MCMC MTSFB TC G039:2023

TECHNICAL CODE

INDUSTRIAL INTERNET OF THINGS - CONNECTIVITY AND COMMUNICATIONS FRAMEWORK

Developed by

Malaysian Technical Standards Forum Bhd

Registered by

MCMC

Registered date: 23 May 2023

MCMC MTSFB TC G039:2023

Development of technical codes

The Communications and Multimedia Act 1998 ('the Act') provides for Technical Standards Forum designated under section 184 of the Act or the Malaysian Communications and Multimedia Commission ('the Commission') to prepare a technical code. The technical code prepared pursuant to section 185 of the Act shall consist of, at least, the requirement for network interoperability and the promotion of safety of network facilities.

Section 96 of the Act also provides for the Commission to determine a technical code in accordance with section 55 of the Act if the technical code is not developed under an applicable provision of the Act and it is unlikely to be developed by the Technical Standards Forum within a reasonable time.

In exercise of the power conferred by section 184 of the Act, the Commission has designated the Malaysian Technical Standards Forum Bhd ('MTSFB') as a Technical Standards Forum which is obligated, among others, to prepare the technical code under section 185 of the Act.

A technical code prepared in accordance with section 185 shall not be effective until it is registered by the Commission pursuant to section 95 of the Act.

For further information on the technical code, please contact:

Malaysian Communications and Multimedia Commission (MCMC)
MCMC Tower 1
Jalan Impact
Cyber 6
63000 Cyberjaya
Selangor Darul Ehsan
MALAYSIA

Tel: +60 3 8688 8000
Fax: +60 3 8688 1000
http://www.mcmc.gov.my

OR

Malaysian Technical Standards Forum Bhd (MTSFB)
MCMC Centre of Excellence (CoE)
Off Persiaran Multimedia
Jalan Impact
Cyber 6
63000 Cyberjaya
Selangor Darul Ehsan
MALAYSIA

Tel: +60 3 8320 0300
Fax: +60 3 8322 0115
http://www.mtsfb.org.my

Contents

Committee representation

This technical code was developed by Internet of Things (IoT) and Smart Sustainable Cities Working Group of the Malaysian Technical Standards Forum Bhd (MTSFB), which consists of representatives from the following organisations:

BNetwork Sdn Bhd

Celcom Axiata Berhad

Favoriot Sdn Bhd

Heriot-Watt University Malaysia

Maxis Broadband Sdn Bhd

Malaysia Digital Economy Corporation Sdn Bhd

My6 Initiative Berhad

SIRIM Berhad

Sunway University College Sdn Bhd

Telekom Malaysia Berhad

Universiti Kuala Lumpur

Universiti Putra Malaysia

Universiti Sains Islam Malaysia

Universiti Teknologi Malaysia

Universiti Teknologi MARA

Webe Digital Sdn Bhd

Foreword

This technical code for Industrial Internet of Things - Connectivity and Communications Framework ('Technical Code') was developed pursuant to section 185 of the Act 588 by the Malaysian Technical Standards Forum Bhd ('MTSFB') via its Internet of Things (IoT) and Smart Sustainable Cities Working Group.

This Technical Code shall continue to be valid and effective from the date of its registration until it is replaced or revoked.

INDUSTRIAL INTERNET OF THINGS - CONNECTIVITY AND COMMUNICATIONS FRAMEWORK

1. Scope

This Technical Code establishes the framework of Industrial Internet of Things (IIoT) connectivity and communications by considering the Malaysian context for industries and other control applications. This Technical Code can be used in various IIoT use cases such as manufacturing, agriculture, and construction.

2. Normative references

The following normative references are indispensable for the application of this Technical Code. For dated references, only the edition cited applies. For undated references, the latest edition of the normative references (including any amendments) applies.

MCMC MTSFB TC G013, *Internet of Things (IoT) - Security Management*

MCMC MTSFB TC G022, *Internet of Things - High Level Functional Architecture*

Recommendation ITU-T Y.4000, *Overview of Internet of things*

Recommendation ITU-T Y.4003, *Overview of smart manufacturing in the context of the industrial Internet of things*

3. Abbreviations

For the purposes of this Technical Code, the following abbreviations apply.

6LoWPAN	IPv6 Low Power Personal Area Network
API	Application Programming Interface
CoAP	Constrained Application Protocol
CRUD	Create/Read/Update/Delete
DDS	Data Distribution Service
GPS	Global Positioning System
HART	Highway Addressable Remote Transducer
HTTP	Hypertext Transfer Protocol
ICT	Information and Communication Technologies
IETF	Internet Engineering Task Force
ID	Identification
IIoT	Industrial Internet of Things
IoT	Internet of Things
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6

ISO	International Organization for Standardization
JSON	JavaScript Object Notation
mDNS	Multicast Domain Name System
MQTT	Message Queuing Telemetry Transport
MTBF	Mean Time Between Failure
NTP	Network Time Protocol
OASIS	Organization for the Advancement of Structured Information Standards
OMG	Object Management Group
oneM2M	one Machine-to-Machine Partnership Project
OPC	Open Platform Communications
OPC UA	Open Platform Communications United Architecture
OSI	Open System Interconnection
PTP	Precision Time Protocol
QoS	Quality of Service
REST	Representational State Transfer
SCADA	Supervisory Control and Data Acquisition
TCP	Transport Control Protocol
UDP	User Datagram Protocol
UPnP	Universal Plug and Play
URI	Uniform Resource Identifier
WPAN	Wireless Personal Area Network
XML	Extensible markup language

4. Terms and definitions

For the purposes of this Technical Code, the following terms and definitions apply.

4.1 Industrial Internet of Things (IIoT)

An Internet of Things (IoT) based enabling approach for industrial transformation by taking advantage of existing and emerging Information and Communication Technologies (ICT).

NOTES:

1. Emerging ICT include technologies for smart machines, robots, advanced industrial networks, industrial cloud computing and industrial data processing.

2. The industrial transformation enabled by the industrial internet of things empowers the industry with, but not limited to, improved efficiency, intelligent production, reduced energy consumption, advanced collaboration modes and new business models. Industrial Internet of Things enables smart manufacturing, providing enhanced capabilities in support of manufacturing.

4.2 Internet of Things (IoT)

A global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving, interoperable ICT.

NOTES:

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

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

5. Overview

The IIoT is a subset of IoT as seen throughout this Technical Code in general. As an IoT-based enabling approach, IIoT addresses the following key aspects to the applications:

a) It encompasses all kinds of sensing, measuring, controlling, and actuating devices, which enable digital transformation of industry to meet the need of real-time monitoring or near real-time of devices and processes that will improve the efficiency with the use of industrial data sets.

b) It concerns all types of connectivity and communications, such as industrial control network, enterprise network and the internet that forms end-to-end transport of the data. These include proprietary and open communications protocol, with the support of legacy systems.

c) It provides capabilities that support industrial data processing and management including, but not limited to, data conversion, data translation, data integration, data processing, storage, and analytics.

d) It provides end-to-end security to ensure that the data is securely acquired, stored, sorted, analysed, and transmitted.

e) It provides the sustainability of standards by enabling open market and competitive technological development for all level of players in IIoT.

The IIoT consists of many proprietary and open technologies for domain-specific use cases. Most of the proprietary technologies have setbacks in sharing of data, designs, architectures, and communications. The IIoT connectivity is aimed to unlock data in the isolated systems and enable data sharing and interoperability at various stages of communications.

This Technical Code defines an open IIoT connectivity and communications framework with the focus on IIoT connectivity stack and framework and support for proprietary protocols with the following goals:

a) to recommend connectivity and communications framework for IIoT; and

b) to provide a baseline framework for industrial communications with legacy support.

6. IIoT system characteristics and requirements

6.1 Basic characteristics

Basic characteristics of a smart industrial system are listed below:

a) Real-time

Smart industrial systems support real-time data processing, which includes sensing parameters and transmission of data, analysis and decision making.

b) Connectivity

The devices attached to the industrial equipment for monitoring activities should have network connectivity by using either a direct connection, or a gateway or any other intermediate devices, so that the data can be transmitted in real-time or near real-time.

c) Actuation

Some devices may have actuation capability that receive commands and controls, which could be either manual or autonomous.

6.2 High level requirements

High level requirements in a smart industrial system are given below:

a) Compatibility

A smart industrial system should be compatible with existing legacy and latest system at the time of deployment. It should allow proprietary technologies to co-exist if open technologies cannot be implemented.

b) Heterogenous connectivity capabilities

A smart industrial system should support appropriate connectivity within the system. Interworking of heterogenous communication technologies should also be supported by inherent connectors or extra module.

c) Interoperability

A smart industrial system should support interoperability at various layers including device, platform and/or application layers. Due to complexity of implementing interoperability, it is usually implemented at platform level.

d) Safety and security

Device and data security are the most important criteria for consideration. Security should be applied at various layers, such as device, storage, connectivity, platform, and application. This is to prevent the system from being compromised.

e) Reliability

The industrial system should be stable and consistent with its operational settings. The devices' efficiency should be monitored regularly to avoid disruption of services. These include components with longer Mean Time Between Failure (MTBF), better battery selection, signal power and sensitivity, automatic detection of disruptions, connectivity, and avoidance of common channels.

f) Scalability

Connectivity function should support horizontal scaling that accommodate increasing number of connectivity endpoints.

g) Heterogenous data handling capabilities

Dataset from different payload and protocol should be encrypted/decrypted or packaged/unpackaged. Appropriate data nomenclature or description must be added for clarity upon presentation for data sharing.

h) Resiliency

The system should be designed to have a resilience against any kind of attacks.

7. Connectivity and Communication

The goal of IIoT is to provide a seamless information sharing across domains and industries, which will create a new value proposition. It typically includes the integration of legacy systems with new smart systems. A connectivity architecture should allow various connectivity technologies to interoperate within an industry and across industries to support the vision of IIoT.

The types of IIoT connectivity are as follows:

a) Direct connectivity from sensor to server (see Figure 1)

Devices are coupled with sensors to measure the environment and equipment where the connectivity module sends the sensing data to the server. In this scenario, the connectivity module can be directly connected to the server.

b) Sensor to gateway to server (see Figure 2)

The architecture of this connectivity type is similar to the direct connectivity from sensor to server (as in Figure 1). However, the connectivity module in the device can only be connected to a gateway, which then forwards the data to the server. This is because the device is fitted with a short-range connectivity module.

c) Multiple connectivity types with multiple gateways (see Figure 3)

Multiple connectivity is where multiple systems are connected to separate gateways, which will then process and forward the data to the server. This could be open systems and/or proprietary systems.

d) Interoperability gateway (see Figure 4)

The interoperability gateway provides seamless information flow from one system to another. The universal gateway should have various connectivity modules to allow connectivity from different devices using different connectivity. For example, one device may use Zigbee, whereas another device may use Wireless Personal Area Network (WPAN) to connect to the universal gateway. Besides the connectivity support, the gateway should also be capable in translating the data structure to an appropriate structure.

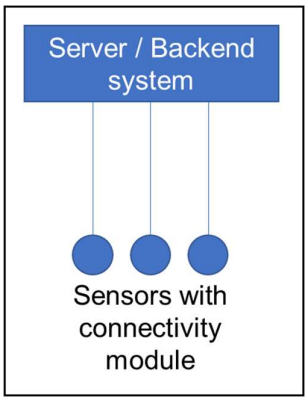

Figure 1. Direct connectivity from sensor to server

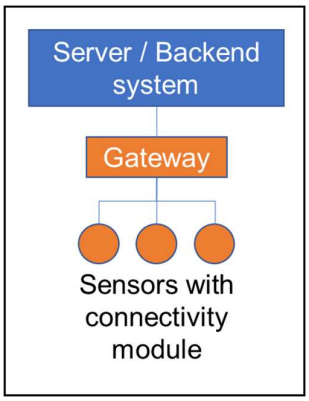

Figure 2. Sensor to gateway to server

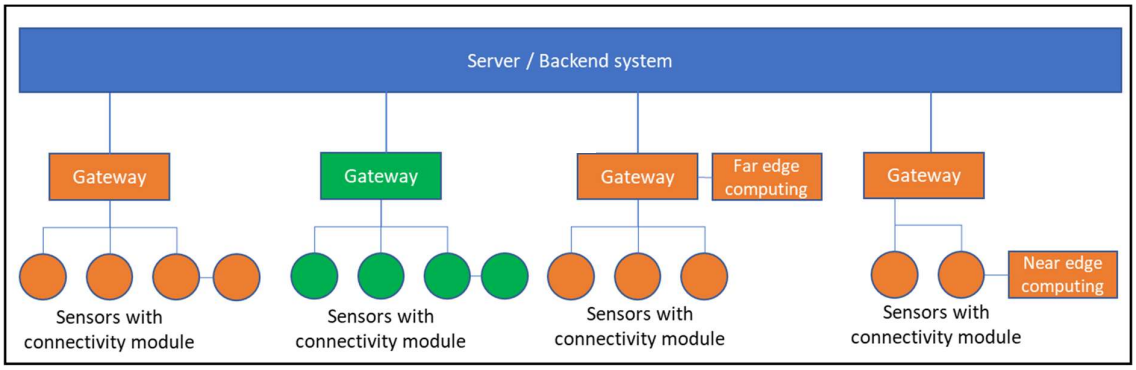

Figure 3. Multiple connectivity types with multiple gateways

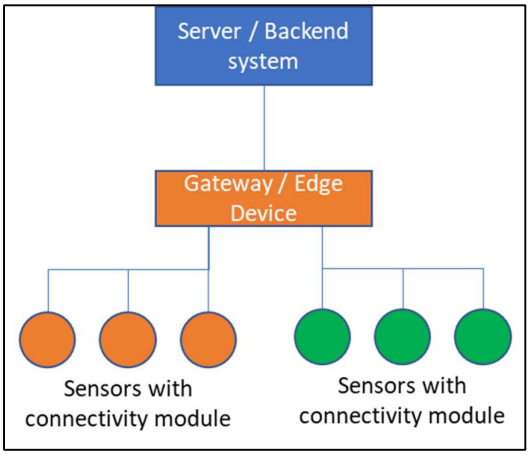

Figure 4. Interoperability gateway

7.1 Core standards criteria

New technologies and standards on connectivity should be integrated with legacy systems to ensure there will be no disruption in the operation of a solution. A connectivity core standard should be aligned to the priorities on the requirements and ecosystem in its functional domain. A connectivity core standard should have the following:

a) provide syntactic interoperability, which is a structured data type shared between endpoints;

b) be an open standard, which is managed by recognised standard bodies;

c) support proprietary standards if open standards can't be implemented;

d) applicable horizontally across industries and neutral in its applicability;

e) stable and deployed in systems across multiple industries; and

f) support all core functions of a connectivity framework.

7.2 IIoT connectivity framework

The connectivity framework layer provides logical data exchange service between each endpoint. It is a logical functional layer on top of the transport layer and should be agnostics of technologies. The main role of this framework is to provide syntactic interoperability among the endpoints. Data is structured in a common, unambiguous data format, independent of endpoint implementation, and decoupled from the hardware and software.

A key benefit of the framework is to abstract and hide the implementation of the various functions by using the Application Programming Interfaces (APIs) for the communication. This will reduce the cost and ease the deployment.

The key connectivity framework functions include data resource model, publish-subscribe, request-reply, data security and quality of service as shown in Figure 5.

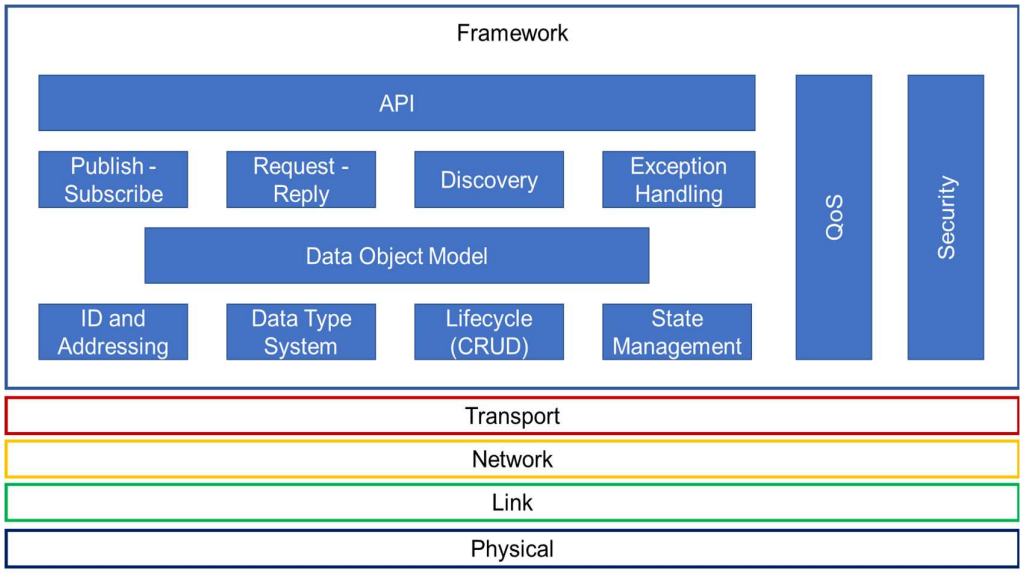

Figure 5. Connectivity in framework layer functions

a) Data object model

Data object is a structured collection of fields, which may be hierarchical and statically or dynamically typed. A connectivity framework propagates the changes to data-object amongst the participants. Data models for different application areas or industries are usually mapped into the abstract data-objects provided by a connectivity framework.

b) Identification (ID) and addressing

An Identification (ID) is used to address a data object and read-write fields in the data object representation. It could be any one of the following:

i) An implicit ID based on specially marked fields in the data object representation;

ii) An explicit ID field in the data-object representation; or

iii) A Uniform Resource Identifier (URI) within the namespace of a device or application or network endpoints.

c) Data type

A data type is a syntactic constraint placed upon the interpretation of data. The data type may be object-oriented, like data types found in statically typed programming such as C and C++, or object-based, like the dynamic data types in programming languages such as JavaScript and Python. Data types provide means of managing the evolution of data types, which include versioning and assignability rules across versions. It also defines serialised data format in communication and storage.

d) Data resource lifecycle

A framework may consist of 4 critical operations of lifecycle of a data object, which commonly known as Create/Read/Update/Delete (CRUD):

i) Create (C) - create a new data object;

ii) Read (R) - observe the state of a data object;

iii) Update (U) - update the state of a data object; and

iv) Delete (D) - delete a data object.

e) State management

Data published could be in real-time, near real-time or it could have been stored in local storage. The framework should handle these different needs of data management, including synchronising endpoints when a connection is re-established.

f) Publish-subscribe

In a publish-subscribe function, a component will publish the data on a topic and another component will subscribe to the topic to receive updates. An endpoint may operate both as publisher and subscriber. It is usually for one-to-one or one-to-many types of data distribution.

g) Request-reply

This function is also referred to as request-response. In this type of data exchange, a requestor can initiate requests to be fulfilled by an endpoint in the replier role. The process could be synchronous, where a requestor waits for the replies before continuing the next request, or it could be asynchronous, where a requestor can have multiple requests and replies processed as they are received.

h) Discovery

The connectivity framework provides mechanism to discover the following:

i) publish-subscribe topics and the associated Quality of service (QoS);

ii) request-reply services and the associated QoS;

iii) data types; and

iv) endpoints in a data exchange.

Discovery, authentication and access to a service should be automated.

i) Exception handling

Exception handling is a function that handles any disruption in connectivity, which is caused by disconnected or intermittent link, switching of network, changes in network configuration, failure of remote endpoint or non-responsive endpoints.

j) QoS

IIoT can have varying requirements for data delivery and it can be configured using the QoS function. The data delivery could be best-effort or reliable delivery, depending on the deployment requirements. The QoS will be impacted by the frequency of data transmission, size of data, network quality and others.

k) Security

Connectivity framework security functions provide the ability to ensure data confidentiality, integrity and authenticity. Security can be applied at various stages of the data exchanges.

l) API

IIoT software should provide an API to support the design and implementation of the application. APIs should be in a standardised format and independent of programming languages to ease the development and deployment of IIoT services and applications.

7.3 IIoT connectivity transport and network layers

The connectivity transport and network layers provide a logical transport and network connecting the endpoints. The key role of the connectivity transport layer is to provide technical interoperability between the endpoints. They consist of endpoint addressing, connectedness, prioritisation, timing and synchronisation, security, communication modes and messaging protocol as illustrated in Figure 6.

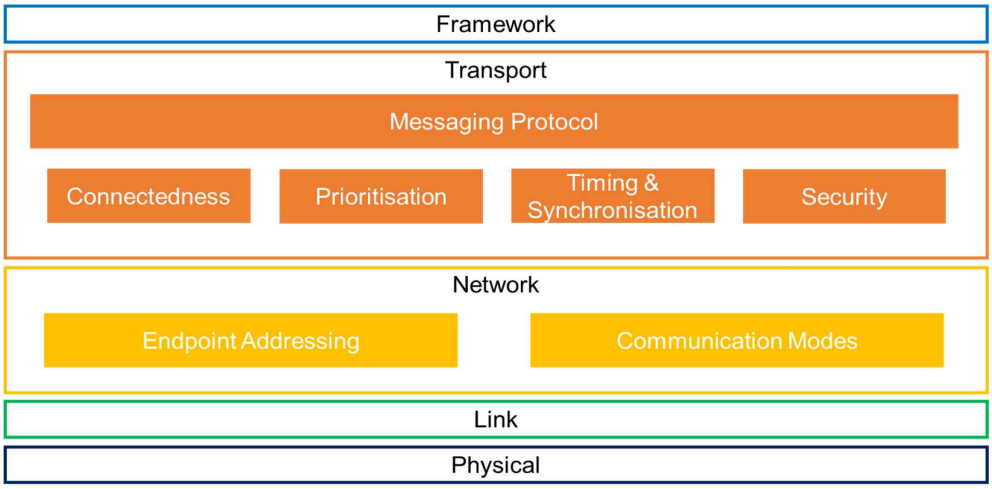

Figure 6. Connectivity transport and network layer function

The transport layer functions consist of the following:

a) Messaging protocol

The messaging protocol is the protocol that describes the format and behaviour of the messages exchanged between the endpoints. It may include discovery, authentication, session establishment, message retry and acknowledgement, fragmentation and reassembly of large messages, data encoding and message reorder. Messaging protocol can be configured and optimised for different network layer configurations based on some parameters, such as bandwidth, round-trip time, maximum message size and QoS.

b) Connectedness

There are 2 main types of communications that are usually used for delivering packets across the network among endpoints, which are connection-oriented and connectionless services. For example, in the Open System Interconnection (OSI) layer, Transport Control Protocol (TCP) is considered connection-oriented, whereas the User Datagram Protocol (UDP) is connectionless. The usage of the packet delivery service depends on the IIoT services. In some solutions, every packet sent should be acknowledged (connection-oriented).

c) Prioritisation

IIoT solutions should ensure that critical data is delivered ahead of non-critical data. Prioritisation function will provide the ability to prioritise some messages over others.

d) Timing and synchronisation

Time synchronisation in IIoT should be provided to synchronise local endpoint clocks. There are a few methods that are already in use nowadays, which include Network Time Protocol (NTP) or Precision Time Protocol (PTP) and Global Positioning System (GPS) clocks. The importance of having this function is to record time of data generated for better implementation of application, synchronisation and security.

e) Security

Transport layer security involves both the messaging protocol and the network layer security. Both messaging protocol and the network layer security should provide mechanisms for endpoint authentication, message encryption and message authentication. There are various security mechanisms available for IIoT. Please refer to the MCMC MTSFB TC G013 for more details.

The network layer functions consist of the following:

a) Communication modes

Connectivity transport in IIoT may support the following communication modes:

i) Unicast (one-to-one communication);

ii) Multicast (one-to-many communication); and

iii) Broadcast (one-to-all communication).

b) Endpoint addressing

An address is used to identify a node for communication purposes. The address should be unique, whether it is local or global, and this setting depends on the deployment of the IIoT solution. It is recommended for the address to be either an Internet Protocol version 4 (IPv4) or Internet Protocol version 6 (IPv6) address. However, in some cases, other types of addressing are also possible.

7.4 Connectivity standards

There are various types of network connectivity in IIoT, ranging from short range to long range. The details of connectivity types are described in Clause 5 of MCMC MTSFB TC G022. Figure 7 below illustrates the main IIoT connectivity framework, which shows that the framework is agnostics of the various industry verticals.

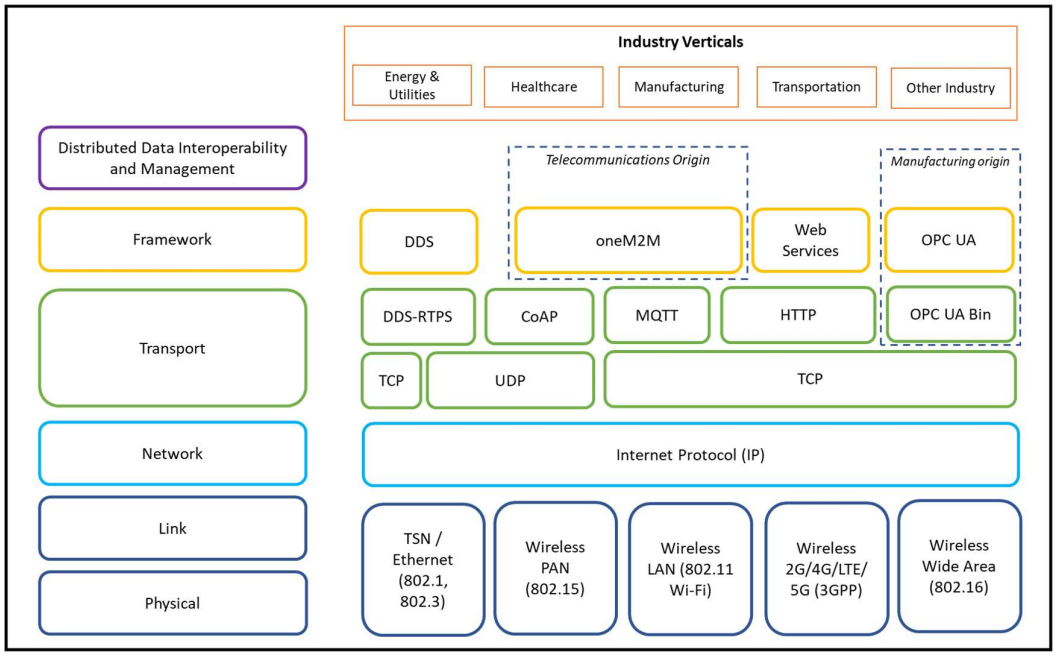

Figure 7. IIoT connectivity framework

IIoT solutions may utilise the different types of frameworks shown in Figure 7, which were introduced by various organisations and consortiums.

7.4.1 IIoT communication framework standards

There are several communication protocols to choose from in designing an IIoT solution. These communications protocols operate on the OSI application layer but it is also combined into the transport layer as demonstrated in Figure 7. Some of the common communication standards are:

a) Constrained Application Protocol (CoAP)

CoAP is a lightweight and specialised web transfer protocol that is suitable for IIoT constrained devices and constrained networks. CoAP provides a request-response interaction model between application endpoints, supports built-in discovery of services and resources, and includes key concepts of the Web such as URIs and Internet media types. CoAP is designed to easily interface with Hypertext Transfer Protocol (HTTP) for integration with the Web while meeting specialized requirements, such as multicast support, very low overhead and simplicity for constrained environments.

b) Message Queuing Telemetry Transport (MQTT)

MQTT is a lightweight publish-subscribe messaging protocol for communication in IoT context and approved by Organization for the Advancement of Structured Information Standards (OASIS) and International Organization for Standardization (ISO) (ISO/IEC 20922). It is ideal for connecting remote devices with a small code footprint and minimal network bandwidth.

MQTT defines 2 types of network entities, namely:

i) MQTT broker: a server that receives all messages from the clients and then routes them to the appropriate destination clients; and

ii) MQTT client: any device, from a micro controller up to a fully-fledged server, which runs a MQTT library and connects to a MQTT broker over a network.

c) Fieldbus technologies

Fieldbus is an industrial network system for real-time distributed control, which is usually used to connect instruments in a manufacturing plant. There are several types of fieldbus technologies such as Profibus (Profinet), Ethernet/Internet Protocol (IP), Modbus, Highway Addressable Remote Transducer (HART), and Foundation Fieldbus family.

d) HTTP

HTTP is an application layer protocol for distributed, collaborative and hypermedia information systems. HTTP is used with the Representational State Transfer (REST) standard in IIoT. The operations (i.e. HTTP methods) that are available are GET, HEAD, POST, PUT, PATCH, DELETE, CONNECT, OPTIONS and TRACE.

7.5 Connectivity characteristics

Connectivity characteristics can be used to identify the essential features of the networking or communications connectivity between devices and the IIoT system, which is demonstrated in Table 1 below.

Table 1. Connectivity category and characteristics

No.	Category	Characteristics
1.	Mechanism	a) Wired i) Electric/electronic ii) Fibre optic b) Wireless i) Spectrum ii) Light iii) Sound
2.	Nature	a) Real-time b) Near real-time c) Asynchronous i) Periodic (push) ii) Scheduled (push) iii) On request (pull)
3.	Initiation	a) By sending device b) By receiving device
4.	Protocols	a) Network protocols (IPv4, IPv6) b) Discovery procotols (mDNS, UPnP) c) Data and application protocols (MQTT, CoAP, HTTP)
5.	Link security	a) Authentication i) Mutual ii) One-way iii) None b) Identification i) Mutual ii) One-way iii) None c) Authorisation i) Required ii) Not required d) Encrypted i) Full ii) Data only iii) None

a) Mechanism

Mechanism refers to the physical mechanism used to convey any communications to or from a device. There are 2 mechanism classes, which are wired and wireless connectivity. In some deployment, the devices store data locally without any connectivity. This is not considered as an IIoT solution in this Technical Code.

b) Nature

Nature signifies whether the connectivity is real-time or near real-time, where continuous connectivity is required; or whether data can be stored and forwarded, either scheduled or on request.

c) Initiation

This relates to how communication is initiated, which is by either the sender device or the receiving device. It could be a sensor node, a server or a backend system.

d) Protocols

These are protocols to establish and manage the link and the exchange of information or data. Examples of these classes are:

i) network - IPv4, IPv6, IPv6 Low Power Personal Area Network (6LoWPAN);

ii) discovery - Multicast Domain Name System (mDNS), HyperCat and Universal Plug and Play (UPnP); and

iii) data and application protocols - MQTT, CoAP, HTTP.

e) Link security

This focuses on the level of security and trust involved in the establishment and operations of the connectivity.

All of the characteristics should be identified prior to the development and deployment of an IIoT service or solution.

7.5.1 IIoT connectivity framework standards

Some of the IIoT connectivity framework standards consist of the following:

a) Data Distribution Service (DDS)

This is an open connectivity framework, which is managed by the Object Management Group (OMG). DDS is generally used in the control, application, information, operations domains and sometimes in the business domain. Its main purpose is to connect IIoT components to other components, which enables a real-time system and system-of-systems. Some of the applications that use DDS include hospital integration, rail control / railway management, asset tracking, oil and gas, Supervisory Control and Data Acquisition (SCADA), ship management, robotics and defence.

b) Web services using Hypertext Transfer Protocol (HTTP)

This is the most commonly used application-specific connectivity framework. It relies on the REST style of architecture using HTTP connectivity transport standard to exchange data that requires TCP transport protocol. Data in HTTP are represented in a textual form (either JavaScript Object Notation (JSON) or Extensible Markup Language (XML)) and embedded in hypermedia context. It is not efficient for device-to-device communications and not suitable for real-time communications. Internet Engineering Task Force (IETF) maintains the HTTP open standard specifications.

c) Open Platform Communications Unified Architecture (OPC UA)

Open Platform Communications (OPC) is a connectivity framework standard used in the manufacturing industry. OPC is designed to support multiple transports. The transport mappings are defined for TCP with an OPC UA binary encoding connectivity transport standard or HTTP connectivity transport as shown in Figure 7. The detailed information on OPC UA is available at OPC Foundation’s website.

d) one Machine-to-Machine Partnership Project (oneM2M)

The oneM2M is managed by a partnership of regional international standards industry organisations in the telecommunications industry. oneM2M provides a common service layer between an application and transport. It can be used in various industries and uses RESTful APIs for interactions. It can also use HTTP, CoAP, MQTT and WebSockets for connectivity. Detailed information about its usage is available at oneM2M's portal.

Bibliography

[1] Object Management Group: DDS Portal - Data Distribution Service

[2] Internet Engineering Task Force (IETF)

[3] OPC Foundation: OPC Unified Architecture, retrieved on 2022-02-05
https://opcfoundation.org/about/opc-technologies/opc-ua

[4] Standard for M2M and the Internet of Things
https://www.onem2m.org/

Acknowledgements

Members of the Internet of Things (IoT) and Smart Sustainable Cities Working Group

Dr Gopinath Rao Sinniah (Chairman/Draft Lead)	Favoriot Sdn Bhd
Mr Mohd Zakir Hussin Baharuddin (Vice Chairman)	Telekom Malaysia Berhad
Mr Mohamad Norzamir Mat Taib/ Ms Norkhadhra Nawawi (Secretariat)	Malaysian Technical Standards Forum Bhd
Mr Tharmaindran K.Gannasin	BNetwork Sdn Bhd
Mr Danial Fairuz Nassurudin	Celcom Axiata Berhad
Ts Pang Jia Yew	Heriot-Watt University Malaysia
Mr Cheong Gze Wei	Maxis Broadband Sdn Bhd
Mr Wong Chup Woh/ Ms Yesotha Surendran/ Mr Jesse Chooi Tze Kheong	Malaysia Digital Economy Corporation Sdn Bhd
Ts Adil Hidayat Rosli	My6 Initiative Berhad
Ms Wan Zarina Wan Abdullah	SIRIM Berhad
Prof Ts Dr Lau Sian Lun	Sunway University College Sdn Bhd
Ms Roziyani Rawi	Universiti Kuala Lumpur
Prof Ir Dr Aduwati Sali/ Prof Dr Borhanuddin Mohd Ali/ Dr Thinagaran Perumal	Universiti Putra Malaysia
Prof Ir Dr Hafizal Mohamad	Universiti Sains Islam Malaysia
Dr Azizul Azizan	Universiti Teknologi Malaysia
Ir Dr Yusnani Mohd Yussoff	Universiti Teknologi MARA
Ms Siti Najwa Muhammad	Webe Digital Sdn Bhd

By invitation:

Dr Navaneethan C. Arjuman	NLTVC Education Sdn Bhd