

Service Component Architecture (SCA) and Service Data Objects (SDO)

Apache Tuscan: Open Source SOA

Jean-Sebastien Delfino
jsdelfino@apache.org

Agenda

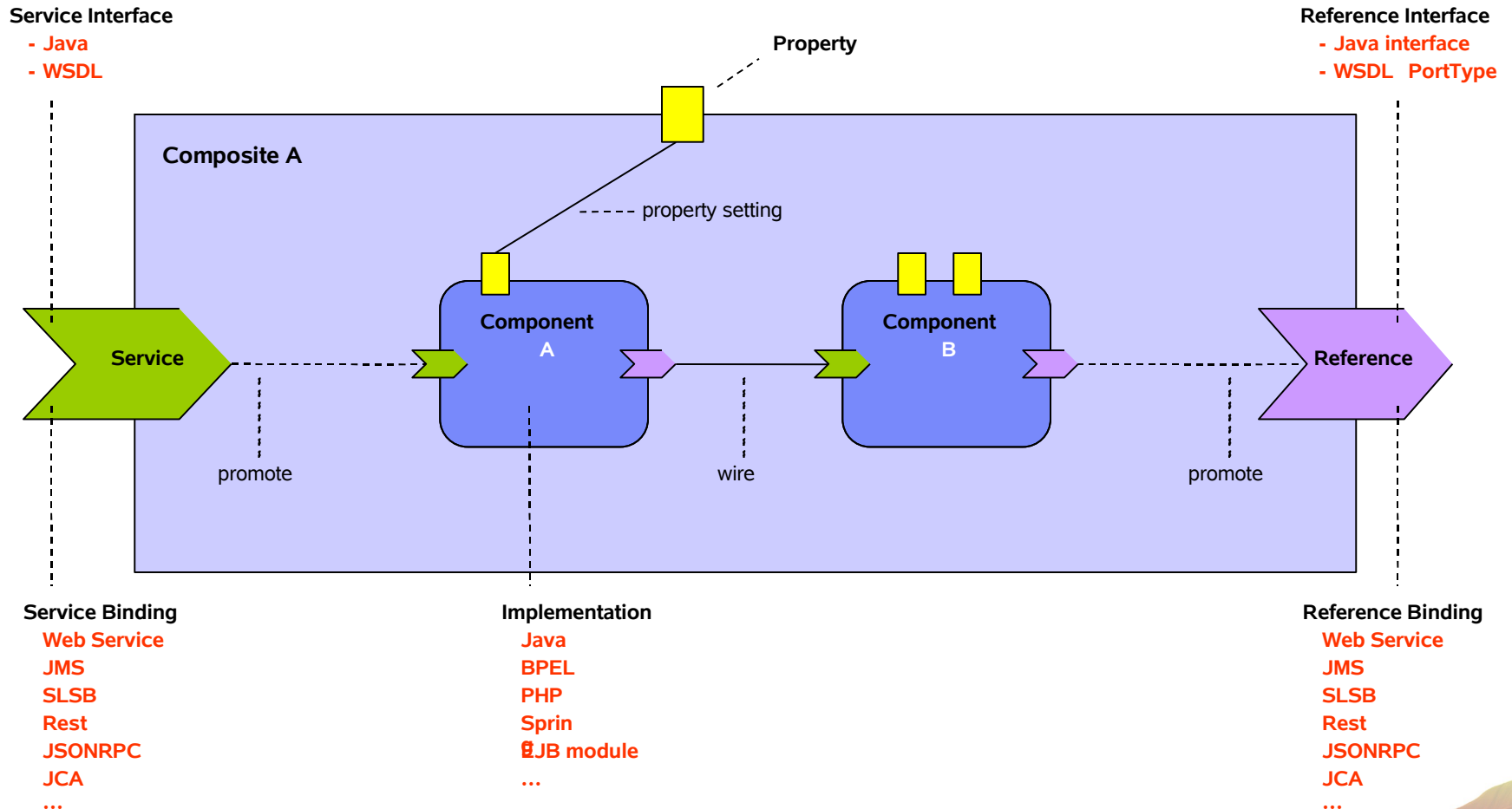
- **What are SCA and SDO?**
- **SCA: Programming Model Overview**
- **Apache Tuscany: Open Source SOA Runtime**
- **Latest SCA and SDO releases and work in progress**
- **Demo**
- **Q&A**

What is SCA?

- Architectural framework for service definition, assembly, and composition
- Separation of implementation technology choices from business service definitions
- Provides a variety of protocol bindings, implementation types, interface definitions and data bindings
- Extensible framework for defining new protocol bindings, implementation types, interface definitions and data bindings
- Multiple language support
- Intent-based policy support for applying qualities of service

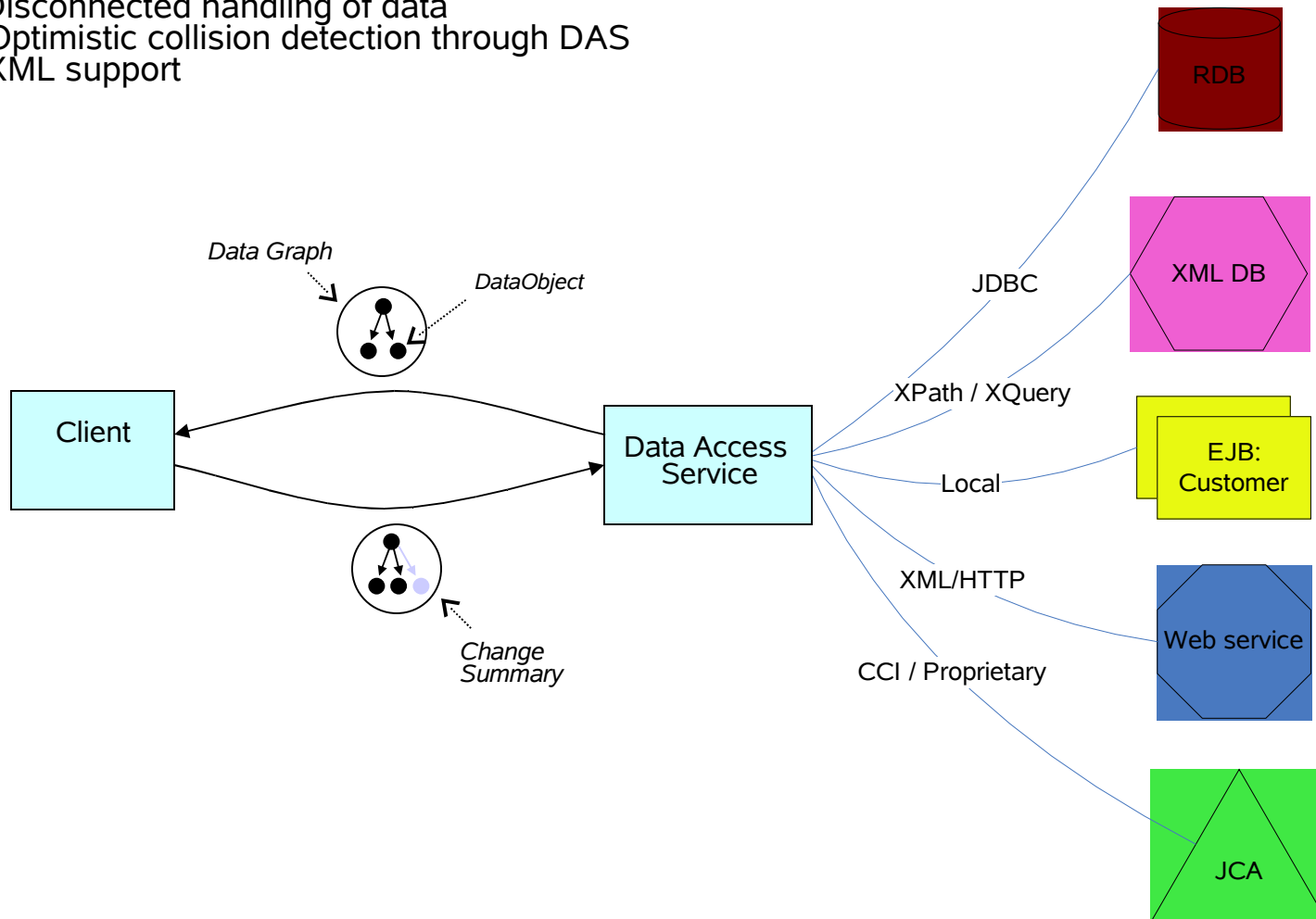
An industry standard developed by 17 companies at www.osoa.org and now going through a formal standardization process at OASIS OpenCSA (www.oasis-opencsa.org)

Bird View: Service Component Architecture

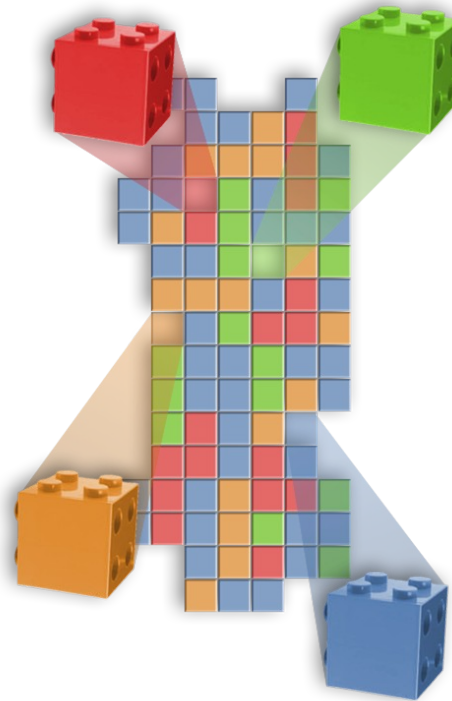


What is SDO?

- Heterogeneous Data Access across the enterprise
- Ability to record change in data
- Disconnected handling of data
- Optimistic collision detection through DAS
- XML support



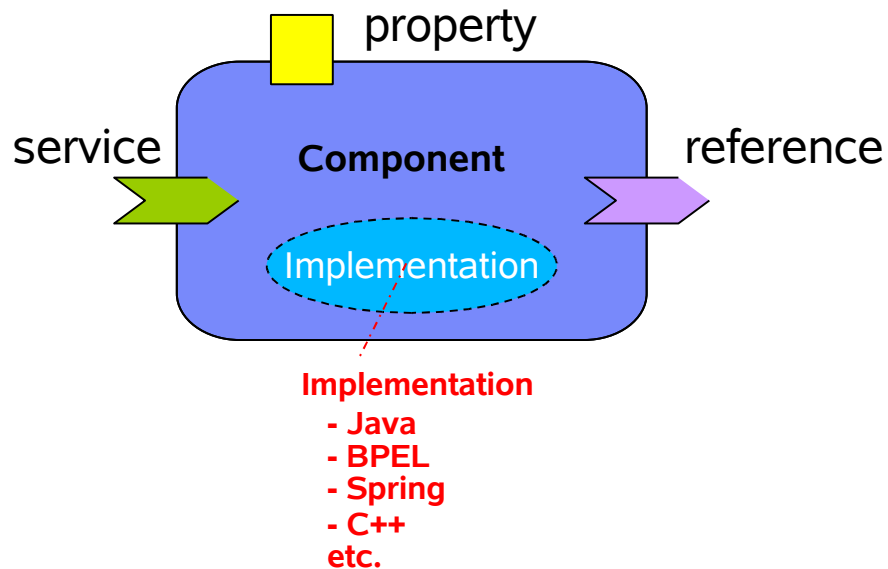
Service Component Architecture Programming Model Overview



What is a Component?

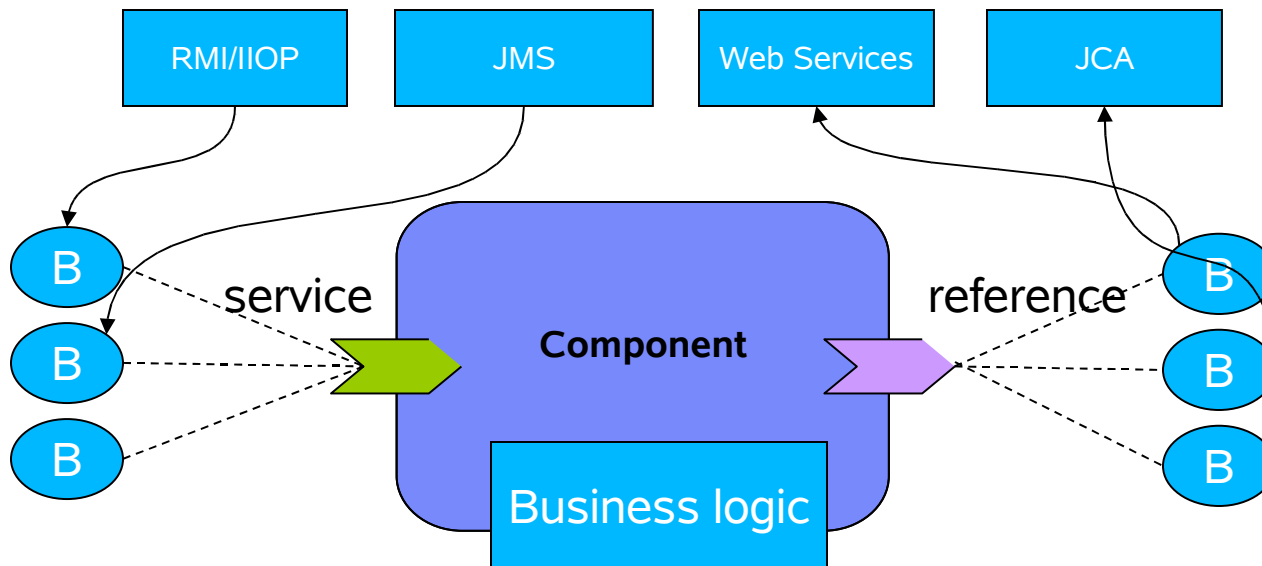
➤ A component is the basic building block that encapsulates business logic.

- What business functions it provides (service) ➡
- What business functions it consumes (reference) ➡
- What settings can be customized (property) ■
- How the business logic is implemented by various technologies (implementation)



Communications with SCA

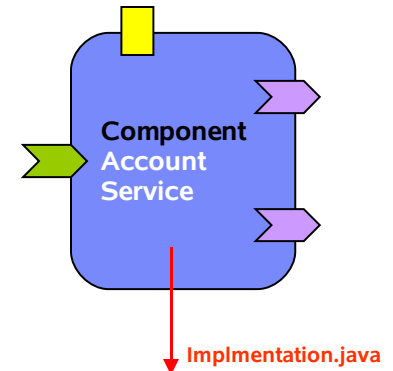
- No more API calls in my application code! With SCA, it's a matter of declaring bindings and switching at your will.
- The business logic is not bound to a pre-defined deployment topology any more.



AccountService: as a java component

@Remotable

```
public interface AccountService {  
    AccountReport getAccountReport(String customerID);  
}  
  
public class AccountServiceImpl implements AccountService {  
    @Reference  
    public void setAccountDataService(AccountDataService value) {  
        accountDataService = value;  
    }  
    @Reference  
    public void setStockQuoteService(StockQuoteService value) {  
        stockQuoteService = value;  
    }  
    @Property  
    public void setCurrency(String value) {  
        currency = value;  
    }  
}
```



AccountService: as a PHP Component

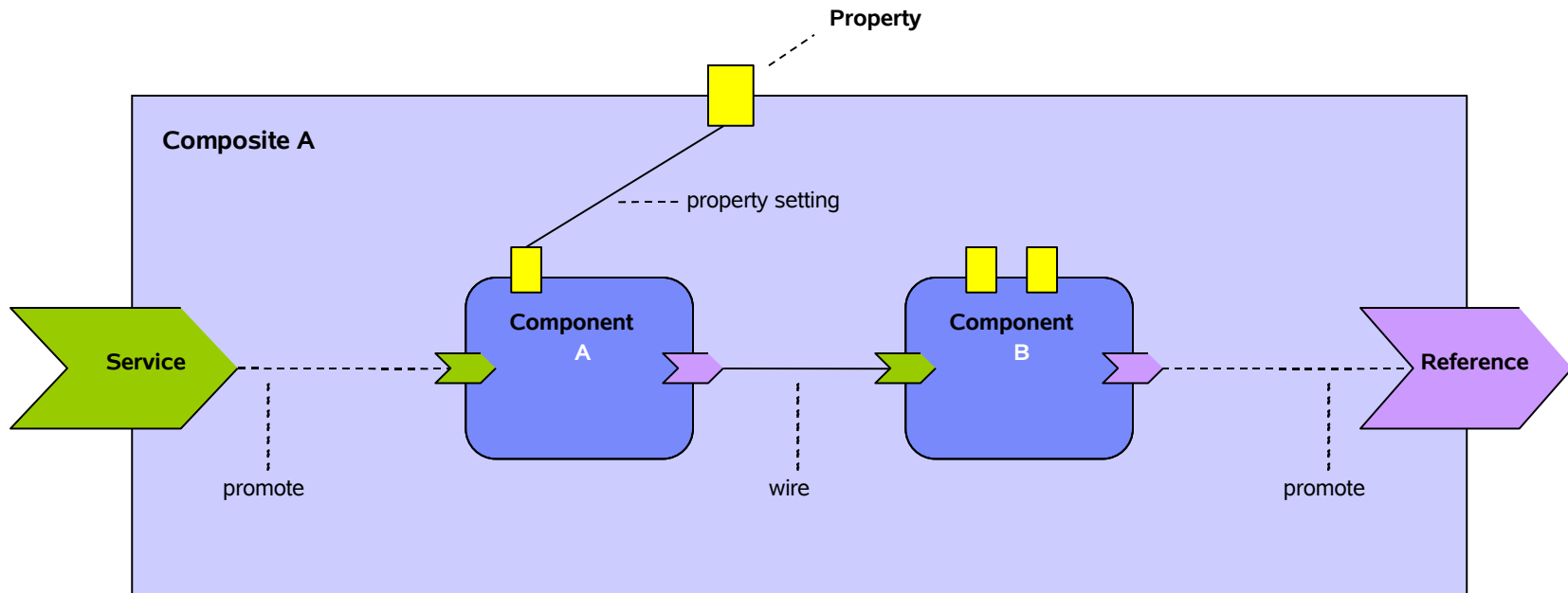
AccountServiceImpl.php

```
/**
 * @service
 * @binding.ws
 * @namespace acc http://www.bigbank.com/AccountService AccountService.xsd
 */
class AccountService {

    /**
     * Returns an account report (a set of account summaries) for a given customer.
     * @param string $customerID
     * @return acc:AccountReport
     */
    function getAccountReport($customerID) {
        ...
    }
}
```

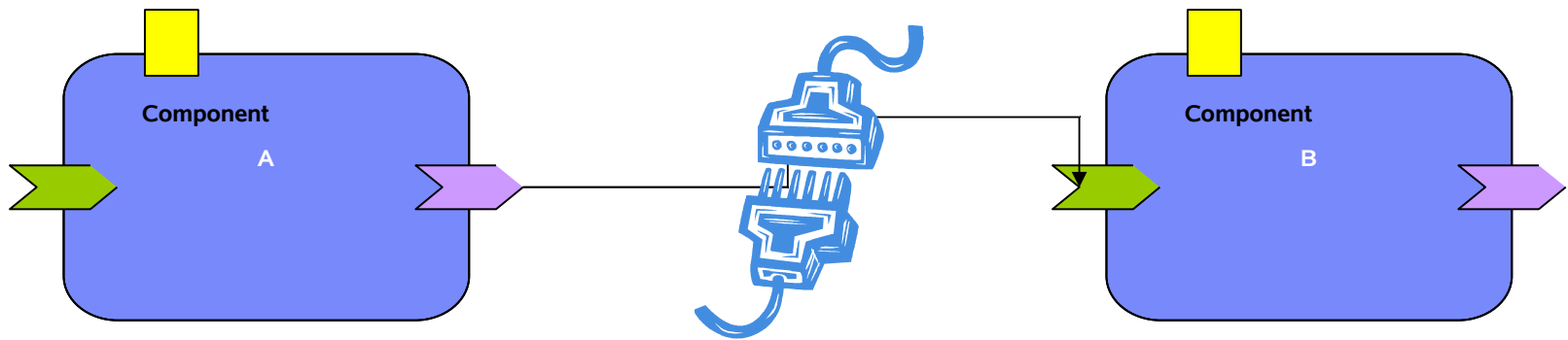
What is a Composite?

- An SCA composite is used to assemble SCA elements in logical groupings. It is the basic unit of composition within an SCA Domain. An SCA composite contains a set of components, services, references and the wires that interconnect them, plus a set of properties which can be used to configure components.



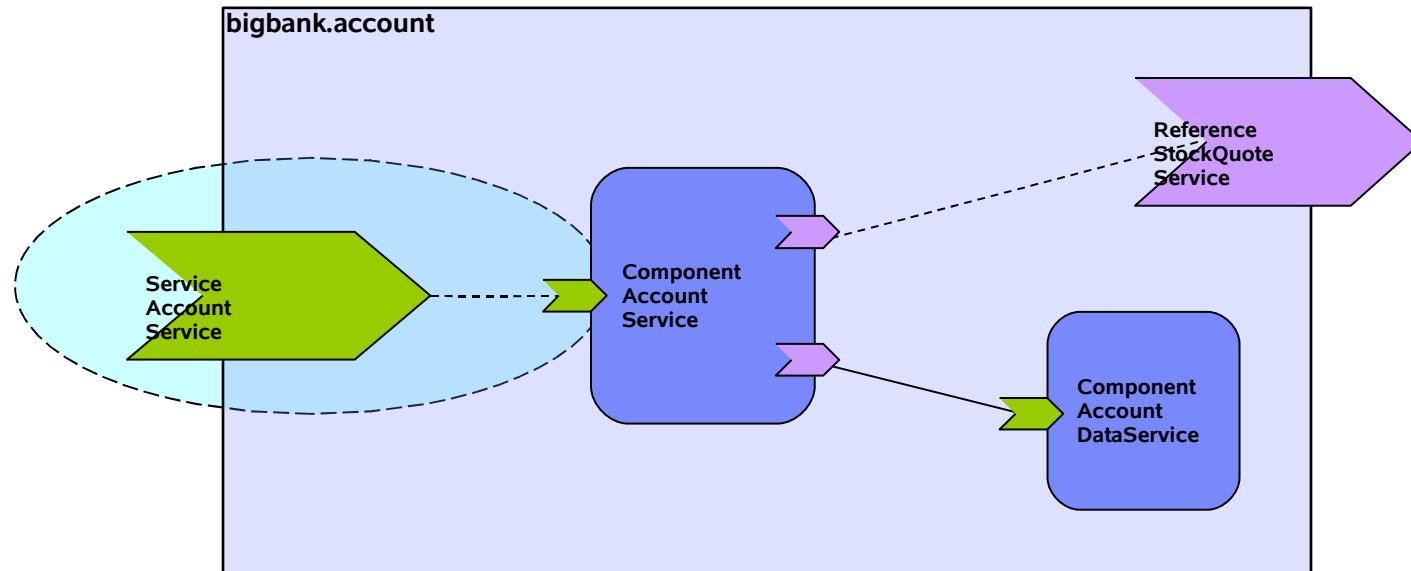
Wiring components together

- **Wiring is about resolving the dependencies. A service consumer is connected to a service provider based on the contracts of the reference and service**
- **The matching is based on the interfaces, bindings and intents/policies**



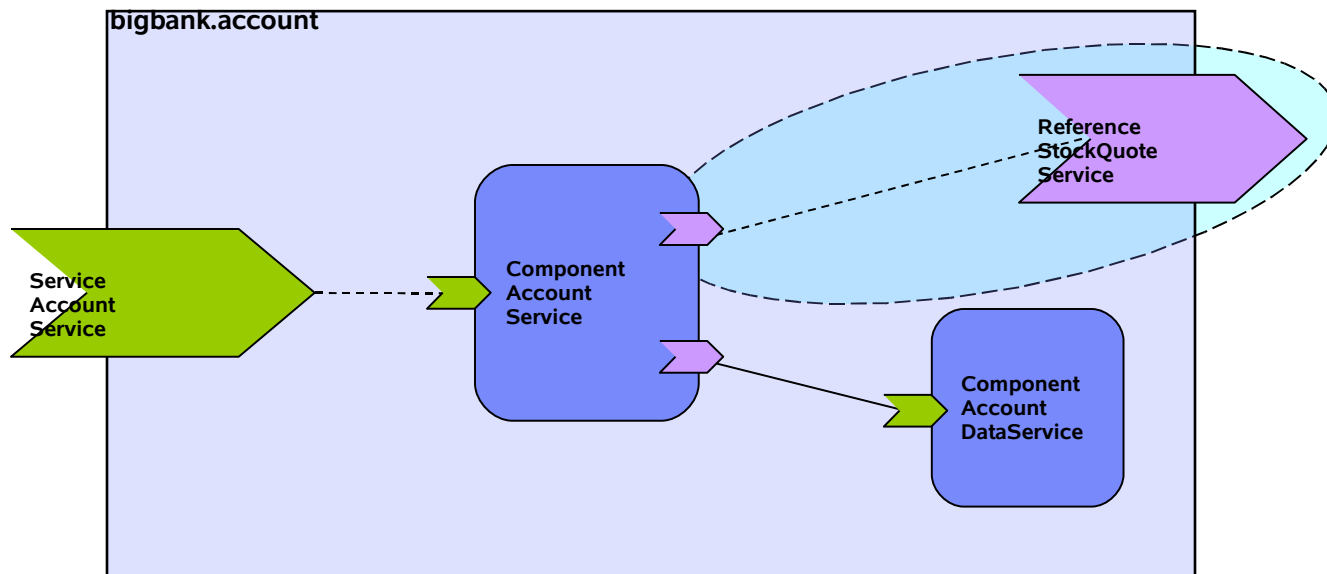
Promoting Services

- Composite services define the public services provided by the composite, which can be accessed from outside the composite.



Promoting References

- Composite references represent dependencies which the composite has on services provided elsewhere, outside the composite.
- The reference can be either connected to an external service or a service provided by the enclosing composite.



Sample composite XML configuration

bigbank.account.composite

```
<composite name="bigbank" xmlns="http://www.oxa.org/xmlns/sca/1.0"
  targetNamespace="http://bigbank.com" ... >

  <service name="AccountService" promote="AccountService">
    <binding.ws requires="soap"/>
  </service>

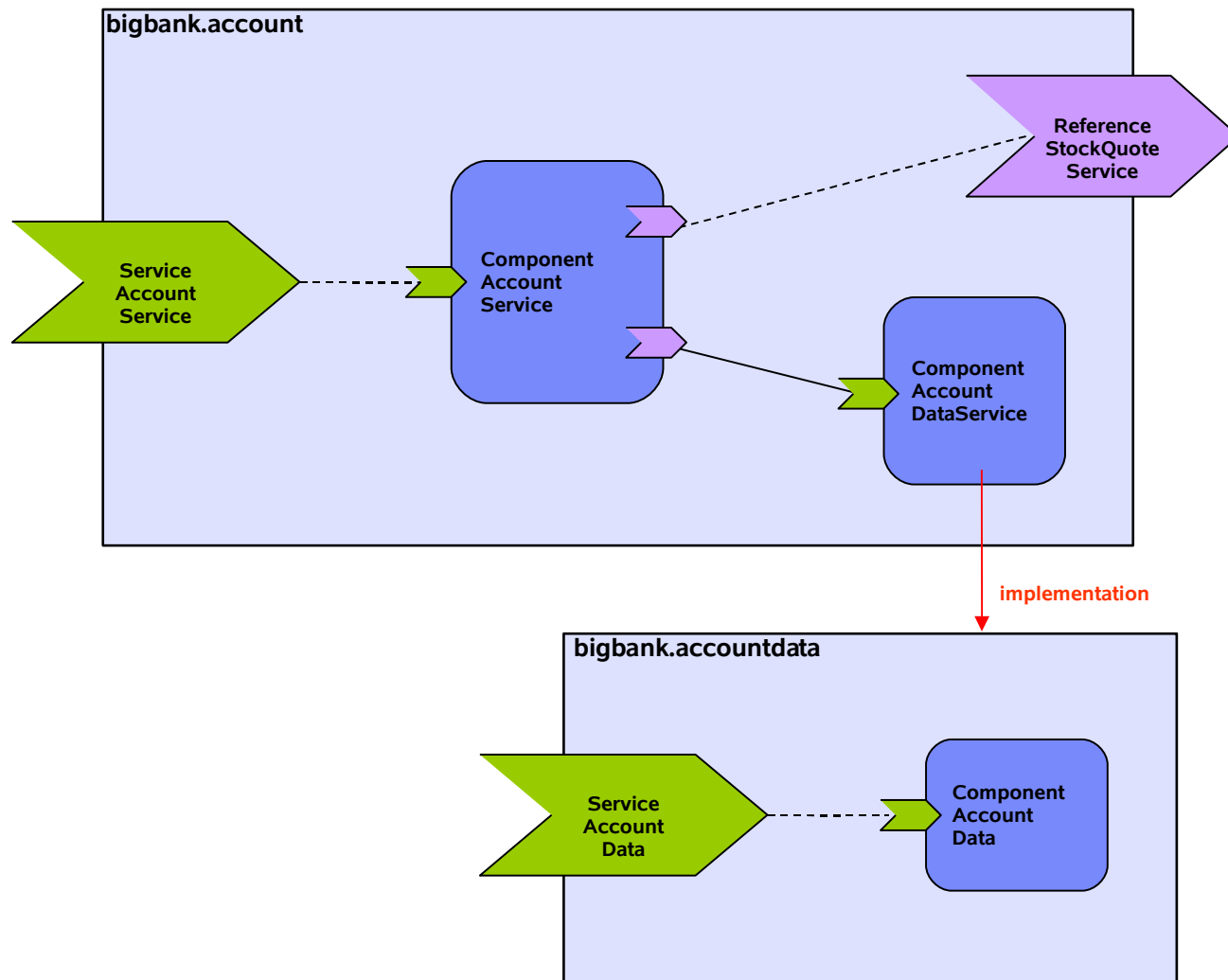
  <reference name="stockquoteService" promote="AccountService/stockQuoteService">
    <binding.ws wsdlElement="http://www.stockquote.com/#
      wsdl.endpoint(StockQuoteService/StockQuoteServiceSOAP)"/>"/>
  </reference>

  <component name="AccountService">
    <implementation.java class="bigbank.account.AccountServiceImpl"/>
    <reference name="accountDataService" target="AccountDataService"/>
    <property name="currency">EURO</property>
  </component>

  <component name="AccountDataService">
    <implementation.java class="bigbank.accountdata.AccountDataServiceImpl"/>
  </component>

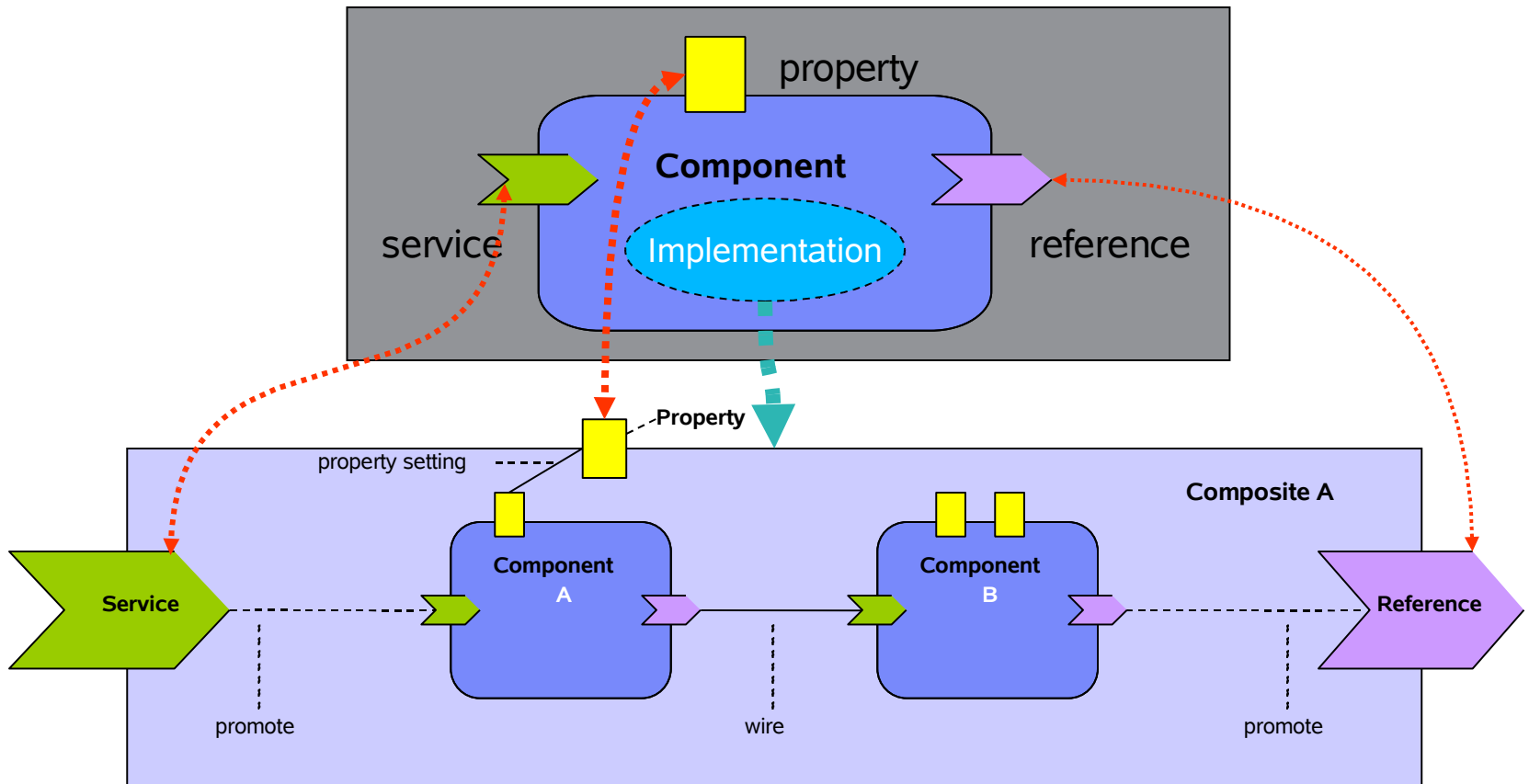
</composite>
```

Assembling Components



Recursive composition

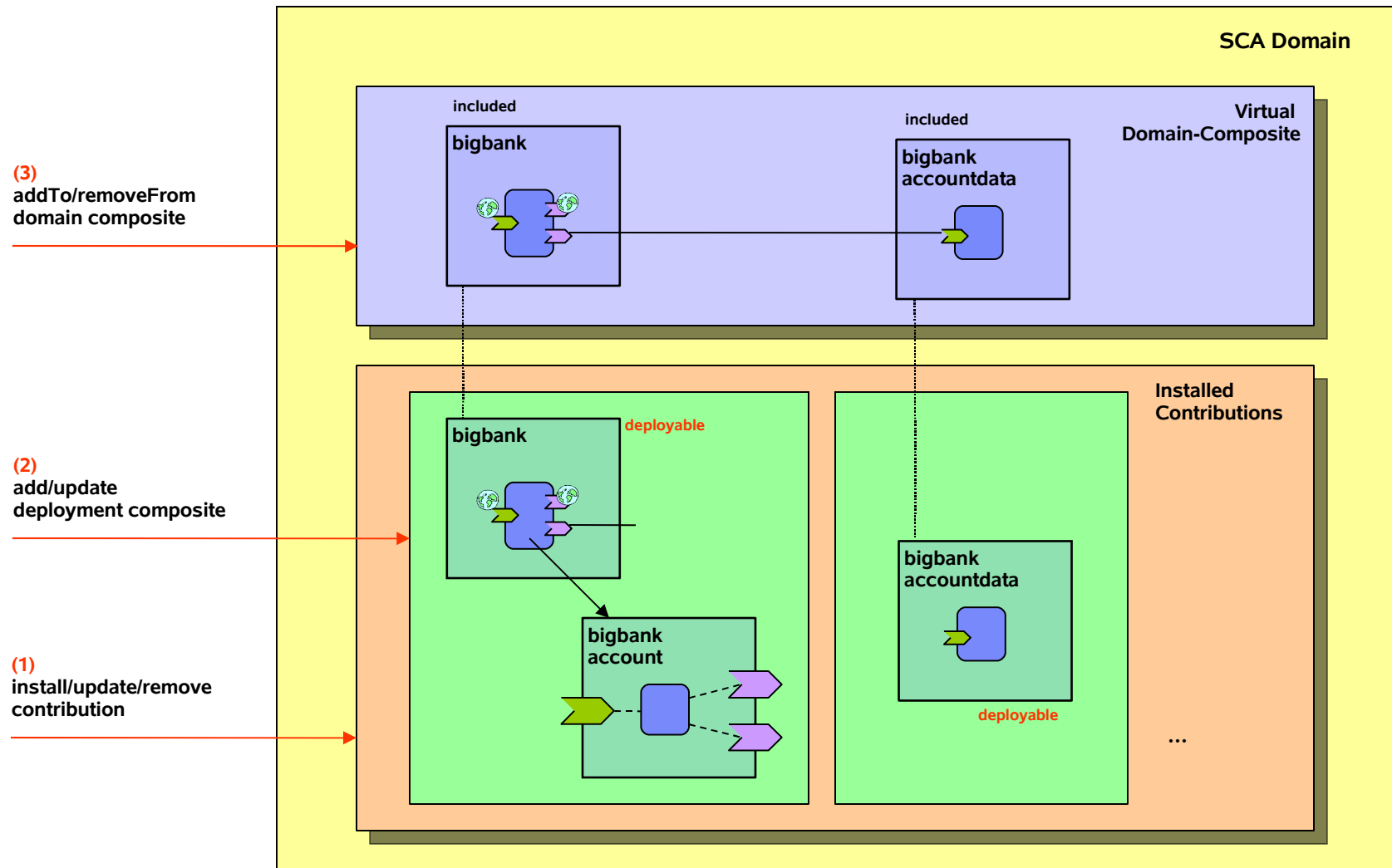
- A composite can be used to implement a component. The partially assembled components as a whole become a bigger building block at higher level.



Other SCA Features

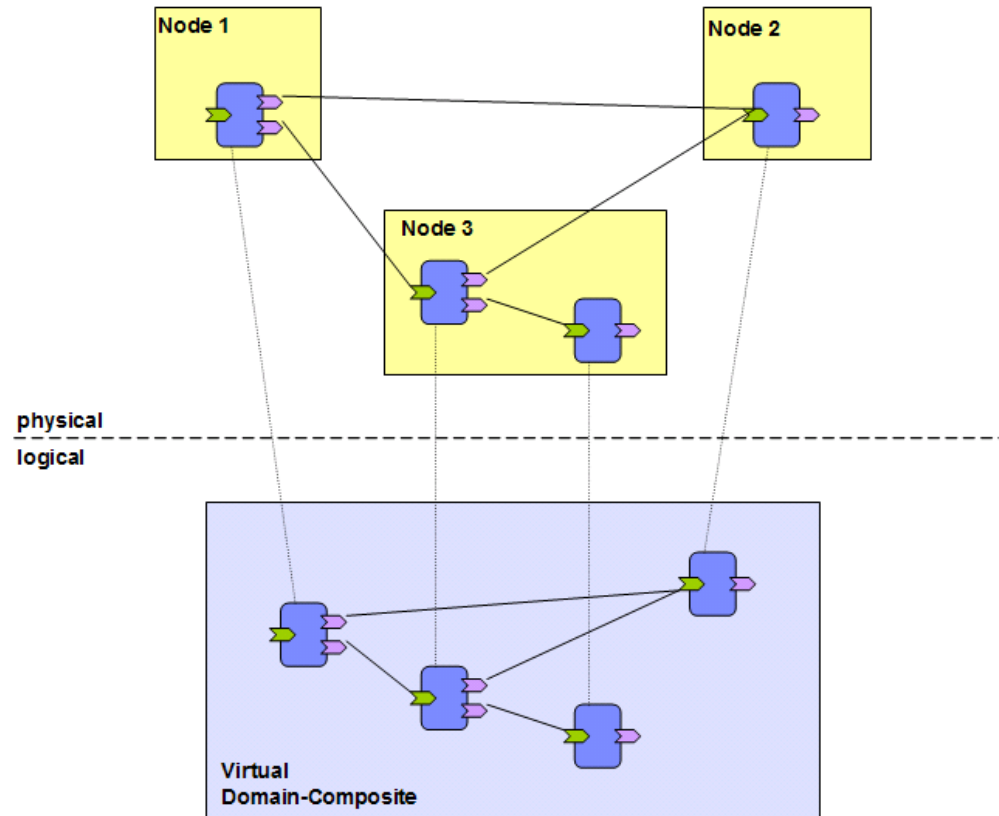
- **Policies – intent, policySet**
- **Asynchronous invocations – oneway, callback**
- **Conversations**
- **Scope – stateless, request, conversation, composite**
- **Packaging and deployment – domains, contributions**

SCA – Deploying Composites



Distributed Deployment

A distributed deployment of the assembly.



Conclusion: Key Benefits of SCA

- **Loose Coupling:** Components integrate with other components without needing to know how other components are implemented
 - Loose coupling - KEY requirement for SOA
- **Flexibility:** Components can easily be replaced by other components
 - Flexibility - KEY requirement for SOA
- **Services** can be *easily* invoked either synchronously or asynchronously
- **Composition** of solutions: clearly described
 - Composition of services - KEY requirement for SOA
- **Productivity:** Easier to integrate components to form composite application

Apache Tuscany

Open Source SOA

Apache Tuscany – A free, Open Source SOA Runtime

➤ **Technology: Provides an infrastructure that supports the simple programming model for developing SOA solutions.**

- It provides an implementation for SCA specification, currently at spec 1.0 level
- It provides an implementation for SDO specification, currently at spec 2.1 level
- It provides an implementation for RDB Data Access Service. More in progress.
- It aims at creating a 'real' solution for SOA challenges through user feedback to enhance Tuscany and to provide feedback to the specification for further improvements.
- Provides support for SCA/SDO/DAS in Java and C++. PHP implementation is in PHP PECL open source library.
- Modular architecture can address the need of various end users: Embedders, system integrators and end users

➤ **Community: Establishes mindshare, creates an ecosystem**

- Establish SCA and SDO mind share and provide a free, open source solution.
- Ecosystem – a 'real' enabler for delivering SOA solutions.
- Integrates well with other open source offerings
- Participants/users: Industry leaders, System Integrators, ISVs, researchers, interested individuals and other open source projects.
- About 500 subscribers and it is growing
- Open to anyone who is interested in helping to develop a simple and real solution for SOA

Apache Tuscany - Technology

➤ **Apache Tuscany infrastructure can easily be extended to support:**

- Integration stack - For example, support for BPEL, Mediations, ...
- Scripting languages - For example support for PHP, JavaScript, Ruby, ...
- Other programming models – For example Spring, ...
- Container technologies – For example OSGI, Tomcat, Geronimo, Jetty,...
- Binding types - For example Web Services, JMS, JCA, JSON, ATOM...
- Data Bindings – For example Axiom, JAXB, SDO, ...
- Data Sources - For example RDB, XML,

** SCA and SDO are implemented in Java and C++. For list of supported container technologies, binding types, Please see the website.

<http://incubator.apache.org/Tuscany>

➤ **Apache Tuscany is lightweight and easy to use**

➤ **SOA tooling for SCA is in progress at <http://www.eclipse.org/stp/>**

Apache Tuscany - Community

- **Project started in Apache incubator in December 2005.**
- **A real community is forming: Users answering user questions and providing patches.**
- **Detailed user discussions on the mailing lists: Indication of real customer implementations based on technologies offered in Tuscany.**
- **Committers: Individuals, Roguewave, Redhat, IONA, IBM**
- **Contributors: Individuals, ISVs, System Integrators, Vendors and growing.**
- **User and developer mailing list subscription steadily growing, at 500 combined now.**
- **Integration with other Apache projects in progress: ODE, ServiceMix, Synapse, Geronimo....**

Tuscany SCA Java – Key Features in 0.90 (06/07)

➤ Core runtime

- SCA 1.0 assembly model, contribution service, and policy enablement
- Synchronous and asynchronous invocations with callbacks

➤ Extension modules

- Bindings for Web Services / Axis2, RMI and JSON-RPC
- Containers for Java, JavaScript, Ruby, Python, Groovy components
- Data bindings for Axiom, SDO and JAXB
- WSDL and Java Interfaces

➤ Host platforms

- Standalone (client)
- Embedded Tomcat and Jetty
- J2EE Web Application

➤ Tools

- Java2WSDL and WSDL2Java Maven plug-ins and command-line tools

➤ Samples

- Showing application development and extension development

Tuscany SCA Java – Key Features in 0.91 (07/07)

➤ **Core runtime**

- Improved coverage of SCA 1.0 assembly and API spec
- Dynamic interface support (for scripting components)

➤ **Extension modules**

- New Bindings for Atom and RSS Feeds, EJB Session Beans, AJAX / DWR
- Static Web Resource component implementation type
- Spring component implementation type

➤ **Host platforms**

- Improved dynamic extension module loading

➤ **Samples**

- Feed Aggregator
- BigBank and Alert Aggregator demos

Tuscany SCA Java – Work in progress...

➤ Core runtime

- Complete support for SCA 1.0 assembly and API spec
 - Conversational
 - Service References and Callbacks
 - SCA contribution imports/exports
- Deployment to a distributed SCA Domain
- SCA Domain Administration

➤ Extension modules

- BPEL component implementations – work in progress with Apache ODE
- Notification components? – prototype in trunk
- OSGI component implementations? prototype in trunk
- JMS Binding – work in progress in trunk

➤ Host platforms

- Geronimo integration – work in progress in Geronimo sandbox
- Starting to work with other Apache projects, ServiceMix, Synapse

➤ Samples

- Tutorials – how to develop, build, deploy
- Distributed SCA domain

Tuscany SDO Java – Features in 1.0–beta1 (05/07)

- **Implementation of the SDO 2.1 spec**
- **Dynamic DataObject support**
- **Static code generation**
(generated code doesn't have any EMF dependencies)
- **Helpers: XMLHelper, XSDHelper, DataFactory, TypeEqualityHelper, CopyHelper**
- **Type scoping**
- **Change Summary support on DataObjects and DataGraphs**
- **Samples showing application development with SDO**

Tuscany SDO Java – Work in progress

➤ **Working on 1.0 release... RC3 review and vote started today**

- Full coverage of SDO 2.1 spec

➤ **SDO Community Test Suite**

- Independent of Tuscany SDO implementation
- Can be used to exercise any SDO 2.1 implementation

Tuscany Native/C++ – Features in 1.0-M3 (05/07)

➤ Core runtime

- SCA Assembly spec 0.96
- SCA C++ API 0.95
- SDO C++ 2.01

➤ Extension modules

- Support for C++, Ruby, Python, PHP component implementations
- Bindings for Web Services / Axis2C and REST / Curl
- SDO / Axiom Data binding

➤ Host Platforms

- Apache HTTPD modules
- Apache AXIS2C
- WinXP, Linux, Mac OS X

➤ Samples

- Showing application development and usage of bindings

Tuscany Native/C++ – Work in progress...

➤ Core runtime

- SCA assembly spec 1.0
- SCA C++ API 1.0
- SDO C++ 2.01
- New build system

➤ Extension modules

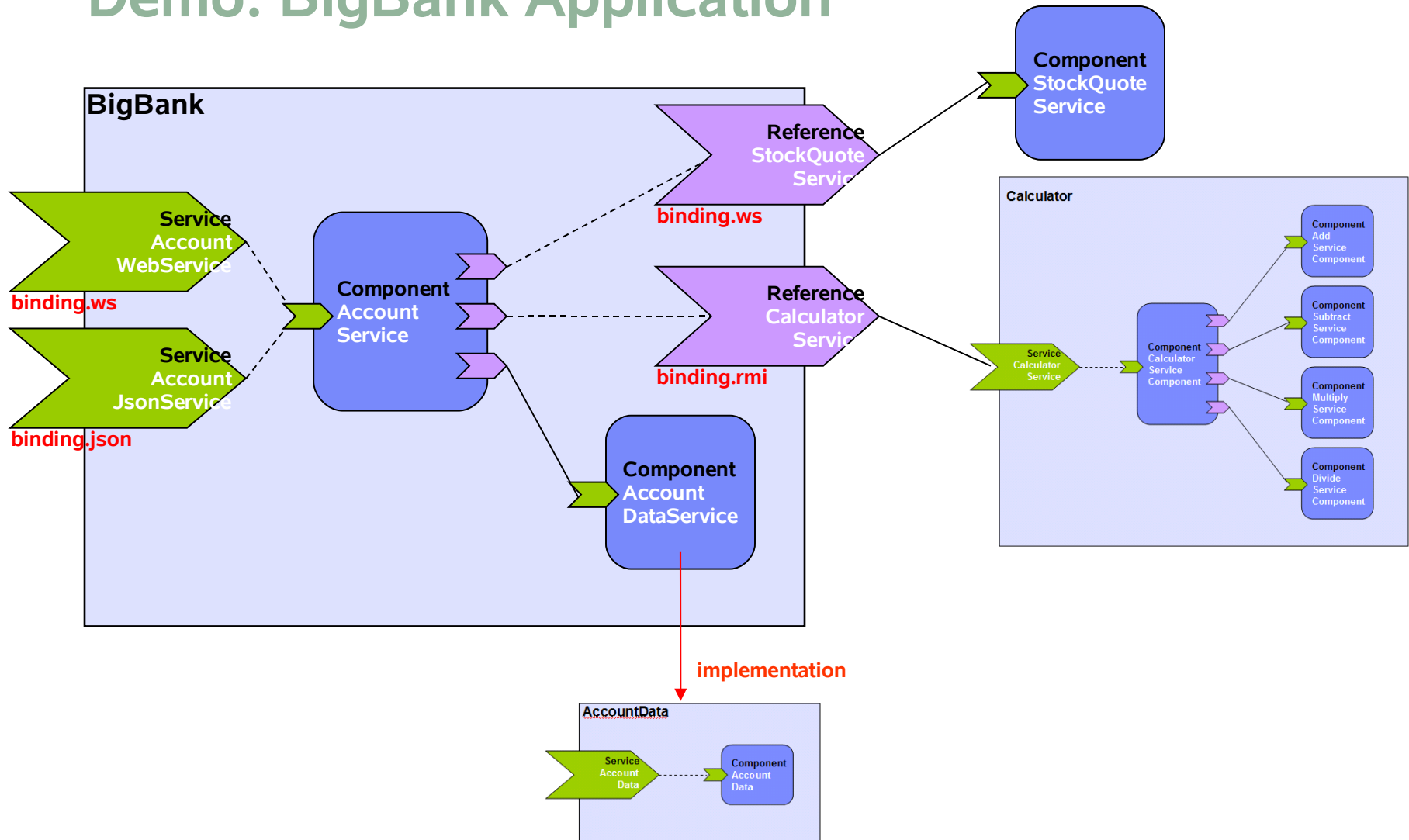
- PHP work in progress
- Upgrade to latest versions of dependencies...

➤ Samples

- More samples and Demos
- Samples showing integration in a distributed SCA domain

Demo

Demo: BigBank Application



<http://svn.apache.org/repos/asf/incubator/tuscany/java/sca/demos/>

Open Source SOA: SCA, SDO and DAS
<http://incubator.apache.org/tuscany>



Questions?

➤ **Apache Tuscany Web site and downloads**

<http://incubator.apache.org/tuscany/>

➤ **Apache Tuscany source code repository**

<http://svn.apache.org/repos/asf/incubator/tuscany/>

➤ **Apache Tuscany mailing lists**

tuscany-dev@ws.apache.org - tuscany-dev-subscribe@ws.apache.org

tuscany-user@ws.apache.org - tuscany-user-subscribe@ws.apache.org

➤ **Apache Tuscany IRC**

[#tuscany on irc.freenode.net](http://irc.freenode.net)

➤ **PHP Extension Community Library (PECL)**

<http://pecl.php.net>

➤ **Open SOA Collaboration (SCA and SDO specs and articles)**

<http://www.osoa.org>

Thank You!