ONEM2M AND IOT

Presented by [Aurindam Bhattacharya]
CURRENT LANDSCAPE OF IOT DEPLOYMENT
Current Landscape of IoT Deployment

Pipe (vertical):
1 Application, 1 NW,
1 (or few) type of Device

Business Application

Business platform

Transport Network (mobile, fixed, Powerline ..)

Gateway

Local N/W

Device

Application Entity
Current Landscape of Smart City Applications

Management Layer

Automotive Application
Home Application
Energy Application
Health Application

Vertical Centric Siloed Ecosystem

In order to share data among all these siloed (proprietary) applications, another layer would have to be created which can extract data from these applications.
Data Sharing Example

- Cars fitted with various sensors send information to the manufacturer
- The service provider servicing the car may also need access to some of the sensor data
- The insurance company providing insurance for the car also needs information as to how the car is driven and based on this info charges the premium. The fraudulent insurance claims would also be minimised.
- The on-road assistance company would require the location information of the car to send appropriate assistance
- The traffic police would like to know accident information to be able to manage traffic.
- This information would be useful for the commuters to select alternate routes
Issues with Non-standardised IOT Deployment

A lot of traffic (upload) gets generated in proprietary manner from the sensors to the respective Servers running the business process. Here the TSPs/ISPs networks are used as mere transport.

Vendor Locked-in Solutions. No interoperability.... The application registration, authentication, security etc are taken care of by the respective Business Functions themselves.
Issues with Non-standardised IOT Deployment

Now imagine the data being shared between multiple business processes

And there are billions of these devices...!!
IOT Deployment with a Standardised Common Service Layer

WITH A COMMON SERVICE LAYER, SHARING OF DATA WOULD NOT REQUIRE DUPLICITY OF SENSORS OR EVEN PAYLOAD ACROSS THE NETWORKING INFRASTRUCTURE

THE ACCESS CONTROL POLICIES IN THE COMMON SERVICE LAYER WOULD GOVERN THE SHARING OF DATA IN A SECURE AND CONSISTENT MANNER.
About oneM2M

• A global partnership among Standards Defining Organizations (SDOs) and Industry Associations like:
  ARIB (Association of Radio Industries and Businesses, Japan),
  ATIS (Advancing Transformation of the ICT Industry, America),
  CCSA (China Communications Standards Association, China),
  ETSI (European Telecommunications Standards Institute, Europe),
  TIA (Telecommunication Industries Association, America),
  TSDSI (Telecommunications Standards Development Society, India),
  TTA (Telecommunications Technology Association, Korea), and
  TTC (Telecommunications Technology Committee, Japan).
Additional partners contributing to the oneM2M work include:
  the BBF (Broadband Forum), Continua, GlobalPlatform, HGI (Home Gateway Initiative), the
  New Generation M2M Consortium - Japan, and OMA (Open Mobile Alliance).
  [C-DOT is also partner Type I (through TSDSI) contributing to the standards]

• In simple terms the main goal to develop technical specifications for an M2M Service Layer
  • A software platform to make M2M devices/applications communicate with each other in a
    secure and efficient manner
OneM2M Architecture Approach

Pipe (vertical):
- 1 Application, 1 NW, 1 (or few) type of Device

Horizontal (based on common Layer)
- Applications share common infrastructure, environments, network elements & data

- Business Application #1
- Business Application #2
- Business Application #n

Common Service Layer

ETSI Symposium on
Achieving Interoperability – Best Practices in Standardization
What is oneM2M?

- It is a software/middleware layer
- It sits between applications and underlying communication networking HW/SW
- It typically rides on top of IP protocol stack
- It provides functions that applications across different industry segments commonly need
- It exposes common set of functions to applications via developer friendly APIs
- It is integrated into devices/gateways/servers and allows distributed intelligence
- It hides complexity of NW usage from apps
- It controls when communication happens
- It stores and shares data
- It supports access control
- It notifies applications about events
What is oneM2M?

Horizontal layer of functions commonly needed across different industry verticals
Motivation for oneM2M

**Vertical Silos**

- Intra-Vertical or Intra-Use Case
  - Lower cost
    - CAPEX & OPEX impact as explained, easier to maintain, focus only on Apps
  - Larger eco system: More choices
    - Transition from one SP to another SP using same standard
    - Chose from different SW/HW vendors that offer standard compliant products
    - Select products that match with individual requirements, minimize cost for customization

- Inter-Vertical or Inter-Use Case
  - All of the motivators above, plus:
    - Sharing Data Across Verticals brings about New Business Opportunities
    - Shared access to sensors/actors/events/data allows for new business models / added value
Layered model for high-level architecture of oneM2M

- **CSE (Common Service Entity):**
  - Offer common functionalities exposed through the Reference Points.
  - CSE contains Common Service Function Modules (CSF).
  - CSE and CSF are represented by Restful resource management and message flows.
OneM2M: High-Level Architectural Model

AE: Application Entity
CSE: Common Services Entity
NSE: Network Services Entity
Mcc: Reference point CSE-CSE
Mca: Reference point CSE-AE
# oneM2M High Level Architecture

<table>
<thead>
<tr>
<th>Reference Point</th>
<th>One or more interfaces - Mca, Mcn, Mcc and Mcc’ (between 2 service providers)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common Services Entity</td>
<td>Provides the set of &quot;service functions&quot; that are common to the M2M environments</td>
</tr>
<tr>
<td>Application Entity</td>
<td>Provides application logic for the end-to-end M2M solutions</td>
</tr>
<tr>
<td>Network Services Entity</td>
<td>Provides services to the CSEs besides the pure data transport</td>
</tr>
<tr>
<td>Node</td>
<td>Logical equivalent of a physical (or possibly virtualized, especially on the server side) device</td>
</tr>
</tbody>
</table>

---

**ETSI Symposium on**

**Achieving Interoperability – Best Practices in Standardization**
Description of Nodes in OneM2M Architecture

**Application Dedicated Node (ADN):**
An Application Dedicated Node is a Node that contains at least one Application Entity and does not contain a Common Services Entity.

**Application Services Node (ASN):**
An Application Service Node is a Node that contains one Common Services Entity and contains at least one Application Entity.

**Middle Node (MN):**
A Middle Node is a Node that contains one Common Services Entity and contains zero or more Application Entities.

**Infrastructure Node (IN):**
An Infrastructure Node is a Node that contains one Common Services Entity and contains zero or more Application Entities.
CONFIGURATIONS SUPPORTED BY oneM2M ARCHITECTURE
The Common Service Functions in Horizontal Service Layer Architecture

- Registration
- Discovery
- Security
- Group Management
- Data Management & Repository
- Subscription & Notification
- Device Management
- Application & Service Management
- Communication Management and Delivery Handling
- Network Service Exposure
- Location
- Service Charging & Accounting
Motivation for the IoT App Developer

• At present majority of the IoT deployments are targeted towards a single use case within a single vertical
• With time there would be a pressing need to connect diverse types of IoT devices based on diverse set of technologies e.g. Networking technologies, Security Frameworks, Service/Management, Content Type Serialization, Semantic Ontologies etc.
• The abstraction capabilities of oneM2M hides the complexities involved in interacting with a diverse set of IoT devices from the App Developer
• oneM2M abstraction capabilities include:
  ❖ IoT Device Services and Management Abstraction
  ❖ Underlying Network Abstraction
  ❖ IoT Semantic Ontology Abstraction
  ❖ Content Type Serialization Abstraction
  ❖ IoT Device Security Abstraction
  ❖ Transport Protocol Abstraction
### oneM2M Abstraction Capabilities

| IoT Device Semantic Ontologies | W3C TD | SAREF | ... |
| IoT Services and Management Technologies | OCF | OMA DM | BBF | LWM2M | ... |
| IoT Device Content Serializations | XML | JSON | CBOR | ... |
| IoT Device Transport Protocols | HTTP(s) | CoAP(s) | MQTT(s) | ... |
| IoT Device Security Models | TLS/PSK | TLS/PKI | DTLS/PSK | ... |
| IoT Device Underlying Networks | Cellular | Fixed | Wi-Fi | ... |
TESTING FRAMEWORK FOR
The oneM2M testing framework is based on concepts defined in ISO/IEC 9646 [i.2], TTCN-3 [i.4], ETSI EG 202 237 [i.3]
Conformance testing

The Conformance testing can show that a product correctly implements a particular standardized protocol, that is, it establishes whether or not the implementation under test meets the requirements specified for the protocol itself.

It will test protocol message contents and format as well as the permitted sequences of messages.
Conformance Testing Methodology

In summary, the oneM2M Conformance Testing methodology consists of:

• Selection of Implementations Under Test (IUT).
• Identification of reference points.
• Development of test specifications, which includes:
  ➢ Development of "Implementation Conformance Statements" (ICS), if not already provided as part of the base standard.
  ➢ Development of "Test Suite Structure and Test Purposes" (TSS&TP).
  ➢ Development of "Abstract Test Suite and Implementation eXtra Information for Test" (ATS&IXIT) including:
    • Definition of the Abstract Protocol Tester (APT).
    • Definition of TTCN-3 test architecture.
    • Development of TTCN-3 test suite, e.g. naming conventions, code documentation, test case structure.
    • Verification of ATS (TTCN-3)
    • IXIT proforma.
Interoperability testing

- Interoperability testing can demonstrate that a product will work with other like products: it proves that end-to-end functionality between (at least) two devices is as required by the standard(s) on which those devices are based. In that context, the system under test is made of the combination of different devices under test coming from different suppliers.

- The important factors which characterize interoperability testing are:
  - Interoperability tests are performed at interfaces that offer only normal control and observation (i.e. not at specialized interfaces introduced solely for testing purposes);
  - Interoperability tests are based on functionality as experienced by a user (i.e. they are not specified at the protocol level). In this context a user may be human or a software application;
  - The tests are performed and observed at functional interfaces such as Man-Machine Interfaces (MMIs), protocol service interfaces and Application Programming Interfaces (APIs).
oneM2M Certification

- **oneM2M Certification** is intended to create an ecosystem of certified products that ensures interoperability among certified products.

- **oneM2M Certification** is a LOGO program, NOT a compulsory program. It will represent to market that oneM2M products and services meet oneM2M Specification and Test requirements which ensure interoperability.

Certification Process

- **App-ID registry**
- **Apply**
- **Test**
- **Certify**
THANK YOU

Aurindam Bhattacharya
Centre for Development of Telematics, New Delhi

aurindam@cdot.in
RESERVE SLIDES

Certification Process
1. Select Certification Body (CB) and Authorized Testing Lab (ATL) of your preference from the list (List of Facilities)

2. Submit the application form (See Annex A), Implementation Conformance Statement (ICS) and Information eXtra Implementation Testing (IXIT) to request for certification of your product to the selected CB and ATL

3. Once the applicant has submitted the appropriate documents, the CB will begin the review process. CB may contact the applicant if any changes or additional information is required.

4. The applicant shall agree to oneM2M standard certification policies as detailed in the oneM2M Standard Certification Program Management Document.
After completion of ‘Apply’ process,
1. CB will generate appropriate Test Plan depending on the type of certification request provided in the ICS and IXIT document.
2. Generated Test Plan will be sent to the selected ATL.
3. ATL will indicate the number of test samples required and other accessory or documents as necessary.
4. ATL may request product information if changes are deemed necessary in the ICS and IXIT document.
5. ATL will perform applicable test cases according to the test plan provided by the CB and will create a test report as an input to the CB and the applicant.
Certify

After completion of Testing process,

1. The ATL submits the test result report to the CB.
2. The CB reviews and assesses the test result report and related documents.
3. When all certification requirements are met, CB issues the oneM2M certification for the product.
4. CB will maintain a list of certified products on the cert.onem2m.org which will also be referenced by www.oneM2M.org. The list will include information such as Product Name, Product Vendor, Product Type, etc.