Open Source SOA with Service Component Architecture and Apache Tuscany
Learn how to build and deploy Composite Service Applications using Service Component Architecture (SCA) and Apache Tuscany

Goal
Agenda

- SCA in a Nutshell
- Apache Tuscany Overview
- Demo – Business Value Scenarios
- SCA Quick Tour
- Tuscany SCA Implementation
- New and Notable
- Tuscany Community
- Summary
SCA in a Nutshell

➢ A development and deployment model for SOA

➢ Service-based models for the
  • Construction
  • Assembly
  • Deployment

  of composite service applications

➢ In a distributed and heterogeneous environment of
  • Multiple languages
  • Multiple container technologies
  • Multiple service access methods
SCA Specifications - History

- Dec 2003: BEA and IBM start collaborating on SCA
- Nov 2005: 0.9 specs published
  - BEA, IBM, Oracle, SAP, IONA, and Sybase
- July 2006: 0.95 specs and OSOA.org (Open SOA)
  - Added: Cape Clear, Interface21, Primeton Technologies, Progress Software, Red Hat, Rogue Wave, Siemens AG, Software AG, Sun, TIBCO
- Mar 2007: 1.0 specs published
  - Submitted to OASIS
- April 2007: OASIS Forms Open CSA Member Section
- Sept 2007: Formal standardization starts in OASIS Open CSA
Open SOA - http://www.osoa.org

The OSOA Collaboration

Supporters of the OSOA Collaboration

Open Source SOA: SCA, SDO & DAS
http://incubator.apache.org/Tuscany
OASIS Open CSA - http://www.oasis-opencsa.org/

The OASIS Open Composite Services Architecture (CSA) Member Section advances open standards that simplify SOA application development. Open CSA brings together vendors and users from around the world to collaborate on the further development and adoption of the Service Component Architecture (SCA) and Service Data Objects (SDO) families of specifications.

**Steering Committee**
23 March 2007 - 1:07pm — jeff.mischkinsky
Open CSA activities are managed by a Steering Open CSA membership in an open process. The

- Graham Barber - IBM
- David Burke - TIBCO
- Patrick Leonard - Rogue Wave Software
- Mark Little - Red Hat
- Jeff Mischkinsky - Oracle
- Sanjay Patel - SAP
- Michael Rowley - BEA Systems

**Committees**
Several technical committees are affiliated with Open CSA:

OASIS Service Component Architecture / Assembly (SCA-Assembly) TC
Defining core SCA composition model to simplify SOA application development

OASIS Service Component Architecture / Policy (SCA-Policy) TC
Defining an SCA policy framework to simplify SOA application development

OASIS Service Component Architecture / Bindings (SCA-Bindings) TC
Standardizing bindings for SCA services and references to communication protocols, technologies and frameworks

OASIS Service Component Architecture / BPEL (SCA-BPEL) TC
Specifying how SCA component implementations for SOA can be written using BPEL

OASIS Service Component Architecture / C and C++ (SCA-C-C++) TC
Standardizing C and C++ use within an SCA domain for SOA

OASIS Service Component Architecture / J (SCA-J) TC
Standardizing Java (tm) use within an SCA domain for SOA
Agenda

- SCA in a Nutshell
- Apache Tuscany Overview
- Demo – Business Value Scenarios
- SCA Quick Tour
- Tuscany SCA Implementation
- New and Notable
- Tuscany Community
- Summary
Open Source SCA – Apache Tuscany

Welcome to the Apache Tuscany project! The Tuscany community is working to create a robust and easy to use infrastructure that simplifies the development of service-based application networks and addresses real business problems posed in SOA.

Tuscany is based on specifications defined by the Open CSA Collaboration:

- **Service Component Architecture (SCA)** - An essential characteristic of SOA is the ability to assemble new and existing services to create brand new applications that may consist of different technologies. Service Component Architecture defines a simple, service-based model for construction, assembly and deployment of network of services (existing and new ones) that is language neutral. Tuscany is working on SCA specification 1.0. [Learn more about SCA](http://incubator.apache.org/tuscany).

- **Service Data Object (SDO)** provides a uniform interface for handling different forms of data, including XML documents, that can exist in a network of services and provides the mechanism for tracking changes. Tuscany supports SDO specification 2.1. [Learn more about SDO](http://incubator.apache.org/tuscany).

- **Data Access Service (DAS)** provides a simple SDO interface to relational databases. [Learn more about DAS](http://incubator.apache.org/tuscany).

The above mentioned technologies provide a full infrastructure for developing and running SOA based applications. They are not dependent on one another and can be used independently.

Please join us to create a simple, practical, extensible SOA infrastructure to address the problems that large-scale applications and service networks are faced with. We look forward to your participation.
What is Apache Tuscany

Some words from the project charter:

“... open-source software for distribution at no charge to the public, that simplifies the development, deployment and management of distributed applications built as compositions of service components.

These components may be implemented with a range of technologies and connected using a variety of communication protocols.

This software will implement relevant open standards including, but not limited to, the SCA standard defined by the OASIS OpenCSA member section."
Apache Tuscany - History

- Project created in Dec 2005 in Apache incubator
- Major Releases
  - 09/07 1.0 release, first implementation of SCA spec v1.0
  - 02/08 1.1 release, bug fixes, JMS binding, improved policy support
  - 04/08 Working on 1.2 release
    - Distributed SCA Domain management
    - Support for JAX-WS annotations
    - New ATOM binding using Apache Abdera
    - Improved JMS binding
- Incremental releases every 6/8 weeks
- Worldwide growing user and developer community!
- Users in production with Tuscany
SCA and Apache Tuscany in Action
Business Value Scenarios

Demo
Demo - The Rise of a Fruit Business

- The Fruit Store
- Merger or Acquisition - The Fruit & Vegetable Store
- The Fruit & Vegetable Store as Supplier
- The Fruit & Vegetable Store Solution Provider
- The Fruit Store Widget - Mashup

All the code available at http://svn.apache.org/repos/asf/incubator/tuscany/java/sca/tutorial
The Fruit Store

Creating an Online Business

```
store
    http
    Store
        Collection
        atom
        ShoppingCart
            currencyCode=USD
        Catalog
            Currency
                Converter
                Total
                jsonrpc
                jsonrpc
```
The Fruit&Vegetable Store

Merger or Acquisition
The Fruit & Vegetable Store

Using a Database in the ShoppingCart implementation

- Store
- Catalog
- ShoppingCart
- Currency Converter
- Fruit Catalog
- Vegetable Catalog

Database: USDB

Currency Code: USD

Collection: atom

http

jsonrpc

Total

Database

jsonrpc

Currency Code
The Fruit & Vegetable Store as Supplier

Being a Supplier for other Online Stores

store-supplier

- Store
  - Catalog
    - ShoppingCart
      - currencyCode=USD
      - database=USDB
  - Currency Converter
  - Fruit Catalog
    - Vegetable Catalog

AllAroundFoods.com
The Fruit & Vegetable Store Solution Provider

Providing a Store Solution in another Geography

store-eu

http

Collection

atom

ShoppingCart

database=EUDB

Fruit Catalog

wo

currencyCode=EUR

Vegetable Catalog

wo

EU Services

Currency Converter

Fruit Catalog

VE Services
The Fruit Store Widget - Mashup

- Store Mashup – Offering the Store as an Open Ajax Widget

[Diagram of the fruit store widget and mashup process]
The Fruit Store Widget - Mashup

Store Mashup – Offering the Store as an Open Ajax Widget
Agenda

- SCA in a Nutshell
- Apache Tuscany Overview
- Demo – Business Value Scenarios
- SCA Quick Tour
- Tuscany SCA Implementation
- New and Notable
- Tuscany Community
- Summary
SCA Quick Tour

➢ Construction
➢ Assembly
➢ Deployment
SCA - Construction

➢ Constructing service component implementations
  • Implementer focuses on business logic
  • Implementer chooses the language that fits the business problem
  • No code is dependent on the means of accessing the service

➢ Defining service dependencies
  • Use business services without knowing how they will be accessed
  • Only the interface is known

➢ Defining other configuration features
  • Properties
  • Policies
SCA - Assembly

Service Interface
- Java
- WSDL

Composite A

Service Binding
- Web Service
- JMS
- JCA
- SLSB
- HTTP
- JSONRPC
- ATOM
- ...

Component A

Property

Implementation
- Java
- BPEL
- SCA Composite
- Spring
- JEE
- Scripting: Groovy, JScript, PHP, Python, Ruby, ...
- XQuery
- ...

Component B

Reference Interface
- Java interface
- WSDL PortType

Reference Binding
- Web Service
- JMS
- JCA
- SLSB
- HTTP
- JSONRPC
- ATOM
- ...

Reference

promote

property setting

wire

promote
SCA - Assembly

➤ Building Solutions from Assets

```
interface AccountService {
  AccountReport getAccountReport(String customerId);
}

class AccountServiceImpl implements AccountService {
  // implementation
}
```

Open Source SOA: SCA, SDO & DAS
http://incubator.apache.org/Tuscany
SCA - Deployment

SCA Domain Composite

storeus

storeeu

storeus.html

ShoppingCartImpl.java

storeeeu.html

Catalog.java

asset contributions

Open Source SOA: SCA, SDO & DAS
http://incubator.apache.org/Tuscany
SCA – Deployment Services

- Logical configuration of the cloud
- Provisioning
- Monitoring

1. addContribution
2. addToDomainComposite
Agenda

- SCA in a Nutshell
- Apache Tuscany Overview
- Demo – Business Value Scenarios
- SCA Quick Tour
- Tuscany SCA Implementation
- New and Notable
- Tuscany Community
- Summary
Tuscany provides implementations of the main SCA specifications including:

- SCA Assembly Model V1.0
- SCA Policy Framework V1.0
- SCA Java Common Annotations and APIs V1.0
- SCA Java Component Implementation V1.0
- SCA Spring Component Implementation V1.0
- SCA BPEL Client and Implementation V1.0
- SCA Web Service Binding V1.0
- SCA JMS Binding V1.0
- SCA EJB Session Bean Binding V1.0
Apache Tuscany – SCA Extensions

➤ SCA bindings
  • RSS and ATOM Feeds, HTTP resources, JSON-RPC, Direct Web Remoting, Pub/Sub Notifications and RMI

➤ SCA implementation types
  • OSGi, XQuery, BPEL, and various dynamic languages including Groovy, JavaScript, Python and Ruby

➤ Databindings
  • JAXB, Service Data Objects (SDO), Axis2's AXIOM, JSON, XMLBeans, SAXON, DOM, SAX and StAX

➤ Policies
  • Logging, connection pooling
Apache Tuscany – Distribution and Hosting

- Tuscany can be configured as a single node SCA domain or as an SCA domain distributed across multiple nodes.

- Simple Administration model: SCA domain and SCA node configuration are Web resources accessed using AtomPub.

- Host deployment options:
  - Standalone
  - Distributed nodes across multiple JVMs
  - Embedded Jetty or Tomcat
  - Standard Web application
    - Tested in Tomcat 5.5.20, 6.0.14, Jetty 6.1.3, Geronimo 2.0.2 Tomcat6 JEE5
    - WebSphere Application Server 6.1, other JEE Application Servers
Agenda

- SCA in a Nutshell
- Apache Tuscany Overview
- Demo – Business Value Scenarios
- SCA Quick Tour
- Tuscany SCA Implementation
- New and Notable
- Tuscany Community
- Summary
Apache Tuscany – New and Notable

➢ Community Update

➢ Latest Release Update

➢ Work in Progress

➢ Next Release
Agenda

- SCA in a Nutshell
- Apache Tuscany Overview
- Demo – Business Value Scenarios
- SCA Quick Tour
- Tuscany SCA Implementation
- New and Notable
- Tuscany Community
- Summary
Tuscany Community – Working in Open Source

What we have learned and are practicing
- Code
  - Small Composable Modules, low barrier to entry
  - A working build, commit every day
- Welcoming environment
  - Open mailing list discussions, share ideas, respect new ideas and feedback
  - Encourage user feedback, be responsive to questions, apply patches quickly
  - Help contributors find areas to contribute to

Release early, release often!

Apache Tuscany community is growing
- Mailing list subscribers
- More and more users contributing patches
For More Information – Apache Tuscany

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

- Tuscany SCA Java
  http://incubator.apache.org/tuscany/sca-java.html
- Getting started
- Getting started using Eclipse
- Tuscany Online Store Tutorial
  http://svn.apache.org/repos/asf/incubator/tuscany/java/sca/tutorial
- Using Tuscany with WebSphere Application Server 6.1
- How to get involved
  http://incubator.apache.org/tuscany/getting-involved.html

• Good introduction to SCA
  http://www.davidchappell.com/articles/Introducing_SCA.pdf

• OASIS Open CSA
  http://www.oasis-opencsa.org/
  • V1 level specs
    http://www.oasis-opencsa.org/sca
  • Open CSA Technical Committees
    http://www.oasis-opencsa.org/committees

• OSOA
  http://osoa.org/display/Main/Home
  • V1 level of specs can also be found here
    http://osoa.org/display/Main/Service+Component+Architecture+Specifications
  • More information on that site
    http://osoa.org/display/Main/SCA+Resources
Agenda

- SCA in a Nutshell
- Apache Tuscany Overview
- Demo – Business Value Scenarios
- SCA Quick Tour
- Tuscany SCA Implementation
- New and Notable
- Tuscany Community
- Summary
Summary

➢ Service Component Architecture (SCA)
  - Construction, Assembly, Deployment of Composite Applications
  - Simple end to end composition
  - You can focus on your business logic

➢ Apache Tuscany
  - Implements SCA V1.0
  - Integrates with the Apache Platform
  - Goes beyond the specifications (Web 2.0, ATOM, Scripting, Admin)
  - Many releases
  - Simple to use

Join Tuscany! We welcome any type of contribution.
http://incubator.apache.org/tuscany/getting-involved.html
Thank You