ISO/IEC TR 30164

Edition 1.0 2020-04

TECHNICAL REPORT

colour inside

Internet of things (IoT) – Edge computing

ISO/IEC TR 30164:2020-04(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 corrigendum or an amendment might have been published.

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 It also gives information on projects, replaced and withdrawn publications.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 once a month by email.

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.

Electropedia - www.electropedia.org
The world's leading online dictionary on electrotechnology, containing more than 22 000 terminological entries 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.

ISO/IEC TR 30164

Edition 1.0 2020-04

TECHNICAL REPORT

colour inside

Internet of things (IoT) – Edge computing

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

ICS 35.020

ISBN 978-2-8322-8087-4

CONTENTS

FOREWORD ... 4
1 Scope ... 5
2 Normative references ... 5
3 Terms and definitions ... 5
4 Abbreviated terms ... 6
5 Overview ... 8
5.1 General ... 8
5.2 Common concepts ... 8
5.3 General concepts of edge computing ... 9
5.4 Example characteristics of edge computing ... 12
5.5 Stakeholders ... 12
6 Viewpoints ... 14
6.1 Conceptual viewpoint ... 14
6.2 Technology viewpoint ... 15
6.2.1 General ... 15
6.2.2 Cloud computing ... 15
6.2.3 Centralized data centres ... 16
6.2.4 Micro data centre ... 18
6.2.5 Real-time in edge computing ... 18
6.2.6 Heterogeneous computing ... 19
6.2.7 Software defined network (SDN) ... 20
6.2.8 Lightweight operating systems ... 20
6.3 Functional viewpoint ... 20
6.3.1 General ... 20
6.3.2 Data interoperability ... 21
6.3.3 Networking ... 22
6.3.4 Security and privacy ... 22
6.4 Deployment viewpoint ... 26
6.4.1 General ... 26
6.4.2 Edge computing three-tier deployment model ... 26
6.4.3 Edge computing four-tier deployment model ... 27
7 Use cases ... 28
7.1 General ... 28
7.2 Smart elevator ... 29
7.2.1 Description of the use case ... 29
7.2.2 Diagram of the use case ... 29
7.2.3 Technical details ... 30
7.3 Smart video monitoring ... 30
7.3.1 Description of the use case ... 30
7.3.2 Diagram of the use case ... 31
7.3.3 Technical details ... 31
7.4 Intelligent transportation systems ... 32
7.4.1 Description of the use case ... 32
7.4.2 Diagram of the use case ... 33
7.4.3 Technical details ... 34
7.5 Process control in the smart factory ... 34

7.5.1 Description of the use case 34
7.5.2 Diagram of the use case 35
7.5.3 Technical details 36
7.6 Centralized monitoring of power plants (CMPP) 36
7.6.1 Description of the use case 36
7.6.2 Diagram of the use case 37
7.6.3 Technical details 38
7.7 Automated crop monitoring and management system 38
7.7.1 Description of the use case 38
7.7.2 Diagram of the use case 40
7.7.3 Technical details 40
7.8 Smart lighting system 41
7.8.1 Description of the use case 41
7.8.2 Diagram of the use case 42
7.8.3 Technical details 43
Bibliography 45

Figure 1 – IoT edge computing conceptual model 14
Figure 2 – Container virtualization on a host system 17
Figure 3 – Lightweight OS architecture 20
Figure 4 – Software defined network architecture 22
Figure 5 – Edge computing three-tier deployment model 27
Figure 6 – Edge computing four-tier deployment model 28
Figure 7 – Concept of a smart elevator 30
Figure 8 – Concept of video monitoring with edge computing 31
Figure 9 – Concept of intelligent transportation systems with edge computing 34
Figure 10 – Example concept of the smart factory using IIoT 36
Figure 11 – Concept of centralized monitoring of power plants 38
Figure 12 – Concept of automated crop monitoring and management system 40
Figure 13 – Logical model: connectivity between various components 42
Figure 14 – Deployment model: single IoT gateway controlling multiple smart lights 43

Table 1 – Example networking table 10
Table 2 – Capabilities of some IoT entities 11
Table 3 – Technical details of the elements in the smart elevator use case 30
Table 4 – Technical details of the elements in the video monitoring use case 32
Table 5 – Technical details for the intelligent transportation use case 34
Table 6 – Technical details for the smart factory use case 36
Table 7 – Technical details of the CMPP use case 38
Table 8 – Technical details of automated crop monitoring and management system 40
Table 9 – Technical details of the smart lighting use case 44

INTERNET OF THINGS (IoT) – EDGE COMPUTING

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.

The main task of IEC and ISO technical committees is to prepare International Standards. However, a technical committee may propose the publication of a Technical Report when it has collected data of a different kind from that which is normally published as an International Standard, for example "state of the art".

ISO/IEC TR 30164, which is a Technical Report, has been prepared by subcommittee 41: Internet of Things and related technologies, of ISO/IEC joint technical committee 1: Information technology.

The text of this Technical Report is based on the following documents:

Enquiry draft	Report on voting
JTC1-SC41/110/DTR	JTC1-SC41/120/RVDTR

Full information on the voting for the approval of this Technical Report can be found in the report on voting indicated in the above table.

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 publication using a colour printer.

INTERNET OF THINGS (IoT) – EDGE COMPUTING

1 Scope

This document describes the common concepts, terminologies, characteristics, use cases and technologies (including data management, coordination, processing, network functionality, heterogeneous computing, security, hardware/software optimization) of edge computing for IoT systems applications. This document is also meant to assist in the identification of potential areas for standardization in edge computing 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) – Vocabulary*

3 Terms and definitions

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

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

- IEC Electropedia: available at http://www.electropedia.org/
- ISO Online browsing platform: available at http://www.iso.org/obp

3.1
edge
boundary between pertinent digital and physical entities, delineated by networked sensors and actuators

3.2
edge computing
distributed computing that takes place at or near the edge, where the nearness is defined by the system's requirements

3.3
software defined network
SDN
network designed, built and managed with separation of the control plane from the forwarding plane and abstraction of the underlying infrastructure, enabling efficient network management and utilization

3.4
personally identifiable information
PII
information that (a) can be used to establish a link between the information and the natural person to whom such information relates, or (b) is or can be directly or indirectly linked to a natural person

[SOURCE: ISO/IEC 29100:2011 [1], 2.9, modified – In the definition, "to identify the PII principal" has been replaced by "to establish a link between the information and the natural person" and "a PII principal" has been replaced by "a natural person".]

3.5
edge computing entity
ECE
thing (physical or non-physical) having a distinct existence in an edge computing system, with connection, storage and computation capabilities

Note 1 to entry: ISO/IEC TR 23188:2020 [2] uses the term "edge computing node" instead of "edge computing entity".

3.6
distributed computing
model of computing in which processing and storage takes place on a set of entities, with activities coordinated by means of digital messages passed between the entities

3.7
physical edge computing entity
edge computing entity that has material existence in the physical world

EXAMPLES: IoT gateway, sensor, actuator

Note 1 to entry: Refer to ISO/IEC 20924 [3] for the definitions of the terms "sensor", "actuator" and "IoT gateway".

3.8
IoT gateway
edge computing entity that connects one or more proximity networks and the edge devices on those networks to each other and to one or more access networks

3.9
edge computing system
system that uses the structure and capabilities of edge computing

4 Abbreviated terms

4G	the fourth generation of broadband cellular network technology
5G	the fifth generation of broadband cellular network technology
AI	artificial intelligence
AMQP	advanced message queuing protocol
API	application programming interface
APP	applications
CAN	controller area network
CPS	cyber physical system
CPU	central processing unit
CT	communication technology
DDoS	distributed denial-of-service
DDS	data distribution service
DER	distributed energy resource
DetNet	deterministic networking
ECE	edge computing entity
ERP	enterprise resource planning
GEO	geosynchronous orbit

GPS	global positioning system
GPU	graphics processing unit
HSA	heterogeneous system architecture
HTTPS	hypertext transfer protocol secure
I/O	input/output
ICT	information and communication technology
IDS	intrusion detection systems
IEC	International Electrotechnical Commission
IIoT	industrial IoT
IoT	Internet of Things
IP	internet protocol
IPS	intrusion prevention systems
ISO	International Organization for Standardization
IT	information technology
JSON	JavaScript Object Notation
JTC	joint technical committee
LEO	low earth orbit,
LAN	local area network
LiDAR	light detection and ranging
M2M	machine to machine
MEO	medium earth orbit
MES	manufacturing execution system
MPLS	multiprotocol label switching
O&M	operation and management
OPC	open platform communication
OPC-UA	OPC unified architecture
OS	operating system
OT	operational technology
PII	personally identifiable information
PLC	programmable logic controller
PLM	product lifecycle management
PO	purchase order
PV	photovoltaic
QoS	quality of service
REST	representational state transfer
SC	subcommittee
SCADA	supervisory control and data acquisition
SDN	software defined networking
TCP	transmission control protocol
TLS	transport layer security
TR	Technical Report
TSN	time sensitive networking
UDP	user datagram protocol

V2I	vehicle to infrastructure
V2V	vehicle to vehicle
VM	virtual machine
VNF	virtualized network function
VPN	virtual private network
VPP	virtual power plant
WAF	web application firewall
XML	extensible markup language

5 Overview

5.1 General

This document was jointly developed by the teams working on ISO/IEC TR 23188 [2] with cloud computing perspectives and ISO/IEC TR 30164 with IoT computing perspectives. The separate documents exist to expand on these particular perspectives starting from a common base of edge computing concepts, which are stated below. ISO/IEC TR 23188 [2] provides more information on how cloud computing relates to edge computing. ISO/IEC TR 30164 provides more information on how IoT devices and IoT systems relate to edge computing.

5.2 Common concepts

Edge computing is a form of distributed computing in which processing and storage takes place on a set of networked machines which are near the edge, where the nearness is defined by the system's requirements. The edge is marked by the boundary between pertinent digital and physical entities (i.e. between the digital system and the physical world) typically delineated by IoT devices and end-user devices. Nearness is determined by the system requirements, which can include physical distance, but can also include digital factors such as network latency and bandwidth.

Pertinent digital entities here means that the digital entities which need to be considered can vary depending on the system under consideration and the context in which those entities are used.

Digital systems can observe and affect the physical world. Sensors, actuators and human user interface devices are at the boundary between the physical world and digital systems (the edge). Edge computing systems generally combine these devices with distributed computing resources to provide the capabilities of the system. When actions need to occur within specific timeframes and latency considerations affect system design, the edge computing systems help to achieve timing requirements by means of appropriate placement of data processing and data storage. The following are the main motivations for edge computing.

a) Latency: actions often need to occur within specific timeframes and latency considerations affect system design and the choice of the placement of data processing and data storage to achieve timing requirements.

b) Disconnected operations: for example, a car in a canyon. All essential functions need to continue to work.

c) Paucity or high cost of the uplink: for example, an oil rig, a cruise ship or an airliner connected via a satellite link. Need to minimize the volume of data transmitted upstream.

d) Data providence: for example, data represents trade secrets and should not leave a geofence (factory space or corporate network).

Edge computing is characterized by networked systems in which significant data processing and data storage takes place on entities at the edge, rather than in some centralized location. Edge computing can be contrasted with centralized computing where the centralized entities are remote from the edge. However, it is important to note that edge computing is complementary to centralized forms of computing and that in any given system, edge computing is often used in conjunction with centralized computing.

An example of the need to consider the context for the meaning of edge is the servers within a cloud data centre. From the perspective of cloud service customers who build systems using cloud services running on these servers, these entities are anything but at the edge – they are highly centralized. However, from the perspective of the cloud service provider having to manage the cloud data centre, it is highly likely that the servers are instrumented with a variety of IoT sensors capable of reporting various physical properties of the servers, for example, their temperature. In this case, those IoT sensors are at the edge and form part of an edge computing system for managing the data centre.

Edge computing involves entities that are highly heterogeneous and which are commonly arranged in tiers of compute and storage capabilities. The multiple edge computing tiers, each containing varying types of entities, are connected by networks which can also vary in nature depending on the tiers involved. In practice, the number of tiers and the type of entity in each tier is variable, depending on the nature of the system involved.

1) The device tier is at the edge. It typically contains entities which contain sensors or actuators or human user interface devices. Such devices often have limited compute and storage capabilities. The networks used by this tier are often proximity networks, with limited bandwidth and limited range.
2) The edge tier typically sits close to the device tier and its role is to provide direct support to the entities in the device tier. One type of entity in the gateway tier is the gateway (an IoT gateway is an example). The role of the gateway is to connect entities in the device tier to the wider network – it is often the case that proximity networks are local and cannot be used for communication over a wide area. The gateway also typically provides a means for managing the entities in the device tier.
3) Another type of entity in the gateway tier is the control entity. The control entity receives data from entities in the device tier – typically data from sensors or input from user interface devices – and responds by issuing instructions to other entities in the device tier, based on control software running in the control entity. Control entities are usually placed in the gateway tier due to issues of latency and timing. The response of a control entity is often time constrained (sometimes called real-time), such that the response needs to be given before some deadline following the receipt of some data or an event.
4) The central tier represents a tier of entities provided in a centralized location, such as an organizational data centre or as public cloud services. The entities in the central tier offer the ability to provide very substantial compute power and data storage (sometimes termed "unlimited"). The central tier is an excellent place to conduct analytics or other processing that requires both a lot of compute power and also access to a lot of information. The central tier can hold large stores of information which can come from many sources – this may be from across the other tiers of the system or from outside locations, potentially sourced from other organizations.

5.3 General concepts of edge computing

When observing and affecting the physical world, sensors and actuators are at the boundary of the physical world and cyber systems. IoT systems generally use distributed computing resources, combined with sensors and actuators, to enable these interactions (i.e. observing and affecting the physical world). Typical solutions in this area have requirements that actions need to be completed within specific timeframes following some event or observation. Therefore, an awareness of the latency between IoT entities (the computing resources, sensors, and actuators) is needed to achieve those timing requirements. Edge computing helps meet those timing requirements. Edge computing is characterized by networked systems ("connection") in which significant data processing ("compute") and information storage ("storage") take place on devices and entities near the edge, rather than in some centralized location. Edge computing provides the system with reduced latency bounds, beneficial to network and computation, potentially leading to efficiency gains for each. Edge computing can be contrasted with centralized computing (for example, a large cloud computing data centre), where the resources are centralized in large remote data centres. However, it is important to note that edge computing is complementary to centralized forms of computing and that in any given system, edge computing is typically used in conjunction with these centralized computing resources.

A significant driver for the increasing use of edge computing is the continuing increase in the processing power and in the data storage capacities of small and low-power devices and systems that can be placed in locations away from traditional data centres, to address the increasing need to process data quickly in response to input from a sensor. The evolution of mobile phones with their high processing power and large data storage capacity in a small and low-power package has undoubtedly been one of the driving forces in this evolution. However, it is also increasingly the case that innovative IoT systems are driving the requirements for more powerful low-power devices, including the evolution of newer forms of devices such as wearables, robots, and large scale distributed sensor networks.

Edge computing serves a need where actuators are affecting the real world – and there is a need for rapid and close control over those actuators. Edge computing can also serve the needs of human users in remote locations – providing them with a user interface and associated applications that enable them to accomplish tasks and activities. Edge computing can deal with situations where substantial volumes of data are being generated at edge locations and where it is impractical or too costly to transmit all that data to a central location for processing – an example of this is where a set of cameras are providing video feeds. It may be possible to perform a substantial amount of processing at the edge and only transmit a much smaller amount of processed data to a central location (e.g. a count of people in a scene). In other cases, data security can be increased by not transferring data to other locations. By extension, device security, connection security, data security, application security and data privacy could be improved by constraining the data and system to a local region.

An example networking table is shown in Table 1.

Table 1 – Example networking table

Technology	Approximate transmission speed	Approximate latency (single hop)
Wired / Ethernet	100 Mbit/s to 10 Gbit/s	0,3 ms
Cellular/2G/3G/4G/5G	3G – 2 Mbit/s 4G – 20 Mbit/s 5G – 10 Gbit/s	3G – 100 ms 4G – 50 ms 5G – 1 ms
Wi-Fi® [a]	54 Mbit/s theoretical for 802.11n	3 ms
Power line communications	100 Mbit/s	10 ms
Low power, long-range wireless	100 bit/s to 300 kbit/s	1 s to 10 s
Satellite communication: low earth orbit (LEO), medium earth orbit (MEO), geosynchronous orbit (GEO)	LEO: kbit/s to Mbit/s (depending upon system and application) MEO: kbit/s to Mbit/s (depending upon system and application) GEO: kbit/s to Mbit/s (depending upon system and application)	LEO: > 10 ms transmission delay MEO: > 100 ms transmission delay GEO: > 250 ms transmission delay

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

Capabilities of some IoT entities are shown in Table 2.

Table 2 – Capabilities of some IoT entities

Device category	Data consumed	Data generated
Sensing	Application specific – may vary from very small to very large amounts of data.	Application specific – may vary from very small to very large amounts of data.
Actuating	Application specific – may vary from very small to very large amounts of data.	Application specific – may vary from very small to very large amounts of data.
Processing	Application specific – may vary from very small to very large amounts of data.	Application specific – may vary from very small to very large amounts of data.
Data storing	Application specific – may vary from very small to very large amounts of data.	Application specific – may vary from very small to very large amounts of data.

Edge entities vary widely in their compute, storage, networking and data acquisition capabilities. They range from an embedded system, a Raspberry Pi™ [1] grade device to a full PC and micro data centre. A partial example classification is as follows.

- Light compute, light data entities, with very limited compute power and limited data generation.

  Such entities are typically optimized for low cost and low power consumption. Devices commonly have embedded firmware and a very limited operating system (or none at all). They are oriented towards fixed-function capabilities. Communication capabilities may take the form of a low-power, limited reach proximity network of some kind.
- Light compute, heavy data entities, with limited compute power and substantial data generation.

  Such entities generate or deliver substantial data, typically need high bandwidth with other edge devices within the network.
- Heavy compute, light data entities, with substantial compute power, but limited data generation.

  Such entities can have complex processing logics and process data locally. Limited data is delivered to other edge devices through simple and standardized APIs.
- Heavy compute, heavy data entities, with substantial compute power and substantial data generation.

  These are typically full-blown computer systems, although they may be in a physically small package to suit the intended environment. These devices have full operating systems and can support a large stack of software, including the capability to dynamically load software from a remote location. Storage capacity can be substantial and may include replicated and redundant storage to cope with hardware failures. Networking capabilities are likely to be substantial and can include both proximity networks and wireless or wired wide area networks.

The boundaries between these classes of edge entities are not hard and fast and are likely to change over time as technologies evolve. Systems that use edge computing are likely to have to deal with sets of edge entities that span all four classes, with impacts on the architecture and organization of the systems as a whole.

In addition, manageability is an important distinguishing characteristic of edge computing (i.e. servers in the field). While servers in the data centre are in tightly controlled environments with easy manual access, the ones in the field are not in controlled environments and manual access is difficult. So, they need to be hardened, monitored and managed remotely. They are also configured differently; for example, omitting ports (USB), to avoid tampering, etc.

[1] Raspberry Pi is a trademark of Raspberry Pi Foundation. This information is given for the convenience of users of this document and does not constitute an endorsement by IEC or ISO.

Due to the limited resources of edge computing entities (ECE), the coordination among ECEs is indispensable. ECE with different capabilities serve as distinct roles, such as smart devices with lightweight computing capability, smart gateways with capability of data collection, moderate computing, control, etc. and distributed computing system. The term "smart" in this document indicates things with intelligence and automation. For example, smart agriculture uses automated crop monitoring and management.

5.4 Example characteristics of edge computing

Through the use of edge computing, the following value can be added to solutions.

- Edge entities may provide data processing capability (including data analysis, processing, aggregation, privacy, security, etc.) with bounded latency, adaptation and agility.
- Provides support for data buffering for intermittent connections.
- Processing resources that are logically closer to the edge offer lower latency.
- Provides bounded latency that commits to a specific latency requirement.
- Guarantee geofencing of data for security, privacy, regulation or policy enforcement Provides support for connection under various network topologies. Dedicated use cases show different topology for connection of things. The topologies include mesh (in manufacturing, electricity, city, home, etc.), ring (in car, campus, etc.), star (in enterprise, campus, etc.), and others.
- Provides support for multiple network capabilities, including but not limited to network management, control, maintenance, VNF, smart routing, band steering.
- Edge entity manages data, determines lifecycle of data and creates value from data.
- Edge computing provides distributed security (such as authorization, authentication, white list, etc.)
- Edge computing supports coordination between edge and centralized data centres which may be providing cloud services. An ECE may be cooperating with multiple centralized data centres.

5.5 Stakeholders

The following is a list of typical stakeholder groups for edge computing, and some of the questions that those stakeholders might ask in relation to developing an edge system.

a) Developers

Persons who develop the applications and services for the edge computing system.

– How do we develop applications for the edge computing system?

– How do we update the software on the edge devices, IoT gateways and centralized data centres without interrupting the running applications?

b) Architects

Persons who design the architecture of the edge computing system.

– How do we design architectures to meet bounded latency requirements?

– How do we realize a scalable and resilient network that supports a dynamically increasing number of connections?

– How do we distribute workloads among resources to meet the requirements of the system?

– How do we achieve bridging and interoperability between distributed computing resources used for edge computing across different application domains?

– How do we orchestrate resources, including computing, storage, and networking, to satisfy the requirements of stakeholders?

– How do we integrate the cross-boundary technologies onto the edge computing system?

– How do we manage intermittent or unavailable connections between ECEs?

c) Service providers

Persons or organizations that undertake commercial or industrial activities using the edge computing system.

- How do we improve efficiency by using the edge computing services?
- How do we reduce the cost of service deployment?
- How do we increase the profit from running the applications and services in the edge computing system?

d) Equipment manufacturers

Persons or organizations that produce the devices used in the edge devices, IoT gateways, centralized data centres, or other edge computing-related devices.

- How do we design edge computing entities that can integrate easily into an edge computing system?
- How do we ensure that edge computing entities are manageable within an edge computing system?
- How do we ensure that edge computing entities have appropriate security and privacy capabilities?

e) Users

Persons who use the edge computing system.

- How do we interact with edge computing systems?
- How do we interact with physical edge computing entities?
- How do we interact with virtual edge computing entities?

f) Administrators

Persons who manage the edge computing systems.

- How do we manage/use the edge computing systems?
- How do we improve the efficiency to manage the IoT devices?
- How do we monitor the status of IoT devices?
- How do we achieve predictive maintenance?
- How do we process personally identifiable information (PII) in compliance with regulatory requirements or an organization's policies?

g) Security personnel

Persons who manage edge computing related information security threats and risks.

- How do we meet privacy and security requirements in the edge computing system?

h) Consumers

People who purchase edge computing devices and related services for their own personal use.

- How do we protect data that is stored and transmitted through the edge computing devices?
- How do we secure the edge computing devices and receive timely security updates as necessary through the life of the product?
- Where consumer data is processed outside the control of the consumer, for example in a centralized service, is that processing made clear to the consumer and are clear statements made about the limitation of processing and removal of consumer data after a specified period of time?

6 Viewpoints

6.1 Conceptual viewpoint

The IoT edge computing conceptual model is to show IoT entities and relationships at a high system level. Further architecture views have then been created to describe IoT edge computing from different perspectives. Figure 1 shows the entity based IoT edge computing conceptual model.

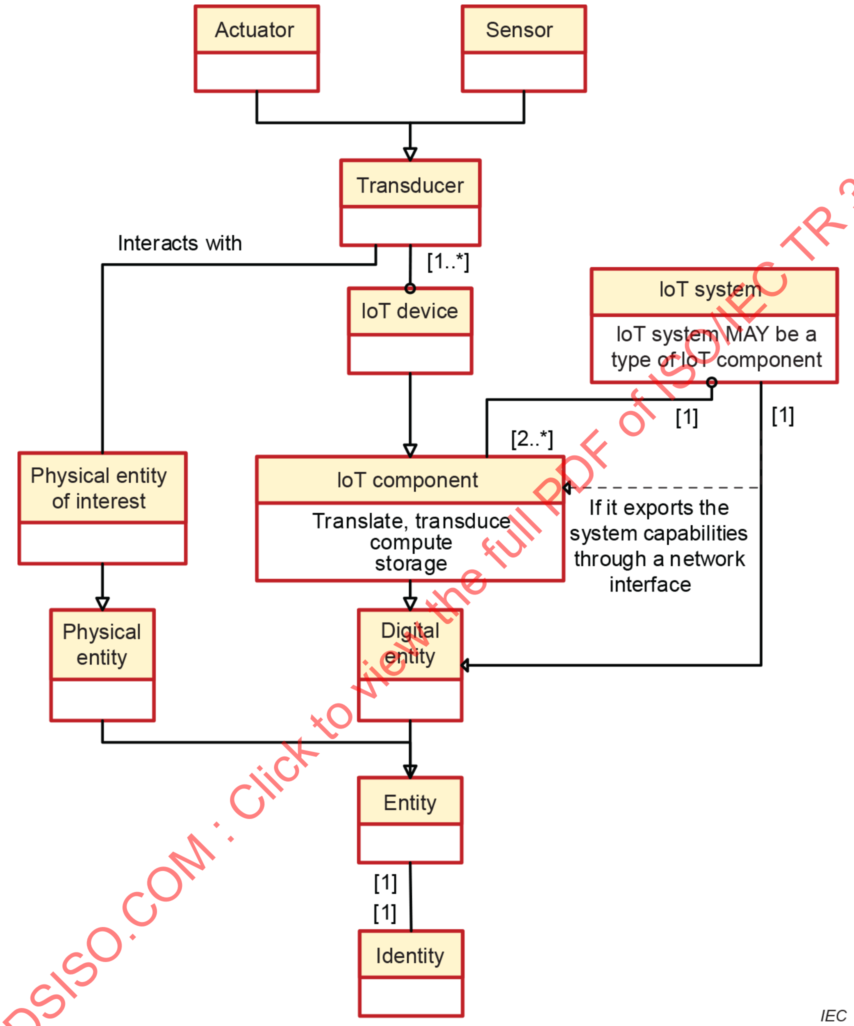

Actuator
Sensor
Transducer
Interacts with
[1..*]
IoT device
IoT system
IoT system MAY be a type of IoT component
[1]
[1]
[2..*]
Physical entity of interest
IoT component
Translate, transduce compute storage
If it exports the system capabilities through a network interface
Physical entity
Digital entity
Entity
[1]
[1]
Identity
IEC

Figure 1 – IoT edge computing conceptual model

Human interface devices start at the top with actuators and sensors (grouped together, sensors and actuators can be referred to as "transducers"). Electronic signal[s] (representing an observation of a physical property of an entity of interest) are converted into information with a transducer. This transducer capability is a defining characteristic of an IoT device. IoT components may sense or act as well as translate, compute or store information for an IoT system. An IoT system may also be an IoT component (if its capabilities are available to be used within another IoT system through a network interface). An entity can be physical or digital (virtual) in nature. Physical entities of interest can be observed and acted on by sensors or actuators. Digital entities consist of IoT components or IoT systems which can then form an identity.

6.2 Technology viewpoint

6.2.1 General

Subclause 6.2 describes some important technologies used in the edge computing, but it does not include all the technologies related to edge computing.

6.2.2 Cloud computing

As defined in ISO/IEC 17788 [4], cloud computing is a paradigm for enabling network access to a scalable and elastic pool of shareable physical or virtual resources with self-service provisioning and administration on-demand. It is common for cloud computing resources to be held in centralized data centres, although cloud computing can be implemented in other locations which could be nearer to the edge. Of more importance to edge computing and IoT is that cloud computing resources are virtualized – and the technologies and practices relating to virtualization are of direct relevance to edge computing.

Cloud computing can be implemented across a spectrum of locations, of different sizes and different locations. While it is common to provide public cloud services in one, or more commonly multiple distributed, large and centralized data centres, cloud computing is also provided in non-centralized more widely distributed locations with smaller sets of resources. Such smaller cloud computing implementations can be provided using the private cloud deployment model, where the customer might own all the resources involved and where the resources are placed on customer premises. However, there are also distributed public cloud computing implementations, such as in mobile phone base stations, located to be close to devices at the edge.

It is also important to recognize that some of the technologies commonly used in cloud computing, such as virtual machines (VMs), containers and their associated management software, are of importance to the implementation of edge computing, even for entities and locations where cloud computing itself is not implemented. For example, containers can be used on small, low power entities used in the edge tier such as the Raspberry Pi.

It is also commonly the case that edge computing systems and IoT systems combine the use of IoT devices and edge tier entities with the use of cloud services, both in centralized locations and in distributed locations. Cloud computing can provide a convenient location to gather data, in potentially high volumes, from across a large number of widely distributed IoT devices and also to perform complex processing on that data that requires the use of a large quantity of compute power.

Cloud computing utilizes virtualization technologies to run software that delivers various types of cloud services. Software that runs in cloud computing is often packaged, delivered and deployed using container technologies. It is typical of virtualized software to run in multiple instances in parallel. The multiple instances may all run in a single data centre, or they may run across two or more physically separated data centres. There are many reasons for running multiple instances in parallel, but principally it is done:

a) to provide scalability – i.e. to enable greater throughput of work for the particular application or service provided by the software;

b) to provide resilience – multiple running instances avoid a single point of failure: should one instance fail for some reason, incoming requests can be distributed to the other working instances. One reason for running instances in different data centres is to cope with a failure that might affect one data centre as a whole, such as a natural disaster or a resource failure such as a power outage or communications failure affecting a particular location.

Similarly, it is typically the case that data storage is virtualized (as a storage service), and data is often replicated to multiple locations, both for resilience purposes and also to reduce latency for the software interacting with the data.

It is the case that the capabilities of devices at or near the edge are continually increasing – in terms of processing power, in terms of storage capacity and in terms of network bandwidth. The increased capabilities change the possibilities for the locations of compute and data storage. However, making effective and efficient use of these increased capabilities often requires the use of new technologies. An example concerns video surveillance. Originally, only the camera itself could be placed at the edge – the video feed would be transmitted to a central location for processing.

Tasks such as vehicle number plate recognition initially demanded more processing capacity than was available at the edge. With increasing capability, the processing for number plate recognition could move first to an edge tier entity and then right into the camera device itself. In more recent times, complex video processing such as the recognition of human beings in a scene, tracking human movements or identifying individuals has been possible in edge entities or in camera devices.

Complex software commonly needs continuous maintenance, either to fix problems or to enhance its capabilities – this places demands on the edge tier entities and the devices for the secure update of the software. It is also the case that running multiple pieces of software on a particular entity might be necessary. These considerations can favour the use of appropriate technologies such as containers, virtual networks and storage services.

6.2.3 Centralized data centres

An aspect of centralized data centres of relevance to edge computing is the fact that centralized data centres often manage large sets of physical and virtual resources. These resources that may be used within an IoT system are often logically distant from the sensors and actuators on the edge.

It is often the case that edge computing is used in conjunction with centralized data centres – edge computing entities are often connected to these centralized data centres. When more compute power is required than is available at the edge, centralized data centres process computation-intensive tasks. Centralized data centres may have access to datasets that are not available to edge computing entities, such as big data datasets, or where the data spans across the whole system, or data that derives from outside the system entirely.

Many IoT systems implement some form of edge computing. It is common for IoT devices to have some form of processing and data storage. It is also common for IoT systems to have gateways and control entities in an edge tier, while at the same time using entities in the central tier to support the IoT devices and the edge tier and enable the realization of all the capabilities required of the IoT system. Containers are a valuable virtualization technology for software which supports the heterogeneous environment represented by edge computing systems. They enable multiple software components to coexist on a single entity with isolation. They enable the deployment of the same software onto a range of different types of entities without the need to change the software. They support the virtualization of both networking and storage services, enabling the software to be deployed on entities with different networking and storage capabilities.

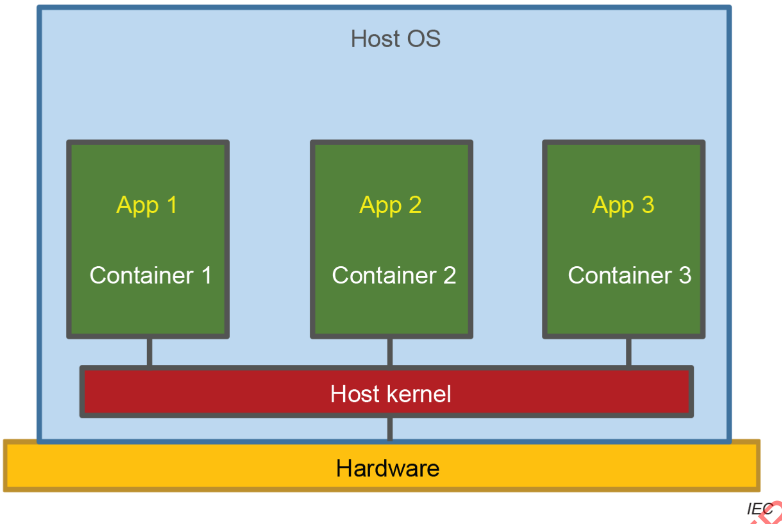


Figure 2 – Container virtualization on a host system

Figure 2 shows three containers running on a physical system. The physical system has its own host operating system (Host OS). Each container contains its own application software and runs that software in a set of processes using resources such as memory, CPU, storage and networking, isolated from the other containers running on the same system, but all sharing the kernel of the host operating system.

The kernel of the host operating system is being shared by all the containers, which essentially means that the operating system used by the software in the containers needs to be compatible with the Host OS kernel. This can allow for the different containers to potentially use different variants of the Linux®[2] operating system where the Host OS is a Linux variant, for example, but it does not allow for the Windows®[3] operating system to be used within a container if the Host OS is a Linux variant (and vice-versa).

Containers are commonly used in association with cloud services, providing horizontal scalability, isolation and resource and lifecycle management. However, it is increasingly the case that containers are being used outside the context of cloud computing – and this is especially relevant to edge computing, with the use of containers on edge tier entities.

There are many implementations of container technology, of which DOCKER™ is probably the most common. However, container images and container runtimes are being standardized by the Open Containers Initiative (OCI) [5] and there is an open source implementation of the standardized version available in the runC project. Containerization helps achieve functionality isolation in the form of services in edge computing that can be deployed anywhere, for example on a smart device, an IoT gateway, in a micro data centre (as explained in 6.2.4), in public or private cloud. Containers permit multiple IoT applications from different vendors to coexist on processing hardware and be maintained and upgraded separately with no reference to other vendor applications, or effect on their operation, while allowing information sharing through defined interfaces. If an application crashes, the applications in other containers won't be affected. Containerization can facilitate the lifecycle management of applications and related resources can be initiated and terminated on demand.

The material in 6.2.3 is derived from ISO/IEC TS 23167:2020 [6].

[2] Linux® is the registered trademark of Linus Torvalds. This information is given for the convenience of users of this document and does not constitute an endorsement by ISO or IEC.

[3] Windows® is the registered trademark of Microsoft Corporation. This information is given for the convenience of users of this document and does not constitute an endorsement by ISO or IEC.

6.2.4 Micro data centre

A micro data centre (MDC) is a smaller or modular data centre architecture that is designed to solve different sets of problems that take types of compute workload that do not require the facilities of a typical large-scale data centre and which can be deployed in a wide variety of locations and environments.

The size may vary – a micro data centre might include fewer than four servers in a single 19-inch rack. An MDC can be supplied as a standalone rack containing all the components of a traditional data centre, including cooling, power supply, power backup, security, and fire suppression.

An MDC can be used as the implementation of a set of entities in the edge tier, in a location that is suitably close to the edge devices, without the overhead and complexity of a full-blown data centre.

6.2.5 Real-time in edge computing

In an edge computing system, computing and storage are located near the system edge to achieve real-time performance. In a real-time system, subsystems have a sense of time and the transmission of the data is scheduled in time to avoid the traffic congestion and provide the system with end-to-end deterministic latency. Various real-time technologies can be used in edge computing and details of the real-time IoT framework will be found in a future International Standard (ISO/IEC 30165). In this document, time-sensitive networking is shown as an example of real-time network technology.

Time-sensitive networking (TSN) is a set of standards in IEEE TSN taskgroup [7] that define time-sensitive data transfer mechanisms, which is an extension for Ethernet technology. TSN is the same as normal networking, but with different TSN standards documents for real-time communication solutions. TSN has the following features for critical data streams.

a) Precision time synchronization: Entities of network and hosts need to have a common understanding of time. Devices are required to have the ability to synchronize their time. End-to-end latency should be less than 1 µs and jitter should be less than 500 ns.

b) Scheduling and traffic shaping: Different traffic classes with different QoS can be integrated into communication packets that transport over network. TSN defines the rules to handle and transmit communication packets, including filtering, policing and pre-emption of the network traffic. As a converged network infrastructure standard, TSN has the ability to guarantee the transmission of the high priority traffic within the network.

c) Resource reservations and fault-tolerance: Resource reservation for critical data streams is managed via configuration or protocol actions. Low packet loss ratios which start at 10^{-6} and extend to 10^{-10} or better can guarantee end-to-end latency for a reserved flow. And redundant path should be applied to achieve fault-tolerance.

d) Centralized user configuration (CUC) and centralized network configuration (CNC): The CUC is intended to provide a user interface for the CNC server in the TSN management system. The CUC communicates with CNC via RESTCONF protocol and Yet Another Next Generation (YANG) model. The CUC/CNC can accelerate user's configurations and provide benefit for the centralized management of network entities.

The TSN technology can equip the edge computing infrastructure with low latency, flexible, reliable and easy-to-maintain network connections. It can help the convergence of the IT and OT networks, changing the physically isolated networks into logically isolated networks.

Besides TSN, deterministic networking (DetNet) is a set of standards under development in the IETF DetNet working group that define time-sensitive data transfer for IP and MPLS network technology. DetNet extends and applies TSN concepts and functionality to IP and MPLS technologies resulting in benefits analogous to TSN benefits for Ethernet.

6.2.6 Heterogeneous computing

6.2.6.1 General concept

Heterogeneous computing is the strategy of deploying multiple types of processing elements within a single system, and allowing each to perform the tasks to which it is best suited. Heterogeneous entities mean variations in the hardware of physical entities within an edge computing system, for example:

a) variations in the hardware of entities within an edge computing system;

b) the amount of RAM available;

c) the amount and type of data storage present;

d) the type and number of network connections present.

A typical edge computing system has many different types of entities, reflecting the nature of the requirements, for example:

1) sensors and actuators can often exist on physical entities with very limited capabilities, designed to deal with relevant environmental factors – location, power usage, network type, cost and so on;

2) edge tier entities can also involve limited capabilities, although usually more capable than device tier entities – limited number and power of processors, limited RAM, limited data storage capacity (devices similar to the Raspberry Pi are a classic example of cheap, flexible and rapidly deployable node hardware).

6.2.6.2 Examples of applications of heterogeneous computing

The popular application of heterogeneous computing is a computer system in which various processing units are coordinated together to improve the computing efficiency.

Since their earliest days, computers have contained central processing units (CPUs) designed to run general programming tasks very well. But in the last couple of decades, mainstream computer systems typically include other processing elements as well. The most prevalent is the graphics processing unit (GPU), originally designed to perform specialized graphics computations in parallel. Over time, GPUs have become more powerful and more generalized, allowing them to be applied to general purpose parallel computing tasks with excellent power efficiency.

With the evolution of cloud technologies and network technologies, another typical application of heterogeneous computing is a disaggregated hardware system.

In a disaggregated hardware system, the computing resources are distributed in multiple physical locations, called computing entities. Each computing entity carries multiple processors (Intel® x86, Atom® and Arm®, etc.[4]) for application-diversified data processing. With resource disaggregation and unified interconnects, on-demand resource allocation can be supported by hardware with fine-granularity, and intelligent management can be conducted to achieve high resource utilization. Required resources from the pools can be appropriately allocated according to the characteristics of applications. Optimized algorithm assigns and schedules tasks on specific resource partitions where customized OSs are hosted. Thus, accessibility and bandwidth of remote memory and peripherals can be ensured within the partition, and hence end-to-end service level agreements (SLA) [8] can be guaranteed.

[4] Intel® and Atom® and are the registered trademarks of Intel Corporation or its subsidiaries; Arm® is the registered trademark of Arm Limited or its subsidiaries. This information is given for the convenience of users of this document and does not constitute an endorsement by ISO or IEC.

6.2.7 Software defined network (SDN)

A software defined network [9] is designed, built and managed with separation of the control plane from the forwarding plane and abstraction of the underlying infrastructure, enabling efficient network management and utilization. SDNs implement network virtualization – enabling the provision of networking capabilities at the application level, isolated from the networking used by other applications. SDN defines a new method of networking and better supports new network architectures and new service innovations.

When SDN is applied to edge computing, the edge computing network can support flexible expansion, enabling efficient and low-cost automatic operation and maintenance. SDN opens centralized network control and network status information to applications so that they can flexibly and quickly drive network resource scheduling.

Other virtualization technologies can play similar roles to SDN, but are not listed in this document, for example network functions virtualization [10].

6.2.8 Lightweight operating systems

Lightweight operating systems is an effective way to provide an open and customizable platform to handle the applications and facilitate the development, as shown in Figure 3. Compared to the cloud, the edge computing entity (ECE) has constrained hardware resources. Therefore, the operating system running on the ECEs should be lightweight, with high start-up speed, low power consumption and fast response abilities.

It has been impossible to realize a truly interconnected world where devices are not interoperable. The import of the lightweight operating systems to the edge intelligence will change this situation. This is because the platform provides the consistent APIs over the connectivity, security, application domain and so on, which hides the vendors' implementation differences and provides only open, clean, and common APIs to work with.

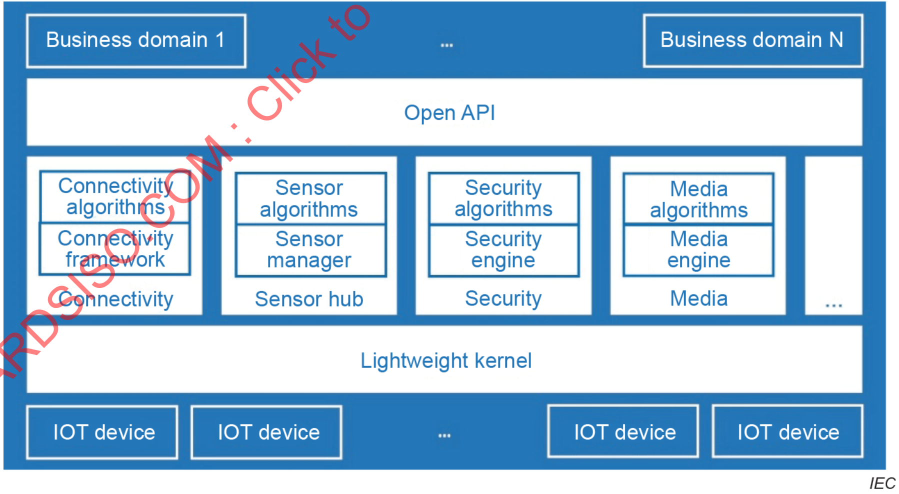


Figure 3 – Lightweight OS architecture

6.3 Functional viewpoint

6.3.1 General

An edge computing entity can have but is not limited to the functions mentioned in 6.3.

6.3.2 Data interoperability

To realize cross-vendor data aggregation, interoperation and analysis, unified semantic meanings are required. Flexible and unified data presentation should also be supported. The technologies enabling the interoperability between machines can refer to DDS and OPC UA. [5]

DDS (Data Distribution Service) is a machine-to-machine (M2M) standard developed by Object Management Group It is a data-centric technology. DDS is applied to enable scalable and interoperable data exchange. DDS is based on UDP/IP (not TCP/IP), and therefore supports real-time publication and subscription of messages for sending and receiving data. It can also serve and receive data from the cloud service. Publishers create topics and publish samples. DDS is a middleware that sends samples to subscribers that subscribe to the topics. DDS provides a three-layer data domain architecture for edge computing. DDS exchanges data between edge computing entities or between edge computing entities and things. For distributed applications, DDS simplifies complex network programming. DDS can be used as a means of adapting interfaces to different machine languages.

[SOURCE: OMG DDS] [11]

OPC (Open Platform Communications) is the interoperability standard developed by the OPC Foundation. It is platform independent and can ensure safe and reliable data exchange from different industrial equipment. The OPC standard is a series of specifications that define the interface between clients and servers, including access to real-time and historical data, monitoring of alarms and events. OPC proposed an information model that models complex data from different devices into information. OPC allows servers to provide an integrated Address Space and service model. OPC Classic provides communication for Windows-based systems. OPC UA is service-oriented architecture that integrates OPC Classic specifications into one extensible framework. OPC UA is not tied to one operating system or programming language. In the Client Server model, OPC UA sends requests and responses between them. In the Pub/Sub model, OPC UA transfers Network Messages between Publishers and Subscribers over different kinds of network. OPC UA supports TCP, HTTPS, AMQP and WebSocket transport protocols. It defines XML/text, binary and JSON data encodings. Through the construction of a unified information model architecture, OPC UA provides the data domain architecture for edge computing. OPC UA achieves cross-vendor data interoperability and the analysis of data between edge computing entities or between edge computing entities and things.

[SOURCE: OPC UA [12][13]]

Data management and coordination

Local data processing capabilities are critical for edge computing entities (ECEs).The ECEs include the IoT devices in the upper layer and the edge entities with data management abilities (e.g. a surveillance camera that could distinguish video clips with crucial messages). In the ECEs, data analysis models need to be adapted to perform real-time data cleansing and analysis, and pre-defined service response policies are triggered based on data analysis results. The computing entities should also be able to distribute more intensive computing power throughout multiple entities. This illustrates the need for modern distributed computing standards to orchestrate the compute tasks either among child entities, at a peer-to-peer level, throughout the cloud, and using virtualized resources or via a local GPU-based compute accelerator.

To summarize, data management, choreography and processing technologies support edge computing and enable local processing of data so decisions are made more quickly and communication costs with centralized servers are reduced.

[5] DDS and OPC-UA are two example methods for data interoperability. Alternative methods can be used according to specific application scenarios.

6.3.3 Networking

The explosion of IoT applications/services requires diversified networking functions. However, traditional networks have weaknesses, including high network complexity, inconsistent policies, difficulties to scale, and vendor dependence.

How to make the network capability adapt to the requirements for diversified service is the key question that SDN answers. Figure 4 shows the general architecture of SDN. A centralized controller has the global information of the network capabilities and achieves the global optimization regarding the requirements of the actual service.

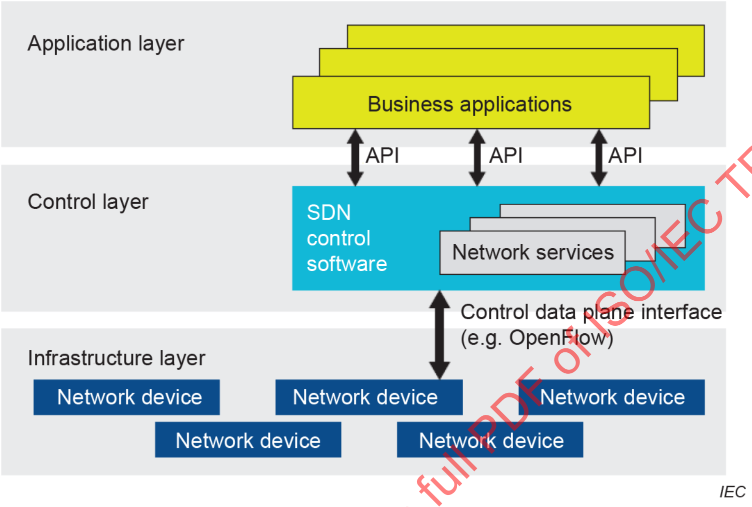


Figure 4 – Software defined network architecture

The network is programmable by operators/enterprise customers so that software can define the network. Moreover, the procedure of network definition does not require an understanding of thousands of communication protocols. Open APIs are defined to provide common network functions, such as routing, multicast, security, access control, bandwidth management, traffic engineering, quality of service, processor and storage optimization, energy usage, and all forms of policy management, custom tailored to meet business objectives.

6.3.4 Security and privacy

6.3.4.1 General

Edge computing has associated risks and as a result needs to incorporate appropriate controls and capabilities that satisfy business expectations and applicable regulations, to secure and protect information, systems and entities. Information security is one key element that applies across all the elements of an edge computing system. Commonly, edge computing systems also process personally identifiable information (PII) and as a result, require taking appropriate privacy protection activities to protect PII data being processed. In some circumstances, some threats under consideration for legacy devices, which do not support hardware security, may be mitigated by physical security measures as an alternative.

As indicated by ISO/IEC 30141 [14], edge computing systems also usually involve more than the aspects of security and privacy. The term "trustworthiness" is applied to these systems, encompassing the aspects of safety, reliability and resilience in addition to security and privacy. These five aspects of trustworthiness interact with one another in ways that can create challenges for edge computing systems that go beyond those that apply to other types of system.

It is also necessary to recognize that the term "cybersecurity" is often applied to edge computing and to IoT systems. Cybersecurity typically involves a different viewpoint than information security, concentrating on the risks that arise from constructing systems of systems, with entities in multiple interacting tiers using networks that can be used to mount attacks on those systems.

Some edge computing systems, particularly those involved in manufacturing and control, also involve combining traditional operational technology systems with information technology systems. This can cause difficulties for security because operational technology systems have a different approach to security than information technology systems.

6.3.4.2 Applying foundational security principles

Edge computing systems apply foundational security principles to:

- secure information to ensure its availability, its integrity and its confidentiality;
- secure systems to ensure that they operate to design, that they cannot be hijacked, that they have no vulnerabilities, that they are available (e.g. address DDoS attacks);
- detect attacks and incidents, record and report attacks and incidents;
- ensure that entities only communicate with other authorized entities and that networks are appropriately protected;
- apply all appropriate data protection principles where personal data is involved, when stored or processed on an entity, or when transmitted on networks between entities;
- ensure that access to or management of entities in the system is subject to authentication and authorization;
- adapt the security functions to the specific architecture of edge computing;
- design security functions that can be flexibly deployed and expanded;
- provision the system to continuously mitigate attacks within a certain period of time;
- provision the system to tolerate function failures within a specified range and limit while basic functions run properly;
- ensure that the entire system can quickly recover from failure.

Security requires a comprehensive approach to all entities in the edge computing system, applying the foundational security principles during the whole lifecycle of the system, from design through implementation, allowing for maintenance and update and finally termination or retirement.

An edge system may involve trust boundaries – edge systems are often built as "systems of systems" and the ownership and control of the component systems could belong to different organizations and/or individuals. Such trust boundaries need to be addressed as part of the overall system design.

6.3.4.3 Secure entities and devices

Given that edge computing systems are built from a networked mesh of entities, it is important to consider the security of the entities – and especially of those entities which are IoT devices, often having fewer capabilities than other entities. It is important that the entities and devices provide mechanisms to ensure that they operate in a trusted manner at all times. This trust should be consistent across both the hardware and software elements of the entity or device. In some circumstances, legacy devices which do not support hardware security may be secured by physical security measures as an alternative.

Items to consider in providing secure entities and devices include:

- hardware root of trust;
- secure cryptographic keys;
- certificate based authentication;
- software compartmentalization;
- protection for all potential attack surfaces, with multiple mitigations for threats;
- software integrity and updateability;

- data integrity and timeliness;
- detection and reporting of incidents.

Hardware-based root of trust is both an approach to defend against low-level software attacks and the basis for ensuring that only authorized software can run on the entity or device.

Cryptographic keys are essential to many of the security capabilities of any entity. Having those keys stored in a hardware-protected vault is important to avoid the compromise of security capabilities. Assigning the keys to the device at creation is one strategy to employ.

Authentication is necessary for the entity or device itself (e.g. in proving its identity) and for capabilities running on the entity – it is typical to achieve this using signed certificates, using protected cryptographic keys.

Software compartmentalization, providing barriers between different software components running on an entity, prevents a problem (including a breach) concerning one component from spreading to other components. The barriers may be hardware imposed or hardware assisted – and special provision is often applied to sensitive areas of memory, such as the storage used for cryptographic keys. Compartmentalization can also apply to data storage, so that only specific software can access and use specific areas of data storage.

It is typical for any particular entity or device to have multiple different attack surfaces – protection needs to be provided for each potential threat and the countermeasures should be provided in depth to mitigate against the effect of a breach.

Only authorized software should run on the entity or device – integrity checking should be applied to the software to ensure that it is both authorized and that it has not been subject to tampering. A desirable capability is the ability to check the validity of software remotely. At the same time, all software should be updateable so that any new vulnerabilities or threats can be addressed.

The integrity of data within the entity should be protected – and the timely availability of the data should be addressed, which can form another type of integrity (i.e. out-of-date data is not used for decision making).

Detection and reporting of incidents is a key requirement – when an attack occurs, it needs to be detected and reported so that appropriate management action can be taken. Reporting of routine activities is also desirable.

6.3.4.4 Connectivity and network security

For edge computing systems, networks connect entities within a tier and also connect entities in different tiers. The networks and the connectivity that takes place over them are a fundamental aspect of edge computing. Network security is an important aspect of the edge computing systems.

Networks need security controls which ensure that only authorized components communicate with each other – this is commonly done at the level of entities. Increasingly, particularly through the use of software defined networking, controls are applied at the level of components.

Data flowing on the networks need to be confidential and potentially private. It needs to not be subject to eavesdropping or to alteration. Equally, the availability of the network to transmit the data is key, especially where the timely arrival of the data is necessary for the correct operation of the edge system. Controls to prevent interference between different data flows taking place over the same physical network are another consideration.

In addition, the security design and implementation need to take the following unique features of edge computing scenarios into consideration.

- Lightweight security functions can be deployed on IoT devices with limited hardware resources.
- The traditional trust-based security model is no longer applicable to the access of a large number of heterogeneous devices. Therefore, the security model (such as the whitelist function) needs to be re-designed based on the minimum authorization principle.
- Isolation between networks and domains is implemented on key entities (such as smart gateways) to control the scope of security attacks and risks, preventing attacks on an entity from spreading to the entire network.
- The security and real-time situation awareness functions are seamlessly embedded into the entire edge computing architecture to achieve continuous detection and response. Automation should be implemented as much as possible, but manual intervention is also required at times.
- Security design needs to cover each tier of the edge computing architecture, and different layers require different security features. In addition, unified situation awareness, security management and orchestration, identity authentication and management, and security operation and maintenance are required to ensure maximum security and reliability of the entire architecture.

Edge computing entities (ECE) security includes basic ECE security, entity security, software hardening and security configuration, secure and reliable remote upgrade, lightweight trusted computing and hardware safety switch. The secure and reliable remote upgrade can fix vulnerabilities and install patches in time, and prevent system failures after the upgrade. Lightweight trusted computing is applicable to simple IoT devices with limited computing (CPU) and storage resources to solve basic trust problems.

Network security includes firewalls, IPS/IDS, anti-DDoS, VPN/TLS, and the reuse of security functions of some transport protocols, such as the REST protocol. Anti-DDoS is particularly important in IoT and edge computing. In recent years, a growing number of attacks on IoT devices are DDoS attacks. Attackers control IoT devices with poor security (such as cameras with fixed passwords) to attack specific targets.

Data security includes data encryption, data isolation and destruction, data anti-tampering, privacy protection (data anonymization), data access control, and data leakage prevention. Data encryption includes encryption during data transmission and storage. Data leakage prevention for edge computing is different from that of traditional systems because edge computing devices are usually deployed in distributed mode. Special considerations are required to ensure that data will not be leaked even if these devices are stolen.

Application security includes security functions such as whitelist, application security audit, malicious code prevention, web application firewall (WAF), and sandbox. The whitelist function is important in the edge computing architecture. The traditional trust-based security model is no longer applicable to the access of a large number of heterogeneous terminals and various services. Therefore, security models (such as the whitelist function) with minimal authorization are used to manage applications and access rights.

Security situation awareness and security management and orchestration: Since a large number of diversified terminals are connected at the network edge and services carried on the network are complex, passive security defence is ineffective. Therefore, more proactive security defence methods are required, including Big Data-based situation awareness, advanced threat detection, unified network-wide security policy execution, and proactive protection. These can facilitate quick responses. In combination with comprehensive operation and maintenance monitoring and emergency response mechanisms, maximum security, availability, and reliability of the edge computing architecture can be ensured.

Identity and authentication management: Covers all function layers. However, accessing a large number of devices at the network edge places much pressure on the performance of the traditional centralized security authentication system, especially when many devices go online within a short period of time. Therefore, the decentralized and distributed authentication and certificate management can be used if needed.

6.3.4.5 Privacy and personally identifiable information in edge computing

Personal data, otherwise known as personally identifiable information (PII), can commonly be present in edge computing systems.

However the PII enters the system, there is a need to provide appropriate privacy protection, which broadly involves the implementation of the privacy principles described in ISO/IEC 29100 [1]. Ideally, this is done right from the inception of the system, using a privacy-by-design approach.

It is necessary to identify and characterize the PII that is present in the system. In edge computing systems, separate data flows may bring disparate data together, which increases the amount of data that is classified as PII. A privacy impact assessment can help identify both the PII that is present, where it is processed in the system and the risks associated with that PII and the processing it is subject to.

There is a general expectation for an individual to engage with the device. At the same time, there is a need to give visibility and control over the PII to the person it relates to (the PII principal) – for example, allowing inspection, correction and potentially deletion of the PII. It is difficult to provide traditional accountability processes for any decentralized data processing functions regardless of whether they are in edge computing systems or not. Notwithstanding, protection of PII is essential – from unauthorized access, from tampering, but also from processing that is not agreed by the PII principal. Devices attached to the edge computing systems are accessed remotely, data including PII can be shared outside beyond the control of the PII principal.

Aims of the edge system include limiting the length of time PII is kept, and anonymizing or minimizing the data wherever possible. For example, in edge computing, consumers' private data can be kept in edge computing entities and only non-sensitive data is stored at the public cloud. There are trade-offs to managing security and privacy risks of the devices attached to the system as managing some risks may affect other types of risks and introduce new ones.

6.4 Deployment viewpoint

6.4.1 General

Two example edge computing deployment models – the three-tier model and the four-layer model – are discussed in 6.4.2 and 6.4.3, respectively.

6.4.2 Edge computing three-tier deployment model

The three-tier deployment model is shown in Figure 5.

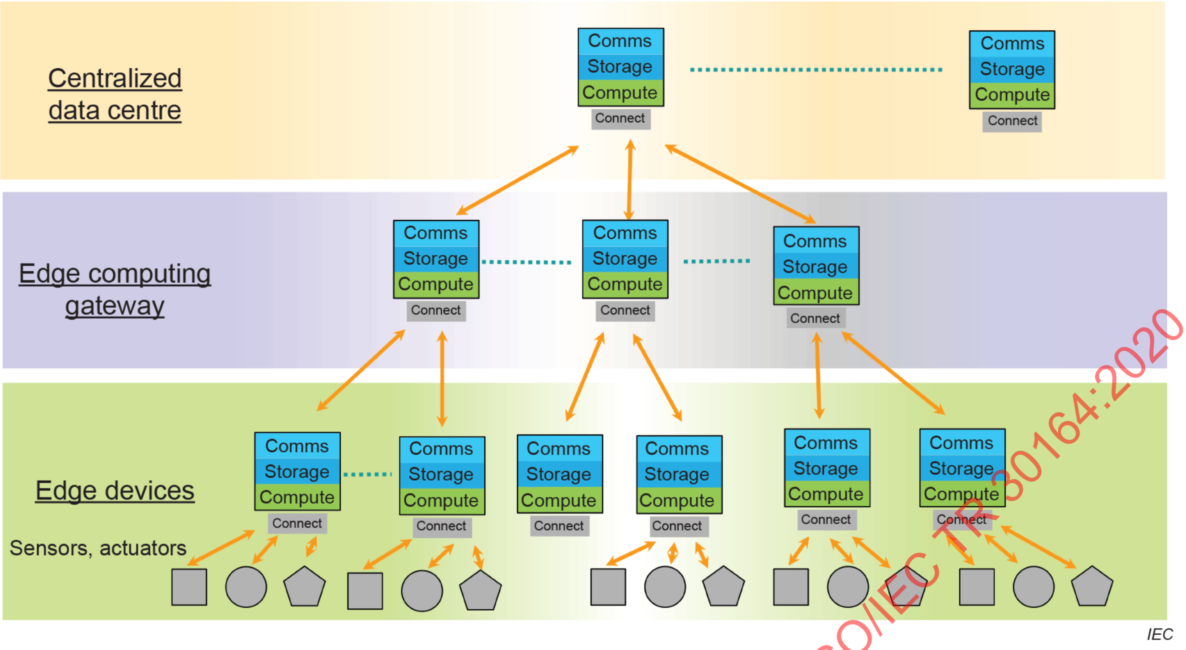


Figure 5 – Edge computing three-tier deployment model

This model is applicable to scenarios where services are deployed in one or more scattered areas, each with a low traffic volume.

Typical scenarios include smart street lamps, smart elevators, and smart environmental protection.

After local processing of IoT devices, multiple types of or multiple service data flows are aggregated on the IoT gateways along the north–south direction. In addition to network functions such as supporting the access and local management of IoT devices and bus protocol conversion, the IoT gateways provide real-time streaming data analysis, security protection, and small-scale data storage. The gateways process real-time service requirements locally, and aggregate and send non-real-time data to the cloud for processing.

6.4.3 Edge computing four-tier deployment model

The four-tier deployment model is shown in Figure 6.

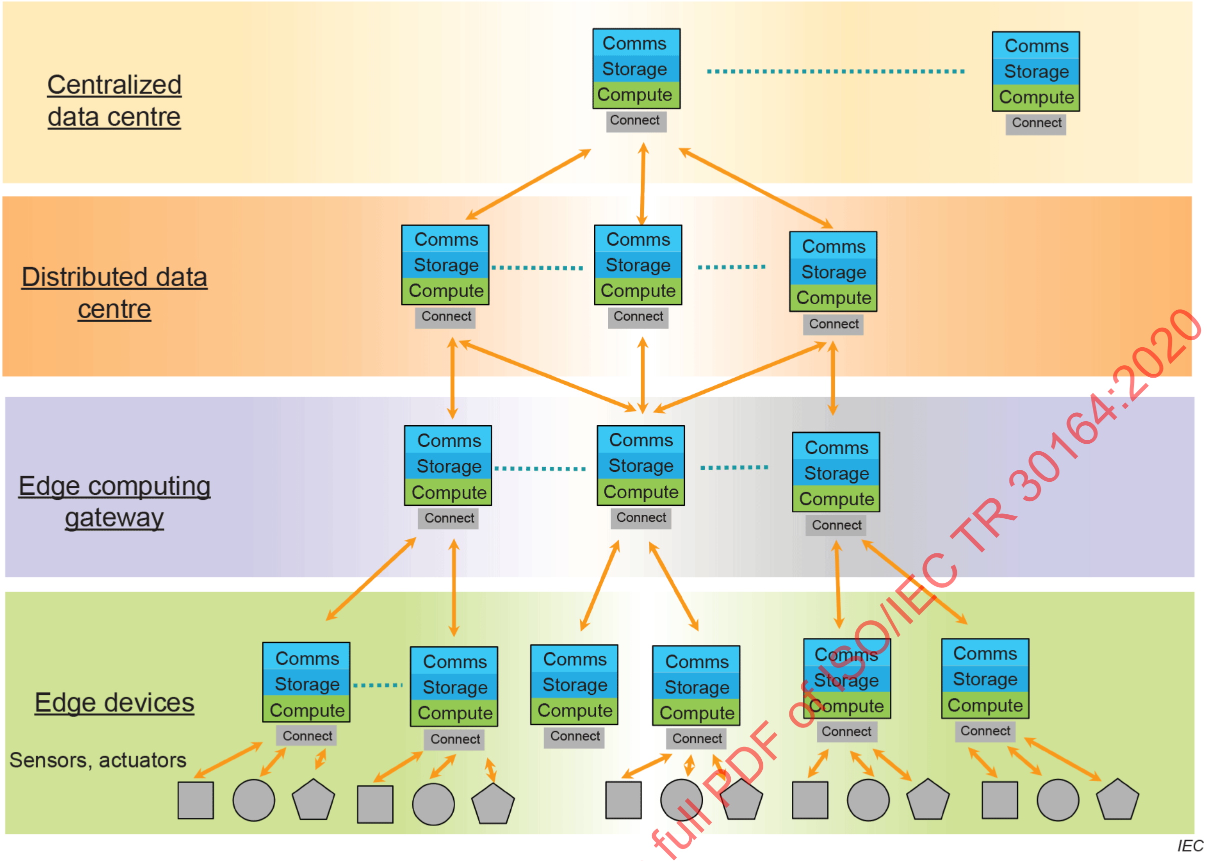


Figure 6 – Edge computing four-tier deployment model

This model is applicable to scenarios where services are deployed centrally and the traffic volume is high.

Typical scenarios include smart video, smart grid, and smart manufacturing.

The typical differences between four-tier deployment and three-tier deployment are as follows: At the edge, there is a large amount of data and many local application systems are deployed. Therefore, a large amount of computing and storage resources is required. After most real-time data processing has been completed on IoT devices and edge gateways, data is aggregated on local distributed data centres for secondary processing. The distributed ECEs exchange data and knowledge through east–west connections, support the horizontal elastic expansion of computing and storage resources and implement real-time decision-making and optimization operations locally.

7 Use cases

7.1 General

Edge computing can be widely applied in industry domains such as manufacturing, energy, utility, transportation, health, etc. In conjunction with cloud computing, edge computing helps enable industry digital transformation and facilitate business innovation by supporting customized smart products and services. Typical use cases can be found but are not limited to smart city, smart car, industrial IoT and smart grid. The use cases in Clause 7 demonstrate specific requirements for edge computing and the benefits of deploying edge computing.

7.2 Smart elevator

7.2.1 Description of the use case

7.2.1.1 Scope

Smart elevator with improved safety and reliability, and predictive maintenance.

7.2.1.2 Objectives

- Predictive maintenance of the smart elevators.
- Monitor the operating status of the smart elevators and the safety of the elevator users.
- Collaborate across the smart elevators to increase time efficiency, and optimize energy usage.

7.2.1.3 Narrative of use case

In a smart elevator solution, the elevators could be connected to a centralized data centre via the IoT gateway. A smart elevator can be monitored by several types of sensors to improve safety and reliability. The data acquired by sensors is uploaded and displayed to the operators, enabling predictive maintenance, remote monitoring and lifecycle management of the elevator components.

In the event of mechanical failure, the computation needs to be performed close to the IoT devices, e.g. sensors and actuators, to activate the IoT gateway to respond within a reasonable time. Using the artificial intelligence data processed through the cloud service, the system can predict a potential or imminent fault, and send out an alert and also activate necessary remedial action. When a fault happens, the application implemented at the ECEs can react rapidly to send out the alarm. If the connection to the cloud service fails, the data can be stored locally at the IoT gateway until the connection recovers. The IoT gateway is also able to handle some primary data processing, such as aggregation, simplification and filtering, to avoid occupying large bandwidth and bringing much pressure to the cloud service. The gateway can also act as the first line of defence against network attack.

Smart elevator solution can help vendors upgrade from the inefficient, expensive preventive maintenance to next-generation real-time, targeted, predictive maintenance, extending value from products to services.

Core values of edge computing in the smart elevator use case include the following.

- The system can manage and monitor a large number of elevators, and enable automatic operation and maintenance.
- Data analytics can locally analyse elevators' operational data, respond to requests in real time, and improve energy efficiency.
- The system can have a built-in security operation.
- The system can improve safety concerns.

7.2.2 Diagram of the use case

The concept of the smart elevator use case is as shown in Figure 7.

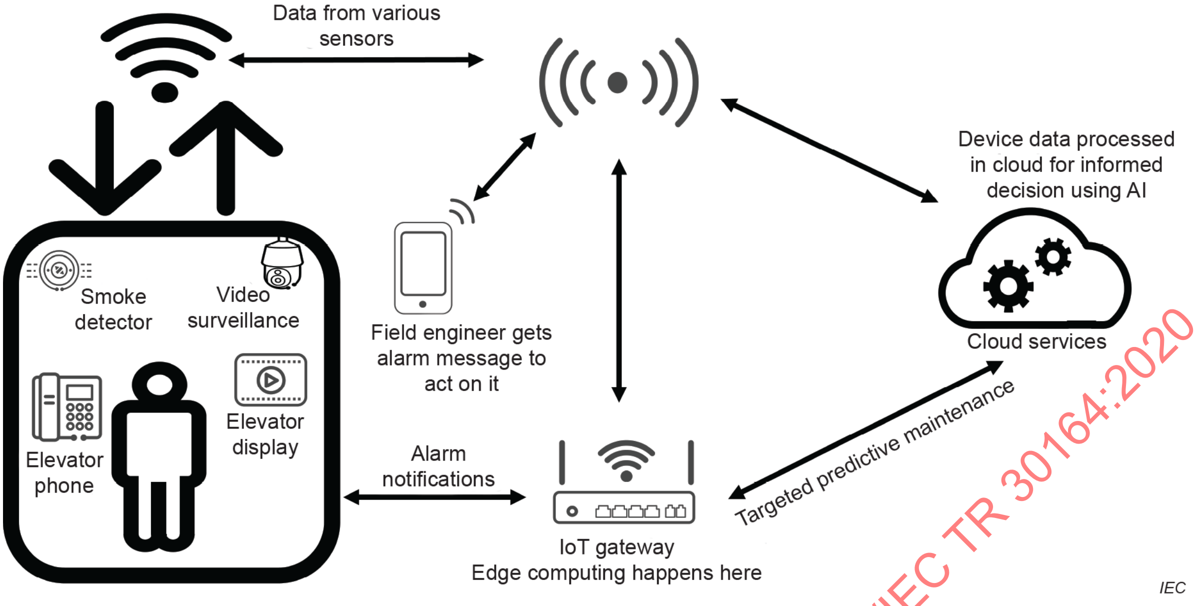

Data from various sensors
Device data processed in cloud for informed decision using AI
Smoke detector
Video surveillance
Elevator phone
Elevator display
Field engineer gets alarm message to act on it
Cloud services
Alarm notifications
Targeted predictive maintenance
IoT gateway
Edge computing happens here
IEC

Figure 7 – Concept of a smart elevator

7.2.3 Technical details

Technical details of the smart elevator use case are shown in Table 3.

Table 3 – Technical details of the elements in the smart elevator use case

Actor name	Actor type	Actor description
Cameras	Sensor	Collect real-time video monitoring signals inside/outside the smart elevator
Smoke detectors	Sensor	Detect the smoke inside/outside of the smart elevator
Elevator telephones	Sensor	The telephone inside the smart elevator for emergency use
Position sensors	Sensor	Detect the positions of the smart elevators
Pressure sensors	Sensor	Detect the pressure of the smart elevators
Temperature sensors	Sensor	Detect the temperature inside/outside of the smart elevators
Smart elevators	IoT device	The elevators equipped with edge computing concept as described in Figure 1
Smart gateway	IoT gateway	The gateway entity with computing and storage ability that connects the IoT devices and the centralized data centre
Centralized data centre	Data centre	The data centre that collects and computes the data collected from the edge gateways
User	Person	The persons that use the smart elevators

7.3 Smart video monitoring

7.3.1 Description of the use case

7.3.1.1 Scope

Smart video monitoring system can cope with large-scale high-resolution video signals at the edges to equip the smart city with safe, secure and high-efficiency environments.

7.3.1.2 Objectives

- Monitor the status of smart city utilities.
- Pre-process the high-resolution video data at the edge.
- Share the useful information across cameras and the public cloud.
- Centralized management of cameras.
- Share useful information across the smart video monitoring systems and other related platforms, such as transportation and social security platforms.
- Automate the processing and analysis of video data through the use of artificial intelligence (AI) technology

7.3.1.3 Narrative of use case

Video monitoring is widely used in public safety and security, transportation surveillance, manufacturing, etc. Hundreds of high-resolution cameras can generate a very high volume of data which could clog wide-area communication channels. Since it is not possible to monitor a city area entirely by human resources, AI at the cloud service is applied. However, the massive volume of raw data demands large storage space at the cloud service, therefore the storage has to be refreshed within a short period to accommodate newly arriving data. The processing of the massive data at the cloud service is very time consuming, which makes the timely response impossible.

A possible solution is to share responsibility at the edge. The AI can be trained at the cloud service, and a well-trained AI providing local video analysis service can be deployed at the edge which identifies people, objects (e.g. vehicles, utilities), and their properties (e.g. licence plates, coordinates, status) – see Figure 8. Only this higher-level information – like traffic accidents, criminal detection, fire detection and other emergency events – would be sent to the cloud service, providing fast response and avoiding enormous data transmission. Dependent on government regulations and local implementation, the information like normal citizens' presence may be anonymous and filtered to prevent any unnecessary collection and processing individuals’ data.

7.3.2 Diagram of the use case

The concept of the video monitoring use case is as shown in Figure 8.

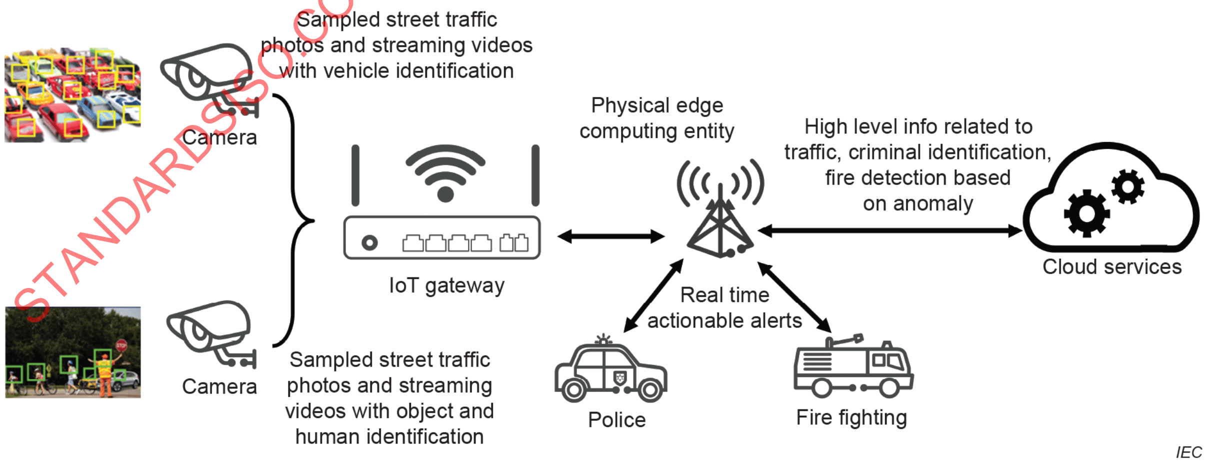

Sampled street traffic photos and streaming videos with vehicle identification
Camera
Physical edge computing entity
High level info related to traffic, criminal identification, fire detection based on anomaly
Cloud services
IoT gateway
Real time actionable alerts
Camera
Sampled street traffic photos and streaming videos with object and human identification
Police
Fire fighting
IEC

Figure 8 – Concept of video monitoring with edge computing

7.3.3 Technical details

Technical details of the elements in the video monitoring use case are shown in Table 4.

Table 4 – Technical details of the elements in the video monitoring use case

Actor name	Actor type	Actor description
Smart cameras	Sensor device, physical entity	Collect high-resolution video data across the smart city. They have limited computing and storage capability.
Smart gateway	IoT gateway, physical entity	The gateway entity with computing and storage ability that connects the IoT devices and the centralized data centre
Centralized data centre	Data centre, physical entity	The data centre that collects and computes the data collected from the edge gateways

7.4 Intelligent transportation systems

7.4.1 Description of the use case

7.4.1.1 Scope

A connected car with automotive internal control systems.

7.4.1.2 Objectives

- Monitoring of information systems such as assisted braking, fuel systems, steering and parking systems.
- Vehicle-to-vehicle communications to reduce accident probability by crash prediction and avoidance.
- Receiving traffic efficiency and management information.
- Predictive maintenance of the car.

7.4.1.3 Narrative of use case

7.4.1.3.1 General

The connected car is a good example of the merger of ICT and OT[6] (operational technologies). Car manufacturers, IT vendors and the telecommunications industry are working closely to make cars safer, more convenient, more efficient and more environmentally friendly with the help of information, telecommunication, and automatic control technologies. These improvements are reflected in the following scenarios, in which edge computing is an important aspect.

7.4.1.3.2 Automotive internal control systems

Modern cars rely heavily on information systems to monitor and control the automotive sub-systems. Assisted braking, fuel systems, steering and accident avoidance systems all are IT based. These systems all use the internal (local) network inside the automobile. Currently, many vehicles use a CAN (controller area network) bus for internal communications, but this will likely shift towards TSN (time sensitive networking) and TCP (transmission control protocol) / IP (internet protocol) based networks in the near future. Safety is a major concern and the consideration for reliability of these systems needs to be included in the following objectives.

- Computation and communication all occur locally to the vehicle and its associated LAN.
- High reliability is required.
- Latency requirements are important.
- Real-time operation is important.
- Data processing requirements vary but tend to be low.

[6] The term OT uses the definition in IIC IIoT vocabulary [15].

- Communication external to the vehicle and its associated LAN is not required, but a system that is secured from outside attacks is essential.

7.4.1.3.3 Coordinated driving (V2V)

Applications of this kind are to reduce the accident probability by crash prediction and avoidance. Vehicles will combine the local information acquired by various sensors like radars, cameras and lasers with the data communicated with neighbouring vehicles, to extend their sensing range so they can act proactively to prevent accidents and drive cooperatively. Example applications included are forward collision warning, lane change warning, cross crash warning, and cooperative adaptive cruise control. The objectives of V2V are as follows.

- The communication between vehicles needs to have low latency. Rely on resources close to the edge.
- The data bandwidth is low to medium.
- The processing is distributed amongst the vehicles.
- Reliability and safety requirements are high.
- A graceful failure mode is necessary if the communication is disrupted.
- Coordinated driving requires the automotive internal control system use case.

7.4.1.3.4 Traffic efficiency and management

In a transportation management system, by acquiring vehicles' density and speed, the traffic situation can be monitored and guidance information can be transmitted to route vehicles around congestion areas via digital traffic signs and the navigation systems inside the vehicles. In a small scale like an intersection, the ECE can act as a traffic manager, which controls the phase of the traffic lights to maximize the throughput of the intersection, according to the positions, velocities and types of incoming vehicles. The manager at the edge can avoid unnecessary start and stop, reducing fuel consumption. Combined with the coordinated driving systems, traffic throughput can be greatly increased without the need to increase the raw vehicle capacity of the road systems.

- Total bandwidth is very high for a centralized approach.
- Processing requirements near the edge are low.
- Processing requirements are high for a centralized approach.

7.4.1.3.5 Predictive maintenance

In this use case, the vehicle's data is uploaded to a centralized data centre for recording and analysis. For example, health status information of components can be used in the predictive analysis to detect impending faults. The vehicle's on-board computing resources will likely pre-process the data and avoid clogging the network.

- Bandwidth requirements can be very high.
- Data can be stored on the local vehicle information systems and transferred in bursts when a high bandwidth connection is available (for example, a traffic signal hotspot or home hotspot that becomes available once parked in owner’s driveway).

7.4.2 Diagram of the use case

Concept of the intelligent transportation system is shown in Figure 9.

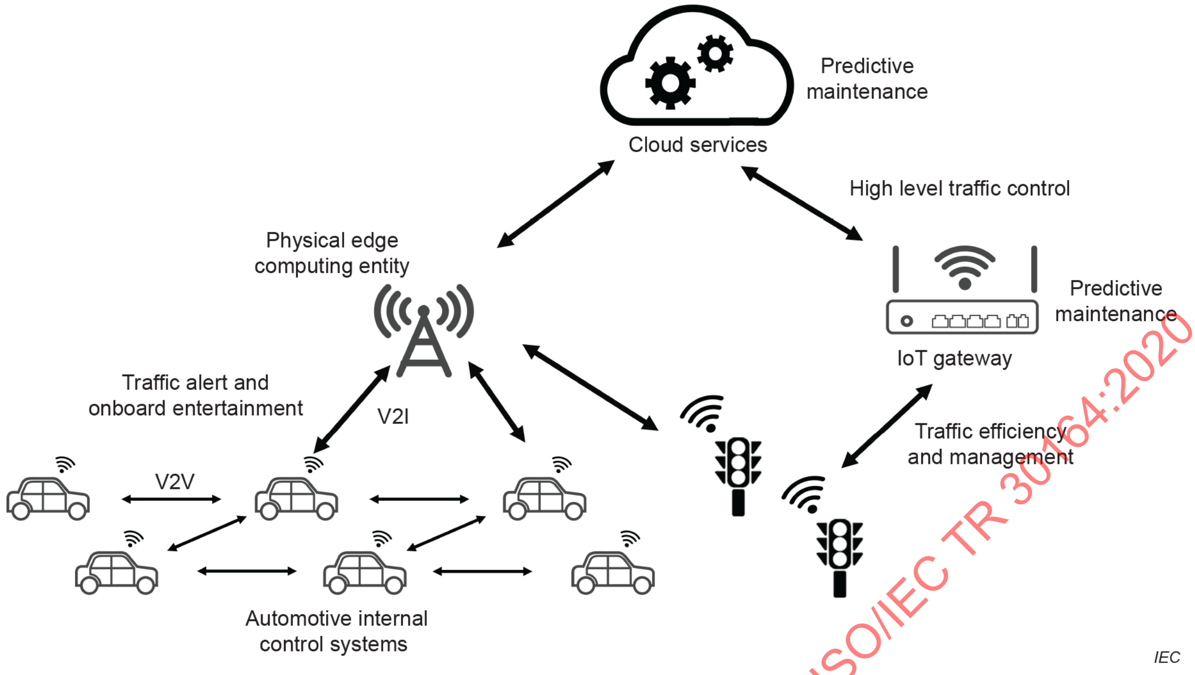

Predictive maintenance
Cloud services
High level traffic control
Physical edge computing entity
Predictive maintenance
IoT gateway
Traffic alert and onboard entertainment
V2I
Traffic efficiency and management
V2V
Automotive internal control systems
IEC

Figure 9 – Concept of intelligent transportation systems with edge computing

7.4.3 Technical details

Technical details for intelligent transportation use case are shown in Table 5.

Table 5 – Technical details for the intelligent transportation use case

Actor name	Actor type	Actor description
Braking system	Actuator, digital entity, physical entity	It includes the brake of a vehicle and the entities to control and monitor the brake.
Steering system	Actuator, digital entity, physical entity	It includes the steering of a vehicle and the entities to control and monitor the steering.
Radar	Sensor	A sensor that monitors the environment outside of the vehicle.
Camera	Sensor	A sensor that monitors the environment inside or outside of the vehicle.
Accident avoidance system	Digital entity	An automobile safety system to prevent the vehicle from a collision or other type of accident.

7.5 Process control in the smart factory

7.5.1 Description of the use case

7.5.1.1 Scope

A smart electric component manufacturing factory with process control, predictive maintenance and centralized data management.

7.5.1.2 Objectives

- Predictions of the device and network failures.
- Low latency and jitter in the control loops at the edge.
- Collect status monitoring data from the production line with high sampling rates.
- Pre-process the raw data to offload the network traffic.

- Protocol conversions to connect different ECEs within the network.

7.5.1.3 Narrative of use case:

Consumers tend to choose customized products. As a result, high-volume manufacturing is being replaced by small-quantity, multi-patch manufacturing. To accommodate this change while keeping yield levels high, control from customer purchase orders down to device test procedures needs efficient management. There could be separate software programmes controlling purchase orders (PO) and device manufacturing, but some data could be shared between them (for example, IoT identification numbers for each PO number).

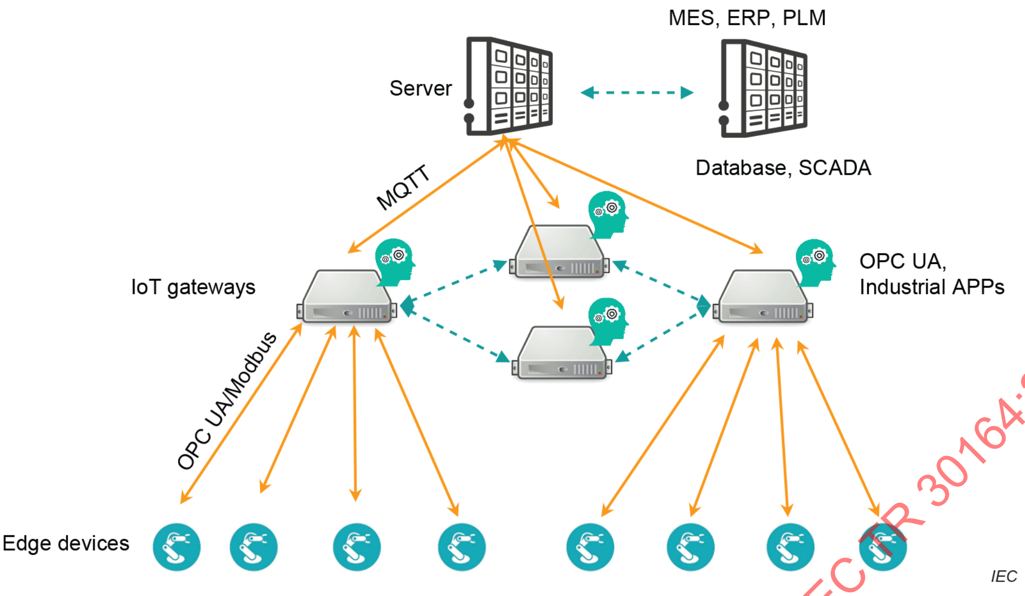
Device manufacturing generates data sets containing information such as date of manufacture down to the test results of each device. Record keeping of those results in real time helps failure analysis while improving the efficiency and reliability of the manufacturing process. Keeping all test programmes locally at a test unit keeps the fast routines locally while allowing only data of interest to be sent over a network.

Moreover, in the smart manufacturing scenario, multiple control loops and applications are included in the same network infrastructure, e.g. IIoT networking. Some of these control loops and applications are time-sensitive and have higher priority than others. The edge gateway increases the bandwidth utilization of the IIoT networks, and also reduces the impact of malfunctioning control loops and applications and isolates them rapidly. Wired or wireless technologies such as TSN, DetNet and 5G can be used to provide the system with low latency or deterministic latency.

Edge computing can help realize smart manufacturing. In an industrial system, edge computing can be seen as an industrial CPS. The system encapsulates on-site devices such as web services through industrial service adaptors. Then the services are connected to the industrial data platform using interconnection over industrial wireless and SDN networks. On a data platform, based on the production procedure and process models, edge computing dynamically manages and schedules on-site resources, and connects to systems such as MES. The industrial CPS supports flexible production plan changes based on changing production line resources. The manufacturing procedures are reset using web interoperation interfaces, realizing the plug-and-play of new devices, and reducing workload. The production cycles and material supply modes can automatically adapt to production adjustments, saving I/O configuration time caused by model shifting and material route switching. Adaptive web-based process model adjustment saves time by over 80 % for PLC (programmable logic controller) reprogramming, powering on/off, and resetting hundreds of OPC (open platform communication) variants in new process deployment.

7.5.2 Diagram of the use case

Example concept of the smart factory using IIoT is shown in Figure 10.



OPC UA open platform communications (OPC) unified architecture [12][13]

MQTT message queuing telemetry transport [16]

Figure 10 – Example concept of the smart factory using IIoT

7.5.3 Technical details

Technical details for the smart factory use case are shown in Table 6.

Table 6 – Technical details for the smart factory use case

Actor name	Actor type	Actor description
Programmable logic controller	Actuator	It is an industrial actuator for the industrial process control.
Camera	Sensor	A sensor that monitors the environment of the factory or the production line.
Edge gateway	Physical entity, IoT gateway	A physical entity with computing and storage ability that connects the IoT devices and the centralized data centre.
Server	Physical entity	A physical entity with heavy storage and heavy computing abilities that connects with lower level devices, such as IoT gateways.
Product lifecycle management (PLM) system	Digital entity	Management system for the entire lifecycle control of the products.
Supervisory control and data acquisition (SCADA) system	Digital entity	High-level process supervisory management system that controls, monitors and manages the factory.
Industrial APPs	Digital entity	Applications that run on physical entities and interact with human beings.

7.6 Centralized monitoring of power plants (CMPP)

7.6.1 Description of the use case

7.6.1.1 Scope

A centrally controlled power plant that aggregates a high number of distributed energy resources (DERs) into an integrated, monitored and managed the network.

7.6.1.2 Objectives

- Connections of DERs, including the generation and storage of electricity, into the edge computing system.
- Connections of VPP control centre, electricity trading centre and power grid dispatch control centre into the edge computing system.
- To relieve the load on the power grid during the peak load period.
- Trading of power consumption and power generation in the electricity trading centre.
- To improve electricity usage efficiency.

7.6.1.3 Narrative of use case

In a virtual power plant, DERs are integrated into a single edge computing CMPP network infrastructure. The CMPP control centre (centralized data centre) gathers data from physical/virtual edge computing entities, including sensors, control units, local control centres, and CMPP applications. It allows the CMPP operator to monitor and control the operation of each power plant and energy storage so that the predictive maintenance of the power grid can be achieved and the power grid efficiency can be improved.

The local control centre (IoT gateway), which is close to the edge, enables rapid response and real-time power dispatch. The applications on the local control centre can be customized to meet the requirements in local areas.

- Distributed response:

  Local control centre is focused on the operational control of the grid, i.e. monitoring currents and voltages in the grid and issuing commands to remote devices such as switches and transformers. Edge computing applications can be implemented on these devices to collect data and react rapidly to faults. Once a fault is detected on a segment, protection switches are activated to isolate it, and the power flow is rerouted to restore the power supply to the largest possible area.
- Distributed renewable energy:

  More and more consumer-built photovoltaic panels and wind turbines are joining in the grid. Thus the distributed generation will coexist with the current centralized generation, turning the one-way energy flow into two-way. Edge computing can be used at each generation point to manage the generated power quantity and its integration to the grid.

Benefits of using edge computing in the CMPP include the following.

- Ultra-low latency enabling real-time control features for power dispatch and other time-sensitive applications.
- Localized data analytics at the smart IoT gateway to off-load data acquisition towards the centralized data centre.
- Additional machine learning capabilities to further improve the optimization process and achieve predictive maintenance based on artificial intelligence.
- The edge-based local control units and applications are easy to upgrade, update and maintain due to the management and orchestration aspects of the virtualization technology supported by the edge system.
- The virtualized control units/edge applications enable dynamic reconfiguration of the applications to align with the operational strategies by the relocation of different applications to different IoT gateways.
- The system has localized cybersecurity features.

7.6.2 Diagram of the use case

Concept of the CMPP is as shown in Figure 11.

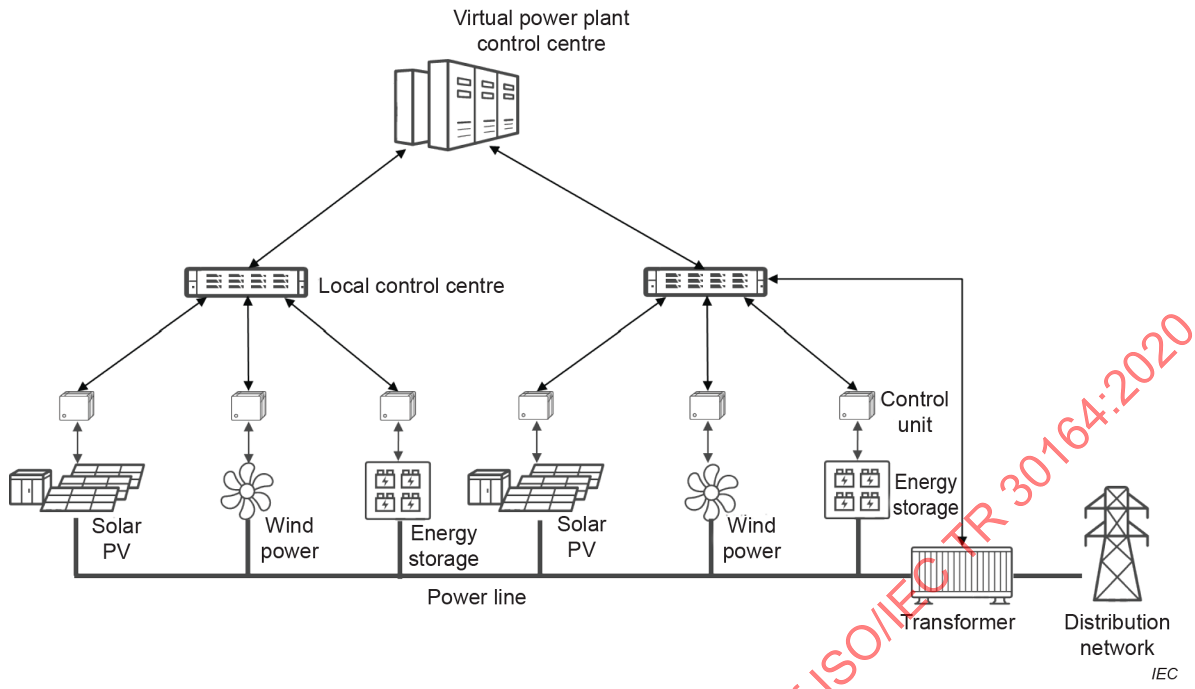


Figure 11 – Concept of centralized monitoring of power plants

7.6.3 Technical details

Technical details of the CMPP use case are shown in Table 7.

Table 7 – Technical details of the CMPP use case

Actor name	Actor type	Actor description
CMPP control centre	Centralized data centre	Manages data from local control centres and interacts with the power grid dispatch control centre and the electricity market trading centre.
Local control centre	IoT gateway	Manages data in local areas and interacts with control units and the local control centre.
Control unit	Actuator	Controls and collects data from energy resources.
Solar PV panel	IoT device	IoT device that absorbs sunlight as an energy source to generate electric power.
Wind power collecting device	IoT device	IoT device that absorbs wind power as an energy source to generate electric power.
Energy storage	IoT device	IoT device that stores energy.
Transformer	IoT device	IoT device that transforms electricity between networks.

7.7 Automated crop monitoring and management system

7.7.1 Description of the use case

7.7.1.1 Scope

Smart agriculture is the application of ICT to agriculture for growing food cleanly and sustainably, and with fewer resources and less waste than traditional farming. A smart agriculture system leverages edge computing technology to improve crop management, increase yield production, lower operational costs, and reduce applications of chemicals and fertilizers.