



# Open and semantically enhanced Middleware

Modelling Optimization of Energy Efficiency in Buildings for Urban Sustainability

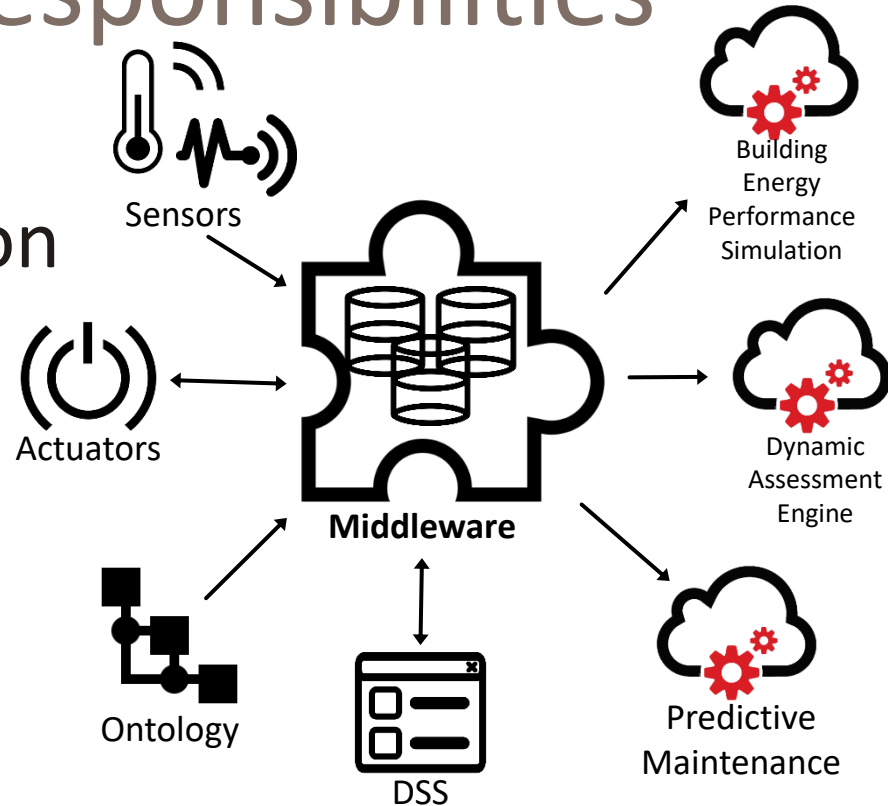
Project duration: November 2015 – April 2019



# Middleware responsibilities

- Unifying platform for the MOEEBIUS solution

- Storage and access to data for the SOA solution
- Providing publish/subscribe message broker for actuation
- Orchestration of the involved services by routing their calls



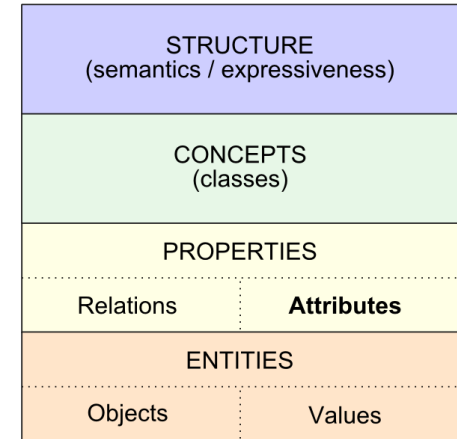
# Why Open is important?

- Not being tied to specific brand
- Based on open standards, not proprietary
- Adaptability to various (less than ideal) circumstances
- Usage of free and open source software





# Data context

- Data without a context is useless
- Semantically enhanced = context aware
- Contextual information about the sites, equipment, sensors and their hierarchy
- Ontology contains IDs for the data store for each measured object
- Using standard schemas for representation of the building metadata



# Middleware structure

- Based on Enterprise Service Bus (ESB)
  - Open-source implementation WSO2 ESB
  - Custom APIs and sequences to process incoming requests and messages
- RabbitMQ message broker
  - Incoming sensor data are immediately cloned to a topic on the message broker
  - Sensor data can be also sent to the message broker to be processed by the ESB
  - Used for actuation and real-time data viewing/processing

# Data storage

- NoSQL distributed wide column store Apache Cassandra
  - Open-source (free)
  - Scalable
  - Fault tolerant
- WSO2 Data Services Server (DSS) accesses data store to store data received via ESB and to retrieve data for other services
- Data information stored:
  - Point name
  - Country ID
  - Building ID
  - Timestamp
  - Value
  - Reliability

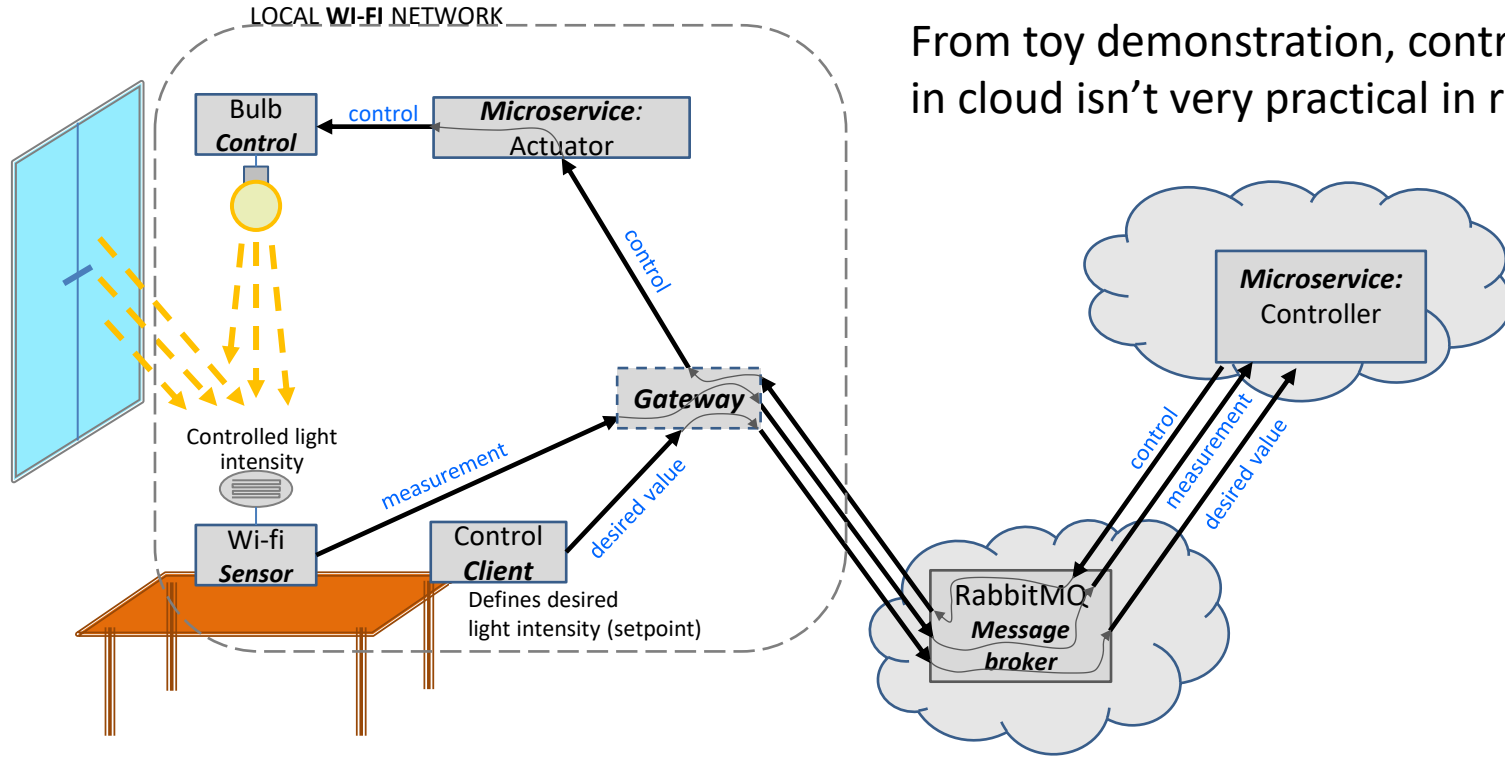


# Leveraging semantically enhanced Middleware in Predictive Maintenance

- Predictive Maintenance tasks need point names to retrieve correct data. These point names can be:
  - specified manually for each rule
  - generated dynamically from the ontology
- Dynamically populated rules simplify the introduction of PM:
  - Only one rule needs to be specified for a class of problem and is applied to all relevant equipment

# Actuation example

From toy demonstration, controller in cloud isn't very practical in reality





# MOEEBIUS Partners



## Project coordinator

Ander Romero Amorrortu  
TECNALIA  
Parque Tecnológico de Bizkaia  
ander.romero@tecnalia.com

[www.moeebius.eu](http://www.moeebius.eu)

