JSP Basics 1

Sang Shin
JPassion.com
"Code with JPassion!"

Agenda

- What is JSP?
- JSP vs Servlet
- JSP/Servlet in big picture of Java EE
- JSP architecture
- Life-cycle of a JSP page
- Steps for building JSP-based web application

What is JSP?

What is JSP page?

- A text-based document capable of returning dynamic content to a client browser
  - It looks and feels like a HTML page
- Contains both static and dynamic content
  - Static content: via HTML, XML
  - Dynamic content: via programming code (not recommended), and JavaBeans, custom tags

Why JSP Technology?

- Enables separation of business logic from presentation
  - Presentation is in the form of HTML or XML/XSLT
  - Business logic is implemented as Java Beans or custom tags
  - Better maintainability, reusability (than just using servlets only)
- Extensible via custom tags
- Builds on Servlet technology
  - A JSP gets compiled into Servlet before deployment

JSP Sample Code

```
<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>
```

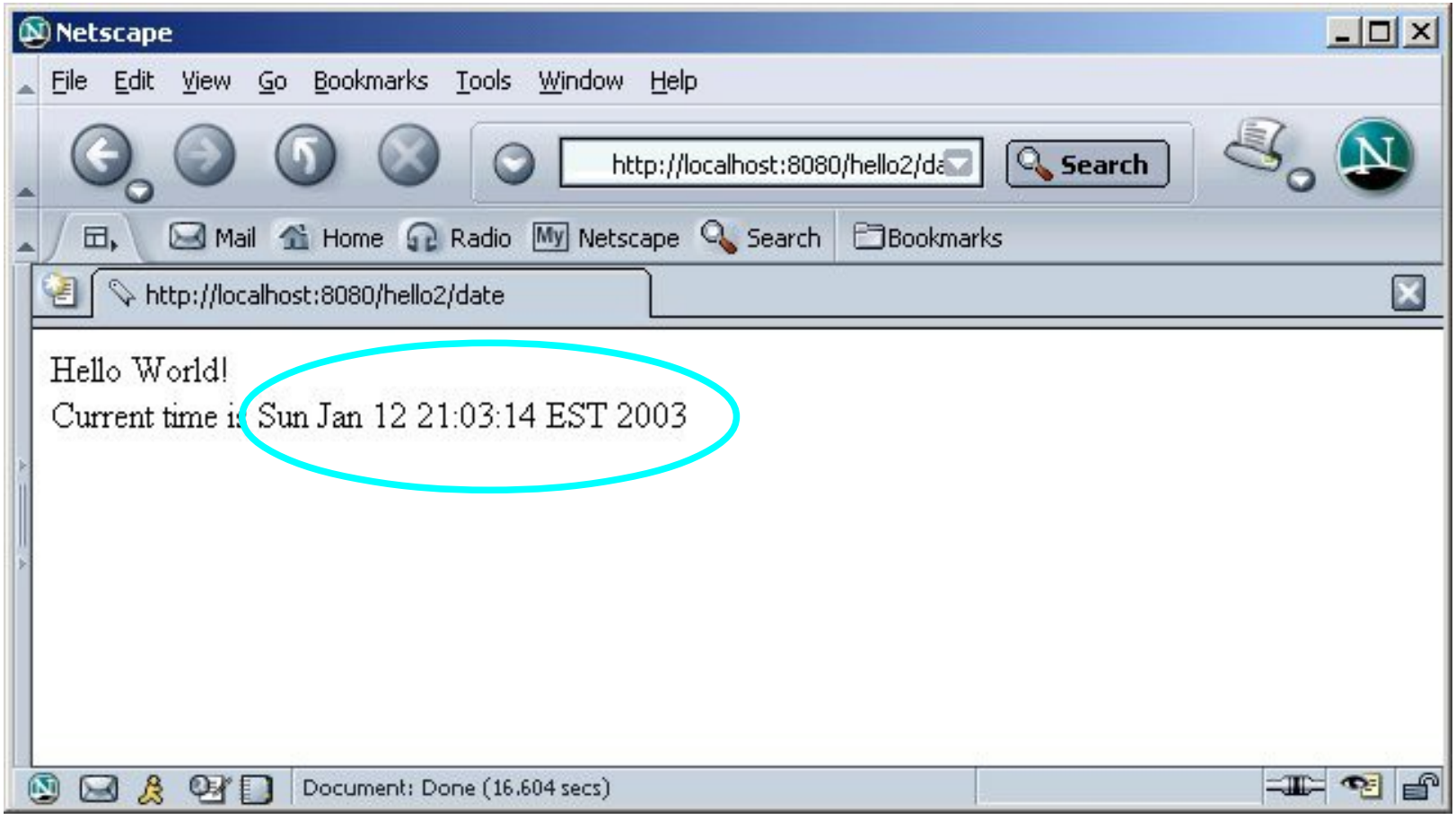
Static vs. Dynamic Contents

- Static contents
  - Typically static HTML page
  - Same display for everyone
- Dynamic contents
  - Contents is dynamically generated based on conditions
  - Conditions could be
    - User identity
    - Time of the day
    - User entered values through forms and selections
  - Examples
    - Etrade webpage customized just for you, my Yahoo

A Simple JSP Page (Blue: static, Red: Dynamic contents)

```
<html>
<body>
  Hello World!
 <br>
 Current time is <%= new java.util.Date() %>
</body>
</html>
```

Output


Netscape
File Edit View Go Bookmarks Tools Window Help
http://localhost:8080/hello2/da
Search
Mail Home Radio My Netscape Search Bookmarks
http://localhost:8080/hello2/date
Hello World!
Current time is: Sun Jan 12 21:03:14 EST 2003
Document: Done (16.604 secs)

JSP vs. Servlet

JSP Benefits over Servlet (for View generation)

- Easier to author web pages (compared to Servlets)
- Content and display logic are separated
- Web application development is simplified with JSP, JavaBeans and tags (JSTL and custom tags)
- Software reuse is possible through the use of components (JavaBeans and tags)

JSP is “Servlet”

- JSP pages get translated into servlets
  - Tomcat translates greeting.jsp into greeting$jsp.java
- Scriptlet (Java code) within JSP page ends up being inserted into jspService() method of resulting servlet
- Implicit objects for servlet are also available to JSP page designers, JavaBeans developers, custom tag designers

Should I Use Servlet or JSP?

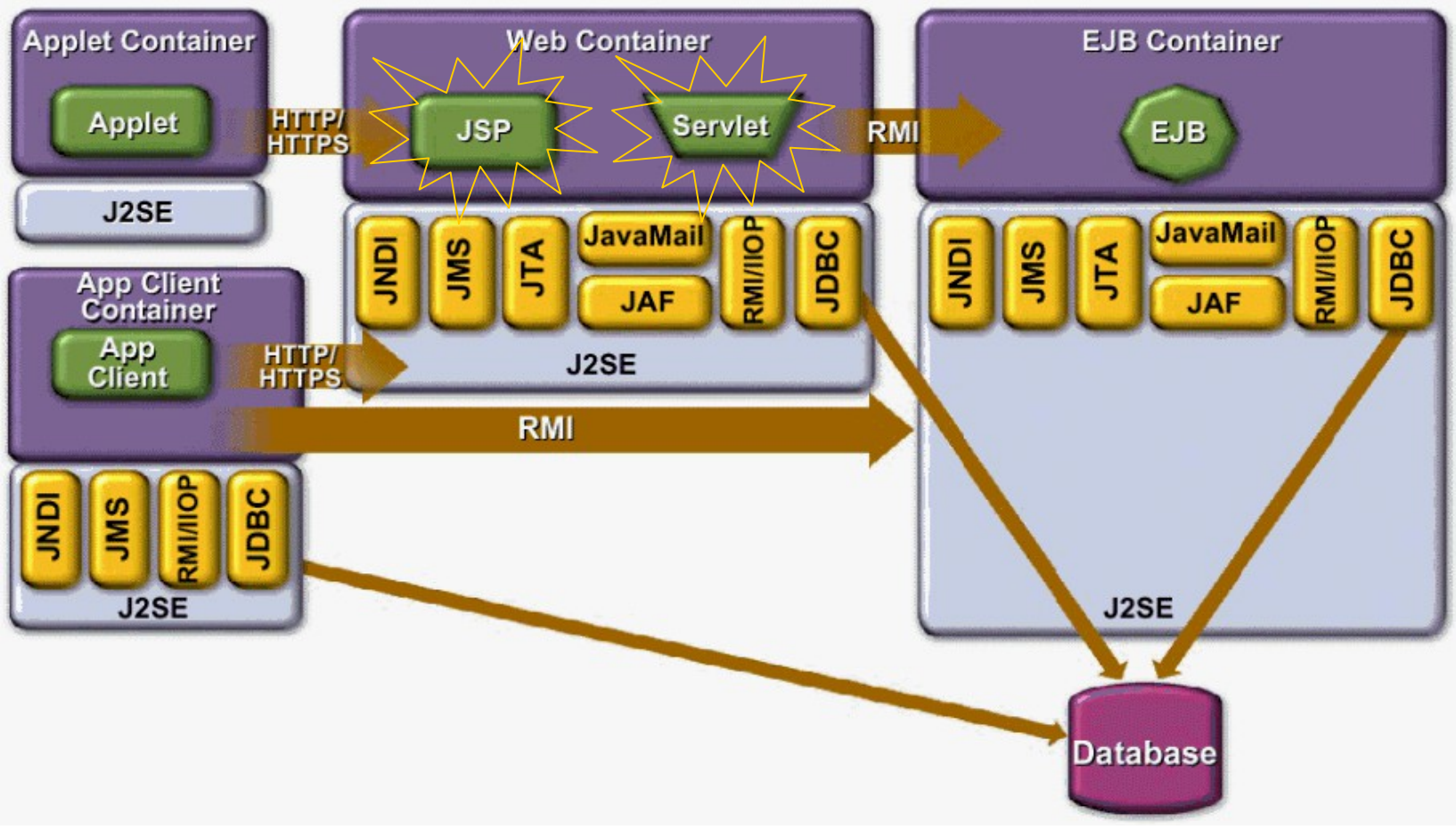
- In practice, servlet and JSP are used together
  - Via MVC (Model, View, Controller) architecture
  - Servlet plays the role of the Controller
  - JSP plays the role of the View

Servlet/JSP vs. Web Frameworks

- Limitations of vanilla Servlet/JSP
  - Vanilla Servlets/JSP are considered too low-level for building real-life production-quality Web applications
  - Vanilla Servlets/JSP do not provide common features needed for building web applications such as Dispatch framework, Data binding, validation, layout, internationalization, etc
- So it is highly likely you will use popular MVC-based Web application frameworks, which provide extra features over vanilla Servlet/JSP
  - SpringMVC, Wicket, Tapestry, Struts, etc

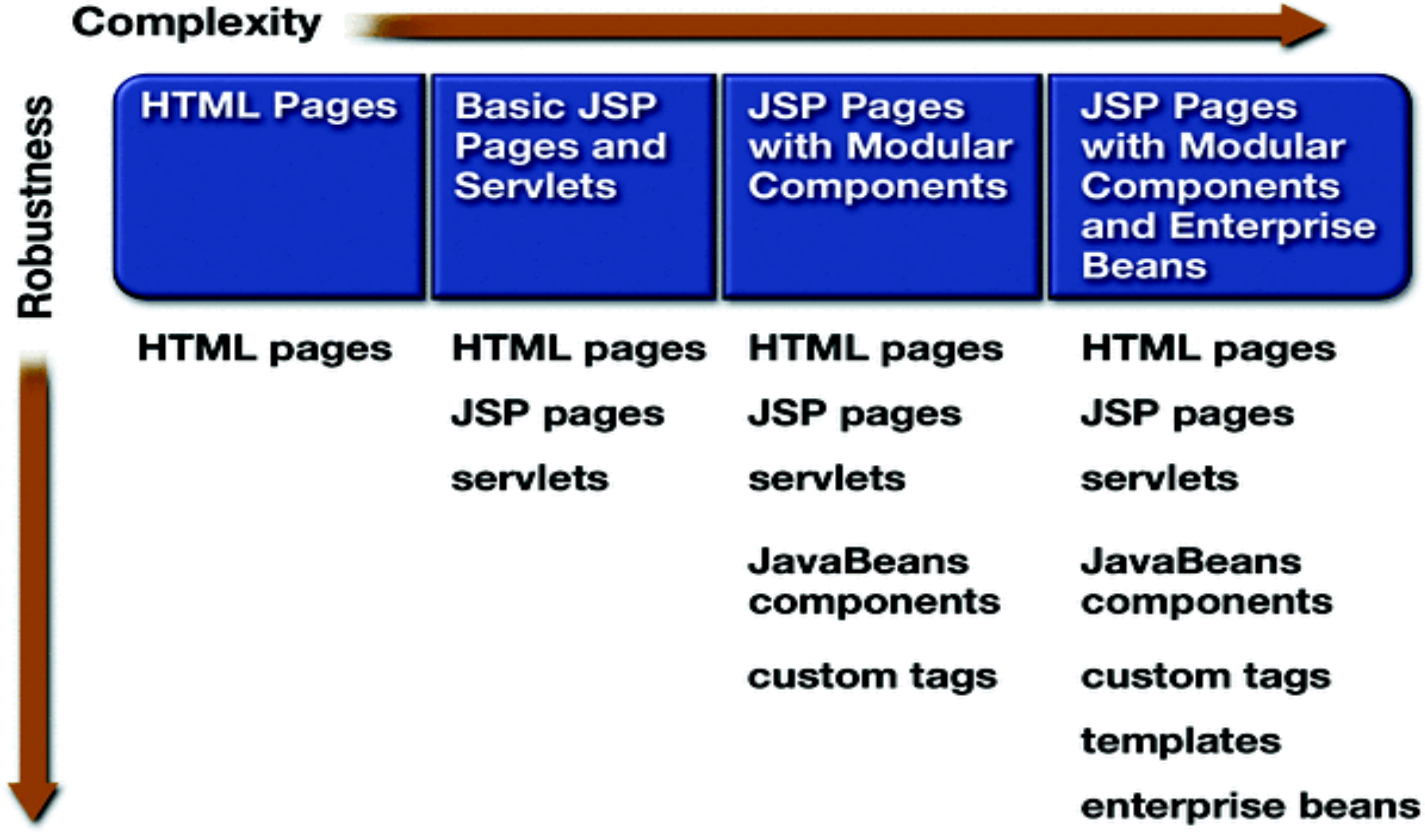
JSP in Big Picture of Java EE

JSP & Servlet as Web Components

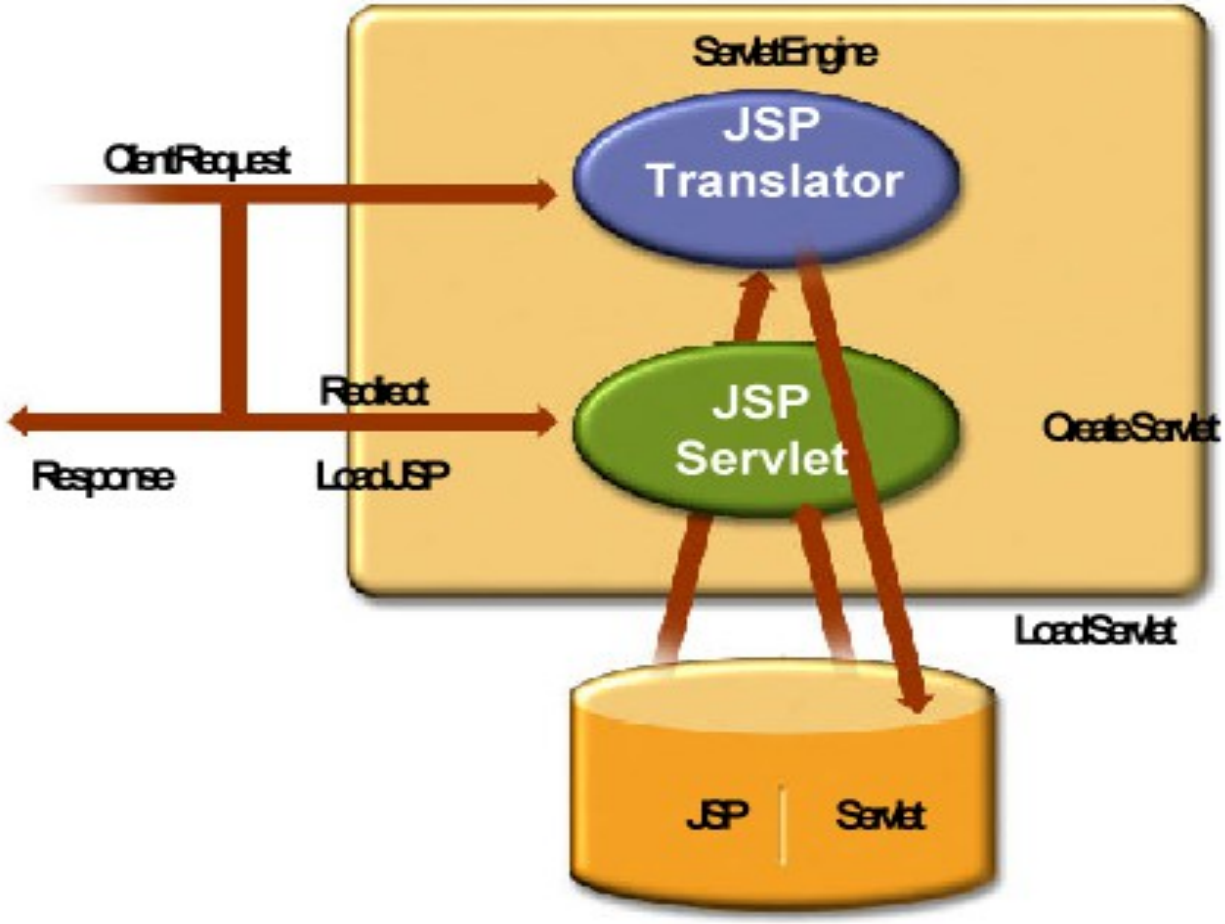


JSP Architecture

Web Application Designs

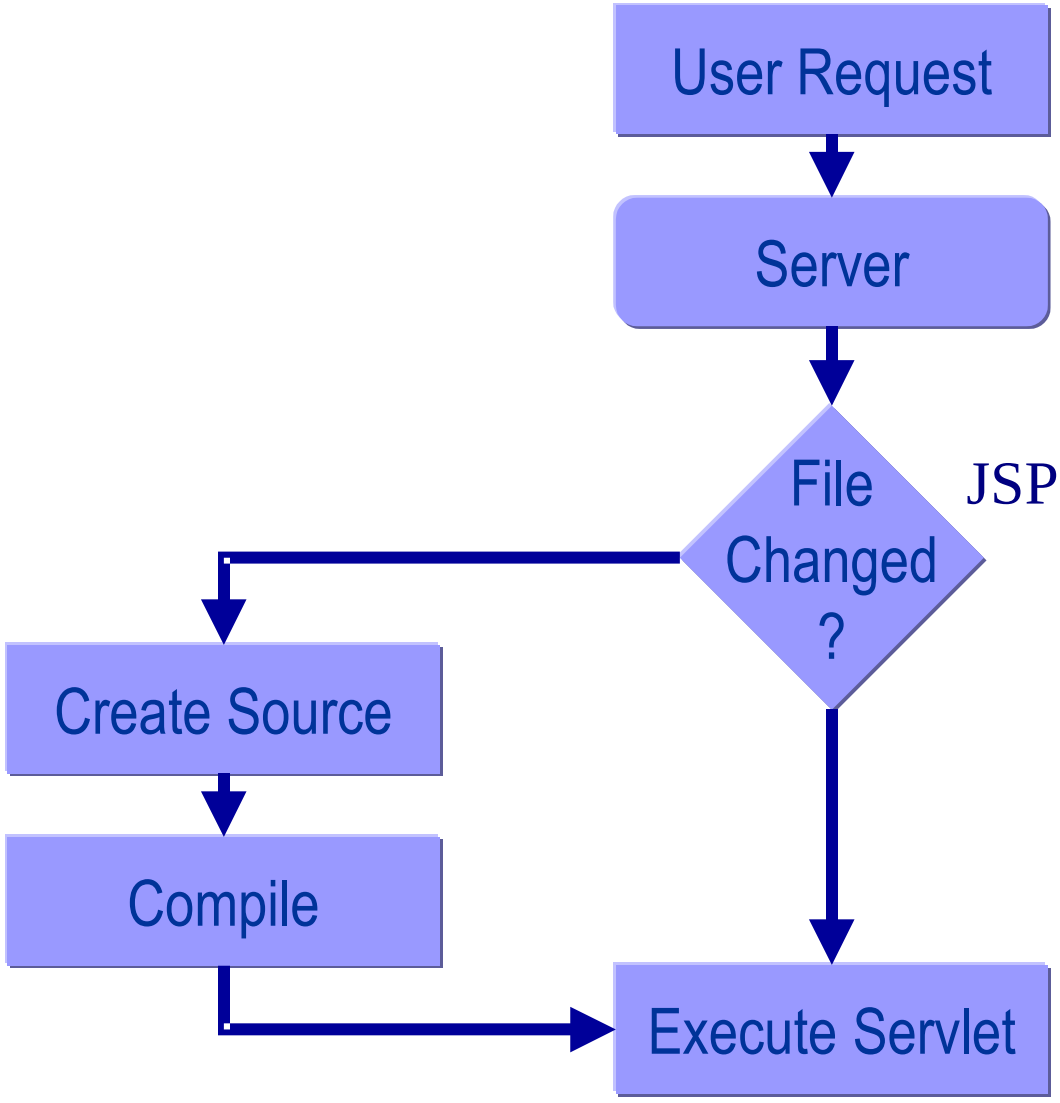


JSP Architecture

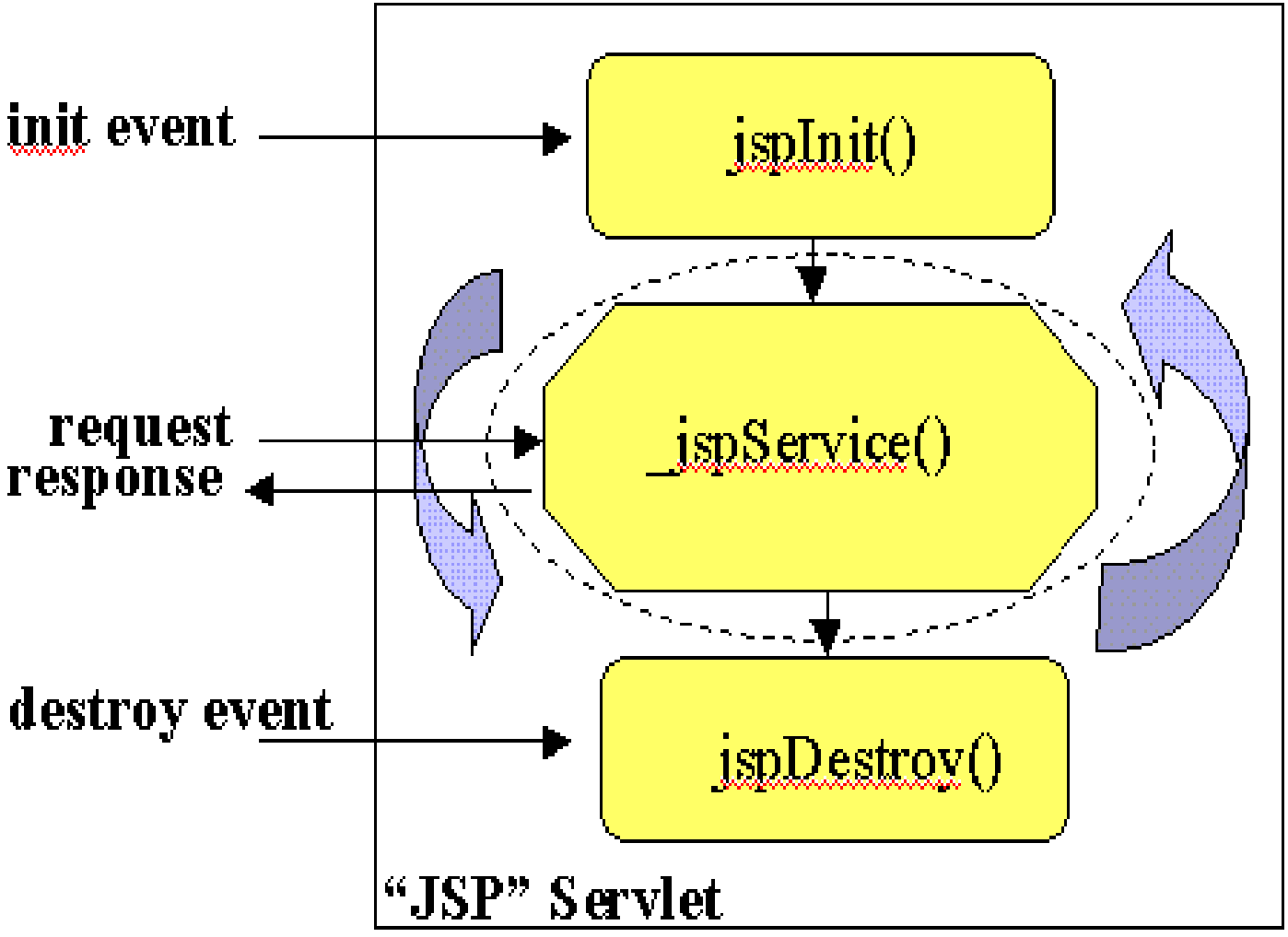


Life cycle of a JSP page

How Does JSP Work?



JSP Lifecycle Methods during Execution Phase



Steps for building JSP-based Web application

Web Application Development and Deployment Steps

1. Write the Web component code (Servlet and JSP)
2. Create any static resources (for example, images or CSS files)
3. Create deployment descriptor (web.xml)
4. Build the Web application (*.war file or deployment-ready directory)
5. Install or deploy the web application into a Web container
   - Clients (Browsers) are now ready to access them via URL

Code with Passion!
JPassion.com