More Web Services/SOAP

WS-Addressing and MTOM

Contents

- MTOM
- WS-Addressing

Overview

- Opaque Non-XML data problem
- MTOM/XOP
- MTOM in Axis2
- Axis2 MTOM Features
  - File Caching
  - Attachment Streaming
  - MTOM data binding
- Secure MTOM
- Soap With Attachment (SwA)

Binary examples

- "Attachments"
  - Images, Sounds, Video
- Existing data formats
  - In some cases not worth transforming to XMl
- Also some people want to use the QoS capabilities of WS-* with existing files or datasets:
  - E.g. Secure RM connection

Opaque Non-XML Data Problem

- Users want to leverage the structured, extensible markup conventions of XML
- Don't want to abandon existing data formats.
  - Needs existing formats to coexist with XML & to be treated as opaque sequences of octets by XML tools and infrastructure.
- Two traditional approaches
  - By value
  - By Reference

By Value

- Embedding **encoded** texts of opaque data in the XML component of data.
  - Base64 encoding
  - HexBinary encoding
- Advantages
  - Ability to process and describe data based on XML component of the data
- Disadvantages
  - Bloating of the size
    - 1.33x with Base64, 2x with HexBinary
  - Processing overhead

Example of “By Value”

```
<soap:Body>
   <m:data xmlns:m='http://example.org/stuff'>
     <m:photo xmlmime:contentType='image/png‘
   >/aWKKapGGyQ=</m:photo>
     <m:sig
   xmlmime:contentType='application/pkcs7-signature
   >Faa7vROi2VQ=</m:sig>
   </m:data>
 </soap:Body>
</soap:Envelope>
```

SwA–By Reference

```
MIME-Version: 1.0
Content-Type: Multipart/Related; boundary=MIME_boundary;
                        type=text/xml;start="<claim061400a.xml@claiming-it.com>"

--MIME_boundary
Content-Type: text/xml; charset=UTF-8
Content-Transfer-Encoding: 8bit
Content-ID: <claim061400a.xml@claiming-it.com>

<?xml version='1.0' ?><SOAP-ENV:Envelope><SOAP-ENV:Body>
     .....
      <theSignedForm href="cid:claim061400a.tiff@claiming-it.com"/>
     ....
</SOAP-ENV:Body></SOAP-ENV:Envelope>

 --MIME_boundary
Content-Type: image/tiff
Content-Transfer-Encoding: binary
Content-ID: <claim061400a.tiff@claiming-it.com>

 ...binary TIFF image...
 --MIME_boundary--
```

XOP- Merging of Two Realms

- **X**ML **O**ptimized **P**ackaging
- XOP package
  - A serialization of the XML Infoset inside an extensible packaging format
- Best of both Worlds.
  - Actually a "by reference" method (Wire format)
  - *But* Attached binary content appears as if it is inline (by value)
    - Can be thought of as being base64-encoded in the XML Document
- Results in *one* programming model

MTOM

- SOAP **M**essage **T**ransmission **O**ptimization **M**echanism
- Describes how XOP is layered in to SOAP/HTTP
- Presents an XML Infoset to the SOAP application
- Selectively encodes portions of the message
  - SOAP Intermediaries have the freedom to decide which parts to optimize or not.
- MIME multipart/related package
- Backward Compatible with SwA
  - Same wire format

MTOM Optimized SOAP Message

```
Content-Type: multipart/related; boundary=MIME_Boundary;
     type="application/xop+xml"; start="<0.1B@apache.org>";
     start-info="text/xml; charset=utf-8"

--MIME_Boundary
content-type: application/xop+xml;charset=utf-8;
                                        type="application/soap+xml;"
content-transfer-encoding: binary
content-id: <0.1B@apache.org>

<?xml version='1.0'?><soapenv:Envelope ....>
  ........
    <xop:Include href="cid:1.80@apache.org"
                 xmlns:xop=http://www.w3.org/2004/08/xop/include/>
  ........
</soapenv:Envelope>

--MIME_Boundary
content-type: application/octet-stream
content-transfer-encoding: binary
content-id: <1.80@apache.org>

.....Binary Data.....
--MIME_Boundary--
```

Infoset preservation

- This is a complex concept, but very important
- The infoset (logical model) is that the binary data is “by value”
  - EVEN IF its actually “by reference”
- Important because, for example, a digital signature must take account of the inlined data as well as the “real XML”

What is WS-Addressing

- WS-Addressing in a model for saying where messages should go to:
  - To, From, Reply-To, etc
- A standard way of talking about referencing an endpoint

Why WS-Addressing

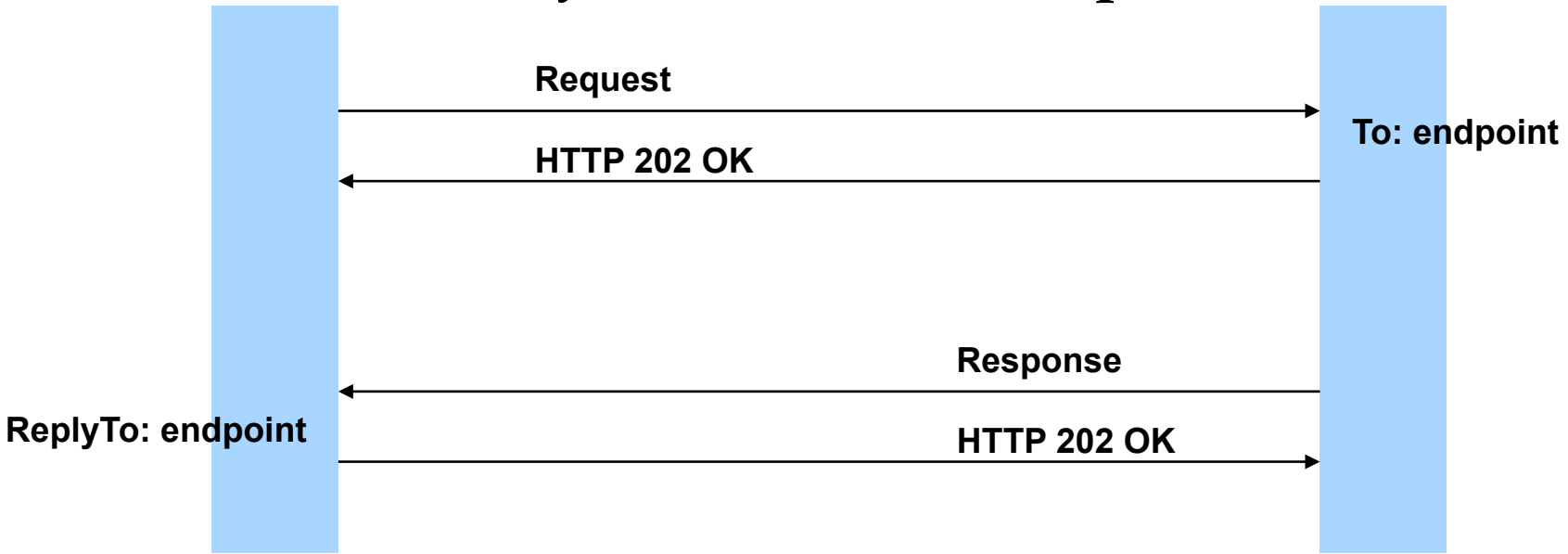
- SOAP is a message and envelope model
  - But without any addresses
- Every other messaging model has addresses:
  - MQ – queue's and managers
  - Web – URLs
  - Ordinary Post
    - Standard approach to addresses
    - Standard places on letters to put "to" and "replyto"

WS-Addressing example

```
<S:Header>
   <wsa:MessageID>http://example.com/6B29FC40-CA47-1067
   </wsa:MessageID>
   <wsa:ReplyTo>
    <wsa:Address>http://example.com/business/client1</wsa:Address>
   </wsa:ReplyTo>
   <wsa:To>http://example.com/fabrikam/Purchasing</wsa:To>
   <wsa:Action>http://example.com/fabrikam/SubmitPO</wsa:Action>
</S:Header>
```

Asynchronous

- WS-Addressing implies an asynchronous model
  - Even over a "synchronous" transport like HTTP



Doing Synchronous with WSA

- If there is no “Reply-To” there is an implicit value of “anonymous”:
  - http://www.w3.org/2005/08/addressing/anonymous
  - You can explicitly use this URI as well
- This implies the response will flow back on the transport-defined backchannel
  - HTTP or TCP response
  - Email: reply-to header
  - JMS: replyTo destination
  - Etc
- What do you think this means?
  - http://www.w3.org/2005/08/addressing/none

MessageIDs

- Can be used to correlate responses in an asynchronous model

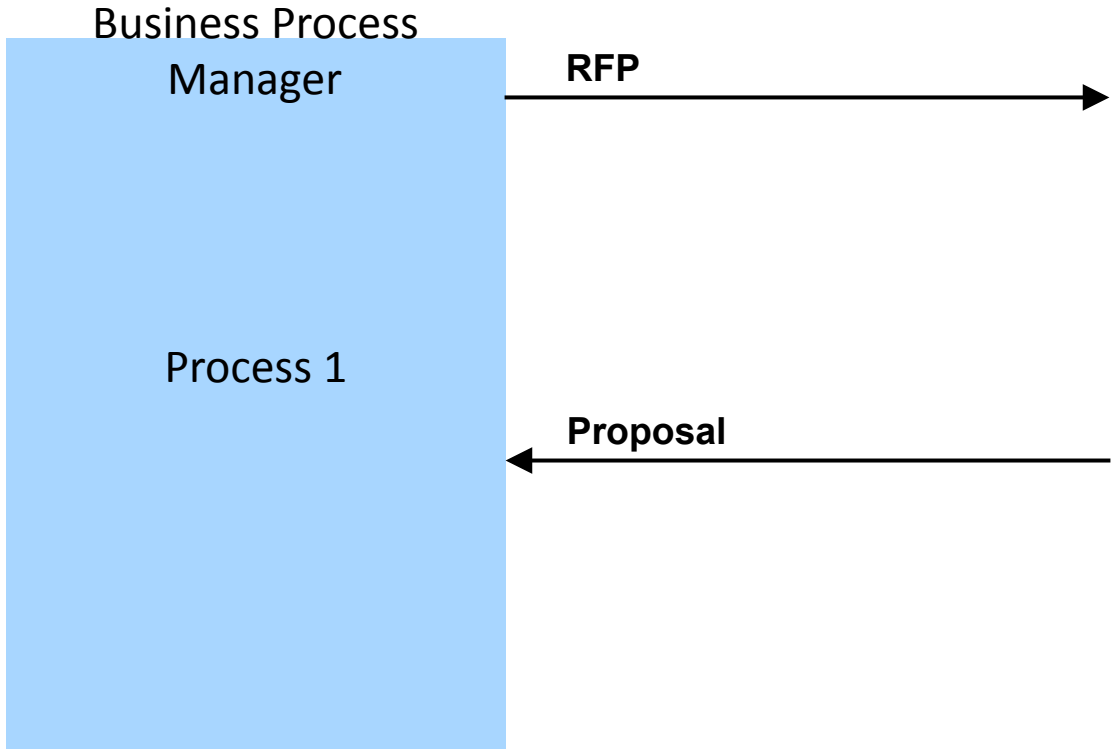
Endpoint References

- An EPR is a “package” containing the address information for an endpoint

```
<wsa:EndpointReference xmlns:wsa="....">
  <wsa:Address>
    http://example.com/fabrikam/acct
  </wsa:Address>
</wsa:EndpointReference>
```

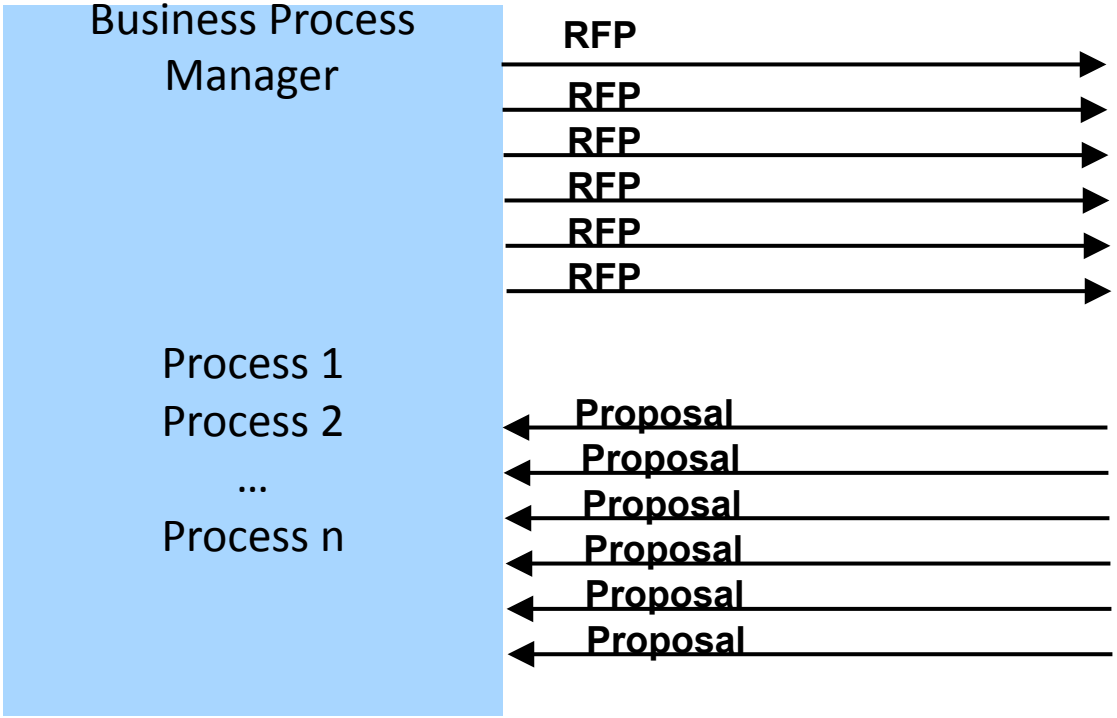
Addresses

- Is a URL/URI enough?



Addresses

- How do you qualify which responses belong to which "process instance"?



A “full” EPR

```
<wsa:EndpointReference
    xmlns:wsa="http://www.w3.org/2005/08/addressing"
    xmlns:wsaw="http://www.w3.org/2006/02/addressing/wsdl"
    xmlns:fabrikam="http://example.com/fabrikam">
  <wsa:Address>http://example.com/fabrikam/acct</wsa:Address>
  <wsa:Metadata>
     <wsaw:InterfaceName>fabrikam:Inventory</wsaw:InterfaceName>
  </wsa:Metadata>
  <wsa:ReferenceParameters>
     <fabrikam:CustomerKey>123456789</fabrikam:CustomerKey>
     <fabrikam:ShoppingCart>ABCDEFG</fabrikam:ShoppingCart>
  </wsa:ReferenceParameters>
</wsa:EndpointReference>
```

Reference Parameters

- When using a given reference
  - Copy the Reference Parameters into the SOAP header
  - And tag them

```
<S:Header>
   <wsa:To>http://example.com/fabrikam/acct</wsa:To>
   <wsa:Action>...</wsa:Action>
   <fabrikam:CustomerKey wsa:IsReferenceParameter='true'>
      123456789
   </fabrikam:CustomerKey>
   <fabrikam:ShoppingCart wsa:IsReferenceParameter='true'>
      ABCDEFG
   </fabrikam:ShoppingCart>
</S:Header>
```

Metadata “Bag”

```
<wsa:EndpointReference
    xmlns:wsa="http://www.w3.org/2005/08/addressing"
    xmlns:wsaw="http://www.w3.org/2006/02/addressing/wsdl"
    xmlns:fabrikam="http://example.com/fabrikam">
  <wsa:Address>http://example.com/fabrikam/acct</wsa:Address>
  <wsa:Metadata>
     <wsaw:InterfaceName>fabrikam:Inventory</wsaw:InterfaceName>
  </wsa:Metadata>
  <wsa:ReferenceParameters>
     <fabrikam:CustomerKey>123456789</fabrikam:CustomerKey>
     <fabrikam:ShoppingCart>ABCDEFG</fabrikam:ShoppingCart>
  </wsa:ReferenceParameters>
</wsa:EndpointReference>
```

Using EPRs in the “body”

- There are scenarios where you might need to send an EPR to someone:
  - I book a trip with a travel agent
  - They need to pass my contact details (endpoint) to the airline
  - The airline calls my service directly
- Similar scenarios exist in middleware – for example transaction co-ordination

Questions?

- Resources:
  - http://www.w3.org/TR/soap12-mtom/
  - http://cxf.apache.org/docs/mtom.html
    http://www.w3.org/2002/ws/addr/
  - http://cxf.apache.org/docs/ws-addressing.html