



JavaOneSM
Sun's 2000 Worldwide Java Developer Conference*



JavaOneSM
Sun's 2000 Worldwide Java Developer Conference

JavaServer PagesTM Technology and JavaTM Servlet Components Overview

Danny Coward
Staff Engineer

Eduardo Pelegri-Llopart
Senior Staff Engineer
Sun Microsystems, Inc.

Agenda

- Java™ Servlet API
- JavaServer Pages™ technology
- Web applications
- The world of web-based Java technology
- More information
- Q&A

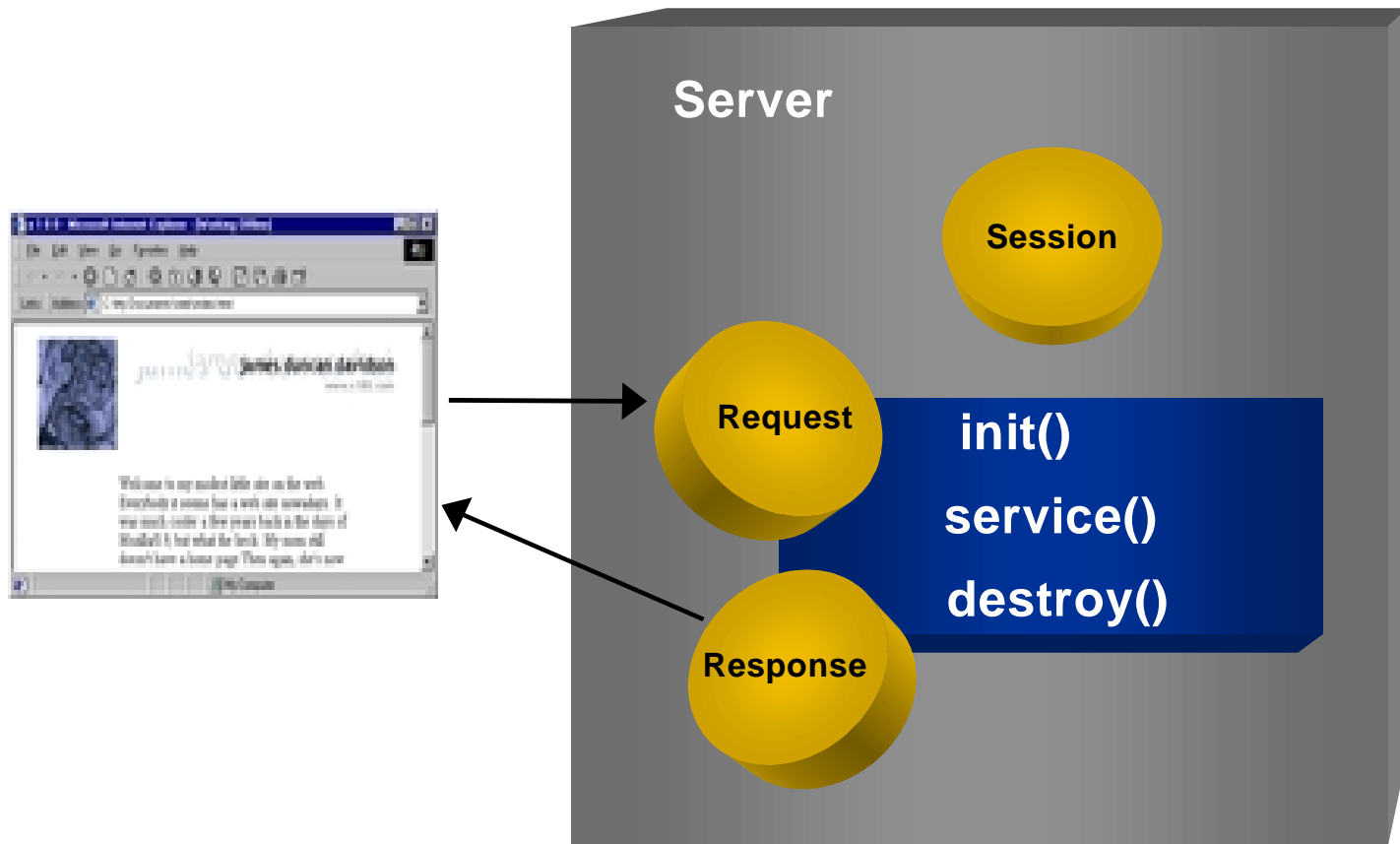


What Is a “Java Servlet”?

- Extension mechanism for dynamic content generation over HTTP, based on the Java[™] Servlet API
- Mapped to URLs
- Simple architecture
- Managed by container
- Very rich platform facilities available



Request and Response



Simple HTML Example

```
public class HelloServlet extends HttpServlet {  
    public void doGet(HttpServletRequest request,  
                      HttpServletResponse response)  
    {  
        response.setContentType("text/plain");  
        PrintWriter out = response.getWriter();  
        out.println("Hello World!");  
    }  
}
```



Body of WAP Servlet Service

```
response.setContentType("text/vnd.wap.wml");
PrintWriter out = response.getWriter();
out.println("<?xml version=\"1.0\"?>");
out.println("<wml>");
out.println("<card id=\"card1\" title=\"Flowers\">");
out.println("<p>\nTypes Available today:<br/>");
for (int i = 0; i < types.length; i++) {
    out.println(types[i]+"<br/>");
}
out.println("<br/> </p>");
out.println("</card>");
out.println("</wml>");
```



Request and Response

- Request
 - HTTP request headers
 - InputStream or reader
 - Form data, cgi data
- Response
 - HTTP response headers
 - OutputStream or writer
 - Setting cookies, redirects, error pages

HttpSession and ServletContext

- **HTTPSession**
 - Encapsulate series of request from same client
 - Hides various mechanisms: SSL, URL rewriting, cookies
- **Servlet context**
 - Sharing application data
 - Dispatching requests to other servlets
 - Convenience methods (e.g., writing log files)



Simple HTML Example

```
public class HelloServlet extends HttpServlet {  
    public void doGet(HttpServletRequest request,  
                      HttpServletResponse response)  
    {  
        response.setContentType("text/plain");  
        PrintWriter out = response.getWriter();  
        out.println("Hello World!");  
    }  
}
```



Longer HTML Example

```
public class HelloServlet extends HttpServlet {
    public void doGet(HttpServletRequest request,
                      HttpServletResponse response)
    {
        response.setContentType("text/plain");
        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println("Hello World!");
        out.println("<br>");
        JspCalendar clock = new JspCalendar();
        out.println("Today is");
        out.println("<ul>");
        out.println("<li>Day of month: ");
        out.println("clock.getDayOfMonth());
        out.println("</ul>");
        out.println("</html>");
    }
}
```



Simple JSP Component

```
<html>
  Hello World!
  <br>
  <jsp:useBean id="clock"
               class="calendar.JspCalendar" />
  Today is
  <ul>
    <li>Day of month: <%= clock.getDayOfMonth() %>
    <li>Year: <%= clock.getYear() %>
  </ul>
</html>
```



JSP Page

- A textual document describing how to create a *response* from a *request*
 - Template data
 - Dynamic content



JavaServer Pages™ (JSP) Components

- JSP Builds on Servlet Semantics
 - “Inside-Out” Servlets
- Benefits:
 - Easier to Author
 - Easier for Tools
 - Designer-serviceable part
- Exploits:
 - Server-Side actions and scripting
 - Templates
 - Encapsulation of Functionality



JSP Component, Elements

- Directives

```
<%@ page errorPage="error.jsp" %>
```

```
<%@ taglib uri="my.tld" prefix="x" %>
```

```
<%@ include uri="header.jsp" %>
```

- Scripting elements

```
<% myBean.setSomething("foo"); %>
```

```
<%= myBean.getResult(); %>
```

- Implicit objects
 - Servlet abstractions

- Actions



Standard Actions

- Available on all compliant containers
- Common actions:
 - `useBean`
 - `setProperty`
 - `getProperty`
 - `include`
 - `forward`



Provide Your Own Actions: Tag Libraries

- Encapsulate functionality
 - Reusable
 - Maintainable
- Easier to
 - Author manually
 - Manipulate by tool
- Provide
 - Portable semantics
 - Portable tool support



Examples of Custom Actions

- A conditional inclusion of dynamic content
`<x:includePersonalHeader />`
 - Action uses:
 - Request information
 - Customization data
- An XSLT transformation
`<x:applyXSL source="url" xsl="url" />`
 - Action uses:
 - Source data



Example of Custom Actions, JSP for WML

```
<?xml version="1.0">
<wml>
  <card id="card1" title="Flowers">
    <p>
      Types available today:
    <br/>
      <x:enumerate inventory="flowers"
                    id="aFlower">
        $aFlower.type
      <br/>
    </x:iterate>
    <br/>
  </p>
</card>
</wml>
```



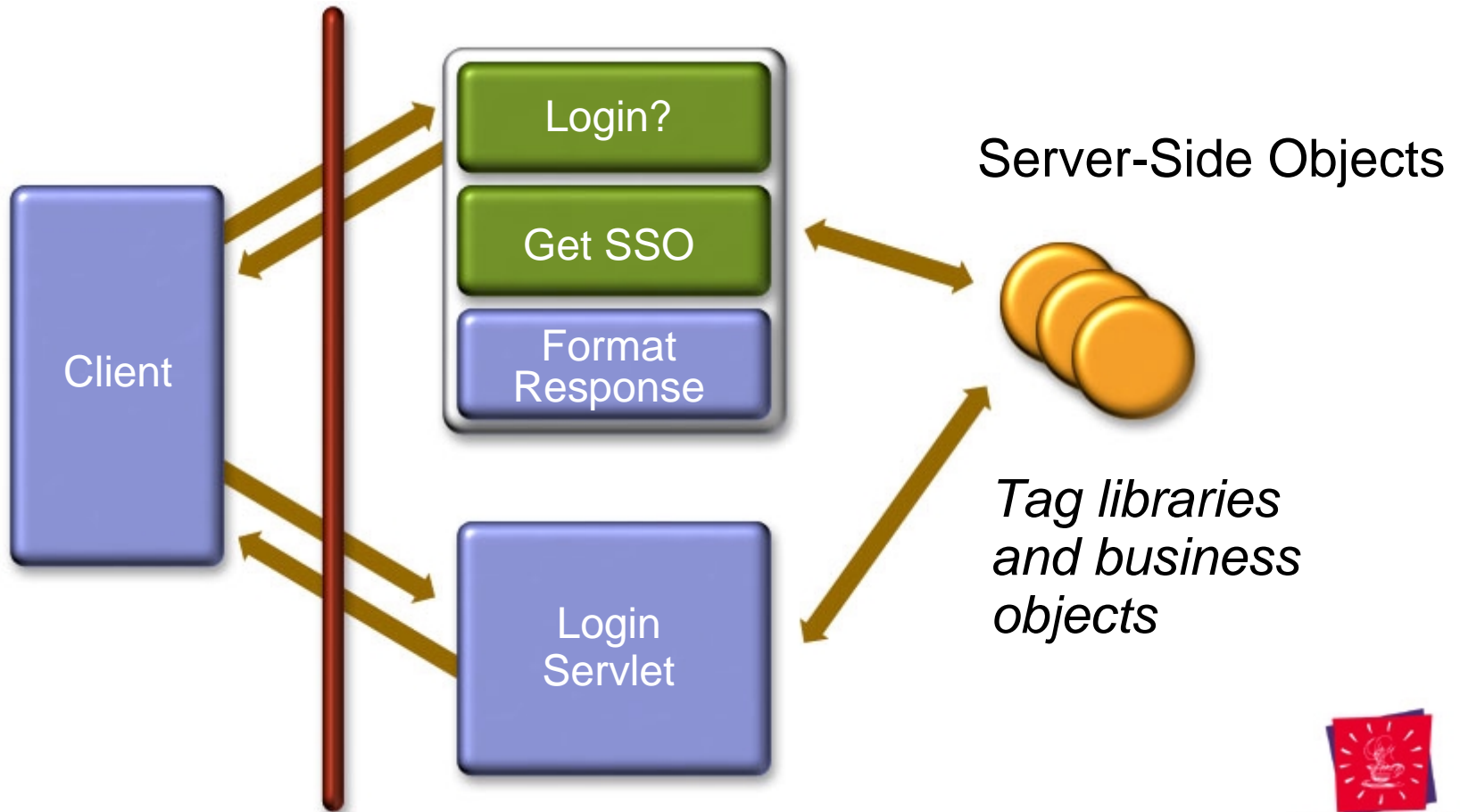
Some Architectures

- Different ways to organize
 - Source
 - Servlets
 - JSP components
- More details in tomorrow's session



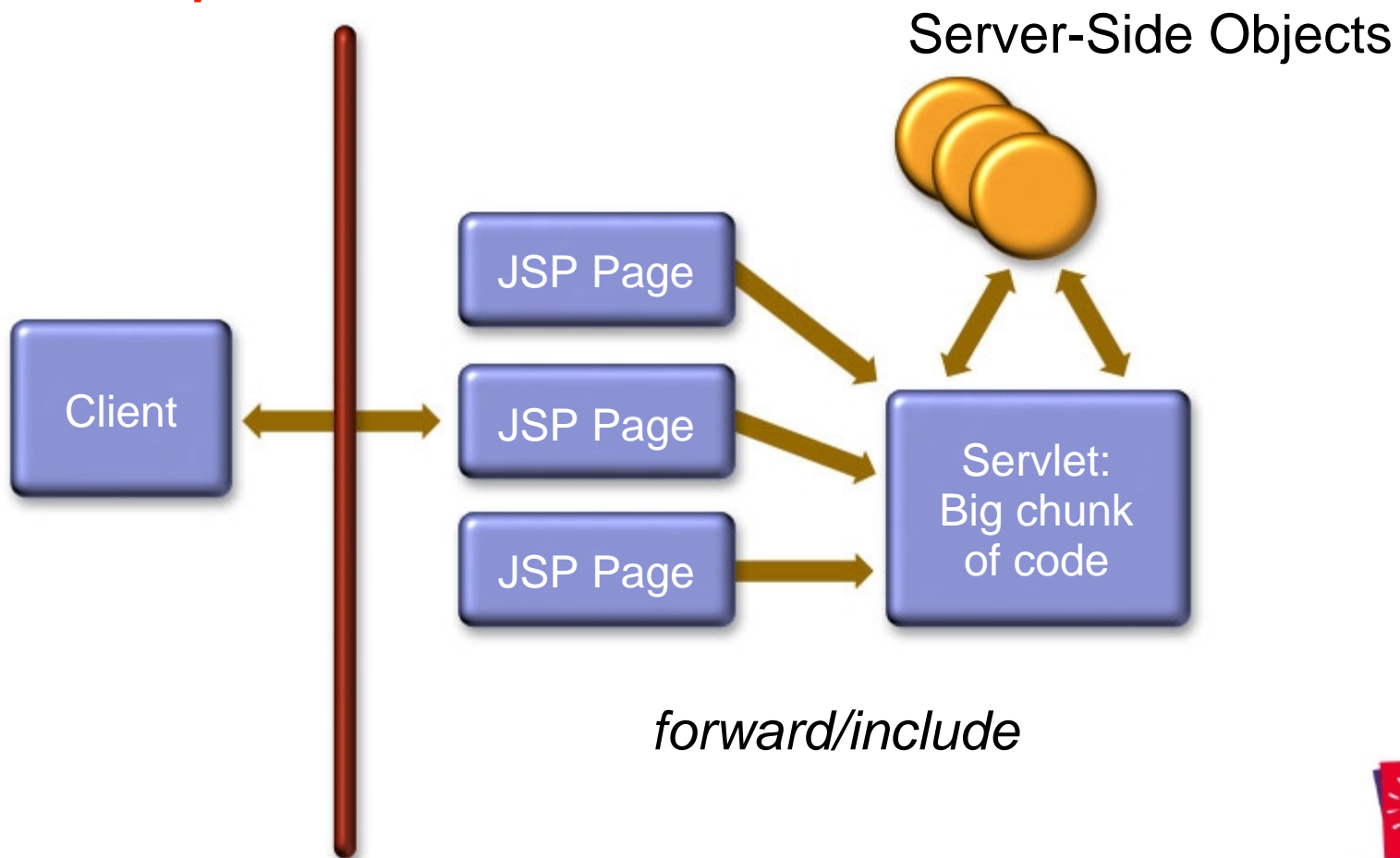
Simple Login + Pages

J1 @ j.s.c site



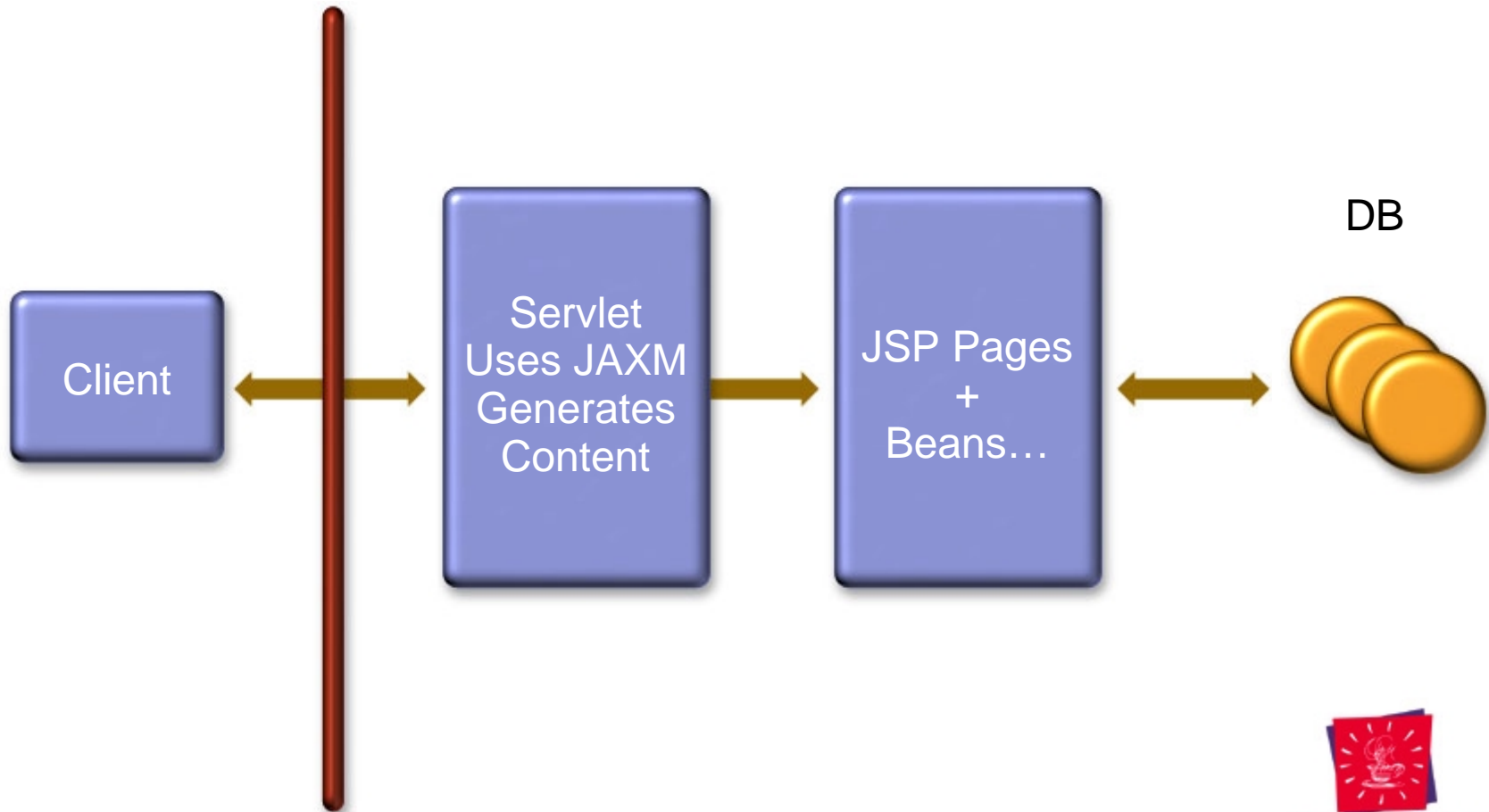
JSP Pages as Front to Servlets

Perceptual robotics



Servlets Processing XML

J1 @ j.s.c



Web Applications

- Grouping of web components with common purpose
- Standard file format (Web ARchive)
- Standardized deployment configuration



Web Applications— Deployment Descriptor

- XML document within WAR
- Structural description
- Mappings of web components to URLs
- Initialization and configuration parameters
- Dependencies on other entities
 - Database connection, EJB[™] component...



Web Application Security

- Authentication
 - HTTP basic
 - Client certificate
 - Customized login
- Authorization
 - User access to portions of web application
 - Transport type

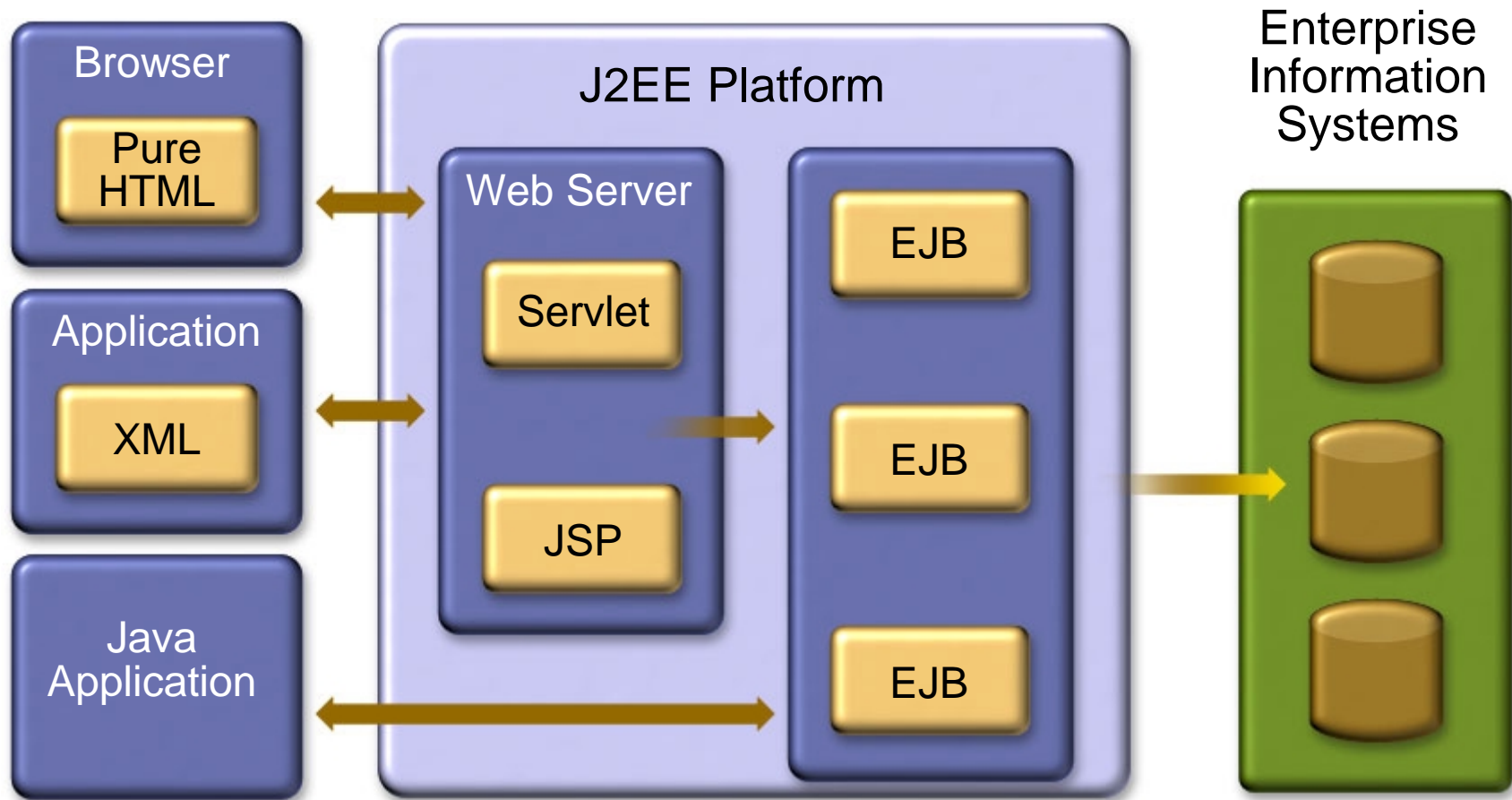


Web Applications—Settings

- Embedded
- Web platforms
 - *Internet editions*
- J2EE™ platform-enabled application servers
 - *Enterprise editions*



Web Components on the J2EE™ Platform



Web-Based Java Technology—History

- Servlets
 - Servlet 2.1 in Jan '99
 - Servlet 2.2 in Dec '99
- JSP API
 - JSP 1.0 in June '99
 - JSP 1.1 in Dec '99
- Tomcat @ Apache
 - June '99
- J2EE 1.2 platform
 - Dec '99 with Servlet 2.2 and JSP 1.1



The JSP Component and Servlet Industry

- Developer tools
 - Servlet development
 - Tag Library development
- Authoring tools
 - Page creation
 - Site creation and site management
- Containers
 - JSP component and Servlet containers
 - J2EE™ platform containers



Industry Support

Containers

- Apache
- Iplanet iWS, iAS
- IBM's WebSphere
- BEA Weblogic
- Inprise AppServer
- ATG Dynamo
- Bluestone

Containers, 2

- Gemstone
- Oracle
- Unify
- Allaire
- Persistence
- Caucho
- ...

Tools

- MacroMedia UltraDev
- Adobe GoLive
- Unify eWeb Studio
- WebGain's Studio
- Forte for Java
- Oracle Jdeveloper
- Inprise Jbuilder
- ATG Dynamo
- ...



Some Web Site Highlights

- JavaOneSM @ java.sun.com/javaone
- JavaOne portal
- www.delta-air.com
- www.food.com
- www.perceptualrobotics.com
- www.carorder.com
- ...



Next Versions...

- Follow the Java Community ProcessSM initiative
 - JSR-053
 - JSR-052
- Next versions are under way
 - Participant draft stage
- Public draft later this year



Next Version Areas of Work

- Plugging content filters
 - XML (DOM, SAX), XSL/T, others...
- Portable support for application lifecycle
- Improved support for page authoring
- Improved XML support



More Information

- Web
 - java.sun.com/products/jsp
 - java.sun.com/products/servlet
- Mail
 - pelegri@eng.sun.com, danny.coward@sun.com
 - servlet-interest@java.sun.com
 - jsp-interest@java.sun.com
- JavaOneSM conference
 - Sessions: Architecture Wed, Tomcat on Thur
 - BOFs: 3 this evening, 1 Wed, 1 Thur
 - Many other sessions
 - Drop by our booth (#119)





JavaOneSM
Sun's 2000 Worldwide Java Developer Conference™

Q&A

Ask Us !



JavaOneSM
Sun's 2000 Worldwide Java Developer Conference*