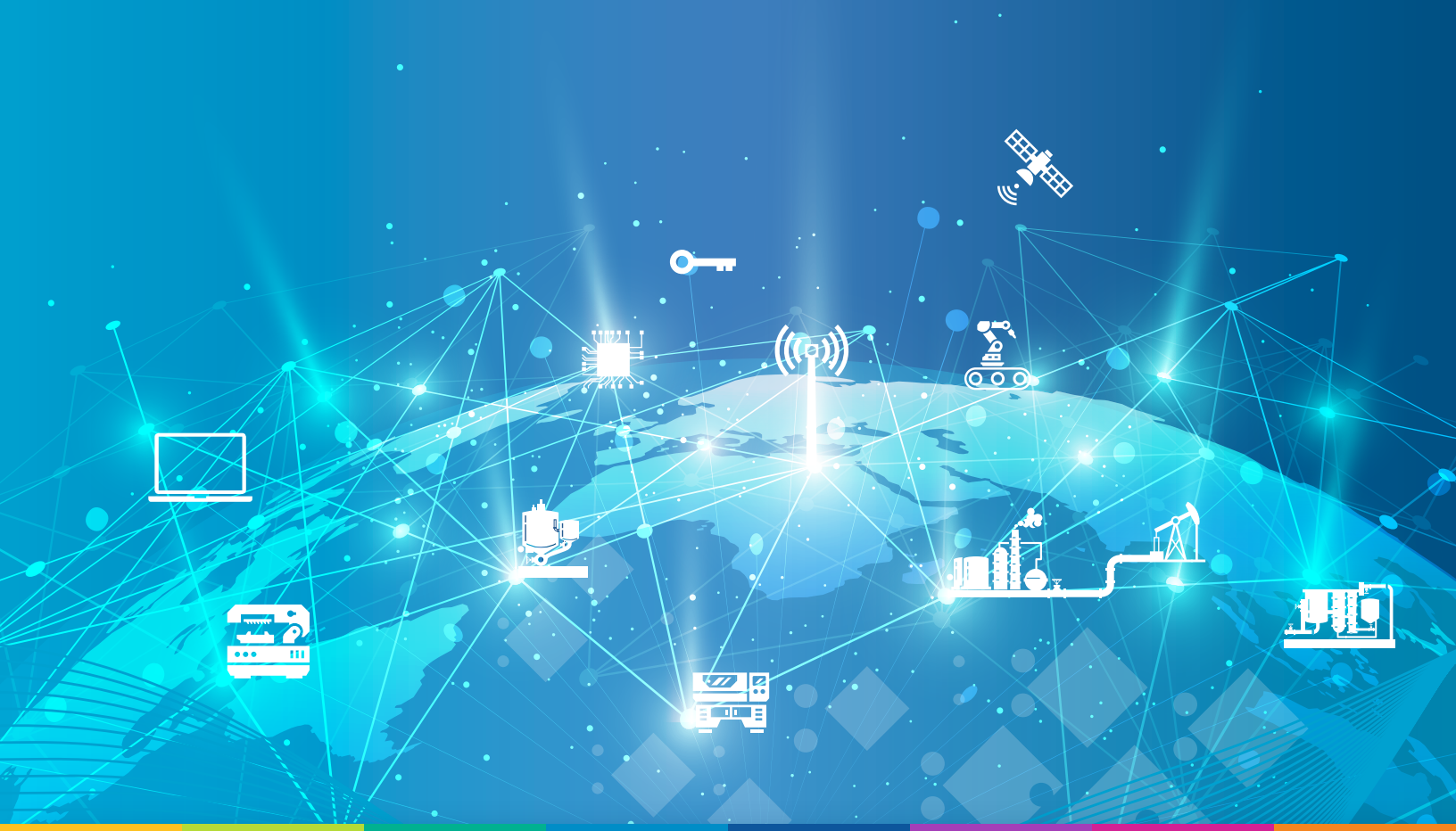
OPC UA for Field Level Communications – A Theory of Operation

Version 1 // November 2020



Technical Paper

Executive Summary

Digitalization of products and systems opens the opportunity to deliver new and enhanced software solutions and enables new digital services and business models. The implementation of concepts is made more difficult because of the heterogeneity of communication protocols at the field level. Although most of today's fieldbus systems and Real-Time Ethernet protocols are standardized by IEC in the 61784/61158 series, automation devices supporting different protocols are not interoperable with each other and often cannot coexist in a common network infrastructure. In addition, device information is structured using different information models, which makes data analysis a labor-intensive and time-consuming task that is also vulnerable to errors, especially in multi-vendor and multi-protocol environments.

However, the trend of moving to seamless interoperability accelerated by the dawn of the Industry 4.0 and Industrial Internet of Things (IIoT) era requires industrial system integration to become vendor-independent and to support end-to-end interoperability from sensor to cloud, including field level devices for all relevant industrial automation use-cases, including real-time, motion, and functional safety.

Standardized communication from sensor to cloud will support the digital transformation across all industries, including factory automation and process automation. End users, machine/skid builders and system integrators will benefit from easier system integration and cross-vendor interoperability. Seamless access to production data and process conditions will facilitate availability and optimization of production processes.

On a technical level, this approach requires standardization to take place on multiple levels: semantics, information modeling, communication protocols, data link layer and physical layer – all embraced by a common cyber-security framework. An important aspect is the convergence of information technology (IT) and operational technology (OT) allowing a common network infrastructure to be shared by IT and OT traffic while guaranteeing different levels of quality of service (QoS) demanded by diverse IT and OT applications. Technologies of particular importance are the Ethernet Advanced Physical Layer (APL) and Ethernet Time-Sensitive Networking (TSN). APL facilitates seamless Ethernet connectivity down to the field level, including long cable lengths and explosion protection via intrinsic safety with power and communication over two wires. TSN enables deterministic communication on standard Ethernet, allowing IT and OT protocols to coexist in a common network infrastructure.

The OPC Foundation's Field Level Communications initiative was established in November 2018 to specify extensions to the OPC UA framework in order to standardize the semantics and behaviors of controllers and field devices from different manufacturers. The main use cases covered by the initiative are controller-to-controller, controller-to-device, and device-to-device including support for IIoT connectivity for both controllers and devices (controller-to-compute and device-to-compute). The technical work is being performed in OPC Foundation multi-vendor working groups that define the technical concepts and specify the different mechanisms to achieve these goals.

Contents

Introduction

Background

The goal of digitalization is to foster the integration of IT technologies with OT products, systems, solutions and services across their complete value chains, which stretches from design and production to maintenance. Once implemented, digitalization of products and systems opens the opportunity to deliver new and enhanced software solutions and enables new digital services and business models.

The Internet of Things (IoT) brings together a broad range of technologies to those OT products, systems and solutions that have traditionally not been connected via today's near ubiquitous IP-based networks. While Ethernet provides the ability for things to 'reach' each other, they still need a common way to communicate. Standardized data connectivity and interoperability addresses this need.

In simple terms, with standardized data connectivity at its core, the Industrial IoT (IIoT) can be looked at from two perspectives: horizontal and vertical data connectivity. An example of horizontal communications is to be clarified: controller-to-controller (C2C) data connectivity between shop floor systems. An example of vertical communications is device-to-

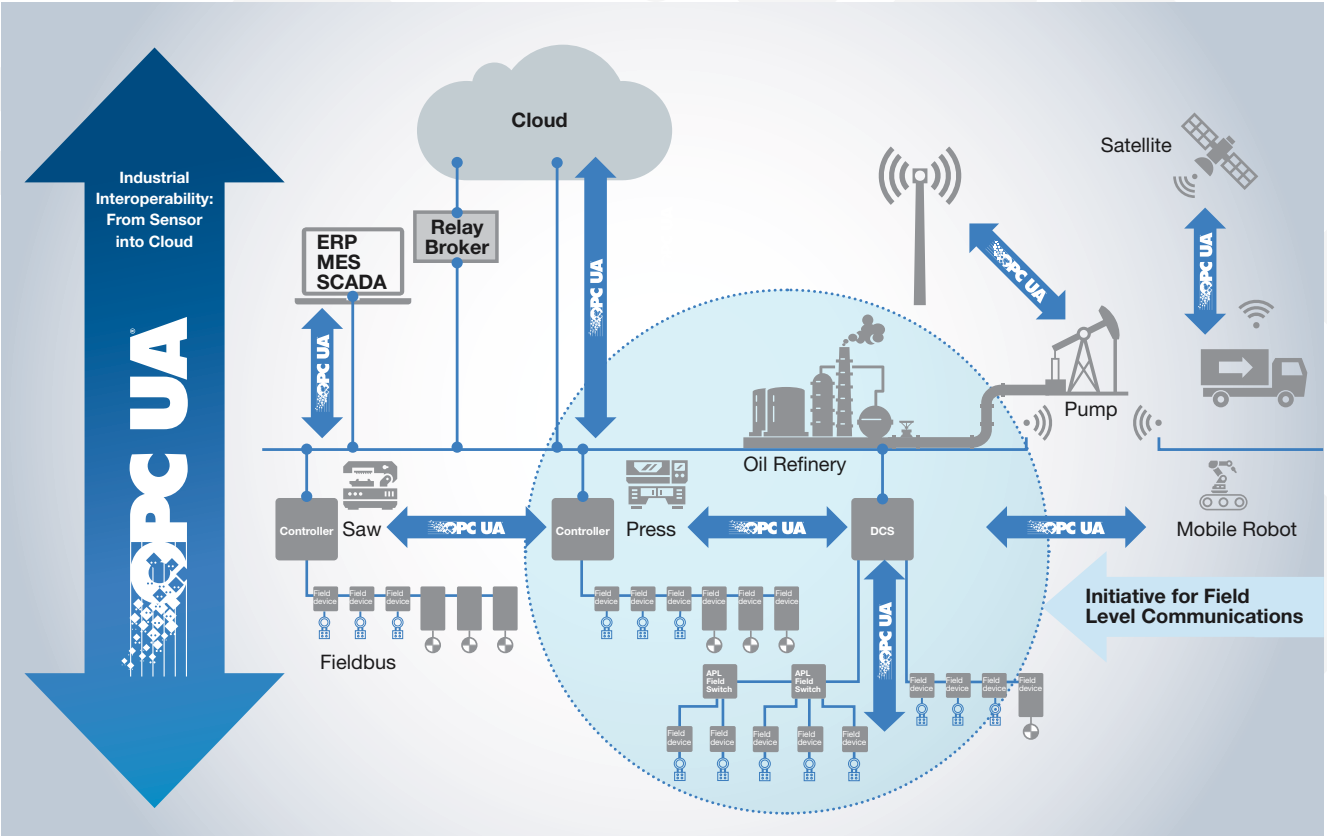


Figure 1: OPC UA use cases and scope of Field Level Communications Initiative

cloud data transfer. In both cases, the OPC Unified Architecture (OPC UA) standard from the OPC Foundation provides a secure, reliable, and robust foundation to facilitate standards-based data connectivity and interoperability. For years, many companies and partner organizations have openly worked together under the umbrella of the OPC Foundation to make this a reality and will keep doing so as it continues to expand its collaboration activities.

A key aspect of improving horizontal and vertical data connectivity is network convergence supporting a common network for IT- and OT-related communication. Ethernet Time-Sensitive Networking (TSN) according to IEEE 802.1 supports communication with bounded latency and jitter. In addition, various data streams, and traffic types can be transmitted over a common network infrastructure, while at the same time guaranteeing the various bandwidth, latency, jitter and reliability requirements of the different applications. Therefore, TSN plays a key role in supporting the convergence of IT and OT. The Ethernet Advanced Physical Layer (APL) is another key technology to drive network convergence as APL delivers seamless Ethernet connectivity to sensors and actuators in process automation – including hazardous areas.

Field Level Communications Initiative

At the SPS IPC Drives Fair 2018 in Nuremberg, Germany, the OPC Foundation officially launched the Field Level Communications Initiative. This initiative aims at extending OPC UA to field level to achieve an open, unified, standards-based IIoT communication solution between sensors, actuators, controllers and cloud addressing all requirements of factory automation and process automation (see Figure 1). Consequently, the OPC Foundation's vision of becoming the worldwide industrial interoperability standard is advanced by extending OPC UA to the field level. Vendor independent end-to-end interoperability between field level devices is provided for all relevant industrial automation use cases, including real-time, functional safety and motion, all requiring secured information exchange.

The IT and OT worlds are converging. The Field Level Communications initiative aims to achieve the following four types of convergence:

1. **OPC UA application convergence** where multiple OPC UA automation devices from multiple vendors share one network.
2. **OT convergence** where multiple systems and devices from multiple vendors using different OT protocols share one network.
3. **IT/OT convergence** where multiple controllers, devices, applications, and systems from different vendors using a combination of IT and OT protocols share one network.
4. **IT/OT organizational convergence** where the boundary between organizations blurs and management of IT and OT groups operate to common strategies and processes.

Target audience

The target audience for this technical paper are engineering managers, automation engineers, technical product managers and technical sales representatives who would like to get an overall understanding of the technical approach and the basic concepts developed by the OPC Foundation's Field Level Communications Initiative.

Document walkthrough

To guide the reader through the document, an overview about the structure and the content of each section is given:

1. The **Technical System Description** (pages 8 – 13) outlines the technical approach taken to extend the OPC UA framework for supporting additional use cases in factory automation and process automation. Details about the system architecture, the software interactions and the communication patterns are given, highlighting the key controller-to-controller (C2C) use cases and the target network architecture that are addressed in the first specification release.

2. The following section **Automation Component Model** (pages 14 – 17) outlines the approach to model Automation Components using an Asset Model and a Functional Model with Functional Entities. Details about connections, Connection Configuration Data and the Connection Manager are given, as they include the key concepts for exchanging data between multiple Automation Components.

3. In the section **Offline Engineering Workflow and Model** (pages 18 – 24) the workflow for a control systems engineer is described to enable the C2C use cases prior to on-site commissioning. Two descriptor concepts are explained: the product descriptor for product-related data of an Automation Component and the configuration descriptor that contains a product-related part and one or more configuration parts. Two examples demonstrate how the workflow looks like in a scenario with one line controller and three subordinate controllers with and without TSN, using Configuration Descriptors which will form part of Configuration Descriptor Packages.

4. In the **Safety and Security** sections (pages 25 – 28) it is explained how data for functional safety applications is exchanged between Functional Entities principle in combination with standard connections. SafetyProviders and SafetyConsumers exchange safe data using OPC UA Safety, a safety transmission protocol which facilitates the use of OPC UA in safety-critical applications. This is followed by an explanation of how connections are secured during connection establishment and data exchange against malicious attacks.

5. In the section **Transport** (pages 29 –31) the supported communication profiles and their structures with regard to the transport facets and network access facets are described. Furthermore, it is explained how interoperability is affected when mixing devices supporting different communication profiles. Afterwards the importance and the impact of the Ethernet Advanced Physical Layer (APL) and Ethernet Time-Sensitive Networking (TSN) are described in the context of extending OPC UA to the field level.

6. The last section **Summary and Outlook** (page 38) gives an overview of the key achievements and future work of the Field Level Communications initiative in order to support all relevant use cases and application scenarios in factory automation and process automation. Furthermore, it is explained what measures are taken to ensure an easy implementation of the technology as well as cross-vendor interoperability.

Technical System Description

System Architecture

OPC UA is the data exchange standard for secure, reliable, manufacturer and platform-independent industrial communication. It is based on specifications that were developed in close cooperation between manufacturers, users, research institutes and consortia, in order to enable secure and reliable information exchange in heterogeneous systems. Nevertheless, the additional mechanisms needed to satisfy the specific OT-related requirements – such as functional safety, determinism, and redundancy – for information exchange between devices and controllers in manufacturing factories and process automation plants were lacking (see Figure 2).

The technical work within the Field Level Communications initiative includes the following topics:

- → definition of a base model for Automation Components that are common to all conformant controllers and devices
- → definition of system behaviors and sequences for common functionalities e.g. bootstrapping, connection establishment, etc.
- → harmonization and standardization of application profiles such as IO, motion control, functional safety, system redundancy
- → standardization of OPC UA information models for field-level devices in online and offline scenarios

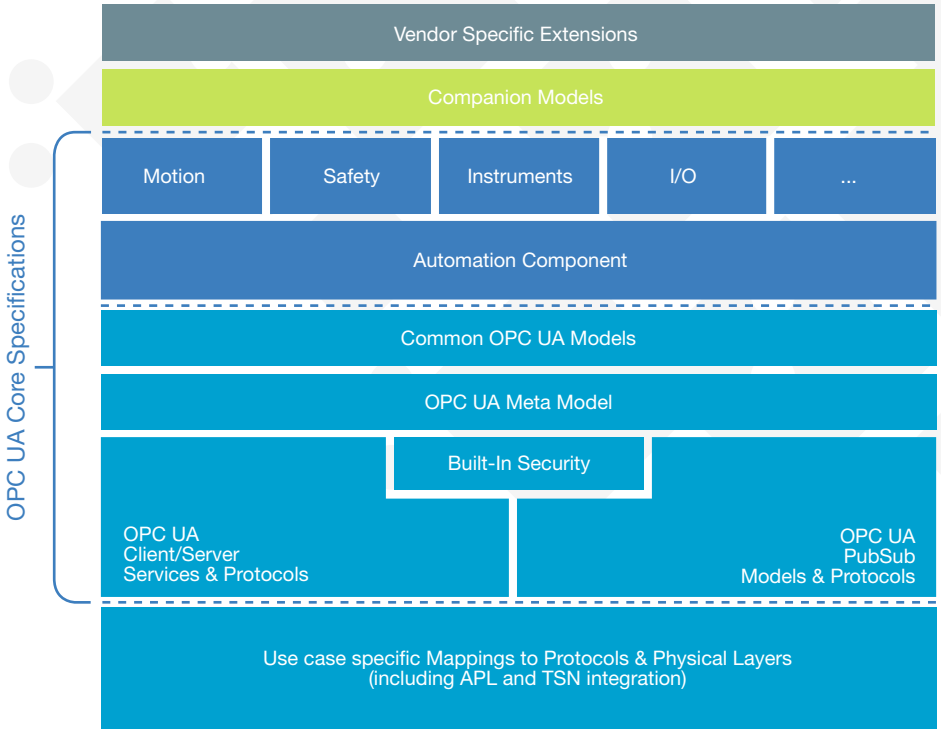


Figure 2: OPC UA for Field Level Communications – System Architecture

e.g. device description, diagnostics, etc.

- → integration of OPC UA companion models
- → support of Ethernet TSN for deterministic communication and IT/OT convergence
- → mapping of application profiles related to real-time operations on Ethernet networks including TSN
- → definition of facets, profiles and conformance units that can be tested to guarantee interoperability across vendors
- → definition of certification procedures

In the **first specification release (Version 1)**, the focus is on the Controller-to-Controller (C2C) use case which includes exchanging both standard and safety real-time data using OPC UA Client/Server and OPC UA PubSub in combination with a peer-to-peer application relationship and basic diagnostics. The target network architecture is shown in Figure 3.

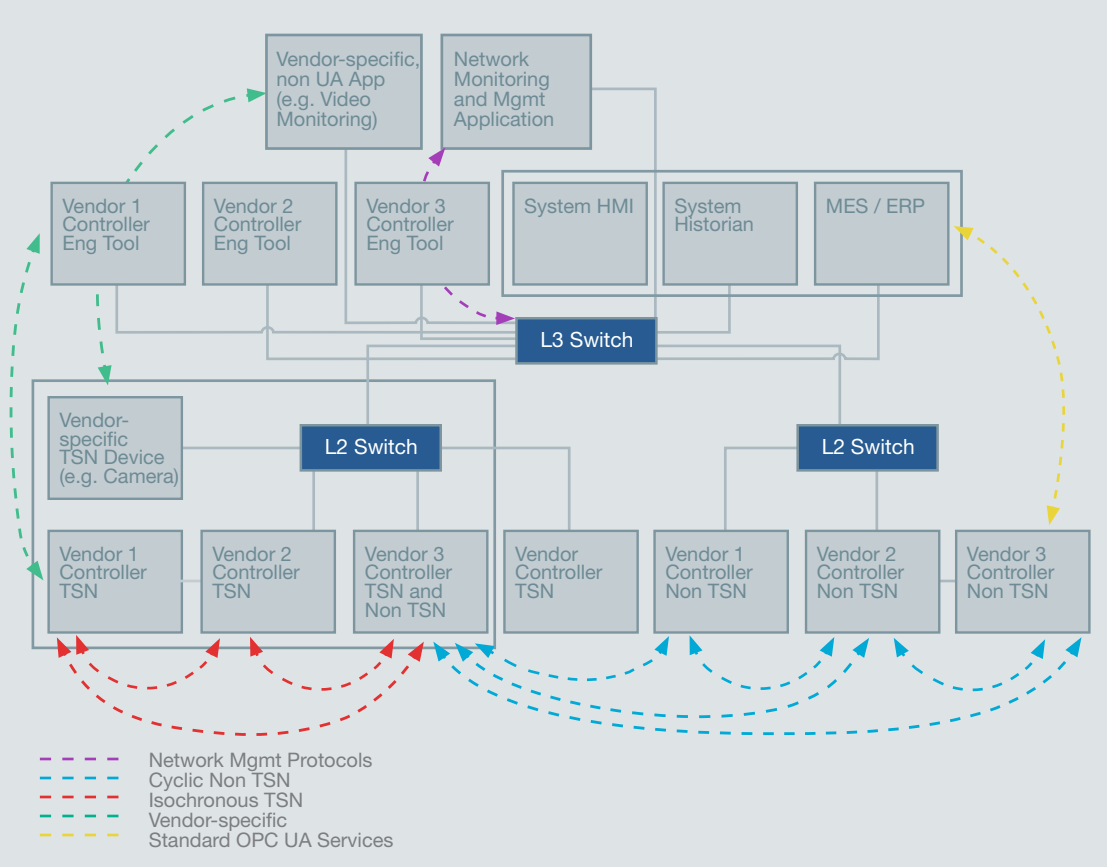


Figure 3: Controller-to-Controller supported network architecture

Interaction Model

In the interaction model shown in Figure 4, a **controller** represents a function typically implemented in a Programmable Automation Controller (PAC), Programmable Logic Controller (PLC) or Distributed Control System (DCS) controller. Today, automation **devices** are typically connected to controllers and can be as simple as an inductive proximity switch or as complex as a Coriolis Flow Meter or Servo Drive. **Compute** refers to standalone software applications running on a variety of hardware platforms, from an edge gateway to a blade server in the cloud. Controllers and Devices have many attributes in common – the term "Automation Component" is used where functions apply to both.

→ Controller-to-Compute

Software running on Compute platforms is a major area of innovation today, whether it is management information in dashboards, long term process optimization, predictive device-level diagnostics or digital twins. These all require information to be extracted from controllers. OPC UA is dominant today, and almost every major controller supplier offers OPC UA directly on its controller or device.

→ Controller-to-Controller

Plant owners and system integrators are assembling complex operations using machinery purchased from different machine/skid builders. They may find that each is fitted with a controller from a different vendor, resulting in a need for an easy way to set up controller-to-controller communications across multiple vendors. This problem has not been solved in industrial automation to date and the controller-to-controller solution created by the Field Level Communication Initiative will be the first to deliver an interoperable real-time solution covering both standard and safety communications for all types of automation applications.

→ Controller-to-Device

The traditional fieldbus approach of having a controller communicate with a subnet of I/O modules, drives, servos, instruments, and other smart Automation Components is well understood in the industrial automation community. Albeit it comes with constraints on network architecture and topology when a converged IT/OT solution is deployed, or dif-

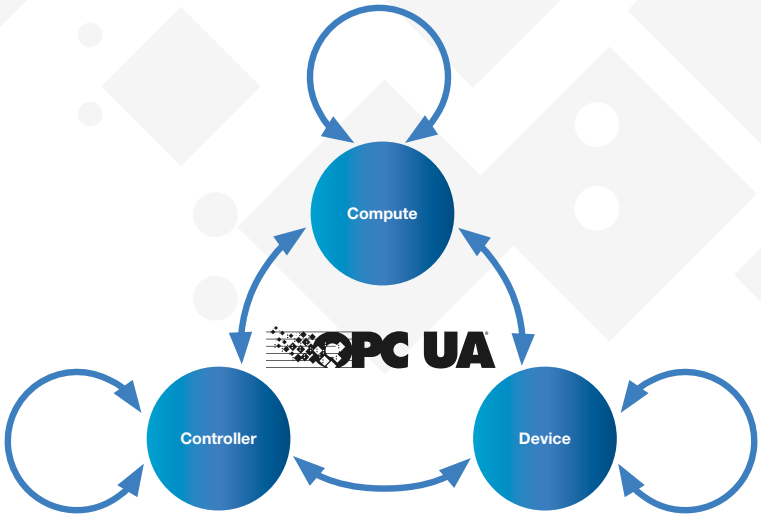


Figure 4: Interaction Model for OPC UA including field level communications

ferent Industrial Automation technologies share the network. The Field Level Communications initiative will deliver controller-to-device communications that meet or exceed the capabilities provided by the IEC 61784 profiles.

→ Device-to-Device

In bringing together best practices for multiple shop-floor technologies and by harmonizing the application profiles used in end devices, applications such as load sharing of inflexible loads across multiple servo drives will become far easier to deploy in an interoperable manner.

→ Device-to-Compute

Controllers often serve as a proxy for devices, add valuable context to the information provided by these devices, and in some cases control access to that information. However, as devices become increasingly complex with an ever-growing number of useful variables and internal and external measurements, the use of a controller as a proxy becomes increasingly impractical. For example, routing thousands of variables from each device through a controller is no longer scalable. OPC UA for Field Level Communications will define the necessary semantics and metadata to contextualize the information from devices for use in diverse compute-based software applications in an open architecture without the controller acting as a bottleneck.

→ Compute-to-Compute

These applications include gateways to IT systems, cloud-to-cloud connectivity, interoperable manufacturing operations management, and many more. The Field Level Communications initiative will use and build on the services, information modeling, and interoperability that have driven the success of OPC UA in compute-to-compute applications over the last decade. While no need for further development of capabilities to support compute-to-compute applications is expected within the Field Level Communications initiative, these applications will inherit and benefit from the increased harmonization delivered at the field level.

Communications Patterns

An example of controller-to-controller communications is where a blending skid of one vendor is integrated into a homogenizer of another vendor, each selecting controllers from different vendors with their own eco-system of devices (see Figure 5). Similar examples exist with machines and distributed automation systems.

Figure 5: Controller-to-controller example

OPC UA for field level communications supports two new enhanced approaches that make use of OPC UA PubSub without preventing the currently available Client/Server mechanisms to exchange data between these machines:

→ Unidirectional

The name unidirectional derives from the flow of application data. Each machine's designer creates a configuration descriptor of output information available and supported configurations (update rates, security, etc.) in their controller for the other controller's use. Other machine designers can then import the descriptor to enable communications and to customize their code to correctly use the data made available by other controllers as inputs to their own machines.

→ Bidirectional

This model extends the unidirectional model and inherits all attributes of that model.

In this model, designer 1 fixes the data and format that their controller both transmits (outputs) and receives (inputs). It is the responsibility of the other controller (and its designer) to initiate communications to designer 1's controller and to provide/consume information in the format demanded by designer 1.

In the unidirectional model, responsibility is symmetrical with both the designer of Machine 1 and Machine 2 performing exactly the same functions in order to establish communications in both directions between the two controllers. In the bidirectional model, one party defines both inputs and outputs from their equipment and the other party establishes a bidirectional connection – the two machine controllers perform different functions in the communication relationship:

- → Machine 1 designer defines the data to be exchanged in both directions, but the controller does not initiate any communication.
- → Machine 2 controller initiates all communications, but its designer must ensure that it is both transmitting and receiving information in a form usable by Machine 1's application code.

In this case, Machine 1's behavior is very similar to that of an I/O module.

Communication Configuration

Standardized configuration descriptors are used to exchange communication configurations between the engineering tools of the controllers. An engineering tool and a controller together can automate the creation of all necessary information model entries, automate the establishment of a connection to the other controller and automate fault handling. Some flexibility may be needed in post-installation communications configuration, especially in cases where multiple identical machines or skids are delivered to a single application (see Figure 6). The level of allowable configuration must be controlled by the machine designer and the actual configuration may be set using by any generic OPC UA Client. In this example, two identical machines have been purchased from Vendor 1 and two identical machines from Vendor 2. None of the machines change function or operation post-installation, but there is no planning in advance of which machine is connected to which, and potentially, no pre-planning of the network identification of each machine once installed. At the time of commissioning, each machine (or more specifically, the controller in each machine) must be given its network identity (e.g. hostname or IP address) and must have the target network identity of the controller in the other vendor's machine using a general- purpose OPC UA client.

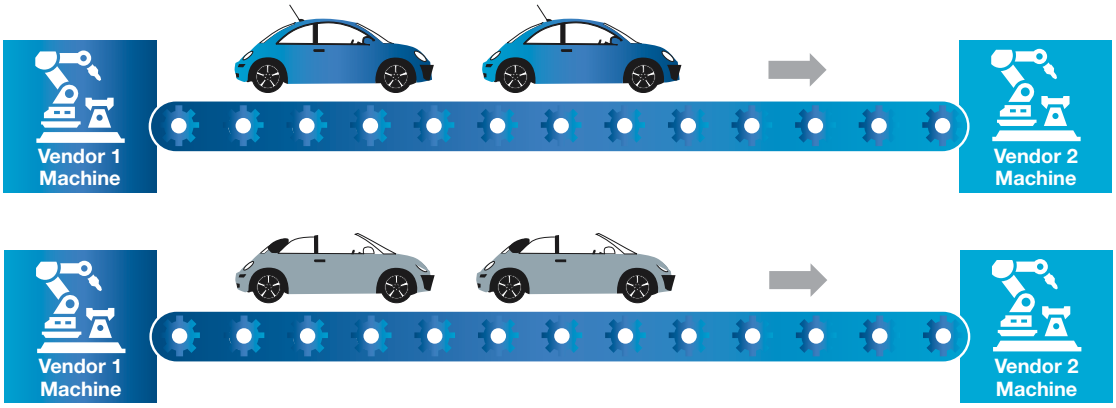


Figure 6: C2C Example with two identical lines (Case 1)

In the following example, two identical blending skids are integrated into a homogenizer of Vendor 1 (see Figure 7). As in the previous example, the network identity of each Vendor 2 skid controller must be applied to the relevant connection in Vendor 1 controller. However, further information must be given to both of the Vendor 2 skids, as Vendor 1 controller has two unique connections for 'a' Vendor 2 skid which function in the same way but carry different data. The two Vendor 2 skid controllers must be told by both the network identity of the Vendor 1 controller, and the internal identity of the connection to which they must be associated.

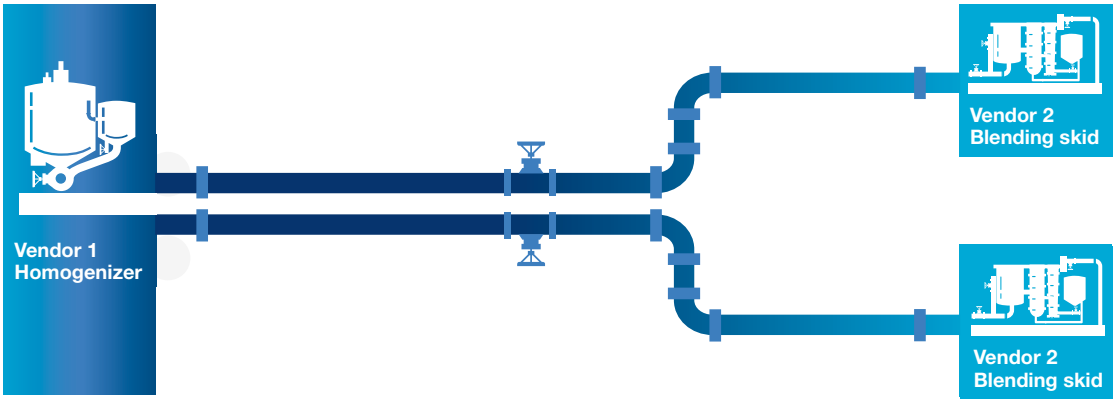


Figure 7: C2C Example with two identical skids in the same line (Case 2)

Automation Component Model

Automation Component – Functional Model and Asset Model

OPC UA systems for field-level must expose their information using a prescribed OPC UA Information Model. The model is based on an Automation Component (AC), which is an entity that performs one or more functions that are part of automation devices (e.g. controllers, drives, instruments, I/O devices) (see Figure 8). All ACs are modeled as one or more Assets, and/or one or more Functional Entities. Additionally, an AC contains information describing the Network Interfaces and Network and Communications Services the AC supports.

The scale of an AC is up to the vendor. It could be as small as an individual standalone I/O device or as large as a complex room-sized machine.

The AC is composed of two major groupings, the asset model and the functional model. Asset information typically describes physical items, but it can also include items that are not physical, such as firmware or licenses. The Asset Model is based on the DI-Information Model (OPC 10000-100 – Part 100: Device Information Model) but extends the use cases in scope of the Field Level Communications Initiative. The Functional Model describes a logical functionality. The functional model consists of one or more Functional Entities which encapsulate features, which can include input/output variables, communication, and device parameters, as well as communication connections. A Functional Entity (FE) is abstracted from the hardware, which allows porting of applications to new hardware. Functional Entities

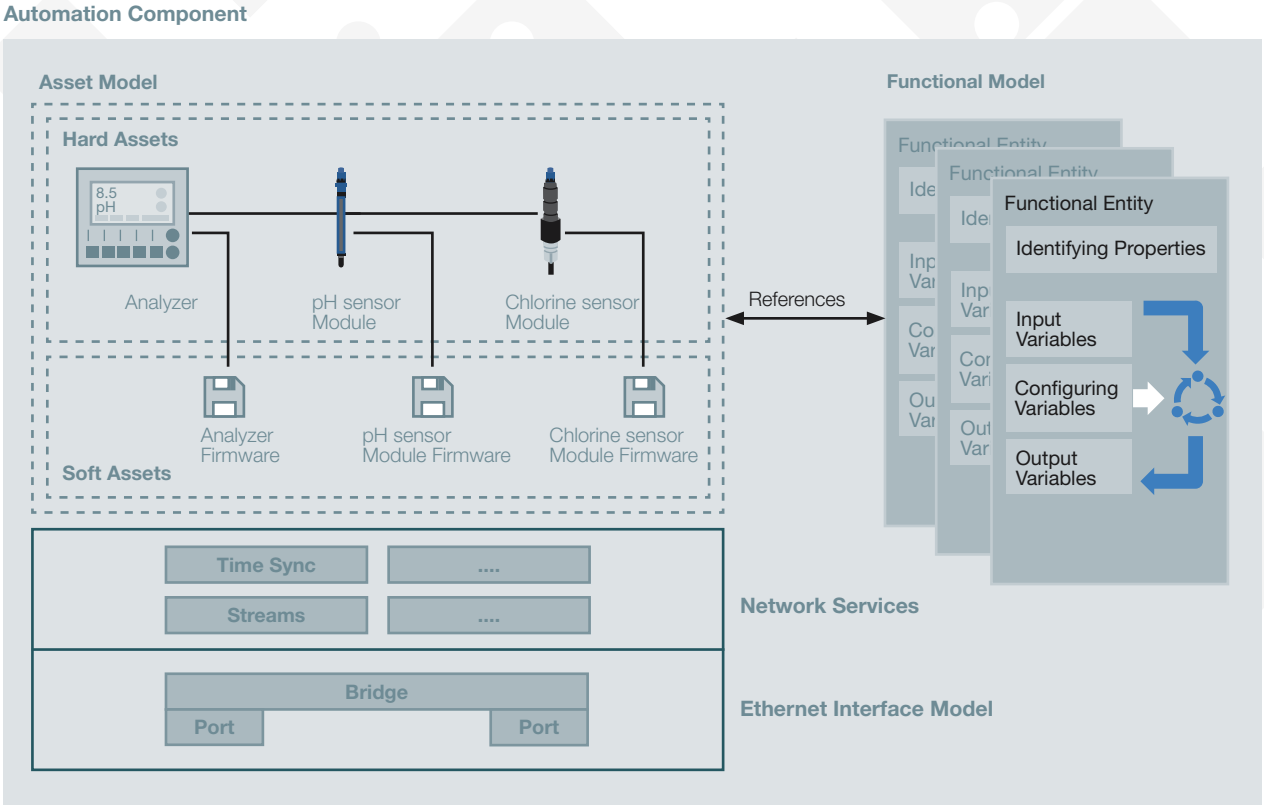


Figure 8: Automation Component Model

reference assets that they are associated with or execute on, and allow applications to confirm that any hardware requirements that the applications have are met. For example, a two-axes drive may be based on one asset including two Functional Entities.

From Functional Entity to the Communication Relationships

A Functional Entity is an element of an AC that represents a functional capability of the AC (see Figure 9). Examples of Functional Entities include application execution engine, motion axis control, a sensor, a relay, I/O control, and variable frequency drive control. There can be multiple FEs in an AC.

Connections are the logical constructs used to exchange a defined set of process data and process data quality information. Inside a connection one or more PubSub DataSetWriter and DataSetReader are responsible for exchanging the data with the other Functional Entity.

For exchanging process data using PubSub, the following connection types are supported:

1. Unidirectional
2. Unidirectional with heartbeat
3. Bidirectional

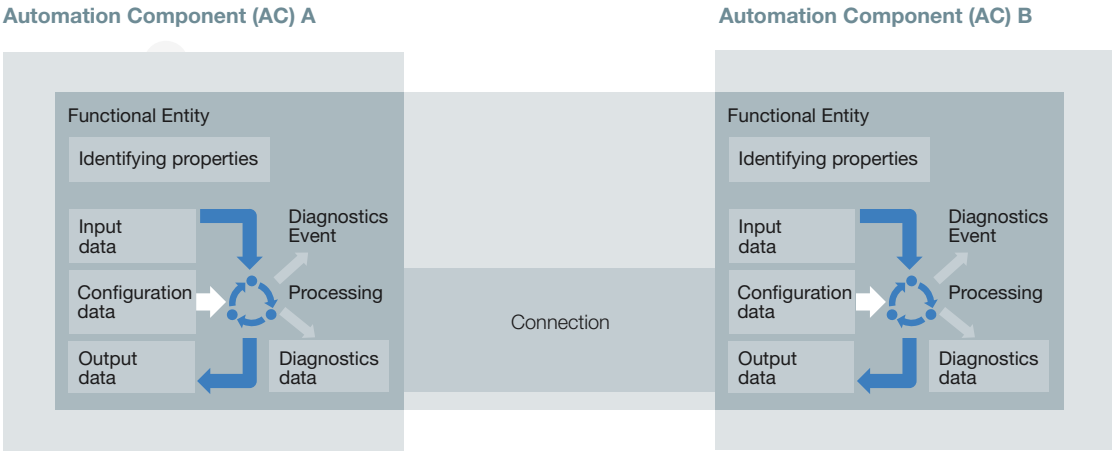


Figure 9: Connections between Functional Entities

The Role of the Connection Manager

The Connection Manager (CM) is a service responsible for establishing connections between FEs. It uses Connection Configuration Data such as partner communication address, update rate and QoS settings, to set up communication with one or more communication partners (see Figure 10).

The CM is modeled as a distinct entity. This entity resides typically in an AC which initiates the connection establishment via an internal mechanism, but may optionally be realized as an external entity.

The Connection Configuration Data consists of the following parameters:

- Description of the endpoints to be connected
  - Local Address, consisting of the address of the OPC UA server located on the AC, and the browse path to the Functional Entity to connect
  - Remote Address
- Choice of unicast or multicast
- QoS option and their pparameters (including TSN)
- For process data
  - PublishingInterval (for the data publisher)
  - MessageReceiveTimeout (for the data subscriber)
- For heartbeat
  - PublishingInterval (for the heartbeat publisher)
  - MessageReceiveTimeout (for the heartbeat subscriber)
- Connection timeout (for the cleanup)
- Compatibility verification parameters

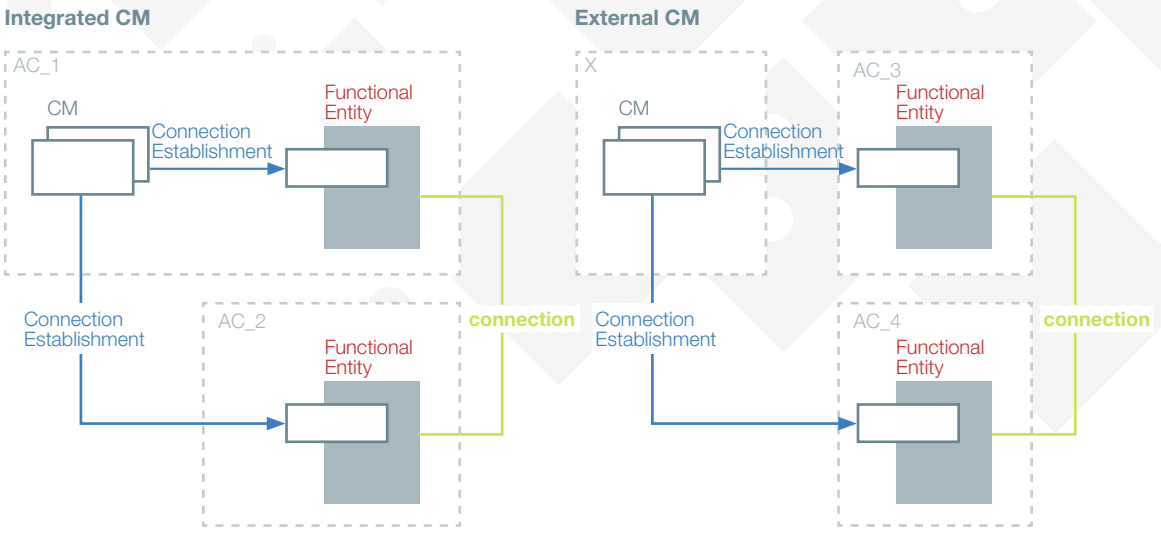


Figure 10: Integrated or External Connection Manager of Automation Components

Connection state machine

The Connection Manager (CM) must establish a connection on two endpoints in parallel. For each of the endpoints a separate connection state machine is needed (see Figure 11).

For joint operation of the two endpoints, it is proposed that the CM executes the state machine for the two endpoints in parallel, meaning step by step on the one and the other endpoint. This will later ease the startup of the communication.

This Connection State Machine extends the OPC 10000-14, chapter 6.2.1 defined PubSub state machine, which only addresses the single state of a PubSub Connection, but not how to reach this status.

A connection is initially set-up using Client/Server mechanisms, where information to establish a bidirectional PubSub Connection, such as compatibility verification, ownership and parameterization, is exchanged. Thereafter the PubSub Connection is prepared and operational.

The Connection State Machine for each connection is defined in the information model of a Functional Entity.

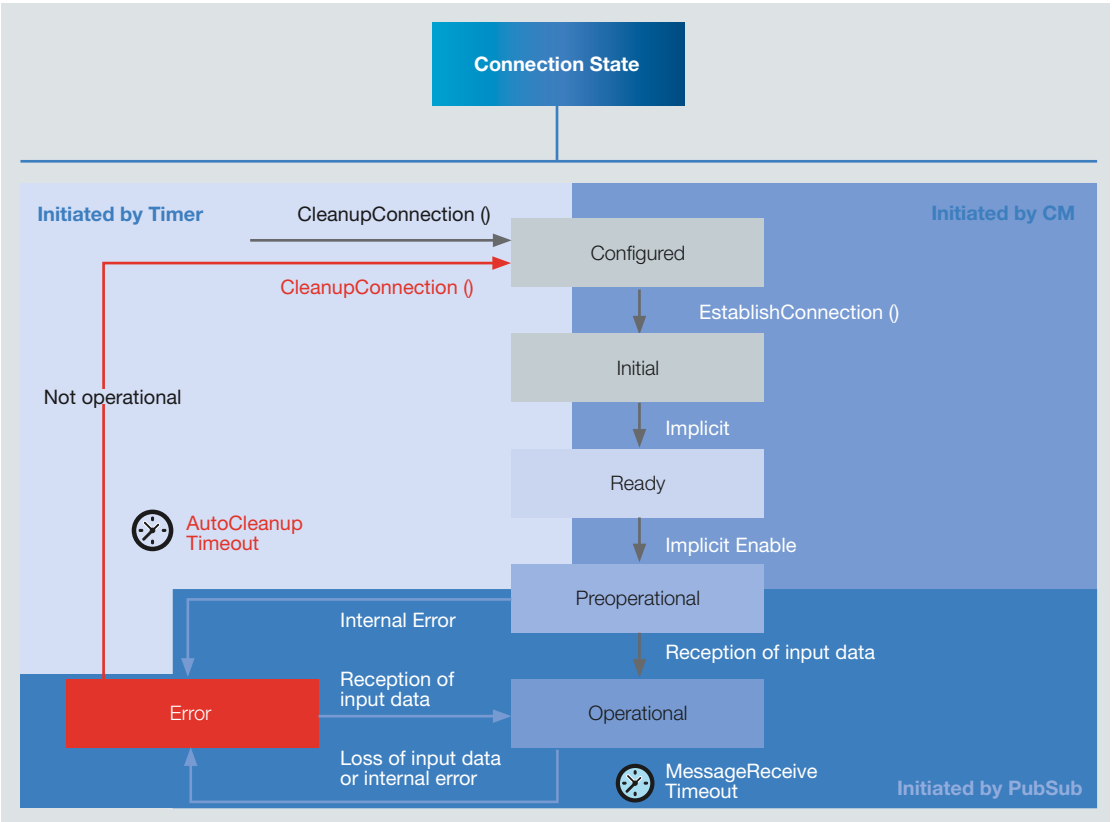


Figure 11: Connection State Machine

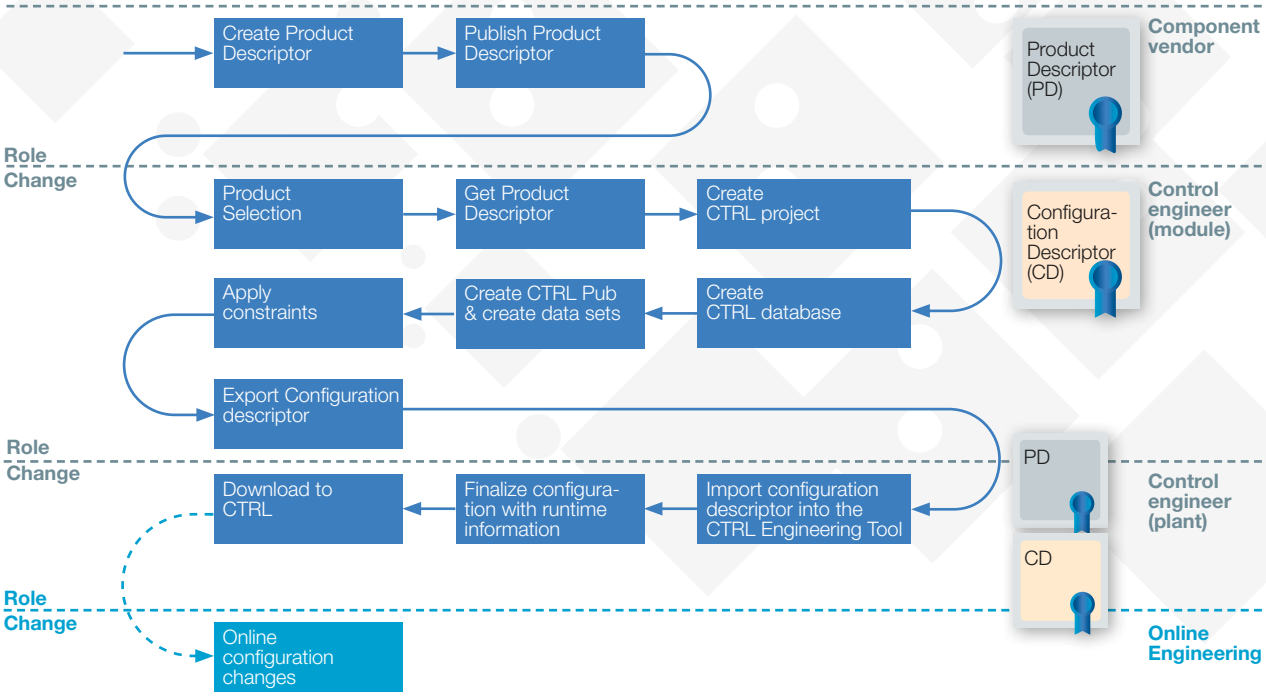
Offline Engineering Workflow and Model

Introduction

Offline Engineering is an important element for the development, operation, and maintenance of an automation system. By allowing the user to be able to understand the operation of the automation system before deploying the system in physical hardware, the user will know that the system will perform the control function reliably and correctly once the physical system is in place. The user will be able to simulate changes and updates to the automation system before making changes to the physical system and be assured the changes will perform up to the expectation of the user and improve the performance of the system.

This chapter describes the configuration workflow that consumes the AC description artifacts in the Offline Engineering phase.

The following diagram is an overview of the workflow steps for Offline Engineering descriptor(s) usage.



CTRL: Controller e.g. PLC, DCS

Figure 12: Overview of the workflow steps for offline engineering descriptor(s) usage

Descriptor Definition

Generally, the Descriptor of an AC is a set of documents containing an OPC UA information model and potentially other useful information for configuration purposes. The information can be for one AC or a group of ACs (like a machine, machine module or skid). The AC Descriptor is delivered in a packaged container format (zip file package) supporting the provisioning and sharing of information in offline engineering. One or more digital signatures in the Descriptor provide integrity for the content.

The documents of an AC Descriptor can be categorized as information model and attachment documents. Information model documents define the Information Model of the AC, while the attachment documents provide supplemental (and optional vendor specific) material for the engineering and deployment process.

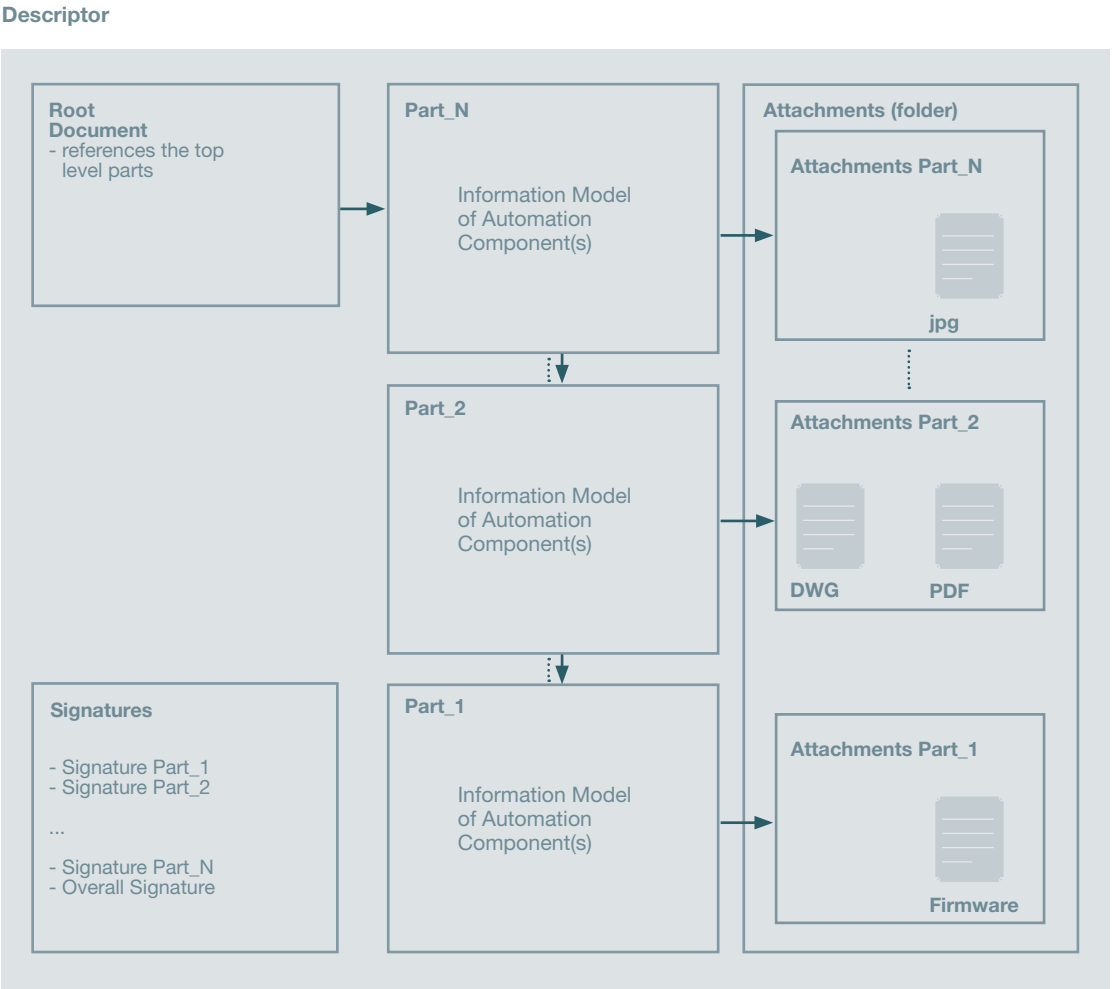


Figure 13: Generic Descriptor Package or Generic Descriptor

As shown in Figure 13, the content is separated into parts. Each part may be produced by a different engineer and is the result of a specific engineering step. The part structure forms a hierarchy where an upper level (later) part depends on the lower level (earlier) parts and adds to or overwrites (if allowed) information that was created in the previous part.
The content of an information model Part consists of one or more AutomationML (AML) documents. AML is a vendor-neutral XML-based format for the storage and change of engineering information.
Two examples of AC Descriptors are described next.

Product Descriptor
A Product Descriptor is a specific AC Descriptor containing product data of the AC (see Figure 14). Usually, the Product Descriptor is provided by the AC vendor. Importing the Product Descriptor into an engineering tool can be the first step in engineering of an AC. In most cases, the Product Descriptor is included in a later descriptor (e.g. Configuration Descriptor, see Figure 15) as the first part.
The Product Descriptor states the identification, structure, and capabilities of the AC assets. For a field or I/O Device, the Descriptor may also contain information about the Asset Component functionality.

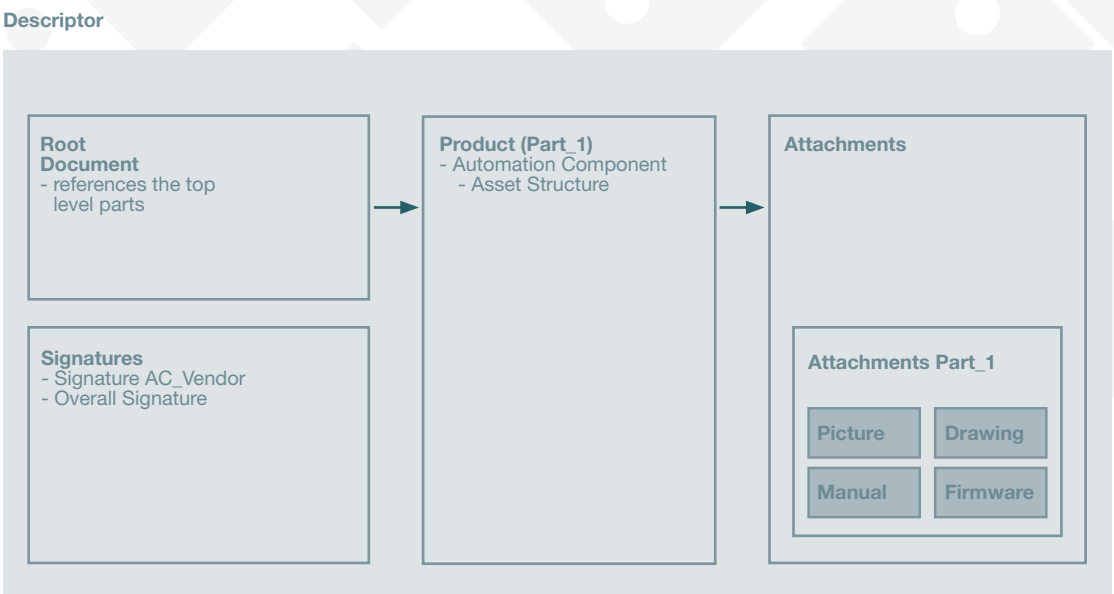


Figure 14: The Product Descriptor

Configuration Descriptor

The Configuration Descriptor shown in Figure 15 is a descriptor with one product part (Part_1) and one or more configuration parts (Part_2). The Configuration Descriptor is created in the engineering process, usually with the intention to share engineering information of an AC with another engineering tool.

The configuration part of the Configuration Descriptor defines the Functional Entities, the communication DataSets, the required Quality of Service (QoS) and the data necessary for connection establishment (like unicast or multicast addresses for OPC UA PubSub). In addition, for a field or I/O device, the configuration part may also contain parametrization data.

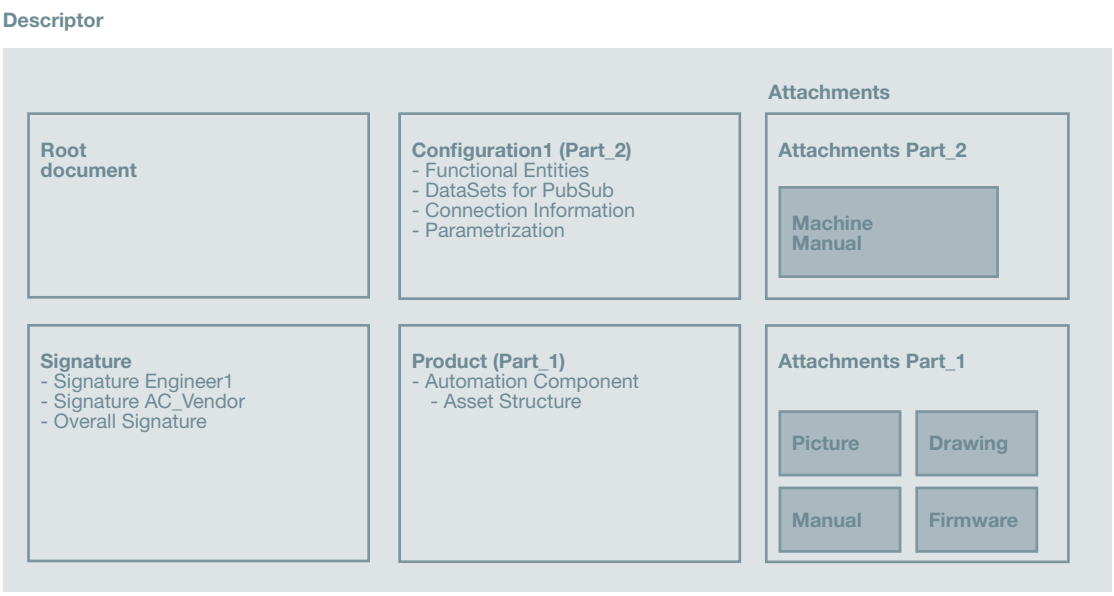


Figure 15: The Configuration Descriptor

Workflow examples

This section will show how the descriptors are used in an offline engineering environment and how the descriptors will represent one or more ACs. In the two examples below, a Line Controller (LC) will set up and provide the overall control to 3 subordinate controllers (PLC/DCS) in an automation system. In the first example, standard Ethernet communications without TSN is used. The system in the second example supports TSN communications.

Below is the workflow for the use case including enumeration for the workflow states (noted in square brackets, e.g. [1]):

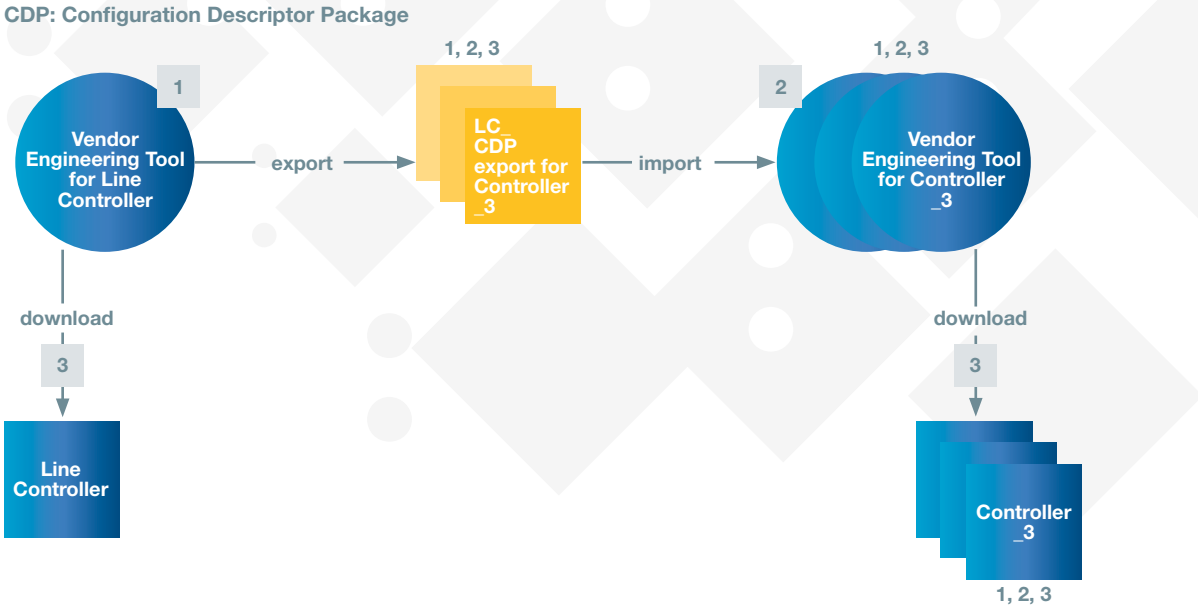
System with a Line Controller and 3 subordinate controllers without TSN

In the offline engineering phase, the engineering tool for the LC has been used to create [1] Configuration Descriptors or Configuration Descriptor Packages (CDP):

→ LC CDP for C_1 to be imported into the engineering tool of C_1
→ LC CDP for C_2 to be imported into the engineering tool of C_2
→ LC CDP for C_3 to be imported into the engineering tool of C_3

Each CDP contains the information necessary to configure the communication relationships between LC and C_X (X=1, 2, 3) (see Figure 16). In addition to the configuration information the CDP contains an index that helps to navigate through the stored information and a digital signature from the author (in this case the development engineer of the system integrator).

When the CDP is imported [2] into the engineering tool of one of the controllers, the control engineer checks the validity of the signature and uses the CDP index to browse and find the publication infor-



C: Controller (e.g. PLC, DCS) LC: Line Controller

Figure 16: Example: A System with a Line Controller and 3 subordinate controllers without TSN

mation. This enables the control engineer to set up the corresponding Subscription and Connection objects in the controller. Once the control engineer has completed the C_X project and the hardware (PLC/DCS) is connected, the configuration can be deployed [3] from the C_X engineering tool.

System with a Line Controller and 3 subordinate controllers with TSN

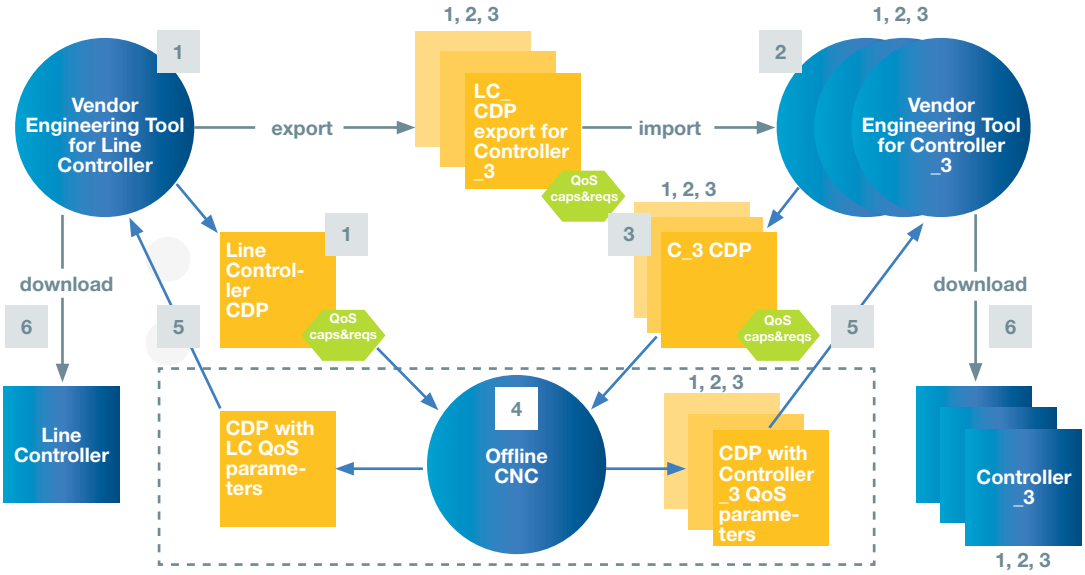
In the offline engineering phase, the engineering tool for the LC has been used to create [1] Configuration Descriptors or Configuration Descriptor Packages (CDP):

- → LC CDP for C_1 to be imported into the engineering tool of C_1
- → LC CDP for C_2 to be imported into the engineering tool of C_2
- → LC CDP for C_3 to be imported into the engineering tool of C_3

Each CDP contains the information necessary to configure the communication relationships between LC and C_X (X=1, 2, 3) (see Figure 17). In addition to the configuration information, the CDP contains an index that helps to navigate through the stored information and a digital signature from the author (in this case the development engineer of the System Integrator).

The CDP for each controller – in addition to the information in the first example – includes the QoS capabilities and requirements provided by the TSN mechanisms for each controller (C_1, C_2 and C_3). The QoS capability is part of the Product Descriptor of the controller (contained also in the CDP), while the QoS requirements are part of the Configuration Descriptor.

CDP: Configuration Descriptor Package

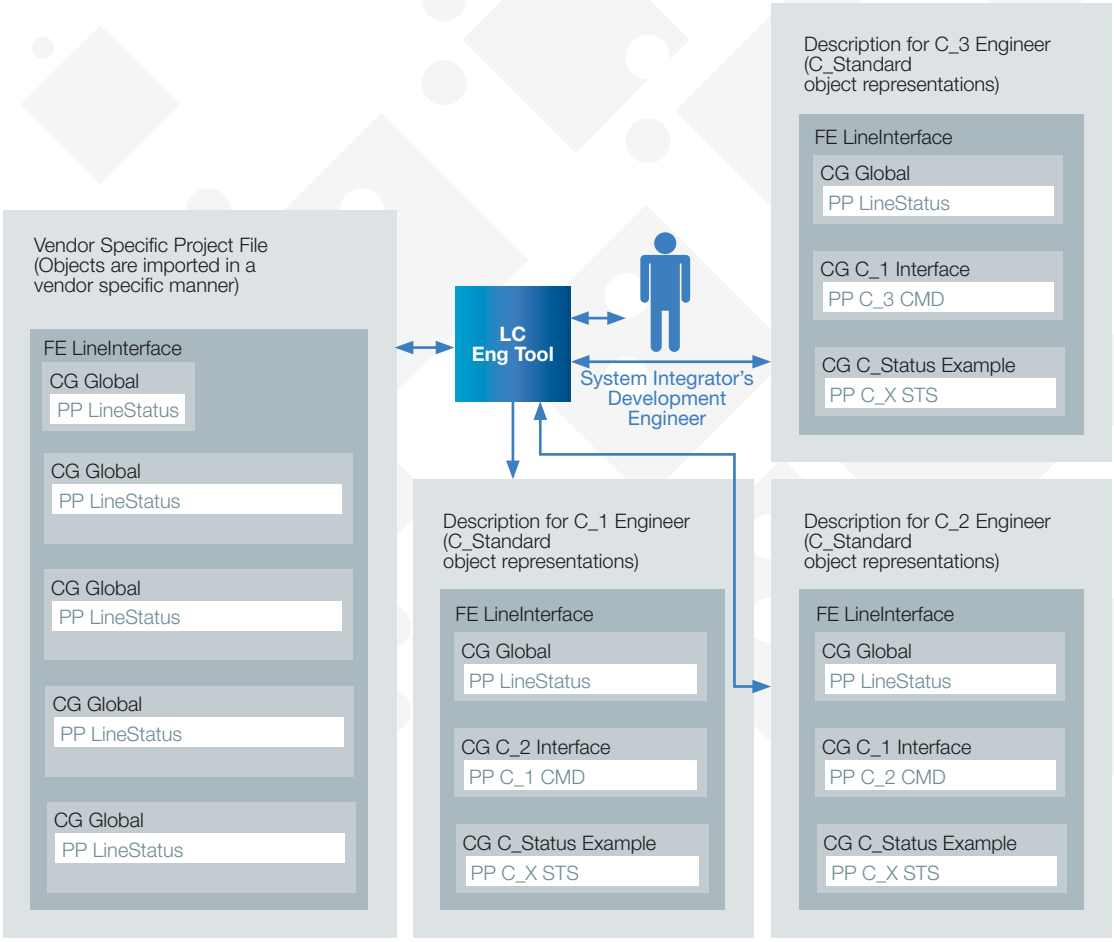


C: Controller (e.g. PLC, DCS), caps&reqs: Capabilities & Requirements, LC: Line Controller

Figure 17: Example: A system with a line controller configuring 3 subordinate controllers

When the CDP is imported [2] into the engineering tool of one of the controllers, the control engineer checks the validity of the signature and uses the CDP index to browse and find the publication information. This enables the control engineer to set up the corresponding Subscription and Connection objects in the controller.
The QoS capabilities and requirements of all controllers (LC, C_1, C_2 and C_3) are imported from the CDPs into the offline Central Network Configuration (CNC) [3] to calculate [4] the information needed for the TSN configuration (e.g. QoS parameters/TSN Stream Settings data).

The output of the calculation is entered into QoS parameters/TSN Stream Settings descriptors – one for each controller C_X. These descriptors are imported again [5] in the C_X and LC engineering tools.
Once the control engineers have completed the C_X project and the hardware (PLC/DCS) is connected, the configuration can be deployed [6] from the C_X engineering tool.



C: Controller (e.g. PLC, DCS)

Figure 18: Development Engineer of System Integrator setting up the Configuration Information

Safety Communication

The specification OPC UA Safety (OPC 10000-15 - Part 15: Safety) describes the services and protocols for the exchange of safety-relevant data using OPC UA mechanisms. It extends OPC UA to fulfill the requirements of functional safety as defined in the IEC 61508 and IEC 61784-3 series of standards.
Implementing this part allows for detecting all types of communication errors encountered in the lower network layers. In case an error is detected, this information is shared with the safety layer which can then act appropriately, e.g. by switching to a safe state. OPC UA Safety is application-independent and does not pose requirements on the structure and length of the application data.

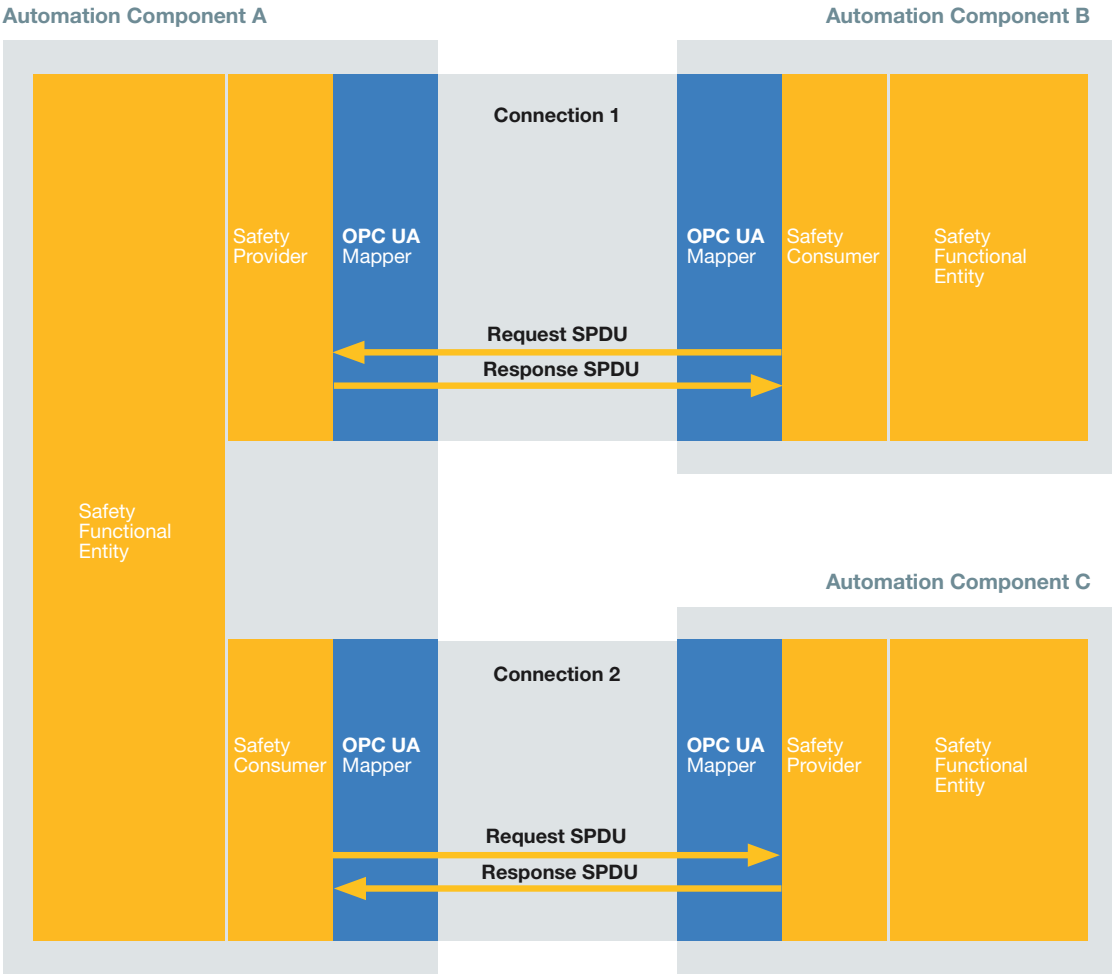


Figure 19: Safety connections between Automation Components

Safety for field-level communications

OPC UA Safety is using standard connections and an additional safety transmission protocol on top of these connections. It is developed to extend the standard data exchange between Functional Entities via connections with safety data (see Figure 19).

This principle delimits the assessment effort to the safe transmission functions, such that the underlying connections do not need any additional functional safety assessment.

Safety Functional Entities may include standard and safe input and output variables. The safety application inside the functional entity has to be developed in a safe workflow as well.

The Safety Application is connected directly with the SafetyProvider / SafetyConsumer, which exchange data by means of the safety protocol. The OPC UA Mapper is used to interface the safety layer and the underlying communication and supports the channel between SafetyProvider and SafetyConsumer.

The most basic type of safety communication is bi-directional communication, where a safety application on one AC (A) sends data to a safety application on another AC (B). The SafetyConsumer initiates the communication with the Request SPDU. The SafetyProvider mirrors the received ID and counters, adds the requested safety data and secures all data via a checksum before responding with the Response SPDU.

One AC can be SafetyConsumer and SafetyProvider at the same time. The connection between SafetyProvider and SafetyConsumer can be established and terminated during runtime, allowing different consumers to connect to the same SafetyProvider at different times.

SafetyProvider State Diagram

The SafetyProvider has a very simple state machine to implement. It is simply waiting for a request and if the request is received, the safety telegram is sent out. All safety checks will be done on the SafetyConsumer side.

SafetyConsumer State Diagram

The SafetyConsumer is initiating the safe data exchange, waits for the response, and checks for potential communication errors (integrity, timeliness, authenticity, according to IEC61784-3). Thereafter the SafeData is provided to the Safety Application inside the AC. If a communication error occurs, fail-safe substitute values will be provided to the Safety Application instead, and an error will be indicated.

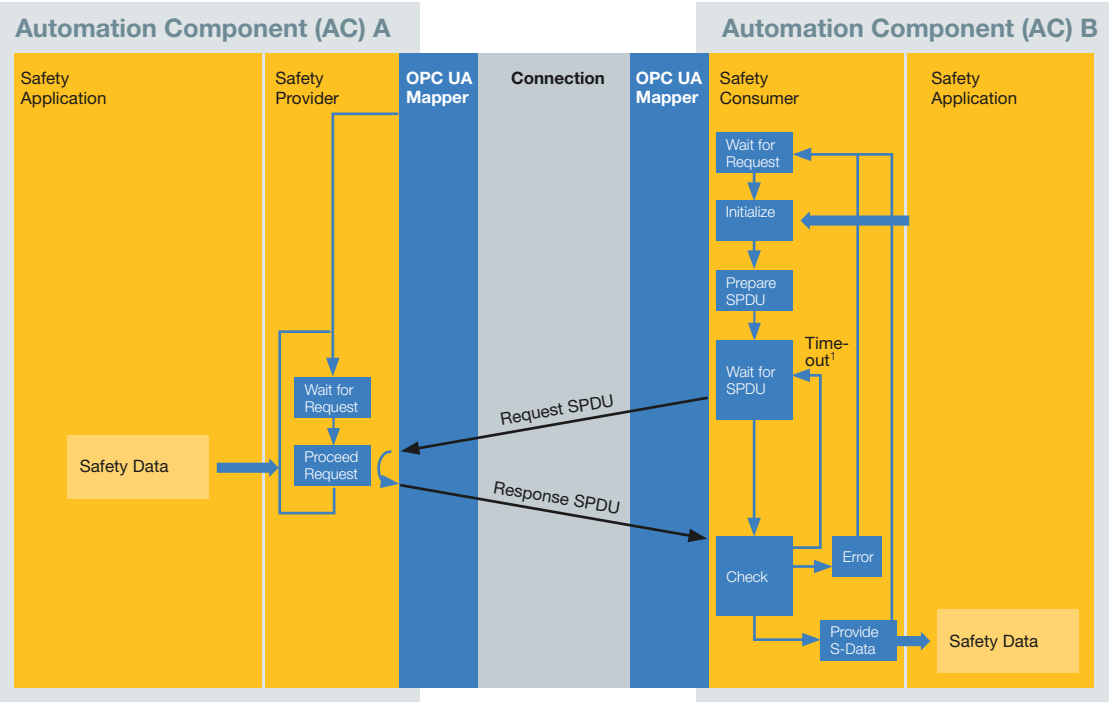


Figure 20: SafetyProvider and Consumer State Machines

[1] To avoid running into safety timeout, SPDUs may also be protected by end-to-end latency guarantee.

Security

Security for the field-level connections
Every field-level connection is authenticated and optionally encrypted by standard OPC UA security mechanisms specified for the Client/Server and PubSub communication. The connection establishment is entered after a OPC UA Secure Session establishment is completed with the use of asymmetric cryptography with certificates and private keys (see Figure 21). In this phase the mutual authentication and the symmetric key exchange for the connection establishment is done. Thereafter the Connection Manager (CM) maintains the connection via this secure session up to the operational state of the connection.

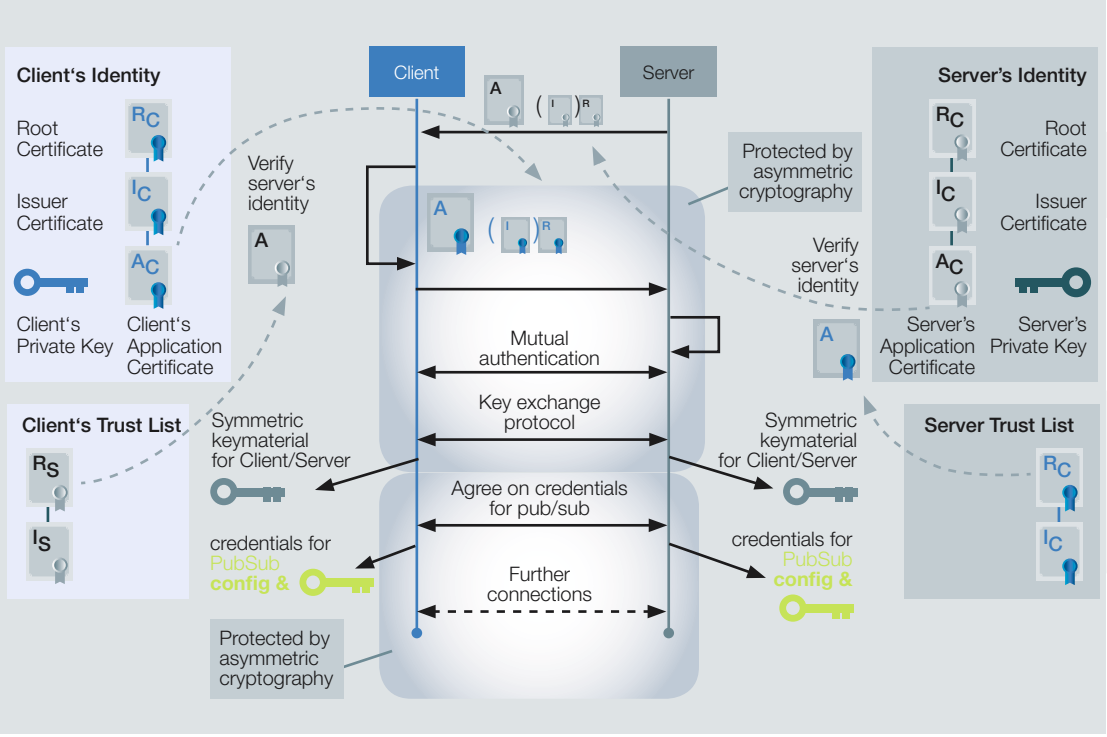


Figure 21: Mutual authentication plus obtaining credentials for PubSub

Transport

Communications Profiles

The current system architecture of a field level communications device allows for two communications profiles to be used with PubSub communications, constructed from at least four communications facets (see Figure 22).

All OPC UA ACs implemented using the Field Level Communications will be capable of supporting one or more of the communication profiles and any or all of its optional facets. Each profile can be used on a per-connection basis. Thus, if a controller conforms to multiple profiles, a control system engineer can make the optimum selection for each connection.

→ Profile A

Profile A is optimized for layer 3 networks typically deployed across an entire plant made of many machines, skids, and cells. Connections using Profile A deliver the most flexible network architecture but consume more network bandwidth and processing resources than those using Profile B. Profile A requires the use of both the Direct Network Access Facet and the UDP UADP Transport Facet. Any two Profile A components installed in an IP network are interoperable, provided IP routes can be established. ACs implementing Profile A may optionally implement the TSN Network Access facet in order to enable operation in a TSN Domain (see Page 35 et seq.).

Profile A and Profile B connections are not interoperable due to incompatible UADP Transport Facets and so an AC must implement both to support interoperability in all cases.

→ Profile B

Profile B is optimized for the layer 2 networks typically seen within a single skid, cell or machine; connections using the Profile B deliver the most efficient network bandwidth usage and performance but cannot operate through a layer 3 switch or router.

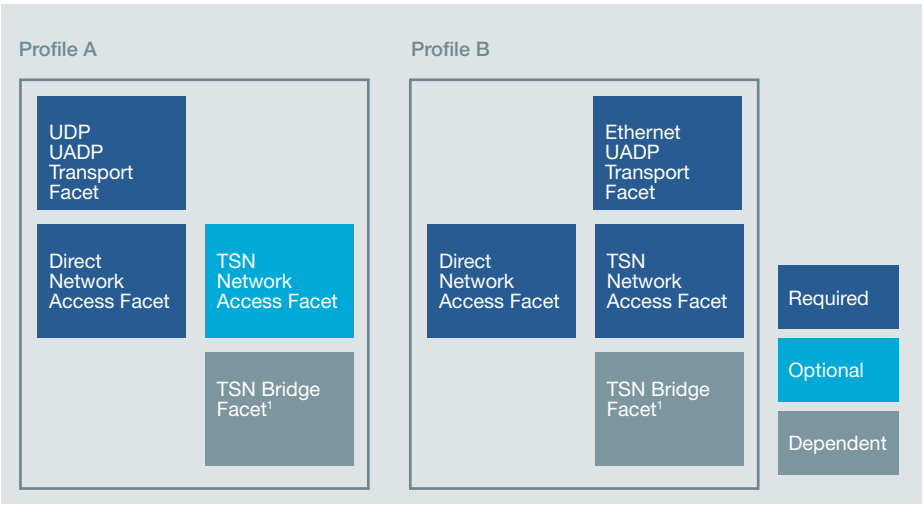


Figure 22: Structure of communications profiles

[1] If an AC supports the TSN bridge facet and is implemented with an embedded Ethernet bridge, then it shall implement the TSN Bridge Facet.

Profile B requires support of the TSN Network Access Facet, the Direct Network Access Facet, and the Ethernet UADP Transport Facet. Any two Profile B components installed in the same TSN Domain are interoperable.
In order to support operation in non-TSN enabled Ethernet networks, a Profile B connection also supports the Direct Network Access Facet.

Transport and Network Access Facets

→ Transport Facets

Ethernet UADP and UDP UADP transport facets are defined in the OPC UA Specifications Part 14 PubSub (OPC 10000-14) and Part 7 Profiles (OPC 10000-7), and are used as-is.

→ Direct Network Access Facet

Communications using non-TSN Ethernet point-to-point or multicast techniques, using the PCP or DSCP prioritization mechanisms are defined in the Direct Network Access Facet. This means that a connection using this facet is infrastructure agnostic, supporting managed and unmanaged switches in all their varieties including TSN switches. In heavily loaded networks, switch buffers may be filled and packets dropped, or latency and jitter may be created resulting in packets being delivered too late to be used by the control application.

→ TSN Network Access Facet

The TSN Network Access Facet is a superset of the Direct Network Access Facet and uses TSN Ethernet point-to-point or multicast techniques using TSN streams. Any two end stations using this facet with TSN bridges along their path can guarantee zero congestion packet loss and bounded network latency and jitter. Even in heavily loaded TSN networks, high priority streams suffer no effects of this loading.

→ TSN Bridge Facet

Network interfaces implementing a TSN bridge are characterized by the TSN Bridge Facet. The Field Level Communications initiative is committed to supporting the IEC/IEEE 60802 TSN Profile for Industrial Automation. It is expected that all Industrial Ethernet variants and IT devices operating in an industrial network using TSN will align with this specification, allowing them to fairly share network resources.
TSN Domains require all bridges (either embedded in an AC or in an infrastructure switch) to comply with IEC/IEEE 60802. If an AC implements the TSN Network Asset Facet and it implements an embedded switch, then it must implement the TSN Bridge Facet and that switch must be an IEC/IEEE 60802-compliant bridge. The mechanisms to connect multiple TSN Domains in a single network and to extend TSN Domains through network routers have yet to be standardized.

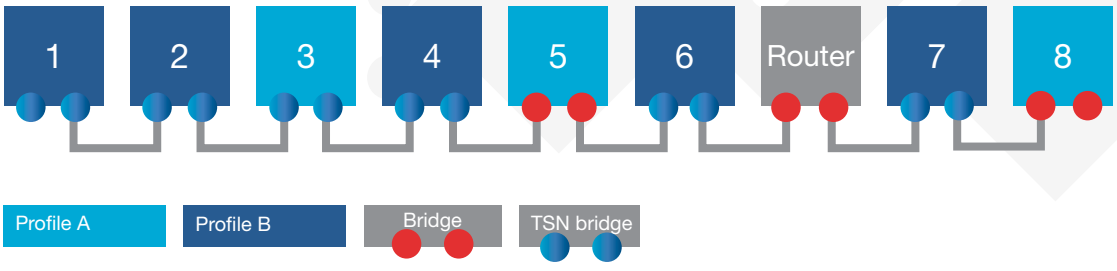


Figure 23: Topology to illustrate interoperability in a mixed Profile A/Profile B scenario

Interoperability Matrix

Figure 23 shows a potential configuration of ACs that illustrates all possible network profile interoperability use cases (allowing for both required and optional facet implementations).

Table 1 shows which of the ACs are interoperable, and indicates likely constraints of that interoperability.

Interoperability only applies to the ability for two components to exchange process data. There are other attributes, typically associated with a device's full profile, that may impact its ability to meet application requirements. For instance, if devices 2 and 6 are motion devices, interoperability does not imply that the system will achieve the update rates, latency and jitter required to successfully control a specific application.

If devices 1 and 7 have implemented support for both, Profile A and Profile B, device 1 would be interoperable with all devices (see Figure 24) and therefore its row would be as shown in Table 2.

	Device 1	Device 2	Device 3	Device 4	Device 5	Device 6	Device 7	Device 8
Device 1		Yes	No[1]	Yes[2]	No[1]	Yes[3]	No[4]	No[1]
Device 2			No[1]	Yes[2]	No[1]	Yes[3]	No[4]	No[1]
Device 3				No[1]	Yes	No[1]	No[1]	Yes[5]
Device 4					No[1]	Yes[3]	No[4]	No[1]
Device 5						No[1]	No[1]	Yes[5]
Device 6							No[4]	No[1]
Device 7								No[1]

[1] Different communication profiles
[2] All devices between devices implement an embedded IEC/IEEE 60802 compliant bridge
[3] Does not use TSN QoS
[4] Layer 2 communications cannot traverse a router
[5] UDP communications are routable

Table 1: Interoperability matrix (Device 1 supporting Profile B)

	Device 1	Device 2	Device 3	Device 4	Device 5	Device 6	Device 7	Device 8
Device 1		Yes	Yes	Yes	Yes	Yes[3]	Yes	Yes

Table 2: Interoperability matrix (Devices 1 and 7 supporting both, Profile A and Profile B)

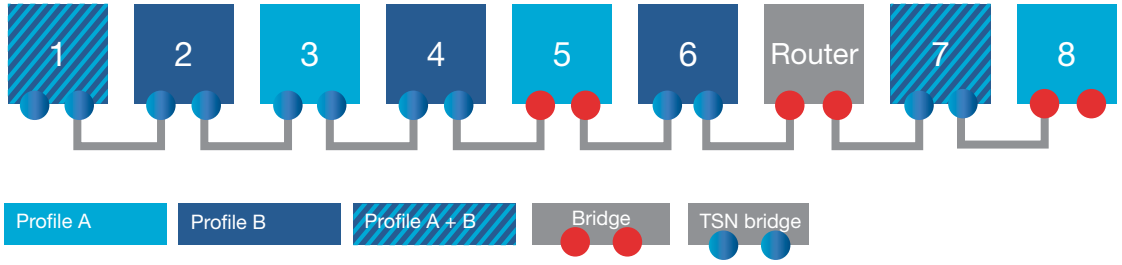


Figure 24: Topology to illustrate interoperability in a mixed Profile A/Profile B scenario

Ethernet – Advanced Physical Layer

The OPC UA framework is transport-agnostic and therefore can be used with different underlying protocols (e.g. TCP, UDP, MQTT, ...) and physical layers. To bring OPC UA down to the field level in process industry applications, OPC UA is combined with the Ethernet Advanced Physical Layer (APL) which will be addressed in a later specification version in the context of the controller-to-device use case.

Ethernet – Advanced Physical Layer [1]

Ethernet-APL is an enhanced physical layer for single-pair Ethernet (SPE) based on 10BASE-T1L as shown in Figure 25. It communicates via a cable length of up to 1000 m at 10 MBit/s, full-duplex. It is the logical extension for Ethernet and provides the attributes required for reliable operation in the field of a process plant. Ethernet-APL is a physical layer that will be able to support OPC UA or any other higher-level protocol.

Ethernet-APL is designed to support various installation topologies, with optional redundancy or resiliency concepts and trunk-and-spur. Ethernet-APL explicitly specifies point-to-point connections only with each connection between communications partners constituting a “segment”. Thus, Ethernet-APL switches isolate communications between segments. This eliminates disturbances such as cross talk and natively protects communications from device faults on a different segment.

Ethernet-APL defines two general types of segments:

- → The “Trunk” provides high power and signal levels for long cable lengths of up to 1000 m.
- → The “Spur” carries lower power with optional intrinsic safety for lengths of up to 200 m (2-WISE).

2-WISE stands for 2-Wire Intrinsically Safe Ethernet. This IEC technical specification, IEC TS 60079-47 (2-WISE), defines intrinsic safety protection for all hazardous Zones and Divisions. For users, this includes simple steps for verification of intrinsic safety without calculations.

Ethernet-APL combines the best attributes of Ethernet communication with two-wire installation techniques. This makes Ethernet-APL easy to deploy as a standard for field applications, from process plants with hazardous areas up to Zone 0 / Division 1 to hybrid plants, employing technologies from factory automation and process automation. Consequently, the use of Ethernet-APL as a physical layer for OPC UA field devices is a key driver for successfully bringing OPC UA down to the field level in process automation applications.

ethernet-apl™
advanced physical layer

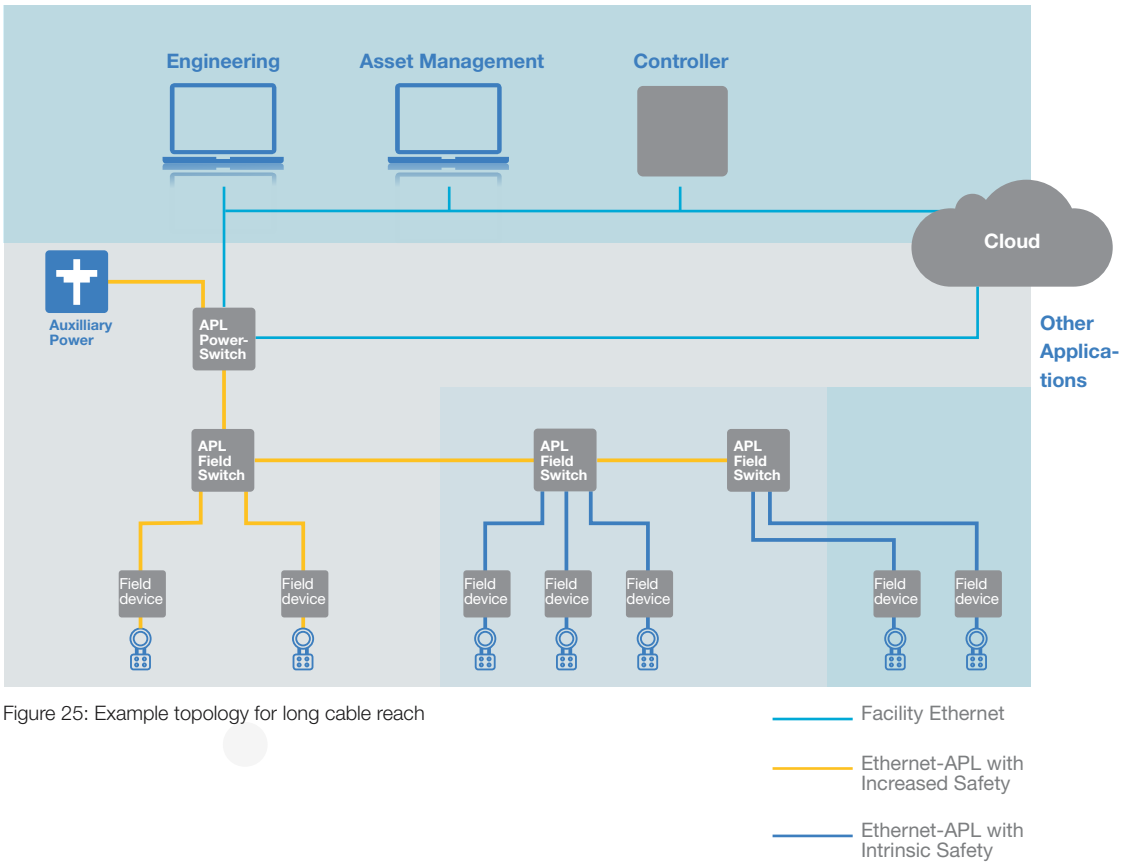


Figure 25: Example topology for long cable reach

[1] Extract from “ethernet-apl advanced physical layer. Ethernet to the field”
https://opcfoundation.org/wp-content/uploads/2020/06/Ethernet-APL_Ethernet-To-The-Field_EN.pdf

Real-Time Communication Model

Quality of Service (QoS) concept

QoS refers to network control mechanisms that can provide various priorities to different devices or data flows, or guarantee a certain level of performance to a data flow in accordance with requests from the application program. QoS guarantees are important if the network performance is critical, especially for real-time control applications.

Prior to the arrival of TSN, the most common approach to delivering QoS in industrial automation networks was by providing differentiated services to different types of traffic. In this approach, some types of traffic are treated better than others by classifying the traffic and using tools such as priority queuing, enabling faster handling, higher average bandwidth, and lower average loss rate for the chosen types. However, this only provides a statistical preference, not a hard and fast guarantee. Different types of industrial Ethernet traffic (such as motion, I/O, and HMI) have different requirements for latency, packet loss, and jitter. The service policy should differentiate services for these types of flows.

The Field Level Communications initiative defines provisions for identifying important OPC UA traffic at the field level with both Layer 3 DSCP (Differentiated Services Code Point, defined in IETF RFC 2474, etc.) and Layer 2 CoS (Class of Service, defined in IEEE 802.1Q) tags for use in non-TSN managed networks.

TSN provides standardized mechanisms to deliver guaranteed service by reserving specific resources from the network for specific types of traffic. Such network guarantees must be mapped to the network application or middleware such as OPC UA PubSub. Application QoS requirements of an OPC UA application should be configurable with no or only little dependencies to the underlying network technology. Hiding network details from the application makes it easier for the application builder to migrate OPC UA applications from one network technology to another or even to interconnect OPC UA applications over different network technologies.

TSN QoS mechanisms

The IEC/IEEE 60802 TSN Profile for Industrial Automation defines a selection of QoS mechanisms specified by the IEEE 802.1 TSN Task Group for use in converged industrial automation networks.

Converged networks promise to enable operational technology (OT) which includes traditional field buses such as PROFINET or EtherNet/IP, and traffic to operate the plant e.g. HMI/SCADA/MES communication to PLCs, and information technology (IT) applications to share the same physical network infrastructure without hampering operation of the other. For many industrial control applications, this implies that certain bandwidth, latency, and deadline requirements must be met, especially in situations where there is contention for network resources.

Types of traffic and their QoS requirements

To allow for the coexistence of different applications on the same network, the infrastructure components have to provide the means to transport different types of traffic with appropriate QoS. The traffic types defined by the Field Level Communications initiative allow for convergence of different types of OT traffic (e.g. process control, factory automation, and fast motion and I/O control) and IT traffic using the same infrastructure.

The following traffic types are defined:

→ Network Control
→ Cyclic Control
→ Event-based Control
→ Configuration and Diagnostics
→ User-defined and
→ Best Effort

A system-wide implementation of these traffic types allows for convergence of factory automation, process control, IT traffic and best-effort traffic on the same network.

TSN Domains and examples for communication relations

Today, many network architectures for industrial automation systems follow a certain physical and logical separation into domains or zones. This separation is often the result of organizational or technical requirements, e.g., interconnection of individual components or entire machines from different vendors, each with their own validated communication network and configuration, or the implementation of zones for security best practice, or to support network redundancy and to further enhance network QoS guarantees. These requirements also may necessitate a logical separation when using TSN.

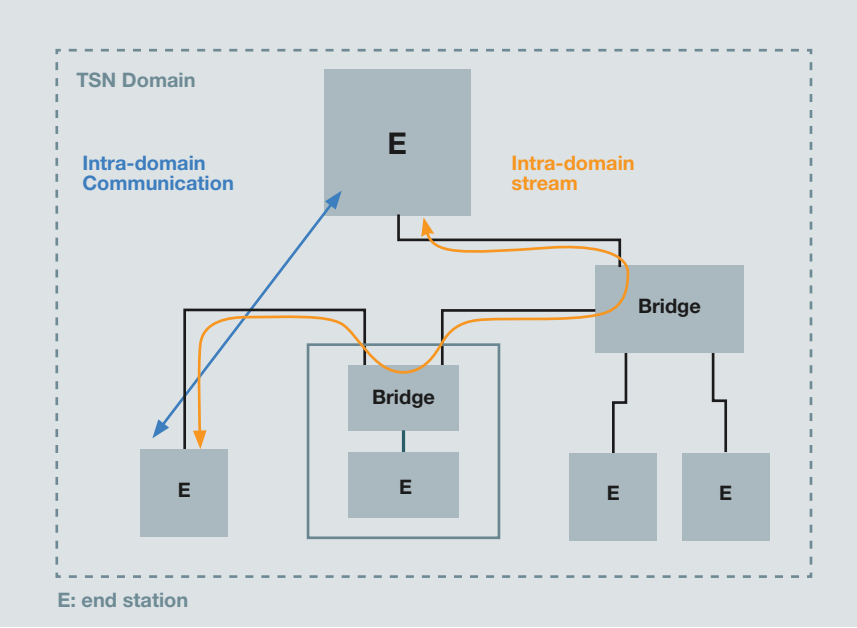


Figure 26: Intra-domain communication

Moreover, separation into different TSN Domains is intended to allow for the centralized and distributed TSN stream reservation approaches to operate side-by-side.

TSN intra-domain and inter-domain communication are distinguished. The selected stream reservation mechanism enables an industrial control application to reserve network resources for the selected TSN QoS mechanisms within the given TSN Domain. This allows leveraging TSN-provided bandwidth and timing guarantees in converged network scenarios, as shown in Figure 26. Intra-domain communication can be utilized to realize C2C, C2D, and D2D communication relations.

Inter-domain communication occurs in communications scenarios for data exchanges of industrial control applications across (multiple) domains. It can be utilized to realize C2C, C2D, and D2D communication relations.

Figure 27 shows such inter-domain communication for a C2C scenario traversing TSN Domains 1, 2, and 3.

As an alternative for inter-domain stream reservations and as a state-of-the-art approach to interconnecting different domains, e.g., representing machines in today's systems, the exchange of process data between two domains (e.g., Domain 1 and Domain 2 in Figure 28) may also logically be decoupled from on-the-wire communication and corresponding TSN stream reservation via application-level gateways.

Table 3 lists examples of communication relationships utilizing either intra- or inter-domain communication.

Inter-domain communications with TSN represent future work in IEC, IEEE, and IETF and so will not be addressed in early releases of OPC UA Field Level Communications specifications. Where inter-domain communications are required by an application, Profile A will ensure interoperable communications between ACs.

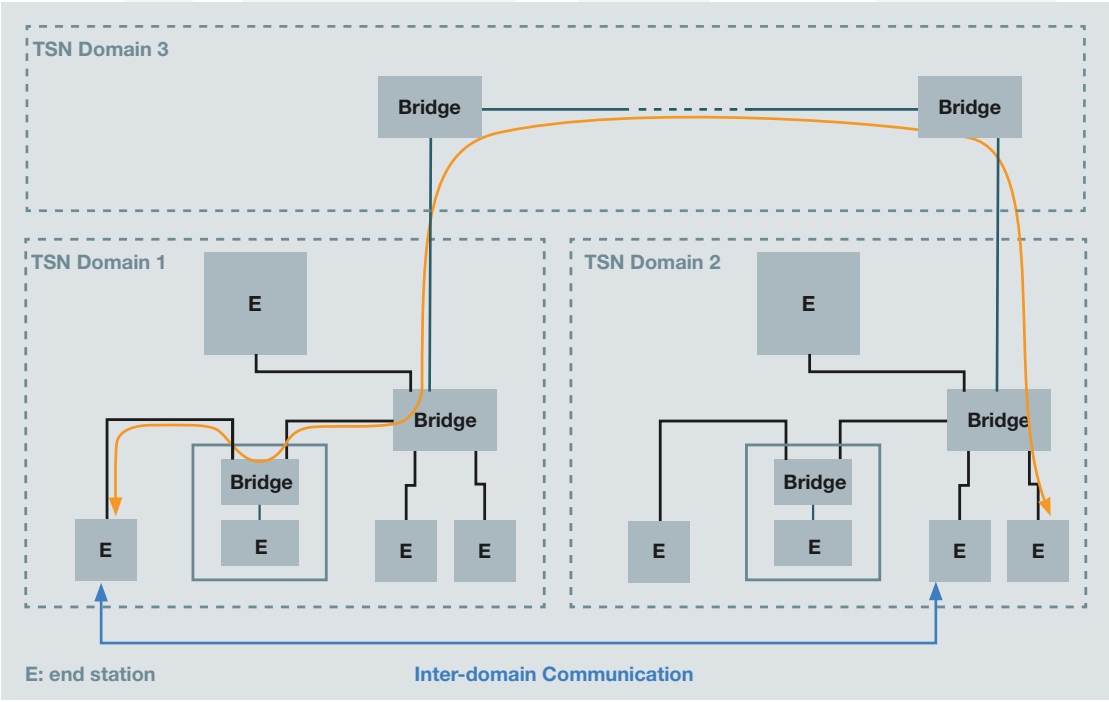


Figure 27: Inter-domain communication

Network Management

Network Management for OPC UA Field Level Communications will be based on open standards. For distribution of a successful network configuration standardized management protocols will be utilized. The actual configuration parameters will be modelled in IEEE-conformant data models.

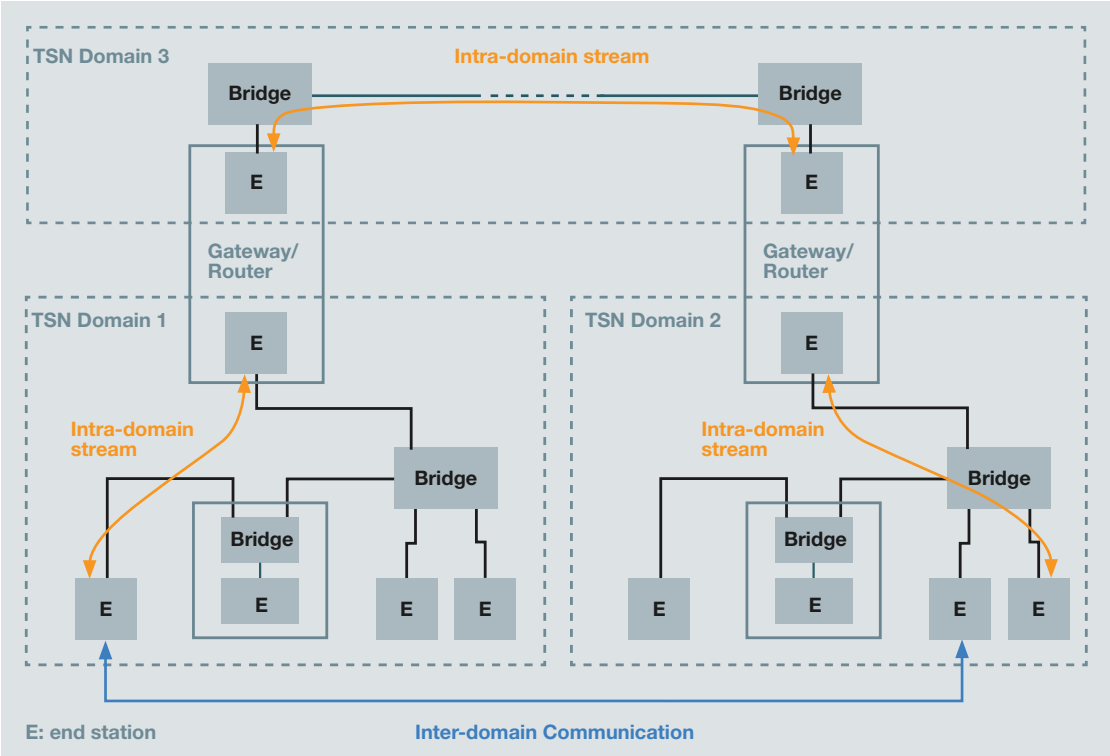


Figure 28: Connection of Domains using application level gateways or DetNet routers

Communication relation	Description / Example
C2D Intra-domain communication	This is probably the most common relationship, when a controller communicates with its peripheral (I/Os, drives, valves, …)
C2C Intra-domain communication	Communication between multiple controllers in the same TSN Domain
D2D Intra-domain communication	To improve reaction times the devices (I/Os, drives, …) sometimes need to establish direct communication
C2D Inter-domain communication	controller synchronizes on encoder signal from a different TSN Domain
C2C Inter-domain communication	Interconnection of machines/skids without dedicated gateways (without controller)
D2D Inter-domain communication	synchronization between motions drives in different TSN Domains

Table 3: Examples of communication relationships

Summary and Outlook

This technical paper described how the Field Level Communications initiative extends the OPC UA framework to facilitate cross-vendor interoperability between controllers by enabling the exchange of data relevant for different use cases including the exchange of real-time and safety-relevant data in a secure way.

After the first specification release with the focus on the controller-to-controller use case, the specifications will be extended to also support controller-to-device (C2D) and device-to-device (D2D) use cases including additional features and device-specific models, e.g. for motion, instrument, I/O, and safety devices.

In parallel to the creation of specifications, open source stack software and code samples are being generated so that an easy adoption of OPC UA for field level communications is facilitated. Furthermore, test specifications and test tools are being developed to provide high-grade cross-vendor interoperability between Automation Components.

With the extensions specified by the Field Level Communications initiative, OPC UA in combination with APL, TSN, and 5G offers a complete, open, standardized, and interoperable solution that fulfils industrial communication requirements and at the same time provides semantic interoperability from field to cloud (see Figure 29).

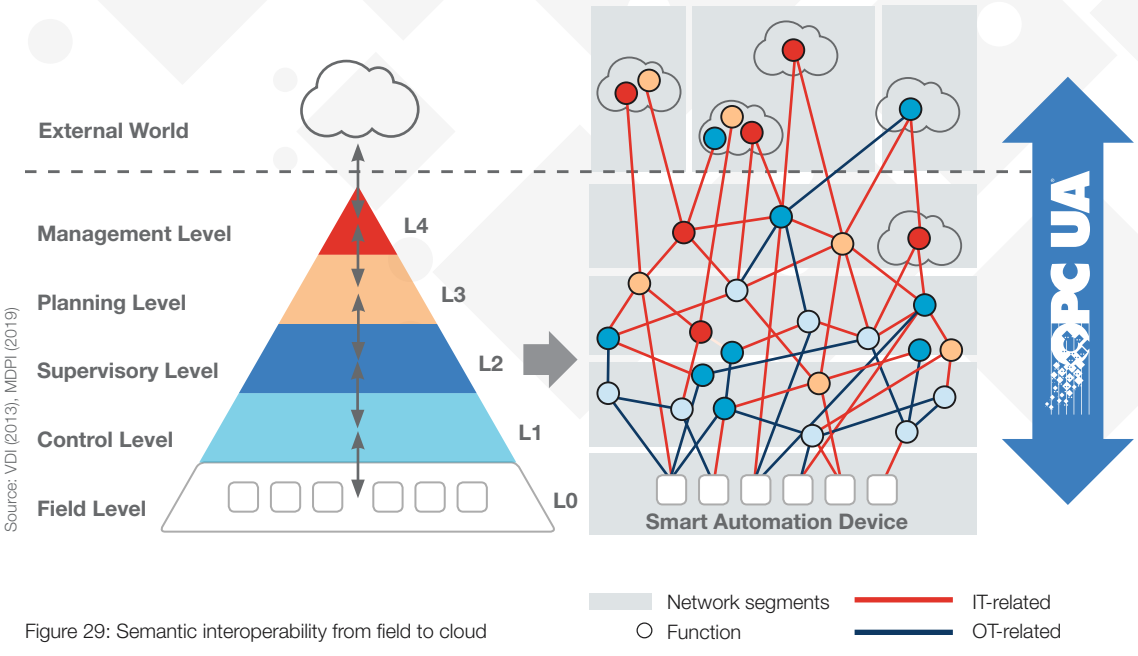


Figure 29: Semantic interoperability from field to cloud

Acronyms

AC	Automation Component
APL	Advanced Physical Layer
C2C	Controller-to-Controller
C2D	Controller-to-Device
CD	Configuration Descriptor
CM	Connection Manager
CNC	Central Network Configuration
CR	Communication Relationship
CUC	Centralized User Configuration
D2D	Device-to-Device
DCS	Distributed Control System
DSCP	Differentiated Services Code Point
ERP	Enterprise Resource Planning
FE	Functional Entity
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IIoT	Industrial Internet of Things
IoT	Internet of Things
IT	Information Technology
L2	Layer 2
L3	Layer 3
MES	Manufacturing Execution System
OE	Offline Engineering
OPC	Open Platform Communication
OPC UA	OPC Unified Architecture
OPCF	OPC Foundation
OT	Operational Technology
PAC	Programmable Automation Controller
PD	Product Descriptor
PCP	Priority Code Point
PLC	Programmable Logic Controller
QoS	Quality of Service
SCADA	Supervisory Control and Data Acquisition
SPE	Single-Pair Ethernet
TSN	Time-sensitive Networking
UADP	Unified Architecture Datagram Packet
UDP	User Datagram Protocol

OPC FOUNDATION HEADQUARTERS
OPC Foundation
16101 N. 82nd Street, Suite 3B
Scottsdale, AZ 85260-1868 USA
Phone: 480 483-6644
office@opcfoundation.org

OPC FOUNDATION EUROPE
opceurope@opcfoundation.org

OPC FOUNDATION CHINA
opcchina@opcfoundation.org

OPC FOUNDATION JAPAN
opcjapan@opcfoundation.org

OPC FOUNDATION KOREA
opckorea@opcfoundation.org

OPC FOUNDATION ASEAN
opcasean@opcfoundation.org

OPC FOUNDATION INDIA
opcindia@opcfoundation.org