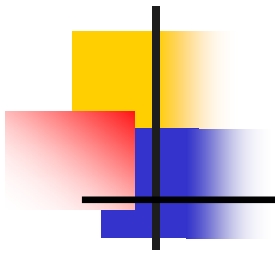
Building an Enterprise Service Bus Using Web Services and Apache Synapse v2

Paul Fremantle

VP of Technology

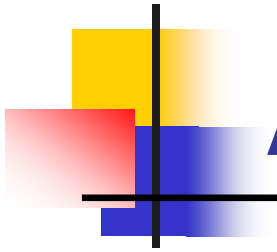
WSO2

paul@wso2.com



About Me

- Co-founder and VP of Technology and Partnerships at WSO2
  - The leading Open Source Web services company
- Co-Chair of the OASIS WSRX Technical Committee
- Committer on Apache Synapse
- Member of the Apache Software Foundation
- Co-author of *Building Web Services in Java 2nd Edition*
- http://bloglines.com/blog/paulfremantle
- Ex IBM Senior Technical Staff Member
  - developed the IBM Web Services Gateway
  - Apache WSIF, Apache Axis C/C++, JWSDL/WSDL4J Apache Woden



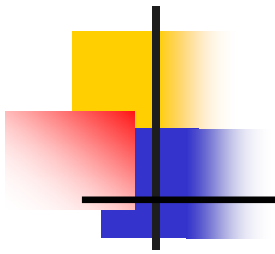
Apache Synapse

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.

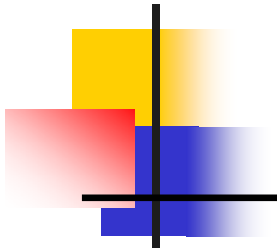
While incubation status is not necessarily a reflection of the completeness or stability of the code, it does indicate that the project has yet to be fully endorsed by the ASF.

In other words… an incubator project
for the moment

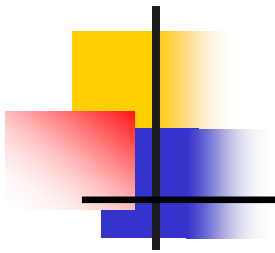


Contents

- What is an ESB?
- A Web services based ESB
- Connect, Manage, Transform
- Apache Synapse
  - Structure and model
  - Deployment
  - Configuration
  - Programming model
- Examples and demo

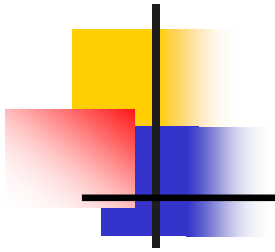


What is an ESB?



A Common ESB Definition

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

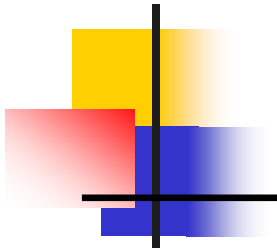


ESB Definition

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

which means

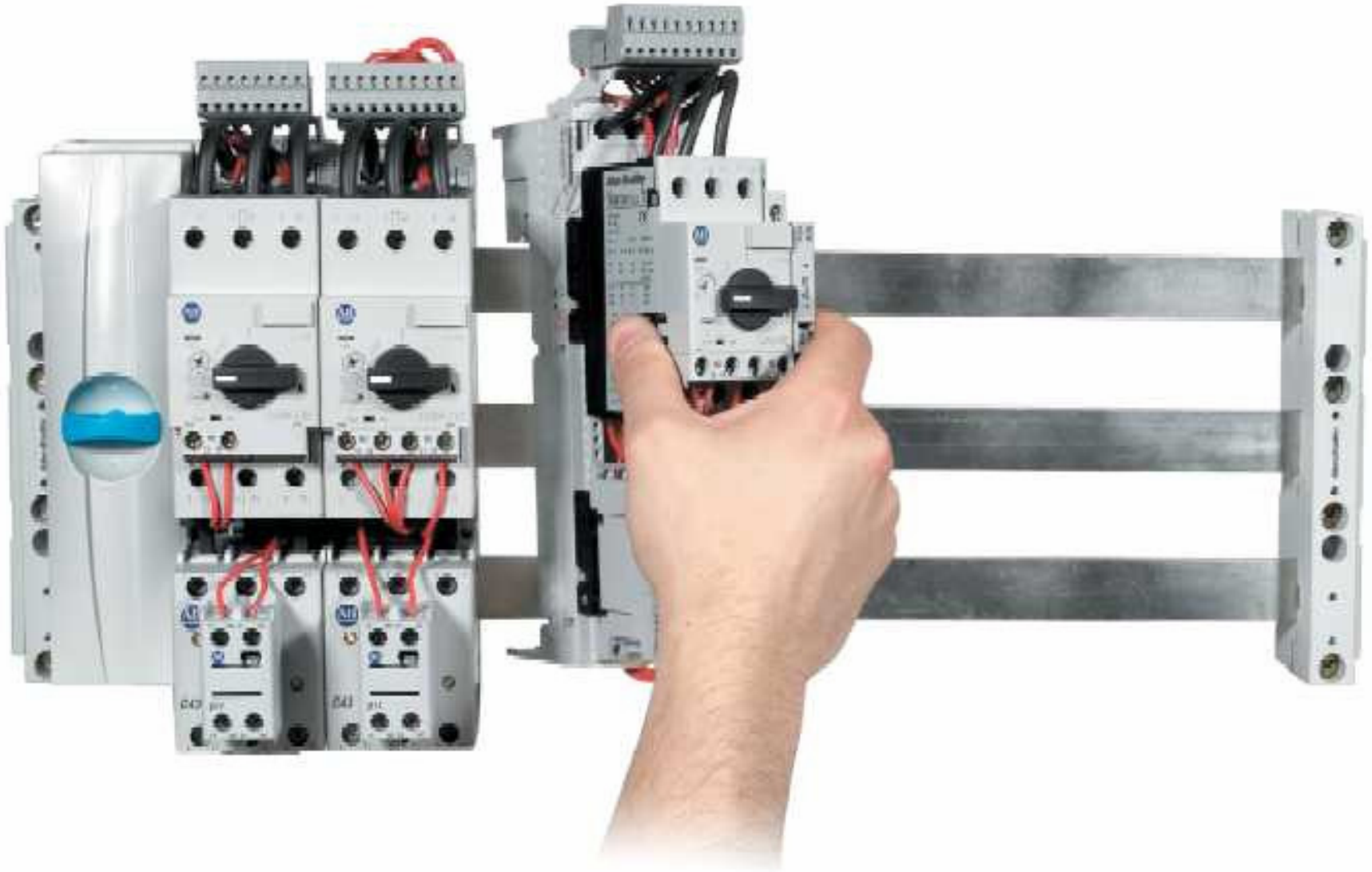
- Our existing middleware re-branded as an SOA platform, with some new web services adapters at the edges



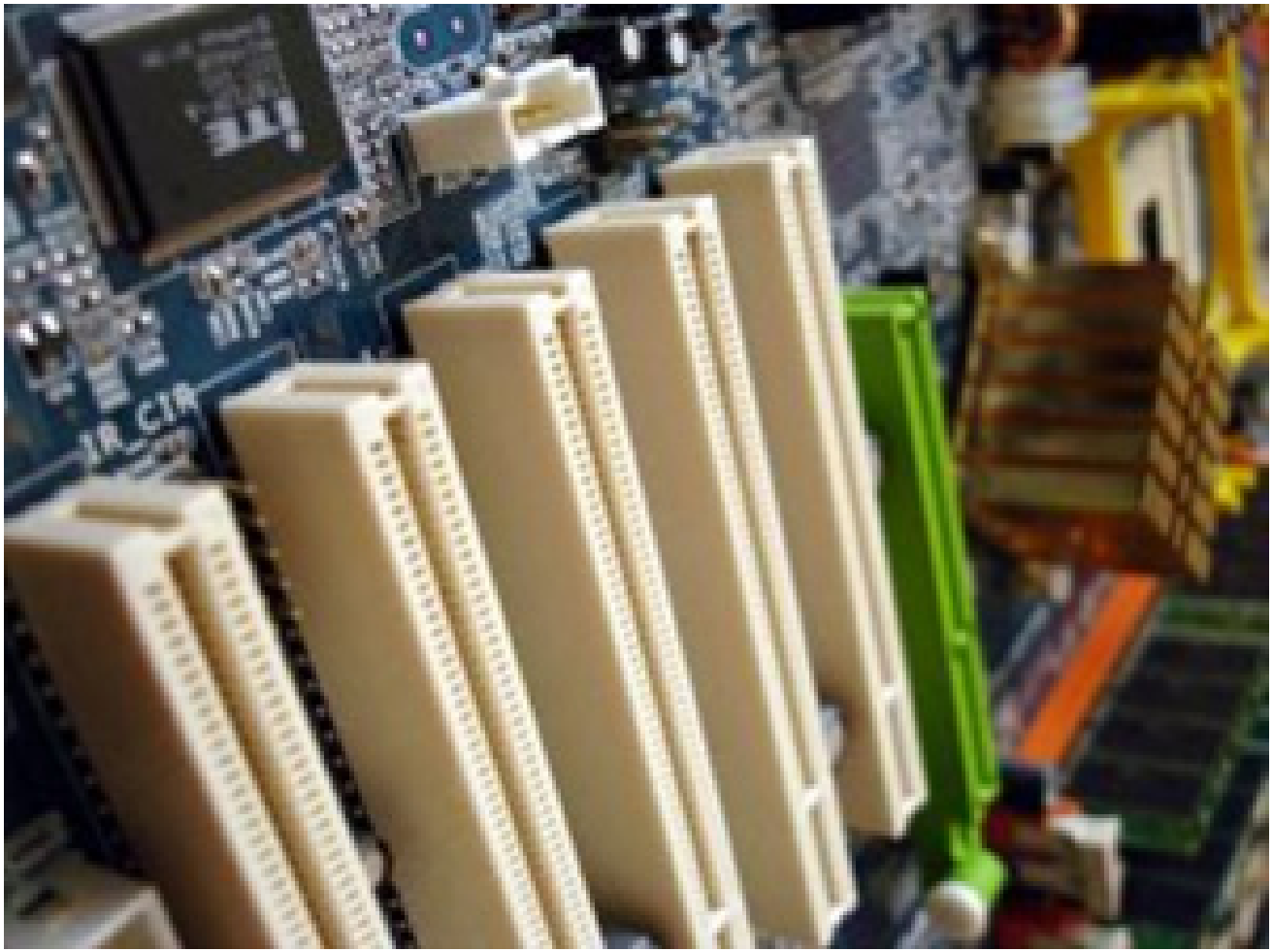
So what *is* an Enterprise Service Bus?

A Bus Bus

Busbar



Bus



"The Original ESB"

What Is Good about ESBs?

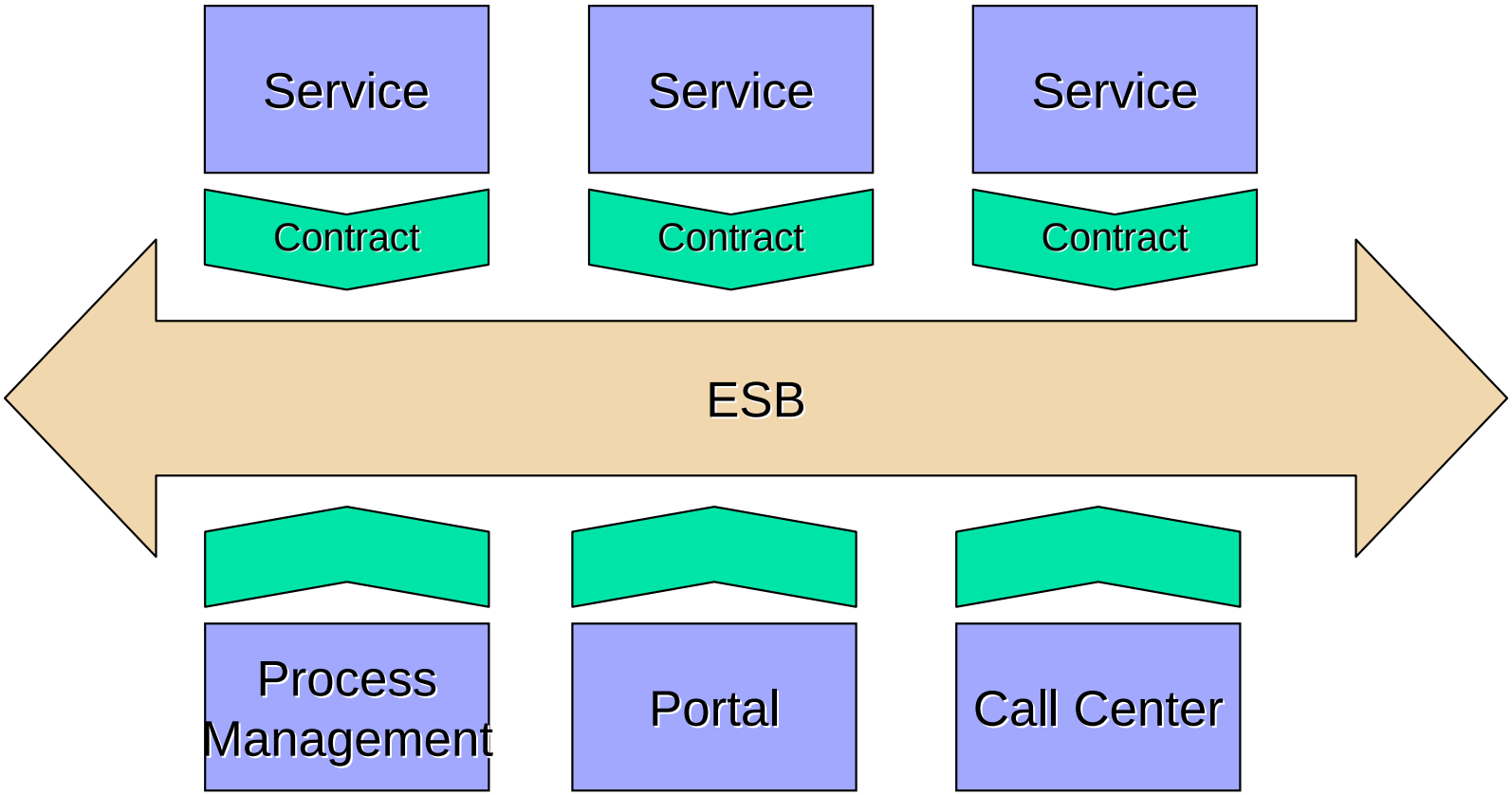
- SOA based integration
  - More than just message passing
  - Metadata about the *contracts* involved
    - *e.g.* WSDL, Schema, Policy
- Bus model is very powerful
  - Every message goes into the “bus” and ends up in the right place

So Why Comments Like This:

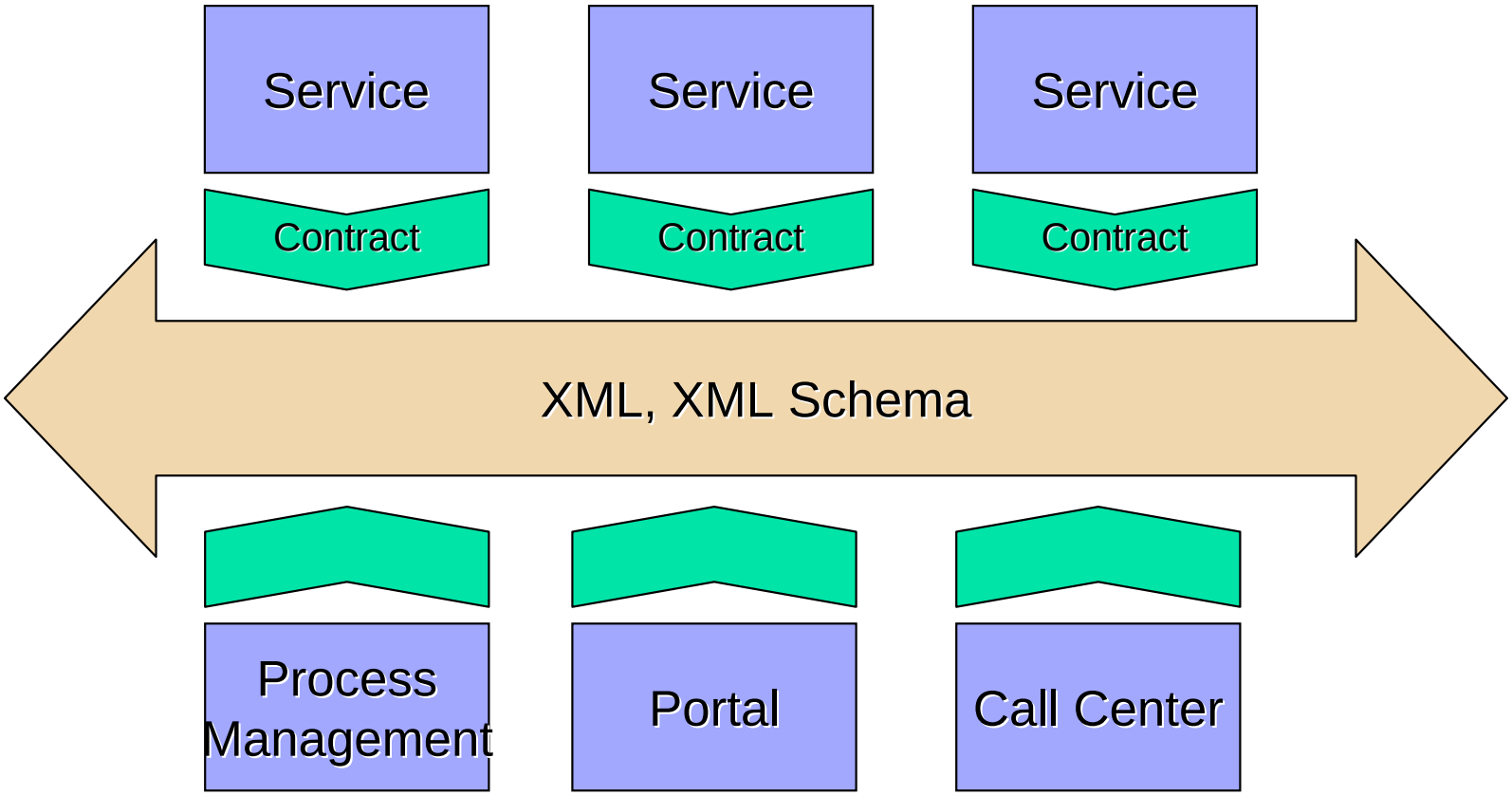
- Anne Thomas Manes:
  - "I do not believe that an ESB is an essential component of SOA. When it comes right down to it, you don't need any new products to do SOA. After all, SOA is about design, not technology."
  - "For the most part, 'ESB' is a marketing term."
  - "But by 2003, a bunch of other vendors hijacked the term and applied it to their own integration products."

Are ESBs a new lock-in point?

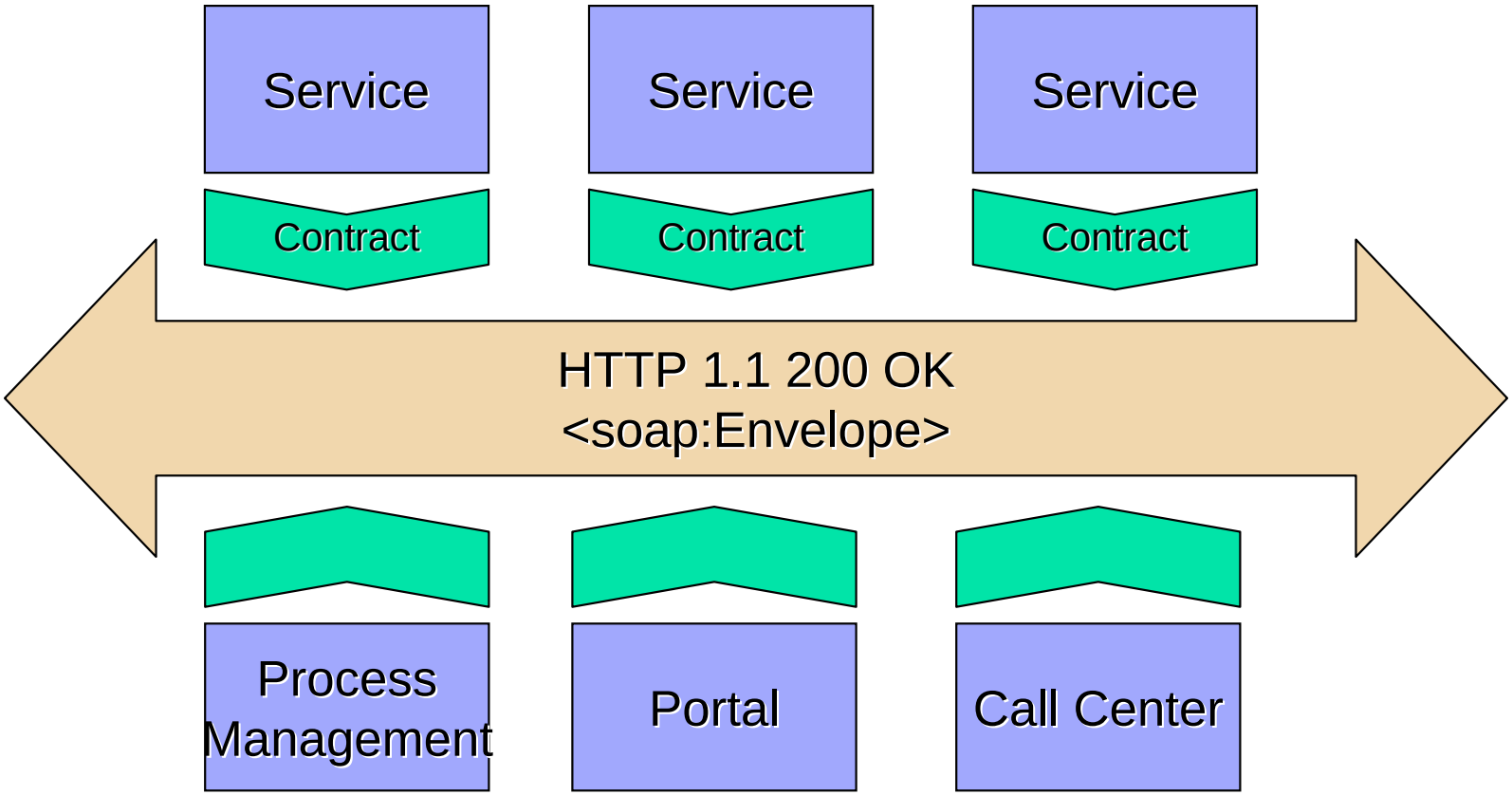
Service Oriented Architecture



The Only Common Data Model



The Only Common Interaction Models



What Do You Need for Enterprise Integration?

- Connectivity
- Data formats
- Security
- Reliability
- Routing
- Metadata
- Management

What Do You Need for Enterprise Integration?

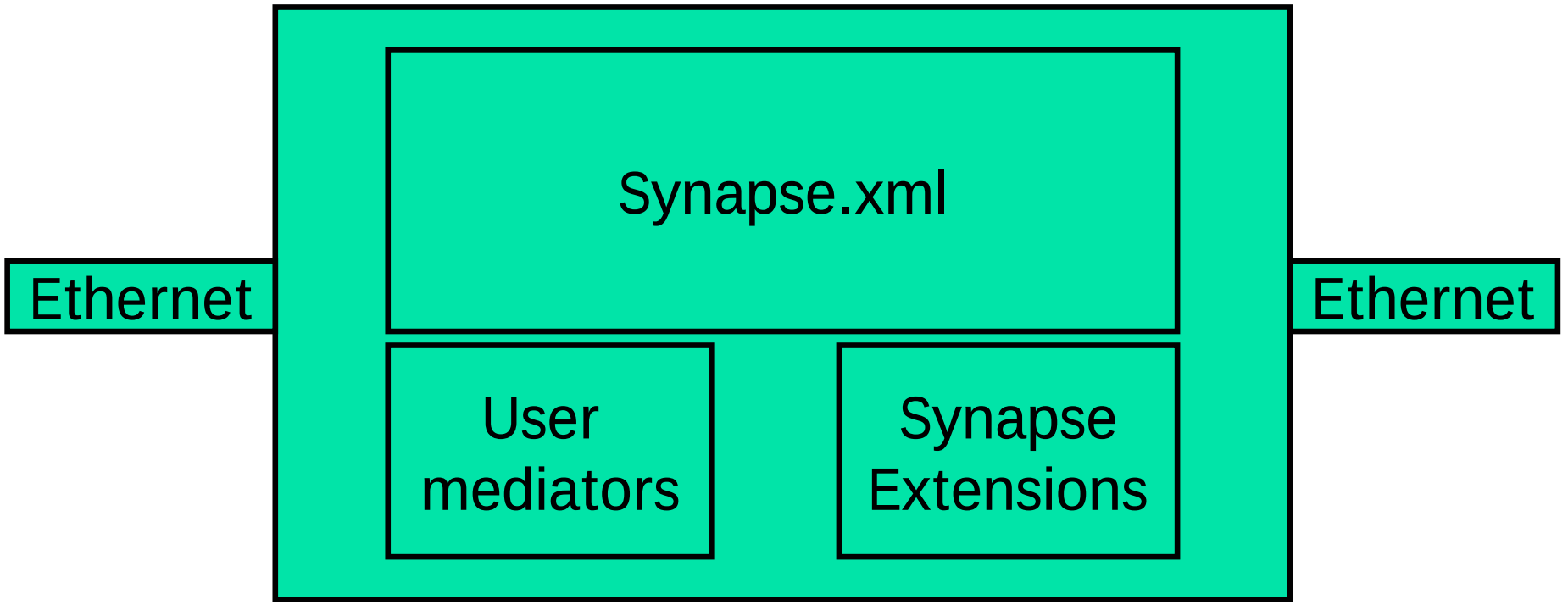
- Connectivity
  - Get data and messages from A to B
- Data formats
  - Understand what shape the data is in
- Security
  - Ensure end-to-end confidentiality, authentication, integrity
- Reliability
  - Ensure message delivery
- Routing
  - Decouple systems from the physical topology
- Metadata
  - Understand what systems are there and what they offer and use

What Do You Get from Web Services?

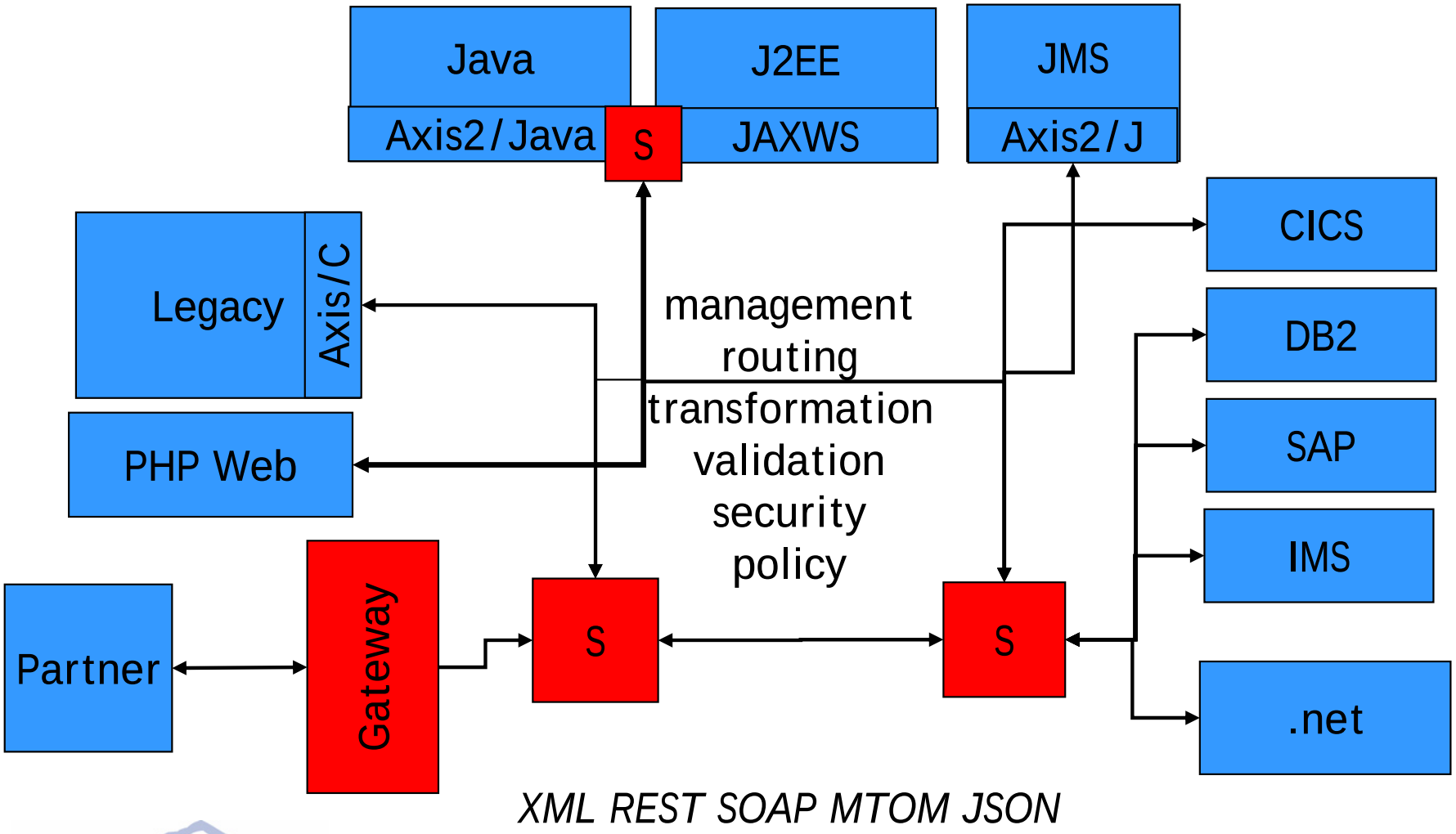
- Connectivity
  - Lightweight universal connectivity
  - *HTTP, REST, SOAP, XOP/MTOM (binary data)*
- Data formats
  - Common data model and data description language
  - *XML, XML Schema*
- Security
  - Encryption, Authentication, Single Signon
  - *WS-Security, WS-Trust, WS-SecureConversation*
- Reliability
  - Exactly once in-order delivery
  - *WS-ReliableMessaging*
- Routing
  - To/From/ReplyTo headers
  - *WS-Addressing*
- Metadata
  - Functional and non-functional description, capabilities and requirements
  - WSDL, WS-Policy, XML Schema

So What Is Synapse?

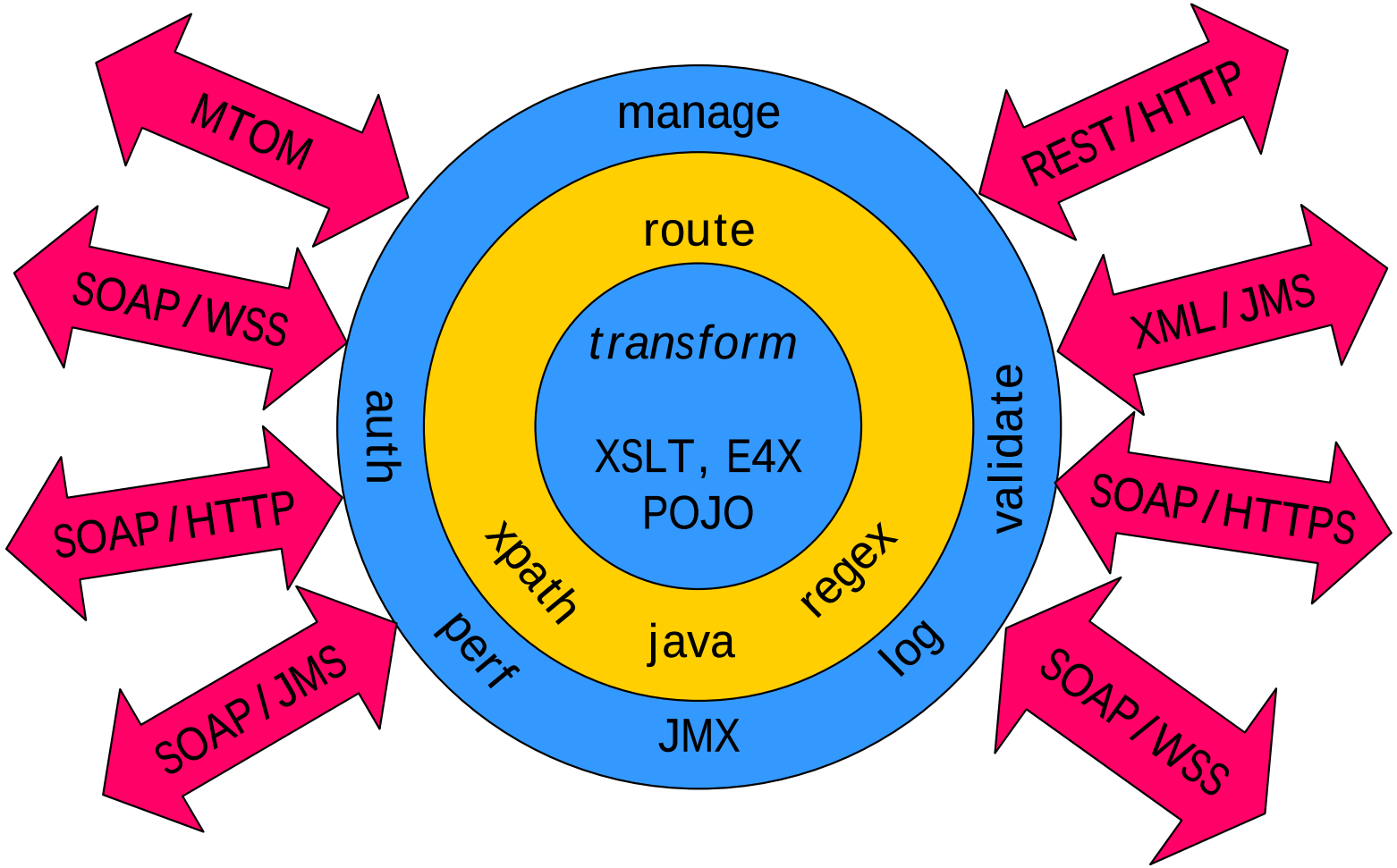
- A Web Service intermediary

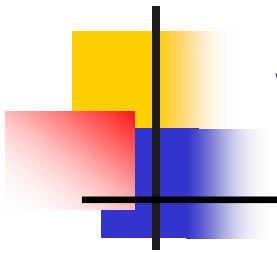


The Synapse Pattern of an ESB



A Web Services Mediation Engine





What Does Synapse Do?

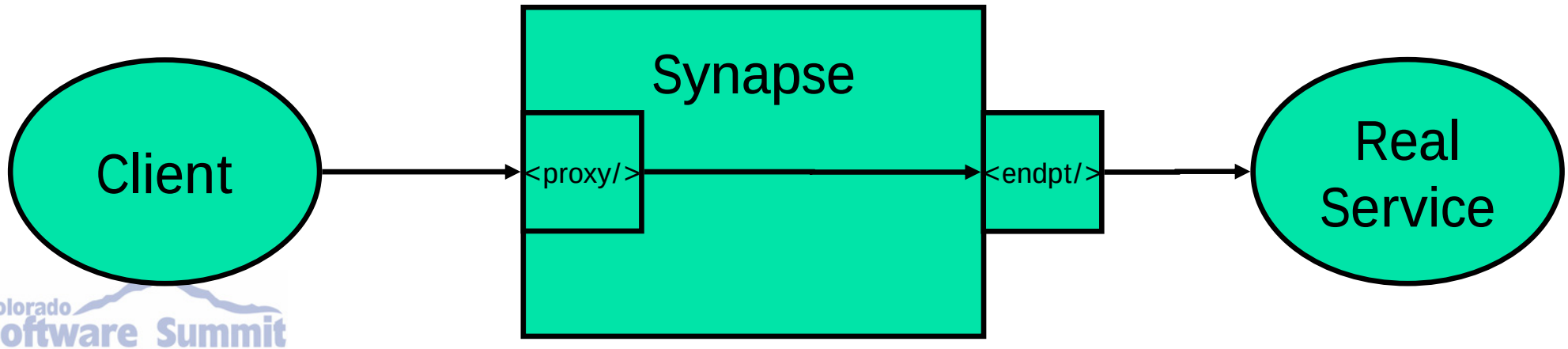
- Connect
- Manage
- Transform

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

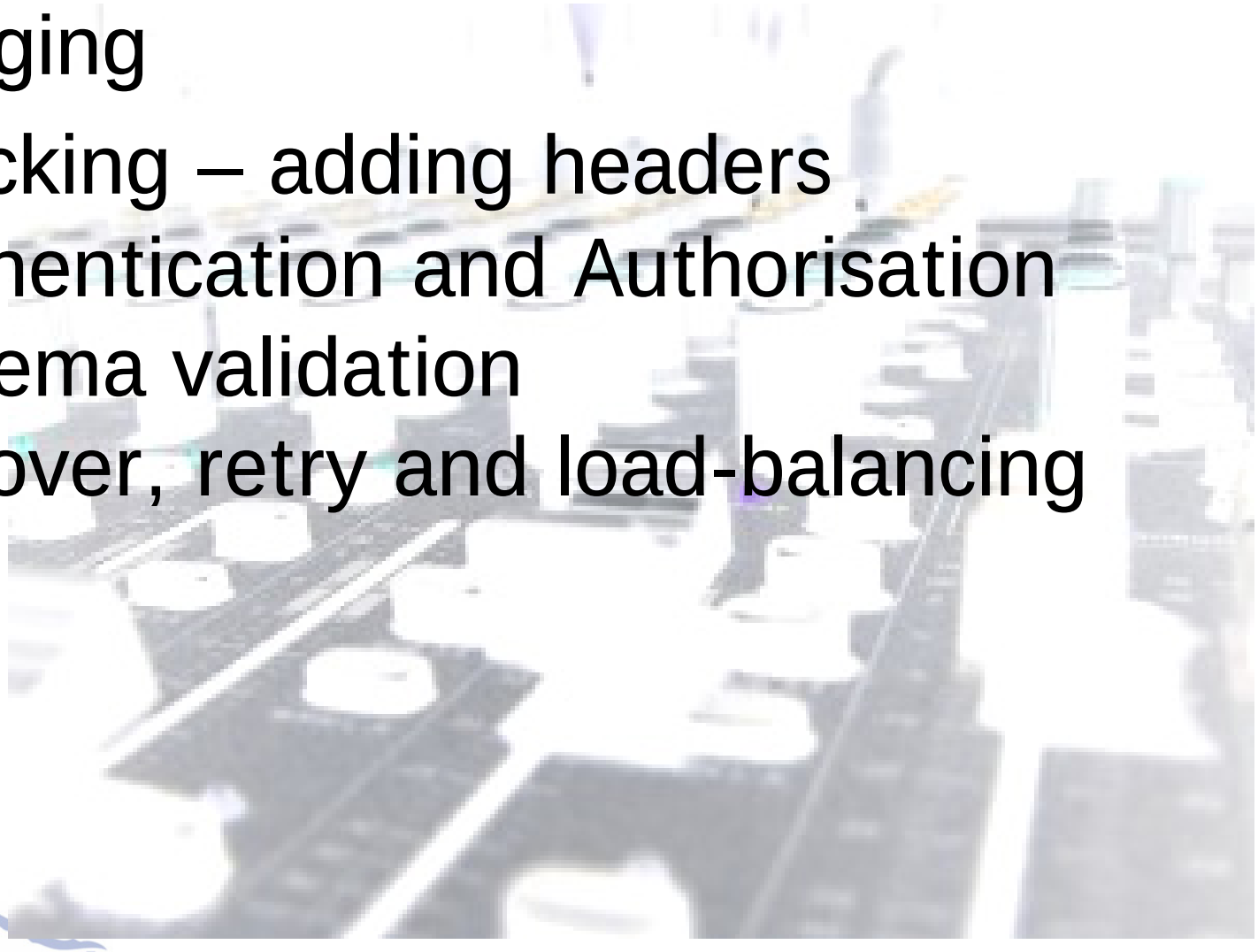
Connect Example

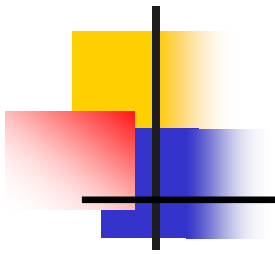
```
<endpoint name="realService"
   address="http://wso2.com/axis2/services/myRealService">
</endpoint>
<proxy name="myVirtualService" description="virtual
   endpoint" transports="all">
        <target endpoint="realService"/>
</proxy>
```



Manage

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





Manage Examples

<log level="full"/>

Logs the whole message

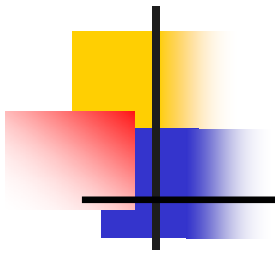
<header name="wso2:logging"
value="Logged"/>

Adds a SOAP header

<wso2:logging>Logged</wso2:logging>

Transform

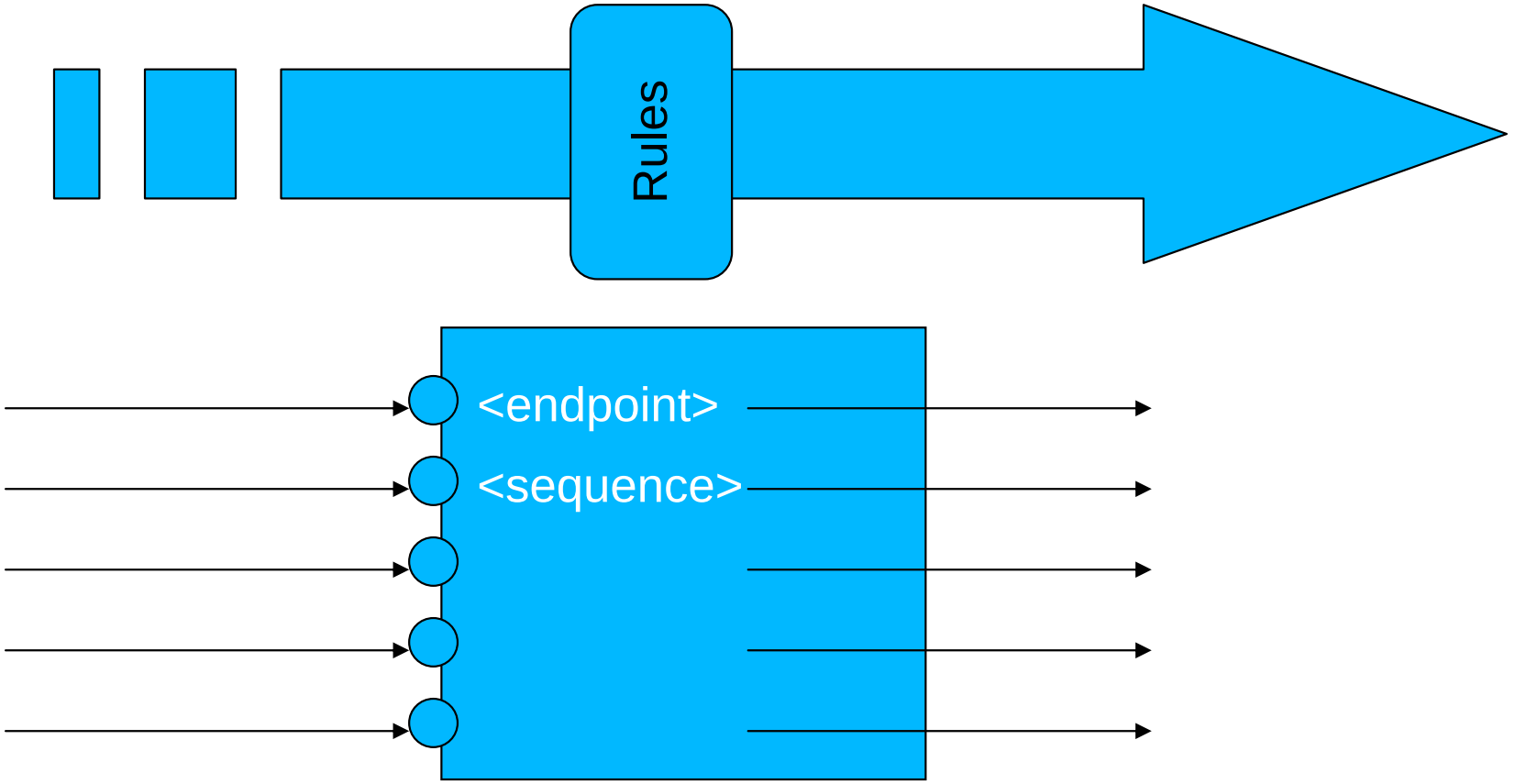
- XSL Transforms
  - Apply an XSLT to the SOAP message or Body
- JavaScript/E4X
  - E4X is a simple mapping of XML *directly* into JavaScript
- POJO
  - Write Java logic that manipulates the message
- JSON ←→ XML
  - Take in JSON/HTTP and send out XML or SOAP



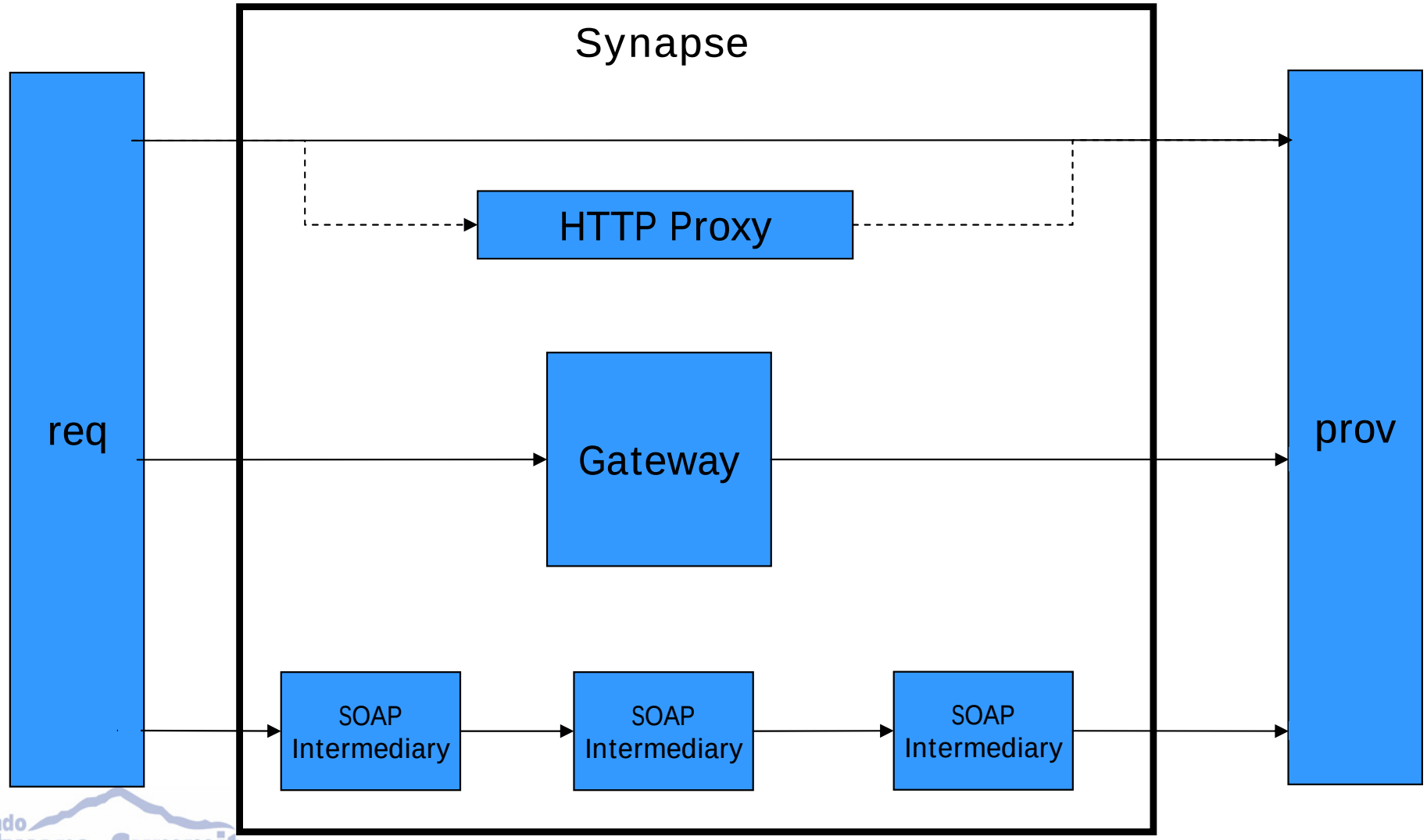
Transform Example

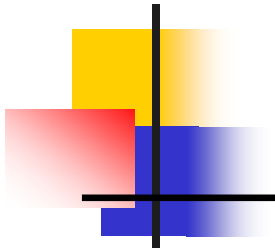
```
<javascript>
<![CDATA[
   function mediate(x) {
        x..*::price[0] *= 2; // double the price
        return x;
   }
]]>
</javascript>
```

Synapse's Two Approaches



Deployment Models





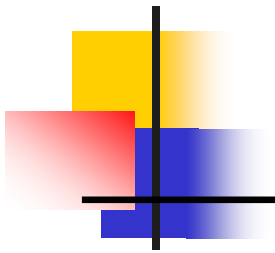
Deployment

- 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*

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

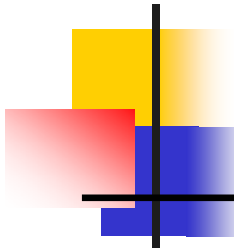
- A chain of actions to take on a given message
  - The actions may be conditional
  - Each sequence can have a name and be re-used

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.asmx"/>
```

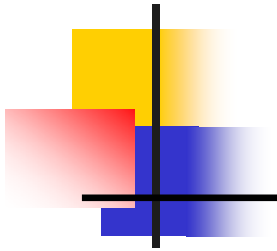
Allows characteristics such as policy, RM, Security to be configured

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
  - <js:javascript>...</js:javascript>
    - Allows user to use JavaScript and E4X to implement transforms
- Extensions use the same programming model as User mediators
  - Enhanced with an XML-based factory



User Mediators

- Simple Java class that implements the Mediator interface:

  boolean mediate(MessageContext mc);

  true= continue processing
  false= halt processing

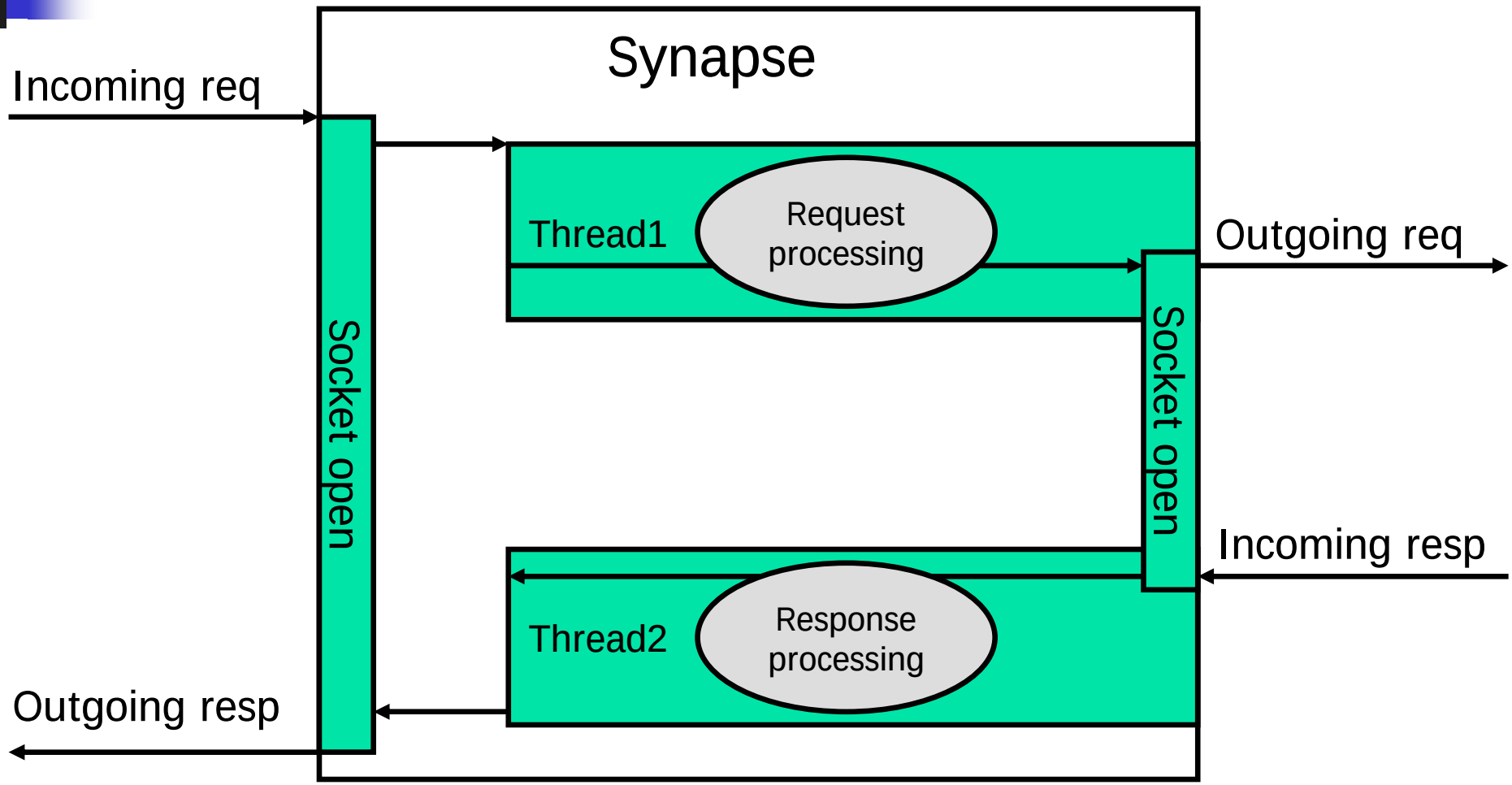
Programming Model *(Continued)*

- o.a.synapse.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()

Asynchronous Internals

- We are just working on a non-blocking IO model for Synapse
  - If full WS-Addressing in place then already asynchronous
    - Updating Synapse to have same behaviour even the client or server block on HTTP
  - Requires a NIO transport for Axis2
    - Since Axis2 does all the IO for Synapse
  - More important for an intermediary than an endpoint
    - Endpoints are usually doing something with the thread while the socket is open

Async IO Diagram

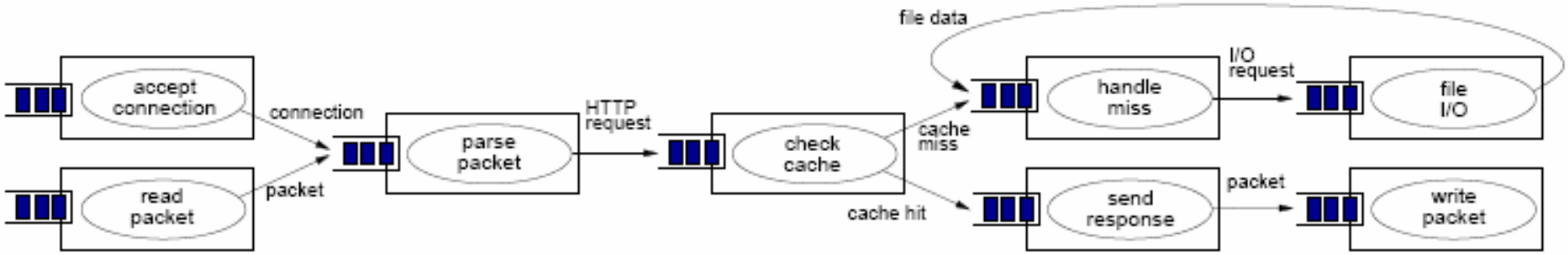


This model means:

1. Synapse threads never blocked during normal processing
2. Number of sockets open >> number of threads

Scalable Event Driven Architecture

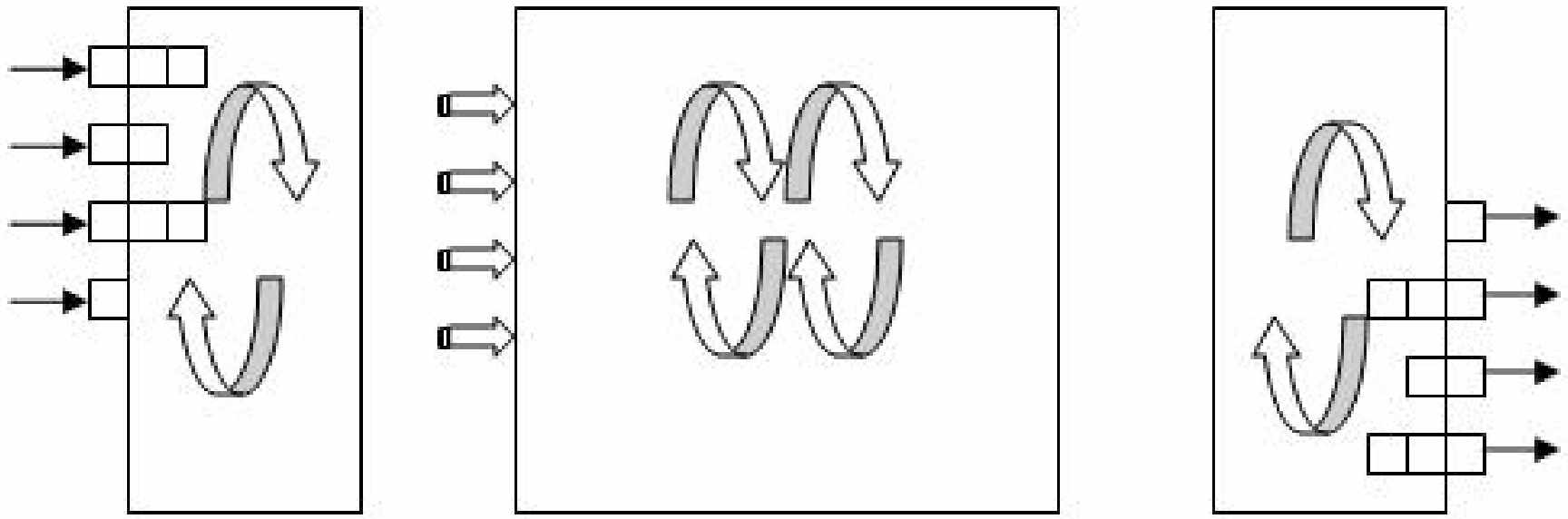
- Simple model of stages and queues for handling load
- Matt Welsh's PhD thesis



Colorado Software Summit

http://www.eecs.harvard.edu/~mdw/proj/seda/

NIO Model Is Effectively SEDA



NIO Listener with two dedicated threads

Synapse executing using its own thread pool

NIO Sender with two dedicated threads

"Registry" Support

- So far everything has been stored in a single XML file
  - Synapse.xml
- Hard to manage
- Also
  - Requirement to update the directives without restarting Synapse

Registry Model

- A registry is just a remote (or local) source of XML files
  - No special APIs or models from the Synapse perspective
- A pluggable model for reading from a registry
  - Synapse ships with a URL based implementation

```
<registry provider="o.a.s.r.URLRegistry"/>
```

Properties and Dynamic Properties

- Properties are just local configuration items
- Can be used to set up global configuration
- Dynamic properties are properties backed by a registry
- The values are cached and read from the registry
  - Allows sequences, endpoints and other config to be loaded dynamically

Dynamic Property example

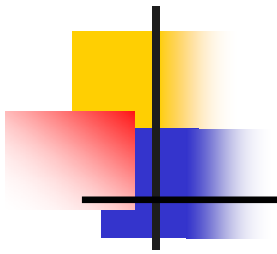
XML file stored at http://mystore.wso2.com/endpoint.xml

```
<endpoint name="remote"
        address="http://myserver.com"/>
```

synapse.xml

```
<!-- load registry values from URLs -->
<registry provider="URLRegistry"/>

<!-- associate with remote endpoint definition -->
<property name="myDynamicEP"
        key="http://mystore.wso2.com/endpoint.xml"/>
<proxy name="exposedService">
        <endpoint key="myDynamicEP"/>
</proxy>
```



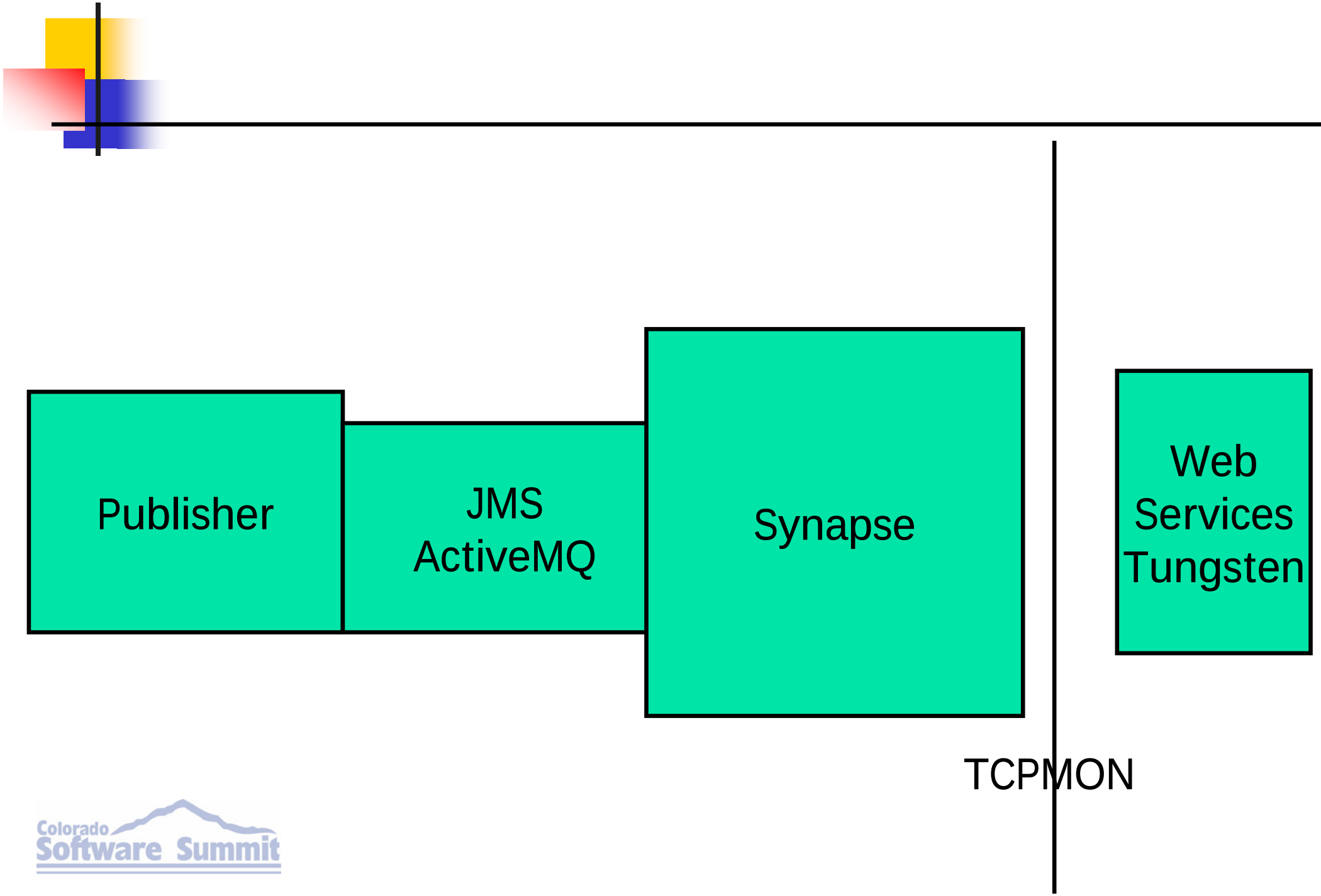
So What Do You Need?

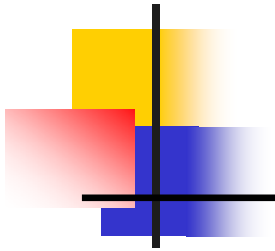
Anne Thomas Manes again:

- One or more service platforms (*endpoints*)
- A SOA management solution (*mediation*)
- A registry (*metadata*)
- An XML gateway if services will be exposed outside the firewall

Synapse can be both the XML Gateway and the SOA management solution

Demo





Status

- Incubator in Apache since August 2005
  - All Apache Contributed code, core committer team
- 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

Looking to Graduate from incubation when Apache gives us the go-ahead

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

Submit use cases, bugs, ideas, or even code (but actually we'd prefer the former)

Questions