ISO/IEC 30141

Edition 1.0 2018-08

INTERNATIONAL STANDARD

colour inside

Internet of Things (IoT) – Reference architecture

ISO/IEC 30141:2018-08(en)

IEC Central Office
3, rue de Varembé
CH-1211 Geneva 20
Switzerland

Tel.: +41 22 919 02 11
info@iec.ch
www.iec.ch

About the IEC
The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications
The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigenda or an amendment might have been published.

IEC Catalogue - webstore.iec.ch/catalogue
The stand-alone application for consulting the entire bibliographical information on IEC International Standards, Technical Specifications, Technical Reports and other documents. Available for PC, Mac OS, Android Tablets and iPad.

IEC publications search - webstore.iec.ch/advsearchform
The advanced search enables to find IEC publications by a variety of criteria (reference number, text, technical committee,…). It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished
Stay up to date on all new IEC publications. Just Published details all new publications released. Available online and also once a month by email.

Electropedia - www.electropedia.org
The world's leading online dictionary of electronic and electrical terms containing 21 000 terms and definitions in English and French, with equivalent terms in 16 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC Glossary - std.iec.ch/glossary
67 000 electrotechnical terminology entries in English and French extracted from the Terms and Definitions clause of IEC publications issued since 2002. Some entries have been collected from earlier publications of IEC TC 37, 77, 86 and CISPR.

IEC Customer Service Centre - webstore.iec.ch/csc
If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: sales@iec.ch.

ISO/IEC 30141

Edition 1.0 2018-08

INTERNATIONAL STANDARD

colour inside

Internet of Things (IoT) – Reference architecture

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

ICS 35.020

ISBN 978-2-8322-5972-6

Warning! Make sure that you obtained this publication from an authorized distributor.

CONTENTS

INTERNET OF THINGS (IoT) – REFERENCE ARCHITECTURE

FOREWORD

1) ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.
2) The formal decisions or agreements of IEC and ISO on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees and ISO member bodies.
3) IEC, ISO and ISO/IEC publications have the form of recommendations for international use and are accepted by IEC National Committees and ISO member bodies in that sense. While all reasonable efforts are made to ensure that the technical content of IEC, ISO and ISO/IEC publications is accurate, IEC or ISO cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
4) In order to promote international uniformity, IEC National Committees and ISO member bodies undertake to apply IEC, ISO and ISO/IEC publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any ISO, IEC or ISO/IEC publication and the corresponding national or regional publication should be clearly indicated in the latter.
5) ISO and IEC do not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. ISO or IEC are not responsible for any services carried out by independent certification bodies.
6) All users should ensure that they have the latest edition of this publication.
7) No liability shall attach to IEC or ISO or its directors, employees, servants or agents including individual experts and members of their technical committees and IEC National Committees or ISO member bodies for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication of, use of, or reliance upon, this ISO/IEC publication or any other IEC, ISO or ISO/IEC publications.
8) Attention is drawn to the normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
9) Attention is drawn to the possibility that some of the elements of this ISO/IEC publication may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

International Standard ISO/IEC 30141 was prepared by subcommittee 41: Internet of Things and related technologies, of ISO/IEC joint technical committee 1: Information technology.

This International Standard has been approved by vote of the member bodies, and the voting results may be obtained from the address given on the second title page.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

IMPORTANT – The 'colour inside' logo on the cover page of this publication indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.

INTRODUCTION

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 areas of ICT. It involves the connecting of Physical Entities (“things”) with IT systems through networks. Foundational to IoT are the electronic devices that interact with the physical world. Sensors collect the information about the physical world, while actuators can act upon Physical Entities. Both sensors and actuators can be in many forms such as thermometers, accelerometers, video cameras, microphones, relays, heaters 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 by providing contextual, real-time and predictive information which has an impact on physical and virtual entities.

IoT can be integrated into existing technologies. Real-time measurements generated by adding sensors to existing technology can improve its functionality and lower the cost of operations (e.g. smart traffic signals can adapt to traffic conditions, lowering congestion and air pollution). The data generated by IoT sensors can support new business models and tailor products and services to the tastes and needs of the customer. In addition to the applications, the technology needs to support supervision and adaptation of the IoT system itself.

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 document 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.

Trustworthiness is recognized as an area of importance, and IoT can leverage current and future best practice. For example, monitoring and analysing deployed IoT systems is essential to maintain reliability and safety and security. Measures such as controlled access can ensure the security of the system.

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 a high level system based reference with subsequent dissection of that model into the four architecture views (functional view, system view, networking view and usage view) from different perspectives.

This document serves as a base from which to develop (specify) context specific IoT architectures and thence actual systems. The contexts can be of different kinds but shall include the business context, the regulatory context and the technological context, e.g. industry verticals, technological requirements and/or nation-specific requirement sets. For more information, see Figure 1.

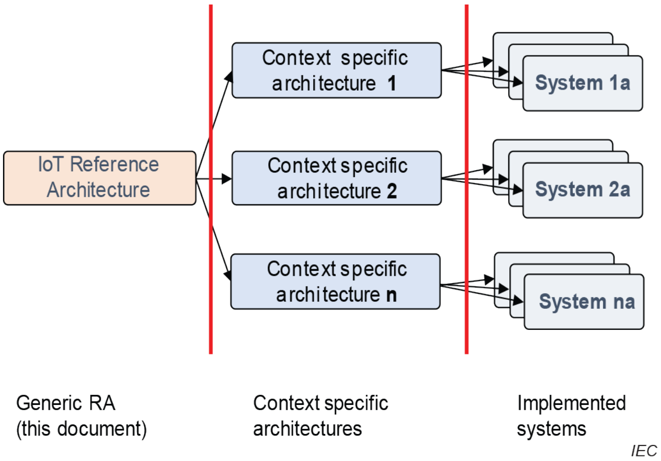

IoT Reference Architecture
Context specific architecture 1
Context specific architecture 2
Context specific architecture n
System 1a
System 2a
System na
Generic RA (this document)
Context specific architectures
Implemented systems
IEC

Figure 1 – From generic Reference Architecture to context specific architecture

INTERNET OF THINGS (IoT) – REFERENCE ARCHITECTURE

1 Scope

This document specifies a 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 (IoT) – Definition and vocabulary*[1]

3 Terms and definitions

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

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/

4 Abbreviated terms

5Vs	volume, velocity, veracity, variability, and variety
API	application programming interface
ASD	Application & Service Domain
BSS	business support systems
CM	Conceptual Model
FQDN	fully qualified domain name
HMI	human machine interface
HTTP	Hypertext Transfer Protocol
HVAC	heating, ventilation and air conditioning
IaaS	infrastructure as a service
ICT	information and communication technologies
IoT	Internet of Things
IoT RA	Internet of Things Reference Architecture
LAN	local area network

[1] Under preparation. Stage at time of publication: ISO/IEC CDV 20924:2018.

LOB	line of business
OMD	Operation & Management Domain
OSS	operational support systems
PaaS	platform as a service
PED	Physical Entity Domain
PII	personally identifiable information
QoS	quality of service
RA	Reference Architecture
RAID	Resource Access & Interchange Domain
RFID	radio-frequency identification
RM	Reference Model
SaaS	software as a service
SCD	Sensing & Controlling Domain
UML	Universal Modelling Language
UD	User Domain
URI	uniform resource identifier
UUID	universally unique identifier

5 Internet of Things Reference Architecture (IoT RA) conformance

To claim conformance, the description of a concrete system architecture as provided by a vendor or system integrator should use the terminology and modelling concepts defined in this document, within the scope of their specific use case.

NOTE A separate conformity guide can be developed that can provide specific guidance to meeting and evaluating conformity to ISO/IEC 30141 to a broader set of entities beyond actual system descriptions.

6 IoT RA goals and objectives

6.1 General

The IoT Reference Architecture (IoT RA) represented in this document describes generic IoT system characteristics, a Conceptual Model, a Reference Model and a number of architectural views aligned with the architecture descriptions defined in ISO/IEC/IEEE 42010. The IoT RA outlines what the overall structured approach for the construction of IoT systems shall be by providing an architectural structure framework. In short, the IoT RA provides guidance for the architect developing an IoT system and aims to give a better understanding of IoT systems to the stakeholders of such systems, including device manufacturers, application developers, customers and users.

This document has the following descriptions:

1) the generic characteristics of IoT systems, outlining the characteristics expected from an IoT system;
2) the Conceptual Model (CM), describing the key concepts characterizing an IoT system;
3) the Reference Model (RM), providing the overall structure of the elements of the architecture;
4) a set of relevant architecture views, describing the architecture from a number of perspectives.

This document supports the following important standardization objectives:

a) to enable the production of a coherent set of standards for IoT;

b) to provide a technology-neutral reference point for defining standards for IoT; and

c) to encourage openness and transparency in the development of a target IoT RA and in the implementation of IoT systems.

Figure 2 illustrates how the IoT RA is derived from a Conceptual Model and a set of characteristics that define a Reference Model and one or more architectural views.

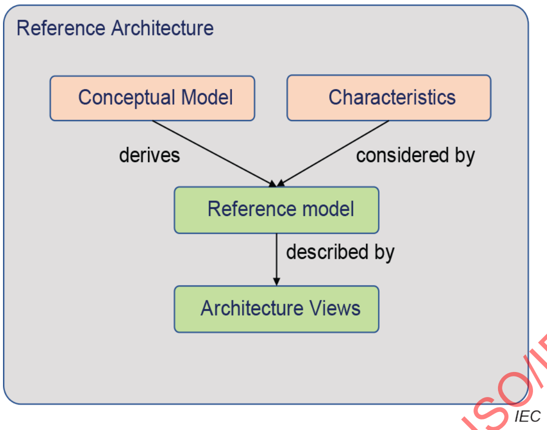


Figure 2 – IoT RA structure

Subclauses 6.2, 6.3 and 6.4 provide a summary of the characteristics, Conceptual Model and Reference Model, respectively.

6.2 Characteristics

The generic characteristics are described in Clause 7, which focuses on a number of key properties that an IoT system typically exhibits. Different application specializations can differ in terms of the actual quantification of these properties, but it is important for an IoT architect to consider how important the respective categories are for the particular system being designed. There are no characteristics that are required of any particular IoT system.

6.3 Conceptual Model

The Conceptual Model (CM) contains a number of vital concepts and describes how they relate to each other logically. Together with the generic characteristics, it provides the background and motivation for the architectural elements discussed in the architectural views in Clause 10. CM is described in Clause 8.

6.4 Reference Model and architecture views

RM is described in Clause 9. The RM and architecture views contain the parts as illustrated in Figure 3.

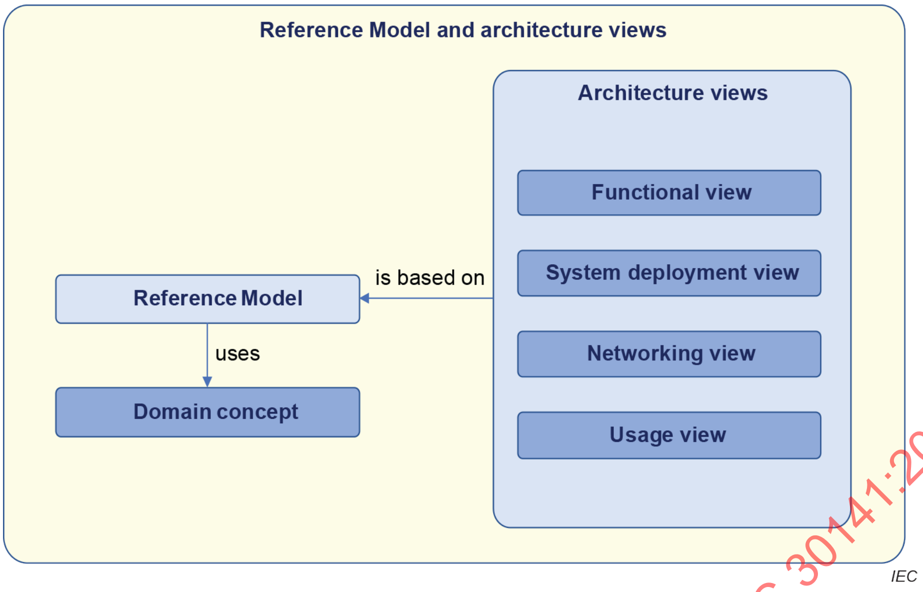


Figure 3 – RM and architecture views

Figure 3 above illustrates the relations between architecture views, Reference Model and domain concept. The domain concept is described in the CM in 8.2.1.1 and 8.2.1.3. Additionally, the RM is founded on the domain concept. A detailed description of domain based RM can be found in Clause 9.2.2.

The respective views are described in Clause 10.

7 Characteristics of IoT systems

7.1 General

Clause 7 provides characteristics of IoT systems. Functions based on all or a part of these characteristics can be implemented in IoT systems. Some of these characteristics are functional, such as network connectivity, while others are non-functional, such as availability and compliance. The characteristics are grouped and summarized in Table 1 and individually explained in 7.2 to 7.4.

Table 1 – Characteristics of IoT systems

Categories	Related characteristics
7.2 IoT system trustworthiness characteristics	7.2.2 Availability
	7.2.3 Confidentiality
	7.2.4 Integrity
	7.2.5 Protection of personally identifiable information
	7.2.6 Reliability
	7.2.7 Resilience
	7.2.8 Safety
7.3 IoT system architecture characteristics	7.3.1 Composability
	7.3.2 Functional and management capability separation
	7.3.3 Heterogeneity
	7.3.4 Highly distributed systems
	7.3.5 Legacy support
	7.3.6 Modularity
	7.3.7 Network connectivity
	7.3.8 Scalability
	7.3.9 Shareability
	7.3.10 Unique identification
	7.3.11 Well-defined components
7.4 IoT system functional characteristics	7.4.1 Accuracy
	7.4.2 Auto-configuration
	7.4.3 Compliance
	7.4.4 Content-awareness
	7.4.5 Context-awareness
	7.4.6 Data characteristics – volume, velocity, veracity, variability and variety
	7.4.7 Discoverability
	7.4.8 Flexibility
	7.4.9 Manageability
	7.4.10 Network communication
	7.4.11 Network management and operation
	7.4.12 Real-time capability
	7.4.13 Self-description
	7.4.14 Service subscription

7.2 IoT system trustworthiness characteristics

7.2.1 General

Trustworthiness is defined in ISO/IEC 20924 as follows:

> "degree of confidence a stakeholder has that the system performs as expected with characteristics including safety, security, privacy, reliability and resilience in the face of environmental disruptions, human errors, system faults and attacks."

Within the scope of this document, security is defined as the combination of availability, confidentiality, and integrity.

7.2.2 Availability

7.2.2.1 Description

Availability is the property of being accessible and usable on demand by an authorized entity.[2] IoT systems can include both human users and service components as "authorized entities".

7.2.2.2 Relevance to IoT systems

In IoT systems, availability can be considered as a characteristic 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 to and 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.

7.2.2.3 Examples

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

7.2.3 Confidentiality

7.2.3.1 Description

Confidentiality is the property that information is not made available or disclosed to unauthorized individuals, entities, or processes.[3]

7.2.3.2 Relevance to IoT systems

In an IoT system, confidentiality protection policies and mechanisms are 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 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 can include PII, e.g. personal health and financial information.

7.2.3.3 Examples

Data flowing through an IoT system could be considered confidential. Confidential data need to be protected from being used for criminal activities, and the inappropriate use of personal data needs to be prevented. For example, IoT motion detection sensors could reveal whether a property is occupied or not, allowing intruders to target the property.

Similar concerns relate to IoT smart meters, where 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.

[2] Source: ISO/IEC 27000:2018, 3.7.

[3] Source: ISO/IEC 27000:2018, 3.10.

7.2.4 Integrity

7.2.4.1 Description

Integrity is the property of accuracy and completeness,[4] usually applied to information within a system.

7.2.4.2 Relevance to IoT systems

Integrity is vital for IoT systems to ensure that the data used for decision-making processes in the system and executable software have not been altered by faulty or unauthorized devices, by malicious actors, or by environmental causes.

7.2.4.3 Examples

In IoT deployments there is a risk that an intermediate device can alter the data and this can have impact on the functioning of the system. For example, an intermediate node can increase the value of the temperature of a room but air-conditioning system can rely on the given setting and not an altered setting from an anomaly.

7.2.5 Protection of personally identifiable information (PII)

7.2.5.1 Description

The concept of privacy overlaps, but does not completely coincide with, the concept of protection of PII. For IoT systems, pertinent entities can include people, technology and processes.

The term PII is used by various policies, rules, laws, and regulations that have their own scope and interpretation of the term. This document uses term PII as defined by ISO/IEC 27018:2014, 3.2:

> "any information that (a) can be used to identify the PII principal to whom such information relates, or (b) is or might be directly or indirectly linked to a PII principal"

Protection of PII is a legal or regulatory requirement in most jurisdictions whenever an IoT system involves personally identifiable information anywhere in its operation. Sensitivity extends to all PII from which sensitive PII can be derived, whether through aggregation, analysis or other means. Protection of PII is governed by a number of principles including, but not limited to, consent and choice; purpose legitimacy and specification; collection limitation; and data minimization. The principle of data minimization requires that organizations process only the minimally necessary PII for the identified purposes. PII should be securely deleted when no longer required.

Protection of PII is a general requirement and is governed by a series of principles which are described in ISO/IEC 29100:

1) consent and choice;
2) purpose legitimacy and specification;
3) collection limitation;
4) data minimization;
5) use, retention and disclosure limitation;
6) accuracy and quality;
7) openness, transparency and notice;

[4] Source: ISO/IEC 27000:2018, 3.36.

8) individual participation and access;

9) accountability;

10) information security;

11) privacy compliance.

It is important that these principles are applied in any IoT system that is processing PII.

NOTE ISO/IEC 29100:2011, Clause 5 provides a more detailed discussion of these principles.

7.2.5.2 Relevance to IoT systems

Any IoT system which does collect, receive, process and/or exchange PII 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 jurisdictions. Situations where a user or person can be identified by data analysis or derived by machine learning mechanisms also need to be considered for protection. System owners in many jurisdictions are required to disclose a data breach. In the event of a compromise, it is important to ensure that they can identify the data that was compromised and report to local agencies.

7.2.5.3 Examples

Independent of the IoT application (e.g. wearables, healthcare monitoring, factory and building systems, automotive, energy, or smart home), the IoT system owner and operator acts as the PII controller. In this role, it is important to ensure the necessary access and data protection controls are identified and implemented for the IoT system. This includes ensuring adherence to local laws and regulatory requirements for data collection and destruction, seeking permission for collection of PII, and the creation of processes to notify users and officials of compromises to this information.

7.2.6 Reliability

7.2.6.1 Description

Reliability is a property of consistent, intended behaviour and results.[5] An appropriate level of reliability in capabilities such as communication, service and data management is important to meet system requirements.

The desired level of reliability should always be seen in relation to the risk associated with the IoT system. Therefore, before changing an IoT system, the designer should evaluate the risk introduced by doing so. This can be handled through a risk management process as outlined in ISO 31010.

7.2.6.2 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.

7.2.6.3 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 communication networks is important for ensuring the

[5] Source: ISO/IEC 27000:2018, 3.55

availability and correct operation of IoT systems, particularly in mission-critical use cases or applications.

Medical devices are one potential IoT application area where the specifications for mean time between failures 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.

7.2.7 Resilience

7.2.7.1 Description

Resilience is the ability of an IoT system or its components to adapt and continue to perform their required functions flexibly in the presence of faults and failures and other ad hoc changes without loss of operation and performance level.

7.2.7.2 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 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.

7.2.7.3 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, a redundant device is available to assume the master role. Industrial control devices and networks are a typical example of this resiliency approach.

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.

The resilience can be compared to a failsafe mode of a computer, or a limp home mode of a car. The essential functions can still be performed, but full functionality will not be possible.

7.2.8 Safety

7.2.8.1 Description

Safety is the state in which the risk of harm (to persons) or damage is limited to an acceptable level.[6] Risk is the probability of the occurrence of harm combined with 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.

7.2.8.2 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 connect will not be granted if safety requirements have not been met.

[6] Source: ISO 21101:2014, 3.34

Even in contexts where compliance with safety standards is optional or voluntary rather than mandatory, proper consideration of safety factors can 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.

7.2.8.3 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 regulations for specific applications such as fire safety, national border safety, and radiation damage monitoring for systems commonly found in hospitals and atomic research centres.

7.3 IoT system architecture characteristics

7.3.1 Composability

7.3.1.1 Description

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

7.3.1.2 Relevance to IoT systems

System integration, interoperability and composability specify how the functional components are assembled to form a complete IoT system, how the functional components connect to each other and which binding mechanisms 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 more stringent requirements than interoperability, in that it requires components not only compatible in their interfaces, but exchangeable with other components of the same kind. At a minimum these components share similar construction and improved characteristics such as timing, performance, scalability and security. When a component is replaced by another of the same kind that is compatible, the overall system functions and characteristics should at a minimum remain unchanged, but consideration can be given to permitting improvements in system functions and characteristics.

7.3.1.3 Examples

One example of composability is a temperature sensor in a building office with well-defined capabilities and a standardized service interface. The temperature sensor can be composed into the building heating, ventilation and air conditioning (HVAC) system for use in controlling the room and building temperature. In the event of a fire emergency in the building, this sensor could also be composed into an emergency response system providing the first responders with data on rooms affected by fire.

A second level of composability (or possibly interoperability) might be an IoT controller 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 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.

7.3.2 Functional and management capability separation

7.3.2.1 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.

7.3.2.2 Relevance to IoT systems

Management capabilities and functional capabilities logically differ as follows:

- purposes (execution/action vs information/description);
- user roles (control and modify behaviour vs transfer or consume facts and information);
- classifications and types of data (technical or system-specific vs personal/sensitive/public);
- access (e.g. an operator can 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 life cycles (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 in capabilities yield differences in risks and hence require distinct security (and other) controls, e.g. retention policy is applicable while dealing with functional data, but might not apply to management data, access control can 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 rendering measures such as physical security controls ineffective. The key value of IoT – the connection of numerous edge components to each other and to IoT service components – increases security concerns, since adding a weak link makes the whole chain weak. Applications and systems previously running in well-protected data centres can 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.

7.3.2.3 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.

7.3.3 Heterogeneity

7.3.3.1 Description

A diverse set of components and Physical Entities of an IoT system that interact in various ways.

7.3.3.2 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 interconnected IoT systems.

7.3.3.3 Examples

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

A second example of heterogeneity is a set of temperature sensors from different manufacturers and with different specifications, integrated into a single system.

There are various industrial communication technologies such as RAPIEnet, EtherCAT, EtherNet/IP, PROFINET, POWERLINK, CC-Link IE, Modbus/TCP, Fieldbus, Profibus, MTConnect, OPC, OPC-UA, OMG DDS, etc., which implies various heterogeneous combinations of communication in connected manufacturing equipment, monitoring, and control devices. A factory can consist of multiple production lines which are expanded as business revenues increase. This gradual expansion can increase the diversity of communication endpoints, and can ultimately increase the heterogeneity of the whole system.

7.3.4 Highly distributed systems

7.3.4.1 Description

Distributed systems are systems which, while being functionally integrated, consist of sub-systems which can be physically separated and remotely located from one another. These sub-systems are normally connected by a communication link (e.g. data bus).[7]

Note that the highly distributed systems are not necessarily stationary over time. Some systems, like RFID tracking, have a high degree of mobility for individual devices, leading to an ever-diversifying topology.

7.3.4.2 Relevance to IoT systems

IoT systems can span whole buildings, whole cities and even the globe. Wide distribution can also apply to data – which can be stored at the edge of the network, centrally, or both. Distribution can also apply to processing – some processing takes place centrally (in cloud services), but processing can also take place at the edge of the network, either in the IoT gateways or even within (more capable types of) sensors and actuators.

7.3.4.3 Examples

For Industry 4.0, production can comprise smart manufacturing systems with distributed assembly lines. These lines are stretched across multiple factories and closely integrated with remotely-located third party suppliers, logistics companies, market providers and customers, which is represented as horizontal integration.

7.3.5 Legacy support

7.3.5.1 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, protocol, device, system, component, technology, or standard that is outdated but which is still in current use can be incorporated into an IoT system.

7.3.5.2 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

[7] Source: ISO 3511-4:1985, 2.5.

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 should be taken when integrating legacy components to ensure that security and other essential performance and functional requirements are met. Legacy components can 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 any given system. This is particularly crucial for IoT, where different connected devices can have very different life cycles and update schedules, often in concert with those of the physical and information systems they are integrated with.

7.3.5.3 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, can be a complex topic. This transition is especially pertinent to IoT, given the enormous number of devices that are destined to be connected.

Many existing standards and application environments still assume and depend on IPv4; yet it is clear that continuing to use IPv4 forever is not a viable strategy. Contributing factors include the lack of sufficient address space, as well as the masquerading of multiple addresses behind a single IP address. The timing of a shift to IPv6 is organization-specific and depends on multiple factors.

7.3.6 Modularity

7.3.6.1 Description

Modularity is a property of components that can be combined to form larger systems of components. Modular components can be removed cleanly from a system and replaced with a module of similar size and with similar physical and logical interfaces.

7.3.6.2 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.

7.3.6.3 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 application-specific integrated circuit (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.

7.3.7 Network connectivity

7.3.7.1 Description

The concept of IoT entities having the capability of communicating with many other entities over a communications network is a core concept of IoT. This many-to-many relationship enables other IoT characteristics including composability, resilience, shareability, scalability, and discoverability while creating challenges in the areas of security, reliability, manageability, accuracy, real-time capability, privacy, and safety. 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 all information in transit. 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.

7.3.7.2 Relevance to IoT systems

IoT systems rely on the ability to exchange information in a structured manner based upon multiple different but inter-networked connections – all within a physical, wired or wireless network. IoT devices are considered "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 can have capabilities such as quality of service (QoS), resilience, encryption, authentication and authorization.

7.3.7.3 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.

7.3.8 Scalability

7.3.8.1 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.

7.3.8.2 Relevance to IoT systems

IoT systems involve various elements such as devices, networks, services, applications, users, stored data, data traffic, and 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.

7.3.8.3 Examples

One example of scalability is an increase in the number of sensor devices attached to an IoT system over a specific period of time. 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.

7.3.9 Shareability

7.3.9.1 Description

Shareability is the capacity of an individual component to be accessed and its resources allocated communally between multiple interconnected systems.

7.3.9.2 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 functionality or outputs of components can be shared among multiple systems.

7.3.9.3 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.

7.3.10 Unique identification

7.3.10.1 Description

Unique identification is the characteristic of an IoT system to unambiguously and repeatably associate the entities within the system with an individual name, code, symbol, or number, and to interact with the entities, or trace or control their activities, by referencing that name, code, symbol or number. These entities include the components of the IoT system itself, such as software components, sensors, actuators, and network components.

7.3.10.2 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 within a given context so that IoT systems can appropriately monitor and communicate with specific entities. Some devices can be hidden behind IoT gateways, or information consolidated to protect privacy (e.g. see ISO/IEC 15045-1). A variety of identification schemes can be supported in specific implementations of IoT systems to meet the application requirements.

7.3.10.3 Examples

IPv4 address, IPv6 address, MAC address, URI, and FQDNs are used as unique, unambiguous identification of network endpoints in Internet applications. Individual hardware devices, software, and other entities can have unique manufacturer’s IDs, object identifiers, universally unique identifiers (OIDs, UUIDs) or other identifiers which similarly allow unique, unambiguous identification.

Physical Entities are often given labels in the form of radio frequency identification (RFID) tags, barcodes and their equivalents. These carriers can contain encoded identifiers that can be sensed by an IoT device. For humans, biometric information can be used to provide unique identification.

7.3.11 Well-defined components

7.3.11.1 Description

IoT entities are considered 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.

7.3.11.2 Relevance to IoT systems

Many components are used to assemble an IoT system. They are typically discovered through an information system interface and the metadata associated with the component cannot be available because the component does not follow established standards or is incapable of storing metadata. 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.

7.3.11.3 Examples

An example of 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 offered 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.

7.4 IoT system functional characteristics

7.4.1 Accuracy

7.4.1.1 Description

Accuracy is a characteristic of various elements in an IoT system:

Sensors make measurements on properties of the physical world. Accuracy of these sensors is the closeness of agreement between the measured values and the actual values of those properties.

Software services can make calculations based on input data. Consider some automated image processing software – examples include number plate recognition for vehicles or facial recognition to establish the identity of people in a scene. Such software can be said to have an accuracy expressed as a percentage with which the recognized number plate text matches the actual number plate text, or the percentage with which the recognized identity of an individual matches the actual identity of the person in the scene.

Actuators operating on the physical world translate digital commands into actions, and accuracy for actuators can be described in terms of the closeness of the actual physical world actions to those intended by the digital command. An example might be a robot arm, where a digital command intends to move the tip of the robot arm to a particular location in 3D space. Accuracy can be expressed by how closely the position of the actual tip of the robot arm matches that of the digital command.

Therefore, in some cases, accuracy can be expressed in terms of the deviation of a continuous quantity in the digital world from its value in the physical world. In other cases it can be expressed as the percentage of instances in which the digital output of a discrete value correctly matches the expected value (i.e. the percentage of correct values).

7.4.1.2 Relevance to IoT systems

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

7.4.1.3 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, while in a home HVAC context, accuracy to plus or minus two degrees might be adequate.

7.4.2 Auto-configuration

7.4.2.1 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 to conditions and the addition and removal of components such as devices and networks. Auto-configuration needs security and authentication mechanisms to ensure that only authorized components can be auto-configured into the system. Security mechanisms need to be organized appropriately for each market segment.

7.4.2.2 Relevance to IoT systems

Auto-configuration is useful for large-scale IoT systems whose configurations change dynamically over time. Auto-configuration's promotion of faulty component elimination and timely maintenance greatly benefits users with demanding reliability requirements.

7.4.2.3 Examples

Examples of auto-configuring devices and protocols include Dynamic Host Configuration Protocol (DHCP), Zero Configuration Networking (Zeroconf™), Bonjour™, UPnP™ (ISO/IEC 29341 series), etc.[8]

7.4.3 Compliance

7.4.3.1 Description

Compliance is the characteristic of conforming to rules, such as those defined by a law, a regulation, a standard or a policy. 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 can be a range of varying granularity or levels of abstraction at which the regulations are applied or enforced.

7.4.3.2 Relevance to IoT systems

Regulations of relevance to IoT systems can 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.

7.4.3.3 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, 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.

[8] Trademarks are given for the convenience of users of this document and do not constitute an endorsement by ISO or IEC.

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), and others.
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 and HVAC, another layer of regulations might be imposed in the form of building codes in various jurisdictions.

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.

7.4.4 Content-awareness

7.4.4.1 Description

Content-awareness is the property of having sufficient knowledge of the information in an IoT component and its associated metadata. Devices and services with content-awareness are able to adapt interfaces, abstract application data, improve information retrieval precision, discover services, and enable appropriate user interactions.

7.4.4.2 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.

7.4.4.3 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.

7.4.5 Context-awareness

7.4.5.1 Description

Context-awareness is the characteristic of an IoT device, service or system being able to monitor its own environment in which it is operating 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.

7.4.5.2 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.

7.4.5.3 Examples

An example of context-awareness would be location-based services, such as a system in which different services are presented 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 be unlocked. The security policy that governs the doors' access can be enhanced with context. The context here is that the building is currently experiencing an emergency situation 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 authorization.

7.4.6 Data characteristics – volume, velocity, veracity, variability and variety

7.4.6.1 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.

7.4.6.2 Relevance to IoT systems

IoT systems are also expected to generate large amounts of data from diverse locations. The data can be aggregated into centralized locations or it can 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, process and secure the data.

7.4.6.3 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.

7.4.7 Discoverability

7.4.7.1 Description

Discoverability is the characteristic of an endpoint on the network to be found dynamically and for that endpoint to report its services and their capabilities through a query mechanism or self-advertizing mechanism; whichever is suitable for the device in question. The endpoints concerned could be IoT devices, services and applications, or even users. Related discovery services allow endpoints to be located, identified and accessed according to variable criteria, such as geographic location or service type.

7.4.7.2 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, or other relevant factors.

7.4.7.3 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. 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™[9] and Consul.

7.4.8 Flexibility

7.4.8.1 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.

7.4.8.2 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 at the other end.

It is possible to break down the general concept of flexibility into different dimensions.

The system aspect of flexibility in the IoT environment comes from the ability to connect IoT services together in different ways, dynamically and during run time. While the capabilities of each IoT service will not vary, the amount and variety of systems that can be created is extensive.

The IoT component aspect of flexibility is illustrated by the distinction between the following kinds of components:

1) a device which has fixed, nonprogrammable, non-extensible functionality – “hard wired, single purpose”;
2) a device which has fixed hard-wired 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) to 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.

7.4.8.3 Examples

An example of differences regarding flexibility in the context of a sensor device is a thermostat. The simplest devices can only offer simple temperature control and reporting of temperature. More sophisticated and flexible thermostats allow for remote control via

[9] AllJoyn is a trademark of AllSeen Alliance, Inc. This information is given for the convenience of users of this document and does not constitute an endorsement by ISO or IEC.

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.

7.4.9 Manageability

7.4.9.1 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.

7.4.9.2 Relevance to IoT systems

Many IoT devices, networks, and systems operate autonomously. It is important that special care is taken to ensure that such systems remain manageable even when parts of the system malfunction, or otherwise become inaccessible, 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.

7.4.9.3 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, deployment, and operational life cycle of the IoT system.

Additionally, servers that maintain firmware and operating system repositories should be able to authenticate the IoT component and vice versa (i.e. mutual authentication). Updates should be digitally signed to ensure their authenticity and integrity. Updates should be transmitted over a secure channel, where possible.

7.4.10 Network communication

7.4.10.1 Description

IoT systems depend on a broad variety of network 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 can be dedicated to the IoT system or which can 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 particular 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.

7.4.10.2 Relevance to IoT systems

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

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

7.4.10.3 Examples

For network communication, protocols such as OpenFlow™[10], Netconf among others will be used in Software-Defined Networking (SDN) enabled IoT environments.

In a proximity network, IoT devices can be connected by wireless technology, e.g. ISO/IEC/IEEE 8802-15-4 and ISO/IEC/IEEE 8802-11:2012/Amd.2:2014 in communication protocols on physical and data link layers. Data can be transported by IoT specific 6LoWPAN. The IoT devices are then connected to a dedicated or general-purpose access network via a gateway which routes data between the proximity network and the wide area network as necessary.

7.4.11 Network management and operation

7.4.11.1 Description

IoT systems require network management. The form and purpose of network management and operation depend on network type, network ownership, and type of communication taking place over the network. Management is required during the initial configuration and deployment 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 involves control over QoS, dynamic extension of the networks (for new or updated IoT devices), fault handling and security control. Networks also handle dynamic and transitory membership of the network by mobile devices as those devices move into or out of the range of the network.

7.4.11.2 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, do not need to be managed as part of the IoT system, since they are general-purpose networks often operated 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.

7.4.11.3 Examples

Using Industry 4.0 as an example, in the factory, most sensors and controllers installed on a production line are using local networks for communication. Such networks are managed and controlled locally by the factory itself. Field bus protocols such as ProfiBus or ProfiNet can be used in such networks. On the other hand, factories can use cloud services which can be hosted remotely. Communication between the cloud service and the factory can use fixed or mobile networks. Such networks are managed and controlled by a network operator and not by the factory itself. However, factories can use certain interfaces which are provided by network operators to ensure secure and robust communication between the factory and the cloud service.

[10] OpenFlow is a trademark of Open Networking Foundation. This information is given for the convenience of users of this document and does not constitute an endorsement by ISO or IEC.

7.4.12 Real-time capability

7.4.12.1 Description

Real-time capability is a characteristic of 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.[11] Additionally, the system has the ability to perform an action or function, or to call a service within a specified period of time, thereby supporting deterministic operations.

7.4.12.2 Relevance to IoT systems

IoT systems often function in real time; data about events in progress flows in continually and there can be a need to produce timely responses to that stream of events. This can 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.

7.4.12.3 Examples

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

7.4.13 Self-description

7.4.13.1 Description

Self-description is the process by which components of an IoT system list their capabilities in order to inform other IoT components or other IoT systems for the purposes of composition, interoperability, and dynamic discovery. 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. Self-described entities are implicitly “well-defined entities” as described in 7.3.11.

7.4.13.2 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. Self-description is also necessary for mobile devices and for devices that hibernate – both of which join and leave networks on a regular basis.

7.4.13.3 Examples

Example of self-description for an IoT system and protocols: A system which uses Bluetooth®[12] in its proximity networks provides device name and supported service list to each other when connecting.

A system broadcasts its status and supported services and current service level. A component may receive such broadcast information from multiple networks and systems and decides based on that information, which network can provide the best match to the service requirement of the component. Based on that decision, the component connects to the selected network. Wi-Fi and Bluetooth work in that manner.

[11] Source: ISO/IEC/IEEE 24765:2017, 3.3327.

[12] Bluetooth and Wi-Fi are registered trademarks of Bluetooth SIG, Inc. and Wi-Fi Alliance, respectively. This information is given for the convenience of users of this document and does not constitute an endorsement by ISO or IEC.

7.4.14 Service subscription

7.4.14.1 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. Subscribing to a service and building a new IoT application can result in new safety requirements to the system. As the manufacturer of the IoT system during provisioning could not foresee this use case, the responsibility of the safety requirement fulfilment lies with the subscriber.

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.

7.4.14.2 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, it is important that the IoT service provider establishes clear mechanisms for establishing and maintaining the subscriptions.

7.4.14.3 Examples

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

8 IoT Conceptual Model (CM)

8.1 Main purpose

CM provides a common structure and definitions for describing the concepts of, and relationships among, the entities within IoT systems. It aims to 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 do the things and services collaborate via the network?

Subclauses 8.2 to 8.3 describe the CM focusing on the above five points. The models presented here use simplified Unified Modelling Language™ (UML®, [13]). Note that the diagrams in Clause 8 show two different types of relationships between entities: generalization and association. This is explained in more detail in Annex A.

[13] UML is a registered trademark of the Object Management Group. This information is given for the convenience of users of this document and does not constitute an endorsement by ISO or IEC.

8.2 Concepts in the IoT CM

8.2.1 IoT entities and domains

8.2.1.1 General

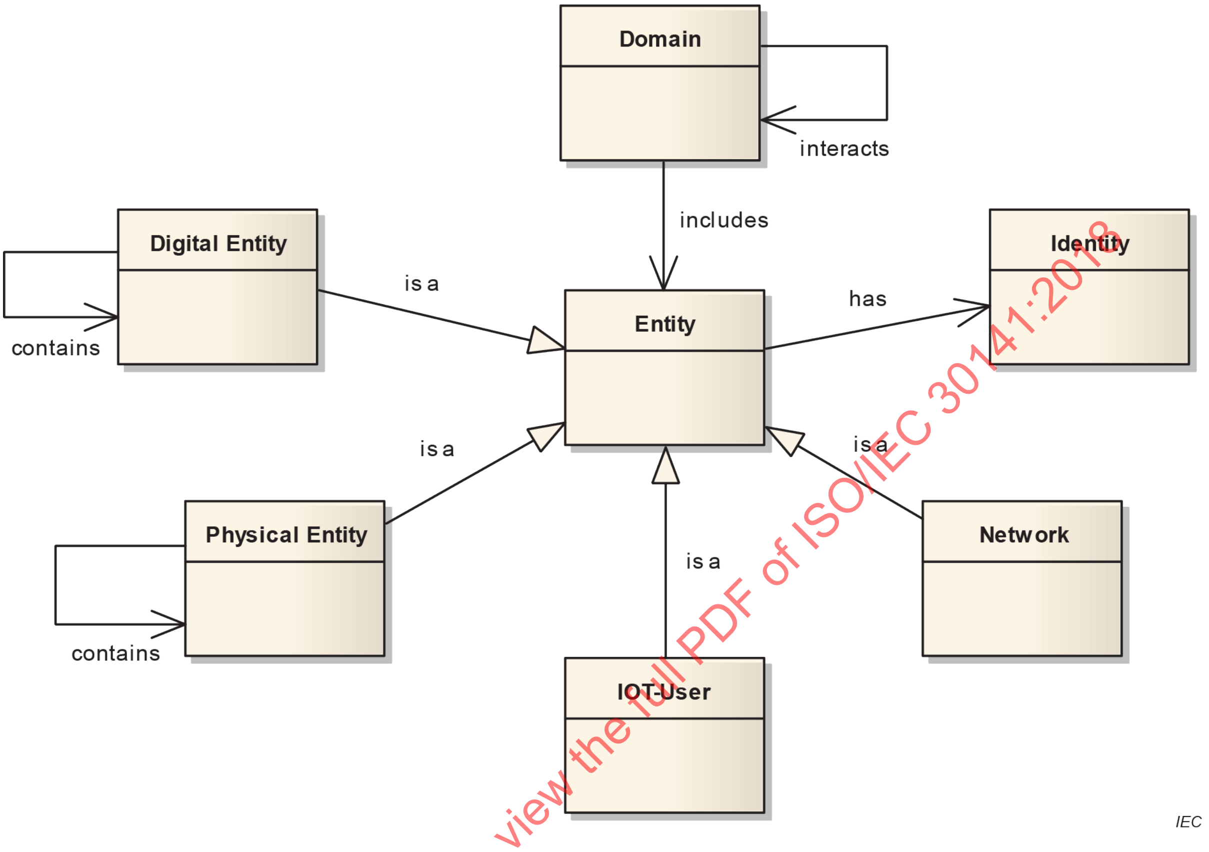


Figure 4 – Entity and domain concepts of the CM

Figure 4 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). Entity relationship tables for the CM are given in Annex B.

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 assist Digital Entities in communicating 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 to group the entities that serve a common purpose such as defined by a domain. Entities from a domain often operate within a sub-system associated with that domain. Sub-systems and entities of a domain can interact with counterparts of another domain. For convenience, it will be said in such cases that these two domains interact with each other. Figure 5 shows that one IoT domain A interacts with another IoT domain B. Of course, one IoT domain can also interact with multiple IoT domains.

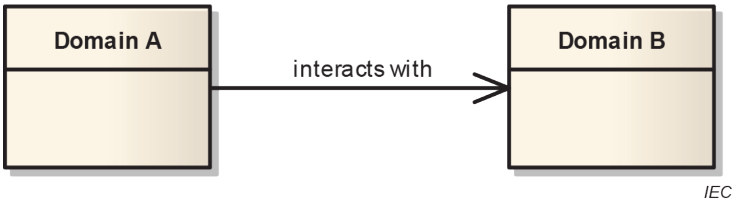


Figure 5 – Domain interactions of the CM

8.2.1.2 Entity

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

8.2.1.3 Domain

A domain is a major functional group of an IoT system. Every entity in an IoT system participates in one or more domains and is said to be included or contained by that domain.

8.2.1.4 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 can contain other Digital Entities.

8.2.1.5 Physical Entity

A Physical Entity is a discrete, identifiable and observable part of the physical environment. Physical Entities can be almost any physical object or environment; from humans or animals to cars; from store or logistics chain items to computers; from electronic appliances to closed or open environments. A Physical Entity is a specialization of entity. A Physical Entity can contain other physical entities.

8.2.1.6 IoT-User

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

8.2.1.7 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.

8.2.2 Identity

8.2.2.1 General

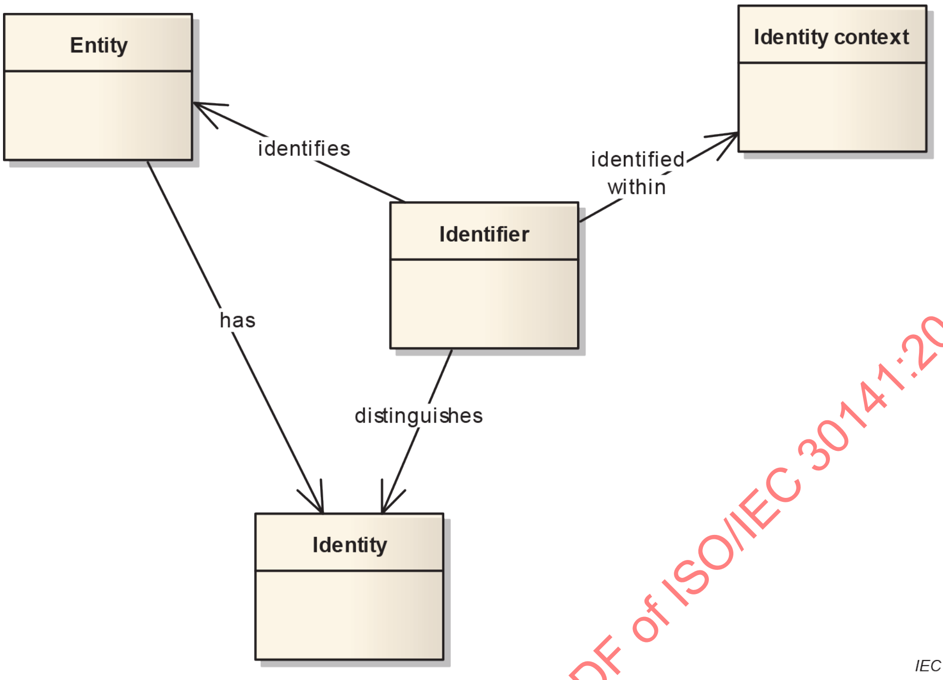


Figure 6 – Identity concept of the CM

Figure 6 shows the identity concept in relation to entities. Most entities in IoT, especially Physical Entities ("things"), have an identity. An identifier is a set of attributes of the entity that can be used to uniquely identify the entity in a 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.

8.2.2.2 Identifier

An identifier identifies an entity. Identifiers distinguish the identity of an entity. An entity can have more than one identifier. Identifiers apply within a given identity context.

8.2.3 Services, network, IoT device and IoT gateway

8.2.3.1 General

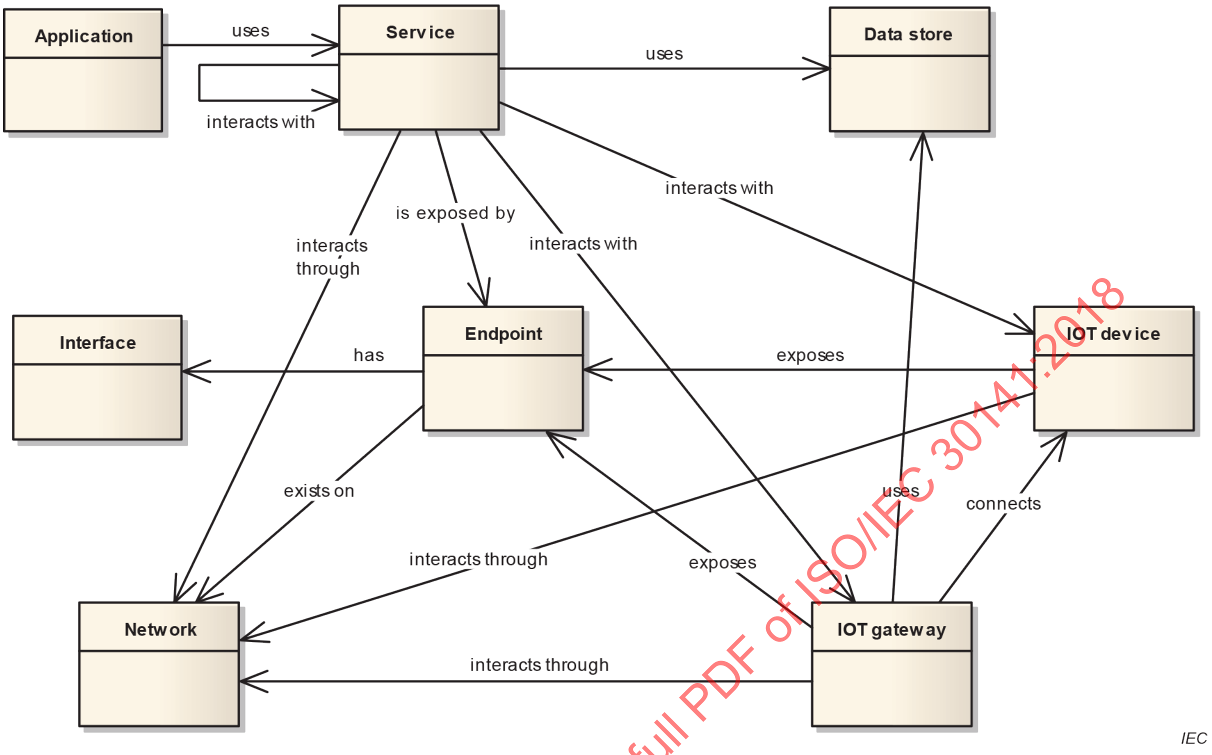


Figure 7 – Service, network, IoT device and IoT gateway concepts of the CM

Figure 7 shows the relationships of services, IoT devices, IoT gateways and the networks that connect them. 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 one 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. Endpoints exist on one or more networks.

Data associated with services, with IoT devices and with IoT gateways can be held in a data store used by one or more entities.

8.2.3.2 Endpoint

An endpoint either implements an interface which can be invoked by other entities, or it connects an entity to the interfaces of other entities. An endpoint can contain more than one interface. An interface is a set of operations and associated parameters which can be used by one Digital Entity to request actions from another Digital Entity.

8.2.3.3 IoT gateway

The key characteristic of IoT gateways is that they form a bridge between different networks: between proximity networks to which IoT devices are attached and access networks (typically wide area networks) which other entities in the IoT system use. IoT gateways can use data stores that are local to the IoT gateway. Gateways can include security functions to protect both endpoints from network-based attacks, or to protect the backhaul networks from malicious or defective endpoints. Such services can be included for either operational or value-added purposes by service providers.

8.2.3.4 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 through sensing and actuating capabilities. Note that an IoT device is a Physical Entity as well as being a Digital Entity – this is significant for some devices in that some of the physical characteristics of the IoT device play a part in its use within an IoT system, such as its location, or its movement and acceleration. An IoT device interacts with one or more networks for the purpose of communicating with other entities. An IoT device has network connectivity and exposes one or more endpoints, and can contain computational capabilities and optionally can use local data stores.

8.2.3.5 Service

A service is a set of distinct capabilities provided through a defined interface. A service can be composed of other services. A service is typically implemented as software. A service defines network interfaces and is 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.

8.2.3.6 Application

An application is a software solution designed to help IoT users perform particular tasks, or to handle particular types of IT problems. For example, automating a business procedure or function [Adapted from ISO/IEC 27034-1:2011, 3.3].

An application can use one or more services. An application can be used by a human user (via a human machine interface (HMI)) or by a digital user (via an API).

8.2.3.7 Interface

An interface is defined as a named set of operations that characterize the behaviour of an entity and is specifically a set of operations and associated parameters which can be used by one Digital Entity to request actions from another Digital Entity.

8.2.3.8 Network

A network is established by connecting IoT devices and an IoT gateway. All pertinent endpoints are accessed through their interfaces.

8.2.3.9 Data store

Data stores are maintained by an IoT gateway and optionally by IoT devices. Data stores hold data relating to IoT systems, which can be data directly derived from IoT devices or can be data resulting from services acting on IoT device data.

8.2.4 IoT-User

8.2.4.1 General

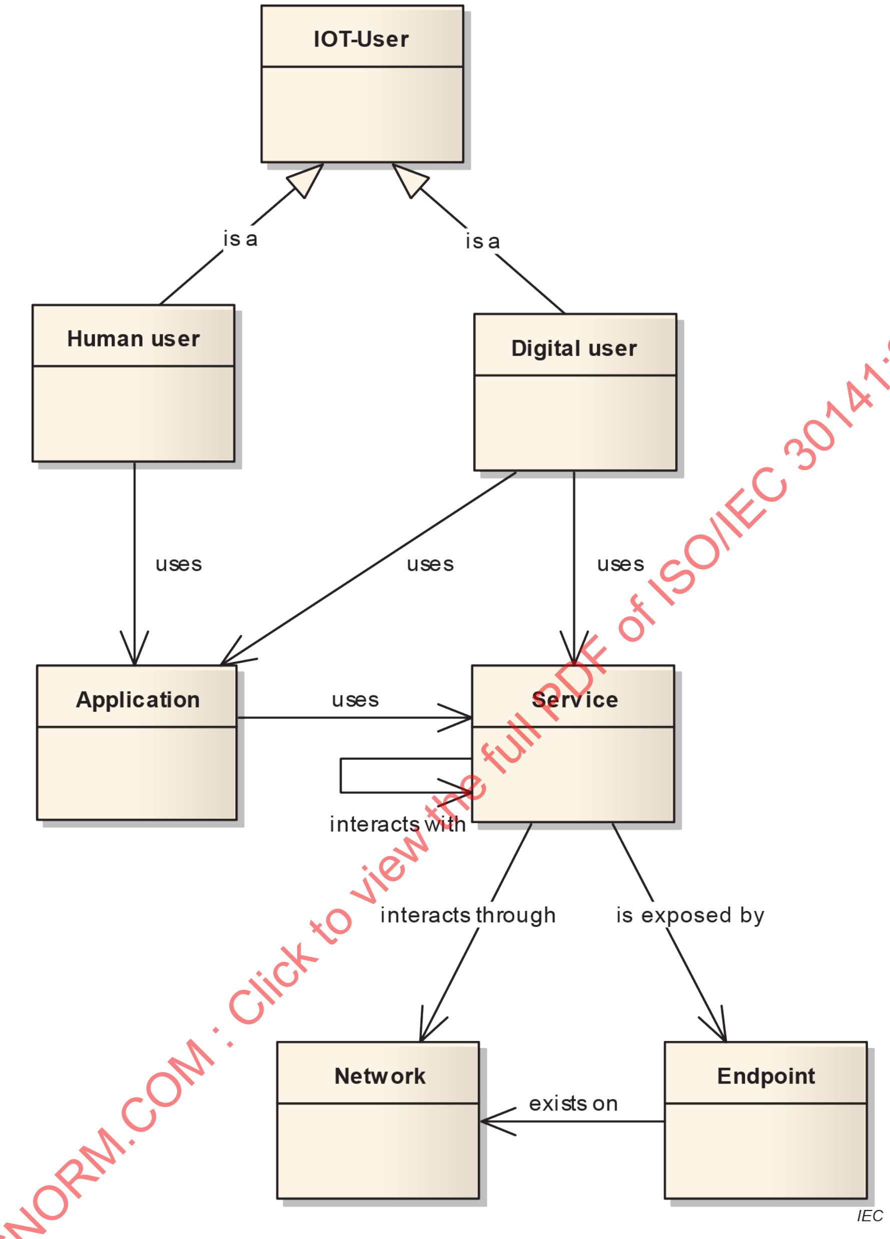


Figure 8 – IoT-User concepts of the CM

As shown in Figure 8, actors of IoT systems are IoT-Users. An IoT-User can be either human (human user) or digital (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 the service endpoint. A human user interacts through one or more applications.

8.2.4.2 Human user

A human user is a 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.

8.2.4.3 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.

8.2.4.4 Application

An application is a software entity or system that offers a collection of functions with which a user can perform a task or achieve a business objective. An application typically uses services. The functions of an application can be presented as services.

8.2.5 Virtual entity, Physical Entity and IoT device

8.2.5.1 General

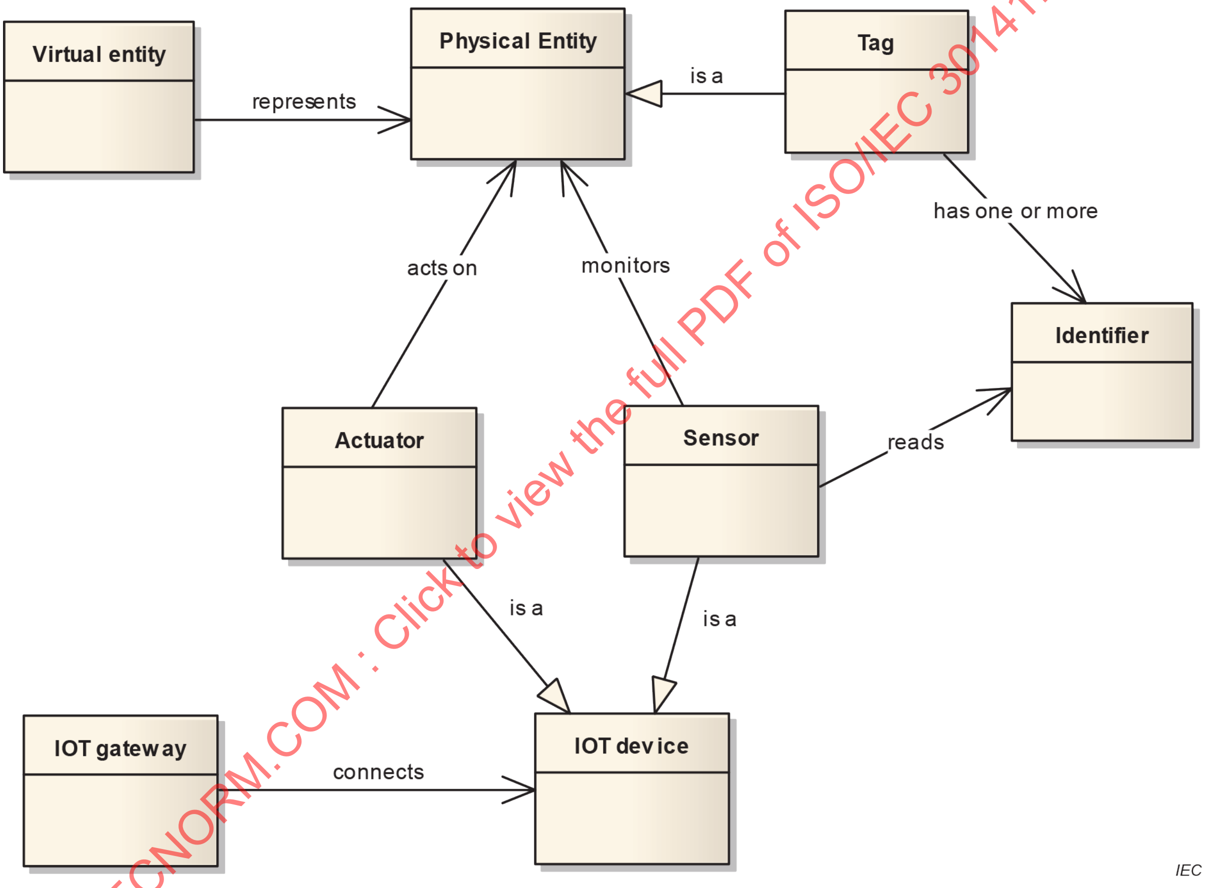


Figure 9 – Virtual entity, Physical Entity, and IoT device concepts of the CM

Figure 9 shows the relationship between virtual entity, Physical Entity and IoT device. Actuators and sensors are IoT devices which have direct or indirect contact with a Physical Entity. Virtual entity, tag, actuator, sensor, identifier, IoT gateway and IoT device are Digital Entities. 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 can have one or more tags attached to it and sensors can monitor the tag rather than the Physical Entity itself. Actuators and sensors are kinds of IoT device, which convert variations in one physical quantity, quantitatively into variations in another. A single IoT device can hold multiple sensors, such as the GPS location sensor and the accelerometer in a smartphone device.

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 can 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.

8.2.5.2 Sensor

A sensor is a device that measures some property of a Physical Entity and outputs digital data representing the measurement that can be transmitted over a network. The sensor's output digital measurement can differ greatly from the original measurement of the actual physical environment, and can be output with a significant time delay, depending on processing of the data within the device. An example would be the recognition of the identity of a person from a surveillance camera device. A sensor is a specialization of an IoT device. A sensor monitors a Physical Entity.

For IoT device, see 8.2.3.4.

8.2.5.3 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 8.2.3.4.

8.2.5.4 Virtual entity

A virtual entity is a digital representation of a Physical Entity, contained within a service. Virtual entity is a Digital Entity. A virtual entity can interact through an endpoint. A virtual entity represents a Physical Entity.

8.2.5.5 Tag

A tag is a Physical Entity that is attached to another Physical Entity in order to assist in identifying and tracking that Physical Entity. Tags can take many forms, some purely passive such as barcodes, others can be more active including RFID tags.

8.3 High level view of CM

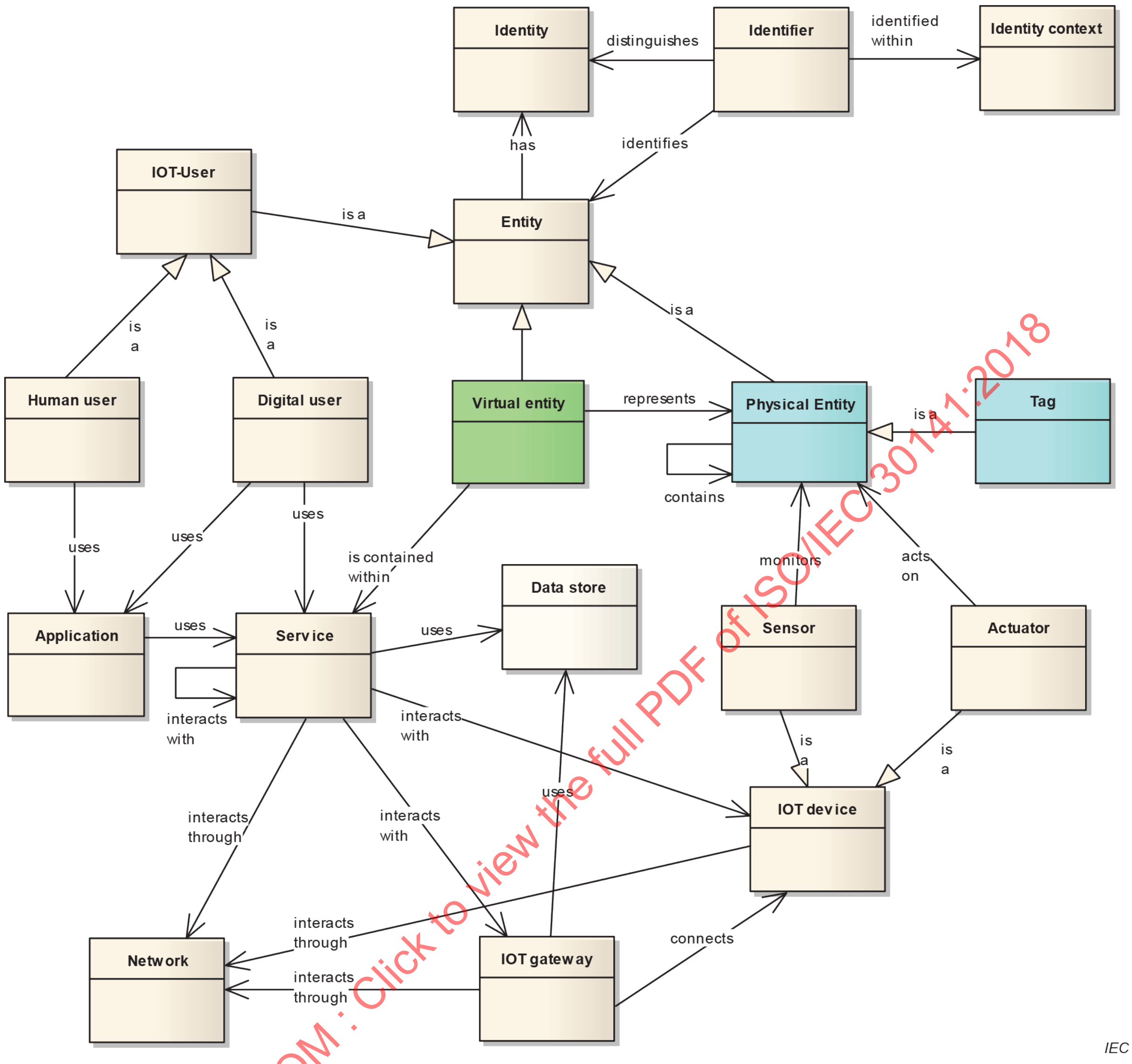


Figure 10 – High level view of CM

Figure 10 provides a high level view of the key IoT concepts contained in the IoT CM, their relationships and their interactions. The diagram is deliberately simplified and does not show all of the IoT concepts which are described in previous figures, where more detail is provided.

An IoT system is a system involving devices which bridge between real-world Physical Entities and Digital Entities, interacting with those Digital Entities via one or more networks over a wide area. All the entities described in Figure 10 and in Clause 8 exist within an IoT system. 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. A digital user consumes services which 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 can 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. A virtual entity is a Digital Entity. Both actuators and sensors are types of IoT devices. 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.

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

9 IoT Reference Model (RM)

9.1 The IoT Reference Model context

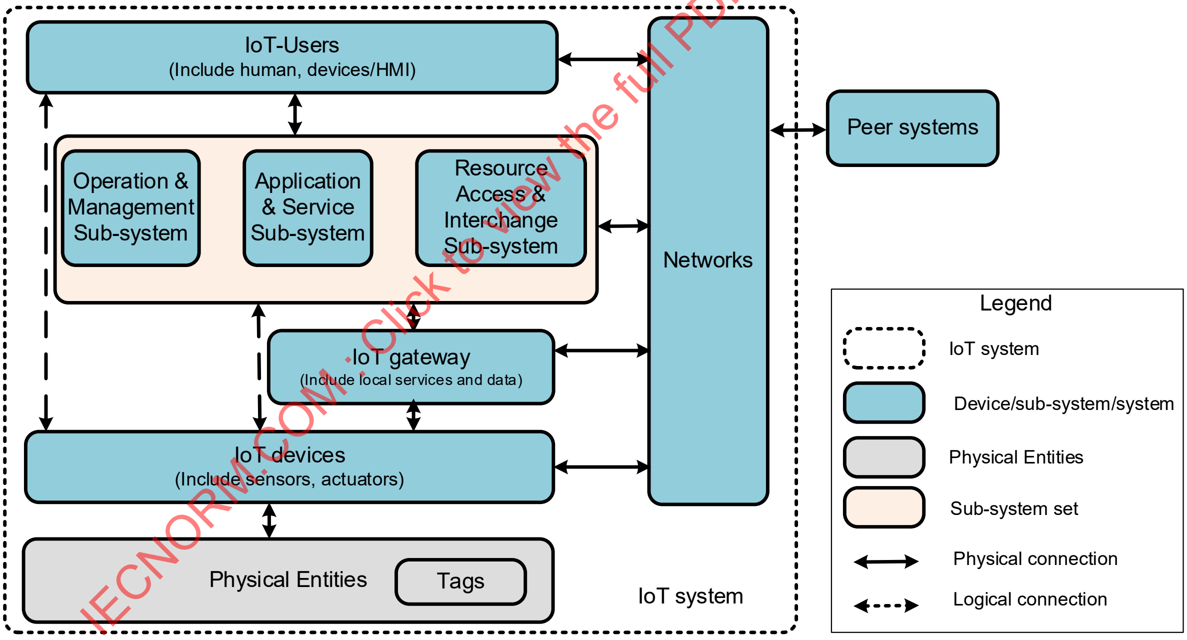
The IoT Reference Model as defined in this document is a part of the overall IoT Reference Architecture as shown in Figure 2.

The Conceptual Model and the characteristics that define the IoT system are covered in Clauses 8 and 7, respectively. Clauses 9 and 10 describe the structure of the IoT RM and the corresponding architecture views, respectively. The relation between CM, RMs and RAs is described in Annex C. The IoT RM is described by entity-based RM and domain-based RM.

9.2 IoT RMs

9.2.1 Entity-based RM

The IoT CM describes IoT at a generic and abstract level. It is helpful to break down the CM into high system level with the help of the domain concept. Clause 9 discusses a composite entity-based RM of the IoT system, and Figure 11 further illustrates the interactions between the major entities using arrowhead lines.


IoT-Users
(Include human, devices/HMI)
Operation & Management Sub-system
Application & Service Sub-system
Resource Access & Interchange Sub-system
IoT gateway
(Include local services and data)
IoT devices
(Include sensors, actuators)
Physical Entities
Tags
Networks
Peer systems
IoT system
Legend
IoT system
Device/sub-system/system
Physical Entities
Sub-system set
Physical connection
Logical connection

IEC

Figure 11 – Entity-based IoT RM

Starting from the entities at the bottom of the diagram, the description of the IoT entity-based RM is as follows.

1) Physical Entities are the real-world things that are sensed and acted upon by the IoT devices.
2) Tags of various types can be attached to Physical Entities to aid in their monitoring and identification.
3) IoT devices interact with the physical world through sensing and actuation. IoT devices include:

a) sensors, which observe a property of a Physical Entity and convert the observation into digital information;

b) actuators, which act on or change some properties of the Physical Entities based on digital instructions.

4) IoT devices communicate via a network. Many IoT devices communicate using a relatively short range and specialized proximity network, due to power and processing limitations. Many other devices are able to communicate using wide area networks such as the Internet.

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 coming 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) Application and service sub-systems 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, for example, monitoring performance. 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) The operation and management sub-system includes 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 and peer systems is provided by the resource access and interchange sub-system, which provides controlled interfaces for service capabilities, for administration capabilities and for business capabilities. The capabilities provided depend on access control capabilities that vary depending on the user, requiring authentication and authorization before the capabilities can be used. Some of the capabilities can be provided through cloud service interfaces by implementations.

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 smartphones, 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. Digital users interact with the IoT systems by means of service APIs controlled by the resource access & interchange sub-system autonomously. Both user devices and digital users communicate with the rest of the IoT system via the user network. For some IoT systems the user devices can interact directly with IoT devices or IoT gateways.

10) Peer systems, which can be other IoT systems or 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.

Based on a study of the decomposition of various IoT systems in different application scenarios, Figure 12 shows the most common IoT entities found in IoT systems. Additionally, this figure provides a very high level relationship between domain and entity.

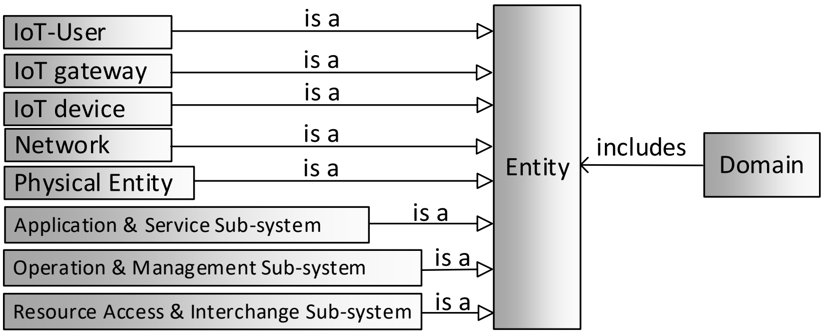

IoT-User
IoT gateway
IoT device
Network
Physical Entity
Application & Service Sub-system
Operation & Management Sub-system
Resource Access & Interchange Sub-system
is a
Entity
includes
Domain

Figure 12 – Domain and entity relationship, and representative conceptual entities in IoT systems

9.2.2 Domain-based RM

9.2.2.1 General

Figure 13 shows the domain representation of the IoT RM. The domains help the designer to focus on the various tasks that have to be performed, by allowing a logical (and sometimes physical) subdivision. Mainly, domains are used to sort functions in areas of responsibility. The identified domains are: User Domain (UD), Operations & Management Domain (OMD), Application & Service Domain (ASD), Resource Access & Interchange Domain (RAID), Sensing & Controlling Domain (SCD), and Physical Entity Domain (PED). Each identified domain is mutually exclusive from all other domains. The relationship between entity-based RM and domain-based RM is given in 9.2.3. From different views, each domain has different kinds of entities, and these entities from different views are introduced in Clause 10.

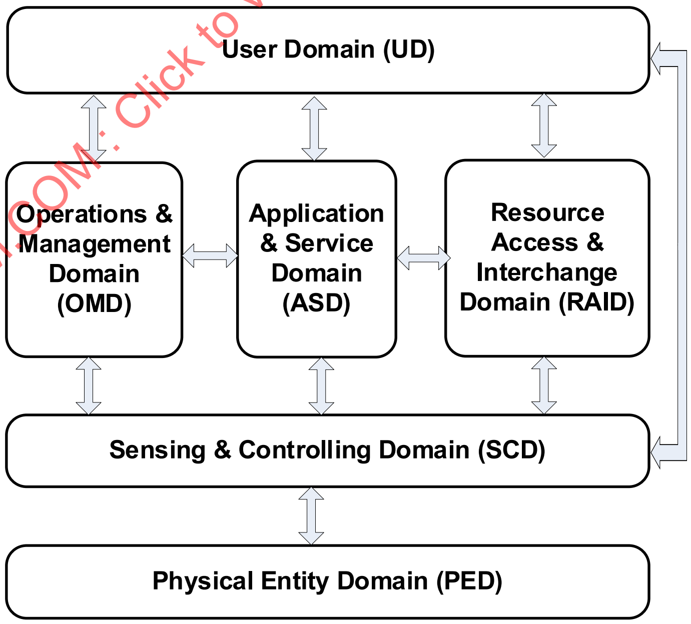

User Domain (UD)
Operations & Management Domain (OMD)
Application & Service Domain (ASD)
Resource Access & Interchange Domain (RAID)
Sensing & Controlling Domain (SCD)
Physical Entity Domain (PED)
IEC

Figure 13 – Domain-based IoT RM

The IoT systems environment extends from the Physical Entity Domain to the User Domain.

The domains and the entities in the domains of the IoT RM interact by means of a set of networks, described in detail in 10.4. The communication networks are not shown in the domain-based IoT RM, but these networks are an essential component of any IoT system.

The network mainly provides pathways for communication and data/information exchange. The key role of the networks is to support and provide communication and data exchange activities and interactions between the entities in the domains and between domains.

9.2.2.2 User Domain (UD)

Users are the actors of the UD. Users include both human users and digital users. Human users interact with services through user devices, through which users access the IoT environment. Such devices can take many forms, including desktop and laptop computers, mobile devices such as smartphones and tablets, and specialized devices tailored for IoT use such as control panels. Digital users interact directly with services through well described interfaces.

9.2.2.3 Physical Entity Domain (PED)

The PED consists of the Physical 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/she may not be an entity in the PED.

9.2.2.4 Sensing & Controlling Domain (SCD)

IoT devices, both sensors, actuators, and complex IoT devices are the actors of the SCD. The SCD consists of the sensors which monitor aspects of the PED and also of the actuators which can act on the PED. The SCD is an essential part of an IoT system in that it bridges between the cyber environment and the real world. The SCD also contains other entities including IoT gateways, local data stores, and local services – particularly control services.

9.2.2.5 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. The OMD typically contains the operation support system (OSS) and business support system (BSS) – the systems by which the IoT system is managed from an operational viewpoint and from a business viewpoint, respectively. The OMD is also responsible for overseeing the secure decommissioning of the IoT system when the time arises.

9.2.2.6 Resource Access & Interchange Domain(RAID)

The RAID provides mechanisms by which external entities can access the capabilities of the IoT system. The main classes of external entities are users (typically interacting via their user devices) and peer systems. The capabilities of the IoT system are offered via one or more service interfaces, with controlled access. The RAID contains the controlled endpoints through which the services are offered – and which services are available depends on the access granted to the particular external entity using the RAID. The underlying capabilities offered by the RAID are implemented by one or more of the other domains – in particular the ASD and the OMD.

9.2.2.7 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.

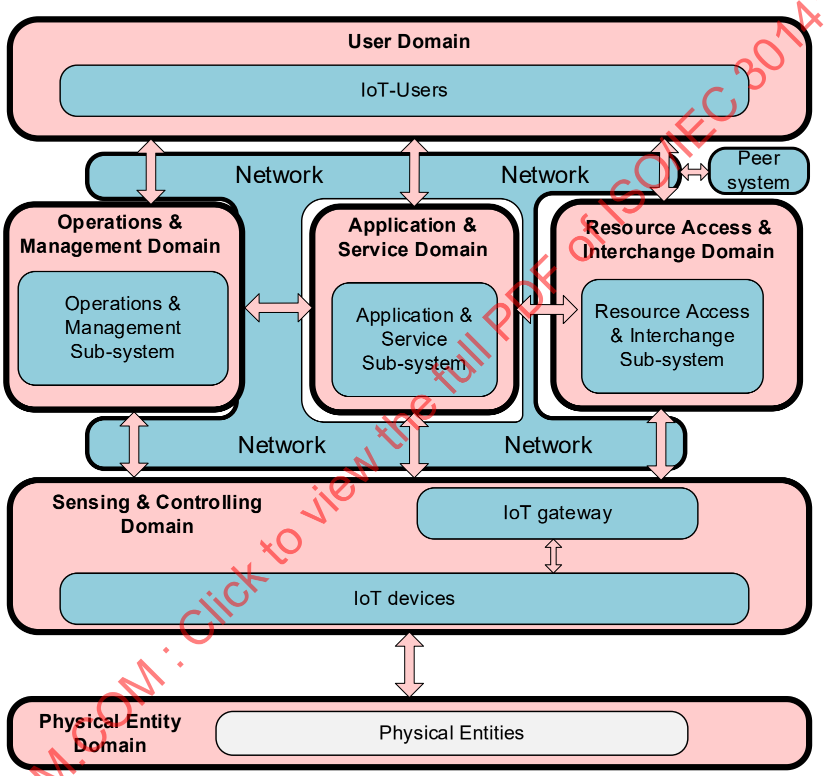
The ASD contains the applications and services offered by the application & service providers. The users in the UD interact with the applications and services to fulfil their requests. The applications and services can also interact with the entities in the SCD (particularly the sensors and actuators) to obtain data or drive actions in the PED.

The applications and services can be provided through cloud services in a certain implementation case and interact with external entities via the RAID, which can involve external organizations such as other IoT and non-IoT systems, governments, law enforcement, financial institutions, and utilities.

The applications and services in the ASD interact with elements in the OMD which are responsible for managing the operational aspects of ASD. The applications and services interact with external entities via the RAID, which can involve external organizations such as other IoT and non-IoT systems, governments, law enforcement, financial institutions, utilities.

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

Taking the entity-based RM in Figure 12 and the domain-based RM in Figure 13, a mapping relation between the two RMs is shown in Figure 14, where these two RMs are consistent with each other.


User Domain
IoT-Users
Network
Network
Peer system
Operations & Management Domain
Operations & Management Sub-system
Application & Service Domain
Application & Service Sub-system
Resource Access & Interchange Domain
Resource Access & Interchange Sub-system
Network
Network
Sensing & Controlling Domain
IoT gateway
IoT devices
Physical Entity Domain
Physical Entities

IEC

Figure 14 – Relation between entity-based RM and domain-based RM

As shown in Figure 14, the relationship between the entities and their domains is as follows. IoT-Users belong to User Domain. Application & Service sub-systems belong to the Application & Service Domain. Operations and Management sub-systems belong to the Operations & Management Domain. The Resource Access & Interchange sub-system belongs to the Resource Access & Interchange Domain. IoT devices and IoT gateway are entities in the Sensing & Controlling Domain. Physical Entities exist in the Physical Entity Domain.

10 IoT Reference Architecture (RA) views

10.1 General description

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

1) IoT RA functional view;

2) IoT RA system deployment view;

3) IoT RA networking view;

4) IoT RA usage view.

The IoT RA is the base when a service-specific system architecture or a target system architecture 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, smart factory and so on.

10.2 IoT RA functional view

10.2.1 General

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

1) intra-domain functions;

2) cross-domain capabilities.

Each functional component is realized by one or more implementations of actual system components, which can be deployed to form a working system. Figure 15 shows the decomposition of the IoT RA functional components. In this figure, there are two parts: domain functions and cross-domain capabilities. The functions and capabilities are not always necessary for certain specific IoT applications. The functions in each domain are very general and optional, depending on specific applications.

10.2.2 Intra-domain functional components

10.2.2.1 General

As shown in Figure 15, the domain functions and cross-domain capabilities are depicted on the left side and the right side, respectively:

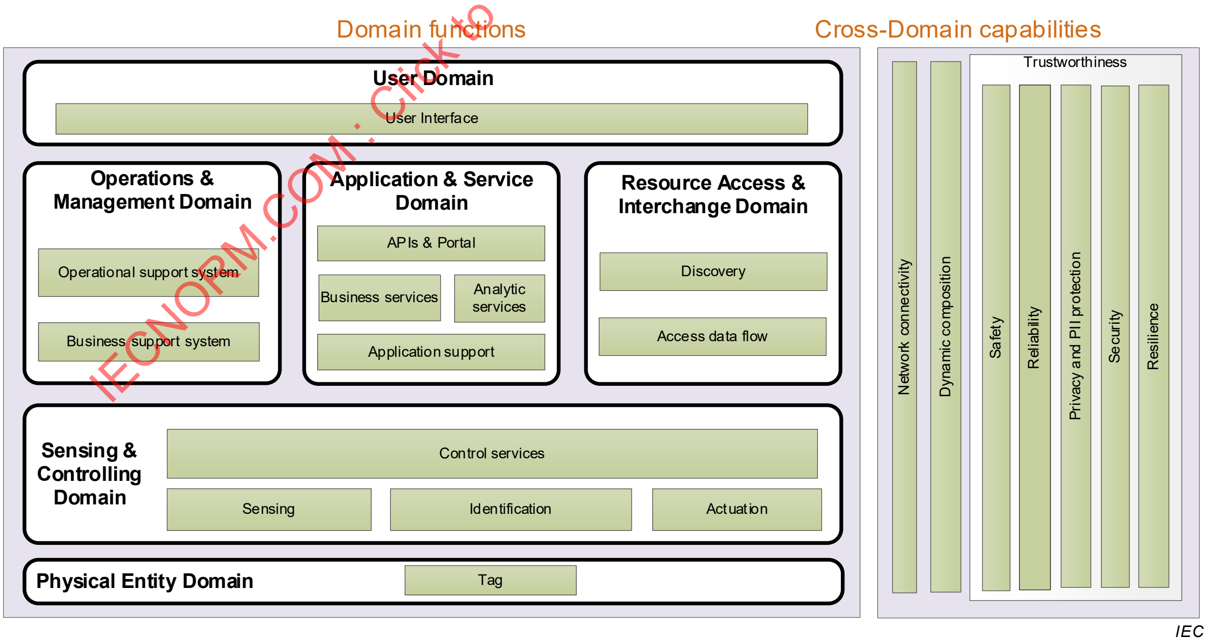


Figure 15 – IoT RA functional view –decomposition of IoT RA functional components

10.2.2.2 Sensing & Controlling Domain (SCD)

The SCD comprises 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. For example, an attention element to tell the sensor what is needed.
2) Actuation is the function that writes data and control signals to an actuator to effect the actuation. Its implementation can span hardware, firmware, device drivers and software elements. It is local in the sense that it closes loops (logically) close to sensors and actuators; they can still be physically co-located with other central resources if resilience and performance requirements can be met.
3) Control services exist within the SCD to control local state – in particular to issue commands to actuators based on input from sensors and other sources. It is common for control services to have real-time behaviour as a result of the need to control dynamic elements in the PED both to ensure appropriate operation of the system and also to ensure safety of operation.
4) Identification is essential for the function of a system which enables the entities to be identifiable, discoverable and traceable, so that the system can distinguish an entity from others. Guidance should note, at a minimum, that:
   a) entities should only be identified when strictly necessary;
   b) only the minimum information necessary to adequately identify the entity should be requested, collected and processed;
   c) a distinction should be made between identifying the device and identifying any associated individual; and
   d) it is possible that identifying information can relate to a natural person, in which case it constitutes PII and should be protected and handled accordingly.

10.2.2.3 Application & Service Domain (ASD)

The ASD represents the collection of functions implementing application and service logic that realizes specific business functionalities for the service providers in the ASD. The ASD contains analytic services, cognitive services, streaming data services, process management services, visualization services, business rules services, control services and application logic. The ASD also contains data stores of various kinds including a device data store, analytics data store, historical data store.

1) Application programming interfaces (APIs) and portal functions: these functions provide controlled ways of accessing IoT system functions, either by digital or human users. Digital users would normally interact through APIs, and human users through access portals. If the users are not part of the trust domain of this IoT system, the access is mediated through the Resource Access & Interchange Domain.
2) Business services: this set of functions assist in creating business process flows, and orchestration of resources to create and manage services.
3) Analytics services: here, gathered data (mainly sensor data streams, but also other context data as well as internal system state) are processed to create insights (real-time events and/or historical data).
4) Application support: this subgroup provides the execution infrastructure that the components deployed in the ASD can utilize to achieve, for example, scalability and configurability. Also, it provides the tools required by a service (and/or application) to do accounting and billing.

10.2.2.4 Operation & Management Domain (OMD)

The OMD contains functional components responsible for the overall management of the IoT system. There are two major functional components, the operational support systems (OSS) and the business support systems (BSS).

The OSS is responsible for the operational management of the IoT system, including provisioning, monitoring and reporting, policy management, service automation, service level management, service catalogues, device registries, device management.

The BSS is responsible for business aspects of the IoT system, including account management, subscription management, billing, accounts, and the product catalogue.

10.2.2.5 Resource Access & Interchange Domain (RAID)

10.2.2.5.1 General

This domain includes all necessary and supportive functions for accessing the IoT system resources – both services and data – or to communicate resources to the IoT system. The access can be in form of service invocation, or data transfer. Accessing or interchanging parties can be internal or external to the IoT system (this distinction can vanish when integrating IoT systems into a larger system).

In this context, the IoT system resources mean any of the capabilities made available by the IoT system. This includes:

1) services of various nature (application, operations and management);
2) data and information from sensors or controllers, including events and notifications;
3) contextual business data;
4) derived knowledge and information generated by applications and services in the IoT system such as intelligence created by analytic processes;
5) metadata about the IoT system and entities with the IoT system;
6) directories and repositories for the above;
7) management and business related capabilities.

Several aspects of interchange and access are usually covered by the above functions.

a) Using data & messaging protocols, not just networking protocols. This includes REST-based protocols, and message-level or data-level IoT protocols.
b) Controlling data flow and event streams, event processing.
c) Semantic descriptions, metadata, taxonomies, supportive of data models and data mappings.
d) Service interfaces and their management.
e) Resource discovery and directory functions.

The RAID supports two main functional groups:

- discovery;
- access data flow.

These functional groups are also called here functional components.

10.2.2.5.2 Discovery

The discovery functional component enables access to appropriate capabilities within the IoT system for external and internal users. Such resources are typically applications, services and data in the ASD, but can also include administration capabilities and business capabilities offered by the OMD. Such resources and related functions include:

1) service endpoint identification & addressing;
2) resource interfaces – in particular for services – and their life cycle management;
3) metadata management and usage;
4) accessing and managing directories and repositories;
5) resource discovery, publishing, search and querying.

10.2.2.5.3 Access data flow

The access data flow functional component has two aspects:

1) It controls all access to the capabilities of the IoT system from external users, which include both IoT users and also peer systems. The access control is responsible for all external endpoints of the IoT system and it organizes the necessary authentication and authorization to ensure that any capabilities of the IoT system can only be used by appropriately authorized users. It includes user management, definition and assignment of roles and groups.
2) It covers all the functions and processes involved in any form of transfer and preparation of any form of data:
   a) For edge data from sensors and controllers: data pre-processing (cleansing, reduction, consolidation, aggregation, transforms, formatting and mappings), and data transfer (data and event streaming, data flow control and routing).
   b) For controls: orchestration and flow of operation controls and commands, dispatching of alarms and notifications, dispatching of executable code to the edge (e.g. fog computing, edge computing), sending configuration data to devices.

10.2.2.6 User Domain (UD)

The User Domain functional components provide user access to the capabilities of the IoT system.

For human users, there are applications which offer user interfaces that enable interaction with capabilities of the IoT system. For machine users, there are APIs through which the capabilities of the IoT system can be invoked over the network.

For human users, the UD also contains the end user devices which support the applications.

10.2.2.7 Physical Entity Domain (PED)

The PED contains the Physical Entities which are of concern to the IoT system.

The functional components that exist in this domain are tags that can be attached to physical objects.

10.2.3 Cross-domain capabilities

10.2.3.1 General

Figure 15 shows a set of cross-domain functional components which span across several domains. The most pervasive of these cross-domain functions is connectivity. In this figure, it shows the functional aspect of trustworthiness (security, privacy and PII protection, safety, reliability and resilience) as cross-domain capabilities. Although Figure 15 emphasizes their functional aspect, the trustworthiness characteristics are defined more broadly as system characteristics (as defined in Clause 7) and have a footprint in all other viewpoints of the RA.

10.2.3.2 Network connectivity

Networks and protocols typically connect entities and sub-systems from more than one functional domain, thus bridging across domains. Networks in an IoT system can be characterized as follows.

- Proximity networking: Proximity networking enables transmission of data from assets or devices on the edge, to entities such as gateways that can process this data for further transmission to entities from other domains, or for enacting controls, such as in edge or fog computing. This networking also enables the control of assets, via actuators or controllers. Proximity networking is often limited to the SCD, but in some occurrences, proximity network and access network are the same network.

- Access networking: This networking function enables the transfer of edge data (from SCD) to application logic (from ASD) or operations logic (from OMD). It also enables the communication of controls to SCD entities, from ASD and OMD entities and sub-systems. It is supportive of management and higher-level communication functions (from RAID).
- Service networking: This networking function connects the applications and services in the ASD, the RAID and the OMD. Service networks are enabling service-based deployment of IoT applications, for example using micro-services and other shared services.
- User networking: This networking function gives both human and digital user entities access and control of ASD and OMD functions. It also enables higher level of integration between different IoT systems as well as with non-IoT systems by supporting user-facing entities of the RAID.

10.2.3.3 Trustworthiness

The trustworthiness characteristics in Clause 7 (safety, privacy and PII protection, security, resilience, and reliability) have a functional footprint that is generally cross-domain.

1) Safety functions in the IoT system ensure the overall safety of the IoT system and also safety relating to specific components within the system.
2) The privacy and PII protection function spans across all elements of the IoT system, wherever PII is created, stored or processed. The PII protection function tracks the existence and location of PII. The PII protection function applies policies to ensure appropriate PII protection is in place, including encryption of PII both at rest and in motion, the application of techniques such as anonymization and aggregation, and the control of access to PII so that only authorized users can access or modify PII.
3) Security functions in the IoT system ensure the confidentiality, availability, integrity, authenticity of information. The IoT RA integrates security policies for IoT components as a 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 life cycle management operations.
4) Resilience functions enable the IoT system to recover operational condition quickly following an incident. Resilience functions are closely related to autonomic computing capabilities of self-healing, self-configuring, self-organizing and self-protecting.
5) Reliability functions ensure that the IoT system, or components within the IoT system, is able to perform its required functions under stated conditions for a specified period of time.

10.2.3.4 Dynamic composition

Dynamic composition functions enable integration and the evolution of the IoT system through the rapid integration of new entities into the system, through the availability of comprehensive metadata and the presence of flexible interfaces.

10.3 IoT RA system deployment view

10.3.1 General

The system deployment 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 deployment view describes it through its implementation components. The system deployment view describes the following aspects:

1) key physical components (e.g. sub-systems, devices, networks) of an IoT system;
2) the general implementation 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.

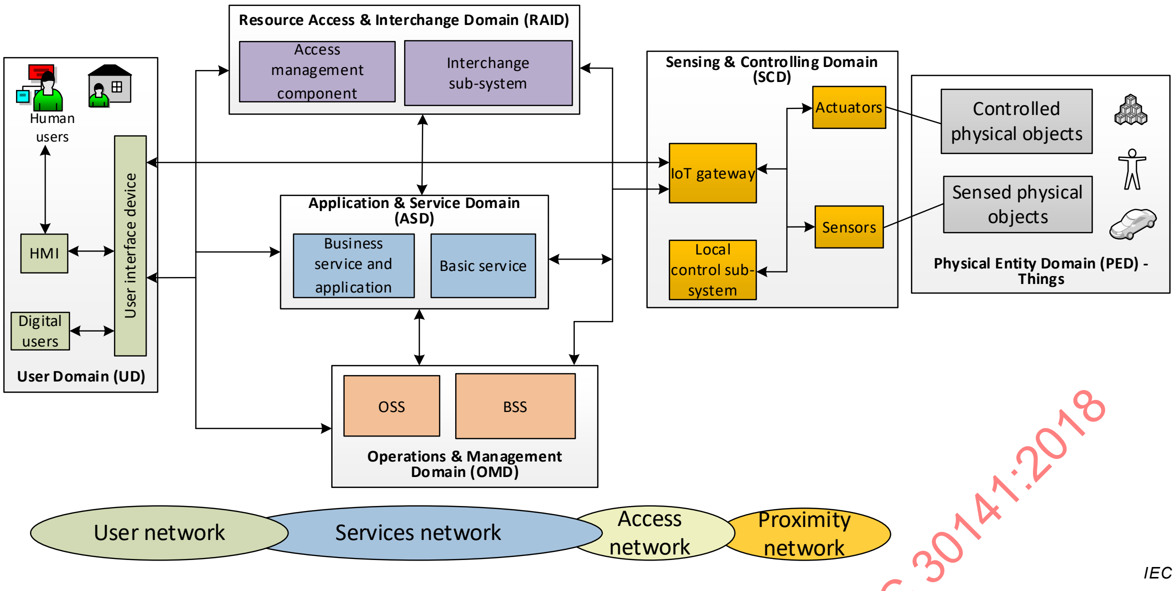

Resource Access & Interchange Domain (RAID)
Access management component
Interchange sub-system
Human users
HMI
Digital users
User interface device
User Domain (UD)
Application & Service Domain (ASD)
Business service and application
Basic service
Sensing & Controlling Domain (SCD)
Actuators
IoT gateway
Sensors
Local control sub-system
Controlled physical objects
Sensed physical objects
Physical Entity Domain (PED) - Things
OSS
BSS
Operations & Management Domain (OMD)
User network
Services network
Access network
Proximity network
IEC

Figure 16 – IoT RA system deployment view

In Figure 16, IoT RA system deployment 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 is given in 10.4.

10.3.2 Systems/sub-systems in Physical Entity Domain (PED)

The PED mainly consists of sensed physical objects and controlled physical objects, which are related to IoT applications and are of interest 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.

10.3.3 Systems/sub-systems in Sensing & Controlling Domain (SCD)

In the SCD, the entities consist primarily of sensors, actuators, and IoT gateways. Sensors sense properties of Physical Entities while actuators change properties of Physical Entities.

Sensors acquire information about a property of a Physical Entity (e.g. physical, chemical, biological properties). Actuators change properties of entities. Both sensors and actuators can interact with Physical Entities 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 integrated with other sensing and controlling devices. The IoT gateway can also perform security functions for constrained IoT devices using the gateway for connectivity to networks.

The SCD might also contain local control systems which are used to run control services, i.e. components for local management of IoT gateway capabilities in scenarios where the IoT gateway is expected to work with or without upstream connectivity.

10.3.4 Systems/sub-systems in Application & Service Domain (ASD)

The purpose of the ASD sub-system is to host the core functions, services and applications that deliver the IoT system functionality to the users (human and/or digital).

The ASD sub-system will provide mainly basic services, including computing services such as data access, data processing, data fusion, data storage, identity resolution, geographic information service and user management, and inventory management.

The ASD sub-system will also host business services and applications built on the generic services, as the ability to host applications will be one of the services provided by the IoT systems.

10.3.5 Systems/sub-systems in Operation & Management Domain (OMD)

The OMD sub-system hosts components responsible for management of IoT devices and control of the operation of the IoT system, to guarantee that the equipment and systems operate safely and reliably. Additionally, it monitors the system to ensure that relevant laws and regulations are not violated.

The OMD contains the operational support system (OSS) and the business support system (BSS).

The OSS is responsible for handling the overall operation of the IoT system and includes capabilities for monitoring and managing all entities of the IoT system over their complete life cycle. The OSS includes compliance systems which enable checking of the IoT system for compliance with laws, regulations and enterprise policies.

A business support system (BSS) is responsible for realization of the business aspects of the IoT system. The business functions include customer relationship management (CRM), subscription management, billing, and payment processing.

10.3.6 Systems/sub-systems in User Domain (UD)

The UD contains both human users and digital users. Digital users are devices of some type and they interact directly with other entities in the IoT system via network interfaces or application programming interfaces. Human users interact using a user device which contains some form of HMI.

HMI sub-system contains the devices and supporting software that allow human users to interact with the IoT system.. Depending on user role, different aspects of the system will be presented for observation and/or control.

10.3.7 Systems/sub-systems in Resource Access & Interchange Domain (RAID)

The RAID contains access management component and interchange sub-system.

Access management component authenticates and authorizes external users of the IoT system wishing to access the capabilities of the IoT system. Note that reverse access management will also be necessary, when the IoT system needs to leverage information and capabilities provided by a partner IoT system.

Interchange sub-system provides exposure of capabilities within the IoT system. Such capabilities include applications, data and services in the ASD as well as administration and business capabilities in the OMD. The latter provide the basis for automation of setting up the trust relationships that provide the authorization data.

10.4 IoT RA networking view

10.4.1 Communications networks

10.4.1.1 General

The IoT RA networking 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 17.

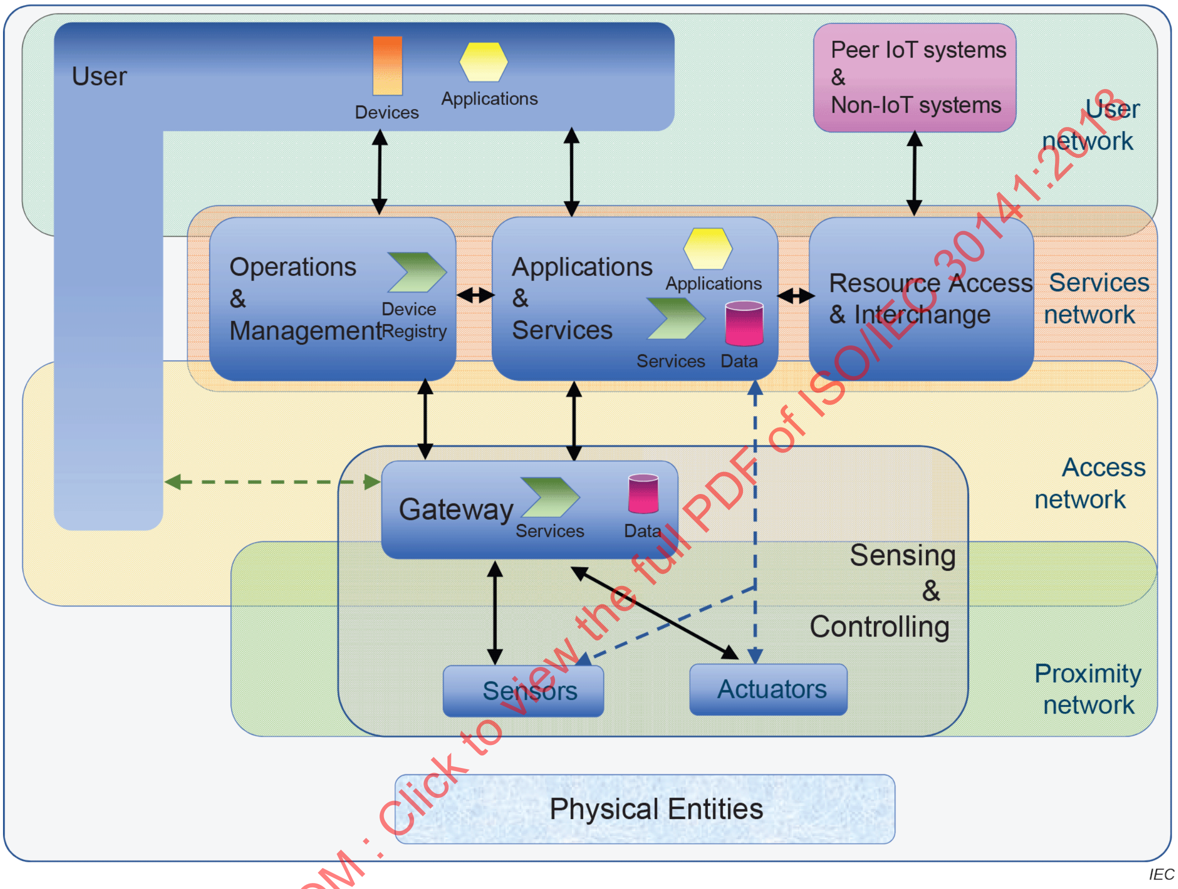


Figure 17 – IoT RA networking view

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 connectivity from one network to another. 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. Business-to-business (B2B) networks are also considered as inter-domain communication networks.

10.4.1.2 Proximity network

This network exists within the Sensing & Controlling Domain. Its main task is to connect sensors and actuators to the IoT system. 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 can use specialized protocols instead of generic protocols such as IP.

It is possible that individual sensors and actuators 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. Examples of proximity networks include IPv6 over Low Power Wireless Personal Area Networks (6LoWPAN), ZigBee®[14], Wireless HART.

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

10.4.1.3 Access network

Access networks are typically wide area networks connecting devices in the SCD to the other domains – the ASD and the OMD. Access networks typically connect to gateways, but when sensors and actuators are more capable and with a limited connection situation, they can connect directly to access networks (dashed lines in Figure 17).

A range of technologies can be used in access networks including wired connections (broadband/ADSL/Fibre) and wireless connections including wireless LANs, mobile (cellular) networks, low power wide area networks, and satellite links (particularly for remote locations). Access networks typically use IP. Access networks can 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.

10.4.1.4 Services network

The services network connects the applications and services in the ASD, the RAID and the OMD, which are typically wired networks within data centres, running IP-based protocols. 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 centre. Where communication spans multiple data centres, a variety of network technologies can be used, including both dedicated connections and Internet connections.

10.4.1.5 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 RAID. This network is typically based on public Internet elements and uses IP. Such networks can use any of the technologies commonly used to carry Internet traffic, including both wired and wireless systems.

10.4.2 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 can use multiple instances of each of these networks to create complete solutions. The key to the interoperability among IoT systems is how data/information is correctly transferred from one type of network to another type of network. One communication network component that will take a vital role to glue the dissimilar communication networks is the gateways specifically designed for IoT, i.e. IoT gateways. Additionally, IoT community's adaptation of the emerging communication networks designed for IoT applications while accommodating the legacy communication networks will also need to promote the interoperability in IoT applications. Network topology is also an important aspect of the IoT systems as the potential IoT systems can use different network topologies to successfully support IoT functionality and capability. The representative

[14] ZigBee is a registered trademark of ZigBee Alliance. This information is given for the convenience of users of this document and does not constitute an endorsement by ISO or IEC.

network topologies are, for example, point-to-point (permanent or switched), bus (linear or distributed), star (extended or distributed), ring, mesh (fully connected or partially connected), ad-hoc, and hybrid (combination of two or more of the topologies above).

In Figure 17, 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 smartphone which contains sensors.

10.5 IoT RA usage view

10.5.1 General description

Whereas the functional view shows the necessary functions and dependencies of the IoT system, the usage 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.

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

10.5.2.1 General

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 18 gives an overview of the three user groups (roles) and their sub-roles. Blue arrows show their interaction when the system is in use. Details about the roles and sub-roles are described in 10.5.2.2, 10.5.2.3 and 10.5.2.4.

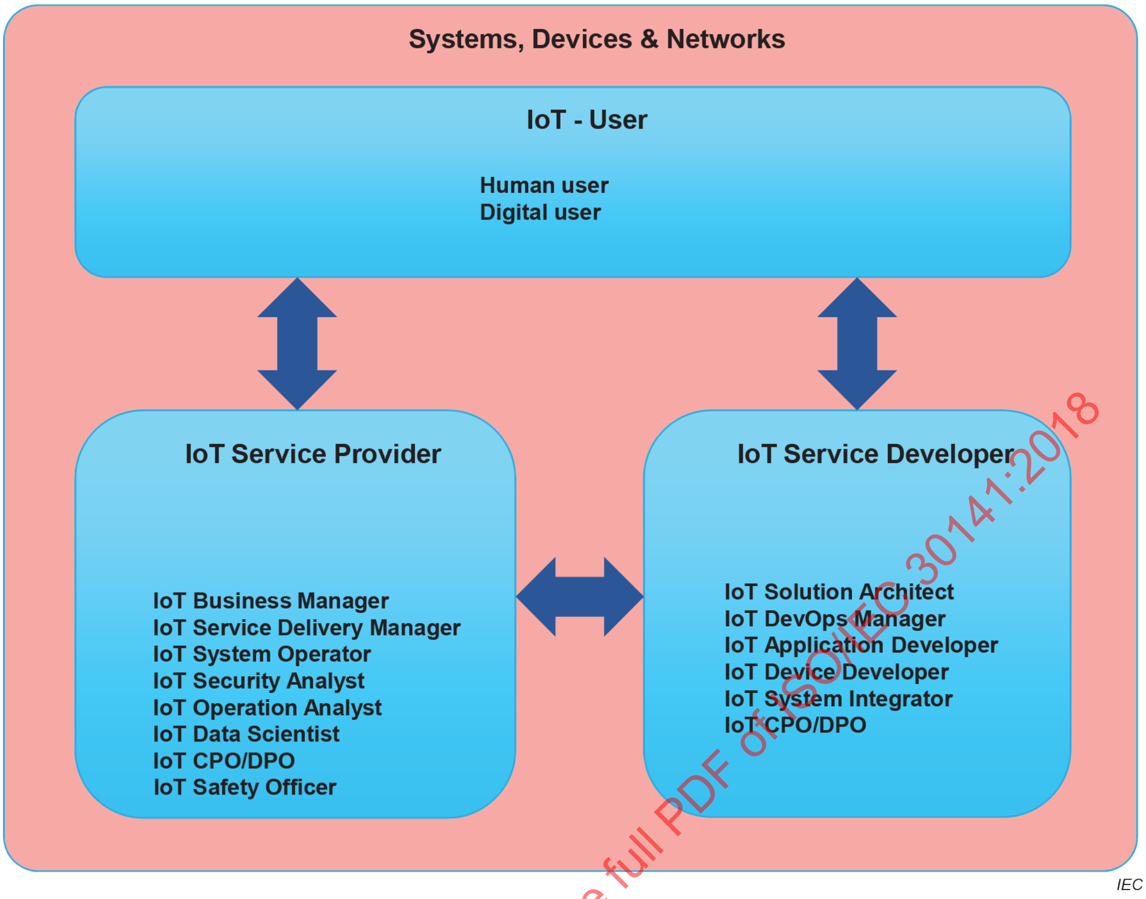


Figure 18 – Roles present when the system is in use

10.5.2.2 IoT service provider

The role of service provider is to manage and to operate IoT services. IoT service providers can also provide network connectivity. Security of this connectivity needs to be addressed and maintained. Additionally, in case of cloud based IoT services, depending on the type of service offered from the data centre (SaaS, PaaS, IaaS), security management, multi-tenancy, tenant security, and separation need be managed at different layers from the hardware up the stack to the application layers. The following sub-roles can be identified.

- An IoT 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 websites and act on solution proposals from the architects.
- An IoT service delivery manager is responsible for a service level agreement between a client and 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 agreement parameters.
- An IoT 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.
- An IoT security analyst mitigates security risks by proactively creating algorithms that detect threats and prevent breaches. They create automatic functions that act on misbehaving IoT devices and users and also ensure compliance through audits.
- An IoT 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.

- An IoT data scientist understands the industry data delivered from devices and the algorithms that provide meaningful analyses. They implement advanced algorithms as services to be used by the LOB analysts and LOB industry applications.
- An IoT chief privacy officer (CPO)/data protection officer (DPO) has several duties that include, but are not limited to, advising the organization of their obligations under relevant privacy/data protection legislation; monitoring the implementation and application of the organization's policies and training on PII protection; monitoring personal data breaches and the response to requests from regulatory authorities.
- An IoT network & infrastructure security manager is responsible for ensuring the operational infrastructure security and connectivity of the networks. This includes, but is not limited to, ensuring availability, integrity, and confidentiality, where applicable, of the networks, system infrastructure, and management of security. The scope of responsibility encompasses sensors and their monitored physical equipment, on premise data centre equipment, public cloud services, connecting infrastructure, tenant access and security aspects of the tenants, amongst others.
- An IoT privacy analyst/privacy engineer considers the overall privacy aspects of the IoT system, including that of the users/consumers of the services offered by the IoT system, with a goal of defending against privacy leakage and regulatory compliance. The IoT privacy analyst/engineer is responsible for designing and evaluating the system from the viewpoint of privacy protection and leverages existing security techniques and architectures to assess the privacy of the services offered by the IoT system.
- An IoT safety officer is responsible for all the safety aspects of the system, including those of various components and sub-systems that comprise the system. This encompasses, but is not limited to, ensuring safety of all users and operators, documenting safety policies and procedures, performing safety inspections, evaluating and implementing safety-related regulatory compliance, and performing accident investigations when necessary.

Figure 19 shows the activities which relate to the sub-roles of IoT service provider.

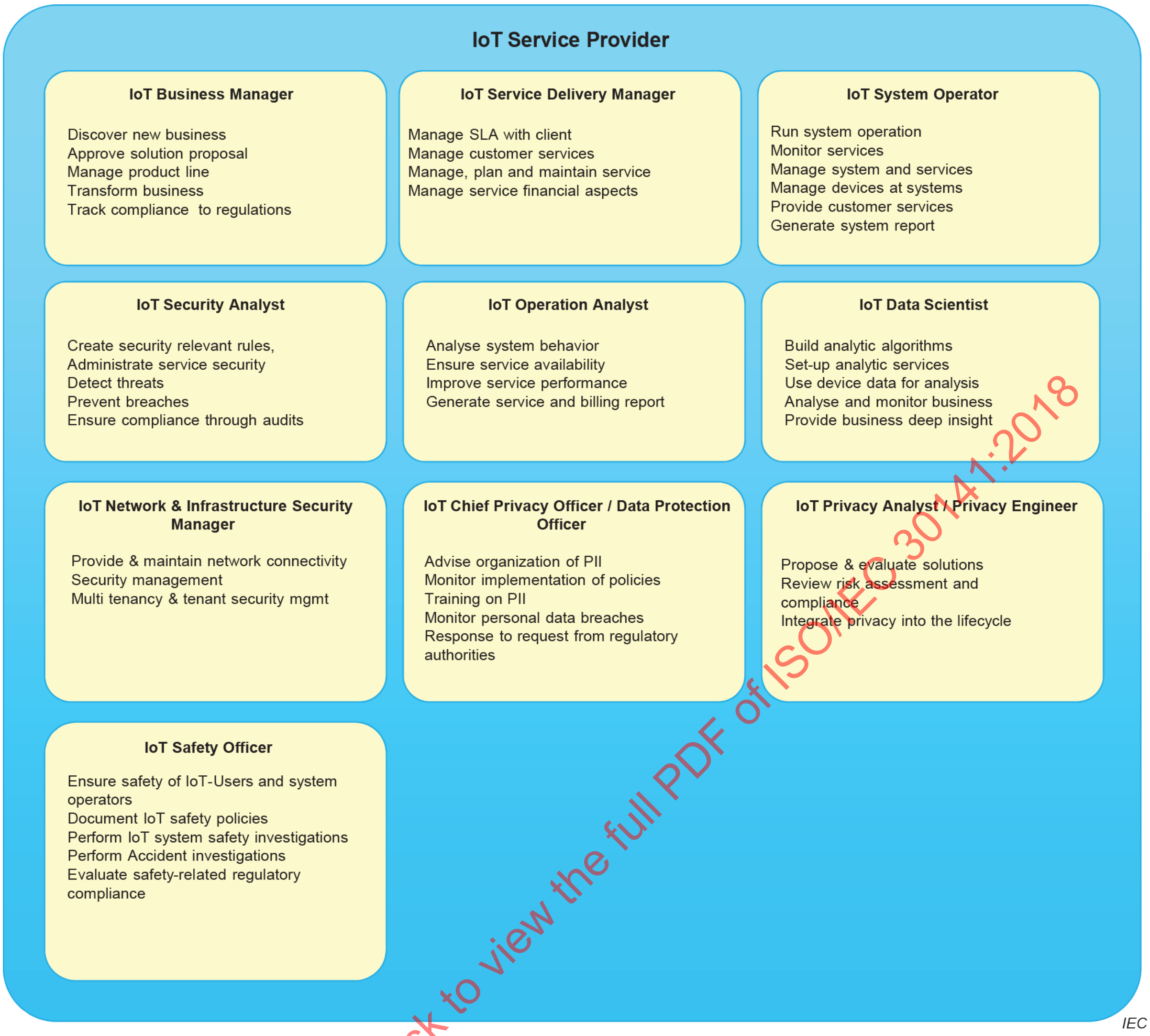

IoT Service Provider
IoT Business Manager
Discover new business
Approve solution proposal
Manage product line
Transform business
Track compliance to regulations
IoT Service Delivery Manager
Manage SLA with client
Manage customer services
Manage, plan and maintain service
Manage service financial aspects
IoT System Operator
Run system operation
Monitor services
Manage system and services
Manage devices at systems
Provide customer services
Generate system report
IoT Security Analyst
Create security relevant rules,
Administrate service security
Detect threats
Prevent breaches
Ensure compliance through audits
IoT Operation Analyst
Analyse system behavior
Ensure service availability
Improve service performance
Generate service and billing report
IoT Data Scientist
Build analytic algorithms
Set-up analytic services
Use device data for analysis
Analyse and monitor business
Provide business deep insight
IoT Network & Infrastructure Security Manager
Provide & maintain network connectivity
Security management
Multi tenancy & tenant security mgmt
IoT Chief Privacy Officer / Data Protection Officer
Advise organization of PII
Monitor implementation of policies
Training on PII
Monitor personal data breaches
Response to request from regulatory authorities
IoT Privacy Analyst / Privacy Engineer
Propose & evaluate solutions
Review risk assessment and compliance
Integrate privacy into the lifecycle
IoT Safety Officer
Ensure safety of IoT-Users and system operators
Document IoT safety policies
Perform IoT system safety investigations
Perform Accident investigations
Evaluate safety-related regulatory compliance
IEC

Figure 19 – IoT service provider sub-roles and activities

10.5.2.3 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.

An IoT 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.

An IoT 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 IoT application developer works in the LOB, in IT or with a third 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.

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

An IoT system integrator tests and integrates IoT services with the IoT enabled platform.

An IoT CPO/DPO has several duties that include, but are not limited to, designing cutting-edge products and services that leverage big data while preserving privacy; proposing and evaluating solutions (e.g. privacy-enhancing technologies) to mitigate privacy risks; conducting privacy-related risk assessments and compliance reviews, responding to incidents; and integrating privacy into the software engineering life cycle phases.

All IoT service developer sub-roles and their activities are shown in Figure 20.

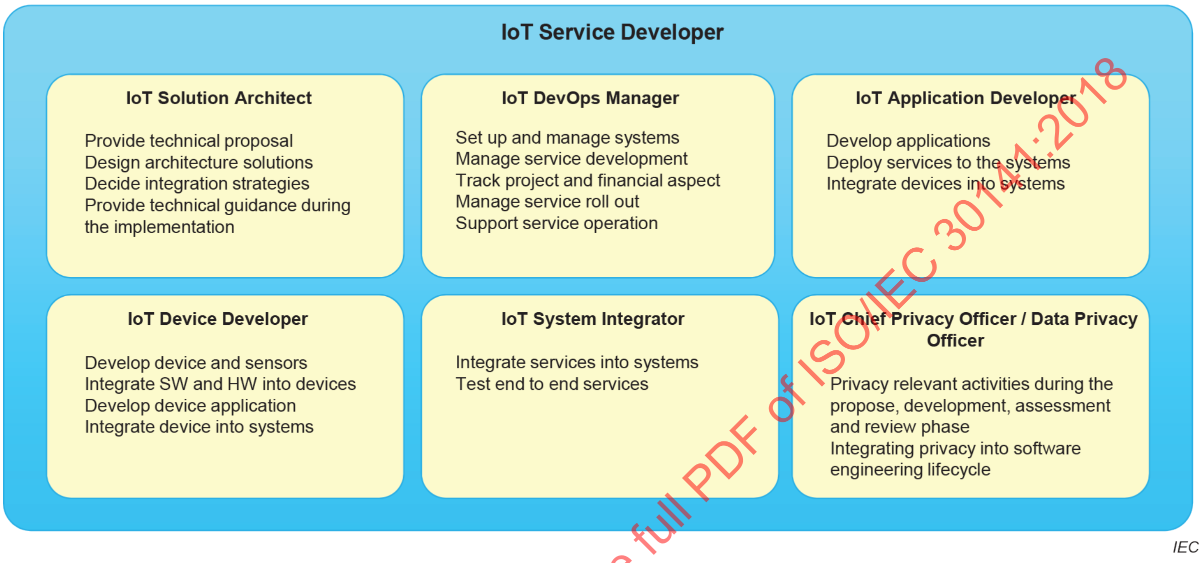

IoT Service Developer
IoT Solution Architect
Provide technical proposal
Design architecture solutions
Decide integration strategies
Provide technical guidance during the implementation
IoT DevOps Manager
Set up and manage systems
Manage service development
Track project and financial aspect
Manage service roll out
Support service operation
IoT Application Developer
Develop applications
Deploy services to the systems
Integrate devices into systems
IoT Device Developer
Develop device and sensors
Integrate SW and HW into devices
Develop device application
Integrate device into systems
IoT System Integrator
Integrate services into systems
Test end to end services
IoT Chief Privacy Officer / Data Privacy Officer
Privacy relevant activities during the propose, development, assessment and review phase
Integrating privacy into software engineering lifecycle
IEC

Figure 20 – IoT service developer sub-roles and activities

10.5.2.4 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 a human user.

All IoT-User sub-roles and their activities are show in Figure 21.

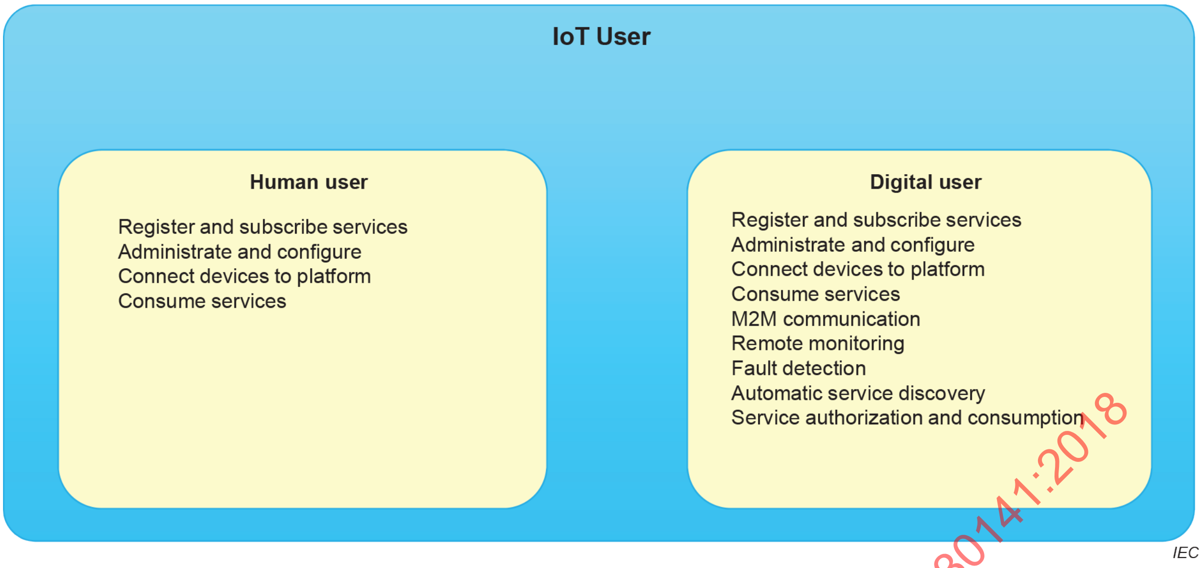

IoT User
Human user
Register and subscribe services
Administrate and configure
Connect devices to platform
Consume services
Digital user
Register and subscribe services
Administrate and configure
Connect devices to platform
Consume services
M2M communication
Remote monitoring
Fault detection
Automatic service discovery
Service authorization and consumption
IEC

Figure 21 – IoT-User sub-roles and activities

10.5.3 Mapping activities, roles and IoT systems in domains

The usage 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 can require interaction among individuals with different roles or skills (see Figure 19).

Table 2 provides an overview of activities and their relevant roles.

Table 2 – Overview of activities and roles

Activities	Roles	IoT Systems in Domains
Device and application development	IoT DevOps manager, IoT device developer, IoT application developer	Application & Service Domain, Sensing & Controlling Domain
Operation of devices, connectivity and applications	IoT system operator, IoT service delivery manager	Operation & Management Domain, Application & Service Domain
Use device data for analytics	IoT data scientist, IoT security analyst, IoT operation analyst	Operation & Management Domain, Access and Communication Domain
Integrate, operate and control data stores and business	IoT solution architect, IoT DevOps manager, IoT system operator, IoT system integrator, IoT service delivery manager	Application & Service Domain, Operation & Management Domain
Use real-time, historic and big data for applications and analytics	IoT data scientist, IoT operation analyst, IoT security analyst, IoT service delivery manager	Application & Service Domain, Operation & Management Domain, Sensing & Controlling Domain, Resource Access & Interchange Domain
Make and operate analytics to run business	IoT data scientist, IoT operation analyst, IoT application developer, IoT DevOps manager	Application & Service Domain, Resource Access & Interchange Domain
Bring in analytics to dashboard	IoT DevOps manager, IoT data scientist, IoT application developer	Application & Service Domain, Operation & Management Domain, Resource Access & Interchange Domain
Monitor system state, act on security risks and breaches	IoT system operator, IoT security analyst	Operation & Management Domain
Track compliance to regulations	IoT business manager, IoT security analyst	Application & Service Domain, User Domain

Figure 22 and Figure 23 show some examples of using IoT systems from different activity perspectives.

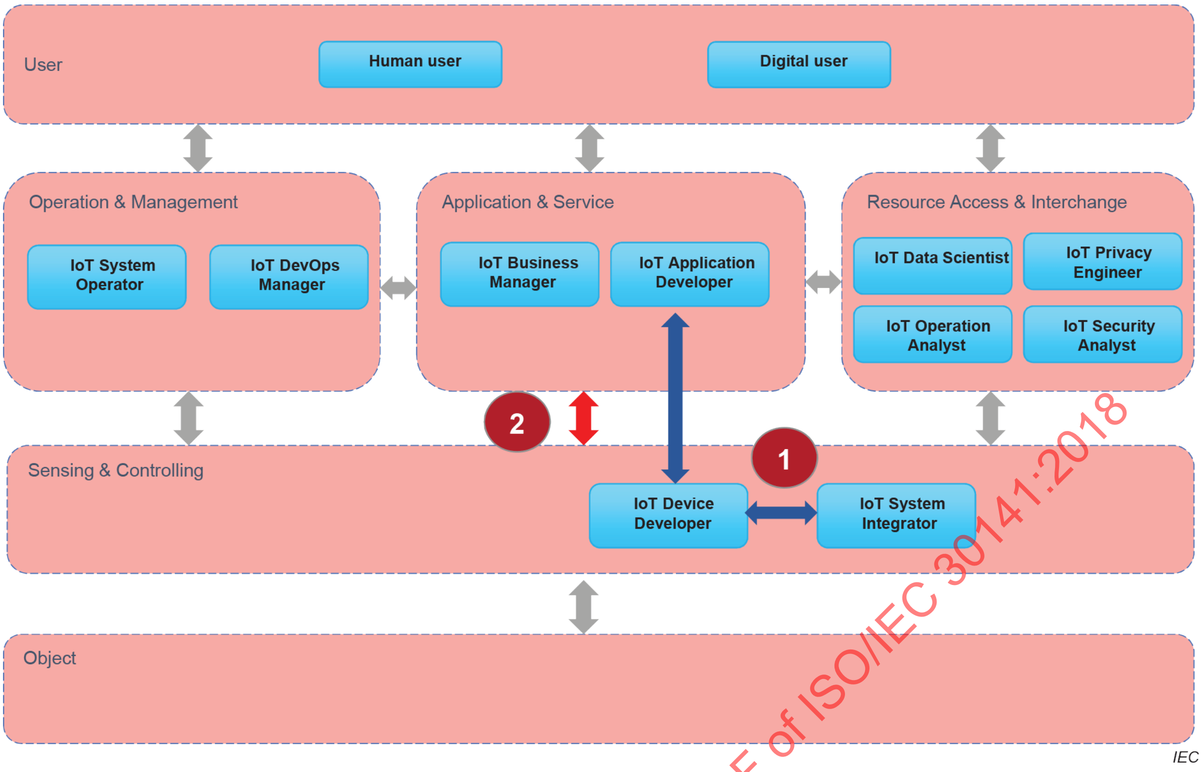


Figure 22 – Activities of device and application development

Figure 22 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 boxes in Figure 22 represent the human users (in this case developers and operators) of IoT systems. The six domains of an IoT system are represented by boxes with dashed lines. 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 Sensing & Controlling Domain will be connected to IoT systems in the Application & Service Domain and end-to-end functions can be tested.

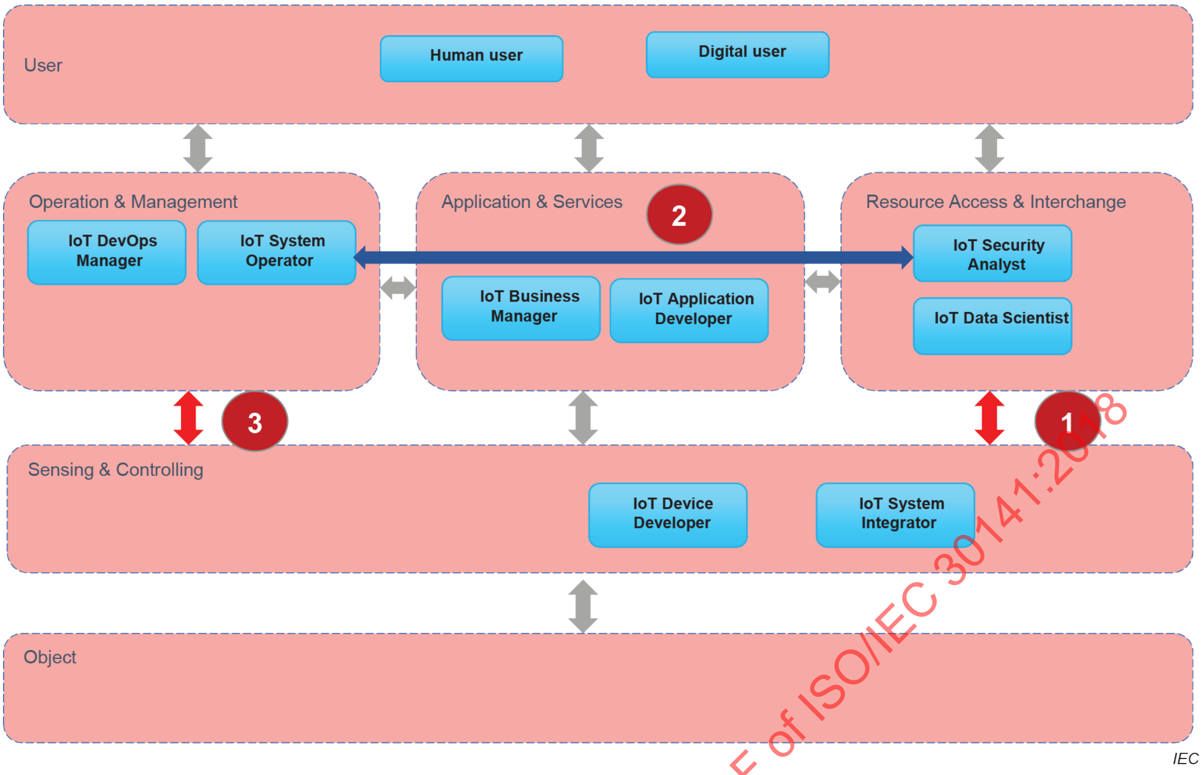


Figure 23 – Using device data for security-related analytics and operations

Figure 23 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 include, but are not limited to:

a) When the device is configured and connected to the communications system, usage data can be sent to the IoT systems in the Resource Access & Interchange Domain. The security analysts and data scientists can use the collected device usage data to perform security-related analyses.

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

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

11 IoT trustworthiness

11.1 General

Trustworthiness of IoT systems covers the aspects of safety, security, privacy and personally identifiable information (PII) protection, resilience and reliability. These aspects are represented as cross-domain functional components in the IoT RA functional view (see 10.2). They are represented in this way to indicate that they are pervasive aspects of IoT systems that have an impact on all domains and on most entities within an IoT system.

Clause 11 describes how safety, security, privacy and PII protection, resilience and reliability apply to IoT systems in the context of the IoT Reference Architecture.

Trustworthiness aspects are described in this specification as system characteristics – or properties. They do not just manifest as cross-system specific functions. Trustworthiness of an entire system can be seen as a set of emergent properties derived from the trustworthiness of its sub-systems and from the overall architectural and functional design. While performing its primary operations and functions, a system might or might not exhibit these properties. Each one of these characteristics has a strong contractual aspect: it is either subject to service level agreements or service level objectives as well as possible policies and regulation. In turn these are subject to measures, audits, and assurance procedures that involve metrics, key performance indicators (KPIs) and targets.

Trustworthiness characteristics need to be taken into account and managed throughout the life cycle of an IoT solution and considered as part of the system as a whole. An integrated approach is necessary for trustworthiness since aspects of trustworthiness not only affect the system individually but also impact each other, either positively or negatively, with tradeoffs based on context and stakeholder needs.

Effectively, each trustworthiness aspect needs to be treated with a "by Design" approach – i.e. "Privacy by Design", "Security by Design" and so on. This means that each aspect of trustworthiness needs to be considered as an essential part of the IoT system from its inception and needs to be built into its design and operation, not added as an afterthought.

Each trustworthiness characteristic demands an appropriate up-front assessment that is used to derive the requirements to be satisfied by the IoT system for that aspect. The assessment usually combines with organizational policies and business goals to complete the requirements. This process should take into account the tradeoffs and relationship with the other trustworthiness characteristics. The detailed controls and mechanisms put in place to satisfy the requirements established by the design process and their continued application during the IoT system life cycle is ensured by the management and operations system.

There is also the need to monitor and measure the IoT system in production with a series of KPIs that are designed and reflected in data gathered from the system as it operates. These can be used to diagnose the operational health of the system.

Finally, IoT systems and their contexts vary a lot from each other. Not every trustworthiness characteristic might be relevant to a particular system. Some will be of primary importance while others will play a supportive role. For example, in many industrial systems safety and reliability will be paramount. Other characteristics will be seen as a means to support these. Different policies and regulations also apply depending on industries and world regions.

As a consequence, metrics and targets for each trustworthiness characteristic can vary greatly from one system to another, across industries, or across regions of the world.

11.2 Safety

Safety is commonly defined as the condition of the system operating without causing unacceptable risk of physical injury or damage to the health of people, either directly, or indirectly as a result of damage to property or to the environment [ISO 10795:2011].

Within safety is also the concept of functional safety. Functional safety focuses on the systems or equipment operating as expected in response to all inputs. It ensures that several aspects are considered that could result in a dangerous condition, leading to harm to person(s), the environment or destruction of property. The goal is to ensure corrective or preventive action that will avoid or reduce the impact of an accident. In several sectors the IEC 61508 series is used to provide proof of conformance to these standards for products in automotive, medical, transportation and manufacturing.

Safety considerations in IoT systems will be given a broader scope than just preventing harm to humans: it will encompass preventing damage to other living beings and the environment, as well as to equipment. While safety to humans is paramount and unconditional, extending it to the environment, to other living beings and to equipment can be subject to conditions relative to the objectives of the IoT system and acceptable risks, trade-offs, and other mitigation options.

Safety is a major concern of many IoT systems. This is because IoT systems interact with the physical world and cause effects in the physical world. The effects caused by IoT systems have the potential to cause harm, both to human beings and other living creatures and also to the environment, equipment and other Physical Entities. This can occur due to explicit actions of the IoT systems, e.g. through control requests sent to actuators. It can also occur due to implicit actions caused for example through recommendations of an IoT system to human users of an IoT application or errors or attacks on the system. There are also issues associated with bad or failing components of the IoT system and how the IoT system detects and deals with such failures.

All parts and subsystems of an IoT system can contribute to safety as safety of an entire system is – like other trustworthiness characteristics – an emergent property. Harm prevention is commonly considered at several levels such as the physical equipment directly controlled by IoT technology, the stress induced by operation conditions for human operators, and the indirect effect on the environment over the longer term, to give some examples.

For actuators, there is a need for a safety system to be in place concerning the operation of the actuators in an IoT system, both as individual entities and also as a combined system.

It is especially important to consider the whole life cycle of the IoT system from a safety perspective and to have processes in place to deal with any changes to the IoT system that take place. For example, the effect on the environment of disposing of equipment or consumables should be considered. Such changes could involve adding to the system or modifying existing parts. Any changes should be evaluated from the safety perspective and appropriate measures taken.

The IEC 61508 series is a basis for developing and operating safe IoT systems, with appropriate safety integrity levels (SILs) applied to the system and to the components within the system. ISO 31000 is one of a number of standards that can assist with safety.

11.3 Security

11.3.1 General

Security in IoT systems covers a variety of concerns, from the information systems involved, to physical assets and products, industrial processes and operational technology (OT) side, and controlling access to equipment.

11.3.2 IoT system Information Security Management System (ISMS)

Information security is a major concern of any ICT system and IoT systems are no exception. IoT systems present particular challenges for information security in that they are highly distributed and involve a large number of diverse entities. This implies that there is a very large attack surface and a significant challenge for the information security management system (ISMS) to apply and maintain appropriate security controls across the whole system. This is a major motivating factor.

Any IoT system should use an ISMS, which is used to identify risks to the IoT system and which identifies and implements sets of security controls that are applied to the IoT system to address those risks. Which controls are applied can be the subject of a maturity model applied to the organization responsible for the IoT system.

Standards that apply to the ISMS are ISO/IEC 27001, which establishes the requirements for the ISMS, and also the closely associated ISO/IEC 27002, which contains sets of security controls to apply as part of the ISMS. The major aim of these standards is to establish and maintain availability, confidentiality and integrity of the IoT system. For IoT systems an additional concern is to prevent abuse of the system, for example through unauthorized and malicious access to interfaces used to control or invoke entities within the system, such as sensors and actuators.

A significant factor for IoT systems is that they often involve operational technology (OT) considerations, which involve a somewhat different set of elements than an ISMS does. In particular, OT places an emphasis on issues of physical security, on safety implications and associated preventive measures and often involves deliberately isolating the entities in the Sensing & Controlling Domain, for example through not using Internet protocols. See the IEC 62443 series, which addresses OT security.

There are other standards that extend the principles of ISO/IEC 27001 to particular contexts and which apply to particular entities within the IoT system and to particular types of IoT system. These include, but are not limited to:

- ISO/IEC 27035 (all parts), *Information technology – Security techniques – Information security incident management*
- ISO/IEC 27034 (all parts), *Information technology – Security techniques – Application security*
- ISO/IEC 27033 (all parts), *Information technology – Security techniques – Network security*
- ISO/IEC 27040, *Information technology – Security techniques – Storage security*
- ISO/IEC 27017, *Information technology – Security techniques – Code of practice for information security controls based on ISO/IEC 27002 for cloud services*
- IEC 62443 (all parts), *Security for industrial automation and control systems*
- NIST SP 800-160, *Systems Security Engineering: Considerations for a Multidisciplinary Approach in the Engineering of Trustworthy Secure Systems*
- NIST SP 1500-201, *Framework for Cyber-Physical Systems*
- NIST SP 800-82, *Guide to Industrial Control Systems (ICS) Security*
- NIST SP 800-53, *Security and Privacy Controls for Federal Information Systems and Organizations*
- NISTIR 7628, *Guidelines for Smart Grid Cybersecurity*
- ISO/IEC 61508 (all parts), *Functional safety of electrical/electronic/programmable electronic safety-related systems*
- ISO/IEC 15045 (all parts), *Home electronic system (HES) gateway*
- ISO/IEC 24767 (all parts), *Information technology – Home network security*

For IoT systems, it is commonly the case that the system is assembled from components that have many different sources. This is particularly the case for the many and diverse devices used both in the Sensing & Controlling Domain and in the User Domain. It is necessary for the organization creating the IoT system to evaluate and understand the information security aspects of each such component and to understand if those aspects meet the requirements of the ISMS of the IoT system as a whole. If such components are deficient in some ways, then appropriate mitigation needs to be applied to the use of such components.

An important aspect of an ISMS is to understand that information security is an ongoing process, not something that is done once and completed. IoT systems are no exception in this respect. Even if there are no changes to the system components, the security of the system can become compromised through newly discovered vulnerabilities and new methods of attack. As a result there can be a need to update components of the system to deal with these vulnerabilities. Careful consideration needs to be given to the processes used to update system components: first, the process to perform updates to components (there have been instances in the past where certain devices had no or very limited update capabilities); second, the security of the update process and its operation only when appropriately authorized and controlled. Impacts on the overall system are also relevant considerations.

It is also wise for an IoT system ISMS to handle the life cycle of the IoT system. A principal concern is to be able to deal with changes in the IoT system itself – the addition of new devices, for example, which might have new characteristics such as the use of different network technologies and communication protocols. Similarly, over time, new applications and services could be introduced which might have novel security implications.

The ISMS needs to be built in to the system development life cycle (SDLC) used by the organization. Testing and validation of the security controls that form part of the ISMS is essential and this needs to be done on an ongoing basis. The ISMS itself is subject to change and update just as much as any other part of the IoT system – where elements of the ISMS are revealed to have vulnerabilities, either found by testing or exposed by security incidents.

11.3.3 IoT system & product Security Life Cycle Reference Model

An organization whose business involves developing, outsourcing or acquiring an IoT system product generally uses a framework of defined processes and activities organized into stages. This framework is commonly named “life cycle model”. Depending on the context, it is referred to as either “IoT system life cycle model”, “IoT product life cycle model” or “software life cycle model”. This is not a new concept – its definition is found in ISO/IEC/IEEE 12207 and ISO/IEC/IEEE 15288.

Such a life cycle model is typically customized for a particular organization.

A specific IoT system or product life cycle, i.e. the evolution from conception through retirement, is an instantiation of the organization’s life cycle model. It is possible for different groups within complex organizations to use different IoT system life cycle models for different projects. Some organizations develop different specialized IoT product life cycle models related to specific IoT product contexts such as web IoT products, real-time IoT products, embedded IoT products, medical IoT products, etc.

Activities performed during the stages of a software or system life cycle are part of organization-wide processes that should be compatible with the normative requirements provided in ISO/IEC/IEEE 12207 and ISO/IEC/IEEE 15288. In addition, the ISO/IEC 24748 and ISO/IEC/IEEE 24748 series provides additional guidance and describes models for system and software development life cycles, life cycle stages and their relationship to life cycle processes.

This document recommends the addition of activities called “IoT system Security Controls” (ISC) to the activities usually performed in the stages defined by the organization’s IoT product life cycle model. Such ISCs are tailored to the life cycle model being employed and also tailored to the nature of the IoT system or product under consideration, which by nature are highly variable.

To address the variability of life cycle models, an IoT system & product Security Life Cycle Reference Model is presented here as a standardized reference for the addition of ISCs to activities performed for IoT product management, IoT product provisioning and operation, infrastructure management and IoT product audit. This Reference Model is a representation of generic stages and activities commonly found in IoT product life cycle models.

The model is not limited to software development. It also makes references to activities from other domains such as governance, software and infrastructure maintenance, project management, audit and control.

The purpose of the IoT system & product Security Life Cycle Reference Model is to:

1) help the organization to validate each of its IoT product life cycles by specifying all processes and actors potentially involved in IoT product security;
2) help the organization to ensure that the security concerns are correctly addressed at all stages of its IoT product life cycles;