IoT Service Layer Standard – Features - Roadmap

Roland Hechwartner, oneM2M TP Chair

Akaino Technical Meetings – Spring
March 1st 2021
Agenda

- The oneM2M Partnership Project
- The Service Layer - An Overview
- Basic Architecture
- Abstraction and Interworking
- Work on Semantics
- The 3GPP Interworking
- some more Release 4 Features
- Conclusion and Outlook
The Partnership Project

- Organisation
- The Standard
- The Service Layer
oneM2M Partnership Project

www.oneM2M.org  All documents and specifications are publically available

More than 200 member organizations in oneM2M

founded¹ July, 24th 2012
TP#1: Sep 24th–29th 2012

[1] Partnership Agreement V 2.0 (Approved March 2013)

Join forces
=> reduce fragmentation

=> Reuse e.g.

e.g. Release 2 transposition
ITU-T SG20 Y.4500.x

© 2021 oneM2M
Organization

http://onem2m.org/about-onem2m/organisation-and-structure
oneM2M Standard – Testing – Certification Program

- Energy
- Residential
- Enterprise
- Other
- Healthcare
- Transportation
- Public Services
- Industry

Requirements

- TS-0002

Technical Reports

Technical Specs

Transposed to Partner Specifications ➔ Regional Standards

Interoperability Test Events

Certification Program

© 2021 oneM2M
oneM2M Breaks Down the Silos

Smart Emergency Services
Smart Transportation
Smart Infrastructure

Vertical Information Flow
Vertical Information Flow
Vertical Information Flow

Horizontal Information Flow

Service Layer

Smart Emergency Services
Smart Transportation
Smart Infrastructure

© 2021 oneM2M
oneM2M is an End-to-End IoT Technology

Flexible Deployment Options

- IoT Cloud / Enterprise
- IoT Gateway
- IoT Edge Device
- IoT User Devices
Cloud provider independent

From Fragmentation to Standards - decoupling device, cloud, and application by open interfaces

Source: oneM2M-TR-0057-V-0.6.0 Getting Started with oneM2M
oneM2M’s modular service functions fit into a coherent framework
oneM2M Feature Summary by Release

Release 1
- Registration
- Discovery
- Security
- Group Management
- Data Mgmt. & Repository
- Subscription & Notification
- Device Management
- Communication Mgmt
- Service Charging
- Network Service Exposure
- App & Service Mgmt
- HTTP/CoAP/MQTT Bindings

Release 2
- Time Series Data
- Flexible Resources that can be customized by app developers (flex container)
- Semantics Description & Discovery
- Security Enhancements
  - Dynamic Authorization
  - Content Security
  - E2E Security
- WebSocket Binding
- Ontology for Home Area Information Model
- oneM2M App-ID Registry
- oneM2M Interworking
  - LWM2M
  - AllJoyn
  - 3GPP Triggering

Release 3
- Semantic Querying/Mashup
- 3GPP SCEF Interworking
  - Non-IP Data Delivery
  - UE reachability Monitoring
  - Device triggering
  - Etc.
- Transaction Management
- Service Layer routing
- Common oneM2M Interworking Framework
- OCF
- OPC-UA
- OSGi
- oneM2M Conformance Tests and Profiles
- Security Enhancements
  - Distributed Authorization
  - Etc.
- Ontology Based Interworking

Release 4 (planned)
- SDT 4.0 and the Information Models for Multiple Domains
- oneM2M Conformance Tests
- Geo Query
- Process Management
- Message Primitive Profiles
- Semantic Reasoning
- Time Management
- Enhanced 3GPP Interworking
  - Session QoS
  - Congestion Monitoring
- Fog/Edge Computing
  - Software Campaigning
  - Resource Synchronization
- Service Subscriber Management
- Security Enhancements
- Group Anycast/Somecast
- Modbus Interworking
- Discovery Based Operations
- Semantic Ontology Mapping

© 2021 oneM2M
Basic Architecture

- TS-0001 Functional Architecture
- **Application Entity (AE)**
  Provides application logic for the end-to-end M2M solutions

- **Common Services Entity (CSE)**
  Provides the set of "service functions" common to the M2M environments

- **Network Services Entity (NSE)**
  Provides connectivity services to the CSEs besides the pure data transport

- **Node**
  Logical equivalent of a physical (or possibly virtualized) device

- **Reference Point – RESTful APIs**
  One or more interfaces -
  - Mca: CSE - AE
  - Mcn: CSE - NSE
  - Mcc: CSE – CSE; Mcc' (between 2 service providers)
A bit more elaborate

Entities
AE (Application Entity), CSE (Common Services Entity) and NSE (Network Services Entity)

Reference Point
One or more interfaces - Mca, Mcn, Mcc and Mcc’

onem2m.org, TS-0001 Functional Architecture
oneM2M is Resource Oriented

- All entities in the oneM2M System, such as AEs, CSEs, data, etc. are represented as resources.
- A resource structure is specified as a representation of such resources.
- Resources are uniquely addressable. The root of the resource structure in a CSE shall be assigned an absolute address.
- Resources have a type
- Resources are formally defined in XSD.

- Square boxes are used for the resources.
- Square boxes with round corners are used for attributes.

onem2m.org, TS-0001 Functional Architecture
RESTful Architecture Style

- oneM2M is based on a RESTful architecture
  - API is based on requests to perform an operation on a resource
  - Operations are Create, Retrieve, Update, Delete

- oneM2M Service Layer supports configurable access control policies that define clear rules dictating, for each resource
  - WHO is authorized to access,
  - WHAT operations are allowed, and under
  - WHICH conditions (e.g. time, location of entity)

Basic Resources
- Common Service Entity (CSE)
- Application Entity (AE)
- Container (CNT)
- Content Instance (CIN)
- ...

Resource access is authorized based upon satisfying at least one Access Control Policy (ACP) rule in one of the linked ACPs

Source: oneM2M Security Webinar
Abstraction and Interworking

- TS-0003 Security Solutions
- TS-0004 Service Layer Core Protocol
- TS-0008 CoAP Protocol Binding
- TS-0009 HTTP Protocol Binding
- TS-0010-MQTT protocol binding
- TS-0020 WebSocket Protocol Binding
- TS-0016 Secure Environment Abstraction
- TS-0026 3GPP Interworking
- TS-0032 MAF and MEF Interface Specification

- TS-0005 Management Enablement (OMA)
- TS-0006 Management Enablement (BBF)
- TS-0014 LWM2M Interworking
- TS-0023 Home Appliances Information Model and Mapping
- TS-0024 OCF Interworking
- TS-0035 OSGi interworking
- TS-0040 Modbus Interworking
- TS-0030 Ontology based Interworking
- TS-0034 Semantics Support
- TS-0033 Interworking Framework
How does oneM2M enable interworking?
Underlying Network Connectivity Abstraction

- oneM2M interworks with underlying network technologies to help manage network connectivity and communication to IoT devices on behalf of the apps
  - Scheduling and buffering of messages based on device reachability
  - Selection of underlying network connectivity options for device communication
  - Triggering of devices to establish a network connection based on when apps need to communicate with devices
  - QoS configuration based on app’s needs

* oneM2M is closely working with 3GPP on interworking via 3GPP defined SCEF API
IoT Device Security Abstraction

- oneM2M hides the different security frameworks of each IoT device technology from the App Developer.
- A Developer’s app can establish a security association with the oneM2M service layer and via this security association, communicate securely with IoT devices.
- The oneM2M service layer establishes and manages the security association with each of the IoT devices on behalf of the app.
  - Enrolment, credential bootstrap/management, authentication, integrity, privacy, and authorization network connectivity of the devices from the app developer.
Transport Protocol Abstraction

- OneM2M hides the different transport protocols used by different devices from the App Developer.
- Applications can use different transport protocols than the one or more different devices they choose to communicate with:
  - E.g. HTTP(s), CoAP(s), MQTT(s), WebSockets
- OneM2M will handle converting the transport protocol so the App Developer does not need to
Content Serialization Abstraction

- oneM2M hides the different content serializations used by the devices from the App Developer.
- Applications can use different types of content serialization formats than the one or more devices they choose to communicate with
  - E.g. XML, JSON, CBOR, Plain-Text
- oneM2M will convert the content serialization format so the App Developer does not have to
IoT Data Model Abstraction

• oneM2M interworks different IoT device data models with one another
  • E.g. OCF, LWM2M, ...
  • All devices are presented to the App via oneM2M API
  • Via standardized oneM2M API, App developers can use device services and manage devices

• Once the data model is abstracted into oneM2M, App Developers can access all devices in a common manner and make use of oneM2M value-add capabilities such as
  • Resource Discovery
  • Generating Events via subscriptions and notifications
  • Grouping
  • Access Controls
oneM2M Abstracted Interworking Information Model

<flexContainer>
  containerDefinition
  ontologyRef
  [customAttribute]
  <subscription>

Specializations

[deviceAirConditioner]
[deviceCamera]
[deviceDoor]
[deviceFan]
[deviceLight]
[deviceOven]
[deviceThermostat]
...
Mapping non-oneM2M Information Models to oneM2M

Specializations
TS-0023

Mappings

OCF (TS-0024)

[deviceAirConditioner]
[deviceCamera]
[deviceDoor]
[deviceFan]
[deviceLight]
[deviceOven]
[deviceThermostat]
IoT Semantics Abstraction

- oneM2M supports a semantic framework and a oneM2M base ontology
- This framework supports interworking different semantic ontologies together
  - Ontologies defined by other organizations can be interworked with the oneM2M base ontology
- Once interworked, the framework enables semantic ontology abstraction
  - Semantic descriptions expressed in terms of other ontologies can be interworked to oneM2M’s Base Ontology to provide abstraction at the semantics level
oneM2M defines an Interworking Proxy Entity (IPE) for interworking different IoT device technologies (e.g. OCF, ...) to the oneM2M service layer

- IPE functions as an adapter that translates non-oneM2M protocols and data models to oneM2M
- E.g. OCF $\leftrightarrow$ oneM2M translation
oneM2M Interworking Architecture

• oneM2M interworking framework can simultaneously interwork different IoT device technologies with one another
  • E.g. OCF, ZWAVE, Bluetooth, ZigBee, ...

• oneM2M provides an abstracted & simplified API for applications to communicate with devices
  • All devices are represented as oneM2M devices regardless of the technology they use
  • Via standardized oneM2M API, App developers can manage devices in a simpler and uniform manner

• Once abstracted into oneM2M, App Developers can sense/control all IoT devices in a common and uniform manner
  • Turn switch on/off, sample sensor reading, etc.
Takeaways

→ Many IoT deployments can have **diverse types of IoT sub-systems and platforms** that require interworking devices, apps and data all to one another

→ **oneM2M** interworking and abstraction capabilities, are able to hide the complexity of interworking from IoT app developers

→ **oneM2M** is able to help **future proof IoT deployments** by enabling different types of brownfield and greenfield technologies to more seamlessly be deployed together

→ **oneM2M** is a standard and **mitigates vendor lock-in**
Work on Semantic

- TS-0012 Base Ontology
- TS-0030 Ontology Based Interworking
Vision: Interconnecting IoT Things

Connecting the un-connected
Share information across domains

Semantic Support
oneM2M base ontology

OneM2M Data sharing
Communication Framework

Vertical ontologies support
SAREF and its extensions
ETSI TC SmartM2M

Source: Enrico Scarrone, TIM
The oneM2M Base Ontology

- oneM2M allows to annotate application specific resources (IoT data) with semantic description.
  - Uses a specialized resource type `<semanticDescriptor>`
  - Can contain proprietary semantics or
  - Semantics according to a published ontology
- The oneM2M base ontology is a top-level ontology that allows to create sub-classes (or equivalence classes) for application-level ontologies
  - Aligned to Smart App Reference Ontology (SAREF)
- Ontologies can be used in oneM2M to describe the application specific data model of an external system for the purpose of interworking.
  - oneM2M Generic Interworking uses such an ontology to enable interworking of oneM2M entities with devices of the external system
Generic interworking

- Non-oneM2M devices are described using the oneM2M base ontology + domain specific extensions.
- The Interworking Proxy Entity (IPE) translates the ontology instance to resources in the CSE based on pre-defined instantiation rules.
SAREF and its extensions

Semantic interoperability

- SAREF Core
- Energy
- Building
- Environment
- Smart Cities
- Industry & Manufacturing
- Smart Agriculture
- Automotive
- E-Health/Aging Well
- Wearables
- Smart Water

01/2017 - Q3/2019 - Q2/2020

© 2021 oneM2M
3GPP Interworking

- TS-0026 3GPP Interworking
- TR-0057 Getting Started with oneM2M
Typical Cellular IoT Deployment

- Use of 3GPP IoT features requires **low-level knowledge** of 3GPP and a business relationship with operator (E.g. Configuration of IoT device sleep times requires intimate knowledge of 3GPP Power Savings Mode (PSM) or extended Idle Mode DRX (eDRX))

→ This presents a high barrier of use and adoption by typical IoT device manufacturers and app developers
- If devices and apps do not properly use these features, **cellular IoT deployments are destined to fail**
- Inefficient use of network resources => **higher costs and less scalability** for operators
- Shortened battery life of devices => inability to deploy cellular IoT devices in many IoT use cases
- Security threats to the network => network, devices and application security will be compromised
oneM2M - 3GPP Interworking using SCEF / NEF

- oneM2M Rel-3 is the first IoT service layer standard to interwork with 3GPP IoT features
- oneM2M provides a complimentary set of value-add services that interwork with 3GPP IoT features
- oneM2M eases the use and adoption of 3GPP IoT features by IoT devices and apps
- oneM2M can be deployed internal or external to an operator’s network
- Enables an operator to move up the value-chain and offer additional value-add IoT services
Some Examples of oneM2M Cellular IoT Value-add Services

- IoT Device Enrollment
- IoT Device Sleep Schedule Management
- IoT Device Location Tracking
- IoT Device Message Delivery Handling
- Network Congestion Control
- Non-IP Data Delivery (NIDD)
- IoT Device Tampering Detection
- Management of Groups of IoT Devices
- IoT Roaming Device Services

Source: Dale Seed: Interworking of oneM2M service layer to underlying 3GPP 4G/5G Networks
IoT Device Service Enrollment

1. oneM2M app requests that device enrolls and registers

2. oneM2M Service Layer triggers device

3. Device receives trigger that contains contact info for oneM2M enrollment function and sends it an enrollment request

4. oneM2M enrollment function enrolls device and bootsraps it with IDs and credentials

5. Device receives enrollment response containing IDs and credentials

6. Device uses IDs and credentials to securely connect and register to oneM2M Service Layer

7. oneM2M Service Layer registers device

8. oneM2M Service Layer notifies app that device is enrolled and registered

Enrollment Function

Request

Notification

3GPP Core Network

Apps

IoT Server

Cellular IoT Devices (NB-IoT, LTE-M)

Trigger

Registration Request

Registration Response

Enrollment Request

Enrollment Response

IoT Features

SCEF
IoT Device Sleep Schedule Management

1. Apps on device specify their desired communication schedule

2. oneM2M app(s) specify their desired schedules to communicate with a device

3. oneM2M Service Layer computes proposed PSM/eDRX times based on aggregated schedule inputs from device and app(s)

4. 3GPP network considers proposed PSM/eDRX timers for device and determines actual timer values

5. Targeted Device is configured with PSM/eDRX timers

6. oneM2M Service Layer notifies app(s) of the device's schedule of availability
IoT Device Message Delivery Handling

1. oneM2M Service Layer subscribes to 3GPP CN to receive notifications when a device enters or exits power savings mode

2. Device enters power savings mode (PSM/eDRX) to conserve its battery

3. 3GPP CN notifies oneM2M Service Layer device has exited power savings mode

4. A oneM2M App issues a request targeting an IoT device

5. oneM2M Service Layer buffers request until device exits power savings mode and becomes reachable

6. Device exits power savings mode (PSM/eDRX)

7. 3GPP CN notifies oneM2M Service Layer device has entered power savings mode

8. oneM2M Service Layer sends buffered requests to device now that it is reachable

9. Device receives request and responds

10. oneM2M Service Layer returns response
IoT Device Location Tracking

1. A oneM2M App asks to be notified if/when device leaves specified geofenced area

2. oneM2M Service Layer subscribes to 3GPP CN to receive location updates for specified device

3. 3GPP CN monitors the current location of the device to detect changes

4. Device moves to a new location

5. 3GPP CN detects device’s current location has changed and sends a notification

6. oneM2M Service Layer receives device’s current location, compares it against geofenced area and detects device has left the area

7. oneM2M Service Layer notifies device has left geofenced area
3GPP Network Congestion Control

1. A oneM2M App asks to be notified if/when device leaves specified geofenced area
2. oneM2M Service Layer subscribes to 3GPP CN to receive location updates for specified device
3. 3GPP CN monitors the current location of the device to detect changes
4. Device moves to a new location
5. 3GPP CN detects device’s current location has changed and sends a notification
6. oneM2M Service Layer receives device’s current location, compares it against geofenced area and detects device has left the area
7. oneM2M Service Layer notifies device has left geofenced area
Non-IP Data Delivery (NIDD)

3. Device sends an uplink oneM2M request over the control plane of 3GPP CN using NIDD (e.g. sensor reading)

2. 3GPP CN processes NIDD configuration and will know where to send NIDD data when the UE establishes a PDN connection

5. (optional) Device receives an downlink oneM2M response via NIDD

4. oneM2M Service Layer processes request and returns a response

1. oneM2M Service Layer issues NIDD configuration request for device that enrolls with NIDD capability

6. A oneM2M App samples sensor reading

7. oneM2M Service Layer returns response (e.g. sensor reading)
IoT Device Tampering Detection

2. A device’s SIM card is tampered with (i.e. removed and swapped out with another SIM)

3. Change in IMSI-IMEI is detected. A notification is generated

4. The tampered device issues a request

5. A oneM2M app sends request targeting device that has been tampered with

6. oneM2M stops servicing requests to/from tampered device

7. oneM2M Service Layer returns error response

1. oneM2M Service Layer subscribes to 3GPP CN to receive device tampering notifications

© 2021 oneM2M
Management of Groups of IoT Devices

6. Devices receive unicast request to join MBMS group

5. oneM2M Service Layer sends requests to devices to join MBMS group

2. oneM2M Service Layer detects if devices are 3GPP MBMS multicast capable

1. A oneM2M app sends request to form a group devices

9. Devices receive group request via MBMS

4. 3GPP CN sets up PDU session for group communication, establishes MBMS group and assigns TMGI

8. oneM2M SL sends group request to SCEF

3. oneM2M Service Layer requests 3GPP CN to establish a MBMS group with an assigned TMGI

7. A oneM2M app sends request to a group of devices

© 2021 oneM2M
oneM2M - 3GPP Interworking Summary

oneM2M supports interworking to underlying 3GPP network services:

- Sending/Receiving request to/from UE over 3GPP control plane (Non-IP Data Delivery NIDD) rather than data plane
- Configuration of UE’s Power Savings Mode or extended Idle Mode DRX (PSM/eDRX) parameters based on App requirements
- Scheduling and buffering of messages based on UE’s reachability and/or App’s background data policies
- Triggering of UE to establish a network connection and/or register or enrol based on App requirements
- Configuration of network QoS parameters based on App requirements
- Querying of UE location and making it available to Apps
- Receiving notifications when the network is congested and scheduling messages to UE accordingly
- Receiving notifications when UE has been tampered with and disabling communication with UE
3GPP Interworking
Architecture and functional mapping for the 3GPP Trust Domain

Optionally present oneM2M entity

Direct connection option not currently supported

Tsp is not focus at this TS

© 2021 oneM2M
Release 4 – Domain Support

- Vertical Domain Support
- Domain Interworking
Release 4 - More Vertical Domain Support

INTERWORKING TOWARDS DOMAIN-SPECIFIC TECHNOLOGIES

- Enhancement on common information models
- Enhancement on common service functions
- Interworking with non-oneM2M systems

REQUIREMENTS

TS-0002

TECHNICAL REPORTS

TECHNICAL SPECS

- Public warning Disaster Alert
- Enterprise
- Smart city
- Railway
- Vehicular
- Industry
Domain models enable data interoperability

- Industry domain models
- Railway domain models

- Use cases and requirements to find gap between the railway domain and oneM2M
- Discovering information model and mapping to adapting oneM2M

<table>
<thead>
<tr>
<th>Module Instance Name</th>
<th>Module Class Name</th>
<th>Multiplicity</th>
</tr>
</thead>
<tbody>
<tr>
<td>binarySwitch</td>
<td>binarySwitch</td>
<td>1</td>
</tr>
<tr>
<td>clock</td>
<td>clock</td>
<td>1</td>
</tr>
<tr>
<td>touchscreen</td>
<td>touchscreen</td>
<td>0..1</td>
</tr>
<tr>
<td>emergencyButton</td>
<td>pushButton</td>
<td>0..1</td>
</tr>
<tr>
<td>audioVolume</td>
<td>audioVolume</td>
<td>0..1</td>
</tr>
<tr>
<td>ticketReader</td>
<td>oppCardReader</td>
<td>1</td>
</tr>
<tr>
<td>crossingSensor</td>
<td>traceSensor</td>
<td>1</td>
</tr>
<tr>
<td>connectivity</td>
<td>connectivity</td>
<td>1</td>
</tr>
<tr>
<td>gateState</td>
<td>runState</td>
<td>1</td>
</tr>
<tr>
<td>directionPanel</td>
<td>directionPanel</td>
<td>0..1</td>
</tr>
<tr>
<td>crossingIndicatorColour</td>
<td>colour</td>
<td>1</td>
</tr>
<tr>
<td>crossingIndicatorColourSaturation</td>
<td>colourSaturation</td>
<td>1</td>
</tr>
<tr>
<td>crossingIndicatorColourBrightness</td>
<td>brightness</td>
<td>1</td>
</tr>
<tr>
<td>crossingBarrier</td>
<td>crossingBarrier</td>
<td>1</td>
</tr>
</tbody>
</table>

Example: device model of `deviceSmartGate`
Emerging technologies for Smart cities

• Smart Lift
  • Focus on predictive maintenance (operating technician/building owners/users and administrators)
  • Services for emergency situation support, remote operation, city services, etc.
  • Develop smart lift data models in the semantic supported by oneM2M

• Ontology on Smart cities
  • Standardize ontologies for Smart City services through integrating SmartM2M SAREF work to oneM2M as the baseline
  • Develop a framework to maintain developed ontologies and get input from various domain actors using a oneM2M SmartCity portal service
Public warning services enablement

IPAWS in US

MoWaS in Germany

CAP in Italy

CBS in Korea

J-ALERT in Japan

oneM2M architecture enables Public Warning Service

Source: TR-0046 Study on Public Warning Service Enabler
SDT 4.0: make it easy for IoT developers

- Provides an abstraction layer for connected devices
- Together with other organizations, such as OCF and OMA, oneM2M Rel 3 defines 84 ModuleClasses and 50 Devices with various functionalities
- In Rel 4 (ongoing) SDT is restructured to fit more the verticals (Home, City, Industry, Health, Automotive,...)

URL to GitLab: https://git.onem2m.org/MAS/SDT (master)
## SDT based Information Models

### ModuleClass

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>R/W</th>
<th>Optional</th>
<th>Unit</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>powerGenerationData</td>
<td>xs:float</td>
<td>R</td>
<td>true</td>
<td>W</td>
<td>Amount of instantaneous generation data.</td>
</tr>
<tr>
<td>roundingGenerat...</td>
<td>xs:integer</td>
<td>RW</td>
<td>false</td>
<td>s</td>
<td>The measurement interval of water consumption.</td>
</tr>
<tr>
<td>significantMultiply...</td>
<td>xs:float</td>
<td>RW</td>
<td>false</td>
<td>s</td>
<td>The measurement interval of reverse water consumption.</td>
</tr>
<tr>
<td>generat...</td>
<td>xs:integer</td>
<td>RW</td>
<td>true</td>
<td>s</td>
<td>The measurement interval of water temperature.</td>
</tr>
<tr>
<td>flowInterval</td>
<td>xs:integer</td>
<td>RW</td>
<td>false</td>
<td>s</td>
<td>The measurement interval of water consumption.</td>
</tr>
<tr>
<td>reverseFlowInterval</td>
<td>xs:integer</td>
<td>RW</td>
<td>true</td>
<td>s</td>
<td>The measurement interval of reverse water consumption.</td>
</tr>
<tr>
<td>waterTemperatureInterval</td>
<td>xs:integer</td>
<td>RW</td>
<td>true</td>
<td>s</td>
<td>The measurement interval of water temperature.</td>
</tr>
<tr>
<td>waterPressureInterval</td>
<td>xs:integer</td>
<td>RW</td>
<td>true</td>
<td>s</td>
<td>The measurement interval of reverse water pressure.</td>
</tr>
<tr>
<td>intensiveSampleInterval</td>
<td>xs:integer</td>
<td>RW</td>
<td>true</td>
<td>s</td>
<td>The time interval of intensive data sampling.</td>
</tr>
<tr>
<td>intensiveReportInterval</td>
<td>xs:integer</td>
<td>RW</td>
<td>true</td>
<td>s</td>
<td>The time interval of intensive data report.</td>
</tr>
<tr>
<td>intensiveReportStartTime</td>
<td>m2m:timeStamp</td>
<td>RW</td>
<td>true</td>
<td></td>
<td>The start time of data intensive report.</td>
</tr>
</tbody>
</table>

### DeviceModel

<table>
<thead>
<tr>
<th>Module Instance Name</th>
<th>Module Class Name</th>
<th>Multiplicity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenLevel</td>
<td>openLevel</td>
<td>0..1</td>
<td>See clause 5.3.1.1</td>
</tr>
<tr>
<td>doorlock</td>
<td>binarySwitch</td>
<td>1</td>
<td>See clause 5.3.1.12.</td>
</tr>
<tr>
<td>doorStatus</td>
<td>runState</td>
<td>0..1</td>
<td>See clause 0.</td>
</tr>
<tr>
<td></td>
<td>dishWasherJobMode</td>
<td>0.1</td>
<td>See clause 5.3.1.29.</td>
</tr>
</tbody>
</table>

© 2021 oneM2M
Some more example R4 features

- Geo-query
- Software Campaign
- Semantic reasoning
- Modbus oneM2M inter-operability
- Discovery based operations

Source: Dale Seed, Convida Wireless, oneM2M Release 4 Webinar
Quick Overview of oneM2M Entities

AE: Application Entity
CSE: Common Services Entity

CSE resource tree for storing metadata/data of applications and devices
Geo Query

Location: New York [40.71, -74.0]
Location: LA [34.05, -118.24]
Location: Chicago [41.85, -87.65]

AE sends a geo query request to discover temperature sensor in New York City

Geo Query Request
geoQuery = Point, [40.71, -74.0], ...

CSE looks up locations of AEs and finds AE-2 matches Geo Query criteria

Response (AE-2)
Software Campaigning

1) Software Campaign Request
[targets, trigger criteria, operation, version, URL, ...]

2) When software campaign trigger criteria are met, CSE-1 performs operations to manage CSE SW on targeted Fog/Edge nodes

3a) Install CSE software
3b) Install CSE software
3c) Install CSE software

4) Response
[Aggregated software campaign results]

AE-1 offloads software management to CSE-1 using software campaigning to orchestrate CSEs deployed on Fog/Edge nodes

AE-1

CSEBase

AE-1

softwareCampaign
version = 1.2.5
name = MN-CSE
targets = Node-2, Node-3, Node-4
softwareOperation = INSTALL
URL = http://images/MN-CSE
softwareTriggerCriteria = subjectResource123

Node-2
CSE-2

Node-3
CSE-3

Node-4
CSE-4

AE: Application Entity
CSE: Common Services Entity

© 2021 oneM2M
Semantic Reasoning

AE sends a semantic based resource discovery request

```
SELECT ?sensor WHERE {
  ?sensor "is-a" "Temperature Sensor"
  ?sensor "isOwnedBy" "Organization-A"
}
```

CSE applies semantic reasoning rules to perform semantic discovery and determine that “AE-2” “is-a” “Temperature Sensor” that “isOwnedBy” “Organization-A”

AE: Application Entity
CSE: Common Services Entity
Modbus Interworking

AE: Application Entity
CSE: Common Services Entity
IPE: Interworking Proxy Entity

AE-1 can retrieve the value of a temperature sensor without having to be aware it is a Modbus device.

Retrieve deviceThermometer1 Request
Response (temperature = 72)

Modbus devices represented as oneM2M devices

© 2021 oneM2M
Discovery-based Operations

AE-2 can send a single request to both discover AEs with labels set to “temperature” and then to update the “appName” attribute to “tempSensor”.

AE: Application Entity
CSE: Common Services Entity
Conclusion and Outlook

- oneM2M Events Timeline
- oneM2M Release 5 – future features examples
- Takeaways
Collaborations Examples

Enablement of Mobile Edge Computing for Internet-of-Things using oneM2M and ETSI MEC ISG

- ETSI ISG MEC
  - oneM2M platform instance can be placed where MEC is running
  - two features allows this to happen:
    - Software Campaign
    - Enhanced resource announcement
  - Joint Whitepaper development (expected Q1 2021)

- GSMA OPG
  - Information exchanged
  - Discussion ongoing
oneM2M Implementation Ecosystem

A vibrant and healthy oneM2M ecosystem continues to build

Industry-driven Open source implementations

Examples of Commercial implementations, Prototypes, Trials

oneM2M.org lists
66 Deployments
List of deployments

Certification Test Houses and Test Tool Vendors

Regular Interop Events (7 Held from 2015-2020)

Based on slides by: Dale Seed, oneM2M Overview 2019
oneM2M Adoption is Global

oneM2M adoption is expanding

- oneM2M Open Source Project
- oneM2M Product Offerings
- oneM2M Trial Deployment
- oneM2M Commercial Deployment
- Use of oneM2M recommended (Smart Cities)
oneM2M Future Feature development

• oneM2M Release 5
  • Use Case and Requirements development
    • Work ongoing in Requirements and Domain Models Working Group
  • Architecture and protocol related work – to be started Q1 2021

Release 5 Work in Progress

- oneM2M System Enhancements to Support Data Protection Regulations [WI-0095]
- Effective IoT Communication to Protect 3GPP Networks [WI-0096]
- oneM2M and SensorThings API [WI-0100]
- Advanced Semantic Discovery [WI-0101]
- System enhancements to support Data License Management [WI-0102]
- ...

© 2021 oneM2M
Takeaways

oneM2M

• is a global open standard, not controlled by a single private company
• specifies a common set of horizontal IoT services
  • architecture, common services functions,
• enables data interoperability
  • Information model, semantics, ontology based interoperability
• interworks with existing IoT technologies
• has interoperability testing and a certification program
• standardized APIs simplify the life for IoT stakeholders
  • minimize development, deployment & maintenance costs
• is a mature and a commercially deployed technology

oneM2M release 4
Expected: Q2 2021

Work in progress on
oneM2M Release 5
Backup
Publicly Accessible Links

**Web Site**
http://www.onem2M.org

**Developer Guides**
http://www.onem2M.org/developer-guides

**Technical Questions**
http://www.onem2M.org/technical/technical-questions

**Published Specifications**
http://www.onem2M.org/technical/published-documents

**Webinars**
http://www.onem2M.org/technical/webinars

**YouTube Channel**
https://www.youtube.com/c/onem2morg

**Events**
http://www.onem2M.org/news-events/events

**Certified Products**
http://www.onem2Mcert.com/sub/sub04_01.php

---

**Smart Device Template**
SDT 3.0 is available under Apache 2 License:
https://git.onem2M.org/MAS/SDT

**TS-0023 : SDT based Information Model and Mapping for Vertical Industries**
The latest published version of TS-0023 is available:
http://www.onem2M.org/technical/published-drafts

**Tools**
A utility for converting SDT to other formats is the SDTTool:
https://github.com/Homegateway/SDTTool

**Twitter**
@oneM2M

**Stackoverflow**
https://stackoverflow.com/questions/tagged/onem2m

© 2021 oneM2M