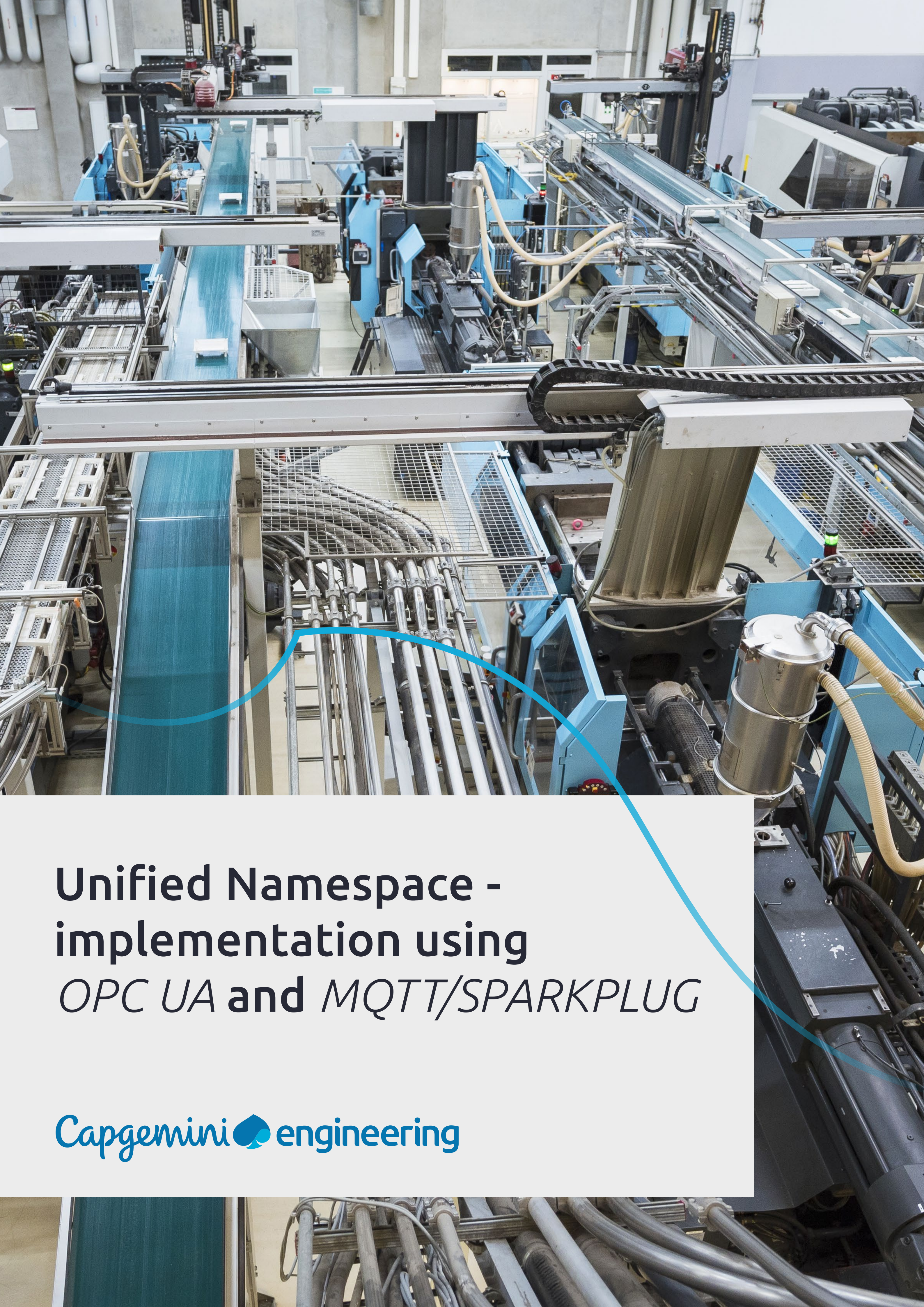


# Unified Namespace - implementation using *OPC UA* and *MQTT/SPARKPLUG*

Capgemini engineering

# Table of Contents

1. OPC UA in a nutshell 3

2. Unified namespace (UNS), OPC UA and MQTT 5

3. Implementation examples using OPC UA 10

4. Sources 13

# 1 OPC UA in a nutshell

*OPC UA is a **platform-independent standard** through which various kinds of systems and devices can communicate by sending request and response Messages between Clients and Servers or NetworkMessages between Publishers and Subscribers over various types of networks. It supports **robust, secure communication** that assures the identity of OPC UA Applications and resists attacks.*

**– OPC UA Specification[18] - Part 1**

OPC UA (Open Platform Communication Unified Architecture) has been created in 2006, as an improvement of OPC Classic, to implement a more modern version including security by design, platform-independence, and the use of an object-oriented model to describe objects and types of the server. OPC UA defined a set of services to interact between a client and a server and evolved in 2017 to specify of publisher/subscriber implementation of the standard.

To keep it simple, OPC UA can be seen as the definition of 3 pillars:

- A communication protocol, which defines standard and secure communication between applications,
- A modeling language, allowing standard modeling and address space representation,
- A conformance model, that guaranty its conformity and usability from sensor to cloud.

## Standard and secure communication

OPC UA defines 2 communication paradigms, which are client / server and publisher / subscriber.

The client / server mode has first been introduced as an improvement of OPC Classic:

- Redefines services,
- Deal with portability,
- Improve interoperability,
- Deal with security.

The specification has then been enriched with a publisher / subscriber mode for OPC UA:

- Defines the OPC UA model to configure a publisher and a subscriber,
- Define the way to bind it to objects in the server,
- Define a standard encoding to use for objects and types,
- Define an extensible set of usable protocols, among which MQTT, AMQP or UADP.

## Standard modelling and address space representation

IEC62541 defines OPC UA's meta model. This meta model defines the way to represent data using an object-oriented model. In OPC UA model, objects are composed of objects, variables, and methods. The model describes types as well as instances.

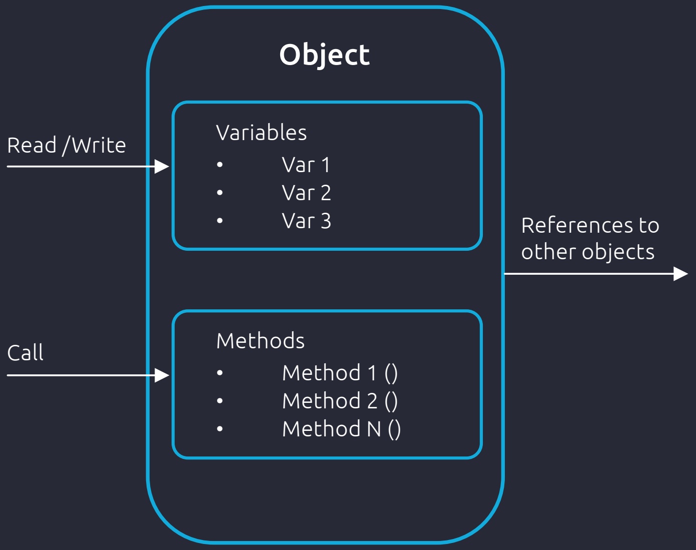
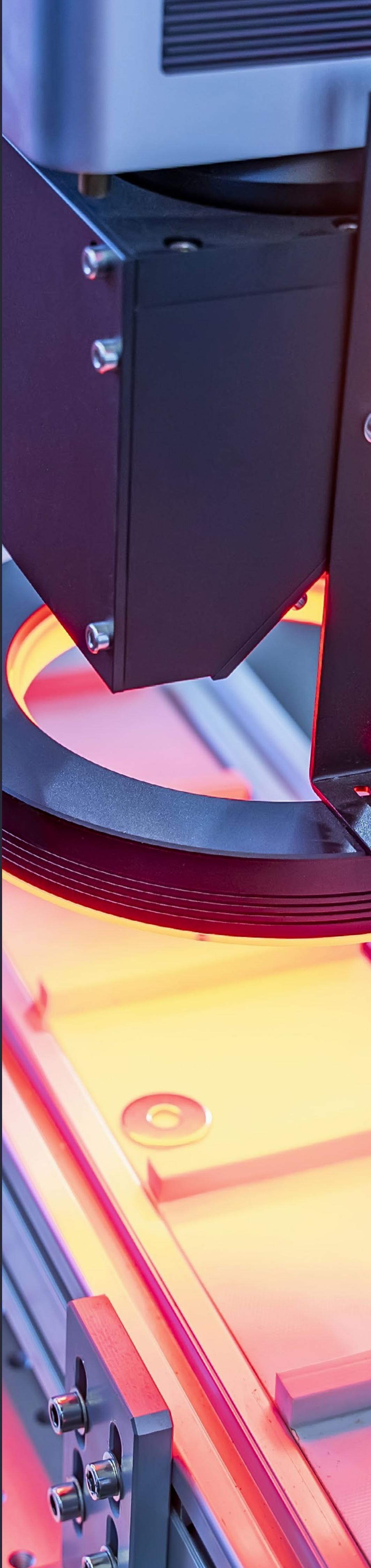


Figure 1 - OPC UA Object model



The set of existing types and instances are represented in a server as a graph, with nodes (types and instances) bound by references. Nodes have attributes, that define what they are, and references that link them to other nodes.

On top of the meta model defined by the OPC UA specification, some working groups have gathered to standardize the use of OPC UA modeling in their fields.

Those initiatives gave birth to companion specifications. The list of these specifications includes several information models aiming at the description of machines such as OPC UA for Machinery, OPC UA for Robotics, OPC UA for MachineTools, etc. Some of these models are dedicated to more general models such as OPC UA for I4 Asset Administration Shell, ISA-95 Common Object Model, OPC UA for AutomationML, etc.

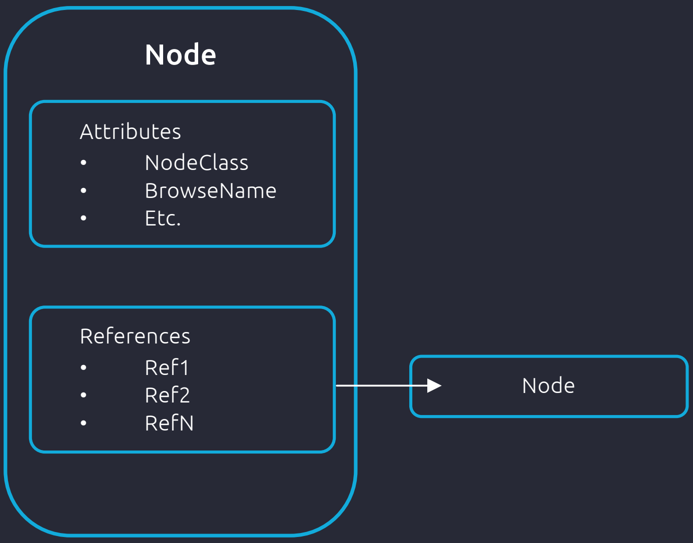
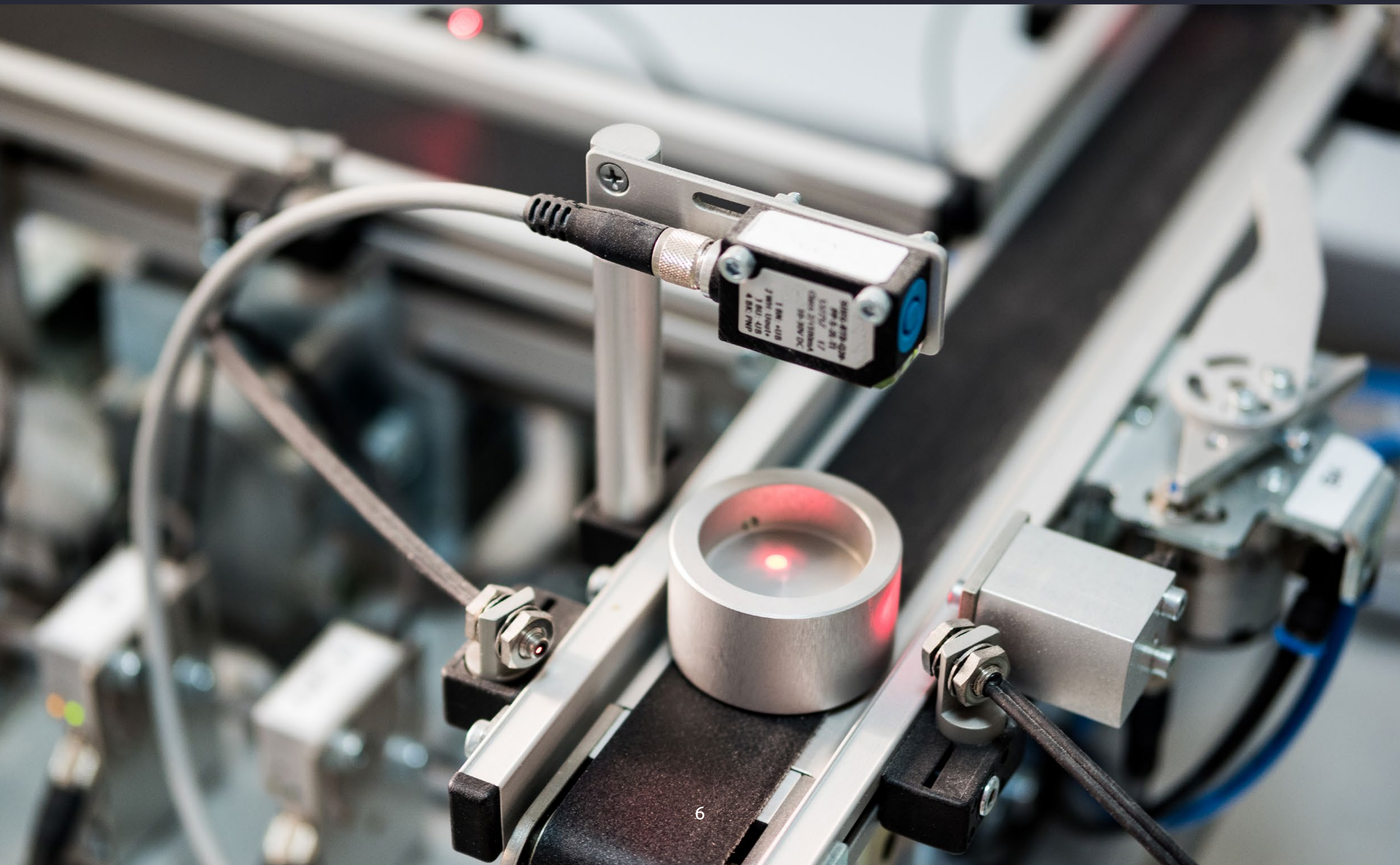


Figure 2 - OPC UA Node model



## Interoperability from sensor to cloud

OPC UA specification part 7 defines a conformance model, which defines several profiles for UA applications. Application can be certified one or more of existing profiles by passing certification tests in a lab.

Profiles are composed of mandatory and optional facets, which are themselves made of conformance units which are a set of features that can be tested as a single entity.

For example, server profiles define the minimum requirements for an OPC UA server to be certified. Several levels exist to allow server to be adapted to their environment. Server profiles defined in 1.04 version of the specification are:

- Nano Embedded Device 2017 Server Profile,
- Micro Embedded Device 2017 Server Profile,
- Embedded 2017 Server Profile,
- Standard 2017 Server Profile.

In the same way, part 7 of UA specification defines profiles for clients, global services and PubSub communications. OPC UA is compatible by design with several transport protocol stacks. The core specification of OPC UA is independent of the underlying communication protocol. Furthermore, the specification allows the addition of new protocols bindings if necessary.

The sensor to cloud communication will exclusively use the OPC UA PubSub **broker-based communication model**, while communication between PLCs and Edge gateways or local SCADA software will focus on the OPC UA Client Server communication model.

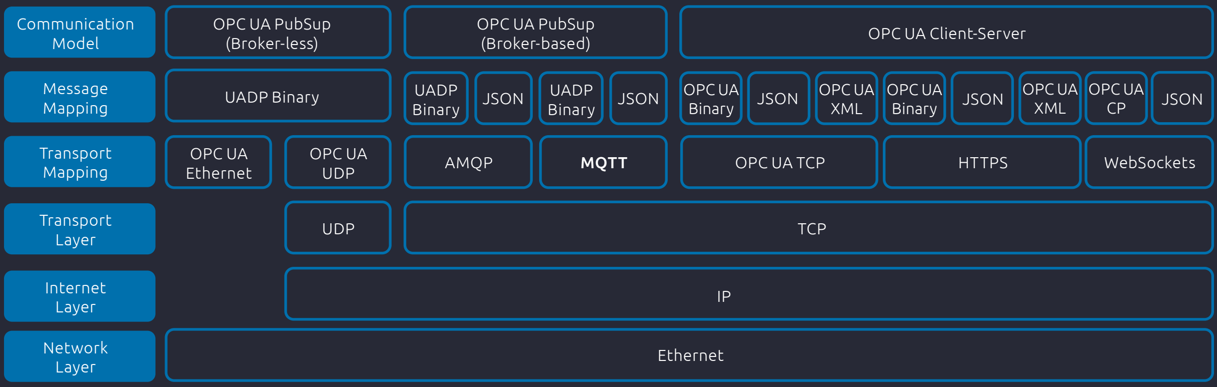


Figure 3 - OPC UA Transport protocol stacks

# 2 Unified namespace (UNS), OPC UA and MQTT

## What is UNS?

### Simple definition

UNS (Unified Namespace) is a consistent data management approach and architecture aiming at the creation of a single source of truth for sharing industrial data for the whole enterprise. It is a way to get rid of silos and centralize data in a common namespace organized using a common hierarchy.

There is no standard for UNS, it can be achieved by using whatever technology, data model, organization, etc. However, for UNS to be implemented successfully, it requires its architecture to be:

- Open and interoperable: shall be implemented based on open standards, without being dependent on any vendor, hardware, software, etc.
- Lightweight and scalable: Shall minimize overhead and scale well to centralize data at the enterprise level,
- Well structured: to fit the enterprise model and be able to retrieve data effectively.

For more information, publications on UNS are available on the internet, such as[3][4].

### Limits and critics

There are some limitations in the definition of an UNS:

- Lack of standard and common definition,
- The name "Unified Namespace" is confusing. Because of the multiplicity of business domains, of manufacturers, of technologies, etc. there will never be one unique namespace, at best a coherent aggregation of namespaces. If we try to build a digital twin of an industrial plant, we will get a graph of objects which is more complex than a hierarchy. Depending on the business process and use case, several hierarchies can be exposed from the same graph.
- UNS is presented as an architecture, which is not an accurate definition. For sure, it is a concept which can be implemented by different logical and technical architectures but not an architecture by itself.

UNS does not define the technology nor the format of the messages. UNS just states that the topic tree must follow some minimal rules. UNS recommends to structure the topics tree as an ISA95 hierarchy.

The main issue is that the contents and semantics of the topics is described by UNS. In a nutshell, Interoperability is defined at the technical level but not defined at the semantic level, according to a very common definition of interoperability by the European Commission[26].

The Interoperability at the semantic level is left to the choice of the implementer, which is not a problem in itself but the UNS alone is not enough to specify a data exchange between 2 parties. That's why some others technical standards such as OPC UA PubSub or MQTT Sparkplug are adding to the UNS, the missing semantic interoperability which is missing by default.

## What is MQTT?

MQTT should not be compared to OPC UA, since it is purely a transport protocol, where OPC UA defines data model too, and can use MQTT as a transport mean [1].

*MQTT is an OASIS standard messaging protocol for the Internet of Things (IoT). It is designed as an extremely lightweight publish/subscribe messaging transport that is ideal for connecting remote devices with a small code footprint and minimal network bandwidth. MQTT today is used in a wide variety of industries, such as automotive, manufacturing, telecommunications, oil and gas, etc.*

**– https://mqtt.org/**

MQTT stands for Message Queueing Telemetry Transport. It has been designed in 1999 to transport telemetry data over satellite. It has become an OASIS standard in 2014.

The protocol enables bi-directional communication between clients, via the use of a broker. A client can be publisher or subscriber (or both). The broker organizes data using topics, to which publisher sends data, and from which subscribers can get data.

MQTT is a lightweight protocol, however it is also designed to be reliable, even on unreliable network, it implements Quality of Service, and can handle security using TLS or authentication protocols, such as OAuth.

As stated in the MQTT specification[2], **the content of the message's payload, as well as its format, is application specific.**

## OPC UA over MQTT

Since its version 1.04, OPC UA define an alternative to its traditional client / server communication protocol. The part 14 of the OPC UA specification[18], defines a PubSub version of the protocol, which can use MQTT as its transport protocol.

MQTT is briefly described in §2.2, which enumerates MQTT advantages, and its main drawback: the lack of data representation standard for its payload. The power of OPC UA PubSub resides in the fact that it uses MQTT, with its forces, and adds its standard modelling and data representation capabilities.

OPC UA applications can be publishers, subscribers, or both. The standard defines the model used to configure a server as a publisher or a subscriber, as well as a way to represent both data and metadata sent or received via MQTT with a Json based format. Metadata can be sent by the publisher to the broker, so that a subscriber can retrieve it and use it, or by other means, such as described in OPC UA specification[18] part 14.

## UNS implementation

Since UNS is a concept, there are several candidate architectures to implement an UNS.

The most important constraint is that UNS requires to use a message broker. A message broker is logical software component which can instantiated with many technologies:

• A MQTT broker,
• An AMQP broker,
• A NATS broker,
• A Kafka broker,
• Any cloud broker such as Microsoft Azure EventHub, Amazon Event Bridge or Google Pub Sub.

We focus only in this chapter on some possible candidate architectures.

## UNS Architecture based on OPC UA

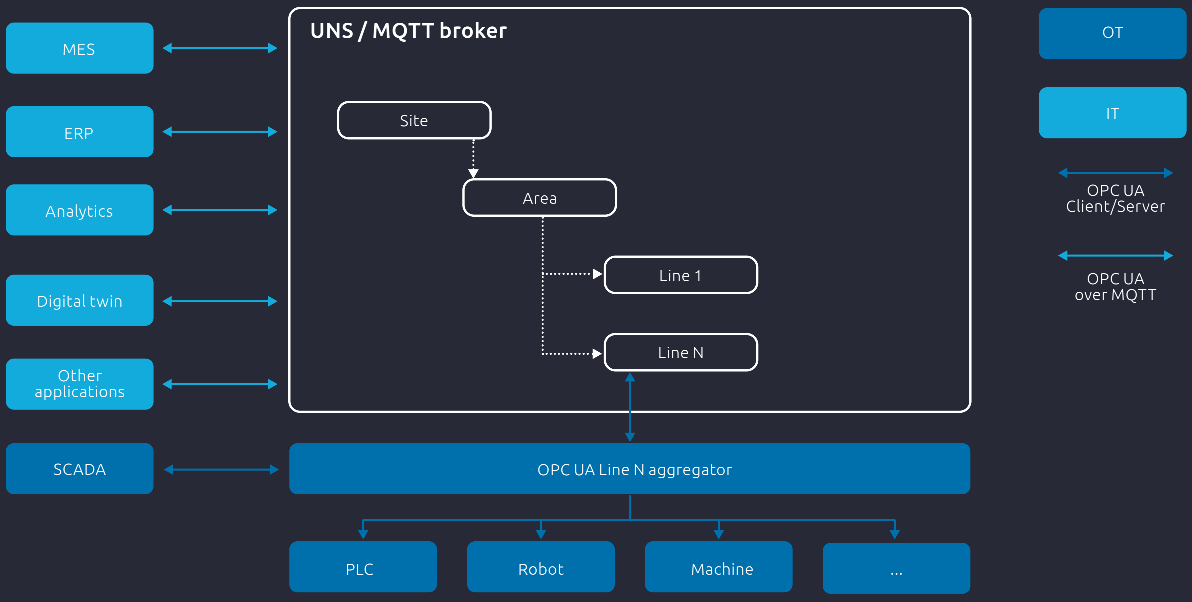


Figure 4 - UNS implementation based on OPC UA over MQTT

The UNS is implemented using a MQTT broker or any other broker running on premise or in the cloud. The broker is just the place where data is exposed by publishers and consumed with subscribers. The data distributed using the standard, lightweight and scalable MQTT protocol. ISA-95 is used to organize data in the broker, with topics that match with its organization (site/area/line).

Assets composing a line are found as sub-topics of the line. Payload is encoded in Json format, using OPC UA PubSub encoding.

The data model of the messages is specified by configuration of the OPC UA Publisher. Each OPC UA datasource exposes a set of objects defined by information models. The OPC UA Cloud Library contains more than 100 models that contain already a lot of semantic objects. This is a huge advantage of this architecture compared to others technical solutions which are not relying on any semantic standardized model.

The Json encoding is also standard and supported by the main cloud providers, and industrial software vendors. Using OPC UA is then a great choice to have a digital continuity without having to implement data convertor at several levels.

Data from the OT side of the infrastructure is exposed via an OPC UA server, with publisher and subscriber capabilities so that it can send data to the UNS, as well as retrieve it. The OPC UA aggregation server get its data from PLC, machines, etc. which expose their data using OPC UA or some others industrial communication protocols.

## Other alternatives

### MQTT + Sparkplug

Sparkplug B is a protocol which specifications[6] are open-source and hosted by the Eclipse foundation. Its goal is to define a topic structure, a payload format and session state management on top of MQTT, so that it can be used generically for IIOT.

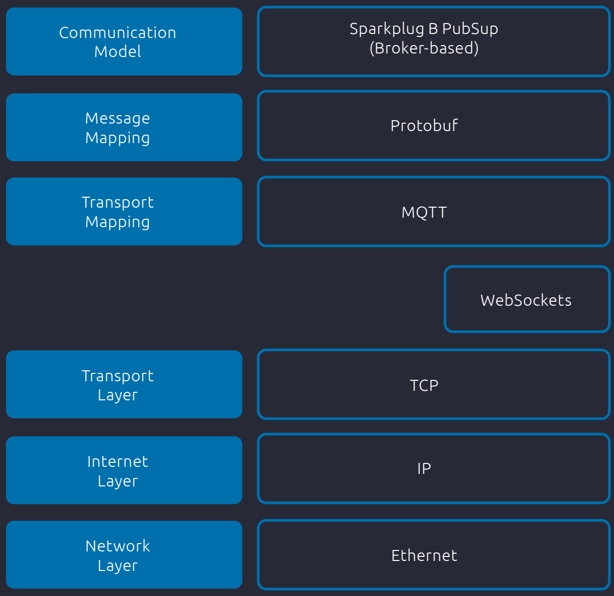


Figure 5 - Sparkplug B network stack

The payload encoding is binary and based on protobuf [19]. This serialization protocol is quite compact, and very efficient for small messages that can be loaded into memory at once. The payload is defined to expose a value, with its timestamp, data type and a name, and a quality (depending on the type of message).

[5]Shows example of implementation of UNS using MQTT and sparkplug.

## Comparison

This comparison shows the strength and weaknesses of the two technical solutions (OPC UA PubSub and MQTT Sparkplug) to implement an UNS, it does not aim to compare all possible connectivity scenarios inside shop floor networks, such as OPC UA over TSN, which is designed for field level connectivity.

The scope of UNS is only focused on the data connectivity between the EDGE level and the IT scope.

| | Sparkplug + MQTT | OPC UA over MQTT | OPC UA Client / Server |
|---|---|---|---|
| Open Standard | 🟢 | 🟢 | 🟢 |
| Lightweight | 🟢 | 🟢 | 🔴 |
| Scalable | 🟢 | 🟢 | 🟠 |
| Define data structure | 🟢 | 🟢 | 🟢 |
| Define payload structure | 🟢 | 🟢 | 🟢 |
| Standard of deserialization | 🟠 | 🟢 | 🟢 |
| Embedded data description | 🔴 | 🟢 | 🟢 |
| Define common models for devices, industrial sectors, etc. | 🔴 | 🟢 | 🟢 |
| IEC / ISO standard | 🟢 | 🟢 | 🟢 |
| Global Community | 🟠 [23] (15 companies) | 🟢 [24] (920 companies) | 🟢 |
| Supported by PLC vendors | 🟠 [9] [10] | 🟠 [7] [8] [25] | 🟢 |
| Supported by IOT gateways | 🟠 [13] | 🟠 [11] [12] [15] [25] | 🟢 |
| Supporter by SCADA vendors | 🟠 [14] [16] | 🟠 [22] | 🟢 |
| Supported by MES vendors | 🟠 [14] [16] | 🟠 [22] | 🟢 |
| Supported for edge / cloud communication | 🟢 | 🟢 | 🔴 |
| Tests and conformance | 🟠 [22] | 🟢 subset of CTT applicable | 🟢 [28] CTT with 1800 test scripts |

## Comparison between SparkplugB and OPC UA Pub Sub.

### UNS with SparkplugB  PROs and CONs

PROs

- If we have already some devices with SparkplugB enabled it makes senses to keep the SparkplugB messages and exposes them to the MQTT Broker.
- The network bandwidth is more optimized because of the binary format which is an advantage for certain use cases.

CONs

- All subscribers applications must be able to understand and decode the Binary messages which is not the case for major ERP or MES commercial software.
- Most PLC and industrial devices are not supporting yet SparkplugB.

### UNS with OPC UA PROs and CONs

PROs

- Most PLC are supporting OPC UA out of the box either with the legacy client/server architecture or with the OPC UA PubSub architecture. It makes sense to keep the same standard from the device to the UNS to avoid any protocol translation.
- The structure of the objects and messages are defined by standardized information models.
- The JSON encoding of the messages published on the UNS is understood by any IT application.
- The DATA as well as the structure (METADATA) of the messages can be published on the UNS allowing subscribers to dynamically discover the structure of the messages.

CONs

- JSON format is more verbose than a Binary format which could be an issue for some use cases.
- The PubSub part of the OPC UA specification is relatively new but the number of compatible publishers and subscribers available on the market is rapidly increasing.

### Conclusion regarding the implementation of an UNS

To ease engineering, deployment, and interoperability, it is preferable to rely on international standards, such as ISA95 or ISA88, and use collectively defined and standard models, such as OPC UA companion specifications as defined by VDMA or others international standardization bodies.

OPC UA is already widely used in its client / server form to connect devices, PLC, and machines to aggregation servers, SCADA, and MES. Moreover, aggregation servers are often able to be OPC UA publishers over MQTT, which allows to connect all the data they manage to the UNS. In cases where an OPC UA server would not able itself to be OPC UA publisher, tools, such as OPC Publisher[17] can send data to a unique data source.

Using OPC UA at every level ensures the continuity of data without requiring translation or data transformation.

For those reasons the choice of a hybrid architecture using OPC UA in both its client/server and PubSub forms to implement UNS seems the best choice compared to MQTT Sparkplug.

# 3 Implementation examples using OPC UA

## Manufacturing ontologies

The digital twin consortium defines architecture and tools tom implement manufacturing ontologies [20]. It aims at creating an UNS, by collecting data in plants, centralizing it and standardizing data models using OPC UA. Data is transferred from edge to cloud using OPC UA over MQTT. The architecture defined is based on Azure components, and consumed by both Azure, custom and third-party tools.

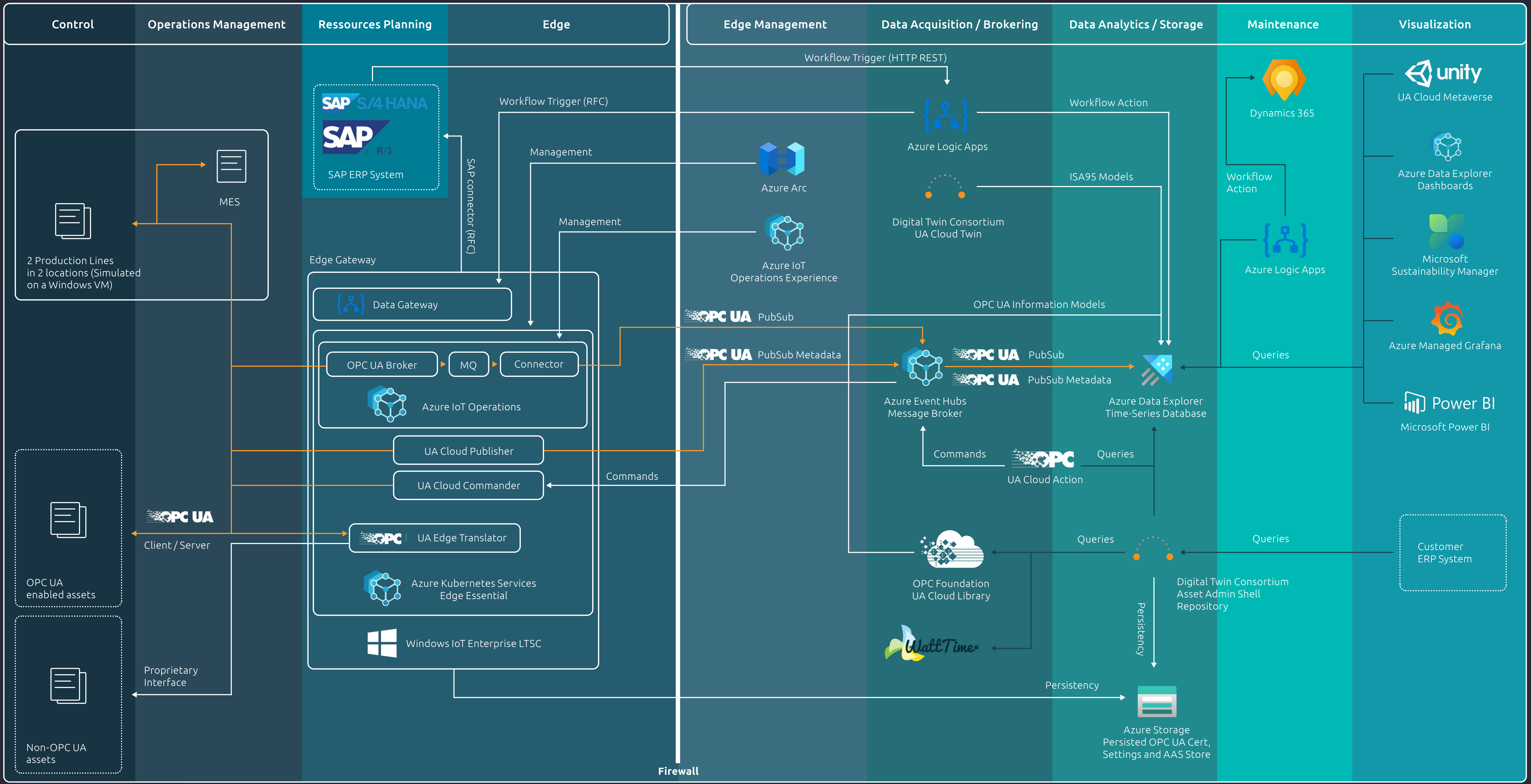


Figure 6 - Manufacturing ontologies detailed architecture by the digital twin consortium [20]

## SEMANTIC & SECURITY MANAGEMENT

Capgemini developed a tool suite to deal with data connectivity and standardization, based on OPC UA, both client/server and PubSub over MQTT. This tool suite accelerates the implementation of connectivity, and interoperability between all components of an industrial information system. The main principles is to be able to provide a Model (Ontology) covering vertical and horizontal integration.

• Vertical: enabling the IT/OT convergence
• Horizontal: facilitating the continuity between Engineering and Manufacturing or Operation

Some typical use cases for product or industrial systems

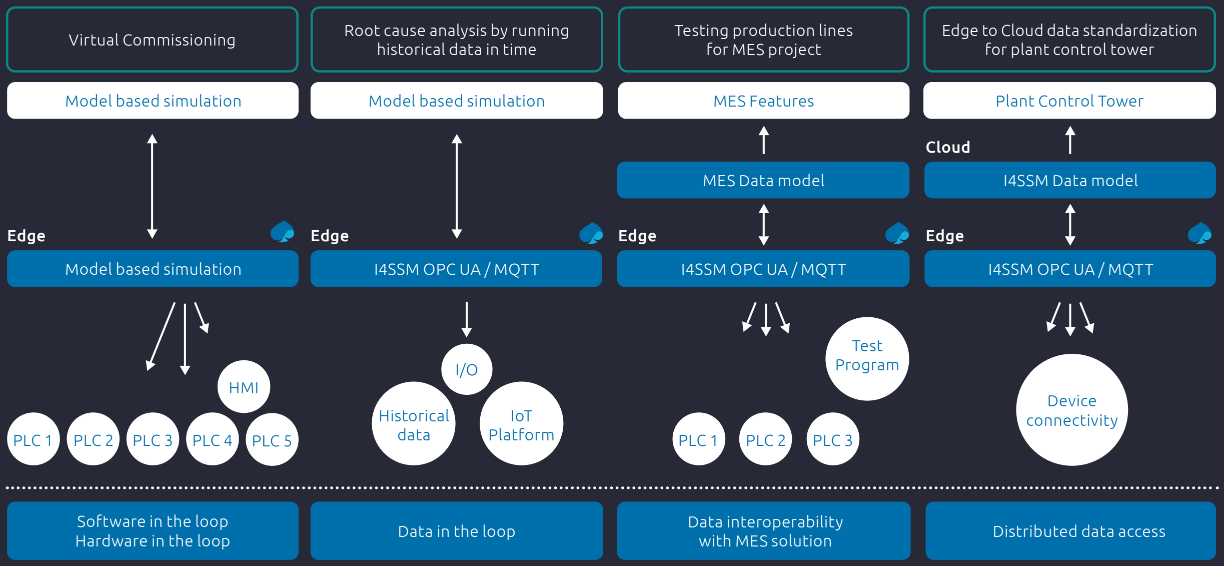


Figure 7 - Typical use cases of I4SSM[21]

The semantic gateway collects data using OPC UA aggregation server, standardize and contextualize it by relying on OPC UA companion specifications. This data can be accessed using OPC UA client/server protocol or published to an MQTT broker using OPC UA over MQTT. The association of the OPC UA aggregator and the MQTT broker is the core of the UNS at the factory level.

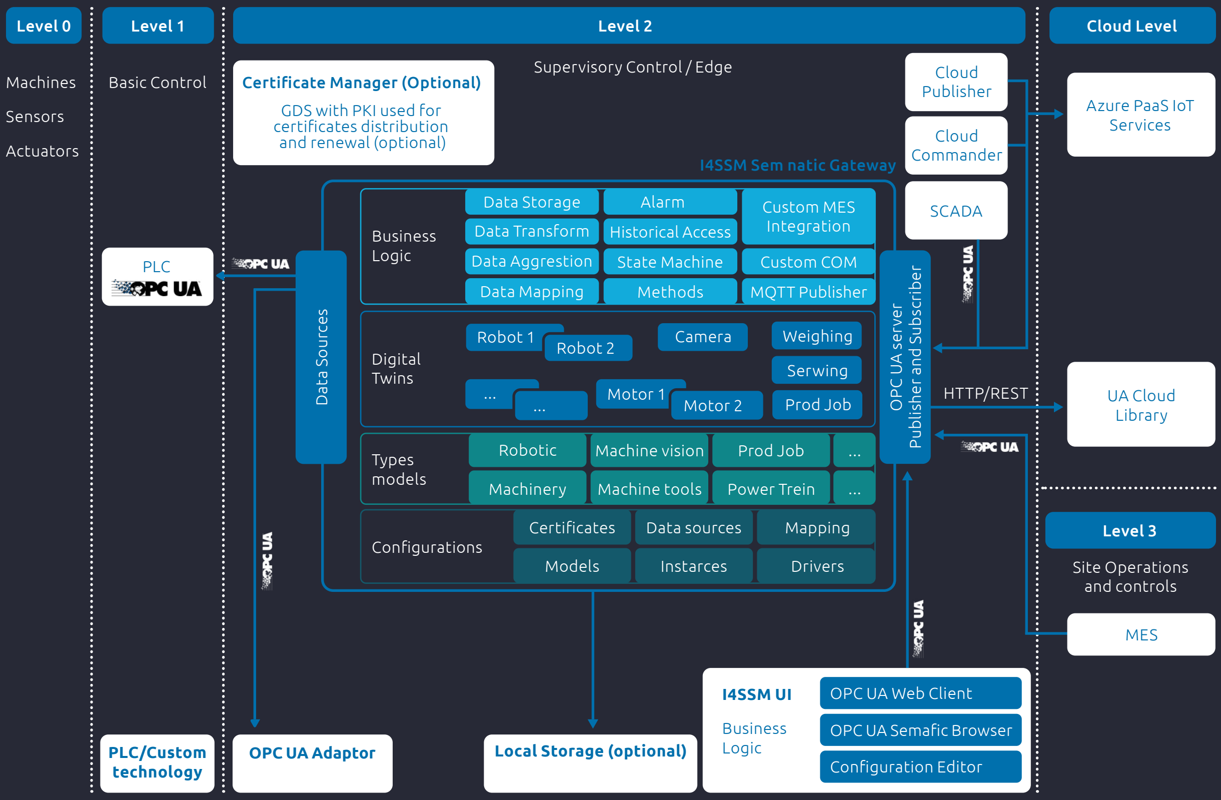


Figure 8 - Technical deep dive for I4SSM

Using OPC UA over MQTT, data is sent to the cloud using the cloud publisher. The cloud infrastructure is then used to store data, using proprietary services (such as Azure PaaS IoT Services) or tools independent from the cloud provider (such as MongoDB Atlas).

# 4 Sources

[1]. Erich BARNSTEDT, OPC UA vs MQTT vs Asset Administration Shell - oh my!, OPC Day International 2022, https://www.youtube.com/watch?v=ezSRRaG1fAE

[2]. OASIS Standard, MQTT Version 5.0, 2019, https://docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.html

[3]. Kudzai MANDITEREZA, What is Unified Namespace (UNS) and Why Does it Matter?, 2022, https://www.hivemq.com/article/what-is-unified-namespace-uns-iiot-industry-40/

[4]. Kudzai MANDITEREZA, How Does a Unified Namespace (UNS) Work?, 2022, https://www.hivemq.com/article/how-does-unified-namespace-uns-work-iiot-industry-40/

[5]. Kudzai MANDITEREZA, Implementing Unified Namespace (UNS) With MQTT Sparkplug, 2022, https://www.hivemq.com/article/implementing-unified-namespace-uns-mqtt-sparkplug/

[6]. Eclipse Sparkplug contributors, Sparkplug specification, 2022, https://sparkplug.eclipse.org/specification/

[7]. Siemens, https://support.industry.siemens.com/cs/document/109797826/opc-ua-pubsub-with-simatic-s7-1500-based-on-mqtt?dti=0&lc=en-UA

[8]. Beckhoff, https://www.beckhoff.com/en-us/products/automation/opc-ua/

[9]. Wago, https://www.wago.com/us/sparkplug

[10]. Opto 22, https://info.opto22.com/industrial-strength-mqtt-sparkplugb-white-paper-offer?utm_term=mqtt&utm_campaign=MQTT+White+Paper&utm_source=adwords&utm_medium=ppc&hsa_tgt=kwd-311606492836&hsa_grp=120094652874&hsa_src=g&hsa_net=adwords&hsa_mt=b&hsa_ver=3&hsa_ad=524177526419&hsa_acc=2623395143&hsa_kw=mqtt&hsa_cam=12208085915&gad_source=1&gclid=EAIaIQobChMI6MKd442chAMVG0NBAh3pDgivEAAYASAAEgK_i_D_BwE

[11]. Melissa TOPP (ICONICS), OPC UA + MQTT = A Popular Combination for IoT Expansion, 2019, https://opcconnect.opcfoundation.org/2019/09/opc-ua-mqtt-a-popular-combination-for-iot-expansion/

[12]. ABB, https://new.abb.com/products/robotics/controllers/opc-ua

[13]. ELPRO, https://elprotech.com/product/industrial-wireless/industrial-wireless-modems/industrial-iot-connectivity/215u-2-wi-fi-i-o-and-gateway/

[14]. AVEVA, 2023, https://www.aveva.com/en/perspectives/blog/new-ways-to-connect-with-aveva-communication-drivers-2023/

[15]. Bosch Rexroth, https://developer.community.boschrexroth.com/t5/Store-and-How-to/ctrlX-AUTOMATION-OPC-UA-Pub-Sub/ba-p/60172

[16]. HiveMQ,
https://www.hivemq.com/solutions/technology/mqtt-sparkplug/

[17]. Microsoft, OPC Publisher github,
https://github.com/Azure/Industrial-IoT/blob/main/readme.md

[18]. OPC Foundation, IEC62541,
https://reference.opcfoundation.org/

[19]. Google, Protobuf documentation,
https://protobuf.dev/overview/

[20]. Digital twin consortium, Manufacturing ontologies Github,
https://github.com/digitaltwinconsortium/ManufacturingOntologies

[21]. Sylvain MARCHAND et al. (Capgemini Engineering), A model driven and hardware agnostic approach of Virtual commissioning, ARCI 2022,
https://arci-conference.com/arci_2022_proceedings.html

[22]. ICONICS, Subscribing to OPC UA Pub Sub Data,
https://iconics.com/en-us/Resources/ICONICS-Institute/Units/Cloud-IoT/6180/Subscribing-to-OPC-UA-Pub-Sub-Data

[23] Sparkplug Members,
https://sparkplug.eclipse.org/index.html#members

[24] OPC Foundation Members,
https://opcfoundation.org/members

[25] Bosch CtrlX,
https://apps.boschrexroth.com/microsites/ctrlx-automation/en/

[26] European Interoperability Framework (EIF),
https://joinup.ec.europa.eu/collection/nifo-national-interoperability-framework-observatory/solution/european-interoperability-framework-eif-toolbox/levels-interoperability

[27] Eclipse Sparkplug TCK Process Version 1,
https://sparkplug.eclipse.org/specification/tck-process/

[28] OPC Foundation Certification Specifications,
https://opcfoundation.org/developer-tools/certification-specifications

## About Capgemini Engineering

World leader in engineering and R&D services, Capgemini Engineering combines its broad industry knowledge and cutting-edge technologies in digital and software to support the convergence of the physical and digital worlds. Coupled with the capabilities of the rest of the Group, it helps clients to accelerate their journey towards Intelligent Industry. Capgemini Engineering has 65,000 engineer and scientist team members in over 30 countries across sectors including Aeronautics, Space, Defense, Naval, Automotive, Rail, Infrastructure & Transportation, Energy, Utilities & Chemicals, Life Sciences, Communications, Semiconductor & Electronics, Industrial & Consumer, Software & Internet.

Capgemini Engineering is an integral part of the Capgemini Group, a global business and technology transformation partner, helping organizations to accelerate their dual transition to a digital and sustainable world, while creating tangible impact for enterprises and society. It is a responsible and diverse group of 340,000 team members in more than 50 countries. With its strong over 55-year heritage, Capgemini is trusted by its clients to unlock the value of technology to address the entire breadth of their business needs. It delivers end-to-end services and solutions leveraging strengths from strategy and design to engineering, all fueled by its market leading capabilities in AI, cloud and data, combined with its deep industry expertise and partner ecosystem. The Group reported 2023 global revenues of €22.5 billion.

Get the future you want

More information | www.capgemini.com

Capgemini engineering