JBI based ESB as backbone for SOI applications

Michael Wisler
Zühlke Engineering AG
Submission ID: 687

Goal of this talk

This session brings the JBI (Java Business Integration) standard in context to SOI (Service- Oriented Integration) and ESB (Enterprise Service Bus) and will give an overview of already available Open Source solutions based on this technology.

AGENDA

Part I

> Deciphering the buzzwords:
  - What is SOI?
  - What is an ESB?
  - What is JBI?

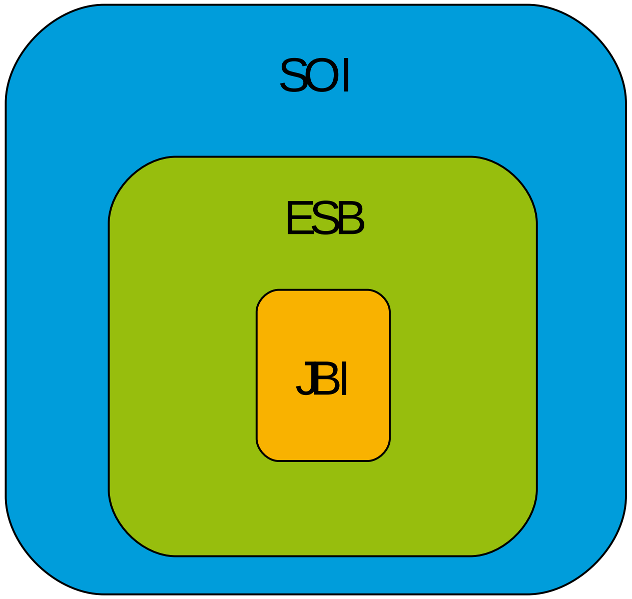
Part II

> JBI Architecture

Part III

> OpenSource ESBs

Conclusions and Outlook



AGENDA

Part I

> **Deciphering the buzzwords:**
- **What is SOI?**
- **What is an ESB?**
- **What is JBI?**

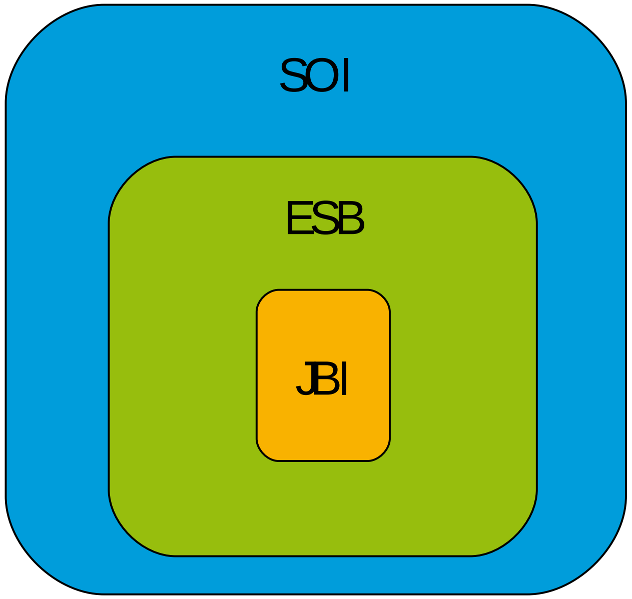
Part II

> JBI Architecture

Part III

> OpenSource ESBs

Conclusions and Outlook



What is SOI (Service- Oriented Integration) ?

> Adapting applications and protocols to a SOA is called service- oriented integration (SOI)
> Existing IT business functions or resources serve as basis for services
> Construction of different service- based components to business need
> Loosely coupled components to be flexible and responsivness to business changes
> Adapting incompatible protocols and message formats (common problem in enterprise integration)

What is an ESB (Enterprise Service Bus)?

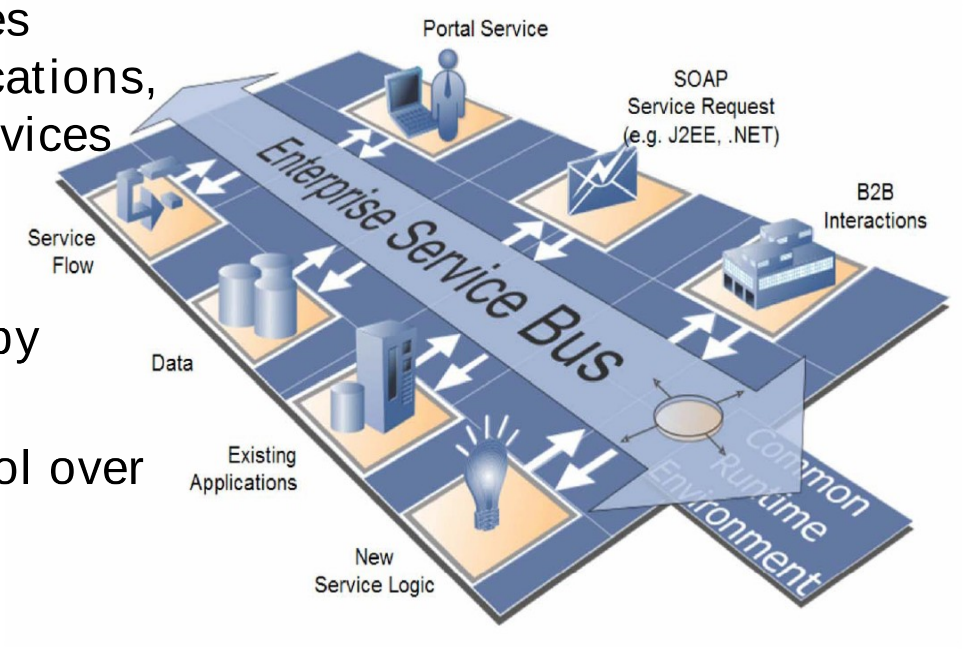
> Infrastructure software that makes services widely available to applications, business processes and other services

> An ESB is mediating between services

> An ESB obtains the value of SOA by increasing connectivity

> An ESB is providing greater control over use of the resources it binds

> Not “Hub-and-Spoke”



Definition of an ESB?

The term ‘ESB’ is not standardized but just one definition:

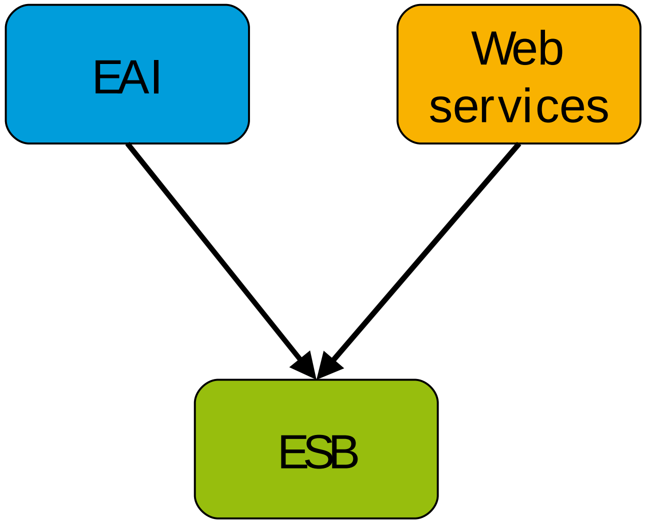
“An ESB is an open standards, message-based, distributed integration solution that provides routing, invocation, and mediation services to facilitate the interactions of disparate distributed IT resources (applications, services, information, platforms) in a reliable manner.”

Where is the ESB technology coming from?

> The ESB technology is grown out of the enterprise application integration (EAI) and the Web services infrastructure.

- EAI technology :
  Integration based on message- oriented middleware (MOM) was adding WS support in response to SOA opportunities

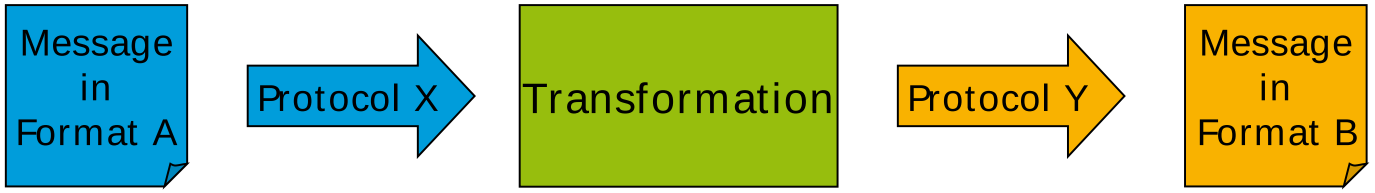
- Web services technology :
  Segmented WS infrastructure providing security, management, registries and more



What does an ESB really do?

> The core function of an ESB is transformation and mapping.

> Data arriving in one format needs to be transformed before being sent to other systems, necessitating the mapping of fields within one document to another



> Let’s see what the JBI standard is covering from all those ESB requirements

What is JBI (Java™ Business Integration)?

> Defines a standard for building system integration applications using WSDL and XML-based messaging
>
> Helps to create a standardized integration platform
>
> Defines a standard meta-container for integrated services
>
> XML service bus works well with WS and WSDL, or XML over HTTP, or XML over JMS, etc.
>
> JBI components are deployable archives (.jar files) containing code and descriptors similar to WAR or EAR packages in Java EE.

> JBI 1.0 (JSR 208, Final release August, 2005)
>
> JBI 2.0 (JSR 312, Q2 2008)

AGENDA

Part I

> Deciphering the buzzwords:
  – What is SOI?
  – What is an ESB?
  – What is JBI?

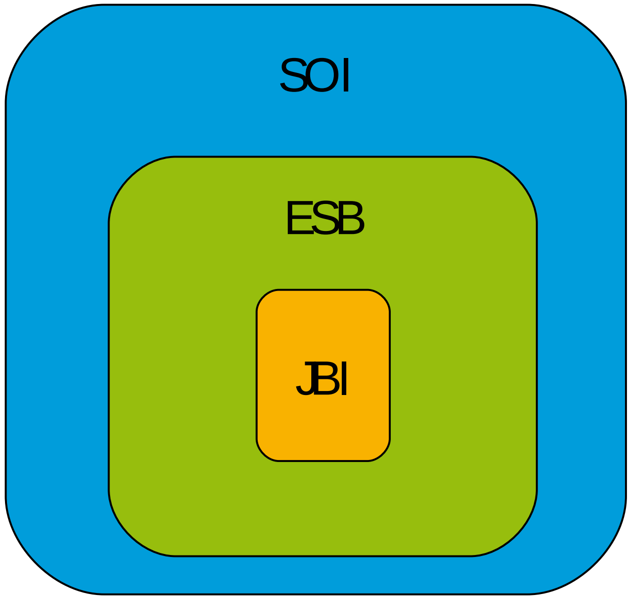
Part II

> **JBI Architecture**

Part III

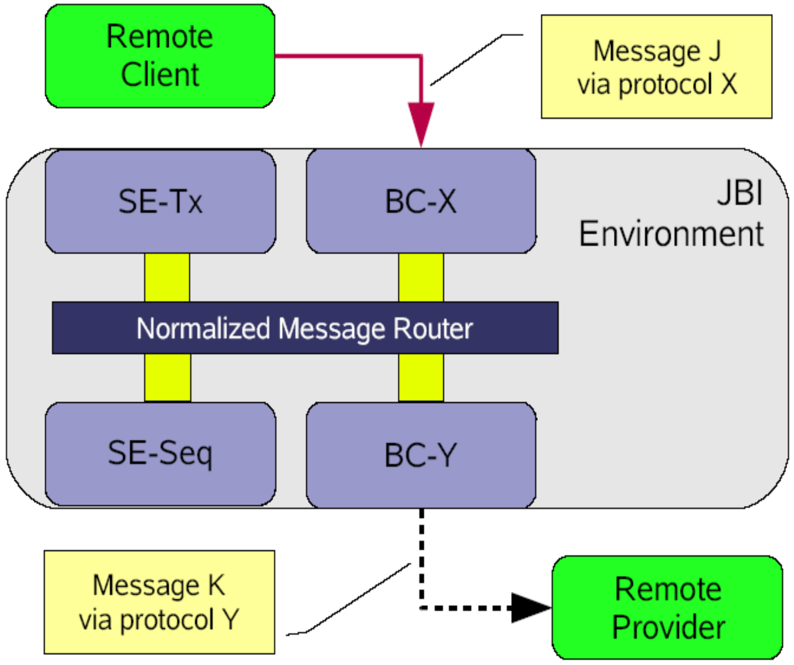
> OpenSource ESBs

Conclusions and Outlook



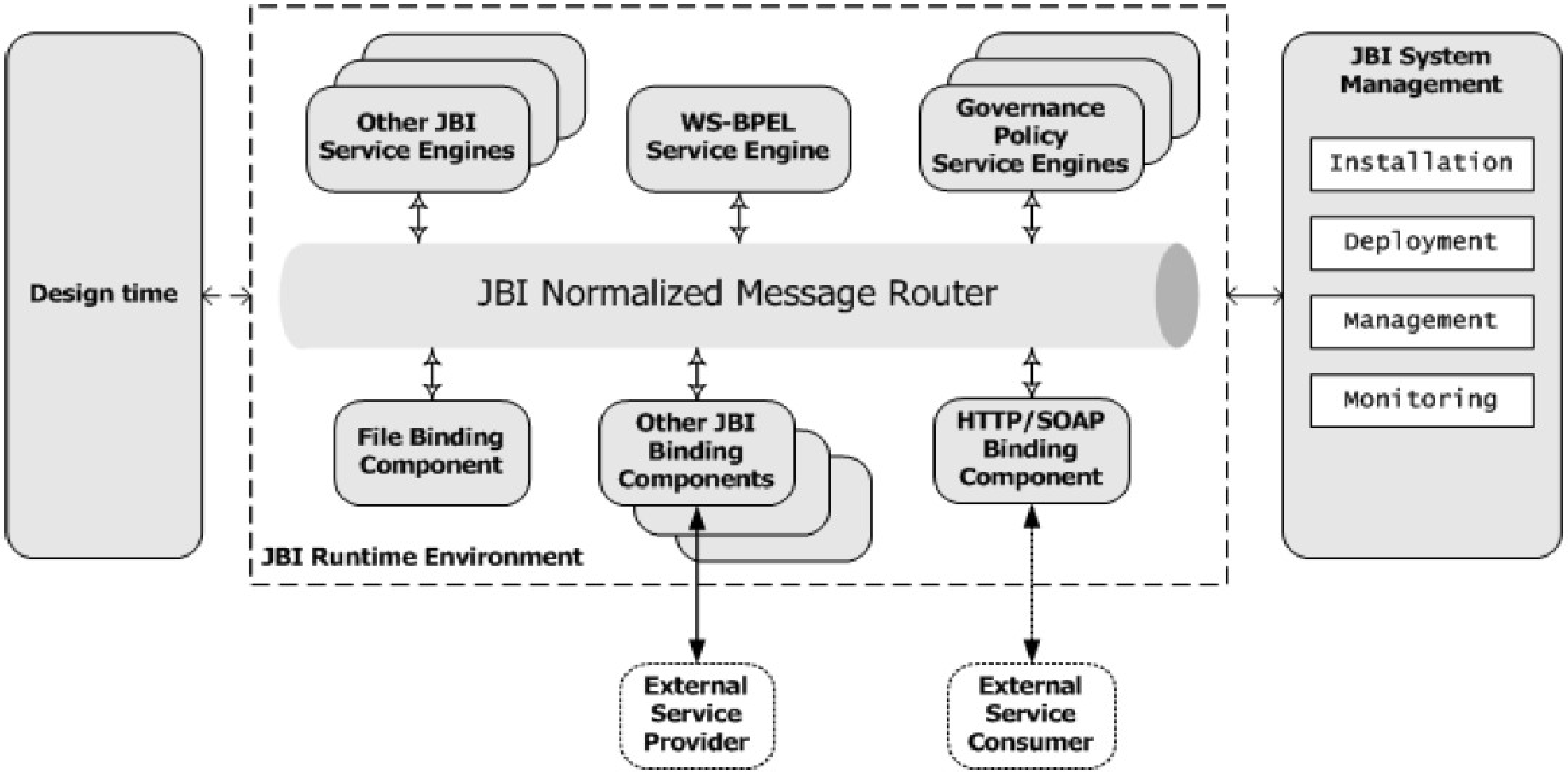
JBI Architecture

- The key pieces of the JBI environment:
  - Service Engines (SE): Enabling pluggable business or transformation logic
  - Binding Components (BC): Enabling pluggable external connectivity
  - Normalized Message Router (NMR): Directing normalized messages from source to destination components
  - JBI (meta) container: The JBI Runtime Environment controls the JBI components (SEs and BCs) and the NMR



Composite Adapter Pattern

JBI Architecture (cont.)


Design time
Other JBI Service Engines
WS-BPEL Service Engine
Governance Policy Service Engines
JBI Normalized Message Router
File Binding Component
Other JBI Binding Components
HTTP/SOAP Binding Component
JBI Runtime Environment
JBI System Management
Installation
Deployment
Management
Monitoring
External Service Provider
External Service Consumer

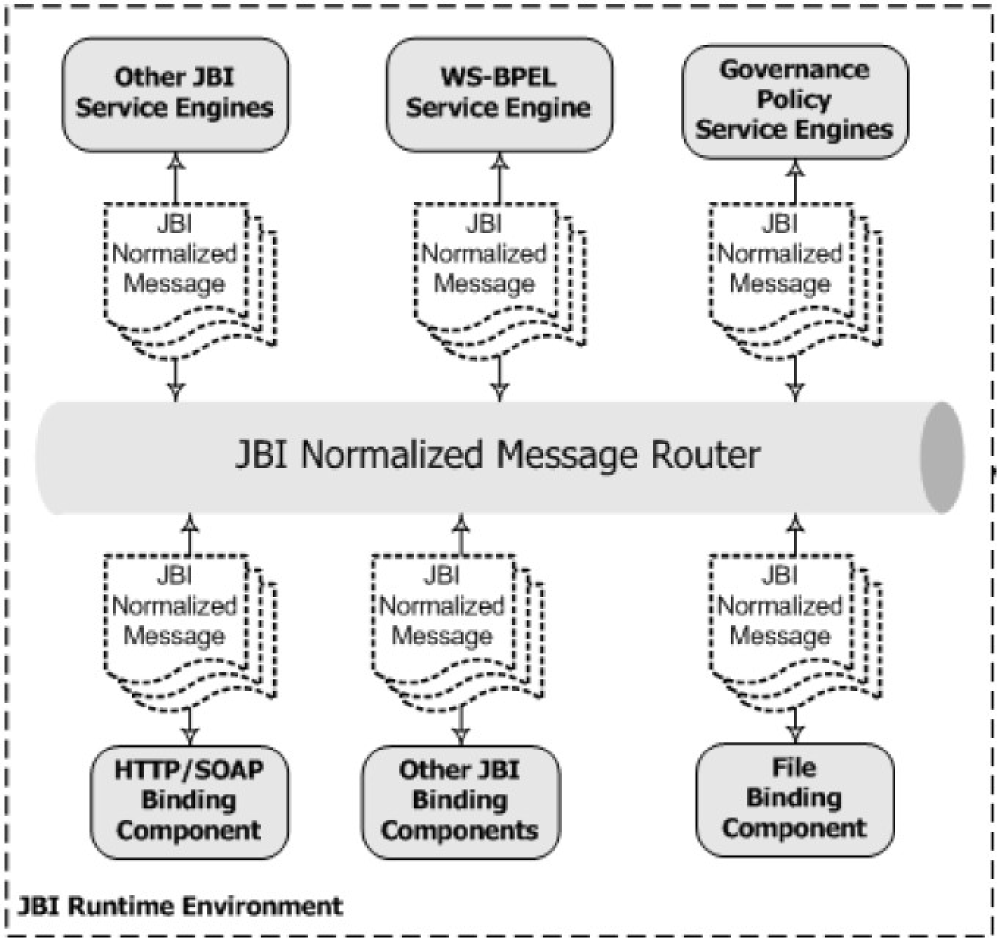
JBI - Service Engine

> The business logic drivers of the JBI system
> Orchestration for long-lived business processes with WS-BPEL
> Processing data/message transformation
> Routing of messages (e.g. message-based routing)
> SEs can serve as service providers, service consumers, or both

JBI - Binding Component

> Used to send and receive messages via particular protocols and transports (HTTP, SOAP, File, …)
> Provide access to local or remote services from within the JBI environment
> Isolate JBI environment from particular protocol allowing the NMR to deal only with normalized messages
> Enable loose coupling by decoupling the service implementation (of SEs) from the access mechanism
> BCs can serve as service providers, service consumers, or both

JBI - Normalized Message Router



JBI - Normalized Message

- XML document used for the JBI message exchange.
- Typically it consists of two parts:
  - Message context (metadata) such as protocol-supplied context information, security tokens, transaction context information, or data specific to other components
  - Message payload. A generic source abstraction that contains all the message data. The payload conforms to an abstract WSDL message type, with no protocol encoding or formatting.
- Service providers may use WSDL 2.0 or WSDL 1.1

JBI Example – Pojo Service Engine

```
import org.apache.servicemix.MessageExchangeListener;
import javax.annotation.Resource;
import javax.jbi.messaging.DeliveryChannel;
import javax.jbi.messaging.ExchangeStatus;
import javax.jbi.messaging.MessageExchange;
import javax.jbi.messaging.MessagingException;

public class ListenerBean implements MessageExchangeListener {

    @Resource
    private DeliveryChannel channel;

    public void onMessageExchange(MessageExchange exchange) throws MessagingException {
        System.out.println("Received exchange: " + exchange);
        exchange.setStatus(ExchangeStatus.DONE);
        channel.send(exchange);
    }
}
```

JBI Example – Pojo Service Engine

Deployment Configuration:

```
<beans xmlns:bean="http://servicemix.apache.org/bean/1.0">
 <bean:endpoint service="test:service" endpoint="endpoint" bean="#listenerBean"/>
 <bean id="listenerBean" class="org.apache.servicemix.bean.beans.ListenerBean"/>
</beans>
```

JBI Example – Pojo Service Engine

Client Access (normalized):

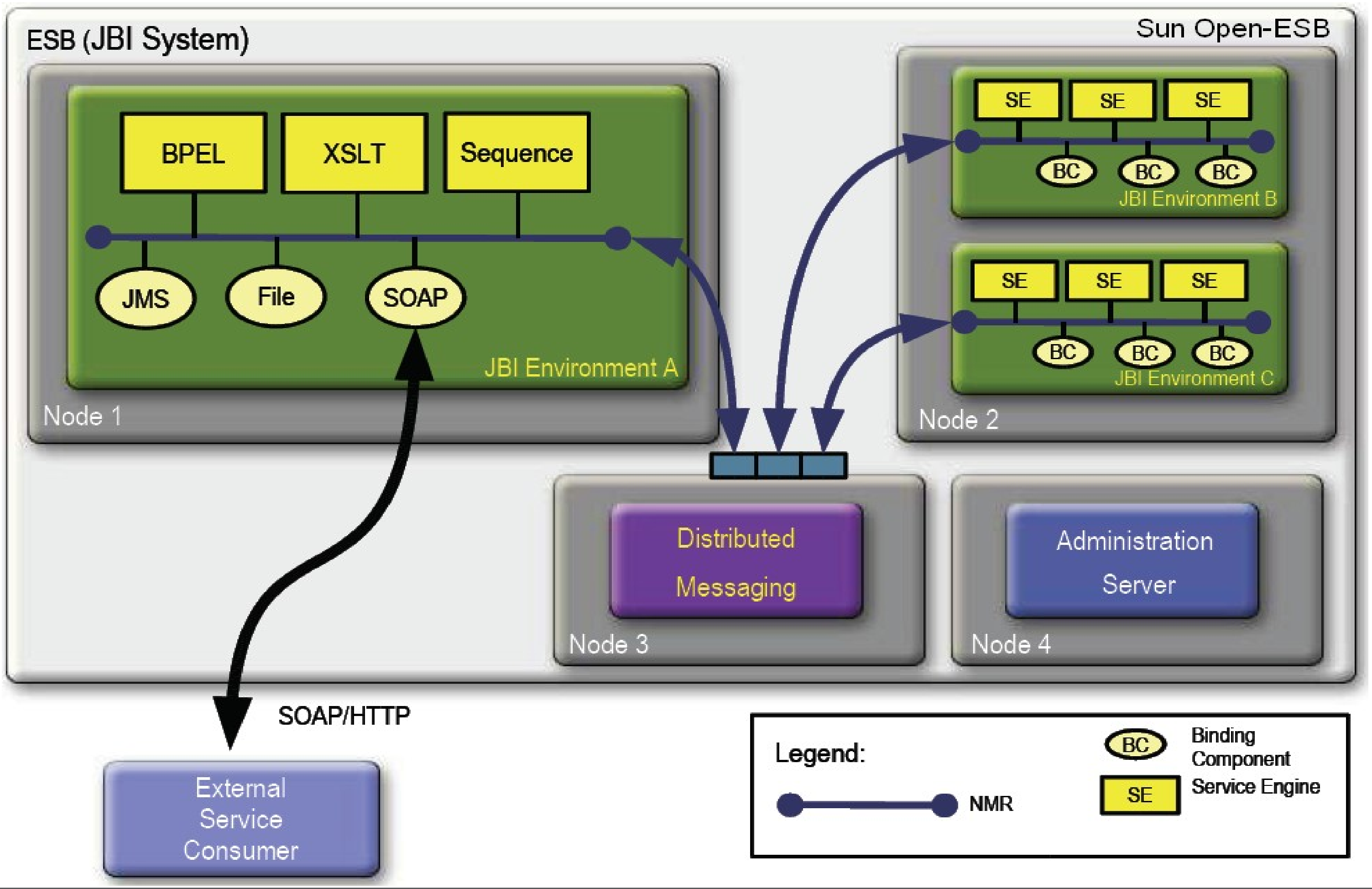
```
import javax.jbi.messaging.InOnly;
import javax.jbi.servicedesc.ServiceEndpoint;
import org.w3c.dom.DocumentFragment;
import org.apache.servicemix.client.DefaultServiceMixClient;

    pojoTest()
    {
                                JBIContainer jbi = (JBIContainer) context.getBean("jbi");
        DefaultServiceMixClient client = new DefaultServiceMixClient(jbi);

        DocumentFragment epr = URIResolver.createWSAEPR("bean:listenerBean");
        ServiceEndpoint se = client.getContext().resolveEndpointReference(epr);

        InOnly exchange = client.createInOnlyExchange();
        exchange.setEndpoint(se);
        exchange.getInMessage().setContent(new StringSource("<hello>world</hello>"));
        client.sendSync(exchange);
    }
```

JBI based ESB as backbone



What’s coming with JBI 2.0

- JBI in clustered or distributed environments
- SOA based approach to creation, deployment and runtime support for Compostie Applications
- Support for WS-Policy
- Support for Web 2.0 technologies
- Alignment with Java EE and transactions
- Alignment with SCA (Service Component Architecture) to make JBI 2.0 a standard Java runtime for SCA
- Support compatibility with OSGi - Open Services Gateway initiative

- Final release planned Q2/2008

→TS1841 (Wed, 11h20): “What’s coming with JBI 2.0”, Peter Walker (JBI co-spec lead)

AGENDA

Part I

> Deciphering the buzzwords:
  - What is SOI?
  - What is an ESB?
  - What is JBI?

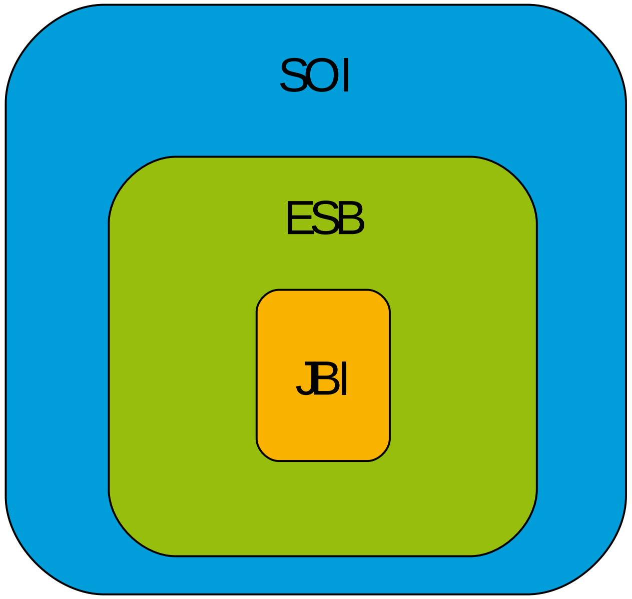
Part II

> JBI Architecture

Part III

> **OpenSource ESBs**

Conclusions and Outlook



Open Source JBI container

> Focusing on Open Source ESBs supporting JBI
> JBI based containers vs. JBI supporting containers

ServiceMix

Open ESB
The Open Enterprise Service Bus

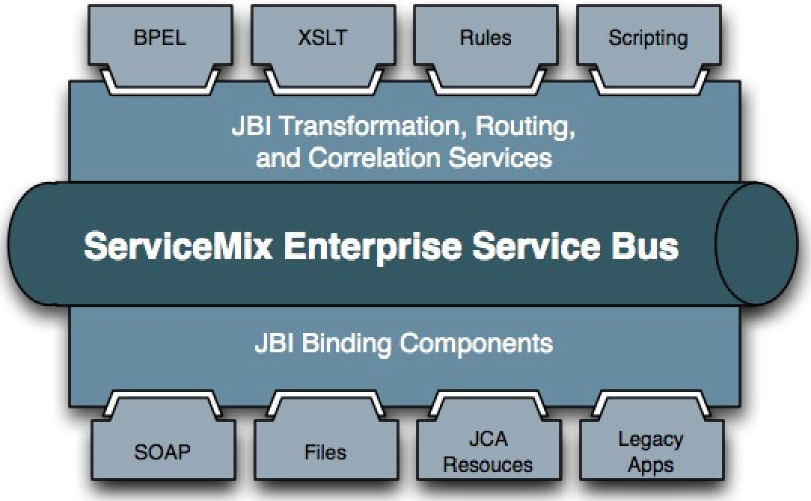
mule

PEtALS

Celtix™
Open Source Java ESB

Apache ServiceMix

> Built ground-up on JBI spec
> Spring support and can be used in Java SE or an Java EE app server
> Integrated into Apache Geronimo
> Uses ActiveMQ MOM to provide remoting, clustering, reliability and distributed failover
> Sponsored by LogicBlaze, which is now bought by Iona (April, 2007)
> http://incubator.apache.org/servicemix


BPEL
XSLT
Rules
Scripting
JBI Transformation, Routing, and Correlation Services
ServiceMix Enterprise Service Bus
JBI Binding Components
SOAP
Files
JCA Resouces
Legacy Apps

Sun’s Open-ESB

> Built ground-up on JBI spec
> Integrated into GlassFish (no stand-alone runtime yet)
> Can be used for conformance validations of JBI components
> Composite Application Editor to ‘wire-together’ services
> Tool support for management and monitoring of services
> http://www.open-esb.org

Mule

- Long-established open-source lightweight messaging framework and distributable object broker
- Not based on JBI but has JBI integration/binding
- Spring support
- Wiring through POJO services (no XML-based NMR) using Universal Message Objects
- Many transport and transformation components already available
- Sponsored by MuleSource Inc.
- http://mule.codehaus.org

From Mule's Webpage: „Mule's ultimate goal is to be the *Swiss-army knife* of integration“ ☺

PEtALS

> Full JBI support
> Uses JORAM (JMS implementation) for Messaging
> Integrated in JOnAS application server
> Basic tooling for monitoring network
> Sponsored EBM WebSourcing

> http://petals.objectweb.org

OpenSource JBI- ESB products

> Celtix Enterprice – Iona:
  – Apache CXF (service framework, ‚SOAP stack‘), ServiceMix
  – http://www.ionaceltix.com, http://celtix.objectweb.org

> Fuse – LogicBlaze:
  – ServiceMix, ActiveMQ
  – http://www.logicblaze.com

> ChainBuilderESB – Bostech Corporation:
  – ServiceMix
  – http:// www.chainforge.net

AGENDA

Part I

> Deciphering the buzzwords:
  – What is SOI?
  – What is an ESB?
  – What is JBI?

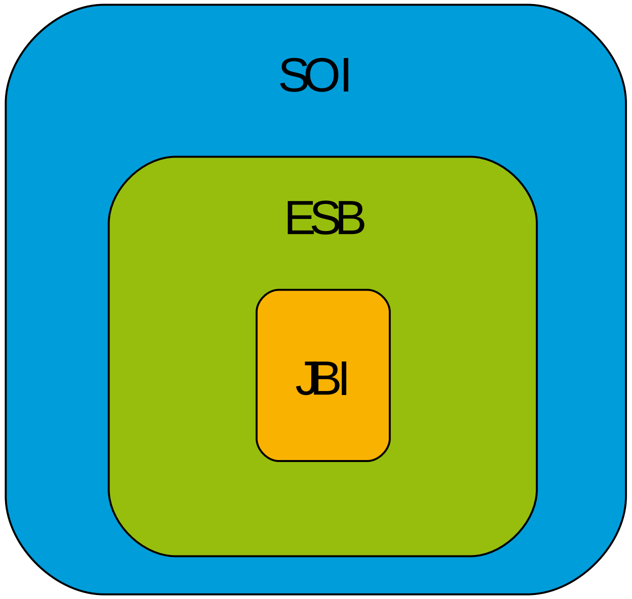
Part II

> JBI Architecture

Part III

> OpenSource ESBs

Conclusions and Outlook



Conclusions

- JBI supports service creation and composition from existing resources in SOA
- JBI based ESBs support:
  - Interchange of services (the plug-in components, SEs and BCs)
  - Less complex service connectivity ($N^2 \rightarrow 2N$ for full connectivity)
  - Less changes when interoperating entities upgrade
  - Versioning of services
  - No vendor lock-in
  - Out-of-the-box JBI binding with GlassFish or Geronimo

- JBI as messaging infrastructure for Java EE
- Usage of available SE's and BC's (no need to know JBI details)

Outlook

- Adoption of JBI 1.0 since 2005
- Adoption of JBI 2.0
- Role of IBM and BEA
- Impact of Apache's SCA (Service Component Architecture) and Apache Tuscany
- Composite applications appear w/o JBI

Q&A

Michael Wisler

Zühlke Engineering AG wis@zuehlke.com

Books/ Documentation:

> Java Business Integration (JBI) 1.0 spec
> http://jcp.org/en/jsr/detail?id=208

> Java Business Integration (JBI) 2.0
> http://jcp.org/en/jsr/detail?id=312

> David A. Chappell. *Enterprise Service Bus*. O'Reilly, 2004

> Gregor Hohpe, Bobby Woolf. *Enterprise Integration Patterns*. Addison-Wesley, 2003
> http://www.enterpriseintegrationpatterns.com

> Whitepapers from Sun Integration
> http://java.sun.com/integration

> Search the Web for "JBI+ESB+Open Source"