

# Handling Binary Data With Apache Axis2

**N.K. Thilina Gunarathne**

University of Moratuwa

Colombo, Sri Lanka

thilina@apache.org

<http://people.apache.org/~thilina/>

In collaboration with



# Overview

- Apache Web Services Project
- Apache Axis 2
- Opaque Non-XML data problem
- Historical solutions
- MTOM/XOP
- MTOM in Axis2
- Axis2 MTOM Features
- Secure MTOM Demonstration

# Vision of WS project

Build web services components that can be assembled into a coherent, interoperable, complete, high performance web services stack

# History

- First there was Apache SOAP
  - donated by IBM
- Axis 1.x designed as a follow-on
  - SAX based parsing
  - Handler architecture
  - Highly successful
    - Re-used in many commercial products
- Forming of Apache WS-Project

# Axis

- “Production Quality”
  - Axis/C++ 1.5
  - Axis/Java 1.3
- “Beta”
  - Axis2/Java
- “Early alpha”
  - Axis2/C M2
  - Axis2/PHP

# Projects

- Axis Add-ons (was WS-FX)
  - WSS4J
    - Web Services Security support
  - Sandesha
    - Web Services Reliable Messaging
  - Kandula
    - WS – Co-ordination, Atomic Transaction and BusinessActivity
  - Addressing
    - Ws-Addressing

# Other Projects

- JaxMe
- JUDDI
- Mirae
- Muse
- Pubscribe
- Scout
- Synapse \*
- Tuscany \*
- Woden \*
- WSIF
- WS-Commons
- EWS

# Motivation for Axis2

- Why do we need Axis2?
  - Changes to the Web services landscape
    - WS-Addressing, Reliable Messaging, Composability
  - Performance
    - Parsers, Optimising based on use
  - Ease of use
    - Deployment of new capabilities, service deployment

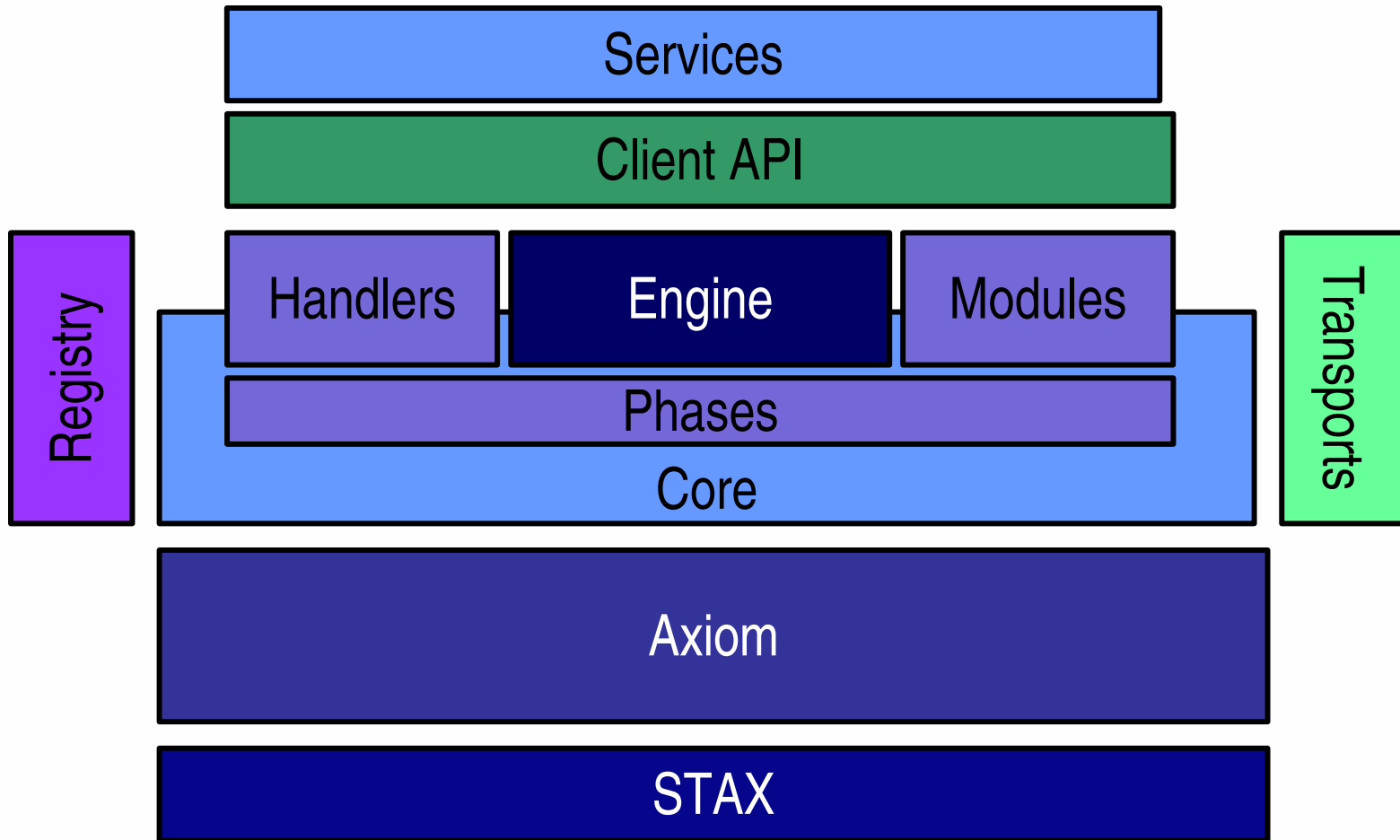


# Axis2 Main Features

- New high performance object model (Axiom)
- Asynchronous and message based model
  - Better support for MEPs(Message Exchange Patterns)
- Improved support for composition and extensibility
  - Modules and phases
- Enhanced deployment and isolation
- Flexible Data Binding
- MTOM support
  - standardised approach to attachments
- REST support



# Architecture Overview



# Opaque Non-XML data problem

- Users want to leverage the structured, extensible markup conventions of XML
- Don't want to abandon existing data formats that do not adhere to XML syntax.
  - 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 opaque data by **encoding with some form** as element or attribute content of the XML component of data.
  - Base64 encoding
  - HexBinary encoding
- Advantages 😊
  - Ability to process and describe data based and looking only on XML component of the data
- Disadvantages ☹️
  - Bloating of the size
  - Processing overhead

# What is Base64 encoding

- A-Z + a-z + 0-9 + two other symbols
- Take 3 bytes with 8 bits and represent with 4 printable characters

# By reference

- Attaching pure binary data as external unparsed general entities outside of the XML document
- Embedding reference URI's to those entities in XML.
- Advantages ☺
  - No encodings used
  - Efficient
- Disadvantages ☹
  - Heavy reliance on DTDs (SOAP prohibits DTD's)
  - Two data models
  - Causes the technologies for processing/description of data based on XML component of the data, to malfunction

— WS-Security

# SOAP With Attachments (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--

# XML Optimized Packaging (XOP)

- XOP package
  - Created by placing a serialization of the XML Infoset inside of an extensible packaging format
- XOP:Include element
- Merging of two realms. 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 same programming model

# SOAP Message Transmission Optimization Mechanism (MTOM)

- “Optimizing the transmission and/or wire format of a SOAP message by selectively encoding portions of the message, while still presenting an XML Infoset to the SOAP application.”
- Describes how XOP is layered in to SOAP/Http
- MIME multipart/related package
- XML containing SOAP envelope as the root part
- SOAP Intermediaries have the freedom to decide parts need to be optimized
- Backward Compatible with SwA
  - Same wire format

# Mime Basics

- MIME (Multipurpose Internet Mail Extensions)
  - Bag full of RFC's

# MTOM Optimized SOAP Message

```
Content-Type: multipart/related; boundary=MIMEBoundary4A7AE55984E7438034;  
    type="application/xop+xml"; start="<0.09BC7F4BE2E4D3EF1B@apache.org>";  
    start-info="text/xml; charset=utf-8"
```

```
--MIMEBoundary4A7AE55984E7438034  
content-type: application/xop+xml; charset=utf-8; type="application/soap+xml";  
content-transfer-encoding: binary  
content-id: <0.09BC7F4BE2E4D3EF1B@apache.org>
```

```
<?xml version='1.0' encoding='utf-8'?>  
<soapenv:Envelope xmlns:soapenv="...."....>  
    .....  
        <xop:Include href="cid:1.A91D6D2E3D7AC4D580@apache.org"  
                    xmlns:xop="http://www.w3.org/2004/08/xop/include">  
            </xop:Include>
```

```
    .....  
</soapenv:Envelope>  
--MIMEBoundary4A7AE55984E7438034  
content-type: application/octet-stream  
content-transfer-encoding: binary  
content-id: <1.A91D6D2E3D7AC4D580@apache.org>
```

*Binary Data.....*

```
--MIMEBoundary4A7AE55984E7438034--
```

# Binary data handling using Axis 2

- MTOM/XOP
  - Deferred building at two levels
  - Infoset level support
    - Programming model - AXIOM
  - File Caching support with threshold value
  - Secure MTOM
- SwA

# AXIOM / OM

- AXIS Object Model
  - A lightweight, low-memory footprint, high-performance object model
- Multiple ways of looking at the same data
  - Tree structure
  - STAX pull stream
  - SAX API
- Supports caching of the tree and pull stream
  - So that you can read the stream and then read the tree
  - Or read the tree and then read the stream
- Supports MTOM – built in binary representation
- Supports DataBinding frameworks
  - *via* STAX and SAX
  - JAXB, Apache XMLBeans

# Programming Model

- AXIOM has the ability to hold binary data
- OMText holds raw binary content
  - XOP is capable of optimizing XML Schema base64Binary datatype data only.
  - Infoset preservation (either optimized or not)
- Ability to specify whether to optimize or not the binary data in a OMText instance.
- Optimum efficiency
  - smaller binary attachments - not optimized (base64encoding)
  - larger attachments - optimized

# Programming Model

- Using a DataHandler

```
OMElement imageElement = fac.createOMElement("image", omNs);  
Image image;  
image = new ImageIO().loadImage( .... );  
ImageDataSource dataSource = new  
    ImageDataSource("test.jpg",image);  
DataHandler dataHandler = new DataHandler(dataSource);
```

```
OMText textData = fac.createText(dataHandler, true);  
imageElement.addChild(textData);
```

- Using Base64 encoded string

```
String base64String = "some_string"; OMText binaryNode =  
    fac.createText(base64String,"image/jpg",true);
```

# Deferred building

- XML parsing level (StAX)
- MIME parts

# Enabling MTOM

- Client Side
  - Set the "enableMTOM" property in the Options to "true"
  - Check whether SOAP envelope actually **contains** optimizable content (OMText nodes containing binary content with optimizable flag "true") before serializing as MTOM optimized
- Server Side
  - Identifies incoming messages based on the content-type and de-serializes accordingly
  - Enable MTOM for outgoing messages – Globally in Axis2.xml

```
<parameter name="enableMTOM" locked="false">true</parameter>
```

# File Caching of Attachments

- Ability to handle very large attachments without buffering them in memory
- Streams the incoming MIME parts directly in to temporary files

```
<parameter name="cacheAttachments" locked="false">true</parameter>  
<parameter name="attachmentDIR" locked="false">temp directory</parameter>  
<parameter name="sizeThreshold" locked="false">4000</parameter>
```

# Securing MTOM messages

- Axis2 Security module works seamlessly with MTOM optimised messages.
- At the recent Microsoft PDC
  - MS demonstrated interop between Indigo and Axis2:
    - SecureMTOM
    - WS-Addressing, MTOM, WSSec 1.0
      - encryption
      - digital signatures

# MTOM in WSDL???

- Still a gray area
- Axis2 uses the schema type base64Binary
- MSFT uses their own policy definition
- Ideally we should use standardized policy.....

## SOAP with Attachments support

- Handles pure SwA messages at the inflow
- MTOM is backward compatible with SwA
  - Same wire format **MTOM** , **SwA**
  - Any valid MTOM optimized message is a valid SwA message
- Can use MTOM programming model to send messages to SwA endpoints

```
public void init(MessageContext msgctx) {
    this.msgctx = msgctx;
}
public OMElement echoAttachment(OMElement omEle) {
    .... //Retrieve the content ID
    MIMEHelper attachments =
(MIMEHelper)msgctx.getProperty(MIMEHelper.ATTACHMENTS);
    .....
}
```

# Summary

- Axis2 is a major improvement of the Web services core engine of Apache
  - Nearly at 1.0
- MTOM/XOP is the standardized approach to send Binary Attachments with SOAP and it's going to stay for years.
- Axis2 provides deeply integrated, efficient, interoperable support for MTOM/XOP.
- Axis2 has basic support for SwA.

# References

- <http://ws.apache.org/axis2/>
  - [http://ws.apache.org/axis2/0\\_94/mtom-guide.html](http://ws.apache.org/axis2/0_94/mtom-guide.html)
- [www.wso2.net](http://www.wso2.net) - Web services developer portal
- Tungsten - application server for Web services
  - <http://www.wso2.com/products/tungsten/>
- <http://www.w3.org/TR/2004/PR-soap12-mtom-20041116/>
- <http://www.w3.org/TR/2004/PR-xop10-20041116/>
- <http://www.w3.org/TR/SOAP-attachments>
- Mailing Lists
  - <http://ws.apache.org/axis2/mail-lists.html>
- <http://www.fourmilab.ch/webtools/base64/>
- [www.mhonarc.org/~ehood/MIME/](http://www.mhonarc.org/~ehood/MIME/)



**Thank You !!!**

