



JavaOne™

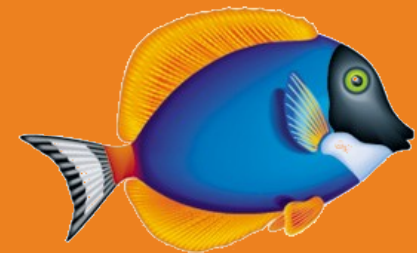
[java.sun.com/javaone](http://java.sun.com/javaone)

# The Best of Both Worlds with Java™ Business Integration and Service Component Architecture

Jos Dirksen, Software Architect

Tijs Rademakers, Software Architect

TS-5870



Show how you can combine the composite application features from SCA with the integration features of Java Business Integration (JBI)



GOAL

# Agenda

- **Introduction**
- Overview of JBI
- Overview of SCA
- Combining JBI and SCA
- Integration: exposing composite services to JBI
- Integration: use JBI CBR between SCA composites
- Summary
- Q&A

# Who are we?

➤ Software Architect at Atos Origin

➤ We love to work with OS integration products:

– ServiceMix



– Mule



– Tuscany etc.

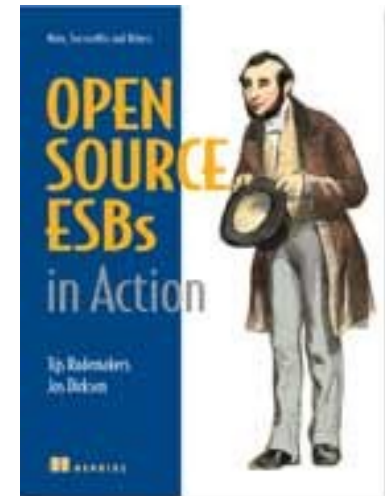


➤ Authors of

– “Open Source ESBs in Action”

– Anticipated publishing date: August 2008

– <http://www.manning.com>



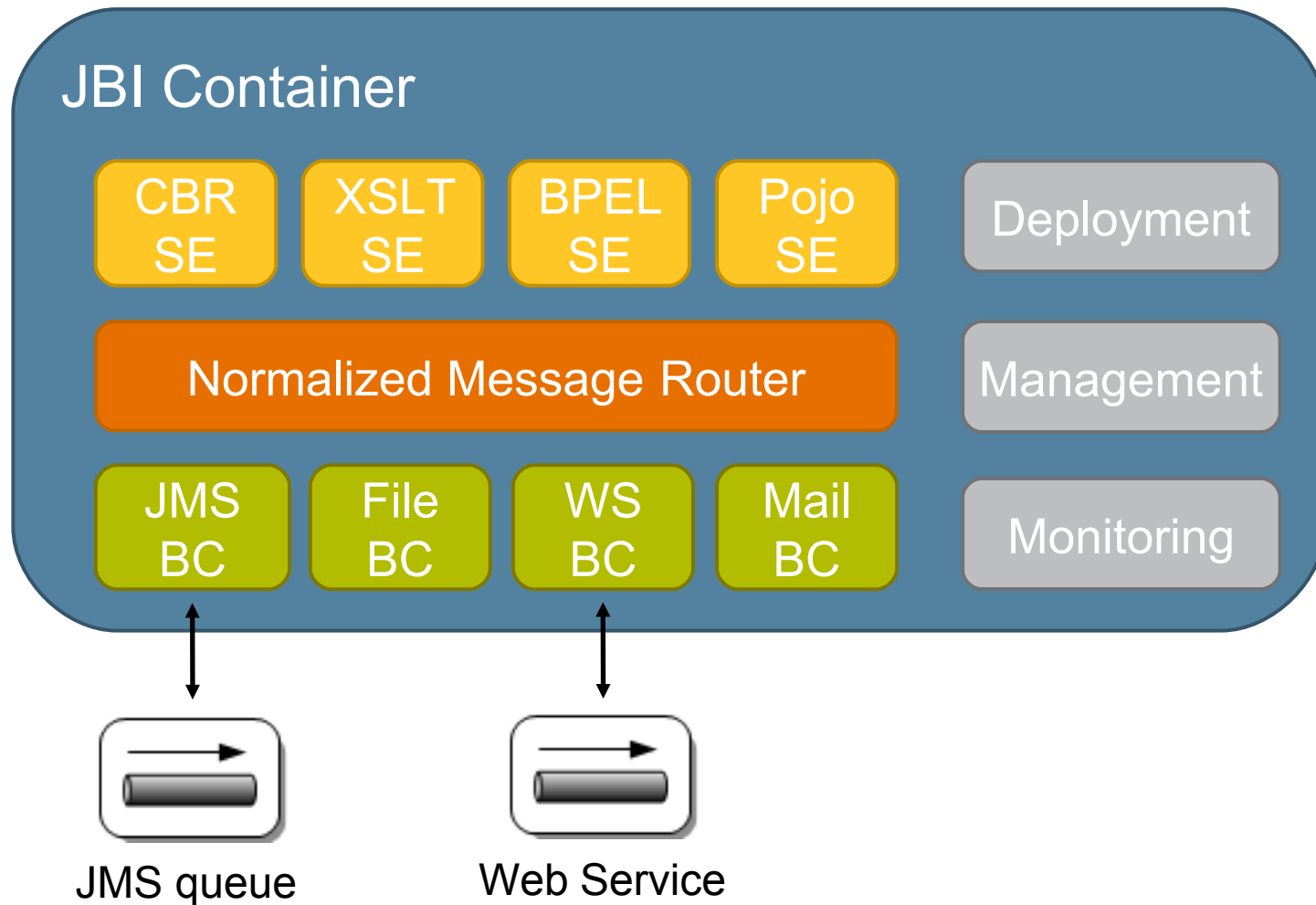
# We are past the dark ages of integration



# Agenda

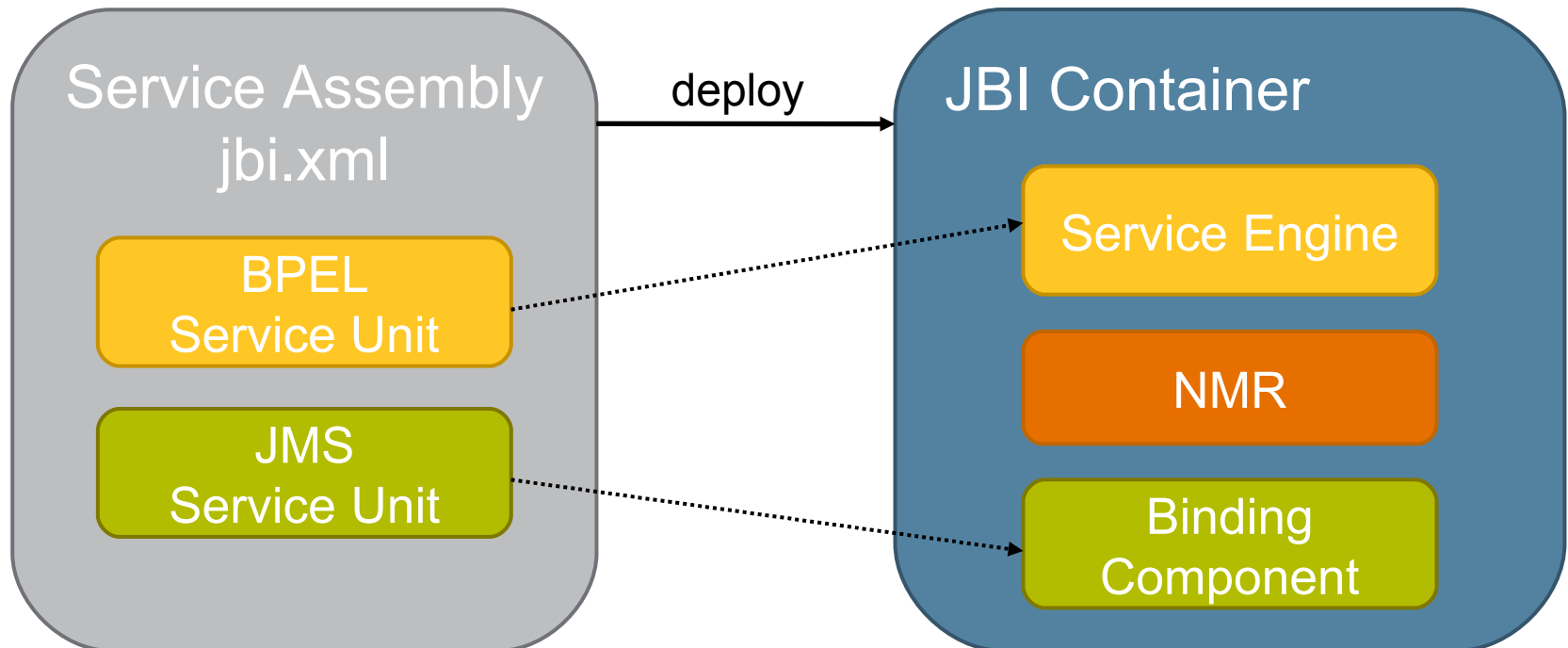
- Introduction
- Overview of JBI
- Overview of SCA
- Combining JBI and SCA
- Integration: exposing composite services to JBI
- Integration: use JBI content based between SCA comp.
- Summary
- Q&A

# JBI container overview



# JBI from a developer perspective

- Developers create Service Units (SU) and Service Assemblies (SA)
- No standard for configuration SUs... Which is a good thing!

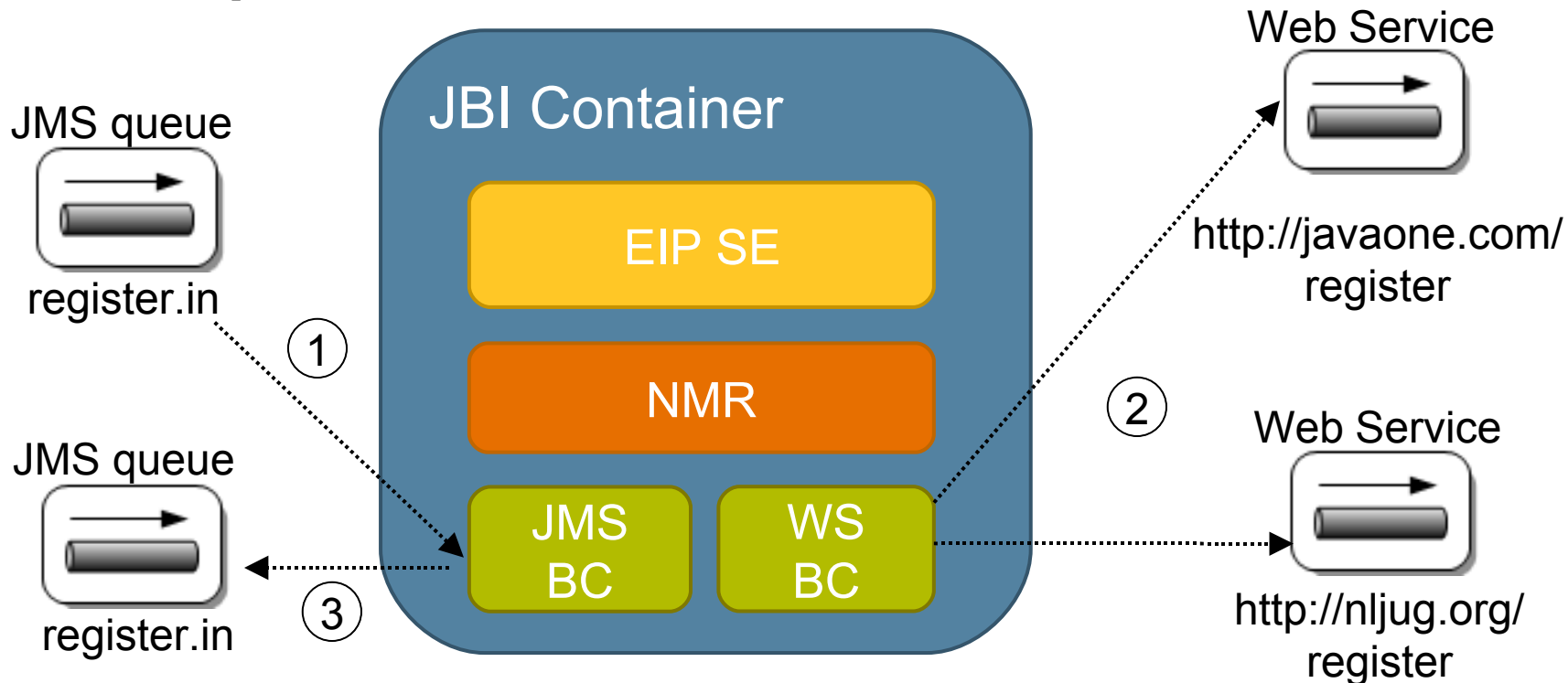


# JBISummary

- Container (JBIS runtime) of containers (SEs, BCs)
- SE: business logic, internal services
- BC: connectivity to external services
- NMR: Loosely coupled, handles routing
- Deployables:
  - JBI Components (SE, BC), over 50 available
  - Service Assembly: collection of SUs
  - Service Units: deployed to a JBI Component
- Management: Java Management Extensions (JMX™) / Ant / Maven
- Everything is hot deployable!



# JBI example



- Receive registration message via JMS queue (JMS BC)
- Determine target web service for registration (EIP SE)
- Invoke the target registration web service (CXF BC)

# We need a configuration...

- Configuration per Service Unit
- Service + endpoint → unique ID in JBI container
- ServiceMix uses XML configuration (Spring based)

```
<beans xmlns:jms="http://servicemix.apache.org/jms/1.0"
      xmlns:esb="http://javaone.com/registration">
  <jms:consumer service="esb:jmsConsumer"
    endpoint="jmsEndpoint"
    targetService="esb:routeRegistration"
    targetEndpoint="routeEndpoint"
    destinationName="register.queue"
    connectionFactory="#connectionFactory"/>
</beans>
```

## JBI Registration Example

- Receive incoming registration message via JMS - Determine the target web service
- Send the registration message to a web service
- Receive registration ID via JMS

DEMO

# Agenda

- Introduction
- Overview of JBI
- Overview of SCA
- Combining JBI and SCA
- Integration: exposing composite services to JBI
- Integration: use JBI CBR between SCA composites
- Summary
- Q&A

# SCA Overview: What is SCA?

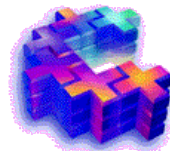
## ➤ Model for:

- Describing service components
- Assembling components into composite components
- Deploying to (distributed) runtime environments



## ➤ Characteristics:

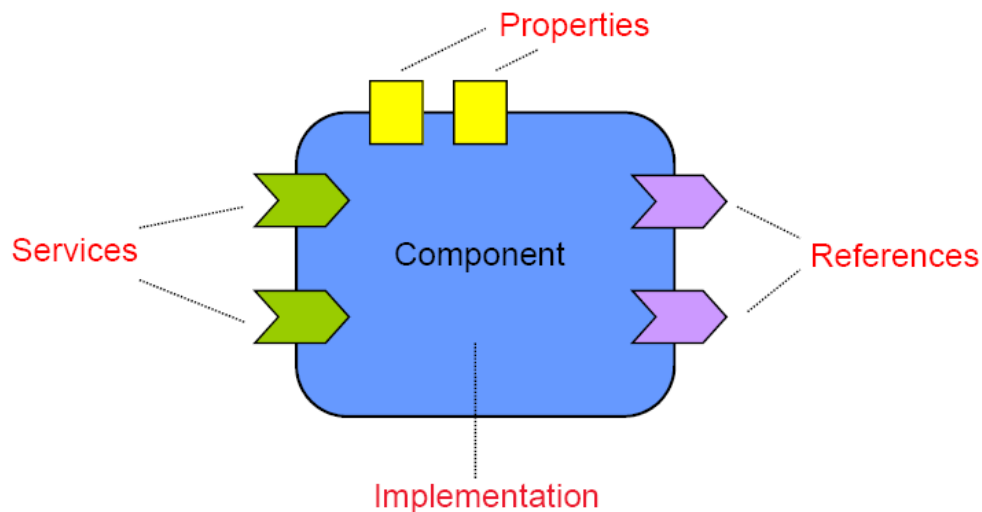
- Services built from new or existing code using SOA principles
- Vendor-neutral – industry wide support
- Language-neutral – large number of languages supported
- Technology-neutral – support for different protocols



# SCA - Component

## ➤ Basic building block, defines:

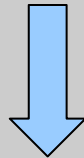
- The implementation: Java technology, Ruby, BPEL etc.
- The services it makes available
- The properties used to configure the implementation class
- Service references



# How to configure a Java platform SCA component?

## ➤ XML configuration in SCDL (annotations also available)

```
<component name="HelloComponent">
  <implementation.java
    class="javaone.HelloService"/>
  <reference name="msgService"
    target="MessageComponent"/>
</component>
```



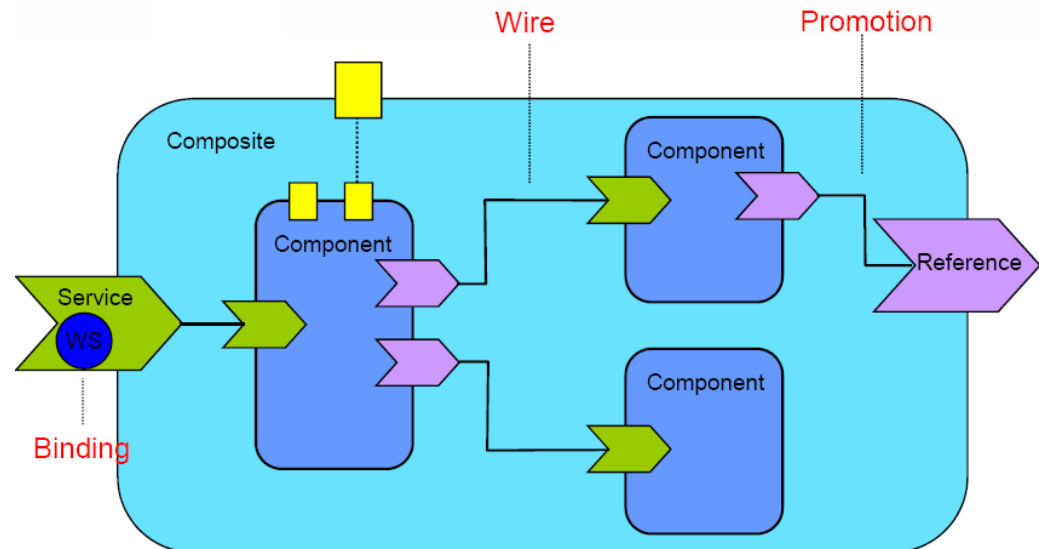
```
<component name="MessageComponent">
  <tuscany:implementation.script
    script="message.rb"/>
</component>
```

```
public class HelloService {
  private MessageIF msgService;
  public hello(String name) {
    return msgService.message()
      + name;
  }
}
```

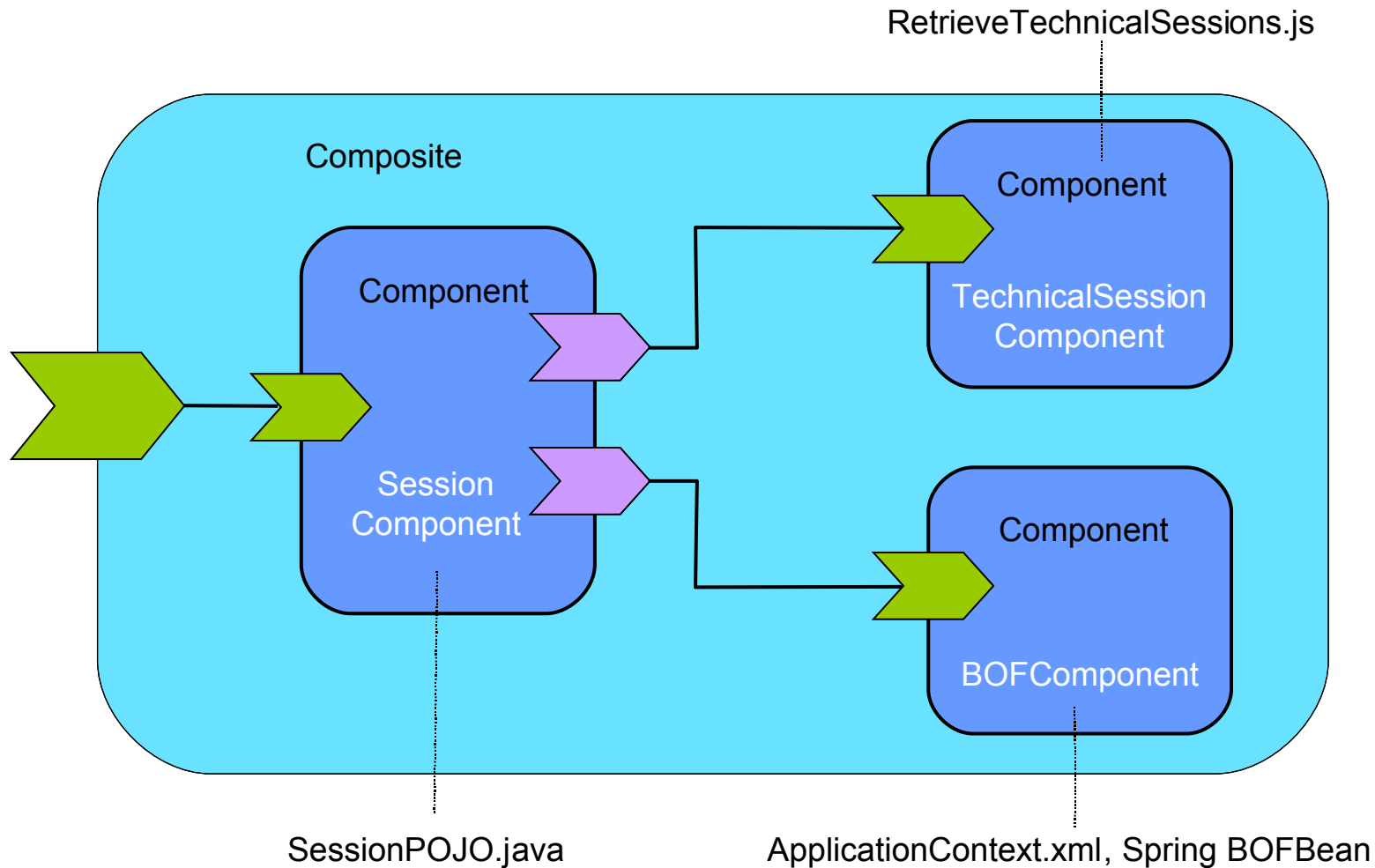
```
def message ()
  return 'hello '
end
```

# SCA - Composite

- Allows for reuse of components
- Expose services from used components
- Must satisfy external dependencies
- Wires connect services/references together



# Let's implement a composite example



## JavaOne<sup>SM</sup> Event Session SCA service

- Session component defined as a POJO
- BOF component defined in Spring
- TS component defined in JavaScript

DEMO

# Agenda

- Introduction
- Overview of JBI
- Overview of SCA
- **Combining JBI and SCA**
- Integration: exposing composite services to JBI
- Integration: use JBI CBR between SCA composites
- Summary
- Q&A

# Should I use JBI or SCA?

## ➤ Arguments for SCA:

- SCA has a strong model for defining composite applications
- Services can be implemented in multiple languages
- Can bind interfaces and references to different technologies

## ➤ Arguments for JBI:

- Defines a standard, loosely coupled, ESB architecture
- SE / BC are exchangeable between JBI implementations
- Provides standard abstraction for all JBI components

# JBI and SCA combined

## ➤ SCA:

- Define the composite applications
- Define the required references
- Define the exposed interfaces

## ➤ JBI:

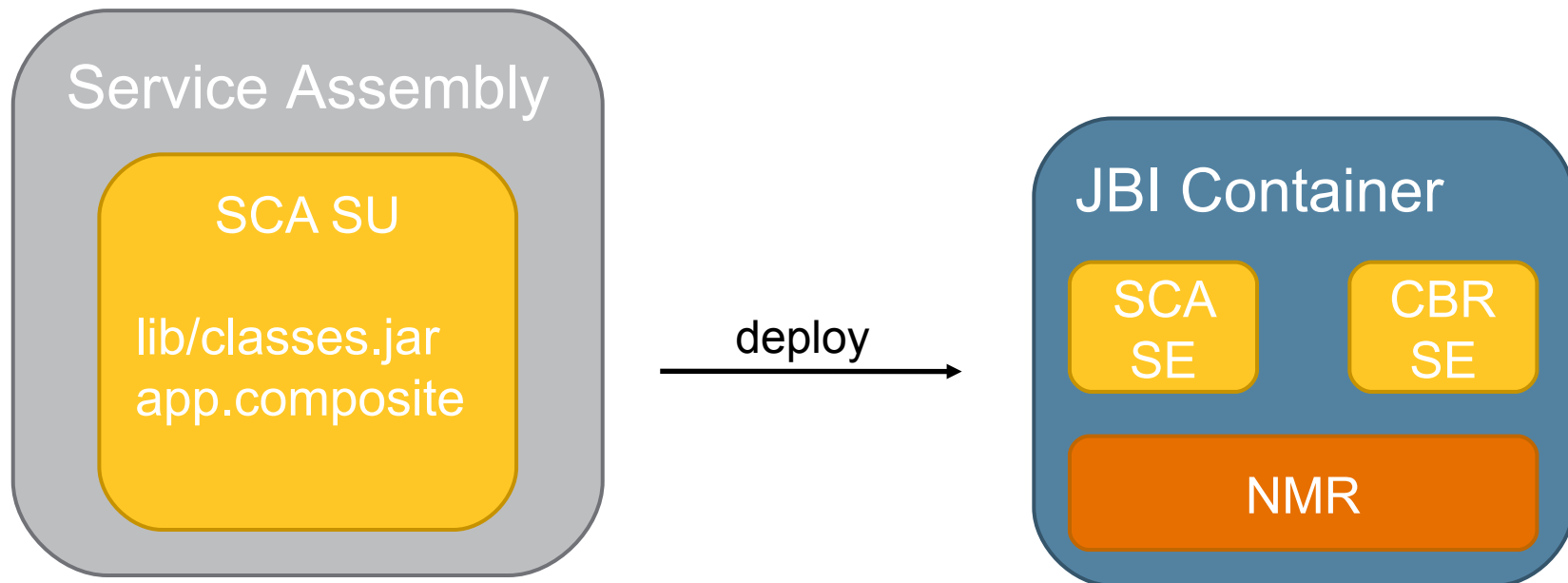
- Resolve the SCA references using the NMR
- Expose the SCA components using the NMR

## ➤ Result:

- All the SEs and BCs can be used from a SCA component
- SCA components can be called from JBI
- SCA applications can be deployed as SU on JBI container

# SCA Service Engine

- To do this we need to create a SCA Service Engine
  - Wrap Tuscany in a Service Engine
  - Create a new JBI deployer that can handle .composite files
  - When a composite file is deployed start a new SCA domain



# Agenda

- Introduction
- Overview of JBI
- Overview of SCA
- Combining JBI and SCA
- **Integration: exposing composite services to JBI**
- **Integration: use JBI CBR between SCA composites**
- Summary
- Q&A

# Exposing composite services to JBI

## > Register SCA service as JBI service-endpoint

```
<service name="EchoService"
  promote="HelloWorldServiceComponent">
  <interface.java interface="helloworld.HelloWorldImpl"/>
  <jbi:binding.jbi namespace="http://javaone.com/2008"
    service="scaAsJbiService"
    endpoint="scaAsJbiEndpoint"/>
</service>
```

## > Reference SCA service as a service endpoint

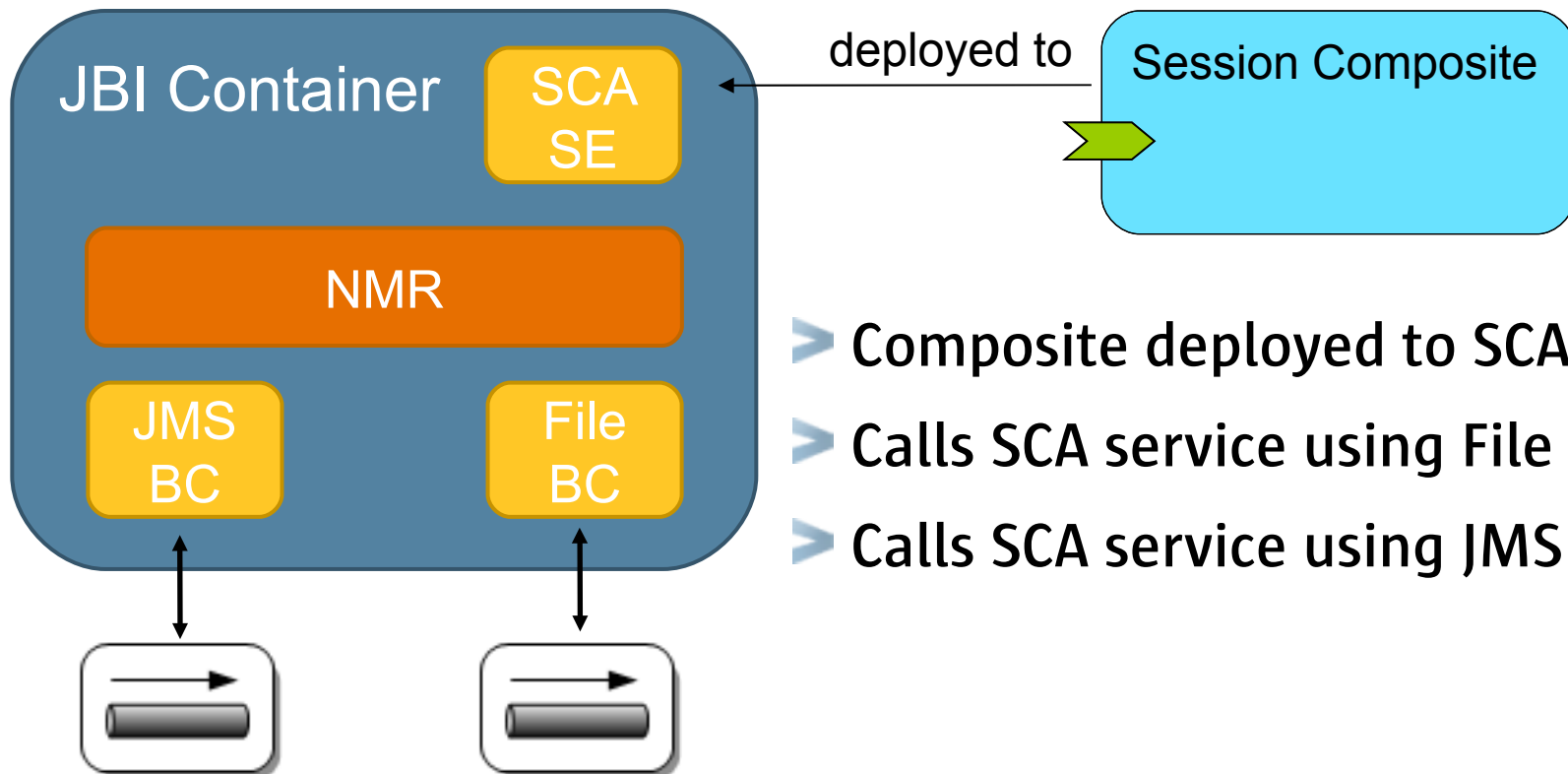
```
<ftp:poller service="jo:ftpPoller" endpoint="poller-endpoint"
  targetService="jo:scaAsJbiService"
  targetEndpoint="scaAsJbiEndpoint"
  uri="ftp://bob:123password@localhost:2121"/>
```

# Service Engine: Technical Details

- Create a Tuscany binding implementation:
  - Handles processing of XML configuration
  - Receive messages from JBI and invoke SCA service
  - Receive invocations from SCA and send as messages to JBI
- In the JBI Service Engine:
  - Read the configuration from the composite
  - Register the exposed SCA Service as JBI service endpoint
- Uses JAXB for Java to XML marshalling

# Exposing composite services example

- Reuse the composite from the SCA example



- Composite deployed to SCA SE
- Calls SCA service using File
- Calls SCA service using JMS

## Exposing composite services to JBI

- Reuse composite from SCA example
- Expose this SCA component to JBI
- Use a File BC to invoke the SCA composite
- Use a JMS BC to invoke the SCA composite

DEMO

# Agenda

- Introduction
- Overview of JBI
- Overview of SCA
- Combining JBI and SCA
- **Integration: exposing composite services to JBI**
- **Integration: use JBI CBR between SCA composites**
- Summary
- Q&A

# Use JBI CBR between SCA composites

- We've called SCA from JBI, but how about the other way

```
<reference name="EchoReference"
    promote="HelloWorldServiceComponent/simpleReference">
  <interface.java interface="helloworld.SimpleReference"/>
  <jbi:binding.jbi service="jo:normalJbiService"
    endpoint="normalJbiEndpoint" />
</reference>
```

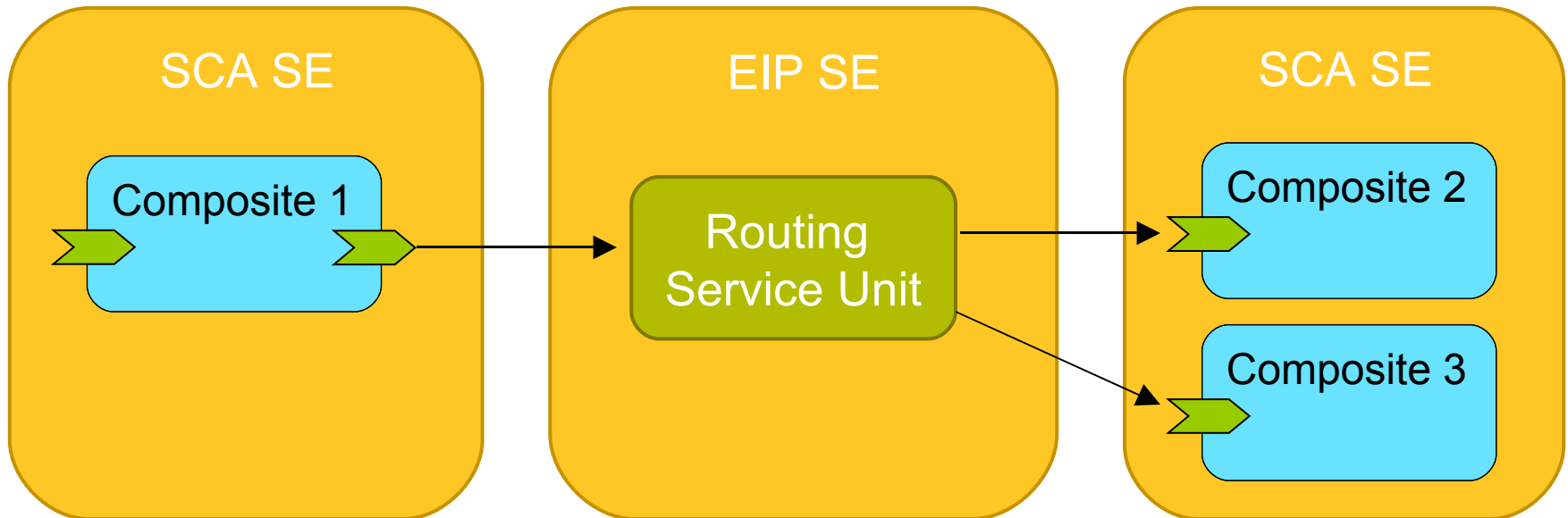
- Call any JBI registered service endpoint

- We can now use:

- Camel for routing through Camel SE
- EIP for routing using EIP SE
- Or any other Binding Component or Service Engine

# Using JBI CBR with SCA services

- Define a binding.jbi in the composite
  - Injects the JBI routing service endpoint
- Define a Service Unit which defines the CBR
  - Handles the real content-based routing



## Use CBR between SCA composites

- Define three SCA composites
- SCA references invoke JBI service endpoints
- Shows usage of a JBI content-based router

DEMO

# Agenda

- Introduction
- Overview of JBI
- Overview of SCA
- Combining JBI and SCA
- **Integration: exposing composite services to JBI**
- **Integration: use JBI CBR between SCA composites**
- Summary
- Q&A

# Summary

- JBI and SCA aren't competitors
  - Each address a different group of developers
- SCA defines a clean composite application model
- JBI defines a strong integration architecture
- Combining SCA and JBI:
  - Re-use of JBI components
  - Flexible programming model of SCA
  - Less fragmentation / avoids re-inventing the wheel
- JBI 2.0 aims for tighter integration with SCA

# For More Information

## ➤ Other JBI and SCA related sessions

- TS-5706 - SCA and Java™ EE Platform: Integration inside
- TS-5850 - SCA: Flexible and Agile Composition of Distributed SOA Applications
- BOF-6586 - Java™ Business Integration 2.0
- PAN-5188 - Open Standards for SOA and Java Technology

## ➤ More information about JBI and SCA

- JSR-208 (JBI 1.0) and JSR-312 (JBI 2.0)
- <http://www.osoa.org>
- <http://servicemix.apache.org>
- <http://incubator.apache.org/tuscany>

# THANK YOU



Jos Dirksen – [jos.dirksen@gmail.com](mailto:jos.dirksen@gmail.com)

Tijs Rademakers – [tijs@apache.org](mailto:tijs@apache.org)

TS-5870

