ETSI TS 118 120 V2.0.0 (2016-09)

TECHNICAL SPECIFICATION

oneM2M;
WebSocket Protocol Binding
(oneM2M TS-0020 version 2.0.0 Release 2)

Reference

DTS/oneM2M-000020

Keywords

IoT, M2M, protocol

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

The present document can be downloaded from:
http://www.etsi.org/standards-search

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the only prevailing document is the print of the Portable Document Format (PDF) version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at
https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx

If you find errors in the present document, please send your comment to one of the following services:
https://portal.etsi.org/People/CommiteeSupportStaff.aspx

Copyright Notification

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Specification (TS) has been produced by ETSI Partnership Project oneM2M (oneM2M).

1 Scope

The present document specifies the binding of Mca and Mcc primitives onto the WebSocket binding.

It specifies:

- Procedures and message formats for operating and closing of WebSocket connections.
- How request and response primitives are mapped into the payload of the WebSocket protocol.

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at https://docbox.etsi.org/Reference/.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are necessary for the application of the present document.

[1] IETF RFC 6455: "The Web Socket Protocol", December 2011.

[2] ETSI TS 118 101: "oneM2M; Functional Architecture (oneM2M TS-0001).

[3] IETF RFC 7230: "Hypertext Transport Protocol (HTTP/1.1): Message Syntax and Routing", June 2014.

[4] ETSI TS 118 103: "oneM2M; Security Solutions (oneM2M TS-0003)".

[5] ETSI TS 118 104: "oneM2M; Service Layer Core Protocol Specification (oneM2M TS-0004)".

[6] IETF RFC 7692: "Compression Extension for WebSocket", December 2015.

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] oneM2M Drafting Rules.

NOTE: Available at http://www.onem2m.org/images/files/oneM2M-Drafting-Rules.pdf.

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

oneM2M WebSocket Client (WS Client): WebSocket Client associated with an AE or a CSE capable of establishing the WebSocket connections

oneM2M WebSocket Server (WS Server): WebSocket Server associated with a CSE which accepts requests to establish WebSocket connections

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

ADN	Application Dedicated Node
AE	Application Entity
ASN	Application Service Node
CBOR	Concise Binary Object Representation
CSE	Common Services Entity
FQDN	Fully Qualified Domain Name
GUID	Globally Unique Identifier
HTTP	Hypertext Transport Protocol
IANA	Internet Assigned Numbers Authority
IETF	Internet Engineering Task Force
IN-CSE	Infrastructure Node Common Services Entity
IP	Internet Protocol
JSON	JavaScript Object Notation
MN	Middle Node
MN-CSE	Middle Node Common Services Entity
NAT	Network Address Translator
SAEF	Security Association Establishment Framework
TCP	Transmission Control Protocol
TLS	Transport Layer Security
URI	Uniform Resource Identifier
WS	WebSocket
XML	eXtensible Markup Language

4 Conventions

The key words "Shall", "Shall not", "May", "Need not", "Should", "Should not" in the present document are to be interpreted as described in the oneM2M Drafting Rules [i.1].

5 Overview on WebSocket Binding

5.1 Use of WebSocket

This binding makes use of the WebSocket protocol IETF RFC 6455 [1] to transport serialized representations of oneM2M request and response primitives over the Mca or Mcc reference points.

Establishment of a WebSocket connection shall be initiated by a WebSocket client by sending a handshake to a WebSocket server as specified in section 4 of IETF RFC 6455 [1]. Once the WebSocket connection is established, both oneM2M request and response primitives can be exchanged bi-directionally between the two endpoints of the connection. Serialized representations of the request and response primitives shall be mapped in the Payload Data field of the WebSocket base framing protocol, as defined in section 5.2 of IETF RFC 6455 [1].

A WebSocket connection employs either a TCP/IP or a TLS over TCP/IP connection. The underlying TCP and TLS connections are established as the first step of the WebSocket handshake.

5.2 Binding Overview

WebSocket binding may be employed for communication between any two endpoints which can be connected over the Mca, Mcc or Mcc' interface reference points supported by the oneM2M Architecture as shown in figure 6.1-1 of ETSI TS 118 101 [2].

When using the WebSocket protocol, one communication endpoint shall act as the WebSocket server. The WebSocket server listens for inbound handshake messages arriving from any WebSocket client to which a WebSocket connection is not yet established. Whether a communication endpoint takes the role of the client or the server shall depend on the registration relationship between the communicating entities as follows: the registree shall always use a WebSocket client, while the associated registrar shall always use a WebSocket server on the respective reference point.

This implies that ADN and ASN always take the role of a WebSocket client when WebSocket binding is employed. An MN-CSE uses a WebSocket server to communicate with its registrees and a WebSocket client to communicate with its own registrar (which can be another MN-CSE or an IN-CSE).

The IN-CSE provides a WebSocket server functionality to communicate with all its registrees, i.e. within a service provider's domain. On the Mcc' reference points, i.e. for communication between IN-CSEs of different Service Provider domains, the IN-CSE shall provide both WebSocket client and server functionality. This enables any IN-CSE to open a WebSocket connection to any IN-CSE of another Service Provider's domain.

Figure 5.2.-1 shows some applicable example system configuration.

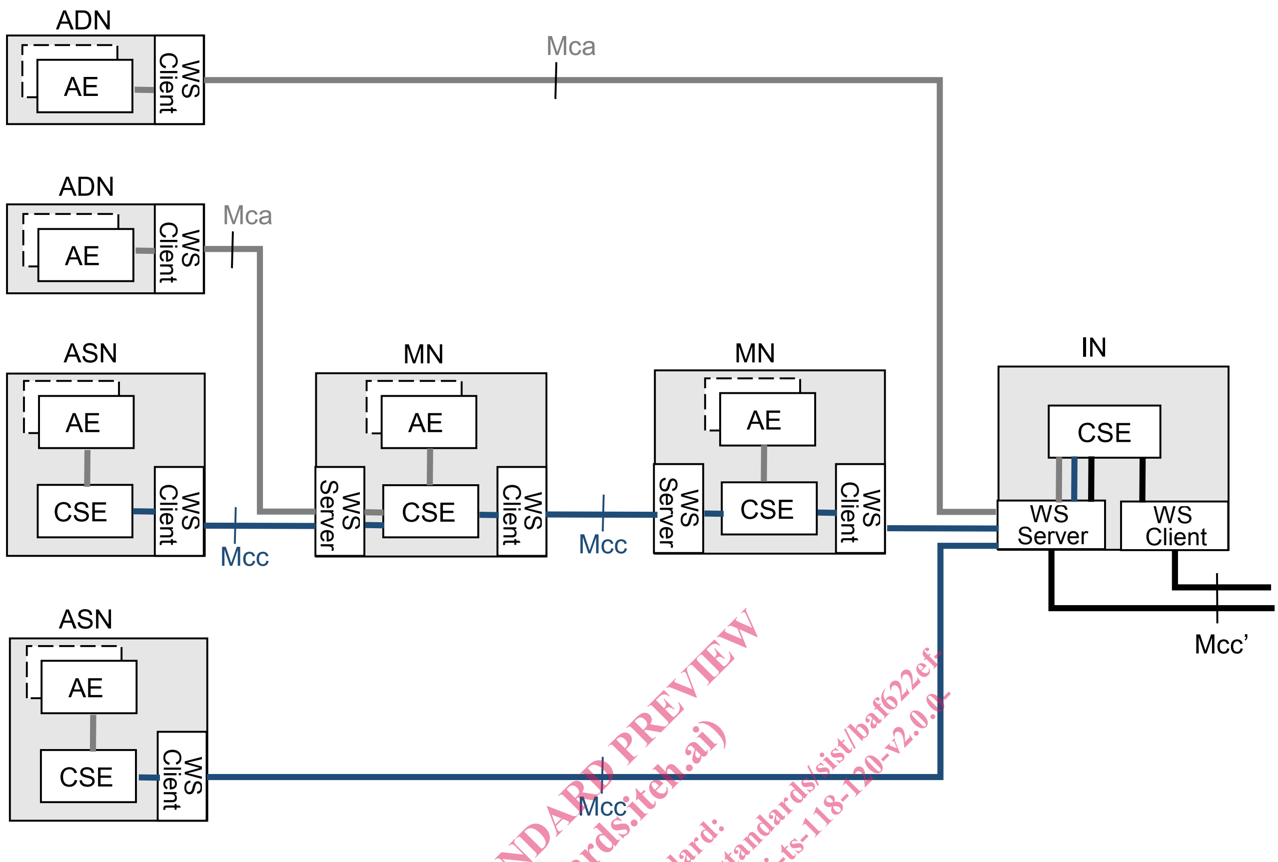

Figure 5.2-1: Example scenarios of WebSocket client and server configurations

There exists a maximum of one WebSocket connection between two nodes. A WebSocket connection is established for the first time when the initial registration procedure of an entity to its registrar is performed. On an established WebSocket connection, request and response primitives can be exchanged in both directions. Any connection may be closed by either the WebSocket client or the server, depending on the communication schedule of either entity. However, the connection can be reopened from the client side only.

If the connection is closed temporarily, it shall be reopened when the next request primitive is sent from the client to the server side, or when the time to become reachable configured at <schedule> resource. If the WebSocket connection with the next-hop entity is not opened, and the WebSocket connection cannot be established due to lack of *pointOfAccess* address for the entity, a sending CSE may enable buffering of primitives which should be sent to the entity until the connection is reopened or their expiration time is reached. See Annex H of ETSI TS 118 104 [5] about buffering of primitives by CMDH functionality.

Figure 5.2-2 shows an example message flow for a scenario where an ADN-AE registers to its registrar MN-CSE using an unsecured TCP connection without proxy and then continues exchanging non-registration request and response primitives.

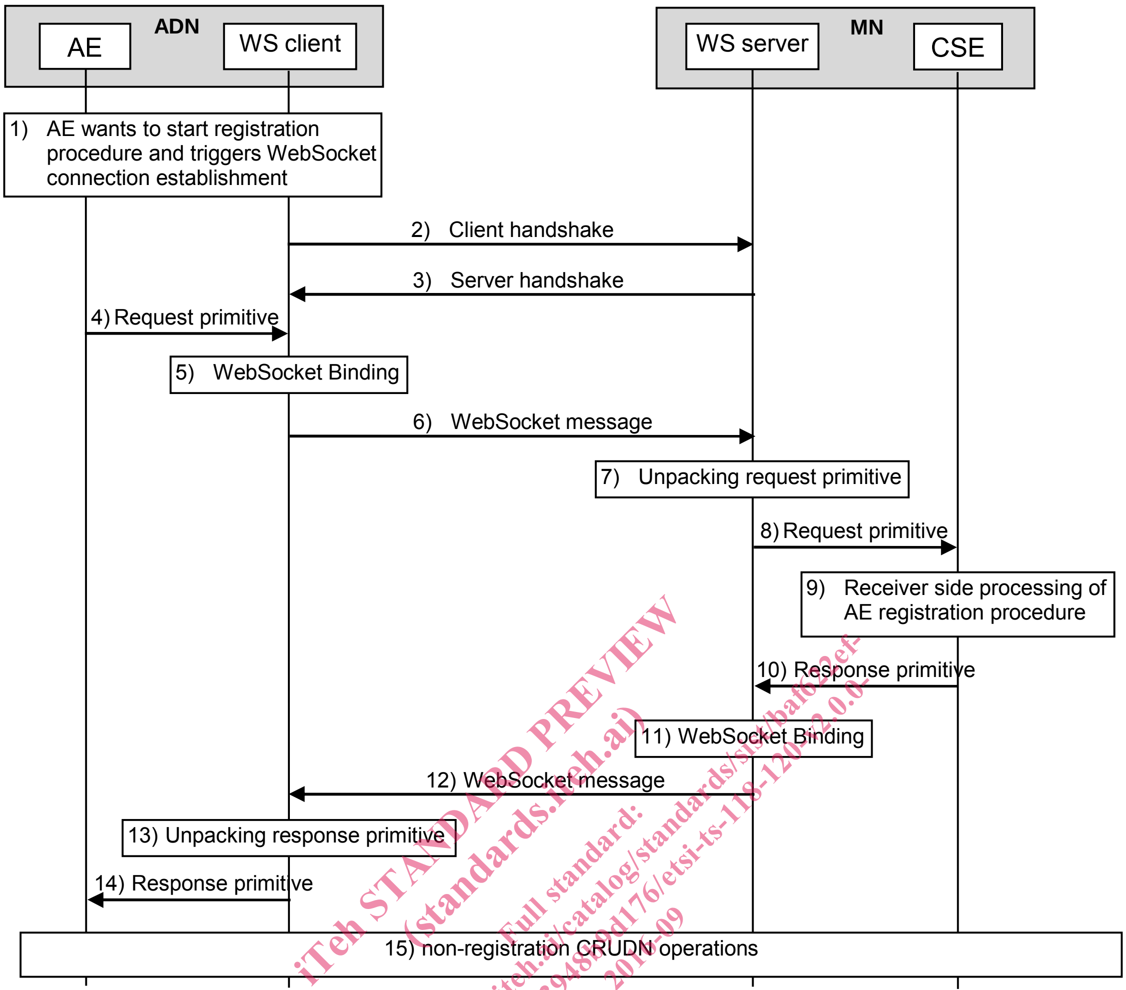

Figure 5.2-2: Example message flow with Websocket binding

1) The ADN-AE wants to register to its registrar MN-CSE. If a WebSocket connection does not exist, it is established by the following steps 2) and 3). It is assumed that the ADN-AE knows the point of access (i.e. WebSocket URI specified in IETF RFC 6455 [1]) under which the registrar CSE can be reached with WebSocket binding.

2) The WebSocket client opens handshake to the server with subprotocol name 'oneM2M-pro-v1.0' following IETF RFC 6455 [1].
If the server can be reached under the WebSocket URI ws://example.net:9000/, the client handshake may look as follows:

```
GET / HTTP/1.1
Host: mncse1234.net:9000
Upgrade: WebSocket
Connection: Upgrade
Sec-WebSocket-Key: ud63env87LQLd4uIV20/oQ==
Sec-WebSocket-Protocol: oneM2M-pro-v1.0
Sec-WebSocket-Version: 13
```

3) The WebSocket server replies with a handshake to the client. In the successful case, the status-line of this HTTP response may look as follow:

```
Request-Version:  HTTP/1.1
Status-Code: 101
Response-Phrase: Switching Protocols
Upgrade: WebSocket
Connection: Upgrade
Sec-WebSocket-Protocol: oneM2M-pro-v1.0
Sec-WebSocket-Accept: FuSSKANnI7C/6/FrPMt70mfBY8E=
```