Cloud APIs for the Smart Home

ETSI IoT week 2017
Essentials 2020

€600 million
Internet of Things & M2M

x6
vs. 2014
Creating the Smart Home ecosystem

services and applications

Smart Home infrastructure

devices
Why choosing oneM2M?

Be attractive to everyone
Use a 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
- Available open source platforms
- Available commercial platforms
- Available specifications
- Device abstraction, semantics
- An interworking framework with existing technology
oneM2M set of Common Service Functions cover all the interfaces to platform, service, device providers.
Smart Device Template (SDT) to model devices and functions
## Temperature module class example

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

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.
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
- ...and many more
Showcase: Make oneM2M applications run simultaneously with both operator platforms without any code change

Partners’ apps, e.g.,

oneM2M Home cloud APIs
Datavenue / eclipse OM2M

Orange LiveBox

Virtual and local devices

Soft@Home
with oneM2M data models

Qivicon / eclipse OM2M

Virtual and local devices

Deutsche Telekom Speedport or home base

oneM2M Home cloud APIs
Qivicon / eclipse OM2M

Datavenue / eclipse OM2M

oneM2M Home cloud APIs
Datavenue / eclipse OM2M

Deutsche Telekom Speedport or home base

Soft@Home
with oneM2M data models

Qivicon
with eclipse OM2M and eclipse SmartHome

oneM2M Home cloud APIs
Qivicon / eclipse OM2M
Together, we push forward open standard cloud APIs

Orange Labs and T-Labs share with the open source community
- Common reference implementation → for the platform
- Application templates and examples → for services
- Bank of open source cloud connectors → for device integration
Contributions to the community beyond the demo

Open source contributions to Eclipse OM2M project

oneM2M implementation available on Eclipse OM2M git repository. 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 (e.g., Philips, OSRAM, NetAtmo devices) and play with a live infrastructure.

APIs are described with Swagger documents.
Thanks