ONEM2M SERVICE LAYER

Presenter: Omar Elloumi (ALU, oneM2M TP chair)
Contributor: Martin Bauer (NEC)
oneM2M www.oneM2M.org
Over 200 member organizations in oneM2M
Ongoing collaborations, reuse of existing standards

• Ongoing collaborations include:
  • oneM2M Interworking specification for AllSeen devices
  • Joint workshop with OIC in July
    • Expected outcomes around interworking
  • Joint workshop with IEEE P2413.

• Reuse of existing standards: IETF, W3C, OMA, BBF, etc.
Purpose & Deliverables

Purpose
To specify and promote an
M2M Common Service Layer

Deliverables
Technical Reports and Technical Specifications
M2M Common Service Layer in a nutshell

• It is a software layer
• It sits between M2M applications and communication HW/SW that provides data transport
• It normally rides on top of IP
• It provides functions that M2M applications across different industry segments commonly need. Those functions are exposed to Applications via IT-friendly APIs.
• It allows for distributed intelligence (device, gateway, cloud apps)
Standardization approach

- Use cases
  - Automotive
  - Home
  - Energy
  - E-Health

- Requirements
  - Security & privacy
  - Device Management
  - Data exchange
  - Interworking

- Architecture APIs and protocols
  - IP communications
  - Restful webservice APIs
  - Reuse of existing protocols
  - Semantics framework (future)

- Test and Interop
  - Reference points
  - Device certification
  - Open source

30-Oct-14 © 2015 oneM2M
oneM2M Architecture approach

Pipe (vertical):
- 1 Application, 1 NW, 1 (or few) type of Device
- Point to point communications

Horizontal (based on common Layer)
- Applications share common service and network infrastructure
- Multipoint communications

Common Service Layer

Communication Network 1

Communication Network 2

Gateway

Local NW

Device

Application
oneM2M Architecture approach

Currently developed solutions are similar are vertically integrated, with limited integration of data models (Zigbee, DLMS for smart meters, etc.).

Horizontal framework, Restful API
Objects represented as resource
Access control policy to access resource

IoT will be based on ontologies (formal description of concepts and relationships, e.g. W3C Semantic Sensor Network) as well as big data frameworks

Automotive Application
Home Application
Energy Application

Common Service Layer
Communication Technologies & Protocols
Communication Networks
Communication Devices & Hardware

Automotive Application
Home Application
Energy Application

TOMORROW
IoT enabled

IoT ready

© 2015 oneM2M
Common Service Functions

- Registration
- Discovery
- Security
- Group Management
- Data Management & Repository
- Subscription & Notification
- Device Management
- Application & Service Management
- Communication Management
- Network Service Exposure
- Location
- Service Charging & Accounting
Why does it matter

| Combat fragmentation | • Healthy eco-system with economies of scale  
|                      | • More partnering choices and opportunities for M2M/IOT industry stakeholders |
| Lower CAPEX          | • Standardized protocols / APIs -> simplifies application development/deployment  
|                      | • Cross-vertical standards -> same devices and back-ends in different industries |
| Lower OPEX           | • Standard features to use networks more efficiently -> get better tariffs  
|                      | • Flexibility for verticals -> utilize best transport network meeting business needs |
| Time to Market       | Reduced development, test and deployment lifecycles through focusing on core business (application logic) |

oneM2M is IoT ready
Technical Specifications

- Requirements
  - TS-0002 (WI-0001)
- Functional Architecture
  - TS-0001 (WI-0002)
- Definitions & Acronyms
  - TS-0011 (WI-0003)
- Service Layer Core Protocols
  - TS-0004 (WI-0009)
- HTTP Protocol Binding
  - TS-0009 (WI-0013)
- CoAP Protocol Binding
  - TS-0008 (WI-0012)
- Management Enabl\textsuperscript{nt} - OMA
  - TS-0005 (WI-0010)
- Management Enabl\textsuperscript{nt} - BBF
  - TS-0006 (WI-0010)
- MQTT Protocol Binding
  - TS-0010 (WI-0014)
- Security Solutions
  - TS-0003 (WI-0007)
- Service Components
  - TS-0007 (WI-0011)

ftp://ftp.onem2m.org/Work Programme/
Open source

• oneM2M does not develop open source but oneM2M members are deeply involved in Linux Foundation and Eclipse Foundation

• Known open source for oneM2M include:
  • Linux Foundation IOTDM (https://wiki.opendaylight.org/view/IoTDM_Overview)

  • Eclipse Foundation OM2M

  • Mobius (S. Korea)
Concluding statements

• oneM2M Release 2 has a semantic interoperability framework
  • Current ongoing development will profit from contributions from IoT community – both to standards and ongoing open source developments
• Domain specific ontologies are needed (oneM2M considered SAREF so far)
  • Need to help development of domain ontologies by domain experts

ftp://ftp.onem2m.org/Work Programme/
Example Scenario – E-Health

- **Patient**
- **Blood Pressure Meter**
- **Scales**
- **Bluetooth Smart Network**
- **Pill dispenser with integrated comm. gateway**
- **Cellular Network**
- **E-Health Web-application**
- **Doctor**
- **Medicalized support**
- **M2M Platform**
- **Tech support Application**
Design principles

- IP-based, but interworks with specific IP and non IP technologies in the M2M Area networks
- RESTful resource oriented APIs, resources are representations of devices, applications, things and related descriptions, etc.
- Distributed intelligence (device, gateway, edge, cloud)
- Reuse of existing device management frameworks
- Reuse of existing data exchange protocols
- Reuse of existing security
- Reuse of underlying network capabilities such as location, triggering, etc.
- Resource access control policies allows many to many communications framework
- Future proof – ready to add semantics support
- No mandated implementation (Database choice, intelligence location, etc.)
**Architecture**

**Application Entity**  Provides application logic for the end-to-end M2M solutions

**Network Services Entity**  Provides services to the CSEs besides the pure data transport

**Node**  Logical equivalent of a physical (or possibly virtualized, especially on the server side) device
Architecture

Reference Point
One or more interfaces - Mca, Mcn, Mcc and Mcc’ (between 2 service providers)

Common Services Entity
Provides the set of "service functions" that are common to the M2M environments

Application Entity
Provides application logic for the end-to-end M2M solutions

Network Services Entity
Provides services to the CSEs besides the pure data transport

Node
Logical equivalent of a physical (or possibly virtualized, especially on the server side) device
Concrete example

Starting assumptions:
- Bootstrapping / DM is done (provisioning of credentials/apps)
- MN-CSE and IN-CSE have logically connected (authentication, binding, encryption)
- Apps have authenticated to xCSE and access right were established
=> Very little effort to synch the different apps
Information Modelling

Resource-based information model

- Information is stored in the system as Resources
- A given Resource can be identified with a Uniform Resource Identifier
- A given Resource is of one of the defined Resource Types
- The Resource Type determines the semantics of the information in the Resource
- Resources can be Created, Read, Updated or Deleted to manipulate the information
- Resources are organized in a tree-like structure and connected by links
Communication Protocols

Reuse IP-based existing protocols

XML or JSON Content serialization

HTTP Example

REQUEST
GET http://provider.net/home/temperature HTTP/1.1
Host: provider.net
From: //provider.net/CSE-1234/WeatherApp42
X-M2M-RI: 56398096
Accept: application/onem2m-resource+json

RESPONSE
HTTP/1.1 200 OK
X-M2M-RI: 56398096
Content-Type: application/onem2m-resource+json
Content-Length: 107
{"typeOfContent":"application/json",
"encoding":1,
"content": "{"timestamp":1413405177000,'value':25.32}"
}
Security Challenges & Solutions

1. Large variety of scenarios
2. Any device in any deployment
3. A device cannot make “judgment calls” on privacy

A. Secure communication
   various authentication options
B. Remote provisioning
   various authentication options
C. Access Control Policy
   express wide variety of rules
Interworking – OMA & BBF

Reuse existing Device Management technologies

oneM2M Domain

DM Domain

Application Entity

IN-CSE

OMA DM 1.3
OMA DM 2.0
OMA LWM2M

OMA

BBF

BBF Server
BBF CPE
BBF Device

DM Server

DM Client

IN-CSE

Mca

OMA DM 2.0
OMA LWM2M

BBF TR-069
Interworking – AllJoyn

<table>
<thead>
<tr>
<th></th>
<th>AllJoyn</th>
<th>oneM2M</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Architecture</td>
<td>Peer-to-Peer in LAN</td>
<td>Server-to-Client in WAN</td>
</tr>
<tr>
<td>API Style</td>
<td>RPC(RMI) API</td>
<td>Resource-based API</td>
</tr>
<tr>
<td>Discovery Style</td>
<td>Proactive Discovery</td>
<td>Passive Discovery</td>
</tr>
</tbody>
</table>
oneM2M in Release 1

Building Management Application

- oneM2M provides resource structure for sensor applications to provide their information
- Syntax and semantics of information not visible to the platform
  - At best limited support for discovery (explicit tags)
  - No support for efficient access to structured information, creation of mash-ups, support for analytics
- Applications using the information have to a-priori know
  - Resources provided by each sensor applications
  - Syntax and semantics of information

Explicit configuration step for every change in available sensors
oneM2M with Semantics

Building Management Application

- oneM2M provides resource structure for sensor applications to provide their information
- oneM2M provides semantic information about resource contents and functionalities making use of it

Functionalities that can be provided or enhanced using semantics
- Queries/Discovery based on semantic descriptions
- Support for analytics (e.g. efficient access to information, deployment of analytics within the platform)
- Support for creation of mash-ups (e.g. enabling IoT scenarios)

Applications using the information can
- Specify what information they are interested in → be notified in case of relevant changes
- Syntax and semantics of information is made explicit, so applications can decide whether they can handle it, what module is needed for processing etc.

Automatic configuration for every change in available sensors
The oneM2M Base Ontology

- oneM2M Base Ontology provides basis for oneM2M functionality, e.g.
  - Interoperability within oneM2M platform
  - Interworking with non-oneM2M devices

- oneM2M Base Ontology not specific to Application Domain.

- Enables mapping to external ontologies – domain-specific, e.g. SAREF (Smart Appliance REFeRence Ontology)
SAREF Example (1) – Resource Representation

- Example of how semantic annotations based on the Smart Appliance REference Ontology (SAREF) can be used to describe an Application Entity (AE) representing a smart appliance.

```
AEwashingMachine1
    ontologyRef
    semanticDescriptor
    startStopContainer
    stateContainer
```

The semantic descriptor for this example is shown on the next slide.
SAREF Example (2) – Semantic Description

```xml
<rdf:RDF
   <rdf:Description rdf:about="http://www.tno.com/saref#WASH_LG_123">
   <saref:hasManufacturer>LG</saref:hasManufacturer>
   <saref:hasDescription>Very cool Washing Machine</saref:hasDescription>
   <saref:hasLocation rdf:resource="http://www.tno.com/saref#Bathroom"/>
   <msm:hasService rdf:resource="http://www.tno.com/saref#WashingService_123"/>
   <msm:hasService rdf:resource="http://www.tno.com/saref#StateService_123"/>
  </rdf:Description>

<rdf:Description rdf:about="http://www.tno.com/saref#WashingService_123">
   <rdf:type rdf:resource="http://www.tno.com/saref#WashingService"/>
   <msm:hasOperation rdf:resource="http://www.tno.com/saref#WashingOperation_123"/>
  </rdf:Description>

<rdf:Description rdf:about="http://www.tno.com/saref#WashingOperation_123">
   <hr:hasMethod>Create</hr:hasMethod>
   <hr:hasURITemplate>/CSE1/WASH_LG_123/startStopContainer </hr:hasURITemplate>
   <msm:hasInput rdf:resource="http://www.tno.com/saref#Action"/>
  </rdf:Description>

<rdf:Description rdf:about="http://www.tno.com/saref#StateService123">
   <rdf:type rdf:resource="http://www.tno.com/saref#StateService"/>
   <msm:hasOperation rdf:resource="http://www.tno.com/saref#StateOperation123"/>
  </rdf:Description>

<rdf:Description rdf:about="http://www.tno.com/saref#StateOperation123">
   <hr:hasMethod>Retrieve</hr:hasMethod>
   <hr:hasURITemplate>/CSE1/WASH_LG_123/state/stateContainer/latest</hr:hasURITemplate>
   <msm:hasOutput rdf:resource="http://www.tno.com/saref#State"/>
  </rdf:Description>
</rdf:RDF>
```
Summary

• oneM2M defines M2M platform with resource-based architecture

• Semantics enables **advanced functionalities** based on **semantic description of functionality and information** within the oneM2M platform

• Semantic support envisioned for
  – Discovery
  – Enabling interworking
  – Supporting mash-ups and analytics
Candidate features for oneM2M next steps

Wide-scale deployment enhancements:
• Home Domain Enablement
• API versioning
• Plug and play scenarios
• Any lessons learnt from prototypes and deployment

Interworking
• AllJoyn
• Enhance 3GPP interworking

Testing and interoperability
• Test specifications
• (external) certification

Security
• E2e security
• Group authentication
• Role based security

New application domains and data models
• Industry and manufacturing
• Home domain

Big data enablement
• Semantics support and use cases
• Ontology, query, reasoning

30-Oct-14
Thank You!

Q&A