

Apache Synapse ESB



Asankha Perera

asankha@wso2.com

The ApacheCon ASIA 2006 logo is set against a red background. It features a colorful feather on the left, the text "ApacheCon" in a large, bold, yellow font, and "ASIA 2006" in a smaller, white font below it. To the right of the text is a stylized orange floral or scrollwork design.

ApacheCon
ASIA 2006

Apache Synapse

The small print

Synapse is an effort undergoing incubation at the Apache Software Foundation (ASF), sponsored by the Web Services PMC. Incubation is required of all newly accepted projects until a further review indicates that the infrastructure, communications, and decision making process have stabilized in a manner consistent with other successful ASF projects.

While incubation status is not necessarily a reflection of the completeness or stability of the code, it does indicate that the project has yet to be fully endorsed by the ASF.

In other words.. An incubator project *for the moment*

Contents

- What is an ESB?
- What can Synapse do?
- Synapse deployment
- Mediation
- Configuration Language
- Extending Synapse
- Status and Future

What is an ESB?

- A common definition

“Any to any data connectivity and transformation (including Web Services) built on an advanced, proven, reliable middleware infrastructure”

What can Synapse do?

- Connect
 - Route messages based on XPath, Regex etc
 - Deal with mismatches
 - Initiate/Terminate WS-RM/Sec
 - Switch
 - POX or REST to SOAP to JSON
 - Transport - JMS/HTTP/SMTP/HTTPS/..
- Virtualization
 - Virtual to real URI mapping

What can Synapse do?... (Contd)

- Manage
 - Logging
 - Tracking - adding headers
 - Authentication and Authorization
 - Schema validation
 - Monitoring and statistics*
 - Failover, retry and load balance*

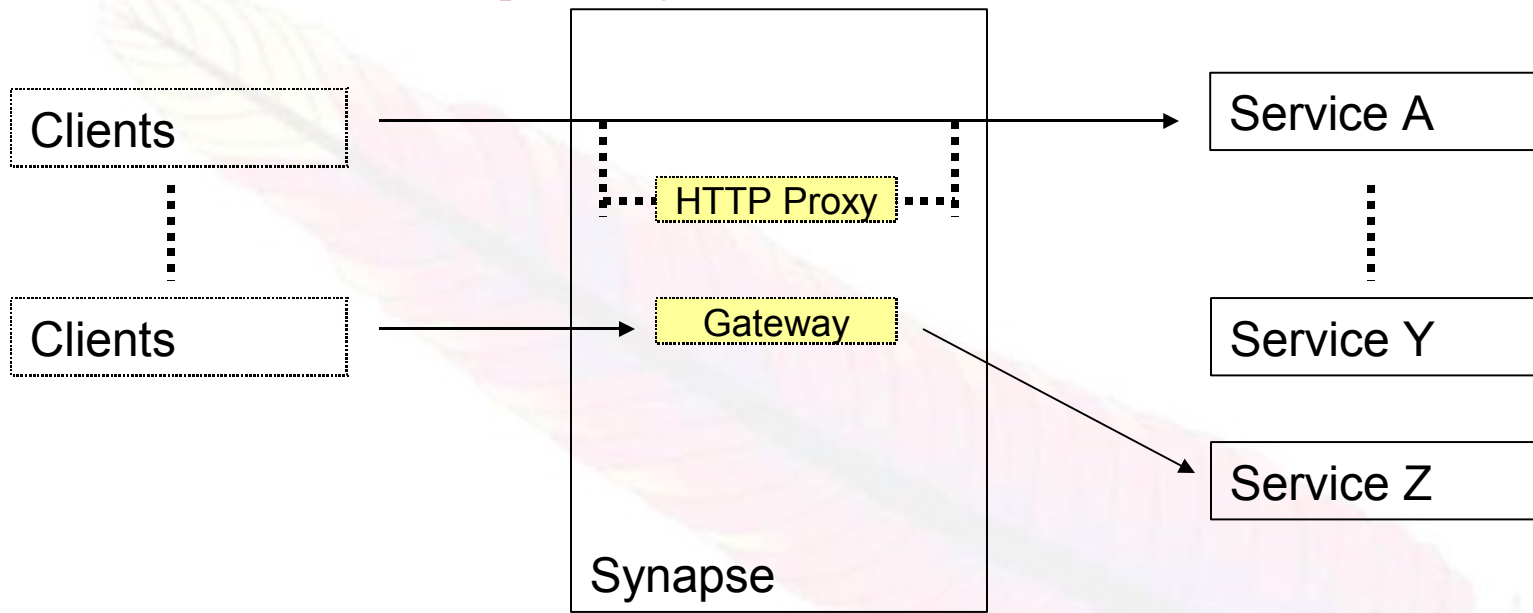
What can Synapse do?... (Contd)

- Transform
 - XSLT transforms between schemas
 - Javascript/E4X
 - SOAP to JSON

Web services model

- Connectivity - SOAP, MTOM, REST
- Routing
 - WS-Addressing
- Description - WSDL, WS-Policy
- Security
 - WS-Sec, WS-Trust, WS-SecureConversation
- Guaranteed delivery
 - WS-Reliable Messaging

Deployment models

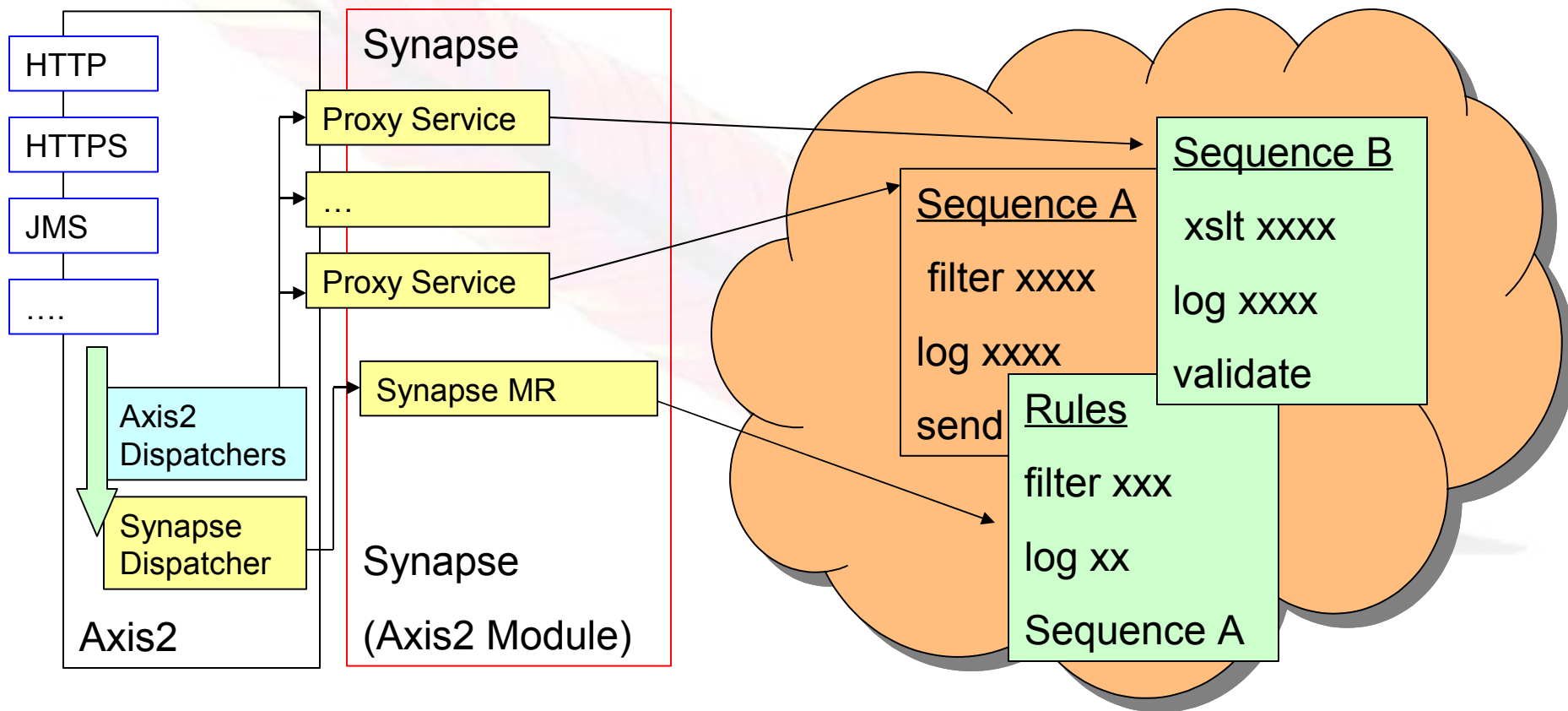


- Transparent proxy
- Gateway

Deployment

- Transparent Proxy
 - Synapse used as a HTTP Proxy
 - With or without WS-Addressing headers
 - With a `Smart Client`
 - WS-Addressing specifies target but transport, Synapse
 - Pure rule based mediation
- Gateway
 - `Proxy Services` model
 - Hosting virtual endpoints in Synapse
 - Redirect to an Endpoint or Apply rules/sequences
 - Allows Transport, QoS switching etc. for the `new` service
 - Endpoint based model
 - Rule based mediation

Message flow into Synapse



Message mediation

- Aim is to build in the common functions:
 - Filter, Switch - on XPath and/or Regex
 - Header - add/remove
 - Transform - XSLT
 - Schema Validation
 - Create fault messages
 - Log, Drop, Send
 - Engage QoS: WS-Security/RM etc
 - Class mediator - custom Java mediation
 - ...

Synapse configuration

- By default is XML file based
 - Could be looked up from a registry
 - Rules, Endpoint definitions, Resources etc. can be cached & updated dynamically from a registry
 - Provides ability to dynamically change behaviour!
- Configuration does not `have` to depend on an XML configuration, could be extended

Configuration Language

```
<synapse>
  <registry ...>

  <definitions>
    <sequence ...>
    <endpoint ...>
    <property ...>
  ...
  <proxies>
    <proxy ...>
  ...
  <rules>
    <filter source="get-property('To')" regex=".*\/StockQuote.*">
      <sequence ref="stockquote"\/>
    </filter>
```



Configuration Elements

- Registry
 - Allows Synapse to pull configuration and resources (e.g. WSDL for proxy services, Schemas for validations etc) from a central registry
 - Content is managed by the registry
 - Content is cached according to meta information specified by the registry
 - Expired content is refreshed if necessary and re-cached

Configuration Elements .. (contd)

- Definitions - Sequences

- A chain of actions to take on a given message
- Sequences are named, and thus can be referenced and reused

e.g.

```
<sequence name="stockquote">
  <filter xpath="//*[wsx:symbol='MSFT']">...>
    <makefault>
      <code value="tns:Receiver" ../>
        <reason value="Error reason.."/>
      </makefault>
    </filter>
  <send/>
</sequence>
```

Configuration Elements .. (contd)

- Definitions - Endpoints
 - A simple local representation of a remote endpoint
 - Named and can be re-used
 - WS-Security/RM settings can be defined based on the endpoint and reused

e.g.

```
<endpoint name="stockquote"  
address="http://localhost:9000/axis2/services/SimpleStockQuoteService"/>
```

Configuration Elements .. (contd)

- Definitions - Properties
 - Named properties can be static or dynamic
 - Static properties can be
 - String literal
 - Looked up from a URL
 - Inlined XML
 - Dynamic properties are looked up from a registry
 - cached/refreshed as requested

e.g.

```
<set-property name="xslt-key-req" src="file:xslts/transform.xslt"/>  
<set-property name="version" value="0.1"/>  
<set-property name="validate_schema" key="validate_schema"/>  
<set-property name="my-xslt"> <xsl:stylesheet ../></set-property>
```

Configuration Elements .. (contd)

- Proxy Services
 - Can exhibit different behaviour than the real service
 - different transports, QoS, Schema etc
 - Provide a base WSDL and Attach Policies
 - ?WSDL will expose new WSDL
 - Support for REST/POX proxies, JMS proxies etc
 - Messages can be
 - Directly forwarded to an endpoint
 - Processed as per a named sequence
 - Processed as per default rules

Configuration Elements .. (contd)

- Mediators
 - Node mediators (List mediator)
 - Filter mediators
 - conditional
 - Leaf mediators
 - no child mediators
 - Many useful mediators provided out of the box in Core synapse
 - Additional mediators provided as extensions

Extending Synapse

- Can easily add custom mediators
 - Simple Java extensions loaded through class mediator

```
<class name="class-name">  
  <property name="string" (value="literal" | expression="xpath")/>*</class>
```
 - Spring bean mediators
 - Use Spring to wire up a mediator

```
<spring:spring bean="exampleBean" key="spring_config"/>
```
 - Custom mediators bundled as packages (JARs)
 - Uses dynamic discovery to register available mediators

User mediators

- Mediator class (bean) must implement the Mediator interface

```
boolean mediate(MessageContext mc)
```

Return:

true : continue processing

false: abort

Programming model

- The MessageContext

- holds references to the
 - SynapseConfiguration
 - Holds all configuration information
 - SynapseEnvironment
 - Access to the underlying SOAP engine to receive and send messages
- Holds local properties
- A representation of a SOAP infoset
 - Get/set Envelope
 - Get/set To, From, ReplyTo, FaultTo etc
 - isResponse(), isMTOM(), is Rest()

Status

- Being Incubated in Apache since August 2005
 - All Apache contributed code
 - Aiming to graduate
- M1 release January 2006
 - Core functions
- M2 release June 2006
 - Cleanup of configuration language
- 0.90 release - soon!
 - Proxy services
 - Repackaged as a MAR (Axis2 Module)
 - Registry support and dynamic properties
 - Error handling support and new mediators
- *Looking to graduate from incubation*

Getting Involved

- Home Page / Wiki
 - <http://incubator.apache.org/synapse/>
 - <http://wiki.apache.org/ws/Synapse>
 - <http://wiki.apache.org/incubator/Synapse/InProgress>
- SVN
 - <http://svn.apache.org/repos/asf/incubator/synapse/trunk/java>
- Bug reports
 - <http://issues.apache.org/jira/browse/SYNAPSE>
- Join US
 - synapse-dev@ws.apache.org

Future

- 1.0 Release / Graduation
 - Support for Registries
 - Support for WS-Security/RM
 - Support for fault handling
 - Support for monitoring/statistics
 - Documentation, Usability
 - More examples and an example suite
- The core model is also amenable to a Synapse/C implementation



Questions

