



Web 2.0 and AJAX

Jim Driscoll

Manager, Java Web Tier

Agenda

- Definitions: Web 2.0, AJAX
 - > Wikis, RSS/Blogs, and REST
 - > AJAX Overview
- Guidelines
- JSF Approach
- AJAX BluePrints

A Little Web History

JGD

- CGI / Perl & C
- Servlets
- JSP / ASP / PHP
- Other scripting
 - > Python, Ruby
- Java based web frameworks
 - > Struts, JSF, (many) others
- Scripting based frameworks
- But all these kept the same (1.0) UI

Web 2.0 by example

JGD

Web 1.0



Web 2.0

BETA

- DoubleClick
- Ofoto
- Britannica Online
- mp3.com
- Personal websites
- Directories (Taxonomy)
- Screen Scraping
- Google AdSense
- Flickr
- Wikipedia
- Napster
- Blogging
- Tagging (Folksonomy)
- Web Services

Web 2.0 Definition – by Tim O'Reilly

JGD

- Web as a Platform
- Collective Intelligence
 - > Folksonomy – Collaborative Categorization
- Data is key and should be shared
- Software is in constantly evolving
 - > Software release cycles dead?
- Lightweight Programming Models
 - > SOAP/REST
- The Network is the computer
 - > (iTunes, mobile devices)
- Rich User Experience

So what is Web 2.0?

JGD

- Fuzzy Term, as popularized by O'Reilly...
 - > ... some negative reaction ("obvious marketing fluff")
 - > ... but trying to capture a real qualitative change
 - > ... featured recently in Time and Business Week
- Technologies
 - > Blogging, Syndication, RSS/Atom
 - > Wikis, Web Services (REST)
 - > AJAX, Rich Internet Clients
- Attitudes
 - > Sharing, Connected, Participatory
 - > Services, Perpetual Beta, Users Engaged

What is Web 2.0? (cont.)

JGD

- Services
 - > Flickr, BitTorrent, iTunes
 - > Maps (Yahoo, Google), Wikipedia
 - > Gmail, AdSense
 - > Yahoo & Google Services
- Drivers
 - > Faster Connectivity
 - > More Available Connectivity, esp at home
 - > More Powerful Machines
 - > Customers More Comfortable with Technology
 - > Browser wars (mostly) over

Blogs / RSS / Atom

JGD

- RSS – Really Simple Syndication
 - > A number of (not fully compatible) specs
 - > Atom is latest, IETF, Standard
- Provide Syndicated Information through HTTP
- Blogs build on RSS/Atom
 - > Aggregation, Content Reuse, Caching
- Strong Social Phenomenon (e.g. politics)
- Rome – a popular RSS/Atom library
- Roller – Apache project donated by Sun
 - > Runs on AS 9
- Blogs.sun.com

Wikis / Collaboration

- Wikis are...
 - > Simplified Web sites (Management, Content)
 - > Collaboratively Created Web Sites
- Example: Wikipedia (uses MediaWiki)
- Java Wiki: JSPWiki (many others)
- Portal Server 7.0
 - > Uses JSP Wiki
 - > Focused on Collaboration

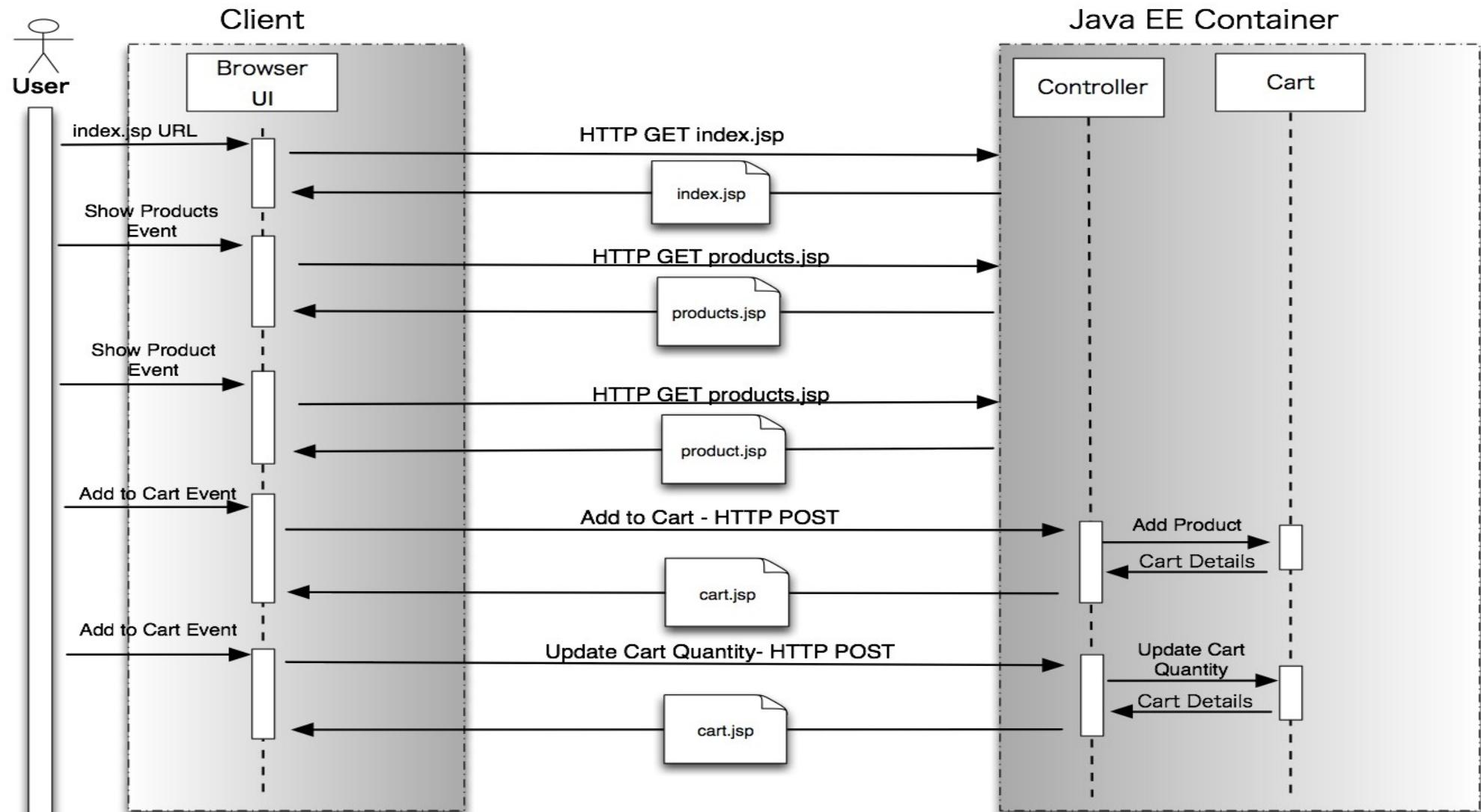
REST

- Web 2.0 delivers Services over the Web
- Pretty much all seems to be over REST
 - > HTTP + typed XML content
 - > (or JSON)
- JAX-WS 2.0 provides some support
 - > More is needed
 - > WSDL is an imperfect match

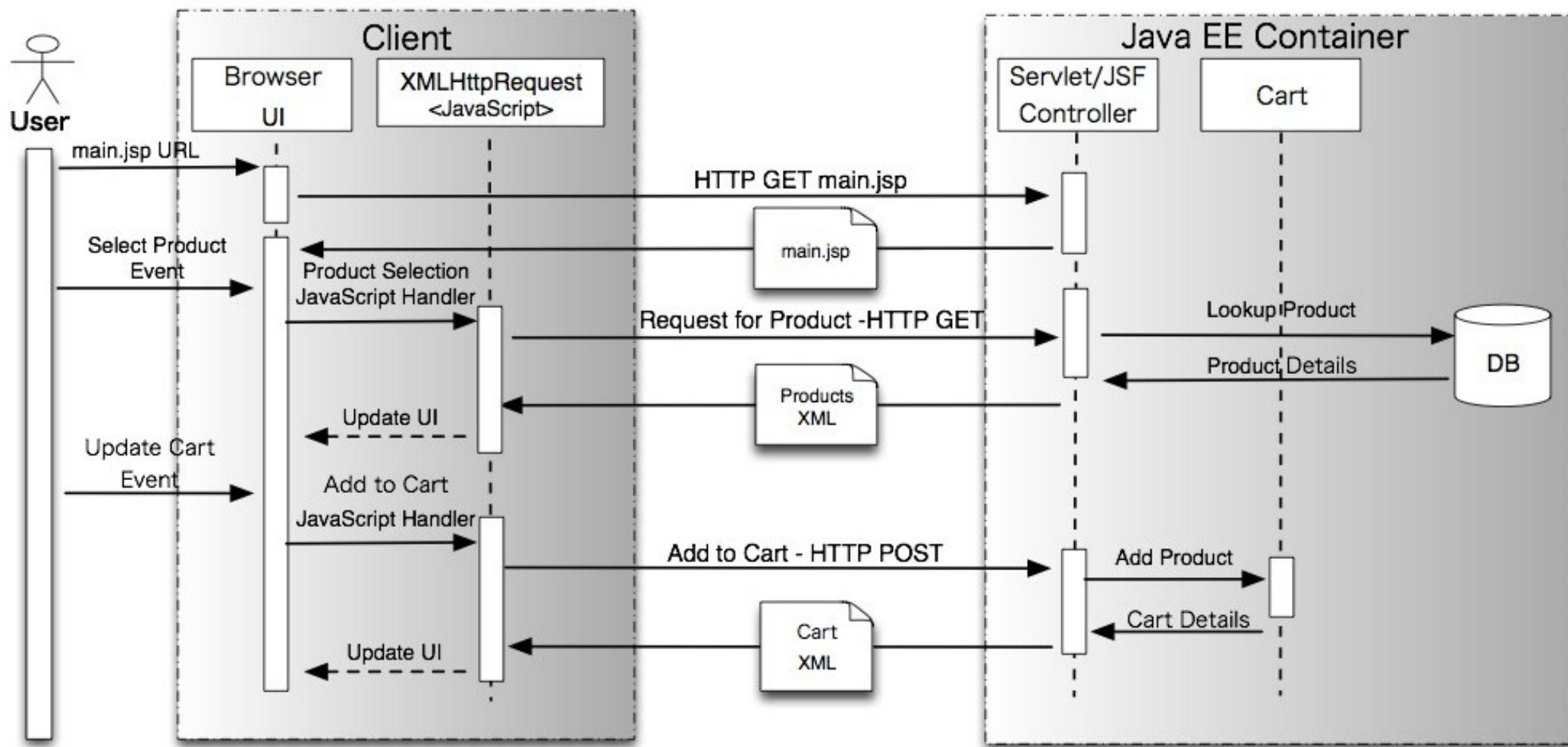
Conventional Rich Web Applications

- Plugins/Applets
- Frames/ iframes
- Dumb browser
- Server Centric
- Page to Page navigation based

Conventional Interaction Model



High Level AJAX Interaction Model

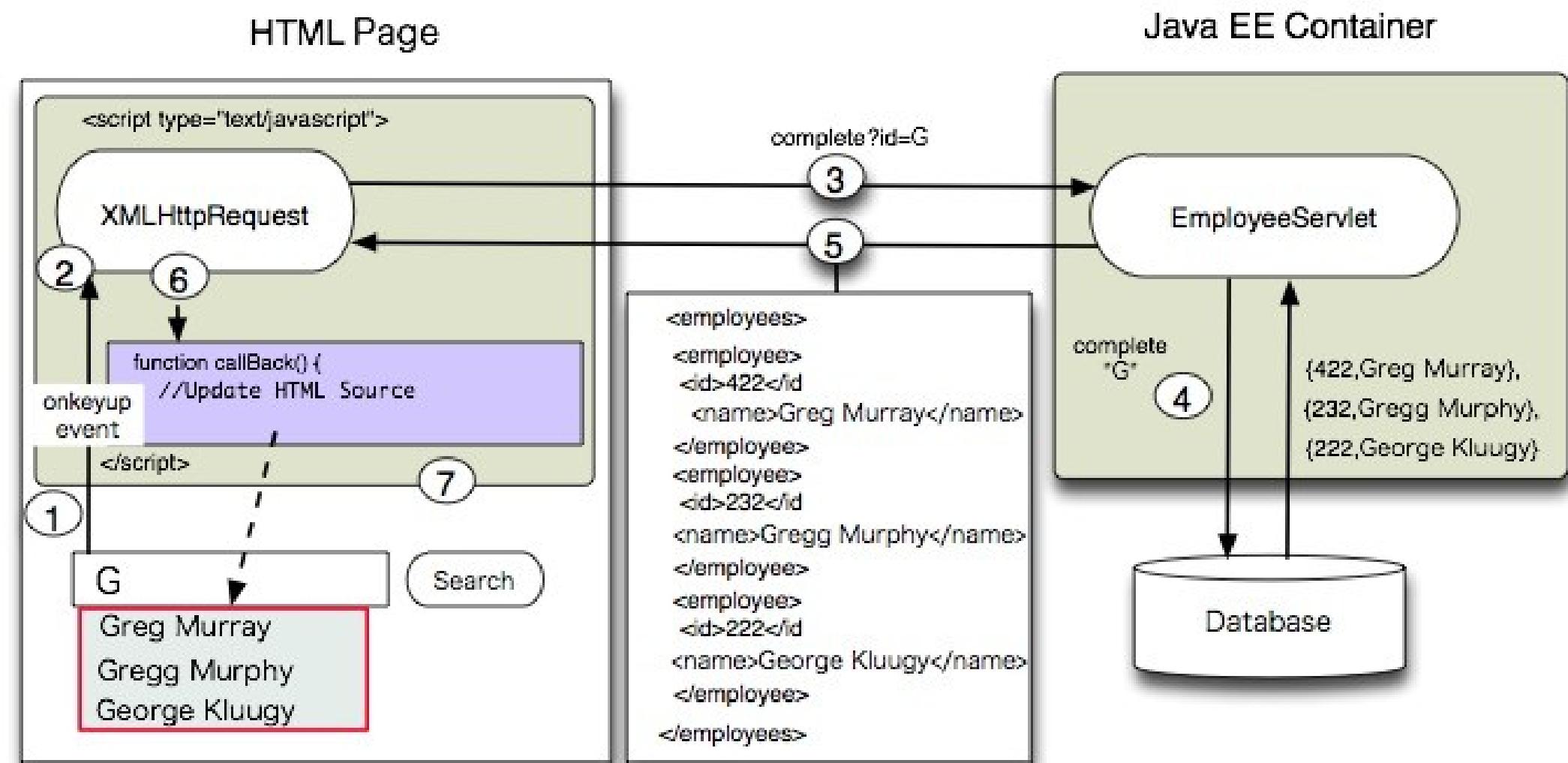


AJAX

Asynchronous JavaScript + XML

AJAX is using JavaScript, namely the XMLHttpRequest object, to communicate asynchronously with a server-side component and dynamically update the source of an HTML page based on the resulting XML/Text response.

Anatomy of an AJAX Interaction



HTML Page Event

```
<form name="autofillform" action="autocomplete" method="get">
<table border="0" cellpadding="5" cellspacing="0">
<tr><td><b>Employee Name:</b></td><td>
<input type="text" id="complete-field" size="20"
       autocomplete="off"
       onkeyup="doCompletion();">
</td><td align="left">
<input id="submit_btn" type="Submit" value="Lookup Employee">
</td></tr>
<tr><td id="auto-row" colspan="2">&ampnbsp</td></tr>
</table>
</form>
<div style="position: absolute; top:170px;left:140px" id="menu-popup">
<table id="completeTable" border="1" bordercolor="black" cellpadding="0"
cellspacing="0" />
</div>
```

JavaScript Event Handler

```
function getXHR() {  
    if (window.XMLHttpRequest) {  
        return new XMLHttpRequest();  
    } else if (window.ActiveXObject) {  
        return new ActiveXObject("Microsoft.XMLHTTP");  
    }  
}  
  
function doCompletion() {  
    var url = "autocomplete?action=complete&id=" +  
encodeURI(target.value);  
    var req = getXHR();  
    req.onreadystatechange = processRequest;  
    req.open("GET", url, true);  
    req.send(null);  
}
```

Servlet

```
public void doGet(HttpServletRequest request, HttpServletResponse response)
    throws IOException, ServletException {
    String targetId = request.getParameter("id");
    Iterator it = employees.keySet().iterator();
    while (it.hasNext()) {
        EmployeeBean e = (EmployeeBean)employees.get((String)it.next());
        if ((e.getFirstName().toLowerCase().startsWith(targetId) ||
            e.getLastName().toLowerCase().startsWith(targetId)) && !targetId.equals(""))
        {
            sb.append("<employee>");
            sb.append("<id>" + e.getId() + "</id>");
            sb.append("<firstName>" + e.getFirstName() + "</firstName>");
            sb.append("<lastName>" + e.getLastName() + "</lastName>");
            sb.append("</employee>");
            namesAdded = true;
        }
    }
    if (namesAdded) {
        response.setContentType("text/xml");
        response.setHeader("Cache-Control", "no-cache");
        response.getWriter().write("<employees>" + sb.toString() + "</employees>");
    } else response.setStatus(HttpServletResponse.SC_NO_CONTENT);
}
```

JavaScript Client Callback

```
function postProcess(responseXML) {  
    clearTable();  
    var employees = responseXML.getElementsByTagName("employees")[0];  
    if (employees.childNodes.length > 0) {  
        completeTable.setAttribute("bordercolor", "black");  
        completeTable.setAttribute("border", "1");  
    } else {  
        clearTable();  
    }  
    for (loop = 0; loop < employees.childNodes.length; loop++) {  
        var employee = employees.childNodes[loop];  
        var firstName = employee.getElementsByTagName("firstName")[0];  
        var lastName = employee.getElementsByTagName("lastName")[0];  
        var employeeId = employee.getElementsByTagName("id")[0];  
        appendEmployee(firstName.childNodes[0].nodeValue,  
lastName.childNodes[0].nodeValue, employeeId.childNodes[0].nodeValue);  
    }  
}
```

Agenda

- Definitions: Web 2.0, Rich Web Applications, AJAX
- Guidelines
- JSF Approach
- AJAX BluePrints

AJAX Guidelines

- JavaScript Libraries
- Usability Issues
- AJAX Design
- HTTP methods
- Security
- Desktop and AJAX

JavaScript Libraries

- Prototype
- RICO
- Script.aculo.us
- Dojo
- Zimbra

Recommendation: Adopt a library and don't try to re-invent the wheel.

Usability

- Back/Forward/Refresh Buttons
- Bookmarking
- URL Sharing
- Printing
- 508 Compliance
- Fallback Strategies
- Remember <blink>?

Recommendation: Consider the meaning of each and weigh the benefits when designing your application.

XMLHttpRequest (XHR)

- HTTP Method
 - > GET - When the result of $N > 0$ requests is the same.
 - > POST - When operation has “side-effects” and changes the state on the server.
- Concurrent Requests
 - > Max is 2 (IE) Consider - Pooling
 - > JavaScript Closures – Inline functions

Recommendation: Take care using the XHR. Use Closures to track the requests/callbacks. Consider using a library.

AJAX Design

JGD

- Add Around the Edges
 - > Small components (autocomplete, tree, partial submit)
- Page is the Application
 - > Client and Server split MVC responsibilities

Recommendation: Consider designing initial AJAX applications around the edges as you gain experience. Don't go overboard.

Response Content Type

- XML
- HTML
- Text
 - > Post processing on client
 - > Inject directly into the page
- JavaScript
 - > Evaluated in JavaScript using eval()
 - > JavaScript object representations of data(JSON)

Recommendation: Use XML for structured portable data. Use plain text for when injecting content into the HTML. Use JavaScript to return object representations data.

Security

- Sandboxed
 - > Cross Domain XMLHttpRequest restricted
 - > Access to file system restricted
- HTTPS – Requires a page refresh
- JavaScript Libraries for Encryption Exist
- JavaScript code visible to the world

Recommendation: Use HTTPS when you want to secure AJAX communication. Don't put compromising code in your JavaScript

Desktop and AJAX

- **AJAX is best for**
 - > Internet deployments (no separate runtime)
 - > Frequently updated, data intensive service apps
 - > Certain kinds of UI methods (HTML & CSS centric)
- **Swing (with or without Java Webstart) is best for**
 - > Disconnected / High Latency use
 - > Computationally intensive apps
 - > Very rich GUI needs
 - > When you need a mature environment NOW

Recommendation: Use the right tool for the job

Agenda

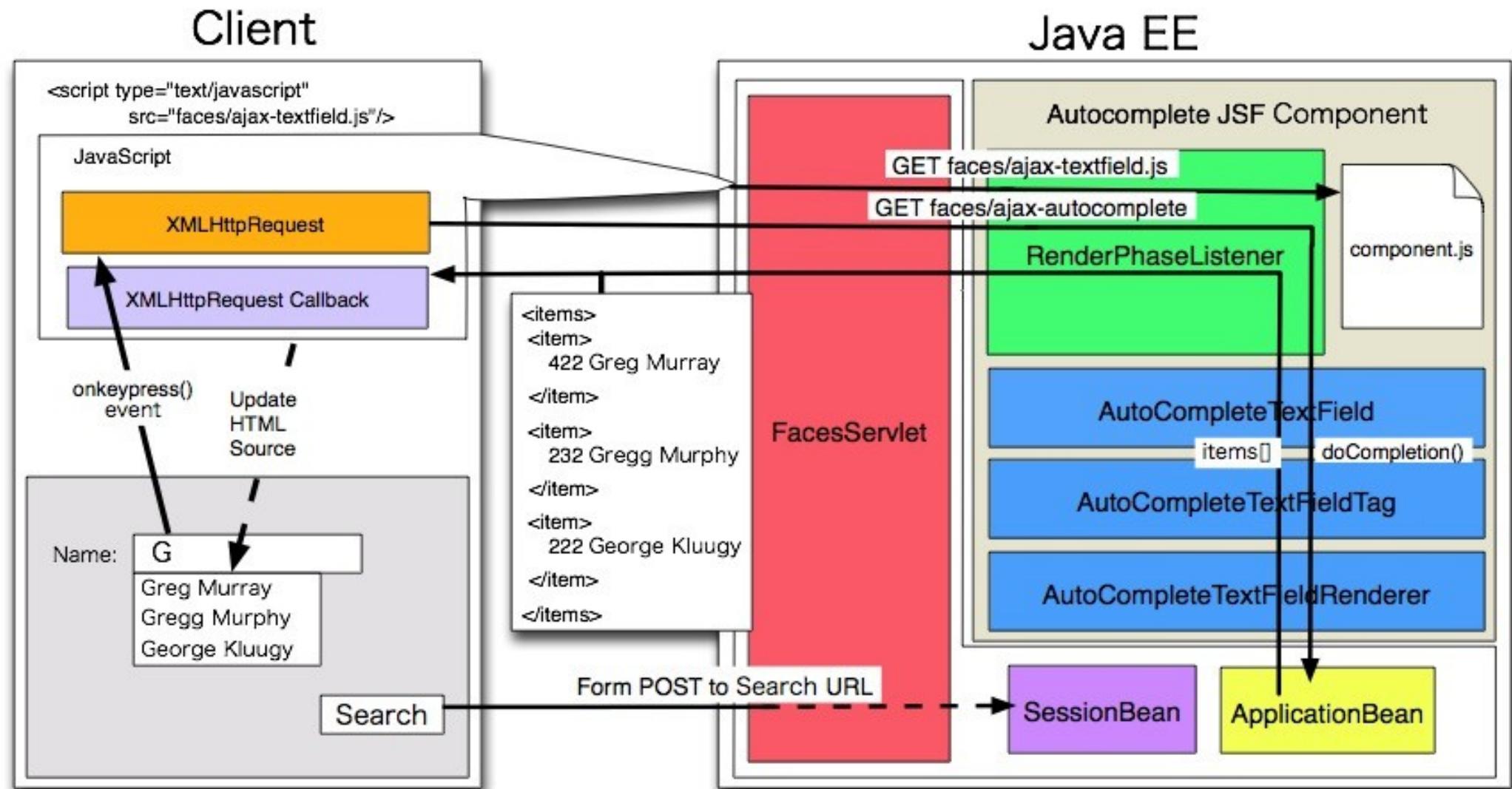
- Definitions: Web 2.0, Rich Web Applications, AJAX
- Guidelines
- JSF Approach
- AJAX BluePrints

JSF Component Approach

Benefits Include:

- Control Content Rendering
- Control of Server Side Logic
- All in one component
- Reusable
- Usable in a tool
- Hide AJAX complexity from page developers

Anatomy of an AJAX enabled JSF Component



Page Developer's View of JSF Component

JGD

```
<ajaxTags:completionField  
    size="40"  id="cityField"  
    completionMethod="  
        #{ApplicationBean.completeName} "  
    />
```

Server Side Logic for JSF Component

```
public String[] completeName() {  
    ArrayList results = new ArrayList();  
    Iterator it = employees.keySet().iterator();  
    while (it.hasNext()) {  
        EmployeeBean e = (EmployeeBean)employees.get((String)it.next());  
        if ((e.getFirstName().toLowerCase().startsWith(targetId) ||  
            e.getLastName().toLowerCase().startsWith(targetId)) &&  
            !targetId.equals("")) {  
  
            results.add(e.getId() + " " +  
                        e.getFirstName() +  
                        e.getLastName());  
        }  
    }  
    return (String[])results.toArray();  
}
```

Agenda

- Definitions: Web 2.0, Rich Web Applications, AJAX
- Guidelines
- JSF Approach
- **AJAX BluePrints**

AJAX BluePrints

JGD

- BluePrints Solutions Catalog Entries on AJAX
 - > For both NetBeans and command line
 - > Written for GlassFish (<http://glassfish.dev.java.net>)
 - > More entries posted all the time
- Java Petstore Demo
 - > Will be released at JavaOne
- Blueprints AJAX components
 - > Google Maps, Paypal
 - > Autocomplete, Ratings, File Upload
 - > Usable in Java Studio Creator, included in updates

Resources

- BluePrints AJAX Page:
<http://java.sun.com/blueprints/ajax.html>
- AJAX FAQ for the Java Developer
<http://blueprints.dev.java.net/ajax-faq.html>



jim.driscoll@sun.com