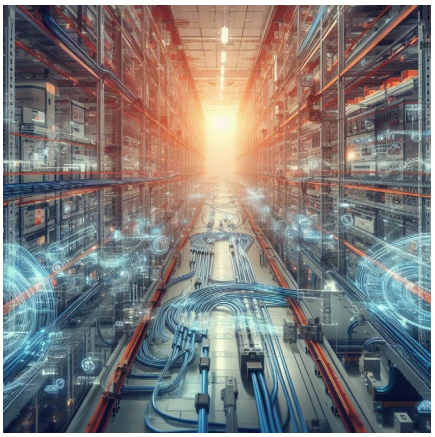
CHAPTER 2
INDUSTRIAL COMMUNICATION AND NETWORKS

Lecture material for TTK 4175 Instrumentation Systems and Safety at the Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU).

Author: Professor Mary Ann Lundteigen, Department of Engineering Cybernetics



The essence of industrial networks?

Illustration generated by Microsoft Copilot (powered by OpenAI), July 2025

The content is updated regularly to improve precision and ensure relevance, which is reflected in the revision number. Please reach out to mary.a.lundteigen@ntnu.no if you have comments or suggestions for improvement.

Rev: 2.0/2026

Revision tracking (most recent)

Rev	Date	Modifications
2.0/26	01.07.2026	Updated after spring semester

Contents

2 Industrial communication and networks

Industrial communication refers to the exchange of data among systems involved in the control and safety operations at industrial facilities. Most of the focus is on real-time communication technologies at the lower levels of the OT network architecture, i.e., levels 0-2 in the Purdue reference architecture, including controllers, field devices, and user interfaces. At these levels, we focus on two categories of communication technologies:

- Analog (current or voltage) signal and binary (voltage) signal transmission
- Digital message transmission

We will learn that real-time digital message transmission is split into two main groups: Fieldbus and Industrial Ethernet (IE). Their distinctions in characteristics are presented and supported by examples of open standardized protocols, including profiles for message exchange between safety-critical systems and devices.

However, we will also address communication technologies used at higher levels (levels 2-3/4) to exchange data for monitoring, analysis, and optimization, with a focus on Open Platform Communications (OPC) Unified Architecture (UA). OPC UA is used to exchange large amounts of data with less stringent real-time requirements, for example, for monitoring, trending, and notification of alarms and events. We will learn that OPC UA is not just a communication technology; it also includes the organization, storage, and mapping of data on servers.

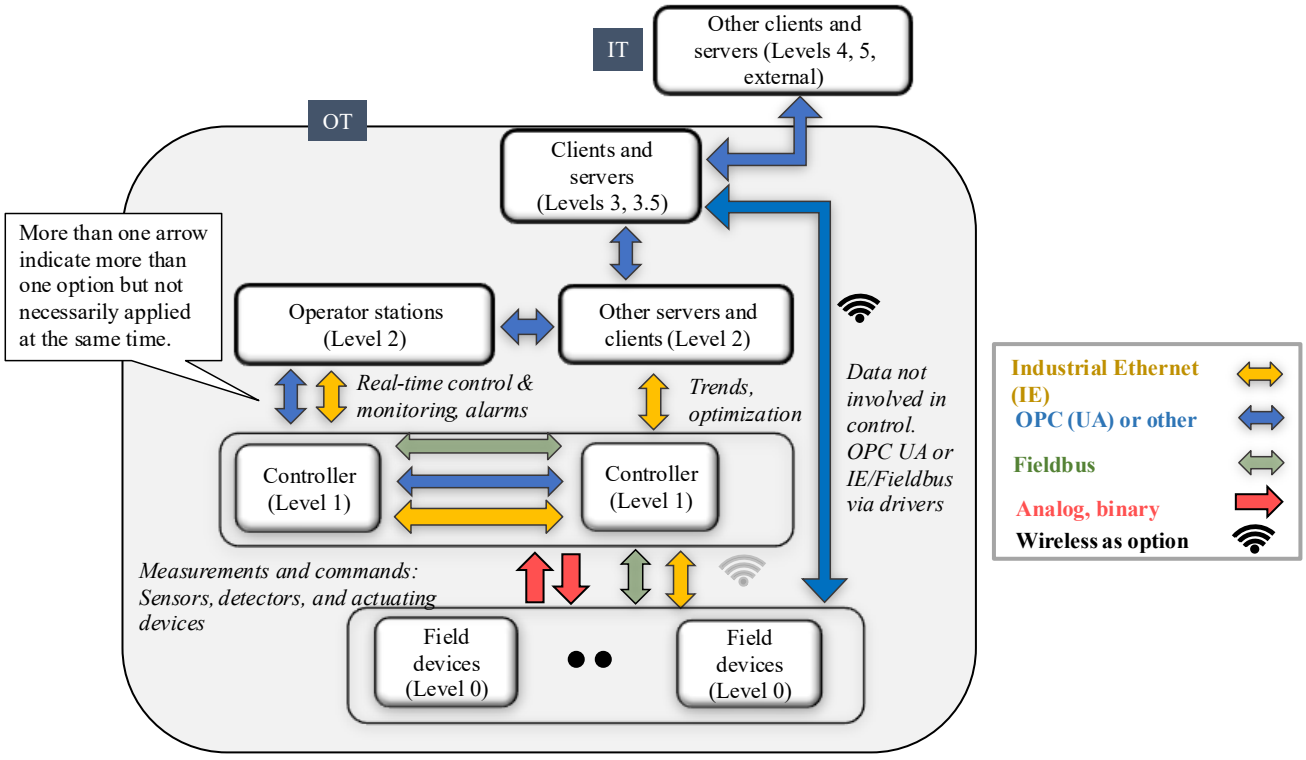


Fig. 1. Examples of the application of industrial communication

OPC UA is one of the technologies central to Industry 4.0 for enabling data exchange between OT and IT. This exchange is important as data clouds and software used for advanced analysis, including artificial intelligence, are placed in IT rather than OT networks. Industry 4.0 began as a German initiative but is now the term used to describe the global digital transformation of manufacturing and industrial processes. While successor concepts such as Industry 5.0 and Industry 6.0 are being introduced, focusing on human-centric solutions and autonomy, respectively, many industries are still focused on implementing Industry 4.0 as a necessary foundation.

Overall, Fig. 1 might serve as a useful overview of the communication technologies covered and hopefully be useful when reading the remainder of this chapter. However, the chapter concludes with some emerging trends and technologies not visualized in the figure. These efforts support a more standardized and seamless exchange of data, which is essential in modern industrial environments.

2.1 Abbreviations

APL	Advanced physical layer
AAS	Asset administration shell
AML	Automation markup language
CDD	Common data dictionary
DC	Direct current
DCS	Distributed control system
DMZ	Demilitarized zone
DP	Distributed peripherals
ESD	Emergency shutdown system
EWS	Engineering workstation
F&G	Fire & Gas (detection system)
FW	Firewall
Gbps	Gigabits per second
HART	Highway Addressable Remote Transducer
HIPPS	High integrity pressure protection (HIPPS)
HMI	Human-machine interface
IMS	Information management system
IEC	International Electrotechnical Commission
I/O	Input/output (card, I, O or combined)
ISO	International Organization for Standardization
IT	Information technology
kbps	Kilo bits per second
Mbps	Megabits per second
NOA	Namur Open Architecture
NORSOK	NORsk SOkkels Konkurranseposisjon
OPA	Open platform automation system
OPC UA	Open process communication - Unified architecture
OPC UA TSN	OPC UA Time-sensitive networking
PA	Process Automation
PCS	Process control system
PDU	Protocol data unit
PLC	Programmable electronic controller
PSD	Process shutdown system
OT	Operational Technology
SAS	Safety and automation system
SIS	Safety instrumented system

2.2 Analog and binary signal transmission

Analog signal transmission can be provided as a continuous current or voltage signal between a field device and a controller. While it may seem old-fashioned to use such signal transmission today, given the option of digital messaging, it remains among the most commonly used methods in many plants, particularly for industrial processes (as opposed to manufacturing).

Analog signal transmission from a field instrument, such as temperature, pressure, flow, or level, to a controller is often provided as a current value expressed in mA. The controller reads the measurement via an analog input card, which converts the reading into a digital format that the central processing unit (CPU) can interpret. Signal

transmission *from* the controller to a field device, such as an actuator, may be a current or a voltage. While many valve actuators open and close with the assistance of pneumatics (pressurized air), a conversion is made from analog to applied pressure (signal) via a current (I)- to-pressure (P) converter.

Analog signal transmission is often chosen for its simplicity (for installation and integration), transmission without nearly any delay, and long-distance capability. It is also regarded as highly reliable, making it attractive for use with safety-instrumented systems.

2.2.1 Namur 4-20 mA standard for analog current transmission

For analog current signals, the most common standard applied globally is the German NAMUR standard (NE-43, 2021). As illustrated in Fig. 2, the NAMUR standard specifies:

- **Normal measurement range 4 – 20 mA (green):** Corresponds to the lowest and the highest permitted values of the measured phenomenon.
- **Saturated or avoided zone (yellow):**
  - Process saturation zone (3.8 to 4.0 mA or 20.0 to 20.5 mA): The process variable has slightly exceeded the transmitter's calibrated lower or upper limits (under-range/over-range). The measured signal is still valid, but the device may require range optimization or recalibration.
  - Transition zones (3.6 to 3.8 mA or 20.5 to 21.0 mA): These are undefined differentiation zones. A healthy NAMUR transmitter should not output values in this range; therefore, readings in this range usually indicate signal noise, line-resistance errors, or an impending hardware fault.
- **General fault states < 3.6 or >21 mA (red):**
  - At 0 mA: Typically indicates a complete cable break, total power loss, or terminal disconnection.
  - >0 mA but < 3.6 mA: Indicates the transmitter is intentionally driving the loop low to report an internal sensor failure, OR a loop supply issue is occurring. Examples include excessive loop impedance, insufficient voltage supply, or circuit degradation due to corrosion and moisture (which can prevent the system from lifting the signal above 4.0 mA.
  - > 21.0 mA: Indicates the current signal is forced above the normal measurement range. This is typically due to an electrical earthing fault (ground loop) injecting stray current, or the transmitter intentionally driving the loop high to report a critical internal hardware failure.

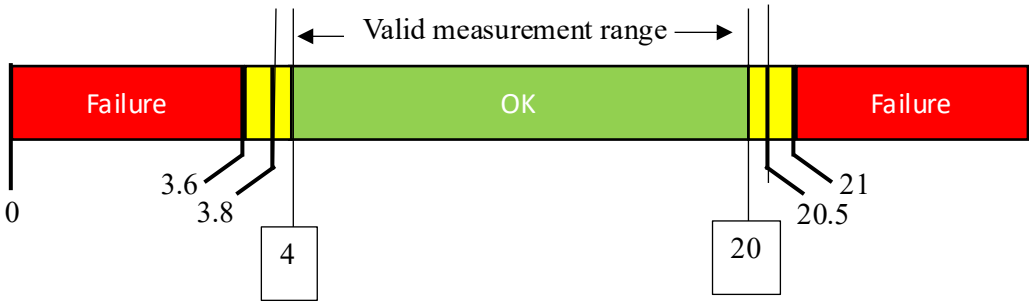


Fig. 2. Namur NE43 standard for current (analog) signal transmission (in mA)

There are two implementations for analog current-signal transmission:

- Two-wire systems (one pair): This is the most commonly used approach and the one reflected in the NAMUR 4-20 mA standard: Here, power supply and signal are transported over the same two wires (also known as loop-powered circuits)
- Four-wire (two-pair): With four wires, the power and measurement signals are separated into their own pairs of wires. This setup provides higher signal accuracy and may be used for applications requiring high precision. For most industrial applications, two wires are usually sufficient.

Fig. 3 illustrates the two-wire option, in which a field device (transmitter) transmits a measurement to an analog input card in a PLC or DCS. It is the digital input card that provides the power supply, and the transmitter changes the resistance, and thus the current, according to the measurement process value. The input card has a

resistor (arrangement) to limit the maximum current in the event of a short circuit in signal transmission. The twisting of the wires is done to make the wires more resistant to electromagnetic interference (EMC).

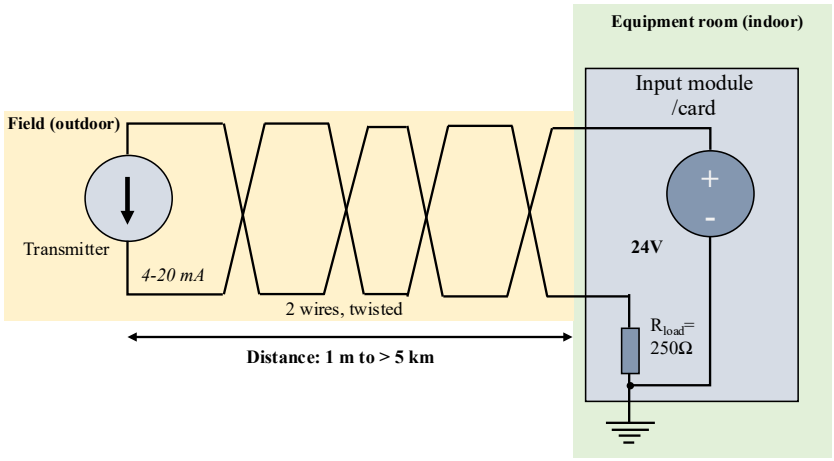


Fig. 3. Analog signal transmission circuit

Considering pressure drops in the wires, the sensor's power consumption, and the measuring resistor, the general rule of thumb suggests a maximum cable length of 5 km.

2.2.2 Analog voltage transmission

Analog current and voltage signals are also used to control the position of valves, contacts, and switches. In this case, the PLC or DCS transmits the signal via an analog output card. While analog signal transmission of current signals follows the NAMUR NE43 4-20mA standard, there is no equivalent standardized specification for voltage signals. However, it is common to use a voltage above 0 as the lower limit to distinguish values read (0V) upon cable breakage.

The ABB oil/water separation lab, part of TTK 4175, uses an analog voltage signal ranging from 2 to 10 V to control the opening of the Belimo modulating rotary actuator valves.

2.2.3 Summing up the pros and cons

To sum up, the main reasons why analog signal transmission is considered attractive are:

- Analog signal transmission is a simple technology, with a two-wire cable and no protocol to configure
- The communication is universal in that it does not need any translation, as protocols may require, if different protocols are applied in the network.
- Transmission allowed over long distances with almost no time delay.
- The simplicity makes the communication reliable and the failure modes well-defined.
- Its broad adoption worldwide has made the technology well-proven.
- The last, but not least important reason for analog transmission is the ability to provide a power supply and a measurement signal over the same cable.

However, there are also some concerns:

- The diagnostic information provided is very limited and limited to a few current or voltage values, indicating that “something is wrong” without the possibility for the field device to report more details.

We will learn later that HART is a protocol that can be superimposed on analog measurements without disturbing the measurement value. The field device must then have a HART interface in addition to the analog interface, and a handhold device or asset management system (AMS), a type of system often provided by the manufacturers for their suite of products, can collect diagnostics data and allow reconfiguring of device parameters and setups. Of course, the “write” capabilities must be properly secured.

2.2.4 On/off signals (binary) using relays

Binary communication, using on/off signals, is a simple way to activate field devices between two states, such as switches (contacts), pushbuttons, and on/off valves.

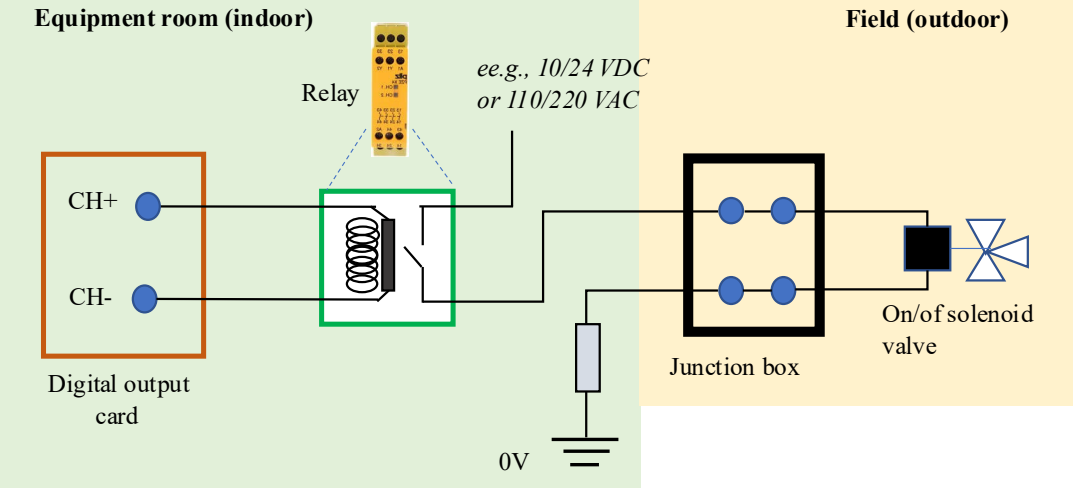


Fig. 4. On/off signal control

Fig. 4 illustrates how the binary (on/off) control positions a solenoid valve via a relay. The voltage level 0 V corresponds to "low," and the voltage level "high" can be either 10 or 24V DC or 110 or 220 V AC, depending on the type of circuit. In Fig. 4, a digital output "high" will energize the relay coil so that the contact closes. The contact completes a circuit that powers the solenoid valve's coil in the field. The solenoid valve is often used as an intermediate stop to control a larger on/off valve installed in a pipe. Energizing and de-energizing the solenoid valve coil changes the valve's position and, consequently, its on/off state. Like analog communication, it is relatively reliable and straightforward.

2.3 Digital data transmission

Digital communication is the exchange of digitally encoded information using a suitable protocol for real-time communication, other (less time-critical) communication, condition (technical health) monitoring that is not involved in control or safety functions, and the exchange of events such as alarms. Digital communication is always used between controllers at level 1 and between these controllers and the higher network levels. However, digital communication offers many more opportunities when applied to devices at level 0 and between levels 0 and 1 to integrate measurement signals with supplementary information, such as measurement quality and device health. Despite the advantages of analog communication, there is a trend, in light of Industry 4.0, towards fully digital solutions, at least for new industrial facilities. However, the lag will be many years, as the infrastructure, competencies, people's preferences, internal company guidelines, sector standards, and guidelines may not always favor a rapid transition from one way of doing things to another.

2.3.1 Network topologies

Digital communication enables several methods of transferring data between systems, not just point-to-point, as in analog and binary (on/off) signal exchange. A network topology describes the physical or logical arrangement of nodes and communication paths in a network.

2.3.1.1 Physical network topologies

There are several types of physical network topologies, and some of the most common are illustrated in Fig. 5. Here, the orange circle indicates the communication master, and the blue circles represent devices that send or receive on request ("slaves"). The networks can be configured for a single protocol or a combination of

protocols, depending on the physical realization (cable type) and the interfaces of the connected devices. Sending messages using different protocols may also require network devices, such as gateways, to interpret and perform address conversions, package analysis, and so on.

Star network: A star network has a central (advanced) hub or switch that acts as a master and controls communication with connected devices (slaves). The topology is easy to manage and supports multiple protocols if the sender and receiver understand each other. Devices can transmit simultaneously, and a new device can be added without affecting the other devices. One disadvantage is that the topology is vulnerable if the master fails.

Tree: A network that organizes communication in a hierarchical manner, integrating features of other network topologies like star, bus, point-to-point, and daisy chain.

More details about the network topologies can be found in IEC 61918 (2024) on Industrial communication networks - Installation of communication networks in industrial premises.

Ring network: A ring network is a closed loop of communicating devices. The configuration typically allows bidirectional communication, meaning that all (functioning) devices can be accessed even if a single device in the network is broken or fails. However, it is necessary to break the loop by temporarily stopping or restricting communication when adding a new device. We also need protective measures to prevent individual devices from blocking network traffic.

Bus network (also called multi-drop): A Bus network is a network topology in which devices share a single logical or physical communication channel. Multi-drop, as a related concept, refers to the physical cable that allows a master and field devices to connect to the same cable. A multi-drop and a physical bus require two defined endpoints (terminators) that allow many devices to communicate. Fig. 5 illustrates how the bus can be split into segments, named trunks, each with a set of spurs that connect devices to the bus. It is possible to choose different network topologies for the spurs, such as bus, star, or ring, depending on whether Fieldbus or Industrial Ethernet protocols are used. A fault within the trunk segments and connections hinders communication with affected devices. A redundant bus network would compensate for this problem, but each device must connect to two different spurs.

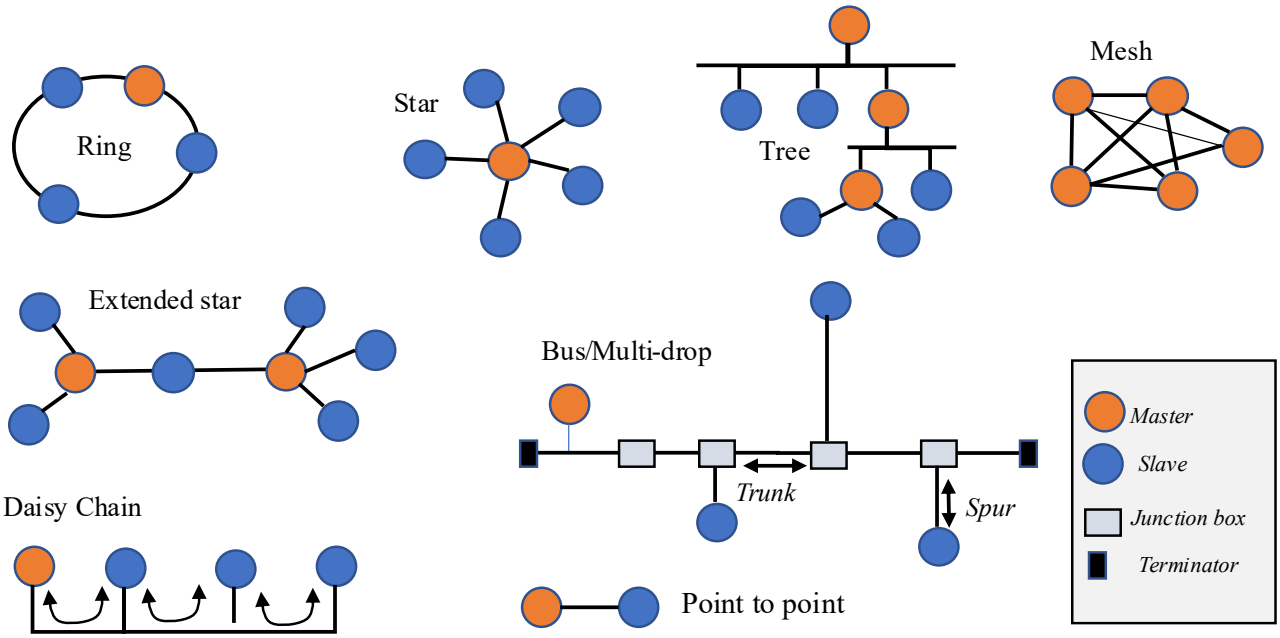


Fig. 5. Examples of network topologies

Daisy chain: A Daisy chain is a network where devices communicate sequentially in a chain. One device hands data to the next in line until the destination is reached, like in a ring network, except that it is “open loop”. Depending on the protocol, traffic can go in both directions. At first glance, it may look very similar to a bus; however, the mentioned differences include scalability and linear communication, meaning that messages must pass through each device along the line.

Point-to-point: Communication between two devices in which one acts as the master and the other as the slave. Individually wired connections between a PLC or DCS output card and a field device implement this topology. Also, each branch of a star topology is point-to-point communication.

Mesh network: A network that allows all devices to communicate with each other using bridges and/or switches. The most common implementation is wireless transmissions. The network may organize and reorganize itself to find the best or alternative ways to exchange information between equipment. With Industry 4.0 standards for seamless, complete interoperability, one can achieve a mesh-like communication over a bus network

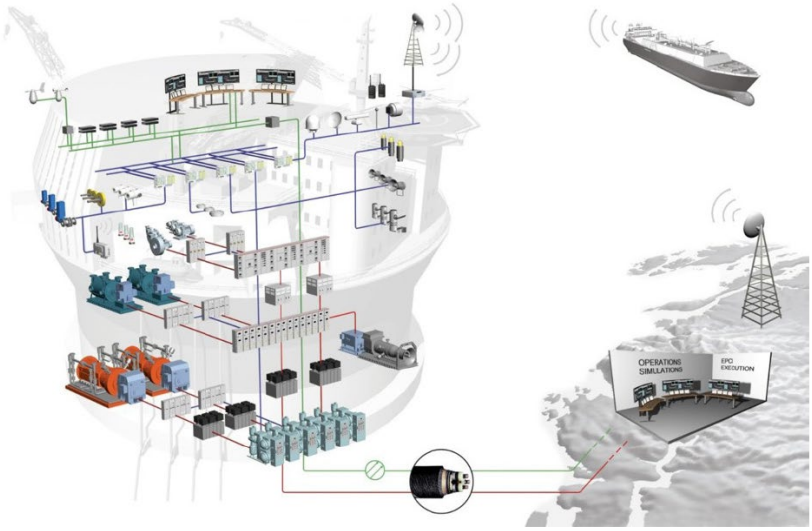


Fig. 6. Example of network architecture. From ABB Ability™ System 800xA® user manual

Several network topologies may be combined in a plant or factory network. For example, Fig. 6 shows how ABB presents a combination of bus, point-to-point, start, and (most likely) ring network.

2.3.1.2 Logical network topologies

Within the Purdue reference architecture's network layers, networks may be logically segmented into virtual local area networks (VLANs). For example, at Purdue layers 1 (controller) and 2 (supervisory, control room), the network traffic can be segmented into:

- VLAN for controller-to-controller communication and local input/output units.
- VLAN for OPC UA clients and servers handling data exchange with higher Purdue model layers.
- VLAN for engineering and maintenance access, for example, for configuration of the controller application program.

Segmentation is helpful for network performance and security, as it allows assigning different priorities to traffic across VLANs and reduces the ease with which attackers can move freely across networks.

2.3.2 Roles for organizing traffic over the network

The exchange of data over the network relies on systems taking on different roles in this process. We often distinguish between master-slave and client-server.

The terms master, slave, client, and server describe the role of each device. The names may seem overlapping, but they do have some principal differences, as summarized in the following:

Master-slave	The network consists of one or more masters that have assigned a group of connected devices named slaves, where: • The master has unidirectional control over the slaves, meaning no communication occurs unless the master asks for it. • The connected devices (slaves) only communicate when the master asks to respond. • The master can write data to the slaves, e.g., a configuration value or set point. • The networks may allow more than one master. In this case, the masters must actively take advantage of opportunities to send their requests when the network is not busy. Fieldbus communication applies this way of organizing data exchange.
Producer-consumer (Consumer-provider)	The network consists of one or more consumers that wait for data and providers that send the data. • Typical consumers are controllers, while typical providers are field devices, • The exchange builds on a "contracted" (subscription) agreement about what type of data the consumer is to receive from connected providers. • The providers publish data as multicast or broadcast, and consumers "pick up" the data relevant to their subscription. • It is the providers that determine when the data is to be sent, and this would be when something "happens", often when a value changes by more than a given threshold. However, the contract can also be to share data at certain intervals, for example, cyclic data transmission with strict regular intervals, down to a few ms or as required for the control or safety functions. Some of the Industrial Ethernet (IE) standards apply this type of data exchange.
Client-server	Client-server refers to communication between software applications and databases, both within OT, between OT and IT, and within IT. The client is the system or software that needs data, and the server is the system or software that holds and provides it. • The clients can request data from one or more servers and, with necessary permissions, perform write operations. • For each read and write operation, the client has a one-to-one connection to the relevant server. • The servers can share or publish data with clients when requested or according to the subscription. • Some devices can take the role of a client and a server, depending on the context. • The communication occurs in a session, initiated and closed by a client. • Clients do not interact directly but always via a server. • Unlike master-client and consumer-provider, client-server (and pub-sub) apply a more semantically rich exchange of data, being organized in node-graph structures (information models). OPC Classic implements client-server communication, while OPC UA offers it but uses it less frequently.
Publish-subscribe (pub-sub)	Publish-subscribe (pub-sub) is the name of a data exchange model for client and server software applications that need data, and those that hold and provide it. • Pub/sub is now the common way to exchange data using OPC UA, where OPC UA servers are publishing while the clients, performing analysis and other uses of the data, are subscribing.

	• The subscription is configured so that different clients may subscribe only according to their needs, and the servers only present data upon certain criteria. The criteria can be that something happens (events) or that a continuous value changes beyond a present threshold. • Compared to client-server, it is much more efficient and scalable, as the client does not need to ask, just in case, if something has happened that requires an exchange. • This means that pub/sub enables the exchange of large amounts of data. While producer-consumer and pub/sub share the same underlying concept of decoupled data exchange, they differ in context: producer-consumer refers to (often real-time) exchange mechanisms available with some IE protocols, whereas Pub/Sub refers to exchange of larger amounts of semantically rich data, for purposes of analysis and monitoring, and not for control and time-critical operations.

2.3.3 Reference model for communication stacks

Messages transported from one device to another rely on a set of logical and physical functions. The sum of these functions is sometimes called a communication stack.

There are two widely recognized reference (i.e., generic) models for such communication stacks:

- The OSI model
- The TCP/IP model.

The two models divide the communication stack into layers, each serving a specific role in the communication process.

2.3.3.1 OSI model

OSI stands for Open Systems Interconnection (OSI), and the corresponding stack model was introduced in ISO/IEC 7498-1 (1994). Despite its age, it remains one of the most widely recognized ways to organize the functions of a communication stack. The OSI model divides the stack functions into seven (7) sub-functions, referred to as layers:

- **Layer 7:** The application layer holds the message structure with data in the connected application's protocol format.
- **Layer 6:** The presentation layer translates the message from the format needed for sending or receiving to the format used by the application layer.
- **Layer 5**: The Session layer starts and closes the sending or receiving of data.
- **Layer 4:** The transport layer splits (when sending) the message into manageable packets for transportation over the network, and alternatively reassembles the pieces (when receiving)
- **Layer 3:** The Network layer adds an address to the packets for the destination that the network can interpret.
- **Layer 2:** The data link layer splits packets (when sending) into smaller pieces, or frames, with addresses that the physical transmission medium can interpret, or rejoins the frames into packets.
- **Layer 1**: The physical layer transfers each bit of the frame over the transmission medium, often a cable, optical fiber, or wireless connection. The cable may support parallel or serial bit transmission.

The functionality of each layer is further detailed in Tab. 1. A protocol need not implement all layers. For example, as explained later, Fieldbus implements layers 1, 2, and 7, while most IE implement layers 1, 2, 3, 4, and 7.

Note: These OSI layers must not be confused with the Purdue reference architecture levels.

Tab. 1. OSI model layers

Layer	Layer name	Function/tasks	Format
7	Application	Provides a user interface that allows a human or a software application to initiate a send or a data request. In addition, organizes data to be sent in the format for a given protocol. For example, data is organized for exchange via the hypertext transfer protocol (HTTP), the file transfer protocol (FTP), or industrial protocols such as Profibus DP, Profinet, Modbus, and HART. In a secure protocol such as HTTPS, the application layer retrieves the public and private keys to be shared with layer 6 for encryption and decryption. The application layer also adds necessary information to characterize the data, such as whether it is an alarm, a continuous measurement, or historical data, and the type of operation.	Message with headings and payload (data)
6	Presentation	Translates the message into a suitable machine-readable format for transmission (0/1s and symbols, like ASCII) – and vice versa. If relevant, a conversion is added for encryption (alternatively, decryption) using an encryption/decryption method and an encryption key. Public and private keys may be used for encryption and decryption, respectively.	
5	Session	Manages the initiation, synchronization, and closing of a message transfer. Also performs authentication and reconnection in the event of an interruption.	
4	Transport	Sending: Decomposes the message into smaller segments (or datagrams), each with a header. The header includes a sequence number, checksums, and code for the transmission type. Two examples of coding alternatives are TCP (Transmission Control Protocol) and UDP (User Datagram Protocol). The transport layer can also generate session keys for encryption and decryption and convert the packets accordingly. When the HTTPS protocol is used, encryption and decryption occur at layers 4 and 6. Receiving: The packets are checked, assembled, and, as needed, decrypted into the original message format.	Segments (TCP) and datagrams (UDP)
3	Network	Sending: Adds a destination address to each packet, identifying the device's location on the network. The most common approach is to use the Internet Protocol (IP): each packet includes information such as the header length, the packet length, and whether the transmission is over TCP or UDP. The IP address identifies the device's location on the network and is used to determine a route to reach it. As such, the IP and MAC addresses (at layer 2) complement each other. Receiving: Reverts each packet to the format expected by the transport layer.	Packets
2	Datalink	Sending: Divides packets into smaller units called frames (or telegrams), manageable pieces for physical transmission (e.g., a line). • With Ethernet, the data link layer manages the physical media access control (MAC) addresses, which uniquely identify a device on the network. The data link layer also initiates the frame transmission when the transmission line is (assumed) free. Manages collision avoidance by stopping when other traffic is detected, then retrying. Time management is also ensured.	Frames (Ethernet/Token ring) (or telegrams for Fieldbuses)

Layer	Layer name	Function/tasks	Format
		• With traditional fieldbuses, device addressing follows the format of the protocol. Collision avoidance and time management are also ensured. Receiving: Assembles the frames into packets in the format needed by the network layer.	
1	Physical	Transmits or receives individual bits within frames or telegrams over the physical (or wireless) transmission line. Serial transmission means that one bit is transmitted at a time. Bits are transferred by synchronizing and modulating the signal over cables or via radio waves for wireless communication. Modulation methods include amplitude modulation, frequency modulation, and phase modulation. Some examples of cable types used are: • RS-232: Cable with 9 wires. Point-to-point communication over shorter distances (15-20 meters) between two devices. • RS-485: Two-wire cable (twisted) that sets up a circuit between two defined endpoints and allows multidrop connections, i.e., where each connected device is physically connected to the electrical circuit used for signal transmission. • Ethernet: Consists of a cable with 2 or 4 pairs of wires plus Ethernet realization of the datalink layer (layer 2) and TCP/IP at layers 3 (network) and 4 (transport). Industrial Ethernet is a variant of this cable used in industrial environments. Typical characteristics include a rugged outer cable layer, more robust connections, and a shield. • Optical fiber: A Cable consisting of thin optical glass or plastic fibers with a diameter comparable to that of human hair. It transmits data via light pulses that travel through the core at roughly 70% of the speed of light in a vacuum. It is highly suited for high-bandwidth point-to-point links, can be multiplexed to serve multiple endpoints, and is predominantly used to span long distances due to minimal signal loss	Bits (as electrical currents or voltages, light pulses, or radio waves)

A simplified illustration of the layer functions is provided in Fig. 7. The padlock symbol refers to encryption, the speech bubbles to translation of data into machine readable coding, hourglass to session management, the handshake to the initiating and maintaining of the connection, the scissors to the splitting of data into smaller pieces, the nameplate & stamp to the adding of address (either IP/logical or physical), the alarm lamp to notify if checksum added to the message is not correct, the giveaway sign to control the access to the network, the car collision to explain collision avoidance and recovery, and (finally) the clock and toothed wheel to indicate the actual movement of bits and related synchronization. The whole sequence of tasks is reversed when data is received over the network and eventually presented to the receiving device.

Fig. 8 illustrates how the layers may apply to a sender (device A), a network device, and a receiver (device B), indicating that not all layers may necessarily be implemented.

Summarized, the OSI layers:

- Defines data formatting: Specifies the structure of the data to be sent, including the data type, the operation to be performed, and the payload content (handled by layers 5–7).

- Encapsulates data into smaller units: Segments data into segments at layer 4, packets at layer 3, and frames or telegrams at layer 2, adding necessary header information at each stage.
- Applies Sequencing (Where applicable): Adds sequence numbers at the transport layer (layer 4) for message reassembly or (in case of safety-critical communication) embeds counter sequences into layer 2 to prevent data repetition.
- Performs multi-layer error checking: Calculates and appends checksums at multiple stages, using layer 2 (e.g., CRC) for frame integrity, layer 3 for routing headers, and layer 4 for true end-to-end segment validation.
- Injects network addressing: Adds logical addresses (e.g., IP addresses at layer 3) for routing across wide networks or the internet, and physical addresses (e.g., MAC addresses or node IDs at layer 2) for direct device-to-device local delivery.
- Secures data: Applies, if included, encryption and cryptographic signatures (typically at layer 6 or via secure tunnels at Layers 3/4) to ensure data confidentiality and integrity.
- Manages reliable delivery: Oversees data re-transmission (via layer 4 TCP or specific deterministic layer 2 recovery mechanisms) if data units are corrupted or lost in transit.
- Converts data into physical signals: Translates raw structured frames into a continuous stream of binary bits at layer 1, encoding them into physical waveforms, such as electrical voltage pulses (RS-485, 4-20mA), light signals (fiber optics), or radio waves, for transmission across the physical medium
- Processes in reverse upon receipt: Unwraps headers, verifies checksums, and reassembles the raw payload in exact reverse order when data reaches its destination.

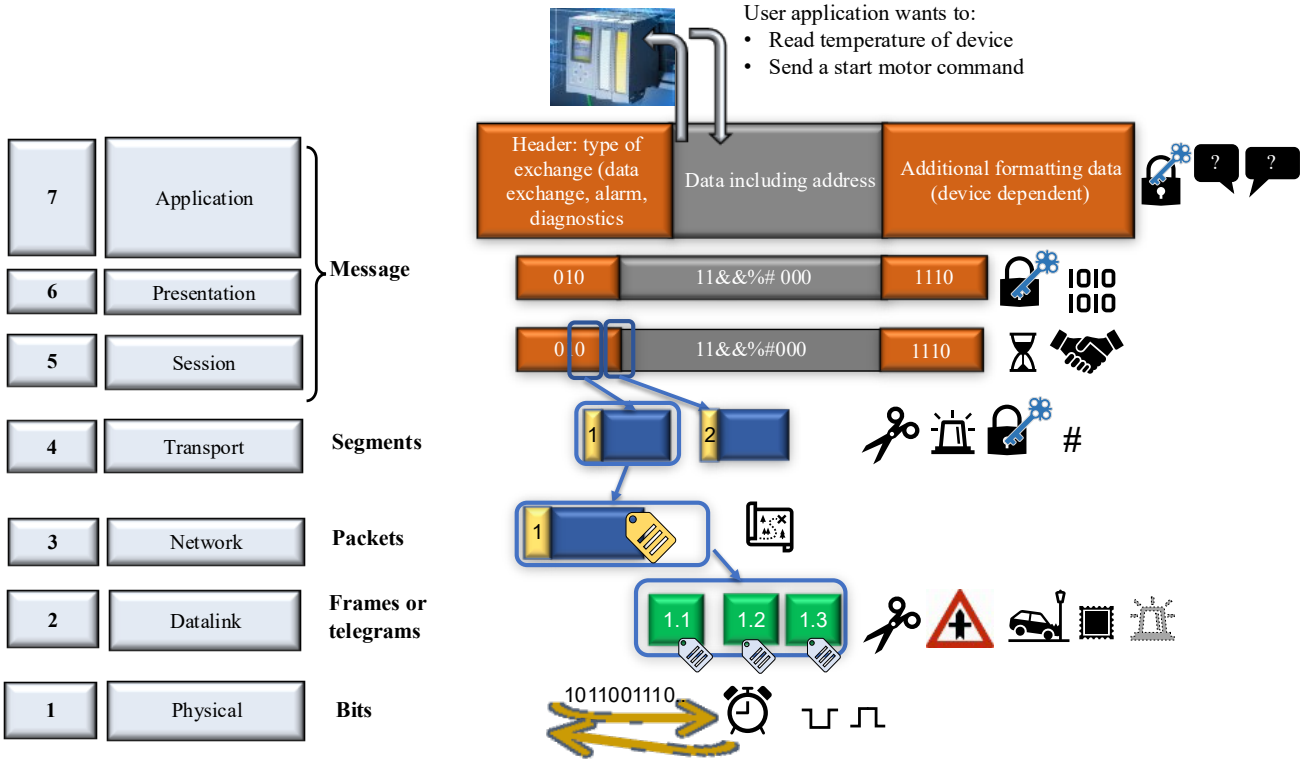


Fig. 7. Functionalities of the different OSI model layers

The exact methods used and the extent to which all layers are implemented depend heavily on whether the protocol is a standard IT network stack (TCP/IP), Industrial Ethernet, or an industrial Fieldbus, as well as on the underlying protocol specification.

	Take a look at how RealPairs explains the OSI model on YouTube.

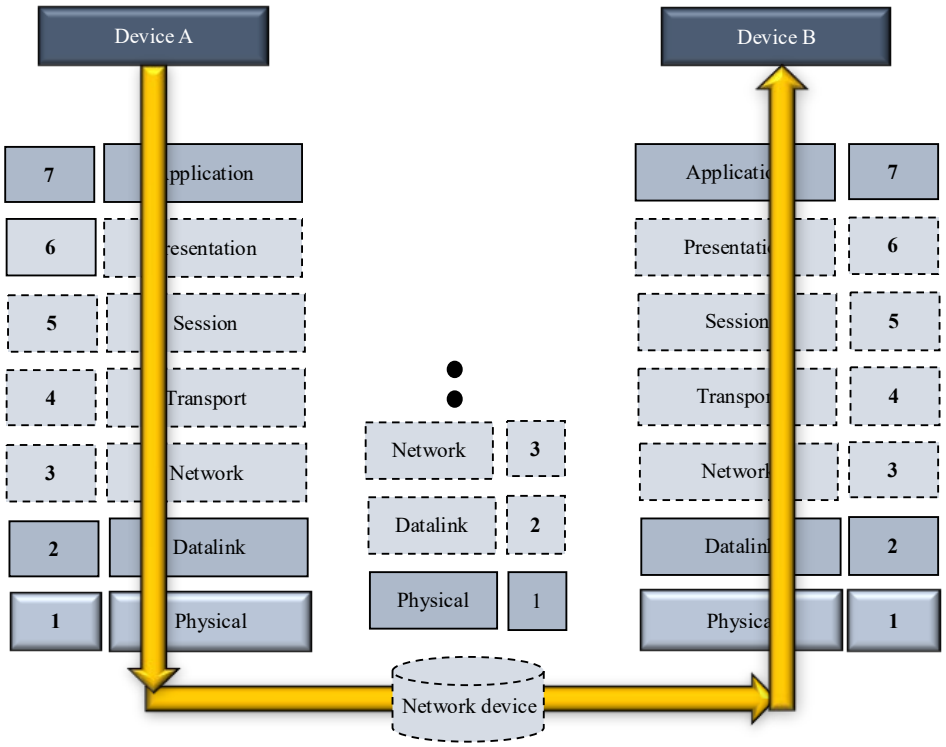


Fig. 8. Data exchange via the OSI layers

2.3.3.2 TCP/IP

The TCP/IP model, shown in Fig. 9, is a variant of the OSI model, tailored for TCP/IP communication stacks. The figure illustrates that TCP/IP covers the same functions as the OSI model, but is more specific about the technologies used.

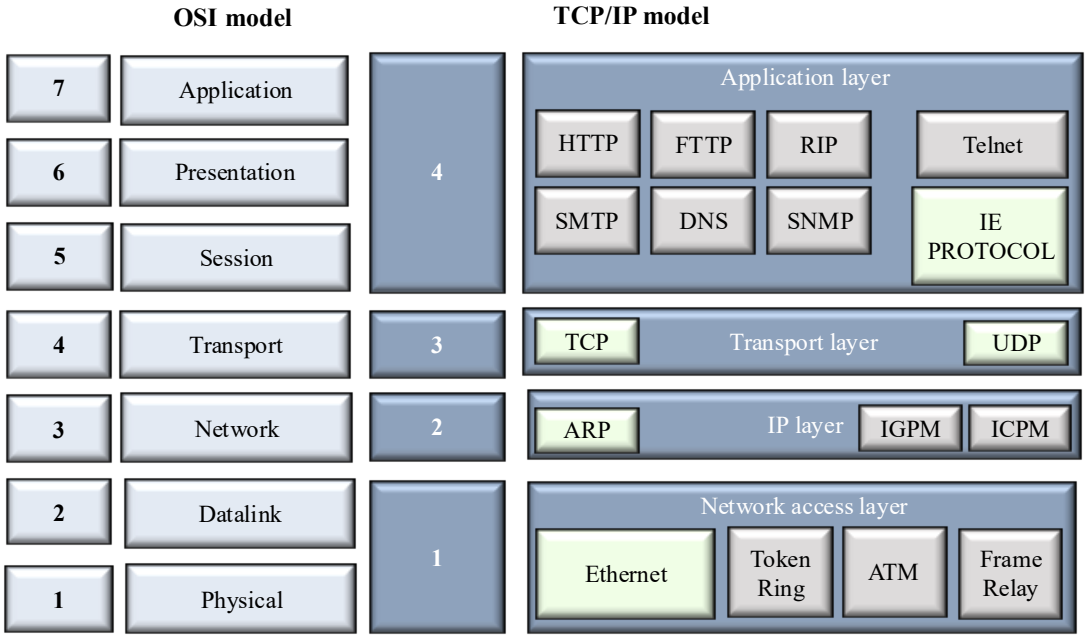


Fig. 9. OSI model compared to TCP/IP

It divides the stack into four subfunctions or layers:

- **Layer 4:** The application layer covers the tasks of the application, presentation, and session layers in the OSI model. A suite of TCP/IP protocols suited for this layer is listed as e.g., hypertext transfer protocol (HTTP(S)), simple mail transfer protocol (SMTP), file transfer protocol (FTP), domain name system (DNS), simple network management protocol (SNMP) or the relevant IE protocol (e.g., Profinet). The IE protocol does not seem to implement the presentation and session part layer 4, as it needs to simplify the architecture to achieve strict real-time performance.
- **Layer 3:** The transport layer transfers messages using TCP or UDP.
- **Layer 2:** The network layer uses IP protocols together with the address resolution protocol (ARP), the Internet Group Management Protocol (IGMP), or the Internet Control Message Protocol (ICMP). These are not explained further; however, a search shows that ARP is used in relation to the Profinet protocol.
- **Layer 1:** The network access layer covers the data link layer and physical layer and may be implemented by Ethernet, Token ring, Asynchronous transfer mode (ATM), or Frame delay (the latter seems less common for newer devices).

Fig. 9 shows that TCP/IP is a special implementation of the OSI model, covering the same type of functions, despite their different organizations. While all protocols can be explained using the OSI model, only some can be explained using the TCP/IP model.

2.3.4 Network devices

The network requires several specific devices to connect and route data, some of which are identified in Fig. 10. Such network devices are often explained a bit differently among different sources, with the primary focus on using the glossary by IEC Electropedia (IEC glossary), Section 782 on computer network technology, when possible.

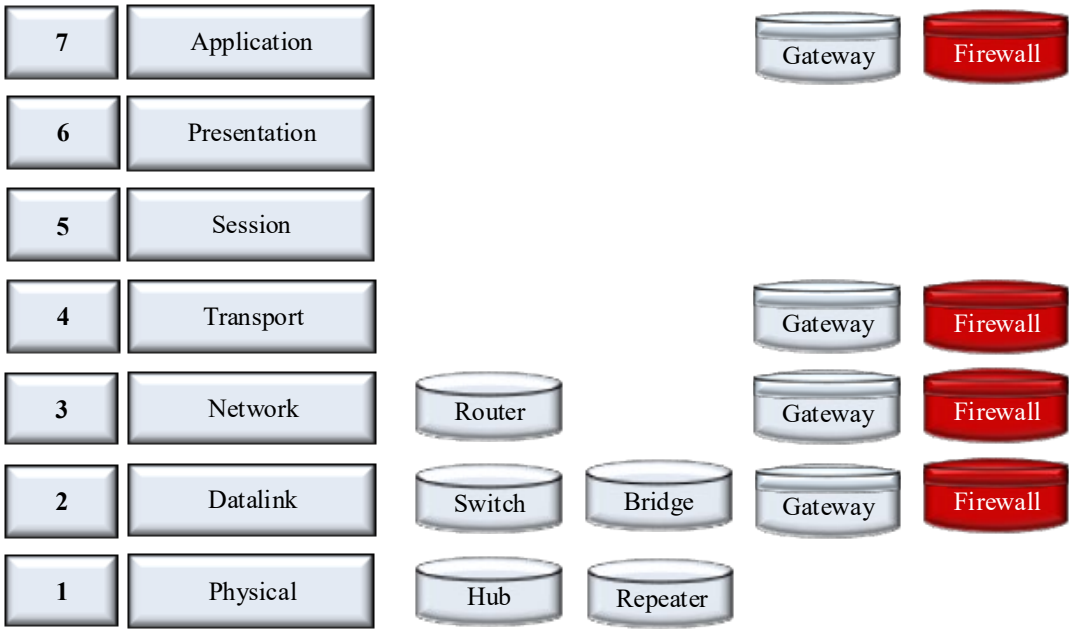


Fig. 10. Possible implementations of network devices in OSI model.

Physical layer:

Repeater: A device for one-to-one (or point-to-point) signal regeneration of signals between two network segments. Repeaters can connect multiple network segments over considerable distances, where cable length can degrade the signal. The repeater makes no analysis messages, but forwards everything it receives.

Hub: A device that can connect one segment to several others. It forwards everything to all and acts as a broadcaster. It is often used to implement star and tree topologies.

Repeaters and hubs are never needed for Ethernet, including IE, but may be needed for Fieldbus

networks.

Datalink layer:

Bridge: A device that is used within a single network and which ensures that the messages reach the correct destination. Compared to a switch, it has simpler functionality.

Switch: A device that connects devices within a single network, allowing for one-to-many or many-to-many connections (or ports). As needed, the network may add additional switches to cover all devices. Compared to a bridge, it offers several advanced features (beyond connecting two networks), such as buffering, handling delays and corruption, and performing operations faster. There are two types of switches:

- Managed switches, which follow a set of rules related to the quality of data, diagnostics, control of speed to prevent malfunctioning, and management of data spikes, storms, and bottlenecks.
- Unmanaged switches, which route messages without restrictions. An unmanaged switch is sometimes referred to as a modern multi-port bridge.

Network layer:

Router: A device that sets up the best path between devices that communicate across different networks. A router is needed for Ethernet when connecting a VLAN to the internet or another VLAN.

Operates at several layers:

Gateway: A device that connects two or more networks using the same or different protocols. It can operate at layers 2, 3, 4, and 7 of the OSI model, depending on the level at which translation is needed.

Firewall: A device through which several data streams can pass (like in a gateway) that checks and restricts data flow that could stem from unauthorized access. Examples of functions conducted by a firewall are:

- Package filtering: A rule-based check of packets to decide whether they are allowed to pass or not.
- Physical (MAC) address and VLAN tag filtering: A rule-based check of MAC addresses and VLAN tags (to permitted networks) that are allowed.
- Application filtering: A control that determines which applications (programs run on connected devices in the network) are allowed to send messages over the network.
- Logging: Logging of what messages and files have been allowed to pass and which ones have been prevented/prohibited.
- Enabler of secure end-to-end communication (“VPN”): Allows secure (authenticated and enciphered) point-to-point communication between devices on the same or in different networks through the firewall.

2.4 Fieldbus and Industrial Ethernet (IE)

Industrial communication is often placed into one of the following categories: Fieldbus or Industrial Ethernet (IE). While some Fieldbus and IE communication protocols are proprietary, most have been standardized and published by international organizations, such as the International Electrotechnical Commission (IEC). This is why we use the term "communication *standard*". The most important overarching international standards that define open industrial communication standards (or protocols) are:

- IEC 61158 - Industrial communication networks. Describes the full details of protocols
- IEC 61784 - Industrial communication networks. Describes commonly applied configurations of protocols, called profiles, so that devices using these are easier to integrate.

Each of the above standards consists of several parts, or sub-standards, each focusing on a specific type of protocol profile, meaning that together they cover more than a thousand pages.

There are also many other international standards not mentioned here, for example, for wireless communication and communication protocols for specific application areas like electrical power distribution plants and railway signaling.

2.4.1 Fieldbus

A Fieldbus is a real-time communication technology that implements OSI layers 1, 2, and 7, alternatively only layers 1 and 2. The application of Fieldbus (as for IE) is controller-to-field device and controller-to-controller communication. Communication is typically organized as master-slave, except for foundation fieldbus, which can also apply a publisher-subscriber model for data exchange.

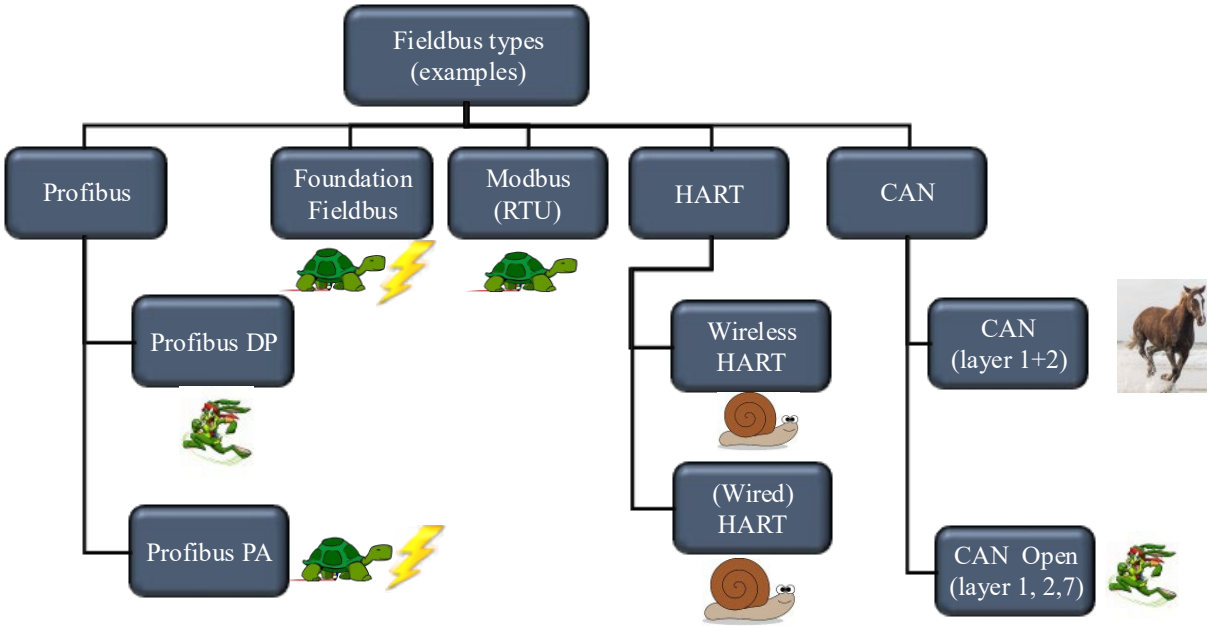


Fig. 11. A simple comparison of some selected Fieldbus standards

Examples of Fieldbus standards or protocols are:

- Profibus, with the two variants Profibus DP and Profibus PA
- Foundation Fieldbus (FF)
- HART (including WirelessHART)
- CAN: CANopen (implements layer 1 and 2 only), DeviceNet
- Modbus RTU

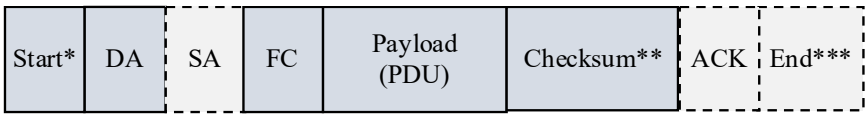
Fig. 11 provides a simple comparison of these, focusing on their transmission speeds. Here, the rabbit symbolizes fast transmission speed, the elephant medium speed, and the turtle low speed. The lightning symbol indicates that power can be supplied over the same cable used to transmit digital messages (as a current loop). A more detailed overview of some of the properties is provided in Tab. 2, based on a review of various webpages.

Tab. 2. Selected properties of commonly used Fieldbus standards

Fieldbus Standard	Transmission speed	Cable type and length	Transmission medium	Message exchange	Devices connected	Frame length
Profibus DP	From 9.6 kbps (1200 m) to 12 Mbps (100 m)	From 100 m -1200 m (RS-485), up to 15 km for optical	RS-485 (two wires), fiber-optic cable	Cyclic request (polling) by the master, limited time synchronization	Up to 126 devices	6-246 bytes (max 244 bytes as payload)

Profibus PA	31.25 kbps	1900 m	Power loop (two wires) according to IEC 61158-2	As for DP	Up to 125 devices	As for DP
Modbus RTU	From 9.6 kbps to 19.2kbps	< (ca) 350 m	RS-485 for multi-drop or RS-232 (three wires) for point-to-point, fiber optical	Asynchronous. No cyclic request, no time synchronization.	Up to 254 devices	4-256 bytes (max 252 bytes as payload)
HART	From 1.2 kbps to 3.5 kbps	< (ca) 350 m	RS-485 (for primary data exchange), RS-232, Bluetooth, and wireless for handheld devices used for configurations and diagnostics checks. HART modems and wirelessHART as options for transferring primary and configuration data.	Asynchronous. No cyclic request, no time synchronization.	Up to 64 devices Wireless. Up to 250 devices	9-284 bytes (max 253 bytes as payload) Wireless: 20 bytes to 127 (max 80-90 bytes as payload)
IO-Link	Three options: 4.8 kbps (COM1), 38.4kbps (COM2, most common), 230.4kbps (COM3)	Specialized cable (typically with 3 wires) Maximum 20 meters (from IO-Link master to field device)	Standard unshielded industrial cable with 3 wires or 5 wires	Cyclic data exchange. No global synchronization. Supports only one-to-one communication between the IO-Link master and each device.	Each IO link master has 4-8 ports. IO-link Wireless can have a maximum of 40 devices connected.	COM2: approx 40-60 bytes (0-32 bytes as payload)
CAN						
CAN (low speed)	125 kbps	500 m	EN 50325/ ISO 11898-3 (two wires), fiber optical	Event-driven. No time synchronization.	Up to 32 devices	Approx 47-159 (8 bytes as payload)
CAN (high speed)	Up to 1 Mbps	20 m (1Mbps)	EN 50325/ ISO 11898-2 (two wires)	As for low speed.	Some mention 50 (or a few more) devices	As for low speed
CAN Open	Up to 1 Mbps	20 m (1Mbps)	EN 50325/ ISO 11898-2	Event-driven. Limited time synchronization.	Up to 127 devices	As for low speed

It is not easy to generalize the structure of a Fieldbus message, as the field formats and coding vary across the Fieldbus standards. However, Fig. 12 identifies some that may be considered as “typical” for Fieldbuses for the data link layer. Stippled fields within the frames are sometimes excluded, depending on the protocol. The payload is often referred to as a protocol data unit (PDU) and may contain only data or additional bits and bytes used for, e.g., error checking.



DA: Destination address FC: Function alternatively command code (read, write,..) PDU: Protocol data unit
SA: Source address ACK: Acknowledge
***CRC or other checksum calculations. ***In case of preamble, end bit/byte is not included*

Fig. 12. Principal structure of Fieldbus frames (data link layer)

Many Fieldbus standards rely on RS-485 as the transmission medium, a cable consisting of three (twisted) wires plus shielding: two for data transfer (send and receive) and one for reference (earth reference). The IEC 61158-2 cable is a two-wire cable for data transfer in a current-loop configuration, meaning that power and signal are carried over the same wires. All cables have shielding to reduce their susceptibility to disturbances in an industrial environment. Common practice is to connect the shield to ground (earthing system) at one end only in case a potential difference occurs between the two connection points, but this practice varies, and opinions differ.

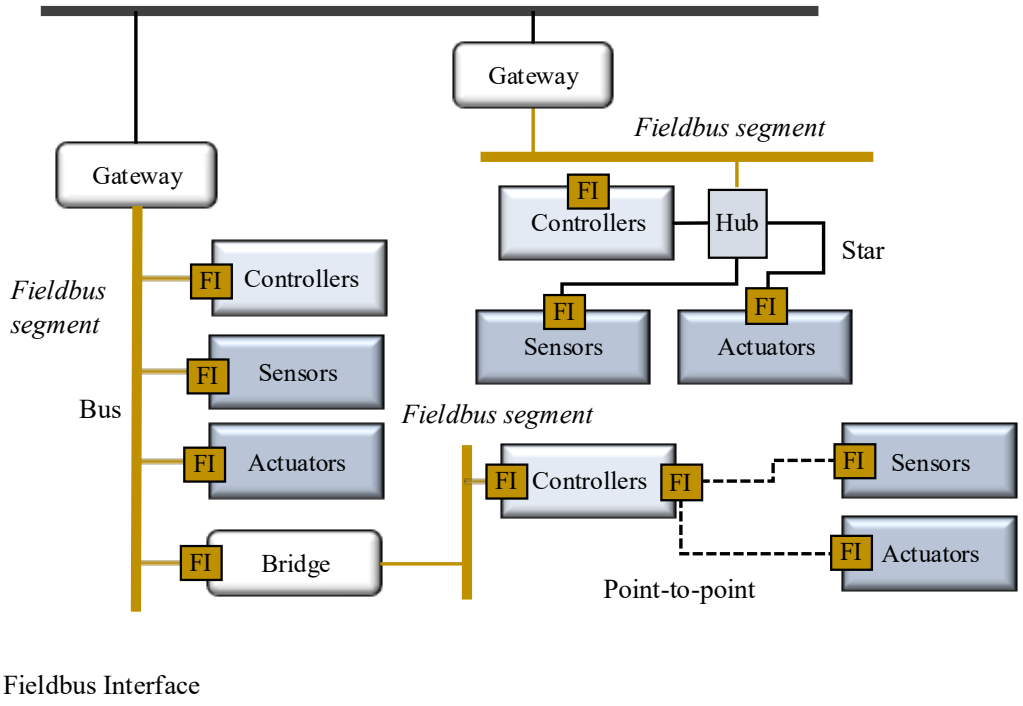


Fig. 13. Example of networks where fieldbus is used (adapted from IEC 61158-1)

All devices and systems connected to a Fieldbus network require an appropriate Fieldbus interface. For example, devices and systems connecting to a Profibus DP network must all have Profibus DP interfaces. Connecting networks that do not use the same protocol requires a gateway, while the connection within a network may use a bridge, as illustrated in Fig. 13. It means that each network segment connects systems using the same Fieldbus standard, and the gateways enable exchange across. In contrast, bridges connect network segments that use the same Fieldbus standard for communication.

2.4.2 Industrial Ethernet (IE)

Industrial Ethernet (IE) covers Ethernet-based protocols and cables, where cables are enforced to withstand an industrial, often harsh, environment. Examples of some IE protocols are:

- Profinet
- EtherCat (using peer-to-peer)
- Modbus TCP

- Ethernet/IP

IE *often* implements layers 1 and 2, layers 3 and 4 with TCP and IP, and layer 7, with the exception of Profinet RT (real-time), which operates at the MAC level and skips levels 3 and 4. Layer 7 is set up according to the specific IE protocol.

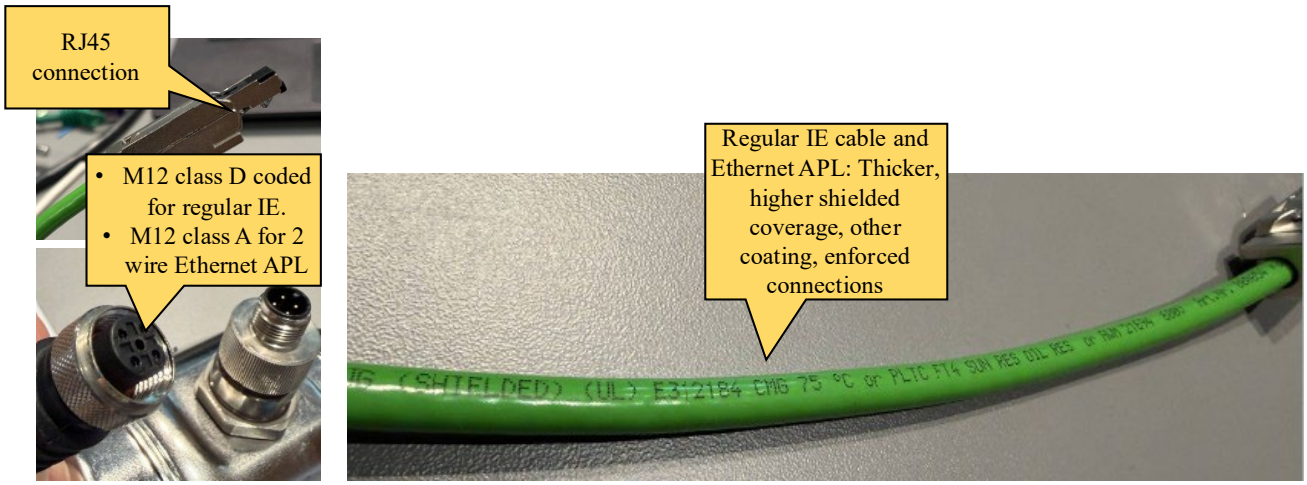


Fig. 14. IE cable and enforced connection types

It is the term "Industrial" added to «Ethernet" that refers to the reinforced Ethernet cable. Measures taken include more robust cable plugs/connectors, cable coating selected for outdoor use, and improved shielding (coverage) and armor to withstand factory noise, some of which are shown in Fig. 14. The Ethernet cable can have 4 or 8 wires, operating at transmission speeds of 100 Mbps or 1 Gbps respectively. Fig. 15 illustrates the high speed, as opposed to Fieldbus protocols, with the leopard symbol. Ethernet Advanced Physical Layer (APL), also illustrated in the figure, is a new type of physical infrastructure with APL power switches, APL field switches, and Power-over-Ethernet cables.

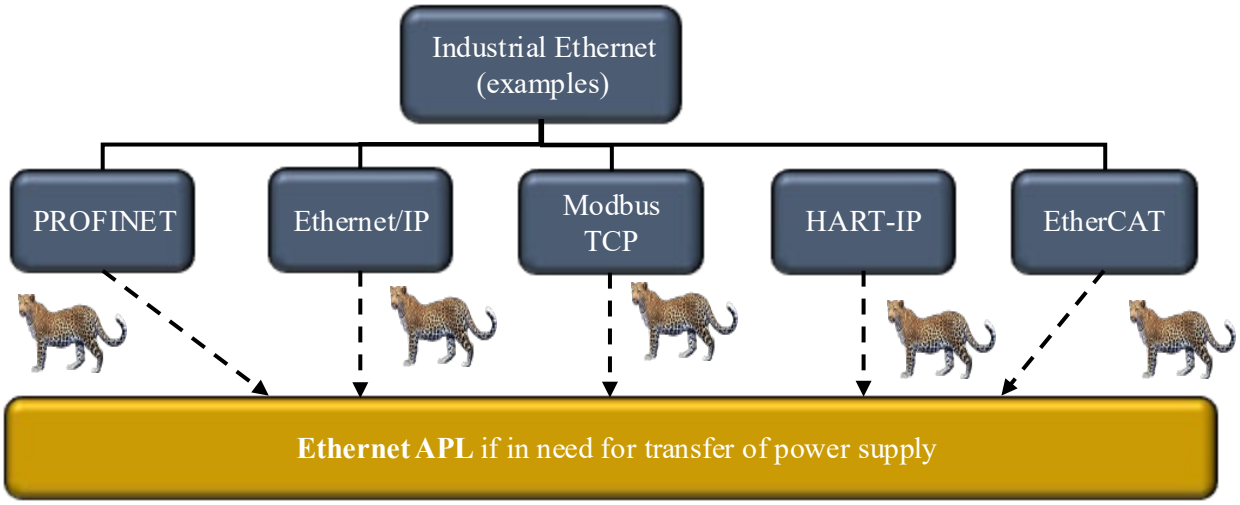


Fig. 15. A simple comparison of some selected IE standards

More properties for each of the IE standards are summarized in Tab. 3. Here, Profinet is the IE variant of the Profibus protocol, and Modbus TCP is the IE variant of the Modbus protocol. Until recently, all IE protocols were limited to transmitting messages only. For process automation, it has been advantageous to share the same wires for power and messages, since the distances from field devices to the controllers are pretty long. Several manufacturers joined forces in 2018 to form the APL project and develop a new transmission solution, Ethernet

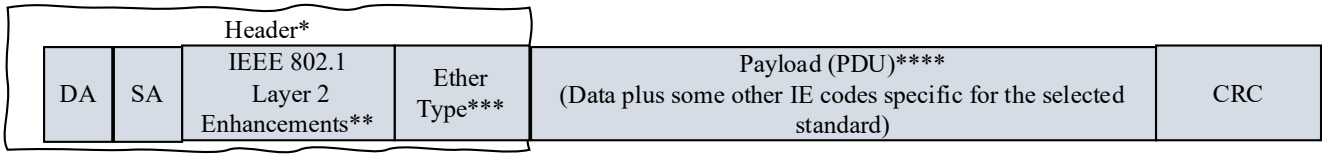
APL. As indicated in Tab. 3, it is possible to combine Ethernet APL with any IE standard, such as Profibus, Modbus, EtherCat, and HART IP.

Tab. 3. Selected properties of commonly used IE standards

IE standard	Transmission speed	Cable length	Transmission medium	Message exchange	Devices connected	Frame lengths	With Ethernet APL
Profinet	100 Mbps – 1 Gbps* (1-100 Gbps)***	100 meters** or Appr. 200 m – 10 km***	Enforced Ethernet Copper cable (or fiber optics***)	Cyclic, time-synchronized when needed for real-time data. Support asynchronous exchange for non-real-time data (diagnostics, configuration)	Up to 256 devices	Up to 1518 bytes (approx. 1440 bytes as payload)	Trunk lengths (all IE standards): Up to 1000 meters. Spur lengths: Up to 200 meters
Ethernet/IP				Cyclic (for real-time), asynchronous for diagnostics and configuration data. Optional time synchronization.	Several hundred devices	Up to 1518 bytes (approx. 1472 bytes as payload)	
Modbus TCP				May be cyclic if configured. No time synchronization.	Theoretically, extremely many, but the limitations are the controller capacity and the available IP addresses.	260 bytes (approx. 253 bytes as payload)	
EtherCat				Cyclic. Time-synchronized (with distributed clocks).		Up to 1518 bytes (approx. 1486 bytes as payload)	
HART IP				May be cyclic if configured. No time synchronization.		Up to 1518 bytes (approx. 1472 bytes as payload)	

*Max 100 Mbps with 4 wires (2 pairs) and 1Gbps with 8 wires (4 pairs). **Limited by the maximum achieved for Ethernet copper cable length. ***Fiber optics allows higher speeds and longer cable lengths; however, details must be consulted in 100Base-Fx, LX/SX, and SR/LR standards.

As for Fieldbus, it is not easy to generalize the structure of IE messages, as it varies across IE standards. However, an attempt is made for the data link layer with Fig. 16. One notable difference between IE frames and Fieldbus frames is complexity, and consequently, IE frames are generally much longer than Fieldbus frames. One exception is Modbus TCP, which is considered one of the simpler (and less complex) of the IE and Fieldbus standards.



**Ethernet Header or other protocol specific header. **Code defined in IEEE 802.1 to support traffic prioritization and segmentation, like TSN, VLAN tag, Priority Code Point, etc. ***Code identifying IE protocol type. ****Payload covers data, error checks, service/function codes, sequence code, etc.*

Fig. 16. Principal structure of an IE message frame (data link layer)

2.4.3 Safety-critical communication

Safety-critical communication is communication between controllers and field devices within a safety system, such as a safety-instrumented system (SIS). Such communication requires additional measures to ensure that lost or corrupted messages are detected and addressed, so that devices that receive a faulty message enter a safe state. It is important to note that the focus is on safety — i.e., handling random or human-error-induced faults — rather than on security, which concerns handling deliberate adverse acts.

The generic requirements for safety-critical communication are found in IEC 61508-2 (2010), the sector-neutral standard on the functional safety of safety-related systems. Here, functional safety refers to the required safety behavior of electrical/electronic/programmable devices and systems to ensure sufficient reliability for use in safety-critical applications. The requirements for design, implementation, and follow-up after deployment are covered in functional safety standards. The specific safety protocols, building on functional safety design principles, are covered in IEC 61784-3 (2021).

2.4.3.1 Black channel vs. white channel

IEC 61508-2 distinguishes between two alternative concepts to achieve safe communication: the White channel and Black channel, as illustrated in Fig. 17.

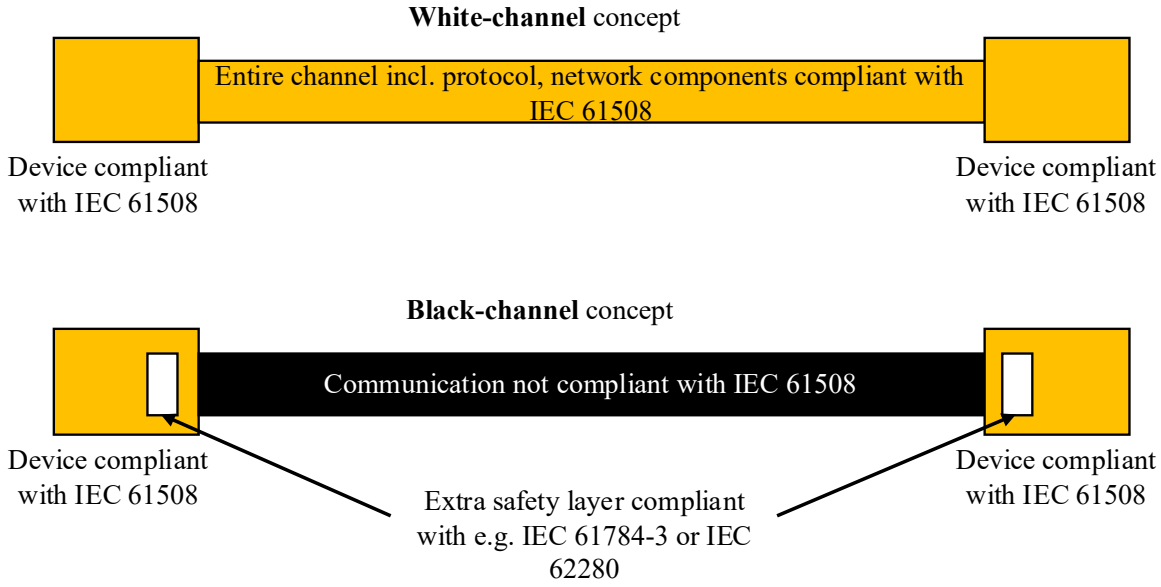


Fig. 17. White channel vs black channel concept

- **White channel:** Devices compliant with IEC 61508 (parts 1,2, and 3) communicate over a network whose protocol, network components, and services are all compliant with IEC 61508. In this way, the channel itself ensures the message is delivered on time and unmodified.

- **Black channel:** Devices compliant with IEC 61508 communicate over a network not compliant with IEC 61508, i.e, it can be any network connection ("black channel"). Instead, safety requirements are imposed on the devices' interfaces to detect faults and take the necessary actions to ensure safety. These features are added to an extra layer of the protocol standard, above level 7, as shown in Fig. 18.

Unlike white-channel communication, black-channel communication can exist on the same network as non-safety communication, as shown in Fig. 18. Reliability (availability) can be enhanced by connecting all devices to a redundant network.

The black-channel concept requires that the protocol's message structure (layer 7) be expanded with additional bytes, such as safety headers (indicating that the message is safety-critical), CRCs, and watchdog counters. The total message length for Profibus DP and Profibus PA is extended by 20-30 bytes (above the typical max message length of 244 bytes), and for Profinet, it is extended by 50-100 bytes (with a limit of 254 bytes for the user data payload).

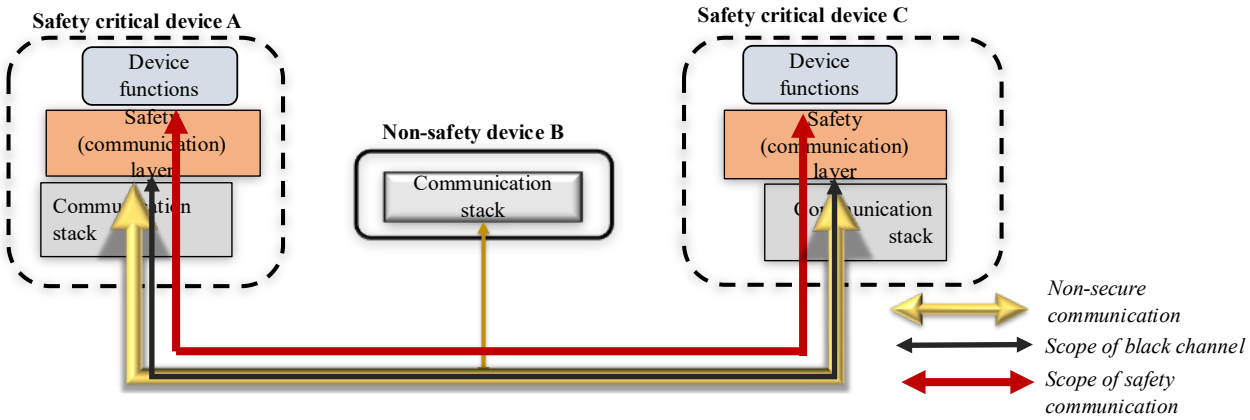


Fig. 18. Black channel concept for safety communication

The black channel is far more common than the white channel because it is less complex. The Fieldbus and IE communication standards for functional safety, implementing the black channel, is (IEC 61784-3, 2021), and its subparts.

Another often-referenced standard for safe communication, developed for the railway sector, is IEC 62280 (2014), a standard for safe communication in the railway sector. CENELEC has kept the content more up-to-date via the European norm/standard EN 50159 (2020).

2.4.3.2 Functional safety communication profiles

Protocol standards that have been extended to a functional safety variant have often added "Safe", "Safety", or "FS" (Functional Safety) to the name, for example:

- ProfiSafe (for Profibus DP, Profibus PA, and Profinet)
- Functional safety over Ethernet (FSoE) for EtherCat
- CANopen Safety

The purpose of the safety communication layer, located in the OSI model stack above layer 7, is to check whether the message has been corrupted or partially lost and, if so, instruct devices to enter a safe state. The safe state is application-dependent and may, for example, involve shutting down or stopping. When the receiver detects an incorrect checksum, it asks the sender to try again. If the checksum remains erroneous, the system is commanded to enter a safe state.

The functional safety profile usually adds safety-related data and additional bits and bytes to each frame of an existing protocol to provide extra integrity and error checking. The generalization of how a safety-critical message may look is not so straightforward, but one attempt is made in Fig. 19, inspired by IEC 61784-Appendix E. Common functions added include codes for revealing, e.g., timestamps, detection of repeated messages,

safety data for cyclic redundancy code (CRC) calculation, and the CRC signature. Some implementation details differ between Fieldbus and IE, and, of course, among different protocol standards.

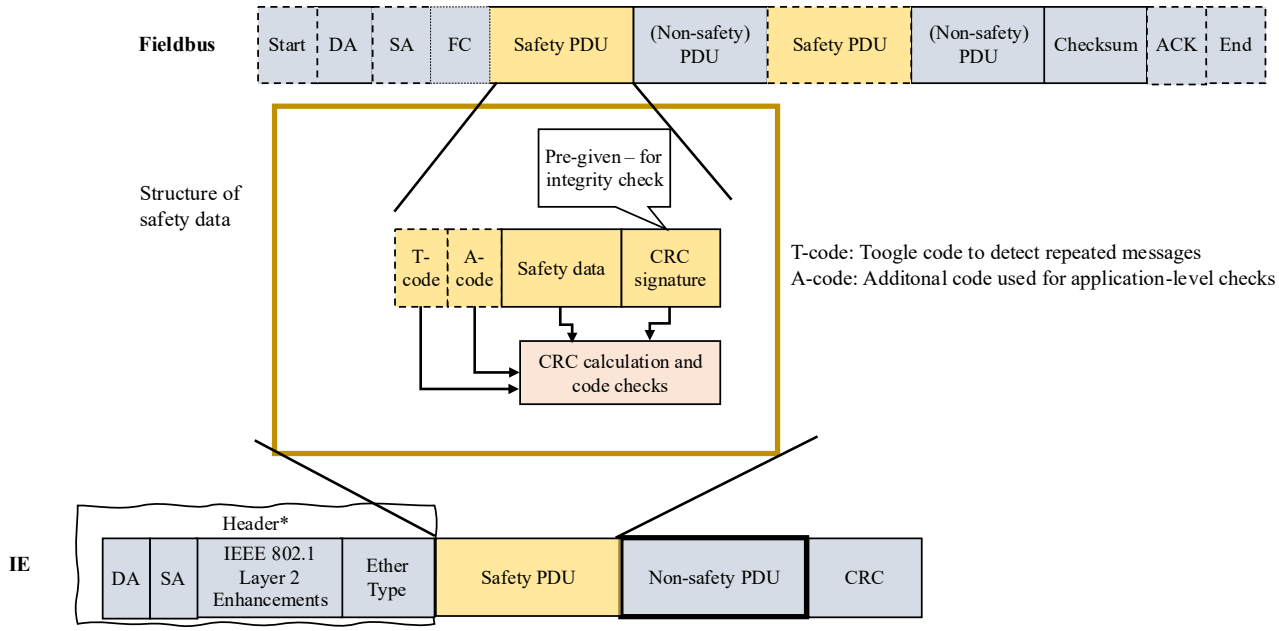


Fig. 19. Generalized format for a safety-critical message frame (adapted from IEC 61784-3)

An IE protocol that lacks a functional-safety profile can use openSAFETY (IEC 61784-3, FSCP 13) to transport safety-critical data. OpenSAFETY has been developed by the Ethernet POWERLINK Standardization Group and uses the black-channel principle, encapsulating each safety message as two identical subframes, each protected by its own CRC, with additional sequence and time-monitoring to ensure integrity and timeliness. Compared to protocols with native functional-safety profiles, openSAFETY operates independently of other (non-safety) protocols.

For example, since Modbus/TCP has no native functional-safety profile, safety-critical data can be exchanged over the same network with openSAFETY (IEC 61784-3-13, FSCP 13) without interference from other Modbus traffic. OpenSafety can also coexist with protocols for which a native functional safety profile could have been used, such as Profinet, EtherNet/IP, SERCOS III, and POWERLINK. While openSAFETY is certified to IEC 61508, certification is still required when it is combined with another protocol per IEC 61784-3.

In what follows, we will examine a specific safety-critical communication profile: the ProfiSafe protocol.

2.4.4 Comparison of Fieldbus and IE

Some key attributes between Fieldbus and IE can be briefly summarized as follows:

Fieldbus	• Simple messaging and routing by implementing OSI layers 1,2, and 7. • Primarily developed for controller (master) to field devices (slaves) communication. • Only one master in the network. • In exceptional cases, it can implement peer-to-peer, where any device can take charge of the communication without needing a central controller. • The physical layer (layer 1) is realized by two-wire shielded twisted pairs of type RS-485, fiber optic cable, or, for some selected protocols, cables that can transfer signal and power simultaneously following IEC 61158-2. • Serial transfer, bit by bit. • Transfer capacity ranges from very low (a few kbps) to medium (100 Mbps)

	• The typical maximum message frame length is 250-260 bytes; the lowest is around 6-12 bytes.
IE	• Allows controller-to-controller as well as controller-to-field devices communication. • Can implement client-server, including publishing-subscribing (pub-sub), or peer-to-peer. Here, the clients have the role of masters. • More advanced messaging and routing with OSI layers 1,2,3,4, and 7 with TCP or UDP at layer 4 and Internet Protocol (IP) at layer 3. The exception is Profinet RT, which does not implement layers 3 and 4. • The physical layer (layer 1) is realized with an Ethernet cable, which has some added protection for use in an industrial environment. • Short outreach, unless with Ethernet APL. • Possibility of parallel transfer of messages, even if packets are transmitted sequentially on each wire. • Transfer capacity from medium (100 Mbps) to very high (1 Gbps) • The typical message frame length is a maximum of 1500 bytes, while the lowest can be in the range of 60-64 bytes.

2.4.5 About the Fieldbus and IE standards IEC 61158 and IEC 61784

As mentioned, the two central standards are IEC 61158 - Industrial communication networks - Fieldbus specifications (several parts) and IEC 61784 - Industrial communication networks – Profiles (several parts). IEC 61158-1 (2023) and its other parts contain technical specifications of IE and Fieldbus protocols and transmission solutions, giving an overview of their full flexibility. In contrast, IEC 61784-1 (2017) and its subsequent parts provide suggested settings that the manufacturer can apply to make their products easier to integrate into networks with other devices that use the same IE or Fieldbus communication interface. The standard refers to each IE or Fieldbus standard where such settings have been applied as a communication profile family (CPF). In total, IEC 61784 covers more than 15 CPFs. For example, whereas IEC 61158 explains all possible configurations of Profibus DP, Profibus PA, and Profinet, and IEC 61784 specifies the specific settings to use and names this variant of Profibus as CPF 3, further split into CP3/1 (for Profibus DP), CP3/2 (for Profibus PA), and CP 3/6 (for Profinet). Without having agreed on such profiles, product vendors could choose different configurations, making it more difficult to exchange data between devices that use this protocol.

An overview of CPFs in IEC 61784 per 2025 is:

- CPF 1: Foundation fieldbus (FF): H1 (Fieldbus), HSE (IE)
- CPF 2: Common Industrial Protocol (CIP), including ControlNet (Fieldbus), EtherNet/IP (IE), DeviceNet (Fieldbus), and CompoNet (Fieldbus)
- CPF 3 Profibus: Profibus DP (Fieldbus), Profibus PA (Fieldbus), and Profinet IO (IE)
- CPF 4: P-Net (Fieldbus)
- CPF 5: CC-link (Fieldbus) and CC-Link IE Field
- CPF 6 Interbus (Fieldbus)
- CPF 8: WorldFIP (Fieldbus)
- CPF 9: EtherCat (IE) (CAN for IE)
- CPF 13: HART RTU (Fieldbus) and HART-IP (IE), including wireless HART
- CPF 15: Modbus TCP (IE) and Modbus RTU (Fieldbus)
- CPF 16: SERCOS III (IE)
- CPF 18: Ethernet Powerlink (IE) and SafetyNet
- CPF 19: Modbus TCP (IE) with some extensions (compared to CPF 15)
- CPF 22: Autobus (IE)

IEC 61784-3 (2021) and its subparts focus solely on extensions of CPFs that meet the requirements for use in safety-critical systems per IEC 61508, referred to as functional safety communication protocols (FSCPs). These variants are per 2025:

- Foundation Fieldbus SIS
- CIP safety
- ProfiSafe
- INTERBUS Safety
- CC-link Safety
- Safety over EtherCAT
- OpenSAFETY (based on Ethernet POWERLINK)
- EPASafety
- RAPIEnet Safety
- SafetyNET

Some may notice that CAN bus is not among the listed protocols. The specifications for Fieldbus variants of CAN were already introduced in other standards, and therefore not added to IEC 61158 and IEC 61784. ISO 11898, developed for road vehicles, contains specifications for CAN as Fieldbus, e.g, ISO 11898-1 (2024) and ISO11898-2 (2024). EN 50325, published by CENELEC, builds on ISO 11898 and provides specifications for CAN Open and CAN Open used for safety-critical communication with:

- EN 50325-1 (2019): General requirements
- EN 50325-4 (2022): CANopen
- EN 50325-5 (2010): Functional safety communication based on EN 50325-4

CAN implemented for IE is named EtherCat and is covered by IEC 61158 and IEC 61784 (as CPF 9).

As a final remark, one should be aware that there are many other international standards more specific to application areas or types of transmission, not only those from CENELEC and IEC, but also from IEEE.

2.4.6 Ring topology management in IE networks

IEC 62439 specifies a protocol for managing ring topologies for high-availability automation networks using Ethernet, where a ring or mesh topology is applied. The protocol is complementary, and not instead of, an IE protocol used for data exchange, and operates at level 2 of the OSI model. Its sole purpose is to manage fault situations in the network, so that data traffic is rerouted in case of cable breakage or other faults with a similar outcome. For example, IEC 62439-1 (2016) provides an overview of protocols implemented at OSI layer 2 (datalink), such as:

- Ring topologies: Media Redundancy Protocol (MRP), defined in IEC 62439-2 (2021)
- Mesh and ring topologies:
    - Parallel Redundancy Protocol (PRP): This protocol ensures zero recovery time by sending duplicate frames over two independent networks.
    - High-availability Seamless Redundancy (HSR): This ensures zero recovery time using the same approach as PRP.
    - Distributed Redundancy Protocols (DRP): These protocols provide redundancy across multiple network segments, not limited to ring networks.

The term “redundancy” may be a bit confusing or misleading, as it does not indicate that the ring itself is redundant. The redundancy refers to detecting cable breakage, unblocking the loop, and ensuring that data traffic can reach any device by automatically rerouting traffic in the opposite direction of the ring.

MRP is perhaps the most relevant in this chapter, as it is the most commonly applied for the IE protocols covered here, to the author's knowledge.

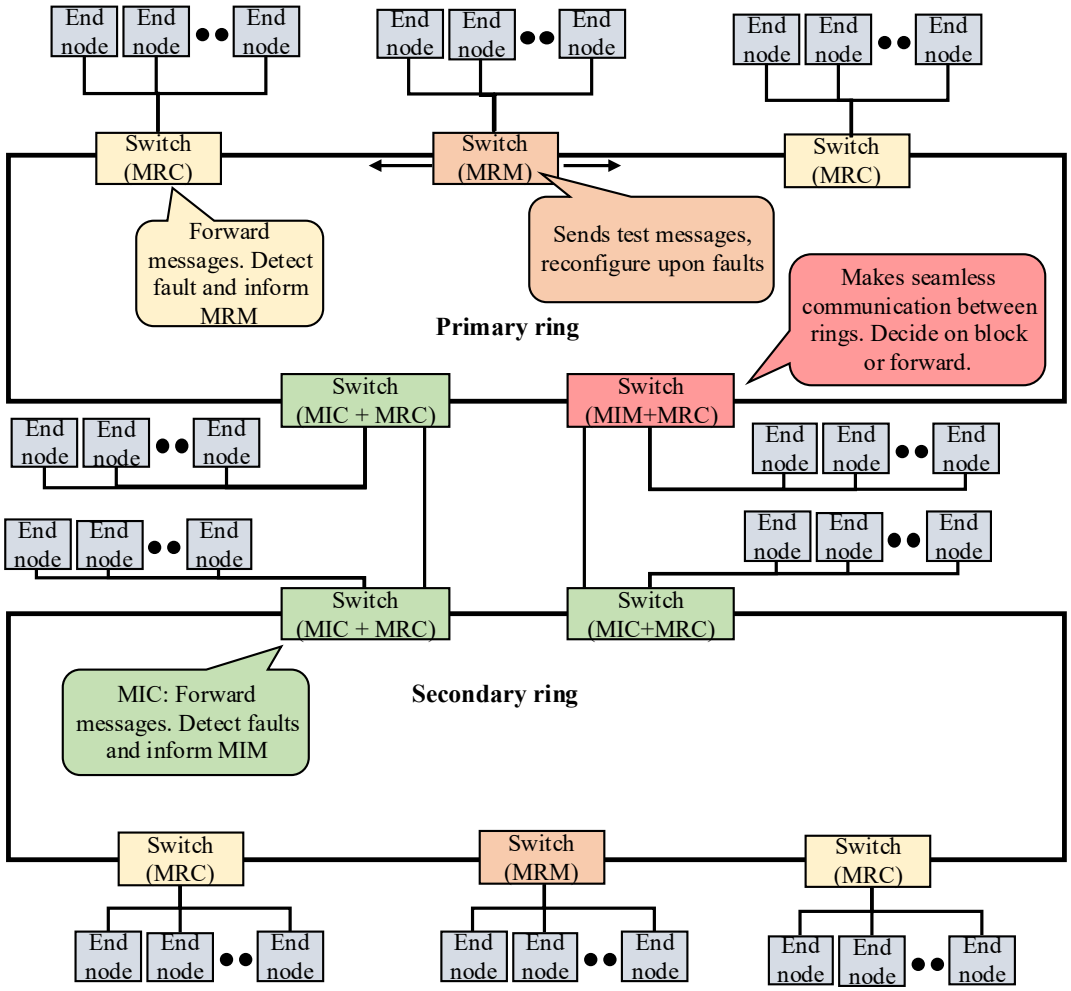


Fig. 20. Key functions of the MRP protocol functions implemented in switches.

A simplified version of MRP and the embedded functionalities that manage a redundant ring network are shown in Fig. 20, developed with inspiration from IEC 62439-2 (2021). All functions are implemented in switches within the ring network that have these capabilities.

- MRM ("integrity test initiator and manager") is the media redundancy manager of a specific ring network that sends test messages in both directions to check for network integrity for faults. Faults are reported back to the MRM by MRCs.
- MRC ("checking and reporter") is the Media Redundancy Client that forwards data, monitors the network, and informs MRM about fault issues.
- MIM is the Media Redundancy Interconnection Manager ("Interconnector checking and reporter") that connects the networks and forwards test messages to determine the state of the connection. Other switches do not overtake the MIM's function if the MIM fails, and MIM failure must be alarmed and followed up with repair or replacement.
- MIC is the Media Redundancy Interconnection Client ("MRC for the interconnection") that monitors the status and forwards regular data traffic between the rings. Any faults are reported to the MIM.

Switches with two roles, e.g., MIC + MRC, have two roles, one in each (ring) network they connect to.

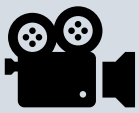
2.5 Fieldbus and IE protocols

The following sections present a few commonly used Fieldbus and IE communication families.

2.5.1 Profibus DP, Profibus PA, and Profinet

Profibus is a short name for "Process Field Bus", a protocol that was first developed by Siemens but, from 1989, was taken over by the Profibus User Organization (PNO) for further promotion and development. The earliest version of the protocol was called Profibus-FMS, but it was later superseded by Profibus DP (1993) and Profibus PA (1995). Profibus DP and Profibus PA implement layers 1, 2, and 7 of the OSI model and are categorized as Fieldbuses.

In the early 2000s, Profibus protocol was developed for IE under the name Profinet . All three Profibus variants can be combined with ProfiSafe, a profile that has additional mechanisms added to the messages between safety-critical systems and devices. Despite Profibus DP and PA being older technologies, they are still in use worldwide. Changing to new technologies requires upgrading field instruments and controllers with new communication interfaces and, sometimes, new cabling. This becomes too costly for many facilities, considering also the downtime for installation and integration. However, as of 2026, it is reported that Profibus DP and PA rely on components (e.g., drivers) that are no longer supported by sub-suppliers.

	Three videos on Profibus and Profinet are: • 1: What is Profibus and how does it differ from Profibus DP: • 2: What is Profibus in layman's terms: • 3: What is the difference between Profibus and Profinet? A summary of the main properties of the protocols are also found at Profinet University.

2.5.1.1 Profibus DP

Profibus DP (Decentralized Peripherals) is a Fieldbus where a central DP master cyclically polls distributed slaves for I/O data. The logical access is master/slave with token passing among masters; slaves respond only to master requests.

Profibus DP has two profiles:

- CP-V0: Real-time, cyclic I/O exchange
- DP-V1: acyclic services for parameterization and diagnostics.

The typical master device is a controller, such as a PLC or DCS, while slaves can be sensors or transmitters, motor drives, remote I/O cards, and valves. An example of a network involving Profibus-DP is shown in the upper part of Fig. 21.

A Profibus DP network can be split into segments, each with up to 32 devices, with a network-wide limit of 126 devices (due to address limitations). The segments are connected via either:

- Repeaters (connect one segment to another)
- Hubs (connect several segments)
- Optical link modules

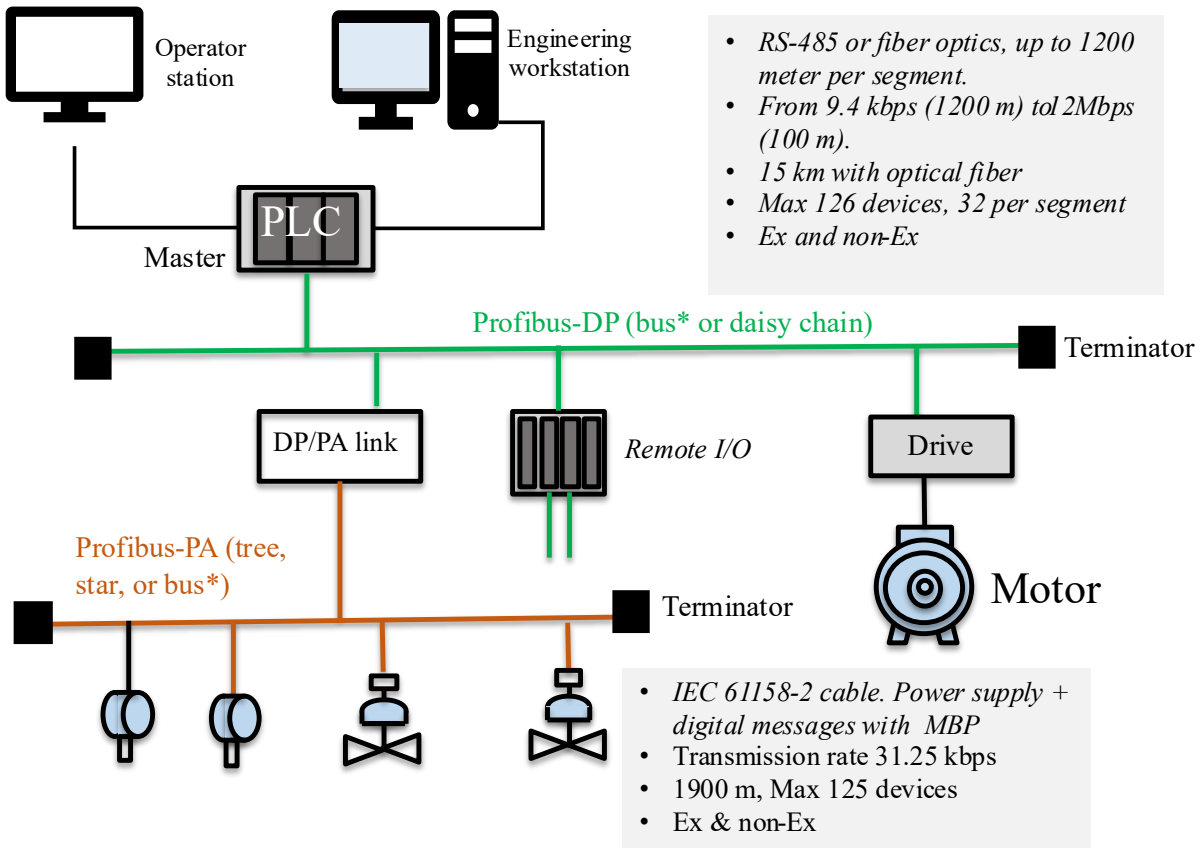
Each device connecting to the network will require its own wired power supply, as Profibus DP cannot transmit power along with digital messages.

The physical layer of Profibus DP uses RS-485 or optical fiber.

- The RS-485 cable length can range from 100 meters to 1200 meters, and with repeaters, the total reach can be further increased. Longer cables give lower transmission speeds: 9.6 kbps for a 1200-meter cable and 12 Mbps for a 100-meter cable.
- The transmission distance can be extended further with an optical cable, depending on the fiber type. Some sources mention up to 15 km.

The network can enter hazardous areas, i.e., areas with explosive atmospheres. In this case, the connected slaves and linking devices must be approved for such an environment, such as intrinsically safe devices using energy-

restricted methods (a topic covered in Chapter 13). Calculations are needed to ensure that the energy of the generated sparks remains below the threshold required to ignite the gas or dust in question.



* *Bus requires a terminator as shown*

Fig. 21. Profibus DP co-existing with Profibus PA

ProfiSafe is an additional application layer that can operate on top of what Profibus (and Profinet) implement for the OSI model's application layer, including Profibus DP. This extra layer adds mechanisms for fault detection and fault management as needed for safety-critical communication. ProfiSafe is explained in more detail in a separate sub-chapter.

2.5.1.2 Profibus PA

Profibus PA (Process Automation) is, as the name suggests, a Fieldbus specifically designed for process automation, where real-time requirements are typically less demanding than in manufacturing applications such as machinery and robotics.

Some notable differences between Profibus DP and PA are, e.g., that:

- It uses the same Profibus protocol as Profibus DP, but with a different physical layer, where power supply to the devices is also provided along with the digital messages.
- The Profibus PA network does not have a master, and must therefore connect with a Profibus DP network via a Profibus DP network where the master is located, via a DP/PA link (or coupler), as shown in Fig. 21.

Cyclic I/O is still the norm, with acyclic parameter/diagnostic services as needed. Some characteristics of the Profibus PA network are:

- Number of connected devices: Depending on the version of the DP/PA link, the PA side is typically 31 with one segment.
    - By adding more DP/PA links or couplers, the network can be extended with several segments, allowing a total of 125 PA devices. This total is constrained by the number of addresses that can be assigned.
    - If the network extends into hazardous areas, meaning areas with explosive gases, the number of devices must be determined through careful calculations to ensure energy levels remain sufficiently low. Some sources suggest a maximum of 10 devices in this case.
- Network topology: Profibus PA is primarily designed for bus communication, but allows point-to-point communication (in practice, start the architecture from the DP/PA link) between the controller and the field device. Ring and start may be achieved using couplers and repeaters.
- Cable lengths:
    - The maximum cable length of one segment is 1900 meters. In hazardous areas, the maximum cable length may be reduced to 1000 meters.
- Cable type:
    - Profibus PA applies a two-wire cable specified in IEC 61158-2 with Manchester coding and bus power (MBP) with synchronization and deterministic timing.
    - The transmission speed of 32.25 kbps is fixed and not dependent on cable length. Ensuring synchronization for deterministic timing.
    - MBP modulates the signal using minor voltage alterations (small enough not to disturb the current signal) at two different frequencies to represent "0" and "1", as shown in Fig. 22. The data bits ("0" and "1") are converted to these two frequencies by using a clock signal to detect when the bit is 0 or 1. There are two variants of this conversion: Manchester per G.E. Thomas and Manchester per IEEE 802.3.

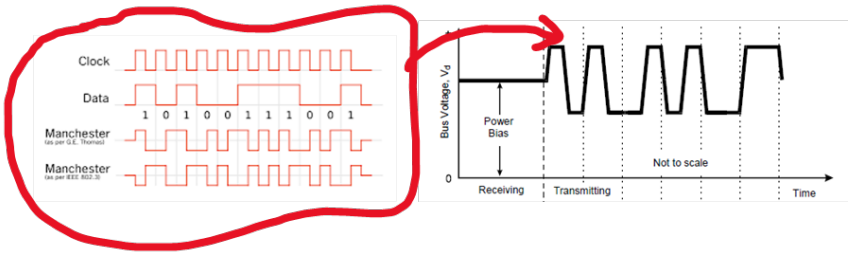


Fig. 22. Manchester coded bus power ensuring synchronization (Wikipedia)

ProfiSafe can act as an extra application layer on top of Profibus PA, as for Profibus DP.

2.5.1.3 Profinet

Profinet is an abbreviation for Process Field Net and is the Profibus version for IE. The network can be configured as a bus, tree, ring, or star and has no specific limit on the number of devices that can be connected.

Profinet has three different communication classes, where the realization is made to fit three different purposes:

- **Profinet RT** (real-time) for time-critical data exchange, which combines synchronous (cyclic) communication for real-time data with asynchronous communication for diagnostics and events.
- **Profinet IRT** (isochronous real-time) is also for time-critical data exchange, but implements synchronization and deterministic scheduling that are often needed for robotics and motion control.
- **Profinet NRT** (non-real-time) for other types of data exchange, such as, e.g., diagnostic checks and device parameterization.

Unlike most IE protocols, Profinet may operate without implementing layers 3 and 4. More specifically, and as shown in Fig. 23:

- Profinet NRT implements layers 1, 2, 3, 4, and 7 (with TCP/IP and UDP/IP)

- Profinet RT and Profinet IRT implement only layers 1, 2, and 7, as Fieldbus does, but with MAC addresses (as needed for Ethernet) directly exchanged between the application layer and the data link layer.
- The main advantage of Profinet (RT and IRT) is its significantly higher speed compared to Profibus DP and Profibus PA, ranging from 100 Mbps (4 wires/2 pairs) to 1 Gbps (8 wires/4 pairs). The downside is the limited cable length to 100 meters, unless fiber optics is used. This limitation, however, is being addressed by the introduction of Ethernet APL to the market. With Ethernet APL, the power supply and signals can also be transmitted over the same (Ethernet) two-wired cable 10 Mbps with 2-wire Ethernet, referred to as 10BASE-T1L (IEEE 802.3cg).

The use of devices and controllers with Profinet interfaces simplifies hardware integration into the network topology. Using the control system's engineering software, such as Siemens TIA Portal and ABB's Control Builder, the device's pre-built configuration can be imported as a Generic Station Description Markup Language (GSDML) file and permanently integrated into the hardware catalog. Drag-and-drop can be used to add devices to the network, and their configuration may be altered as needed. This type of configuration upload formats is unique to Profinet, but *some* other protocols provide similar possibilities with other file formats like Electronic data sheets (EDD) with Ethernet /IP and IO-Link Device Description (IODD) files with IO-Link.

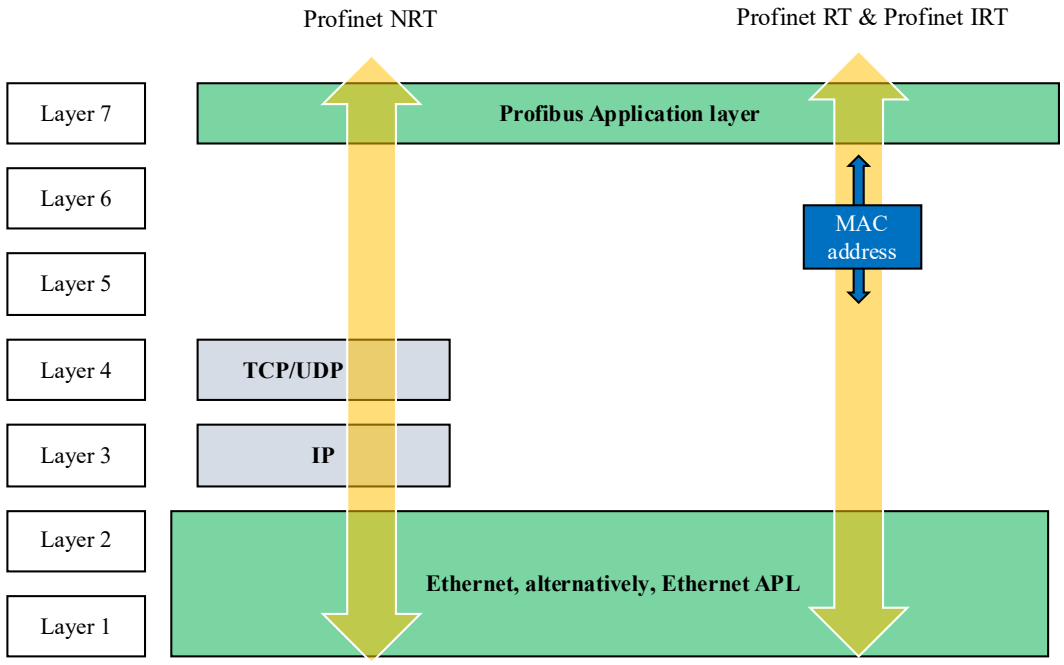


Fig. 23. Implementation of OSI layers for Profinet

When Profinet is mentioned without any qualifier, it typically refers to Profinet RT. Two related terms, not to be confused with the communication classes, are:

- Profinet IO devices: Field devices that implement Profinet, e.g., Profinet RT
- Profinet IO controllers: Controllers that implement Profinet, e.g., Profinet RT

Profinet can be combined with application profiles, serving as an additional layer on top of the already existing application layer of Profinet (i.e., above layer 7). Examples include:

- ProfiSafe (see separate sub-chapter)
- ProfiDrive: Adding specific commands for motion control (of drives) and applying Profinet RT or IRT as appropriate.

Fig. 24 illustrates a Profinet network that co-exists with Profibus DP and Profibus PA. Profinet is typically used for communication between:

- Controllers and field devices, with or without remote I/O cards (marked 1)
- Controllers (marked 2)
- Between controllers and operator stations (marked 3)

It is indicated that some networks may be redundant (stippled line) to enhance availability.

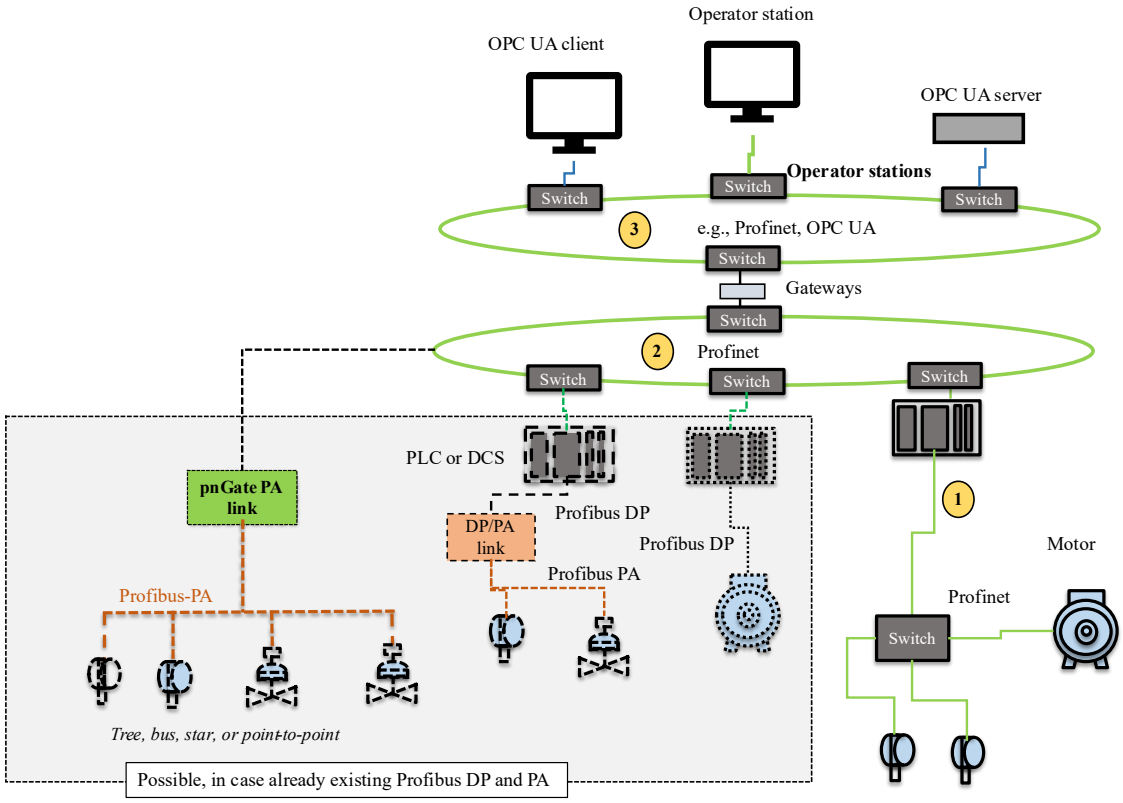


Fig. 24. Profibus and Profinet environment

Profinet is increasingly replacing Profibus DP. Profinet can also connect directly to the Profibus PA network using a linking device called pnGate PA. This option can be useful if upgrading parts of the network topology, but not all the way out to the field devices.

2.5.1.4 ProfiSafe

ProfiSafe is an additional application-layer profile on top of the Profibus and Profinet application layers specified in IEC 61784-3-3 (2021). ProfiSafe is a black-channel approach to implementing safe communication, where all checks and fault handling are performed by the communicating devices, so that it does not matter what is in between, meaning the type of cable, wireless, or optical connection. In this way, the same cable can be used for safety as well as non-safety (e.g., regular control), as illustrated in Fig. 25 , however, with priorities given to the safety-critical ones. ProfiSafe has been certified for SIL 3, meaning that it can be used in safety systems that perform functions with a safety integrity level (SIL) 3 requirement according to IEC 61508 (2010). SIL and IEC 61508 are topics of the Compendium Chapter 9.

ProfiSafe messages, marked “F” (F – functional safety/fail-safe) can be sent over the same network as standard Profisafe or Profinet (S) messages, as illustrated in Fig. 26. The F messages have a few additional bytes, the same ones for Profibus as well as for Profinet, explained in detail below.

The payload data (data exchanged between the safety devices) is in a format similar to standard messages, except that its size is reduced due to the extra safety-related bytes. The extra bytes related to safety (F) messages include:

- Status/Control (S/C): 1 byte. Status indicates a response message from the slave to the master, while control indicates that a command is being sent.
- A cyclic redundancy code (CRC): 2 or 4 bytes. The CRC is calculated by the device that sends the ProfiSafe message, and the receiving device recalculates it and compares it to the received value. Note that the CRC is not effective at protecting against unauthorized manipulation during a cybersecurity attack, since the attacker can reveal the CRC calculation algorithm.

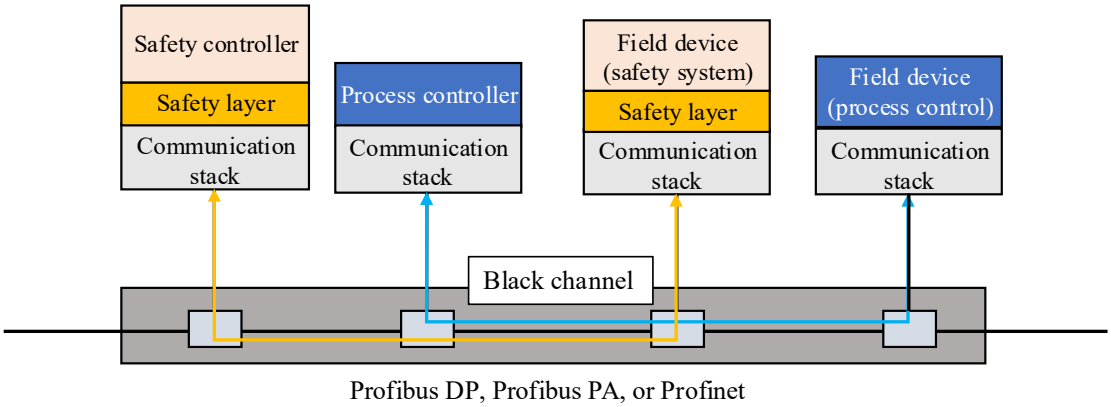


Fig. 25. How ProfiSafe co-exist in networks using Profibus or Profinet

Additional information, indirectly available as they are needed to calculate the CRC, is:

- Virtual Consecutive Number (VCN): VCN is a counter that increments each communication cycle to ensure the correct sequence. As a consequence, the receiving device detects a loss or duplication.
- Codename sender. A unique identifier for the sender that ensures the message comes from the correct source
- Codename receiver: A unique identifier for the sender ensures the message is received at the correct destination.

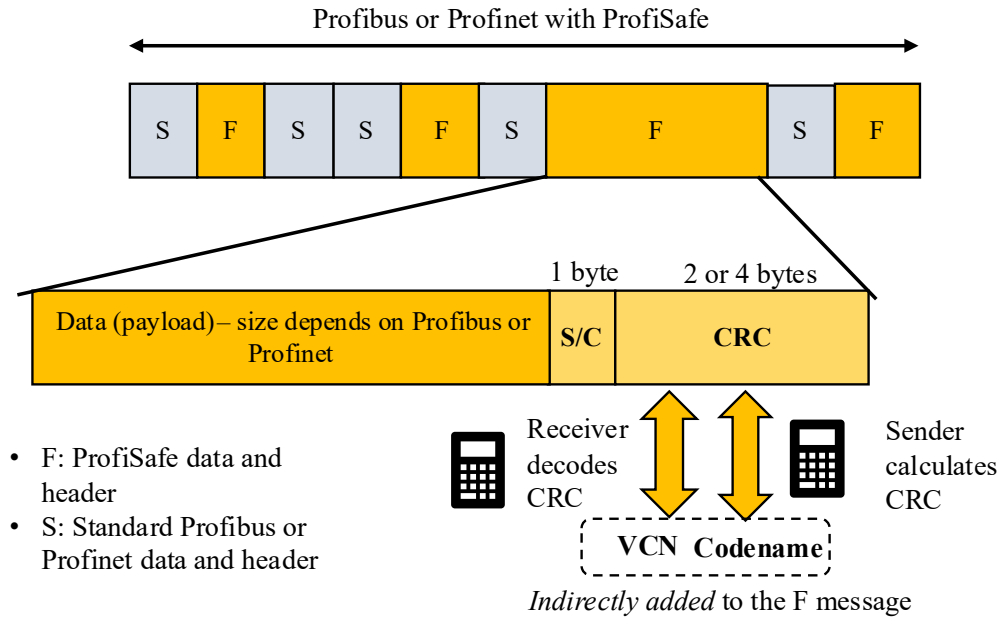


Fig. 26. ProfiSafe message structure

The fault types that can be detected with ProfiSafe are shown in Tab. 4.

Tab. 4. Fault types detected by the ProfiSafe safety layer.

ProfiSafe Fault types	Response	Detection mechanism			
		(Virtual) consecutive number (counter)	Timeout with receipt (watchdog waiting for valid message)	Codename (sender and receiver)	CRC check
Repetition	Immediate rejection	X			
Deletion/loss: Hole or part of the message lost	Keeps the last good data*	X	X		
Insertion: Extra message introduced	Immediate rejection	X	X (waits for the valid)	X	
Resequencing/ Incorrect sequence	Immediate rejection	X			
Data corruption	Immediate rejection				X
Delay	Keeps the last good data*		X		
Masquerating (standard message mimic fail/safe)	Immediate rejection		X	X	X
Revolving memory failure: Order of messages changed in a device's memory	Immediate rejection	X			

Messages that the receiver immediately rejects will immediately lead to a fail-safe state; for example, an automatic stop is triggered, as illustrated in Fig. 27. ProfiSafe has no mechanisms to correct any of these failures. The sender will never know whether the message was rejected, as there is no confirmation (such as a receipt).

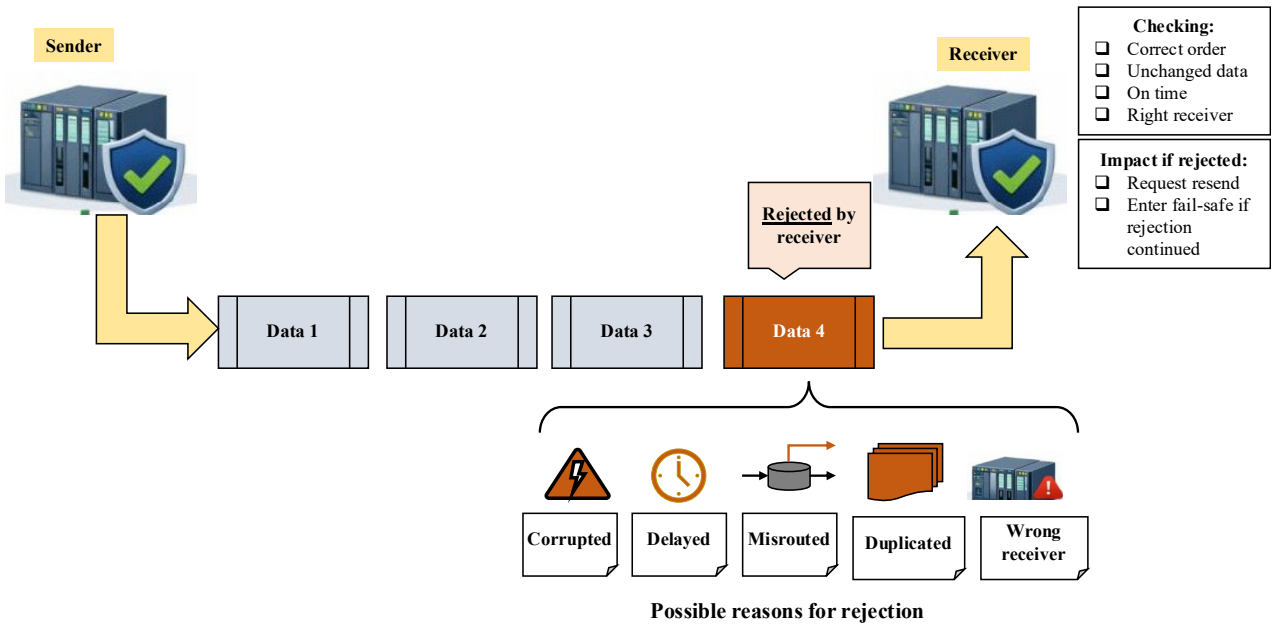


Fig. 27. Illustration of ProfiSafe error checking and response (Adapted from P. Lukas/Siemens)

Messages that are delayed, fully lost, or partially deleted will cause the receiver to keep the last known data, but only for a certain time determined by a watchdog. Sources on the internet suggest that the configuration of the watchdog time will depend on the application area, for example:

- High-speed/high-risk machinery and robotics: 20-50 ms
- Standard industrial applications: 100-200 ms
- Process control: up to 500 ms

2.5.2 Modbus RTU and Modbus TCP

Modbus was originally a Fieldbus introduced by Modicon (later Schneider Electric) in 1979 and has stayed popular since then. One explanation is that it was among the first to be shared openly, and the other is that it has a simple structure that relies on master-slave (serial-type) communication.

There are three main profiles for this protocol, all being asynchronous:

- Modbus RTU Fieldbus
- Modbus ASCII as Fieldbus (not covered here, as it seems to be seldom used)
- Modbus TCP as IE

	Two YouTube videos on What is Modbus and does it work are: https://youtu.be/JBGaInI-TG4 and https://youtu.be/txi2p5_OjKU.

Modbus RTU is based on a master-slave concept. The master can communicate with the slaves in two separate ways:

1. Unicast mode (asking for something), where the master sends a request to *one* slave. The transmission carries two messages: the master's request and the slave's response.
2. Broadcast mode (just telling something): In this mode, the master sends a request to all the slaves, but the slaves do not respond.

Messages are sent using a request-response approach. As illustrated in Fig. 28, the data (payload) is wrapped into a message consisting of:

- Address field, in the range of 247 available addresses
- Function code, defining the operation as read, write, or some other predefined function.
- CRC: Cyclic redundancy check, based on arithmetic operations of the message content

The messages are decomposed into what Modbus calls 11-bit character frames, which are sent bit by bit over the wires.

- Each packet is separated by a fixed time delay (silent interval) corresponding to how long it takes to send 3.5 characters (one character = 11 bits), considering the transmission speed. With a transmission rate of 9.6 kBps, it means around 4 ms.
- Each frame starts by a start bit (1) and a stop bit (1)
- The second last bit is added for error checking purposes. Modbus can be configured to either require even, odd, or not set number of “1”s in a character frame. The parity bit is chosen to achieve this. In case of no parity, two stop bits are applied.

Transmission speed or rate is typically:

- RS-485 (two or three wires): 9.6kbps –19.2 kbps over a distance of a maximum of 1200 meters
- RS-232 (three wires): 9.6kbps –19.2 kbps over up to 15 meters
- Optical cable: 9.6 kbps - 115 kbps over 3-4 km.

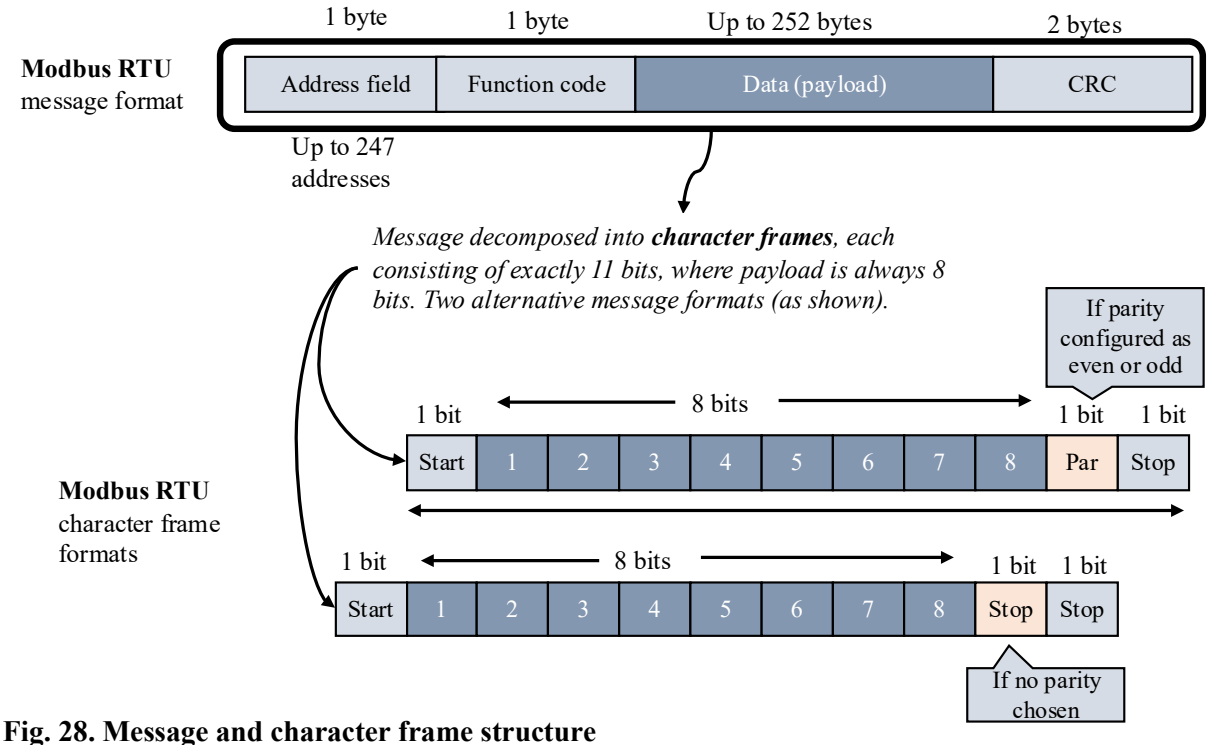


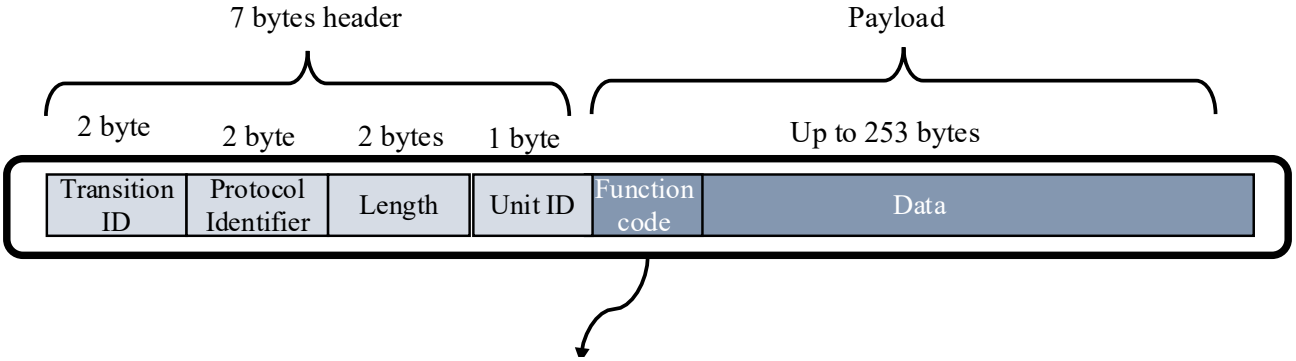
Fig. 28. Message and character frame structure

Modbus TCP is the IEEE version of Modbus, and it implements layers 1, 2, 3, 4, and 7 of the OSI model. Compared to Modbus RTU, the following similarities and differences are notable, some of which are illustrated in Fig. 29:

- The data exchange continues in a request-response manner, organized similarly to master-slave, but with TCP/IP, it is referred to as Client-Server, where the client makes requests, and the server responds.
- The message structure remains similar to Modbus RTU (for TCP, it is 260 bytes, while for RTU it is 256 bytes)
- The message header is much longer, as it consists of 7 bytes (56 bits):
    - Transaction ID (2 bytes), a unique serial number generated by the m
    - Protocol Identifier (2 bytes), always set to 00 00 for standard Modbus. It acts as a confirmation that the message *is* Modbus. It is also mentioned for possible future usages, for example to identify the need for encryption or extension of addresses memory addresses inside the device that buffers up the received data.
    - Length (2 bytes), which identifies how many bytes are left in the message
    - Unit ID (1 byte), which can be used to set the slave address, if the device requested is behind a gateway in a Modbus RTU network. Not used (and set to 00 or FF) when an IP address is generated.
- The remaining message is the payload, and consists of:
    - Function code is part of the data payload of the message and identifies (with 1 byte) if the function is, e.g., to read or write.
    - Data to be exchanged (variable length)
- The concept of character frame disappears, as decomposing of message into smaller segments (layer 4) frames (layer 3) and packets (layer 2) following the Ethernet realizations of layers 1-4. It means that routing and addressing use IP (Internet Protocol) at the network level and MAC (Media Access Control) at the data link level of the OSI model.
- CRC is not part of the message, as the Ethernet network layer automatically handles packet checking and error corrections.
- The protocol allows limited diagnostic data exchange
- It is not possible to add a safety layer to the protocol (as ProfiSafe for Profinet)

-

Modbus TCP message format



Message decomposed as specified with Ethernet realization of layers 1-4, with segments (layer 4), frames (layer 3) and packets (layer 2). Addressing with IP (layer 3) and MAC (layer 2).

Fig. 29. Modbus TCP message format

- The transmission speed is much higher than for Modbus RTU, with 100 Mbps (Fast Ethernet, four wires) -1 Gbps (Gigabit Ethernet, 8 wires) for cable lengths of up to 100 meters, depending on the number of wires.
- With Ethernet APL, the cable length can be extended, but the speed may be reduced to 10 Mbps with 2-wire Ethernet (referred to as 10BASE-T1L (IEEE 802.3cg).

Modbus TCP is sometimes preferred over Profinet for its simplicity, wide adoption in products, use of a standard TCP/IP stack, and lower implementation cost (for example, switches can be simpler and less expensive).

2.5.3 HART, HART IP, and wireless HART

HART stands for Highway Addressable Remote Transducer and is a protocol developed in the 1980s by Rosemount, a supplier of process automation equipment. Many field sensors had inbuilt diagnostics and reconfiguration interfaces that were not accessible via analog signal transmission. HART offered a solution to this problem by using a protocol that sent serial messages over the same wires, superimposed on the analog signals, with an amplitude sufficiently low to avoid disturbance.

The protocol supports master-slave communication via two-wire serial transmission. The master instructs (by asking) one or more slaves to send specified data, such as measured values, until the slaves are told to quit.

HART has three profiles or variants:

- HART (regular) as a Fieldbus
- HART IP for industrial Ethernet (IE)
- Wireless HART following the WirelessHART standard

HART (regular) uses start and stop bits, and messages are sent only when the master requests them. Because the master does not use fixed cyclic polling, the messages are event-driven and therefore asynchronous. Wireless HART, in contrast, uses time-synchronized message exchange with a network clock distributed by the gateway. As HART IP is an IE protocol, it combines synchronous (cyclic) communication for real-time data and asynchronous data for diagnostics and events.

Regular HART allows the following network topologies:

- The multidrop bus topology has a maximum of 64 devices connected.

- The point-to-point communication that allows HART messages to be superimposed on top of a 4-20 mA signal transmission. In this way, analog measurements can be used for real-time control, while HART messages can be used to share measurements, diagnostic status, and configuration settings for monitoring. HART can also be used to configure settings on field devices.

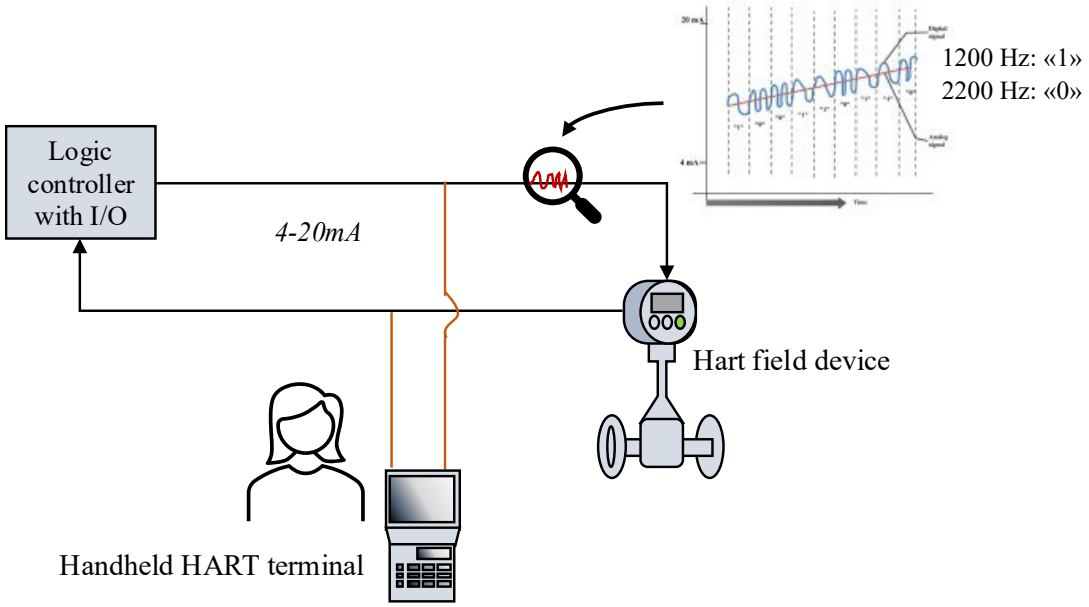


Fig. 30. HART messages superimposed on the 4-20 mA measurement signal.

Fig. 30 illustrates how the HART messages are superimposed on the 4-20 mA measurement signal. HART message transmission for Fieldbus is based on the Bell 202 modem standard, in which "1" and "0" are represented by different frequencies using Frequency-Shift Keying (FSK). The amplitude of the modulated signal is so small that it does not disturb the current signal.

Regular HART has a 1.2 kbits/sec transmission rate, the lowest among Fieldbus types, even compared with Profibus PA. This is why HART, as a fieldbus, is not used for real-time control but only to transport non-critical information, such as diagnostic data and data from sensors used for condition monitoring. This is done by "taping off" digital messages using a handheld terminal, as shown in the figure, or via a more permanent solution in which a computer or server receives them. In this way, the same data used for control can be used for training, diagnostics, and configuration. Remote configuration via the terminals is also possible.

HART-IP is the IE variant for HART, published in 2007. This variant is for digital messages only and not for superimposing on other signals. With IE, the transmission speed can range from 100 Mbps to 1 Gbps over distances up to 100 meters. With Ethernet APL, the same speed is maintained over longer distances.

HART-IP extends the conventional HART protocol by encapsulating standard HART commands within TCP/IP, enabling communication over Ethernet networks. While it builds on the same command structure as traditional HART, it replaces the analog 4–20 mA physical layer with a fully digital, network-based approach.

HART-IP is primarily used in process automation environments, where it supports applications such as device configuration, diagnostics, and asset management. The protocol does not provide strict real-time determinism and is therefore less suited for time-critical applications such as motion control in manufacturing or robotics. Instead, its performance is adequate for typical process control tasks, where response times in the order of tens of milliseconds are acceptable.

WirelessHART includes HART access points and devices with a WirelessHART interface and antennas, as shown in Fig. 31. The wireless network can connect devices in a mesh-like structure, making it more flexible and reliable. The specification of Wireless HART is provided in IEC 62591 (2016) and is based on regular HART, not HART IP.

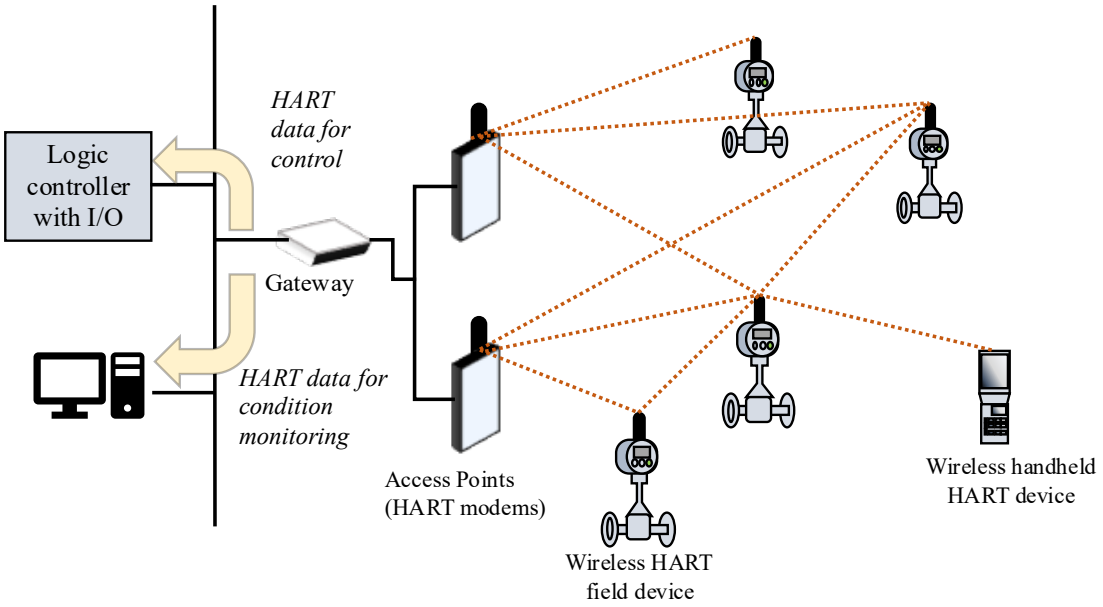


Fig. 31. Wireless HART network

WirelessHART, like many wireless industrial communication standards, is based on standardized (wireless) communication technologies. High-bandwidth applications often rely on IEEE 802.11 (Wi-Fi), whereas low-power field devices commonly use IEEE 802.15.4. WirelessHART applies IEEE 802.15.4 along with the higher-OSI-layer functionalities of the regular HART protocol. HART-IP can also be deployed in wireless environments; while it is most commonly run over high-speed Wi-Fi networks, it is an application layer protocol that can utilize any underlying media capable of transporting standard TCP/IP traffic.

Another wireless standard applied in the process industry, as an alternative to WirelessHART, is ISA-100.11a. ISA-100.11a was developed by the International Society of Automation and builds on the same IEEE standards mentioned above. Both standards are highly applicable for the process industry sector when considering network topologies and response-time requirements. However, unlike WirelessHART, ISA-100.11a features a flexible application layer that can also transport other protocols, such as Modbus, Profibus, and Foudnation Fieldbus, over the same wireless infrastructure.

2.5.4 IO-Link

IO-Link is a point-to-point, bi-directional communication standard between an IO-Link master and field devices and can be used for Fieldbus as well as IE. It seems popular within manufacturing facilities, especially for machinery control, while less common in the process industry.

Communication Basics:

- Standard IO-Link uses 3 wires: power supply, ground, and communication.
- It supports digital communication for sensor/command data, diagnostics, and configuration.
- IO-Link uses an asynchronous, serial, start-stop byte framing method, like universal asynchronous receiver/transmitter, a hardware communication protocol for serial data transmission between devices.

Integration:

- IO-Link masters can be added to existing Fieldbus or Industrial Ethernet networks as long as the master has the appropriate interface.

- Each master typically offers 4, 8, or 16 ports, each connecting to one field device.
- Field devices must have an IO-Link interface.

An example of how the IO-Link master connects to a Fieldbus or IE and is a hub for connecting field devices is shown in Fig. 32. A network can add several IO-Link masters.

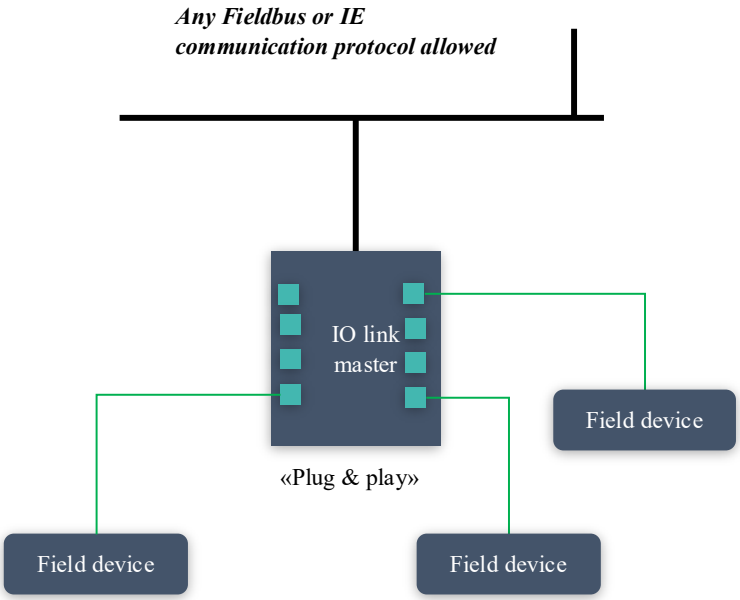


Fig. 32. IO-Link master with connected devices

Realizations: IO-Link enables plug-and-play integration of sensors and actuators. It supports two types of communication:

- Digital communication (full IO-Link protocol)
- Switching signals (SIO) for simple on/off control

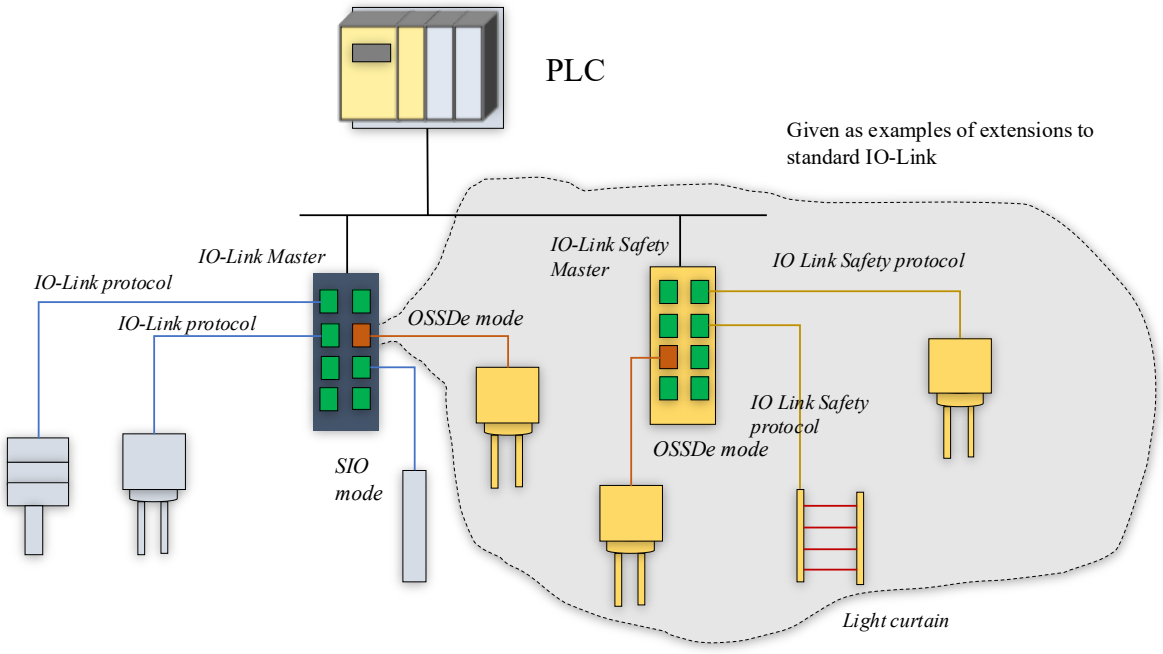


Fig. 33. IO-Link network extensions

Extensions: The following extensions seem to be available (but the status of commercial use is a bit unclear):

- IO-Link Safety: Adds redundancy for safety-critical applications. Requires additional pins (typically 5, with 1 spare).
- IO-Link Wireless: Under development for wireless connectivity in harsh environments.
- Safety Variant ("OSSDe"): IO-Link Master and IO-Link Safety Master can handle a variant of SIO for safety named Output Switching Signal Device (OSSDe) mode. It can be applied for the standard IO-master as well as with the IO-Link master for safety.

Examples of how an IO-Link master for safety, as well as OSSDe as an option for safety communication, may be added to the network topology are shown in Fig. 33. Communication with the SIO and IO-Link protocols is also illustrated.

2.5.5 To what extent is hardwired safer than the black channel?

There are different views in the industry on whether hardwired, meaning directly wired analog or binary signals, is safer than using functional safety protocols like ProfiSafe. Based on industry presentations, particularly one by Pavel Lukes from Siemens at the Norsk Forening for Elektro og Automasjon (NFEA) safety system conference, some interesting perspectives have been presented. The example focused on commands sent from a fire and gas detection (F&G) system to the emergency shutdown system (ESD), and an exchange is necessary because the ESD system's responsible for isolating ignition sources if a fire or gas has been detected. The current industry practice in Norway is to implement this communication with hardwired signals, and in the example, it was compared with a black channel (referred to as communicated). The example did not address the use of hardwired and communicated communication between field devices and controllers, and in the example, they were maintained as hardwired.

The comparison of hardwired and communicated was illustrated similarly to Fig. 34. Despite relying on hardwired signal transmission between safety controllers and between safety controllers and the field devices, there are several steps within the safety controller that rely on a black channel, for example, ProfiSafe, or another open or proprietary protocol. In this communication, both safety-certified (yellow) and non-safety-certified (white) items are allowed under the black channel concept. Here, the abbreviations applied are:

- AI: Analog input
- IM: Interfacing module
- CP: Communication processor
- DI: Digital input
- DO: Digital output
- SL: Safety layer (extra added for safety-checking when black channel is applied)
- CPU: Central processing unit

The aim of the comparison was to illustrate that even hardwired data exchange relies on a black channel, even if it is internal to the safety controllers. For the specific exchange between F&G and ESD, we note the following.

- The hardwired communication relies on an extra IM at both ends plus a relay, compared to the communicated alternative. The relay transfers the status of the F&G DO card, i.e., energizing or de-energizing the relay coil, to the relay contacts operated by the coil, determining the status read by the ESD DI card.
- The communicated transmission includes some software aspects implemented in the CPUs (marked by SL) to perform all the safety-related checking of the communication. Even if it looks much simpler in terms of the number of components involved, it adds some software complexity.

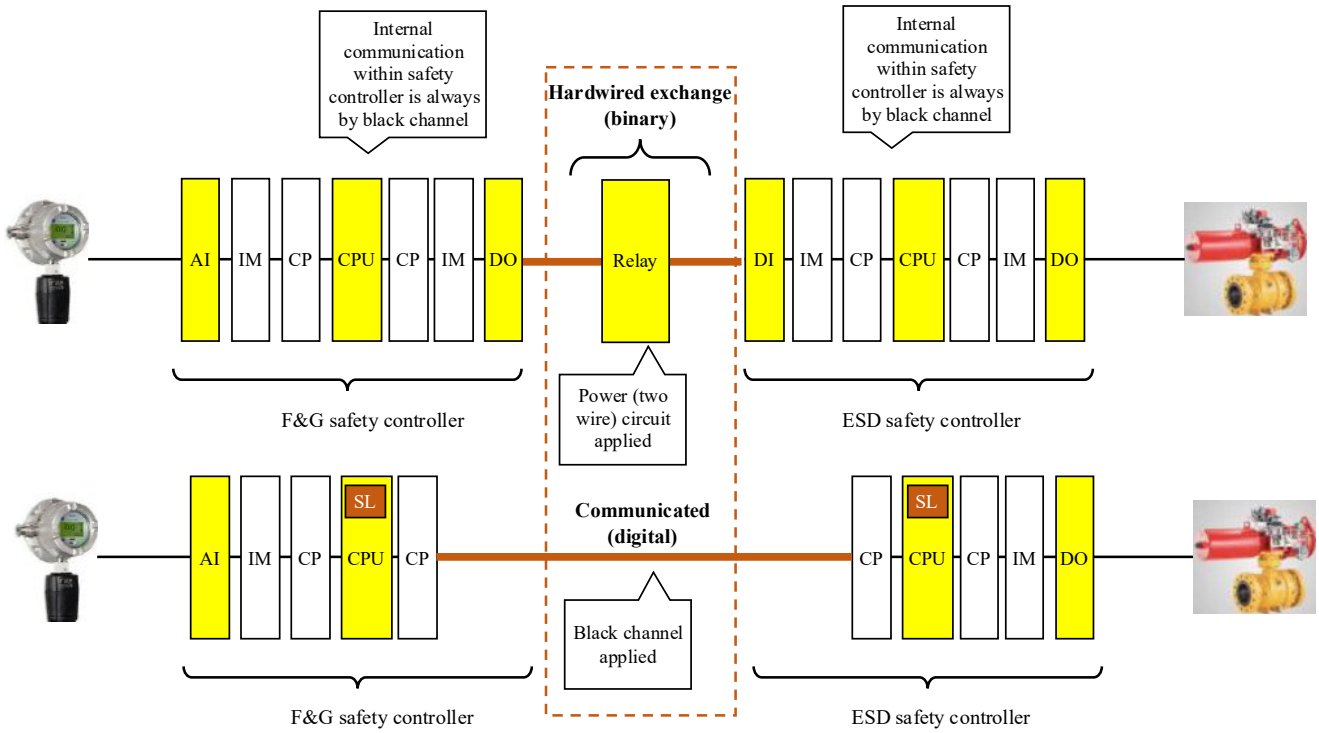


Fig. 34. Comparison between hardwired and black channel communication between two safety controllers (adapted from P. Lukas, Siemens).

2.6 IE with Ethernet APL

IE has, up to now, had two essential drawbacks, as seen in the process industry: for high-speed communication, it is suitable only for short distances, and power cannot be supplied to devices alongside digital messages, thereby requiring separate power cables. To close this gap, a new Ethernet cable that enables longer reach and power-over-signal capabilities, along with network devices (APL power switches and APL field switches), has been developed and is named the Ethernet Advanced Physical Layer (Ethernet-APL).

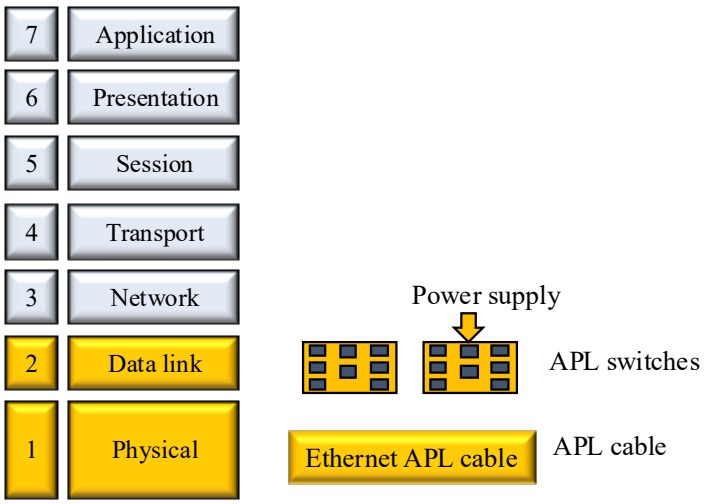


Fig. 35. Ethernet APL and its relation to other OSI

With reference to the OSI model, Ethernet APL is introducing new technology at level 1 and level 2 as illustrated in Fig. 35. Four industry associations have been involved in the development: FieldComm Group, ODVA, OPC Foundation, and Profibus PI, as well as several key suppliers of field instrumentation and control systems. The

webpage https://www.ethernet-apl.org shares the status of the initiative, and the report "Ethernet to the field " by FieldComm Group (2021) provides a nice overview of the solution.

BASF, a German company, has built an APL test center, as explained in an article in the Online Magazine Control. There, they are testing out how Profinet can be used with Ethernet APL, testing the integration with systems from several manufacturers like ABB, Emerson, Endress+Hauser, HIMA, Krone, Samson, Siemens, and Yokogawa.

Photo: BASF

Ethernet APL cable  Photo: Pepperl+Fuchs	A two-wire, loop-powered Ethernet cables that implement power over Ethernet (PoE) following the 10BASE-T1L specification in the IEEE standard IEEE 802.3cg-2019 (2019), where the number "10" indicates a data transmission rate of 10 Mbps in full-duplex mode. The full-duplex mode is ensured, despite transmitting simultaneously over two wires in opposite directions, as the signals superimpose rather than collide, creating a new signal pattern that each communicating device (receiving data) can decode by subtracting its own signal (of sent data). It is also possible to reuse the Profibus PA cable type IEC 61158-2, provided the connected field devices are replaced or upgraded with an Ethernet APL interface.
APL power switch (located in control cabinet) – Can be replaced by an APL rail field switch.	APL power switch: The "backbone" of the APL infrastructure that provides power to APL field switches when the APL field switches over a trunk-spurt topology, when the APL field devices are not provided with a separate power supply. It seems that many manufacturers skip developing this and instead mount an APL field switch with an external power supply. No product picture was therefore identified. Ex-zones are covered in Chapter 13 of the compendium.
Trunk-powered APL field switch. 	APL field switches are used to create the trunk-spur topology, distributing messages and power to individual field devices using the Ethernet 2-wire APL cable. The APL field switch can be connected directly to a (regular) Ethernet network or IE. The switches can have an external power supply or a power supply via the trunk. The field switches can be of different sizes. For example, Pepperl+Fuchs provides 8, 16, or 24 intrinsically safe (IS) spur ports that can be led into ex zone 1 and zone 0 (with IS barriers integrated), as well as in and out for trunk connections.

Photo: Pepperl+Fuchs (APL field switch)	APL field switches with external power supply, as well as trunk-powered APL switches, can be located in ex zone 2, if designed accordingly.

All field devices connecting to the Ethernet APL infrastructure must have an APL interface. Today (as of 2026), many manufacturers are expanding their product portfolios to include devices with this capability.

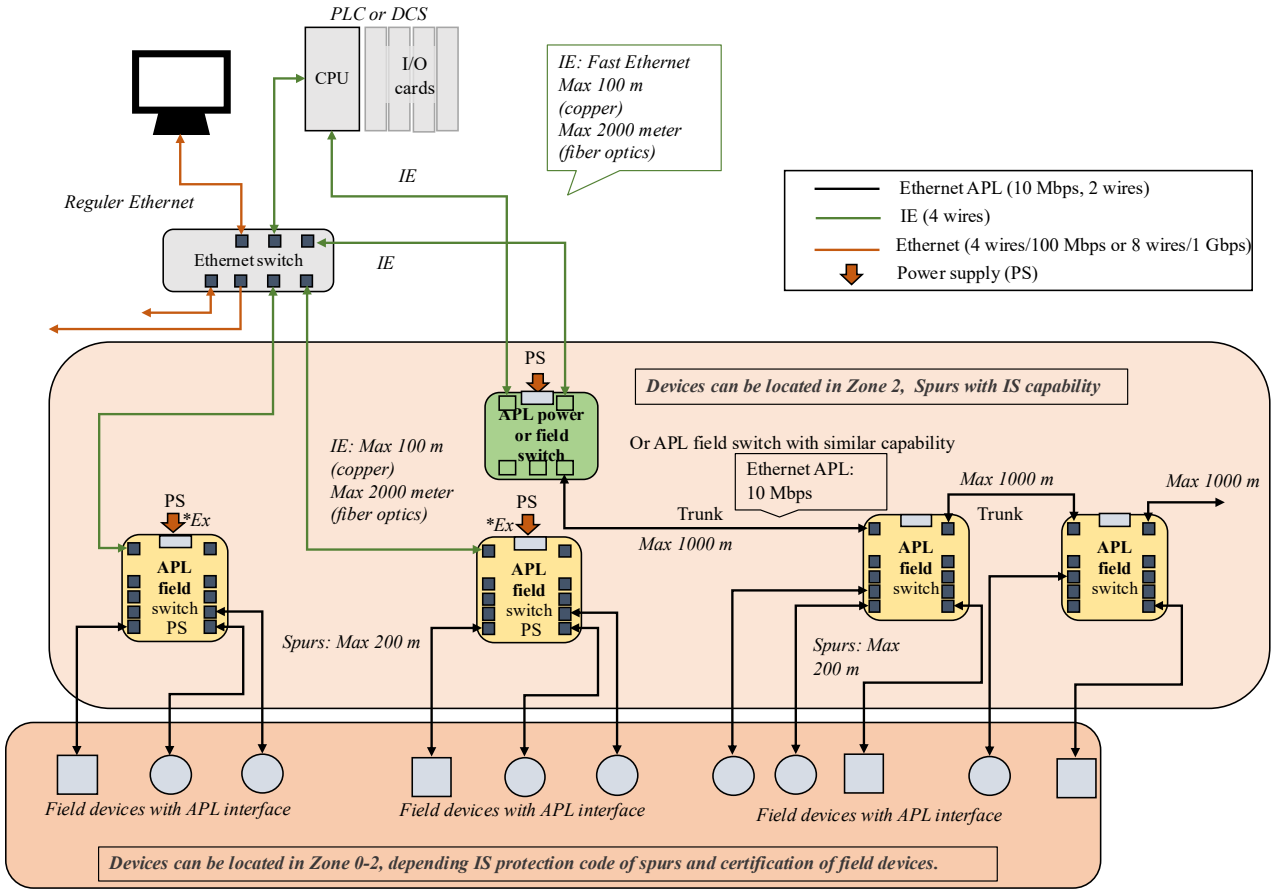


Fig. 36. Ethernet APL topology example

An Ethernet APL network is always an extension to an IE network, as illustrated in Fig. 36. If organized in a star topology, APL switches connect directly to the IE network, whereas an APL power switch is required when APL field switches are connected in a bus topology.

The APL cables are what are named spurs and trunks in Fig. 36. We notice that the bus network has a more extensive reach than APL field switches connected directly to IE networks, unless fiber optics is used.

Due to available ATEX certification, APL field switches can be installed in areas with explosive atmospheres where ATEX requirements apply (see Chapter 12 about Ex and ATEX). In this case, the cables are subject to additional requirements for robustness and fire resistance, and the field devices must have relevant Ex coding. With reference to Fig. 36, it would mean that:

- The trunk cables have Ex be certification, meaning the construction of terminals connecting the cables is designed to prevent the creation of sparks.
- The spur cables are of type WISE-2 (2-Wire Intrinsically Safe Ethernet), which provides energy restrictions that prevent ignition of explosive atmospheres and can connect Ex i-certified devices. Depending on the switch model, type, and configuration, the number of devices connected to a single field switch may be lower than in non-Ex zones.

APL field switches with separate power supplies (not via APL power switches) must ensure that the power supply connections have the necessary Ex coding.

Ethernet APL may be considered as a special case of single-pair Ethernet (SPE). While Ethernet APL focuses on the process industry sector and the possibility of exchanging data with field instruments in hazardous (ex) areas where energy restriction is important, Ethernet SPE covers many other, broader use cases where power over data line is desirable, such as in manufacturing and building automation. A key advantage of general-purpose Ethernet SPE is its lower hardware and deployment cost, as these systems bypass the expensive certifications required for hazardous areas. Furthermore, liberated from intrinsic safety constraints, general SPE can deliver higher power and voltages over the same data lines. Ethernet SPE can also achieve higher transmission rates with shorter trunk lengths, e.g., up to 500 meters.

	Endess+Hauser has made a nice visual presentation of Ethernet APL on YouTube.

2.7 Intelligent (smart) field devices

Intelligent (smart) field devices are devices capable of transferring digital data over Fieldbus or IE networks. The term "smart" refers to the functionality of microprocessors and digital communication interfaces, whether wired or wireless.

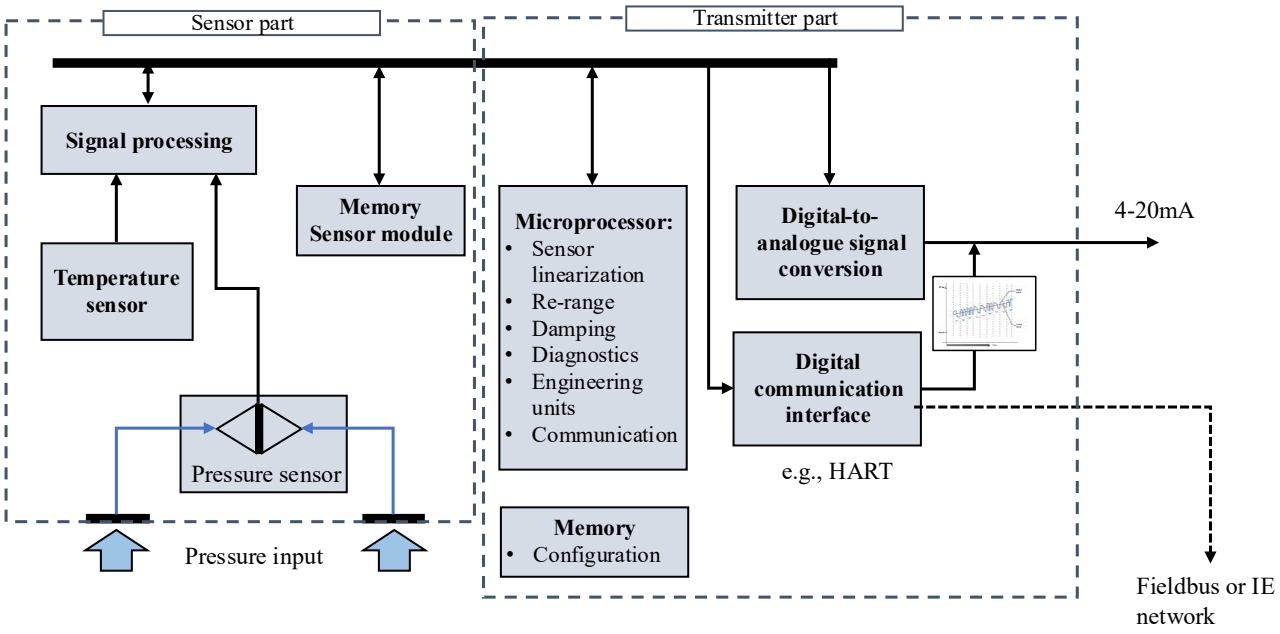


Fig. 37. General layout of a smart transmitter (Source: Exida FMEDA report for Rosemount 3051 pressure transmitter)

Fig. 37 illustrates the main components of a smart pressure transmitter. The sensor is in direct contact with the environment or process, such as pressurized gases, fluids, or temperature. The sensor exposure is converted into a current or voltage signal, which the transmitter, via the microcontroller, converts to a digital format. The microcontroller performs various self-checks (diagnostics) and auto-corrections, e.g., to compensate for zero-point drift. The Fieldbus (or IE) interface converts data from the microcontroller to the format of the chosen protocol. Most smart transmitters for the process industry also have digital-to-analog signal conversion for 4-20 mA. Many process facilities prefer this interface to the digital one for control and safety purposes, while digital communication interfaces are primarily used to harvest diagnostic data and update the configuration.

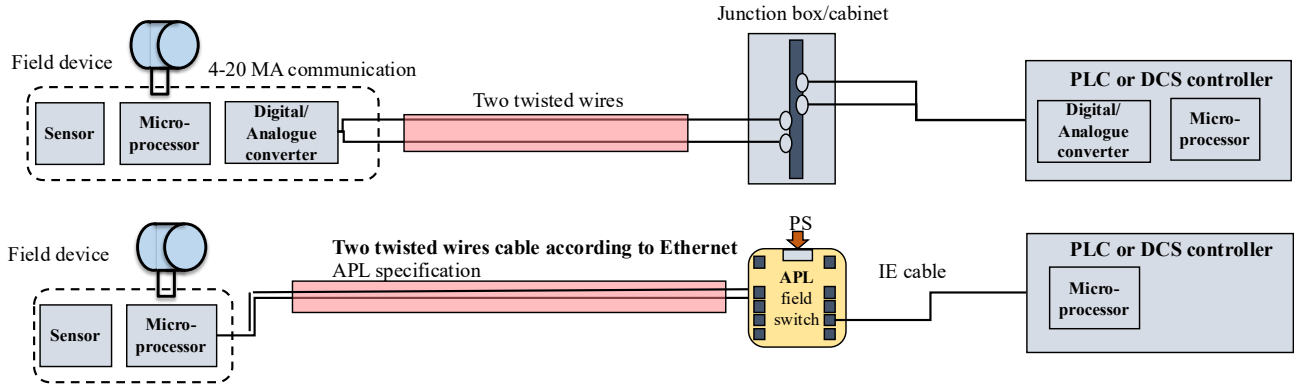


Fig. 38. Smart field devices – analog signal transmission vs Ethernet APL data transfer

Other examples of smart field devices include smoke detectors, optical gas detectors, optical flame detectors, and intelligent (smart) valves capable of trending valve movements, energy consumption, applied forces, and auto-correct settings to compensate for degradation and friction.

Fig. 38 compares traditional 4-20 mA for communication between a controller and a smart device with Ethernet APL. We notice that both use twisted-pair wiring, but they are built to different specifications. While 4-20 mA signals can be transmitted via several cables connected through junction boxes and cabinets, Ethernet is used for APL field switches that connect to an IE network.

2.8 Wireless communication

This compendium does not go into the details of wireless communication. However, it is worth mentioning some wireless standards, shown in Tab. 5, even though wireless is not extensively used in the manufacturing and processing industries compared to cabled solutions. With wireless communication, it may still be necessary to install separate power-supply cables for the field devices if batteries are not a suitable alternative.

Tab. 5 Overview of wireless standards and application areas

Wireless standard	Characteristics	Relevant for		
		Control	Safety	Condition-monitoring
ISA-100 Wireless (ISA100.11a)	Low-Rate Wireless Personal Area Network (250 kbps). Can be combined with several other protocols. Supports several topologies: mesh, star, and hybrid mesh/star.	(x) Limited (non-critical closed loop control, providing secondary process variables)	(x) Limited (e.g., Safety-related monitoring, not SIL-rated functions)	(x) Applicable, but costly infrastructure
Wireless HART based on IEEE 802.15.4:	Low-Rate Wireless Personal Area Network (250 kbps). Primarily designed for HART communication, integration with other systems occurs via gateways rather than by tunneling multiple protocols in the network. Supports only a full mesh topology. Every field device must be capable of routing data for its neighbors	(x) Limited (slow closed-loop control, like in the ABB lab, suitable for monitoring purposes)	(x) Limited (e.g., Safety-related monitoring, not SIL-rated functions)	X Well-suited

LoRaWAN (low-range wide-area network)	Low-Power Wide Area Network (0.3 kbps to 50 kbps). Dedicated to its own lightweight LoRaWAN protocol. It passes simple, compressed raw data payloads rather than tunneling industrial automation protocols. Supports only a star-of-stars topology. Field devices talk directly to long-range gateways; they cannot route or hop data through each other. High latency, limited downlink capacity, and duty-cycle constraints make it unsuitable for control	Not suitable	Not suitable	X Well-suited for low-rate, low-amount exchange needs (and long outreach)

2.9 OPC UA

OPC UA is an abbreviation for Open Platform Communications (OPC) for Unified Architecture (UA), a standard for the seamless organization and exchange of data. PC UA was developed as a platform-independent framework to enable interoperable, secure, and semantically rich data exchange across industrial systems, from field level to enterprise and cloud, as illustrated by examples in Fig. 39.

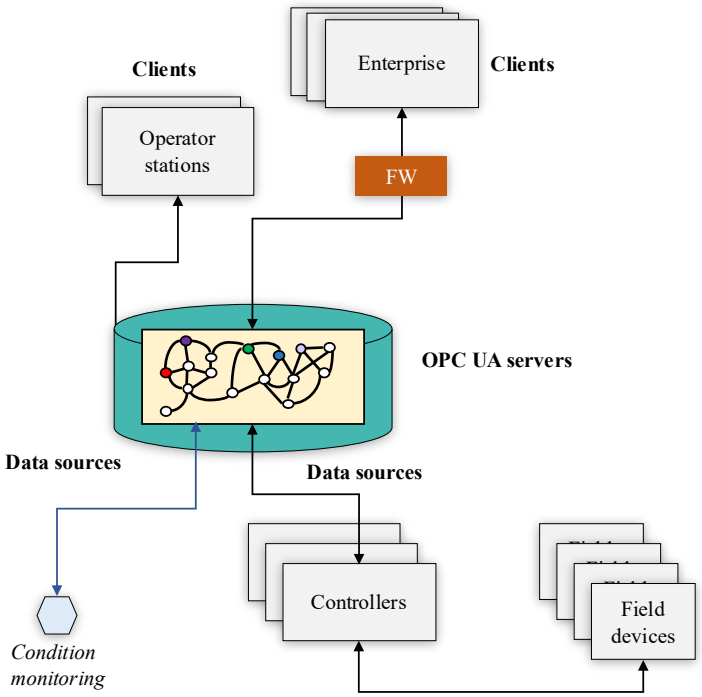


Fig. 39. OPC UA as a unified exchange framework

OPC UA is generally not used for real-time control, such as closed-loop control, but for sharing static data, dynamic measurements, and events for monitoring, trending, and event analysis. OPC UA servers collect or receive data, optionally store it, and expose it to clients.

It means that OPC UA is more than a protocol for data exchange; it can also organize data in graph-like models, providing context. Such models are referred to as information models, as they provide context for data, treating

it as information rather than just data. With these extended capabilities, OPC UA is considered an industrial communication framework.

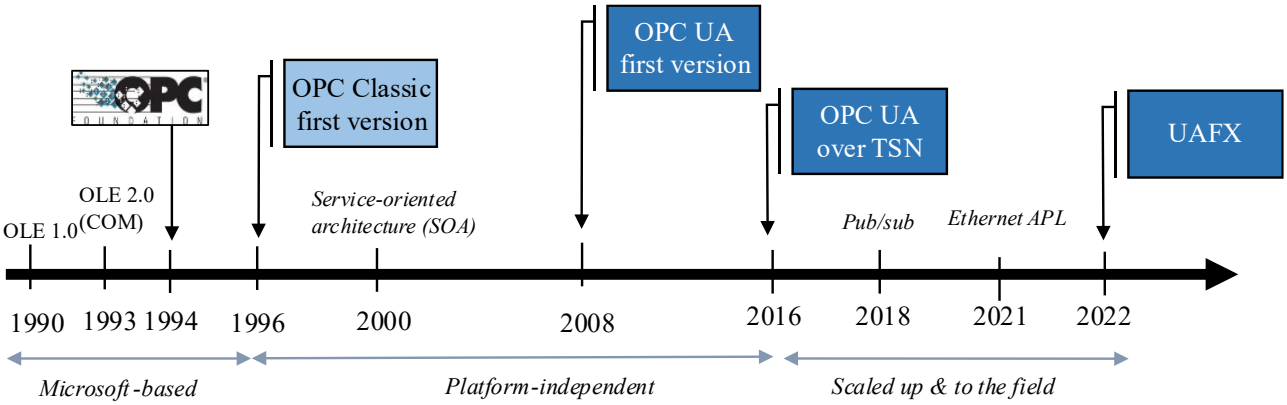


Fig. 40. Development timeline for OPC UA

OPC UA and some of its more recent extensions are part of an evolution illustrated by the timeline in Fig. 40. The first version introduced in the mid-1990s was named OPC and later referred to as OPC Classic. OPC Classic was introduced to aggregate and exchange data between controllers and users, regardless of whether the data was received via 4-20 mA or digital protocols such as HART, Foundation Fieldbus, Modbus, or Profibus. The first name for OPC was an abbreviation derived from the term OLE for Process Control. Here, OLE is the object linking and embedding functionality implemented by Microsoft. There are still many installations of OPC Classic in the industry today, given that many industrial applications and facilities operate for 20-40 years.

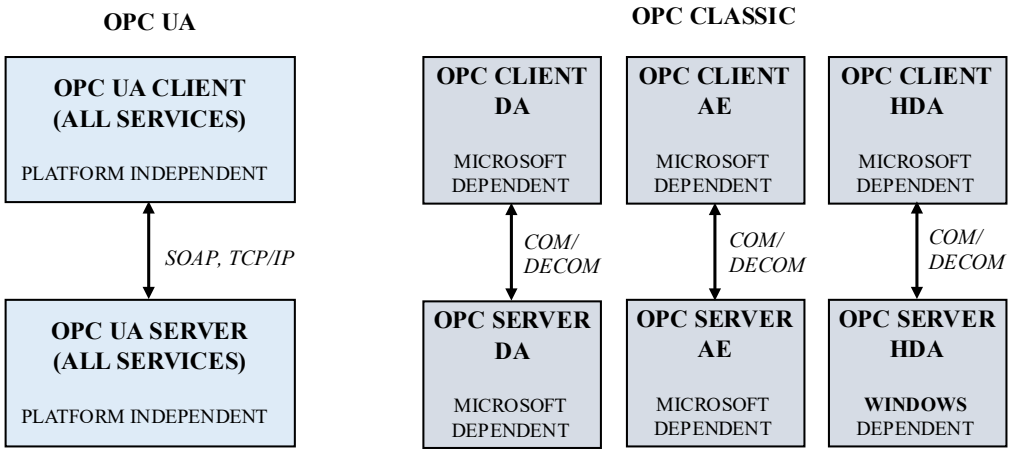


Fig. 41. OPC UA vs OPC classic

OPC UA became the new version, superseding OPC Classic and eliminating its dependency on Microsoft components. OPC UA is a framework comprising several specifications published by the OPC Foundation, some available for free and others requiring purchase unless you are an OPC Foundation member. The specifications describe the types of services and related formats needed for data storage and exchange. Some of the services provided are:

- Exchange of real-time data (DA – data access)
- Aggregation of historical data (historical access – HA)
- Alarms and events (Alarms and conditions – AC)

Except for OPC UA HA, OPC UA DA and AC are not intended to replace the real-time communication mechanisms between control systems and operator stations. In particular, alarm management is an integral part of the DCS, SCADA, and safety system architecture and is designed to meet the required performance,

determinism, and response-time requirements. OPC UA DA and AC primarily provide interoperable access to process data, alarms, and events for integration, monitoring, and information exchange between systems.

It is then up to OPC UA users to implement the services in C, C++, Pascal, or other languages. The first OPC UA version implemented client-server communication, which remains widely used; it can become less efficient in large-scale or high-frequency data distribution scenarios. Pub/Sub was introduced to improve scalability and decouple producers and consumers. In 2018, the OPC Foundation published a more scalable alternative in which servers publish data to clients via subscriptions, either directly or via message brokers (such as e.g., MQTT servers, explained later). This publish-subscribe (pub/sub) functionality is now the preferred solution.

To summarize, with the support of Fig. 41 OPC Classic differs from OPC UA in the following ways:

- OPC Classic must have distinct clients and servers for each of the services: data access (DA), alarms and events (AE, corresponding to AC in OPC UA), and historical data access (HAD, corresponding to HA in OPC UA). Therefore, the number of clients and servers is much larger than with OPC UA.
- The client and server functions and their communication must be implemented using Microsoft functionalities. Some of these Microsoft features have been shown to have exploitable vulnerabilities.

Initially, the intent was not to use OPC or OPC UA for real-time control, but only for data aggregation and the exchange of alarms and events. However, in 2016, OPC UA published its time-sensitive networking (TSN) profile, enabling the transmission of time-critical data with guaranteed timing. Its successor, OPC UA Field Exchange (UAFX), came a few years later, in 2022. UAFX added features for managing the safety and security of communication and, in addition to regular Ethernet communication, can support 5G, Wi-Fi, and (in the future) Ethernet APL for transmission.

In the remainder of this section, we will focus solely on OPC UA.

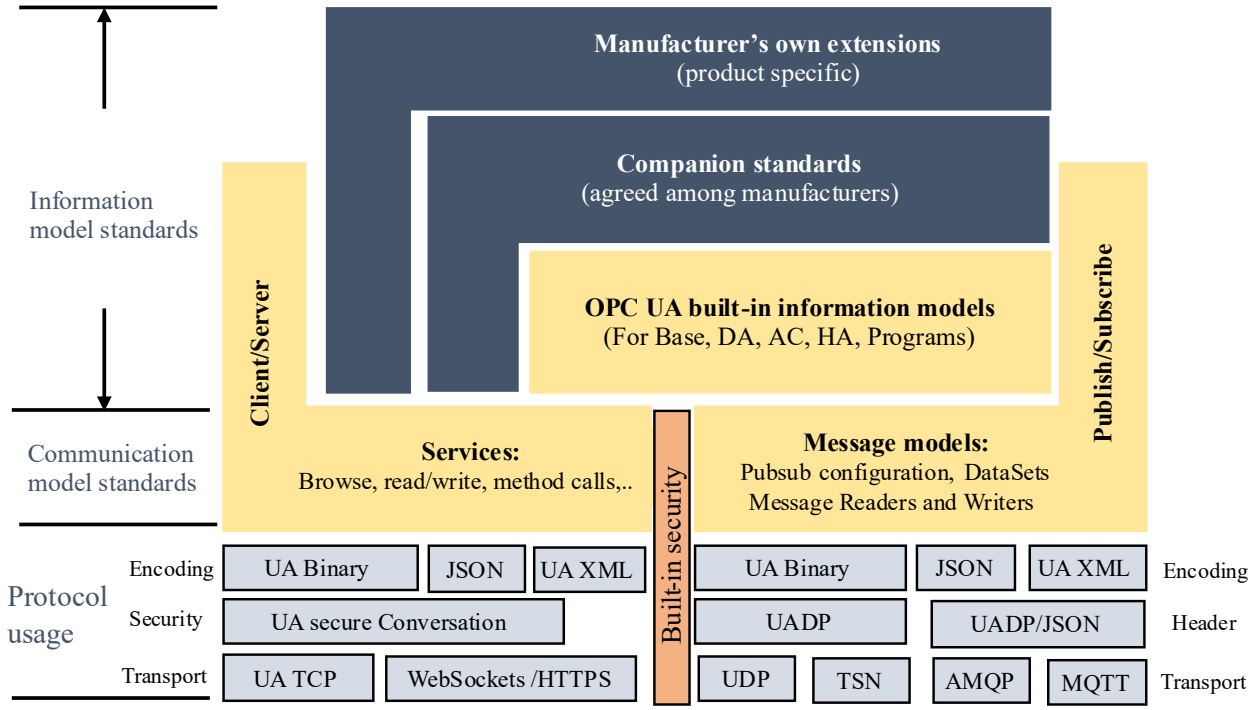


Fig. 42. Overview of OPC UA standards from OPC Foundation

2.9.1 OPC UA framework

Fig. 42 identifies that the OPC UA framework contains the following main modules:

- Information modeling standards, covering OPC UA information model building blocks, OPC UA core information models, companion information models, and manufacturer-specific extensions.

- Information model access, covering services and functions needed to read, write, execute logical operations (methods), configure (the information model), and handle notifications about new data and events. Some functions and services are used in the client-server setup, while others are used for the publish-subscribe setup.
- Choice of protocol mappings for implementation, covering the selection of protocols for encoding, security, and transport.
- Data exchange platforms, covering client-server and pub-sub implementations.

OPC UA provides many basic functions for modeling the structure and context of data, e.g., classes, properties, and relationships, both as templates and instances. Companion standards add "pre-made" OPC UA modeling blocks explicitly suited to an industrial application area or product, agreed upon by key stakeholders in that area (which eventually leads to the standard being approved by the OPC Foundation). An overview of specifications per 2022 is shown in Tab. 6. The specifications were later also published as IEC 62541 (2020), a standard split up into several parts with numbering similar to Tab. 6.

The specifications include requirements for all the mentioned building blocks. Various tools for coding servers and interacting with clients use software development kits (SDKs), such as Unified Automation's UA Modeler. According to their web page, the UA modeler provides graphical, hierarchical modeling of the information model using OPC UA notations and syntax (similar to UML). However, OPC UA developers often prefer C++, Python, or Java, and the SDK mentioned uses C++.

Tab. 6. Overview of OPC UA specifications (per 2022)

Core specifications	Access type specifications	Other specifications	Companion standards
Part 1: Concepts Part 2: Security Part 5: Services Part 4: Address space model Part 5: Information model Part 6: Mappings Part 7: Profiles	Part 8 – Data access Part 9: Alarms and conditions Part 10 – Programs Part 11: Historical access Part 12: Pub/sub	Part 12: Discovery (of servers and endpoints) Part 13: Aggregates Part 14: Publish–Subscribe (Pub/Sub) Part 15: Safety Part 16: State machine +++	• ISA Common Object Model • PLC model based on IEC 61131-3 • AutomationML • Process Automation devices • Profinet • Robotics • Machine vision • Asset administration shell • Kitchen machines… • +++++
Look here for a comprehensive overview: https://reference.opcfoundation.org/. Tips: Click on the part of interest and then «scope» to see what it is all about.			

The OPC Foundation, https://reference.opcfoundation.org, provides more extensive explanations of OPC UA in a web-based format via its menu "Documents."

2.9.2 Service-oriented architecture

OPC UA is designed around the principles of service-oriented architecture (SOA). In SOA, functionality is exposed as services that clients can discover and invoke. OPC UA implements this by:

- Servers: Provide services such as reading/writing data, subscribing to events, and calling methods. These services are described in a standardized information model.
- Clients: Consume these services by connecting to servers, browsing the information model, and invoking operations.

With SOA, clients and servers can be logically decoupled and coexist on the same or different hardware. One example of an OPC client is an operator station in the control room at level 2 of the Purdue reference architecture. Other examples are a software application for condition monitoring and a personal computer with a plant optimization tool at level 4. A network architecture may therefore include several OPC UA servers and clients on the same or different hardware, as shown in Fig. 43, distributed (as needed) at all layers of the Purdue reference architecture, even if level 0 is less common.

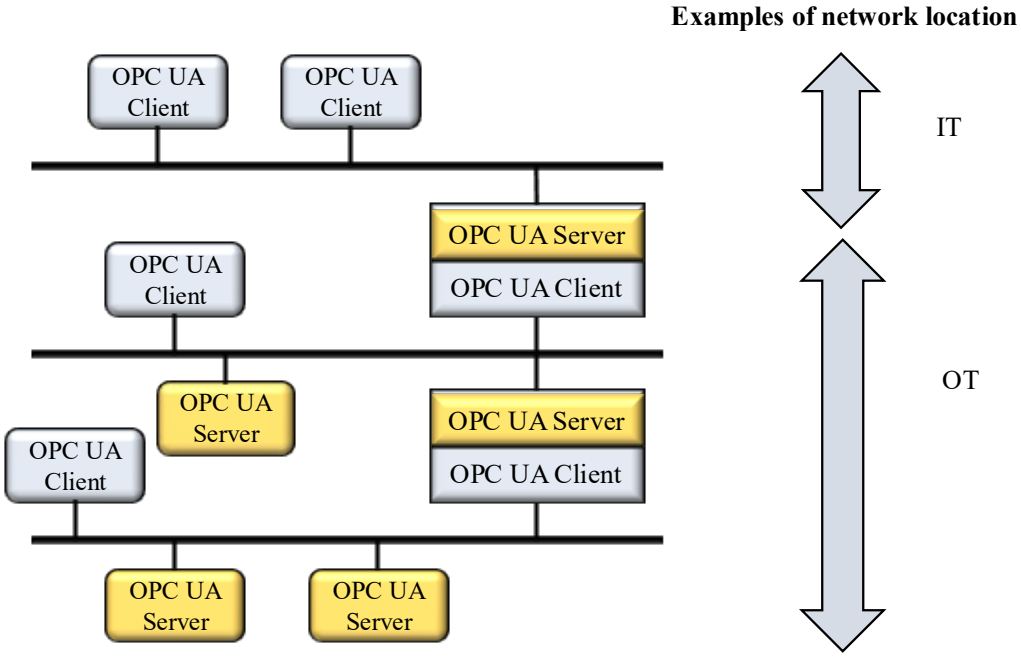


Fig. 43. Simplified architecture for OPC UA

2.9.3 OPC UA exchange profiles

OPC UA clients and servers can exchange data within the address space synchronously via a request-response relationship, or asynchronously via a publish-subscribe (pub-sub) subscription. OPC UA DA and AC services typically exchange data only when a significant change occurs, either an event, e.g., an alarm for AC, or a measurement change beyond a configured threshold (commonly around 0.5%) for DA. If no such change occurs, no data is transmitted.

In contrast, the HA service is often configured to retrieve data at regular intervals (e.g., every 1–5 seconds), even if the values remain unchanged. Depending on the configuration, intermediate changes between logging intervals may not be stored. The sampling and publishing intervals are usually sufficient for monitoring and trending, but they may fail to capture rapid dynamics (e.g., in power distribution systems and compressor control). This limitation is why operator stations and control systems rely on Fieldbus or IE for real-time control and to supply the control room with data for presentation on operator stations.

Fig. 44 illustrates the principle of pub-sub. Here, the server organizes a set of sessions for each client. Each session retrieves all data from the subscription's address space and publishes it to the client. Pub/sub is becoming increasingly popular because it is more flexible, scalable, and less resource-intensive than the client-server request-response model.

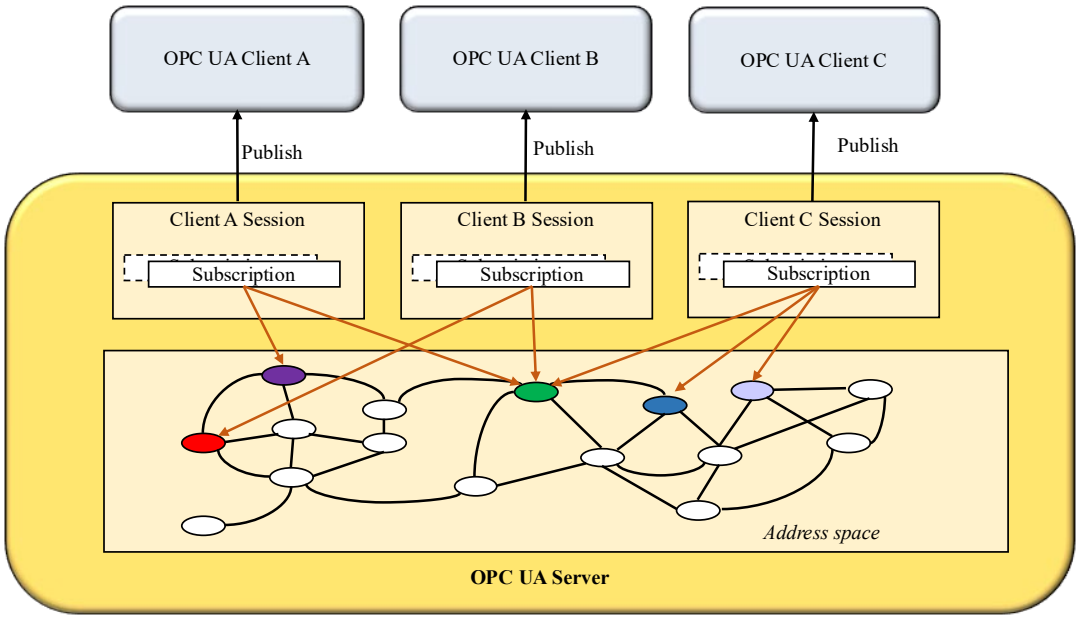


Fig. 44. Publish-subscribe with OPC UA server and clients

An address space is not necessarily confined to one physical server but can stretch over several servers, as shown in Fig. 45. Nodes are uniquely identified within an OPC UA server's address space using NodeIds, which are paired with a Namespace URI to ensure global uniqueness. If equivalent data is represented on multiple separate OPC UA servers, it is modeled as distinct nodes with different NodeIds unique to each server's context.

However, OPC UA does define a standard global identity mechanism across servers using ExpandedNodeIds. This mechanism includes the full Namespace URI and a Server Index (pointing to a remote server's unique application URI). This allows standard References to link nodes across different servers that are either related or replicated, though semantic consistency across independent systems is typically enforced through unified information models or external mapping configurations.

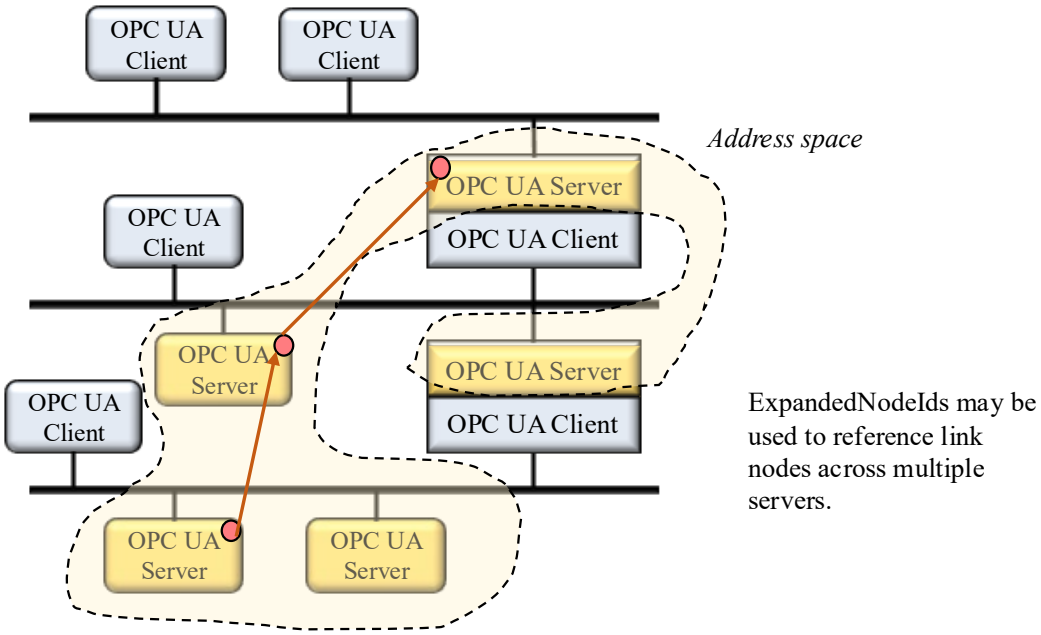


Fig. 45. Address space and name space.

2.9.4 Application of OPC UA

OPC UA is typically implemented with the following steps:

1. Create the information model and transform it into a suitable XML format: Create an information model, for example, using UML with OPC UA-specific symbols. This step is essential when the data structure and relationships are not yet defined. It helps determine:
    - What data to include about the system
    - How data nodes relate (both hierarchical and non-hierarchical)

    A possible tool to generate the information model is Unified Automation's OPC UA Modeler. The same tool can be used to export the information model to a format suitable for the configuration of a server, such as NodeSet XML.
2. Configure the server and load the information model at runtime. Several implementation approaches may be used:
    - Runtime import: Import the information model directly into the server at runtime (e.g., from a NodeSet XML file), automatically creating nodes and relationships in the server's address space (the set of all nodes managed by the server).
    - Code generation: Convert the NodeSet XML into source code (e.g., C++ or Python), which is then compiled and executed to instantiate nodes and relationships, allowing further customization.
    - Manual implementation: Define the information model directly in source code and configure the server from scratch.

    A possible tool to use is Unified Automation's software development kit (SDK),.
3. Configure the client and connect to the server: Set up the OPC UA client to connect to the server. This involves:
    - Specifying the endpoint URL of the server (where OPC UA exposes its service)
    - Configuring the security settings (certificates, encryption, authentication)
    - Selecting the namespace and data nodes of interest, meaning the logical grouping of node identifiers that are within the namespace and of interest to a specific client.

    A possible tool to use is Unified Automation's OPC UA SDK.
4. Access and browse data from the client according to request or subscription: Once connected, the client can:
    - Browse the server's address space to explore hierarchical and semantic relationships.
    - Read and write values, subscribe to data changes, and invoke methods.
    - Validate that the information model behaves as intended in real use cases.

    A possible tool to use is Unified Automation's UAExpert.

Many vendors are developing their own tools, some of which build on Unified Automation's software platform.

2.9.5 Creating an information model

An information model consists of nodes (representing data) and the relationships between them. The structure and relationships of the data contained in the information model are often first derived using tools such as UML. OPC UA has introduced its own UML symbols that align with its node and relation types. More specifically, the UA UML legend consists of eight basic building blocks for nodes, as reproduced in Fig. 46. Some of the building blocks are used to create templates, i.e., reusable information model elements, and instances tailored to specific system realizations where the data is relevant.

1. Object type: Describes a type of asset. An asset can be a system, a process, a service, a machine, or equipment/device. A general specification of a pressure transmitter, for example, can be an object type.
2. Object: An instance of an object type. For example, a specific Rosemount pressure transmitter can be used.

ObjectTypes and Objects are both NodeClasses that can act as organizing elements in the information model hierarchy.

3. Variable type: A variable associated with an object and its associated value(s) format. The values can be static or dynamic, and of data types such as string, integer, and real. For a pressure transmitter, it means to specify what kinds of variables to include, such as, e.g., the measuring principle of the format string, the manufacturer of the format string, the pressure measuring range, whether the format is real or float, the measurement reading of the format real or float, and the configuration parameters, where formats are specified separately as needed.
4. Variable: An instance of a specific variable assigned a static value or updated as a dynamic value. For example, a description of the measuring principle, the manufacturer's name, the pressure range, the actual measurement (dynamic), and the configuration parameters of a specific pressure transmitter can be included.
5. A data type that defines the format of values, such as a string, float, structure, or integer.
6. Method: Specifies calls to functions executed on an object, such as starting or stopping something, or making a calculation. A method relevant for a pressure transmitter is to calculate the difference between the measured value and the setpoint.
7. View: Specifies the nodes that belong to a specific browsing choice, for example, the objects of interest. For example, we may want a pressure-temperature view to browse data from all pressure transmitters.
8. Reference type: Describes how the different node classes connect to each other.

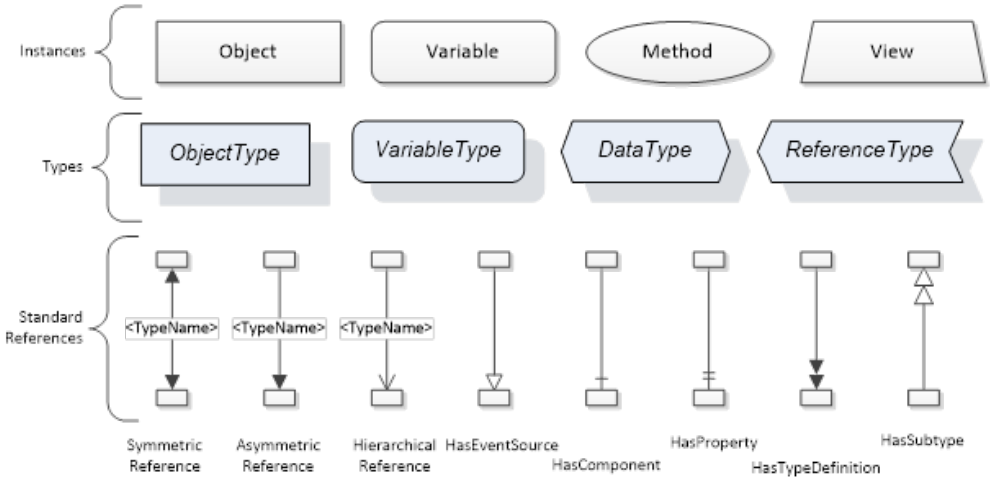


Fig. 46. OPC UA modeling symbols (OPC foundation)

There are eight types of references:

- Hierarchical references are a general relation that can organize nodes into a tree-like structure of nodes and sub-nodes in a parent-child style. For example, hierarchical references can logically break down a system into its components and sub-components. As indicated, it can be given a name. HasComponent and HasProperty are two special cases, each identified by its own relation symbol.
- Non-hierarchical references create the relations that exist across these trees. For example, a non-hierarchical reference can link nodes in different hierarchical trees.
- Symmetric references define more of a functional relationship between nodes, for example, that a pressure transmitter connects to a logic solver to allow sending a measurement. Here, the pressure transmitter and the logic solver could exist within the same tree structure, with this symmetric reference as an added relationship.
- Asymmetric references define a bi-directional relationship between nodes, where the meaning is different for each direction. For example, an asymmetric reference can define a pressure transmitter with a specific test procedure; however, it does not make the same sense to say that the test procedure has a pressure transmitter.

- HasComponent is a special case of a hierarchical reference for functional and physical relationships between objects and subobjects (e.g., systems and system parts), main data values (variables), and methods.
- HasProperty is also a special case of a hierarchical reference, but is only used to reference variables that contain metadata, meaning something descriptive about an object, a variable, or a method.
- HasTypeDefinition makes a reference from an instance node (object, variable, or method) to its corresponding type (or "template").
- HasSubtype is a hierarchical reference that identifies how one type (below) inherits what was defined for another type (above).

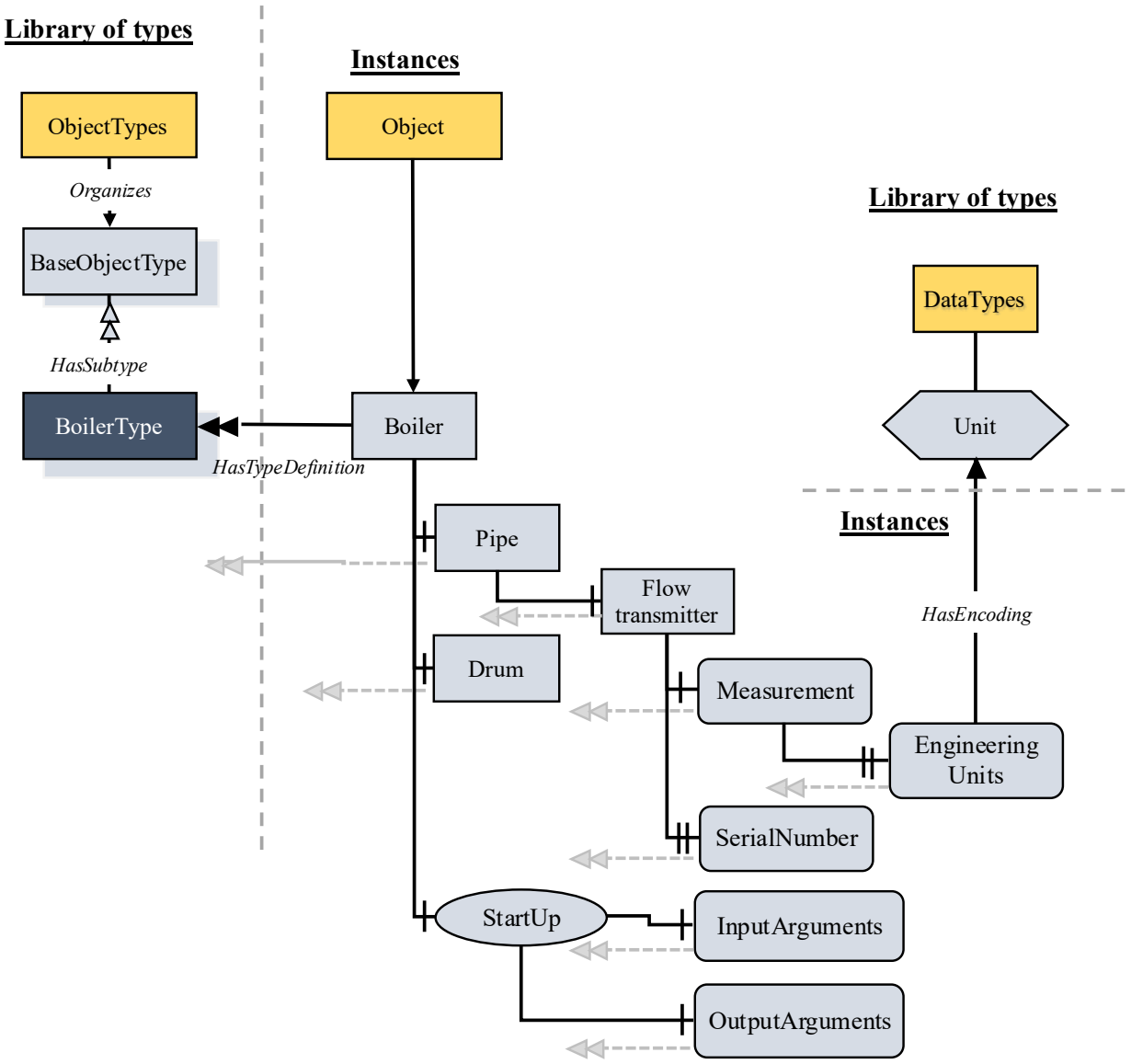


Fig. 47. Information model example (adopted from OPC Foundation)

As shown in Fig. 42, the OPC UA framework contains graphical symbols and coding implementation specifications for the generic building blocks of the eight listed node types. These are supplemented by a large number of extensions published as companion standards, agreed upon by a broader community, and manufacturers' product-specific standards. Users can also develop their own (internal) extensions to serve their needs.

The textual or graphical information model specification is used to construct the server's address space. For example, UaModeler from https://www.unified-automation.com/ is a tool that allows both direct coding and coding via XML and JSON. Clients will have a browser tool that provides access to all or part of the implemented address space. Examples of client views are found at https://www.opclabs.com.

Fig. 47 illustrates an information model of a boiler system modeled with OPC UA (graphical) building blocks. The new building blocks (or template) for a Boiler named BoilerType are generated from the general building block BaseObjectType.

Although not shown (except with light grey arrows), similar templates or types may exist for other object instances, including the pipe, the drum, and the flow transmitter.

- ObjectTypes, Object, and DataTypes are NodeClasses that organize, by linking to their parts of the information model below.
- The Boiler object is an instantiation of BoilerType and relates to a specific boiler. The Boiler object has three sub-components, organized into a hierarchical tree: Pipe, drum (tank), and the method StartUp.
- The Pipe object has one Flow transmitter object installed. The Pipe and the Flow transmitter are instantiations of corresponding types (not shown). The Flow transmitter object has two variables: A measurement (dynamic data) and a serial number (static data).
- The value format of "Measurement" must be Data Type EngineeringUnits. This implies that applicable units are kg, m, bar, m/s, and so on.
- HasComponent is the reference from FlowTransmitter and the Measurement, as the measurement is a related part (or a function).
- In contrast, HasProperty is used as the reference between FlowTransmitter and SerialNumber, as the SerialNumber provides descriptive information *about* the FlowTransmitter.

The StartUp object can control the start and stop of the boiler system, based on input arguments.

2.9.6 OPC UA FX

OPC UA Field eXchange (UAFX), introduced around 2022, extends OPC UA toward field-level communication, including controller-to-controller interaction, safety, and motion use cases. OPC UA FX enables deterministic communication and synchronization using Time-Sensitive Networking (TSN). TSN refers to a set of IEEE 802.1 standards for deterministic data transmission over Ethernet and includes the following features:

- Time synchronization among devices.
- Traffic scheduling and shaping, enabling prioritization of critical data streams.
- Resource reservation, ensuring bandwidth and time slots for deterministic traffic.
- Fault-tolerant transmission (e.g., frame replication over multiple paths).

These features provide the capabilities required for real-time operation, similar to those of industrial Ethernet protocols.

OPC UA FX supports two main communication types, illustrated with Fig. 48 :

- Controller-to-controller (C2C) communication
- Controller-to-field device (C2D) communication

To coordinate communication, one controller may be assigned the role of connection manager (CM).

Information models remain central, with controllers and devices exposing data via OPC UA servers referred to as automation components (ACs). TSN-capable switches provide deterministic transport. Data exchange may use client-server or Pub/Sub communication profiles. OPC UA FX can co-exist in networks that exchange data over standard OPC UA and other Fieldbus and industrial Ethernet protocols, e.g., IEC 61158/IEC 61784. This is illustrated in the upper part, the lower middle, and the left part of Fig. 48.

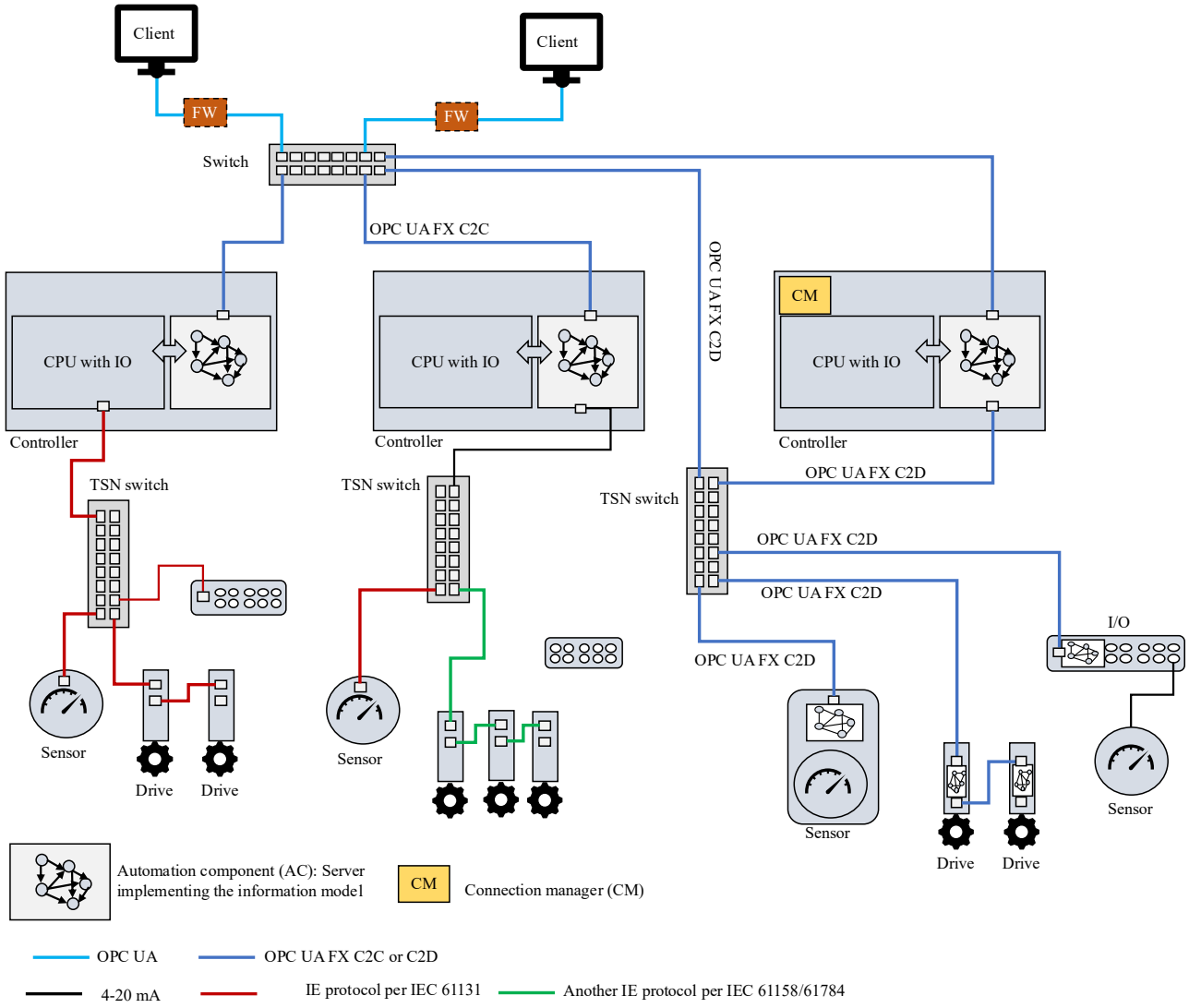


Fig. 48. Network incorporating OPC UA FX (inspired by iebmedia.com)

2.9.7 OPC UA Safety

OPC UA and its extensions, such as OPC UA FX, can implement OPC UA Safety. This profile is specified in IEC 62541-15 (2025). It is based on the black-channel principle defined in IEC 61508 (2010), where a safety layer is implemented above the communication stack (OSI layers 1-7).

According to the OPC Foundation, OPC UA Safety supports safety integrity levels (SIL) up to SIL 4. The safety layer adds mechanisms such as unique identifiers, sequence counters, and cyclic redundancy checks (CRC) to detect errors and ensure data integrity.

The safety profile is not designed to protect against malicious manipulation; however, OPC UA security mechanisms (e.g., encryption and authentication) may be applied where timing constraints allow.

2.9.8 MQTT architecture with OPC UA Pub/Sub binding

Message Queuing Telemetry Transport (MQTT) is a lightweight publish–subscribe application-layer protocol that uses a compact binary format for control messages. It is commonly used with an MQTT broker to relay

messages between clients, enabling one-to-many communication. Although the protocol itself is binary, the payload can be either binary or text-based because MQTT does not interpret the data.

As defined in OPC UA Part 14, MQTT can be used as a transport mechanism for OPC UA PubSub. It enables efficient data distribution, particularly in scenarios with many subscribers to the same data stream..

MQTT provides quality-of-service (QoS) levels that ensure message delivery through mechanisms such as retransmission and broker-side retention. This complements OPC UA PubSub by adding reliable, broker-based distribution, which is not part of the native PubSub model.

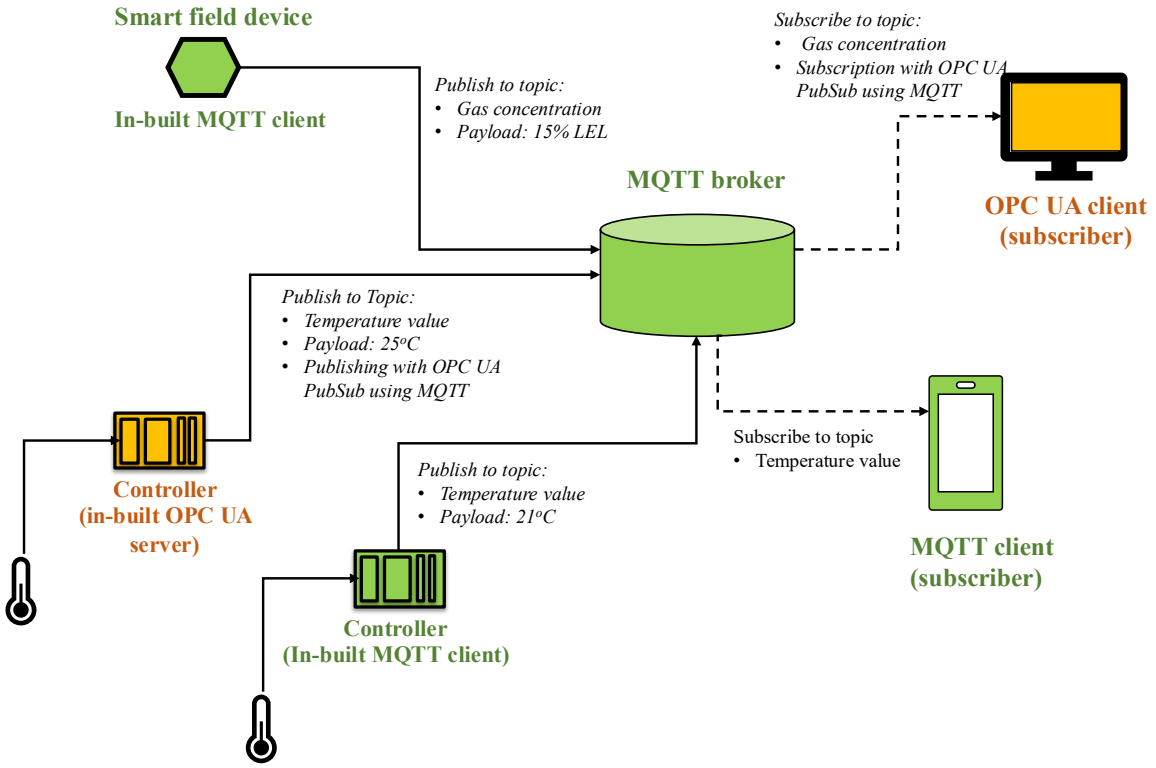


Fig. 49. Example of MQTT Architecture with OPC UA Pub/Sub Binding

With its low overhead and minimal bandwidth requirements, MQTT serves as middleware, enabling efficient and reliable data exchange between distributed systems. Devices with an embedded OPC UA server can publish data using OPC UA PubSub encodings (e.g., UADP binary or JSON) over MQTT to a broker, which then distributes the messages to subscribers. On the same network, devices with built-in MQTT client functionality can publish data directly, which is particularly relevant for Industrial Internet of Things (IIoT) devices where implementing a full OPC UA stack would be resource-intensive. A possible way to illustrate an OPC UA environment integrating MQTT clients and a broker is shown in Fig. 49.

So, to sum up, why MQTT is useful for OPC UA:

- MQTT adds broker-based reliability to OPC UA Pub/Sub, providing delivery guarantees such as retransmission and message retention through its QoS mechanisms. It also improves scalability by efficiently distributing data to many subscribers through topic-based filtering, so subscribers receive only data from selected topics.
- In contrast, OPC UA PubSub without MQTT can still operate efficiently in brokerless setups (e.g., UDP-based communication). However, it transmits predefined data sets rather than selectively filtered data and is therefore best suited for low-latency, deterministic communication, where delivery guarantees and large-scale distribution are less critical.

2.10 Digitalization efforts with Industry 4.0

Industry 4.0 is the short name for the fourth industrial revolution, first introduced by Klaus Martin Schwab, a German engineer and economist, and the founder of the World Economic Forum (WEF) (wiki). The name was used to describe technologies for the flexible and seamless exchange of data within a factory or plant and between the plant and cloud applications. Industry 4.0 involves wireless and Ethernet-based communication, the industrial Internet of Things, digital twins, artificial intelligence, and more autonomous and intelligent devices. The association "Norsk Industri" is among those that have put Industry 4.0 on the agenda; see https://www.norskindustri.no/dette-jobber-vi-med/internasjonalt/industry-4.0/.

The German Platform Industrie 4.0 website is a good place to learn about Industry 4.0. For example, they exemplify the vision for future factories and plants, as shown in Fig. 50. Instead of stringent network segmentation, they envision a factory environment where IT and OT are more integrated, with all devices and software applications communicating seamlessly and securely. This vision may be achievable in manufacturing, but it is not fully applicable to the process industry, where more control over data flow is needed to prevent major accidents.

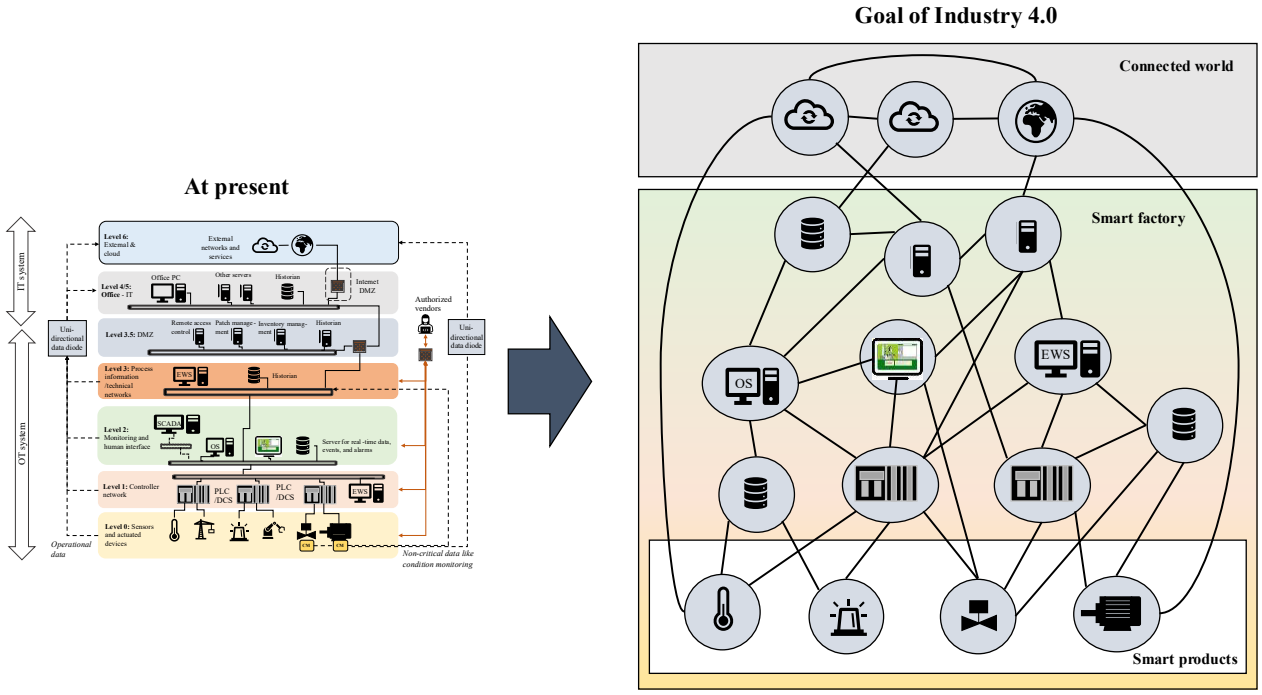


Fig. 50. Goal of Industry 4.0 network architecture

Industry 4.0 was initially platform-driven in the manufacturing industry, where levels of automation, connectivity, and device integration were lower than in the process industry, at least since the 1990s. Examples of Industry 4.0 technologies specifically relevant for the process industry sector are:

- Asset Administration Shell (AAS), which is a platform for generating digital twins for key assets of the process from components to composed systems.
- Namur Open Architecture (NOA) for secure sharing of data directly from the lower levels of the Purdue model to the higher levels of the enterprise, to avoid having all data pass through the levels that are focused on real-time operation.
- Open Process Automation System (O-PAS), which is an open platform for seamless integration of control systems from different manufacturers.
- Model-based Type Package (MTP), which is an open and standardized way of specifying the functions of devices and controllers to more easily integrate them into an operator station.

OPC UA and Automation ML may complement this list, as they either interface with or are part of the implementation.

It may be noted that new industry ambitions beyond Industry 4.0 have been formulated under names like Industry 5.0 (human-centric solutions) and Industry 6.0 (hyper-connected solutions). With Industry 6.0, AI and autonomy are expected to play a key role, with minimal human involvement.

2.10.1 NAMUR open architecture (NOA)

NAMUR open architecture (NOA) is a secure data exchange concept and framework introduced in NAMUR NE175 (2020), also explained in a presentation (NAMUR, 2020). NAMUR is the German user organization for automation technology in the process industry, and many of their standards are adopted globally.

The primary focus of NOA is on a secure and efficient way to send data from the lower levels of the Purdue reference architecture, including systems at levels above and the Cloud. NOA splits the architecture into two main zones: Maintenance and Optimization (M+O), where plant data is needed, and process control, which covers all data available and used for operation, including control systems, safety systems, and utility systems such as the power supply.

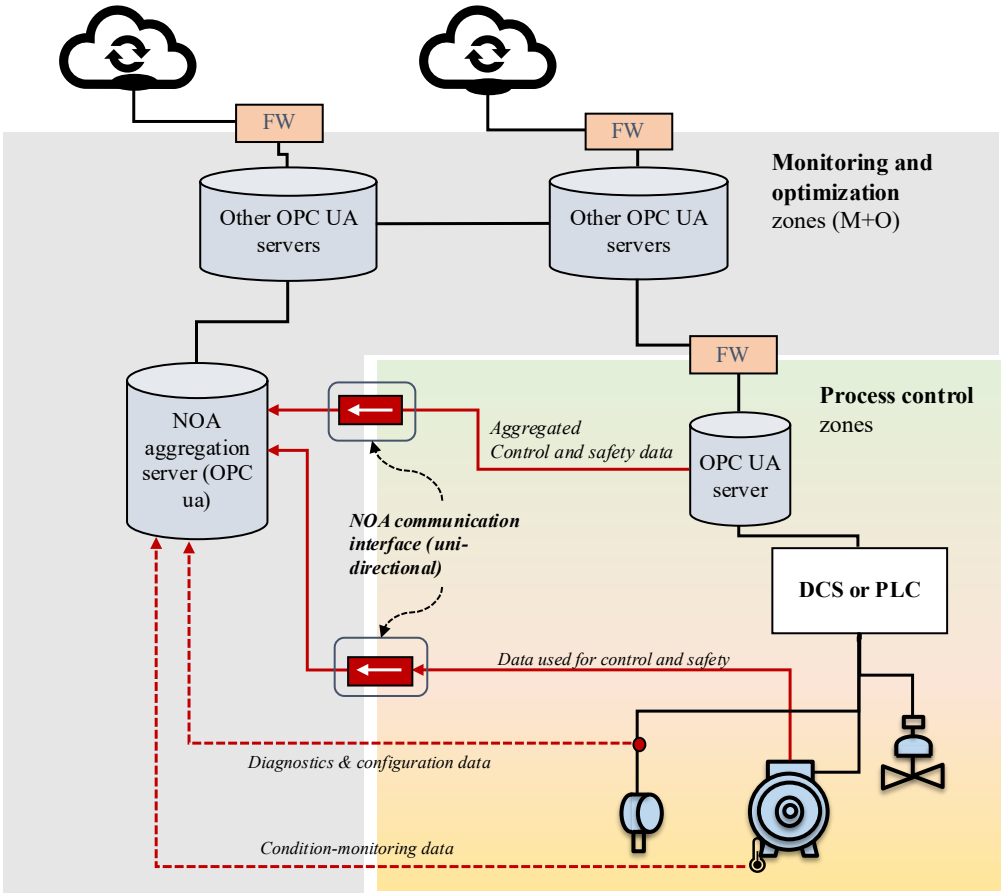


Fig. 51. NOA for open sharing of process plant data

Fig. 51 illustrates how NOA realizes the data transfer from the process control zone to the M+O zone. NOA suggests that:

- Condition-monitoring data, not directly used for control or safety functions, can be sent directly to the M+O zone, which, in practice, means Purdue level 3 and beyond. Such data includes data from devices installed for condition monitoring and from smart field devices, such as HART.
- Process control zone data can be sent directly to the M+O zone if using a NOA communication interface that guarantees a read-only pathway.
- NOA proposes two alternatives for communication interfaces:
    - Software (+ hardware): NOA gateway, with special configuration to control direction.

- Hardware: Unidirectional by design

Not described in Fig. 51 is the suggested approach for secure and verified writing from the M+O zone to the process control zone. Here, secure and verified include authentication, data validation (correct and complete), approval by operations, logging and traceability, and a fail-safe response action that blocks write operations if verification fails. A firewall complemented by an approval procedure and general security measures for authentication, authorization, and logging would cover these functions.

2.10.2 Automation ML (AML)

Automation Markup Language, referred to as Automation ML or just AML, is an XML-based file format for engineering data in manufacturing and process industry systems and equipment. The basic architecture was developed by several companies, as explained in Drath et al. (2008), and is an alternative to the traditional methods of storing and exchanging data via PDF documents, Excel files, and databases in various formats.

Many industrial design and engineering tools support AML file import, making it easier and less labor-intensive to share data between tools and actors when the AML content structure is standardized.

- For example, design data about a product from the manufacturer can be directly imported into the equipment database that the customer needs to keep updated.
- Technical drawings for an electrical circuit in an AML file, e.g., with cable tag numbers, wire tag numbers, terminal numbers, and references to input/output cards, can be automatically shared with systems that track installation, commissioning, and testing.
- Technical documentation that specifies control logic, like the system control diagram (SCD) explained in Chapter 3, can be modeled in AML and directly imported into the controller programming editor and converted to executable code without manual effort.

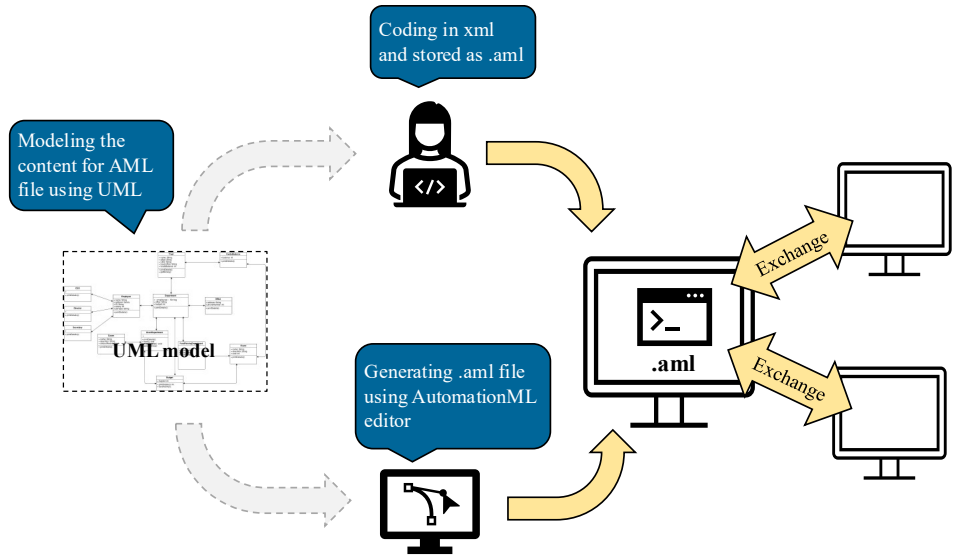


Fig. 52 Application of AML files.

AML is, therefore, referred to as a tool for computer-aided engineering exchange, sometimes abbreviated as CAEX. Although this technology has been available for some time, it has gained increased attention in recent years, as it supports the implementation of OPC UA and other Industry 4.0 technologies (explained later).

The generation of an AML file with data about a system or component can be done by coding in XML according to the AML classes and formatting, or via a graphical interface in an Automation ML editor, as shown in Fig. 52Some prefer to create a more general software architecture model as a starting point, for example, using the Unified Modeling Language (UML); however, this is not a requirement. UML is often used in standards and guidelines to show metamodels for how AML classes and formats are applied, as in OPC UA and the asset administration shell (AAS).

The AutomationML.org website provides helpful information on software and AML. Another essential reference is IEC 62714: Here, part 1, i.e., IEC 62714-1 (2018), focuses on the architecture and general requirements for AML graphical modeling and XML coding using concepts such as classes, instances, relations, references, hierarchies, and basic (inbuilt) AutomationML libraries. In short, AML implements an object-oriented file structure, organized according to the following object categories:

- A system unit library with reusable/premade objects, where the objects can be physical components, systems, and structures that show how systems interact.
- Instance hierarchy, with instances of objects from the system unit library and their relationships.
- Interface class library, covering the types of relationships between objects and between instances.
- Role class library adds a higher-level abstraction, or naming, of the objects.

Adding attributes to objects containing the stored data and its format is possible. The attribute can also refer to a file or another source where the data is stored if it is not within the AML file.

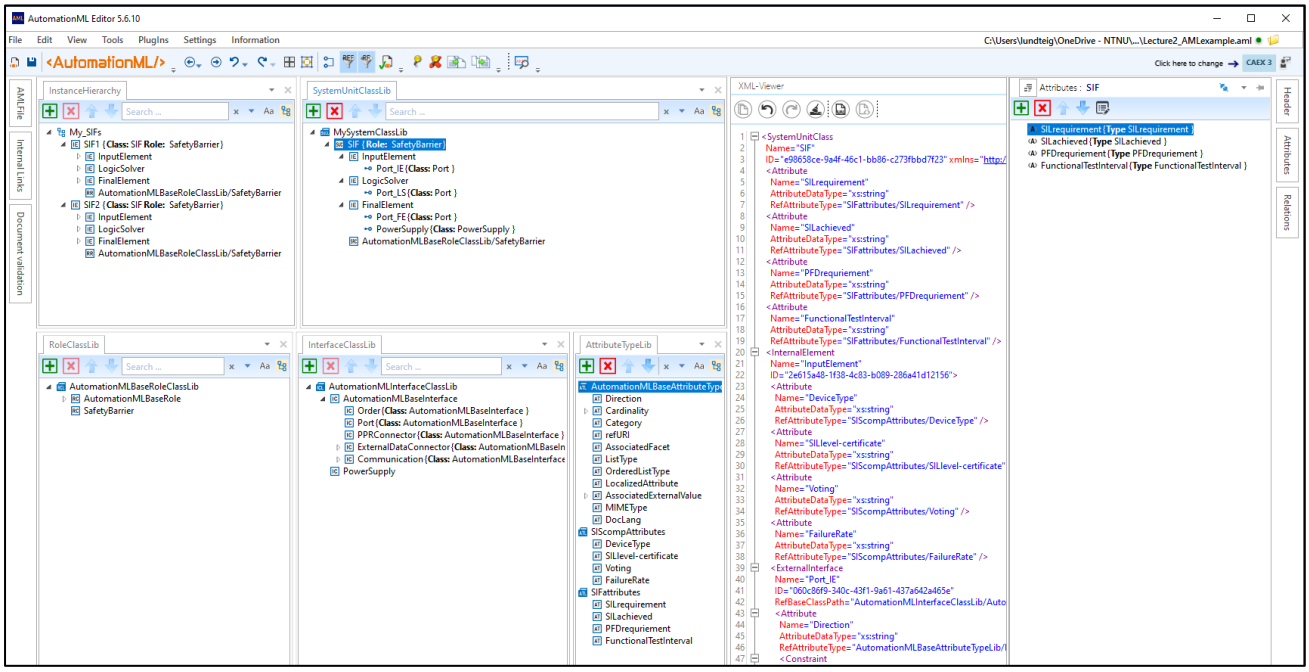


Fig. 53. A view into the AutomationML editor

An AML file can be coded directly in XML or configured using a graphical editor. The graphical editor is often a good starting point for modeling and interpreting engineering data structures, such as the open-source software AutomationML editor.

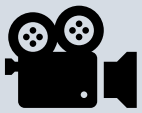
	Watch the YouTube explanation of how the AutomationML editor can create the AML structure of architecture and data associated with a robot. The video is a few years old, so the editor version is different from the most recent (version 5). However, the principles are the same or at least similar.

Fig. 53 illustrates how a safety instrumented function (SIF) and its device properties can be represented in an AutomationML editor.

The editor has several windows for modeling data content, structure, and relationships. The corresponding XML code for the data can always be viewed in any of these windows. On the left, the complete AML file, internal links, and document validation messages are shown; on the right, a header (or metadata) for each object, the corresponding list of attributes, and any relations are shown.

- A SIS may perform one or more safety instrumented functions (SIF). The attributes that apply to the overall function are named “SIFattributes”.

- Devices (or components) are often grouped into three main subsystems (Input element, logic solver, and final element) that perform an SIF. The device attributes are named “SIScompAttributes”.
- A SIF acts as one of several safety barriers that prevent accidents. Therefore, we added the role “safety barrier” to the SIF template.
- The template is instantiated for two specific SIFs named SIF1 and SIF2, along with their specific subsystems.

This simple example did not implement external interfaces and internal links.

The Automation ML organization is a good starting point to learn more about the standard and its practical application. See https://www.automationml.org/about-automationml/automationml/. As OPC UA provides a companion standard that specifies how AML files can be converted into OPC UA information models and coding, the OPC Foundation also has some explanation of Automation ML.

2.10.3 Module Type Package (MTP)

The production process is often split into modules that are designed and engineered separately. Typical examples include produced water treatment skids, chemical injection packages, gas/liquid separation, and gas dehydration units. When an equipment manufacturer delivers one of these physical process units, it arrives fully fabricated with its own field instruments, safety valves, and control and safety systems. In the end, all these systems must be fully integrated into a single overarching control and safety system with a common human-machine interface (HMI). A problem today is that integration is not always straightforward, as many different manufacturers and system integrators are involved.

Module Type Package (MTP) is an industry framework developed under the NAMUR NE 148 recommendation and formalized by the VDI/VDE/NAMUR 2658 guideline, aimed at easing the integration and reducing engineering hours. It is represented by a digital specification of these individual process modules (encompassing instrumentation and control) to achieve seamless interoperability. This specification is delivered as a file in a standardized, machine-readable package written in the XML-based automation language AutomationML. This single file serves as a unified digital description of the module's entire functional profile, split into (but not limited to) the following parts:

- Human-machine interface (HMI) integration
- Alarm management integration
- Maintenance and diagnostics
- Process control operations and functions
- Safety functions
- Security functions

With these MTP files prepared for each process module, it is ideally not necessary for system integrators to manually program data mappings or design the layout of the operator stations from scratch. Instead, they can simply import MTP files into the higher-level DCS or supervisory SCADA system, leveraging an MTP central orchestration feature. Within this DCS or SCADA environment, the central orchestration layer:

- Orchestrates process functions: Maps and connects the pre-programmed process control functions and module "Services" (defined in the MTP file) into a centralized, high-level plant sequence or recipe, rather than rewriting the lower-level executable controller code. Each process module is assumed to include its own control and safety system controllers and executable code. Integration with MTP does not replace these systems; rather, it integrates them more seamlessly into the overall control infrastructure.
- Generates the unified HMI: Automatically instantiates a common, integrated HMI for all the piecewise-engineered process modules on the operator stations. It builds the associated system faceplates, alarm

management paths, and maintenance diagnostics directly from the HMI specifications embedded in the MTP files. As with many evolving technologies, the maturity and native availability of these MTP features still vary among automation system suppliers and integrators. However, with the industry's strong push toward interoperability, modularity, and plug-and-produce design, MTP stands out as a highly promising concept.

This concept of pieces being puzzled together may be illustrated, a bit simplistically, as shown in Fig. 54 .

Many more thorough explanations are available online, for example, from ProfiNews by PI, the organization for Profibus and Profinet.

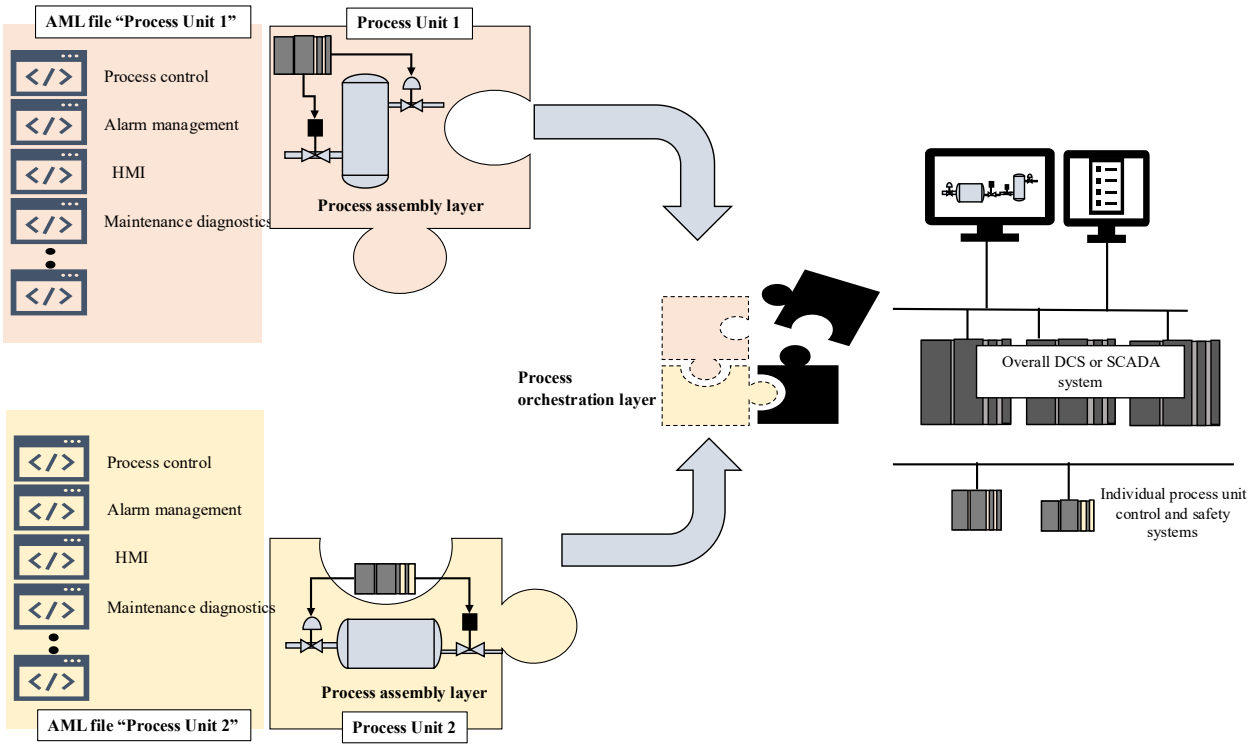


Fig. 54. MTP as illustrated by PI, the Profibus and Profinet organization

2.10.4 Asset administration shell (AAS)

Asset Administration Shell (AAS) is one of the latest Industry 4.0 standards for generating digital twins of physical and non-physical assets with lifecycles spanning engineering and the operation of process facilities and manufacturing plants. The specifications and details for implementing AAS are available from:

- Platform Industry 4.0 webpage https://www.plattform-i40.de/. Several reports detail the asset administration shell.
- International Digital Twin Association (IDTA) webpage: https://industrialdigitaltwin.org/en/content-hub/aasspecifications
- IEC 63278 series. For now (2025), only one part is available: The IEC 63278-1 (2023) Asset Administration Shell for industrial applications - Part 1: Asset Administration Shell structure.

The AAS specifications have introduced a shell-like symbol, as shown in Fig. 55 to visualize that the digital twin encapsulates all relevant information about an asset. AAS can also be generated for systems that include several assets, for example, a manufacturing line with conveyor belts, machines, and robots. An essential feature of AAS is enabling seamless interoperability for data exchange between systems, meaning an agreed-upon, machine-interpretable representation of data.

AAS implements three key features: the first is to organize data in a standardized, machine-readable format as a zip archive containing .aasx files. However, while such digital representations of assets in the AAS format are sufficient for the design and engineering phase, they do not meet the needs of operation, where the status and performance of the assets need to be tracked. AAS therefore has a second feature that allows digital representations to be stored on servers and linked to the real physical assets. However, this does not fully meet the need for intelligent and autonomous operations, and AAS has therefore added a third feature that also enables data exchange between AASs.

More specifically, the AAS targets the following implementation alternatives, where one or more of them can be used in combination:

- **Type 1:** The AAS is a static digital twin for file exchange: AAS and/or AAS submodel files exchange static data and information in a machine-readable and interoperable format between stakeholders in design and operation.
- **Type 2:** The AAS is placed onto AAS servers to become a digital twin that interacts with real-time operations. Application programming interfaces (APIs) such as HTTP/REST, MQTT, and OPC UA can subscribe to or request information. If allowed, the AAS submodels may perform operations on the data and return the results.
- **Type 3 (not yet available as of 2024): The** AASs' interaction is enabled, allowing the AASs to interact autonomously, using a new standardized Industry 4.0 language. For example, one submodel can make reliability predictions based on condition-monitoring data and provide input to another submodel that decides the impact on maintenance scheduling. They can instruct AASs on when their corresponding physical asset should be inspected, repaired, or replaced at the next inspection.

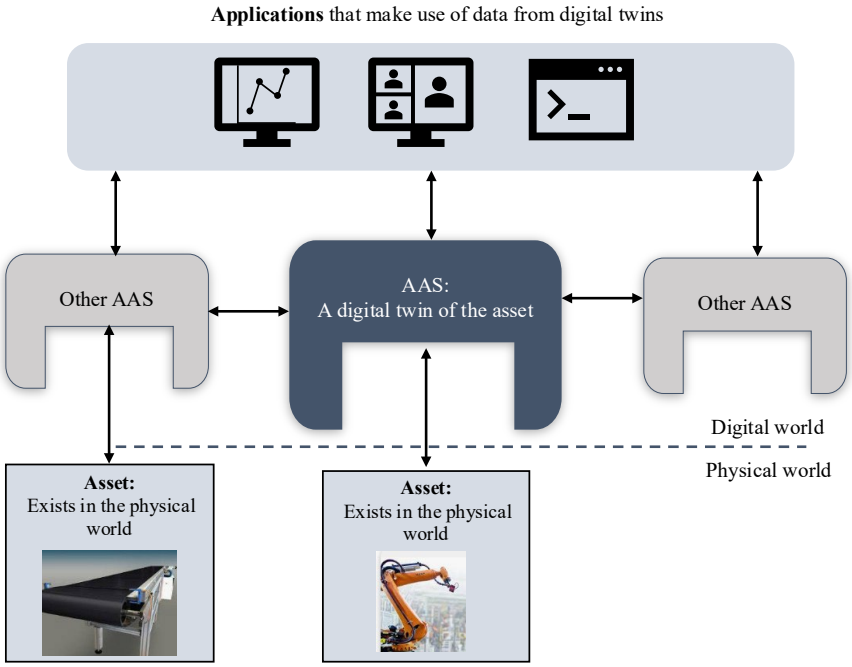


Fig. 55. Digital twins (AASs) and connected devices and interactions.

Fig. 56 illustrates the core structure of the AAS information model and what is represented and stored inside the AAASX (.aasx) package of files:

- **AssetAdministrationShell:** The top-level element representing the digital twin of an asset. The AAS provides identification and administrative information and maintains references to the associated asset and its submodels. The self-referencing association (derivedFrom) establishes an optional traceability link to another Asset Administration Shell from which the current AAS originates. This relationship

can be used to reference predecessor versions, template versions, or other source AASs, but it does not represent inheritance in the object-oriented sense.

- **Submodel (SM):** A mechanism for organizing information that naturally belongs together within a specific domain or use case. An AAS can contain one or more submodels. Examples include a digital nameplate containing submodels for manufacturer information, device identification and design parameters, and technical drawings. For AAS deployed for lifecycle follow-up, submodels for operational and maintenance data are relevant.
- **Submodel element/ or collection** (SME or SMC): A sub-structure within a submodel that organizes properties within a submodel. A submodel element can be a property, an operation, or similar, while the submodel element collection groups them into meaningful categories.

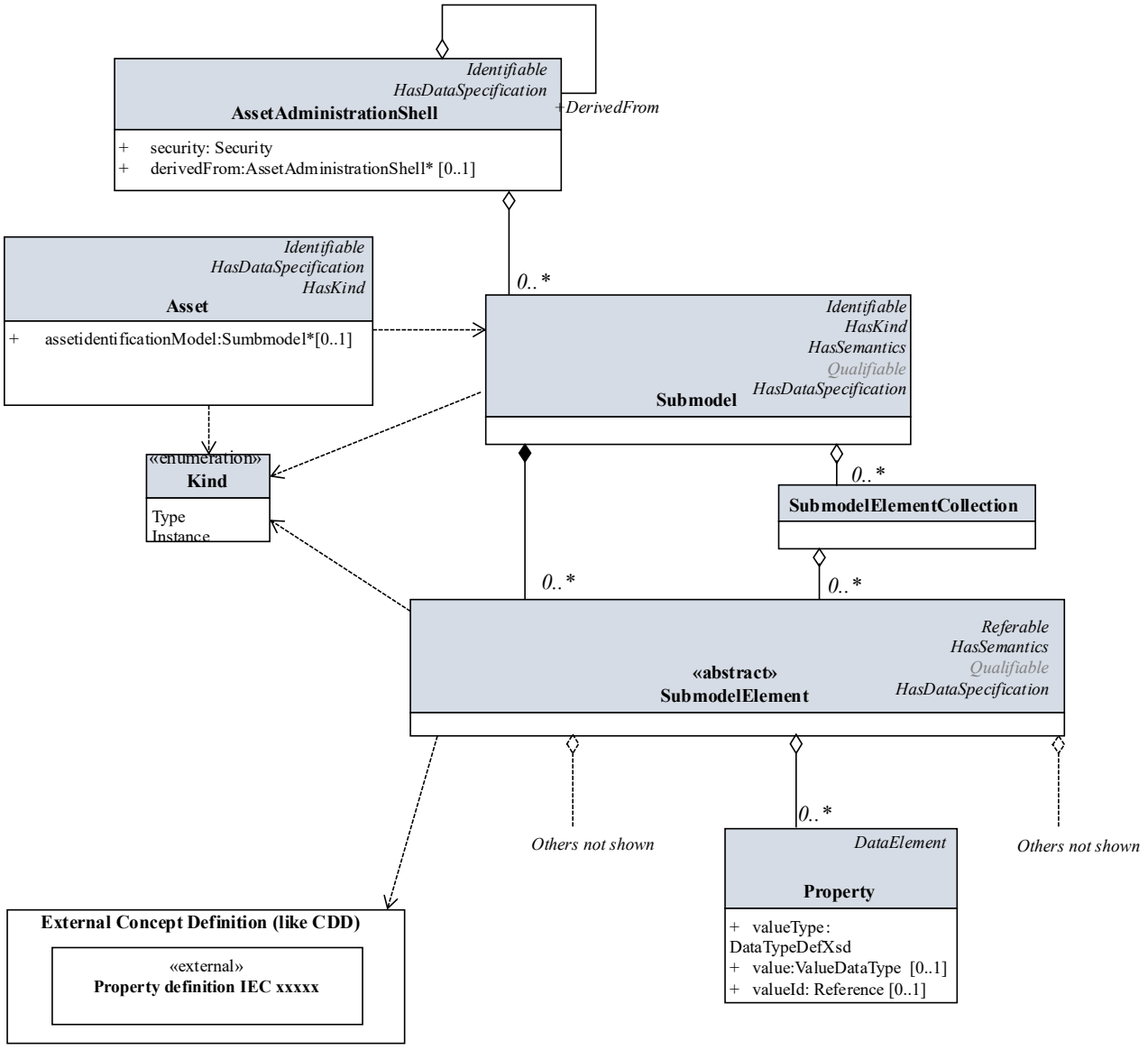


Fig. 56. AAS metamodel example (based on IEC 63278-1)

- **Property:** A basic data element used to describe an asset or one of its characteristics. Examples include manufacturer name, device name, measured speed, or rated speed. A property is supplemented with metadata such as the data type (e.g., string, integer, or real number), engineering units, and semantic references that provide a standardized machine-readable meaning through external dictionaries or concept definitions.

- **Asset:** The representation of the physical or logical object to which the Asset Administration Shell refers. The asset may represent a physical device, component, system, software application, or another identifiable entity in the real world.
- **Kind:** Indicates whether an element represents a Type or an Instance. A Type defines a generic template for organizing data for a type of asset, while an Instance represents the use of that template for a specific asset, with actual values and attributes. For example, an AAS template (type) for a motor may require specifying the serial number and rated voltage, along with the associated value formats for each, whereas a motor instance represents an individual motor with the values filled in. The association arrows from Asset, SM, and SMC to Kind do not represent inheritance or containment; rather, each can be classified as either a type or an instance.

More details on the metamodel are provided in IEC 63278-2 (currently not published), Platform Industrie 4.0 (2022), and what has been pre the OPC Foundation.

For AAS to be useful, it is important for industry to develop templates for typical use cases and systems. The leading organization for this effort is the International Digital Twin Association (IDTA), which organizes working groups that develop templates that are eventually published by IDTA. By 2026, around 100 AAS templates for digital nameplate, time series data, handover documentation, Functional safety and reliability, battery data, predictive maintenance, and artificial intelligence datasets will be either in progress or already published.

The details for how the AAS files are programmed in software are specified with UML metamodels. The highest-level UML model for an AAS is shown in Fig. 56. The actual construction of an AAS can be done using proprietary tools or open-source software such as the AASX package explorer.

Fig. 57 shows the user interface of the AASX package where the .aasx file for a Siemens safety controller has been opened. The software is not that user-friendly, but the AASX package explorer GitHub hosted by IDTA provides useful videos and samples. The alternative to using an editor like this is to program the .aasx files. The file is actually a collection of files organized into a zip file that can be extracted into individual JSON or XML formats.

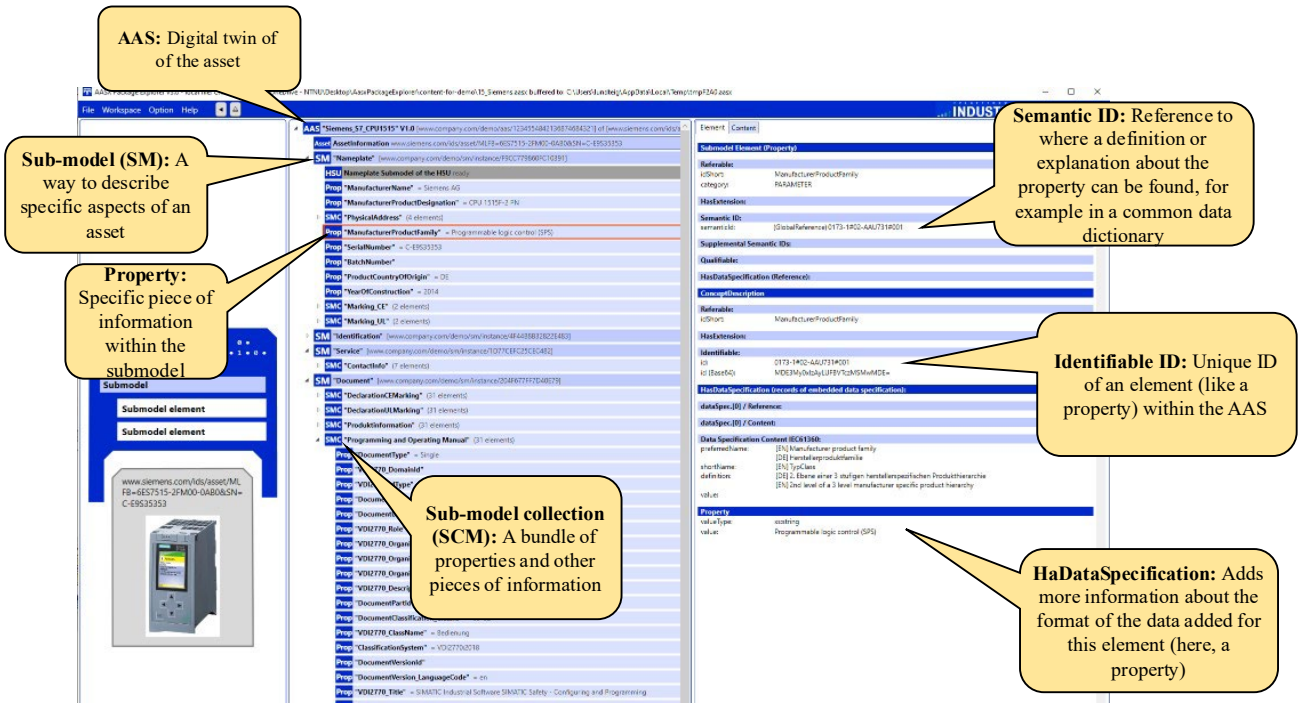


Fig. 57. Example of an AAS generated in the AASX package explorer.

AAS is most useful when employed throughout the entire system lifecycle, with type 1 and type 2 implementations. If only applied for fil-exchange, AML may be a better alternative. Fig. 58 identifies an example:

Engineering:

- Request the manufacturers to provide instance documentation as .assx files.
- Enrich the AAS file with more submodels and details as the design and engineering continue, and use the .aasx format when exchanging data between stakeholders and applications.
- Add new AAS as needed to organize individual AAS of products and subsystems.

Operation:

- Set up AAS servers to enable a "living" digital twin: Import AAS files to the AAS server. Make necessary connections to source systems on the OT side, for example, via OPC UA server(s).
- Store .aasx files: Maintain a repository with .aasx files with version control.

Want to check for yourself and invest a bit of time? You can download the AAS package explorer from https://github.com/eclipse-aaspe/package-explorer/releases. Samples to download and open in the tool are found here: https://www.admin-shell-io.com/samples/. The server to store generated AASs can be downloaded from https://github.com/eclipse-aaspe/server.

	You may take a look at this presentation of how AAS can be applied at YouTube

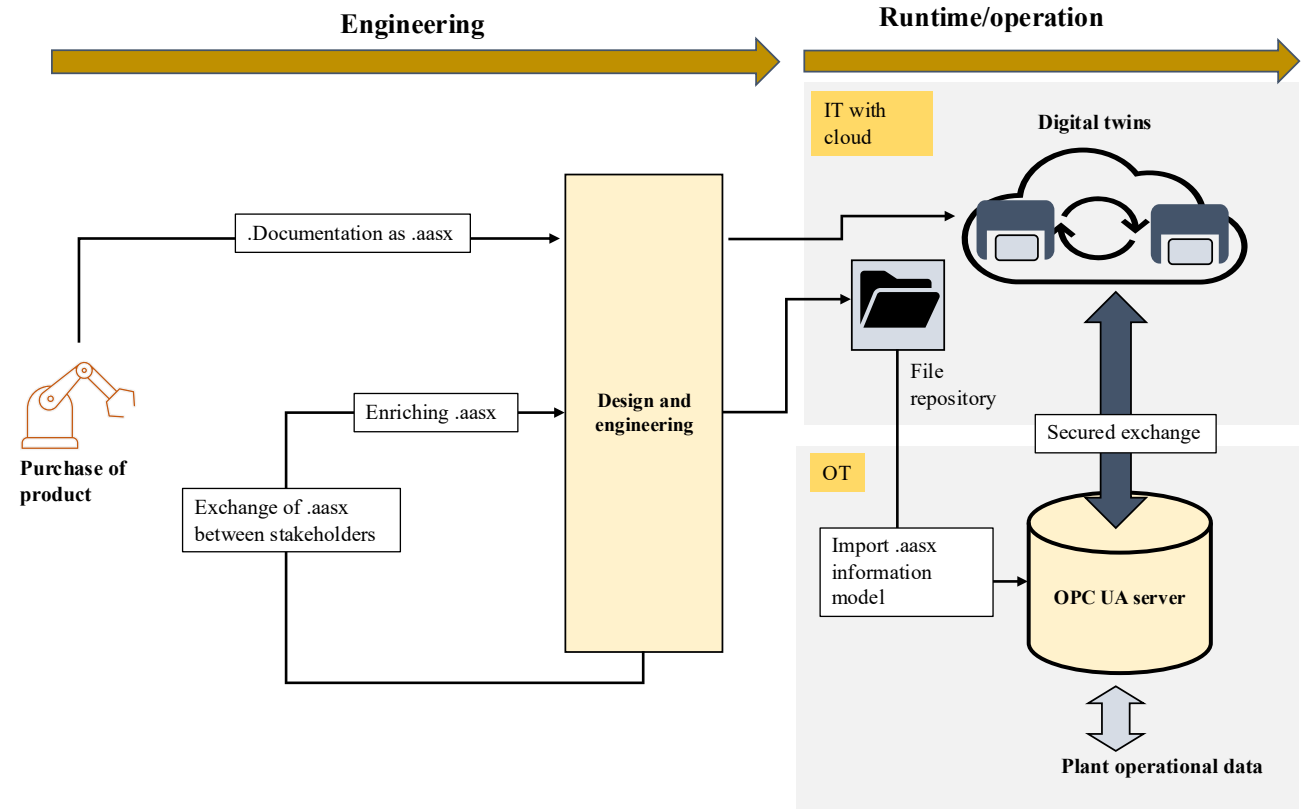


Fig. 58. AAS applied to the lifecycle

2.10.5 Open process automation platform (O-PAS)

O-PAS stands for Open Process Automation Standard and was developed by the Process Automation Forum (OPAF), part of https://www.opengroup.org. The main goal of O-PAS is to standardize the interfaces of devices connecting to the network, enabling more "plug-and-play" for systems in the OT and connected cloud services. For example, if you want to replace a Siemens PLC with an ABB PLC, it should be as easy as removing the Siemens PLC and plugging in the ABB PLC, as long as the application program covers the necessary functions.

So far, the development of O-PAS devices has focused on integrating controllers for real-time communication. Future expansions will also cover more types of real-time applications, including safety-instrumented systems and devices for secure data exchange with cloud solutions. The latter aligns with the aim of the NAMUR open architecture, and this thus requires interaction between OPAF and NAMUR, which has already begun.

The architecture in Fig. 59 identifies the main O-PAS devices:

- OCF: The OPAS Connectivity Framework is a standardized communication backbone, or communication layer with the necessary OSI layers implemented, that allows devices with a DCN interface to exchange data reliably and securely. The communication is based on OPA UA, meaning it is a technology platform independent of hardware and software.
- DCN (Distributed control node), an add-on or built-in O-PAS device based on the O-PAS specification, can manage real-time control and I/O, and replace proprietary protocol solutions. The DCN can be built into a device (sensor or controller) or added as a hardware or software interface.
- OCI (O-PAS communication interface) is a software part built into the product. It allows different suppliers to access and configure their equipment connected to OCF with their own proprietary equipment and software. This equipment does not have DCN functionality for real-time operation, and the primary purpose of OCI is to retrieve or update configuration data and other data not used in real-time operation (such as condition monitoring).

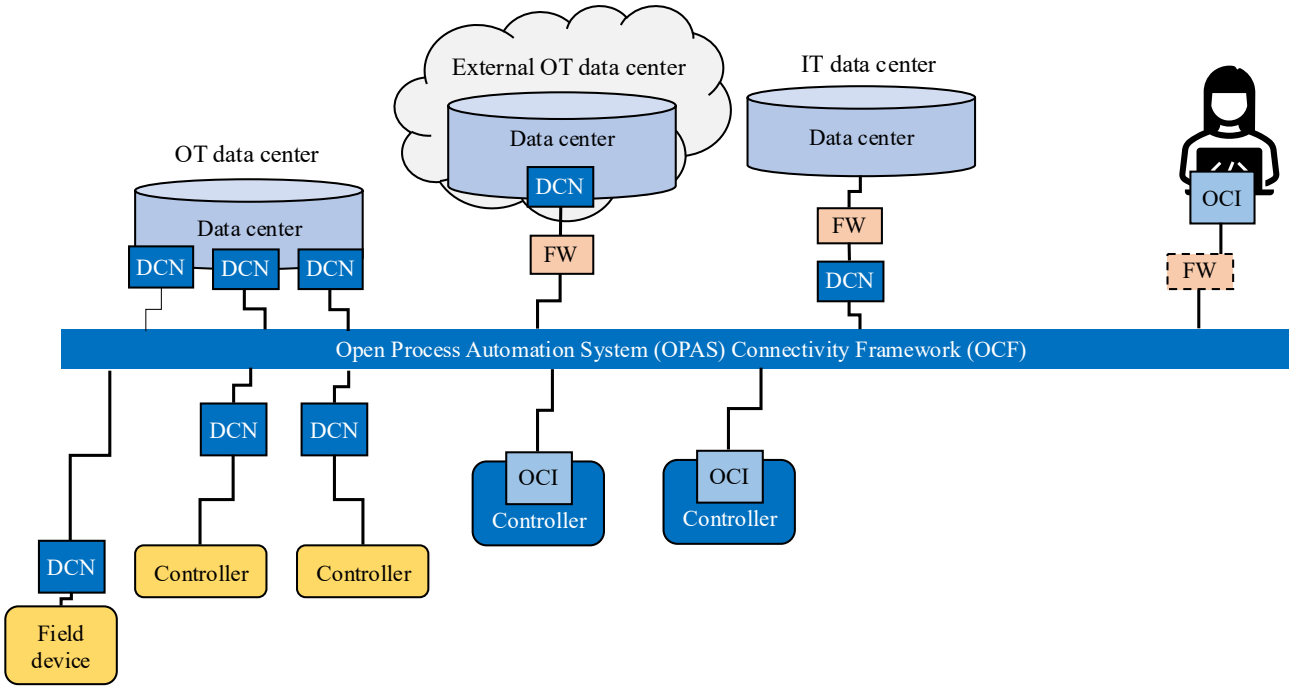


Fig. 59. O-PAS architecture (Inspired by OPAF webpages)

O-PAS has gained the most interest in facilities that are highly modularized and where different production systems or cells are likely to be replaced or reconfigured from time to time. In such situations, the “plug-and-play” approach for fast, efficient integration of new and configured systems is an attractive solution.

2.10.6 IEC CDD

Industry associations and international standardization bodies are working to develop machine-readable codes for equipment, processes, and operations. The IEC Common Data Dictionary (IEC CDD) and ECLASS are two relevant examples for the process industry.

Machine-readable codes are essential for ensuring interoperability between systems that exchange data, as they enable systems to interpret the same information consistently. A machine-readable code allows machines to to distinguish the meanings of semantically similar or identical words and terms. For example, assume you intend to purchase a button for your jacket and send a request from your machine to another machine that will place the order. Unfortunately, you cannot guarantee that the computer receiving the request to purchase the button will interpret it in the same way you do. Alternative interpretations could be:

- Emergency stop button for a machine
- Button (like a pin) on your jacket
- Button for clothes

As humans, we would grasp the correct meaning because we know the context. However, machines may be unable to do this, leading to wrong information.

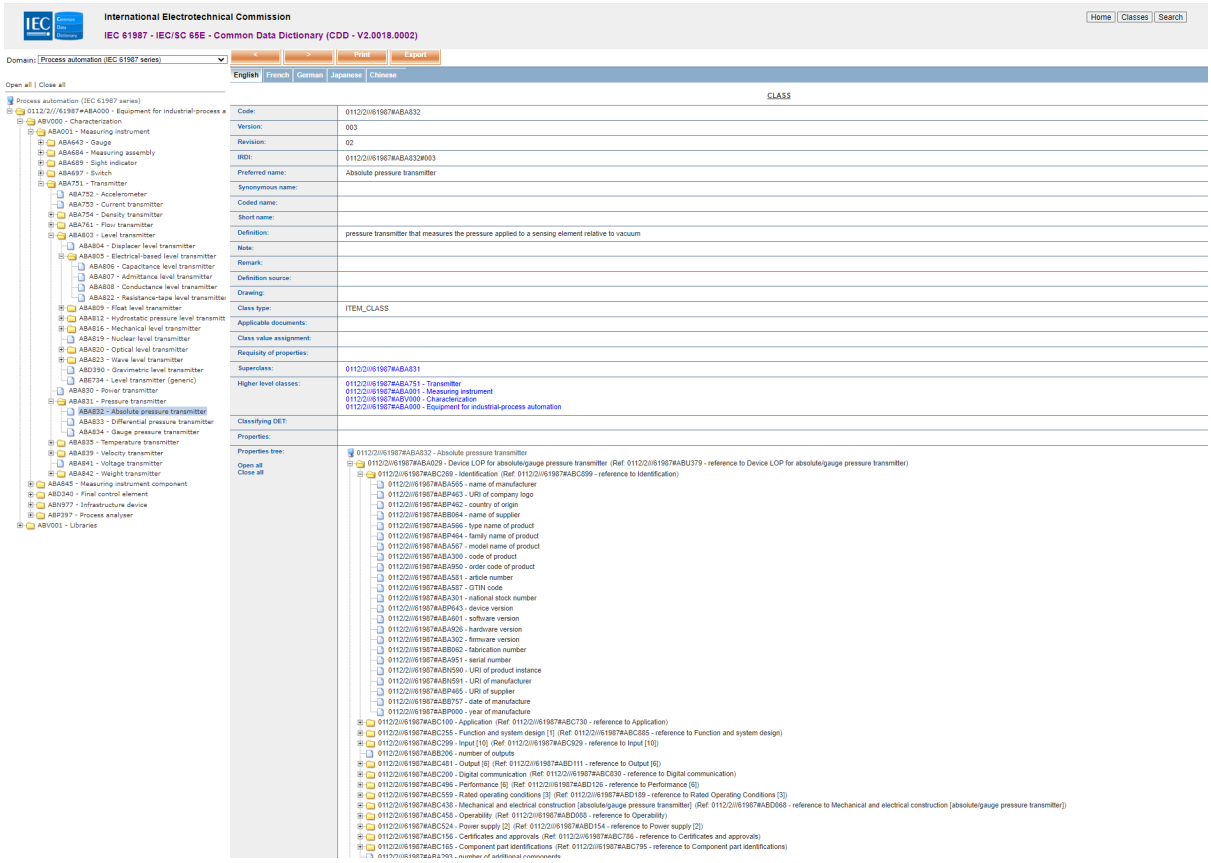


Fig. 60. Extract from IEC CDD for equipment properties relevant for e.g., IEC 61987 standard

IEC CDD is a public repository of machine-readable codes, associated terms, and definitions that cover a wide range of equipment. The machine-readable code is formatted according to an international standard called IRDI - International Registration Data Identifier. A lookup in the IEC CDD for code 0112/2///61987#ABA832 (see Fig. 61) indicates that it represents an absolute pressure transmitter. The content of this code is interpreted as follows:

- The number “0112” is the international code designator, where 0112 means ISO or IEC
- The number “2” means the organization identifier, where 2 means IEC.
- The number “61987” means the number of the IEC standard used as a basis for defining the term.

- The mixed letter and number code" ABA831 means the unique item code for the term (in our case, for the pressure transmitter)

In the datasheet, the code is linked to human-readable information, such as the preferred name (Absolute pressure transmitter) and the definition (a pressure transmitter that measures pressure applied to a sensing element relative to vacuum).

2.11 Bibliography

Drath, R., Luder, A., Peschke, J., & Hundt, L. (2008). AutomationML - the glue for seamless automation engineering. 2008 IEEE International Conference on Emerging Technologies and Factory Automation, Hamburg, Germany.

EN 50159. (2020). *Railway applications - Communication, signalling and processing systems - Safety-related communication in transmission systems*. CENELEC.

EN 50325-1. (2019). *Industrial communications subsystem based on ISO 11898 (CAN) for controller-device interfaces Part 1: General requirements*. CENELEC.

EN 50325-4. (2022). *Industrial communications subsystem based on ISO 11898 (CAN) for controller-device interfaces. Part 4: CANopen*. CENELEC.

EN 50325-5. (2010). *Industrial communications subsystem based on ISO 11898 (CAN) for controller-device interfaces Part 5: Functional safety communication based on EN 50325-4*. CENELEC.

FieldComm Group. (2021). *Ethernet - To the field*. https://www.ethernet-apl.org/.

IEC 61158-1. (2023). *Industrial communication networks - Fieldbus specifications - Part 1: Overview and guidance for the IEC 61158 and IEC 61784 series*. International Electrotechnical Commission.

IEC 61508. (2010). *Functional safety of electrical/electronic/programmable electronic safety-related systems (Seven parts)*. International Electrotechnical Commission.

IEC 61508-2. (2010). *Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 2: Requirements for electrical/electronic/programmable electronic safety related systems*. International Electrotechnical Commission.

IEC 61784-1. (2017). *Industrial communication networks - Profiles - Part 1: Industrial communication networks - Profiles* International Electrotechnical Commission.

IEC 61784-3. (2021). *Industrial communication networks - Profiles - Part 3: Functional safety fieldbuses - General rules and profile definitions*. International Electrotechnical Commission.

IEC 61784-3-3. (2021). *Industrial communication networks - Profiles - Part 3-3: Functional safety fieldbuses - Additional specifications for CPF 3*. International Electrotechnical Commission.

IEC 61918. (2024). *Industrial communication networks - Installation of communication networks in industrial premises*. International Electrotechnical Commission.

IEC 62280. (2014). *Railway applications - Communication, signalling and processing systems - Safety related communication in transmission systems*. International Electrotechnical Commission.

IEC 62439-1. (2016). *Industrial communication networks - High availability automation networks - Part 1: General concepts and calculation methods*. International Electrotechnical Commission.

IEC 62439-2. (2021). *Industrial communication networks - High availability automation networks - Part 2: Media Redundancy Protocol (MRP)*. International Electrotechnical Commission.

IEC 62541. (2020). *OPC Unified Architecture (Several parts)*. International Electrotechnical Commission.

IEC 62541-15. (2025). *OPC Unified Architecture - Part 15: Safety*. International Electrotechnical Commission.

IEC 62591. (2016). *Industrial networks - Wireless communication network and communication profiles - WirelessHART*. International Electrotechnical Commission.

IEC 62714-1. (2018). *Engineering data exchange format for use in industrial automation systems engineering*. International Electrotechnical Commission.

IEC 63278-1. (2023). *Asset Administration Shell for industrial applications - Part 1: Asset Administration Shell structure*. International Electrotechnical Commission.

IEC glossary. *IEC Electropedia: The World's Online Electrotechnical Vocabulary (webpage)*. International Electrotechnical Commission. Retrieved 15.05.24 from https://www.electropedia.org/

IEEE 802.3cg-2019. (2019). *IEEE Standard for Ethernet - Amendment 5: Physical Layer Specifications and Management Parameters for 10 Mb/s Operation and Associated Power Delivery over a Single Balanced Pair of Conductors*. Institute of Electrical and Electronics Engineers (IEEE).

ISO11898-2. (2024). *Road vehicles — Controller area network (CAN) — Part 2: High-speed physical medium attachment (PMA) sublayer*. International Organization for Standardization.

ISO 11898-1. (2024). *Road vehicles — Controller area network (CAN) — Part 1: Data link layer and physical coding sublayer*. International Organization for Standardization.

ISO/IEC 7498-1. (1994). *Information technology -- Open Systems Interconnection -- Basic Reference Model: The Basic Model*. International Electrotechnical Commission, International Organization for Standardization.

NAMUR. (2020). *NAMUR Open Architecture Overview (presentation) at https://www.namur.net/fileadmin/media_www/fokusthemen/20200710_NAMUR_NOA_Overview_EN.pdf*. NAMUR - Interessengemeinschaft Automatisierungstechnik der Prozessindustrie e.V.

NAMUR NE175. (2020). *NAMUR Open Architecture NOA Concept*. NAMUR – Interessengemeinschaft Automatisie-rungstechnik der Prozessindustrie e.V.

NE-43. (2021). *Standardization of the Signal Level for the Failure Information of Digital Transmitters*. NAMUR

Platform Industrie 4.0. (2022). *Details of the Asset Administration Shell - Part 1 The exchange of information between partners in the value chain of Industrie 4.0 (Version 3.0RC02)*. Federal Ministry for Economic Affairs and Climate Action.