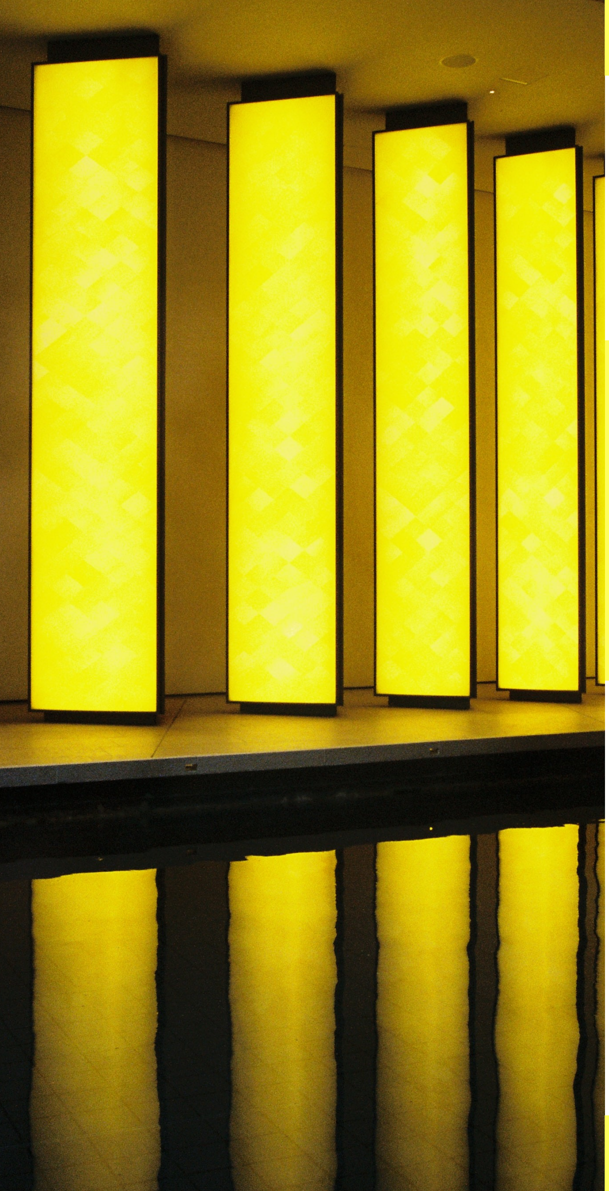


Distribution of Components in Event-Driven Sensor Networks

Seminar presentation

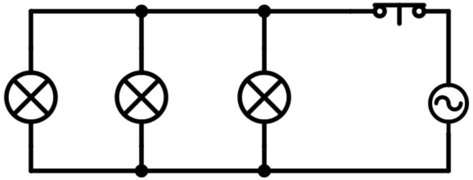
Bc. Miroslav Hájek
Software Architecture

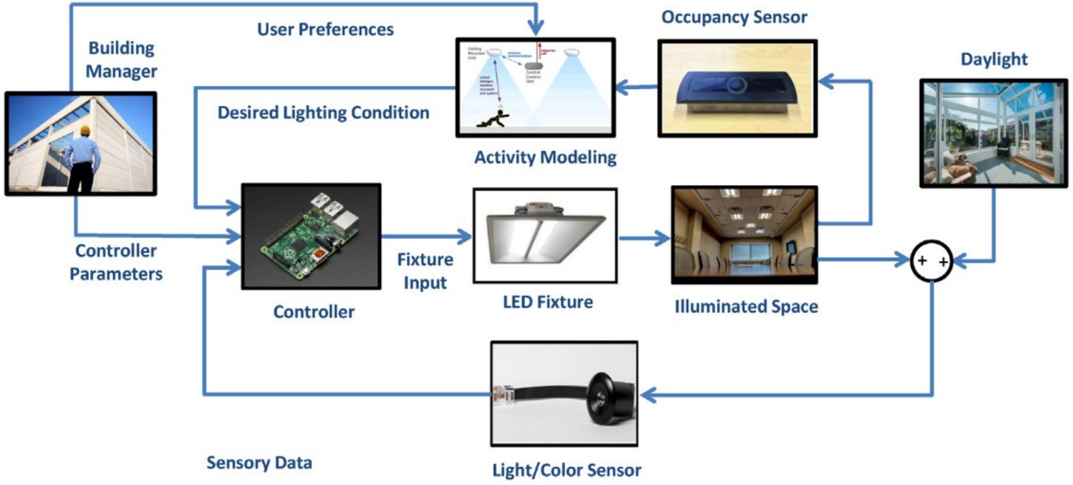
Topics outline

- **Lighting control** system design
- **Event-based messaging**
- **Buliding Management Systems IoT architectures:**
  - CONDE, SorBet
- **OpenAIS reference architecture**
  - ODM: Object data model
  - Network stack
  - OGC: Object Group Communication
  - Deployment variations
- **Example app design** using OpenAIS

Lighting System in Smart Building

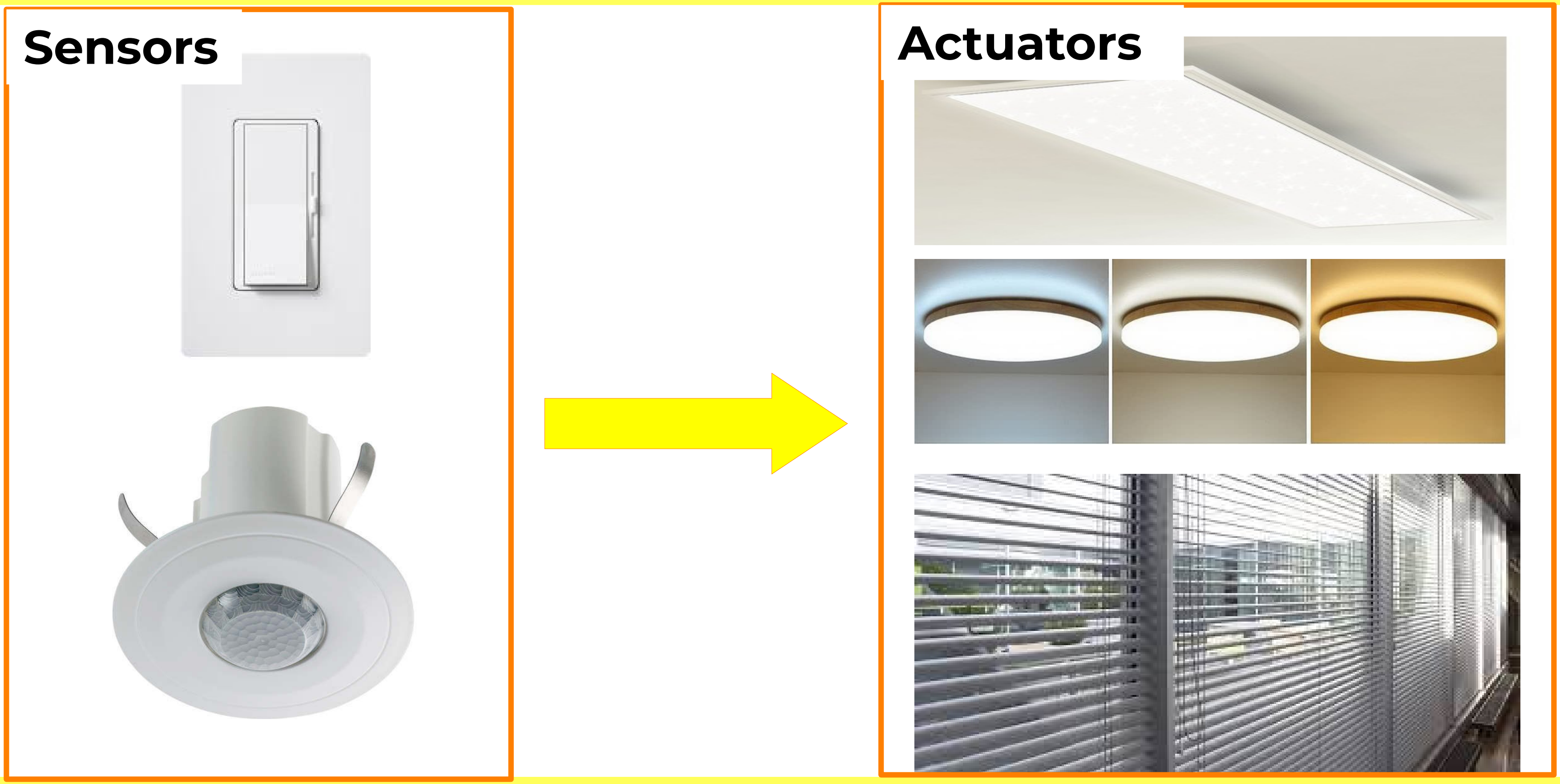
- **Part of Building Management System**
- **Objectives**
  - Uniform and bright illumination
  - Color matching
  - Inhabitant preferences
  - Power savings by:
    - Occupancy-based lighting
    - Daylight harvesting





Survey of feedback control methodologies: https://doi.org/10.1016/j.enbuild.2016.08.088

Hardware components for Lighting

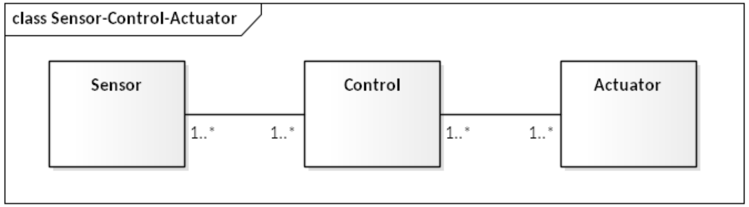


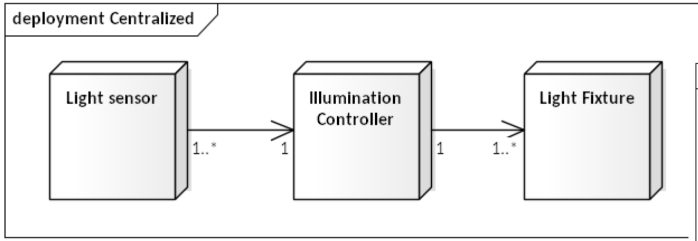
Sensors

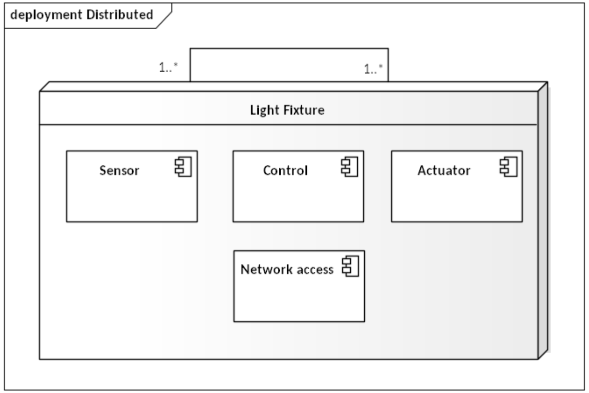
Actuators

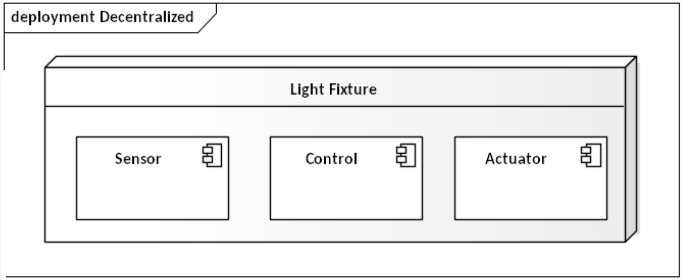
Control Architectures for Lighting

- **Centralized:** all sensors → single controller → all fixtures
- **Decentralized:** each fixture has all components
- **Distributed:** local communication between fixtures



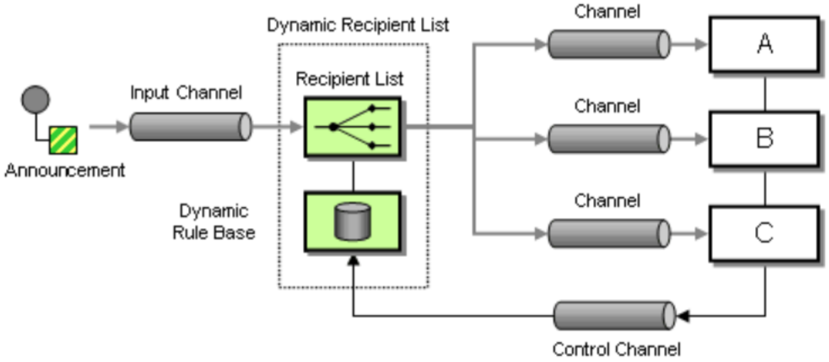






Event based system

- **Building blocks:**
  - Publisher (Producer) → Message → Channel →Subscriber (Consumer)
- **Design Patterns:**
  - Mediator – decoupling of components
  - Observer – notitication on update



- **Event-driven SOA**

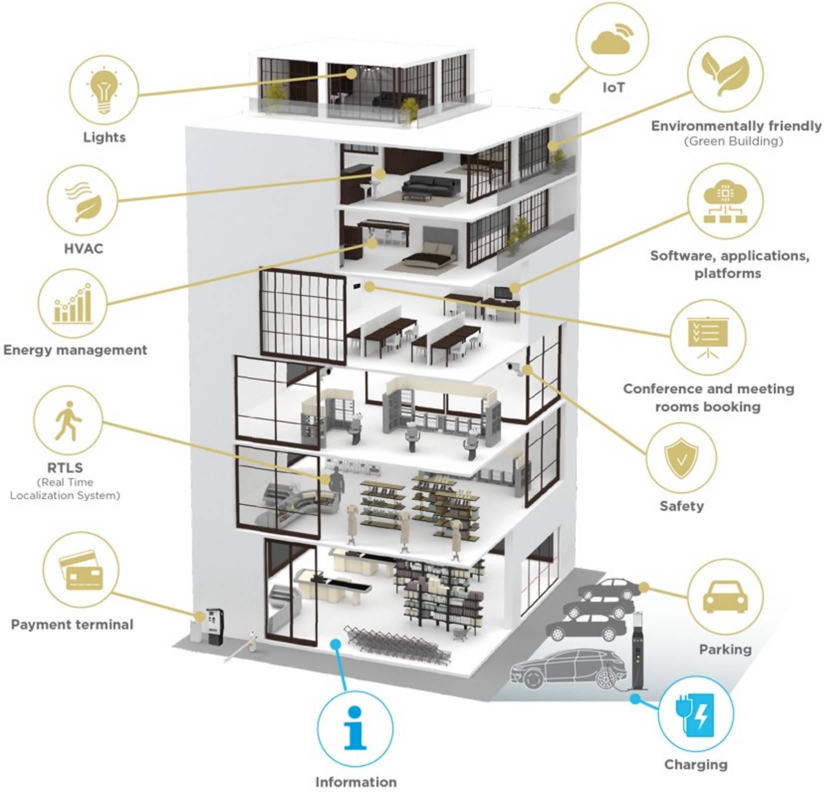
  1) Event can trigger the invocation of service(s)

  2) Service execution can produce new events

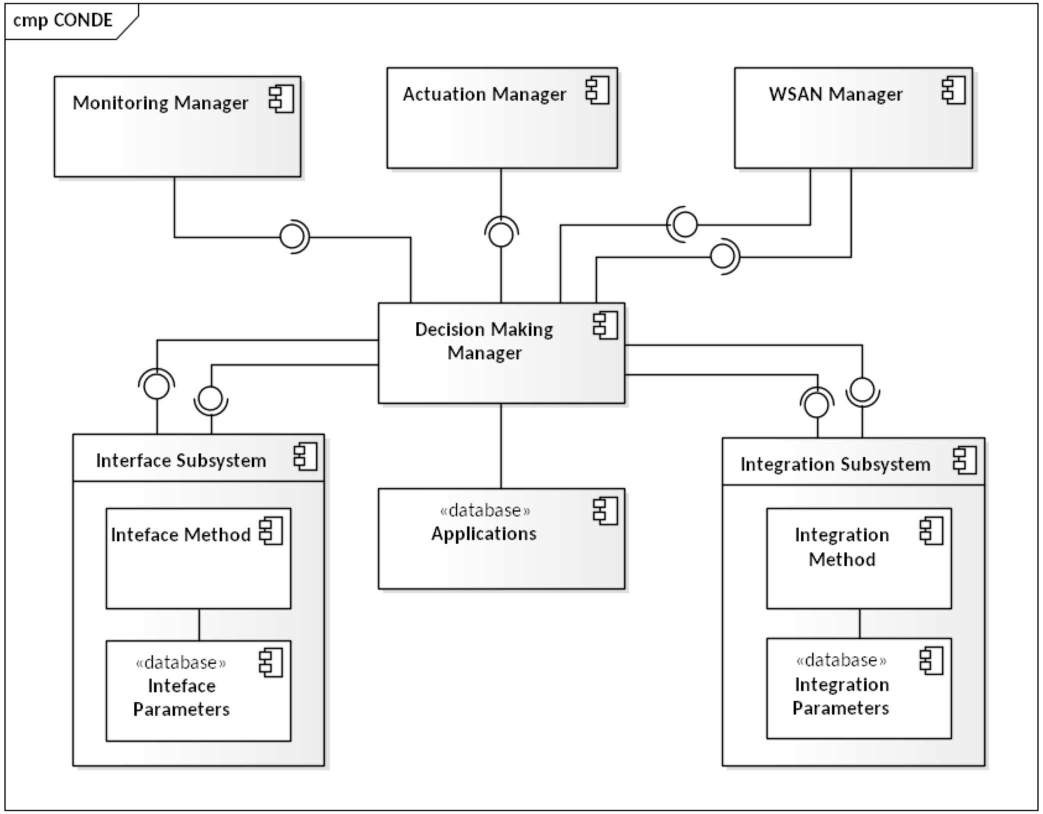
Building Management System (BMS / BAS)

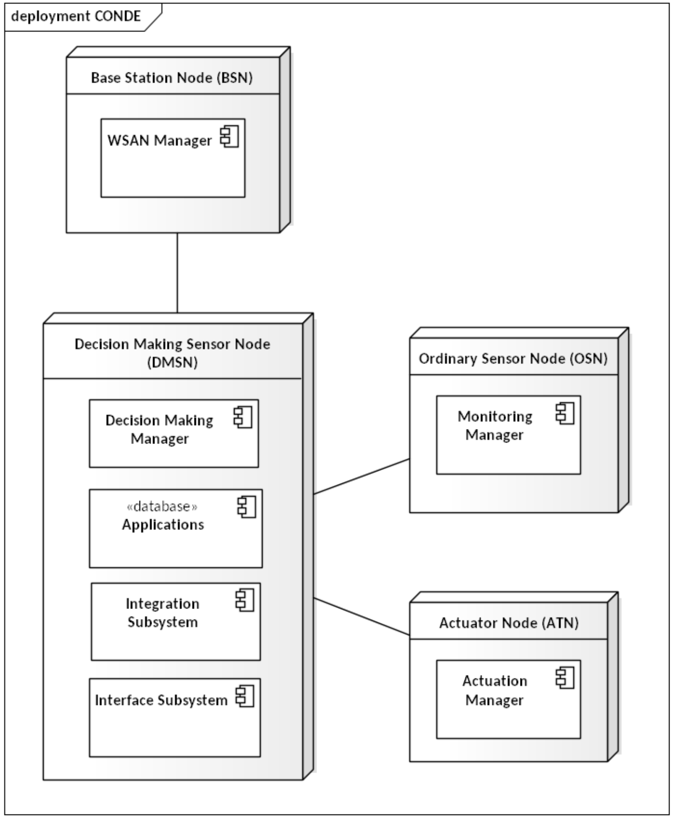
- **Building Automation:**
  - sensing the environment
  - making decisions
  - acting when needed transparently

- Centralized control - single point of failures
- Mainly wired communication
- Human intervention to configure new devices
- Proprietary technology devices
- Non-scalable architectures
- Isolation approach to security

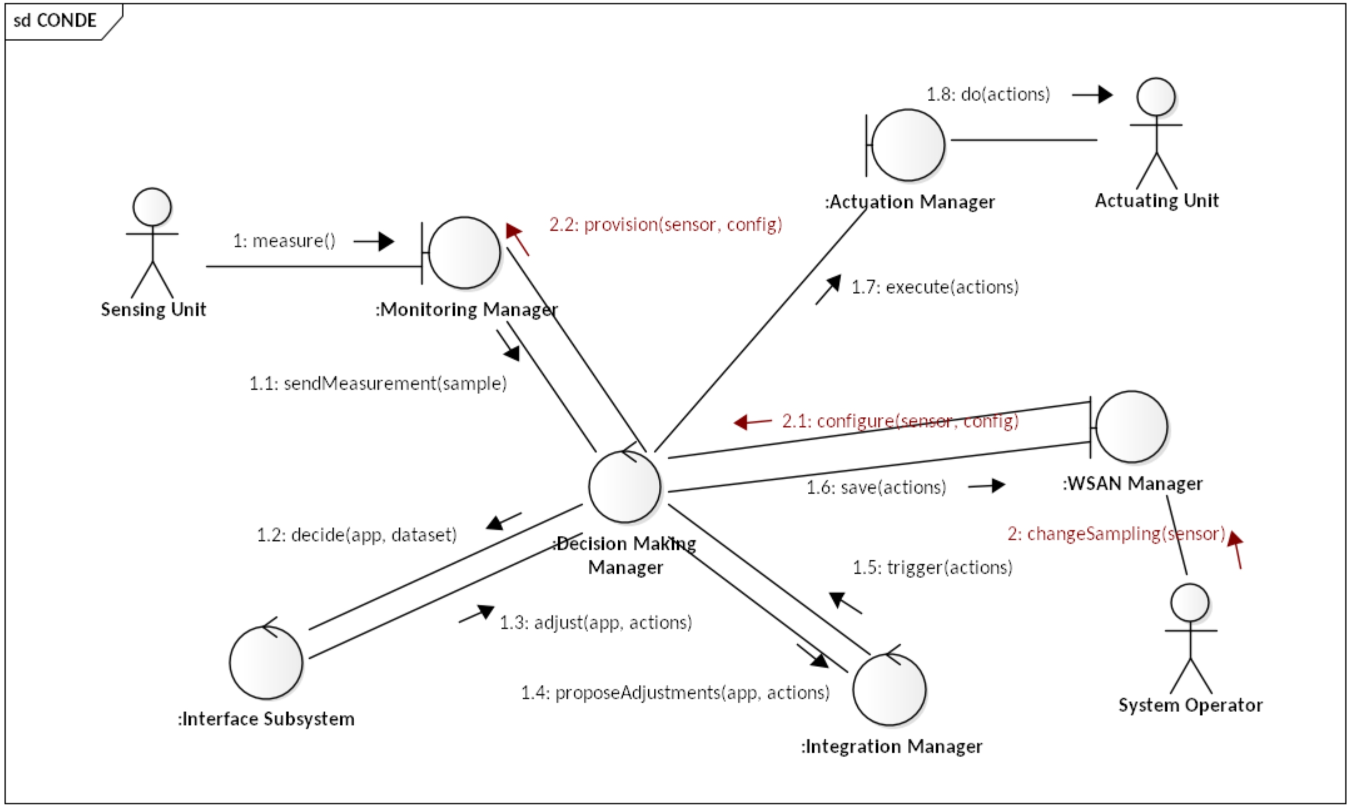


CONDE – Component view





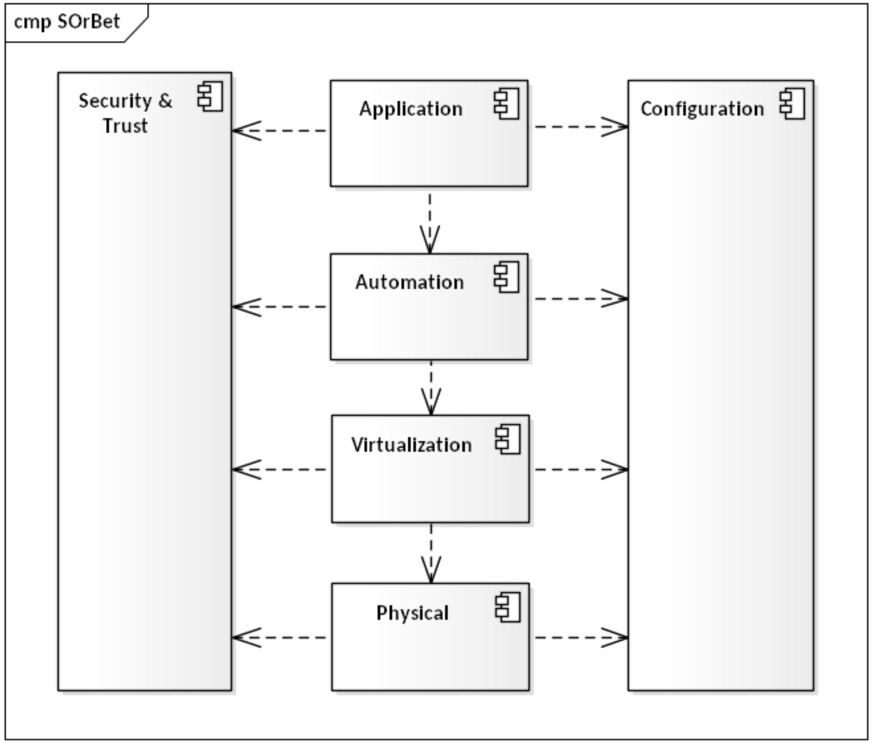
CONDE – Communication view

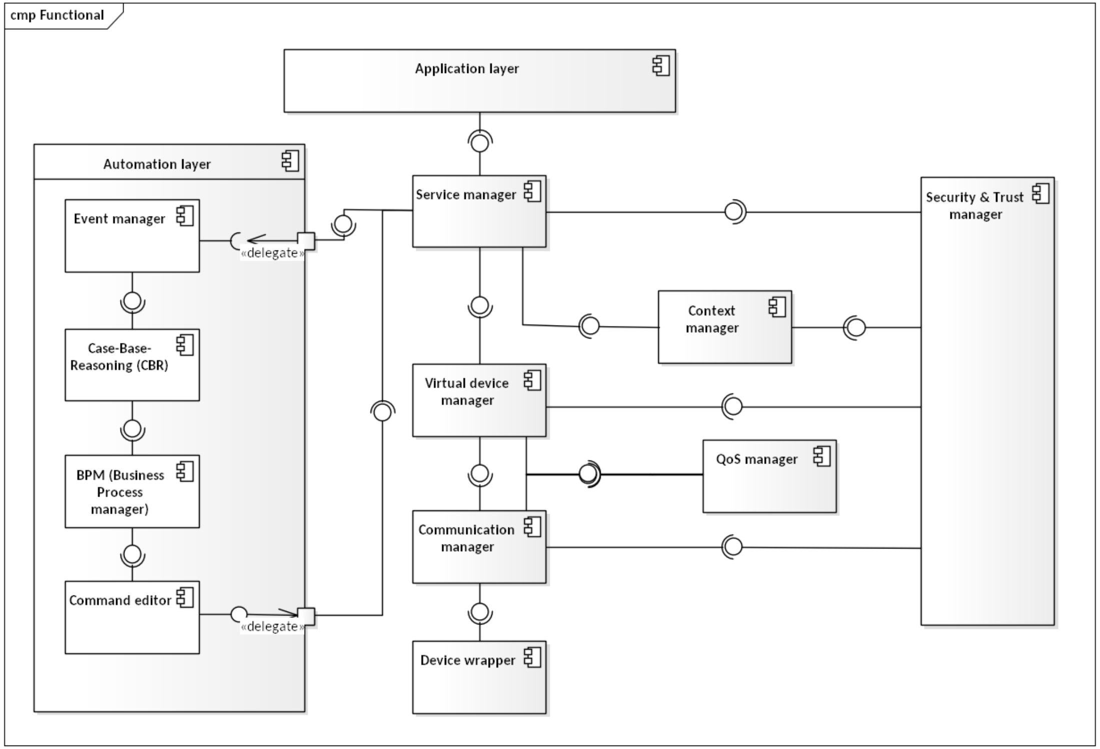


Decentralized system

- for decision and control
- in smart building apps
- using WSAN

SOrBet Architecture



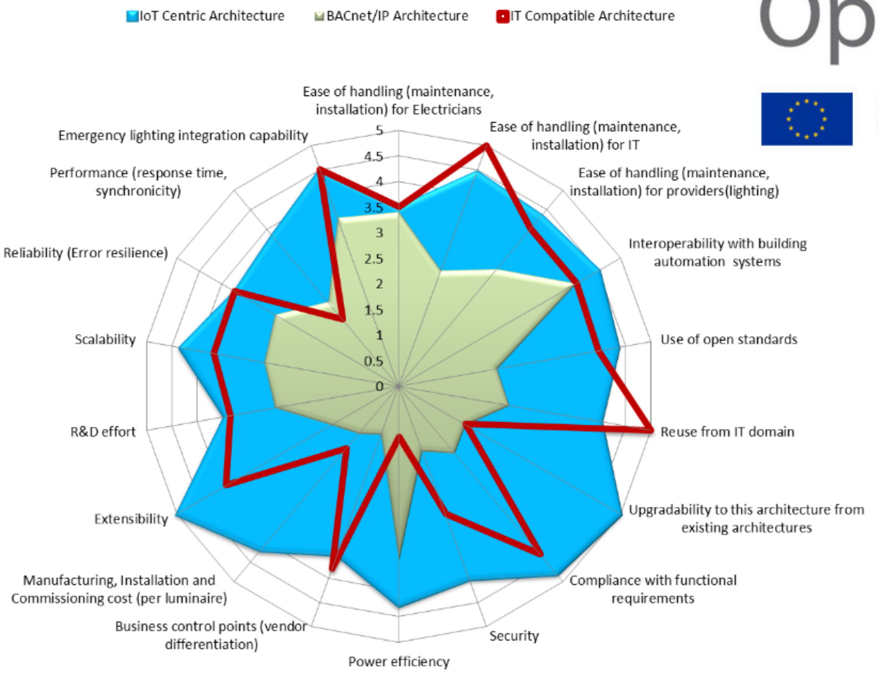


OpenAIS Reference Architecture

- Horizon 2020 project (https://cordis.europa.eu/project/id/644332)
- Open Architectures for Intelligent Solid State Lighting Systems (http://openais.eu/en/results)

Supported by the Horizon 2020 funding of the European Union

- *“OpenAIS aims at setting the **leading standard** for inclusion of lighting for professional applications in to IoT, with a focus on **office lighting**. This will enable a **transition from** the currently existing **closed and command oriented lighting control systems**”*



predecessor e.g.: DALI (Digital Addressable Lighting Interface)

Pilot project validation in office space

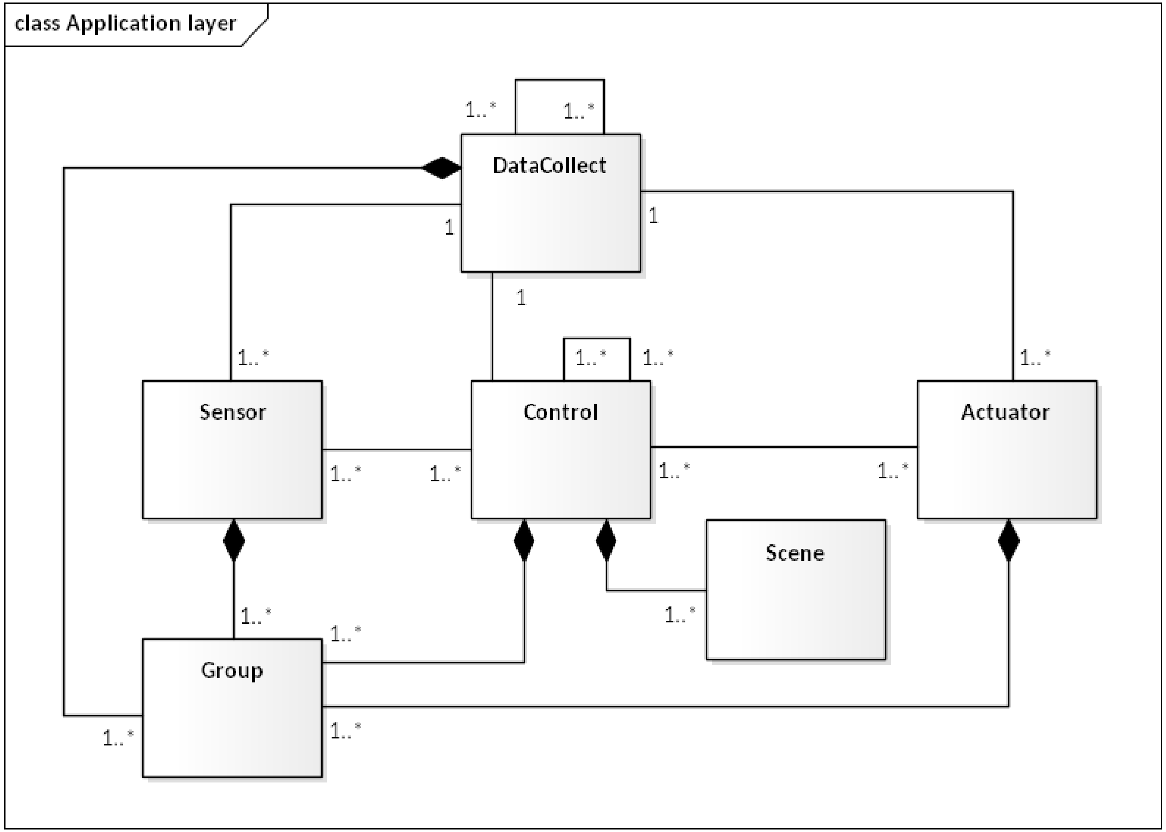
- Installation in Witte Dame, Eindhoven in 2018
- 400 luminaires with embedded sensors

OpenAIS IoL Reference Architecture

- **Key objectives:**
  - Define an open architecture with standardized APIs
  - Interoperable with BAS, cloud services and other systems
  - Increase the building value by combining IoT, LED technology and smart grids
  - Easy to specify, buy, install, maintain and use for all stakeholders in the value chain

- **Viewpoints for stakeholders:**
  - **Logical** - logical functions and their relations
  - **Physical** - mapping of logical functions to real SW and HW
  - **Networking** – communication stack and protocols
  - *Security – requirements based on LWM2M (CIA triad)*

Logical view: ODM – Application layer



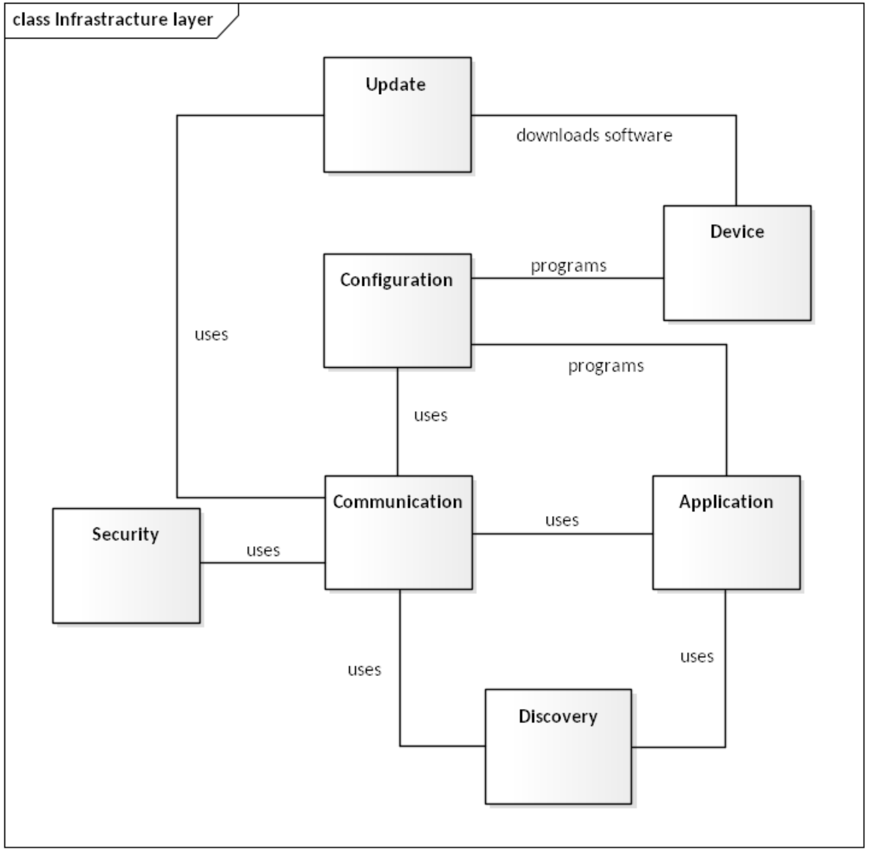
- **Actuator** – light fixtures
- **Sensor** - presence, light-level
- **Control** – lighting controller
  - *Stacked control* (priority)

- **DataCollect** – collection, storage
- *Support* - device / function
  Grouping and **Scene** setting
- **Gateway** – legacy system interface

Structural support for:

- Central controller to all lighting
- Fully distributed control in every single luminaire

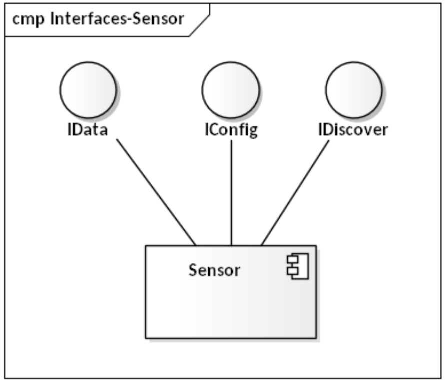
Logical view: ODM – Infrastructure layer



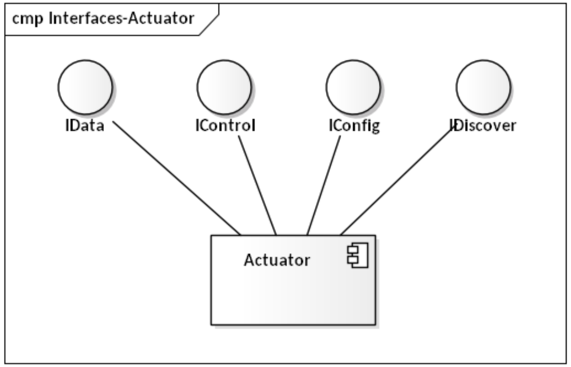
- **Communication** – network stack (L1 – L6 OSI)
- **Discovery** – of all available application functions *(Sensor, Control, Actuator, DataCollect)*
- **Device** – properties of a physical device
- **Configuration**
  - configuration parameters of the system
  - modify only by commisioning tool
- **Application** – OpenAIS application layer
- **Security** – not isolated, but in entire system
- **Update** – remote software updates

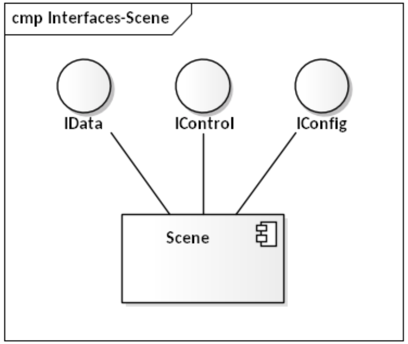
3.3.1.3 and 3.3.3 OpenAIS - p. 44

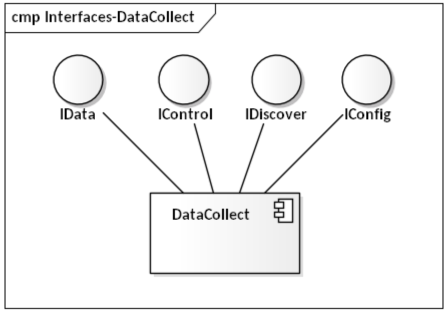
Logical view: ODM – Interfaces



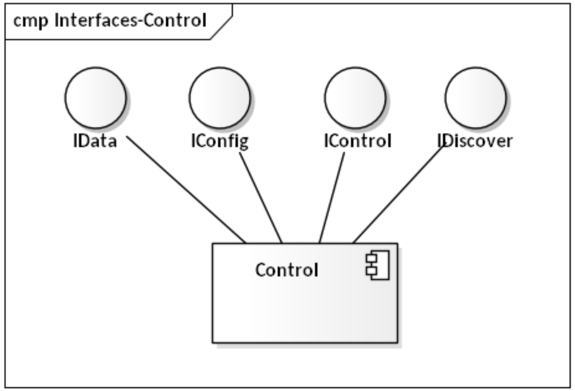
- **IData** – *(getter)* produce and send measurements
- **IControl** – *(setter)* execute funtion based on caller parameters

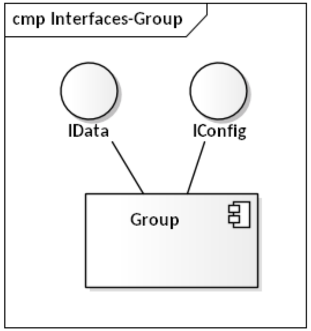






- **IDiscover** – advertise presence on network
- **IConfig** – commisioning, algorithmic, regulation settings





3.3.4 OpenAIS - p. 52

Networking view

Recommended MAC :

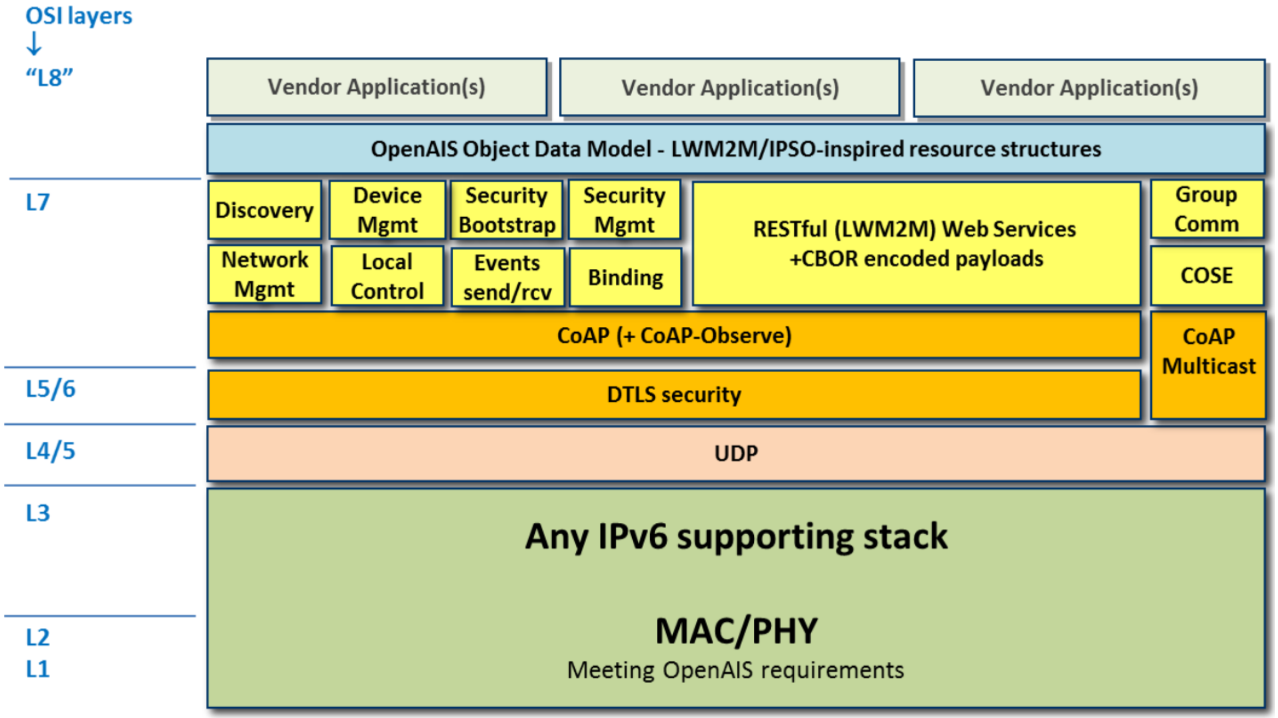
- Wired: Ethernet
- Wireless: 6LoWPAN / Thread

Transport:

- IPv6 (+ Multicast)
- UDP + DTLS

RESTful services

- CoAP (+ Multicast, + Observe)
- LWM2M
- CBOR format



OSI layers ↓ "L8"

Vendor Application(s) | Vendor Application(s) | Vendor Application(s)

OpenAIS Object Data Model - LWM2M/IPSO-inspired resource structures

L7

Discovery | Device Mgmt | Security Bootstrap | Security Mgmt | RESTful (LWM2M) Web Services +CBOR encoded payloads | Group Comm

Network Mgmt | Local Control | Events send/rcv | Binding | COSE

CoAP (+ CoAP-Observe) | CoAP Multicast

L5/6

DTLS security

L4/5

UDP

L3

Any IPv6 supporting stack

L2
L1

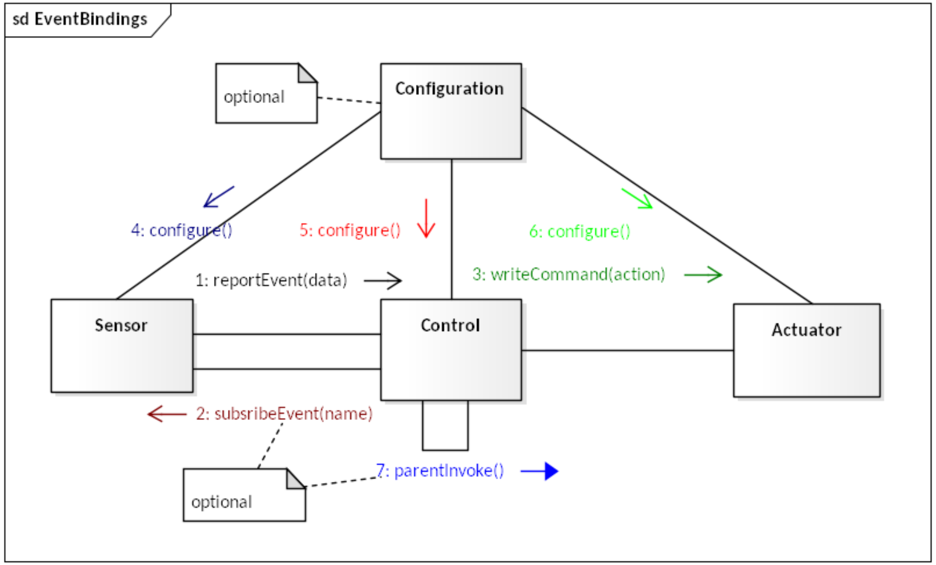
MAC/PHY

Meeting OpenAIS requirements

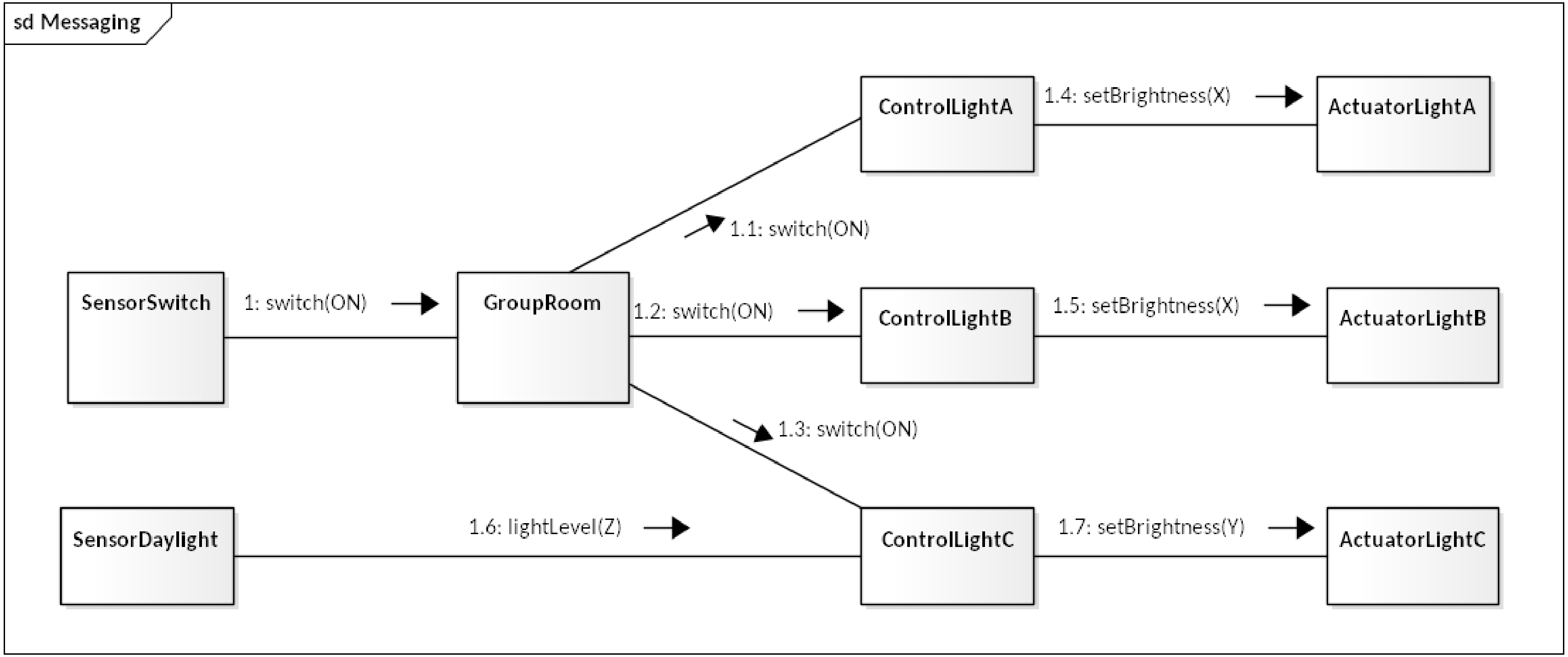
3.5 OpenAIS - p. 81 - 86

Event Messaging

- Sensors – *Time and Sequence number*:
  - **Push-button**: CLICK, HOLD, BREAK
  - **Presence detector**: PRESENCE, NO PRESENCE
  - **Light sensor** : BRIGHTNESS, STATUS
- Control commands:
  - **Absolute settings:** *go to level x*
  - **Relative settings:** *step up x*
  - **Referenced settings:** *scene x recall*
- Actuator:
  - Listen to Control commands
  - Regular status report to Control



Group Communication Example



3.3.5.4 OpenAIS - p. 60

CoAP API - URI Path Structure

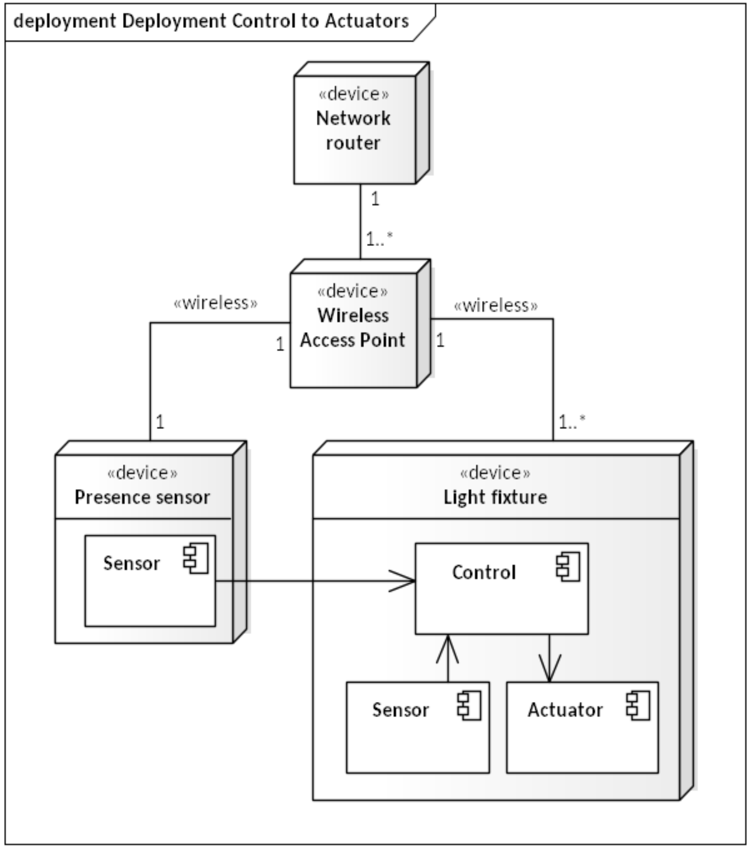
- **Configuration access** to OpenAIS Group Communication
  - `/4006/<object-instance-id>`
- **Access multiple group members** *(CoAP multicast or serial unicast)*
  - `/g/<object-ID>/<group-ID>/<resource-ID>`
- **Access single Object instance**
  - `/s/<object-ID>/<object-instance-ID>/<resource-ID>`

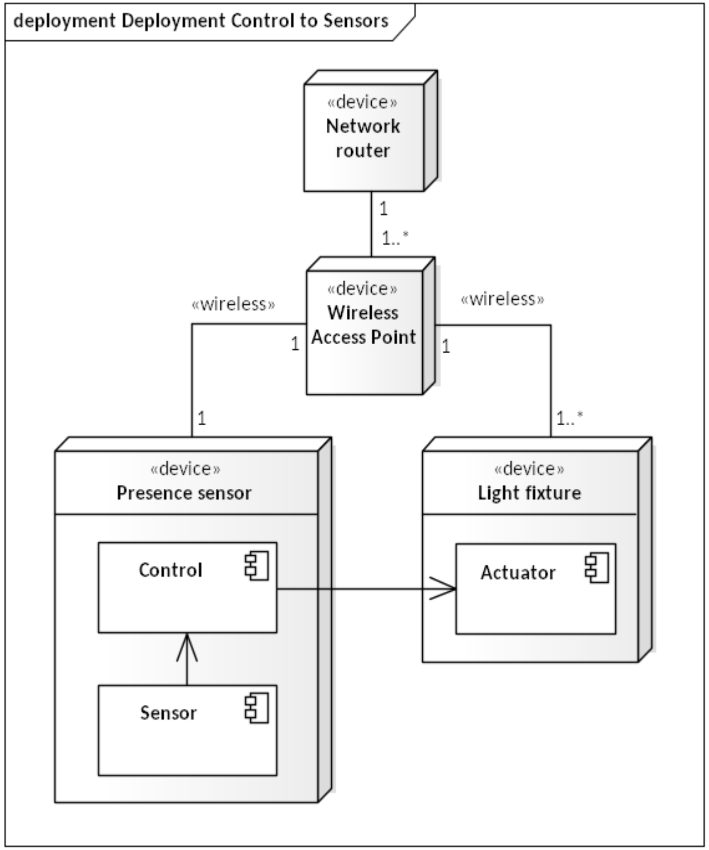
Object & Resource ID: Short IDs – 16-bit integers – LWM2M Registry & OpenAIS Appendix A

Example: Object: *Logical Light-Point Actuator,* **Resouce:** *Switch*

- **/#/4001/#/117**
- *POST coap://<target IP address>/<dir>/<Obj ID>/<object-instance>/117 {0:false, 2:2}*

Physical view - Deployment

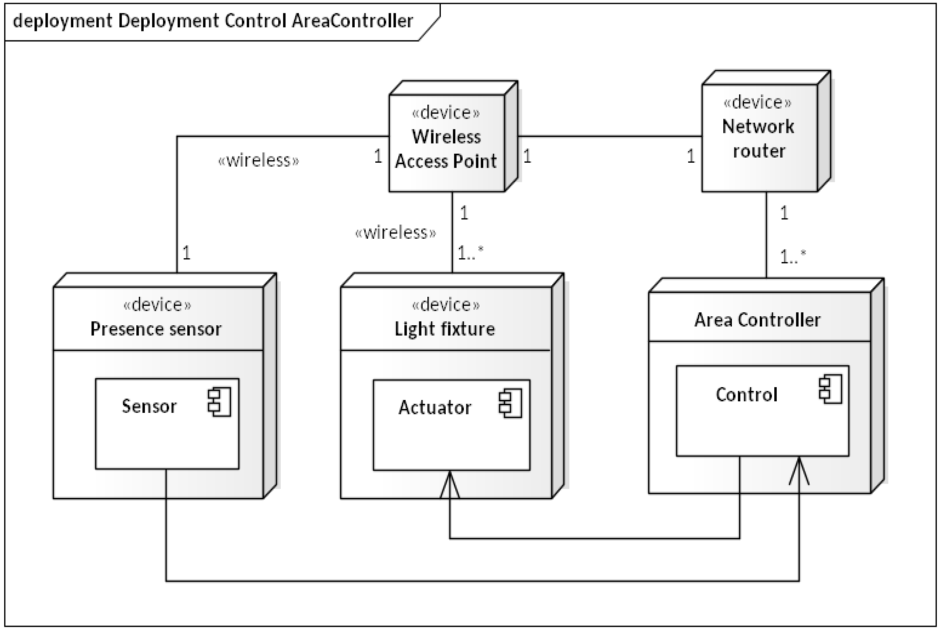


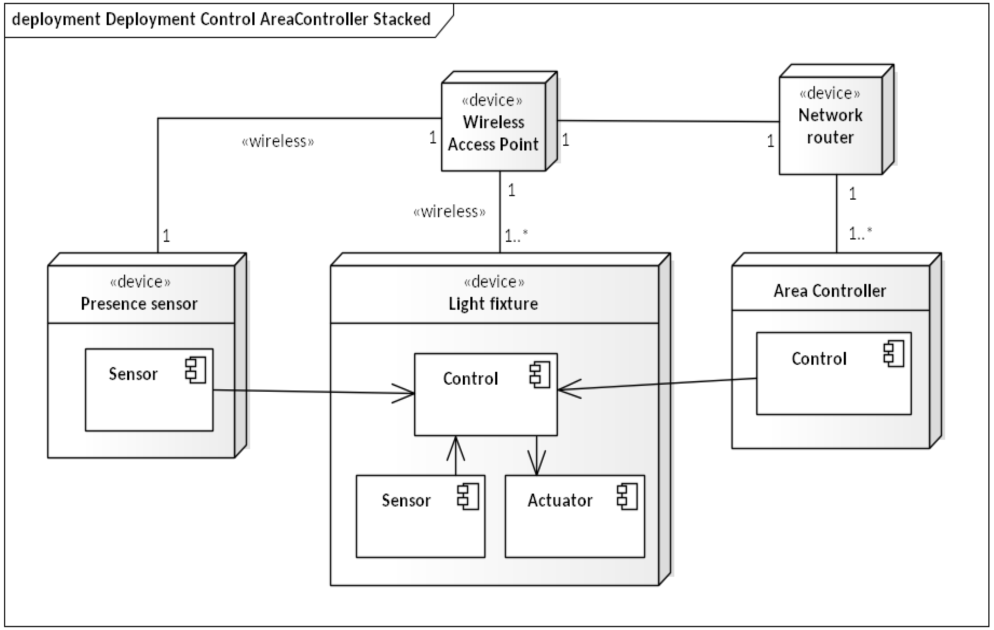


4.3. OpenAIS - p. 144

Control function - no direct link to hardware

Deployment - Area controller





4.3. OpenAIS - p. 147

Control function - no direct link to hardware

Example OpenAIS Application Design

- **Goals:**
  - **Compare** deployment of **Control function** to nodes:
    - (1) Sensor, (2) Actuator, (3) AreaController *(slide 21, 22)*
  - **Recreate** Group communication example *(slide 19)*

- **GUI Applications in Rust with Docker network**
  - Logical Light-Point Actuator (/4001) – *Actuator*
  - Logical Illuminance Sensor (/4004) – *Sensor*
  - Logical Push Button (/4002) – *Sensor*

Necessary ODM resources

- **Rust + Device(FLTK) + Communication(COAP + CBOR)**
- **IDATA, ICONFIG, ICONTROL**

- **4006 "oA Group"**
  - 901 "Documentary Description" string get put
  - 600 "Application Group ID" uint16 get put
  - 602 "IP Addresses" multi-instances: [union] get put post
  - 603 "Members" multi-instances: [string] get put post
- **4001 "oA Logical Light-Point Actuator"**
  - 901 "Documentary Description" string get put
  - 100 "Target ON/OFF" Boolean get
  - 101 "Target Intensity" float get
  - 921 "Priority" uint8 get put
  - 106 "Dimming Time" uint16 get put
  - 109 "Step Size" float get put
  - 117 "Switch" structure post
  - 118 "Dim" boolean post
  - 120 "Step" structure post
- **4004 "oA Logical Illuminance Sensor"**
  - 901 "Documentary Description" string get put
  - 404 "Sensor Value" uint16 get
  - *400 "Less Than" uint16 get put (Optional)*
  - *401 "Greater Than" uint16 get put (Optional)*
  - *402 "Step" uint16 get put (Optional)*
  - 403 "Minimum Update Interval" uint16 get put
- **4002 "oA Logical Push-Button Sensor"**
  - 901 "Documentary Description" string get put
  - 202 "Push-Button Event Value" enumeration get
  - 203 "Single Click Time" uint16 get put
- **4012 "oA Status Report Structure"**
  - 850 "Status Report Structure ID" uint8 get put
  - 851 "Keys" multi-instances: [string] get put post
- **Same for every sensor and actuator:**
  - 903 "Application Group ID" uint16 get put
  - 904 "Status Resend Time" uint16 get put
  - 919 "Status Report Structure ID" uint8 get put