

CoAP functionality expected in a LWM2M system  
draft-jimenez-t2trg-coap-functionality-lwm2m-00

## Abstract

This document provides a strawman summary of information that should be used for the LWM2M specification [LWM2M]. LWM2M is based on CoAP, on top of which it describes certain management interfaces and data models that go beyond the CoAP specifications itself. However LWM2M does not describe all behavior that should be expected from implementations of the CoAP specifications. This document attempts to clarify what should be present in a LWM2M system beyond what is specified in the LWM2M documents. Additionally, this document also adds information about IPSO Objects [IPSO] and their usage with LWM2M as application protocol.

## Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on May 4, 2017.

## Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction . . . . .	2
2. Terminology . . . . .	2
3. Interaction Model . . . . .	3
3.1. Device and Manager configuration. . . . .	3
3.2. Device to Device configuration. . . . .	4
3.3. Device to Application configuration. . . . .	4
4. Data Model . . . . .	5
5. Web Linking . . . . .	6
6. Collaboration . . . . .	6
7. Informative References . . . . .	6
Author's Address . . . . .	8

1. Introduction

The current LWM2M protocol is probably the main Device Management protocol based on CoAP today. It defines the application layer communication protocol between a LWM2M Server and a LWM2M Client, which is located in a LWM2M Device.

2. Terminology

The LWM2M Specification tends to use its own terminology for client, server, etc. In this document, we use the existing terminology from [RFC7252].

For example the use of LWM2M "Client" and "Server" and the roles they play has confused developers that are initiating on the protocol, mainly because a CoAP server runs on the device, just like a LWM2M client does. Moreover, most LWM2M devices will often work both as client and server depending on the interfaces used, it would be good to explore the use of terms like "servients" for devices that regularly support both.

Similarly, the reference to existing drafts of RFCs often can mislead the reader to believe that the full RFC has been implemented. It would be better to state the support to an IETF CoRE WG document when applicable.

For example, the Registration interface in LWM2M is based on the CoAP Resource Directory. However, it is not sufficient to implement just

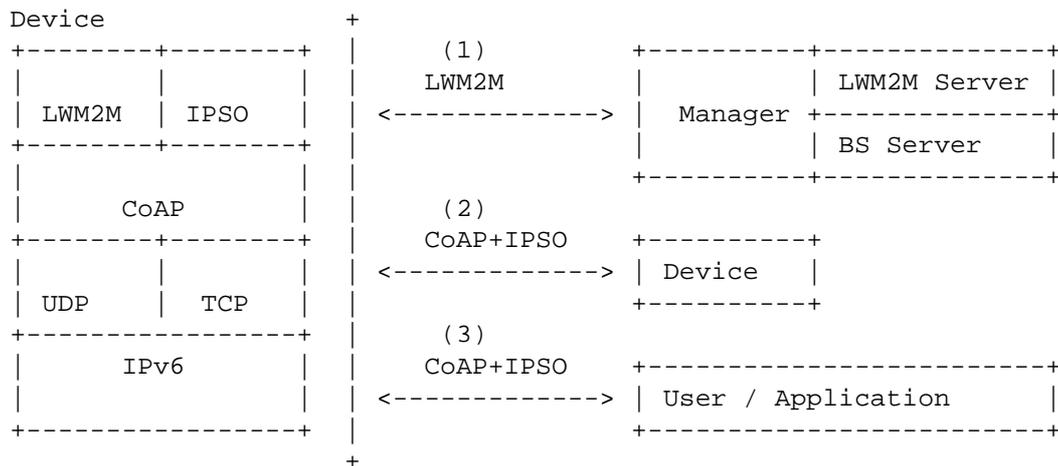
the interface described to obtain the benefits provided by the CoAP Resource Directory.

### 3. Interaction Model

LWM2M has been created with a strong focus on centralizing control and management. Devices set associations with their manager and all traffic is directed to the cloud. All this is fine but in the process some functionalities that could be used locally device to device and device to application have not been explicitly described.

Below we have common configurations that make use of LWM2M.

- o (1) Device and Manager configuration.
- o (2) Device to Device configuration.
- o (3) Device to Application configuration.



#### 3.1. Device and Manager configuration.

This is covered by common LWM2M compliant implementations we have today. However there are upcoming RFCs and drafts that greatly enhance LWM2M with more CoAP features.

For example TCP support is soon going to be added to CoAP. The draft [[I-D.ietf-core-coap-tcp-tls](#)] outlines the changes required to use CoAP over TCP, TLS, and WebSockets transports.

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

For systems in which endpoints work behind a gateway or use LWM2M for managing the gateways, it might be good to implement other types of cryptographic protection than DTLS. For example some of the setups using OSCoAP [[I-D.ietf-core-object-security](#)] allow for "smarter" gateways.

### 3.2. Device to Device configuration.

Beyond what is described in the LWM2M documentation, devices will often talk to each other. Specially in cases when all devices are under the same subnet, this could be pretty common. For example devices could be more resilient if they did not have to contact their manager constantly; in case of lack of internet connectivity the local IoT network would still function. Managers could just set policies on the devices and they would operate more autonomously.

For this setup to take place, LWM2M would use more of the device-to-device functionality of CoAP. A more complete Resource Directory implementation [[I-D.ietf-core-resource-directory](#)] would be needed, either on the LWM2M server in addition to the registration interface or standalone. Devices should be able to perform lookup on that RD and get the series of links to resources elsewhere. They should be able to find new functionality through /.well-known/core. If not, they should be able to use IP multicast as expressed on [[RFC7390](#)].

Needless to say, it is assumed that devices would be running a CoAP Server on them and would support CoAP Observe [[RFC7641](#)], so that devices can subscribe to updates from one another, thus becoming more autonomous.

There are also updates on ACE security framework, that allow for securing the communication between two devices via an Authorization Server [[I-D.ietf-ace-oauth-authz](#)].

The current LWM2M Data Model needs more expressiveness when it comes to data types; More on that in [Section 4](#). Also Web Linking will be dealt at [Section 5](#).

### 3.3. Device to Application configuration.

In some other cases applications would be running on the phone connecting locally to sensors and/or control actuators. A smartphone can access directly a CoAP home sensor using a mutually authenticated 'https' request, provided its home router runs a HTTP to CoAP (HC) proxy and is configured with the appropriate certificate. For this scenario to happen, the GW should implement a HC proxy. It is highly recommended then that they make use of [[I-D.ietf-core-http-mapping](#)] to properly do the URI mapping and specific ABNF queries.

Just like other devices, smartphone applications should be able to discover devices using standard methods, thus they would need access to the RD as well.

#### 4. Data Model

The LWM2M Object Model is specified in [LWM2M]. Other models that build on it like IPSOs or OneM2M have spawned out of it. They normally introduce incremental features. They usually allow for performing any set of operations on a device through a CoAP interface. Resources are exposed as Objects using the same data model used for management.

For example, in the case of a temperature sensor we can access and subscribe to the readings of the device (using [IPSO]).

```
Req: GET /3303/0/5700 Observe_Option=1
Res: 2.05 Content (25 C)
Res (Notify): 2.05 Content (26 C)
```

There has also been much work on different serialization and compression mechanisms that LWM2M could consider adopting. A serialized JSON file like the one below could be greatly compressed (about 46% max, depending on the case) using CBOR representation format [RFC7049] instead.

```
{
  "e": [{
    "bn": "/3303/0",
    "e": [{
      "n": "5700",
      "v": 20.0 }, {
      "n": "5701",
      "v": "c" }, {
      "n": "5603",
      "v": 10 }, {
      "n": "5604",
      "v": 40
    }],
    "bn": "/3302/0",
    "e": [{
      "n": "5500",
      "v": true }, {
      "n": "5501",
      "v": 23
    }
  ]
}
```

LWM2M ResourceIDs at the moment have no specific semantic meaning like ObjectIDs do. Adding a similar registry for ResourceIDs could

be useful. Specially to those using LWM2M for their applications. For example IPSO uses such ResourceIDs to register resources univocally, so that the string `_5701_` consistently represents units.

## 5. Web Linking

One thing that that could be very useful in the future is some form of Web Link resource type. ObjectLinks are not sufficient to represent links between devices or applications. There has been much work on web linking on [RFC6690] that could be used in the LWM2M spec. For example a new Data Type named "Web Link" could be a simple, yet useful addition. Instead of the current `_ObjectID:InstanceID_` expressed now, a full WebLink would be used. That would take advantage of other features like [I-D.ietf-core-links-json] or even newer Object Models.

Other use cases contemplate some form of Object Redirection to help decouple management and applications. LWM2M expects that the management servers will observe resources and collect telemetry on the management server itself. If LWM2M is to be used as application protocol as well as management, it should provide a way for applications or CoAP Clients to observe resources on the devices, together with their required credentials. Such credentials should be stored on the device in some way, maybe a new Object.

## 6. Collaboration

To further develop the relationship between the LWM2M specification and other specifications based on CoAP, it would also be advisable to foster collaboration between organizations developing CoAP-based standard implementations. At the moment there is no forum for inter group communication nor discussion. That should change.

The IETF CoRE WG has quite some people also interested in device management. Communication would be mutually beneficial. Example of that work is on COMI [I-D.ietf-core-yang-cbor] or data model translation.

OMA LWM2M already has benefited from workshops that gather most of the industry, such as [IOTSI] and [IOTSU]. Similarly, specifications can be developed in the IETF with a view to be directly usable in LWM2M.

## 7. Informative References

- [I-D.ietf-ace-oauth-authz]  
Seitz, L., Selander, G., Wahlstroem, E., Erdtman, S., and H. Tschofenig, "Authentication and Authorization for Constrained Environments (ACE)", [draft-ietf-ace-oauth-authz-04](#) (work in progress), October 2016.
- [I-D.ietf-core-coap-tcp-tls]  
Bormann, C., Lemay, S., Tschofenig, H., Hartke, K., Silverajan, B., and B. Raymor, "CoAP (Constrained Application Protocol) over TCP, TLS, and WebSockets", [draft-ietf-core-coap-tcp-tls-05](#) (work in progress), October 2016.
- [I-D.ietf-core-etch]  
Stok, P., Bormann, C., and A. Sehgal, "Patch and Fetch Methods for Constrained Application Protocol (CoAP)", [draft-ietf-core-etch-03](#) (work in progress), October 2016.
- [I-D.ietf-core-http-mapping]  
Castellani, A., Loreto, S., Rahman, A., Fossati, T., and E. Dijk, "Guidelines for HTTP-to-CoAP Mapping Implementations", [draft-ietf-core-http-mapping-16](#) (work in progress), October 2016.
- [I-D.ietf-core-links-json]  
Li, K., Rahman, A., and C. Bormann, "Representing CoRE Formats in JSON and CBOR", [draft-ietf-core-links-json-06](#) (work in progress), July 2016.
- [I-D.ietf-core-object-security]  
Selander, G., Mattsson, J., Palombini, F., and L. Seitz, "Object Security of CoAP (OSCOAP)", [draft-ietf-core-object-security-00](#) (work in progress), October 2016.
- [I-D.ietf-core-resource-directory]  
Shelby, Z., Koster, M., Bormann, C., and P. Stok, "CoRE Resource Directory", [draft-ietf-core-resource-directory-08](#) (work in progress), July 2016.
- [I-D.ietf-core-yang-cbor]  
Veillette, M., Pelov, A., Somaraju, A., Turner, R., and A. Minaburo, "CBOR Encoding of Data Modeled with YANG", [draft-ietf-core-yang-cbor-02](#) (work in progress), July 2016.
- [IOTSI] IAB, "IoT Workshop for Semantic Interoperability (IOTSI)", 2016, <<https://www.iab.org/activities/workshops/iotsi/>>.

- [IOTSU] IAB, "Internet of Things Software Update Workshop (IoTSU)", 2016, <<https://www.iab.org/activities/workshops/iotsu/>>.
- [IPSO] IPSO, "IPSO Object Model", n.d., <<http://ipso-alliance.github.io/pub/>>.
- [LWM2M] OMA, "LWM2M specification", n.d., <<http://technical.openmobilealliance.org/Technical/technical-information/release-program/current-releases/oma-lightweightm2m-v1-0>>.
- [RFC6690] Shelby, Z., "Constrained RESTful Environments (CoRE) Link Format", RFC 6690, DOI 10.17487/RFC6690, August 2012, <<http://www.rfc-editor.org/info/rfc6690>>.
- [RFC7049] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", RFC 7049, DOI 10.17487/RFC7049, October 2013, <<http://www.rfc-editor.org/info/rfc7049>>.
- [RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, DOI 10.17487/RFC7252, June 2014, <<http://www.rfc-editor.org/info/rfc7252>>.
- [RFC7390] Rahman, A., Ed. and E. Dijk, Ed., "Group Communication for the Constrained Application Protocol (CoAP)", RFC 7390, DOI 10.17487/RFC7390, October 2014, <<http://www.rfc-editor.org/info/rfc7390>>.
- [RFC7641] Hartke, K., "Observing Resources in the Constrained Application Protocol (CoAP)", RFC 7641, DOI 10.17487/RFC7641, September 2015, <<http://www.rfc-editor.org/info/rfc7641>>.

Author's Address

Jaime Jimenez  
Ericsson

Phone: +358-442-992-827  
Email: [jaime.jimenez@ericsson.com](mailto:jaime.jimenez@ericsson.com)