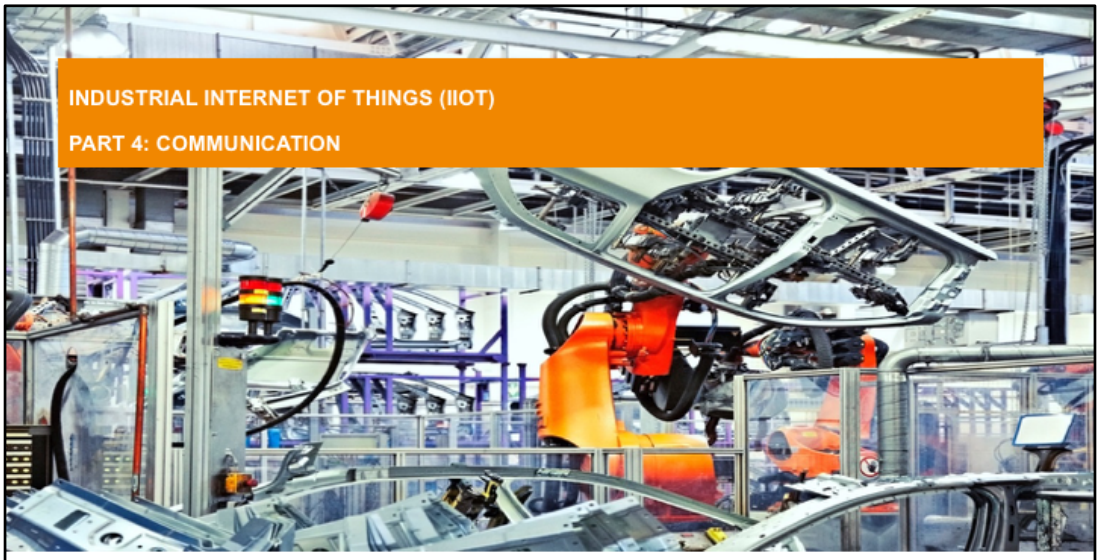


INDUSTRIAL INTERNET OF THINGS (IIOT)

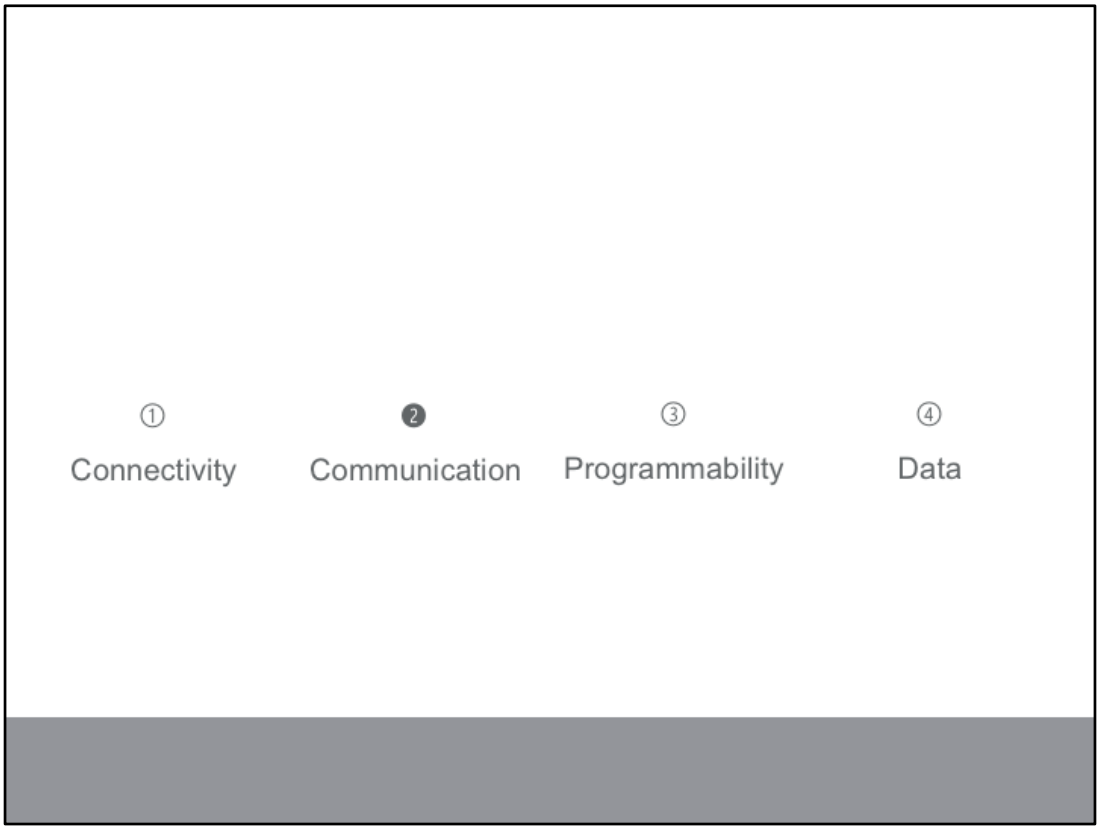
PART 4: COMMUNICATION

AV Lecture in Summer Term 2018
Dr.-Ing. Alexander **Willner**, Ronald Steinke

Fraunhofer FOKUS

THE LAST LECTURE

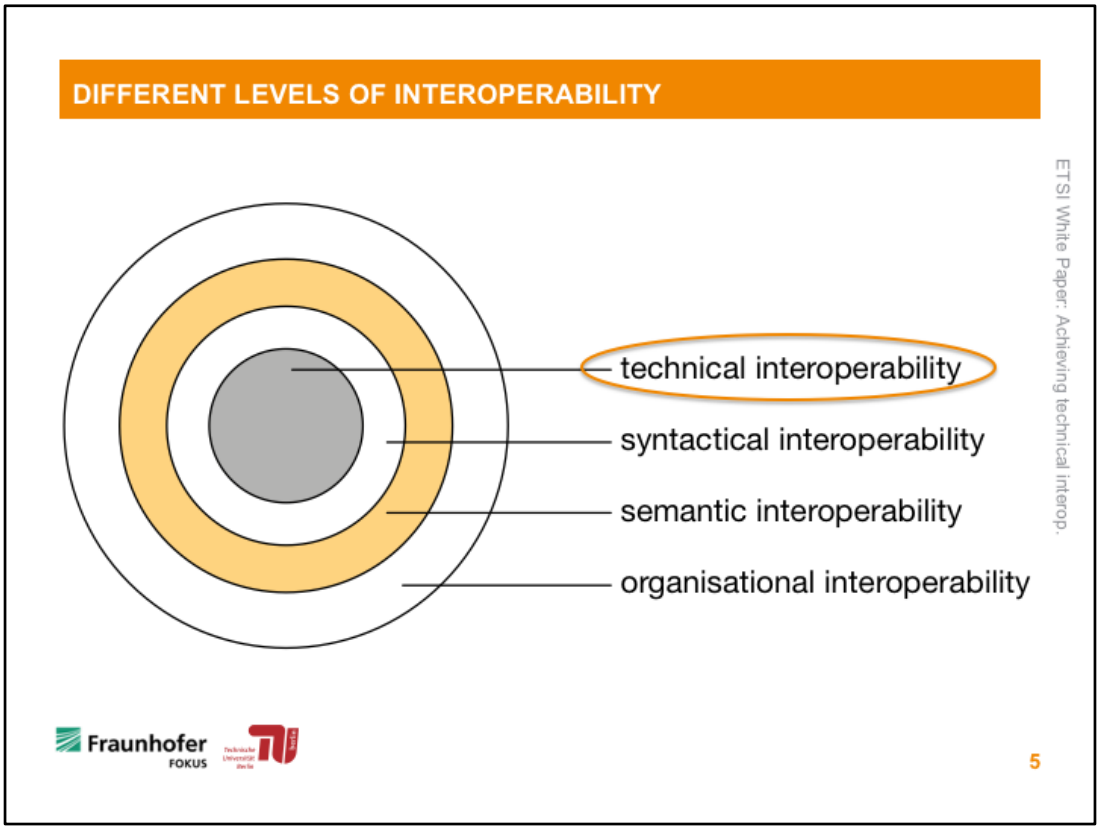
Skipped



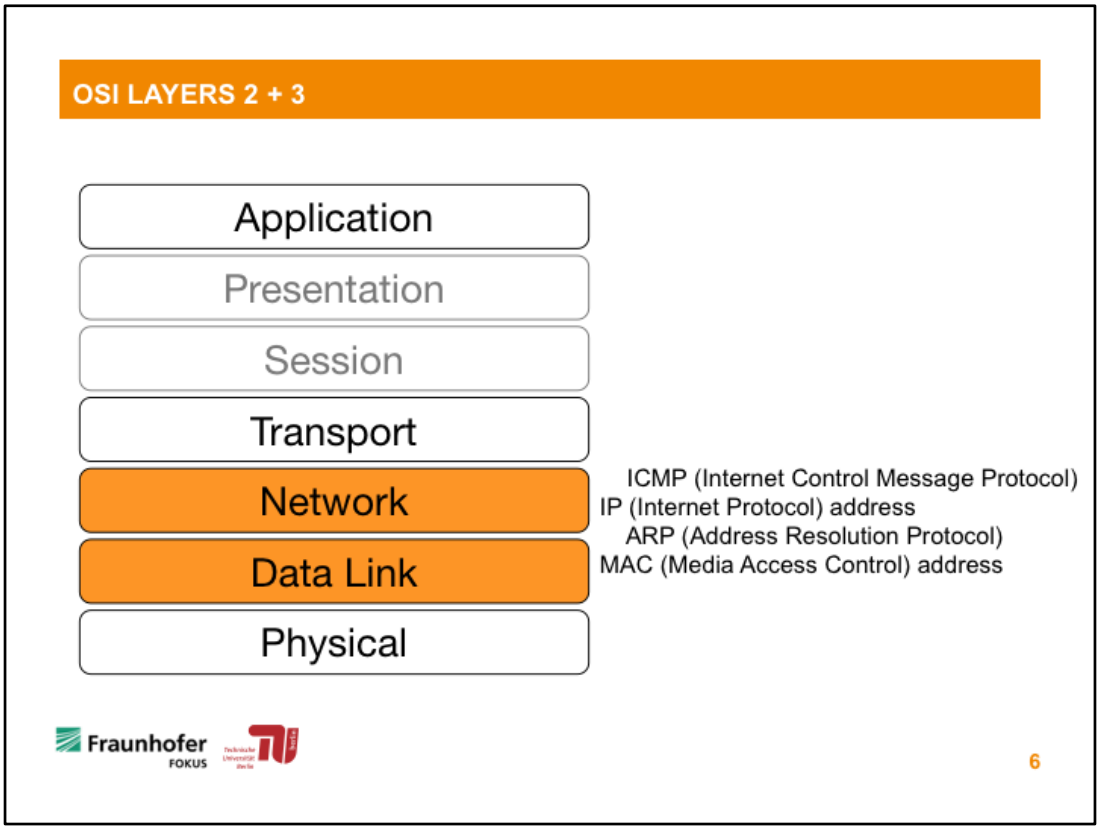
- This lecture is divided into 4 different areas
- Now, we focus on the second one

COMMUNICATION OVERVIEW

30 minutes

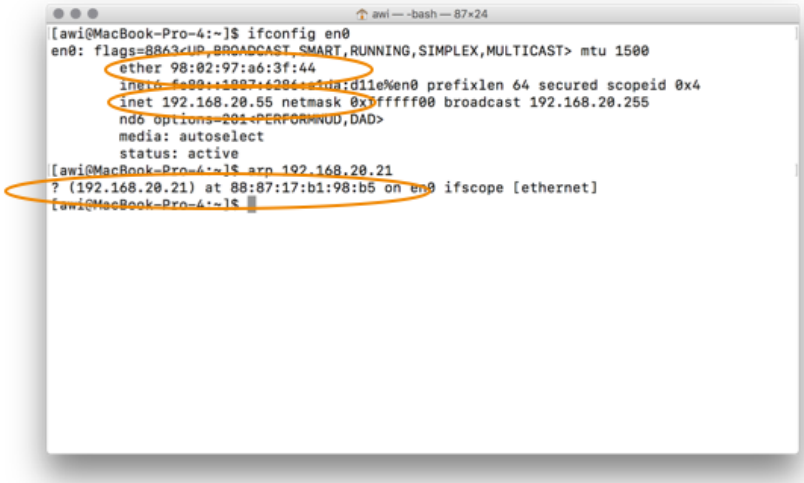


- We still focus on technical interoperability
- However, we start to touch the next two levels as well



- **Remember**: statistical time division multiplexing for a shared medium
- **Remember**: addressing mechanism required for multiple access
- MAC (Media Access Control) address in IEEE 802: 48 bit. Example: 98:02:97:a6:3f:44
    - ARP (Address Resolution Protocol): translating between MAC and IP
    - IP (Internet Protocol) address (v4): 32 bit. Example: 192.168.20.55
    - Internet Control Message Protocol (ICMP): e.g. diagnose reachability of nodes

EXAMPLE (IPV4, MAC, ARP)

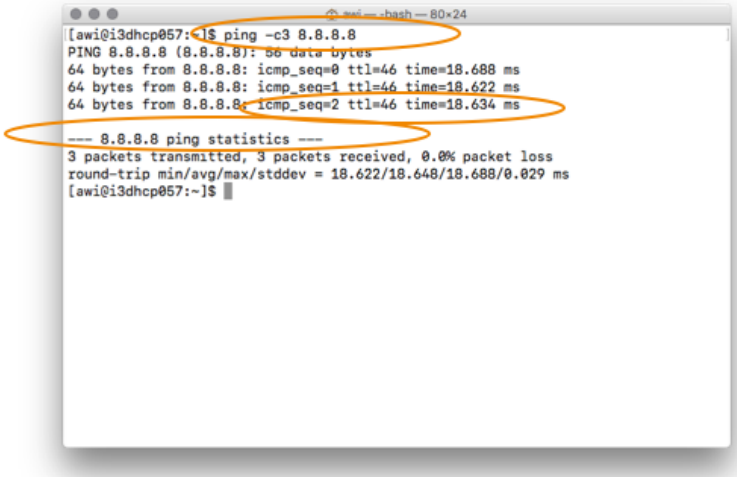


Fraunhofer FOKUS

7

- MAC address is the hardware address
- Will be mapped to IP address with the help of the Address Resolution Protocol (ARP)

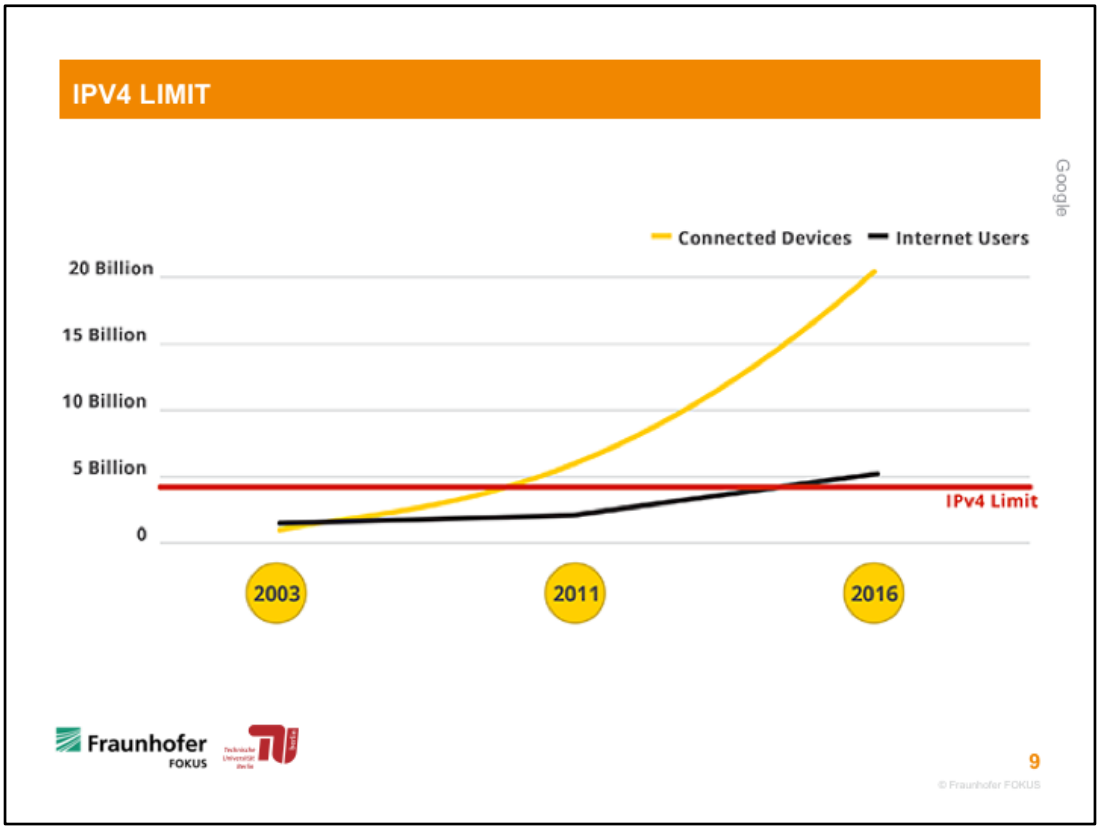
EXAMPLE (ICMP)



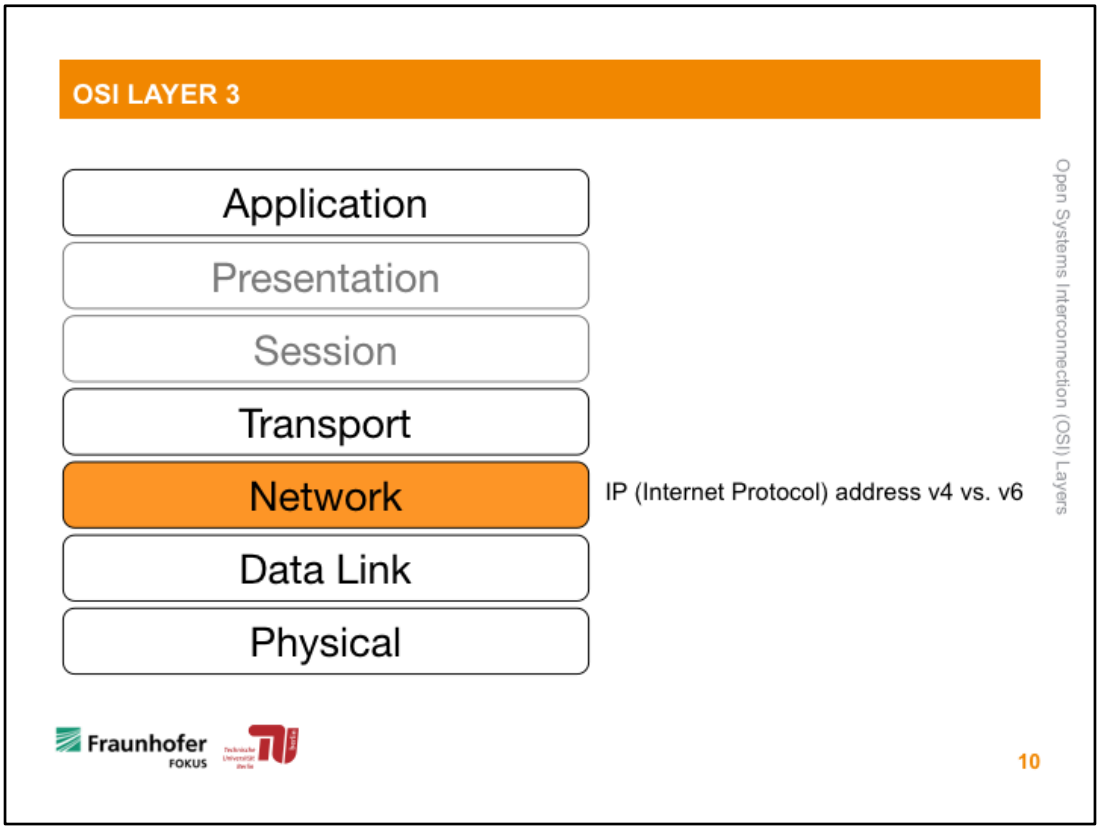
Fraunhofer FOKUS

8

- ICMP for organizing the messaging and routing
- Most popular the ping command

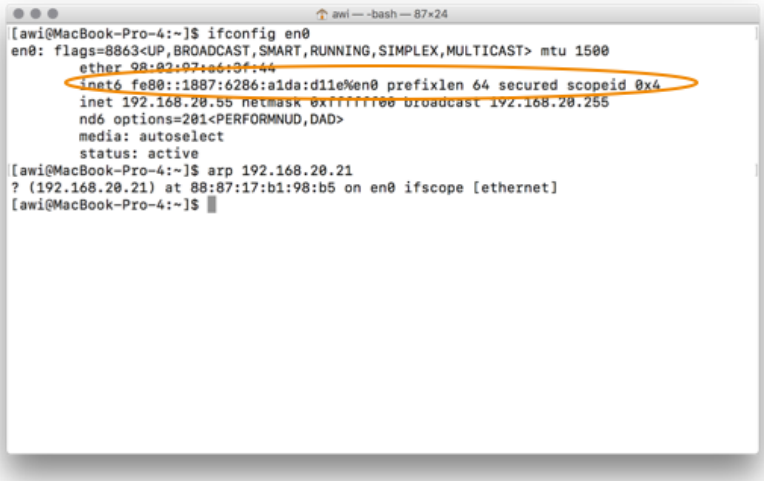


- There will be more and more connected devices that need an IP address
- The limit of the address range for IPv4 was already reached
- IPv6 is needed 128 bit instead of 32 bit



- **Remember:** more IoT devices in the world than people
- IP (Internet Protocol) address (**v4**): 32 bit. Example: 192.168.20.55
  - around 4.3 billion addresses (note: around 7.6 billion people on the world)
  - IPv4 address exhaustion: 31.01.2011, the last address block was allocated
- IP (Internet Protocol) address (**v6**): 128 bit. Example: fe80::1887:6286:a1da:d11e
  - Steve Leibson: “we could assign an IPv6 address to EVERY ATOM ON THE SURFACE OF THE EARTH, and still have enough addresses left to do another 100+ earths.”

EXAMPLE (IPV6)

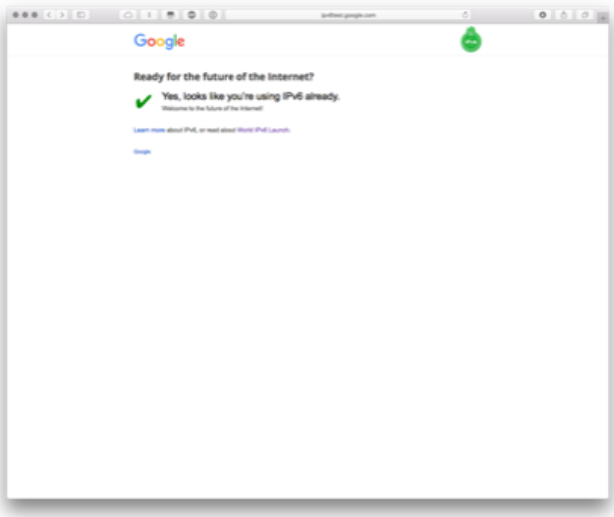


Fraunhofer FOKUS

11

- IPv6 different address notation (hexadecimal instead of decimal)
- Other advantages
  - Local address per interface
  - Better handling of local networks, can also have public IPs
  - Pre-defined address like local gateway or local DNS servers
  - Better autoconfiguration

IPV6 TESTS

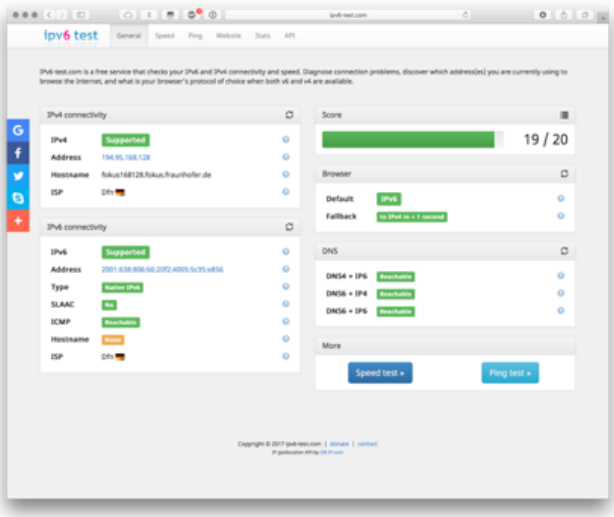


Fraunhofer FOKUS

12

http://ipv6test.google.com/

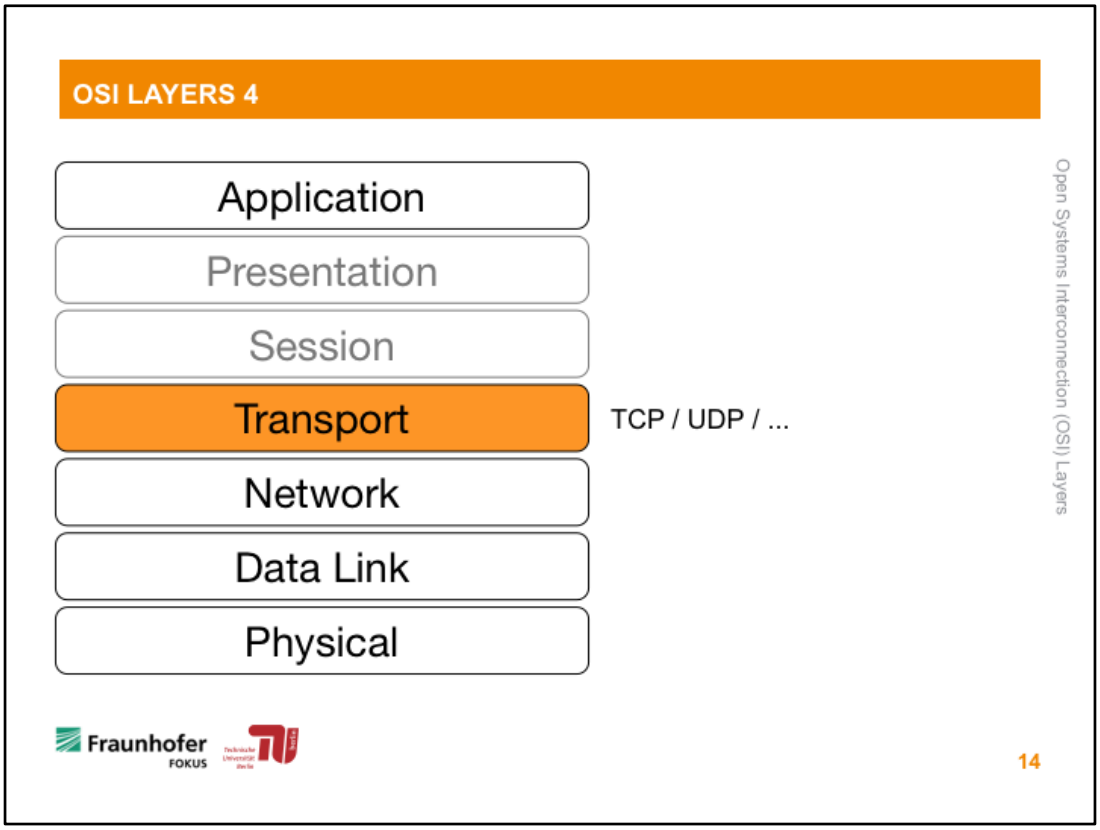
IPV6 TESTS



Fraunhofer FOKUS

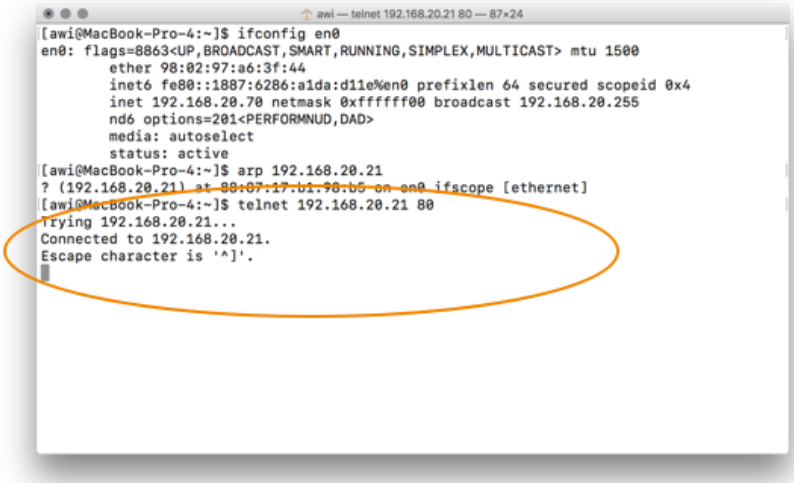
13

- http://ipv6-test.com
- SLAAC = Stateless Address Autoconfiguration



- **Transmission Control Protocol (TCP):** Provides
  - connection-based, reliable, ordered, and error-checked
  - delivery of a stream for
  - web, mail, file transfer, and other applications offering
  - port numbers, handshaking, congestion control, retransmission, ...
- **User Datagram Protocol (UDP):** simple connectionless delivery of datagrams offering checksums and port numbers
- **Remember**: real-time requirement within industrial application domains -> rather UDP or TCP?
- **Remember:** IoT devices may have limited resources -> rather UDP or TCP?

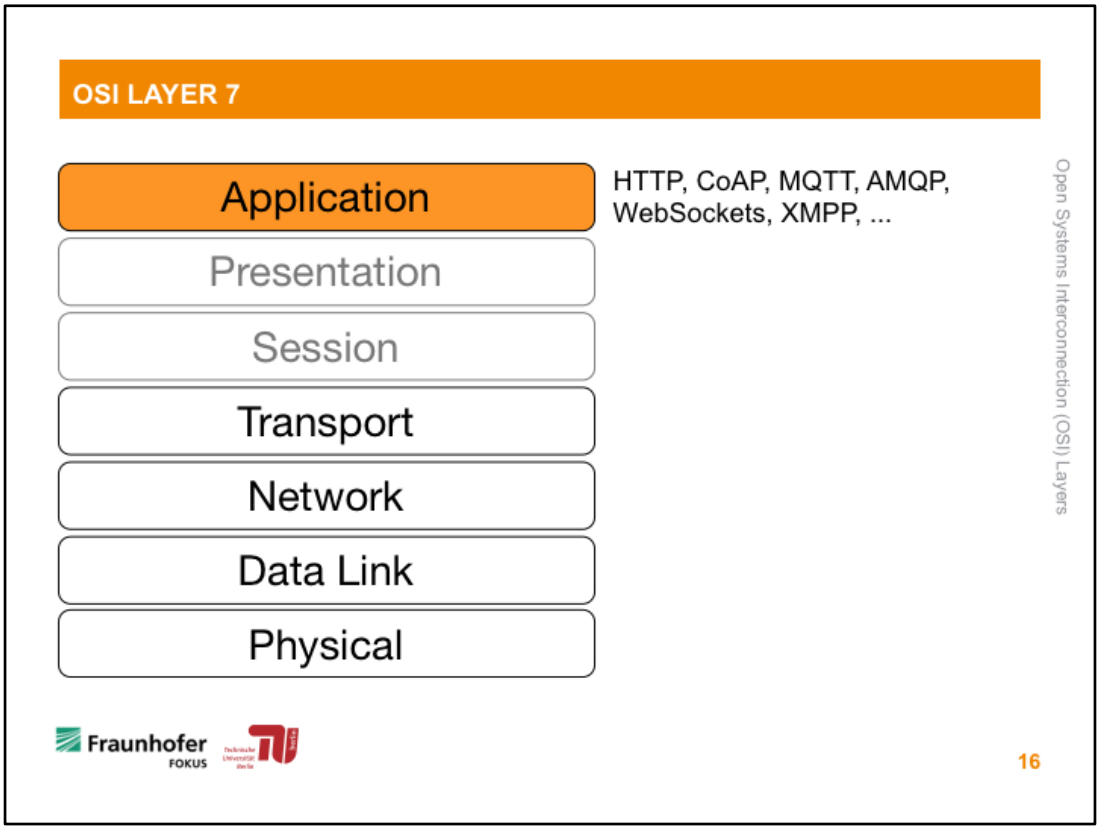
EXAMPLE (TCP)



Fraunhofer FOKUS

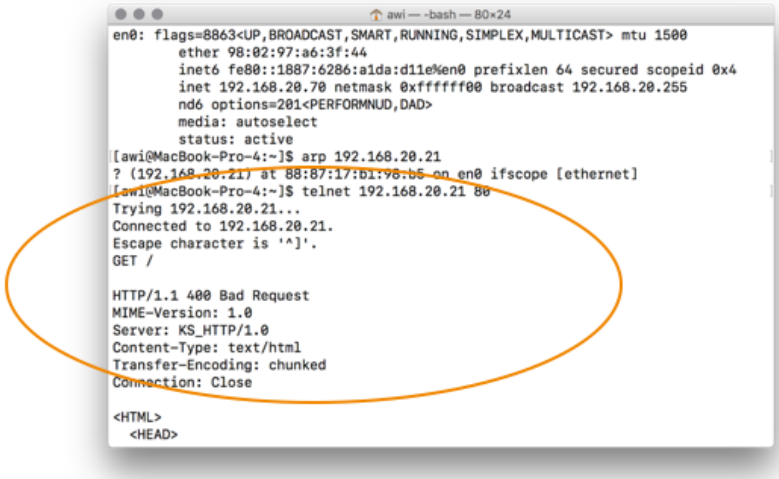
15

- Simple example



- Protocols to provide functionalities for specific applications
- Examples include file transfer, remote login, mail exchange, name resolution, …
- **The applications themselves are not part of this layer**

EXAMPLE (HTTP)



Fraunhofer FOKUS

17

- HTTP can be used directly in TCP connection (here with telnet)
- GET / is the REQUEST
- RESPONSE send back in the same session
- This is a layer 7 protocol
- Why do we get an error message?

EXAMPLE (HTTP)

Fraunhofer FOKUS

19

- HTTP example with curl
- Here only the response headers
- Machine readable call for action (change location)

EXAMPLE (HTTP)

Fraunhofer FOKUS

- HTTP example continuation
- Here the responded body -> a serialization of a data model (more about this later in this course)
- Human readable call for action (change location)

APPLICATION-LEVEL COMMUNICATION PROTOCOLS

Protocol	HTTP/1.1	HTTP/2	CoAP	MQTT(-SN)	AMQP
Standards	IETF RFC 2616 (1996, 1999)	IETF RFC 7540 (2015, based on Google SPDY)	IETF RFC 7252	OASIS	OASIS
Architecture Style	Client/server model	Client/servers model	Client/server model	Brokered style	Brokered style
Transport	TCP	TCP	UDP	TCP (UDP)	TCP
Messaging	Request/Response	Supports multiplexing of request/response + push	Request/Response + Notify	Publish/Subscribe	Publish/Subscribe (P2P or Brokered)
Service levels (QoS)	All messages get the same level of service	Priority mechanism of streams	Confirmable or non-confirmable messages	Three quality of service settings	Three quality of service settings
Data distribution	One-to-one	One-to-one , and one-to-many	One-to-one, and one-to-many	One-to-one , and one-to-many	One-to-one , and one-to-many
Security	TLS (Transport Layer Security)	TLS (Transport Layer Security) - **mandatory**	Datagram Transport Layer Security (DTLS)	Simple Username/Password Authentication, TLS for data encryption	Simple Authentication and Security Layer (SASL), TLS for data encryption
Header	Text-based	Binary (header compression)	4 Bytes binary--based	Fixed-length header of 2 bytes	Header 8 bytes

Fraunhofer FOKUS

22

- Hyper Text Transfer Protocol (HTTP) over TCP: typical client/server model
- Constrained Application Protocol (CoAP) over UDP: for constrained devices including notification
- Message Queue Telemetry Transport (MQTT) over TCP: most relevant publish/subscribe protocol currently in the IoT context
- MQTT for Sensor Networks (MQTT-SN) over UDP: for more constraint devices
- Advanced Message Queuing Protocol (AMQP): more complex publish/subscribe protocol

We'll skip the details about each protocol.

Fraunhofer FOKUS

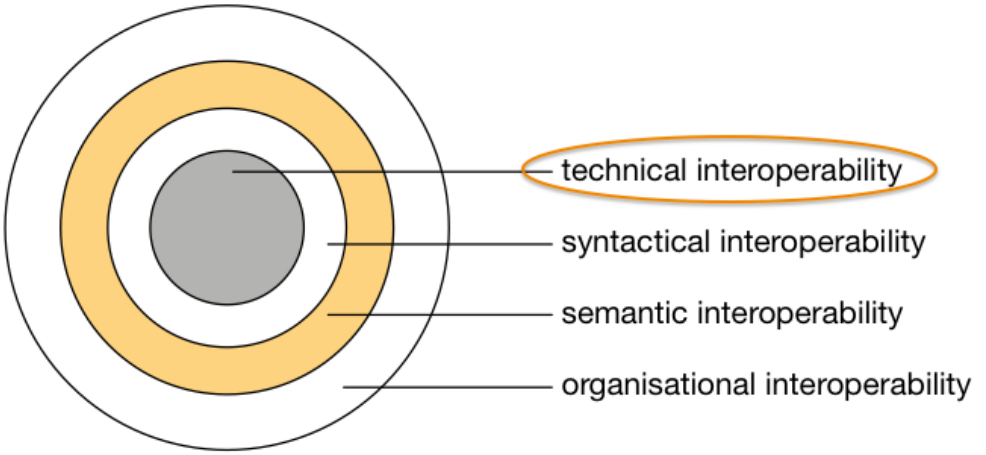
COMMUNICATION OVERVIEW

Questions?

MIDDLEWARE OVERVIEW

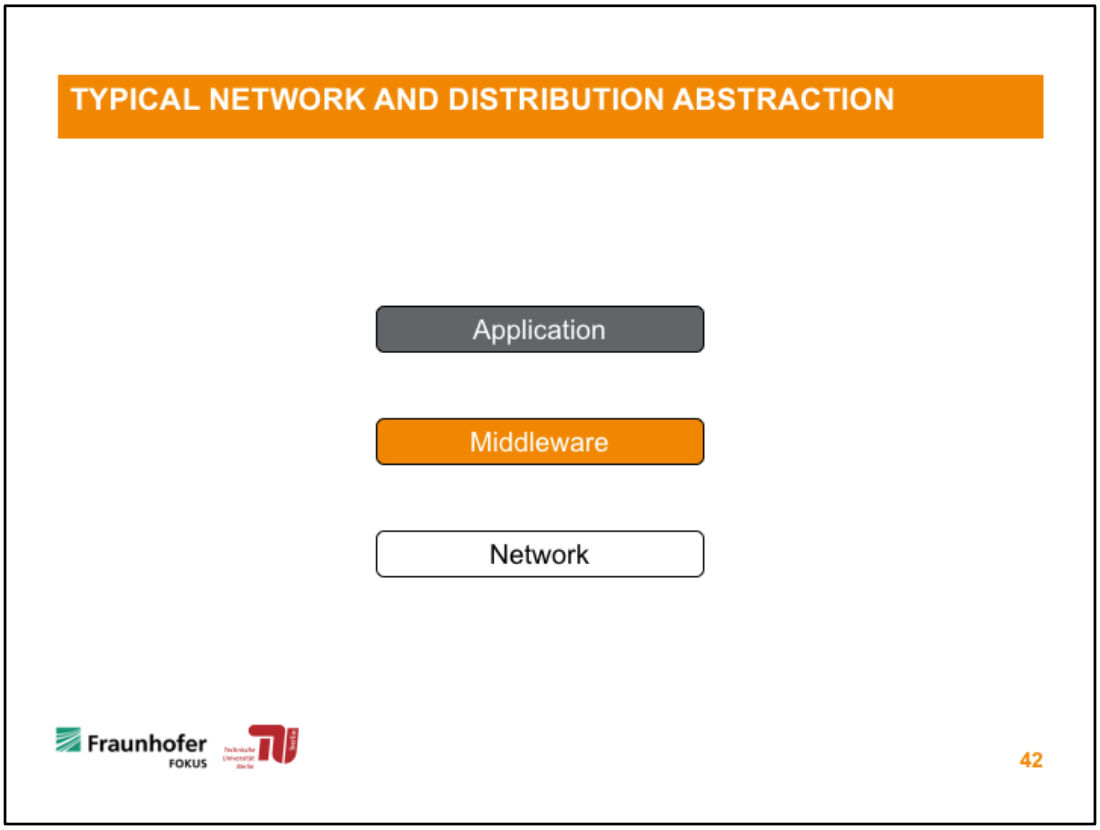
15 minutes

DIFFERENT LEVELS OF INTEROPERABILITY



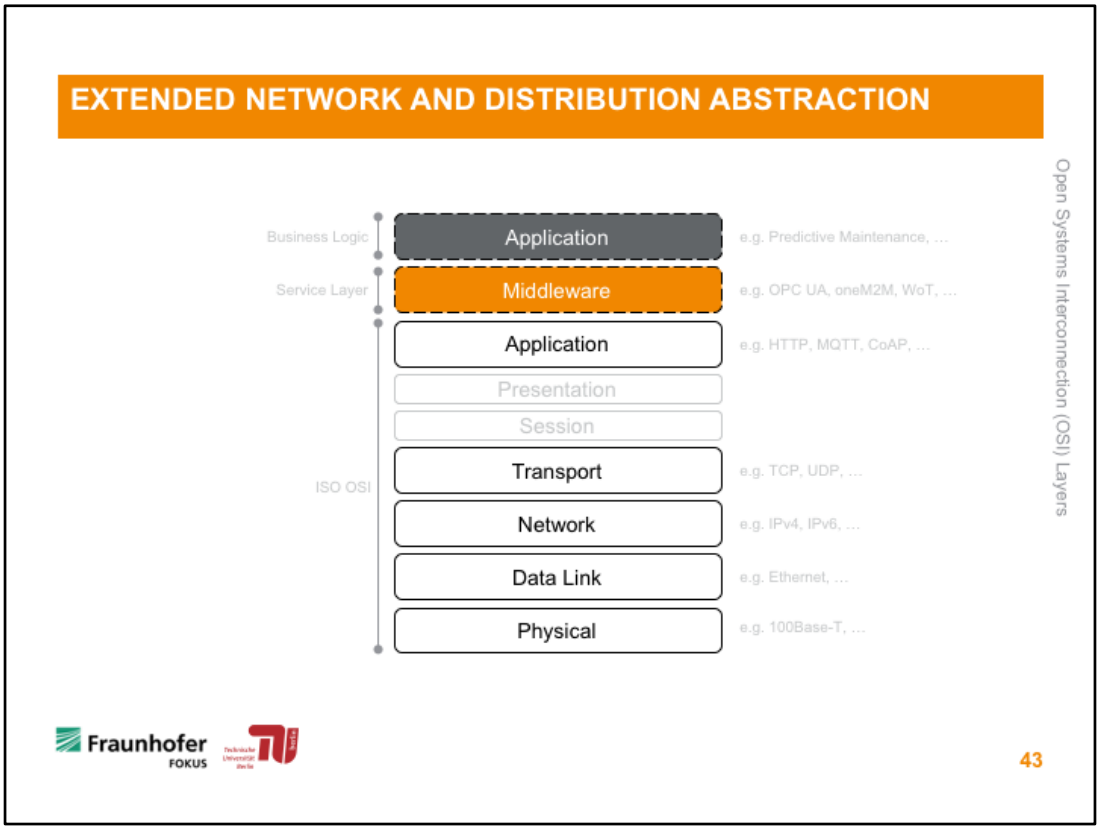
ETSI White Paper: Achieving technical interop.

Fraunhofer FOKUS



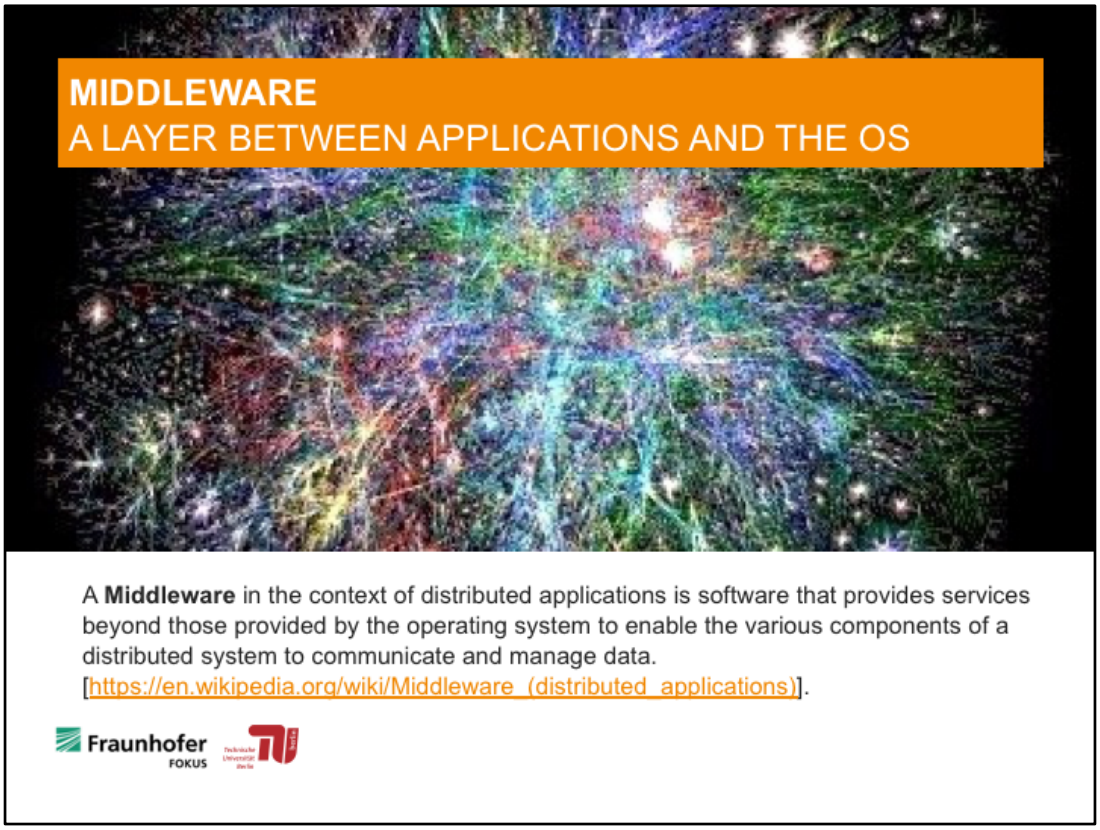
Reminder:

- Typically, when looking at distributed systems, we distinguish between three different layers.
- Network: the physical connections and protocols. That is what we'll focus on today and the next lectures.
- Middleware: an interoperability layer between the actual application and the network. We'll talk about the middleware later.
- Application: the actual business logic (and user interface).
- We'll extend this stack lecture after lecture.

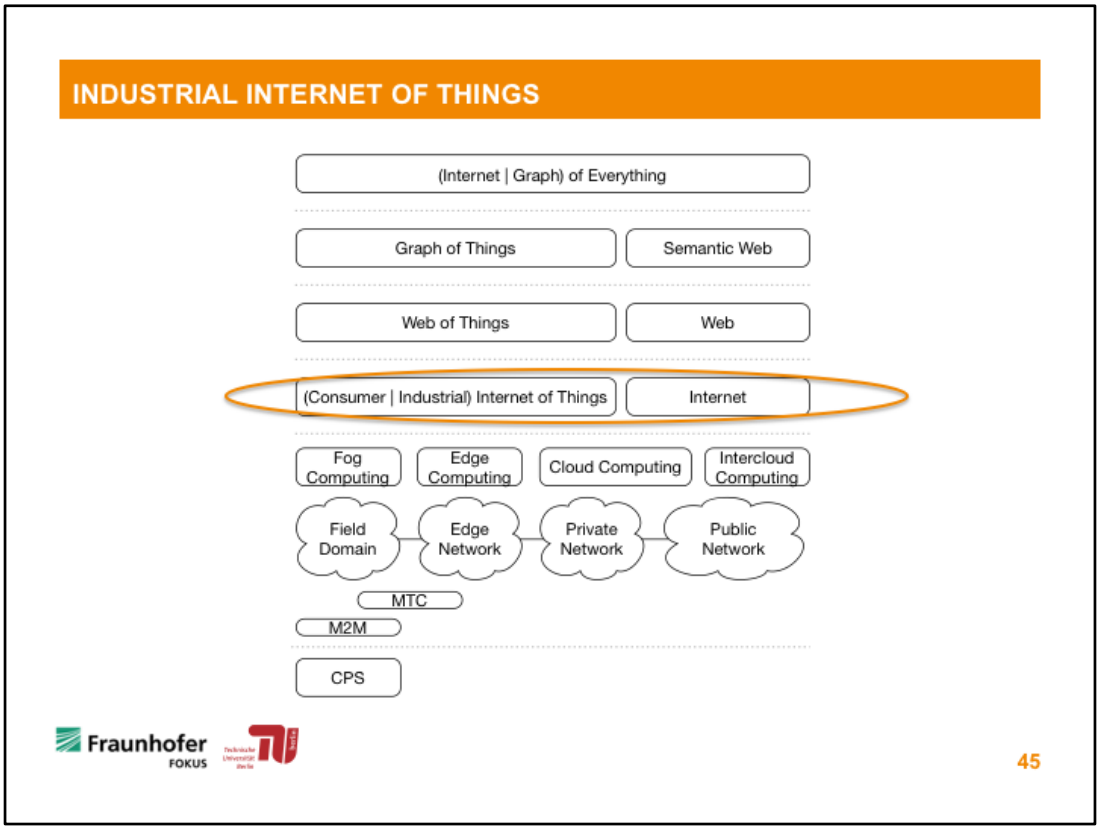


Reminder:

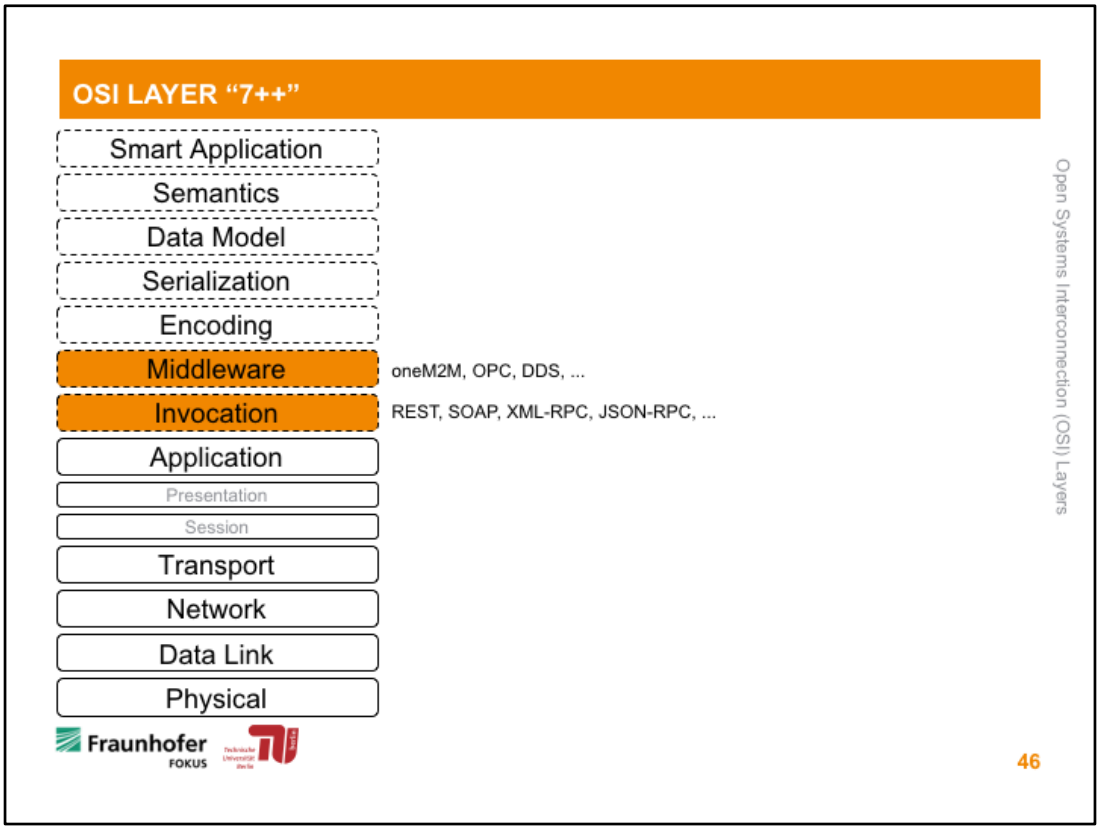
- Starting with looking closer at the ISO OSI stack.
- Note that this stack is about the network only.
- A common misconception is that the ISO OSI „Application“ layer is about the application. It‘s actually the interface for applications to the network.



- Not clearly defined
- It simplifies complex distributed applications.
- Often enables interoperability
- OS = Operating System



- Internet of Things → communication using Internet technologies



- **Note:** there are no OSI layers above layer 7 – "layer 8" sometimes denoted as the "political" layer
- **Invocation**: defines how information can be retrieved or modified (see Remote Procedure Calls (RPC))
- **Middleware:** provides a set of communication services for distributed applications

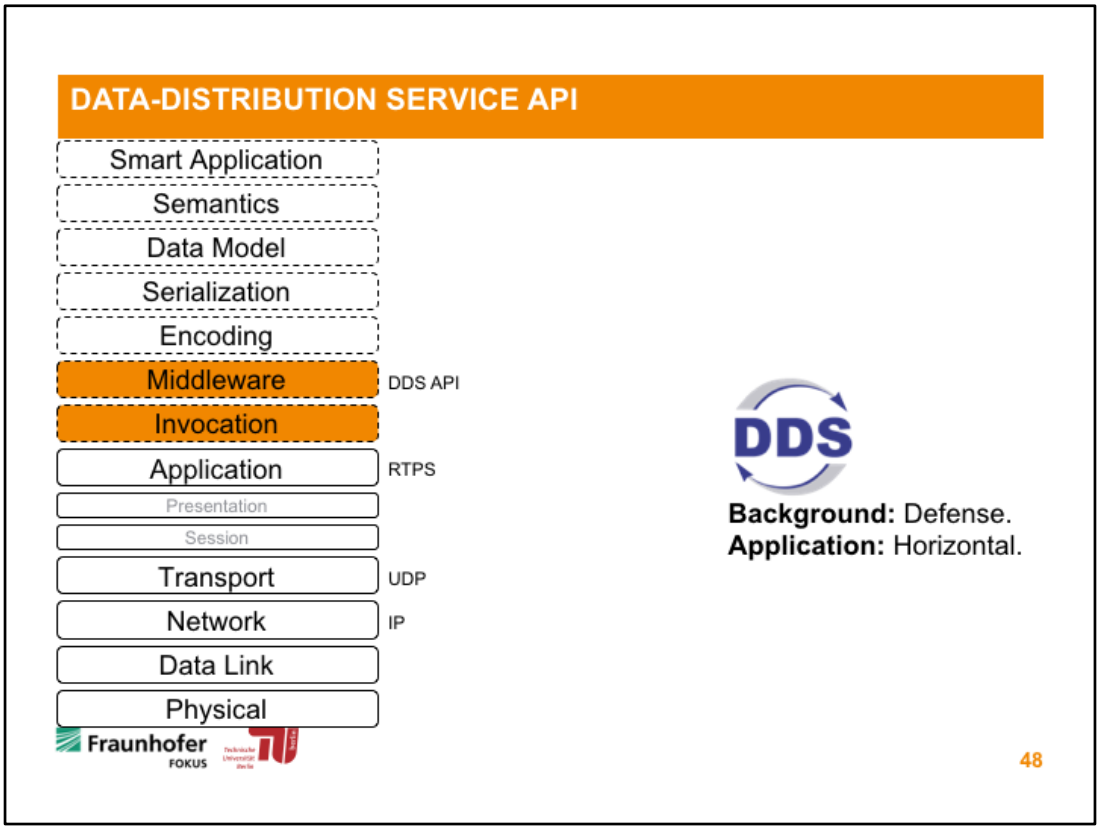
OPEN PLATFORM COMMUNICATIONS UNIFIED ARCHITECTURE

Smart Application	
Semantics	
Data Model	
Serialization	
Encoding	
Middleware	OPC UA API
Invocation	Binary, XML RPC
Application	HTTP, AMQP, Proprietary
Presentation	
Session	
Transport	TCP
Network	IP
Data Link	
Physical	

Background: Automation.
Application: Vertical.

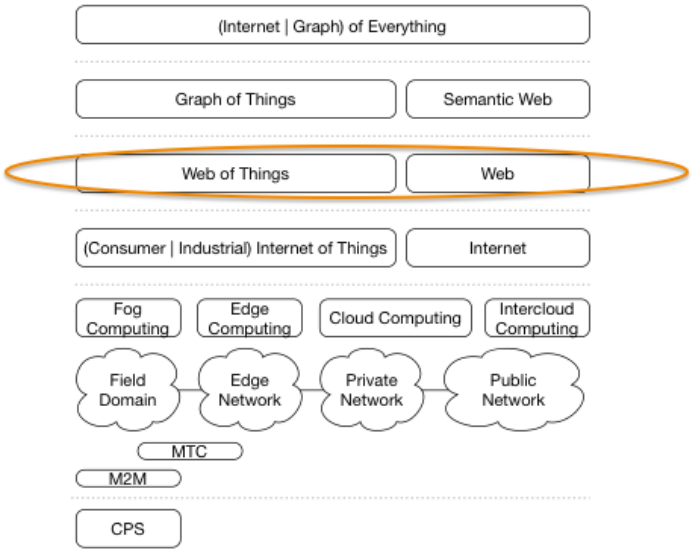
Fraunhofer FOKUS

47



- Real-time Publish-Subscribe Wire Protocol

WEB OF THINGS



Fraunhofer FOKUS

49

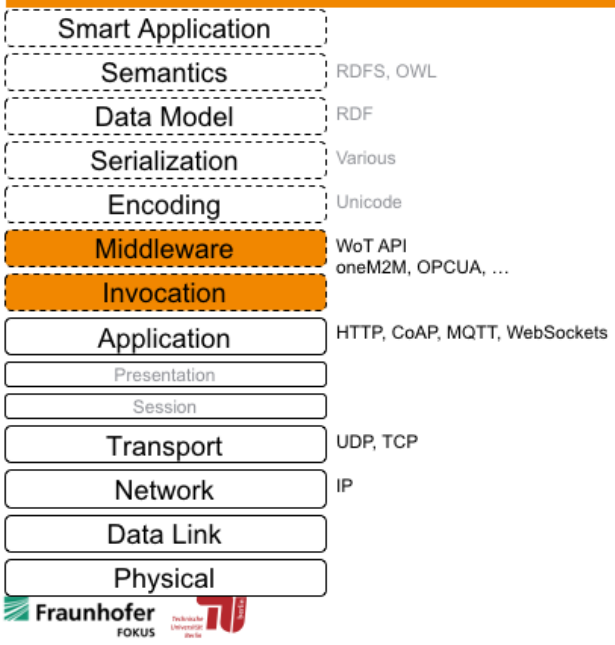
- Web of Things → communication using Web technologies

ONEM2M

Smart Application	
Semantics	
Data Model	
Serialization	
Encoding	
Middleware	oneM2M API
Invocation	RESTful
Application	HTTP, CoAP, MQTT, WebSockets
Presentation	
Session	
Transport	UDP, TCP
Network	IP
Data Link	
Physical	

Background: Telecom.
Application: Horizontal.

W3C WEB OF THINGS (WOT)



Background: lack of interoperability across platforms

Fraunhofer FOKUS

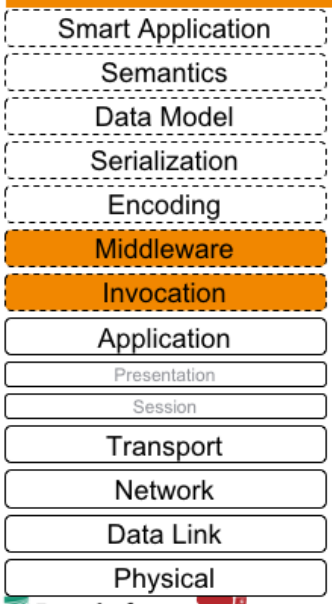
MIDDLEWARE OVERVIEW

Questions?

OPC

15 minutes

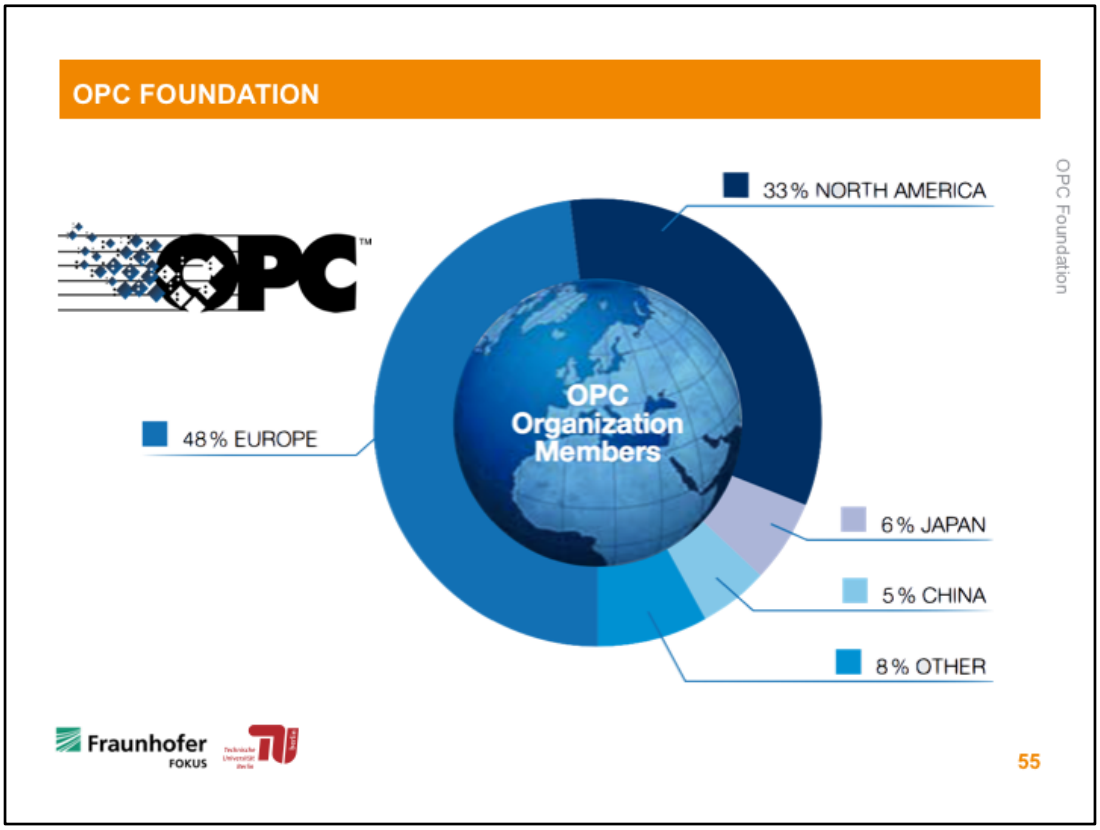
OPEN PLATFORM COMMUNICATIONS UNIFIED ARCHITECTURE



Background: Automation.
Application: Vertical.

Fraunhofer FOKUS

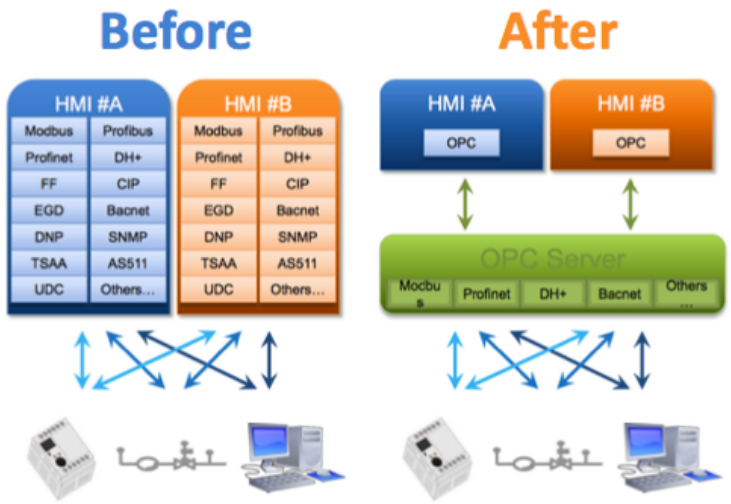
54



- OPC UA members mostly in Europe and US/Canada

- ~ 500 members from small system integrators to the world's largest automation and industrial suppliers.
- Coordinates the further development of the OPC standard in collaboration with users, manufacturers and researchers.
    - Development and maintenance of specifications
    - Certification and compliance tests of implementations
    - Cooperation with other standards organizations

MOTIVATION

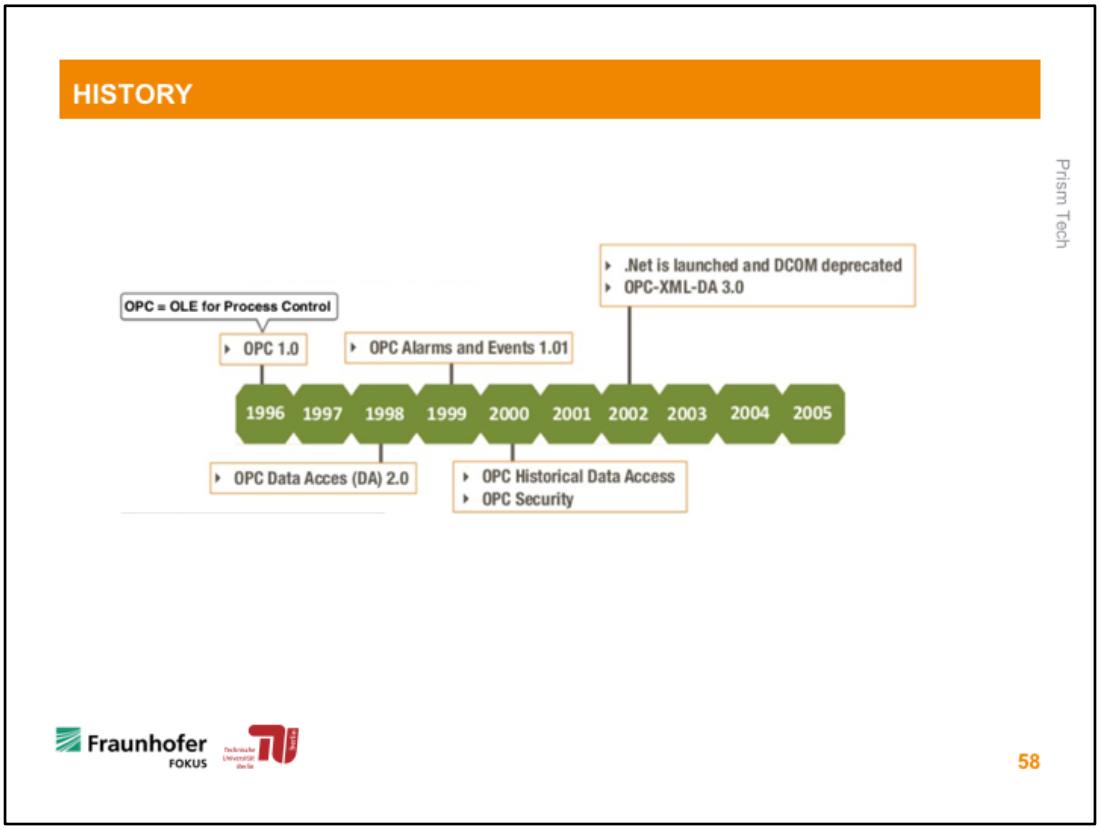


Prismtech, based on OPC Foundation Slides

Fraunhofer FOKUS

57

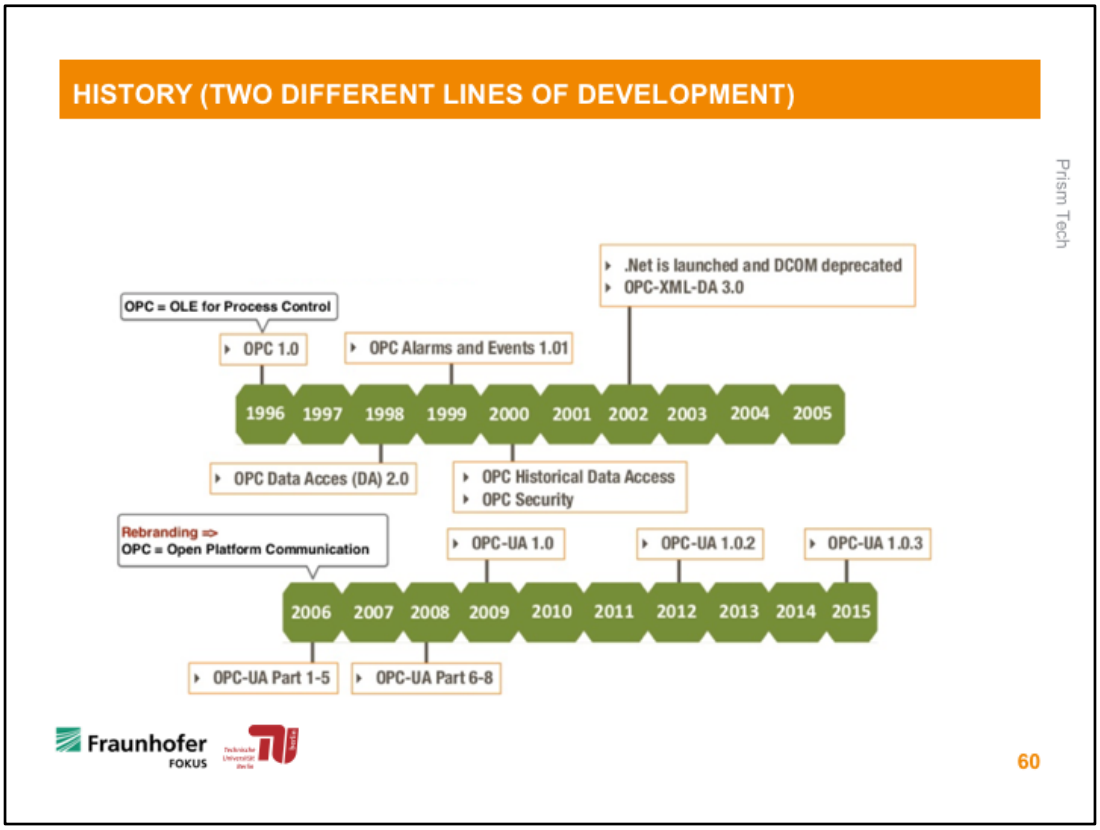
- Remember: Field bus systems, SCADA network
- Before, the HMIs had to handle all the different protocols of the field devices
- The addition of an OPC server abstracts the different protocols away, HMIs need to implement OPC UA only



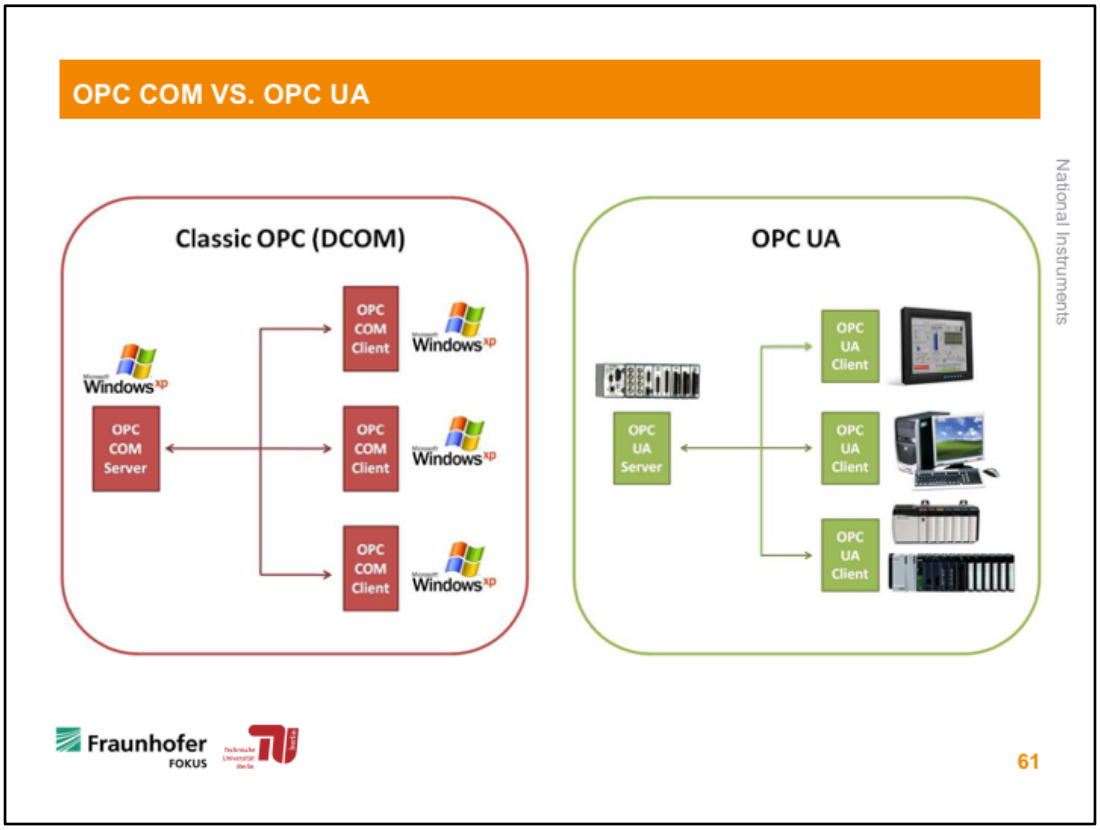
- **OLE** = Object Linking and Embedding (Microsoft technology to embed and link to documents)
- OPC is a standard interface to communicate between numerous data sources, including devices on a factory floor, laboratory equipment, test system fixtures, and databases.
- It utilizes the Microsoft-based COM/DCOM technology to provide standard specifications for data access (DA), historical data access (HDA), and alarms and events (A&E).
- Several limitations because of this reliance on the Microsoft Windows platform, in the form of security issues and platform dependency.

OPC UA

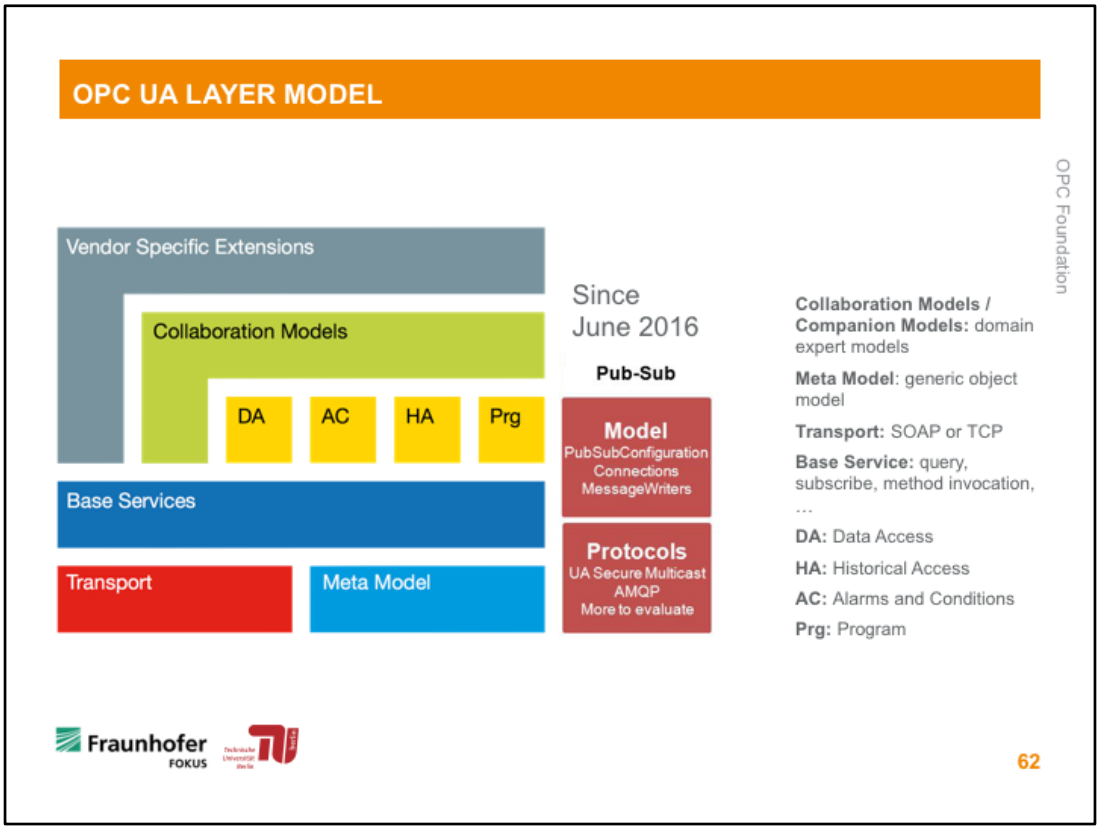
15 minutes



- In 2004 OPC Foundation decided to propose a new architecture:
  - Microsoft has deprecated COM
  - OPC Vendors want a single set of services to expose the OPC data models (DA, A&E, HDA ...)
  - OPC Vendors want to implement OPC on non-Microsoft operating systems, including embedded devices
  - Other collaborating organizations need a reliable, efficient way to move higher level structured data
- The first parts of this new "Unified Architecture" were released in June'2006.



- OPC COM was dependent on windows OS on the devices
- OPC UA supporting a broader variety of different devices
- OPC Unified Architecture is a set of specifications applicable to manufacturing software in application areas such as Field Devices, Control Systems, Manufacturing Execution Systems and Enterprise Resource Planning Systems.
- These systems are intended to exchange information and to use command and control for industrial processes.
- OPC Unified Architecture defines a common infrastructure model to facilitate this information exchange.

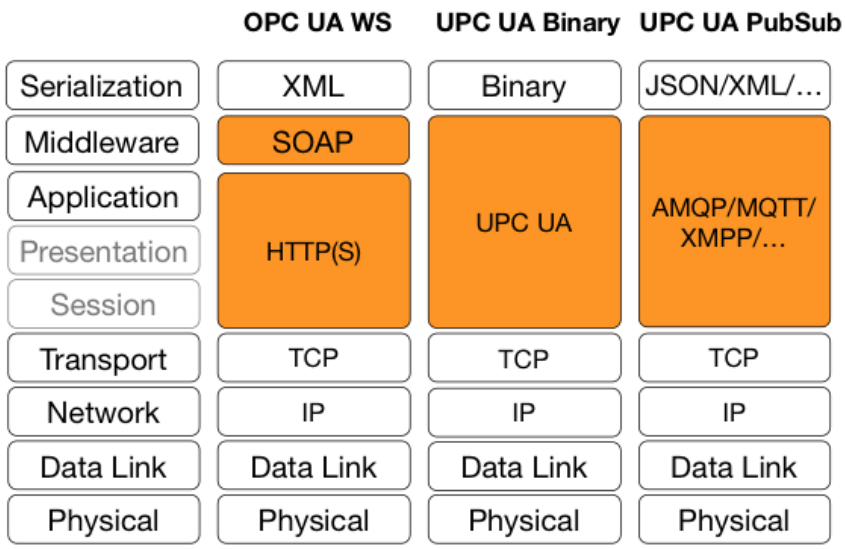


- OPC Unified Architecture specifies the following:
  - The information model to represent structure, behavior, “semantics”, and infrastructure of the underlying real-time system.
  - The message model to interact between applications.
  - The communication models to transfer data between end-points.
  - The conformance model to guarantee interoperability between systems.
  - The security model to guarantee cyber security addressing client/server authorization, data consistency, encryption.

OPC UA SPECIFICATION PARTS OVERVIEW

#	Topic
1	Concepts
2	Security Model
3	Address Space Model
4	Services
5	Information Model
6	Mappings
7	Profiles
8	Data Access
9	Alarms and Conditions
10	Programs
11	Historical Access
12	Discovery
13	Aggregates
14	PubSub

THREE UPC UA COMMUNICATION TYPES



Fraunhofer FOKUS

64

OPERATION MODE 1: OPC CLIENT-SERVER CONCEPT

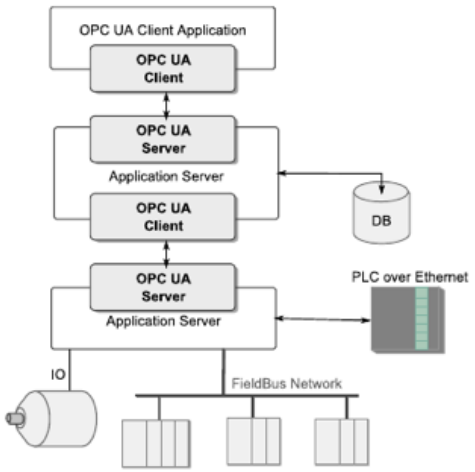


Figure 9: OPC UA Stacked Architecture

Source: http://themanufacturingconnection.com/2016/06/opc-ua-publish-subscribe/

Fraunhofer FOKUS

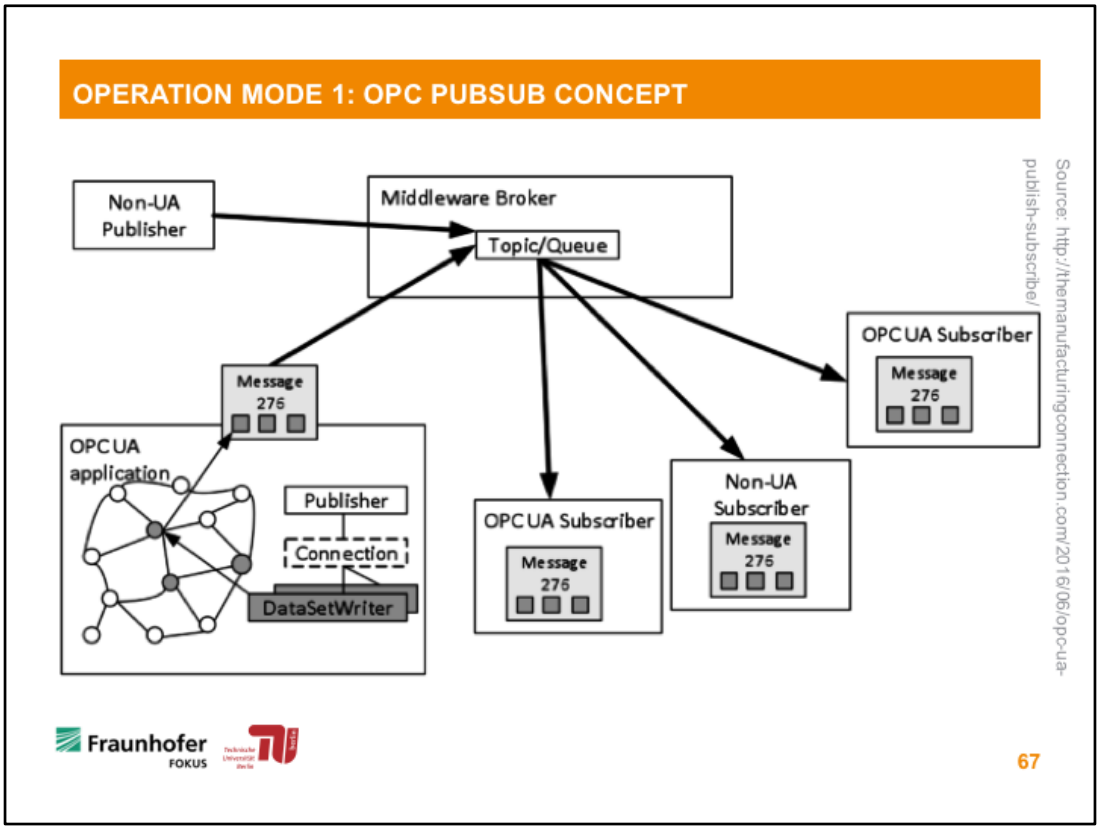
66

• OPC implements a typical client – server architecture, where the client is responsible for management of relations with the server e.g.:

- •Server discovery and localization.
- •Establishing the connection.
- •Generation of requests for selected data and provision of services to the server.
- •Disconnection.

•To establish connection with the selected server, the client must know how to address it, namely the client must know two important details:

- •The network address of the workstation (usually IP address) hosting the server.
- •A unique identifier of the server on the host.



- OPC UA Spec Part 14 specifies PubSub a loosely coupled message protocol that can be used with multiple encodings (e.g. JSON, UA Binary or XML) and multiple transports (e.g. AMQP, MQTT, XMPP et. al.).