OM2M: Standardized service platform for M2M interoperability

www.om2m.org
Agenda

• Introduction
• ETSI M2M standardization
• OM2M service Platform
• Subscribe/Notify scenarios
• Conclusion
The Web evolution

Web 1.0: Mostly Static.
One Way. The HTML Pages were viewed by the users with little interaction.

Web 2.0: Interactive.
Two way Communication. The data flows between users and web site. Most of transactions are initiated by the humans.
Example: Social Networking (Facebook, Google+), web services.

Web 3.0: Not only humans but Things/Objects/Machines will also interact with each other. They will initiate the transactions and influence each other.
Internet of Things, Machine-to-Machine (M2M), Communications.

• Smart buildings
• Smart Home
• Smart grid
• Smart factories
• etc.

Machine-to-Machine

http://www.webofthings.org/2011/02/04/lift11-talk-transcript/

6/18/2014
ETSI M2M Standardization

Proprietar M2M applications («verticals»)

ETSI M2M Vision («horizontal»)

M2M Business Application 1
M2M Business Application 2
M2M Business Application n

Existing ICT Infrastructure

Dedicated Devices

M2M Gateway

M2M Device

Application Infrastructure

Core Network Infrastructure

Access Network Infrastructure

Wireless / Fixed Line
ETSI M2M Standardization

- ETSI M2M standard provides a Service Capability Layer (SCL) including a set of common services for M2M interoperability.
OM2M features

- OM2M implements a RESTful API (using an URI and CRUD Methods)
- All M2M communications are performed based on simple primitive procedures
OM2M Demo 1

« Yes! I can turn it ON 😊 »

LAAS ADREAM experimental building

WAN

M2M Server
(NSCL)

LAN

LAAS

BeagleBone Black
M2M Gateway
(GSCL)

Phidgets HUB

Lamp Relay

Fan Relay

Temperature Sensor

Luminosity Sensor
OM2M component diagram

- The CORE plugin routes received request to the correct controller.
- It checks access rights, persist data, notifies interested subscribers, do request redirect or resource announcement if needed.
OM2M building blocks and plugins

- OM2M runs on top of an OSGi Equinox runtime.
- Each SCL includes required plugins and is build as an Eclipse product using Maven and Tycho.
Interworking Proxy Unit

Phidgets Request

Phidgets Response

Technology independent Request

Technology independent Response

Service Capability Layer (SCL)
Interworking Proxy Unit

Phidgets Request

Phidgets Response

Zigbee Request

Zigbee Response

Technology independent Request

Technology independent Response

Your techno

Service Capability Layer (SCL)

Your techno IPU

Phidgets IPU

Zigbee IPU

HTTP App

Your techno IPU!
HTTP Request

URI: http://ipAddress:port/context/resourceTargetId
Method: GET, POST, PUT, DELETE

HTTP Response

StatusCode: 200, 201, 404

Protocol Independent request

targetId: resourceTargetId
Method: RETRIEVE, CREATE, UPDATE, DELETE, EXECUTE

CoAP Request

URI: coap://ipAddress:port/context/resourceTargetId
Method: GET, POST, PUT, DELETE

CoAP Response

StatusCode: 2.00, 2.01, 4.04

CoAP App

Communication Protocol mapping

HTTP App

CoAP App

Service Capability Layer (SCL)

6/18/2014
ETSI Communication techniques

- request/response
- long polling (NA server enabled)
- **subscribe/notify** (NA server enabled)

Subscribe/Notify scenario

6/18/2014
OM2M Demo2
« Yes! I can remotely monitor it »

Group Management
✓ Switch ON/OFF a group of lamps

NA
(server enabled)

Subscribe/Notify

6/18/2014
Thanks!
Questions?

Project Contributors
Thierry Monteil
Yassine Banouar
Mahdi Ben Alaya
Christophe Chassot
Khalil Drira