Zephyr® Project

Developer Summit

Zephyr® Project
Developer Summit

Connecting and Managing Zephyr Devices Remotely Using LWM2M and Eclipse Leshan

Julien Vermillard, *EdgeIQ*

@vrmvrm

https://www.linkedin.com/in/jvermillard

Agenda

Overview of OMA Lightweight M2M

- Bootstrap
- Registration
- Object model

Introduction to Eclipse Leshan

Zephyr’s built in LWM2M client

When to use LWM2M and up & downs

CoAP the Constrained Application Protocol

Replace TCP/HTTP in a UDP based light protocol

RESTful protocol designed from scratch

- URIs, Internet Media Types
- GET, POST, PUT, DELETE

Transparent mapping to HTTP

Additional features for M2M scenarios

- Observe
- Blockwise transfer

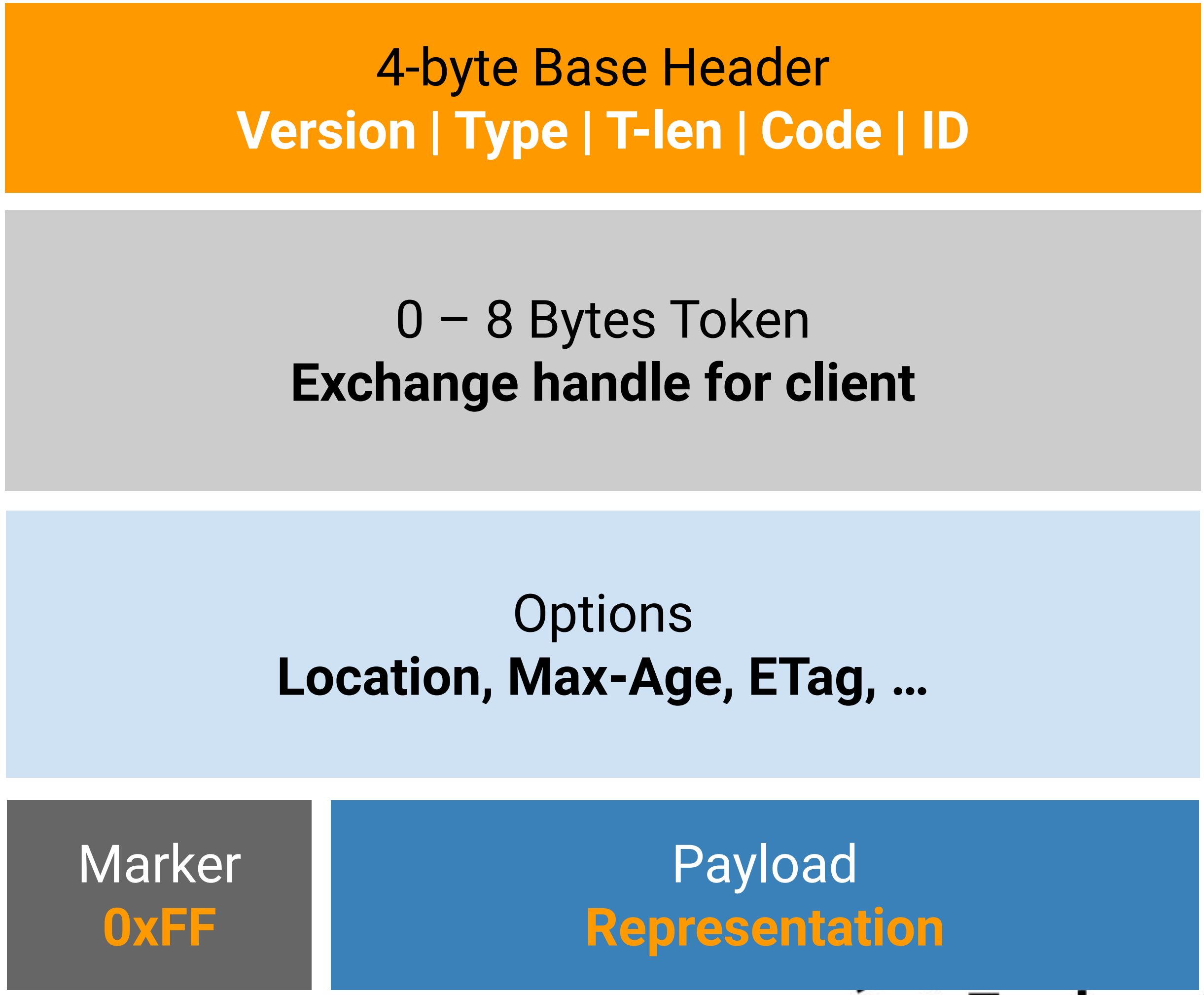
CoAP the Constrained Application Protocol

Binary protocol

- Low parsing complexity
- Small message size

Options

- Binary HTTP-like headers



4-byte Base Header

Version | Type | T-len | Code | ID

0 – 8 Bytes Token

Exchange handle for client

Options

Location, Max-Age, ETag, …

Marker

0xFF

Payload

Representation

Zephyr® Project
Developer Summit

Lightweight M2M

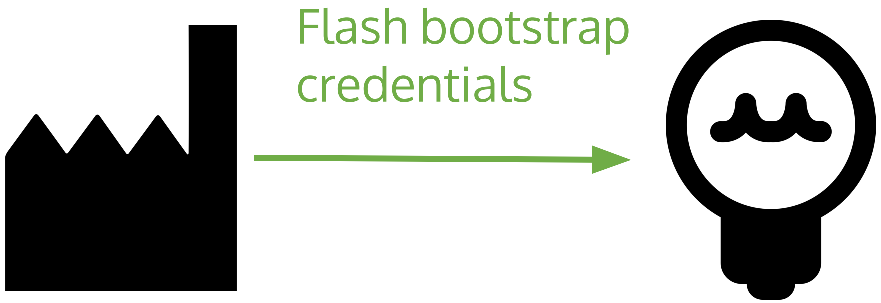
OMA Lightweight M2M

A standard device management Protocol on top of CoAP

First published in 2017 (v1.0) by the Open Mobile Alliance

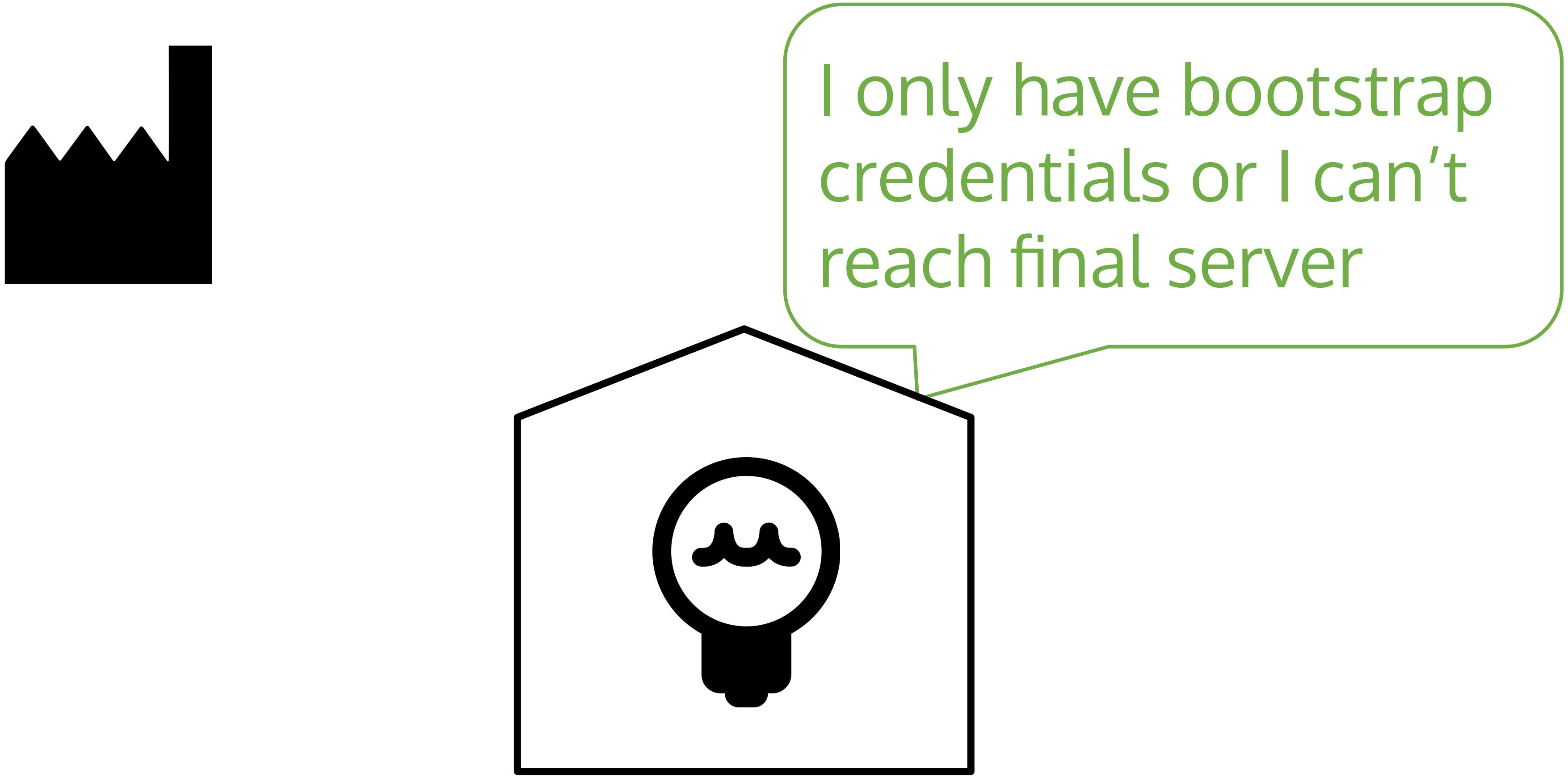
Mean to replace OMA-DM for new LPWA networks (LTE-M, NB-IoT)

Bootstrap phase



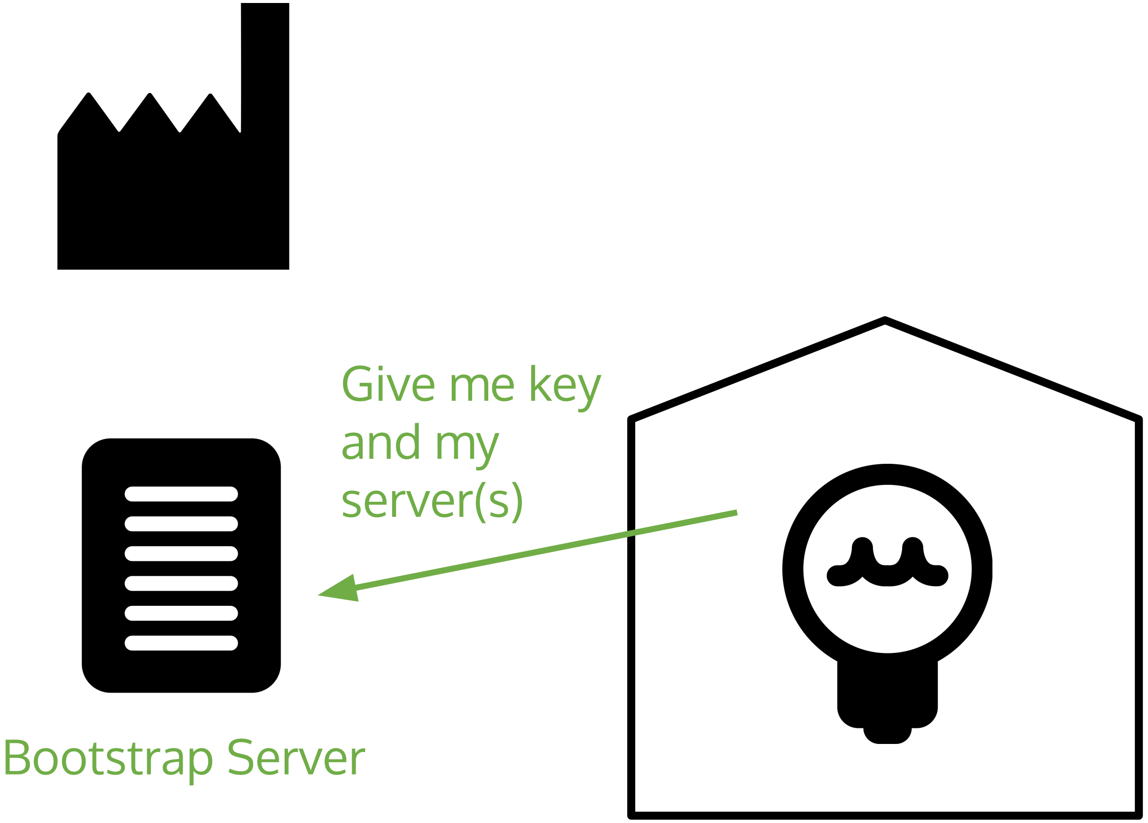
Flash bootstrap credentials

Bootstrap phase

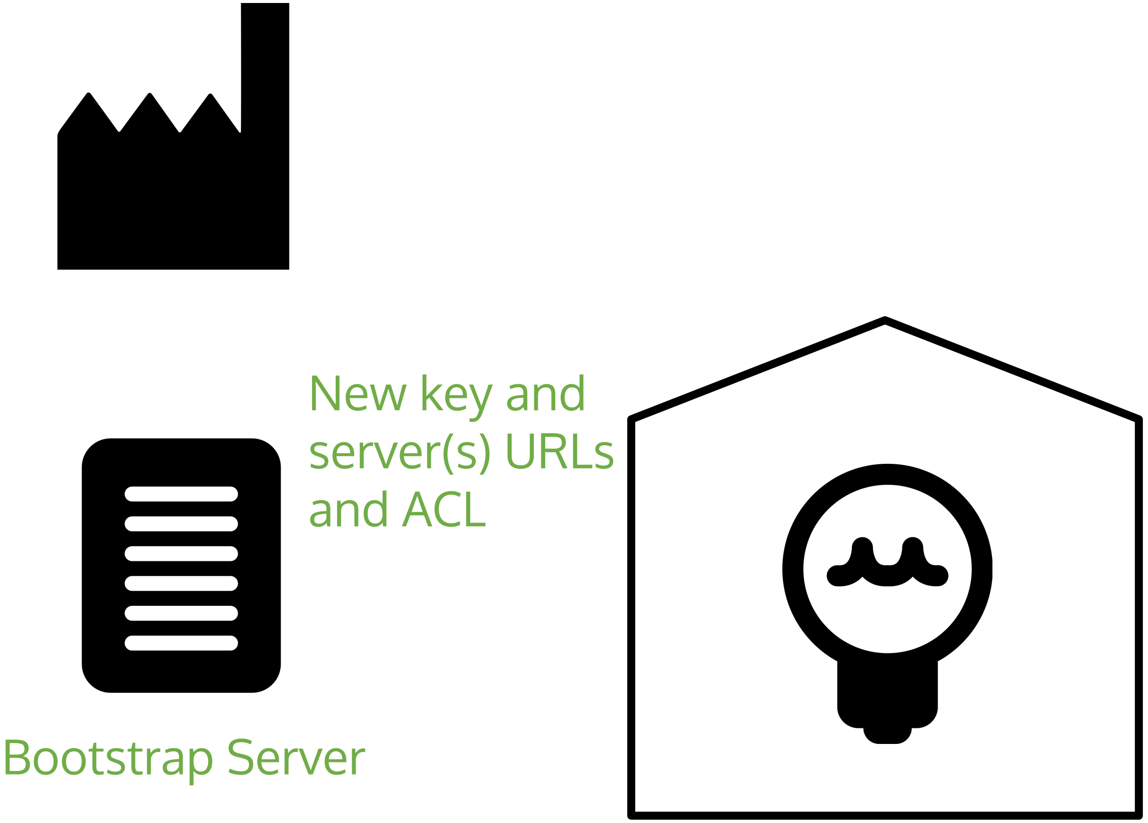


I only have bootstrap credentials or I can’t reach final server

Bootstrap phase



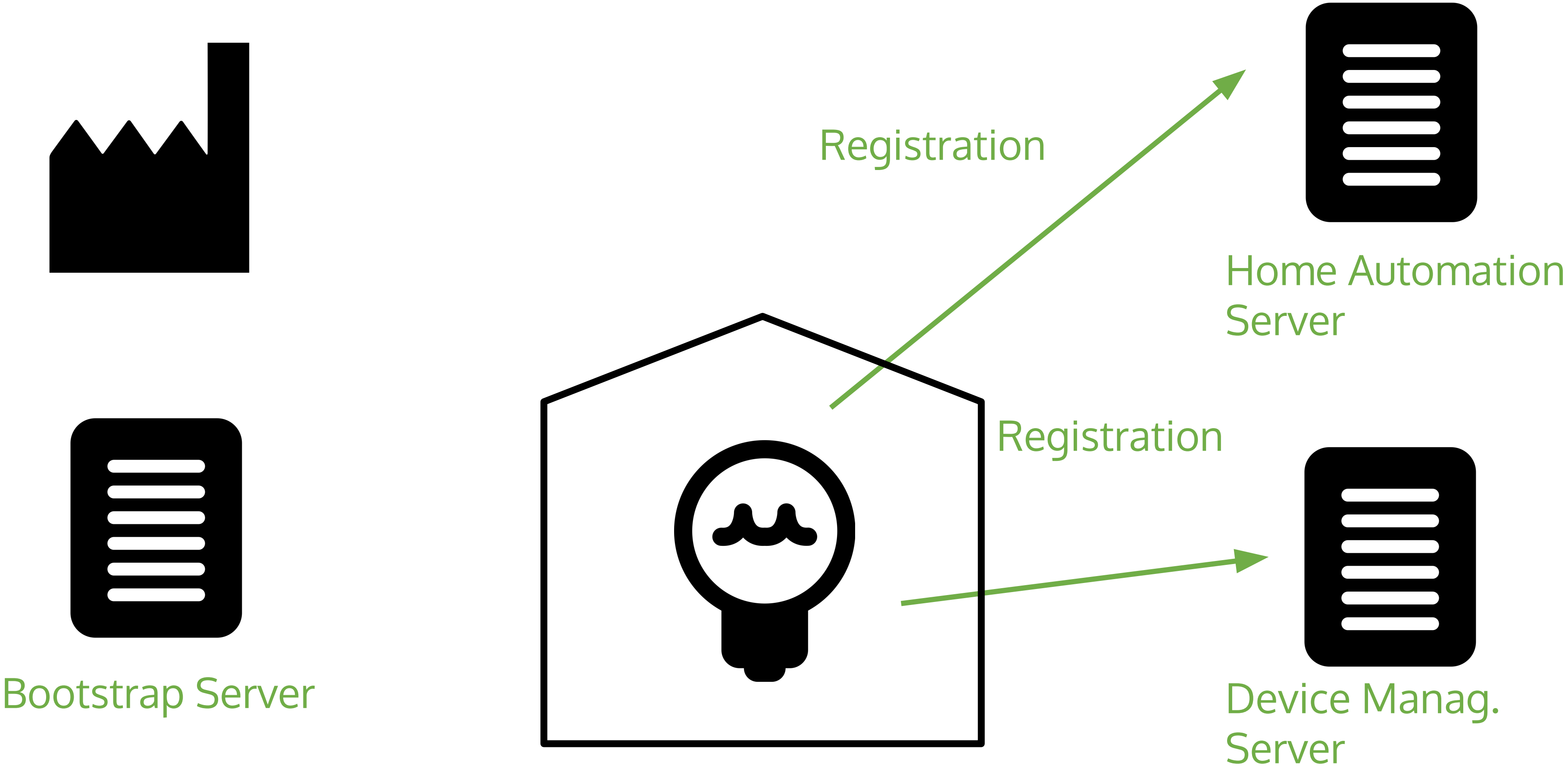
Bootstrap phase



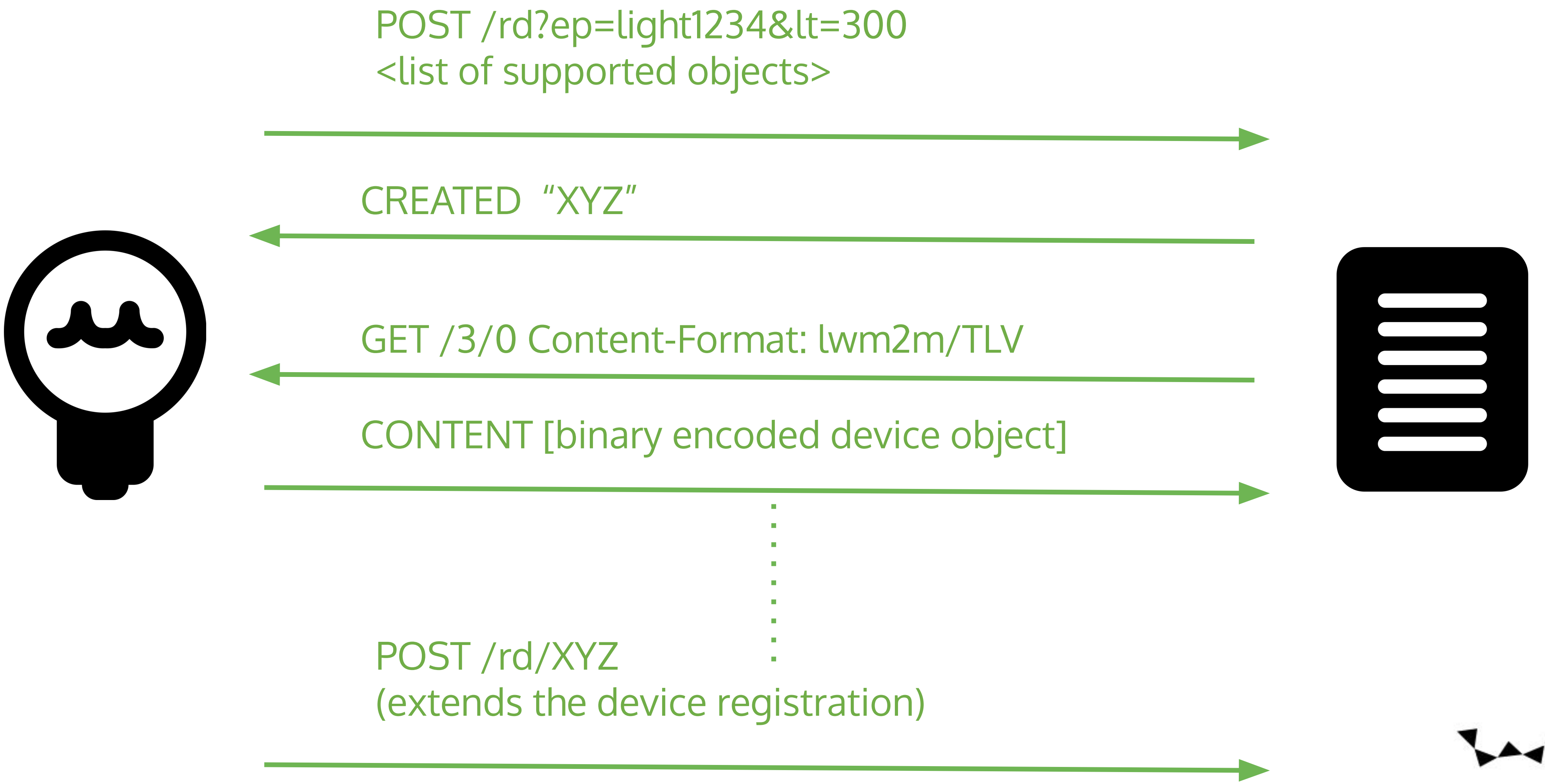
New key and server(s) URLs and ACL

Bootstrap Server

Bootstrap phase



Registration phase



POST /rd?ep=light1234&lt=300
<list of supported objects>

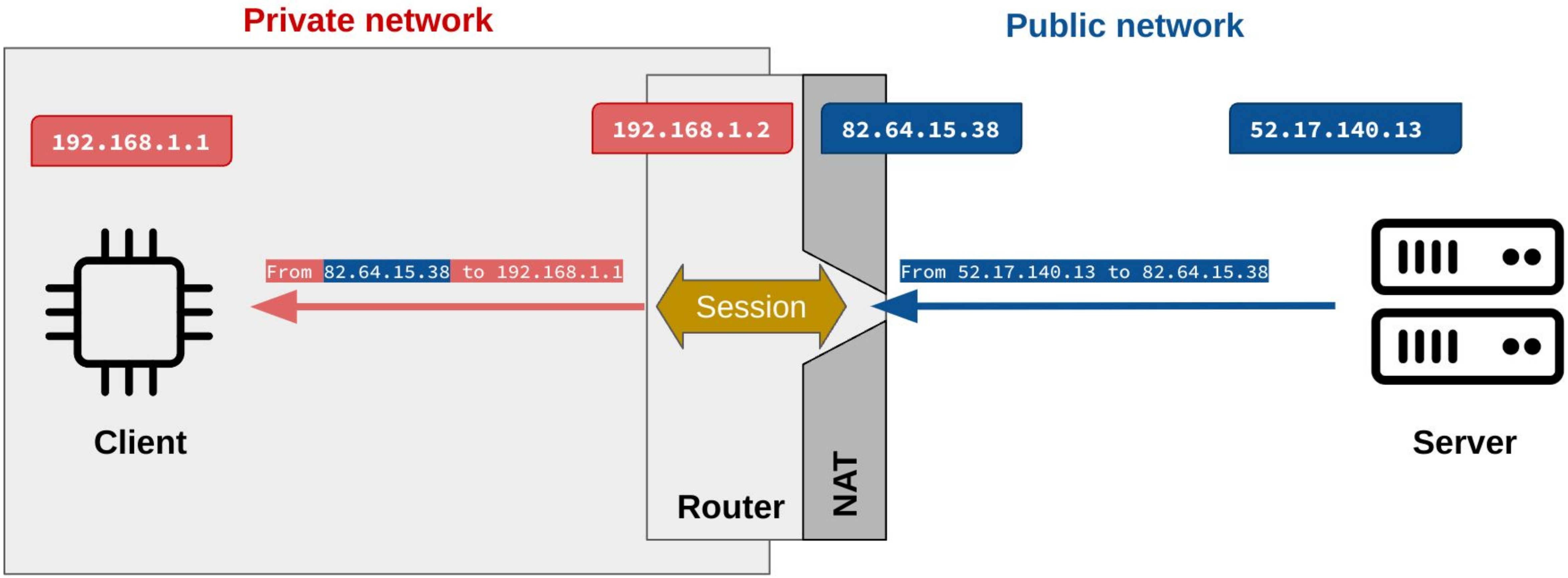
CREATED “XYZ”

GET /3/0 Content-Format: lwm2m/TLV

CONTENT [binary encoded device object]

POST /rd/XYZ
(extends the device registration)

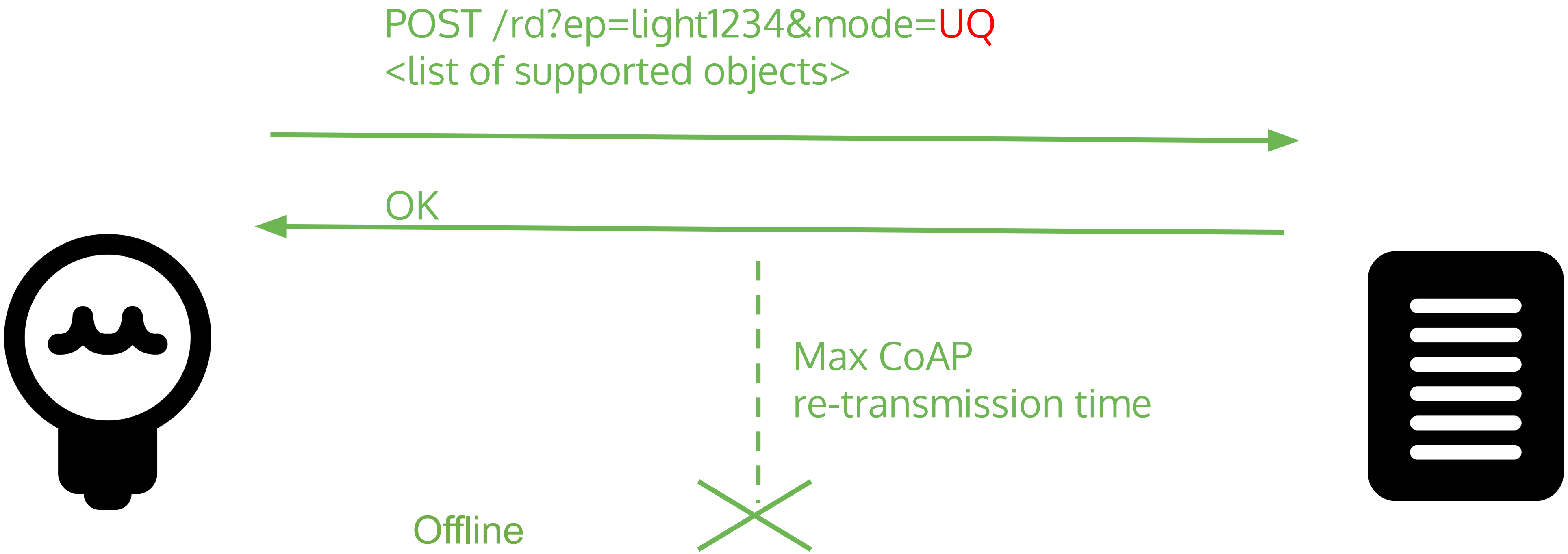
Registration & NAT



Zephyr® Project
Developer Summit

Queue Mode

Specified by the client as a registration mode parameter,
Mean the Device will Register and disconnect after a given amount of time, to save power



Object Model

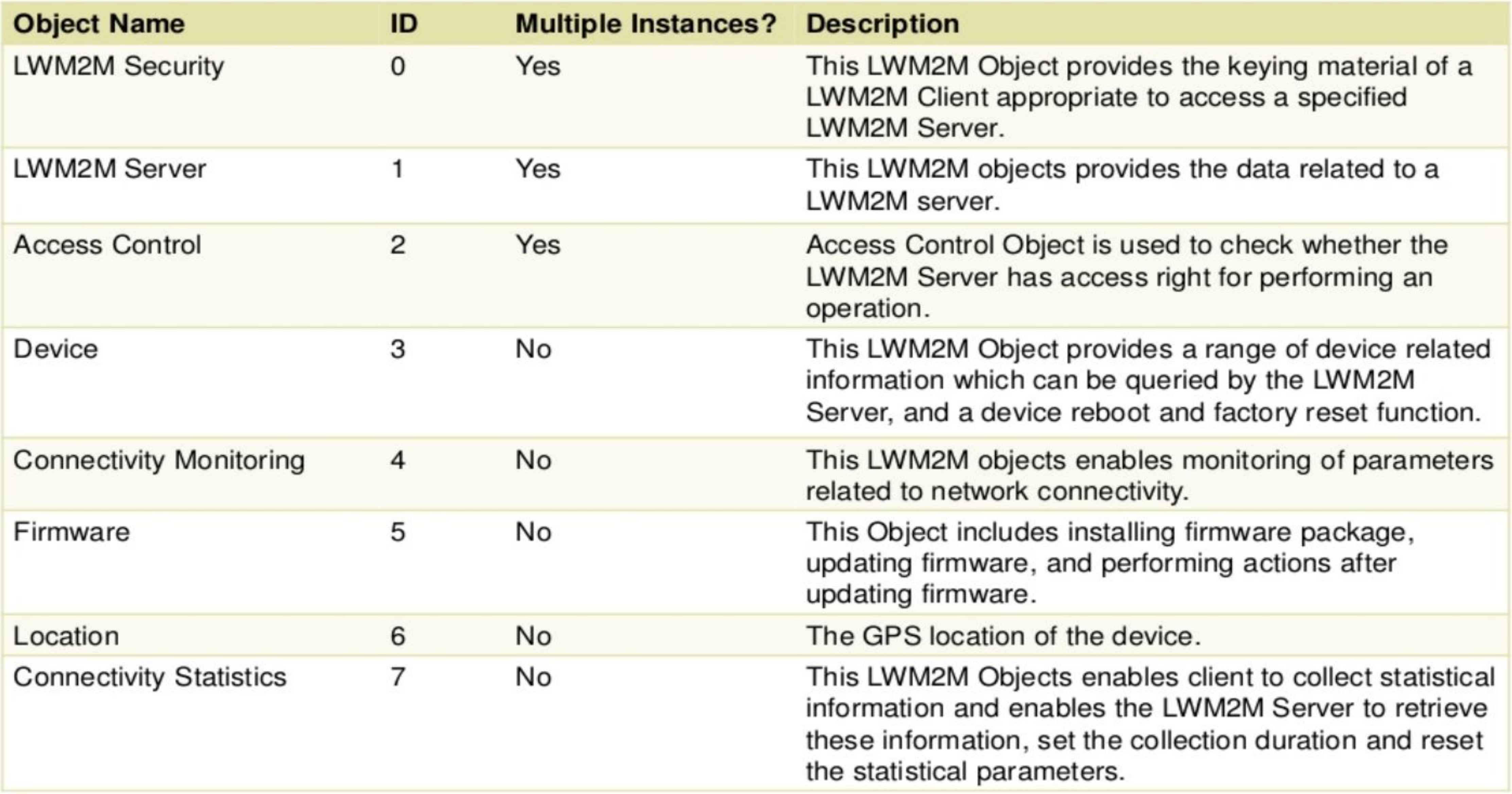
coap://{deviceip} / {object} / {instance} / {resource}

Examples:

"/6/0" the whole location object (binary record)

"/6/0/1" only the longitude (degree)

Object Model

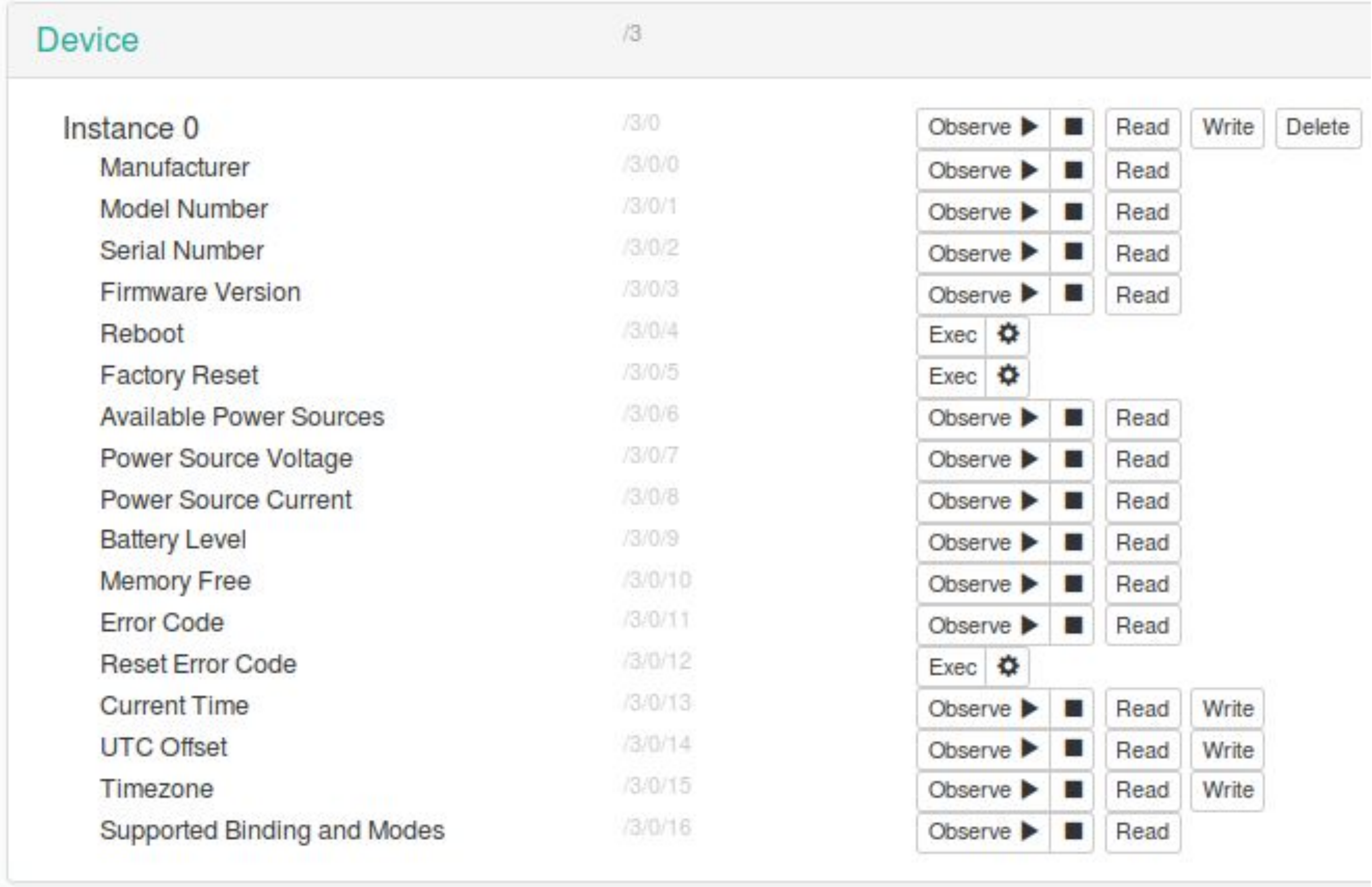


Object Name	ID	Multiple Instances?	Description
LWM2M Security	0	Yes	This LWM2M Object provides the keying material of a LWM2M Client appropriate to access a specified LWM2M Server.
LWM2M Server	1	Yes	This LWM2M objects provides the data related to a LWM2M server.
Access Control	2	Yes	Access Control Object is used to check whether the LWM2M Server has access right for performing an operation.
Device	3	No	This LWM2M Object provides a range of device related information which can be queried by the LWM2M Server, and a device reboot and factory reset function.
Connectivity Monitoring	4	No	This LWM2M objects enables monitoring of parameters related to network connectivity.
Firmware	5	No	This Object includes installing firmware package, updating firmware, and performing actions after updating firmware.
Location	6	No	The GPS location of the device.
Connectivity Statistics	7	No	This LWM2M Objects enables client to collect statistical information and enables the LWM2M Server to retrieve these information, set the collection duration and reset the statistical parameters.

oject

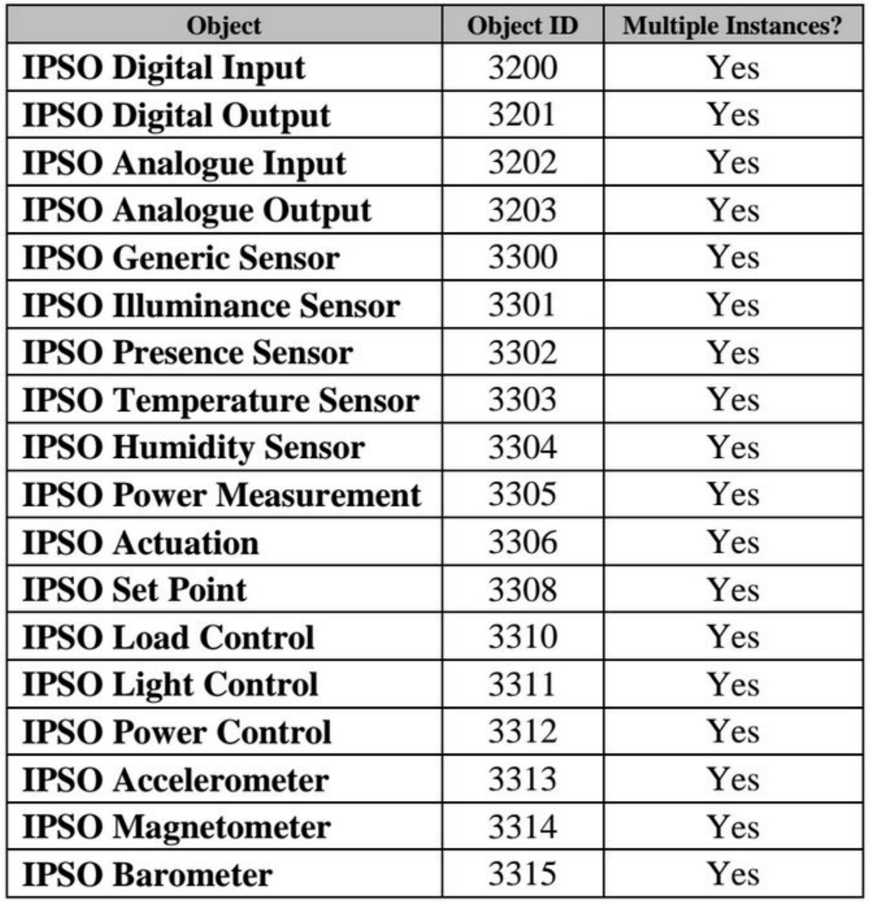
nit

Object Device

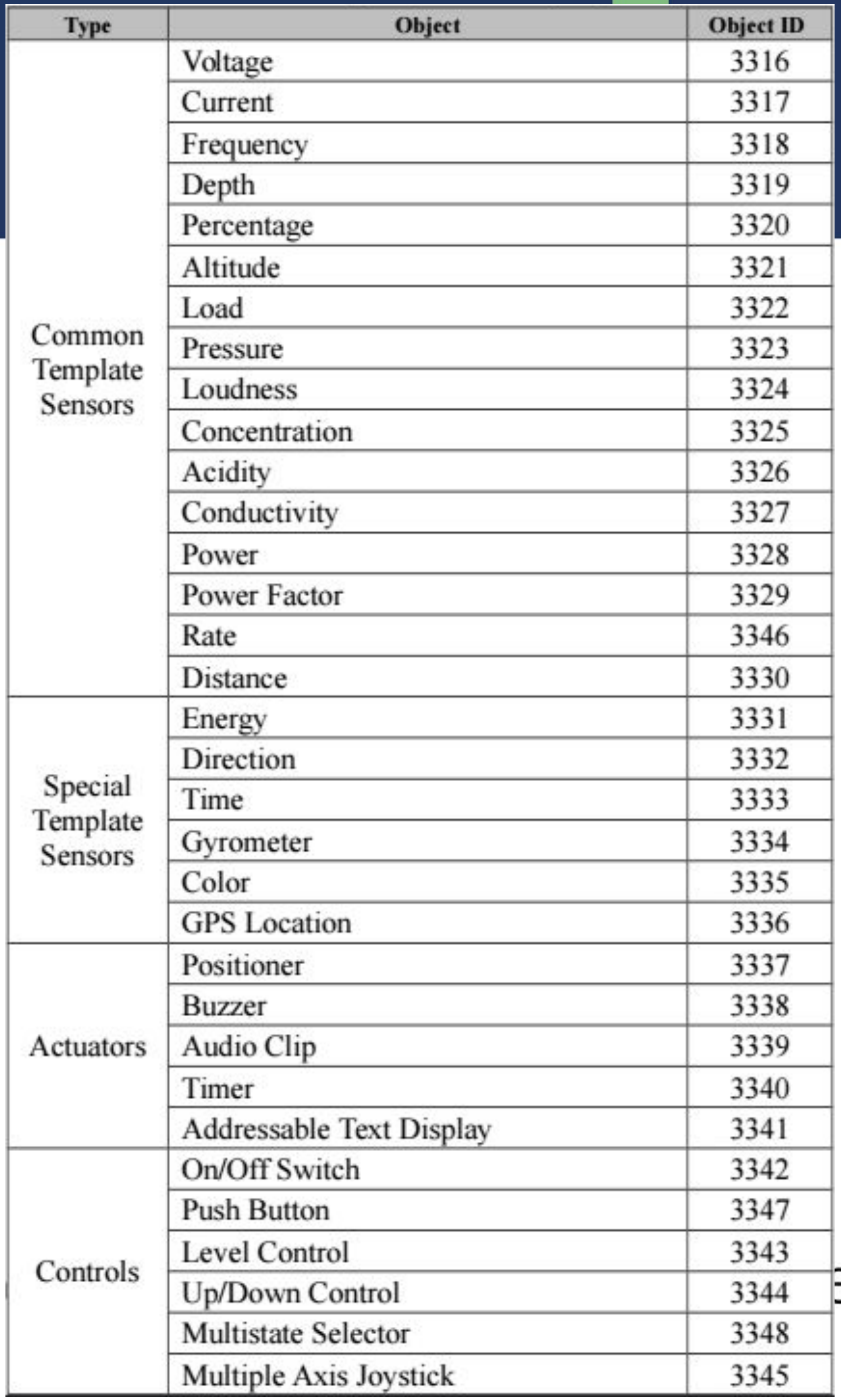


Device	/3	
Instance 0	/3/0	Observe ▶ ■ Read Write Delete
Manufacturer	/3/0/0	Observe ▶ ■ Read
Model Number	/3/0/1	Observe ▶ ■ Read
Serial Number	/3/0/2	Observe ▶ ■ Read
Firmware Version	/3/0/3	Observe ▶ ■ Read
Reboot	/3/0/4	Exec ⚙
Factory Reset	/3/0/5	Exec ⚙
Available Power Sources	/3/0/6	Observe ▶ ■ Read
Power Source Voltage	/3/0/7	Observe ▶ ■ Read
Power Source Current	/3/0/8	Observe ▶ ■ Read
Battery Level	/3/0/9	Observe ▶ ■ Read
Memory Free	/3/0/10	Observe ▶ ■ Read
Error Code	/3/0/11	Observe ▶ ■ Read
Reset Error Code	/3/0/12	Exec ⚙
Current Time	/3/0/13	Observe ▶ ■ Read Write
UTC Offset	/3/0/14	Observe ▶ ■ Read Write
Timezone	/3/0/15	Observe ▶ ■ Read Write
Supported Binding and Modes	/3/0/16	Observe ▶ ■ Read

IPSO sensor objects



Object	Object ID	Multiple Instances?
IPSO Digital Input	3200	Yes
IPSO Digital Output	3201	Yes
IPSO Analogue Input	3202	Yes
IPSO Analogue Output	3203	Yes
IPSO Generic Sensor	3300	Yes
IPSO Illuminance Sensor	3301	Yes
IPSO Presence Sensor	3302	Yes
IPSO Temperature Sensor	3303	Yes
IPSO Humidity Sensor	3304	Yes
IPSO Power Measurement	3305	Yes
IPSO Actuation	3306	Yes
IPSO Set Point	3308	Yes
IPSO Load Control	3310	Yes
IPSO Light Control	3311	Yes
IPSO Power Control	3312	Yes
IPSO Accelerometer	3313	Yes
IPSO Magnetometer	3314	Yes
IPSO Barometer	3315	Yes

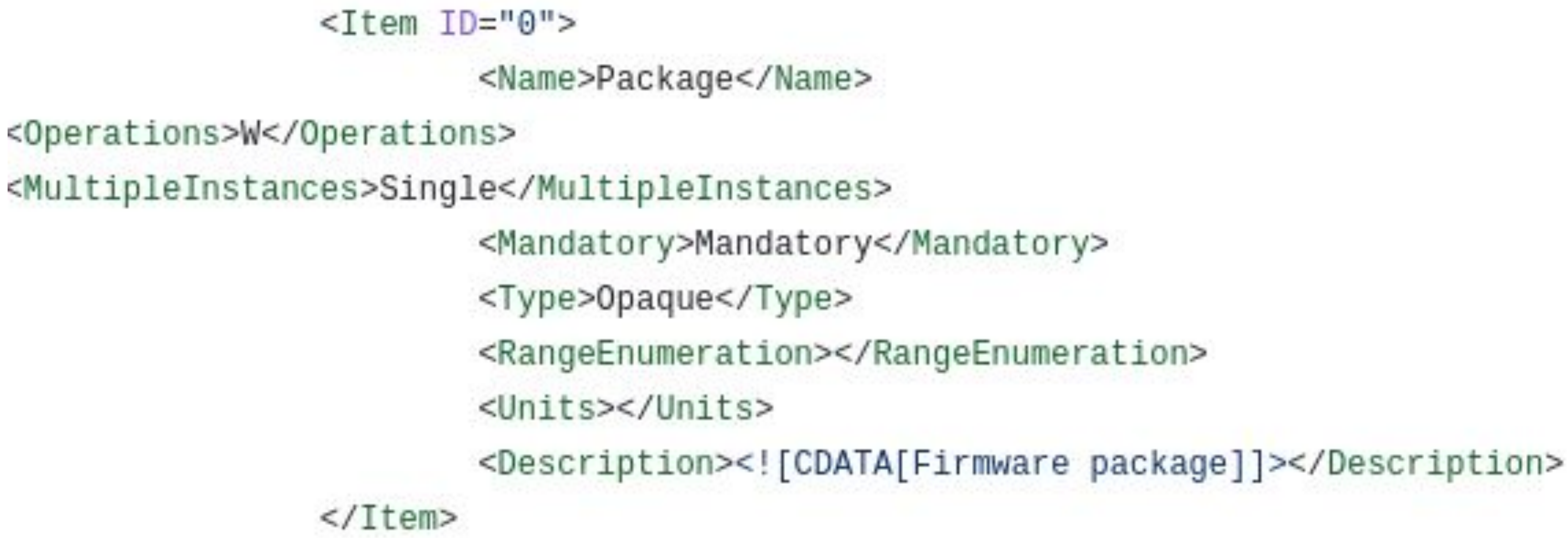


Type	Object	Object ID
Common Template Sensors	Voltage	3316
	Current	3317
	Frequency	3318
	Depth	3319
	Percentage	3320
	Altitude	3321
	Load	3322
	Pressure	3323
	Loudness	3324
	Concentration	3325
	Acidity	3326
	Conductivity	3327
	Power	3328
	Power Factor	3329
	Rate	3346
	Distance	3330
Special Template Sensors	Energy	3331
	Direction	3332
	Time	3333
	Gyrometer	3334
	Color	3335
	GPS Location	3336
Actuators	Positioner	3337
	Buzzer	3338
	Audio Clip	3339
	Timer	3340
	Addressable Text Display	3341
Controls	On/Off Switch	3342
	Push Button	3347
	Level Control	3343
	Up/Down Control	3344
	Multistate Selector	3348
	Multiple Axis Joystick	3345

Objects model

Objects are defined in a central repository:

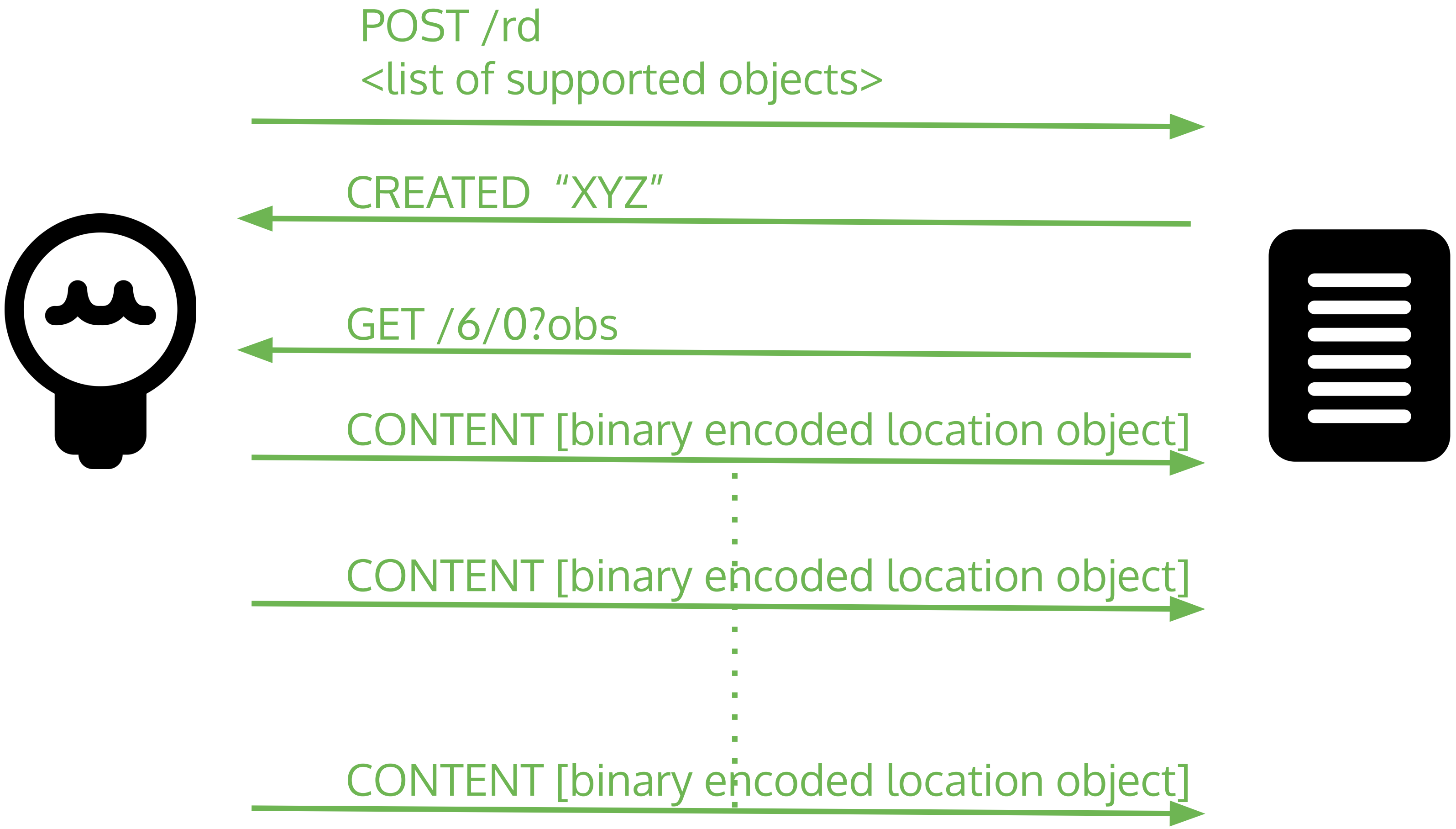
https://github.com/OpenMobileAlliance/lwm2m-registry/



```
                <Item ID="0">
                        <Name>Package</Name>
<Operations>W</Operations>
<MultipleInstances>Single</MultipleInstances>
                        <Mandatory>Mandatory</Mandatory>
                        <Type>Opaque</Type>
                        <RangeEnumeration></RangeEnumeration>
                        <Units></Units>
                        <Description><![CDATA[Firmware package]]></Description>
                </Item>
```

Telemetry?

Observation: A GET where the response is sent again if the value change



POST /rd
<list of supported objects>

CREATED “XYZ”

GET /6/0?obs

CONTENT [binary encoded location object]

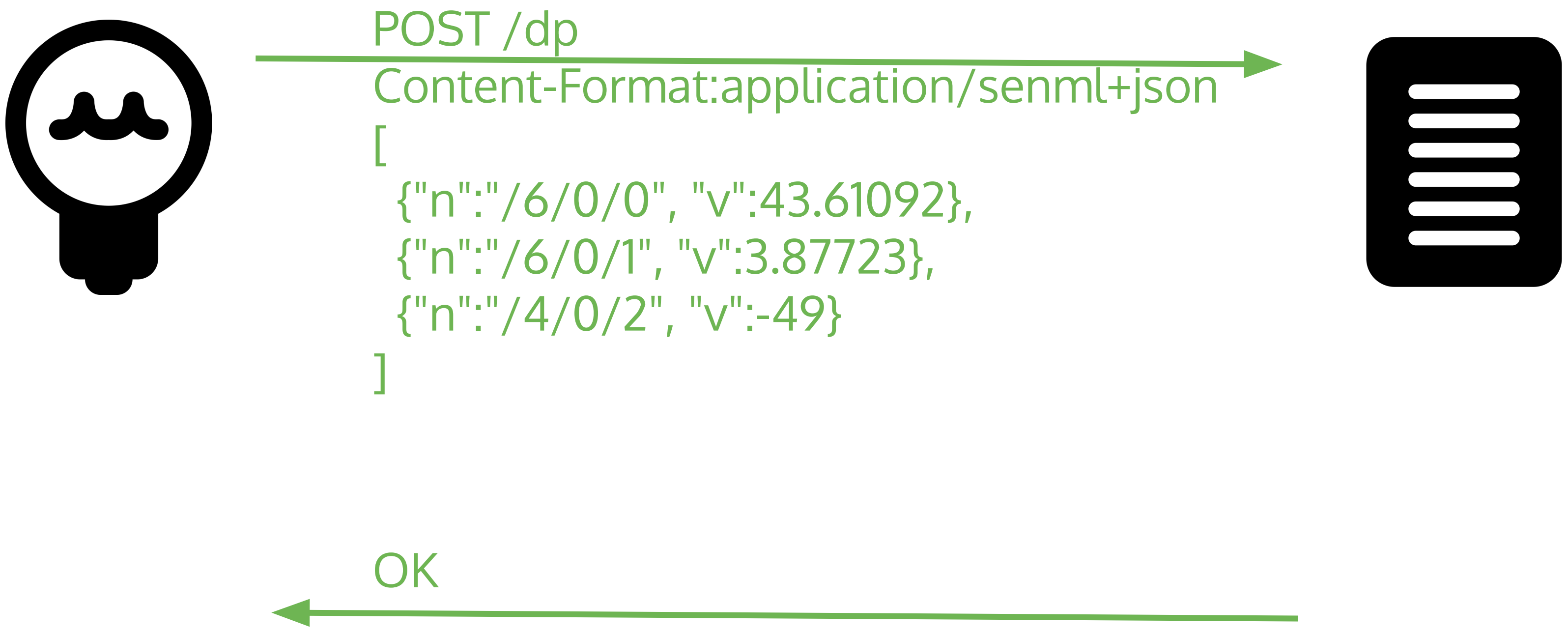
CONTENT [binary encoded location object]

CONTENT [binary encoded location object]

Zephyr® Project
Developer Summit

Telemetry?

Send Operation: A POST on /dp by the client



```
POST /dp
Content-Format:application/senml+json
[
  {"n":"/6/0/0", "v":43.61092},
  {"n":"/6/0/1", "v":3.87723},
  {"n":"/4/0/2", "v":-49}
]
```

OK

Eclipse Leshan

Eclipse Leshan

Java library for implementing servers & clients

Friendly for any Java developer

Simple (no framework, few dependencies)

But also a Web UI for discovering and testing

Build using “mvn install”

Eclipse Californium based (CoAP/DTLS1.2)

Eclipse Leshan

A public IPv4/IPv6 sandbox: https://leshan.eclipseprojects.io

Bleeding edge: deployed on master commit

Leshan v1 is now stable (LWM2M 1.0)

Leshan v2 is under development (LWM2M 1.0 & 1.1.x)

https://github.com/eclipse/leshan/wiki/Roadmap

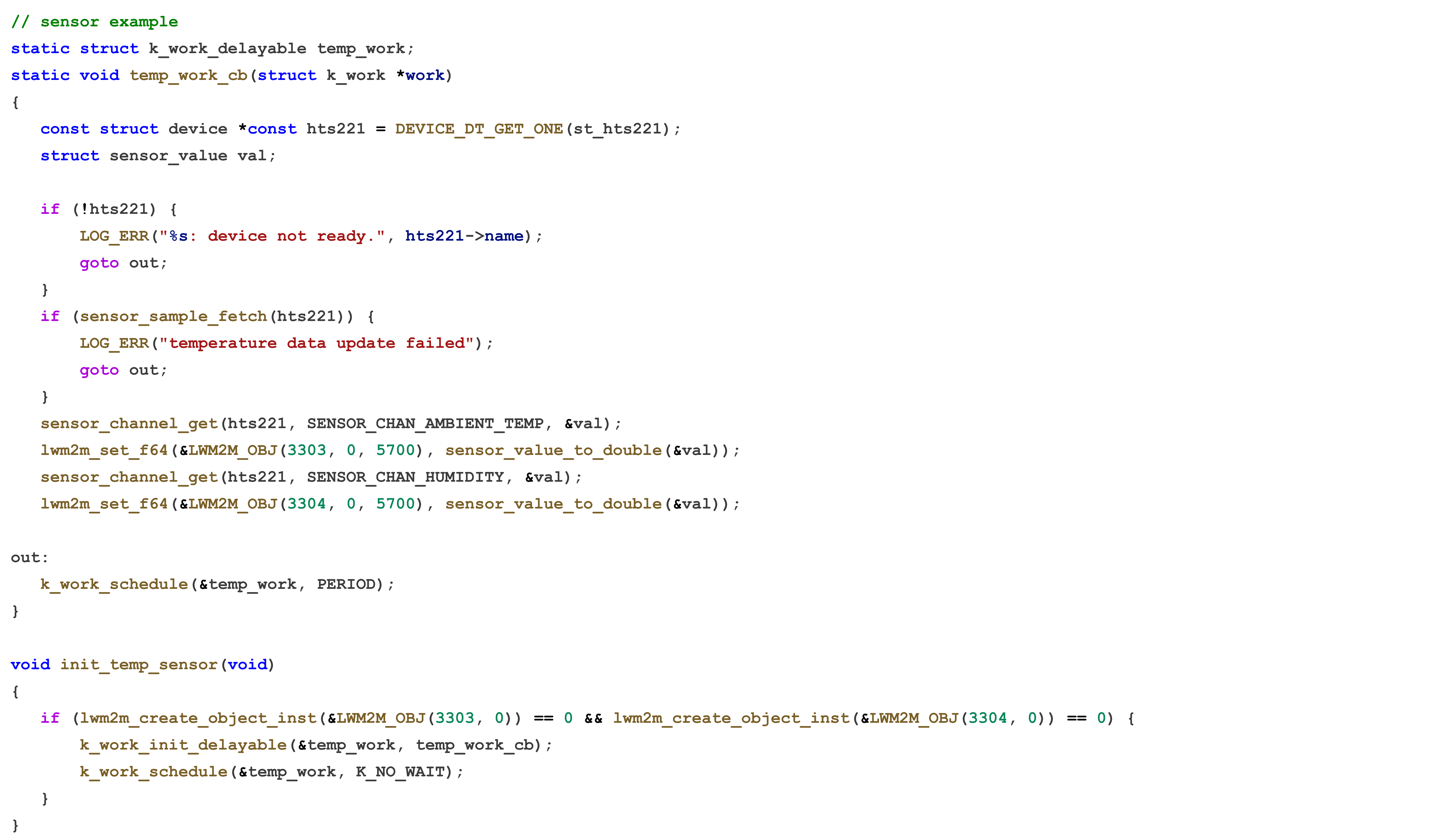
Leshan Server Sample

```
// Prepare LWM2M server
var builder = new LeshanServerBuilder();

// create an Eclipse Californium based endpoint
var cfBuilder = new CaliforniumServerEndpointsProvider.Builder(
          new CoapServerProtocolProvider(), new CoapsServerProtocolProvider());
cfBuilder.addEndpoint("coap://0.0.0.0:5683");
cfBuilder.addEndpoint("coaps://0.0.0.0:5684");

builder.setEndpointsProvider(cfBuilder.build());

// load the default object models (from the standard definition)
var models = ObjectLoader.loadAllDefault();

// add your own custom ones
models.addAll(ObjectLoader.loadDdfResources("/models/", new String[] { "27623.xml", "27739.xml" }));

// Create leshan server and start waiting for client connections
var server = builder.build();
server.start();
```

Zephyr® Project
Developer Summit

Zephyr’s LWM2M client

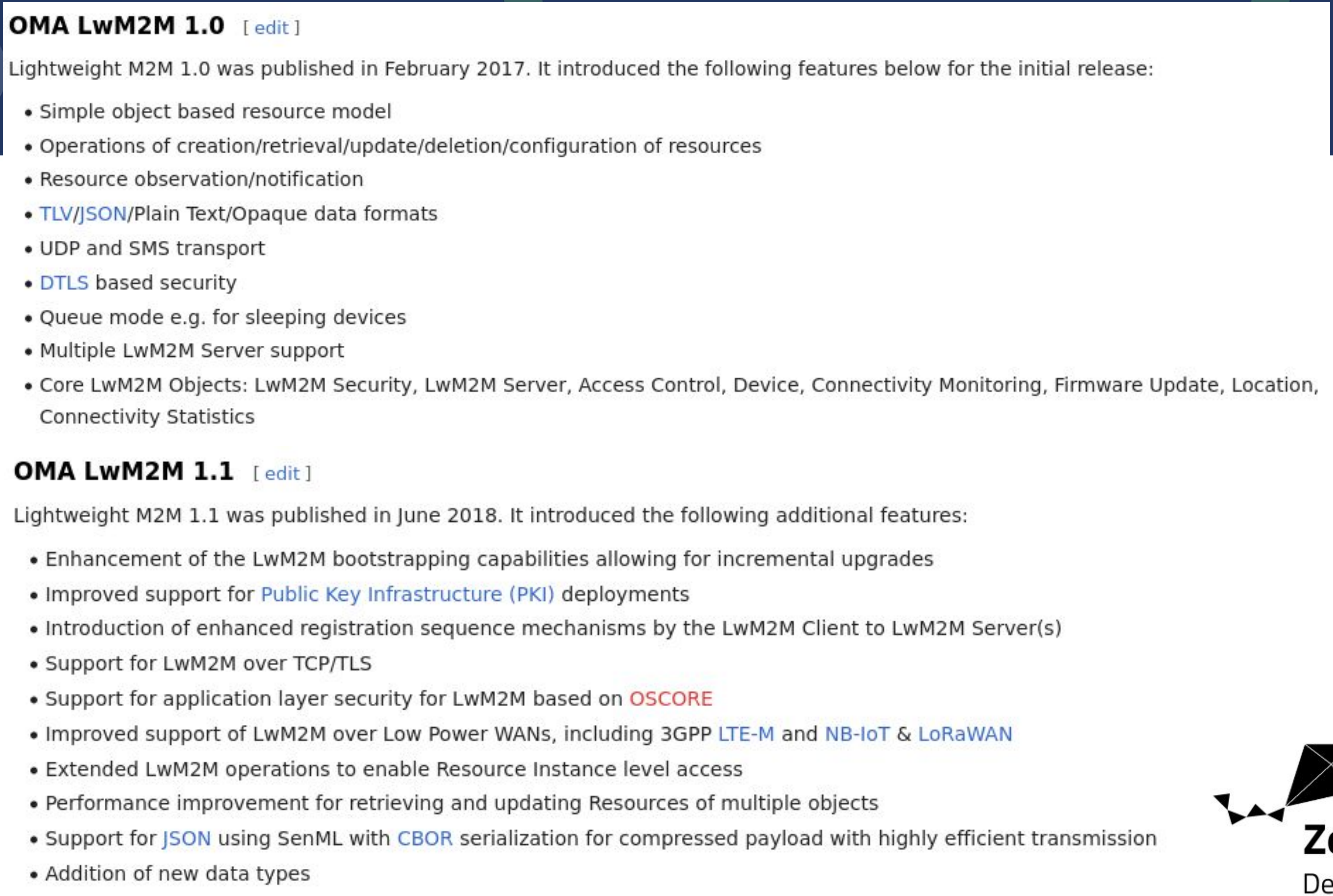
```
// set base object values
lwm2m_set_string(&LWM2M_OBJ(0, 0, 0), "coaps://myserver:5684");
lwm2m_set_u8(&LWM2M_OBJ(0, 0, 2), 0 /* PSK MODE */);
lwm2m_set_string(&LWM2M_OBJ(0, 0, 3), "my PSK identity");
lwm2m_set_opaque(&LWM2M_OBJ(0, 0, 5), (void *)psk, psk_len);
lwm2m_set_string(&LWM2M_OBJ(3, 0, 0), "Acme Corp.");
   ...


lwm2m_rd_client_start(&client, ENDPOINT, 0, rd_client_event, observe_cb);
```



```c
// sensor example
static struct k_work_delayable temp_work;
static void temp_work_cb(struct k_work *work)
{
   const struct device *const hts221 = DEVICE_DT_GET_ONE(st_hts221);
   struct sensor_value val;

   if (!hts221) {
       LOG_ERR("%s: device not ready.", hts221->name);
       goto out;
   }
   if (sensor_sample_fetch(hts221)) {
       LOG_ERR("temperature data update failed");
       goto out;
   }
   sensor_channel_get(hts221, SENSOR_CHAN_AMBIENT_TEMP, &val);
   lwm2m_set_f64(&LWM2M_OBJ(3303, 0, 5700), sensor_value_to_double(&val));
   sensor_channel_get(hts221, SENSOR_CHAN_HUMIDITY, &val);
   lwm2m_set_f64(&LWM2M_OBJ(3304, 0, 5700), sensor_value_to_double(&val));

out:
   k_work_schedule(&temp_work, PERIOD);
}

void init_temp_sensor(void)
{
   if (lwm2m_create_object_inst(&LWM2M_OBJ(3303, 0)) == 0 && lwm2m_create_object_inst(&LWM2M_OBJ(3304, 0)) == 0) {
       k_work_init_delayable(&temp_work, temp_work_cb);
       k_work_schedule(&temp_work, K_NO_WAIT);
   }
}
```

LWM2M Up & Down

Lightweight
!=
Simple



OMA LwM2M 1.0 [edit]

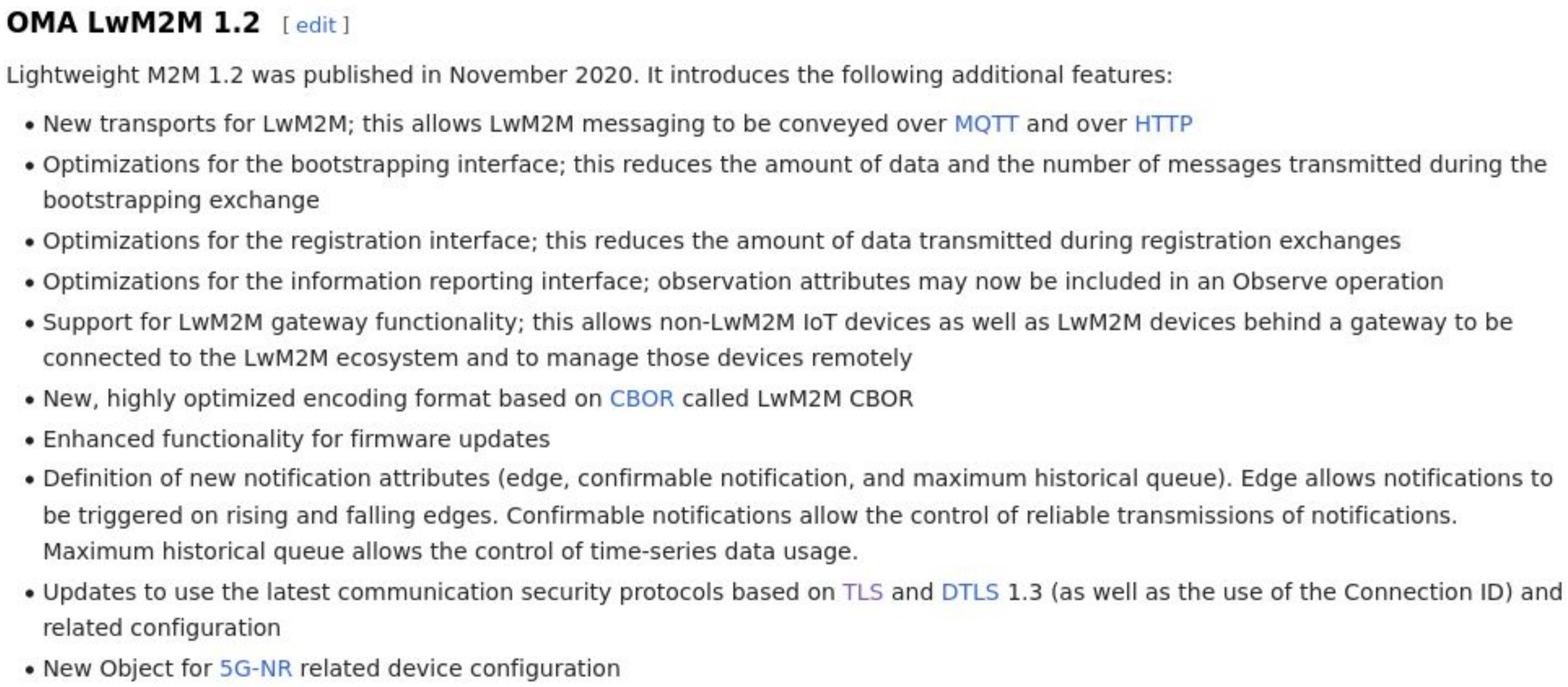
Lightweight M2M 1.0 was published in February 2017. It introduced the following features below for the initial release:

- Simple object based resource model
- Operations of creation/retrieval/update/deletion/configuration of resources
- Resource observation/notification
- TLV/JSON/Plain Text/Opaque data formats
- UDP and SMS transport
- DTLS based security
- Queue mode e.g. for sleeping devices
- Multiple LwM2M Server support
- Core LwM2M Objects: LwM2M Security, LwM2M Server, Access Control, Device, Connectivity Monitoring, Firmware Update, Location, Connectivity Statistics

OMA LwM2M 1.1 [edit]

Lightweight M2M 1.1 was published in June 2018. It introduced the following additional features:

- Enhancement of the LwM2M bootstrapping capabilities allowing for incremental upgrades
- Improved support for Public Key Infrastructure (PKI) deployments
- Introduction of enhanced registration sequence mechanisms by the LwM2M Client to LwM2M Server(s)
- Support for LwM2M over TCP/TLS
- Support for application layer security for LwM2M based on OSCORE
- Improved support of LwM2M over Low Power WANs, including 3GPP LTE-M and NB-IoT & LoRaWAN
- Extended LwM2M operations to enable Resource Instance level access
- Performance improvement for retrieving and updating Resources of multiple objects
- Support for JSON using SenML with CBOR serialization for compressed payload with highly efficient transmission
- Addition of new data types

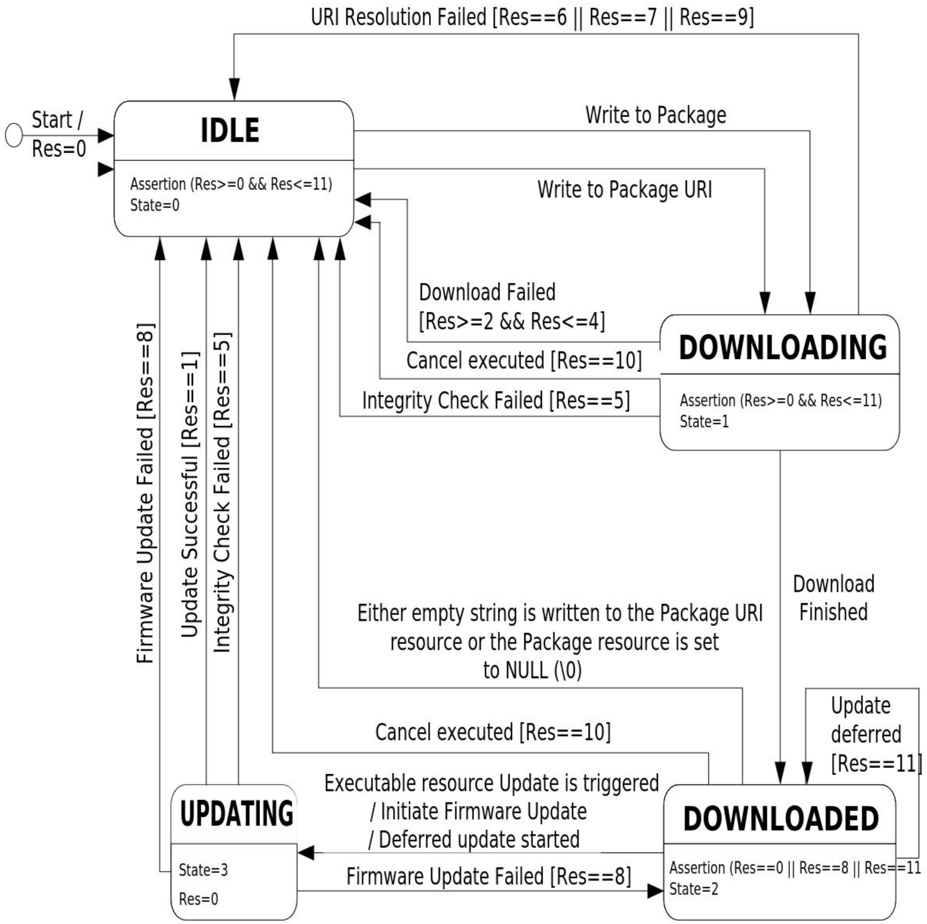


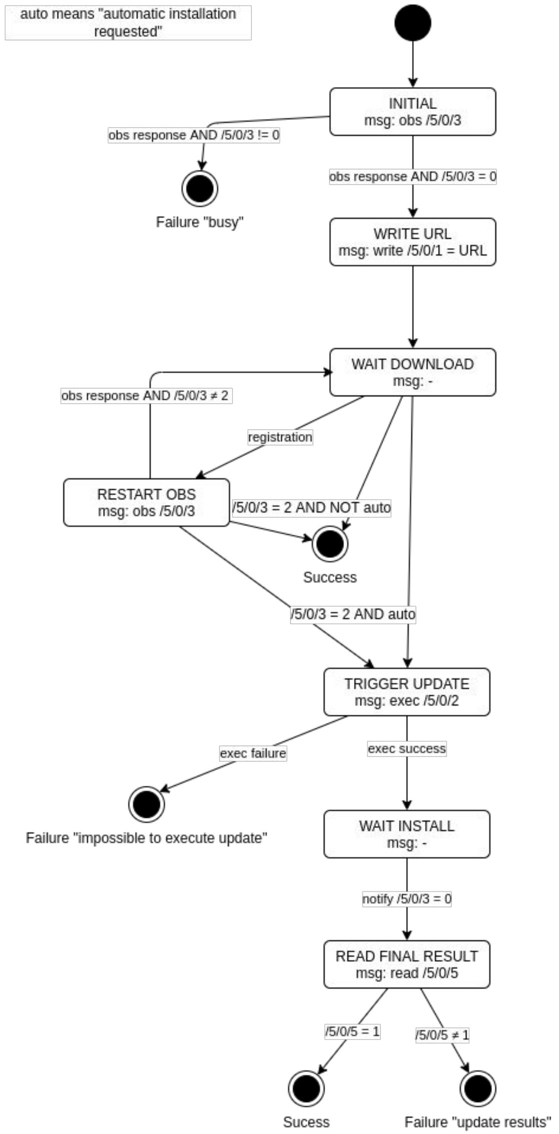
OMA LwM2M 1.2 [edit]

Lightweight M2M 1.2 was published in November 2020. It introduces the following additional features:

- New transports for LwM2M; this allows LwM2M messaging to be conveyed over MQTT and over HTTP
- Optimizations for the bootstrapping interface; this reduces the amount of data and the number of messages transmitted during the bootstrapping exchange
- Optimizations for the registration interface; this reduces the amount of data transmitted during registration exchanges
- Optimizations for the information reporting interface; observation attributes may now be included in an Observe operation
- Support for LwM2M gateway functionality; this allows non-LwM2M IoT devices as well as LwM2M devices behind a gateway to be connected to the LwM2M ecosystem and to manage those devices remotely
- New, highly optimized encoding format based on CBOR called LwM2M CBOR
- Enhanced functionality for firmware updates
- Definition of new notification attributes (edge, confirmable notification, and maximum historical queue). Edge allows notifications to be triggered on rising and falling edges. Confirmable notifications allow the control of reliable transmissions of notifications. Maximum historical queue allows the control of time-series data usage.
- Updates to use the latest communication security protocols based on TLS and DTLS 1.3 (as well as the use of the Connection ID) and related configuration
- New Object for 5G-NR related device configuration

Firmware updates


URI Resolution Failed [Res==6 || Res==7 || Res==9]
Start / Res=0
IDLE
Assertion (Res>=0 && Res<=11)
State=0
Write to Package
Write to Package URI
Download Failed [Res>=2 && Res<=4]
Cancel executed [Res==10]
Integrity Check Failed [Res==5]
DOWNLOADING
Assertion (Res>=0 && Res<=11)
State=1
Firmware Update Failed [Res==8]
Update Successful [Res==1]
Integrity Check Failed [Res==5]
Either empty string is written to the Package URI resource or the Package resource is set to NULL (\0)
Download Finished
Update deferred [Res==11]
Cancel executed [Res==10]
Executable resource Update is triggered / Initiate Firmware Update / Deferred update started
Firmware Update Failed [Res==8]
UPDATING
State=3
Res=0
DOWNLOADED
Assertion (Res==0 || Res==8 || Res==11
State=2



Software updates (object9)

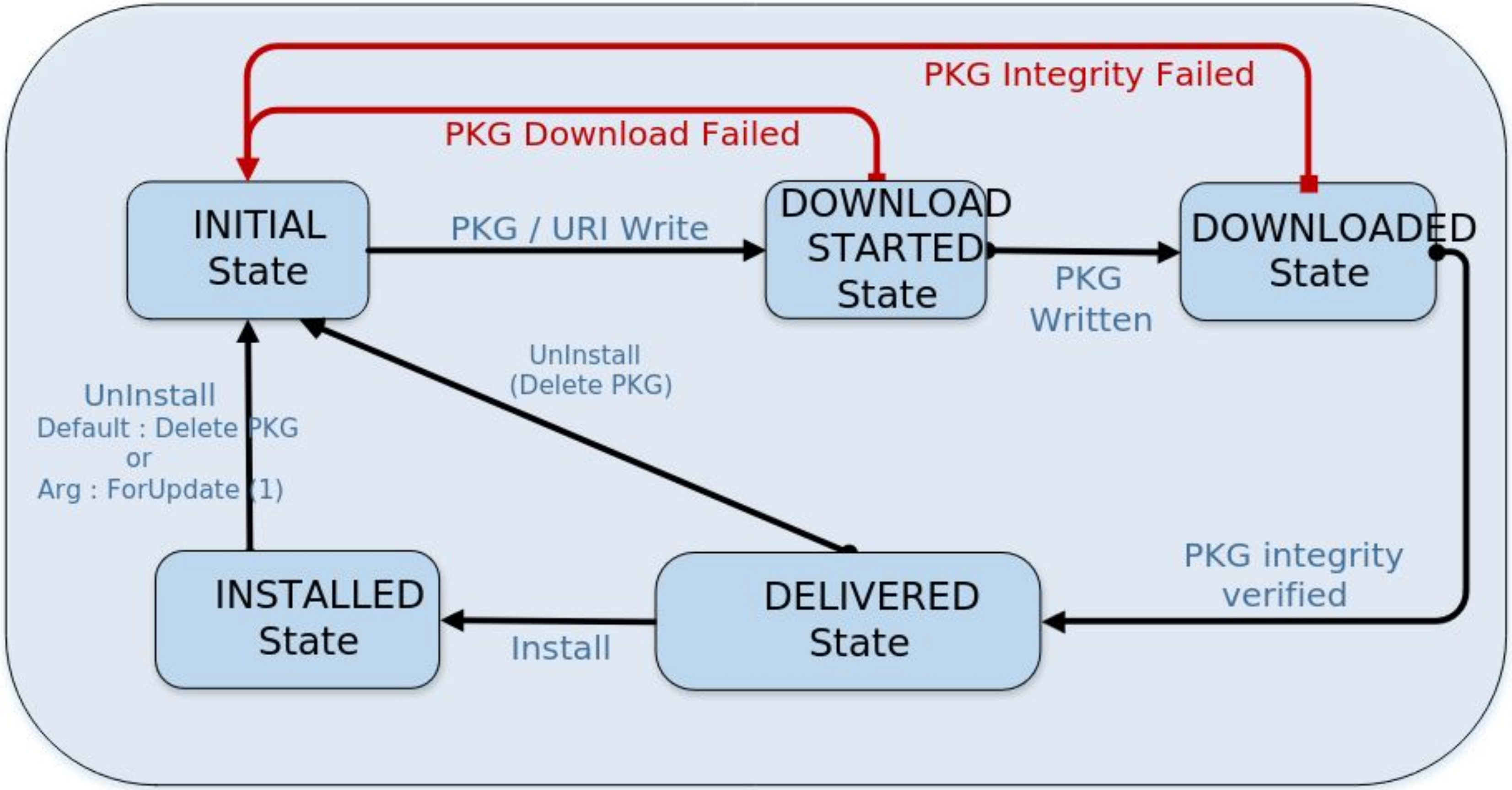
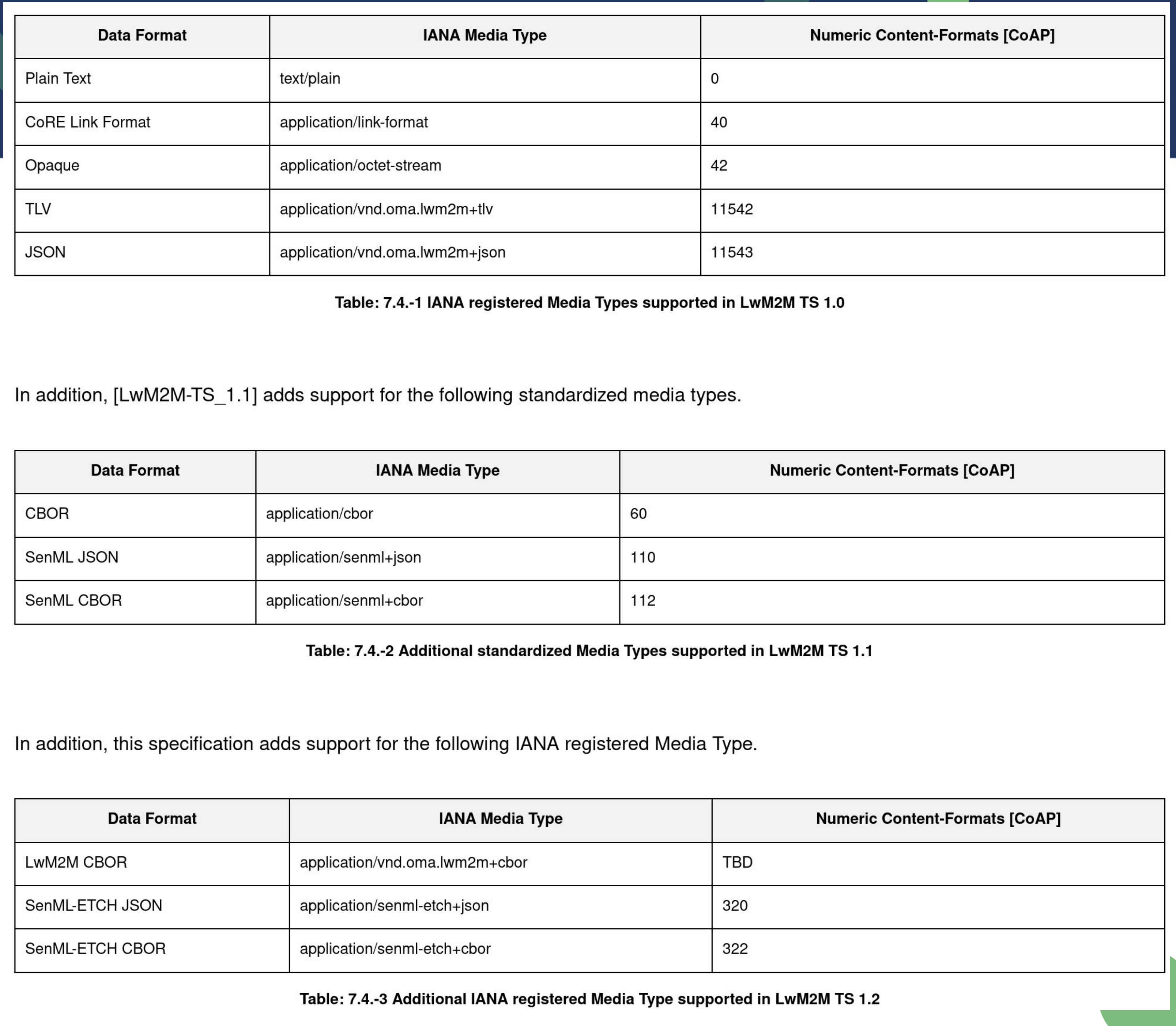


Figure: 5.1.1.-1 Package Installation State Machine



Data Format	IANA Media Type	Numeric Content-Formats [CoAP]
Plain Text	text/plain	0
CoRE Link Format	application/link-format	40
Opaque	application/octet-stream	42
TLV	application/vnd.oma.lwm2m+tlv	11542
JSON	application/vnd.oma.lwm2m+json	11543

Table: 7.4.-1 IANA registered Media Types supported in LwM2M TS 1.0

In addition, [LwM2M-TS_1.1] adds support for the following standardized media types.

Data Format	IANA Media Type	Numeric Content-Formats [CoAP]
CBOR	application/cbor	60
SenML JSON	application/senml+json	110
SenML CBOR	application/senml+cbor	112

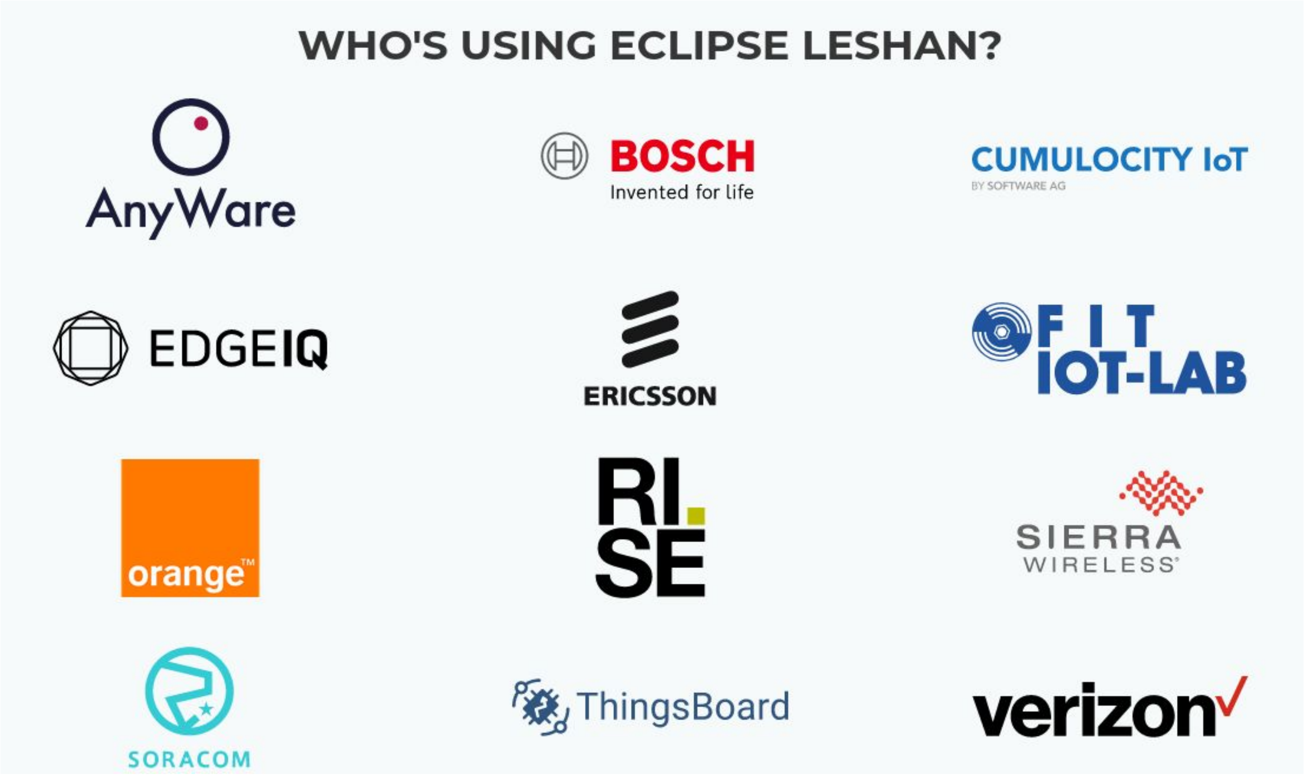
Table: 7.4.-2 Additional standardized Media Types supported in LwM2M TS 1.1

In addition, this specification adds support for the following IANA registered Media Type.

Data Format	IANA Media Type	Numeric Content-Formats [CoAP]
LwM2M CBOR	application/vnd.oma.lwm2m+cbor	TBD
SenML-ETCH JSON	application/senml-etch+json	320
SenML-ETCH CBOR	application/senml-etch+cbor	322

Table: 7.4.-3 Additional IANA registered Media Type supported in LwM2M TS 1.2

Adoption & Scaling?



WHO'S USING ECLIPSE LESHAN?

Omnipresent in Cellular IoT:

- mandated for carrier certifications
- multiple vendors for scaling

Multiple millions of devices deployed

Zephyr® Project
Developer Summit

Questions & Answers

Thank you!

https://www.linkedin.com/in/jvermillard