Building composite applications with SCA and Apache Tuscany

Luciano Resende

lresende@apache.org

http://lresende.blogspot.com

ApacheCon

**Leading the Wave
of Open Source**

Today's quote

- Anyone who's interested in the future of application development should also be interested in SCA.

  – David Chappell, Introducing SCA davidchappell.com/articles/ Introducing_SCA.pdf

What's Apache Tuscany

- Some words from 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."
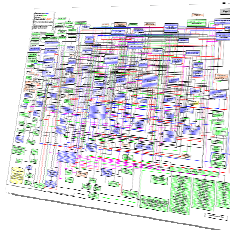
Leading the Wave of Open Source

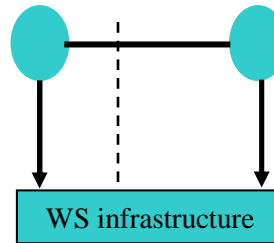# Why SCA ?

Enterprise Application Development Evolution

More Flexibility
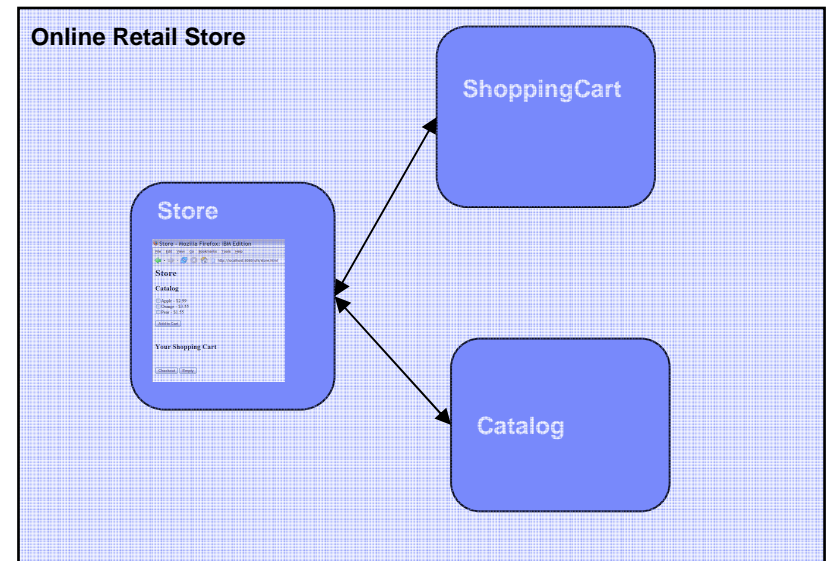
Greater Agility

Higher ROI

WS infrastructure

Service Component Architecture (SCA)

✓ Business logic separate from Service call API, handled by infrastructure

✓Business logic separate from protocol handling, handled by infrastructure

✓Multiple protocols supported. Protocol is a choice

✓Multiple technologies can be used in one solution

✓ Manage as part of the enterprise solution

Standards Based, JAX-WS

✓Business logic separate from Service call API, handled by infrastructure

X Business logic mixed with protocol handling

X Single protocol enforced, for example HTTP

X Single technology stack enforced for a solution

X Cannot manage as part of the enterprise solution

Home Grown, Proprietary

X Business logic mixed with Service call API and protocol handling

X Single protocol enforced

X Cannot manage as part of the enterprise solution
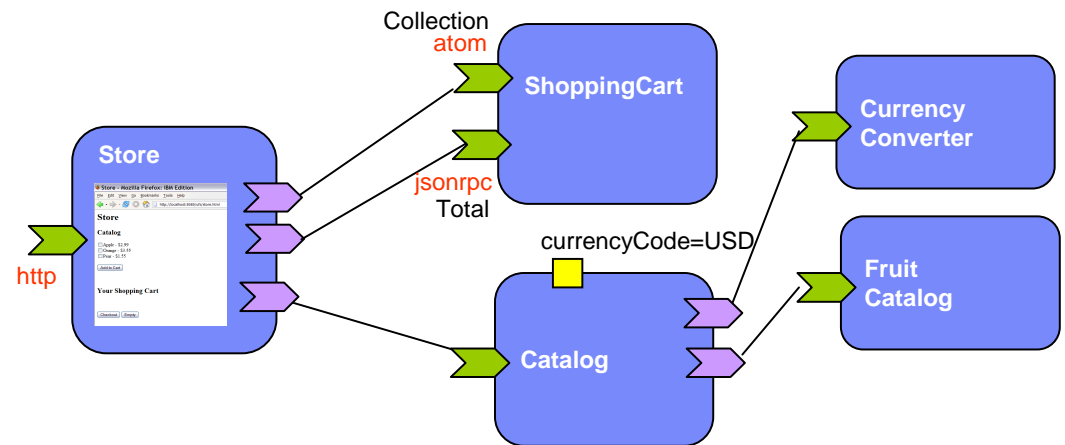
Brittle application

Time

The Store Scenario

- An online retail store with three main business functionality
  - Product Catalog
  - Customer Shopping Cart
  - Front End web store application
- Challenges
  - Business logic mixed with protocol handling
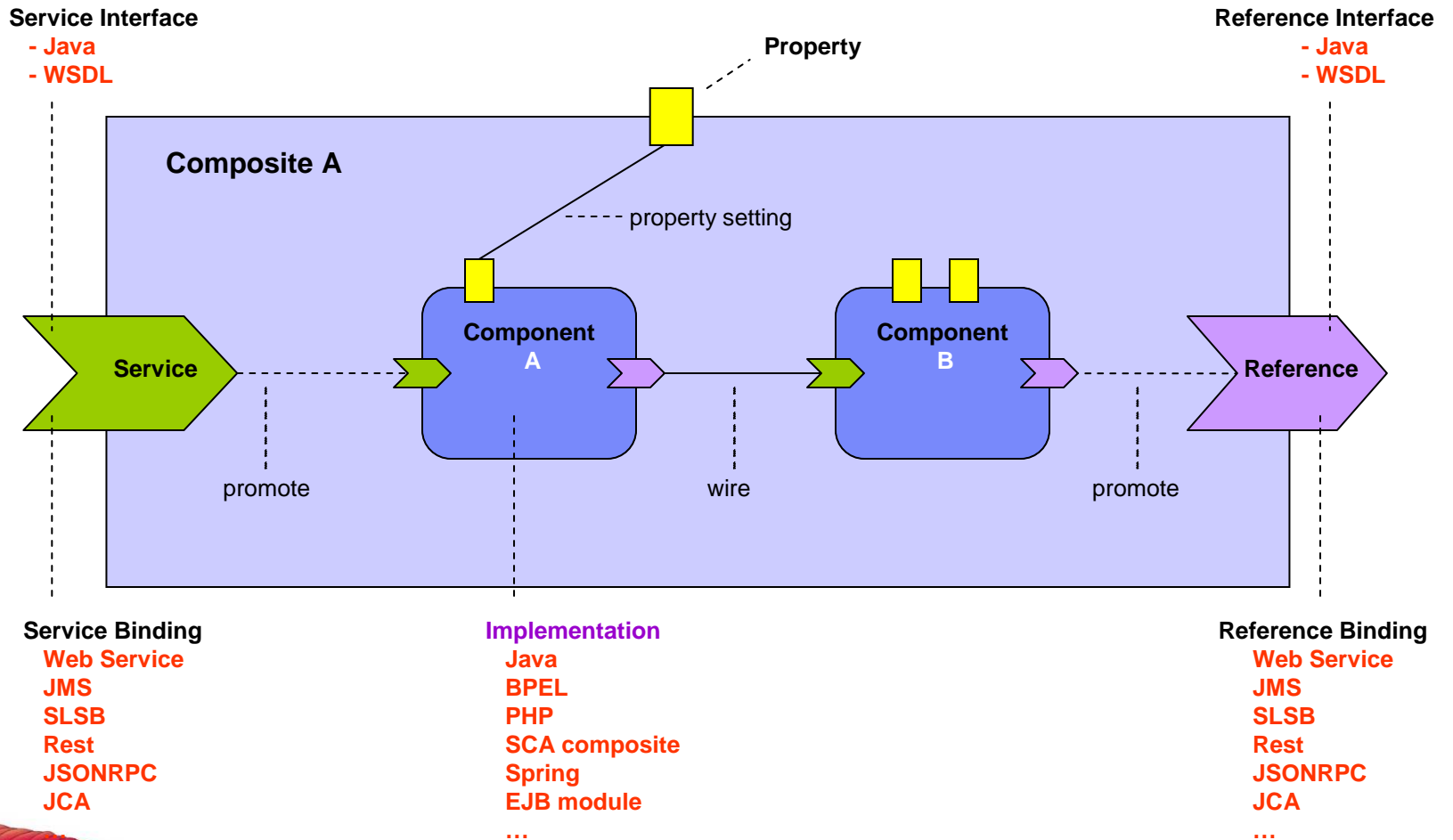 - Single protocol enforced

The Store Scenario using SCA

- An online retail store with three main business functionality
  - Product Catalog
  - Customer Shopping Cart
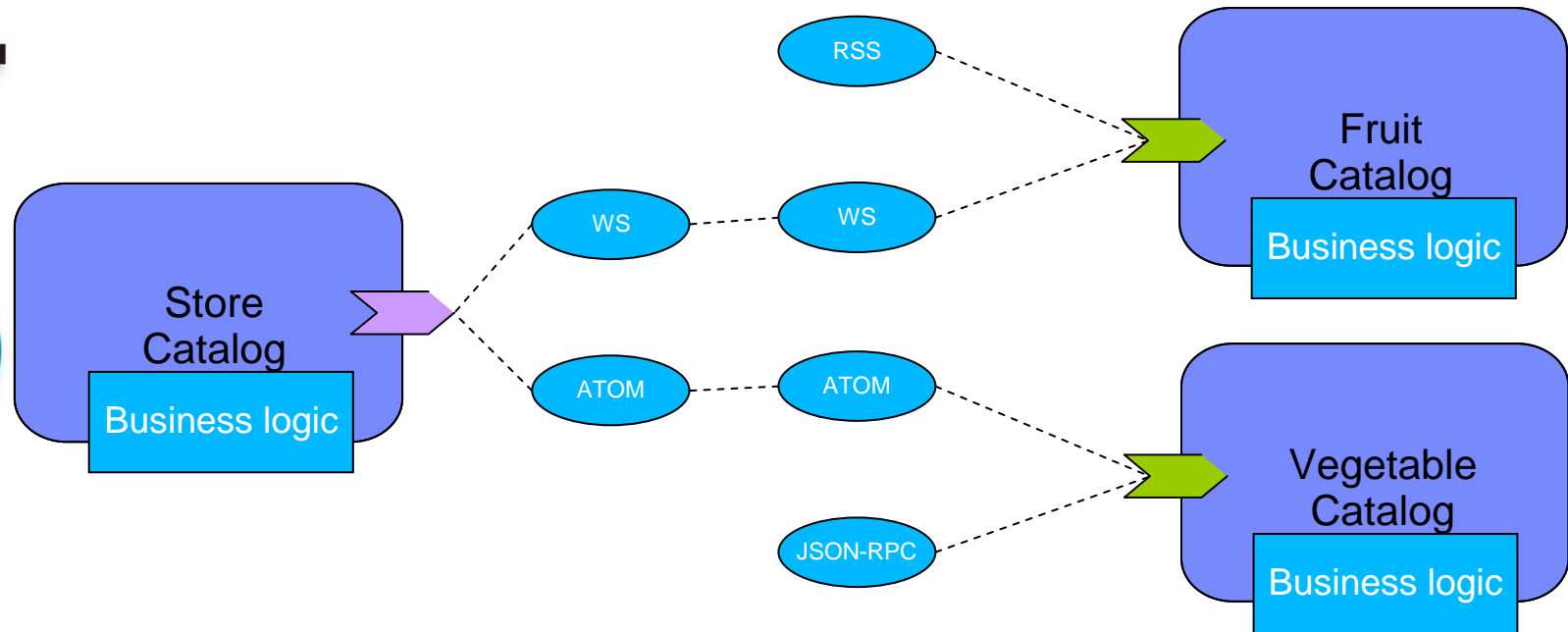 - Front End web store application

SCA Overview

Service Interface
 - **Java**
 - **WSDL**

Property

Reference Interface
 - **Java**
 - **WSDL**

Composite A

property setting

Service

Component A

Component B

Reference

promote

wire

promote

Service Binding
 Web Service
 JMS
 SLSB
 Rest
 JSONRPC
 JCA

Implementation
 Java
 BPEL
 PHP
 SCA composite
 Spring
 EJB module
 …

Reference Binding
 Web Service
 JMS
 SLSB
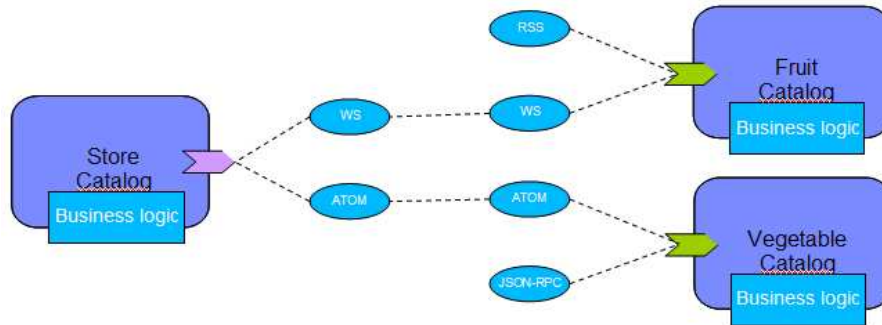 **Rest**
 JSONRPC
 JCA
 …

SCA composition is simple and flexible

- No more plumbing 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.

Store Catalog — Business logic

WS — WS — RSS → Fruit Catalog — Business logic

ATOM — ATOM → Vegetable Catalog — Business logic

JSON-RPC

Store Composition


```xml
<composite          xmlns=http://www.osoa.org/xmlns/sca/1.0
      xmlns:t=http://tuscany.apache.org/xmlns/sca/1.0
      targetNamespace=http://store   name="catalog">
      <component name="Catalog">
        <implementation.java class="services.CatalogImpl"/>
          <service name="Catalog">
            <t:binding.jsonrpc/>
          </service>
          <reference name="fruitsCatalog" target="FruitsCatalog">
            <binding.ws/>
          </reference>
          <reference name="vegetablesCatalog"
target="VegetablesCatalog">
            <t:binding.atom/>
          </reference>
      </component>
      <component name="FruitsCatalog">
        <implementation.java class="services.FruitsCatalogImpl"/>
        <service name="Catalog">
          <binding.ws/>
          <t:binding.rss/>
        </service>
      </component>
      <component name="VegetablesCatalog">
        <implementation.java
class="services.VegetablesCatalogImpl"/>
        <service name="Catalog">
          <t:binding.atom/>
          <t:binding.jsonrpc/>
        </service>
      </component>
</composite>
```
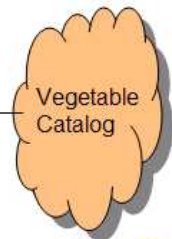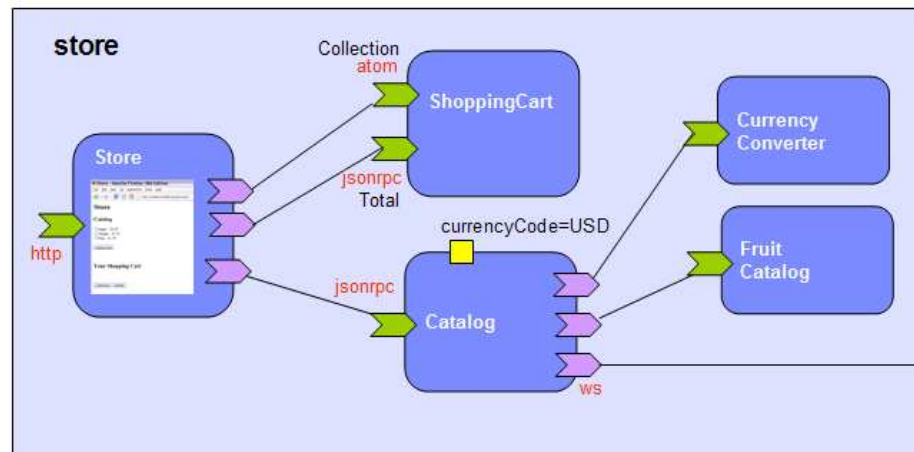
Tuscany SCA and Web 2.0

- Web 2.0 bindings: JSONRPC, DWR, Feed (ATOM, RSS)
- Tuscany Widget implementation to represent web components and provide dependency injection for JavaScript
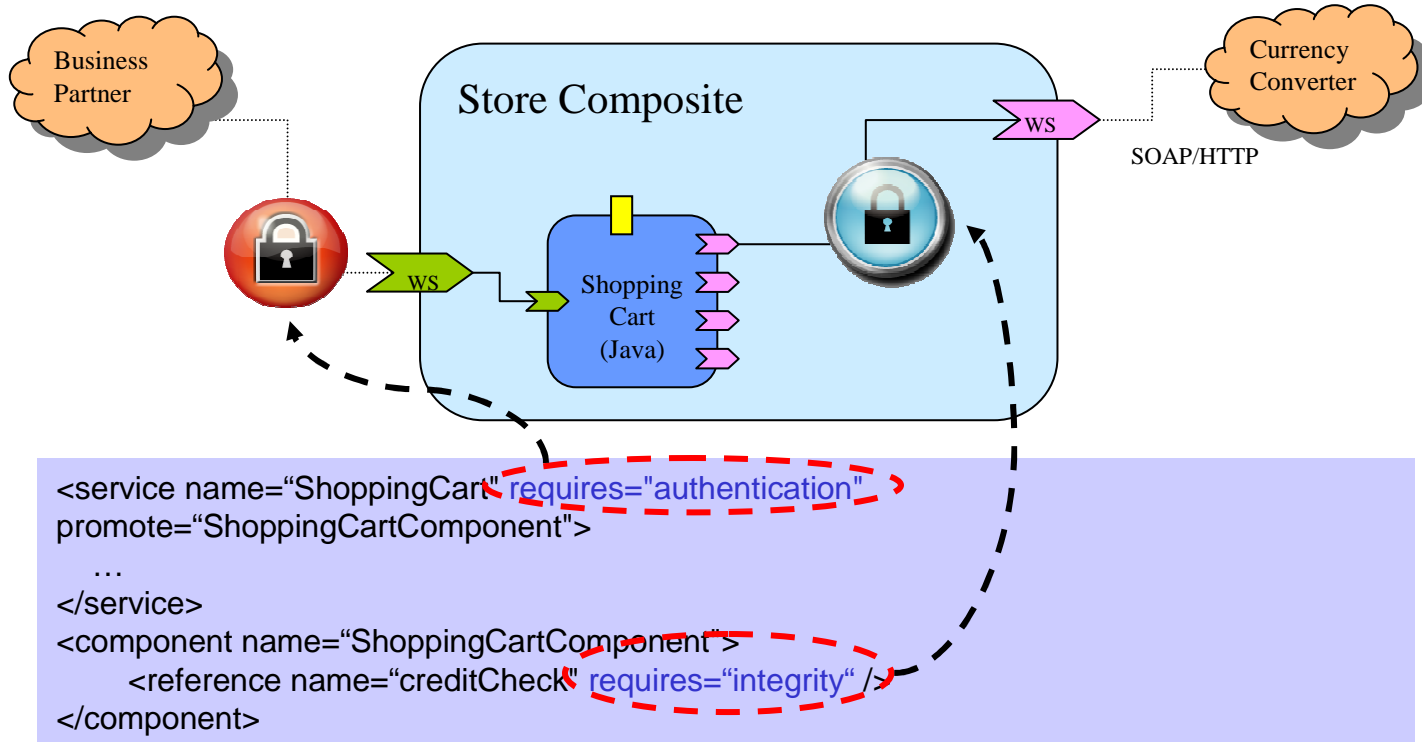- Other scripting implementation types

Enforcing the enterprise capabilities/constraints

- Intent: State the QoS requirements in a high-level abstract form, independent of the detailed configuration of the runtime and bindings
  - Security: authentication, authorization, integrity
  - Transaction: managedTransaction.global, noManagedTransaction, supendsTransaction, propagatesTransaction
- Policy: Describe some capability or constraint that can be applied to service components or to the interactions between service components represented by services and references.
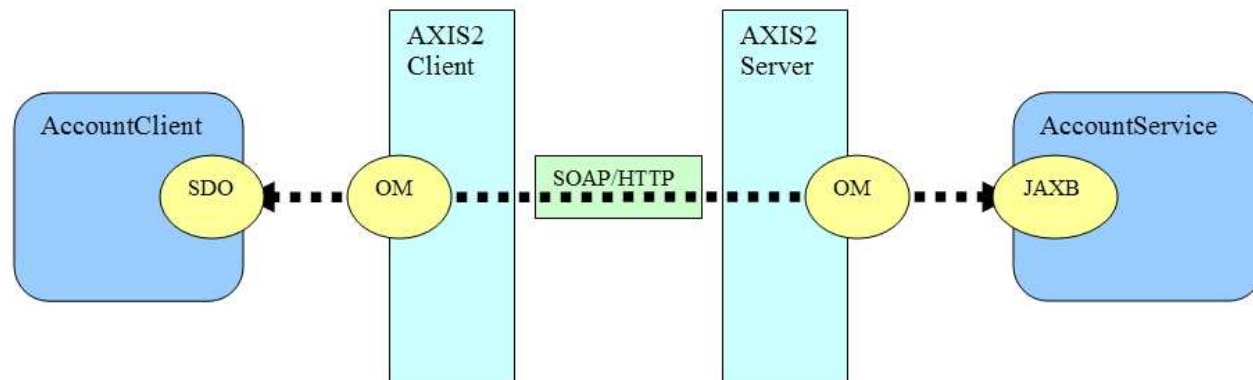
Enforcing the enterprise capabilities/constraints


```
<service name="ShoppingCart" requires="authentication"
promote="ShoppingCartComponent">
  …
</service>
<component name="ShoppingCartComponent">
    <reference name="creditCheck" requires="integrity" />
</component>
```
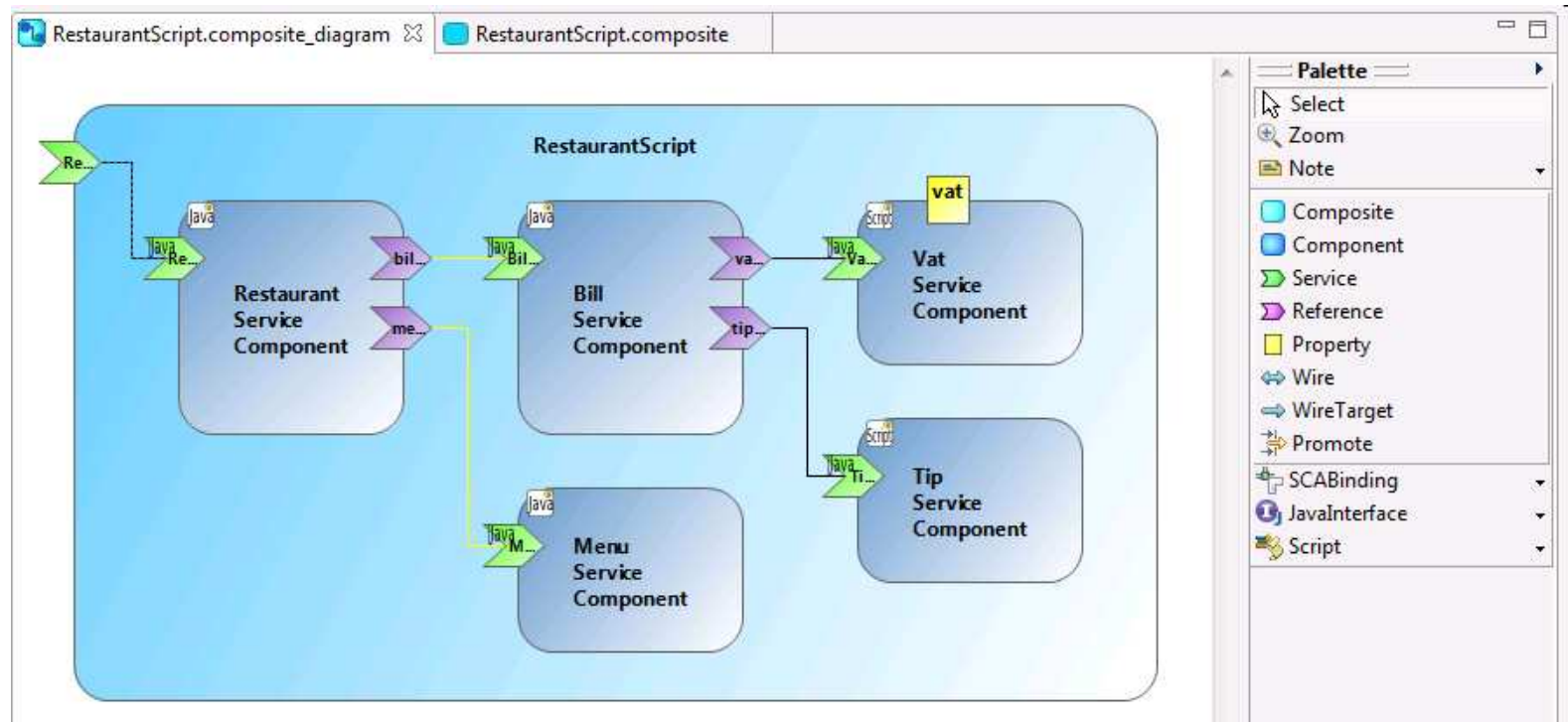
Handling data representations

- Business services talk to each other by data exchange. The same data can be represented in various formats.
  - A Customer can be represented as:
    - Plain Old Java Bean (POJO)
    - SDO Data Object, JAXB object, XML String, DOM Node, StAX XMLStreamReader, AXIOM
    - JSON

Eclipse Tools

- Eclipse STP SCA project

ApacheCon

Leading the Wave
of Open Source

Apache Tuscany
Demo
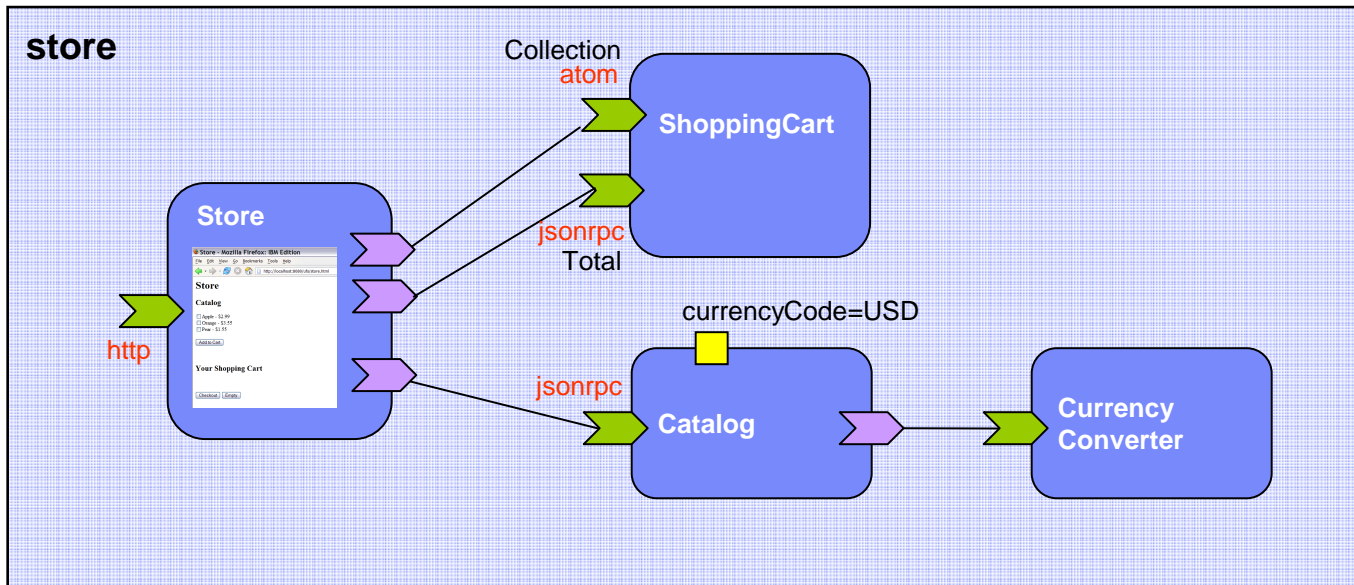## The Raise of a Fruit Business

http://www.rarefruit.org/photos/fruit_store.jpg
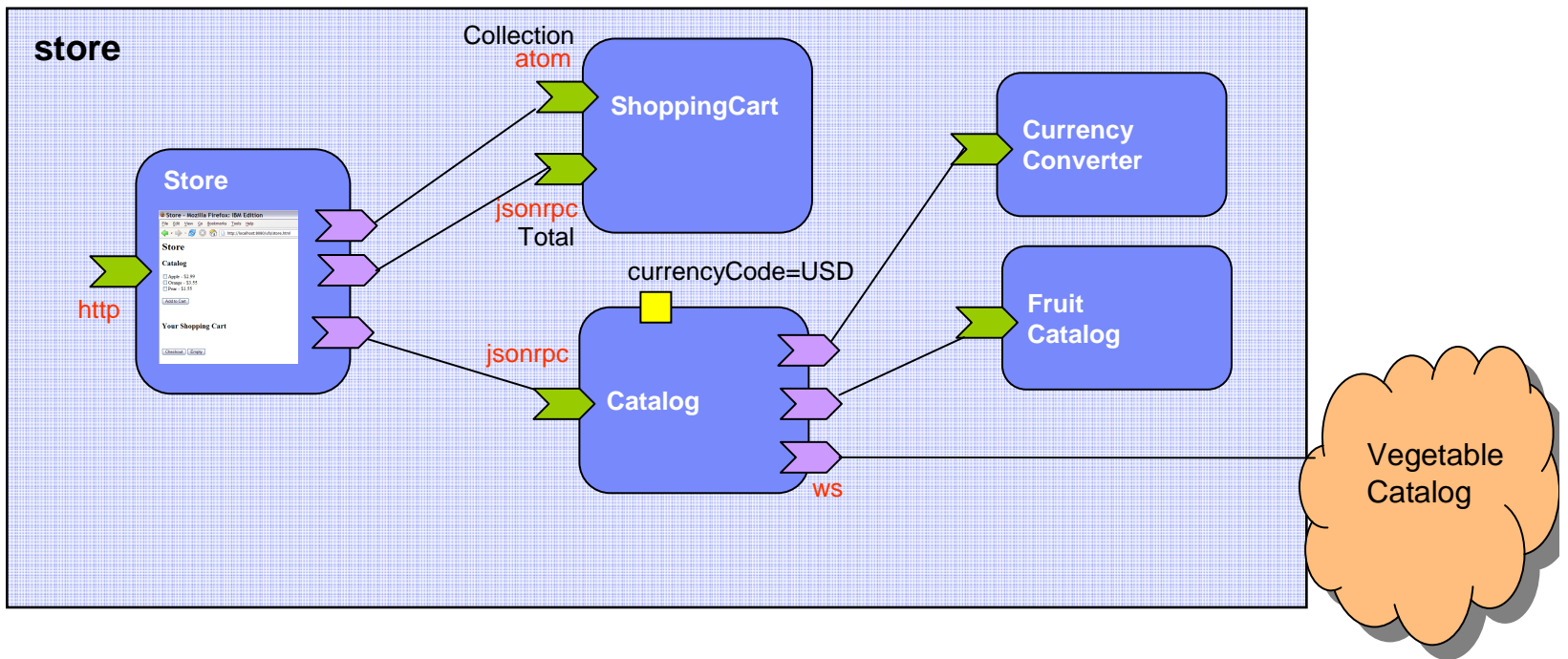
Leading the Wave of Open Source

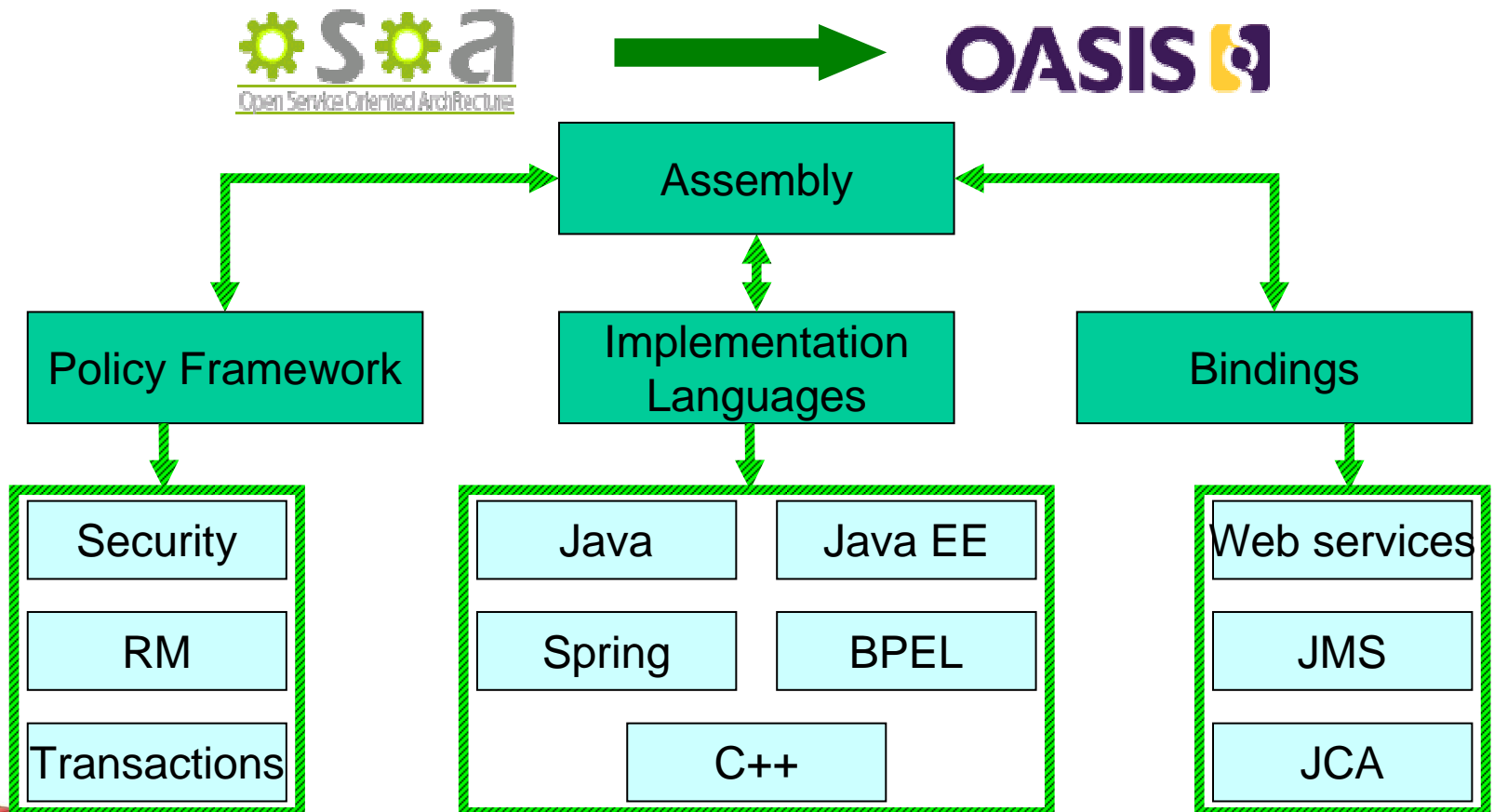# The Fruit Store

Creating an Online Business

The Fruit&Vegetable Store

Merger or Acquisition

The SCA Specifications

➢SCA is going through a formal standardization process in the OASIS OpenCSA (http://www.oasis-opencsa.org)

Resources – SCA

- **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
  - More information on that site
    - http://osoa.org/display/Main/SCA+Resources

Resources - Tuscany

- ## Apache Tuscany
  - http://tuscany.apache.org

- ## Getting Involved
  - http://tuscany.apache.org/getting-involved.html

- ## Tuscany SCA Java Releases
  - http://tuscany.apache.org/sca-java-releases.html

- ## Tuscany SCA Java Documentation
  - http://tuscany.apache.org/java-sca-documentation-menu.html

- ## Tuscany Dashboard
  - http://tuscany.apache.org/tuscany-dashboard.html

**Leading the Wave
of Open Source**

THANK YOU !!!

Help Tuscany by taking 2 min user survey
– http://tuscany.apache.org/survey.html

**Leading the Wave
of Open Source**