



JavaServer™ Faces 2.0

A complete tour

Ed Burns <<http://ridingthecrest.com/>>

Senior Staff Engineer

Enterprise Java Platforms

Sun Microsystems, Inc.



Agenda

- Traverse all new features in JSF2
- Breadth, Not Depth
- All of this and more will be in Ed's upcoming McGraw-Hill Book: *JavaServer Faces 2.0: The Complete Reference*, Fall 2009.





JavaOne™

Java University at the JavaOneSM Conference

The Topics You Want Training On.
Delivered By the Experts!

Java University
May 31-June 1, 2009

JavaOneSM Conference
June 2-5, 2009

San Francisco, CA

There's never been a better time to give your skill-set a boost and network with fellow developers and experts in the industry. That's why we've added an extra day to Java University at JavaOneSM this year to include the hot topics that developers are asking for.

Sample Topics Include:

- Using Java™ EE and SOA to Help in Architecting and Designing Robust Enterprise Applications
- Designing and Implementing Secure Java Web Services
- Developing Enterprise Applications with the Spring Framework
- Web 2.0: Building Dynamic Websites with Ajax and the Dojo toolkit
- Web 2.0: Leveraging the Project jMaki and Google Web Toolkits for Rapid Website Development

Credit Where It's Due

- This was a really great Expert Group
- Representing a really great community of developers
- Both of whom deeply understand the essence of web applications and the state of the art of building them

Some of the Faces of Faces



Alexandr Smirnov



Jacob Hookom



Matthias Weißendorf



Adam Winer



Andy Schwartz



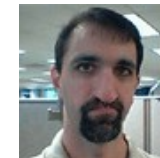
Ken Paulsen



Pete Muir



Gavin King



Joe Ottinger



Dan Allen



Martin Marinschek



Alexander Jesse



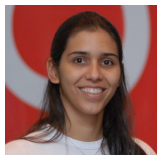
Kito Mann



Lincoln Baxter III



Rick Hightower



Yara Senger



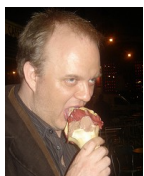
Neil Griffin



Mike Freedman



Jason Lee



Imre Oßwald



Stan Silvert



Ted Goddard

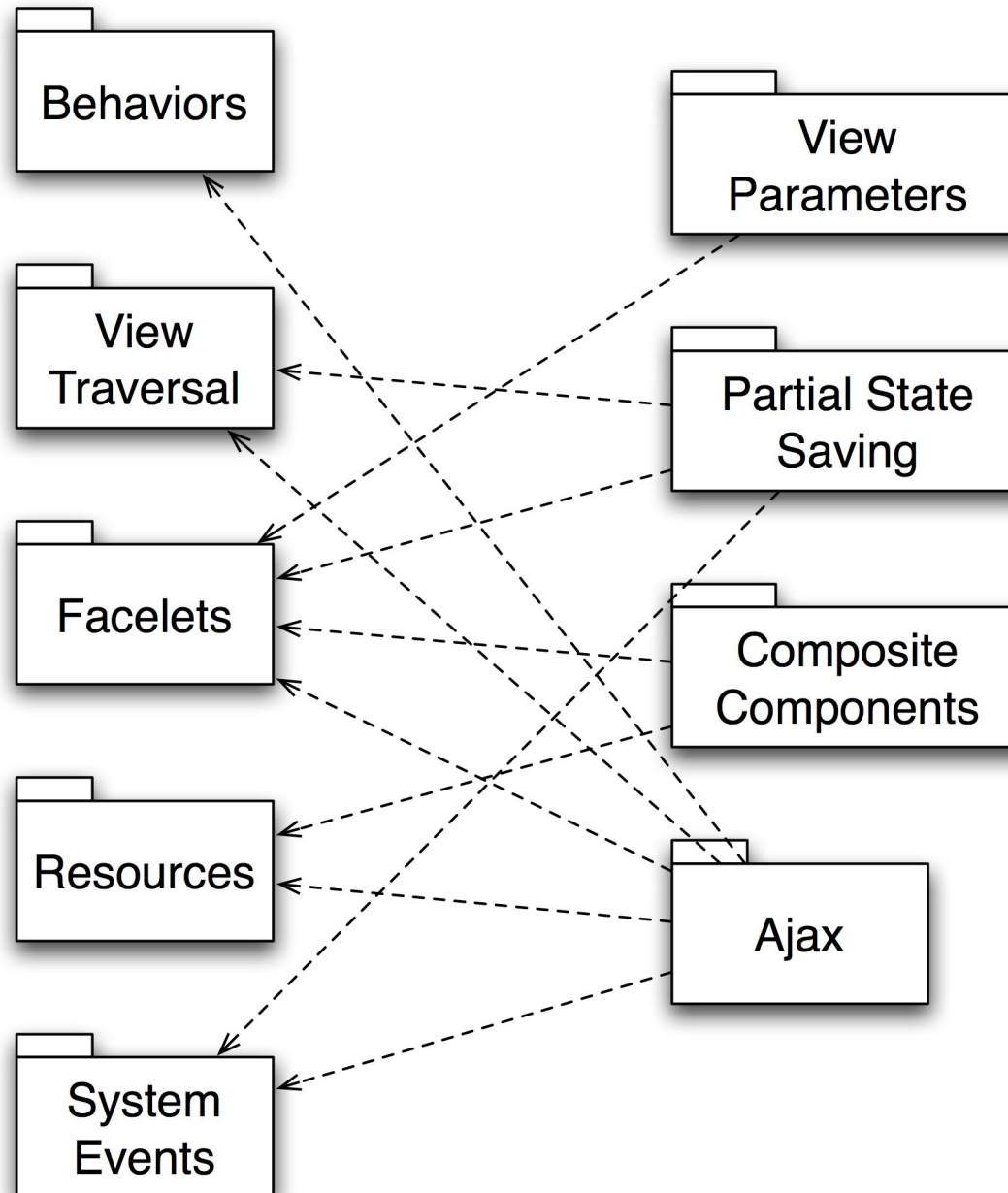


David Geary

High Level Categories of Features

Foundational New Features

Large New Features



High Level Categories of Features

Medium Sized New Features

New
Scopes

Validation

EL

Exceptions

Navigation

Config
Annotations

ProjectStage

UISelect*

FacesContext During
Startup/Shutdown

Small New Features

Petite New Features

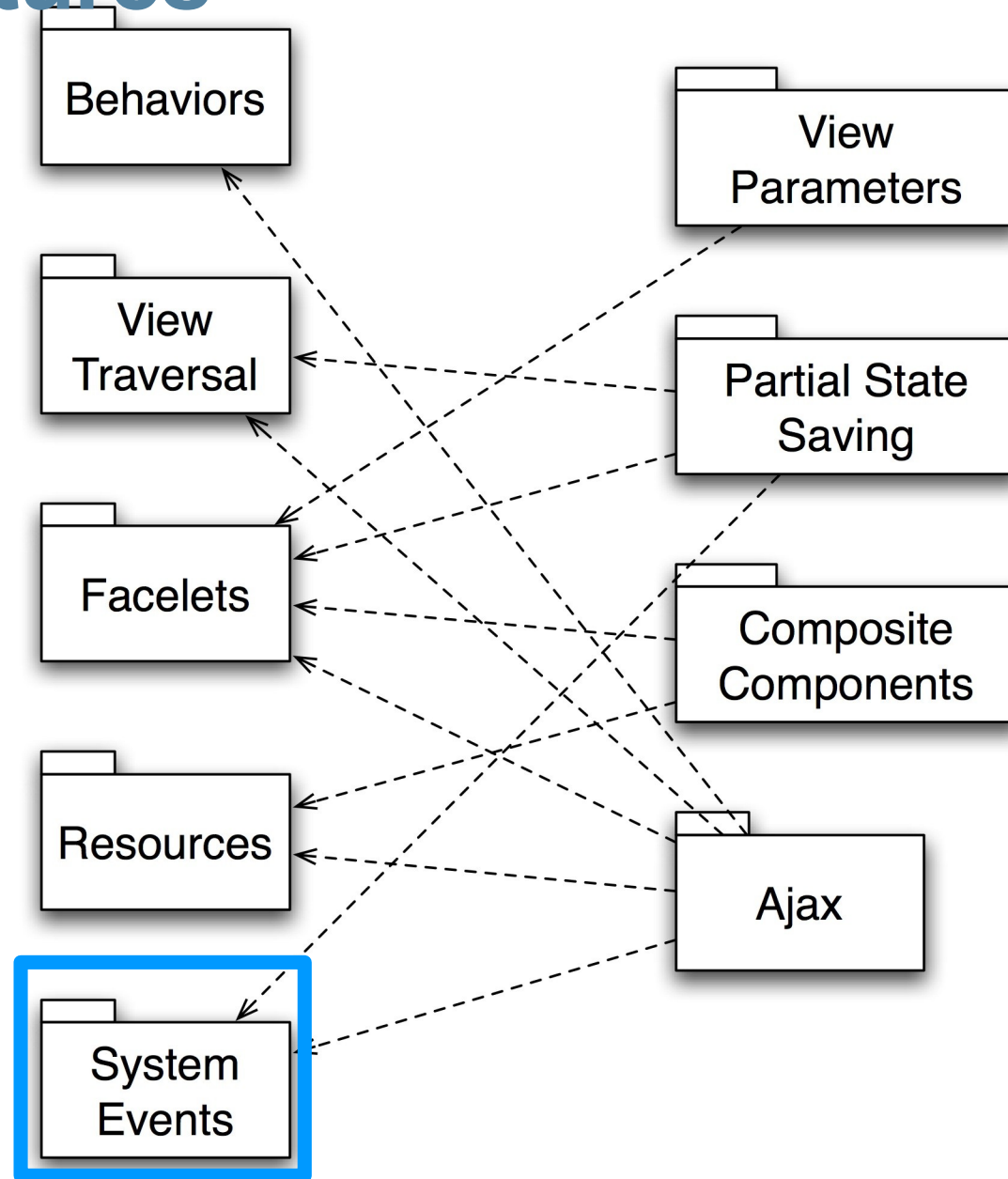
- **ExternalContext**
 - > **cookie, headers, realPath, contentLength, scheme, serverName, port, invalidateSession, responseReset...**
- **FacesContext**
 - > **currentPhaseId, messageList, validationFailed, postback**
- **RenderKit (componentFamilies, rendererTypes)**
- **Wrappers for decoratable artifacts**
- **DataModel.iterator**
- **javax.faces.
DATETIMECONVERTER_DEFAULT_TIMEZONE_IS_SYSTEM_TIMEZONE**

Petite New Features

- **EditableValueHolder.resetValue()**
- **UIViewRoot.getPhaseListeners()**
- **SEPARATOR_CHAR_PARAM_NAME, UINamingContainer.getSeparatorChar(), NamingContainer.SEPARATOR_CHAR** deprecated
- Ordering of artifacts is well-defined now

High Level Categories of Features

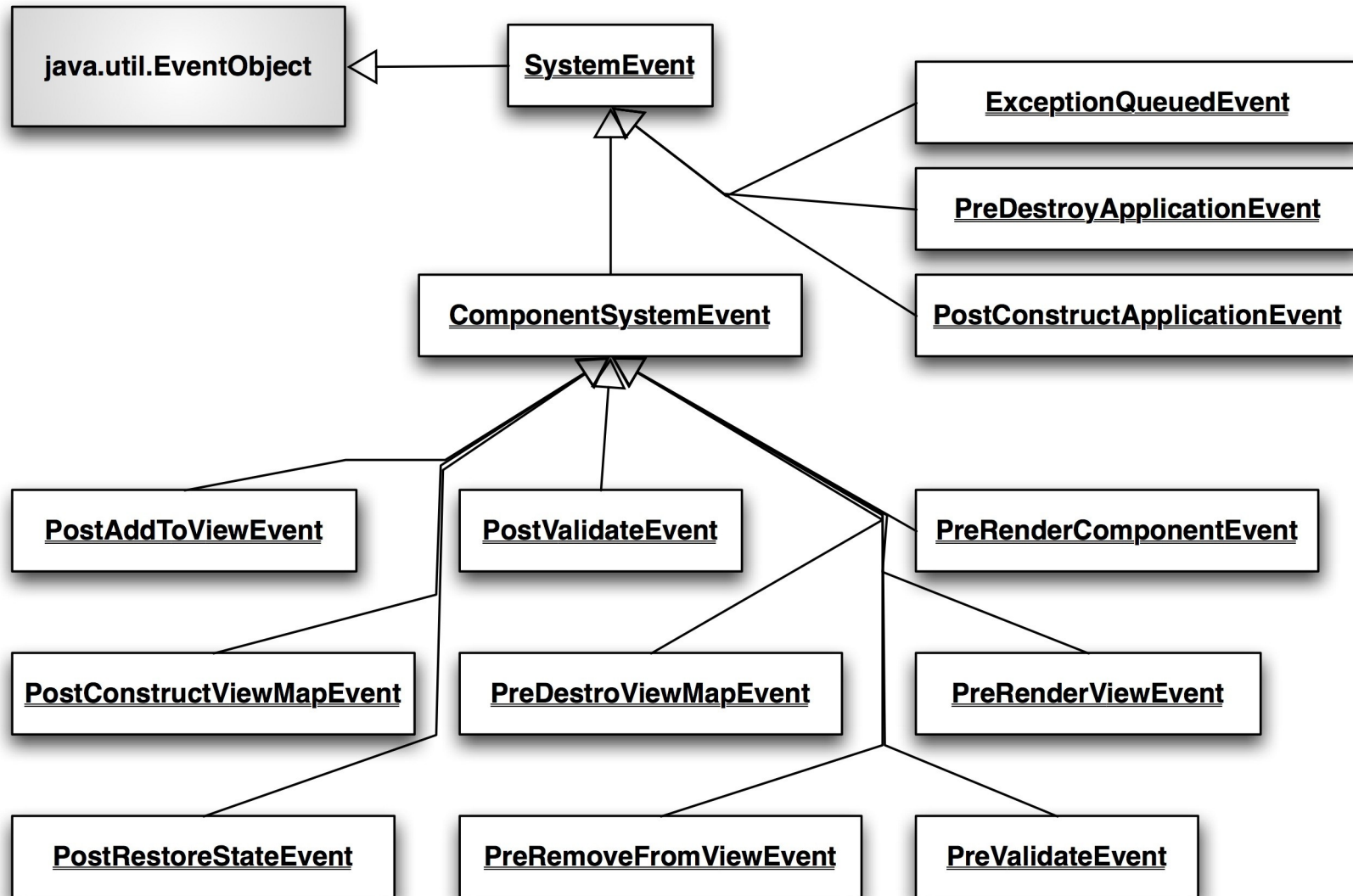
Foundational New Features



System Events

- Influenced by Ken Paulsen's JSFTemplating, and RedHat's experience with Seam
- Publish/Subscribe event bus for things that happen during the JSF Lifecycle, **not application specific**
- Inspired by Solaris Dtrace, Linux strace, truss, etc.
- Listeners can be registered at three scopes
 - > component **UIComponent.subscribeToEvent()**
 - > view **UIViewRoot.subscribeToEvent()**
 - > application **Application.subscribeToEvent()**
- Publish is always with **Application.publishEvent()**

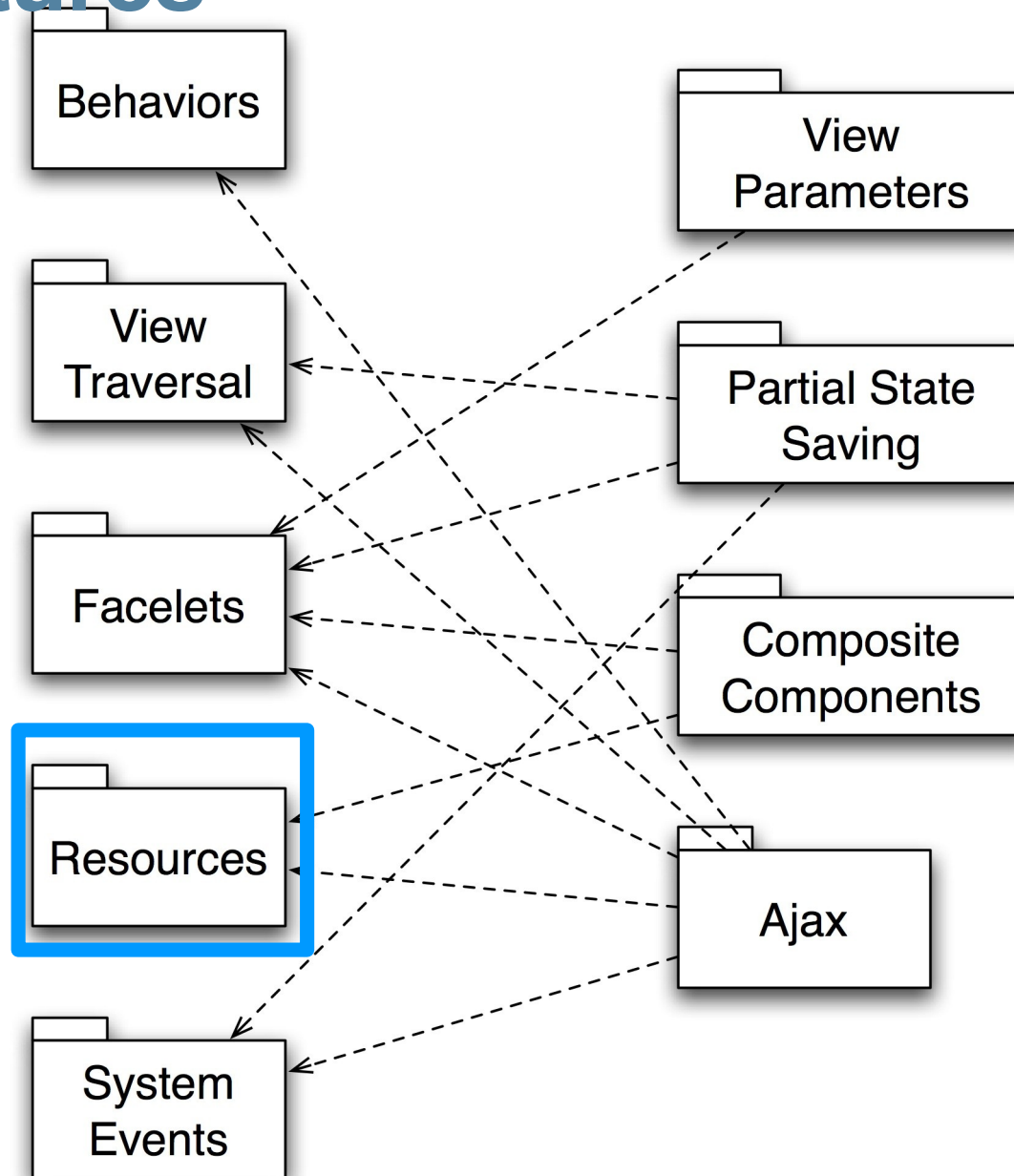
System Events: Standard Event Types



High Level Categories of Features

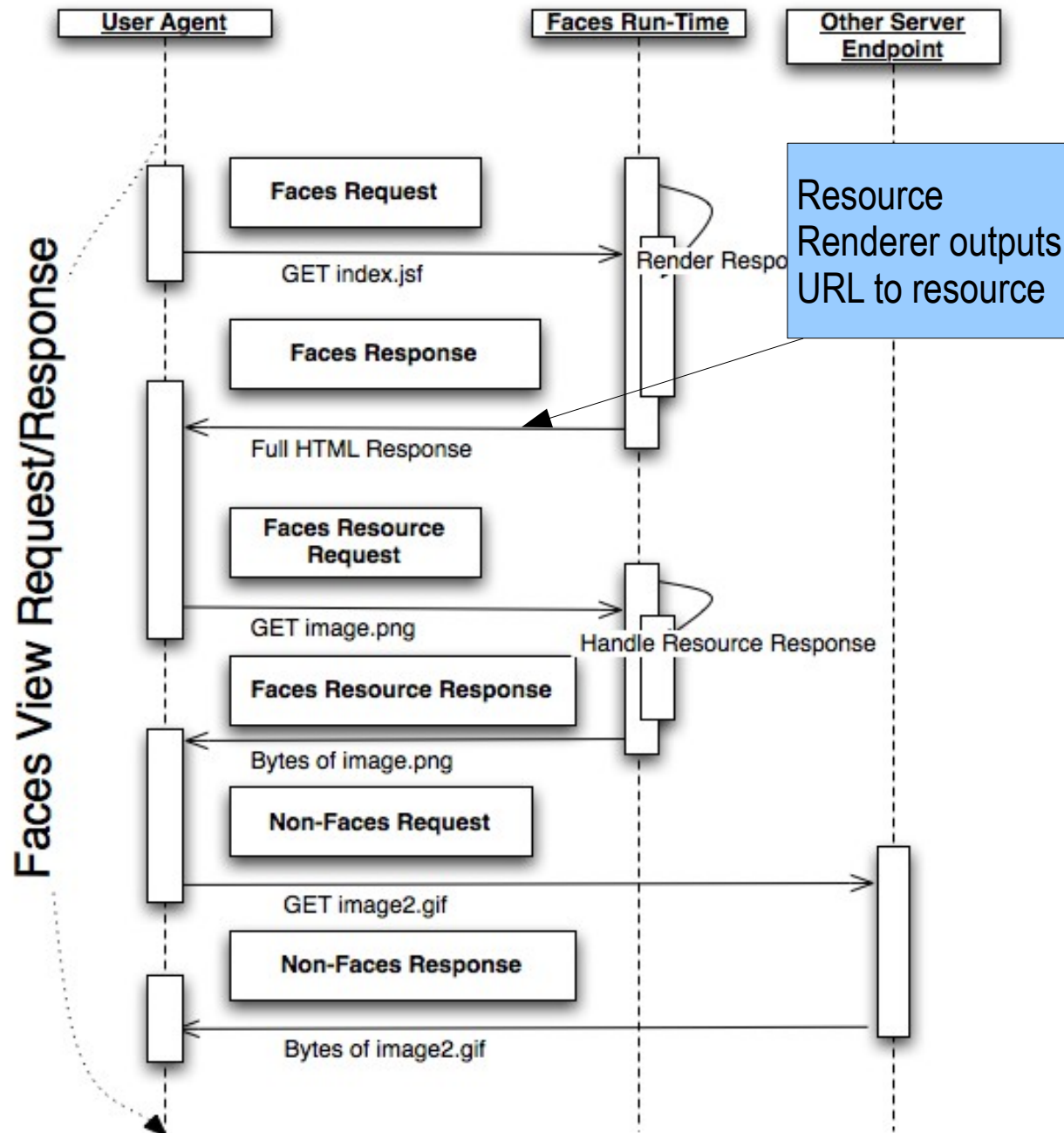
Foundational New Features

Large New Features



Resources

- Avoid need for separate Filter or Servlet.
- Resources are logically related to components, treat them that way.
- Load from Classpath, or filesystem
- Full support for “Library” concept, I18N, Versioning



Resources: Java API

- **@ResourceDependency** or **@ResourceDependencies** annotation on
 - > **UIComponent, Renderer, Validator, Converter, ClientBehavior**
- attributes for resource name, library name
- attribute for target: head, body, form
- Built in support for CSS, Image, JavaScript resources

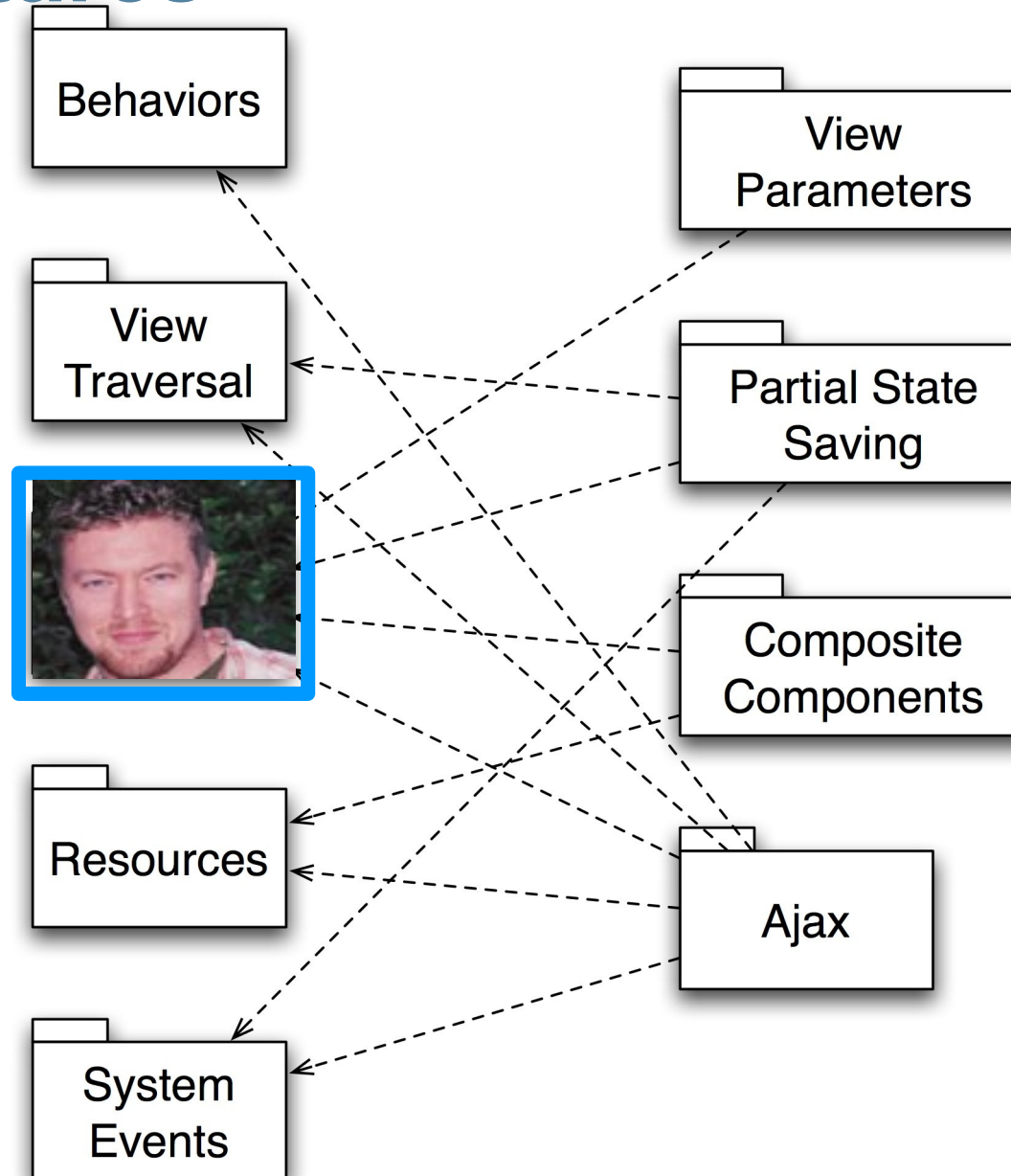
Resources: Markup API

- **<h:head>, <h:body>**
- **<h:outputScript>, <h:outputStylesheet>**
- Similar attributes for resource name, library name
- Similar attribute for target: head, body, form

High Level Categories of Features

Foundational New Features

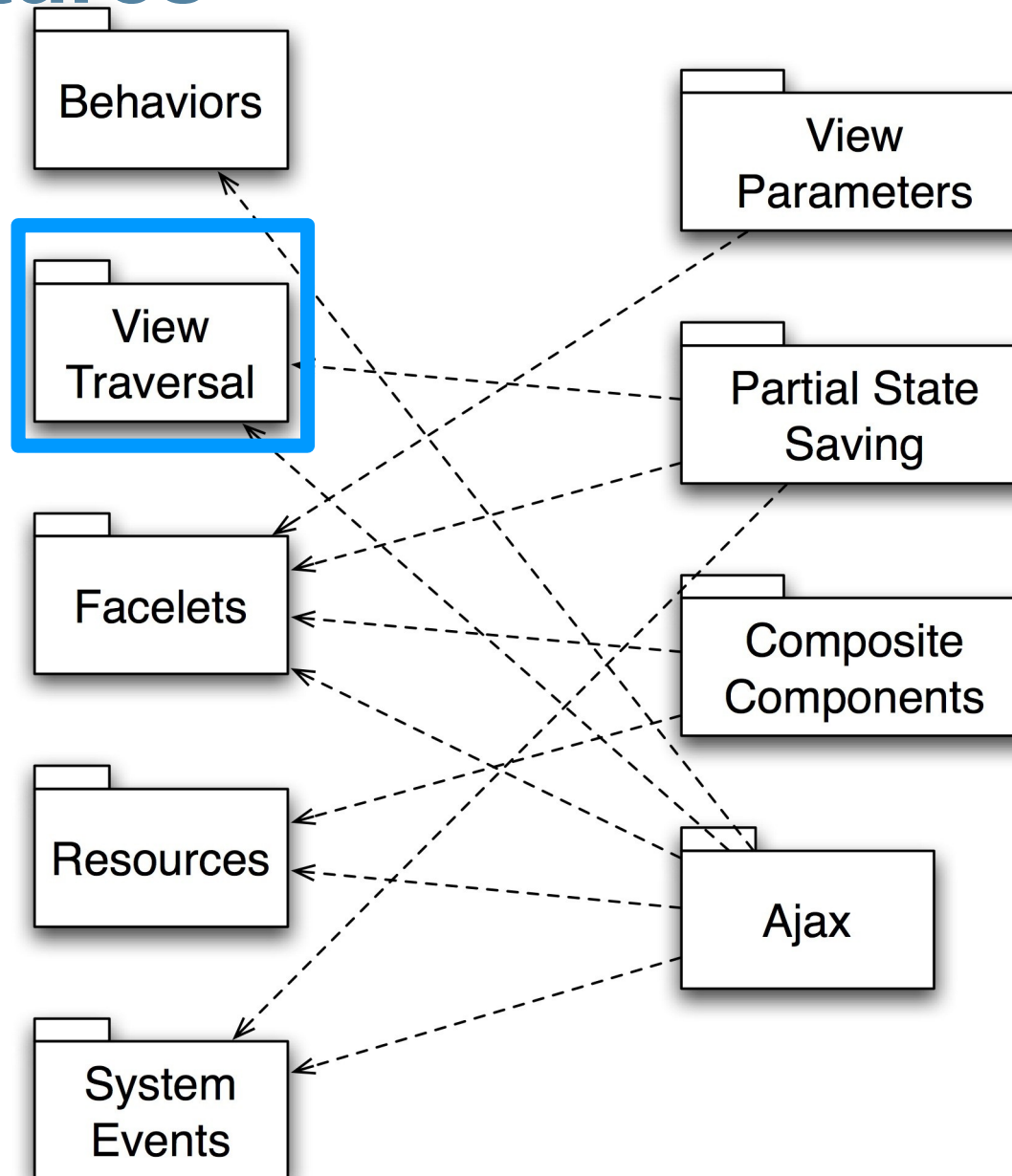
Large New Features



High Level Categories of Features

Foundational New Features

Large New Features



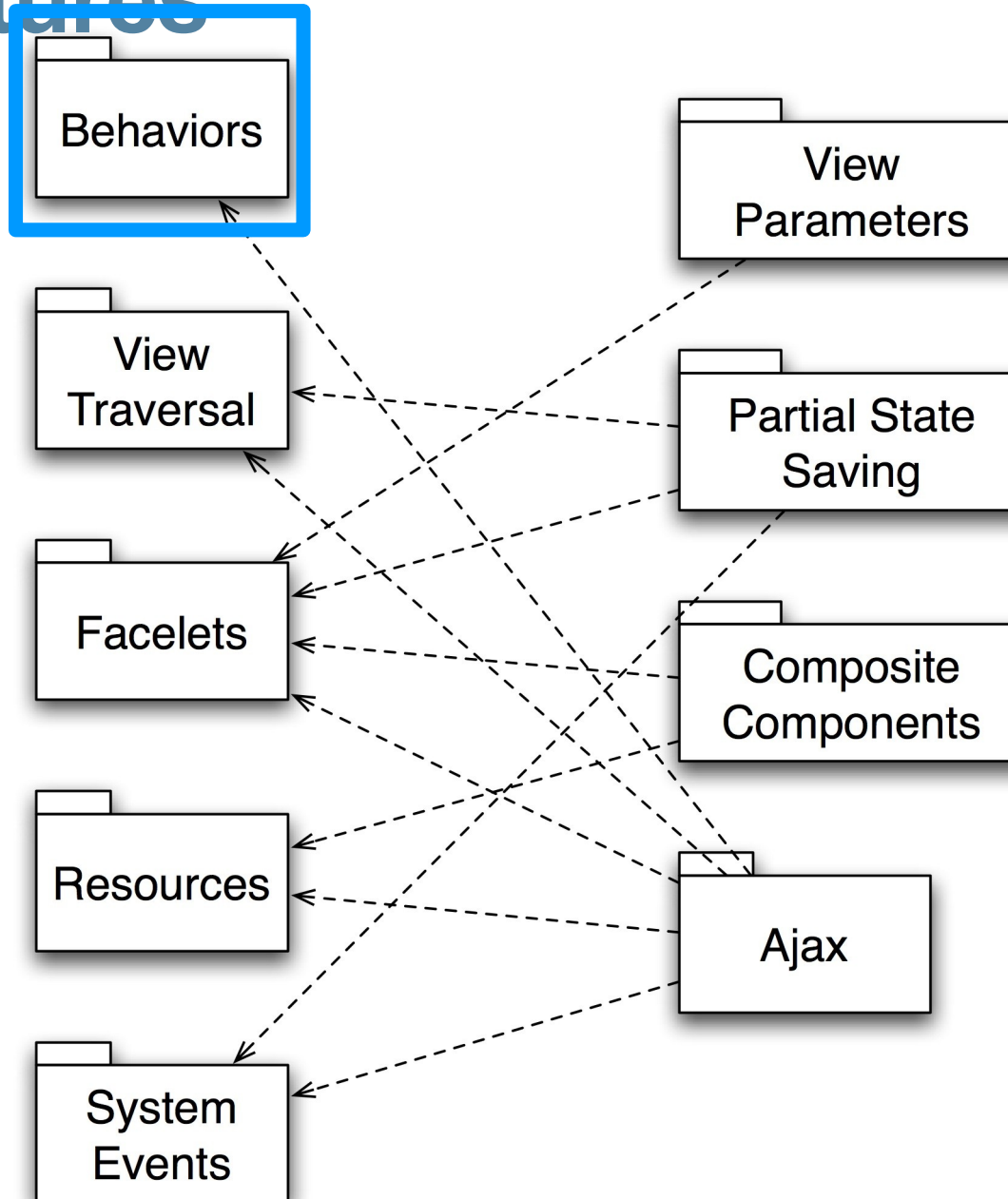
View Traversal

- Contributed by Andy Schwartz, Oracle
- **VisitContext.createVisitContext()**
- **UIComponent.visitTree(VisitContext)**
- Used on Ajax requests for partial traversal
- Used by new **StateManagementStrategy**

High Level Categories of Features

Foundational New Features

Large New Features



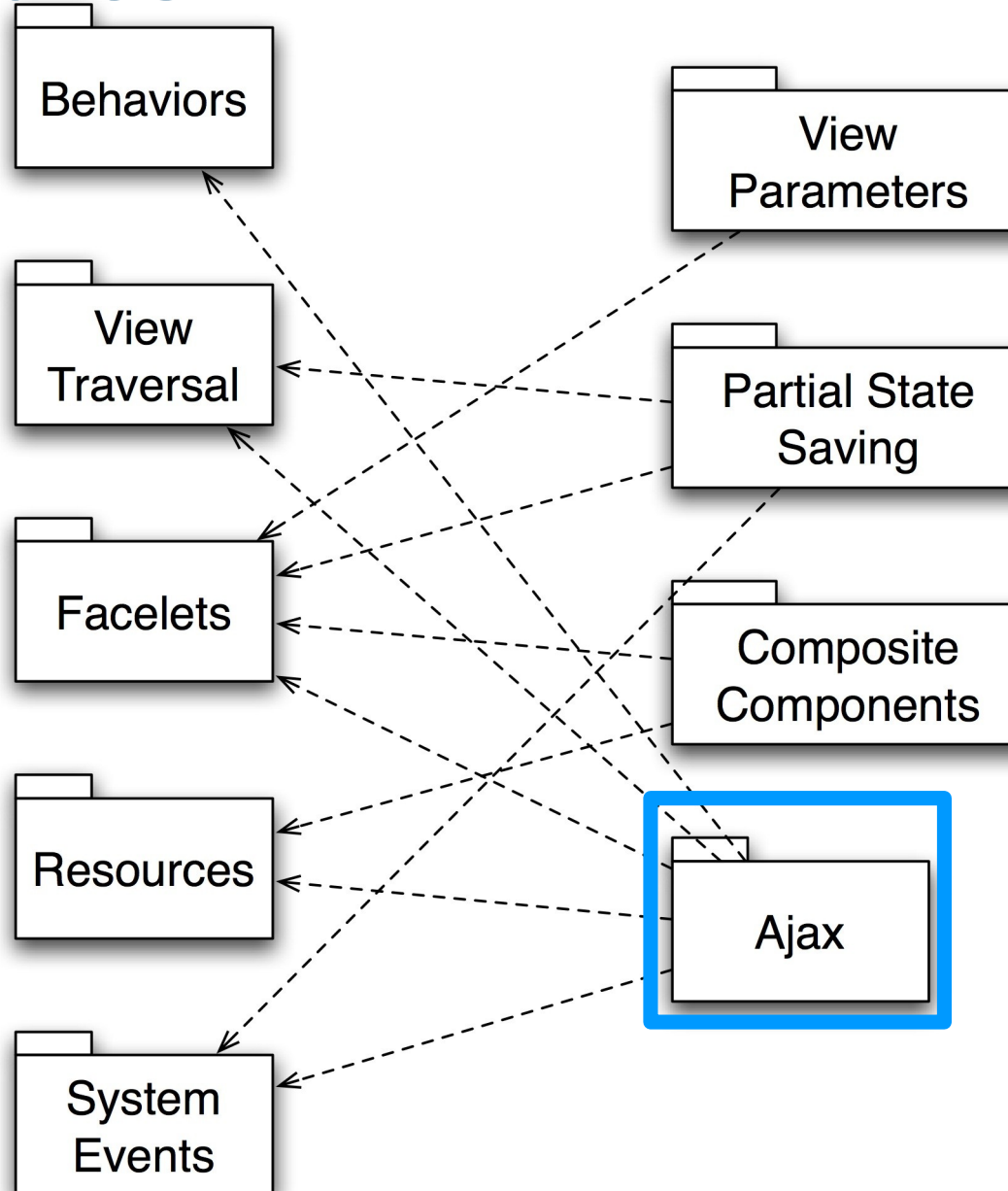
Behaviors

- Contributed by Andy Schwartz, Oracle and Alexandr Smirnov, Exadel
- A **Behavior** is an interface and invents a new type of “attached object” which takes part in decode/encode.
- **ClientBehavior** extends **Behavior**
- Define things that happen in the view, in the browser
- Currently only one standard **ClientBehavior** implementation: **AjaxBehavior**
- New interface: **ClientBehaviorHolder**, implemented by all components in **javax.faces.component.html**
- Special **ClientBehaviorRenderer**, does not extend **Renderer**.
 - > **getScript()**, **decode()**

High Level Categories of Features

Foundational New Features

Large New Features



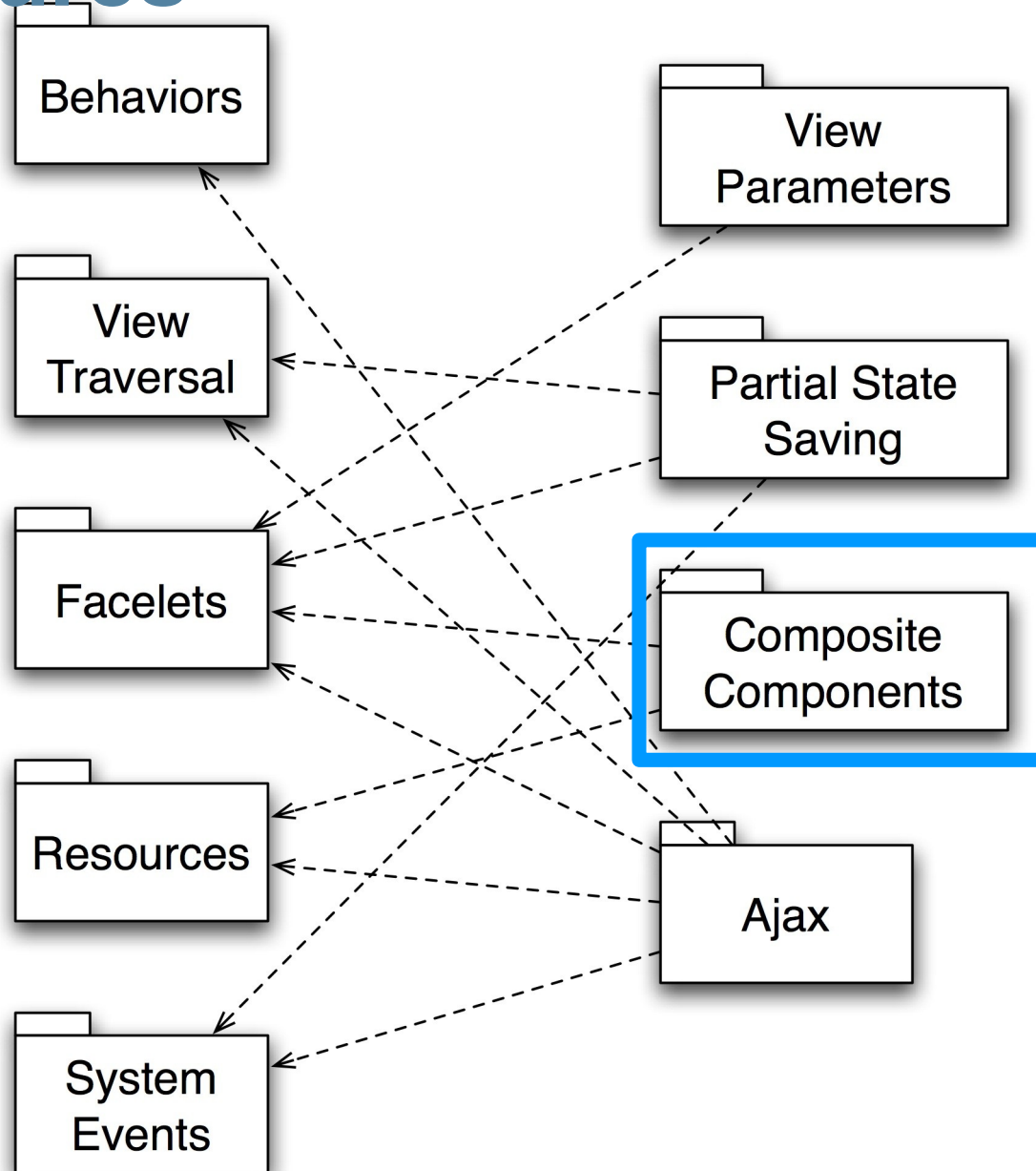
Ajax

- Inspired by ADF Faces, RichFaces, IceFaces, DynamicFaces
- Two entry points to the feature
 - > Declarative: **<f:ajax>** tag, uses **AjaxBehavior**
 - > Programmatic ajax: resource library **javax.faces**, resource name **jsf.js**, JavaScript namespace **jsf.ajax**
- Declarative ajax useful to ajaxify non ajax pages
 - > **AjaxBehavior** gives great flexibility for most ajax needs
- Programmatic ajax API usage pattern
 - > **jsf.ajax.request()** JavaScript method

High Level Categories of Features

Foundational New Features

Large New Features



Composite Components

- Enable True Abstraction
 - > Create a true, reusable, OO Black Box Component from an arbitrary region of a page
 - > Full support for using attached objects in the using page, and mapping them to arbitrary targets within the composite component
 - > Full support for Faces Event System
- Heavily leverages naming conventions
- Builds on top of Resources and Facelets
- Top Level Component may be written in script, Mojarra supports Groovy.

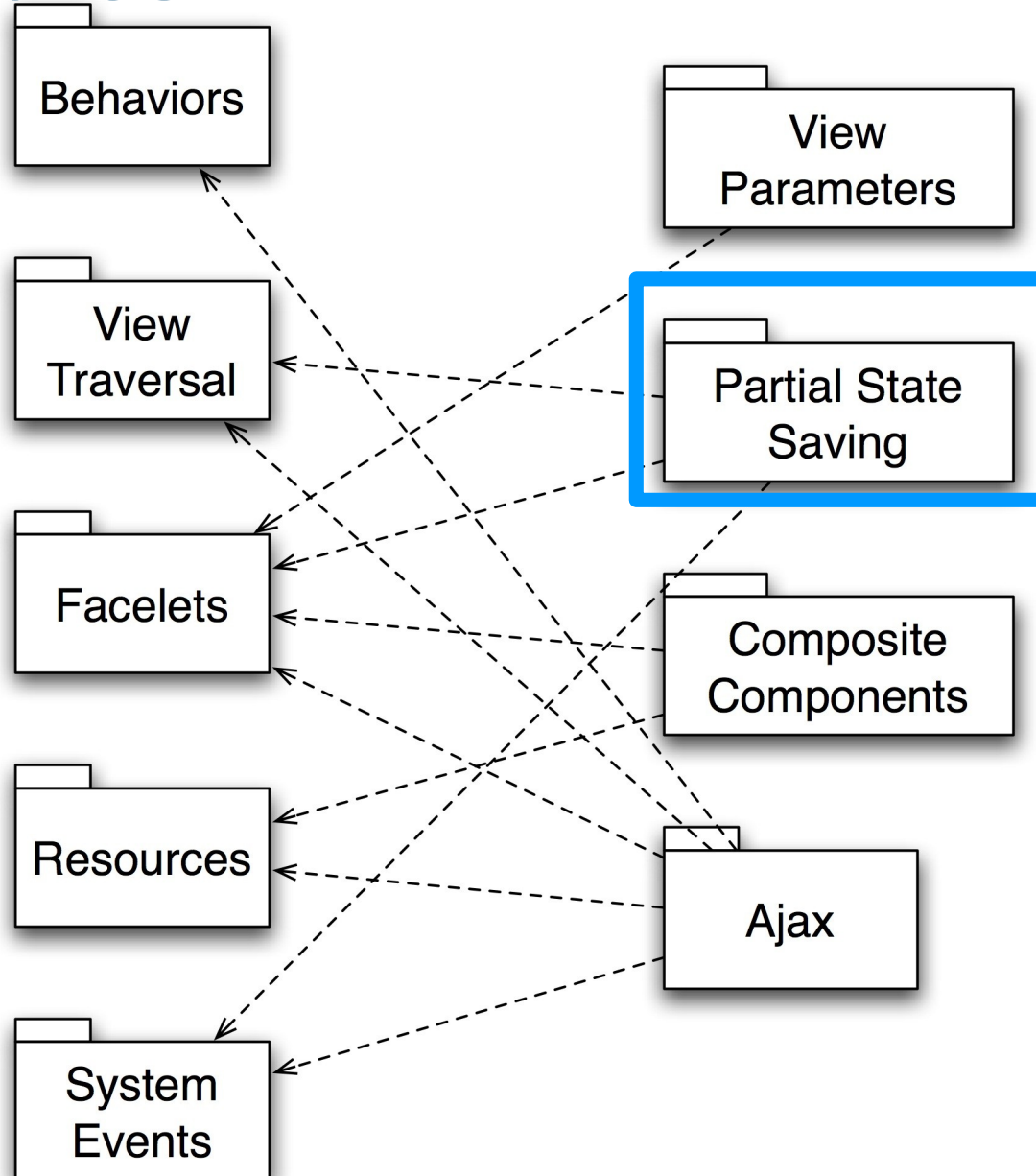
Composite Components – more than Facelets

- true component creation
 - > attached objects
 - > facets
 - > children
- dynamic component discovery
- interface declaration (via meta-data)
- per component **ResourceBundle**
- define top level component type (including script-based components)

High Level Categories of Features

Foundational New Features

Large New Features



Partial State Saving

- Inspired by Trinidad state saving
- Default for pages written with Facelets for JSF 2.0
- Of interest to Java custom component authors
- All of the standard components implement this feature.
This means your composite components will automatically take advantage of it.
- Per-view state size up to 4X smaller than before
- **PartialStateHolder** behavioral interface, extends existing **StateHolder**.
- **StateHelper**, from **UIComponent**, does most of the work: replace ivars with keys within **StateHelper**.

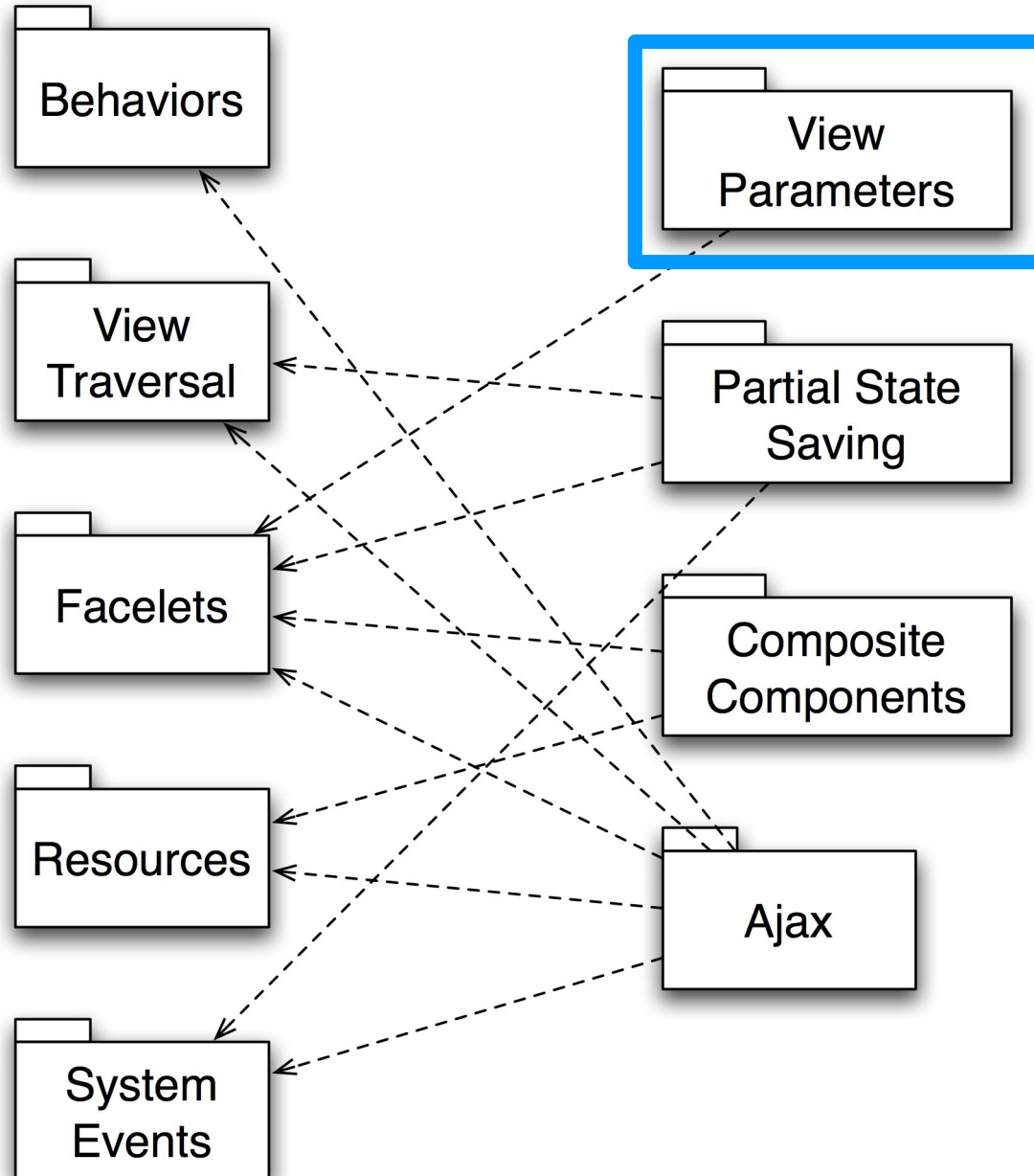
Simplifies custom components

```
public class MyInput extends UIComponentBase {  
    private enum Keys { name }  
    public String getName() {  
        return getStateHelper().get(name);  
    }  
    public void setName(String nameParam) {  
        getStateHelper().put(name, nameParam);  
    }  
}
```

High Level Categories of Features

Foundational New Features

Large New Features



View Metadata and Parameters

- Inspired by Page Parameters from JBoss Seam, Pete Muir and Dan Allen
- Provides a way to map request parameters to special components within the view
- **<f:metadata><f:viewParam></f:metadata>**
- **UIViewParameter** extends **UIInput**
- All attached objects valid for **UIInput** are valid
- Breaks the tyranny of POST
- Lays the foundation for Bookmarkable pages

High Level Categories of Features

Medium Sized New Features

New
Scopes

Validation

EL

Exceptions

Navigation

Config
Annotations

ProjectStage

UISelect*

FacesContext During
Startup/Shutdown

Small New Features

Navigation

- Bookmarkable Pages
 - > Uses View Parameters feature to ensure request params are validated before rendering page
 - > New components **<h:link>**, **<h:button>**, cause the browser to submit GET request, contrast with **<h:command{Button,Link}>**
- Conditional navigation **<if>#{shoppingCart.filled}</if>**
- Implicit Navigation
 - > If you don't want all that **<navigation-rules>** stuff, you can put the navigation logic straight in the page.
- **ConfigurableNavigationHandler**
 - > Query navigation rules at run-time

High Level Categories of Features

Medium Sized New Features

New
Scopes

Validation

EL

Exceptions

Navigation

Config
Annotations

ProjectStage

UISelect*

FacesContext During
Startup/Shutdown

Small New Features

Exceptions

- Contributed by Pete Muir
- Exceptionhandler
 - > can be decorated
 - > all interesting exceptions flow through the exception handler!
 - > new exception type: UpdateModelException
 - > exceptions during update model don't queue faces-messages
- **ValidatorException** contains multiple **FacesMessages**

High Level Categories of Features

Medium Sized New Features

New
Scopes

Validation

EL

Exceptions

Navigation

Config
Annotations

ProjectStage

UISelect*

FacesContext During
Startup/Shutdown

Small New Features

EL

- **`{component}`**, **`{cc}`**: get the “currently” processed component / composite component
- **`{component.clientId}`**
- **`{component.messageList}`**
- Make `ActionEvent` param optional for action-listeners
- For EE6, EL method invocation supports parameters, thanks to Gavin King with support from Jacob Hookom
 - > makes `f:setPropertyActionListener` obsolete

High Level Categories of Features

Features

Medium Sized New

New
Scopes

Validation

EL

Exceptions

Navigation

Config
Annotations

ProjectStage

UISelect*

FacesContext During
Startup/Shutdown

Small New
Features

Validation

- Integration with JSR 303: Bean Validation from Emmanuel Bernard at JBoss
 - > **@NotEmpty private String name;**
 - > default validator: javax.faces.Bean – automatically applied to all input fields
- Default-validator: hook up a validator for all instances of EditableValueHolder
- preValidate/postValidate system events

High Level Categories of Features

Features

Medium Sized New

New
Scopes

Validation

EL

Exceptions

Navigation

Config
Annotations

ProjectStage

UISelect*

FacesContext During
Startup/Shutdown

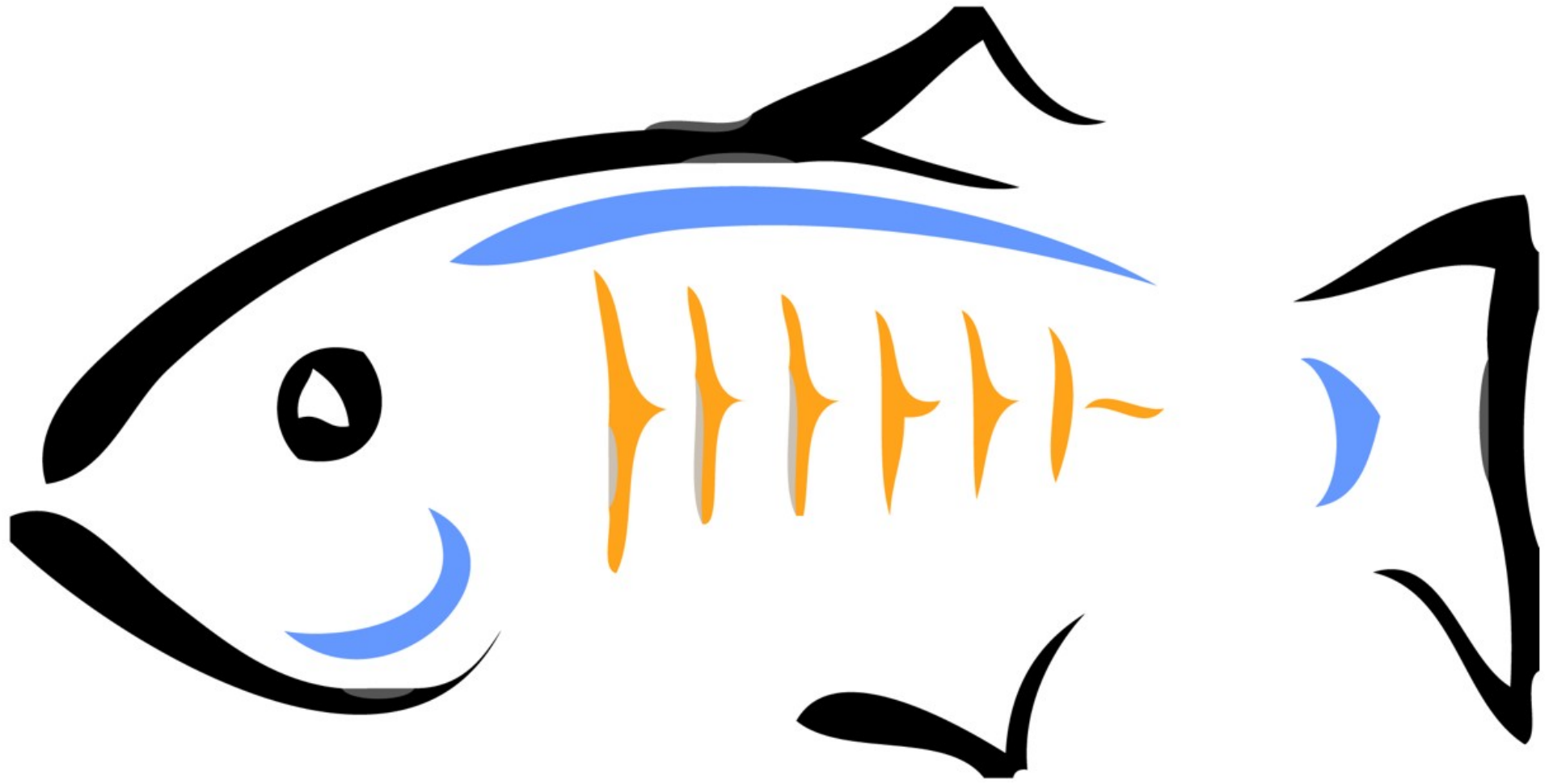
Small New
Features

New Scopes

- “conversation” comes from Java Contexts and Dependency Injection (JSR 299) nee WebBeans
- Flash
 - > inspired by Ruby on Rails
 - > accessible from EL or ExternalContext
 - > useful for storing data for use on the “next” request (e.g. messages across redirects)
- View
 - > persists as long as user interacts with the same view
 - > managed-beans can be put in view-scope
- `FacesContext.getAttributesMap()`

And Now a Word from Our Sponsor

What is GlassFish ?



GlassFish: It's a community!



GlassFish: Open Source and Enterprise Quality

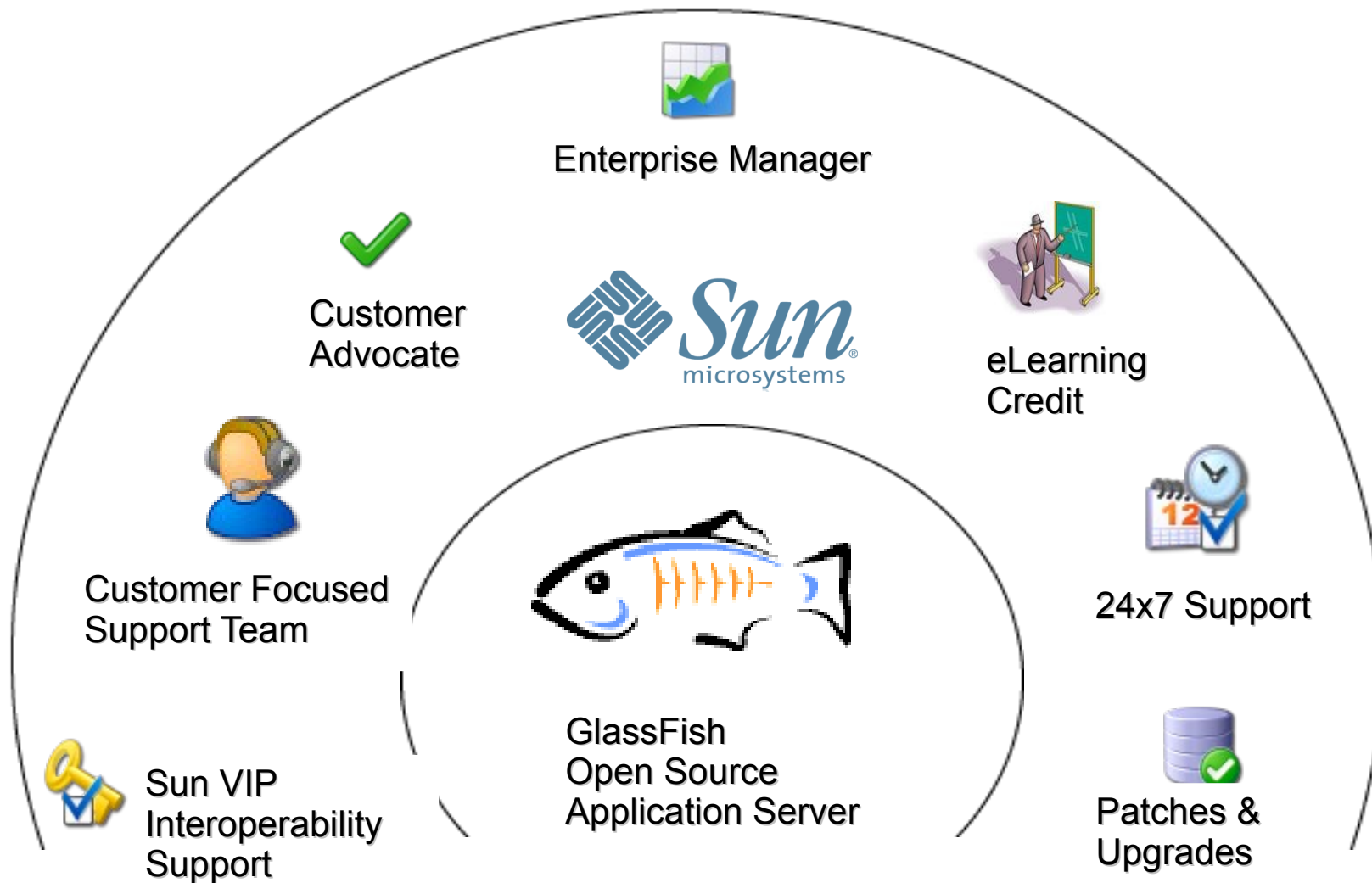


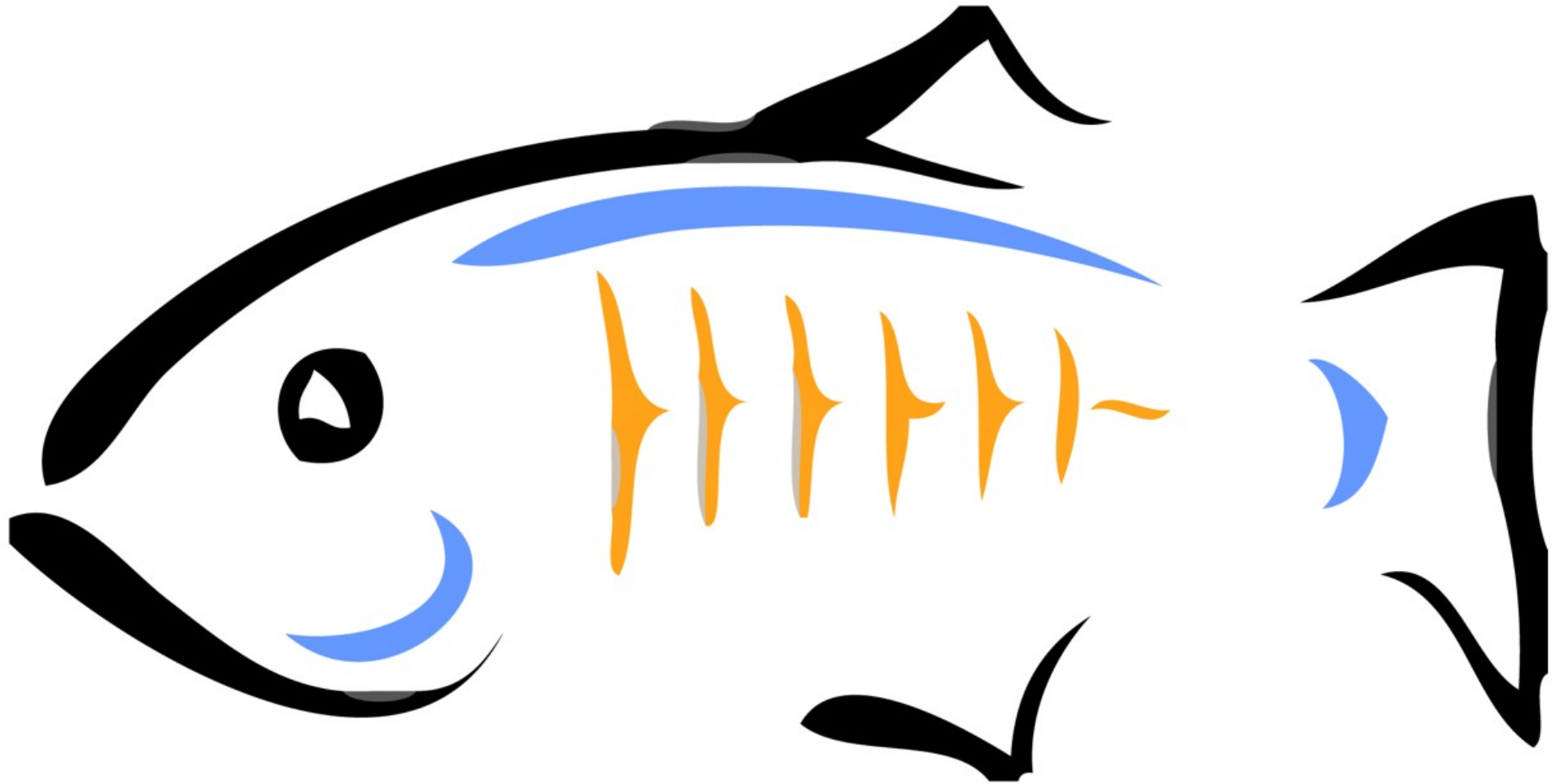
GlassFish: Java EE 5 and 6 Reference Implementation

Java EE 5
Java EE 6



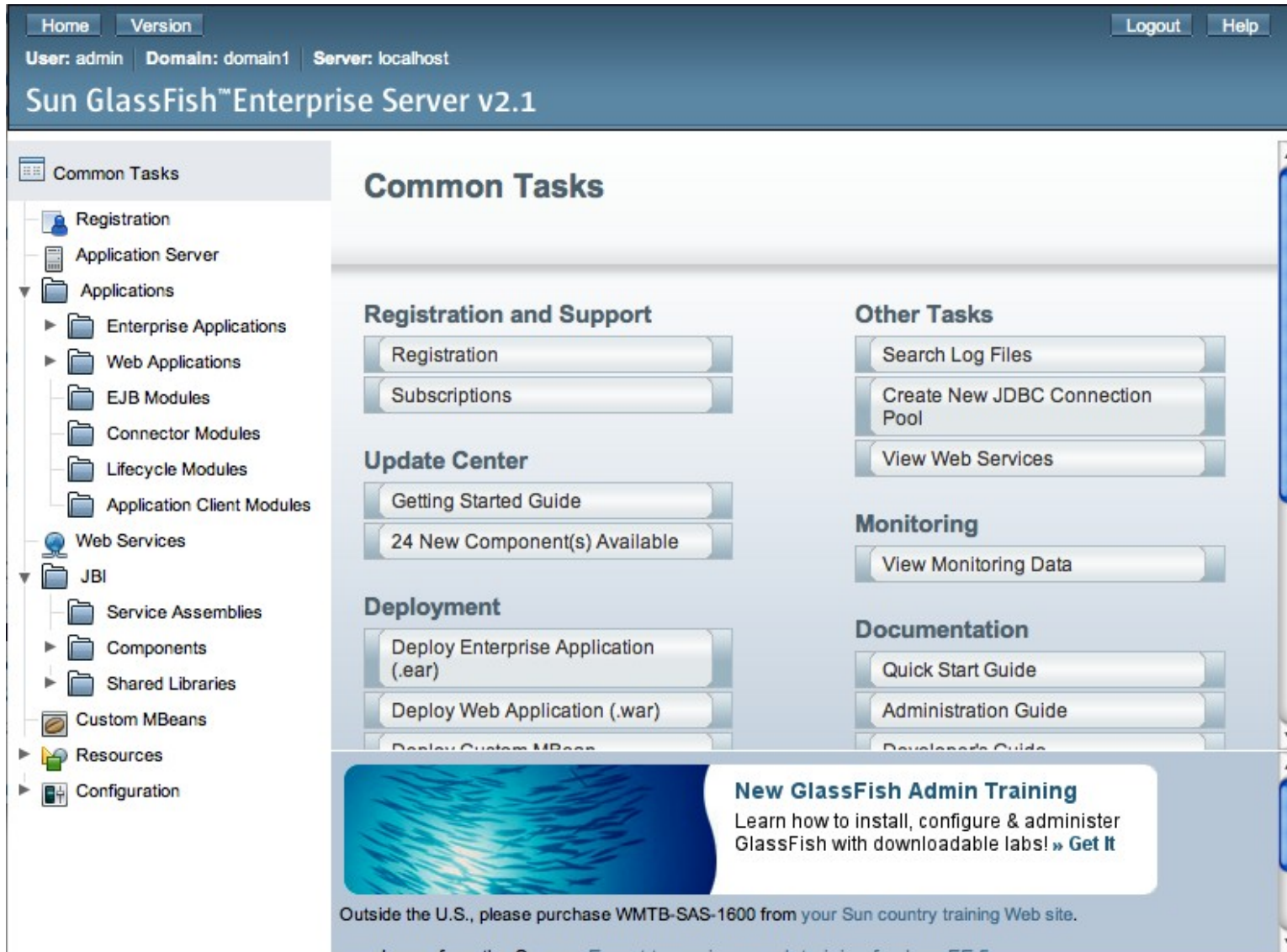
Sun GlassFish Enterprise Server





GlassFish v2

GlassFish v2: Web-based Administration



The screenshot displays the Sun GlassFish Enterprise Server v2.1 web-based administration interface. The top navigation bar includes links for Home, Version, Logout, and Help. Below this, the user information is shown: User: admin, Domain: domain1, and Server: localhost. The main title is "Sun GlassFish™ Enterprise Server v2.1".

On the left side, there is a "Common Tasks" sidebar with a tree view containing the following items:

- Registration
- Application Server
- Applications
 - Enterprise Applications
 - Web Applications
 - EJB Modules
 - Connector Modules
 - Lifecycle Modules
 - Application Client Modules
- Web Services
- JB1
 - Service Assemblies
 - Components
 - Shared Libraries
- Custom MBeans
- Resources
- Configuration

The main content area is titled "Common Tasks" and is divided into several sections:

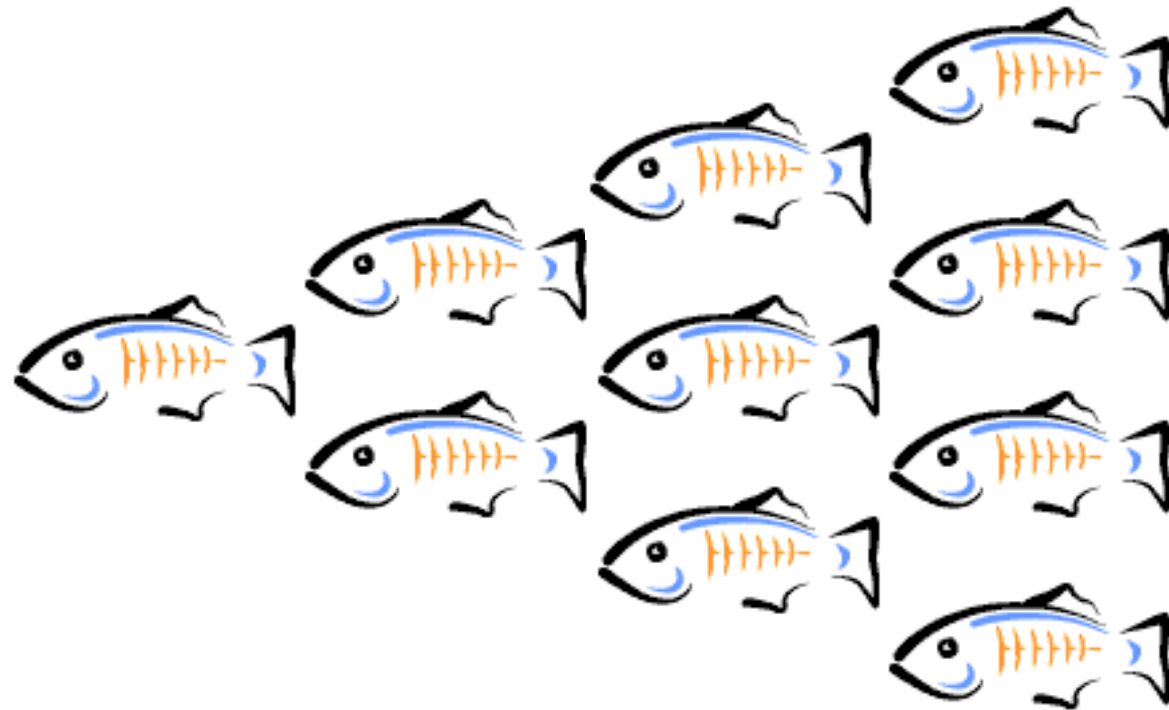
- Registration and Support**
 - Registration
 - Subscriptions
- Update Center**
 - Getting Started Guide
 - 24 New Component(s) Available
- Deployment**
 - Deploy Enterprise Application (.ear)
 - Deploy Web Application (.war)
 - Deploy Custom MBeans
- Other Tasks**
 - Search Log Files
 - Create New JDBC Connection Pool
 - View Web Services
- Monitoring**
 - View Monitoring Data
- Documentation**
 - Quick Start Guide
 - Administration Guide
 - Developer's Guide

At the bottom of the main content area, there is a banner for "New GlassFish Admin Training" with the text: "Learn how to install, configure & administer GlassFish with downloadable labs! » [Get It](#)". Below the banner, a note states: "Outside the U.S., please purchase WMTB-SAS-1600 from your Sun country training Web site."

GlassFish v2: SOAP, REST, .NET interoperability



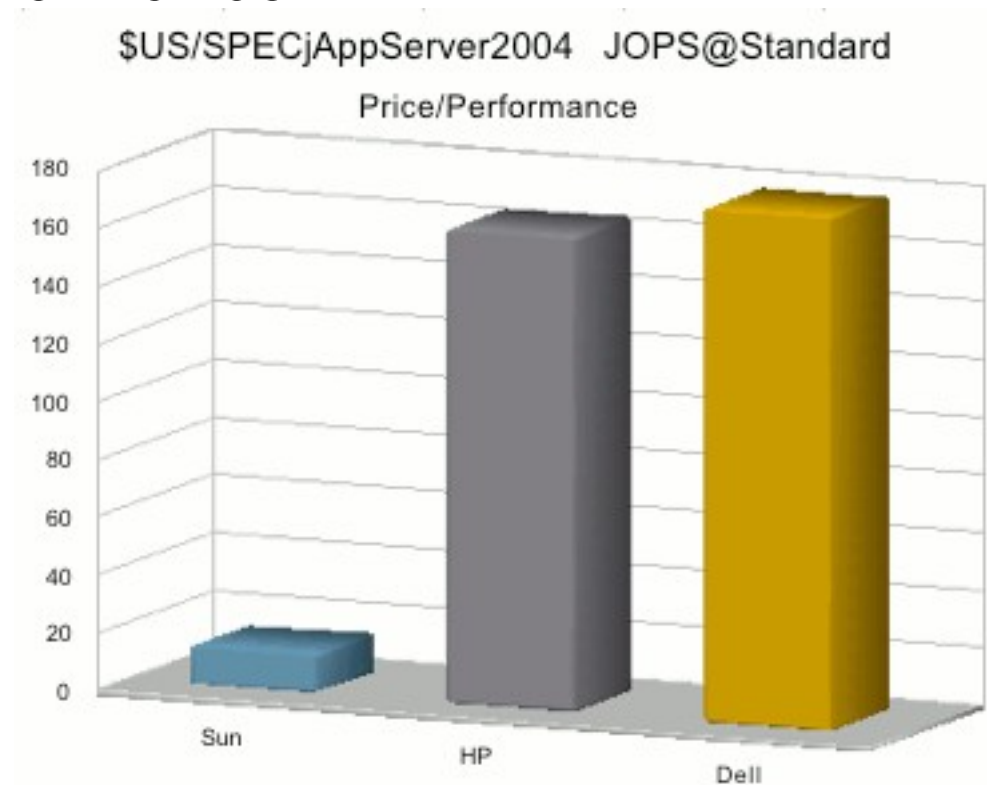
GlassFish v2: Clustering and Load Balancing



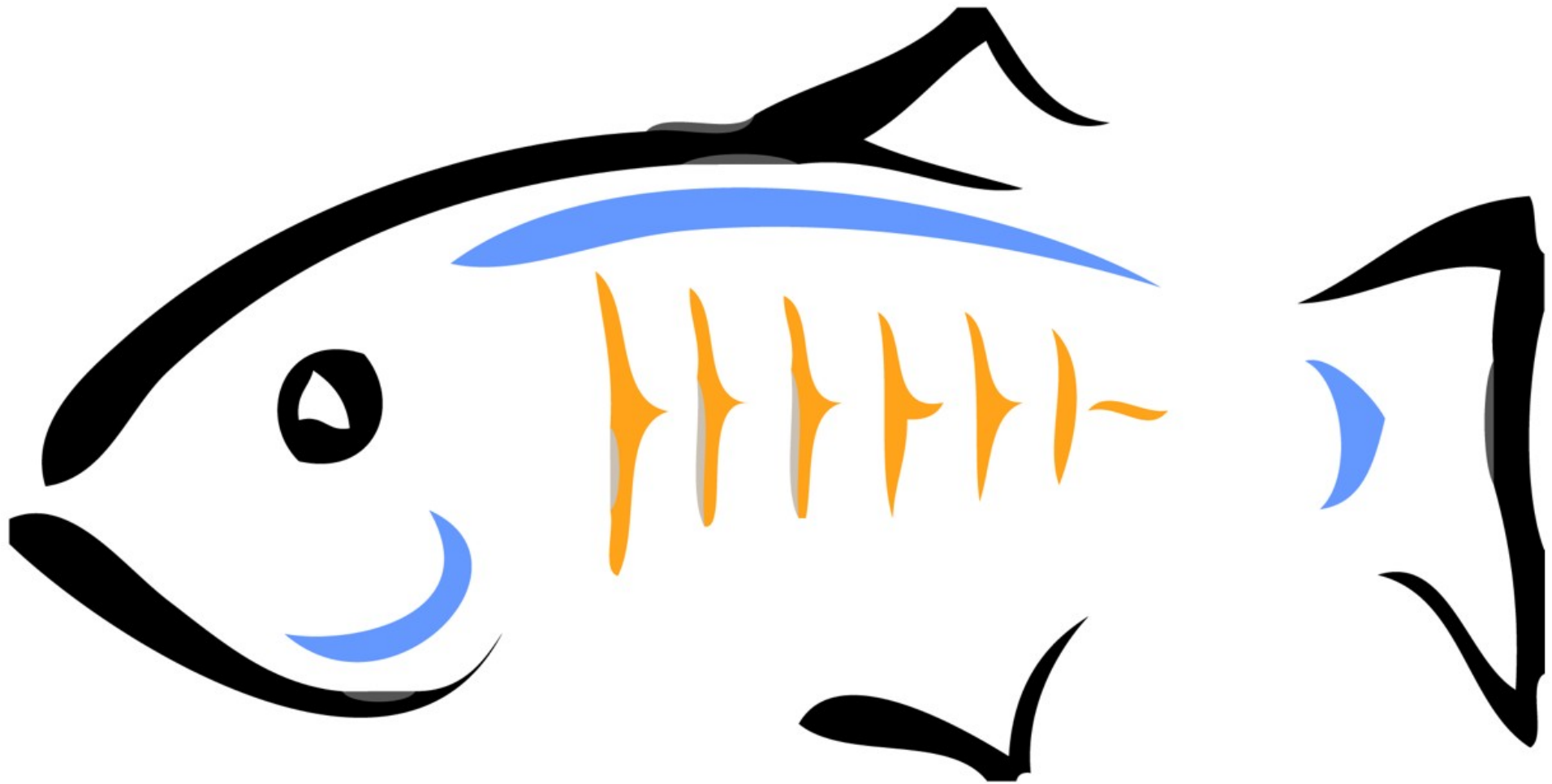
Best Price Performance Ever

SpecjAppServer2004 (Nov 2008)

- 2% cost of Dell, 5% cost of HP (Oracle)
- 13 times better price/performance



<http://www.sun.com/servers/x64/x4150/benchmarks.jsp#3>



GlassFish v3

GlassFish v3: Modular



GlassFish v3: Modular



GlassFish v3: Embeddable



GlassFish v3: Extensible



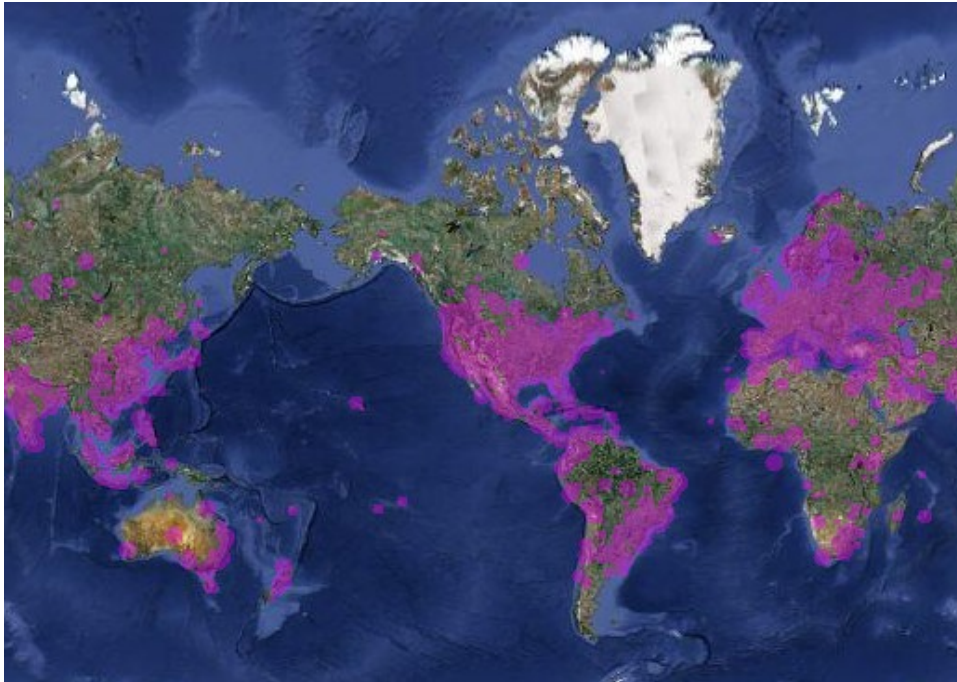
GlassFish v3: Containers



Dynamic Languages & Frameworks



Some Adoption Indicators



- 8.1m hits in 2 years
- 535,568 in Apr 2009

- 395k registrations
- 9m+ downloads in 2008
- 844K in Feb 2009



<http://beta.glassfish.java.net:81/maps/>

High Level Categories of Features

Medium Sized New Features

New
Scopes

Validation

EL

Exceptions

Navigation

Config
Annotations

ProjectStage

UISelect*

FacesContext During
Startup/Shutdown

Small New Features

High Level Categories of Features

Medium Sized New Features

New
Scopes

Validation

EL

Exceptions

Navigation

Config
Annotations

ProjectStage

UISelect*

FacesContext During
Startup/Shutdown

Small New Features

High Level Categories of Features

Medium Sized New Features

New
Scopes

Validation

EL

Exceptions

Navigation

Config
Annotations

ProjectStage

UISelect*

FacesContext During
Startup/Shutdown

Small New Features

High Level Categories of Features

Medium Sized New Features

New
Scopes

Validation

EL

Exceptions

Navigation

Config
Annotations

ProjectStage

UISelect*

FacesContext During
Startup/Shutdown

Small New Features

Summary

- JSF 2.0 is a major step forward for JSF
- Lots of new features which will simplify writing JSF applications
- Lots of new features which will simplify writing JSF components and extensions
- Plus: significant performance enhancements

Questions?

Answers!

Thanks for your attention.