Module 2
Java Web Applications
Web, Servlets, JSP, Web Containers

Web Pages (1)

- 1993
  - Originally web browser used to just download static files (HTML,
  - JPEG, TXT)
  - Web browser only responsible for displaying
  - Web server sends files over HTTP protocol
    - Web server used to be just a file repository
- 1996+
  - Web pages started to collect data from user
  - Form concept introduced:
    - GET to download a file from the server
    - **POST to send filled in form to the server**
    - PUT to upload a file to the server
    - DELETE removing file on the server

Web Pages (2)

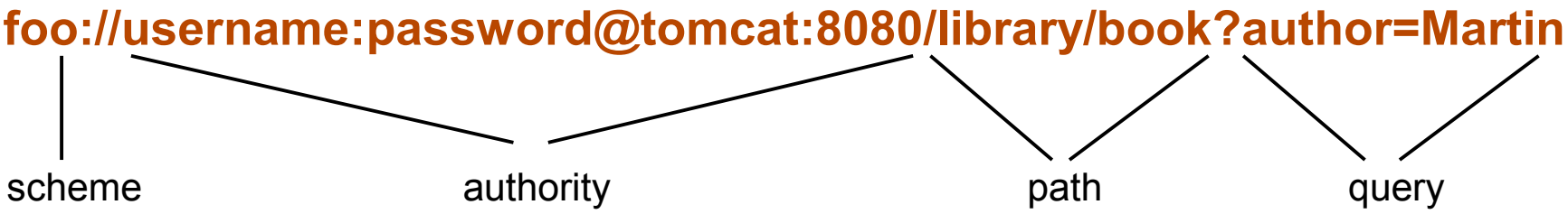
- Separation
  - layout (web page)
  - rendering (web browser) is perfect
- Current web servers send *"virtual"* files
  - Pages with data dynamically filled in from DB
- Java web server:
  - HTML files dynamically constructed using
    - JSP (similar to PHP or ASP.NET)
    - Java classes – Servlets
    - JSF (server side web components like table, tree, calendar ..., similar to ASP.NET Web Forms)

HTTP Protocol

- Request-Response Stateless Application protocol
- Foundation of data communication for the WWW
- HTTP Request methods
  - GET : Requests a representation of the specified resource
  - POST : Requests that the server accept the entity enclosed in the request as a new subordinate of the web resource identified by the URI
  - PUT : Requests that the enclosed entity be stored under the supplied URI
  - DELETE : Deletes the specified resource
  - HEAD, OPTIONS, TRACE, PATCH, CONNECT

URI, URN and URL

- Uniform Resource Identifier
- String of characters used to identify a name of a web resource
- Can be classified as
  - name (URN) : an item's identity (*urn:isbn:1234567890*)
  - locator (URL) : provides a method for finding identity (*http://server/library/book?isbn=1234567890*)



Servlets

- Java EE component
  - Java class (an object in runtime) and configuration piece
  - Assigned to a certain URL (suffix)
  - Has methods `doGet()`, `doPost()`, `doPut()`,`doDelete()`, which generate (usually) HTML
  - Output is written to Writer or OutputStream

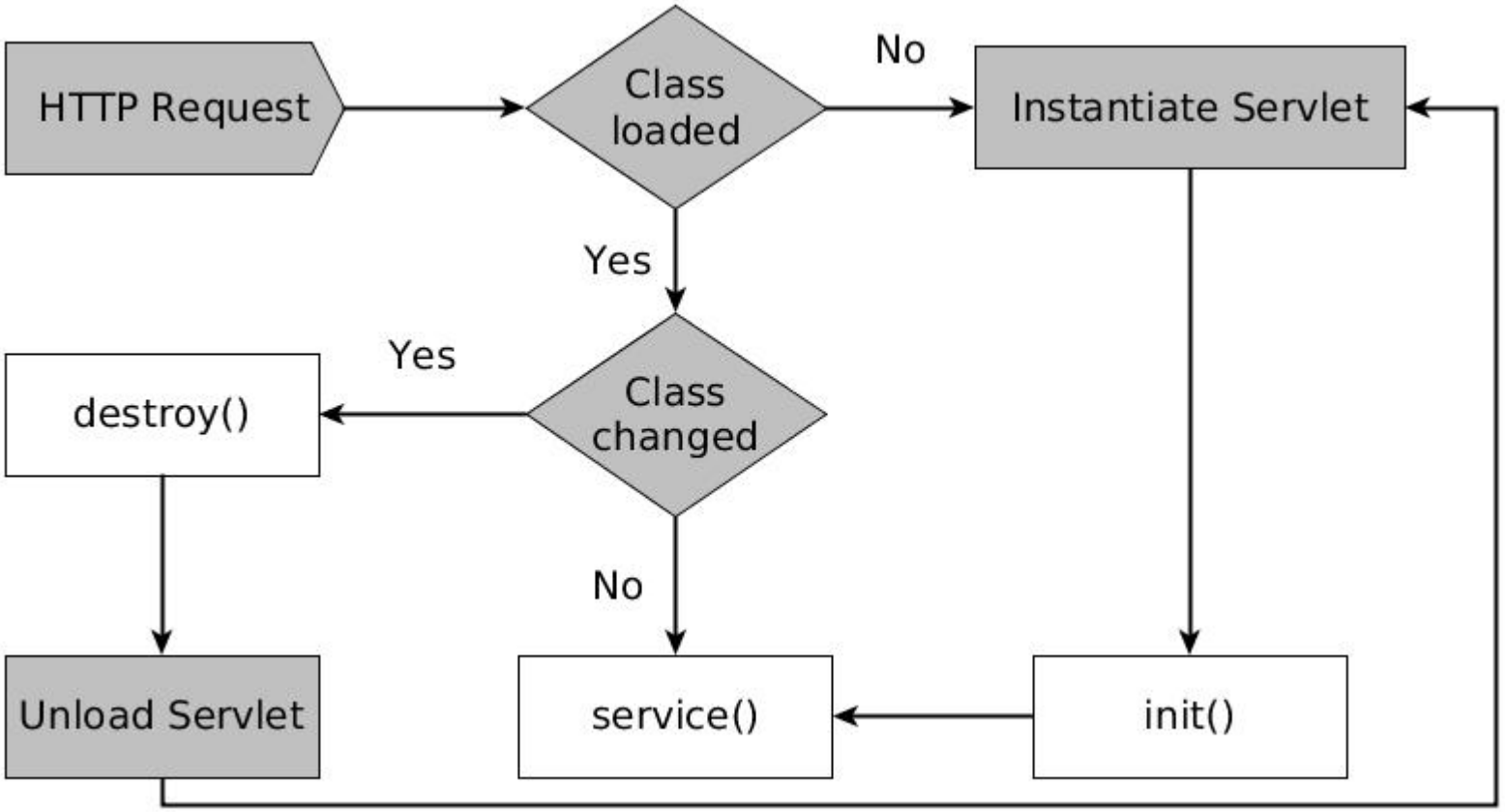
```
@WebServlet("/demo.html")
public class DemoServlet extends HttpServlet {

    public void doGet(HttpServletRequest request,
                      HttpServletResponse response) {

        Writer out = response.getWriter();
        out.write("<html><body>Text</body></html>");
        out.close();

    }
}
```

Servlet Lifecycle



Servlet Request Processing

- When the HttpServlet.service() method is invoked
  - It reads the HTTP method type in the request
  - It uses this value to determine which method to invoke

HTTP method	HttpServlet class method
GET	doGet(HttpServletRequest,HttpServletResponse)
POST	doPost(HttpServletRequest,HttpServletResponse)
PUT	doPut(HttpServletRequest,HttpServletResponse)
DELETE	doDelete(HttpServletRequest,HttpServletResponse)
HEAD	doHead(HttpServletRequest,HttpServletResponse)
OPTIONS	doOptions(HttpServletRequest,HttpServletResponse)
TRACE	doTrace(HttpServletRequest,HttpServletResponse)

Deployment to a Java EE Server

- Application is packaged to WAR file
- Then deployed to Java EE Server (Tomcat, Jetty, Wildfly, Glassfish, ...)

```
- META-INF
    - MANIFEST.MF
- WEB-INF
    - classes
        - ite
            - librarymaster
                - controller
                    - ListBookServlet.class
    - lib
        - log4j-1.2.17.jar
    - web.xml
- index.html
```

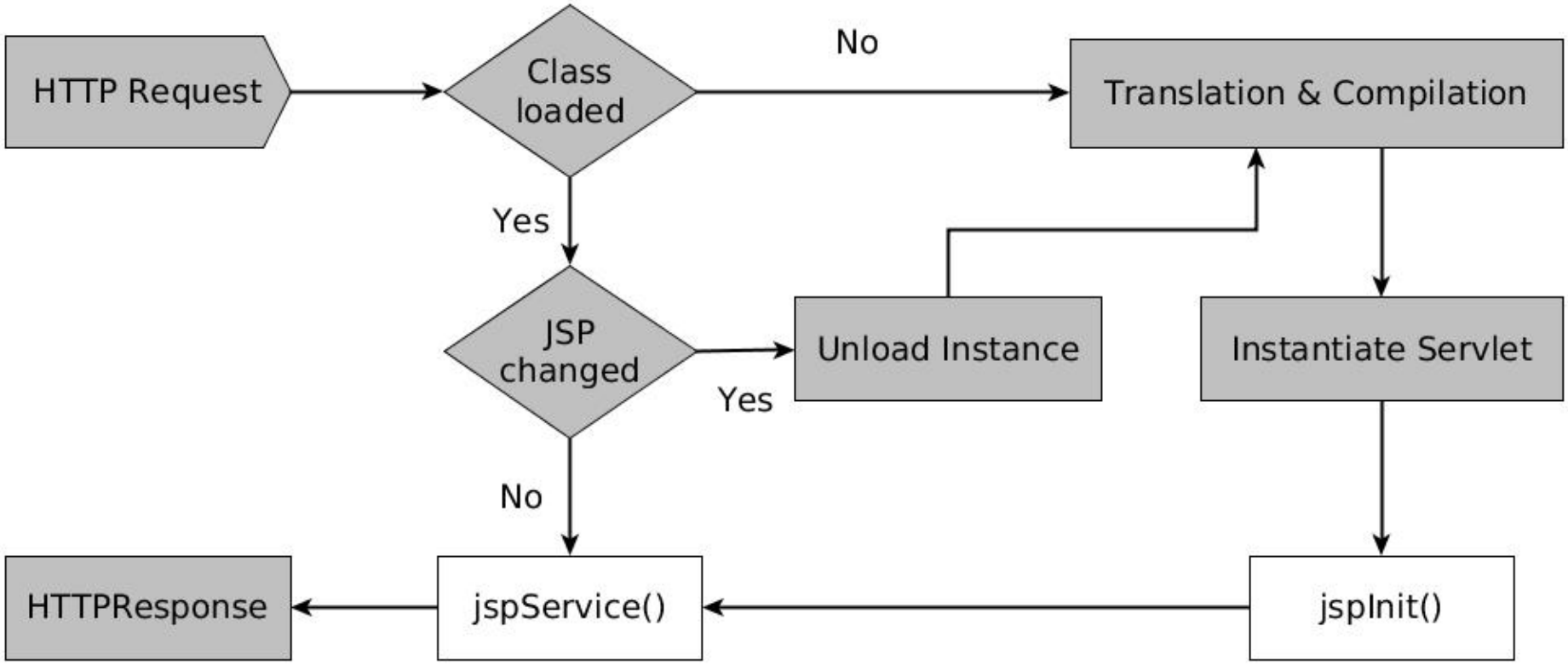
Apache Tomcat 8

- Open source Java-based Web Application container which is used to run Servlet and Java Server Pages (JSP) Web applications
  - Servlet 3.1
  - JavaServer Pages 2.3
- Integrated also in Java EE Application servers as Web container
  - JBoss EAP 6
  - Apache TomEE

Java Server Pages

- Simplified, fast way to create dynamic web content
- Better separation of business logic from view
- You write JSP page instead Java code
  - You can combine JSF tags with Java code
- You can utilize JSP Standard Tags Library (JSTL)
  - Collection of tag libraries that implement general-purpose functionality common to many Web applications
- You can use Expression Language (EL) to bind Java

JSP Lifecycle (1)



JSP Lifecycle (2)

- Translation
  - The JSP file is translated to Java Servlet source
- Compilation
  - The generated Servlet class is compiled
- Loading
  - The compiled Servlet is loaded in memory
- Instantiation
- Initialization
- Servicing request
- Destruction