PROTOCOLS
INDUSTRIAL IOT

Software and Services Group
IoT Developer Relations, Intel

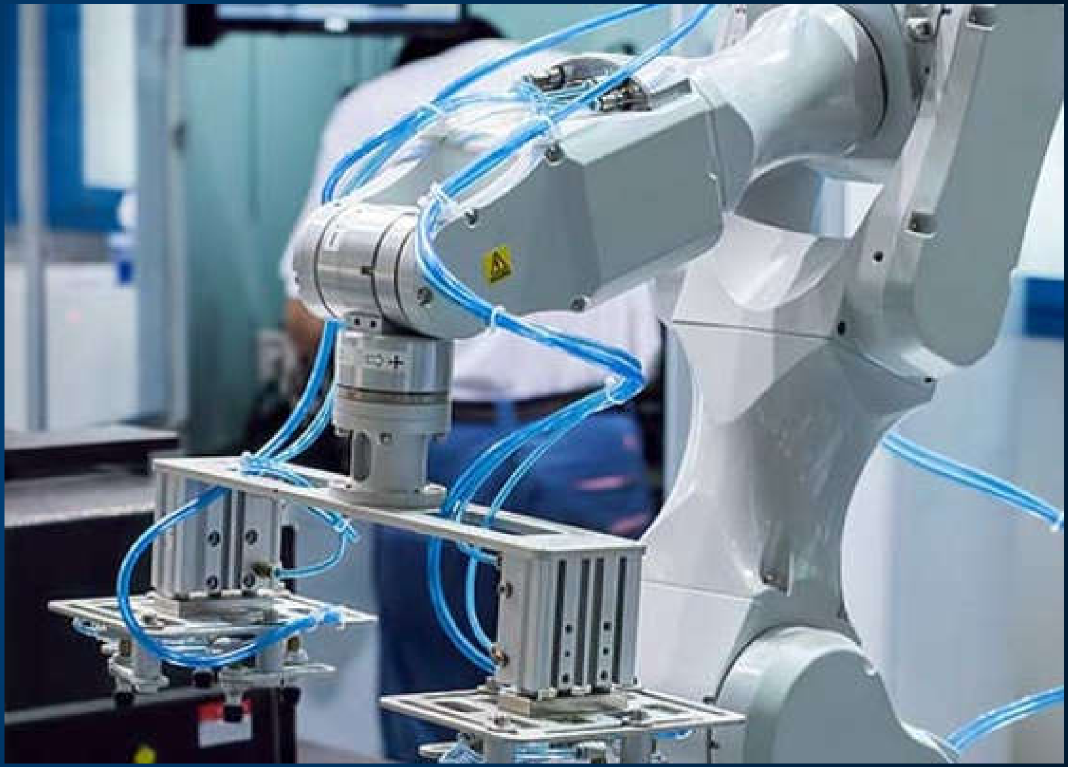
LEGAL NOTICES AND DISCLAIMERS

IIOT CONNECTIVITY CHALLENGE

The goal of the industrial internet is to enable seamless information sharing across domains and industries.

Past capital investments in equipment have created a myriad of domain specific connectivity technologies, tightly vertically integrated and optimized to solve domain specific needs.

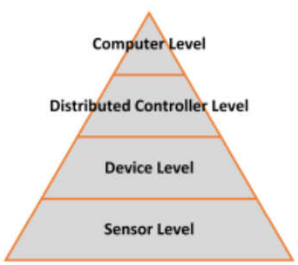
IIoT systems usually integrate with **brownfield** technologies to preserve the capital investments, and **greenfield** technologies to spur innovation.



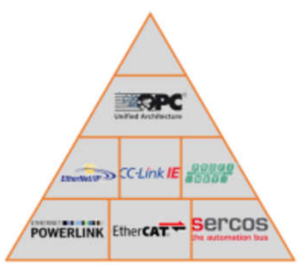
NETWORKING FOR AUTOMATION AND ROBOTICS

For machine-to-machine and process communication (the distributed controller level), the role **of OPC UA (IEC 62541) is rapidly increasing in significance alongside the traditional Ethernet-based M2M fieldbus systems** (PROFINET, EtherNet/IP, CC-Link IE).

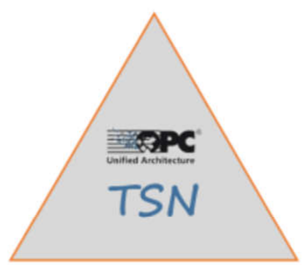
Although these technologies share common requirements, their implementations differ substantially. Hence, comparing them depends heavily on the intended application (process control, motion, I/O, centralized vs. decentralized control, etc.).



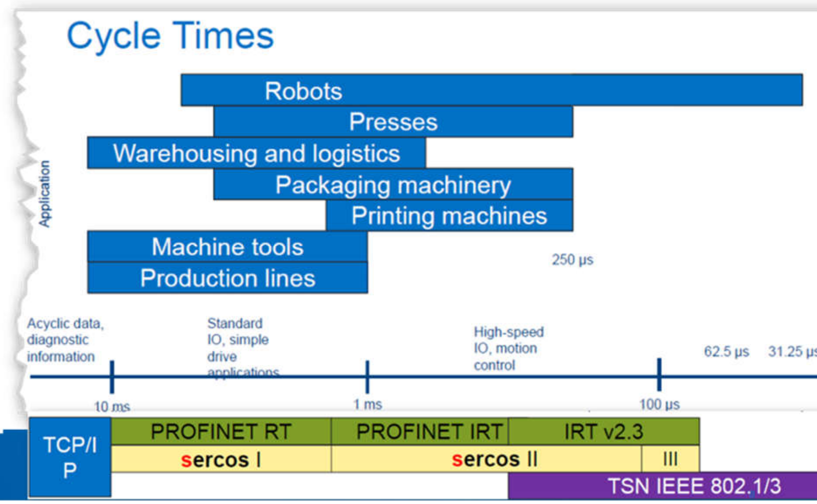
(a) Levels of the automation pyramid



(b) Today's distinct ecosystems in the automation pyramid

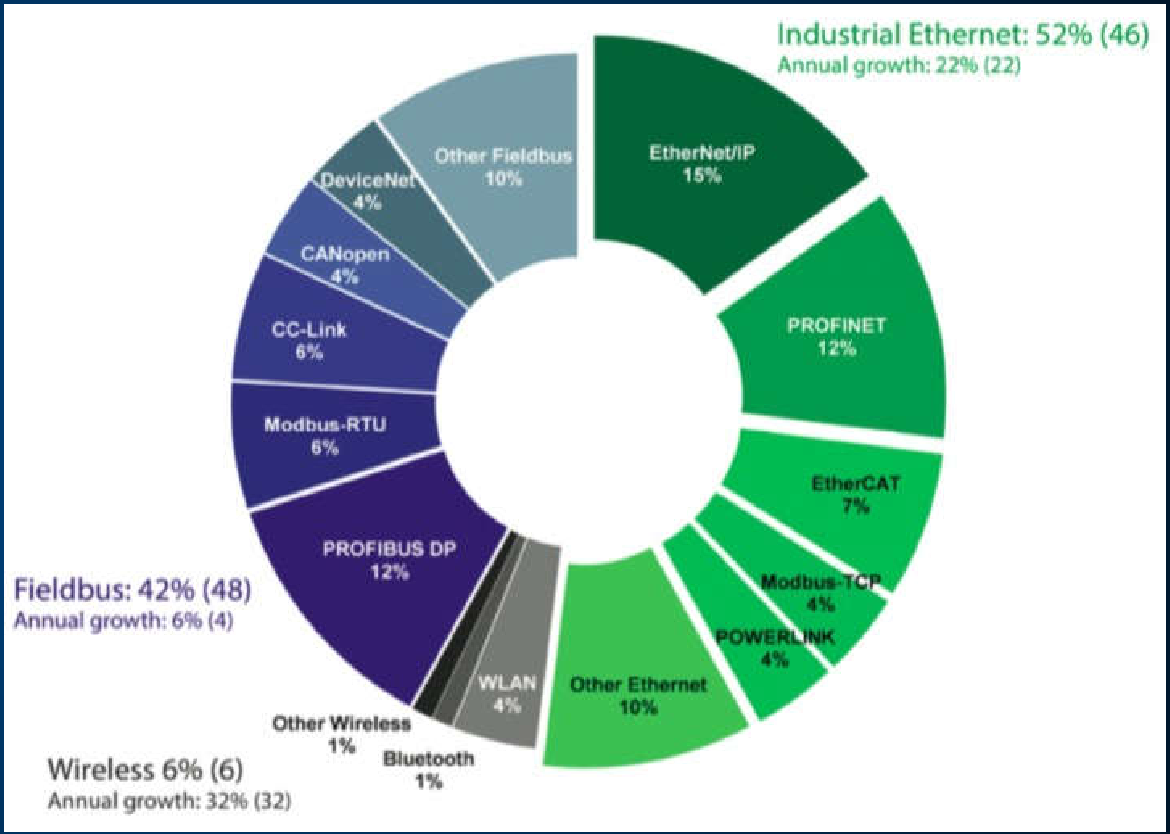


(c) Holistic communication from the sensor to the cloud using OPC UA (and TSN)



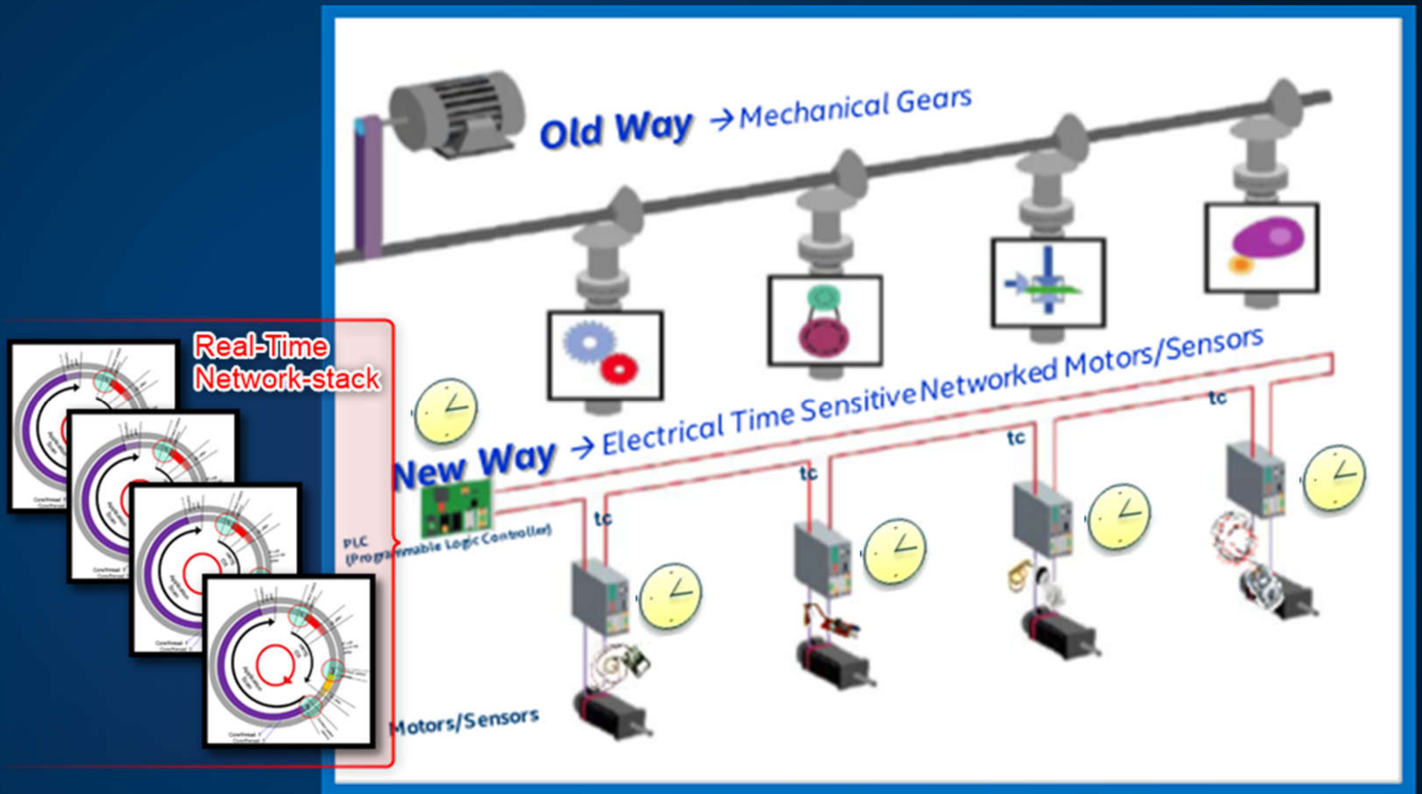
INDUSTRIAL ETHERNET MARKET SHARE

- Growth powered by IIoT (e.g OT/IT)
- Wireless redefining network picture
- Regional network variations :
  - EMEA - PROFINET and EtherNet/IP are dominant
  - US - Movement towards EtherNet/IP
  - Asia - no network stands out as truly market-leading: PROFINET, EtherNet/IP, PROFIBUS, EtherCAT, Modbus and CC-Link
- Fieldbuses expected to decline
  - PROFIBUS and Modbus still significant



NETWORKING FOR AUTOMATION AND ROBOTICS

- Time-Synchronized (IEEE 1588v2 or 802.1AS) real-time control.
- Down the wire updates
- Reduced OPEX
- Flexible function
- Reconfigurable production



MEASUREMENT OF REAL-TIME NETWORK CYCLE TIMES

TIME SENSITIVE NETWORKING (TSN)

Intel support for TSN

- Intel® Ethernet Controller I210
- FPGA
- Open Source contributions
- Active member (e.g. IEEE, IIC, AVNU alliance)
- IoTG/ISD OT Testbed

OPC UA – TSN

Rockwell hops aboard the OPC UA / TSN bandwagon

[Link](#)

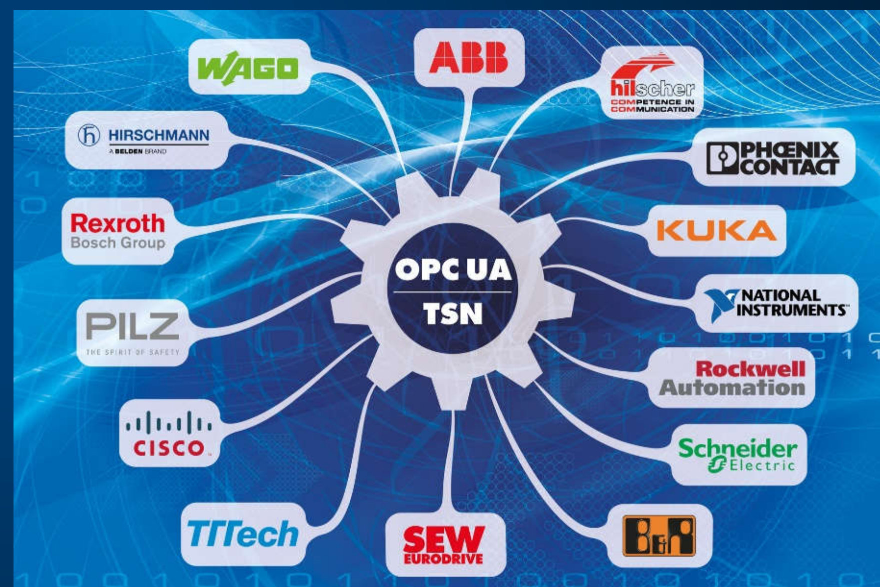
27 APRIL, 2018

HANNOVER MESSE

Rockwell Automation is the latest major automation player to join the group of suppliers that are backing the use of OPC UA for sharing information across multiple vendor systems, and TSN (time-sensitive networking) for improving the latency and robustness of converged industrial networks.

- Rapidly growing number of supporting companies ("Traffic-Shaper")
- Real-Time capable & multi-threaded
- IT/OT convergence

B&R Demo SPS drives, Nov 2017



Shaper Image

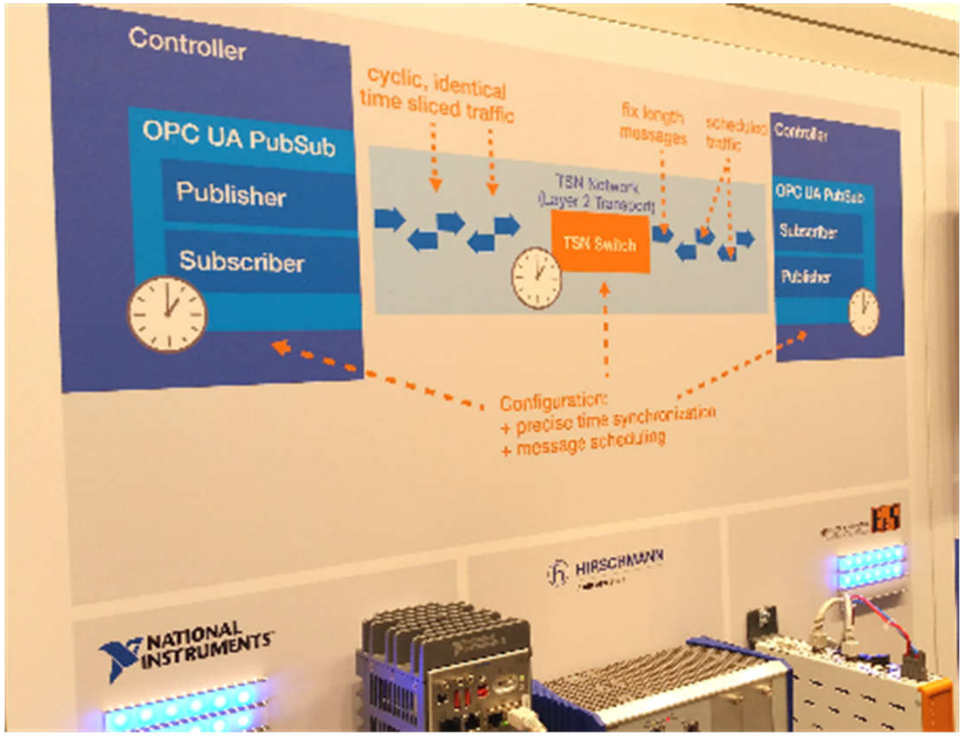
OPC UA PUB/SUB - TSN

GE (2014): “Standards like OPC-UA make communication between the PLC and HMI easier [...] “

04/06/2016: OPC Fdn Working Group to standardize **Publisher/Subscriber for OPC-UA**

03/27/2018: OPC Fdn **release OPC UA PubSub 1.4 spec.** and Industry KOLs acknowledging “ *[...] important basic milestone to be tunneled through [...] TSN*”

07/24/2018: … next episode **@TSN/A conference Stuttgart, Germany**.



OPC booth: Hannover Fair, May 2018

SOME IMPORTANT OPEN-INTEROPERABILITY TSN TEST

Focus

Next

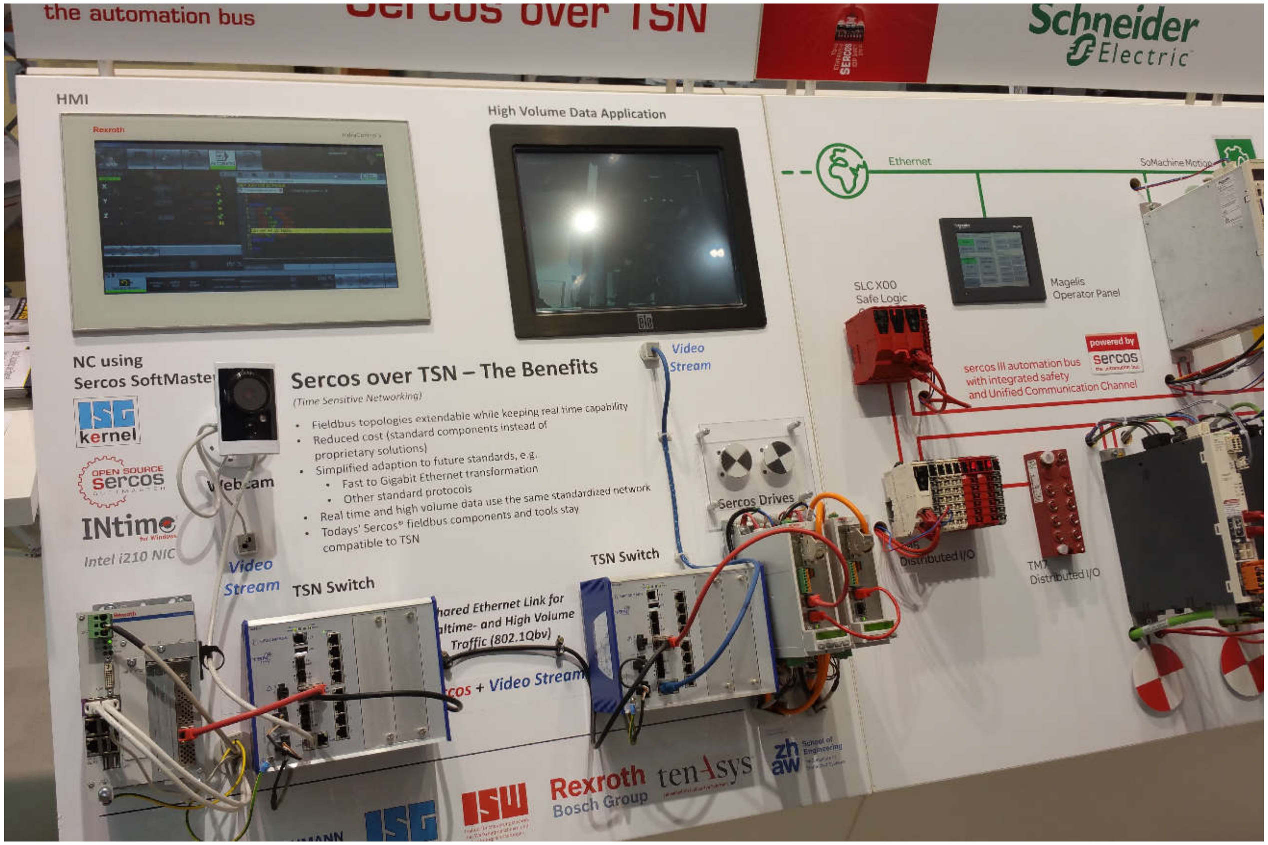
IEEE 802.1	Name	Category
AS (rev)	Timing & Synchronization	Time Synchronization
Qbv	Scheduled Traffic	
Qbu	Frame Pre-emption	Bounded Latency
Qch	Cyclic Queuing and Forwarding	
CB	Seamless Redundancy	
Qca	Path Control and Reservation	Redundancy & Reliability
Qci	Ingress Policing	
Qcc	Central Configuration (SRP)	Resource Management

FIELD-BUS OVER TSN – EXAMPLE: SERCOS OVER TSN

Brownfield installations

Taking advantage of TSN, e.g.

- Guaranteed Latency
- Low-jitter
- RT Traffic-Class & Bulk
  - IT/OT convergence
  - DSS cameras
  - Remote access



Sercos booth: Hannover Fair, May 2018

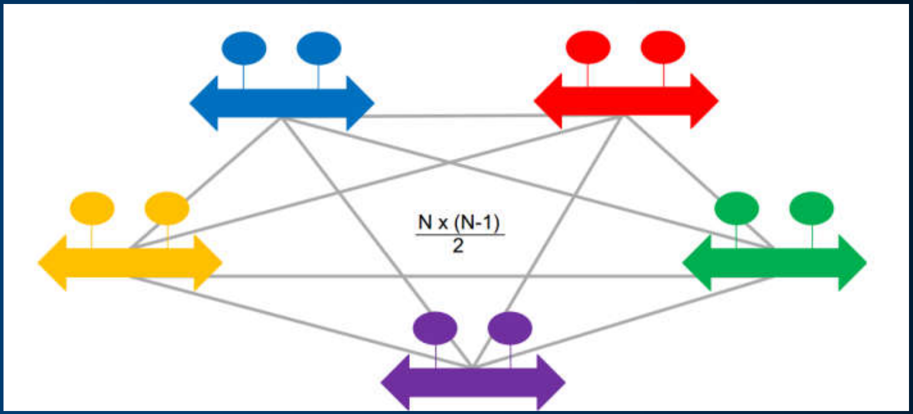
THE NEED FOR PROTOCOL TRANSLATION GATEWAYS

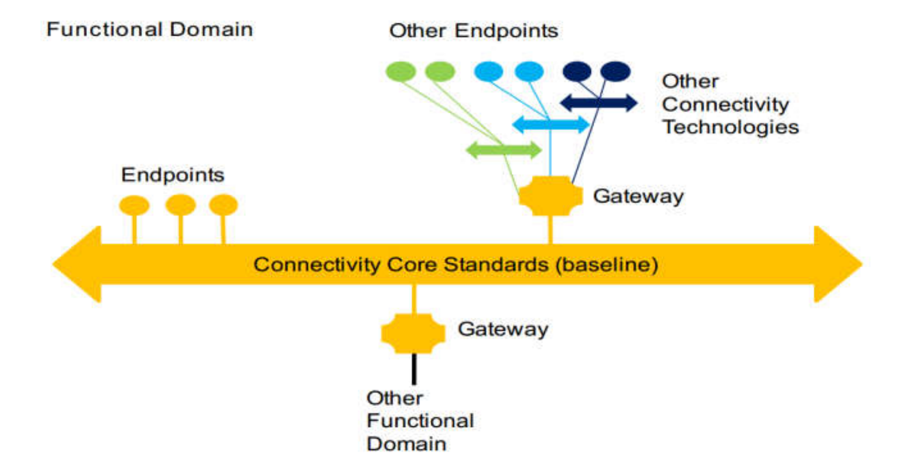
$$N \times \frac{N-1}{2} = O(N^2)$$

Complete connectivity rapidly becomes unmanageable.

To keep the connectivity architecture manageable, a connectivity technology standard is chosen as the baseline within a functional domain, and referred to as the "**connectivity core standard**"

To facilitate information exchange, one has to build bridges to each of the other connectivity technologies.





BRIDGING CORE CONNECTIVITY STANDARDS

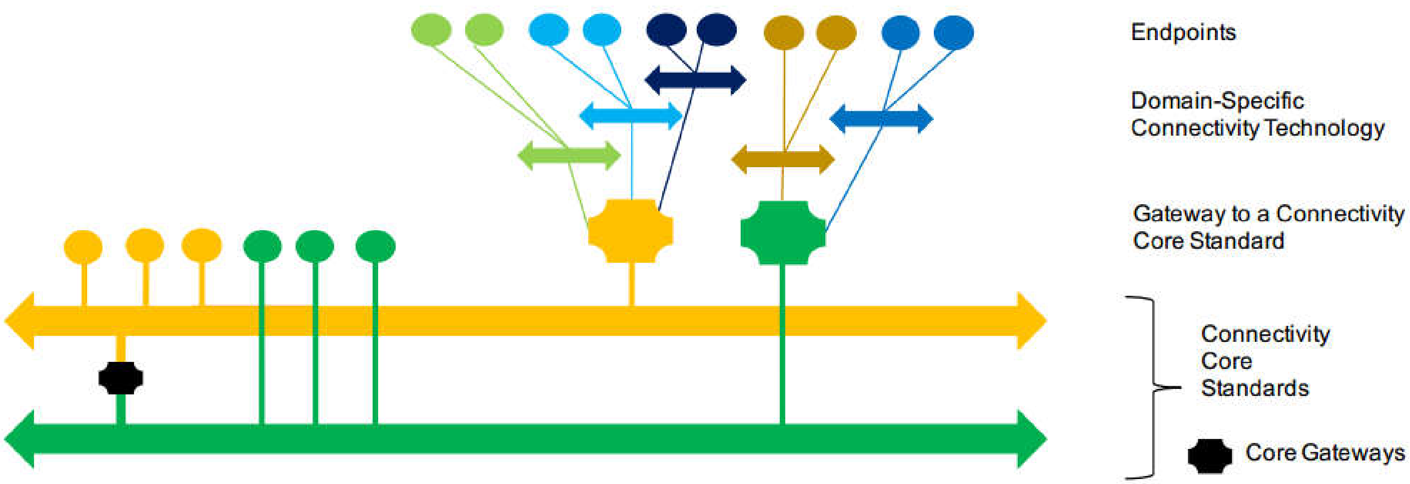


Figure 3-3: A standardized gateway between core connectivity standards can allow domain-specific endpoints connected to one core standard to communicate with domain-specific endpoints integrated over another core standard.

TOOLS: PROTOCOL ASSESSMENT TEMPLATE

The assessment template is intended to be a tool for understanding any connectivity technology in the context of the IIoT needs.

The worksheet is helpful for:

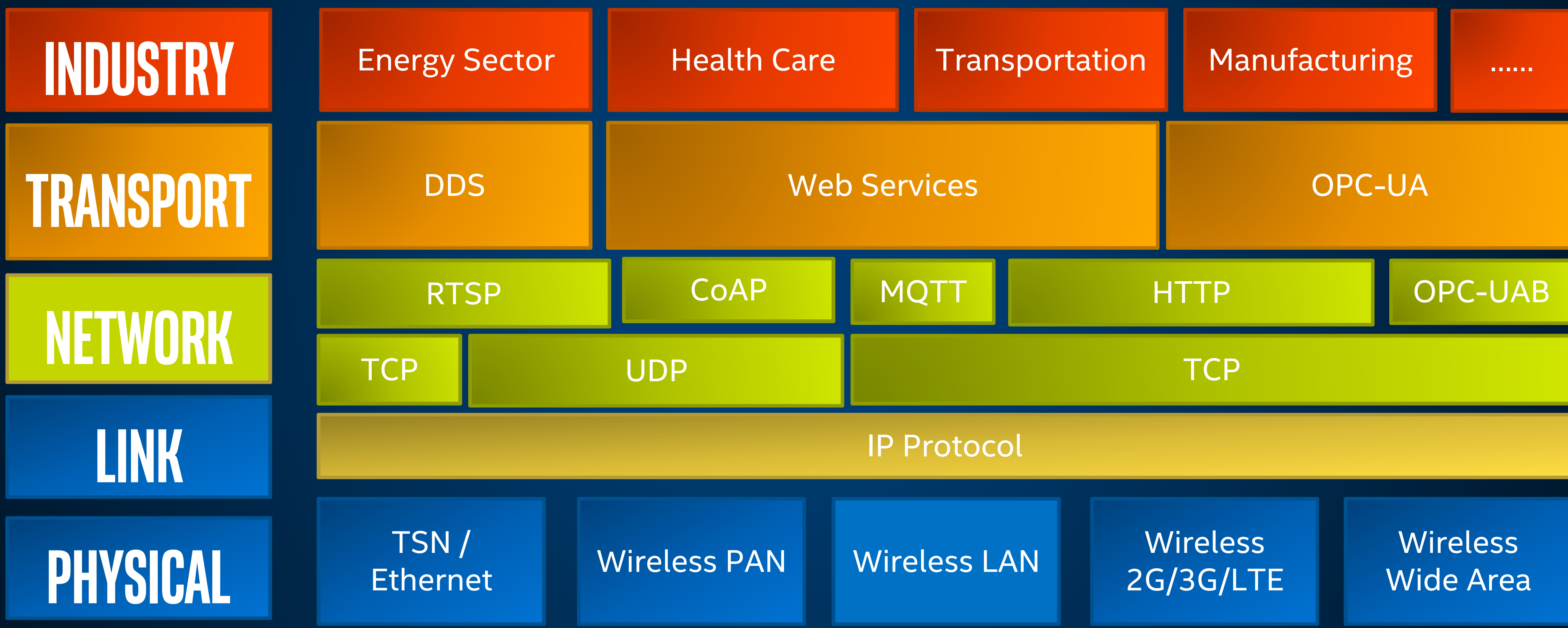
- understanding how a connectivity technology supports specific IIoT functional needs,
- evaluating a connectivity technology's trades-offs for typical IIoT considerations and
- determining a connectivity technology's suitability for a particular use case (once the specific requirements are understood).

Core Standard Criterion	Protocol Checklist
Provide **syntactic interoperability**	√
Open standard with strong **independent, international** governance	X
Horizontal in its applicability across industries	
Stable and **deployed** across multiple vertical industries	
Have **standards-defined *Core Gateways*** to all other core connectivity standards	
Meets connectivity **functional** requirements	
Meet **non-functional** requirements of performance, scalability, reliability, resilience	
Meet **security** and **safety** requirements	
Not require any single component from any single vendor	
Have readily-available SDKs both **commercial** and **open source**	

CONNECTIVITY FRAMEWORK CORE FUNCTIONS

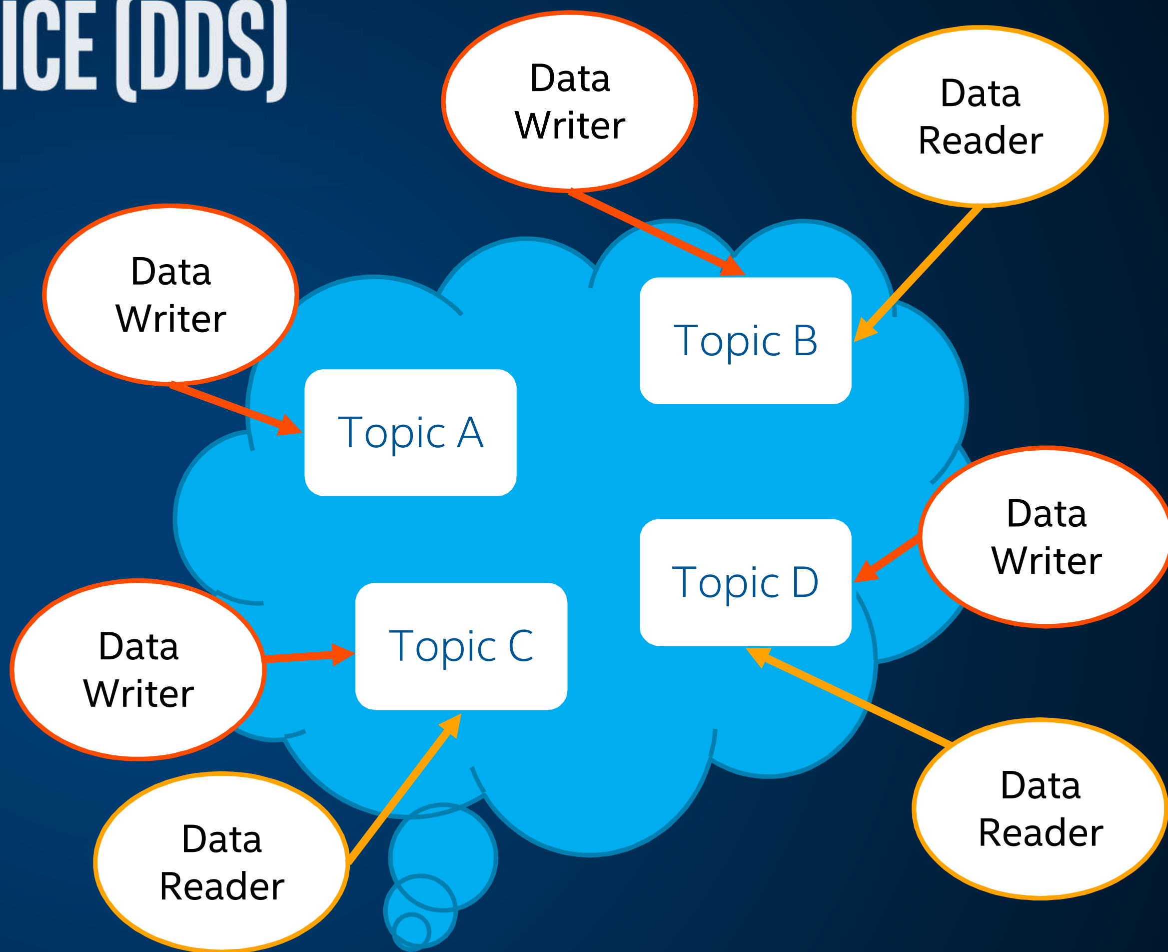
- Data Resource Model – represents structured data objects that can change state over time
- ID and Addressing – provides the means to identify and address each data object
- Data Type System – provides a way to describe the constrains place on data, includes a method of evolving and versioning the data syntax
- Publish/Subscribe – supports the well-know pubsub pattern for decoupled data exchange
- Request/Reply - supports the well-know request/reply pattern for data exchange
- Discovery – must be able to find pubsub services, request/reply services, endpoints and datatypes.
- Exception Handling – mechanisms for handling disconnections, changes in configuration or quality, endpoint failures, etc…
- Data Quality of Service – QoS method implemented, best-effort vs. reliable delivery
- Data Security – confidentiality, integrity, authenticity and non-repudiation of the data
- Data Governance – is there a standards body that directs this protocol's evolution

IIOT CONNECTIVITY STACK MODEL



DATA DISTRIBUTION SERVICE (DDS)

The **Data Distribution Service** for real-time systems (**DDS**) is an Object Management Group (OMG) machine-to-machine(sometimes called middleware) standard that aims to enable scalable, real-time, dependable, high-performance and interoperable data exchanges using a publish–subscribe pattern. DDS addresses the needs of applications like financial trading, air-traffic control, smart grid management, and other big data applications. The standard is used in applications such as smartphone operating systems,[1] transportation systems and vehicles,[2] software-defined radio, and by healthcare providers. DDS was promoted for use in the Internet of things.[3]

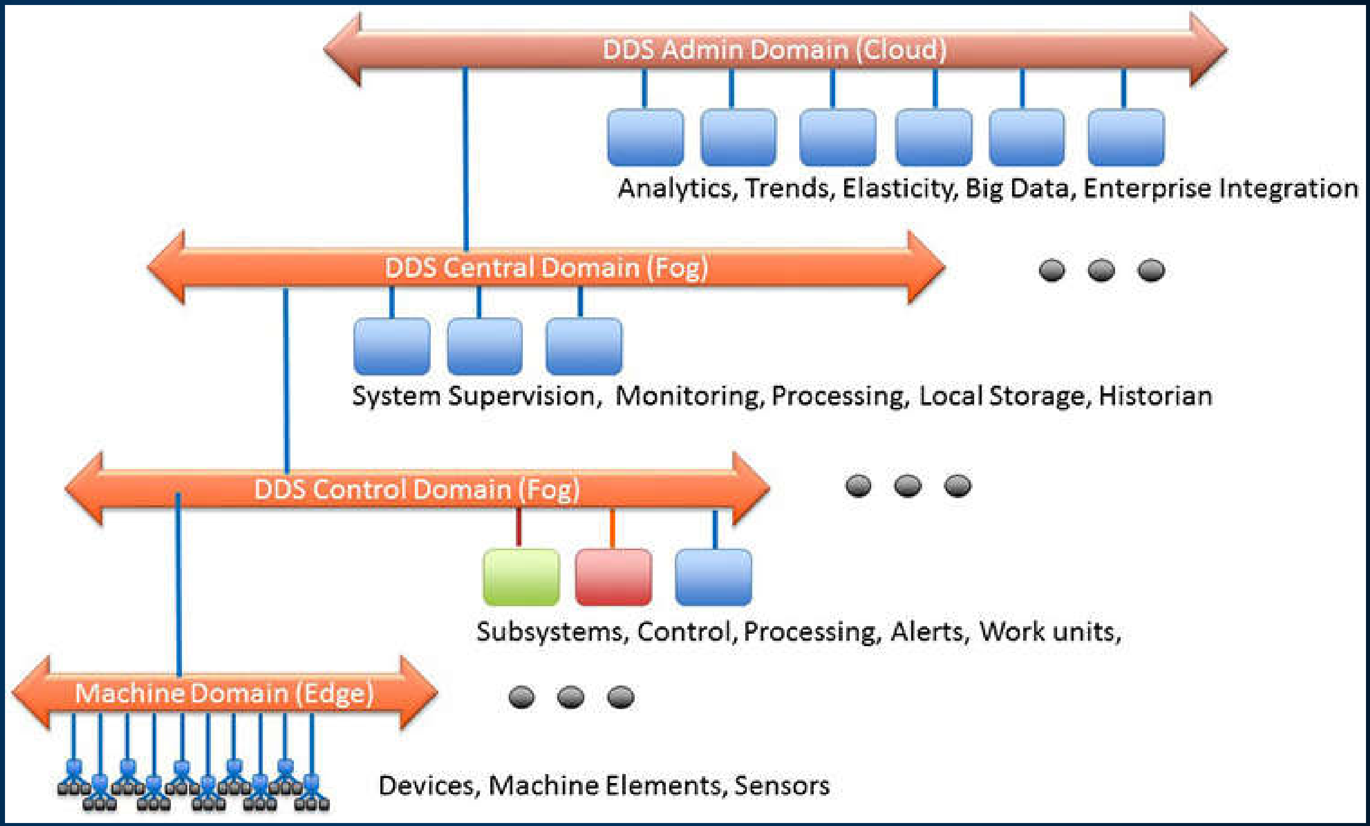


PROPERTIES OF DDS

Quality of Service

Dynamic Discovery

Scalable Architecture



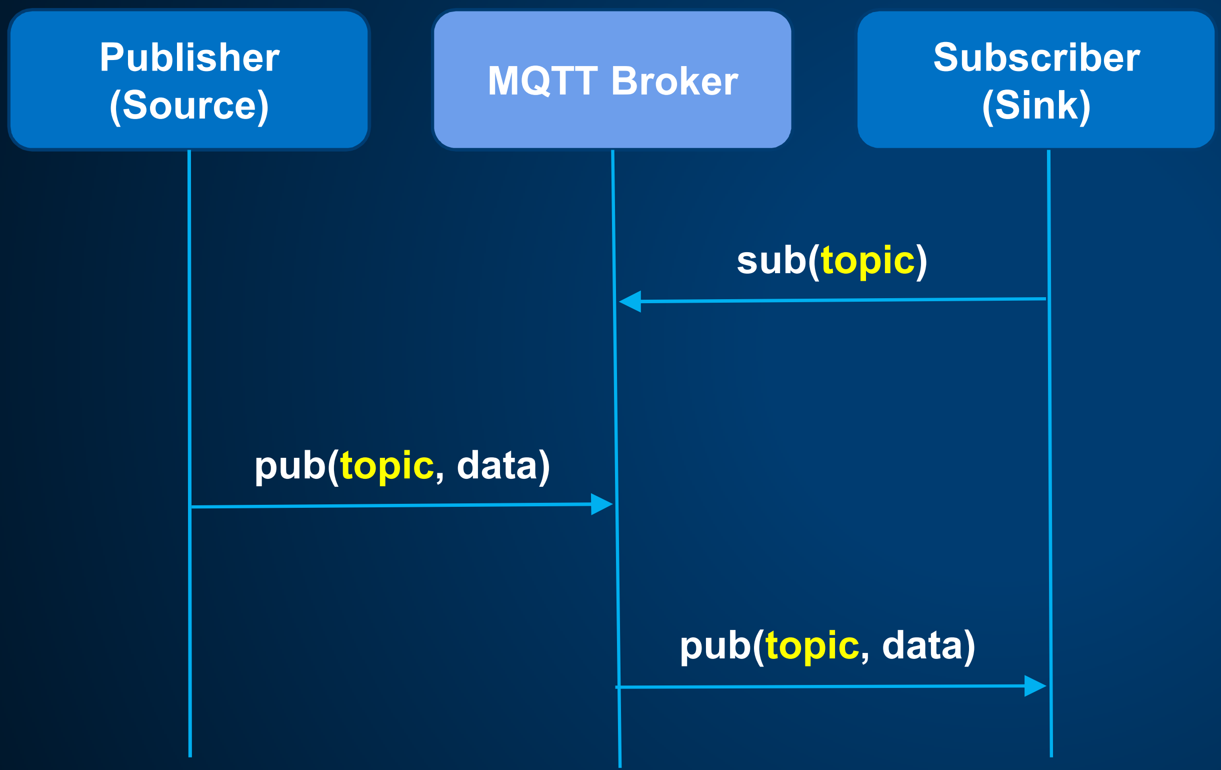
DDS: FUNCTIONAL SUMMARY

DDS has been applied in multiple verticals to realize higher domain-specific interoperablity

open architecture specifications. These include:
• SGIP OpenFMB v1.0 (uses CIM extensions over DDS) - NAESB Standard
• MDPnP OpenICE Integrated Clinical Environment for Medical Device
Interoperability
• ROS: Robot Operating System (Open Source)
• EUROCAE ED-133 flight data exchange between air traffic control centers
• Generic Vehicle Architecture (GVA)
• Future Airborne Capability Environment (FACE)
• Open Mission Systems (OMS)
• Open Architecture Radar Interface Standard (OARIS)
• Unmanned Aircraft Systems Control Segment (UCS)
• Joint Architecture for Unmanned Systems (JAUS) over DDS
• Layered Simulation Architecture
• Navy Open Architecture

Core Standard Criterion	Protocol Checklist
Provide **syntactic interoperability**	√
Open standard with strong **independent, international** governance	√
Horizontal in its applicability across industries	√
Stable and **deployed** across multiple vertical industries	Military, Software Integration
Have **standards-defined *Core Gateways*** to all other core connectivity standards	Web Services, OPC-UA
Meets connectivity **functional** requirements	√
Meet **non-functional** requirements of performance, scalability, reliability, resilience	√
Meet **security** and **safety** requirements	√
Not require any single component from any single vendor	√
Have readily-available SDKs both **commercial** and **open source**	√

MQTT - MESSAGE QUEUE TELEMETRY TRANSPORT



Network decoupling: publisher and subscriber do not need to know each other IP address

Time decoupling: Publisher and subscriber do not need to run at the same time.

Synchronization decoupling: pub/sub is non-blocking.
Pub/Sub provides a greater scalability than the traditional client-server approach because its operations can be highly parallelized and event-driven.

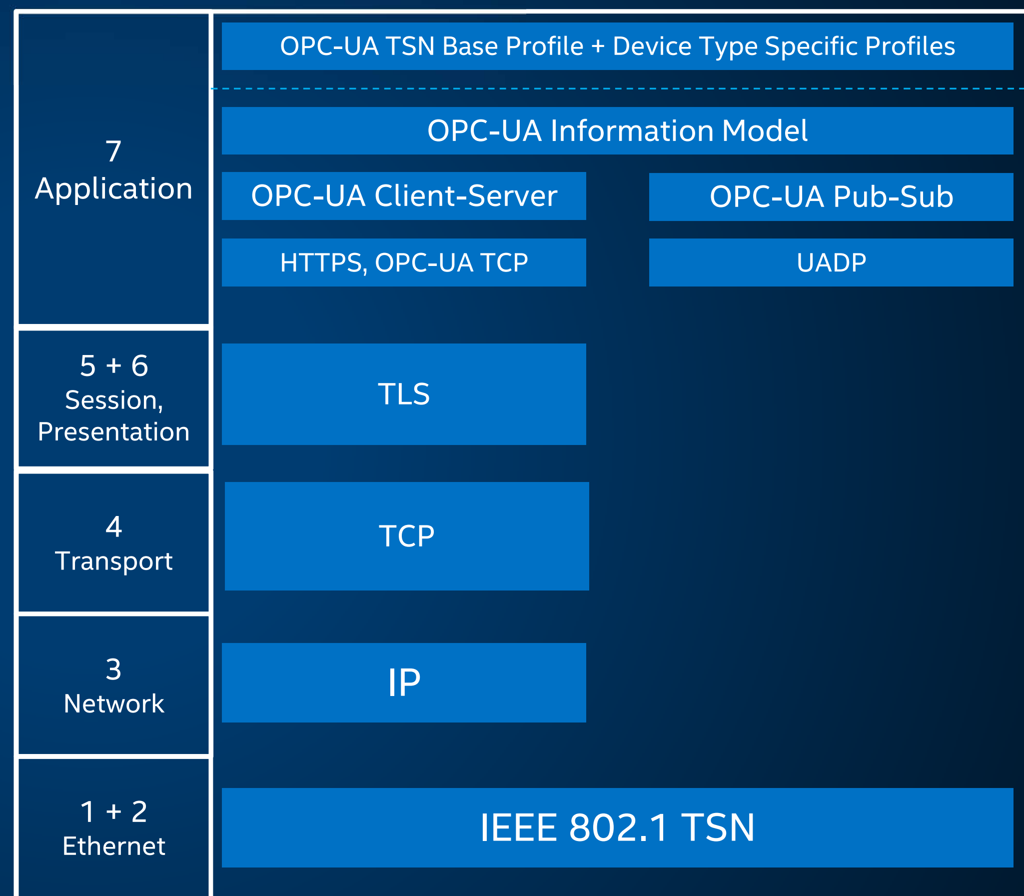
WHAT IS OPC-UA?

OPC UA is a protocol for industrial communication and has been standardized in the IEC 62541 series. At its core, OPC UA defines:

- OPC Unified Architecture (OPC-UA) is a platform independent service-oriented architecture that integrates all the functionality of the individual OPC Calssica specifications into one extensible framework.
- an asynchronous protocol (built upon TCP, HTTP or SOAP) that defines the exchange of messages via sessions, (on top of) secure communication channels, (on top of) raw connections,
- a type system for protocol messages with a binary and XML-based encoding scheme,
- a meta-model for information modeling, that combines object-orientation with semantic triple-relations, and
- a set of 37 standard services to interact with server-side information models. The signature of each service is defined as a request and response message in the protocol type system.

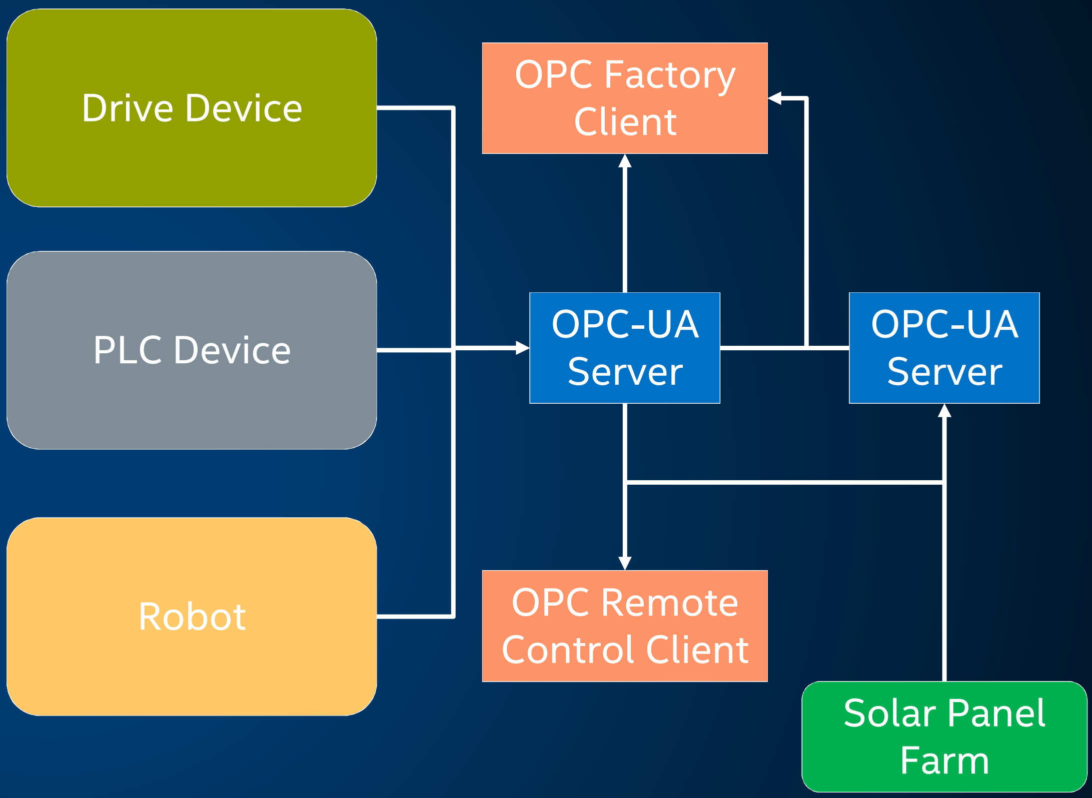
The standard itself can be purchased from IEC or downloaded for free on the website of the OPC Foundation at https://opcfoundation.org/.

OPC-UA TSN IN THE OSI REFERENCE MODEL

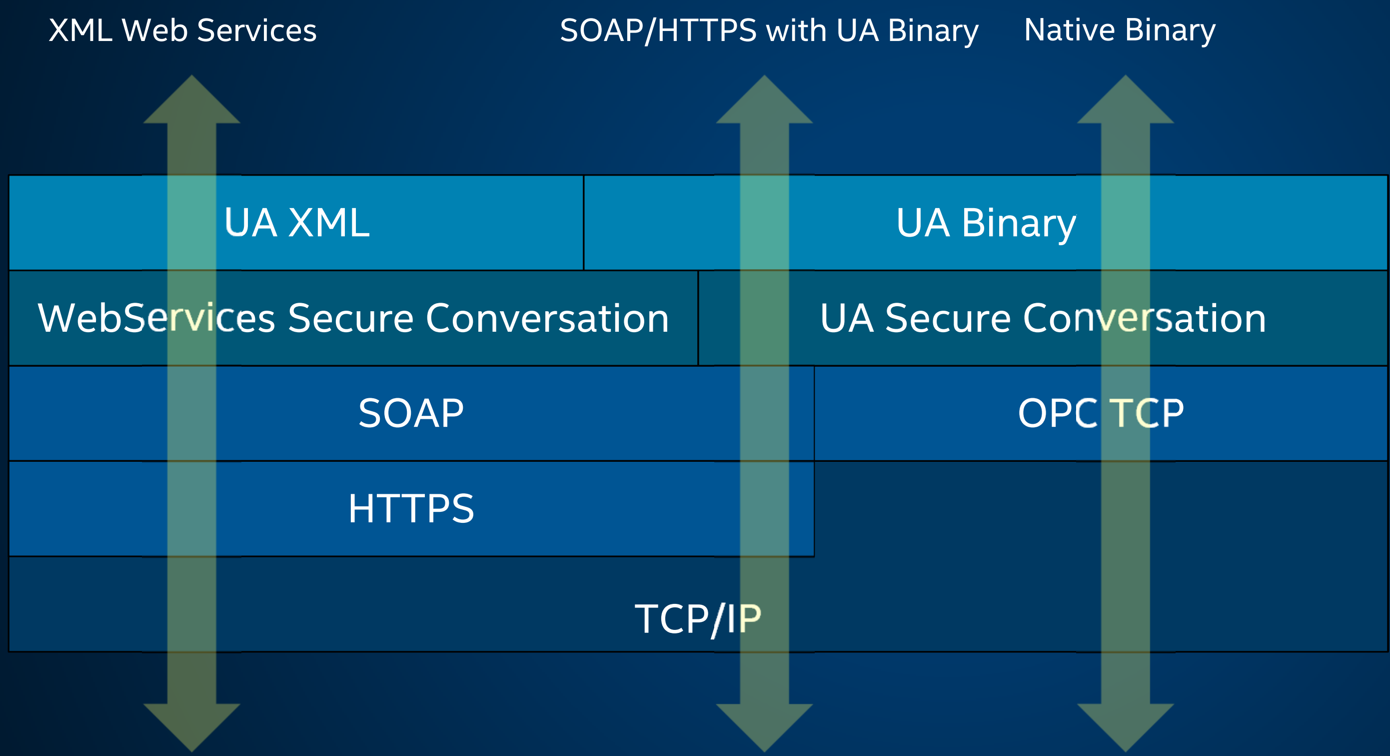


OPC-UA

- OPC-UA tries to solve large scale industrial networking and automation process with platform independent, service oriented and object base data communication architecture.
- OPC-UA provides integrated models and services to ease management of numerous devices and client connections.
- Device/Sensor represented as node object with object-oriented approach custom methods, data fields can be added, OPC Server service are used to access nodes through the provided address spaces.
- Client are end points where control and automation handled with connecting to server.



SECURE CLIENT/SERVER COMMUNICATIONS



- Authentication
- Authorization
- Encryption
- Data integrity

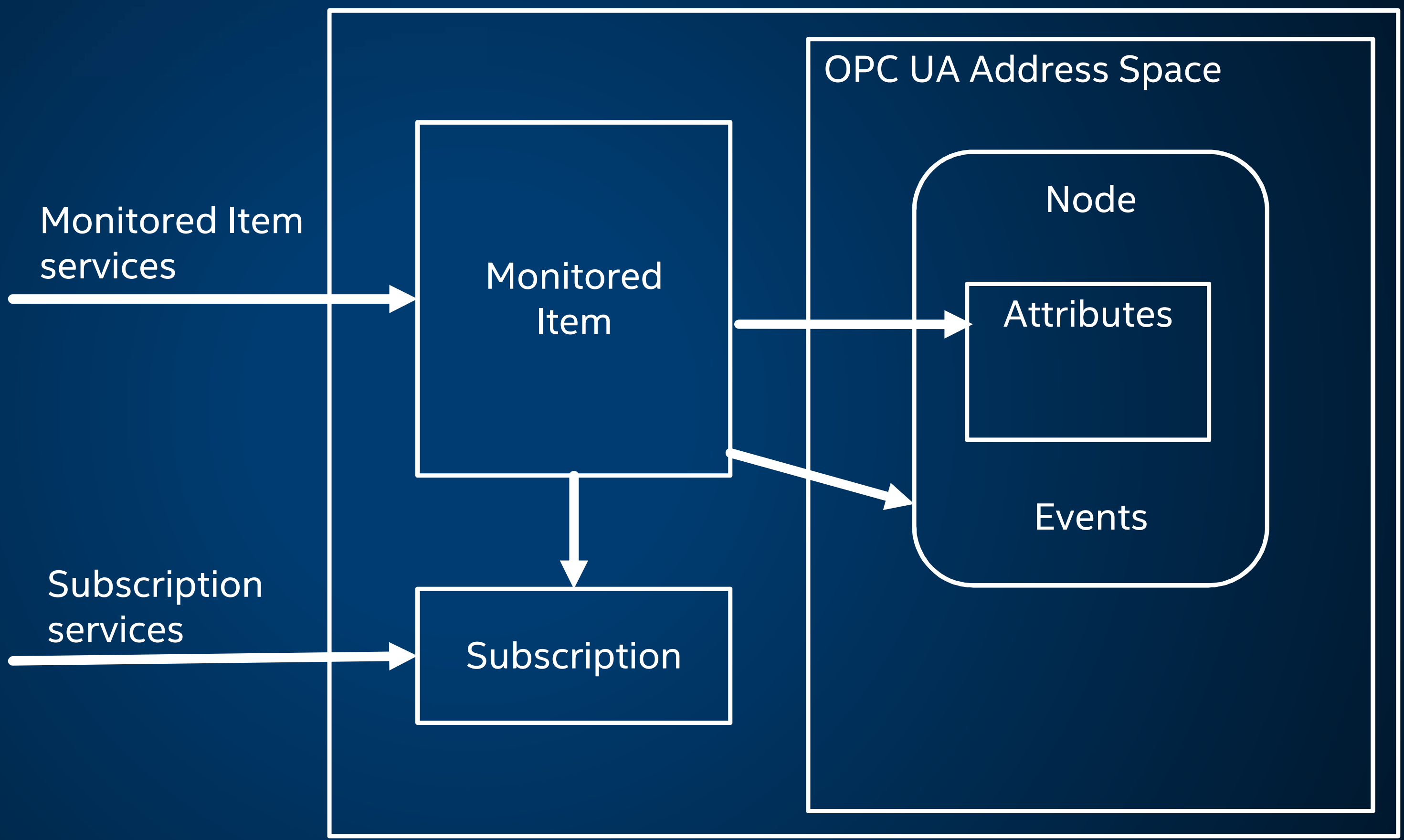
Protocol Stack diagram: GFDL-self/cc-by-sa3.0 https://de.wikipedia.org/wiki/Datei:Uaprotocols.png

SERVICE DISCOVERY MODELS

The *Subscription Service Set* defines *Services* that allow *Clients* to create, modify and delete *subscriptions*.

Subscriptions send *notifications* generated by *MonitoredItems* to the *Client*.

Subscription Services also provide for *Client* recovery from missed *Messages* and communication failures.



SERVICE DISCOVERY MODELS

The *Subscription Service Set* defines *Services* that allow *Clients* to create, modify and delete *subscriptions*.

Subscriptions send *notifications* generated by *MonitoredItems* to the *Client*.

Subscription Services also provide for *Client* recovery from missed *Messages* and communication failures.

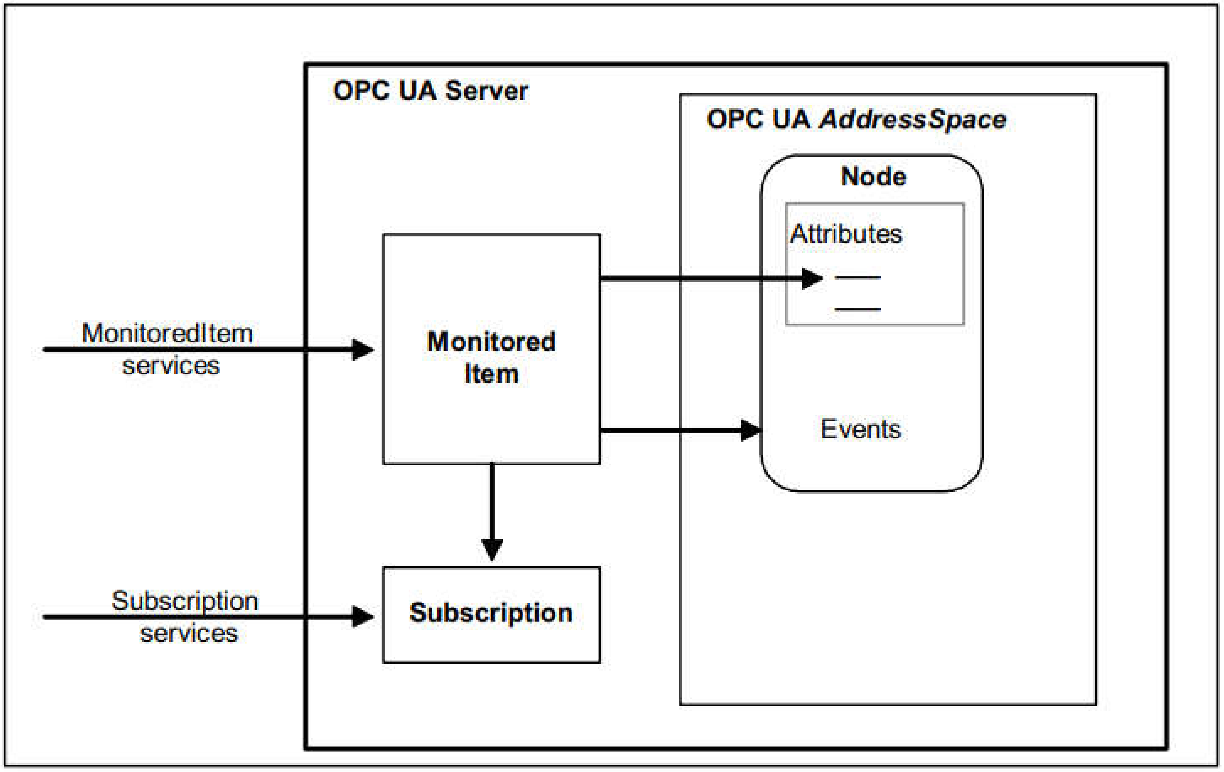


Figure 8 – MonitoredItem and Subscription Service Sets

RESILIENT COMMUNICATIONS

Redundancy

- OPC UA client and server high-availability
- Client: Active/Active
- Server: Passive/Active

Bidirectional "heartbeat"

- OPC UA clients and servers detect connection failures

Buffering

- OPC UA clients detect missing data and may request again

SOPHISTICATED INTERACTIONS

I need to drill this plate

Are you able to drill materials?

Yes, which one?

Steel

Fine, which diameter?

1 inch (2.54 cm)

Fine, please provide coordinates and depth

X: 10, Y: 5, Z: 1

Fine, I am available now

I am sending you the plate

Ok, waiting for it

THE OPC-UA UNIFIED ARCHITECTURE SPECIFICATION

- Part 1: OPC UA Specification: Part 1 – Concepts
- Part 2: OPC UA Specification: Part 2 – Security Model
- Part 3: OPC UA Specification: Part 3 – Address Space Model
- Part 4: OPC UA Specification: Part 4 – Services
- Part 5: OPC UA Specification: Part 5 – Information Model Release
- Part 6: OPC UA Specification: Part 6 – Mappings
- Part 7: OPC UA Specification: Part 7 – Profiles
- Part 8: OPC UA Specification: Part 8 – Data Access
- Part 9: OPC UA Specification: Part 9 – Alarms and conditions
- Part 10: OPC UA Specification: Part 10 – Programs
- Part 11: OPC UA Specification: Part 11 – Historical Access
- Part 12: OPC UA Specification: Part 12 – Discovery and Global Services
- Part 13: OPC UA Specification: Part 13 – Aggregates
- Part 14: OPC UA Specification: Part 14 – Pub/Sub

OPEN62541 FEATURES

Communication Stack

- model – Support for all OPC UA node types (including method nodes) – Support for adding and reOPC UA binary protocol
- Chunking (splitting of large messages)
- Exchangeable network layer (plugin) for using custom networking APIs (e.g. on embedded targets)

Information model

- Support for all OPC UA node types (including method nodes)
- Support for adding and removing nodes and references also at runtime.
- Support for inheritance and instantiation of object- and variable-types (custom constructor/destructor, instantiation of child nodes)

Subscriptions

- Support for subscriptions/monitor items for data change notifications
- Very low resource consumption for each monitored value (event-based server architecture)

Code-Generation

- Support for generating data types from standard XML definitions
- Support for generating server-side information models (nodesets) from standard XML definitions

RESOURCES

- [MQTT.org]
- [Publishing Sensor Data with MQTT Video]

TIME SYNCHRONIZATION FOR AAA2C

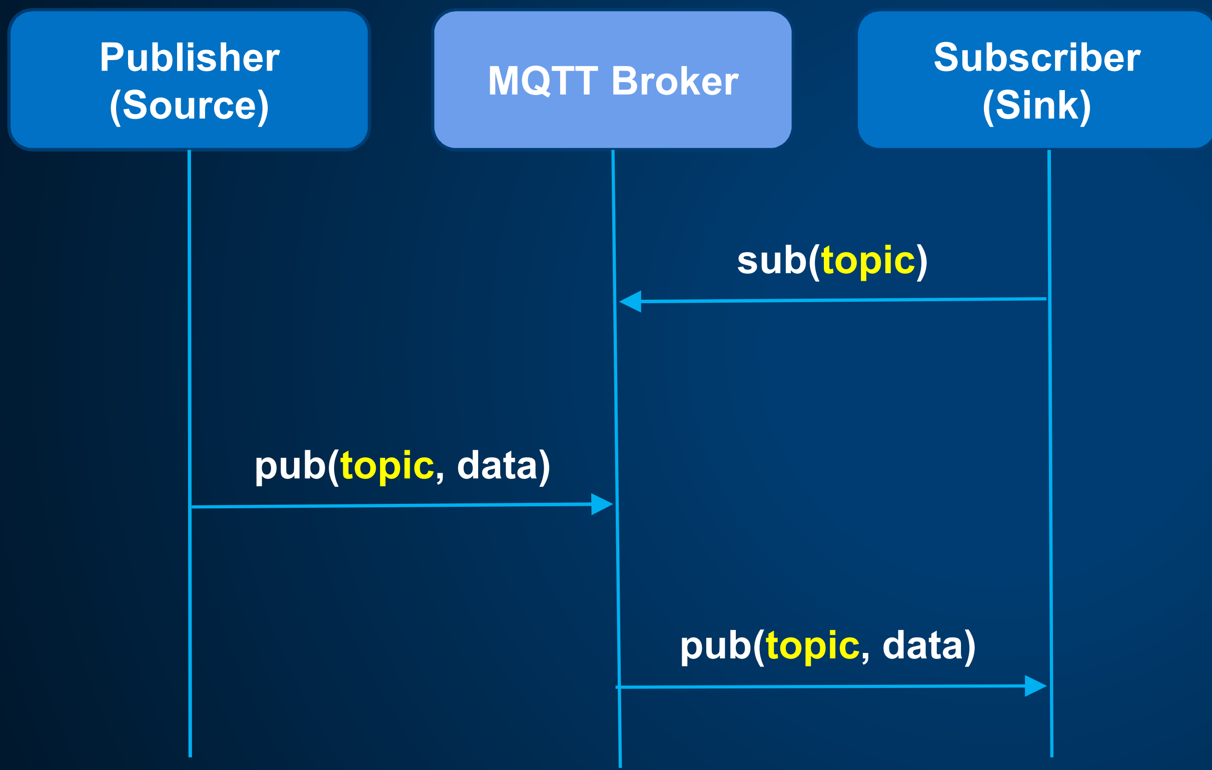
IEEE Std 802.1AS™-2011

802.1AS GOALS

Distribute a single, accurate time reference that is optimized for audio and video synchronization.

- Accurate – Worst-case error less than +/- 500ns in standard AVB LAN
- Completely Immune to LAN traffic bursts and jitter – Only equipment failure degrades traffic
- Self Configuring
  - Grand Master clock is selected automatically
  - Time stabilizes in a fraction of a second
  - Clock tree reconfigures automatically if Grand Master is lost

MQTT - MESSAGE QUEUE TELEMETRY TRANSPORT



Network decoupling: publisher and subscriber do not need to know each other IP address

Time decoupling: Publisher and subscriber do not need to run at the same time.

Synchronization decoupling: pub/sub is non-blocking.
Pub/Sub provides a greater scalability than the traditional client-server approach because its operations can be highly parallelized and event-driven.

EXAMPLE RESTFUL HTTP API

HTTP Verb	URI Path	Purpose
GET	/lcd/text/	Returns the current text on the LCD screen
POST	/lcd/text/ json: {value:"Hello World"}	Sets the text on the LCD
DELETE	/lcd/text/	Clear Texts
GET	/lcd/backlight/	Returns the current state of the LCD backlight
POST	/lcd/backlight/ json:{r:255, b:0, g:0}	Sets the backlight to rgb(255,0,0) or RED
DELETE	/lcd/backlight/	Turns the backlight off

- Addressable Resources
- Constraint Interface
- Resources may be consumed in multiple formats (JSON, XML, etc..)
- All stateful information is held on the client side.