Apache Synapse

Paul Fremantle

paul@wso2.com

http://bloglines.com/blog/paulfremantle

About me

- EX IBM STSM
  - developed the IBM Web Services Gateway
  - Apache WSIF
  - Apache Axis C/C++
  - JWSDL/WSDL4J – now Woden
- Co-founded WSO2
- Committer on Apache Synapse
- Member of the ASF

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 – and what is *our* mindset
- Connect, Manage, Transform
- Synapse structure and model
- Deployment
- Configuration
- Programming model

What is an ESB?

A common ESB definition

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

ESB definition

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

which means

- Our existing middleware re-branded as an SOA platform, with some new web services adapters at the edges

Confusion

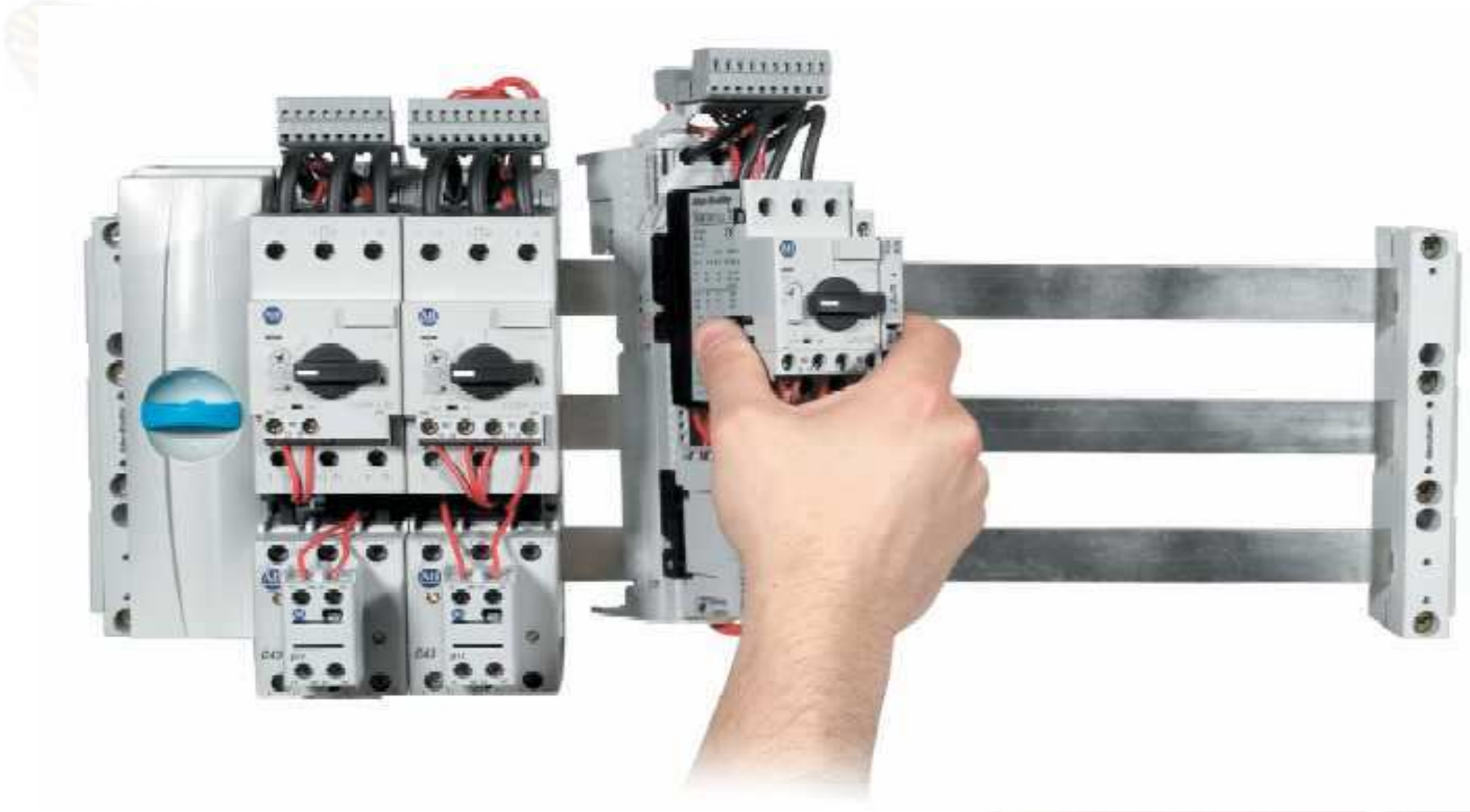
- “Lots of people use the ESB word these days to talk about lots of different kinds of technology, so its a pretty meaningless term. (e.g. Some Synapse folks claim its an ESB too while others claim its an Axis2 mediation framework).”

  James Strachan

So what *is* an Enterprise Service Bus?

A bus bus

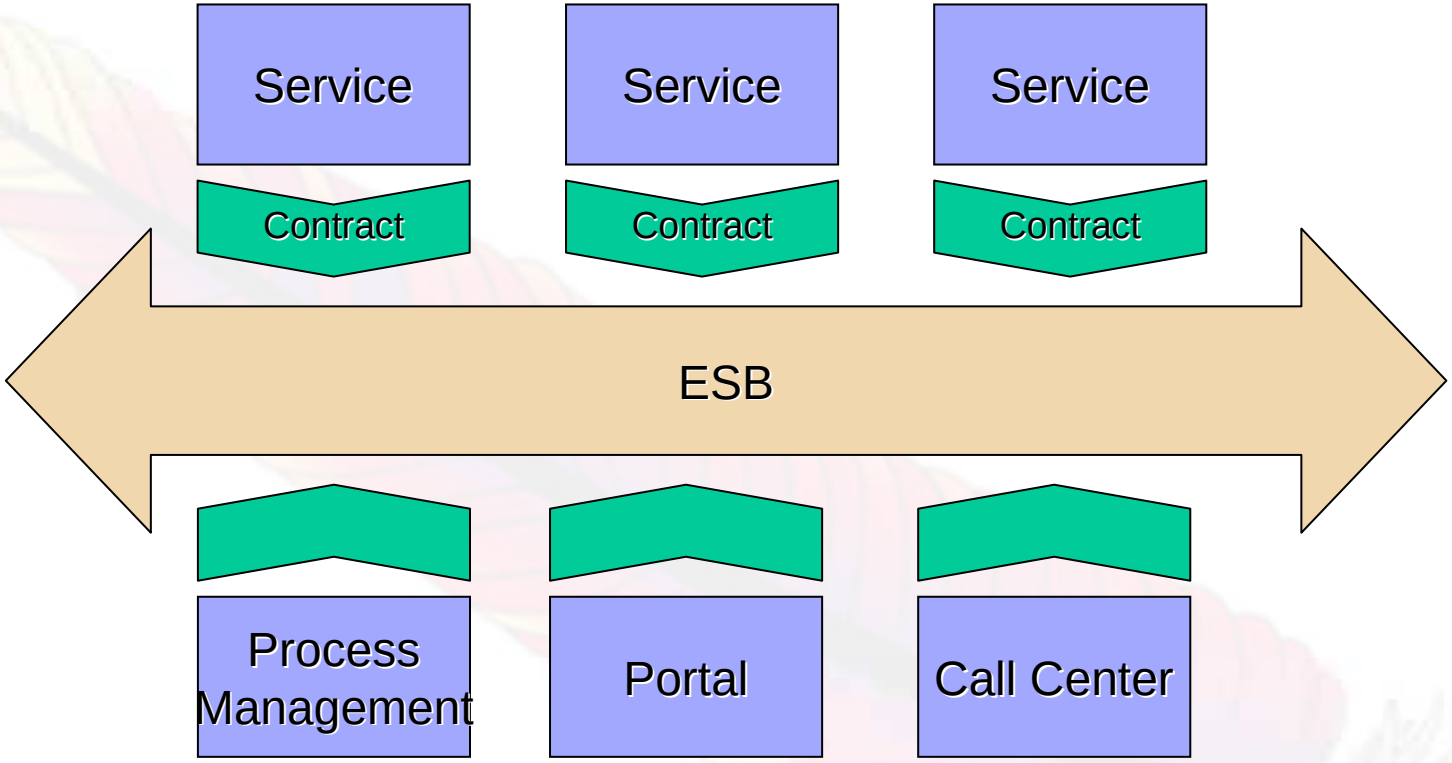
Busbar



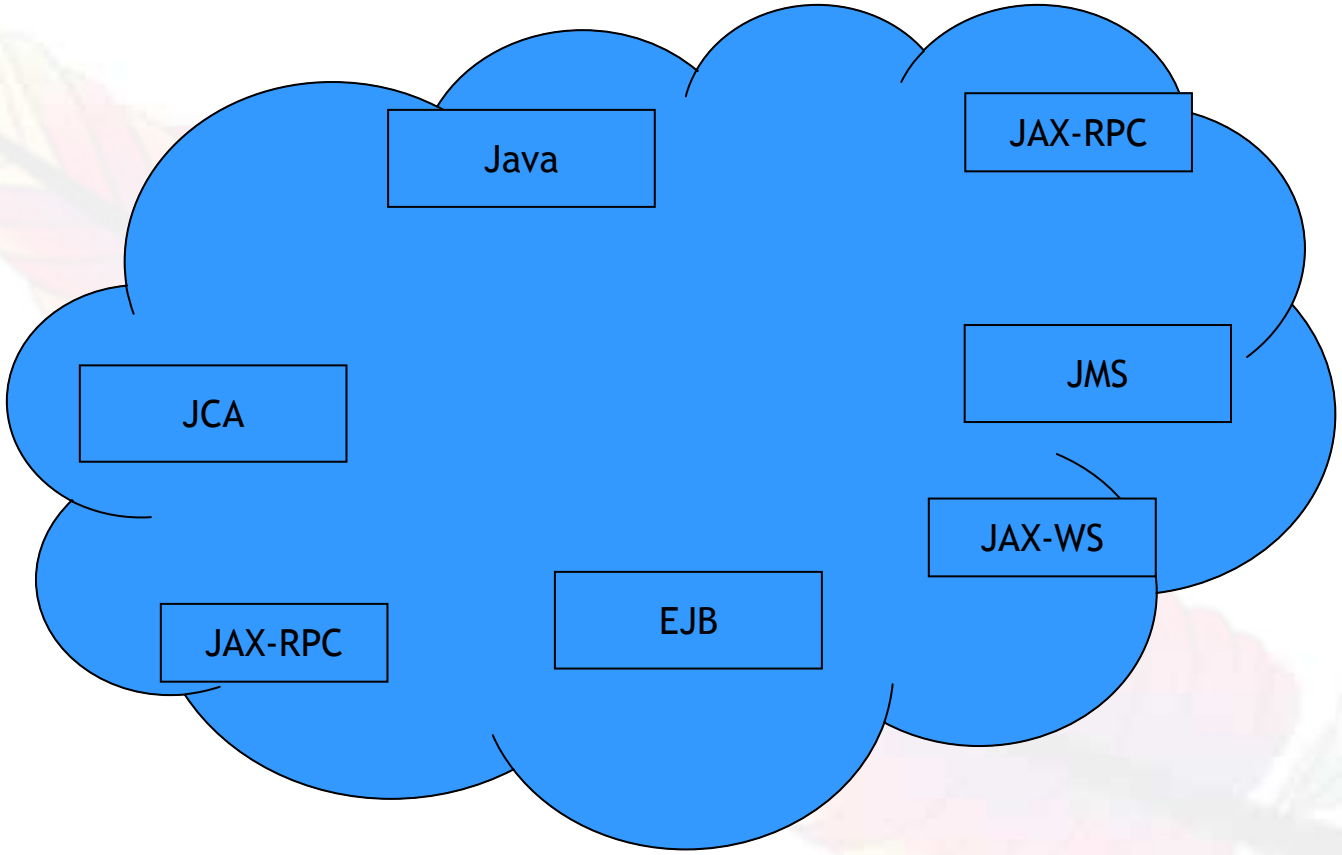
Bus

“The original ESB”

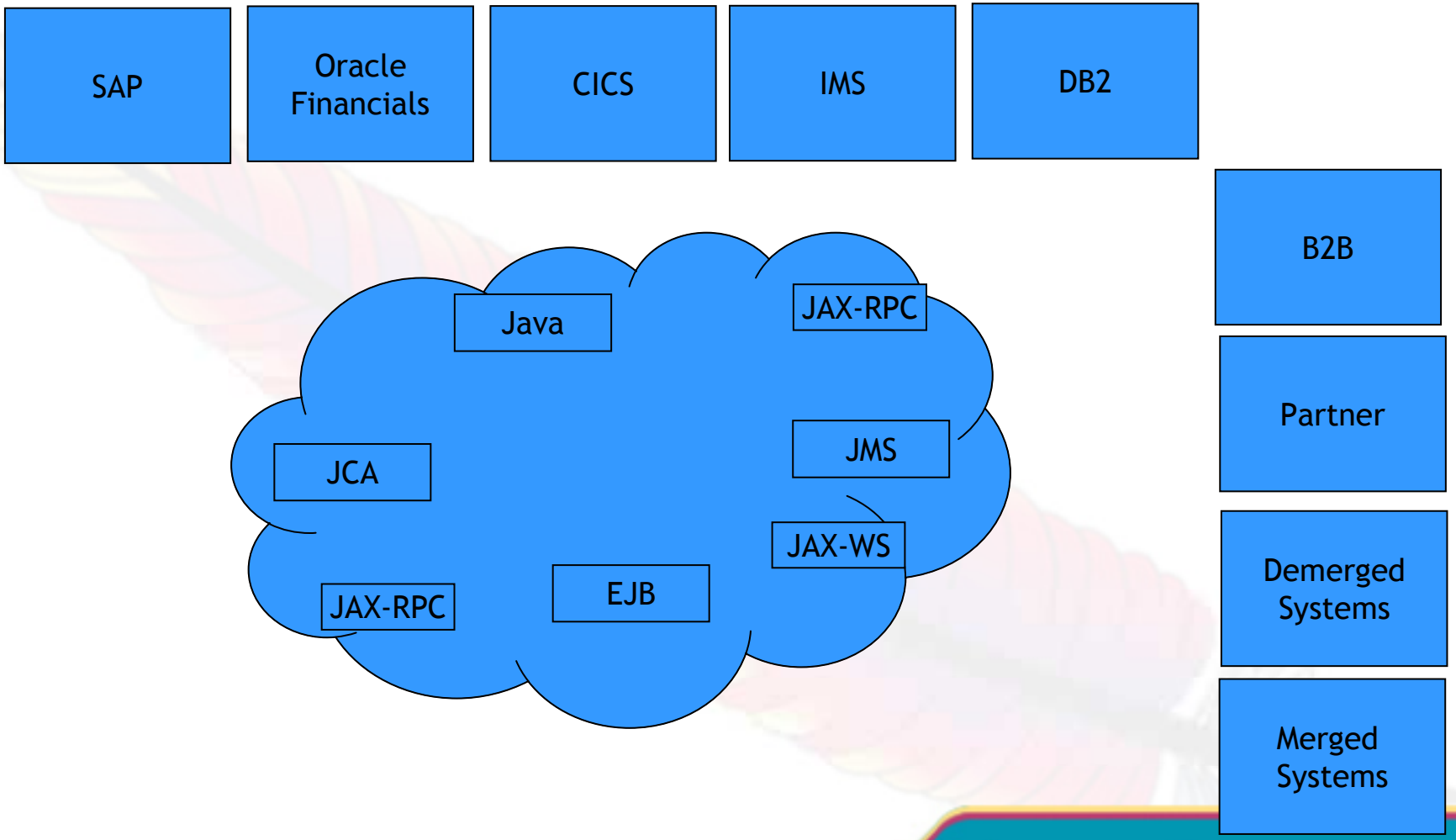
Service Oriented Architecture



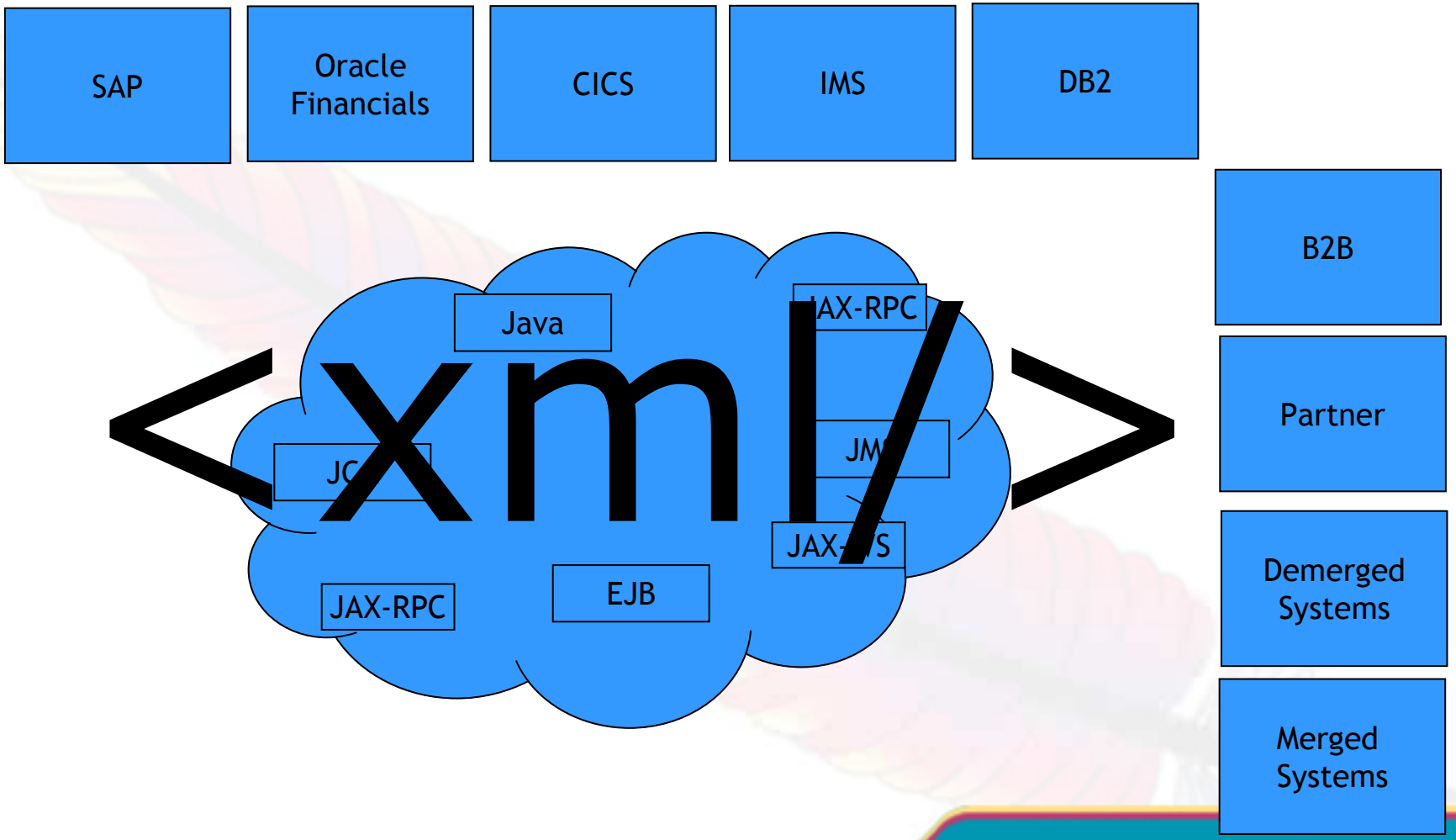
The ESB pattern?



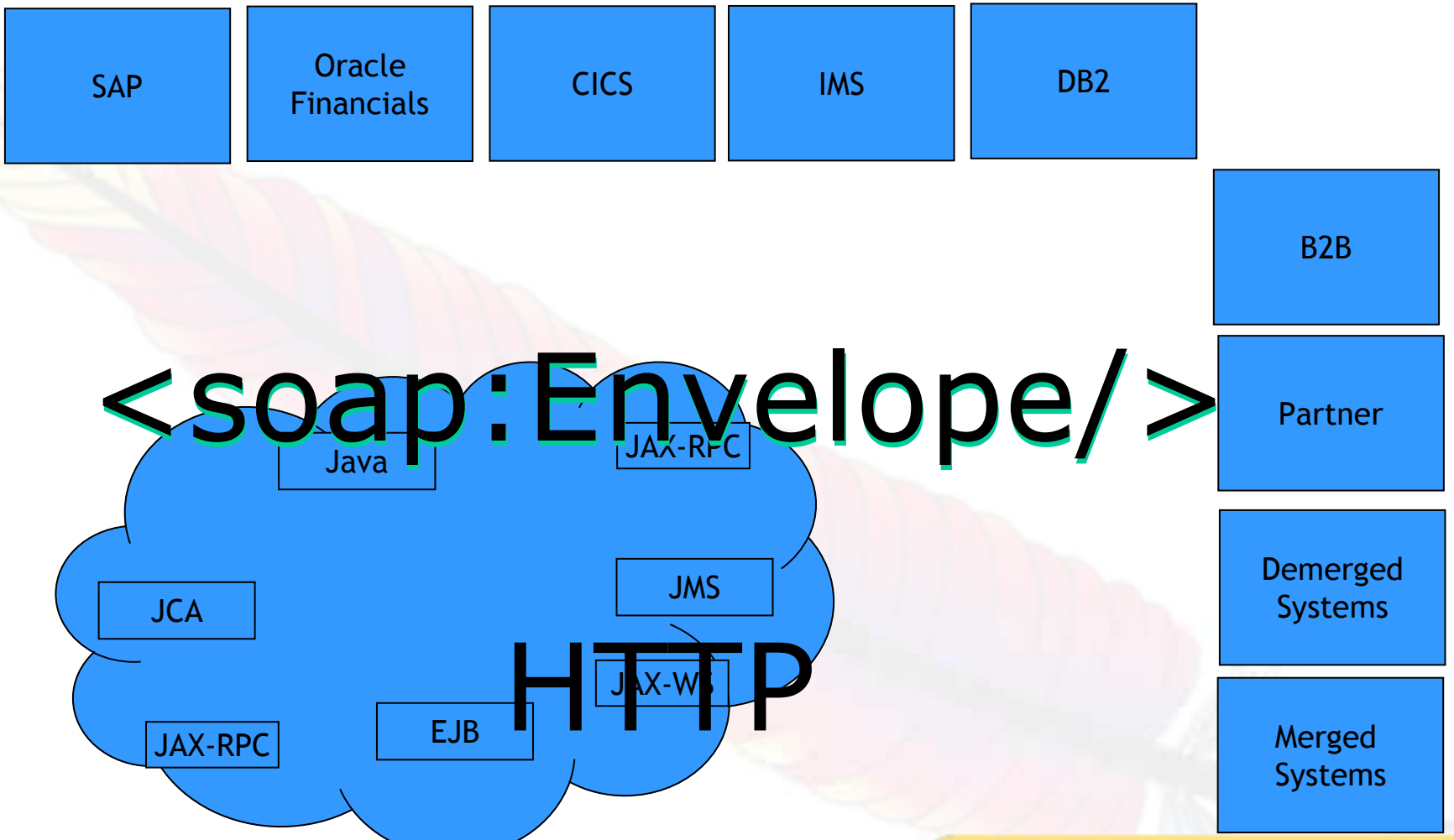
The problem space



The only common data model



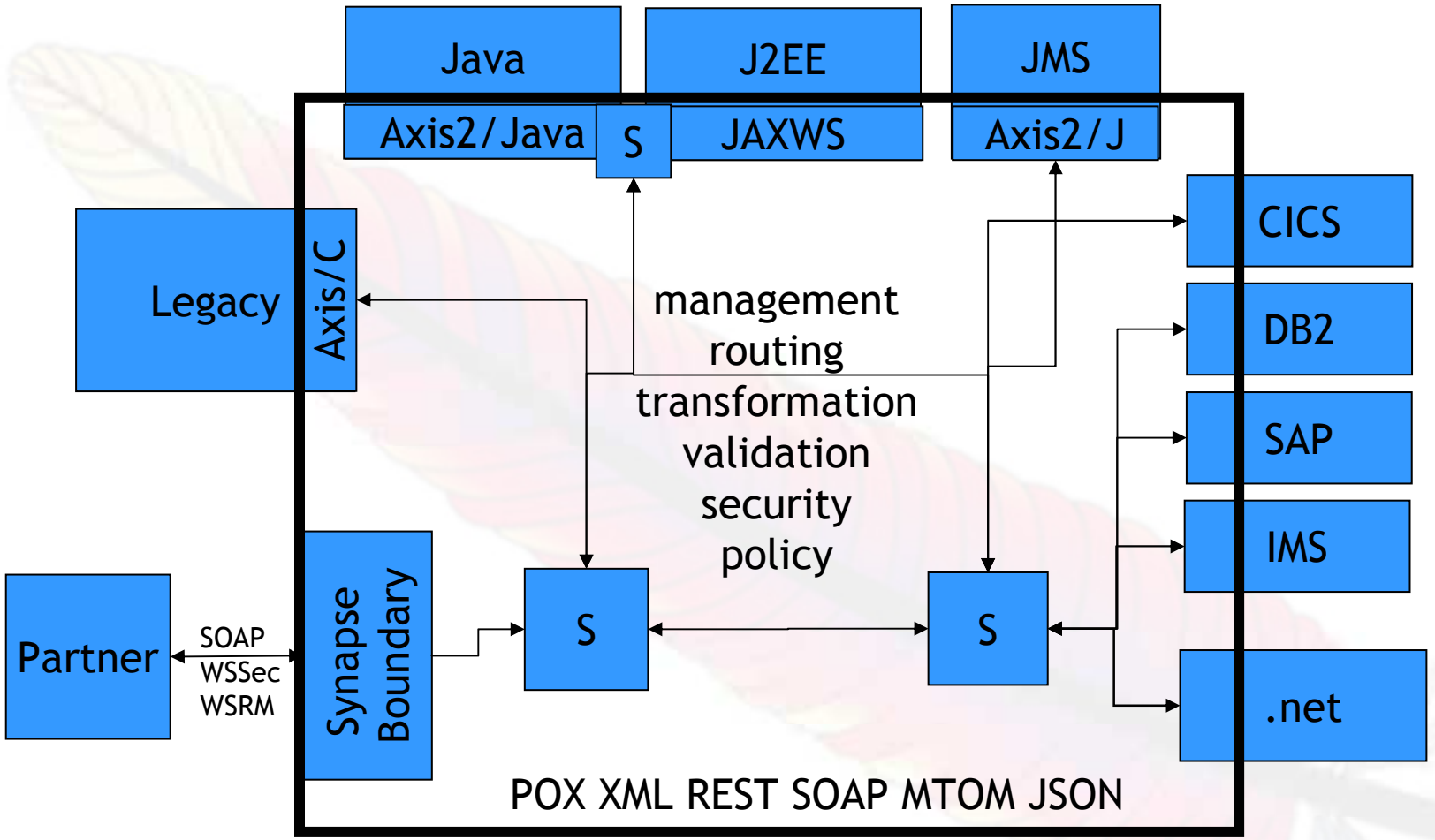
Common interaction models



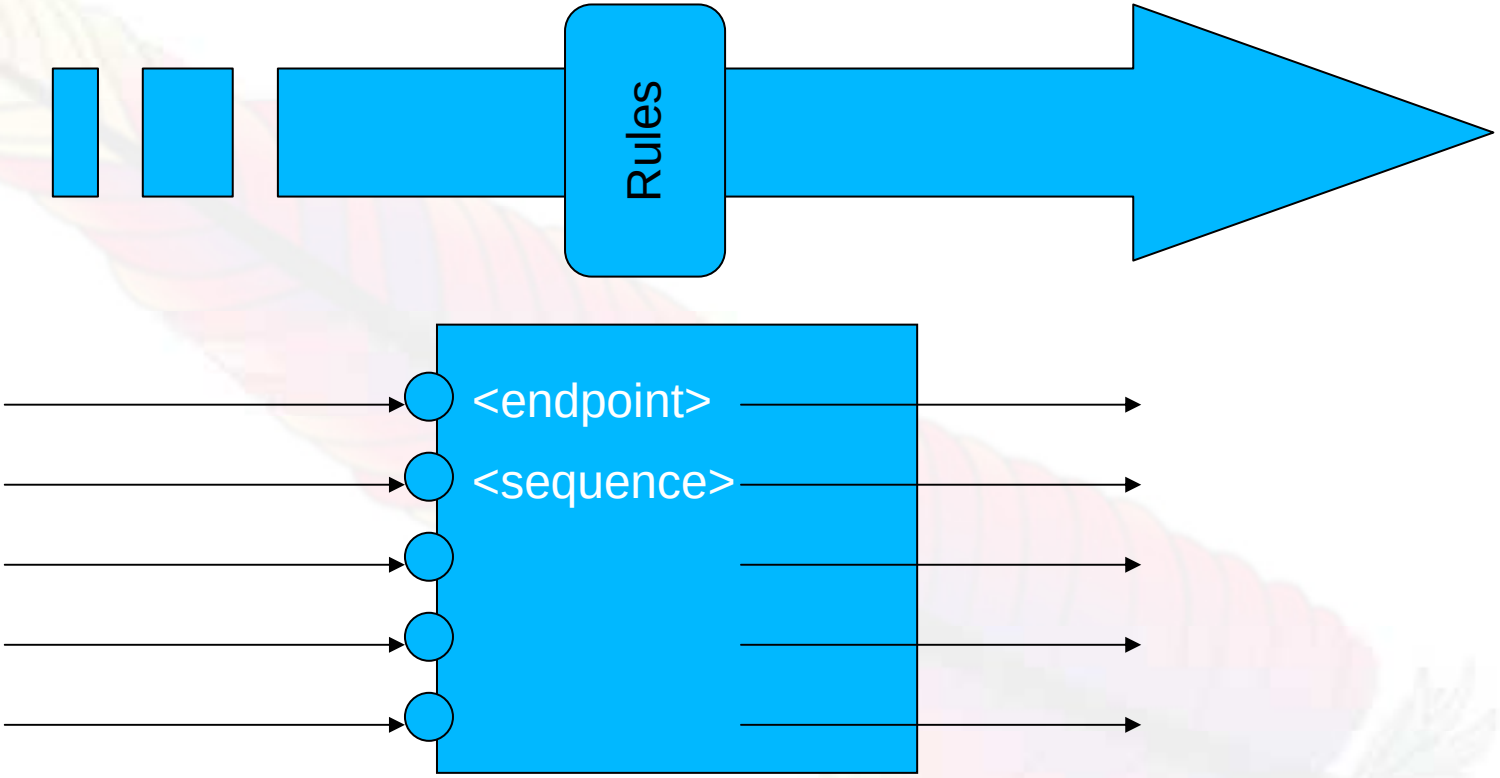
Web Services model

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

The Synapse pattern of an SOA



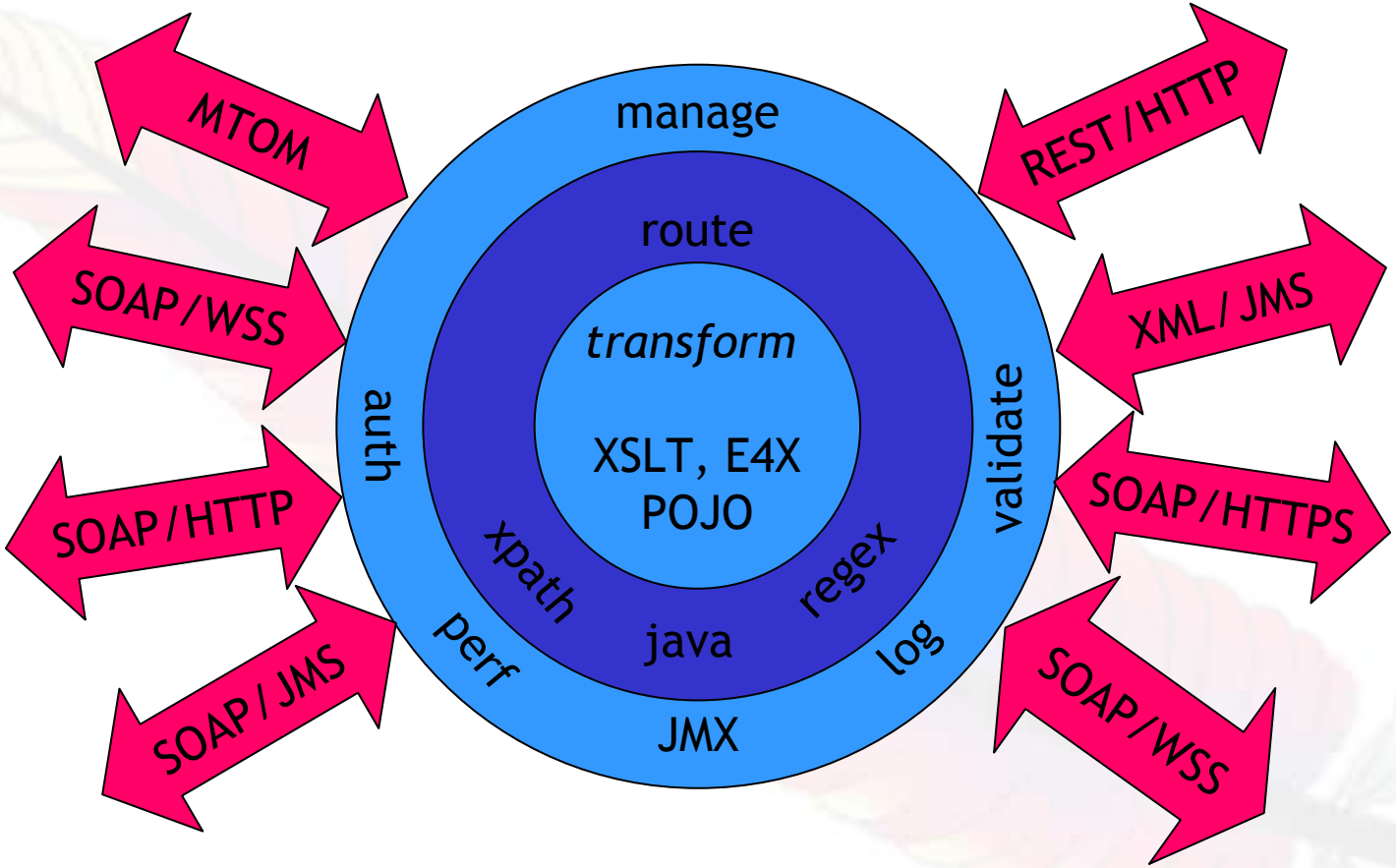
Two approaches



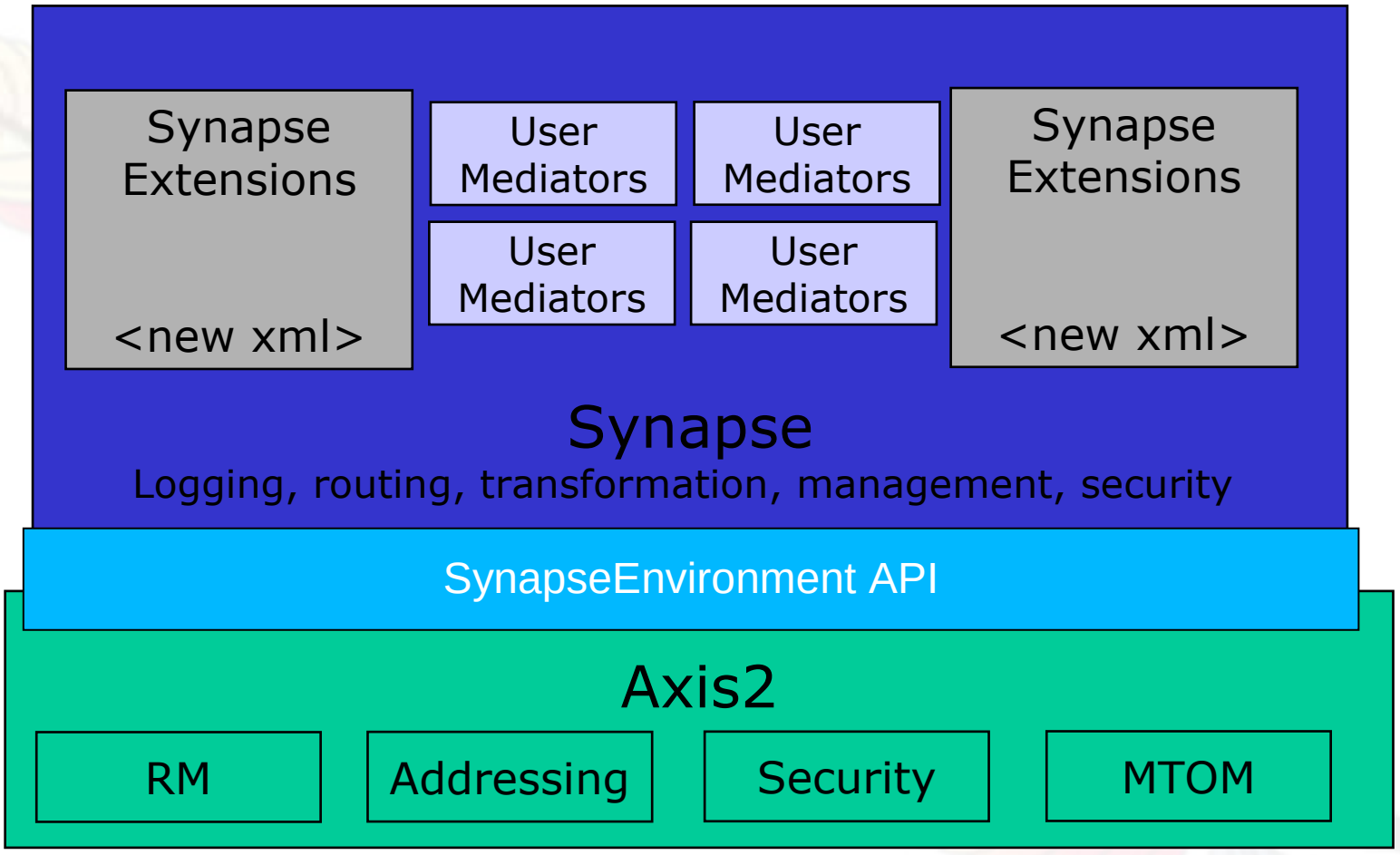
What do you need to do?

- Connect
- Manage
- Transform

A Web services mediation engine…



A lightweight infrastructure

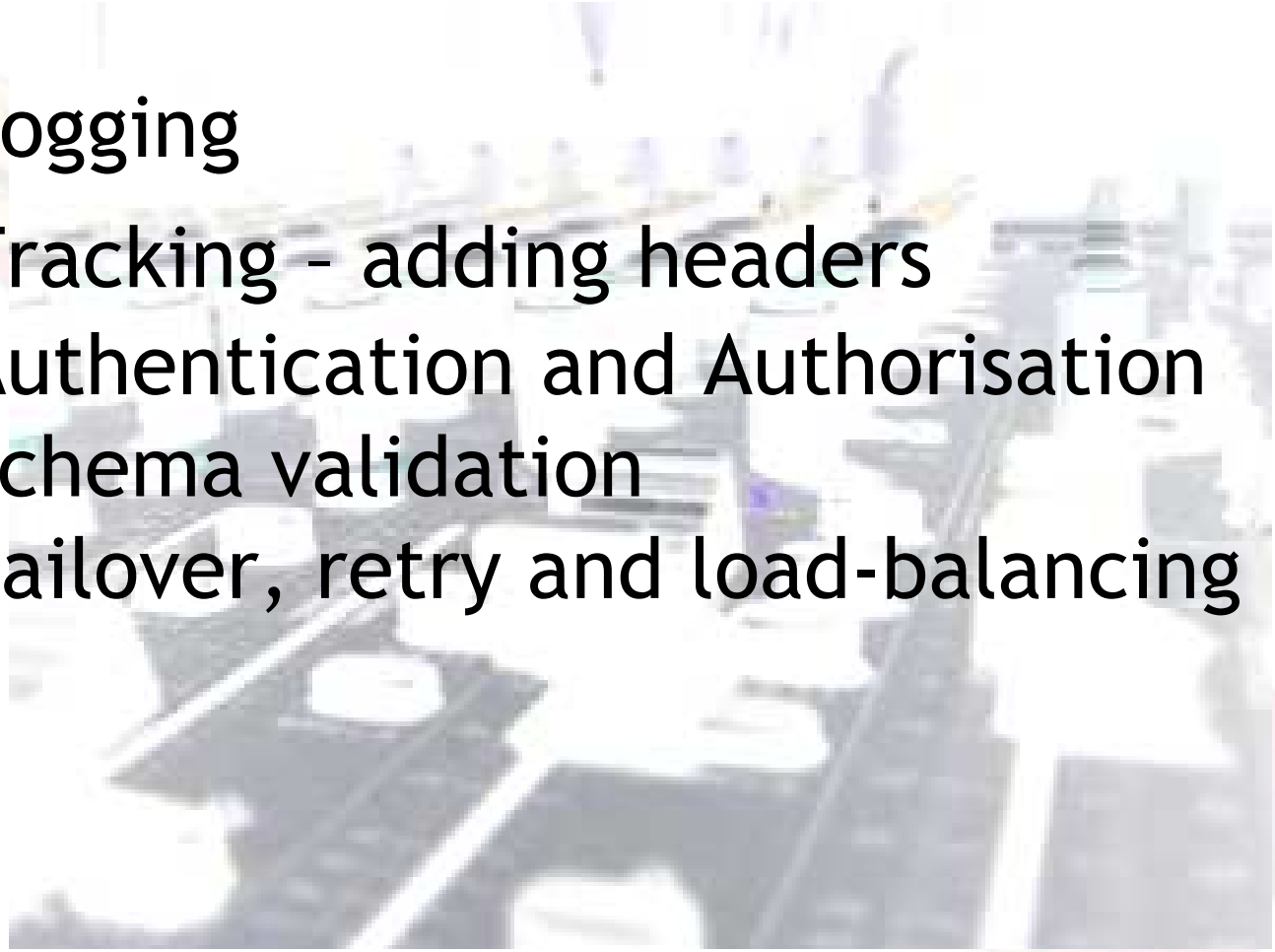


Connect

- Route messages
  - Based on XPath, Regex, etc
- Deal with mismatch
  - Initiate/Terminate RM, WS-Sec
- Switch
  - POX or REST to SOAP to JSON
  - JMS to HTTP to SMTP
- Virtualisation
  - Virtual URI to real URI mapping

Manage

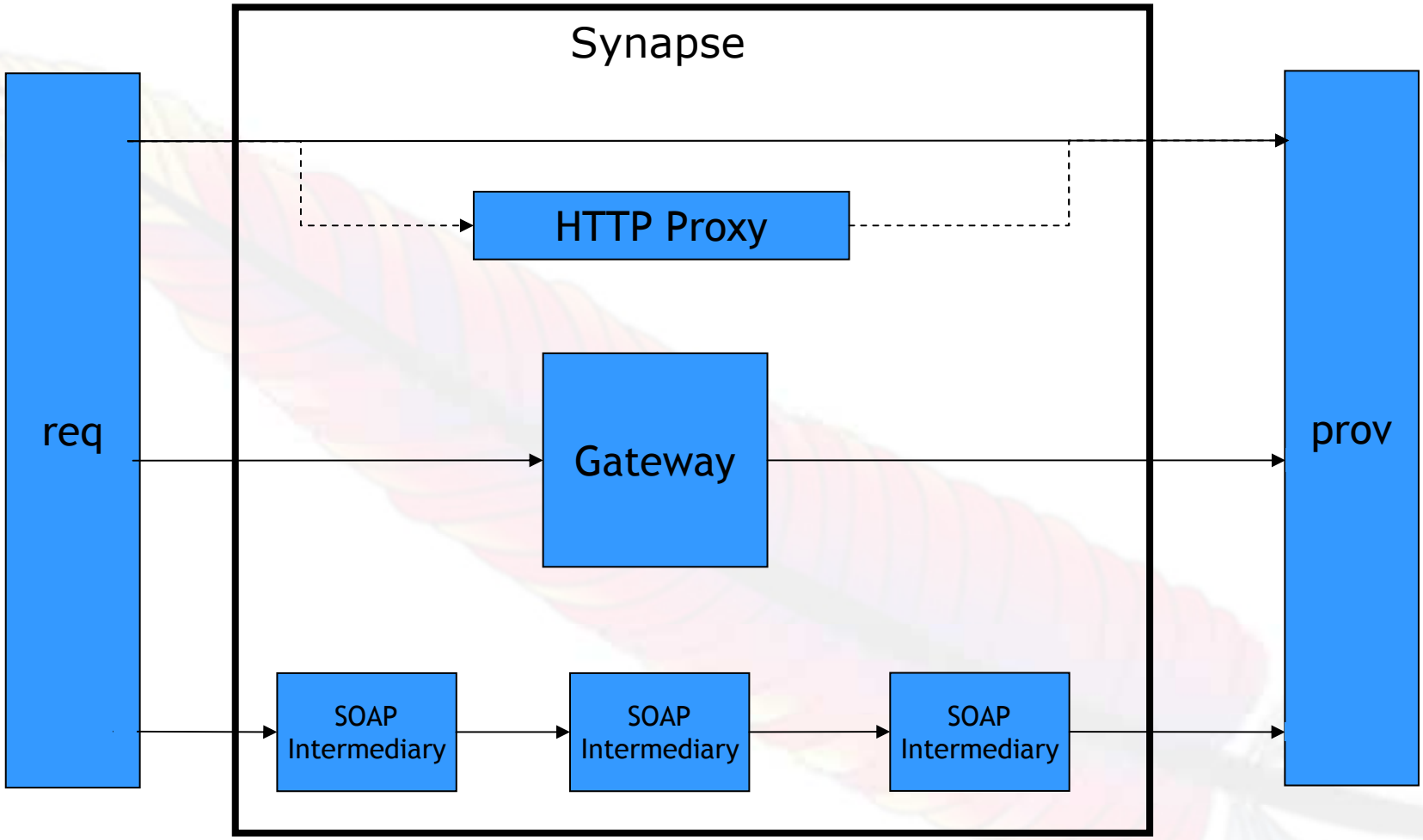
- Logging
- Tracking – adding headers
- Authentication and Authorisation
- Schema validation
- Failover, retry and load-balancing



Transform

- XSLT
- JavaScript/E4X
  - E4X is a simple mapping of XML *directly* into JavaScript
- POJO
- JSON ←→ XML

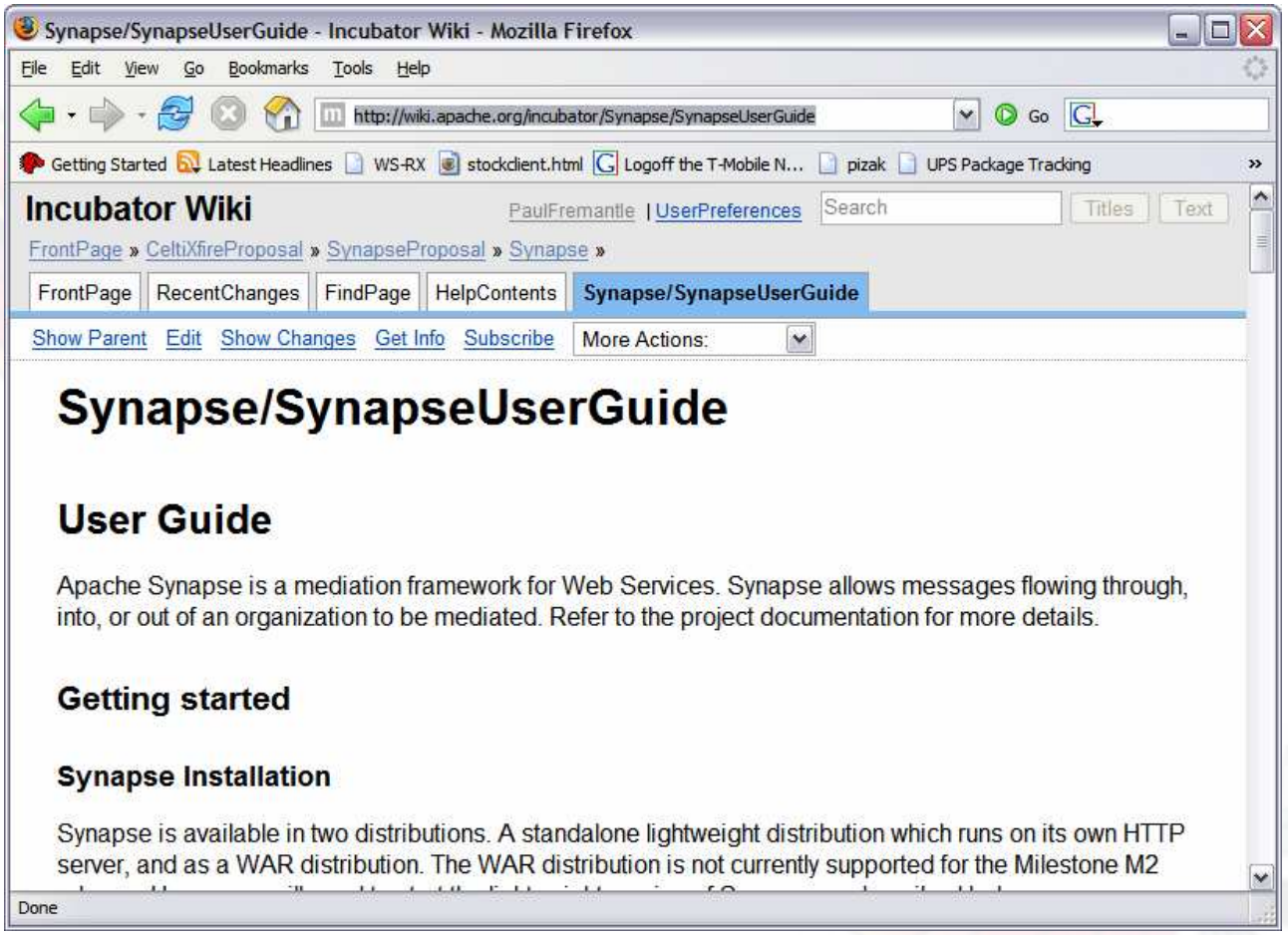
Deployment models



Deployment

- Transparent proxy
  - Configure the client to use Synapse as a proxy
    - Works with or without WS-Addressing headers
  - Or use a “smart client” and set <wsa:To>
  - *Pure rule based model*
- Gateway
  - New “Proxy Services”
    - Hosting virtual endpoints in Synapse
    - Either simply direct to an endpoint or via a sequence
    - Simple config of WSRM, WSSec
  - *Endpoint based model*

Synapse User Guide



First rule of the Synapse Programming Model

- ***"Don't use the Synapse Programming Model"***

- The aim of Synapse is to build in the common functions:
  - Log, route, modify headers, etc
  - Engage QoS: WSS, WSRM, WSA, etc
- And use XML models:
  - XPath, XSLT, etc
- Configure the routing and transformation using a simple XML file

Synapse Config Language

```
<synapse>
  <definitions>
      <sequence>...</sequence>
        <endpoint>...</endpoint>

        <property>...</property>


        ...
  <definitions>
  <rules>
    mediators
  </rules>
</synapse>
```

Sequences

- A chain of actions to take on a given message
  - The actions may be conditional
  - Each sequence can have a name and be re-used

Example

```
<sequence name="stockquote">
        <header name="To"
         value="http://ws.invesbot.com/stockquotes.asmx"/>

        <filter xpath="//*[wsx:symbol='MSFT']"
                xmlns:wsx="http://www.webserviceX.NET/">
                <makefault>
            <code value="tns:Receiver"
                        xmlns:tns="..."/>
            <reason
                                value="Isn't there a Windows API
for that?"/>
                 </makefault>
      </filter>

     <send/>
</sequence>
```

Endpoints

A simple local representation of a remote endpoint

```
<endpoint
  name="invesbot"
   address=

   "http://ws.invesbot.com/stockquotes.as
   mx"
```

Useful mediators

- send
  - Send on
- drop
  - Ditch the message
- log
  - log message
- makefault
  - Send a fault on or back
- transform
  - XSLT
- header
  - Set a header in the message
- filter
  - Do this if the XPath or Regex matches
- class
  - Call a user mediator
- validate
  - XSD validation
- switch
  - Conditional processing

Synapse Extensions

- Packaged as a JAR
- Uses dynamic discovery to register new XML elements into the configuration model
- e.g.
  - <spr:springmediator…>
    - Allows users to use Spring to wire up a mediator
  - <synxsl:xslt xsl="url_to_xslt"/>
    - Transforms messages using XSLT
- Extensions use the same programming model as User mediators + XML-based factory

User Mediators

- Simple Java class that implements the Mediator interface:

  boolean mediate(MessageContext mc);

  true= continue processing

  false= halt processing

Programming model (cont)

- MessageContext
- A representation of a SOAP infoset
  - Routing model
    - get/setTo, From, ReplyTo, FaultTo, MessageId etc
  - get/setEnvelope()
    - AXIOM representation
  - isResponse(), isMTOM(), isRest()
  - get/setProperties()

Status

- Incubator in Apache since August
  - All Apache Contributed code, core committer team, aiming to graduate
- M1 release – January 2006
  - Core function, HTTP Proxy support, XSLT, XPath
- M2 release – June 2006
  - Clean up of config language
- 0.90 release – *soon! (we always say that)*
  - Proxy services
  - Repackage as a MAR
  - JavaScript/E4X support

Currently looking to Graduate from incubation

Getting involved

HomeWiki / Home site

- http://wiki.apache.org/ws/Synapse
- http://incubator.apache.org/synapse/

SVN

- https://svn.apache.org/repos/asf/incubator/synapse/trunk/java

Try it out

- http://incubator.apache.org/synapse/download/M2/download.cgi
- Submit bug reports
  - http://issues.apache.org/jira/browse/SYNAPSE

Join us

synapse-dev@ws.apache.org

Future

- 1.0 Focus:
  - Support for WS-RM, WS-Sec
  - Fault handling
  - Support for remote registries
  - Documentation, Usability
    - We need your help to try it out and tell us what works
- The core model is also amenable to a Synapse/C implementation

Questions