



JavaOne™

ORACLE®

What's Next for JSF? BOF4241



Edward Burns
Manfred Riem
Java Community Process Specification Leads
Java EE Platform group



Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

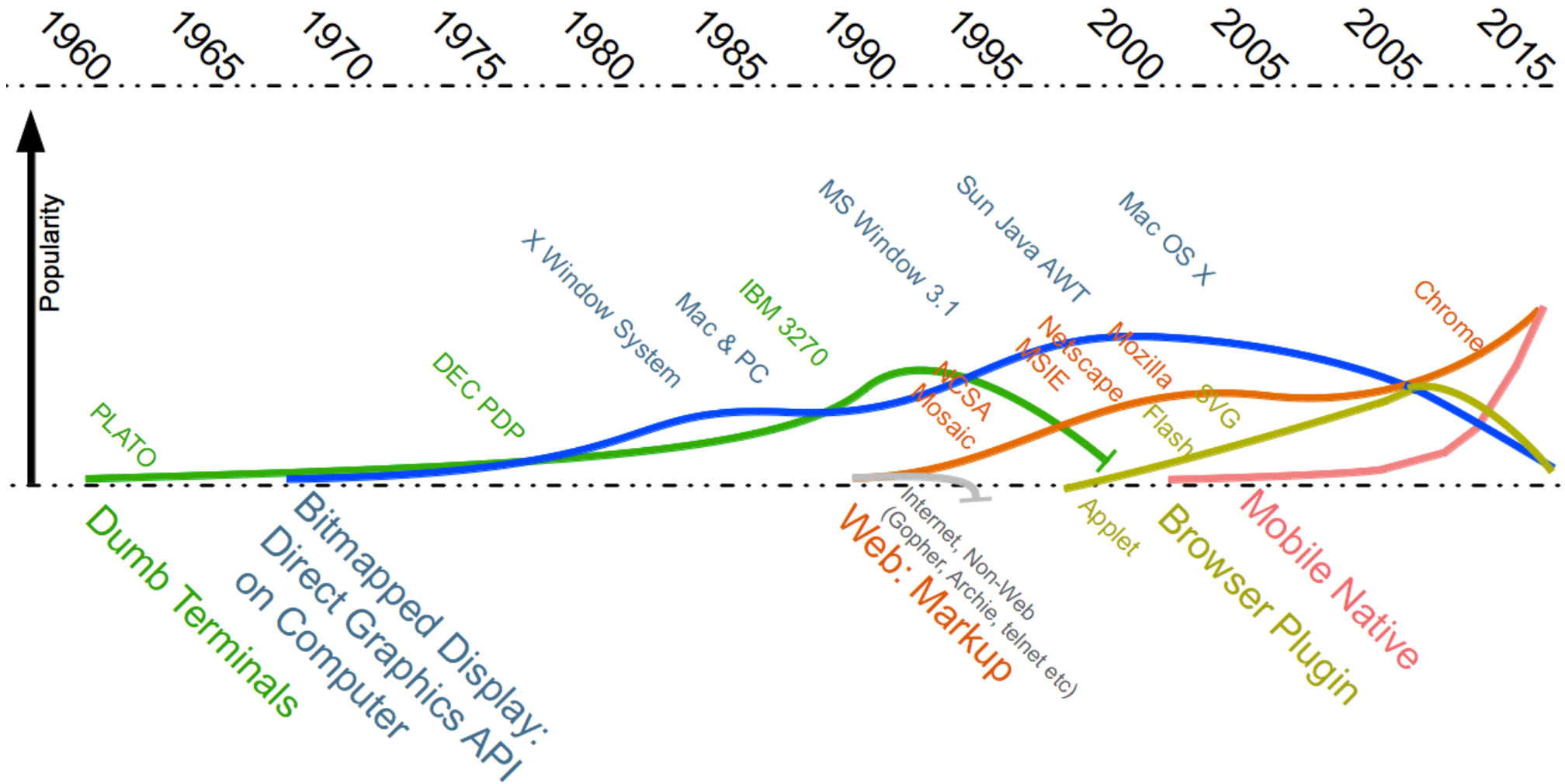
My Plan for Your Time Investment

- 1 ➤ UI trends
- 2 ➤ What does this mean for JSF?
- 3 ➤ What about Java EE 8?
- 4 ➤ Preserving your investment in JSF
- 5 ➤ Q&A

My Plan for Your Time Investment

- 1 UI trends
- 2 What does this mean for JSF?
- 3 What about Java EE 8?
- 4 Preserving your investment in JSF
- 5 Q&A

Distributed App UI History



Mobile First Web Design

- Mobile is now more important than desktop
- What kind of app?
 - Mobile Web
 - Native App (Android, iOS, Windows)
- Lots of help out there
 - twitter bootstrap and its enablers



DEPARTMENT OF STATE



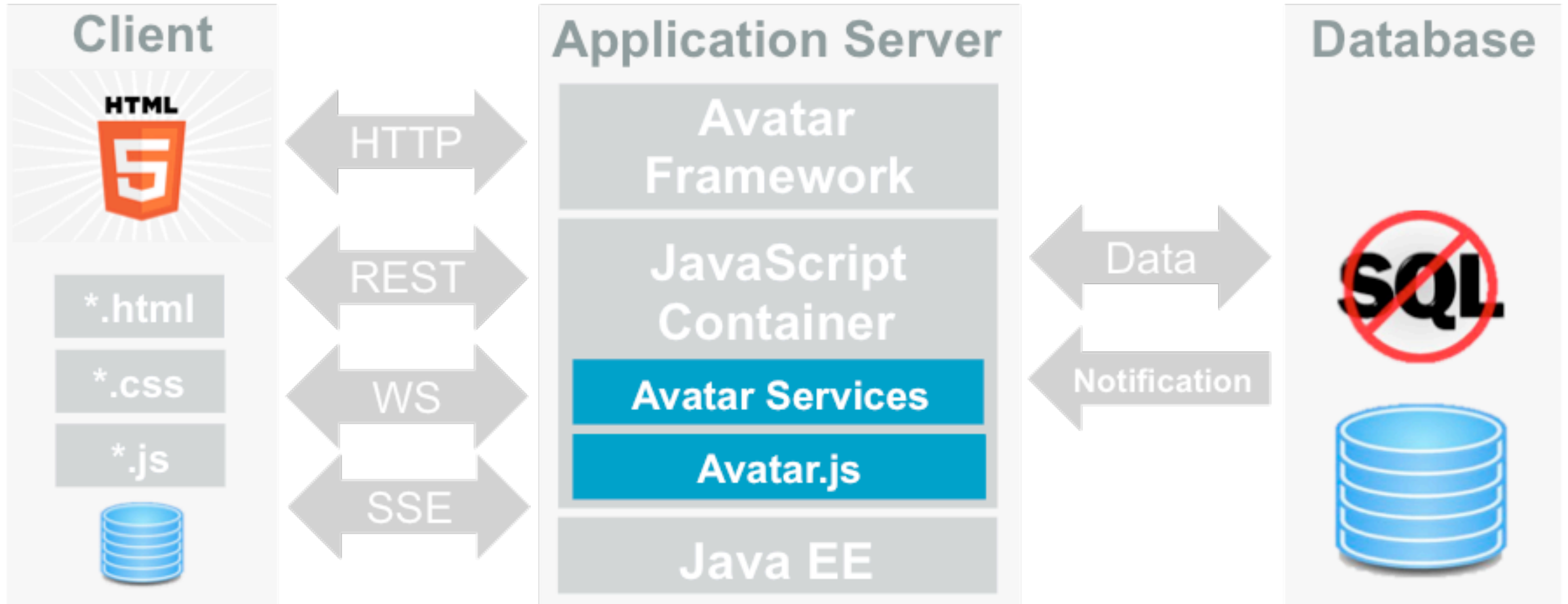
All UI State in client

All UI State on server

MEAN

- MongoDB
- ExpressJS
- AngularJS
- Node.js

Oracle's Answer: Project Avatar

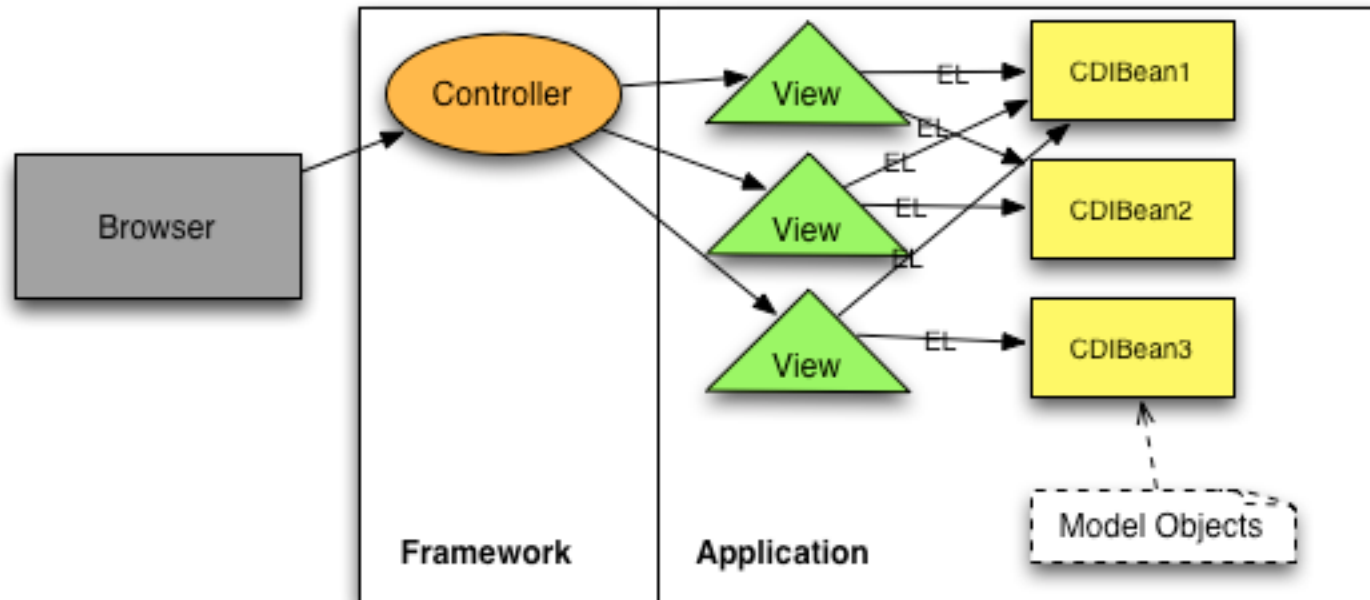


What Does All This Mean for JSF?

- Mobile
 - Use a mobile optimized JSF component library
- State
 - Design for statelessness
- JavaScript
 - Still held at arm's length
 - A component level concern

What Does All This Mean for JSF?

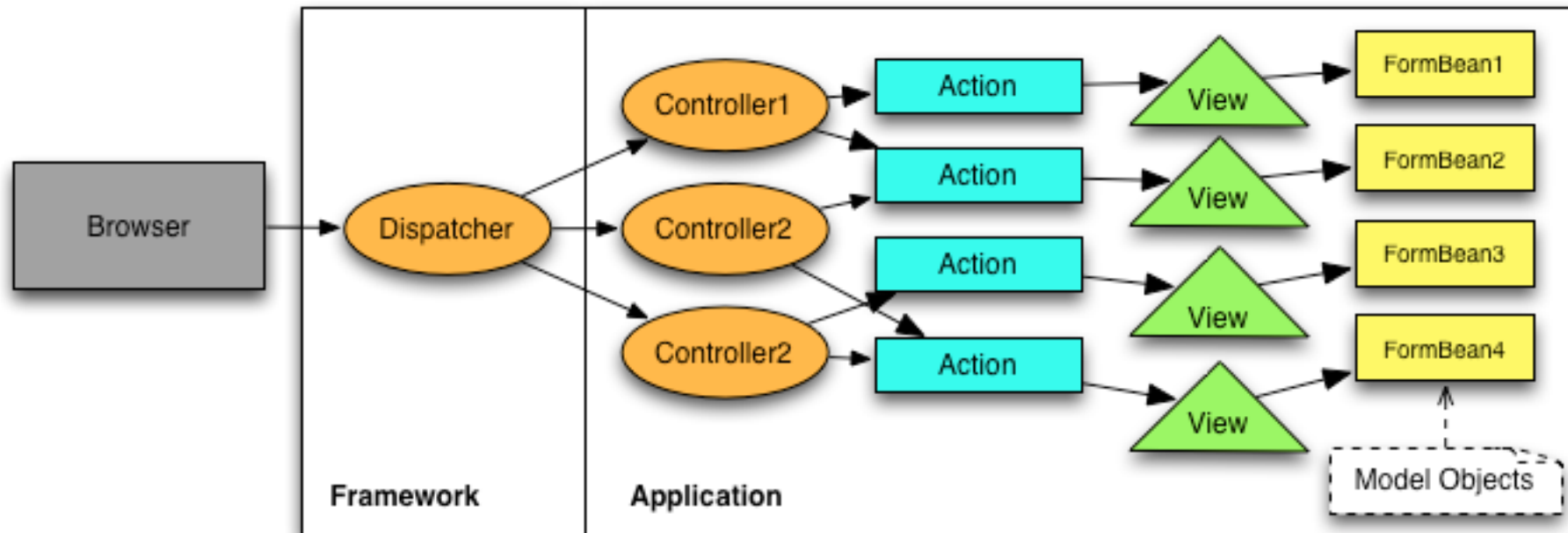
What about the MVC JSR?



UI Component Oriented MVC

What Does All This Mean for JSF?

What about the MVC JSR?



Action Oriented MVC

What About the MVC JSR?

Alignment between JSF and MVC

- Facelets
- Scopes
 - Flow
 - Flash
 - View Scope

JSF 2.3 in Java EE 8

Two feature drivers, each with their own agenda



ORACLE

- Tie up the loose ends
 - Specification clarifications
 - CDI alignment
 - Ease of use e.g. `@Inject FacesContext`
 - `@Inject` into Validators, Converters, etc.
 - `HtmlInputHidden` is `ClientBehaviorHolder`
- Small scale new features



- Subject to Spec Lead oversight
 - JSON ajax component rendering
 - Stateless enhancements
 - GET enhancements
 - Adopt-a-JSR support

JSF 2.3 Small Scale New Features

- End of the line for non-CDI managed beans
 - Redefine meaning of `@javax.faces.bean.ManagedBean` to be CDI bean
 - Redefine corresponding XML syntax to be CDI bean
- Ajax method invocation
 - A Faces request
 - Not specific to a view
 - Is specific to a ClientWindow
 - Returns results in JSON
- Multi-component validation



Community Corner

JSF Expert Group Members



- Ian Hlavats, Tarantula Consulting
- Kito Mann, Virtua



Preserving Your Investment In JSF

JSF 2.2 New Feature Recap

JSF 2.2 Big Ticket Features

- Markup Evolution
 - 1090 HTML5 Friendly Markup
- Multi-tenancy and Modularity
 - 730 Flows
 - 1142 Resource Library Contracts

1090 HTML5 Friendly Markup

The best part of Wicket comes to JSF

- This is a JSF page

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:myNS="http://xmlns.jcp.org/jsf">
<form myNS:id="form">
  <input name="textField" type="text" myNS:value="#{bean.text1}" />
  <input type="submit" myNS:id="submitButton" value="submit" />
  <p>submitted text: #{bean.text1}.</p>
</form>
</html>
```

1090 HTML5 Friendly Markup

Let the elegance of HTML shine through

- Before JSF 2.2
 - JSF tags hide complexity of underlying HTML+script+css+images
 - JSF “Renderer”:
 - encode: markup to browser
 - decode: name=value from browser

`<html>...`

```
<my:colorPicker value="#{colorBean.color2}" />
```

```
<my:calendar value="#{calendarBean.date1}" />
```

`</html>`

- Context: Missing feature in browser? Write a JSF component.

1090 HTML5 Friendly Markup

Let the elegance of HTML shine through

- With JSF 2.2
 - Pure HTML+script+css+images in the JSF page
 - JSF Renderer handles decode from browser
 - Leverage the strength of the JSF lifecycle
 - Leverage the expressiveness of HTML5

```
<html>...
```

```
<input type="color" jsf:value="#{colorBean.color2}"/>
```

```
<input type="date" jsf:value="#{calendarBean.date1}"/>
```

```
</html>
```

- Context: New feature in browser? Use “pass through elements”

1142 Resource Library Contracts

730 Faces Flows

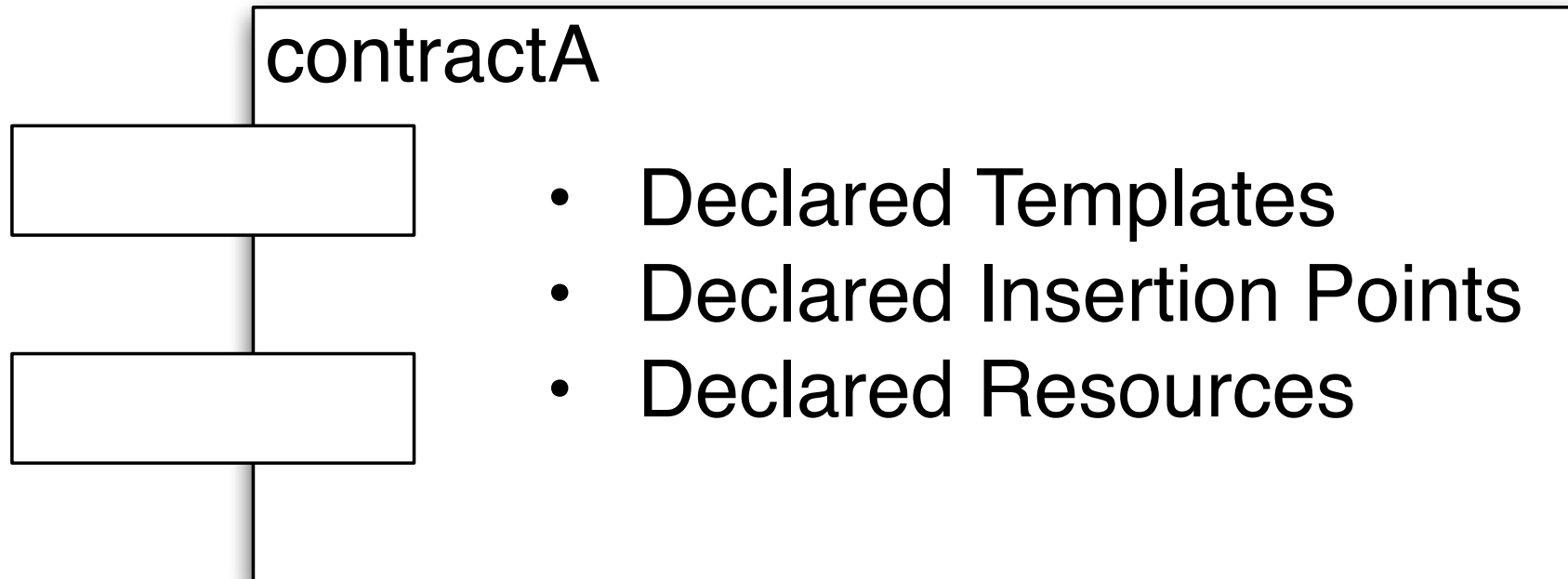
Modularity and Multi-tenant capability

- Allow composing a JSF app as a collection of modules
 - Faces Flows modularize behavior
 - Resource Library Contracts modularize appearance
- Well defined contract for each



Resource Library Contracts

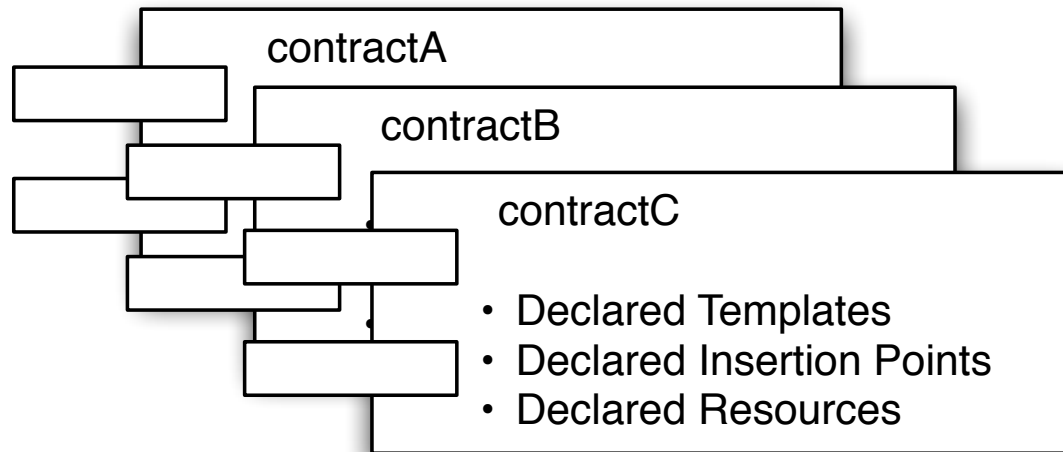
A Contract is Born



Resource Library Contracts

Loading Conventions

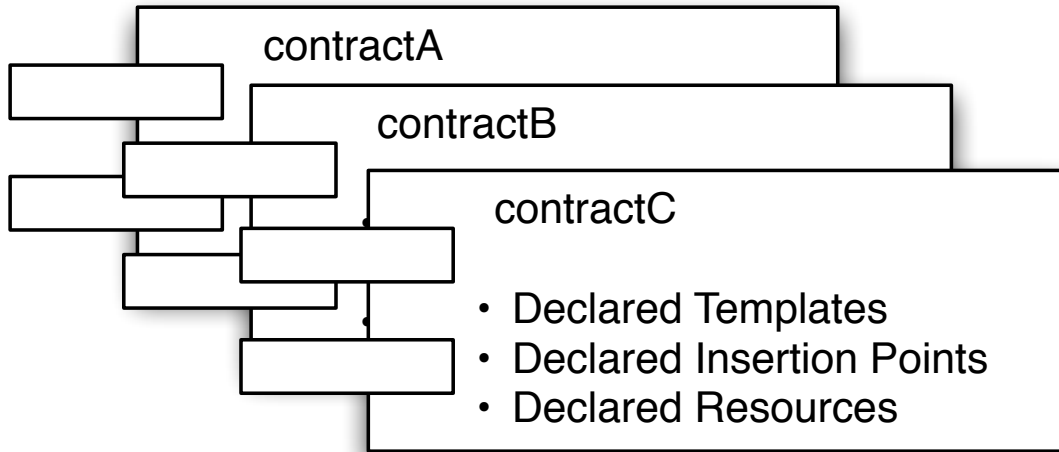
`<web-app-root>/contracts`



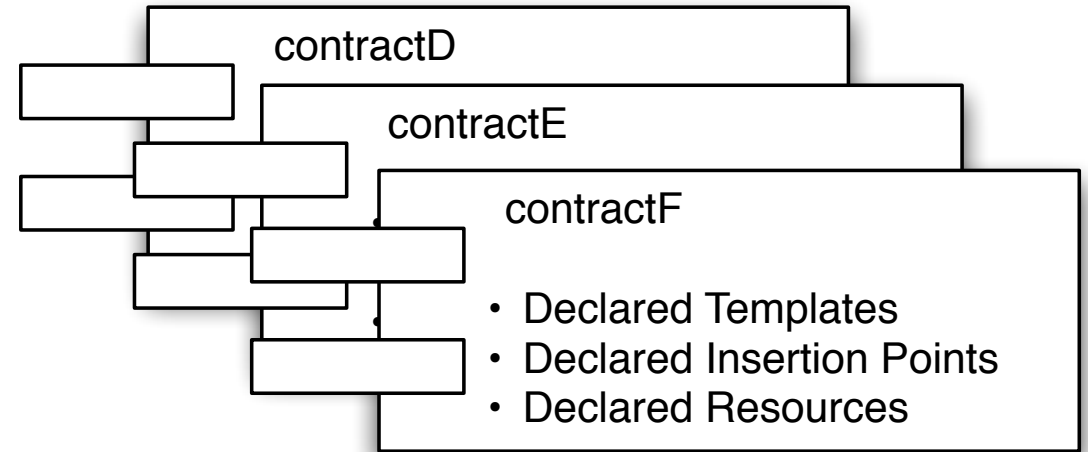
Resource Library Contracts

Loading Conventions

<web-app-root>/contracts

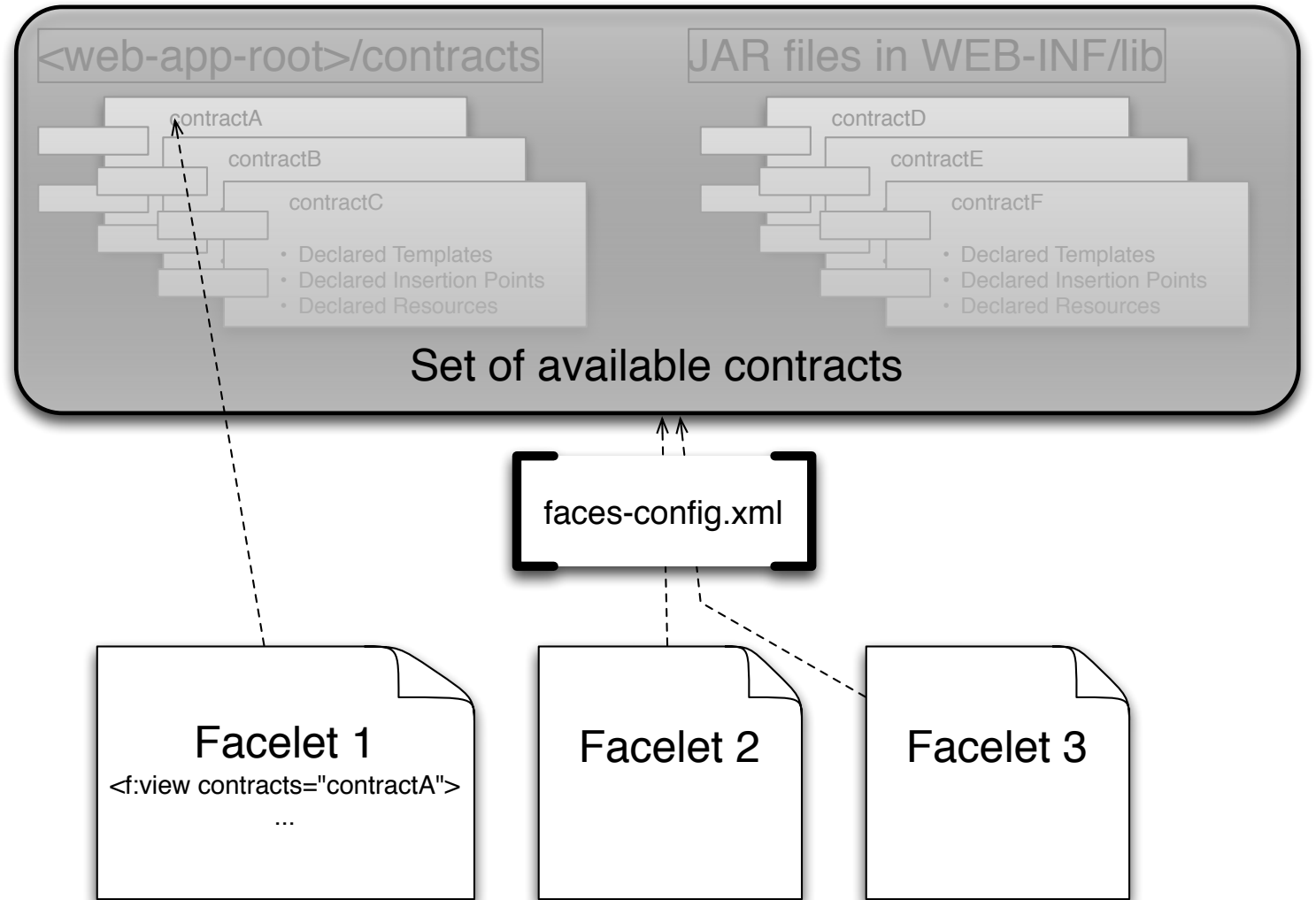


JAR files in WEB-INF/lib



Resource Library Contracts

Loading Configuration



Flow Navigation

- Navigation is no longer just between pages
- Navigation is now between flow “nodes”
- Information Hiding comes to JSF. Welcome to 1972!
- Multiple node types:
 - View
 - Method Call
 - Switch
 - Flow Call
 - Flow Return

Flow Definition

- Name of the flow
- Where does the flow start?
- Input values
- Optional initializer & finalizer



Safe Harbor Statement

The preceding is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.



JavaOne™

ORACLE®