Enterprise Service Bus

Oxford University

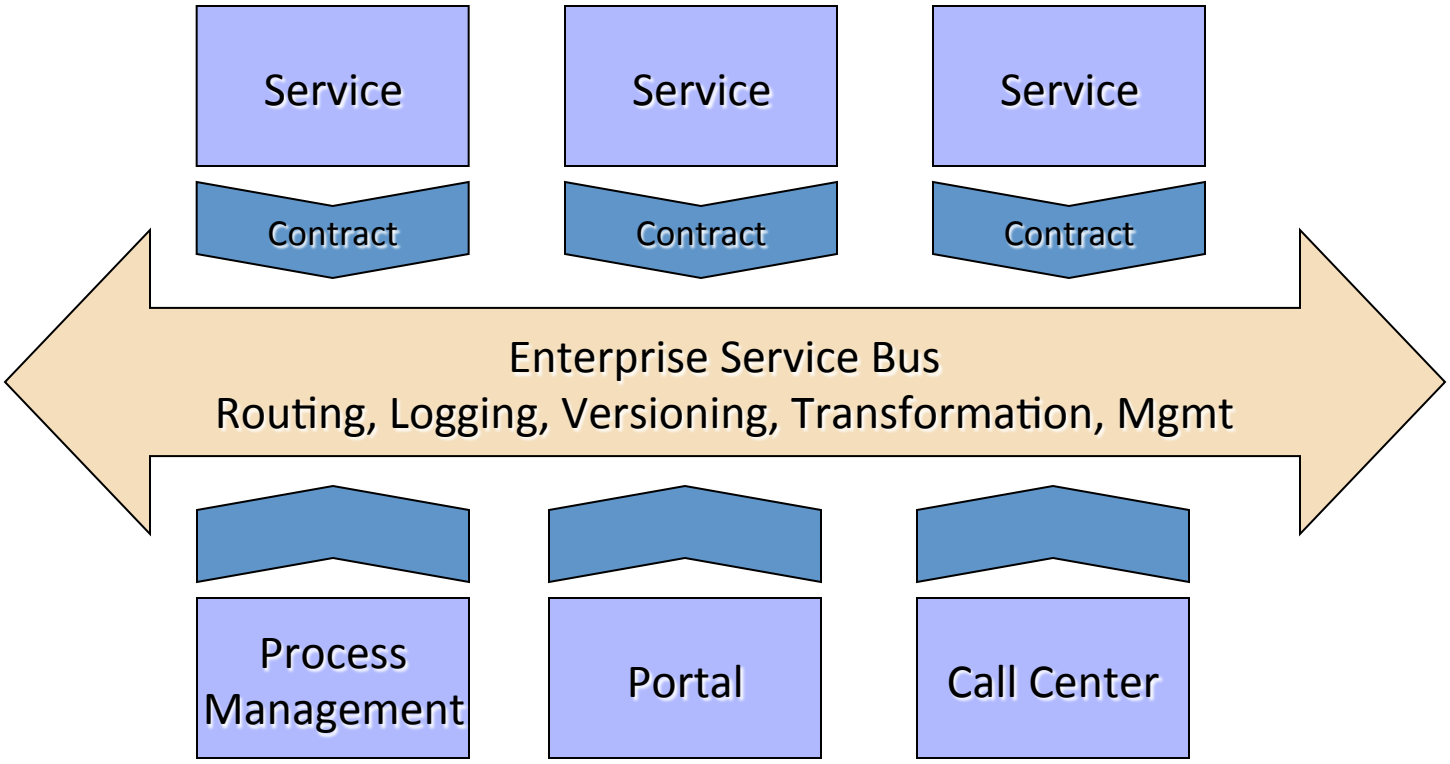
Software Engineering Programme

Dec 2014

Enterprise Service Bus (ESB)

- A software architecture
  - A logical intermediary through which every message flows
  - Offers a policy based approach to decide what to do to each message or interaction
- The benefits of the gateway model
  - Without a physical hub and spoke
- Many vendors offer ESB products
  - Often a layer over an existing messaging framework

ESB as the implementation of SOA



Different approaches

- Point to Point
- Traditional EAI
- ESB
- Event Driven Architecture

Pros and Cons of an ESB

Pros

- Faster and cheaper accommodation of existing systems
- Increased flexibility: easier to change as requirements change
- Standards-based
- Scales to enterprise wide deployment
- Configuration rather than coding
- No central broker

Cons

- May end up with a proprietary solution
  - no common standards for the overall config and policies yet
- Requires more hardware to run
- New skills to learn to configure ESB
- Hard to get ROI on a small number of projects

ESB options

- Proprietary
  - IBM, Oracle, Tibco, SAP
- Open Source
  - Mule, Fuse, WSO2
  - Apache ServiceMix, Apache Synapse, Apache Camel

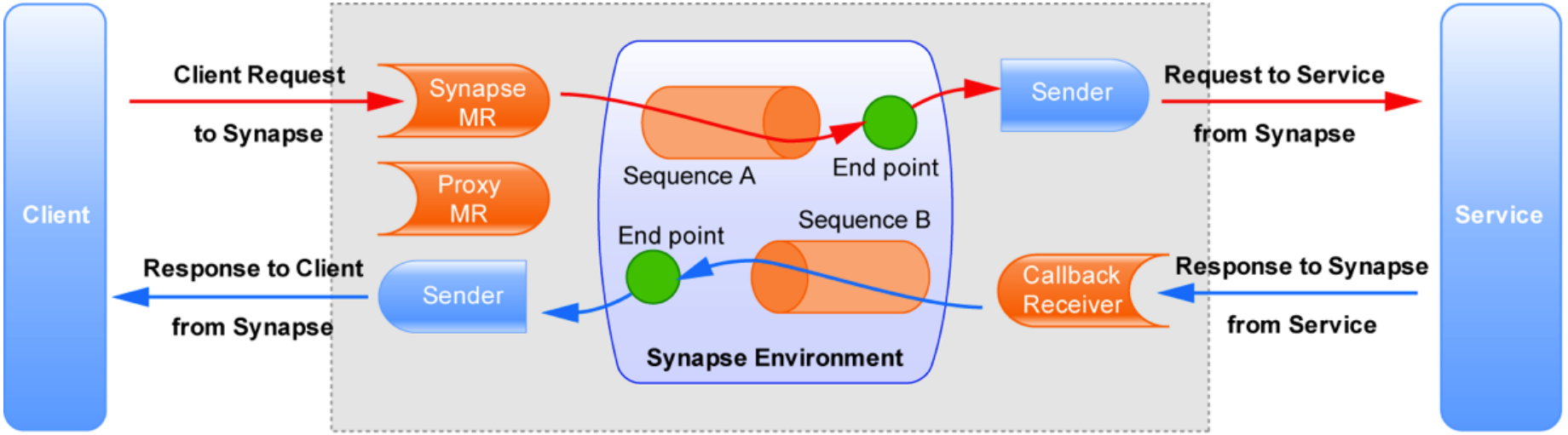
ESB models

- Almost all ESBs work on the same principle

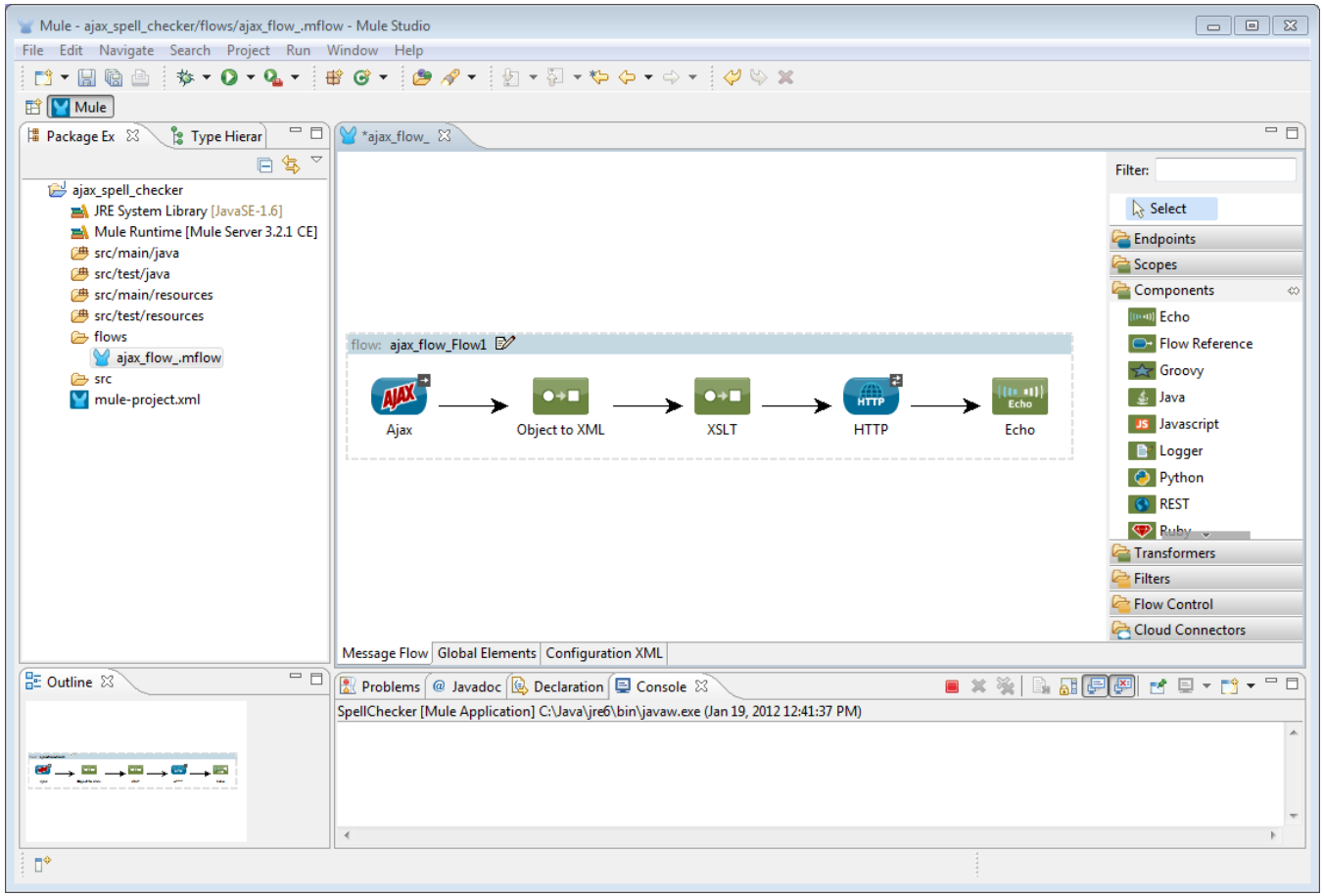
- Message arrives
- Sequence of actions (Pipeline)
- Message is sent on

Graphically

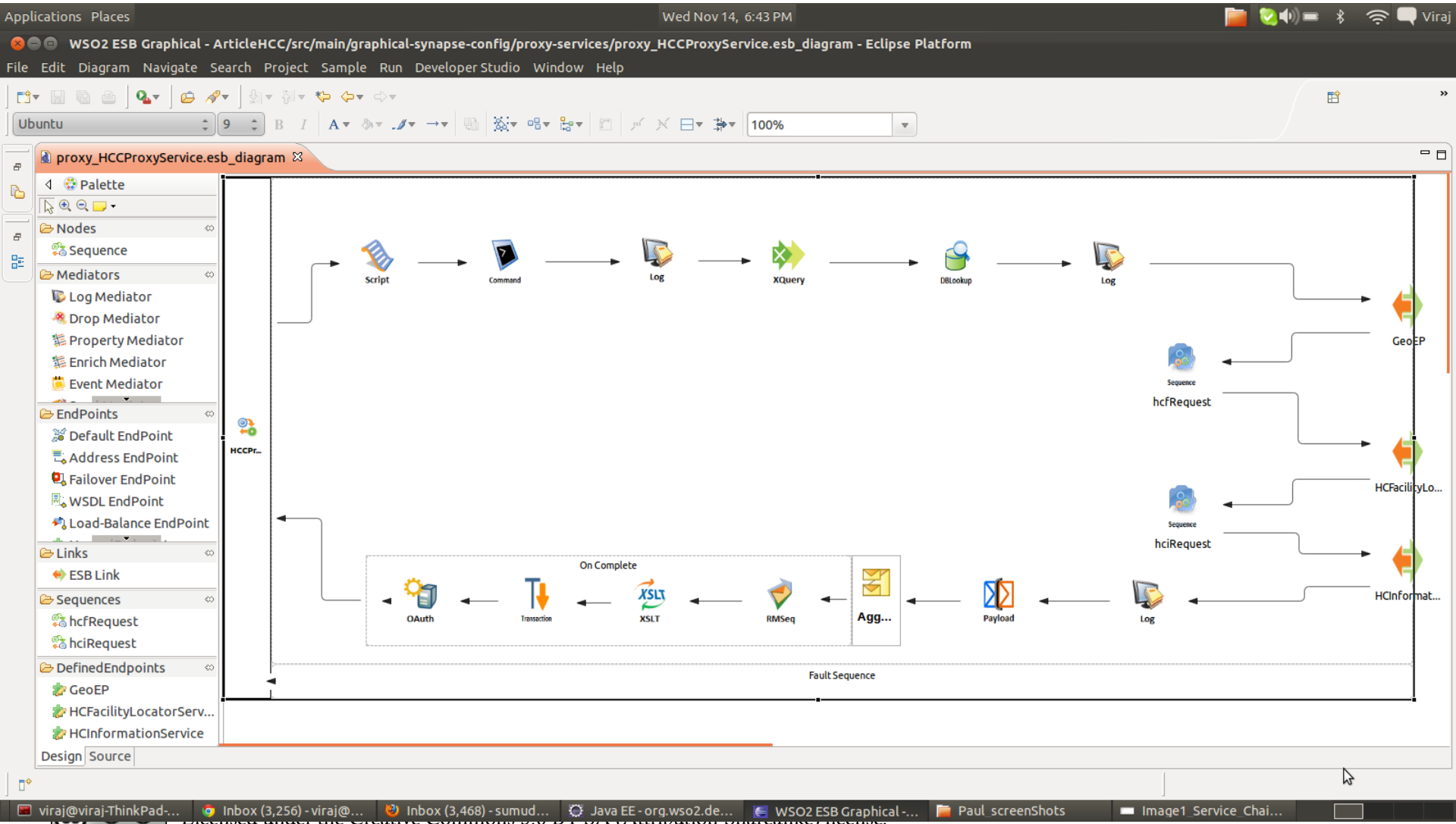
Apache Synapse terminology used

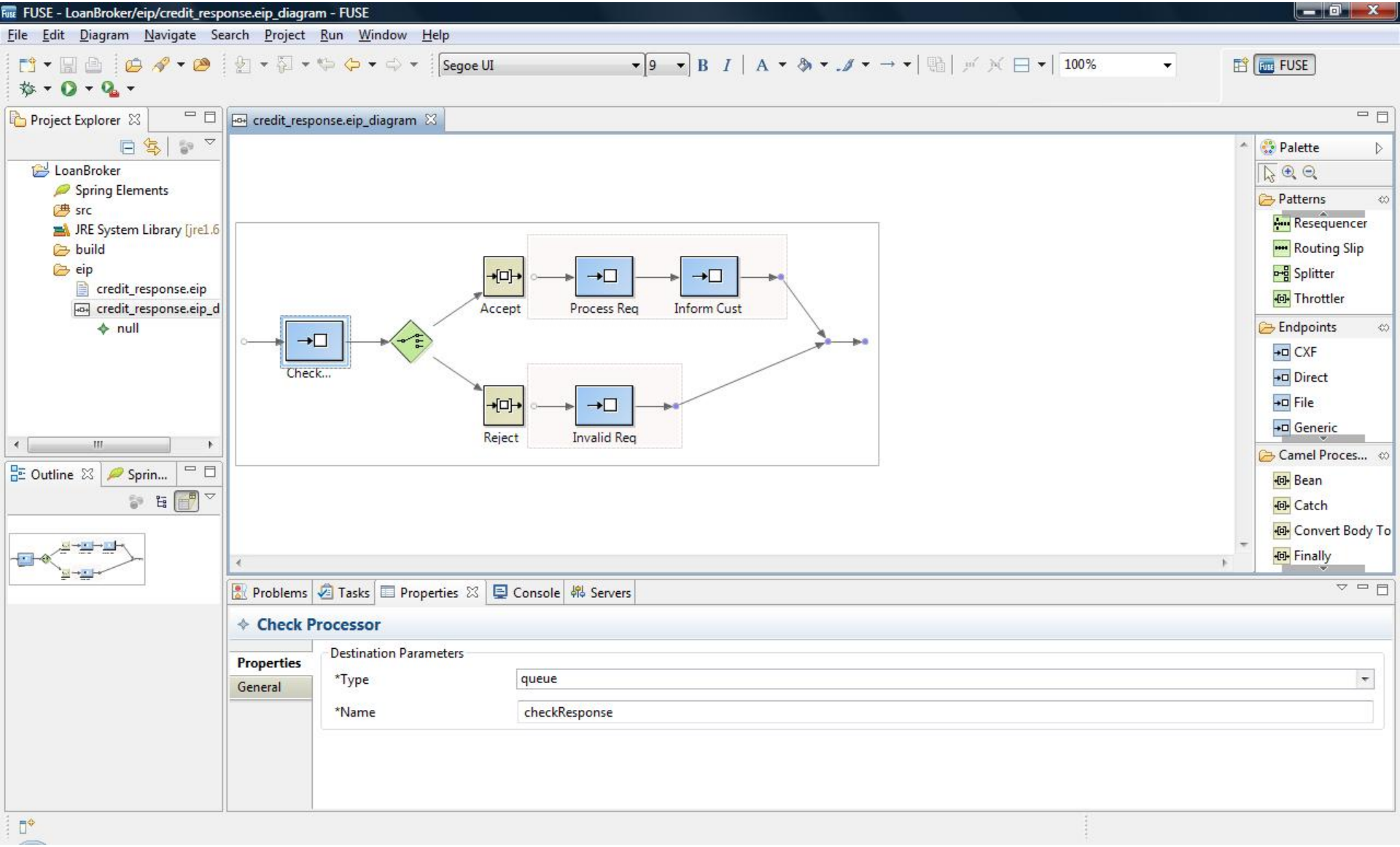


From some tooling

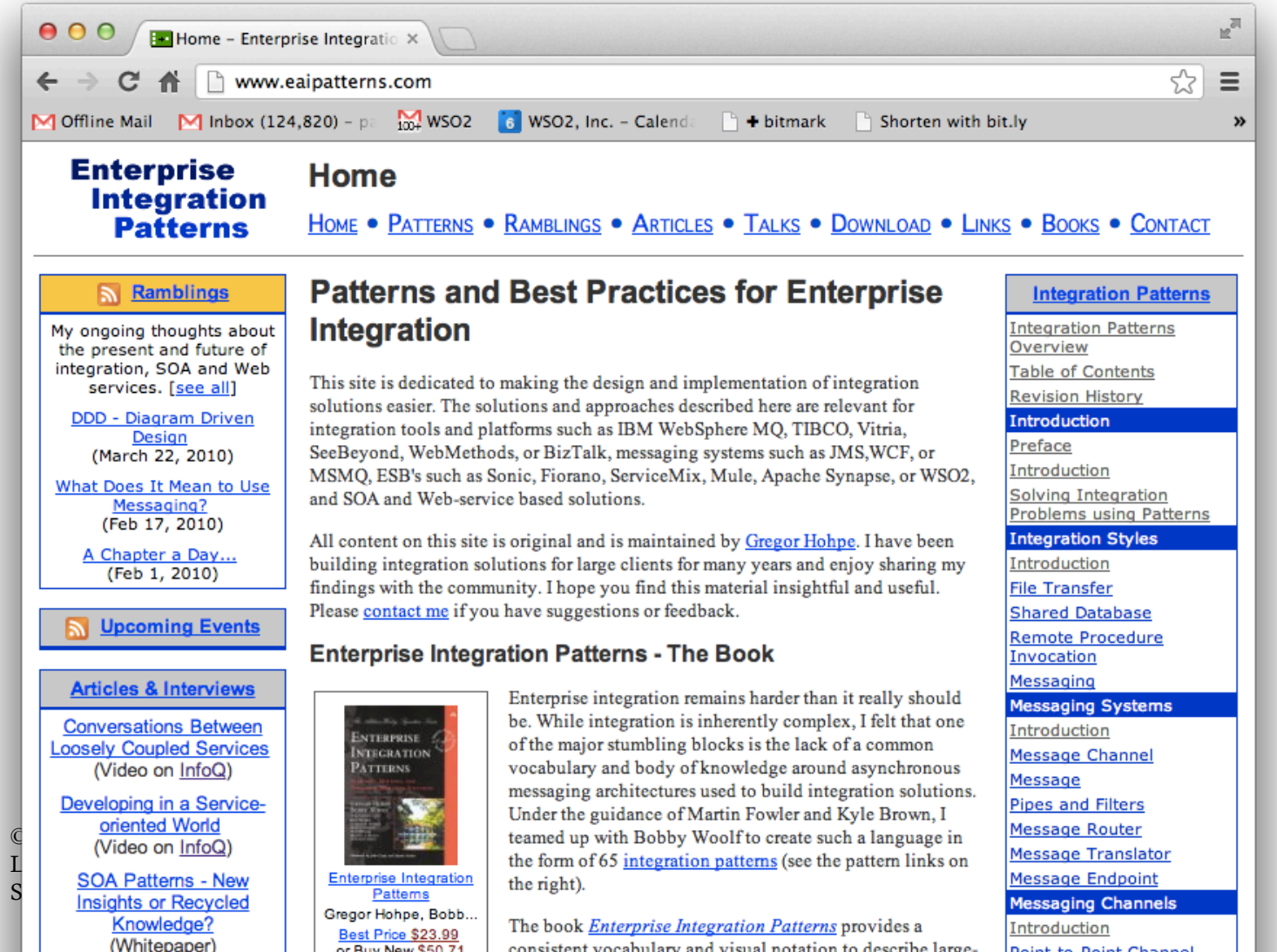


More Tooling



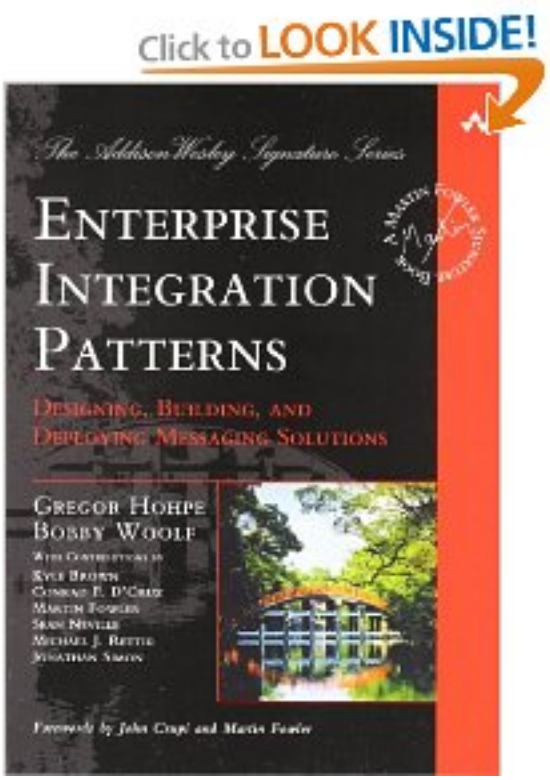


Enterprise Integration Patterns



Enterprise Integration Patterns

- http://www.eaipatterns.com/
- The book
  - Enterprise Integration Patterns
  - Gregor Hohpe, Bobby Woolf



What actions

- The aim is to re-use existing adapters, transports and mediators/transformers
- Why?
  - Minimize custom coding
  - Utilize optimal components
    - e.g. streaming high-performance
  - Shorten test cycles
  - Be more agile

Common mediators

- Logging
- Routing
- Transformation
  - XSLT
  - Xquery
  - Template-ing
- Split/Aggregate
- Filter
- Clone/Tee
- Callout
- Enrich
- Drop
- Fault
- etc

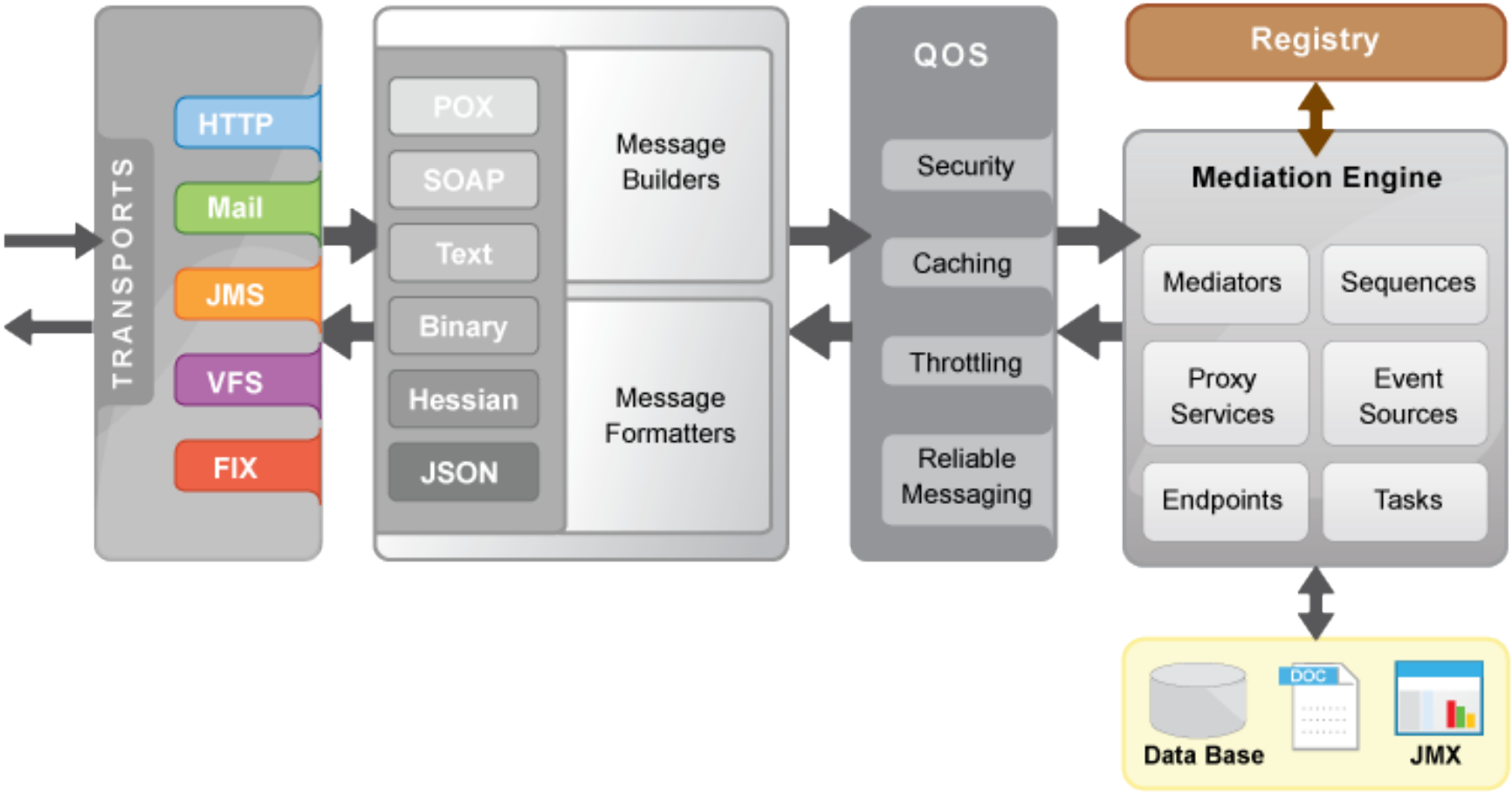
Apache Synapse

- Designed to be simple to use and manage
  - XML configuration
  - No complex deployment
  - Hot deploy and update if needed
  - Separation of configs for different teams
  - Highly performant and scalable
  - Asynchronous core / non-blocking model

C10k Problem

- How to handle 10k concurrent requests
- Without 10k concurrent threads ☺
- Need to disassociate the socket from the thread
- Async handling
- Reactor pattern

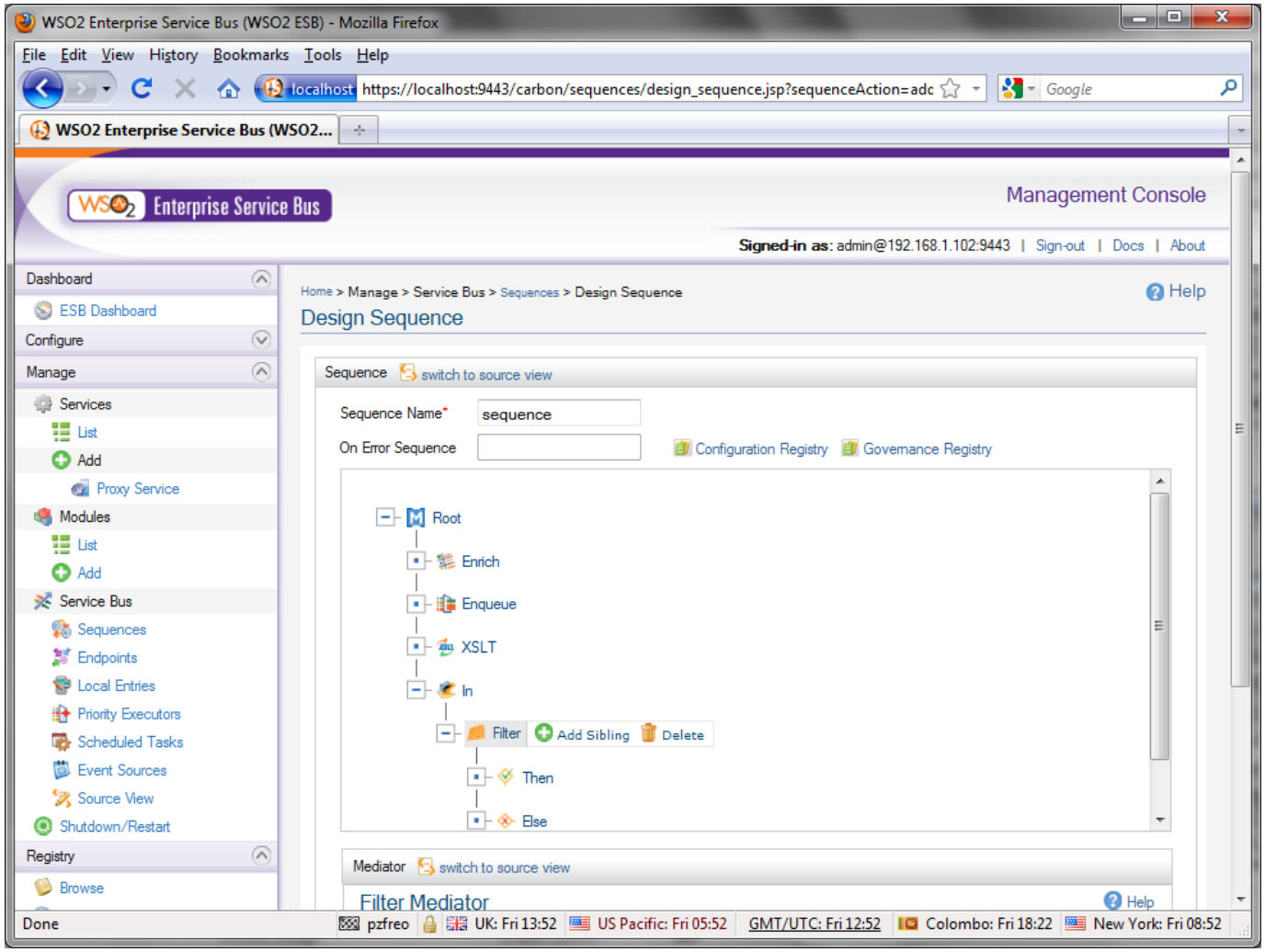
Apache Synapse



WSO2 ESB

- Also Apache License Open Source
- Adds a Graphical Web Interface
- Registry/Repository
- Deployment management/synchronization
- Other pluggable components

ESB UI

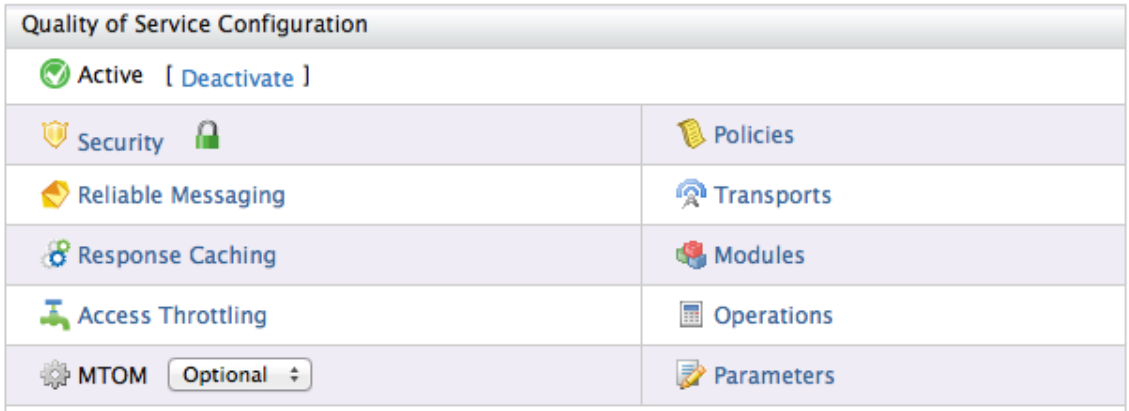


Basic Mediators

Name	Description
Log Mediator 	Logs full or part of the message, at various severity levels (Trace, Debug, etc)
Sequence Mediator 	Invokes existing sequence - Sequence name can be static or dynamic
Send Mediator 	Sends a message out, using static information or endpoint definition.
Callout Mediator 	Performs a blocking external service invocation.
Switch Mediator 	Evaluates messages contents against regular expression and invokes the corresponding mediator (switch-case-default)
Validate Mediator 	Validates message or parts of message against XML schema (schema can be local or in registry)
Drop Mediator 	Stops processing of current message
Fault Mediator 	Transforms current message into custom Fault message

Policy Driven

- Apply out-of-the-box policies to proxy services for
  - Security
  - Caching
  - Throttling
- Create and apply WS-Policies
- Apply Policies stored in Registry



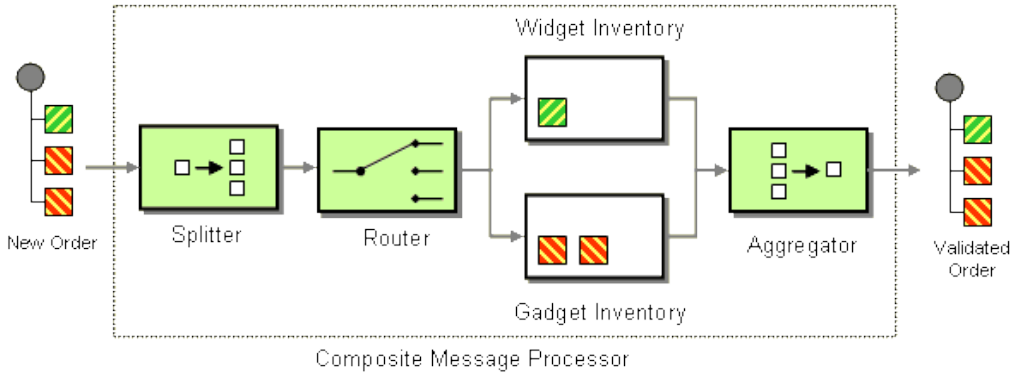
Transformation

- Transform via XSLT, XQuery, or Smooks
- Enrich via XPATH
- URL/Headers Management

Name	Description
XSLT Mediator	Invokes XSLT transformation on current message (v1.0 and v2.0 are supported)
XQuery Mediator	Invokes XQuery transformation on current message
Smooks Mediator	Invokes embedded Smooks Engine (v1.5) - Supports binary transformations (EDI, CSV, etc.)
Enrich Mediator	Enrich message contents using XPATH (replace, append, remove)
URL Rewrite Mediator	Rewrite protocol / URL contents
Header Mediator	Set / Remove Headers
Payload Factory	Override Message Contents

Enterprise Integration Patterns

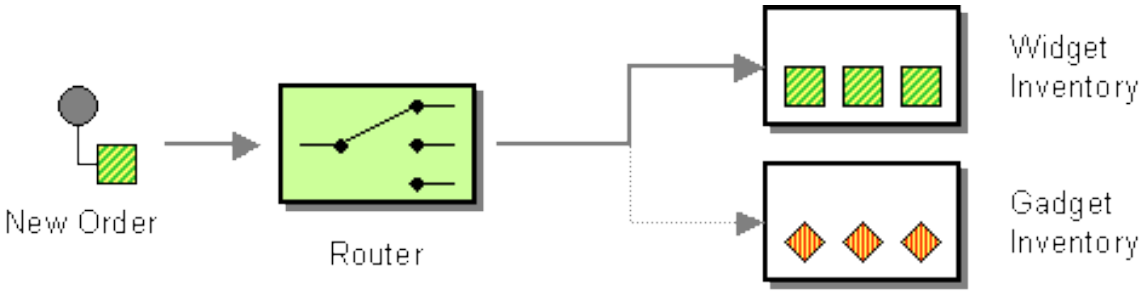
- Native Support for Common EIP
  - Content-based Router
  - Command Message
  - Message Filter
  - Message Splitter
  - Message Aggregator



Name	Description
Route Mediator	Routes message to given endpoint
POJOCommand	Creates instance of specific command class.
Iterate Mediator	Iterates over message and splits it into number of different messages derived from the parent message using XPATH.
Clone Mediator	Clones the entire message N times, each message is then treated in parallel
Aggregate	Aggregates multiple responses or messages, using XPATH.
Filter Mediator	Executes action based on evaluation of message contents against regular expression.

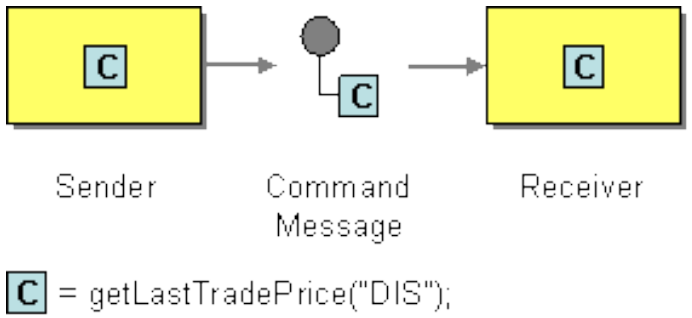
Content-Based Router

- <router> mediator



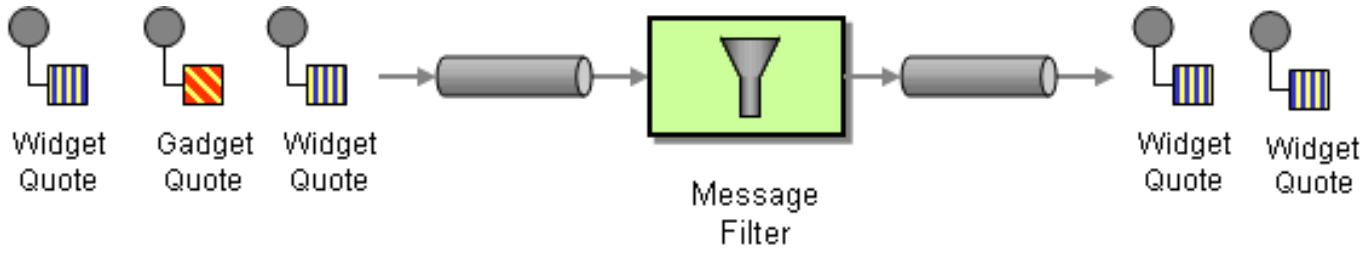
Command Message

- <callout> mediator



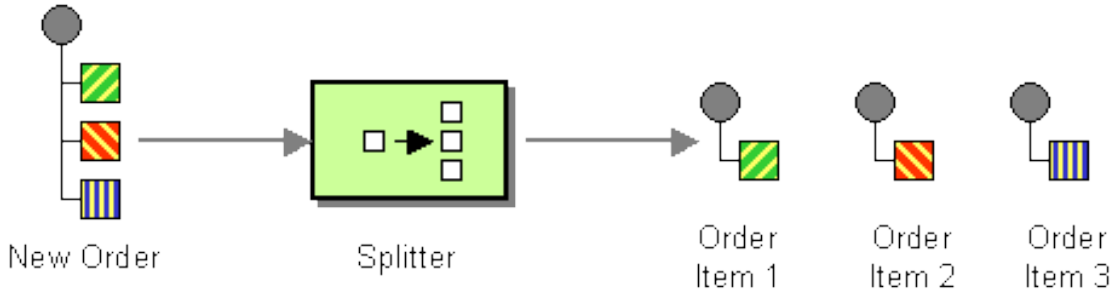
Message Filter

- <filter> mediator (with <drop> mediator)



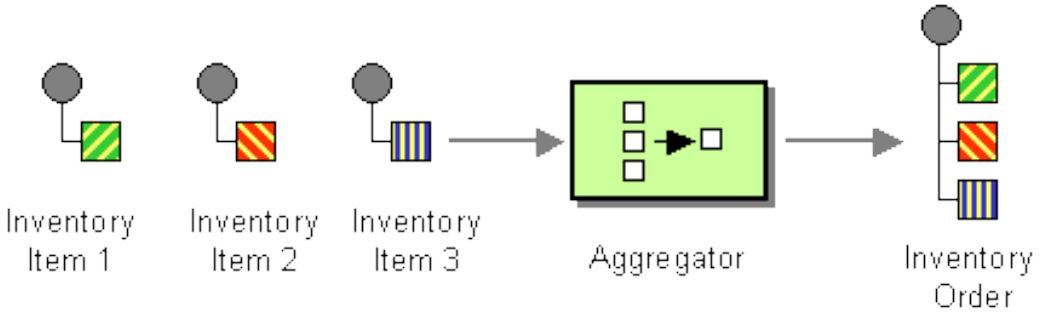
Splitter

- Iterate Mediator



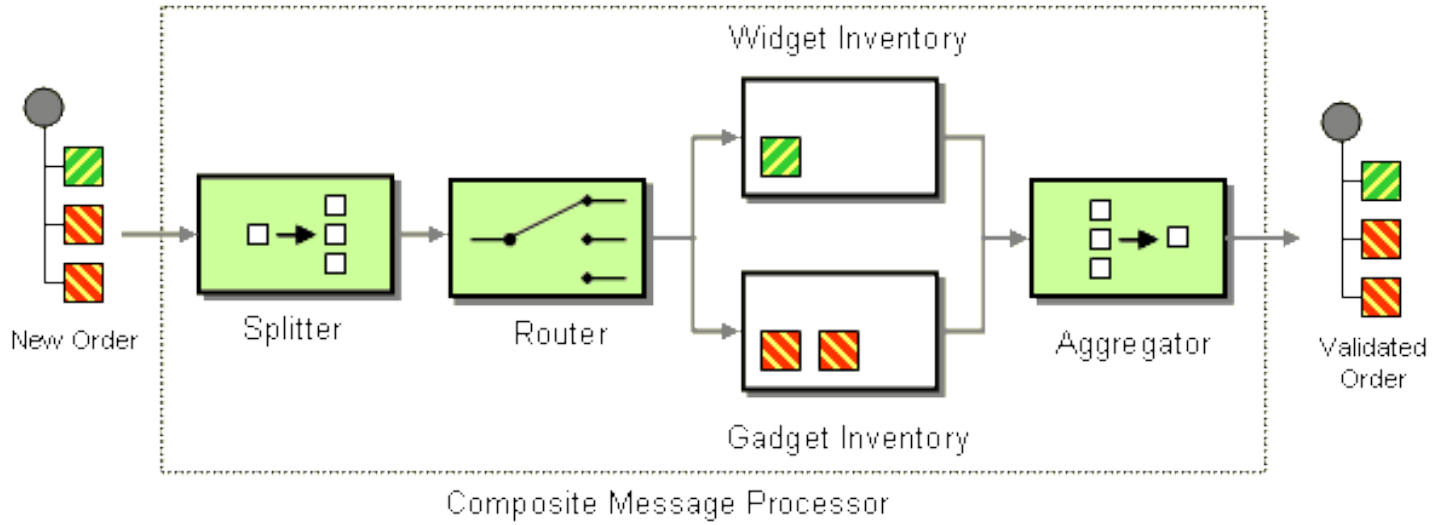
Aggregator

- Aggregate mediator



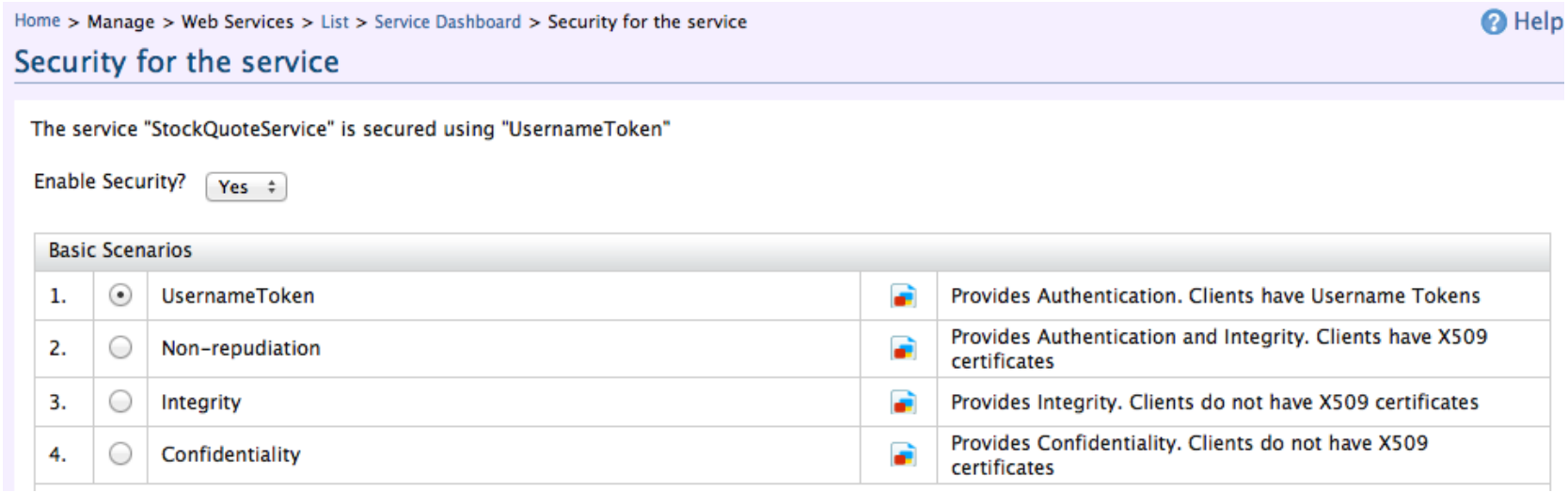
Composed Message Processor

- <sequence>



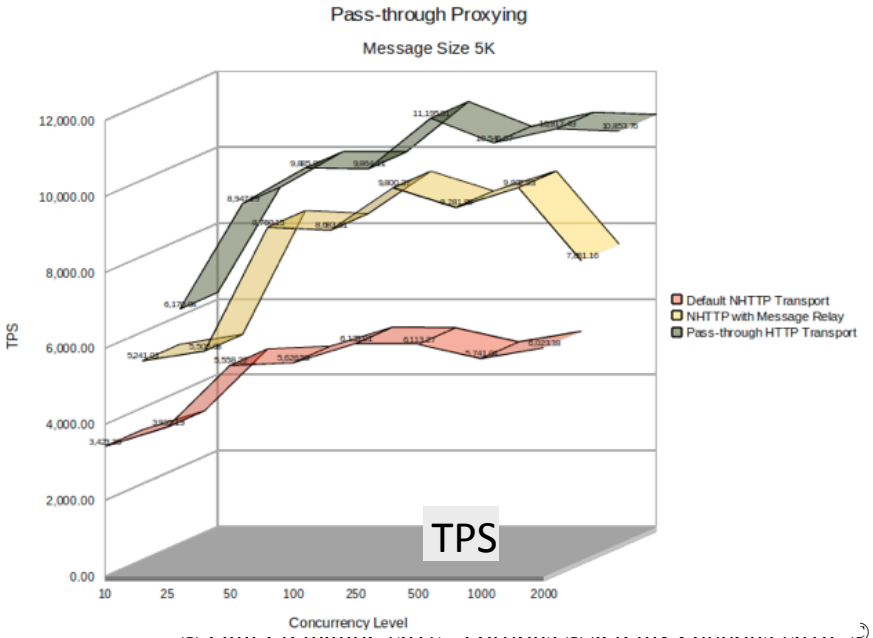
Security

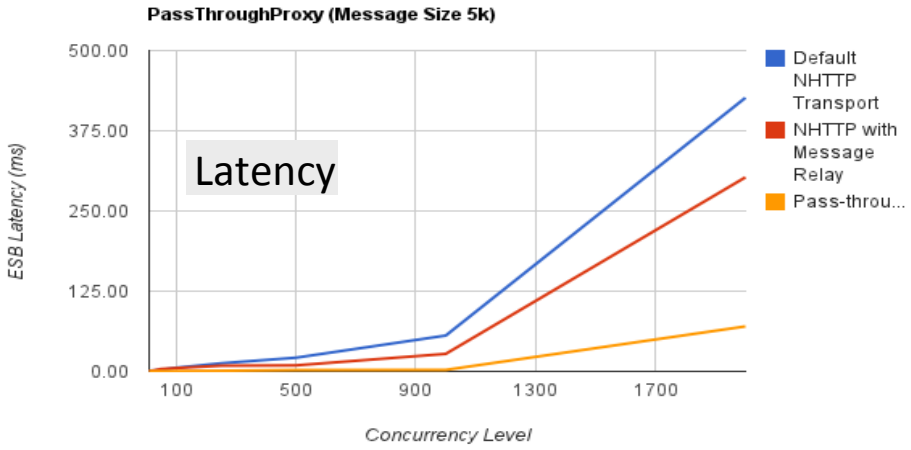
- Supports Authentication via HTTP Basic, UserName Token, SSL, OAuth, Kerberos, OpenID, SAML
- Integration with various LDAP servers (OpenDS, Oracle, IBM..)
- XML Encryption, Digital Signatures, WS-Secure Conversations
- Acts as PEP for fined-grained authorization (entitlements) using XACML



High Performance and Stability

- Supports 1000s of concurrent non-blocking HTTP transaction per server
- Pure streaming and Optimization using Message relay (on-demand processing of messages)
- Very Low latency (0.5 ms for Non-Blocking IO transport)
- Long Term Execution Stability with Low Resources Utilization
- Response Caching





High Availability and Scalability

- Supports Active/Active, Active/Passive Scenarios

- ESB itself can act as load-balancer.
- Auto-scaling using Load Balancer component
- Deployment Synchronizer can be used to maintain configuration across clusters.

Extensibility

- Supports Scripting Language (JavaScript, JRuby, Groovy)
- Java extension via POJO calls
- Can be extended via custom mediators

Name	Description
Script Mediator	Calls scripts via Bean Scripting Framework (Java, JRuby, Groovy)
Class Mediator	Invoke your own mediator

- Extend configuration vocabulary with custom domain-specific languages via **templates**.

Resources

- Wikipedia!
  - http://en.wikipedia.org/wiki/Enterprise_service_bus
- Books
  - David Chappell: ESB
  - Open Source ESBs in Action
- Open Source
  - synapse.apache.org
  - wso2.com/products/enterprise-service-bus
  - servicemix.apache.org