



Service Component Architecture Meets the Java™ Platform, Enterprise Edition (Java™ EE Platform)

Ron Barack

ron.barack@sap.com

SAP SI

<http://www.sdn.sap.com/>

Henning Blohm

henning.blohm@sap.com

SAP AG

TS-41500

Goal of This Session

Learn about the concepts of Service Component Architecture and see how it integrates with Java Platform, Enterprise Edition (Java EE platform).

Disclaimer

About the contents of this presentation

- The content of this presentation is work in progress and captures the current thinking of SCA-enabling a Java EE platform runtime
- SCA Java EE platform integration has not been fully specified as of today
- Visit <http://www.osoa.org/display/Main/SCA+Resources> for more details

Agenda

SCA in a Nutshell

Mapping Java EE Platform to SCA

Why SCA With Java EE Platform

SCA Assembly in a Java EE
Platform Context

Step-by-Step Example

Demo

Agenda

SCA in a Nutshell

Mapping Java EE Platform to SCA

Why SCA With Java EE Platform

SCA Assembly in a Java EE
Platform Context

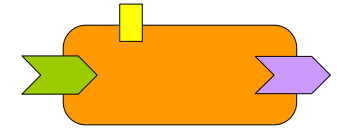
Step-by-Step Example

Demo

SCA in a Nutshell

Lesson I: Service components

Service components



- Serve to abstract component implementations into SCA assembly
- Can have **reference**, **services**, and **properties**
- Can be implemented by a variety of technologies, including SCA composites
- Are declared in **Service Component Description Language** (SCDL)

SCA in a Nutshell

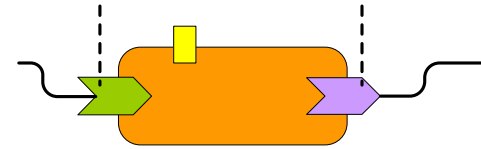
Lesson II: Services and references

Services

- Are offered by service components
- Have an **interface**
- Can be wired to from references
- Can be bound using **bindings**

References

- Are declared on service components
- Have an **interface**
- Can be wired to services
- Can be bound using **bindings**



SCA in a Nutshell

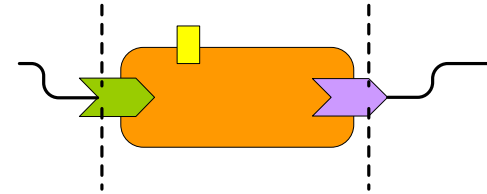
Lesson III: Bindings

Bindings provide non-SCA service technologies to SCA:

- Exposure of services according to some technology/protocol
- Consumption of services provided by some technology/protocol

Examples:

- `binding.ejb`—Consume/expose session beans
- `binding.ws`—Consume/expose web service
- `binding.jms`—Receive/send Java Message Service (JMS) message
- `binding.jca`—Integrate/expose EIS

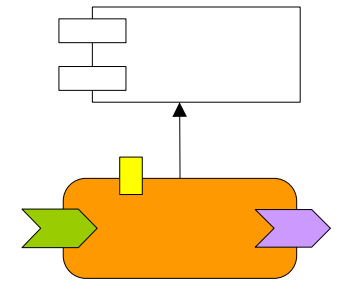


SCA in a Nutshell

Lesson IV: Implementation types

Implementation types

- Link service components to implementation technologies
- Describe how to derive a **Component Type** from an implementation artifact



Examples

- Java class files (`<implementation.java class="..." />`)
- Session beans (`<implementation.ejb ejb-link="..." />`)
- BPEL process (`<implementation.bpel process="..." />`)
- Spring application contexts (`<implementation.spring />`)

SCA in a Nutshell

Lesson V: Composites

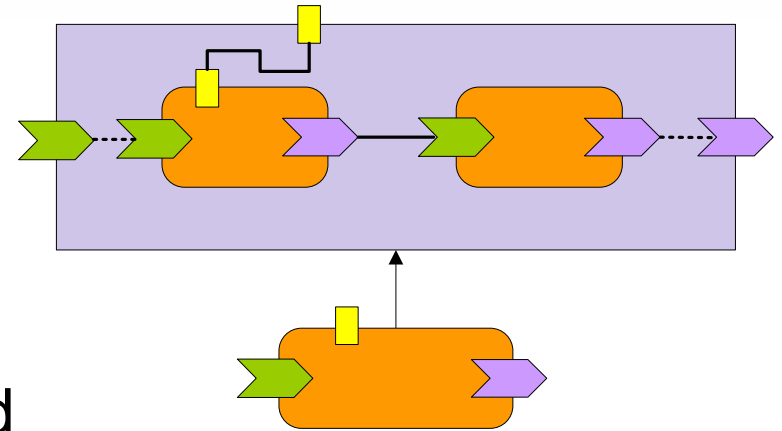
Composites

- Group service components and provide encapsulation
- Provide implementations of service components (recursive assembly:

```
<implementation.composite name="..." />
```

The Composite concept is used in other places

- Important for SCA deployment
- Important for the SCA domain concept
- Important for the Java EE platform integration



SCA in a Nutshell

Lesson VI: Policies

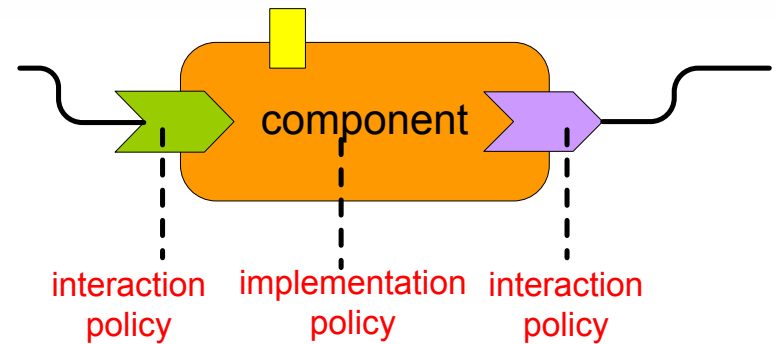
- **Interaction Policies**

- Declaration of service capabilities
- Declaration of reference requirements
- For example: security, reliability

- **Implementation Policies**

- Declaration of constraints between component and runtime (for example, transaction demarcation)
- Simplification of complex policy languages by supporting abstract policies called **intents**:

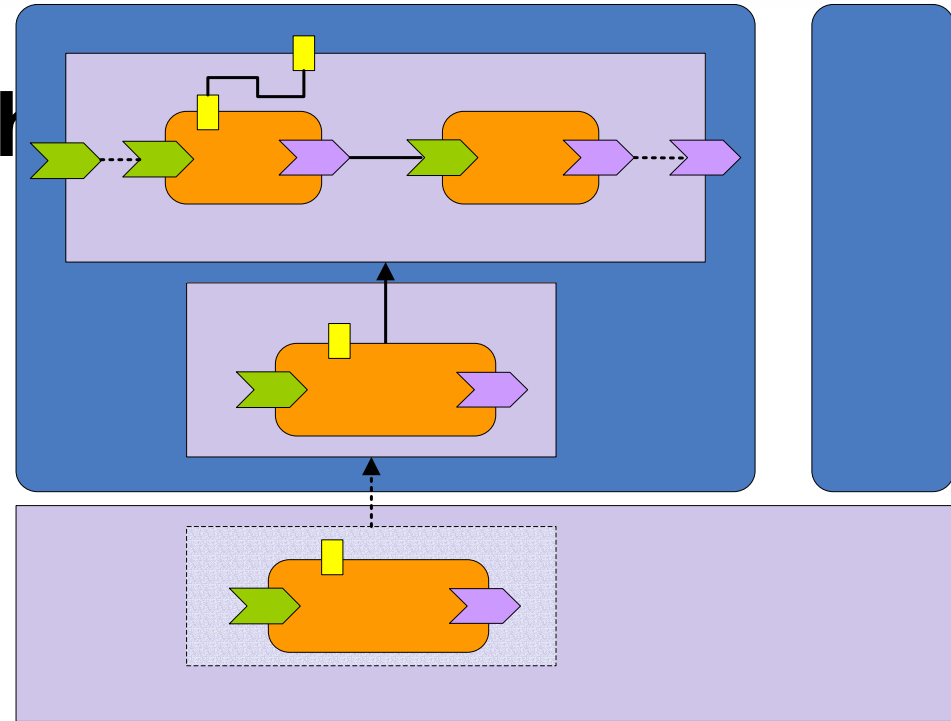
```
@Authentication("message")  
public String hello(String message) {...}
```



SCA in a Nutshell

Lesson VII: Deployment

- Runtime artifacts are deployed as part of **Contributions**
- Contributions are deployed into a **Domain**
- Contributions provide Composites for inclusion into the Domain



Where Is SCA Today?

Standardization progress

- Until recently all SCA work was hosted at <http://www.osoa.org/>
- **1.0 specs now move to OASIS as part of the Open Composite Services Architecture (Open-CSA)**
<http://www.oasis-opencsa.org/>

Agenda

SCA in a Nutshell

Mapping Java EE Platform to SCA

Why SCA With Java EE Platform

SCA Assembly in a Java EE
Platform Context

Step-by-Step Example

Demo

Integration Use-Cases

- Consume SCA-exposed services from Java EE platform components
- Consume session beans from service components
- Expose SCA services as session beans
- Use session beans as Service Component Implementations
- Expose enterprise applications into an SCA domain
- Use recursive SCA assembly in enterprise applications

Mapping Java EE Platform to SCA

Java EE platform implementation types

- Session beans
 - Concepts of services, references, properties exist
 - In Java EE platform no separation of component type and component
- Web components, message-driven beans
 - Service exposure not appropriate
 - But have references and properties

Mapping Java EE Platform to SCA

Java EE platform bindings

- Several Java EE technologies map to bindings:
 - Java Message Service (JMS)
 - Session Beans
 - Local consumption and exposure
 - Co-located and remote consumption and exposure
 - J2EE™ Connector Architecture/EIS (binding spec is work in progress)

Mapping Java EE Platform to SCA

Packaging and deployment

- Java EE platform suggests contribution packaging
 - Enterprise application packages (.ear)
 - Single modules (.war, .jar, .rar)
- Packages enriched with SCA meta-data for:
 - Internal assembly
 - Domain assembly
- SCA metadata may be applied over existing packages

Agenda

SCA in a Nutshell

Mapping Java EE Platform to SCA

Why SCA With Java EE Platform

SCA Assembly in a Java EE
Platform Context

Step-by-Step Example

Demo

Why SCA With Java EE Platform?

Expected value-add of SCA

- Additional implementation types and bindings on the application level
 - BPEL, Spring, ...
 - More to expect: For example, ESB style interactions
- Move protocol specifics to configuration via bindings
- Cross-Application Assembly
 - Domain assembly extends Enterprise App Assembly to the cross-app level
- ...

Evolutionary Approach

Style of the integration

- Preserve investment in Java EE technology and Java EE platform knowledge
 - Java EE platform components preserve their client view
 - SCA annotations can be used to add SCA concepts in a mix with Java EE platform annotations
 - SCA assembly as addition to Java EE platform assembly (“Java EE platform assembly as a dialect of SCA assembly”)

Agenda

SCA in a Nutshell

Mapping Java EE Platform to SCA

Why SCA With Java EE Platform

**SCA Assembly in a Java EE
Platform Context**

Step-by-Step Example

Demo

SCA Assembly in a Java EE Platform Context

Representing applications and modules

- Java EE platform application and module assembly to be preserved
 - `META-INF/application.composite` provides root of ear-application assembly
 - Similarly for Enterprise JavaBeans™ (EJB™) software modules and Web components resp:
 - `META-INF/ejb-jar.composite`
 - `WEB-INF/web.composite`
 - Single-Module deployments start from module composites
- Java EE platform components can be used by declaration in application-level or module-level composites

SCA Assembly in a Java EE Platform Context

More composites and domain-level assembly

- Modules can provide further SCA artifacts to be used
 - As composite implementations
 - For inclusion into other composites
- Domain-level composition to be supported by declaring deployment composites in:
 - `META-INF/sca-contribution.xml`

Agenda

SCA in a Nutshell

Mapping Java EE Platform to SCA

Why SCA With Java EE Platform

SCA Assembly in a Java EE
Platform Context

Step-by-Step Example

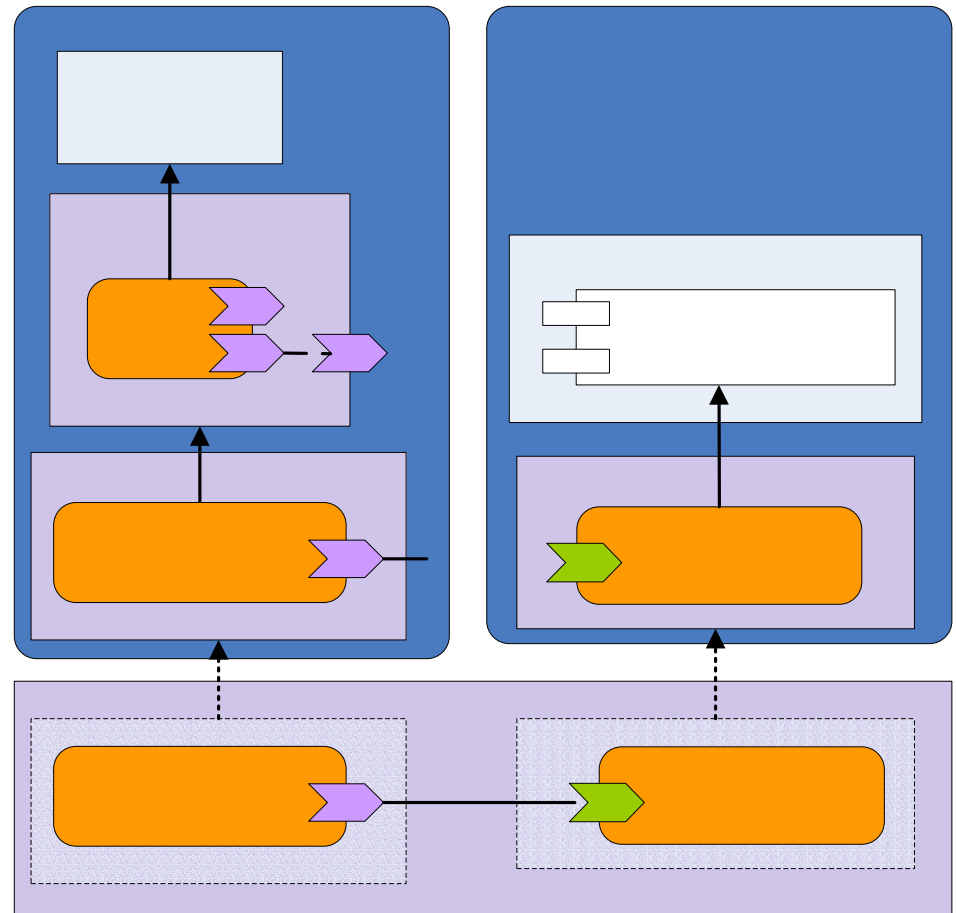
Demo

Step-by-Step 0

What we want to achieve

Goal

Web App exposed via special deployment composite to domain and wired to service implemented by an EJB component model



Step-by-Step I

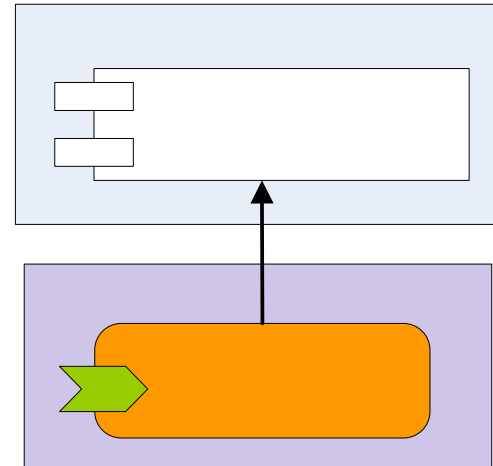
Composite with a session bean component

```

<?xml version="1.0" encoding="UTF-8"?>
<composite name="beancomposite"
  targetNamespace="http://www.sample.org"
  xmlns="http://www.oesa.org/xmlns/sca/1.0">

  <component name="org.sample.Accounting">
    <implementation.ejb
      ejb-link="module.jar#RemotableAccountingBean"/>
  </component>
</composite>

```



Step-by-Step II

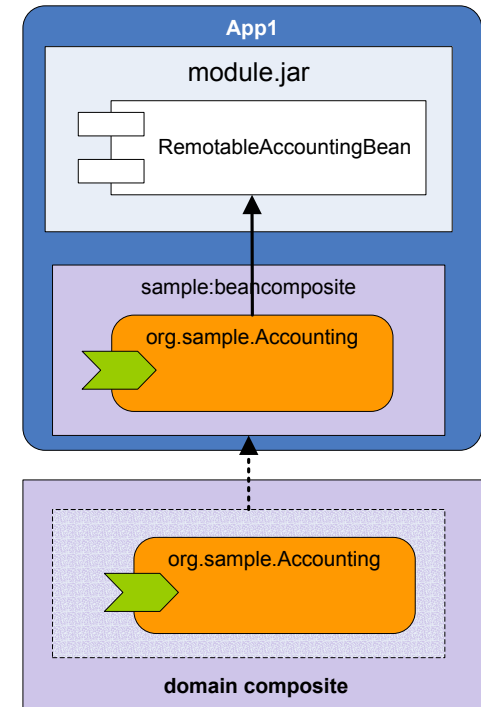
Including it into the domain

Declare `sample:beancomposite` as Deployment Composite in META-INF/sca-contribution.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<contribution
```

```
    xmlns="http://www.oxa.org/xmlns/sca/1.0"
    xmlns:sample="http://www.sample.org" >
```

```
    <deployable composite="sample:beancomposite"/>
</contribution>
```



Step-by-Step III

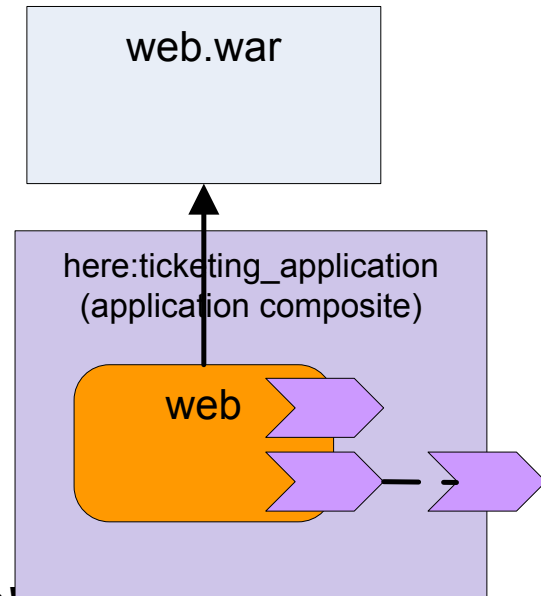
Web component in application composite

In META-INF/application.composite:

```
<?xml version="1.0" encoding="UTF-8"?>
<composite name="ticketing_application"
  targetNamespace="http://www.acme.com"
  xmlns="http://www.osoa.org/xmlns/sca/1.0">
```

```
  <component name="web">
    <implementation.web module="web.war"/>
  </component>
  <reference name="AccountReporting"
    promote="web/AccountReporting"/>
```

```
</composite>
```



Step-by-Step IV

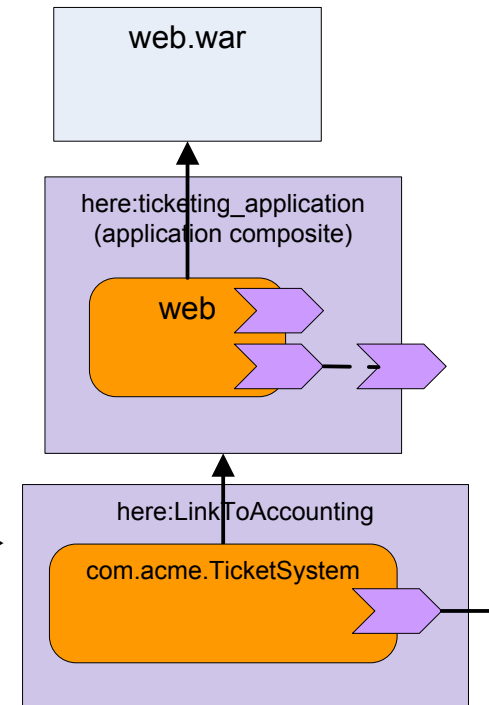
Facade for domain

```

<?xml version="1.0" encoding="UTF-8"?>
<composite name="LinkToAccounting"
  targetNamespace="http://www.acme.com"
  xmlns:here="http://www.acme.com"
  xmlns="http://www.osoa.org/xmlns/sca/1.0">

  <component name="com.acme.TicketSystem">
    <implementation.composite
      name="here:ticketing_application"/>
    <reference name="AccountReporting"
      target="org.sample.Accounting/AccountReporting"/>
  </component>
</composite>

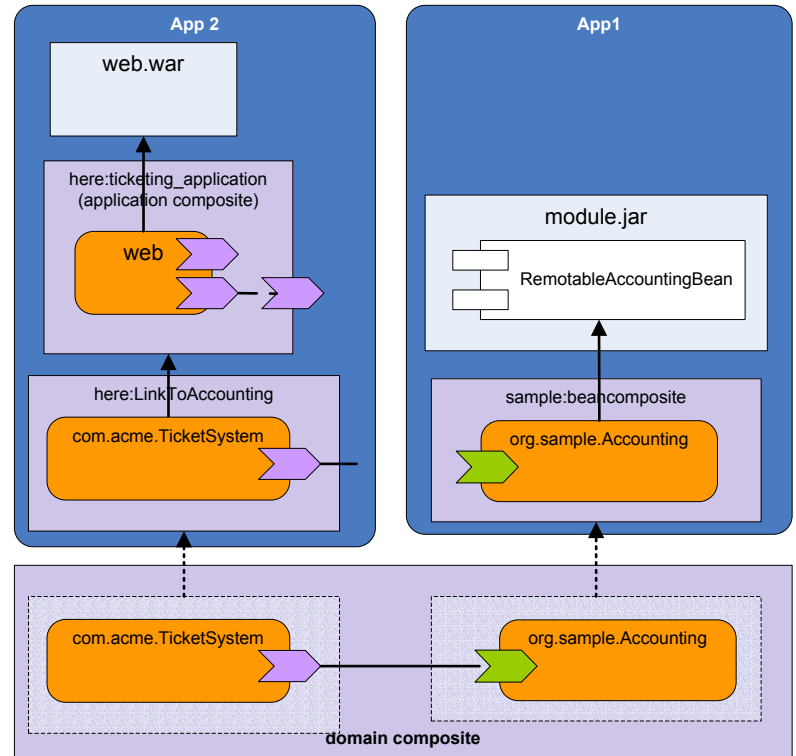
```



Step-by-Step V

Bringing it together at the domain

Declare
here:LinkToAccounting
 as deployment composite



Agenda

SCA in a Nutshell

Mapping Java EE Platform to SCA

Why SCA With Java EE Platform

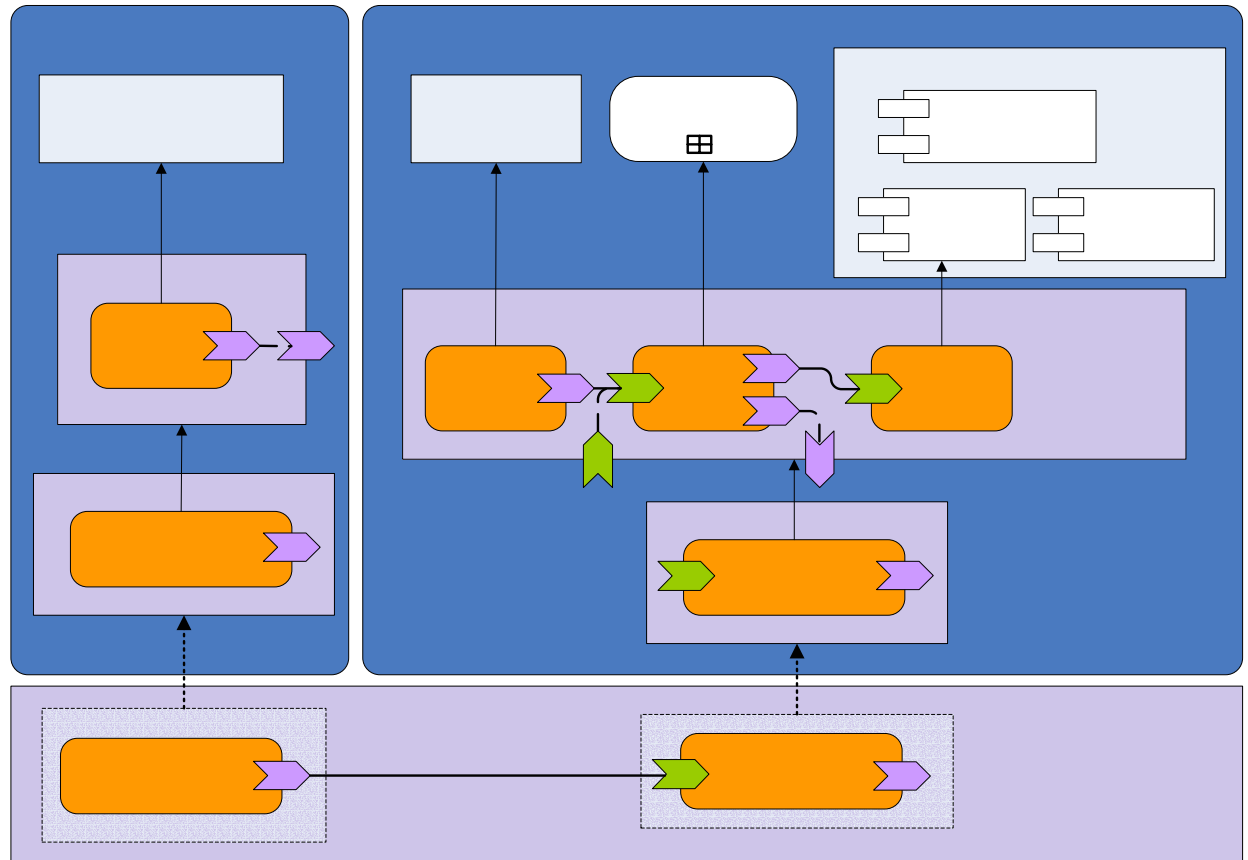
SCA Assembly in a Java EE
Platform Context

Step-by-Step Example

Demo

Demo—Scenario Overview

A web shop with a BPEL implemented order process and domain-level extension





DEMO



Summary

- Abstract SCA concepts can be mapped to Java EE platform
- SCA-enabling of Java EE platform open evolutionary path to Java EE platform products
- Potential for integration-standards with non-Java EE technologies
- Domain concept allows to express cross-application coupling

For More Information

- SCA Sessions
 - TS-8554: Building, Assembling, Deploying Composite Services Applications
 - TS-8194: SCA and Spring
 - TS-80955: Open for Business: Using Open Standards to Make SOAs Safe for Developers
- TS-8835: Panel on SDO and SCA
- BOF-8238: Building Composite Services Applications
- Links:
 - OSOA: <http://www.osoa.org/>
 - SCA specifications
<http://www.osoa.org/display/Main/Service+Component+Architecture+Specifications>
 - SCA white papers
<http://www.osoa.org/display/Main/SCA+Resources>



Q&A

<code />



Service Component Architecture Meets the Java™ Platform, Enterprise Edition (Java™ EE Platform)

Ron Barack

ron.barack@sap.com

SAP SI

<http://www.sdn.sap.com/>

Henning Blohm

henning.blohm@sap.com

SAP AG

TS-41500