Extending OPC UA to the field: OPC UA for Field eXchange (FX)

Version 3 // November 2023

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 field to cloud, including field-level devices for all relevant industrial automation use cases, including real-time, motion, instrumentation, I/O, and functional safety.

Standardized communication from field 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 with bounded latency and jitter based on standard Ethernet.

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, i.e, controller-to-compute and device-to-compute, respectively. 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. The specifications to extend OPC UA for field-level communications are named OPC UA FX (Field eXchange), abbreviated UAFX.

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 and decommissioning. 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: Controller-to-Controller (C2C) data connectivity between shop floor system or process skids.

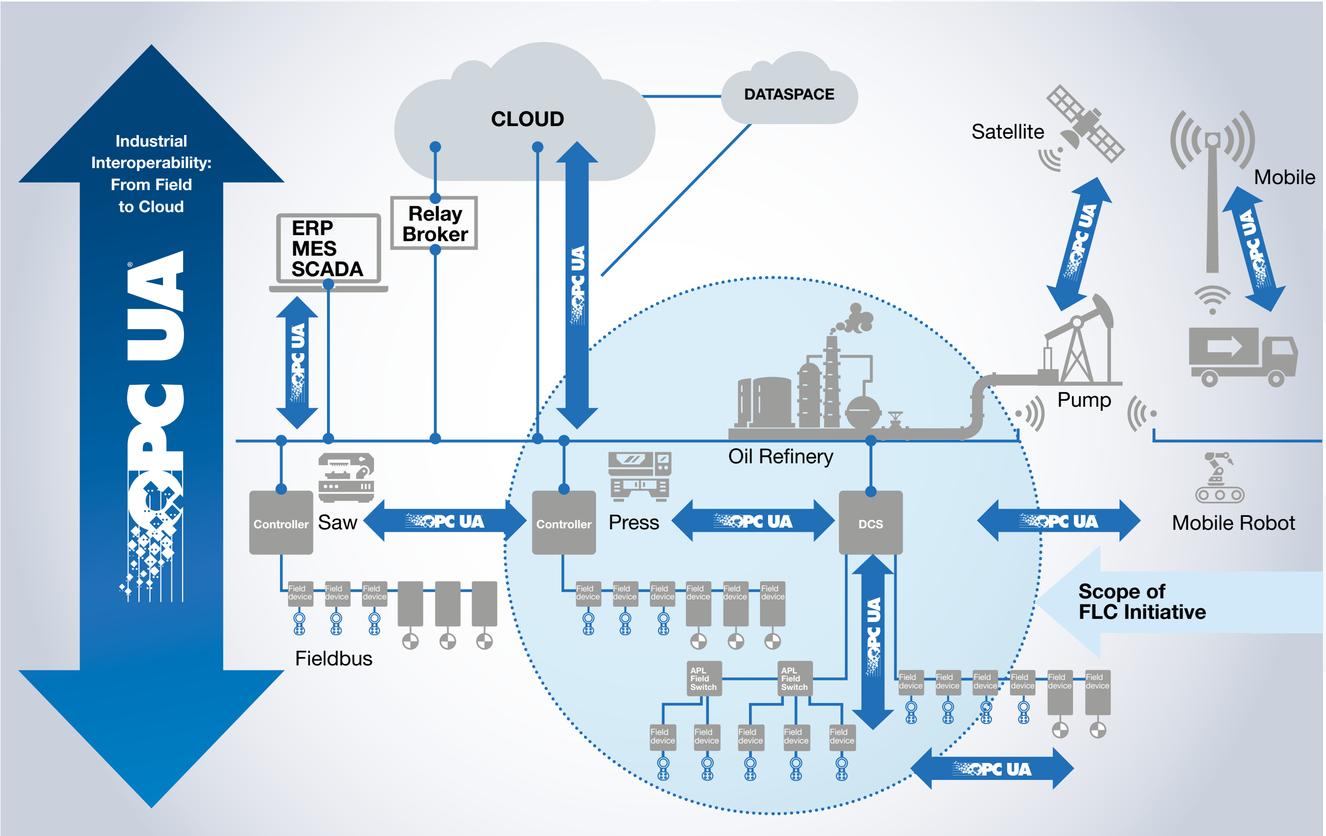


Figure 1: OPC UA use cases and scope of OPC Foundation's Field Level Communications (FLC) Initiative

An example of vertical communications is device-to-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. The OPC Foundation will continue to expand these 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 802.1Q, supports communication of specific data streams with bounded latency and jitter as required by some applications. It permits additional data streams and traffic types to be transmitted over a common network infrastructure in the remaining bandwidth. 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

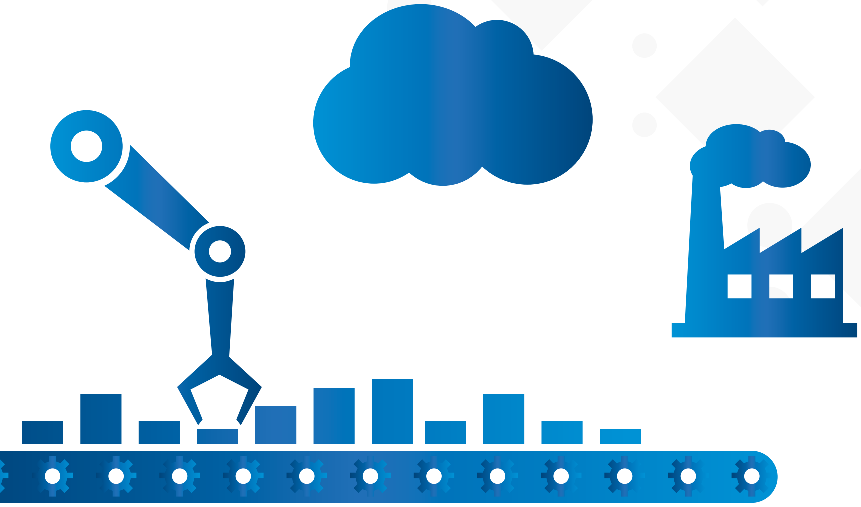
At the SPS IPC Drives Fair 2018 in Nuremberg, Germany, the OPC Foundation officially launched the Field Level Communications (FLC) Initiative. This Initiative aims at extending OPC UA to the field level resulting in an open, unified, standards-based IIoT communication solution between sensors, actuators, 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, instrumentation and motion, all requiring secured information exchange.

The IT and OT worlds are converging. The OPC Foundation with its Field Level Communications Initiative is committed to work with other organizations including IEEE 802.1 and TIACC (TSN Industrial Automation Conformance Collaboration) to achieve the following four types of convergence:

1. **UAFX application convergence** where multiple UAFX automation devices from multiple vendors share one network and exchange application data between each other.
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 gain 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 Overview** (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 OPC UA 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 UAFX connections, Connection Configuration Data and the Connection Manager are given, as they express the key concepts for exchanging data between multiple Automation Components.

3. In the section **Offline Engineering Workflow and Model** (pages 18 – 24) the descriptor concept is explained and the workflow for a control systems engineer is described to enable the C2C use cases prior to on-site commissioning. 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.

4. In the **Safety and Security** sections (pages 25 – 28) it is explained how data for functional safety applications is exchanged between Functional Entities by means of UAFX 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 UAFX connections are secured during connection establishment and data exchange against malicious attacks.

5. In the section **Transport** (pages 29 – 37) the UAFX-supported transport architecture is described. Furthermore, it is explained how interoperability is ensured using the concept of graceful degradation of Quality of Service (QoS). Afterwards the importance and the impact of Ethernet TSN and Ethernet-APL 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 explains what measures are taken to ensure an easy implementation of the technology as well as cross-vendor interoperability.

Technical System Overview

System Architecture
OPC UA is a 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, end users, research institutes and consortia, in order to enable secure and reliable information exchange in heterogeneous systems. Nevertheless, it lacks additional mechanisms needed to satisfy 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 (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 UAFX-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 I/O, motion control, functional safety, instrumentation
- standardization of OPC UA information models for field-level devices in online and offline scenarios e.g. device description, diagnostics, etc.

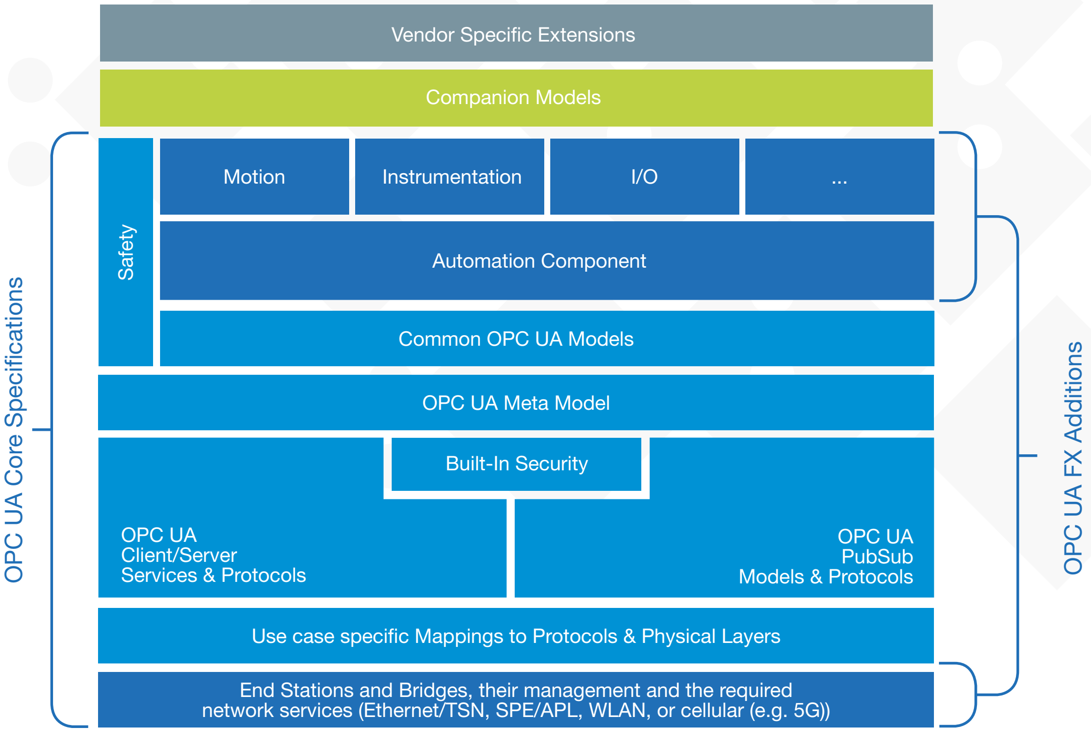

Vendor Specific Extensions
Companion Models
Safety
Motion
Instrumentation
I/O
...
Automation Component
Common OPC UA Models
OPC UA Meta Model
Built-In Security
OPC UA Client/Server Services & Protocols
OPC UA PubSub Models & Protocols
Use case specific Mappings to Protocols & Physical Layers
End Stations and Bridges, their management and the required network services (Ethernet/TSN, SPE/APL, WLAN, or cellular (e.g. 5G))
OPC UA Core Specifications
OPC UA FX Additions

Figure 2: Extending OPC UA for the field: OPC UA for Field eXchange (FX) System Architecture

- → integration of OPC UA companion models
- → support of Ethernet TSN for bounded latency and jitter
- → 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
- → utilization of OPC Foundation's certification procedures

The technical work results in specifications that extend the OPC UA framework. These specifications are identified as OPC UA FX (Field eXchange).

In the **first OPC UA FX 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 PubSub in combination with a peer-to-peer application relationship and basic diagnostics. The target network architecture is shown in Figure 3.

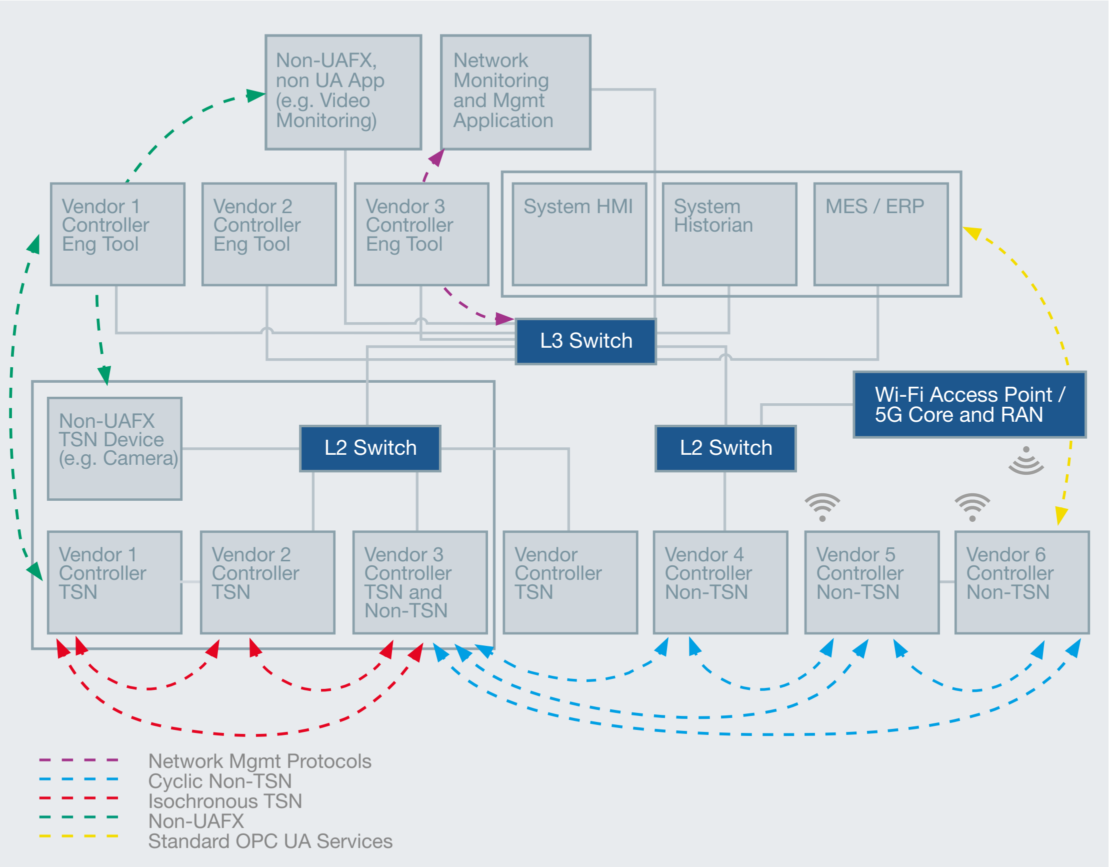

Non-UAFX, non UA App (e.g. Video Monitoring)
Network Monitoring and Mgmt Application
Vendor 1 Controller Eng Tool
Vendor 2 Controller Eng Tool
Vendor 3 Controller Eng Tool
System HMI
System Historian
MES / ERP
L3 Switch
Wi-Fi Access Point / 5G Core and RAN
Non-UAFX TSN Device (e.g. Camera)
L2 Switch
L2 Switch
Vendor 1 Controller TSN
Vendor 2 Controller TSN
Vendor 3 Controller TSN and Non-TSN
Vendor Controller TSN
Vendor 4 Controller Non-TSN
Vendor 5 Controller Non-TSN
Vendor 6 Controller Non-TSN
Network Mgmt Protocols
Cyclic Non-TSN
Isochronous TSN
Non-UAFX
Standard OPC UA Services

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 attributes and 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 controllers and devices.

→ 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 the need for an easy way to set up controller-to-controller communications across multiple vendors including real-time and safety data exchange. This problem has not been solved in industrial automation to date and the UAFX controller-to-controller solution created by the Field Level Communications 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 subset of I/O modules, drives, servos, instruments, and other smart Automation Components is well understood in the industrial automation community. However, it comes with constraints on network architecture and topology

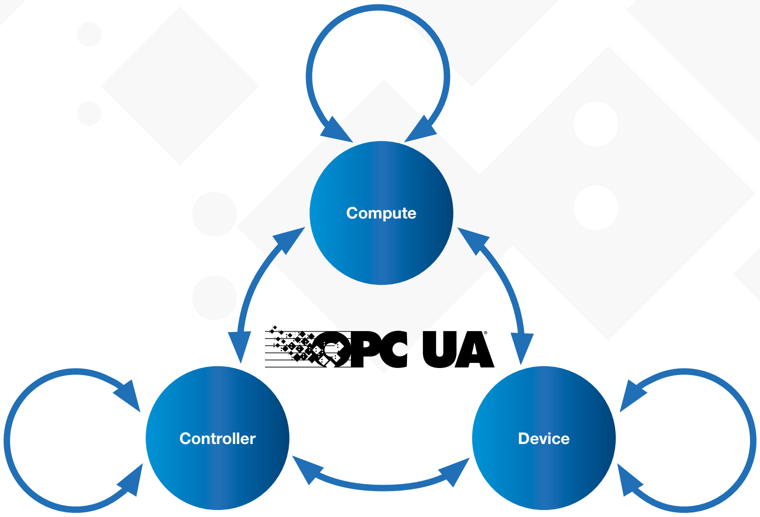


Figure 4: Interaction Model for OPC UA including field-level communications (UAFX)

when a converged IT/OT solution is deployed or different 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 existing IEC 61784 profiles.

→ Device-to-Device

To improve reaction times, devices such as I/Os, drives, … sometimes need to establish direct communication between each other. This also allows specific applications, such as load sharing of inflexible loads across multiple servo drives, to 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 amount of useful information 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. Furthermore the UAFX defined information models are designed to easily interact with the existing OPC UA compute models. While further development of capabilities to support compute-to-compute applications is not 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 communication 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 ecosystem of devices (see Figure 5). Similar examples exist with machines and distributed automation systems.

OPC UA FX supports a new, enhanced approach that makes use of OPC UA PubSub without preventing the currently available Client/Server mechanisms to exchange data between these machines:

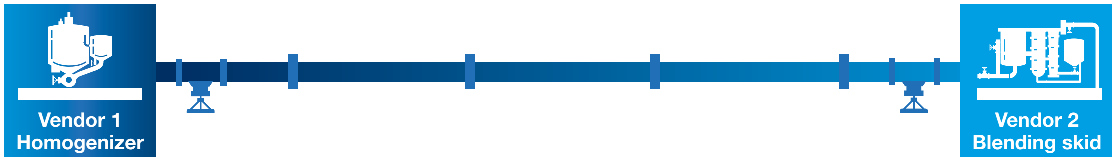


Figure 5: Controller-to-controller example

→ Unidirectional

The name unidirectional is derived from the flow of application data. Each machine's designer creates a configuration of output information available and supported configurations (update rates, security, etc.) for the other controller's use. Other machine designers can then import configuration and the supported configurations 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 machine controller 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 designers 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 in Machine 1 does not initiate any communication.
- Machine 2 controller initiates all communications, and its designer must ensure that it is both transmitting and receiving information in a format usable by Machine 1's application code.

In this case, Machine 1's behavior is very similar to that of an I/O module with fixed functionality in that its inputs and outputs are predefined.

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 can be controlled by the machine designer and the actual configuration or customization may be set using any generic OPC UA Client. An example of this: 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 the installation which machine is connected to which. Furthermore, potentially, there is no pre-planning of the network identification of each machine until it is 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 network identity of the target controller in the other vendor's machine. Moreover, additional information needs to be exchanged between the controllers, e.g. certificates.

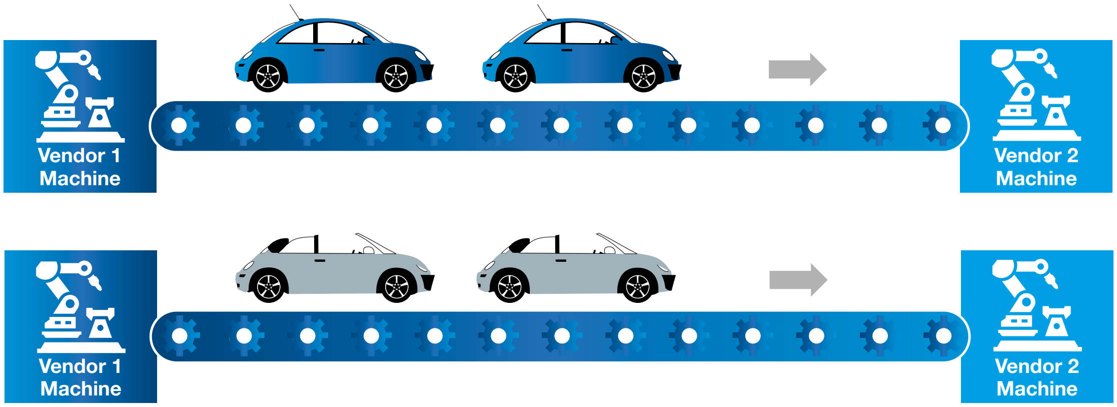


Figure 6: C2C Example with two identical lines (Case 1)

An additional example: two identical blending skids are integrated with the 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 the controller of the Vendor 1 homogenizer. However, further information must be given to both Vendor 2 skids, as the Vendor 1 controller has a unique connection for each Vendor 2 skid which functions in the same way but carry different data. The connections to the two Vendor 2 skid controllers (with Vendor 1 controller) rely on their distinct network identities which are utilized by the Vendor 1 controller.

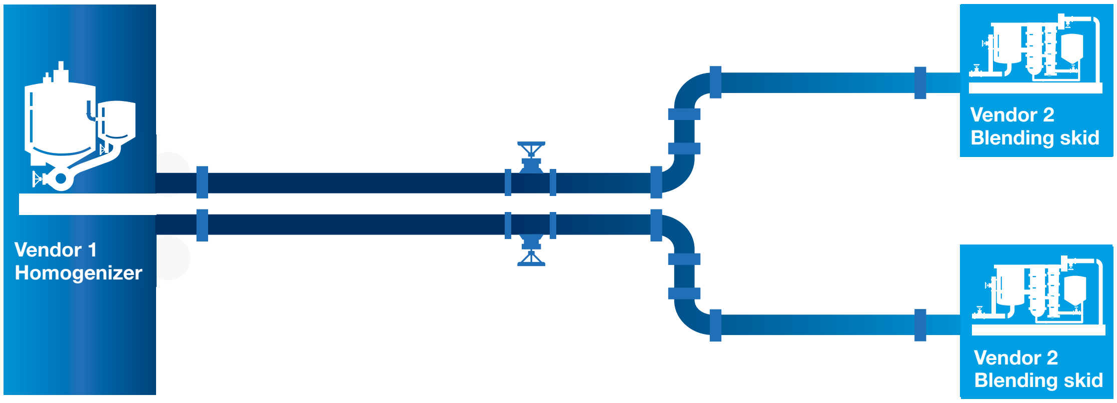


Figure 7: C2C Example with two identical skids in the same line (Case 2)

Automation Component Model

Automation Component – Functional Model and Asset Model

UAFX systems 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 make them automation devices (e.g. controllers, drives, instruments, I/O devices) (see Figure 8). All ACs are modeled as one or more Assets, and one or more Functional Entities. Additionally, Network Interfaces, and Network and Communication Services are provided which 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 is extended by a method for verifying the compatibility of an Asset. The Functional Model consists of one or more Functional Entities which encapsulate logical functionality. Functional Entities include input/output variables and configuration parameters, as well as supporting connections between Functional Entities. A Functional Entity (FE) is abstracted from the hardware, which allows porting of applications to new hardware. Functional Entities reference Assets that they are associated with or execute on, and allow applications

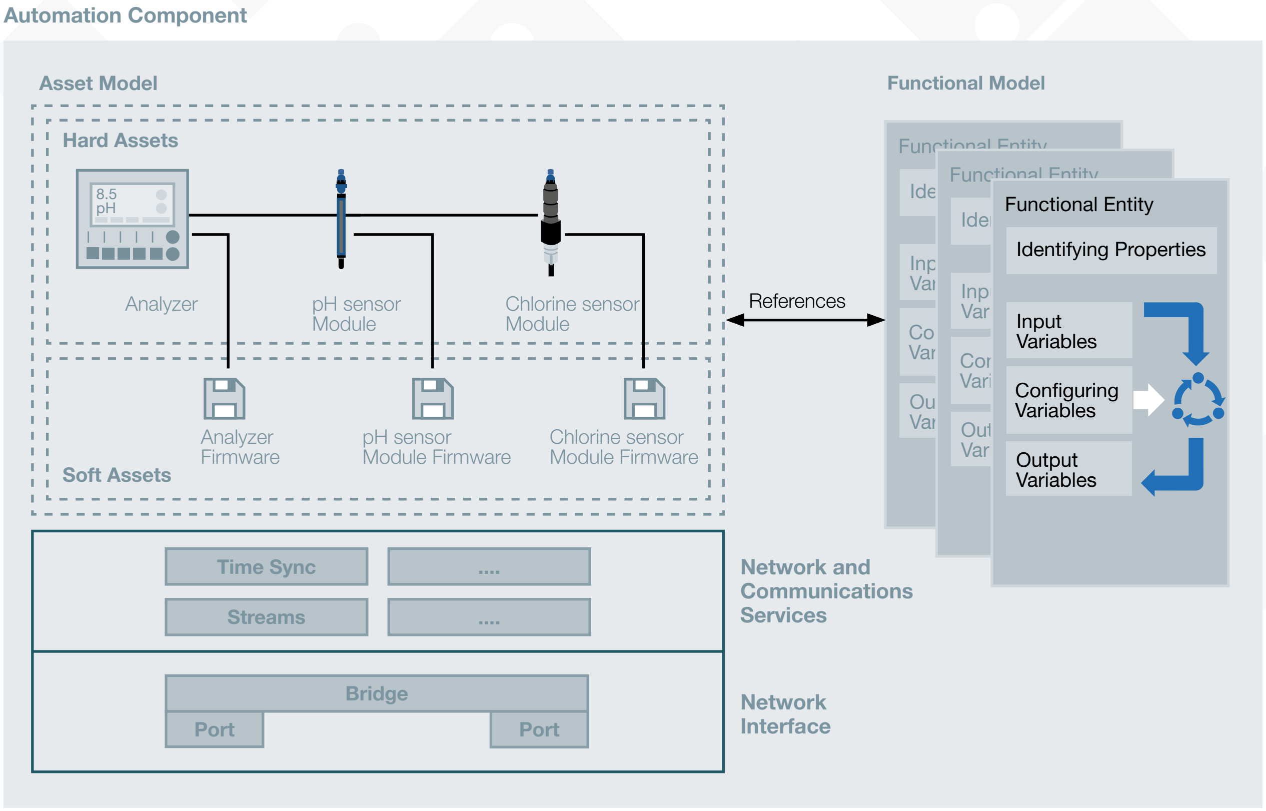


Figure 8: UAFX Automation Component Model

to confirm that any hardware requirements of the applications are met. For example, a two-axes drive may be based on one Asset including two single-axis Functional Entities.

From Functional Entity to Connections

A Functional Entity is an element of an AC that represents the 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 Functional Entities in an AC.

UAFX connections are the logical constructs used to exchange a defined set of process data and process data quality information between two Functional Entities. Inside a UAFX connection, PubSub DataSetWriter and DataSetReader elements are responsible for exchanging the data between the connected Functional Entities. As of now, UAFX connections are using only PubSub for the data exchange. However, in future, other mappings might also be defined, e.g. UAFX connections using Client/Server.
For exchanging process data, the following connection types are supported:

1. Unidirectional
2. Unidirectional with heartbeat
3. Bidirectional

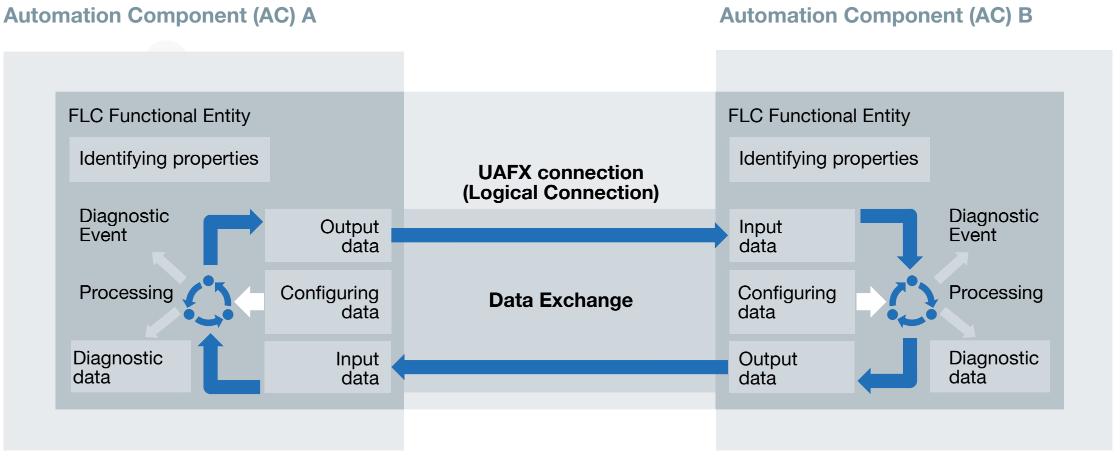


Figure 9: UAFX connections between Functional Entities

The Role of the Connection Manager

The Connection Manager (CM) is a service responsible for establishing connections between FEs (see Figure 10).

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

The CM uses the Connection Configuration Data to establish connections which can contain parameter for the following:

- → Address information for FEs to be connected
- → Choice of unicast or multicast
- → QoS (including TSN)
- → Connected process data
  - – Input/output variables
  - – Update rate
  - – Receive timeout
- → Cleanup timeout
- → Compatibility verification parameters
- → Parameters for configuring the application behaviour

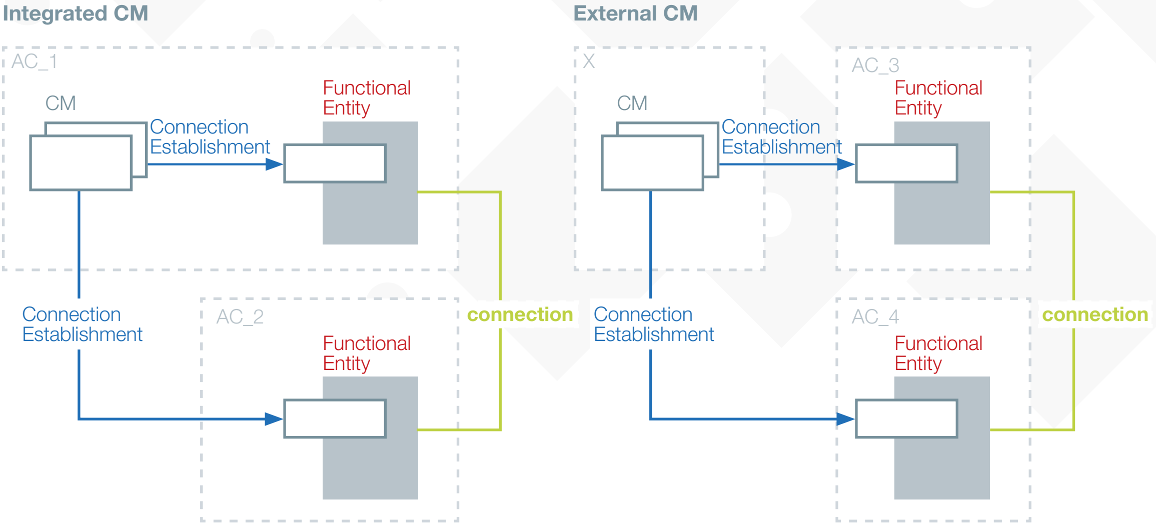


Figure 10: Integrated or External Connection Manager of Automation Components

Connection State Machine

The CM establishes a UAFX connection between two endpoints in parallel. On each of the endpoints a separate connection state machine exists (see Figure 11).

The UAFX Connection State Machine for each UAFX connection is defined in the information model of an FE.

A key feature of the state machine is the ability to detect a communication problem and to clean up connections in the event of a problem. For bidirectional or unidirectional with heart beat connection types this can include application problems on either side of a connection.

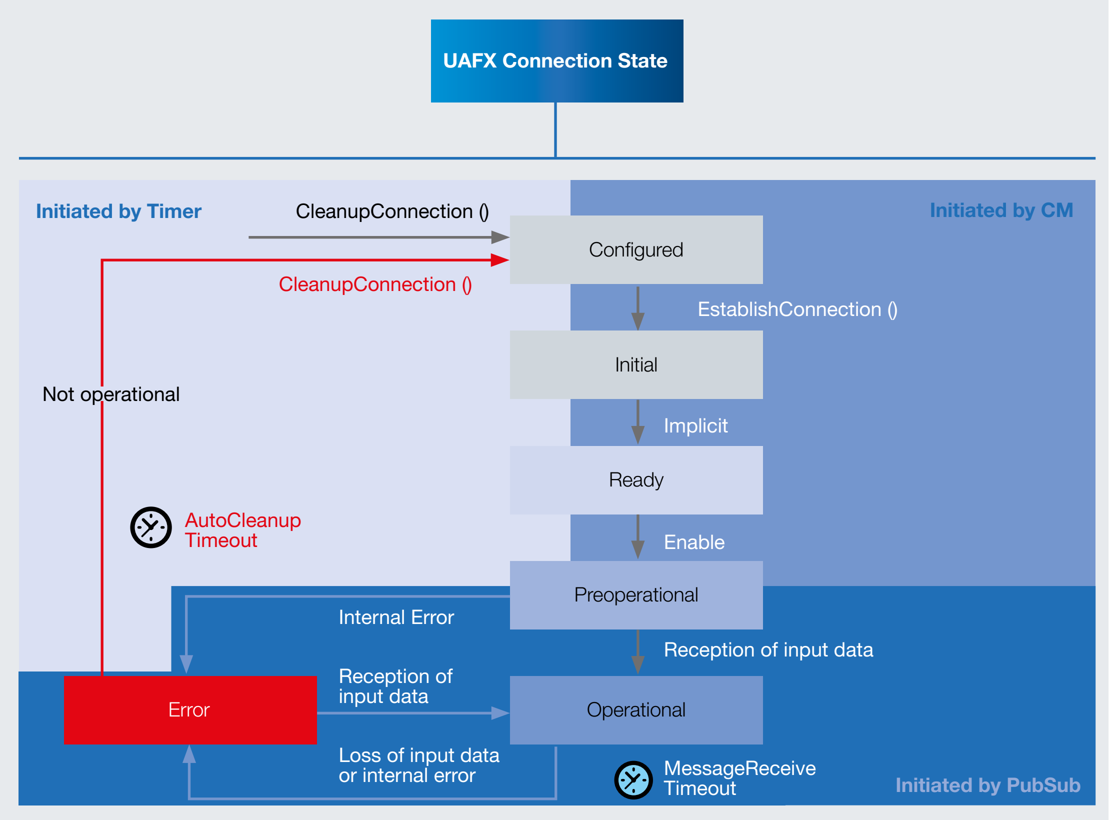


Figure 11: UAFX Connection State Machine

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 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 defines the Descriptor concept and describes the configuration workflow that creates and consumes Descriptors in the Offline Engineering phase.

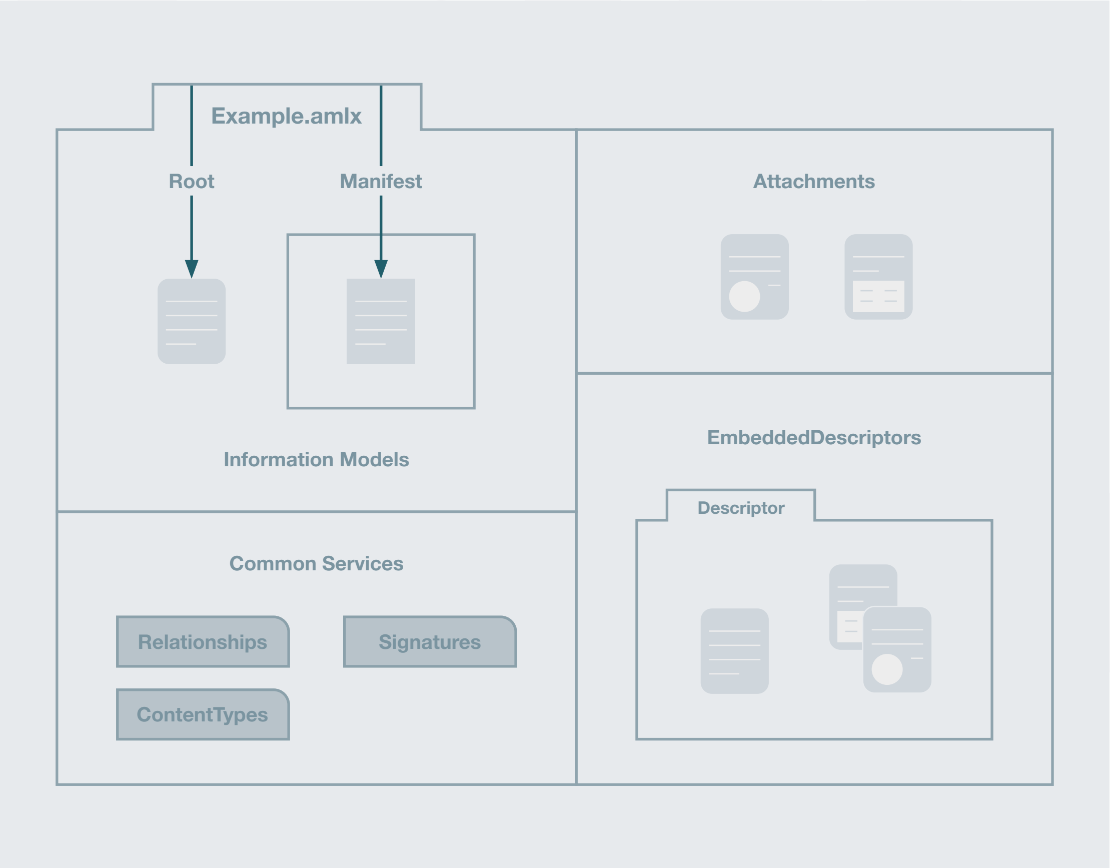


Figure 12: Descriptor Structure and Content

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 (e.g. a controller or a field device) or a group of ACs (e.g. machine, skid, modular I/O station). The AC Descriptor is delivered in a packaged container format (AML container) supporting the provisioning and sharing of information in offline engineering. A digital signature in the Descriptor provides integrity for the content.

There are five types of documents in a Descriptor (see Figure 12):

- A manifest file stores meta-data about the contents of the descriptor
- Information Model files provide the information model of the AC
- Attachment files provide supplemental and/or optional vendor-specific material
- Common Services files provide internal and external references and digital signatures
- Embedded Descriptors make it possible to include a Descriptor within a Descriptor. This can be useful for creating a completely self-contained Descriptor without external references or creating a module or a skid composed of multiple ACs.

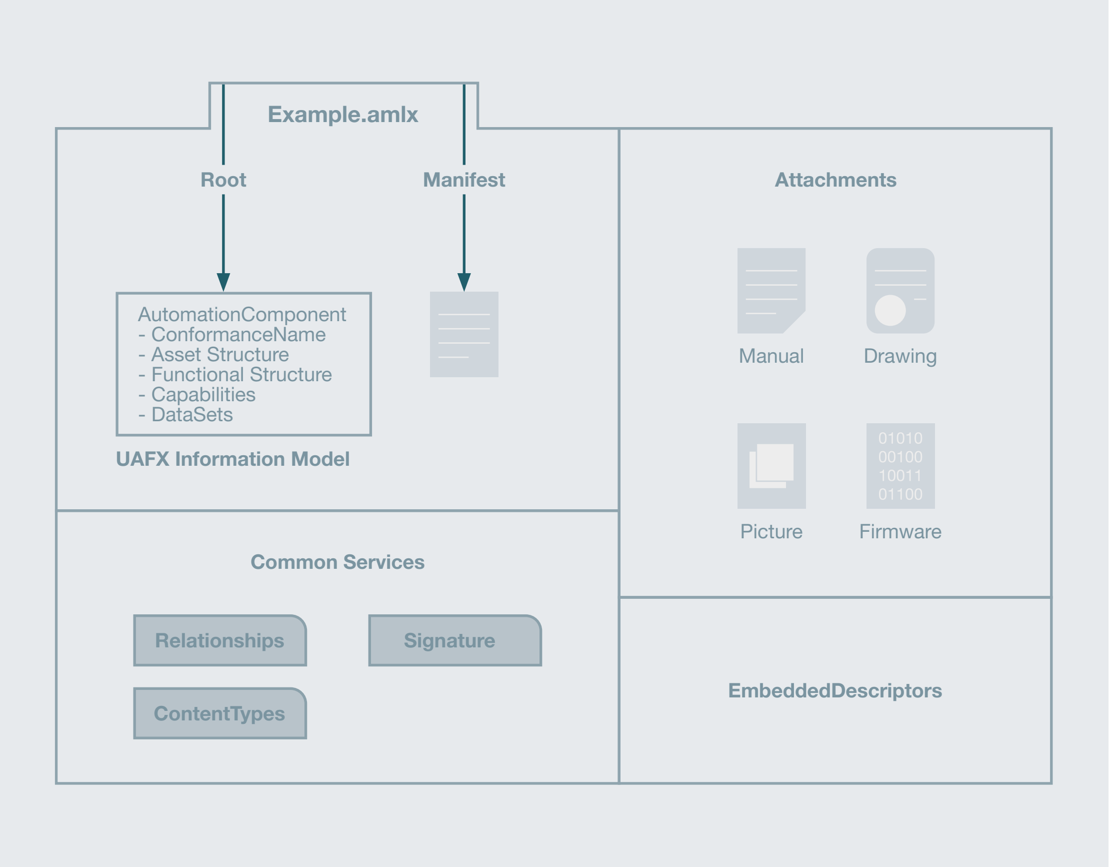


Figure 13: The Product Descriptor

The OPC UA Information Model of a Descriptor is defined using the AutomationML (AML) language. AML is a vendor-neutral XML-based format for the storage and exchange of engineering information. Two examples of AC Descriptors are described in the following section.

Product Descriptor

A descriptor with product information, which is called Product Descriptor, is a specific AC Descriptor containing product data of the AC (see Figure 13). Usually, the Product Descriptor is provided by the AC vendor. Importing the Product Descriptor into an engineering tool can be the first step when engineering an AC. In most cases, the Product Descriptor is included (or referenced) in another descriptor (e.g. Configuration Descriptor, see Figure 14).

The Product Descriptor states the identification, structure, features and capabilities of the AC. For some ACs (e.g. field devices), the Product Descriptor may also contain information about the AC's Functional Entities.

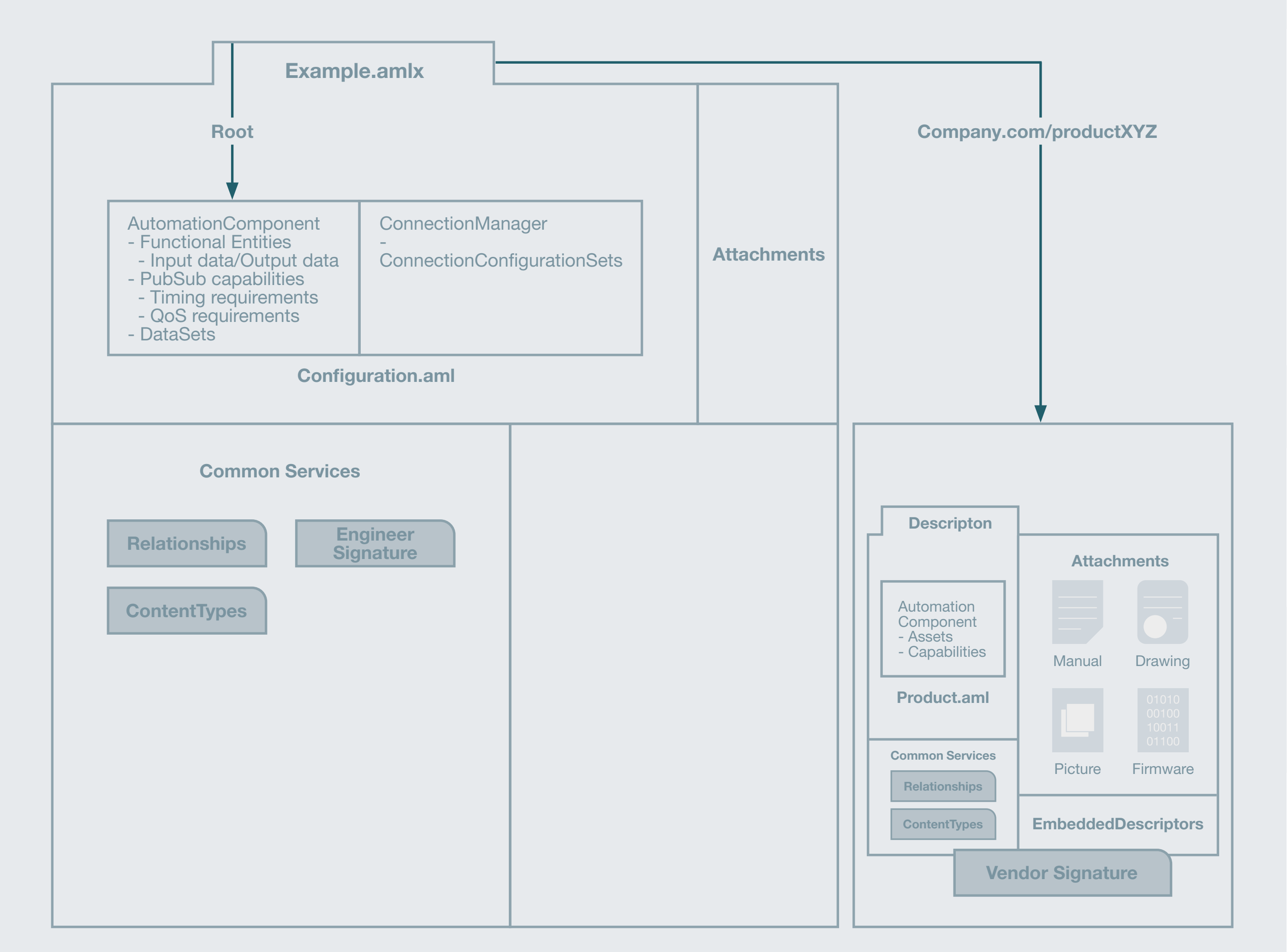


Figure 14: The Configuration Descriptor referencing a Product Descriptor

Configuration Descriptor

The Configuration Descriptor shown in Figure 14 is a specific AC Descriptor containing configuration information of an AC. It usually contains or references the Product Descriptor on which the configuration is based. The Configuration Descriptor is created in the engineering process, usually with the intention of sharing engineering information of an AC with another engineering tool.
The information model of the Configuration Descriptor defines the Functional Entities, the PubSub DataSets, the required Quality of Service (QoS) and the data necessary for connection establishment (such as unicast or multicast addresses for OPC UA PubSub). In addition, for a field or I/O device, the information model may also contain parametrization data.

Offline Engineering Workflow using Descriptors

Figure 15 gives an overview about the workflow steps for offline descriptor(s) usage.

Remark: For a better understanding the terms Product Descriptor and Configuration Descriptor are being used. However, the specification OPC 10000-83 UAFX Offline Engineering is using Descriptor as a general term.

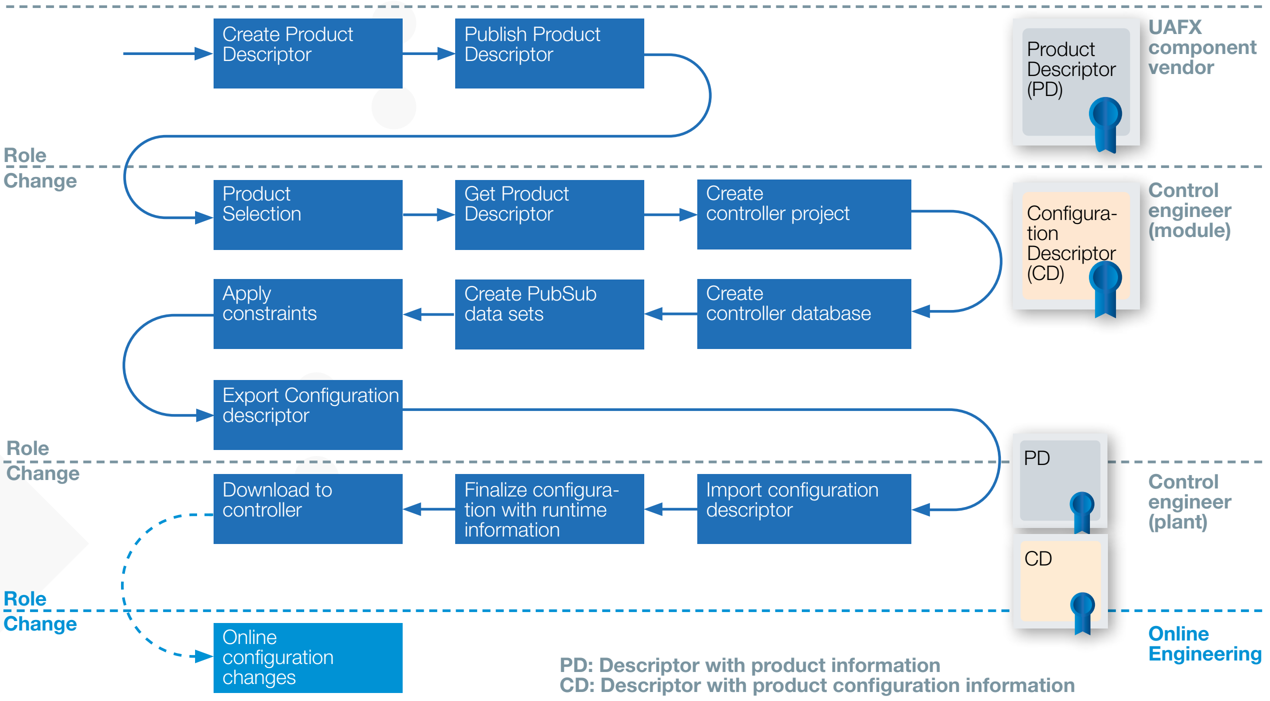

Create Product Descriptor
Publish Product Descriptor
Product Descriptor (PD)
UAFX component vendor
Role Change
Product Selection
Get Product Descriptor
Create controller project
Configuration Descriptor (CD)
Control engineer (module)
Apply constraints
Create PubSub data sets
Create controller database
Export Configuration descriptor
Role Change
Download to controller
Finalize configuration with runtime information
Import configuration descriptor
PD
CD
Control engineer (plant)
Role Change
Online configuration changes
Online Engineering
PD: Descriptor with product information
CD: Descriptor with product configuration information

Figure 15: Overview of the workflow steps for offline engineering descriptor(s) usage

Workflow Examples

This section shows two examples how descriptors can be used in an offline engineering environment and how the descriptors represent one or more ACs. In the two examples below, a Line Controller (LC) sets up and provides the overall control to 3 subordinate controllers (PLC/DCS) (C1, C2, C3) in an UAFX 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]):

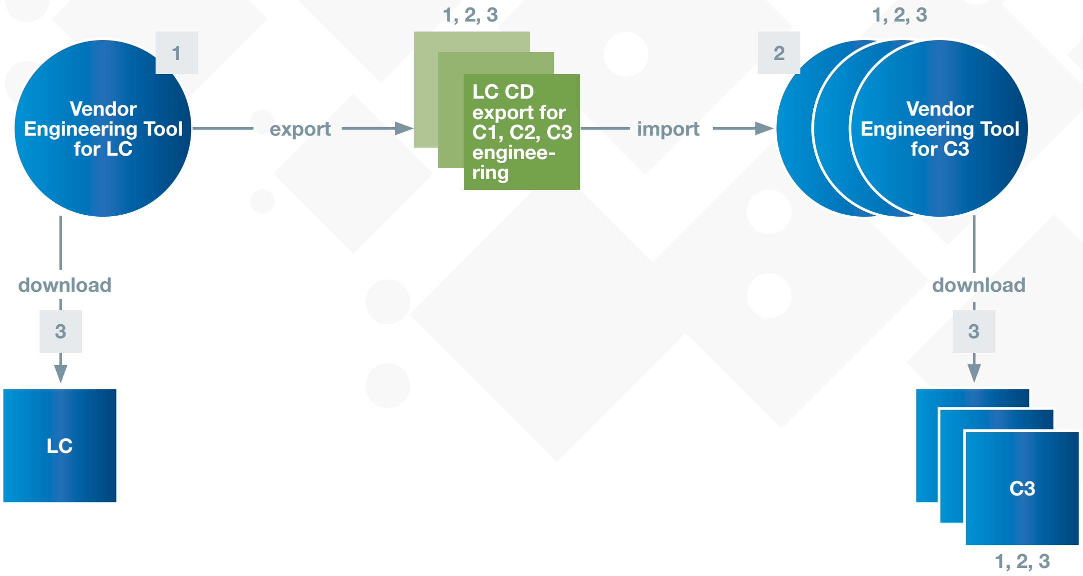
System with a Line Controller and 3 subordinate Controllers without TSN

In the offline engineering phase, the engineering tool for the LC is used to create [1] the following Configuration Descriptors (CDs):

- → LC CD for C1 to be imported into the engineering tool of C1
- → LC CD for C2 to be imported into the engineering tool of C2
- → LC CD for C3 to be imported into the engineering tool of C3

Each CD contains the information necessary to create and configure the communication relationships between LC and CX (X=1, 2, 3) (see Figure 16). In addition to the configuration information, the CD contains a digital signature from the author (in this case the development engineer of the system integrator).

When the CD is imported [2] into the engineering tool of one of the C1, C2, C3 controllers, the control engineer checks the validity of the signature and browses the information model to find the PubSub dataset information.


1
Vendor Engineering Tool for LC
export
1, 2, 3
LC CD export for C1, C2, C3 enginee-ring
import
2
1, 2, 3
Vendor Engineering Tool for C3
download
3
LC
download
3
C3
1, 2, 3

C1, C2, C3: Controller (e.g. PLC, DCS) LC: Line Controller

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

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

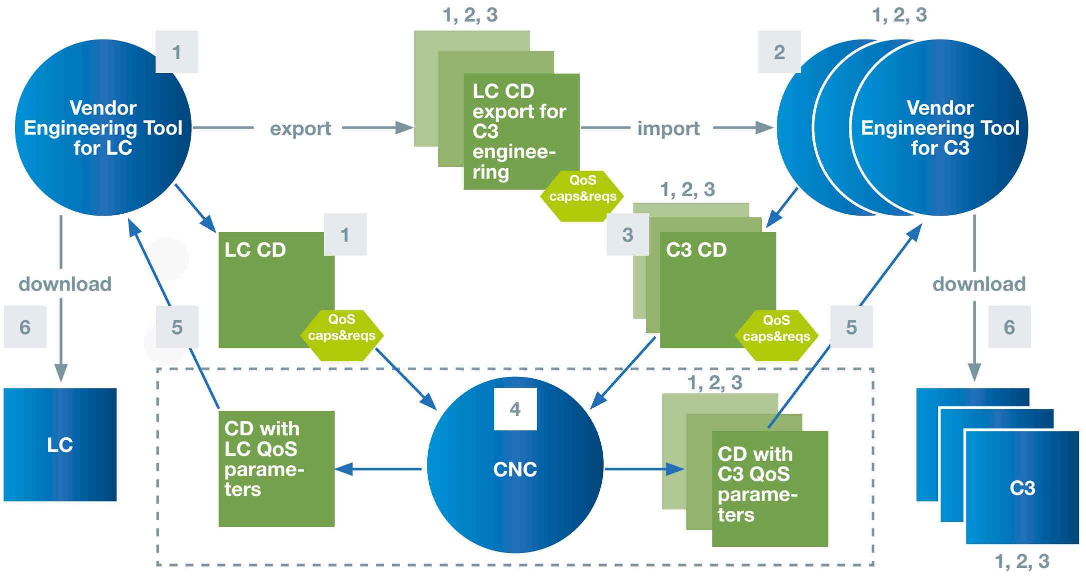
System with a Line Controller and 3 subordinate Controllers with TSN

In the offline engineering phase, the engineering tool for the LC is used to create [1] the following Configuration Descriptors (CDs):

- → LC CD for C1 to be imported into the engineering tool of C1
- → LC CD for C2 to be imported into the engineering tool of C2
- → LC CD for C3 to be imported into the engineering tool of C3

Each CD contains the information necessary to configure the communication relationships between LC and CX (X=1, 2, 3) (see Figure 17). In addition to the configuration information, the CD contains a digital signature from the author (in this case the development engineer of the system integrator).

The CD 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 (C1, C2 and C3). The QoS capability is part of the Product Descriptor (contained also in the CD), while the QoS requirements are part of the Configuration Descriptor.


1
Vendor Engineering Tool for LC
export
1, 2, 3
LC CD export for C3 engineering
QoS caps&reqs
import
2
1, 2, 3
Vendor Engineering Tool for C3
1
LC CD
QoS caps&reqs
1, 2, 3
3
C3 CD
QoS caps&reqs
download
6
LC
5
5
download
6
4
CNC
CD with LC QoS parameters
1, 2, 3
CD with C3 QoS parameters
C3
1, 2, 3

C1, C2, C3: Controller (e.g. PLC, DCS), caps&reqs: Capabilities & Requirements, LC: Line Controller, CNC: Central Network Configuration

Figure 17: Example: A system with a Line Controller configuring 3 subordinate controllers with TSN

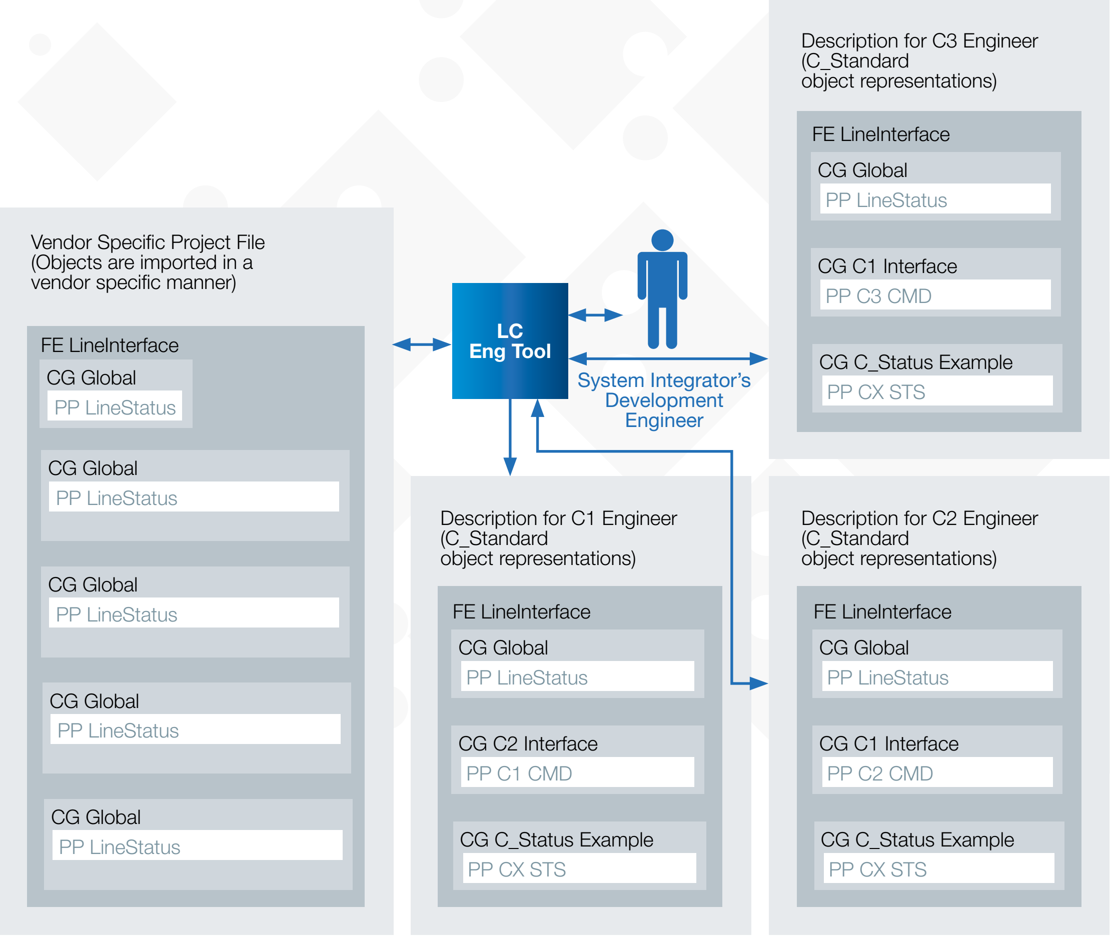
When the CD is imported [2] into the engineering tool of one of the controllers, the control engineer checks the validity of the signature and browses the information model to find the PubSub dataset information. This enables the control engineer to set up the corresponding PubSub and Connection objects in the controller.

The QoS capabilities and requirements of all controllers (LC, C1, C2 and C3) are made available to the 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 – one for each controller CX. These descriptors are imported again [5] in the CX and LC engineering tools.

Once the control engineers have completed the CX project and the controller hardware (e.g. PLC, DCS) is connected, the configuration can be deployed, e.g. [6] from the CX engineering tool.

Remark: The workflows described above are only examples, also other workflows are supported.



C1, C2, C3: 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 possible 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 fail-safe state or delivering fail-safe values. OPC UA Safety is application-independent and does not pose requirements on the structure and length of the application data.

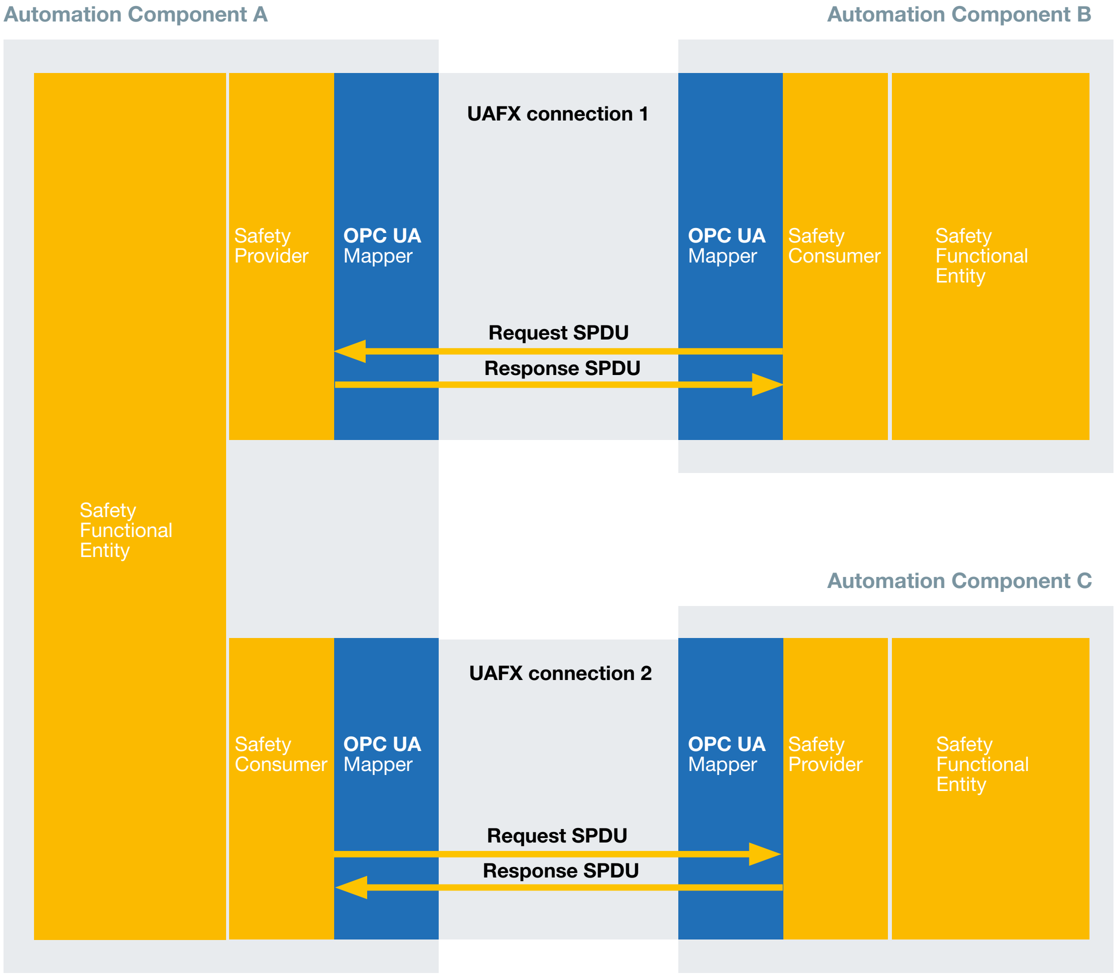


Figure 19: Safety connections between Automation Components

Safety for Field-Level Communications

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

This principle delimits the assessment effort to the safe transmission functions, such that underlying UAFX 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 must be developed in a safety-related way.

The Safety Application is connected directly with the SafetyProvider and SafetyConsumer, which exchange data by means of the safety protocol (see Figure 20). The OPC UA Mapper is the interface between the safety layer and the underlying communication channel.

The simplest type of safety communication is unidirectional from a logical safety point of view, where a Safety Application on one AC (A) sends data to a Safety Application on another AC (B). In terms of the underlying safety protocol, this is realized as a bidirectional communication where the SafetyConsumer initiates the communication to the SafetyProvider by sending a RequestSPDU. The SafetyProvider mirrors the received ID and counters, adds the requested safety data and secures all data via a checksum before responding with a Response SPDU.

Since the SafetyConsumer always initiates communication with the SafetyProvider, it can measure possible timeouts without the need of a safely synchronized clocks across the network. One AC can be SafetyConsumer and SafetyProvider at the same time. Even more so, several instances of SafetyConsumers and/or SafetyProviders can be present on an AC. 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

The SafetyProvider's state machine is very simple. It waits for a request, and if a request is received the corresponding response is sent out to the SafetyConsumer. All safety checks are done on the SafetyConsumer's side.

SafetyConsumer

The SafetyConsumer initiates the safe data exchange, waits for the SafetyProvider's response, and checks for potential communication errors (integrity, timeliness, authenticity, according to IEC 61784-3). Thereafter, the SafeData is provided to the Safety Application inside the AC. If a communication error occurs, fail-safe substitute values are provided to the Safety Application instead, and an error is indicated.

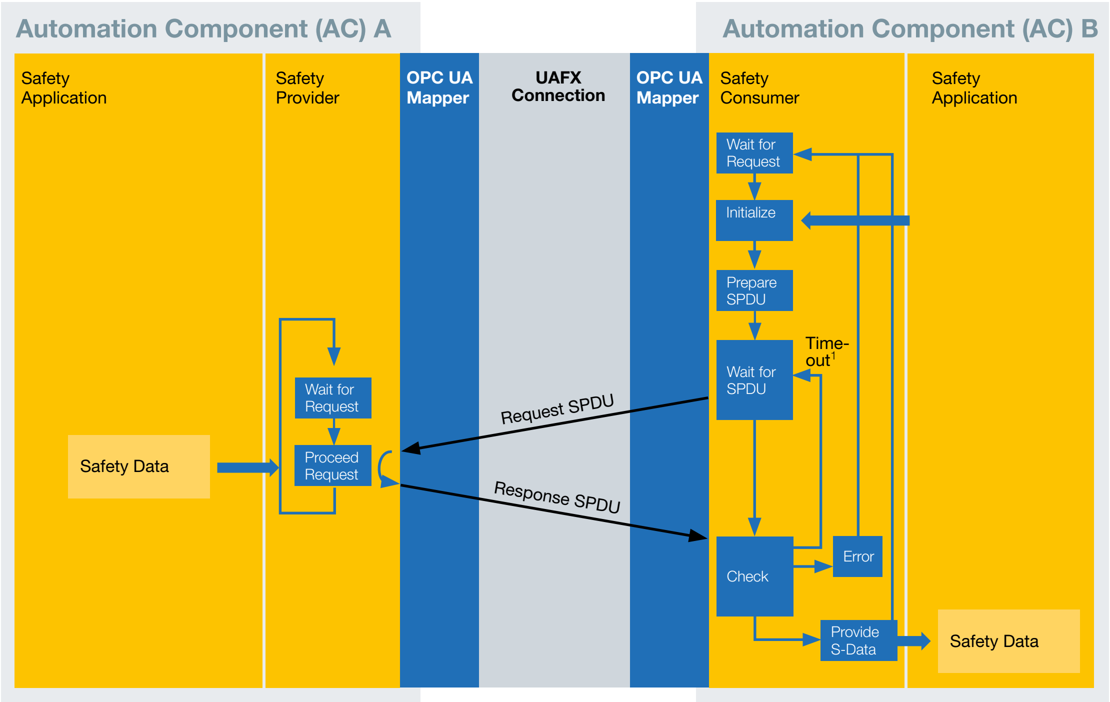


Figure 20: Simplified View of SafetyProvider and SafetyConsumer State Machines

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

Security

Security for UAFX Connections

Every UAFX connection is authenticated and optionally encrypted by standard OPC UA security mechanisms specified for the Client/Server and PubSub communication. The connection establishment process is secured by OPC UA Secure Session Conversation 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 CM creates a secure session with the CM credentials and is authorized to execute methods required to establish connections. The CM maintains the connection via this secure session up to the operational state of the connection.

PubSub connections are secured with keys exchanged with the Security Key Service (SKS). As part of the connection establishment process, the CM creates a security group for a PubSub connection and configures the SKS with information about the PubSub participants. The SKS then creates or uses a pre-configured secure connection to the PubSub participants and pushes keys for the PubSub connection. The SKS will periodically update the PubSub keys to ensure the connection remains secure.

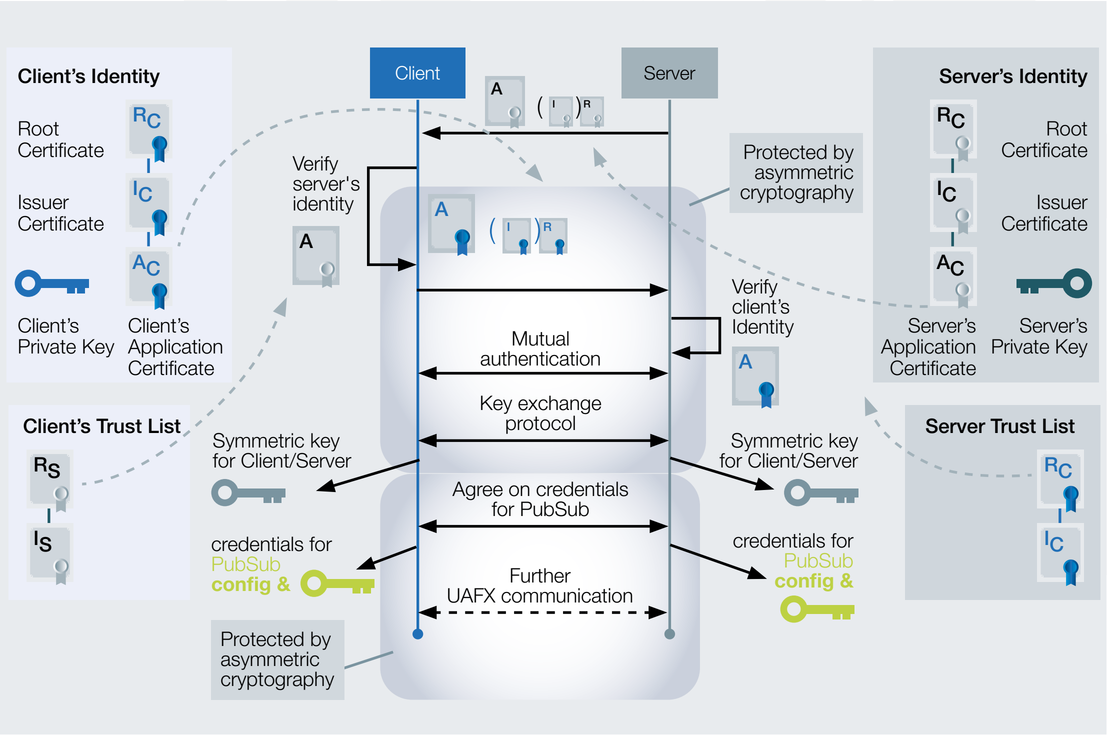

Client's Identity
Root Certificate
RC
Issuer Certificate
IC
AC
Client's Private Key
Client's Application Certificate
Client's Trust List
RS
IS
Client
Server
A
(I)R
Verify server's identity
A
Protected by asymmetric cryptography
Server's Identity
RC
Root Certificate
IC
Issuer Certificate
AC
Server's Application Certificate
Server's Private Key
Verify client's Identity
A
Mutual authentication
Key exchange protocol
Symmetric key for Client/Server
Symmetric key for Client/Server
Server Trust List
RC
IC
Agree on credentials for PubSub
credentials for PubSub config &
credentials for PubSub config &
Further UAFX communication
Protected by asymmetric cryptography

Figure 21: Mutual authentication plus obtaining credentials for PubSub

Transport

Architecture

The transport architecture of an OPC UA Field eXchange device allows for interoperability between controllers and devices using periodic communications. PubSub communication is required in all devices, which share a common base set of features for this periodic communication. Additional features and capabilities build on top of each other with compatible subsets of features, to progressively support more advanced capabilities, including TSN functionality (see Figure 22).

Common Capabilities

All controllers and devices are interoperable across the layer 3 networks (IP networks) typically deployed across an entire plant made of many machines, skids, and cells. All controllers and devices support OPC UA PubSub using UDP UADP and are recommended to implement remote management features if an embedded bridge has been integrated.

Managed Bridges

Managed bridges are Virtual Local Area Network (VLAN) and Quality of Service (QoS) aware. They also offer support for topology detection services and implement IT-standard management protocols. Implementation of management features is mandatory in bridges (see Base Bridge Component Facet below). This facilitates that network monitoring and management tools can be developed able to operate most UAFX controllers and devices within a single system view, irrespective of whether or not they have implemented TSN features. However, OPC UA FX may be deployed also on devices with embedded unmanaged bridges (i.e. that are not compliant with any of the UAFX Bridge Component Facets described in Table 1).

Time-Sensitive Networking

TSN provides mechanisms to realize networks with zero congestion packet loss and bounded network latency required by some automation applications. It is designed for layer 2 networks typically seen within a single skid, cell or machine. Connections using TSN deliver the greatest application determinism but operation through a layer 3 switch or router requires enhancements such as the use of Detnet (in development by IETF).

The optional capabilities for TSN-enabled embedded bridges are defined in the Advanced & Full Bridge Component Facets.

The principle of 'graceful degradation' of operation is applied whereby any two controllers or controller and device are interoperable at the highest level of Quality of Service available through the network path through which they are connected.

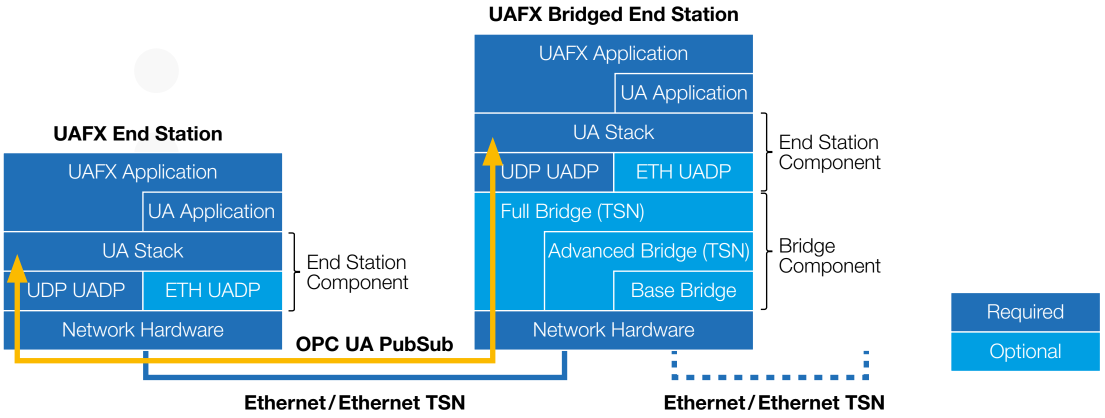


Figure 22: UAFX Transport Architecture

The Field Level Communications Initiative is committed to supporting the IEC/IEEE 60802 TSN Profile for Industrial Automation, when it is published. It is expected that all Industrial Ethernet variants and IT devices operating in an industrial network using TSN will align with this specification, with the network management tool allocating the necessary network resources to each application. 'IEC/IEEE 60802 Configuration Domains' require all bridges (either embedded in a controller, a device, or in an infrastructure switch) to comply with IEC/IEEE 60802. If a controller or device implements the TSN support and it also implements an embedded bridge, then it must implement either the Advanced Bridge Component Facet or the Full Bridge Component Facet and that bridge must be IEC/IEEE 60802-compliant. The mechanisms to connect multiple 'IEC/IEEE 60802 Configuration Domains' in a single network and to extend them through network routers have yet to be standardized. Once a network is correctly configured with enough resources allocated, TSN ensures both zero packet loss due to network congestion and bounded latency of delivery to the target.

Frame Preemption

One of the key capabilities defined by the TSN Task Group within IEEE is frame preemption, according to IEEE 802.3 and IEEE 802.1Q, which allows a large, lower-priority packet to be broken into multiple fragments and higher priority packets to be transmitted in between these fragments. This substantially reduces jitter and latency in a network by reducing the amount of time that a higher-priority packet must wait in the queue.

Feature	Base	Advanced	Full
C-VLAN Component Support	✓	✓	✓
Frame Filtering	✓	✓	✓
Strict Priority	✓	✓	✓
C-VLAN 8 (8 user-defined VLAN IDs)	✓	✓	✓
Queue 4 (support of 4 queues)	✓	✓	✓
Remote Management (NETCONF Protocol for configuration of YANG models)	✓	✓	✓
Regenerating Priority	✓	✓	✓
Buffer frames on egress port for a period of: • 500 µs at 100 Mbps • 200 µs at 1 Gbps	O	✓	✓
Queue 8 (support of 8 queues)	O	✓	✓
TE-MSTID		✓	✓
Preemption Minimum Non-Final Fragment Size 64		✓	✓
gPTP Time Synchronization		✓	✓
Enhancements For Scheduled Traffic (EST = Qbv/TAS)		O	✓
Per-Stream Filtering And Policing		O	O
Scheduled Traffic Gate Control List entries: • 16 at <= 100 Mbps • 64 at >= 1 Gbps			✓
Scheduled Traffic Cycle Times: • 800 µs and • 1000 µs (basic cycles to accommodate multiples of 31.25 µs and 25 µs)			✓
Scheduled Traffic Gate Control List: • 128 schedule entries			O

Table 1: Features of UAFX Bridge Component Facets

Scheduled Traffic

Another key capability defined by the TSN Task Group within IEEE is the scheduling of time-critical communication on the network. Bridges that support this functionality must implement IEEE 802.1Q Enhancements for Scheduled Traffic (EST) on the network. EST defines cyclic transmission windows for each traffic class. Each stream therefore will only be transmitted during one of the transmission windows of its traffic class.

Bridge Component Facets

Three optional UAFX Bridge Component Facets have been defined, which take IEC/IEEE 60802 profiles, make some optional capabilities mandatory and define UAFX-specific quantities above and beyond those defined in the underlying specification. Further, the required data rate of 1 Gbps is made optional for UAFX controllers and devices.

IEC/IEEE 60802 TSN Profile for Industrial Automation[2]

It is anticipated that IEC/IEEE 60802 will support two conformance classes, ccA and ccB[3]. The features of these conformance classes will not be finalized until its publication in 2024 or 2025. A bridge vendor may choose to implement compliance with one profile and add features necessary to comply with one or more UAFX Bridge Component Facets. However, it is expected that ccA will be more closely aligned with both the Advanced Bridge Component Facet and the Full Bridge Component Facet. IEC/IEEE 60802 conformance tests are being jointly developed by TI-ACC (TSN Industrial Automation Conformance Collaboration) which will be used as the base test for both the Advanced Bridge Component Facet and the Full Bridge Component Facet[4].

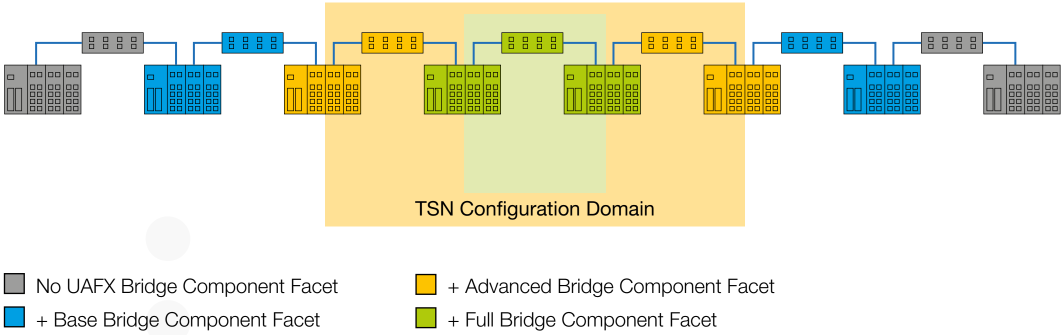

TSN Configuration Domain
▨ No UAFX Bridge Component Facet
🟦 + Base Bridge Component Facet
🟨 + Advanced Bridge Component Facet
🟩 + Full Bridge Component Facet

Figure 23: Model illustrating graceful degradation of QoS (Onion Ring Model)

[2] https://1.ieee802.org/tsn/iec-ieee-60802/

[3] https://www.ieee802.org/1/files/public/docs2021/60802-Steindl-ccMatrix-0321-v02.pdf

[4] https://opcfoundation.org/news/press-releases/single-common-conformance-test-plan-to-be-available-for-the-iec-ieee-60802-tsn-profile-for-industrial-automation/

Graceful Degradation of QoS

The UAFX Bridge Component Facets are designed to allow graceful degradation of QoS, such that a component implementing the Full Bridge Component Facet can be configured to operate like one implementing the Advanced Bridge Component Facet, or one implementing the Base Bridge Component Facet (and if unconfigured, as one not implementing any Bridge Component Facet) to improve interoperability between UAFX controllers and devices.

When designing the network topology, users need to consider application demands when positioning controllers (and in future, devices) in a network.

If an application demands the features associated with the Full Bridge Component Facet (see Table 1 above) then those controllers must be clustered together in the network topology and any standalone switches must be validated against those bridge requirements (see Figure 23).

Similarly, if an application requires the features demanded in the Advanced Bridge Component Facet (Table 1) then all controllers must support it and all standalone bridges must be validated against the features of the Advanced Bridge Component Facet. They should be configured within a single TSN configuration domain such that the CNC configures all bridges (controller or standalone switch) within that domain to deliver application needs.

The CNC is unaware of application needs of controllers implementing the Base Bridge Component Facet or those that do not implement any Bridge Component Facet. Therefore, the user must ensure that configuration of the controllers in the TSN configuration domain does not interfere with the application needs of these non-TSN capable controllers.

Placing a controller, device or standalone switch either with no Bridge Component Facet implemented, or with the Base Bridge Component Facet between two TSN-capable controllers, will result in two TSN configuration domains (see Figure 24) and TSN-level QoS cannot be guaranteed between them, as operation between the TSN configuration domains will gracefully degrade to match the capability of the interconnecting controller or switch. Similarly, placing a controller or standalone switch that is not compliant to the remote configuration mechanisms defined in the Base Bridge Component Facet between components supporting that facet may result in QoS not being respected, LLDP being not forwarded across all ports and tools not being able to monitor the network accurately.

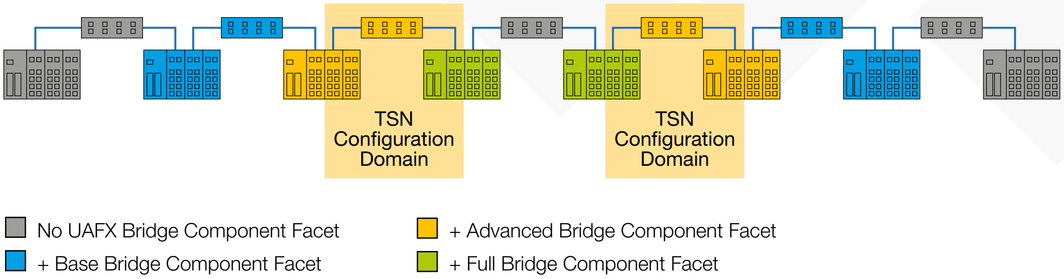

TSN Configuration Domain
TSN Configuration Domain
▨ No UAFX Bridge Component Facet
🟦 + Base Bridge Component Facet
🟨 + Advanced Bridge Component Facet
🟩 + Full Bridge Component Facet

Figure 24: Model illustrating graceful degradation of QoS

Ethernet – Advanced Physical Layer

ethernet-apl™
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 Mbps, full-duplex. It is an 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 is 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.

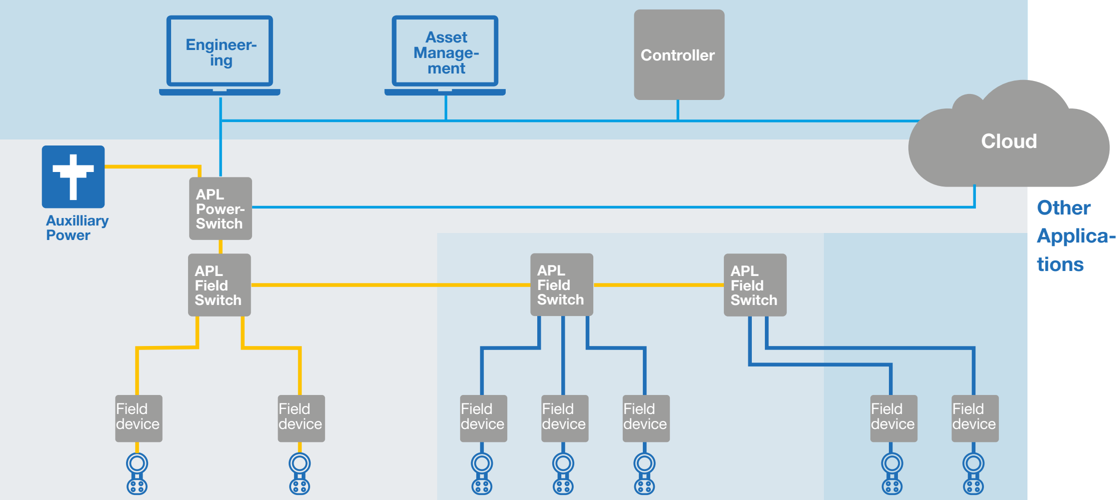


Figure 25: Example topology for long cable reach

[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 mechanisms that can provide various priorities to different devices or data flows and 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 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 IT applications to share the same physical network infrastructure without hampering the 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 UAFX 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 for UAFX systems:

- → 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 Configuration 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/skids 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.

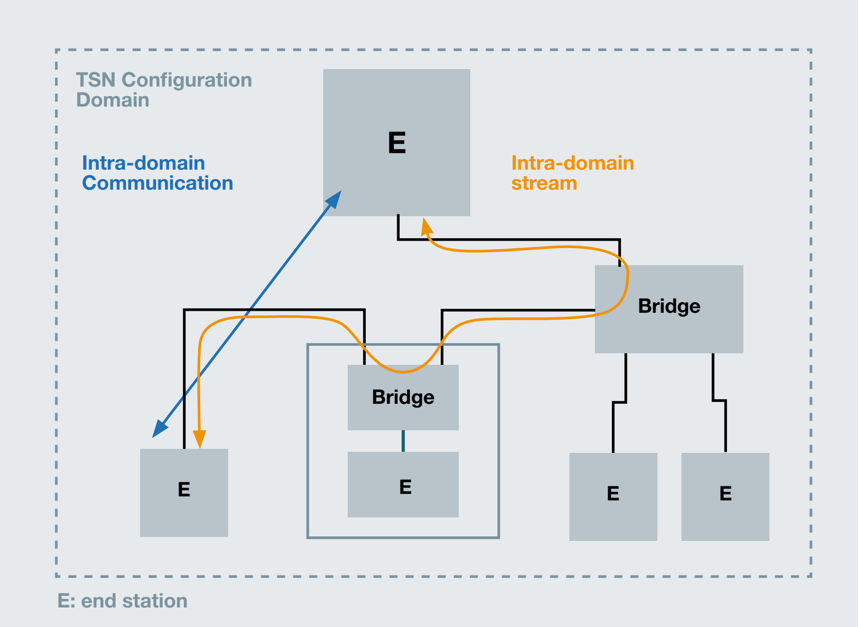


Figure 26: Intra-domain communication

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

The selected stream reservation mechanism enables an industrial control application to reserve network resources for the selected TSN QoS mechanisms within the given TSN Configuration 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 Configuration Domains 1, 2, and 3.

As an alternative to 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 FX specifications.

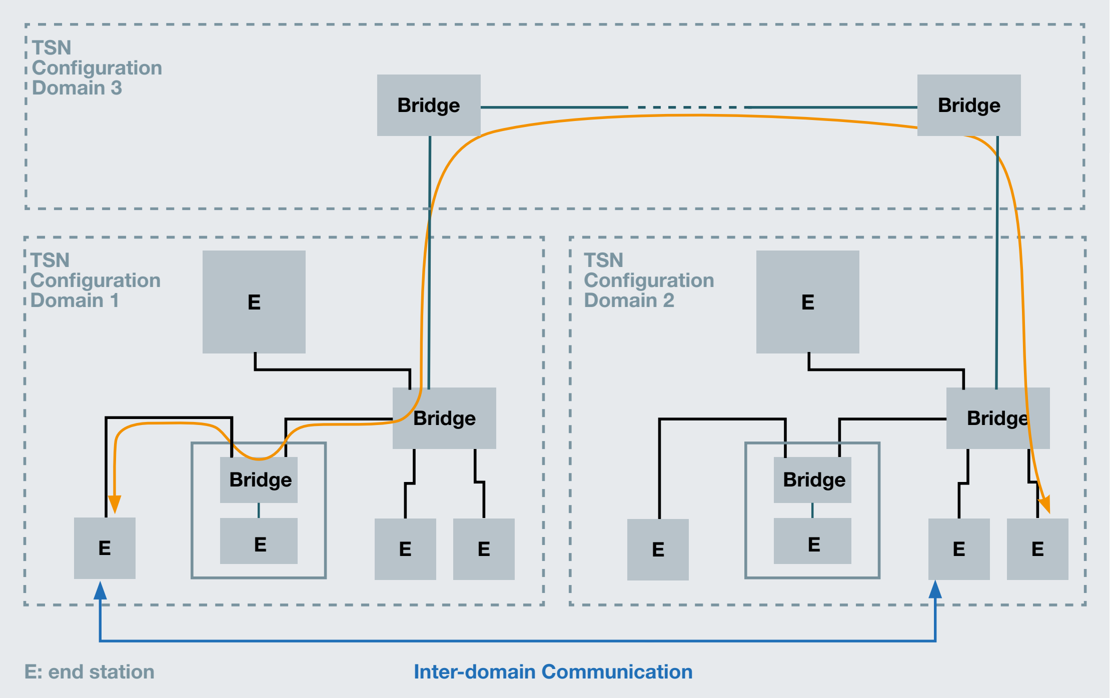

TSN Configuration Domain 3
Bridge
Bridge
TSN Configuration Domain 1
E
Bridge
Bridge
E
E
E
E
TSN Configuration Domain 2
E
Bridge
Bridge
E
E
E
E
E: end station
Inter-domain Communication

Figure 27: Inter-domain communication

Network Management

UAFX Network Management is based on the standardized NETCONF management protocol. The actual configuration parameters are modelled in the YANG data modeling language.

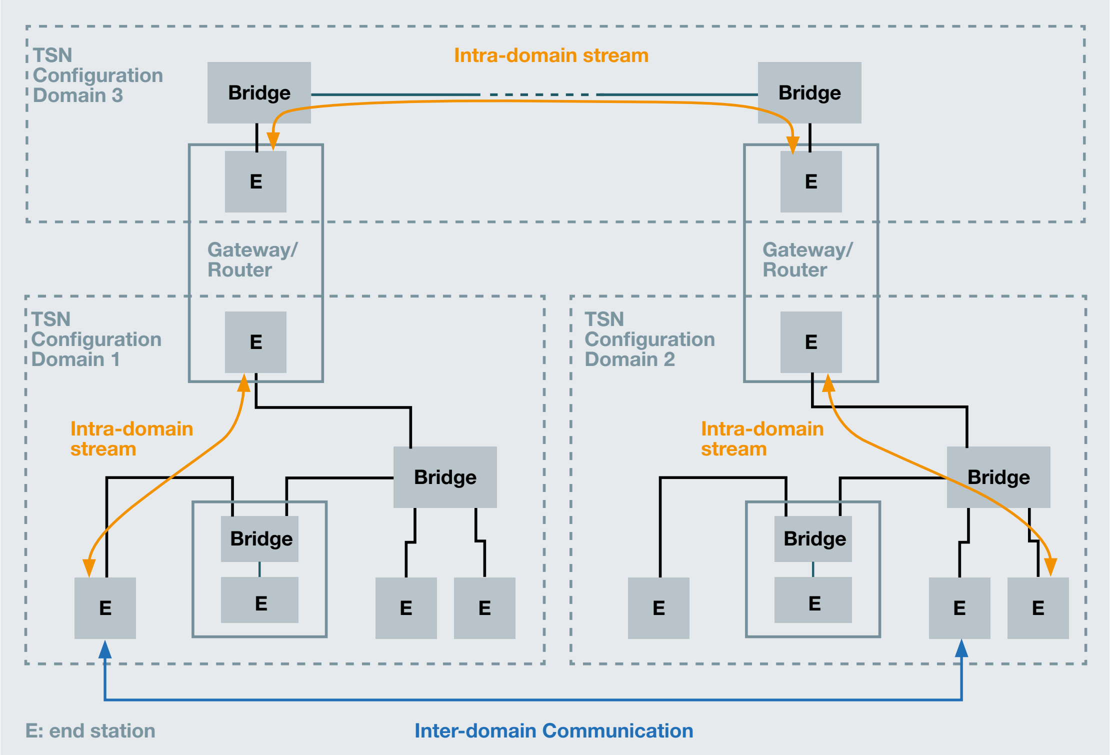


Figure 28: Connection of Domains using application level gateways or DetNet routers

Communication relation	Description / Example
C2D Intra-domain communication	This is probably the most common relationship, where a controller communicates with its peripheral (I/Os, drives, valves, …)
C2C Intra-domain communication	Communication between multiple controllers in the same TSN Configuration 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 Configuration Domain
C2C Inter-domain communication	Interconnection of machines/skids
D2D Inter-domain communication	Synchronization between motions drives in different TSN Configuration Domains

Table 3: Examples of communication relationships

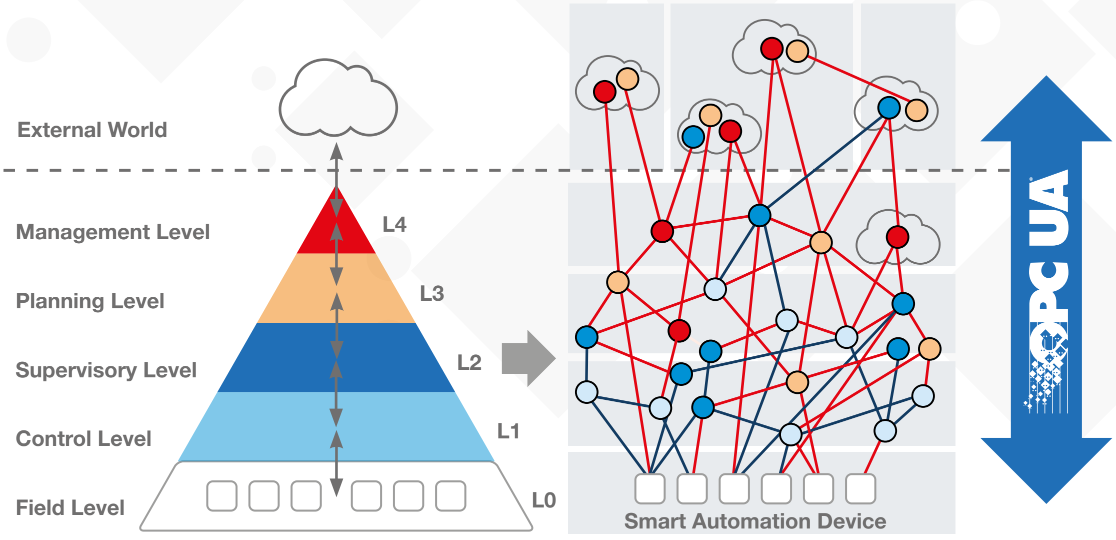
Summary and Outlook

This technical paper describes how the Field Level Communications Initiative extends the OPC UA framework to facilitate cross-vendor interoperability between controllers and devices 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 OPC UA FX 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 UAFX is facilitated. Furthermore, test specifications, automated testing and test tools are being developed. They include testing offline configuration exchange as well as online communication. This will provide high-grade cross-vendor interoperability between Automation Components.

With the extensions specified by the Field Level Communications Initiative, OPC UA in combination with Ethernet-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).



Source: VDI (2013), MDPI (2019)

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
DetNet	Deterministic Networking
DSCP	Differentiated Services Code Point
ERP	Enterprise Resource Planning
EST	Enhancements for Scheduled Traffic
FE	Functional Entity
gPTP	Generalized Precision Time Protocol
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
IP	Internet Protocol
IT	Information Technology
L2	Layer 2
L3	Layer 3
LLDP	Link Layer Discovery Protocol
MES	Manufacturing Execution System
MQTT	Message Queuing Telemetry Transport
OE	Offline Engineering
OPC	Open Platform Communication
OPC UA	OPC Unified Architecture
OPCF	OPC Foundation
OT	Operational Technology
PAC	Programmable Automation Controller
PCP	Priority Code Point
PD	Product Descriptor
PLC	Programmable Logic Controller
QoS	Quality of Service
RAN	Radio Access Network
SCADA	Supervisory Control and Data Acquisition
SPE	Single-Pair Ethernet
TAS	Time-Aware Shaper
TCP	Transmission Control Protocol
TE-MSTID	Traffic Engineering Multiple Spanning Tree Instance Identifier
TIACC	TSN Industrial Automation Conformance Collaboration
TSN	Time-Sensitive Networking
UADP	Unified Architecture Datagram Packet
UAFX	Unified Architecture for Field eXchange
UDP	User Datagram Protocol
VLAN	Virtual Local Area Network
Wi-Fi	Wireless Fidelity
WLAN	Wireless Local Area Network

HEADQUARTERS / USA
OPC Foundation
16101 N. 82nd Street
Suite 3B
Scottsdale, AZ 85260-1868
Phone: (1) 480 483-6644
office@opcfoundation.org

OPC EUROPE
Huelshorstweg 30
33415 Verl
Germany
opceurope@opcfoundation.org

OPC JAPAN
c/o Microsoft Japan Co., Ltd
2-16-3 Konan Minato-ku, Tokyo
1080075 Japan
opcjapan@opcfoundation.org

OPC KOREA
c/o KETI
22, Daewangpangyo-ro 712,
Bundang-gu, Seongnam-si, Gyeonggi-do
13488 South Korea
opckorea@opcfoundation.org

OPC CHINA
B-8, Zizhuyuan Road 116,
Jiahao International Center, Haidian District,
Beijing, P.R.C
P.R.China
opcchina@opcfoundation.org

V3

www.opcfoundation.org