Standard-based IoT Service Platform

Mahdi Ben Alaya
Samir Medjiah
Thierry Monteil
Khalil Drira
OM2M?

› **Horizontal** service platform for M2M **interoperability**
› **Restful** API with a **generic** set of service **capabilities**
› **OSGi-based** architecture **extensible** via plugins
› Allow developing services **independently** of the underlying network
› Facilitate **deployment** of **vertical** applications
› **Compliant** to **SmartM2M** Standard and will be the basis of **OneM2M** Standard

› Main features:
  Machine registration, application deployment, container management, resource discovery, access right authorization, subscription / notification, group management, and resource announcement.
Dev / Code

- Simple and fast plugin development and build using Maven and Tycho.

OM2M main building blocks
Architecture

End user devices

Data Analytic

SCADA Interface

M2M Server

Network domain

Device and Gateway domain
Website stats

1. France: 1676 (23.84%)
2. Japan: 824 (11.64%)
3. Taiwan: 567 (8.01%)
4. India: 540 (7.63%)
5. South Korea: 531 (7.50%)
6. United States: 424 (5.99%)
7. Germany: 328 (4.63%)
8. Portugal: 216 (3.05%)
9. Tunisia: 169 (2.39%)
10. Italy: 163 (2.30%)
Roadmap

› Release 1.0.0 (Planned for 31 mars 2015)
  – Support the SmartM2M standard.
  – Protocol-independent CORE module to handle generic REST request.
  – HTTP communication binding based on Jetty and Apache HTTP
  – CoAP communication binding based on the Californium.
  – Simple interworking driver connecting a set of simulated devices
  – Web interface for browsing and debugging the resource structure.
  – DAO persistence layer based on EclipseLink JPA
  – Embedded Apache H2 database by default.
  – Optimized resource structure and database access

› Release 2.0.0 (Planned for September 2015)
  – Support the OneM2M standard
Key challenges

› MQTT communications
  – Seamless integration of client and broker

› ZigBee IP
  – Configuration problems related to some devices

› 6lowpan IP
  – Very short data frames (~50 bytes)
  – New data format (JSON, Core Link)

› A Dashboard!
  – OpenSCADA? Birt? Home made?

› Evolution towards the OneM2M standard
  – Updating the data structures and interface
Collaboration opportunities

› Already using:
  – Equinox for OSGi
  – Tycho for plugin build
  – Californium for CoAP
  – Jetty for HTTP
  – EclipseLink for database

› Planning to use
  – Mosquito/Moquette for MQTT Broker
  – Paho for MQTT Client
  – Birt for data visualization
  – OpenSCADA for dashboard
  – Wakaama for LWM2M server
  – Leshan for LWM2M client
  – Concierge for small-footprint OSGi
Thank you

eclipse.org/om2m
om2m-dev@eclipse.org