

CORE
Internet-Draft
Intended status: Informational
Expires: December 19, 2011

G. Moritz, Ed.
University of Rostock
June 17, 2011

SOAP-over-CoAP Binding
draft-moritz-core-soap-over-coap-01

Abstract

The scope of this document is to provide the basis for a lightweight SOAP over CoAP binding. By the binding described in this document, SOAP Web services can also be used in resource constrained networks.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#). This document may not be modified, and derivative works of it may not be created, and it may not be published except as an Internet-Draft.

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 December 19, 2011.

Copyright Notice

Copyright (c) 2011 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	3
1.1.	Motivation	3
1.2.	Requirements Language	3
1.3.	Terminology and Definitions	4
1.4.	Requirements	4
2.	Use of CoAP	5
2.1.	CoAP Media Types	5
3.	Binding Name	5
4.	Transport Layer Binding	5
4.1.	Source Address and Port	6
5.	Addressing	6
5.1.	URI Scheme	6
5.2.	Destination Addressing	6
6.	Message Patterns	6
6.1.	Request response	7
6.2.	Retransmission	8
7.	CoAP Header Options	8
7.1.	Unicast one-way	8
7.2.	Unicast request-response	10
8.	IANA Considerations	11
9.	Security Considerations	11
10.	References	11
10.1.	Normative References	11
10.2.	Informative References	12
Appendix A.	Changelog	12
	Author's Address	13

1. Introduction

The intention of this document is to provide the basic approach for a SOAP-over-CoAP binding. Readers of this document should be basically familiar with the CoAP draft [[I-D.ietf-core-coap](#)], SOAP [[W3C.REC-soap12-part0-20070427](#)], the SOAP HTTP binding [[W3C.REC-soap12-part1-20070427](#)] and the SOAP UDP binding [[SOAP-over-UDP](#)]. Parts of this document are derived from these existing specifications. This document will not provide a comprehensive specification. It may be used as basis for further discussions and to identify required changes in the current CoAP [[I-D.ietf-core-coap](#)] protocol design, which is on the way to become an IETF standard. This binding does not exploit from all features of CoAP, but uses CoAP as an application layer transport mechanism for SOAP envelopes.

1.1. Motivation

The motivation behind this document is based on the initial I-D [[I-D.moritz-6lowapp-dpws-enhancements](#)] and the resulting discussions. By combining SOAP with EXI, message size can be reduced significantly as presented in [[I-D.moritz-6lowapp-dpws-enhancements](#)]. Beside EXI, the herein described binding is the second major enabler towards usage of SOAP Web services in highly resource constrained environments. By implementing this binding, SOAP Web services are not required to use inappropriate mechanisms like TCP handshakes and congestion control implied by the existing SOAP-over-HTTP binding. But in contrast to the existing standard SOAP-over-UDP binding, reliably messaging is guaranteed by the SOAP-over-CoAP binding through CoAP internal mechanisms. In summary the major advantages are:

- o more compact (binary) message format compared to SOAP-over-HTTP binding
- o probably lower message parsing efforts compared to standard HTTP headers
- o avoid inappropriate TCP usage implied by SOAP-over-HTTP binding
- o avoid unreliable nature of the SOAP-over-UDP binding

1.2. Requirements Language

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 [RFC 2119](#) [[RFC2119](#)].

1.3. Terminology and Definitions

Defined below are the basic definitions for the terms used in this specification.

SOAP/CoAP message

A CoAP message containing a SOAP envelope in the CoAP message body

Receiver

The endpoint terminating a SOAP/CoAP message

Sender

The endpoint originating a SOAP/CoAP message

This specification uses the constructs [action], [destination], [message id], [reply endpoint], [address] as defined in WS-Addressing [[W3C.PR-ws-addr-core-20060321](#)].

The SOAP CoAP Binding is optional and SOAP nodes are not required to implement it. A SOAP node that correctly and completely implements the SOAP CoAP Binding may be said to 'conform to the CoAP Binding.'

1.4. Requirements

This specification intends to meet the following requirements:

- o Support a one-way message-exchange pattern (MEP) where a SOAP envelope is carried in a CoAP message from Sender to Receiver only.
- o Support a request-response message-exchange pattern (MEP) where SOAP envelopes are carried in CoAP messages from Sender to Receiver and back from the origin Receiver to the origin Sender.

Even if supported by CoAP, supporting multicast transmissions of SOAP envelopes carried in CoAP messages are out of scope of this version of this document and require further research. For such multicast messages, the existing SOAP-over-UDP binding may be sufficient.

This binding supports SOAP 1.2 [[W3C.REC-soap12-part0-20070427](#)] envelopes. Supporting SOAP 1.1 envelopes is out of scope but might be added in future versions of this document.

This specification relies on WS-Addressing. The SOAP envelope MUST use the mechanisms defined in WS-Addressing [[W3C.PR-ws-addr-core-20060321](#)].

Note: CoAP has no header option which corresponds to HTTP Accept and the existing SOAPAction HTTP header extension field. Thus the web methods feature known from the HTTP binding is not possible. In the current CoAP draft only few information are available how to define own header fields.

2. Use of CoAP

This binding of SOAP to CoAP is intended to make appropriate use of CoAP as an application protocol. For example, successful and failure responses are indicated by the corresponding CoAP response codes (e.g. 2.xx, 4.xx, 5.xx). This binding is not intended to fully exploit the features of CoAP, but rather to use CoAP specifically for the purpose of communicating with other SOAP nodes implementing the same binding.

2.1. CoAP Media Types

Conforming implementations of this binding:

- o MAY be capable of sending and receiving SOAP/CoAP messages serialized using media type 'application/xml'.
- o MAY be capable of sending and receiving SOAP/CoAP messages serialized using media type 'application/exi'.
- o MUST be capable of sending and receiving SOAP/CoAP messages using such media types providing for at least the transfer of SOAP XML Infoset, including 'application/xml' and 'application/exi'.

A SOAP/CoAP message MUST contain the CoAP Content-Type option. This option MUST contain a value which satisfies the three points above.

3. Binding Name

This binding is identified by the URI (see SOAP 1.2 Part 1 [[W3C.REC-soap12-part1-20070427](http://www.w3.org/TR/2007/REC-soap12-part1-20070427)] SOAP Protocol Binding Framework): <http://www.ws4d.org/2011/06/soap/bindings/CoAP/>

Note: The binding name is tentative but required to indicate the corresponding binding e.g. in the WSDL of a service.

4. Transport Layer Binding

The CoAP binding MUST operate over UDP transport layer.

Note: CoAP defines a maximum message size which refers to the IP layer. The existing SOAP-over-UDP binding instead refers only to UDP and defines a general maximum packet size independent of the IP layer. Hence, it might be required to define the CoAP Block mechanism as mandatory as follows: Endpoints which support only messages serialized using the media type 'application/xml' SHOULD implement CoAP Block.

4.1. Source Address and Port

The source address MUST be supplied at the IP packet level and MUST be the IPv4 address (limited to unicast addresses) or IPv6 address (limited to unicast addresses) of the sender; the receiver SHOULD reject IP packets containing a SOAP/CoAP message that have inappropriate values for the source address.

Even though CoAP is intended to be used on the well-known ports as defined in CoAP, the use of CoAP is not limited to these ports. As a result, it is possible to have a dedicated CoAP server for handling SOAP processing on a distinct port.

5. Addressing

5.1. URI Scheme

The SOAP CoAP binding defines a base URI according to the rules in CoAP. I.e., the base URI is the CoAP Request-URI options.

5.2. Destination Addressing

WS-Addressing defines a URI, 'http://www.w3.org/2005/08/addressing/anonymous', that can appear in the [address] property of an endpoint reference. If the [reply endpoint] property of a SOAP message transmitted over CoAP has an [address] property with this value, the UDP source address (and source port) is considered to be the address to which reply messages should be sent.

In absence, the implied value of the [reply endpoint] property for SOAP messages transmitted over CoAP is an endpoint reference with an [address] property whose value is 'http://www.w3.org/2005/08/addressing/anonymous'.

6. Message Patterns

This specification supports the following message patterns:

- o Unicast one-way
- o Unicast request, unicast response

as described in the remainder of this section.

All SOAP/CoAP messages MUST use the POST method of CoAP. In the response, success SHOULD be indicated by the response code '2.04 Changed'. This changes the original meaning of the response code and is only valid for this SOAP-over-CoAP binding.

Note: In the current draft of CoAP-06, POST allows no payload in the response. This will be changed in future versions of the CoAP draft.

Note: The CoAP draft is very strict in how each response code must be interpreted. Since there is no generic code similar to '200 OK' of HTTP, an existing response code must be adapted to conform to this binding. The code might be changed after the adaptations of CoAP to allow payload in POST requests and responses. Further details are required how to map the response codes at a HTTP/CoAP proxy to conform to this binding.

Note: There is no CoAP Action option similar to SOAPAction in HTTP. Hence the web method feature of the HTTP binding cannot be made possible without extensions of CoAP.

Note: CoAP defines Proxy mechanism for caching of responses. Because the CoAP binding defined herein is intended for SOAP transport and not RESTful resource manipulation, caching should be avoided. CoAP defines the Max-Age option to be default a non-zero value. But the POST method is already not cachable. Hence, it is not required by a SOAP/CoAP message to contain the Max-Age option with a value of zero.

6.1. Request response

To match a request with a response, the CoAP Token Option can be used. The CoAP Token Option SHOULD be carried in all request-response SOAP/CoAP messages. WS-Addressing defines the [message id] property to identify messages in time and space and also to match requests with response. In case of using the SOAP/CoAP binding, the [message id] property SHOULD be empty and MUST contain a value in case if the CoAP Token Option is not present.

Note: The intention of this paragraph is to reduce message size. The [message id] property has a typical size of 45 Bytes and cannot be compressed by knowledge based encodings like EXI, because the value of this property is unique for each request/response. The CoAP Token Option may be much more compact by providing the same functionality.

CoAP defines the feature of 'separated responses' (c.f. piggy backed response). The ACK message of a separated response SHOULD NOT carry any payload (e.g. a SOAP Envelope) in the CoAP message body. If the value of the [reply endpoint] is not 'http://www.w3.org/2005/08/addressing/anonymous', the origin Receiver cannot guarantee that the response is intended to be sent to the same entity like the origin Sender and SHOULD include the origin Token Option value in the ACK of the separated response to provide details of the request for the entity behind the [reply endpoint].

6.2. Retransmission

To avoid repeated packet collisions, any retransmission implementation SHOULD observe good practices such as using exponential back-off algorithms and spreading. An implementation SHOULD use the Confirmable (CON) transaction message mechanism described in the CoAP specification to avoid unnecessary retransmissions. For each transmission of such a message, the value of the [message id] property and the CoAP Token Option MUST be the same.

Note: WS-Event Delivery should not use CON due to ACK flood at Event Source. Multicast messages also should use NON messages. Because this specification focuses SOAP in general, it is not sure if such requirements are in scope of this document.

7. CoAP Header Options

In this section, the CoAP header and CoAP header options usage is described in detail.

7.1. Unicast one-way

The unicast one-way message pattern consists one complete CoAP request/response, which again can be separated in different CoAP message exchanges. Only the request carries a SOAP envelope in the message body while the response message body is empty.

The request is formulated according to the table below, but can be extended for application specific needs.

CoAP header option	Value
CoAP Method	POST is the only supported method of this binding.
Request URI	The request URI confirming a CoAP URI and identifying the WS-Addressing [address] endpoint property (network address). If the value of the WS-Addressing [address] endpoint property is 'http://www.w3.org/2005/08/addressing/anonymous' (directly set or implied by an empty [address] property), the CoAP Uri-Path Option and the CoAP Uri-Query Option are empty.
Token Option	Token in accordance to CoAP specification to match the request/response in time and space.
Content-type Option	Media type of CoAP message body.
CoAP message body	SOAP message serialized according to the rules for carrying SOAP messages in the media type given by the Content-type Option.

Table 1: Unicast one-way Request

The response is formulated according to the table below, but can be extended for application specific needs.

CoAP header option	Value
CoAP Status Code	Status Code in accordance to codes defined in CoAP and in this binding. Note: CoAP describes only a subset of HTTP status codes and adds own codes. This requires further alignment.
Token Option	Token in accordance to CoAP specification to match the request/response in time and space.
CoAP message body	MUST NOT be included in the one-way message pattern.

Table 2: Unicast one-way Response

7.2. Unicast request-response

The unicast request-response message pattern consists one complete CoAP request/response, which again can be seperated in different CoAP message exchanges. Both request and response cary a SOAP envelope in the message body.

The request is formulated according to the table below, but can be extended for application specific needs.

CoAP field/header option	Value
CoAP Method	POST is the only supported method of this binding.
Request URI	The request URI confirming a CoAP URI and identifying the WS-Addressing [address] endpoint property (network address). If the value of the WS-Addressing [address] endpoint property is 'http://www.w3.org/2005/08/addressing/anonymous' (directly set or implied by an empty [address] property), the CoAP Uri-Path Option and the CoAP Uri-Query Option are empty.
Content-type	Media type of CoAP message body.
Option	
Token Option	Token in accordance to CoAP specification to match the request/response in time and space.
CoAP message body	SOAP message serialized according to the rules for carrying SOAP messages in the media type given by the Content-type Option.

Table 3: Unicast request-response Request

If the request is a Confirmable CoAP message, the response consists of a CoAP ACK which may carry the response SOAP envelope as data in the CoAP message body. For the response, the origin receiver may also initiate a new CoAP transaction after sending the CoAP ACK to the origin Sender, which can be either also Non-Confirmable or Confirmable. (see separated vs. piggy backed responses in CoAP)

CoAP field/header option	Value
CoAP Status Code	Status Code in accordance to codes defined in CoAP and in this binding. Note: CoAP describes only a subset of HTTP status codes and adds own codes. This requires further alignment.
Content-type Option	Media type of CoAP message body.
Token Option	Token in accordance to CoAP specification to match the request/response in time and space.
CoAP message body	SOAP message serialized according to the rules for carrying SOAP messages in the media type given by the Content-type Option.

Table 4: Unicast request-response Response

8. IANA Considerations

No IANA issues have been identified in this draft.

9. Security Considerations

Will be added in future versions.

10. References

10.1. Normative References

- [I-D.ietf-core-coap]
Shelby, Z., Hartke, K., Bormann, C., and B. Frank,
"Constrained Application Protocol (CoAP)",
[draft-ietf-core-coap-06](#) (work in progress), May 2011.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [W3C.PR-ws-addr-core-20060321]
Gudgin, M., Hadley, M., and T. Rogers, "Web Services Addressing 1.0 - Core", W3C PR PR-ws-addr-core-20060321, March 2006.

[W3C.REC-soap12-part0-20070427]

Mitra, N. and Y. Lafon, "SOAP Version 1.2 Part 0: Primer (Second Edition)", World Wide Web Consortium Recommendation REC-soap12-part0-20070427, April 2007, <<http://www.w3.org/TR/2007/REC-soap12-part0-20070427>>.

10.2. Informative References

[DPWS] Driscoll and Mensch, "OASIS Devices Profile for Web Services (DPWS) Version 1.1", 2009, <<http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01>>.

[I-D.moritz-6lowapp-dpws-enhancements]

Moritz, G., "DPWS for 6LoWPAN", [draft-moritz-6lowapp-dpws-enhancements-01](#) (work in progress), June 2010.

[SOAP-over-UDP]

Jeyaraman, "OASIS SOAP-over-UDP Version 1.1", 2009, <<http://docs.oasis-open.org/ws-dd/soapoverudp/1.1/os/wsdd-soapoverudp-1.1-spec-os.html>>.

[W3C.REC-soap12-part1-20070427]

Nielsen, H., Hadley, M., Karmarkar, A., Lafon, Y., Mendelsohn, N., Moreau, J., and M. Gudgin, "SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)", World Wide Web Consortium Recommendation REC-soap12-part1-20070427, April 2007, <<http://www.w3.org/TR/2007/REC-soap12-part1-20070427>>.

Appendix A. Changelog

Changed from soap-over-coap-00 to soap-over-coap-01:

- o Updated to coap-06
- o Changed behavior of one-way MEP
- o Added response code considerations
- o Updated CoAP header option usage
- o Changed caching considerations
- o Editorial updates

Author's Address

Guido Moritz (editor)
University of Rostock
18051 Rostock,
Germany

Phone: +49 381 498 7269

Email: guido.moritz@uni-rostock.de