

INDUSTRIAL INTERNET OF THINGS (IIOT)

PART 4: COMMUNICATION



AV Lecture in Summer Term 2018

Dr.-Ing. Alexander Willner, Ronald Steinke



THE LAST LECTURE

Skipped

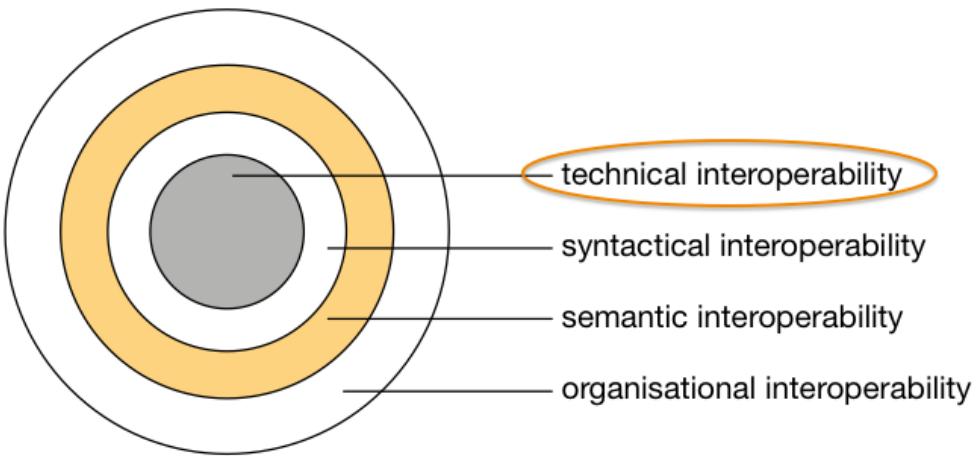


- This lecture is divided into 4 different areas
- Now, we focus on the second one

COMMUNICATION OVERVIEW

30 minutes

DIFFERENT LEVELS OF INTEROPERABILITY



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

OSI LAYERS 2 + 3

Application

Presentation

Session

Transport

Network

ICMP (Internet Control Message Protocol)

IP (Internet Protocol) address

ARP (Address Resolution Protocol)

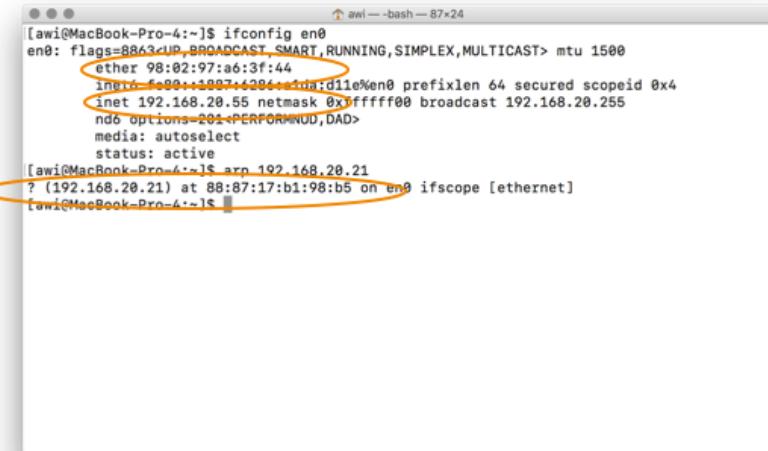
MAC (Media Access Control) address

Data Link

Physical

- **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)

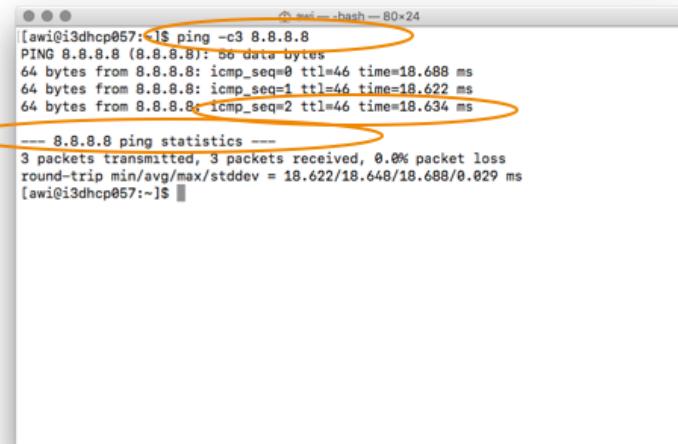


```
[awi@MacBook-Pro-4:~]$ ifconfig en0
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    ether 98:97:a6:3f:44
    inet 192.168.20.55 netmask 0xffffffff broadcast 192.168.20.255
        nd6 options=201<PERFORMNUD,DAD>
        media: autoselect
        status: active
[awi@MacBook-Pro-4:~]$ arp 192.168.20.21
? (192.168.20.21) at 88:87:b1:98:b5 on en0 ifs scope [ethernet]
[awi@MacBook-Pro-4:~]$
```

The terminal window shows the output of the 'ifconfig' command for the 'en0' interface. The MAC address '98:97:a6:3f:44' and the IP address '192.168.20.55' are circled in orange. Below it, the output of the 'arp' command is shown, where the MAC address '88:87:b1:98:b5' is also circled in orange.

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

EXAMPLE (ICMP)



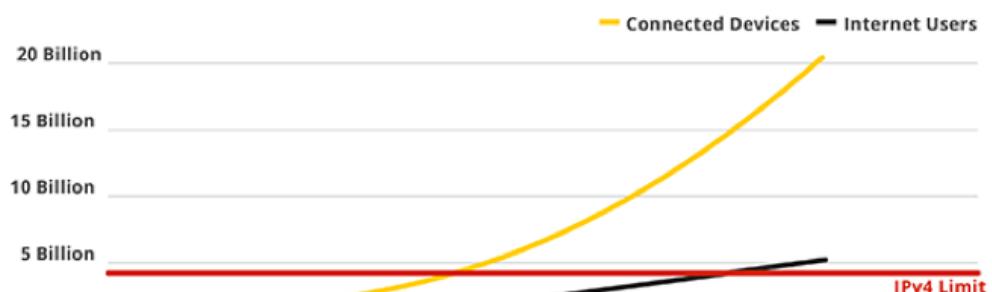
A screenshot of a terminal window titled "Terminal - hash - 80x24". The window displays the output of a ping command:

```
[awi@i3dhcp057: ~]$ ping -c3 8.8.8.8
PING 8.8.8.8 (8.8.8.8): 64 data bytes
64 bytes from 8.8.8.8: icmp_seq=0 ttl=46 time=18.688 ms
64 bytes from 8.8.8.8: icmp_seq=1 ttl=46 time=18.622 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=46 time=18.634 ms

--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 18.622/18.648/18.688/0.029 ms
```

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

IPV4 LIMIT



Fraunhofer
FOKUS

© Fraunhofer FOKUS

Google

- 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

OSI LAYER 3

Application

Presentation

Session

Transport

Network

IP (Internet Protocol) address v4 vs. v6

Data Link

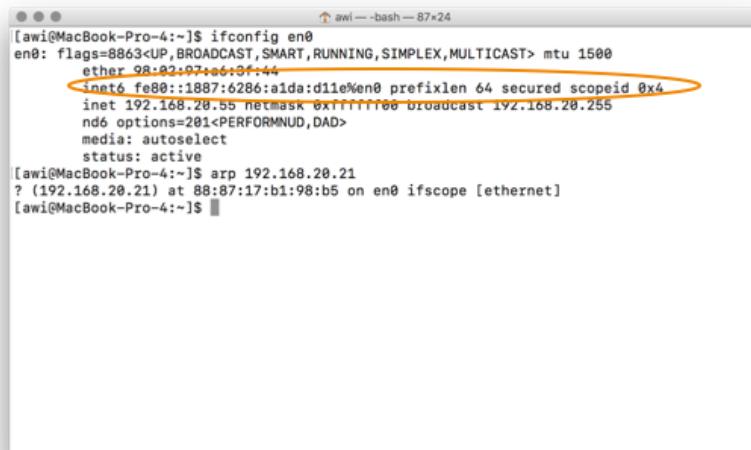
Physical



10

- **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.”

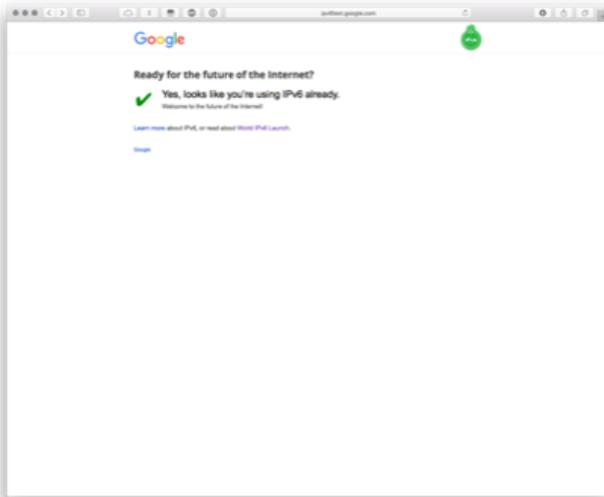
EXAMPLE (IPV6)



```
[awi@MacBook-Pro-4:~]$ ifconfig en0
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
      ether 98:02:07:66:01:14
      inet6 fe80::1887:6286:1ida:d11e%en0 prefixlen 64 secured scopeid 0x4
        inet 192.168.20.55 netmask ffffff00 broadcast 192.168.20.255
          nd6 options=201<PERFORMNUD,DAD>
          media: autoselect
          status: active
[awi@MacBook-Pro-4:~]$ arp 192.168.20.21
? (192.168.20.21) at 88:87:17:b1:98:b5 on en0 ifs scope [ethernet]
[awi@MacBook-Pro-4:~]$
```

- 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

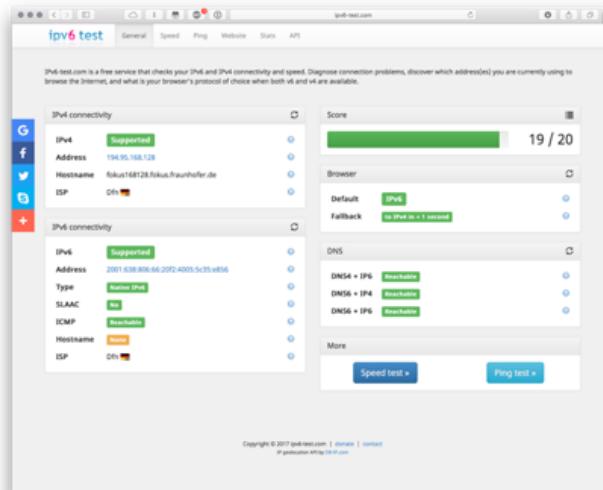


12

© Fraunhofer FOKUS

<http://ipv6test.google.com/>

IPV6 TESTS



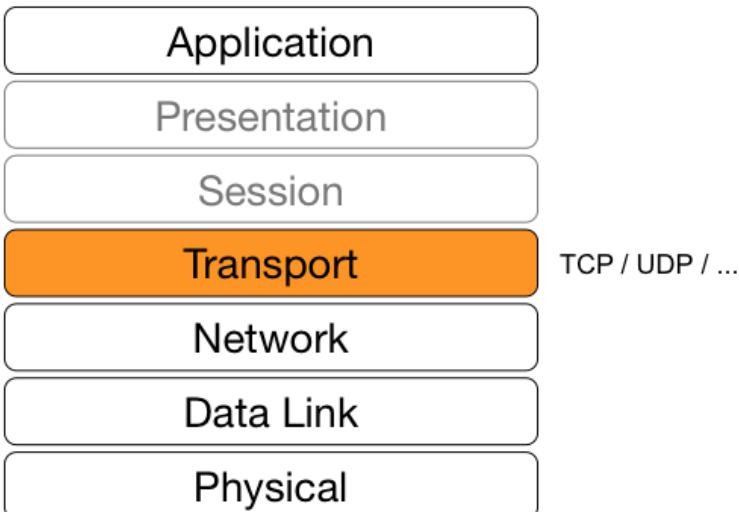
Fraunhofer
FOKUS

13

© Fraunhofer FOKUS

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

OSI LAYERS 4



Open Systems Interconnection (OSI) Layers

- **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)



The screenshot shows a terminal window titled "awi — telnet 192.168.20.21 80 — 87x24". It displays the output of several commands:

```
[awi@MacBook-Pro-4:~]$ ifconfig en0
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    ether 98:02:97:a6:3f:44
    inet6 fe80::180:2ff:fea6:3f44 prefixlen 64 secured scopeid 0x4
        inet 192.168.20.70 netmask 0xffffffff broadcast 192.168.20.255
            nd6 options=201<PERFORMNUD,DAD>
        media: autoselect
        status: active
[awi@MacBook-Pro-4:~]$ arp 192.168.20.21
? (192.168.20.21) at 00:0c:17:b1:98:b5 on en0 ifscope [ethernet]
[awi@MacBook-Pro-4:~]$ telnet 192.168.20.21 80
Trying 192.168.20.21...
Connected to 192.168.20.21.
Escape character is '^'.
```

A large orange oval highlights the command "arp 192.168.20.21" and its output, which shows the MAC address of the target host.

- Simple example

OSI LAYER 7

Application

HTTP, CoAP, MQTT, AMQP,
WebSockets, XMPP, ...

Presentation

Session

Transport

Network

Data Link

Physical

Open Systems Interconnection (OSI) Layers

- 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)

```
awi ~ -bash — 80x24
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    ether 98:02:97:a6:3f:44
    inet6 fe80::1887:6286:afda:d11e%en0 prefixlen 64 secured scopeid 0x4
        inet 192.168.20.70 netmask 0xffffffff broadcast 192.168.20.255
            nd6 options=201<PERFORMNUD,DAD>
            media: autoselect
            status: active
[awi@MacBook-Pro-4:~]$ arp 192.168.20.21
? (192.168.20.21) at 88:87:17:b1:98:05 on en0 ifscope [ethernet]
[awi@MacBook-Pro-4:~]$ telnet 192.168.20.21 80
Trying 192.168.20.21...
Connected to 192.168.20.21.
Escape character is '^'.
GET /
HTTP/1.1 400 Bad Request
MIME-Version: 1.0
Server: KS_HTTP/1.0
Content-Type: text/html
Transfer-Encoding: chunked
Connection: Close
<HTML>
<HEAD>
```

- 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)

```
[awi@i3dhcp057:~]$ curl -I google.com
HTTP/1.1 302 Found
Cache-Control: private
Content-Type: text/html; charset=UTF-8
Referrer-Policy: no-referrer
Location: http://www.google.de/?gfe_rd=cr&ei=WW4VWb-iNIfSXrWUmpgE
Content-Length: 256
Date: Fri, 12 May 2017 08:12:09 GMT
```

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

EXAMPLE (HTTP)

```
[awi@i3dhcp057:~]$ curl google.com
<HTML><HEAD><meta http-equiv="content-type" content="text/html; charset=utf-8">
<TITLE>302 Moved</TITLE></HEAD><BODY>
<H1>302 Moved</H1>
The document has moved
<A HREF="http://www.google.de/?gfe_rd=cr&ei=ZW4VWbKBPJTSXvuOgegG">here</A>.
</BODY></HTML>
[awi@i3dhcp057:~]$
```

- 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

- 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

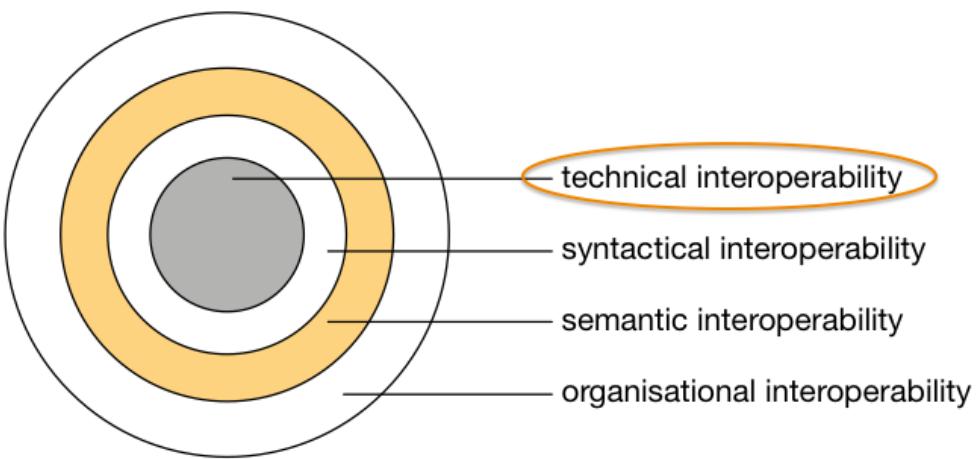
Questions?

MIDDLEWARE OVERVIEW

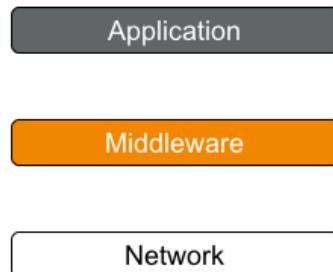
15 minutes

DIFFERENT LEVELS OF INTEROPERABILITY

ETSI White Paper: Achieving technical interop.



TYPICAL NETWORK AND DISTRIBUTION ABSTRACTION

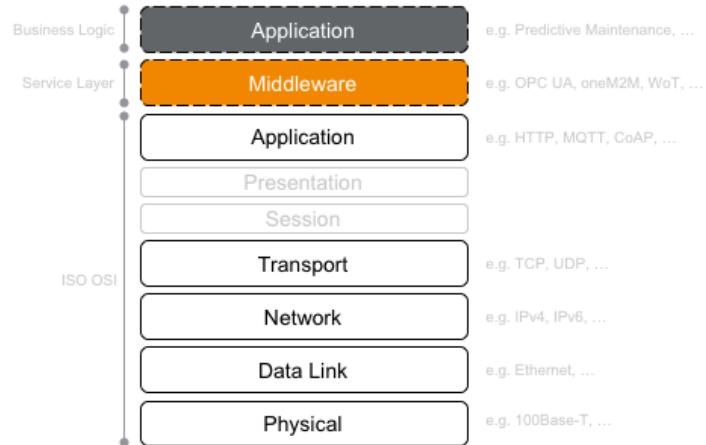


Reminder:

- 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.

EXTENDED NETWORK AND DISTRIBUTION ABSTRACTION

Open Systems Interconnection (OSI) Layers



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.

MIDDLEWARE

A LAYER BETWEEN APPLICATIONS AND THE OS

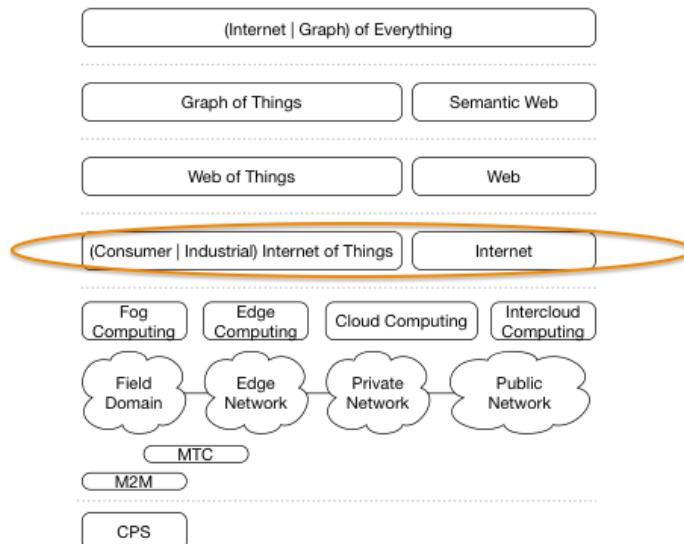
A **Middleware** in the context of distributed applications is software that provides services beyond those provided by the operating system to enable the various components of a distributed system to communicate and manage data.

[[https://en.wikipedia.org/wiki/Middleware_\(distributed_applications\)](https://en.wikipedia.org/wiki/Middleware_(distributed_applications))].

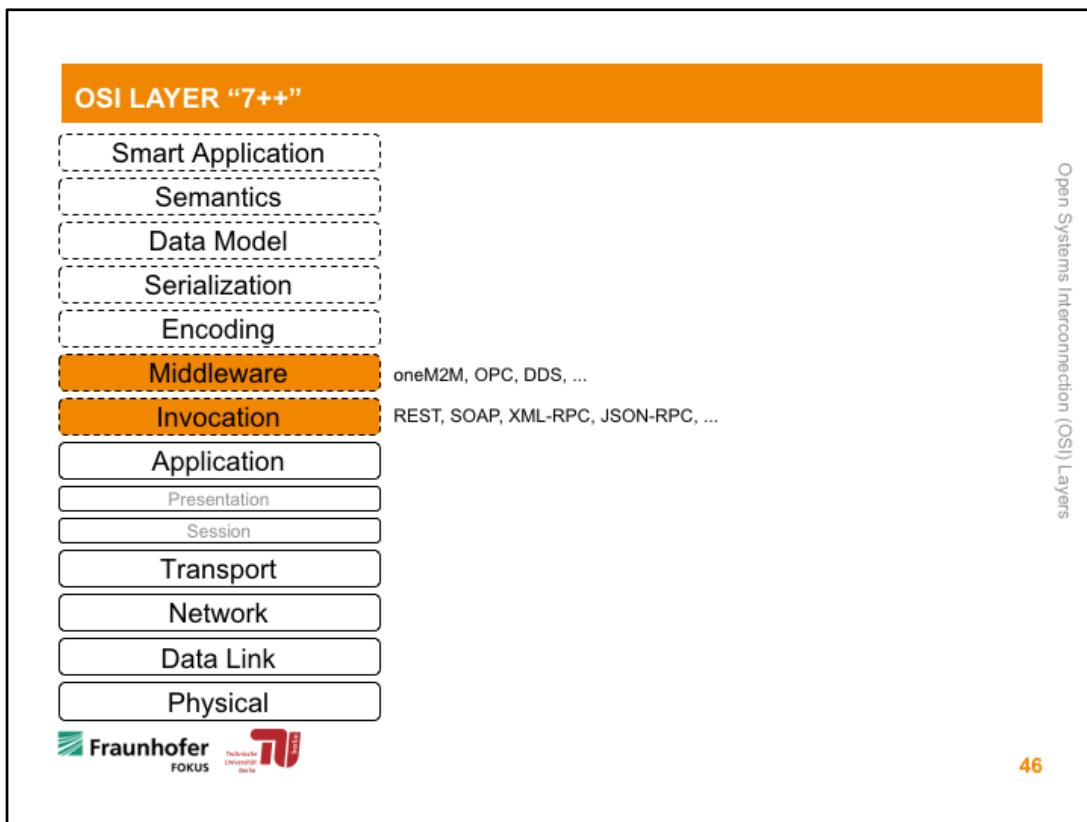


- Not clearly defined
- It simplifies complex distributed applications.
- Often enables interoperability
- OS = Operating System

INDUSTRIAL INTERNET OF THINGS

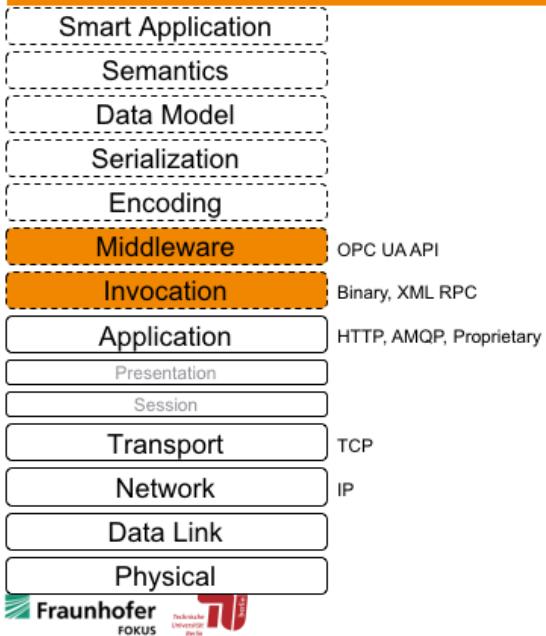


- Internet of Things → communication using Internet technologies



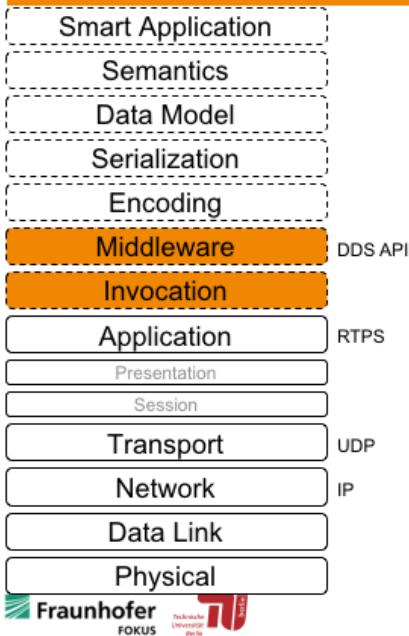
- **Note:** there are no OSI layers above layer 7 – “layer 8” sometimes denoted as the “political” layer
- **Invocation:** defines how information can be retrieved or modified (see Remote Procedure Calls (RPC))
- **Middleware:** provides a set of communication services for distributed applications

OPEN PLATFORM COMMUNICATIONS UNIFIED ARCHITECTURE



Background: Automation.
Application: Vertical.

DATA-DISTRIBUTION SERVICE API

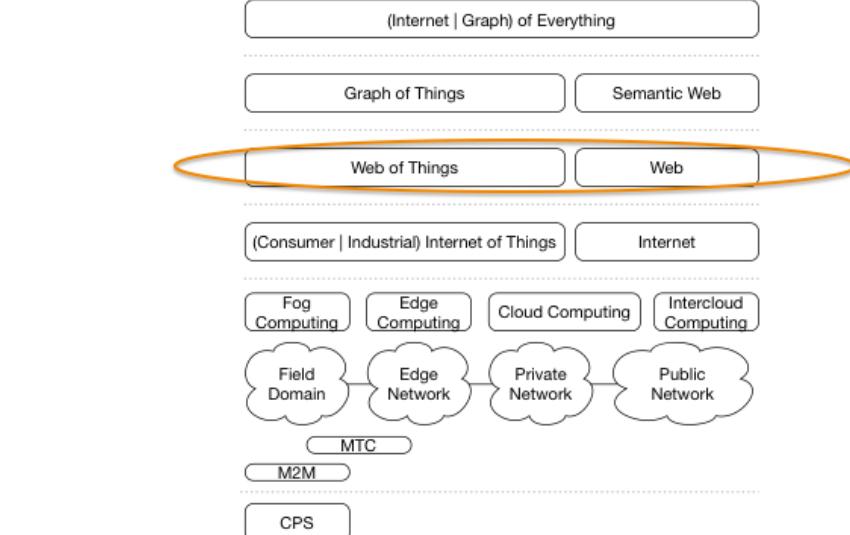


Background: Defense.
Application: Horizontal.

48

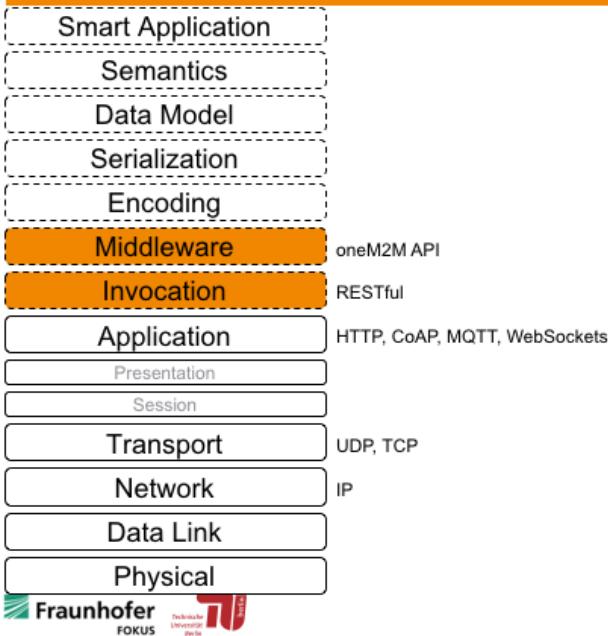
- Real-time Publish-Subscribe Wire Protocol

WEB OF THINGS



- Web of Things → communication using Web technologies

ONEM2M



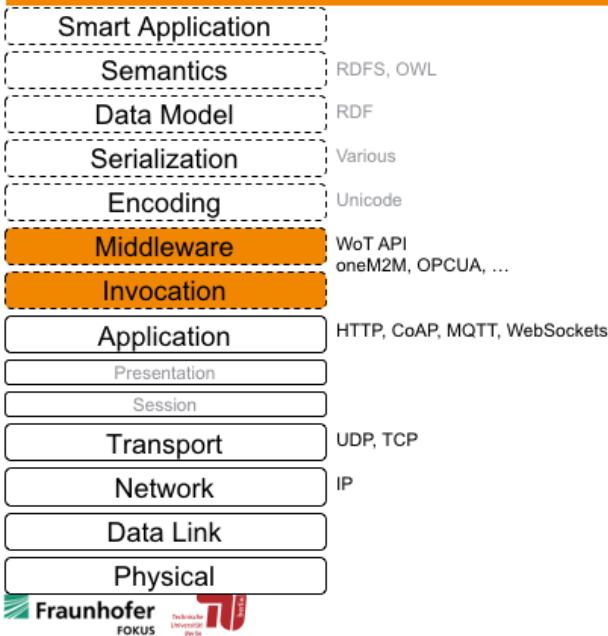
Fraunhofer
FOKUS
Technische Universität Berlin



Background: Telecom.
Application: Horizontal.

50

W3C WEB OF THINGS (WOT)



Background: lack of interoperability across platforms

51

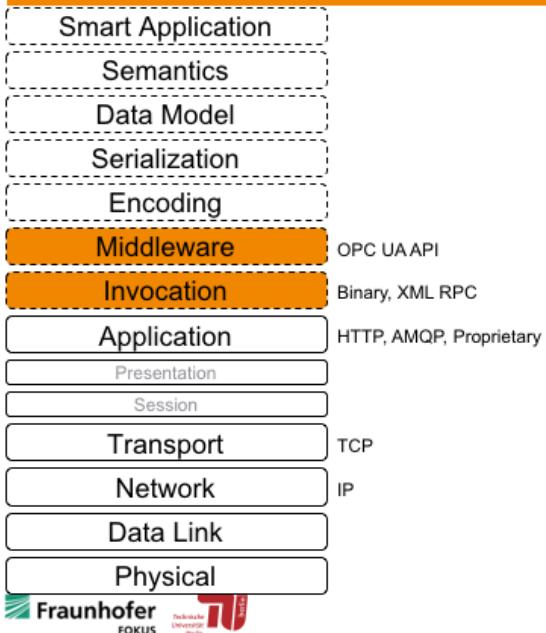
MIDDLEWARE OVERVIEW

Questions?

OPC

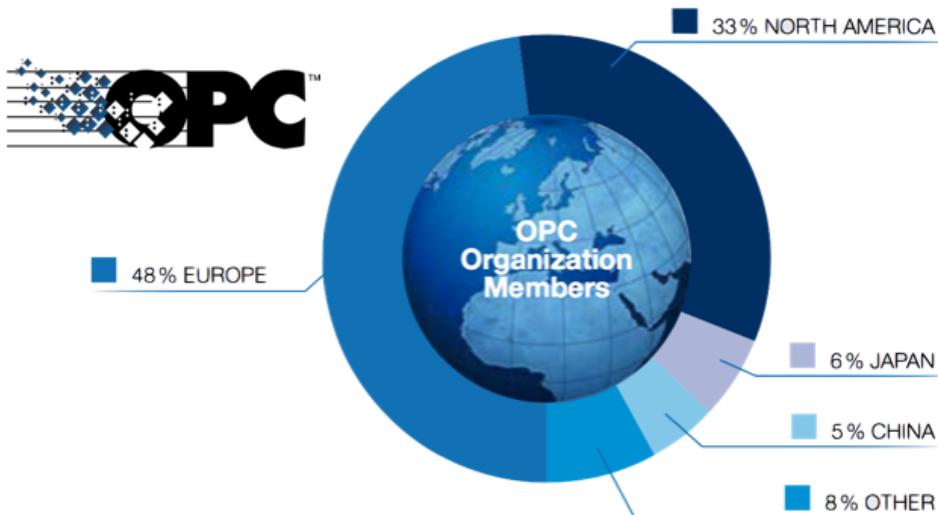
15 minutes

OPEN PLATFORM COMMUNICATIONS UNIFIED ARCHITECTURE



Background: Automation.
Application: Vertical.

OPC FOUNDATION



Fraunhofer
FOKUS

55

- OPC UA members mostly in Europe and US/Canada

OPC FOUNDATION - A GLOBAL NON-PROFIT ORGANIZATION



Microsoft



SAP



SIEMENS

Rexroth
Bosch Group

ORACLE

YOKOGAWA

FESTO

ThyssenKrupp

Fraunhofer
FOKUS

TU
Berlin

56

- ~ 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

Before

HMI #A		HMI #B	
Modbus	Profinet	Modbus	Profinet
Profinet	DH+	DH+	CIP
FF	CIP	FF	Bacnet
EGD	Bacnet	EGD	Bacnet
DNP	SNMP	DNP	SNMP
TSAA	AS511	TSAA	AS511
UDC	Others...	UDC	Others...



After

HMI #A		HMI #B	
OPC	OPC	OPC	OPC

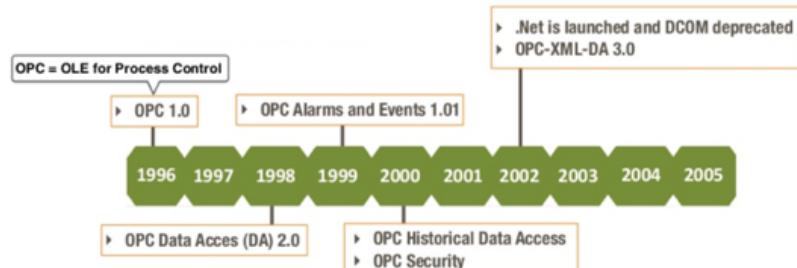
OPC Server



- 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

HISTORY

Prism Tech



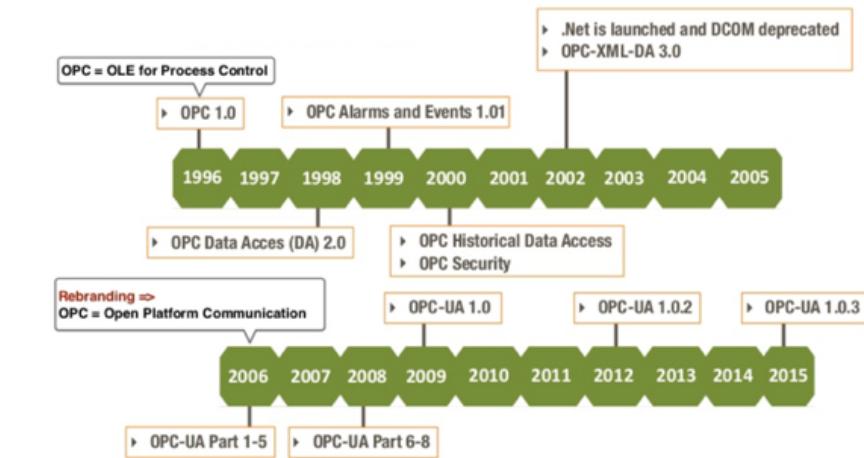
- **OLE** = Object Linking and Embedding (Microsoft technology to embed and link to documents)
- OPC is a standard interface to communicate between numerous data sources, including devices on a factory floor, laboratory equipment, test system fixtures, and databases.
- It utilizes the Microsoft-based 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

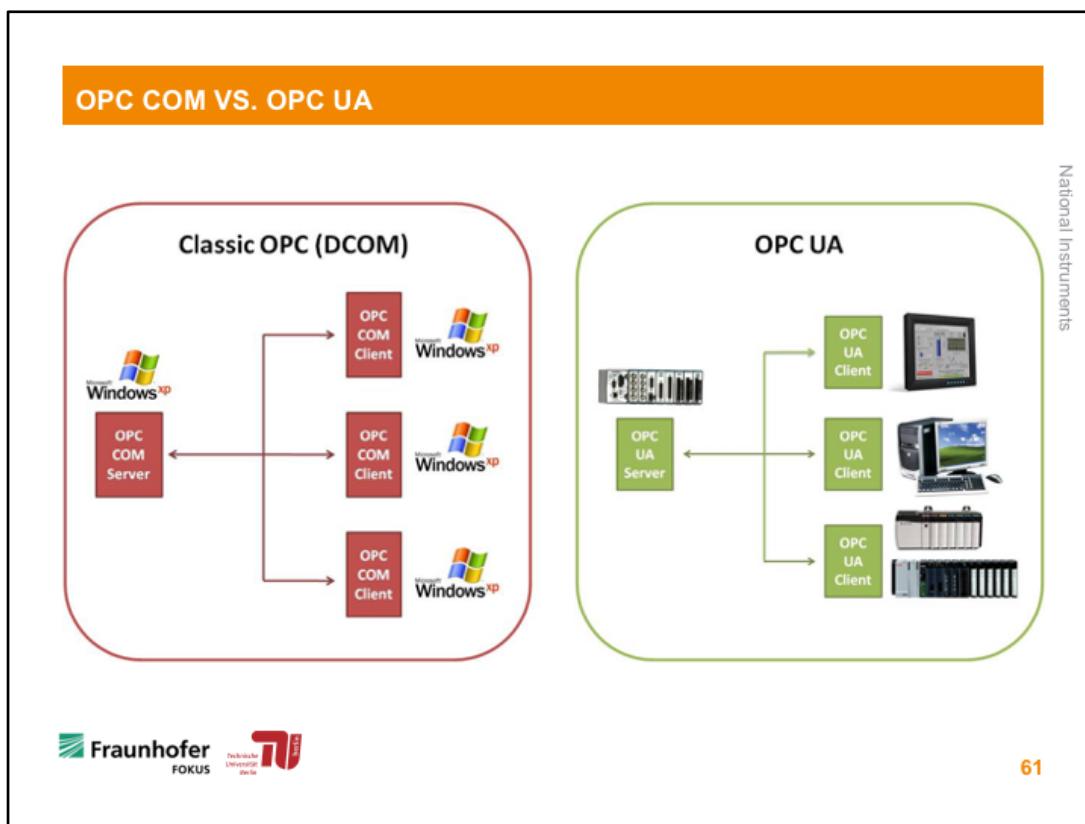
HISTORY (TWO DIFFERENT LINES OF DEVELOPMENT)

Prism Tech



- 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.

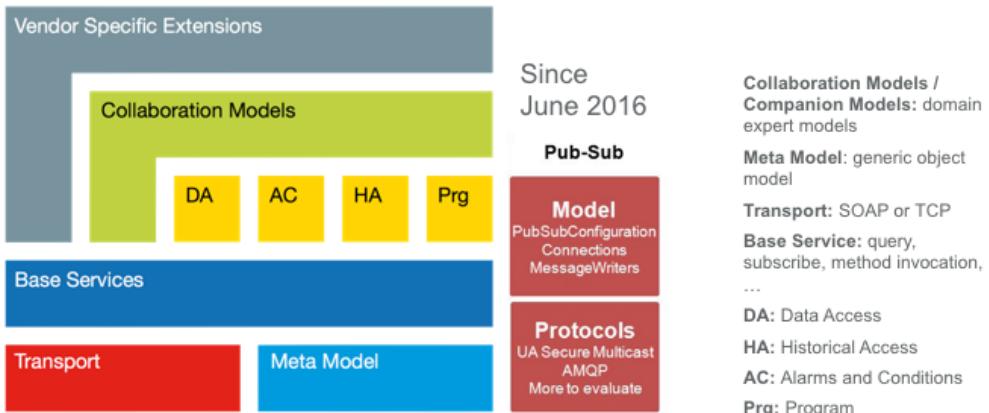
OPC COM VS. OPC UA



- 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 UA LAYER MODEL

OPC Foundation



- 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

FOKUS   

THREE OPC UA COMMUNICATION TYPES

	OPC UA WS	OPC UA Binary	OPC UA PubSub
Serialization	XML	Binary	JSON/XML/...
Middleware	SOAP		
Application		UPC UA	AMQP/MQTT/XMPP/...
Presentation	HTTP(S)		
Session		TCP	
Transport	TCP	TCP	TCP
Network	IP	IP	IP
Data Link	Data Link	Data Link	Data Link
Physical	Physical	Physical	Physical

OPERATION MODE 1: OPC CLIENT-SERVER CONCEPT

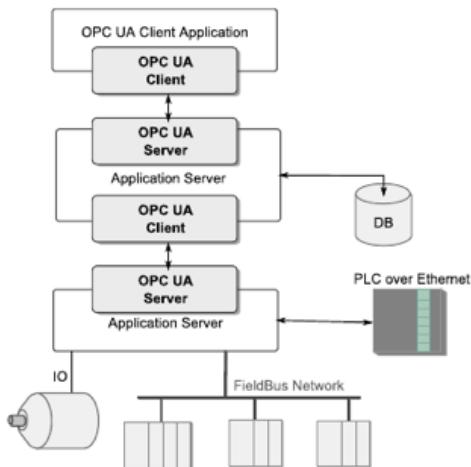
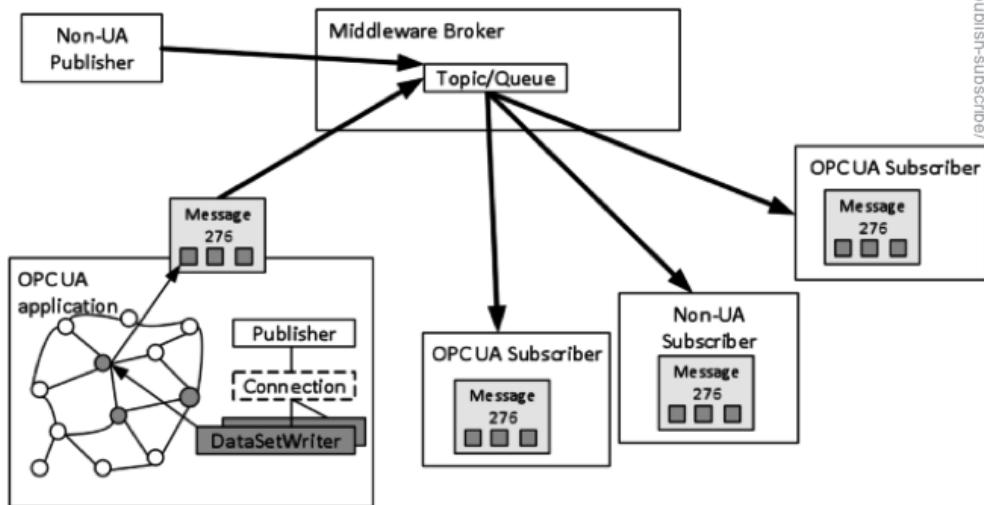


Figure 9: OPC UA Stacked Architecture

Source: <http://ithemanufacturingconnection.com/2016/06/opc-ua-publish-subscribe/>

- 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.

OPERATION MODE 1: OPC PUBSUB CONCEPT



Source: <http://themanufacturingconnection.com/2016/06/opc-ua-pub-sub/>

67

- 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.).