

ORACLE®

Duncan Mills
Java Evangelist
Oracle Corporation

<http://www.groundsider.com/blog>

Java Server Faces

Everything you always wanted to know but were afraid to ask...

Agenda

- JSF Introduction
 - Why JSF
 - Architecture
 - JSF Technology
- JSF and Oracle
 - JDeveloper
 - ADF Faces
 - Future

Why JSF?

- So what's all the fuss about?
- What makes JSF so compelling?
- Why are IBM, Sun and Oracle devoting serious resources to JSF?
- Why another framework?

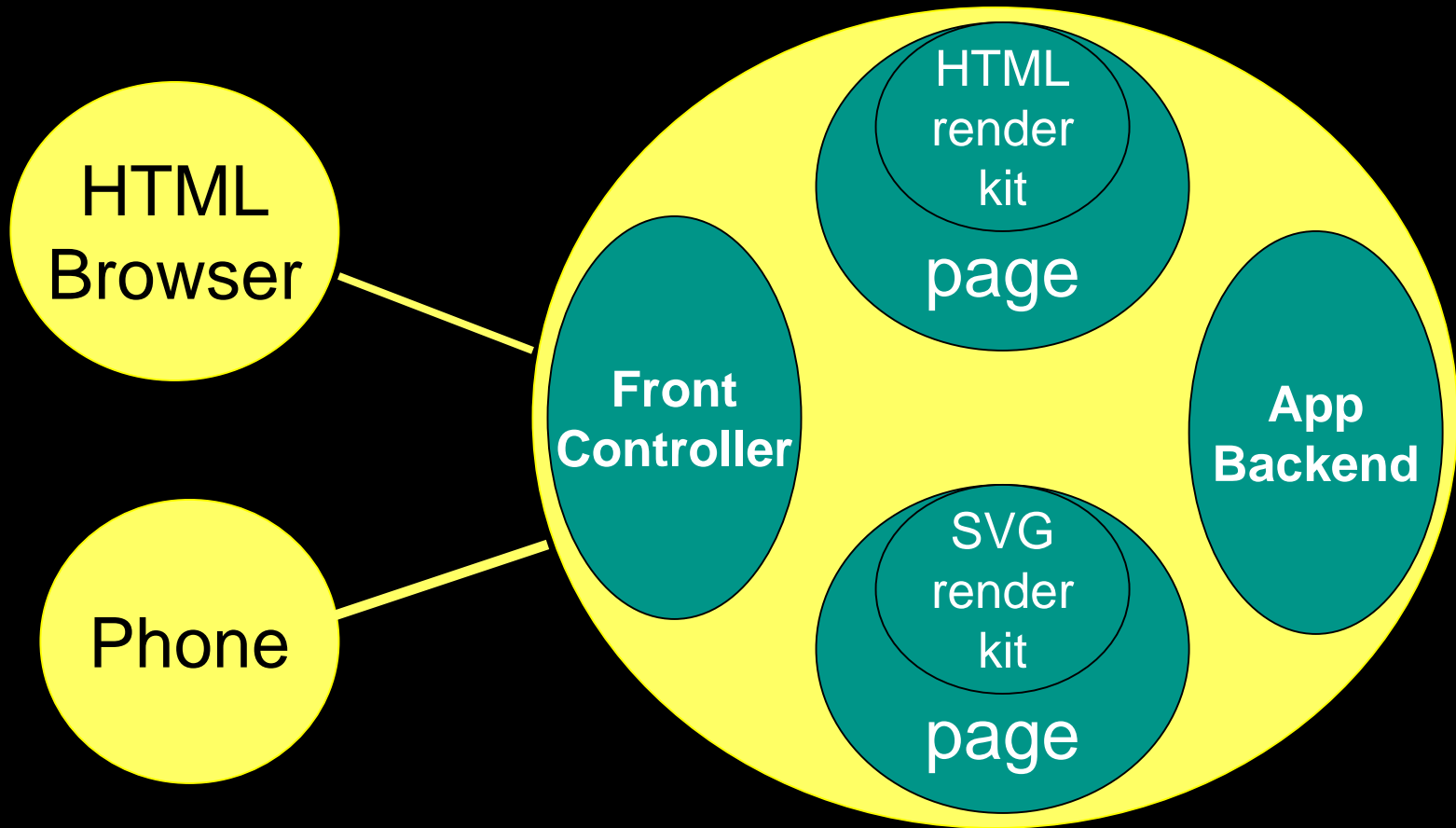
Here is why!

- JSF is a standard specification
 - JSR 127
 - Part of J2EE 1.5
- JSF exposes a component and event based programming model
- JSF is tool friendly
- JSF is implementation agnostic

JSF Architecture

- Abstracts away from HTTP
 - Developers don't need to understand request-response cycle
- Abstracts away from HTML
 - Developers work with components, not markup
 - Developers don't need to decode requests
- Abstracts away from JSP
 - Supports more than just one type of UI definition

Architecture



JSF Technology

Anatomy of a JSF application

- A standard J2EE 1.3+ Web application with...
- JSF and Commons jar files in WEB-INF/lib
- Faces controller servlet configured in the web.xml
- A Faces-config.xml file

JSF Technology

The Faces-config.xml file

- Similar to struts-config.xml
- Contains two main types of components
 - Managed Beans
 - Navigation Model

JSF Technology

A typical JSF application consists of:

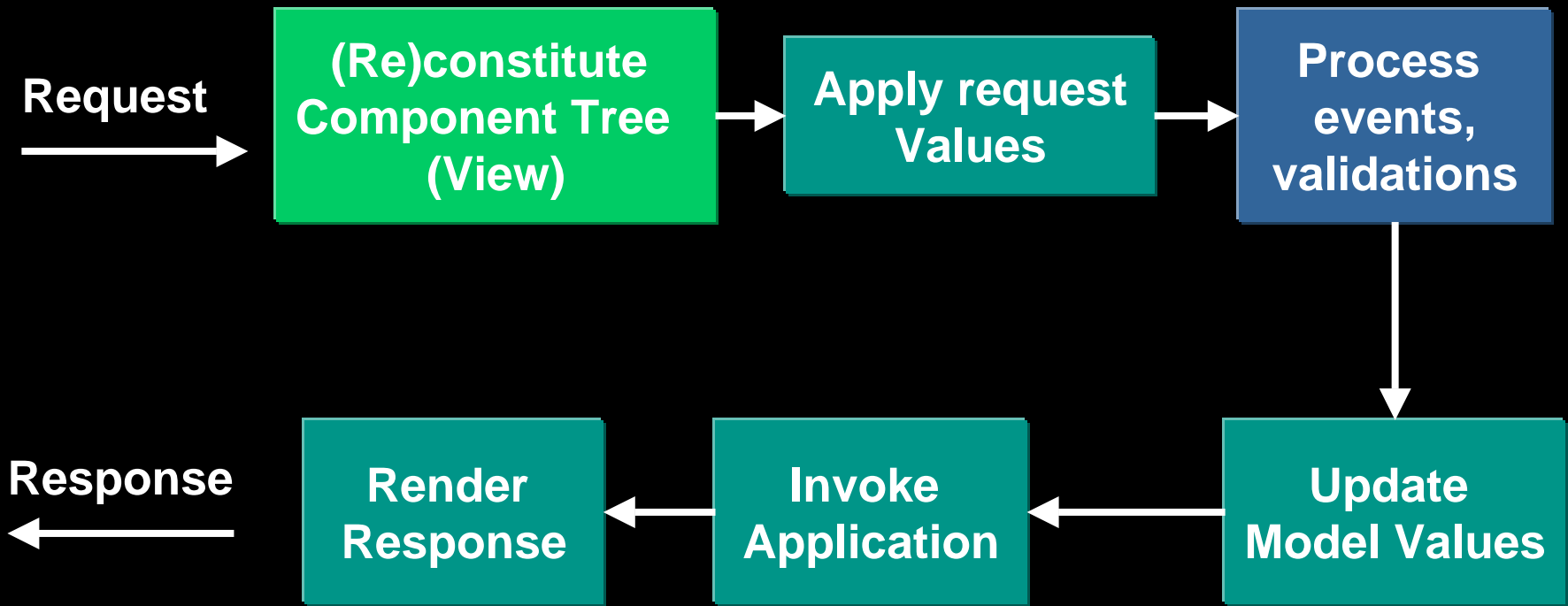
- JSP pages containing UI Components encapsulated in JSP tag libraries: Core, HTML
- A Navigation Model specified in the faces-config.xml
- A set of Managed Beans which facilitate the UI logic of the application
 - Kind of like Struts Form beans and Actions combined

Lifecycle of JSF page

- A JSF page is represented by a tree of UI components, called a view
- When a client makes a request for the page, the lifecycle starts
- During the lifecycle, JSF implementation must build the view while considering state saved from the previous postback
- When the client performs a postback of the page, JSF implementation must perform lifecycle steps
 - Validation
 - Conversion

JSF Technology

The JSF Lifecycle



Request Processing

- Life-cycle handles two types of requests
 - Initial request & Postback
- Initial request
 - A user requests the page for the first time
 - Lifecycle only executes the restore view and render response phases
- Postback
 - A user submits the form contained on a page that was previously loaded into the browser as a result of executing an initial request
 - Lifecycle executes all phases

JSF Technology

JSF UI Components

- Basic building blocks of a JSF application
- Can represent simple to complex User Interface components ranging from a button or input field to a complete page.
- Can be associated to Model data objects through Value Binding
- UI Components use helper objects: validators, converters, listeners/events

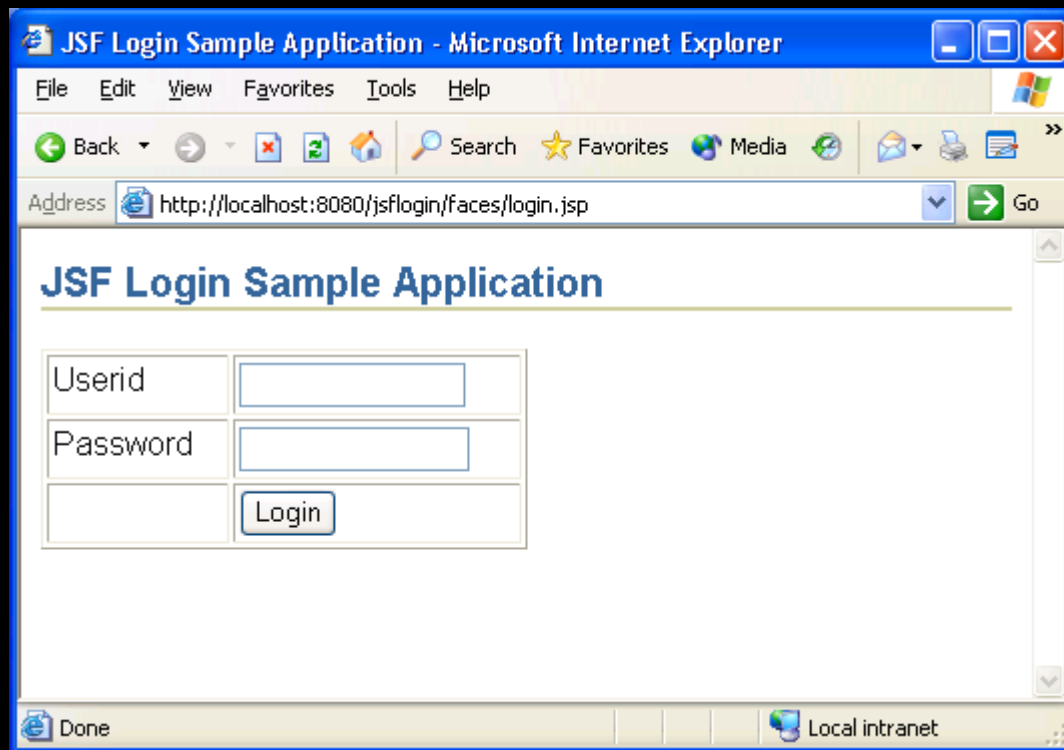
JSF Technology

JSF UI Components

- UI Components can be rendered in different forms based on invocation:
 - UIOutput -> text label, date label..
- UI Components can be invoked directly or from JSP using JSF tag libraries
 - Example: UIInput can be invoked in a JSP with `<h:inputText ..>`
- UI Components are extendable
 - Developers can create new, custom components

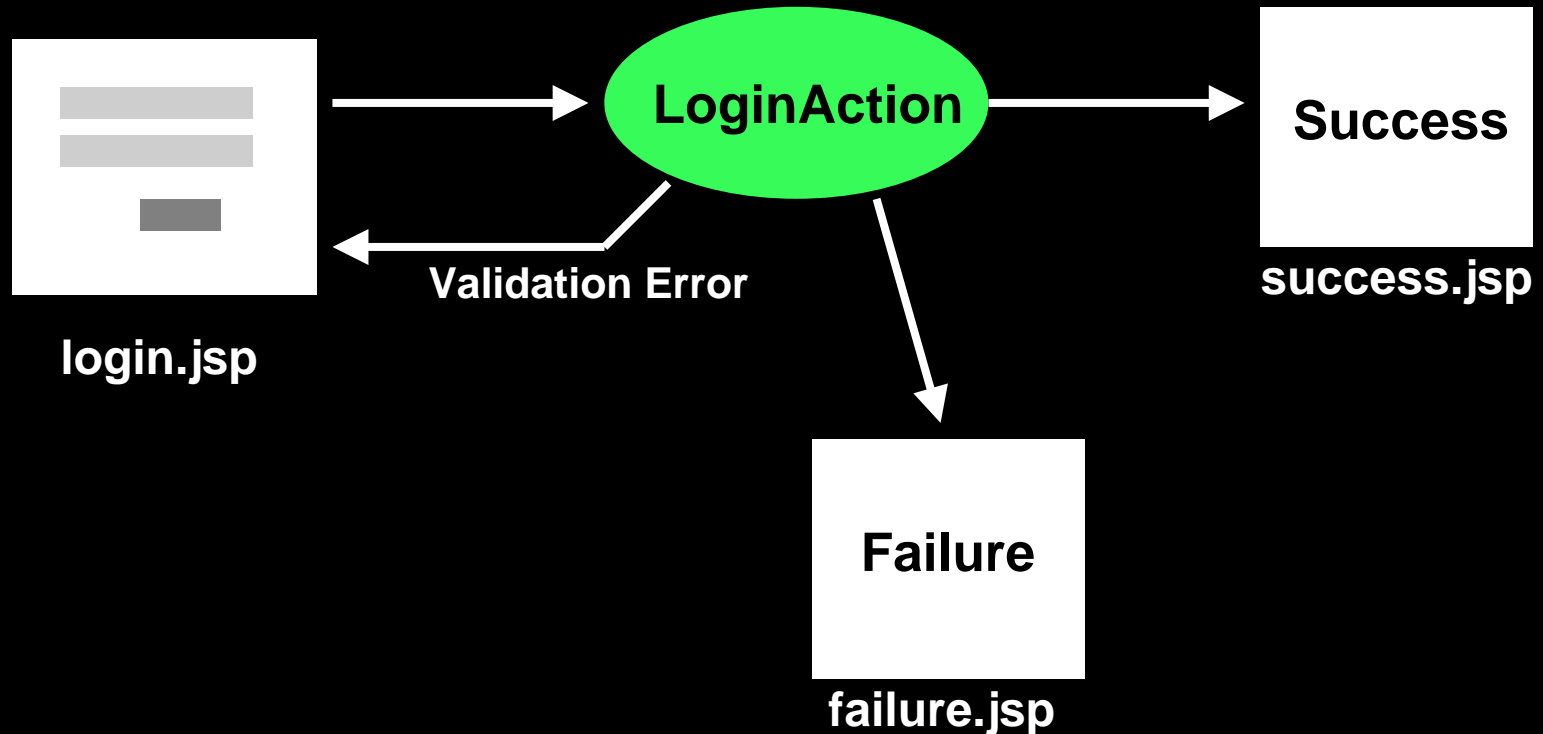
JSF Technology

A simple example – A JSF Login Application



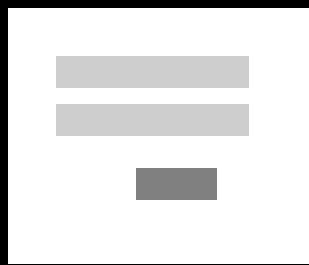
JSF Technology

A simple example – A JSF Login Application



JSF Technology

UI Components and Value Binding to a Managed Bean



login.jsp

`<h:InputText value=#{Login.userid} />`

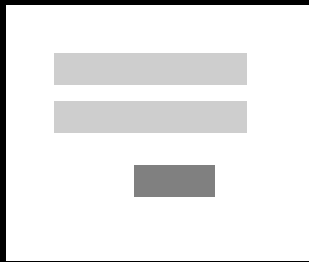
`<h:InputSecret value=#{Login.password} />`



Login – A “managed bean” with fields:
userid, password

JSF Technology

How to use a built-in Validation method

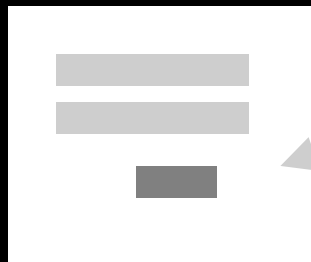


login.jsp

```
<h:InputText id="userid"
  value="#{Login.userid}" >
  <f:validateLength minimum="4" />
</inputText>
<h:messages for="userid"/>
```

JSF Technology

A Command UI Component bound to an “action”



login.jsp

```
<h:commandButton action="#{Login.loginAction}"/>
```

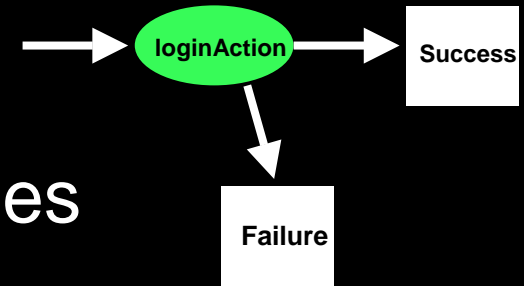


Login – A “managed bean” with action method: loginAction

Returns String: e.g. “failure” or “success”

JSF Technology

Login.jsp navigation rule with 2 cases



```
<navigation-rule>
```

```
<from-view-id>login.jsp</from-view-id>
```

```
<navigation-case>
```

```
<from-action>#{LoginServer.loginAction}</from-action>
```

```
<from-outcome>success</from-outcome>
```

```
<to-view-id>/success.jsp</to-view-id>
```

```
</navigation-case>
```

```
<navigation-case>
```

```
...failure...
```

```
</navigation-case>
```

```
</navigation-rule>
```

JSF Technology

Localizing the login app..

Faces-config.xml

```
<application>
```

```
  <message-bundle>path-to.Resources</message-bundle>
```

```
  <locale-config>
```

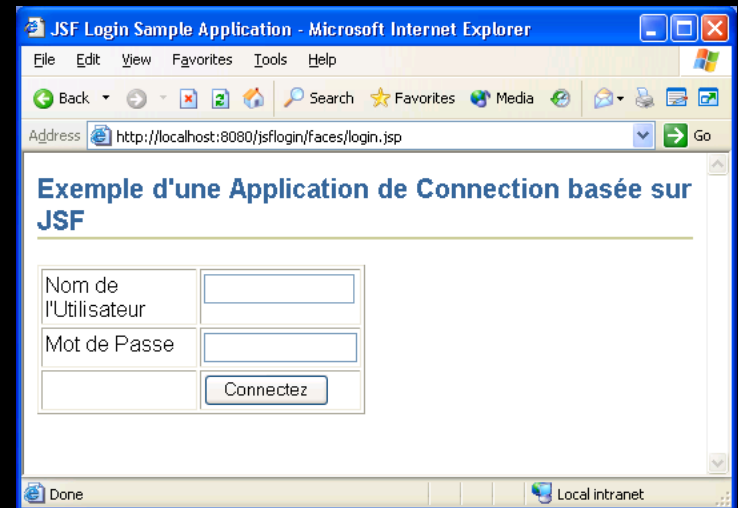
```
    <default-locale>en</default-locale>
```

```
    <supported-locale>fr</supported-locale>
```

```
    <supported-locale>es</supported-locale>
```

```
  </locale-config>
```

```
</application>
```



JSF Technology

Localizing the login app..

In your page

```
<f:loadBundle basename="jsflogin.Resources"  
var="jsfloginBundle"/>
```

```
<h:outputText value="#{jsfloginBundle.useridLabel}"
```


JSF Technology

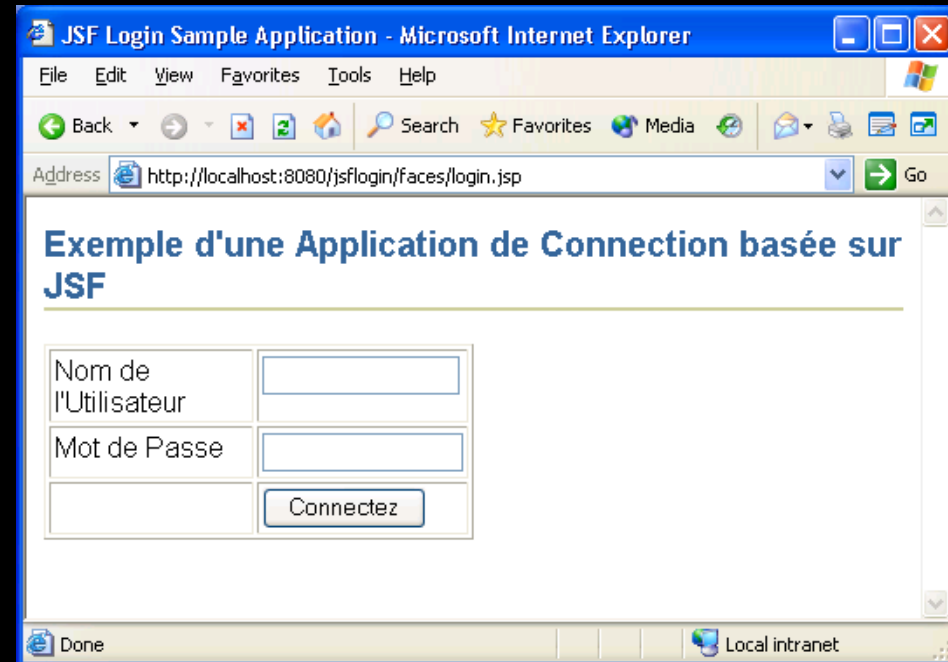
Localizing the login app..

Resources.Properties

useridLabel=Userid

Resources_fr.Properties

useridLabel=Nom de l'Utilisateur



Agenda

- JSF Introduction
 - Why JSF
 - Architecture
 - JSF Technology
- JSF and Oracle
 - JDeveloper
 - ADF Faces
 - Rich Web Clients

JSF and JDeveloper 10.1.2

- Need to manually integrate JSF RI
 - 5 Easy steps:
 1. Create new JDeveloper library for JSF jars.
 2. Add JSF tag libraries to JDeveloper (including Component Palette)
 3. Create a new Web project and configure web.xml for faces servlet.
 4. Add Faces jar files to Web-Inf/lib (classpath)
 5. Add empty faces-config.xml Web-Inf

http://otn.oracle.com/products/jdev/howtos/10g/jsf_howto/jsf.html

ADF Faces and JDeveloper 10.1.3

- Download the extension from the Update center
 - Help >> Check For Updates
- Regular Updates as new versions released
 - EA13 is current

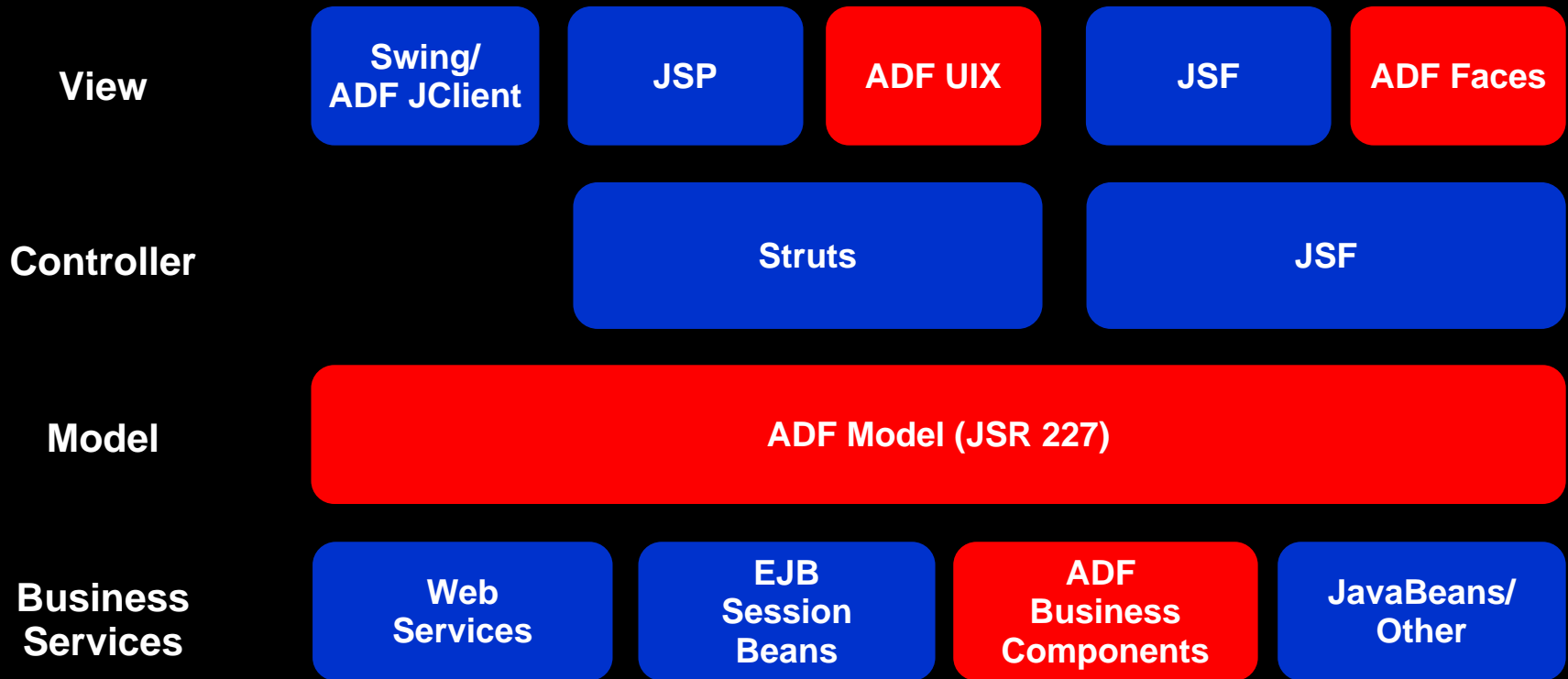
JSF and JDeveloper 10.1.3

- JSF Visual Editing
 - JSF UI Component visual editing
 - Provides backend code generation (double-click)
- Faces Config Editing Console
 - Console for productive editing of faces-config.xml
- JSF Page Flow
 - Visually design Faces navigation model

What Is Oracle ADF?

- Application Development Framework (ADF)
- A runtime framework for J2EE applications
 - Thin web clients (html based) & desktop applications
 - Based on MVC architecture
- Made up of several layers
 - Business Services Layer – EJB, ADF BC, WebServices etc.
 - ADF Model (JSR 227)
 - Struts Controller
 - JSP/JSF, ADF UIX, Swing and ADF Faces
- Supported by JDeveloper 10g

Oracle ADF Architecture



ADF Faces

- Evolution of ADF UIX
- Built on top of JSF APIs
- Deployable on any compliant implementation of JSF
- Provides much of what is missing in JSF 1.1
 - The component set is small
 - No client-side behavior
 - No support for customization
 - Little direct support for working with databases
 - Tools will of course support JSF – but how well?

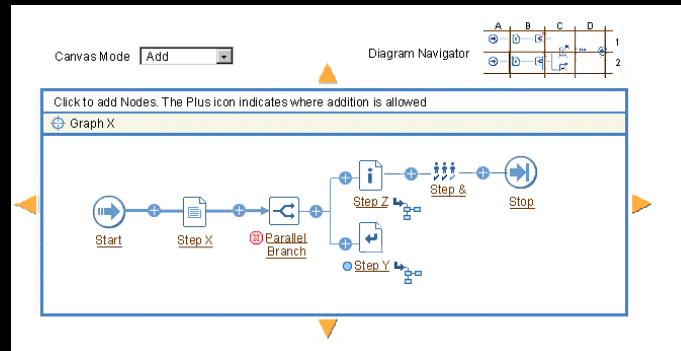
What ADF Faces brings to JSF

- Much larger component set
 - Over 100 different components
- Far more advanced and interesting components
 - Partial-page rendering, etc.
- Rich feature set for customizing applications
- ADF model support out-of-the-box
- ADF Faces skins (Look and Feel)
- ADF Databinding (JSR 227)

ADF Faces Components Examples

Show All Details | Hide All Details Previous 1-3 of 6 Next 3

Details	Name	Age
Hide	Sally	24
Detailed Information		
Name	Sally	
Age	24	
Title	Doctor	
Patients	30	
Hospital	UCSF	
Show	Willy	25
Show	Kelly	28



136-255-136

Root > Computer > Hardware > CDROM

Product Name that spans 2 lines

Home Shopping Cart Order History

Processing: Baking the cake

Using processingLayout element. Cancel

In this demo we use the processingLayout element. It is intended to make it easier to create the BLAF Processing Template.

- ✓ All the ingredients have been mixed.
- ✓ The oven has been preheated to 400 degrees.
- ✓ The pan has been greased and has the batter in it.
- ✓ The pan is in the oven.

The cake is almost done.

Cancel

- ▶ [UIX Components](#)
- ▼ **UIX Controller**
 - ◊ [Architecture](#)
 - ◊ [Release](#)
- ▶ [Architecture](#)
- ▶ [Release](#)



? ? Finance Go

Home Help Switch Applications



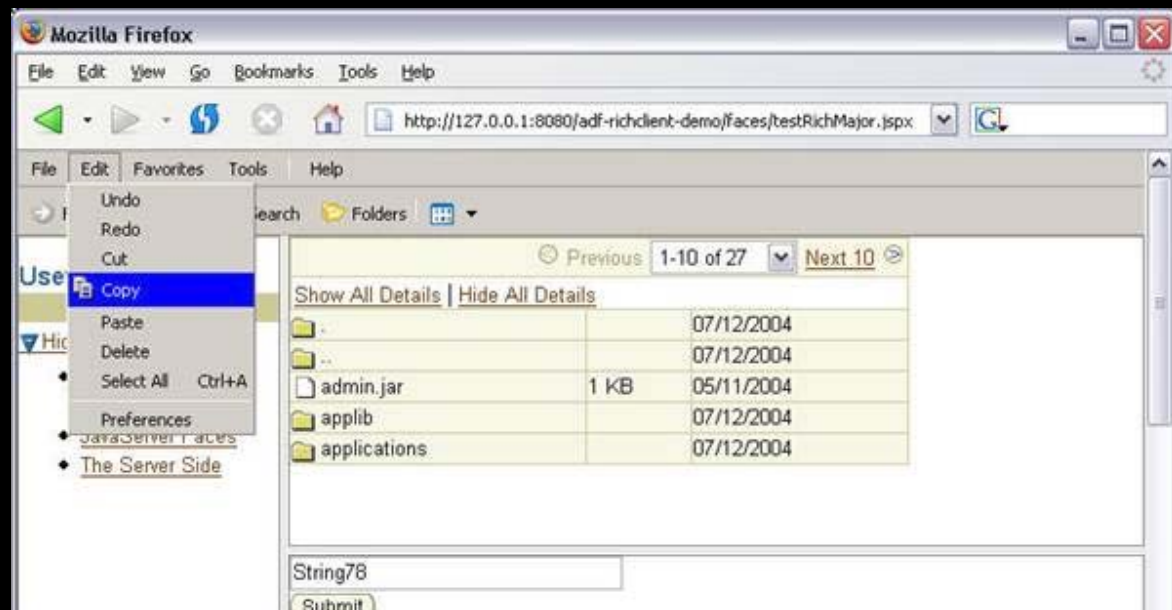
Look And Feel Customization

- ADF Faces will provide two “as is” look and feels:
 - Oracle’s Browser Look and Feel (BLAF)
 - Minimal Look And Feel (MLAF)
- Planned support for declarative L&F definition



ADF Faces.*NEXT*

- Rich Web Client components
- Extending ADF Faces
- JSF components with live updates in the Browser
- Popup menus, Trees, Splitters etc...



ORACLE®

D E M O N S T R A T I O N

Building User interfaces with Oracle ADF Faces

ORACLE®

Summary

- JSF is poised to be one of the standard ways to build web apps
- Looking towards rich clients, JSF component based paradigm is a good fit
- Look forward to better and better IDE support

A large, stylized logo in the background consisting of a grey 'Q', a red ampersand '&', and a grey 'A'. The text 'QUESTIONS' and 'ANSWERS' is overlaid on this logo.

QUESTIONS
ANSWERS

ORACLE®