



Open ESB

- **Sang Shin**
 - Java Technology Evangelist
- Sun Microsystems, Inc.
-
- **Raffaele Spazzoli**
- Imola Informatica
-



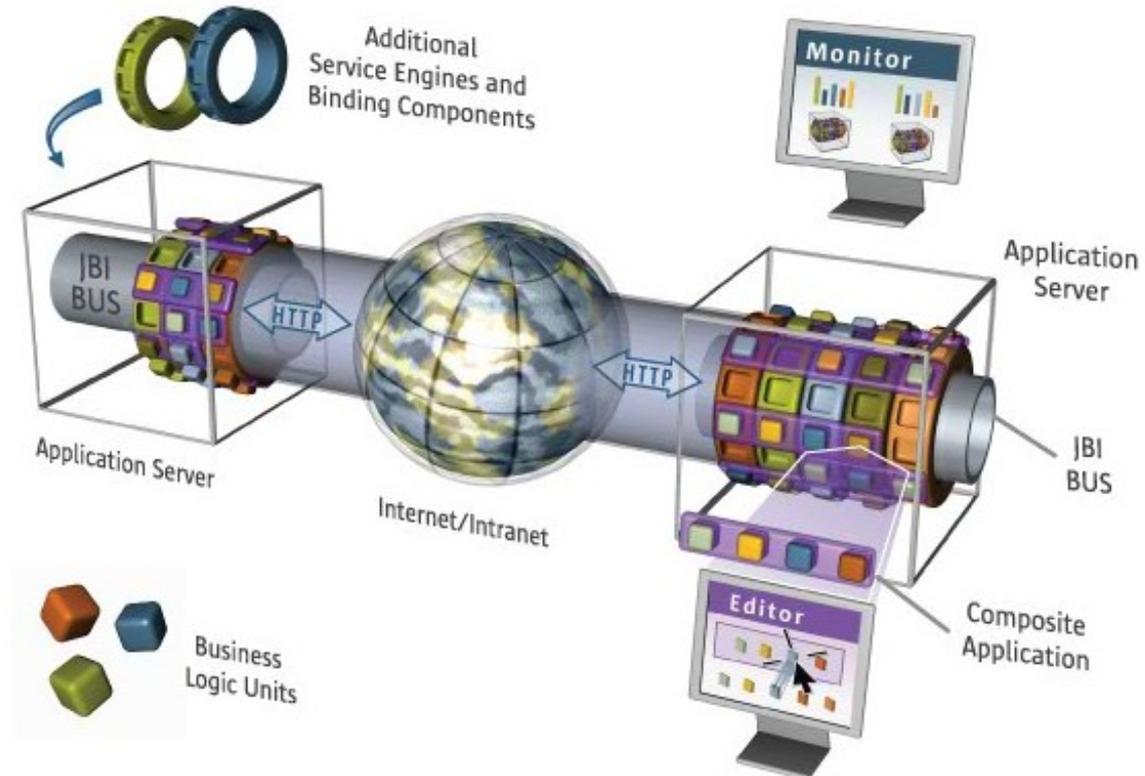
Topics

- What is Open ESB?
- What is JBI?
- JBI and GlassFish
- Usage Scenario
- Open ESB Development & Deployment Environment
- SE's and BC's (available right now)
- NetBeans support of Open ESB
- Java EE SE, IEP SE, Aspect SE, etc
- IEP (Intelligent Event Processing) SE Demo

What is Open ESB?

Open ESB

- Open Source Enterprise Service Bus runtime implemented atop the Java Business Integration (JBI) foundation
 - > <http://open-esb.org/>
- Runs within Glassfish/Sun App Server



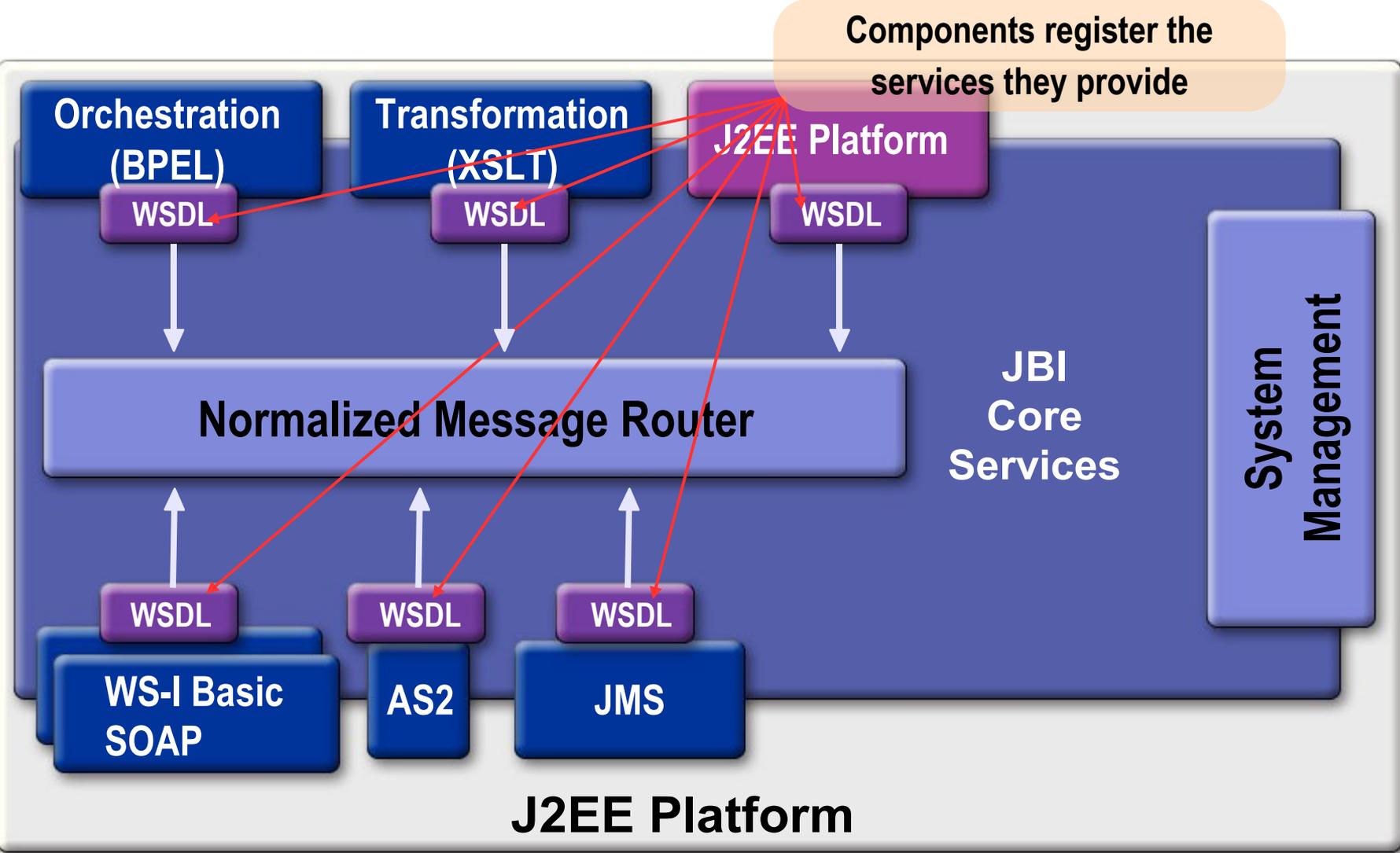
What is JBI?

What Is JBI?



- Standard “meta-container” for integrated services
- Provides for plug-in:
 - > Service Engines (SE): business logic
 - > Binding Components: communications protocols
- Standard deployment model
- Loose coupling via WSDL message exchanges between WSDL described Services

Service Provider Self-Description



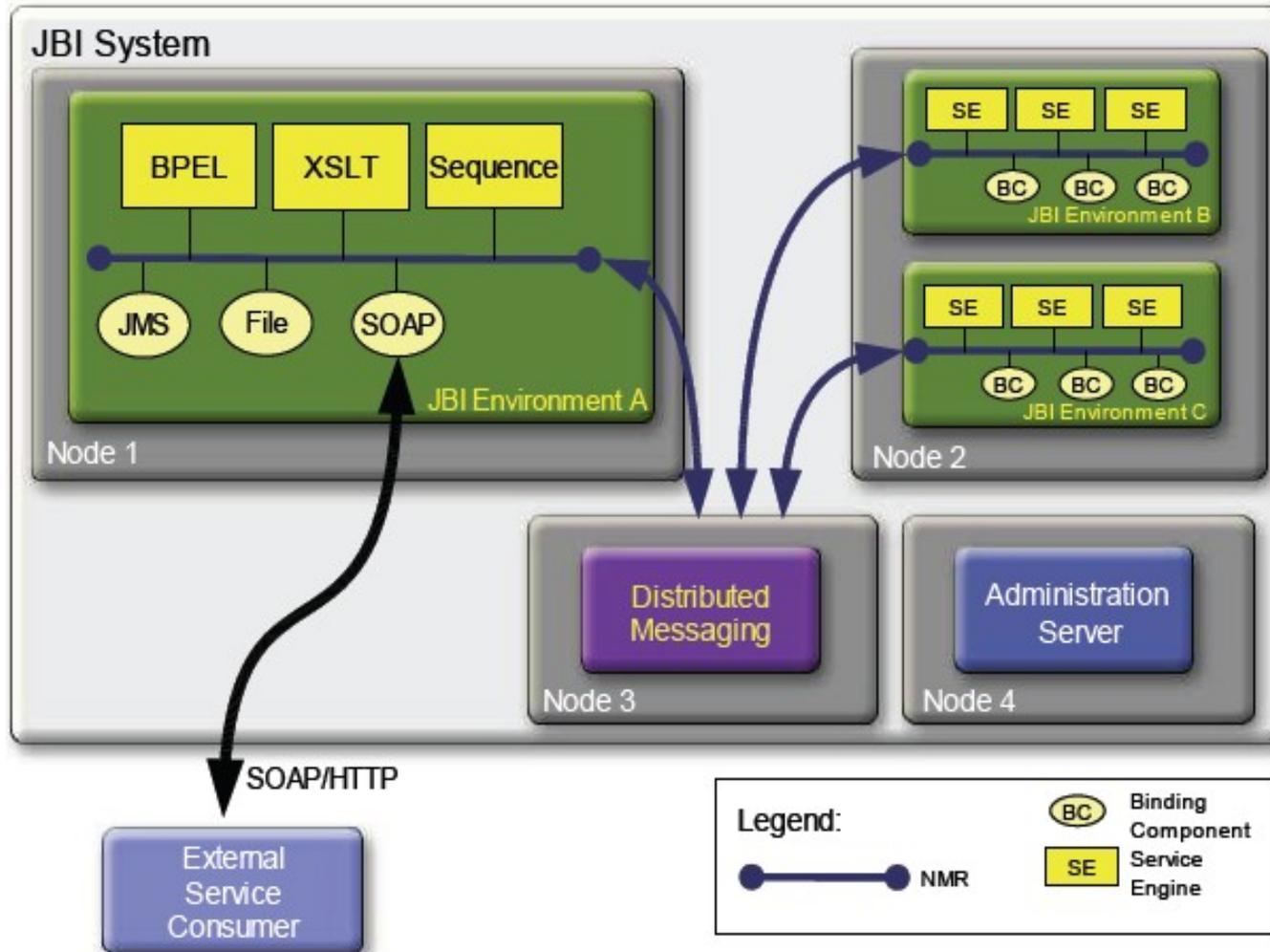
Normalized Message Router

- Key to interoperation between components
- Mediated Message Exchange
- Normalized Message
 - > Abstract Message (payload) +
 - > Message Properties (metadata)
- Message Exchange Pattern
 - > Support for simple communications primitives

Administration

- Component Life Cycle (containers)
 - > Installation
- Packaging and Deployment to Components
 - > Service Units
 - > Service Assemblies

JBI and ESBs



JBI v2

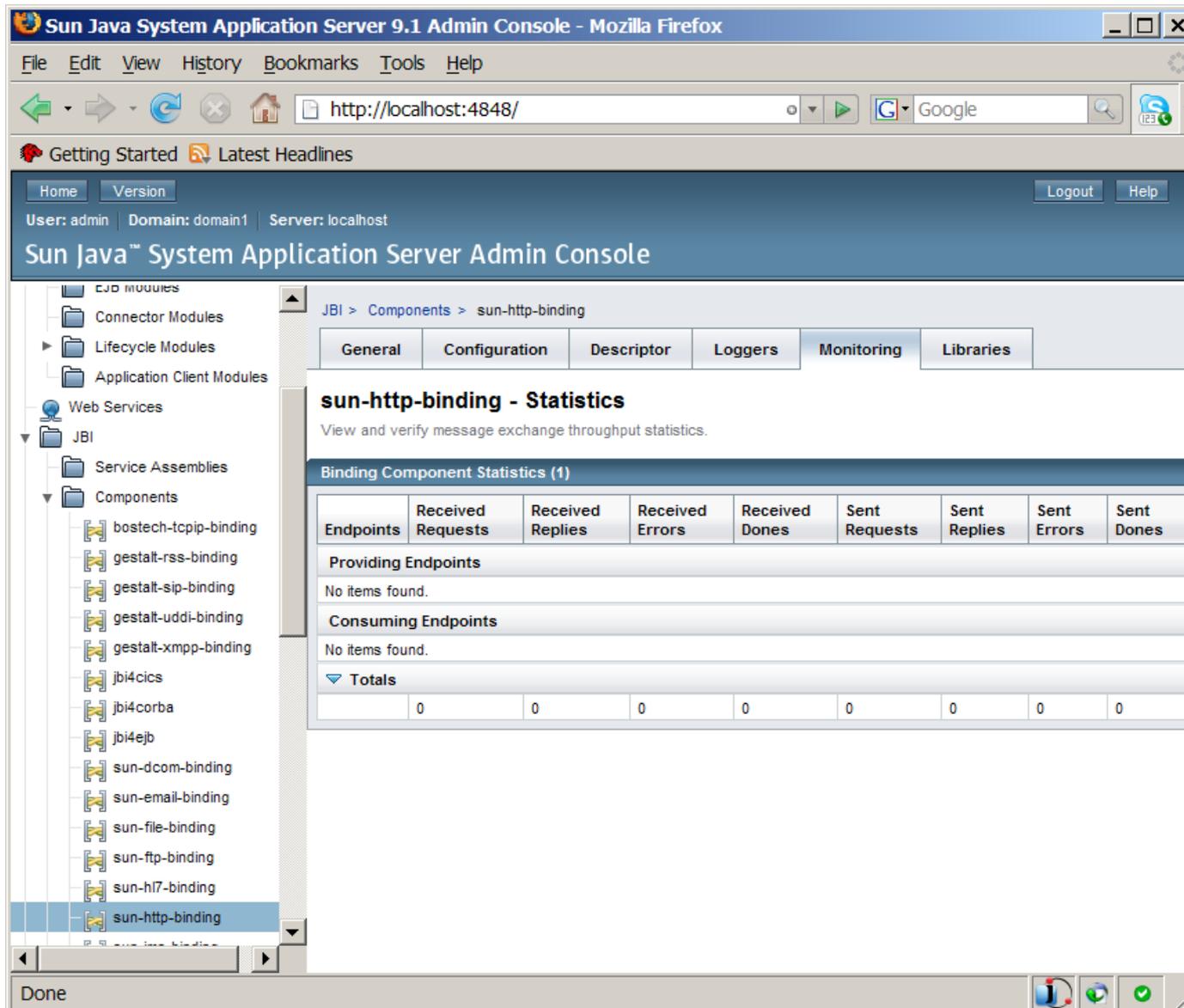
- Submitted: 2007-03-13 (JSR 312)
- Public Review: mid-2007
 - > Looking more end-users on the EG
- Focus
 - > Administration of clustered / distributed enviros
 - > Better alignment with Java EE
 - > Interceptors
 - > Policy/capabilities
 - > Greater coverage of JBI enablement of Composite Apps
 - > Choreography
 - > SCA alignment

JBI and GlassFish

JBI Support in GlassFish

- A JBI runtime has been integrated with GlassFish V2
- GlassFish admin console now supports JBI
- Java EE Service Engine act as the bridge between Java EE applications and JBI
- A Java EE application archive (ear/war/jar) can be packaged in a JBI composite application
- JBI runtime has been enhanced to adhere to the appserver clustering architecture
 - > Each instance in the appserver cluster will also have a JBI runtime in it

JBI in Admin Console



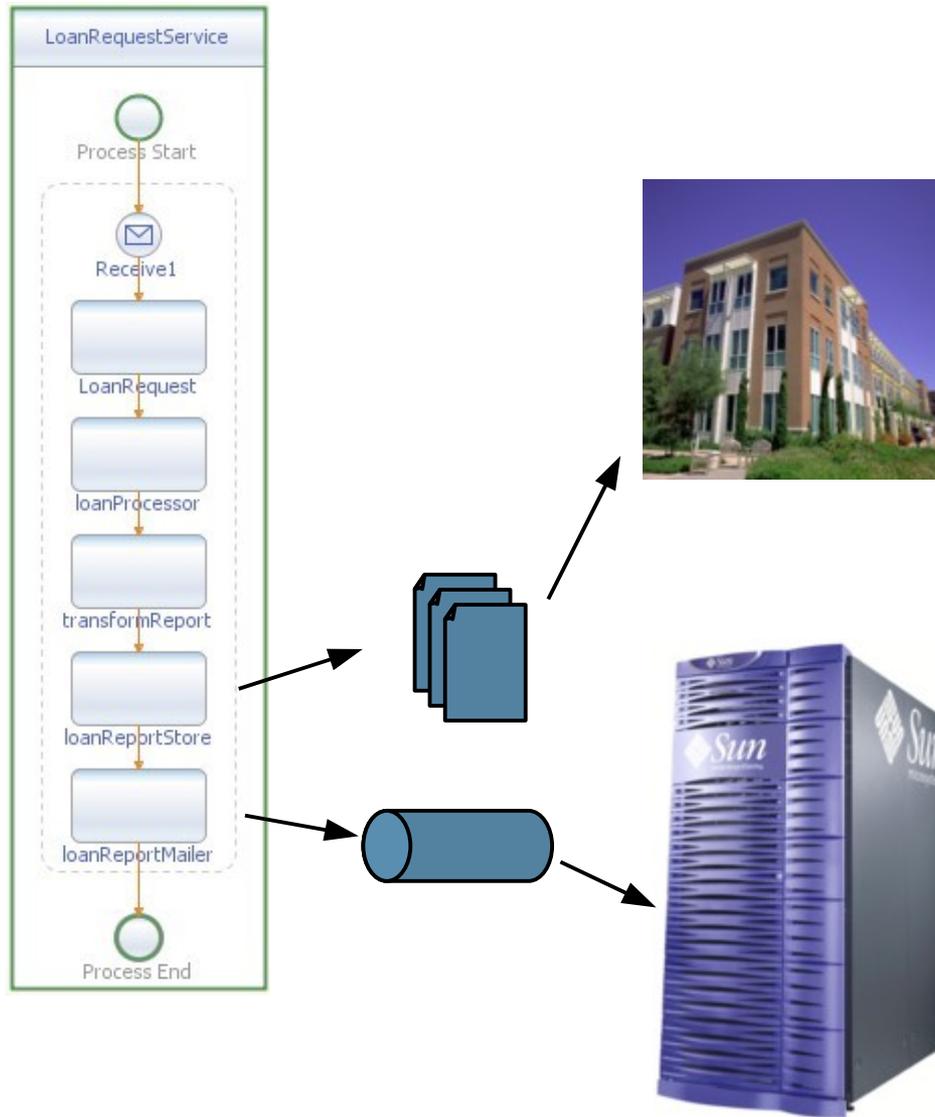
The screenshot shows the Sun Java System Application Server 9.1 Admin Console in Mozilla Firefox. The browser address bar shows `http://localhost:4848/`. The console interface includes a navigation menu on the left with categories like EJB Modules, Connector Modules, Lifecycle Modules, Application Client Modules, Web Services, and JBI. Under JBI, the 'Components' folder is expanded, listing various binding components, with 'sun-http-binding' selected.

The main content area displays the 'sun-http-binding - Statistics' page. It includes tabs for General, Configuration, Descriptor, Loggers, Monitoring, and Libraries. The 'Monitoring' tab is active, showing a table of 'Binding Component Statistics (1)'. The table has columns for Received Requests, Received Replies, Received Errors, Received Dones, Sent Requests, Sent Replies, Sent Errors, and Sent Dones. The 'Totals' row shows all values as 0.

Endpoints	Received Requests	Received Replies	Received Errors	Received Dones	Sent Requests	Sent Replies	Sent Errors	Sent Dones
Providing Endpoints								
No items found.								
Consuming Endpoints								
No items found.								
Totals								
	0	0	0	0	0	0	0	0

Usage Scenario

Usage Scenario: Loan Processing



- Loan Requestor Service:
 - > LoanRequestProcess
 - > WS-I BP
 - > BPEL Orchestration
 - > LoanProcessor
 - > JavaEE
 - > TransformReport
 - > XSLT
 - > LoanReportStore
 - > Business Partner thru FTP
 - > LoanReportMailer
 - > Legacy thru JMS

JBI-based Infrastructure

BPEL

JavaEE

XSLT

NMR

WS-I BP

JMS

File



**JB1-based
Infrastructure**

BPEL
Loan
Request
Process

JavaEE
Loan
Processor
EJB

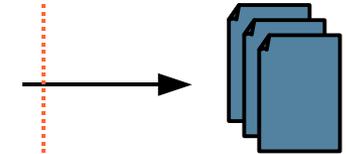
XSLT
Transform
Report

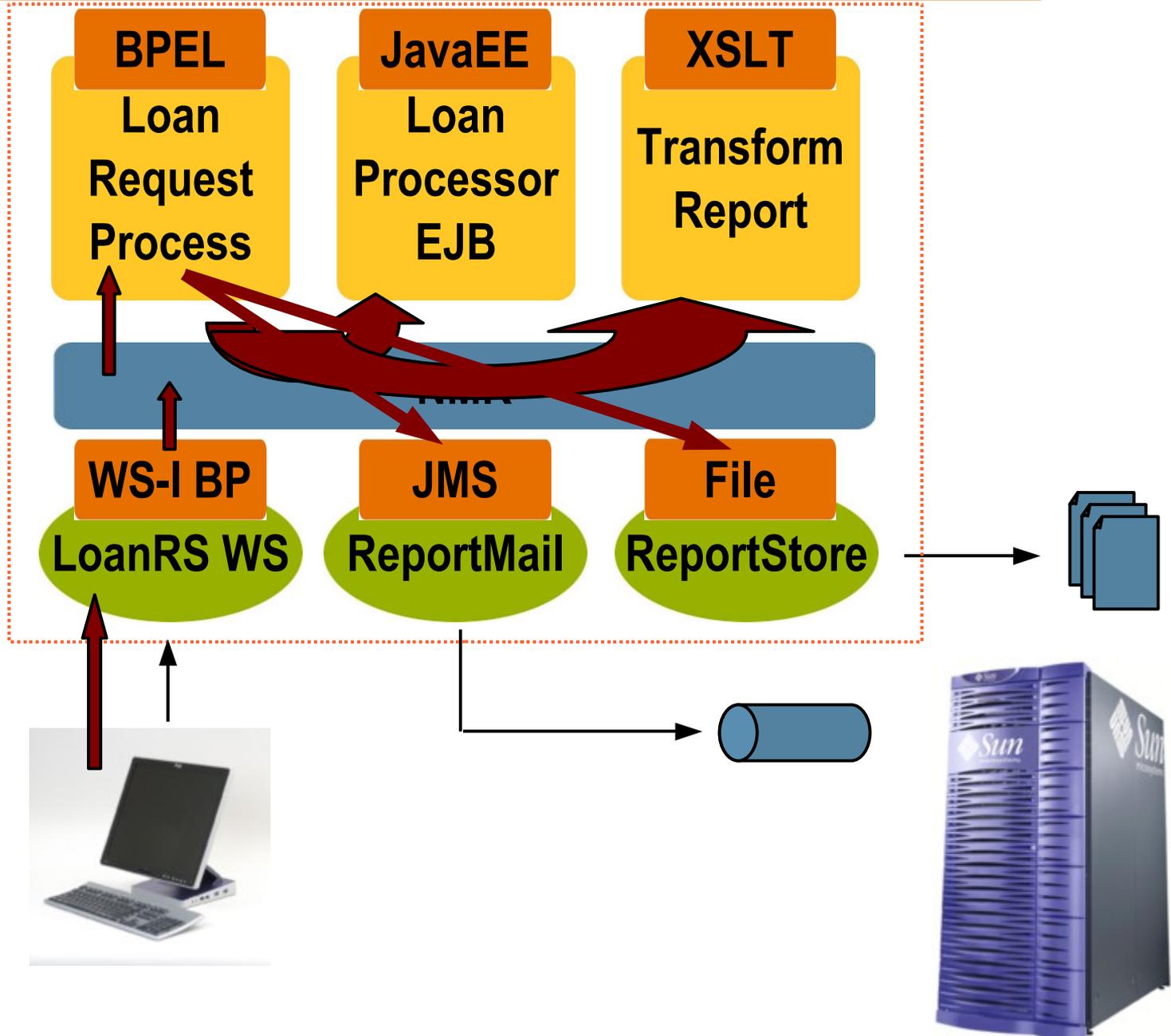
NMR

WS-I BP
LoanRS WS

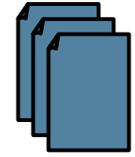
JMS
ReportMail

File
ReportStore





Architecture Refactoring



BPEL
Loan Request Service

XSLT
Transform Report

RulesEngine
Loan Processor

JavaEE
ReportStore

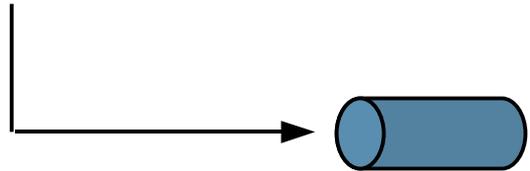
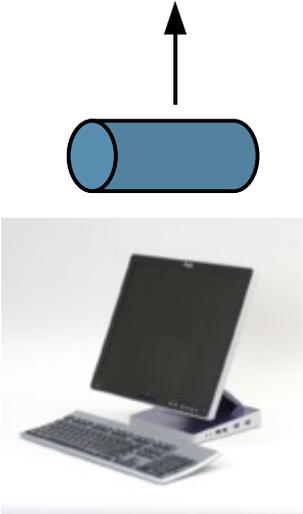
NMR

JMS
LoanRS Q

WS-I BP
LoanRS WS

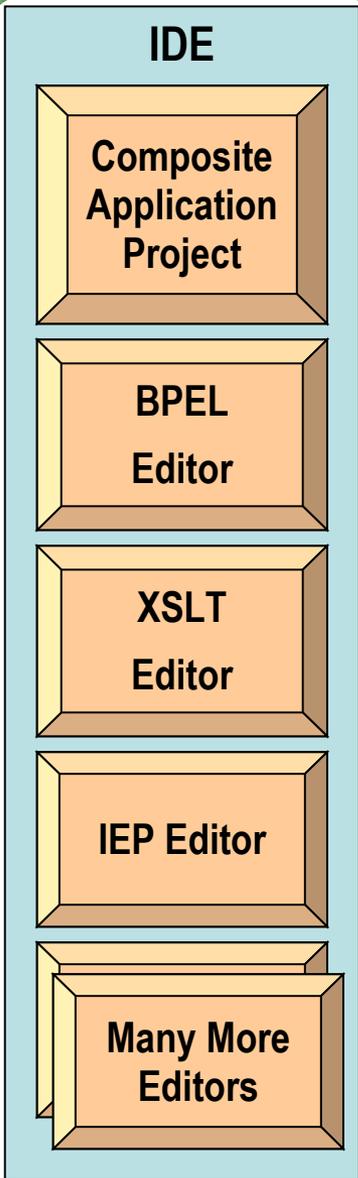
JMS
ReportMail

File

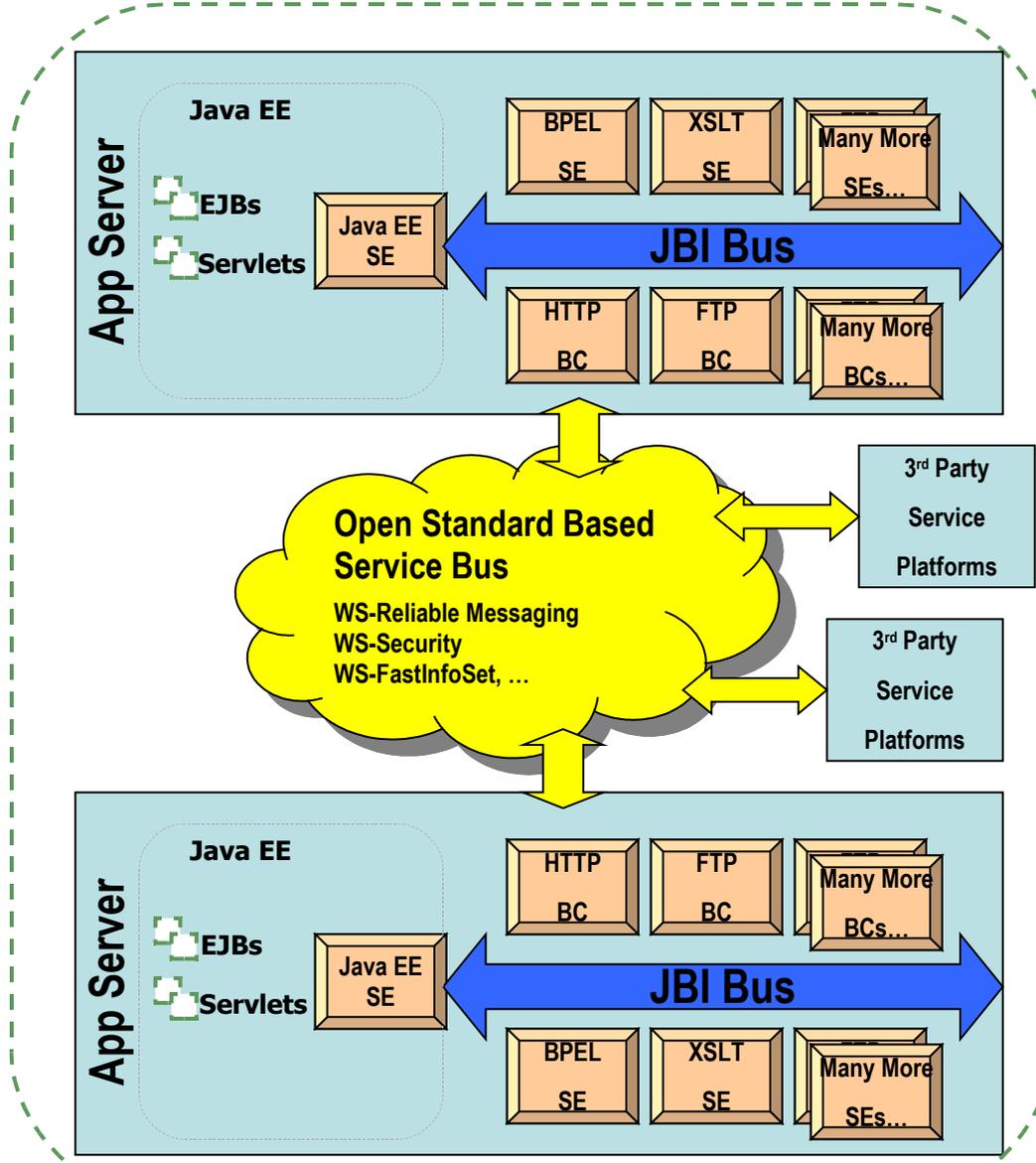


Open ESB Development & Deployment Environment

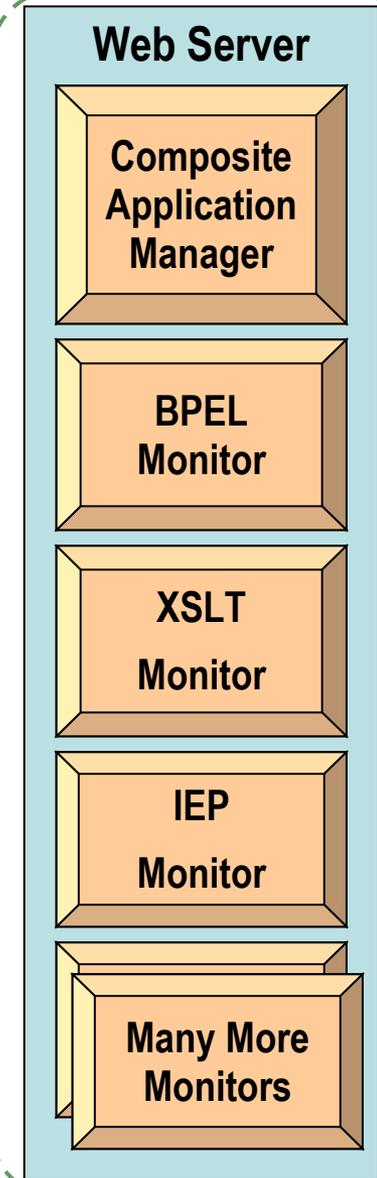
Design-Time



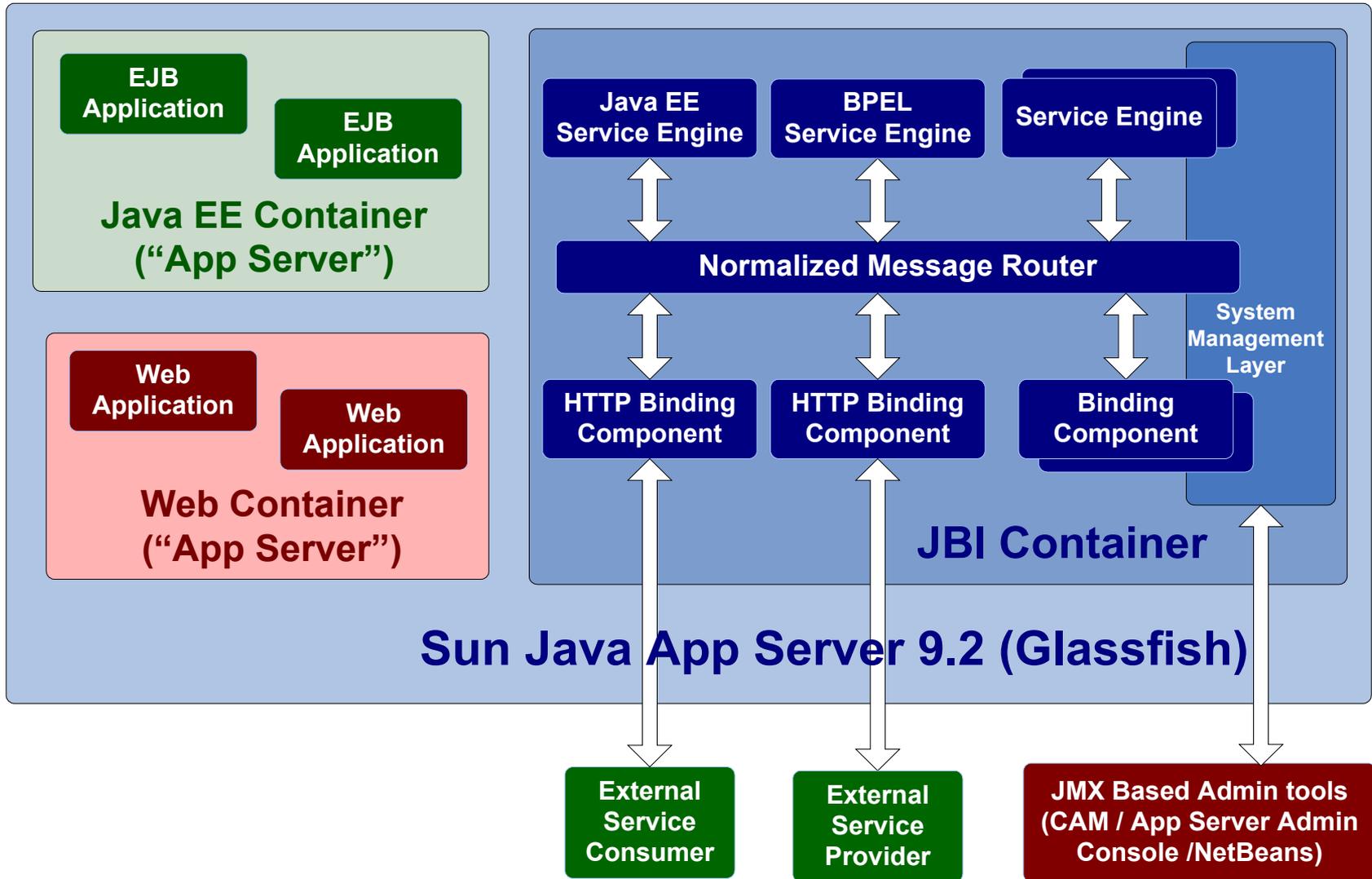
Runtime



Management



OpenESB Architecture



Service Engines (SE) & Binding Components (BC)

JBI Components

- **Service Engines**
 - > BPEL SE
 - > XSLT SE
 - > JavaEE SE
 - > IEP SE
 - > ETL SE
 - > SQL SE
 - > Workflow SE
- **Binding Comps**
 - > MQSeries BC
 - > HL7 BC
 - > SAP BC
 - > SMTP BC
 - > HTTP BC
 - > JMS BC
 - > File BC
 - > CICS BC
 - > DCOM BC
 - > CORBA BC
 - > ...
- **Other**
 - > Clustering
 - > CASA
 - > JBI Mock
 - > WSIT Tech
- **In Progress**
 - > CAM
 - > Aspect SE
 - > Encoding SE
 - > Rules SE
 - > Scripting SE

open-esb.dev.java.net/Components.html

open-esb: Components - Mozilla Firefox

File Edit View History Bookmarks Tools Help

https://open-esb.dev.java.net/Components.html

Getting Started Latest Headlines

The Source for Java Technology Collaboration
[Login](#) | [Register](#)

My pages
Projects
Communities
java.net

Projects > java-enterprise > open-esb

Open ESB Project

- [Home](#)
- [All Downloads](#)
- [Architecture](#)
- [Components](#)
- [Documentation](#)
- [FAQs](#)
- [Licensing](#)
- [Get Help](#)

Project Tools

- [Announcements](#)
- [Build Instructions](#)
- [Developer Tools](#)
- [Source Code \(CVS\)](#)
- [Fisheye of Source](#)
- [Issue Tracker](#)

Go to Issue#

Open ESB Community

- [Community Home](#)
- [Developer Wiki](#)
- [News](#)
- [Events](#)
- [Blogs](#)
- [Forums](#)
- [Mailing Lists](#)
- [Partners](#)
- [Developers](#)
- [Get Involved](#)

Open ESB Components

The following list of components are currently available at the Open JBI Components project. In this page you can find a component by their function, their solution or by their name. Clicking on a component link will provide details on how they are built, what their current capabilities are and the future plans around these components.

[Learn More about Open JBI Components Project](#)

[View the JBI Community Wiki's components](#)

By Function

Application Mashup
[AOSD](#)

Aspects
[Aspect SE](#)

Communications
These components provide different methods of communicating with other components of the Enterprise Service Bus.

- [ADABAS Natural](#)
- [CICS BC](#)
- [CORBA BC](#)
- [DCOM BC](#)
- [File BC](#)
- [FTP BC](#)
- [HL7 BC](#)
- [HTTP BC](#)
- [SIP BC](#)
- [UDDI BC](#)

By Solution

Healthcare
These components provide solutions for the healthcare industry.

[HL7 BC](#)

Mainframe
These components provide mainframe solutions to integration.

- [CICS BC](#)
- [IMS BC](#)

Telecommunications
These components provide solutions for the telecommunications industry.

- [CORBA BC](#)
- [SIP BC](#)
- [SNMP BC](#)
- [YMDP BC](#)

By Name

Alphabetical Listing
If you know the name of the project you are looking for, find it here.

Applications
[Mural](#)

Binding Components

- [CICS BC](#)
- [CORBA BC](#)
- [DCOM BC](#)
- [eMail BC](#)
- [File BC](#)
- [FTP BC](#)
- [HL7 BC](#)
- [HTTP BC](#)
- [IMS BC](#)
- [JDBC BC](#)
- [JMS BC](#)
- [LDAP BC](#)
- [MQ Series BC](#)
- [MSMQ BC](#)
- [RSS BC](#)

Search

Latest Downloads

[Open ESB 2.0 Preview](#)

Featured Partner

Service Your Data

XCalia makes it easy for enterprises to build composite applications in SOAs while reusing valuable information resources, optimizing IT investments and reducing operating costs.

[Read more...](#)

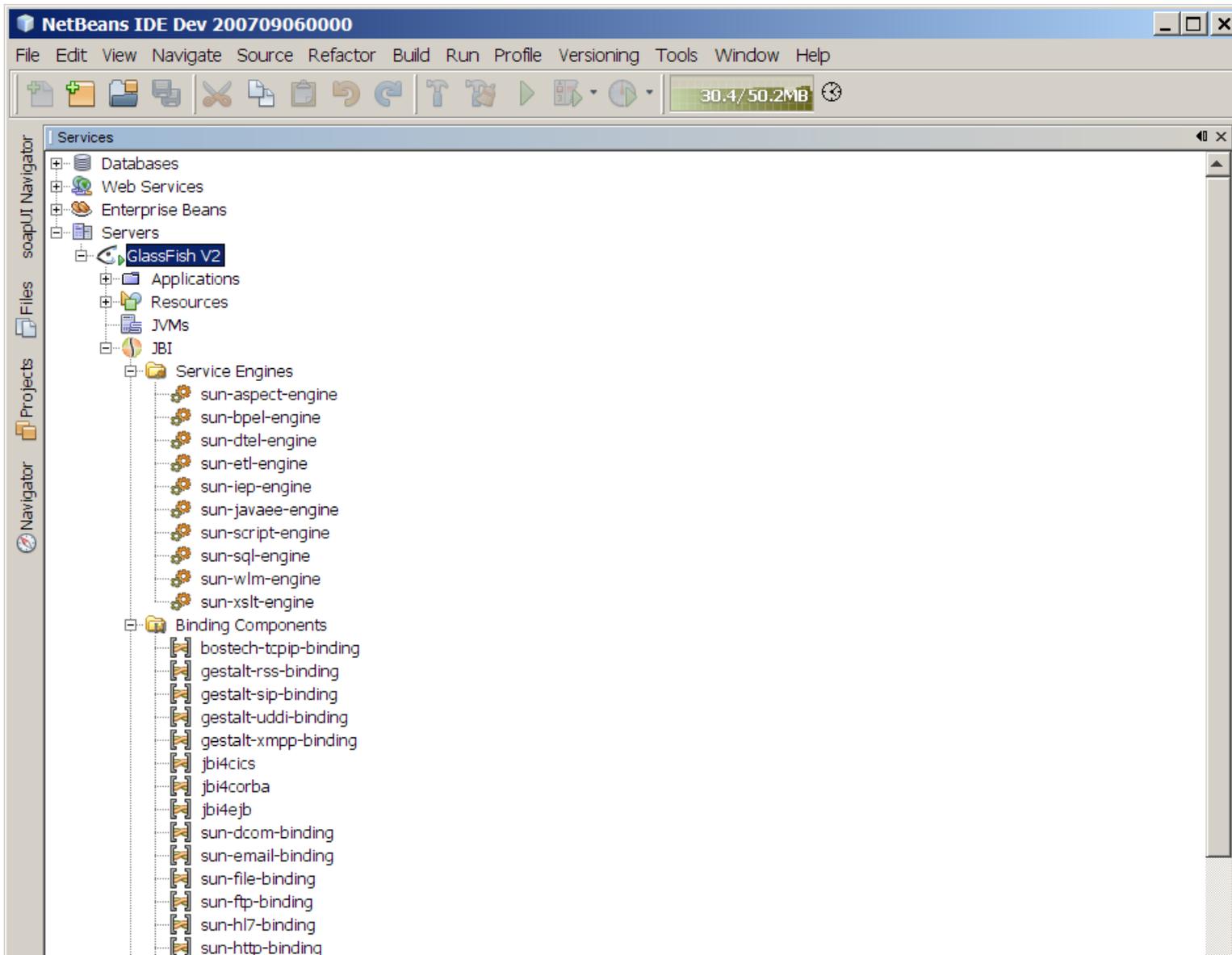
[See All Partners](#)

[Become a Partner Today!](#)

Most Read Blogs

- [Fred Aabedi's Blog](#)
- [Prakash Aradhya's Blog](#)
- [Keith Babo's Blog](#)

Open ESB Package Ships Many SE's/BC's

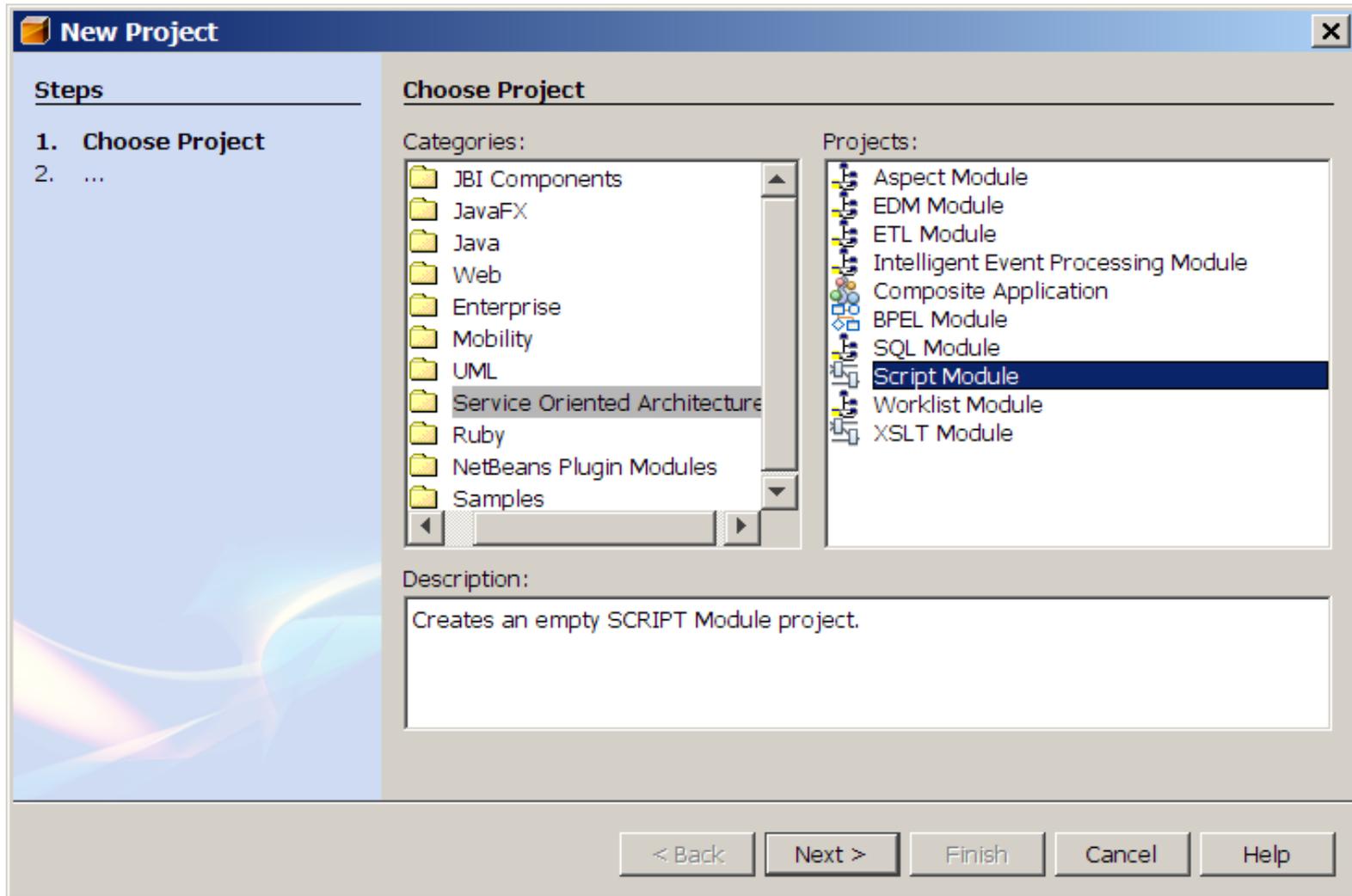


NetBeans Support of Open ESB

Types of SOA “NetBeans” Projects

- When creating a composite application, you typically use the following types of SOA “NetBeans” projects:
 - > BPEL Module project (NetBeans 6.0)
 - > XSLT Module project (NetBeans 6.0)
 - > SQL Module project (NetBeans 6.0)
 - > Composite Application project (NetBeans 6.0)
 - > IEP Module project (OpenESB package)
 - > Worklist Module project (OpenESB package)
 - > ETL (Extract, Transform, and Load) (OpenESB package)
 - > EDM (Enterprise Data Mashup) (OpenESB package)
 - > And more

Types of SOA “NetBeans” Projects



BPEL Module Project

- BPEL Module project is a group of source files which includes
 - > XML Schema (*.xsd) files
 - > WSDL files
 - > BPEL files
- Within a BPEL Module project, you can author a business process compliant with the WS-BPEL 2.0 language specification.
- Will be added to a Composite application as a JBI module

Composite Application Project

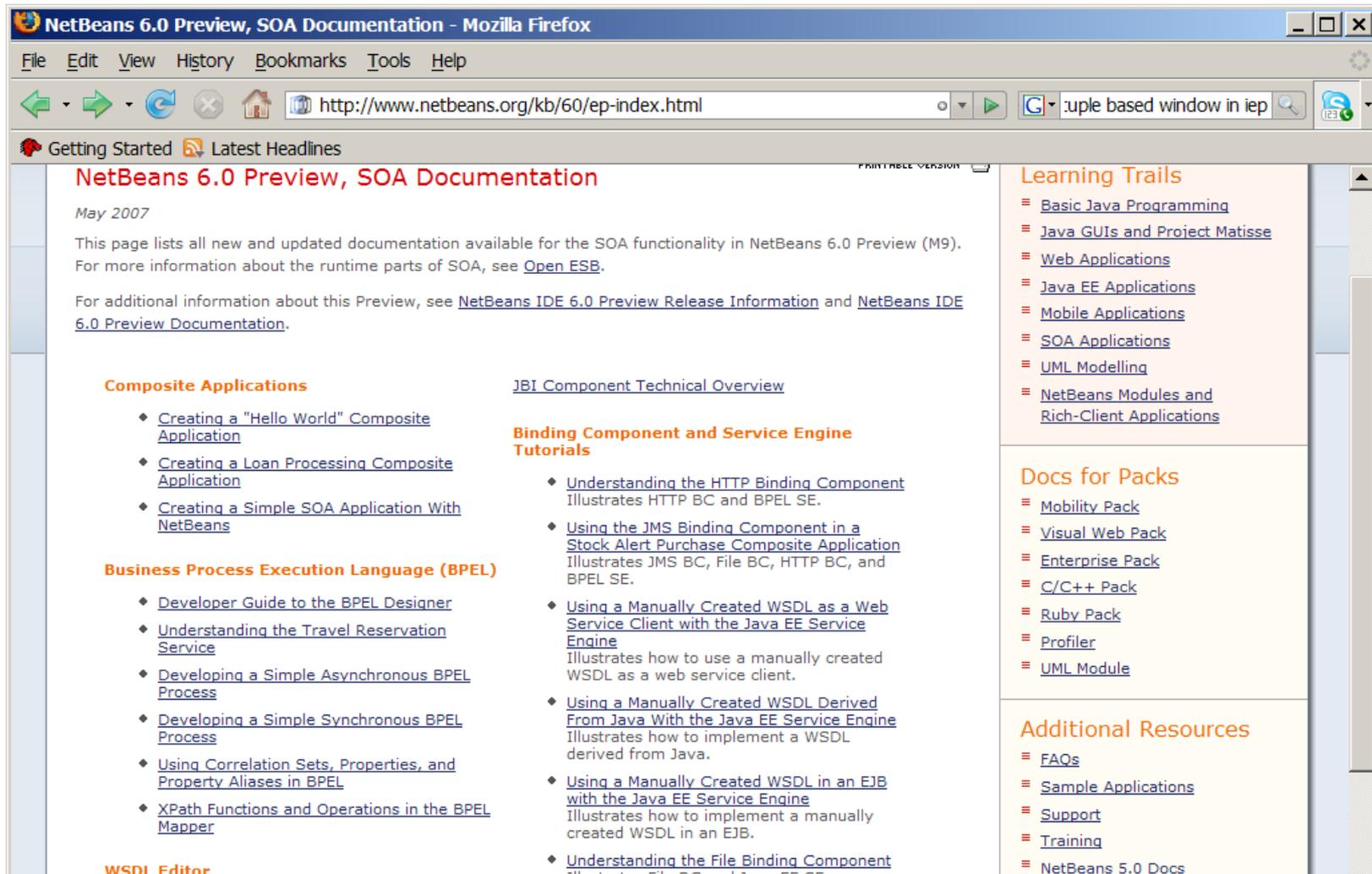
- Composite Application project is a project whose primary purpose is to assemble a deployment unit for the Java Business Integration (JBI) server
 - > BPEL Module projects must be added to a Composite Application project in order to be deployed to the BPEL runtime.
- The Composite Application Project can also be used to create and execute test cases that can then be run, in JUnit fashion, against the deployed BPEL processes.

Composite Application Project

- With a Composite Application project, you can:
 - > Assemble an application that uses multiple project types (BPEL, XSLT, IEP, SQL, etc.)
 - > Configure external/edge access protocols (SOAP, JMS, SMTP, and others)
 - > Build JBI deployment packages
 - > Deploy the application image to the target JBI server
 - > Monitor the status of JBI server components and applications

Lots of Step by Step Tutorials

- <http://www.netbeans.org/kb/60/ep-index.html>



NetBeans 6.0 Preview, SOA Documentation
May 2007

This page lists all new and updated documentation available for the SOA functionality in NetBeans 6.0 Preview (M9). For more information about the runtime parts of SOA, see [Open ESB](#).

For additional information about this Preview, see [NetBeans IDE 6.0 Preview Release Information](#) and [NetBeans IDE 6.0 Preview Documentation](#).

Composite Applications

- ◆ [Creating a "Hello World" Composite Application](#)
- ◆ [Creating a Loan Processing Composite Application](#)
- ◆ [Creating a Simple SOA Application With NetBeans](#)

Business Process Execution Language (BPEL)

- ◆ [Developer Guide to the BPEL Designer](#)
- ◆ [Understanding the Travel Reservation Service](#)
- ◆ [Developing a Simple Asynchronous BPEL Process](#)
- ◆ [Developing a Simple Synchronous BPEL Process](#)
- ◆ [Using Correlation Sets, Properties, and Property Aliases in BPEL](#)
- ◆ [XPath Functions and Operations in the BPEL Mapper](#)

WSDL Editor

[JBI Component Technical Overview](#)

Binding Component and Service Engine Tutorials

- ◆ [Understanding the HTTP Binding Component](#)
Illustrates HTTP BC and BPEL SE.
- ◆ [Using the JMS Binding Component in a Stock Alert Purchase Composite Application](#)
Illustrates JMS BC, File BC, HTTP BC, and BPEL SE.
- ◆ [Using a Manually Created WSDL as a Web Service Client with the Java EE Service Engine](#)
Illustrates how to use a manually created WSDL as a web service client.
- ◆ [Using a Manually Created WSDL Derived From Java With the Java EE Service Engine](#)
Illustrates how to implement a WSDL derived from Java.
- ◆ [Using a Manually Created WSDL in an EJB with the Java EE Service Engine](#)
Illustrates how to implement a manually created WSDL in an EJB.
- ◆ [Understanding the File Binding Component](#)

Learning Trails

- ◆ [Basic Java Programming](#)
- ◆ [Java GUIs and Project Matisse](#)
- ◆ [Web Applications](#)
- ◆ [Java EE Applications](#)
- ◆ [Mobile Applications](#)
- ◆ [SOA Applications](#)
- ◆ [UML Modelling](#)
- ◆ [NetBeans Modules and Rich-Client Applications](#)

Docs for Packs

- ◆ [Mobility Pack](#)
- ◆ [Visual Web Pack](#)
- ◆ [Enterprise Pack](#)
- ◆ [C/C++ Pack](#)
- ◆ [Ruby Pack](#)
- ◆ [Profiler](#)
- ◆ [UML Module](#)

Additional Resources

- ◆ [FAQs](#)
- ◆ [Sample Applications](#)
- ◆ [Support](#)
- ◆ [Training](#)
- ◆ [NetBeans 5.0 Docs](#)

BPEL SE

BPEL SE

- Standards
 - > BPEL 2.0 (subset)
 - > WSDL1.1
- BPEL SE Configuration
 - > num threads
 - > persistence
 - > failover
- BPEL Editor
 - > BPMN
 - > Debugger

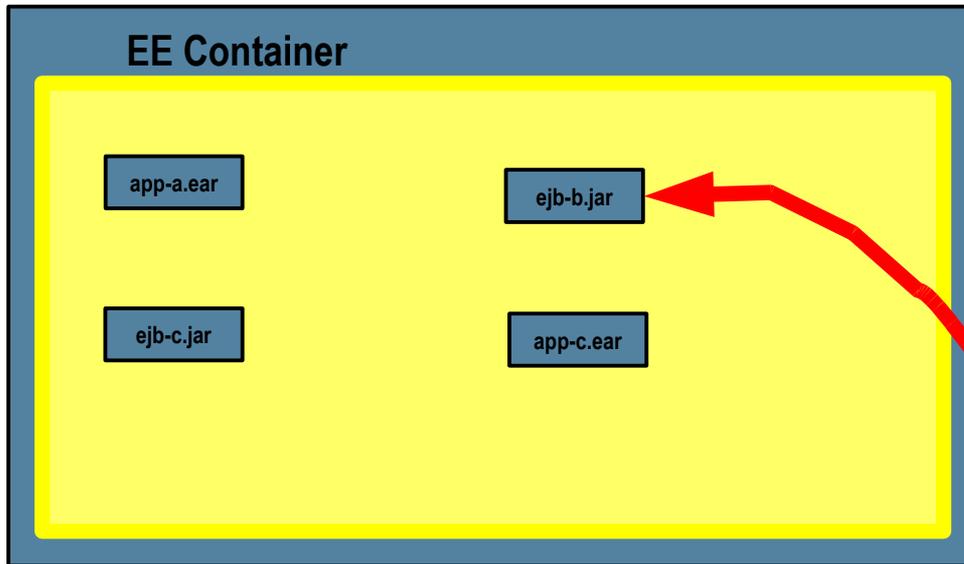
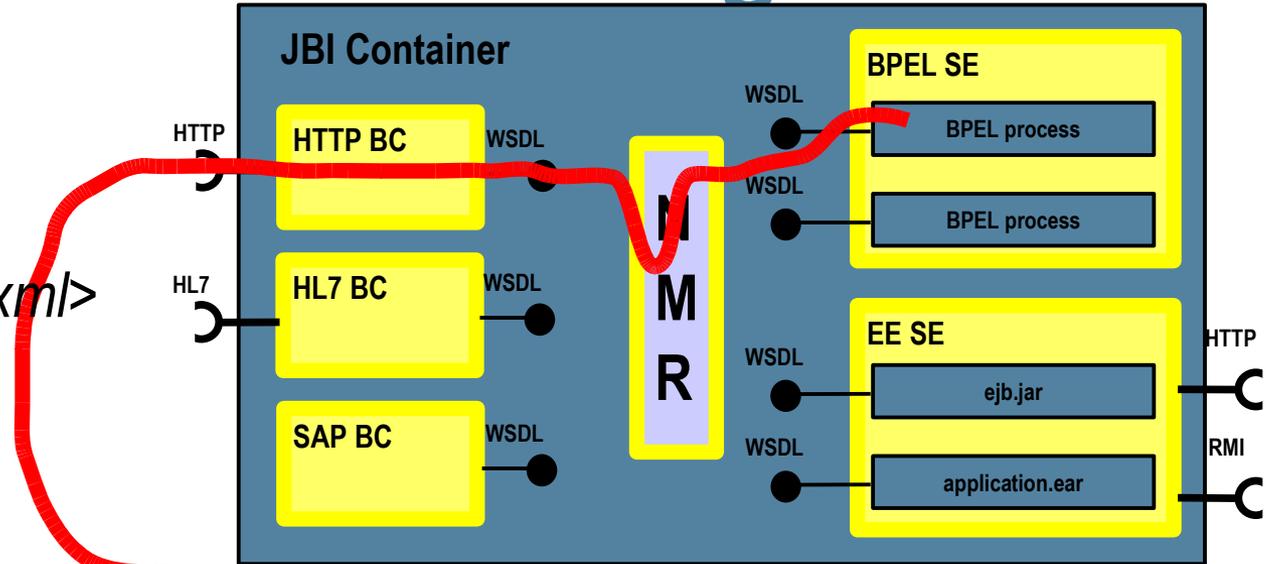
Java EE SE

JavaEE SE

- Ideal place to execute complex business logic
- Bridge between JavaEE container and JBI container
- Provides support for
 - > Transactions
 - > Resource Pooling
 - > Security
- Code re-use – Invoke your EJBs/web applications from OpenESB components (BPEL SE)
- Ability to expose your EJB/Web applications to multiple transports (using BCs) – just add bindings to your WSDL

Scenario 1: Remote through HTTP BC

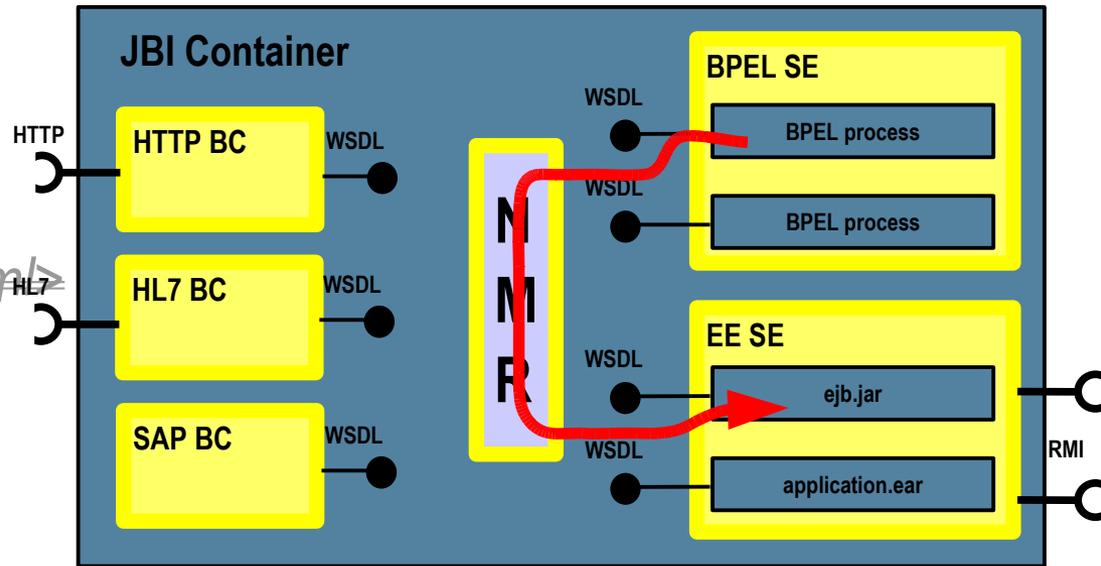
- BPEL
- NMR
- HTTP BC
- *Marshall DOM to <xml>*
- SOAP/HTTP
- Network layer



- Network layer
- SOAP/HTTP
- JAXWS
- *Unmarshal <xml>*
- *Create DOM*
- JAXB
- WS.helloWorld(name)

Scenario 2: Local through NMR

- BPEL
- NMR
- ~~HTTP BC~~
- ~~Marshall DOM to <xml>~~
- ~~SOAP/HTTP~~
- ~~Network layer~~
- ~~SOAP/HTTP~~
- JAXWS
- ~~Unmarshal <xml>~~
- ~~Create DOM~~
- JAXB
- WS.helloWorld(name)



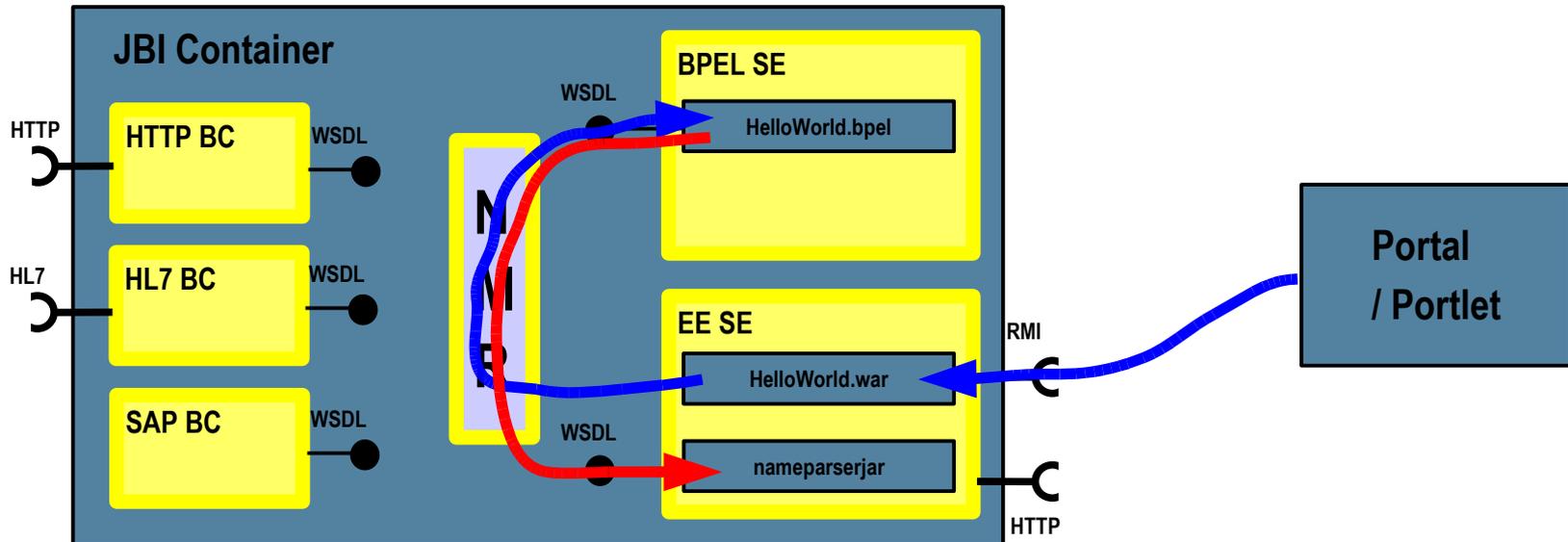
Advantages:

- > Performance
- > Transaction propagation
- > Security context propagation

Likewise: EJB to BPEL

Scenario: Portal + EE + BPEL

- Portlet gets name, invokes WAR which calls BPEL to orchestrate process
- BPEL activity requires complex business logic
 - > executes faster in EJB right

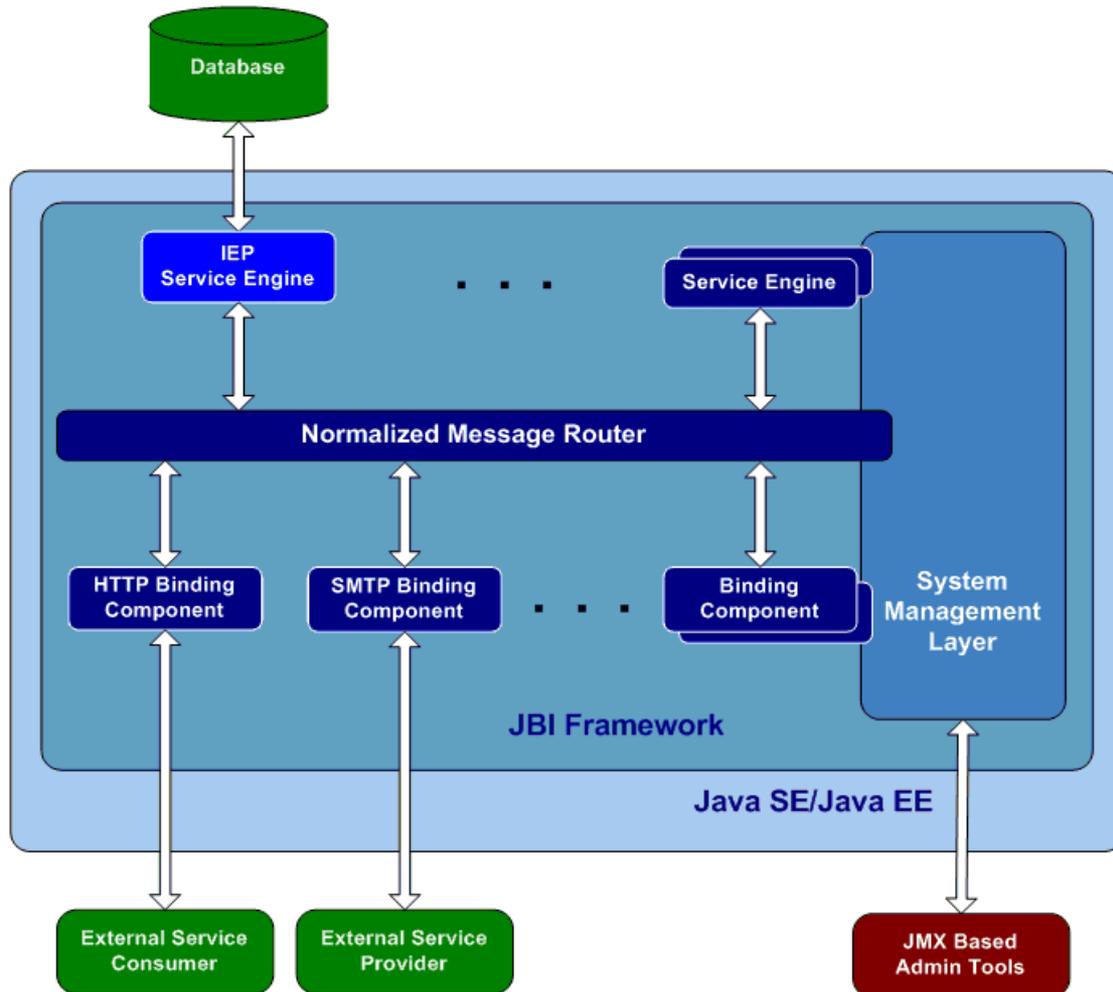


Intelligent Event Processing (IEP) SE

Intelligent Event Processing (IEP) SE

- Handles real time events and process them to higher level events which then can be used for further analysis or monitoring.
 - > Aggregation
 - > Filtering
 - > Correlation
 - > Partition
- Provides real time event notifications and triggers
 - > Update database in realtime

IEP SE



Usage Scenario

- Many modern applications require long-running, or continuous, queries over continuous unbounded streams of data.
- The need exists to detect business-critical issues as they happen, and to route, filter and pre-process data continuously over an indeterminate period of time.
- Event processing involves the continuous processing and analysis of high volume, high-speed data streams from inside and outside an organization.

IEP Operators

- **Input**
 - > Stream-Input
 - > Table-Input
- **Output**
 - > Stream-Output
 - > Relation-Output
 - > Table-Output
- **Correlation and Filtering**
 - > Stream-Project-and-Filter
 - > Tuple-Serial-Correlation
 - > Relation-Map
- **Aggregator**
 - > Time-Based-Aggregator
 - > Tuple-Based-Aggregator
 - > Relation-Aggregator.
- **Stream Converter**
 - > Tuple-Based-Window
 - > Time-Based-Window
 - > Attribute-Based-Window
 - > Partitioned-Window
- **Relation Converter**
 - > Insert-Stream
 - > Delete-Stream
 - > Relation-Stream
 - > Table
- **Relation Operator**
 - > Distinct
 - > Union
 - > Union-All
 - > Minus

IEP Support in NetBeans

The screenshot displays the NetBeans IDE interface for developing an IEP. The main workspace shows a complex data flow graph for 'zmart.iep'. The graph starts with a 'Sensor' input that branches into several paths. One path goes through a 'Window10Seconds' operator to a 'SignalOlderThan10Seconds' operator. Another path goes through a 'Window1SignalPerProduct' operator to a 'LatestSignalPerProduct' operator. A third path goes through a 'Window10Hours' operator to a 'Payment' operator. These paths converge into a 'ProductWithDeadEmitter' operator, which then connects to an 'AlertSecurity' output. Another path from the 'Sensor' goes through a 'ProductType' operator to a 'StolenProduct' operator, which also connects to an 'ExitAlarm' output. The left sidebar contains a palette of operators categorized into Aggregator, Correlation and Filter, Input, Output, Relation Converter, Relation Operator, and Stream Converter.

Finished building build.xml (test).

Intelligent Event Processing (IEP) SE Demo

You can try this demo yourself!
<http://www.javapassion.com/hands-onlabs/openesbiep/>

Demo Scenario

- External program keep sending stock quote data stream events to the IEP (through JBI)
 - > The IEP receives the stock quote stream as real time events
- The IEP send notifications to the database and the database gets updated continuously

Steps to follow

1. Create IEP module project
 - Create *quotes.iep*
 - Generate *quotes.wsdl*
2. Create a Composite application
 - Add IEP module to the Composite application
3. Run the test application that sends stock quote stream to the composite application

Aspect SE

Aspect SE

- Aspects help to encapsulate cross-cutting expressions in one place.
- By applying an Advice, at various points in an application called Join-Points, Aspects can alter the behavior of the non-aspect parts of a software application.
- There are two types of aspect patterns that are addressed:
 - > Gateway Pattern
 - > Aspect-Weaving Pattern

CASA

CompApp Service Assembly Editor

NetBeans IDE 5.5.1 070201 - CompositeApp8

File Edit View Navigate Source Refactor Build Run CVS Tools Window Help

Project: CompositeApp8
 Files: SynchronousClient, SynchronousServer

Navigator - casaPort1
 CASA Logical View
 CASA
 WSDL Ports
 casaPort1 name="casaPort1"
 casaPort1 Binding="cas"
 file:address
 casaBinding1 PortType
 portClient name="portClient"
 Service Units
 Service Unit name="Synchr"
 Service Unit name="Synchr"
 Connections
 portClient <-> partnerlinkty
 partnerlinktyperole1_partn

WSDL Ports: SOAP portClient, FILE casaPort1

JBI Modules: (BPELSE) SynchronousClient (partnerlinktyperoleClient_myRole, partnerlinktyperole1_partnerRole), (BPELSE) SynchronousServer (partnerlinktyperole1_myRole)

Palette
 WSDL Bindings: file, ftp, hl7, soap, jms, mq, msmq, sap, smtp
 Service Units: Internal, External
 Endpoints: Consume, Provide

casaPort1 - Properties
 Identification
 Endpoint Name: casaPort1
 Component Name: com.sun.fileb...

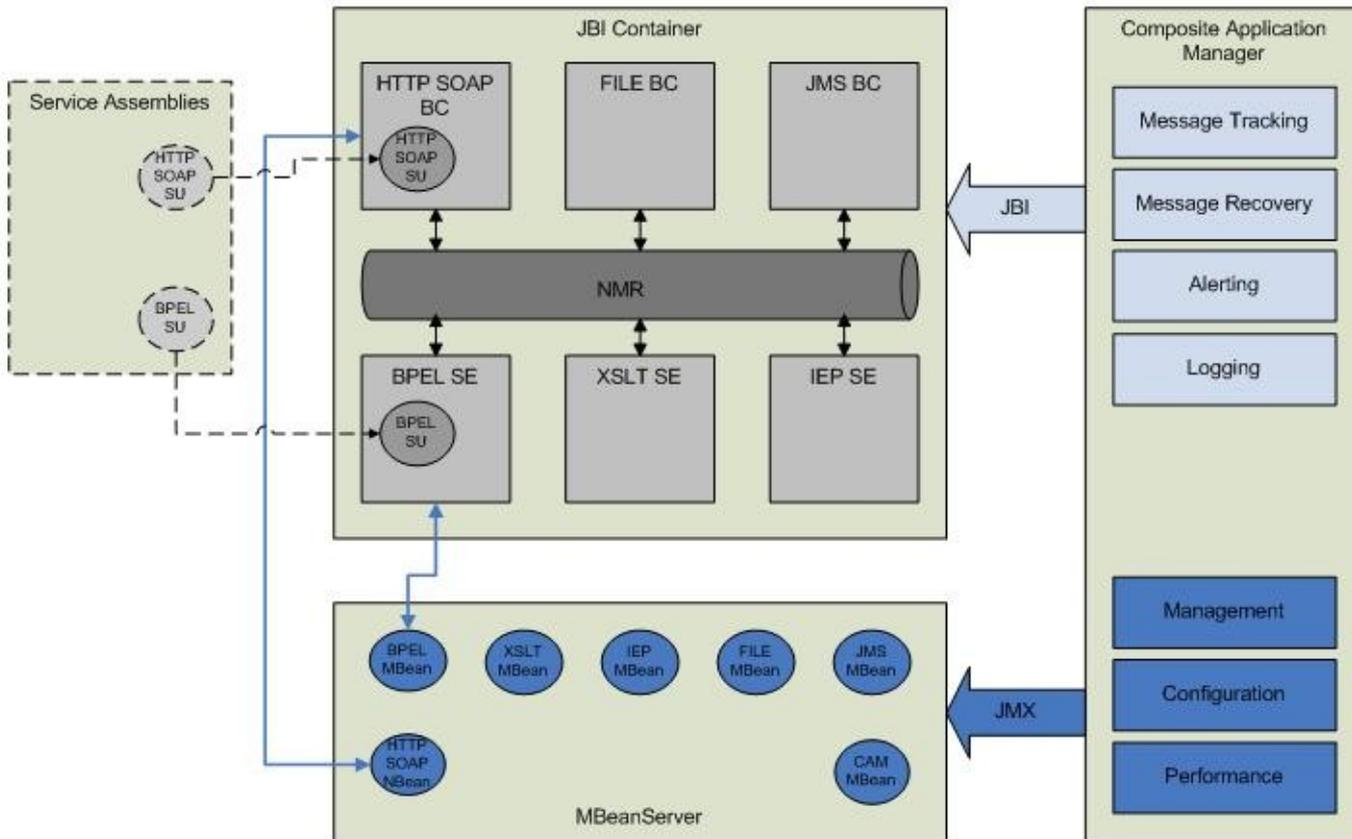
JUnit Test Results
 Output - build.xml (dist)

```

init-check:
init-taskdefs:
init:
deps-sub-project:
PackageRetrievedFiles:
Deleting directory E:\a2k7\casa2k7\test01\sample02\SynchronousClient\build\dependentProjectFiles
Created dir: E:\a2k7\casa2k7\test01\sample02\SynchronousClient\build\dependentProjectFiles
dist_se:
Building jar: E:\a2k7\casa2k7\test01\sample02\SynchronousClient\build\SEDeployment.jar
do-dist:
post-dist:
dist:
BUILD SUCCESSFUL (total time: 0 seconds)
  
```

Finished building build.xml (dist).

CompApp Manager



System Architecture Overview

- collect statistics for endpoints, SUs / SAs
- monitor and configure runtime parameters for a managed component
- control managed components (e.g. start/stop/shutdown/etc)

OpenESB: Projects

- Glassfish JBI Integration
 - > Place where JBI runtime will be implemented
 - > will become the OpenJBI project
- Open JBI Components
 - > Component Development
 - > “Independant” of OpenESB. E.g., could be used in other JBI based environment. E.g., JBossESB
- Open ESB
 - > Umbrella Project, includes runtime and components
- Open B2B
 - > B2B specific components: HIPAA, RFID, EDI, ebXML
- Netbeans Enterprise Pack

Open ESB Distribution

- Open ESB
 - > JBI Runtime
 - > Full collection of OpenESB components
 - > NetBeans based tooling (see Tooling section of presentation)
 - > Latest builds
 - > <http://open-esb.dev.java.net>
 - > http://java.sun.com/developer/technicalArticles/J2EE/sdk_overview
 - > <http://enterprise.netbeans.org/>
- Java Application Platform SDK
 - > Glassfish
 - > JBI Runtime
 - > Milestone of OpenESB components
 - > Includes other open source projects
 - > Portal, OpenSSO, etc

Open ESB and JavaCAPS

- Open ESB
 - > JBI Runtime
 - > Glassfish v2 AppServer
 - > Full collection of OpenESB components
 - > NetBeans v6 based tooling
 - > Combination of Sun and 3rd party components
 - > Constantly evolving
 - > Community Support
- JavaCAPS 5.2 ++
 - > JBI Runtime + JavaCAPS 5.1 Runtime
 - > Glassfish v2 AppServer
 - > Selection of OpenESB components
 - > NetBeans v6 based tooling (incl Enterprise Designer components)
 - > Combination of Sun and 3rd party components
 - > Sun Support

NetBeans IDE Dev 070430

File Edit View Navigate Source Refactor Build Run Profile Versioning Tools Window Help *File *Edit *View

41.2/67.1MB

Project Explorer

- RepositoryName (HEAD)
 - Project1
 - Project1
 - Map1
 - Map2
 - Queue1
 - Topic1
 - PRJ1
 - Project2
 - Map1
 - Sun SeeBeyond
 - **UnNamed**
 - eInsight

Projects | Files | Runtime

JavaApplication1

- Source Packages
 - javaapplication1
 - Main.java
- Test Packages
- Libraries
- Test Libraries

```

/*
 * Main.java
 *
 * Created on May 1, 2007, 7:45:16 PM
 *
 * To change this template, choose Tools | Template Manager
 * and open the template in the editor.
 */

package javaapplication1;

/**
 *
 * @author mjenkins
 */
public class Main {

    /** Creates a new instance of Main */
    public Main() {
    }

    /**
     * @param args the command line arguments
     */

```

More Info

- JBI
 - > <http://www.jcp.org/en/jsr/detail?id=208>
 - > <http://java.sun.com/integration/>
- OpenESB Project
 - > <http://open-esb.dev.java.net/>
 - > <https://open-jbi-components.dev.java.net/>
 - > <http://www.glassfishwiki.org/jbiwiki/Wiki.jsp?page=Jbicomps>
- JavaEE SDK
 - > <http://java.sun.com/javaee/downloads/index.jsp>
- Examples and Demos:
 - > <http://enterprise.netbeans.org/>



OpenESB

- **Sang Shin**
- Java Technology Evangelist
- Sun Microsystems, Inc.

