Emerging Technologies for the Enterprise

March 28-29, 2007 Drexel University, Philadelphia, PA

AJAX Development with JSF

Look Ma! No JavaScript!

Kito D. Mann
Principal Consultant
Virtua™
technical excellence. it's that simple.

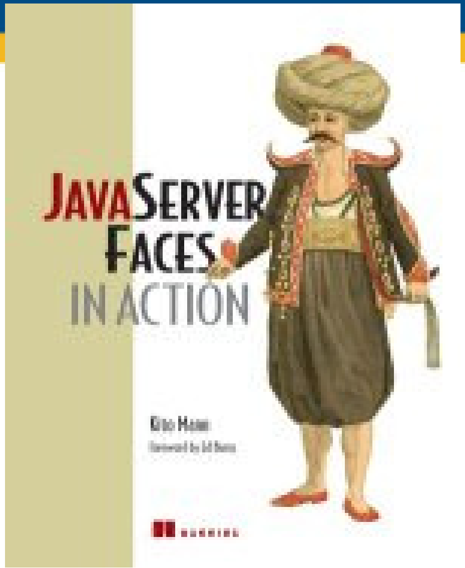
Look Ma! No JavaScript!

Learn how to take advantage of JSF components and libraries to provide AJAX behavior without writing JavaScript.

Kito D. Mann

- Trainer, consultant, architect, mentor
- Internationally recognized speaker
  - JavaOne, JavaZone, TSS Symposium, JavaPro Live!
- Author, JavaServer Faces in Action
- Founder, JSF Central
  - http://www.jsfcentral.com

Kito D. Mann



- JCP Member
  - JSF 1.2, JSP 2.1, Design-Time API for JavaBeans, Design-Time Metadata for JavaServer Faces Components, WebBeans, etc.
- Experience with Java since its release in 1995
- Web development since 1993

Agenda

- JavaServer Faces from 1,000 feet
- Ajaxian Faces
- Can’t we all get along?
- Which one?
- JSF 1.2 and beyond
- Summary
- Q&A

Agenda

- JavaServer Faces from 1,000 feet
- Ajaxian Faces
- Can’t we all get along?
- Which one?
- JSF 1.2 and beyond
- Summary
- Q&A

What is JavaServer Faces?

- Standard web user interface (UI) framework for Java
  - JSF 1.0: Standardized through Java Community Process (JCP) in 2004 (JSR 127)
  - JSF 1.2: Standardized through JCP in 2006 (JSR 252)
    - Part of Java EE 5.0

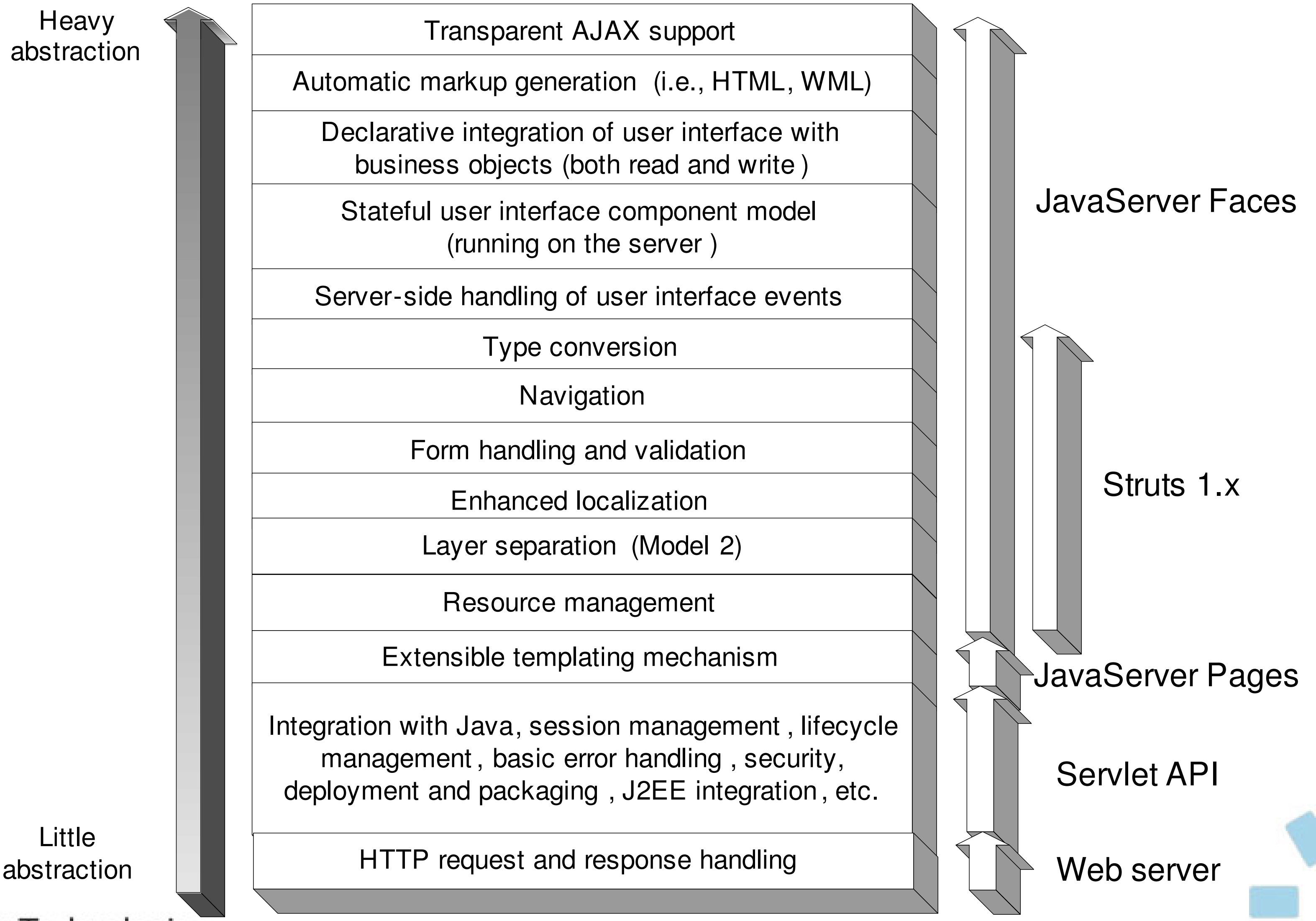
What is JavaServer Faces?

- Specification consists of:
  - Server side UI component and event model
  - Set of basic UI components
  - Basic application infrastructure
    - Can automatically synchronize UI components with application objects
    - Includes basic Dependency Injection container

What is JavaServer Faces?

- Extensive tool support
  - Sun, Oracle, IBM, BEA, Exadel, Borland, JetBrains, Genuietc, others
- Enables RAD-style approach to Java web development
- Facilitates third-party UI component market
- Built on top of Servlet API
- Works with JSP, but does not require it

JSF and Struts



Heavy abstraction

Transparent AJAX support

Automatic markup generation (i.e., HTML, WML)

Declarative integration of user interface with business objects (both read and write)

Stateful user interface component model (running on the server)

Server-side handling of user interface events

Type conversion

Navigation

Form handling and validation

Enhanced localization

Layer separation (Model 2)

Resource management

Extensible templating mechanism

Integration with Java, session management , lifecycle management , basic error handling , security, deployment and packaging , J2EE integration , etc.

HTTP request and response handling

Little abstraction

JavaServer Faces

Struts 1.x

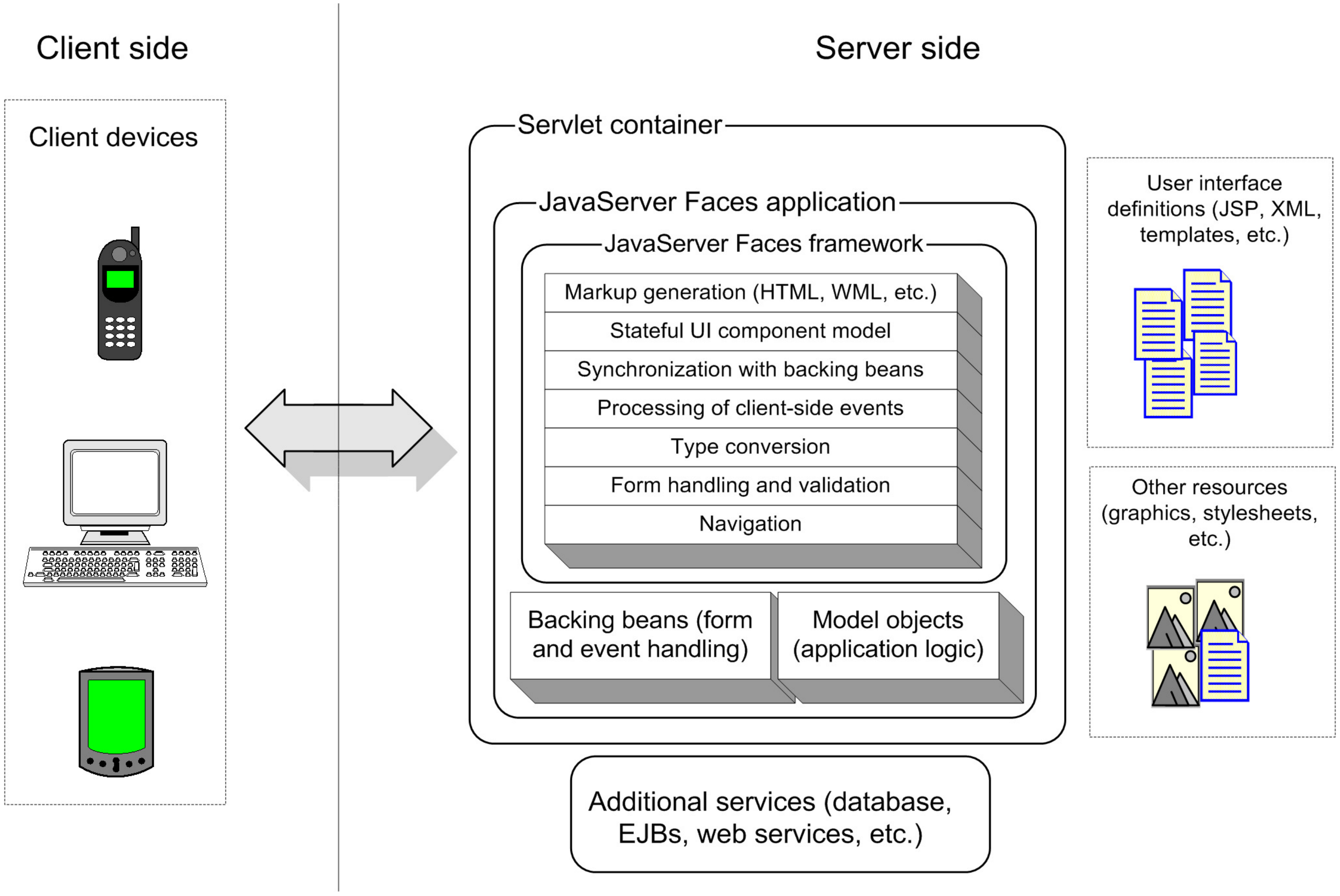
JavaServer Pages

Servlet API

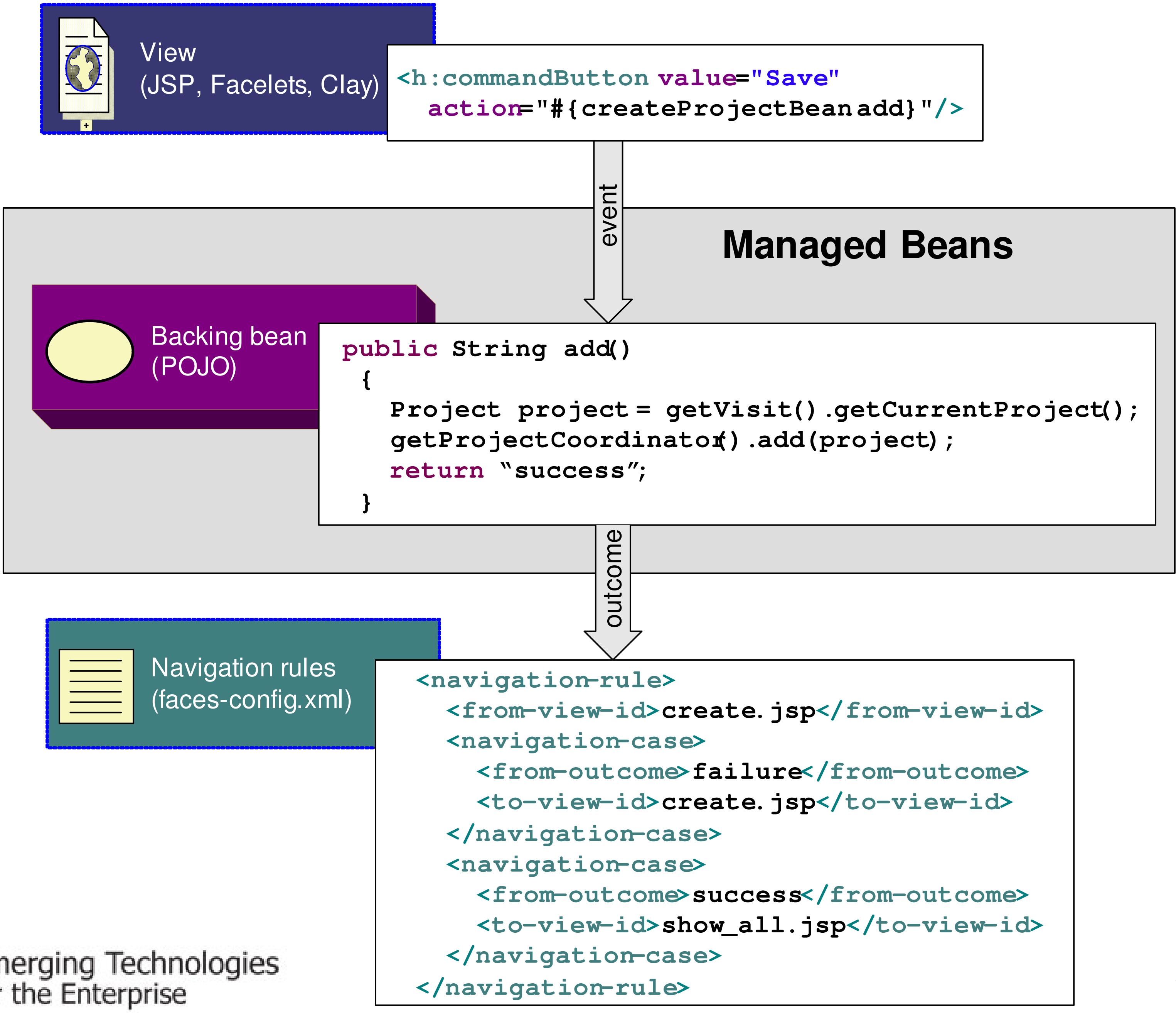
Web server

What is Java Server Faces?

JSF Application Architecture


Client side
Server side
Client devices
Servlet container
JavaServer Faces application
JavaServer Faces framework
Markup generation (HTML, WML, etc.)
Stateful UI component model
Synchronization with backing beans
Processing of client-side events
Type conversion
Form handling and validation
Navigation
Backing beans (form and event handling)
Model objects (application logic)
User interface definitions (JSP, XML, templates, etc.)
Other resources (graphics, stylesheets, etc.)
Additional services (database, EJBs, web services, etc.)

The JSF Programming Model



```
<h:commandButton value="Save"
  action="#{createProjectBean add}"/>
```

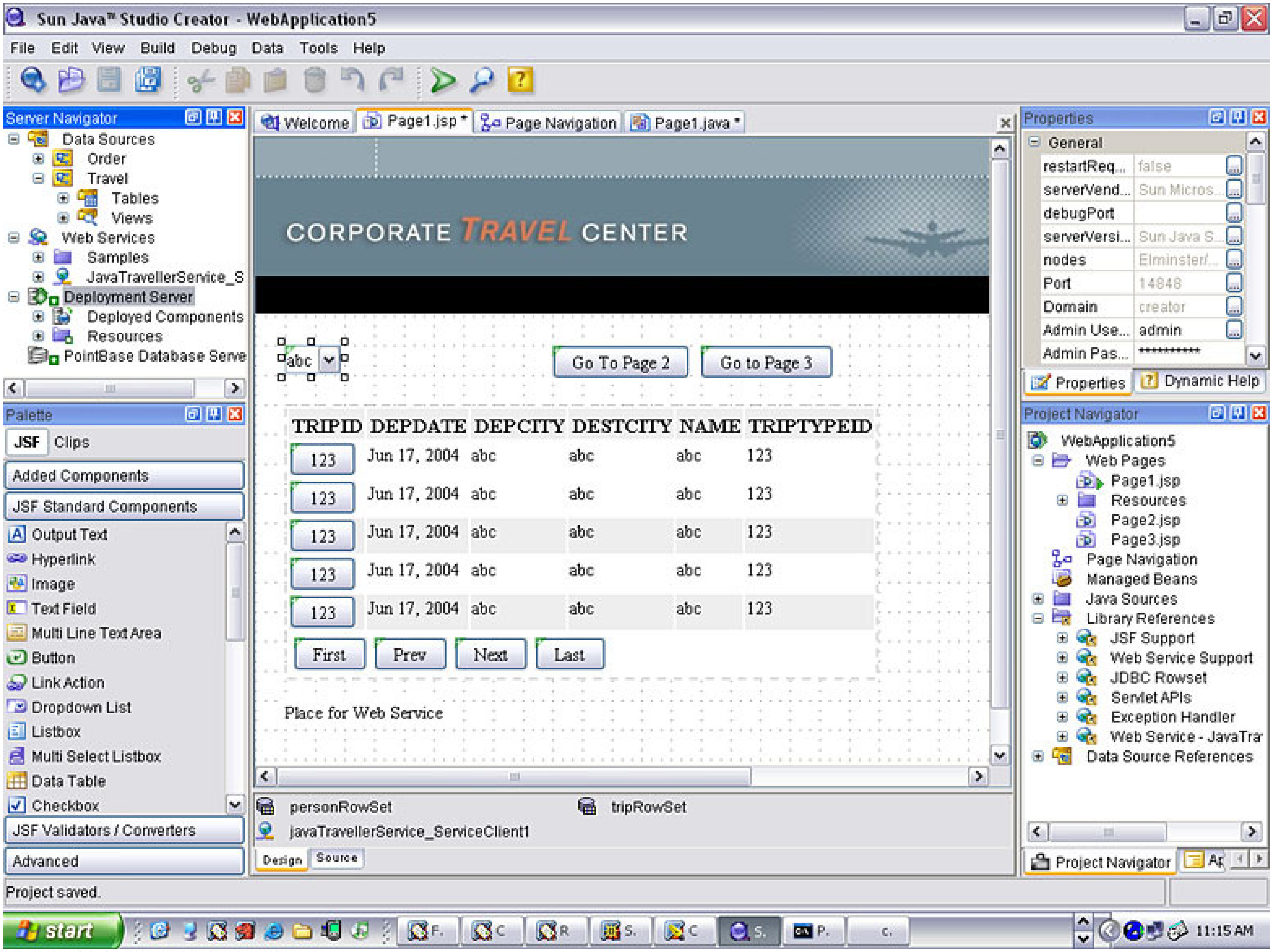
```
public String add()
 {
   Project project = getVisit().getCurrentProject();
   getProjectCoordinator().add(project);
   return “success”;
 }
```

```
<navigation-rule>
  <from-view-id>create.jsp</from-view-id>
  <navigation-case>
    <from-outcome>failure</from-outcome>
    <to-view-id>create.jsp</to-view-id>
  </navigation-case>
  <navigation-case>
    <from-outcome>success</from-outcome>
    <to-view-id>show_all.jsp</to-view-id>
  </navigation-case>
</navigation-rule>
```

The IDE Effect

- Designed to facilitate easy-to-use IDE integration
  - Since JSF is a standard, vendors more likely to create tools
- All major IDE vendors support JSF
- Level and style varies
- Caveat:
  - Generated artefacts
- Does not *require* tools

Sun Java Studio Creator

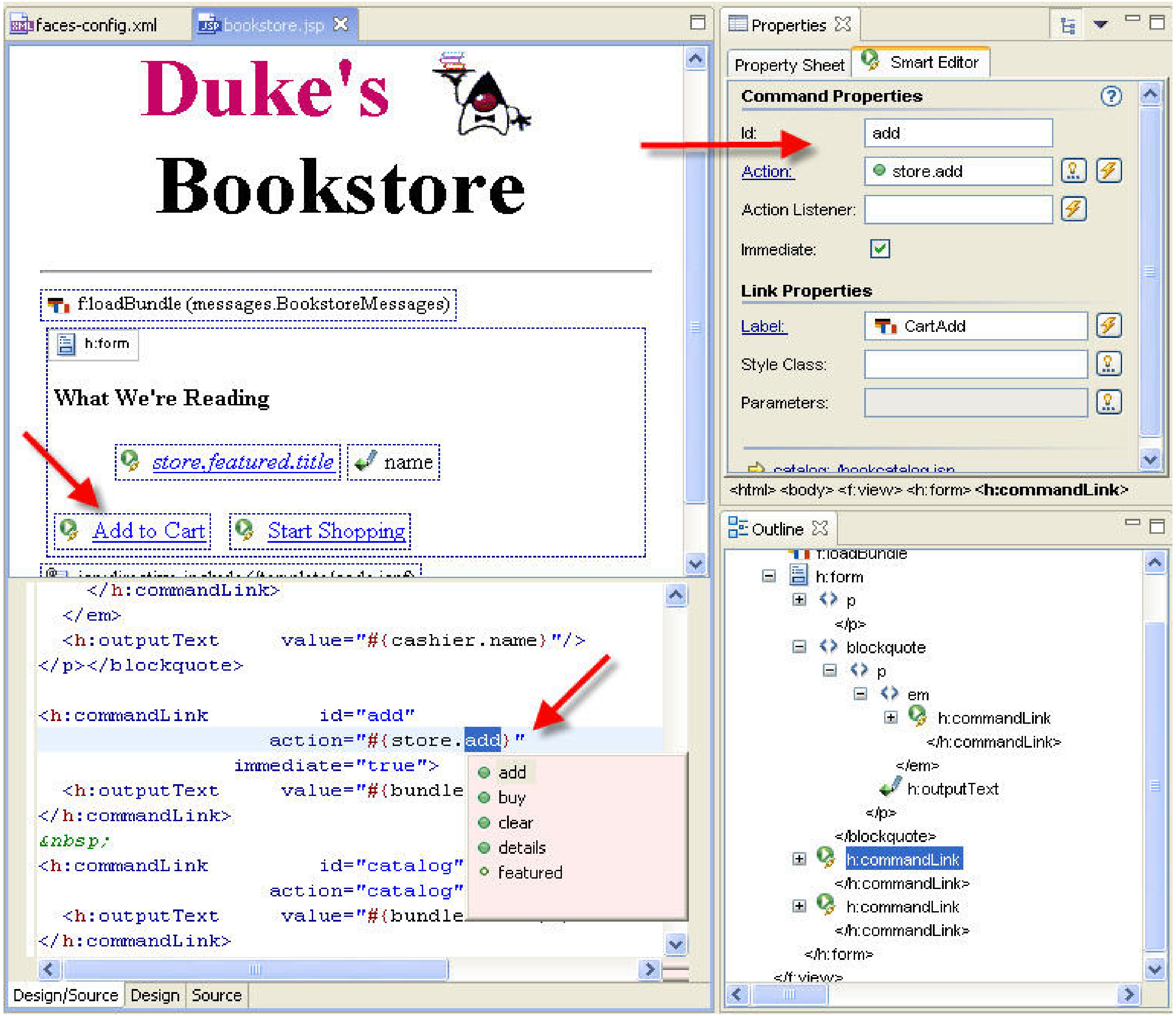


TRIPID	DEPDATE	DEPCITY	DESTCITY	NAME	TRIPTYPEID
123	Jun 17, 2004	abc	abc	abc	123
123	Jun 17, 2004	abc	abc	abc	123
123	Jun 17, 2004	abc	abc	abc	123
123	Jun 17, 2004	abc	abc	abc	123
123	Jun 17, 2004	abc	abc	abc	123

restartReq...	false
serverVend...	Sun Micros...
debugPort	
serverVersi...	Sun Java S...
nodes	Elminster/...
Port	14848
Domain	creator
Admin Use...	admin
Admin Pas...	**********

Project saved.

BEA Workshop Studio



```
    </h:commandLink>
  </em>
  <h:outputText     value="#{cashier.name}"/>
</p></blockquote>

<h:commandLink           id="add"
                  action="#{store.add}"
               immediate="true">
  <h:outputText     value="#{bundle
</h:commandLink>
&nbsp;
<h:commandLink           id="catalog"
                  action="catalog"
  <h:outputText     value="#{bundle
</h:commandLink>
```

- add
- buy
- clear
- details
- featured

Command Properties

Id: add

Action: store.add

Action Listener:

Immediate: ☑

Link Properties

Label: CartAdd

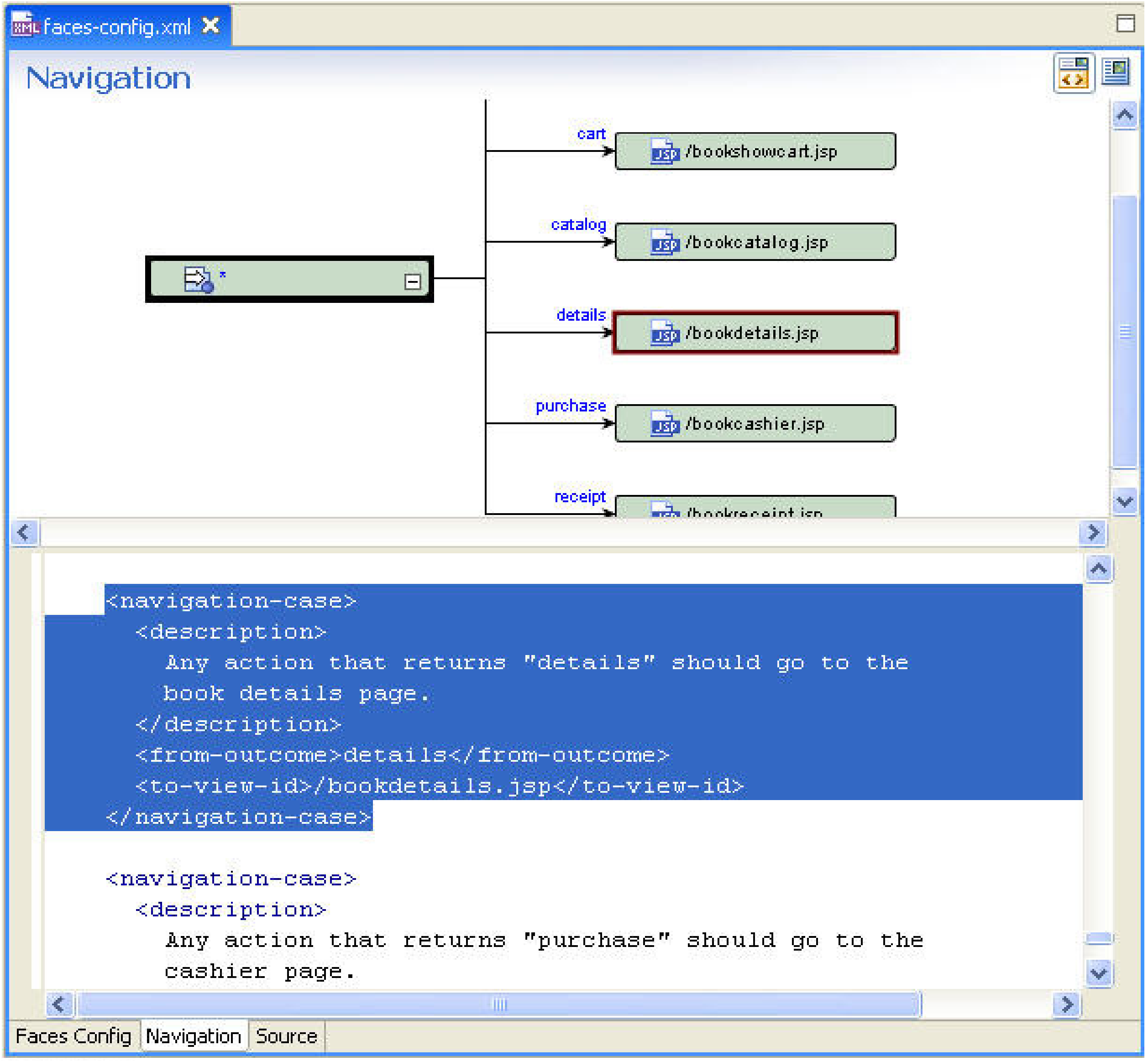
Style Class:

Parameters:

<html> <body> <f:view> <h:form> **<h:commandLink>**

```
f:loadBundle
⊟ h:form
  ⊞ p
    </p>
  ⊟ blockquote
    ⊟ p
      ⊟ em
        ⊞ h:commandLink
          </h:commandLink>
        </em>
        h:outputText
      </p>
    </blockquote>
  ⊞ h:commandLink
    </h:commandLink>
  ⊞ h:commandLink
    </h:commandLink>
  </h:form>
</f:view>
```

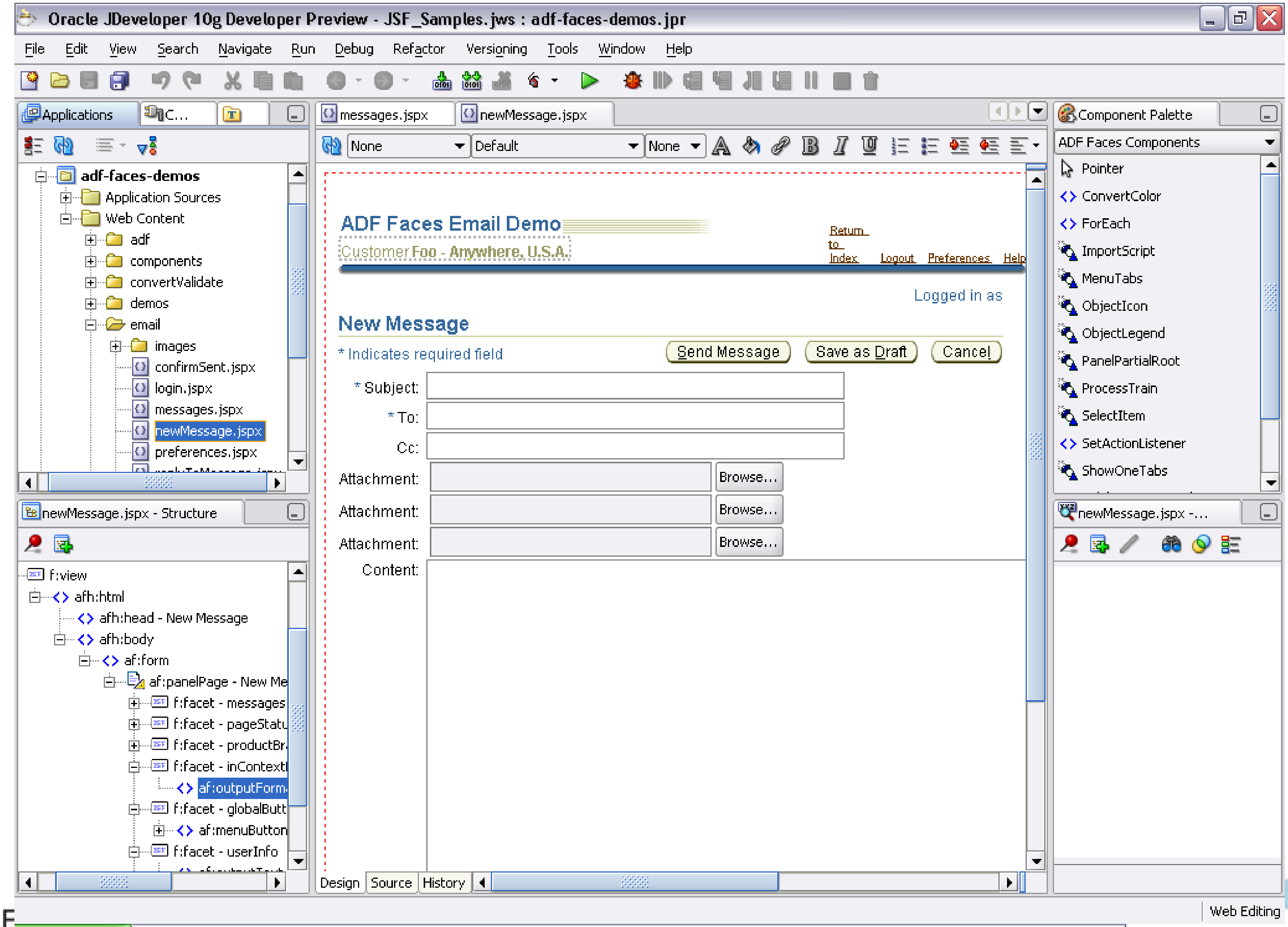
BEA Workshop Studio



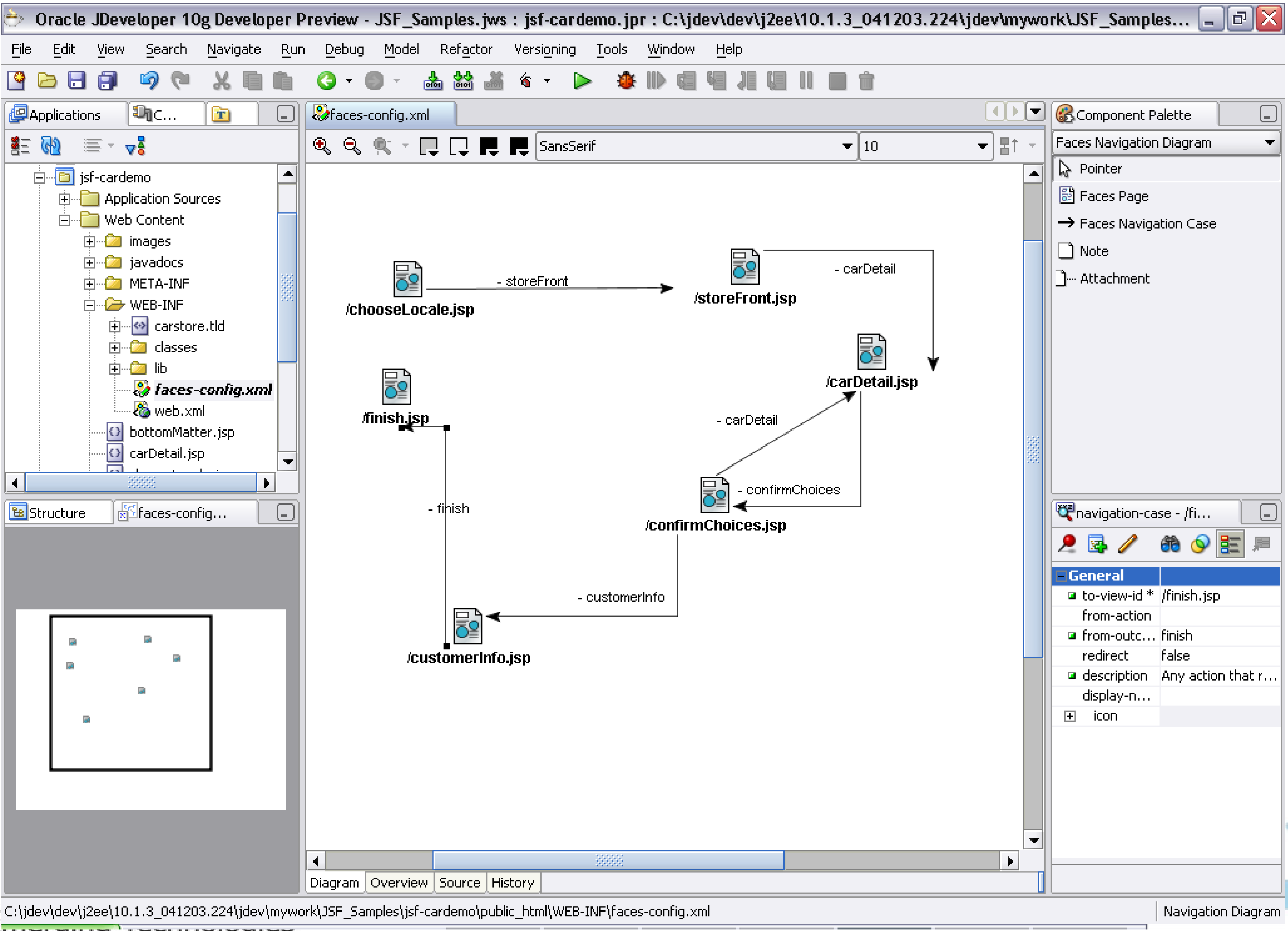
```
    <navigation-case>
      <description>
        Any action that returns "details" should go to the
        book details page.
      </description>
      <from-outcome>details</from-outcome>
      <to-view-id>/bookdetails.jsp</to-view-id>
    </navigation-case>

    <navigation-case>
      <description>
        Any action that returns "purchase" should go to the
        cashier page.
```

Oracle JDeveloper



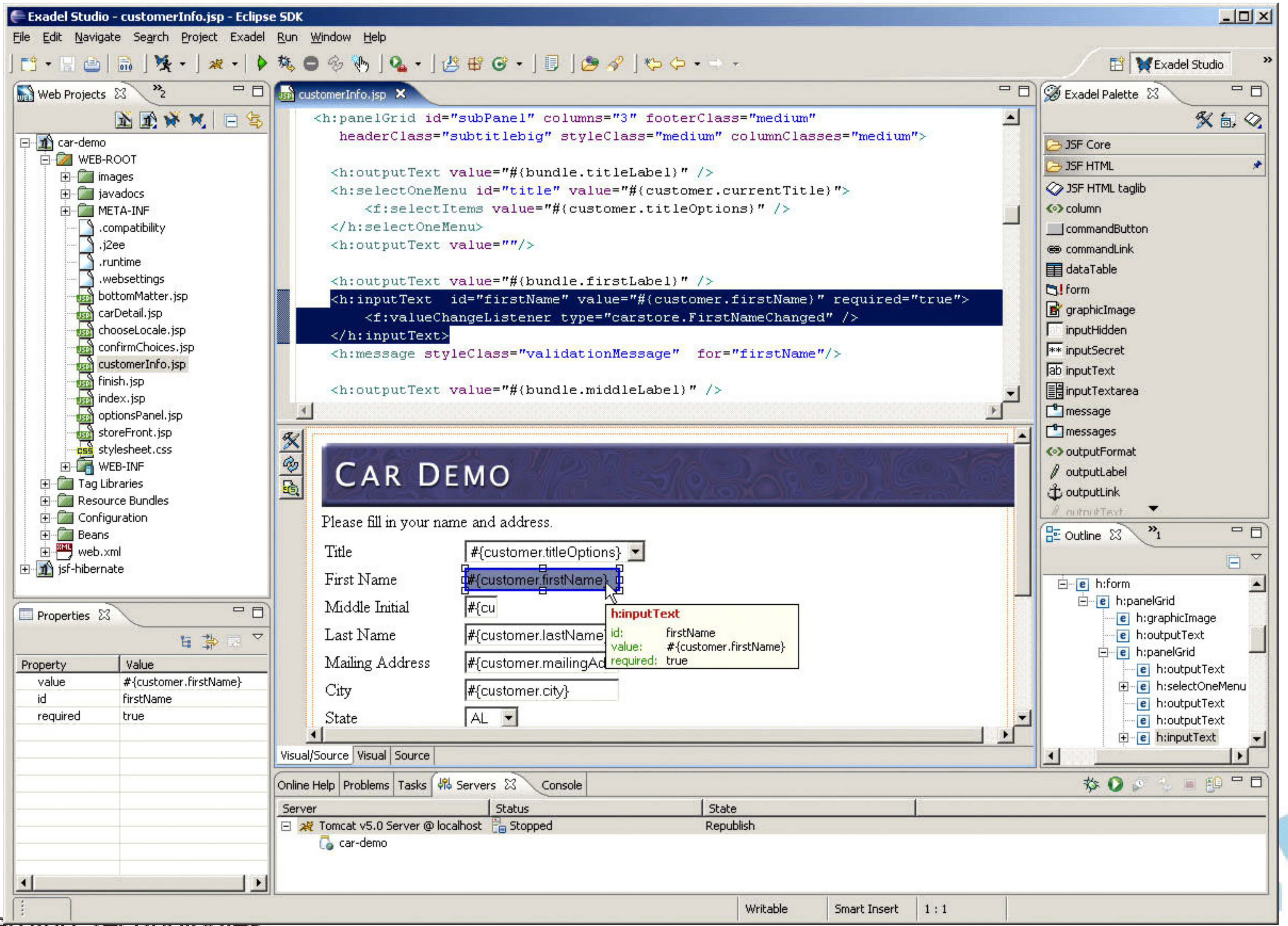
Oracle JDeveloper



General	
to-view-id *	/finish.jsp
from-action	
from-outc...	finish
redirect	false
description	Any action that r...
display-n...	
icon	

C:\jdev\dev\j2ee\10.1.3_041203.224\jdev\mywork\JSF_Samples\jsf-cardemo\public_html\WEB-INF\faces-config.xml

Exadel Studio Pro



```
<h:panelGrid id="subPanel" columns="3" footerClass="medium"
   headerClass="subtitlebig" styleClass="medium" columnClasses="medium">

  <h:outputText value="#{bundle.titleLabel}" />
  <h:selectOneMenu id="title" value="#{customer.currentTitle}">
      <f:selectItems value="#{customer.titleOptions}" />
  </h:selectOneMenu>
  <h:outputText value=""/>

  <h:outputText value="#{bundle.firstLabel}" />
  <h:inputText  id="firstName" value="#{customer.firstName}" required="true">
      <f:valueChangeListener type="carstore.FirstNameChanged" />
  </h:inputText>
  <h:message styleClass="validationMessage"  for="firstName"/>

  <h:outputText value="#{bundle.middleLabel}" />
```

Please fill in your name and address.

Title	#{customer.titleOptions}
First Name	#{customer.firstName}
Middle Initial	#{cu
Last Name	#{customer.lastName
Mailing Address	#{customer.mailingAd
City	#{customer.city}
State	AL

h:inputText

id:	firstName
value:	#{customer.firstName}
required:	true

Properties

Property	Value
value	#{customer.firstName}
id	firstName
required	true

Server	Status	State
Tomcat v5.0 Server @ localhost	Stopped	Republish
car-demo		

DEMO

Agenda

- JavaServer Faces from 1,000 feet
- Ajaxian Faces
- Can’t we all get along?
- Which one?
- JSF 1.2 and beyond
- Summary
- Q&A

Ajaxian Faces

- JSF’s architecture makes it easy to add Ajax support
  - Components and renderers are separate
  - PhaseListeners can be invoked during request processing
  - Enables *transparent* AJAX support
    - Same programming model with or without AJAX

Ajaxian Faces

- Three AJAX integration strategies:
  - Add AJAX support to existing components:
    - Ajax4jsf
    - AjaxAnywhere
    - DynaFaces

Ajaxian Faces

- Three AJAX integration strategies:
  - Integrate AJAX support into JSF components
    - Simplica ECruiser AJAX Suite for JSF
    - Infragistics NetAdvantage for JSF
    - ICEsoft ICEfaces
    - Exadel RichFaces Visual Component Platform
    - TeamDev QuipuKit
    - Backbase Java Edition
    - Sun Project Woodstock

Ajaxian Faces

- Three AJAX integration strategies:
  - Wrap existing AJAX widgets
    - jMaki
    - DojoFaces
- Full listing of products at http://www.jsfcentral.com/products/

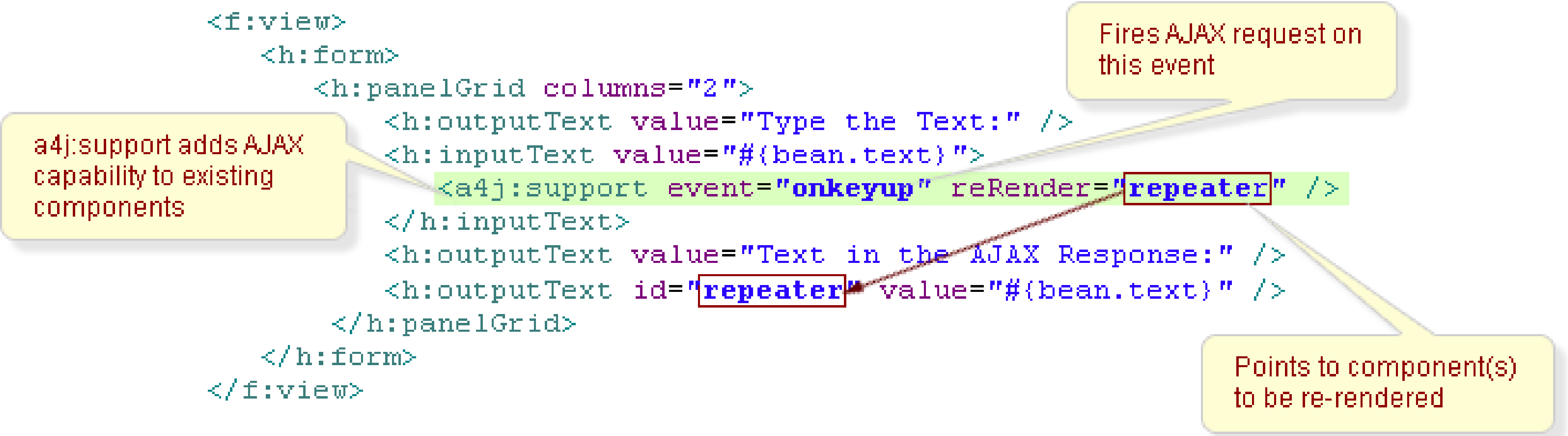
Ajax4Jsf

- Adds Ajax support to any JSF view
- Open source
  - Originally developed by Exadel
  - Now maintained by JBoss
- Features:
  - Can add Ajax support to any JSF component with JavaScript events
  - Can define regions of the page (or entire page) that can be updated via AJAX
  - Status component
  - Polling component

Ajax4Jsf

- Features:
  - MediaOutput component
  - ActionListeners and ValueChangeListeners invoked via Ajax
  - Also features AjaxListeners

Ajax4Jsf



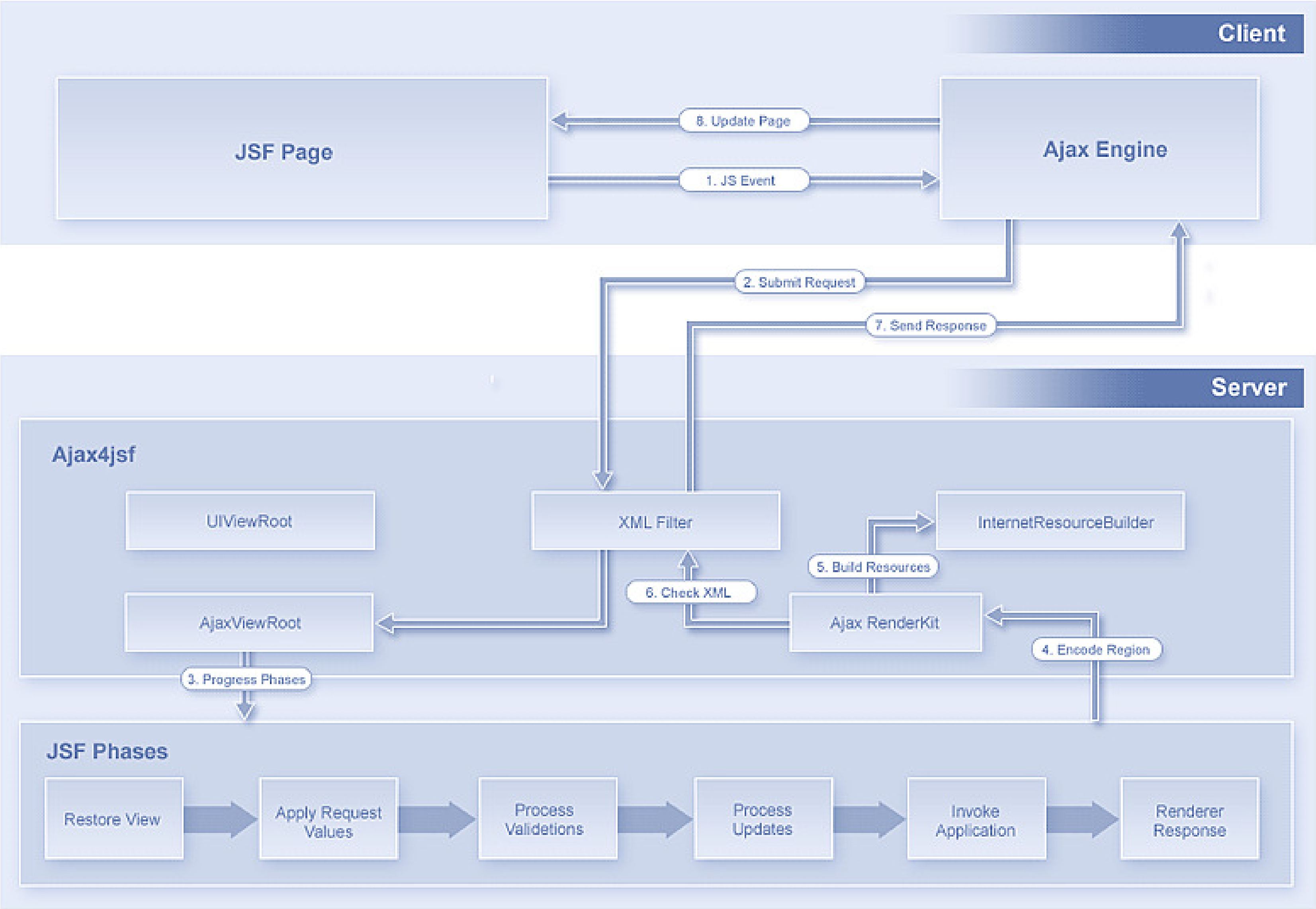
```
<f:view>
   <h:form>
      <h:panelGrid columns="2">
          <h:outputText value="Type the Text:" />
          <h:inputText value="#{bean.text}">
             <a4j:support event="onkeyup" reRender="repeater" />
          </h:inputText>
          <h:outputText value="Text in the AJAX Response:" />
          <h:outputText id="repeater" value="#{bean.text}" />
       </h:panelGrid>
   </h:form>
</f:view>
```

a4j:support adds AJAX capability to existing components

Fires AJAX request on this event

Points to component(s) to be re-rendered

Ajax4Jsf


Client
JSF Page
8. Update Page
1. JS Event
Ajax Engine
2. Submit Request
7. Send Response
Server
Ajax4jsf
UIViewRoot
XML Filter
InternetResourceBuilder
5. Build Resources
6. Check XML
AjaxViewRoot
Ajax RenderKit
4. Encode Region
3. Progress Phases
JSF Phases
Restore View
Apply Request Values
Process Validetions
Process Updates
Invoke Application
Renderer Response

DEMO

ECruiser AJAX Suite for JSF

- Suite of basic JSF components
  - DataTable, GridTable, Tree, Tab, Menu, etc.
- Integrated AJAX support
  - Lazy loading of data sets
  - Full set of AJAX-aware basic input components
  - Ability to execute any backing bean method for JavaScript events

DEMO

Infragistics NetAdvantage for JSF

- Suite of bread-and-butter components:
  - Tree, Table, Menu, Tabs, MenuBar, Input controls
- Full Ajax support
  - Dynamic loading of data sets
  - Updating part of the page with a selection
  - All components can send Ajax requests and update other components dynamically

DEMO

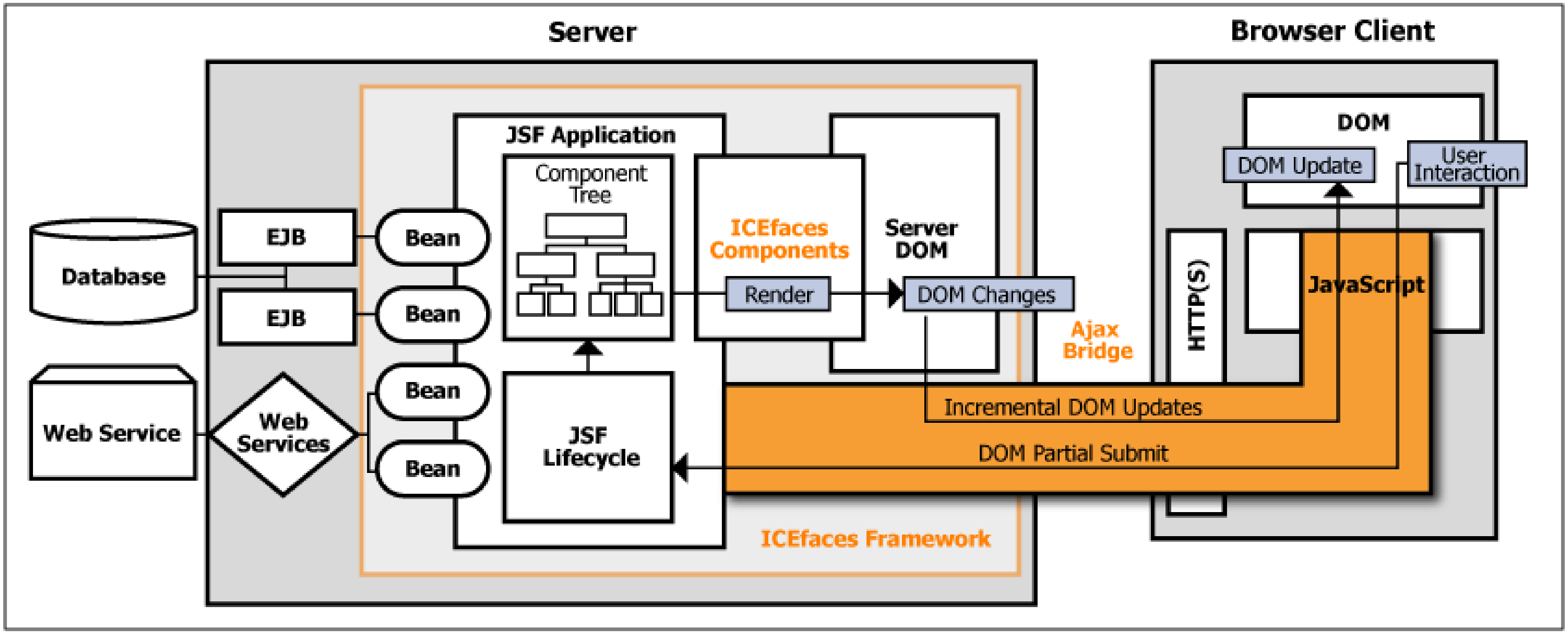
ICEsoft ICEfaces

- Innovative take on Ajax browser/server integration
  - Direct-to-DOM: maintains copy of DOM on server which is synchronized with DOM on the client
  - Requires a persistent connection between client and server
  - Supports Ajax Push

ICEsoft ICEfaces

- Three elements:
  - Framework
    - Implemented as a JSF RenderKit
  - Ajax bridge
  - Component suite
    - Chart, Data Table, DynamicImage, Collapsable Panel, Modal Popups, Tabs, TreeTable, Input controls

ICEsoft ICEfaces



DEMO

jMaki

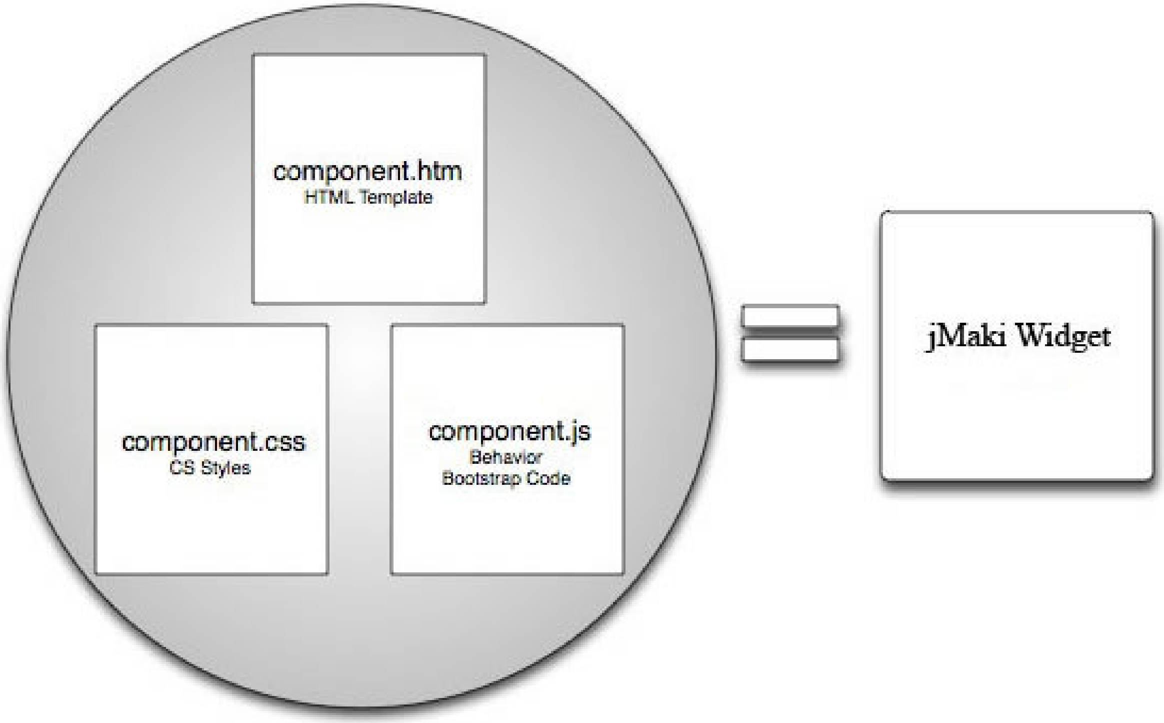
- Provides JSF and JSP wrappers for popular AJAX widgets
- Originally written by Greg Murray, Servlet 2.4 spec lead
  - Currently open-sourced on Java.net
- Supports widgets from:
  - Dojo, Scriptaculous, Yahoo!, Google, and others

jMaki

- Caveats
  - Requires JDK 1.5x or higher
  - Use without Glassfish not documented
  - JSF not supported for all widgets
  - Not as tightly integrated with JSF as other solutions

jMaki

- Easy to create new widgets



DEMO

Agenda

- JavaServer Faces from 1,000 feet
- Ajaxian Faces
- Can’t we all get along?
- Which one?
- JSF 1.2 and beyond
- Summary
- Q&A

Can’t we all get along?

- ICEfaces may not work with other components
  - Includes standard JSF components and ICEfaces custom components
  - Limited interoperability with other components
- Backbase doesn’t work with other components
  - Uses single-page model

Can’t we all get along?

- Ajax4jsf and DynaFaces duplicate functionality
  - Ajax4jsf is more mature
  - ICEsoft involved with DynaFaces

Agenda

- JavaServer Faces from 1,000 feet
- Ajaxian Faces
- Can’t we all get along?
- Which one?
- JSF 1.2 and beyond
- Summary
- Q&A

Which one?

- Pick a component suite
  - Tight integration with JSF
    - No incompatibility problems within the same suite
  - Well documented and supported
  - Choose as few as possible (preferably just one)
  - How much AJAX do you need?
    - Other suites (such as WebGalileo Faces and Apache MyFaces Tomahawk have some AJAX support as well)

Which one?

- To add support to existing JSF components
  - Use Ajax4Jsf
    - Works well with Apache MyFaces Tomahawk
- If you need AJAX push, choose ICEfaces
  - Well documented
  - Free version available
  - Enterprise addition provides support and enhanced scalability
- Use jMaki for eye candy or Web 2.0 components
- Don’t forget tools support!

Agenda

- JavaServer Faces from 1,000 feet
- Ajaxian Faces
- Can’t we all get along?
- Which one?
- JSF 1.2 and beyond
- Summary
- Q&A

JSF 1.2 and Beyond

- One problem with JSF 1.x Ajax support:
  - The request processing lifecycle
- JSF 1.2 provides specific Ajax support
  - JSF 1.2 Reference Implementation available now
    - Requires either a Java EE 5 server (with JSP 2.1) or Facelets (open source JSP replacement)
  - MyFaces working on JSF 1.2

JSF 1.2 and Beyond

- JSF 2.0 will incorporate features from:
  - DynaFaces
    - Single, integrated API for Ajax component developers
    - Will include JavaScript bindings
  - Apache Shale
  - JBoss Seam
  - Facelets

Agenda

- JavaServer Faces from 1,000 feet
- Ajaxian Faces
- Can’t we all get along?
- Which one?
- JSF 1.2 and beyond
- Summary
- Q&A

Summary

- JavaServer Faces is the standard Java EE web framework
- It is a component-based framework
- Provides a simple, cohesive programming model
- JSF's architecture lends itself to Ajax development

Summary

- Ajax development with JSF doesn’t require JavaScript!
  - Same programming model with or without Ajax
- Three different approaches to Ajax development with JSF
  - Integrate Ajax support into the components
  - Wrap regular JSF components with Ajax support
  - Wrap pure Ajax widgets with Ajax support

Summary

- Many different component suites and solutions exist currently
- JSF 1.2 makes Ajax integration easier for component developers
- JSF 2.0 will provide a single mechanism for Ajax integration for component vendors

Agenda

- JavaServer Faces from 1,000 feet
- Ajaxian Faces
- Can’t we all get along?
- Which one?
- JSF 1.2 and beyond
- Summary
- Q&A

Q&A

Resources

- Official JSF site
  - http://java.sun.com/javaee/javaserverfaces/
- JSF Central product directory
  - http://www.jsfcentral.com/products
- Ajax component suite comparison
  - http://www.theserverside.com/tt/articles/article.tss?l=JSFComparison
- Ajax-JSF Comparison Matrix
  - http://www.ajaxjsf.net/