

SCA Reaches the Cloud

Developing Composite Applications for the Cloud with Apache Tuscany

Luciano Resende
lresende@apache.org
<http://lresende.blogspot.com>



Jean-Sebastien Delfino
jsdelfino@apache.org
<http://jsdelfino.blogspot.com>



Agenda

- **Cloud Computing – Goals and Challenges**
- **SCA – Goals and Overview**
- **SCA – Typical Scenarios**
- **Apache Tuscany**
- **Tuscany Demo – Rewiring Components in the Cloud**
- **Apache Nuvem**
- **Nuvem Demo – Cloud friendly Components**
- **Your Wish list?**
- **Getting Involved**

Cloud Computing

Cloud Computing – Some Goals

- Up and running in seconds
- Cheap
- Scale up and down
- Agile, reconfigure applications as business evolves



Cloud Computing – Not so easy?

- **Different platforms and APIs (even languages) to learn... does my business logic need to know?**
- **Am I getting OS images on demand? Infrastructure? an application platform?**
- **Changing pricing models?**
- **How do I integrate hybrid clouds, on premise + public cloud?**
- **How do the various parts of my app communicate? Which protocols am I using?**
- **How do I assemble / integrate them?**
- **How do I configure the QOS I need?**
- **How do I automate deployment?**
- **Can I move some parts from one cloud to another?**

What is SCA ?

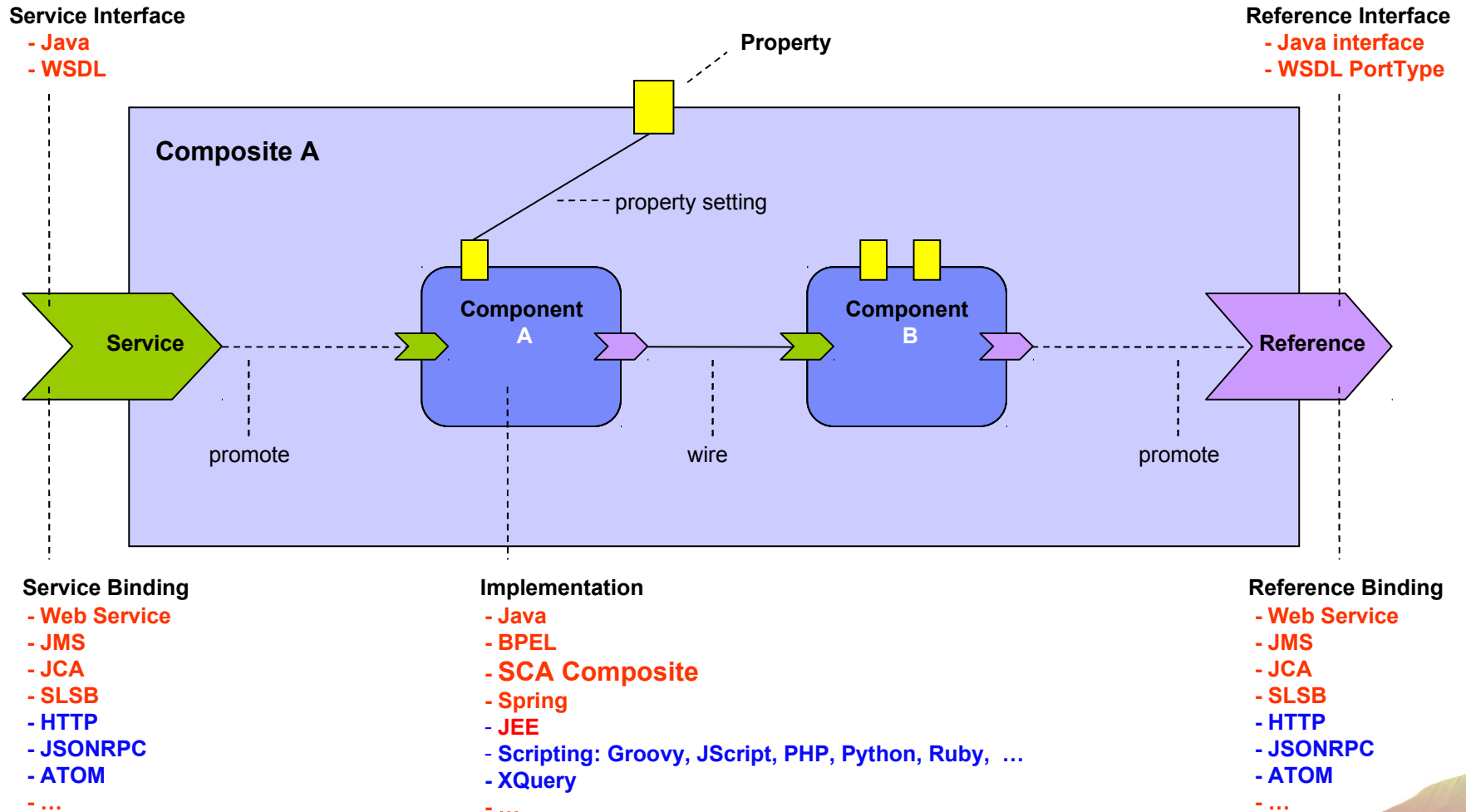
SCA - Goals

- **Abstract out technical APIs, protocols, QOS**
- **Allow me to focus on my business logic**
- **Give me a structure for componentizing my app**
- **Help me assemble, wire, rewire, move parts around**
- **OASIS Standard (in progress)**
- **Open Source implementations**
 - Apache Tuscany
 - Fabric3
 - a few others
- **Product implementations**
- **Initial target: SOA, Web Services, multi-language apps, application integration**

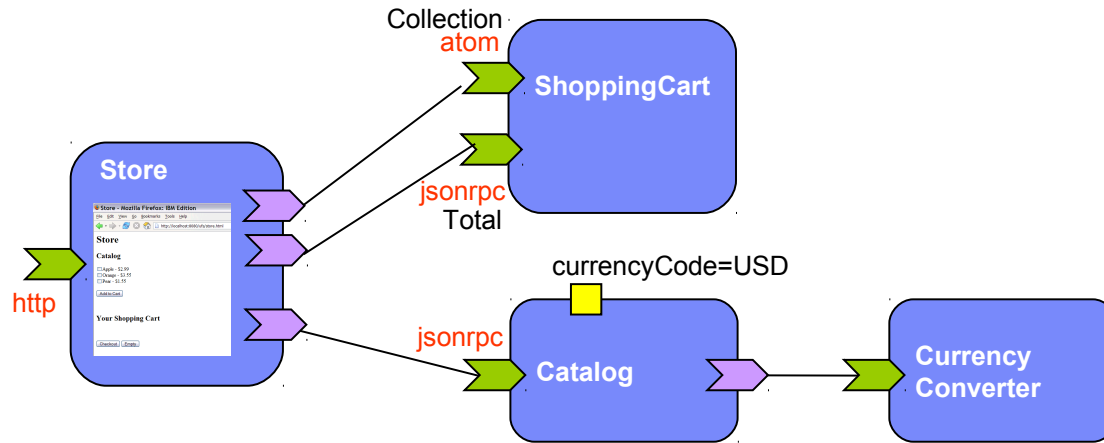
Can SCA components help you in the cloud?

- **We've been using different clouds in our Apache Tuscany work and are starting to realize that SCA components can help there too!**
- **Components that easily communicate over a network**
- **Components that shield you from different infrastructures**
- **A way to describe your app and how it's assembled / wired**
- **and can be distributed in a cloud or multiple clouds**
- **Move components around clouds and rewire your app**

SCA – Assembly Model



SCA – Example assembly



```
<composite xmlns="http://docs.oasis-open.org/ns/opencsa/sca/200912"
  xmlns:t="http://tuscany.apache.org/xmlns/sca/1.1"
  targetNamespace="http://store"
  name="store">

  <component name="Store">
    <t:implementation.widget location="uiservices/store.html"/>
    <service name="Widget">
      <t:binding.http uri="/store"/>
    </service>
    <reference name="catalog" target="Catalog"/>
    <reference name="shoppingCart" target="ShoppingCart/Cart"/>
    <reference name="shoppingTotal" target="ShoppingCart/Total"/>
  </component>

  <component name="Catalog">
    <implementation.java class="services.FruitsCatalogImpl"/>
    <property name="currencyCode">USD</property>
    <service name="Catalog">
      <t:binding.jsonrpc/>
    </service>
    <reference name="currencyConverter" target="CurrencyConverter"/>
  </component>

  <component name="ShoppingCart">
    <implementation.java class="services.ShoppingCartImpl"/>
    <service name="Cart">
      <t:binding.atom uri="/ShoppingCart/Cart"/>
    </service>
    <service name="Total">
      <t:binding.jsonrpc/>
    </service>
  </component>

  <component name="CurrencyConverter">
    <implementation.java class="services.CurrencyConverterImpl"/>
  </component>

</composite>
```

SCA – if you don't like XML

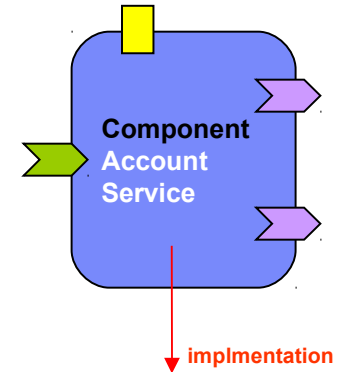
```
final Composite comp =
    build(composite("http://sample", "test",
        component("client-test",
            implementation(ClientTest.class,
                service(Client.class),
                reference("jello", Hello.class),
                reference("wello", Hello_wsdl)),
            reference("jello", "jello-test"),
            reference("wello", "wello-test")),
        component("wello-test",
            implementation(WelloTest.class,
                service(Hello_wsdl),
                reference("upper", Upper_wsdl)),
            reference("upper", "upper-test")),
        component("jello-test",
            implementation(JelloTest.class,
                service(Hello.class),
                reference("upper", Upper.class)),
            reference("upper", "upper-test")),
        component("upper-test",
            implementation(UpperTest.class,
                service(Upper.class))), ec);
```

SCA – if you like Java annotations

```
@Remotable
public interface AccountService {

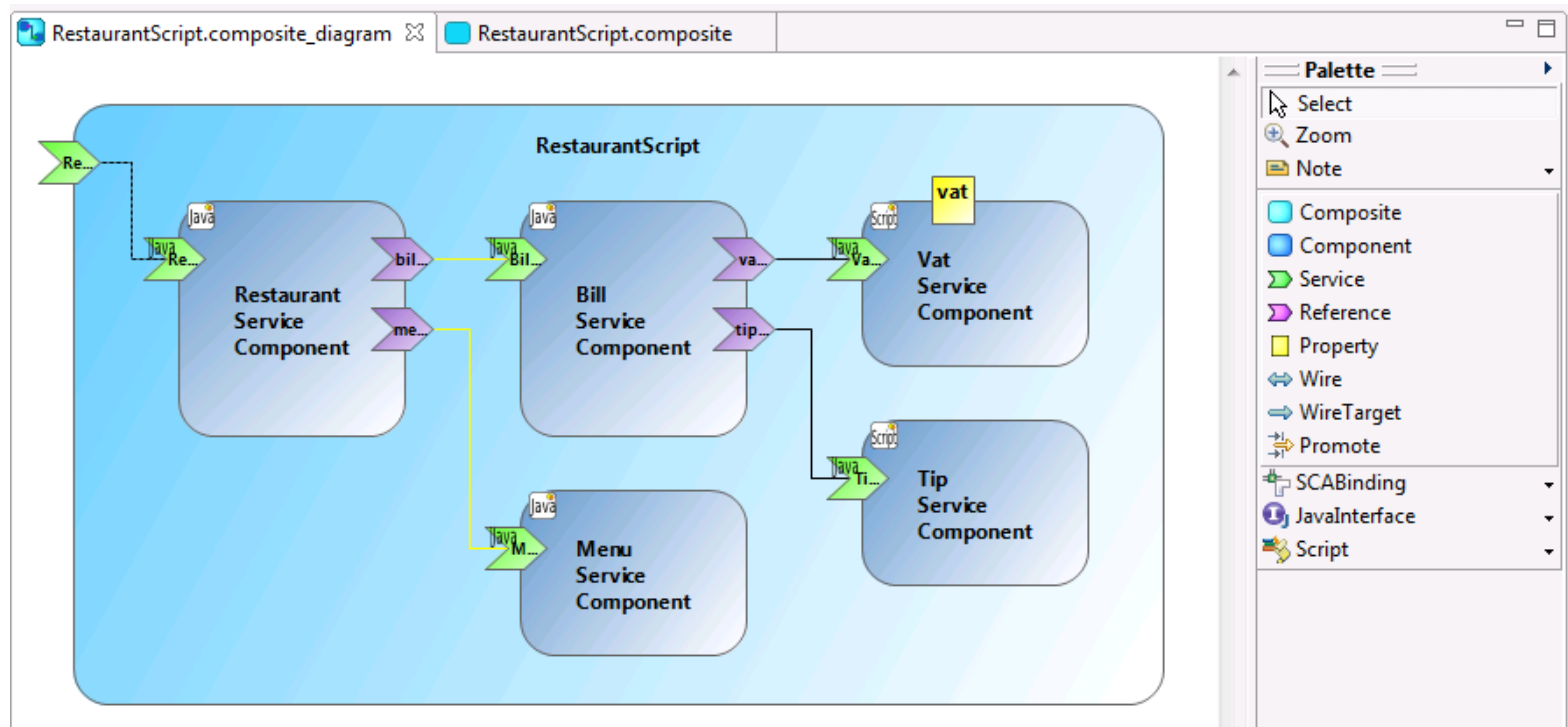
    AccountReport getAccountReport(String customerID);
}

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;
    }
    ...
}
```



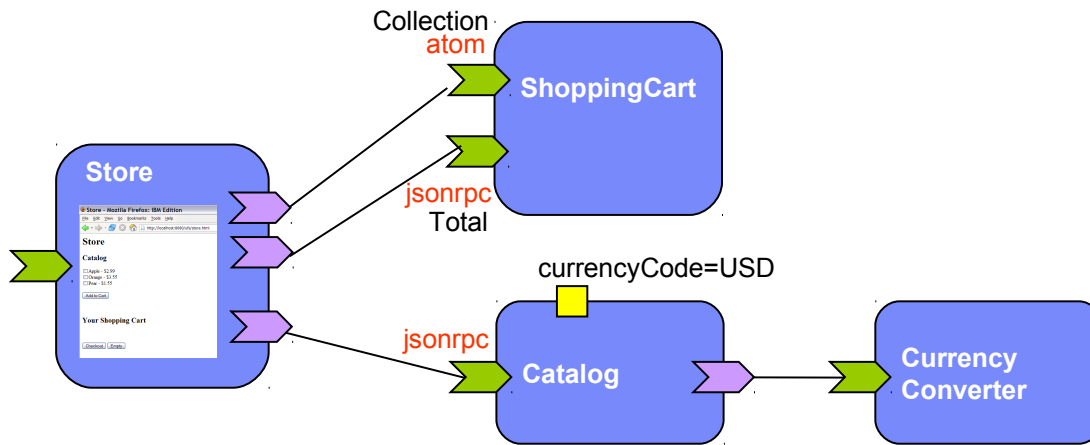
SCA – if you like to click around

- Eclipse STP Tools project provides SCA tooling



SCA – Typical Scenarios

Online Store



```

<composite xmlns="http://docs.oasis-open.org/ns/opencsa/sca/200912"
  xmlns:t="http://tuscany.apache.org/xmlns/sca/1.1"
  targetNamespace="http://store"
  name="store">

  <component name="Store">
    <t:implementation.widget location="uiservices/store.html"/>
    <service name="Widget">
      <t:binding.http uri="/store"/>
    </service>
    <reference name="catalog" target="Catalog"/>
    <reference name="shoppingCart" target="ShoppingCart/Cart"/>
    <reference name="shoppingTotal" target="ShoppingCart/Total"/>
  </component>

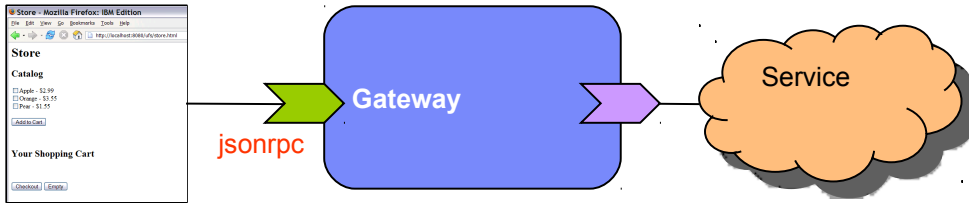
  <component name="Catalog">
    <implementation.java class="services.FruitsCatalogImpl"/>
    <property name="currencyCode">USD</property>
    <service name="Catalog">
      <t:binding.jsonrpc/>
    </service>
    <reference name="currencyConverter" target="CurrencyConverter"/>
  </component>

  <component name="ShoppingCart">
    <implementation.java class="services.ShoppingCartImpl"/>
    <service name="Cart">
      <t:binding.atom uri="/ShoppingCart/Cart"/>
    </service>
    <service name="Total">
      <t:binding.jsonrpc/>
    </service>
  </component>

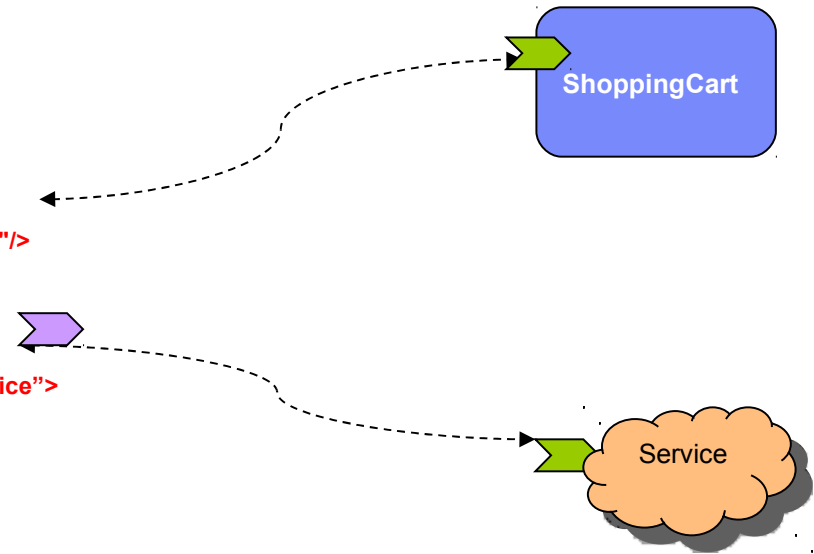
  <component name="CurrencyConverter">
    <implementation.java class="services.CurrencyConverterImpl"/>
  </component>

</composite>
  
```

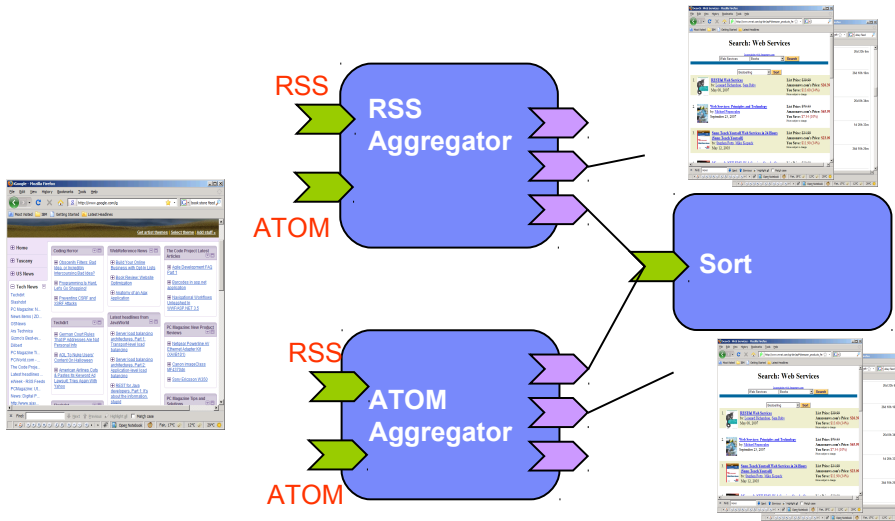
Gateway / Mediation



```
<composite xmlns=http://docs.oasis-open.org/ns/opencsa/sca/200903
xmlns:t=http://tuscany.apache.org/xmlns/sca/1.1
targetNamespace=http://store name="gateway">
  <component name="Gateway">
    <implementation.java class="services.GatewayImpl"/>
    <service name="A">
      <t:binding.jsonrpc/>
    </service>
    <reference name="refService">
      <binding.x uri="http://domain:8080/atomService">
    </reference>
  </component>
</composite>
```



Feed Aggregator / Converter



```
<composite xmlns=http://docs.oasis-open.org/ns/opencsa/sca/200903
xmlns:t=http://tuscany.apache.org/xmlns/sca/1.1
targetNamespace=http://store name="feedAgregator">
```

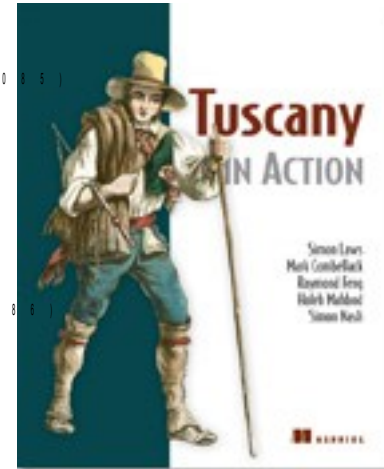
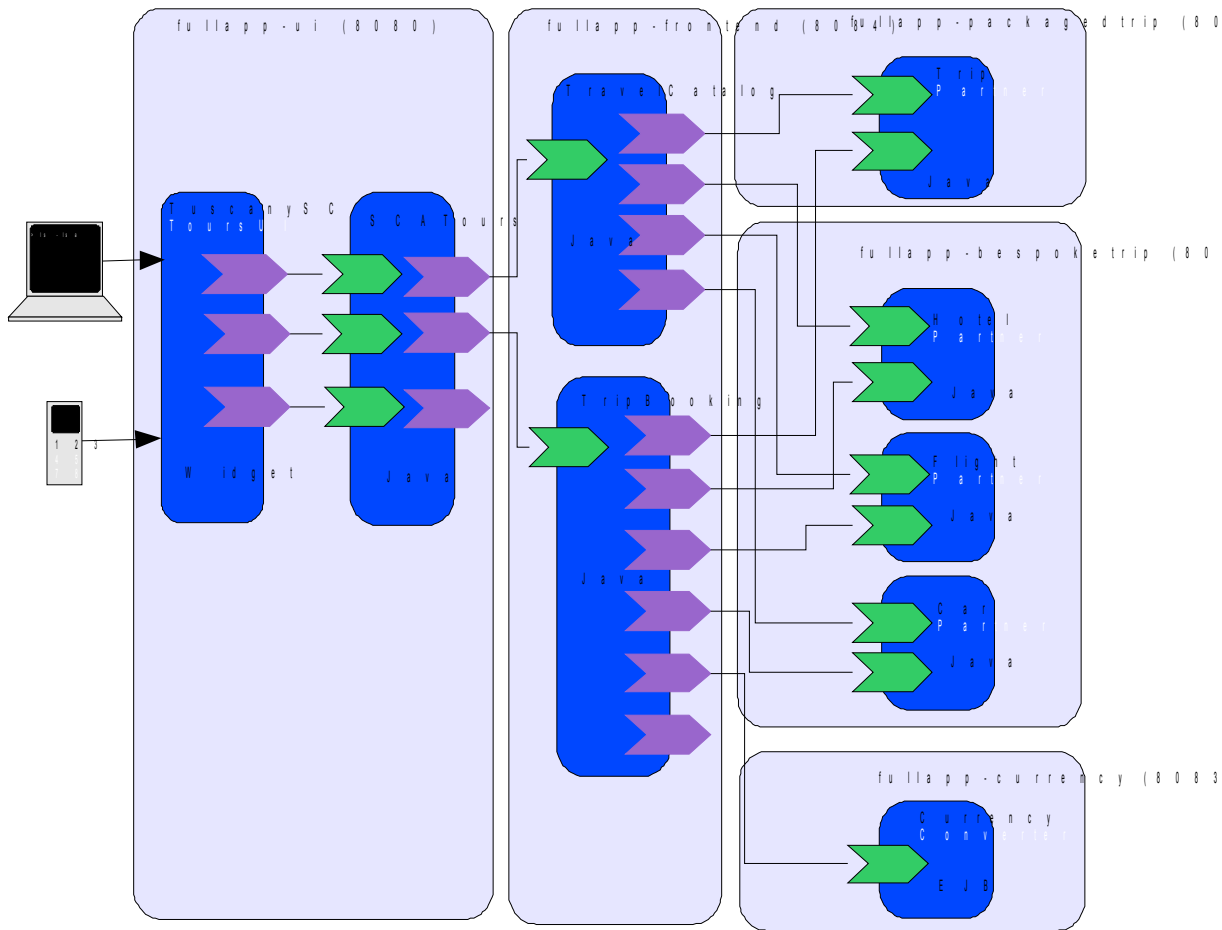
```
<component name="AtomAggregator">
  <implementation.java class="feed.AggregatorImpl"/>
  <reference name="sort" target="Sort"/>
```

```
<reference name="atomFeed1">
  <tuscany:binding.atom
    uri="http://apache-tuscany.blogspot.com/feeds/posts/default"/>
</reference>
<reference name="atomFeed2">
  <tuscany:binding.atom
    uri="http://feeds.feedburner.com/blogspot/Dcni?format=xml"/>
</reference>
<property name="feedTitle">Atom Aggregator Sample</property>
</component>
```

```
<component name="Sort">
  <implementation.java class="feed.SortImpl"/>
  <property name="newFirst">true</property>
</component>
```

```
</composite>
```

Business Integration – Travel Booking Process



JEE Components
POJOs
Spring Assemblies
Scripting Components
BPEL Processes

Apache Tuscany

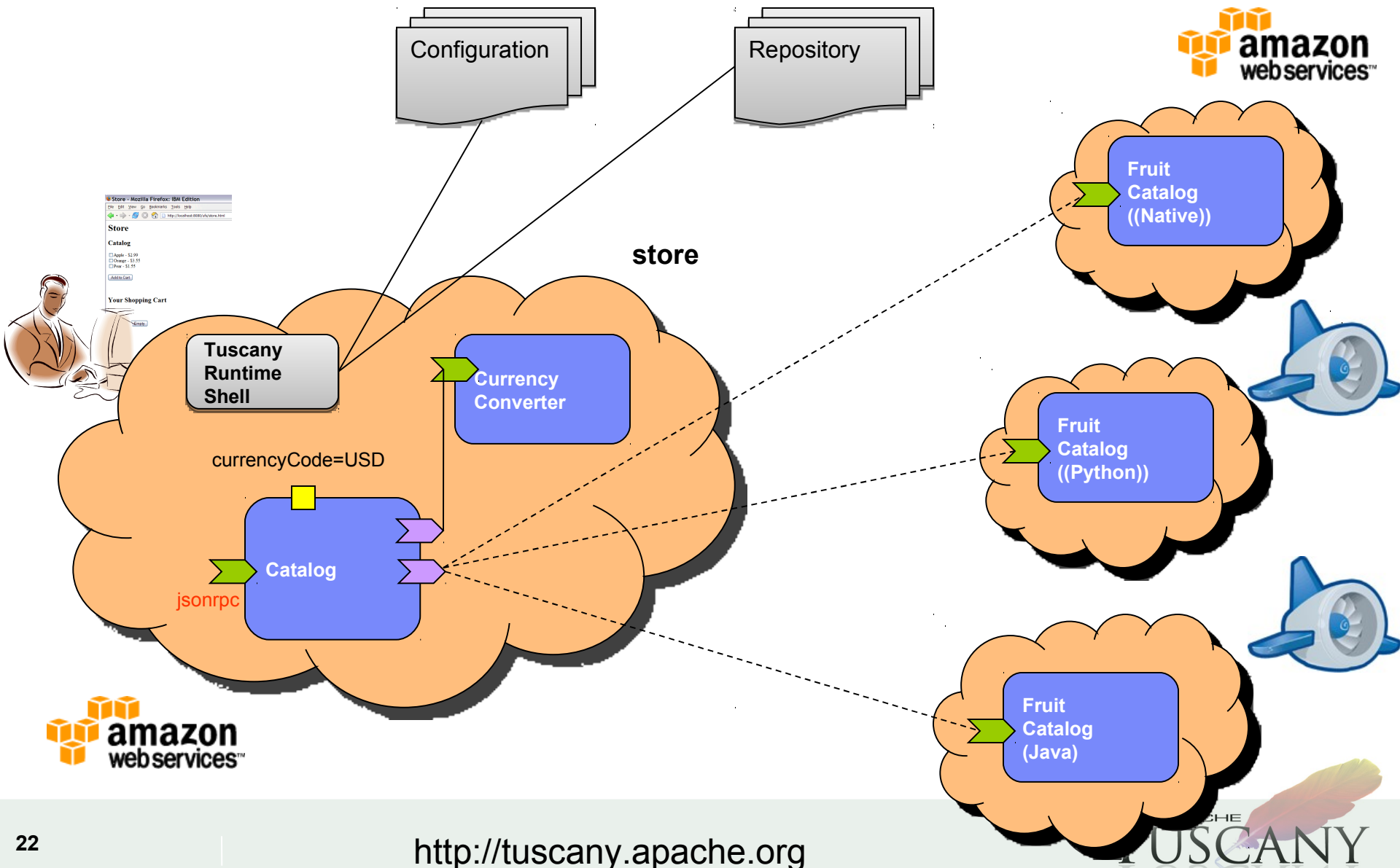
Apache Tuscany

- **Lightweight SCA runtimes**
- **Leverage and integrate with the Apache platform**
- **Active Open-Source community, started in 2005**
- **“Release early release often”, many releases**
- **Two release streams, 1.x (stable), 2.x (trunk)**
- **Working on OASIS SCA compliance**
- **Innovations beyond the SCA spec (JSON, REST, ATOM, Comet etc)**
- **SCA Java runtime, standalone or on Google AppEngine / Java, supports Java, scripting, BPEL, Spring components, and many protocol bindings**
- **SCA Python runtime on Google AppEngine / Python**
- **SCA Native, supports C++ and Python components**

Tuscany Demo – SCA Component Rewiring

- **SCA Java Application on EC2**
- **Push one component out to Google AppEngine / Java**
- **Rewrite it in Python and move it Google AppEngine / Python**
- **Move it to a native SCA runtime on EC2**
- **Easy runtime reconfiguration as you rewire the app**
- **You've got choices, and can be agile!**

Tuscany Demo – SCA Component Wiring



Apache Nuvem Components for the Cloud

Apache Nuvem - Overview

- **New project in the Apache incubator**
- **Initial code contribution from Apache Tuscany**
- **A few technical components already there**
- **Running on Google AppEngine, today's demo also on EC2**
- **Project is just starting so there's a lot of room for innovation!**

Nuvem, REST, and Cloud friendly Components

- **With REST, components get a simple GET/POST/PUT/DELETE interface**
- **More importantly it's a fixed interface**
- **Making components easier to assemble and compose**
- **Like Lego blocks!**

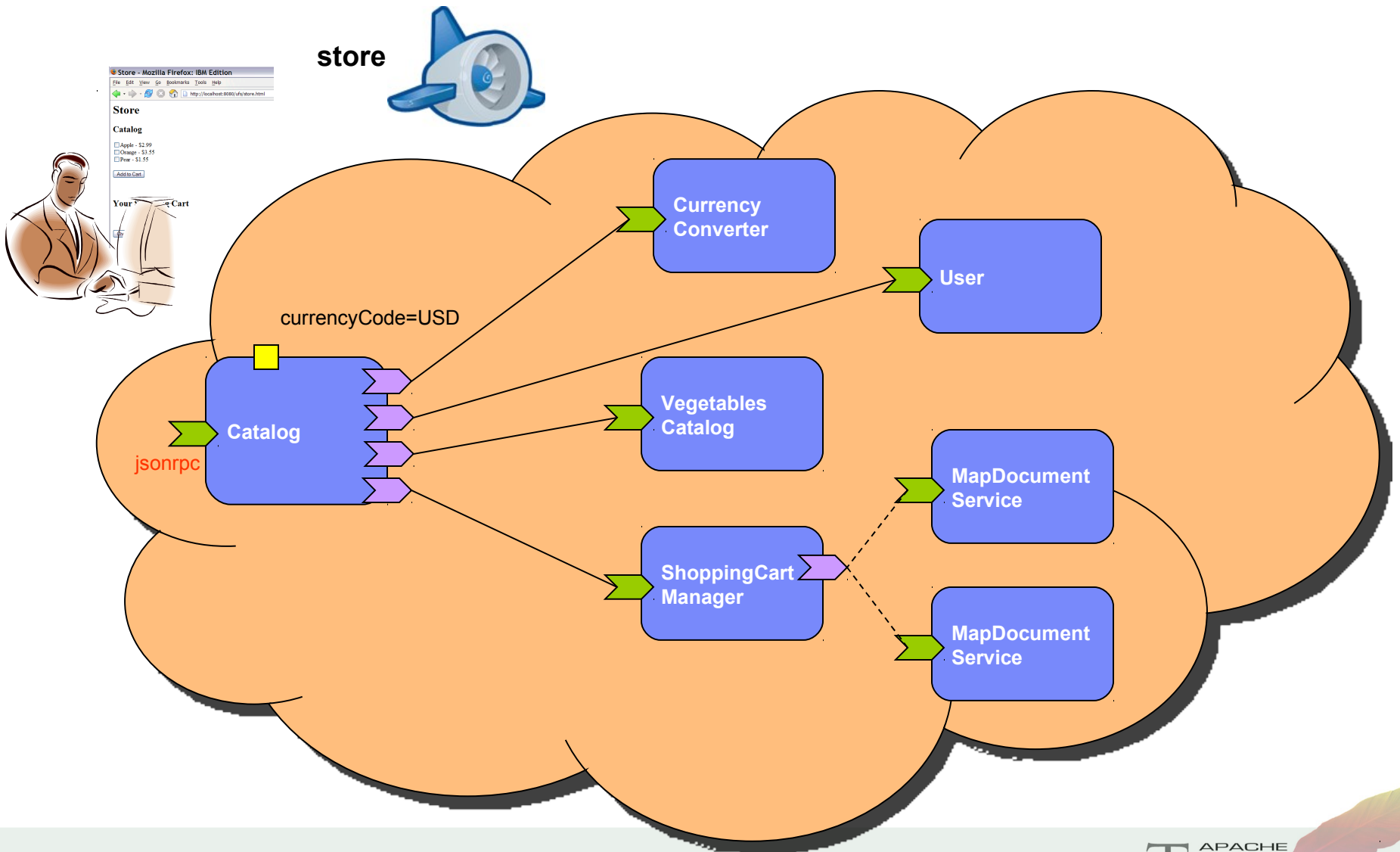
- **Web apps are starting to favor a protocol short list**
- **HTTP verbs with some format variations (XML, ATOM, RSS, JSON)**
- **That helps too!**

- **What if you had a palette of Cloud friendly components?**
- **Accessible through a simple REST interface**
- **To help simplify your apps and enable them to work on different clouds?**

Nuvm Demo – Technical Components

- **SCA Java runtime on Google AppEngine / Java**
- **Using different implementations of a simple datastore component**
- **First using a HashMap**
- **Second using Google's Memcached**

Nuvem Demo – Technical Components



Nuvm Components – Wish list

- **Simple data store cache**
- **Hierarchical cache, which can delegate to another cache**
- **Invocation cache, which caches responses to requests**
- **Key/value datastore**
- **Simple (S)QL datastore**
- **Datastore that understands master/slave replication and sharding**
- **XMPP chat**
- **Message queue**
- **Oauth 1.0/2.0 + OpenID**
- **User profile**

Cloud Components

➤ **What's your wish list?**

Getting Involved with Apache Tuscany

SCA - Resources

➤ **Good introduction to SCA**

- http://www.davidchappell.com/articles/Introducing_SCA.pdf

➤ **OASIS Open CSA – <http://www.oasis-opencsa.org>**

➤ **V1.1 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>

Apache Tuscany Resources

➤ 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-2x-releases.html>
- <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>

Getting Involved with Apache Nuvem

Apache Nuvem Resources

➤ Apache Nuvem

- <http://incubator.apache.org/nuvem/>

➤ Getting Involved

- <http://incubator.apache.org/nuvem/nuvem-getting-involved.html>

Thank You !!!

