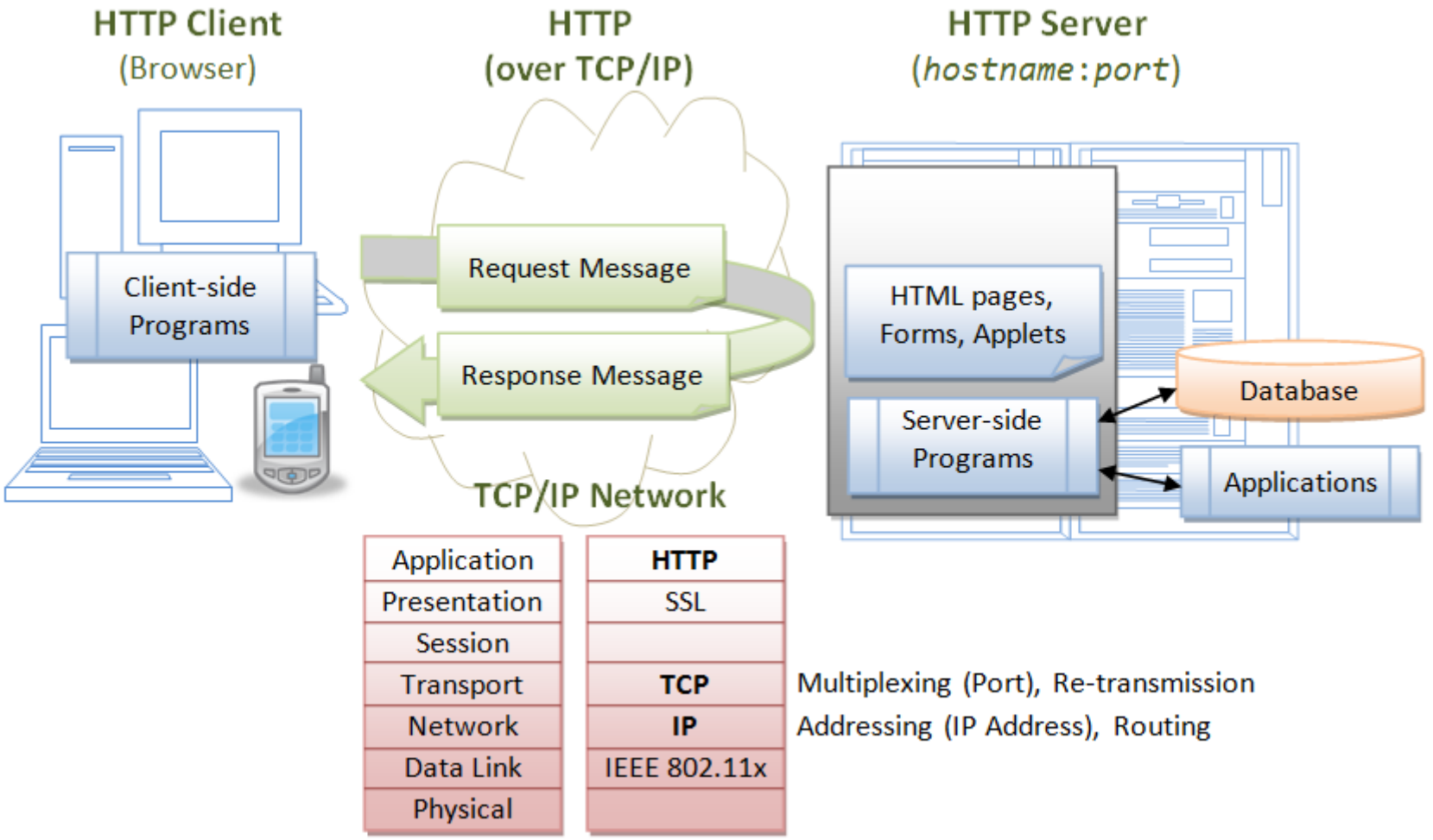
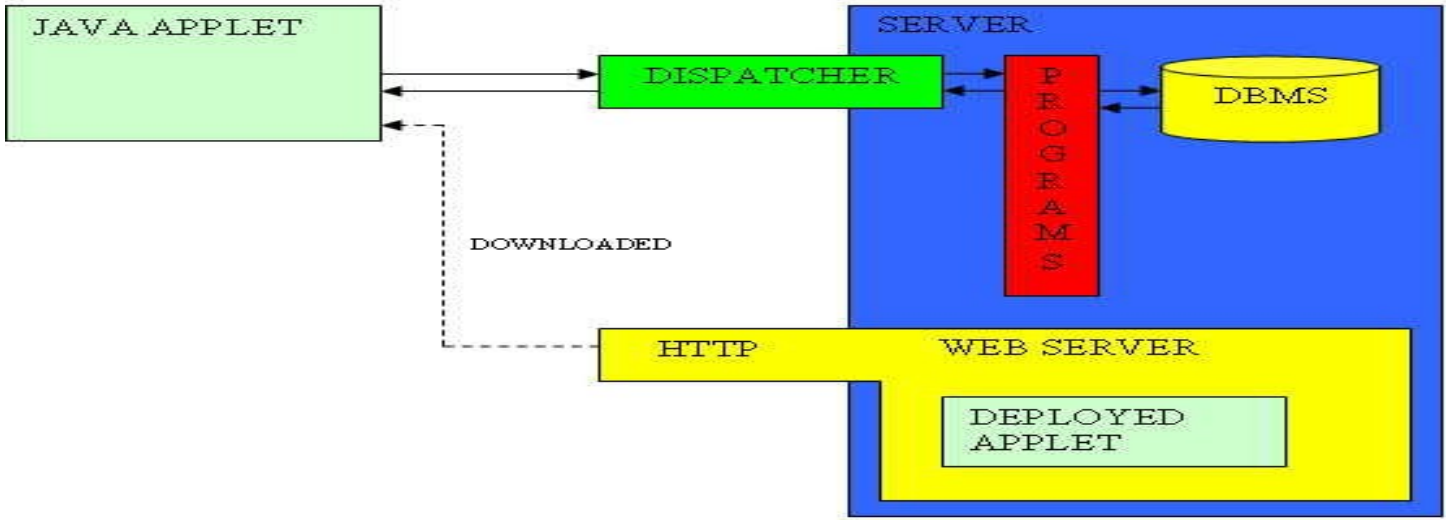
Servlets

- Client server Architecture :-

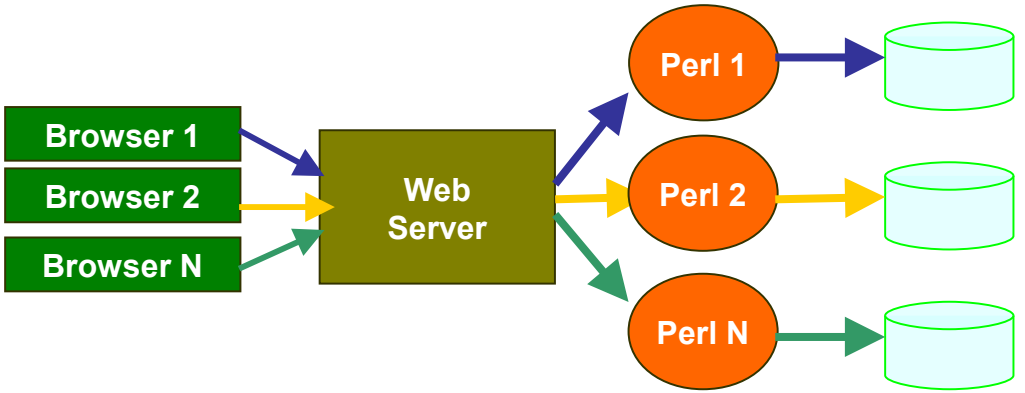
Accessing static data



- Applets :-
  - better GUI but applet is downloaded to the client, and executes in a JRE inside the browser, and can display whatever it wants to display within the applet frame.
  - Program output directly embedded into webpages, browser point to it and byte code downloaded to client's browser and then executed.
  - For small program comfortable
  - For large programs heavy(time consuming).

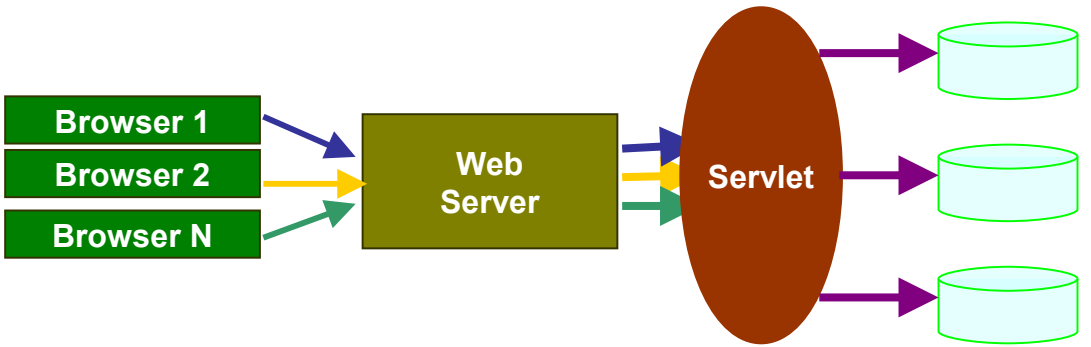


- # CGI :- Common Gateway Interface
  - For each request it starts a new process. (process based)
  - CGI program needs to be loaded and started for each CGI request.
  - More time to send response
  - Platform dependent
  - No cookie handling, session tracking



- # Servlets:-

- Instances are created instead of each process creation request. (thread based)
- Dynamic response based on user's request.
- Runs on server side with help of request and response objects.
- So applets which run on servers side of connection.
- Java servlets typically run on the HTTP protocol. HTTP is an asymmetrical request-response protocol. The client sends a request message to the server, and the server returns a response message.



Performance superior to CGI as no new process creation for each client request.

Due to its distinct and unique multi-threading capability, it is possible for hundreds and even thousands of users can access the same servlet simultaneously without affecting the load of the web server.

Platform independent

Both Presentation and Controller/Logic are separate separation

An extension of the Java platform, they can access whole Java API, access to the rich Java library that will help speed up the development process.

Managed by the Java Virtual Machine

- Hierarchy of the servlet :-

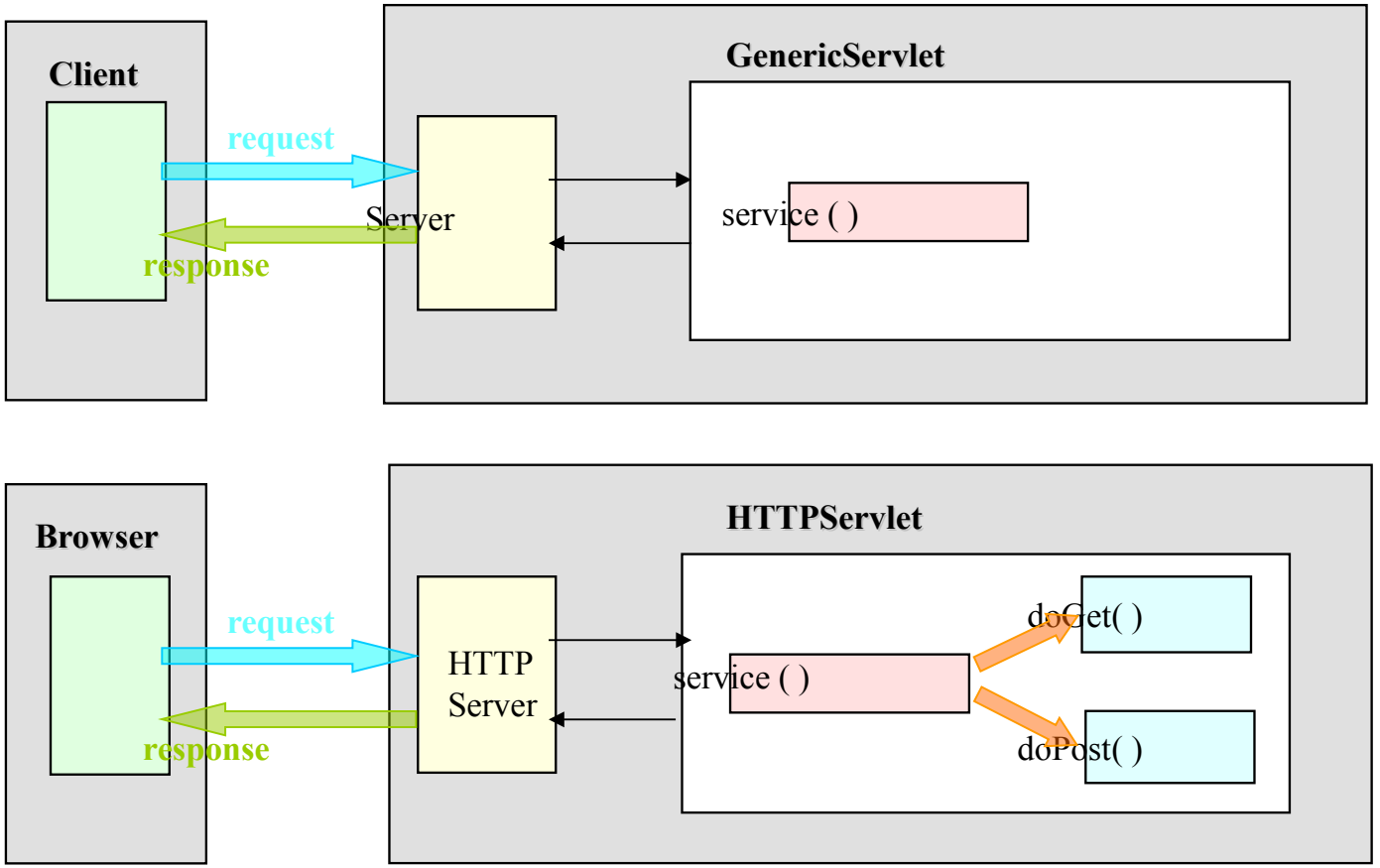
Servlet

|

GenericServlet

|

HttpServet



- Generics Servlet:- javax.servlet package

- contains protocol independent classes and interfaces so can handle any type of requests
- Implements Servlet, ServletConfig and java.io.Serializable interfaces
- No support for cookies.
- HttpSession and Session tracking are not supported.

- Http servlet :- javax.servlet.http package
- contains classes and interfaces that are specific to HTTP
- Extends GenericServlet and implements java.io.Serializable.
- Inherits properties of Generic Servlet
- subclass of HttpServlet must override at least one method of doGet(), doPost(),doPut(), doDelete(), init(), destroy(), getServletInfo()

Tasks done by servlet :-

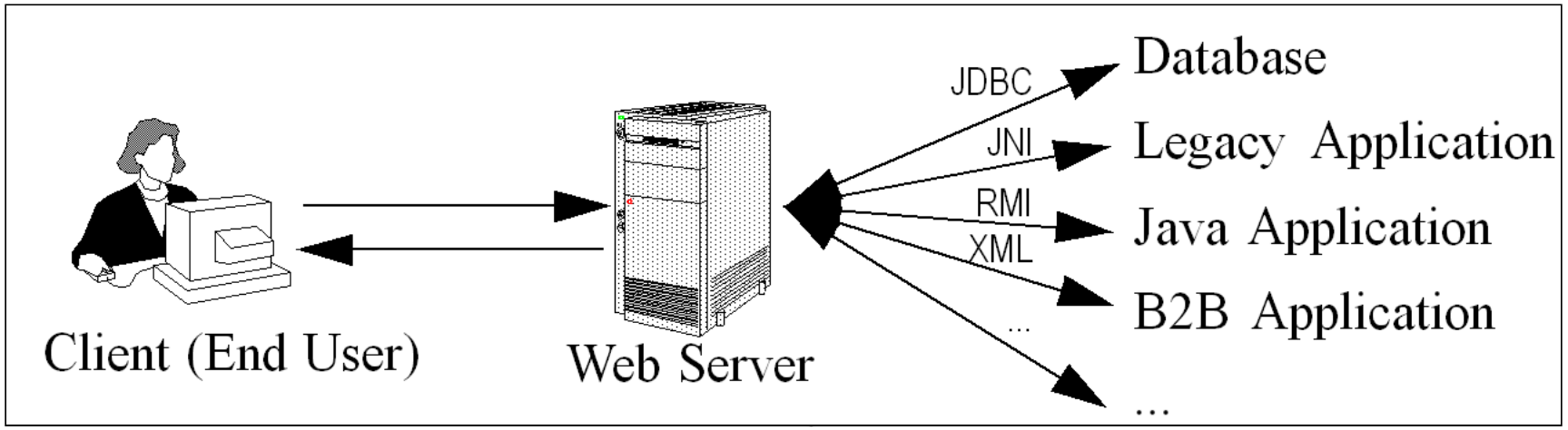
Send implicit data sent by client

Send explicit data sent by client

Generate results

Send the explicit data back to client

Send the implicit data to client



- Form data
-
- Request headers
-
- HTML content
-
- status codes and response headers

Lifecycle:-

- Controlled by the container in which the servlet has been deployed. When a request is mapped to a servlet, the Web container performs the following steps:-
- Loads the servlet class.
- Creates an instance of the servlet class.
- Initializes the servlet instance by calling the init method.
- Invokes the service method, passing a request and response object.

container loads servlet when request is made with reference of web.xml

After loading of the servlet, the container creates the instances of the servlet.

container calls the init() method and passes the parameters to the init() method.

only once throughout lifecycle

Available for service if loaded successfully otherwise the servlet container unloads the servlet

Creates seperate threads for each request.

The sevlet container calls the service() method for servicing requests.

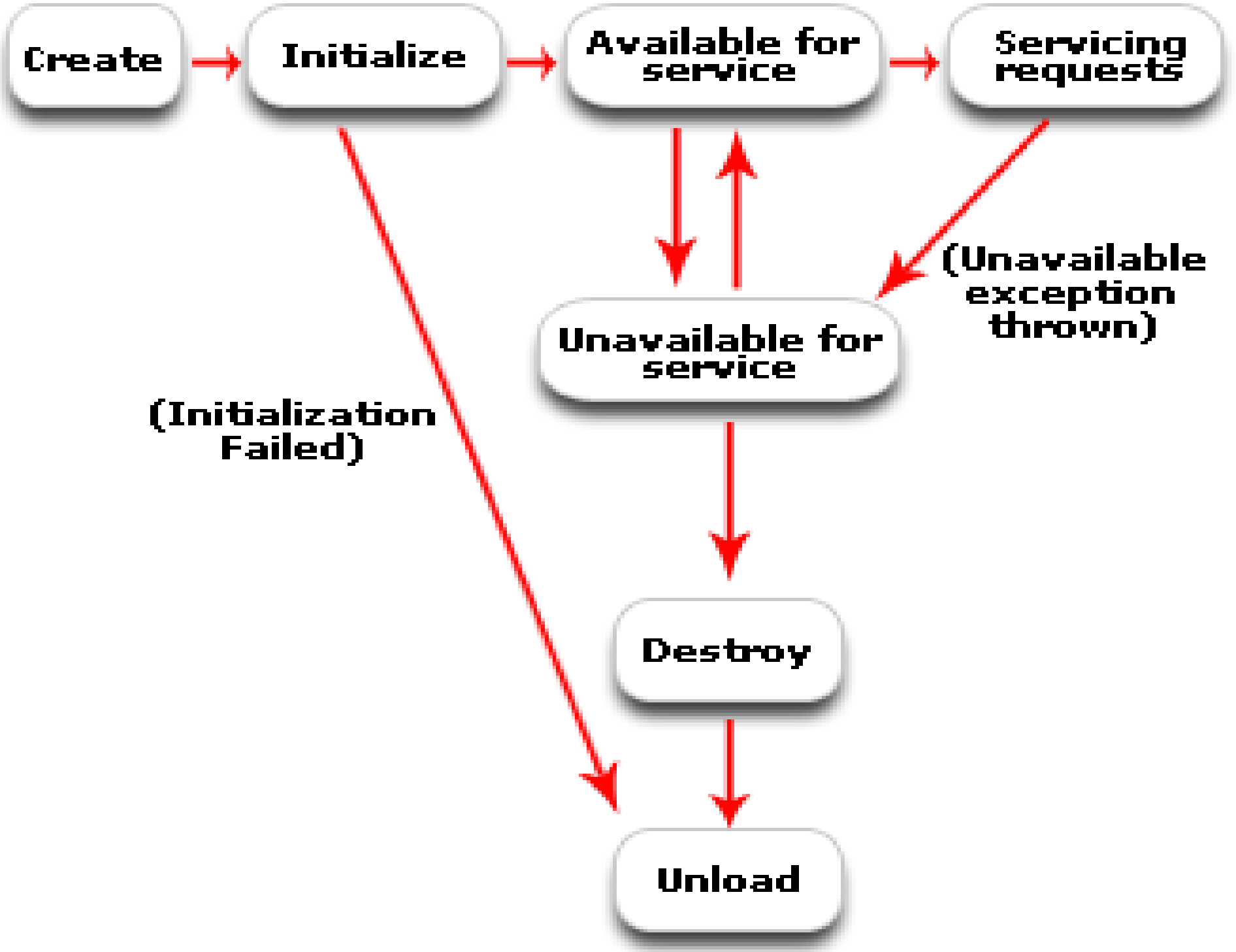
According to kind of requests calls the appropriate method (doGet() or doPost()) and sends response to the client using the methods of the response object.

Wait for another request if no more requests are there calls the destroy().

Releases all the resources associated with it, JVM claims memory associated with resource for garbage collection.

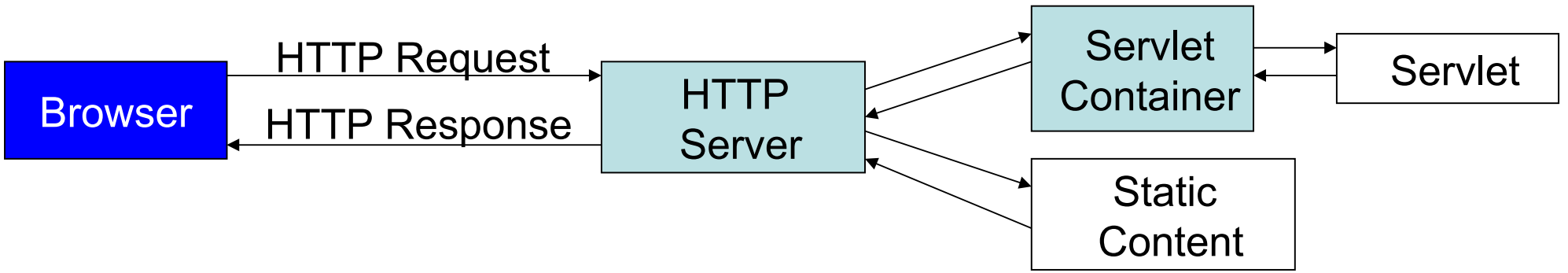
Container calls it once, we can any number of times.

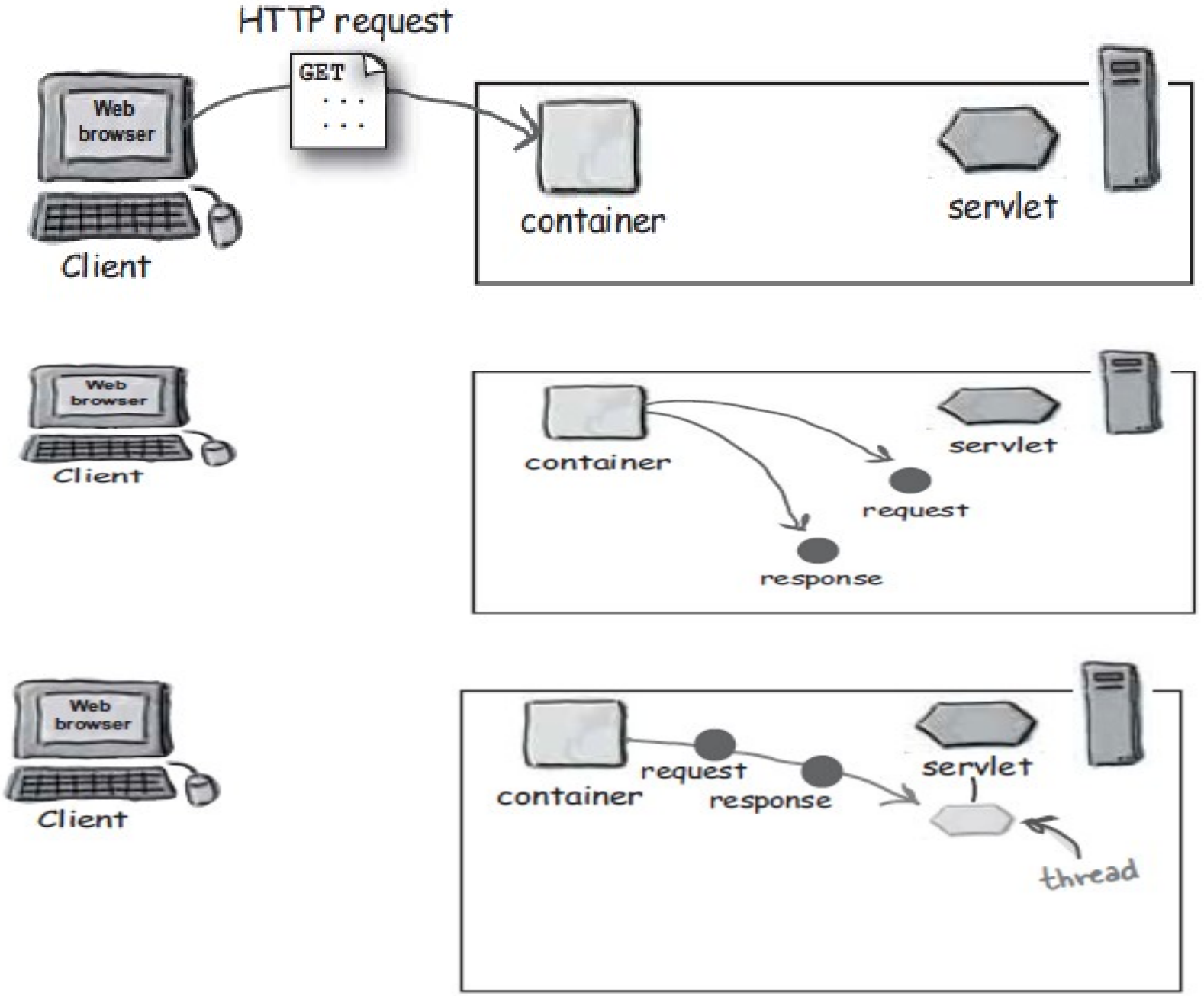
- container loads servlet with reference of web.xml when request is made
- creates the instances of the servlet and call init() passes parameters.(once)
- If loads successfully, available for service. Container calls the service() method for servicing requests.
- Creates separate threads for each request.
- According to kind of requests calls the appropriate method (doGet() or doPost()) and sends response to the client using the methods of the response object.
- Wait for more requests, if no more, destroy(), Container calls it once, we can any number of times.
- Release resources and claimed for Garbage collection

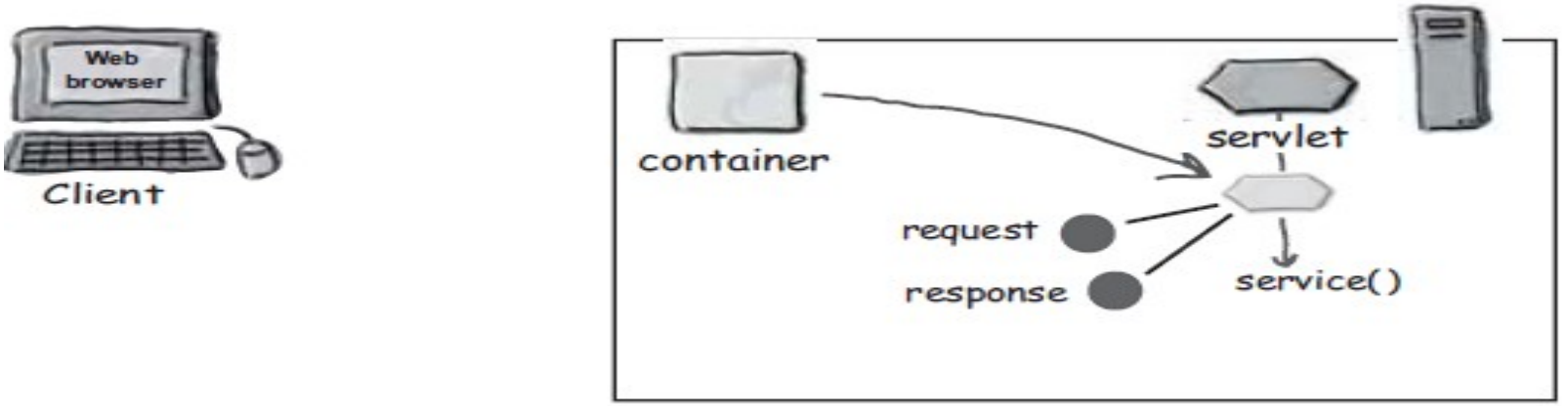

Create
Initialize
Available for service
Servicing requests
(Unavailable exception thrown)
Unavailable for service
(Initialization Failed)
Destroy
Unload

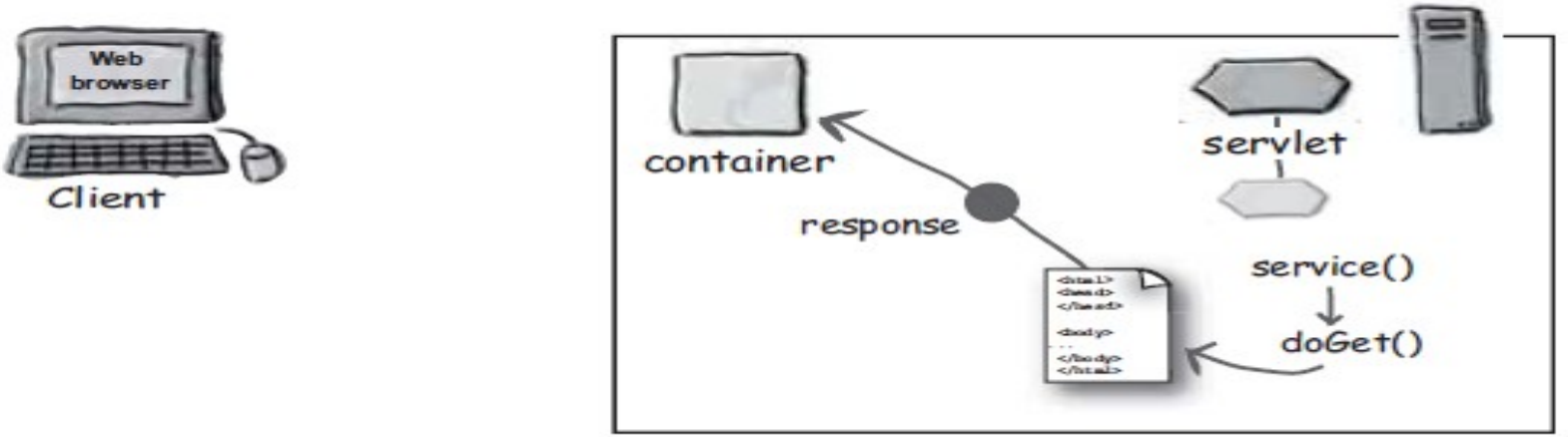
- Containers :-

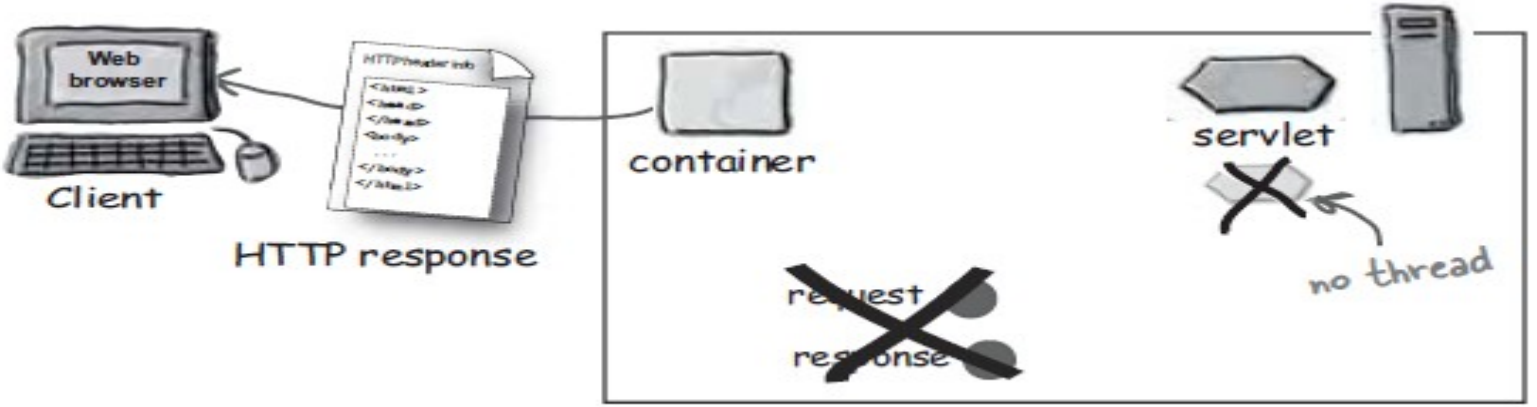
  Servlet containers contain servlets, filters, listeners, etc. and manages their state and lifecycle.


Browser
HTTP Request
HTTP Response
HTTP
Server
Servlet
Container
Servlet
Static
Content


HTTP request
GET
Web browser
Client
container
servlet
Web browser
Client
container
servlet
request
response
Web browser
Client
container
request
response
servlet
thread

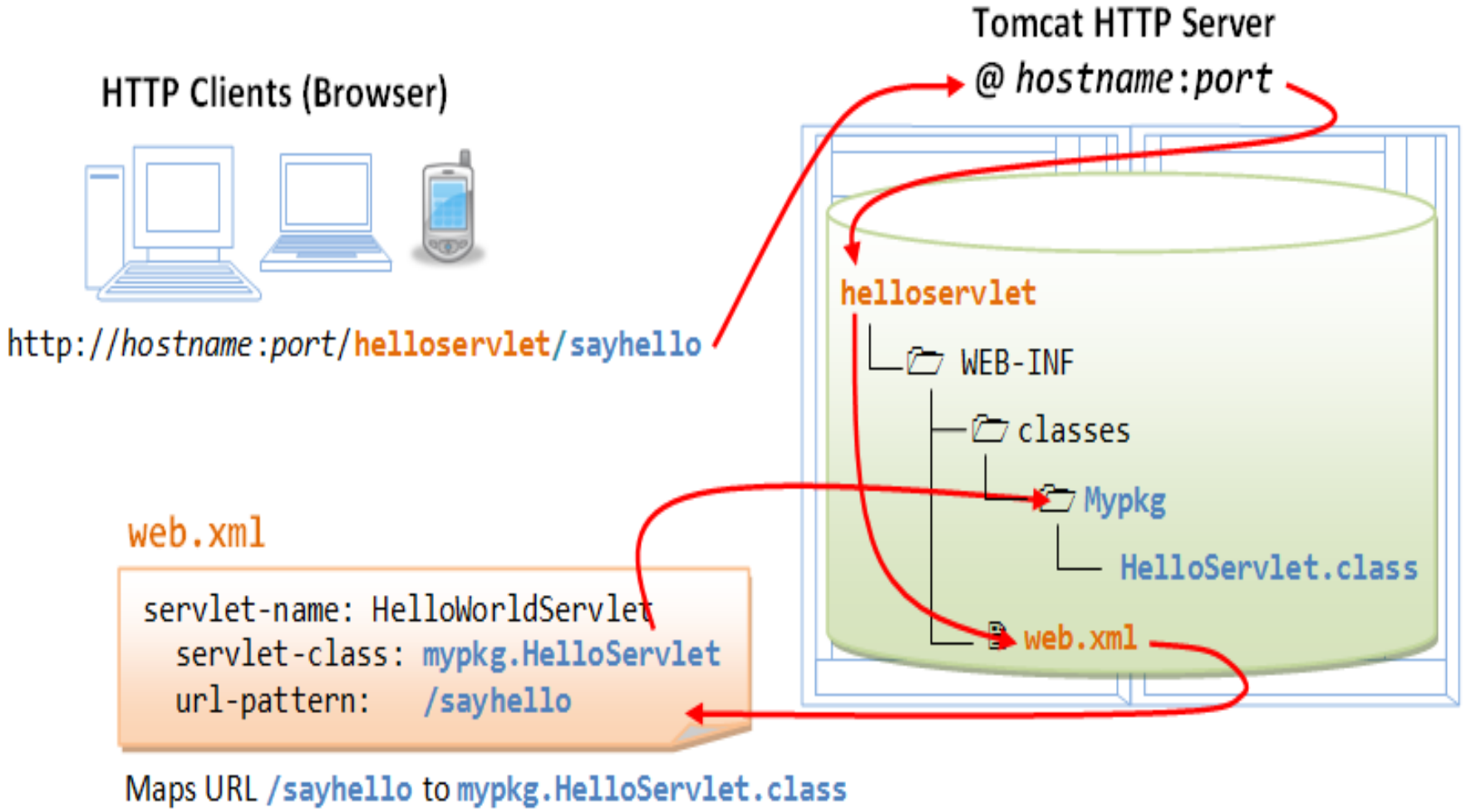

Web browser
Client
container
servlet
request
response
service()


Web browser
Client
container
servlet
response
service()
doGet()


Web browser
Client
HTTP response
container
servlet
no thread

- Web.xml :-

- Also called web Deployment descriptor
- Servlet mapping,
- Contains Welcome files, initialization parameters, define tag libraries,
- Session configuration,
- Cookie management,
- Security and Authentication


Tomcat HTTP Server
@ hostname:port
HTTP Clients (Browser)
http://hostname:port/helloservlet/sayhello
helloservlet
WEB-INF
classes
Mypkg
HelloServlet.class
web.xml
web.xml
servlet-name: HelloWorldServlet
servlet-class: mypkg.HelloServlet
url-pattern: /sayhello
Maps URL /sayhello to mypkg.HelloServlet.class

Advantages:-

Secure communication between Web server and servlet(XML),

JSP Support,

To add dynamic control from server side uses HTTP Protocol

Inexpensive,

Portable,

Cookie management,

Session tracking

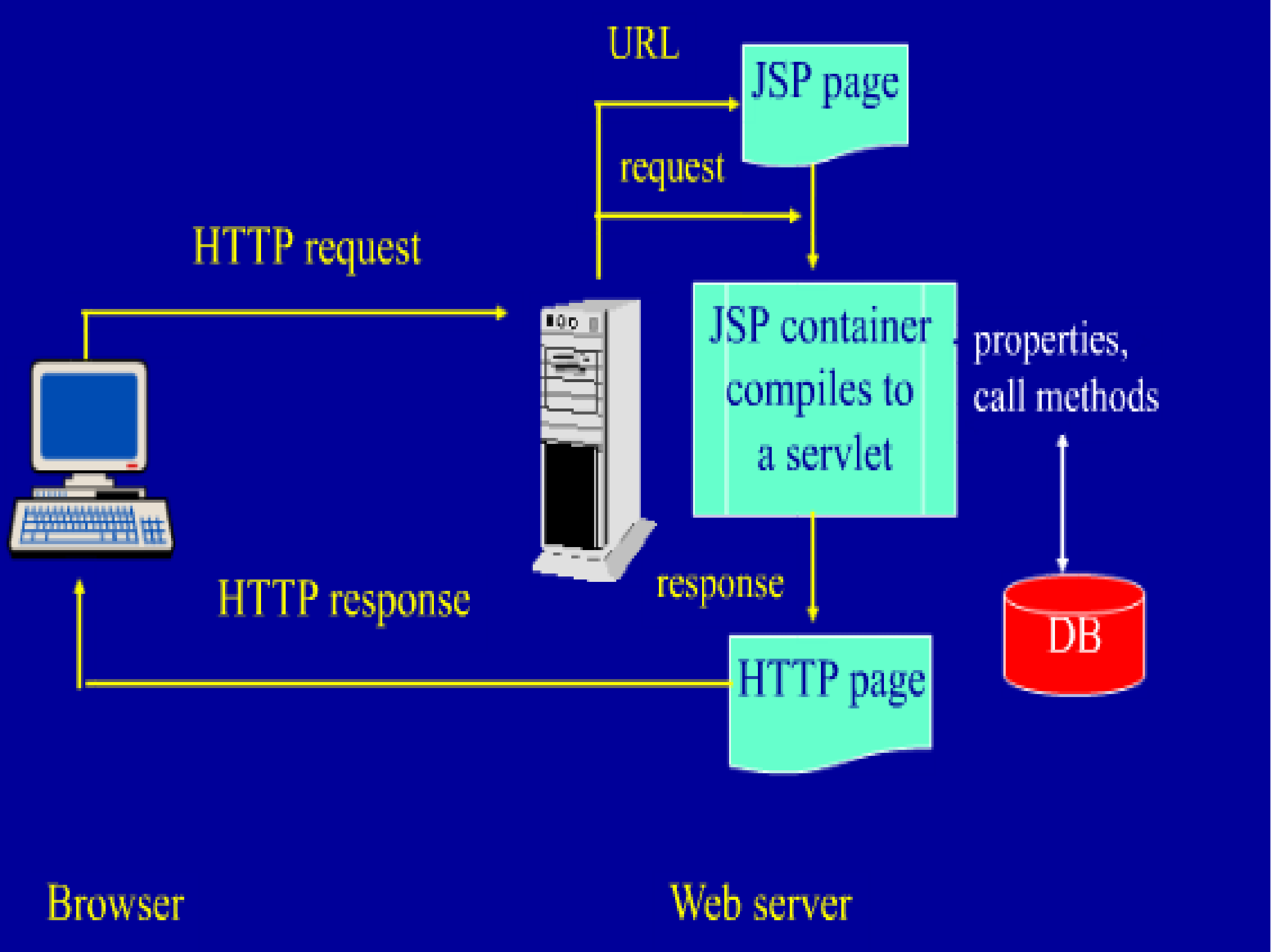
```java
package P1;

import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class MyServlet extends HttpServlet
{
    public void doGet(HttpServletRequest req, HttpServletResponse res)
    {
        res.setContentType("text/html");
        PrintWriter out = res.getWriter();
        out.println( "<HTML><HEAD><TITLE> Hellowwwwwwww" +
                    "</Title></HEAD>" +
                    "<Body> Hiiiiiiiiiiiiiiiii</BODY></HTML>" );
      out.close();
    }
   }
```

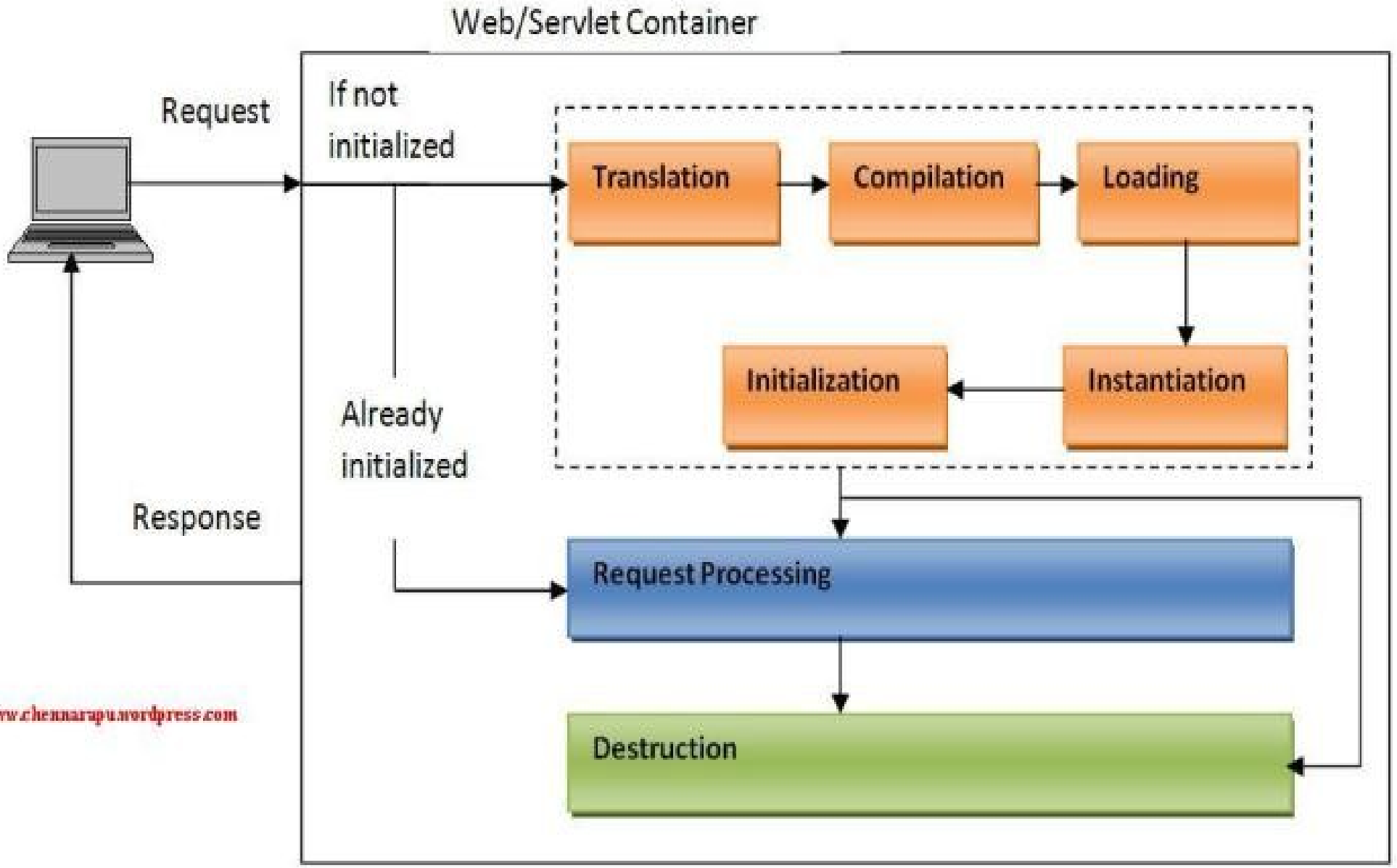
JSP

- HTML :- same and static content
- Technology that lets us mix static content with dynamically generated HTML pages.
- Fast way to create web pages that display dynamic content
- Doesn't give you anything that you can't do with servlets.
- Compact and easier to understand.
- Code enclosed in out.println() statements of servlet
- A template like structure in which logic is embedded.
-


URL
JSP page
request
HTTP request
JSP container compiles to a servlet
properties, call methods
response
HTTP response
DB
HTTP page
Browser
Web server

- Instantiation:
- When a web container receives a jsp request (may be first or subsequent), it checks for the jsp's servlet instance. If no servlet instance is available or if it is older than the jsp, then, the web container creates the servlet instance using following stages.
-
- Translation
- Compilation
- Loading
- Instantiation
- Initialization
-

JSP LifeCycle



JSP Expressions

Used to insert Java values directly into the output

Such type of expressions in <%=. . . %> tags will be evaluated and printed to HTML

```
<HTML>
    <HEAD>
        <TITLE>The Current Date and Time</TITLE>
    </HEAD>
    <BODY>
        tym is <= new java.util.Date() %>
    </BODY>
</HTML>
```

JSP Scriptlets

- Don't produce a value directly
- Just a java code in <% . . . %> tags in HTML, inserted into servlet exactly as written
- Contain any number of statements or method declarations

```
<%
    java.util.Date now = new java.util.Date();
%>
<HTML>
    <BODY>
        The time is now <%= now %>
    </BODY>
</HTML>
```

JSP Declarations

- To declare values and methods.
- Syntax :- <%! ….. %>

  i.e. <%! int i=0; %>

JSP Directives

Syntax :- <%@ directive attribute="value" %>

- 3 types :- page, include, taglib
- Page:- to add special feature (**import**, extends, info)
  <%@ page content type="text/html"%>
- Include:- to include the contents of any other file
  <%@ include file="index.html" %>
- Taglib :- to define a tag library that defines many tags.
  <%@ taglib uri="www.javatags.com/tags"
  prefix="mytag"%>

The

End