

# OMA Lightweight M2M Resource Model

Author: Joaquin Prado ([jprado@omaorg.org](mailto:jprado@omaorg.org)) – OMA Technical Director

## Introduction

This paper gives an introduction to standard developed at the Open Mobile Alliance (OMA), Lightweight Machine to Machine (LWM2M). LWM2M provides several interfaces built on top of Constrained Application Protocol (CoAP) to perform management of a wide range of remote embedded devices and connected appliances in the emerging Internet of Things, to perform remote service enablement and remote application management.

The OMA Lightweight M2M Enabler (LWM2M) is targeted in particular at constrained devices, e.g. devices with low-power microcontrollers (40 MHz), small amounts of Flash (100 KB) and RAM (10KB); battery to last many years over network variable availability [1]. At the same time, LWM2M can also be utilized with more powerful embedded devices that benefit from efficient communication.

LWM2M provides a light and compact secure communication interface along with an efficient data model, which together enables device management and service enablement for M2M devices.

While LWM2M is used for device management operations, its Object Model is being used to provide a resource model for Applications. IPSO Alliance [6] is defining IPSO Objects built on the LWM2M Object Model for smart city/building applications. IPSO provides a reusable Object Model so that any party can define their own objects and either suggest them for standardization to OMA, or just use them on their applications without standardisation.

Furthermore, has been implemented both in C and Java under the umbrella of the Eclipse Foundation [9].

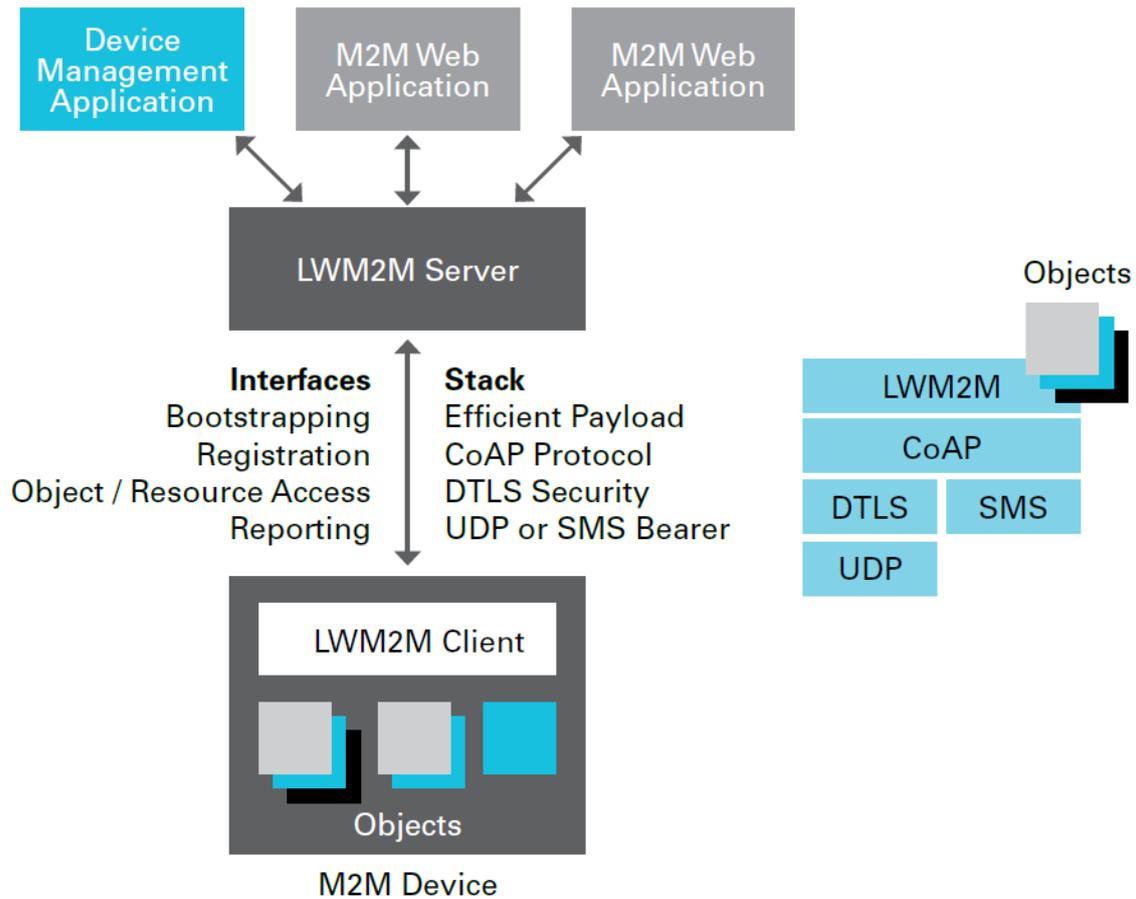
## Lightweight M2M Architecture

The LWM2M Enabler defines the application layer communication protocol between a server and a client. The LWM2M Server is typically located in a private or public data centre and can be hosted by the M2M Service Provider, Network Service Provider or Application Service Provider. The LWM2M Client resides on the device and is typically integrated as a software library or a built-in function of a module or device.

Four logical interfaces are defined between server and client namely:

- 1) Bootstrap
- 2) Device Discovery and Registration
- 3) Device Management and Service Enablement
- 4) Information Reporting [2].

The LWM2M architecture is shown below:



The Lightweight M2M architecture with Lwm2M Server and Lwm2M Client

## CoAP

The LWM2M protocol stack [3] utilizes the IETF Constrained Application Protocol (CoAP) [4] as the underlying transfer protocol over UDP, TCP and SMS bearers. CoAP defines the message header, request/response codes, message options, and retransmission mechanisms. The CoAP protocol was defined by the IETF Constrained RESTful Environment (CoRE) working group. The main goal for the CoRE working group was to keep message overhead small, limit fragmentation, support multicast and create a simplistic protocol for M2M.

CoAP creates an alternative to HTTP for RESTful APIs on resource-constrained devices and supports the basic methods of GET, POST, PUT, DELETE (as with HTTP), which are easily mapped to those of HTTP. Unlike HTTP, CoAP messages are exchanged asynchronously between CoAP end-points over a datagram-oriented transport such as UDP. A subset of response codes is supported for LWM2M response message mapping. Built-in resource discovery is supported using the CoRE Link Format standard. CoAP messages are encoded in a simple binary format, allowing this functionality starting with just a 4-byte overhead. LWM2M defines the UDP Binding with CoAP as mandatory whereas the SMS Binding with CoAP is optional. This means, that LWM2M client-server interaction can happen both via SMS and UDP.

## Security

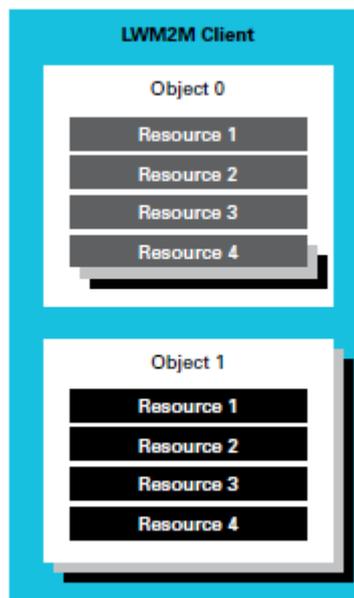
The LWM2M includes state of the art security to secure communications between client and server using Datagram Transport Layer Security (DTLS) [5]. DTLS is used to provide a secure channel between the LWM2M Server and the LWM2M Client for all the messages interchanged. DTLS security modes include both pre-shared key and public key technology to support both very limited and more capable embedded devices. The LWM2M standard defines provisioning and bootstrapping functionality that allows a LWM2M Bootstrap Server to manage the keying, access control and configuration of a device to enrol with a LWM2M Server. The security

identifiers, endpoint identifiers and keys are used uniformly throughout the LWM2M system to provide a complete security lifecycle solution.

## Resource Model

The LWM2M Enabler defines a simple resource model where each piece of information made available by the LWM2M Client is a Resource. The Resources are further logically organized into Objects. The LWM2M Client can have any number of Resources, each of which belongs to an Object.

The figure illustrates the structure and relationship between Resources, Objects, and the LWM2M Client.



The LWM2M data model and the open OMA naming authority registry for Objects provide easily accessible and reusable semantics for both device management and application data for the whole Internet of Things industry [7].

OMA is also developing a Lwm2M editor tool [8] to safely construct these objects/resources models.

## Object/Resource Template [3].

### Object definition:

Name	Object ID	Instances	Mandatory	Object URN
Object Name	16-bit Unsigned Integer	Multiple/Single	Mandatory/Optional	urn:oma:lwm2m:{oma,ext,x}:{Object ID}

- **Name:** specifies the Object name.
- **Object ID:** specifies the Object ID.
- **Instances:** indicates whether this Object supports multiple Object Instances or not. If this field is “Multiple” then the number of Object Instance can be from 0 to many. If this field is “Single” then the number of Object Instance can be from 0 to 1. If the Object field “Mandatory” is “Mandatory” and the Object field “Instances” is “Single” then, the number of Object Instance MUST be 1.

- **Mandatory:** if this field is “Mandatory”, then the LWM2M Client MUST support this Object. If this field is “Optional”, then the LWM2M Client SHOULD support this Object.
- **Object URN:** specifies the Object URN. The format of the Object URN is “urn:oma:lwm2m:{oma,ext,x}:{Object ID}” and {} part means that those values are variable and filled with real value. For example, Object URN of LWM2M Server Object is “urn:oma:lwm2m:oma:1”.

**Resource definition:**

ID	Name	Operations	Instances	Mandatory	Type	Range or Enumeration	Units	Description
0	Resource Name	R (Read), W (Write), E (Execute)	Multiple/ Single	Mandatory/Optional	String, Integer, Float, Boolean, , Opaque, Time, ObjInk none	If any	If any	Description

- **ID:** specifies the Resource ID which is unique within Object.
- **Name:** specifies the Resource name.
- **Operations:** indicates which operations the Resource supports in the “Device Management & Service Enablement” Interface. This field can be set to a combination of R (Read, Observe, Discover, Write Attributes), and W (Write), or can be set to E (Execute); Executable Operation is exclusive regarding the two others (R,W). This field may also have an empty value, which means that this field is not allowed to be accessed via “Device Management & Service Enablement” Interface but allowed to be accessed via “Bootstrap” Interface.
- **Instances:** indicates whether this Resource supports multiple Resource Instances or not. If this field is “Multiple” then the number of Resource Instance can be from 0 to many. If this field is “Single” then the number of Resource Instance can be from 0 to 1. If the Resource field “Mandatory” is “Mandatory” and the field “Instances” of the Resource is “Single” then, the number of Resource Instance MUST be 1. Resource which supports “Execute” operation MUST have “Single” as value of the “Instances” field.
- **Mandatory:** if this field is “Mandatory”, then the LWM2M Server and the LWM2M Client MUST support the Resource. If this field is “Optional”, then the LWM2M Server and the LWM2M Client SHOULD support the Resource.
- **Type:** Data Type indicates the type of Resource value. Resource which supports “Execute” operation MUST have no associated Data Type (none)
- **Range or Enumeration:** this field limits the value of Resource.
- **Units:** specifies the unit of the Resource value.
- **Description:** specifies the Resource description.

In addition to the object and resource definition tables, an object containing Executable Resource(s) is specified in third Table, gathering the definition of the arguments of all the Executable Resources of that Object.

This table provides the properties of arguments

### Executable Resource Arguments Definition

ID	Resource Name	Order	Name	Type	Range or Enum	Unit	Description
id	Resource name	[0:9]	String	LWM2M Data Types	If any	If any	Description

Example of an Executable Resource Arguments Definition Table for an Object having 3 Executable Resource

### Execution Resource Arguments definitions

ID	Resource Name	Order	Name	Type	Range or Enum	Unit	Description
5	Delete	0	-	none	-	-	1 argument EXECUTE /X/0/5 0
7	Update	0	Remove	none	-		2 arguments Ex EXECUTE /X/0/7 0,1='2'
		1	Keep	Integer	[0-2]		
10	Create						

### References

- [1] Gholkar V., "An introduction to IoT protocols", O'Reilly, OSCON, Portland, OR, Jul 2014
- [2] Klas G., Rodermund F., Shelby Z., Akhouri S., Holler J., "OMA Whitepaper LightweightM2M", 2014
- [3] "OMA Lightweight Machine to Machine Protocol v1, 0"  
[http://technical.openmobilealliance.org/Technical/release\\_program/lightweightM2M\\_v1\\_0.aspx](http://technical.openmobilealliance.org/Technical/release_program/lightweightM2M_v1_0.aspx)
- [4] Shelby, Z., Hartke, K., Bormann, C., and B. Frank, "Constrained Application Protocol (CoAP)", draft-ietfcore-coap-18 (final), June 2013.
- [5] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2," RFC 6347, January 2012.
- [6] IPSO Alliance <http://www.ipso-alliance.org/>
- [7] OMA Name Authority, OMNA, for LightweightM2M, <http://technical.openmobilealliance.org/Technical/technical-information/omna/lightweight-m2m-lwm2m-object-registry>
- [8] OMA LightweightM2M Editor tool, <https://github.com/OpenMobileAlliance/OMA-Lwm2m-Public-Review/wiki>
- [9] Eclipse Foundation <http://www.eclipse.org/>