Open Geospatial Consortium

Publication Date: 2016-02-03

Approval Date: 2015-02-02

Posted Date: 2015-07-22

Reference number of this document: OGC 15-077r1

Reference URL for this document: http://www.opengis.net/doc/PER/tb11-SOAP

Category: Public Engineering Report

Editor: Ingo Simonis

OGC® Testbed-11 SOAP Interface Engineering Report: Comparison on the Usage of SOAP Across OGC Web service interfaces

Warning

This document is not an OGC Standard. This document is an OGC Public Engineering Report created as a deliverable in an OGC Interoperability Initiative and is not an official position of the OGC membership. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard. Further, any OGC Engineering Report should not be referenced as required or mandatory technology in procurements.

Document type: OGC® Engineering Report
Document subtype: NA
Document stage: Approved for public release
Document language: English

License Agreement

Contents

Page

Tables

Page

Abstract

A number of OGC service interface standards define SOAP bindings. Despite the current hype around REST or RESTful interfaces, SOAP services are still used intensively, in particular in security-critical environments. A number of OGC Web service interfaces support SOAP bindings (see chapter 6). Unfortunately, those bindings are not fully consistent across the suite of OGC service standards. Differences can be found in terms of SOAP versions, used namespaces, error handling, capabilities documentation, or transport of non-XML data; i.e. aspects that should be harmonized by a cross-standard working group.

This document seeks to provide an overview of the current situation and guidance on future SOAP harmonization across all OGC Web services. A number of change requests have been developed during the development process for this document. Though this document provides recommendations in chapter 8, it is highly recommended to either form a new SOAP working group, or preferably to assign the development of SOAP best practices to reduce the risk of missed requirements and architecture arguments to the newly reformed OWS Common SWG. The best practices could then be applied to all OGC service standards that offer SOAP bindings.

Business Value

Having a consistent design of SOAP bindings across all OGC Web service interface standards is highly important. Clients supporting SOAP shall be enabled to interact with all service types.

Keywords

ogcdocs, ogc documens, testbed-11, SOAP, Web services

Testbed-11 SOAP Interface Engineering Report: Comparison on the Usage of SOAP Across OGC Web service interfaces

1 Introduction

1.1 Scope

This OGC Engineering Report provides an overview of the definition of SOAP bindings across the OGC suite of standards. The ER identifies inconsistencies and gaps in defining SOAP interfaces and provides recommendations for how all OGC services could be extended with SOAP bindings in a consistent way. The ER further provides recommendations on how to implement the transfer of binary data for those services transferring either XML Schemas in binary form or other binary data such as images.

1.2 Document contributor contact points

All questions regarding this document should be directed to the editor or the contributors:

Name	Organization
Ingo Simonis	OGC

1.3 Future work

No future work is planned to this document. It is expected that this document may result in changes in other documents.

1.4 Forward

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

2 References

The following documents are referenced in this document. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. For undated references, the latest edition of the normative document referred to applies.

OGC 06-121r3, *OGC® Web Services Common Standard*

OGC 06-121r9, *OGC® Web Services Common Standard*

06-042 OpenGIS Web Map Service WMS Implementation Specification

06-121r3 OGC Web Services Common Specification version 1.1.0 with Corrigendum 1

07-006r1 OpenGIS Catalogue Services Specification V2.0.2

07-057r7 Web Map Tile Service Standard

07-063r1 OpenGIS Web Map Services - Application Profile for EO Products

07-158 Wrapping OGC HTTP-GETPOST Services with SOAP

08-068r2 Web Coverage Processing Service WCPS Language Interface Standard

09-000 OGC Sensor Planning Service Implementation Standard 2.0

09-001 OpenGIS SWE Service Model Implementation Standard 2.0

09-006 OWS-6 DSS Engineering Report - SOAPXML and REST in WMTS

09-025r1 OGC Web Feature Service WFS 2.0 ISOFDIS 19142 Geographic information - Web Feature Service

09-102 DGIWG WMS 1.3 Profile and systems requirements for interoperability for use within a military environment

09-149r1 OGC Web Coverage Service 2.0 Interface Standard - XMLSOAP Protocol Binding Extension

10-059r2 OWS-7 WPS Profiling Engineering Report

12-006 OGC Sensor Observation Service Interface Standard

12-029 Web Processing Service 1.0.0 Best Practices

14-065 Web Processing Service 2.0 Interface Standard

12-111r1 Best Practices for WMS with Time or Elevation dependent data

12-133 Web Services Facade for OGC IP Engineering Report

15-022 Testbed-11 Implementing Common Security Across the OGC Suite of Service Standards ER

3 Terms and definitions

For the purposes of this report, the definitions specified in Clause 4 of the OWS Common Implementation Standard [OGC 06-121r9] shall apply.

4 Conventions

4.1 Abbreviated terms

CSW	Catalog Service Web
KVP	Key Value Pair
SOAP	SOAP
SOS	Sensor Observation Service
WCPS	Web Coverage Processing Service
WCS	Web Coverage Service
WFS	Web Feature Service
WMS	Web Map Service
WMTS	Web Map Tile Service
WPS	Web Processing Service
WSDL	Web Service Description Language

4.2 Highlighted Sections

Sections highlighted in yellow like this section lead to change requests.

5 SOAP

A number of OGC service interface standards define SOAP bindings. Despite the current hype around REST or RESTful interfaces, SOAP services are still used intensively, in particular in security-critical environments. SOAP by itself does not provide any security other than what is inherited from underlying transport protocols. However, a number of extensions exist that allow for secure end-to-end communication (e.g. WS-Security and WS-Policy). For more details on security features see OGC document 15-022. In the past, a number of mapping approaches have been tested (see chapter 7) and service interfaces have been extended to support SOAP bindings (see chapter 6). Unfortunately, those bindings are not fully consistent across the suite of OGC service standards. Differences can be found in terms of SOAP versions, used namespaces, error handling, capabilities documentation, or transport of non-XML data.

This document seeks at providing an overview of the current situation and guidance on future harmonization across all OGC Web services. A number of change requests have been developed along the development process of this document. The following observations have been made and require harmonization across all services. Though this document provides recommendations in chapter 8, it is highly recommended to either form a new SOAP working group, or preferably to assign the development of SOAP best practices to reduce the risk of missed requirements and architecture arguments to the newly reformed OWS Common SWG. The best practices could then be applied to all OGC service standards that offer SOAP bindings.

5.1 SOAP Versions in OGC standards

Different OGC services support different versions of SOAP. Some only support the outdated version SOAP 1.1 (e.g. WFS v2.0), others support the current SOAP version 1.2 exclusively (e.g. CSW 2.0 or WCS 2.0), or even both (e.g. SOS 2.0). As some important aspects changed between SOAP version 1.1 and 1.2 (e.g. SOAPAction headers or SOAP fault values), harmonizing the supported SOAP version would simplify client development in multi-service environments.

5.2 Advertisement of SOAP Support in GetCapabilities responses

The various services make use of different mechanisms to advertise SOAP support in GetCapabilties response documents. The various approaches need to be harmonized.

5.3 WSDL Files

Some of the services provide WSDL files (WCS, SPS), others don’t. The preferred approach should be discussed within the OGC and then implemented by the various SWGs. In order to do so, some of the SWGs would need to be re-chartered prior to doing new work.

5.4 Error Handling

In most OGC standards, error handling makes use of standard SOAP elements such as "soap:Fault". Unfortunately, the various versions use different code values, mostly depending on the applied SOAP version.

Additional requirements or constraints may arise from security-related aspects. For a discussion on error reporting for secured services see [OGC 15-022][1].

5.5 Transmission of non-XML encoded result sets

Some OGC enabled services provide data using different mime types, e.g geotiff for coverages in response to a WCS-GetCoverage request, or XML Schemas in response to a WFS-DesribeFeatureType operation. Currently, the different OGC enabled services use different SOAP features to accomplish this task. Web Coverage Service (WCS) for examples defines that coverages returned in response to a GetCoverage request shall be returned using SOAP with attachments (SwA). Other services, such as Web Feature Service (WFS) define the use of soap:Body inline base64 encodings.

Both mechanisms are widely accepted techniques for handling opaque data in XML-formatted messages, although some disadvantages occur. Encoding with base64 inline in the SOAP payload tends to enlarge the size of the SOAP message. Note that base64 encoding might double the size of the binary data. SOAP with attachments has been superseded by SOAP Message Transmission Optimization Mechanism (MTOM). MTOM describes a mechanism for optimizing the transmission or wire format of a SOAP message by selectively re-encoding portions of the message while still presenting an XML Information Set (Infoset) to the SOAP application. The number of OGC service implementations using MTOM has yet to be determined.

SwA and MTOM are conceptually similar, and both encode binary data as a MIME attachment in a MIME document. MTOM, which uses XML-binary Optimized Packaging (XOP) for the optimized transmission of XML binary data of type xs:base64Binary or xs:hexBinary in SOAP messages, was accepted in January 2005 as a W3C recommendation. If SOAP over HTTP is used (which is the only transport defined across all OGC standards evaluated in the Testbed activity), MIME attachments are used to carry data while at the same time allowing both the sender and the receiver direct access to the XML data in the SOAP message without having to be aware that any MIME artifacts were used to marshal the base64Binary or hexBinary data. The process of optimized binary data transmission includes the five following steps: 1) encode the binary data, 2) remove the binary data from the SOAP envelope, 3) compress the binary data, 4) attach the binary data to the MIME package, and 5) add references to the MIME package in the SOAP envelope. It is important to note that the MTOM specification does not require using XOP binary optimization when transmitting base64binary or hexBinary data. Rather, the specification allows the runtime to choose to do so. This is because in

[1] 15-022 Testbed-11 Implementing Common Security Across the OGC Suite of Service Standards ER

certain cases the runtime may decide that it is more efficient to send the binary data directly in the SOAP Message; an example of such a case is when transporting small amounts of data in which the overhead of conversion and transport consumes more resources than just providing the data as is inline.

6 Overview of OGC Web Service Interfaces

The following table highlights support of SOAP for the various OGC Web service interfaces that have been examined for this engineering report. The columns are defined as follows:

- The ***OWSCommon*** column identifies the referenced version of OWSCommon, which might be either 06-121r3 (deprecated, does not define SOAP) or 06-121r9 (current version, defines SOAP 1.2 and MTOM).
- ***SOAP version*** identifies the referenced version of SOAP, which might be either 1.1 or 1.2 (current version).
- The ***namespace*** column defines if namespaces including prefixes are provided (Y) or not (N).
- ***Error handling*** defines the elements used to report errors as part of a SOAP message. The following structure is used: elementName/value
- The ***Capabilities*** column defines the value of a constraint as provided as part of the capabilities document.
- ***WSDL*** defines the availability (Y) or absence (N) of a WSDL file.
- ***SOAP Action Header*** defines the availability (Y) or absence (N) of a SOAP Action Headers
- ***Non-XML transmission*** identifies the approach defined for transmitting non-XML encoded data. Possible values are inline, SwA (SOAP with attachments), or not defined (-).

Table 1: Service comparison

Service	OWS Common	SOAP Version	Namespace	Error handling	Capabilities	WSDL	SOAP Action Header	Non-XML transmission
WFS 2.0	r3	conflict	Y	soap:faultcode/soap:Server & soap:faultcode/soap:faultstring	SOAPEncoding	Y	N	inline
CSW 2.0.2	r9	1.2	Y	soap:Fault/ soap:Server & soap:Text	SOAP	N	N	SwA
WCS 2.0	r9	1.2	N	soap:Fault/ soap:Server & soap:Text	SOAP	Y	N	SwA
SPS 2.0	r3	1.1 & 1.2	Y (soap11/soap12)	soap:Fault/ soap:Sender & soap:SubCode[2]	SOAP[2]	Y	Y	-
SOS 2.0	r3	1.1 & 1.2	Y	soap:Fault/ soap:Sender & soap:SubCode[2]	SOAP[2]	Y	Y	-
WMS 1.1/2/3	-	not defined	N	-	-	-	-	-
WMTS 1.0.0	r3	1.2	Y	soap:Fault/ soap:Server	SOAP	Y	-	inline

[2] As defined in SWE Common Service model, OGC 09-001

6.1 Catalog Service - CSW

Table 2: CSW SOAP characteristics

Evaluated version	SOAP Version	Namespaces	Prefixes
Catalogue Service v2.0.2 **OGC 07-006r1**	1.2	SOAP envelope: http://www.w3.org/2003/05/soap-envelope; SOAP encoding data types: http://www.w3.org/2003/05/soap-encoding	xmlns:soap=http://www.w3.org/2003/05/soap-envelope soap:encodingStyle="http://www.w3.org/2003/05/soap-encoding">

6.1.1 General Observations

- CSW uses incorrect QNames as part of soap:fault elements. The currently developed CSW 3.0 specification references OGC Common 06-121r9, which includes the same errors (i.e. *soap:server* instead of *soap:sender*).
- There is no information about SOAP action attributes given in the standard. (optional anyway!)
- No WSDL provided

6.1.2 Recommendations on usage

A client may send CSW requests to a compatible catalogue using the body of a SOAP envelope. The client simply encodes the CSW request as the content of the <soap:Body> element in the request message. The CSW then responds by generating a SOAP message where the response to the client's request is the content of the <soap:Body> element.

6.1.3 SOAP Support in Capabilities

CSW servers need to indicate whether or not SOAP encodings of operation requests can be transferred. This indication needs to use an ows:Constraint element named PostEncoding, within the ows:OperationsMetadata section of the Capabilities document. This PostEncoding constraint specifies the formats that can be used with HTTP POST transfer of operation requests. The value SOAP for PostEncoding indicates that SOAP encoded operation requests can be handled. The value XML indicates that (bare) XML encoded operation requests can be handled. If the connect point URL is the same for all SOAP-encoded and base-XML operation requests, the ows:Constraint element is included in the ows:OperationsMetadata element. If the connect point URL is different for SOAP-encoded and base-XML operation requests, this ows:Constraint element is included in each ows:Post element.

6.1.4 Exception handling:

If an exception is encountered while processing a CSW request encoded in a SOAP envelope, the CSW server should generate a SOAP response message where the content of the <soap:Body> element is a <soap:Fault> element. The defined value for the "soap:Value" element is bound to soap:Server, though should be bound to soap:Sender according to Table 4: SOAP Fault Codes as part of SOAP Version 1.2 Part 1: Messaging Framework (Second Edition), W3C Recommendation 27 April 2007.

```
<soap:Body>
  <soap:Fault>
    <soap:Code>
        <soap:Value>soap:Server</soap:Value>
    </soap:Code>
     <soap:Reason>
       <soap:Text>A server exception was encountered.</soap:Text>
     </soap:Reason>
    <soap:Detail>
      <ows:ExceptionReport>
      ...
      </ows:ExceptionReport>
    </soap:Detail>
  </soap:Fault>
</soap:Body>
```

The <soap:Value> element in the <soap:Code> element should have the content "soap:Server" indicating that this is a server exception. The <soap:Text> element in the <soap:Reason> element should have the content "Server exception was encountered.". This fixed string is used since the details of the exception will be specified in the <soap:Detail> element using an <ows:ExceptionReport> element as defined in document [OGC 05-008c1].

6.1.5 WSDL

No information provided.

6.2 Sensor Observation Service – SOS

Table 3: SOS SOAP characteristics

Evaluated version	SOAP Version	Namespaces	Prefixes
Sensor Observation Service v. 2.0 OGC12-006	1.1 & 1.2	soap11: http://schemas.xmlsoap.org/soap/ soap: http://www.w3.org/2003/05/soap-envelope	xmlns:soap11 xmlns:soap

6.2.1 General Observations

The SOS standard does not provide sufficient information on the usage of SOAP. The current document OGC 12-006 references Web Services Common document OGC 06-121r3 (see p. 21 for GetCapabilities responses), which is deprecated. The deprecated document does not define SOAP, but emphasizes the importance to add SOAP in one of its future releases. The current document, OGC 06-121r9, does support SOAP version 1.2 exclusively, which collides with the SOS standard that allows SOAP v1.1 and 1.2.

6.2.2 SOAP Support in Capabilities

Operations metadata refers to OGC 06-121r3, (OGC Web Services Common Standard), which is deprecated and does not define SOAP. No further information is given.

6.2.3 Exception handling:

Exception handling as defined in OGC 09-001 chapter 19 (SWE Service Model). OGC 09-001 defines values for the code, reason, and detail elements of a soap:fault element. There are differences between SOAP 1.1 and SOAP 1.2. In contrast to CSW, SOS makes use of soap:SubCode elements. The soap:Value of those elements is bound to the ows:<ExceptionName>, e.g. ows:OperationNotSupported. The soap:Reason is bound to the exception message defined in OGC 06-121r3. For SOAP 1.2, the following structure is defined:

```
<soap:Body>
  <soap:Fault>
    <soap:Code>
        <soap:Value>soap:Sender</soap:Value>
      <soap:SubCode>
        <soap:Value>ows:OperationNotSupported</soap:Value>
```



```
      </soap:Subcode>
    </soap:Code>
    <soap:Reason>
       <soap:Text>A server exception was encountered.</soap:Text>
     </soap:Reason>
    <soap:Detail>
      <ows:ExceptionReport>
      ...
      </ows:ExceptionReport>
    </soap:Detail>
  </soap:Fault>
</soap:Body>
```

6.2.4 WDSL

WSDL file provided.

6.2.5 SOAPAction Headers

The SOAP 1.1 standard states the following about SOAP action headers:

"The SOAPAction HTTP request header field can be used to indicate the intent of the SOAP HTTP request. The value is a URI identifying the intent. SOAP places no restrictions on the format or specificity of the URI or that it is resolvable. An HTTP client MUST use this header field when issuing a SOAP HTTP Request."

In SOAP 1.2, the SOAPAction header has been replaced with the "action" attribute on the application/soap+xml media type typically provided as part of the HTTP header (SOAP 1.1. uses "text/xml"), thus does not use the SOAP 1.1 header line SOAPAction anymore. The following example illustrates the differences:

Table 4: SOAP headers v1.1 vs. 1.2

SOAP 1.1 message header	SOAP 1.2 message header
Content-Type: text/xml SOAPAction: "http://opengeospatial.org/DescribeSensor"	Content-Type: application/soap+xml; action=http://opengeospatial.org/DescribeSensor

In contrast to other specifications, SOS v2.0 via SWE Sensor Model v2.0 makes use of SOAPAction headers for SOAP 1.1 messages and action attributes for SOAP 1.2 messages respectively. For this reason, a number of action URIs are defined (p.97/98 of OGC 09-001). In addition, OGC09-001 allows for WS-Addressing elements as part of soap:Header elements.

6.3 Sensor Planning Service – SPS

Table 5: SPS SOAP characteristics

Evaluated version	SOAP Version	Namespaces	Prefixes
Sensor Observation Service v. 2.0 OGC12-006	1.1 & 1.2	soap11: http://schemas.xmlsoap.org/soap/ soap12: http://www.w3.org/2003/05/soap-envelope	xmlns:soap11 xmlns:soap12

6.3.1 General Observations

The SPS standard provides detailed information about the usage of SOAP. Though it references OWSCommon 06-121r3, it provides sufficient definitions as part of the document.

6.3.2 SOAP Support in Capabilities

Requirement “42-Advertising Supported Operation Encodings” defines how SOAP support is advertised in the Capabilities document.

6.3.3 Exception handling:

Exception handling as defined in OGC 09-001 chapter 19 (SWE Service Model). OGC 09-001 defines values for the code, reason, and detail elements of a soap:fault element. There are differences between SOAP 1.1 and SOAP 1.2. In contrast to CSW, SWES and therefore SPS makes use of soap:SubCode elements. The soap:Value of those elements is bound to the ows:<ExceptionName>, e.g. ows:OperationNotSupported. The soap:Reason is bound to the exception message defined in OGC 06-121r3. For SOAP 1.2, the following structure is defined:

```
<soap:Body>
  <soap:Fault>
    <soap:Code>
        <soap:Value>soap:Sender</soap:Value>
      <soap:SubCode>
        <soap:Value>ows:OperationNotSupported</soap:Value>
      </soap:Subcode>
```

```
    </soap:Code>
    <soap:Reason>
       <soap:Text>A server exception was encountered.</soap:Text>
     </soap:Reason>
    <soap:Detail>
      <ows:ExceptionReport>
      ...
      </ows:ExceptionReport>
    </soap:Detail>
  </soap:Fault>
</soap:Body>
```

6.3.4 WDSL

A WSDL document is provided.

6.3.5 SOAPAction Headers

The SOAP 1.1 specification states the following about SOAP action headers:

"The SOAPAction HTTP request header field can be used to indicate the intent of the SOAP HTTP request. The value is a URI identifying the intent. SOAP places no restrictions on the format or specificity of the URI or that it is resolvable. An HTTP client MUST use this header field when issuing a SOAP HTTP Request."

In SOAP 1.2, the SOAPAction header has been replaced with the "action" attribute on the application/soap+xml media type typically provided as part of the HTTP header (SOAP 1.1. uses "text/xml"), thus does not use the SOAP 1.1 header line SOAPAction anymore. The following example illustrates the differences:

Table 6: SOAP headers v1.1 vs. 1.2

SOAP 1.1 message header	SOAP 1.2 message header
Content-Type: text/xml SOAPAction: "http://opengeospatial.org/DescribeSensor"	Content-Type: application/soap+xml; action=http://opengeospatial.org/DescribeSensor

In contrast to other OGC standards, SPS v2.0 via SWE Sensor Model v2.0 makes use of SOAPAction headers for SOAP 1.1 messages and action attributes for SOAP 1.2 messages respectively. For this reason, a number of action URIs are defined. In addition, OGC09-001 allows for WS-Addressing elements as part of soap:Header elements.

6.4 Web Coverage Processing Service Language Interface Standard - WCPS

WCPS v1.0 as defined in OGC 08-068r2 does not specify a SOAP binding.

6.5 Web Coverage Service - WCS

Table 7: WCS SOAP characteristics

Evaluated version	SOAP Version	Namespaces	Prefixes
OGC Web Coverage Service 2.0 Interface Standard - XML/SOAP Protocol Binding Extension v1.0.0, OGC 09-149r1[3]	1.2	not defined	not defined

6.5.1 General Observations

The WCS v.2.0 misses some definitions on namespaces for SOAP and uses outdated fault element values.

6.5.2 SOAP Support in Capabilities

Support for SOAP is advertised by including a profile element as part of the ows:ServiceIdentification section in a GetCapabilities response. The value is bound to the following URI: http://www.opengis.net/spec/WCS_protocol-binding_soap/1.0

In addition, for those WCSServiceMetadata elements inherited from OWSServiceMetadata, WCS servers shall specify the HTTP POST request encodings accepted by including an ows:Constraint element, with "PostEncoding" as the value of the name attribute and with a value of "SOAP" to indicate that SOAP encoding is allowed.

6.5.3 SOAP with Attachments

The WCS defines that GetCoverage responses shall make use of the outdated technology SOAP with attachments. Originally developed in SOAP 1.1, SOAP with Attachments is a

[3] This document specifies an extension to the OGC Web Coverage Service (WCS) 2.0 core to allow for client/server communication using SOAP with XML encoding.

"*SOAP feature that represents an abstract model for SOAP attachments. It provides the basis for the creation of SOAP bindings that transmit such attachments along with a SOAP envelope, and provides for reference of those attachments from the envelope. SOAP attachments are described using the notion of a compound document structure consisting of a primary SOAP message part and zero or more related documents parts known as attachments.* [...] *The Attachment Feature document has been superceded by the SOAP Message Transmission Optimization Mechanism document, which describes attachment related features along with some implementation details. The XMLP WG does not intend to do any further work on the Attachment Feature document*" (W3C: http://www.w3.org/TR/soap-af/).

The two requirements defined by the WCS standard contradict each other as they currently stand. Requirement 5 "/req/soap/soap-with-attachments" defines that a GetCoverage SOAP response shall be encoded as "SOAP with Attachments" as defined in [W3C Note 11], but using SOAP 1.2 rather than SOAP 1.1.

Whereas requirement 6 "/req/soap/single-body-element" states that in a GetCoverage response, the SOAP Envelope shall contain one Body element which contains the Coverage as its single element.

It is recommended to clarify the WCS specification on SOAP with attachments. In this context, it should be considered to recommend using SOAP Message Transmission Optimization Mechanism instead of SOAP attachments; see chapter "General Observations" for further details. This approach is required by OwsCommon 06-121r9.

6.5.4 Exception handling

Exception handling in WCS makes use of SOAP 1.2 fault elements. The standard defines that the WCS server shall generate a SOAP response message where the content of the Body element is a Fault element containing an ows:ExceptionReport element [OGC 06-121r9], with the soap:Value element element having the fixed string "soap:server" and the soap:Text having the fixed string "Server exception was encountered." Though the same value is used by CSW, other services (e.g. SOS) make use of the value enumeration defined by the W3C env:faultCodeEnum type, which speaks of "env:Server" instead of "soap:Sender", though allows for additional values if necessary. It is recommended that the SOAP best practices group or preferably the OWS Common SWG discusses this aspect and defines if either the W3C recommendation will be followed or whether an OGC specific extension should be used.

6.5.5 Conformance Classes

Single conformance class covering all SOAP aspects.

6.6 Web Feature Service - WFS

Table 8: WFS SOAP characteristics

Evaluated version	SOAP Version	Namespaces	Prefixes
OpenGIS Web Feature Service 2.0 Interface Standard OGC 09-025r1 and ISO/DIS 19142	1.1	SOAP envelope: http://schemas.xmlsoap.org/soap/envelope/	not defined

6.6.1 General Observations

The WFS standard confuses the reader by making references to various versions of SOAP. On page 206, a SOAP message is defined as an XML document containing a number of elements: "The elements above are declared in the namespace for the SOAP envelope version 1.1", but the SOAP 1.2 namespace is provided: "http://schemas.xmlsoap.org/soap/envelope/". And further: "Services that conform to this International Standard may optionally support SOAP. For maximum interoperability, these services shall support SOAP version 1.1 (see W3C SOAP) for web feature service requests and responses."

This conflicts with SOAP fault messages, as defined on page 208. Here it is stated that SOAP fault messages shall be constructed according to Section 4.4, SOAP Fault, of W3C SOAP:2007, which is SOAP version 1.2. It is recommended that the SOAP versions to be used is clarified. This is important as namespaces and in particular fault elements have been modified from SOAP version 1.1 to 1.2.

6.6.2 SOAP Support in Capabilities

WFS states a requirement to add a <ows:Constraint name="SOAPEncoding"> element to the OperationsMetadata section of the GetCapabilities response to indicate that the SOAP conformance class is implemented by the server instance.

6.6.3 Exception handling

SOAP Fault messages shall be constructed according to Section 4.4, SOAP Fault, of W3C SOAP:2007. This reference links to SOAP version 1.2, which conflicts with the definition of SOAP namespace elements from version 1.1.

Though the WFS error handling follows the error handling defined by other services, it still makes use of SOAP version 1.1 element values "soap:Client" and "soap:Server"

instead of “soap:Sender” and “soap:Receiver” as recommended by SOAP version 1.2. It is recommended to clarify the use of SOAP versions across WFS.

6.6.4 Base64 Encoding of XML Schemas

A WFS provides an instance of XML Schema in response to a DescribeFeatureType operation. As the direct encapsulation of an XML Schema in a SOAP Body may cause several problems, WFS prescribes the usage of a base64 encoding of the schema as part of the soap:Body element. This is a different mechanism compared to WCS, which uses the SOAP attachment approach. to the OGC should consider updating the WFS standard to support SOAP 1.2 and to make use of the SOAP Message Transmission Optimization Mechanism approach, which could be defined as a mechanism to encode XML Schemas. See chapter “General Observations” for further details.

6.6.5 WSDL

WFS v2.0 contains a WSDL definition.

6.7 Web Map Service - WMS

Evaluated version: “OpenGIS® Web Map Server Implementation Specification v.1.3.0”, OGC document 06-042 nor the “DGIWG WMS 1.3 Profile and systems requirements for interoperability for use within a military environment v.0.9.0”, OGC document 09-102, define a SOAP binding.

The “OpenGIS® Web Map Services - Profile for EO Products”, OGC document 07-063r1, provides a single reference for SOAP as part of an abstract test suite “GetMap Minimal Request”, which does not make much sense given that SOAP is not defined for WMS.

“OGC Best Practice for using Web Map Services (WMS) with Time- Dependent or Elevation-Dependent Data”, OGC document -111r1, discusses SOAP as a potential option in cases where HTTP Headers are used.

A “WMS Change Request: Support for WSDL & SOAP”, OGC document 04-050r1, was published April 22, 2005, but never implemented.

6.8 Web Map Tile Service - WMTS

Table 9: WMTS SOAP characteristics

Evaluated version	SOAP Version	Namespaces	Prefixes
Evaluated version: OpenGIS® Web Map Tile Service	1.2	SOAP envelope: http://www.w3.org/20	not defined

Implementation Standard v1.0.0 OGC document 07-057r7		03/05/soap-envelope	

6.8.1 General Observations

WMTS has very detailed definitions for the SOAP binding. Still, WMTS references deprecated version OWS Common OGC 06-121r3, which does not define SOAP but acknowledges it as a future work item. WMTS should be evaluated if updating the reference to the current version OGC 06-121r9 makes sense.

WMTS does not make use of any of the binary data transmission mechanisms recommended by W3C, SOAP with Attachment (SwA) or Message Transmission Optimization Mechanism (MTOM), to transport binary tile data. It is recommended that MTOM be considered as the preferred mechanism.

6.8.2 SOAP Support in Capabilities

A WMTS server needs to declare support for SOAP encoding for each operation by means of the OperationsMetadata section of its ServiceMetadata document. This is the same mechanism as used by WFS v2.0:

```
<ows:Operation name="GetCapabilities">
  <ows:DCP>
    <ows:HTTP>
      <ows:Post xlink:href="http://www.opengis.uab.es/cgi-bin/world/MiraMon5_0.cgi?">
        <ows:Constraint name="PostEncoding">
      <ows:AllowedValues>
        <ows:Value>SOAP</ows:Value>
      </ows:AllowedValues>
    </ows:Constraint>
      </ows:Post>
    </ows:HTTP>
  </ows:DCP>
</ows:Operation>
```

6.8.3 Binary Encoding of Tiles

WMTS should define the response of a successful SOAP-encoded GetTile operation request as an image with the MIME type specified by the Format parameter of the request, wrapped in the SOAP version 1.2 envelope. If the image is binary (such as is the case with image/png and image/jpeg images), the response should be base64 encoded and placed within the BinaryPayload XML element inside a <![CDATA[ ]]> wrapper.

Thus, WMTS does not make use of any of the binary data transmission mechanisms recommended by W3C, SOAP with Attachment (SwA) or Message Transmission Optimization Mechanism (MTOM).

6.8.4 Exception Handling

WMTS makes use of the general practice of embedding a soap:Fault element in the soap:Envelope. Here, the outdated "soap:Server" is defined. It is recommended to change this to the currently recommended "soap:Sender". In addition, the code example provided on page 60 lists "soap:Receiver", though such a message is never sent from a WMTS.

6.8.5 WSDL

WSDL definitions are provided in Annex F of the WMTS standards document.

6.9 Web Processing Interface Standard - WPS

Evaluated version: OGC® WPS 2.0 Interface Standard, OGC document 14-065. According to the WPS document clause "Preface": "WPS 2.0 provides a core conceptual model that may be used to specify a WPS in different architectures such as REST or SOAP." The standard does not provide any further details or guidance.

7 HTTP GET and POST mappings to/from SOAP

In the past, direct mappings with proxies have been evaluated to add SOAP bindings to OGC services without SOAP support. Eventually, those approaches were discontinued for various reasons. First, none of the mapping-based approaches can support the richness of SOAP extensions such as WS-Security. WS-Security provides full support for end-to-end security including important features such as reliable confidentiality and integrity, which is not supported by proxy solutions using off-the shelf products.

Second, it has been proven that the development of SOAP bindings for OGC Web services does not add considerable more work to the development process, and most modern service specifications include SOAP binding definitions.

7.1 OGC 07-158 Mapping architecture

The mapping architecture discussed in OGC discussion paper OGC 07-158 makes use of a server-side proxy and a client-side proxy. The client-side proxy receives service requests via HTTP-GET and -POST and transforms them into a SOAP protocol. The server-side proxy receives these SOAP requests and restores the initial HTTP-GET or -POST request. This transformation allows applying functionalities to HTTP-GET and -POST services, which are defined for SOAP only, such as security or rights management.

To convert original KVP requests into SOAP XML structures, OGC 07-158 suggests an XML Schema to serialize each key-value pair as a property element. Original XML-based POST requests remain unchanged.

To encode responses, OGC 07-158 suggests making use of MTOM (SOAP Message Transmission Optimization Mechanism) in combination with XOP (XML-binary Optimized Packaging) to transmit any non-XML data (e.g. plain text, html, or binary data). It provides a simple XML Schema for this reason that defines a single element “binaryPayload”.

Though OGC 07-158 provides interesting concepts, the developments have been discontinued due to the lack of interest by OGC Member organizations. In addition, more working groups started to develop SOAP bindings for the various services.

7.2 OWS-9: Web Service Façades

In OWS-9, mappings have been done using Web Service Façades, an extensible, open source tool, which supports translations between different protocols for a specific web service. For the OWS-9 testbed, a façade was implemented that set up a translation between POST and SOAP services for a Web Feature Service. However, it could be configured to support translations between multiple protocols, such as REST, SOAP, KVP, JSON, as well as supporting multiple web services. The approach is documented in OGC document 12-133, OGC® Web Services Facade for OGC IP Engineering Report.

The approach has been discontinued, as it became obvious that considerably more configuration work would be necessary to establish a solid KVP to SOAP mapping. This is due to the way OGC has implemented the standard approach to SOAP by embedding standard OGC XML in a SOAP Envelope. Unfortunately KVP to SOAP mappings cannot be generated generically as the SOAP mapper needs to map the parsed KVP key/value pairs into XML. This is non-trivial and code needs to be developed in order to create this proxy mapping. In addition, one of the particular strengths of SOAP, its extensibility with other standards offering confidentiality and integrity protection from the creation of the message to it's consumption (WS-Security), can hardly be leveraged using the façade approach.

8 Recommendations

8.1 SOAP Capabilities

Recommendation: A common mechanism shall be defined that will be used by any OGC service to declare support for SOAP encoding for each operation by means of the OperationsMetadata section of its ServiceMetadata document. One option would be to use the mechanism defined by WMTS, as illustrated below.

```
 <ows:Operation name="GetCapabilities">
     <ows:DCP>
        <ows:HTTP>
           <ows:Post xlink:href="http://www.opengis.uab.es/cgi-
bin/world/MiraMon5_0.cgi?">
              <ows:Constraint name="PostEncoding">
       <ows:AllowedValues>
          <ows:Value>SOAP</ows:Value>
       </ows:AllowedValues>
    </ows:Constraint>
           </ows:Post>
        </ows:HTTP>
     </ows:DCP>
  </ows:Operation>
<ows:Operation name="GetTile">
    <ows:DCP>
        <ows:HTTP>
           <ows:Post xlink:href="http://www.opengis.uab.es/cgi-
bin/world/MiraMon5_0.cgi?">
              <ows:Constraint name="PostEncoding">
                 <ows:AllowedValues>
                    <ows:Value>SOAP</ows:Value>
                 </ows:AllowedValues>
              </ows:Constraint>
           </ows:Post>
        </ows:HTTP>
    </ows:DCP>
</ows:Operation>
```

8.2 Binary Encoding and Encoding of non-XML data

The various service make use of different mechanism to transport binary or XML Schema data, e.g. SOAP with attachment, inline base64 encoding, or Message Transmission Optimization Mechanism (MTOMin combination with XML-binary Optimized Packaging (XOP). Each mechanism has advantages and disadvantages. The working group on SOAP shall identify appropriate solutions here.

8.3 Error Handling

Recommendation: All errors SHALL be reported following the SOAP 1.2 recommendation of embedding a soap:Fault element in the soap:Body of an soap:Envelop, see example below. The value of the soap:Code/soap:Value shall be bound

to “soap:Sender”. Optionally, the type of error message may be indicated as part of the soap:SubCode element.

The detailed error message shall be part of the ows:ExceptionReport and follow the rules provided in OWSCommon, OGC06-121r9.

```
<soap:Body>
  <soap:Fault>
    <soap:Code>
        <soap:Value>soap:Sender</soap:Value>
      <soap:SubCode>
        <soap:Value>ows:OperationNotSupported</soap:Value>
      </soap:Subcode>
    </soap:Code>
    <soap:Reason>
       <soap:Text>A server exception was encountered.</soap:Text>
     </soap:Reason>
    <soap:Detail>
      <ows:ExceptionReport>
      ...
      </ows:ExceptionReport>
    </soap:Detail>
  </soap:Fault>
</soap:Body>
```

8.4 WSDL Support

A mechanism to define WSDL files shall be discussed by the working group on SOAP.

8.5 Conformance Classes

Recommendation: The OWS Common SWG shall develop recommendations on how to organize support for SOAP in conformance classes.

8.6 SOAP Action Headers

Recommendation: The OWS Common SWG shall develop recommendations on the usage of the “action" attribute on the application/soap+xml media type typically provided as part of the HTTP header.

Revision history

Date	Release	Editor	Primary clauses modified	Description
Jun 15, 2015	1.0	I. Simonis	all	initial version
Jul 22, 2015	1.1	I. Simonis	all	comments integrated, SPS added
Sept 3, 2015		Carl Reed	Various	Preparation for publication

Bibliography

[1] SOAP Version 1.2 Part 1: Messaging Framework (Second Edition). W3C Recommendation 27 April 2007. http://www.w3.org/TR/soap-part1/