MICROEJ®

LwM2M over MQTT

Eclipse IOT Days – March 9, 2017 – Grenoble

frederic.riviere@microej.com

DISCLAIMER

Quick Overview

- CoAP - Constrained Application Protocol
  - REST based (GET/POST/PUT/DELETE), Device is viewed as a server
  - Eclipse Reference Implementation: Californium
- MQTT – Message Queue Telemetry Transport
  - Publish/Subscribe paradigm with quality of services
  - Centralized Server (broker)
  - Eclipse Reference implementation: Paho
- LwM2M - Lightweight Machine to Machine
  - Provisioning & Device Management
  - Standard for Core Objects (Device, Firmware Update)
  - Eclipse IOT Reference implementation: Leshan

Why This Specification ?

- LwM2M is becoming the standard protocol for device management
- LwM2M specification is tightly coupled with CoAP/UDP/DTLS:

> *[...] The LWM2M Enabler uses the Constrained Application Protocol (CoAP) with UDP and SMS bindings. Datagram Transport Layer Security (DTLS) provides security for UDP transport layer. [...]*

- Still an industrial issue to access a device behind NAT (home topology)
  - DTLS (re-)negotiation latency
  - Keep NAT alive (1 minute ?)

Why This Specification ?

- When MQTT is already a project requirement
- MQTT widely used for domain data transfer
  - Device <-> Cloud
  - Optional Peripherals (smartphones, …)
- A large offering of scalable MQTT Servers for production
- Regular TCP/TLS connection

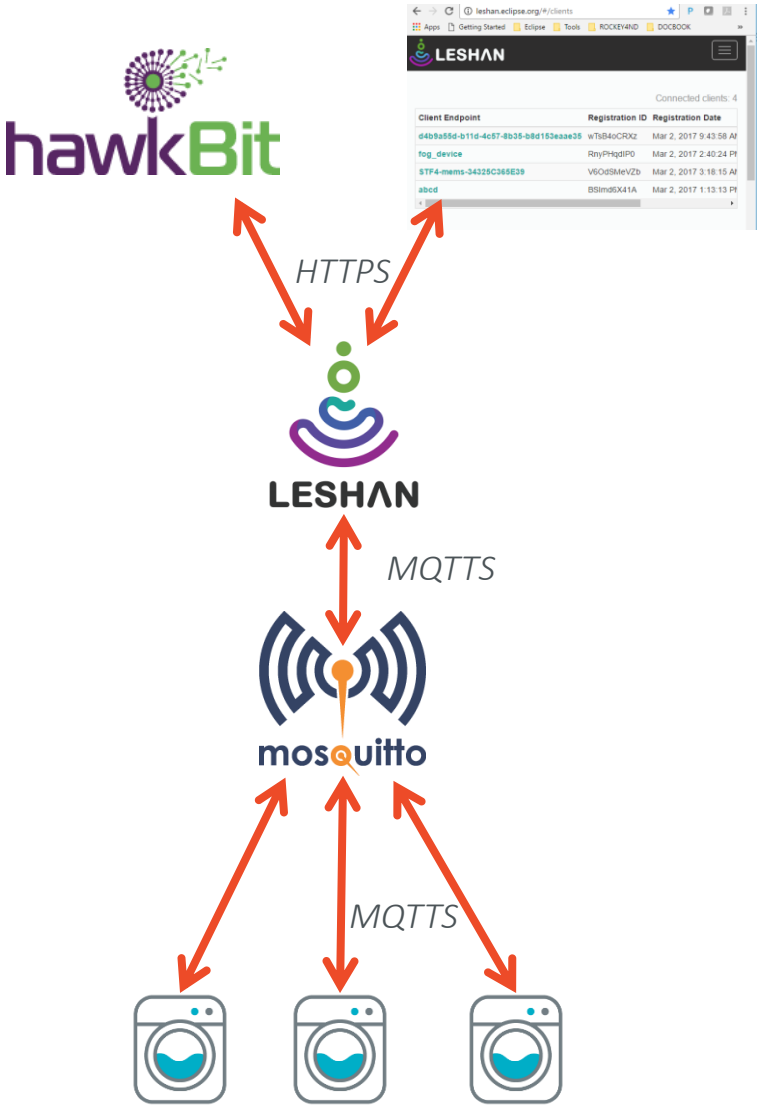
Topology (Eclipse Projects)

Fleet Management
Dashboards, …

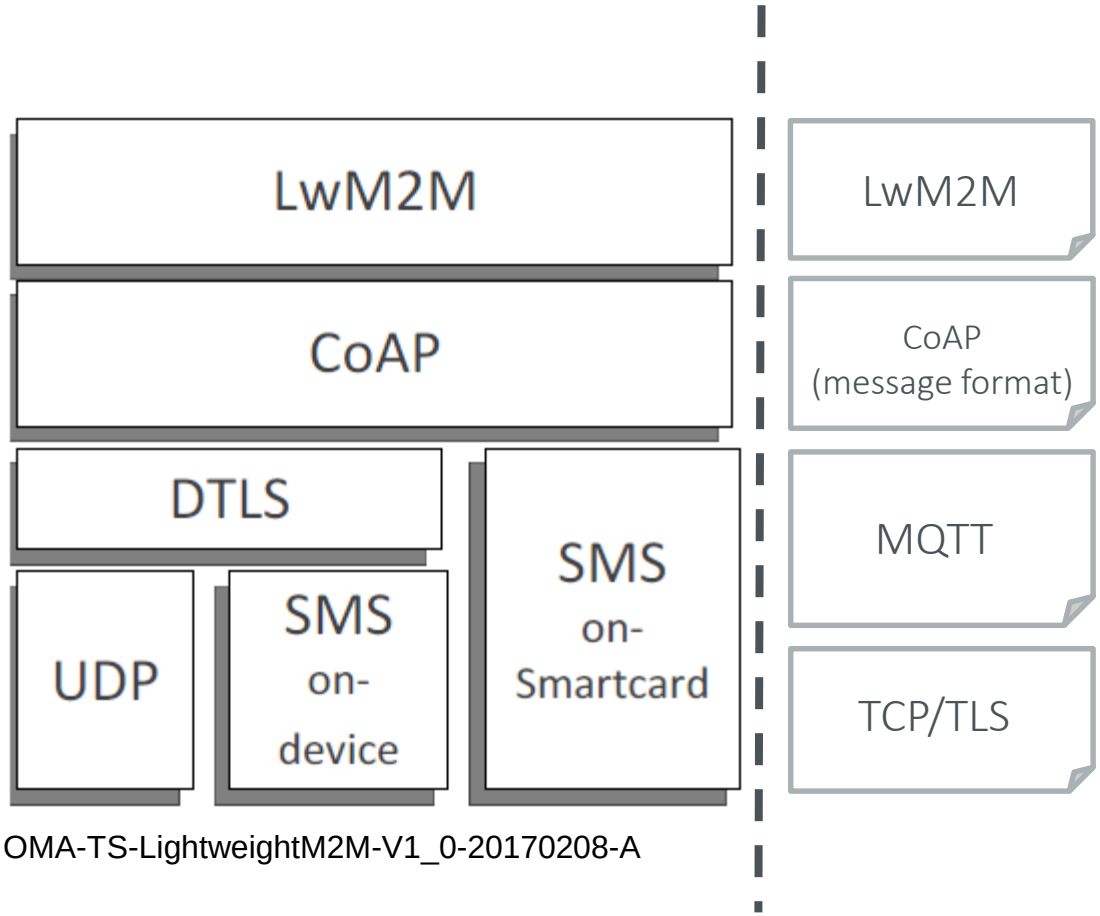
LWM2M Server

MQTT Server

Devices Field



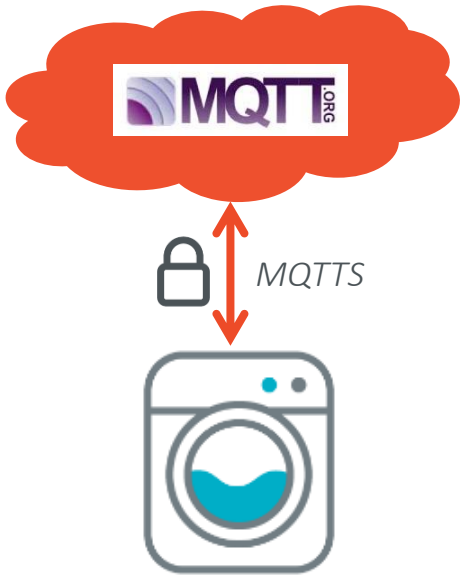
LwM2M (extended) Protocol Stack



OMA-TS-LightweightM2M-V1_0-20170208-A

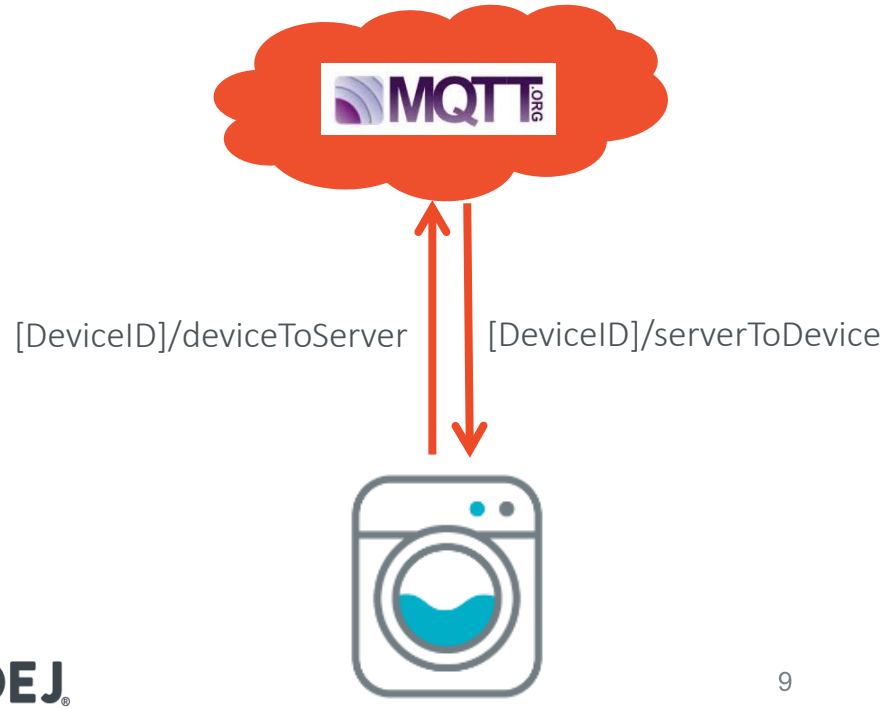
Key Point - Single Link to the Cloud

- For all kind of data transfer
  - Data & Device management
- Security
  - Not Yet Another Link to Secure
  - No additional protocol for large data transfer (firmware upgrade)
- Device Code Footprint
  - Embed only one vertical stack



Key Point - Dedicated Topics for LwM2M

- 2 general purpose topics: “transport topics”
  - “deviceToServer” for ascending messages
  - “serverToDevice” for descending messages
- Prefixed per device
- Payload = CoAP messages



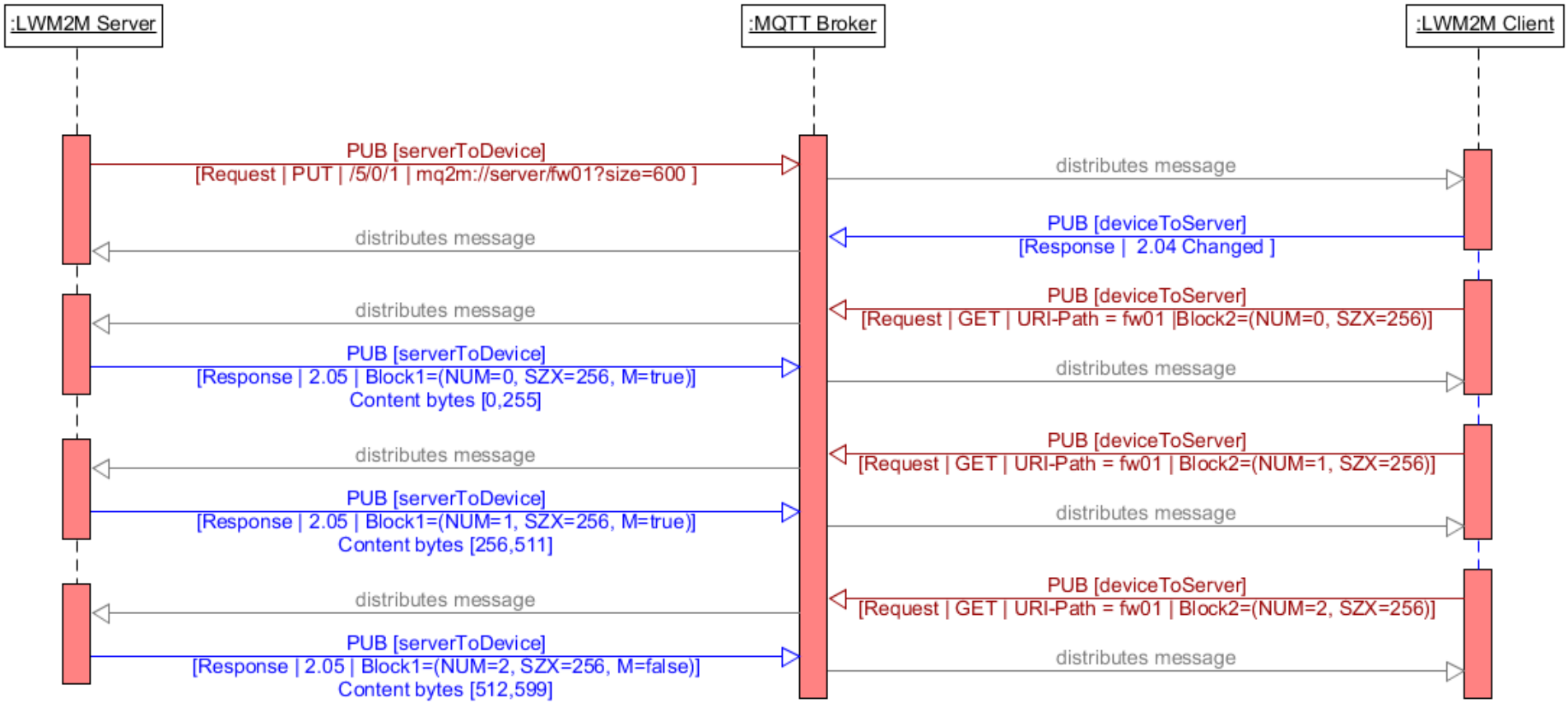
Key Point - Remove CoAP Reliability Stuff

- Connection is maintained at TCP layer
- No Confirmable (CON) and Acknowledgment (ACK) messages
- No duplication
  - QoS set to 0
  - Retain mode set to false
  - CleanSession flag set to 1
- Only CoAP compact message format is kept

Key Point - Large Binary Transfer

- Firmware Upgrade / Application Upgrade
- Use of LwM2M Package URI resources (no Package)
- A new URI scheme for fetching a resource
  - `mq2m://[owner]/[path]?size=xxx`
  - `mq2m://server/firmware-1.1.0?size=12345678`
- Device controls the download rhythm/rate/schedule
- Block oriented
  - Using Block-wise Transfer Block Options semantic

Large Binary Transfer - Example



Current Specification Hosting

- ESR - Publicly available for comments and implementation
- http://e-s-r.net/download/specification/ESR030-LWM2M-MQTT-1.0-A.pdf

Specification Next Steps

- Ownership
  - Move to appropriate organization OMA
  - Anyone interested can help us on the process ?

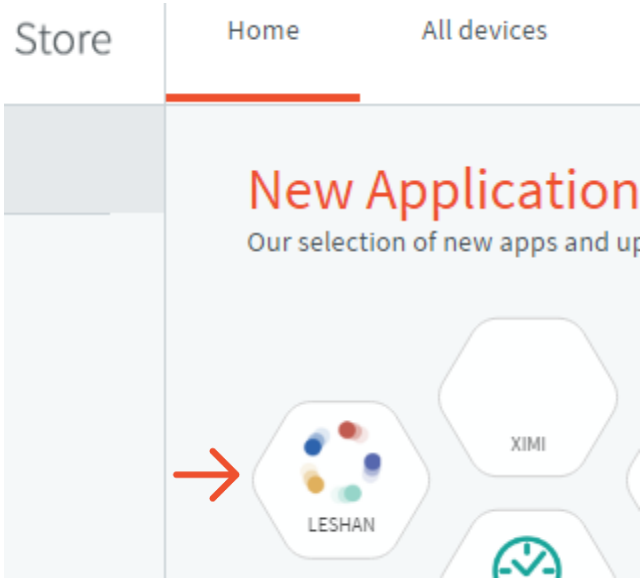
- Content
  - Final update against Feb 2017 LwM2M 1.0 release
  - Self-contained specification
    - No CoAP IETF dependency
    - CoAP message format may be derived
  - Proposal for LwM2M specification split
    - LwM2M semantics & core normative objects
    - LwM2M bindings (UDP / SMS / TCP / MQTT…)

MicroEJ Reference Implementation (1)

- Leshan client port on MicroEJ (ARM Cortex-M)

MicroEJ Reference Implementation (2)

- MicroEJ Leshan Server runs LwM2M over MQTT
  - http://leshan.microej.com:8080/#/clients
- MicroEJ Leshan Client App available on MicroEJ Store
  - https://communitystore.microej.com



Reference Implementation Next Steps

- Introduced to Eclipse IOT WG call
- Can be delivered as a Leshan sub-project
- Suggest core modifications to main stream to allow to plug this new backend

Frequently Asked Questions

- Why not LwM2M over CoAP over TCP ?
  - Reduced CoAP scope to only cover the LwM2M case
  - Would break the assumption to keep a single secured link for everything
- Why over MQTT ?
  - Assumption that MQTT is a requirement
- What about IETF “draft-koster-core-coap-pubsub-05” ?
  - This is a proposal for a pub/sub paradigm on top of CoAP

THANK YOU

FOR YOUR ATTENTION!