

# JavaServer Faces with MyFaces

Building Standards Based  
Web Applications

# Agenda

- JSF Overview
- Patterns & Architecture of JSF
- JSF Request/Response Lifecycle
- Components
- Validation & Conversion
- Events & Listeners
- Renderers
- MyFaces Extras

# JSF Overview

- What is JSF?
- Goals of JSF

# What is JSF?

- Standard Web Application Framework
  - Defines the Programming Model
  - Reusable Components
  - Builds HTML from Components
  - Validation & Conversion
  - Component Tree

# Goals of JSF

- Ease/Simplify Development
- Support from/by tools
- Events easily tied to server side code
- Ease of extension
- Manage UI state

# Agenda

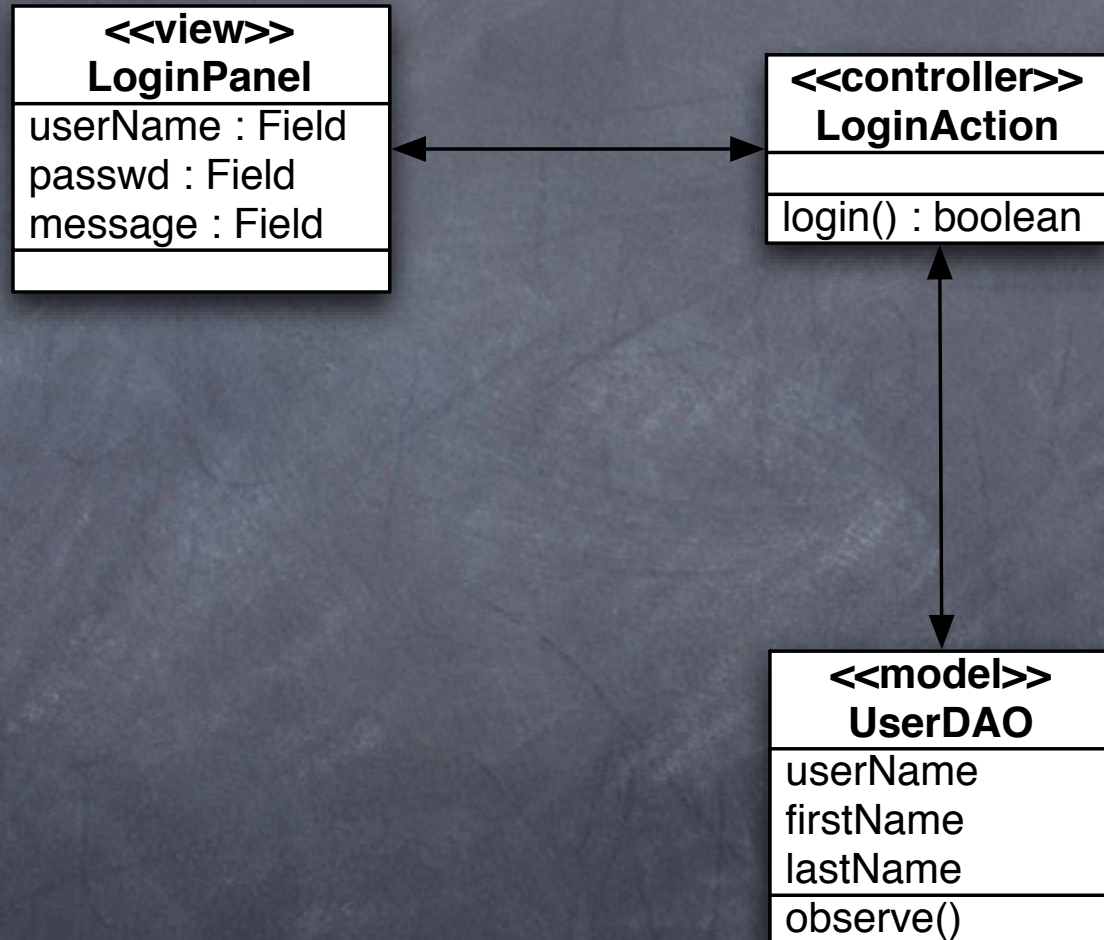
- JSF Overview
- Patterns & Architecture of JSF
- JSF Request/Response Lifecycle
- Components
- Validation & Conversion
- Events & Listeners
- Renderers
- MyFaces Extras

# Patterns & Architecture of JSF

- Model-View-Controller
- Factory Based
- Composite Component Tree
- Server Side Events

# Classic MVC

- Login
- Panel
- Action
- DAO



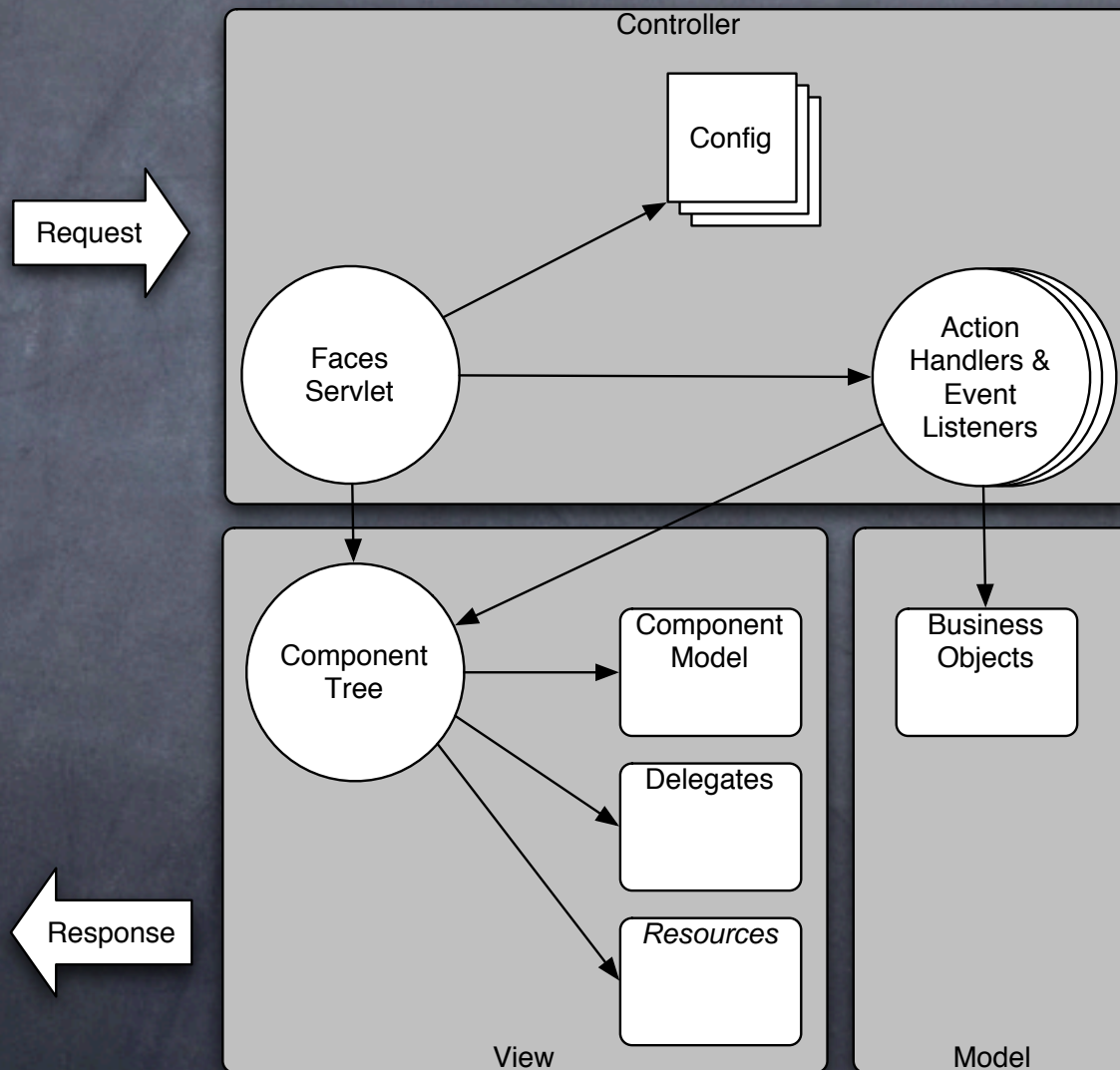
# Classic MVC

- View, Model & Controller Loosely coupled
- View responds by invoking Controller
- Controller responds by updating Model
- Model responds by posting events
- Other Controllers & View Observe events

# No Dice on the Web

- Browsers don't listen to server events
- What objects would observe models?
- So what to do to get the benefits?

# Enter JSF Web MVC



# JSF Provides Web MVC

- Controller
  - FacesServlet
  - LifeCycle
  - Application
  - NavigationHandler
  - ActionListener
  - Conversion and Validation

# JSF Provides Web MVC

- View

  - RenderKit

  - Renderers

- Model

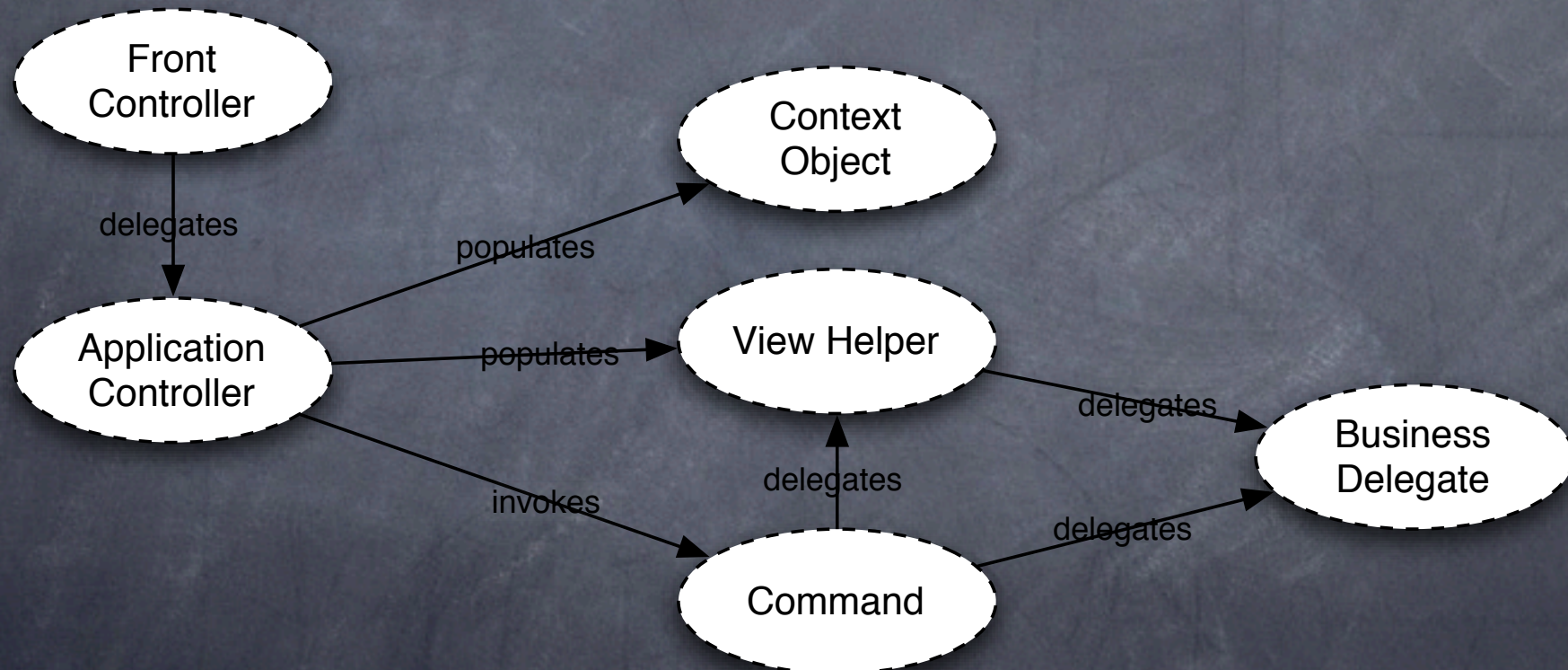
  - JavaBeans

  - EJB's

- Extension Points

  - View & Controller are extensible

# JSF & J2EE Patterns



# Factory Based

- FactoryFinder

  - A factory for factories

  - Deploy time configurable

- Factories

  - Application

  - FacesContext

  - Lifecycle

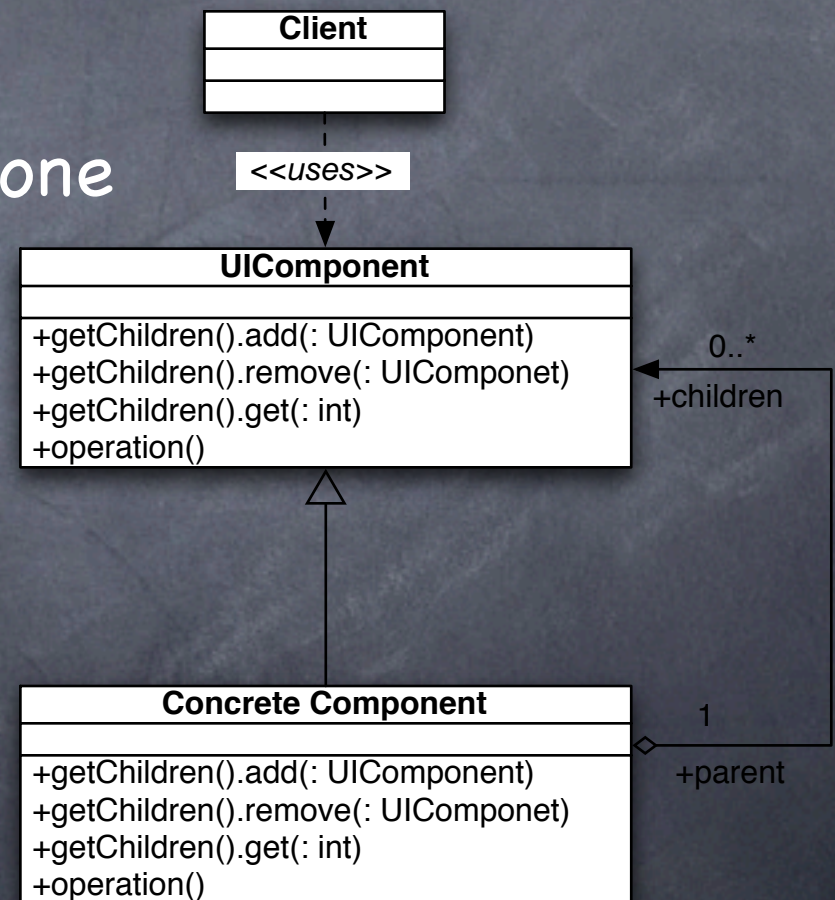
  - RenderKit

# Factory Configuration

- Implementation Provided Defaults
- Easily Overridden
  - Configuration files
  - Full implementation
  - Extend & Delegate

# Composite Component Tree

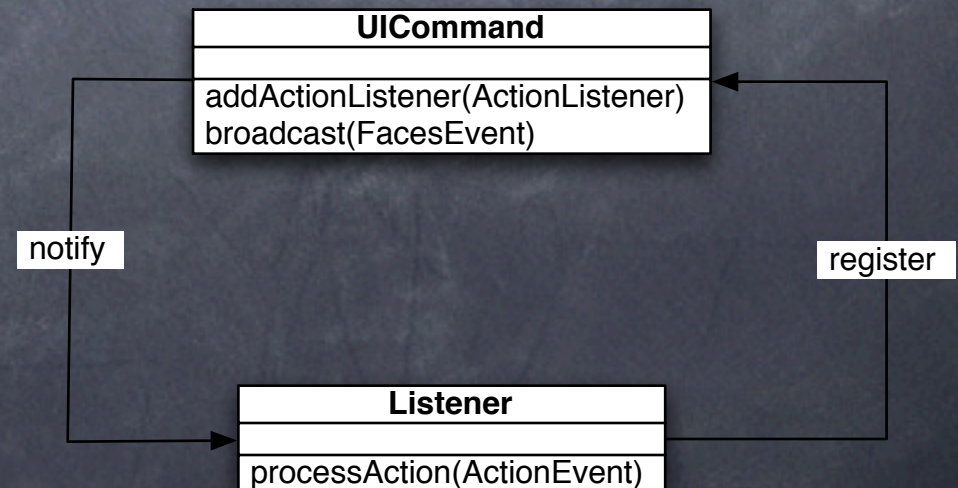
- Client view is simple
  - Whole tree looks like one
- Implementation
  - Simplified



# Server Side Events

- Events are cued & consumed server side
  - Typically from user actions (button clicks etc.)
  - Possible to add programatically

## • Observer Pattern

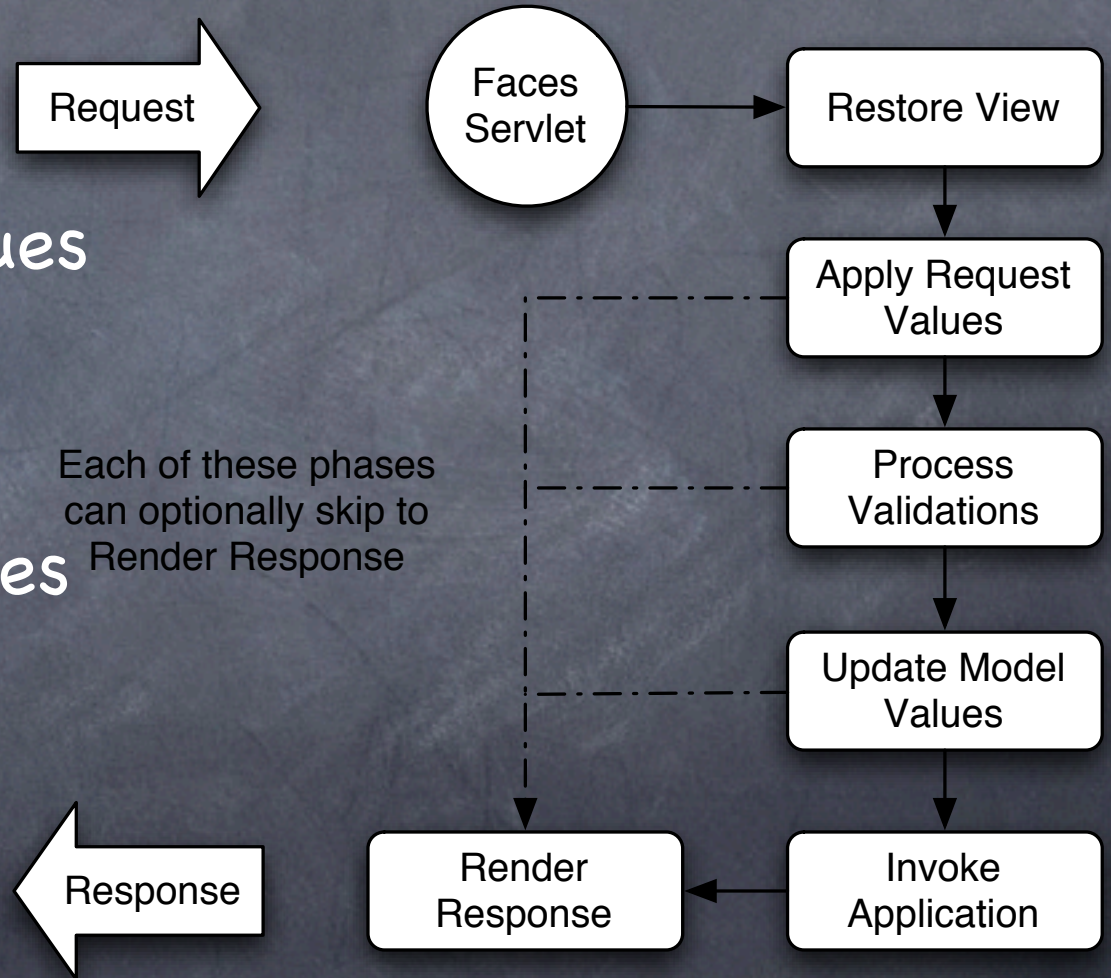


# Agenda

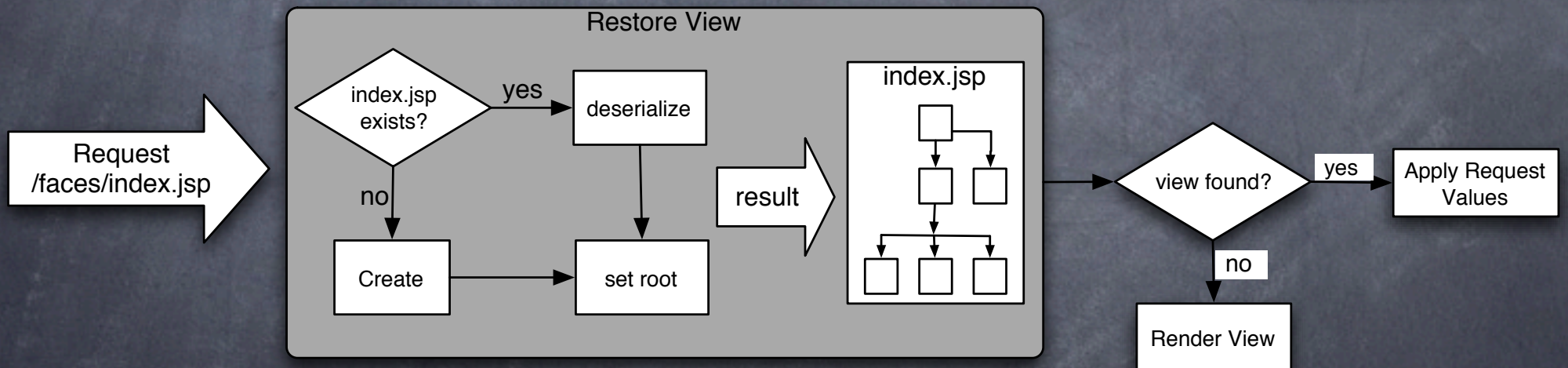
- JSF Overview
- Patterns & Architecture of JSF
- JSF Request/Response Lifecycle
- Components
- Validation & Conversion
- Events & Listeners
- Renderers
- MyFaces Extras

# JSF Request/Response Lifecycle

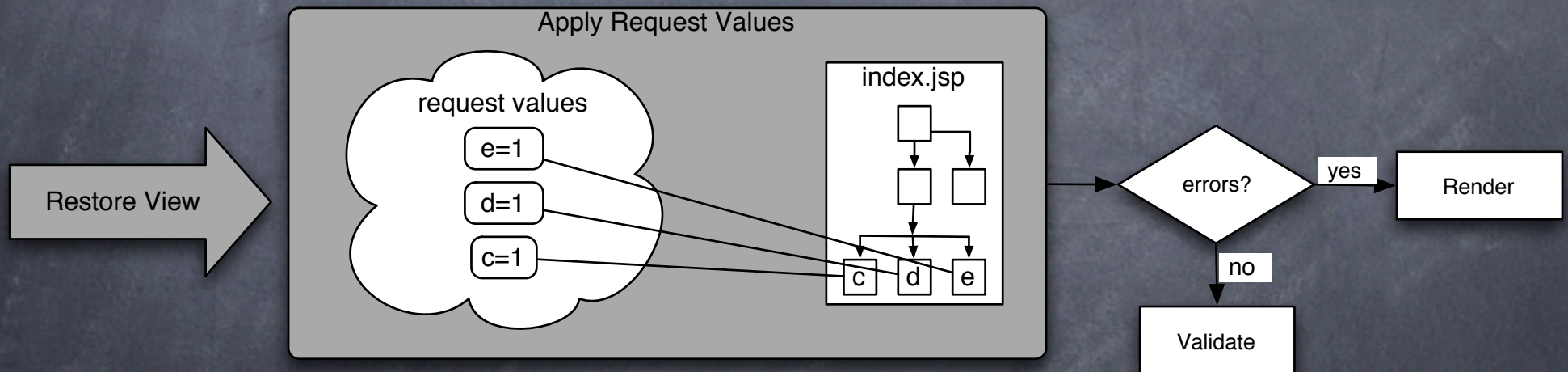
- Restore View
- Apply Request Values
- Process Validations
- Update Model Values
- Invoke Application
- Render Response



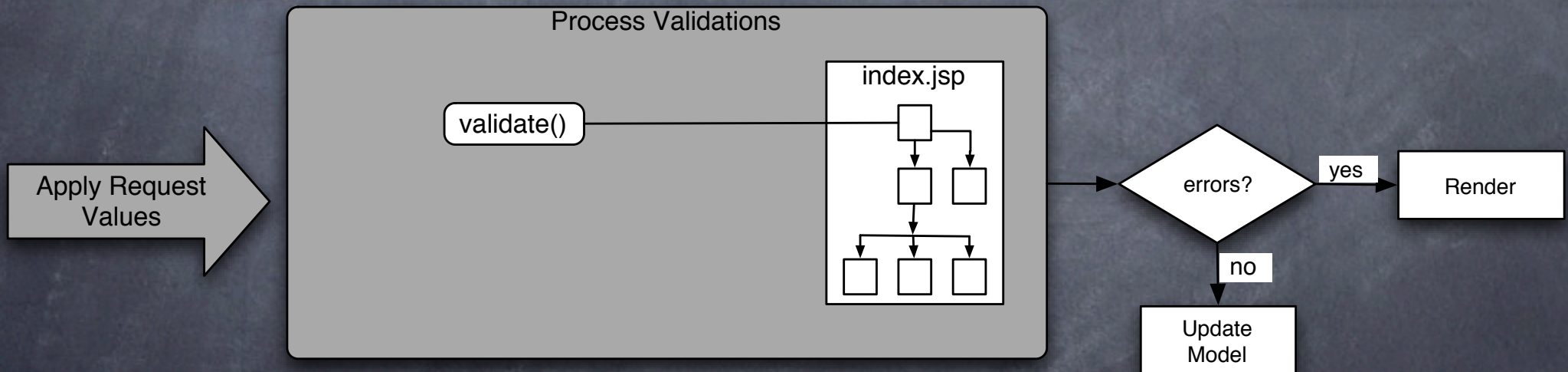
# Restore View



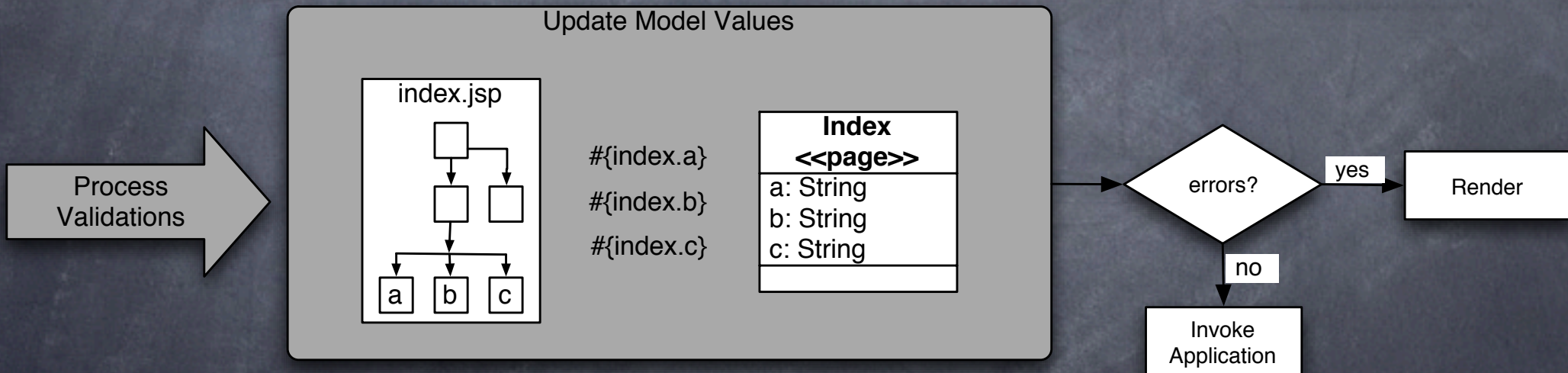
# Apply Request Values



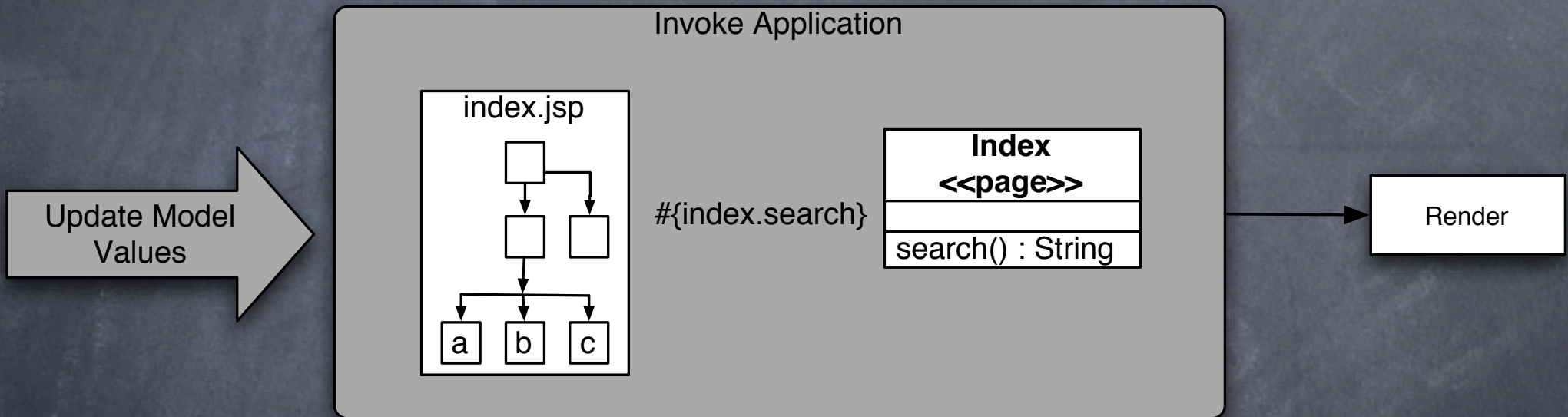
# Process Validations



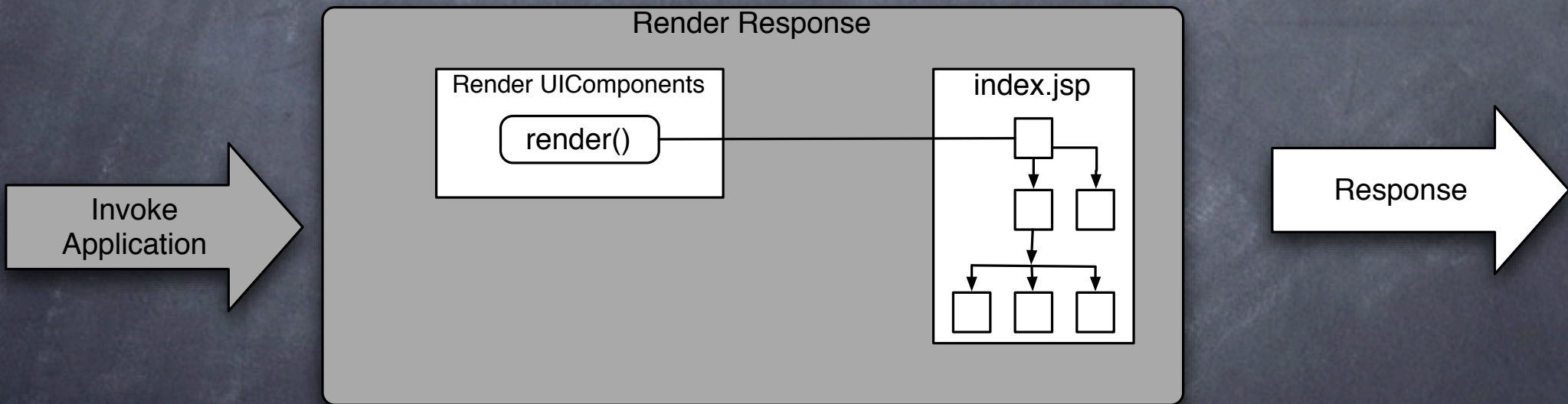
# Update Model Values



# Invoke Application



# Render Response



# Agenda

- JSF Overview
- Patterns & Architecture of JSF
- JSF Request/Response Lifecycle
- Components
- Validation & Conversion
- Events & Listeners
- Renderers
- MyFaces Extras

# UIComponent

- Hub of the framework
- Typically Delegate
  - Rendering
  - Validating
  - Converting

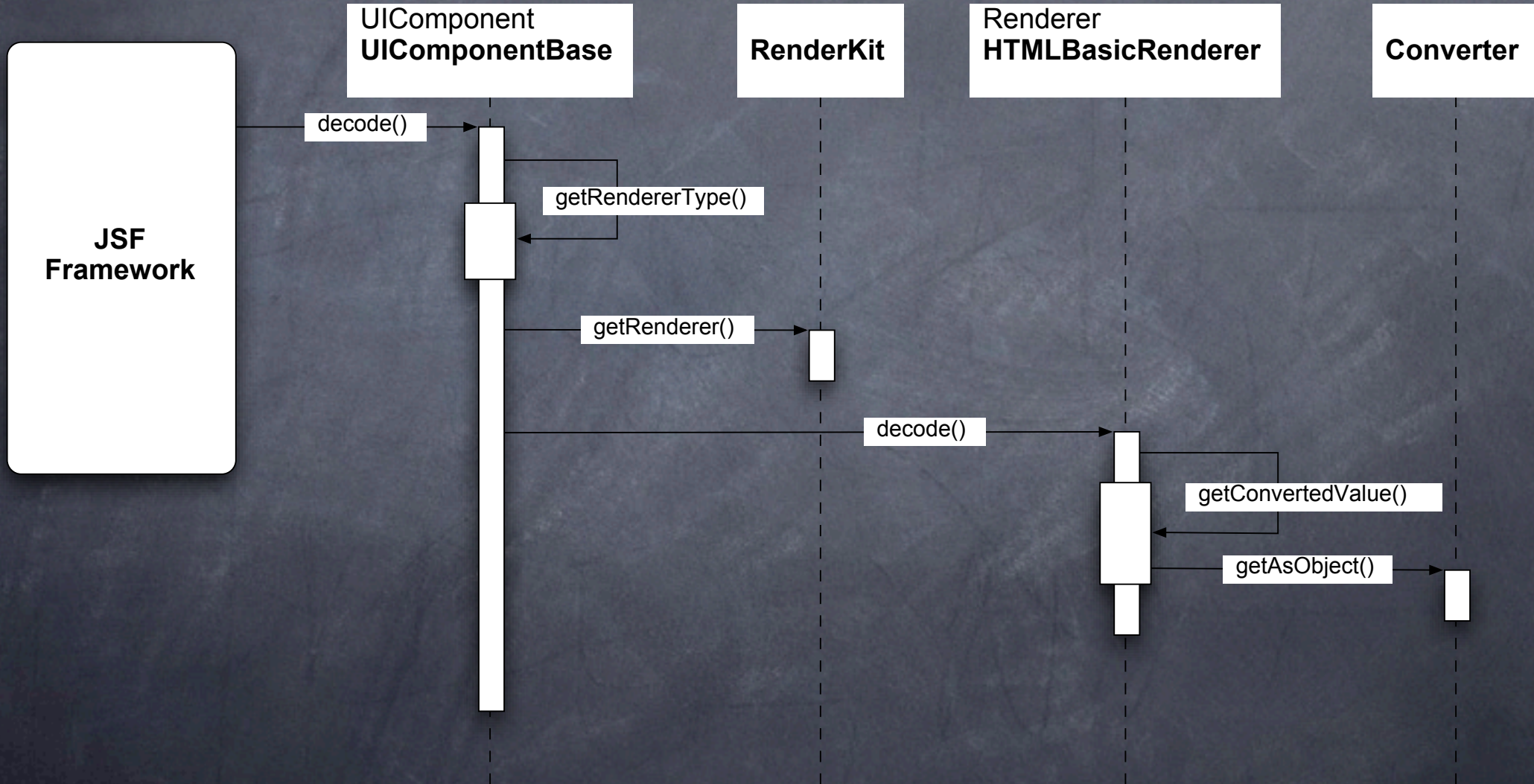
# Agenda

- JSF Overview
- Patterns & Architecture of JSF
- JSF Request/Response Lifecycle
- Components
- Conversion & Validation
- Events & Listeners
- Renderers
- MyFaces Extras

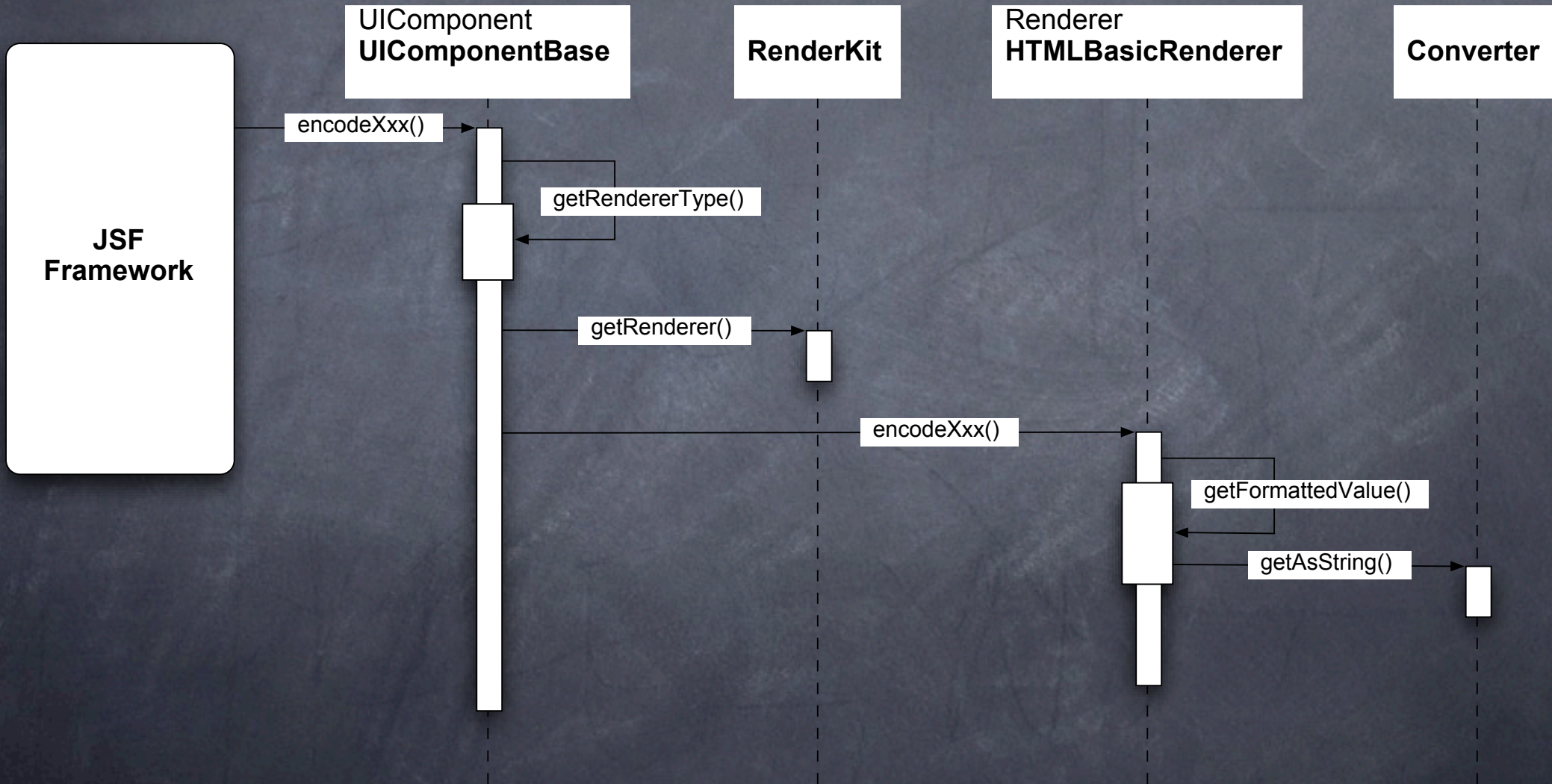
# Conversion

- Process Strings to Objects and back
  - Process Validations
  - Apply Request Values
- Request
  - Convert String to Object
- Response
  - Convert Object to String

# Conversion - Request



# Conversion - Response



# Validation

- Semantic Validation
  - Is the value in the right range
  - Is the value the proper length
  - Are required values set
- Extensible
  - method binding
  - implement an interface

# Agenda

- JSF Overview
- Patterns & Architecture of JSF
- JSF Request/Response Lifecycle
- Components
- Validation & Conversion
- Events & Listeners
- Renderers
- MyFaces Extras

# Events & Listeners

- Server Side
  - Posted by decoding of UICommands
  - Other processing
- Listened to by your classes
- Augmented by `addListener` method bindings

# Agenda

- JSF Overview
- Patterns & Architecture of JSF
- JSF Request/Response Lifecycle
- Components
- Validation & Conversion
- Events & Listeners
- Renderers
- MyFaces Extras

# Renderers

- Builds the output for the client
  - Easy to replace with your own implementation
  - Customizable
  - Overridable

# Rendering

- Creates client side view of component tree
- Managed by the component
  - Most cases delegated to renderers
- Responsible for decoding client response
  - Take request parameters and put them into the correct component

# Agenda

- JSF Overview
- Patterns & Architecture of JSF
- JSF Request/Response Lifecycle
- Components
- Validation & Conversion
- Events & Listeners
- Renderers
- MyFaces Extras

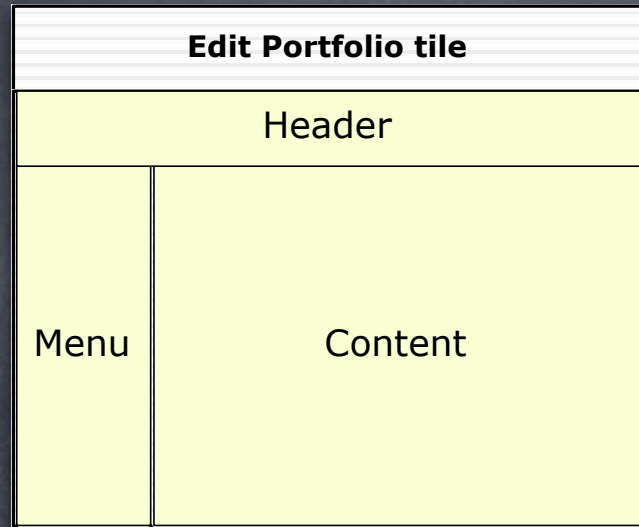
# Components

- DataScroller
- SortHeader
- Calendar
- File Upload

# Validators

- Email
- Credit Card
- Regular Expression

# Tiles



Welcome to the Edit Portfolio Page

The "Content" configuration dialog box contains the following elements:

- Name:** A text input field containing the value "Main".
- Positions:** A section containing a list box with the title "Symbol".
- Symbol List:** A list box with three items: "MSFT", "INTC", and "AAPL". The "INTC" item is currently selected and highlighted in blue. The list box has a vertical scrollbar and arrow buttons at the top and bottom.
- Buttons:** Three buttons at the bottom: "Cancel", "Remove", and "Save".

- Item 1
- Item 2
- Item 3

# Configuration

```
<application>
```

```
  <view-handler>...JspTilesViewHandlerImpl</view-handler>
```

```
</application>
```

```
<context-param>
```

```
  <param-name>tiles-definitions</param-name>
```

```
  <param-value>/WEB-INF/tiles.xml</param-value>
```

```
</context-param>
```

# Q & A

- Fill out evals