CoAP usages for Device Management

Jaime Jiménez
jaime.jimenez@ericsson.com

Managing Networks of Things workshop
draft-jimenez-t2trg-coap-functionality-lwm2m

@jaim - jaimejim.github.io

ERICSSON

Constrained Application Protocol (CoAP)

- It is a RESTful protocol for constrained devices and networks, very similar to HTTP.
  - Client/server & Request/Response
  - GET, POST, PUT, DELETE but also PATCH, iPATCH, FETCH Methods.
  - Same concepts (Media types, URL, URN…)
- The well-known URI
  - `GET coap://[ip6address]/.well-known/core`
- IPv6 oriented (using 6LowPAN)
  - IP Multicast support

Constrained Application Protocol (CoAP)

- Resource discovery via the Resource Directory (RD)
- Compact 4-byte Header
- Can run on UDP or SMS
  - Reliability is ensured by using with different message types:
    - Confirmable (CON), non-confirmable (NON), acknowledgement (ACK) and reset (RST).
- TCP currently being standardised.
- Observe/Notify, adding an “observe” flag in the CoAP GET Request
  - Introduces a Publish/Subscribe model for constrained devices.
- Facilitates new ways of interacting with devices and managing them
  - CoMI/CoOL
  - LWM2M

Constrained Management and Objects Language (CoOL/CoMI)

- Describes a management function set adapted for constrained devices and constrained networks using YANG.
- Interactions with objects use CoAP a application protocol.
- Payloads are encoded in CBOR data format.

Roadmap

Current targets

Encoding
I-D.ietf-core-yang-cbor

Identifiers
I-D.somaraju-core-sid

Protocol operations
I-D.veillette-core-cool

Discovery
I-D.veillette-core-cool-library

Future work

Security
- Boot strapping
- Authorization

(Profile of existing methods)

Protocol extensions
- Multicast
- Binding table
- Application management
- OTA upgrade

Support for LWM2M

Constrained Management and Objects Language (CoOL/CoMI)

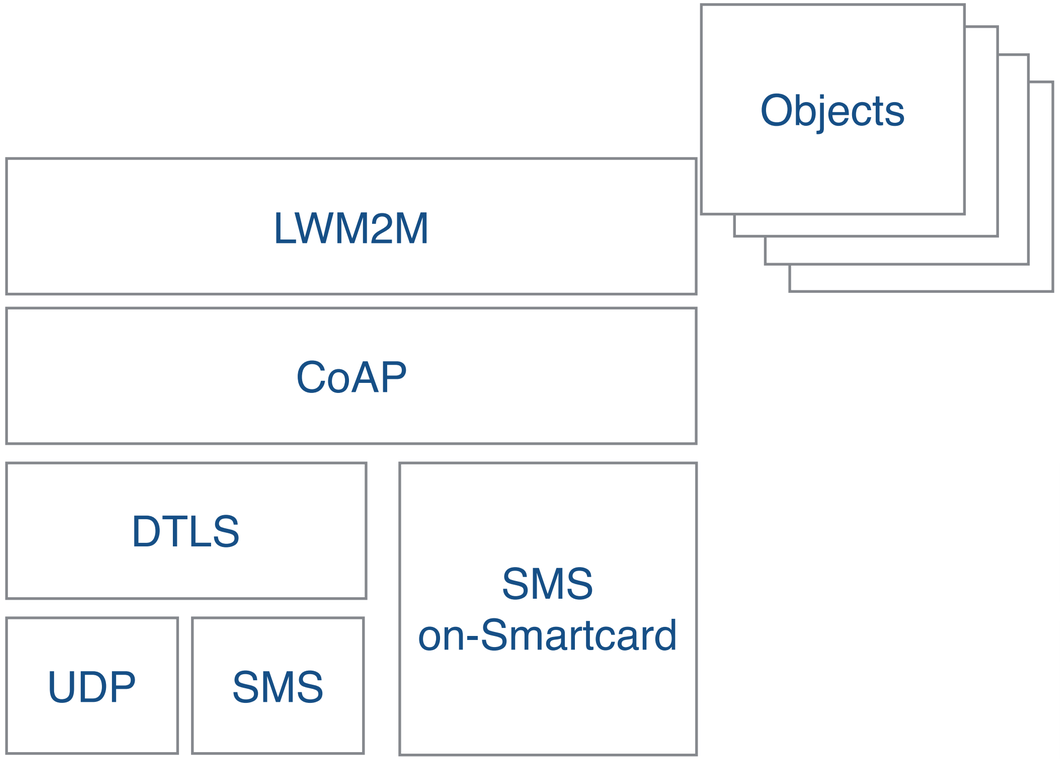
- Similar to RESTCONF but:
  - uses CoAP/UDP as transfer protocol. RESTCONF uses HTTP/TCP.
  - uses YANG-CBOR as payload format. RESTCONF uses YANG-JSON or YANG-XML.
  - *CoMI encodes YANG identifiers as numbers, where RESTCONF encodes them into strings (WIP).*
  - CoMI uses the methods FETCH and iPATCH, not used by RESTCONF (because HTTP does not have that)
  - RESTCONF uses the HTTP methods HEAD, and OPTIONS, which are not used by CoMI (because CoAP does not have that).
  - *YANG used as modelling language but no specific data model (WIP).*

  - … and many more at https://tools.ietf.org/html/draft-vanderstok-core-comi-10#page-7

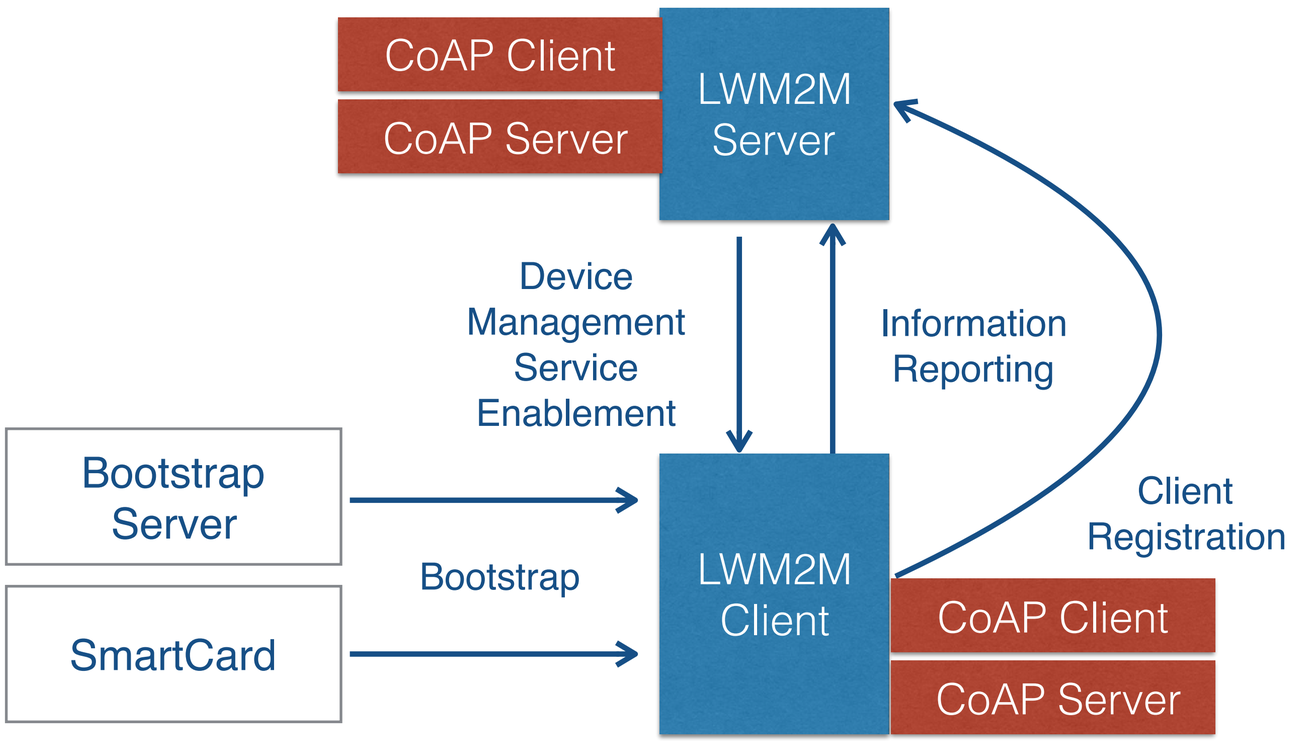
OMA Lightweight M2M (LWM2M)

- Management Interfaces for CoAP.
  - Bootstrap: bootstrapping and upgrading a device.
  - Registration: taking a device into a logical group.
  - Device Management: by writing / creating objects inside the device.
  - Information Reporting: reading objects inside a device.
- LWM2M defines the Object Model.
  - Objects can correspond to sensors or actuators.
  - *Defines data model but has no modelling language (XML kinda).*

OMA Lightweight M2M (LWM2M)



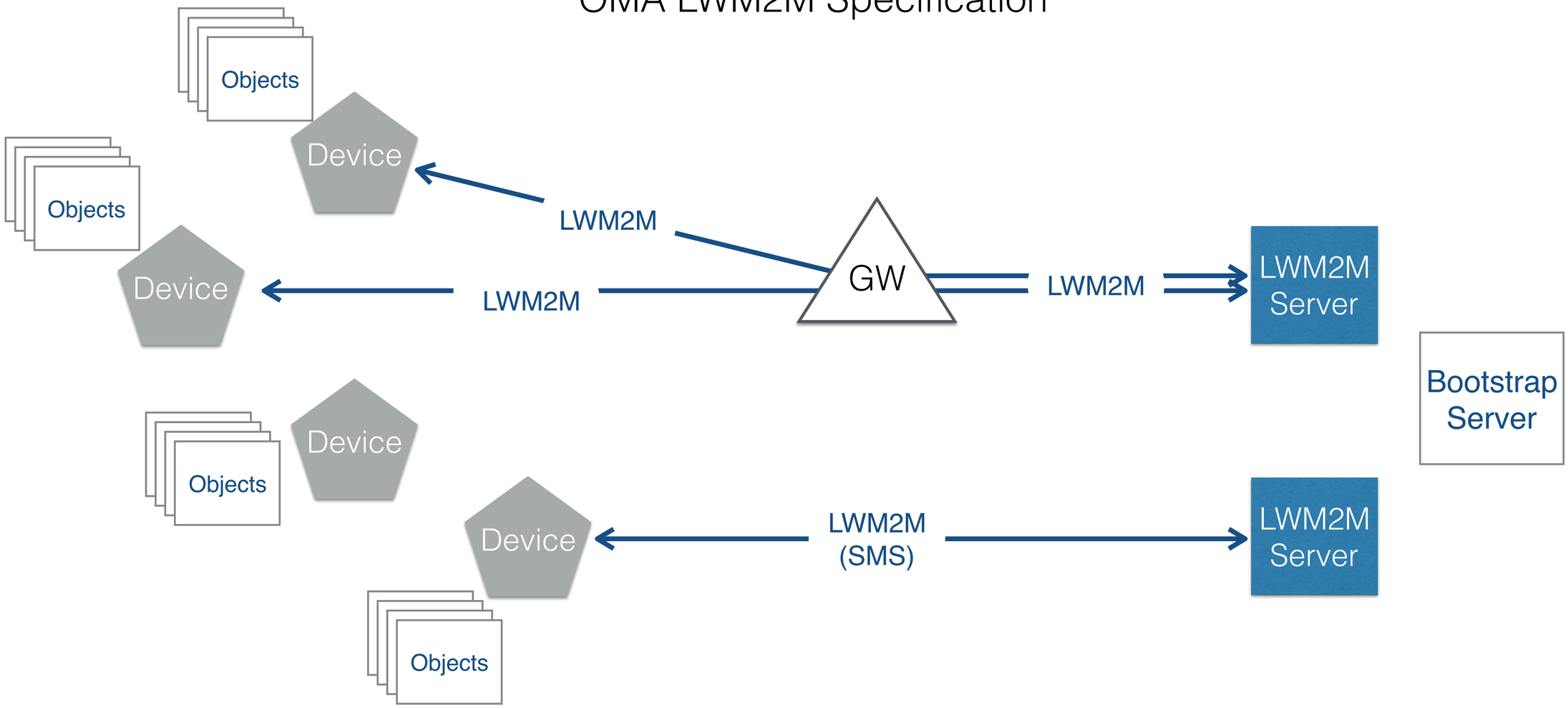
LWM2M Device Stack



LWM2M Architecture

LWM2M Interactions

OMA LWM2M Specification



Possible LWM2M Additions

1. **Device and Manager configuration.**

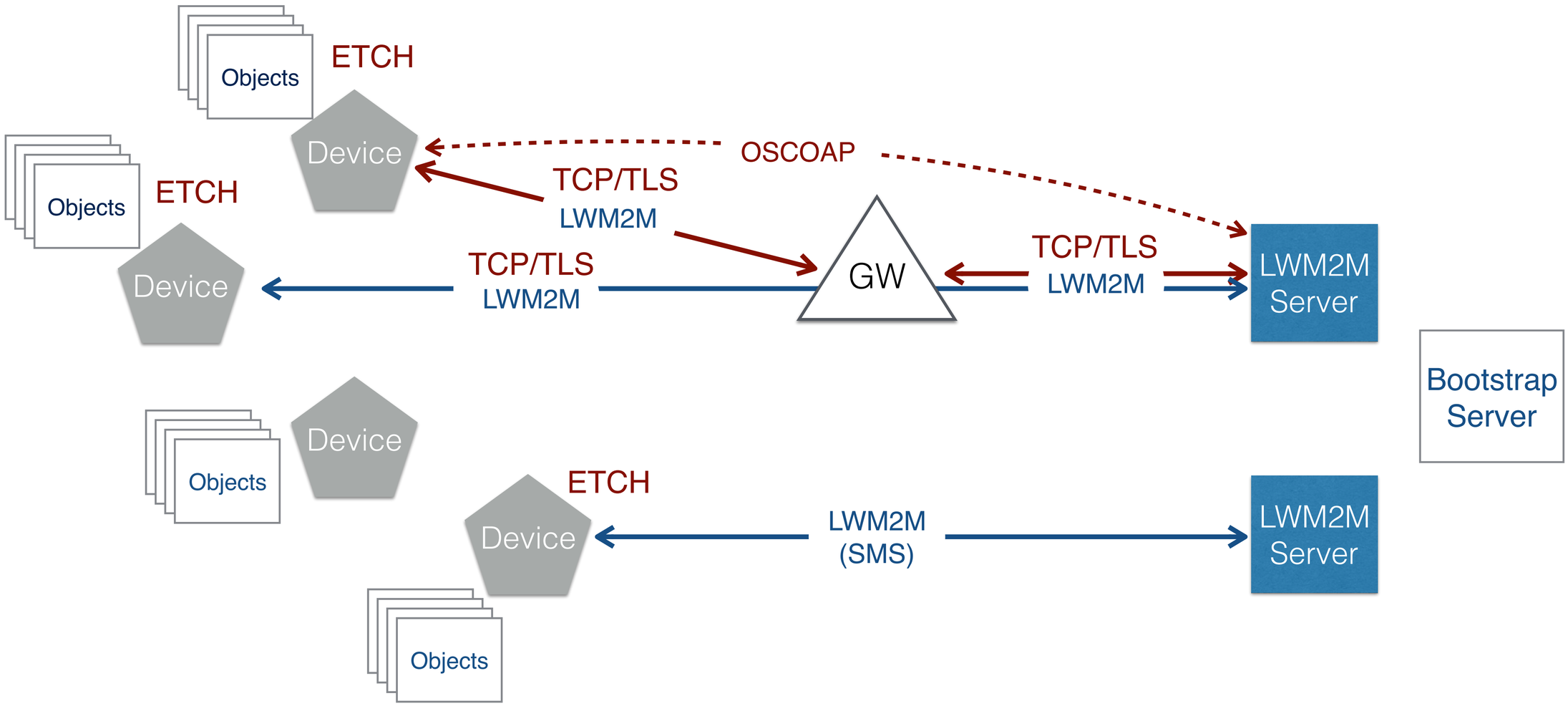
   Currently covered by LWM2M.

- *[I-D.ietf-core-coap-tcp-tls]* outlines the changes required to use CoAP over TCP, TLS, and WebSockets transports.

- *[I-D.ietf-core-object-security]* For systems in which endpoints work behind a gateway or use LWM2M for managing the gateways, it might be good to implement other types of cryptographic protection than TLS/DTLS.

- *[I-D.ietf-core-etch]* Support for features like PATCH/FETCH could be greatly beneficial for things like firmware upgrade or observing relatively large sets of resources.

Possible LWM2M Additions



Possible LWM2M Additions

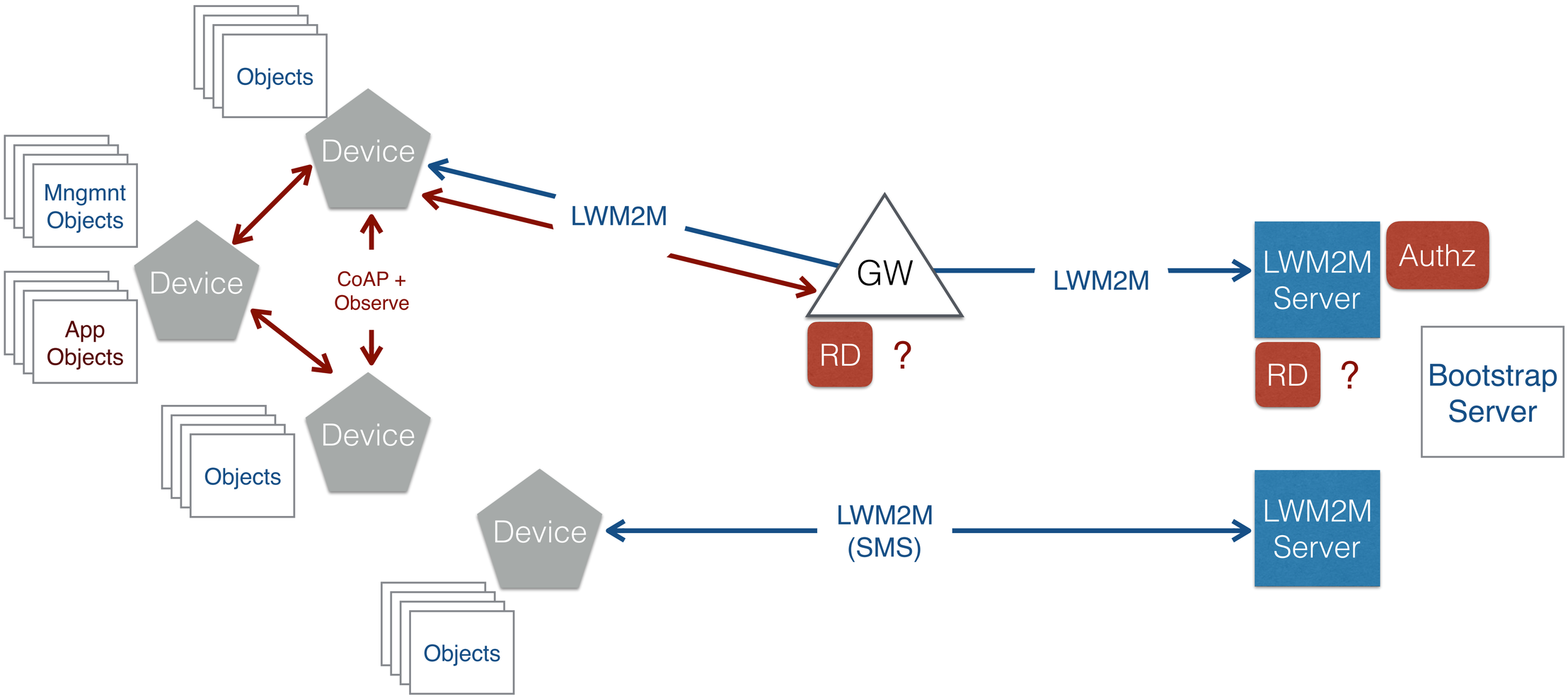
2. Device to Device configuration.

- *[I-D.ietf-core-resource-directory]* CoAP's in-built discovery would be beneficial to support cases in which devices talk to each or in which a more autonomous management approach is preferred. For now devices under the same subnet can use IP multicast as expressed on [RFC7390] and through /.well-known/core.

  Devices would support CoAP Observe [RFC7641] between each other in order to subscribe to updates from one another.

- *[I-D.ietf-ace-oauth-authz]* could be used as security framework and the LWM2M Server would act as Authorization Server.

Possible LWM2M Additions



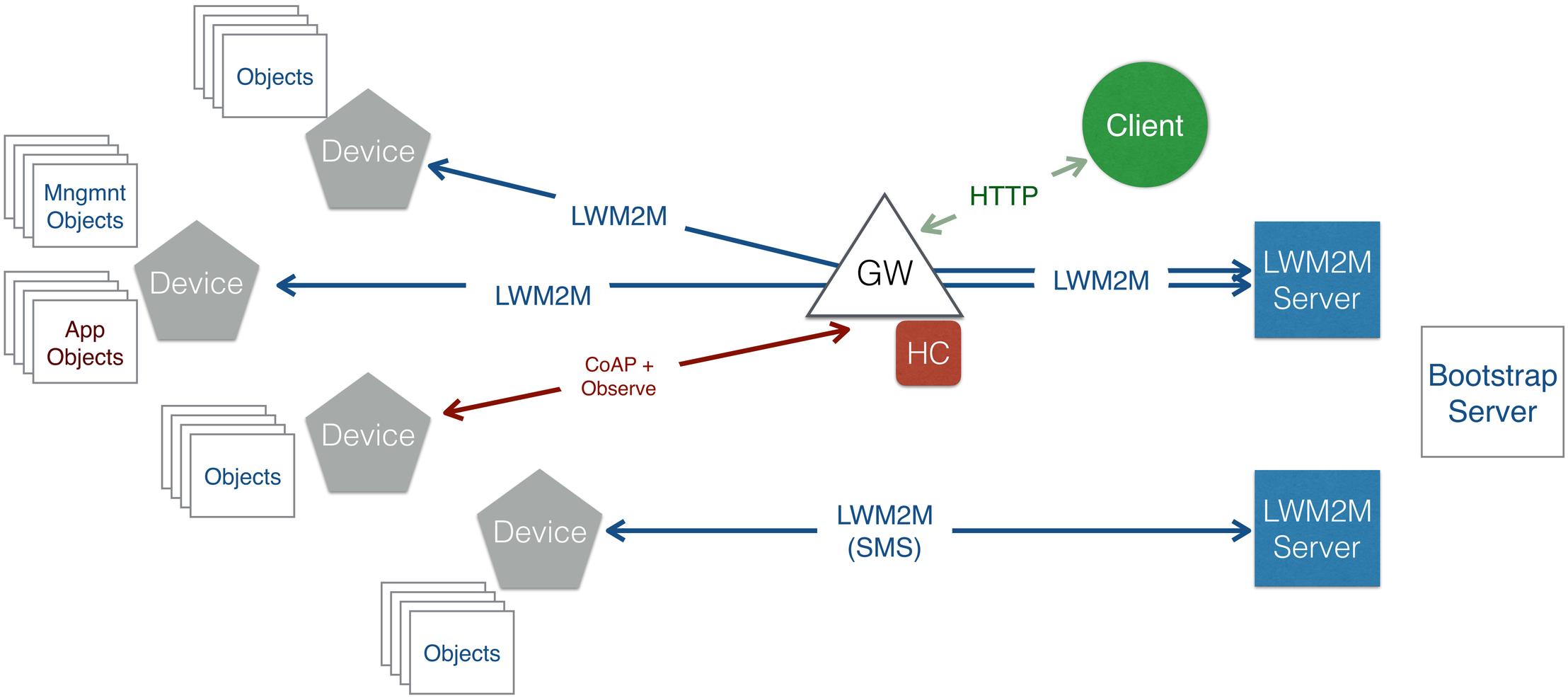
Possible LWM2M Additions

3. **Device to Application configuration.**

Including the aforementioned on (1) and (2).

[I-D.ietf-core-http-mapping] in cases of phone talking to GW. GW should implement a HC proxy.

Possible LWM2M Additions



LWM2M Data Model

- [RFC6690] Web Linking. ObjectLinks (String<ObjectID:InstanceID>) are not sufficient to represent links between devices or applications.

- Use unique ResourceIDs and register them to consistently use the same identifiers for the same resources.

- Update the serialization format [RFC7049]. JSON can be greatly compressed to CBOR format.

- A lot of work has happened on the Data Model space, perhaps it is time to revisit the Object Model. [IOTSI]

Assorted References

REST	https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm
CoAP	https://tools.ietf.org/html/rfc7252
CoRE Link-Format	https://tools.ietf.org/html/rfc6690
CoAP Observe	https://tools.ietf.org/html/rfc7641
CBOR	https://tools.ietf.org/html/rfc7049
IOTSI	https://www.iab.org/activities/workshops/iotsi/
IOTSU	https://www.iab.org/activities/workshops/iotsu/
CoRE RD	https://datatracker.ietf.org/doc/draft-ietf-core-resource-directory/
LWM2M	https://github.com/OpenMobileAlliance/
CoMI	https://tools.ietf.org/wg/core/draft-ietf-core-yang-cbor/
CoAP-SNMP Interworking	https://tutcris.tut.fi/portal/files/1076133/lindholm_ventola_coap_snmp_interworking.pdf
CoAP TCP+TLS	https://tools.ietf.org/wg/core/draft-ietf-core-coap-tcp-tls/
IPSO	http://ipso-alliance.github.io/pub/
LWM2M to YANG	https://tools.ietf.org/html/draft-vanderstok-core-yang-lwm2m-00
OSCOAP	https://tools.ietf.org/wg/core/draft-ietf-core-object-security/
CoAP for LWM2M	https://tools.ietf.org/html/draft-jimenez-t2trg-coap-functionality-lwm2m