



J-Fall

11 oktober 2007 Spant! - Bussum



Understanding the hype of SCA

(Service Component Architecture)

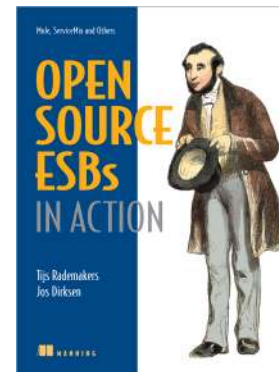
Tijs Rademakers





Whoami

- Software Architect at Atos Origin
- Focus on J2EE and integration
 - J2EE architecture in general
 - ESB and SOA technology – WebSphere Process Server, Mule, ServiceMix, Synapse, Tuscany
- Open Source ESBs in Action
 - www.manning.com/rademakers
 - First chapters available (MEAP)
- For questions and remarks: tijs@apache.org





Agenda

- Introduction into Service Component Architecture
 - Background of SCA
 - SCA terminology
- Examples with Apache Tuscany
 - Introduction into Apache Tuscany
 - Reference POJO's, Spring bean, Scripting
 - Conversations, expose component as web service
- SCA in practice
 - Benefits from an application perspective
 - Tool support (WID and Eclipse SOA tools project)
- Summary, further investigation, questions



SCA in the news



DAVID CHAPPELL'S WEBLOG

ATOM Feed

More heated debate on Service Component Architecture — is it vendor lock-in?

Why Service Component Architecture Is Big News

FRIDAY MARCH 23, 2007



JBI and SCA Are Complimentary!

Service Component Architecture V1.0 Specs released



SCA = Scary Component Architecture?

Service Component Architecture Forges Ahead

Foundations for Service-Oriented Applications: Comparing WCF and SCA





Service Component Architecture

- Architecture model for SOA implementation
 - Glue for connecting your services
 - Abstraction of the service implementations in your SOA
 - **Realization of your service architecture**
- Developed by 18 companies at OSOA
 - BEA, IBM, Interface21, Oracle, Red Hat, SAP, SUN, Tibco
 - SCA Specification 1.0 (March 2007)
- Formal standardization by OASIS



What is SCA?

- Model for:
 - **Describing** service components
 - **Assembling** components into composite components
 - **Deploying** to (distributed) runtime environments
- Characteristics
 - Service components 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

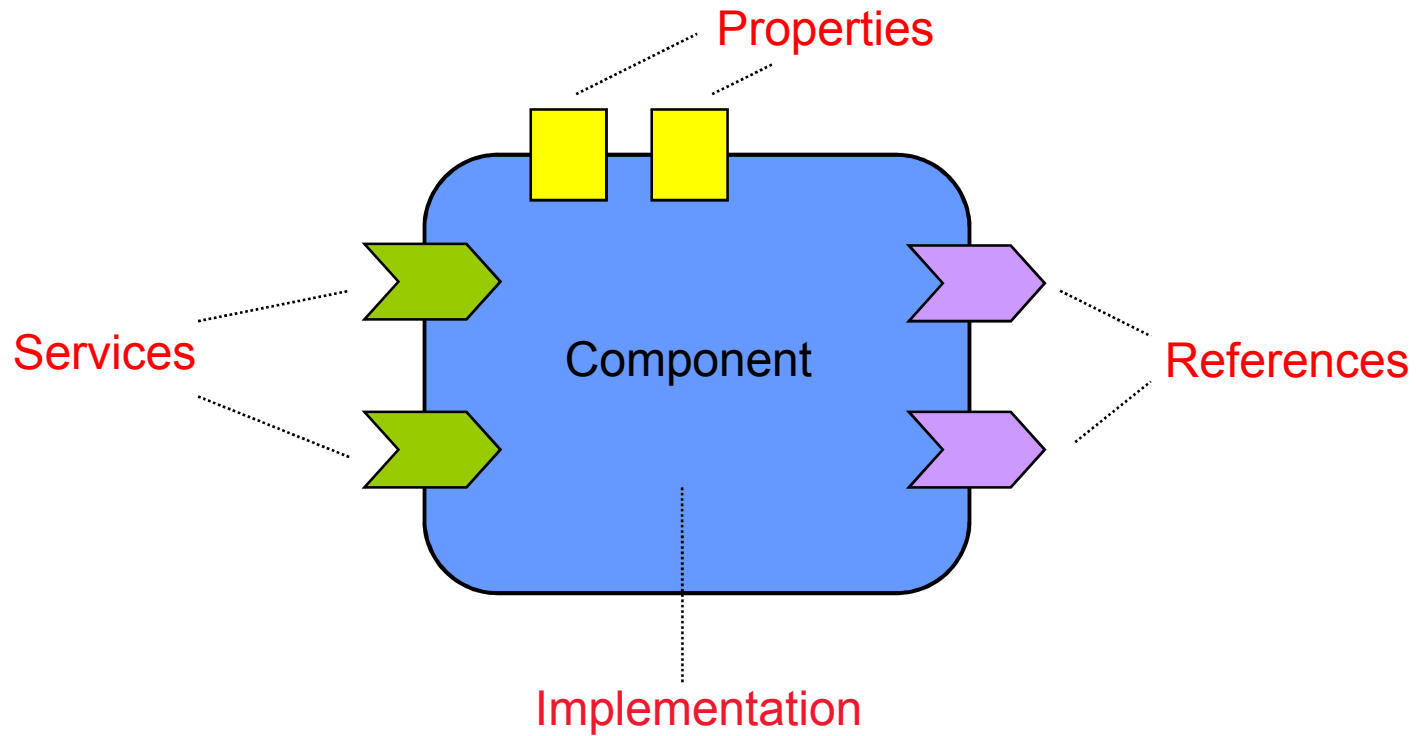


Why use SCA?

- Provides flexibility
 - Program against interfaces
 - Protocol binding configured with SCA
 - Inversion of Control
 - Provides insight in your service architecture
- Promotes reusability
 - Composition of existing service components
 - Sharing of service components across applications
- Consistent architecture model for new and existing technologies / frameworks

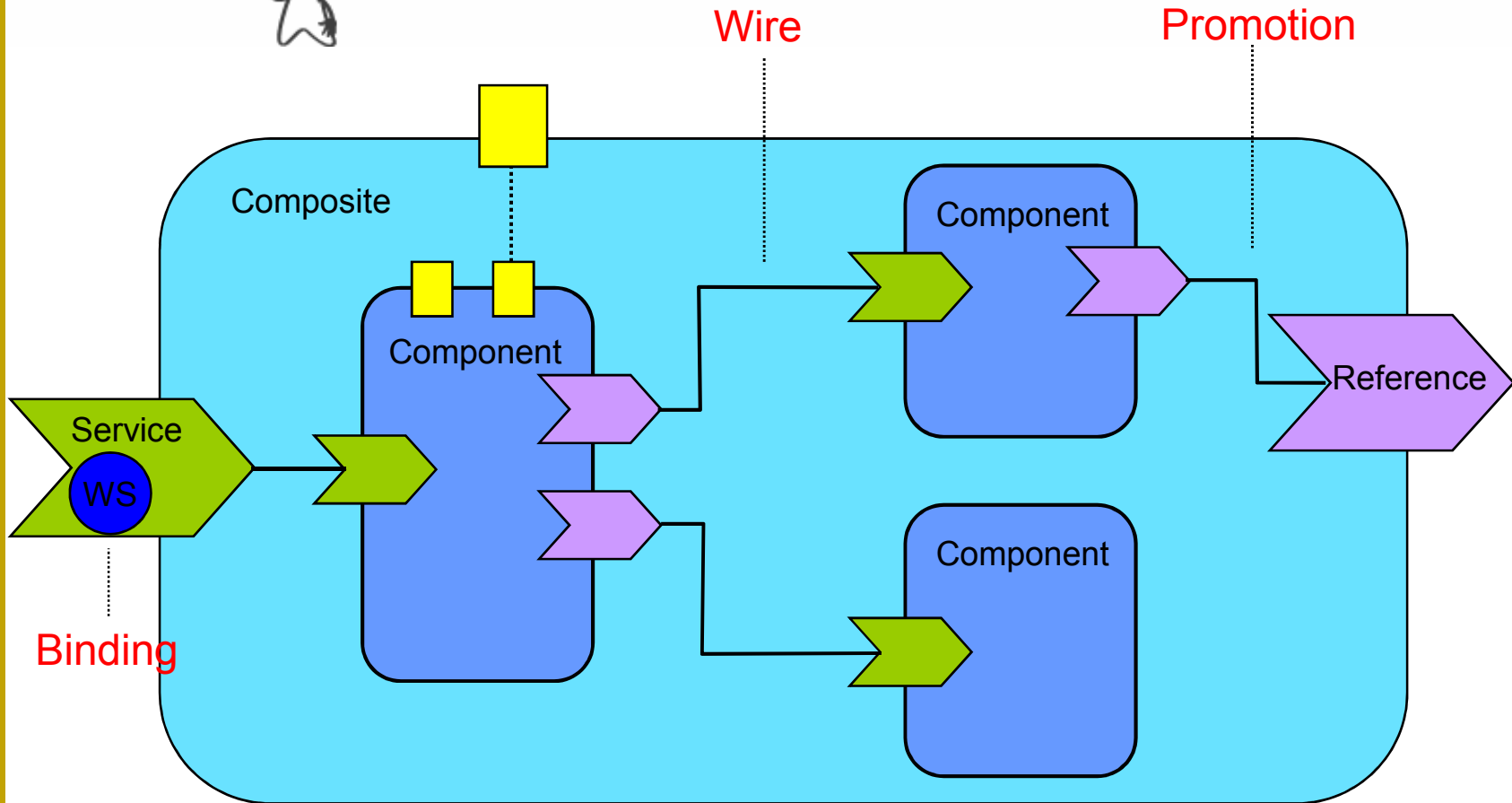


SCA Component



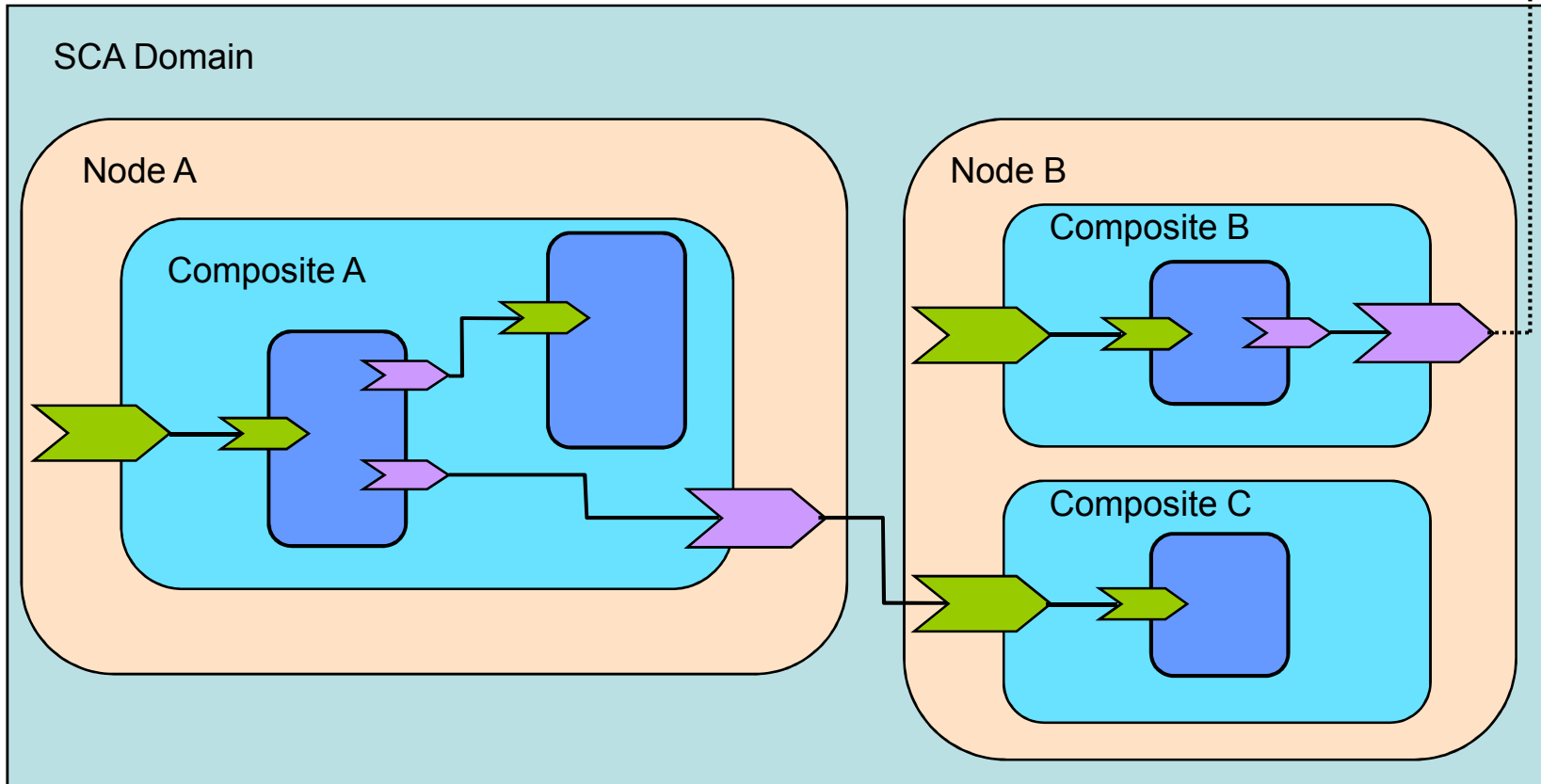
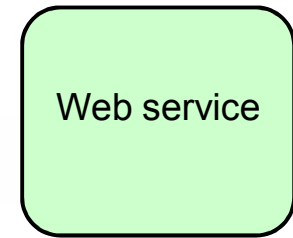


SCA Composite





SCA Domain





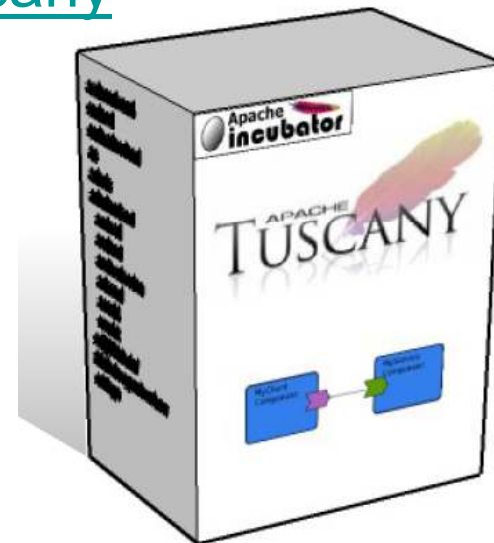
Agenda

- Introduction into Service Component Architecture
 - Background of SCA
 - SCA terminology
- Examples with Apache Tuscany
 - Introduction into Apache Tuscany
 - Reference POJO's, Spring bean, Scripting
 - Conversations, expose component as web service
- SCA in practice
 - Benefits from an application perspective
 - Tool support (WID and Eclipse SOA tools project)
- Summary, further investigation, questions



Overview of Apache Tuscany

- <http://incubator.apache.org/tuscany>
- Implementation of SCA
 - Java
 - Native (C++)
 - PHP (sister project)
- Service Data Objects (SDO)
- Data Access Service (DAS)
- SCA Java version 1.0 release (24 september)





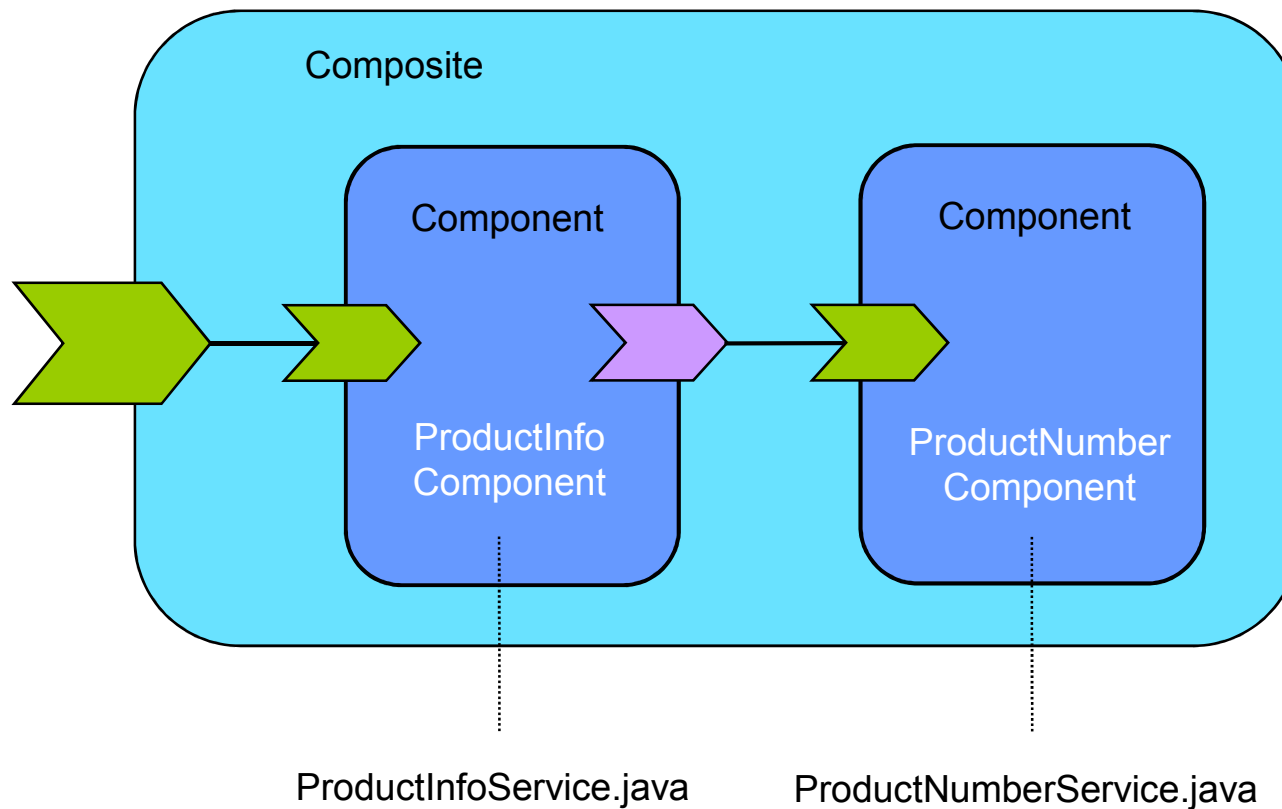
TuSCAny Java 1.0

- SCA 1.0 specification implementation
- Component implementation types
 - Java, Spring beans, scripting
 - BPEL, OSGI, XQuery
- Protocol bindings
 - SCA, Web services, EJB, JMS (in development)
 - JSON-RPC, RSS, ATOM, AJAX
- Hosting environments
 - Embedded (local JVM)
 - Tomcat, Geronimo, Jetty, WAS (SOA feature pack)





POJO composite example





ProductInfo component

```
public class ProductInfoService implements ProductInfoServiceIF {  
  
    private ProductNumberServiceIF productNumberService;  
  
    public String getProductInfo(String productName) {  
        return productNumberService.retrieveProductNumber(productName);  
    }  
  
    @Reference  
    public void setProductNumberService(ProductNumberServiceIF  
        productNumberService) {  
        this.productNumberService = productNumberService;  
    }  
}
```



ProductNumber component

```
public class ProductNumberService implements
    ProductNumberServiceIF {

    public String retrieveProductNumber(String productName) {
        if("Laptop".equals(productName)) {
            return "1111";
        } else {
            return "2222";
        }
    }
}
```



SCDL productInfo.composite

```
<composite xmlns="http://www.osoa.org/xmlns/sca/1.0"
  name="productInfo">

  <component name="ProductInfoComponent">
    <implementation.java
      class="org.nljug.jfall.productinfo.ProductInfoService"/>
    <reference name="productNumberService"
      target="ProductNumberComponent"/>
  </component>

  <component name="ProductNumberComponent">
    <implementation.java
      class="org.nljug.jfall.productinfo.ProductNumberService"/>
  </component>

</composite>
```



But I can do this with Spring!

- Example showed inversion of control pattern
- Programming against interfaces

```
<beans xmlns="http://www.springframework.org/schema/beans">  
  
  <bean id="ProductInfoBean"  
    class="org.nljug.jfall.productinfo.ProductInfoService">  
    <property name="productNumberService">  
      <ref bean="ProductNumberBean"/>  
    </property>  
  </bean>  
  
  <bean id="ProductNumberBean"  
    class="org.nljug.jfall.productinfo.ProductNumberService"/>  
  
</beans>
```

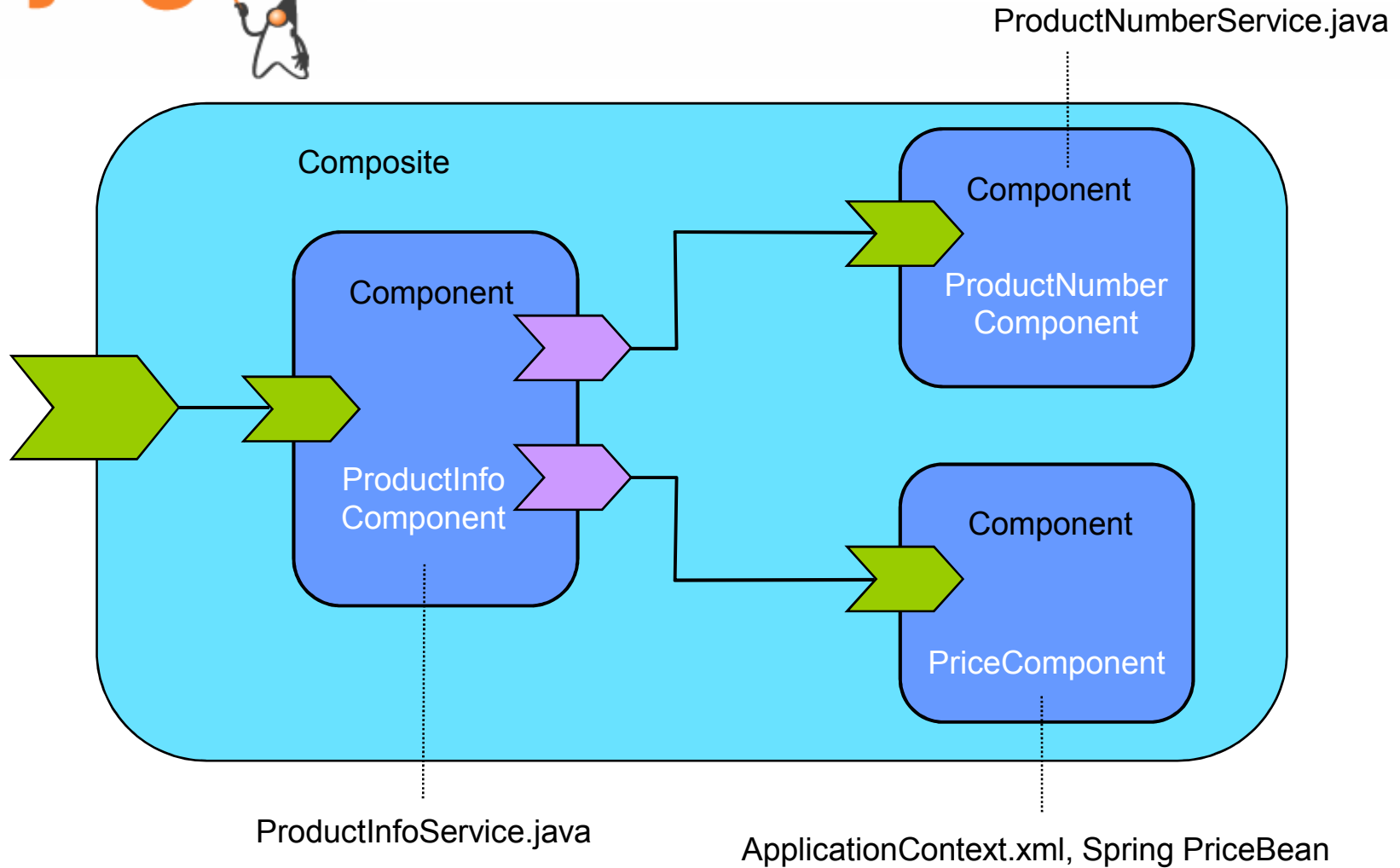


SCA with Spring

- SCA and Spring have similarities
 - Inversion of control, dependency injection
 - Wiring of references
 - Some of the SCA bindings are supported by Spring
- But there are differences
 - Spring is more focused on local wiring
 - SCA is technology neutral (C++, PHP, Scripting etc)
 - SCA exposes service interfaces via multiple bindings
 - SCA supports policies, conversations, async invocations
 - SCA is an open standard
- Spring can be used as implementation type in SCA



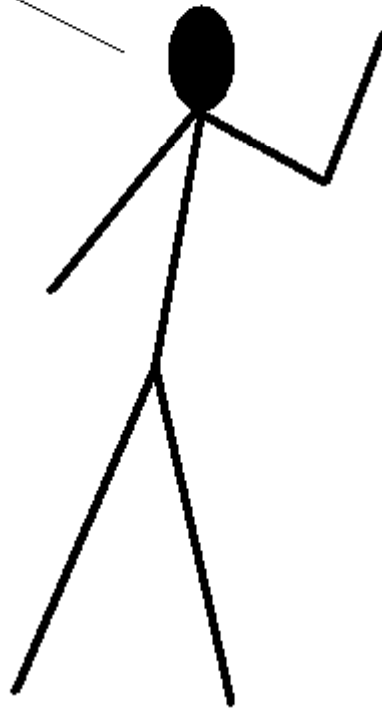
Spring example





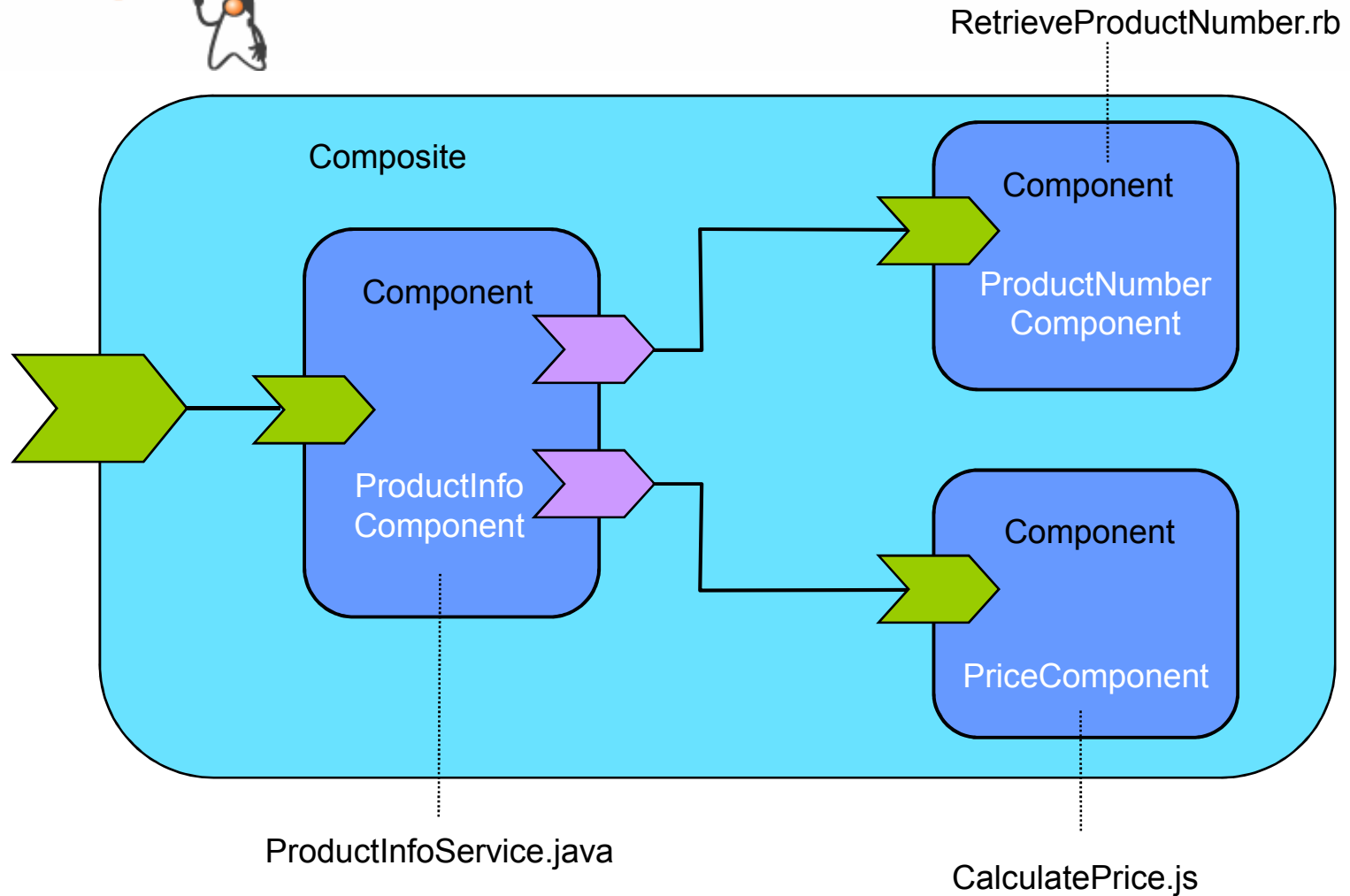
Spring demo

Hello world!





Scripting example





Scripting demo



PHP

TRAINING WHEELS WITHOUT THE BIKE



Support for conversations

@Remotable

@Conversational

```
public interface ProductInfoServiceIF {
```

```
    public String getProductNumber(String productName);
```

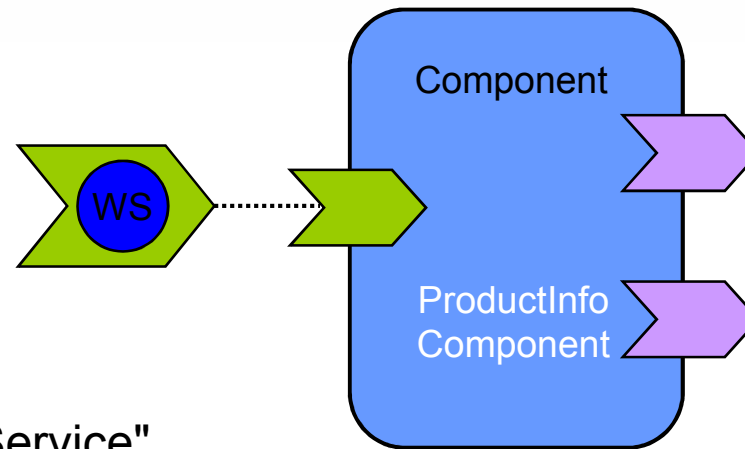
@EndsConversation

```
    public String getPrice();
```

```
}
```



Expose as a web service



```
<service name="ProductInfoService"
  promote="ProductInfoComponent">
  <interface.java
    interface="org.nljug.jfall.productinfo.ProductInfoServiceIF"/>
  <binding.ws uri="http://localhost:8085/ProductInfoService"/>
</service>
```

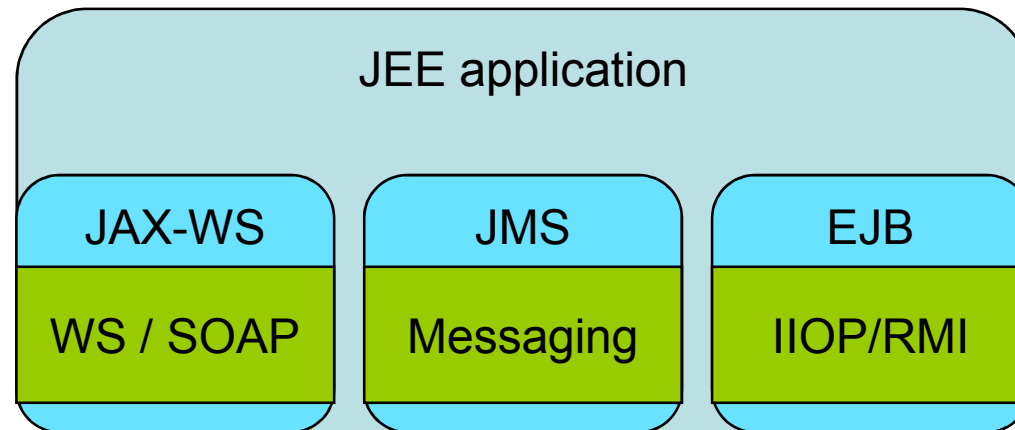


Agenda

- Introduction into Service Component Architecture
 - Background of SCA
 - SCA terminology
 - Introduction into Apache Tuscany
- Examples with Apache Tuscany
 - Reference POJO's, Spring bean, Scripting
 - Conversations, expose component as web service
- **SCA in practice**
 - Benefits from an application perspective
 - Tool support (WID and Eclipse SOA tools project)
- Summary, further investigation, questions



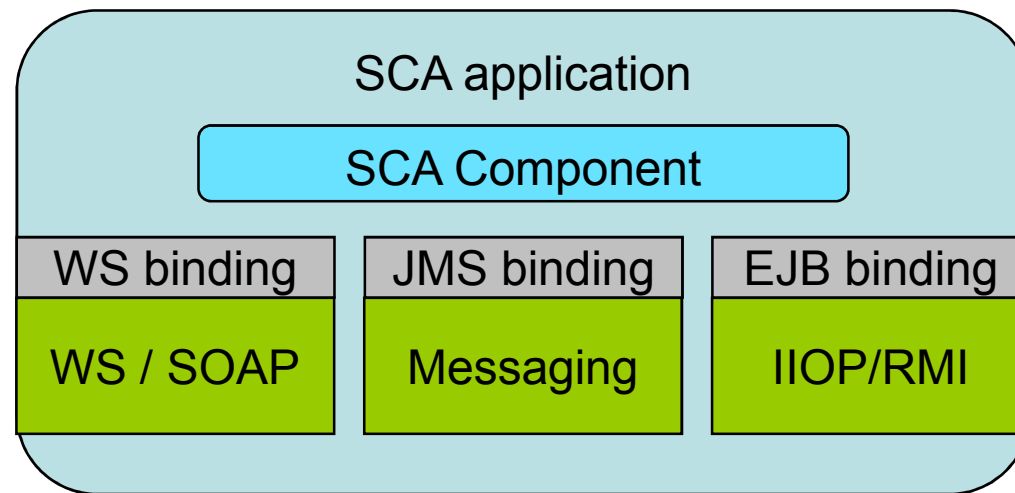
Building services with JEE



- Developer must know several APIs
 - JAX-WS, JMS, EJB, JCA etc.
- Service logic is bound to implementation type



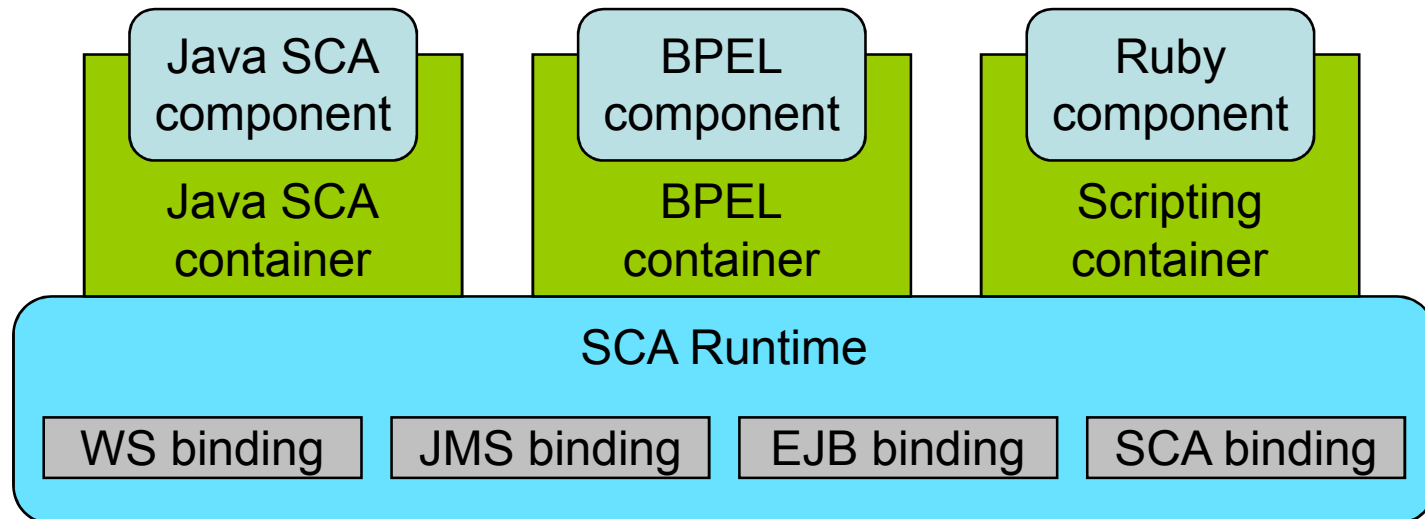
Building services with SCA



- Service logic independent of implementation type
- Developer can focus on the business logic

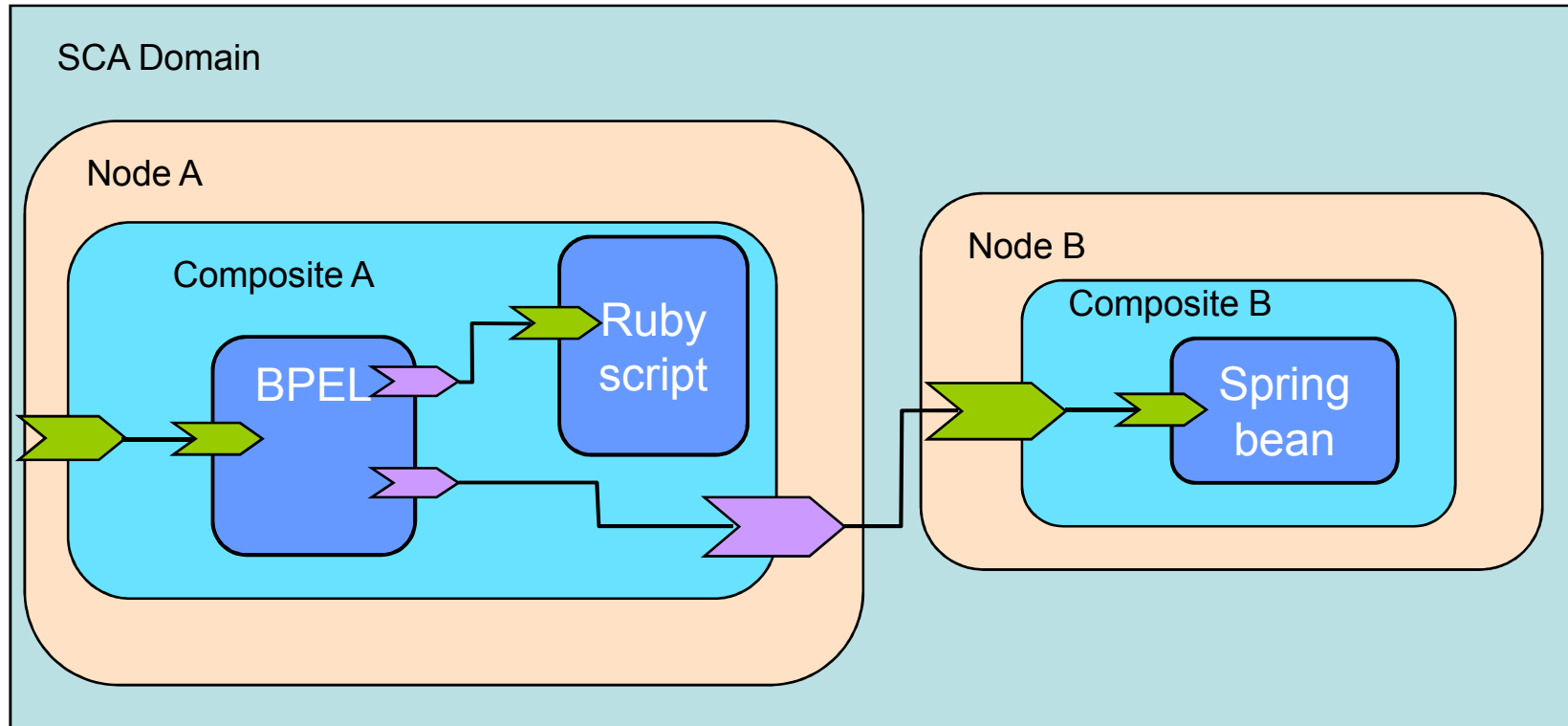


SCA containers





Service architecture





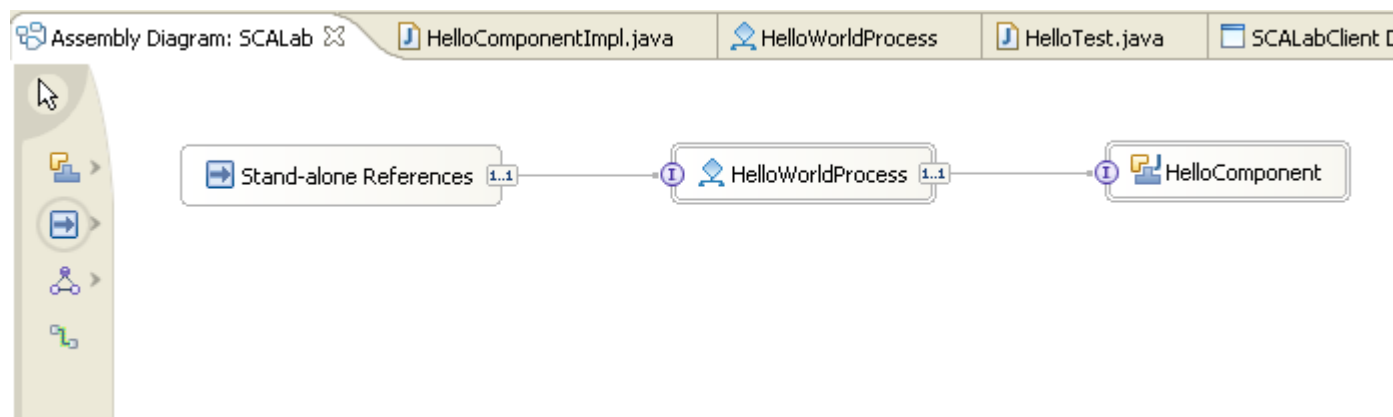
SCA based products

- IBM
 - WebSphere ESB
 - WebSphere Process Server
- Tibco
 - ActiveMatrix Service Grid
- Oracle
 - Fusion
- Open Source
 - Apache Tuscany
 - CodeHaus Fabric3



WebSphere Integration Developer

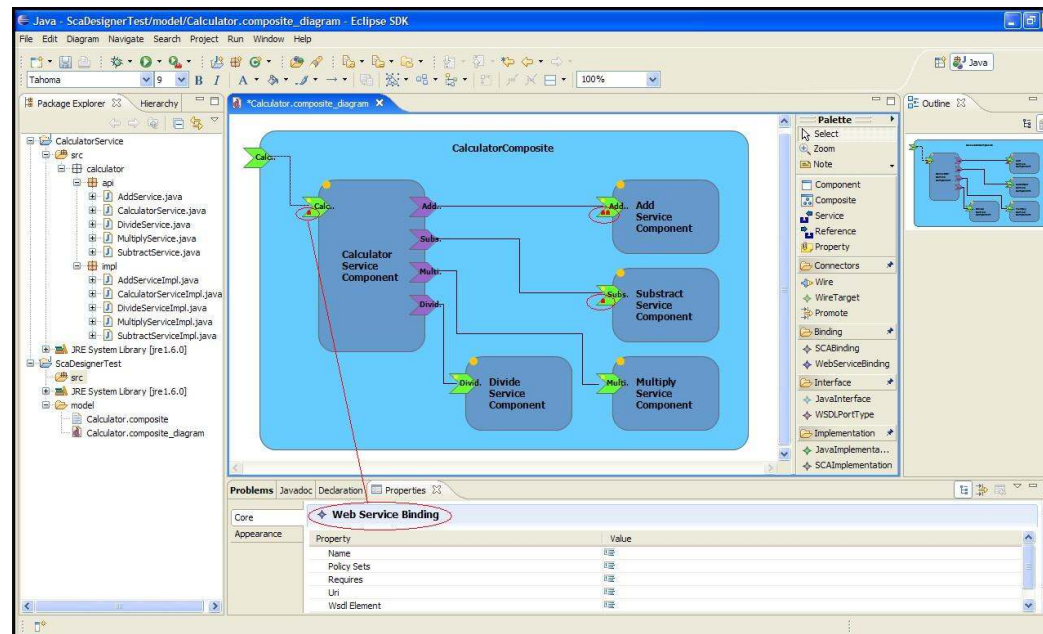
- Development tool for:
 - WebSphere Process Server (Process integration, BPEL)
 - WebSphere ESB
- Architecture is based on SCA / SDO (version 0.9)
- Different naming conventions





Eclipse SOA tools project

- SCA Composite Editor
 - Effort of SAP and Obeo to provide SCA tool support
 - http://wiki.eclipse.org/SCA_Composite_Editor
- Started in June 2007





Agenda

- Introduction into Service Component Architecture
 - Background of SCA
 - SCA terminology
 - Introduction into Apache Tuscany
- Examples with Apache Tuscany
 - Reference POJO's, Spring bean, Scripting
 - Conversations, expose component as web service
- SCA in practice
 - Benefits from an application perspective
 - Tool support (WID and Eclipse SOA tools project)
- **Summary, further investigation, questions**



Summary

- SCA provides a foundation for SOA
 - Standardization in defining service components
 - Provides abstraction to connectivity layer
 - Vendor, language and technology neutral
- Apache Tuscany provides a SCA runtime
 - SCA 1.0 implementation
 - Java, Spring, Scripting, EJB, WS, BPEL support
- What do we need in addition?
 - SCA Domain management needed
 - Support for versioning – maybe based on OSGI?
 - More tools and vendor products



Further investigation

- Lots of articles and blog postings
 - **SCA whitepaper**
http://www.davidchappell.com/articles/Introducing_SCA.pdf
 - **Industry hotshots discussion**
<http://www.infoq.com/news/2007/07/scaproblem>
- OSOA (<http://www.osoa.org>)
- OASIS Open CSA (<http://www.oasis-openca.org>)
- Tuscany website (incubator.apache.org/tuscany)

The screenshot shows the Apache Tuscany website. At the top is the Apache Tuscany logo. Below it is a breadcrumb trail: Apache Tuscany > Home > SCA Overview > SCA Java. The page is divided into two main sections. The left section is titled 'General' and contains links for Home, License, Get Involved, and Found a Bug?. The right section is titled 'Tuscany SCA Java' and contains introductory text: 'Tuscany Java SCA provides the infrastructure for easily developing and running applications user provisioned to, a number of different host environments without much effort. SCA Java is integr...' and 'In Tuscany Java SCA, the SCA domain can run on a single JVM (Node) or it can run across multi concept that bounds all of the components within an application.'



Questions?

