WS-I Usage Scenarios for the WS-I Attachments Profile 1.0

Document Status: Member Review Draft

Version: 1.02

Date: 2 September, 2004

Editors: Marc Goodner, SAP
Barbara McKee, IBM

Table of Contents

Table of Figures

Revision History

November 24, 2003	Initial version
January 16, 2004	First review session
January 21, 2004	Results of Jan. 16th call
January 22, 2004	First pass mapping
January 27, 2004	AI from 1/23 call incorporated
January 30, 2004	Updated from call discussion
February 18, 2004	Changes accepted at F2F
February 25, 2004	Post F2F update
March 31, 2004	Editorial changes for Work Group Draft
April 20, 2004	Mappings to AP 4/19 ED
April 26, 2004	Updates from BP feedback
April 28, 2004	Editorial changes for Work Group Approval Draft
June 10, 2004	Editorial changes for Board Approval Draft
July 9, 2004	Updated per feedback during BdAD, changes tracked in Sample Apps WG Issue List
August 19, 2004	Updated with revisions proposed to Board and edits discussed in Sample Apps WG

Copyright

Confidentiality

This document contains proprietary information that is confidential and shall not be made available to unauthorized persons.

1 Introduction

1.1 WS-I Usage Scenarios for the Attachments Profile 1.0 define how Web services that have attachments can be used, identifying basic interoperability requirements for such interactions, and mapping the scenario to the requirements of the WS-I Attachments Profile 1.0 (hereafter, Attachments Profile) [3]Referenced attachments

. The scenarios in this document represent only the most common usage patterns of SOAP with Attachments. There are other uses of SOAP with Attachments not covered here, such as the use of unsolicited attachments. Scenarios are independent of any application domain. WS-I Use Cases employ Scenarios to model high-level definitions of specific applications.

The scenarios presented here can be composed or extended. That is, they describe Web service design patterns for attachments that can be combined and built upon like building blocks, and with the WS-I Usage Scenarios for the Basic Profile 1.0 [4]. For example, the Explicitly Bound Attachments scenario in this document describes an attachment pattern which can apply to the Synchronous Request/Response scenario or Basic Callback scenario from the Usage Scenarios for Basic Profile 1.0.

1.2 How to use this document

This document describes the most common usage patterns associated with the WS-I Attachments Profile. The Attachments Profile constraints and requirements are referenced directly and the reader is expected to use the Attachments Profile in conjunction with this document to interpret the referenced information.

The two scenarios presented in this document are intended to provide sufficient information so that a user of this document can create Web services that are conformant with the Attachment Profile using one or more of the scenarios. All applicable guidelines and restrictions for the messages and service description instances for each scenario are provided.

2 Usage Scenario Taxonomy

The Usage Scenario taxonomy defined for the WS-I Usage Scenarios for the Basic Profile 1.0 [4] is used here as well, for applying the Attachment Profile constraints. The taxonomy consists of a Web services stack and a set of activities, grouped by the layers of the stack that a Web service instance executes as part of the Web service Usage Scenario. Some of these layers and activities map to the Usage Scenarios for the Basic Profile and are not relevant to the Attachment scenarios per se. These are included because the Attachment scenarios augment the Basic Profile scenarios. The constraints of the Attachments Profile are applied to each activity as well as to the optional components of the scenario, e.g., the WSDL for the description of the Web service instance. There are two types of constraints on scenarios:

- Flow Constraints applying to each activity that takes part in the flow of the Web service. These include: expressing the Web service data model in XML, creating and consuming messages using SOAP, transporting messages using HTTP
- Description Constraints applying to the *description* of the Scenario. Operationally, the description of a Web service instance occurs in WSDL and possibly UDDI; therefore, these constraints are applied to the WSDL and UDDI describing the Scenario.

The following are attributes of WS-I Usage Scenarios for Attachments (note that there are other attributes of using SOAP with Attachments that are not reflected by the common patterns chosen for this document; unsolicited attachments, for example, are not described with a WSDL description):

- They include a flow description, linking together the set of activities specific to the scenario,

- They include optional components, such as SOAP headers or security,
- They are described with a WSDL document,
- Each activity within a scenario has constraints applied to it by the Attachments Profile, and
- They represent a real-world Web service implementation.

2.1 Web Service Stack

The Usage Scenario taxonomy is based on a Web services stack. Each layer of the stack represents one of the fundamental functional areas of a Web service instance. Not all possible functional areas are represented (e.g., security or coordination), only the most basic. This document will only address new additions to this stack to facilitate attachments in Web services. Please see the Usage Scenarios 1.0 [4] for information on underlying layers. These layers are depicted in the following diagram.

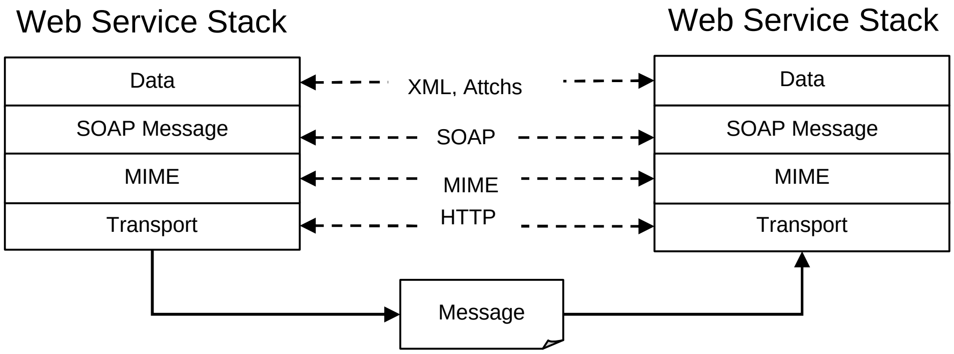


Figure 2-1 Web services stack

A Web service application may include several logical layers incorporating functions such as the Web service instance and application business logic. The Profiles and Usage Scenarios do not address application business logic except where the functionality of any part of the Web services stack is implemented within the business logic.

The details of each layer of the Web service stack are:

Data Layer

The data layer translates the application specific data into the model chosen for the specific Web service. The data layer includes the functions necessary to support flexible data typing. This layer maps to the `wsdl:types` and `wsdl:message` definitions within a WSDL document. It also handles attachment data as indicated in wsdl:binding definitions.

SOAP Message Layer

The SOAP message layer is the infrastructure that processes SOAP messages, dispatches them, and may optionally fulfill Quality of Service requirements. On the sending side the message layer writes SOAP messages, based on the data model defined in portTypes and bindings. On the receiving side the message layer processes the SOAP messages and dispatches requests to the correct application or method. When attachment parts are present the SOAP Message Layer may be called upon to create or resolve references to attachments.

MIME Layer

The MIME layer is the infrastructure that processes MIME headers, and MIME entity bodies for the root part and attachments. On the sending side the MIME Layer writes a MIME header for each part, including the SOAP Message (root) part, and the attachment parts themselves. In some cases the MIME Layer interacts with the SOAP Message Layer to insert references to attachment parts. On the receiving side the MIME Layer processes the MIME parts (MIME headers and MIME entity bodies), resolves references to attachment parts, and ensures that all MIME content is dispatched properly to the SOAP message layer. The MIME layer handles attachments that are expected through explicit declaration in a WSDL document, and those that are unsolicited.

Transport Layer

The transport layer sends and receives messages. For the Basic Profile, this includes only HTTP client and server platforms. This layer maps to the `wsdl:binding` and `wsdl:port` definitions within the WSDL document.

2.2 Activities

A set of activities is defined for each layer of the Web service stack. Activities are the fundamental operations that comprise a Web service. A single activity may have several constraints applied to it from the Basic Profile, and may also have constraints that apply from the Attachments Profile. For example, one activity might be “Send HTTP” and the specifications and guidelines for how to fulfill that activity come from the Use of SOAP in HTTP section of the Basic Profile and MIME section of the Attachments Profile.

The following table summarizes these activities.

Layer	**Activity**
Data Layer	Write XML Write attachment data Process XML Process attachment data
SOAP Message Layer	Write SOAP envelope Process SOAP envelope Write SOAP body Process SOAP body Write SOAP header Process SOAP header
MIME Layer	Write MIME header Process MIME header Write MIME entity body Process MIME entity body
Transport Layer	Send HTTP Receive HTTP

Table 1 - Activities grouped by Web services stack layer

2.2.1 Data Layer Activities

In addition to the activities related to handling XML data that were previously described in the Usage Scenarios for the Basic Profile 1.0 for activities, two additional activities have been added to the Data Layer for Usage Scenarios for the Attachments Profile. These are:

- Write attachment data
- Process attachment data.

2.2.2 SOAP Message Layer Activities

There are no additional activities for the SOAP Message Layer required for Usage Scenarios for the Attachments Profile. The Usage Scenarios for the Basic Profile 1.0 [4] contain descriptions for activities that are executed within the SOAP Message Layer. One caveat is that the SOAP Message Layer may be required to record and resolve references to attachments in the SOAP body, header, or fault activities.

2.2.3 MIME Layer Activities

The following activities occur within the MIME Layer to deal with MIME parts that are either described with a mime:multipartRelated WSDL binding, or to deal with unsolicited MIME parts:

MIME header

A MIME header precedes each MIME entity body in a multipart/related SOAP message.

- Write MIME header
- Process MIME header.

MIME entity body

When using MIME bindings in a WSDL description, the associated message may contain one or more MIME parts. When this occurs, the SOAP envelope is contained within the root MIME part. Other MIME parts contain MIME content.

- Write MIME entity body
- Process MIME entity body.

2.2.4 Transport Layer Activities

There are no additional activities for the Transport Layer required for Usage Scenarios for the Attachments Profile. The Usage Scenarios for the Basic Profile 1.0 [4] contain descriptions for activities that are executed within this Layer. Because attachment activities affect the HTTP header, additional constraints over and above those recorded in the Usage Scenarios for the Basic Profile 1.0 are imposed upon this layer by the Attachments Profile.

2.2.5 Web Service Actors

In WS-I Web services scenarios for attachments there are no additional actors from what was described in Usage Scenarios for BP 1.0 [4], Consumer and Provider. These are not related to SOAP Actors as defined in SOAP 1.1.

3 Usage Scenarios

This section defines the two Usage Scenarios developed to complement the Attachment Profile [3]:

- Explicitly bound attachments
- Referenced attachments

3.1 Explicitly bound attachments

3.1.1 Description

An attachment is bound to a mime:part in the wsdl:soap:binding for a request or a response. One or more mime:content elements are specified for the attachment. No elements that provide references to the attachment, i.e. swaRef, are allowed in either the description or within the SOAP envelope.

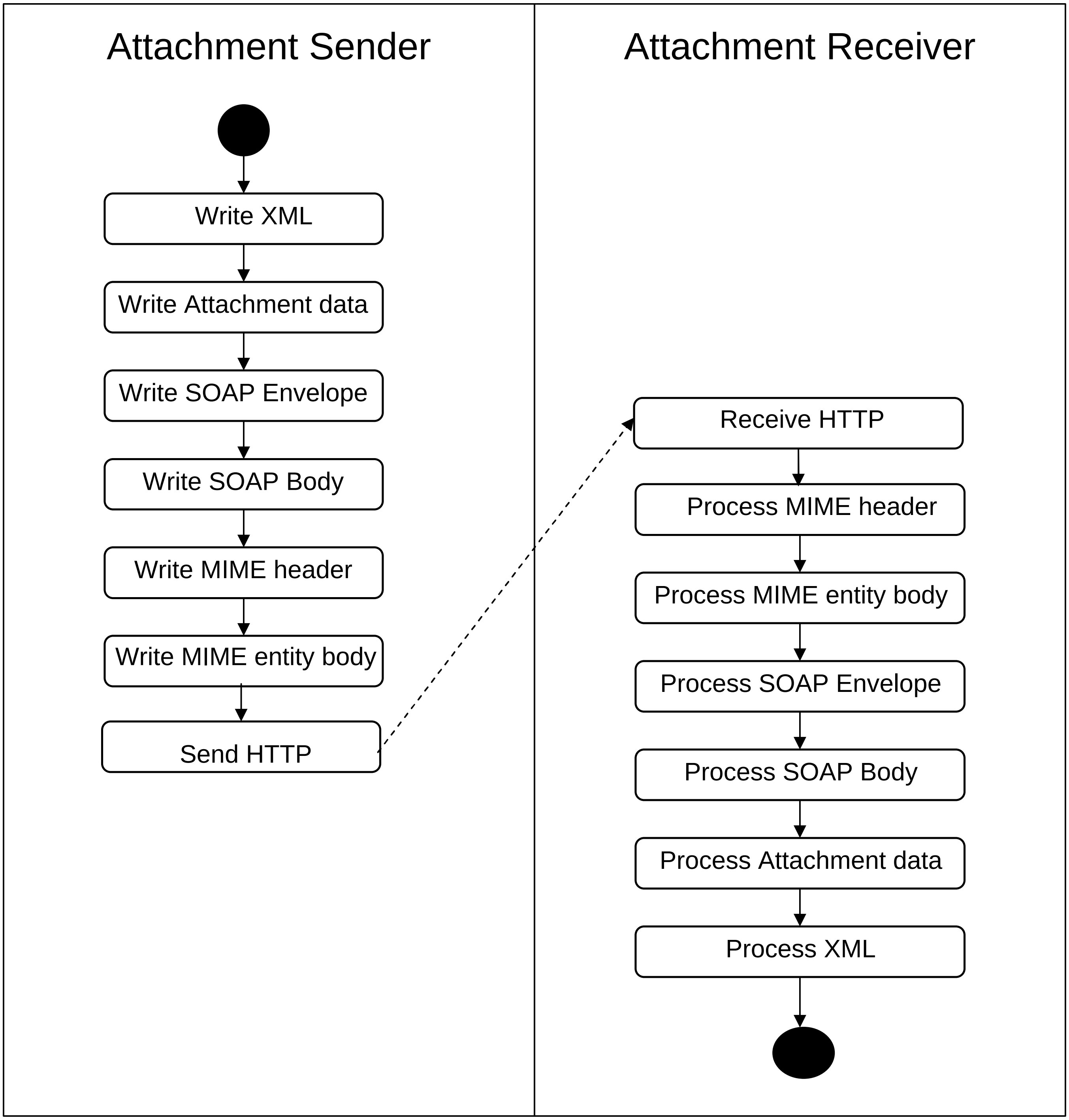
This Scenario applies to situations where there are a finite number of known attachments. There is at least one attachment described in the mime:content of the wsdl:binding. wsdl:parts that are explicitly bound to mime:content are required. Attachment order is not guaranteed, so when multiple attachments of the same content type appear in a message their order cannot be used to distinguish them.

Assumptions:

- This scenario describes a runtime set of events; it does not describe the design or deployment activities.
- Order of the events is not significant.
- The data model, the application semantics, and the transport bindings are all agreed upon and implemented a priori to this scenario.
- All parts of this scenario are defined in conformance with the guidelines and recommendations of the Attachments Profile [3] and Basic Profile 1.1 [2].
- This scenario is “composable”, that is, it may be combined with one of the Basic Profile Usage Scenarios 1.0 [4].

3.1.2 Flow

The detailed flow for this scenario, using the activities defined in Section 2.2, is described below. Each bulleted item represents the activities performed within one layer of the stack required to complete the flow. Each activity may have constraints imposed upon it from the Attachments Profile. Which actor is involved in the interaction is not significant (e.g., the Message Exchange pattern is not significant in the flow). The order of activities within an actor is not significant.



Write MIME part

An actor initiates a MIME message:

- Data Layer
  - Write XML. The payload is created according to the data model.
  - Write attachment data (optional)

- SOAP Message Layer
  - Write SOAP envelope
  - Write SOAP body (may include references to attachments)
- MIME Layer
  - Write MIME header
  - Write MIME entity body
- Transport Layer
  - Send HTTP

A receiving actor receives the MIME message:

- Transport Layer
  - Receive HTTP
- MIME Layer
  - Process MIME header
  - Process MIME entity body
- SOAP Message Layer
  - Process SOAP envelope
  - Process SOAP body
- Data Layer
  - Process XML. The data payload is processed according to the data model and dispatched to the application.
  - Process attachment data (optional)

3.1.3 Flow Constraints

The following are the flow constraints upon this Usage Scenario.

- Write XML, as defined in Basic Profile 1.1 [2]; see Usage Scenarios for the Basic Profile 1.0 [4]
- Write attachment data, as defined in Appendix 1, Section 4.5
- Write SOAP envelope, as defined in Basic Profile 1.1; see Usage Scenarios for the Basic Profile 1.0
- Write SOAP body, as define in Basic Profile 1.1; see Usage Scenarios for the Basic Profile 1.0
- Write MIME header, as defined in Appendix 1, Section 4.1
- Write MIME entity body, as defined in Appendix 1, Section 4.3
- Send HTTP, as defined in Basic Profile 1.1; see Usage Scenarios for the Basic Profile 1.0
- Receive HTTP, as defined in Basic Profile 1.1; see Usage Scenarios for the Basic Profile 1.0
- Process MIME header, as defined in Appendix 1, Section 4.2
- Process MIME entity body, as defined in Appendix 1, Section 4.4
- Process SOAP envelope, as defined in Basic Profile 1.1; see Usage Scenarios for the Basic Profile 1.0

- Process SOAP body, as defined in Basic Profile 1.1; see Usage Scenarios for the Basic Profile 1.0
- Process XML, as defined in Basic Profile 1.1; see Usage Scenarios for the Basic Profile 1.0
- Process attachment data, as defined in Appendix 1, Section 4.6

3.1.3.1 Error conditions and SOAP Fault

A SOAP Fault can be returned as defined in Basic Profile 1.1 [2], and is further constrained by the Attachments Profile 1.0. If the wsdl:output element is defined using the WSDL MIME binding, faults can be sent using the SOAP HTTP binding or the MIME binding. If a fault is sent using the MIME binding, it can include attachments. However, faults explicitly defined in the WSDL definition cannot be defined using the WSDL MIME binding.

3.1.3.2 SOAP Headers

Use of a SOAP header is optional for this scenario. If one is used, it must follow the constraints for the Write SOAP Header and Process SOAP Header activities, as defined in Section 5.7 and 5.8, respectively of WS-I Usage Scenarios for the Basic Profile 1.0 [4].

3.1.4 Description Constraints

The WSDL should follow the patterns for one of the WS-I Usage Scenarios for the Basic Profile 1.0 [4]. Only additional constraints related to definitions that have explicitly bound attachments are listed here.

3.1.4.1 types

A description of a Web service that has an explicitly bound attachment must not include a swaRef reference to that attachment. Type constraints for attachments that have references defined are covered by the Referenced Attachments Usage Scenario in Section 3.2. Any other type is permissible for describing an attachment part, but the actual type of the attachment is determined by the content type specified in the mime:content binding. There are no further constraints on the types required by this Usage Scenario.

See the WS-I Usage Scenarios for the Basic Profile 1.0. Constraints on WSDL types are listed in Section 5.12 of that document.

3.1.4.2 messages

All explicitly bound attachments must be defined with a wsdl:part in the input or output message, along with any part information for the soap:body, soap:header, soap:fault, and soap:headerFault. Message format depends upon the data model (doc/literal or rpc/literal).

3.1.4.2.1Document messages

Document message soap:body parts are composed from Schema element definitions. Parts for attachments can be composed from Schema element definitions or Schema type declarations (R2910). Attachment parts described using a global schema element definition are serialized to an XML Infoset whose root element is described by the part's referenced element (R2942).

```
<wsdl:message …>
        <wsdl:part name="Body" element="..">
        <wsdl:part name="Attachment1" type="..">
        <wsdl:part name="Attachment2" element="..">
        …
        <wsdl:part name="Attachmentn" type="..">
</wsdl:message>
```

3.1.4.2.2RPC messages

RPC message soap:body parts are composed from Schema type definitions. Parts for attachments can be composed from Schema element definitions or Schema type declarations (R2910). Attachment parts described using a global schema element definition are serialized to an XML Infoset whose root element is described by the part's referenced element (R2942).

```
<wsdl:message …>
        <wsdl:part name="Body1" type=".." />
        <wsdl:part name="Body2" type=".." />
        <wsdl:part name="Attachment1" type=".." />
        <wsdl:part name="Attachment2" element=".." />
        …
        <wsdl:part name="Attachmentn" type=".." />
</wsdl:message>
```

3.1.4.3 portTypes

There are no unique constraints for portTypes imposed by the Attachments Profile. See the WS-I Usage Scenarios for the Basic Profile 1.0, Section 5.14, for constraints on WSDL portTypes.

3.1.4.4 binding

The wsdl:binding section is extended by this scenario to explicitly bind attachment parts to mime content. Defined faults cannot be bound to mime:mutlipartRelated (R2930). Constraints in the Attachment Profile as well as Basic Profile 1.1 affect the Web service description for this scenario. The constraints listed in Section 5.15 of the WS-I Usage Scenarios for the Basic Profile 1.0 also apply.

```
<wsdl:binding …>
        <soap:binding style="rpc|document" transport="http://schemas.xmlsoap.org/soap/http">
                <wsdl:xxput …>
                        <mime:multipartRelated>
                                <mime:part>
                                        <soap:body parts="Body"…/>
                                </mime:part>
                                <mime:part>
                                        <mime:content part="Attachment1" type="…"/>
                                </mime:part>
                                <mime:part>
                                        <mime:content part="Attachment2" type="text/xml"/>
                                </mime:part>
                                …
                                <mime:part>
```

```
                                        <mime:content part="Attachmentn" type="..."/>
                                <mime:part>
                        </mime:multipartRelated>
                </wsdl:xxput>
        </soap:binding>
</wsdl:binding>
```

Other constraints are listed in Appendix 1, Section 4.10.

3.1.4.5 port

There are no unique constraints for ports imposed by the Attachments Profile. See the WS-I Usage Scenarios for the Basic Profile 1.0 [4]. Constraints on WSDL ports are listed in Section 5.16 of that document.

3.1.5 UDDI

Advertisement of Web services patterned after this scenario adheres to the "Using WSDL in a UDDI Registry, Version 1.08" Best Practice document. A `uddi:tModel` representing the Web service type references the file containing the `wsdl:binding` for the message operation. The `uddi:bindingTemplate` captures the service endpoint and references the `uddi:tModel`(s) for the Web service type.

Advertising Web services in this way enables discovery using the inquiry patterns supported by the UDDI Inquiry API set (see http://www.uddi.org/pubs/ProgrammersAPI-V2.04-Published-20020719.pdf). These include the browse pattern, the drill-down pattern and the invocation pattern.

General UDDI Constraints are listed in Section 5.17 of the Usage Scenarios for the Basic Profile [4].

3.1.6 Security

As of this writing no specific threat has been identified as being singularly relevant to this Usage Scenario, we encourage you to read the Basic Security Profile Security Scenarios [5] to better understand security requirements related to Web services in general.

3.2 Referenced attachments

3.2.1 Description

An attachment is referenced using the swaRef type for an element within the SOAP body or header for a request or a response. No mime:content elements are specified for the attachment.

This Scenario is useful when an attachment is optional, when the content type is unknown, when an attachment repeats an unknown number of times with the same or different content type, and when an attachment is logically considered to be a part of a larger data structure that appears within a SOAP envelope. Note that it is not possible to describe the referenced attachment's content type in the WSDL.

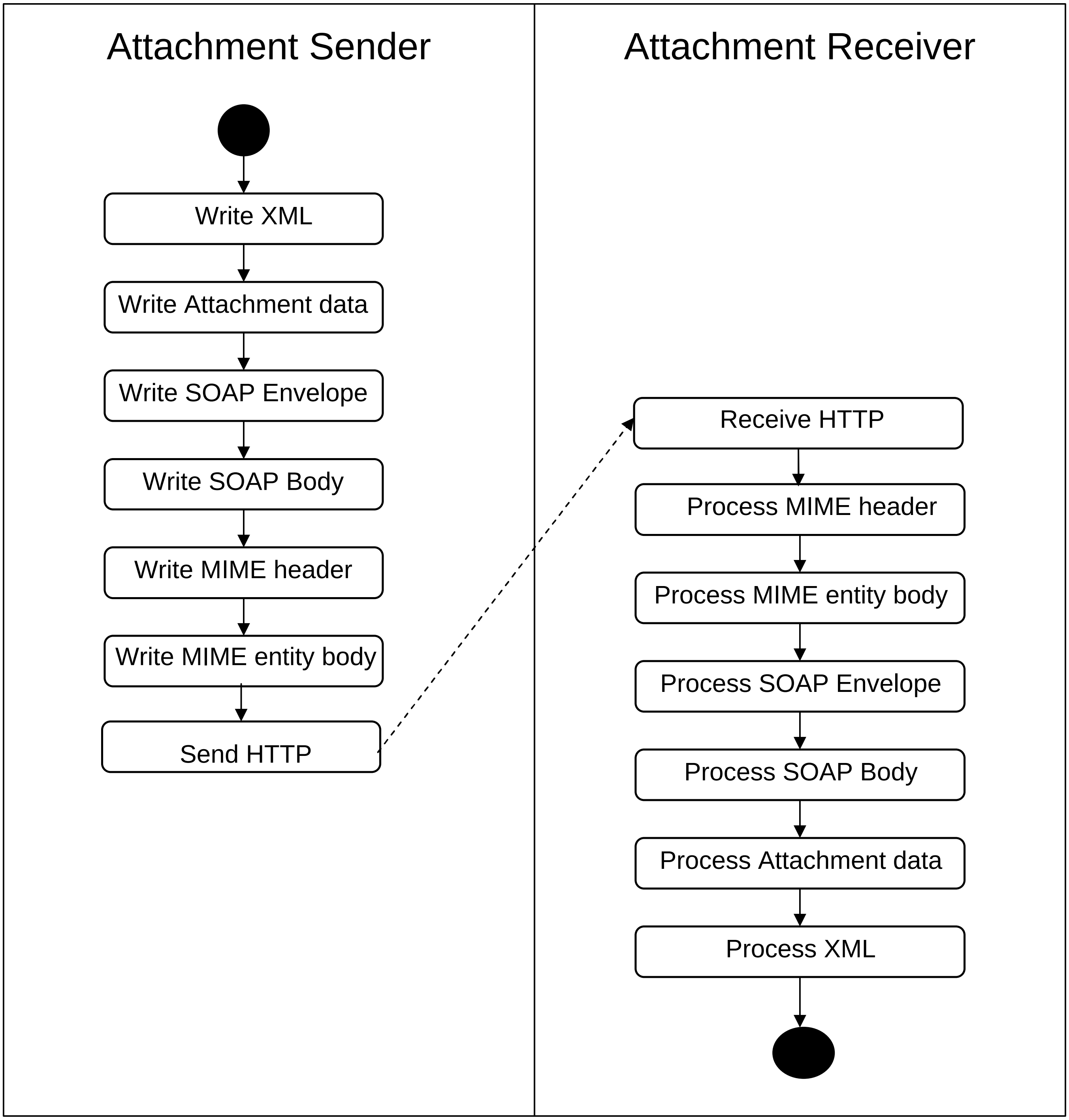
Assumptions:

- This scenario describes a runtime set of events; it does not describe the design or deployment activities.
- Order of the events is not significant.
- The data model, the application semantics, and the transport bindings are all agreed upon and implemented a priori to this scenario.

- All parts of this scenario are defined in conformance with the guidelines and recommendations of the Attachments Profile [3] and Basic Profile 1.1 [2].
- This scenario is “composable”, that is, it may be combined with one of the Basic Profile Usage Scenarios 1.0 [4].

3.2.2 Flow

The detailed flow for this scenario, using the activities defined in Section 2.2, is described below. Each bulleted item represents the activities performed within one layer of the stack required to complete the flow. Each activity may have constraints imposed upon it from the Attachments Profile. Which actor is involved in the interaction is not significant (e.g., the Message Exchange pattern is not significant in the flow). The order of activities within an actor is not significant.



Write MIME part

An actor initiates a MIME message:

- Data Layer
  - Write XML. The payload is created according to the data model.
  - Write attachment data (optional)

- SOAP Message Layer
    - Write SOAP envelope
    - Write SOAP body (will include references to attachments)
- MIME Layer
    - Write MIME header
    - Write MIME entity body
- Transport Layer
    - Send HTTP

A receiving actor receives the MIME message:

- Transport Layer
    - Receive HTTP
- MIME Layer
    - Process MIME header
    - Process MIME entity body
- SOAP Message Layer
    - Process SOAP envelope
    - Process SOAP body
- Data Layer
    - Process XML. The data payload is processed according to the data model and dispatched to the application.
    - Process attachment data (optional)

3.2.3 Flow Constraints

The following are the flow constraints upon this Usage Scenario.

- Write XML, as defined in Basic Profile 1.1 [2]; see Usage Scenarios for the Basic Profile 1.0 [4]
- Write attachment data, as defined in Appendix 1, Section 4.5
- Write SOAP envelope, as defined in Basic Profile 1.1; see Usage Scenarios for the Basic Profile 1.0
- Write SOAP body, as define in Basic Profile 1.1; see Usage Scenarios for the Basic Profile 1.0
- Write MIME header, as defined in Appendix 1, Section 4.1
- Write MIME entity body, as defined in Appendix 1, Section 4.3
- Send HTTP, as defined in Basic Profile 1.1; see Usage Scenarios for the Basic Profile 1.0
- Receive HTTP, as defined in Basic Profile 1.1; see Usage Scenarios for the Basic Profile 1.0
- Process MIME header, as defined in Appendix 1, Section 4.2
- Process MIME entity body, as defined in Appendix 1, Section 4.4
- Process SOAP envelope, as defined in Basic Profile 1.1; see Usage Scenarios for the Basic Profile 1.0

- Process SOAP body, as defined in Basic Profile 1.1; see Usage Scenarios for the Basic Profile 1.0
- Process XML, as defined in Basic Profile 1.1; see Usage Scenarios for the Basic Profile 1.0
- Process attachment data, as defined in Appendix 1, Section 4.6

3.2.3.1 Errors and SOAP Faults

A SOAP Fault can be returned as defined in Basic Profile 1.1 [2], and is further constrained by the Attachments Profile 1.0. If the wsdl:output element is defined using the WSDL MIME binding, faults can be sent using the SOAP HTTP binding or the MIME binding. If a fault is sent using the MIME binding, it can include attachments. However, faults explicitly defined in the WSDL definition cannot be defined using the WSDL MIME binding.

3.2.3.2 SOAP Headers

Use of a SOAP header is optional for this scenario. If one is used, it must follow the constraints for the Write SOAP Header and Process SOAP Header activities, as defined in Section 5.7 and 5.8, respectively of WS-I Usage Scenarios for the Basic Profile 1.0 [4].

3.2.4 Description Constraints

The WSDL should follow the patterns for one of the WS-I Usage Scenarios for the Basic Profile 1.0. Only additional constraints related to definitions that have referenced attachments are listed here.

3.2.4.1 types

A description of a Web service that has a referenced attachment may use the WS-I swaRef type to unambiguously indicate that a reference URI corresponds to an attachment contained in the message. When swaRef is used to describe an attachment reference it should only be bound to a soap:body or soap:header. It should not be bound to mime:content. There are no further constraints on the types required by this Usage Scenario.

See the WS-I Usage Scenarios for the Basic Profile 1.0 [4]. Constraints on WSDL types are listed in Section 5.12 of that document.

3.2.4.2 messages

Parts that are defined with the swaRef type or that are complex types that contain the swaRef type are included in messages along with other part information bound to the soap:body, soap:header, soap:fault, and soap:headerFault. Parts of type swaRef should only be bound to soap:body or soap:header (rpc style bindings), or swaRefs should be contained within a complex type associated with a part that is bound to soap:body or soap:header (document style binding) (R2940). Parts that are not bound are ignored. Message format depends upon the data model (doc/literal or rpc/literal).

3.2.4.2.1Document messages

Document message soap:body parts are composed from Schema element definitions. swaRef parts are either bound directly to soap:body or soap:header, or swaRefs are contained within a structure that is bound to soap:body or soap:header (shown below).

```
<wsdl:message …>
        <wsdl:part name="Body" element="ref:elementContaining_swaRef..">
</wsdl:message>
```

3.2.4.2.2RPC messages

RPC message soap:body parts are composed from Schema type definitions. swaRef parts are either bound directly to soap:body or soap:header (shown below), or swaRefs are contained within a structure that is bound to soap:body or soap:header.

```
<wsdl:message …>
        <wsdl:part name= “Body1” type=“..” />
        <wsdl:part name=“AttchRef” type=“wsiref:swaRef” />
</wsdl:message>
```

3.2.4.3 portTypes

There are no unique constraints for portTypes imposed by the Attachments Profile [3]. See the WS-I Usage Scenarios for the Basic Profile 1.0 [4], Section 5.14, for constraints on WSDL portTypes.

3.2.4.4 binding

The `wsdl:binding` section is extended by this scenario to indicate that there are mime:parts associated with the operation. While referenced attachments are not explicitly bound to mime:content, the soap:body must still be placed in a multipart:related binding element. Defined faults cannot be bound to mime:mutlipartRelated (R2930). Constraints in the Attachment Profile [3] as well as Basic Profile 1.1 [2] affect the Web service description for this scenario. The constraints listed in Section 5.15 of the WS-I Usage Scenarios for the Basic Profile 1.0 [4] also apply.

In this example an RPC style binding contains two message parts bound to soap:body – the Body1 part and the AttchRef part.

```
<wsdl:binding …>
        <soap:binding style=“rpc” transport=“http://schemas.xmlsoap.org/soap/http”>
                <wsdl:xxput …>
                        <mime:multipartRelated>
                                <mime:part>
                                        <soap:body parts=“Body1 AttchRef”…/>
                                </mime:part>
                        </mime:multipartRelated>
                </wsdl:xxput>
        </soap:binding>
</wsdl:binding>
```

In this example a Document style binding contains a single message part bound to soap:body. The part contains one or more attachment reference elements whose type is swaRef. The Basic Profile 1.1 restricts the number of message parts that can be bound to soap:body to one for document style bindings.

```
<wsdl:binding …>
        <soap:binding style=“document” transport=“http://schemas.xmlsoap.org/soap/http”>
                <wsdl:xxput …>
                        <mime:multipartRelated>
                                <mime:part>
                                        <soap:body parts=“Body”…/>
                                </mime:part>
                        </mime:multipartRelated>
                </wsdl:xxput>
        </soap:binding>
</wsdl:binding>
```

Other constraints are listed in Appendix 1, Section 4.10.

3.2.4.5 port

There are no unique constraints for ports imposed by the Attachments Profile. See the WS-I Usage Scenarios for the Basic Profile 1.0 [4]. Constraints on WSDL ports are listed in Section 5.16 of that document.

3.2.5 UDDI

Advertisement of Web services patterned after this scenario adheres to the "Using WSDL in a UDDI Registry, Version 1.08" Best Practice document. A uddi:tModel representing the Web service type references the file containing the wsdl:binding for the message operation. The uddi:bindingTemplate captures the service endpoint and references the uddi:tModel(s) for the Web service type.

Advertising Web services in this way enables discovery using the inquiry patterns supported by the UDDI Inquiry API set (see http://www.uddi.org/pubs/ProgrammersAPI-V2.04-Published-20020719.pdf). These include the browse pattern, the drill-down pattern and the invocation pattern.

General UDDI Constraints are listed in Section 5.17 of the Usage Scenarios for the Basic Profile [4].

3.2.6 Security

As of this writing no specific threat has been identified as being singularly relevant to this Usage Scenario, we encourage you to read the Basic Security Profile Security Scenarios [5] to better understand security requirements related to Web services in general.

4 Appendix 1 – Constraints

This section provides a mapping of constraints listed in the Basic Profile 1.1 [2] and Attachments Profile [3] to each of the flow activities identified in Section 2 that are new or changed within one of these profiles and within each scenario. In carrying out each activity, the listed constraints should be consulted in the Basic Profile to check for compliance with the details of the constraint.

4.1 Write MIME Header

- Use of SOAP with Attachments: R2902, R2933, R2934

4.2 Process MIME Header

- R2933, R2934

4.3 Write MIME entity body

- Root part: R2931, R2915, R2927, R2929, R2928

- MIME parts in general: E0001, R2916, R2917, R2919, R2920, R2929, R2923, R2926, R2936, R2942, R2943, R2944

4.4 Process MIME entity body

- Dereferencing attachments: R2918, R2921, R2929, R2922, R2912, R2928
- MIME parts: R2936

4.5 Write attachment data

- Encoding: R2935

4.6 Process attachment data

- Encoding: R2935

4.7 Send HTTP

- HTTP headers: R2925, R2913, R2945, R2932, R2917

4.8 Receive HTTP

- HTTP headers: R2922

4.9 Constraints on WSDL parts

- Mime parts: R2910, R2925, R2943, R2944

4.10 Constraints on WSDL bindings

- MIME bindings: R2901, R2902, R2903, R2904, R2905, R2906, R2909, R2911, R2930, R2920, R2933, R2941, R2946, R2940, R2943, R2944, R2947
- MIME binding schema: R2907, R2908

5 References

[1] WS-I Basic Profile version 1.0 from www.ws-i.org.

[2] WS-I Basic Profile version 1.1 from www.ws-i.org.

[3] WS-I Attachments Profile version 1.0 from www.ws-i.org.

[4] WS-I Usage Scenarios version 1.01 from www.ws-i.org.

[5] Basic Security Profile Security Scenarios Working Group Draft from www.ws-i.org.

6 Acknowledgements

This is the work of the WS-I Sample Applications Working Group, whose members have included:

Heidi Buelow (Quovadx), Marc Goodner (SAP), Arun Gupta (Sun), Jason Hogg (Microsoft), Tim Julien (Oracle), Barbara McKee (IBM), Jesse Pangburn (Quovadx), Vijay Rajan (Novell), Rimas Rekasius (IBM), Bruce Stephens (BEA)