OMA Lightweight Machine to Machine Requirements

Candidate Version 1.2 – 24 Jan 2019

Open Mobile Alliance

OMA-RD-LightweightM2M-V1_2-20190124-C

Contents

1. SCOPE (INFORMATIVE) 5
2. REFERENCES 6
   - 2.1 NORMATIVE REFERENCES 6
   - 2.2 INFORMATIVE REFERENCES 6
3. TERMINOLOGY AND CONVENTIONS 7
   - 3.1 CONVENTIONS 7
   - 3.2 DEFINITIONS 7
   - 3.3 ABBREVIATIONS 8
4. INTRODUCTION (INFORMATIVE) 9
   - 4.1 VERSION 1.2 9
   - 4.2 VERSION 1.1 9
   - 4.3 VERSION 1.0 9
5. LWM2M 1.2 DESCRIPTION (INFORMATIVE) 11
   - 5.1 END-TO-END SERVICE DESCRIPTION 11
     - 5.1.1 Software Management/Firmware Management 11
     - 5.1.2 LwM2M Gateway 11
     - 5.1.3 Message Identity 11
     - 5.1.4 Group Firmware Upgrade 11
     - 5.1.5 Registration Interface 11
     - 5.1.6 Bootstrap and Registration 12
     - 5.1.7 Bootstrap Process 12
     - 5.1.8 Transports 12
     - 5.1.9 Transport Bindings 12
     - 5.1.10 Version Negotiation 13
     - 5.1.11 Reduction of the bandwidth usage of LwM2M 13
6. REQUIREMENTS (NORMATIVE) 14
   - 6.1 HIGH-LEVEL FUNCTIONAL REQUIREMENTS 14
     - 6.1.1 Trigger Mode 14
     - 6.1.2 Information Reporting Interface 15
     - 6.1.3 Device Management and Service Enablement Interface 15
     - 6.1.4 Security 15
   - 6.2 OVERALL SYSTEM REQUIREMENTS 16
   - 6.3 FIRMWARE UPDATES REQUIREMENTS 16
   - 6.4 BOOTSTRAPPING REQUIREMENTS 16
   - 6.5 PROFILE IDENTIFIER REQUIREMENTS 16
   - 6.6 REGISTRATION INTERFACE 16
   - 6.7 BOOTSTRAP AND REGISTRATION OPTIMIZATIONS 17
   - 6.8 REGISTRATION AND DISCOVERY 17
   - 6.9 BOOTSTRAP CLARIFICATIONS 18
   - 6.10 VERSION NEGOTIATION 18
   - 6.11 ENCODING AND STANDARDIZED DATA MODELS 18
   - 6.12 CORE 18
   - 6.13 TRANSPORTS 19

APPENDIX A. CHANGE HISTORY (INFORMATIVE) 20
   - A.1 APPROVED VERSION HISTORY 20
   - A.2 DRAFT/CANDIDATE VERSION 1.2 HISTORY 20

Tables

Table 1: High-Level Functional Requirements **14**
Table 2: Requirements - Trigger Mode Items **14**
Table 3: Requirements - Information reporting Interface **15**
Table 4: Requirements – Device Management and Service Enablement Interface **15**
Table 5: Requirements – Security Items **15**
Table 6: Requirements – Communication Security **15**
Table 7: Requirements – Firmware Updates **16**
Table 8: Requirements – Bootstrapping **16**
Table 9: Requirements – Profile Identifier Support **16**
Table 10: Requirements – Registration Interface **16**
Table 11: Requirements – Optimization of Bootstrap and Registration **17**
Table 12: Requirements – Registration and Discovery Items **17**
Table 13: Requirements – Bootstrap Clarifications **18**
Table 14: Requirements – Version Negotiation **18**
Table 15: High-Level Functional Requirements – Encoding **18**
Table 16: Requirements – Core **18**
Table 17: Requirements – Transport **19**

1. Scope (Informative)

This document represents Lightweight M2M version 1.2 consolidated requirements.

2. References

2.1 Normative References

[RFC2119] "Key words for use in RFCs to Indicate Requirement Levels", S. Bradner, March 1997, URL: https://www.ietf.org/rfc/rfc2119.txt

2.2 Informative References

[OMADICT] "Dictionary for OMA Specifications", Version x.y, Open Mobile Alliance™, OMA-ORG-Dictionary-Vx_y, URL:http://www.openmobilealliance.org/

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

Bootstrap Provisioning	The process of providing initial parameters and/or applications on the LwM2M Device so that it can be brought under management
Connection Address	Connection Address is a network identifier that is used by the LwM2M server to access the LwM2M device via a specific communication bearer.
Connectivity Status	Connectivity Status is a state for a specific network connection of a LwM2M Device. It tells that whether a LwM2M Device and LwM2M Server are able to exchange message through a specific network bearer or not. There are two statuses, connected or disconnected.
Device Discovery	The process of identifying the LwM2M Device by the LwM2M Server
Device Registration	The process of adding the information of the LwM2M Device to the database so that remote access and management to the LwM2M Device is achievable
Device State	Device State is a unique condition that represents available capabilities of LwM2M Device. Information Note: possible Device State value could be ‘Active’, ‘Idle’, ‘Sleep’.
Disabling Device	Disabling a device or a Physical Resource is the process of disable their full capabilities, with the exception of being able to process an Enabling request.
Enabling Device	Enabling a device or a Physical Resource is the process of enable its capabilities.
LwM2M Client	A logical component residing in the LwM2M Device conforming to the requirements for the LwM2M Client specified in this enabler. This LwM2M Client serves as an end-point of the LwM2M protocol, and communicates with the LwM2M Server to execute the operations from the LwM2M Server for the device and the service management
LwM2M Device	A LwM2M Device is a device that runs (a) M2M application(s) and communicates through the Network Provider’s network.
LwM2M Server	A logical component residing within the M2M Service Provider or the Network Provider which serves as an end-point of the LwM2M protocols. The LwM2M Server provides the following high level functionalities: discovery and registration, bootstrap provisioning, and device and service management
LwM2M Service	LwM2M Service is a service that is provided to M2M Users by M2M Service Provider
M2M Network Subscriber	M2M Network Subscriber is a M2M User or a M2M Service Provider that has a contractual relationship with the Network Provider to use the network communication service.
M2M Service Provider	A M2M Service Provider provides (a) M2M service(s) to the M2M User by communicating to the LwM2M Client through the Network Provider’s network.
M2M Service Subscriber	M2M Service Subscriber is the M2M User that has a contractual relationship with a M2M Service Provider to use M2M Services.
M2M User	A M2M User uses the service provided by the M2M Service Provider.
Network Provider	A Network Provider offers network communication services over its wireless and/or wireline network.
Physical Resource	Physical Resource is any physical entity that works as a part of the LwM2M device or works as a peripheral.
Power Saving Mode	Power Saving Mode is a setting for the LwM2M Device that helps to decrease its power consumption and meanwhile keep full or partial capabilities available

3.3 Abbreviations

APN	Access Point Name
CBOR	Compact Binary Object
IOT	Internet Of Things
LwM2M	Lightweight Machine to Machine
OMA	Open Mobile Alliance
SenML	Sensor Markup Language
SMS	Short Message Service
TCP	Transmission Control Protocol
UDP	User Datagram Protocol

4. Introduction (Informative)

4.1 Version 1.2

This document augments LwM2M version 1.1. Version 1.2 is backwards compatible to v1.1 with respect to mandatory features.The following are enhancements and new features are introduced:

1. Bootstrap Enhancements
2. Security Enhancements
3. Compatibility Negotiation
4. Software/Firmware Update Enhancements
5. Registration Optimizations
6. Transport Binding Conflict Resolution

4.2 Version 1.1

This version 1.1 augments LwM2M version 1.0. Version 1.1 is backwards compatible to v1.0 with respect to mandatory features.This specification version 1.1 of the LwM2M protocol defines the following new features:

1. Enhancement of the LwM2M bootstrapping capabilities allowing for incremental upgrades.
2. Improved support for Public Key Infrastructure (PKI) deployments.
3. Introduction of enhanced registration sequence mechanisms by the LwM2M Client to LwM2M Server(s).
4. Support for LwM2M over TCP/TLS to better support firewall and NAT traversal.
5. Support for application layer security for LwM2M based on OSCORE
6. Better support of LwM2M over Low Power WANs, including 3GPP CIoT & LoRaWAN.
7. Extended LwM2M commands to enable Resource Instance level access.
8. Performance improvement for retrieving and updating Resources of multiple objects.
9. Support for JSON using SenML with CBOR serialization for compressed payload with highly efficient transmission.
10. Addition of new data types.

Additional transport & security enhancements in LwM2M v1.1 are in a separate document specifying the transport bindings of the LwM2M protocol version 1.1.

The split between the LwM2M core and the transport binding specification improves readability, allows a cleaner separation between the LwM2M messaging layer and the underlying protocols for conveying these messages, and ultimately better extensibility.

This version 1.1, adds support for CoAP over TCP/TLS, and CoAP over Non-IP, namely 3GPP CIoT and LoRaWAN.

This version 1.1 also supports the application layer security protocol OSCORE which enables support for proxy operations and end-to-end security independently of transport layer protocol.

4.3 Version 1.0

The version 1.0 of the enabler defines the application layer communication protocol between a LwM2M Server and a LwM2M Client, which is located in a LwM2M Device. The OMA Lightweight M2M enabler includes device management and service enablement for LwM2M Devices. The target LwM2M Devices for this enabler are mainly resource constrained devices. Therefore, this enabler makes use of a light and compact protocol as well as an efficient resource data model.

A Client-Server architecture is introduced for the LwM2M Enabler, where the LwM2M Device acts as a LwM2M Client and the M2M service, platform or application acts as the LwM2M Server. The LwM2M Enabler has two components, LwM2M Server and LwM2M Client.

LwM2M v1.0 offers the following features:

1. Simple resource model with the core set of objects and resources defined in this specification. The full list of registered objects can be found at OMNA.
2. Operations for creation, update, deletion, and retrieval of resources.
3. Asynchronous notifications of resource changes.
4. Support for several serialization formats, namely TLV, JSON, Plain Text and binary data formats and the core set of LightweightM2M Objects.
5. UDP and SMS transport support.
6. Communication security based on the DTLS protocol supporting different types of credentials.
7. Queue Mode offers functionality for a LwM2M Client to inform the LwM2M Server that it may be disconnected for an extended period and when it becomes reachable again.
8. Support for use of multiple LwM2M Servers.
9. Provisioning of security credentials and access control lists by a dedicated LwM2M bootstrap-server.

5. LwM2M 1.2 description (Informative)

Enhancements to Lightweight M2M v1.1

5.1 End-to-end Service Description

5.1.1 Software Management/Firmware Management

Usage Pattern

- Retail cases needs wait state for specific user inputs, whereby the devices have small entry display for certain user inputs and act as a go sign to progress on upgrade
- Mission critical devices like medical applications needs device to be in ready condition before the FOTA push, as they be in the middle of performing certain actions which cannot be overridden
- Agricultural use cases like remote device not-so-available online always needs to be factored in during a push

5.1.2 LwM2M Gateway

LwM2M Gateway can act in the following modes (similar to OMA-DM Gateway)

- Transparent: The LwM2M Gateway assists the LwM2M Server in sending a LwM2M Notification to the End Device(s) behind the DM Gateway
- Proxy: The LwM2M Gateway manages End Device(s) behind the LwM2M Gateway on behalf of the LwM2M Server over LwM2M protocol
- Adaptation: Similar to the Proxy Mode with the difference that the LwM2M Gateway manages End Device(s) behind the LwM2M Gateway on behalf of the DM Server over non-LwM2M protocols (such as UPnP, TR069, USP, Zigbee, Bluetooth, OMA-DM etc.)
- OMA-DM: In this mode the intention is to replace OMA-DM interface through LwM2M gateway. In order to create the commands of OMA-DM flow through LwM2M. In certain configuration cases, LwM2M could straight replace XML file configurations.

5.1.3 Message Identity

Multicast needs would make LwM2M server relationship to a particular message gets lost in the response, it would be essential to make available an identity on the LwM2M level of the protocol in order to keep the synergy. This may also be applicable in cases where unicast is used and the LwM2M message is traversed through different 3rd party systems. It may be essential to interpret and understand the response based on the original request. Supporting such an identity would make LwM2M more robust across boundaries of service entities and telecommunication networks.

5.1.4 Group Firmware Upgrade

Firmware upgrade of LPWAN based devices is important feature to manage the lifecycle of devices. LPWAN devices are expected to be deployed for many years (e.g. Smart Meters), firmware upgrade is a business imperative. Constrained devices with low power, limited CPU, limited memory and battery operated with expectation to be running for multiple years independent of power source or battery replacement(s), need a solution which is LPWAN based in order to make it efficient and sustainable.

5.1.5 Registration Interface

In case of multiple LwM2M server accounts configured on the LwM2M client, LwM2M enabler mandates the client to register to all the servers using registration priority order(s). However, there are several use cases in which the client is configured with multiple LwM2M server accounts but does not require registering to all the servers. Registration of the client

to a selected list of LwM2M server(s) is sufficient for the use case and non-connected LwM2M server account(s) on the client can be considered as offline.

For the client to determine which LwM2M server(s) to connect to, the client needs to be configured on which terms and conditions the registration shall happen. This configuration should not impose proprietary methods not to break interoperability, instead use the structures defined in the LwM2M enabler and minimize the resources used on the device itself.

5.1.6 Bootstrap and Registration

Information written on the device during bootstrapping is most of the time the same for devices of same or similar type. Similarly, registration information retrieved from the client is generally the same for devices of same or similar type. Transmitting mostly static information introduces overhead in terms of transmission overhead and delay. Hence, both bootstrap and registration interfaces can be optimized further.

5.1.7 Bootstrap Process

The currently defined bootstrap process does not allow initialization of LwM2M Servers from multiple sources including factory provisioning, smart card and one or more "bootstrap" servers. The bootstrap process needs clarification and enhancement to enable successful initial bootstrapping and re-bootstrapping from multiple sources concurrently.

5.1.8 Transports

The Constrained Application Protocol (CoAP) supports several bindings which are UDP, Datagram Transport Layer Security (DTLS) over UDP, TCP, Transport Layer Security (TLS) over TCP and SMS. While these bindings cover most of the transport protocols used in Internet of Things (IoT) deployments, publish-subscribe based transfer protocols still have a large portion of use in IoT deployments.

While having their drawbacks, the reasons why publish-subscribe transfer protocols still share a large portion can be listed as enabling to collect information from devices across networks with IP architecture and not being blocked by firewalls generally because of TCP-oriented connections.

However, there is no standardized way of handling device management and service enablement in IoT deployments with publish-subscribe transfer protocols. These ecosystems are highly fragmented with proprietary methods for device management and service enablement with obvious problems of interoperability between devices and networks. Hence, there is opportunities for LwM2M to enable solutions to standardized device management and service enablement capabilities for publish-subscribe transfer protocols.

5.1.9 Transport Bindings

LwM2M resources exist to indicate capabilities of both the server and client in terms of the supported transport bindings. The server declares which bindings are supported by the server in binding resource (Resource 7) of the LwM2M server object (Object 1). The client declares to the server which bindings are supported in the client in Supported Binding and Modes resource (Resource 16) in the Device object (Object 3).

The server also declares a single preferred binding to be used by the client in Preferred Transport resource (Resource 22) in LwM2M server object (Object 1). The server can override the Preferred Transport for a single packet data network connection by including an argument when the Registration Update Trigger resource (Resource 8) in the LwM2M server object (Object 1) is executed.

There are many cases where these values and other transport binding related information (e.g. uri values, APN Connectivity profile resources) are inconsistent such that the client must choose between specified transport bindings. The algorithm to enable a consistent selection needs to be clarified.

5.1.10 Version Negotiation

As the LwM2M enabler continues to evolve, additional enabler versions will exist concurrently in deployment scenarios. To support independent deployment of servers and clients, version negotiation should occur during bootstrapping and registration to enable an agreement between the server and client on the version to be used.

5.1.11 Reduction of the bandwidth usage of LwM2M

To transmit a single resource value, LwM2M allows four encodings:

- Plain text (3 bytes “123” or 5 bytes “12.34”) overhead varies.
- SenML (24 bytes “[{"n":"/1/0/1","v":123}]” or 26 bytes “[{"n":"/6/0/0","v":12.34}]”) overhead is at least 21 bytes.
- CBOR (7 bytes “0xA1 0x00 0xA2 0x2F31 0x18 0x7B” or 14 bytes “0xA1 0x00 0xA2 0x2F30 0xFA 0x414570A4”) overhead is 6 bytes.
- TLV (3 bytes “0xC1 0x01 0x7B” or 6 bytes “0xC4 0x00 0x414570A4”) overhead is 2 bytes

With the deprecation of the TLV format, a low overhead representation of single resource values is missing.

A possible technical solution is to transmit the value as a binary blob. There would be no overhead (“0x7B” or “0x414570A4”).

In the TS, this would be implemented in Appendix C (Data Types) by adding a new column for binary that would duplicates what is stated for TLV encoding.

E.g.:

Data Type	Text Format	TLV Format	Binary
Unsigned Integer	Represented as an ASCII unsigned integer.	Represented as a binary unsigned integer in network byte order. The value may be 1 (8-bit), 2 (16-bit), 4 (32-bit) or 8 (64-bit) bytes long as indicated by the Length field.	Represented as a binary unsigned integer in network byte order. The value may be 1 (8-bit), 2 (16-bit), 4 (32-bit) or 8 (64-bit) bytes long as indicated by the Payload Length.
Float	Represented as an ASCII signed numeric representation.	Represented as a binary floating point value [IEEE 7542008] [FLOAT]. The value may use the binary32 (4 byte length) or binary64 (8 byte length) format as indicated by the Length field.	Represented as a binary floating point value [IEEE 7542008] [FLOAT]. The value may use the binary32 (4 byte length) or binary64 (8 byte length) format as indicated by the Payload Length.
Boolean	Represented as the ASCII value 0 or 1.	Represented as an 8 bit unsigned Integer with value 0, or 1. The Length of a Boolean value MUST always be 1 byte.	Represented as an 8 bit unsigned Integer with value 0, or 1. The Payload Length of a Boolean value MUST always be 1 byte.

The may issue is the content-format to use when transmitting such a payload.

One possibility is to reuse “application/octet-stream”. Both the Server and the Client would know how to interpret this value because of the LwM2M Data Model which defines the data type of the targeted resource. This is similar to the usage of “text/plain”.

Another possibility is to register new media types for the various data types defined in LwM2M e.g. “application/signed-integer”, “application/unsigned-integer”, “application/float” etc.

6. Requirements (Normative)

6.1 High-Level Functional Requirements

Label	Description	Release
Sw-Mgmt-01	The LwM2M Enabler & SwMgmt Enabler SHALL create single software management/firmware upgrade ability for the market.	1.2
Sw-Mgmt-02	The SwMgmt Enabler SHALL create standardized mechanisms for manifest of the upgrade process	1.2
Sw-Mgmt-03	The SwMgmt Enabler SHALL provide abilities to indicate the performance of the upgrade procedures (time of acceptance and actions interface by the user)	1.2
Gw-Enab-01	The LwM2M Enabler and LwM2M Gateway Enabler SHALL provide abilities to manage the end device through transparent mode, whereby LwM2M Gateway is pass through for commands on either direction	1.2
Gw-Enab-02	The LwM2M Enabler and LwM2M Gateway Enabler SHALL provide abilities to manage the end device through proxy mode, whereby LwM2M Gateway is acting as a LwM2M server on behalf of LwM2M server	1.2
Gw-Enab-03	The LwM2M Enabler and LwM2M Gateway Enabler SHALL provide abilities to manage the end device through adaptation mode, whereby LwM2M Gateway translates the different end device protocol as LwM2M towards the server and vice versa	1.2
Msg-Iden-01	The LwM2M Enabler SHALL provide message identity as a unique way of identifying the commands and responses.	1.2
Grp-Fw-01	The LwM2M Enabler SHALL support blob delivery for a group of LwM2M clients	1.2
Grp-Fw-02	LwM2M Enabler SHALL support blob delivery for a group over unicast	1.2
Grp-Fw-03	LwM2M Enabler MUST support blob delivery for a group over multicast	1.2

Table 1: High-Level Functional Requirements

6.1.1 Trigger Mode

Label	Description	Release
LwM2M-TRIG-01	Lightweight M2M MUST extend the Trigger Mode to other transports than SMS.	1.2
LwM2M-TRIG-02	Lightweight M2M MUST support Trigger Mode broadcast on transports that support broadcast capability.	1.2
LwM2M-TRIG-03	Lightweight M2M MUST limit the Trigger Mode to Execute operations on the Registration Update Trigger and BootstrapRequest Trigger resources.	1.2
LwM2M-TRIG-04	Lightweight M2M MUST restrict the Trigger Mode to authorized entities.	1.2
LwM2M-TRIG-05	Lightweight M2M MUST allow the Trigger Mode to trigger a registration to the LwM2M Server.	1.2

Table 2: Requirements - Trigger Mode Items

6.1.2 Information Reporting Interface

Label	Description	Release
LwM2M-INFO-01	LwM2M Observe Operation MUST support attributes valid only for the observation in a standard way.	1.2
LwM2M-INFO-02	LwM2M Observe-Composite Operation MUST support attributes valid only for the observation in a standard way.	1.2
LwM2M-INFO-03	LwM2M MUST provide a way to observe only raising edge or falling edge transitions of Boolean resources.	1.2

Table 3: Requirements - Information reporting Interface

6.1.3 Device Management and Service Enablement Interface

For some Objects, when creating a new instance, the LwM2M Server does not have the required Resource values, e.g. the Location Object. For other Objects, the Client does not have the capability to create the new Instance, e.g. a sensor Object.

Label	Description	Release
LwM2M-DMSE-01	The LWM2M enabler SHALL allow the LWM2M Client to refuse a Create or Delete operation from the LwM2M Server.	1.2
LwM2M-DMSE-02	The LWM2M enabler SHALL give the LwM2M Server the ability to request the LwM2M Client to create Object Instances without being required to provide all of the Resource values of the new Instance.	1.2

Table 4: Requirements – Device Management and Service Enablement Interface

6.1.4 Security

Label	Description	Release
LwM2M-SMS-01	The LwM2M Enabler SHALL provide a security mechanism for LwM2M enabler messages over SMS initiated by LwM2M Server is unique.	1.2

Table 5: Requirements – Security Items

6.1.4.1 Communication Security

Label	Description	Release
LwM2M-CS-01	The LwM2M enabler SHALL support the use of TLS 1.3. TLS 1.3 reduces the number of roundtrips.	1.2
LwM2M-CS-02	The LwM2M enabler SHALL support the use of DTLS 1.3. DTLS 1.3 in addition to the roundtrip improvements also optimizes the record layer format, which leads to lower over-the-wire overhead.	1.2
LwM2M-CS-03	The LwM2M enabler SHALL support the use of the DTLS 1.2 Connection ID. The Connection ID extension for DTLS adds an additional record layer header field to improve an alternative demultiplexing strategy. As a result, changes of the IP address and ports by NATs will not have an impact on the correct processing of DTLS-protected packets. Note that the Connection ID functionality is also available for DTLS 1.3.	1.2

Table 6: Requirements – Communication Security

6.2 Overall System Requirements

6.3 Firmware Updates Requirements

Label	Description	Release
LwM2M-FW-01	The LwM2M enabler SHALL support security protection of the firmware image and associated meta-data in an end-to-end fashion. Note that there is currently work ongoing to standardize the format of the meta-data along with the end-to-end security protection mechanism.	1.2
LwM2M-FW-02	The LwM2M enabler SHALL minimize the amount of redundant information contained in the LwM2M Firmware Update Object with respect to what is already contained in the end-to-end protected meta-data.	1.2
LwM2M-FW-03	The LwM2M enabler SHALL support extensible and flexible error reporting so that trouble-shooting is simplified.	1.2

Table 7: Requirements – Firmware Updates

6.4 Bootstrapping Requirements

Label	Description	Release
LwM2M-BS-01	The LwM2M specification SHALL ability to rotate bootstrap credentials. This feature is, for example, needed when the long-term credential expires and needs to be replaced.	1.2

Table 8: Requirements – Bootstrapping

6.5 Profile Identifier Requirements

Introduction of LwM2M Profile Identifier to further optimise registration message

Label	Description	Release
LwM2M-Ident-01	The LwM2M enabler SHALL support optional use of Profile ID in the client Registration message	1.2
LwM2M-Ident-02	When LwM2M Profile ID is present in the registration message, objects and instances MAY be omitted	1.2
LwM2M-Ident-03	Semantics and meaning of LwM2M profile IDs are outside scope of LwM2M specifications	1.2
LwM2M-Ident-04	LwM2M Profile Ids have to be registered with OMA and have publicly available precise definition and semantics	1.2

Table 9: Requirements – Profile Identifier Support

6.6 Registration Interface

Label	Description	Release
LwM2M-REG-01	The LWM2M enabler SHALL allow LWM2M client to determine which LWM2M server(s) to register to when multiple LWM2M server accounts are configured.	1.2
LwM2M-REG-02	The LWM2M enabler SHALL allow configuration of LWM2M client to determine which LWM2M server(s) to register to.	1.2
LwM2M-REG-03	The LWM2M enabler SHALL allow LWM2M client to change its registration during its lifetime between LWM2M server(s) based on the provided configuration.	1.2

Table 10: Requirements – Registration Interface

6.7 Bootstrap and Registration Optimizations

Label	Description	Release
LwM2M-Opt-01	The LWM2M enabler SHALL provide methods to optimize bootstrap sequence and payload(s) delivered during bootstrapping.	1.2
LwM2M-Opt-02	The LWM2M enabler SHALL provide methods to optimize registration sequence and payload(s) delivered during registration.	1.2
LwM2M-TRAN-BIND-01	The client SHALL use the binding indicated in the Preferred Transport resource (Resource 24) in LwM2M server object (Object 1), if defined, unless unable to do so.	1.2
LwM2M-TRAN-BIND-02	The client SHALL use the argument when the Registration Update Trigger resource (Resource 8) in the LwM2M server object (Object 1) is executed unless not supported by the client or not indicated as supported by the server in binding resource (Resource 7) of the LwM2M server object (Object 1).	1.2
LwM2M-TRAN-BIND-03	When the Registration Update Trigger resource (Resource 8) in the LwM2M server object (Object 1) is executed and contains an argument indicating an overriding binding that is supported by the client, the client SHALL immediately release the existing packet data network connection and establish a new packet data network connection using the overriding binding if that overriding binding is different than the current packet data network connection.	1.2
LwM2M-TRAN-BIND-04	The client SHALL use a client-preferred binding supported by both the server and client if the Preferred Transport resource (Resource 24) in LwM2M server object (Object 1) is not defined.	1.2
LwM2M-TRAN-BIND-05	The client SHALL assume that the server supports UDP binding even if the server does not include UDP (“U”) in the binding resource (Resource 7) of the LwM2M server object (Object 1).	1.2
LwM2M-TRAN-BIND-06	If defined, the PDN Type (Resource 24) in an APN Connection Profile (Object 11) SHALL be preferred when that PDN Type is present in the binding resource (Resource 7) of the LwM2M server object (Object 1).	1.2
LwM2M-TRAN-BIND-07	When configurations related to the bindings change, those new values SHALL be applied to the future communications.	1.2
LwM2M-TRAN-BIND-08	When configurations related to the bindings change, those new values SHALL NOT affect ongoing communications.	1.2
LwM2M-TRAN-BIND-09	Configurations related to bindings SHOULD NOT affect LwM2M client registrations.	1.2

Table 11: Requirements – Optimization of Bootstrap and Registration

6.8 Registration and Discovery

Label	Description	Release
LwM2M-RD-01	Lightweight M2M SHOULD allow to declaring the version of several Objects in a registration payload at once.	1.2
LwM2M-RD-02	Lightweight M2M DISCOVER Operation SHOULD include a way to limit the depth of the returned response.	1.2

Table 12: Requirements – Registration and Discovery Items

6.9 Bootstrap Clarifications

Label	Description	Release
LwM2M-BOOT-PROV-01	The bootstrap process SHALL be able to initialize the Security object (Object 0) and LwM2M server objects (Object 1) from multiple sources including from the factory provisioning, smart card and network.	1.2
LwM2M-BOOT-PROV-02	Access Control on a per source basis SHALL be enabled on the Security object (Object 0) and LwM2M server objects (Object 1).	1.2

Table 13: Requirements – Bootstrap Clarifications

6.10 Version Negotiation

Label	Description	Release
LwM2M-VER-01	The client and server SHALL be able to negotiate the version of the enabler to be used during the bootstrap process.	1.2
LwM2M-VER-02	The client and server SHALL be able to negotiate the version of the enabler to be used for LwM2M server communications.	1.2
LwM2M-VER-03	A LwM2M server SHALL be backward compatible with LwM2M clients using the same major version of the enabler.	1.2
LwM2M-VER-04	A LwM2M client SHALL be backward compatible with LwM2M servers using the same major version of the enabler.	1.2

Table 14: Requirements – Version Negotiation

6.11 Encoding and Standardized Data Models

Label	Description	Release
LwM2M-ENC-01	Lightweight M2M SHOULD provide further optimization of encoding method of single resource value in a standardized encoding.	1.2
LwM2M-ENC-02	Lightweight M2M SHOULD provide further optimization of encoding method of Object or Object Instance values in a standardized encoding.	1.2

Table 15: High-Level Functional Requirements – Encoding

6.12 Core

Existing LwM2M Bootstrap Server Trigger (1/x/9) initiates the entire Bootstrapping process and does not allow for other operations to be performed by the Bootstrap Server.

As an example, the Bootstrap Server Trigger can be used to perform Unbootstrapping.

Label	Description	Release
LwM2M-Core-01	The LWM2M enabler SHALL allow the LWM2M Client to be triggered to contact the LwM2M Bootstrap Server with necessary indicators.	1.2

Table 16: Requirements – Core

6.13 Transports

Support for LwM2M to be used with publish-subscribe transfer protocols.

Label	Description	Release
LwM2M-Trans-01	The LWM2M enabler SHALL support MQTT.	1.2
LwM2M-Trans-02	The LWM2M enabler SHALL support HTTP/1.x	1.2

Table 17: Requirements – Transport

Appendix A. Change History (Informative)

A.1 Approved Version History

Reference	Date	Description
OMA-RD-LightweightM2M-V1_0-20170208-A	08 Feb 2017	Status changed to Approved by TP TP Ref # OMA-TP-2017-0009-INP_LightweightM2M-V1_0_ERP_for_Final_Approval
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

A.2 Draft/Candidate Version 1.2 History

Document Identifier	Date	Sections	Description
Draft Versions OMA-RD-LightweightM2M-V1_2	10 Oct 2018	n/a	Initial Draft
	20 Oct 2018	1, 2, 5, 6,	Incorporates input : • OMA-DM-LightweightM2M-2018-0054R01-CR_Version_N... • OMA-DM-LightweightM2M-2018-0053R01-CR_Bootstrap • OMA-DM-LightweightM2M-2018-0052R01-CR_Transport_Bindings • OMA-DM-LightweightM2M-2018-0050R01-CR_RD_v12_part1 • OMA-DM-LightweightM2M-2018-0049R01-CR_Support_for_new_publish_subscribe_transport • OMA-DM-LightweightM2M-2018-0048-CR_Bootstrap_and_Registration_Optimizations • OMA-DM-LightweightM2M-2018-0047-CR_Registration_to_selected_list_of_servers • OMA-DM-LightweightM2M-2018-0046R01-CR_Transient_attributes OMA-DM-LightweightM2M-2018-0046R01-CR_Transient...
	23 Nov 2018	1,2,5,6	Incorporates input : • OMA-DM-LightweightM2M-2018-0044R02-CR_Registration_Payload Deletes useless text
	30 Nov 2018	1, 2, 3,4, 5, 6 & A	Incorporated input: • OMA-DM-LightweightM2M-2018-0045R01-CR_Trigger Noted Comments/questions ARM Deletes useless Sections
	06 Dec 2018	1, 2, 3,4, 5, 6	Incorporated input: • OMA-DM-LightweightM2M-2018-0061-CR_Profile_identifier • OMA-DM-LightweightM2M-2018-0041R05-CR_Data_encoding • email inputs from DM&SE WG members
	02 Jan 2019	All sections	Changes to fully conform with OMA-DM&SE-2018-0127-INP_RD_Review_Gating_Criteria_observations
Candidate Version OMA-RD-LightweightM2M-V1_2	24 Jan 2019	n/a	Status changed to Candidate by DMSE DMSE ref # OMA-DM_SE-2019-0006-INP_OMA_RD_LightweightM2M_V1_2_for_Approval