EPFL

Week 6
Industrial Communication Protocols

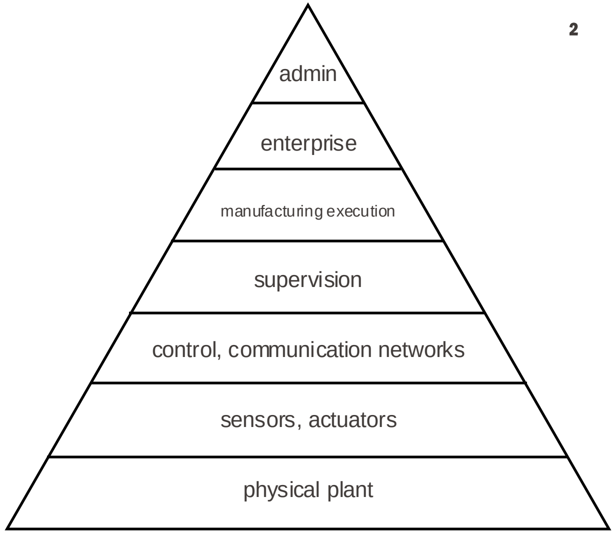
Dr. Philipp Sommer

Spring 2025

The material of this course has been initially created by Prof. Dr. H. Kirrmann

Outline

- Protocols Basics
- Device Management Protocols
  - HART
  - Modbus/Profibus/Profinet
  - Open Platform Communications (OPC UA)
- Message Queuing Telemetry Transport (MQTT)



What is a protocol?

- Communication Protocol defines:
  - How several entities communicate with each other
  - Message formats and rules

- Example of Protocols:
  - Hypertext Transfer Protocol (HTTP/HTTPS)
  - TCP/IP
  - File Transfer Protocol (FTP)
  - Simple Mail Transfer Protocol (SMTP)

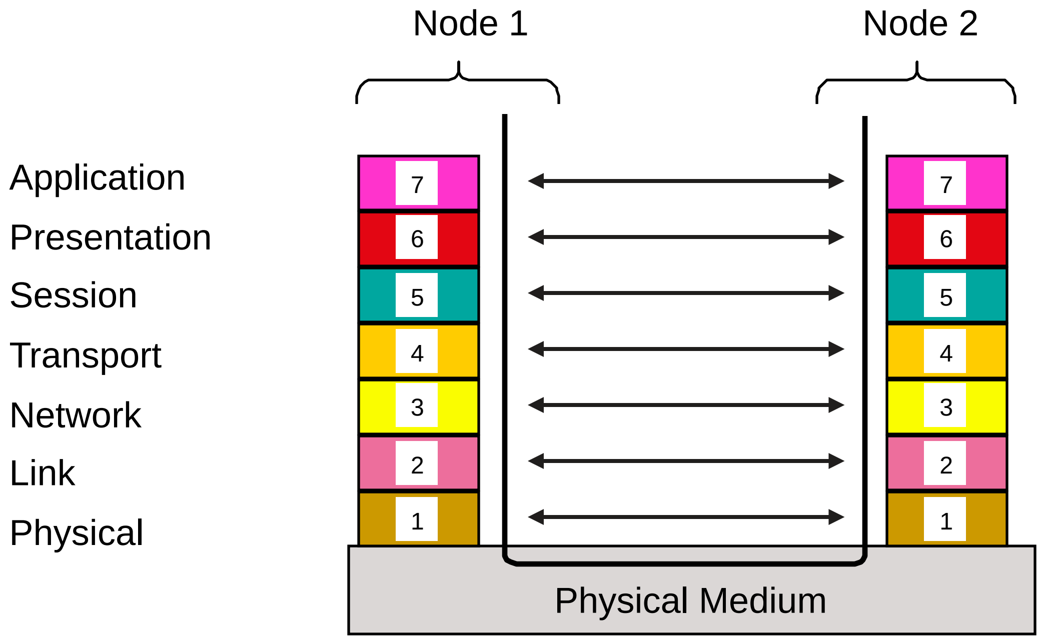
The OSI Model

- The Open System Interconnection (OSI) model is a standard way to structure communication software that is applicable to any network.
- was developed to structure telecommunication protocols in the 1970s (Pouzin & Zimmermann)
- standardized by CCITT and ISO as ISO / IEC 7498
- all communication protocols (e.g. TCP/IP) can be mapped to the OSI model.
- it's a model, not a standard protocol, but a suite of protocols with the same name has been standardized by UIT / ISO / IEC for open systems data interconnection (but with little success).

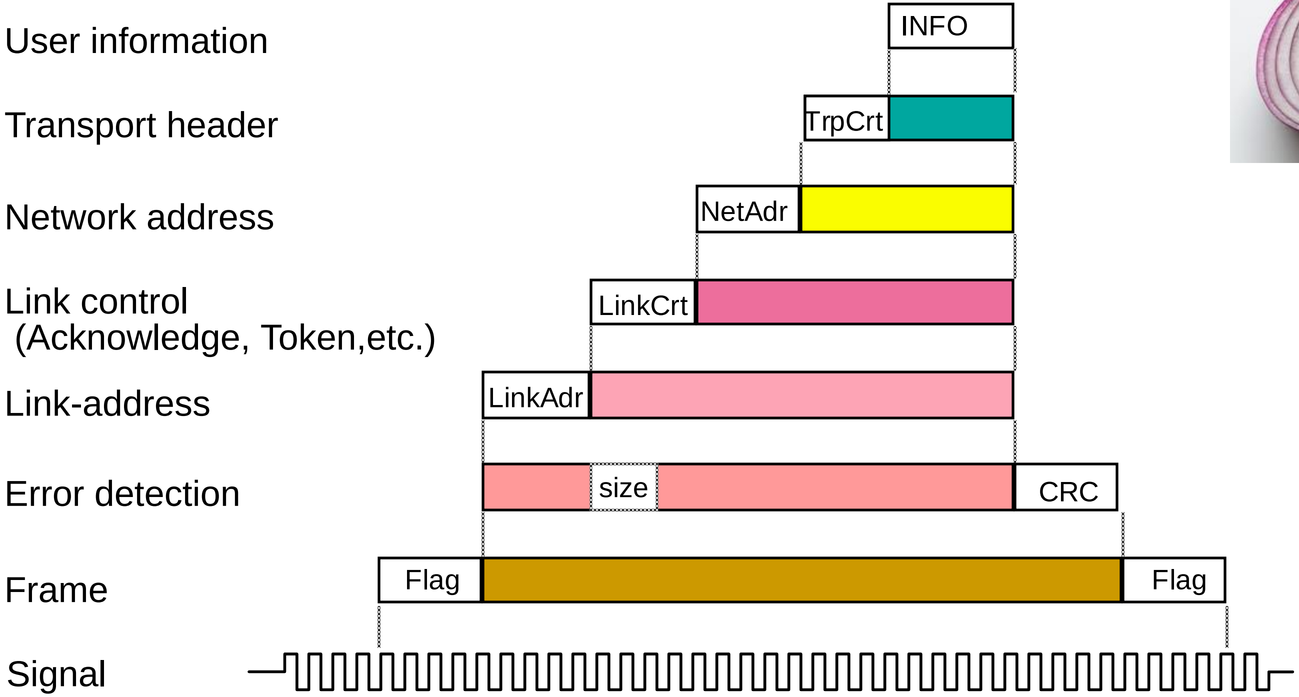
OSI Model (ISO/IEC standard 7498)

"Application" protocols	7	Application	All services directly called by the end user (Mail, File Transfer,...) e.g. Telnet, SMTP, HTTP
	6	Presentation	Definition and conversion of the data formats (e.g. ASN 1)
	5	Session	Management of connections (e.g. ISO 8326)
"Transport" protocols	4	Transport	End-to-end flow control and error recovery (e.g. TCP, TP4)
	3	Network	Routing, possibly segmenting (e.g. IP, X25)
	2	Link	Error detection, Flow control and error recovery, medium access (e.g. HDLC)
	1	Physical	Coding, Modulation, Electrical and mechanical coupling (e.g. Ethernet, RS485)

OSI Model with two nodes

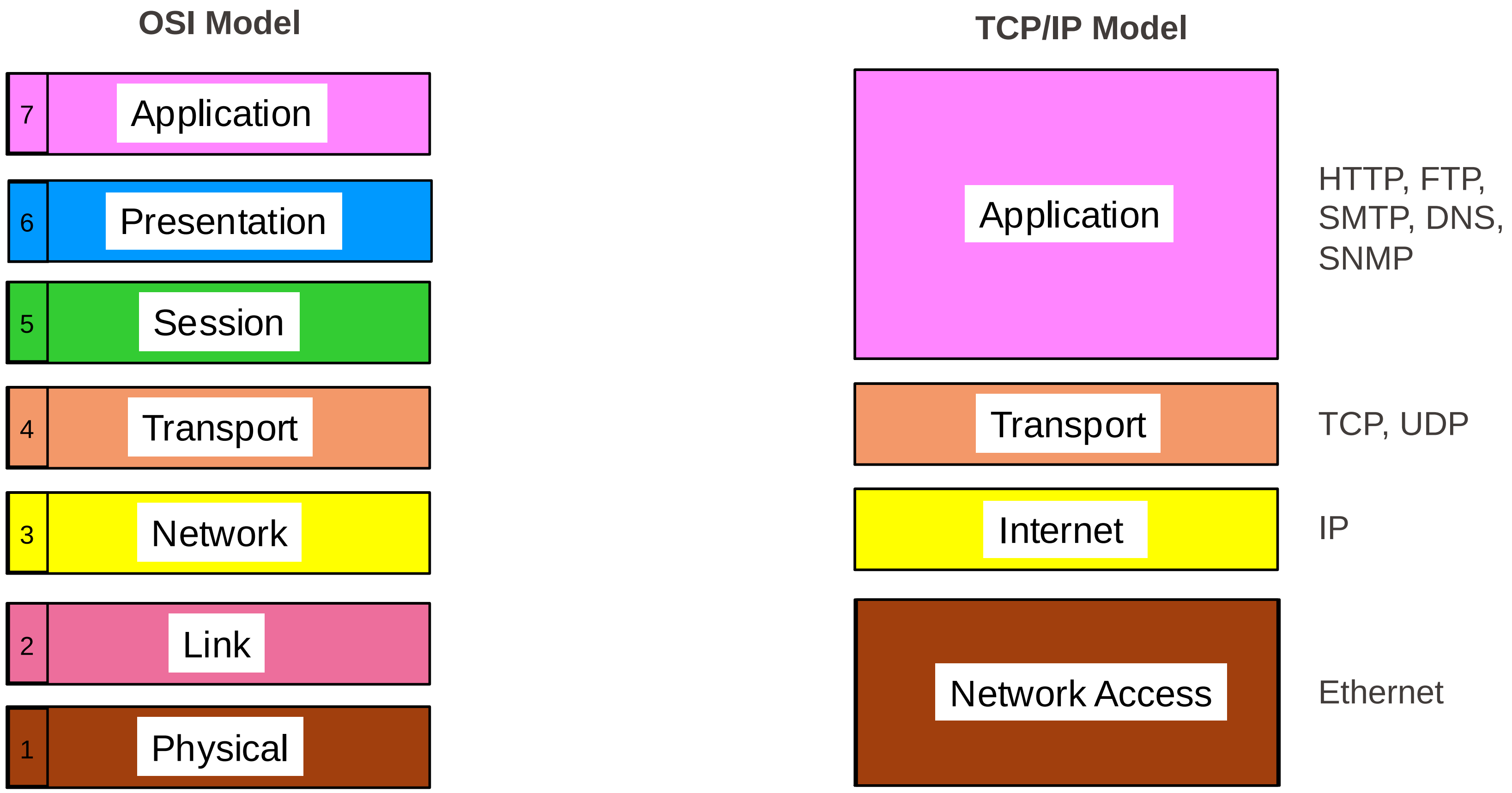


Encapsulation

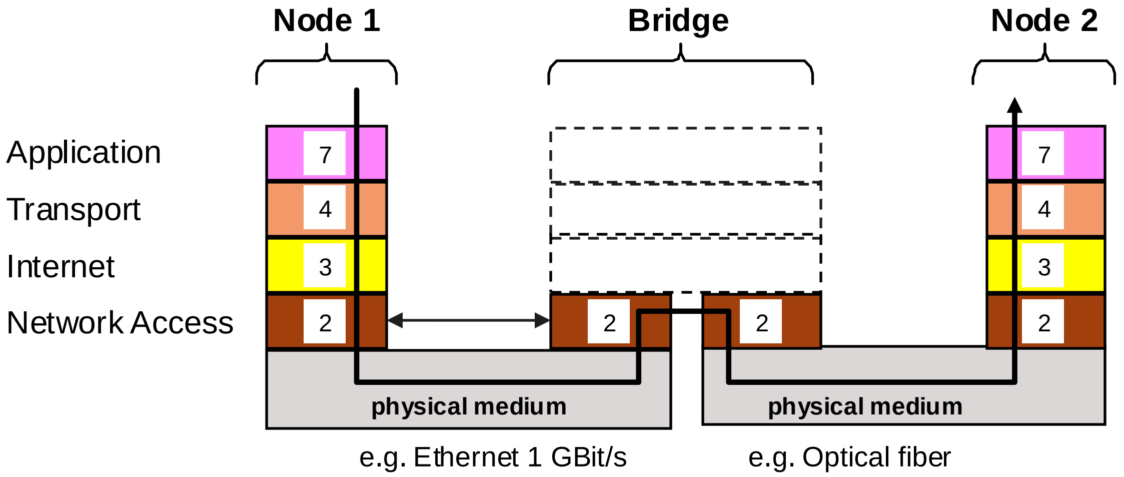


Each layer introduces its own header and overhead

OSI Model vs. Internet Protocol Suite



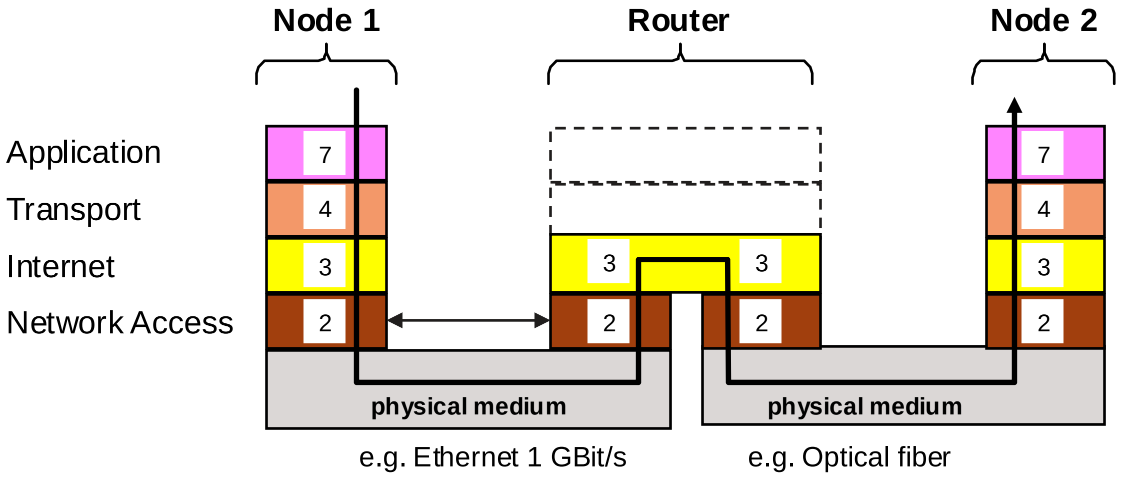
Ethernet Switch/Bridge (Layer 2)



The subnet on both sides of a bridge/switch have:
• the same frame format (except header)
• the same address space (different addresses on both sides of the bridge)
• the same network access protocol
Bridges/switches filter the frames on the base of their link addresses.

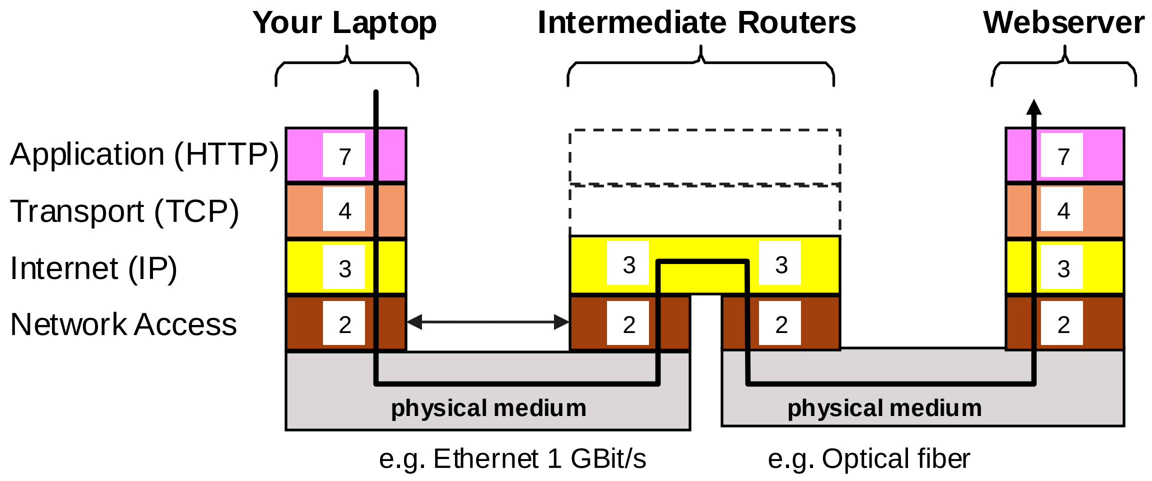
IP Router (Layer 3)

Cisco Catalyst 8500 Series



The router routes the frames on the base of their network address.
The subnets may have different link layer protocols.
Frames in transit are handled in the network layer.

Example: Browsing a Website



1. The client (browser) resolves the IP address of the webserver using the Domain Name System (DNS) protocol (e.g. epfl.ch -> 128.178.211.3)
2. A TCP connection is established to IP 128.178.211.3, port 80 (http).
3. The Hypertext Transfer Protocol (HTTP) is used to fetch the website.

Assessment

1. Name the layers of the OSI model and describe their function.
2. What are the reasons for using layered protocols?
3. What is the difference between a switch/bridge and a router?
4. What is encapsulation?
5. A system has an n-layer protocol hierarchy. Applications generate messages of length M bytes. At each of the layers, an h-byte header is added. What fraction of the network bandwidth is filled with headers?

Device Management Protocols

Device Management Protocols

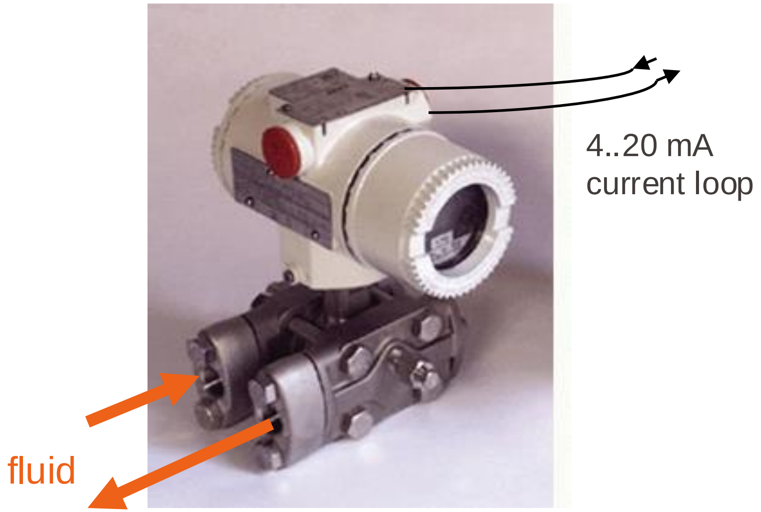
- Application layer protocols offering remote services for large networks of devices:
  - Monitoring (health of devices and network, get current state)
  - Management (deployment, configuration, change settings)

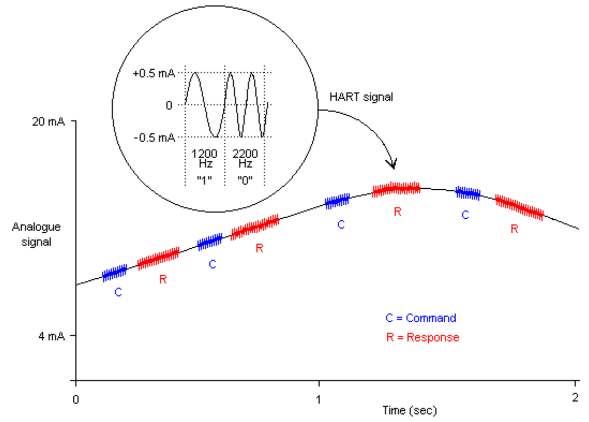
Examples:

- HART
- Modbus
- OPC UA

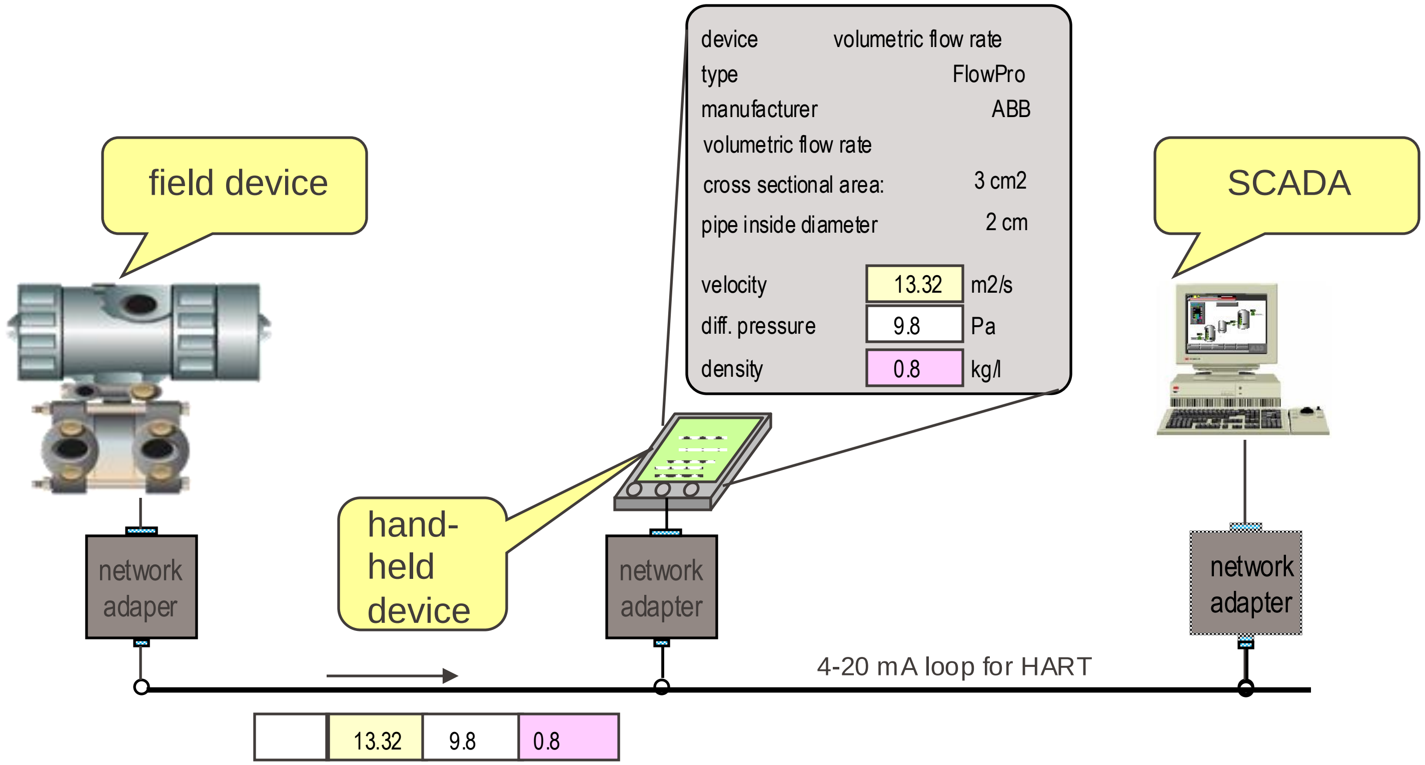
HART

- Data over 4..20 mA loops
- Practically all 4..20mA devices come equipped with HART today.
- Millions of devices supporting HART are installed worldwide.



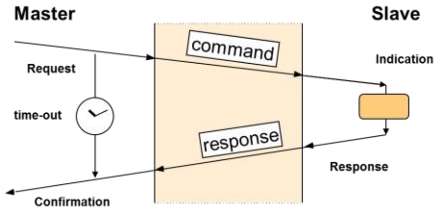


HART Device Access



device	volumetric flow rate	
type	FlowPro	
manufacturer	ABB	
volumetric flow rate		
cross sectional area:	3 cm2	
pipe inside diameter	2 cm	
velocity	13.32	m2/s
diff. pressure	9.8	Pa
density	0.8	kg/l

HART commands summary



Universal Commands	Common Practice Commands	Device-Specific Commands (example)
• Read manufacturer and device type • Read variable and units • Read current output and percent of range • Read up to four predefined dynamic variables • Read/write tag, descriptor, date • Read/write 32-character message • Read device range values, units, and damping time constant • Read or write final assembly number • Write polling address	• Read selection of up to four dynamic variables • Write damping time constant • Write device range values • Calibrate (set zero, set span) • Set fixed output current • Perform self-test • Perform master reset • Trim variable zero • Write variable unit • Trim DAC zero and gain • Write transfer function (square root/linear) • Write sensor serial number • Read or write dynamic variable assignments	• Read or write low-flow cut-off • Start, stop, or clear totalizer • Read or write density calibration factor • Choose variable (mass, flow, or density) • Read or write materials or construction information • Trim sensor calibration • enable PID, write PID setpoint • Valve characterization • Valve setpoint • Travel limits • User units • Local display information

Modbus

- Protocol was published in 1979 by Modicon (now Schneider Electric)
- Works with several transport layers: Serial line, Ethernet, TCP/IP
- Client/server architecture (server holds the objects of interest)
- Object types:
  - Coil (discrete output): read/write, 1 bit
  - Discrete input: read-only, 1 bit
  - Input register: read-only, 16 bits
  - Output holding register: read-write, 16 bits
- Protocol versions / physical layers:
  - Modbus RTU (Remote Terminal Unit), serial communication (RS-485)
  - Modbus TCP, based on TCP/IP
  - Other variants (not widely used)

Modbus Example

- Scenario: Motor controlled by an ABB ACS-880 variable speed drive

1. Set the speed reference register:

   register = ACS880_SPEED_REFERENCE_ADDRESS
   value = 1000

2. Start the drive by writing to the drive control register:

   register = ACS880_START_STOP_ADDRESS
   value = 1

3. Stop the drive by writing to the drive control register:

   register = ACS880_START_STOP_ADDRESS
   value = 0

Profibus DP

- Profibus (Process Field Bus) is a field bus standard developed in the 1990s
- Two variants are in use today:
  - PROFIBUS DP (Decentralised Peripherals)
  - PROFIBUS PA (Process Automation)
- Twisted pair cables, bit rates from 9.6 kbit/s to 12 Mbit/s
- Cable length up to 1200m between repeaters

Source: https://en.wikipedia.org/wiki/Profibus

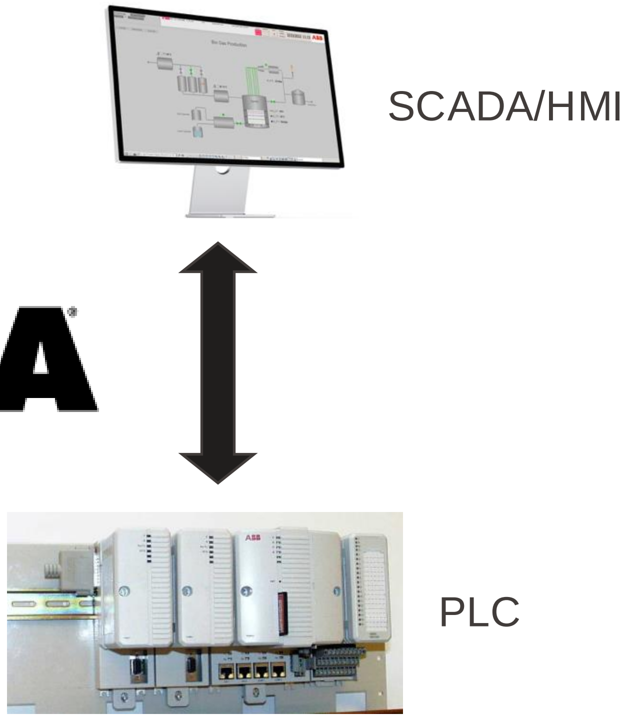
PROFINET

- PROFINET (Process Field Network) is an industrial communication standard introduced in 2003
- Most widely used industrial communication protocol in 2024
- Based on top of Ethernet communication
- Can be used with Time Sensitive Networking (TSN) for time critical applications

Assessment

1. What is the purpose of the HART protocol?
2. Which communication is used between a hand-held and a field device?
3. Which categories of HART commands do exist?

Open Process Control (OPC)

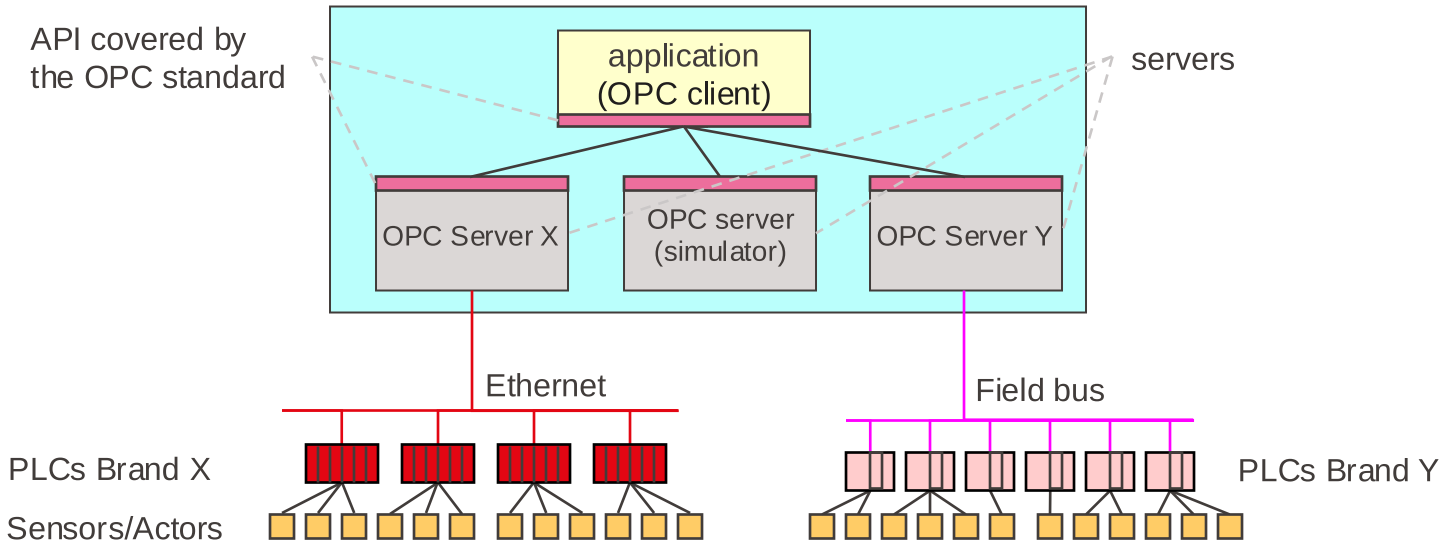


Open Process Control: API for Automation

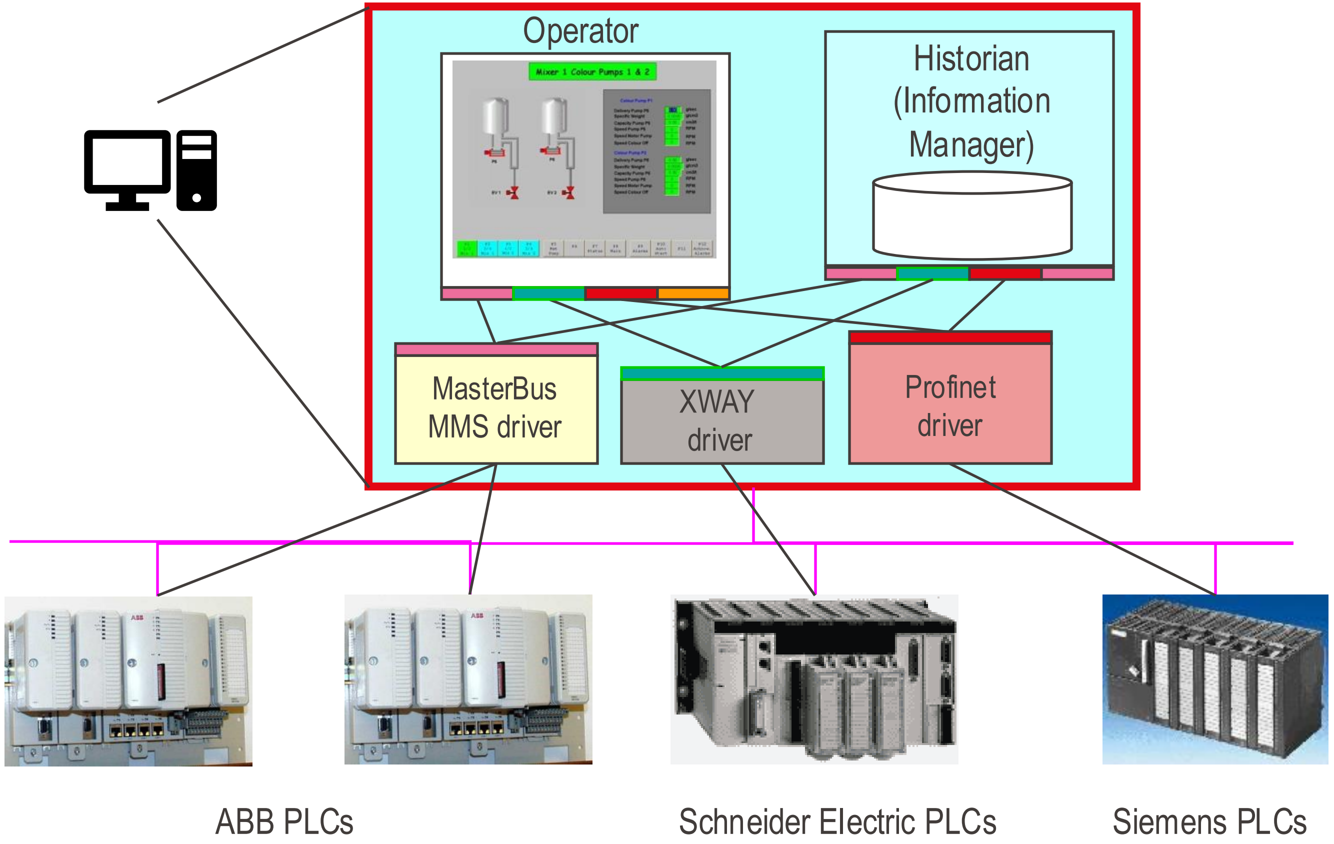
- Manufacturer-independent application programming interface (API)
  - To implement clients which can access plant data coming from remote devices (PLCs, field bus devices, real-time databases) easily
  - Set of commands to access OPC servers
- OPC clients
  - Read and write process variables, read alarms and events, acknowledge alarms, retrieve historical data from databases according to several criteria
  - Implemented on automation platforms (e.g. ABB 800xA), which may act themselves as OPC servers to publish their data, events and historical data.
- OPC server
  - Supplied by manufacturer of automation devices
  - Communicates with its devices through the OPC protocol
  - Manages several devices of the same type, several servers can run in parallel, each server can be accessed by several clients in same network

What is OPC?

Industry standard set up by the ***OPC Foundation*** *(http://www.opcfoundation.org/)* specifying the software interface (objects, methods) to a server that collects data produced by field devices and programmable logic controllers.



Before OPC


Operator
Mixer 1 Colour Pumps 1 & 2
Historian
(Information
Manager)
MasterBus
MMS driver
XWAY
driver
Profinet
driver
ABB PLCs
Schneider Electric PLCs
Siemens PLCs

With OPC

application software is written independently from the type of controller

Proprietary drivers still exist, clients do not see them anymore

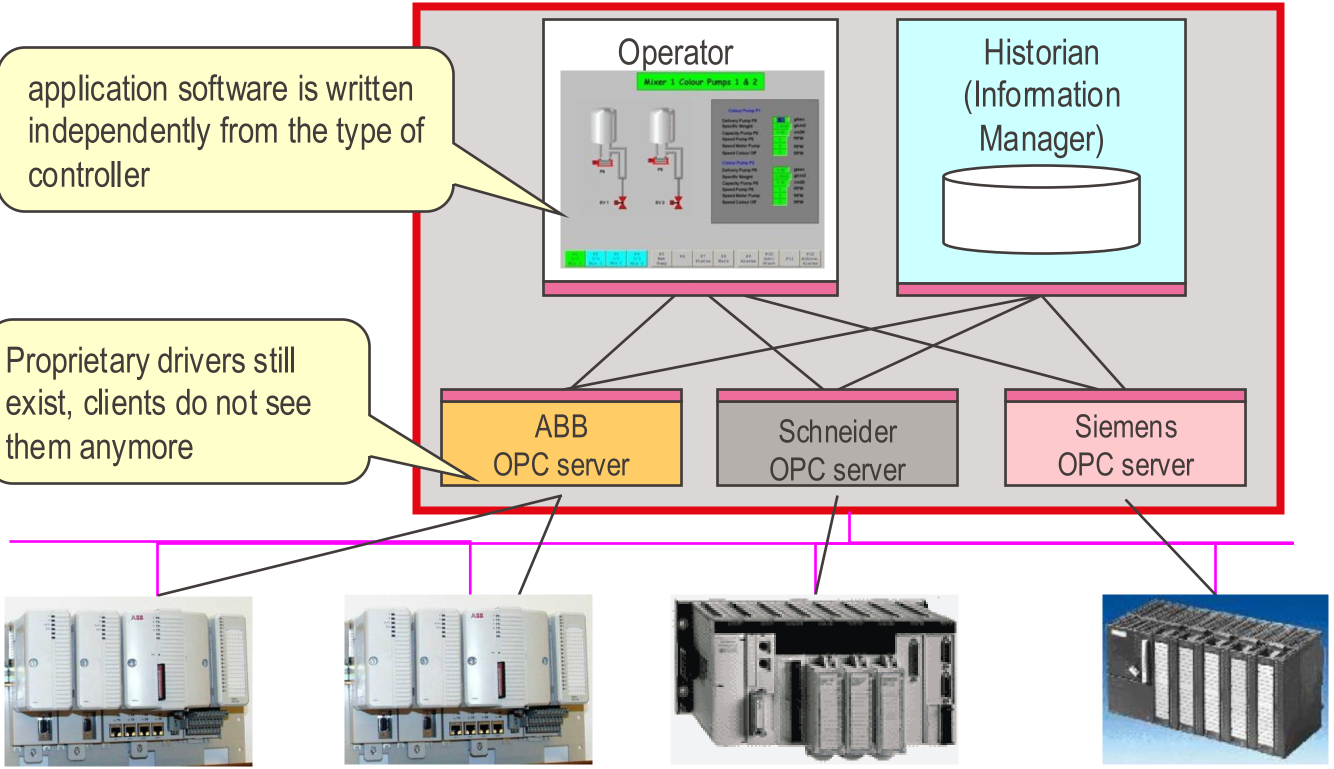


ABB PLCs

Schneider Electric PLCs

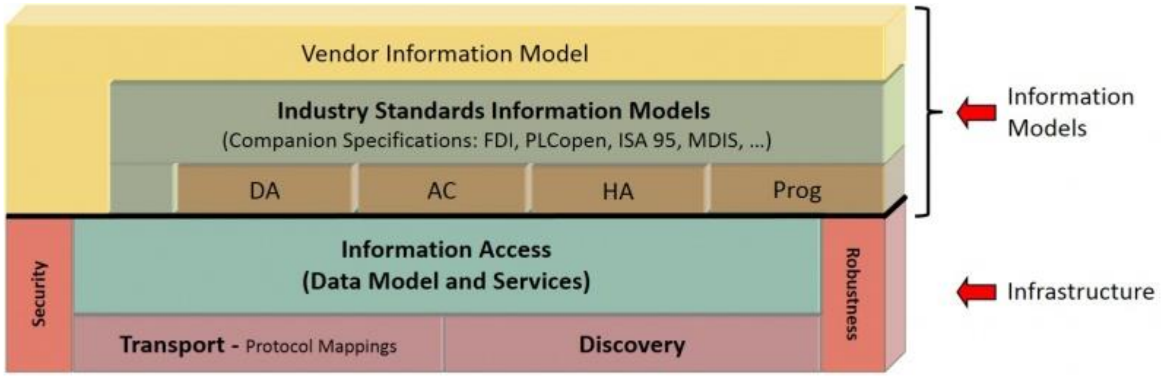
Siemens PLCs

History & Importance of OPC

- Greatest improvement in automation since PLC programming language standardization (IEC 61131).
- Thousands of suppliers offer OPC compatible products to connect their PLCs, field bus devices, displays and visualization systems.
- Used for data exchange between applications and for accessing databases.
- **OPC Classic** was established in the 1990s and was based on Microsoft Windows COM and DCOM (Distributed Component Object Model) technology.
- **OPC UA** (Unified Architecture) became available in 2006:
  - Agnostic of the operating system (Windows, Linux, RTOS, etc…)
  - Open standard, implementable under GPL 2.0 license
  - Service-oriented Architecture (SOA)
  - Built-in security features (encryption/authentication)
  - Complex standard (1250 pages)
  - Several open-source libraries available to implement OPC UA in C/C++, Python, Java, ..

OPC UA (IEC 62541) – Service Oriented Architecture

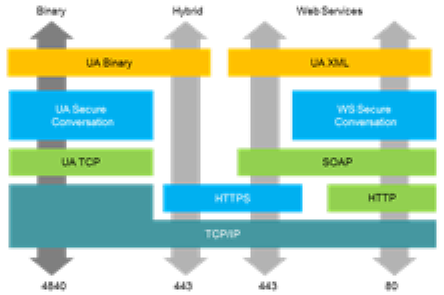
- OPC UA is a service-oriented communication framework for the exchange of information models, replacing OPC “Classic” which was based on Microsoft DCOM communication.
- Service provider receives requests, processes them and sends results back.


Vendor Information Model
Industry Standards Information Models
(Companion Specifications: FDI, PLCopen, ISA 95, MDIS, ...)
DA
AC
HA
Prog
Information Access
(Data Model and Services)
Security
Robustness
Transport - Protocol Mappings
Discovery
Information Models
Infrastructure

http://wiki.opcfoundation.org/index.php/Main_Page

OPC UA Communication

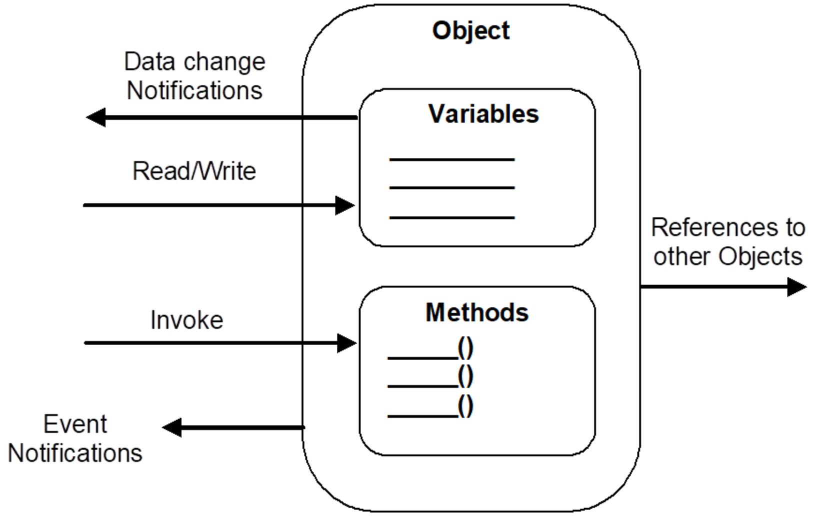
- OPC UA does not only standardize the interfaces, but also the transmitted data and enables encryption and authentication of process data.
- Two main protocol bindings available:
  1. optimized TCP-based binary protocol
  2. HTTP/HTTPS web service with binary or XML messages



More info: https://opcfoundation.org/wp-content/uploads/2016/05/OPC-UA-Interoperability-For-Industrie4-and-IoT-EN-v5.pdf

OPC UA Information Model

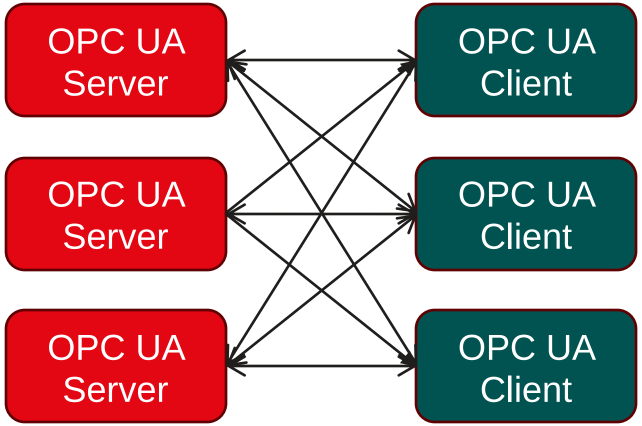
- Browse model instances and their references
- Read/write current and historical data
- Notify data change and events
- Execute methods



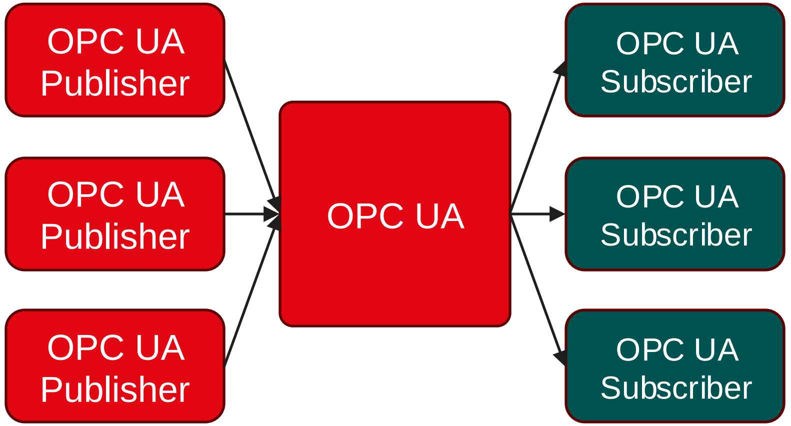
https://reference.opcfoundation.org/v104/Core/docs/Part3/4.2/

OPC UA Communication: Client-Server vs Pub-Sub

- Client – Server



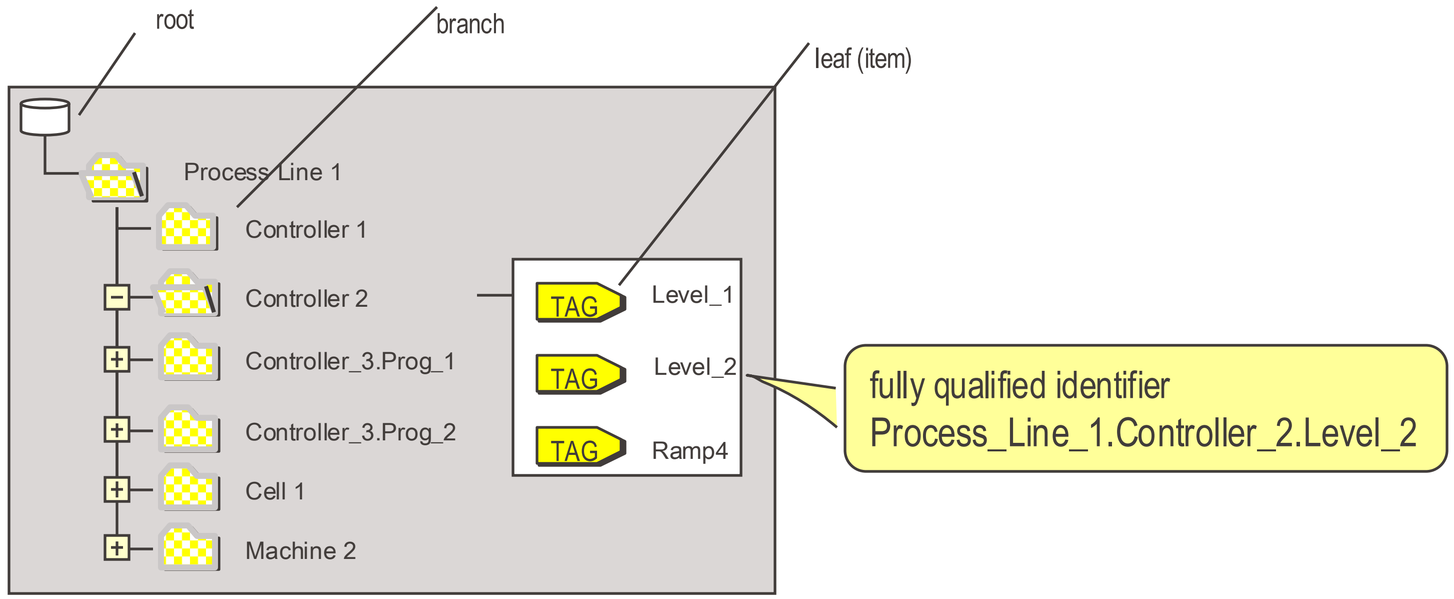
- Publisher – Subscriber (PubSub)



OPC UA - Discovery

- Address space is structured hierarchically for interoperability. Top levels are standardized for servers.
- Discovery mechanisms:
  - identification of devices and their functions within a network.
  - aggregation across subnets and intelligent, configuration-less procedure (e.g. Zeroconf) to identify and address network participants.
- Ability to browse a OPC server's address space to list all nodes.

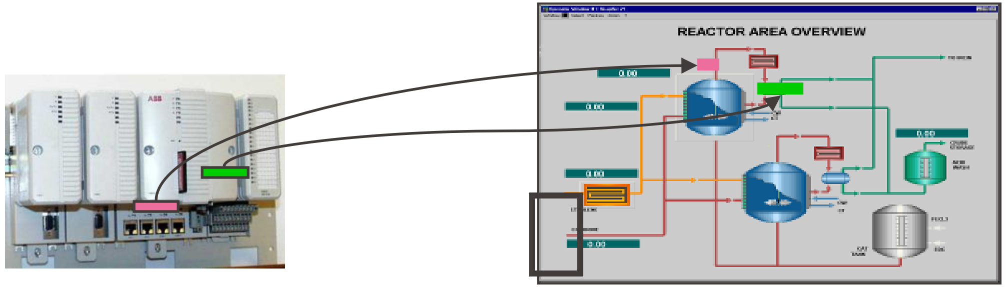
OPC UA Address Space



Structure is defined during engineering of the attached devices and sensor/actors.

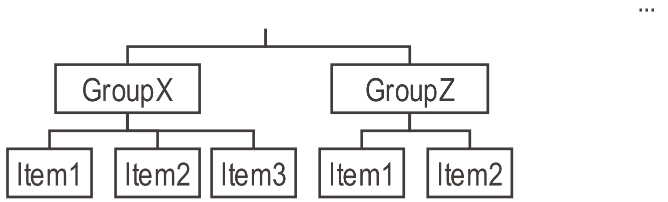
OPC UA Data Access

- Process variables describe the state of the industrial plant, generated by the sensors or calculated in the programmable logic controllers (PLCs).
- Process variables sent upon a change, on demand or when given time elapsed. OPC UA Data Access (DA) specification addresses collecting Process Variables.
- Main clients of OPC UA DA are visualization and (soft-) control.



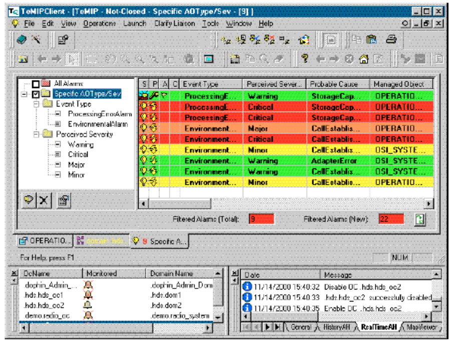
OPC UA Data Access: Subscriptions

- full-fledged PLC may export some 10’000 items
- client needs only a subset
- client subscribes for groups of items with similar time requirements
- client can be notified by the server on changes (event)



OPC Alarms & Conditions

- OPC Alarms & Conditions specifies alarms and conditions, their subscription, under which conditions they are filtered and sent with their associated messages.
  - Conditions represent the state of a system (e.g. temperature exceeds limit)
  - Alarms are abnormal states that require attention, such as "low oil pressure"
- The main clients are the Alarms and Event loggers:
  - determine the exact time of change (time stamping)
  - categorize by priorities
  - log for further use
  - acknowledge alarms


TeMIPClient - [TeMIP - Not-Closed - Specific AOType/Sev - [9]]

S	P	A	C	Event Type	Perceived Sever...	Probable Cause	Managed Object
				ProcessingE...	Warning	StorageCap...	OPERATIO...
				ProcessingE...	Critical	StorageCap...	OPERATIO...
				ProcessingE...	Critical	StorageCap...	OPERATIO...
				Environment...	Major	CallEstablis...	OPERATIO...
				Environment...	Critical	CallEstablis...	OPERATIO...
				Environment...	Minor	CallEstablis...	OSI_SYSTE...
				Environment...	Warning	AdapterError	OSI_SYSTE...
				Environment...	Warning	CallEstablis...	OSI_SYSTE...
				Environment...	Minor	CallEstablis...	OPERATIO...

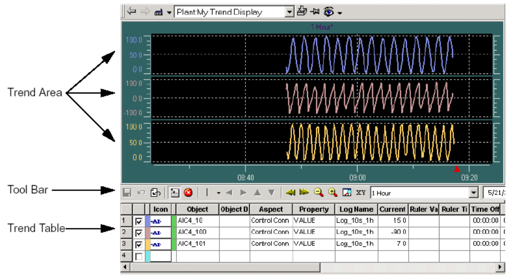
Filtered Alarms (Total): 9 Filtered Alarms (New): 22

For Help, press F1

OcName	Monitored	Domain Name
.dophin_Admin_...		.dophin_Admin_Dom
.hds.hds_oc1		.hds.dom1
.hds.hds_oc2		.hds.dom2
.demo.radio_oc		.demo.radio_system

Date	Message
11/14/2000 15:40:32	Disable OC .hds.hds_oc2
11/14/2000 15:40:33	.hds.hds_oc2 successfuly disabled
11/14/2000 15:40:35	Enable OC .hds.hds_oc2

OPC UA Historical Data Access: Purpose



		Icon	Object	Object D	Aspect	Property	Log Name	Current	Ruler Va	Ruler Ti	Time Off
1	✓	AI	AIC4_10		Control Conn	VALUE	Log_10s_1h	15 0			00:00:00
2	✓	AI	AIC4_100		Control Conn	VALUE	Log_10s_1h	-90 0			00:00:00
3	✓	AI	AIC4_101		Control Conn	VALUE	Log_10s_1h	7 0			00:00:00
4											

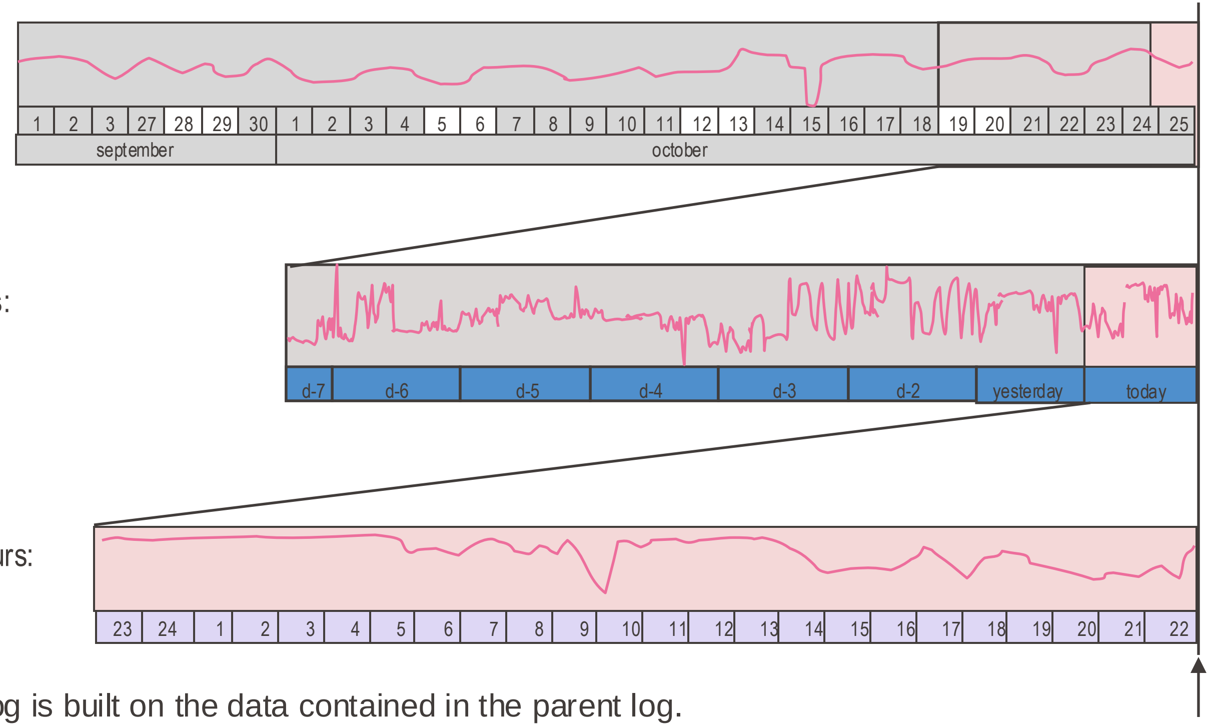
- Server: access to a historical database (logs) from process have been collected and time-stamped
- Clients: require ordered access to data, e.g., trend analysis, product tracking or data mining
- Services
  - browse historical database
  - retrieve data through proper filtering, e.g. by date range, by identity, by property
  - build aggregates over the retrieved data, such as average, minimum, maximum.
  - enter new entries, correct entries or remove entries
  - enter / delete annotations in the historical database

Hierarchical logs

1 hour forever:
≅ 5.2 MB / Year

1 minute for 7 days:
≅ 6 MB

1 second for 24 hours:
50 traces @ 12 B
≅ 52 MB



Actual data

A hierarchical log is built on the data contained in the parent log.
To reduce the log size, several aggregations can be applied:
record only maximum, minimum, average over a period, etc...

Assessment Overview

1. What is the objective of OPC?
2. What is an OPC Server, what is an OPC client?
3. What do the main OPC specifications describe?
4. On which technologies does OPC UA rely?
5. Can a change of an OPC variable be notified as an event, or shall the client poll?
6. What is the difference between an alarm and a condition?

Message Queuing Telemetry Transport (MQTT)

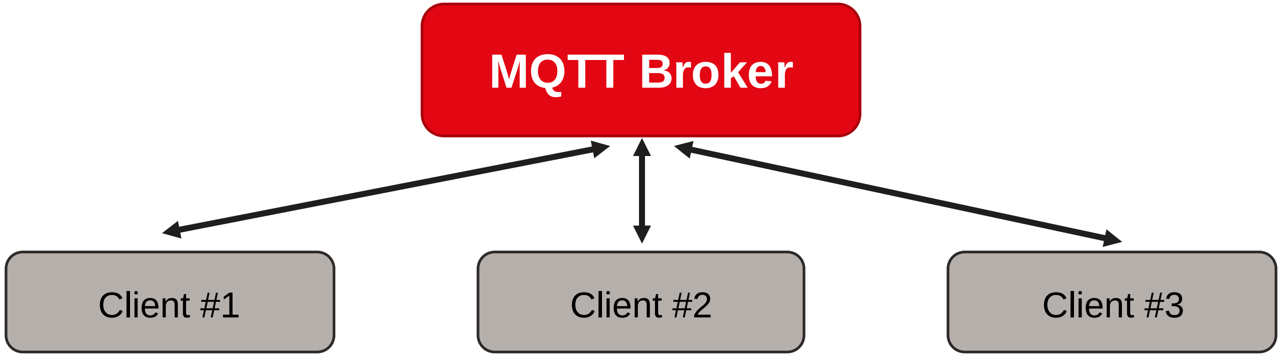
What is MQTT?

- MQTT is an OASIS standard **messaging protocol** for the Internet of Things (IoT).
- It is designed as an extremely lightweight **publish/subscribe** messaging transport that is ideal for connecting remote devices with a **small code footprint** and **minimal network bandwidth**.
- MQTT today is used in a wide variety of industries, such as automotive, manufacturing, telecommunications, oil and gas, etc.
- Developed in 1999 by Andy Stanford-Clark (IBM) and Arlen Nipper (Arcom).

[Source: http://mqtt.org]

MQTT System Architecture

- Application layer protocol running on top of TCP (port 1883)
- Binary wire protocol (fixed header & variable payload), optimized for small messages (few kilobytes), encryption using TLS
- Quality of Service (QoS) flags: best effort / at least once / exactly once
- Clients connect to a message broker



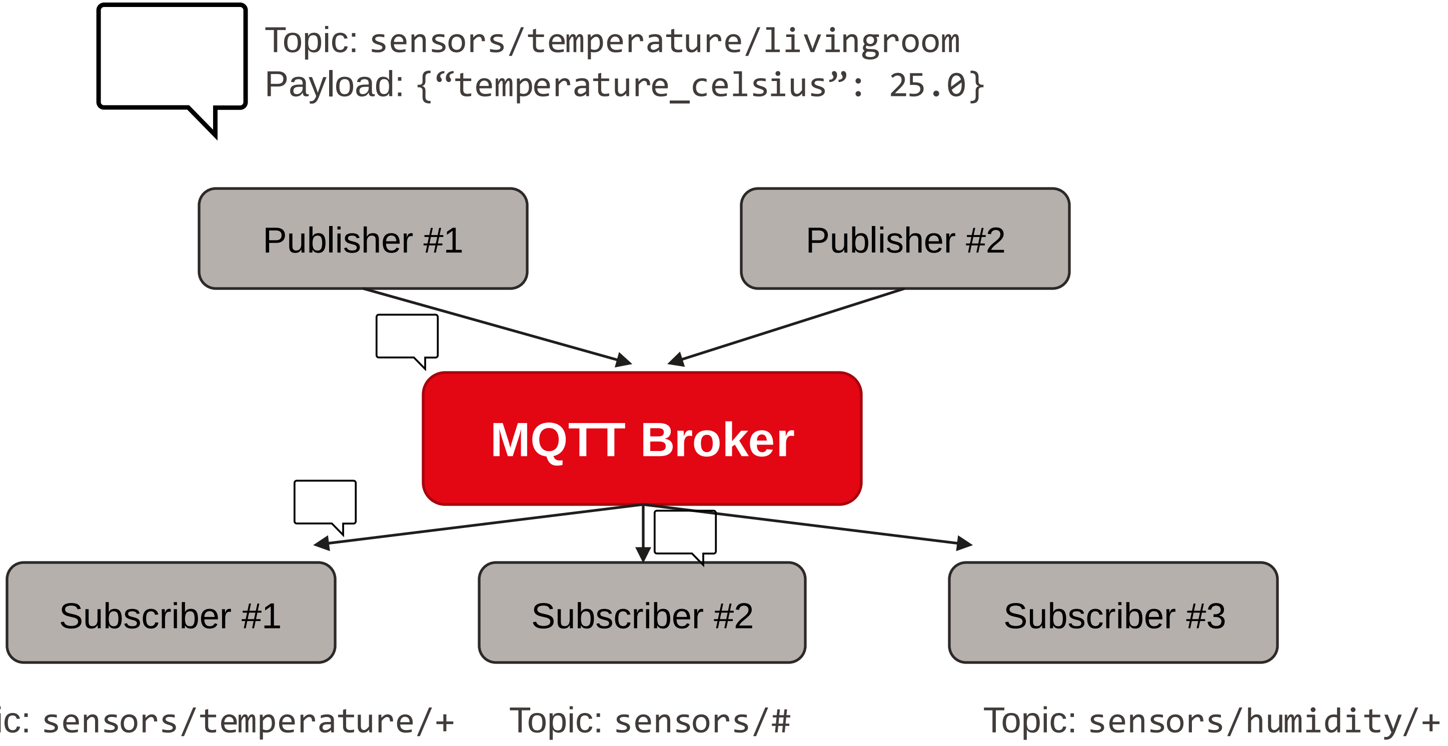
Publish & Subscribe with MQTT

- Topics with hierarchical structure managed by the broker, for example:

  ```
  sensors/temperature/livingroom
  sensors/temperature/bedroom
  sensors/humidity/office
  ```

- Clients publish messages to a specific topic
- Clients can subscribe to topic(s) of interest
  - Single topic, for example: `‘sensors/temperature/livingroom’`
  - Multiple topics using wildcard patterns:
    - Single-level wildcard ’+’:
      `‘sensors/temperature/+’ or ‘sensors/+/office’`
    - Multi-level wildcard ‘#’:
      `‘sensors/#’`

Publish/subscribe example with MQTT


Topic: sensors/temperature/livingroom
Payload: {“temperature_celsius”: 25.0}
Publisher #1
Publisher #2
MQTT Broker
Subscriber #1
Subscriber #2
Subscriber #3
Topic: sensors/temperature/+
Topic: sensors/#
Topic: sensors/humidity/+

Other Communication Protocols

- There are many other communication protocols used in automation systems:
  - CAN bus (mainly for automotive)
  - IO-Link
  - AMQP
  - BACNET (building automation)
  - KNX (building automation)
  - Wireless protocols (Matter/Thread, Wireless HART, NB-IoT, etc.)
  - And many more …