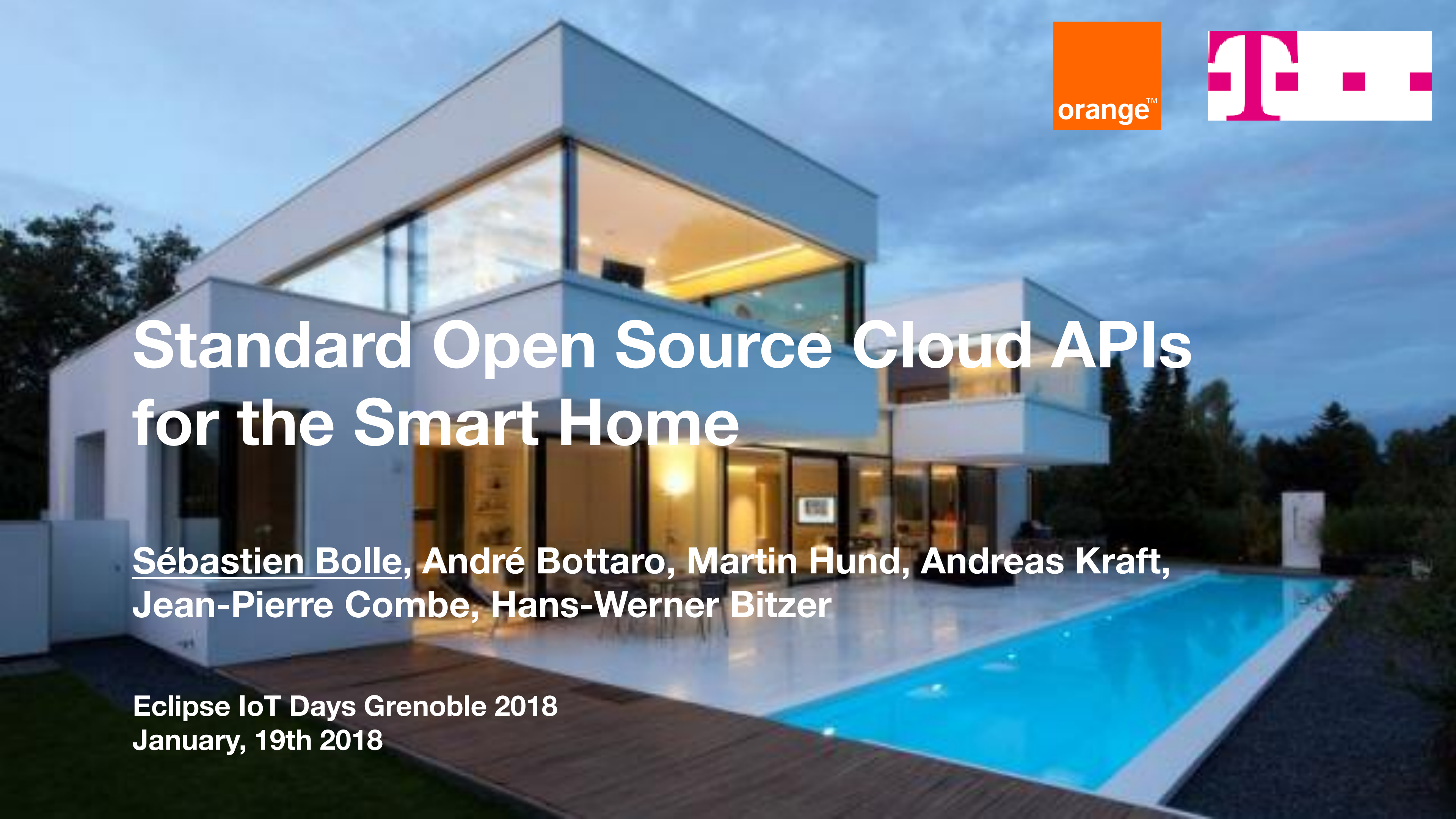
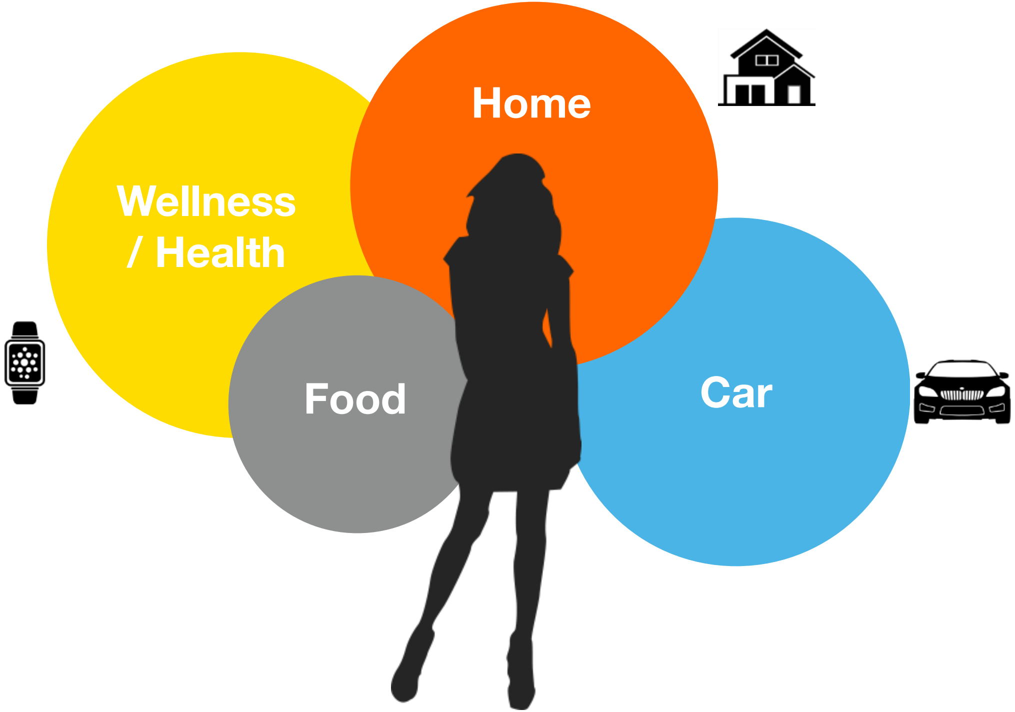


Standard Open Source Cloud APIs for the Smart Home

Sébastien Bolle, André Bottaro, Martin Hund, Andreas Kraft,
Jean-Pierre Combe, Hans-Werner Bitzer

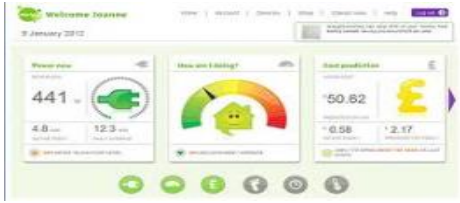
Eclipse IoT Days Grenoble 2018
January, 19th 2018

Smart Home: A new world of services

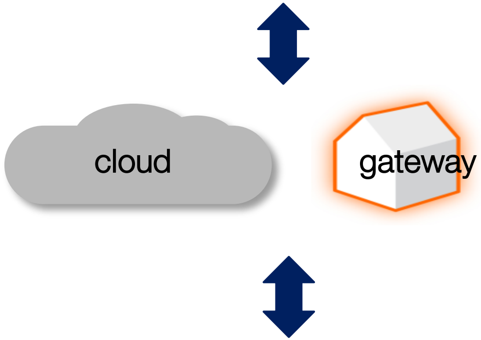


The Smart Home infrastructure,
a typical infrastructure in the Internet of Things

services and applications



Smart Home infrastructure

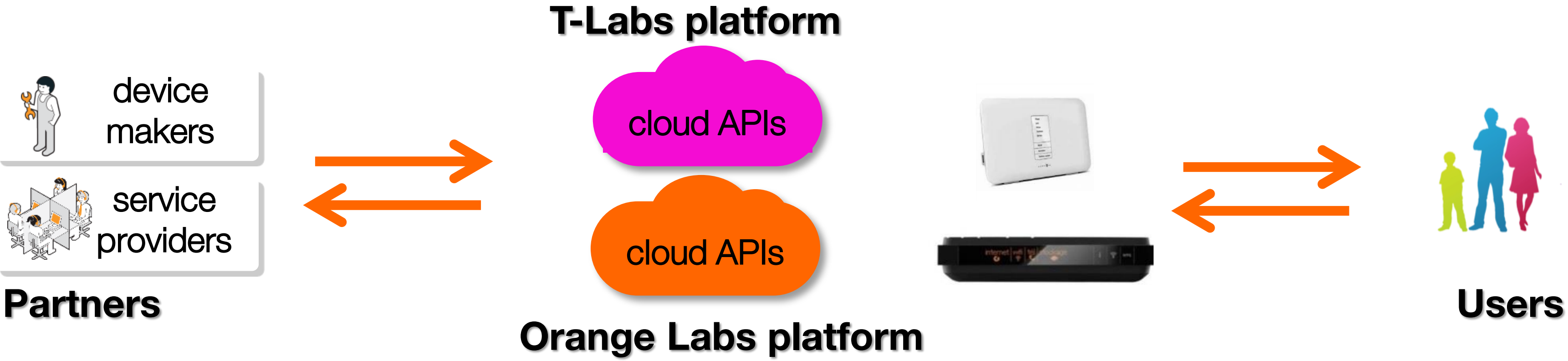


devices

One main technical challenge

Deliver interoperable APIs and data models for infrastructure operators, service and device providers

Together, we push forward open standard cloud APIs



T-Labs platform

device makers

service providers

Partners

cloud APIs

cloud APIs

Orange Labs platform

Users

Orange Labs and T-Labs share open source outcomes with the community

- **Common reference implementation** → **integration by platform operators**
- **Application templates and examples** → **integration with service providers**
- **Repository of cloud connectors** → **integration with device providers**

Why choosing standards in the Internet of Things?

Propose a universal approach

Use an emerging standard backed by a large organization and set of partners.

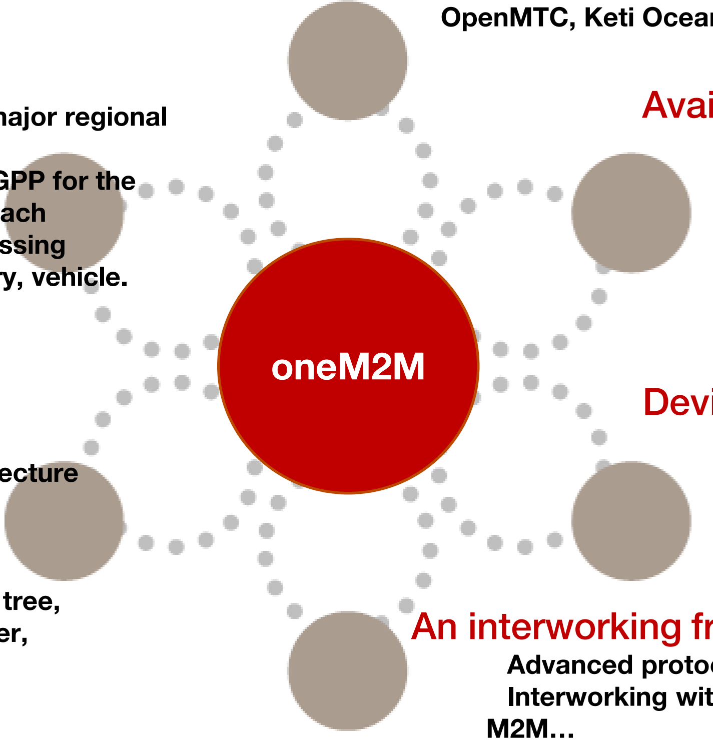
Scale up

Leverage available open source implementations and communities.

Go fast

Capitalize on available specifications covering all technical aspects

oneM2M in a nutshell



An international standard

A partnership project gathering 8 major regional organizations, e,g, ETSI

An analog partnership project to 3GPP for the service layer with the same global reach

A cross-vertical layer for IoT addressing multiple domains: home, city, industry, vehicle.

Available specifications

An end-to-end IoT reference architecture

10 common service functions (communication, data, device management, …)

RESTful access to a resource data tree, with sophisticated features:, e.g., filter, search, subscription, access rights

Available open source platforms

eclipse OM2M, Cisco IoTDM, Fokus OpenMTC, Keti Ocean today

Available commercial platforms

Cisco, Huawei, HP Enterprise, Ericsson, Sierra Wireless, Actility,…

Operators: SK Telecom, LGU+, …

Device abstraction, semantics

Smart Device Template: Abstraction of devices and functions

Smart Home enablement with data models

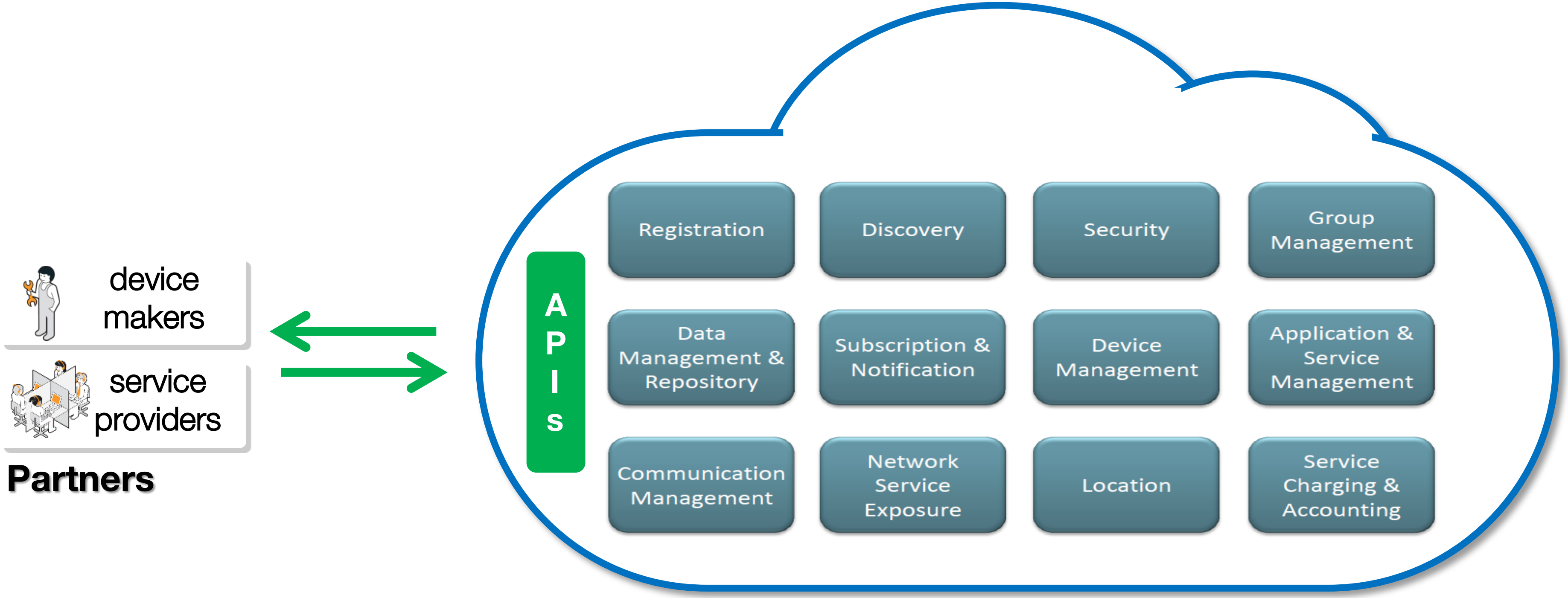
Base ontology, semantic descriptions

An interworking framework with existing technology

Advanced protocol bindings: HTTP, CoAP, MQTT, WebSockets

Interworking with various technologies, Iotivity, AllJoyn, OMA LW M2M…

oneM2M set of Common Service Functions cover all the interfaces to platform, service, device providers



device makers

service providers

Partners

APIs

Registration

Discovery

Security

Group Management

Data Management & Repository

Subscription & Notification

Device Management

Application & Service Management

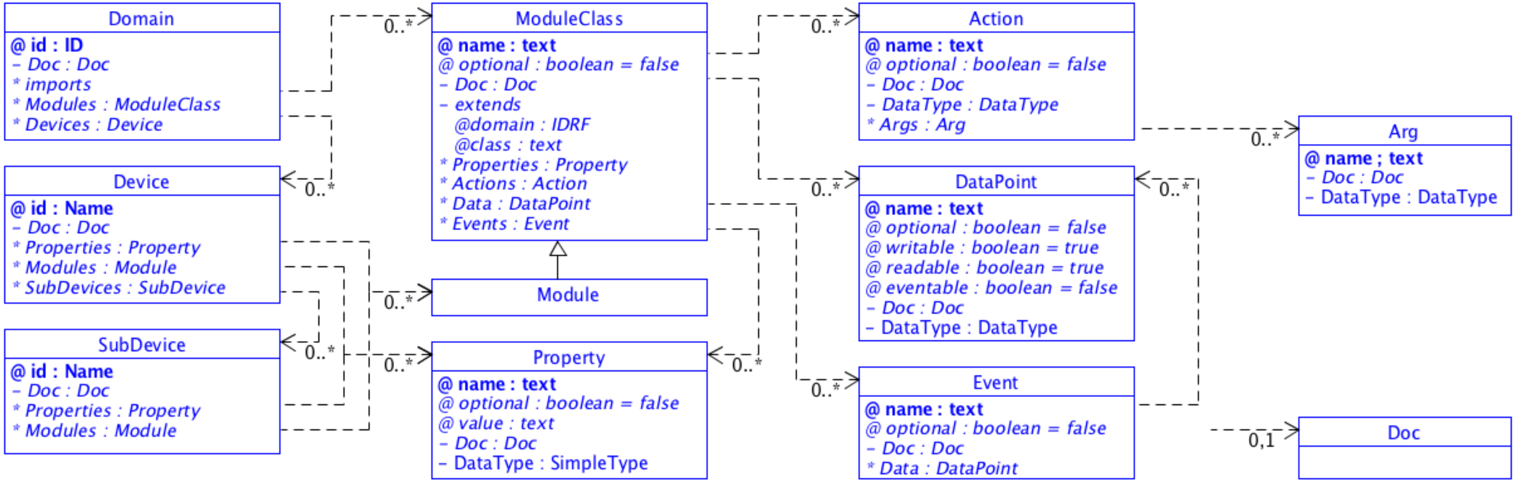
Communication Management

Network Service Exposure

Location

Service Charging & Accounting

oneM2M Smart Device Template to model devices and functions



Domain
@ id : ID
– Doc : Doc
* imports
* Modules : ModuleClass
* Devices : Device

Device
@ id : Name
– Doc : Doc
* Properties : Property
* Modules : Module
* SubDevices : SubDevice

SubDevice
@ id : Name
– Doc : Doc
* Properties : Property
* Modules : Module

ModuleClass
@ name : text
@ optional : boolean = false
– Doc : Doc
– extends
@domain : IDRF
@class : text
* Properties : Property
* Actions : Action
* Data : DataPoint
* Events : Event

Module

Property
@ name : text
@ optional : boolean = false
@ value : text
– Doc : Doc
– DataType : SimpleType

Action
@ name : text
@ optional : boolean = false
– Doc : Doc
– DataType : DataType
* Args : Arg

DataPoint
@ name : text
@ optional : boolean = false
@ writable : boolean = true
@ readable : boolean = true
@ eventable : boolean = false
– Doc : Doc
– DataType : DataType

Event
@ name : text
@ optional : boolean = false
– Doc : Doc
* Data : DataPoint

Arg
@ name ; text
– Doc : Doc
– DataType : DataType

Doc

Description of devices with 3 levels

devices

functions

data, actions, events

oneM2M Home Appliances Information Model and Mapping (TS-0023)

Home Appliances described as SDT devices and modules

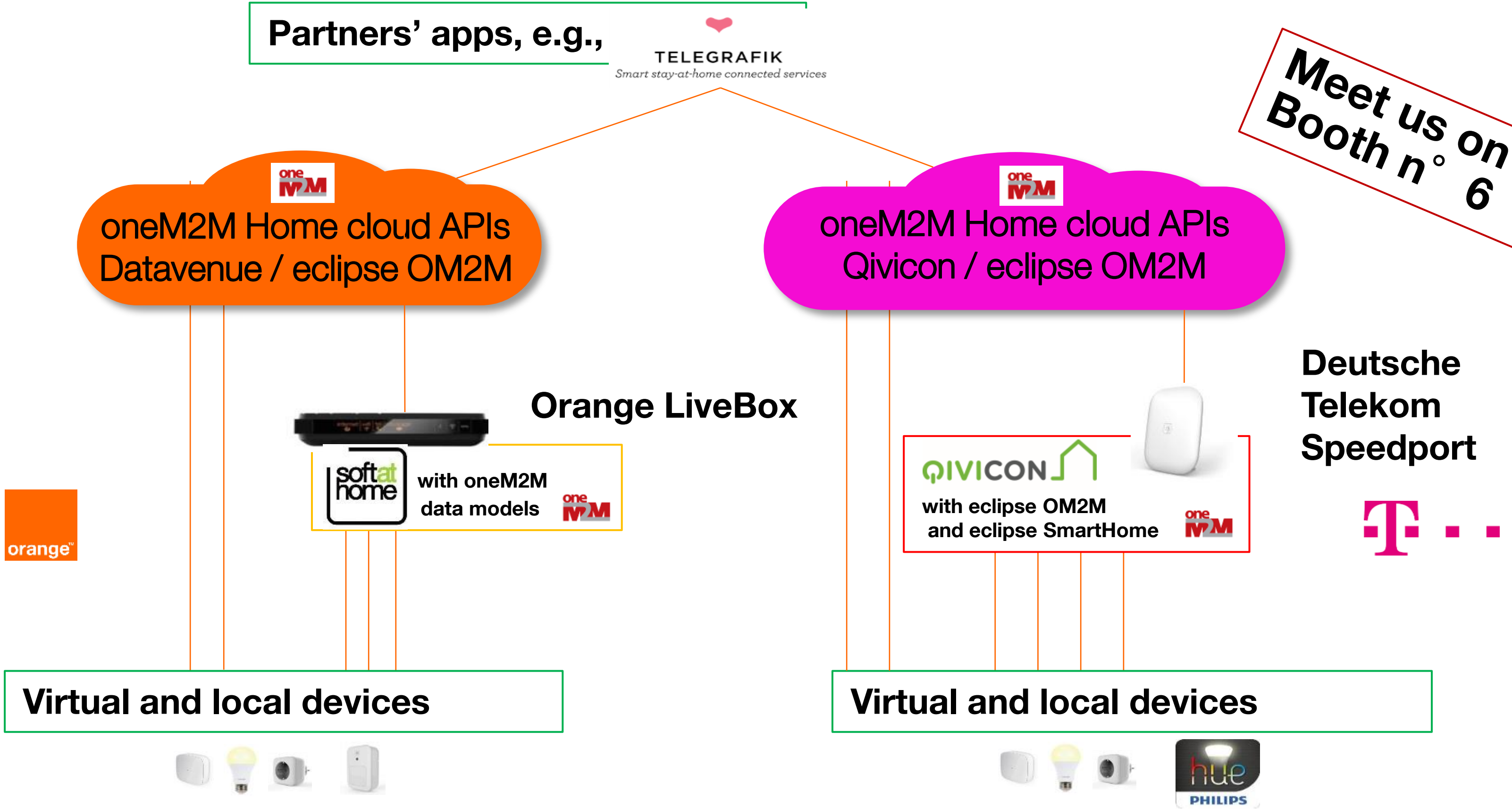
- Light
- Motion Sensor
- Thermostat
- Thermometer
- Humidity sensor
- Smoke Sensor
- Meter
- Battery
- Oven
- Refrigerator
- Television
- Air Conditioner
- Water Heater
- Clothes washer
- Robot Cleaner
- …

External organizations are contributing data models to oneM2M, too

e.g., Open Connectivity Consortium, Echonet

Showcase: Make oneM2M applications run simultaneously with both operator platforms without any code change

eclipsecon Europe 2017



Partners' apps, e.g.,

Meet us on Booth n° 6

Contributions to the community beyond the demo

Open source contributions to Eclipse OM2M project

oneM2M end-to-end implementation available in new OM2M 1.1.0 release.

With ‘SDT Viewer’ tool, applications and Java connectors for various devices.

An online oneM2M Smart Home platform for experiments

Orange Data Share is exposed in a oneM2M version for experimental purposes.

Developers can connect devices and play with a live infrastructure.

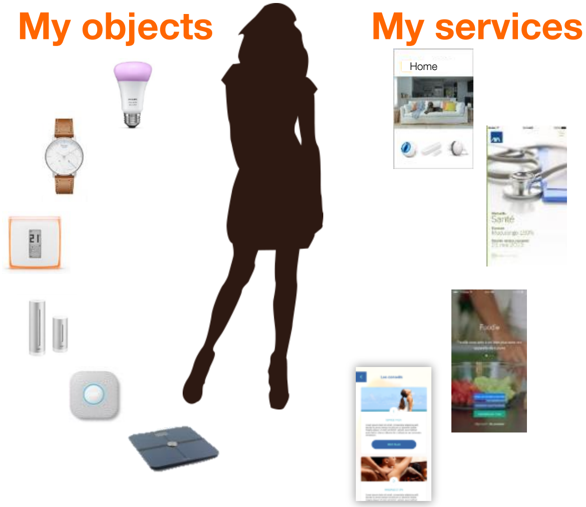
A bridge between Eclipse SmartHome embedded middleware and Eclipse OM2M infrastructure

Eclipse OM2M release 1.1.0

- Features implemented in Eclipse OM2M last release (1.1.0)
  - oneM2M release 2 support
  - FlexContainer resource
  - Smart Device Template (SDT)
  - MQTT communication binding
  - NoSQL MongoDB storage
  - Dynamic Authorization
  - Resource Announcement
  - Enocean interworking
  - Hue interworking
  - Netatmo interworking
  - SmarterCoffee interworking
  - LIFX interworking
  - OSGi DAL (Device Abstraction Layer) interworking
  - Several test suites

Eclipse OM2M 1.1.0 has been released in October 2017 for EclipseCon Europe
Current version is 1.2.0

Online oneM2M server for experiments



- Orange Data Share: datashare.orange.com

  a user dashboard for objects, services & user consents

- Connect your things and play with oneM2M APIs

  e.g., Philips, OSRAM, NetAtmo devices

  Swagger documentation

  https://datashare.orange.com/api-explorer/index.html?url=/om2m/v2/api-docs

  (will move to Orange Partner : https://developer.orange.com/apis/datashare)

Bridging Eclipse OM2M infrastructure with Eclipse SmartHome (ESH) embedded middleware

- Objectives
  - Benefit from ESH bindings with dozens of devices
  - Benefit from OM2M balanced infrastructure between a local box and the cloud.

- Implementation
  - Specify the conversion between oneM2M and ESH device abstraction layers
  - Implement an interworking proxy representing ESH devices into oneM2M resource data tree with oneM2M device data models.

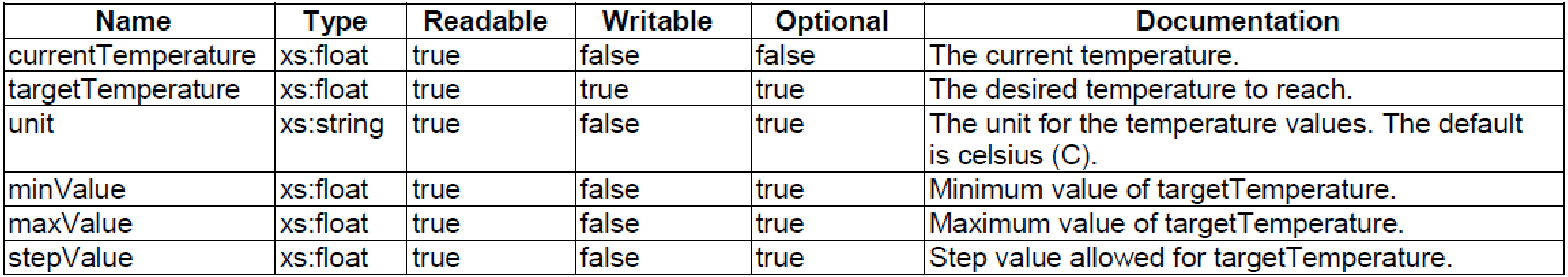
- Availability: soon on Eclipse SmartHome and OM2M

Where do we go from here?

- OM2M and SmartHome:
  - RESTful device connector concept,
  - hands-on sessions with the community,
  - 5-step guide for application developers and device connector developpers
  - oneM2M base ontology implementation, to welcome easily other abstraction layers
  - device management protocols implementation, e.g., BBF TR-069, OMA Lightweight M2M
- oneM2M
  - Addition of semantic descriptors in Smart Device Template
  - Serialization of semantic descriptors in JSON-LD, next to current RDF-XML descriptors
  - Abstraction of device management features

Thanks

Temperature module class example



Name	Type	Readable	Writable	Optional	Documentation
currentTemperature	xs:float	true	false	false	The current temperature.
targetTemperature	xs:float	true	true	true	The desired temperature to reach.
unit	xs:string	true	false	true	The unit for the temperature values. The default is celsius (C).
minValue	xs:float	true	false	true	Minimum value of targetTemperature.
maxValue	xs:float	true	false	true	Maximum value of targetTemperature.
stepValue	xs:float	true	false	true	Step value allowed for targetTemperature.

A temperature sensor may implement the module class with only currentTemperature data attribute.

An Air Conditioner may implement the module class with all optional data attributes.