ws-ISBM 1.0

Web Service Information Service Bus Model

OpenO&M Standard

2014-09-08

WSDLs

http://www.openoandm.org/isbm/1.0/wsdI/ChannelManagementService.wsdl
http://www.openoandm.org/isbm/1.0/wsdI/NotificationService.wsdl
http://www.openoandm.org/isbm/1.0/wsdI/ProviderPublicationService.wsdl
http://www.openoandm.org/isbm/1.0/wsdI/ConsumerPublicationService.wsdl
http://www.openoandm.org/isbm/1.0/wsdI/ProviderRequestService.wsdl
http://www.openoandm.org/isbm/1.0/wsdI/ConsumerRequestService.wsdl

Packaged Specification

http://www.openoandm.org/isbm/ws-isbm-1.0.zip

Latest Version
http://www.openoandm.org/isbm
Editors

MIMOSA

Avin Mathew, Assetricity
Ken Bever, Assetricity

ISA

Dennis Brandl, BR&L Consulting

Abstract

This document defines a SOAP Web Service implementation of the ISA 95.00.06 Messaging Service Model. The ws-ISBM defines a minimal interface subset to Enterprise Service Buses (ESB) and other message exchange middleware, using a standard interface consisting of channels and topics. The benefit from this approach is to allow applications to expose a single, standardized interface rather than having to be custom built for every version and format of ESB or message exchange system.

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.

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. You can alternatively contact MIMOSA directly and arrangements will be made to transpose appropriate remarks to the public bug database. All feedback is welcome.

Notices

Copyright MIMOSA 2014. 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.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to MIMOSA, except as needed for the purpose of developing any document or deliverable produced by a MIMOSA Technical Committee (in which case the rules applicable to copyrights, as set forth in the MIMOSA IPR Policy, must be followed) or as required to translate it into languages other than English.
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 ws-ISBM is released under the MIMOSA License Agreement.
Contents

Abstract ........................................................................................................................................... 2
Status ............................................................................................................................................... 2
Notices ............................................................................................................................................ 2
1 Introduction ................................................................................................................................... 7
   1.2 Notational Conventions ........................................................................................................ 7
   1.3 Namespaces .......................................................................................................................... 7
2 Service Requirements ................................................................................................................... 7
   2.1 Message Content Format ....................................................................................................... 7
   2.2 Security .................................................................................................................................. 7
   2.3 Error Handling ....................................................................................................................... 8
      2.3.1 SOAP Faults .................................................................................................................. 8
      2.3.2 Parameter Faults ........................................................................................................... 8
      2.3.3 Invalid Notification URL ............................................................................................ 8
   2.4 Content-Based Filtering ....................................................................................................... 8
   2.5 Message Expiry ...................................................................................................................... 9
3 Service Definitions ....................................................................................................................... 9
   3.1 Terminology .......................................................................................................................... 9
   3.2 Channel Management Service ............................................................................................... 10
      3.2.1 Create Channel ............................................................................................................. 10
      3.2.2 Add Security Tokens .................................................................................................... 10
      3.2.3 Remove Security Tokens ............................................................................................. 11
      3.2.4 Delete Channel ............................................................................................................ 11
      3.2.5 Get Channel ................................................................................................................ 11
      3.2.6 Get Channels .............................................................................................................. 12
   3.3 Notification Service ................................................................................................................ 12
      3.3.1 Notify Listener .............................................................................................................. 12
   3.4 Provider Publication Service .................................................................................................. 12
      3.4.1 Open Publication Session ............................................................................................. 12
      3.4.2 Post Publication ............................................................................................................ 13
      3.4.3 Expire Publication ......................................................................................................... 13
      3.4.4 Close Publication Session ........................................................................................... 13
   3.5 Consumer Publication Service .............................................................................................. 14
      3.5.1 Open Subscription Session ......................................................................................... 14

© 1998 - 2014 MIMOSA. All rights reserved.
1 Introduction

This ws-ISBM specification defines a SOAP Web Service implementation of the ISA 95.00.06 Messaging Service Model.

The ws-ISBM defines a minimal interface subset to Enterprise Service Buses (ESB) and other message exchange middleware using a standard Web Service interface. 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 filtering for granular content-based filtering. An asynchronous Web Service callback is also provided to clients for notification of applicable messages. Token-based security for channels is specified to support multiple authorization models, from basic credential exchange to federated identity providers.

The benefit of a ws-ISBM implementation is to allow applications to expose a single, standardized interface rather than having to be custom built for every version and format of ESB or message exchange system. The goal is to further interoperability in application to application communications.

1.2 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.

This specification uses the following syntax to define XML structures: Element Name (Type) [Cardinality]. The namespaces for Types are defined in the following section. For example, the Topic element defined as an XML Schema string with one to many cardinality would be defined as: Topic (xs:string) [1..*].

1.3 Namespaces

The following namespaces are used in this document:

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Namespace</th>
</tr>
</thead>
<tbody>
<tr>
<td>xs</td>
<td><a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a></td>
</tr>
<tr>
<td>isbm</td>
<td><a href="http://www.openoandm.org/isbm/">http://www.openoandm.org/isbm/</a></td>
</tr>
</tbody>
</table>

2 Service Requirements

The following items define shared requirements that are applicable across the various services defined in Service Definitions. These requirements supplement the service requirements specified by ISA 95.00.06 but are contextualized for SOAP Web Services.

2.1 Message Content Format

Content is associated with a message through the use of an XML Schema any element. The XML content MUST be valid XML. A ws-ISBM Service Provider SHOULD preserve significant whitespace and comments within the XML content.

Note: As only an XML element is associated with a message, an XML declaration specifically for the content is not supported. Instead, the message content will inherit the XML declaration of the SOAP body.

2.2 Security

Security in the ws-ISBM specification only provide authorization of channels. Authorization of services is considered out-of-scope.
All ws-ISBM implementations MUST support transport layer security (e.g. SSL/TLS) in order to secure tokens and messages, and to prevent replay attacks.

All ws-ISBM implementations MUST support the WS-Security UsernameToken using PasswordText as a basic level of security token. Examples of its use can be found in Example HTTP Flows.

A ws-ISBM Service Provider MAY choose to support additional forms of security tokens (e.g. SAML assertions, OAuth tokens) and it is RECOMMENDED that a ws-ISBM Service Provider support out-of-band token exchange standards such as SAML, WS-Federation or OAuth.

As security tokens in the 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 ws-ISBM Service Provider MUST define the supported formats.

A ws-ISBM Service Provider MUST validate security tokens for every service operation except for the Channel Management Service CreateChannel operation (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).

2.3 Error Handling

2.3.1 SOAP Faults

SOAP 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). For a SOAP 1.2 implementation this is provided through the SOAP Reason element (see SOAP 1.2, SOAP Reason Element).

Note The declared SOAP Faults specified by the services and WSDLs do not have any elements or attributes defined. 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.

2.3.2 Parameter Faults

If any parameter for an operation is malformed or not optional and blank, then a ws-ISBM Service Provider MUST return a SOAP Fault and SHOULD include undeclared isbm:ParameterFault element in the fault details to aid senders in determining the type of error. The fault MUST carry the offending parameter name/s.

2.3.3 Invalid Notification URL

If a provider/consumer application provides an invalid URL or a URL that does not host a NotifyListener service, the ws-ISBM Service Provider MAY choose not to send or defer a NotifyListener SOAP request.

2.4 Content-Based Filtering

To allow efficient content-based filtering of messages, an XPath expression MAY be added to a subscription or read request session to provide a filtering definition. The XPath expression MUST be defined as an XPath v1.0 expression. An XPath evaluation that returns an empty value or node set MUST NOT cause a notification to be generated nor will the message be visible to the receiving system. For an XPath expression that use namespaces, multiple namespace prefixes and names are added upon session creation.

Note An empty result from an XPath evaluation will result in the whole message being filtered; the message content itself is not filtered.
2.5 Message Expiry

During posting of certain messages, a sender MAY specify an expiry duration for the message. A ws-ISBM Service Provider MUST hide an expired message from potential receivers unless the receiver has already read the message, in which case it will always remain visible to that particular receiver. This is to ensure the message is still available to the receiver to ensure message removal removes the correct message.

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

Note Responses can still be posted for an expired request message, and Consumers will still receive response notifications and be able to read and remove these responses.

3 Service Definitions

All services defined in ISA 95.00.06 are defined as SOAP Web Services in this specification. The SOAP 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 a 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.

3.1 Terminology

ChannelDescription
The description of a 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.

Expiry
The duration until the expiration of the message.

ListenerURL
The URL endpoint, reachable by the ws-ISBM Service Provider, which hosts a ws-ISBM Notification Service. Used to indicate when a new message is available for a session.

MessageContent
The XML content of a message.

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

NamespaceName
The namespace name used for an XPath filter expression.

NamespacePrefix
The namespace prefix used for an XPath filter expression.

SecurityToken
A token that can be assigned to a channel to control authorization.
**SessionID**
An identifier generated by the ws-ISBM Service Provider upon creation of a session. Identifiers SHOULD be made non-obvious and not easily guessable.

**XPathExpression**
The XPath 1.0 expression that is used to filter message content.

### 3.2 Channel Management Service

The Channel Management Service is available as a WSDL description.

#### 3.2.1 Create Channel

<table>
<thead>
<tr>
<th>Name</th>
<th>CreateChannel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Creates a new channel.</td>
</tr>
</tbody>
</table>

**Input**

- ChannelURI (xs:string) [1]
- ChannelType (isbm:ChannelType) [1]
- ChannelDescription (xs:string) [0..1]
- SecurityToken (isbm: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

#### 3.2.2 Add Security Tokens

<table>
<thead>
<tr>
<th>Name</th>
<th>AddSecurityTokens</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Adds security tokens to a channel.</td>
</tr>
</tbody>
</table>

**Input**

- ChannelURI (xs:string) [1]
- SecurityToken (isbm:SecurityToken) [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 a specified SecurityToken is already assigned to the channel, then no further action is taken to maintain a distinct list.

**Output**

N/A

**Faults**

ChannelFault
### 3.2.3 Remove Security Tokens

<table>
<thead>
<tr>
<th>Name</th>
<th>RemoveSecurityTokens</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Removes security tokens from a channel.</td>
</tr>
<tr>
<td>Input</td>
<td>ChannelURI (xs:string) [1] SecurityToken (isbm:SecurityToken) [1..*]</td>
</tr>
<tr>
<td>Behavior</td>
<td>If the ChannelURI does not exist, then a ChannelFault is returned.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>If any specified SecurityToken is not assigned to the channel, then an SecurityTokenFault is returned. No tokens are removed from the channel, even if they are valid.</td>
</tr>
<tr>
<td>Output</td>
<td>N/A</td>
</tr>
<tr>
<td>Faults</td>
<td>ChannelFault</td>
</tr>
<tr>
<td></td>
<td>SecurityTokenFault</td>
</tr>
</tbody>
</table>

### 3.2.4 Delete Channel

<table>
<thead>
<tr>
<th>Name</th>
<th>DeleteChannel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Deletes a channel.</td>
</tr>
<tr>
<td>Input</td>
<td>ChannelURI (xs:string) [1]</td>
</tr>
<tr>
<td>Behavior</td>
<td>If the ChannelURI does not exist, then a ChannelFault is returned.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>The channel along with associated sessions and messages are deleted. No notification is provided to any applications with active sessions.</td>
</tr>
<tr>
<td>Output</td>
<td>N/A</td>
</tr>
<tr>
<td>Faults</td>
<td>ChannelFault</td>
</tr>
</tbody>
</table>

### 3.2.5 Get Channel

<table>
<thead>
<tr>
<th>Name</th>
<th>GetChannel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Gets information about a channel.</td>
</tr>
<tr>
<td>Input</td>
<td>ChannelURI (xs:string) [1]</td>
</tr>
<tr>
<td>Behavior</td>
<td>If the ChannelURI does not exist, then a ChannelFault is returned.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>Output</td>
<td>Channel (isbm:Channel) [1], composed of:</td>
</tr>
<tr>
<td></td>
<td>ChannelURI (xs:string) [1]</td>
</tr>
<tr>
<td></td>
<td>ChannelType (isbm:ChannelType) [1]</td>
</tr>
<tr>
<td></td>
<td>ChannelDescription (xs:string) [0..1]</td>
</tr>
</tbody>
</table>
### 3.2.6 Get Channels

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Input</th>
<th>Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetChannels</td>
<td>Gets information about all channels.</td>
<td>N/A</td>
<td>The channels returned are filtered by those that match the security token. Any channel that does not have security tokens assigned are returned regardless.</td>
</tr>
</tbody>
</table>

**Output**

Channel (isbm:Channel) [0..*], composed of:

- ChannelURI (xs:string) [1]
- ChannelType (isbm:ChannelType) [1]
- ChannelDescription (xs:string) [0..1]

### Faults

N/A

---

### 3.3 Notification Service

The Notification Service is available as a WSDL description.

#### 3.3.1 Notify Listener

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotifyListener</td>
<td>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.</td>
</tr>
</tbody>
</table>

**Input**

- SessionID (xs:string) [1]
- MessageID (xs:string) [1]
- Topic (xs:string) [0..*]
- RequestMessageID (xs:string) [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

---

### 3.4 Provider Publication Service

The Provider Publication Service is available as a WSDL description.

#### 3.4.1 Open Publication Session

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Input</th>
<th>Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenPublicationSession</td>
<td>Opens a publication session for a channel.</td>
<td>ChannelURI (xs:string) [1]</td>
<td>If the ChannelURI does not exist, then a ChannelFault is returned.</td>
</tr>
</tbody>
</table>
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 (xs:string) [1]

Faults
ChannelFault
OperationFault

### 3.4.2 Post Publication

<table>
<thead>
<tr>
<th>Name</th>
<th>PostPublication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Posts a publication message on a channel.</td>
</tr>
<tr>
<td>Input</td>
<td>SessionID (xs:string) [1]</td>
</tr>
<tr>
<td></td>
<td>MessageContent (isbn:MessageContent) [1]</td>
</tr>
<tr>
<td></td>
<td>Topic (xs:string) [1..*]</td>
</tr>
<tr>
<td></td>
<td>Expiry (xs:duration) [0..1]</td>
</tr>
<tr>
<td>Behavior</td>
<td>If the SessionID does not exist or does not correspond to a publication session, then a SessionFault is returned.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>Output</td>
<td>MessageID (xs:string) [1]</td>
</tr>
<tr>
<td>Faults</td>
<td>SessionFault</td>
</tr>
</tbody>
</table>

### 3.4.3 Expire Publication

<table>
<thead>
<tr>
<th>Name</th>
<th>ExpirePublication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Expires a posted publication.</td>
</tr>
<tr>
<td>Input</td>
<td>SessionID (xs:string) [1]</td>
</tr>
<tr>
<td></td>
<td>MessageID (xs:string) [1]</td>
</tr>
<tr>
<td>Behavior</td>
<td>If the SessionID does not exist or does not correspond to a publication session, then a SessionFault is returned.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>If the MessageID does not correspond with the SessionID or the corresponding message has already expired, then no further action is taken.</td>
</tr>
<tr>
<td></td>
<td>The message is expired for all topics associated with the message.</td>
</tr>
<tr>
<td>Output</td>
<td>N/A</td>
</tr>
<tr>
<td>Faults</td>
<td>SessionFault</td>
</tr>
</tbody>
</table>

### 3.4.4 Close Publication Session

<table>
<thead>
<tr>
<th>Name</th>
<th>ClosePublicationSession</th>
</tr>
</thead>
</table>

© 1998 - 2014 MIMOSA. All rights reserved.
### Description
Closes a publication session.

### Input
- **SessionID** (xs:string) [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

### 3.5 Consumer Publication Service

The Consumer Publication Service is available as a WSDL description.

#### 3.5.1 Open Subscription Session

<table>
<thead>
<tr>
<th>Name</th>
<th>OpenSubscriptionSession</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>Opens a subscription session for a channel.</td>
</tr>
<tr>
<td><strong>Input</strong></td>
<td>ChannelURI (xs:string) [1]</td>
</tr>
<tr>
<td></td>
<td>Topic (xs:string) [1..*]</td>
</tr>
<tr>
<td></td>
<td>ListenerURL (xs:string) [0..1]</td>
</tr>
<tr>
<td></td>
<td>XPathExpression (xs:string) [0..1]</td>
</tr>
<tr>
<td></td>
<td>XPathNamespace (isbm:Namespace) [0..*], composed of:</td>
</tr>
<tr>
<td></td>
<td>NamespacePrefix (xs:string) [1]</td>
</tr>
<tr>
<td></td>
<td>NamespaceName (xs:string) [1]</td>
</tr>
</tbody>
</table>

### 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** (xs:string) [1]

### Faults
- ChannelFault
- NamespaceFault
- OperationFault

#### 3.5.2 Read Publication

<table>
<thead>
<tr>
<th>Name</th>
<th>ReadPublication</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>Returns the first non-expired publication message or a previously read expired message that</td>
</tr>
</tbody>
</table>
satisfies the session message filters.

<table>
<thead>
<tr>
<th>Input</th>
<th>SessionID (xs:string) [1]</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Behavior</th>
<th>If the SessionID does not exist or does not correspond to a subscription session, then a SessionFault is returned.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>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.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Output</th>
<th>PublicationMessage (isbm:PublicationMessage) [0..1], composed of:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MessageID (xs:string) [1]</td>
</tr>
<tr>
<td></td>
<td>MessageContent (isbm:MessageContent) [1]</td>
</tr>
<tr>
<td></td>
<td>Topic (xs:string) [1..*]</td>
</tr>
</tbody>
</table>

Fausts | SessionFault

### 3.5.3 Remove Publication

<table>
<thead>
<tr>
<th>Name</th>
<th>RemovePublication</th>
</tr>
</thead>
</table>

| Description | Removes the first, if any, publication message in the subscription queue. |

<table>
<thead>
<tr>
<th>Input</th>
<th>SessionID (xs:string) [1]</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Behavior</th>
<th>If the SessionID does not exist or does not correspond to a subscription session, then a SessionFault is returned.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>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.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Output</th>
<th>N/A</th>
</tr>
</thead>
</table>

| Faults  | SessionFault |

### 3.5.4 Close Subscription Session

<table>
<thead>
<tr>
<th>Name</th>
<th>CloseSubscriptionSession</th>
</tr>
</thead>
</table>

| Description | Closes a subscription session. |

<table>
<thead>
<tr>
<th>Input</th>
<th>SessionID (xs:string) [1]</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Behavior</th>
<th>If the SessionID does not exist (non-existent or already closed) or does not correspond to a publication session, then a SessionFault is returned.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>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.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Output</th>
<th>N/A</th>
</tr>
</thead>
</table>

| Faults  | SessionFault |

### 3.6 Provider Request Service

The Provider Request Service is available as a WSDL description.

#### 3.6.1 Open Provider Request Session

<table>
<thead>
<tr>
<th>Name</th>
<th>OpenProviderRequestSession</th>
</tr>
</thead>
</table>

| Description | Opens a provider request session for a channel for reading requests and posting responses. |

© 1998 - 2014 MIMOSA. All rights reserved.
Input

ChannelURI (xs:string) [1]
Topic (xs:string) [1..*]
ListenerURL (xs:string) [0..1]
XPathExpression (xs:string) [0..1]
XPathNamespace (ism:Namespace) [0..*], composed of:
  NamespacePrefix (xs:string) [1]
  NamespaceName (xs:string) [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 (xs:string) [1]

Faults

ChannelFault
NamespaceFault
OperationFault

3.6.2 Read Request

Name ReadRequest
Description Returns the first non-expired request message or a previously read expired message that satisfies the session message filters.

Input SessionID (xs: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.
The returned Topic will correspond to the first topic that matched the posted request.

Output RequestMessage (ism:RequestMessage) [0..1], composed of:
  MessageID (xs:string) [1]
  MessageContent (ism:MessageContent) [1]
  Topic (xs:string) [1]

Faults SessionFault

3.6.3 Remove Request

Name RemoveRequest
3.6.4 Post Response

Name: PostResponse
Description: Posts a response message on a channel.
Input:
- SessionID (xs:string) [1]
- RequestMessageID (xs:string) [1]
- MessageContent (isbm: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.
- If there is no request message that can be matched to RequestMessageID, then no further action is taken.
Output:
- MessageID (xs:string) [1]
Faults: SessionFault

3.6.5 Close Provider Request Session

Name: CloseProviderRequestSession
Description: Closes a provider request session.
Input:
- SessionID (xs:string) [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

3.7 Consumer Request Service

The Consumer Request Service is available as a WSDL description.

3.7.1 Open Consumer Request Session

Name: OpenConsumerRequestSession
<table>
<thead>
<tr>
<th>Description</th>
<th>Opens a consumer request session for a channel for posting requests and reading responses.</th>
</tr>
</thead>
</table>
| **Input**   | ChannelURI (xs:string) [1]   
|             | ListenerURL (xs:string) [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 (xs:string) [1]                                                       |
| **Faults**  | ChannelFault                                                                    |
|             | OperationFault                                                                  |

### 3.7.2 Post Request

<table>
<thead>
<tr>
<th>Name</th>
<th>PostRequest</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>Posts a request message on a channel.</td>
</tr>
</tbody>
</table>
| **Input**    | SessionID (xs:string) [1]   
|             | MessageContent (isbm:MessageContent) [1]                                           |
|             | Topic (xs:string) [1]                                                         |
|             | Expiry (xs:duration) [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 (xs:string) [1]                                                       |
| **Faults**  | SessionFault                                                                    |

### 3.7.3 Expire Request

<table>
<thead>
<tr>
<th>Name</th>
<th>ExpireRequest</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>Expires a posted request message.</td>
</tr>
</tbody>
</table>
| **Input**    | SessionID (xs:string) [1]   
|             | MessageID (xs:string) [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. |
| **Output**  | N/A                                                        |
### 3.7.4 Read Response

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReadResponse</td>
<td>Returns the first response message, if any, in the session message queue associated with the request.</td>
</tr>
</tbody>
</table>

**Input**
- SessionID (xs:string) [1]
- RequestMessageID (xs:string) [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 (isbm:ResponseMessage) [0..1], composed of:
  - MessageID (xs:string) [1]
  - MessageContent (isbm:MessageContent) [1]

**Faults**
- SessionFault

### 3.7.5 Remove Response

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RemoveResponse</td>
<td>Deletes the first response message, if any, in the session message queue associated with the request.</td>
</tr>
</tbody>
</table>

**Input**
- SessionID (xs:string) [1]
- RequestMessageID (xs:string) [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

### 3.7.6 Close Consumer Request Session

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CloseConsumerRequestSession</td>
<td>Closes a consumer request session.</td>
</tr>
</tbody>
</table>

**Input**
- SessionID (xs:string) [1]

**Behavior**
- If the SessionID does not exist (non-existent or already closed) or does not correspond to a Request
A 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.

<table>
<thead>
<tr>
<th>Output</th>
<th>N/A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Faults</td>
<td>SessionFault</td>
</tr>
</tbody>
</table>

## 4 XML Data Structures

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

### Channel

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

### ChannelType

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

### MessageContent

```xml
<xsd:complexType name="MessageContent">
  <xsd:sequence>
    <xsd:any minOccurs="1" maxOccurs="1" namespace="#any" processContents="lax"/>
  </xsd:sequence>
</xsd:complexType>
```

### Namespace

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

### PublicationMessage

```xml
<xsd:complexType name="PublicationMessage">
  <xsd:sequence>
    <xsd:element minOccurs="1" maxOccurs="1" name="MessageID" type="xs:string"/>
  </xsd:sequence>
</xsd:complexType>
```
5 Conformance

Any assessment of conformance of a ws-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 Filter Expressions in an XPath 1.0 format
10. Support for other Security Tokens formats
11. 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
Appendix A Example HTTP Flows

A.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/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/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/isbm/">
    </isbm:CreateChannelResponse>
  </soap:Body>
</soap:Envelope>

2. 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/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>
  </soap:Body>
</soap:Envelope>
3. 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/isbm/RemoveSecurityToken"

<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <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/isbm/"
                             xmlns:isbm:ChannelURI='/Enterprise/Site/Area/WorkCenter'/
                             xmlns:isbm:SecurityToken>
      <wsse:UsernameToken>
        <wsse:Username>Application1</wsse:Username>
        <wsse:Password>&lt;s9.vQfLDx9gL</wsse:Password>
      </wsse:UsernameToken>
    </isbm:SecurityToken>
  </soap:Body>
</soap:Envelope>
4. 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/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/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>
      <!-- Fault details go here -->
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
5. 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/isbm/GetChannels"

```xml
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <isbm:GetChannels xmlns:isbm="http://www.openoandm.org/isbm/"/>
  </soap:Body>
</soap:Envelope>
```

**HTTP Response**

HTTP/1.1 200 OK  
Content-Type: text/xml; charset=utf-8  
Content-Length: 442

```xml
<?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/isbm/"/>
  </soap:Body>
</soap:Envelope>
```
6. 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/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?rFum48mg</wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </soap:Header>
    <soap:Body>
        <isbm:DeleteChannel xmlns:isbm="http://www.openoandm.org/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/isbm/>
    </soap:Body>
</soap:Envelope>
A.2 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/isbm/OpenSubscriptionSession"

```xml
<?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 xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
```

© 1998 - 2014 MIMOSA. All rights reserved.
2. OpenSubscriptionSession

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/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>ProviderApplication</wsse:Username>
                <wsse:Password>qEJaz4F?U4rW;q</wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </soap:Header>
    <soap:Body>
        <isbm:OpenSubscriptionSession xmlns:isbm="http://www.openoandm.org/isbm/">
            <isbm:ChannelURI>/Enterprise/Site/Area/WorkCenter</isbm:ChannelURI>
            <isbm:Topic>Text</isbm:Topic>
            <isbm:ListenerURL>http://consumer.example.com/NotificationService</isbm:ListenerURL>
        </isbm:OpenSubscriptionSession>
    </soap:Body>
</soap:Envelope>
3. 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/isbm/PostPublication"

<?xml version="1.0" encoding="UTF-8"?>
  <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/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
4. 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/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/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>
</soap:Body>
</soap:Envelope>

5. 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/isbm/ReadPublication"
6. 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/isbm/ExpirePublication"

<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
<soap:Body
<isbm:DeletePublication xmlns:isbm="http://www.openoandm.org/isbm/"
<isbm:SessionID>e94c645a-6450-411e-8ec7-4b70620d3a98</isbm:SessionID>
</isbm:DeletePublication>
</soap:Body>
</soap:Envelope>
7. 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/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/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/isbm/"/>
  </soap:Body>
</soap:Envelope>

8. 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/isbm/ClosePublicationSession"

<?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"
  xmlns:isbm="http://www.openoandm.org/isbm/"
  xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-2.0.xsd">
  <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/isbm/"/>
    <isbm:SessionID>ac0ee730-ca88-421a-b348-ce01babdb1c</isbm:SessionID>
  </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/"
  xmlns:isbm="http://www.openoandm.org/isbm/"
  xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-2.0.xsd">
  <soap:Body>
    <isbm:ClosePublicationSessionResponse xmlns:isbm="http://www.openoandm.org/isbm/"/>
  </soap:Body>
</soap:Envelope>
9. 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/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/isbm/"
      isbm:SessionID="e94c645a-6450-411e-8ec7-4b7062d3a98"/>
  </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/isbm/"/>
  </soap:Body>
</soap:Envelope>
A.3 Request-Response Example

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

© 1998 - 2014 MIMOSA. All rights reserved.
**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/isbm/OpenProviderRequestSession"

```xml
<?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/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
<?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/isbm/">
      <isbm:SessionID>16ee00f6-8fa9-4e80-8c36-a9d6d2b551</isbm:SessionID>
    </isbm:OpenProviderRequestSessionResponse>
  </soap:Body>
</soap:Envelope>
```

### 2. 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/isbm/OpenConsumerRequestSession"

```xml
<?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/isbm/">
      <isbm:SessionID>16ee00f6-8fa9-4e80-8c36-a9d6d2b551</isbm:SessionID>
    </isbm:OpenProviderRequestSessionResponse>
  </soap:Body>
</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.7oFM9jnnC</wsse:Password>
        </wsse:UsernameToken>
    </wsse:Security>
</soap:Header>

<soap:Body>
    <isbm:OpenConsumerRequestSession xmlns:isbm="http://www.openoandm.org/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/isbm/">
            <isbm:SessionID>a9b5c71a-d1b5-4fc8-81d1-ba1fee3af0df</isbm:SessionID>
        </isbm:OpenConsumerRequestSessionResponse>
    </soap:Body>
</soap:Envelope>

3. 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/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.7oFM9jnnC</wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </soap:Header>
    <soap:Body>
<isbm:PostRequest xmlns:isbm="http://www.openoandm.org/isbm/"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <isbm:SessionID>a9b5c71a-d1b5-4fc8-81d1-ba1fee3af0df</isbm:SessionID>
    <isbm:MessageContent>
        <Text>Ping!</Text>
    </isbm:MessageContent>
</isbm:PostRequest>

4. 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/isbm/NotifyListener"

<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:isbm="http://www.openoandm.org/isbm/">
    <soap:Body>
        <isbm:NotifyListener xmlns:isbm="http://www.openoandm.org/isbm/"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
            <isbm:SessionID>16ee00f6-8fa9-4e80-8c36-a9d6d2db551</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/"
    xmlns:isbm="http://www.openoandm.org/isbm/"/>
5. **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/isbm/ReadRequest"

```
<?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"  
  xmlns:isbm="http://www.openoandm.org/isbm/"  
  <wsse:UsernameToken>
    <wsse:Username>ProviderApplication</wsse:Username>
    <wsse:Password>9gy#gXENxph8?W</wsse:Password>
  </wsse:UsernameToken>
  <soap:Body>
    <isbm:ReadRequest xmlns:isbm="http://www.openoandm.org/isbm/"
      xmlns:isbm:SessionID>16ee00f6-8fa9-4e80-8c36-a9d6d2dbb551</isbm:SessionID>
  </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/"
  xmlns:isbm="http://www.openoandm.org/isbm/"  
  xmlns:isbm:ReadRequestResponse>
  <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>
</soap:Body>
</soap:Envelope>
```

6. **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/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/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/isbm/"></isbm:RemoveRequestResponse>
  </soap:Body>
</soap:Envelope>

7. 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/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>
  </soap:Body>
</soap:Envelope>
<wsse:Password>9gy#gXENxph8?W</wsse:Password>
</wsse:UsernameToken>
</soap:Security>
</soap:Header>

<soap:Body>
<isbm:PostResponse xmlns:isbm="http://www.openoandm.org/isbm/">
<isbm:SessionID>16ee00f6-8fa9-4e80-8c36-a9d62b4551</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>
    </soap:Body>
  </soap:Envelope>

8. 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/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/isbm/">
      <isbm:SessionID>a9b5c71a-d1b5-4fca-81d1-ba1fde3af0df</isbm:SessionID>
      <isbm:MessageID>af250a33-d5af-4c25-bb57-56802d8fe479</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/isbm/">
    </soap:Body>
  </soap:Envelope>

9. 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/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/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/isbm/">
      <isbm:ResponseMessage>
        <isbm:MessageID>af250a33-d5af-4c25-bb57-56802d8f6a79</isbm:MessageID>
        <isbm:MessageContent>
          <Text>Pong!</Text>
        </isbm:MessageContent>
      </isbm:ResponseMessage>
    </isbm:ReadResponseResponse>
  </soap:Body>
</soap:Envelope>
10. 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/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/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/isbm/"/>
  </soap:Body>
</soap:Envelope>

11. 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/isbm/CloseConsumerRequestSession"

<?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/isbm/"/>
  </soap:Body>
</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/isbm/">
    <isbm:SessionID>a9b5c71a-d1b5-4fc8-81d1-ba1fee3af0df</isbm:SessionID>
  </isbm:CloseConsumerRequestSession>
</soap:Body>

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/isbm/">
    </isbm:CloseConsumerRequestSessionResponse>
  </soap:Body>
</soap:Envelope>

12. 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/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/isbm/">
      <isbm:SessionID>16ee00f6-8fa9-4e80-8c36-a9d6d2b5b51</isbm:SessionID>
    </isbm:CloseProviderRequestSession>
  </soap:Body>
</soap:Envelope>
<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/isbm/"/>
  </soap:Body>
</soap:Envelope>

**Acknowledgements**

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

- Adrian Park, Intergraph
- Brandon Mathis, Assetricity
- Chris Madison, IBM
- Chuck Pharris, SAP
- Frode Myren, IBM
- Georg Grossmann, University of South Australia
- Gopal Gopalkrishnan, OSIsoft
- James Fort, Microsoft
- Juan Ordoñez, Universidad del Cauca
- Keith Wells, IBM
- Lakshmi Narasimhan, Western Kentucky University
- Mahesh Punyamurthula, Microsoft
- Manesh Raveendran, Terawe
- Paul Nguyen, Microsoft
- Randy Armstrong, OPC Foundation
- Rick Solis, ExxonMobil
- Robert Chumbley, IBM
- Rowinson Medina, Universidad del Cauca
- Sebastian Steinhauer, SAP