Lightweight Machine to Machine Requirements

Approved Version 1.1 – 10 Jul 2018

Open Mobile Alliance

OMA-RD-LightweightM2M-V1_1-20180710-A

Contents

Figures

Tables

1. Scope (Informative)

This document represents 1_1 requirements consolidated.

2. References

2.1 Normative References

2.2 Informative References

3. Terminology and Conventions

3.1 Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

All sections and appendixes, except “Scope” and “Introduction”, are normative, unless they are explicitly indicated to be informative.

3.2 Definitions

3.3 Abbreviations

CBOR	Compact Binary Object
OMA	Open Mobile Alliance
SenML	Sensor Markup Language

4. Introduction (Informative)

4.1 Version 1.0

4.2 Version 1.1

The following are new features are introduced:

1) Improved ability to use LwM2M for device management of low power wide area network devices like 3GPP CIoT, LoRA.
2) Alignment with better current security practices by using the recommendations from RFC 7925 and better performance with new TLS/DTLS extensions provided thereof.
3) TCP and TLS to enable better support for CoAP in environments with firewalls and other middleboxes. The use of TCP also lowers the need for frequent keepalives messages in the presence of network address translators.
4) Improved maintenance during firmware upgrade, bootstrapping and registration procedures.

5. LwM2M v1_1 (Informative)

5.1 End-to-end Service Description

LwM2M v1_0 implementation has brought us several new dimensions to get towards LwM2M v1_1, the following are enhanced solutions in the new release:

5.1.1 LwM2M core functionality

5.1.1.1 Transports

- The Constrained Application Protocol (CoAP) was designed for Internet of Things deployments, assuming that UDP or Datagram Transport Layer Security (DTLS) over UDP can be used unimpeded. UDP is a good choice for transferring small amounts of data across networks that follow the IP architecture and where no firewalls block UDP-based traffic.
- Some LwM2M deployments need to integrate well with existing enterprise infrastructures, where UDP-based protocols may not be well-received or may even be blocked by firewalls. Middleboxes that are unaware of CoAP usage can make the use of UDP brittle, resulting in lost or malformed packets.

5.1.1.2 Bootstrap and Registration

- [LwM2M-BSR-2] Bootstrap Extended Capability: in LwM2M 1.0, performing incremental Bootstrap phases – e.g. for adding a new Server and upgrading/adapting Client Configuration – is limited, because the Bootstrap Server is blind regarding the Access Rights already in place in the Client (no way to upgrade the ACL Instances).

  LwM2M 1.1 provides to the Bootstrap Server a way to discover the Access Control Rights already in place in a LwM2M Client, authorizing a safe/consistent upgrade of the Client Configuration (no conflict to solve).
- [LwM2M-BSR-3] In LwM2M 1.0, the "Server Initiated Bootstrap Mode" is not considered as a reliable Mode. Defining a mechanism which allows a Bootstrap Server to trigger the "Client Initiated Bootstrap Mode" in a Client, is a simple and reliable way for providing to a Server, the capability to Initiate a Bootstrap Sequence, while reusing an already proved mechanism.
- Bootstrapping and registration procedures are linked as transitions between the bootstrapping and registration modes must be supported both for normal operation and for error conditions.
- During bootstrapping, the LwM2M client must be able to use the bootstrapping sequence to successfully transition to communications with the configured LwM2M servers defined by the LwM2M server objects.
- During registration with the configured LwM2M servers defined by the LwM2M server objects or other objects, if significant errors occur, the LwM2M client must be able to return to the bootstrapping procedures to correct any configuration errors in the LwM2M server objects.
- Resources defined in the LwM2M server objects or other objects must guide the LwM2M client behaviour for both normal and error conditions.
- A LwM2M client may need to use a specific APN to communicate with a LwM2M server.

5.1.1.3 Device Management & Service Enablement Interface

- [LwM2M-OSR-1] Address extension: while LwM2M multi-Instances Resources are supported in LwM2M TS 1.0, this enabler is only capable to address such a Resource as a whole. In LwM2M 1.1, through Device Management and Service Enablement Interface, the individual Read and Write accesses on a certain Instance of a LwM2M Multi-Instances Resource is supported.
- [LwM2M-DSE-001] & [LwM2M-IR-001] In LwM2M TS 1.0 the Read and Write operations are bound to CoAP Get and Put methods respectively where the scope of operation can be either all instances of an object, a single instance, or a resource within a given instance. If the server requires access to a subset of resources in an instance or across instances of the same or different objects it has to issue multiple requests. For constrained low power devices this present an undesirable overhead as it means the device has to stay up longer over multiple transactions. This should be addressed in LwM2M 1.1; at the transport level where CoAP is concerned the newly defined

FETCH and PATCH methods in RFC 8132 already address this requirement, but the LwM2M layer currently has no way of utilising such capabilities.

- [LwM2M-DSE-003] In LwM2M TS 1.0, any data sent from a LwM2M Device is either the result of a direct READ Operation or the result of an OBSERVE Operation. However, some applications may have dynamic data models that make it impossible for the Server to know in advance which data they should READ or OBSERVE. The only workaround today is to have a Server send an OBSERVE generic enough that a lot of data (among which potentially a lot is not really desired by the Server) is OBSERVE'd and then sent back by the Client. This generates a lot of useless data transmissions both in terms of undesired data and in terms of establishment of the OBSERVE.
- [LwM2M-DSE-004] Allowing LwM2M clients to report unsolicited data to the LwM2M server, as required in LwM2M-DSE-003, could pose a certain risk in some situations, in particular the risk of too many devices reporting too much data to the server. In order to mitigate this risk, the requirement LwM2M-DSE-004 was introduced.

5.1.1.4 Maintenance and Upgrade

5.1.1.4.1 Maintenance

Maintenance and upgrade scenarios in LwM2M v1_1 is essential as the migration from older releases LwM2M v1_0 or LwM2M v1_0_x would be a reality in the field. The intention of these requirements is to provide:

- Clear guideline for devices to perform before, during and after upgrade of the devices
- Effective abilities to setup or request for right configurations and parameters before, during or after upgrade
- The upgrade procedures can also guide the maintenance phase of the devices during restarts and other bad conditions.

5.1.1.4.2 Upgrade

The following usages are needed to be addressed during the upgrade to save the configurations

- Constrained nature of the devices is naturally connecting to a very low bandwidth options pushes the necessity to be efficient to preserve what is not needed to retransmitted after the upgrade
- The nature of the devices in IoT space would like to differentiate the need for device-default configurations which normally are not in the hands of LwM2M server. In the following diagram, the blue/green boxes can be considered as examples which need to survive the upgrade and normally the owner LwM2M server may not possess those configurations at any point of time in the lifecycle of the device. It is expected these configurations are loaded into the device through pre-agreed methods. These could be requested explicitly to be retained during the upgrade.

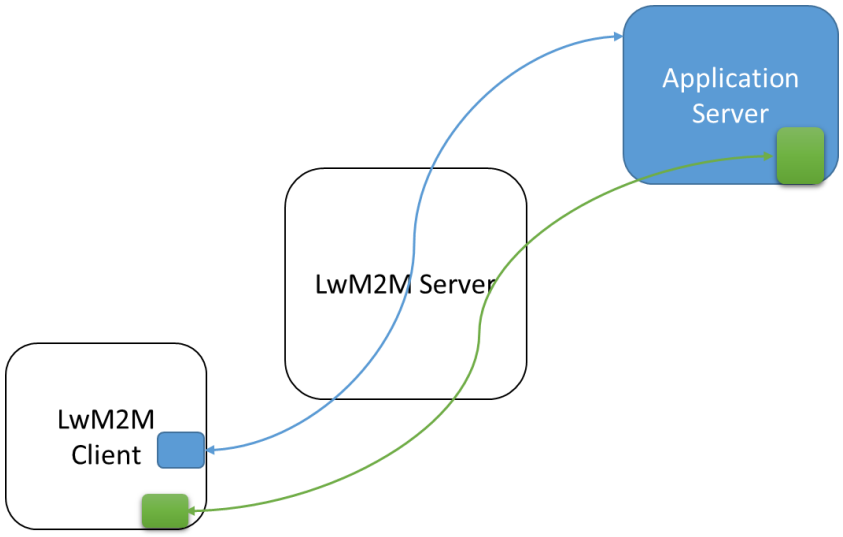

Figure 1 : 3rd party configurations

The following usages are needed to be addressed during the upgrade and create fall back during the upgrade

- The upgrade would essentially due to faulty conditions can lead to looking for a wrong package in the network
- Late realization of a package faults

5.1.1.4.3 Client HoldOff Timer

The client hold off timer ensures that the client registers with that LwM2M server after certain time duration. This is to ensure the relevant systems in the backend areas (OSS systems), gets ready for coordinating with the device as soon as the device registers. When supporting multiple LwM2M server instances it is sometime necessary to have such a configurable timer (seconds) with which the client waits after being bootstrapped to register the FIRST time with each to the server instances. This provides an opportunity for a provisioning system to learn of the characteristics of the devices across the backend of the servers in order to have proper server instances prior to client registering.

The following picture provides an overview of the need in this area.

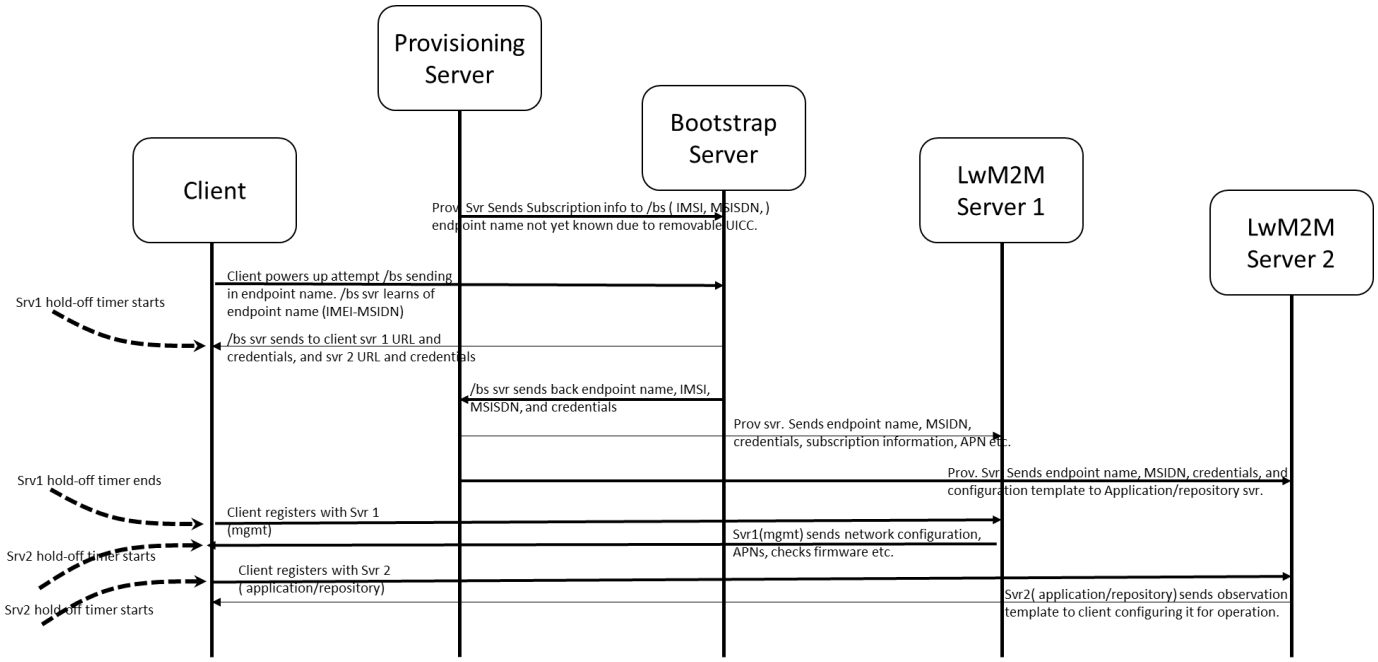

Figure 2 : Client HoldOff Timer

The above example gives a sequential need, but there could be other usage pattern necessary like 2 LwM2M server together getting the initial registration and LwM2M server 3 and 4 comes later in a specific hold-off timings for the LwM2M client.

5.1.1.5 Observation

5.1.1.5.1 Reporting Mode

When reporting notifications of observations, additional information is required to both control the configuration of the reporting and indicating the cause for reporting.

5.1.2 LwM2M Gateway functionality

The LwM2M Legacy Gateway functionality for LwM2M v1.1 aims to interconnect different IoT islands, as shown in Figure 3. The LwM2M Legacy Gateway in the center consists of several components, namely

- A LwM2M client,
- a protocol translator, and
- a data model translator.

The LwM2M Client uses the LwM2M protocol to interact with the Bootstrap Server as well as with one or multiple LwM2M Server(s). It is responsible for representing the non-LwM2M devices (also referred as legacy devices) to the LwM2M Server(s).

The protocol translator is responsible for converting LwM2M protocol messages from and to the protocol on the other side of the gateway, such as BLE. The details of this protocol translation are outside the scope of the LwM2M v1.1 specification and implementation dependent. As an example, a RESTful LwM2M request has to be translated to a BLE query that uses an RPC-alike mechanism, which requires state to be maintained at the gateway. Similarly, the response message from the BLE device has to be translated to the corresponding LwM2M response message.

The data model translator is responsible for translating the LwM2M data model into corresponding representations at the legacy protocol side. For example, BLE uses services and characteristics, which conceptually map to LwM2M objects and resources. Of course, it is only possible to map the data model in a lossless fashion if the corresponding abstractions exist in both data models. For example, there is no LwM2M object currently defined that represents the Heart Rate service defined by the Bluetooth SIG. In such a case, the use of the LwM2M BinaryAppDataContainer object may be a possible mechanism to tunnel data to the LwM2M server without the need to perform translation at the legacy gateway.

There are two design approaches to provide the necessary functionality, which is described below. We refer to them as single instance vs. multiple instance approach.

Design Approach #1: Single LwM2M Client Instance

In this approach the LwM2M server's view is that there is only a single LwM2M Client instance running on the LwM2M Legacy Gateway. To retrieve objects and resources from attached legacy devices connected to the LwM2M Legacy Gateway it represents those in the LwM2M data model.

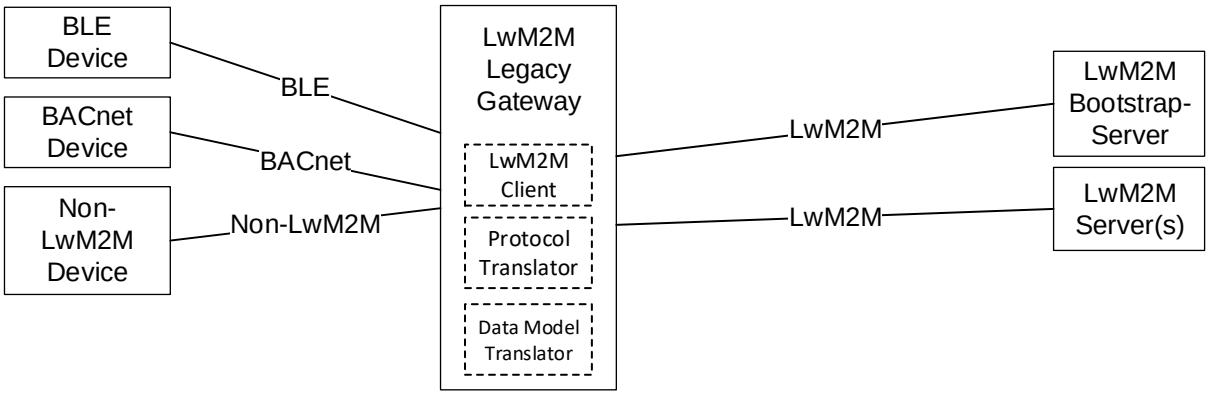

Figure 3 : Legacy Gateway Architecture for the Single LwM2M Client Instance

Design Approach #2: Multiple LwM2M Client Instances

In this approach the LwM2M server's view is that each of the legacy devices is represented by an independent instance of an LwM2M client running on the LwM2M Legacy Gateway. Additionally, a separate LwM2M client instance is running on the gateway that represents the LwM2M Legacy Gateway itself. Figure 4 shows this design graphically.

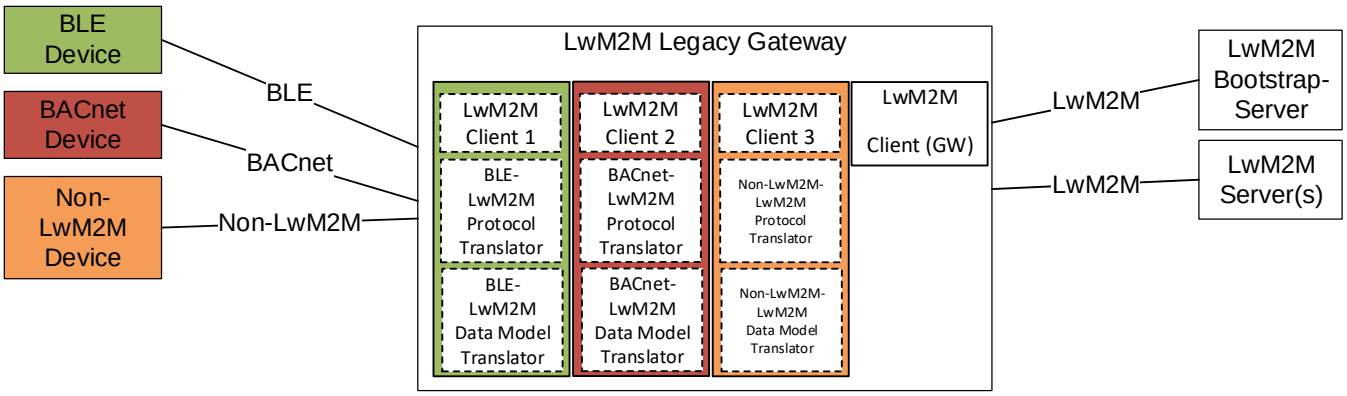

Figure 4 : Legacy Gateway Architecture for the Multiple LwM2M Client Instance

Which model is most appropriate depends on the number of connected legacy devices and other programming model differences. For example, mobile phone apps are typically developed by different parties and include libraries rather than having to rely on an operating system provided shared library. In such a case the single LwM2M client instance is easier to deploy. For a standalone home gateway that connects many IoT devices the multiple LwM2M client instance approach is better.

Use Case #1: Health Care / Well Being

A Bluetooth Low Energy (BLE) heart rate sensor is connected to a smart phone. This BLE heart rate sensor may use the heart rate profile and service defined by the Bluetooth SIG or may make use of a proprietary profile and service. Additionally, the BLE device may support other (potentially proprietary) profiles and services, such as support for firmware updates. Since the data obtained via this heart rate is not only used by an app on the smart phone but also communicated to a cloud-based server for big data analysis. Additionally, device management functionality (like firmware updates) is provided by the manufacturer.

The smart phone therefore includes a translation component to communicate with the BLE heart rate device on one side and with the LwM2M server infrastructure on the other side.

Use Case #2: Commercial Indoor lighting

In this case a stand-alone LwM2M Legacy Gateway is used to translate between a BACnet building automation network on one side of the gateway and to an LwM2M environment on the other side. Different to use case #1 is that this gateway is likely going to connect a much larger number of devices and needs to be managed itself since this gateway will be similar to a networking appliance without any direct user interaction.

5.1.2.1 Groups & Topology

The group concepts does not exists in LwM2M v1_0_x, this would be interesting to establish group of devices at Gateway level for the server to address them in one go, typically a group of BLE devices attached to the gateway say for a particular patient, group of lights in area/floor etc., it is a useful concept for the gateway helping to address those groups through single commands from LwM2M server towards the legacy gateway.

The location of a LwM2M client typically can be in a sensor at different floors or rooms differentiated by functionalities like Kitchen, Lift room etc., It would be good that the legacy gateway could support these topological views, further to understand the nature of the location where the LwM2M Client is residing.

5.1.3 Security Enhancement

5.1.3.1 Extended PKI Support

LwM2M version 1.0 provides support for bootstrapping using certificates but suffers from a few limitations:

- no recommendations are offered for certificate revocation,
- only pinned certificates are supported, and
- no recommendations for obtaining secure time information are available.

For a functioning and practical deployment of a certificate-based authentication infrastructure it is essential to offer recommendations for certificate revocation, ways to obtain time information securely. Additionally, while the use of pinned certificates is convenient and secure it is not the most common way to deploy a public key infrastructure since scalability suffers. This also enables new use cases where customers can use their already deployed CA infrastructure for use with LwM2M.

5.1.3.2 TLS/DTLS Guidance

LwM2M version 1.0 provides guidance for how to use DTLS to offer communication security for CoAP over various transports.

5.1.3.3 Secure Component Support

- LwM2M 1.1 will take benefit of supporting a flexible framework based on Secure Component (i.e. Secure Element-SE or Trusted Execution Environment-TEE) to extend LwM2M Security capability
- This Framework will be composed of:

- a Secure Component containing sensitive information (e.g Key Storage) as well as sensitive algorithms (e.g. on-board key pairs generation ...) running in it
- a LwM2M 1.1 Core Object dedicated to manage the Secure Component and the services which can be deployed (e.g. eUICC-M2M profile capability, Device Secure Boot, LwM2M 1.0 Smartcard Bootstrap)
- standardized protocols having to be established between the LwM2M Secure Component Admin Object and the Secure Component

5.1.3.4 E2E security

The purpose of end-to-end security is to protect communication between endpoints against attacks launched from on-path attackers. In order to define end-to-end security, the endpoints need to be specified. Different Use Cases will require different endpoints to be considered and different protocols can be supported by the endpoints.

In order to formulate non-trivial requirements some specific settings must be considered, containing assumptions regarding a) what are the endpoints, and/or b) what application layer protocol(s) (CoAP, HTTP, etc.) are being used between the endpoints. More information about the settings is provided in separate sections for each setting.

Endpoints for a few scenarios are listed below:

- LwM2M Server - LwM2M Client
- Application Server - LwM2M client
- LwM2M Server - non-LwM2M IoT Device

5.1.3.4.1 General E2E security requirements for LwM2M Endpoints

Currently, LwM2M V1.0 security requirements are essentially defined in terms of DTLS. For LwM2M V1.1 the requirements should explicitly state the security properties needed for E2E security –

Integrity protection

An on-path adversary may change the operation or response, e.g. from Read to Delete, which object, instance or resource the operation applies to, attributes, the payload of the message, the error status (from Failure to Success), error code, etc. An on-path attacker may also remove or inject information. To prevent from manipulation, the operations and responses over LwM2M interfaces must be integrity protected end-to-end.

Encryption

An on-path adversary may eavesdrop on the communication and learn about the content or nature of the operation. For confidentiality and privacy, the communication over LwM2M interfaces needs to be encrypted end-to-end.

Replay protection

An on-path adversary may record an operation and later play back the operation, e.g. resetting an old key or reconfiguring an object instance with an old value. The operations over LwM2M interfaces must be replay protected end-to-end.

Binding response to operation

An on-path adversary may record and block a response to one operation sent from a LwM2M Server, and later block a second operation and send back the response of the first operation, giving the Server wrong information of the result of the operation. For an example, see Figure 5 of [2].

The end-to-end security solution must bind the response to the operation.

Freshness

An on-path adversary may delay an operation and later deliver the operation at a selected occasion, giving the LwM2M Client the impression that the LwM2M Server recently sent the operation. The LwM2M Client must be able to verify the end-to-end freshness of certain operations. For an example, see Figure 3 of [2].

One general assumption is that the underlying binding protocol may be different on the path between the endpoints, for example on the path between LwM2M Client and LwM2M Server, and may include reliable transport such as TCP, unreliable and unordered transport like UDP, and other protocols including SMS and NB-IoT, see Figure 5.

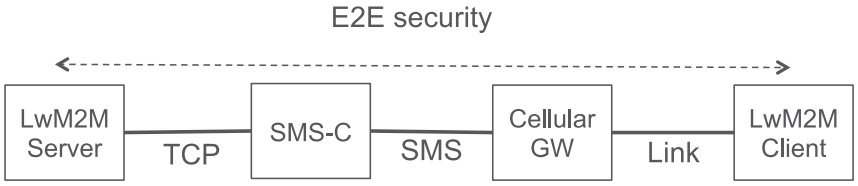

Figure 5 : Example of E2E security of LwM2M Server and LwM2M Client over varying transport

Scenario 1: LwM2M Server and LwM2M Client with Intermediary Nodes in LwM2M V1.0

The communication between LwM2M Server and Client is as of V1.0 based on the application layer protocol CoAP. Different application layer protocols may be used in future versions of LwM2M, but the E2E security solution must in particular protect LwM2M operations using CoAP end-to-end.

In LwM2M v1.0, DTLS support is limited to scenarios where intermediary nodes do NOT exist between the LwM2M Server and LwM2M Client. Since SMS is supported as a Transport Binding for LwM2M 1.0, there are several security threats that can emerge due to this shortcoming.

The deployment setting may involve intermediate nodes (e.g. proxies, SMS-C, cellular gateways) which are not necessarily trusted by the endpoints and from which adversaries can launch attacks. The operations performed over the LwM2M interfaces must be protected end-to-end between LwM2M Server and LwM2M Client through intermediate nodes such that the operations and responses are preserved, see Figure 6.

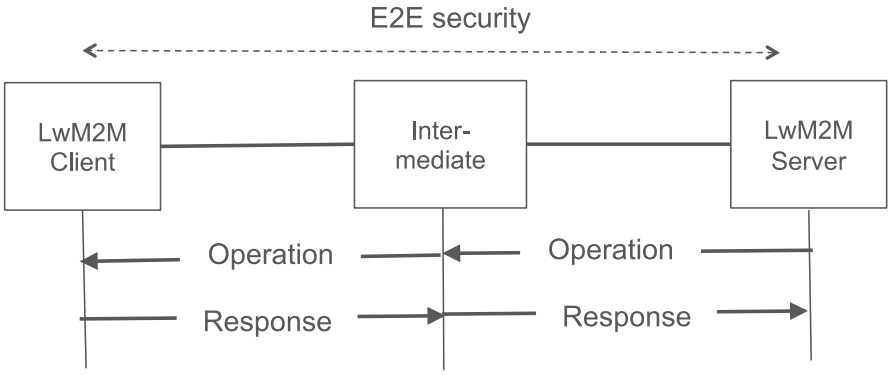

Figure 6 : E2E security of operation and response

The intermediate node may be a LwM2M aware or LwM2M unaware. The e2e security requirements described previously apply irrespective of whether the intermediate node is aware or unaware of LwM2M. In the case of a LwM2M aware intermediary node, such as the LwM2M gateway, additional requirements may be put on the intermediary.

Figure 7 : E2E Security with Intermediate Nodes

Note that in settings where an intermediate node translates between e.g. MQTT on one side and a RESTful protocol on the other side, the mapping needs to be faithful to the LwM2M operations and comply with the e2e security requirements previously described. New security solutions have to be explored to address these scenarios.

Scenario 2: E2E security between Application Server and LwM2M Client

Applications may require e2e security between a LwM2M node and a non-LwM2M node. This scenario addresses the use case where an Application in a LwM2M Client needs to enforce secure operations requested by an Application Server using LwM2M operations, but where the LwM2M Server is not trusted to read or modify the operation or response. For example, the LwM2M server may not be trusted to retrieve location information about a LwM2M client -- only the Application should be able to request and read location information. One rationale for this use case is separation of concerns, where the LwM2M Server is hosted by a partner which should not have access to certain resources at the LwM2M Client.

Nodes between Application Server and LwM2M Client may support a combination of protocols, e.g. in a common setting the communication between Application Server and LwM2M Server is HTTP, see Figure 8.

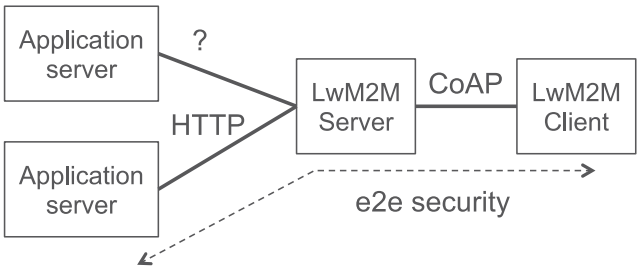

Figure 8 : Example of E2E Security of LwM2M Server and LwM2M Client over varying transport

The ? refers to protocols other than HTTP that can be used to communicate with the Application Servers. The application server can be accessed using API based on MQTT, AMQP, XMPP etc.

If end-to-end security of the communication between Application Server and LwM2M Client is needed (e.g. in the case of securely requesting application layer data which should not be accessible to the LwM2M Server) then all general E2E security requirements previously described applies also to this case. Note that the endpoint for communication is typically an application running in the LwM2M Client which has access to both the LwM2M API (for LwM2M Client-Server communication) and the Application Server API for exchanging messages with the Application on the Application Server. All general E2E security requirements described previously apply to this scenario.

Scenario 3: Secure Fragmentation

Downloading large amounts of data to a LwM2M client may be used as an attack vector for Denial-of-Service. If the client cannot perform any verification before the entire transfer is completed an on-path attacker can inject data and thereby block/reduce functionality of the client. In order to mitigate this threat, a server must be able to fragment the message in a secure way, such that the client can verify fragments as they are received.

This requirement applies in particular to firmware updates [1]. Note that a signature over the firmware does not solve this problem, since the entire firmware needs to be downloaded before the client can verify the signature. Moreover, in order to perform fragmentation suitable for a certain network, the function of making the fragmentation is typically separate from the code repository. One example setting is shown in Figure 5.

Firmware download may alternatively be performed outside LwM2M.

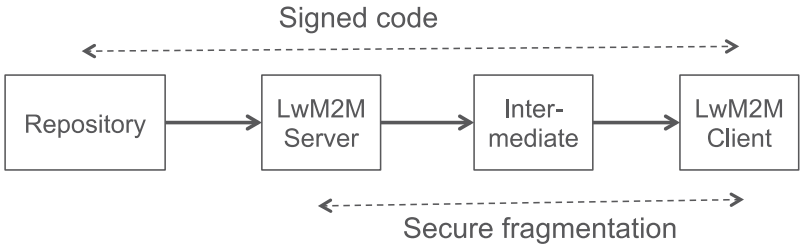

Figure 9 : Example of nodes involved in securing firmware update

Scenario 4: E2E Security between LwM2M Server and IoT Device

This scenario describes another use case of e2e security between a LwM2M node and a non-LwM2M node (compare Scenario 2). In this scenario, the IoT device is a non-LwM2M device enforcing secure operations requested by an LwM2M Server, and a Legacy Gateway which is not trusted to read or modify the operation or response. E.g. it should be possible that the data from the IoT device can be encrypted for the LwM2M server and not visible in the Legacy Gateway, see Figure 10. (The case with LwM2M 1.1 Gateway is already covered in scenario 1.)

One rationale for this use case is if the Legacy Gateway is hosted in an unprotected environment and by being concentrator of multiple IoT Devices would become an attractive target for an attack. All general E2E security requirements described previously applies to this scenario.

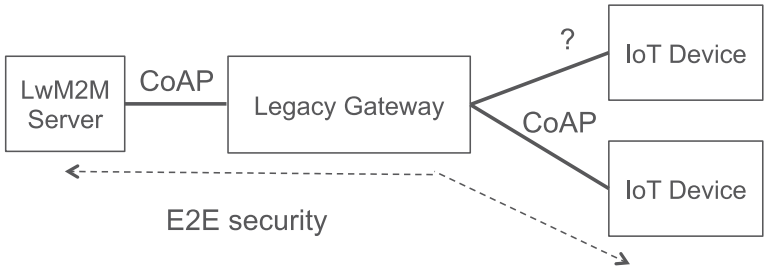

Figure 10 : Example of E2E Security of LwM2M Server and IoT Device

The ? refers to protocols other than CoAP that can be used to communicate with the IoT Device.

5.1.4 Evolution of new LPWAN standards

LwM2M is the management and data plane for the application layer in constrained IoT/M2M devices. The GSMA Mobile IoT project dealing with new 3GPP standardization inputs in the area of cellular LPWAN. The Mobile IoT standards from 3GPP define similar scenarios for the three standardization threads: EC-GSM-IoT, LTE-MTC (Cat-M1) and NB-IoT (Cat-NB1).

NIDD (Non-IP Data Delivery) carries delays due to the inherent nature built to make the device smaller, efficient both in battery operations and network operations. This means NIDD path is for devices needing such possibilities. For real-time critical communications like fire alarm choose IP delivery path with right small retry timeouts.

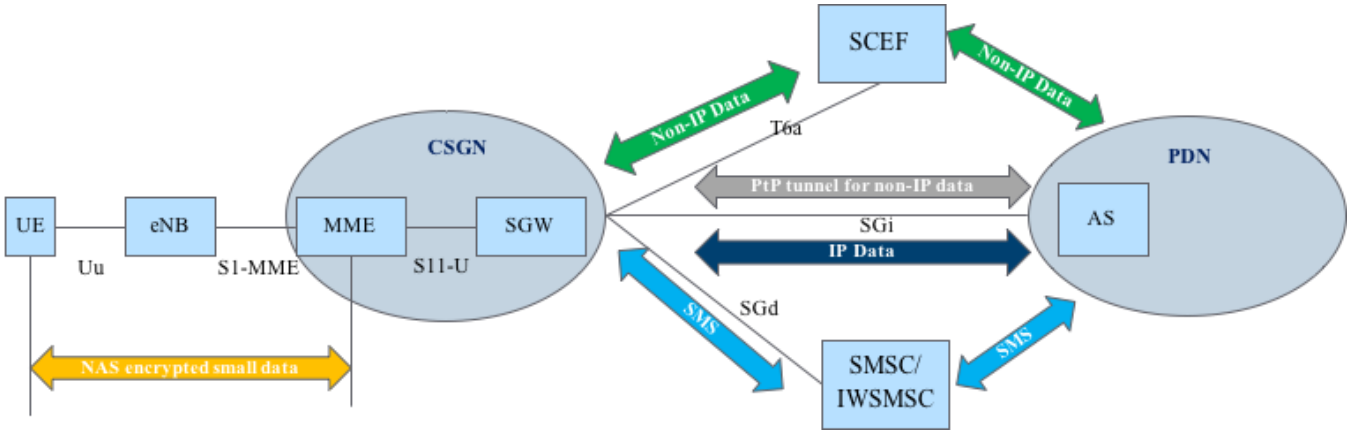

Figure 11 : 3GPP CIoT IP & Non-IP Data Paths

3GPP CIoT architecture, described in 3GPP TR 23.720 (TS 23.628), the various scenarios indicated by Non-IP and IP paths are indicated in the following figures. Control plane and User plane options provided by 3GPP CIoT, including ability to have complete IP path. 3GPP CIoT also provides non-IP path in the last mile to make the device more efficient in amount of bytes it utilizes as well as the energy it consumes (battery saving).

LwM2M v1_0 provides optimal model and protocol capabilities serving the management and data plane needs for the applications residing in IoT/M2M devices. The 3GPP-CIoT scenario for delivery of data over the User plane (see figure 4) is applicable in LwM2M v1_0 already and can be used with any new additions in LwM2M v1_1 without any specific modifications. The 3GPP-CIoT scenario for delivery of data over the Control plane needs the introduction of specific requirements and adaptations as part of LwM2M v1_1 in order to cater to the emerging needs from 3GPP CIoT in the scenarios indicated below (see control plane delivery figures).

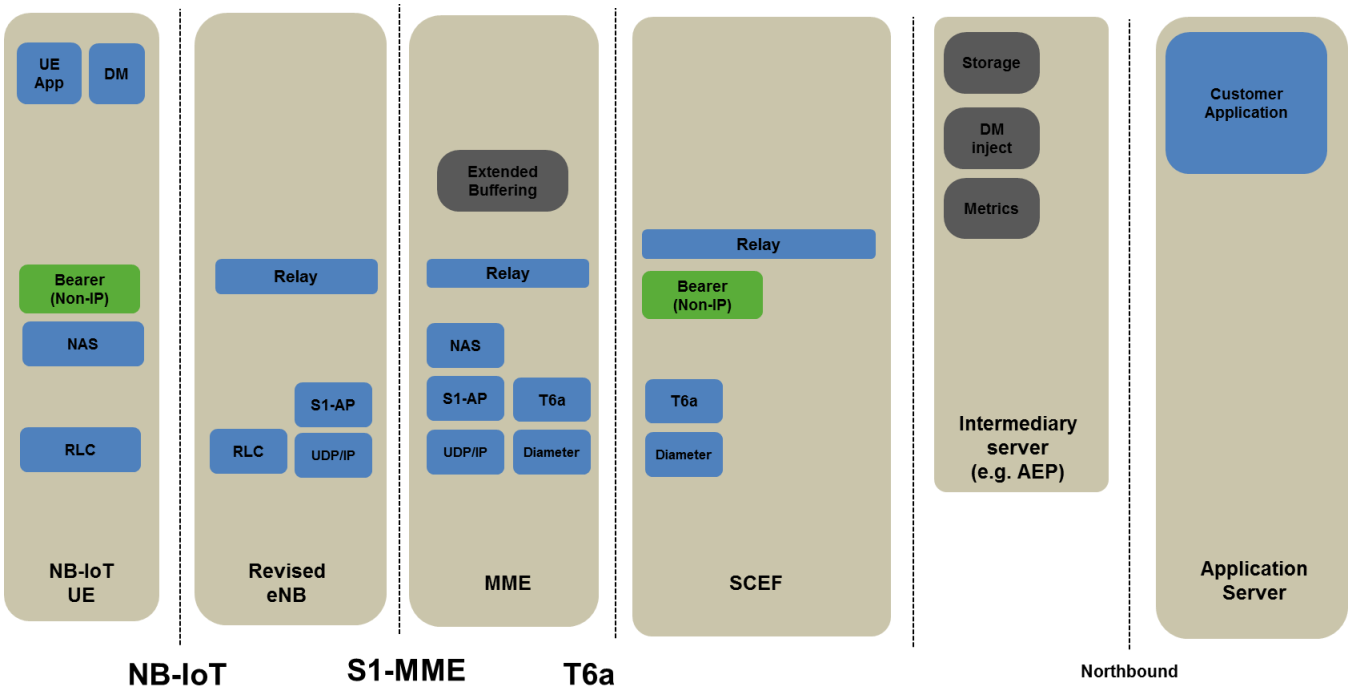

Figure 12 : non-IP delivery using control plane

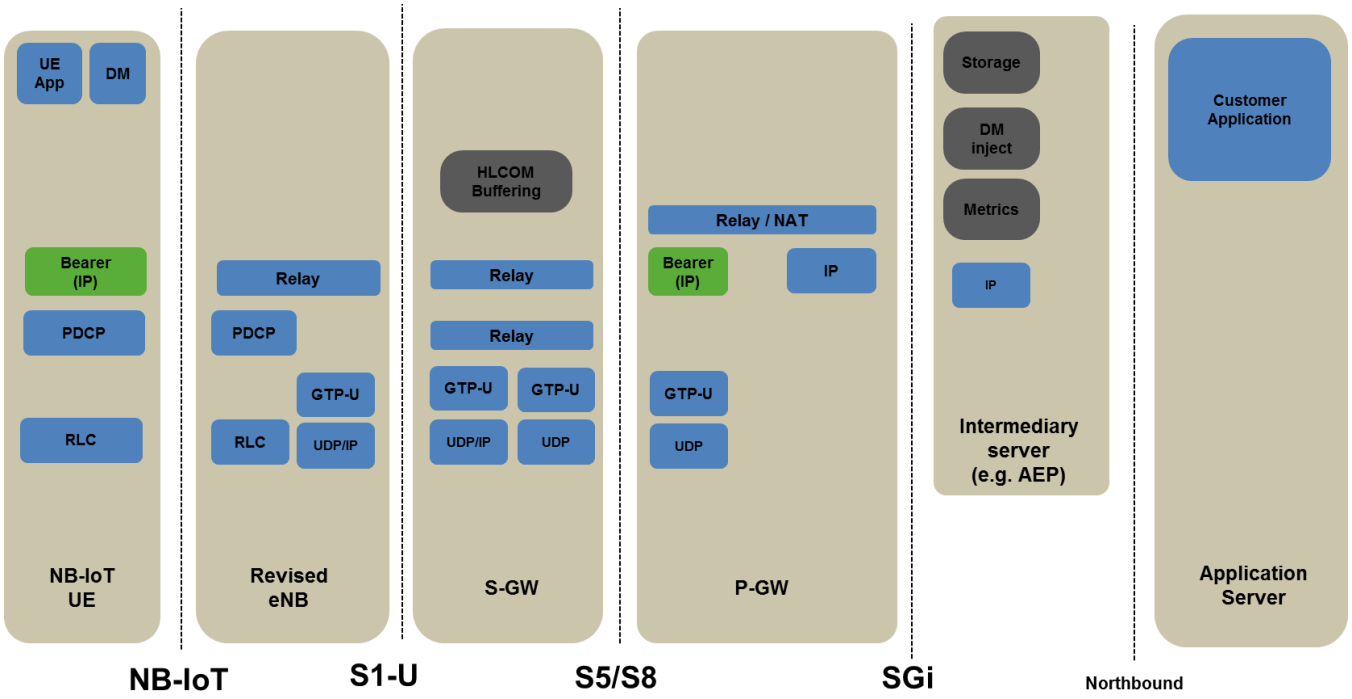

Figure 13 : Non-IP and IP scenarios using control plane

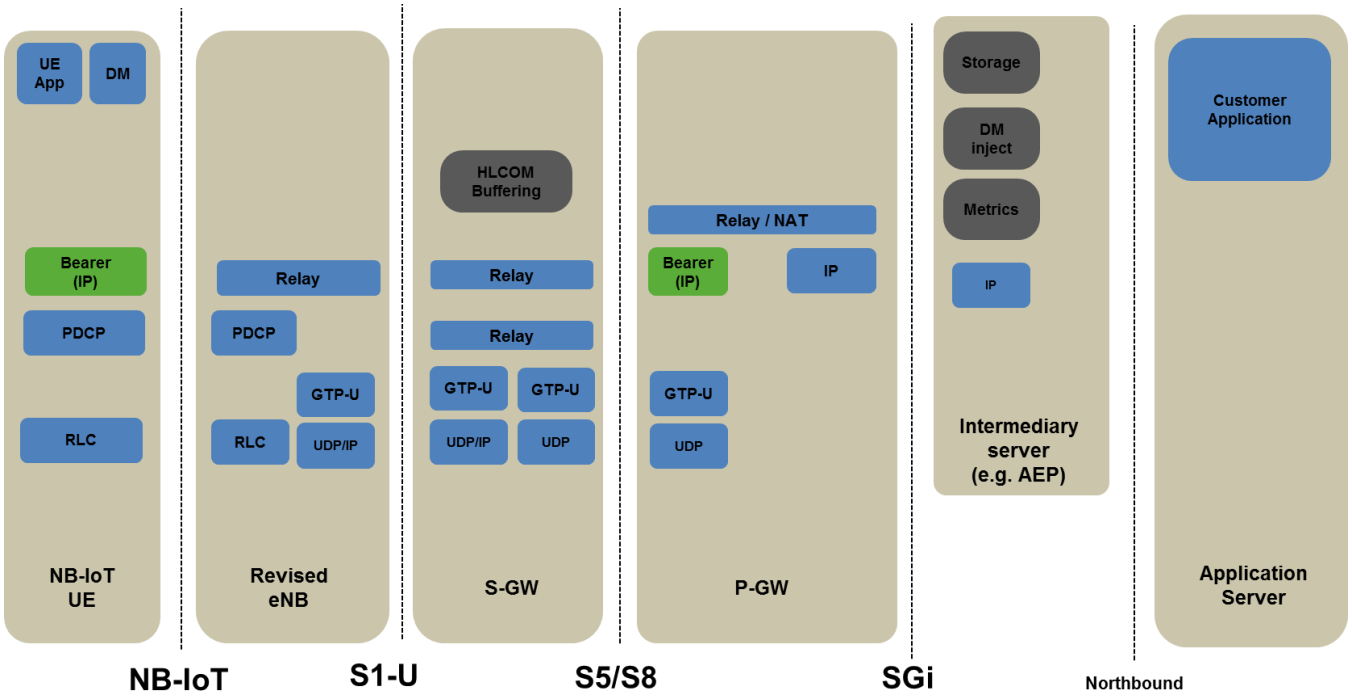

Figure 14 : User plane solution

6. Requirements (Normative)

6.1 Enabler Domains

6.1.1 LwM2M core functionality

6.1.1.1 Transports

Label	Description	Release
LightweightM2M-MBT-001	The LwM2M Enabler MUST define transport binding(s) that allows LwM2M messages to traverse all network topologies, for example in enterprise networks, more easily.	1_1
LightweightM2M-MBT-002	The LwM2M Enabler specification SHOULD separate LwM2M messaging from the underling transport bindings to allow further transports to be added more easily in the future.	1_1

Table 1 : Transports

6.1.1.2 Encoding and Standardized Data Models

Label	Description	Release
LightweightM2M-ENC-1	Lightweight M2M MUST support compact data format for information exchange	1.1
LightweightM2M-ENC-2	Lightweight M2M MUST support efficient encoding and decoding to support constrained devices and high volume	1.1
LightweightM2M-ENC-3	Lightweight M2M MUST support standardized data format for sensor and sensor measurements	1.1
LightweightM2M-ENC-4	Lightweight M2M MUST provide standardised data types for URIs	1.1
LightweightM2M-ENC-5	Lightweight M2M MUST provide standardised data types for bitmap	1.1

Table 2 : Encoding and Standardized Data Models

6.1.1.3 Bootstrap and Registration

Label	Description	Release
LightweightM2M-BsR-001	The Lightweight M2M enabler MUST provide to the Bootstrap Server, a way to retrieve the full knowledge of the Access Control Rights already in place in a LwM2M Client.	1_1
LightweightM2M-BsR-002	The Lightweight M2M enabler 1.1 MUST define a mechanism allowing a Bootstrap Server to trigger the “Client Initiated Bootstrap Mode” in a Client	1_1
LightweightM2M-BsR-003	The LwM2M Client SHALL support a configurable fall-back mechanism when errors occur during communications with the LwM2M servers.	1_1
LightweightM2M-BsR-004	Error recovery procedures SHALL be controlled by resources defined in LwM2M objects (e.g. number of retries, retry back-off timers).	1_1
LightweightM2M-BsR-005	When multiple LwM2M server accounts are configured, the LwM2M client SHALL use a resource to determine the order of registration sequence.	1_1
LightweightM2M-BsR-006	The LwM2M client SHALL use resources to determine the pre-conditions that must be met prior to a registration attempt to a LwM2M server.	1_1
LightweightM2M-BsR-007	The LwM2M client SHALL provide a mechanism for the LwM2M bootstrap server to determine whether the existing LwM2M server accounts are invalid.	1_1
LightweightM2M-BsR-008	The LwM2M client SHALL provide a mechanism for sending along with object(s) the values of certain object(s)/Resource(s) to the LwM2M server during registration.	1_1
LightweightM2M-BsR-009	The LwM2M server MAY designate a specific APN to be used by the LwM2M client when communicating with the LwM2M server.	1_1

Table 3 : Bootstrap and Registraion

6.1.1.4 Maintenance and Upgrade

Label	Description	Release
LightweightM2M-MaU-1	LwM2M SHOULD support settings save feature during upgrade to a new firmware	1_1
LightweightM2M-MaU-2	LwM2M SHOULD support reverting back to old firmware which was running prior to upgrade	1_1
LightweightM2M-MaU-3	LwM2M SHOULD support bringing back the settings which were prior to upgrade of firmware	1_1
LightweightM2M-MaU-4	LwM2M SHOULD support to make initial registrations to the LwM2M servers in a lag mode (i.e., not one go for all LwM2M servers together)	1_1
LightweightM2M-MaU-5	LwM2M SHOULD be able to configure the lag mode	1_1

Table 4 : Maintenance and Upgrade

6.1.1.5 Observation - Reporting Mode

Label	Description	Release
LightweightM2M-RMo-001	For the transitions evaluated by the defined “gt”, “lt” and “st” values in an observation, a reporting mode MAY be configured to control the reporting in LwM2M.	1_1
LightweightM2M-RMo-002	When configured to report causes of the observation, the LwM2M Client SHALL report all causes that resulted in the notification to the LwM2M server.	1_1
LightweightM2M-RMo-003	The LwM2M Client SHALL support multiple observe conditions for the same resource.	1_1

Table 5 : Reporting Mode

Label	Description	Release
LightweightM2M-DSE-001	The LwM2M 1.1 enabler MUST provide the server with the capability to use a single operation to target a list of selected resources which may be spread across same or different objects and instances. Note: This could be achieved at the data model level, transport level, or other methods.	1_1
LightweightM2M-DSE-002	The Lightweight M2M enabler 1.1 MUST define a mechanism allowing for a Server to set or to retrieve the value of a single Instance of a Multi-Instances Resource.	1_1
LightweightM2M-DSE-003	The Lightweight M2M enabler 1.1 MUST define a mechanism allowing for a Device to report unsolicited data (Objects/Resources) even though the Server has not previously sent an Observation for that data. Note: This could be used when the data model of the application is dynamic.	1_1
LightweightM2M-DSE-004	The mechanism to report unsolicited data from the LwM2M Device to the LwM2M Server MUST be configurable by the Server (at least enabling/disabling the mechanism altogether and access control).	1_1
LightweightM2M-DSE-005	The mechanism to report unsolicited data from the LwM2M Device to the LwM2M Server MUST respect the structure and format of LwM2M-defined Objects when used to report defined LwM2M-Objects.	1_1

Table 6 : Device Management & Service Enablement

Label	Description	Release
LightweightM2M-IR-001	The LwM2M Enabler 1.1 MUST define a mechanism for atomic reporting of resources across different objects in a single notification from the client to the server. e.g. when reporting both Battery level and Temperature, the resources on these objects should be returned at the same time.	1_1

Table 7 : Information Reporting

6.1.2 LwM2M Gateway functionality

Label	Description	Release
LightweightM2M-LeG-001	LwM2M SHOULD support a range of different legacy devices and their protocols and application data to be translated through the LwM2M Gateway towards the LwM2M server.	1_1
LightweightM2M-LeG-002	The LwM2M server SHOULD support configuration of the legacy gateway. (e.g., single/multiple instance for LwM2M Client)	1_1
LightweightM2M-LeG-003	LwM2M SHOULD support firmware upgrade scenarios for devices, which are behind Gateway	1_1
LightweightM2M-LeG-004	LwM2M SHOULD support group concepts on the LwM2M Gateway as represented in the Group Section)	1_1
LightweightM2M-LeG-005	LwM2M SHOULD support various topologies which the devices are configured behind the Gateway for service enablement data mapping from those devices	1_1
LightweightM2M-LeG-006	The LwM2M enabler SHOULD support the ability to retrieve the capabilities of the legacy device and security mechanisms used between the legacy device and the LwM2M gateway. (e.g., integrity protection, encryption)	1_1

Table 8 : Legacy Gateway

6.1.3 Security Enhancement

6.1.3.1 Extended PKI Support

Label	Description	Release
LightweightM2M-PKI-001	The LwM2M Enabler SHOULD define a method for checking the status of certificates.	1_1
LightweightM2M-PKI-002	The LwM2M Enabler SHOULD define a method for initializing time information.	1_1
LightweightM2M-PKI-003	The LwM2M Enabler MUST offer additional modes besides pinned certificates, for example CA certificates.	1_1

Table 9 : Extended PKI Support

6.1.3.2 TLS/DTLS Guidance

Label	Description	Release
LightweightM2M-TLS-001	The LwM2M Enabler MUST offer guidance for the use of TLS and DTLS over various transport bindings. Examples include signature algorithm choices, session resumption, use of compression, PFS, keep-alive mechanisms, timeouts, random number generators, truncated MAC / Encrypt-then-MAC extension, Server Name Indication extension, maximum fragment length negotiation, session hash, re-negotiation attacks, crypto agility, key length and algorithm recommendations, false start support, etc. Whenever possible, these recommendations SHOULD be referenced rather than copied or reproduced.	1_1
LightweightM2M-TLS-002	Errors in the DTLS / TLS procedures MUST be handled by the LwM2M procedures.	1_1

Table 10 : TLS/DTLS Guidance

6.1.3.3 Secure Component Support

Label	Description	Release
LightweightM2M-SCS-1	The Lightweight M2M Client MAY rely on a Secure Component - located in the LwM2M Device - containing sensitive information such as Credentials and Security Services mechanisms for extending the LwM2M Client Security capability.	1_1
LightweightM2M-SCS-2	When a LwM2M Client is configured to use a Secure Component, the LwM2M Client MUST be able to send to the LwM2M Server the identification data needed in order to authenticate the Secure Component.	1_1
LightweightM2M-SCS-3	When a LwM2M Client is configured to use a Secure Component, the Secure Component MUST be manageable by the LwM2M Server through the usage of a specific LwM2M Object.	1_1
LwM2M-SCS-4	The LwM2M Server MUST be able to address an Object Instance with identical operations regardless of the storage location of the Object Instance e.g. in the Secure Component or not.	1.1
LwM2M-SCS-5	When a Secure Component is available, the Server that owns the Object Instance MUST be able to indicate where that Object Instance is stored (e.g. Secure Component or not)	1.1
LwM2M-SCS-6	If a Server has the appropriate security access for an Object Instance, the Server MUST be able to query the storage location of the Object Instance (e.g. Secure Component or not).	1.1

Table 11 : Secure Component Support

6.1.3.4 E2E Security between LwM2M Server and Client

A security solution MUST support secure E2E operations between LwM2M Client and LwM2M Server via LwM2M unaware and LwM2M aware intermediate nodes.

Label	Description	Release
LightweightM2M-E2E-1	A security solution MUST be able to support E2E integrity between LwM2M Client and LwM2M Server via LwM2M unaware intermediate nodes	1.1
LightweightM2M-E2E -2	A security solution MUST be able to support E2E encryption between LwM2M Client and LwM2M Server via LwM2M unaware intermediate nodes	1.1
LightweightM2M-E2E-3	A security solution MUST be able to provide replay protection of LwM2M Operations via LwM2M unaware intermediate nodes.	1.1
LightweightM2M-E2E -4	A security solution MUST be able support authentication between the LwM2M Client and Server via LwM2M unaware intermediate nodes.	1.1
LightweightM2M-E2E-5	A security solution MUST be able to securely bind LwM2M responses with LwM2M requests via LwM2M unaware intermediate nodes	1.1
LightweightM2M-E2E-6	For certain operations, the LwM2M Client MUST be able to verify the end-to-end freshness of the request via LwM2M unaware intermediate nodes.	1.1
LightweightM2M-E2E-7	A security solution MUST be able to support E2E integrity between LwM2M Client and LwM2M Server via LwM2M aware intermediate nodes	1.1
LightweightM2M-E2E-8	A security solution MUST be able to support E2E encryption between LwM2M Client and LwM2M Server via LwM2M aware intermediate nodes	1.1
LightweightM2M-E2E-9	A security solution MUST be able to provide replay protection of LwM2M Operations via LwM2M aware intermediate nodes.	1.1
LightweightM2M-E2E-10	A security solution MUST be able support authentication between the LwM2M Client and Server via LwM2M aware intermediate nodes.	1.1
LightweightM2M-E2E-11	A security solution MUST be able to securely bind LwM2M responses with LwM2M requests via LwM2M aware intermediate nodes	1.1

LightweightM2M-E2E-12	For certain operations, the LwM2M Client MUST be able to verify the end-to-end freshness of the request via LwM2M aware intermediate nodes.	1.1
LightweightM2M-E2E-13	A security solution MUST be able to support secure fragmentation of the messages between LwM2M Server and LwM2M Client into fragments that can be verified separately, in particular in the case of firmware updates.	1.1

Table 12 : LwM2M E2E Security Requirements

6.1.3.5 E2E Security for endpoints outside of LwM2M Server and Client

Label	Description	Release
LightweightM2M-E2Eo-1	A security solution MUST be able to support E2E integrity between an IoT Device and LwM2M Server via Legacy Gateway	1.1
LightweightM2M-E2Eo-2	A security solution MUST be able to support E2E encryption between IoT Device and LwM2M Server via LegacyGateway	1.1
LightweightM2M-E2Eo-3	A security solution MUST be able to provide replay protection of LwM2M Operations via Legacy Gateway	1.1
LightweightM2M-E2Eo-4	A security solution MUST be able support authentication between the IoT Device and Server via Legacy Gateway	1.1
LightweightM2M-E2Eo -5	A security solution MUST be able to securely bind LwM2M responses with LwM2M requests via Legacy Gateway	1.1
LightweightM2M-E2Eo-6	For certain operations, the LwM2M Client MUST be able to verify the end-to-end freshness of the request via Legacy Gateway	1.1
LightweightM2M-E2Eo-7	A security solution MUST be able to support E2E integrity between LwM2M Client and Application Server via LwM2M Server	1.1
LightweightM2M-E2Eo -8	A security solution MUST be able to support E2E encryption between LwM2M Client and Application Server via LwM2M Server	1.1
LightweightM2M-E2Eo -9	A security solution MUST be able to provide replay protection of LwM2M Operations between LwM2M Client and Application Server via LwM2M Server	1.1
LightweightM2M-E2Eo-10	A security solution MUST be able support authentication between the LwM2M Client and Application Server via LwM2M Server.	1.1
LightweightM2M-E2Eo -11	A security solution MUST be able to securely bind LwM2M responses with LwM2M requests between the LwM2M Client and Application Server via LwM2M Server	1.1
LightweightM2M-E2Eo -12	For certain operations, the LwM2M Client MUST be able to verify the end-to-end freshness of the request between the LwM2M Client and Application Server via LwM2M Server	1.1

Table 13 : E2E Security Requirements outside LwM2M

6.1.4 LwM2M over LPWAN

Label	Description	Release
LightweightM2M-LPW-1	LwM2M SHOULD support External Identifier for 3GPP Cellular LPWAN	1_1
LightweightM2M-LPW-2	LwM2M SHOULD support delayed/no acknowledgement methods between LwM2M server and LwM2M client	1_1
LightweightM2M-LPW-3	LwM2M SHOULD support SMS using external Identifier (instead of MSISDN)	1_1
LightweightM2M-LPW-4	LwM2M SHALL interoperate through 3GPP CIoT network (for example SCEF API's, message identity etc.,)	1_1

LightweightM2M-LPW-5	LwM2M SHOULD support Time-to-Live for messages from LwM2M server to LwM2M Client	1_1
LightweightM2M-LPW-6	LwM2M SHOULD Support rate and byte quota in LwM2M server per device	1_1
LightweightM2M-LPW-10	LwM2M MUST support a binding for fiers LoRaWAN (eg. DevAddr, NwkSKey, AppSKey)	1_1
LightweightM2M-LPW-11	LwM2M MUST provide a mechanism for activating a LoRaWAN device using interoperable activation method (using LoRa identifiers: DevEUI, AppEUI, AppKey)	1_1
LightweightM2M-LPW-12	LwM2M MUST interoperate with LoRaWAN security protocol	1_1

Table 14 : LwM2M over LPWAN

Appendix A. Change History (Informative)

A.1 Approved Version History

Reference	Date	Description
OMA-RD-LightweightM2M-V1_1-20180710-A	10 Jul 2018	Status changed to Approved by DM Doc Ref # OMA-DM&SE-2018-0076-INP_LightweightM2M_V1_1_RD_for_final_Approval

Appendix B. Use Cases (Informative)

B.1 use case and high level requirement – all

ID	Use Case	High level requirement	Reference	Presented Company(s)	Version planned
B1	3GPP CIoT and emergence of LPWAN technologies to be supported by LwM2M releases (NB-IoT and LTE CAT-M)		2017_INP91	Vodafone, Nokia, ARM, Orange, Gemalto, Ublox	1_1
		Support of connection management and 3GPP CIoT parameters in LwM2M			
		Support of SCEF path in 3GPP CIoT			
		Support of IP path inside 3GPP CIoT			
B1’	LoRA ~~& LTE Cat M~~ Support	A general framework for supporting LPWAN (non IP) in LwM2M is expected in the field	2017 _INP60	ORANGE*, Gemalto	1_1
B2	Maintenance and upgrade of constrained devices would be necessary in the field once deployed. This should avoid unnecessary overheads in terms of configuration and reconfigurations.		2017_INP91	Nokia, Qualcomm, Sierra Wireless	1_1
		Upgrading firmware preserves pre-		Nokia, Qualcomm, Sierra	

		upgrade settings and device comes online without bootstrapping		Wireless	
		Reverting to previous firmware is made possible and device comes online without bootstrapping		Nokia, Qualcomm, Sierra Wireless	
B3	When supporting multiple LwM2M server instances it is sometime necessary to have a configurable timer (seconds) with which the client waits after being bootstrapped to register the first time with each to the server instances. This provides an opportunity for a provisioning system to learn of the deviceID and bs/ created credentials and provision them to the proper server instances prior to client registering		2017_INP91	Nokia, Verizon, Qualcomm	1_1
B4	In field scenarios without having ability to address resource instances directly in the LwM2M command	Extended addressing for multiple resource instances	2017 _INP60	Gemalto*, Sierra Wireless, ARM, ORANGE, Nokia	1_1
B4’		ACL to be extended for resource level.	2017_INP91	Nokia, Verizon	>1_1
B5	In order to reduce the size of payload further CBOR can be	OMA-CBOR compact Media Type	2017 _INP60	Gemalto*, Sierra Wireless	1_1

	utilized in LwM2M to achieve better compression as well utilize all the functionalities provided by OMA-JSON.	introduction		ARM ORANGE Ericsson Nokia, u-blox	
B6	Remove the last limitation for Incremental Bootstrap capability	Bootstrap Enhancement	2017 _INP60	Gemalto*, ORANGE, u-blox	1_1
B7	An Instance of a virtual object refers resources of various Object Instances.	Virtual Object Concept	2017 _INP60	Gemalto*, ORANGE, u-blox, HUAWEI	1_1
B7’	1) Aggregation of service and DM data in a single message 2) Observing multiple resources with a single notify	Define New LwM2M operations with binding to CoAP FETCH and PATCH methods as specified in RFC 8132	2017_INP89	u-blox, Huawei, Vodafone	1_1
B7’’	When a LwM2M server needs to get/observe resource data from the different objects at the same time, a mechanism that reports the data in a single message instead of in several messages is desirable to reduce the number of message exchanges, so as to save the power consumption and the bandwidth of the constrained device/network.	Support aggregation of resource data from different objects in a single reporting message.	2017_INP74	Huawei	1_1
B9	Generalized framework for	Secure Element	2017 _INP60	Gemalto*,	1_1

	extending LwM2M Security (SE, eUICC-M2M support).	Support		Ericsson, ORANGE, Nokia	
B10	Make registration interface more efficient	Add support for Patch	0121-INP_patch_support (note also 0089R01-INP_Binding_to_CoAP_FETCH_PATCH)	ARM, Nokia	1_1
B11	When a LwM2M server needs to issue the same commands or data to a group devices, a group multicast mechanism can increase transmission efficiency and save the bandwidth.	Support group multicast (RFC7390) for issuing the same commands or data to a group of devices.	2017_INP83	Huawei u-blox	>1_1
B12	Extended support of the PKI infrastructure	Add new certificate provisioning types, and discuss secure time and revocation strategy	OMA-DM- 2017-0042- INP_security_ features	ARM, Sierra Wireless, Gemalto	1_1
B13	Alignment with IoT DTLS/TLS security recommendations		OMA-DM- 2017-0043- INP_rfc7925 _support	ARM, u-blox	1_1
B14	Adding support for user identity management		OMA-DM- 2017-0044- INP_user_ identity_ management	ARM	>1_1
B15	Support for multi-tenancy	Add additional identifier to indicate customer	OMA-DM-2017-0045-INP_multi_tenancy	ARM	>1_1
B16	Make communication with Bootstrap Server more defined in error situations.	Provide more details on when a LwM2M has to re-connect to the bootstrap server to recover from error conditions	0114-INP_bootstrap_reconnect	ARM, u-blox, Gemalto, Sierra-wireless	1_1

B17	Support non-LwM2M legacy devices (behind a gateway)	Extend the data model so that a LwM2M server can interface with a LwM2M client running on a gateway to interface non-LwM2M-enabled devices.	0115-INP_legacy_gateway	ARM, Nokia, Sierra Wireless, Orange, Gemalto, Ericsson	1_1
B18	Make queue mode more efficient by taking the resource volatility into account.	Support resource volatility	0117-INP_volatility	ARM	>1_1
B19	Make LwM2M robust in environments where firewalls block UDP traffic.	Support CoAP over TCP specification	0118-INP_coap_over_tcp	ARM, Ericsson, Sierra wireless	1_1
B20	Provide Application layer end to end Security for LwM2M		INP98	Ericsson, Nokia	1_1
B21	Update the LwM2M specification to use the more updated Sensor Markup Language Reference specification	Support SenML Version 1_1	CR64R02	Ericsson, Sierra Wireless, ARM	1_1