

IoT in the Smart Home

ETSI Workshop

21-22 March 2016

Opening a smart home infrastructure to 3rd party applications and devices

Sébastien Bolle (speaker), David Excoffier, André Bottaro
Smart Home and IoT Research, Orange Labs



<http://openthebox.org>

@oopenthebox 



The Smart Home market

Seven application domains

Security, energy, comfort, health, wellness, multimedia, games

Emergence of new products

Smart – Easy to install – Wireless

Smart Objects: thermostats, scales, keys, cookers, ...

Smart Appliances: washing machine, dryer, fridge, ...

Affordable self-install systems: security monitoring, smoke detection, ...



A new world of applications will emerge on the long term from the variety of sensors, actuators, devices that become available

Players expectations in the future smart home ecosystem

| | |
|--|--|
| Users | A home open to any service, any thing, assisting them in their everyday life with a unified multimodal customer experience |
| Service providers | sell services on the largest market, i.e., international, over the top deploy services without any entry barrier (e.g., no hardware, common software skills) benefit from infrastructure services provided by others |
| Device providers | sell connected devices on the largest market provide devices for all services and their providers benefit from infrastructure services provided by others |
| Smart Home operator | sell infrastructure services: device management, device/application certification, enterprise support partner with all providers to offer rich app and device stores deploy the infrastructure in any home (i.e., over the top) |
| 3rd party software developer | develop apps on standard APIs – “write once, run everywhere” access to a large set of libraries |



services and applications



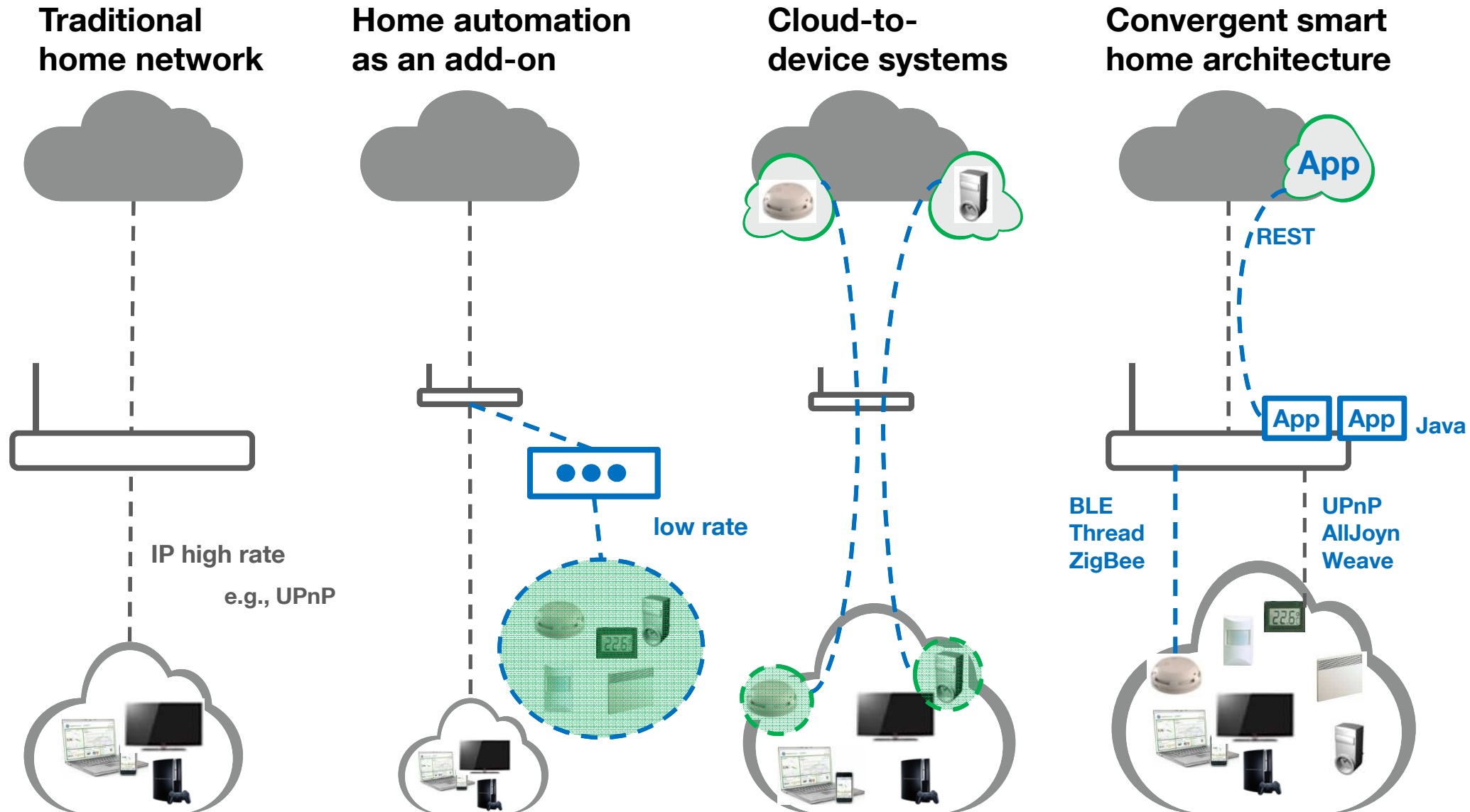
smart home infrastructure



devices

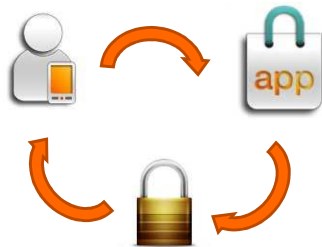
A hybrid execution environment

Applications in the box and the cloud



Technical challenges

A device and application representation synchronized between the box and the cloud



Management of user and application access rights to devices and applications at every level

Interoperability thanks to standards



oneM2M Common Service Layer in a nutshell

A software layer that

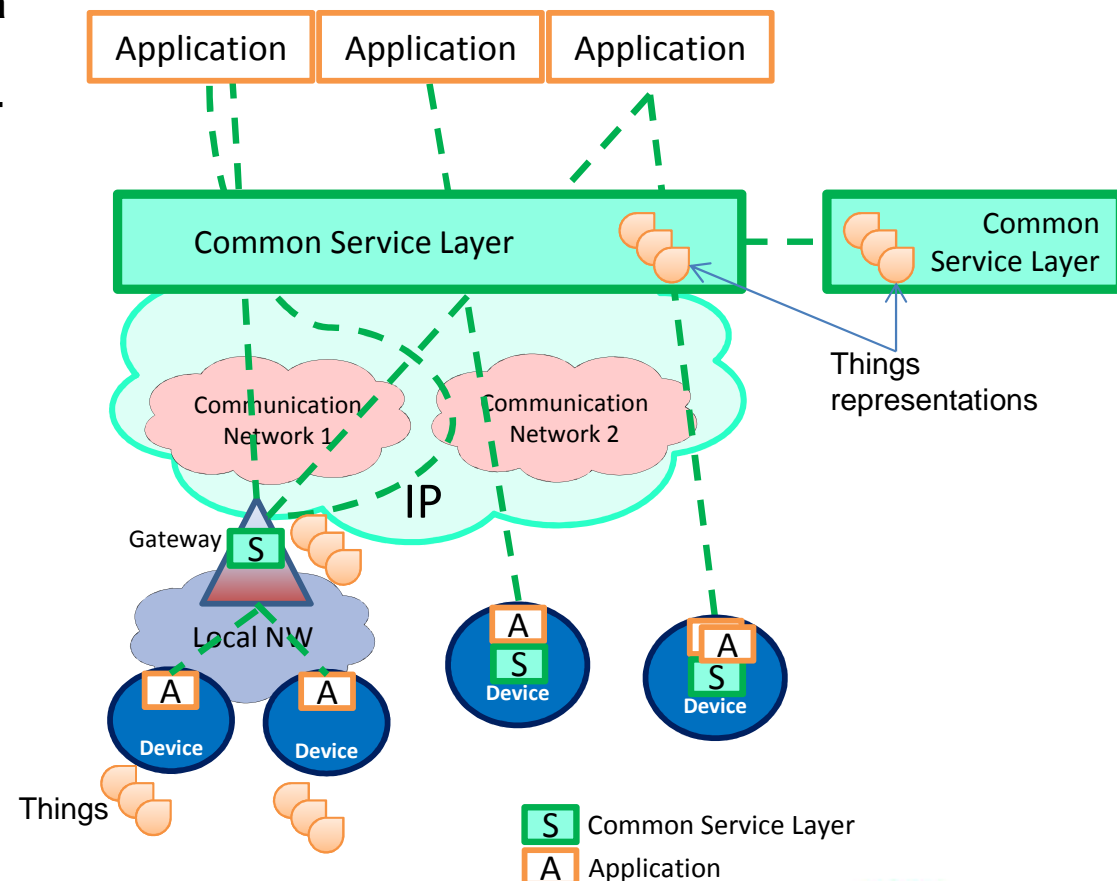
- is language agnostic.
- sits between M2M applications and communication HW/SW that provides data transport
- specifies a RESTful approach for M2M/IoT communication
- with a mapping to common industry protocols such as CoAP, MQTT and HTTP
- allows for distributed intelligence (device, gateway, cloud apps)

OneM2M has been created by 8 international and regional standard bodies: ARIB, ATIS, CCSA, ETSI, TTA, TSDSI, TTA, TTC

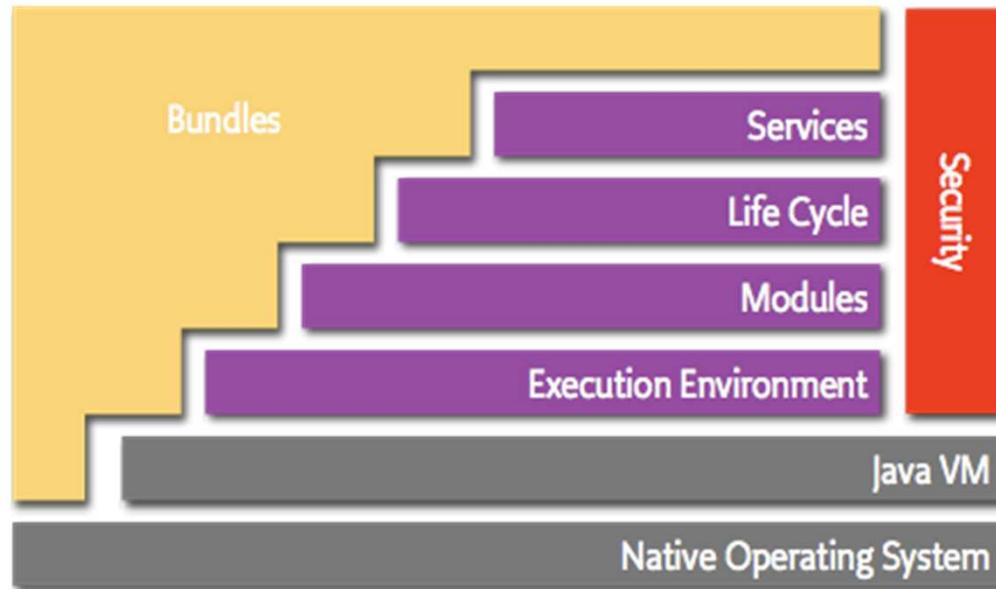
200 members contribute to oneM2M

Horizontal (based on common Layer)

Applications share common service and network infrastructure
Multipoint communications

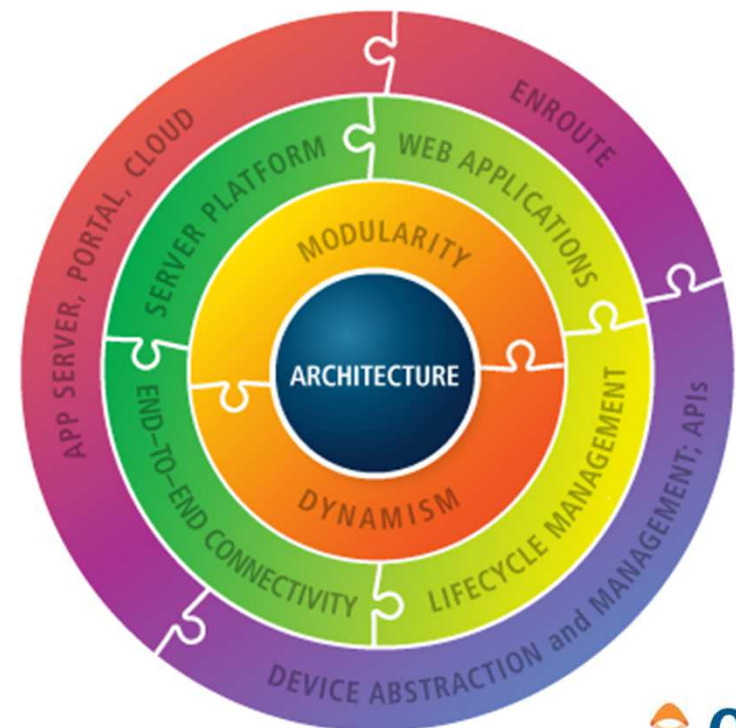


OSGi in a nutshell

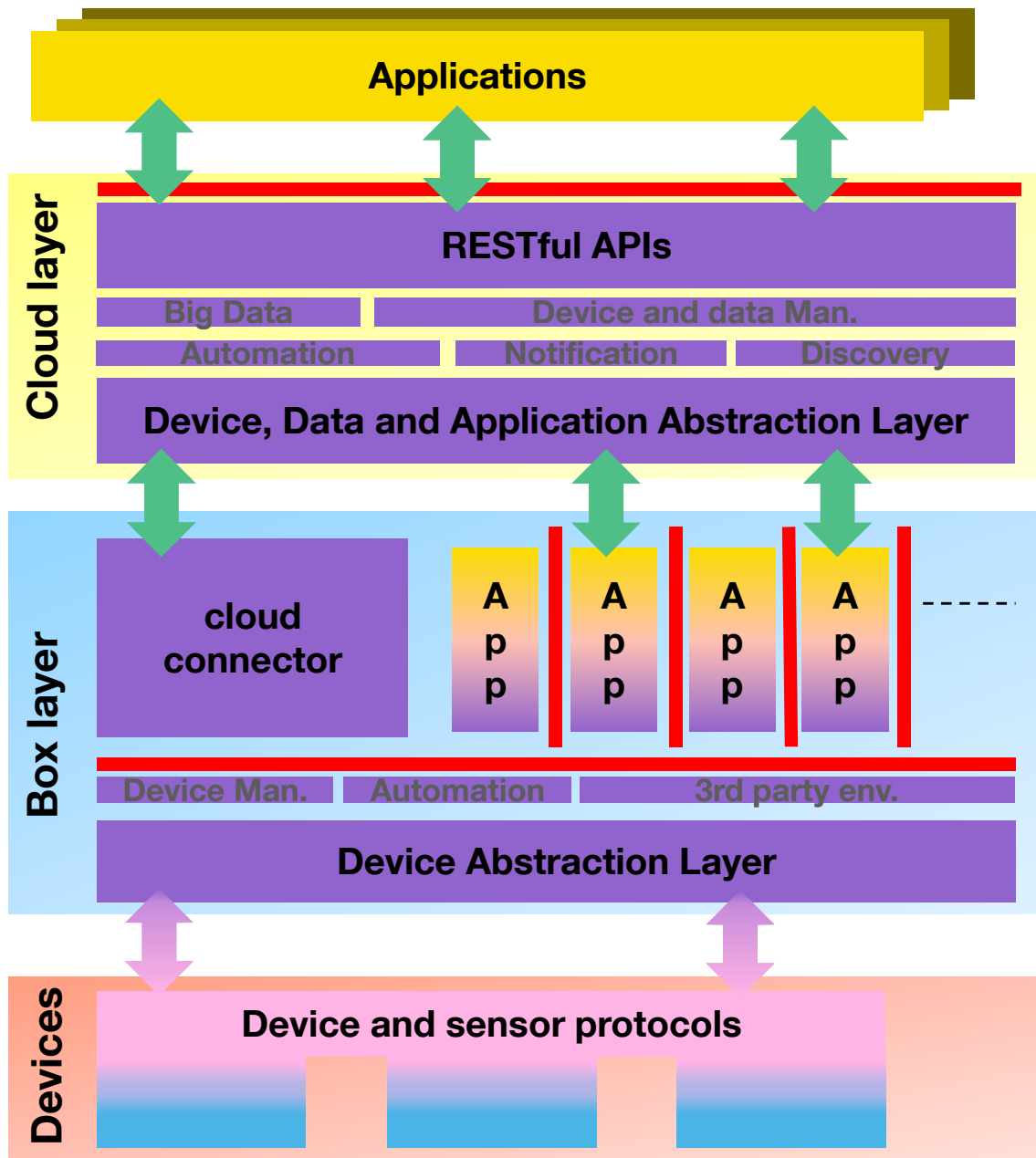


- **OSGi is a standardized software platform**
 - Generic framework based on Java
 - Hot deployment of service bundles
- **Open specifications with many standardized services**
- **Strong open source communities**
- **Openness to third parties**
 - Flexible cooperation model between bundles (service orientation)
 - Mechanisms to allow third parties to only access shared interfaces
 - Home and IoT: ideal for openness to third party apps and Device Management
- **Agile programming and software portability**

- **Founded in 1999 with Home requirements, now a success on Enterprise domain**
 - Led by IBM, many software editors, some operators (NTT, DT, Orange, Telecom Italia), and manufacturers (Huawei, Schneider Electric, NEC, Amdocs...) for Smart Home and IoT
- **Current orientations**
 - A residential expert group (REG) on Device management, Smart Home technologies APIs and Device Abstraction
 - A recent IoT Expert Group launched in 2015



An open and standard Smart Home architecture



apps, data, devices are discovered as resources by 3rd party cloud apps

access rights on devices and embedded apps are first checked at cloud level



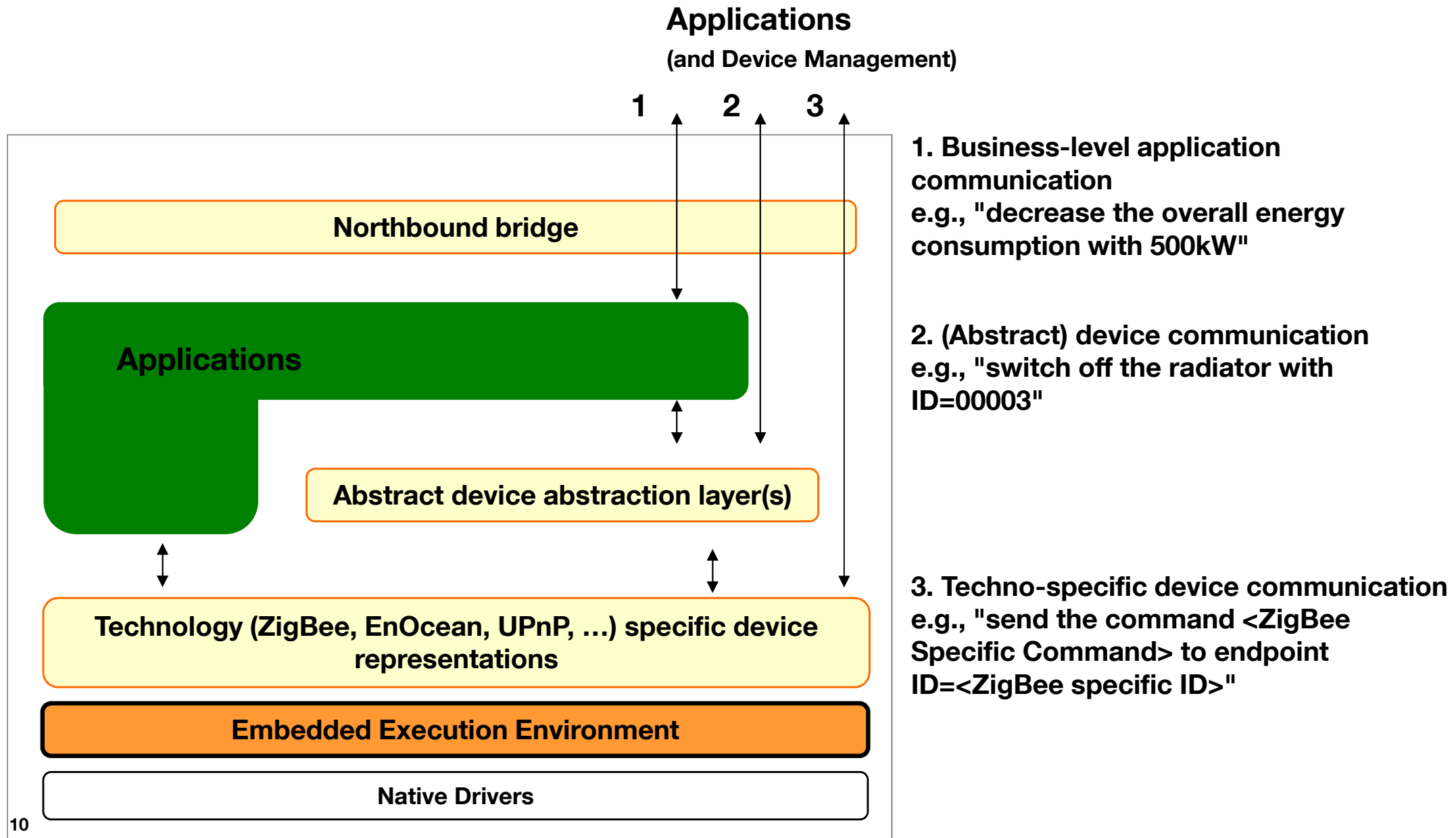
apps and devices are discovered as services by embedded apps

embedded and Cloud apps expose external services (APIs) whose access rights are checked

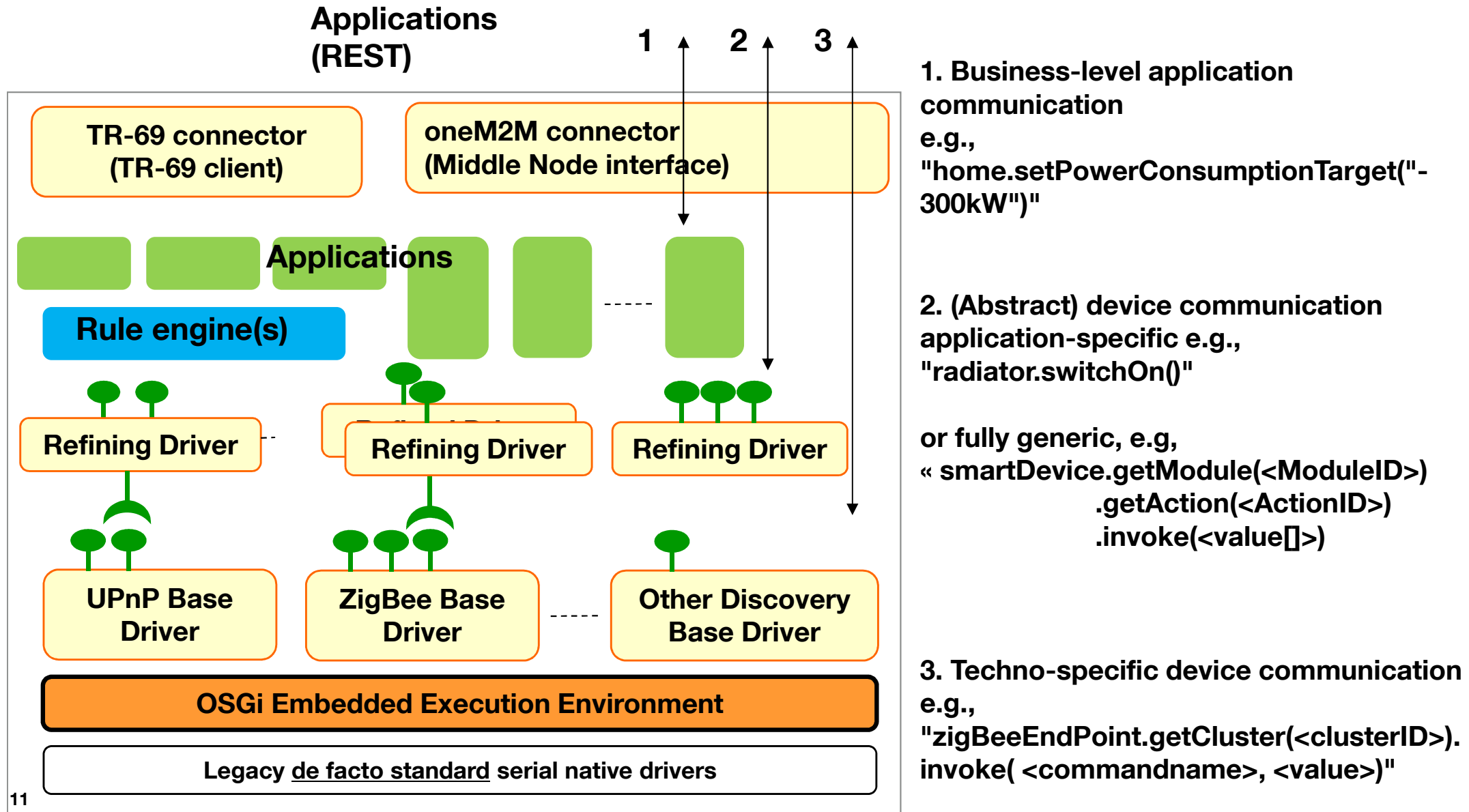
local access rights on technical services and networked devices are checked at the embedded framework level

Device abstraction layer

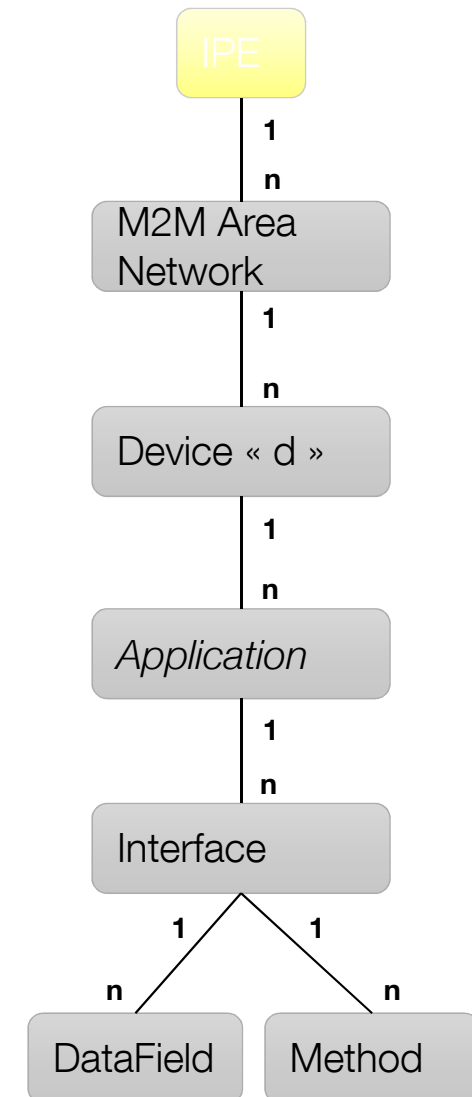
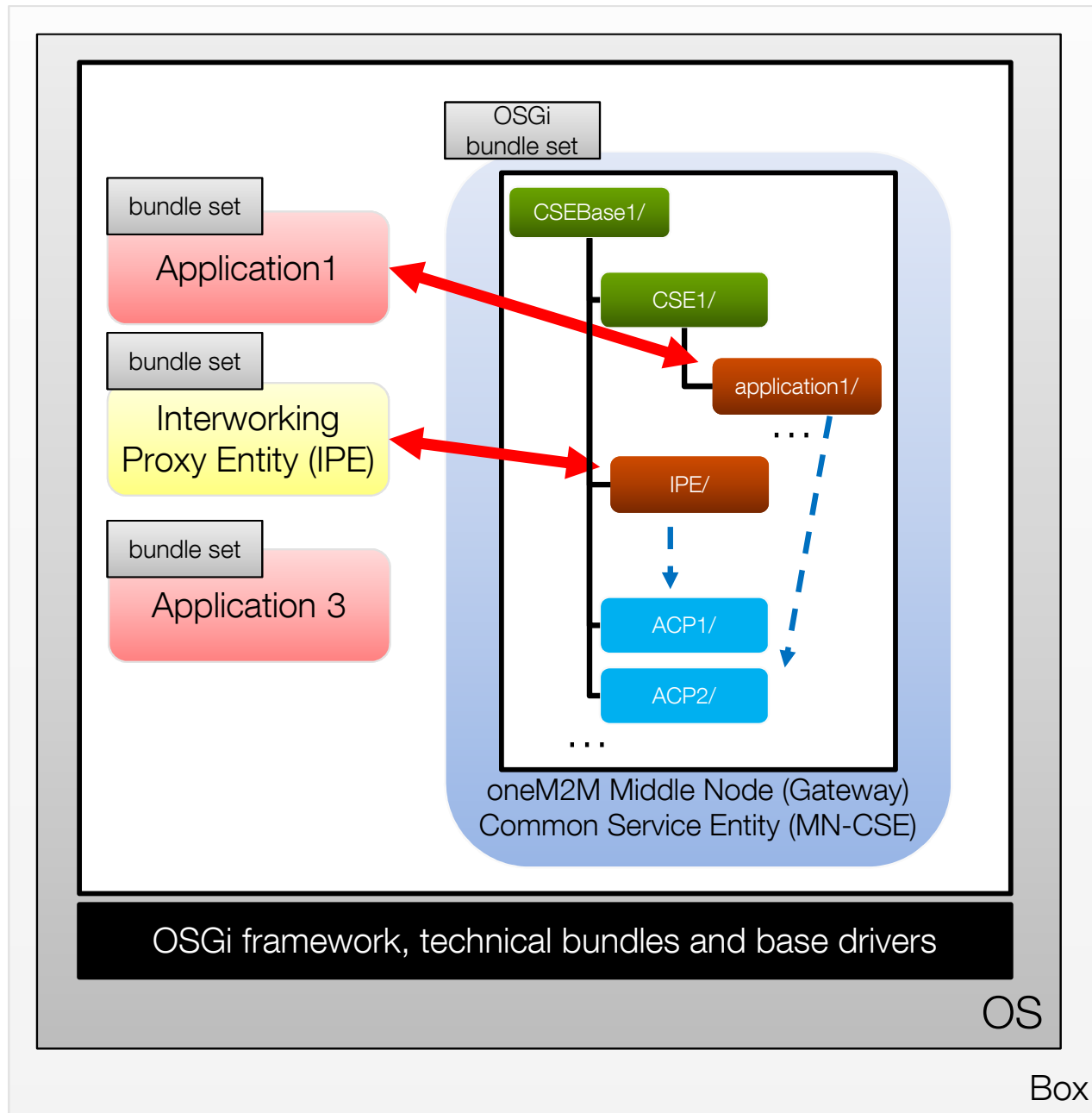
3 levels to represent local home devices



3 levels to represent home connected things

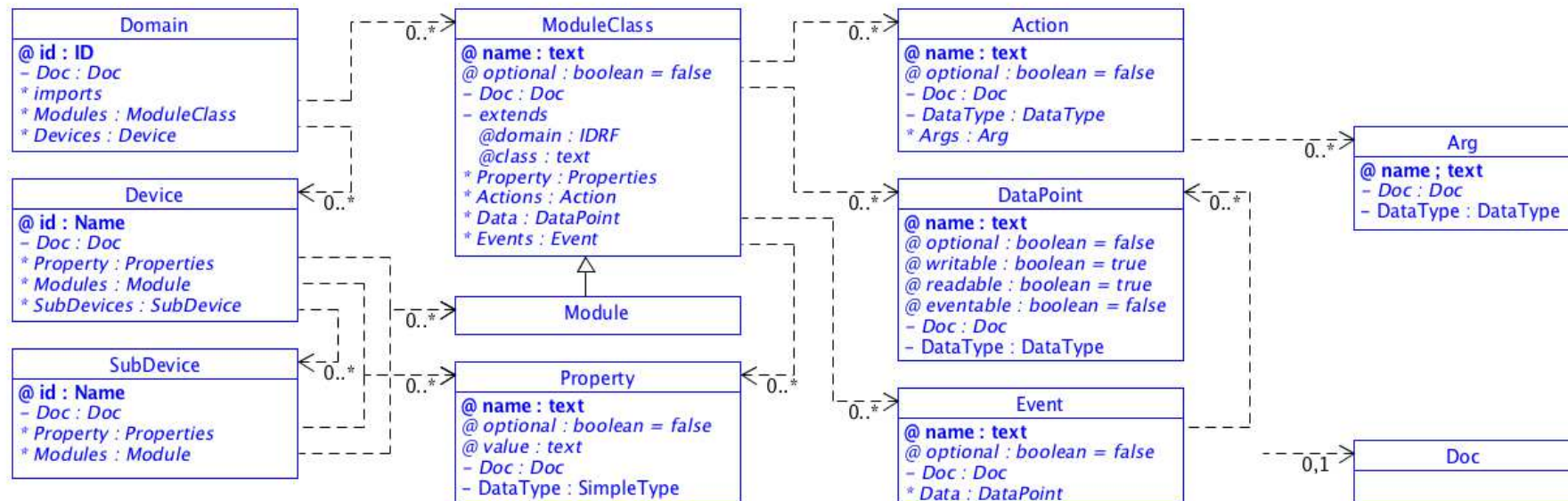


OSGi applications mapped in the oneM2M data model



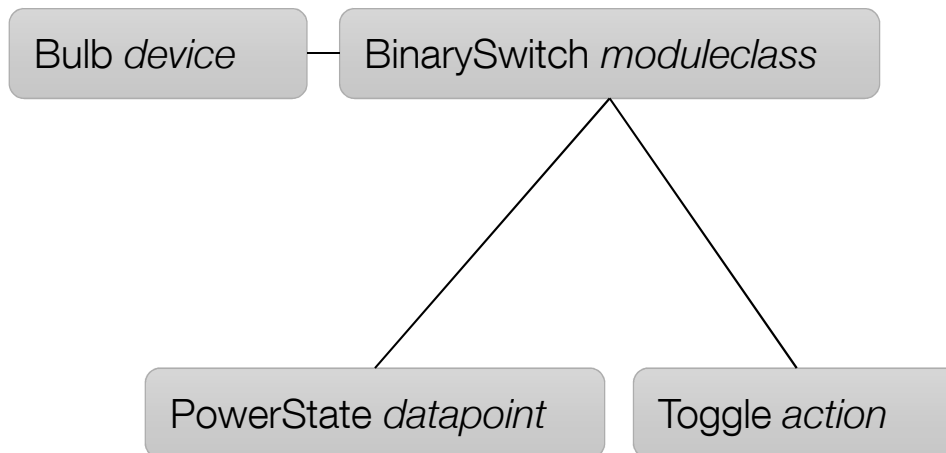
Smart Device Template and specific instantiations

New ways to describe devices in oneM2M data tree model



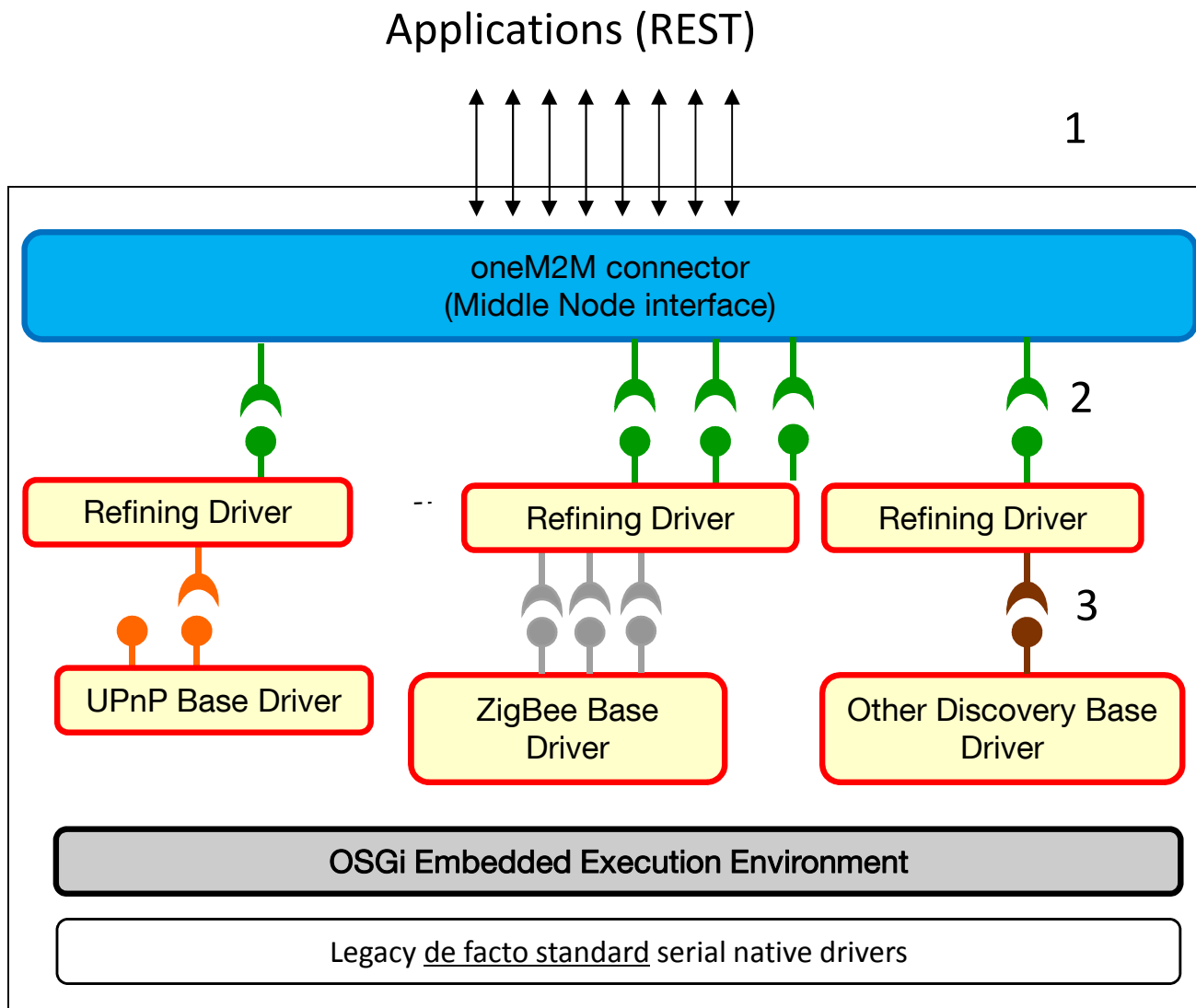
oneM2M Smart Device Template enables the representation of any device in a unified model

- a fully generic abstraction
- and an application-specific abstraction with modules implementing « module classes »



Device direct control from oneM2M cloud applications

oneM2M Interworking Proxy Entity and base drivers



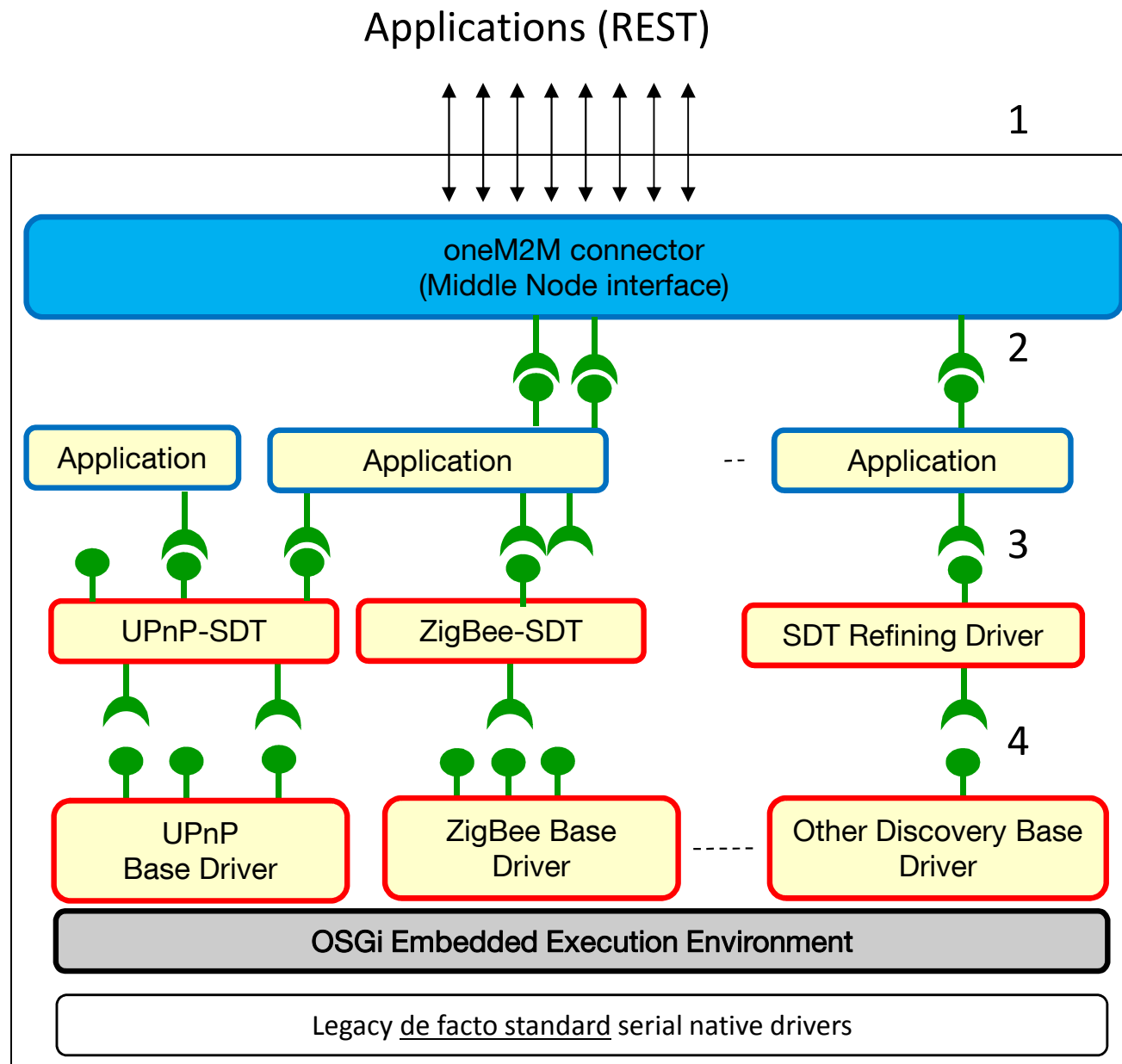
1. (M2M) device communication, e.g.,
M2M.getNetworkIPU("ZigBee").
.getDevice(<DeviceID>)
.getApplication(<EndpointID>)
.getInterface(<clusterID>)
.getMethod(<commandname>)
.putValue("OFF")

2. (M2M) device communication, e.g.,
context.getService(<filter:
M2MApplication, ZigBee, DeviceID,
EndpointID>)
.getInterface(<clusterID>)
.getMethod(<commandname>)
.putValue("OFF")

3. Techno-specific device communication, e.g.,
ZigBeeEndPoint.getCluster(<clusterID>)
.invoke(<commandname>, <value>)

oneM2M cloud apps and embedded apps communication

oneM2M Connector, oneM2M apps and base drivers



1. (M2M) device communication, e.g.,
`mnCSE.getApplication(<AppID>)`
`.getInterface(<ServiceName>)`
`.getMethod(<methodname>)`
`.putValue(<parameters>)`

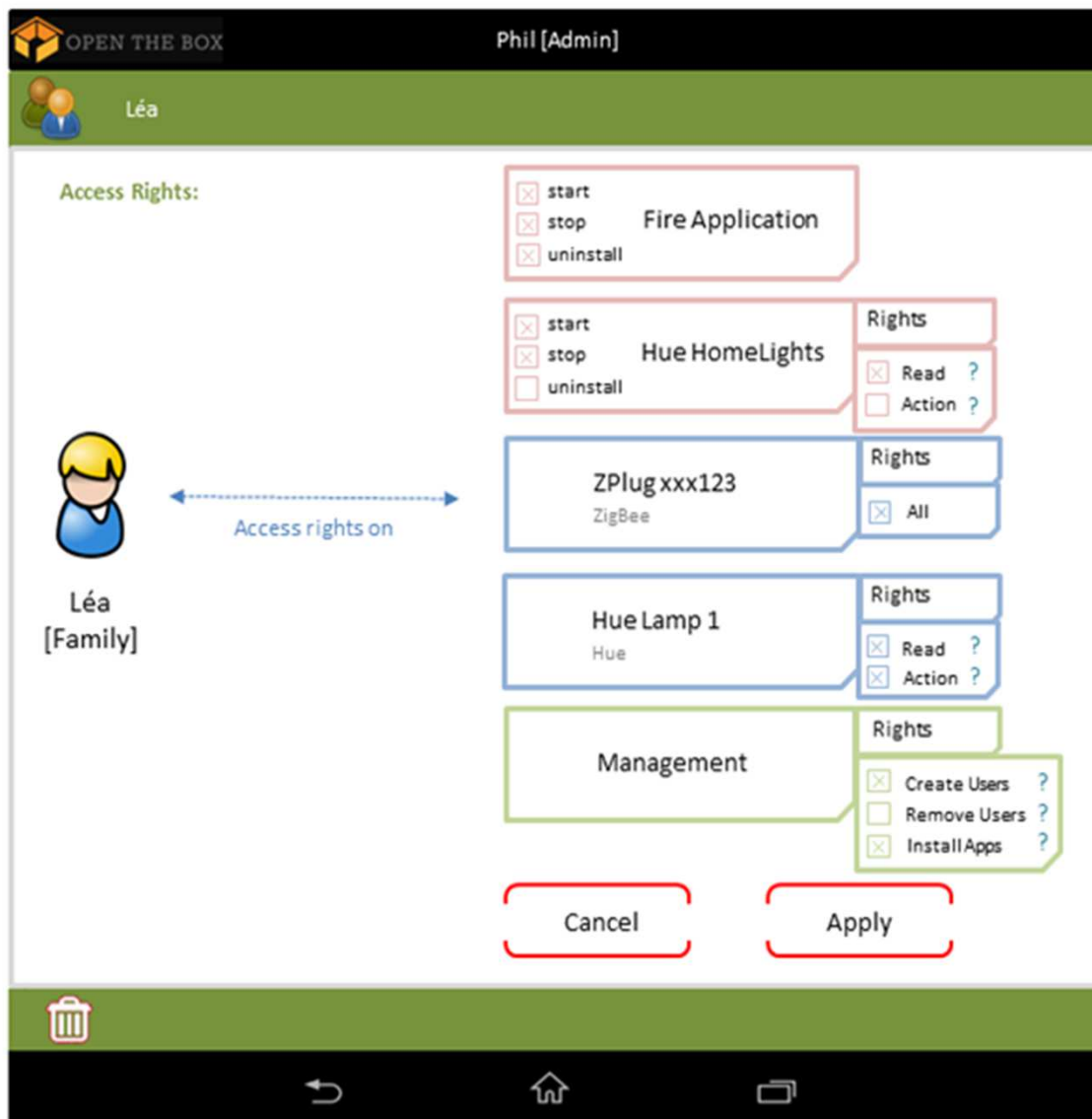
2. (M2M) device communication, e.g.,
`context.getService(<filter: M2MApplication,`
`AppID, ServiceName>)`
`.getInterface(<ServiceName>)`
`.getMethod(<methodName>)`
`.putValue(<parameters>)`

3. Techno-agnostic application-specific communication, e.g.,
`"radiator.switchOn()"`

4. Techno-specific device communication, e.g.,
`zigBeeEndPoint.getCluster(<clusterID>)`
`.invoke(<commandname>, <value>)`

User and application access rights

Right management for users

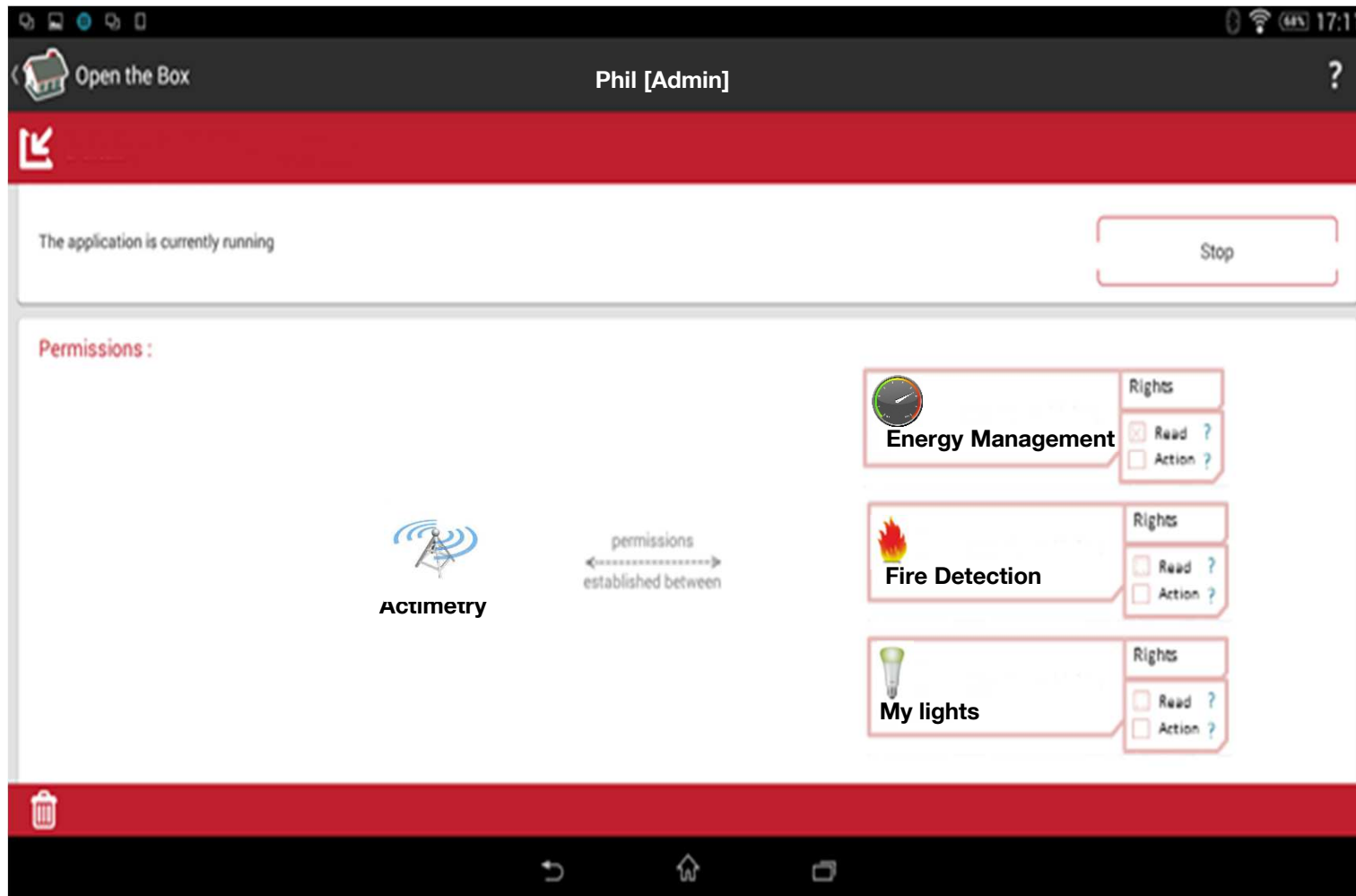


A Smart Home administrator manages user rights for devices, applications and management



The dashboard for Léa will be personalised regarding granted rights

Right management for applications



To perform its actimetry service a partner (e.g. Actimetry) is granted for energy management application API for reading energy consumption data

Implementation and open source contributions

The architecture is today implemented with

- Open the Box OSGi-based framework
 - Application model and access right management
 - ZigBee, Hue OSGi base drivers
- eclipse OM2M 1.0 OSGi-based framework
 - oneM2M implementation
- eclipse SmartHome
 - OSGi EnOcean base driver

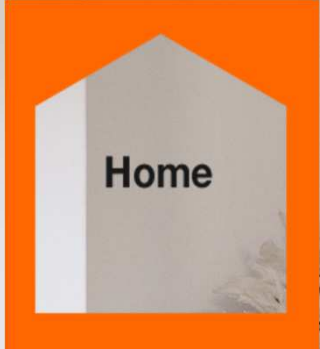


Conclusion and lessons learned

oneM2M standard set is large, very flexible and addresses many aspects.

OSGi technology is complementary and fits the implementation of a secure hybrid architecture between the cloud and embedded gateways.

While first component implementations are published by eclipse community and other communities, efforts remain to build a full Smart Home and IoT SDK.



THANK YOU



A smart home open to any application, any thing,
assisting the users in their everyday life



Opening a smart home infrastructure to 3rd party applications and devices

**IoT in the Smart Home
ETSI Workshop
21-22 March 2016**