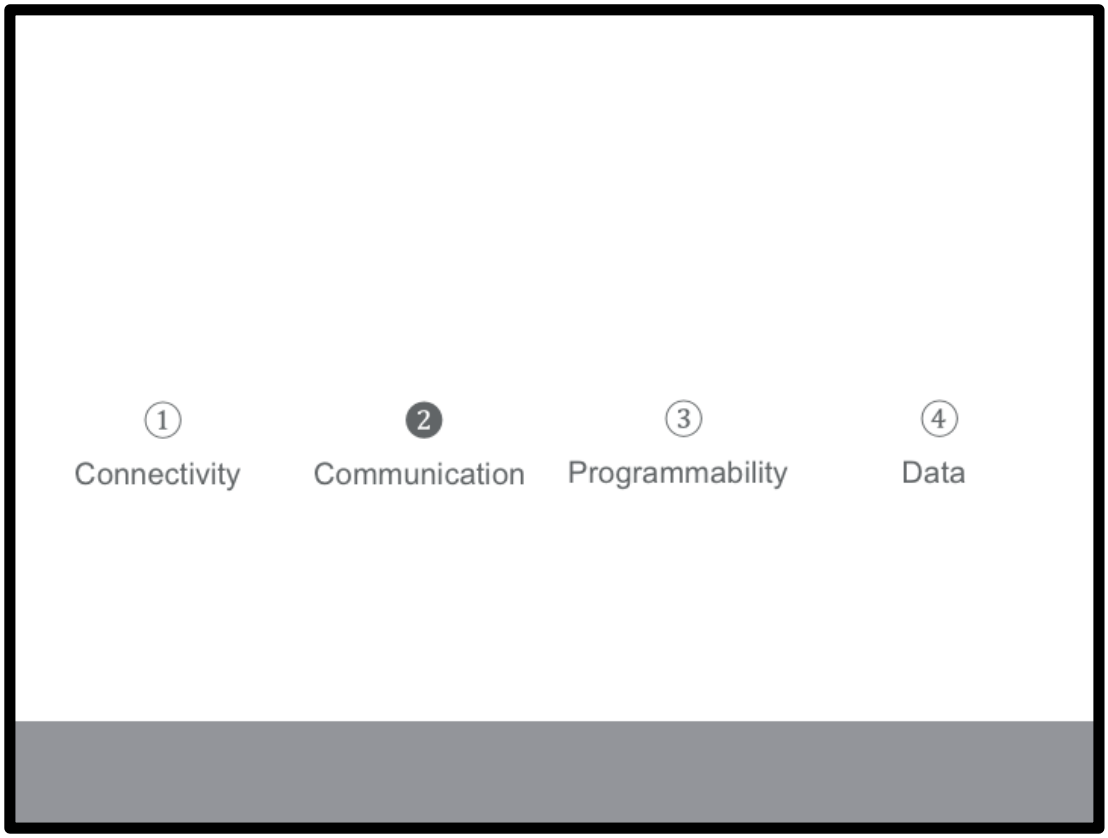
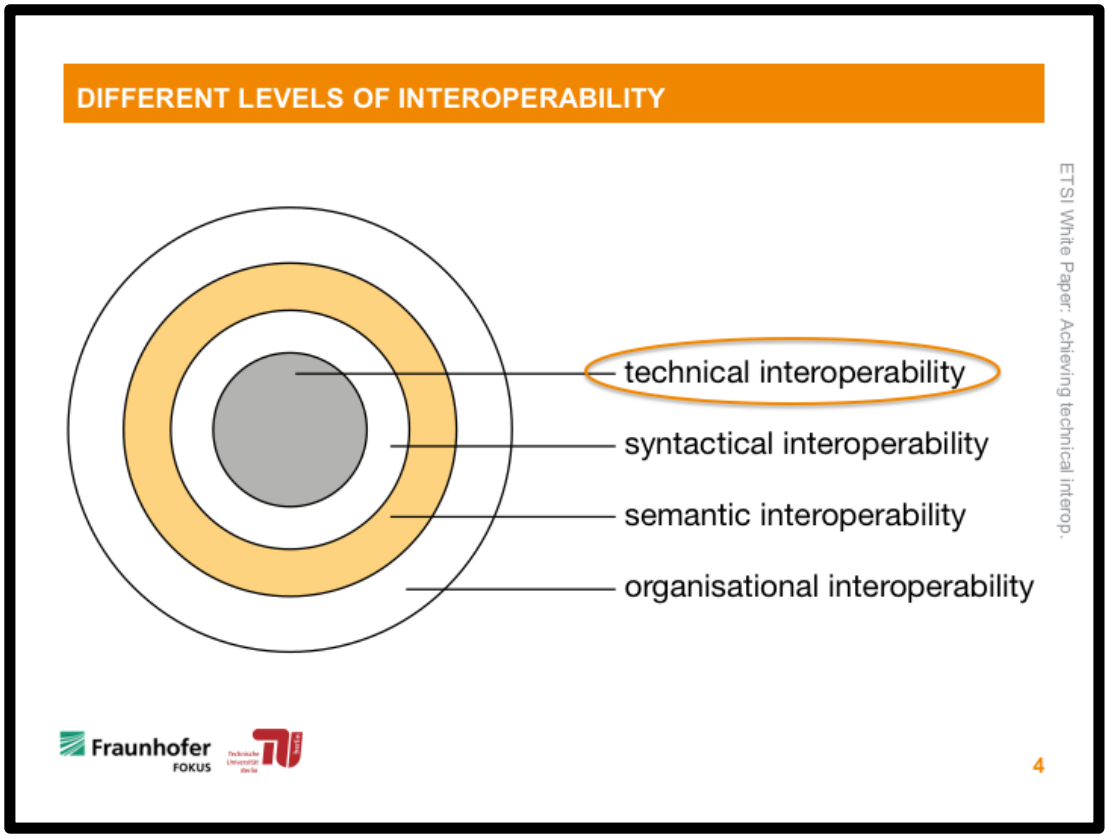

INDUSTRIAL INTERNET OF THINGS (IIOT)
PART 4: COMMUNICATION (2)
AV Lecture in Summer Term 2018
Dr.-Ing. Alexander Willner, Ronald Steinke
Fraunhofer FOKUS
We'll start at 14:15pm

THE LAST LECTURE

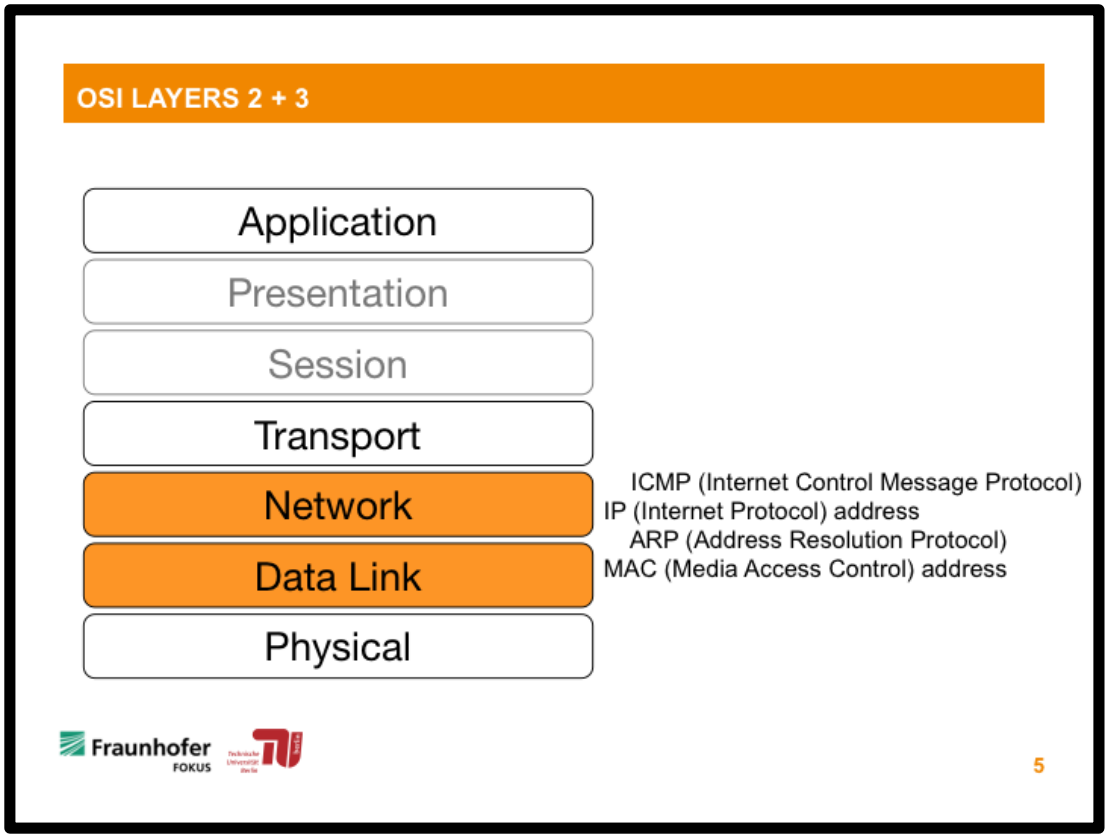
8 minutes



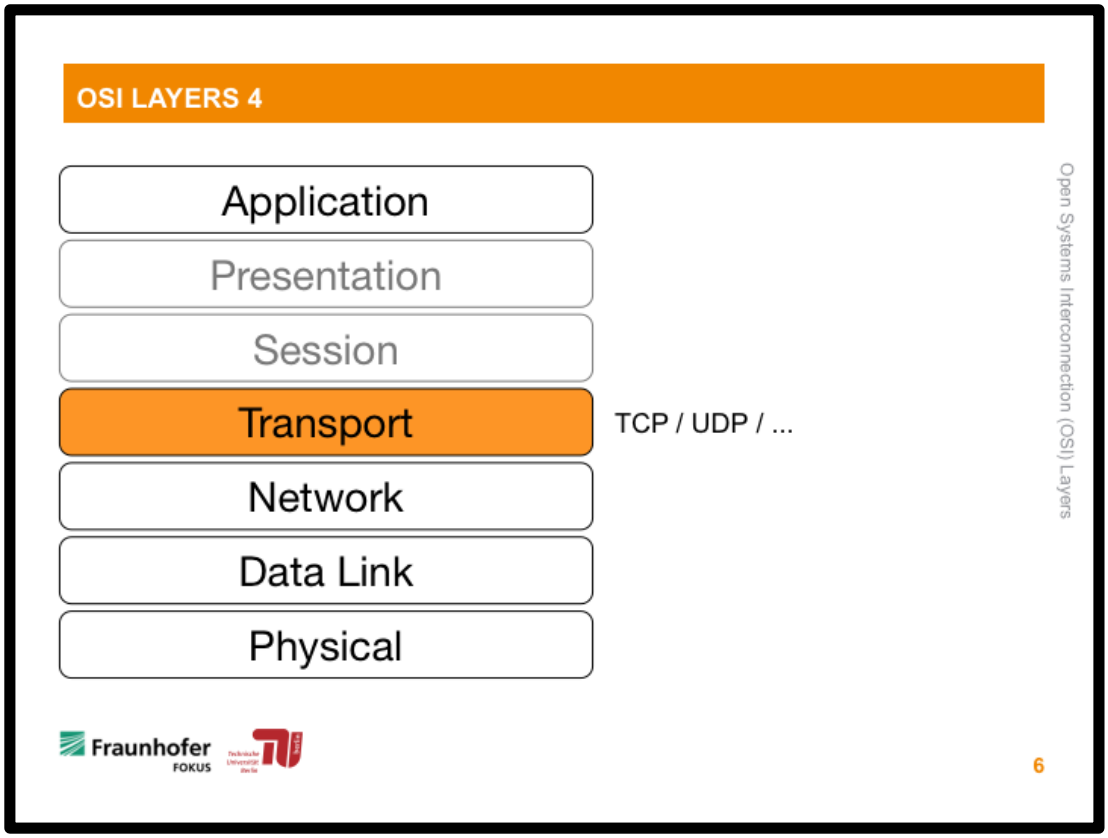
- This lecture is divided into 4 different areas
- Now, we focus on the second one



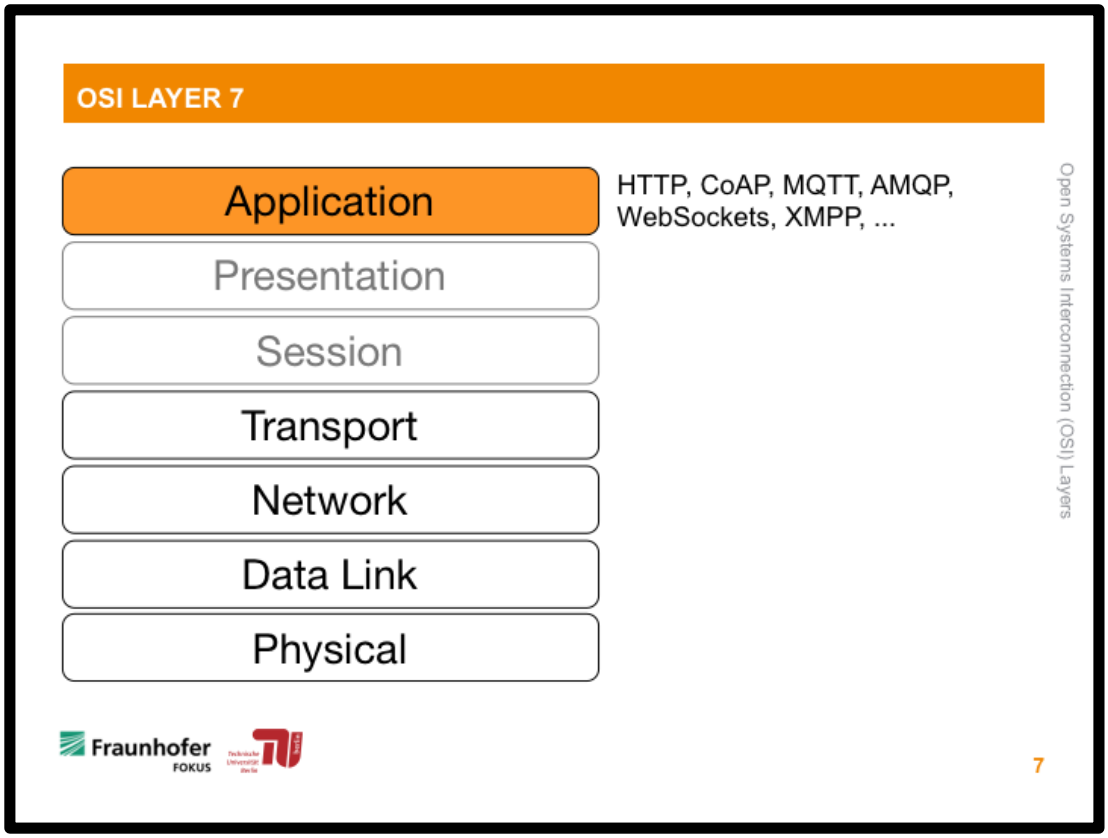
- We still focus on technical interoperability
- However, we start to touch the next two levels as well



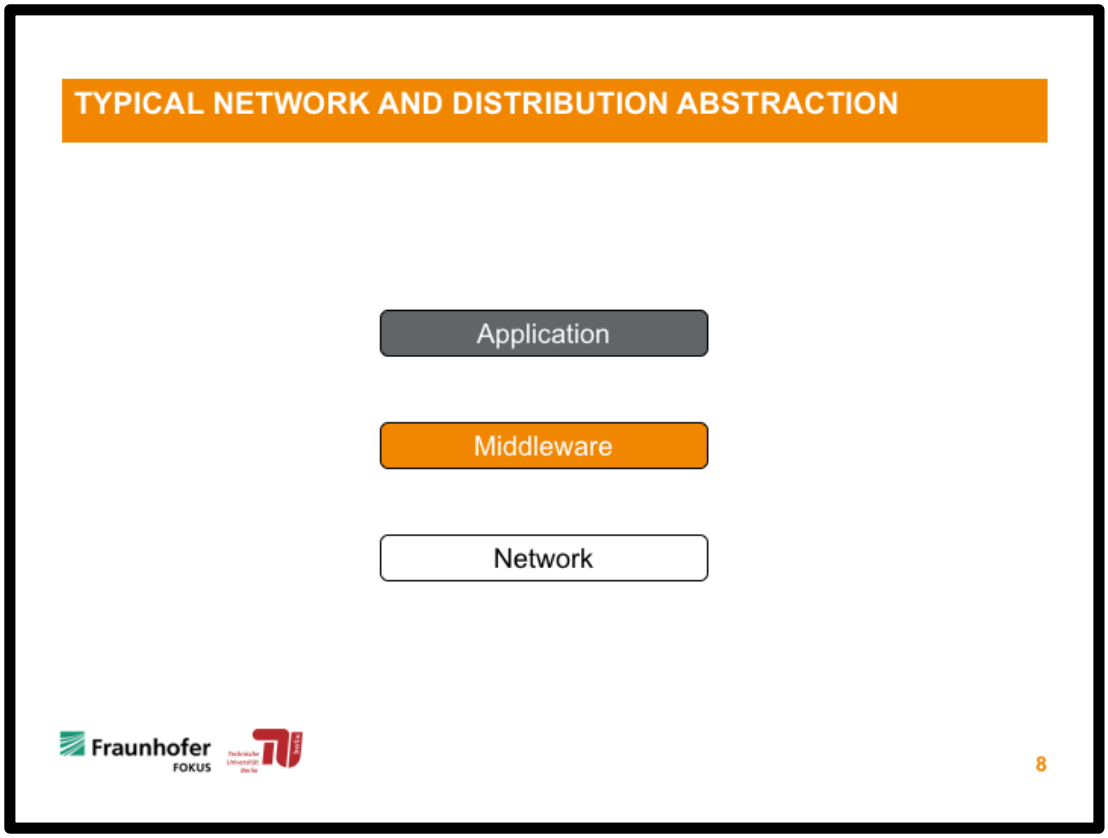
- MAC (Media Access Control) address in IEEE 802: 48bit.
- ARP (Address Resolution Protocol): translating between MAC and IP
- IP (Internet Protocol) address (v4/32 bit, v6/128bit)
- Internet Control Message Protocol (ICMP): e.g. diagnose reachability of nodes



- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)
- Different metrics to select one over the other

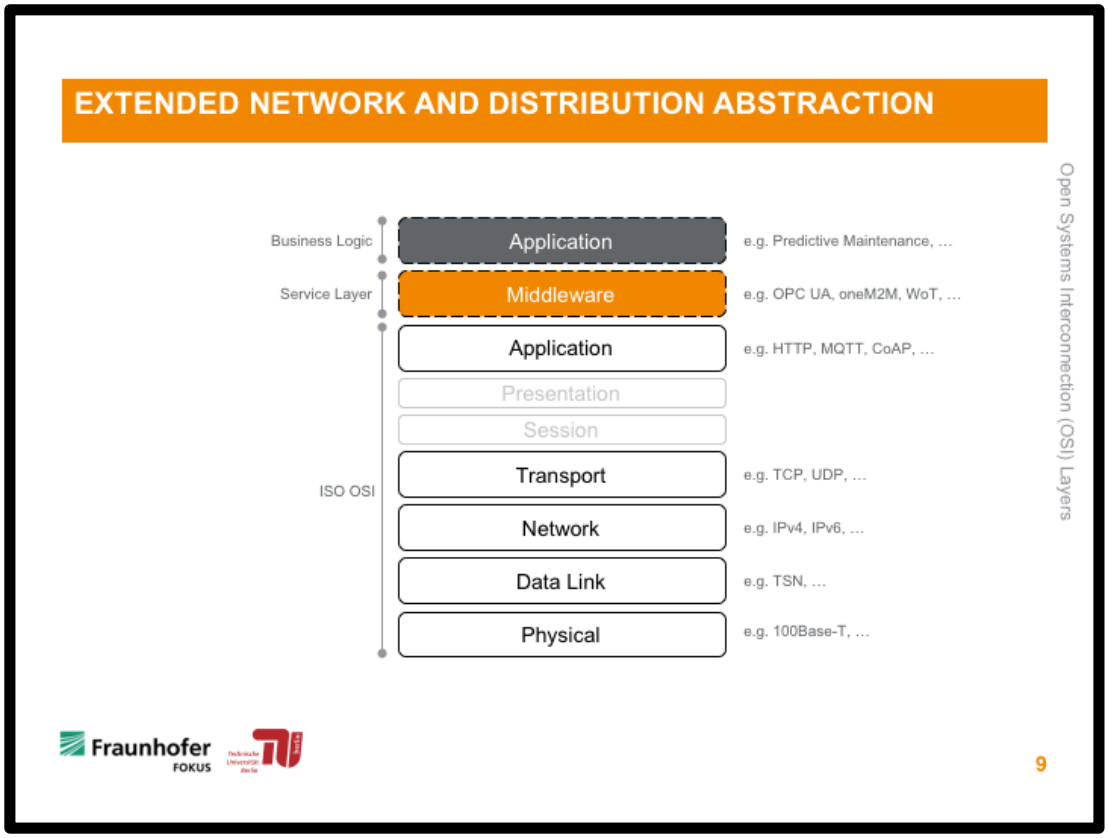


- Protocols to provide functionalities for specific applications
- Examples include file transfer, remote login, mail exchange, name resolution, …



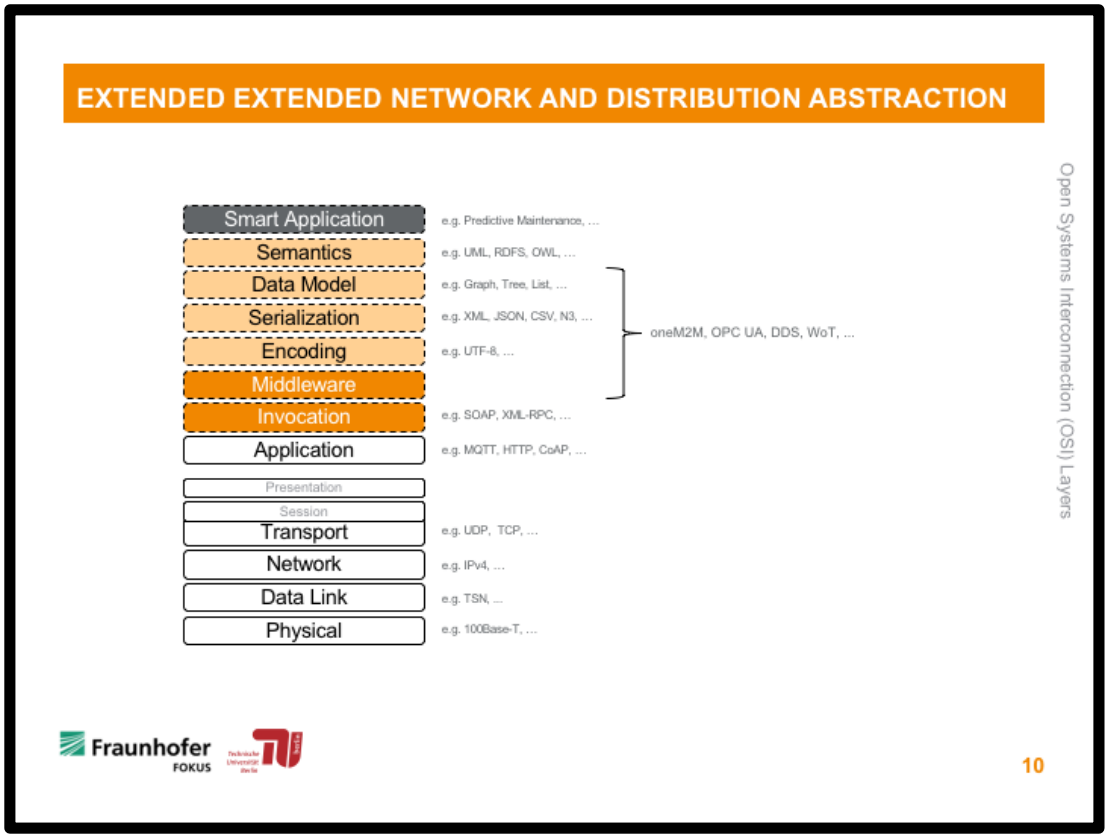
Reminder:

- 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.

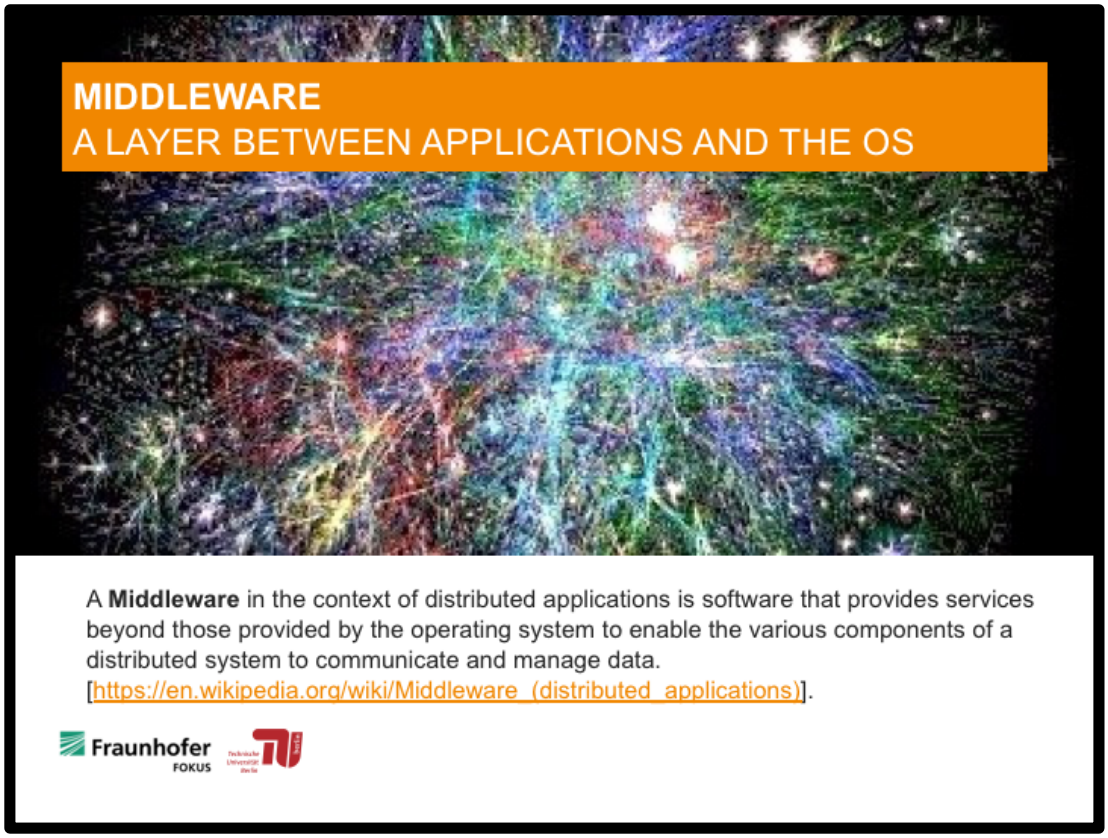


Reminder:

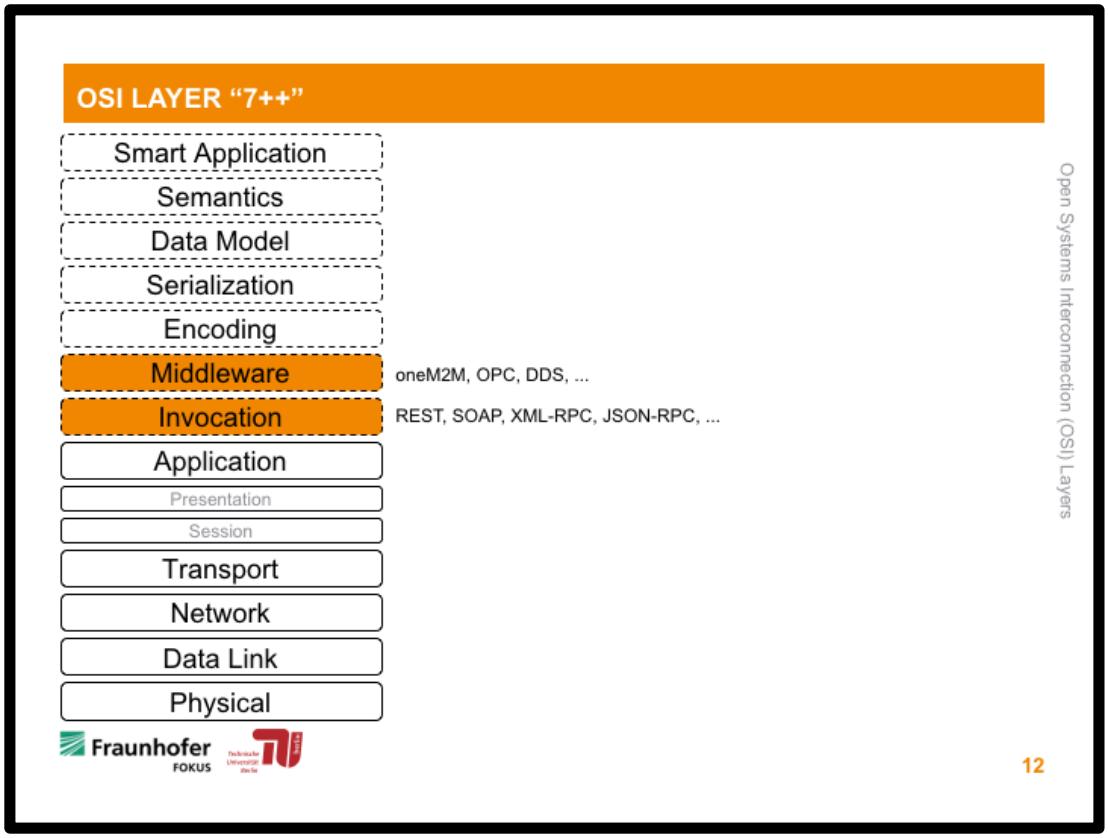
- Starting with looking closer at the ISO OSI stack.
- Note that this stack is about the network only.
- A common misconception is that the ISO OSI „Application“ layer is about the application. It‘s actually the interface for applications to the network.



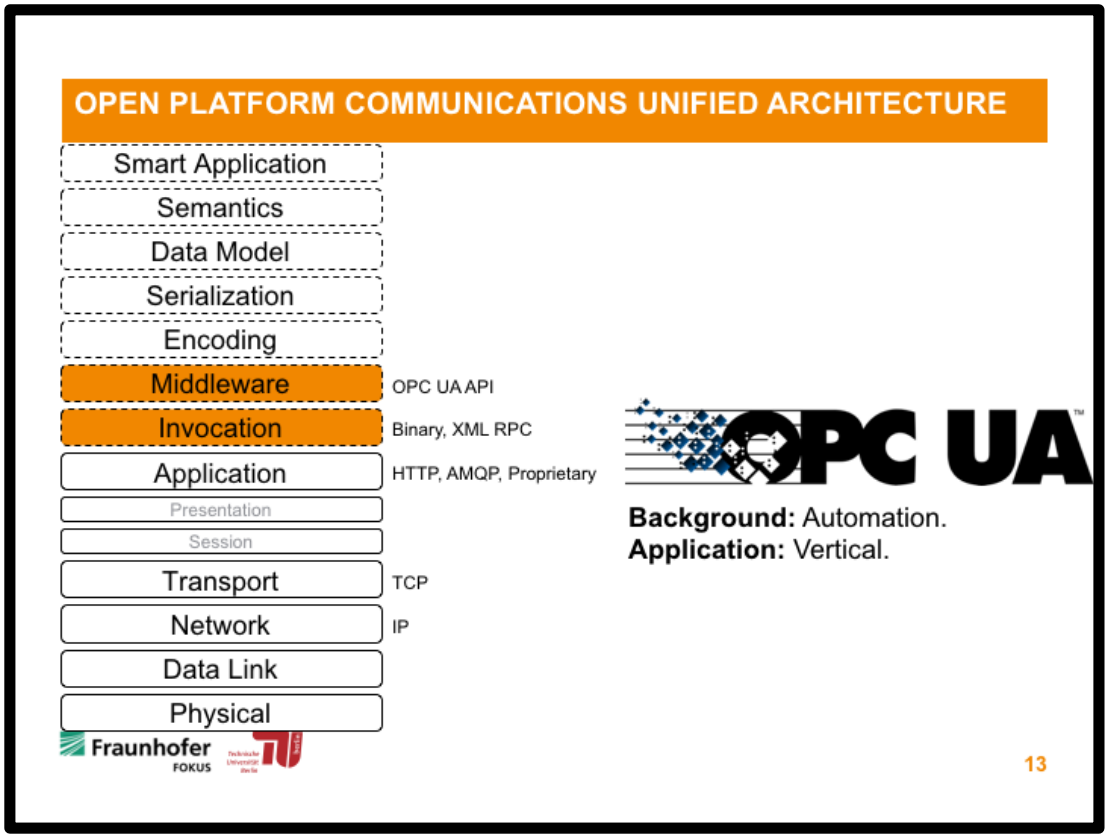
- **Note:** there are no OSI layers above layer 7 – “layer 8” sometimes denoted as the “political” layer
- **Invocation**: defines how information can be retrieved or modified (see Remote Procedure Calls (RPC))
- **Middleware:** provides a set of communication services for distributed applications



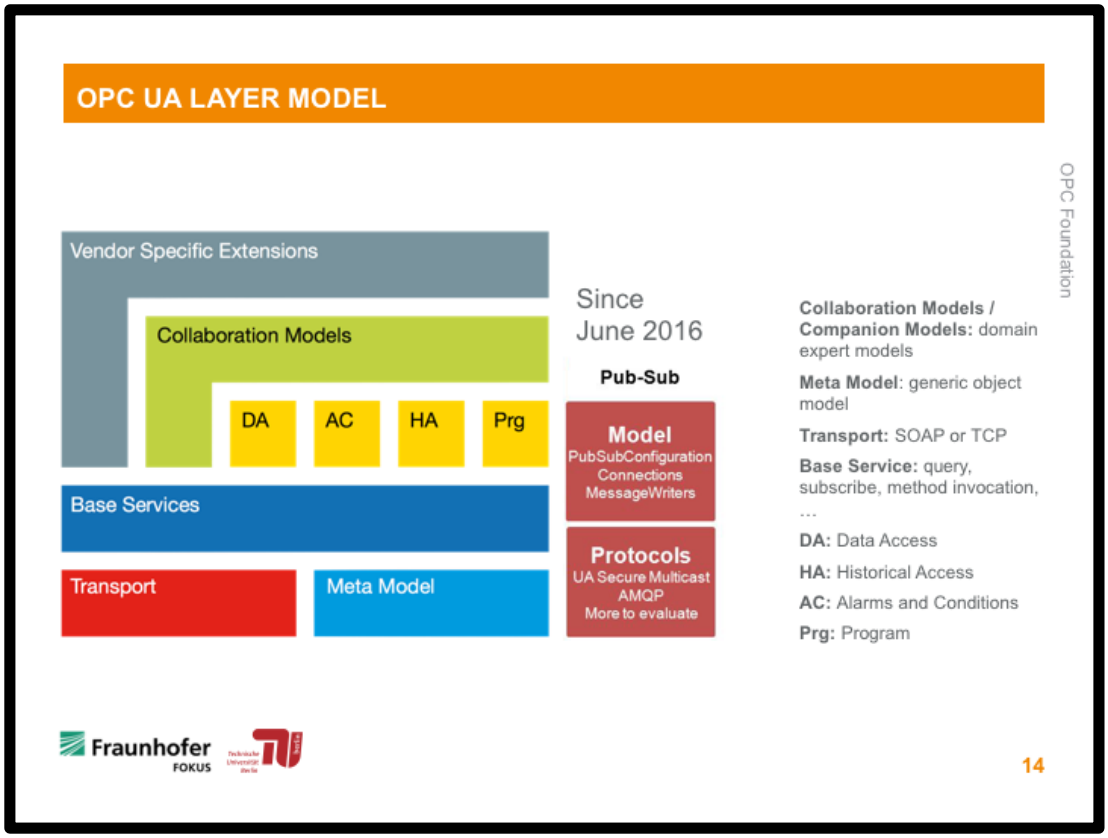
- Not clearly defined
- It simplifies complex distributed applications.
- Often enables interoperability
- OS = Operating System



- **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



- We started to talk about OPC UA



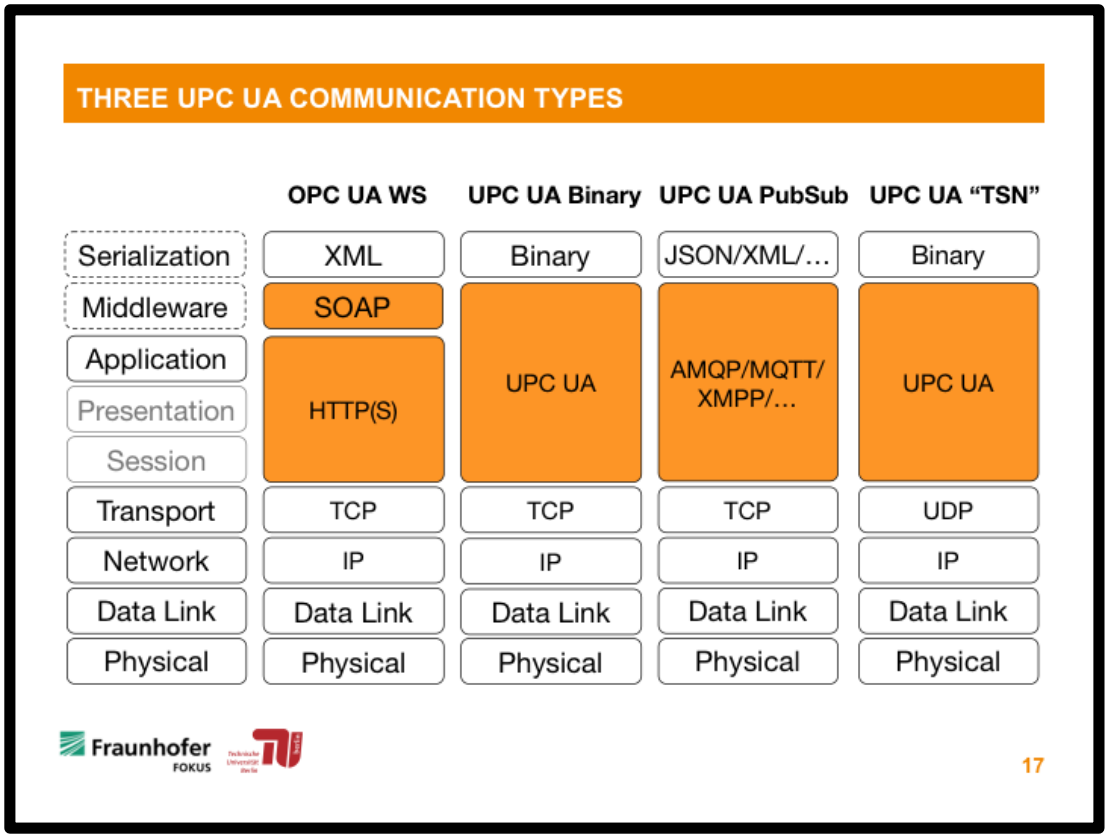
- OPC Unified Architecture specifies the following:
  - The information model to represent structure, behavior, “semantics”, and infrastructure of the underlying real-time system.
  - The message model to interact between applications.
  - The communication models to transfer data between end-points.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

THE LAST LECTURE

Questions?

OPC UA (2)

8 minutes



- Different modes
- XML over SOAP for interoperability
- Binary for efficient communication (less overhead for smaller resources)
- PubSub for more scalable communication
- Another: UPC UA with RAW UDP sockets for TSN

OPERATION MODE 1: OPC CLIENT-SERVER CONCEPT

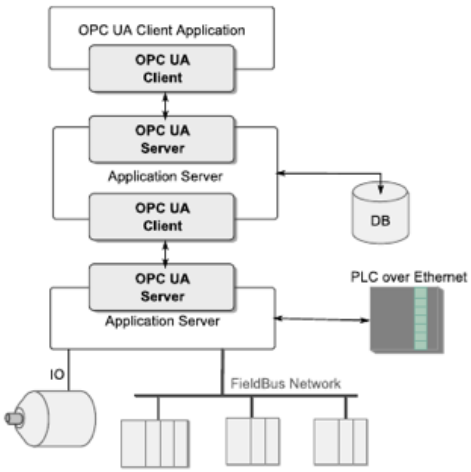


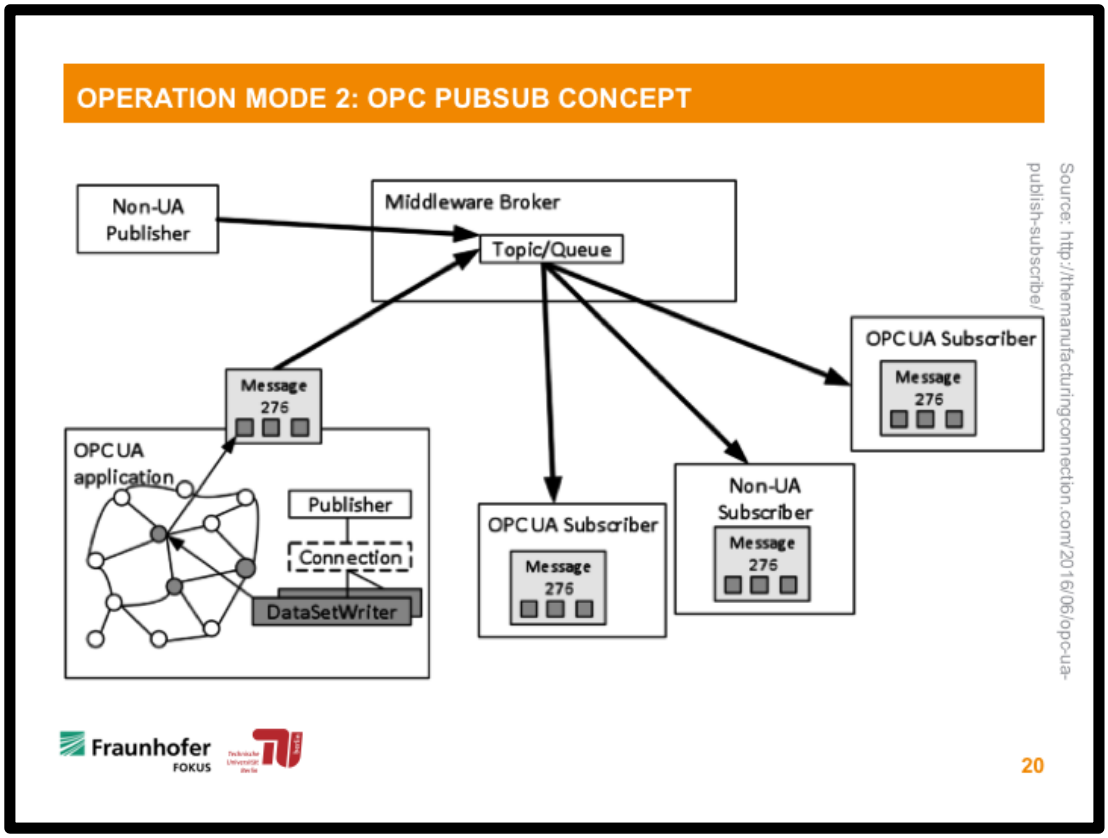
Figure 9: OPC UA Stacked Architecture

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

Fraunhofer FOKUS

19

- OPC implements a typical client – server architecture
- Client: responsible for management of relations with the server
    - Server discovery and localization
    - Establishing the connection
    - Generation of requests for selected data and provision of services to the server
    - Disconnection
- The client must know how to address it:
    - The network address of the workstation (usually IP address) hosting the server
    - A unique identifier of the server on the host.

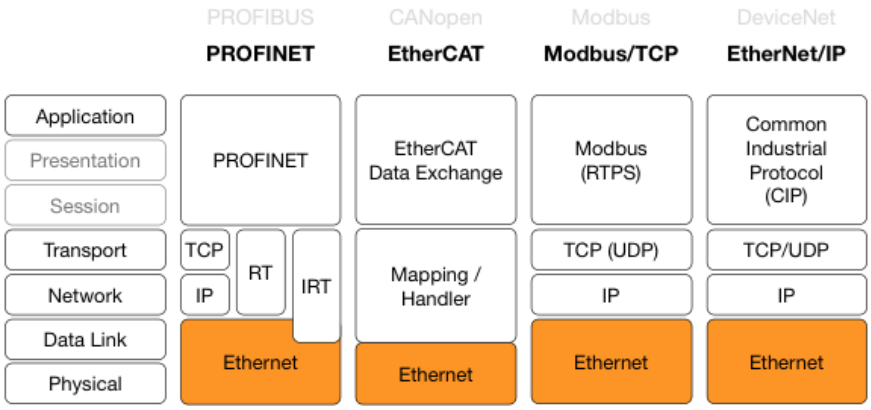


- OPC UA Spec Part 14 specifies PubSub
- A loosely coupled message protocol that can be used with
    - multiple encodings (JSON, UA Binary or XML)
    - and multiple transports (e.g. AMQP, MQTT, XMPP, …).

OPC UA INFORMATION MODEL

20 minutes

REMEMBER: FIELDBUS → INDUSTRIAL ETHERNET

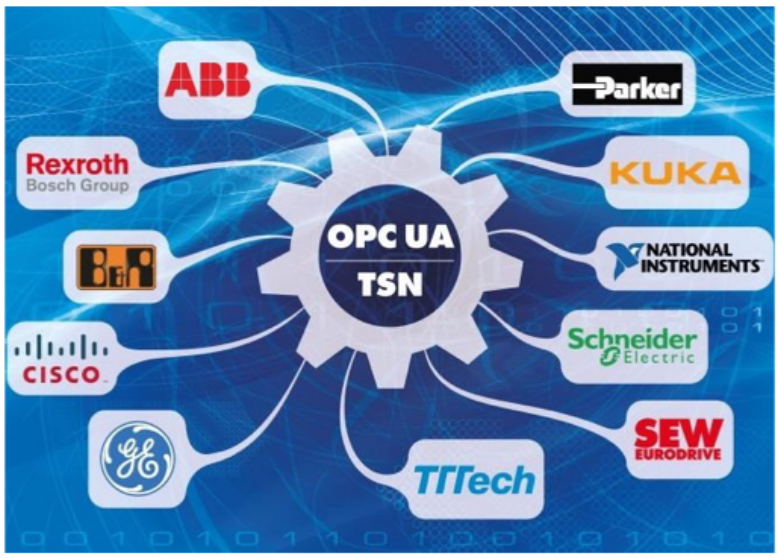


Fraunhofer FOKUS

23

- Same physical connectors
- Similar MAC protocols
- Depending on the real-time requirements, modification of the data link layer
- **Note:** Industrial Ethernet protocols also include layer 7 functionality!

OPC UA OVER TSN

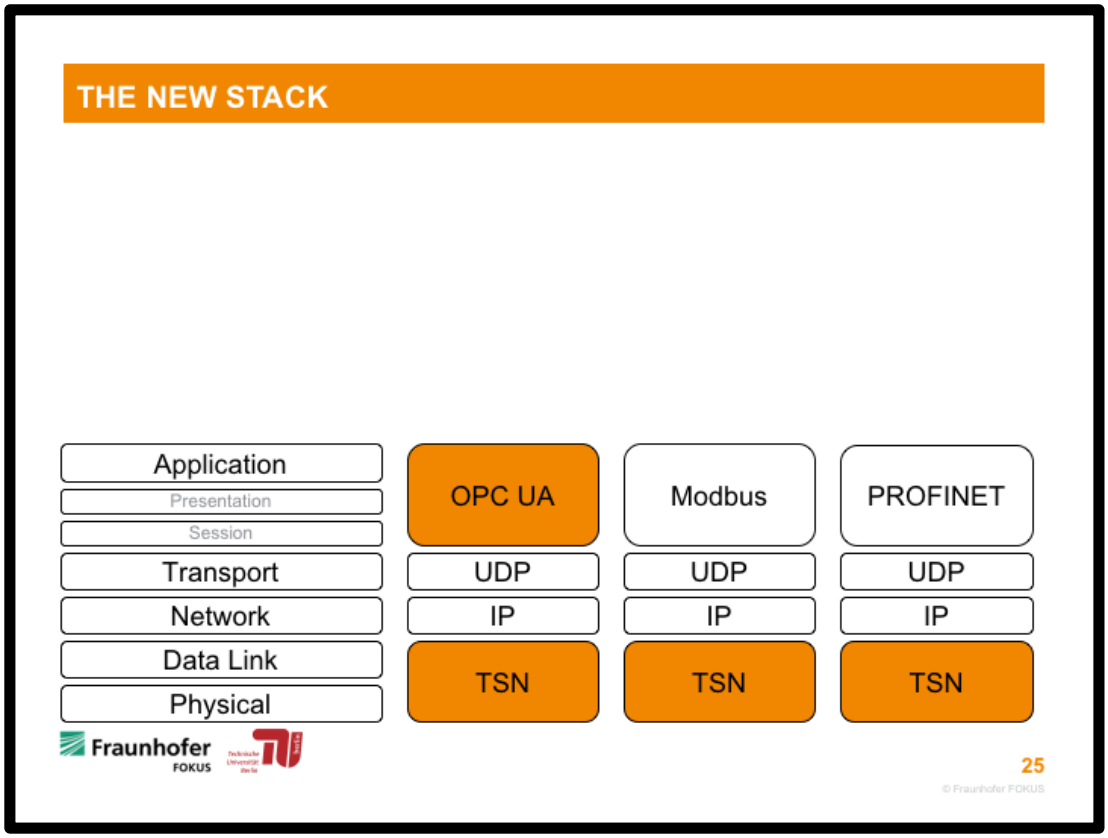


https://www.br-automation.com/en/technologies/opc-ua/tsn-and-pubsub/

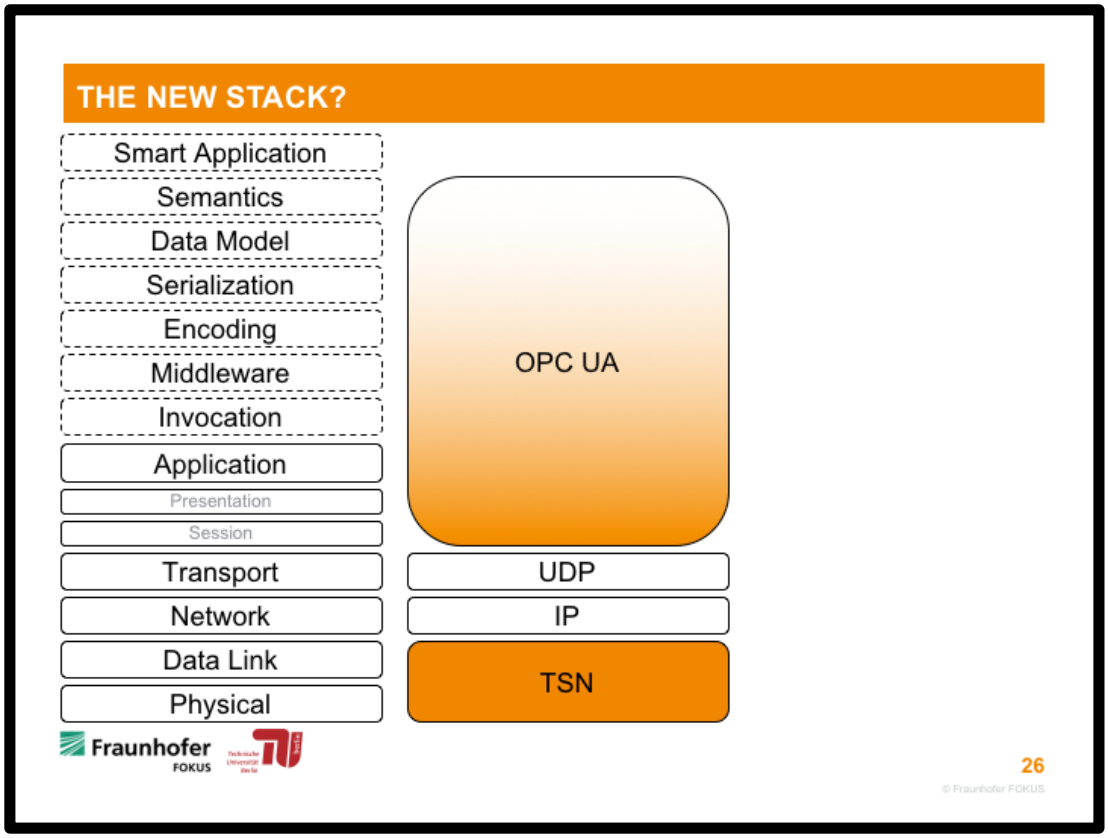
Fraunhofer FOKUS

24

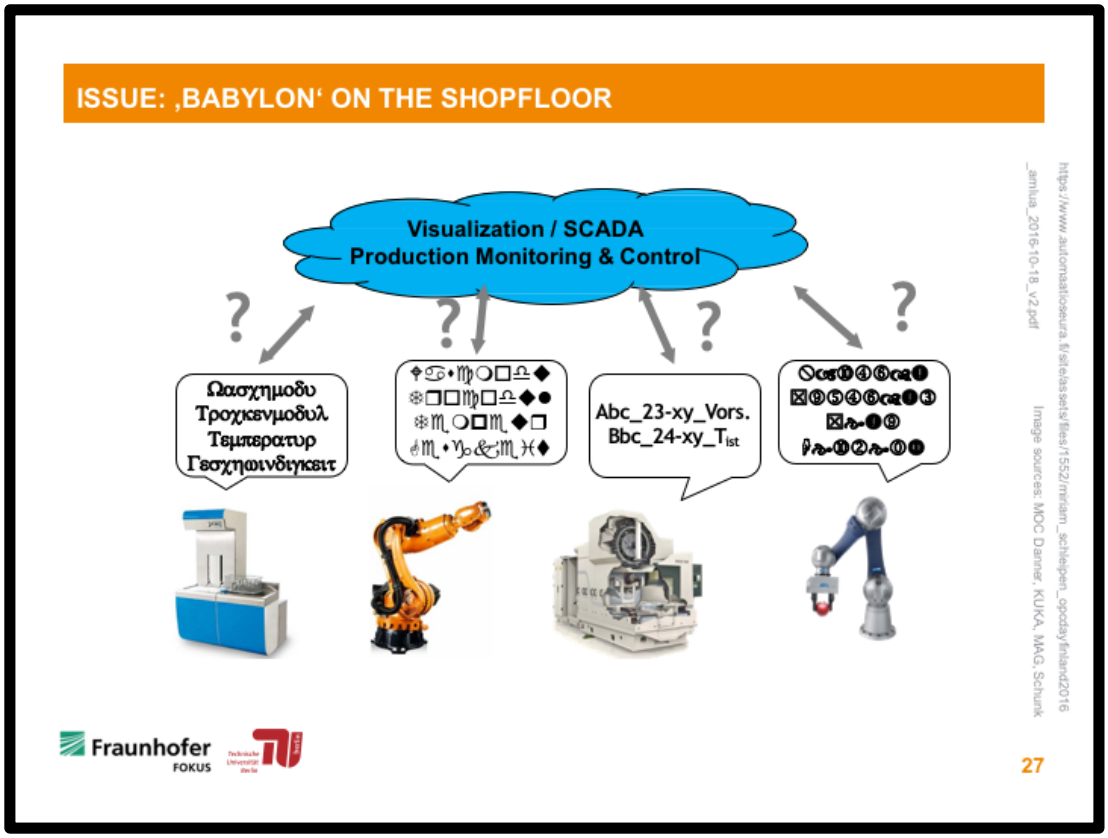
- OPC UA over TSN is currently under development
- It is the most prominent combination of technologies for the Industry 4.0 context
- Will provide real-time capable Peer-to-Peer communication
- Combines OPC UA Pub/Sub technology with IEEE TSN Ethernet standards



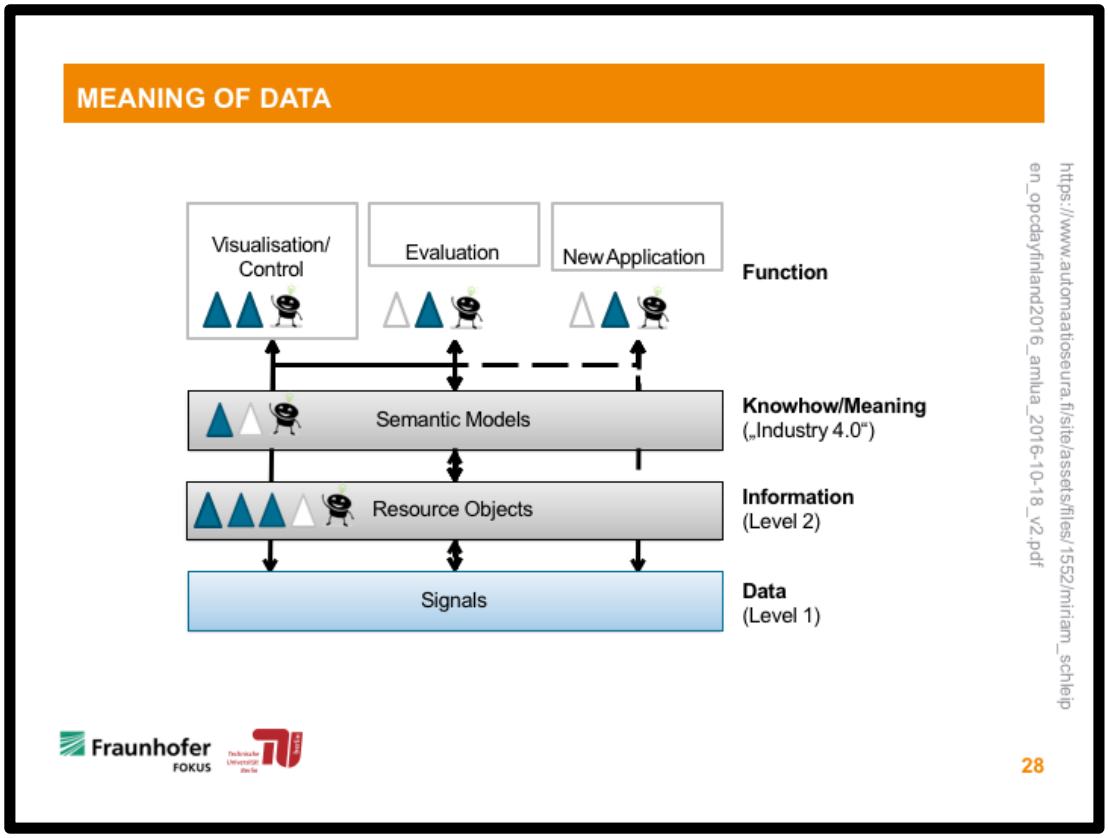
- PROFINET, Sercos, Modbus, … over TSN are currently offered
- However, in the future OPC UA over TSN will replace Industrial Ethernet technologies



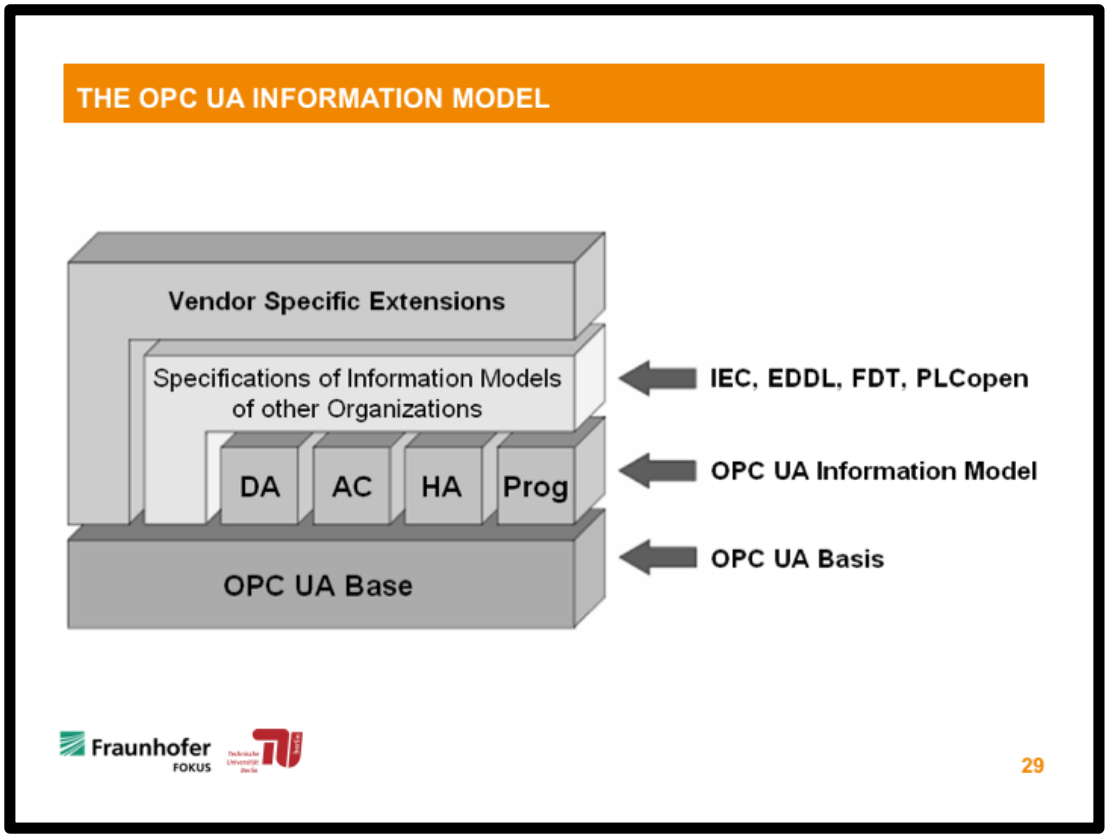
- However, there are still open issues to be solved before typical Industrial Ethernet stacks can be replaced



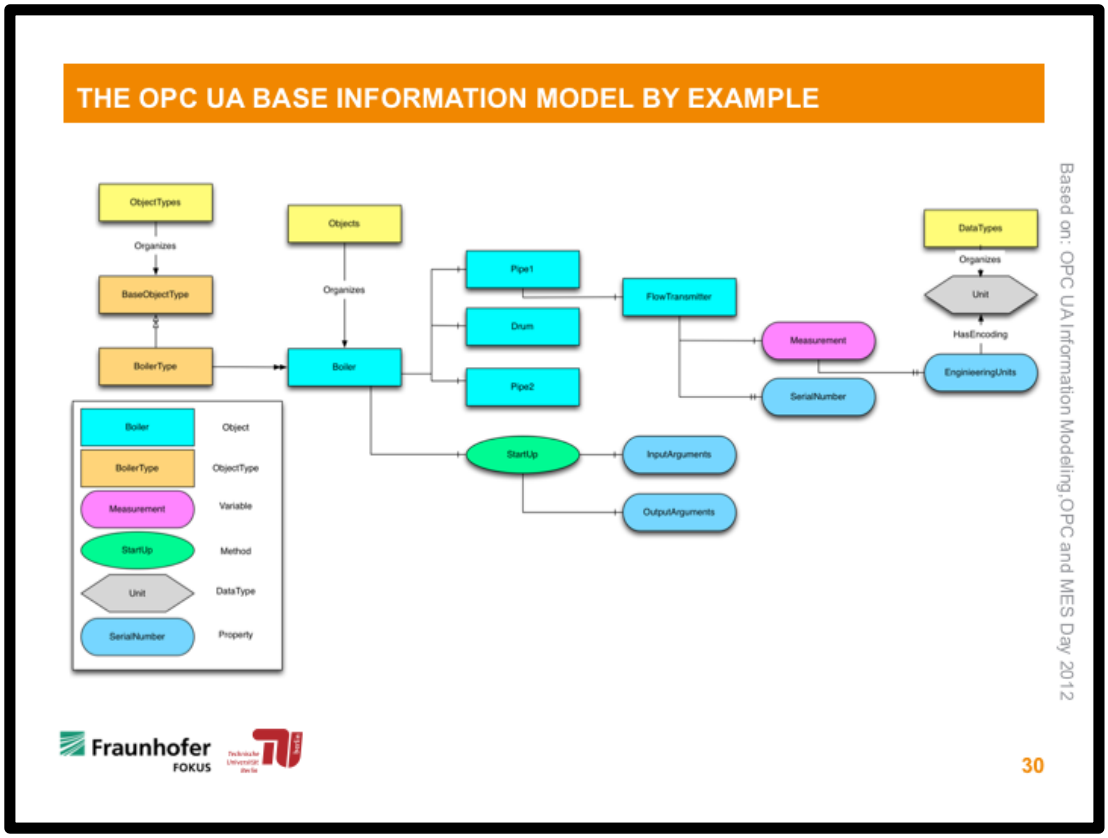
- Different “languages” per machine
- SCADA & MES need to understand all different technologies on the shopfloor



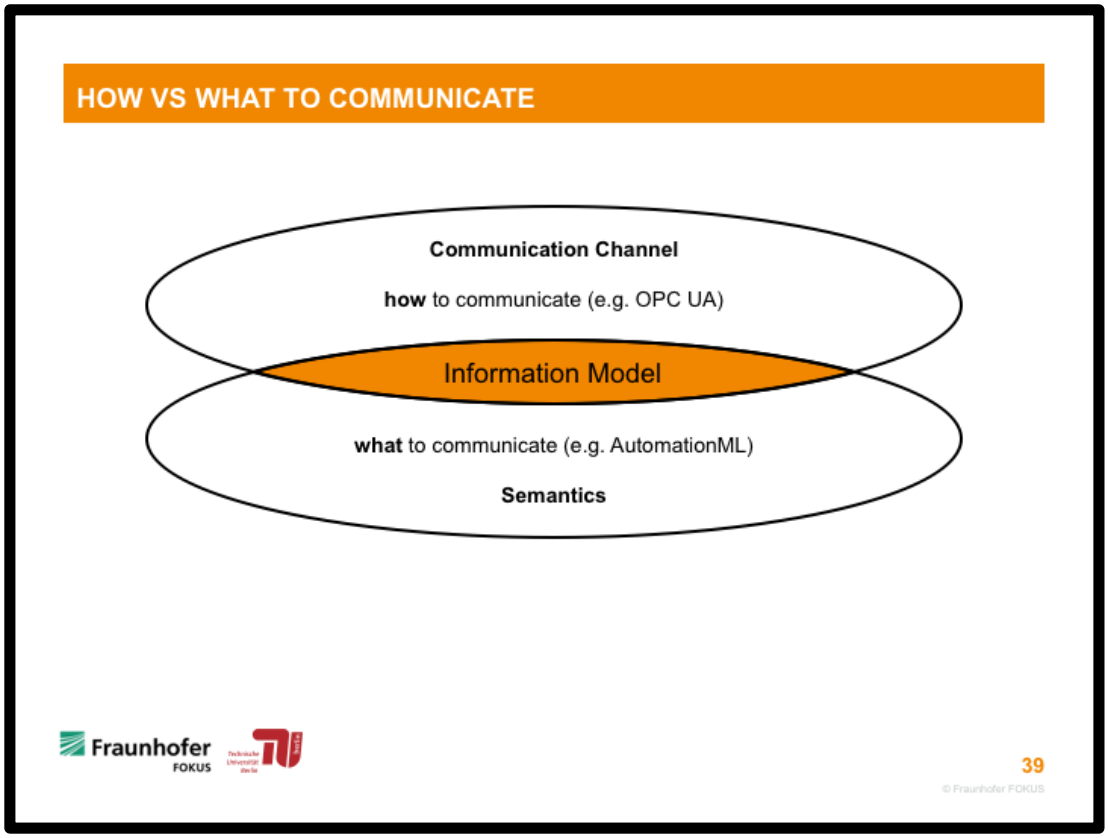
- Level 1 / Industry 2.0: Raw Data, connecting the different I/O from the devices
- Level 2 / Industry 3.0: adding information to devices
- Level 3 / Industry 4.0: Meaning by semantic annotations
- More about this at the end of this course



- Information Model describes standardized nodes of a server's Address Space
- Nodes are standardized types as well as standardized instances used for diagnostics or as entry points to server specific nodes
- Thus, the Information Model defines the Address Space of an empty OPC Unified Architecture server



- Yellow = OPC UA core information model (types)
- Different objects can be introduced and relations can be established
- For modeling different kind of devices
- Information Model Example for a boiler
    - Has to pipes and a drum
    - One pipe has a flow transmitter
    - The boiler has a “StartUp” function



- OPC UA specifies HOW to communicate.
- AutomationML specifies WHAT

AUTOMATION ML INITIATIVE

- Founded in 2006 as industrial consortium, registered association since 2009
- Goal: (Self-)Description of Components, Machines and Plants
  - Production system components and their skills
  - Function-oriented descriptions of production tasks
  - Methods for an automatic matching/comparison
  - Description of the access path to the functions

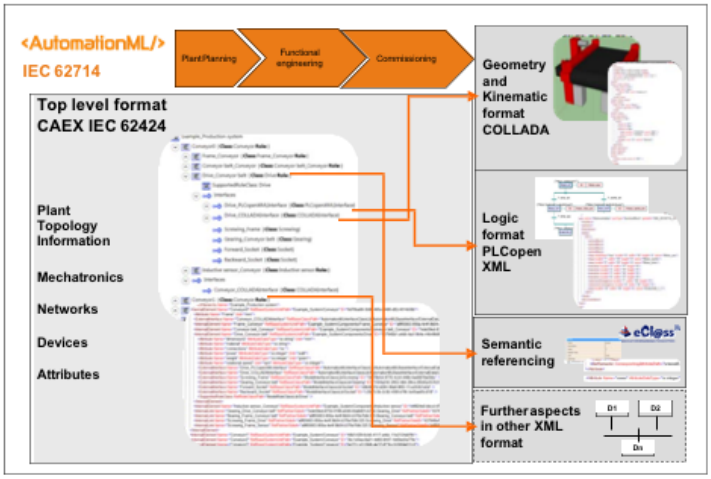
- Promoting and further development of AutomationML
  - Standardizing data exchange
  - engineering process of production system
- Develops and maintains data representation standard
  - *Open, neutral, XML based, and free*
  - Designed for industry purposes
  - Enables domain and company crossing transfer

Fraunhofer FOKUS

40

- Is ”XML” really about semantics? More about this later.

AUTOMATION ML BY EXAMPLE



https://www.automationml.org

Fraunhofer FOKUS

41

- CAEX (Computer Aided Engineering Exchange) is a neutral data format that allows storage of hierarchical object information
- Pointers to other data formats (COLLADA, PLCopen XML, …)
- Semantic referencing → pointer to eCl@ss characteristics catalogue

OPC UA INFORMATION MODEL

Questions?

ONEM2M

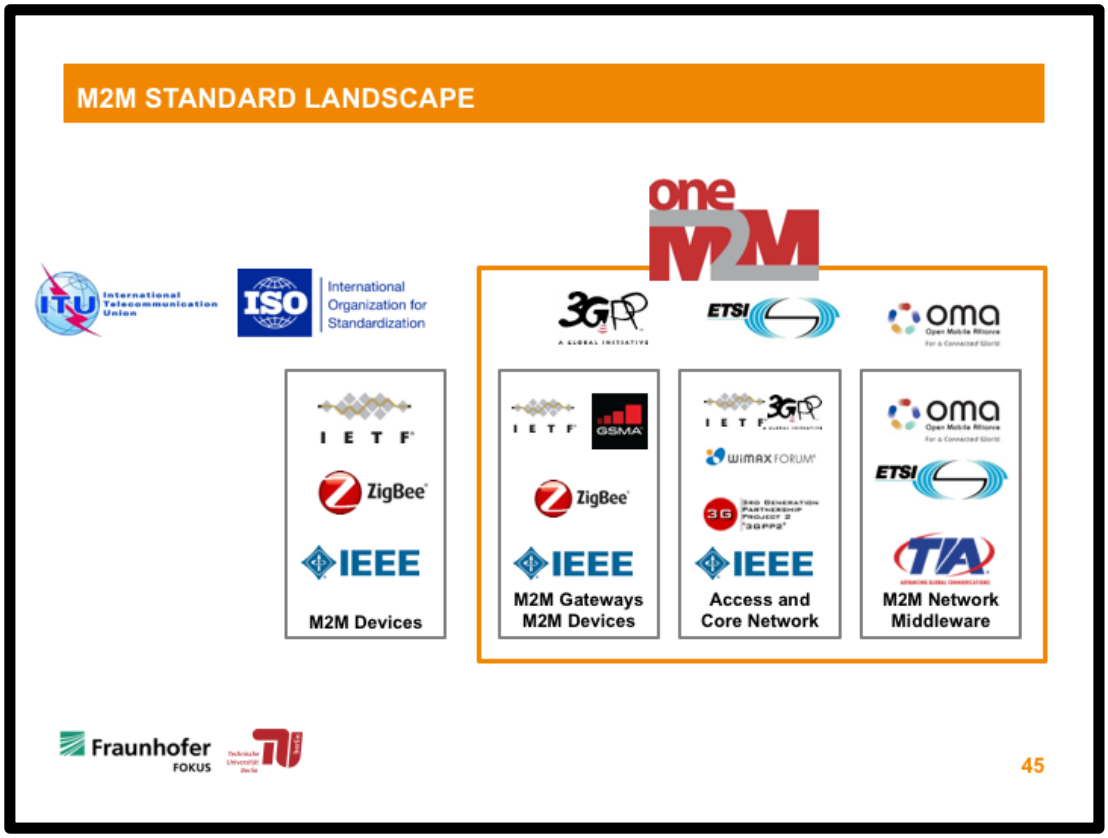
10 minutes

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.

Fraunhofer FOKUS

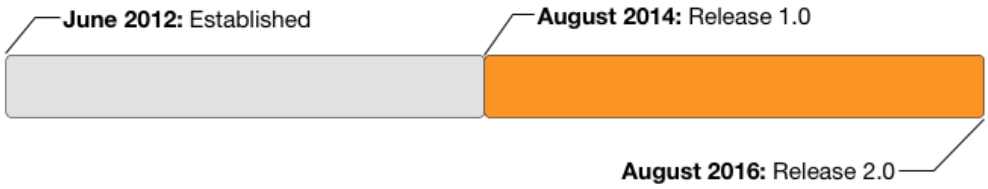


- M2M standards address in general only one part of the M2M communication
- ETSI, OMA & 3GPP standards together develop a complete network oriented M2M communication architecture
    - Release 1 (published Jan 2015, updated Mar 2016)
    - Release 2 (published Aug 2016)

- 8 standardization bodies (Partner 1) -> ETSI, CCSA,…
- 4 standardization bodies (Partner 2) -> CEN, CENELEC, OMA, GlobalPlatform
- 6 Associate Members -> NIST, …
- 176 Members -> Fraunhofer FOKUS, …

HISTORY (PARTLY AS A CONTINUATION OF ETSI M2M, WHICH STARTED 2009)

ETRI



Fraunhofer FOKUS

47

ONEM2M GENERAL ARCHITECTURE

Application Layer (AE):

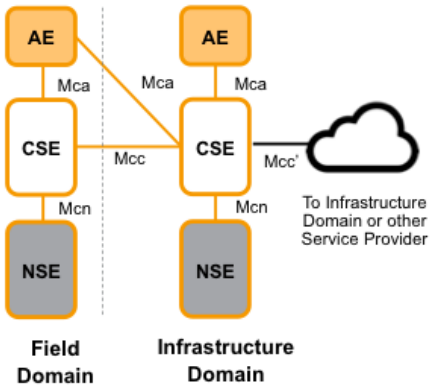
- Comprises oneM2M Applications and related business and operational logic.

Common Services Entity (CSE):

- A set of service functions common to the M2M environment.
- Common Services Entity can utilize Underlying Network capabilities and can interact with each.

Underlying Network Services Entity (NSE):

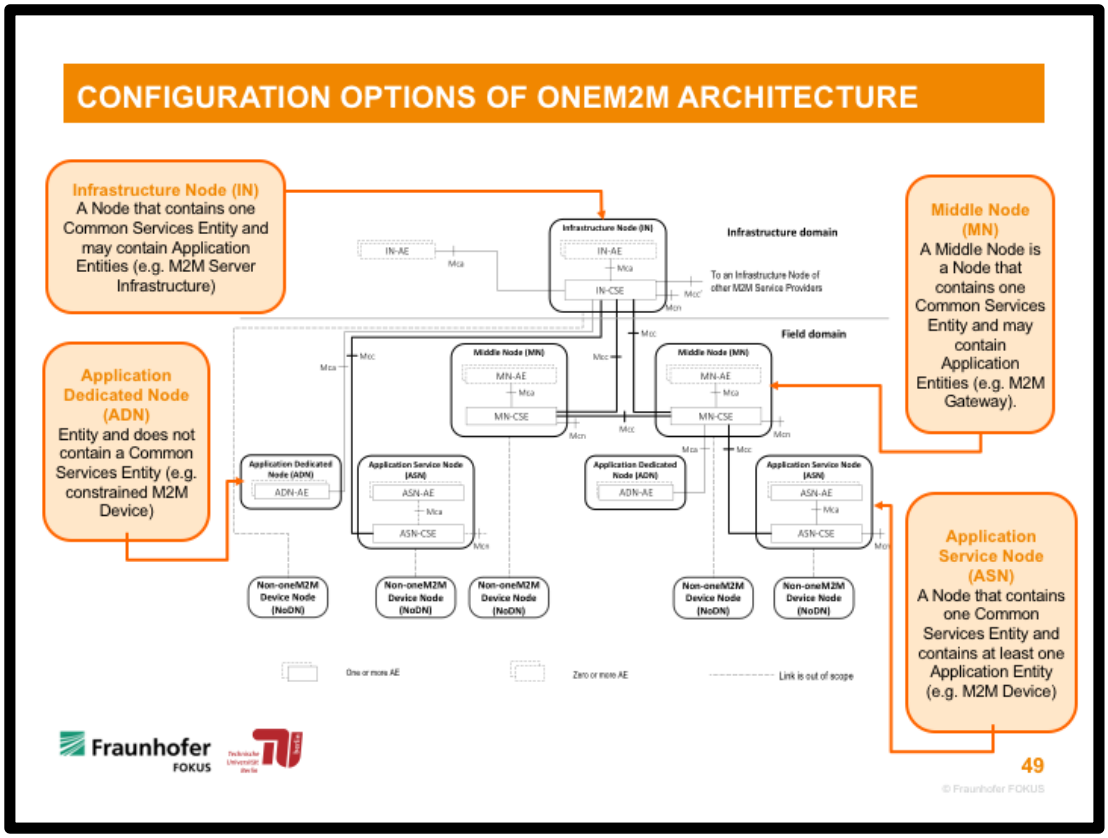
- Provides services to the CSEs.
- Examples of such services include device management, location services and device triggering.



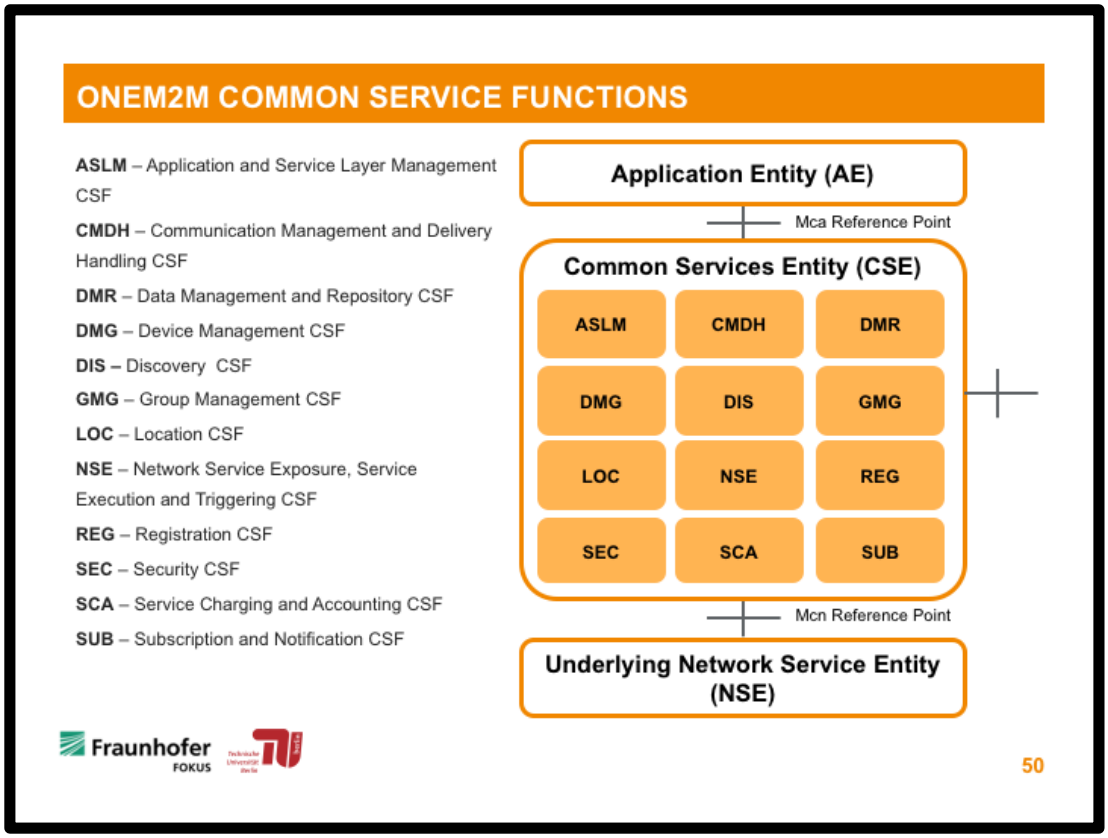
Fraunhofer FOKUS

48

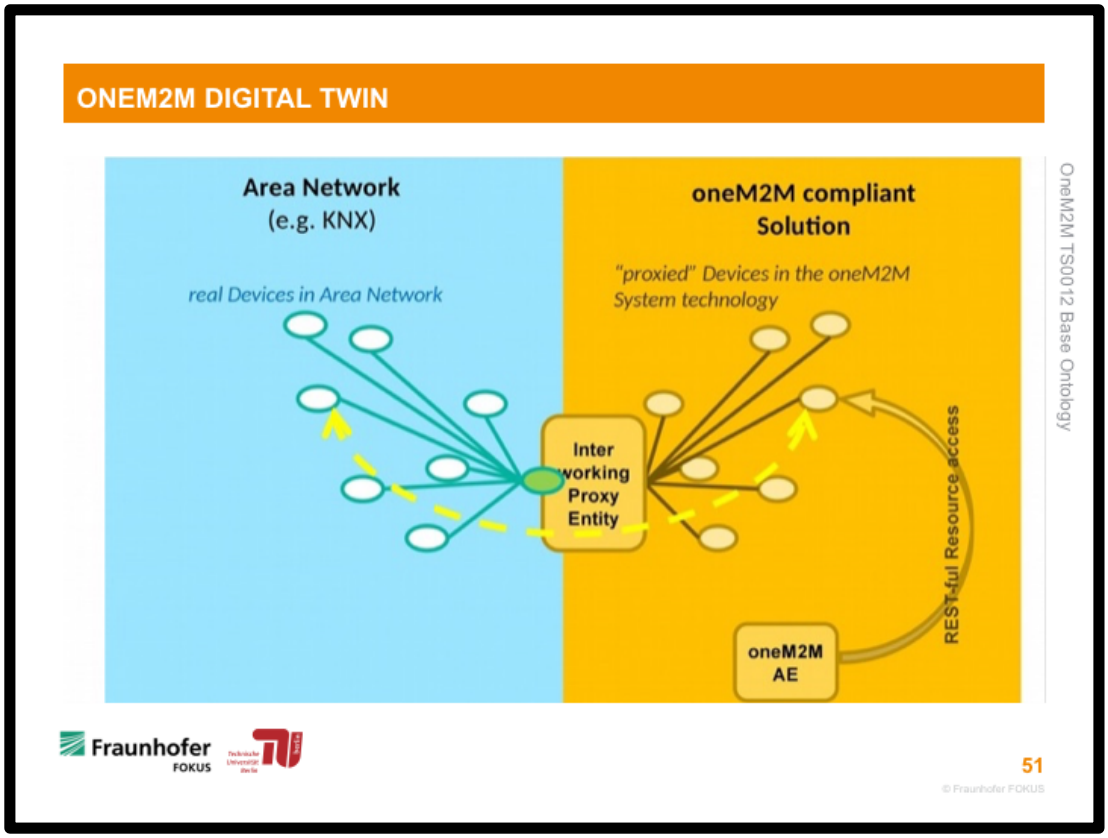
- Three horizontal layers (AE, CSE, NSE)
  - AE -> applications
  - CSE -> middleware
  - NSE -> other service
- Two vertical domains (Field, Infrastructure)
  - Infrastructure -> management
  - Field -> shop floor, devices
- Three different interface (Mca, Mcc, Mcn)
  - Mca -> AE to CSE
  - Mcc -> CSE to CSE
  - Mcn -> CSE to NSE
- One Network per Service Provider



- Four different kind of nodes (IN, MN, ASN, AND)
  - Infrastructure Node in the Infrastructure Domain
  - Middle Node, Application Service Node and Application Dedicated Node in the Field Domain
- Nodes hosts AEs and can host CSEs -> concatenation of node type and entity type
- IN-CSE as central point per Service Provider
  - The root of the tree-like organization
  - Provides the Mcc' interface to provide access to other Service Providers
  - Resides in the Infrastructure Domain
- MN-CSE and ASN-CSE act as service point for field domain applications
- Non-oneM2M Device Nodes (NoDN) can be implemented (devices that are not oneM2M-aware)

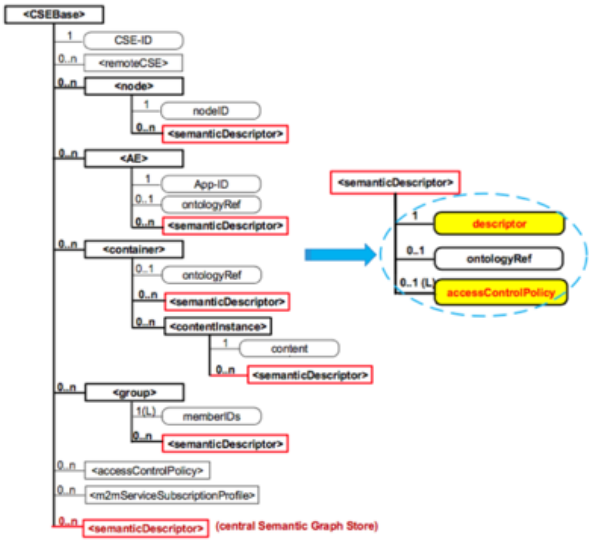


- Example for a set of functions offered by a complex middleware
- Service Functions describe a set of functionality
- Logical grouping, not necessarily programmatically
- Different functions are overlapping or related (for example DIS, SUB and DMG)
- Used to described all functionalities last year in detail



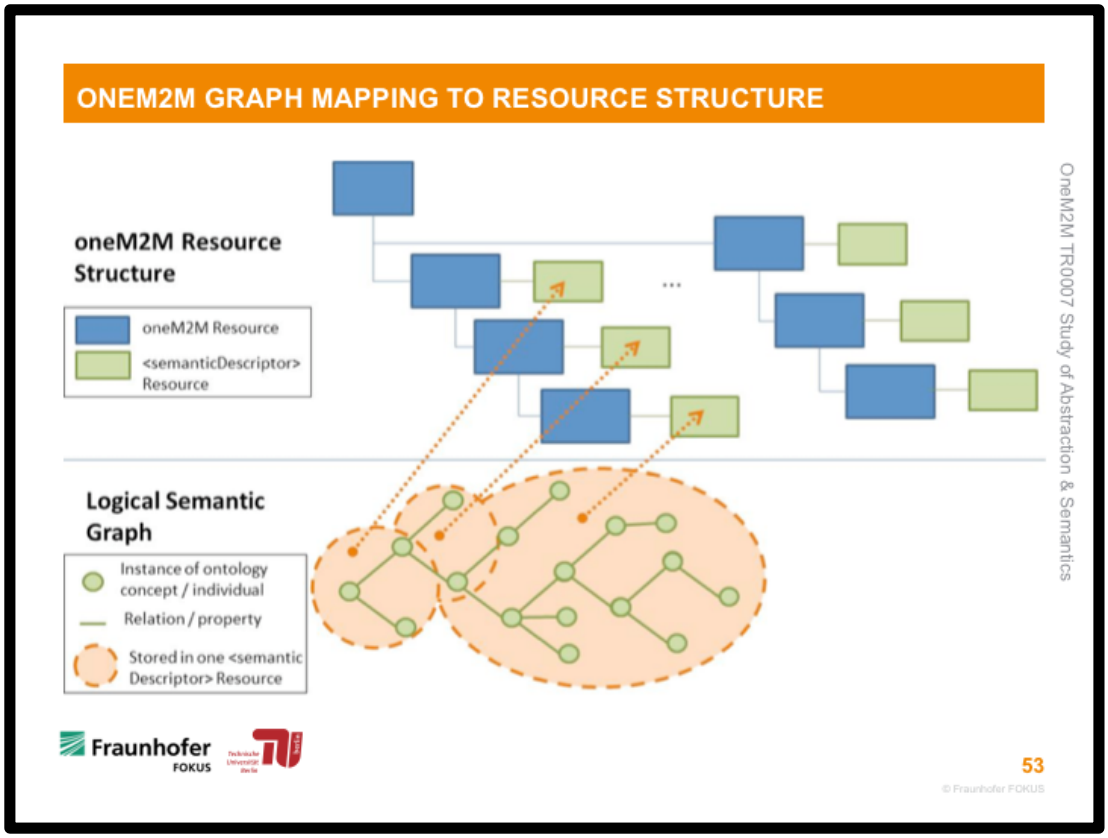
- Proxy for various technologies
- Access to resource information through oneM2M information model

ONEM2M INFORMATION MODEL: TREE AND SEMANTIC DESCRIPTOR



Fraunhofer FOKUS

- In version 1.0: tree-based information model with CSEBase, AE's, Nodes, Containers, …
- In version 2.0: use of semanticDescriptors pointing to RDF graphs



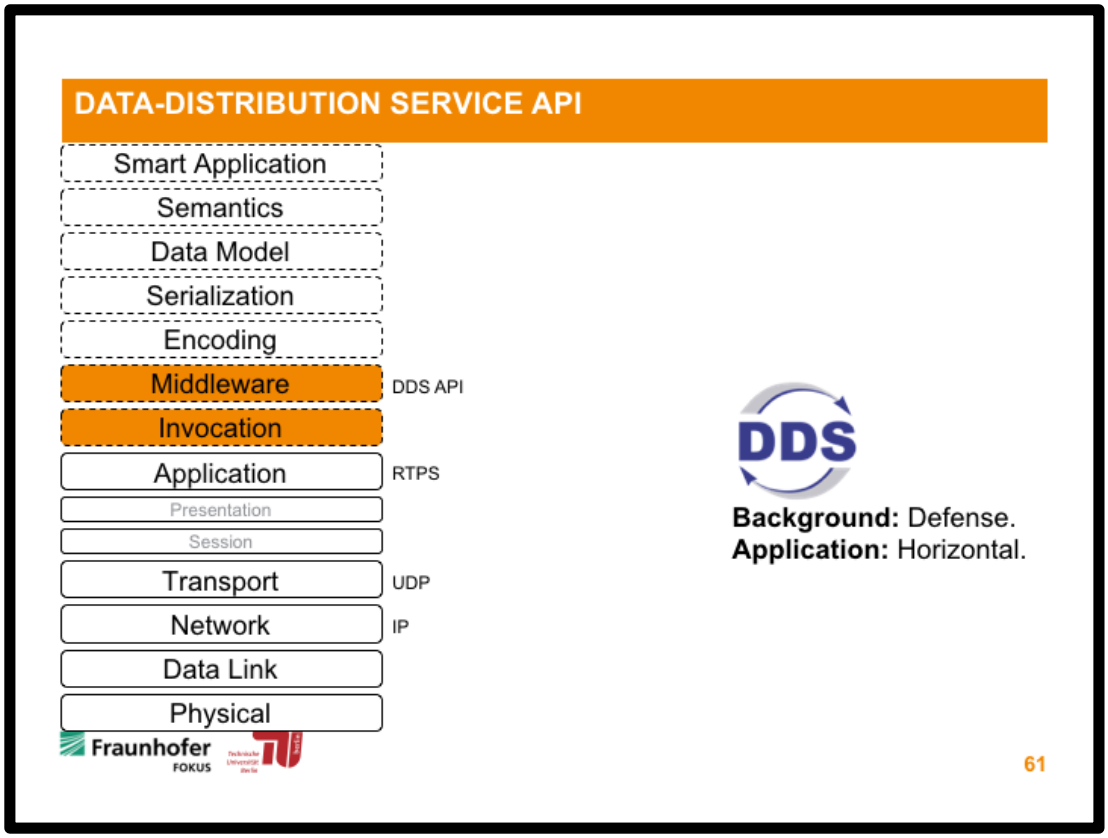
- oneM2M resource tree and semantic resources used in parallel
- Semantic resource description spans a (connected) semantic graph of information

ONEM2M

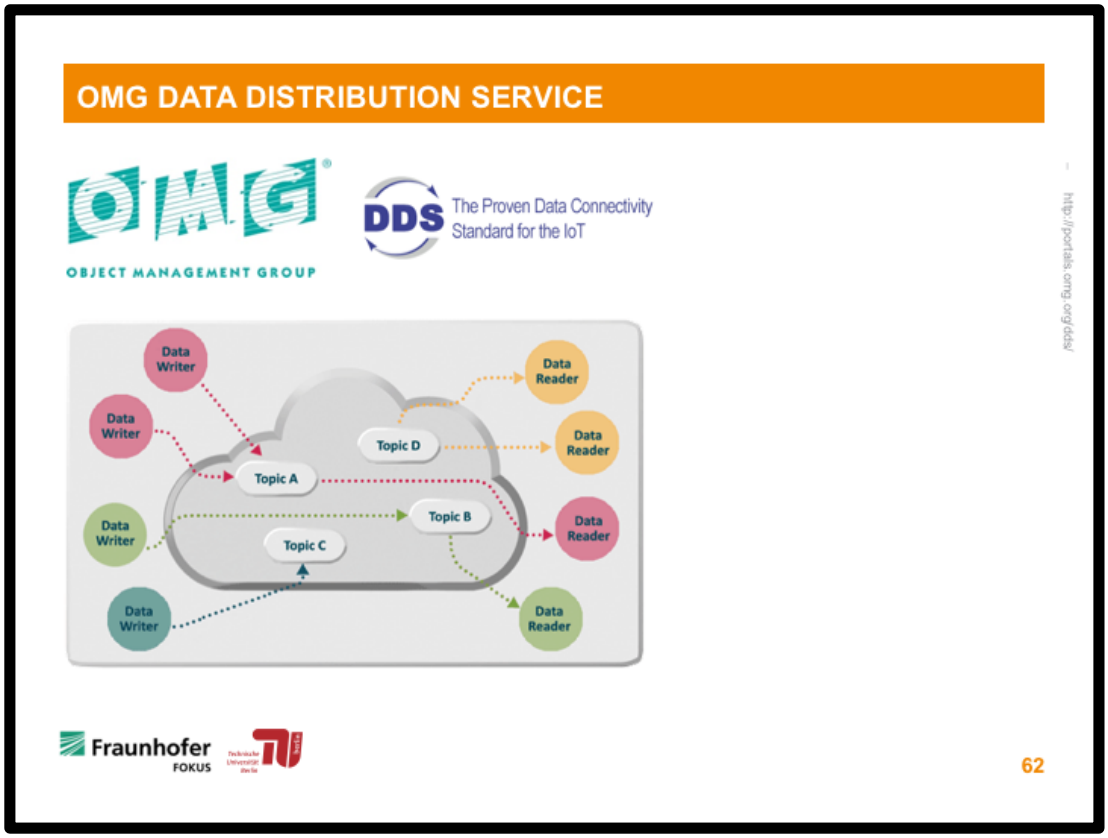
Questions?

DDS

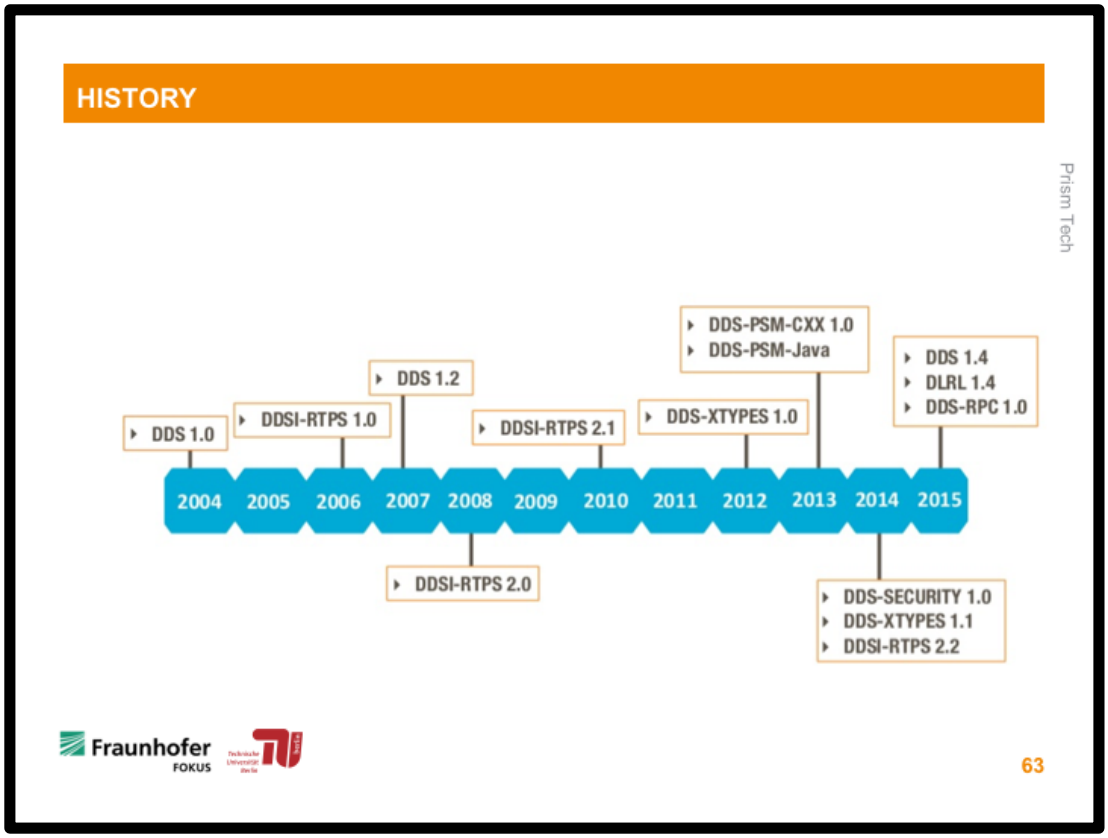
10 minutes



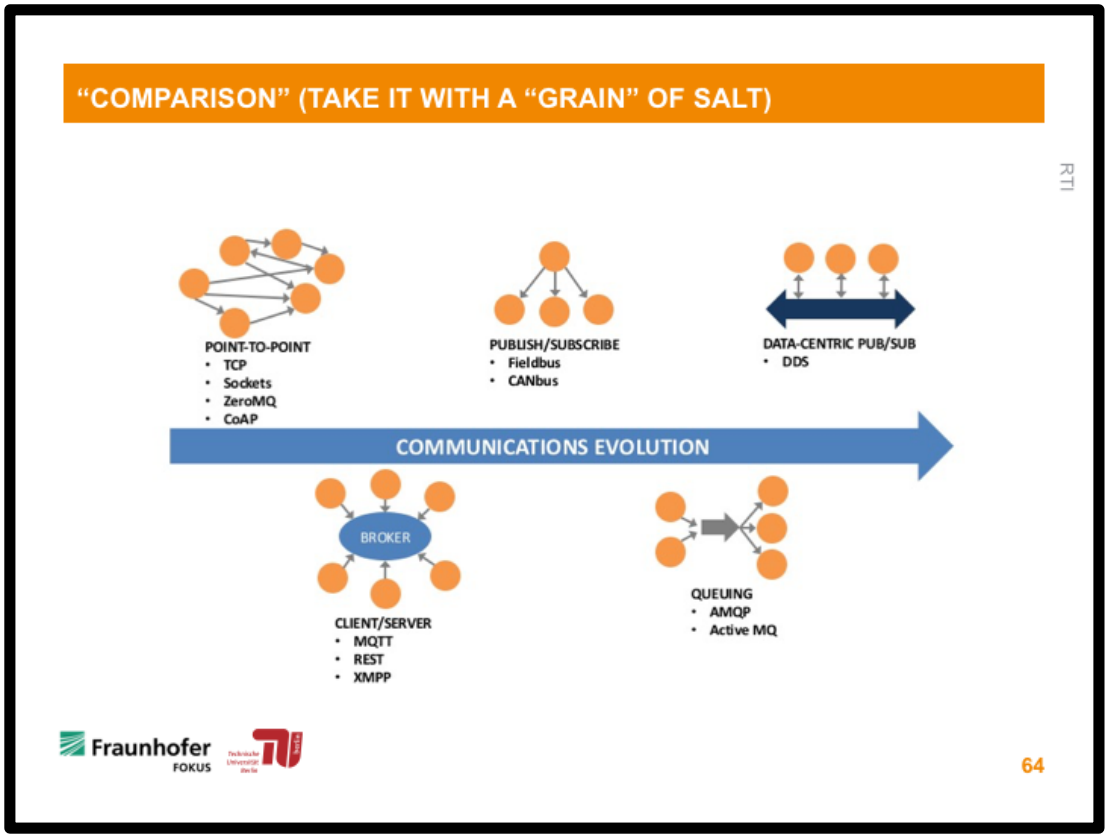
- Real-Time Publish-Subscribe (RTPS) Wire Protocol



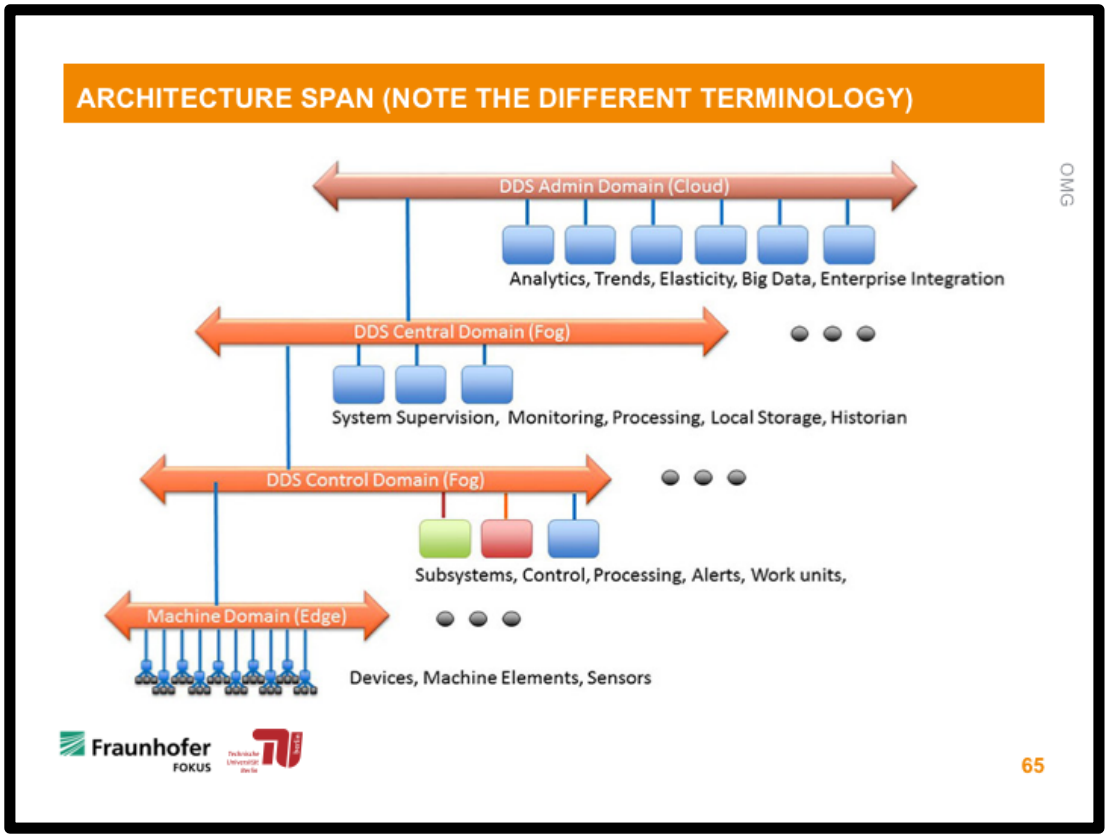
- Data Distribution Service (DDS) is a middleware protocol and API standard for data-centric connectivity from the Object Management Group
- Provides low-latency data connectivity
- Based on Publish/Subscribe mechanism
  - DDS middleware ensures all subscribers have a correct and consistent view of the data
  - Each publisher writes data to the global data space
  - Each subscriber gets notified of changes
  - Dynamic discovery mechanism matches DataReaders and DataWriters



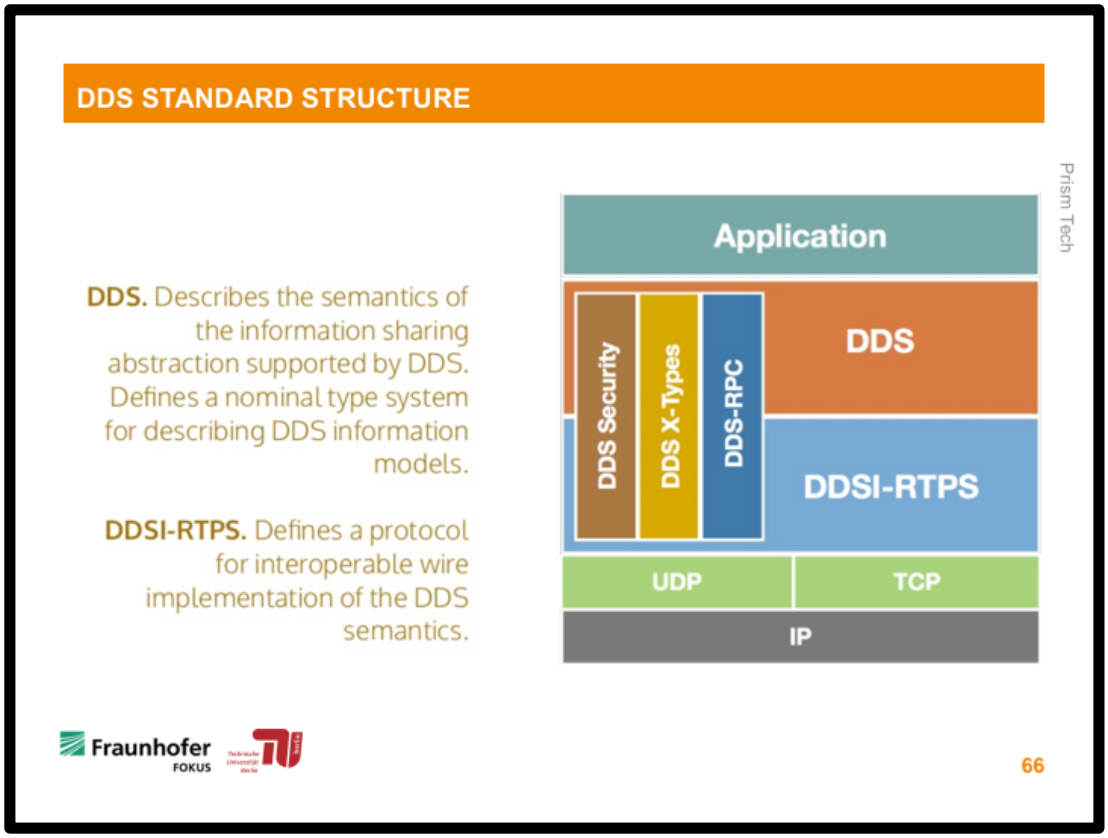
- Long history back to 2004
- More details about the extensions later



- Data-centric pub/sub as "latest evolutionary step"?
- Source by the most prominent vendor of a DDS implementation
- Different technologies mixed



- Four different domains
  - Different dimensions
  - Different clients (publisher/subscribers)
  - Different Tasks
- Some data stays in one domain, some data transferred between domains

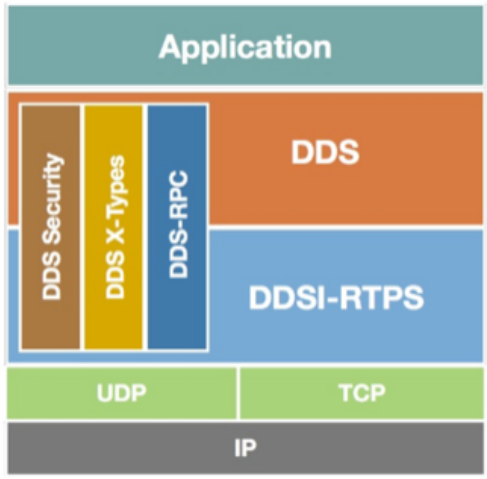


- DDS API for the application based on Data-centric publish-subscribe (DCPS), which is targeted towards the efficient delivery of the proper information to the proper recipients
- DDS Interoperability Real-Time Publish-Subscribe Wire Protocol (DDSI-RTPS) for interoperability between several implementations

DDS STANDARD STRUCTURE

DDS-XTypes. Extends the DDS type system with support for structural typing as well as a dynamic type definition.

DDS-Security. Introduces information centric security in DDS for data in movement as as well as data at rest.



Prism Tech

Fraunhofer FOKUS

67

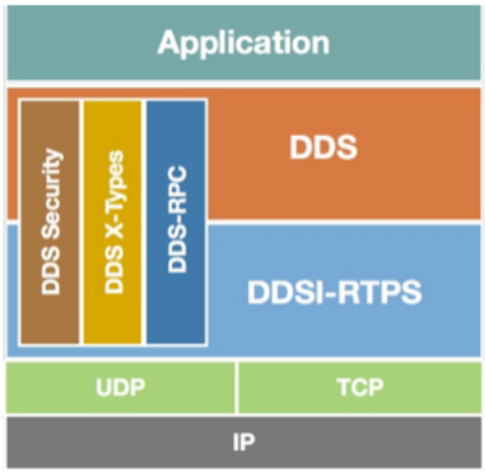
- X-Types → object oriented information model for shared “semantics”

DDS STANDARD STRUCTURE

DDS-RPC. Extends DDS with support for Remote Procedure Calls.

DDS-PSM-*. Defines highly ergonomic and optimised API mapping for specific programming languages instead of deriving those for the DDS-PSM-IDL

DLRL. Defines a language independent Object/Relational Mapping for DDS

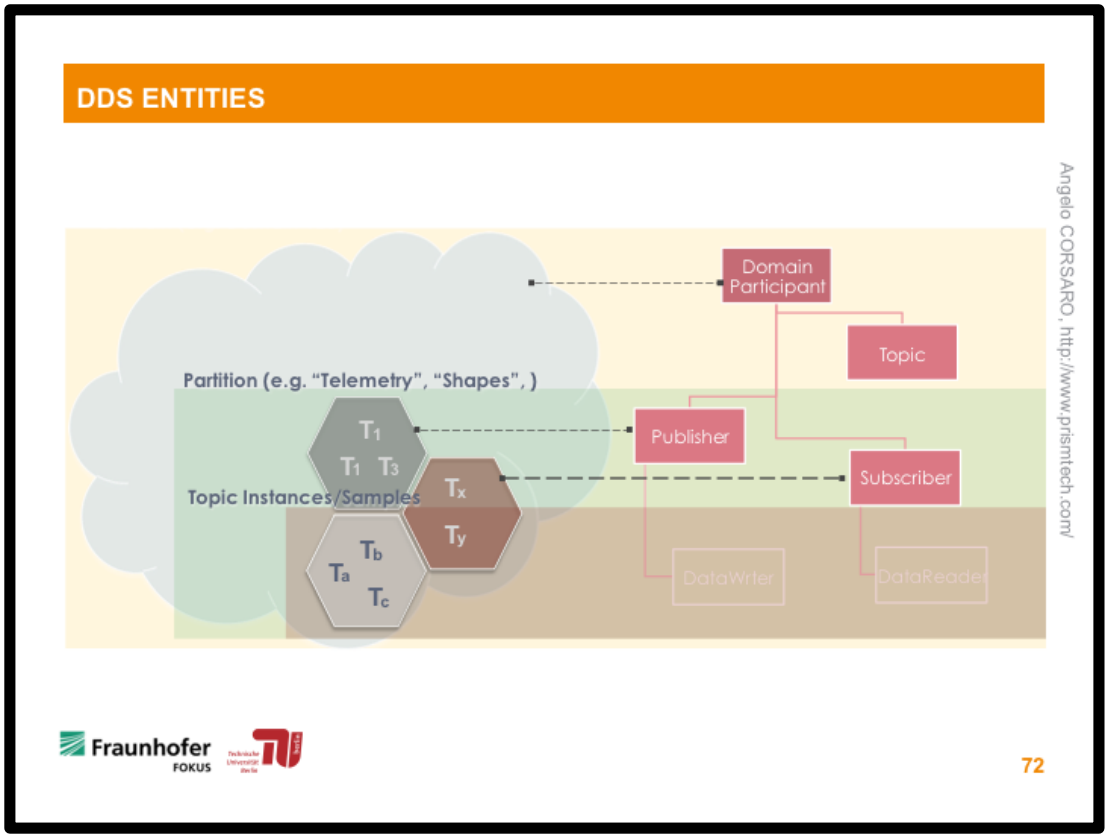


Prism Tech

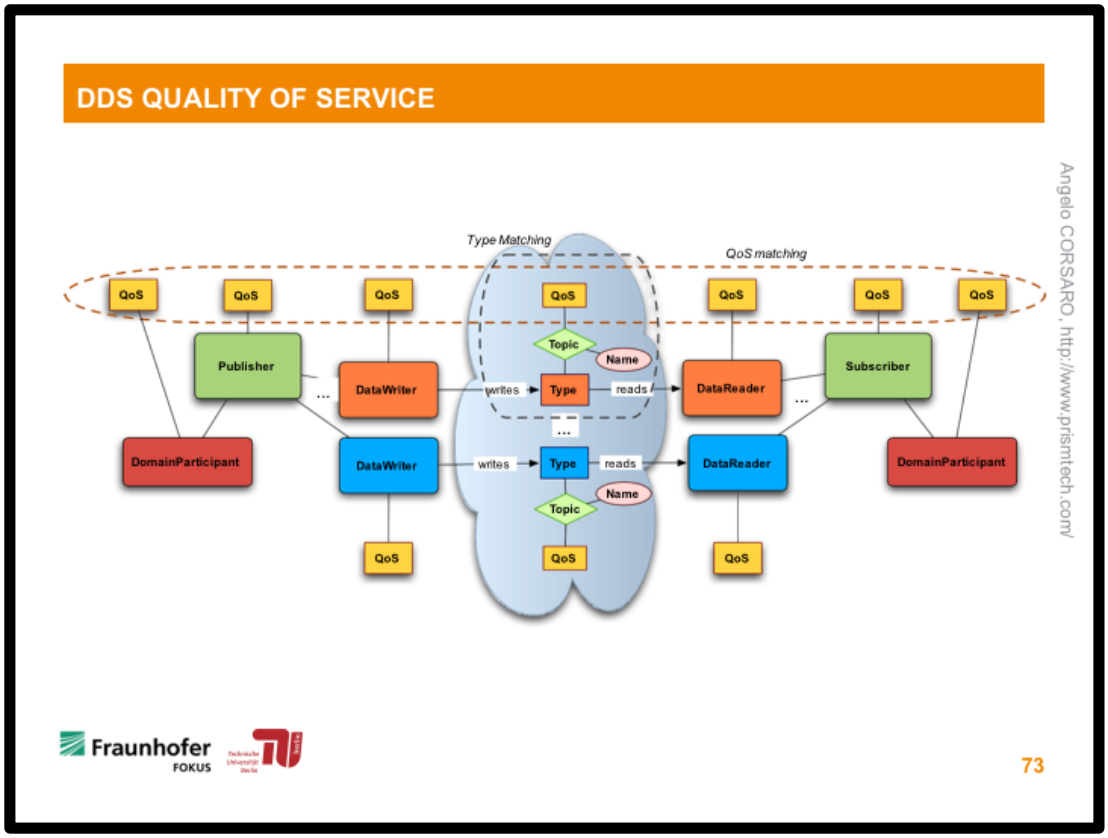
Fraunhofer FOKUS

68

- RPC → Remote Procedure Calls added to the protocol after 11 years
- PSM → code generator for high level programming languages
- Data local reconstruction layer (DLRL) allows for a simple integration of DDS into the application layer



- DomainParticipant: Provides access to a data cloud -- called a domain in DDS
- Topic: Domain-wide definition of a kind of Information
- Publisher/Subscriber: Provide scope to data sharing through the concept of partitions
- DataReader/DataWriter: Allow to read/write data for a given topic in the partitions their Subscriber/Publisher are associated with.



- Each DDS entity has an attached QoS-Policy
- QoS-Policies control local and end-to-end properties of DDS entities
- Local properties controlled by QoS are related to resource usage
- End-to-end properties controlled by QoS are related to temporal and spatial aspects of data

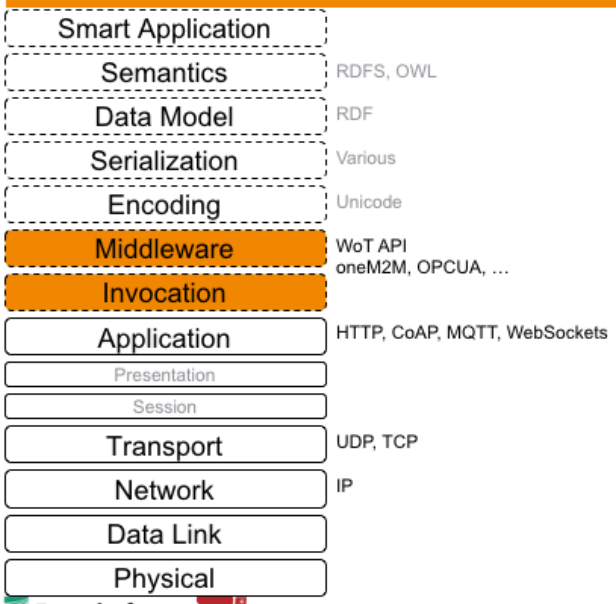
DDS

Questions?

W3C WEB OF THINGS

10 minutes

W3C WEB OF THINGS (WOT)



Background: lack of interoperability across platforms

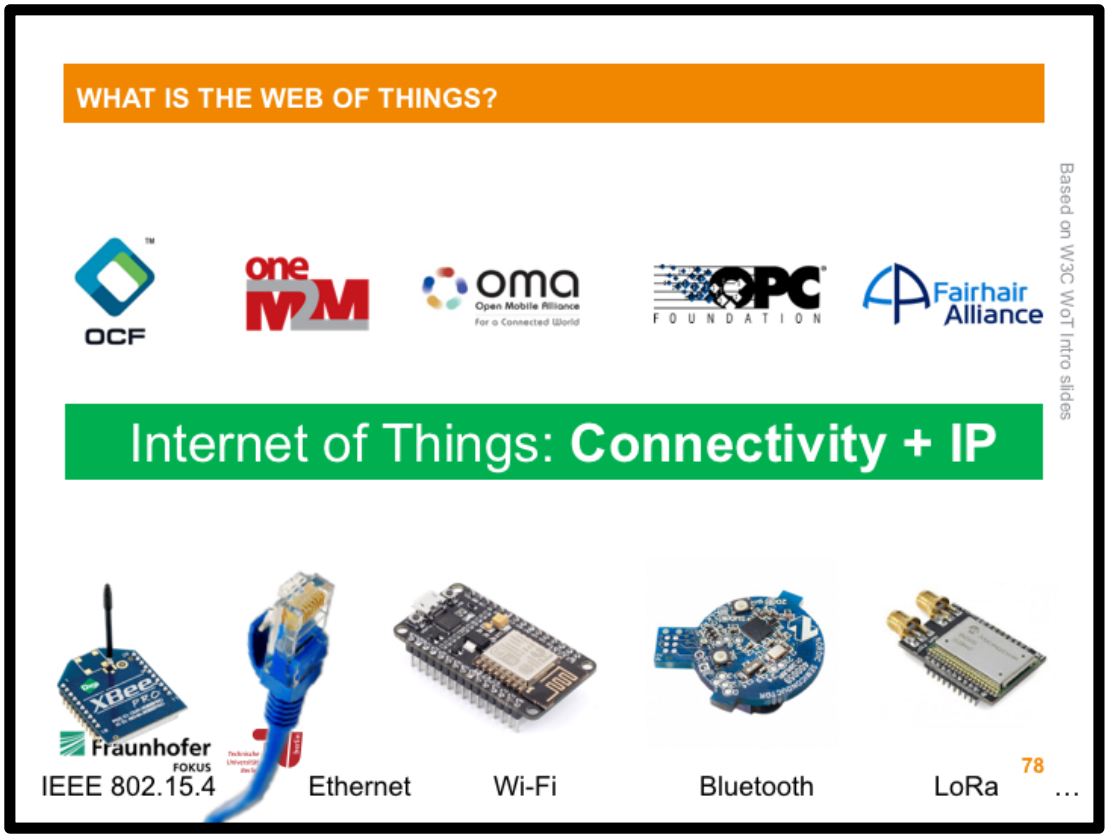
WHAT IS THE WEB OF THINGS?

Based on W3C WoT Intro slides

Internet of Things: **Connectivity + IP**

IEEE 802.15.4 Ethernet Wi-Fi Bluetooth LoRa …

77



Open Connectivity Foundation (OCF)

WHAT IS THE WEB OF THINGS?

Based on W3C WoT Intro slides

Web of Things: Application Layer

Internet of Things: Connectivity + IP

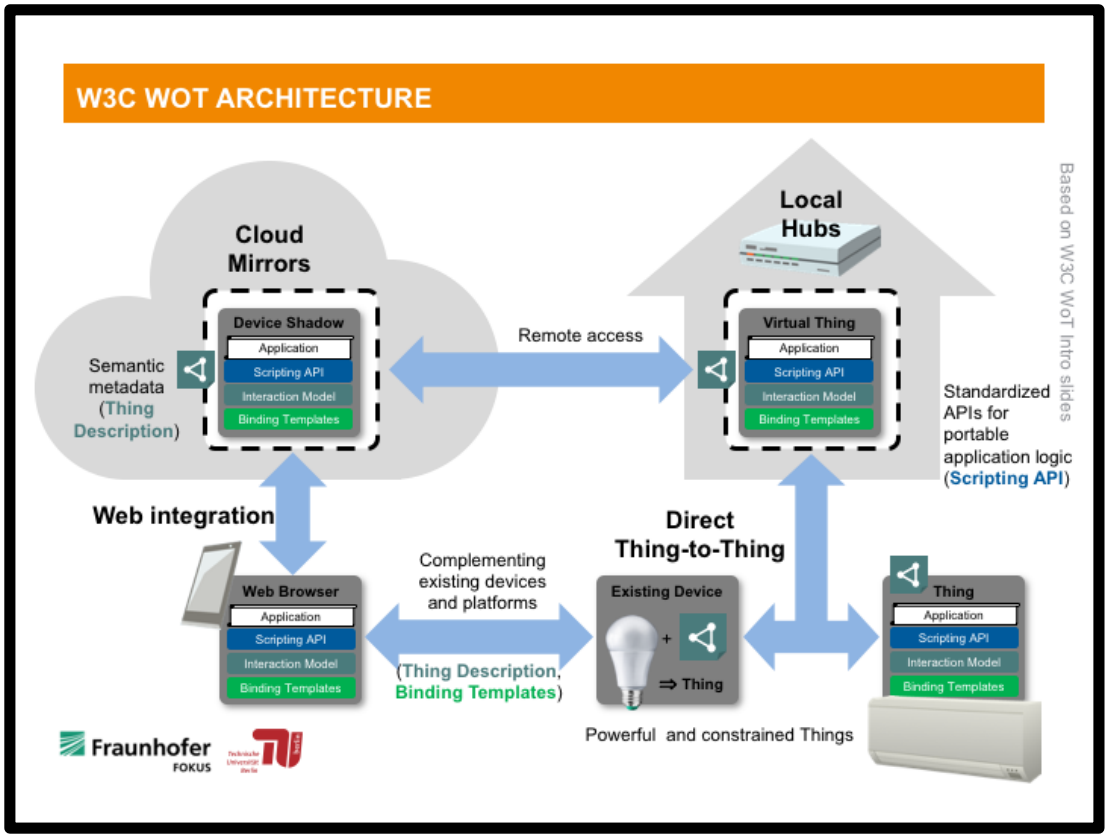
IEEE 802.15.4 Ethernet Wi-Fi Bluetooth LoRa …

79

MISSION: NOT YET ANOTHER STANDARD (STARTED 2015)

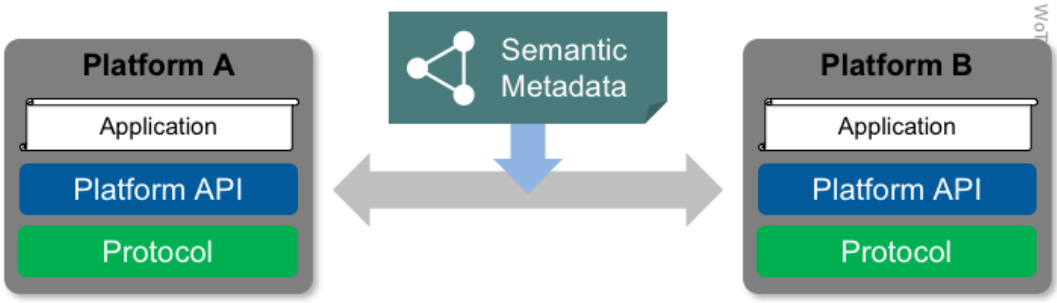
Based on W3C WoT Intro slides

Fraunhofer FOKUS



- Deployment of stack in different topological positions
- At least having a Things Description (TD)

SEMANTIC METADATA FOR INTEROPERABILITY

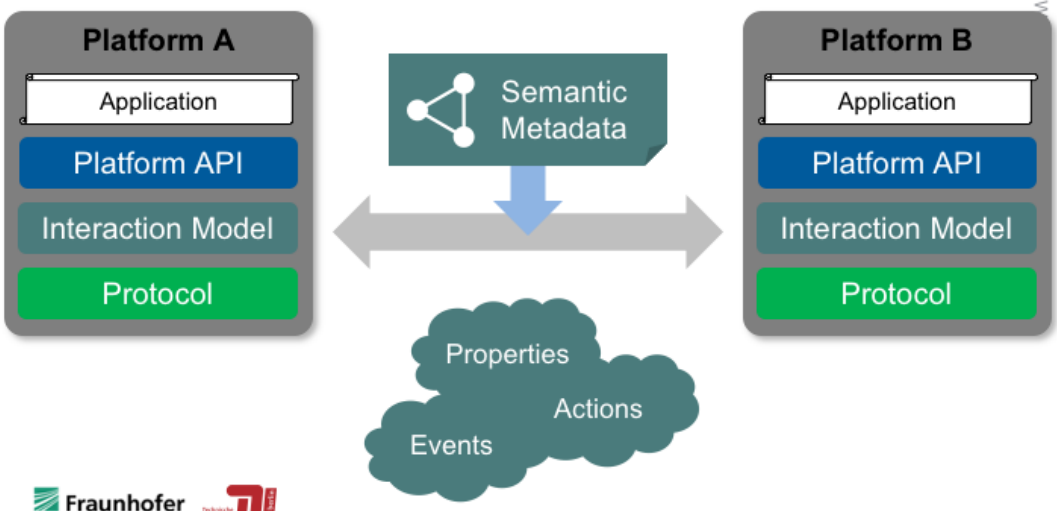


Based on W3C WoT

Fraunhofer FOKUS

82

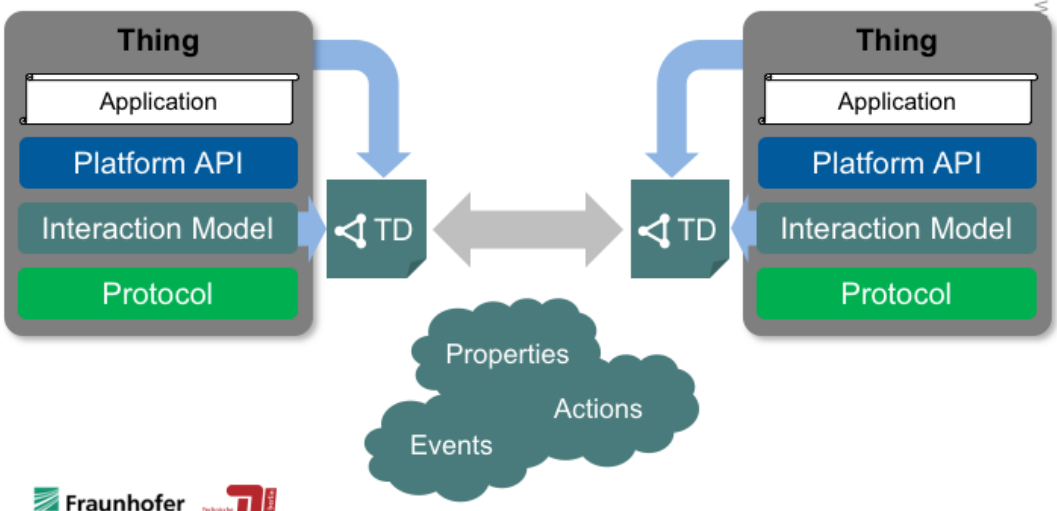
SIMPLE, COMMON INTERACTION MODEL



Fraunhofer FOKUS

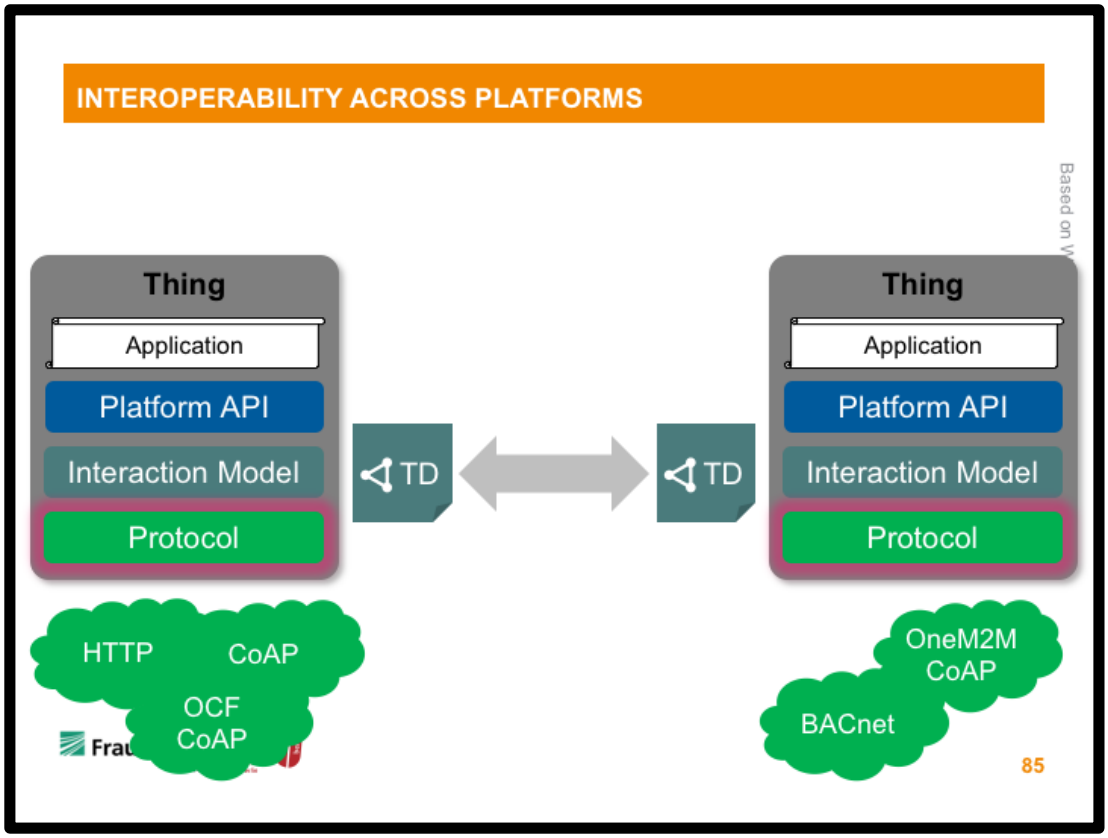
83

HTML FOR THINGS: THING DESCRIPTION (TD)

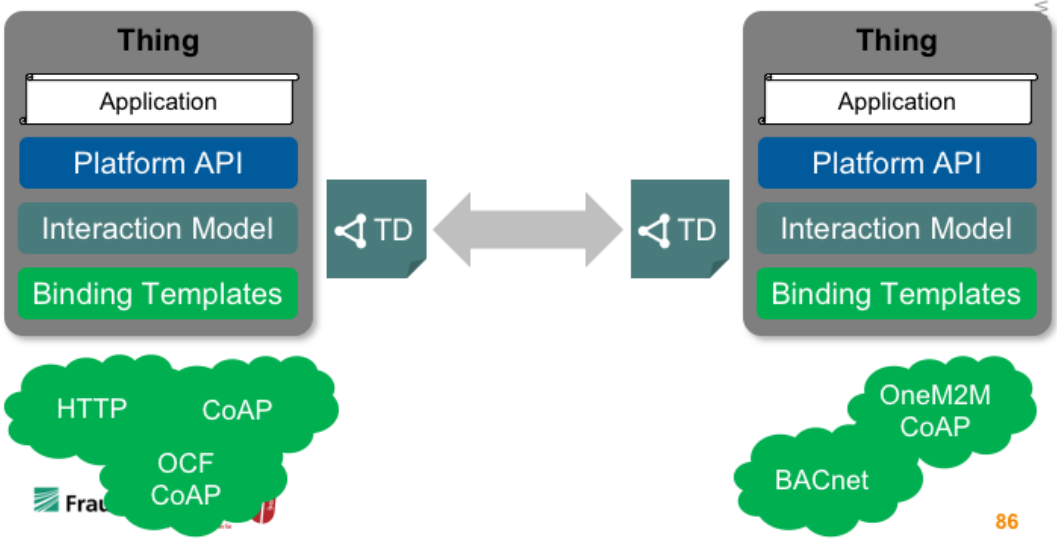


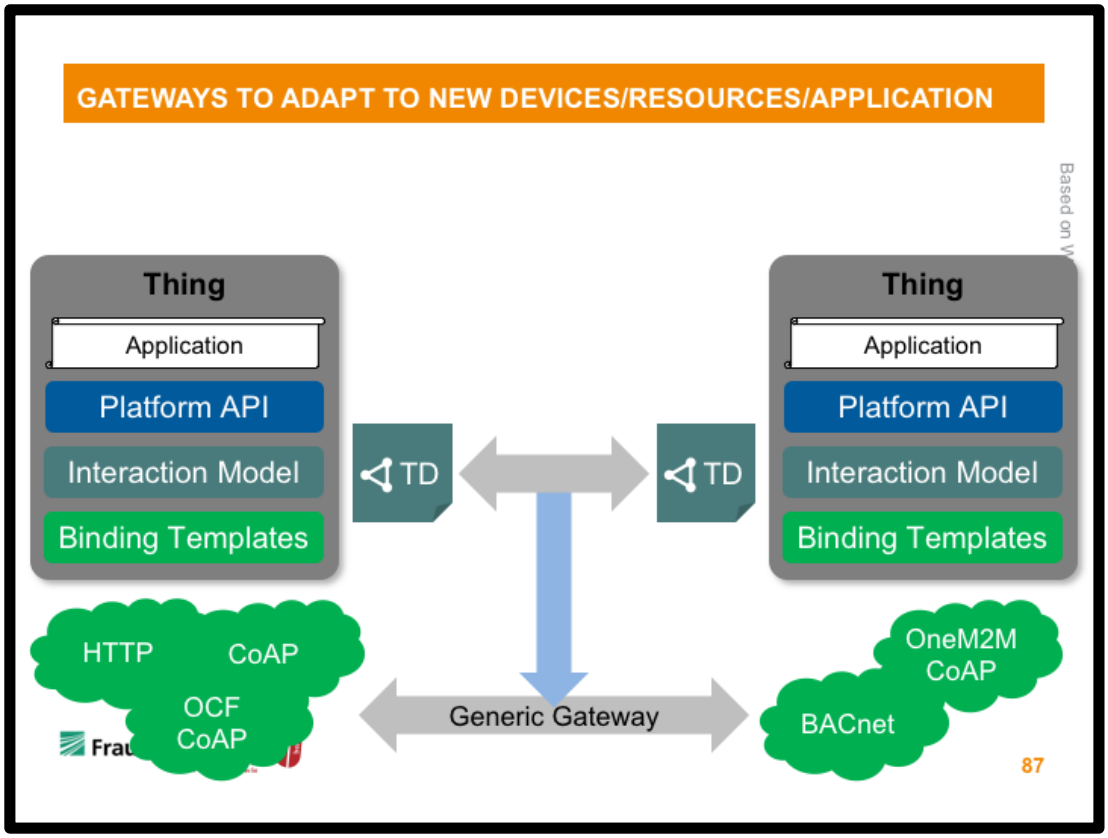
Fraunhofer FOKUS

84

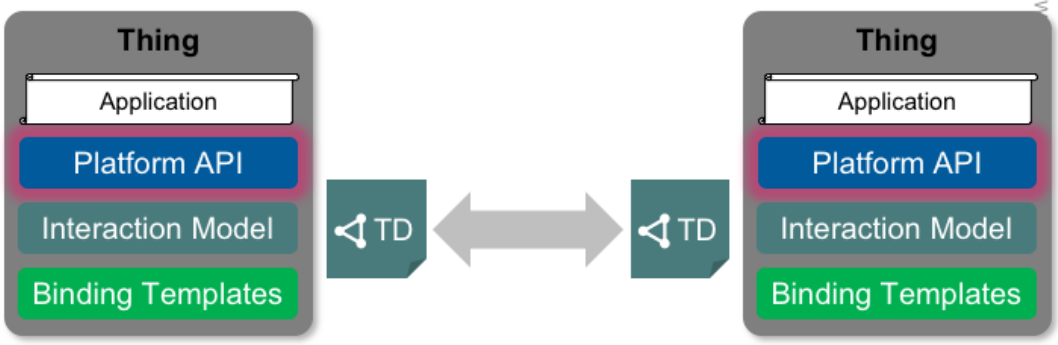

INTEROPERABILITY ACROSS PLATFORMS
Thing
Application
Platform API
Interaction Model
Protocol
TD
TD
Thing
Application
Platform API
Interaction Model
Protocol
HTTP
CoAP
OCF
CoAP
OneM2M
CoAP
BACnet
85

DESCRIPTION OF IOT PROTOCOLS: BINDING TEMPLATES




GATEWAYS TO ADAPT TO NEW DEVICES/RESOURCES/APPLICATION
Based on W
Thing
Application
Platform API
Interaction Model
Binding Templates
TD
TD
Thing
Application
Platform API
Interaction Model
Binding Templates
HTTP
CoAP
OCF
CoAP
Generic Gateway
OneM2M
CoAP
BACnet
87

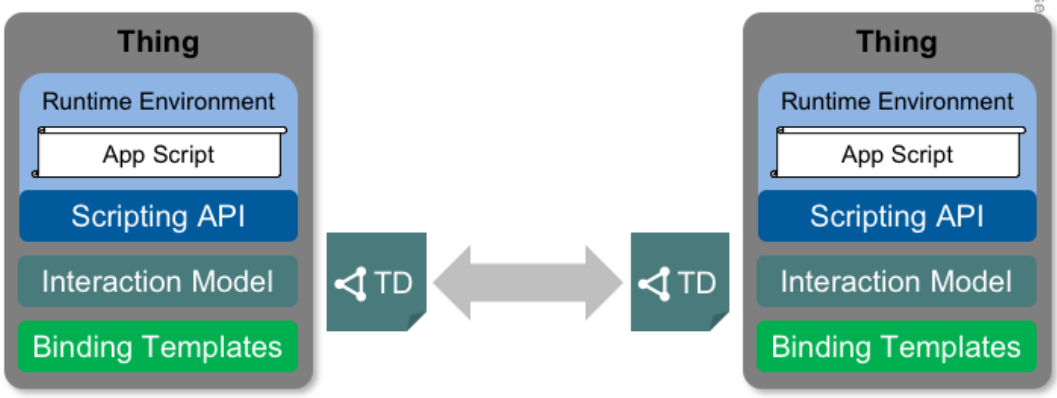
SIMPLIFY APPLICATION DEVELOPMENT



Fraunhofer FOKUS

88

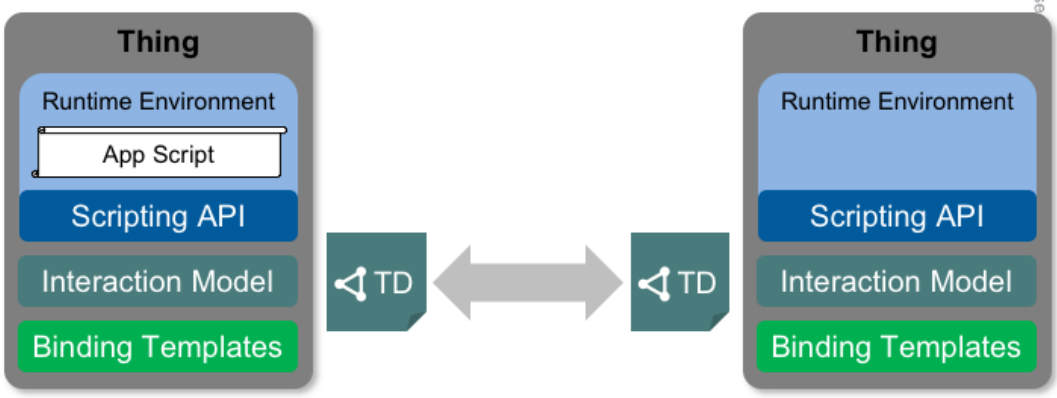
BROWSER-LIKE RUNTIME FOR IOT APPS: SCRIPTING API



Fraunhofer FOKUS

89

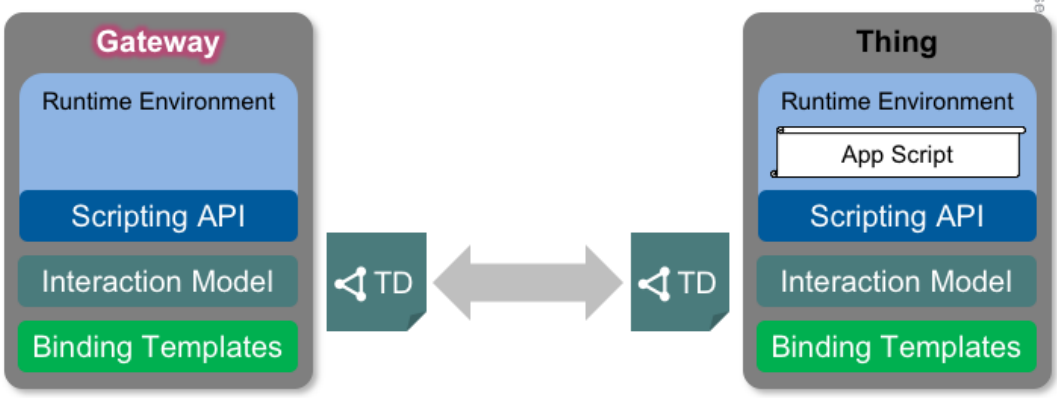
PORTABLE APPS ACROSS VENDORS



Fraunhofer FOKUS

90

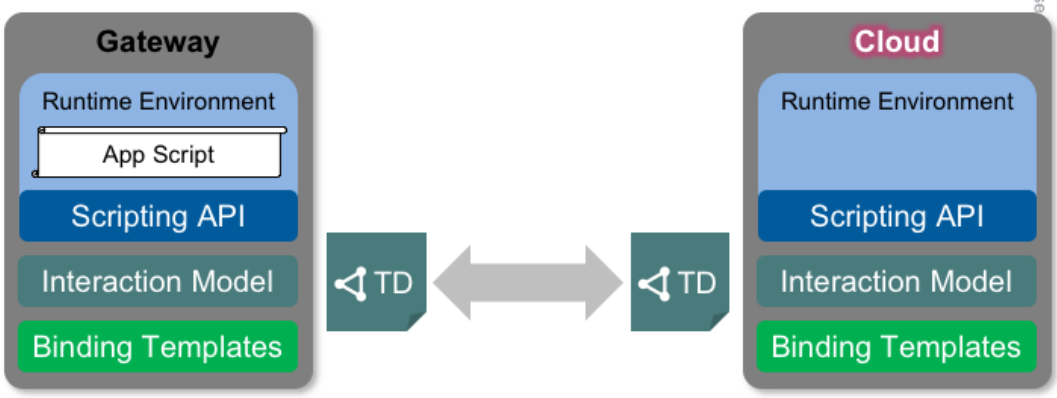
PORTABLE APPS ACROSS COMPONENTS



Fraunhofer FOKUS

91

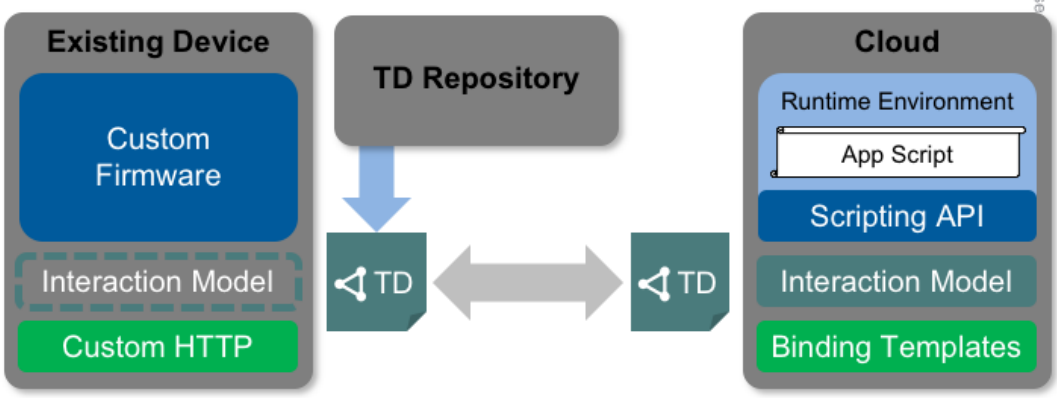
PORTABLE APPS ACROSS COMPONENTS



Fraunhofer FOKUS

92

TD TO AUGMENT EXISTING THINGS



Fraunhofer FOKUS

93

W3C WOT BUILDING BLOCKS

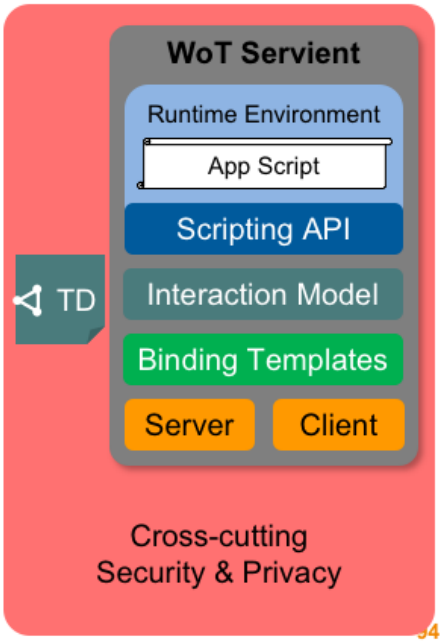
WoT Scripting API:
A standardized API to simplify IoT application development and enable portable scripts across vendors and device, gateway, and cloud platforms. The API allows to expose and consume Things according to the TD Interaction Model.

WoT Thing Description (TD):
Provides metadata of the interactions, data model, communication, as well as security mechanisms of the Thing. Using JSON-LD, the TD can be consumed by classic JSON parsers, but provides extension points for optional rich semantic tooling.

WoT Binding Templates:
The TD also describes the usage of protocols. A vanilla protocol stack can be configured at runtime to produce message that will be understood by the targeted Thing (cf. different HTTP APIs or OCF, oneM2M, and LWM2M dialects of CoAP).

Security & Privacy:
W3C WoT does not invent new mechanisms, but ensures that all building blocks provide means to describe the security and privacy mechanisms used in a specific platform and provides adversary testing of Things.

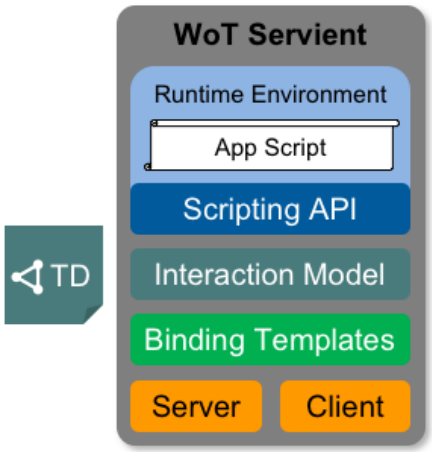
Fraunhofer FOKUS



Based on W3C WoT Intro slides

SERVIENT REFERENCE ARCHITECTURE

- Most building blocks are optional (e.g., WoT Scripting API)
- **Minimum requirement is is to provide a valid WoT TD** for the implemented protocols and data/interaction model



Fraunhofer FOKUS

95

- The Things Description (TD) is a good handover to the “data” topic

W3C WEB OF THINGS

Questions?