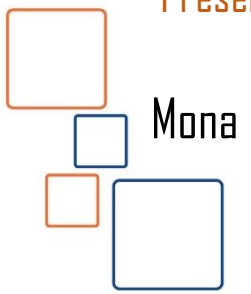
Building Dynamic Web Applications Using Servlets



Presented By

Mona Mahrous,MSc.

Java™ Education and Technology Services

Invest In Yourself, Develop Your Career

Why ?

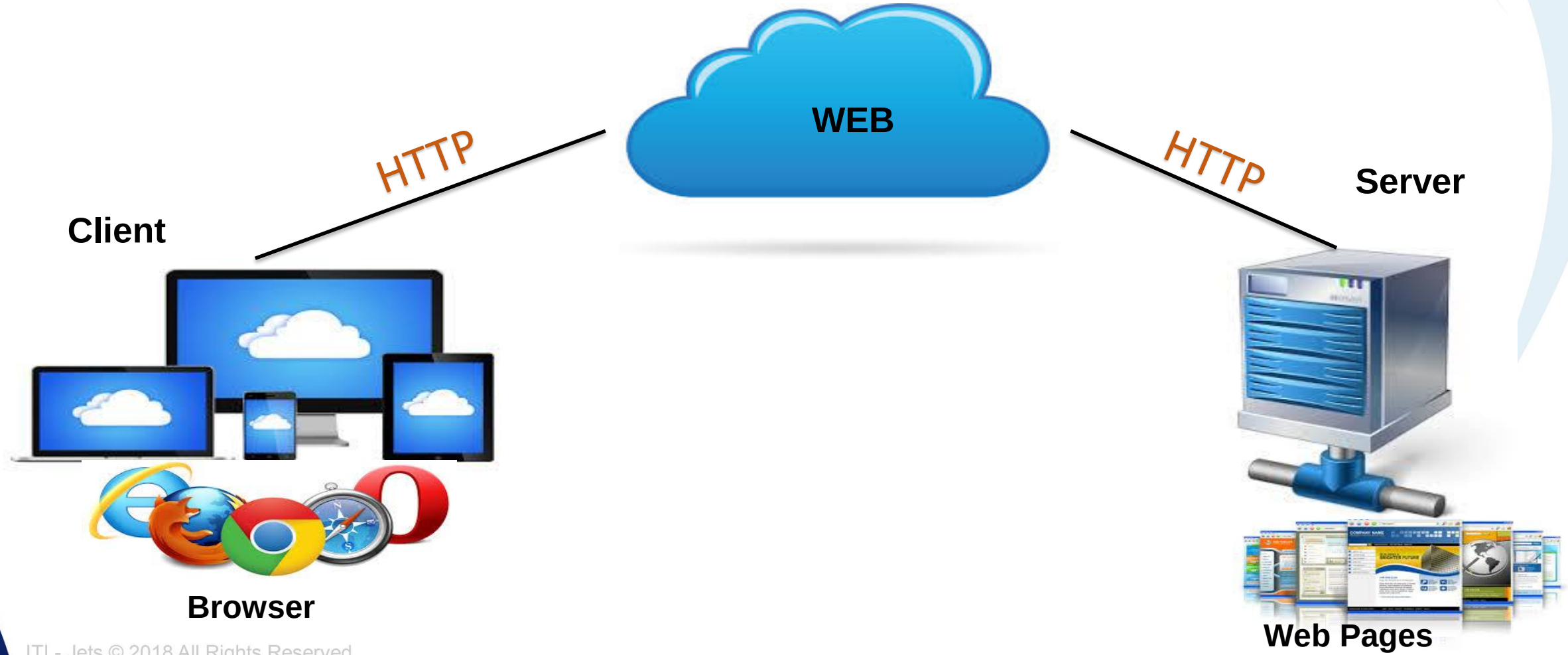
Java™ Education and Technology Services

Chapter 1

Introduction To Web Technology

Ch1 : Web World

❑ How it goes in the *Web*

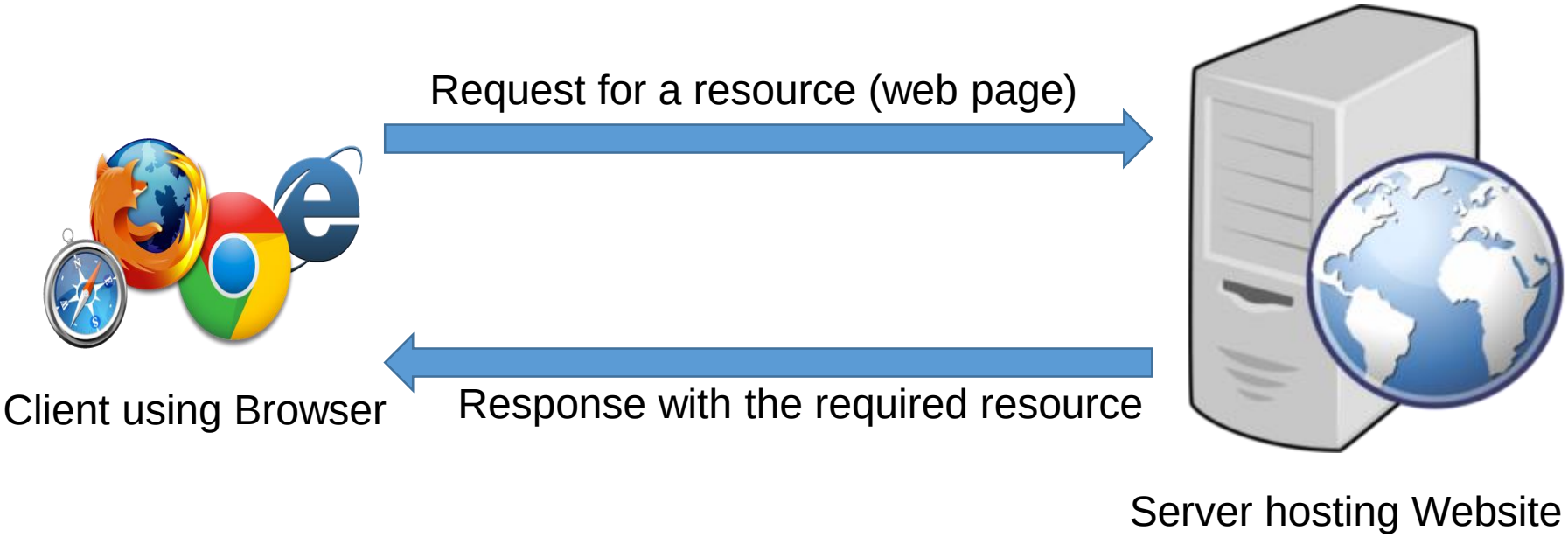


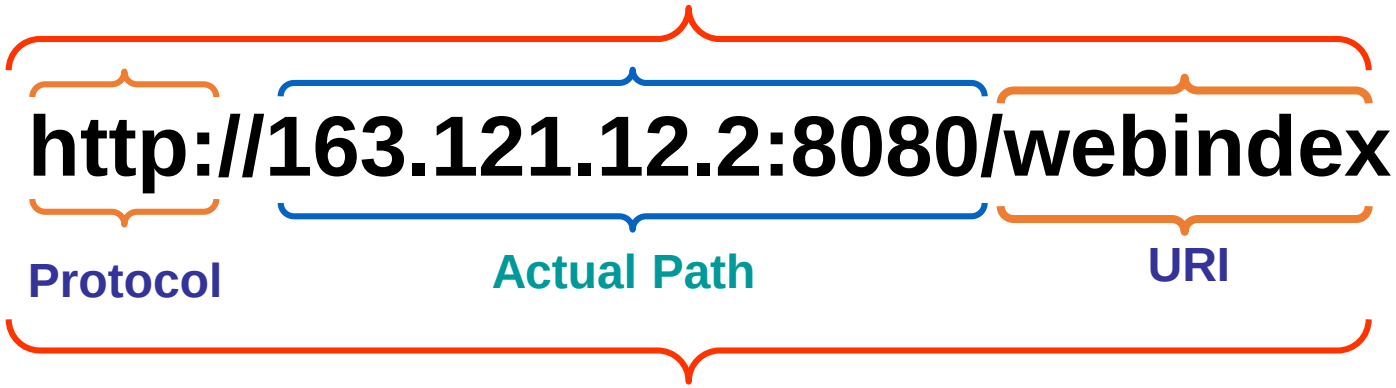
Ch1 : HTTP Protocol

❑ What is *HTTP*?

- Hyper Text Transfer Protocol.
- Stateless Protocol.
- It is an application layer protocol.
- Working as request-response protocol

Ch1 : HTTP Protocol





Ch1 : HTTP Request

Java™ Education and Technology Services

- HTTP Request consists of 3 components:

  1.Request-Line

  Method / Request-URI / Protocol-Version

  - Methods could be : Get ,Post, Head, Put, Trace, Options, Delete

  - Example:
    - Get /servlet /default.jsp HTTP/1.1
    - Post / servlet /index.jsp HTTP/1.1

2.Request Headers

- Indirectly set by the browser and sent immediately after the request line. The following are examples of request headers:

  Accept: text/plain ;text/html (Note : type/* , */*)

  Accept-language: en-ar

  Accept-Encoding : gzip

  Connection : keep- Alive

  Host : www.iti.gov.eg or localhost:8080

  Referer : http://www.mcit.gov.eg/main.htm

  User-agent : Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)

  Content-length : 33 *(in case of POST)*

  If-Modified-Since: specified date

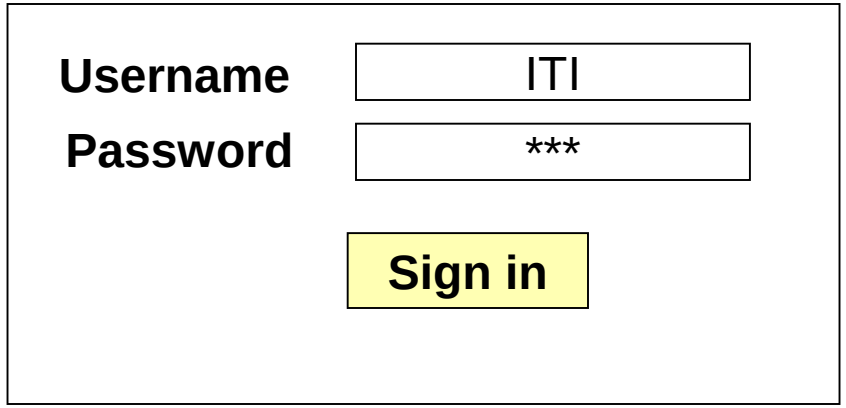
  If-Unmodified-Since : Sat, 29 Oct 1994 19:43:31 GMT

  Cookie: userID=id456578

Ch1 : HTTP Request

3. Entity Body (in case of POST method):

UserName=ITI & Password =iti



- Note: If the GET method was used:
  - There will be no entity body.
  - URL in the address bar of the browser will look something like:

  www.sitename.com/MyServlet? userName=ITI&password=iti

Ch1 : HTTP Response

- HTTP Response consists of 3 components:

  1. Status-Line

  Protocol-Version / Status Code / Description

  - Status Code Ranges:
    - 100-199 informational, the client must respond with a action
    - 200-299 : request is successful
    - 300-399 : for moving files , it includes a location header indicating the new address
    - 400-499 :indicates an error by the client
    - 500-599 : indicates an error by the server
  - **Example**
    - HTTP/1.1 200 ok
    - HTTP/1.1 404 error

Ch1 : HTTP Response

2.Response Headers

Server: Tomcat/9.5

Date: Mon, 3 Jan 2006 13:13:33 GMT

Content-Type: text/html

Last-Modified: Mon, 11 Jan 2005 13:23:42 GMT

Content-Length: 112

Content-Encoding = gzip

Ch1 : HTTP Response

3.Entity Body

```
<HTML>
  <HEAD>
      <TITLE>HTTP Response Example      </TITLE>
  </HEAD>
  <BODY>
      Welcome to servlets and jsp course ☺
  </BODY>
</HTML>
```

Java™ Education and Technology Services

Chapter 2

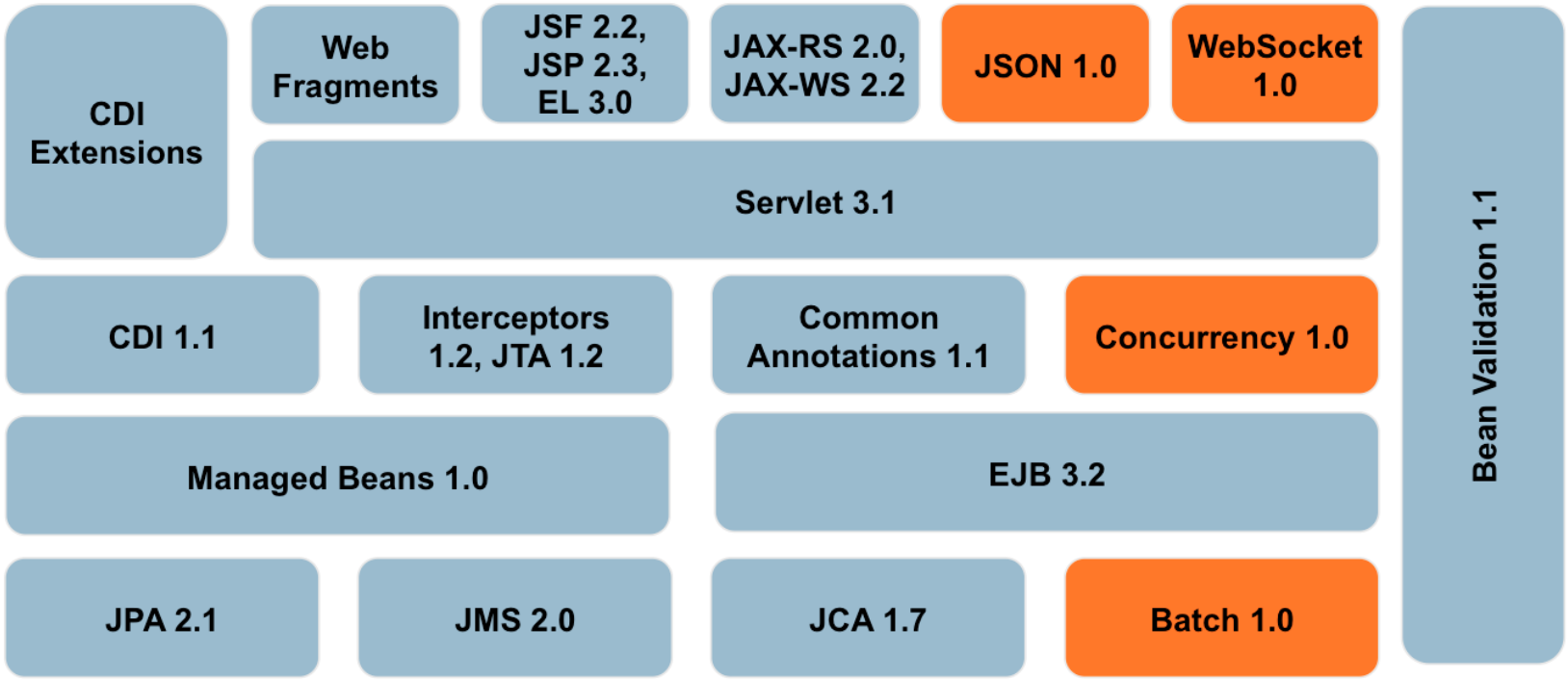
Introduction To Servlet Technology

Ch2 : What is Servlets?

- Servlet API provides a way for developers to develop dynamic web Applications.
- Servlets Ver. 1.0 was released by Sun Microsystems in 1997.
- Servlet API is part of Java EE

Ch2 : What is Servlets?

Java EE 7 APIs



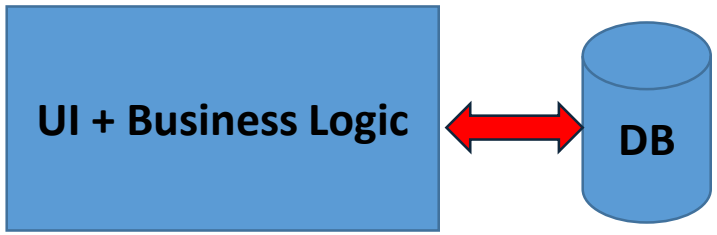
Java EE 8 (JSRs approved on 22 Sep, 2014, Final Release was available on September 2017)

Ch2 : What is Servlets?

- **A *Servlet*** is a Java program that runs on a **server**, that produces dynamic pages typically in **response** to client **requests**.

Ch2 : Application Architecture

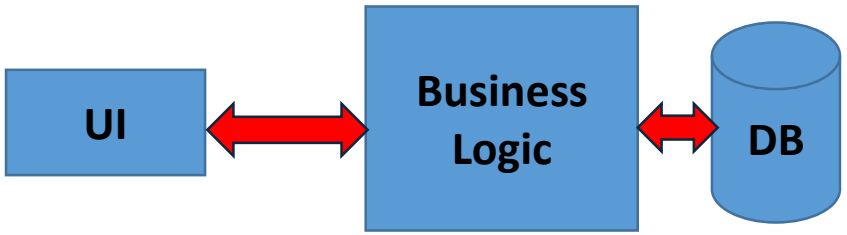
In desktop applications



Application on all users machines

What is so called Two Tier Architecture

In Web applications



User's web Browser

Server Hosting Web Application(Servlets)

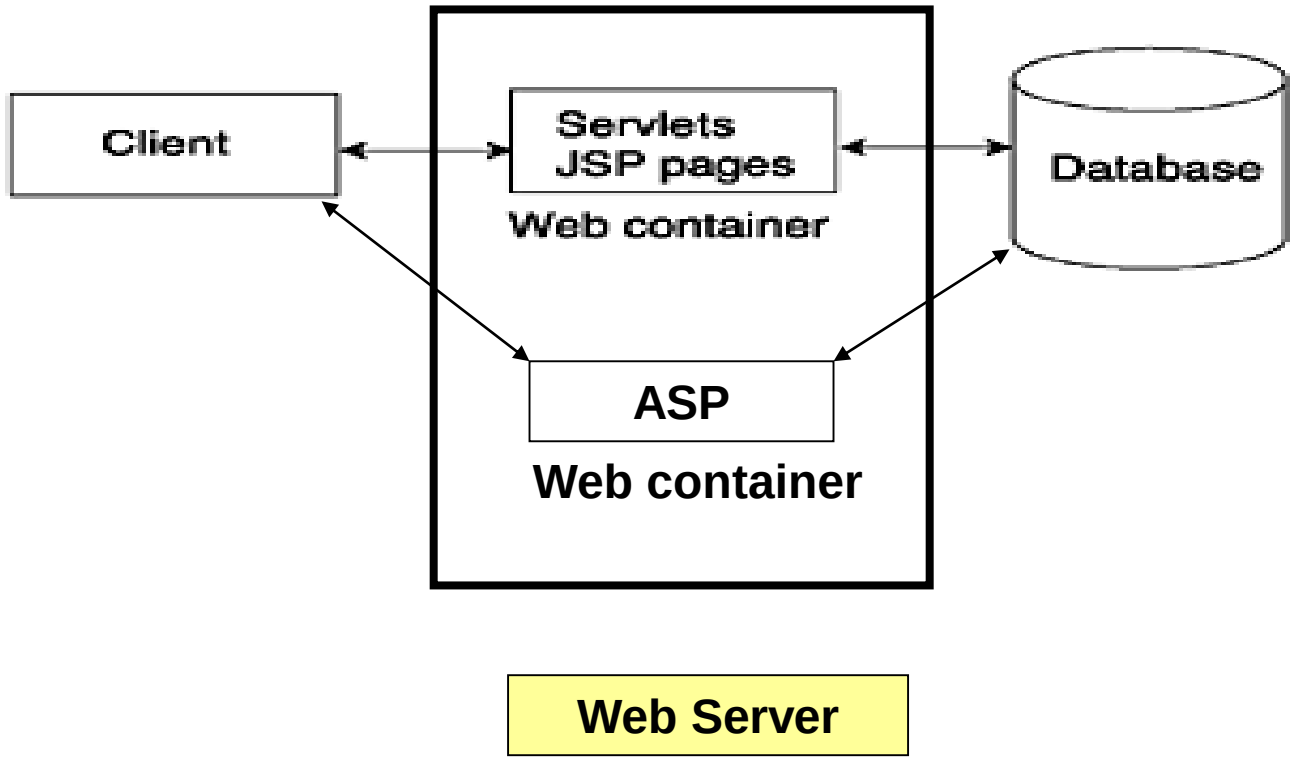
What is so called Three Tier Architecture

Ch2 : Types of Servers

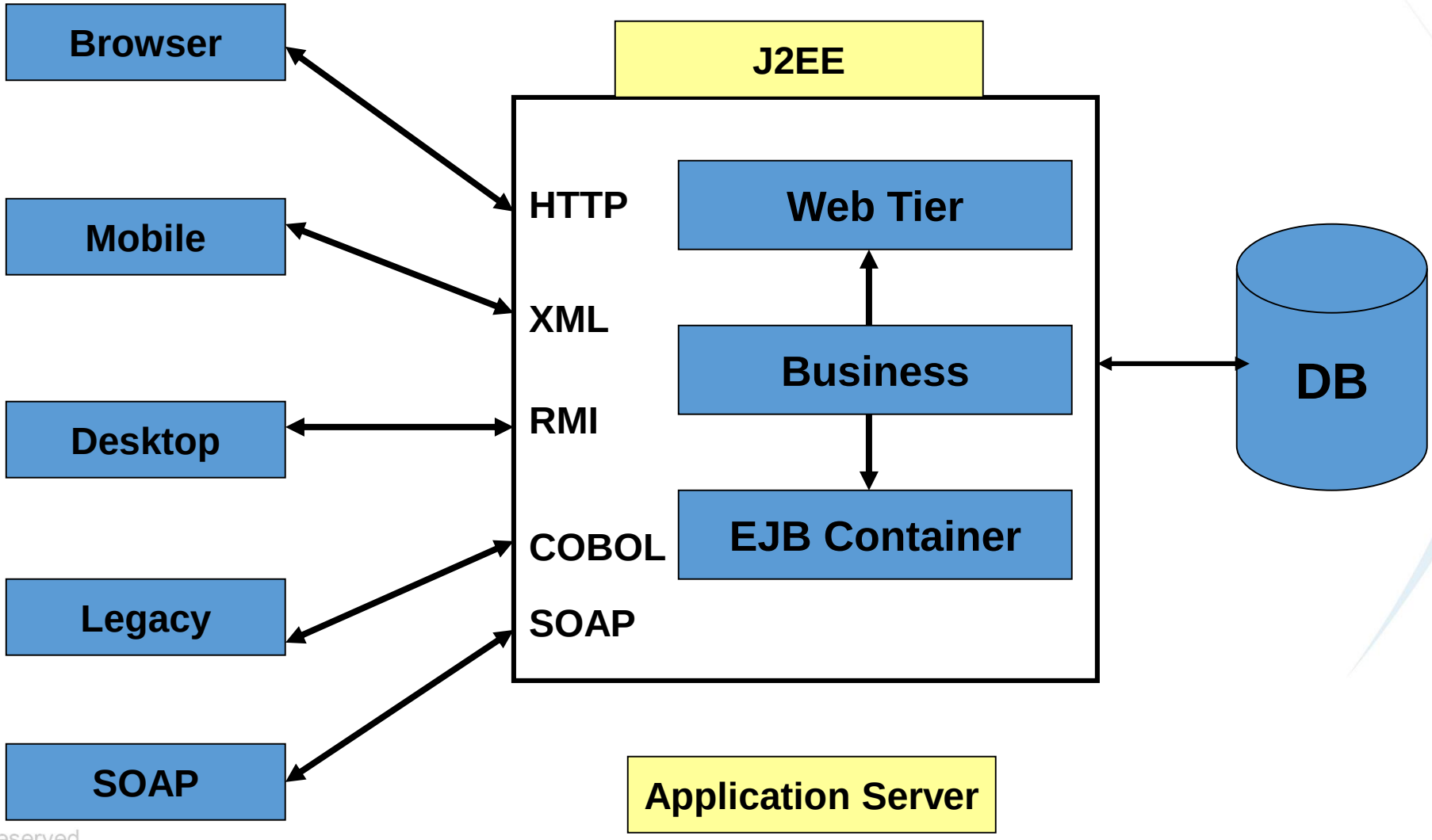
- There are two types of server that can host Web application written in Servlet
  - *Web Servers*
  - *Application Servers*

Ch2 : Types of Servers

- *Web Servers*



Ch2 : Types of Servers



Ch2 : Web Technologies

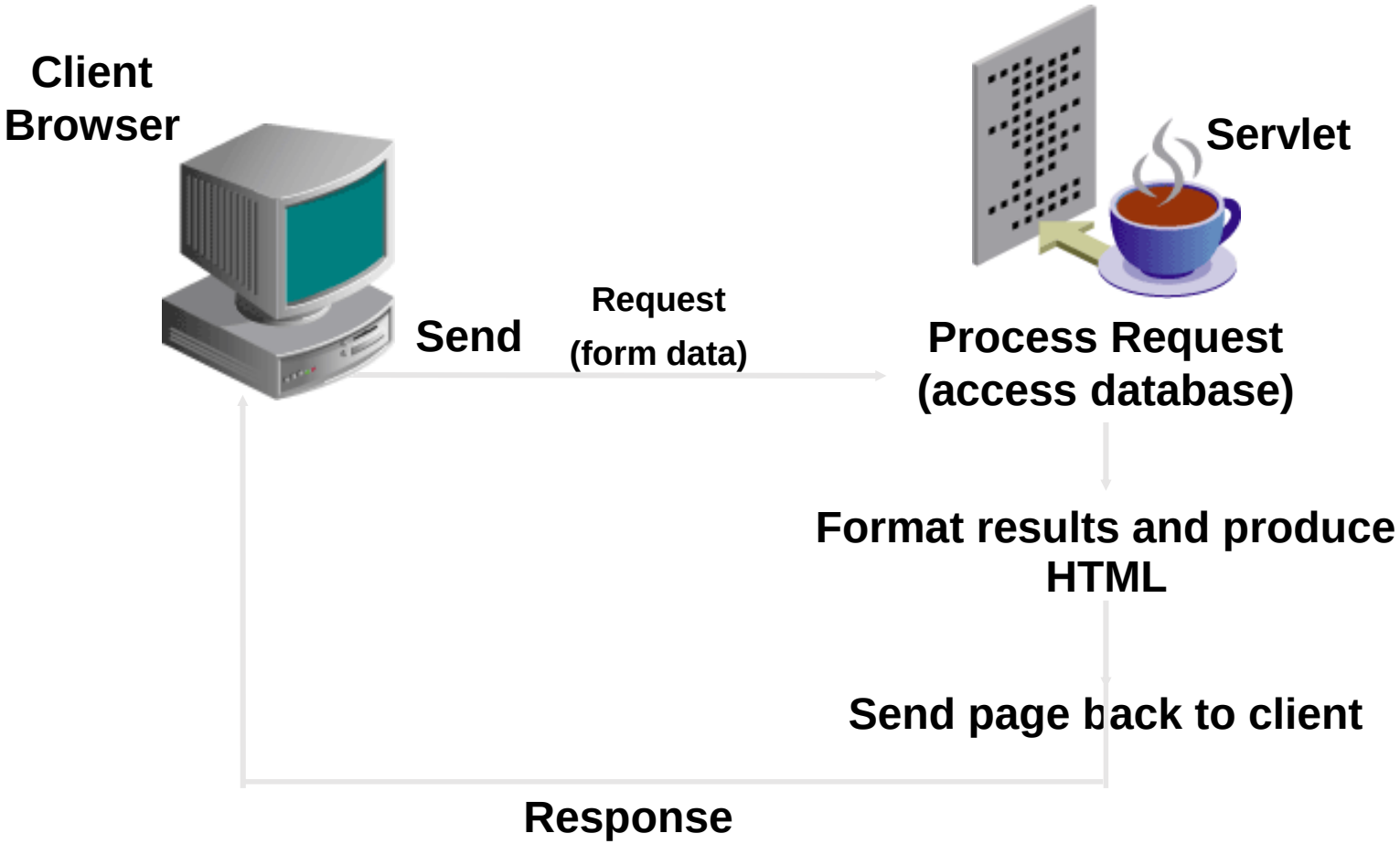
- CGI (Common Gateway Interface)
- Cold Fusion
- Server Side Java Script (SSJS)
- PHP
- Servlet
- JSP
- ASP
- ASP.NET

Java™ Education and Technology Services

Chapter 3

How Servlets Work

Ch3: How does Servlet Work?



Ch3: How does Servlet Work?

- **Servlet main jobs:**

  - ✓It reads and process data sent by the client.
  - ✓Send the data to the client with a proper format.

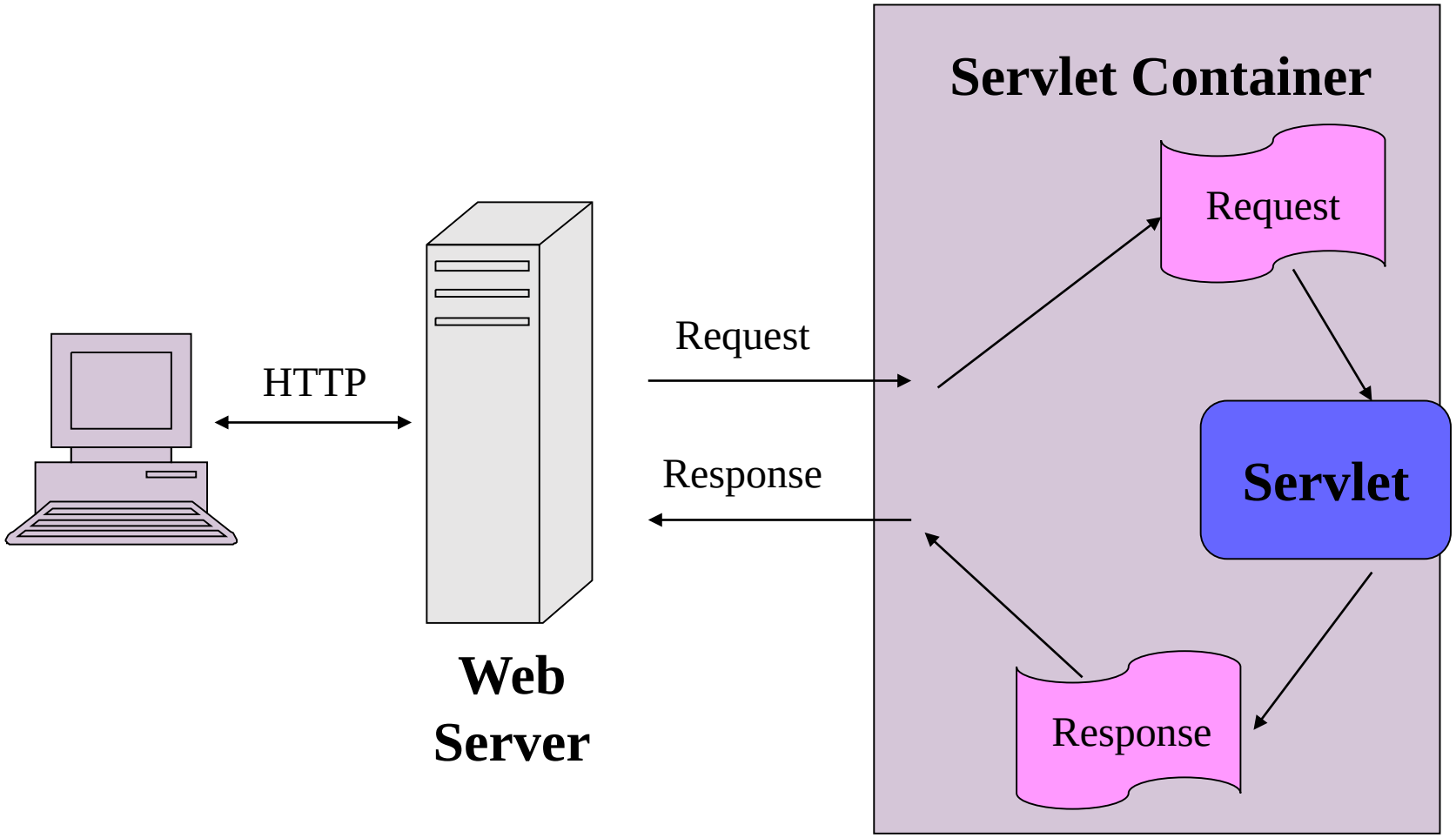
- It’s not restricted to HTTP requests

Ch3: How does Servlet Work?

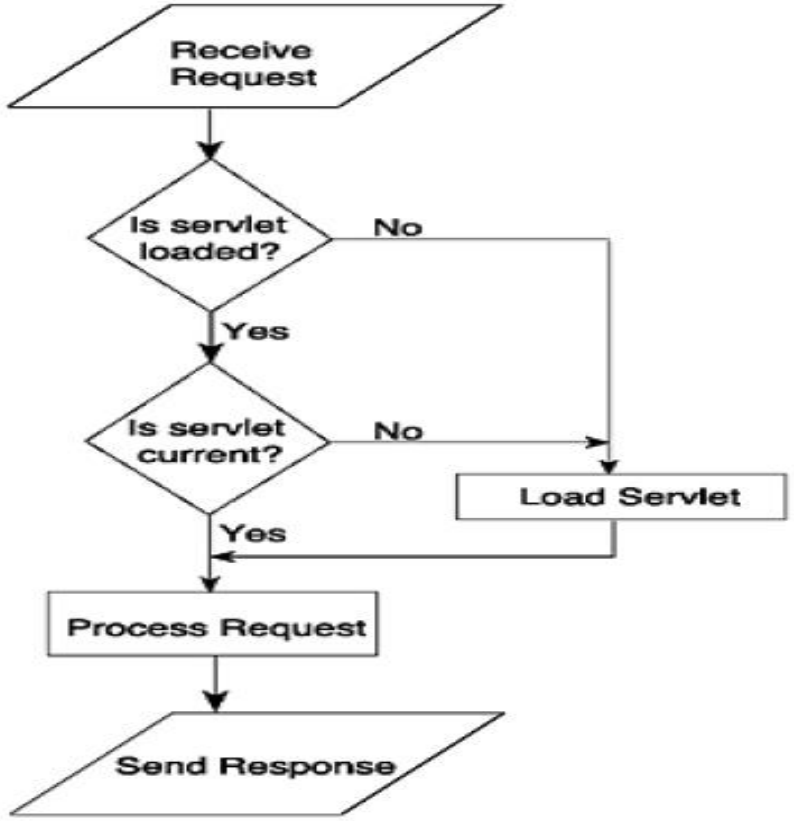
- Since Servlets is a Java Class it needs something to be responsible of the following:

  - ✓ *loading*
  - ✓ *Instantiating*
  - ✓ *unloading.*
  - ✓ *Managing servlet's life cycle.*
  - ✓ *creates and manages request and response objects*
  - ✓ *And this is the job of the Servlet Container*

Ch3: Servlet Container



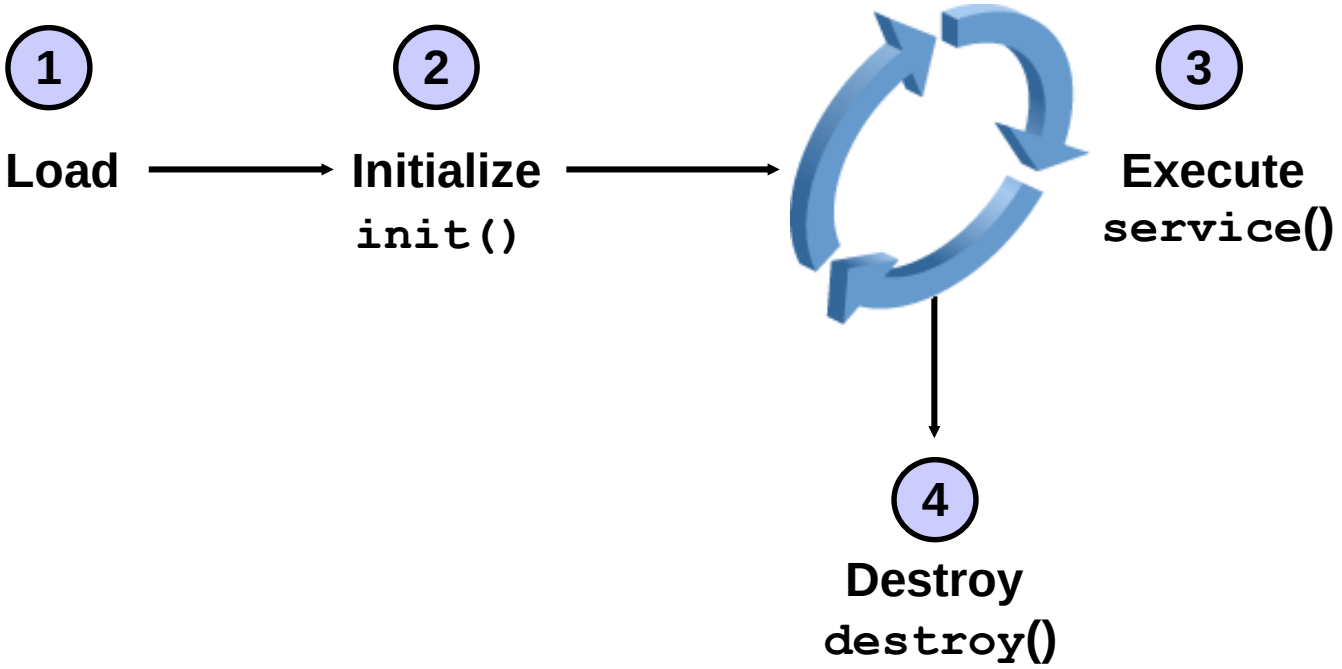
Ch3: Servlet Container



Ch3: Servlet Container

- Servlet containers:
  - Apache Tomcat
  - GlassFish
  - WebSphere
  - JBoss
  - WebLogic

Ch3: Servlet Lifecycle



Ch3: Benefits of Servlets

- Performance
- Portability - Widespread acceptance
- Rapid development cycle
- Robustness
- Secure
- Inexpensive

Ch3: Web application structure

- Any servlet/jsp application must contains
  - ***WEB-INF*** Folder which may contains:
    - *classes* folder: representing the servlets after compilation (only if the application contains servlets)
    - Deployment descriptor (*web.xml*)
    - *lib* folder
  - Also the web application may optionally contains:
    - images, html , tld and jsp pages

Ch3: What is the deployment descriptor ?

- Contains meta-data for a Web Application and that is used by the container when loading a Web Application.

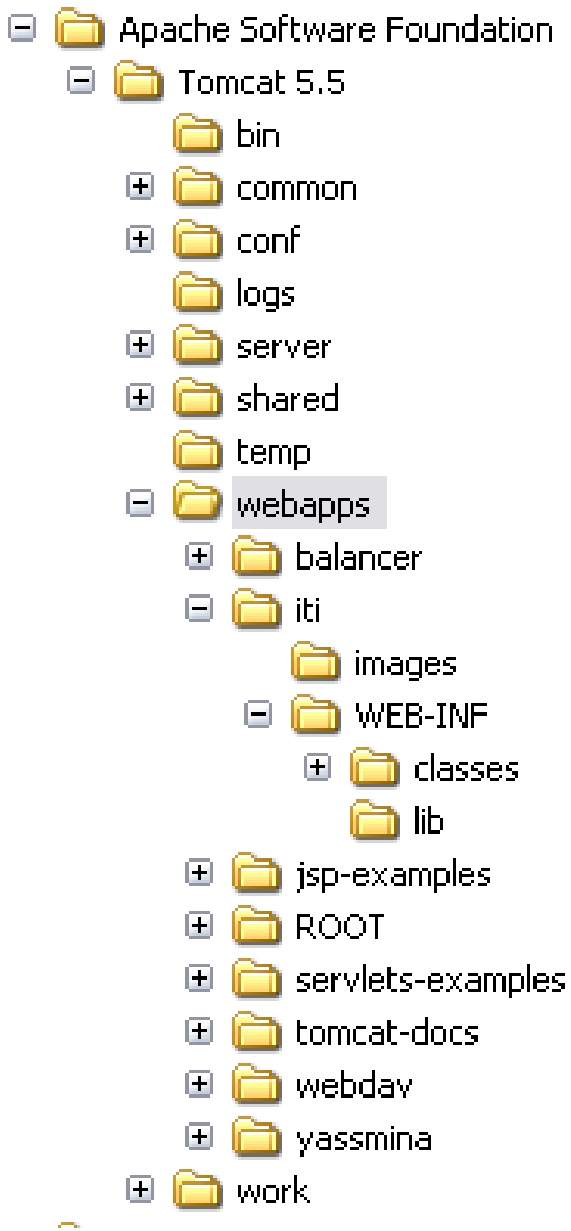
  - ✓It identifies all the elements contained in the web application for the servlet container to be able to know them
  - ✓It maps them into URL patterns to be able to call them as clients
  - ✓It contains configuration settings specific to that application.

Ch3: What is the deployment descriptor ?

```
<web-app xmlns= “https://jakarta.ee/xml/ns/jakartaee” xmlns:xsi= "http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation= "https://jakarta.ee/xml/ns/jakartaee    https://jakarta.ee/xml/ns/jakartaee/web-app_5_0.xsd"
 version= "5.0“   metadata-complete= “false">
      <servlet>
         <servlet-name>Testing</servlet-name>
         <servlet-class>TestingServlet</servlet-class>
      </servlet>

      <servlet-mapping>
           <servlet-name>Testing</servlet-name>
           <url-pattern>/hi</url-pattern>
        </servlet-mapping>

       <welcome-file-list>
           <welcome-file>index.html</welcome-file>
      </welcome-file-list>

</web-app>
```

Ch3: Tomcat setup and structure

- Making your first servlet:
  - There are six steps to do so :
    1. Create a directory structure under tomcat for your application



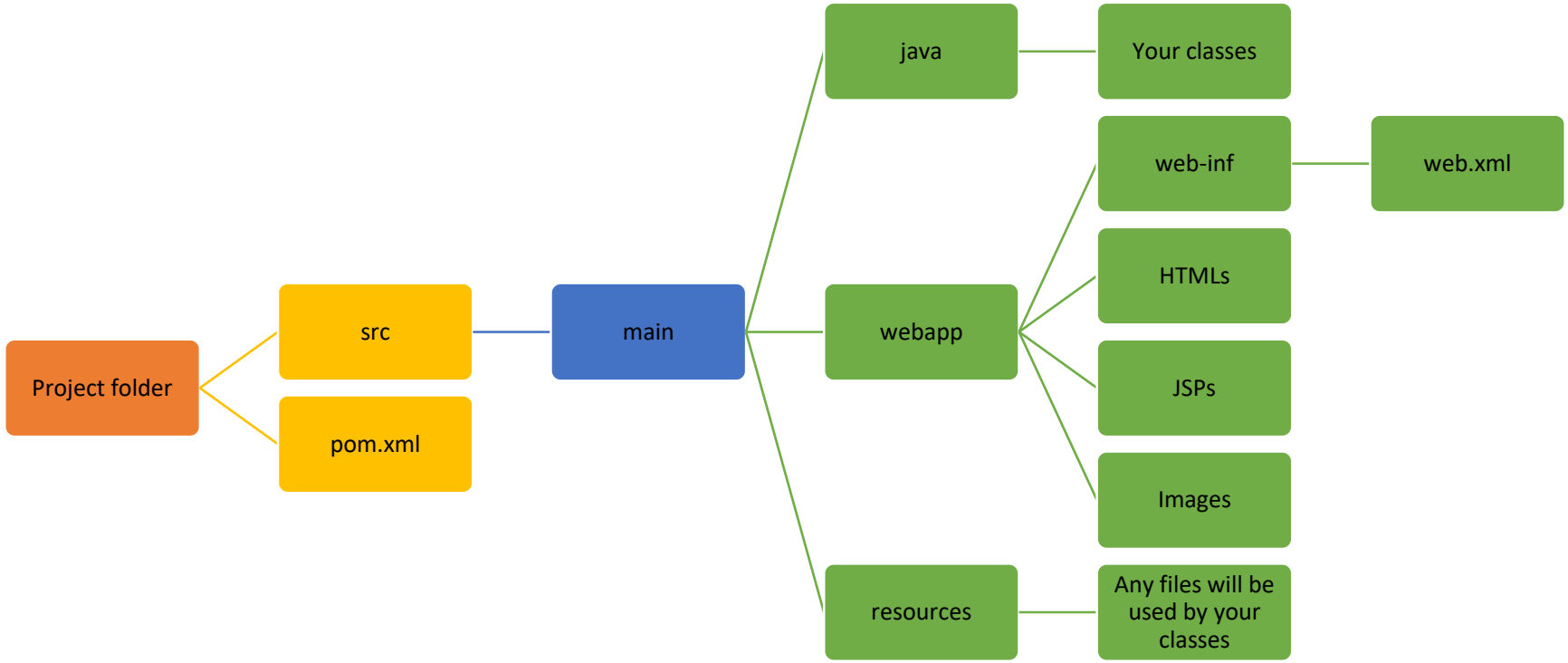
Ch3: Making your first servlet

2. Write your servlet Code
3. Compile your servlet
4. Remember that we have to include the servlet and jsp libraries to the class path
   - *javac -classpath C:…\tomcat\common\lib\servlet.jar TestingServlet.java*
   - *Or may it permanently in the classpath environment variable*
   - *Or put it in jdk/jre/lib/ext*

Ch3: Making your first servlet

or you can simply use Maven

Now the folder structure of the project will look like this:



Ch3: Making your first servlet

Java™ Education and Technology Services

Now in pom.xml use war for packaging

```
<packaging>war</packaging>
```

Use the following dependency for Jakarta EE 9

```
<dependency>
        <groupId>jakarta.platform</groupId>
        <artifactId>jakarta.jakartaee-api</artifactId>
        <version>9.0.0</version>
        <scope>provided</scope>
</dependency>
```

Ch3: Making your first servlet

Then add the following plugins

This one for compiling

```
<plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-compiler-plugin</artifactId>
        <version>3.8.1</version>
        <configuration>
          <source>11</source>
          <target>11</target>
        </configuration>
</plugin>
```

Ch3: Making your first servlet

This one for generating war file

```
<plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-war-plugin</artifactId>
        <version>3.3.0</version>
        <configuration>
          <failOnMissingWebXml>false</failOnMissingWebXml>
        </configuration>
 </plugin>
```

Ch3: Making your first servlet

And you can use this plugin for deploying:

```
<plugin>
        <groupId>org.apache.tomcat.maven</groupId>
        <artifactId>tomcat7-maven-plugin</artifactId>
        <version>2.2</version>
        <configuration>
                <username>admin</username>
                <password>admin</password>
                <url>http://localhost:9191/manager/text</url>
                <path>/test</path>
        </configuration>
</plugin>
```

The credentials of the tomcat user with “manger-script” and “manger-gui” roles

The URL of you tomcat server, so the plugin can deploy the application

The context name that will be used to access the application after deployement

Ch3: Making your first servlet

```
<web-app xmlns= “https://jakarta.ee/xml/ns/jakartaee “   xmlns:xsi=
   "http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation= "https://jakarta.ee/xml/ns/jakartaee    https://jakarta.ee/xml/ns/jakartaee/web-
   app_5_0.xsd"
 version= "5.0“   metadata-complete= “false">

      <servlet>
         <servlet-name>Testing</servlet-name>
         <servlet-class>MyFirstServlet</servlet-class>
      </servlet>

      <servlet-mapping>
            <servlet-name>Testing</servlet-name>
            <url-pattern>/MyTest</url-pattern>
        </servlet-mapping>


</web-app>
```

Ch3: Making your first servlet

5. Run the Tomcat
6. If you are using maven run the command

```
mvn install tomcat7:deploy
```

5. Call your Servlet from the Web Browser
   - *http://domain-name/virtual-directory/servlet-name*

   Example:

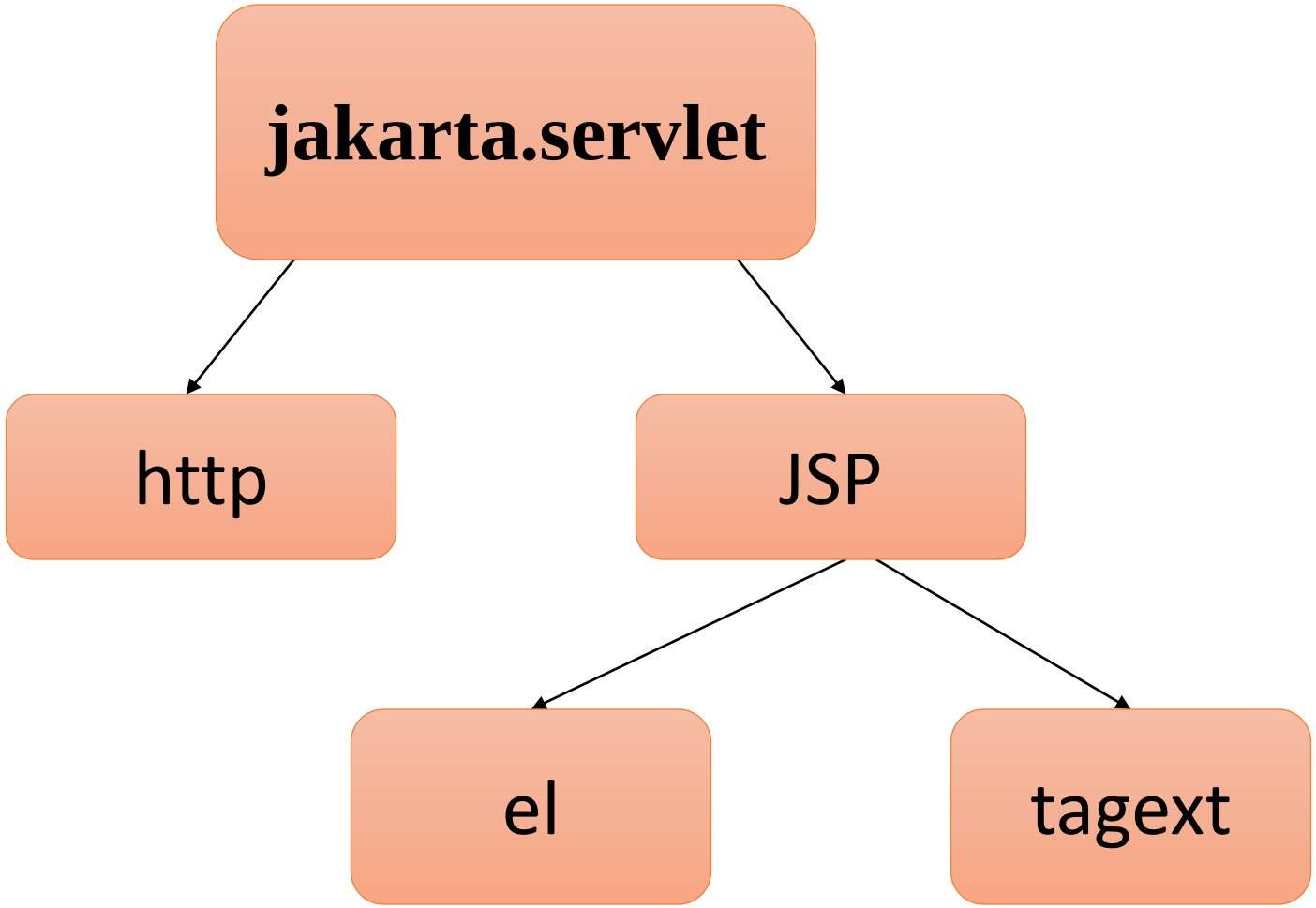
   http://localhost:8080/iti/MyTest

Java™ Education and Technology Services

Chapter 4

How to write Servlets Code

Ch4 : Servlet Package Structure



Ch4 : javax.servlet.*

- Some basic interfaces :
  - Servlet
  - ServletConfig
  - ServletContext
  - ServletRequest
  - ServletResponse
  - SingleThreadModel
  - Request Dispatcher

Ch4 : javax.servlet.*

- Some basic classes :
  - GenericServlet
  - ServletInputStream
  - ServletOutputStream
- The Exception Classes are:
  - ServletException
  - UnavailableException

Ch4 : Servlet Interface

- Servlet : it’s the basic interface , any servlet must implements it directly or indirectly
- It consists of 5 main methods :
  - init (ServletConfig config)
  - service(ServletRequest , ServletResponse)
  - destroy()
  - getServletConfig()
  - getServletInfo()

Ch4 : Servlet Example

```
import javax.servlet.*;
import java.io.*;
public class MyServlet implements Servlet
{
  public void init(ServletConfig config) throws ServletException
  {
        System.out.println("I am inside the init method");
  }
  public void service(ServletRequest request,
                              ServletResponse response)
                                throws ServletException, IOException
   {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<br>Welcome to Servlets and JSP Course");
        System.out.println("I am inside the service method");
  }
```

Write to the response

Ch4 : Servlet Example

```
  public void destroy()
  {
        System.out.println("I am inside the destroy method");
  }

  public String getServletInfo()
  {
        return null;
  }

  public ServletConfig getServletConfig()
  {
        return null;
  }
}
```

Java™ Education and Technology Services

Lab Exercise

Lab Exercise

- Make a welcome home page file.html which contains a link to your first servlet.