 

ISBM 2.0

Information Service Bus Model

OpenO&M Standard

YYYY-MM-DD

Editors

**MIMOSA** Matt Selway, University of South Australia  
 Karamjit Kaur, University of South Australia

**ISA** Dennis Brandl, BR&L Consulting  
 Douglas Brandl, BR&L Consulting

Status

This specification was last revised and approved by OpenO&M on the above date. Check the Latest Version for possible later revisions of this document.

This document is considered stable and may be used as reference material or cited as a normative reference from another document.

The latest stable version of the editor's draft of this specification is always available on the [MIMOSA ISBM Git repository](https://github.com/mimosa-org/ws-isbm) [https://github.com/mimosa-org/ws-isbm].

If you wish to make comments regarding this specification in a manner that is tracked by OpenO&M, please submit them via [the public bug database](https://github.com/mimosa-org/ws-isbm/issues) [https://github.com/mimosa-org/ws-isbm/issues]. You can alternatively [contact MIMOSA directly](http://www.mimosa.org/contact) [http://www.mimosa.org/contact] and arrangements will be made to transpose appropriate remarks to the public bug database. All feedback is welcome.

Latest Version

The latest version of this specification can always be found at: [http://www.openoandm.org/isbm](http://www.openoandm.org/ws-isbm)

Notices

Copyright MIMOSA 2019. All Rights Reserved.

All capitalized terms in the following text have the meanings assigned to them in the MIMOSA Intellectual Property Rights Policy (the "MIMOSA IPR Policy"). The full Policy may be found at the [MIMOSA website](http://www.mimosa.org/policy-charters/mimosa-intellectual-property-rights-policy/) [http://www.mimosa.org/policy-charters/mimosa-intellectual-property-rights-policy/].

The limited permissions granted above are perpetual and will not be revoked by MIMOSA or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and MIMOSA DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

MIMOSA requests that any MIMOSA Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this MIMOSA Final Deliverable, to notify MIMOSA TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the MIMOSA Technical Committee that produced this deliverable.

MIMOSA invites any party to contact the MIMOSA TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this MIMOSA Final Deliverable by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the MIMOSA Technical Committee that produced this MIMOSA Final Deliverable. MIMOSA may include such claims on its website but disclaims any obligation to do so.

MIMOSA takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this MIMOSA Final Deliverable or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on MIMOSA's procedures with respect to rights in any document or deliverable produced by a MIMOSA Technical Committee can be found on the MIMOSA website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this MIMOSA Final Deliverable, can be obtained from the MIMOSA TC Administrator. MIMOSA makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

The OpenO&M ISBM is released under the [MIMOSA License Agreement](http://www.mimosa.org/policy-charters/mimosa-license-agreement/) [http://www.mimosa.org/policy-charters/mimosa-license-agreement/].

Contents

[Foreword 1](#_Toc26878662)

[Introduction 2](#_Toc26878663)

[1 Scope 3](#_Toc26878664)

[2 Normative References 3](#_Toc26878665)

[3 Conformance to ISA-95.00.06 3](#_Toc26878666)

[4 Terms, Definitions, and Conventions 3](#_Toc26878667)

[4.1 Terms 3](#_Toc26878668)

[4.2 Notational Conventions 5](#_Toc26878669)

[4.3 XML Namespaces 5](#_Toc26878670)

[5 Service Requirements 5](#_Toc26878671)

[5.1 Message Content Format 5](#_Toc26878672)

[5.1.1 SOAP Interface Requirements 5](#_Toc26878673)

[5.1.2 REST Interface Requirements 6](#_Toc26878674)

[5.2 Security 8](#_Toc26878675)

[5.2.1 SOAP Interface Requirements 9](#_Toc26878676)

[5.2.2 REST Interface Requirements 9](#_Toc26878677)

[5.3 Error Handling 9](#_Toc26878678)

[5.3.1 Parameter Faults 9](#_Toc26878679)

[5.3.2 Invalid Notification URL 10](#_Toc26878680)

[5.4 Content-Based Filtering 10](#_Toc26878681)

[5.5 Message Expiry 11](#_Toc26878682)

[5.6 Feature Set Declaration 11](#_Toc26878683)

[6 Service Definitions 11](#_Toc26878684)

[6.1 Channel Management Service 11](#_Toc26878685)

[6.1.1 Create Channel 12](#_Toc26878686)

[6.1.2 Add Security Tokens 13](#_Toc26878687)

[6.1.3 Remove Security Tokens 14](#_Toc26878688)

[6.1.4 Delete Channel 16](#_Toc26878689)

[6.1.5 Get Channel 16](#_Toc26878690)

[6.1.6 Get Channels 16](#_Toc26878691)

[6.2 Notification Service 17](#_Toc26878692)

[6.2.1 Notify Listener 17](#_Toc26878693)

[6.3 Provider Publication Service 19](#_Toc26878694)

[6.3.1 Open Publication Session 19](#_Toc26878695)

[6.3.2 Post Publication 20](#_Toc26878696)

[6.3.3 Expire Publication 21](#_Toc26878697)

[6.3.4 Close Publication Session 23](#_Toc26878698)

[6.4 Consumer Publication Service 24](#_Toc26878699)

[6.4.1 Open Subscription Session 24](#_Toc26878700)

[6.4.2 Read Publication 26](#_Toc26878701)

[6.4.3 Remove Publication 27](#_Toc26878702)

[6.4.4 Close Subscription Session 28](#_Toc26878703)

[6.5 Provider Request Service 29](#_Toc26878704)

[6.5.1 Open Provider Request Session 29](#_Toc26878705)

[6.5.2 Read Request 31](#_Toc26878706)

[6.5.3 Remove Request 33](#_Toc26878707)

[6.5.4 Post Response 34](#_Toc26878708)

[6.5.5 Close Provider Request Session 35](#_Toc26878709)

[6.6 Consumer Request Service 36](#_Toc26878710)

[6.6.1 Open Consumer Request Session 36](#_Toc26878711)

[6.6.2 Post Request 38](#_Toc26878712)

[6.6.3 Expire Request 39](#_Toc26878713)

[6.6.4 Read Response 40](#_Toc26878714)

[6.6.5 Remove Response 42](#_Toc26878715)

[6.6.6 Close Consumer Request Session 43](#_Toc26878716)

[7 XML Data Structures 44](#_Toc26878717)

[7.1 Channel 44](#_Toc26878718)

[7.2 ChannelType 44](#_Toc26878719)

[7.3 MessageContent 45](#_Toc26878720)

[7.4 Namespace 45](#_Toc26878721)

[7.5 PublicationMessage 46](#_Toc26878722)

[7.6 RequestMessage 46](#_Toc26878723)

[7.7 ResponseMessage 46](#_Toc26878724)

[7.8 SecurityToken 46](#_Toc26878725)

[8 JSON Data Structures 46](#_Toc26878726)

[8.1 Channel 46](#_Toc26878727)

[8.2 ChannelType 47](#_Toc26878728)

[*8.3* Message 47](#_Toc26878729)

[8.4 MessageContent 48](#_Toc26878730)

[*8.5* MessageType 48](#_Toc26878731)

[8.6 SecurityToken 48](#_Toc26878732)

[9 Security Architecture 49](#_Toc26878733)

[9.1 Security Level 1 – None 49](#_Toc26878734)

[9.1.1 Usage Scenarios 49](#_Toc26878735)

[9.2 Security Level 2 – Core Security 49](#_Toc26878736)

[9.2.1 Usage Scenarios 50](#_Toc26878737)

[9.3 Security Level 3 – Inter-Enterprise Security 50](#_Toc26878738)

[9.3.1 Usage Scenarios 50](#_Toc26878739)

[9.4 Security Level 4 – Defense 50](#_Toc26878740)

[9.4.1 Usage Scenarios 51](#_Toc26878741)

[9.5 Security Level Matrix 51](#_Toc26878742)

[10 Conformance 51](#_Toc26878743)

[Annex A. Specification Files 53](#_Toc26878744)

[A.1 OpenAPI Definitions 53](#_Toc26878745)

[A.2 WSDLs 53](#_Toc26878746)

[A.3 Packaged Specification 53](#_Toc26878747)

[Annex B. Example HTTP Flows 54](#_Toc26878748)

[B.1 Channel Management Example 54](#_Toc26878749)

[B.1.1 CreateChannel 54](#_Toc26878750)

[B.1.2 AddSecurityToken 55](#_Toc26878751)

[B.1.3 RemoveSecurityToken 56](#_Toc26878752)

[B.1.4 GetChannel 57](#_Toc26878753)

[B.1.5 GetChannels 58](#_Toc26878754)

[B.1.6 DeleteChannel 59](#_Toc26878755)

[B.2 Publish-Subscribe Example 60](#_Toc26878756)

[B.2.1 OpenSubscriptionSession 60](#_Toc26878757)

[B.2.2 OpenPublicationSession 61](#_Toc26878758)

[B.2.3 PostPublication 62](#_Toc26878759)

[B.2.4 NotifyListener 63](#_Toc26878760)

[B.2.5 ReadPublication 63](#_Toc26878761)

[B.2.6 ExpirePublication 64](#_Toc26878762)

[B.2.7 RemovePublication 65](#_Toc26878763)

[B.2.8 ClosePublicationSession 66](#_Toc26878764)

[B.2.9 CloseSubscriptionSession 67](#_Toc26878765)

[B.3 Request-Response Example 68](#_Toc26878766)

[B.3.1 OpenProviderRequestSession 68](#_Toc26878767)

[B.3.2 OpenConsumerRequestSession 69](#_Toc26878768)

[B.3.3 PostRequest 70](#_Toc26878769)

[B.3.4 NotifyListener 71](#_Toc26878770)

[B.3.5 ReadRequest 72](#_Toc26878771)

[B.3.6 RemoveRequest 73](#_Toc26878772)

[B.3.7 PostResponse 73](#_Toc26878773)

[B.3.8 NotifyListener 74](#_Toc26878774)

[B.3.9 ReadResponse 75](#_Toc26878775)

[B.3.10 RemoveResponse 76](#_Toc26878776)

[B.3.11 CloseConsumerRequestSession 76](#_Toc26878777)

[B.3.12 CloseProviderRequestSession 77](#_Toc26878778)

[Acknowledgements 79](#_Toc26878779)

[Bibliography 80](#_Toc26878780)

Foreword

This document defines a SOAP Web Service implementation of the ISA 95.00.06 Messaging Service Model as well as describing a plain HTTP/JSON REST interface defined by the Open O&M Joint Working Group.

OpenO&M is an initiative of multiple industry standards organizations to provide a harmonized set of standards for the exchange of Operations & Maintenance (O&M) data and associated context. OpenO&M is an open, collaborative, effort composed of diverse groups of relevant organizations and subject matter experts. The members of OpenO&M initiative include ISA, MESA, MIMOSA, OAGi, and the OPC Foundation.

The OpenO&M Initiative involves multiple industry standards organizations collaborating to provide a harmonized set of information standards for the exchange of Operations & Maintenance data. MIMOSA, ISA and OPC Foundation information standards are applicable to multiple industries. Participating organizations work together to cross-reference their related standards and to collaborate on the content.

* MIMOSA provides asset management related information standards
* ISA provides industrial automation standards
* OPC Foundation provides data acquisition and transport standards

Introduction

This specification defines a SOAP Web Service and a HTTP/JSON REST implementation of the ISA-95.00.06 Messaging Service Model.

It defines a minimal interface subset to message exchange middleware using standard Web Service and REST interfaces. Publish-subscribe and request-response messaging patterns are supported through a consistent and unified model. Message routing is conducted through shared channels and topics, and optionally, XPath/JSONPath filtering for granular content-based filtering. An asynchronous Web Service callback or an asynchronous callback REST service is also provided to clients for notification of received messages. Token-based security for channels is specified to support multiple authorization models, from basic credential exchange to federated identity providers.

The benefit of this implementation specification is that it allows applications to expose a single, standardized interface instead of a custom-built interface for every version and format of message exchange systems. It also allows applications to select REST or Web Services based on the application requirements. The goal is to further interoperability in application to application communications.

ISBM 2.0—Information Service Bus Model

# Scope

This is an implementation specification of a set of Web Services for the messaging services described in ISA-95.00.06 Messaging Service Model. The Web Services are defined for both SOAP (1.1 and 1.2) and as a RESTful API.

# Normative References

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

*ANSI/ISA-95.00.06-2014, Enterprise-Control System Integration – Part 6: Messaging Service Model*

# Terms, Definitions, and Conventions

## Terms

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

* ISO Online browsing platform: available at <http://www.iso.org/obp>
* IEC Electropedia: available at <http://www.electropedia.org/>

ChannelDescription

The description of a channel.

[SOURCE: ISA-95.00.06-2014, 3.1.1]

ChannelFault

The error returned when a Channel URI is invalid, or the application does not have the appropriate Security Token to access the channel.

ChannelType

Indicates whether the channel is for publications or requests/responses. Defined ChannelTypes are Publication and Request.

ChannelURI

The primary identifier for a channel.

Expression

The expression that is used to filter message content.

Expiry

The duration until the expiration of the message.

interface

ListenerURL

The URL endpoint, reachable by the ISBM Service Provider, which hosts a ISBM Notification Service.

NOTE 1 Used to indicate when a new message is available for a session.

MessageContent

The message content MAY be XML, JSON or any other information representation.

MessageID

An identifier generated by the ISBM Service Provider upon creation of a message.

NamespaceFault

The error returned when duplicate namespace prefixes occur in the Namespace parameters. Namespaces prefixes MUST be unique.

NamespaceName

The namespace name used for an XPath/JSONPath filter expression.

NamespacePrefix

The namespace prefix used for an XPath/JSONPath filter expression.

OperationFault

The error returned when attempting to open a Session on a Channel of the wrong ChannelType. The channel type MUST be of Publication type or Request type

resource

SecurityToken

A token that can be assigned to a channel to control authorization.

SecurityTokenFault

The error returned when an invalid Security token is used.

SessionFault

The error returned when attempting to access a Session of the wrong SessionType.

SessionID

An identifier generated by the ISBM Service Provider upon creation of a session. Identifiers SHOULD be made non-obvious and not easily guessable.

SessionType

Indicates which service the session is for. Defined SessionTypes are Publication Provider, Publication Consumer, Request Provider, and Request Consumer.

Topic

The topic name given to a message.

Web Service

NOTE Only applies when this capitalization is used.

[SOURCE: https://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/#webservice]

## Notational Conventions

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](http://www.ietf.org/rfc/rfc2119.txt).

This specification uses the following syntax to define conceptual structures and schema elements:

Element Name (Type) [Cardinality]

The namespaces for Types are defined in [the following section](http://www.openoandm.org/ws-isbm/1.0/ws-isbm.html#namespaces). For example, the Topic element defined as an XML Schema string with one to many cardinality would be defined as: Topic (xs:string) [1..\*].

## XML Namespaces

The following namespaces are used in this document:

|  |  |
| --- | --- |
| Prefix | Namespace |
| xs | http://www.w3.org/2001/XMLSchema |
| xsi | http://www.w3.org/2001/XMLSchema-instance |
| isbm | [http://www.openoandm.org/isbm/](http://www.openoandm.org/ws-isbm/) |
| isbm-rest | [http://www.openoandm.org/isbm/rest/](http://www.openoandm.org/ws-isbm/rest/) |
| json | Used for the differentiation of basic data types |

# Service Requirements

The following items define shared requirements that are applicable across the various services defined in [Service Definitions](#service-definitions). These requirements supplement the service requirements specified by ISA-95.00.06 but are contextualized for SOAP Web Services and REST interfaces.

## Message Content Format

Messages are exchanged using a format appropriate to the service interface, for example, XML for the SOAP Interface and JSON for the REST interface. To support transparency of the interface and environments with mixed Web Service implementations, the Message Content MAY be a content type different to the type used to represent the message. For example, JSON Message Content in an XML message over the SOAP interface.

All ISBM service providers MUST support JSON and XML content types as Message Content.

The ISBM service providers MAY support other content types as Message Content.

If a ISBM service provider implements both the SOAP interface and REST interface, messages received via one interface MUST be able to be forwarded via the other interface.

NOTE While the Message Topic describes the specific format and schema of the Message Content, the message schemas themselves, see below, describe the MIME Type to ensure accurate processing of the Message Content.

### SOAP Interface Requirements

The XML Schemas for the SOAP interface are defined such that they allow the exchange of XML, JSON, and other Message Content types within the XML SOAP messages. The [XML schema for Message Content](#_MessageContent) defined for the SOAP interface makes use of type inheritance to support the different content types: XML, String, and Binary.

For XML Message Content, the content is associated with a message through the use of an XML Schema any (xs:any) element with the processing requirement defined as strict (xs:strict). The XML content MUST be valid XML, specify a valid XML Schema, and validate against the specified XML schema. A ISBM Service Provider SHOULD preserve significant whitespace and comments within the XML content. An XML declaration MUST NOT appear within the XML Message Content.

For String Message Content, the content is associated through the use of an XML element of type xs:string. This allows content represented by textual data formats, such as JSON, to be exchanged within the XML message. The String content MUST have its mediaType specified. A list of media types is available [here](https://www.iana.org/assignments/media-types/media-types.xhtml). The String content MUST be correctly escaped according to the XML syntax if it includes protected XML characters.

A SOAP-based ISBM Service Provider SHOULD NOT exchange XML Message Content using the String content type.

For Binary Message Content, the content is associated through the use of an XML element containing Base64 encoded data (xs:base64binary). This allows content represented by binary formats to be exchanged within the XML message. The Binary content MAY specify the mediaType of the content. A list of media types is available [here](https://www.iana.org/assignments/media-types/media-types.xhtml).

A SOAP-based ISBM Service Provider SHOULD NOT exchange XML Message Content using the Binary content type.

In an XML message, the XML Schema Instance 'type' (xsi:type) attribute MUST be used to indicate the specific content type.

#### XML Message Content Example

<MessageContent xsi:type="XMLContent">

  <Property>There could be arbitrary XML content (with a single root node) included here.</Property>

</MessageContent>

#### String Message Content Example

<MessageContent xsi:type="StringContent" mediaType="application/json">

<Content>

{

  "prop": "There could be a JSON message, or anything else really."

}

</Content>

</MessageContent>

#### Binary Message Content Example

<MessageContent xsi:type="BinaryContent" mediaType="application/json">

<!-- strictly speaking there should be no newlines after/before the element tags below -->

<Content>

ew0KwqDCoCJwcm9wIjrCoCJUaGVyZcKgY291bGTCoGJlwqBhwqBKU09OwqBtZXNz

YWdlLMKgb3LCoGFueXRoaW5nwqBlbHNlwqByZWFsbHkuIg0KfQ==

</Content>

</MessageContent>

### REST Interface Requirements

The JSON Schemas for the REST interface are defined such that they allow the exchange of XML, JSON, and other Message Content types within JSON messages. The [JSON schema for Message Content](#_MessageContent_1) defined for the REST interface makes use of a flexibly defined ‘content’ property to support the different content types: JSON, String, and Binary.

For JSON Message Content, the content is associated with a message through the use of a JSON object as the property value. The JSON content MUST be valid JSON. The JSON content MUST NOT specify a mediaType nor contentEncoding. The JSON content MAY specify the URL of a JSON Schema if the ISBM Service Provider is to validate the JSON content against a schema.

For String Message Content, the content is associated through the use of a string as the property value. This allows content represented by textual data formats, such as XML, to be exchanged within the JSON message. The String content MUST have its mediaType specified. A list of media types is available [here](https://www.iana.org/assignments/media-types/media-types.xhtml). The String content MUST be correctly escaped according to the JSON syntax if it would include protected JSON characters.

A REST-based ISBM Service Provider SHOULD NOT exchange JSON Message Content using the String content type within a JSON message.

A REST-based ISBM Service Provider SHOULD NOT exchange XML Message Content using the String content type within an XML message.

For Binary Message Content, the content is associated through the use of a string as the property value and an additional contentEncoding property that specifies the encoding type, e.g., base64. This allows content represented by binary formats to be exchanged within the JSON message. The Binary content MUST specify the contentEncoding of the content. The contentEncoding value MUST be encoding types commonly supported by HTTP. The list of encoding types is available at XXX. The Binary content MAY specify the mediaType of the (decoded) content. A list of media types is available [here](https://www.iana.org/assignments/media-types/media-types.xhtml).

A REST-based ISBM Service Provider SHOULD NOT exchange JSON Message Content using the Binary content type within a JSON.

A REST-based ISBM Service Provider SHOULD NOT exchange XML Message Content using the Binary content type within an XML message.

The Channel URIs MUST be encoded when used within the URL of a REST call, for example: '<http://server/channels/encoded%2Fchannel%2FURI'>

#### JSON Message Content Example

The following is an HTTP request for postPublication containing JSON Message Content within a JSON message.

POST /sessions/321/publications HTTP/1.1

Host: http://example.com

Accept: application/json

Content-Type: application/json

Content-Length: 183

{

  "topics": ["topic1", "etc"],

  "expiry": "P1D",

  "messageContent": {

    "content": {

"somejson": "This is some JSON native content"

}

  }

}

#### String Message Content Example

The following is an HTTP request for *postPublication* containing XML content using the String Message Content type within a JSON message.

POST /sessions/321/publications HTTP/1.1

Host: [http://example.com](http://example.com/)

Accept: application/json

Content-Type: application/json

Content-Length: 187

{

"topics": ["topic1", "etc"],

"expiry": "P1D",

"messageContent": {

"mediaType": "application/xml",

"content": "<someXml>This is XML content in JSON</someXml>"

}

}

#### Binary Message Content Example

The following is an HTTP request for *postPublication* containing XML content using the Binary Message Content type within a JSON message. The content would decode to the same as the [String Message Content Example](#_String_Message_Content).

POST /sessions/321/publications HTTP/1.1

Host: http://example.com

Accept: application/json

Content-Type: application/json

Content-Length: 238

{

  "topics": ["topic1", "etc"],

  "expiry": "P1D",

  "messageContent": {

    "mediaType": "application/xml",

    "contentEncoding: "base64",

    "content": "PHNvbWVYbWw+VGhpcyBpcyBYTUwgY29udGVudCBpbiBKU09OPC9zb21lWG1sPg=="

  }

}

## Security

Security in the ISBM specification only provides authorization of channels. Authorization of services is considered out-of-scope.

All ISBM implementations MUST support transport layer security (e.g. SSL/TLS) in order to secure tokens and messages, and to prevent replay attacks.

All ISBM implementations MUST support username/password authentication as a basic level of security. This will differ for implementations of the different service types: for example, WS-Security UsernameToken for the SOAP interface and HTTP basic or digest authentication for the REST interface.

A ISBM Service Provider MAY choose to support additional forms of security tokens (e.g., SAML assertions, OAuth tokens) and it is RECOMMENDED that a ISBM Service Provider support out-of-band token exchange standards such as [SAML](http://saml.xml.org/saml-specifications), [WS-Federation](http://docs.oasis-open.org/wsfed/federation/v1.2/os/ws-federation-1.2-spec-os.html) or [OAuth](http://oauth.net/).

A ISBM Service Provider MUST validate security tokens for every service operation except for the Channel Management Service [CreateChannel operation](#_3.2.1_Create_Channel) (since the channel does not exist at the point in time when invoking CreateChannel). For the provider and consumer services, tokens are validated upon every operation to ensure that an application has valid credentials even after a session is opened (in the event of token revocation).

### SOAP Interface Requirements

All ISBM SOAP implementations MUST support the [WS-Security UsernameToken](https://www.oasis-open.org/committees/download.php/16782/wss-v1.1-spec-os-UsernameTokenProfile.pdf) using PasswordText as a basic level of security token. Examples of its use can be found in [Example HTTP Flows](http://www.openoandm.org/ws-isbm/1.0/ws-isbm.html#example-http-flows).

NOTE the requirement for supporting the use of WS-Security is in place of the, usually optional, SecurityToken listed as an input to most operations in the service descriptions of the ISA-95.00.06 MSM specification.

As security tokens in the [Channel Management Service](http://www.openoandm.org/ws-isbm/1.0/ws-isbm.html#channel-management-service) are specified using XML Schema any element, tokens MUST be able to be represented in an XML format. For tokens that do not have a canonical XML representation, a ISBM Service Provider MUST define the supported formats.

### REST Interface Requirements

All ISBM REST implementations MUST support the standard [HTTP/1.1 authentication](https://tools.ietf.org/html/rfc7235) and authorization headers. with potential support for security tokens. The credentials will be compared to SecurityTokens associated with the channel. The REST security tokens can be the same tokens used in the SOAP interface.

NOTE the requirement for supporting HTTP/1.1 authentication and authorization headers is in place of the, usually optional, SecurityToken listed as an input to most operations in the service descriptions of the ISA-95.00.06 MSM specification.

All ISBM REST implementations MUST support the basic authentication type of HTTP. It is RECOMMENDED that the JWT (Javascript Web Tokens) authentication type be supported.

As security tokens in the Channel Management Service are specified using a JSON Object, tokens MUST be able to be represented in a JSON format. For tokens that do not have a canonical JSON representation, a ISBM Service Provider MUST define the supported formats. The [UsernameToken](#_SecurityToken) schema defined in this specification MUST be supported by ISBM REST implementations.

## Error Handling

Faults MUST have an accompanying human readable explanation. For a SOAP 1.1 implementation this is provided through the SOAP faultstring element (see [SOAP 1.1, SOAP Fault](http://www.w3.org/TR/soap11/#_Toc478383507)). For a SOAP 1.2 implementation this is provided through the SOAP Reason element (see [SOAP 1.2, SOAP Reason Element](http://www.w3.org/TR/soap12-part1/#faultstringelement)).

For REST implementation, this is provided through a simple object schema containing the fault property (e.g., XXX).

NOTE The declared Faults specified by the services do not have any elements or attributes defined (other than the fault property for the REST definitions). This is because the sender can interpret the fault based on the supplied parameters and/or the operation behavior. For example, a ChannelFault returned by the DeleteChannel operation means that the ChannelURI provided by the sender did not exist.

### Parameter Faults

If any parameter for an operation is malformed or not optional and blank, then a ISBM Service Provider MUST return a ParameterFault to aid senders in determining the type of error.

For the SOAP interface, the undeclared isbm:ParameterFault element MUST be used in the fault details.

For the REST interface, the defined json:ParameterFault object MUST be used with a HTTP Response of 400 ‘Bad Request’.

The Fault MAY carry the offending parameter name/s and it is RECOMMENDED that the parameter names be included only in non-production environments, in order to eliminate information that may compromise security in production environment.

NOTE Parameter Faults are implicit for all Service Definitions and do not appear in the list of faults.

### Invalid Notification URL

If a provider/consumer application provides a URL that does not host a NotifyListener service, the ISBM Service Provider MAY choose to defer or not to send (possibly after some time) a NotifyListener request considering intermittent network issues.

NOTE If a provider/consumer application provides a malformed URL, a ParameterFault is returned. An invalid Notification URL is one in which the address is unreachable.

## Content-Based Filtering

To allow efficient content-based filtering of messages, FilterExpression/s MAY be added to a subscription or read request session to provide a filtering definition. As part of a conforming specification a ISBM Service Provider MUST declare the expression languages for which it provides support (possibly none). A ISBM Service Provider MUST ignore any FilterExpression/s not specified in a supported expression language by treating the expression to be defined using the special ‘ALLOW-ALL’ expression language.

A special expression language ‘ALLOW-ALL’ MUST be supported by all ISBM Service Providers that support content-based filtering. When the expression language is ‘ALLOW-ALL’ the expression MAY be empty. The result of evaluating an expression of the ‘ALLOW-ALL’ language MUST always return the complete MessageContent.

It is RECOMMENDED that an XPath expression be used for XML content and a [JSONPath](https://goessner.net/articles/JsonPath/) expression be used for JSON content. Other valid expression languages MAY also be used.

An XPath expression MUST be defined as an XPath v1.0 expression.

The FilterExpression MAY specify the mediaType/s to which the expression applies. If no mediaType is specified, the expression MUST be applied to Message Content of all content types.

More than one FilterExpression MAY be added to a subscription or read request session to allow different expressions, expressions in different languages, or expressions for different mediaType/s to be specified. For example, an XPath expression for XML content and a JSONPath expression for JSON content.

Only one FilterExpression for any particular combination of expression language and mediaType SHOULD be allowed.

If a FilterExpression is present then a notification MUST NOT be generated and the message MUST NOT be made available to the receiving system under any of the following conditions:

* The FilterExpression’s mediaType matches that of the Message Content and evaluation of the expression returns an empty value or node set (or otherwise is considered to not match the content based on the rules of the expression language); or
* There is no expression with a mediaType that matches that of the Message Content; or
* The evaluation of the expression is not possible due to being incompatible with the Message Content.

NOTE Based on this definition users must use the ‘ALLOW-ALL’ expression if you want to define expressions that filter XML content but not JSON content, for example.

For expression types that use namespaces (such as XPath), multiple namespace prefixes and names are added upon session creation.

NOTE An empty result from an expression evaluation will result in the whole message being is filtered; the message content itself is not filtered.

NOTE When expressions are present, a message will only be visible to the receiving application and/or have a notification generated for it if all applicable expressions match both the content type and evaluate to a non-empty result on the content.

NOTE Alternative expressions can be provided by opening multiple sessions on the same channel, each with its own filter expression.

## Message Expiry

During posting of certain messages, a sender MAY specify an expiry duration for the message. A ISBM Service Provider MUST not deliver an expired message to potential receivers unless the receiver has already read the message. If the message was read, then it MUST remain visible to that particular receiver. This is to ensure the message is always available to the receiver so that message removal removes the correct message.

If a sender specifies a negative Expiry duration, then a ISBM Service Provider MUST consider it equivalent to a blank duration.

NOTE Responses can still be posted for a previously read expired request message because the receiver has no indication that the message expired, and Consumers will still receive response notifications and be able to read and remove these responses.

## Feature Set Declaration

All ISBM Service Providers MUST declare their supported feature set through the ISBM Configuration Discovery Service (see Section 5.8).

The ISBM Configuration Discovery Service allows a ISBM Service Provider to provide information regarding its supported feature set in a machine interpretable way, allowing clients to configure themselves appropriately. The declared features are for the specific instance of the provider, not the possible capabilities of the implementation; those should be documented by the supplier.

Features that MUST be declared by a ISBM Service Provider include:

* Security level conformance
* Supported authentication token types
* Whether content-based filtering is supported
* Supported expression languages/versions for content-based filtering
* What else?

# Service Definitions

All services defined in ISA 95.00.06 are defined as SOAP Web Services or REST services in this specification. The SOAP and REST service definitions below are to be interpreted in the context of the corresponding ISA 95.00.06 service.

NOTE ISA 95.00.06 does not define an Expire Request operation within the Consumer Request Service, but it has been specified below for a consistent message expiry model across services.

All service operations have corresponding HTTP examples shown in Example HTTP Flows.

## Conformance to ISA-95.00.06

The following Service Definitions have been assessed for conformance to the requirements specified in ISA-95-00.06-2014 according to the provisions therein and listed as follows:

Terminology:

Services (to be repeated and completed for each service):

* Notification service
* Filter expression support
* Definition of service, optional components, non-compliance,

Any implementation of the ISBM specification that is assessed as conforming to the ISBM specification will be considered to conform to ISA-95-00.06-2014 as well, with the exceptions noted above.

## Channel Management Service

The Channel Management Service for SOAP Interface is [available as a WSDL description](http://www.openoandm.org/ws-isbm/1.1/wsdl/ChannelManagementService.wsdl) and for REST Interface is [available as OpenAPI 3.0.1 descriptions in YAML](http://www.openoandm.org/ws-isbm/1.1/yaml/channel_management_service.yml).

### Create Channel

The Create Channel service in general MUST have the behavior, inputs, outputs and return the faults as defined by the following table.

|  |  |
| --- | --- |
| Name | CreateChannel |
| Description | Creates a new channel. |
| Input | ChannelURI [1]  ChannelType [1]  ChannelDescription [0..1]  SecurityToken [0..\*] |
| Behavior | If the ChannelURI already exists then a ChannelFault is returned.  The SecurityTokens are assigned to the channel upon its creation.  If duplicate SecurityTokens exist, these result in a single token being assigned to the channel to maintain a distinct list. |
| Output | N/A |
| Faults | ChannelFault |

#### SOAP Mapping

The Create Channel general interface is mapped into SOAP 1.1/1.2 as embedded XML schemas in WSDL descriptions according to the following schema types.

The behavior of the SOAP interface MUST conform to that of the general description.

|  |  |
| --- | --- |
| Input | CreateChannel ([isbm:CreateChannel](http://www.openoandm.org/ws-isbm/1.1/wsdl/ChannelManagementService.wsdl))   * ChannelURI ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1] * ChannelType ([isbm:ChannelType](#_ChannelType)) [1] * ChannelDescription ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [0..1] * SecurityToken ([isbm:SecurityToken](#security-token-xml)) [0..\*] |
| Output | CreateChannelResponse ([isbm:CreateChannelResponse](http://www.openoandm.org/ws-isbm/1.1/wsdl/ChannelManagementService.wsdl))   * No content |
| Faults | ChannelFault ([isbm:ChannelFault](http://www.openoandm.org/ws-isbm/1.1/wsdl/ChannelManagementService.wsdl)) |

#### REST Mapping

The Create Channel general interface is mapped into a RESTful interface as an OpenAPI description according to the following rules.

The behavior of the REST interface MUST conform to that of the general description with necessary adjustments to conform to REST principles and HTTP specifications.

|  |  |
| --- | --- |
| HTTP Method | POST |
| URL | /channels |
| HTTP Body | createChannel ([json:createChannel](http://www.openoandm.org/ws-isbm/1.1/yaml/channel_management_service.yml))   * ChannelURI “uri” ([json:string](https://github.com/OAI/OpenAPI-Specification/blob/master/versions/3.0.1.md" \l "data-types)) [1] * ChannelType “channelType” ([json:ChannelType](#_ChannelType_1)) [1] * ChannelDescription “description” ([json:string](https://github.com/OAI/OpenAPI-Specification/blob/master/versions/3.0.1.md" \l "data-types)) [0..1] * SecurityToken “securityTokens” ([json:SecurityToken](#_SecurityToken)) [0..\*] |
| HTTP Response (Success) | 201 Created |
| Output | Channel ([json:Channel](#_Channel)) excluding securityTokens |
| HTTP Response  (Error) | ChannelFault ([json:ChannelFault](http://www.openoandm.org/ws-isbm/1.1/yaml/channel_management_service.yml)) – 409 Conflict |

NOTE Although not required by the general interface, success returns the Channel object in conformance with the requirements of a HTTP 201 response (<https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>).

NOTE The output Channel omits the SecurityTokens to prevent leakage of sensitive information.

### Add Security Tokens

The Add Security Tokens service in general MUST have the behavior, inputs, outputs and return the faults as defined by the following table.

|  |  |
| --- | --- |
| Name | AddSecurityTokens |
| Description | Adds security tokens to a channel. |
| Input | ChannelURI [1]  SecurityToken [1..\*] |
| Behavior | If the ChannelURI does not exist, then a ChannelFault is returned.  If the specified channel is assigned security tokens and the token provided to the operation for authentication does not match a token assigned to the channel, then a ChannelFault is returned.  If a specified SecurityToken is already assigned to the channel, then no further action is taken to maintain a distinct list.  If a SecurityToken is being added to a Channel that was created without any security tokens, then an OperationFault MUST be returned. |
| Output | N/A |
| Faults | ChannelFault  OperationFault – This is in addition to the Service Definition provided by ISA-95.00.06 |

#### SOAP Mapping

The Add Security Tokens general interface is mapped into SOAP 1.1/1.2 as embedded XML schemas in WSDL descriptions according to the following schema types.

The behavior of the SOAP interface MUST conform to that of the general description.

|  |  |
| --- | --- |
| Input | AddSecurityTokens ([isbm:AddSecurityTokens](http://www.openoandm.org/ws-isbm/1.1/wsdl/ChannelManagementService.wsdl))   * ChannelURI ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1] * SecurityToken ([isbm:SecurityToken](#security-token-xml)) [1..\*] |
| Output | AddSecurityTokensResponse ([isbm:AddSecurityTokensResponse](http://www.openoandm.org/ws-isbm/1.1/wsdl/ChannelManagementService.wsdl))   * No content |
| Faults | ChannelFault ([isbm:ChannelFault](http://www.openoandm.org/ws-isbm/1.1/wsdl/ChannelManagementService.wsdl)) |

#### REST Mapping

The Add Security Tokens general interface is mapped into a RESTful interface as an OpenAPI description according to the following rules.

The behavior of the REST interface MUST conform to that of the general description with necessary adjustments to conform to REST principles and HTTP specifications.

|  |  |
| --- | --- |
| HTTP Method | POST |
| URL | /channels/{channel-id}/security-tokens |
| HTTP Body | addSecurityTokens ([json:addSecurityTokens](http://www.openoandm.org/ws-isbm/1.1/yaml/channel_management_service.yml))   * SecurityToken “securityTokens” ([json:SecurityToken](#_SecurityToken)) [1..\*] |
| HTTP Response (Success) | 201 Created |
| Output | None |
| HTTP Response  (Error) | ChannelFault ([json:ChannelFault](http://www.openoandm.org/ws-isbm/1.1/yaml/channel_management_service.yml)) – 404 Not Found |

NOTE Returns nothing in contrast to the requirements of a HTTP 201 response (<https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>), which would expect the security tokens to be returned, to prevent leakage of sensitive information..

### Remove Security Tokens

The Remove Security Tokens service in general MUST have the behavior, inputs, outputs and return the faults as defined by the following table.

|  |  |
| --- | --- |
| Name | RemoveSecurityTokens |
| Description | Removes security tokens from a channel. |
| Input | ChannelURI [1]  SecurityToken [1..\*] |
| Behavior | If the ChannelURI does not exist, then a ChannelFault is returned.  If the specified channel is assigned security tokens and the token provided to the operation for authentication does not match a token assigned to the channel, then a ChannelFault is returned..  If any specified SecurityToken is not assigned to the channel, then a SecurityTokenFault is returned. No tokens are removed from the channel, even if they are valid. |
| Output | N/A |
| Faults | ChannelFault  SecurityTokenFault |

#### SOAP Mapping

The Remove Security Tokens general interface is mapped into SOAP 1.1/1.2 as embedded XML schemas in WSDL descriptions according to the following schema types.

The behavior of the SOAP interface MUST conform to that of the general description.

|  |  |
| --- | --- |
| Input | RemoveSecurityTokens ([isbm:RemoveSecurityTokens](http://www.openoandm.org/ws-isbm/1.1/wsdl/ChannelManagementService.wsdl))   * ChannelURI ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1] * SecurityToken ([isbm:SecurityToken](#security-token-xml)) [1..\*] |
| Output | RemoveSecurityTokensResponse ([isbm:RemoveSecurityTokensResponse](http://www.openoandm.org/ws-isbm/1.1/wsdl/ChannelManagementService.wsdl))   * No content |
| Faults | ChannelFault ([isbm:ChannelFault](http://www.openoandm.org/ws-isbm/1.1/wsdl/ChannelManagementService.wsdl))  SecurityTokenFault ([isbm:SecurityTokenFault](http://www.openoandm.org/ws-isbm/1.1/wsdl/ChannelManagementService.wsdl)) |

#### REST Mapping

The Remove Security Tokens general interface is mapped into a RESTful interface as an OpenAPI description according to the following rules.

The behavior of the REST interface MUST conform to that of the general description with necessary adjustments to conform to REST principles and HTTP specifications.

|  |  |
| --- | --- |
| HTTP Method | DELETE |
| URL | /channels/{channel-id}/security-tokens |
| HTTP Body | removeSecurityTokens ([json:removeSecurityTokens](http://www.openoandm.org/ws-isbm/1.1/yaml/channel_management_service.yml))   * SecurityToken “securityTokens” ([json:SecurityToken](#_SecurityToken)) [1..\*] |
| HTTP Response (Success) | 204 No Content |
| Output | None |
| HTTP Response  (Error) | ChannelFault ([json:ChannelFault](http://www.openoandm.org/ws-isbm/1.1/yaml/channel_management_service.yml)) – 404 Not Found  SecurityTokenFault ([json:SecurityTokenFault](http://www.openoandm.org/ws-isbm/1.1/yaml/channel_management_service.yml)) – 409 Conflict |

### Delete Channel

The Delete Channel service in general MUST have the behavior, inputs, outputs and return the faults as defined by the following table.

|  |  |
| --- | --- |
| Name | DeleteChannel |
| Description | Deletes a channel. |
| Input | ChannelURI [1] |
| Behavior | If the ChannelURI does not exist, then a ChannelFault is returned.  If the specified channel is assigned security tokens and the provided token does not match a token assigned to the channel, then a ChannelFault is returned.  The channel along with associated sessions and messages are deleted. No notification is provided to any applications with active sessions. |
| Output | N/A |
| Faults | ChannelFault |

### Get Channel

|  |  |
| --- | --- |
| Name | GetChannel |
| Description | Gets information about a channel. |
| Input | ChannelURI [1] |
| Behavior | If the ChannelURI does not exist, then a ChannelFault is returned.  If the specified channel is assigned security tokens and the provided token does not match a token assigned to the channel, then a ChannelFault is returned. |
| Output | Channel [1], composed of:      ChannelURI [1]      ChannelType [1]      ChannelDescription [0..1] |
| Faults | ChannelFault |

### Get Channels

|  |  |
| --- | --- |
| Name | GetChannels |
| Description | Gets information about all channels. |
| Input | N/A |
| Behavior | The channels returned are filtered by those that match the security token. Any channel that does not have security tokens assigned are returned regardless. |
| Output | Channel [0..\*], composed of:      ChannelURI ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1]      ChannelType ([isbm:ChannelType](http://www.openoandm.org/ws-isbm/1.0/ws-isbm.html" \l "channel-type-xml)) [1]      ChannelDescription ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [0..1] |
| Faults | N/A |

#### SOAP Mapping

The following table shows the SOAP and REST interfaces for the Channel Management Services.

|  |  |
| --- | --- |
| Input | GetChannels ([isbm:GetChannel](http://www.openoandm.org/ws-isbm/1.1/wsdl/ChannelManagementService.wsdl)s)   * ChannelURI ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1] |
| Output | GetChannelsResponse ([isbm:GetChannelsResponse](http://www.openoandm.org/ws-isbm/1.1/wsdl/ChannelManagementService.wsdl))   * Channel ([isbm:Channel](#channel-xml)) [0..\*], composed of:   + ChannelURI ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1]   + ChannelType ([isbm:ChannelType](#_ChannelType)) [1]   + ChannelDescription ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [0..1] |
| Faults | N/A |

#### REST Mapping

The Get Channels general interface is mapped into a RESTful interface as an OpenAPI description according to the following rules.

The behavior of the REST interface MUST conform to that of the general description with necessary adjustments to conform to REST principles and HTTP specifications.

|  |  |
| --- | --- |
| HTTP Method | GET |
| URL | /channels |
| HTTP Body | None |
| HTTP Response (Success) | 200 OK |
| Output | Channel ([json:Channel](#_Channel)) [0..\*] excluding securityTokens |
| HTTP Response  (Error) | N/A |

## Notification Service

The Notification Service for SOAP Interface is [available as a WSDL description](http://www.openoandm.org/ws-isbm/1.1/wsdl/NotificationService.wsdl) and for REST Interface is [available as OpenAPI 3.0.1 descriptions in YAML](http://www.openoandm.org/ws-isbm/1.1/yaml/notification_service.yml). The notification service is a callback from the ISBM provider to an application that has opened a channel using a notification option. The NotifyListener service is a service that MUST be implemented by the application. The notification service provides the ability to wait for responses and not require polling of sessions to determine if a message is available.

### Notify Listener

The Notify Listener service in general MUST have the behavior, inputs, outputs and return the faults as defined by the following table.

|  |  |
| --- | --- |
| Name | NotifyListener |
| Description | Provides a notification of a new message being able to be read for a session. The Listener URL invoked was provided when the application desiring notifications subscribed to the channel. |
| Input | SessionID [1]  MessageID [1]  Topic [0..\*]  RequestMessageID [0..1] |
| Behavior | Topic MUST NOT be used for consumer request session response notification.  RequestMessageID allows correlation with the original request and thus it MUST only be used for consumer request session response notification. |
| Output | N/A |
| Faults | N/A |

#### SOAP Mapping

The Notify Listener general interface is mapped into SOAP 1.1/1.2 as embedded XML schemas in WSDL descriptions according to the following schema types.

The behavior of the SOAP interface MUST conform to that of the general description.

|  |  |
| --- | --- |
| Input | NotifyListener ([isbm:](http://www.openoandm.org/ws-isbm/1.1/wsdl/ChannelManagementService.wsdl)NotifyListener)   * SessionID ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1] * MessageID ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1] * Topic ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [0..\*] * RequestMessageID ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [0..1] |
| Output | NotifyListenerResponse ([isbm:NotifyListenerResponse](http://www.openoandm.org/ws-isbm/1.1/wsdl/ChannelManagementService.wsdl))   * No content |
| Faults | N/A |

#### REST Mapping

The Notify Listener general interface is mapped into a RESTful interface as an OpenAPI description according to the following rules.

The behavior of the REST interface MUST conform to that of the general description with necessary adjustments to conform to REST principles and HTTP specifications.

|  |  |
| --- | --- |
| HTTP Method | PUT |
| URL | /notifications/{session-id}/{message-id} |
| HTTP Body | notifyListener ([json:notifyListener](http://www.openoandm.org/ws-isbm/1.1/yaml/channel_management_service.yml))   * Topic “topic” (json:string) [0..\*] * RequestMessageID “requestMessageId” ([json:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [0..1] |
| HTTP Response (Success) | 204 No Content |
| Output | None |
| HTTP Response  (Error) | None |

NOTE Session-id and message-id are provided in the URL to identify the resource for the PUT method. Both session-id and message-id have been used to ensure uniqueness across different sessions.

## Provider Publication Service

The Provider Publication Service for SOAP Interface is [available as a WSDL description](http://www.openoandm.org/ws-isbm/1.1/wsdl/NotificationService.wsdl) and for REST Interface is [available as OpenAPI 3.0.1 descriptions in YAML](http://www.openoandm.org/ws-isbm/1.1/yaml/notification_service.yml).

### Open Publication Session

The Open Publication Session service in general MUST have the behavior, inputs, outputs and return the faults as defined by the following table.

|  |  |
| --- | --- |
| Name | OpenPublicationSession |
| Description | Opens a publication session for a channel. |
| Input | ChannelURI [1] |
| Behavior | If the ChannelURI does not exist, then a ChannelFault is returned.  If the specified channel is assigned security tokens and the provided token does not match a token assigned to the channel, then a ChannelFault is returned.  If the channel type is not a Publication type, then an OperationFault is returned. |
| Output | SessionID [1] |
| Faults | ChannelFault  OperationFault |

#### SOAP Mapping

The Open Publication Session general interface is mapped into SOAP 1.1/1.2 as embedded XML schemas in WSDL descriptions according to the following schema types.

The behavior of the SOAP interface MUST conform to that of the general description.

|  |  |
| --- | --- |
| Input | OpenPublicationSession ([isbm:](http://www.openoandm.org/ws-isbm/1.1/wsdl/ChannelManagementService.wsdl)OpenPublicationSession)   * ChannelURI ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1] |
| Output | OpenPublicationSessionResponse ([isbm:](http://www.openoandm.org/ws-isbm/1.1/wsdl/ChannelManagementService.wsdl)OpenPublicationSessionResponse)   * SessionID ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1] |
| Faults | ChannelFault  OperationFault |

#### REST Mapping

The Open Publication Session general interface is mapped into a RESTful interface as an OpenAPI description according to the following rules.

The behavior of the REST interface MUST conform to that of the general description with necessary adjustments to conform to REST principles and HTTP specifications.

|  |  |
| --- | --- |
| HTTP Method | POST |
| URL | /channels/{channel-id}/publication-sessions |
| HTTP Body | None |
| HTTP Response (Success) | 201 Created |
| Output | Session (json:Session)   * SessionID “sessionId” ([json:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1] |
| HTTP Response  (Error) | ChannelFault (json:ChannelFault) – 404 Not Found  OperationFault (json:OperationFault) – 422 Unprocessable Entity |

### Post Publication

The Post Publication service in general MUST have the behavior, inputs, outputs and return the faults as defined by the following table.

|  |  |
| --- | --- |
| Name | PostPublication |
| Description | Posts a publication message on a channel. |
| Input | SessionID [1]  MessageContent [1]  Topic [1..\*]  Expiry [0..1] |
| Behavior | If the SessionID does not exist or does not correspond to a publication session, then a SessionFault is returned.  If the channel associated with the session is assigned security tokens and the provided token does not match a token assigned to the channel, then a SessionFault is returned. |
| Output | MessageID [1] |
| Faults | SessionFault |

#### SOAP Mapping

The Post Publication general interface is mapped into SOAP 1.1/1.2 as embedded XML schemas in WSDL descriptions according to the following schema types.

The behavior of the SOAP interface MUST conform to that of the general description.

|  |  |
| --- | --- |
| Input | PostPublication (isbm:PostPublication)   * SessionID ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1] * MessageContent ([isbm:MessageContent](http://www.openoandm.org/ws-isbm/1.0/ws-isbm.html" \l "message-content-xml)) [1] * Topic ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1..\*] * Expiry ([xs:duration](http://www.w3.org/TR/xmlschema-2/" \l "duration)) [0..1] |
| Output | PostPublicationResponse ([isbm:](http://www.openoandm.org/ws-isbm/1.1/wsdl/ChannelManagementService.wsdl)PostPublicationResponse)   * MessageID ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1] |
| Faults | SessionFault |

#### REST Mapping

The Post Publication general interface is mapped into a RESTful interface as an OpenAPI description according to the following rules.

The behavior of the REST interface MUST conform to that of the general description with necessary adjustments to conform to REST principles and HTTP specifications.

|  |  |
| --- | --- |
| HTTP Method | POST |
| URL | /sessions/{session-id}/publications |
| HTTP Body | postPublication(json:postPublication)   * Message (json:Message) [1]   + Content ([json:MessageContent](http://www.openoandm.org/ws-isbm/1.0/ws-isbm.html" \l "message-content-xml)) [1]   + Topic ([json:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1..\*]   + Expiry (json:duration) [0..1] |
| HTTP Response (Success) | 201 Created |
| Output | Message (json:Message)   * MessageID “messageId” ([json:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1] |
| HTTP Response  (Error) | SessionFault (json:SessionFault) – 404 Not Found  SessionFault (json:SessionFault) – 422 Unprocessable Entity – session exists but does not correspond to a publication session type |

### Expire Publication

|  |  |
| --- | --- |
| Name | ExpirePublication |
| Description | Expires a posted publication. |
| Input | SessionID [1]  MessageID [1] |
| Behavior | If the SessionID does not exist or does not correspond to a publication session, then a SessionFault is returned.  If the channel associated with the session is assigned security tokens and the provided token does not match a token assigned to the channel, then a SessionFault is returned.  If the MessageID does not correspond with the SessionID or the corresponding message has already expired, then no further action is taken.  The message is expired for all topics associated with the message. |
| Output | N/A |
| Faults | SessionFault |

#### SOAP Mapping

The Expire Publication general interface is mapped into SOAP 1.1/1.2 as embedded XML schemas in WSDL descriptions according to the following schema types.

The behavior of the SOAP interface MUST conform to that of the general description.

|  |  |
| --- | --- |
| Input | ExpirePublication (isbm:ExpirePublication)   * SessionID ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1] * MessageID ([xs:string](http://www.openoandm.org/ws-isbm/1.0/ws-isbm.html" \l "message-content-xml)) [1] |
| Output | ExpirePublicationResponse ([isbm:](http://www.openoandm.org/ws-isbm/1.1/wsdl/ChannelManagementService.wsdl)ExpirePublicationResponse)   * No content |
| Faults | SessionFault |

#### REST Mapping

The Expire Publication general interface is mapped into a RESTful interface as an OpenAPI description according to the following rules.

The behavior of the REST interface MUST conform to that of the general description with necessary adjustments to conform to REST principles and HTTP specifications.

|  |  |
| --- | --- |
| HTTP Method | DELETE |
| URL | /sessions/{session-id}/publications/{message-id} |
| HTTP Body | None |
| HTTP Response (Success) | 204 No Content |
| Output | N/A |
| HTTP Response  (Error) | SessionFault (json:SessionFault) – 404 Not Found  SessionFault (json:SessionFault) – 422 Unprocessable Entity – session exists but does not correspond to a publication session type |

### Close Publication Session

The Close Publication Session service in general MUST have the behavior, inputs, outputs and return the faults as defined by the following table.

|  |  |
| --- | --- |
| Name | ClosePublicationSession |
| Description | Closes a publication session. |
| Input | SessionID [1] |
| Behavior | If the SessionID does not exist (non-existent or already closed) or does not correspond to a publication session, then a SessionFault is returned.  If the channel associated with the session is assigned security tokens and the provided token does not match a token assigned to the channel, then a SessionFault is returned.  All unexpired messages that have been posted during the session will be expired. |
| Output | N/A |
| Faults | SessionFault |

#### SOAP Mapping

The Close Publication Session general interface is mapped into SOAP 1.1/1.2 as embedded XML schemas in WSDL descriptions according to the following schema types.

The behavior of the SOAP interface MUST conform to that of the general description.

|  |  |
| --- | --- |
| Input | ClosePublicationSession (isbm:ClosePublicationSession)   * SessionID ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1] |
| Output | ClosePublicationSessionResponse ([isbm:](http://www.openoandm.org/ws-isbm/1.1/wsdl/ChannelManagementService.wsdl)ClosePublicationSessionResponse)   * No content |
| Faults | SessionFault |

#### REST Mapping

The Close Publication Session general interface is mapped into a RESTful interface as an OpenAPI description according to the following rules.

The behavior of the REST interface MUST conform to that of the general description with necessary adjustments to conform to REST principles and HTTP specifications.

|  |  |
| --- | --- |
| HTTP Method | DELETE |
| URL | /sessions/{session-id} |
| HTTP Body | None |
| HTTP Response (Success) | 204 No Content |
| Output | N/A |
| HTTP Response  (Error) | SessionFault (json:SessionFault) – 404 Not Found |

## Consumer Publication Service

The Consumer Publication Service for SOAP Interface is [available as a WSDL description](http://www.openoandm.org/ws-isbm/1.1/wsdl/NotificationService.wsdl) and for REST Interface is [available as OpenAPI 3.0.1 descriptions in YAML](http://www.openoandm.org/ws-isbm/1.1/yaml/notification_service.yml).

### Open Subscription Session

The Open Subscription Session service in general MUST have the behavior, inputs, outputs and return the faults as defined by the following table.

|  |  |
| --- | --- |
| Name | OpenSubscriptionSession |
| Description | Opens a subscription session for a channel. |
| Input | ChannelURI [1]  Topic [1..\*]  ListenerURL [0..1]  XPathExpression [0..1]  XPathNamespace [0..\*], composed of:      NamespacePrefix [1]      NamespaceName [1] |
| Behavior | If the ChannelURI does not exist, then a ChannelFault is returned.  If the specified channel is assigned security tokens and the provided token does not match a token assigned to the channel, then a ChannelFault is returned.  If the channel type is not a Publication type, then an OperationFault is returned.  If multiple NamespacePrefixes exist with different NamespaceNames, then a NamespaceFault is returned. |
| Output | SessionID [1] |
| Faults | ChannelFault  NamespaceFault  OperationFault |

#### SOAP Mapping

The Open Subscription Session general interface is mapped into SOAP 1.1/1.2 as embedded XML schemas in WSDL descriptions according to the following schema types.

The behavior of the SOAP interface MUST conform to that of the general description.

|  |  |
| --- | --- |
| Input | OpenSubscriptionSession (isbm:OpenSubscriptionSession)   * ChannelURI ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1] * Topic ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1..\*] * ListenerURL ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [0..1] * XPathExpression ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [0..1] * XPathNamespace ([isbm:Namespace](http://www.openoandm.org/ws-isbm/1.0/ws-isbm.html" \l "namespace-xml)) [0..\*], composed of:   + NamespacePrefix ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1]   + NamespaceName ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1] |
| Output | OpenSubscriptionSessionResponse ([isbm:](http://www.openoandm.org/ws-isbm/1.1/wsdl/ChannelManagementService.wsdl)OpenSubscriptionSessionResponse)   * SessionID ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1] |
| Faults | ChannelFault  NamespaceFault  OperationFault |

#### REST Mapping

The Open Subscription Session general interface is mapped into a RESTful interface as an OpenAPI description according to the following rules.

The behavior of the REST interface MUST conform to that of the general description with necessary adjustments to conform to REST principles and HTTP specifications.

|  |  |
| --- | --- |
| HTTP Method | POST |
| URL | /channels/{channel-id}/subscription-sessions |
| HTTP Body | openSubscriptionSession (json:openSubscriptionSession)   * Session (json:Session)   + Topic “topics” (json:string) [1..\*]   + ListenerURL “listenerUrl” (json:string) [0..1]   + XPathExpression ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [0..1]   + XPathNamespace ([isbm:Namespace](http://www.openoandm.org/ws-isbm/1.0/ws-isbm.html" \l "namespace-xml)) [0..\*], composed of:     - NamespacePrefix ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1]     - NamespaceName ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1] |
| HTTP Response (Success) | 201 Created |
| Output | Session (json:Session)   * SessionID “sessionId” (json:string) [1] |
| HTTP Response  (Error) | ChannelFault (json:ChannelFault) – 404 Not Found  NamespaceFault (json:Namespacefault) – 400 Bad Request  OperationFault (json:OperationFault) – 422 Unprocessable Entity |

### Read Publication

|  |  |
| --- | --- |
| Name | ReadPublication |
| Description | Returns the first non-expired publication message or a previously read expired message that satisfies the session message filters. |
| Input | SessionID [1] |
| Behavior | If the SessionID does not exist or does not correspond to a subscription session, then a SessionFault is returned.  If the channel associated with the session is assigned security tokens and the provided token does not match a token assigned to the channel, then a SessionFault is returned.  The returned Topics will correspond to the intersection of the Topics of the posted publication and the Topics specified in the subscription session. |
| Output | PublicationMessage [0..1], composed of:      MessageID [1]      MessageContent [1]      Topic [1..\*] |
| Faults | SessionFault |

#### SOAP Mapping

The Read Publication general interface is mapped into SOAP 1.1/1.2 as embedded XML schemas in WSDL descriptions according to the following schema types.

The behavior of the SOAP interface MUST conform to that of the general description.

|  |  |
| --- | --- |
| Input | ReadPublication (isbm:ReadPublication)   * SessionID ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1] |
| Output | ReadPublicationResponse ([isbm:](http://www.openoandm.org/ws-isbm/1.1/wsdl/ChannelManagementService.wsdl)ReadPublicationResponse)   * PublicationMessage ([isbm:PublicationMessage](http://www.openoandm.org/ws-isbm/1.0/ws-isbm.html" \l "publication-message-xml)) [0..1], composed of:   + MessageID ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1]   + MessageContent ([isbm:MessageContent](http://www.openoandm.org/ws-isbm/1.0/ws-isbm.html" \l "message-content-xml)) [1]   + Topic ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1..\*] |
| Faults | SessionFault |

NOTE A no message response is a successful with no PublicationMessage element.

#### REST Mapping

The Read Publication general interface is mapped into a RESTful interface as an OpenAPI description according to the following rules.

The behavior of the REST interface MUST conform to that of the general description with necessary adjustments to conform to REST principles and HTTP specifications.

|  |  |
| --- | --- |
| HTTP Method | GET |
| URL | /sessions/{session-id}/publication |
| HTTP Body | None |
| HTTP Response (Success) | 200 OK |
| Output | Message (json:Message)   * MessageID “messageId” (json:string) [1] * MessageContent “messageContent” (json:[MessageContent](http://www.openoandm.org/ws-isbm/1.0/ws-isbm.html#message-content-xml)) [1] * Topic “topics” ([json:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1..\*] |
| HTTP Response  (Error) | SessionFault (json:SessionFault) – 404 Not Found  SessionFault (json:SessionFault) – 422 Unprocessable Entity – session exists but does not correspond to a subscription session type |

NOTE A no message response is returned as a 404 Not Found rather than an "empty" message as it maps better to the concept of resources in a RESTful API. If there are no messages on the queue, the resource does not exist and, hence, 404 should be returned.

### Remove Publication

The Remove Publication service in general MUST have the behavior, inputs, outputs and return the faults as defined by the following table.

|  |  |
| --- | --- |
| Name | RemovePublication |
| Description | Removes the first, if any, publication message in the subscription queue. |
| Input | SessionID [1] |
| Behavior | If the SessionID does not exist or does not correspond to a subscription session, then a SessionFault is returned.  If the channel associated with the session is assigned security tokens and the provided token does not match a token assigned to the channel, then a SessionFault is returned. |
| Output | N/A |
| Faults | SessionFault |

#### SOAP Mapping

The Remove Publication general interface is mapped into SOAP 1.1/1.2 as embedded XML schemas in WSDL descriptions according to the following schema types.

The behavior of the SOAP interface MUST conform to that of the general description.

|  |  |
| --- | --- |
| Input | RemovePublication (isbm:RemovePublication)   * SessionID ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1] |
| Output | RemovePublicationResponse ([isbm:](http://www.openoandm.org/ws-isbm/1.1/wsdl/ChannelManagementService.wsdl)RemovePublicationResponse)   * No Content |
| Faults | SessionFault |

#### REST Mapping

The Remove Publication general interface is mapped into a RESTful interface as an OpenAPI description according to the following rules.

The behavior of the REST interface MUST conform to that of the general description with necessary adjustments to conform to REST principles and HTTP specifications.

|  |  |
| --- | --- |
| HTTP Method | DELETE |
| URL | /sessions/{session-id}/publication |
| HTTP Body | None |
| HTTP Response (Success) | 204 No Content |
| Output | None |
| HTTP Response  (Error) | SessionFault (json:SessionFault) – 404 Not Found  SessionFault (json:SessionFault) – 422 Unprocessable Entity |

### Close Subscription Session

The Close Subscription Session service in general MUST have the behavior, inputs, outputs and return the faults as defined by the following table.

|  |  |
| --- | --- |
| Name | CloseSubscriptionSession |
| Description | Closes a subscription session. |
| Input | SessionID [1] |
| Behavior | If the SessionID does not exist (non-existent or already closed) or does not correspond to a publication session, then a SessionFault is returned.  If the channel associated with the session is assigned security tokens and the provided token does not match a token assigned to the channel, then a SessionFault is returned. |
| Output | N/A |
| Faults | SessionFault |

#### SOAP Mapping

The Close Subscription Session general interface is mapped into SOAP 1.1/1.2 as embedded XML schemas in WSDL descriptions according to the following schema types.

The behavior of the SOAP interface MUST conform to that of the general description.

|  |  |
| --- | --- |
| Input | CloseSubscriptionSession (isbm:CloseSubscriptionSession)   * SessionID ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1] |
| Output | CloseSubscriptionSessionResponse ([isbm:](http://www.openoandm.org/ws-isbm/1.1/wsdl/ChannelManagementService.wsdl)CloseSubscriptionSessionResponse)   * No Content |
| Faults | SessionFault |

#### REST Mapping

The Close Subscription Session general interface is mapped into a RESTful interface as an OpenAPI description according to the following rules.

The behavior of the REST interface MUST conform to that of the general description with necessary adjustments to conform to REST principles and HTTP specifications.

|  |  |
| --- | --- |
| HTTP Method | DELETE |
| URL | /sessions/{session-id} |
| HTTP Body | None |
| HTTP Response (Success) | 204 No Content |
| Output | None |
| HTTP Response  (Error) | SessionFault (json:SessionFault) – 404 Not Found |

## Provider Request Service

The Provider Request Service for SOAP Interface is [available as a WSDL description](http://www.openoandm.org/ws-isbm/1.1/wsdl/NotificationService.wsdl) and for REST Interface is [available as OpenAPI 3.0.1 descriptions in YAML](http://www.openoandm.org/ws-isbm/1.1/yaml/notification_service.yml).

### Open Provider Request Session

The Open Provider Request Session service in general MUST have the behavior, inputs, outputs and return the faults as defined by the following table.

|  |  |
| --- | --- |
| Name | OpenProviderRequestSession |
| Description | Opens a provider request session for a channel for reading requests and posting responses. |
| Input | ChannelURI [1]  Topic [1..\*]  ListenerURL [0..1]  XPathExpression [0..1]  XPathNamespace [0..\*], composed of:      NamespacePrefix [1]      NamespaceName [1] |
| Behavior | If the ChannelURI does not exist, then a ChannelFault is returned.  If the specified channel is assigned security tokens and the provided token does not match a token assigned to the channel, then a ChannelFault is returned.  If the channel type is not a Request type, then an OperationFault is returned.  If multiple NamespacePrefixes exist with different NamespaceNames, then a NamespaceFault is returned. |
| Output | SessionID [1] |
| Faults | ChannelFault  NamespaceFault  OperationFault |

#### SOAP Mapping

The Open Provider Request Session general interface is mapped into SOAP 1.1/1.2 as embedded XML schemas in WSDL descriptions according to the following schema types.

The behavior of the SOAP interface MUST conform to that of the general description.

|  |  |
| --- | --- |
| Input | OpenProviderRequestSession (isbm:OpenProviderRequestSession)   * ChannelURI ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1] * Topic (xs:string) [1..\*] * ListenerURL (xs:string) [0..1] * XPathExpression (xs:string) [0..1] * XPathNamespace (isbm:Namespace) [0..\*], composed of: * NamespacePrefix (xs:string) [1] * NamespaceName (xs:string) [1] |
| Output | OpenProviderRequestSessionResponse (isbm:OpenProviderRequestSessionResponse)   * SessionID ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1] |
| Faults | ChannelFault  NamespaceFault  OperationFault |

#### REST Mapping

The Open Provider Request Session general interface is mapped into a RESTful interface as an OpenAPI description according to the following rules.

The behavior of the REST interface MUST conform to that of the general description with necessary adjustments to conform to REST principles and HTTP specifications.

|  |  |
| --- | --- |
| HTTP Method | POST |
| URL | /channels/{channel-id}/provider-request-sessions |
| HTTP Body | * Session (json:Session)   + Topic “topics” (json:string) [1..\*]   + ListenerURL “listenerUrl” (json:string) [0..1]   + XPathExpression ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [0..1]   + XPathNamespace ([isbm:Namespace](http://www.openoandm.org/ws-isbm/1.0/ws-isbm.html" \l "namespace-xml)) [0..\*], composed of:     - NamespacePrefix ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1]     - NamespaceName ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1] |
| HTTP Response (Success) | 201 Created |
| Output | Session (json:Session)   * SessionID “sessionId” (json:string) [1] |
| HTTP Response  (Error) | ChannelFault (json:ChannelFault) – 404 Not Found  NamespaceFault (json:Namespacefault) – 400 Bad Request  OperationFault (json:OperationFault) – 422 Unprocessable Entity |

### Read Request

The Read Request service in general MUST have the behavior, inputs, outputs and return the faults as defined by the following table.

|  |  |
| --- | --- |
| Name | ReadRequest |
| Description | Returns the first non-expired request message or a previously read expired message that satisfies the session message filters. |
| Input | SessionID [1] |
| Behavior | If the SessionID does not exist or does not correspond to a provider request session, then a SessionFault is returned.  If the channel associated with the session is assigned security tokens and the provided token does not match a token assigned to the channel, then a SessionFault is returned.  The returned Topic will correspond to the first topic that matched the posted request. |
| Output | RequestMessage [0..1], composed of:      MessageID [1]      MessageContent [1]      Topic [1] |
| Faults | SessionFault |

#### SOAP Mapping

The Read Request general interface is mapped into SOAP 1.1/1.2 as embedded XML schemas in WSDL descriptions according to the following schema types.

The behavior of the SOAP interface MUST conform to that of the general description.

|  |  |
| --- | --- |
| Input | ReadRequest (isbm:ReadRequest)   * SessionID ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1] |
| Output | ReadRequestResponse ([isbm:](http://www.openoandm.org/ws-isbm/1.1/wsdl/ChannelManagementService.wsdl)ReadRequestResponse)   * RequestMessage ([isbm:RequestMessage](http://www.openoandm.org/ws-isbm/1.0/ws-isbm.html" \l "request-message-xml)) [0..1], composed of:   + MessageID ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1]   + MessageContent ([isbm:MessageContent](http://www.openoandm.org/ws-isbm/1.0/ws-isbm.html" \l "message-content-xml)) [1]   + Topic ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1..\*] |
| Faults | SessionFault |

#### REST Mapping

The Read Request general interface is mapped into a RESTful interface as an OpenAPI description according to the following rules.

The behavior of the REST interface MUST conform to that of the general description with necessary adjustments to conform to REST principles and HTTP specifications.

|  |  |
| --- | --- |
| HTTP Method | GET |
| URL | /sessions/{session-id}/request |
| HTTP Body | None |
| HTTP Response (Success) | 200 OK |
| Output | Message (json:Message)   * MessageID “messageId” (json:string) [1] * MessageContent “messageContent” (json:[MessageContent](http://www.openoandm.org/ws-isbm/1.0/ws-isbm.html#message-content-xml)) [1] * Topic “topics” ([json:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1..\*] |
| HTTP Response  (Error) | SessionFault (json:SessionFault) – 404 Not Found  SessionFault (json:SessionFault) – 422 Unprocessable Entity |

NOTE In contrast to the SOAP web-service, no message is returned as a 404 rather than an "empty" message. This maps better to a RESTful API that is based on the idea of resources. If there are no messages on the queue, the resource does not exist and, hence, 404 should be returned.

### Remove Request

The Remove Request service in general MUST have the behavior, inputs, outputs and return the faults as defined by the following table.

|  |  |
| --- | --- |
| Name | RemoveRequest |
| Description | Deletes the first request message, if any, in the session message queue. |
| Input | SessionID ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1] |
| Behavior | If the SessionID does not exist or does not correspond to a provider request session, then a SessionFault is returned.  If the channel associated with the session is assigned security tokens and the provided token does not match a token assigned to the channel, then a SessionFault is returned. |
| Output | N/A |
| Faults | SessionFault |

#### SOAP Mapping

The Remove Request general interface is mapped into SOAP 1.1/1.2 as embedded XML schemas in WSDL descriptions according to the following schema types.

The behavior of the SOAP interface MUST conform to that of the general description.

|  |  |
| --- | --- |
| Input | RemoveRequest (isbm:RemoveRequest)   * SessionID ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1] |
| Output | RemoveRequestResponse ([isbm:](http://www.openoandm.org/ws-isbm/1.1/wsdl/ChannelManagementService.wsdl)RemoveRequestResponse)   * No Content |
| Faults | SessionFault |

#### REST Mapping

The Remove Request general interface is mapped into a RESTful interface as an OpenAPI description according to the following rules.

The behavior of the REST interface MUST conform to that of the general description with necessary adjustments to conform to REST principles and HTTP specifications.

|  |  |
| --- | --- |
| HTTP Method | DELETE |
| URL | /sessions/{session-id}/request |
| HTTP Body | None |
| HTTP Response (Success) | 204 No Content |
| Output | None |
| HTTP Response  (Error) | SessionFault (json:SessionFault) – 404 Not Found  SessionFault (json:SessionFault) – 422 Unprocessable Entity |

### Post Response

The Post Response service in general MUST have the behavior, inputs, outputs and return the faults as defined by the following table.

|  |  |
| --- | --- |
| Name | PostResponse |
| Description | Posts a response message on a channel. |
| Input | SessionID [1]  RequestMessageID [1]  MessageContent [1] |
| Behavior | If the SessionID does not exist or does not correspond to a provider request session, then a SessionFault is returned.  If the channel associated with the session is assigned security tokens and the provided token does not match a token assigned to the channel, then a SessionFault is returned. |
| Output | MessageID [1] |
| Faults | SessionFault |

#### SOAP Mapping

The Post Response general interface is mapped into SOAP 1.1/1.2 as embedded XML schemas in WSDL descriptions according to the following schema types.

The behavior of the SOAP interface MUST conform to that of the general description.

|  |  |
| --- | --- |
| Input | PostResponse (isbm:PostResponse)   * SessionID ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1] * RequestMessageID ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1] * MessageContent ([isbm:MessageContent](http://www.openoandm.org/ws-isbm/1.0/ws-isbm.html" \l "message-content-xml)) [1] |
| Output | PostResponseResponse ([isbm:](http://www.openoandm.org/ws-isbm/1.1/wsdl/ChannelManagementService.wsdl)PostResponseResponse)   * MessageID ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1] |
| Faults | SessionFault |

#### REST Mapping

The Post Response general interface is mapped into a RESTful interface as an OpenAPI description according to the following rules.

The behavior of the REST interface MUST conform to that of the general description with necessary adjustments to conform to REST principles and HTTP specifications.

|  |  |
| --- | --- |
| HTTP Method | POST |
| URL | /sessions/{session-id}/requests/{request-id}/responses |
| HTTP Body | postResponse(json:postResponse)   * Message (json:Message) [1]   + Content ([json:MessageContent](http://www.openoandm.org/ws-isbm/1.0/ws-isbm.html" \l "message-content-xml)) [1] |
| HTTP Response (Success) | 201 Created |
| Output | Message (json:Message)   * MessageID “messageId” ([json:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1] |
| HTTP Response  (Error) | SessionFault (json:SessionFault) – 404 Not Found  SessionFault (json:SessionFault) – 422 Unprocessable Entity |

### Close Provider Request Session

The Close Provider Request Session service in general MUST have the behavior, inputs, outputs and return the faults as defined by the following table.

|  |  |
| --- | --- |
| Name | CloseProviderRequestSession |
| Description | Closes a provider request session. |
| Input | SessionID [1] |
| Behavior | If the SessionID does not exist (non-existent or already closed) or does not correspond to a Request session, then a SessionFault is returned.  If the channel associated with the session is assigned security tokens and the provided token does not match a token assigned to the channel, then a SessionFault is returned. |
| Output | N/A |
| Faults | SessionFault |

#### SOAP Mapping

The Close Provider Request Session general interface is mapped into SOAP 1.1/1.2 as embedded XML schemas in WSDL descriptions according to the following schema types.

The behavior of the SOAP interface MUST conform to that of the general description.

|  |  |
| --- | --- |
| Input | CloseProviderRequestSession (isbm:CloseProviderRequestSession)   * SessionID [1] |
| Output | CloseProviderRequestSessionResponse ([isbm:](http://www.openoandm.org/ws-isbm/1.1/wsdl/ChannelManagementService.wsdl)CloseProviderRequestSessionResponse)   * No Content |
| Faults | SessionFault |

#### REST Mapping

The Close Provider Request Session general interface is mapped into a RESTful interface as an OpenAPI description according to the following rules.

The behavior of the REST interface MUST conform to that of the general description with necessary adjustments to conform to REST principles and HTTP specifications.

|  |  |
| --- | --- |
| HTTP Method | DELETE |
| URL | /sessions/{session-id} |
| HTTP Body | None |
| HTTP Response (Success) | 204 No Content |
| Output | None |
| HTTP Response  (Error) | SessionFault (json:SessionFault) – 404 Not Found |

## Consumer Request Service

The Consumer Request Service for SOAP Interface is [available as a WSDL description](http://www.openoandm.org/ws-isbm/1.1/wsdl/NotificationService.wsdl) and for REST Interface is [available as OpenAPI 3.0.1 descriptions in YAML](http://www.openoandm.org/ws-isbm/1.1/yaml/notification_service.yml).

### Open Consumer Request Session

The Open Customer Request Session service in general MUST have the behavior, inputs, outputs and return the faults as defined by the following table.

|  |  |
| --- | --- |
| Name | OpenConsumerRequestSession |
| Description | Opens a consumer request session for a channel for posting requests and reading responses. |
| Input | ChannelURI [1]  ListenerURL [0..1] |
| Behavior | If the ChannelURI does not exist, then a ChannelFault is returned.  If the specified channel is assigned security tokens and the provided token does not match a token assigned to the channel, then a ChannelFault is returned.  If the channel type is not a Request type, then an OperationFault is returned. |
| Output | SessionID [1] |
| Faults | ChannelFault  OperationFault |

#### SOAP Mapping

The Open Consumer Request Session general interface is mapped into SOAP 1.1/1.2 as embedded XML schemas in WSDL descriptions according to the following schema types.

The behavior of the SOAP interface MUST conform to that of the general description.

|  |  |
| --- | --- |
| Input | OpenConsumerRequestSession (isbm:OpenConsumerRequestSession)   * ChannelURI ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1] * ListenerURL (xs:string) [0..1] |
| Output | OpenConsumerRequestSessionResponse(isbm:OpenConsumerRequestSessionResponse)   * SessionID ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1] |
| Faults | ChannelFault  OperationFault |

#### REST Mapping

The Open Consumer Request Session general interface is mapped into a RESTful interface as an OpenAPI description according to the following rules.

The behavior of the REST interface MUST conform to that of the general description with necessary adjustments to conform to REST principles and HTTP specifications.

|  |  |
| --- | --- |
| HTTP Method | POST |
| URL | /channels/{channel-id}/consumer-request-sessions |
| HTTP Body | openConsumerRequestSession (json: openConsumerRequestSession)   * Session (json:Session)   + ListenerURL “listenerUrl” (json:string) [0..1] |
| HTTP Response (Success) | 201 Created |
| Output | Session (json:Session)   * SessionID “sessionId” (json:string) [1] |
| HTTP Response  (Error) | ChannelFault (json:ChannelFault) – 404 Not Found  OperationFault (json:OperationFault) – 422 Unprocessable Entity |

### Post Request

The Post Request service in general MUST have the behavior, inputs, outputs and return the faults as defined by the following table.

|  |  |
| --- | --- |
| Name | PostRequest |
| Description | Posts a request message on a channel. |
| Input | SessionID [1]  MessageContent [1]  Topic [1]  Expiry [0..1] |
| Behavior | If the SessionID does not exist or does not correspond to a consumer request session, then a SessionFault is returned.  If the channel associated with the session is assigned security tokens and the provided token does not match a token assigned to the channel, then a SessionFault is returned. |
| Output | MessageID [1] |
| Faults | SessionFault |

#### SOAP Mapping

The Post Request general interface is mapped into SOAP 1.1/1.2 as embedded XML schemas in WSDL descriptions according to the following schema types.

The behavior of the SOAP interface MUST conform to that of the general description.

|  |  |
| --- | --- |
| Input | PostRequest (isbm:PostRequest)   * SessionID ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1] * MessageContent ([isbm:MessageContent](http://www.openoandm.org/ws-isbm/1.0/ws-isbm.html" \l "message-content-xml)) [1] * Topic ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1..\*] * Expiry ([xs:duration](http://www.w3.org/TR/xmlschema-2/" \l "duration)) [0..1] |
| Output | PostRequestResponse ([isbm:](http://www.openoandm.org/ws-isbm/1.1/wsdl/ChannelManagementService.wsdl)PostRequestResponse)   * MessageID ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1] |
| Faults | SessionFault |

#### REST Mapping

The Post Request general interface is mapped into a RESTful interface as an OpenAPI description according to the following rules.

The behavior of the REST interface MUST conform to that of the general description with necessary adjustments to conform to REST principles and HTTP specifications.

|  |  |
| --- | --- |
| HTTP Method | POST |
| URL | /sessions/{session-id}/requests |
| HTTP Body | postRequest(json:postRequest)   * Message (json:Message) [1]   + Content ([json:MessageContent](http://www.openoandm.org/ws-isbm/1.0/ws-isbm.html" \l "message-content-xml)) [1]   + Topic ([json:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1..\*]   + Expiry (json:duration) [0..1] |
| HTTP Response (Success) | 201 Created |
| Output | Message (json:Message)   * MessageID “messageId” ([json:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1] |
| HTTP Response  (Error) | SessionFault (json:SessionFault) – 404 Not Found  SessionFault (json:SessionFault) – 422 Unprocessable Entity |

### Expire Request

The Expire Request service in general MUST have the behavior, inputs, outputs and return the faults as defined by the following table.

|  |  |
| --- | --- |
| Name | ExpireRequest |
| Description | Expires a posted request message. |
| Input | SessionID [1]  MessageID [1] |
| Behavior | If the SessionID does not exist or does not correspond to a consumer request session, then a SessionFault is returned.  If the channel associated with the session is assigned security tokens and the provided token does not match a token assigned to the channel, then a SessionFault is returned.  If the MessageID does not correspond with the SessionID or the corresponding message has already expired, then no further action is taken.  Any unread responses associated with the request MAY be removed from the queue of the consumer. |
| Output | N/A |
| Faults | SessionFault |

NOTE It has been left open for vendor’s implementation to document what should happen to responses that have already been posted against a request that has subsequently expired. The two options are, should responses be available for consumer to read or should they be removed from the consumer’s queue (unless they have been read previously).

#### SOAP Mapping

The Expire Request general interface is mapped into SOAP 1.1/1.2 as embedded XML schemas in WSDL descriptions according to the following schema types.

The behavior of the SOAP interface MUST conform to that of the general description.

|  |  |
| --- | --- |
| Input | ExpireRequest (isbm:ExpireRequest)   * SessionID ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1] * MessageID ([xs:string](http://www.openoandm.org/ws-isbm/1.0/ws-isbm.html" \l "message-content-xml)) [1] |
| Output | ExpireRequestResponse ([isbm:](http://www.openoandm.org/ws-isbm/1.1/wsdl/ChannelManagementService.wsdl)ExpireRequestResponse)   * No content |
| Faults | SessionFault |

#### REST Mapping

The Expire Request general interface is mapped into a RESTful interface as an OpenAPI description according to the following rules.

The behavior of the REST interface MUST conform to that of the general description with necessary adjustments to conform to REST principles and HTTP specifications.

|  |  |
| --- | --- |
| HTTP Method | DELETE |
| URL | /sessions/{session-id}/requests/{message-id} |
| HTTP Body | None |
| HTTP Response (Success) | 204 No Content |
| Output | N/A |
| HTTP Response  (Error) | SessionFault (json:SessionFault) – 404 Not Found  SessionFault (json:SessionFault) – 422 Unprocessable Entity |

### Read Response

The Read Response service in general MUST have the behavior, inputs, outputs and return the faults as defined by the following table.

|  |  |
| --- | --- |
| Name | ReadResponse |
| Description | Returns the first response message, if any, in the session message queue associated with the request. |
| Input | SessionID [1]  RequestMessageID [1] |
| Behavior | If the SessionID does not exist or does not correspond to a consumer request session, then a SessionFault is returned.  If the channel associated with the session is assigned security tokens and the provided token does not match a token assigned to the channel, then a SessionFault is returned.  If the RequestMessageID does not correspond to a message in the message queue, then no message is returned. |
| Output | ResponseMessage [0..1], composed of:      MessageID [1]      MessageContent [1] |
| Faults | SessionFault |

#### SOAP Mapping

The Read Response general interface is mapped into SOAP 1.1/1.2 as embedded XML schemas in WSDL descriptions according to the following schema types.

The behavior of the SOAP interface MUST conform to that of the general description.

|  |  |
| --- | --- |
| Input | ReadResponse (isbm:ReadResponse)   * SessionID ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1] * RequestMessageID (xs:string) [1] |
| Output | ReadResponseResponse ([isbm:](http://www.openoandm.org/ws-isbm/1.1/wsdl/ChannelManagementService.wsdl)ReadResponseResponse)   * ResponseMessage ([isbm:ResponseMessage](http://www.openoandm.org/ws-isbm/1.0/ws-isbm.html" \l "request-message-xml)) [0..1], composed of:   + MessageID ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1]   + MessageContent ([isbm:MessageContent](http://www.openoandm.org/ws-isbm/1.0/ws-isbm.html" \l "message-content-xml)) [1] |
| Faults | SessionFault |

#### REST Mapping

The Read Response general interface is mapped into a RESTful interface as an OpenAPI description according to the following rules.

The behavior of the REST interface MUST conform to that of the general description with necessary adjustments to conform to REST principles and HTTP specifications.

|  |  |
| --- | --- |
| HTTP Method | GET |
| URL | /sessions/{session-id}/requests/{request-id}/response |
| HTTP Body | None |
| HTTP Response (Success) | 200 OK |
| Output | Message (json:Message)   * MessageID “messageId” (json:string) [1] * MessageContent “messageContent” (json:[MessageContent](http://www.openoandm.org/ws-isbm/1.0/ws-isbm.html#message-content-xml)) [1] |
| HTTP Response  (Error) | SessionFault (json:SessionFault) – 404 Not Found  SessionFault (json:SessionFault) – 422 Unprocessable Entity |

NOTE In contrast to the SOAP web-service, no message is returned as a 404 rather than an "empty" message. This maps better to a RESTful API that is based on the idea of resources. If there are no messages on the queue, the resource does not exist and, hence, 404 should be returned.

### Remove Response

The Remove Response service in general MUST have the behavior, inputs, outputs and return the faults as defined by the following table.

|  |  |
| --- | --- |
| Name | RemoveResponse |
| Description | Deletes the first response message, if any, in the session message queue associated with the request. |
| Input | SessionID [1]  RequestMessageID [1] |
| Behavior | If the SessionID does not exist or does not correspond to a consumer request session, then a SessionFault is returned.  If the channel associated with the session is assigned security tokens and the provided token does not match a token assigned to the channel, then a SessionFault is returned.  If the RequestMessageID does not correspond to a message in the message queue, then no further action is taken. |
| Output | N/A |
| Faults | SessionFault |

#### SOAP Mapping

The Remove Response general interface is mapped into SOAP 1.1/1.2 as embedded XML schemas in WSDL descriptions according to the following schema types.

The behavior of the SOAP interface MUST conform to that of the general description.

|  |  |
| --- | --- |
| Input | RemoveResponse (isbm:RemoveResponse)   * SessionID ([xs:string](http://www.w3.org/TR/xmlschema-2/" \l "string)) [1] * RequestMessageID (xs:string) [1] |
| Output | RemoveResponseResponse ([isbm:](http://www.openoandm.org/ws-isbm/1.1/wsdl/ChannelManagementService.wsdl)RemoveResponseResponse)   * No Content |
| Faults | SessionFault |

#### REST Mapping

The Remove Response general interface is mapped into a RESTful interface as an OpenAPI description according to the following rules.

The behavior of the REST interface MUST conform to that of the general description with necessary adjustments to conform to REST principles and HTTP specifications.

|  |  |
| --- | --- |
| HTTP Method | DELETE |
| URL | /sessions/{session-id}/requests/{request-id}/response |
| HTTP Body | None |
| HTTP Response (Success) | 204 No Content |
| Output | None |
| HTTP Response  (Error) | SessionFault (json:SessionFault) – 404 Not Found  SessionFault (json:SessionFault) – 422 Unprocessable Entity |

### Close Consumer Request Session

The Close Consumer Request Session service in general MUST have the behavior, inputs, outputs and return the faults as defined by the following table.

|  |  |
| --- | --- |
| Name | CloseConsumerRequestSession |
| Description | Closes a consumer request session. |
| Input | SessionID [1] |
| Behavior | If the SessionID does not exist (non-existent or already closed) or does not correspond to a Request session, then a SessionFault is returned.  If the channel associated with the session is assigned security tokens and the provided token does not match a token assigned to the channel, then a SessionFault is returned.  All unexpired requests that have been posted during the session will be expired. |
| Output | N/A |
| Faults | SessionFault |

#### SOAP Mapping

The Close Consumer Request Session general interface is mapped into SOAP 1.1/1.2 as embedded XML schemas in WSDL descriptions according to the following schema types.

The behavior of the SOAP interface MUST conform to that of the general description.

|  |  |
| --- | --- |
| Input | CloseConsumerRequestSession (isbm:CloseConsumerRequestSession)   * SessionID [1] |
| Output | CloseConsumerRequestSessionResponse ([isbm:](http://www.openoandm.org/ws-isbm/1.1/wsdl/ChannelManagementService.wsdl)CloseConsumerRequestSessionResponse)   * No Content |
| Faults | SessionFault |

#### REST Mapping

The Close Consumer Request Session general interface is mapped into a RESTful interface as an OpenAPI description according to the following rules.

The behavior of the REST interface MUST conform to that of the general description with necessary adjustments to conform to REST principles and HTTP specifications.

|  |  |
| --- | --- |
| HTTP Method | DELETE |
| URL | /sessions/{session-id} |
| HTTP Body | None |
| HTTP Response (Success) | 204 No Content |
| Output | None |
| HTTP Response  (Error) | SessionFault (json:SessionFault) – 404 Not Found |

## ISBM Configuration Discovery Service

The ISBM Configuration Discovery Service for SOAP Interface is available as a WSDL description and for REST Interface is available as OpenAPI 3.0.1 descriptions in YAML.

### Get ISBM Configuration

The Get ISBM Configuration in general MUST have the behavior, inputs, outputs, and return the faults as defined by the following table.

|  |  |
| --- | --- |
| Name | XXX |
| Description |  |
| Input |  |
| Behavior |  |
| Output |  |
| Faults |  |

#### SOAP Mapping

The XXX general interface is mapped into SOAP 1.1/1.2 as embedded XML schemas in WSDL descriptions according to the following schema types.

The behavior of the SOAP interface MUST conform to that of the general description.

|  |  |
| --- | --- |
| Input |  |
| Output |  |
| Faults |  |

#### REST Mapping

The XXX general interface is mapped into a RESTful interface as an OpenAPI description according to the following rules.

The behavior of the REST interface MUST conform to that of the general description with necessary adjustments to conform to REST principles and HTTP specifications.

|  |  |
| --- | --- |
| HTTP Method |  |
| URL |  |
| HTTP Body |  |
| HTTP Response (Success) |  |
| Output |  |
| HTTP Response  (Error) |  |

# XML Data Structures

The following data structures are used by the services defined in [Service Definitions](#service-definitions) and are defined using XML Schema. All types have a target namespace of http://www.openoandm.org/isbm/.

## Channel

<xs:complexType name="Channel">  
 <xs:sequence>  
 <xs:element minOccurs="1" maxOccurs="1" name="ChannelURI" type="xs:string"/>  
 <xs:element minOccurs="1" maxOccurs="1" name="ChannelType" type="isbm:ChannelType"/>  
 <xs:element minOccurs="0" maxOccurs="1" name="ChannelDescription" type="xs:string"/>  
 </xs:sequence>  
</xs:complexType>

## ChannelType

<xs:simpleType name="ChannelType">  
 <xs:restriction base="xs:string">  
 <xs:enumeration value="Publication"/>  
 <xs:enumeration value="Request"/>  
 </xs:restriction>  
</xs:simpleType>

Expression

<xs:complexType name="Expression">

  <xs:sequence>

    <xs:element minOccurs="1" maxOccurs="1" name="ExpressionString">

<xs:simpleContent>

<xs:extension base="xs:string">

<!-- Recognized languages/versions are: XPath/1.0; JSONPath -->

<xs:attribute name="language" type="xs:token" use="required"/>

          <xs:attribute name="languageVersion" type="xs:token" use="optional"/>

</xs:extension>

</xs:simpleContent>

</xs:element>

    <xs:element minOccurs="0" maxOccurs="unbounded" name="Namespace" type="isbm:Namespace"/>

  </xs:sequence>

<xs:attribute name="applicableMediaTypes" type="isbm:MediaTypeList" use="optional"/>

</xs:complexType>

<xs:simpleType name="MediaTypeList">

<xs:list itemType="xs:token"/>

</xs:simpleType>

## MessageContent

<xs:complexType name="MessageContent" abstract="true">

</xs:complexType>

<complexType name="BinaryContent">

  <xs:complexContent>

    <xs:extension base="isbm:MessageContent">

      <xs:attribute use="optional" name="mediaType" type="xs:string" />

      <xs:sequence>

        <xs:element minOccurs="1" maxOccurs="1" name="Content" type="xs:base64binary" />

      </xs:sequence>

    </xs:extension>

  </xs:complexContent>

</complexType>

<complexType name="StringContent">

  <xs:complexContent>

    <xs:extension base="isbm:MessageContent">

      <xs:attribute use="required" name="mediaType" type="xs:string" />

      <xs:sequence>

        <xs:element minOccurs="1" maxOccurs="1" name="Content" type="xs:string" />

      </xs:sequence>

    </xs:extension>

  </xs:complexContent>

</complexType>

<complexType name="XMLContent">

  <xs:complexContent>

    <xs:extension base="isbm:MessageContent">

      <xs:sequence>

        <xs:any minOccurs="1" maxOccurs="1" namespace="##any" processContents="lax"/>

      </xs:sequence>

    </xs:extension>

  </xs:complexContent>

</complexType>

## Namespace

<xs:complexType name="Namespace">  
 <xs:sequence>  
 <xs:element minOccurs="1" maxOccurs="1" name="NamespacePrefix" type="xs:string"/>  
 <xs:element minOccurs="1" maxOccurs="1" name="NamespaceName" type="xs:string"/>  
 </xs:sequence>  
</xs:complexType>

## PublicationMessage

<xs:complexType name="PublicationMessage">  
 <xs:sequence>  
 <xs:element minOccurs="1" maxOccurs="1" name="MessageID" type="xs:string"/>  
 <xs:element minOccurs="1" maxOccurs="1" name="MessageContent" type="isbm:MessageContent"/>  
 <xs:element minOccurs="1" maxOccurs="unbounded" name="Topic" type="xs:string"/>  
 </xs:sequence>  
</xs:complexType>

## RequestMessage

<xs:complexType name="RequestMessage">  
 <xs:sequence>  
 <xs:element minOccurs="1" maxOccurs="1" name="MessageID" type="xs:string"/>  
 <xs:element minOccurs="1" maxOccurs="1" name="MessageContent" type="isbm:MessageContent"/>  
 <xs:element minOccurs="1" maxOccurs="1" name="Topic" type="xs:string"/>  
 </xs:sequence>  
</xs:complexType>

## ResponseMessage

<xs:complexType name="ResponseMessage">  
 <xs:sequence>  
 <xs:element minOccurs="1" maxOccurs="1" name="MessageID" type="xs:string"/>  
 <xs:element minOccurs="1" maxOccurs="1" name="MessageContent" type="isbm:MessageContent"/>  
 </xs:sequence>  
</xs:complexType>

## SecurityToken

<xs:complexType name="SecurityToken">  
 <xs:sequence>  
 <xs:any minOccurs="1" maxOccurs="1" namespace="##any" processContents="lax"/>  
 </xs:sequence>  
</xs:complexType>

# JSON Data Structures

The following data structures are used by the services defined in [Service Definitions](#_3_Service_Definitions) and are defined using JSON Schema.

## Channel

"Channel": {

"type": "object",

"properties": {

"uri": {

"type": "string",

"format": "uri"

},

"channelType": {

"$ref": "#/components/schemas/ChannelType"

},

"description": {

"type": "string"

},

"securityTokens": {

"description": "This can be provided when creating a channel but should never be returned.",

"type": "array",

"items": {

"$ref": "#/components/schemas/SecurityToken"

}

}

},

"required": [

"uri",

"channelType"

]

}

## ChannelType

"ChannelType": {

"type": "string",

"enum": [

"Publication",

"Request"

]

}

## Message

"Message": {

"type": "object",

"description": "Message Content may be XML, JSON, or possibly an arbitrary type. However, XML and JSON must be supported. When receiving a Message object as the result of a POST, MUST only include the message ID confirming the creation of the Message.The message type is implicit based on the context and MUST NOT appear in request/response bodies.",

"properties": {

"messageId": {

"type": "string"

},

"messageType": {

"$ref": "#/components/schemas/MessageType"

},

"messageContent": {

"$ref": "#/components/schemas/MessageContent"

},

"topics": {

"description": "The Topic(s) to which the message will be posted.",

"type": "array",

"items": {

"type": "string"

},

"minItems": 1

},

"expiry": {

"type": "string",

"format": "duration",

"description": "The duration after which the message will be automatically expired. Negative duration is no duration. Duration as defined by XML Schema xs:duration, http://w3c.org/TR/xmlschema-2/#duration",

"pattern": "[-]?P([0-9]+Y)?([0-9]+M)?([0-9]+D)?(T([0-9]+H)?([0-9]+M)?([0-9]+([.][0-9]+)?S)?)?"

},

"requestMessageId": {

"type": "string",

"description": "Only valid for Response messages; refers to the original Request message."

}

}

}

## MessageContent

"MessageContent": {

"type": "object",

"properties": {

"mediaType": {

"type": "string",

"description": "The MIME type of the content. If not present, it is assumed to be the same as the Content-Type of the request/response body."

},

"content": {

"type": "object",

"additionalProperties": true

}

},

"required": [

"content"

]

}

## MessageType

"MessageType": {

"type": "string",

"enum": [

"Request",

"Response",

"Publication"

]

}

## SecurityToken

"SecurityToken": {

"type": "object",

"description": "Exact security token types are implementation specific. Support must be provided for at least UsernameToken.",

"anyOf": [

{

"$ref": "#/components/schemas/UsernameToken"

}

],

"additionalProperties": true

}

"UsernameToken": {

"type": "object",

"properties": {

"username": {

"type": "string"

},

"password": {

"type": "string",

"format": "password"

}

},

"required": [

"password",

"username"

]

}

# Security Architecture

The general Service Requirements only provide security requirements related to authenticating operations against the channels on which they will be performed. This section considers security from an inter-enterprise context. It defines 4 levels of security to which ISBM implementations may conform. In this version of the specification, these security levels are introduced and briefly discussed. In future revisions of this specification, the following security levels will be associated with concrete requirements for the services in general and requirements for specific services where necessary.

## Security Level 1 – None

Security Level 1 is characterized by fulfilling no security criteria. That is:

* SSL/TLS are NOT used for transport layer security
* Security tokens are NOT used to secure channels, or tokens are exchanged in the clear without encryption
* Security tokens MAY or MAY NOT be stored encrypted, if used
* Certificates are NOT used for confirming identity

### Usage Scenarios

This security level is NOT RECOMMENDED for production environments. However, it MAY be suitable for use in development and testing environments. It MAY also be used in known restricted environments, such as isolated networks.

## Security Level 2 – Core Security

Security Level 2, Core Security, provides a basic set of security requirements. In contrast to Security level 1, this security level is characterized by providing transport layer security and securing tokens at rest, that is:

* All the communications MUST use transport layer security, e.g., SSL/TLS
* Security tokens MAY be used but MUST be are stored encrypted by the ISBM Service Provider
* Best practices are used to exchange/configure security tokens out-of-band

The Core Security level MAY also utilize Role-Based Access Control for configuring the services and performing their operations.

### Usage Scenarios

The Core Security level MAY be used for production environments and is most appropriate for *intra*-enterprise connectivity. It is NOT RECOMMENDED for environments, even intra-enterprise environments, that require a higher level of security.

## Security Level 3 – Inter-Enterprise Security

Security Level 3, or Inter-Enterprise Security, includes all the requirements of Core Security and adds requirements suitable for an inter-enterprise context. In particular, this level of security is characterized by confirming the identity of interacting systems. The additional requirements to the meet the Inter-Enterprise Security level include:

* Certificates (or other mechanisms) MUST be used to identify ISBM Service Providers and Clients, including Notification endpoints specified by subscription sessions.
* Role-Based Access Control MUST be used for configuration of the Service Provider and performing the operations on the Services.
* All channels MUST be configured with security tokens and non-authenticated operations (GetChannels and CreateChannel) only respond to recognized systems based on the identity check (such as checking their certificates).

At this level of security, ISBM Service Providers MAY use 3rd party services (Key Management Services) to encrypt/decrypt security tokens on demand.

### Usage Scenarios

The Inter-Enterprise Security Level is RECOMMENDED for most Inter-Enterprise scenarios. It is NOT RECOMMENDED for use in highly secure environments that require additional security guarantees.

## Security Level 4 – Defense

The Defense Security Level, Security Level 4, includes all the requirements of Inter-Enterprise Security (and Core Security) with the addition of requirements necessary for highly secure environments, such as Defense. This level of security is characterized by securing the messages and other data within the ISBM Service Provider. Additional requirements for the Defense Security Level include:

* Full end-to-end encryption of messages MUST be performed, that is, the message content is encrypted on the server/s of the ISBM Service Provider
* Security keys (for messages encryption) and security tokens (for channel access) MUST be stored encrypted
* It is RECOMMENDED that 3rd party KMSs (Key Management Services) be used to encrypt/decrypt the security keys and tokens.
* Access to a ISBM Service Provider at this level of security by systems of lower-security levels MUST be performed using appropriate negotiation protocols.

NOTE A typical approach to encrypting a message may be as follows: a random encryption key is created for each message; the message is encrypted using its key and the encrypted message is stored; the key is encrypted using a 3rd party KMS (to prevent the 3rd party from the seeing the confidential messages); the encrypted key is then stored in the database.

### Usage Scenarios

The Defense Security Level is RECOMMENDED for highly secure environments such as those often required by Defense. This may be both intra- and inter-enterprise scenarios depending on the requirements of the deployment.

## Security Level Matrix

The following table summarizes the four levels of security discussed above.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Transport Layer Security | Uses Security Token | Encrypts Security Tokens | Uses Identity Certificate | Role-Based Access Control | 3rd Party KMS | End-to-end encryption? | Suitable for |
| **Security Level 1** | No | Maybe | No | No | No | No | No | Development and Testing environment |
| **Security Level 2** | Yes | Yes | Yes | No | Maybe | No | No | Intra-enterprise connectivity |
| **Security Level 3** | Yes | Yes | Yes | Yes | Yes | Maybe | No | Inter-Enterprise connectivity |
| **Security Level 4** | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Highly secure environments |

# Conformance

Any assessment of conformance of a ISBM implementation MUST be qualified by the following:

1. Support for the Channel Management Service
2. Support for the Notification Service
3. Support for the Provider Publication Service
4. Support for the Consumer Publication Service
5. Support for the Provider Request Service
6. Support for the Consumer Request Service
7. Support for SOAP 1.1 and SOAP 1.2 services
8. Support for HTTP 1.1
9. Support for OpenAPI 3.0.1 services
10. Support for Filter Expressions in an XPath 1.0 format for XML content
11. Support for Filter Expressions in an JSONPath format for JSON content
12. Support for transport layer security (e.g. SSL/TLS) in order to secure tokens and messages, and to prevent replay attacks.
13. Support for Security Tokens using WS-Security UsernameToken
14. Support for HTTP basic and/or digest authentication and authorization
15. Support for other Security Tokens formats (including HTTP authentication/authorization token formats)
16. A statement of the total conformance concerning services and security methods supported or, in case of partial conformance, a statement identifying explicitly the areas of non-conformance
17. Specification Files

The following lists the files containing the Web Services descriptions for SOAP (WSDL format) and REST (OpenAPI format).

* 1. OpenAPI Definitions

<http://www.openoandm.org/isbm/2.0/openapi/channel_management_service.yml>

<http://www.openoandm.org/isbm/2.0/openapi/channel_management_service.json>

<http://www.openoandm.org/isbm/2.0/openapi/notification_service.yml>

<http://www.openoandm.org/isbm/2.0/openapi/notification_service.json>

<http://www.openoandm.org/isbm/2.0/openapi/provider_publication_service.yml>

<http://www.openoandm.org/isbm/2.0/openapi/provider_publication_service.json>

<http://www.openoandm.org/isbm/2.0/openapi/consumer_publication_service.yml>

<http://www.openoandm.org/isbm/2.0/openapi/consumer_publication_service.json>

<http://www.openoandm.org/isbm/2.0/openapi/provider_request_service.yml>

<http://www.openoandm.org/isbm/2.0/openapi/provider_request_service.json>

<http://www.openoandm.org/isbm/2.0/openapi/consumer_request_service.yml>

<http://www.openoandm.org/isbm/2.0/openapi/consumer_request_service.json>

<http://www.openoandm.org/isbm/2.0/openapi/isbm_complete.yml>

<http://www.openoandm.org/isbm/2.0/openapi/isbm_complete.json>

* 1. WSDLs

<http://www.openoandm.org/isbm/2.0/wsdl/ChannelManagementService.wsdl>

<http://www.openoandm.org/isbm/2.0/wsdl/NotificationService.wsdl>

<http://www.openoandm.org/isbm/2.0/wsdl/ProviderPublicationService.wsdl>

<http://www.openoandm.org/isbm/2.0/wsdl/ConsumerPublicationService.wsdl>

<http://www.openoandm.org/isbm/2.0/wsdl/ProviderRequestService.wsdl>

<http://www.openoandm.org/isbm/2.0/wsdl/ConsumerRequestService.wsdl>

* 1. Packaged Specification

http://www.openoandm.org/isbm/isbm-soap-2.0.zip

http://www.openoandm.org/isbm/isbm-rest-2.0.zip

http://www.openoandm.org/isbm/isbm-all-2.0.zip

1. Example HTTP Flows
   1. Channel Management Example



* + 1. CreateChannel

The Application creates a channel on the ISBM Service Provider and assigns a WS-Security security token.

NOTE XML special characters must be escaped, as seen with the **<** character in the Password element.

HTTP Request

POST /ChannelManagementService HTTP/1.1  
Host: ws-isbm.example.com  
Content-Type: text/xml; charset=utf-8  
Content-Length: 705  
SOAPAction: "http://www.openoandm.org/ws-isbm/CreateChannel"  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
 <soap:Body>  
 <isbm:CreateChannel xmlns:isbm="http://www.openoandm.org/ws-isbm/">  
 <isbm:ChannelURI>/Enterprise/Site/Area/WorkCenter</isbm:ChannelURI>  
 <isbm:ChannelType>Publication</isbm:ChannelType>  
 <isbm:SecurityToken>  
 <wsse:UsernameToken xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">  
 <wsse:Username>Application1</wsse:Username>  
 <wsse:Password>&lt;s9.vQfLDx9LgL</wsse:Password>  
 </wsse:UsernameToken>  
 </isbm:SecurityToken>  
 </isbm:CreateChannel>  
 </soap:Body>  
</soap:Envelope>

HTTP Response

HTTP/1.1 200 OK  
Content-Type: text/xml; charset=utf-8  
Content-Length: 238  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
 <soap:Body>  
 <isbm:CreateChannelResponse xmlns:isbm="http://www.openoandm.org/ws-isbm/"/>  
 </soap:Body>  
</soap:Envelope>

* + 1. AddSecurityToken

The Application assigns an additional security token to the channel.

HTTP Request

POST /ChannelManagementService HTTP/1.1  
Host: ws-isbm.example.com  
Content-Type: text/xml; charset=utf-8  
Content-Length: 892  
SOAPAction: "http://www.openoandm.org/ws-isbm/AddSecurityToken"  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">  
 <soap:Header>  
 <wsse:Security>  
 <wsse:UsernameToken>  
 <wsse:Username>Application1</wsse:Username>  
 <wsse:Password>&lt;s9.vQfLDx9LgL</wsse:Password>  
 </wsse:UsernameToken>  
 </wsse:Security>  
 </soap:Header>  
 <soap:Body>  
 <isbm:AddSecurityToken xmlns:isbm="http://www.openoandm.org/ws-isbm/">  
 <isbm:ChannelURI>/Enterprise/Site/Area/WorkCenter</isbm:ChannelURI>  
 <isbm:SecurityToken>  
 <wsse:UsernameToken>  
 <wsse:Username>Application2</wsse:Username>  
 <wsse:Password>chHM?rFum{48mg</wsse:Password>  
 </wsse:UsernameToken>  
 </isbm:SecurityToken>  
 </isbm:AddSecurityToken>  
 </soap:Body>  
</soap:Envelope>

HTTP Response

HTTP/1.1 200 OK  
Content-Type: text/xml; charset=utf-8  
Content-Length: 241  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
 <soap:Body>  
 <isbm:AddSecurityTokenResponse xmlns:isbm="http://www.openoandm.org/ws-isbm/"/>  
 </soap:Body>  
</soap:Envelope>

* + 1. RemoveSecurityToken

The Application removes the original security token from the channel.

HTTP Request

POST /ChannelManagementService HTTP/1.1  
Host: ws-isbm.example.com  
Content-Type: text/xml; charset=utf-8  
Content-Length: 898  
SOAPAction: "http://www.openoandm.org/ws-isbm/RemoveSecurityToken"  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">  
 <soap:Header>  
 <wsse:Security>  
 <wsse:UsernameToken>  
 <wsse:Username>Application2</wsse:Username>  
 <wsse:Password>chHM?rFum{48mg</wsse:Password>  
 </wsse:UsernameToken>  
 </wsse:Security>  
 </soap:Header>  
 <soap:Body>  
 <isbm:RemoveSecurityToken xmlns:isbm="http://www.openoandm.org/ws-isbm/">  
 <isbm:ChannelURI>/Enterprise/Site/Area/WorkCenter</isbm:ChannelURI>  
 <isbm:SecurityToken>  
 <wsse:UsernameToken>  
 <wsse:Username>Application1</wsse:Username>  
 <wsse:Password>&lt;s9.vQfLDx9LgL</wsse:Password>  
 </wsse:UsernameToken>  
 </isbm:SecurityToken>  
 </isbm:RemoveSecurityToken>  
 </soap:Body>  
</soap:Envelope>

HTTP Response

HTTP/1.1 200 OK  
Content-Type: text/xml; charset=utf-8  
Content-Length: 244  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
 <soap:Body>  
 <isbm:RemoveSecurityTokenResponse xmlns:isbm="http://www.openoandm.org/ws-isbm/"/>  
 </soap:Body>  
</soap:Envelope>

* + 1. GetChannel

The Application attempts to retrieve channel information using the original security token and receives an authorization failure.

HTTP Request

POST /ChannelManagementService HTTP/1.1  
Host: ws-isbm.example.com  
Content-Type: text/xml; charset=utf-8  
Content-Length: 656  
SOAPAction: "http://www.openoandm.org/ws-isbm/GetChannel"  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
 <soap:Header>  
 <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">  
 <wsse:UsernameToken>  
 <wsse:Username>Application1</wsse:Username>  
 <wsse:Password>&lt;s9.vQfLDx9LgL</wsse:Password>  
 </wsse:UsernameToken>  
 </wsse:Security>  
 </soap:Header>  
 <soap:Body>  
 <isbm:GetChannel xmlns:isbm="http://www.openoandm.org/ws-isbm/">  
 <isbm:ChannelURI>/Enterprise/Site/Area/WorkCenter</isbm:ChannelURI>  
 </isbm:GetChannel>  
 </soap:Body>  
</soap:Envelope>

HTTP Response

HTTP/1.1 500 Internal Server Error  
Content-Type: text/xml; charset=utf-8  
Content-Length: 401  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
 <soap:Body>  
 <soap:Fault>  
 <faultcode>soap:Client</faultcode>  
 <faultstring>Channel is not accessible.</faultstring>  
 <detail>  
 <isbm:ChannelFault xmlns:isbm="http://www.openoandm.org/ws-isbm/"/>  
 </detail>  
 </soap:Fault>  
 </soap:Body>  
</soap:Envelope>

* + 1. GetChannels

The Application retrieves information about channels filtered by the newly assigned security token.

HTTP Request

POST /ChannelManagementService HTTP/1.1  
Host: ws-isbm.example.com  
Content-Type: text/xml; charset=utf-8  
Content-Length: 559  
SOAPAction: "http://www.openoandm.org/ws-isbm/GetChannels"  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
 <soap:Header>  
 <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">  
 <wsse:UsernameToken>  
 <wsse:Username>Application2</wsse:Username>  
 <wsse:Password>chHM?rFum{48mg</wsse:Password>  
 </wsse:UsernameToken>  
 </wsse:Security>  
 </soap:Header>  
 <soap:Body>  
 <isbm:GetChannels xmlns:isbm="http://www.openoandm.org/ws-isbm/"/>  
 </soap:Body>  
</soap:Envelope>

HTTP Response

HTTP/1.1 200 OK  
Content-Type: text/xml; charset=utf-8  
Content-Length: 442  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
 <soap:Body>  
 <isbm:GetChannelsResponse xmlns:isbm="http://www.openoandm.org/ws-isbm/">  
 <isbm:Channel>  
 <isbm:ChannelURI>/Enterprise/Site/Area/WorkCenter</isbm:ChannelURI>  
 <isbm:ChannelType>Publication</isbm:ChannelType>  
 </isbm:Channel>  
 </isbm:GetChannelsResponse>  
 </soap:Body>  
</soap:Envelope>

* + 1. DeleteChannel

The Application removes the channel from the ws-ISBM Service Provider.

HTTP Request

POST /ChannelManagementService HTTP/1.1  
Host: ws-isbm.example.com  
Content-Type: text/xml; charset=utf-8  
Content-Length: 659  
SOAPAction: "http://www.openoandm.org/ws-isbm/DeleteChannel"  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
 <soap:Header>  
 <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">  
 <wsse:UsernameToken>  
 <wsse:Username>Application2</wsse:Username>  
 <wsse:Password>chHM?rFum{48mg</wsse:Password>  
 </wsse:UsernameToken>  
 </wsse:Security>  
 </soap:Header>  
 <soap:Body>  
 <isbm:DeleteChannel xmlns:isbm="http://www.openoandm.org/ws-isbm/">  
 <isbm:ChannelURI>/Enterprise/Site/Area/WorkCenter</isbm:ChannelURI>  
 </isbm:DeleteChannel>  
 </soap:Body>  
</soap:Envelope>

HTTP Response

HTTP/1.1 200 OK  
Content-Type: text/xml; charset=utf-8  
Content-Length: 238  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
 <soap:Body>  
 <isbm:DeleteChannelResponse xmlns:isbm="http://www.openoandm.org/ws-isbm/"/>  
 </soap:Body>  
</soap:Envelope>

* 1. Publish-Subscribe Example



* + 1. OpenSubscriptionSession

The Consumer Application opens a subscription session with the ISBM Service Provider and receives a session identifier.

HTTP Request

POST /ConsumerPublicationService HTTP/1.1  
Host: ws-isbm.example.com  
Content-Type: text/xml; charset=utf-8  
Content-Length: 812  
SOAPAction: "http://www.openoandm.org/ws-isbm/OpenSubscriptionSession"  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
 <soap:Header>  
 <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">  
 <wsse:UsernameToken>  
 <wsse:Username>ConsumerApplication</wsse:Username>  
 <wsse:Password>Dj8(bCU)4bnhjc</wsse:Password>  
 </wsse:UsernameToken>  
 </wsse:Security>  
 </soap:Header>  
 <soap:Body>  
 <isbm:OpenSubscriptionSession xmlns:isbm="http://www.openoandm.org/ws-isbm/">  
 <isbm:ChannelURI>/Enterprise/Site/Area/WorkCenter</isbm:ChannelURI>  
 <isbm:Topic>Text</isbm:Topic>  
 <isbm:ListenerURL>http://consumer.example.com/NotificationService</isbm:ListenerURL>  
 </isbm:OpenPublicationSession>  
 </soap:Body>  
</soap:Envelope>

HTTP Response

HTTP/1.1 200 OK  
Content-Type: text/xml; charset=utf-8  
Content-Length: 366  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
 <soap:Body>  
 <isbm:OpenSubscriptionSessionResponse xmlns:isbm="http://www.openoandm.org/ws-isbm/">  
 <isbm:SessionID>e94c645a-6450-411e-8ec7-4b70620d3a98</isbm:SessionID>  
 </isbm:OpenSubscriptionSessionResponse>  
 </soap:Body>  
</soap:Envelope>

* + 1. OpenPublicationSession

The Provider Application opens a publication session with the ISBM Service Provider and receives a session identifier.

HTTP Request

POST /ProviderPublicationService HTTP/1.1  
Host: ws-isbm.example.com  
Content-Type: text/xml; charset=utf-8  
Content-Length: 684  
SOAPAction: "http://www.openoandm.org/ws-isbm/OpenPublicationSession"  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
 <soap:Header>  
 <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">  
 <wsse:UsernameToken>  
 <wsse:Username>ProviderApplication</wsse:Username>  
 <wsse:Password>qEJaz4F?U4rW;q</wsse:Password>  
 </wsse:UsernameToken>  
 </wsse:Security>  
 </soap:Header>  
 <soap:Body>  
 <isbm:OpenPublicationSession xmlns:isbm="http://www.openoandm.org/ws-isbm/">  
 <isbm:ChannelURI>/Enterprise/Site/Area/WorkCenter</isbm:ChannelURI>  
 </isbm:OpenPublicationSession>  
 </soap:Body>  
</soap:Envelope>

HTTP Response

HTTP/1.1 200 OK  
Content-Type: text/xml; charset=utf-8  
Content-Length: 364  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
 <soap:Body>  
 <isbm:OpenPublicationSessionResponse xmlns:isbm="http://www.openoandm.org/ws-isbm/">  
 <isbm:SessionID>ac0ee730-ca88-421a-b348-ce0a1babdb1c</isbm:SessionID>  
 </isbm:OpenPublicationSessionResponse>  
 </soap:Body>  
</soap:Envelope>

* + 1. PostPublication

The Provider Application posts a publication message to the ISBM Service Provider and receives a message identifier.

HTTP Request

POST /ProviderPublicationService HTTP/1.1  
Host: ws-isbm.example.com  
Content-Type: text/xml; charset=utf-8  
Content-Length: 799  
SOAPAction: "http://www.openoandm.org/ws-isbm/PostPublication"  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
 <soap:Header>  
 <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">  
 <wsse:UsernameToken>  
 <wsse:Username>ProviderApplication</wsse:Username>  
 <wsse:Password>qEJaz4F?U4rW;q</wsse:Password>  
 </wsse:UsernameToken>  
 </wsse:Security>  
 </soap:Header>  
 <soap:Body>  
 <isbm:PostPublication xmlns:isbm="http://www.openoandm.org/ws-isbm/">  
 <isbm:SessionID>ac0ee730-ca88-421a-b348-ce0a1babdb1c</isbm:SessionID>  
 <isbm:MessageContent>  
 <Text>Hello World!</Text>  
 </isbm:MessageContent>  
 <isbm:Topic>Text</isbm:Topic>  
 </isbm:PostPublication>  
 </soap:Body>  
</soap:Envelope>

HTTP Response

HTTP/1.1 200 OK  
Content-Type: text/xml; charset=utf-8  
Content-Length: 350  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
 <soap:Body>  
 <isbm:PostPublicationResponse xmlns:isbm="http://www.openoandm.org/ws-isbm/">  
 <isbm:MessageID>8007a3fa-70e3-4e90-a2b9-d8469cae2e5a</isbm:MessageID>  
 </isbm:PostPublicationResponse>  
 </soap:Body>  
</soap:Envelope>

* + 1. NotifyListener

The ISBM Service Provider notifies the Consumer Application of an applicable publication message.

HTTP Request

POST /NotifyListener HTTP/1.1  
Host: consumer.example.com  
Content-Type: text/xml; charset=utf-8  
Content-Length: 444  
SOAPAction: "http://www.openoandm.org/ws-isbm/NotifyListener"  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
 <soap:Body>  
 <isbm:NotifyListener xmlns:isbm="http://www.openoandm.org/ws-isbm/">  
 <isbm:SessionID>e94c645a-6450-411e-8ec7-4b70620d3a98</isbm:SessionID>  
 <isbm:MessageID>8007a3fa-70e3-4e90-a2b9-d8469cae2e5a</isbm:MessageID>  
 <isbm:Topic>Text</isbm:Topic>  
 </isbm:NotifyListener>  
 </soap:Body>  
</soap:Envelope>

HTTP Response

HTTP/1.1 200 OK  
Content-Type: text/xml; charset=utf-8  
Content-Length: 239  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
 <soap:Body>  
 <isbm:NotifyListenerResponse xmlns:isbm="http://www.openoandm.org/ws-isbm/"/>  
 </soap:Body>  
</soap:Envelope>

* + 1. ReadPublication

The Consumer Application reads the publication message from the ISBM Service Provider.

HTTP Request

POST /ConsumerPublicationService HTTP/1.1  
Host: ws-isbm.example.com  
Content-Type: text/xml; charset=utf-8  
Content-Length: 672  
SOAPAction: "http://www.openoandm.org/ws-isbm/ReadPublication"  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
 <soap:Header>  
 <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">  
 <wsse:UsernameToken>  
 <wsse:Username>ConsumerApplication</wsse:Username>  
 <wsse:Password>Dj8(bCU)4bnhjc</wsse:Password>  
 </wsse:UsernameToken>  
 </wsse:Security>  
 </soap:Header>  
 <soap:Body>  
 <isbm:ReadPublication xmlns:isbm="http://www.openoandm.org/ws-isbm/">  
 <isbm:SessionID>e94c645a-6450-411e-8ec7-4b70620d3a98</isbm:SessionID>  
 </isbm:ReadPublication>  
 </soap:Body>  
</soap:Envelope>

HTTP Response

HTTP/1.1 200 OK  
Content-Type: text/xml; charset=utf-8  
Content-Length: 552  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
 <soap:Body>  
 <isbm:ReadPublicationResponse xmlns:isbm="http://www.openoandm.org/ws-isbm/">  
 <isbm:PublicationMessage>  
 <isbm:MessageID>8007a3fa-70e3-4e90-a2b9-d8469cae2e5a</isbm:MessageID>  
 <isbm:MessageContent>  
 <Text>Hello World!</Text>  
 </isbm:MessageContent>  
 <isbm:Topic>Text</isbm:Topic>  
 </isbm:PublicationMessage>  
 </isbm:ReadPublicationResponse>  
 </soap:Body>  
</soap:Envelope>

* + 1. ExpirePublication

The Provider Application manually expires the publication message from the ISBM Service Provider. The message is still visible to the Consumer Application since it has already been read.

HTTP Request

POST /ProviderPublicationService HTTP/1.1  
Host: ws-isbm.example.com  
Content-Type: text/xml; charset=utf-8  
Content-Length: 752  
SOAPAction: "http://www.openoandm.org/ws-isbm/ExpirePublication"  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
 <soap:Header>  
 <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">  
 <wsse:UsernameToken>  
 <wsse:Username>ProviderApplication</wsse:Username>  
 <wsse:Password>qEJaz4F?U4rW;q</wsse:Password>  
 </wsse:UsernameToken>  
 </wsse:Security>  
 </soap:Header>  
 <soap:Body>  
 <isbm:ExpirePublication xmlns:isbm="http://www.openoandm.org/ws-isbm/">  
 <isbm:SessionID>ac0ee730-ca88-421a-b348-ce0a1babdb1c</isbm:SessionID>  
 <isbm:MessageID>8007a3fa-70e3-4e90-a2b9-d8469cae2e5a</isbm:MessageID>  
 </isbm:ExpirePublication>  
 </soap:Body>  
</soap:Envelope>

HTTP Response

HTTP/1.1 200 OK  
Content-Type: text/xml; charset=utf-8  
Content-Length: 242  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
 <soap:Body>  
 <isbm:ExpirePublicationResponse xmlns:isbm="http://www.openoandm.org/ws-isbm/"/>  
 </soap:Body>  
</soap:Envelope>

* + 1. RemovePublication

The Consumer Application removes the publication message from the ISBM Service Provider.

HTTP Request

POST /ConsumerPublicationService HTTP/1.1  
Host: ws-isbm.example.com  
Content-Type: text/xml; charset=utf-8  
Content-Length: 677  
SOAPAction: "http://www.openoandm.org/ws-isbm/RemovePublication"  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
 <soap:Header>  
 <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">  
 <wsse:UsernameToken>  
 <wsse:Username>ConsumerApplication</wsse:Username>  
 <wsse:Password>Dj8(bCU)4bnhjc</wsse:Password>  
 </wsse:UsernameToken>  
 </wsse:Security>  
 </soap:Header>  
 <soap:Body>  
 <isbm:RemovePublication xmlns:isbm="http://www.openoandm.org/ws-isbm/">  
 <isbm:SessionID>e94c645a-6450-411e-8ec7-4b70620d3a98<</isbm:SessionID>  
 </isbm:RemovePublication>  
 </soap:Body>  
</soap:Envelope>

HTTP Response

HTTP/1.1 200 OK  
Content-Type: text/xml; charset=utf-8  
Content-Length: 242  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
 <soap:Body>  
 <isbm:RemovePublicationResponse xmlns:isbm="http://www.openoandm.org/ws-isbm/"/>  
 </soap:Body>  
</soap:Envelope>

* + 1. ClosePublicationSession

The Provider Application closes the publication session with the ISBM Service Provider.

HTTP Request

POST /ProviderPublicationService HTTP/1.1  
Host: ws-isbm.example.com  
Content-Type: text/xml; charset=utf-8  
Content-Length: 688  
SOAPAction: "http://www.openoandm.org/ws-isbm/ClosePublicationSession"  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
 <soap:Header>  
 <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">  
 <wsse:UsernameToken>  
 <wsse:Username>ProviderApplication</wsse:Username>  
 <wsse:Password>qEJaz4F?U4rW;q</wsse:Password>  
 </wsse:UsernameToken>  
 </wsse:Security>  
 </soap:Header>  
 <soap:Body>  
 <isbm:ClosePublicationSession xmlns:isbm="http://www.openoandm.org/ws-isbm/">  
 <isbm:SessionID>ac0ee730-ca88-421a-b348-ce0a1babdb1c</isbm:SessionID>  
 </isbm:ClosePublicationSession>  
 </soap:Body>  
</soap:Envelope>

HTTP Response

HTTP/1.1 200 OK  
Content-Type: text/xml; charset=utf-8  
Content-Length: 248  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
 <soap:Body>  
 <isbm:ClosePublicationSessionResponse xmlns:isbm="http://www.openoandm.org/ws-isbm/"/>  
 </soap:Body>  
</soap:Envelope>

* + 1. CloseSubscriptionSession

The Consumer Application closes the subscription session with the ISBM Service Provider.

HTTP Request

POST /ConsumerPublicationService HTTP/1.1  
Host: ws-isbm.example.com  
Content-Type: text/xml; charset=utf-8  
Content-Length: 690  
SOAPAction: "http://www.openoandm.org/ws-isbm/ClosePublicationSession"  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
 <soap:Header>  
 <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">  
 <wsse:UsernameToken>  
 <wsse:Username>ConsumerApplication</wsse:Username>  
 <wsse:Password>Dj8(bCU)4bnhjc</wsse:Password>  
 </wsse:UsernameToken>  
 </wsse:Security>  
 </soap:Header>  
 <soap:Body>  
 <isbm:CloseSubscriptionSession xmlns:isbm="http://www.openoandm.org/ws-isbm/">  
 <isbm:SessionID>e94c645a-6450-411e-8ec7-4b70620d3a98</isbm:SessionID>  
 </isbm:CloseSubscriptionSession>  
 </soap:Body>  
</soap:Envelope>

HTTP Response

HTTP/1.1 200 OK  
Content-Type: text/xml; charset=utf-8  
Content-Length: 249  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
 <soap:Body>  
 <isbm:CloseSubscriptionSessionResponse xmlns:isbm="http://www.openoandm.org/ws-isbm/"/>  
 </soap:Body>  
</soap:Envelope>

* 1. Request-Response Example



* + 1. OpenProviderRequestSession

The Provider Application opens a provider request session with the ISBM Service Provider and receives a session identifier.

HTTP Request

POST /ProviderRequestService HTTP/1.1  
Host: ws-isbm.example.com  
Content-Type: text/xml; charset=utf-8  
Content-Length: 819  
SOAPAction: "http://www.openoandm.org/ws-isbm/OpenProviderRequestSession"  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
 <soap:Header>  
 <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">  
 <wsse:UsernameToken>  
 <wsse:Username>ProviderApplication</wsse:Username>  
 <wsse:Password>9gy#gXENxph8?W</wsse:Password>  
 </wsse:UsernameToken>  
 </wsse:Security>  
 </soap:Header>  
 <soap:Body>  
 <isbm:OpenProviderRequestSession xmlns:isbm="http://www.openoandm.org/ws-isbm/">  
 <isbm:ChannelURI>/Enterprise/Site/Area/WorkCenter</isbm:ChannelURI>  
 <isbm:Topic>Text</isbm:Topic>  
 <isbm:ListenerURL>http://provider.example.com/NotificationService</isbm:ListenerURL>  
 </isbm:OpenProviderRequestSession>  
 </soap:Body>  
</soap:Envelope>

HTTP Response

HTTP/1.1 200 OK  
Content-Type: text/xml; charset=utf-8  
Content-Length: 372  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
 <soap:Body>  
 <isbm:OpenProviderRequestSessionResponse xmlns:isbm="http://www.openoandm.org/ws-isbm/">  
 <isbm:SessionID>16ee00f6-8fa9-4e80-8c36-a9d6d2bdb551</isbm:SessionID>  
 </isbm:OpenProviderRequestSessionResponse>  
 </soap:Body>  
</soap:Envelope>

* + 1. OpenConsumerRequestSession

The Consumer Application opens a consumer request session with the ISBM Service Provider and receives a session identifier.

HTTP Request

POST /ConsumerRequestService HTTP/1.1  
Host: ws-isbm.example.com  
Content-Type: text/xml; charset=utf-8  
Content-Length: 783  
SOAPAction: "http://www.openoandm.org/ws-isbm/OpenConsumerRequestSession"  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
 <soap:Header>  
 <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">  
 <wsse:UsernameToken>  
 <wsse:Username>ConsumerApplication</wsse:Username>  
 <wsse:Password>^Um.7oFM9jrnnC</wsse:Password>  
 </wsse:UsernameToken>  
 </wsse:Security>  
 </soap:Header>  
 <soap:Body>  
 <isbm:OpenConsumerRequestSession xmlns:isbm="http://www.openoandm.org/ws-isbm/">  
 <isbm:ChannelURI>/Enterprise/Site/Area/WorkCenter</isbm:ChannelURI>  
 <isbm:ListenerURL>http://consumer.example.com/NotificationService</isbm:ListenerURL>  
 </isbm:OpenConsumerRequestSession>  
 </soap:Body>  
</soap:Envelope>

HTTP Response

HTTP/1.1 200 OK  
Content-Type: text/xml; charset=utf-8  
Content-Length: 372  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
 <soap:Body>  
 <isbm:OpenConsumerRequestSessionResponse xmlns:isbm="http://www.openoandm.org/ws-isbm/">  
 <isbm:SessionID>a9b5c71a-d1b5-4fc8-81d1-ba1fee3af0df</isbm:SessionID>  
 </isbm:OpenConsumerRequestSessionResponse>  
 </soap:Body>  
</soap:Envelope>

* + 1. PostRequest

The Consumer Application posts a request message to the ISBM Service Provider and receives a message identifier.

HTTP Request

POST /ConsumerRequestService HTTP/1.1  
Host: ws-isbm.example.com  
Content-Type: text/xml; charset=utf-8  
Content-Length: 748  
SOAPAction: "http://www.openoandm.org/ws-isbm/PostRequest"  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
 <soap:Header>  
 <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">  
 <wsse:UsernameToken>  
 <wsse:Username>ConsumerApplication</wsse:Username>  
 <wsse:Password>^Um.7oFM9jrnnC</wsse:Password>  
 </wsse:UsernameToken>  
 </wsse:Security>  
 </soap:Header>  
 <soap:Body>  
 <isbm:PostRequest xmlns:isbm="http://www.openoandm.org/ws-isbm/">  
 <isbm:SessionID>a9b5c71a-d1b5-4fc8-81d1-ba1fee3af0df</isbm:SessionID>  
 <isbm:MessageContent>  
 <Text>Ping!</Text>  
 </isbm:MessageContent>  
 </isbm:PostRequest>  
 </soap:Body>  
</soap:Envelope>

HTTP Response

HTTP/1.1 200 OK  
Content-Type: text/xml; charset=utf-8  
Content-Length: 342  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
 <soap:Body>  
 <isbm:PostRequestResponse xmlns:isbm="http://www.openoandm.org/ws-isbm/">  
 <isbm:MessageID>e8cfecb1-d2fc-4167-88f7-c90d60fc53ee</isbm:MessageID>  
 </isbm:PostRequestResponse>  
 </soap:Body>  
</soap:Envelope>

* + 1. NotifyListener

The ISBM Service Provider notifies the Provider Application of an applicable request message.

HTTP Request

POST /NotifyListener HTTP/1.1  
Host: provider.example.com  
Content-Type: text/xml; charset=utf-8  
Content-Length: 444  
SOAPAction: "http://www.openoandm.org/ws-isbm/NotifyListener"  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
 <soap:Body>  
 <isbm:NotifyListener xmlns:isbm="http://www.openoandm.org/ws-isbm/">  
 <isbm:SessionID>16ee00f6-8fa9-4e80-8c36-a9d6d2bdb551</isbm:SessionID>  
 <isbm:MessageID>e8cfecb1-d2fc-4167-88f7-c90d60fc53ee</isbm:MessageID>  
 <isbm:Topic>Text</isbm:Topic>  
 </isbm:NotifyListener>  
 </soap:Body>  
</soap:Envelope>

HTTP Response

HTTP/1.1 200 OK  
Content-Type: text/xml; charset=utf-8  
Content-Length: 239  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
 <soap:Body>  
 <isbm:NotifyListenerResponse xmlns:isbm="http://www.openoandm.org/ws-isbm/"/>  
 </soap:Body>  
</soap:Envelope>

* + 1. ReadRequest

The Provider Application reads the request message from the ISBM Service Provider.

HTTP Request

POST /ProviderRequestService HTTP/1.1  
Host: ws-isbm.example.com  
Content-Type: text/xml; charset=utf-8  
Content-Length: 664  
SOAPAction: "http://www.openoandm.org/ws-isbm/ReadRequest"  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
 <soap:Header>  
 <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">  
 <wsse:UsernameToken>  
 <wsse:Username>ProviderApplication</wsse:Username>  
 <wsse:Password>9gy#gXENxph8?W</wsse:Password>  
 </wsse:UsernameToken>  
 </wsse:Security>  
 </soap:Header>  
 <soap:Body>  
 <isbm:ReadRequest xmlns:isbm="http://www.openoandm.org/ws-isbm/">  
 <isbm:SessionID>16ee00f6-8fa9-4e80-8c36-a9d6d2bdb551</isbm:SessionID>  
 </isbm:ReadRequest>  
 </soap:Body>  
</soap:Envelope>

HTTP Response

HTTP/1.1 200 OK  
Content-Type: text/xml; charset=utf-8  
Content-Length: 529  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
 <soap:Body>  
 <isbm:ReadRequestResponse xmlns:isbm="http://www.openoandm.org/ws-isbm/">  
 <isbm:RequestMessage>  
 <isbm:MessageID>e8cfecb1-d2fc-4167-88f7-c90d60fc53ee</isbm:MessageID>  
 <isbm:MessageContent>  
 <Text>Ping!</Text>  
 </isbm:MessageContent>  
 <isbm:Topic>Text</isbm:Topic>  
 </isbm:RequestMessage>  
 </isbm:ReadRequestResponse>  
 </soap:Body>  
</soap:Envelope>

* + 1. RemoveRequest

The Provider Application removes the request message from the ISBM Service Provider.

HTTP Request

POST /ProviderRequestService HTTP/1.1  
Host: ws-isbm.example.com  
Content-Type: text/xml; charset=utf-8  
Content-Length: 669  
SOAPAction: "http://www.openoandm.org/ws-isbm/RemoveRequest"  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
 <soap:Header>  
 <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">  
 <wsse:UsernameToken>  
 <wsse:Username>ProviderApplication</wsse:Username>  
 <wsse:Password>9gy#gXENxph8?W</wsse:Password>  
 </wsse:UsernameToken>  
 </wsse:Security>  
 </soap:Header>  
 <soap:Body>  
 <isbm:RemoveRequest xmlns:isbm="http://www.openoandm.org/ws-isbm/">  
 <isbm:SessionID>16ee00f6-8fa9-4e80-8c36-a9d6d2bdb551<</isbm:SessionID>  
 </isbm:RemoveRequest>  
 </soap:Body>  
</soap:Envelope>

HTTP Response

HTTP/1.1 200 OK  
Content-Type: text/xml; charset=utf-8  
Content-Length: 238  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
 <soap:Body>  
 <isbm:RemoveRequestResponse xmlns:isbm="http://www.openoandm.org/ws-isbm/"/>  
 </soap:Body>  
</soap:Envelope>

* + 1. PostResponse

The Provider Application posts a response message to the ISBM Service Provider.

HTTP Request

POST /ProviderRequestService HTTP/1.1  
Host: ws-isbm.example.com  
Content-Type: text/xml; charset=utf-8  
Content-Length: 840  
SOAPAction: "http://www.openoandm.org/ws-isbm/PostResponse"  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
 <soap:Header>  
 <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">  
 <wsse:UsernameToken>  
 <wsse:Username>ProviderApplication</wsse:Username>  
 <wsse:Password>9gy#gXENxph8?W</wsse:Password>  
 </wsse:UsernameToken>  
 </wsse:Security>  
 </soap:Header>  
 <soap:Body>  
 <isbm:PostResponse xmlns:isbm="http://www.openoandm.org/ws-isbm/">  
 <isbm:SessionID>16ee00f6-8fa9-4e80-8c36-a9d6d2bdb551</isbm:SessionID>  
 <isbm:RequestMessageID>e8cfecb1-d2fc-4167-88f7-c90d60fc53ee</isbm:RequestMessageID>  
 <isbm:MessageContent>  
 <Text>Pong!</Text>  
 </isbm:MessageContent>  
 </isbm:PostResponse>  
 </soap:Body>  
</soap:Envelope>

HTTP Response

HTTP/1.1 200 OK  
Content-Type: text/xml; charset=utf-8  
Content-Length: 237  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
 <soap:Body>  
 <isbm:PostResponseResponse xmlns:isbm="http://www.openoandm.org/ws-isbm/"/>  
 </soap:Body>  
</soap:Envelope>

* + 1. NotifyListener

The ISBM Service Provider notifies the Consumer Application of an applicable response message.

HTTP Request

POST /NotifyListener HTTP/1.1  
Host: consumer.example.com  
Content-Type: text/xml; charset=utf-8  
Content-Length: 498  
SOAPAction: "http://www.openoandm.org/ws-isbm/NotifyListener"  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
 <soap:Body>  
 <isbm:NotifyListener xmlns:isbm="http://www.openoandm.org/ws-isbm/">  
 <isbm:SessionID>a9b5c71a-d1b5-4fc8-81d1-ba1fee3af0df</isbm:SessionID>  
 <isbm:MessageID>af250a33-d5af-4c25-bb57-56802d8fea79</isbm:MessageID>  
 <isbm:RequestMessageID>e8cfecb1-d2fc-4167-88f7-c90d60fc53ee</isbm:RequestMessageID>  
 </isbm:NotifyListener>  
 </soap:Body>  
</soap:Envelope>

HTTP Response

HTTP/1.1 200 OK  
Content-Type: text/xml; charset=utf-8  
Content-Length: 239  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
 <soap:Body>  
 <isbm:NotifyListenerResponse xmlns:isbm="http://www.openoandm.org/ws-isbm/"/>  
 </soap:Body>  
</soap:Envelope>

* + 1. ReadResponse

The Consumer Application reads the response message from the ISBM Service Provider.

HTTP Request

POST /ConsumerRequestService HTTP/1.1  
Host: ws-isbm.example.com  
Content-Type: text/xml; charset=utf-8  
Content-Length: 756  
SOAPAction: "http://www.openoandm.org/ws-isbm/ReadResponse"  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
 <soap:Header>  
 <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">  
 <wsse:UsernameToken>  
 <wsse:Username>ConsumerApplication</wsse:Username>  
 <wsse:Password>^Um.7oFM9jrnnC</wsse:Password>  
 </wsse:UsernameToken>  
 </wsse:Security>  
 </soap:Header>  
 <soap:Body>  
 <isbm:ReadResponse xmlns:isbm="http://www.openoandm.org/ws-isbm/">  
 <isbm:SessionID>a9b5c71a-d1b5-4fc8-81d1-ba1fee3af0df</isbm:SessionID>  
 <isbm:RequestMessageID>e8cfecb1-d2fc-4167-88f7-c90d60fc53ee</isbm:RequestMessageID>  
 </isbm:ReadResponse>  
 </soap:Body>  
</soap:Envelope>

HTTP Response

HTTP/1.1 200 OK  
Content-Type: text/xml; charset=utf-8  
Content-Length: 495  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
 <soap:Body>  
 <isbm:ReadResponseResponse xmlns:isbm="http://www.openoandm.org/ws-isbm/">  
 <isbm:ResponseMessage>  
 <isbm:MessageID>af250a33-d5af-4c25-bb57-56802d8fea79</isbm:MessageID>  
 <isbm:MessageContent>  
 <Text>Pong!</Text>  
 </isbm:MessageContent>  
 </isbm:ResponseMessage>  
 </isbm:ReadResponseResponse>  
 </soap:Body>  
</soap:Envelope>

* + 1. RemoveResponse

The Consumer Application removes the response message from the ISBM Service Provider.

HTTP Request

POST /ConsumerPublicationService HTTP/1.1  
Host: ws-isbm.example.com  
Content-Type: text/xml; charset=utf-8  
Content-Length: 760  
SOAPAction: "http://www.openoandm.org/ws-isbm/RemoveResponse"  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
 <soap:Header>  
 <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">  
 <wsse:UsernameToken>  
 <wsse:Username>ConsumerApplication</wsse:Username>  
 <wsse:Password>^Um.7oFM9jrnnC</wsse:Password>  
 </wsse:UsernameToken>  
 </wsse:Security>  
 </soap:Header>  
 <soap:Body>  
 <isbm:RemoveResponse xmlns:isbm="http://www.openoandm.org/ws-isbm/">  
 <isbm:SessionID>a9b5c71a-d1b5-4fc8-81d1-ba1fee3af0df</isbm:SessionID>  
 <isbm:RequestMessageID>e8cfecb1-d2fc-4167-88f7-c90d60fc53ee</isbm:RequestMessageID>  
 </isbm:RemoveResponse>  
 </soap:Body>  
</soap:Envelope>

HTTP Response

HTTP/1.1 200 OK  
Content-Type: text/xml; charset=utf-8  
Content-Length: 239  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
 <soap:Body>  
 <isbm:RemoveResponseResponse xmlns:isbm="http://www.openoandm.org/ws-isbm/"/>  
 </soap:Body>  
</soap:Envelope>

* + 1. CloseConsumerRequestSession

The Consumer Application closes the consumer request session with the ISBM Service Provider.

HTTP Request

POST /ConsumerRequestService HTTP/1.1  
Host: ws-isbm.example.com  
Content-Type: text/xml; charset=utf-8  
Content-Length: 696  
SOAPAction: "http://www.openoandm.org/ws-isbm/CloseConsumerRequestSession"  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
 <soap:Header>  
 <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">  
 <wsse:UsernameToken>  
 <wsse:Username>ConsumerApplication</wsse:Username>  
 <wsse:Password>^Um.7oFM9jrnnC</wsse:Password>  
 </wsse:UsernameToken>  
 </wsse:Security>  
 </soap:Header>  
 <soap:Body>  
 <isbm:CloseConsumerRequestSession xmlns:isbm="http://www.openoandm.org/ws-isbm/">  
 <isbm:SessionID>a9b5c71a-d1b5-4fc8-81d1-ba1fee3af0df</isbm:SessionID>  
 </isbm:CloseConsumerRequestSession>  
 </soap:Body>  
</soap:Envelope>

HTTP Response

HTTP/1.1 200 OK  
Content-Type: text/xml; charset=utf-8  
Content-Length: 252  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
 <soap:Body>  
 <isbm:CloseConsumerRequestSessionResponse xmlns:isbm="http://www.openoandm.org/ws-isbm/"/>  
 </soap:Body>  
</soap:Envelope>

* + 1. CloseProviderRequestSession

The Provider Application closes the provider request session with the ISBM Service Provider.

HTTP Request

POST /ProviderRequestService HTTP/1.1  
Host: ws-isbm.example.com  
Content-Type: text/xml; charset=utf-8  
Content-Length: 696  
SOAPAction: "http://www.openoandm.org/ws-isbm/CloseProviderRequestSession"  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
 <soap:Header>  
 <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">  
 <wsse:UsernameToken>  
 <wsse:Username>ProviderApplication</wsse:Username>  
 <wsse:Password>9gy#gXENxph8?W</wsse:Password>  
 </wsse:UsernameToken>  
 </wsse:Security>  
 </soap:Header>  
 <soap:Body>  
 <isbm:CloseProviderRequestSession xmlns:isbm="http://www.openoandm.org/ws-isbm/">  
 <isbm:SessionID>16ee00f6-8fa9-4e80-8c36-a9d6d2bdb551</isbm:SessionID>  
 </isbm:CloseProviderRequestSession>  
 </soap:Body>  
</soap:Envelope>

HTTP Response

HTTP/1.1 200 OK  
Content-Type: text/xml; charset=utf-8  
Content-Length: 252  
  
<?xml version="1.0" encoding="UTF-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
 <soap:Body>  
 <isbm:CloseProviderRequestSessionResponse xmlns:isbm="http://www.openoandm.org/ws-isbm/"/>  
 </soap:Body>  
</soap:Envelope>

Acknowledgements

The following individuals have participated in the creation of this specification and are gratefully acknowledged:

* Georg Grossmann, University of South Australia
* James Fort, Microsoft
* .
* .
* .
* .

Bibliography

1. [XML Path Language (XPath) Version 1.0](http://www.w3.org/TR/xpath/), James Clark and Steven DeRose, Editors. World Wide Web Consortium, 16 Nov 1999. This version is http://www.w3.org/TR/1999/REC-xpath-19991116. The [latest version](http://www.w3.org/TR/xpath/) is available at [*http://www.w3.org/TR/xpath*](http://www.w3.org/TR/xpath).