contexts where compliance with safety standards is optional or voluntary rather than mandatory, proper consideration of safety factors may have significant impact on aspects such as: continuity of operations, reduction of loss, prevention of injury or death, insurance premiums, torts and liability, and other issues.

6.8.4.3 Examples

IoT contexts where safety standards or requirements might need to be considered include medical or health care applications, transport such as aviation and automotive applications, consumer products, buildings, and environment monitoring. Many countries will have specific regulations related to such applications.

6.9 Protection of personally identifiable information

6.9.1 Description

Protection of personally identifiable information (PII) is a legal or regulatory requirement in most jurisdictions whenever an IoT system involves personally identifiable information anywhere in its operation.

Privacy is the right of individuals to control or influence what information related to them may be collected and stored and by whom and to whom that information may be disclosed. (Based on ISO/TS 17574:2009, 3.16) The concept of privacy overlaps, but does not coincide, with the concept of data. With respect to data protection it ensures that PII is not gathered or processed without the informed consent of the PII principal, and is not disclosed to unauthorized entities. For IoT systems, entities include both people, machines and processes.

The principle of data minimisation applies to PII: the quantity of PII collected is the minimal necessary to support the application. The PII which is necessarily present should be securely deleted when no longer needed. This protects the individual and minimizes legal risk to the organization using the PII. If PII is disclosed it must be based on prior informed consent given by the PII principal for the intended purpose.

6.9.2 Relevance to IoT systems

Many IoT systems do not collect or interchange PII. However, any IoT system which does collect, receive and/or interchange personal information needs to ensure that such IoT systems and their interactions with other IoT systems (or IT systems in general) are in full compliance with privacy protection requirements of applicable jurisdictional domains.

One aspect of IoT systems is that the nature of the data handled by the system can be unclear. For example, a home automation IoT system may appear to be dealing in data that is not PII, but if (say) the electrical usage data of a house is present in the system, if the data can be connected with a specific house, it is likely that the data is connected with specific people and can be regarded as PII.

IoT systems need careful analysis to understand if any of the data they handle is or is potentially PII. If PII is present, then the IoT system must be designed to meet appropriate data protection regulations and laws in the relevant jurisdiction(s).

6.9.3 Examples

Many IoT applications involve end-users and the collection of specific data relating to them. For example, traffic speed cameras record a number plate and often an image of the driver's face. This information is correlated with licensing records to allow fines to be levied. However, such data cannot be retained beyond a pre-defined time and should not be made available for other purposes. Mobile phone location can be tracked and while this can be useful for a user to receive information about facilities in the area, access to such information should be controlled; it may be required for police investigations but users may not want to receive adverts for local venues.

In particular, with healthcare monitoring and other such monitoring of specific individuals there is a need for the data to be provided only for the agreed purpose for example to update a GP or personal healthcare record and not for use by other institutions such as insurance companies.

Driver may be providing data for traffic monitoring systems (location and speed) allowing traffic congestion to be reported but would not necessarily expect this data to be linked to an employer's system. Similarly, tracking people movement in offices may be possible with a building surveillance and access system but noting times of rest breaks etc., may not have been part of the purpose of the data collection.

Smart metering applications are another example where an individual may grant access to data for a particular purpose. The smart meter is collecting real-time information about electricity usage in the home and transmitting it to the electricity utility, who may use the data for a variety of purposes, including demand management and differential pricing. It is clear that the data relates to the people living in that home and may reveal significant details about their lives. It is necessary for the electricity utility organization to inform the householder about the PII they are gathering and to be clear about its use. The electricity utility also needs to apply appropriate protection to the data (e.g. encrypt data streams flowing from the smart meter), and apply privacy principles to the processing of the data, including minimising the data, anonymizing the data and deleting the data as soon as possible.

Several governments and group of states has issues laws based on the European directive 2016/680 on Privacy, especially to protect citizens from the increasing exposures of their PII in the daily digital life on the net. A set of laws to be implemented to provide rules for the business how to handle PII and make it certain that PII is not exposed more than the business relation requires.

6.10 Other characteristics

6.10.1 Data– Volume, Velocity, Veracity, Variability and Variety

6.10.1.1 Description

The "Data 5Vs" of volume, velocity, veracity, variability and variety often apply to IoT systems. The Data 5Vs derive from Big Data systems – but it is often the case that IoT systems are the source of data which is large in volume, delivered at speed across network links, whose veracity needs to be validated (e.g. due to malfunctioning sensors), which can vary over time and can contain a wide variety of different data types from different IoT components.

6.10.1.2 Relevance to IoT systems

IoT Systems are also expected to generate large amounts of data from diverse locations. The data may be aggregated into centralized locations or it may be stored in distributed locations (depending on the nature of the data, the processing required on the data and the communication link characteristics), which generates a need to appropriately index, store and process the data.

6.10.1.3 Examples

A logistics company uses big data analytics for an On-Road Integrated Optimization and Navigation service. The system uses numerous address data points, plus other data collected during deliveries, to optimize delivery routes.

6.10.2 Heterogeneity

6.10.2.1 Description

An IoT system typically is composed of a diverse set of components and physical entities that interact in various ways.

6.10.2.2 Relevance to IoT systems

IoT is typically cross-system, cross-product, and cross-domain. Realizing the full potential of IoT requires interoperability between heterogeneous components and systems. This heterogeneity creates numerous challenges for the resulting IoT systems.

6.10.2.3 Examples

A smart container using RFID tags for identity and related RFID sensors needs interworking of RFID systems and sensor network systems.

6.10.3 Regulation compliance

6.10.3.1 Description

IoT systems, services, components and applications can be deployed in circumstances which require adherence to a variety of laws, policies or regulations. Such support might be inherent in the IoT device or system, or might require specific configuration, programming, modification or extension to ensure compliance.

Additionally, there might be a range of different granularity or levels of abstraction at which the regulations are applied or enforced.

6.10.3.2 Relevance to IoT systems

Regulations of relevance to IoT systems might take many forms, including regulations to assure interoperability, to mandate or constrain functionality or capability, to assess the ability of the IoT device or system to function in a certain usage context without causing damage, and to impose at least minimal balance between contribution to the collective good and self-interest on the part of system owners or operators.

6.10.3.3 Examples

Regulations which might apply to an IoT context include one or more of the following categories:

1) Safety regulations – These might include flight safety standards for IoT devices operating in aircraft, or regulations covering the manufacture and sale of devices intended for consumer use in the home, regulations for automotive systems, or regulations for devices or systems used in a medical context.

2) RF related regulations – This category might include national or international regulations governing RF emanations, adherence to frequency band restrictions, signal strength, spurious signals (such as side channels, noise, or harmonics produced outside of the device's nominal frequency allocation), etc.

3) Consumer protection regulations– These might include national and international regulations invoked whenever an IoT system involves a consumer anywhere in its operation.

In some IoT contexts, such as home automation, HVAC, etc. another layer of regulations might be imposed in the form of building codes in various jurisdictions.

While the area is still developing, it is quite possible that at some point, there will be regulations imposed or referenced by insurance companies as part of their risk models for pricing coverage of structures, vehicles, systems, or businesses incorporating IoT systems and devices.

6.10.4 Scalability

6.10.4.1 Description

Scalability is the characteristic of a system to continue to work effectively as the size of the system, its complexity or the volume of work performed by the system is increased.

6.10.4.2 Relevance to IoT systems

IoT systems involve various elements such as devices, networks, services, applications, users, stored data, data traffic, event reports. The amount of each of these elements can change over time and it is important that the IoT system continues to function effectively when the amounts increase.

6.10.4.3 Examples

One example of scalability is when the number of sensor devices attached to an IoT system is increased. If a system changes from monitoring temperature sensors in a single building to monitoring temperature sensors on all buildings in a city there will be a significant increase in the volume of sensor data flowing in the system, in the volume of data being stored in databases, in the number of devices handled by the management system, and in the number of temperature readings processed by services and applications.

6.10.5 Trustworthiness

6.10.5.1 Description

Trustworthiness is the degree to which a user or other stakeholder has confidence that a product or system will behave as intended.

6.10.5.2 Relevance to IoT systems

Device, data and service trustworthiness is of utmost importance for IoT systems to ensure that only trusted devices participate in the decision-making process of the system, resulting in the provision of trustworthy applications. Device executable processes and data must be trusted to ensure that the device/system operates as intended.

6.10.5.3 Examples

Where an IoT system that monitors the average measurement of a room taking the mean value reported by x sensors, if y sensors report false values, due to a fault or malicious programming the resulting mean measurement will be false. Detection, assessment and potential exclusion of anomalous readings is necessary to ensure trustworthy data.

7 IoT CM

7.1 Main purpose

CM provides a common structure and definitions for describing the concepts of, and relationships among, the entities within IoT systems. It must be generic, abstract and simple. In order to achieve this goal, it is important to clarify the fundamentals of the IoT systems by asking the following questions:

1) What is the overall model of IoT entities and their relationships?

2) What are the key concepts in a typical IoT system?

3) What are the relationships between the entities, especially between digital entities and their physical entities?

4) Who and where are the actors?

5) How the things and services collaborate via the network?

The following clauses describe the CM focusing on the above five points. The models presented here use simplified Unified Modelling Language™ (UML®, hereafter “UML”). A short description of the simplified UML[1] in order to help readers to better understand CM diagrams can be found in Annex A.

7.2 Overall model

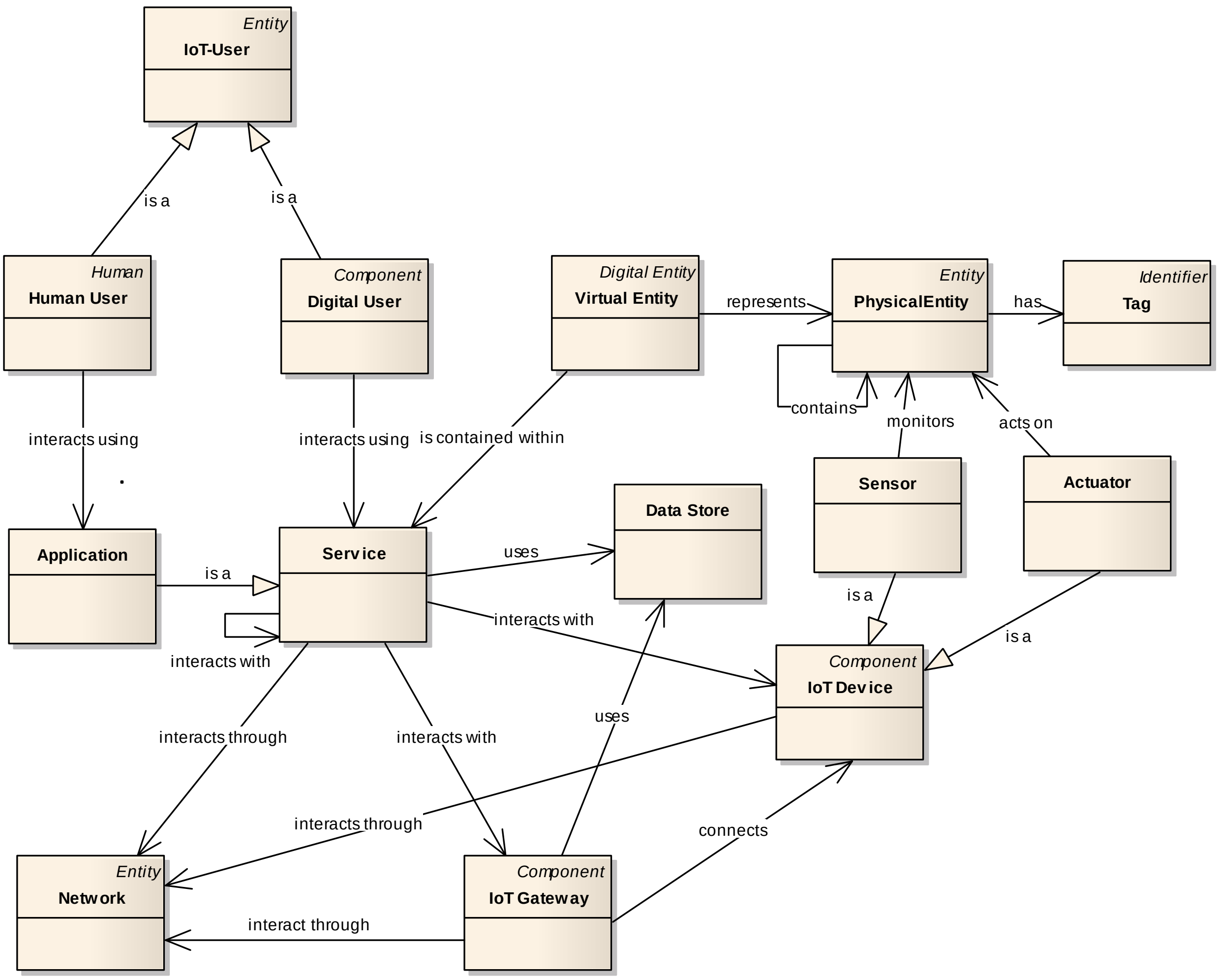

Figure 4 – Overall model for IoT concepts of the CM

Figure 4 provides the overall model of all key IoT entities defined in this CM, their relationships and their interactions. The IoT-User can be human (human user) or non-human (digital user) such as robots or automation services, which act on behalf of human users. Digital user consumes services which are interact through the communication network. A human user interacts using applications, which are a specialized form of service. Some applications interact with other services via the network.

Physical entity here is the real-world thing which is controlled by an actuator or monitored by a sensor. The physical entity may have an attached tag which is monitored by a sensor, rather than the physical entity itself. A virtual entity represents a physical entity in the IT world. Both actuators and sensors are kinds of IoT device. IoT devices interact through a network and can either communicate widely directly or are connected with an IoT gateway which is capable of communicating widely.

[1] ISO/IEC 19501:2005(en) Information technology — Open Distributed Processing — Unified Modelling Language (UML)

Data Stores hold data relating to IoT systems, which may be data directly derived from IoT devices or may be data resulting from services acting on IoT device data.

7.3 Concept

7.3.1 IoT entities and domains

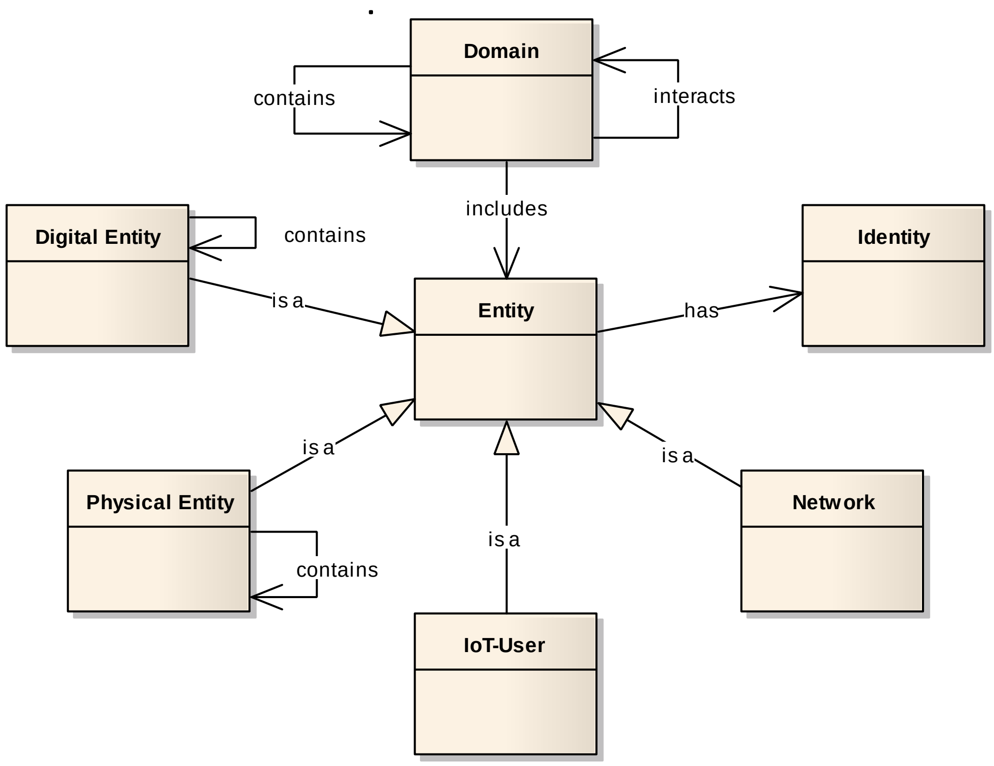

Figure 5 – Entity and domain concepts of the CM

Figure 5 shows entity and domain concepts of the CM. A thing with distinct and independent existence is called an entity, for example, a person, an organization, a device, a subsystem, or a group of such items. Everything in an IoT system is a kind of entity. In order to have a simple concept about IoT entities and their relationship, four fundamental entities are defined here, the thing (Physical Entity), the user (IoT-User), IT systems (Digital Entity) and the communication networks (Network).

A digital entity is one of the computational and data elements of an IoT system, which includes applications, services, virtual entities, data stores, IoT devices and IoT gateways. An IoT-User is an entity which can be human or non-human, while a physical entity is discrete, identifiable and observable. A network is another important entity in the IoT system, through which other entities communicate with each other. Entities have an identity with an associated identifier, and identifiers are one way for a digital entity to get into communication with other digital entities through the network. There are many forms of identifier, which can vary depending on the nature of the entity.

When considering IoT systems, there is a need to decompose the system into smaller parts and group the elements with similar or common characteristics into what is termed a specific *Domain*. Each domain has its own boundary. Showing interaction between domains instead of between all the entities in a system can provide a simpler high level view of how the complex system works. Figure 6 shows that one IoT domain A interacts with another IoT domain B. Of course, one IoT domain can also interact with multiple IoT domains.

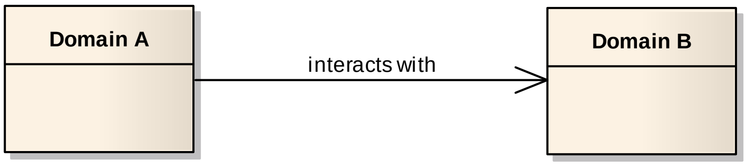

Figure 6 – Domain interactions of the CM

Domains are composed of various types of entity, sometimes one large domain can be segmented into more sub-domains. Figure 7 shows that Domain A contains two sub domains, Domain C and Domain D.

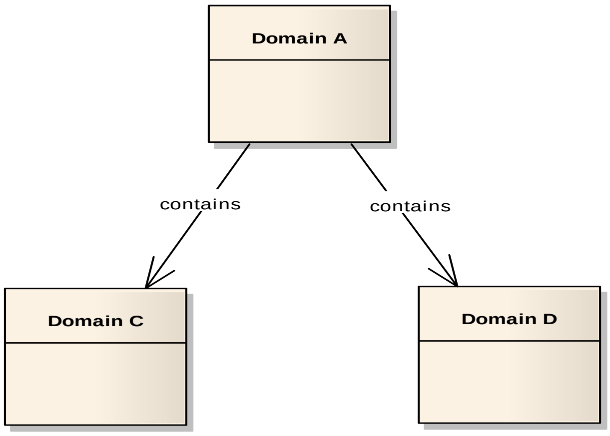

Figure 7 – Domain composition of the CM

The following sub-clauses contain tables depicting the associations shown in the above diagrams. To avoid duplication in the description of relationships between two entities, only entities with outgoing relationships will be described.

7.3.1.1 Entity

An entity is anything (both physical and non-physical) which has a distinct and independent existence. Every entity has a unique identity.

7.3.1.2 Domain

A domain is group of entities with similar or common characteristics or activities. A domain includes one or more entities. A domain may contain sub domains. A domain may interact with other domains.

7.3.1.3 Digital entity

A digital entity is a computational or data element of an IoT system. These elements include applications, services, virtual entities, data stores, IoT devices and IoT gateways. A digital entity is a specialization of entity. A digital entity may contain other digital entities.

7.3.1.4 Physical entity

A physical entity is a real-world thing which is controlled by an actuator and/or monitored by a sensor. A physical entity is a specialization of entity. A physical entity may contain other physical entities.

7.3.1.5 IoT-User

An IoT-User is a user of an IoT system, which can be human or non-human. An IoT-User is a specialization of entity representing a human user or digital user.

7.3.1.6 Network

A network is infrastructure that connects a set of digital entities, enabling communication of data between them. A network is a specialization of entity.

7.3.2 Identity

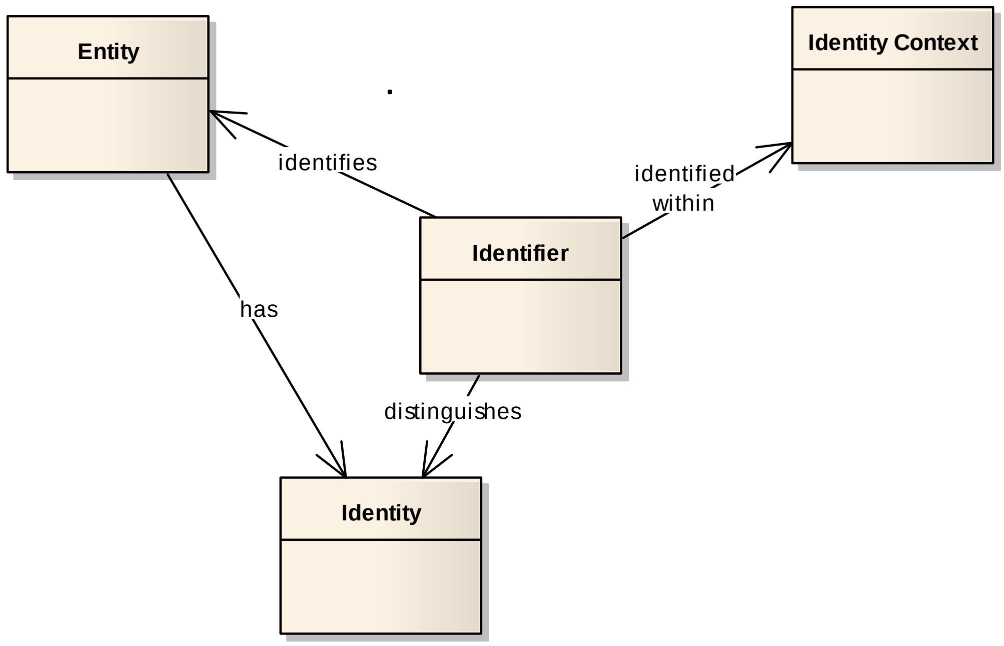

Figure 8 – Identity concept of the CM

Figure 8 shows the identity concept in relation to Entities. Most entities in IoT especially physical entity (“Thing”) have an identity. An identifier is a dedicated, publicly known attribute or name for an identity. Typically, identifiers are valid within a specific context. An entity can have more than one identifier, but it requires at least one unique identifier within any identity context through which it can be accessed. For example, the identity information from a tag can be used as an Identifier to identify the physical entity to which it is attached.

7.3.2.1 Identifier

An identifier is a unique publicly known attribute or name for the identity of an entity, typically valid and unique within a specific context. Identifier identifies entity. Identifier distinguishes identity. Identity may have more than one identifier. Identifier identified within a given identity context.

7.3.3 Services, network, IoT device and IoT gateway

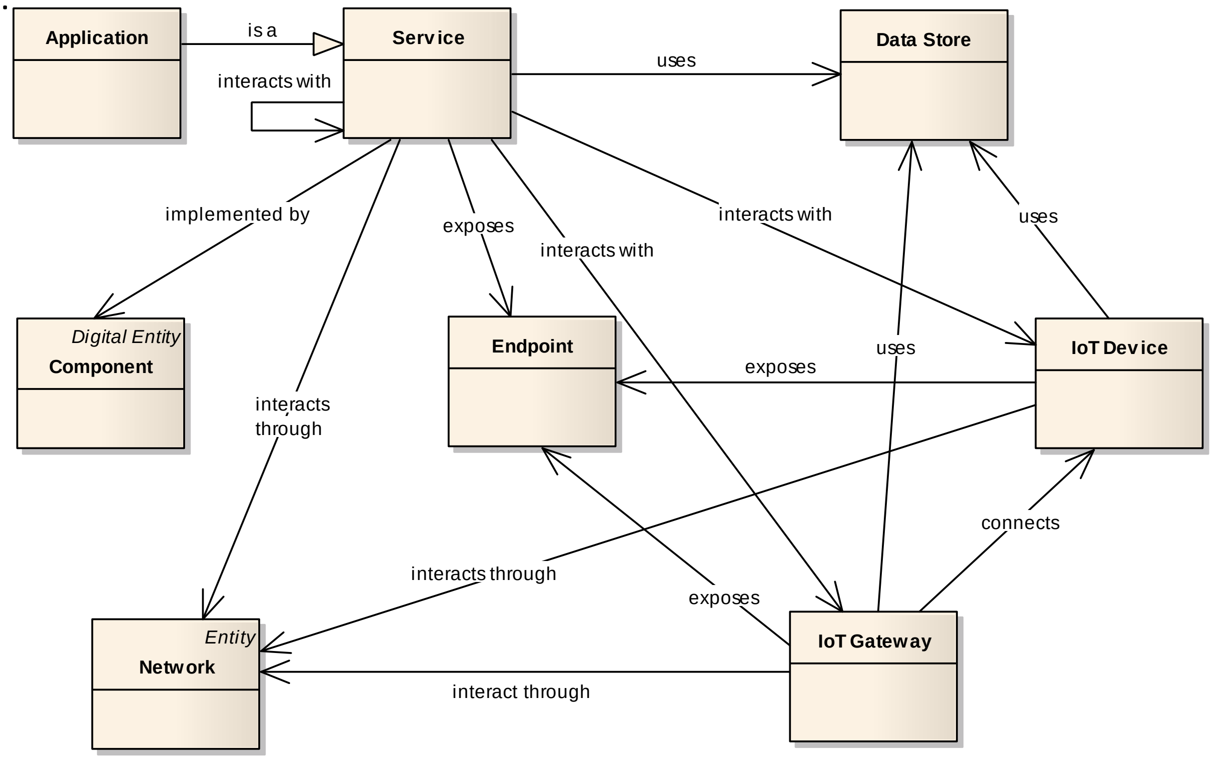

Figure 9 – Service, network, IoT device and IoT gateway concepts of the CM

Figure 9 shows how services, IoT devices and IoT gateways are connected through network. Service is an abstract concept. A service is implemented by one or more components. There could be multiple alternative implementations of the same service.

Entities which interact via networks do so by exposing one or more endpoints on a network A network connects endpoints. A service exposes zero or more endpoints by which it can be invoked. An endpoint has one or more network interfaces. Services, which are located remotely, can be reached by endpoints through network interfaces across a communication network.

Data associated with services, with IoT device and with IoT gateway can be held in a data store used by that entity.

7.3.3.1 Endpoint

An endpoint is one of two components that either implements and exposes an interface to other components or uses the interface of another component. An endpoint may contain more than one network interface.

7.3.3.2 IoT gateway

An IoT gateway is a digital entity that acts as a means to connect one or more IoT devices to a wide-area network. IoT gateway interacts through network. IoT gateway exposes endpoint. IoT gateway connects IoT device. IoT gateway uses data store.

7.3.3.3 IoT device

An IoT device is a digital entity which bridges between real-world physical entities and the other digital entities of an IoT system. IoT device interacts one or more networks through which interactions are made with other entities. IoT device exposes one or more endpoints by which interactions are made. IoT device uses zero or more data stores used by it.

7.3.3.4 Service

A service is a set of distinct capabilities provided by a software component through a defined interface, which may be composed of other services. A service is implemented by one or more components. A service defines network interfaces and exposed by an Endpoint. A service interacts with other entities via one or more Networks. A service interacts with zero or more IoT gateways. A service interacts with zero or more IoT devices. A service interacts with zero or more other services. Zero or more data stores are used by the service.

7.3.4 IoT-User

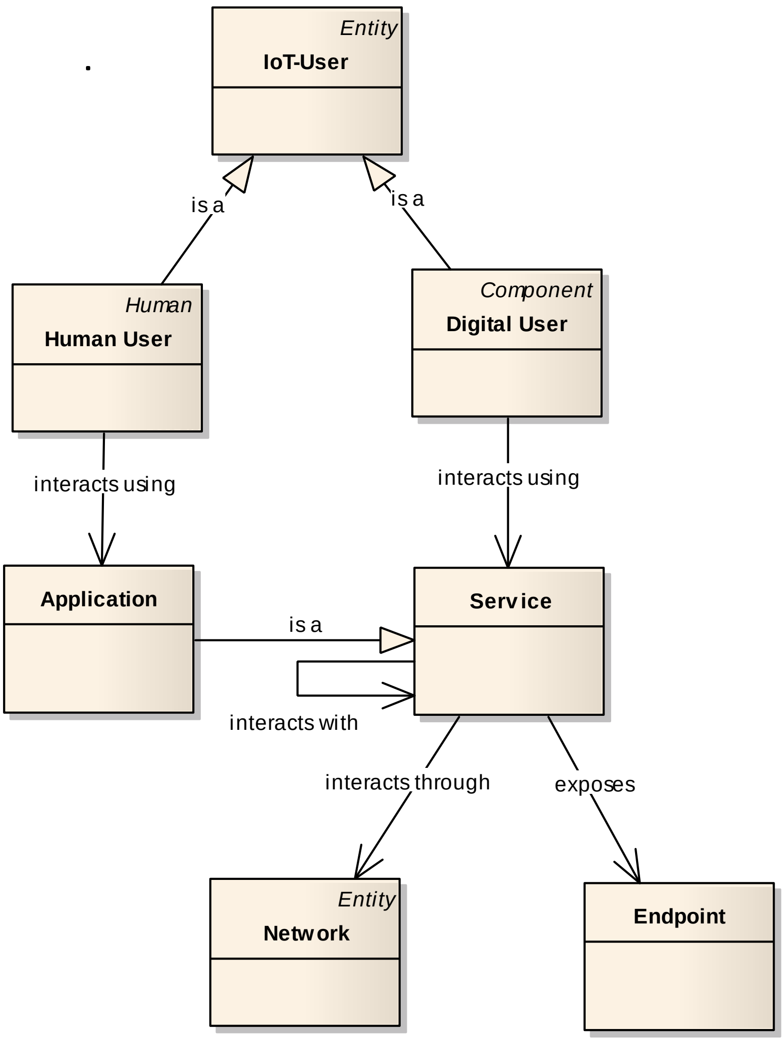

Figure 10 – IoT-User concepts of the CM

As shown in Figure 10, actors of IoT systems are IoT-Users. An IoT-User can be either human (Human User) or digital component (Digital User). A digital user includes automation services that act on behalf of human users, for example in machine to machine interactions. A digital user interacts with one or more services directly or indirectly through its endpoint. A human user interacts through one or more applications. An application is a specialized form of service and can interact with other services.

7.3.4.1 Human user

A human user is person who uses an IoT system. A human user is a specialization of an IoT-User. A human user interacts across the network via an application.

7.3.4.2 Digital user

A digital user is a digital entity which uses an IoT system. A digital user is a specialization of an IoT-User. A digital user interacts with one or more services offered by the IoT system across the network.

7.3.4.3 Application

An application is a software component that offers a collection of functions with which a user can perform a task. An application is a service.

7.3.5 Virtual entity, physical entity and IoT device

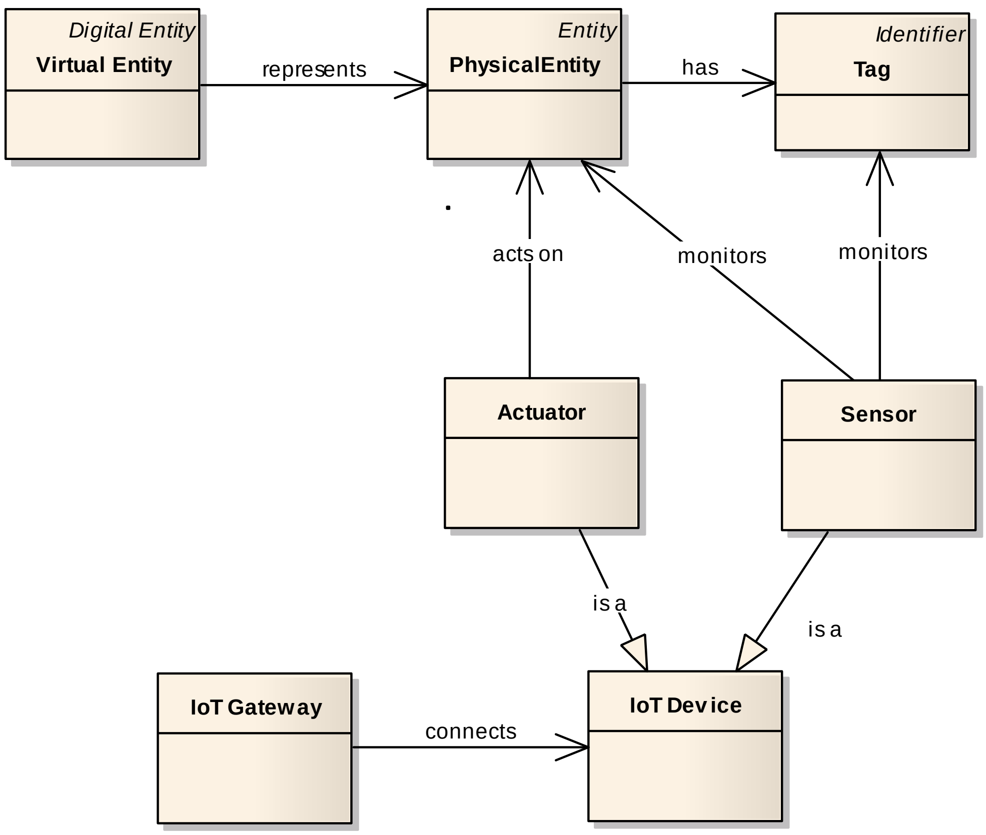

Figure 11 – Virtual entity, physical entity, and IoT device concepts of the CM

Figure 11 shows the relationship between virtual entity, physical entity and IoT device. Actuator and sensor are IoT devices which have direct or indirect contact with a physical entity. An actuator operates on received digital information to act on (change) some property of a physical entity. A sensor perceives certain characteristics of a physical entity and transforms them into a digital representation which can be communicated. A physical entity may have one or more Tags attached to it and sensors can monitor the tag rather than the physical entity itself. Actuator and sensor are kinds of IoT device, which converts variations in one physical quantity, quantitatively into variations in another.

A smartphone, for example, can have a sensor to detect temperature of its surroundings. Another example is where a Bluetooth app on a smartphone communicates with an air conditioner to control the room temperature; the air conditioner is an actuator in this case.

Another example is where a smartphone has a barcode reading application – the application may have a locally installed database (local data store) to lookup the barcode information of a scanned object, or it might communicate with a remote service hosting a catalogue via the mobile network. The barcode itself is one form of a tag attached to a physical object.

7.3.5.1 Sensor

A sensor is a device that detects and responds to some type of input from the physical environment and outputs digital data that can be transmitted over a network. A sensor is a specialization of an IoT device. A sensor monitors a physical entity.

For IoT Device, see Clause 7.3.3.3.

7.3.5.2 Actuator

An actuator is a device that accepts digital inputs and which acts on (changes) one or more properties of a physical entity on the basis of those inputs. An actuator is a specialization of an IoT device. An actuator acts on a physical entity.

For IoT device, see Clause 7.3.3.3.

7.3.5.3 Virtual Entity

A virtual entity is a digital representation of a physical entity, contained within a service. A virtual entity interacts through an endpoint. A virtual entity represents a physical entity.

8 IoT RM and RA views

8.1 Relation between CM, RMs and RAs

A RM is an abstract framework for understanding significant relationships among the entities of an environment, and for the development of consistent standards or specifications supporting that environment. A RM is based on a small number of unifying concepts and can be used as a basis for education and explaining standards to a non-specialist. A RM is not directly tied to any standards, technologies or other concrete implementation details, but it does provide common semantics that can be used unambiguously across and between different implementations [SOURCE: OASIS SOA RM technical committee [1]].

There are a number of concepts rolled up into that of a RM. The RM is abstract, and it provides information about environments of a certain kind.

A RM describes the type or kind of entities that may occur in such an environment, not the particular entities that actually do occur in a specific environment. A RM describes both types of entities or domains and their relationships. A list of entities, by itself, doesn't provide enough information to serve as a RM. A RM does not describe all entities in the framework; it can be used to clarify a specific instance.

To be useful, a RM includes a clear description of the problem that it solves, and the concerns of the stakeholders who need the problem to be solved. A RM typically is intended and is technology agnostic; A RM does not make assumptions about the technology or platforms in place in a particular computing environment. A RM, typically, is intended to promote understanding of a class of problems, not to provide specific solutions for those problems. With respect to this, a RM aids the process of inventing and evaluating a variety of potential solutions in order to assist the practitioner.

A RM is useful to: create standards for both the objects that inhabit the model and their relationships to one another; educate stakeholders; improve communication between people; create clear roles and responsibilities; and allow the comparison of different entities.

The RA can be understood as contexts provided with common features, vocabulary, and requirements, together with supporting artefacts to enable their use. The artefacts are the description of the major foundational architecture components, which provide guidelines and constraints for instantiating solution architectures. The solution architectures can be defined not only from different viewpoints but also at many different levels of detail and abstraction; they consist of a list of entities and functions and some indication of the connections, interrelations and interactions with each other and with functions located outside of predefined architecture patterns representing the entities and functions. Figure 12 shows the architecture continuum from the CM through the entity-based RM and domain-based RM to a number of different views of the RA. The consistent architecture continuum should be maintained not only in this hierarchy (e.g., CM → RM → RA) but also in evolutionary updates over time; the architecture descriptions should be clearly documented.

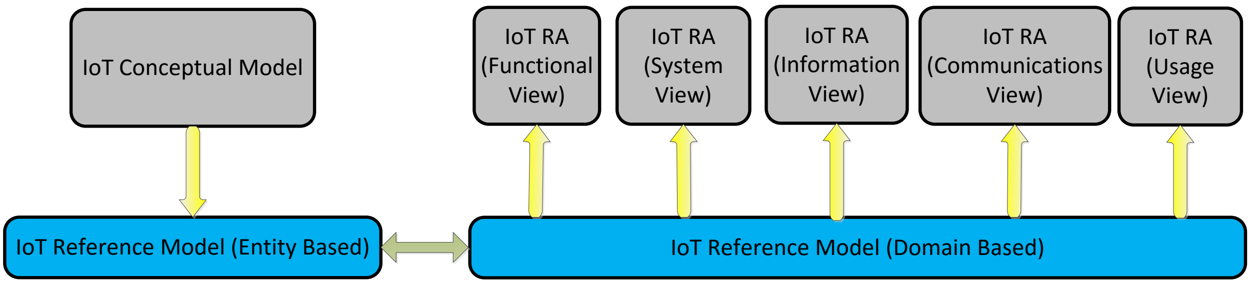

Figure 12 – Relation between IoT CM, RM, and RA

Domains of IoT systems are identified by focusing on the IoT systems' stakeholders and hardware, software, and using common and representative domains provides an effective and representative RM of the IoT systems for the various purposes and uses of the RM.

8.2 IoT RMs

8.2.1 Entity-based RM

Based on the previous high level IoT CM, a composite entity-based RM of IoT systems is described in this clause. The entity-based RM of IoT systems is shown in Figure 13. This figure illustrates the interactions between the major entities using arrowhead lines.

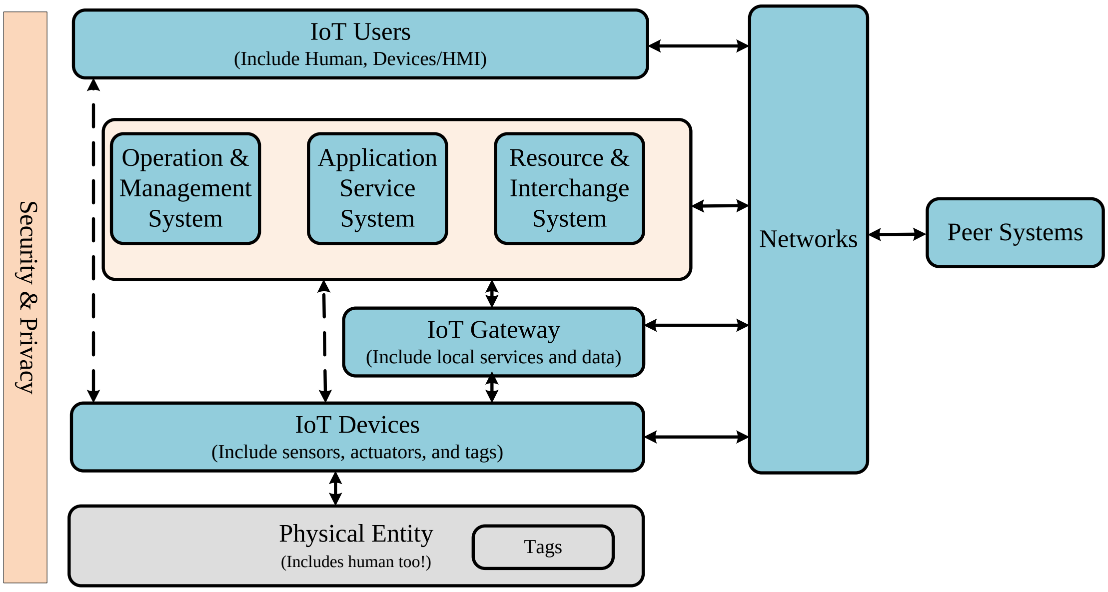

Figure 13 – Entity-based IoT RM

Starting the description of the IoT entity-based RM from the entities at the bottom of the diagram:

1) Physical entities are the real-world things that are the essential part of an IoT system

2) Tags of various types can be attached to physical entities to aid in their monitoring and identification

3) IoT devices connect the physical entities to the IoT system. IoT devices consist of:

   a) sensors, which monitor or scan the physical entities to retrieve some information about them

   b) actuators, which act on or change some properties of the physical entities based on digital instructions

4) IoT devices communicate via a network. It is common for IoT devices to communicate using a relatively short range and specialized proximity network, due to power and processing limitations. However, some devices are able to communicate at internet scale using an access network of some kind.

5) IoT Gateways are commonly used in IoT systems. They form a connection between the local proximity network(s) and the wide area access network. IoT Gateways can contain other entities and provide a wider range of capabilities. An IoT Gateway often contains a management agent, providing remote management capabilities. The IoT Gateway can contain a device data store, storing data from the associated IoT devices – this can either support local ("edge" or "fog") processing capabilities or be a means of dealing with intermittent communications networks. One or more analytics services can be supported by the IoT Gateway, typically operating on data streaming from the IoT devices or from the device data store. The IoT Gateway can also contain applications – these can be control applications, where rapid local processing is required to direct actuators based on input from sensors.

6) Applications & Services of various kinds exist in most IoT systems, with associated data stores. There is often a device data store, containing data derived from the IoT devices. There can be an analytics data store containing results from analytics services operating on device data and data from other sources. Analytics services of various types are usually present, processing device

data and other data to derive insights. Process management is usually present, controlling processes associated with the IoT system. There are applications that reflect the capabilities of the IoT system itself. Finally, there are business services which provide capabilities related to the commercial use of the system, either by end users or by other external peer systems. The applications and services communicate with IoT Gateways and IoT devices using the access network, while they communicate with each other using the services network.

7) Other applications, services and data stores are devoted to the operation and management of the IoT system itself. These include the device registry data store and an associated device identity service, which provides lookup capabilities for applications and services. There is a device management application, which provides monitoring and administration capabilities for the IoT devices in the system. There is an operational support system that provides various capabilities relating to the monitoring and management of the overall IoT system, including the offering of administration capabilities to users.

8) Access to the capabilities of the IoT system for users is provided by the access & interchange entities, which provide controlled interfaces for service capabilities, for administration capabilities and for business capabilities. Which capabilities are provided depends on access control capabilities that vary depending on the user, requiring authentication and authorization before the capabilities can be used.

9) Users of the IoT system can include both human users and digital users. Human users typically interact with the IoT system using some kind of user device. The user device can take many forms – including smart phones, personal computer, tablet or a more specialized device. In all cases, some form of application interface is offered to the human user, where the capabilities are supplied by an underlying application that interacts with the rest of the IoT system.

10) Digital systems can use IoT systems – providing for autonomic use of the system. Both user devices and digital users communicate with the rest of the IoT system via the user network, which can be the internet or can be other more specialized forms of network. For some IoT systems the user devices can interact directly with IoT devices or IoT Gateways. One of the common examples of such a system occurs with a smartphone or a wearable device, where the IoT devices and the user device are both part of a single device.

11) Peer systems, which can be other IoT systems or can be non-IoT systems, can be users of an IoT system and/or offer services to the IoT system.

Peer systems interact with the IoT system through the user network – typically the internet.

Security & privacy elements apply across the complete IoT system. These can include authentication, authorization, certificates, encryption, key management, logging and auditing, data protection such as anonymization and pseudonymization.

Based on a study of the decomposition of various IoT systems in different application scenarios, Figure 14 shows the most common IoT entities found in IoT systems. Additionally, this figure provides a very high level relationship between Domain and Entity.

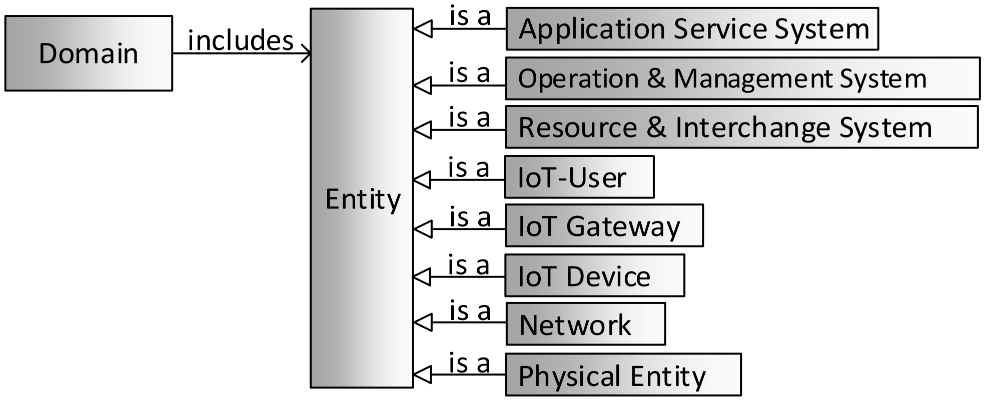

Figure 14 – Domain and entity relationship, and representative conceptual entities in the IoT systems

8.2.2 Domain-based RM

8.2.2.1 Introduction

Figure 15 shows the domain representation of the IoT RM. The domain-based RM is composed of User Domain (UD), Operations & Management Domain (OMD), Application Service Domain (ASD), Resource & Interchange Domain (RID), Sensing & Controlling Domain (SCD), and Physical Entity Domain (PED). Each identified domain is mutually exclusive from all other domains.

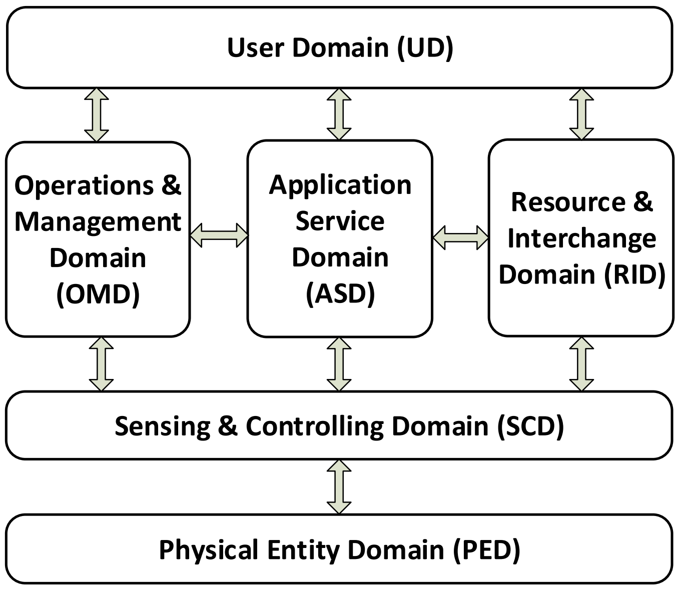

Figure 15 – Domain-based IoT RM

The IoT system's environment is mainly formed by the PED, but in certain situations, part of the SCD entities can be allotted as a part of the environment. Hardware (i.e. physical entities) and software (i.e. virtual entities) which appear in the domains other than the PED and the SCD support functions and capabilities of the domain to which they belong and they do not interact (e.g., sense and actuate) with an environment for which an IoT system is responsible and monitoring. The IoT system's environment is mainly formed by the PED, but in certain situations, part of the SCD entities can be considered as a part of the environment. Hardware (i.e. physical entities) and software (i.e. virtual entities) which appear in domains other than the PED and the SCD support functions and capabilities of the domain to which they belong and they do not interact (e.g., sense and actuate) with the environment for which an IoT system is responsible and monitoring.

The IoT domain-based RM supports planning and organization of the diverse, expanding collection of interconnected networks. Interconnected networks provide communication connectivity, including data links. These can be point-to-point links in or between IoT systems, both inter- and intra-domain, and with other systems and organizations. The connected networks should maintain interoperability from one network to another. The network mainly provides pathways for communication and data exchange. Thus, the key role of the networks is to support and provide communication and data exchange activities and interactions. The types of the activities and interactions between two entities, between two domains, or between two IoT systems determine their relationships between the entities, domains, and IoT systems, respectively. Although the inter-domain communication networks are not specifically designated as part of one of the six domains, these networks play a critical role in an IoT system. Depending on the infrastructure of IoT systems, the inter-domain communication networks can be local area network, Internet, Intranet, enterprise backbone network, or wide area network, etc. Business-to-business (B2B) networks are also considered as inter-domain communication networks.

8.2.2.1.1 The user domain (UD)

Users are the stakeholders and actors of the UD. A user can be an individual person, a group of persons such as a household, a society, an organization or a government department.

8.2.2.1.2 The physical entity domain (PED)

The PED consists of the physical and virtual entities in an IoT system. Therefore, the PED is the primary environment within which an IoT system is responsible for tasks or functions such as monitoring, sensing, and controlling. People can be one of the entities in the PED but while the owner of the PED is a stakeholder he may not be an entity in the PED.

8.2.2.1.3 The sensing & controlling domain (SCD)

The SCD is an essential domain in an IoT system because the SCD provides critical information about an environment (i.e. PED) to all the other domains of an IoT system. In addition, the SCD can manipulate the state of Physical Entities in the IoT system environment through actuation.

8.2.2.1.4 The operations & management domain (OMD)

System operators and managers are the actors of the OMD. The operators and managers maintain the overall health of IoT systems. The OMD represents the collection of functions responsible for provisioning, managing, monitoring and optimization of the systems' operational performance in real-time.

8.2.2.1.5 The resource & interchange domain (RID)

Organizations that participate in an IoT system are the stakeholders of the RID. These organizations can range from a coffee shop to utility companies or governmental organizations.

The RID interacts with the external entities, applications, services, and systems in terms of resources. The resource can be physical, monetary, or digital depending on the transactions executed through the RID.

From the perspective of the digital resources, the domain-based RM has an underlying data layer covering all six domains because the data is generated and consumed in a distributed fashion by all domains in the RM. In order to play its role, the RID needs access to the digital resources by permission of other domains (the UD, OMD, ASD, and SCD). Thus, this particular RID role can be viewed as the RID having a pseudo-information database domain. The actual data processing such as data "analytics" are performed typically in the ASD and the data after processing are stored in the service providers' database. In the RID, additional data processing may be performed, if required, to accommodate the external organizations. This additional processing may include data quality assurance, data transformation, distribution and storage.

8.2.2.1.6 The application service domain (ASD)

Application service providers are the actors of the ASD. Application service providers offer services to the IoT-User in the UD.

The ASD contains all types of service providers involved in an IoT system. Thus, the service providers interact not only with the users in the UD to fulfil their requests but also with entities in the SCD (i.e. sensors and actuators) to gain data from entities in the PED. Additionally, the ASD interacts with the OMD if an OMD stakeholder becomes a client of a service provider in the ASD. The service providers in the ASD are likely to interact with external organizations, such as other IoT systems and platforms, law enforcement, utilities, financial institutions, and governments, via the RID.

The application service providers form a business domain within the ASD; the business domain functions enable end-to-end service operations of the IoT systems.

8.2.3 Relation between entity-based RM and domain-based RM

Taking the entity-based RM in Figure 13 and the domain-based RM in Figure 15, a mapping relation between the two RMs is shown in Figure 16, where these two RMs are consistent with each other.

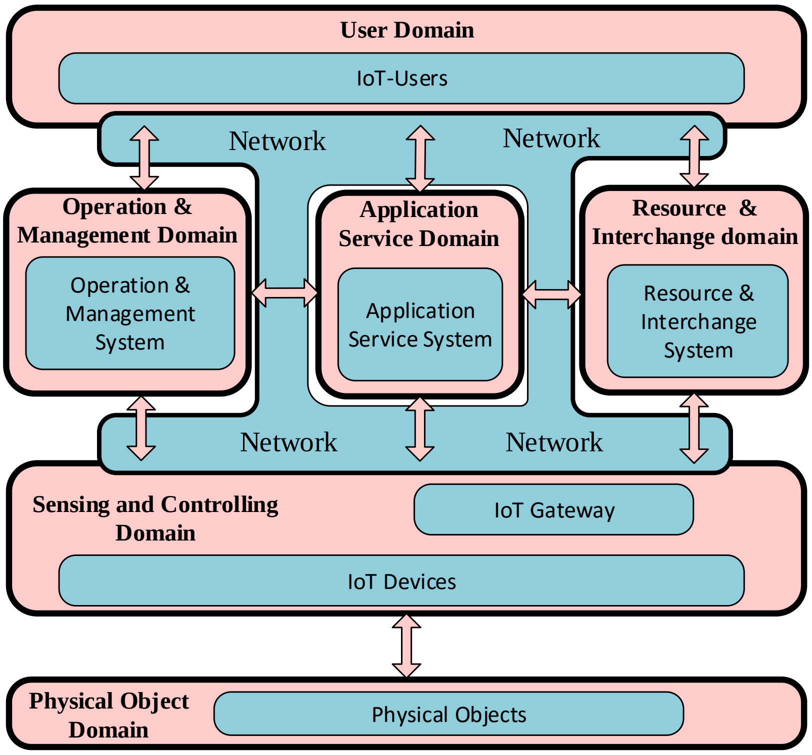

Figure 16 – Relation between entity-based RM and domain-based RM

As shown in Figure 16, the relationship between the entities and their domains is as follows. IoT-Users belong to user domain. Application service systems, operation & management systems and resource & interchange systems work in application service domain, operation & management domain and application service domain, respectively. IoT devices and IoT gateway are entities in sensing and controlling domain. Physical entity exists in physical entity domain.

8.3 IoT RA views

8.3.1 General description

The IoT RA is described by the following five RA views:

1) IoT RA functional view
2) IoT RA system view
3) IoT RA communications view
4) IoT RA information view
5) IoT RA usage view

The IoT RA becomes an application- or service-specific system architecture or a target system architecture when the RA is tailored to a specific set of requirements. Examples of specific systems are: agricultural system, environmental system, smart grid system, smart home/building, smart city, etc.

8.3.2 IoT RA functional view

The functional view is a technology-agnostic view of the functions necessary to form an IoT system. The functional view describes the distribution of and dependencies among functions for support of activities described in the user view, and addresses the following concepts:

1) Intra-domain functions
2) Cross-domain functions

Each functional component is realized by one or more implementations of actual system components, which may be deployed to form a working system. Figure 20 shows the decomposition of the IoT RA functional components. In this figure, there are two parts: intra-domain functions and cross-domain functions. The functional components are not necessary for some specific applications and therefore, in their corresponding IoT system, may not exist.

8.3.2.1 Intra-domain functions

As shown at left side of the figure, the intra-domain and cross domain functions are depicted as follows:

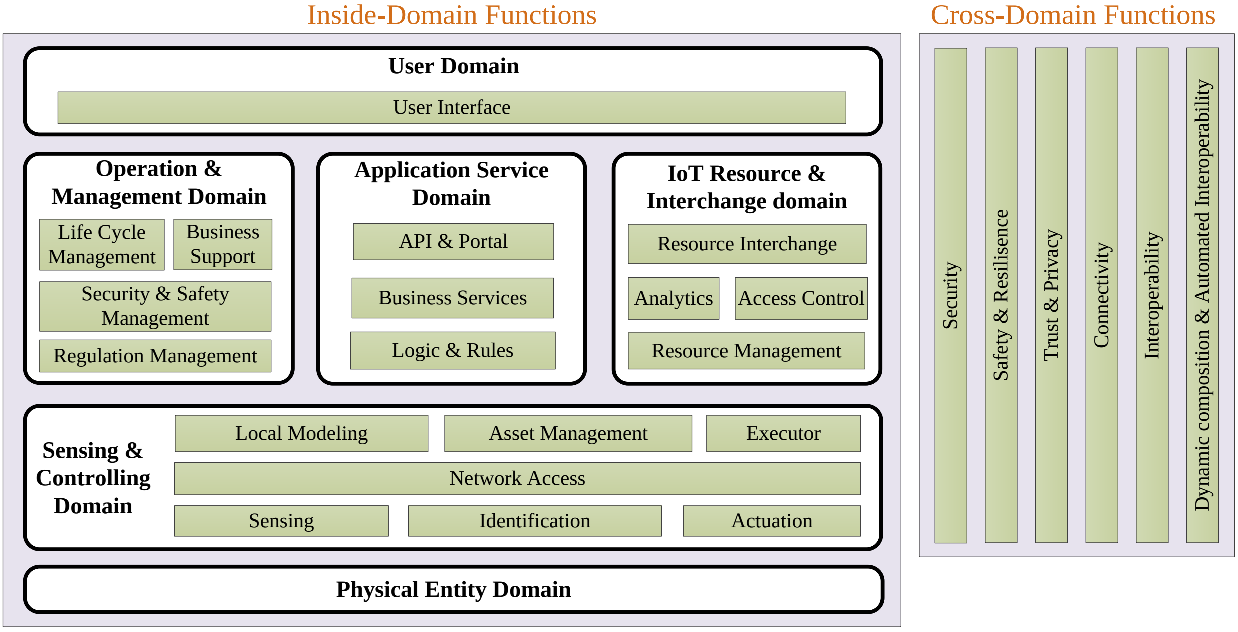

Figure17 – IoT RA functional view –decomposition of IoT RA functional components

8.3.2.1.1 The sensing & controlling domain (SCD)

The SCD is comprised of a set of common functional components whose implementation complexity depends on the infrastructure of IoT systems.

1) Sensing is the function that reads sensor data from sensors. Its implementation spans hardware, firmware, device drivers, and software elements. Recursive sensing requires control and actuation, and thus has tighter requirements than the rest of the control system.

2) Actuation is the functional component that writes data and control signals to an actuator to effect the actuation. Its implementation may span hardware, firmware, device drivers and software elements.

3) Execution is the function that executes logic which controls states, conditions, and behaviour of the system and its environment, in accordance with control objectives.

4) Identification is an essential function in a system which enables the entities to be identifiable and traceable, so that the system can distinguish an entity from others.

5) Network Access mechanisms are functions that enable connection between sensors, actuators, controllers, gateways, and other edge systems. Networks take different forms, such as a bus (local to an underlying system platform or remote), or networked architecture (hierarchical, hubs and spokes, meshed, point-to-point), some statically configured, and others dynamically. QoS characteristics such as latency, bandwidth, jitter, reliability, and resilience must be taken into account.

6) Local Modelling functions support understanding the states, conditions and behaviours of the systems under control and those of peer systems by interpreting and correlating data gathered from sensors and peer systems.

7) Asset Management functions enables operations management of the control systems including system configuration, policy, system, software/firmware updates and other lifecycle management operations. Note that it is subservient to the executor so as to ensure that policies

(such as safety and security) are always under the responsibility and authority of the edge entity.

A stakeholder is an owner or owners of the SCD, yet, this stakeholder may not show up as an entity in the SCD. The SCD could have data/processing platform and various kinds of virtual objects supporting the entities in the SCD. Thus, actors in the SCD can be physical entities (e.g., sensors, controllers, actuators, computers, etc.) or virtual entities (e.g., software).

8.3.2.1.2 The application service domain (ASD)

The ASD domain represents the collection of functions implementing application logic that realizes specific business functionalities for the service providers in the ASD. The application service domain has components such as logic and rules, functional components, APIs and portal functional component.

8.3.2.1.3 The operation & management domain (OMD)

The OMD represents the collection of functions responsible for life cycle management, business support, security and safety management, and regulation management. The management functions enable management centres to issue a suite of management commands to the control systems or the corresponding devices. The life cycle management provides several types of functional components for the IoT system operations: provisioning, deployment, monitoring, maintenance, prognostics, diagnostics, optimization, billing, etc.

1) Provisioning and deployment functions include of a set of functions required to configure, on-board, register, and track assets, and to deploy and retire assets from operations. These functions must be able to provision and bring assets online remotely, securely and at scale.

2) Monitoring and diagnostics functions enable detection and identification of problems.

3) Prognostics functional component consists of a set of functions that serve as a predictive analytics engine of the IoT system. The main goal is to identify potential issues before they occur and provide recommendations on their mitigation.

4) Optimization functional component consists of a set of functions that improve asset reliability and performance, reduce energy consumption, and increase availability and output in correspondence to how the assets are used.

8.3.2.1.4 The resource & interchange domain (RID)

The main functional components are resource management, analytics, resource interchange, access control, and so on. The IoT resource, which can be shared within an IoT system or with other IoT systems, could be intelligence, knowledge, information, data, etc. The IoT resource & interchange domain performs interchange of the IoT resource for the whole IoT system with other systems. Moreover, stakeholders in the RID need to provide, and be provided with, data regarding the IoT system, analyse the resource data and receive analyses, and store data in the cloud.

8.3.2.1.5 The user domain (UD)

The main function of the UD is to provide access to IoT services and information on how to use them. Here the functional components are users and HMIs which provides the interface for user to access, subscribe and receive the services provided by the application service domain.

8.3.2.1.6 The physical entity domain (PED)

PED has sensed physical objects and controlled physical objects which are the subject of functions in other domains.

8.3.2.2 Cross-Domain functions

Figure shows cross-domain functions which are functions exist all six domains as described in the domain-based IoT RM. These functions include security, safety, resilience, trust and privacy, connectivity, interoperability, dynamic composition and automated interoperability, etc. Each function can include functional components in different domains and be expanded in the functional domain decomposition as illustrated below.

1) The privacy function is realized through the data privacy protection in the sensing and transmission, API & portals, monitoring, information resource interchange and HMI etc.

2) The security function refers to the ability of IoT system to ensure the confidentiality, integrity, authenticity and confirmation of the exchanged information. The IoT RA integrates security policies for IoT components as key part of system design. For example, asset management in the SCD enables operations management including system configuration, policy, software and firmware updates and other lifecycle management operations. In the RID, access control and the resource management are responsible for data security, data access control and data rights management.

3) The safety and resilience function is a superset of fault tolerance and closely related to autonomic computing capabilities of self- healing, self-configuring, self-organizing and self-protecting, e.g., the IoT component can take advantage of the hierarchical network to do self-optimization.

4) The trust & privacy function is to distinguish different levels of trust for a party (e.g., application, system, network, etc.) during data transmission or exchange in order to protect the confidentiality of data. Usually, validation is required before the trust is established and trust can be enhanced by reputation-based approaches. Privacy may be achieved mostly via authentication. To prevent leaking of confidential data, additional data access rules may be used to meet necessary requirements for data requisition, removal, encryption, etc.

5) The connectivity function provides the capability of heterogeneous integration for IoT components, which may belong to different networks or using different technologies, to achieve seamless connection of each entity.

6) The interoperability function provides the capability to exchange information of an IoT system with a common interpretation of information. Basically, two levels of data interoperability are considered. Syntactic interoperability is to exchange information in a common data format with a common protocol to structure the data. Semantics interoperability is to interpret the meaning of the symbols in the messages correctly.

7) The dynamic composition & automated interoperability function provides a flexible method of composing services so that the IoT components can be dynamically integrated at run-time to enable adaptable services. Semantic interoperability is required to support the dynamic composition.

8) Privacy is achieved mostly via confidentiality. To prevent data leakage and satisfy privacy requirements access rules may be used for data requisition, removal, encryption, etc.

8.3.3 IoT RA system view

The system view describes the generic components including devices, sub-systems, and networks to form an IoT system. While the functional view describes an IoT system through its functional components, the system view describes it through its physical components. The system view describes the following aspects:

1) Key physical components (e.g. sub-systems, devices, networks) of an IoT system.

2) The general architecture of an IoT system, including the structure of an IoT system, the distribution of components, and the topology of the interconnectivity of the components.

3) A technical description of its components, including behaviours and other properties.

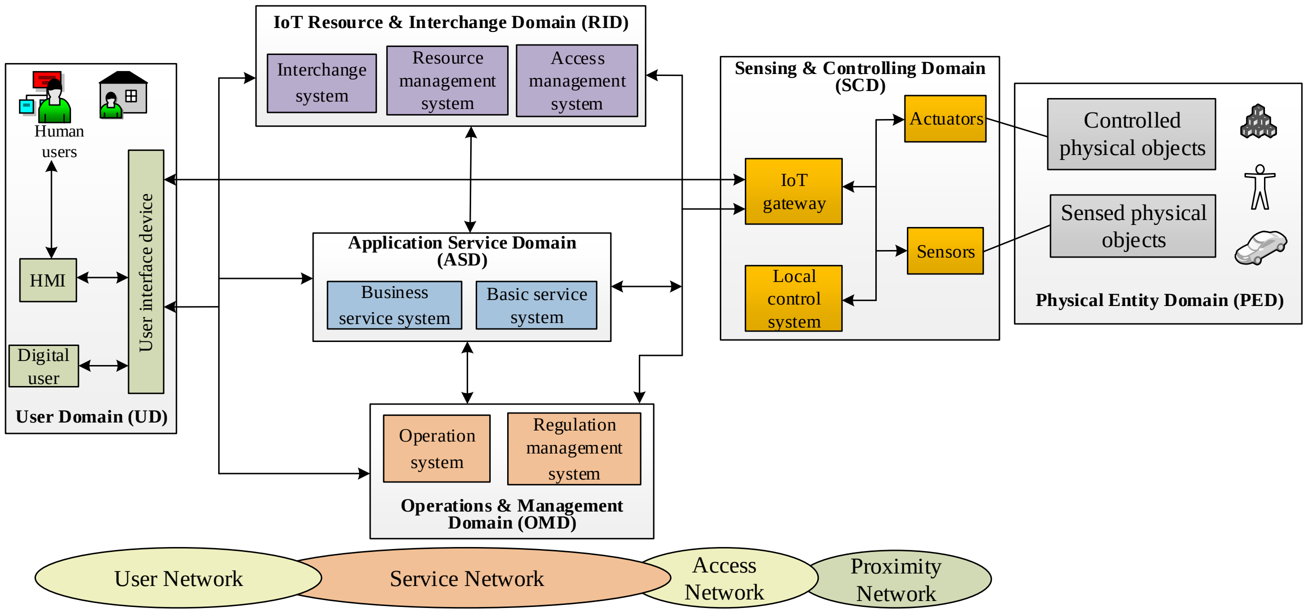

Figure 18 – IoT RA system view

In Figure 18, IoT RA system view is shown together with all the entities involved in each domain and the connections between them. The entities in each domain are very general and optional, depending on specific applications. There are four different kinds of networks to connect the physical components in the six domains of an IoT system: proximity network, access network, services network, and user network. More detailed description about these four networks will be introduced in 8.3.4 IoT RA communications view.

8.3.3.1 Systems/sub-systems in physical entity domain (PED)

In the PED, there are no devices or sub-systems. Instead, it mainly consists of sensed physical objects and controlled physical objects, which are related to IoT applications and are of interested to users. A sensed physical object is a physical entity from which information is acquired by sensors, while a controlled physical object is a physical entity which is subject to actions of actuators.

8.3.3.2 Systems/sub-systems in sensing & controlling domain (SCD)

In the SCD, the local entities consist primarily of sensors, actuators, and IoT gateways and endpoints. Sensors and actuators act upon physical entities, while IoT gateways connect the SCD to communications channels.

Sensors acquire information from the sensed physical object, e.g., physical, chemical, biological properties, etc. Actuators perform operations on controlled physical objects through controlling function units. Both sensors and actuators can act upon physical objects independently or collaboratively.

IoT gateways are devices which connect SCD with other domains. IoT Gateways provide functions such as protocol conversion, address mapping, data processing, information fusion, certification, and equipment management. IoT gateways can be either independent equipment or be integrated with other sensing and controlling devices.

The SCD might also include some local control systems such as Asset Management, Executor, etc., depending on the complexity of the IoT system infrastructure.

8.3.3.3 Systems/sub-systems in application service domain (ASD)

In the ASD, there are basic service system and business service system.

A basic service system provides fundamental data services, which include data access, data processing, data fusion, data storage, identity resolution, geographic information service, user management, and inventory management, etc.

A business service system is responsible for realization of traditional or new Internet specific types of business functions. The business functions include enterprise resource management (ERP), customer relationship management (CRM), asset management, service lifecycle management, billing, payment processing, human resource activities, work planning and scheduling systems.

8.3.3.4 Systems/sub-systems in operation & management domain (OMD)

The OMD includes operation systems and regulation management systems. Operation systems are responsible for management of IoT devices and control of the operation of the IoT system, enabling the equipment and systems operate safely and reliably. Regulation management systems act to ensure that the IoT system complies with relevant laws and regulations. They provide monitoring, supervision and execution of relevant laws and regulations.

8.3.3.5 Systems/sub-systems in user domain (UD)

In the UD, users can be human or digital. Both of them interact with other domains via the user interface device, which has an added HMI for the human user. The devices in the UD are the HMI and the user interface device.

8.3.3.6 Systems/sub-systems in IoT resource and interchange domain (RID)

In the RID, there are three major sub-systems:

1) Resource management system: This system stores and processes the resources. The resources can be divided into two types. The first is for interior usage, the second is to be shared to and from the external systems.

2) Interchange system: This system executes the interchange of the resources.

3) Access management system: This system controls access to stored resources in RID and any other resources within an IoT system. The RID serves as a bridge between an IoT system and the outside world.

The working procedure of the systems in the RID is described as follows.

Case 1:

When the external systems require resources from an IoT system, including data, financial transaction, etc., they first send request to the Interchange Systems in the RID. Then Interchange System forwards the request to the Access Management System, which decides whether to accept this request. If accepted, it authorizes the ASD, SCD or other domains to provide relevant resources, which are sent back to the resource management system for data format conversion, data fusion, etc. Then the data management system transmits those requested resources to the interchange system, which acts as an interface between the IoT system and external systems. If not accepted, the access management system directly sends a response of "No" to the interchange system, which says "No" to the request from the external systems.

Case 2:

When the IoT system requires some resources from external systems, it first sends a request to the access management system in the RID for access authority. After that, the access is authorized and the AMS sends a resource interchange request to the interchange system in the RID, which forwards this request to the external systems. If the interchange system gets positive response, it delivers relevant resources and stores them in the data management system in the RID. Then the data management system further forwards the resources to the originator of the request in the interior IoT system, such as the ASD or SCD. If the Interchange System receives negative response, it delivers this message to the originator of the request.

8.3.4 IoT RA communications view

8.3.4.1 Communications networks

The IoT RA communications view describes the principal communications networks which are involved in IoT systems and the entities with which they connect. The four principal communications networks are shown in Figure 19.

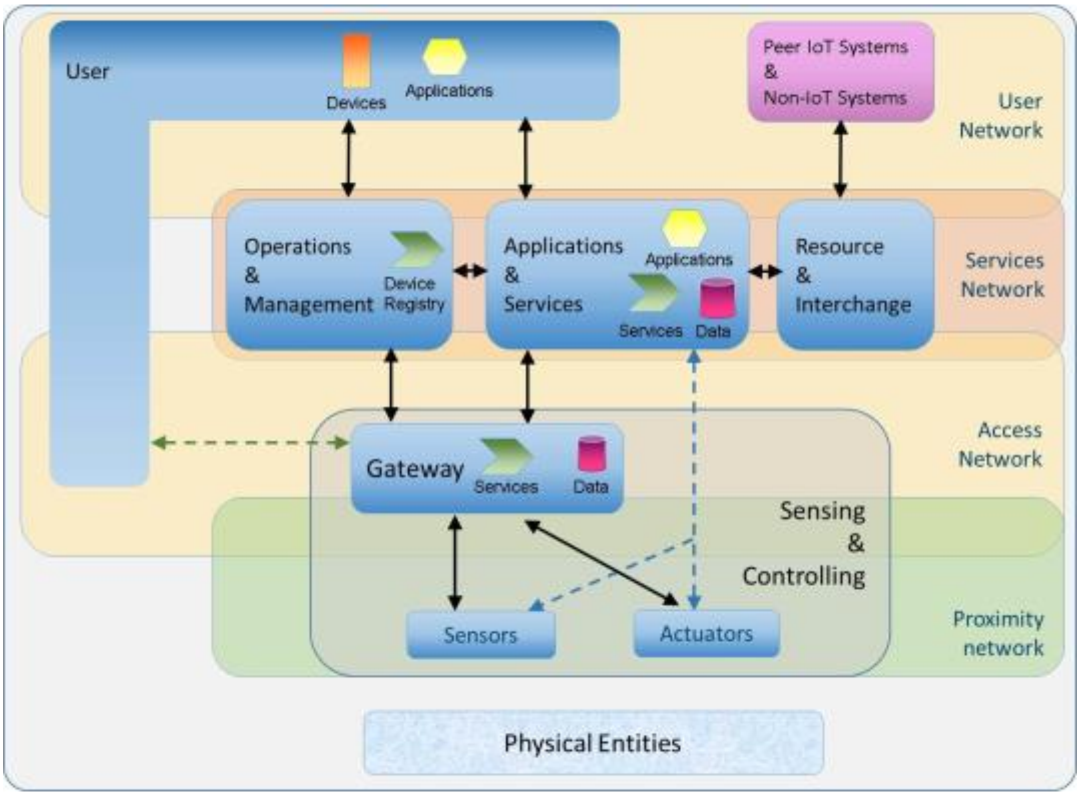

Figure 19 – IoT RA communications view

8.3.4.1.1 Proximity network

This network exists within the Sensing and Controlling domain. Its main task is to connect the sensors and actuators to gateway. Proximity networks are typically local and limited in range, largely necessary because sensors and actuators are low power, or are in locations that make wide area connections (such as the internet) difficult or impossible to provide.

Proximity networks may use specialized protocols and may not use IP.

Proximity networks often uses low power limited range wireless and wired technologies. Current examples include IPv6 over Low Power Wireless Personal Area Networks (6LoWPAN), ZigBee, Narrow-Band IoT.

Individual sensors and actuators may have limited power and limited hardware capabilities, which means that simple, local, and low-power networks are needed to connect them to gateways. These are more powerful and can in turn connect to access networks.

Proximity networks may involve the use of an address translation capability to translate between their local addressing schemes and addressing schemes used on access networks.

8.3.4.1.2 Access network

Access networks are typically wide area networks connecting devices in the SCD to the other domains – the ASD and the OMD. Access Network typically connects to gateways, but When Sensors and Actuators are more capable, they may connect directly to Access Networks (dashed lines in Figure 22). A range of technologies can be used in access networks including wired connections (Broadband / ADSL / Fiber) and wireless connections including Wireless LANs (Wi-Fi), Mobile (cellular) networks and Satellite links (particularly for remote locations). Access Networks typically use IP. Access networks may involve the use of a device registry that holds data about the IoT devices associated with the IoT system and how to communicate with them.

8.3.4.1.3 Services network

This network connects elements within and between the ASD, the RID, and the OMD. This network can include both Internet elements and also (private) intranet elements. It is typical for intranet networks to be used where the elements of the other domains exist within a single data center. Services networks typically use IP.

Service Networks connect the applications and services in the ASD, the RID and the OMD, which are typically wired networks within data centers, running IP-based protocols. Where communication spans multiple data centers, a variety of network technologies may be used, including both dedicated connections and Internet connections.

8.3.4.1.4 User network

This network connects the User domain with the ASD and OMD. It also connects Peer IoT Systems and non-IoT systems with the RID. This network is typically based on public Internet elements and uses IP.

User Networks connect to user devices, to peer IoT systems and to other non-IoT systems, which can be internet-based. Such networks can use any of the technologies commonly used to carry internet traffic, including both wired and wireless systems.

8.3.4.2 Communication networks implementation

Each of the principal communications networks can be implemented by means of a range of different network technologies, which are used depending on the particular characteristics and requirements of the IoT system. IoT system implementations may use multiple instances of each of these networks to create complete solutions.

In Figure , the user domain is shown spanning both the user network and the access network. This describes those cases where user devices and their applications connect directly to the SCD, for example when the user device is a smart phone which contains sensors.

8.3.5 IoT RA information view

8.3.5.1 General description

The information is generated by using, monitoring, controlling and analysing connected entities.

Some information is static, e.g. identifier of an entity, while other information may be variable, e.g. location of an entity. Some static information is also key for associating variable information to an entity. Some information may also be static as information but variable in its usage.

Both raw and processed information is used by application (service), operator, manager, administrator, customers, users, etc. to fulfil intended task for a given activity in an IoT system. The information can stay within a “domain” or be exchanged between “domains”. Figure 20 illustrates some examples of data which are stored in relevant domains.

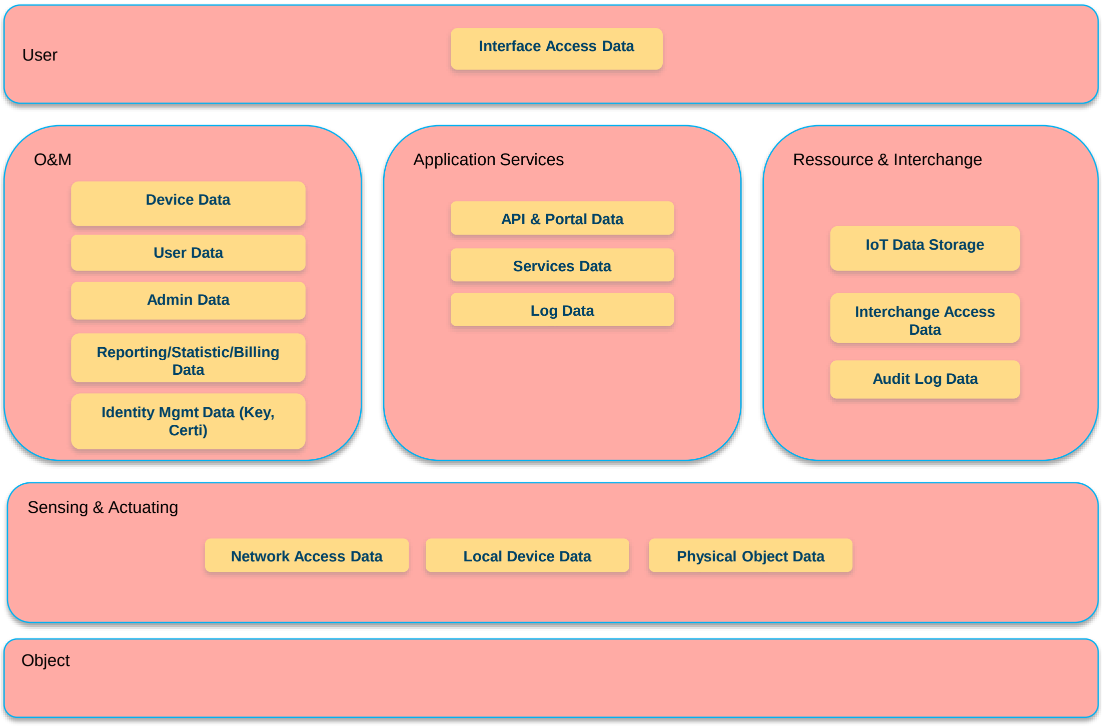

Figure 20 – Type of information related to domains

The IoT RA information view defines the structure (e.g. relations, attributes, services) of the information for Entities on a conceptual level. Data is defined as pure values without relevant or useable context. Information adds the right context to data and offers answers to typical questions like why, who, what, where and when.

The description of the representation of the information (e.g. binary, XML, etc.) is not part of the IoT information view.

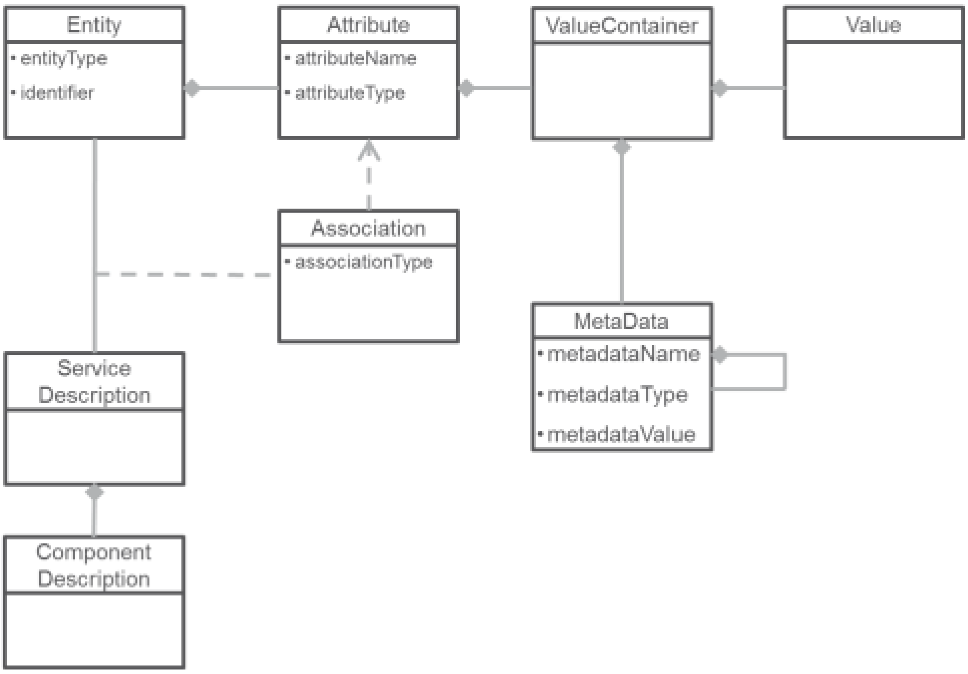

Figure 21 – Type of information related to domains.

8.3.5.2 Minimum required information for identification

The table below states the minimum information for to be able to identify an entity.

Table 2 – Information for identification

Element	Comment
Entity Identity	Key element against which attributes are connected
Entity Name	
Entity Description	

8.3.5.3 Minimum required information for communication

The table below states the minimum information for to be able to connect an entity to other entities.

Table 3 – Information for communication

Element	Comment
Entity Identity	Key element against which attributes are connected
Entity Name	
Entity Description	

8.3.5.4 Minimum required information for authentication

The table below states the minimum information for to be able to ensure authentication.

Table 4 – Information for authentication

Element	Comment
Entity Identity	Key element against which attributes are connected
Entity Secure key	
Authentication Type	

8.3.6 IoT RA usage view

8.3.6.1 General description

Whereas the functional view shows the necessary functions and dependencies of the IoT system, the user view focuses on how the IoT system is developed, tested, operated and used from a user perspective. This view addresses the following concepts:

1) Activities;

2) Roles and sub-roles;

3) Services and cross-cutting aspects.

8.3.6.2 Description of the roles, sub-roles and related activities

All IoT related activities can be categorized into three user groups as listed below:

1) IoT service provider
2) IoT service developer
3) IoT-User

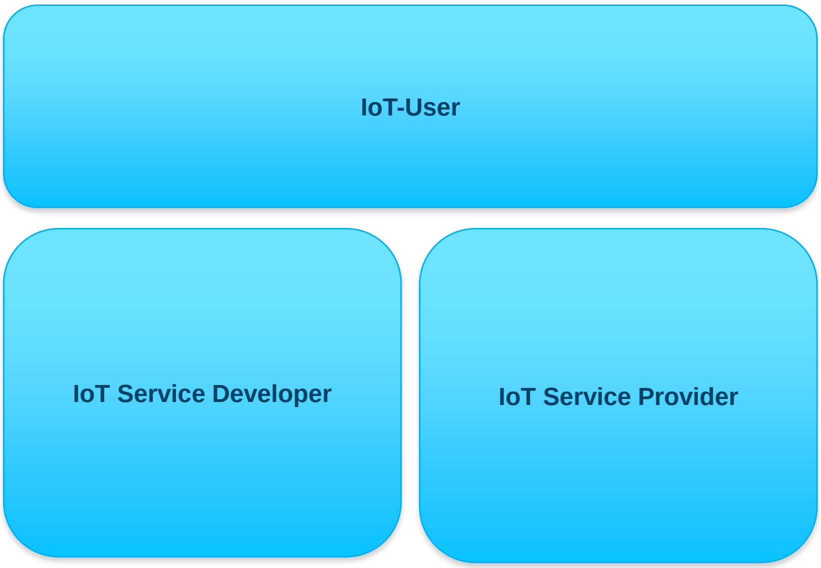

Figure 22 – IoT-User groups and roles

8.3.6.2.1 IoT service provider

Figure 23 – IoT service provider

The role of the IoT service provider is to manage and to operate IoT services. The following sub-roles can be identified:

A business manager is leading a business of existing and new products, who wants to understand how to leverage the data and connectivity of devices to create new streams of revenue. They will discover industry content on company web site and act on solution proposals from architect.

A service delivery manager is responsible for a SLA with a client to the LOB. With a team of maintenance engineers, they use the IoT enabled platform and LOB industry applications for planning, installing, monitoring and servicing equipment. This role ensures that overall service delivery quality is within the service level agreements parameters.

A system operator handles the day to day system operations for a customer by enrolling new users and making sure that new device types and devices are registered, are behaving correctly, and are up to date with the current secure firmware.

A security analyst mitigates security risks by proactively creating algorithms that detect threats and prevent breaches. They create automatic functions that act on misbehaving devices and users and also ensure compliance through audits.

An operations analyst is responsible for the availability of specific assets in the LOB product line and uses big data analysis capabilities in the IoT platform and the data scientist's algorithmic service extensions to ensure such availability.

A data scientist understands the industry data delivered from devices and the algorithms that provide meaningful analyses. He implements advanced algorithms as services to be used by the LOB analysts and LOB industry applications.

Figure 24 shows the activities which relate to the sub-roles of IoT service provider

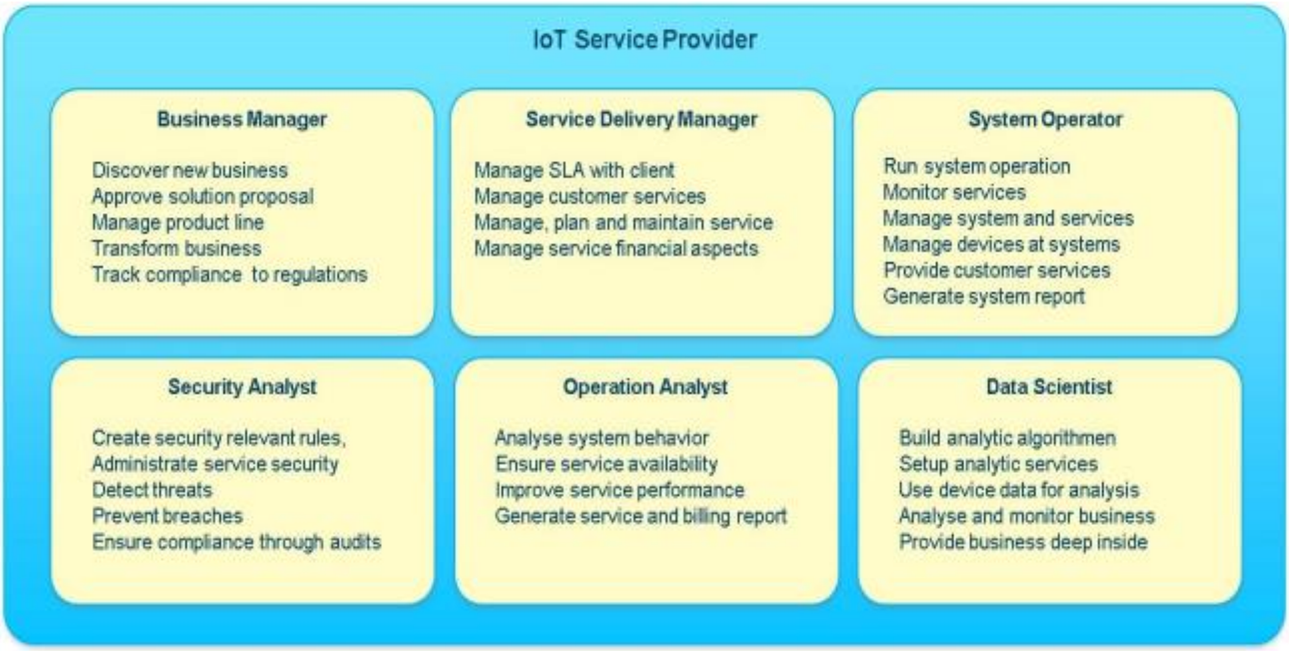

Figure 24 – IoT service provider sub-roles and activities

8.3.6.2.2 IoT service developer

Figure 25 – IoT service developer

The roles of the IoT service developer include implementation, testing and integration of IoT services with the IoT platform. Sub-roles of the IoT service developer are described as follows.

A solution architect proposes, proves and deploys the IoT enabled platform to the LOB deciding on integration strategies and architectures for the new IoT enabled platform, existing business systems and devices in production.

A development operations manager sets up, configures and operates the IoT enabled platform, relevant services and acts as a project manager by supporting IT services for LOB operations and development.

An application developer works in the LOB, in IT or with a 3rd party developing IoT industry applications for the LOB and uses development operation capabilities to develop, deploy and fix applications that integrate IoT devices, data and services.

A device developer integrates hardware and software into devices and applications, developing and maintaining device firmware that securely connects devices to an IoT-enabled platform.

A system integrator tests and integrates IoT services with the IoT enabled platform. All IoT service developer sub-roles and their activities are shown in Figure 26.

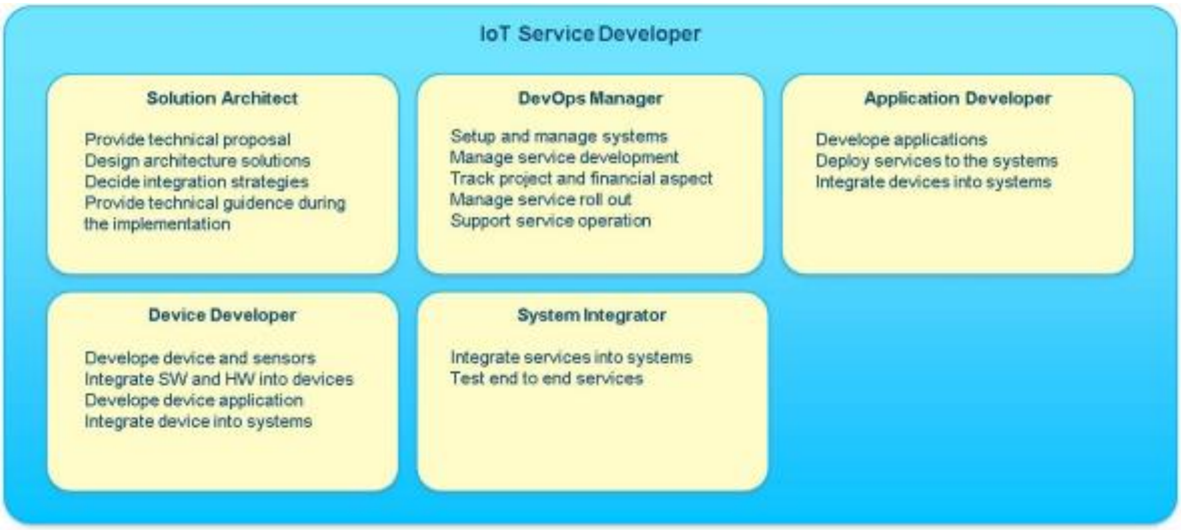

Figure 26 – IoT service developer sub-roles and activities

8.3.6.2.3 IoT-User

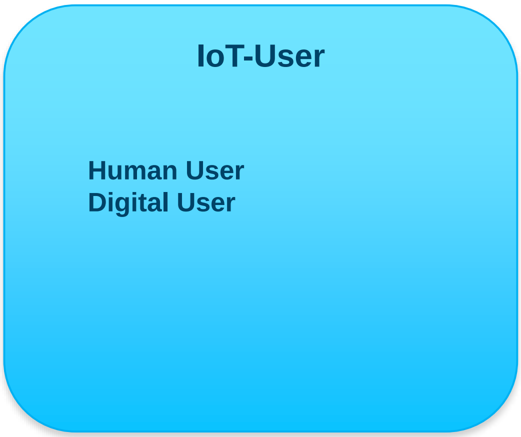

Figure 27 – IoT-User

The IoT-User is the end-user of IoT services and can be categorized into human users and digital users.

Human users are individuals who use IoT services. Digital users are non-human users of the IoT system; they can include automation services that act on behalf of human user.

All IoT-User sub-roles and their activities are show in Figure 28.

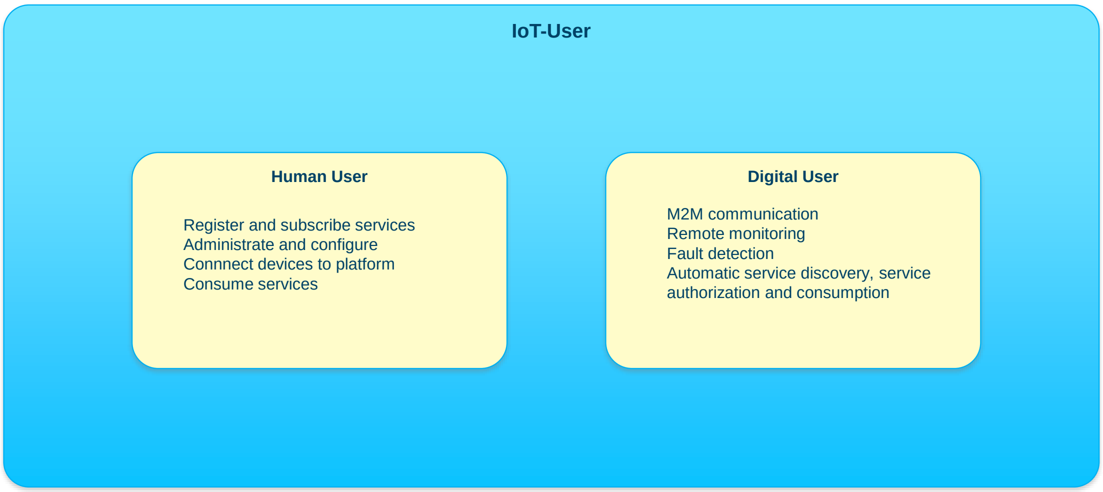

Figure 28 – IoT-User sub-roles and activities

8.3.6.3 Mapping activities, roles and IoT systems in domains

The user view addresses the concerns of expected system usage.

Roles and activities involving IoT-Users to deliver functionality achievable with the fundamental system capabilities are represented by this view. Activities which create, implement, test, integrate and operate IoT services in desired systems may require co-operation among individuals with different roles or skills.

Figure 29 shows the roles when the system is in use and opportunities for co-operation.

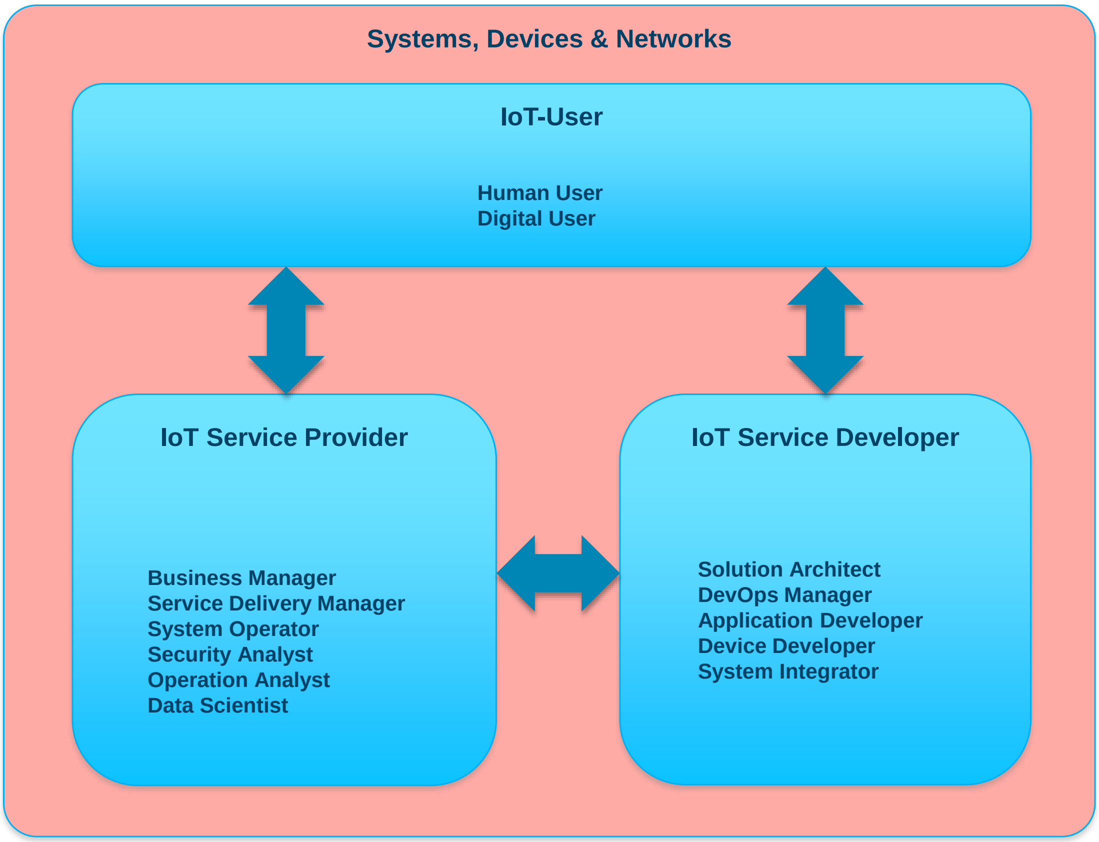

Figure 29 – Roles present when the system is in use

Table 5 provides an overview of activities and their relevant roles.

Table 5 – Overview of activities and roles

Activities	Roles	IoT Systems in Domains
Device and Application Development	DevOps Manager, Device Developer, Application Developer	Application Service Domain, Sensing & Controlling Domain
Operation of devices, connectivity and applications	System Operator, Service Delivery Manager	Operation & Management Domain, Application Service Domain
Use device data for analytics	Data Scientist, Security Analyst, Operation Analyst	Operation & Management Domain, Information& Interchange Domain
Integrate, operate and control data stores and business	Solution Architect, DevOps Manager, System Operator, System Integrator, Service Delivery Manager	Application Service Domain, Operation & Management Domain
Use real-time, historic and big data for applications and analytics	Data Scientist, Operation Analyst, Security Analyst, Service Delivery Manager	Application Service Domain, Operation & Management Domain, Sensing & Controlling Domain, Information& Interchange Domain

Make and operate analytics to run business	Data Scientist, Operation Analyst, Application Developer, DevOps Manager	Application Service Domain, Information& Interchange Domain
Bring in analytics to dashboard	DevOps Manager, Data Scientist, Application Developer	Application Service Domain, Operation & Management Domain, Information& Interchange Domain
Monitor system state, act on security risks and beaches	System Operator, Security Analyst	Operation &Management Domain
Track compliance to regulations	Business Manager, Security Analyst	Application Service Domain, User Domain

Figure 30, Figure 31, and Figure 32 show some examples of using IoT systems from different activity perspectives.

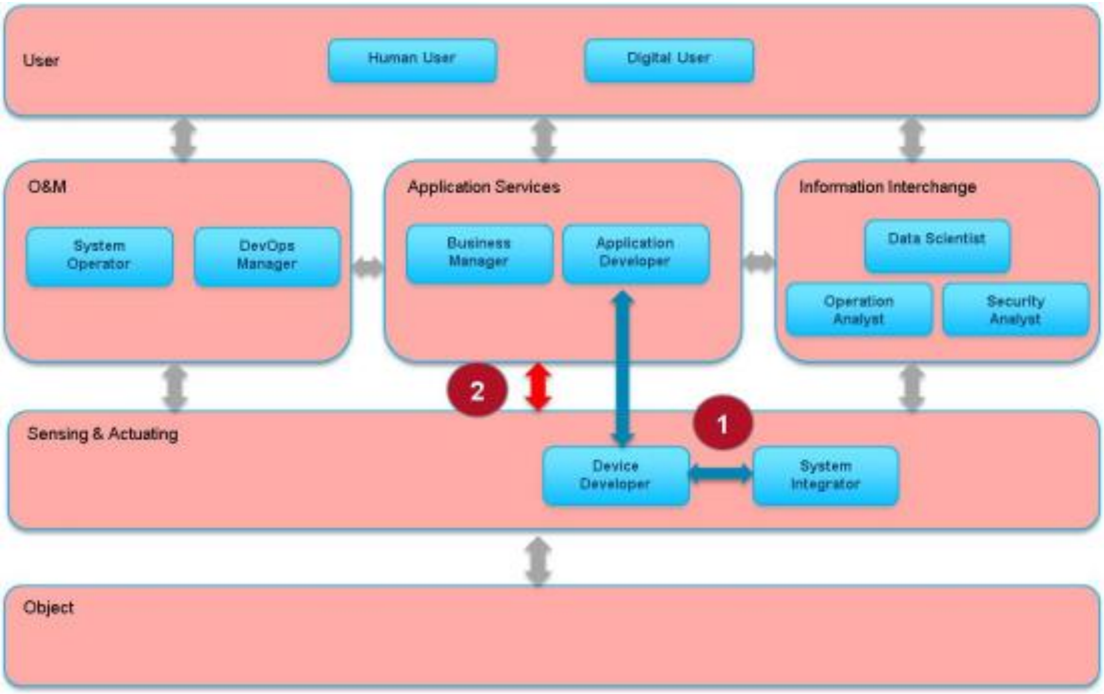

Figure 30 – Activities of device and application development

Figure 30 shows an example of activities and information exchange during device application development between device developers, system integrators and application developers. An example of a specific user activity is connecting a new device to the IoT platform. The blue boxes in Figure 30 represent the human users (in this case developers and operators) of IoT systems. The six domains of an IoT system are represented by pink boxes. For this activity:

1) The device developer communicates with the system integrator during the implementation phase. They discuss API definitions and functional behaviour between the device and the IoT platform and agree a specification.

2) The application and device developers implement and test APIs and their functions related to the device and the IoT platform. At this stage, devices in the SCD will be connected to IoT systems in the ASD and end to end functions can be tested.

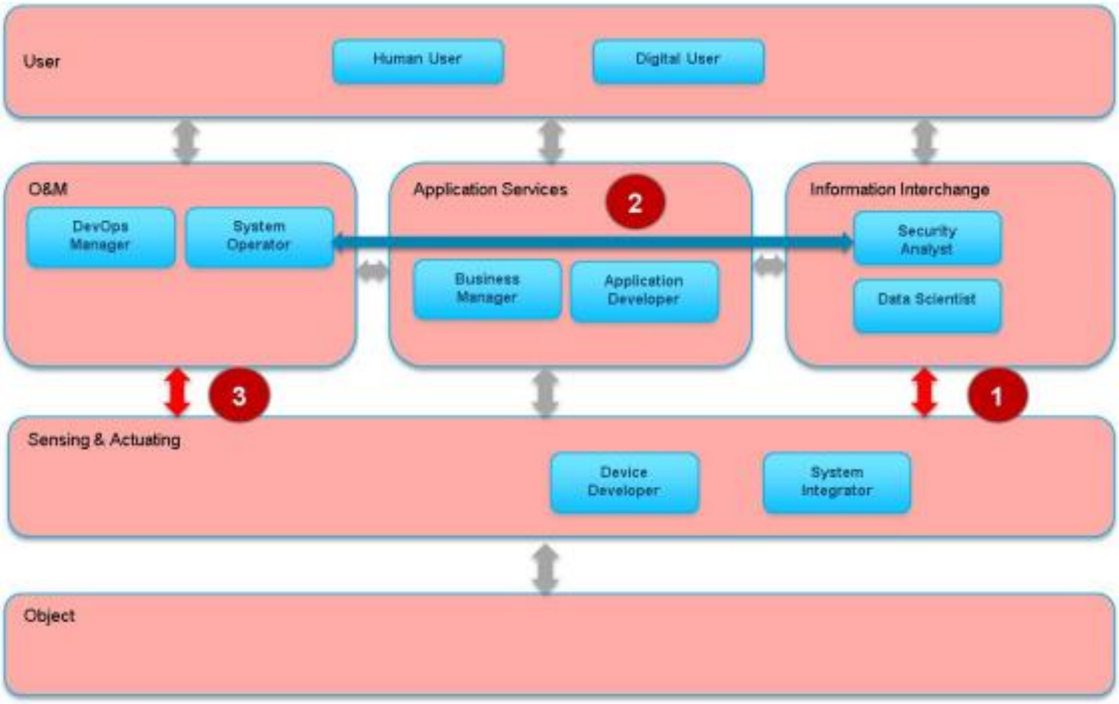

Figure 31 – Using device data for security-related analytics and operations

Figure 31 shows an example of activities involved in using device data for security-related analytics and operations. In this case the users of the IoT systems are the data analyst and security operator. Activities are:

1) When the device is configured and connected to the communications system, usage data can be sent to the IoT systems in the RID. The security analysts and data scientists can use the collected device usage data to perform security-related analyses.

2) Security analysts communicate with system operators with findings and results from their analyses.

3) Security analysts together with system operators proactively create rules to protect systems and to prevent breaches.

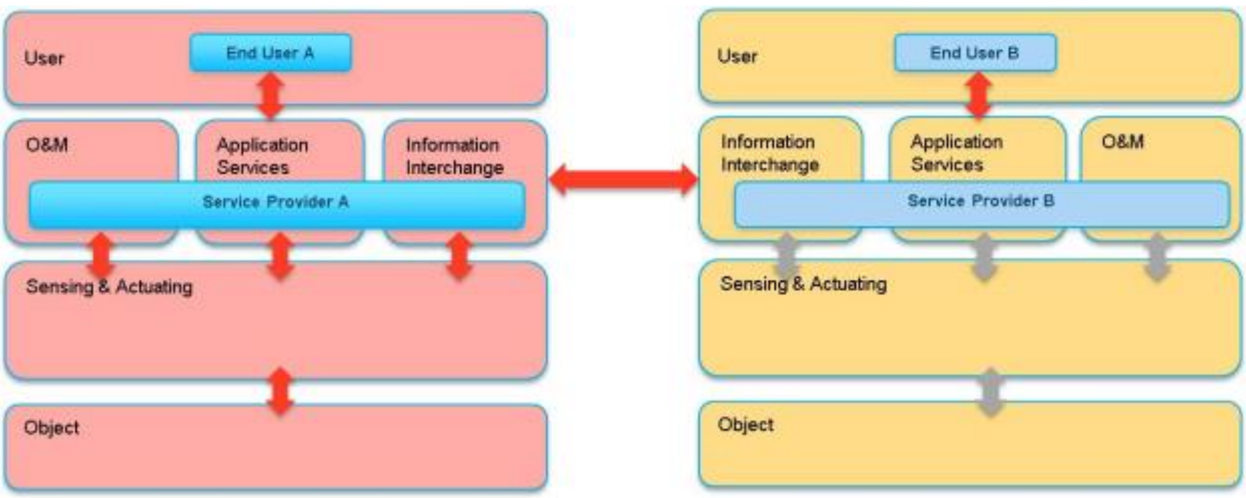

Figure 32 – Using IoT Services across Vertical Sectors

Figure 32 shows an example of using an IoT service across vertical sectors. This example is related to consumers and the product manufacturing industry (vehicle manufacturing). End user A represents the customer who is the owner of a new car. End user B represents an engineer or designer in the vehicle industry.

1) Sensors installed in the car can provide the vehicle run time status data.

2) IoT services perform analytics and inform the driver of defects (e.g. low coolant levels) or a need for inspection.

3) Such customer car usage data together with millions of other customer's car usage data can be sent to a centralised manufacturers' database through a resource interchange interface.

Based on the data collected from the customer end user B can get real-time information of car usage, and identify which mechanical or electrical part needs maintenance, replacement or is not working reliable. End user B can aggregate the real-time information with other data and further analyse potential reasons for the problem. Such information may also improve the design of components or help in the design of better quality new cars. Roles and activities during the IoT product life cycle (can be moved into Annex later)

IoT services can be used by all vertical sectors to support and transform every part of the business. Generally, they can be considered as a kind of enablement platform, which improves operations, or lowers the cost, or creates new products and business models, or drives engagement and customer experiences.

Figure 33 shows roles and activities involved when existing systems create, develop, operate and finally decommission IoT services.

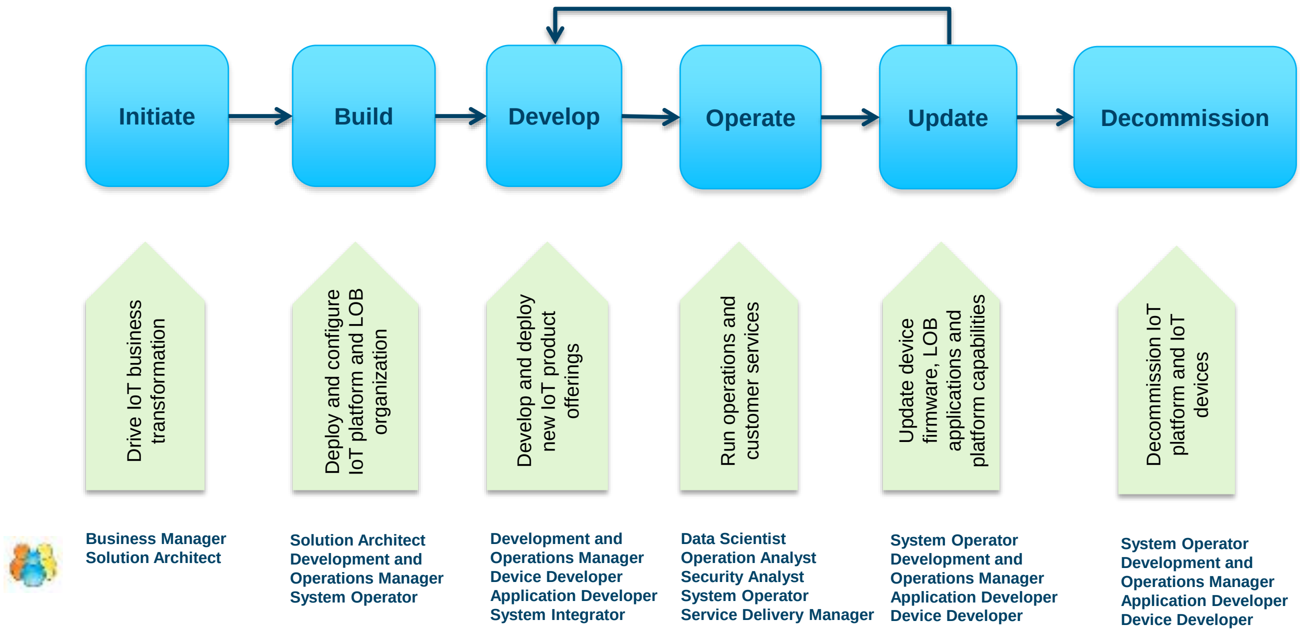

Figure 33 – Roles and Activities during the IoT Product Life Cycle

Annex A
(informative)

Interpreting model diagram

In this document, UML Class diagrams have the following restrictions:

1) Concepts are represented as UML Classes with no attributes.

2) The documentation for each concept is the definition of the concept.

Only two kinds of associations are used:

1) Generalization (an "is-a" relationship): For example, a sensor is an IoT device. This generalization relationship can be expressed as shown in Figure A.1:

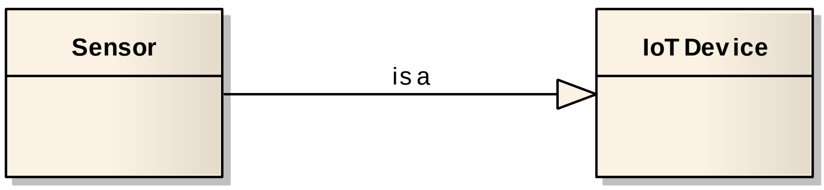

Figure A.1 – Generalization

2) Directed association expresses relationship between concepts. These association names are verbs. Figure A.2 expresses the association relationship that the Sensor monitors the Physical Entity (the thing).

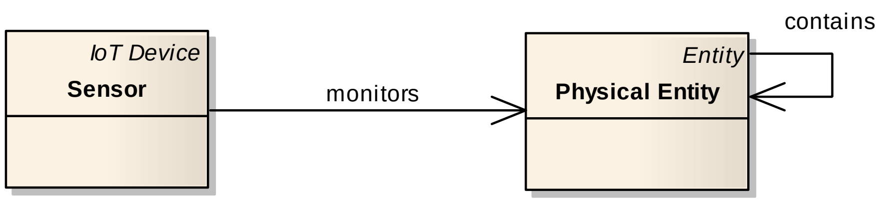

Figure A.2 – Association

Cardinality constrains on association ends are not shown. They vary from one kind of association to another, but can be inferred from the descriptions in the following clauses.

If a concept, which is a generalization of a concept on the diagram, is not itself shown on the diagram, the name of that generalized concept appears in italics at the top right corner of the box as shown in Figure A.2 ("*Entity*" and "*IoT device*").

Annex B
(informative)

Entity relationship tables for the CM

B.1 IoT entities and domains

Table A.1 – Entity

No	Relationship Type	Name	Related Concept	Description
1	Association	has	Identity	Entity has identity.

Table A.2 – Domain

No	Relationship Type	Name	Related Concept	Description
1	Association	includes	Entity	A domain includes one or more entities.
2	Association	contains	Domain	A domain may contain sub domains.
3	Association	interacts	Domain	A domain may interact with other domains.

Table A.3 – Digital entity

No	Relationship Type	Name	Related Concept	Description
1	Generalization	is a	Entity	A digital entity is a specialization of entity.
2	Association	contains	Digital Entity	A digital entity may contain other digital entities.

Table A.4 – Physical entity

No	Relationship Type	Name	Related Concept	Description
1	Generalization	is a	Entity	A physical entity is a specialization of entity.
2	Association	contains	Physical Entity	A physical entity may contain other physical entities.

Table A.5 – IoT-User

No	Relationship Type	Name	Related Concept	Description
1	Generalization	is a	Entity	An IoT-User is a specialization of entity representing a human user or digital user.

Table A.6 – Network

No	Relationship Type	Name	Related Concept	Description
1	Generalization	is a	Entity	A network is a specialization of entity.

B.2 Identity

Table A.7 – Identifier

No	Relationship Type	Name	Related Concept	Description
1	Association	identifies	Entity	Identifier identifies entity.
2	Association	distinguishes	Identity	Identifier distinguishes identity. Identity may have more than one identifier.
3	Association	identified	Identity Context	Identifier identified within a given identity context.

B.3 Services, network, IoT device and IoT gateway

Table A.8 – Endpoint

No	Relationship Type	Name	Related Concept	Description
1	Association	contains	Network Interface	An endpoint may contain more than one network interface.

Table A.9 – IoT Gateway

No	Relationship Type	Name	Related Concept	Description
1	Association	interacts through	Network	One or more networks through which interactions are made with other entities.

2	Association	exposes	Endpoint	One or more endpoints by which interactions are made.
3	Association	uses	Data Store	Zero or more data stores used by the IoT gateway.
4	Association	connects	IoT Device	One or more IoT devices which are connected via the IoT gateway.

Table A.10 – IoT Device

No	Relationship Type	Name	Related Concept	Description
1	Association	interacts through	Network	One or more networks through which interactions are made with other entities.
2	Association	exposes	Endpoint	One or more Endpoints by which interactions are made.
3	Association	uses	Data Store	Zero or more data stores used by the IoT gateway.

Table A.11 – Service

No	Relationship Type	Name	Related Concept	Description
1	Association	implemented by	Component	A Service is implemented by one or more components.
2	Association	exposes	Endpoint	A service defines network interfaces and exposed by an endpoint.
3	Association	interacts through	Network	A service interacts with other entities via one or more networks.
4	Association	interacts with	IoT Gateway	A service interacts with zero or more IoT gateways.
5	Association	interacts with	IoT Device	A service interacts with zero or more IoT devices.
6	Association	interacts with	Service	A service interacts with zero or more other services.
7	Association	uses	Data Store	Zero or more data stores used by the service.

B.4 IoT-User

Table A.12 – Human user

No	Relationship Type	Name	Related Concept	Description
1	Generalization	is a	IoT-User	A human user is a specialization of an IoT-User.
2	Association	interacts	Application	A human user interacts across the network via an application.

Table A.13 – Digital user

No	Relationship Type	Name	Related Concept	Description
1	Generalization	is a	IoT-User	A digital user is a specialization of an IoT-User.
2	Association	interacts	Service	A digital user interacts with one or more services offered by the IoT system across the network.

Table A.14 – Application

No	Relationship Type	Name	Related Concept	Description
1	Generalization	is a	Service	An application is a service.

B.5 Virtual entity, physical entity and IoT device

Table A.15 – Sensor

No	Relationship Type	Name	Related Concept	Description
1	Generalization	is a	IoT Device	A sensor is a specialization of an IoT device.
2	Association	monitors	Physical Entity	A sensor monitors a physical entity.

Table A.16 – Actuator

No	Relationship Type	Name	Related Concept	Description
1	Generalization	is a	IoT Device	An actuator is a specialization of an IoT device.
2	Association	acts	Physical Entity	An actuator acts on a physical entity.

Table A.17 – Virtual entity

No	Relationship Type	Name	Related Concept	Description
1	Association	interacts	Endpoint	A virtual entity interacts through an endpoint.
2	Association	represents	Physical Entity	A virtual entity represents a physical entity.

Annex C
(informative)

Overall IoT infrastructure at high-level

Figure A.3 shows how one IoT system can be combined with another. The arrows in the figure represent the communication and data exchange between the IoT systems, which is enabled by the RID in each IoT system. This is illustrated by one IoT System connecting to another, e.g., IoT Systems A, B and C in Figure A.3.

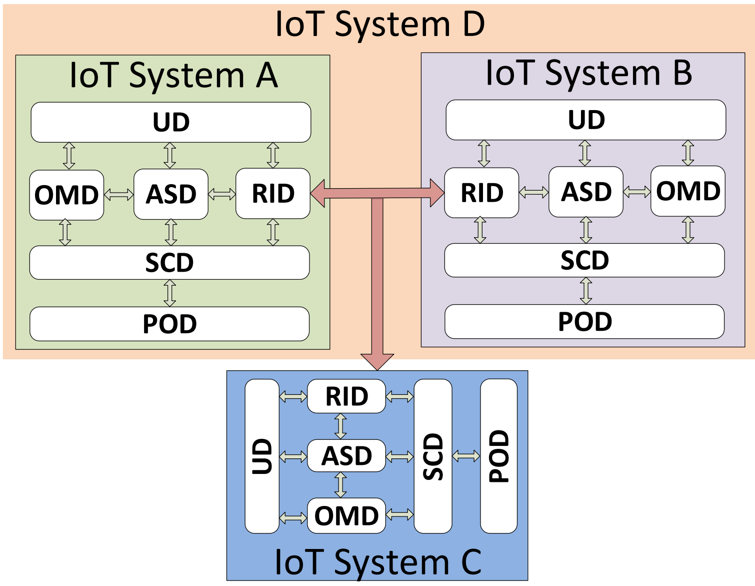

Figure A.3 – Integration of an IoT system with others

In Figure A.4, an overall IoT infrastructure is presented from a system point of view. It illustrates how various types of IoT systems in vertical ASDs can be integrated for interoperability through IoT platforms at different organizational levels (e.g. national, provincial, corporation, enterprise or global.).

Additionally, one IoT system can directly interact with other IoT systems when both mutually benefit from the direct interaction. Furthermore, an IoT system can access services implemented on external, third party, systems such as banking and financial services, medical services, billing services, etc.

The lines in Figure A.4 represent network connectivity, and the grey circles represent interoperable access points (e.g., IoT gateways).

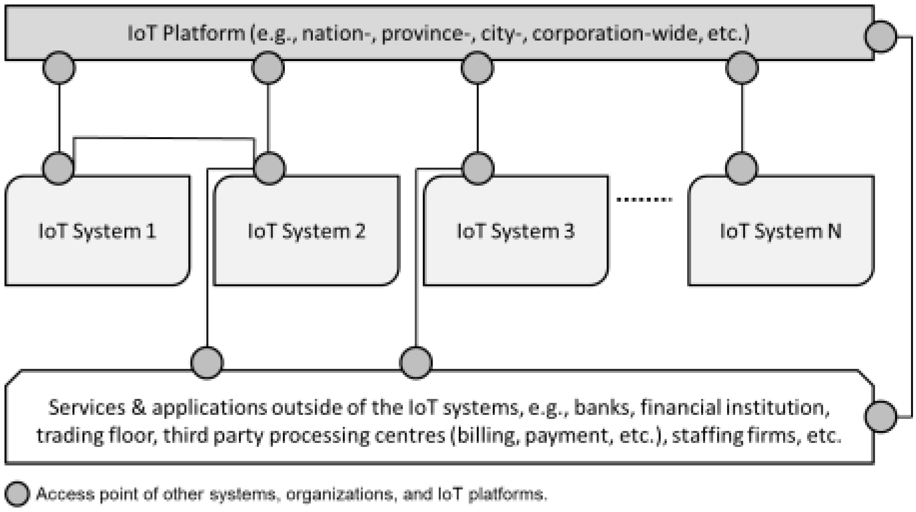

Figure A.4 – An Overall IoT Infrastructure

Bibliography:

[1] https://www.oasis-open.org/committees/soa-rm/faq.php

