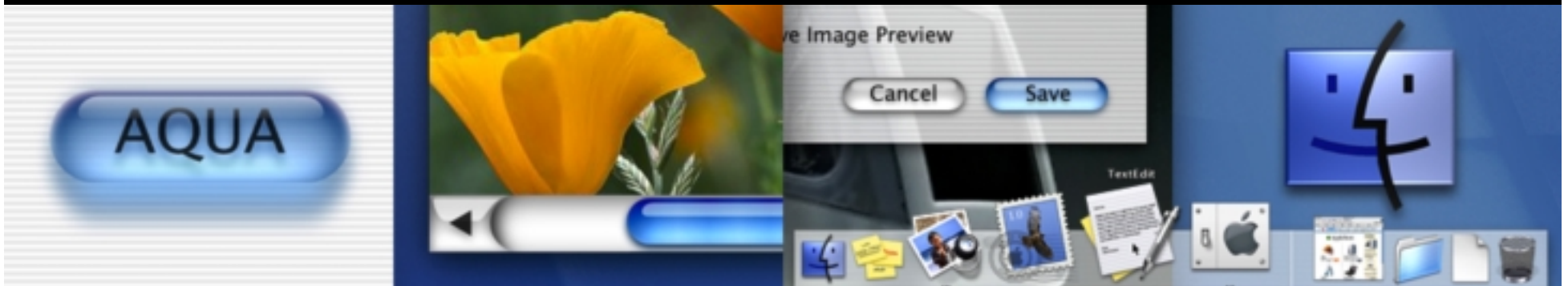




Session 402

WebObjects Builder: Dynamic HTML Components



James Dempsey
Senior Developer Trainer, Apple iServices

Introduction

- Introduce components – the heart of any WebObjects application
- Introduce WebObjects Builder – the tool used to edit components
- Tour new features in version 4.5 of WebObjects Builder



What You'll Learn

- How components interact with users
- How to build a basic component
- New features in WebObjects Builder 4.5



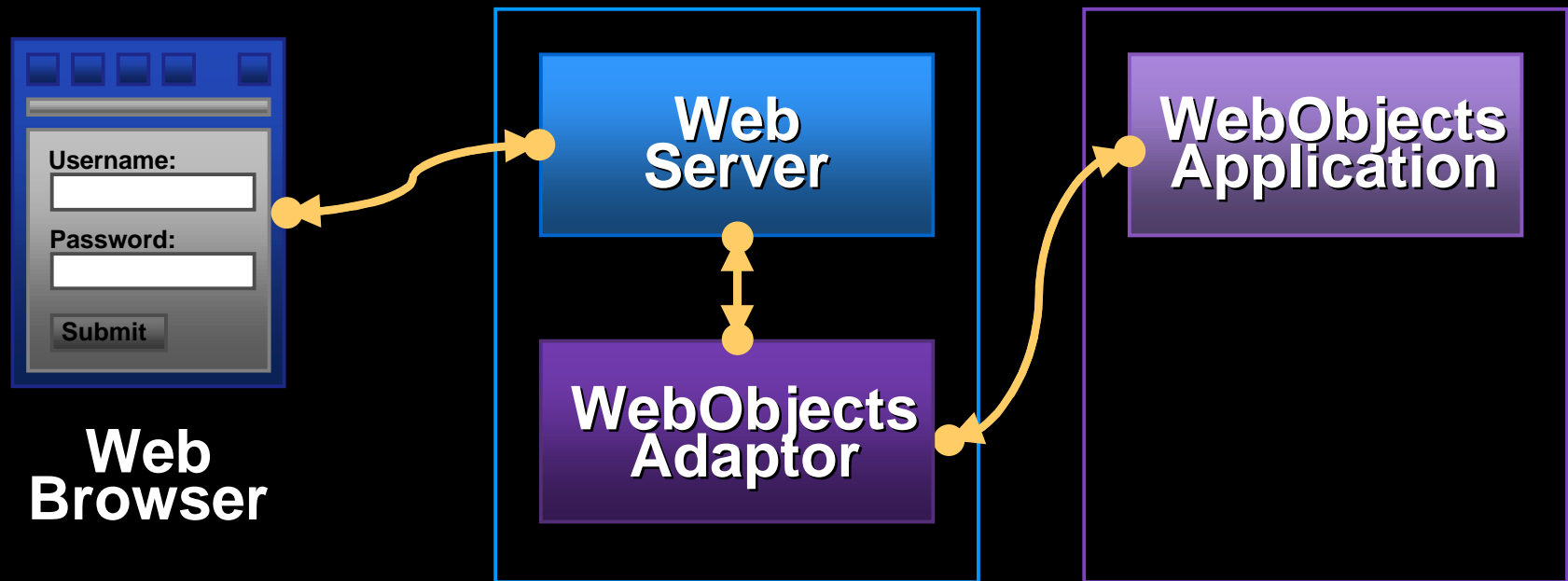
Request/Response Loop Review



**Web
Browser**



Request/Response Loop Review



Request/Response Loop Review

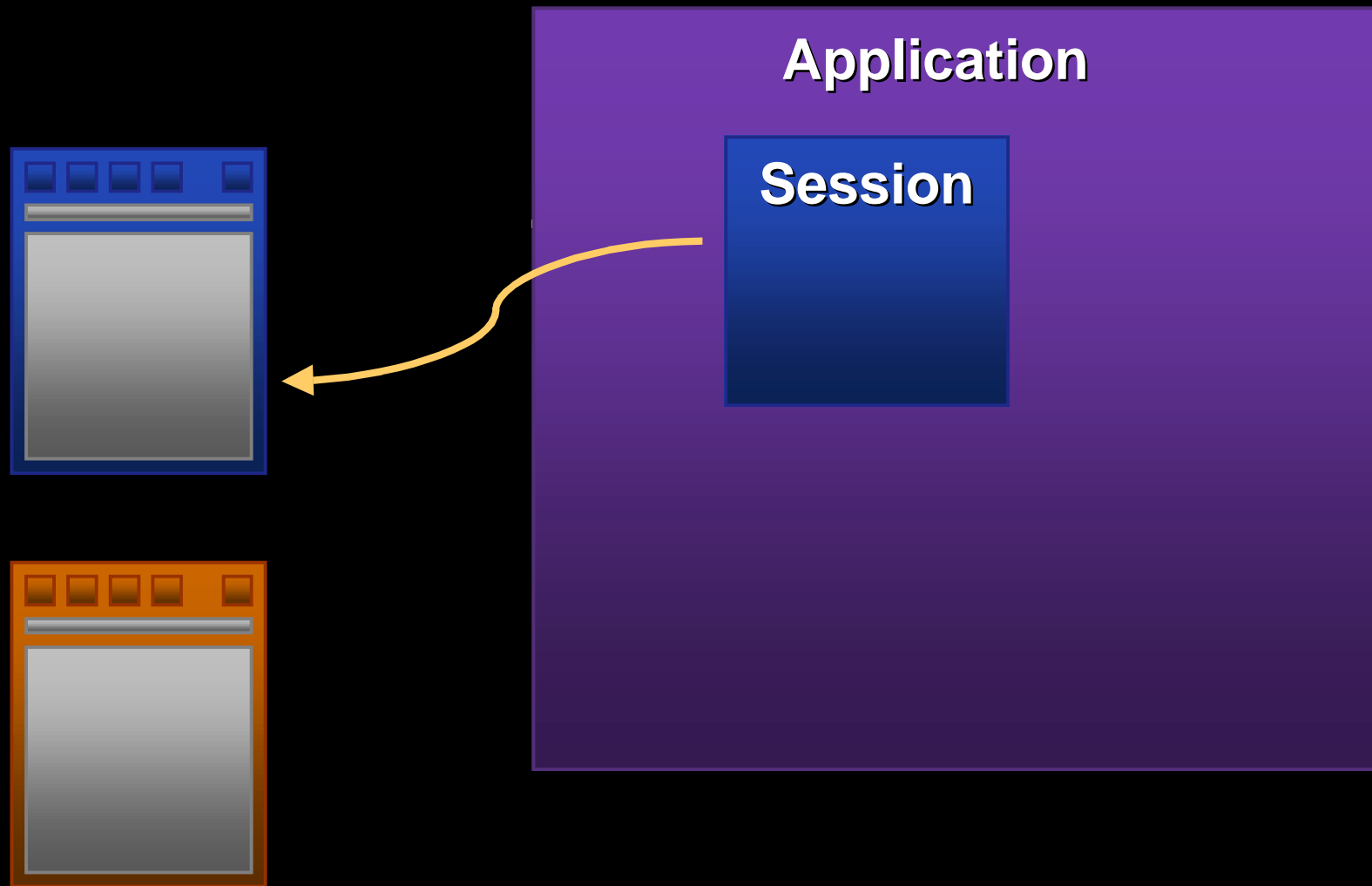


**Web
Browser**

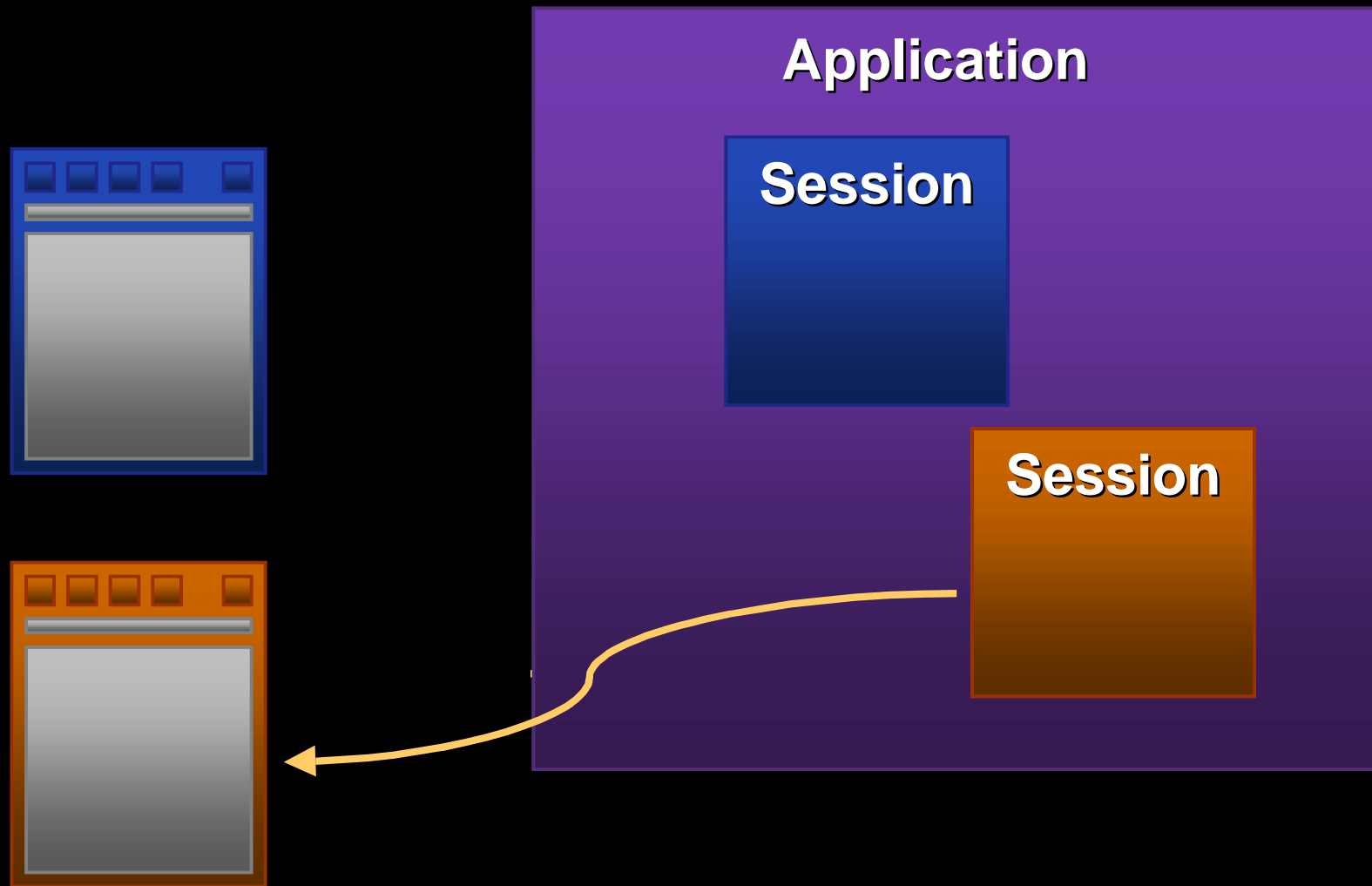
Application



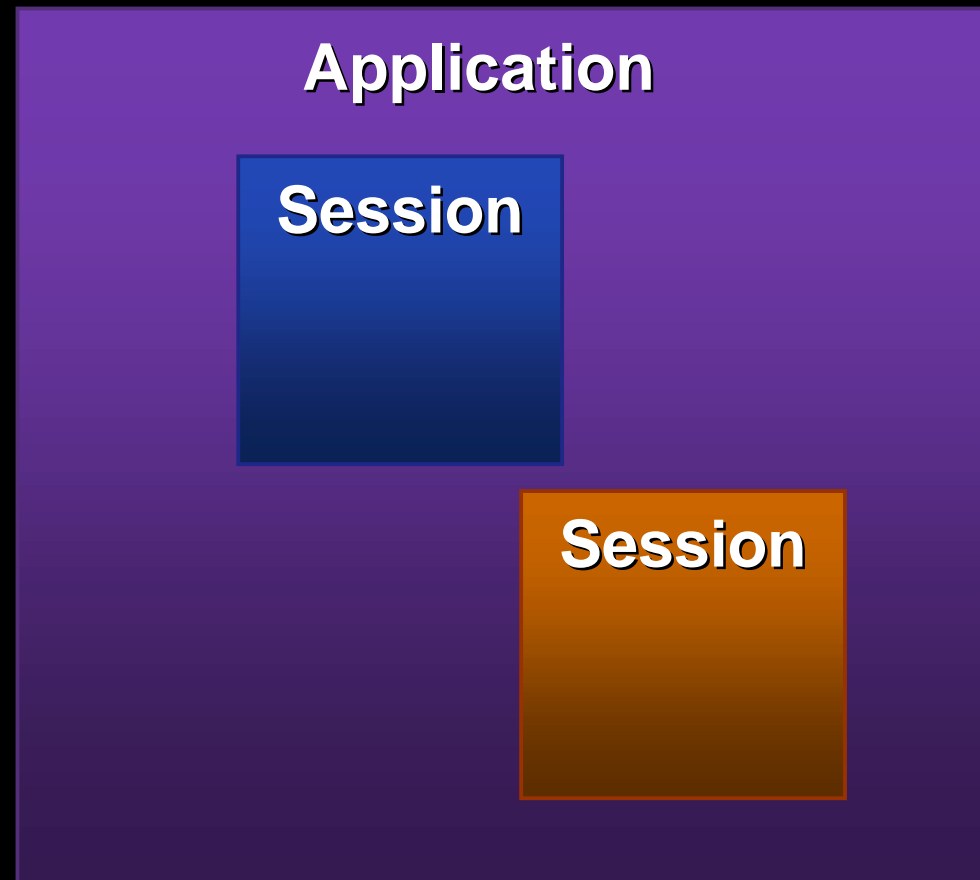
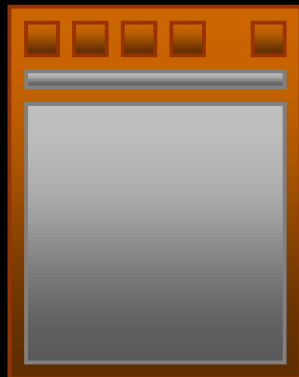
Request/Response Loop Review



Request/Response Loop Review

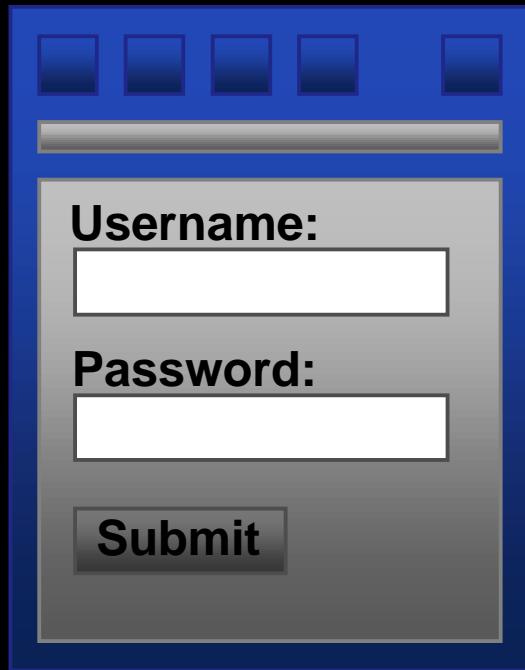


Request/Response Loop Review



Components Interact With the User

Dynamically generates HTML



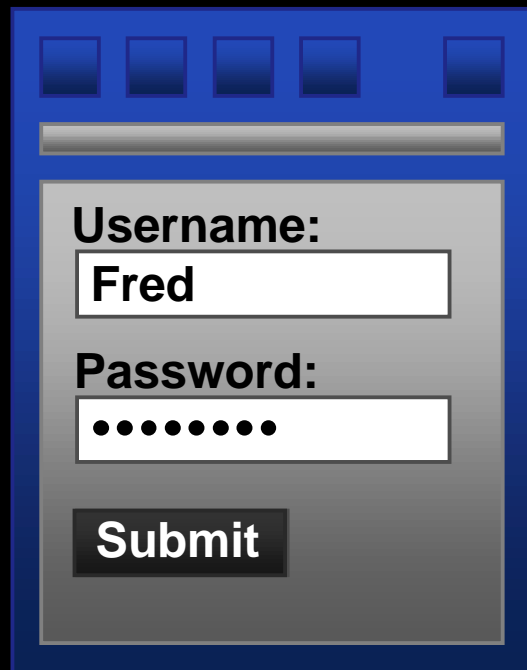
A screenshot of a web browser window with a blue title bar. The window contains a login form with a light gray background. The form has two input fields: one labeled "Username:" and one labeled "Password:". Below the input fields is a "Submit" button. The browser window also shows five small square icons in the top-left corner and a scroll bar below the title bar.

Component



Components Interact With the User

User interacts with HTML page



A browser window with a blue title bar and five window control buttons. The main content area is a light gray rectangle containing a login form. The form has two text input fields: the first is labeled "Username:" and contains the text "Fred"; the second is labeled "Password:" and contains seven black dots. Below the password field is a dark gray button with the text "Submit" in white.

Request

Component



Components Interact With the User

Takes form values from the request

Username:
Fred

Password:
.....

Submit

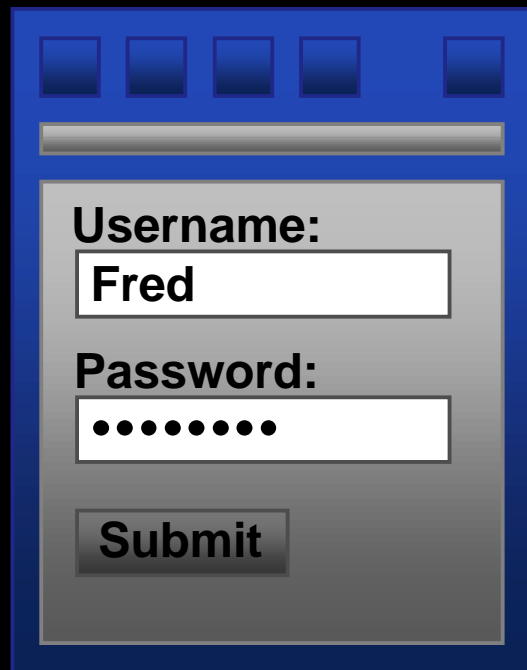
Request

Component



Components Interact With the User

Invokes method: take custom action



Username:
Fred

Password:
.....

Submit

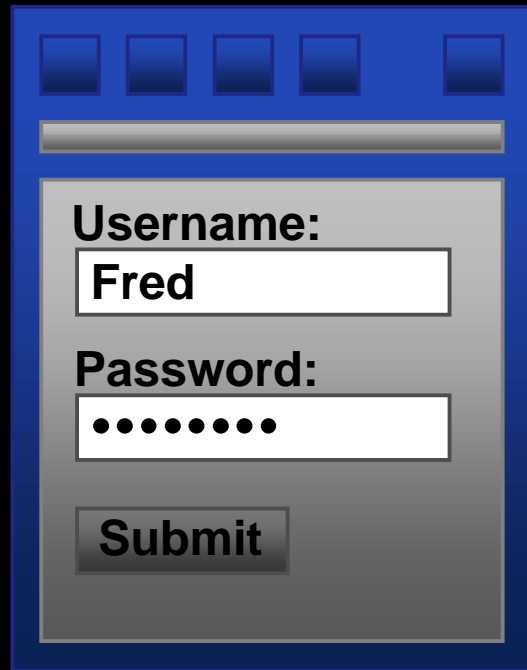
Request

Component



Components Interact With the User

Invokes method: returns next component



Username:
Fred

Password:
•••••••

Submit

Component



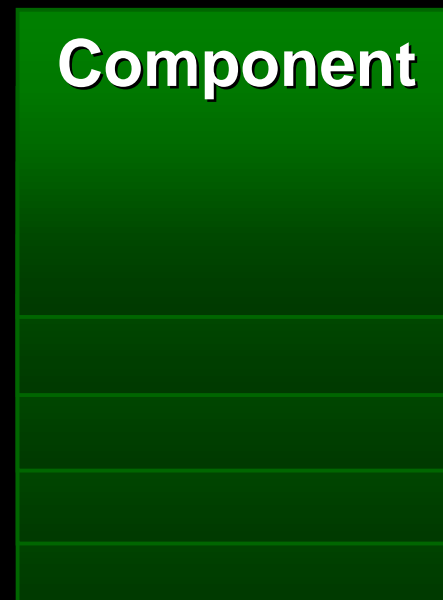
Components Interact With the User

Next component interacts with user



A blue window with a title bar containing five window control buttons. Below the title bar is a horizontal scrollbar. The main content area is a light gray rectangle containing the following elements:

- Username:** A text input field containing the text "Fred".
- Password:** A text input field containing seven black dots.
- Submit**: A button with the text "Submit".



Components Interact With the User

Next component interacts with user



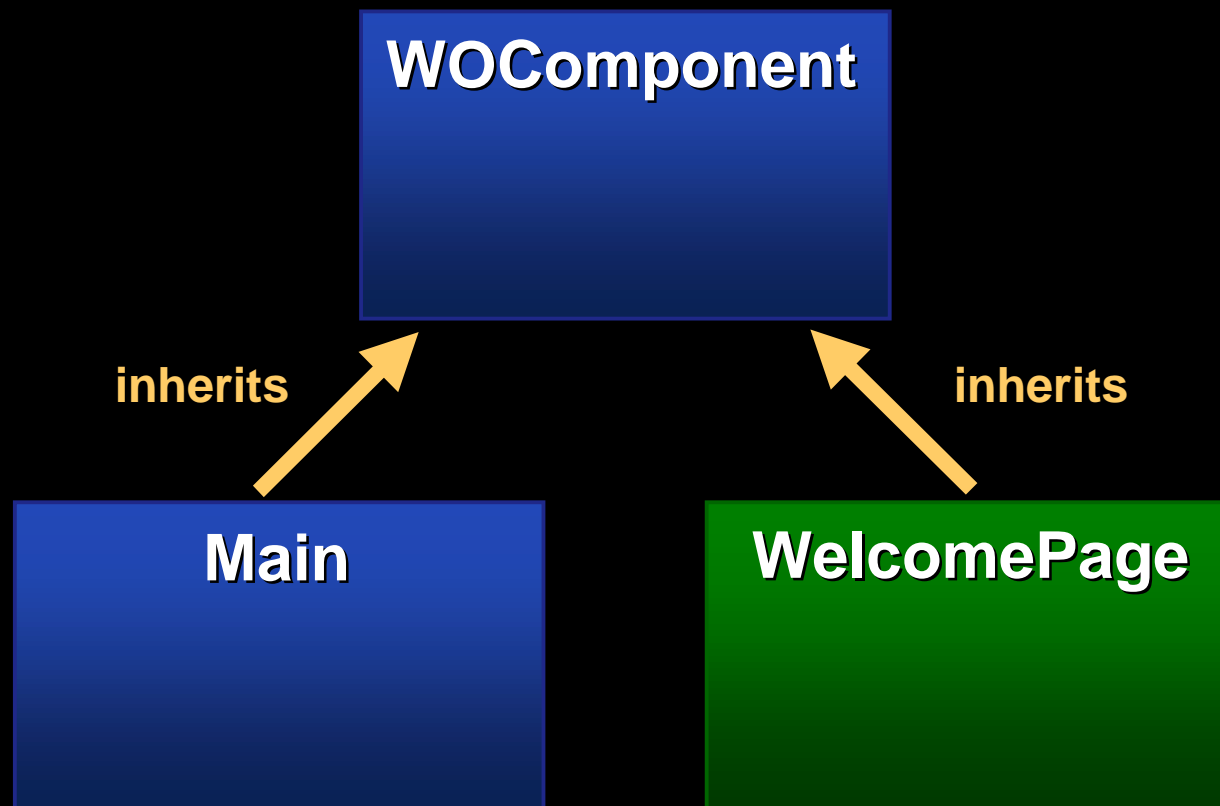
Component Phases

- Generate HTML into response
appendToResponse()
- Take form values from incoming request
takeValuesFromRequest()
- Invoke action method
 - Custom processing
 - Return next component to interact with user
invokeAction()



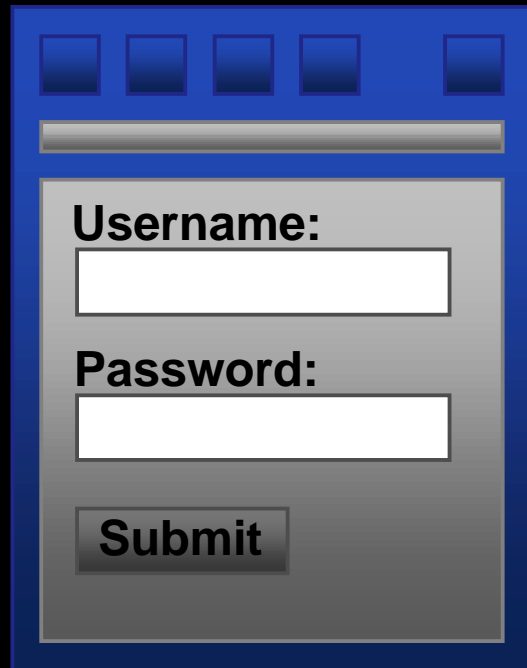
WOComponent

Your components subclass WOComponent



A Quick Review

`appendToResponse()`

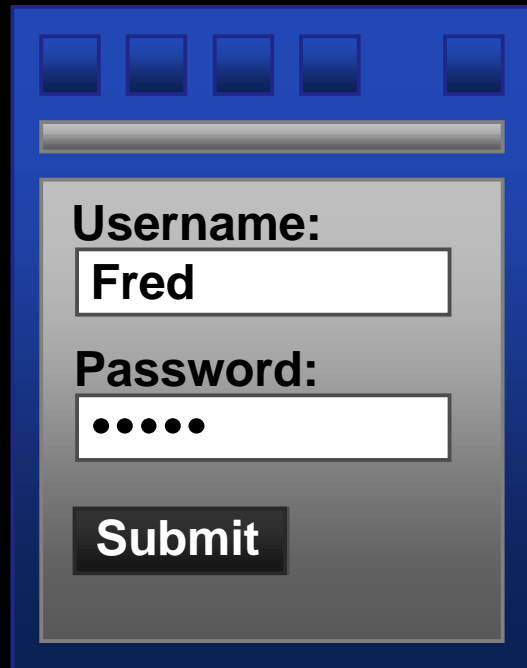


A screenshot of a web browser window with a blue title bar and five window control buttons. The main content area has a light gray background and contains a login form. The form has two text input fields: the first is labeled "Username:" and the second is labeled "Password:". Below the input fields is a "Submit" button.



A Quick Review

User interacts with HTML page



A browser window with a blue title bar and five window control buttons. The main content area is a light gray rectangle containing a login form. The form has two text input fields: the first is labeled "Username:" and contains the text "Fred"; the second is labeled "Password:" and contains five black dots. Below the password field is a dark gray button with the text "Submit" in white.

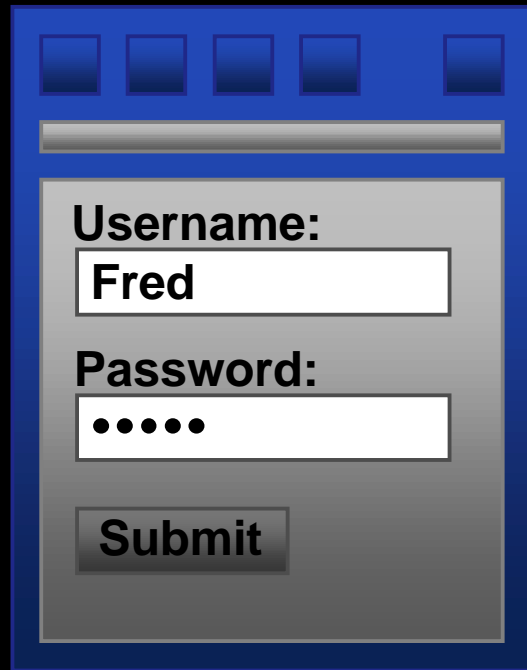
Request

Main



A Quick Review

`takeValuesFromRequest()`



Username:
Fred

Password:
.....

Submit

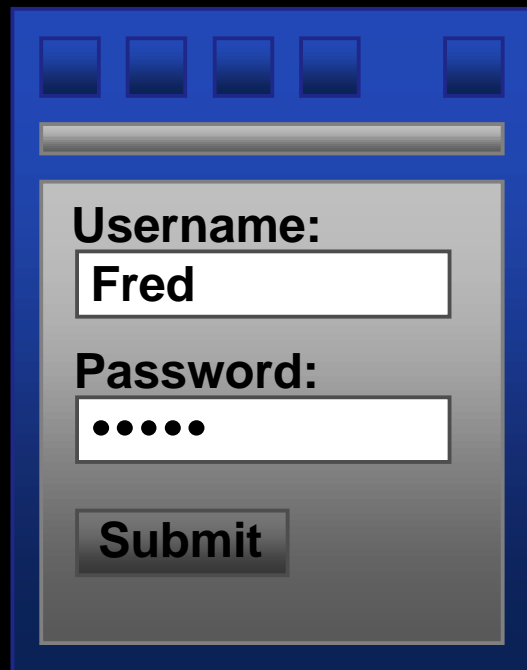
Request

Main



A Quick Review

invokeAction()



Username:
Fred

Password:
.....

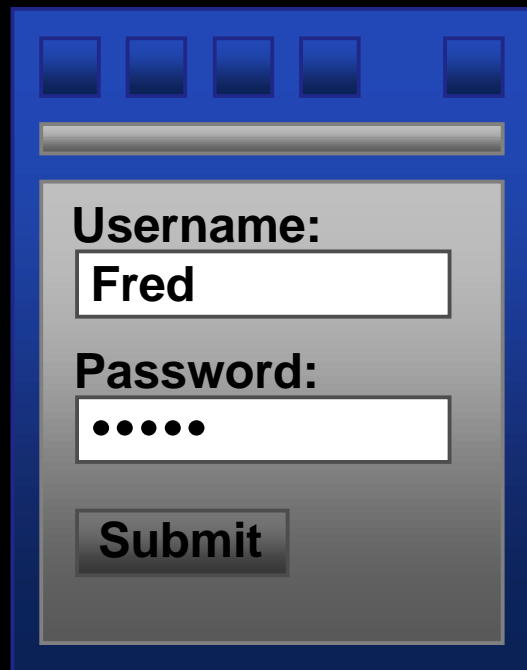
Submit

Request



A Quick Review

Same component instance used again



Username:
Fred

Password:
.....

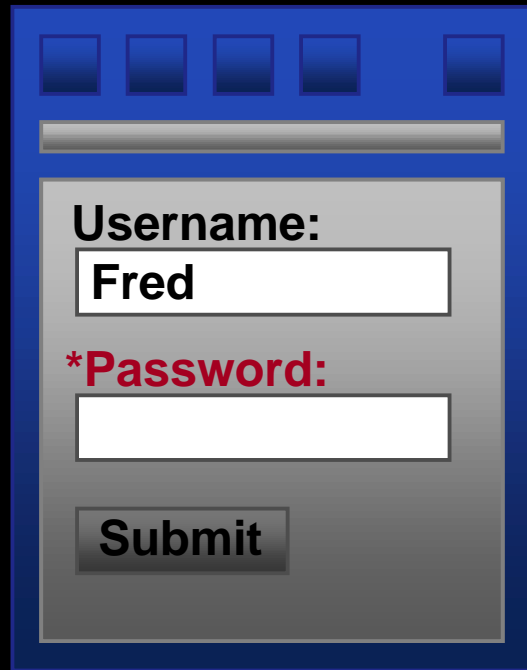
Submit

Request



A Quick Review

`appendToResponse()`



Username:
Fred

*Password:

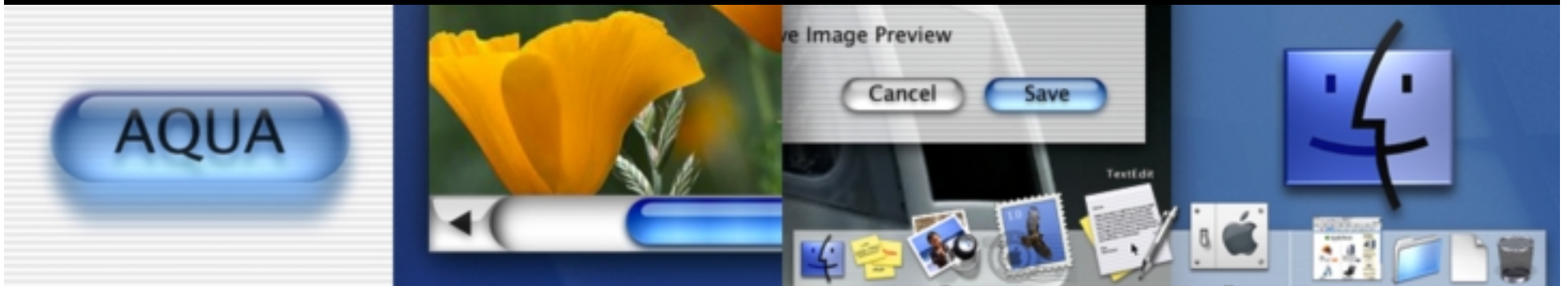
Submit

Main





DEMO



What We Didn't Need to Do

- Route requests to user session and component
- Process the incoming request
- Parse form values in incoming request
- Convert form values to objects
- Manually manage state for each user
- Create and fill the HTTP response



Component HTML and Bindings Files

HTML

```
...<BODY>
  <H1> Welcome
  <WEBOBJECT Name=String1> </WEBOBJECT>
</H1> <BR>
  <WEBOBJECT Name=Hyperlink1>
  Return Home </WEBOBJECT>
</BODY>...
```

WOD

```
String1 : WOString {
  value = username;
}
Hyperlink1 : WOHyperlink {
  action = login;
}
```



Component HTML and Bindings Files

HTML

```
...<BODY>
  <H1> Welcome
  <WEBOBJECT Name=String1> </WEBOBJECT>
  </H1> <BR>
  <WEBOBJECT Name=Hyperlink1>
  Return Home </WEBOBJECT>
</BODY>...
```

WOD

```
String1 : WOString {
  value = username;
}
Hyperlink1 : WOHyperlink {
  action = login;
}
```



Component HTML and Bindings Files

HTML

```
...<BODY>
  <H1> Welcome
  <WEBOBJECT Name=String1> </WEBOBJECT>
  </H1> <BR>
  <WEBOBJECT Name=Hyperlink1>
  Return Home </WEBOBJECT>
</BODY>...
```

WOD

```
String1 : WOString {
  value = username;
}
Hyperlink1 : WOHyperlink {
  action = login;
}
```



Components Have an HTML Template

Dynamic objects in a page of static HTML

...<BODY>

<H1> Welcome **WOString**

</H1>

WOHyperlink **Return Home**

</BODY>...



Generating HTML

appendToResponse()

...<BODY>

<H1> Welcome

WOString

</H1>

WOHyperlink

Return Home

</BODY>...



Generating HTML

appendToResponse()

...<BODY>

<H1> Welcome **WOString**

</H1>

WOHyperlink **Return Home**

</BODY>...



Generating HTML

appendToResponse()

...<BODY>

<H1> Welcome

WOString

</H1>

WOHyperlink

Return Home

</BODY>...



Generating HTML

appendToResponse()

...<BODY>

<H1> Welcome

WOString

</H1>

WOHyperlink

Return Home

</BODY>...



Generating HTML

appendToResponse()

...<BODY>

<H1> Welcome

WOString

</H1>

WOHyperlink

Return Home

</BODY>...



Generating HTML

appendToResponse()

...<BODY>

<H1> Welcome

WOString

</H1>

WOHyperlink

Return Home

</BODY>...



Generating HTML

appendToResponse()

...<BODY>

<H1> Welcome

WOString

</H1>

WOHyperlink

Return Home

</BODY>...



Generating HTML

`appendToResponse()`

Fred

NSString

WelcomePage

username "Fred"

`valueForKey("username")`

"Fred"



Taking Form Values From the Request

takeValuesFromRequest()

WForm

Username: **WOTextField** **
**

Password: **WOTextField** **
**

WOSubmitButton



Taking Form Values From the Request

takeValuesFromRequest()

Request

WOTextField

takeValueForKey("Fred", "username")

Main

username	"Fred"
password	null

login



Invoking an Action Method

invokeAction()

WOForm

Username: **WOTextField** **
**

Password: **WOTextField** **
**

WOSubmitButton



Invoking an Action Method

invokeAction()

Request

WOSubmitButton

Main

username	"Fred"
password	null

login ○ ○ ○ ○ ○ ○

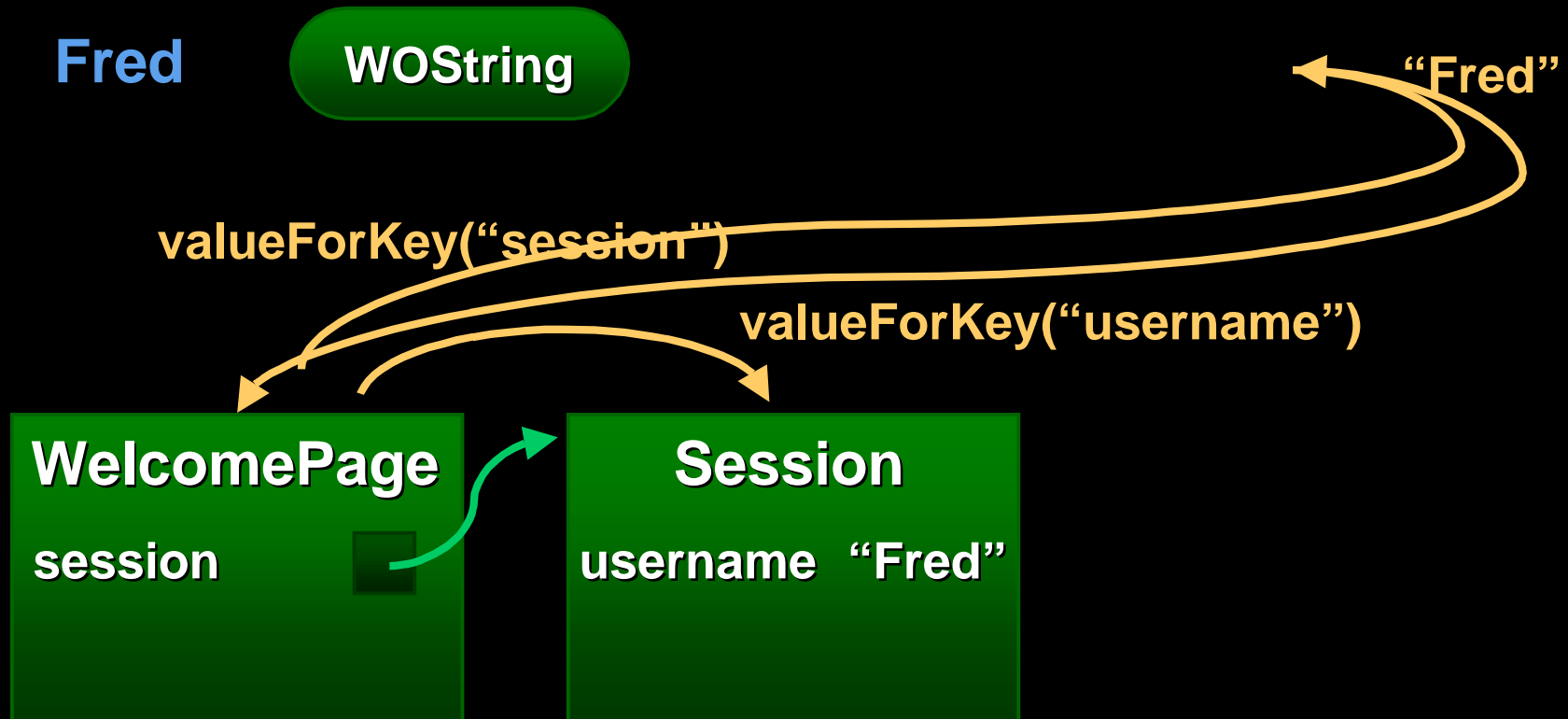
login()



Accessing Session Values

Fred

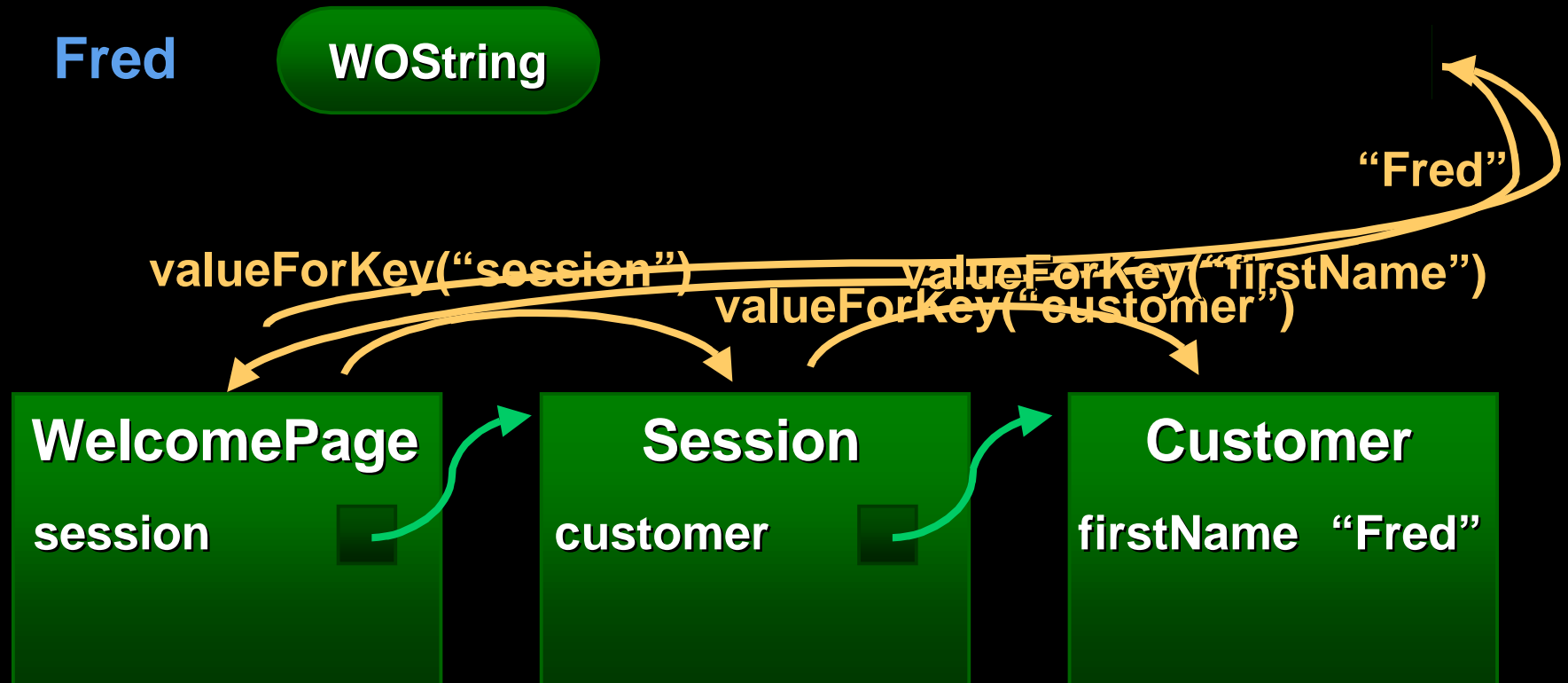
WOString



Arbitrary Key Paths

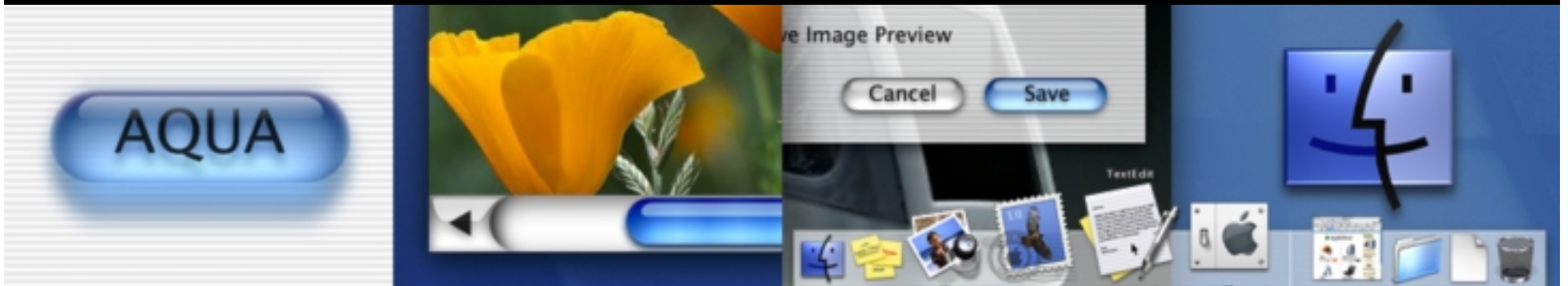
Fred

NSString





DEMO



WebObjects Builder 4.5

Focus on ease of development

- Expanded source file editing options
- Extensive contextual menu support
- Improved table editing
- More palettized objects
- Path view for HTML selection and manipulation
- Binding validation



WebObjects Builder 4.5

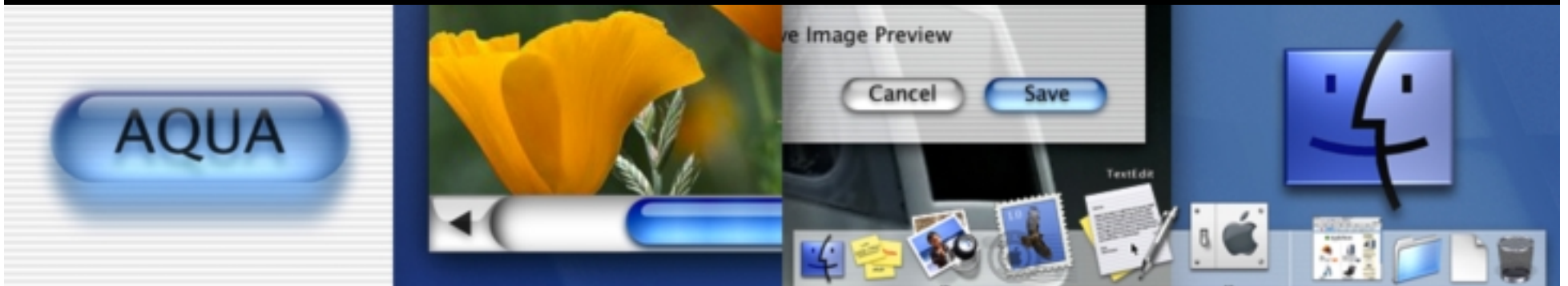
Focus on ease of development

- HTML syntax coloring and selection shortcuts
- Unified toolbar
- Improved binding user interface
- Improved font tag support
- Numerous new display options
- API editor for nested components





DEMO



Roadmap

403 Persistent Object Modeling with EOModeler

Database connectivity in WebObjects

Room J2
Tues., 3:00 p.m.

404 Deploying Applications

You've built your app. Now what?

Room J2
Tues., 4:30 p.m.

411 Integrating Client Side Technologies

Components: Not just HTML

Room J2
Thurs., 10:30 a.m.

416 Designing Reusable Components

Do the work once, reuse it everywhere

Room J2
Fri., 10:30 a.m.



For More Information

<http://www.apple.com/webobjects>

Visit the WebObjects lab downstairs!
Everyday from 11:00 a.m.–2:00 p.m.

Try out your WebObjects 4.5 Evaluation CD!

WebObjects Community BOF
Wed., 6:30 p.m.–8:00 p.m.



Who to Contact

Toni Trujillo Vian

Director, WebObjects Engineering
wofeedback@group.apple.com

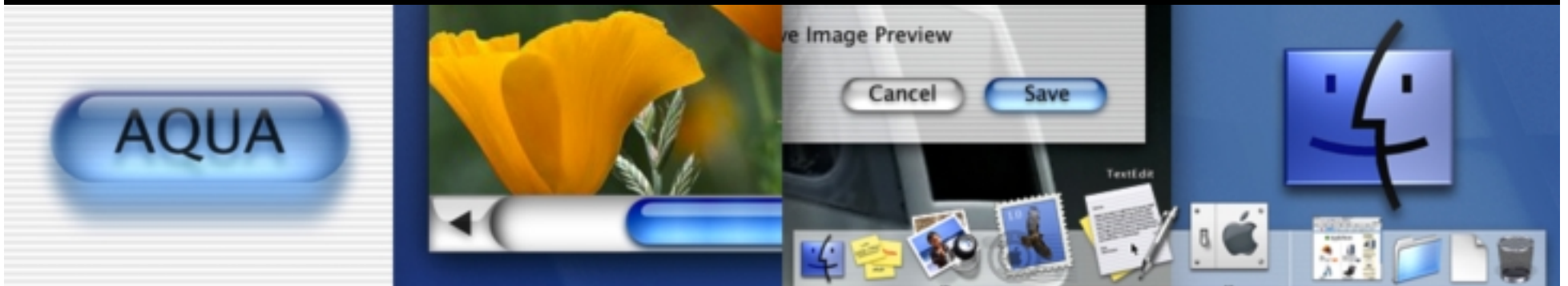
Ernest Prabhakar

Product Line Manager, WebObjects
wofeedback@group.apple.com





Q&A



Ben Haller, WebObjects Engineering
Tom Naughton, WebObjects Engineering



WWDC

Worldwide Developers Conference 2000



Think different.