JBoss WORLD
CHICAGO 2009

Next Generation Open Source SOA

Burr Sutter
Sr. Product Manager, SOA
(JBossESB, Riftsaw, jBPM, Drools)
September 3, 2009

presented by

Agenda

Is SOA Dead?

JBoss SOA Platform Overview & Roadmap

RESTful inclinations

Complex Event Processing (CEP) w/ Drools Fusion

Infinispan + ESB

BPEL

JBoss WORLD CHICAGO 2009

Is SOA Dead?

"The report of my death was an exaggeration."

Mark Twain

SOA, ROA & WOA – Oh My!

ROA – Resource Oriented Architecture - REST (Representational State Transfer – Roy Fielding)

WOA – "Web Oriented Architecture (WOA) is a style of software architecture that extends service-oriented architecture (SOA) to web based applications, and is sometimes considered to be a light-weight version of SOA. WOA is also aimed at maximizing the browser and server interactions by use of technologies such as REST and POX." - Wikipedia

SOA is **NOT** SOAP

- architectural style vs wire protocol

SOA is **NOT** WS-*

- patterns & principles vs a large body of standards

SOAP vs REST Technical Differences

HTTP Verbs:
- SOAP – POST
- REST – GET, PUT, POST, DELETE

Contract Definition:
- SOAP – WSDL – operations & messages
- REST – HTTP Verbs are the operations, messages may or may not have a contract via XSD and/or JSON (POX – plain 'ol XML)

Content-Type:
- SOAP – focused on XML
- REST – allows for any payload – XML, JSON, Atom, RSS, etc.

Clients:
- SOAP – requires a client that understands WSDL
- REST – requires a client that understand HTTP

Forget Services, I want Events! - EDA

EDA – "*Event Driven Architecture is a software architecture pattern promoting the production, detection, consumption of, and reaction to events. An event can be defined as a significant change in state*" - Wikipedia

Services have historically been more request-response focused – synchronous processing – normally for responding to awaiting users/consumers.

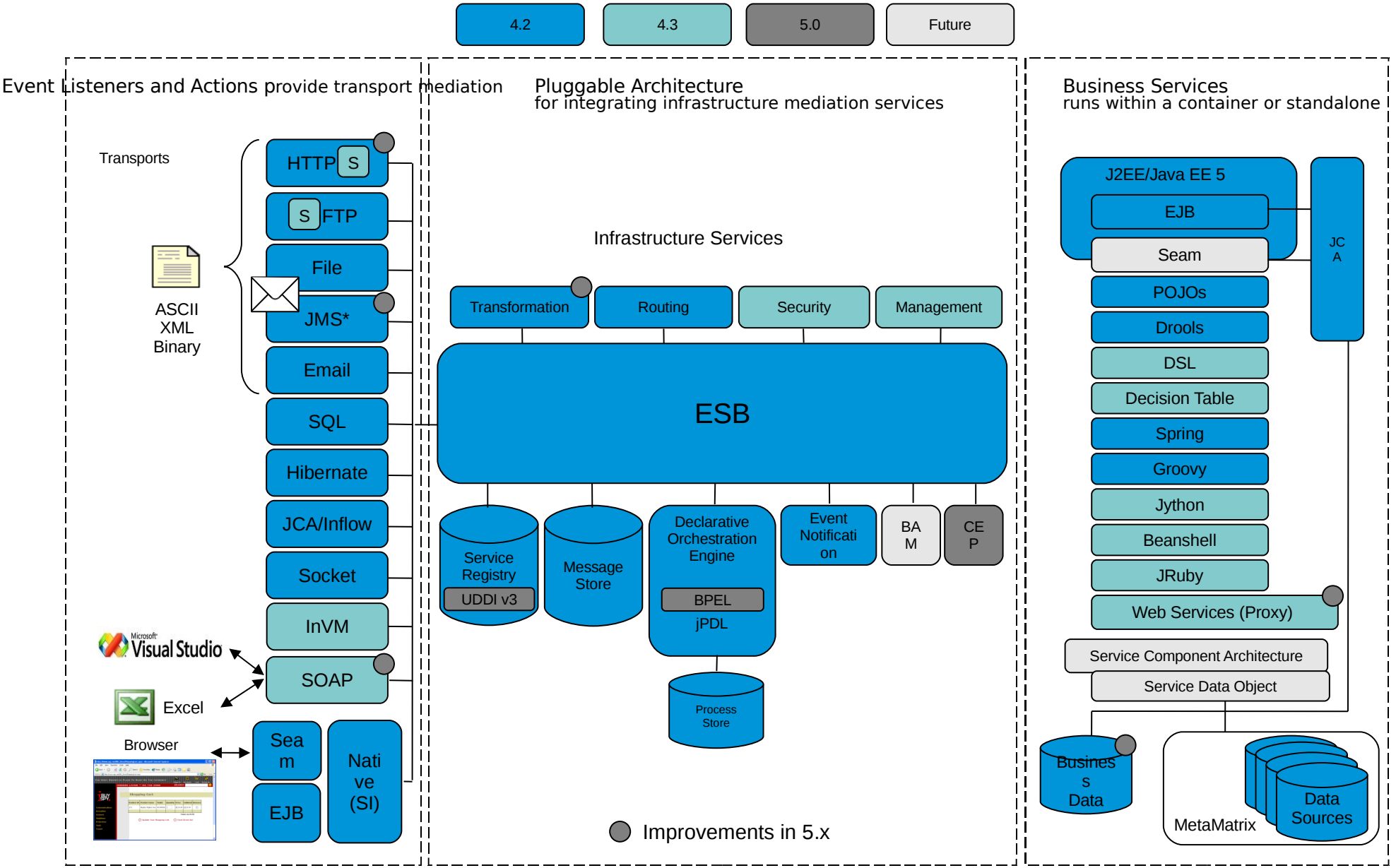
Events are more real-time alerting, no waiting – asynchronous processing – think push.

"*Event-driven architecture can complement service-oriented architecture (SOA) because services can be activated by triggers fired on incoming events*" - Wikipedia.

JBoss WORLD CHICAGO 2009

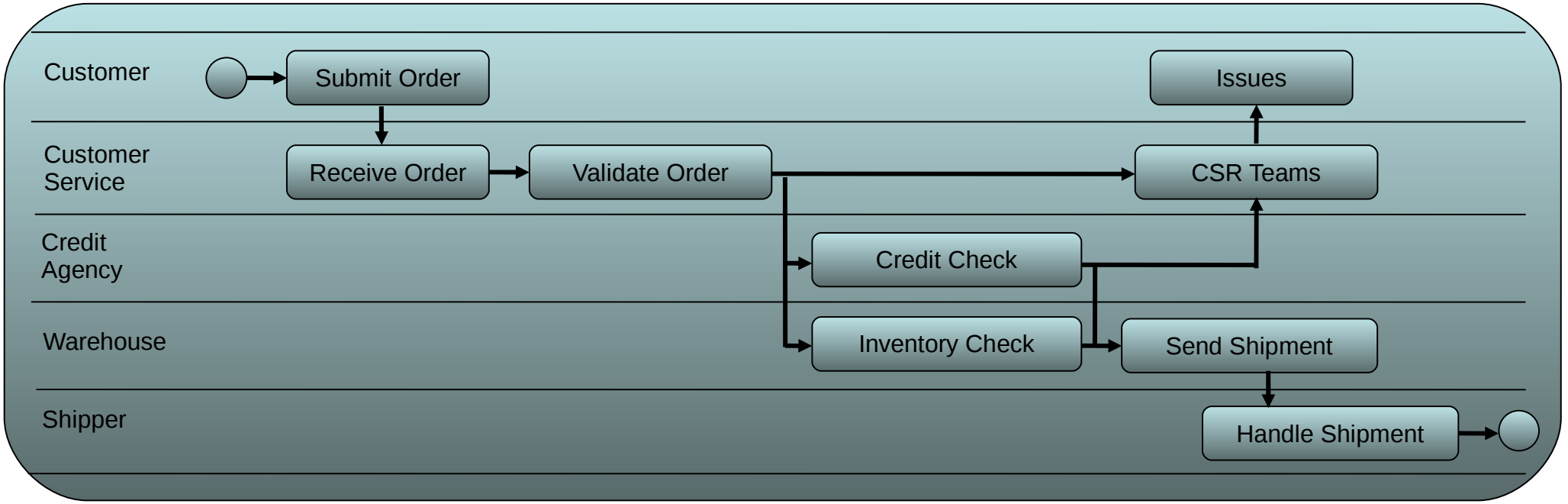
JBoss SOA Middleware Overview

presented by



JMS* - JBoss Messaging, IBM WebsphereMQ, TIBCO EMS, MRG-M

Beyond ESB: Context



Validate Order

a Parse XML
b Transform
c Apply Business Rules

Credit Check

a Create Outbound Msg
b Handle Response
c Apply Business Rules

Send Shipment

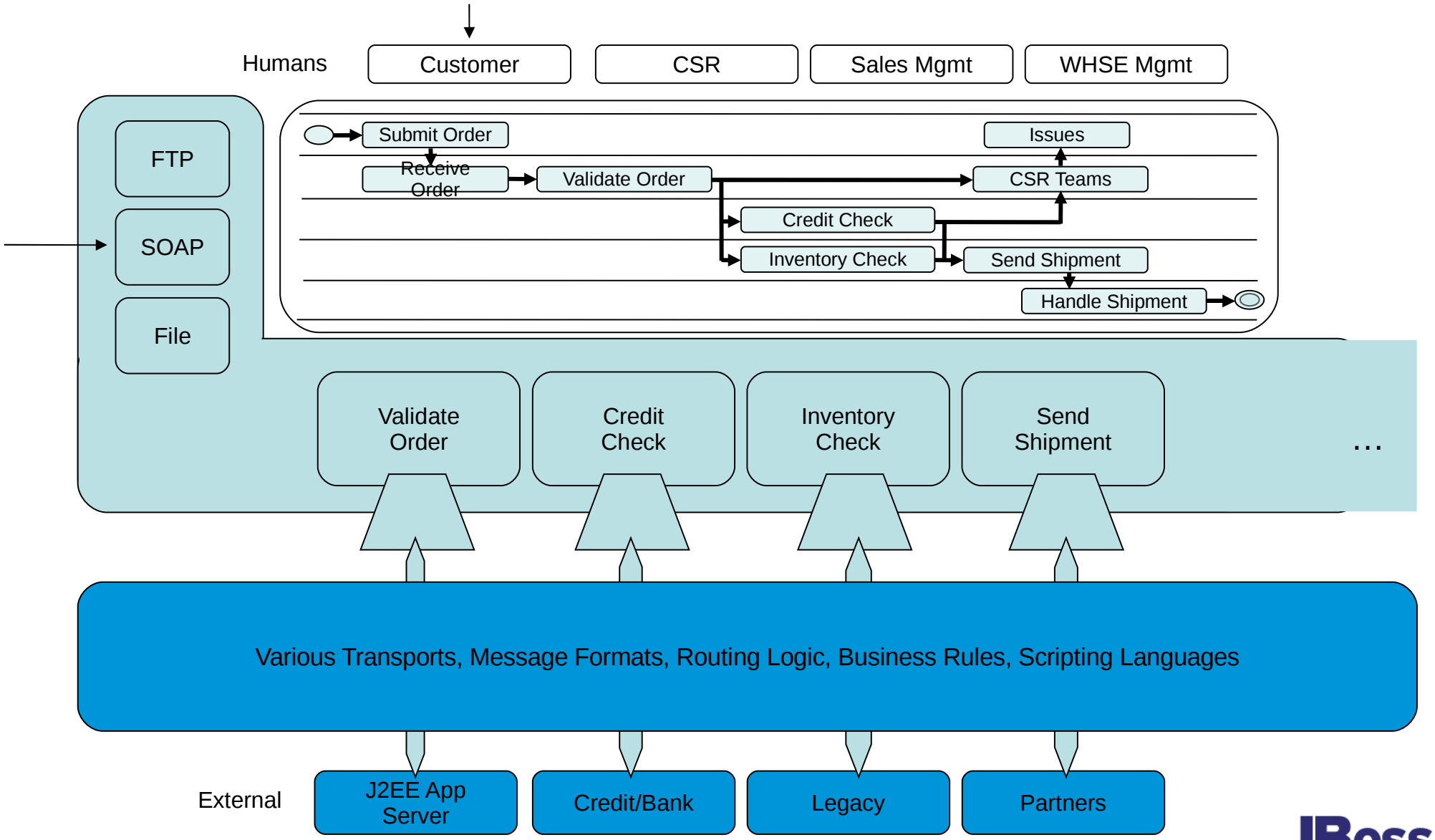
a Determine Shipper(s)
b Print Labels
c Print Pick Tickets
d Create & Send ASNs

Inventory Check

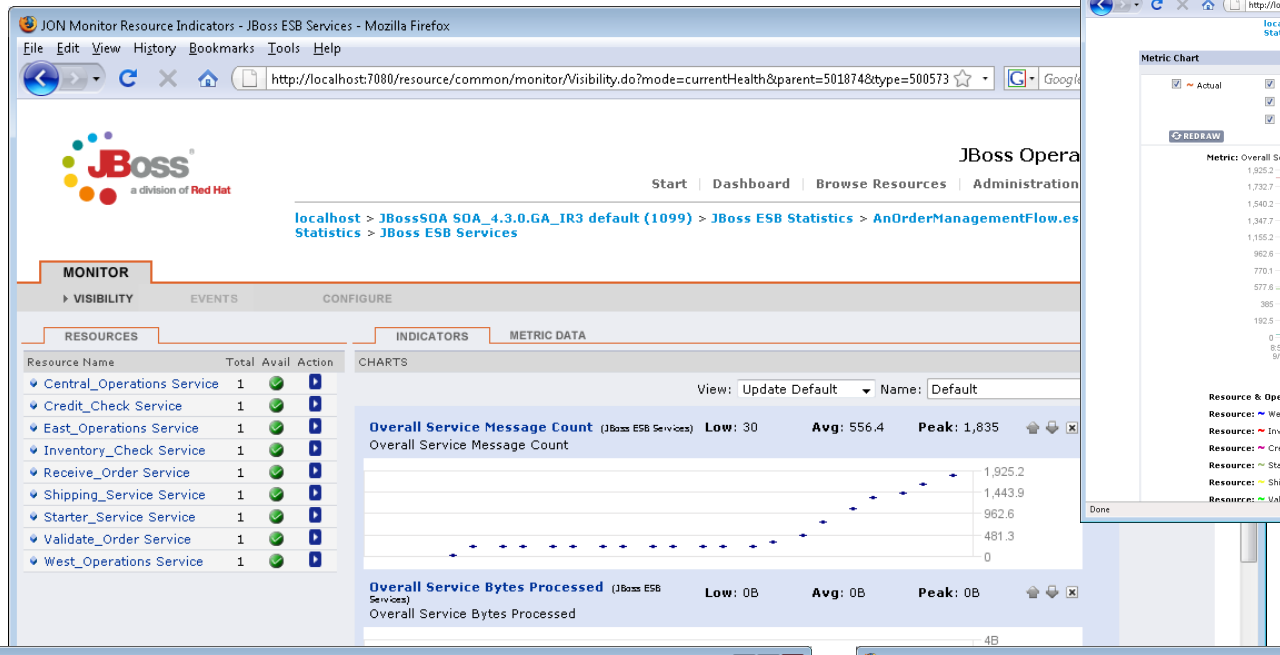
a Send to N Warehouses
b Handle N Responses
c Determine Best WHSEs
d Handle Drop-Ships

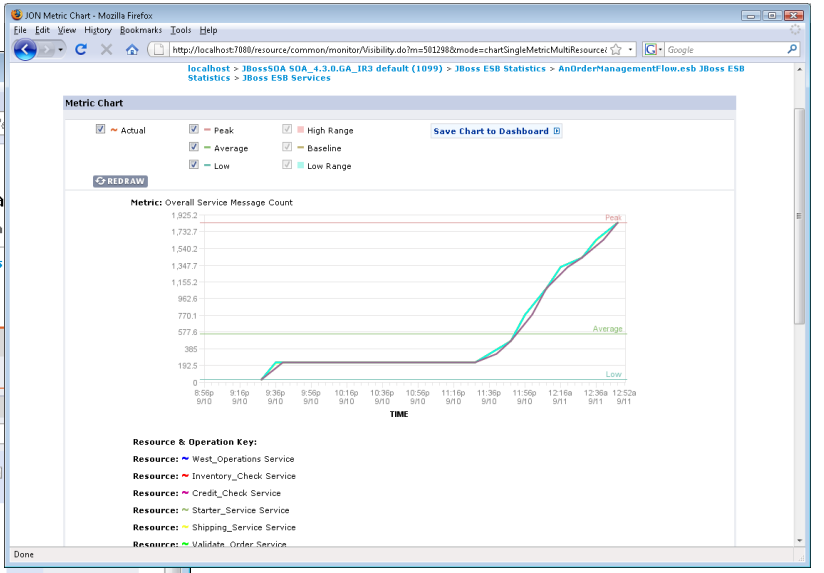
ESB Mediates
& Provides Services

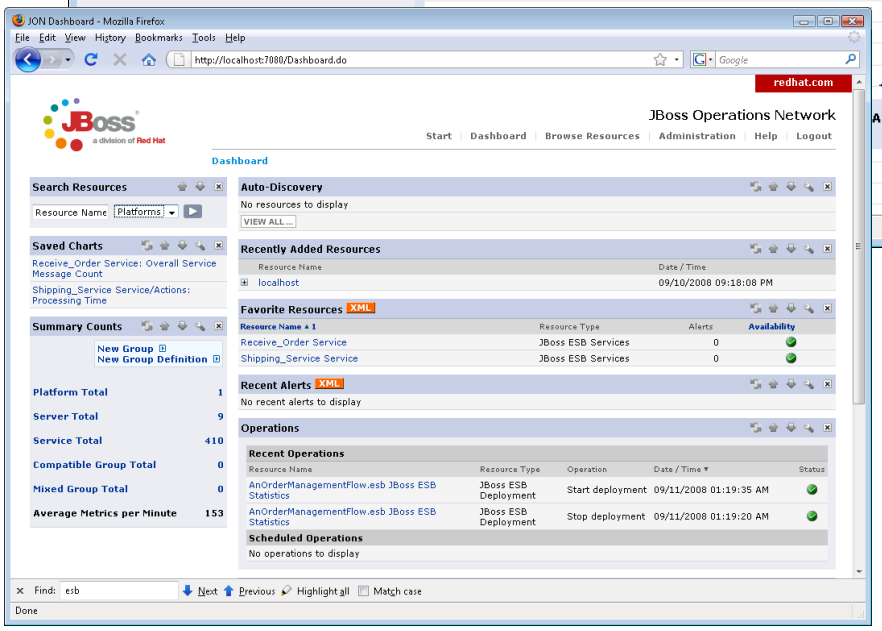
Synergy of ESB + BPM + Rules

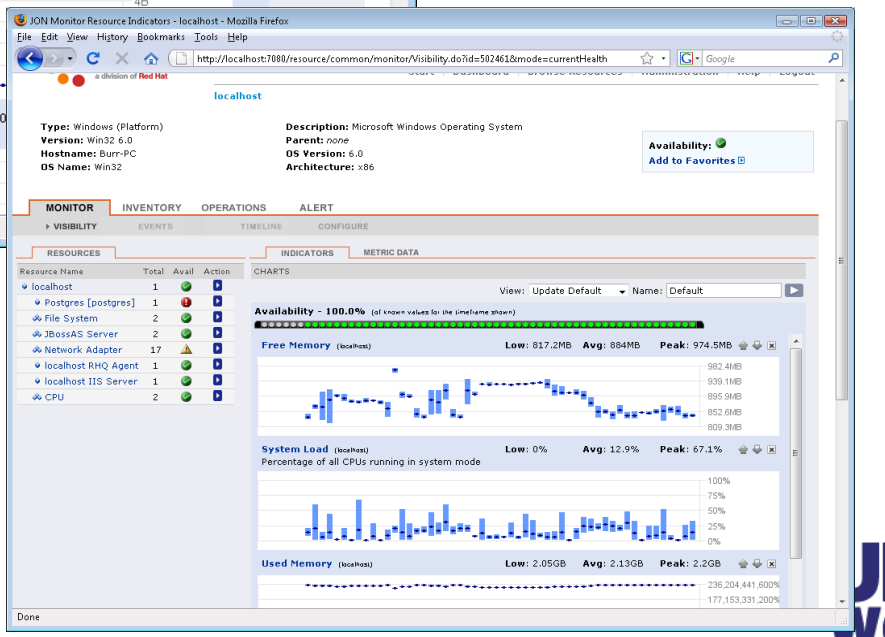


SOA Monitoring and Management









JBoss WORLD
CHICAGO 2009

REST

presented by

Value of REST with an ESB

JBossESB is not hard-wired into Web Services nor JMS, it is async event-driven and also allows for sync request-response.

RESTful style interactions (GET, PUT, POST, DELETE) allow for Groovy, JavaScript, Ruby, etc clients to more easily consume a service.

JBossESB now allows for different content types and use of HTTP response codes.

REST is based on HTTP – new http-provider (ESB 4.7)

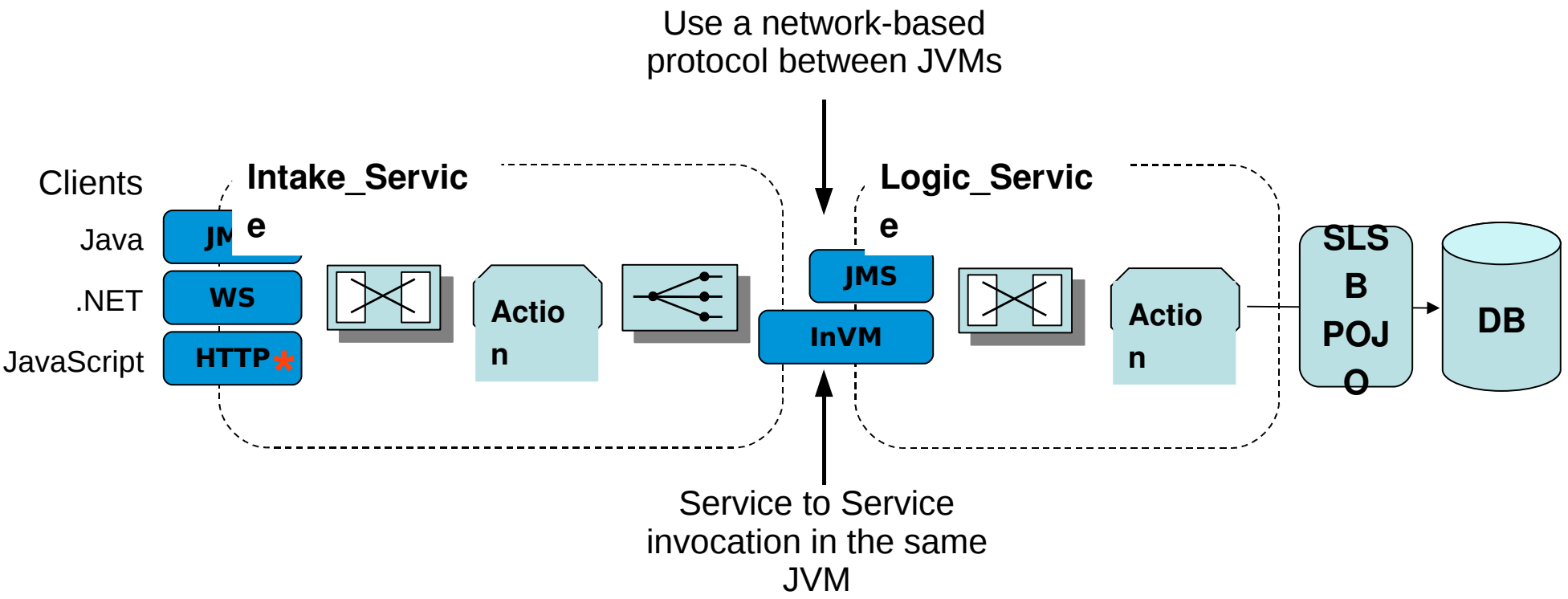
New HTTP Provider (jboss-esb.xml)

```
  <providers>
        <http-provider name="http">
            <http-bus busid="http_gateway"/>
            <exception
             mappingsFile="/http-exception-
mappings.properties" />
        </http-provider>
    </providers>

  <services>
        <service category="Sales" name="List"
          description="" invmScope="GLOBAL">
            <listeners>
                <http-gateway name="sales"
                 busidref="http_gateway"
                 urlPattern="sales/*" />
            </listeners>
            <actions mep="RequestResponse">
                <action name="createAtomFeed"
                        class="atom_publisher.MyAction"/>
            </actions>
        </service>
    </services>
```

http://localhost:8080/MyESBArchive/http/sales/*

ESB Transport Mediation



ATOM Demo

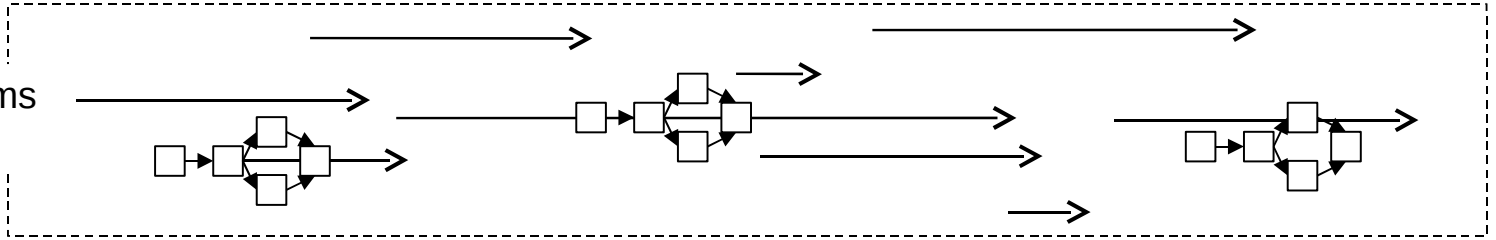
JBoss WORLD
CHICAGO 2009

Complex Event Processing

presented by

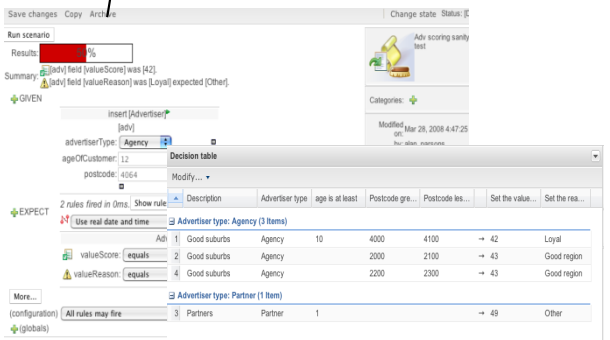
ESP + CEP via ESB, Rules & JON

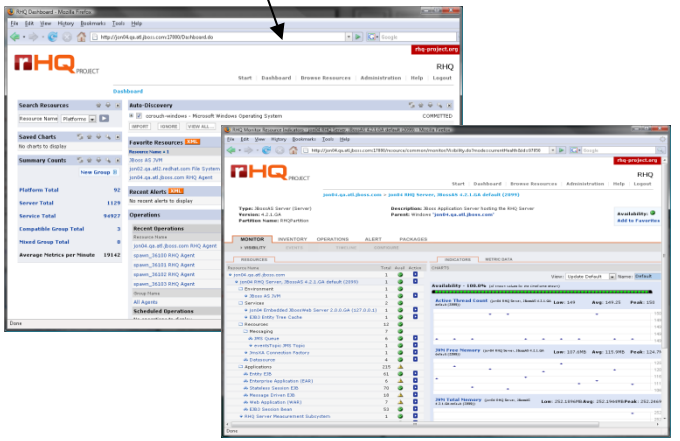
Many Streams
Or Flows



ESB: consumption, capture, transformation, routing, orchestration
Rules: selection, aggregation, correlation, generation and publication

Governance
Tools





Repository: for editing, versioning, testing and publishing new rules and SOA artifacts

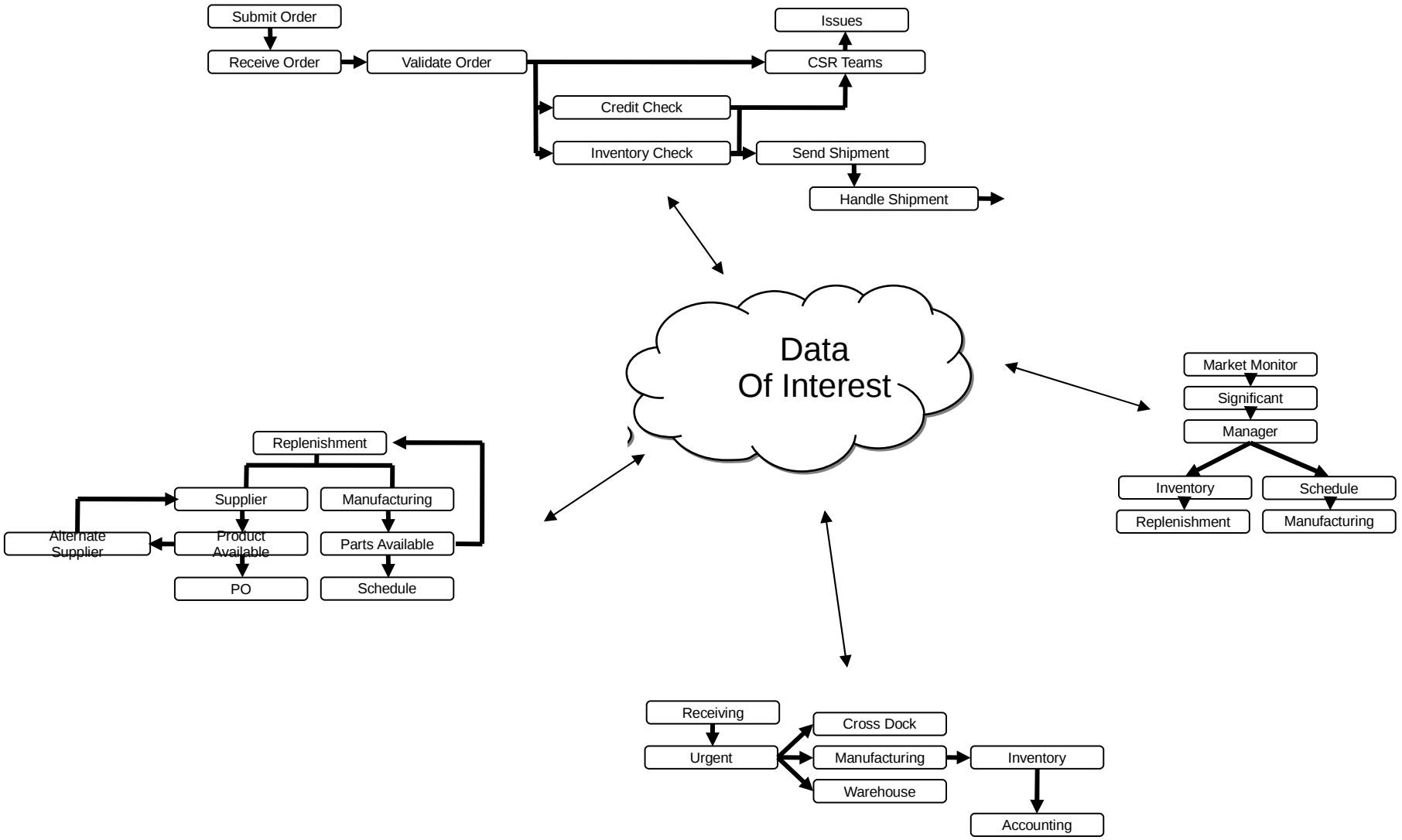
JON: start/stop services, monitor and alert on service and action-level performance, monitor and alert on business metric values

Event Declaration & Selection

```
declare StockTick
    @role( event )
    @expires( 2m )
end
```

```
rule "average over last minute"
when
    $stat : Statistics( $symbol : symbol )
    Number( $av : doubleValue ) from accumulate(
        StockTick( symbol == $symbol, $p : price )
    over window:time( 1m )                                    from entry-poin
        average( $p ) )
then
    modify( $stat ) {
        average = $av
    }
end
```

Deal flow – Business Interceptor



Value of CEP – Drools Fusion

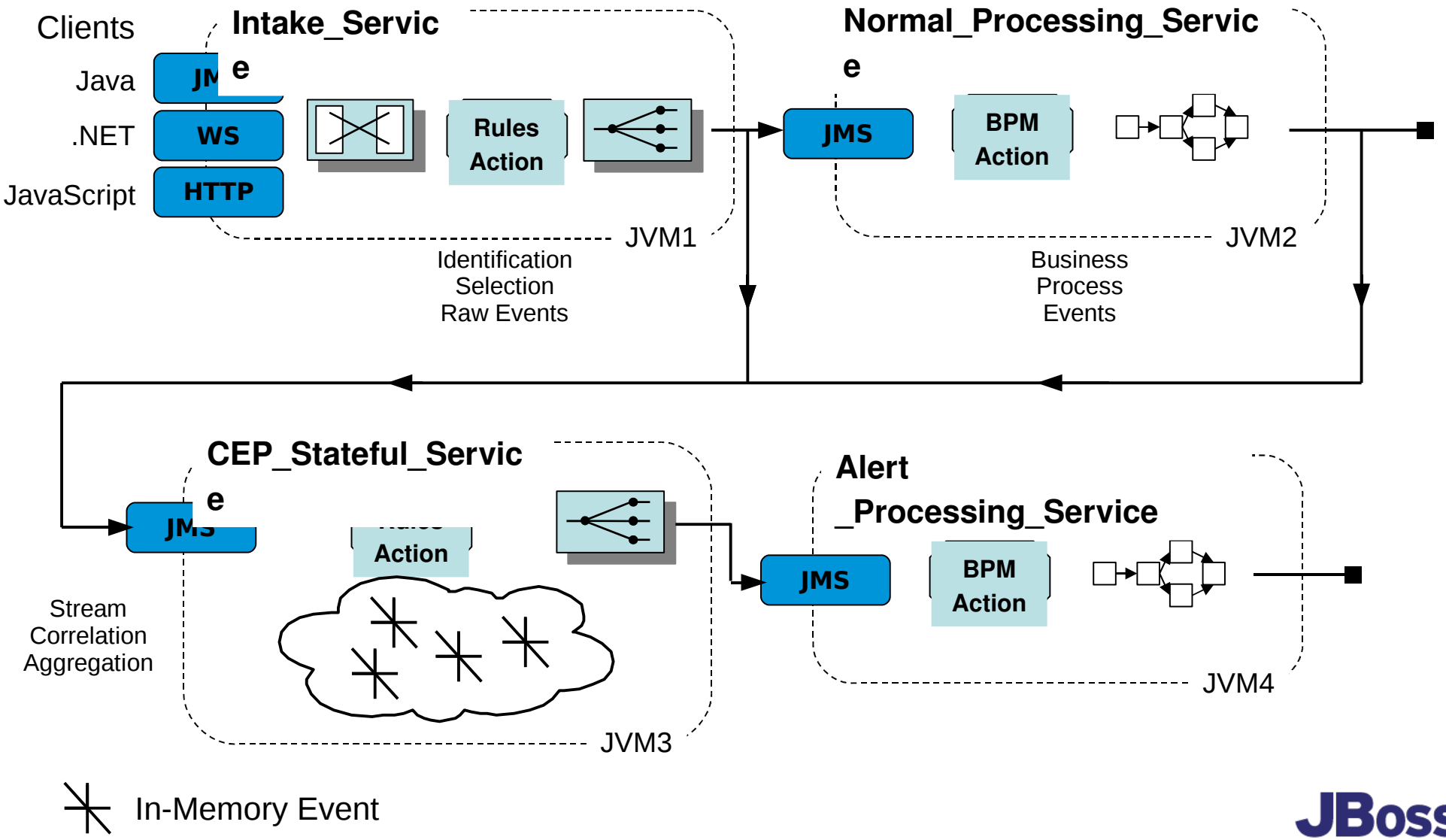
Stateful Rules Engines live in ESB Nodes, they retain the real-time history of the event flow

Rules look for significant/interesting events in the stream, expiring older, insignificant events (memory management) – stateless rules engines/ESB nodes can route to stateful engines.

Event data can be aggregated/accumulated over a time window or event series

New events or messages back through the ESB can be generated, providing an alerting mechanism.

CEP via JBossESB



CEP Demo

JBoss WORLD CHICAGO 2009

Infinispan + ESB

presented by

Quick Introduction: Infinispan

Spiritual successor to JBoss Cache > Data Grid

Still a “peer to peer” architecture, client-server is also available

JSR 107 – JCACHE

Speaks REST, memcached and a custom binary protocol

Improve response times for service requests

Reduce load on critical backend systems

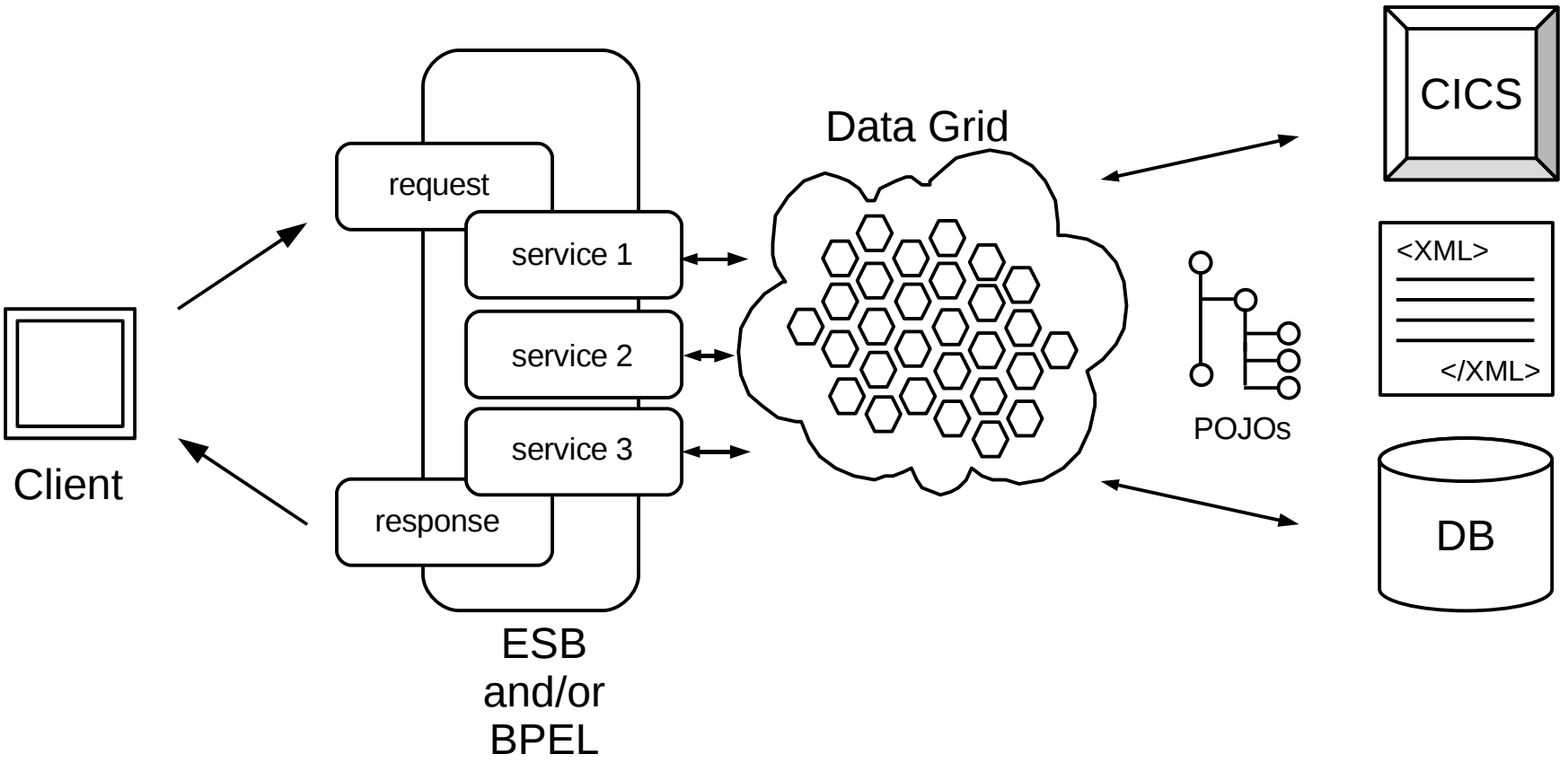
Improve fault tolerance, throughput, performance

Linear scalability

Predictable latency under increasing load

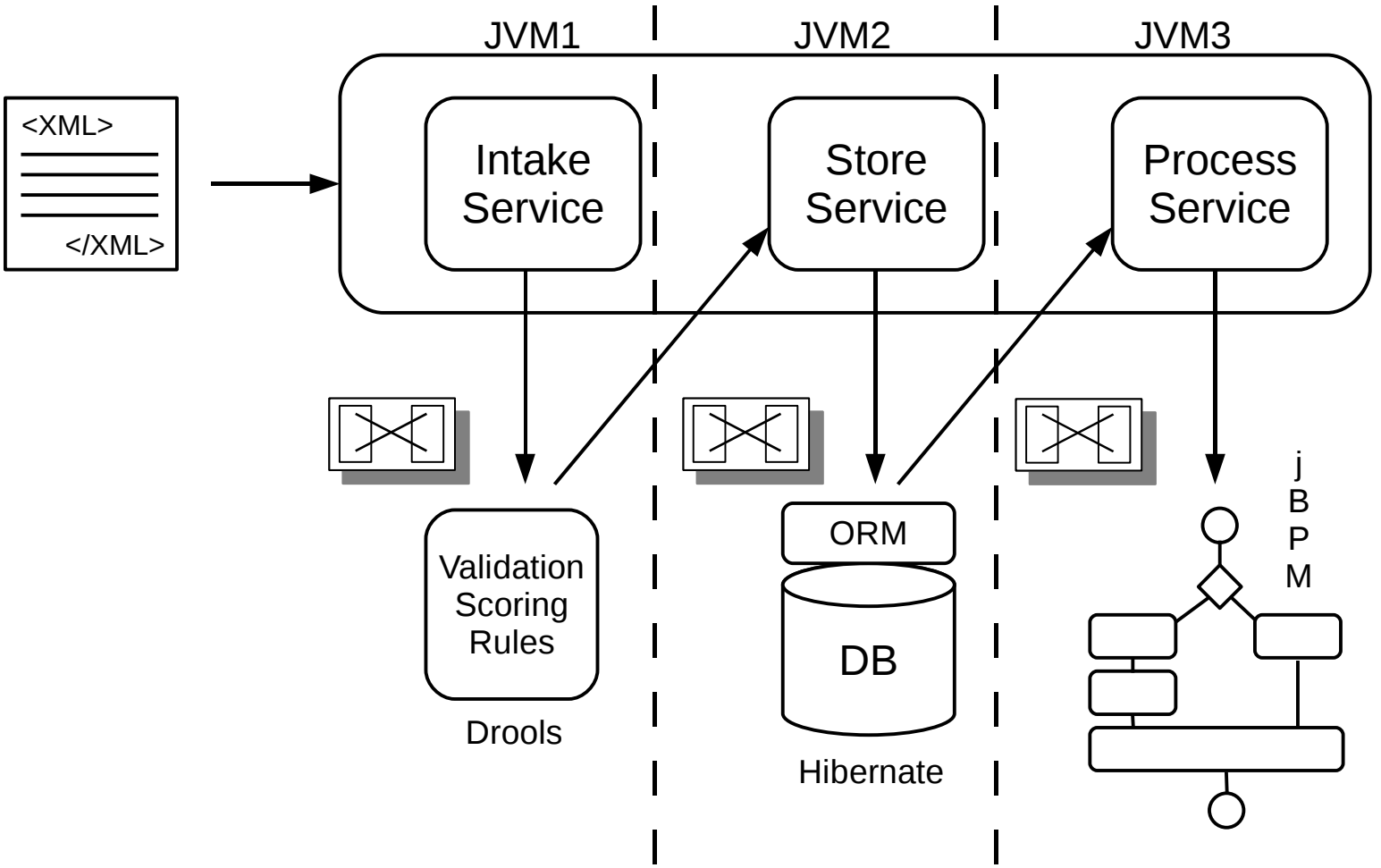
Access to more than a single JVM's Heap

Service Result Cache



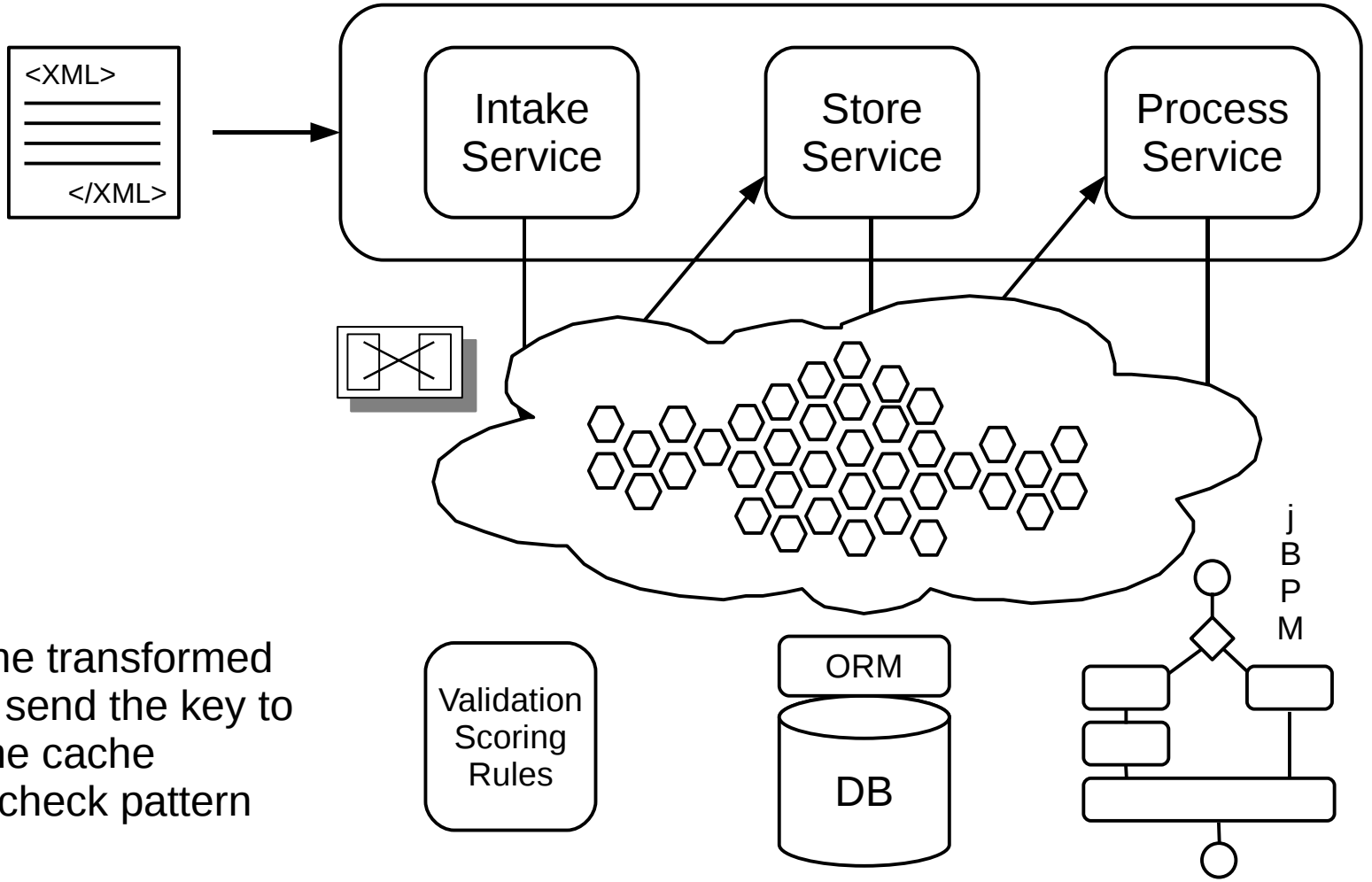
Results of service invocations to “expensive” resources can be cached by the middle-tier architecture. Transformations to POJOs can be time consuming

Hoppity, Hop, Hop



Serializing large XML, transforming multiple times, serializing large POJOs across multiple network/JVM boundaries can be expensive

Still Hopping, less load



Keep the transformed version, send the key to the cache
claim check pattern

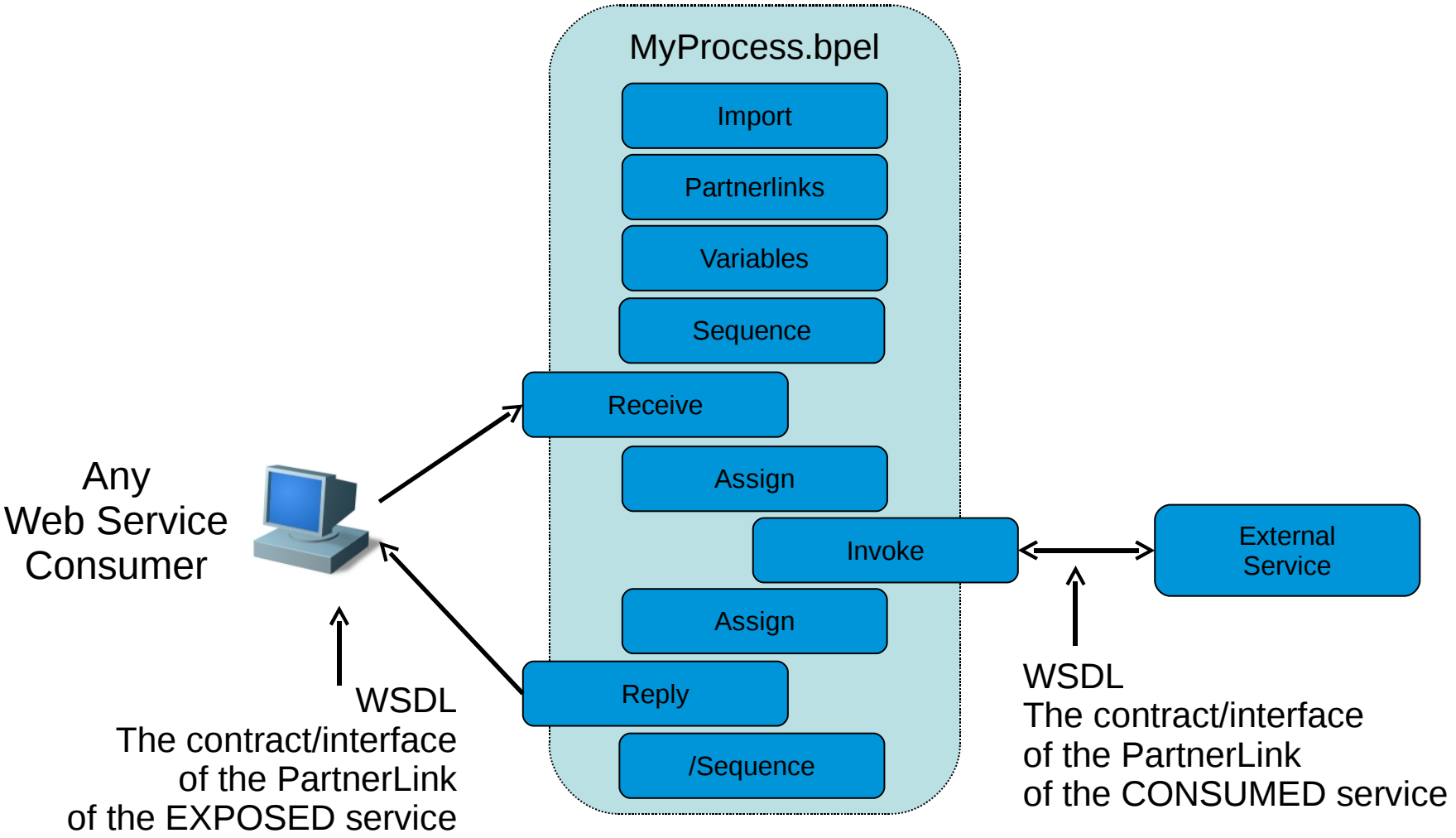
Data Grid Demo

JBoss WORLD
CHICAGO 2009

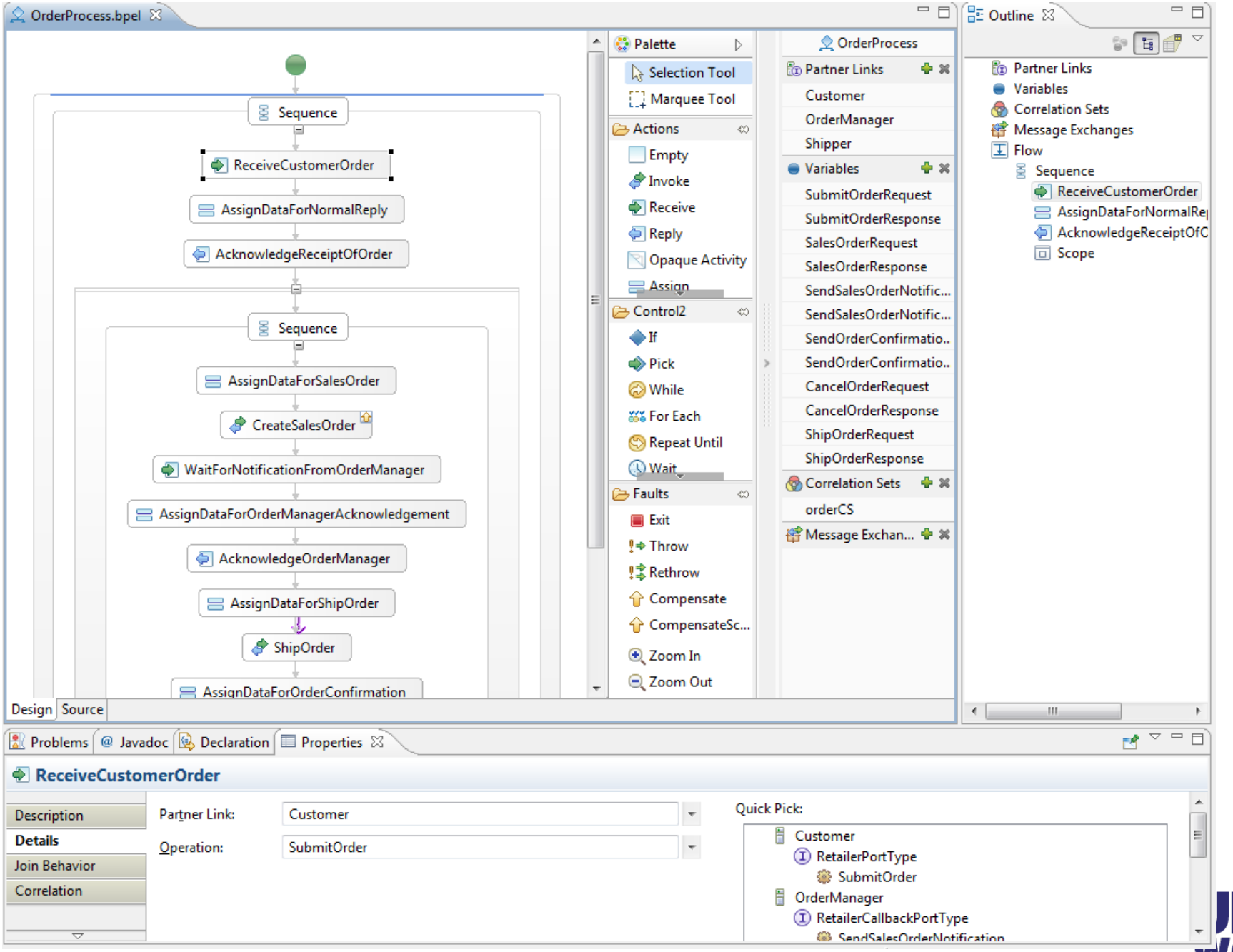
BPEL

presented by

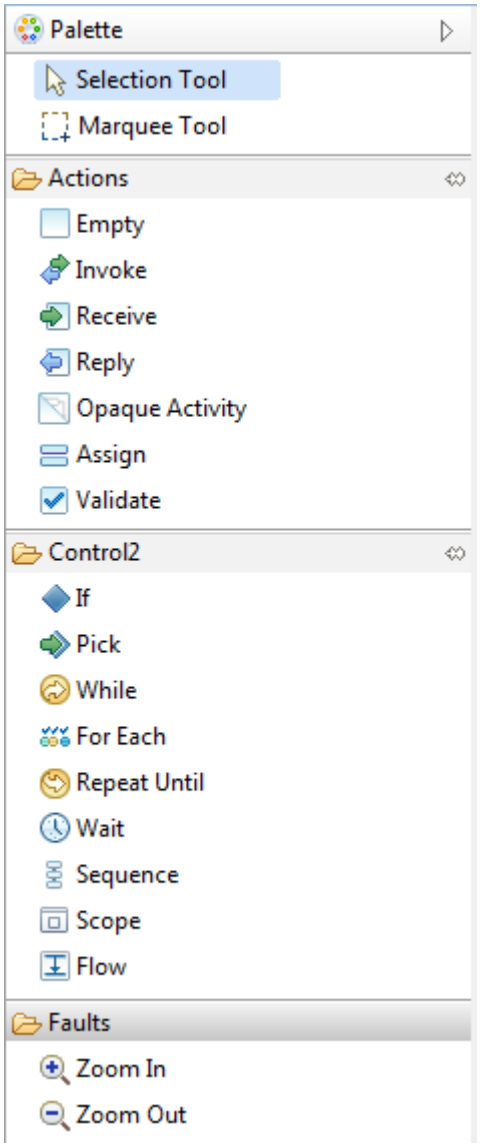
Simple BPEL Process Structure



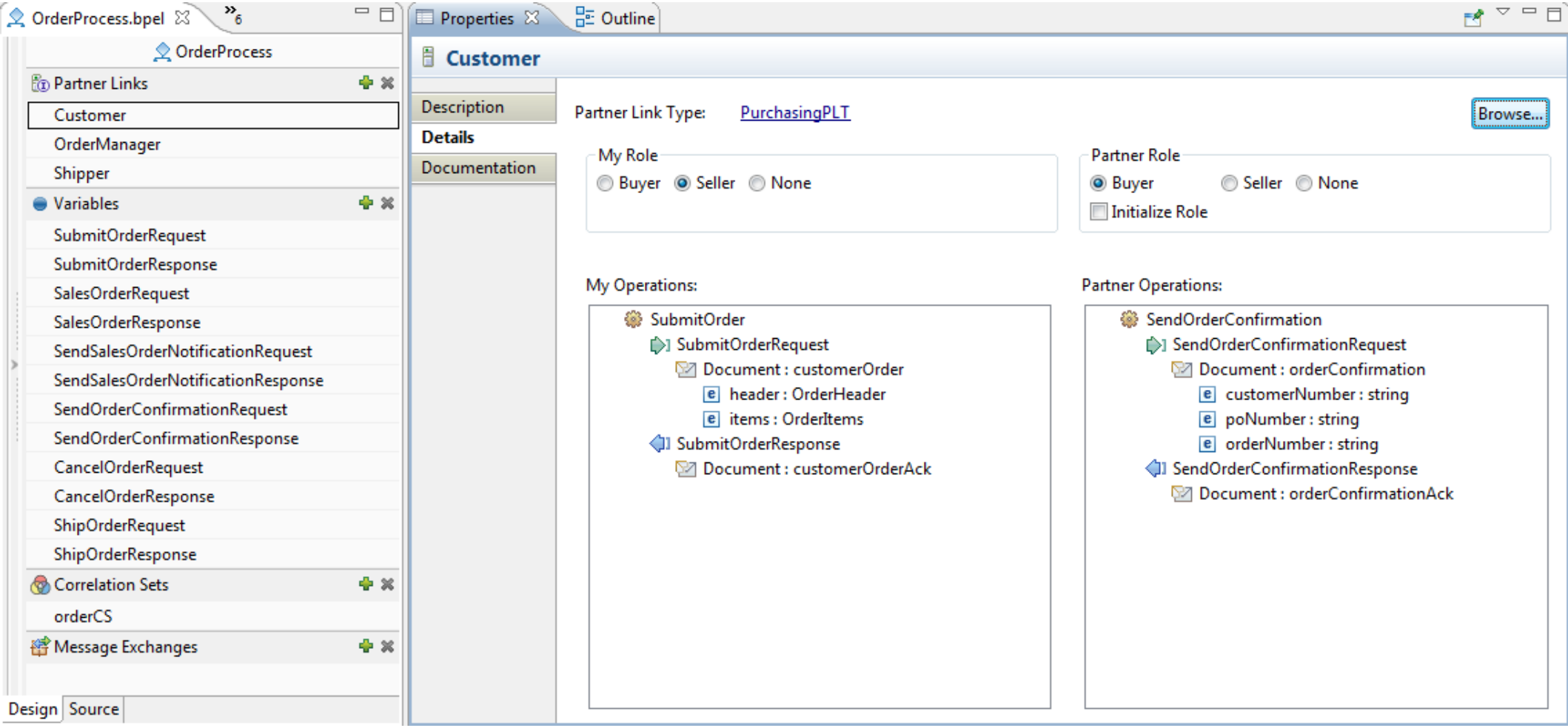
Tools – BPEL Editor



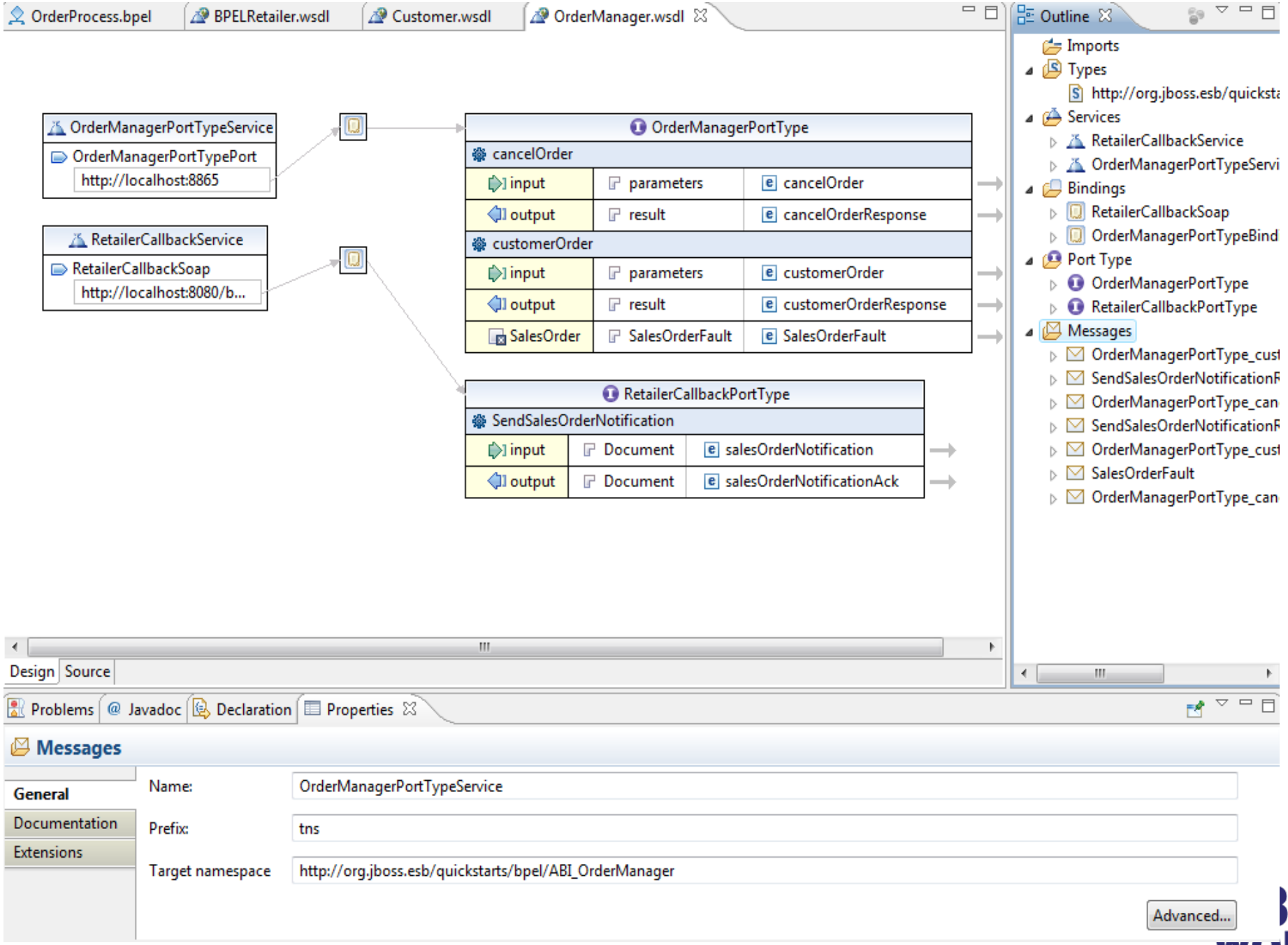
BPEL Palette



Partner Link Properties



Tools – WSDL Editor



BPEL Resources

http://www.jboss.org/riftsaw

Riftsaw Forums

WS-BPEL Primer:
http://docs.oasis-open.org/wsbpel/2.0/Primer/wsbpel-v2.0-Primer.html

WS-BPEL 2.0 Specification:
http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html

Apache Ode
http://ode.apache.org/

RiftSaw BPEL Server - JBoss Community - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://www.jboss.org/riftsaw

JBoss Community

Home | Members | Projects | Products

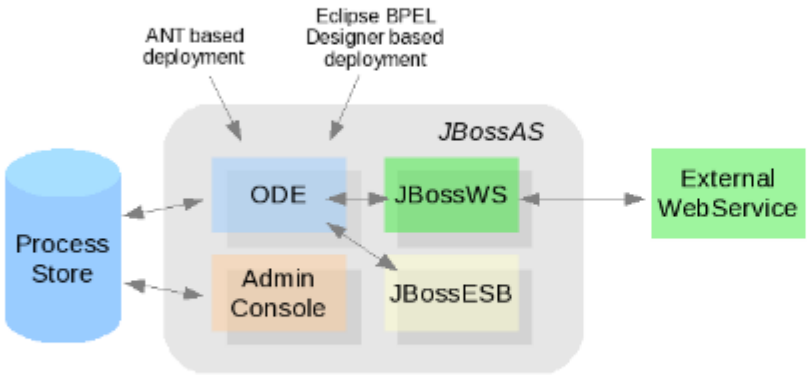
Login | Register | Cool Stuff | Search the Community

RiftSaw

Overview | Downloads | Documentation | Community | Issue Tracker | Source Code | Build | Search project pages

JBoss BPEL Server

Project Riftsaw is a WS-BPEL 2.0 engine that is optimized for the JBoss Application Server container. WS-BPEL 2.0 is an XML-based language for defining business processes that orchestrate web services.



Riftsaw supports :

Useful Links

- Apache ODE
- WS-BPEL 2.0 Specification
- Eclipse BPEL Designer (Bundled with JBoss Tools 3.1)
- Screenshot Eclipse BPEL Designer
- Roadmap
- Slides of the July 22 Webinar
- Playback of the July 22 Webinar
- RiftSaw Blog

Done

One active download (24 minutes remaining)

Professional Enterprise Support

RESTful HTTP – SOA Platform 5 (Q1 2010)

CEP – SOA Platform 5 (Technology Preview)

Infinispan – SOA Platform 5.1 or 6.0 (TBD)

BPEL – BPEL Platform 2.0 (Q1 2010)

For more information at JBoss World 2009

ESB – Kevin Conner (Newcastle)

CEP via Drools Fusion – Edson Tirelli (Toronto)

Infinispan – Manik Surtani (London)

BPEL via Riftsaw – John Graham (Boston)

QUESTIONS?

TELL US WHAT YOU THINK:
REDHAT.COM/JBOSSWORLD-SURVEY

JBoss WORLD
CHICAGO 2009

FOLLOW US:

TWITTER.COM/REDHATSUMMIT

TWEET ABOUT US:

ADD #SUMMIT AND/OR #JBOSSWORLD TO THE END OF YOUR EVENT-RELATED TWEET

presented by