M2M world of connected services
• The current marketplace is extremely fragmented, which has increased the R&D cost in each specific domain.

• Current M2M silo model is not an efficient way to communicate, it is a barrier to further development.

• Many vertical M2M solutions have been designed independently and separately for different applications, which impedes large-scale M2M deployment.
143 organizations around the world are involved in M2M standardization according to the Global Standards Collaboration M2M Task Force.
Standards for M2M service capabilities:
Target: end-to-end enablement across servers, gateways, devices with standardized service interfaces.

M2M network

Source: ETSI M2M
Smart M2M standard (ETSI M2M)

- Provide a horizontal M2M service platform with a generic set of service capabilities to enable M2M interoperability in terms of communication and data.

- Define a Service Capability Layer (SCL) on top of connectivity layers:
  - SCL deployed in network server, gateways and devices.
  - SCL enables discovery, registration, authentication, data-transfer using containers, publish/subscribe, groups, access rights, security, etc.

- Interface with existing technologies:
  - Multiple communication protocol binding.
  - Re-use existing remote device management.
  - Interwork with vendor-specific and legacy devices.

- Facilitate innovation across industries by exposing data and providing services.

Source: ETSI M2M
OM2M is an open source implementation of the Smart M2M standard, and is a member of Eclipse IoT Working Group.

OM2M web site
http://om2m.org

OM2M Interested parties

Eclipse IoT collaboration
OM2M architecture

End user devices

Data Analytic

SCADA Interface

Network domain

Device and Gateway domain

M2M Server

OM2M gateway

HTTP/CoAP

ZIGBEE

OM2M Connecting things

OM2M Connecting things

Java OSGi Alliance

Java OSGi Alliance

HTTP/CoAP

HTTP/CoAP

M2M Server

LAAS-CNRS

MQTT

phidgets

ZigBee

LoWPAN
OM2M functional architecture

- OM2M provides a Service Capability Layer (SCL) including a set of common services for M2M interoperability.
- A SCL can be deployed on the Network domain, or on the Device and Gateway domain.

Diagram:
- Device and Gateway domain
  - Device Applications (DA)
  - Gateway Applications (GA)
- Network domain
  - Network Applications (NA)
  - Service Capability Layer (SCL)
  - Application Enablement (AE)
  - Communication Selection (CS)
  - Generic Communication (GC)
  - Routing Function (RF)
  - Interworking Proxy (IP)
  - Reachability, Addressing Repository (RAR)
  - Security (SEC)
  - Remote Entity Management (REM)
OM2M implements a RESTful API.
All M2M communications are performed based on simple primitive procedures.

- Each SCL data model is structured using a standardized resource tree.
- The resources can be simply triggered using basic CRUD requests.
OM2M resource tree example

Gateway

Applications

Smart_Meter_APP

Containers

DESCRIPTOR

ContentInstances

DATA

ContentInstances

MEASUREMENT_3

MEASUREMENT_2

MEASUREMENT_1

HTTP/CoAP or Interworking Proxy

Smart Meter

HTTP/CoAP

M2M server

End user
Binding to multiple communication protocols

**HTTP Request**
- **URI**: http://192.168.0.3:8080
- **Method**: GET, POST, PUT, DELETE
- **Headers**: Authorization, etc.
- **Body**: XML/JSON

**Request**
- **TargetID**: resourceURI
- **Method**: RETRIEVE/CREATE/UPDATE/DELETE/EXECUTE
- **RequestingEntity**: username/password
- **Representation**: Object

**Response**
- **Status**: OK, CREATED/NOT_FOUND, Etc.
- **URI**: location
- **Representation**: Object

**Mapping**
- **Protocol independent Request**
- **Protocol independent Response**
OM2M building blocks

- OM2M is a java platform running on top of an OSGi Equinox runtime which make it highly extensible via plugins.

- Each SCL includes required plugins and is build as an Eclipse product using maven and Tycho.
M2M smart Metering Example 1
A representation of the NSCL and DSCL along with their corresponding resource structure.
M2M smart Metering Example 1

Step 1 - Network Application Registers to the NSCL
M2M smart Metering Example 1

Step 3 - The Smart Meter Registers to the NSCL
Step 3 - Device Application Registers to the DSCL
M2M smart Metering Example 1

Step 4 - Device Application Creates An ElectData container
M2M smart Metering Example 1

Step 5 - Device Application Creates a Data contentInstance
M2M smart Metering Example 1

Step 5 - Network Application Read Data content Instance
Step 5'- Network Application Create Subscription on ElectData contentInstances
Step 6’ - Network Application receives notification
M2M smart Metering Example 2
A representation of the NSCL and DSCL along with their corresponding resource structure.
M2M smart Metering Example 2

Step 1 - Network Application Registration to the NSCL
Step 2 - NA Subscribes for Registering Smart Meters
M2M smart Metering Example 2

Step 3 - The Smart Meter Registers to the NSCL
M2M smart Metering Example 2

Step 4- Notifying the Network Application about a Registered Smart Meter

POST
M2M smart Metering Example 2

Step 5 - Device Application Registration to the DSCL
Step 6 – Announcing a Registered DA to the NSCL
Step 7 - Reporting Meter Data through the Use of Container Resource
Demonstration
Thank you for your attention