Chapter 1

Concepts and Use Cases

Prof. Dr.-Ing. Michael Scharf

Hochschule Esslingen – University of Applied Sciences

Concepts and Use Cases

Chapter Content

1. Internet of Things (IoT)
2. MQTT as IoT Protocol
3. Industrial Cyber-Physical Networks

Internet of Things (IoT)

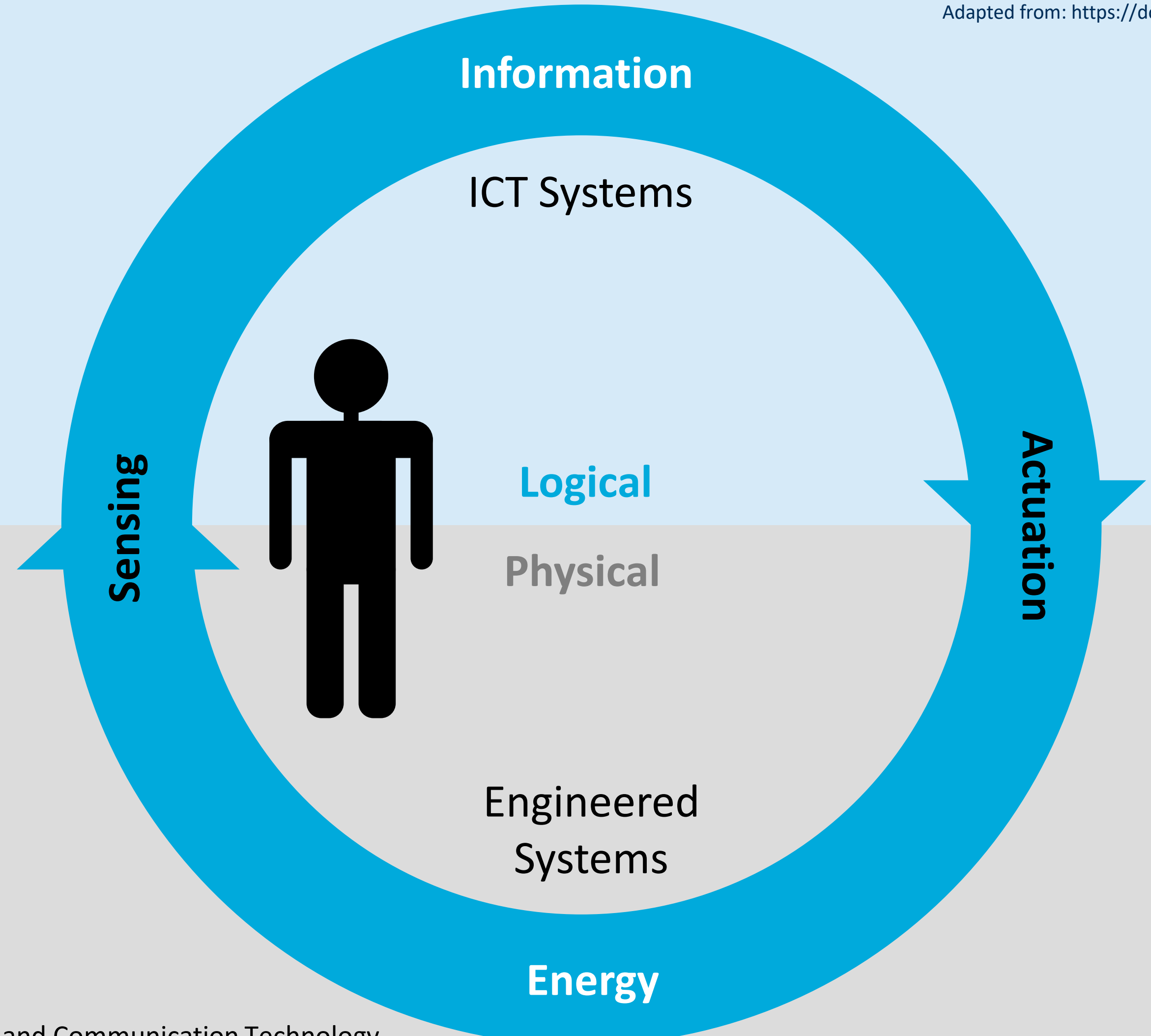
Prof. Dr.-Ing. Michael Scharf

Hochschule Esslingen – University of Applied Sciences

Scope and Use Cases

Scope and Use Cases
“Internet-of-Things” (IoT) Characteristics

Adapted from: https://doi.org/10.6028/NIST.SP.1900-202



Legend: ICT .. Information and Communication Technology

Scope and Use Cases
Typical Internet-of-Things (IoT) Use Cases

- Logistics (tracking by “RFID tags”)
- Building and home automation (“smart home”)
- Digital infrastructure (“smart city”)
- Medical and healthcare (telediagnosis, etc.)
- Agriculture and environmental monitoring
- …
- Increasing industrial usage (cf. later chapter)

Scope and Use Cases
Examples for Related Terms

- Ubiquitous Computing
- Massive Machine-to-Machine (M2M) Communication
- Internet of Everything
- Industrial Internet of Things (IIoT)
- Industrial Internet
- Industrial Cyber-Physical Systems (ICPS)
- Cyber–Physical Production Systems (CPPS)
- Industry 4.0 / „Industrie 4.0“

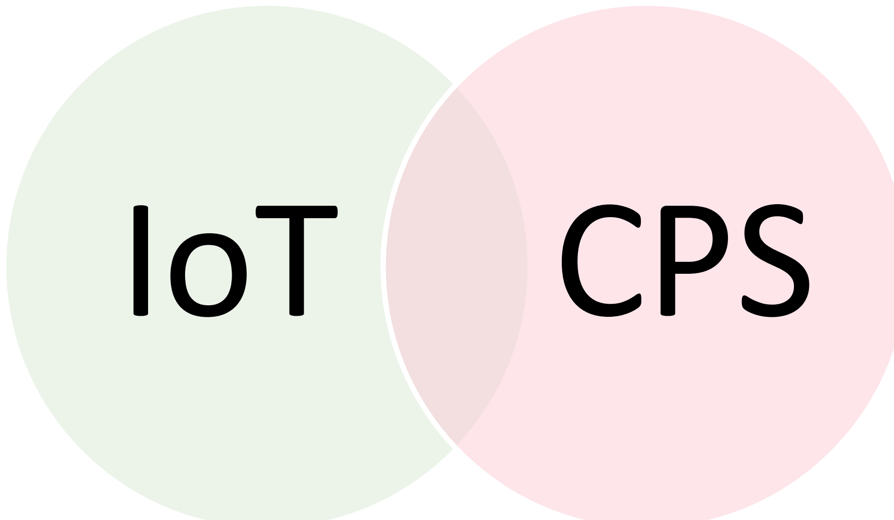
Industrial usage

Overlapping terminology with no clear separation and/or definition

Scope and Use Cases
"IoT" and "CPS" Terminology

Example **Many definitions and unclear scope of "IoT" and "CPS"**

Source: https://doi.org/10.6028/NIST.SP.1900-202



Example definition of "IoT":
The term "Internet of Things" (IoT) denotes a trend where a large number of embedded devices employ communication services offered by Internet protocols. Many of these devices, often called "smart objects", are not directly operated by humans but exist as components in buildings or vehicles, or are spread out in the environment. (Source: IETF/IAB)

Example definitions of "CPS":
Cyber-Physical Systems (CPS) comprise interacting digital, analog, physical, and human components engineered for function through integrated physics and logic. (Source: NIST)

Scope and Use Cases
When can a System be Labeled CPS, IoT, or both?

Example

Source: https://doi.org/10.6028/NIST.SP.1900-202

1. Does the system have one or more elements in each of the **component** categories: **logical**, **physical**, **transducing**, and **human**? (Note that the relevant capabilities of the human component vary with differing roles such as user, component, environmental factor, etc.)
2. Are these elements **integrated** to provide for **transmission, transformation, and storage** of **energy for physical elements** and **information for logical elements**; as well as **input, processing, and output functions** for **transducing elements**?
3. Does the system have one or more CPS/IoT functions where such a function is defined as involving the **linkage of logical and physical system states**?

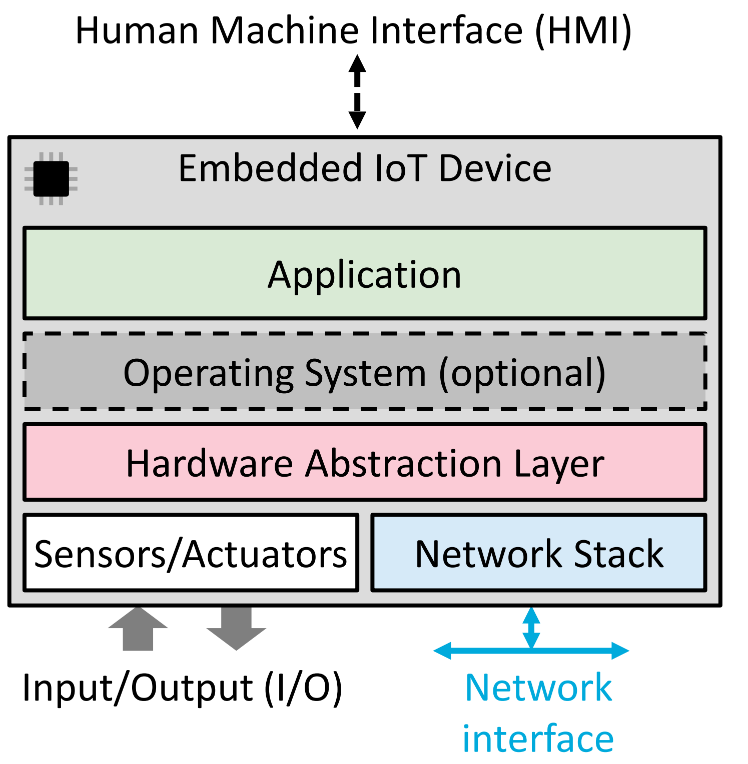
➔ If the **answers to all three are 'yes'** — in other words if the system has the components, capabilities, and functions of a CPS/IoT system — then it can be **appropriately labeled 'CPS,' 'IoT,' and both**.

Devices

- "IoT" term originates from **Radio-Frequency Identification** (**RFID**) devices
- Today includes all kinds of embedded devices

> **Embedded device**: A computer system that has a dedicated function within a larger mechanical or electronic system

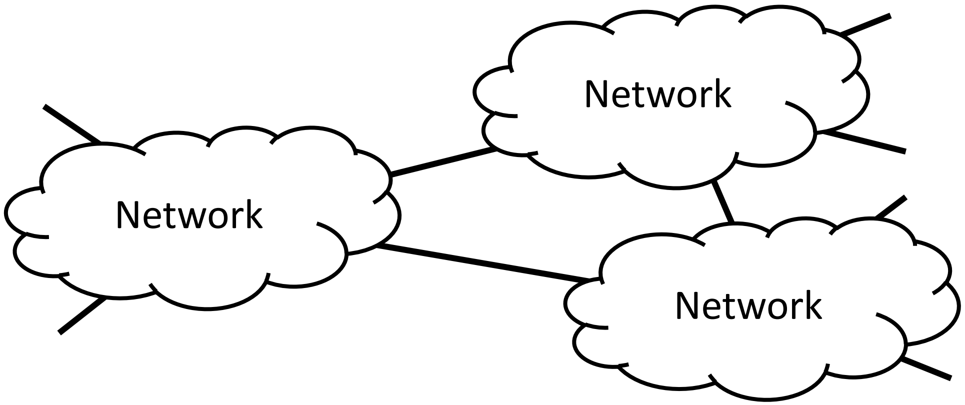
- Large variety of hardware
  - **CPU**: Constrained **microcontroller** ("**µC**") or full **microprocessor** ("**µP**")
  - **RAM**: From few KiB to many GiB
  - Software: Without or with **operating system** (OS)
  - **I/O: Typically sensors** and/or **actuators** towards the real-world (e.g., process interface)
  - **Network**: Large variety including both wireline (e.g. Ethernet, special fieldbusses) and wireless technologies (e.g., WLAN, Bluetooth, LoRaWAN)
  - Possibly also a **Human-Machine Interface** (HMI)
- Prototyping often by single-board computers



Devices
Classification of Devices

Characteristics	Microcontroller (µC)	Microprocessor (µP, CPU)
Architecture	8, 16, or 32 bit	64 bit (32 bit compatible)
Instruction set	Small	Large and complex
Cores	Often 1, multiple possible	1 – 100
Clock frequency	1 MHz – more than 1 GHz	500 MHz – more than 5 GHz
Power consumption	Some µW – 5 W	2 – 500 W
Addressable memory	Few KiB – multiple MiB	Multiple GiB – multiple TiB
Cache	Seldom	Typical, up to many MiB
Typical operating system (OS)	None, or real-time OS	Windows, Linux, macOS, …
Optimized for real-time interrupts	Yes	No
Number of transistors	Thousands – Millions	Billions
Realized as Systems-on-Chip	Frequently	Seldom
Number of units per year	More than 20 billion	2 – 2.5 billion
Cost	10 Cent – 20 EUR	30 EUR – 30,000 EUR
Typical example	Arduino Uno Atmel Atmega 1x 8 bit	Raspberry Pi 4 Broadcom 4x ARM Cortex-A72

Internet Fundamentals



- A **network of networks**
- **Complex, layered, distributed** system
- Packet network
  - Connectionless delivery of **Internet Protocol (IP) packets**
  - Forwarding based on destination **IP address**
  - **Best effort service** typically without Quality-of-Service (QoS) mechanisms

Internet Fundamentals
Protocols

Protocol: Set of rules governing the communication between endpoints
- Syntax (what can be communicated)
- Semantics (how it can be communicated)

- Most protocols specify one or more of …
  - Message format and delimiters
  - Addresses, identifiers and/or naming
  - Control (setup, handshaking, negotiation, termination)
  - Handling of errors and other events
- Communication pattern
  - Unicast vs. multicast vs. anycast vs. broadcast
  - Reliable vs. unreliable
  - Connection-less (packet-switching) vs. connection-oriented (circuit-switching)
- Transport of control data
  - In-band signaling, e.g., HyperText Transfer Protocol (HTTP)
  - Out-of-band signaling, e.g., File Transfer Protocol (FTP) or Session Initiation Protocol (SIP)
- Maintaining of state, stored in nodes
  - Hard state: State is explicitly installed and removed by messages
  - Soft state: State is installed by a message and expires after a timeout unless refreshed

Internet Fundamentals
Hybrid Reference Model

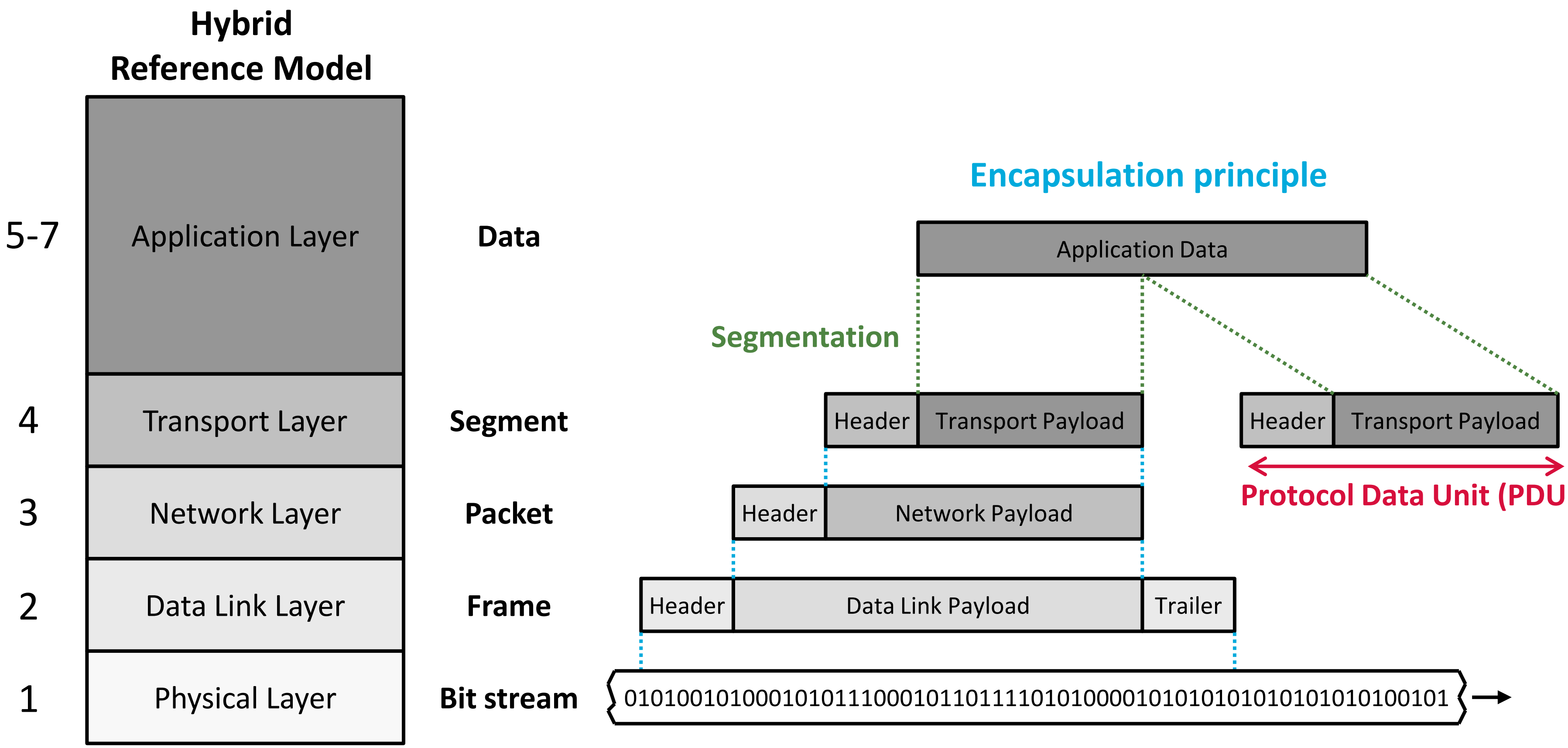
	Hybrid Reference Model	Examples for technologies and protocols frequently used in the Internet
5-7	Application Layer	• Hypertext Transfer Protocol (**HTTP**) • Secure Shell (**SSH**) • Domain Name System (**DNS**) • …
4	Transport Layer	• Transmission Control Protocol (**TCP**) • User Datagram Protocol (**UDP**)
3	Network Layer	• Internet Protocol Version 4 (**IPv4**) • Internet Protocol Version 6 (**IPv6**)
2	Data Link Layer	• **Ethernet** • Wireless LAN (**WLAN**)
1	Physical Layer	• **Cables** such as Twisted Pair (TP) or fiber • **Radio Frequency (RF)** communication

Code < >

Bits 01101001

Hybrid reference model: Abstract model for communication consisting of five layers with different functions, as a compromise between the Internet and the OSI reference model.

Internet Fundamentals
Encapsulation Principle

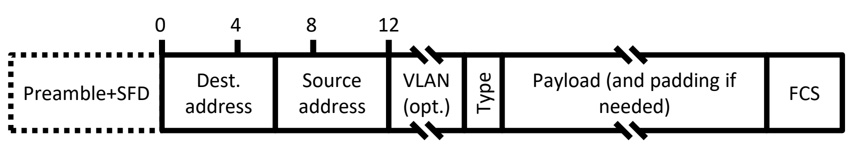


Encapsulation principle: A Protocol Data Unit (PDU) of an upper-layer protocol is transported as **payload** of the lower-layer protocol and control information in a **header** and/or **trailer** is added.

Internet Fundamentals
Ethernet in a Nutshell

- **Connection-less transport of frames**
- Functions of physical layer (**PHY layer**)
  - Physical transmission rate from 10 Mbit/s to more than 100 Gbit/s
  - Different physical media (e.g., twisted pair cable, fiber) over short-range and medium-range distances
- Functions of data link layer (**MAC layer**)
  - Transport of payload limited by the **Maximum Transmission Unit (MTU)** of 1500 byte
  - Switching of frames and address learning
  - Loop prevention e.g. by **Spanning Tree Protocol (STP)**
  - Historical media access by Carrier Sense Multiple Access / Collision Detection (CSMA/CD)
- Addressing
  - 48 bit **Media Access Control (MAC) address** with typically globally unique addresses
  - Address lookup in IPv4 networks by **Address Resolution Protocol (ARP)**
- Network elements
  - **Switch** (also called “bridge” if implemented in software, as well as in some standards)
  - **Media converter** (historically also hubs and repeaters)
- Extensions such as **Virtual LAN (VLAN)**

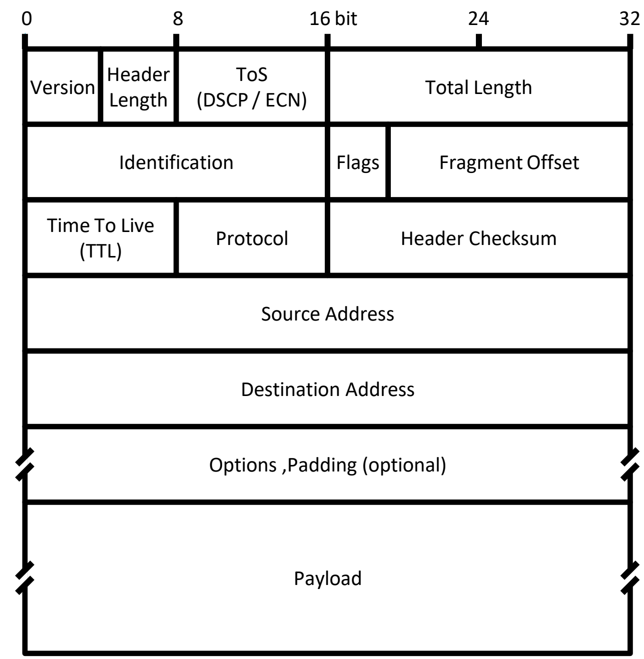
Ethernet II Frame:



Legend:
SFD … Start Frame Delimiter
FCS … Frame Check Sequence

Internet Fundamentals
Internet Protocol (IP) in a Nutshell

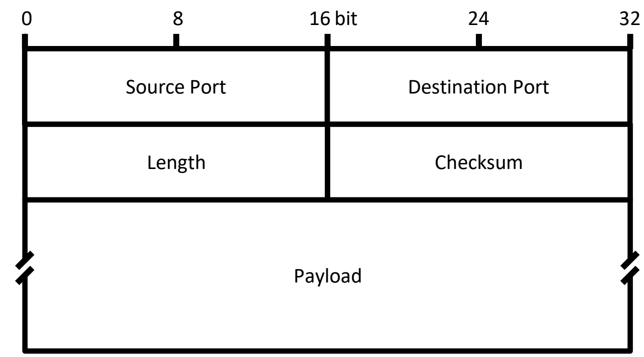
- **Connectionless, best effort transport of packets**
- Functions
  - Next-hop forwarding based on destination address
  - Fragmentation and reassembly
- Addressing
  - 32 bit address for **IP Version 4 (IPv4)**
  - 128 bit address for **IP Version 6 (IPv6)**
  - Globally structured address space
- Network elements
  - Router
  - Firewalls, NAT Gateways, …



IPv4 Packet

Internet Fundamentals
User Datagram Protocol (UDP) in a Nutshell

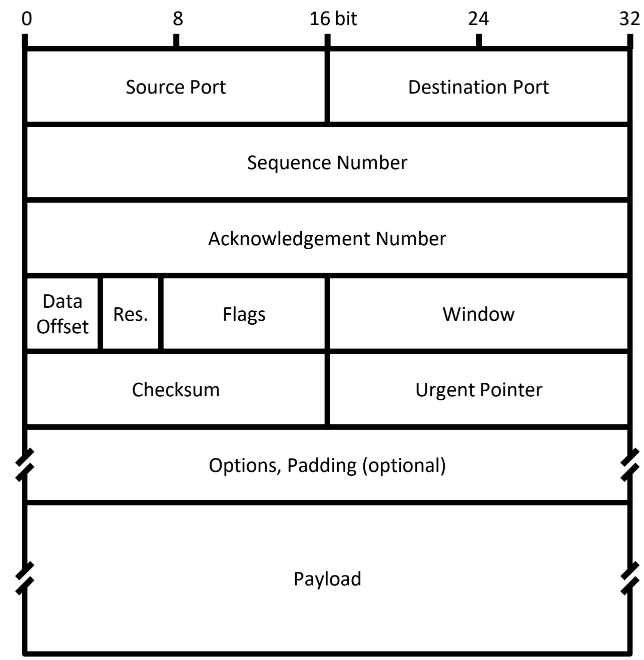
- **Connectionless unreliable datagram transport**
- Functions
  - Port multiplexing/demultiplexing
  - Error detection by checksum (optional)
  - No error recovery, no flow control, no congestion control
- Typical usage
  - Simple transactional interactions
  - Multicast



UDP Datagram

Internet Fundamentals
Transmission Control Protocol (TCP) in a Nutshell

- **Point-to-point, connection-oriented, reliable, in-order duplex byte-stream transport**
- Functions
  - Port multiplexing/demultiplexing
  - Connection management
  - Segmentation and reassembly
  - Reliable transport with error detection and retransmission-based recovery
  - Flow control and congestion control
- Typical usage
  - Default transport for most Internet applications
  - Often Transport Level Security (TLS) on top of TCP for confidentiality and integrity



TCP Segment

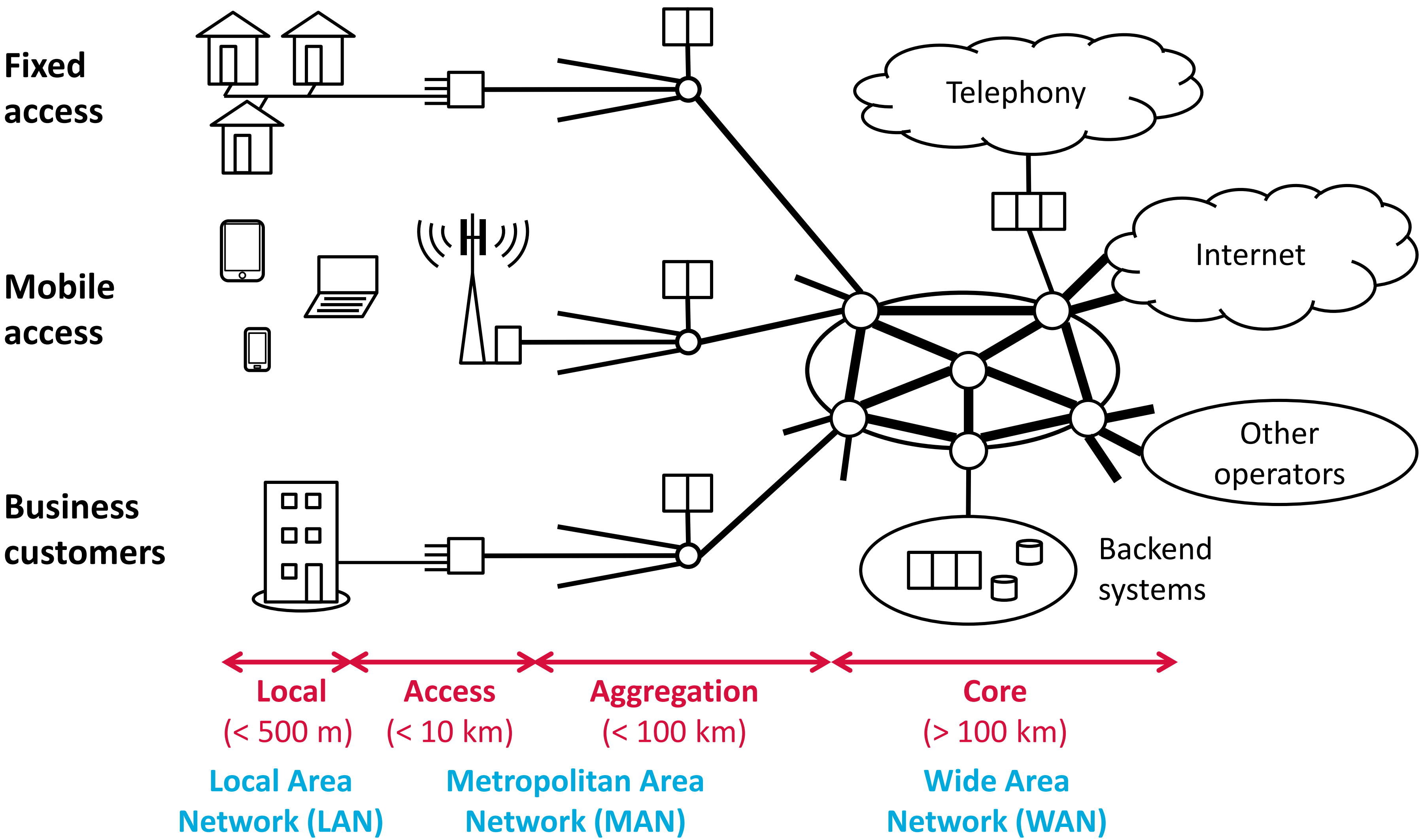
Internet Fundamentals
Examples of Standardization Organizations (SDOs)

- Third Generation Partnership Project (3GPP): Cellular 3G, 4G, 5G, … networks
- American National Standards Institute (ANSI): ASCII, Language C, …
- Alliance for Telecommunications Industry Solutions (ATIS): Telecommunication services
- Broadband Forum (BBF): Digital Subscriber Line (DSL), …
- European Telecommunications Standards Institute (ETSI): GSM, DECT, …
- International Electrotechnical Commission (IEC): Fieldbuses, …
- **Institute of Electrical and Electronics Engineers (IEEE)**: Ethernet, WLAN, …
- **Internet Engineering Task Force (IETF)**: TCP/IP protocol family, …
- International Standards Organization (ISO): ISO/OSI reference model, …
- International Telecommunications Union (ITU-T): Audio and video codecs (MPEG), security (X.509), optical transport networks, …
- MEF: Metro Ethernet, …
- Optical Interworking Forum (OIF): Optical transport networks, …
- Organization for the Advancement of Structured Information Standards (OASIS): MQTT, …
- TeleManagement Forum (TMF): Network management
- World Wide Web Consortium (W3C): HTML, XML, CSS, SVG, PNG, …
- …

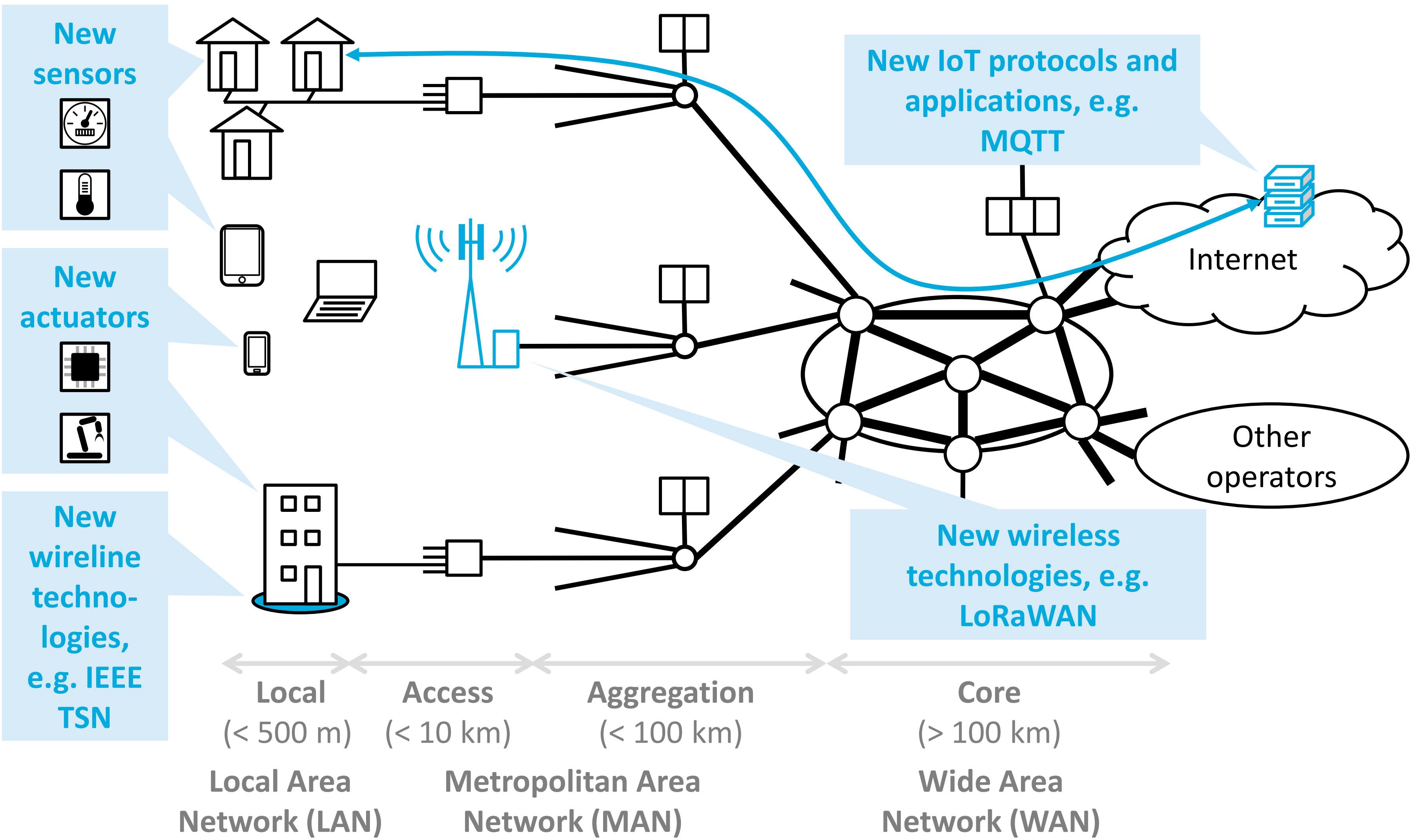
Internet Fundamentals
Examples for Network Technologies

- **Fixed access**
  - Digital Subscriber Line (DSL)
  - Cable networks
  - Fiber-to-the-Home (FTTH)
- **Mobile access**
  - 2G, 3G, 4G or 5G cellular networks
  - Wireless LAN (WLAN) hotspots
- **Core networks** or backbone
  - Optical transport networks
  - Multiprotocol Label Switching (MPLS)/Internet protocol (IP) core
- **Network services**
  - Leased Lines and Virtual Private Networks (VPNs) for business customers
  - Telephony
  - Internet Protocol Television (IPTV)
  - … and much more

Internet Fundamentals
Networks of an Internet Service Provider (ISP)



Internet Fundamentals
Evolution to IoT



Example for an IoT Protocol Stack

Exercise

The application layer protocol **Constrained Application Protocol (CoAP)** is designed for machine-to-machine (M2M) applications such as smart energy and building automation. CoAP messages are encapsulated in UDP datagrams.

A CoAP client runs on a small embedded device (e.g., a Raspberry Pi computer) with an Ethernet port that connects via IPv6 to a router. Sketch the protocol stack in the embedded device that is used for communication with CoAP.

Resulting protocol stack with layers:

CoAP
UDP
IPv6
Ethernet (MAC)
Ethernet (PHY)

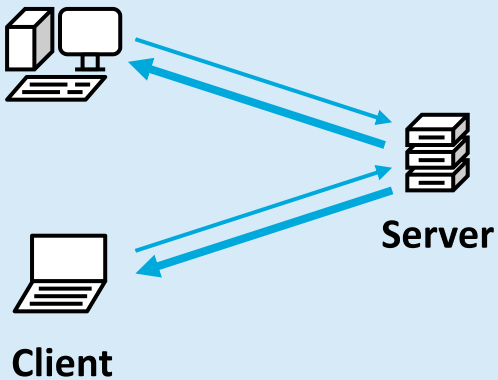
MQTT as IoT Protocol

Prof. Dr.-Ing. Michael Scharf

Hochschule Esslingen – University of Applied Sciences

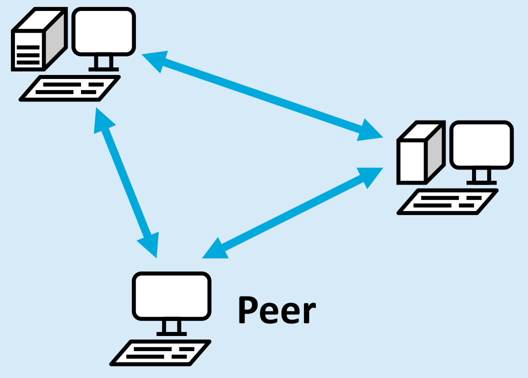
Communication Patterns

Client-Server



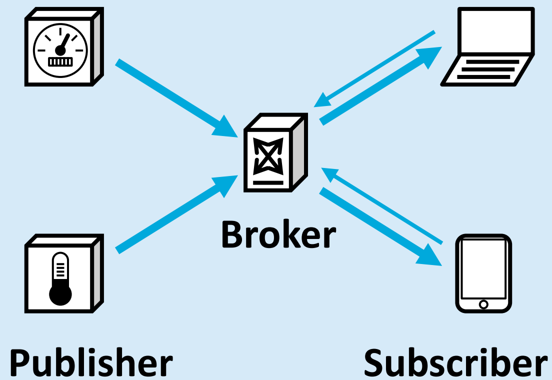
- Components
  - Client
  - Server
- Examples
  - Web (HTTP)
  - Databases

Peer-to-Peer



- Components
  - Peers
  - No hierarchy
- Examples
  - File sharing
  - Infrastructureless

Publish-Subscribe

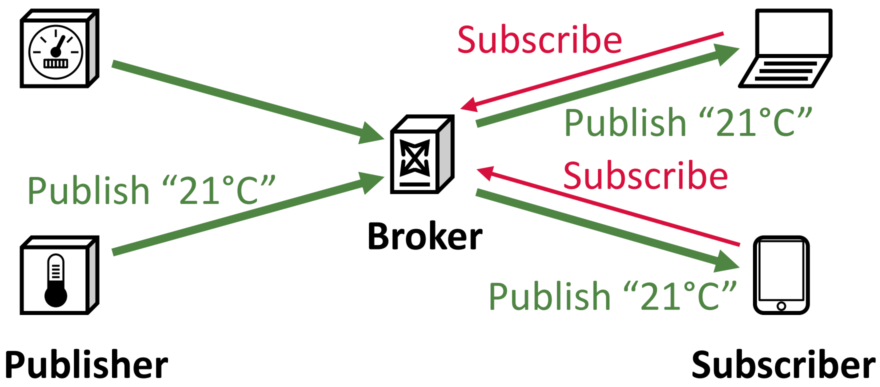


- Components
  - Broker as proxy
  - Separate roles
- Examples
  - IoT (MQTT)
  - Content sharing

MQTT Communication

- **Message Queuing Telemetry Transport (MQTT)**
  - Lightweight, event and message-oriented protocol for efficient asynchronous communication in constraint environments
  - Publish-subscribe architecture on top of TCP/IP
- Current **de-facto standard** in IoT
  - Originally developed in year 1999 inside IBM for supervision of oil pipelines
  - Standardized by OASIS since 2013, also ISO/IEC 20922 standard
  - Software support by Eclipse Foundation
- Important protocol versions
  - **Version 3.1.1**
    - Year 2014
    - URI: https://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html
  - **Version 5.0** – not backward compatible
    - Year 2019
    - URI: http://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html

MQTT Communication
Publish-Subscribe Pattern



- Publish-subscribe pattern decouples communication
  - **Subscribers** do not have to know **publishers**
  - **Asynchronous communication**, e.g., publishers can be turned off
- **Client**
  - Clients can be publisher, subscriber, or both
  - Bi-directional communication between each client and a broker
- **Broker**
  - Server that forwards content organized in topics between clients
  - Brokers can serve many clients (i.e., many publishers and/or subscribers)

MQTT Communication
Topics

- Topics to define communication channels
  - Publishers send messages for at least one topic
  - Subscribers receive these messages if subscribed to this topic
- Topics defined by a string
  - UTF-8 encoding of characters
  - Example for topic:

Topic level   Topic separator

myhome/groundfloor/kitchen/temperature

- Hierarchical structure of **topic levels**
  - Levels separated by slash (“**/**”) as **topic separator**
  - **Wildcards** for entire topic levels
    - Single-level wildcard “**+**” for one level: Allowed one or multiple times
    - Multi-level wildcard “**#**” for all subsequent levels: Allowed only at the end
  - No partial use of a wildcard inside topic level

MQTT Communication
Use of Wildcards

Example

Example topic: **myhome/groundfloor/kitchen/temperature**

String	**Matching**	**Not matching**
Single-level	myhome/groundfloor/kitchen/temperature	myhome/groundfloor/livingroom/temperature
Single-level with wildcard	myhome/groundfloor/+/temperature	myhome/groundfloor/kitchen
	+/groundfloor/kitchen/temperature	groundfloor/+/kitchen/temperature
	myhome/groundfloor/kitchen/+	myhome/groundfloor/k+/temperature *)
Single-level wildcard with recursion	myhome/+/+/temperature	groundfloor/+/+/temperature
	myhome/+/kitchen/+	myhome/+/groundfloor/+
	+/+/+/+	+/+/+
Multi-level	myhome/groundfloor/#	myhome/firstfloor/#
	+/groundfloor/#	myhome/#/temperature *)
	#	groundfloor/#

Legend: *) invalid MQTT syntax

MQTT Communication
System Topics

- System topics "**$SYS**" for internals of the MQTT broker
- Subscription to "#" does not match "$SYS"
- Subscription to "$SYS/#" required

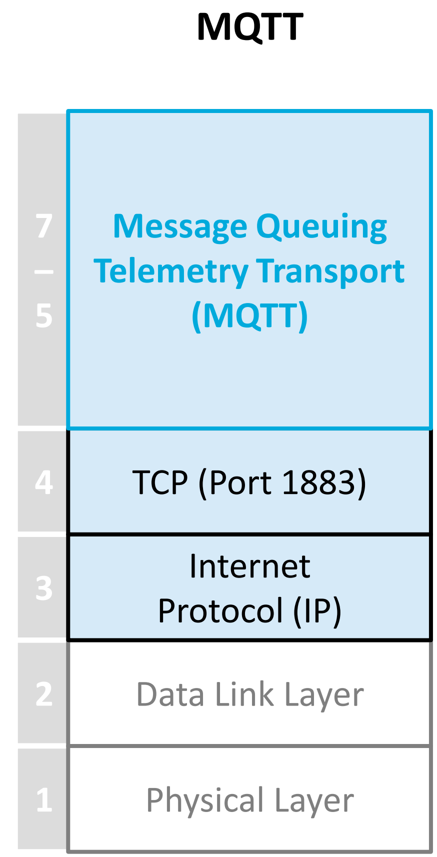
Example **Example $SYS topics in Mosquitto broker**

Source: https://github.com/mqtt/mqtt.org/wiki/SYS-Topics

Topic	Description
$SYS/broker/clients/connected	The number of currently connected clients.
$SYS/broker/messages/received	The total number of messages of any type received since the broker started.
$SYS/broker/messages/sent	The total number of messages of any type sent since the broker started.
$SYS/broker/messages/publish/received	The total number of PUBLISH messages received since the broker started.
$SYS/broker/messages/publish/sent	The total number of PUBLISH messages sent since the broker started.
$SYS/broker/subscriptions/count	The total number of subscriptions active on the broker.
$SYS/broker/time	The current time on the server.
$SYS/broker/uptime	The amount of time in seconds the broker has been online.
$SYS/broker/version	The version of the broker. Static.
…	…

MQTT Messages

- MQTT can use **any reliable transport**
- Default operation: Use of TCP
  - Unencrypted (plaintext) on **port 1883**
  - Secured by TLS on port 8883
- Some brokers also support transport over WebSockets
- **MQTT messages** with a simple format
  - Fixed header [2 – 5 byte]
  - Variable header [optional]
  - Payload [optional]



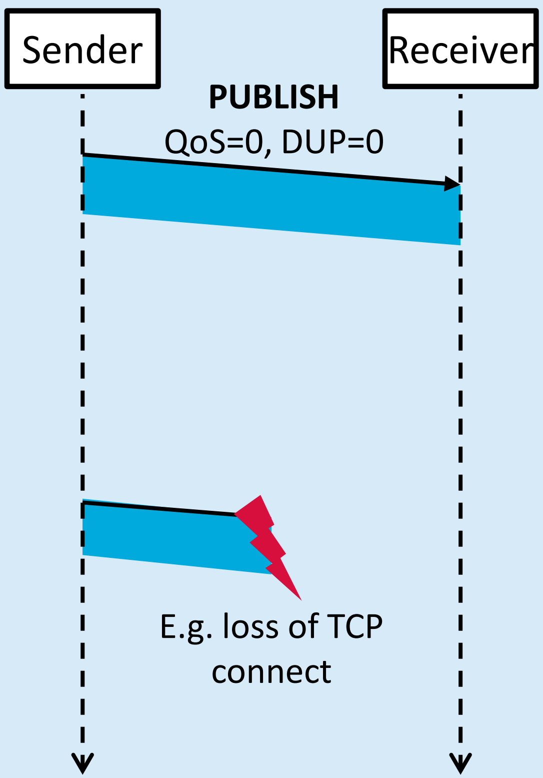
MQTT Messages
Message Types

- 14 message types
  - **Login/logout**: CONNECT, CONNACK, DISCONNECT
  - **Publication**: PUBLISH, PUBACK, PUBREC, PUBREL, PUBCOMP
  - **Subscription**: SUBSCRIBE, SUBACK, UNSUBSCRIBE, UNSUBACK
  - **Monitoring**: PINGREQ, PINGRESP
- Different **Quality-of-Service** (QoS) levels
  - **QoS Level 0**: At most once (“fire and forget”)
  - **QoS Level 1**: At least once
  - **QoS Level 2**: Guaranteed once

Warning: MQTT uses acronym “QoS” in a very specific way!

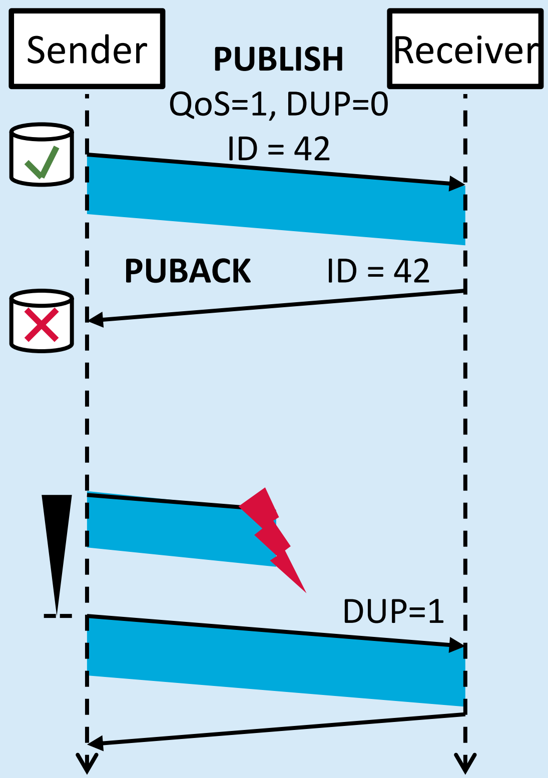
MQTT Messages
QoS Levels

QoS Level 0



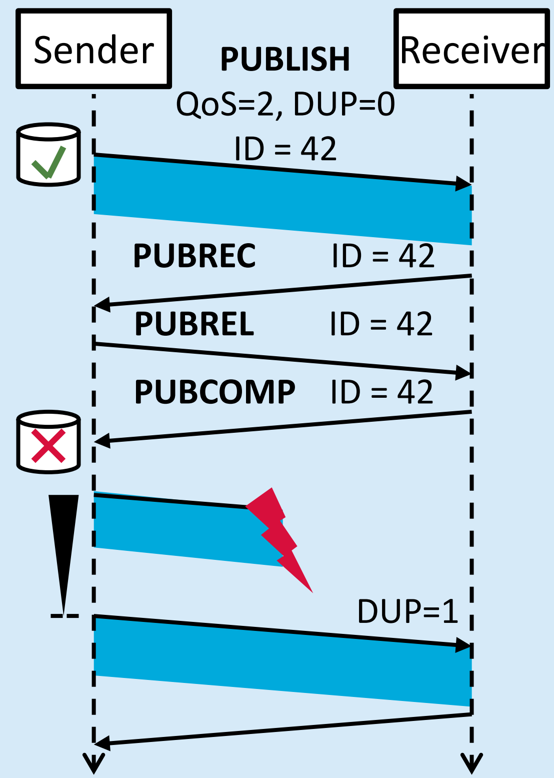
- At most once (“<=1”)
- Not reliable

QoS Level 1



- At least once (“>=1”)
- Duplication possible

QoS Level 2

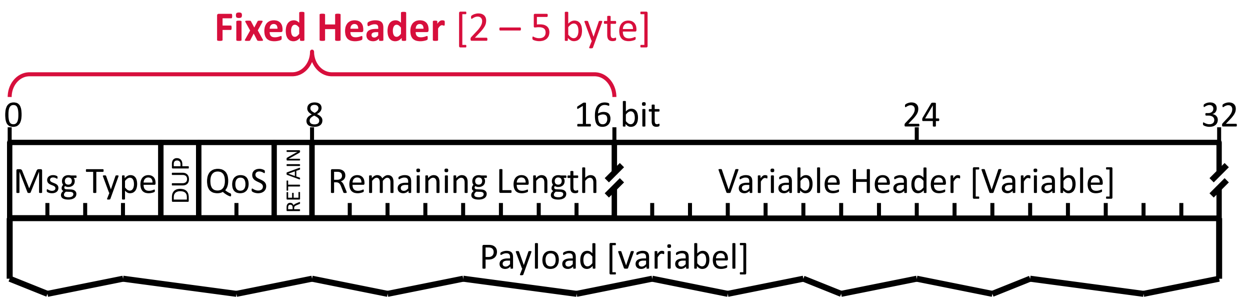


- Exactly once (“==1”)
- Reliable

MQTT Messages
List of all Messages

Name	Msg Type	Direction	Description
Reserved	0000 (0x00)	Forbidden	Reserved
CONNECT	0001 (0x01)	Client to server	Client request to connect to server
CONNACK	0010 (0x02)	Server to client	Connect acknowledgment
PUBLISH	0011(0x03)	Both	Publish message
PUBACK	0100 (0x04)	Both	Publish acknowledgement
PUBREC	0101 (0x05)	Both	Publish received (step 1)
PUBREL	0110 (0x06)	Both	Publish released (step 2)
PUBCOMP	0111 (0x07)	Both	Publish complete (step 3)
SUBSCRIBE	1000 (0x08)	Client to server	Subscribe request
SUBACK	1001 (0x09)	Server to client	Subscribe acknowledgement
UNSUBSCRIBE	1010 (0x0A)	Client to server	Unsubscribe request
UNSUBACK	1011 (0x0B)	Server to client	Unsubscribe acknowledgement
PINGREQ	1100 (0x0C)	Client to server	PING request
PINGRESP	1101 (0x0D)	Server to client	PING response
DISCONNECT	1110 (0x0E)	Client to server	Client disconnecting
Reserved	1111 (0x0F)	Forbidden	Reserved

MQTT Messages
Fixed Header



MQTT Message Format

Byte order:
Msg Type: Bits 4-7
Flags: Bits 0-3

- Fixed header with 2 – 5 byte length in all messages
- Structure of first byte
  - Msg Type [4 bit]
  - Flags [4bit]
    - Duplication (DUP)
    - QoS
    - RETAIN
- Further structure defined by field Remaining Length

MQTT Messages
Flags Field in Fixed Header

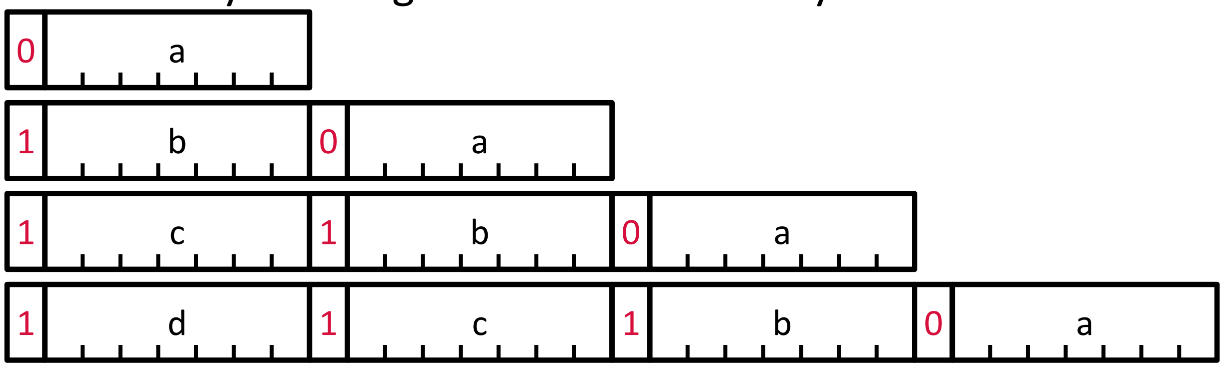
- Flags are only set in PUBLISH messages
- **DUP** [1 bit]
  - Default is value 0
  - If set to 1, the message is a retransmission (in QoS 1 or 2)
- **QoS** [2 bit]

QoS level	Bit 2	Bit 1	Description	Meaning
0	0	0	At most once “<=1”	Fire and forget
1	0	1	At least once “>=1”	Acknowledged delivery
2	1	0	Exactly once “==1”	Assured delivery
3	1	1	Reserved	

- **RETAIN** [1 bit]
  - If retained flag is set to true, broker stores the last retained message (and QoS) for that topic
  - Each client that subscribes to a matching topic receives retained message immediately after subscription
  - Broker stores only one retained message per topic

MQTT Messages
Remaining Length Field in Fixed Header

- **Remaining Length** [1 – 4 byte]
  - Encodes the remaining number of bytes
  - 1 to 4 bytes with 7 bit digits
    - Min length: 0 byte (message only with fixed header)
    - Max. length: 268,435,455 byte (ca. 256 MiB)
- Field format defined by most significant bit in each byte
  - 1 digit:
  - 2 digits:
  - 3 digits:
  - 4 digits:



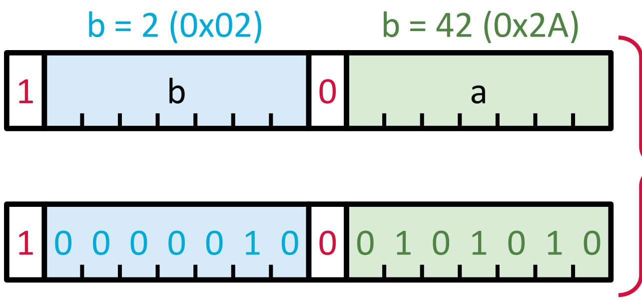
- Encoding of numerical value of the Remaining Length with 7 bit digits

Digits	From	To	Value
1	0 (0x00)	127 (0x7F)	a
2	128 (0x8001)	16,383 (0xFF7F)	$b \cdot 128 + a$
3	16,384 (0x808001)	2,097,151 (0xFFFF7F)	$c \cdot 128^2 + b \cdot 128 + a$
4	2,097,152 (0x80808001)	268,435,455 (0xFFFFFF7F)	$d \cdot 128^3 + c \cdot 128^2 + b \cdot 128 + a$

MQTT Messages
Example for Remaining Length Encoding

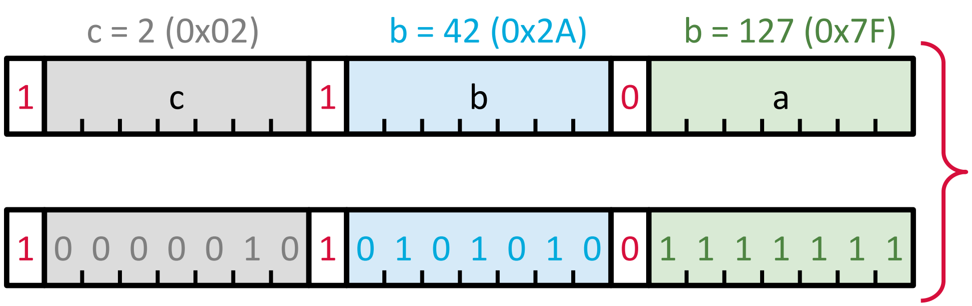
Example

Example 1:



Value = $2 \cdot 128 + 42 = 298$

Example 2:



Value = $2 \cdot 128^2 + 42 \cdot 128 + 127 = 38271$

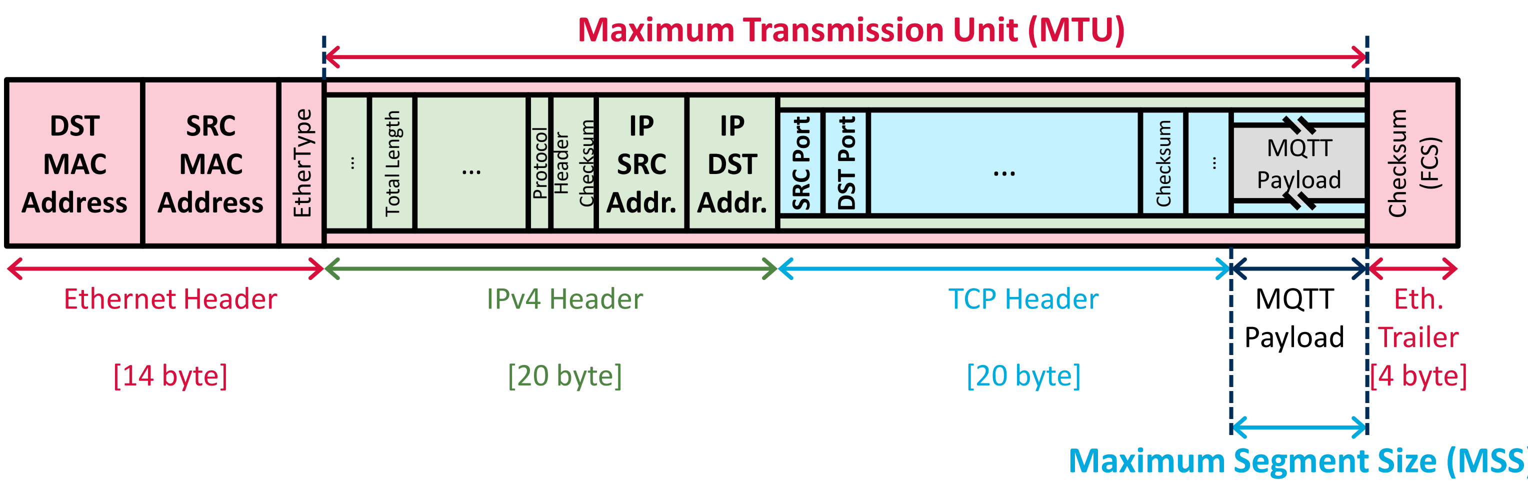
MQTT Messages
Message Content Format

- **Variable header** [optional]
  - Multiple variable headers allowed
  - Examples
    - Topic [2 byte + content]
    - Packet identifier in QoS levels 1 and 2
- **Payload** [optional]
  - Actual content
  - Length results from total length in fixed header minus length of variable header

MQTT Messages
Transport

- MQTT messages between 2 byte and ca. 268 MB (≈ 256 MiB)
- Transport as byte stream over **TCP connection**
  - TCP offers reliable transport, flow control and congestion control
  - Maximum Segment Size (MSS) of 1460 byte (IPv4) or 1440 byte (IPv6) avoids IP packet fragmentation for Ethernet MTU of 1500 byte
- MQTT only has to deal with TCP connection failures

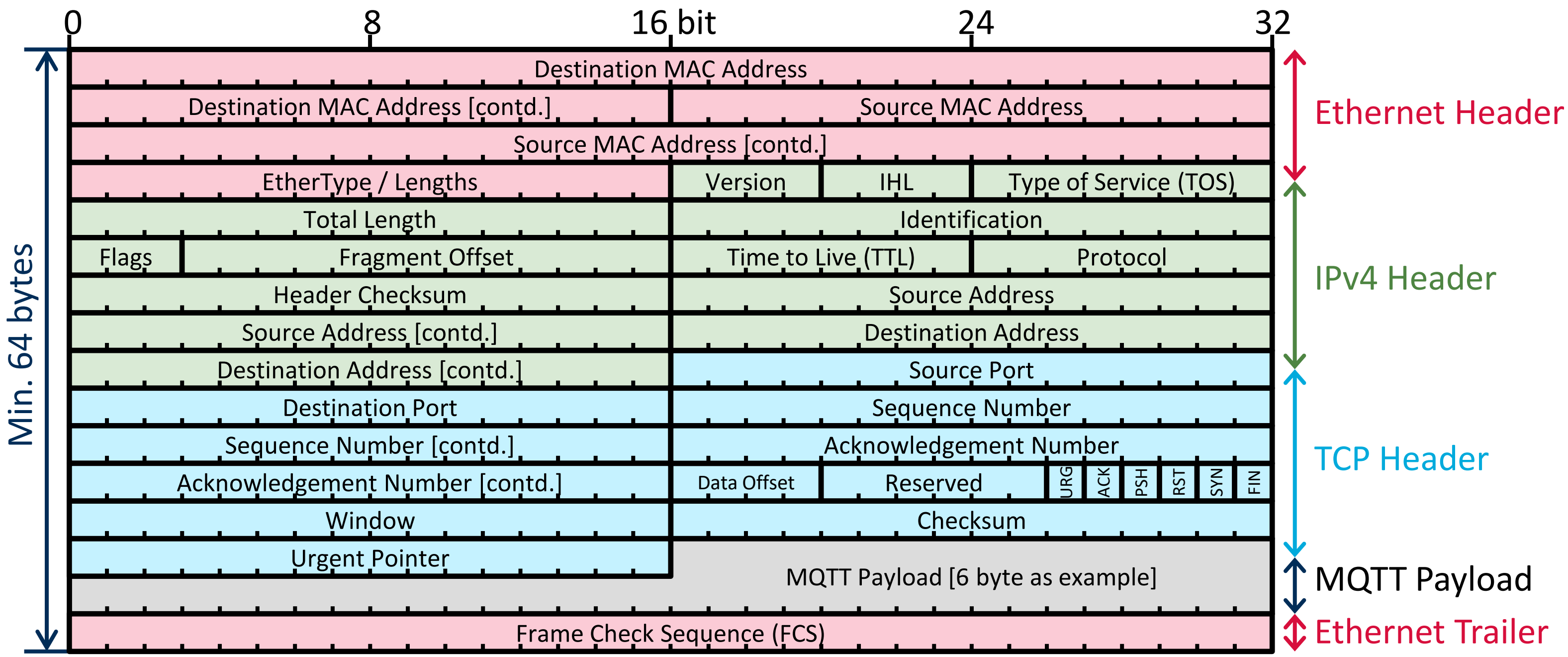
Example **Ethernet II frame**



MQTT Messages
Encapsulation in Ethernet, IPv4, and TCP

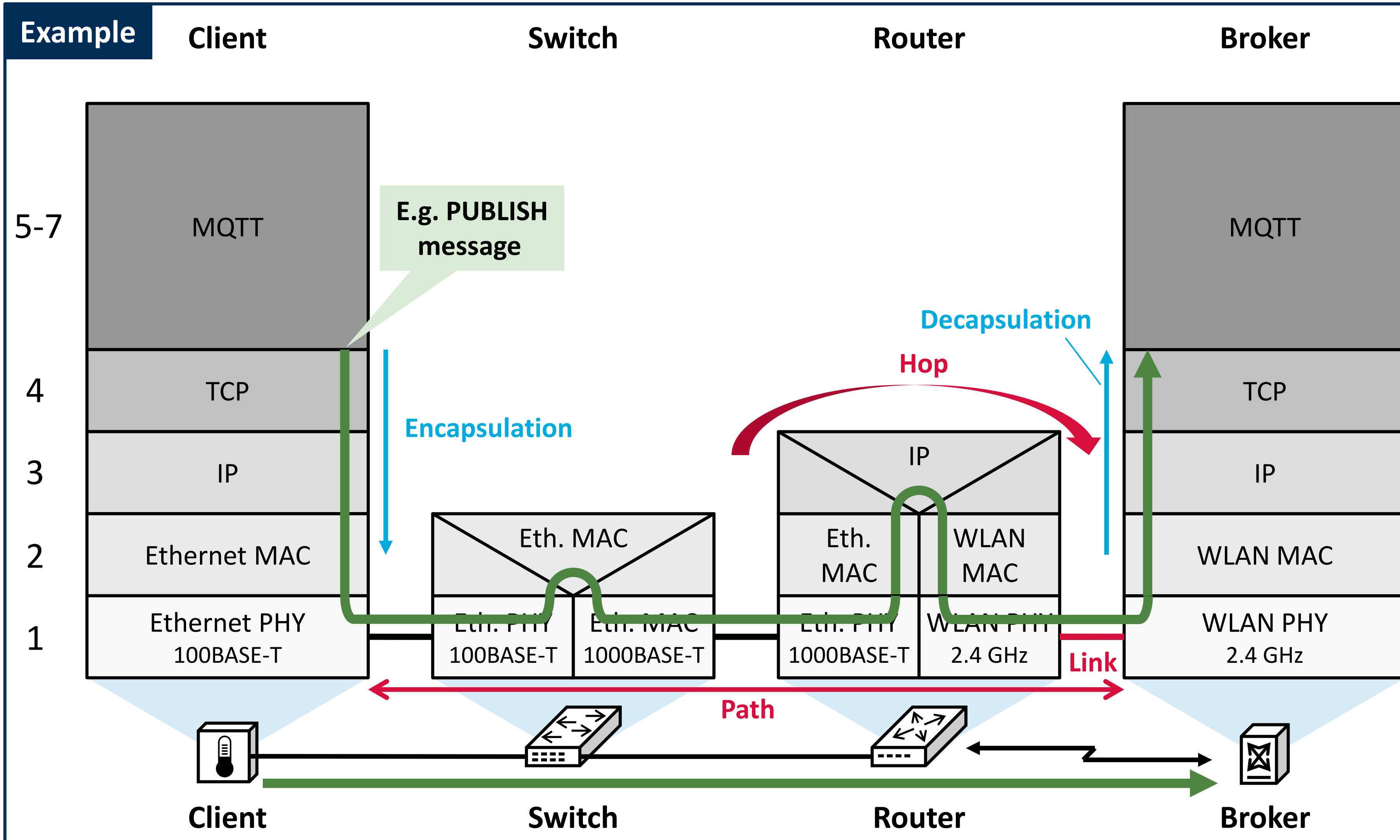
Example

TCP Segment mit 6 byte MQTT Payload:



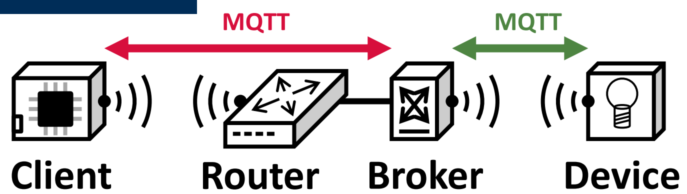
Same principle for encapsulation in IPv6

MQTT Messages
Example for Traversal of Protocol Stacks



MQTT Example

Example (1/3) Control of a smart home socket (from Delock) powering a light bulb



No.	Source	Destination	Prot.	Info
1	dc:a6:32:87:7b:c8	ff:ff:ff:ff:ff:ff	ARP	Who has 192.168.0.1? Tell 192.168.0.184
2	ec:08:6b:53:39:da	dc:a6:32:87:7b:c8	ARP	192.168.0.1 is at ec:08:6b:53:39:da
3	192.168.0.184	192.168.2.1	TCP	37094 → 1883 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128
4	192.168.2.1	192.168.0.184	TCP	1883 → 37094 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=8
5	192.168.0.184	192.168.2.1	TCP	37094 → 1883 [ACK] Seq=1 Ack=1 Win=64256 Len=0
6	192.168.0.184	192.168.2.1	**MQTT**	**Connect Command**
7	192.168.2.1	192.168.0.184	TCP	1883 → 37094 [ACK] Seq=1 Ack=38 Win=29200 Len=0
8	192.168.2.1	192.168.0.184	**MQTT**	**Connect Ack**
9	192.168.0.184	192.168.2.1	TCP	37094 → 1883 [ACK] Seq=38 Ack=5 Win=64256 Len=0
10	192.168.0.184	192.168.2.1	**MQTT**	**Subscribe Request (id=1) [stat/delock/POWER]**
11	192.168.2.1	192.168.0.184	**MQTT**	**Subscribe Ack (id=1)**
12	192.168.0.184	192.168.2.1	**MQTT**	**Subscribe Request (id=2) [tele/delock/#]**
13	192.168.2.1	192.168.0.184	**MQTT**	**Subscribe Ack (id=2)**
14	192.168.0.184	192.168.2.1	TCP	37094 → 1883 [ACK] Seq=82 Ack=15 Win=64256 Len=0
15	192.168.2.1	192.168.0.184	**MQTT**	**Publish Message [tele/delock/LWT]**
16	192.168.0.184	192.168.2.1	TCP	37094 → 1883 [ACK] Seq=82 Ack=40 Win=64256 Len=0
17	192.168.0.184	192.168.2.1	TCP	37096 → 1883 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128
29	192.168.2.1	192.168.0.184	**MQTT**	**Publish Message [stat/delock/POWER]**
30	192.168.0.184	192.168.2.1	TCP	37094 → 1883 [ACK] Seq=82 Ack=63 Win=64256 Len=0
43	192.168.2.1	192.168.0.184	**MQTT**	**Publish Message [stat/delock/POWER]**
44	192.168.0.184	192.168.2.1	TCP	37094 → 1883 [ACK] Seq=82 Ack=87 Win=64256 Len=0
45	192.168.0.184	192.168.2.1	**MQTT**	**Disconnect Req**
46	192.168.0.184	192.168.2.1	TCP	37094 → 1883 [FIN, ACK] Seq=84 Ack=87 Win=64256 Len=0
47	192.168.2.1	192.168.0.184	TCP	1883 → 37094 [FIN, ACK] Seq=87 Ack=85 Win=29200 Len=0
48	192.168.0.184	192.168.2.1	TCP	37094 → 1883 [ACK] Seq=85 Ack=88 Win=64256 Len=0

MQTT Example
Selected Messages

Example (2/3) **Control of a smart home socket (from Delock) powering a light bulb**

CONNECT:

```
Frame 6: 91 bytes on wire (728 bits), 91 bytes captured (728 bits) on interface 0
Ethernet II, Src: dc:a6:32:87:7b:c8, Dst: ec:08:6b:53:39:da
Internet Protocol Version 4, Src: 192.168.0.184, Dst: 192.168.2.1
Transmission Control Protocol, Src Port: 37094, Dst Port: 1883, Seq: 1, Ack: 1, Len: 37
MQ Telemetry Transport Protocol, Connect Command
    Header Flags: 0x10, Message Type: Connect Command
        0001 .... = Message Type: Connect Command (1)
        .... 0000 = Reserved: 0
    Msg Len: 35
    Protocol Name Length: 4
    Protocol Name: MQTT
    Version: MQTT v3.1.1 (4)
    Connect Flags: 0x02, QoS Level: At most once delivery (Fire and Forget), Clean Session Flag
        0... .... = User Name Flag: Not set
        .0.. .... = Password Flag: Not set
        ..0. .... = Will Retain: Not set
        ...0 0... = QoS Level: At most once delivery (Fire and Forget) (0)
        .... .0.. = Will Flag: Not set
        .... ..1. = Clean Session Flag: Set
        .... ...0 = (Reserved): Not set
    Keep Alive: 60
    Client ID Length: 23
    Client ID: mosqsub|708-raspberrypi
```

Frame 6

Explanation

- Ethernet MAC Addresses: Client dc:a6:32:87:7b:c8, Router ec:08:6b:53:39:da
- IPv4 Addresses: Client 192.168.0.184, Broker 192.168.2.1
- Port numbers: Client 37094, Broker 1883
- Application: Client mosquitto_sub, Broker mosquitto

MQTT Example
Selected Messages (Contd.)

Example (3/3) **Control of a smart home socket (from Delock) powering a light bulb**

SUBSCRIBE:

```
Frame 10: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface 0
Ethernet II, Src: dc:a6:32:87:7b:c8, Dst: ec:08:6b:53:39:da
Internet Protocol Version 4, Src: 192.168.0.184, Dst: 192.168.2.1
Transmission Control Protocol, Src Port: 37094, Dst Port: 1883, Seq: 38, Ack: 5, Len: 24
MQ Telemetry Transport Protocol, Subscribe Request
    Header Flags: 0x82, Message Type: Subscribe Request
        1000 .... = Message Type: Subscribe Request (8)
        .... 0010 = Reserved: 2
    Msg Len: 22
    Message Identifier: 1
    Topic Length: 17
    Topic: stat/delock/POWER
    Requested QoS: At most once delivery (Fire and Forget) (0)
```

Frame 10

PUBLISH:

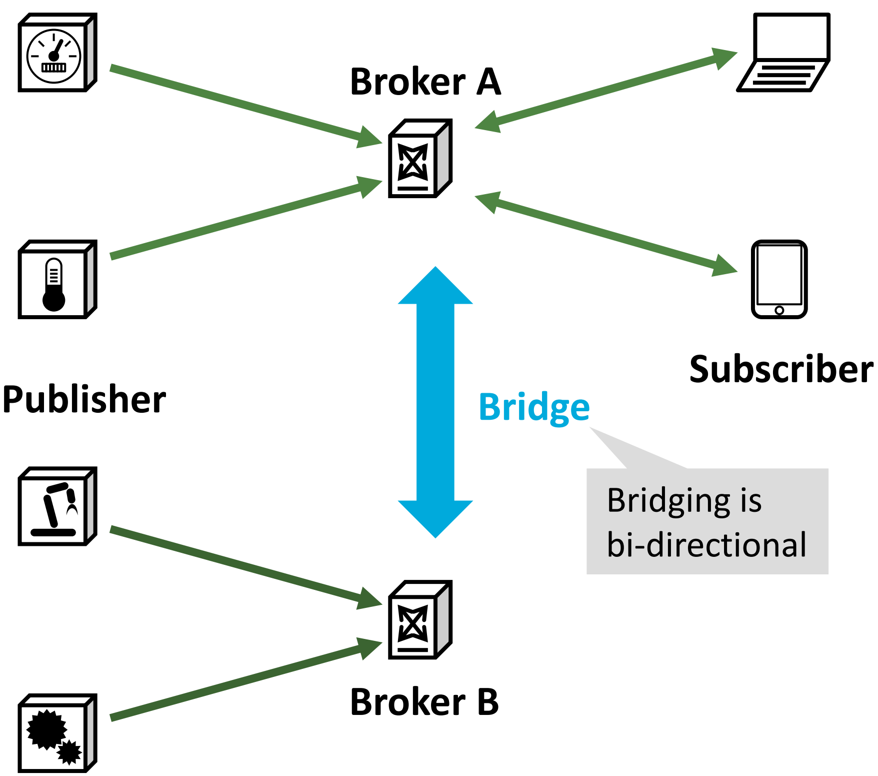
```
Frame 29: 77 bytes on wire (616 bits), 77 bytes captured (616 bits) on interface 0
Ethernet II, Src: ec:08:6b:53:39:da, Dst: dc:a6:32:87:7b:c8
Internet Protocol Version 4, Src: 192.168.2.1, Dst: 192.168.0.184
Transmission Control Protocol, Src Port: 1883, Dst Port: 37094, Seq: 40, Ack: 82, Len: 23
MQ Telemetry Transport Protocol, Publish Message
    Header Flags: 0x30, Message Type: Publish Message, QoS Level: At most once delivery (Fire and Forget)
        0011 .... = Message Type: Publish Message (3)
        .... 0... = DUP Flag: Not set
        .... .00. = QoS Level: At most once delivery (Fire and Forget) (0)
        .... ...0 = Retain: Not set
    Msg Len: 21
    Topic Length: 17
    Topic: stat/delock/POWER
    Message: ON
```

Frame 29

MQTT Broker

- Broker is an **MQTT-specific server logic**
  - Clients connect to brokers
  - Brokers manage the hierarchy of topics
- Brokers receive all messages from publishers and forward them to appropriate subscribers
  - If there are no subscribers for a topic, message is discarded
  - Exception: If the message is flagged as retained, the last retained message is stored in broker
  - Retained messages enable new subscribers to a topic to receive the most recent value immediately
- **Last-will message**
  - Broker keeps track of all the session to clients
  - Clients can specify a last-will message, i.e., a normal MQTT message with a topic, retained message flag, QoS, and payload
  - Last-will message is stored in the broker
  - When a client disconnects ungracefully, the broker sends the last-will message to all subscribed clients of the last-will message topic
  - Last-will message is discarded by the broker if a client disconnects gracefully, i.e., with DISCONNECT message

MQTT Broker
Bridges



- **Bridging** connects **multiple brokers**
- Example use cases
  - Load balancing (e.g., horizontal scaling)
  - Resiliency, redundancy

MQTT Broker
Security

- **User authentication and authorization** possible
  - Username and password
  - Tokens, e.g. OAuth 2.0 token
  - Unencrypted transport in plaintext (port 1883)
- Better alternative **MQTT over TLS** (port 8883)
  - Confidentiality and integrity protection
  - Encryption of MQTT communication
  - Authentication by X.509 client certificates possible

Software Examples

- **MQTT Brokers (and Clients)**
  - Eclipse Mosquitto (in C): https://mosquitto.org
  - HiveMQ (in Java): https://www.hivemq.com
  - …
- **MQTT Client Libraries**
  - Eclipse Paho (Java, C, Python, …): https://www.eclipse.org/paho
  - …
- Other MQTT-enabled applications
  - Node-RED for browser-based apps: https://nodered.org
  - HAProxy for scaling: https://www.haproxy.org
  - …

Software Examples
Simple Python Client on Raspberry Pi

Example **MQTT Publishing of Temperature and Humidity from DHT22 Sensor with Paho Library**

```
#!/usr/bin/python3

# Import required Python libraries
import paho.mqtt.client as mqtt
import Adafruit_DHT
import time

# Configuration
dht22gpiopin=17
broker='broker.picoIOT.test'
port=1883
publish_topic="raspberry/dht22"
clientid='raspberry-mqtt-dht22'
username='mosquitto'
password='password'
qos=1
retain_message=True

while True:
    # Establish the MQTT connection
    client=mqtt.Client(clientid)
    client.username_pw_set(username, password)
    client.connect(broker, port)
    client.loop_start()

    # Publish temperature and humidity
    humidity, temperature = Adafruit_DHT.read_retry(Adafruit_DHT.AM2302, dht22gpiopin)
    client.publish("{}/temperature".format(publish_topic),"{:.1f}".format(temperature),qos,retain_message)
    client.publish("{}/humidity".format(publish_topic),"{:.1f}".format(humidity),qos,retain_message)

    client.disconnect()
    client.loop_stop()
    time.sleep(10)
```

Alternatives to MQTT

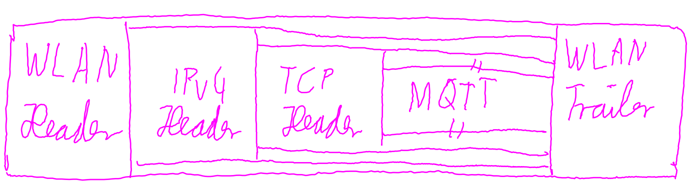
- **Constrained Application Protocol (CoAP)**
  - Lightweight protocol for M2M without full TCP/IP stack
  - UDP-based protocol similar to HTTP
  - Similar to MQTT
- **Advanced Message Queuing Protocol (AMQP)**
  - Message-oriented middleware, e.g. for enterprise integration
  - Binary protocol
  - Widely used in large-scale distributed systems

Example for MQTT Message Encapsulation

Exercise

A sensor sends an MQTT message by Wireless LAN (WLAN) to an MQTT broker that is running on a server with IP address 192.168.2.1. Protocol data units in WLAN, i.e. frames, include both a header and a trailer. Sketch the resulting structure of a WLAN frame that encapsulates MQTT data if no encryption is used, i.e., the series of bytes that are sent by the sensor.

Structure of an MQTT message over TCP, IPv4 and WLAN:

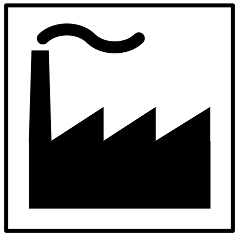
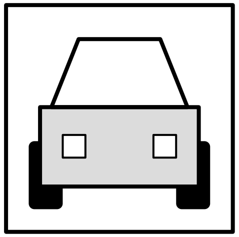


Industrial Cyber-Physical Networks

Prof. Dr.-Ing. Michael Scharf

Hochschule Esslingen – University of Applied Sciences

Industrial Revolutions

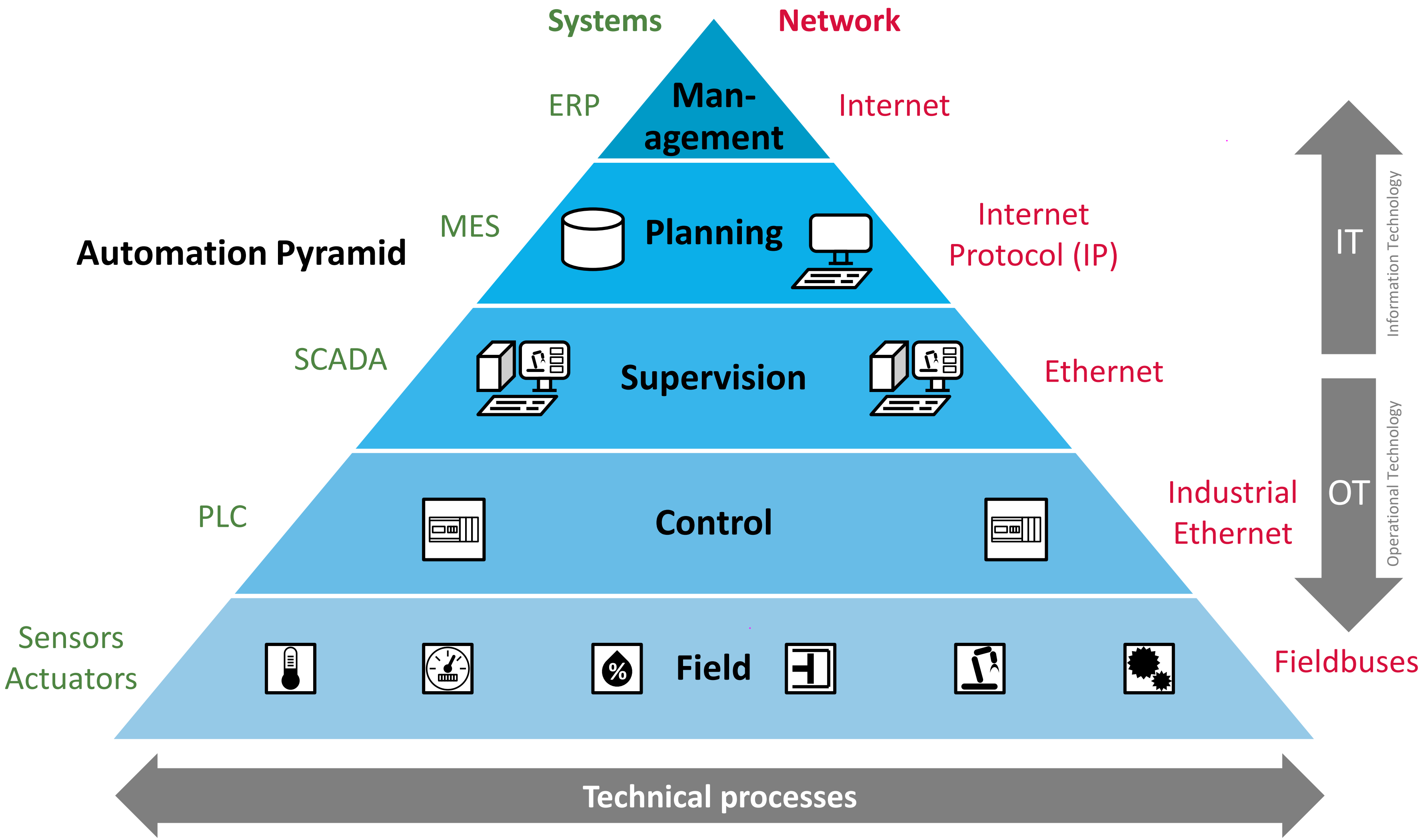
			
1st 1784	**2nd** 1870	**3rd** 1969	**4th** Today
Mechanization	**Mass production**	**Automation**	**Cyber-Physical Systems**
▪ Water and steam power ▪ Machines ▪ Railroads	▪ Electricity and electric power ▪ Assembly line ▪ Roads	▪ Electronics and computer ▪ Robotics ▪ Internet and WWW	▪ Ubiquitous connectivity ▪ Smart devices ▪ Internet of Things (IoT)

Industrial Revolutions
Example Use Cases Related to “Industry 4.0”

- Manufacturing (“Smart factory”)
  - Robotics
  - “Digital Twin”
- Vertical industries
  - Transportation, e.g. railroad
  - Energy and utilities, e.g., oil and gas industry, electrical power grid
  - …
- Emerging new use cases
  - Augmented reality (AR) / virtual reality (VR)
  - Remotely operated vehicles such as drones, cars, etc.
  - …

Operational Technology (OT), also known as Industrial Control System (ICS), differs to Information Technology (IT) using Commodity-of-the-Shelf (COTS) technology

Industrial Networks

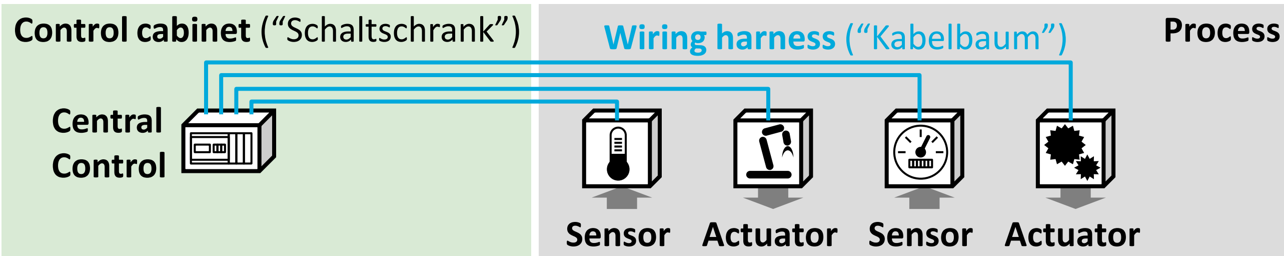


Industrial Networks
Industrial Automation Systems

- **Enterprise Resource Planning (ERP)**
  - Management of business processes
  - Example: SAP S/4HANA
- **Manufacturing Execution System (MES)**
  - Management of production processes
  - Example: SAP Manufacturing Execution
- **Supervision Control And Data Acquisition (SCADA)**
  - High-level supervision of production processes
  - Typically including a **Human-Machine Interface (HMI)**
  - Example: Siemens WinCC
- **Programmable Logic Control (PLC)**
  - Also known in German as "*Speicherprogrammierbare Steuerung (SPS)*"
  - Real-time control of production processes
  - Example: Siemens Simatic
- **Sensors and Actuators**

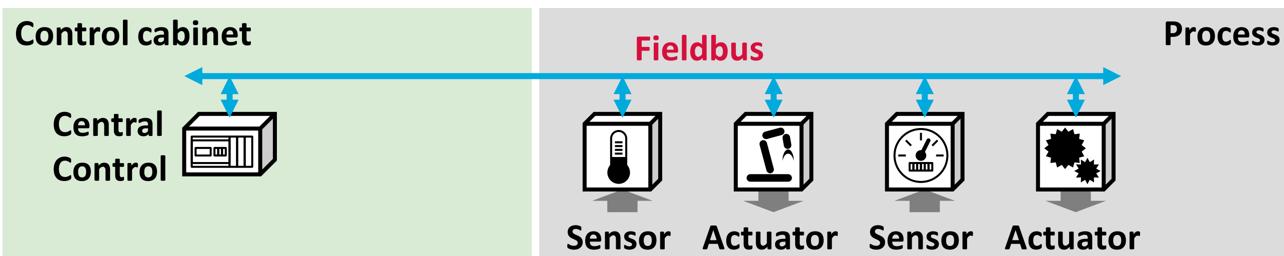
Industrial Networks
Historical Evolution

1. **Wiring harness** (main approach until ca. 1980)



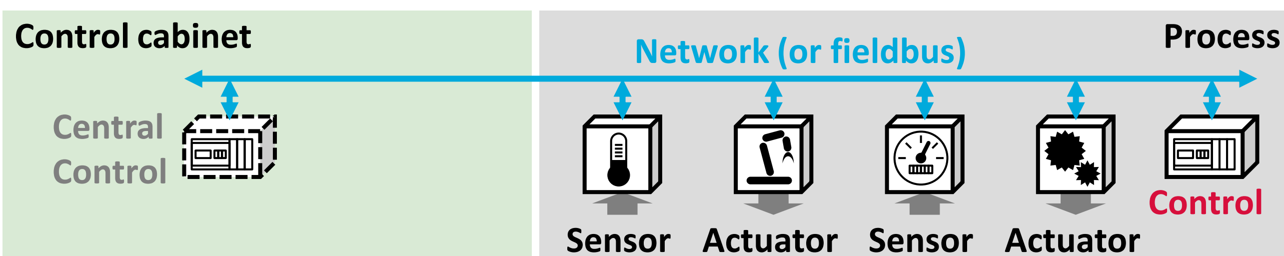
- Central control of devices out of a control cabinet
- One cable per signal

2. **Field bus** (since ca. 1980)



- Fieldbus for signals from/to multiple devices
- Simple technologies using shared bus

3. **Networks** (since ca. 2000)

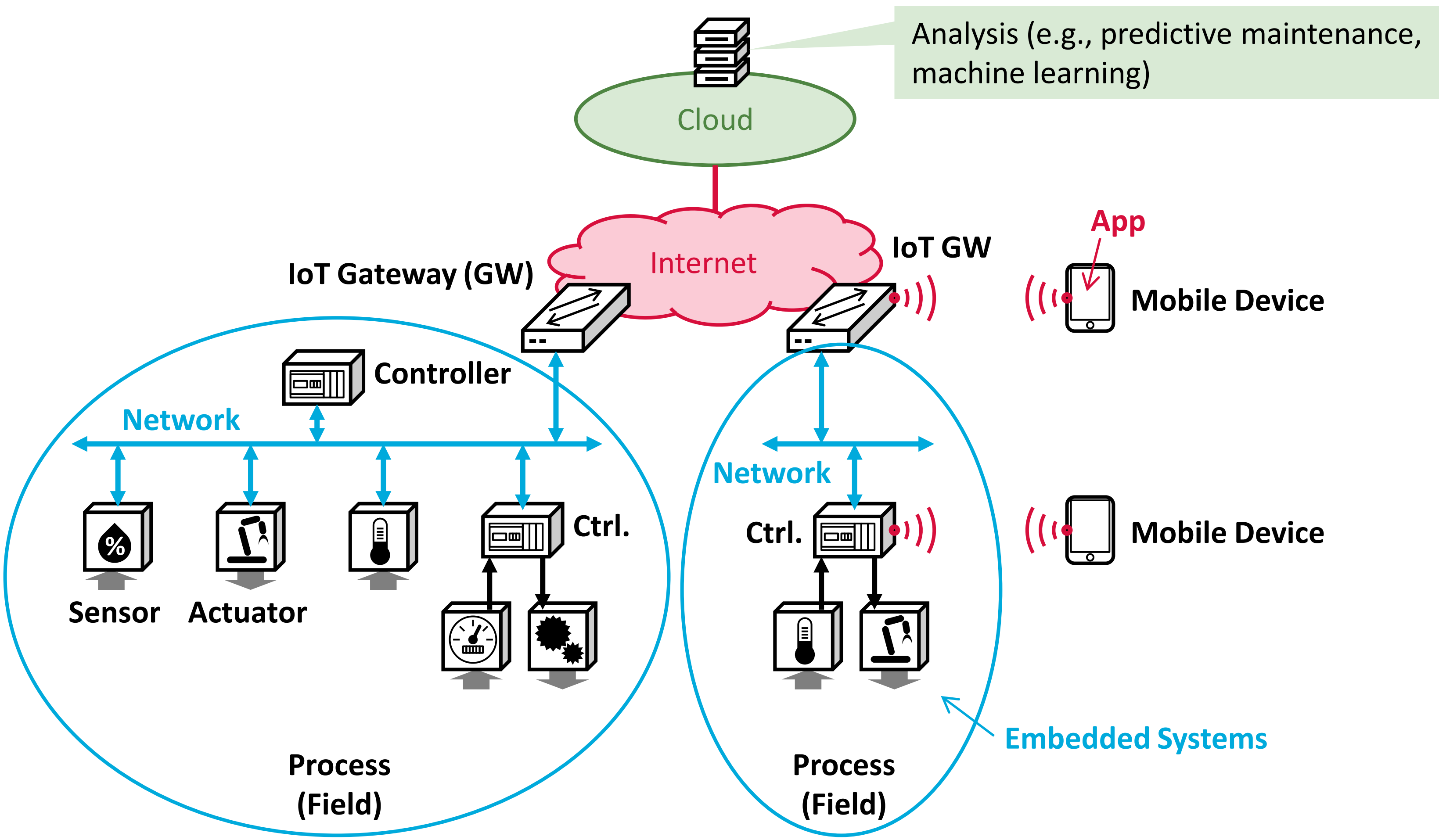


- Distributed control, several controllers
- Complex topologies
- Trend towards Industrial Ethernet

4. **Internet Connection** (since ca. 2015)

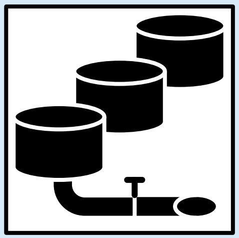
Industrial Networks
Internet Connection

Example Example for an industrial network



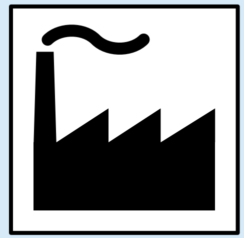
Industrial Networks
Industry Sectors

Process Automation



- Relatively slow processes
- Oil, gas, chemical industry, energy, water, …
- Pumps, compressors, mixers, temperature/ pressure/flow sensors, …
- **Delay ~1s**

Factory Automation



- Time-critical processes
- Most manufacturing, food and beverages, pharmaceuticals, …
- Metal forming, welding, stamping, cutting, packaging, filling, …
- **Delay 1 ms – 100 ms**

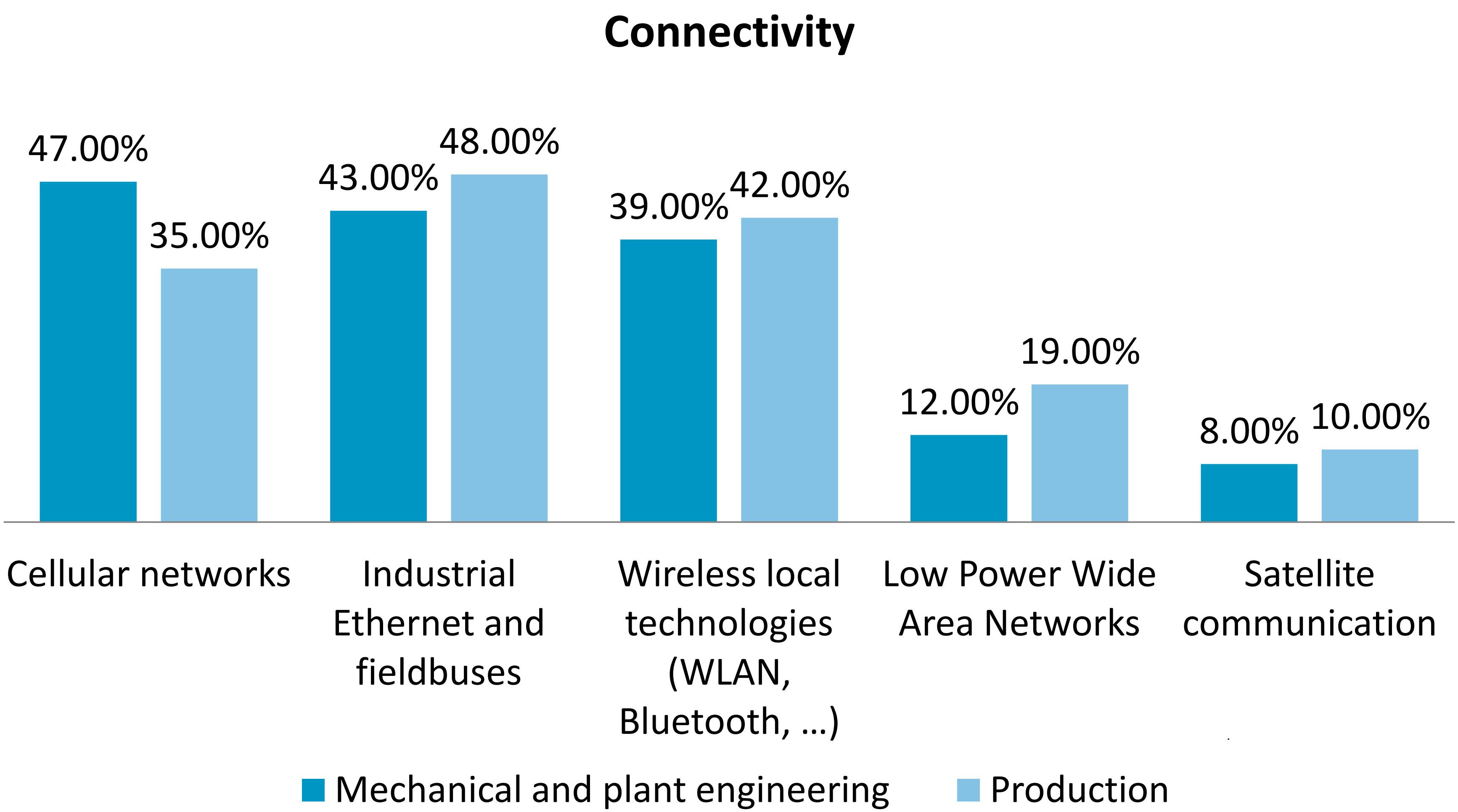
Motion Control

- Multi-axis motion control
- Utilities, advanced factory automation, life/equipment safety
- Printing presses, wire drawing, web making, picking and placing, …
- **Delay 100 µs – 10 ms**

Industrial Networks
Survey of Deployed Network Technologies

Example

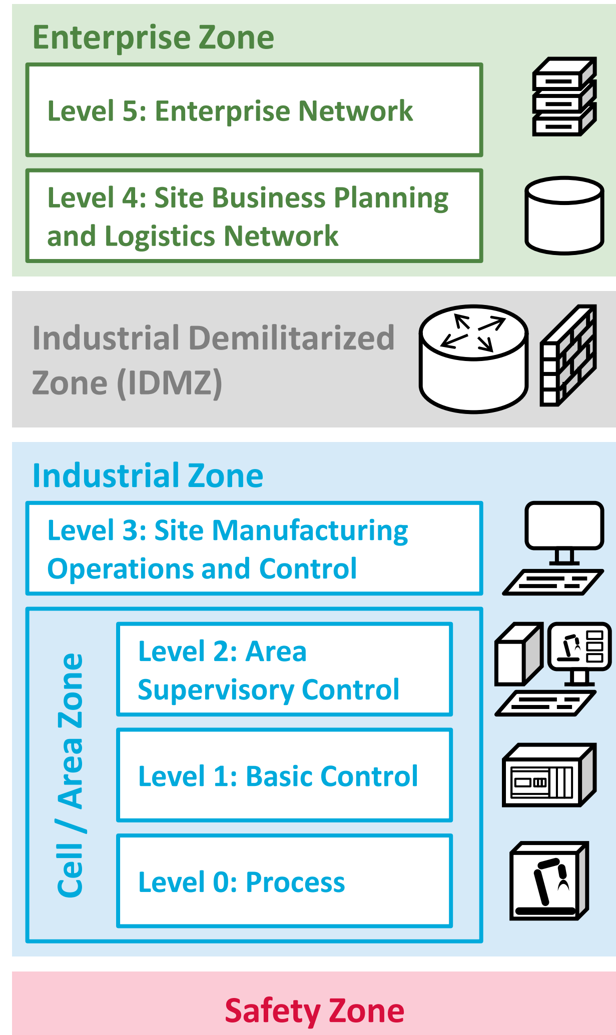
Source: iX Special 2018 – Industrial Internet of Things



Industrial Networks
Typical Network Architecture

- **Enterprise Zone**
  - Level 5: Enterprise Network
  - Level 4: Site Business Planning and Logistics Network
- **Industrial Demilitarized Zone** (IDMZ)
  - Border between IT and OT networks
  - Firewalls with security rules limit communication
- **Industrial Zone** (Manufacturing Zone)
  - Level 3: Site Manufacturing Operations and Control
  - Level 2: Area Supervisory Control
  - Level 1: Basic Control
  - Level 0: Process

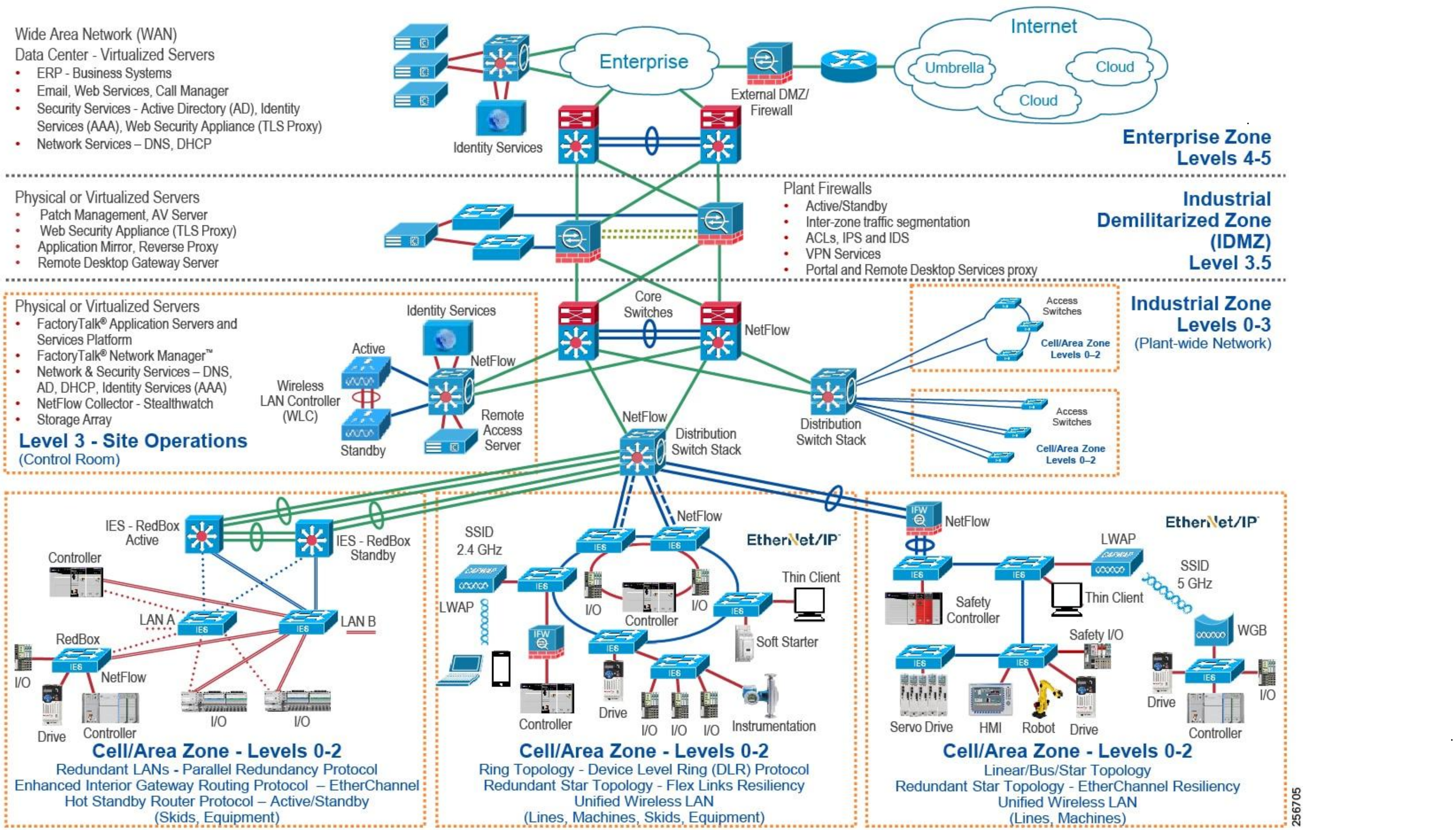
Cells (areas)



Industrial Networks
IT/OT Network Example 1

Example **Cisco & Rockwell Automation**

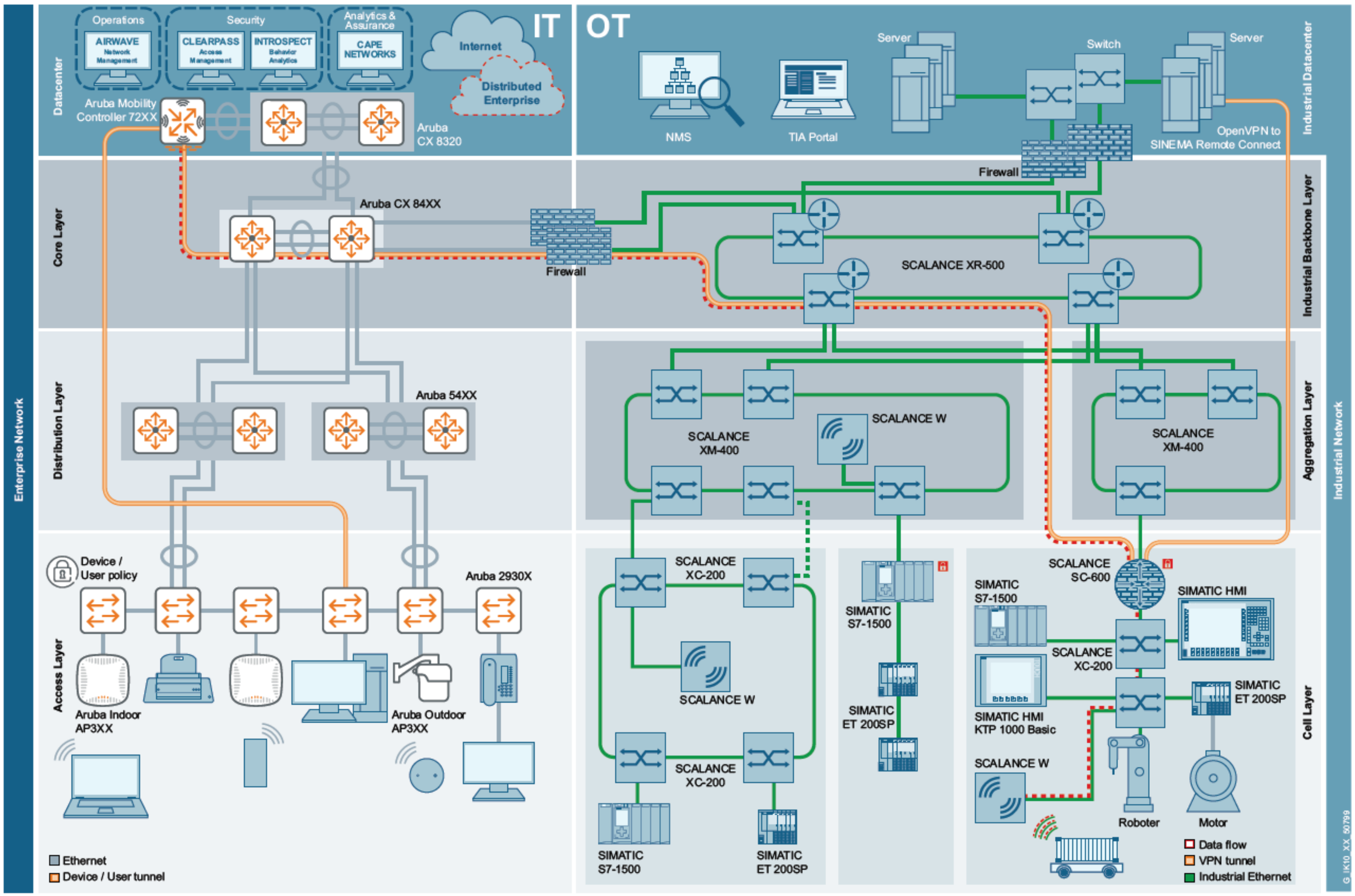
Source: https://www.cisco.com/c/en/us/td/docs/solutions/Verticals/CPwE/5-1/Phy_Arch/CPwE_PhyArch_AppGuide/CPwE_PhyArch_Chap1.html



Industrial Networks
IT/OT Network Example 2

Example **Aruba & Siemens**

Source: "Aruba & Siemens – Partner Solution Overview", https://www.arubanetworks.com/assets/pso/PSO_Siemens.pdf



Network Requirements

	IT Network **Information Technology**	**OT Network** **Operational Technology**	**ISP Network** **Internet Service Provider**
Physical environment	• Office or server rooms • Air conditioned • Protected against dust, water, etc.	• Outside or shop floor • Wide temperature range • Electrical disturbance, mechanical vibrations, etc.	• Outdoor cabinets possible • Extended temperature range • Typically some protection against dust and water
Topology	Star/tree in LAN, mesh in WAN	Static linear or ring structures	Mesh or ring
Port density	High	Typically low	Low to high
Outage risk	Some commercial impact	Production downtime	Significant commercial impact
Devices	Commodity-of-the-Shelf (COTS)	Often proprietary	Few major vendors
Lifecycle	3 – 5 years	Decades possible	5 – 10 years
Rollout	Dedicated IT department	Machine rollout personal	Dedicated rollout personal
Operation	Dedicated IT department	Part of SCADA operation	Dedicated operation team
Priorities	1. Efficiency and usability 2. Sufficient performance, e.g. high throughput 3. Confidentiality, integrity	1. Availability and reliability 2. Determinism and real-time transmission 3. Safety and plant protection	1. Availability and reliability 2. Performance to fulfill Service Level Agreements (SLAs) 3. Confidentiality, integrity

Trend to IT/OT convergence ...

Network Requirements
Summary of key requirements

- **Availability** (“Verfügbarkeit”)
- **Reliability** (“Zuverlässigkeit”)
- **Safety** (“Funktionssicherheit”)
- **Security** (“Sicherheit”)
- **Real-time support** (“Echtzeitunterstützung”)
- … and …
  - Robustness in rough environments
  - Electromagnetic compatibility
  - Long lifetime
  - Ease of maintenance
  - Low Operational Expenditures (OPEX)
  - Low Capital Expenditures (CAPEX)
  - etc.

Network Requirements
Ingress Protection (IP) Rating

Example IP ratings according to standard IEC 60529

Source: https://www.iec.ch/ip-ratings

IEC Making electrotechnology work for you.

Ingress protection (IP) ratings guide

IP ratings are represented by combining the first and second digits of the below columns

1st numeral - solid foreign objects

0	No protection
1	Protected against solid foreign objects of 50 mm Ø and greater
2	Protected against solid foreign objects of 12,5 mm Ø and greater
3	Protected against solid foreign objects of 2,5 mm Ø and greater
4	Protected against solid foreign objects of 1,0 mm Ø and greater
5	Dust-protected
6	Dust-tight

Example:

IP 6 5

5 → Protected against water jets

6 → Dust-tight

2nd numeral - water

0	No protection	-
1	Protected against vertically falling water drops	Vertically falling drops shall have no harmful effects
2	Protected against vertically falling water drops when enclosure tilted up to 15°	Vertically falling drops shall have no harmful effects when the enclosure is tilted at any angle up to15° on either side of the vertical
3	Protected against spraying water	Water sprayed at an angle up to 60° on either side of the vertical shall have no harmful effects
4	Protected against splashing water	Water splashed against the enclosure from any direction shall have no harmful effects
5	Protected against water jets	Water projected in jets against the enclosure from any directions shall have no harmful effects
6	Protected against powerful water jets	Water projected in powerful jets against the enclosure from any direction shall have no harmful effects
7	Protected against the effects of temporary immersion in water	Ingress of water in quantities causing harmful effects shall not be possible when the enclosure is temporarily immersed in water under standardized conditions of pressure and time
8	Protected against the effects of continuous immersion in water	Ingress of water in quantities causing harmful effects shall not be possible when the enclosure is continuously immersed in water under conditions which shall be agreed between manufacturer and user but which are more severe than for numeral 7
9	Protected against high pressure and temperature water jets	Water projected at high pressure and high temperature against the enclosure from any direction shall not have harmful effects

Example for an IP Rating

Exercise

Image source: https://www.lorixone.io/sites/default/files/2018-11/lorixone_details_0.png

The data sheet of an outdoor wireless base station lists protection according to IP65. What protection does this rating imply?

IP65:

5 -> Protected against water-jets: No harmful effect from water projected by a (small) nozzle from any direction

6 -> Dust-tight: No ingress of dust

Illustration: