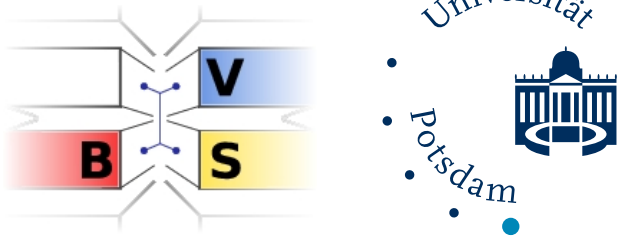
Internet of Things (IoT)

Kristina Sahlmann

Betriebssysteme und Verteilte Systeme
Institut für Informatik und Computational Science
Universität Potsdam



Sommersemester 2020

Dates

- Kick-Off lecture: 20.04.2020
- MYNO project introduction: 27.04.2020 (online)
- Presentation training: 04.05.2020 (online, Petra Vogel)
- Presentations as a block course: 15.06., 22.06. and 29.06.

Supervisors: Dipl.-Inf. (FH) Kristina Sahlmann
kristina.sahlmann@uni-potsdam.de
Website: `https://www.cs.uni-potsdam.de/bs/teaching/docs/courses/ss2020/iot/`

Requirements

- Make an appointment for discussion 2 weeks before presentation
- Deliver the presentation draft 2 weeks before the presentation
- Successful presentation: max. 45 min. incl. Code-Review + 15 min. Discussion,
- Handout/Glossary is necessary: max. 1 DIN A4 page
- Deliver the documentation within 1 week after the presentation.
- For documentation: use LaTeX templates and advices from our Website `https://www.cs.uni-potsdam.de/bs/teaching/studentHints.html`
- **Participation for all students is mandatory**

Handout

The handout shall be interesting for the other participants. It should include the most important and useful sources (books, papers, websites). If you put figures on the handout, don't forget to cite the source. (The same applies for your slides.)

Therefore, the handout includes:

- the name of the presenter,
- the source of re-used tables or figures,
- recommended sources for further reading,
- a summary/conclusion of your talk: This is the take-away of your talk!
- ...

Grade

The grade is composed as following:

- 10% presentation draft
- 30% presentation content
- 30% successful presentation (style)
- 30% documentation (PDF and double-sided printed)

Topics for presentations

How to apply for a topic?

1. Send an email with your favorite topic and one alternative until 4th of May! State your preference for the presentation date: June or September!
2. I will do my best (aka FIFO) and will send an email with the final mapping until 8th of May.

Internet of Things (IoT) and Protocols

Internet of Things (IoT)

„The Internet of Things is a system of physical objects that can be discovered, monitored, controlled, or interacted with by electronic devices that communicate over various networking interfaces and eventually can be connected to the wider Internet.“ These physical objects are equipped with sensors and actuators.

from the book „Building the Web of Things“, 2016, Guinard und Trifa

Interoperability

Interoperability refers to the possibility to integrate **different systems and data in a single workflow**. This assumes that **syntax and semantics of data and systems** will be provided in a **uniform way** to the user.
... At the **conceptual level**, interoperability means that a **common understanding of the facts (common world view)** exists between the participants.
At the **system level**, interoperability requires that **different software applications communicate directly and smoothly with each other**...

from the book “Grundlagen der Geoinformationssysteme“, 2010, Ralf Bill

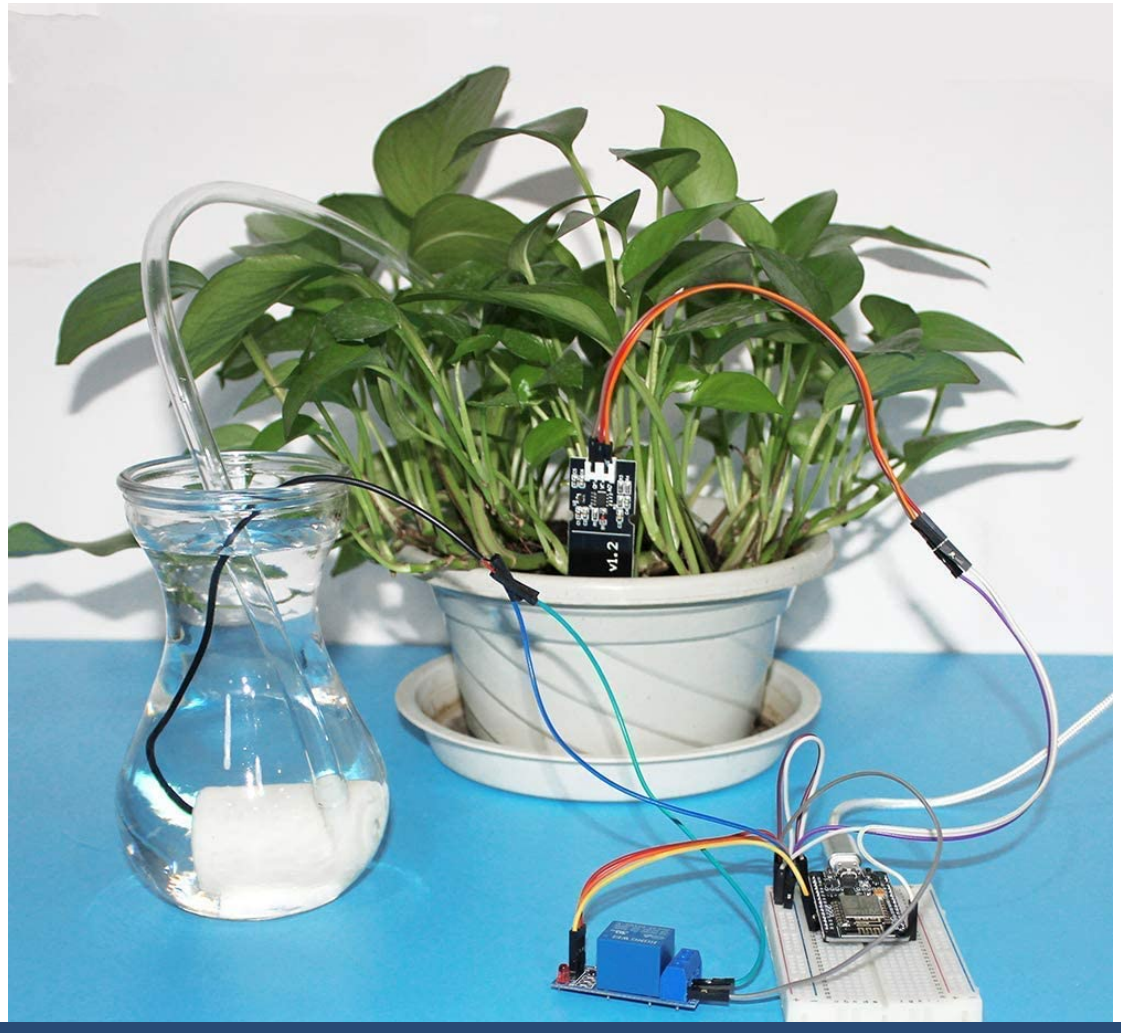
Smart Agriculture / Smart Home: Monitoring with Wireless Sensor Network (WSN)

Different IP-based sensor boards: WLAN, IEEE 802.15.4 (6LoWPAN)
Building monitoring: temperature, humidity, light, etc.

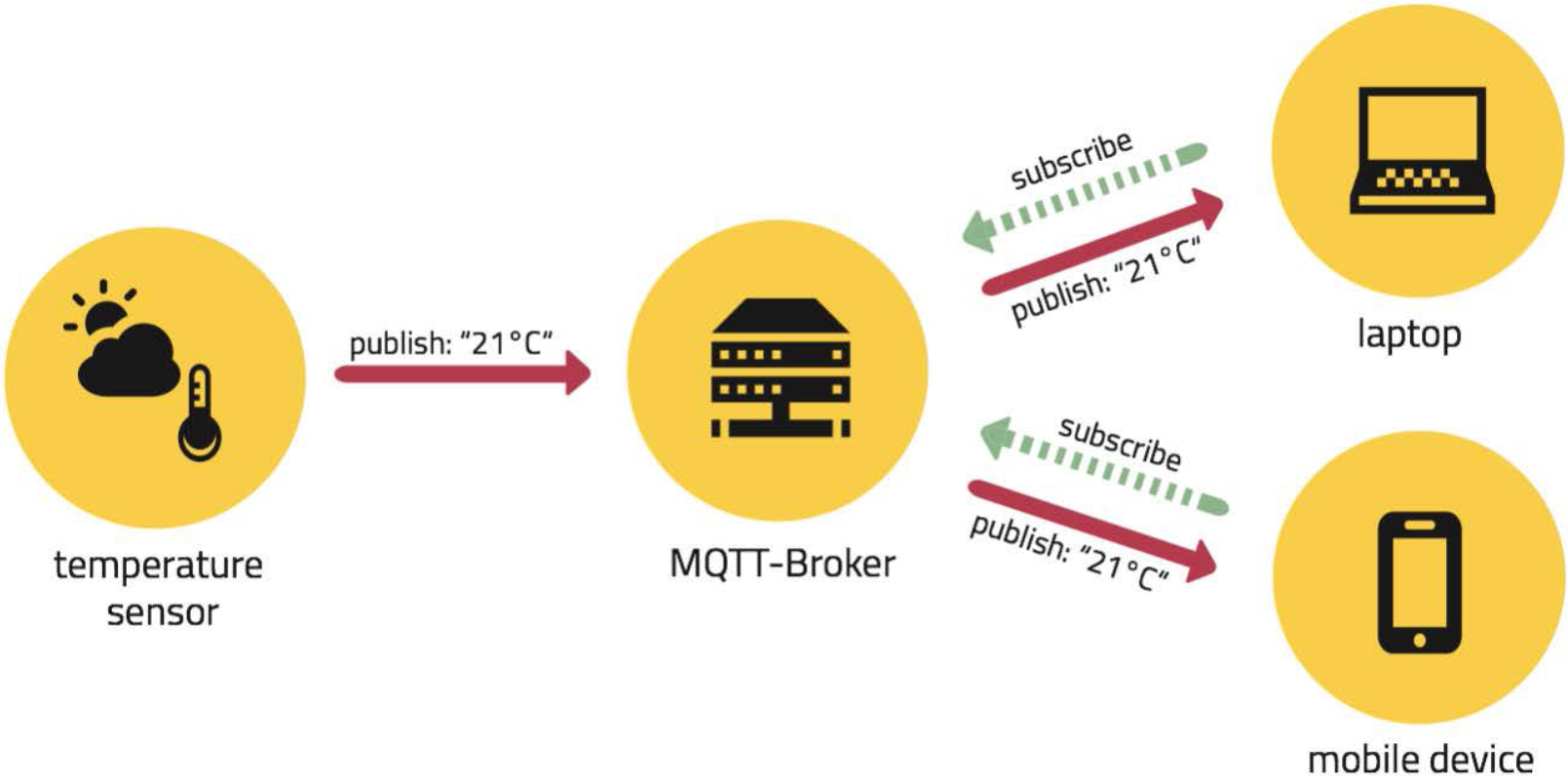
Requirements:

- heterogeneity: sensor boards must be replaceable
- self-organization of sensor network
- uniform way to control the actuators
- sensor data can be collected and analyzed
- alarm/event can be triggered

Sensor Networks for Precision Agriculture (on the windowsill)

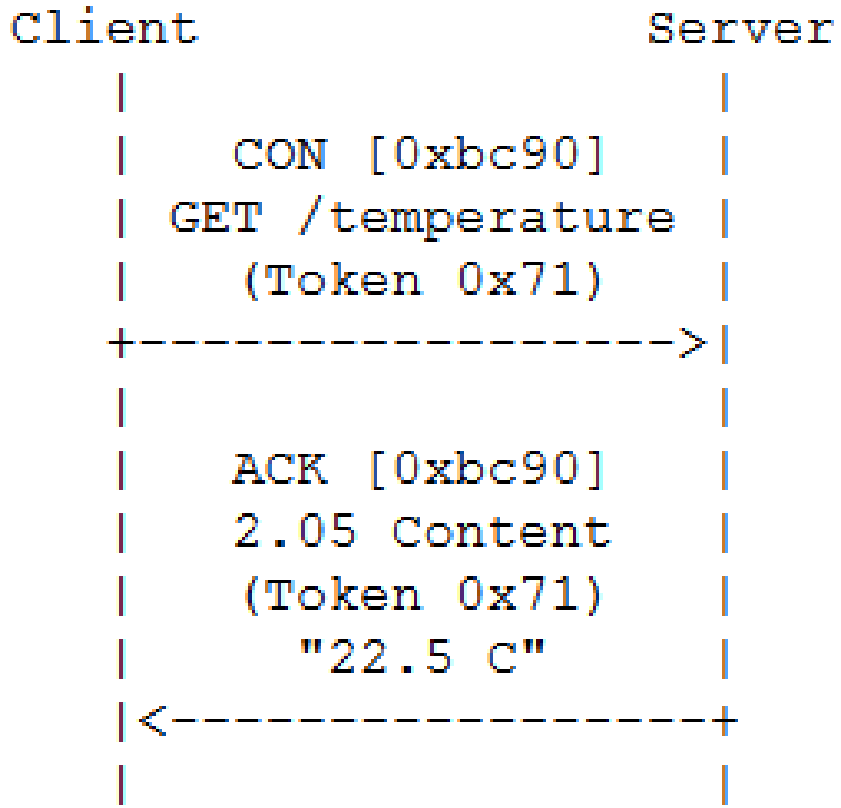


Publish/Subscribe Paradigm



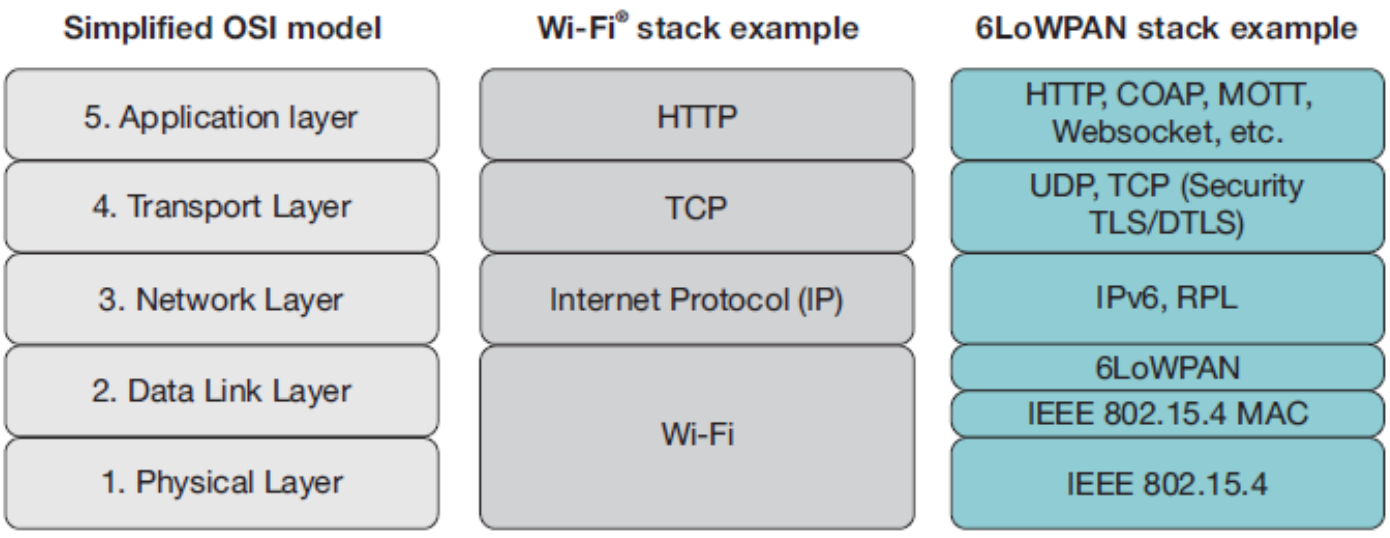
`http://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html`

RESTful protocol



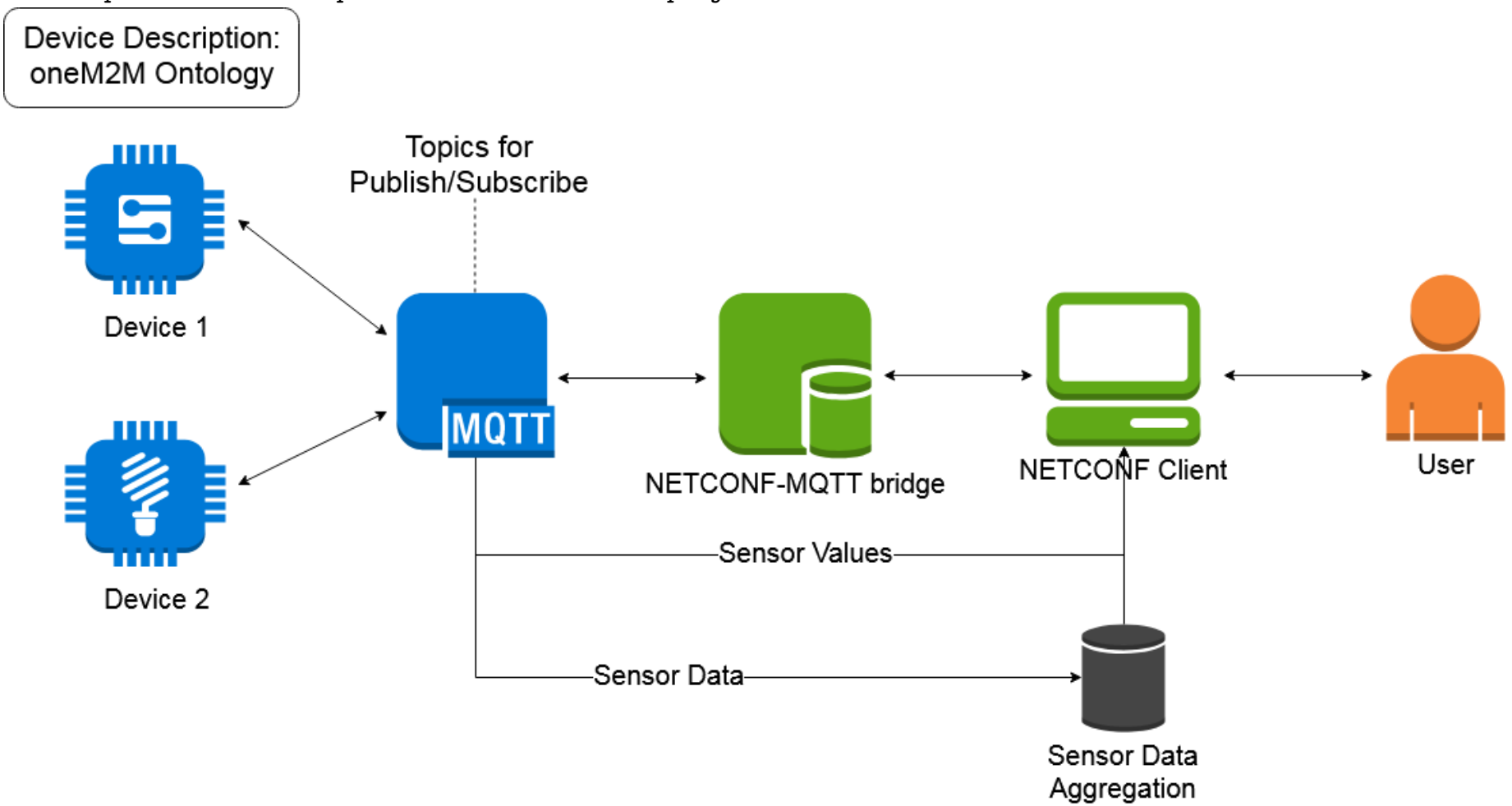
RFC 7252, RFC 7390, RFC 6690, `http://coap.technology/`

6LoWPAN = IPv6 over Low-Power Wireless Personal Area Network



Semantic Device Descriptions for the Interoperability in Network Management in the IoT

https://www.cs.uni-potsdam.de/bs/research/projectIot.html



A survey

Service discovery techniques in Internet of Things: a survey,

by Hela Zorgati, Raoudha Ben Djemaa, Ikram Amous Ben Amor, in 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)

The survey: Comparison of reviewed works

	Approach	Description	Architecture	Discovery Scope	IoT requirements			
					Interoperability	*Scalability*	*Context-awareness*	*Security*
Protocol-based	[16]	Syntactic	Centralized and Distributed	Local	×	×	o	o
	[17]	Syntactic	Distributed	Local and remote	×	✓	o	o
	[20]	Syntactic	Centralized	Local	×	×	×	×
	[21]	Syntactic	Distributed	Local	×	×	o	×
	[22]	Syntactic	Distributed	Local	×	×	o	×
	[23]	Syntactic	Centralized	Local	×	×	o	×
Semantic-based	[26]	Semantic	Distributed	Local	✓	✓	×	×
	[27]	Semantic	Distributed	Remote	✓	✓	×	×
	[29]	Semantic	Centralized	Remote	✓	✓	✓	×

✓ Supported × Not supported o Partially supported

[27] MYNO project

Comparison aspects

- semantic-based vs. protocol-based (syntactic)
- architecture: centralized vs. distributed
- Cloud vs. Edge
- discovery scope: local vs. remote
- interoperability
- scalability
- context-awareness
- security
- further aspects: supported protocols, bootstrapping, network/device management

IoT Architectures / Frameworks

Tasks for assessment of the IoT framework

- facts: who is behind, since when, recent activities, etc.
- which tasks / purposes are completed by a framework?
- comparison between its specification and implementation: what is missing?
- how is the implementation to use? is it comfortable for developers?
- which premises / knowledge is required?
- is interoperability ensured? where and how?
- maturity of the framework
- pro and contra of the framework / implementation
- ask for IoT Hardware for evaluation (Raspberry Pi or similar)

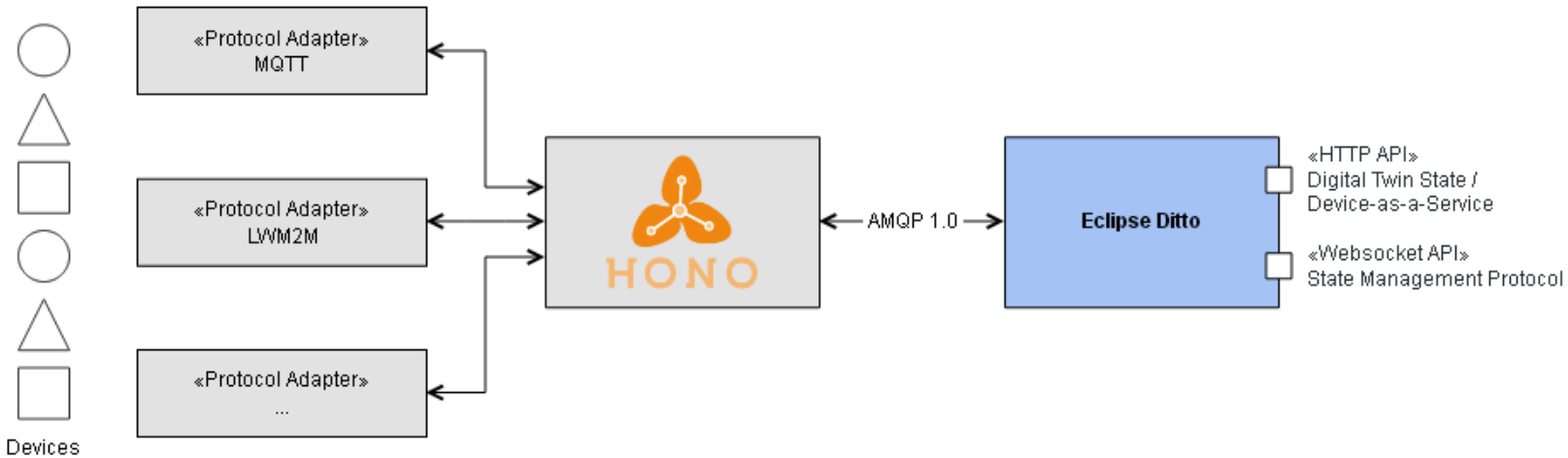
1 Eclipse Vorto Project from Bosch

- **Why?** (i) Bosch is a one of the big players in the IoT; (ii) Vorto uses a semantic approach related to MYNO project.
- Vorto consists of the meta information model, the tool set to create information models, the code generators and the repository to manage existing information models.
- Introduction of related Eclipse Projects: Eclipse Ditto (Digital Twin) and Eclipse Hono (IoT Connector cloud), also from Bosch

Links:

Vorto Project `https://projects.eclipse.org/projects/iot.vorto`, `https://www.eclipse.org/vorto/`, `https://github.com/eclipse/vorto/blob/master/docs/gettingstarted.md`
Ditto Project `https://projects.eclipse.org/projects/iot.ditto`, `https://www.eclipse.org/ditto/`,
Hono Project `https://projects.eclipse.org/projects/iot.hono`, `https://www.eclipse.org/hono/`

Eclipse Ditto (based on Vorto) and Hono



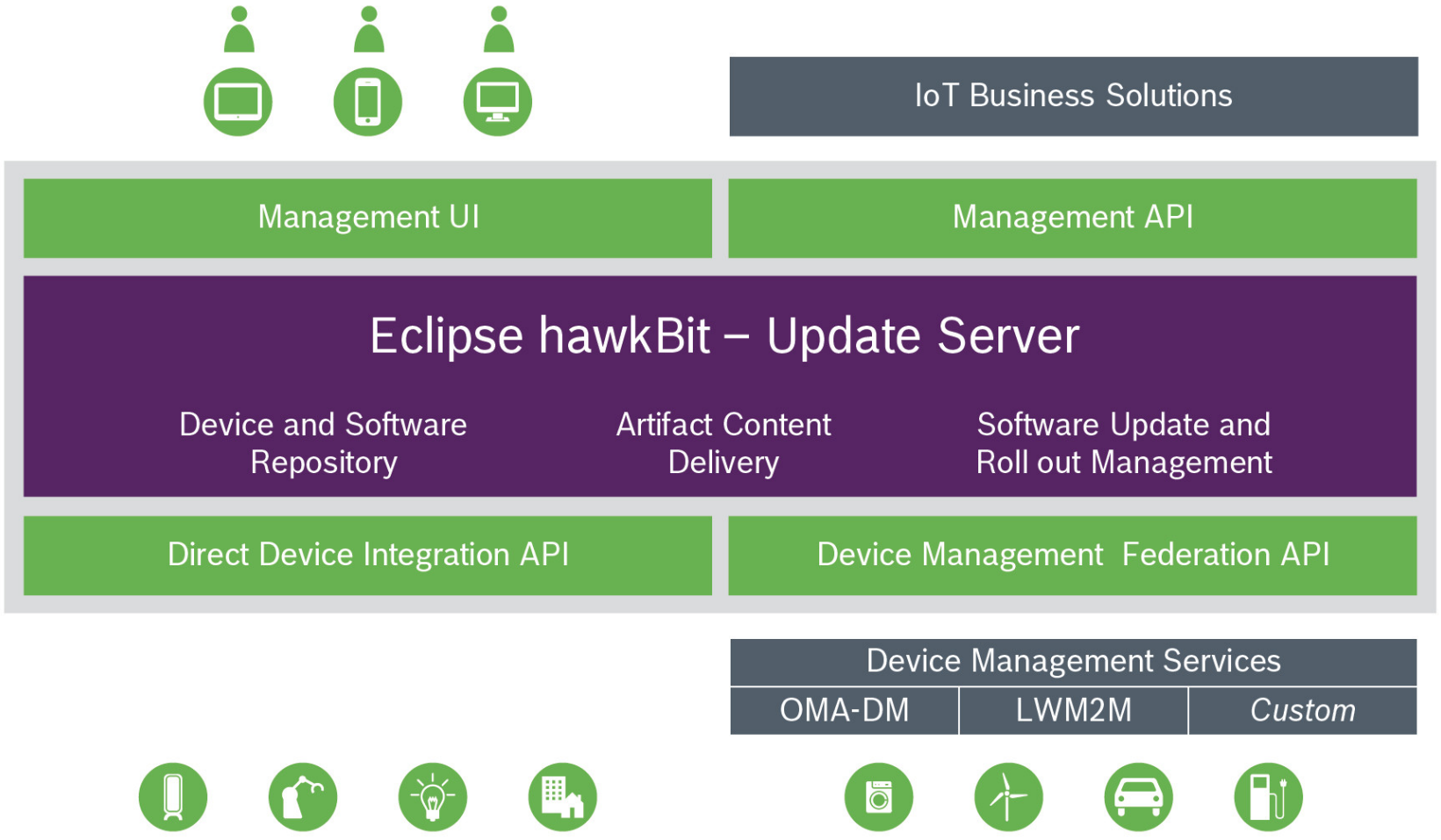
2 Eclipse hawkBit - Over-The-Air (OTA) Updates from Bosch

- **Why?** (i) Bosch is a one of the big players in the IoT; (ii) related to MYNO Update Protocol (MUP);
- hawkBit is a domain independent back end solution for rolling out software updates to constrained edge devices

Links:

hawkBit Project `https://projects.eclipse.org/projects/iot.hawkbit`, `https://www.eclipse.org/hawkbit/`

Eclipse hawkBit

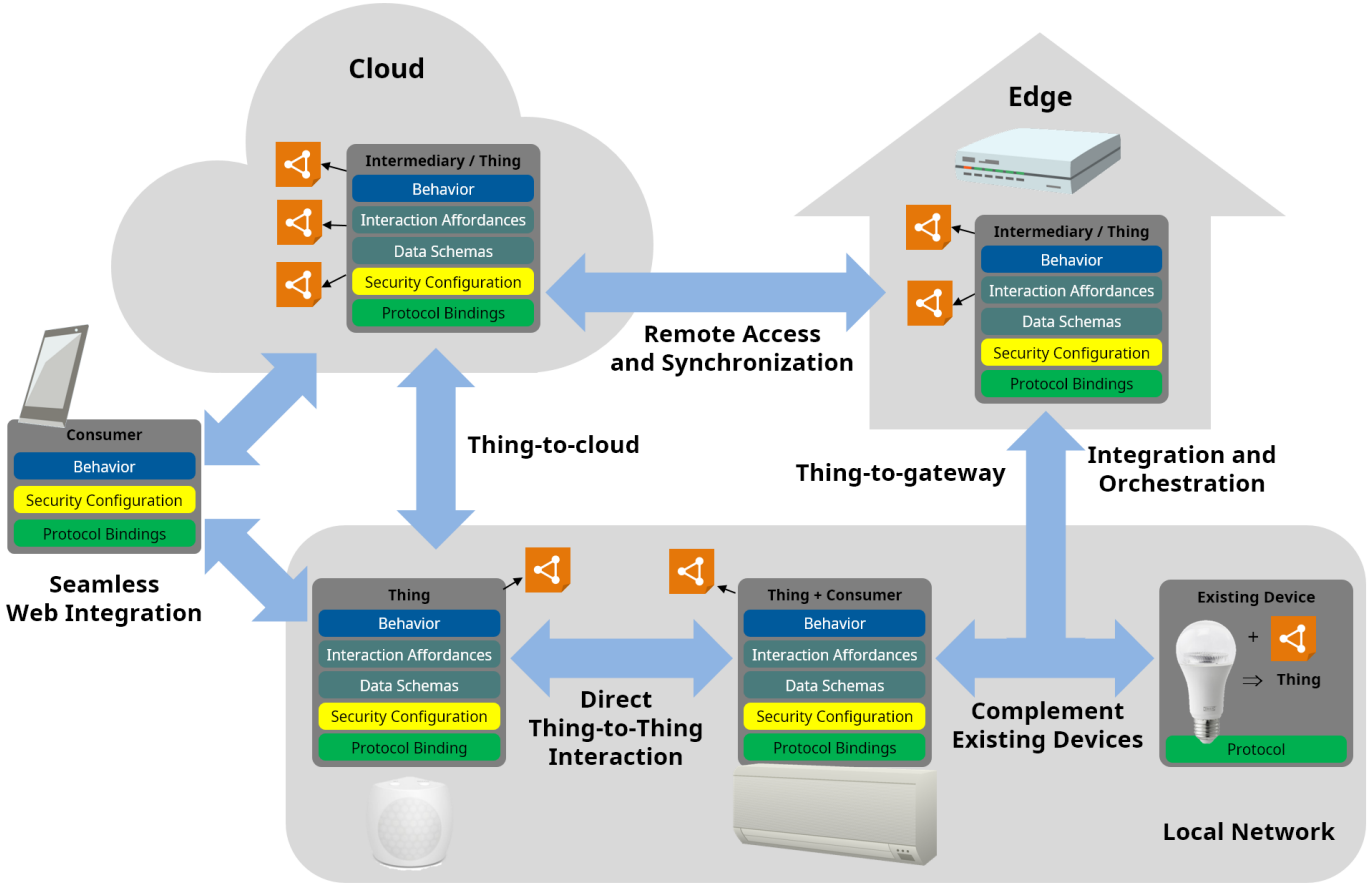


3 Mozilla IoT and W3C Web of Things

- **Why?** (i) Both represent an approach of Web of Things;
  (ii) Mozilla IoT is an open-source implementation;
  (iii) W3C Web of Things is a new W3C standard;
  (iv) Both use a semantic approach related to MYNO project.
- Mozilla WebThings: Gateway and Framework
- Mozilla adopts the idea of the (original) Web of Things (WoT) from Guinard and Trifa: Web Thing API specification
- Comparison between Mozilla IoT and W3C WoT approaches

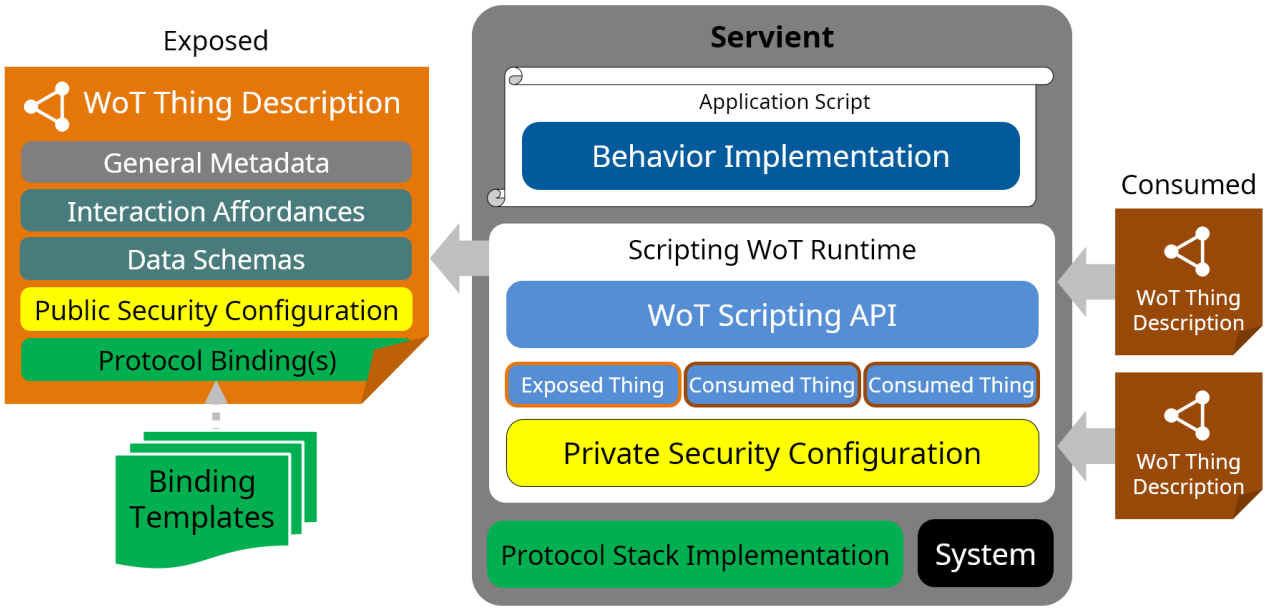
Links:
Mozilla IoT `https://iot.mozilla.org/`
W3C Web of Things `https://www.w3.org/WoT/`

W3C Web of Things (WoT) Architecture



W3C WoT Servient

a Servient is a software stack that implements the WoT building blocks: Thing Description, Binding Templates, Scripting API. Servients can host and expose Things and/or consume Things (i.e., host Consumers). Depending on the Protocol Binding, Servients can perform in both server and client role.



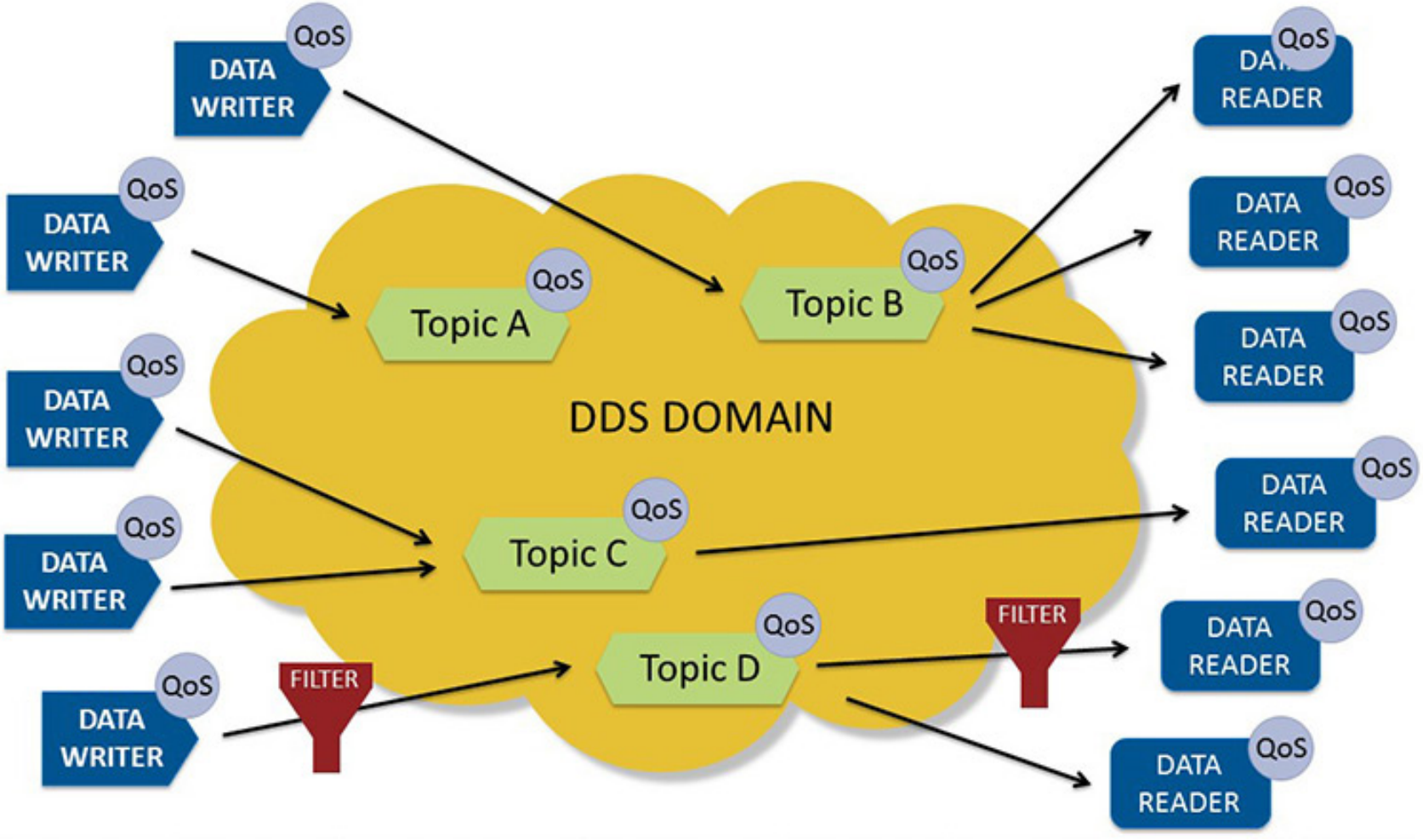
4 OMG Data Distribution Service (DDS)

- **Why?** (i) Industrial IoT (Industry 4.0);
  (ii) DDS is an open standard and have open source implementations; (iii) The data-centric Publish/Subscribe paradigm of DDS is related to MYNO project;
- Two Layers specified: DCPS (Data Centric Publish Subscribe) layer and DLRL (Data Local Reconstruction Layer)
- Related specifications: DDS Interoperability Wire Protocol (DDSI-RTPS), DDS For Extremely Resource Constrained Environments (DDS-XRCE), DDS Security
- open-source implementations: Eclipse Cyclone DDS (DDS Foundation) and OpenDDS

Links:

DDS 1.4 Specification `https://www.omg.org/spec/DDS/`
DDSI-RTPS `http://www.omg.org/spec/DDSI-RTPS/`
DDS-XRCE `https://www.omg.org/spec/DDS-XRCE/`
DDS Security `https://www.omg.org/spec/DDS-SECURITY/`
DDS Foundation `https://www.dds-foundation.org/`
Eclipse Cyclone DDS `https://projects.eclipse.org/projects/iot.cyclonedds`
OpenDDS `http://opendds.org/`
`https://www.informatik-aktuell.de/betrieb/netzwerke/zuverlaessige-datenkommunikation-im-industrial-internet-of-things-mit-dds.html`

DDS - data-centric Publish/Subscribe



5 OMA Lightweight M2M (LWM2M)

- **Why?** (i) device management related to MYNO project; (ii) open-source implementation;
- Device Management Protocol based on a CoAP Basis (REST)
- Open-source implementation: Eclipse Leshan (Java) and Eclipse Wakaama (C files for POSIX compliant systems)

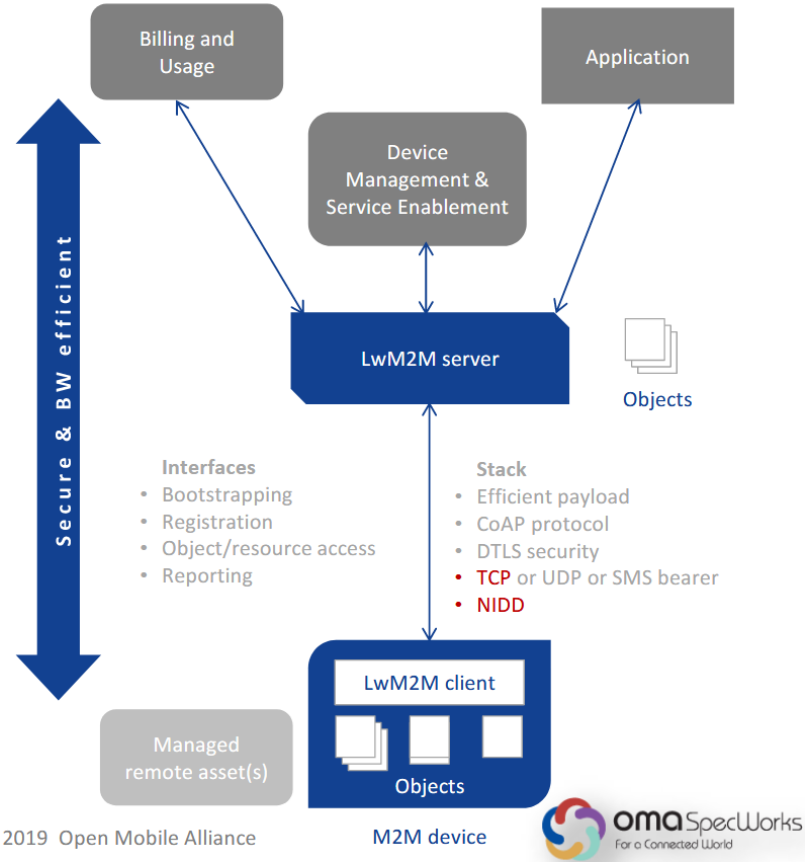
Links:

RFC 7252 - The Constrained Application Protocol (CoAP)
LWM2M Specification
`https://www.omaspecworks.org/what-is-oma-specworks/iot/lightweight-m2m-lwm2m/`
Server `https://github.com/contiki-ng/example-lwm2m-standalone`
Client `https://github.com/eclipse/wakaama`

OMA Lightweight M2M (LWM2M) Introduction

LwM2M is recommended for Device Management and Service Enablement because its benefits include:

- Increased bandwidth efficiency based on COAP bandwidth optimization
- Transport-agnostic design that supports UDP, TCP, SMS
- Developer toolkit for application development
- DTLS-based security based on CoAP (IETF)
- Low power client foot print designed for battery constrained devices
- End to end security using IETF OSCORE



The information in this presentation is public. | Copyright © 2019 Open Mobile Alliance

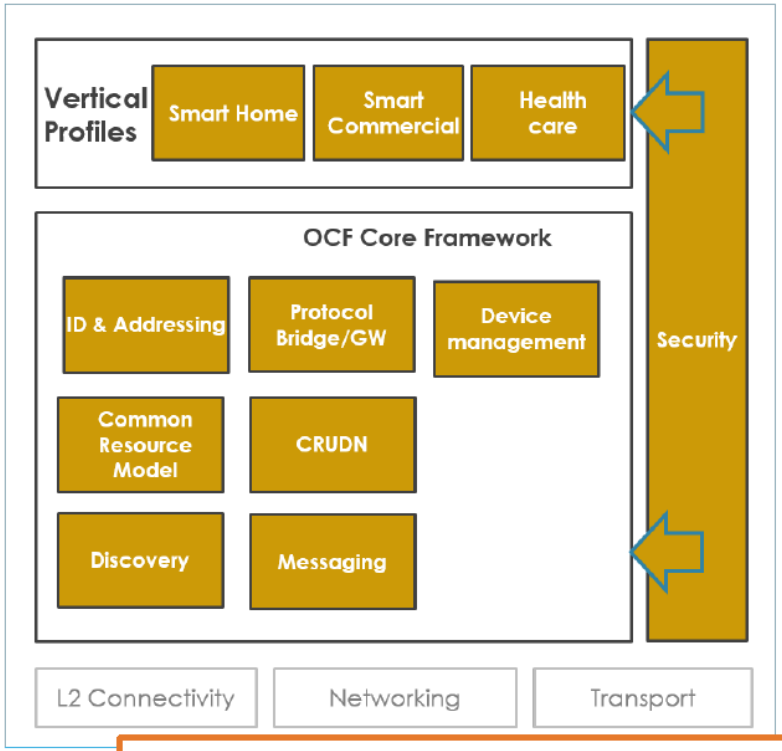
6 Open Connectivity Foundation (OCF) Framework

- **Why?** (i) addresses interoperability; (ii) specifications published as International Standards: ISO/IEC 30118 series; (iii) semantic approach related to MYNO project;
- OCF specification 2.1.2: Core Framework, Core Optional Framework, Security, Resource Type, Device
- Promise: Enable the development of vertical profiles (e.g. Smart Home, Smart Commercial) while maintaining fundamental interoperability via an architecture that is scalable from resource constrained devices to resource rich devices

Links:
OCF `https://openconnectivity.org/foundation/`
OCF Specification `https://openconnectivity.org/developer/specifications/`

ISO/IEC 30118 standards `https://standards.iso.org/ittf/PubliclyAvailableStandards/`

OCF Framework



1. **Discovery:** Common method for device discovery (Multicast CoAP to All OCF Nodes Address)
2. **Messaging:** Constrained device support as default (IETF CoAP) as well as protocol translation via bridges
3. **Common Resource Model:** Real world entities defined as data models (resources)
4. **CRUDN:** Simple Request/Response mechanism with Create, Retrieve, Update, Delete, and Notify operations
5. **ID & Addressing:** Device Identifiers and OCF URIs (map to transport protocol)
6. **Protocol Bridge**: Framework provided by the Bridging Specification

Security is fundamental to the OCF ecosystem and applies to all elements

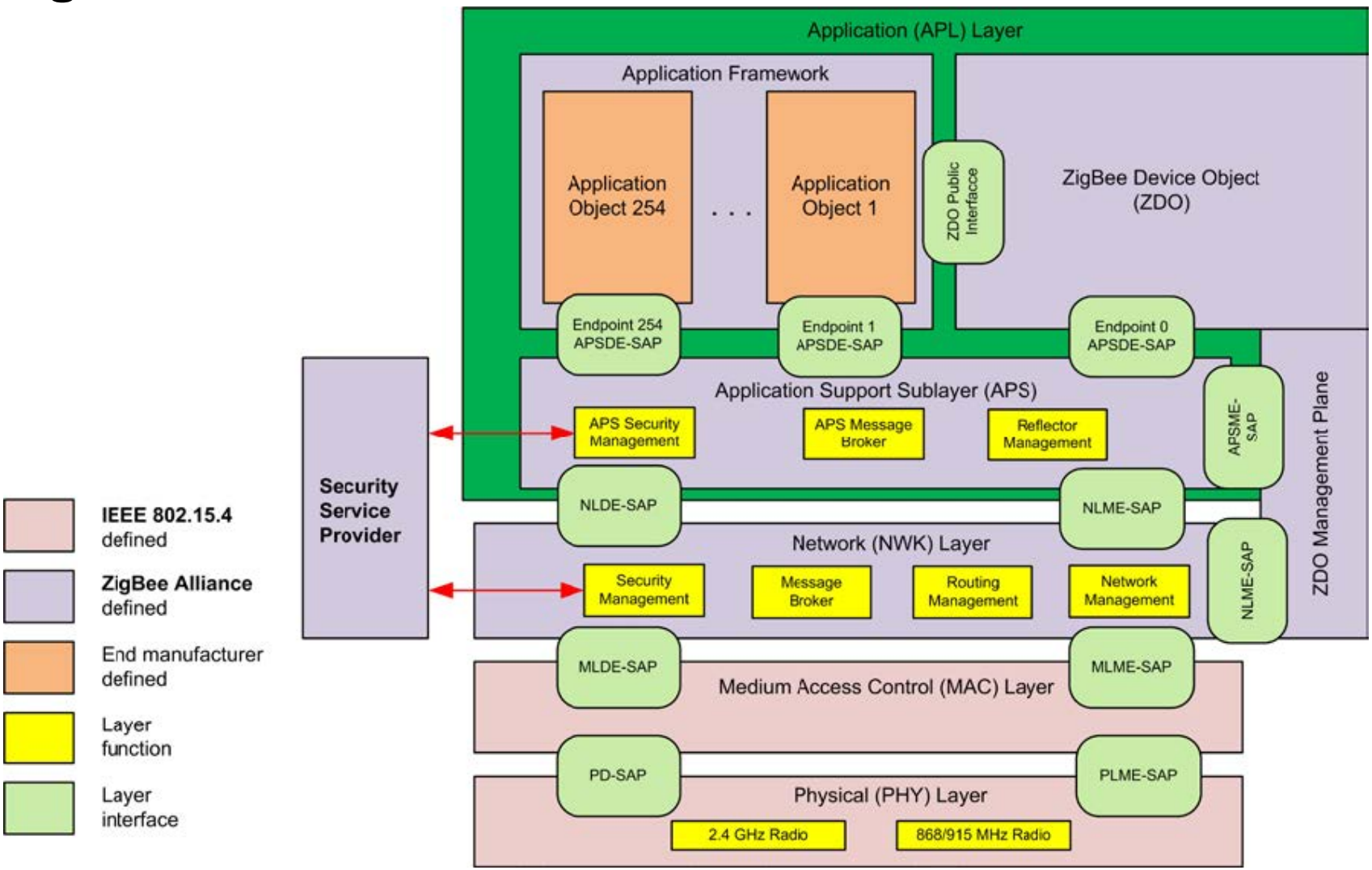
7 Zigbee Protocol

- **Why?** (i) based on IEEE 802.15.4 radio (like 6LoWPAN); (ii) widely adopted in the IoT; (iii) a full stack standard; (iv) approach related to MYNO project;
- Zigbee includes: the application support sub-layer (APS), the ZigBee device objects (ZDO), ZigBee device profile (ZDP), the application framework, the network layer (NWK), ZigBee security services

Links:

Zigbee Specification `https://zigbeealliance.org/solution/zigbee/`
Zigbee Interoperability `https://zigbeealliance.org/wp-content/uploads/2019/12/04-2017-Interoperability-ORIGINAL-White-Paper-Final-Musa-and-Shashank-1.pdf`

Zigbee

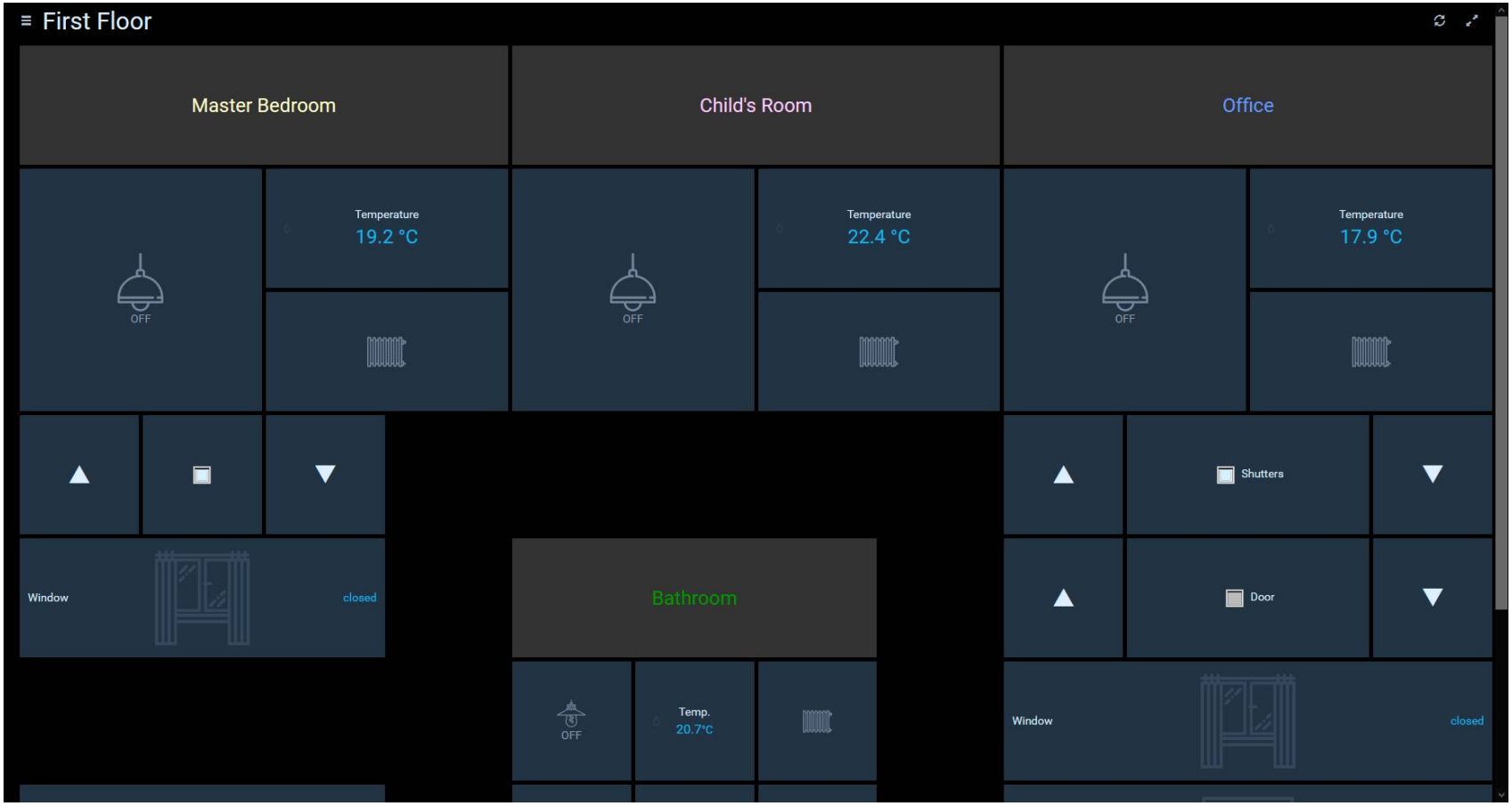

Application (APL) Layer
Application Framework
Application Object 254
. . .
Application Object 1
ZDO Public Interfacce
ZigBee Device Object (ZDO)
Endpoint 254 APSDE-SAP
Endpoint 1 APSDE-SAP
Endpoint 0 APSDE-SAP
Application Support Sublayer (APS)
APS Security Management
APS Message Broker
Reflector Management
APSME-SAP
ZDO Management Plane
Security Service Provider
NLDE-SAP
NLME-SAP
Network (NWK) Layer
Security Management
Message Broker
Routing Management
Network Management
NLME-SAP
MLDE-SAP
MLME-SAP
Medium Access Control (MAC) Layer
PD-SAP
PLME-SAP
Physical (PHY) Layer
2.4 GHz Radio
868/915 MHz Radio
IEEE 802.15.4 defined
ZigBee Alliance defined
End manufacturer defined
Layer function
Layer interface

8 openHAB Software

- **Why?** (i) automatic things discovery; (ii) open source implementation based on Eclipse SmartHome; (iii) approach related to MYNO project;
- openHAB is an open source, technology agnostic automation software for smart home
- Java implementation, based on Eclipse SmartHome Project and OSGi

Links:
openHAB `https://www.openhab.org/`
Eclipse SmartHome project `https://www.eclipse.org/smarthome/`
`https://projects.eclipse.org/projects/iot.smarthome`

openHAB Software



9 Eclipse Kura and Kapua

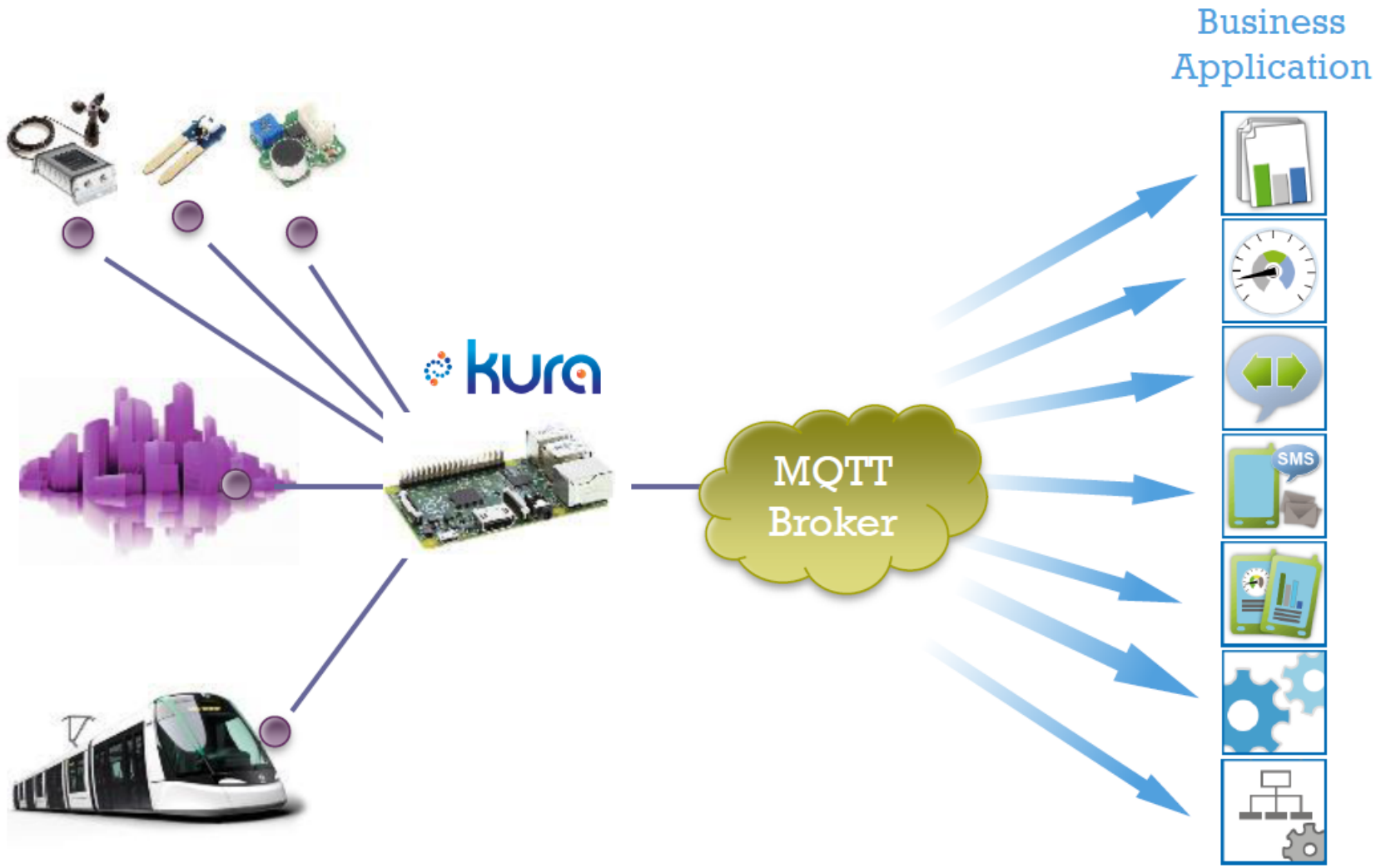
- **Why?** (i) open-source implementations; (ii) MQTT-based approach related to MYNO project;
- Kura is an extensible open source Java/OSGi IoT Edge Framework
- Kura provides gateways services like configuration and data service, networking, remote Management, etc.
- Kapua is an open and modular IoT Cloud Platform based on a micro-services architecture (REST API)
- Kapua provides device connectivity and management, message routing, data management, etc.
- Kura and Kapua is supported by Eurotech company

Links:

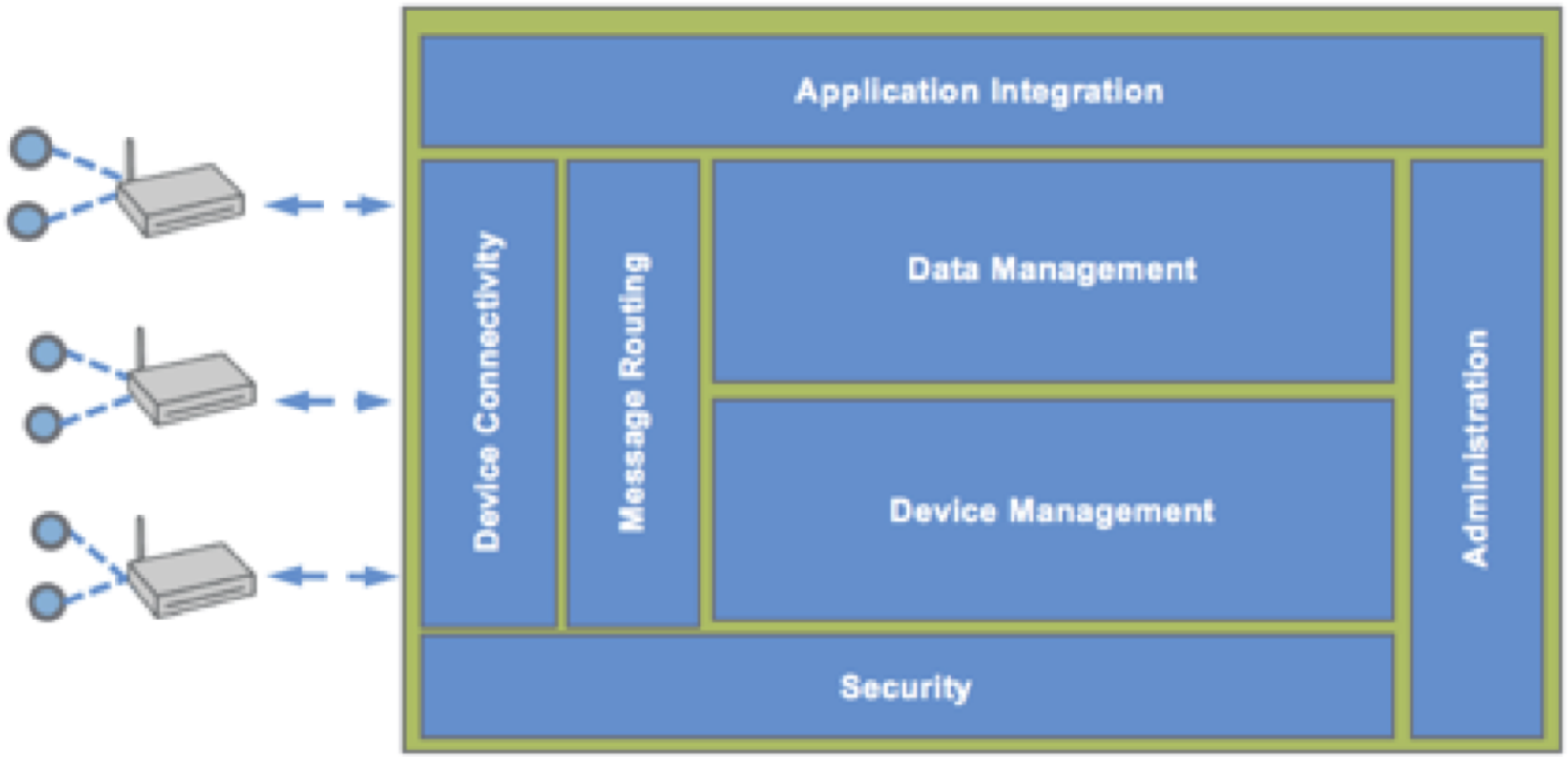
Kura `https://www.eclipse.org/kura/`
`https://projects.eclipse.org/projects/iot.kura`
Kapua `https://projects.eclipse.org/projects/iot.kapua`
`https://www.eclipse.org/kapua/`

Eurotech `https://ec.eurotech.com/docs`

Eclipse Kura



Eclipse Kapua

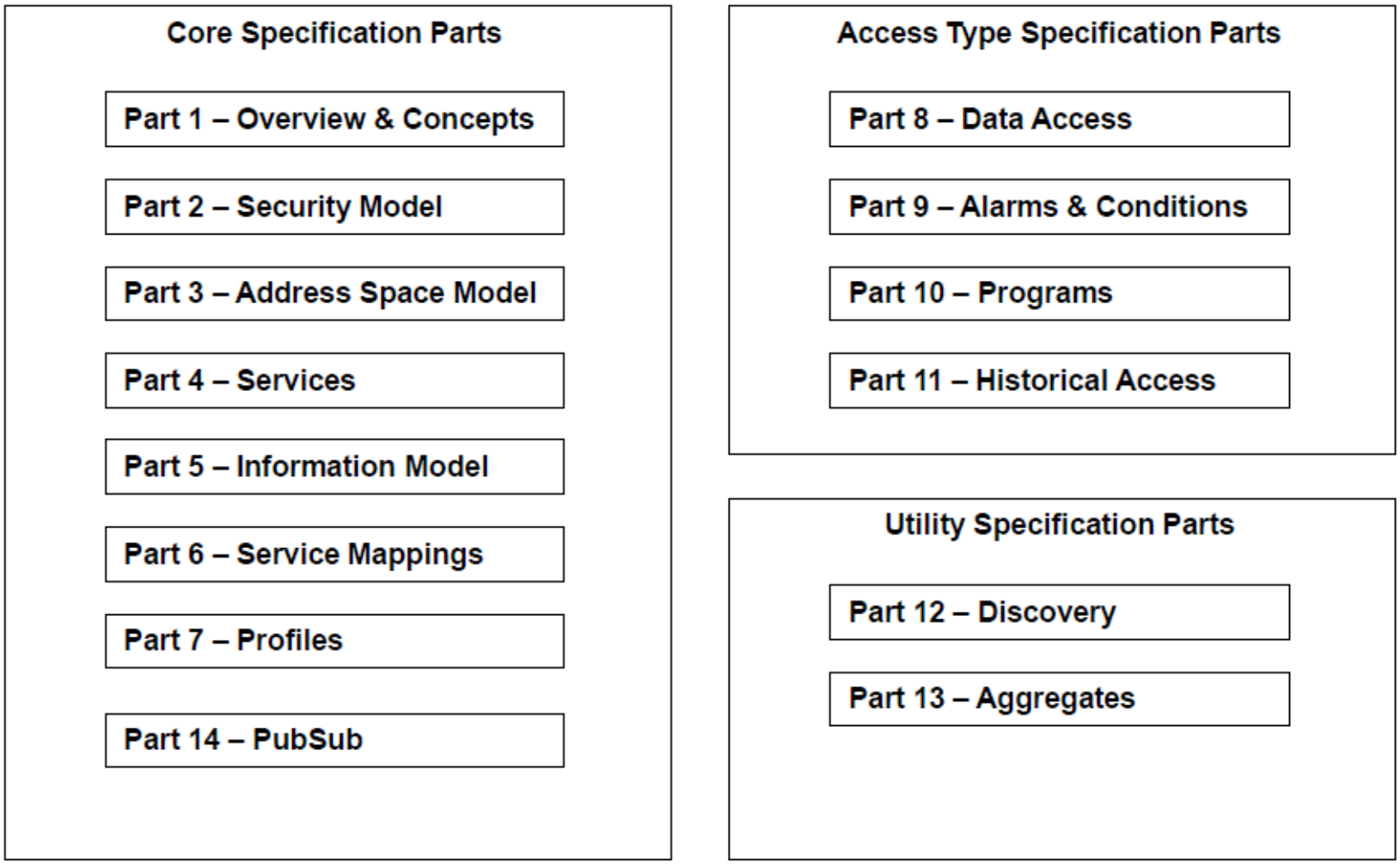


10 OPC Unified Architecture (UA)

- **Why?** (i) Industrial IoT (Industry 4.0);
  (ii) OPC UA is an industrial standard; (iii) Pub/Sub approach related to MYNO project;
- OPC stands for Open Platform Communications
- OPC UA is a platform independent service-oriented architecture that provide services like discovery and aggregation, management, security, etc.

Links:
OPC UA `https://opcfoundation.org/developer-tools/specifications-unified-architecture`

OPC UA Multi-Part Specification



Overview of all Topics

1. **Eclipse Vorto Project from Bosch**
2. **Eclipse hawkBit - OTA Updates from Bosch**
3. **Mozilla IoT and W3C Web of Things**
4. **OMG Data Distribution Service (DDS)**
5. **OMA Lightweight M2M (LWM2M)**
6. **Open Connectivity Foundation (OCF) Framework**
7. **ZigBee Protocol**
8. **openHAB Software**
9. **Eclipse Kura and Kapua**
10. **OPC Unified Architecture (UA)**

Literature and Books

[WSN] Protocols and architectures for wireless sensor networks, H. Karl and A. Willig, John Wiley & Sons, 2007

6LoWPAN: The wireless embedded internet, Z. Shelby and C. Bormann, John Wiley & Sons, 2009

IoT in five days, Antonio Liñán et al., `https://github.com/marcozennaro/IPv6-WSN-book`, 2016

Building the Web of Things: With examples in Node.js and Raspberry Pi, D. Guinard and V. Trifa, Manning, 2016

"Present to inform, not to impress; if you inform, you will impress."
- Frederick P. Brooks, Jr.