



J-Fall

11 oktober 2007 Spant! - Bussum



Real World JBI

Jos Dirksen

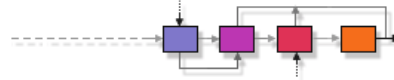


Agenda

- Introduction
- **What is JBI?**
- JBI in practice
- JBI together with BPEL and SCA
- Future of JBI: JBI 2.0
- Q&A



JSR-208



Community Development of
Java Technology Specifications

- Open Standard to define a pluggable architecture
 - Final draft approved June 2005
 - EG members:
 - Apache, Borland, Cap Gemini, Deutsch Post, Fujitsu, Intalio, IONA, IOPSYS Software, JBoss, Nokia, Novell, Oak Grove Systems, Oracle, Research in Motion (RIM), SAP, SeeBeyond, Sonic, Sun Microsystems, Sybase, TIBCO, Vignette, WebMethods

“JBI, also known as JSR 208, may be a Java-based specification, but it will have a profound impact on Web services and software markets in general.” - Gartner



What does JBI do?

- A container to plug in engines and bindings
 - Engine to run BPEL processes...
 - Binding to provide JMS connectivity...
- Extensible administration
- Loose coupling: WSDL driven
- Complex specification: JSR-208

“There is no simple answer for Enterprise Integration”
- Hohpe, Woolf

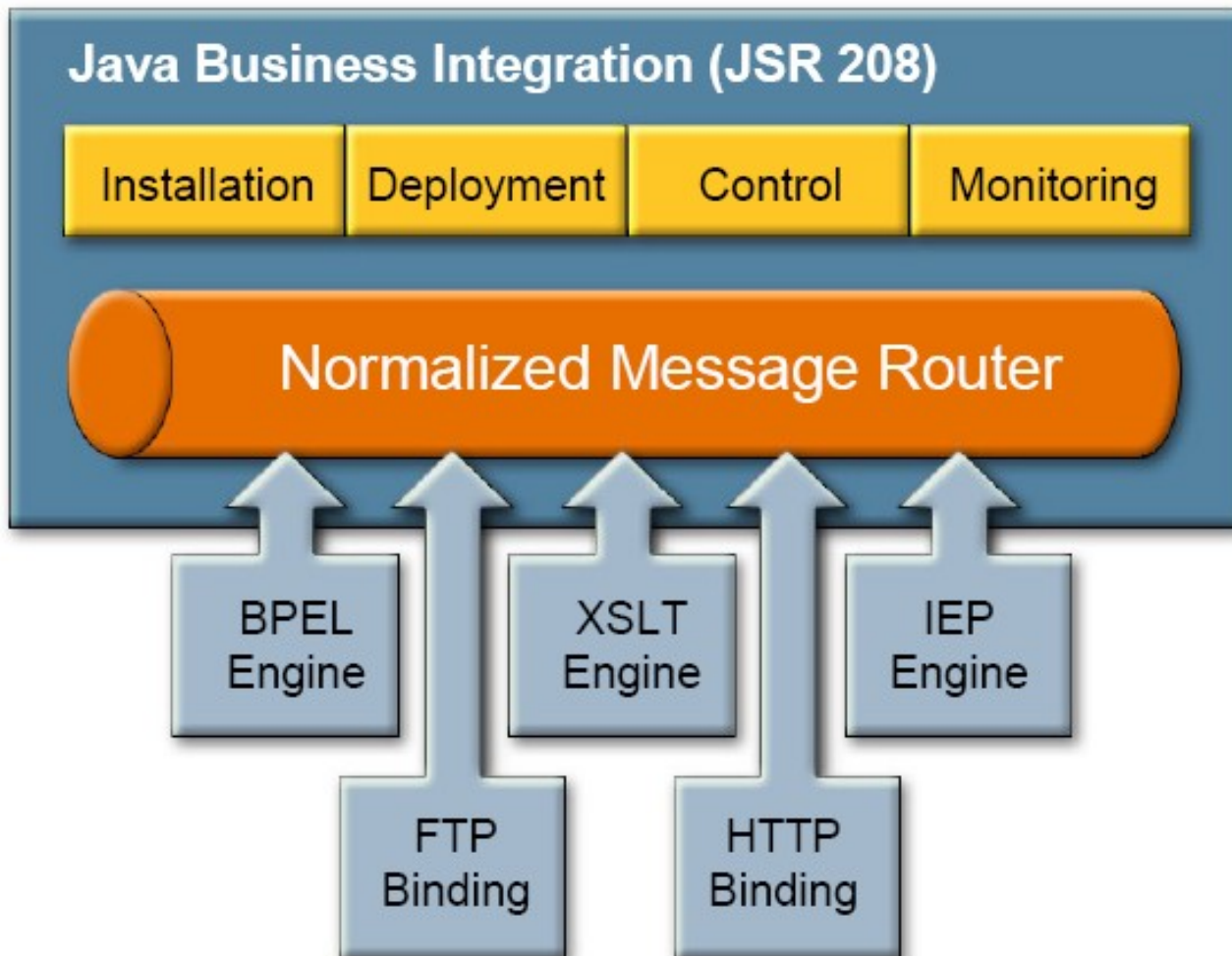


Agenda

- Introduction
- **What is JBI?**
- JBI in practice
- JBI together with BPEL and SCA
- Future of JBI: JBI 2.0
- Q&A



High level architecture 1



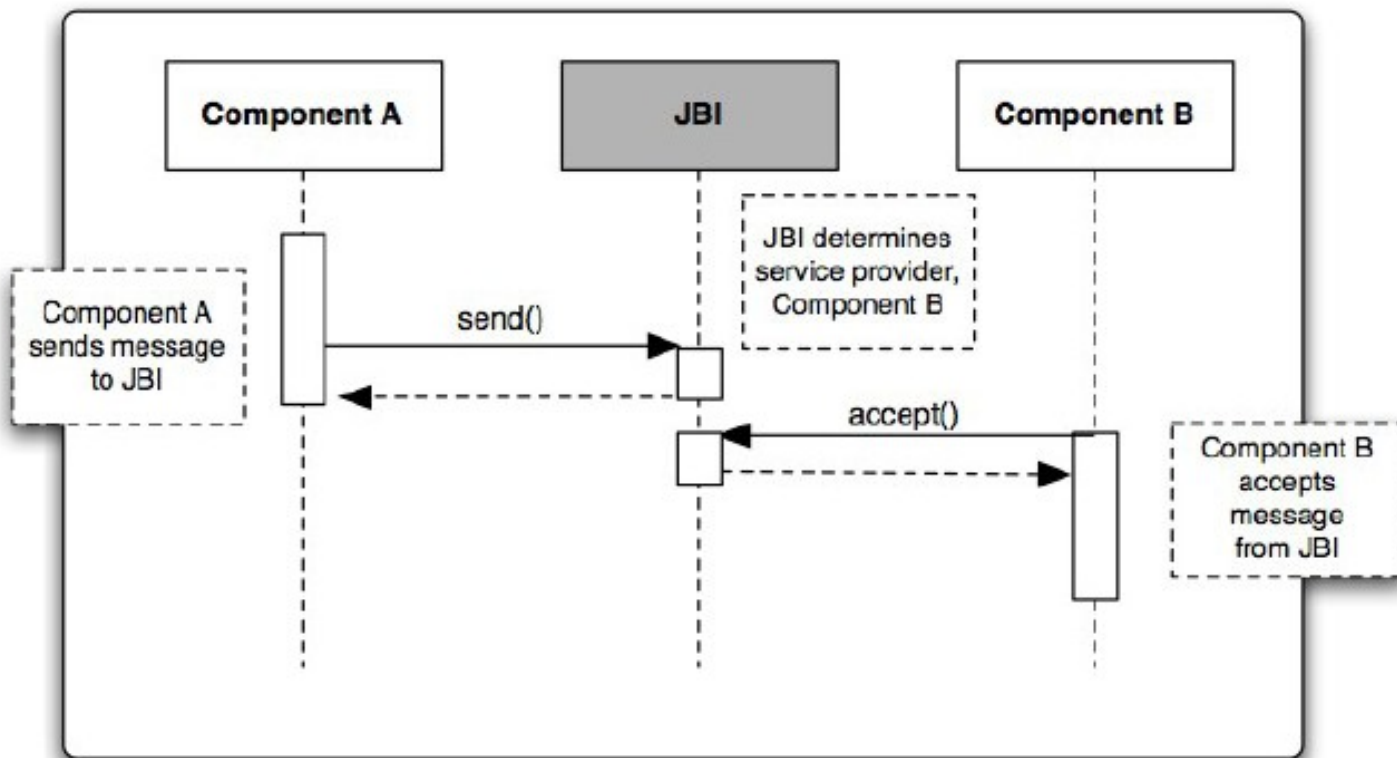
.nl. jug JBI Components

- Service Engines:
 - Business Processes (orchestration BPEL)
 - Transformation (XSLT, EDI)
 - Business Logic (EJBs)
- Binding Components:
 - SOAP, AS2, ebXML MSH, REST, JMS
 - Proxy to remote service providers
 - BC-proxied services
 - Access for remote service consumers
 - SE-provided services



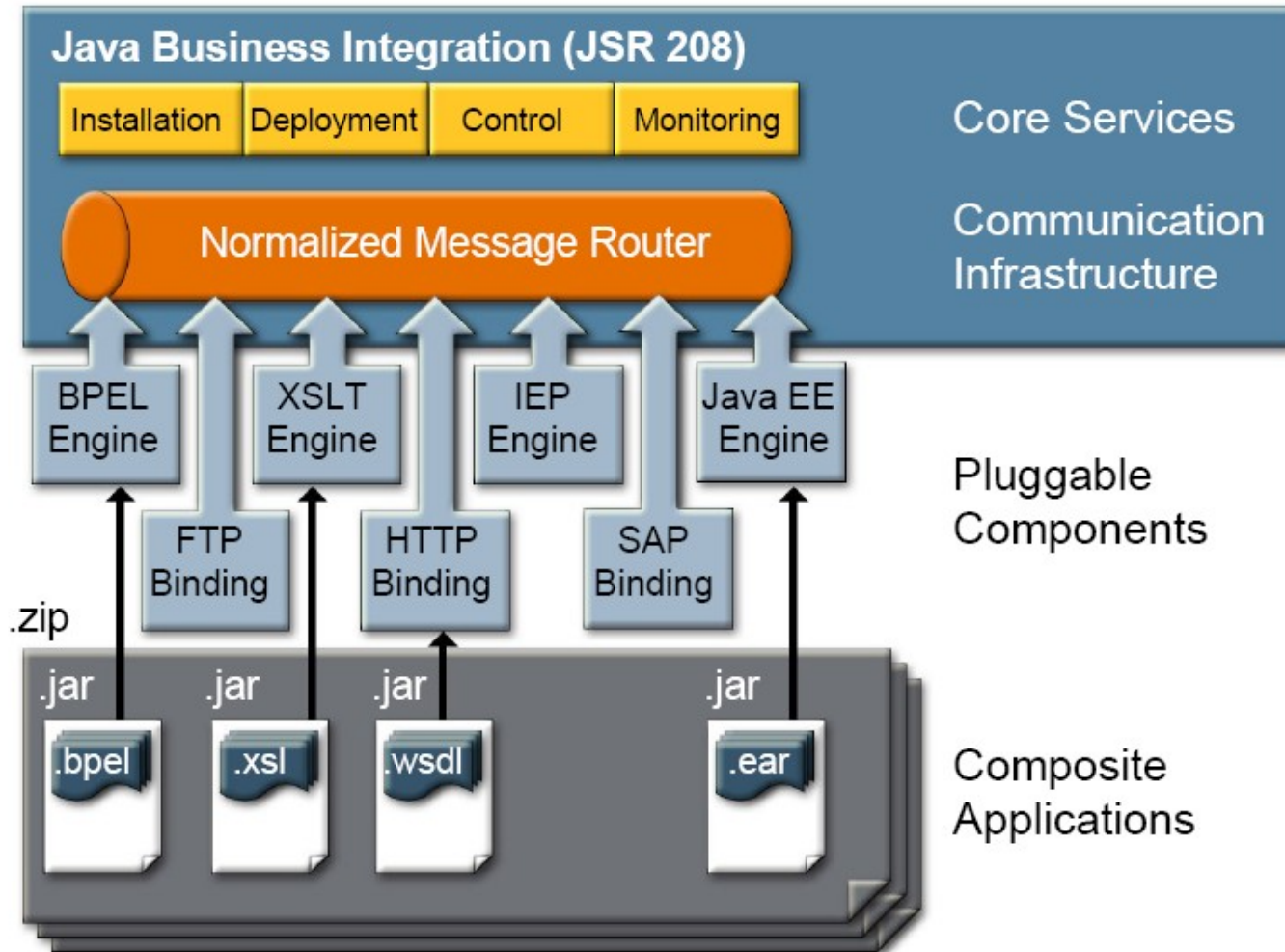
Normalized Message Router

- Sits between the components
- Routes the messages





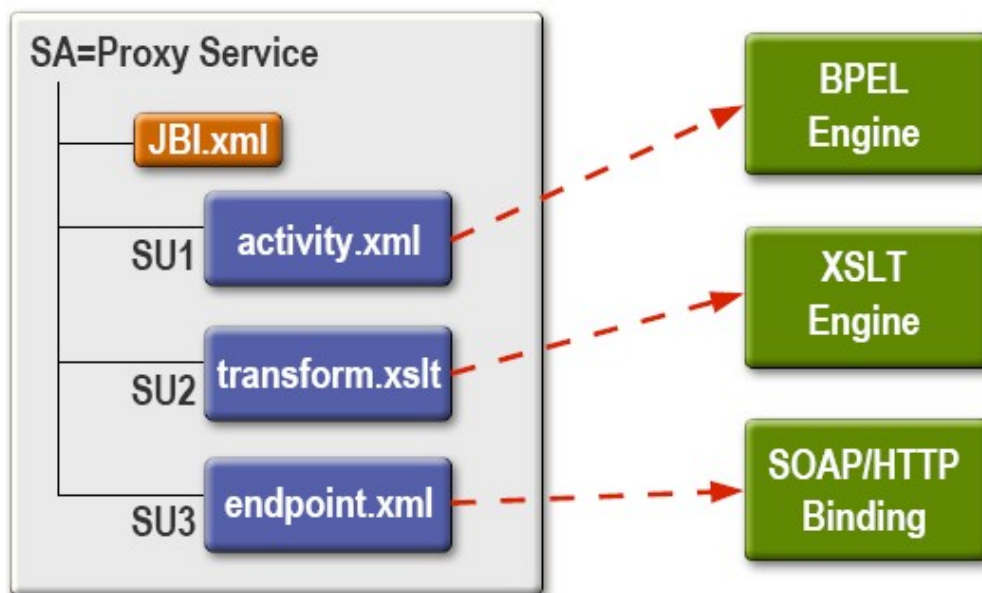
High Level architecture 2





Artifacts deployment

- Service Unit
 - Define consumers and providers
 - xsrts, EARs, bpel
- Service Assembly
 - Package artifacts multiple engines
- Components





JBI.xml: Service Unit

```
<jbi version="1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://java.sun.com/xml/ns/jbi"
  xmlns:payroll="http://www.payroll.org/payroll.wsd1">
  <services>
    <consumes service-name="payroll:PayrollService"
      interface-name="payroll:PayrollIF"
      endpoint-name="ftpEndpoint"/>
    <provides service-name="payroll:CustomerService"
      interface-name="payroll:CustomerIF"
      endpoint-name="customerServiceEndpoint"/>
  </services>
</jbi>
```

.nl. jug JBI Summary

- Container of containers
- SE: business logic, BC: connectivity
- NMR: Loosely coupled / MEP
- Deployables:
 - Components (SE, BC)
 - Service Units: deployed to a component
 - Service Assembly: collection of SU
- Management: JMX / ant
- Everything is hot deployable



Agenda

- Introduction
- What is JBI?
- **JBI in practice**
- JBI together with BPEL and SCA
- Future of JBI: JBI 2.0
- Q&A

.nl. jug JBI implementations

- Open Source
 - Apache ServiceMix
 - PEtALS, ObjectWeb
 - Open ESB, Sun
 - FUSE, IONA, based on ServiceMix
- Commercial:
 - Oracle Fusion
 - Tibco ActiveMatrix
 - and more..



Open ESB
The *Open* Enterprise Service Bus

ORACLE

.nl. jug JBI Demo

- ServiceMix, the XML approach:
 - Webservice -> XSLT -> javabean -> File
 - EIP for routing
- Open ESB, the GUI approach:
 - Everything through WSDLs
 - Routing done through BPEL
- Management
 - JMX
 - Ant

.nl. jug JBI in practice

- JBI is very simple to use
 - When you know the basics
- Implementations hide JBI complexity
 - Almost no editing of descriptors needed
 - No need to know the inner details of JBI
 - Focus on using the BCs and the SEs
- Facilitate component development
 - PEtALS provide “PEtALS CDK”



Agenda

- Introduction
- What is JBI?
- JBI in practice
- **JBI together with BPEL and SCA**
- Future of JBI: JBI 2.0
- Q&A

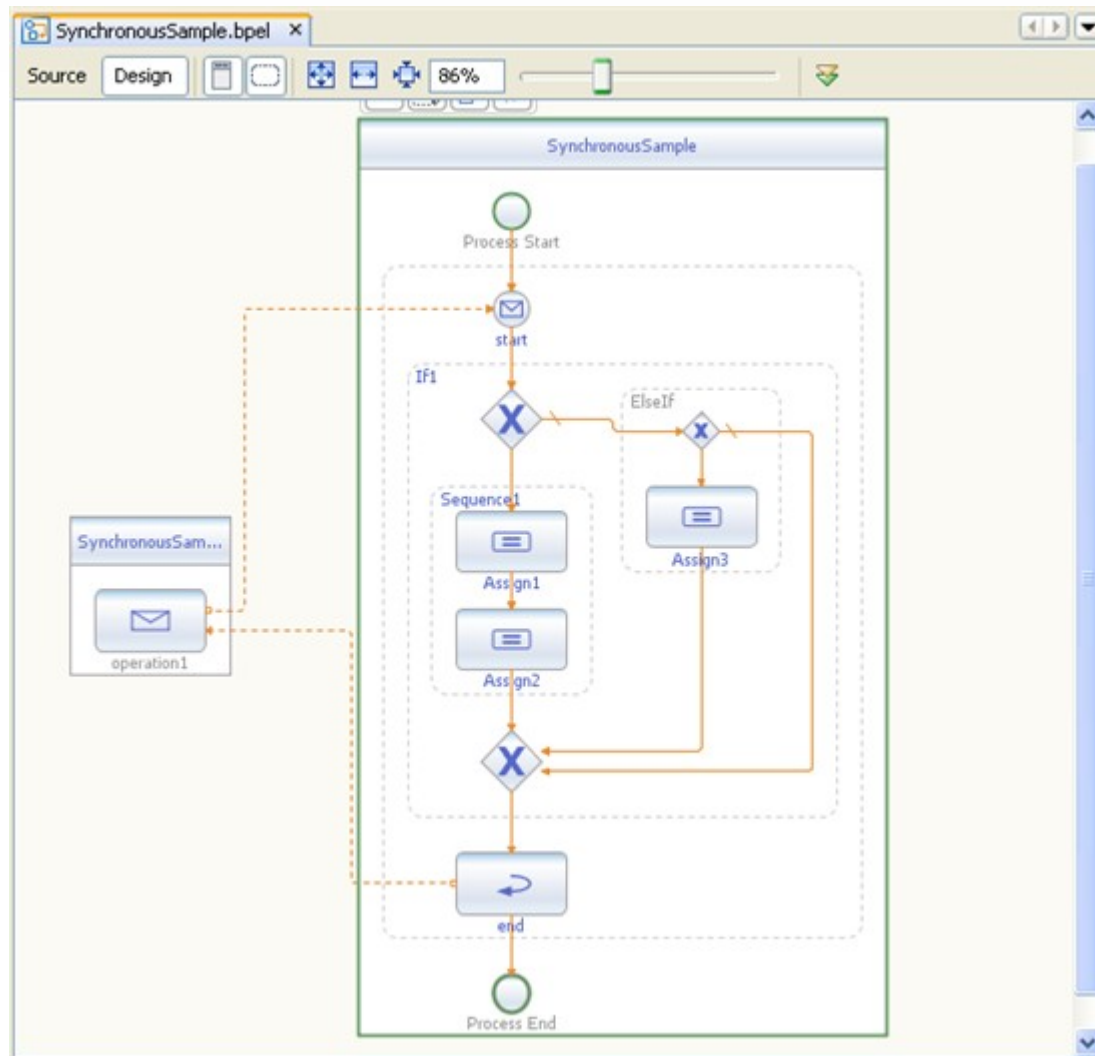


JBI and BPEL: What is BPEL?

- BPEL:
 - **B**usiness **P**rocess **E**xecution **L**anguage
 - XML based language
 - Not meant for human consumption
 - Focuses on tooling
- Orchestration of web services
- Stateful long-running transactions
- Uses only 'abstract' part of the WSDL
- Runs inside a BPEL engine
 - PXE, activeBPEL, ODE etc.



JBI and BPEL: What is BPEL?





JBI and BPEL together

- JBI provides:
 - Messaging infrastructure
 - Services are self describing (WSDL)
 - BC for external connectivity
- BPEL provides:
 - A way to orchestrate services
 - means to run long running transactions
 - knows how to invoke and receive based on the abstract WSDL
- BPEL Partner Links mapped JBI endpoints



JBI and BPEL example

```
<dd:deployment-descriptor
```

```
  <dd:endpoint-refs>
```

```
    <dd:endpoint-ref
```

```
      partnerLinkName="SomePartnerLink"
```

```
      partnerLinkRole="myRole">
```

```
        <jbi:end-point-reference
```

```
          service-name="{http://bla}service"
```

```
          end-point-name="myEndpoint" />
```

```
        </dd:endpoint-ref>
```

```
      </dd:endpoint-refs>
```

```
    </dd:deployment-descriptor>
```

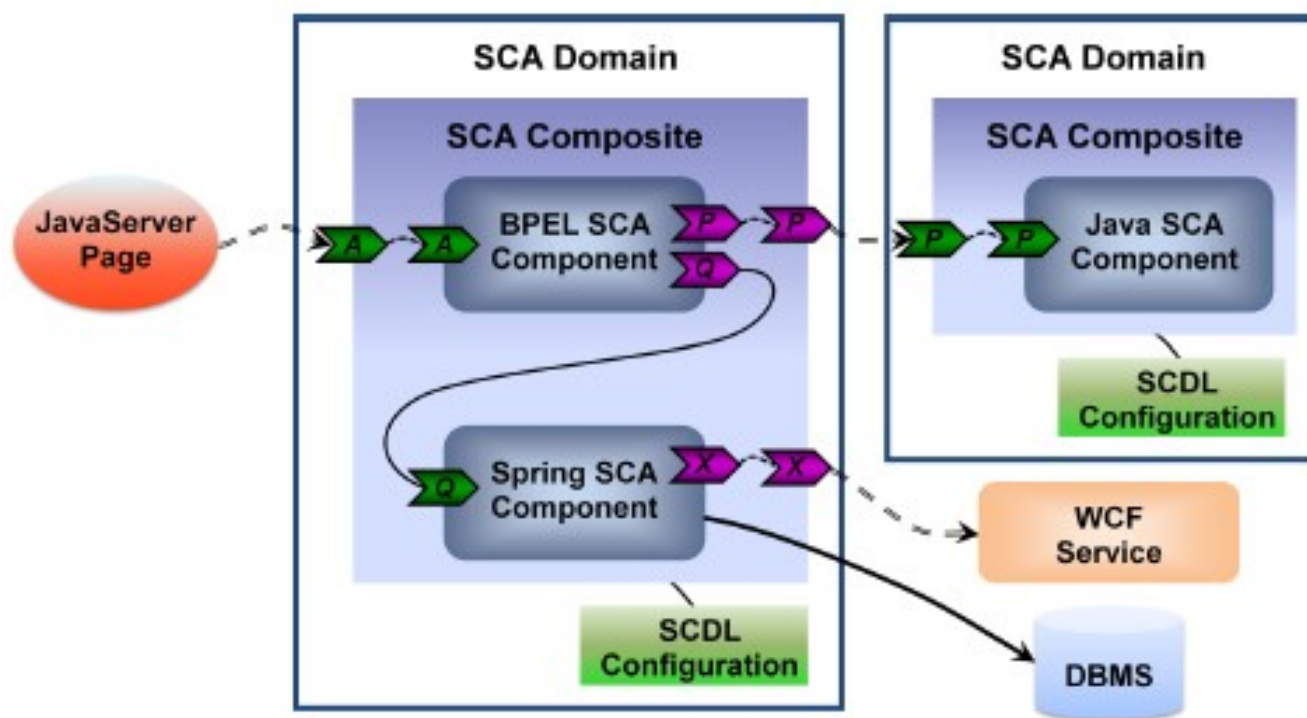


JBI and SCA: what is SCA

- **S**ervice **C**omponent **A**rchitecture
- Specifies how to:
 - build service components
 - combine them into a composite application
 - deploy to a SCA runtime
- Service can be build using:
 - Spring, Java, EJB, BPEL, C++, PHP etc.
- Vendor, language and technology neutral
- Doesn't specify the container architecture



JBI and SCA: what is SCA



- Service
- Reference
- Promotion
- Wire (Domain-defined communication)
- Web services or other interoperable communication
- Database access

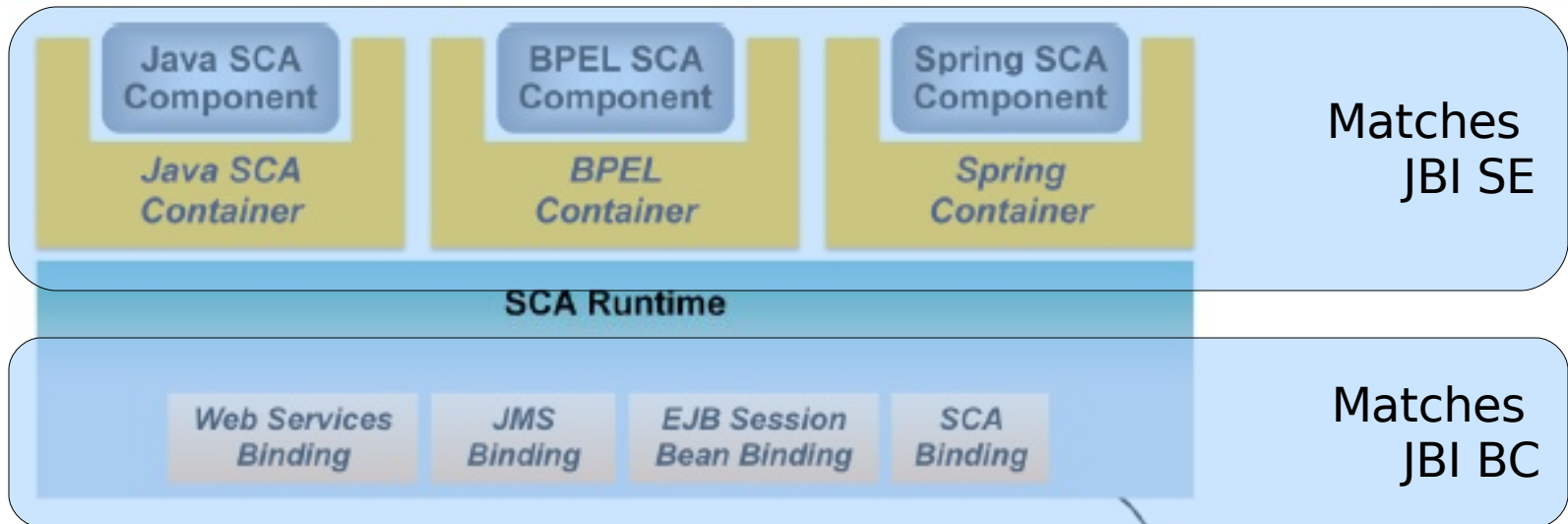


SCA and JBI together

- They aren't competitors
- SCA and JBI focus on different roles
 - JBI:
 - Middleware developers
 - integration components
 - enterprise integration
 - SCA:
 - Application developers / application assemblers
 - composite applications
- SCA could use JBI as it's backend



SCA and JBI together



```
<composite ...>
  <component name="Component1">
    <implementation.java ... />
  </component>
  <component name="Component2">
    <implementation.bpel ... />
  </component>
  <component name="Component3">
    <implementation.spring ... />
  </component>
</composite>
```

SCDL Configuration



Agenda

- Introduction
- What is JBI?
- JBI in practice
- JBI together with BPEL and SCA
- **Future of JBI: JBI 2.0**
- Q&A



Future of JBI: JBI 2.0

- JSR 312: Java Business Integration 2.0
 - Very early (started April 2007)
 - Final Specification: late Q2/2008
- Some of the features:
 - Make JBI 2.0 the default runtime for SCA
 - Performance optimization
 - Message normalization
 - Full support for OSGi
 - Clustered / distributed environments

"It's like EJB3 for ESBs",

Little, Spec Lead JBoss ESB

.nl. jug Q&A



- Any questions?
- More info:
 - jos.dirksen@atosorigin.com
 - Open Source ESBs in action
 - First chapters available in MEAP
 - <http://www.manning.com/rademakers>
 - JBI 1.0
 - <http://jcp.org/en/jsr/detail?id=208>
 - JBI 2.0
 - <http://jcp.org/en/jsr/detail?id=312>

