CDAC MUMBAI

ECGC Training : Day 2

Servlet

Dr Kiran Waghmare
&
Vipul Tembulwar

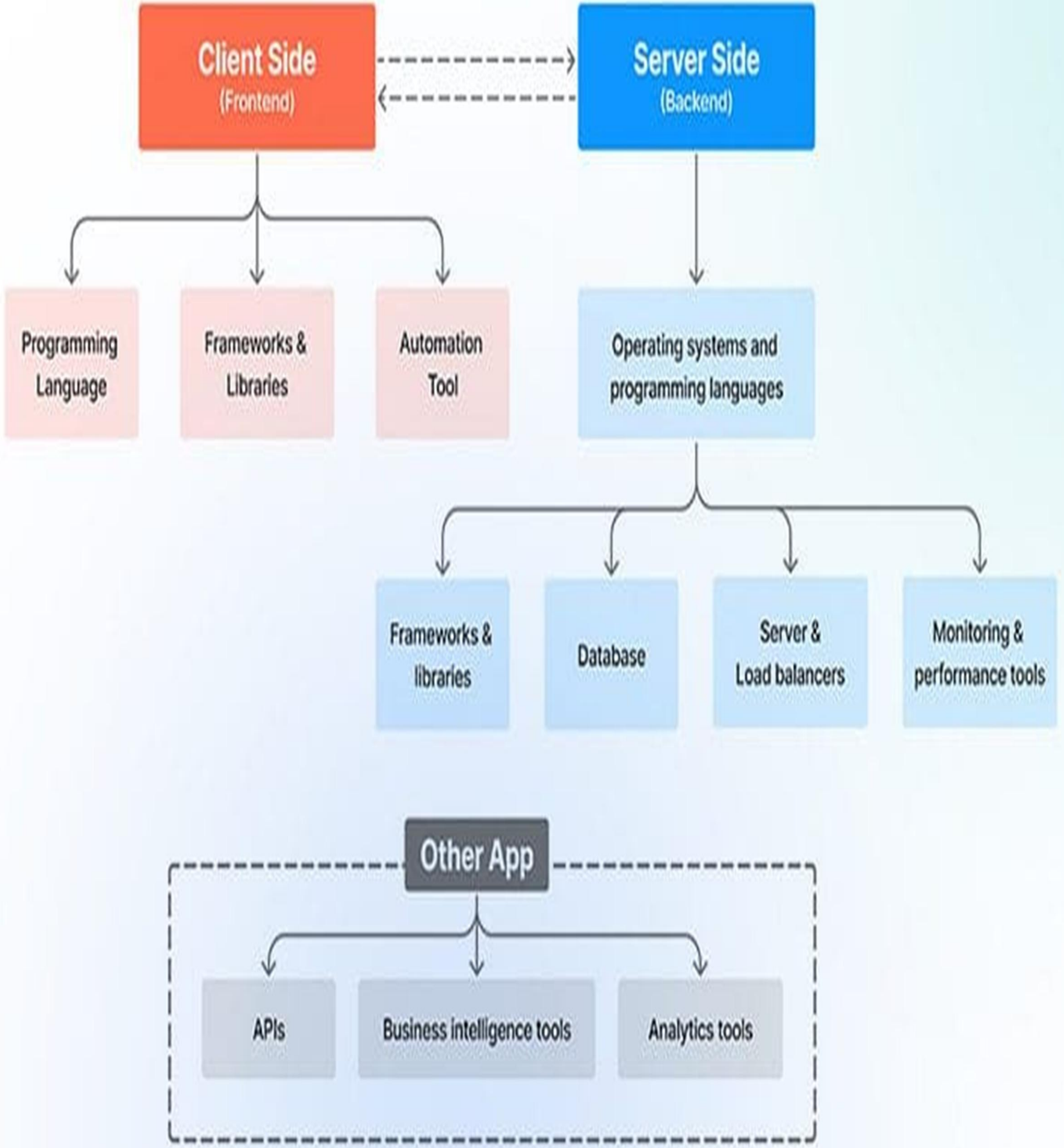
Start Slide ›

Agenda

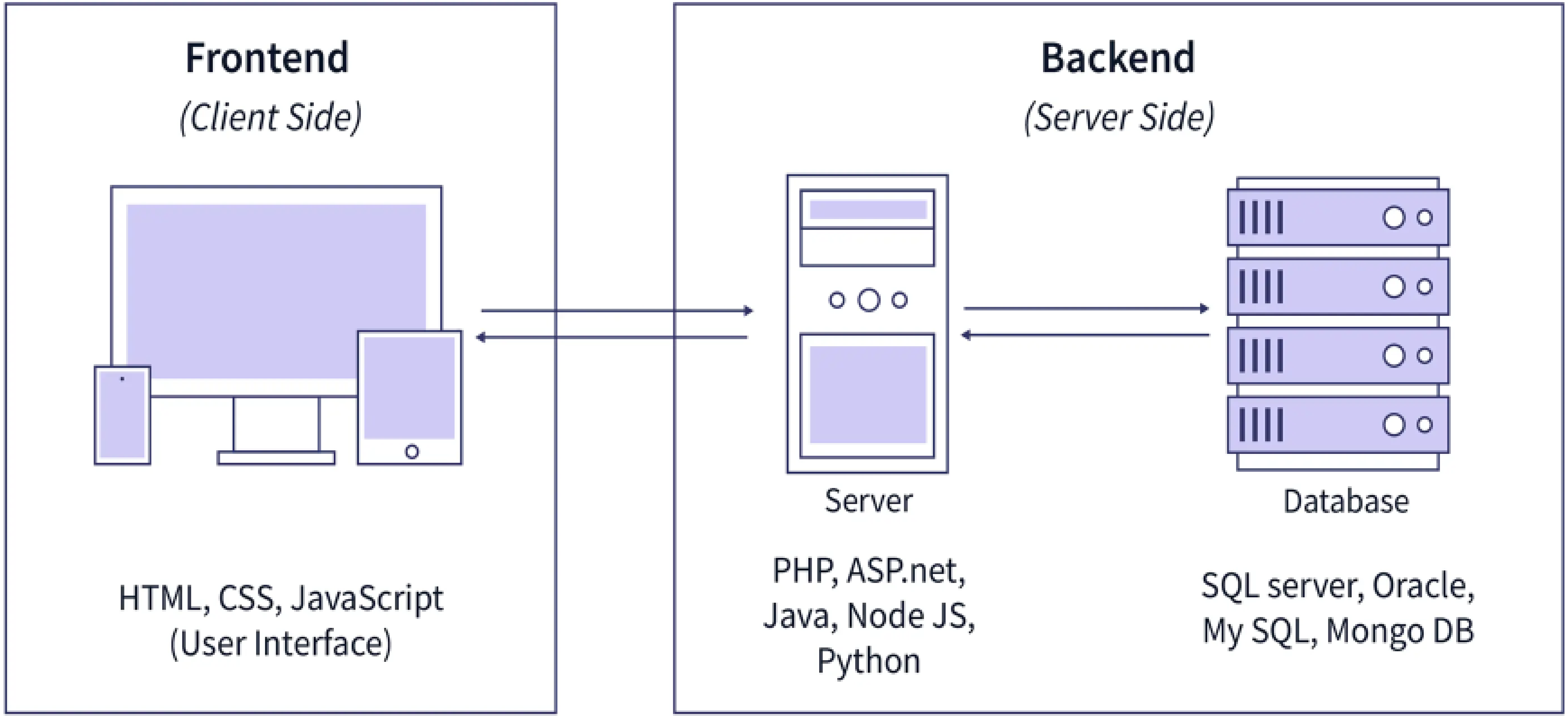
- Web Application Structure
- Servlet
- Servlet Life Cycle
- HTTP Methods in Servlets
- HTTPRequest and HTTPResponse
- Servlet Parameters
- Request Dispatcher
- Redirection
- Servlet Context & Servlet Config

Overviev

- A **web application** is built by combining several components that work together to provide dynamic, interactive user experiences. The structure of a web application typically includes:
  - **Frontend (Client-Side):**
    - This includes everything the user interacts with directly in the browser.
  - **Backend (Server-Side):**
    - This handles the business logic, processes user requests, and interacts with databases or external services.
  - **Database and External APIs:**
    - These provide storage for data and additional functionality via third-party services.



Web Application



Frontend

(Client Side)

HTML, CSS, JavaScript
(User Interface)

Backend

(Server Side)

Server

PHP, ASP.net,
Java, Node JS,
Python

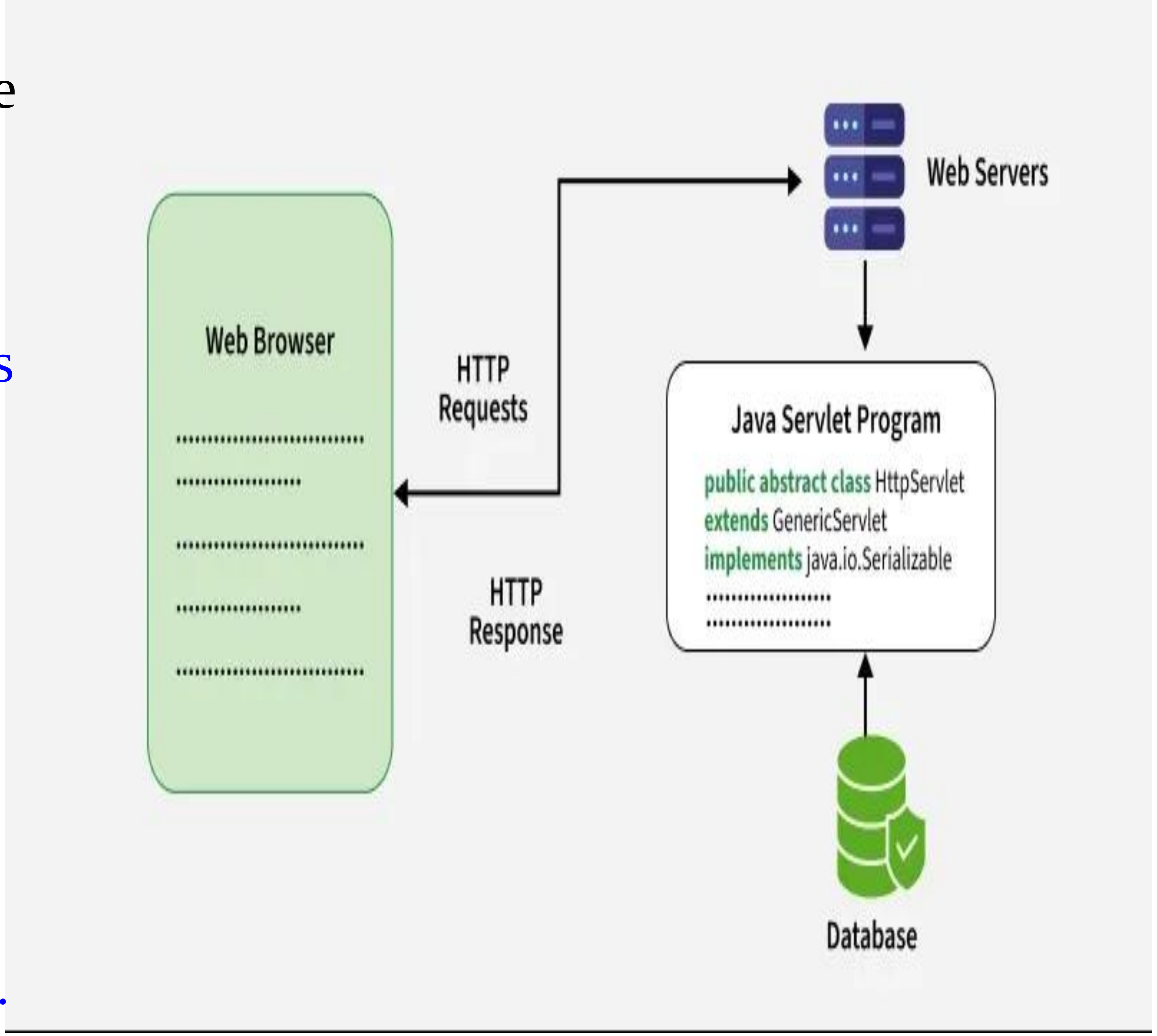
Database

SQL server, Oracle,
My SQL, Mongo DB

Java Servlet

- **Definition:**
  - Servlet is a server-side technology used to create dynamic web applications.
  - It runs on a web server and generates dynamic content based on incoming HTTP requests.
  - Servlets are Java classes that respond to requests from a client (usually a browser).

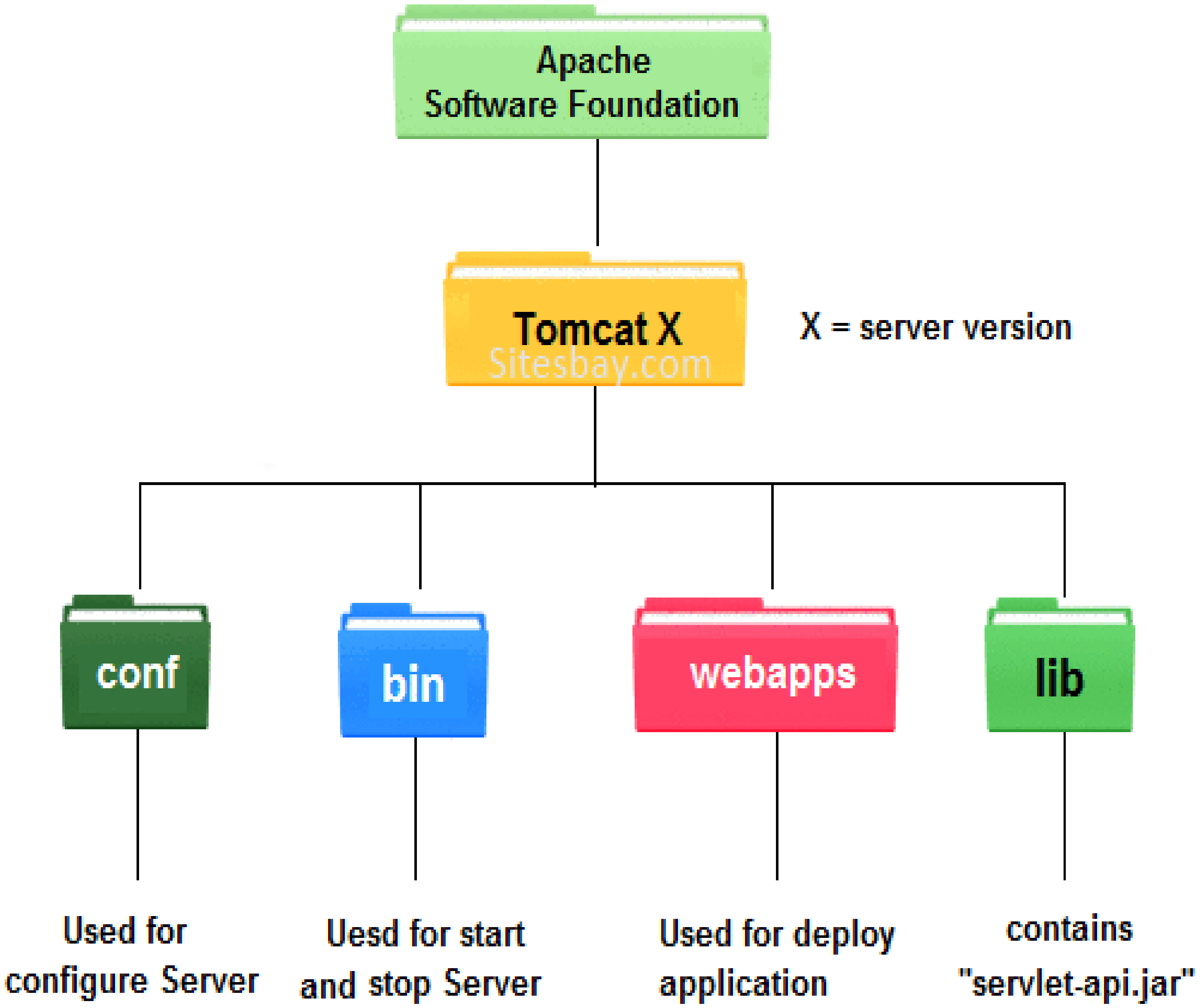
- **Key Points:**
  - Servlet Technology:
    - A robust and scalable solution using Java.
  - Java Servlet API:
    - Contains interfaces and classes for building web components, such as Servlet, GenericServlet, HttpServlet, ServletRequest, and ServletResponse.



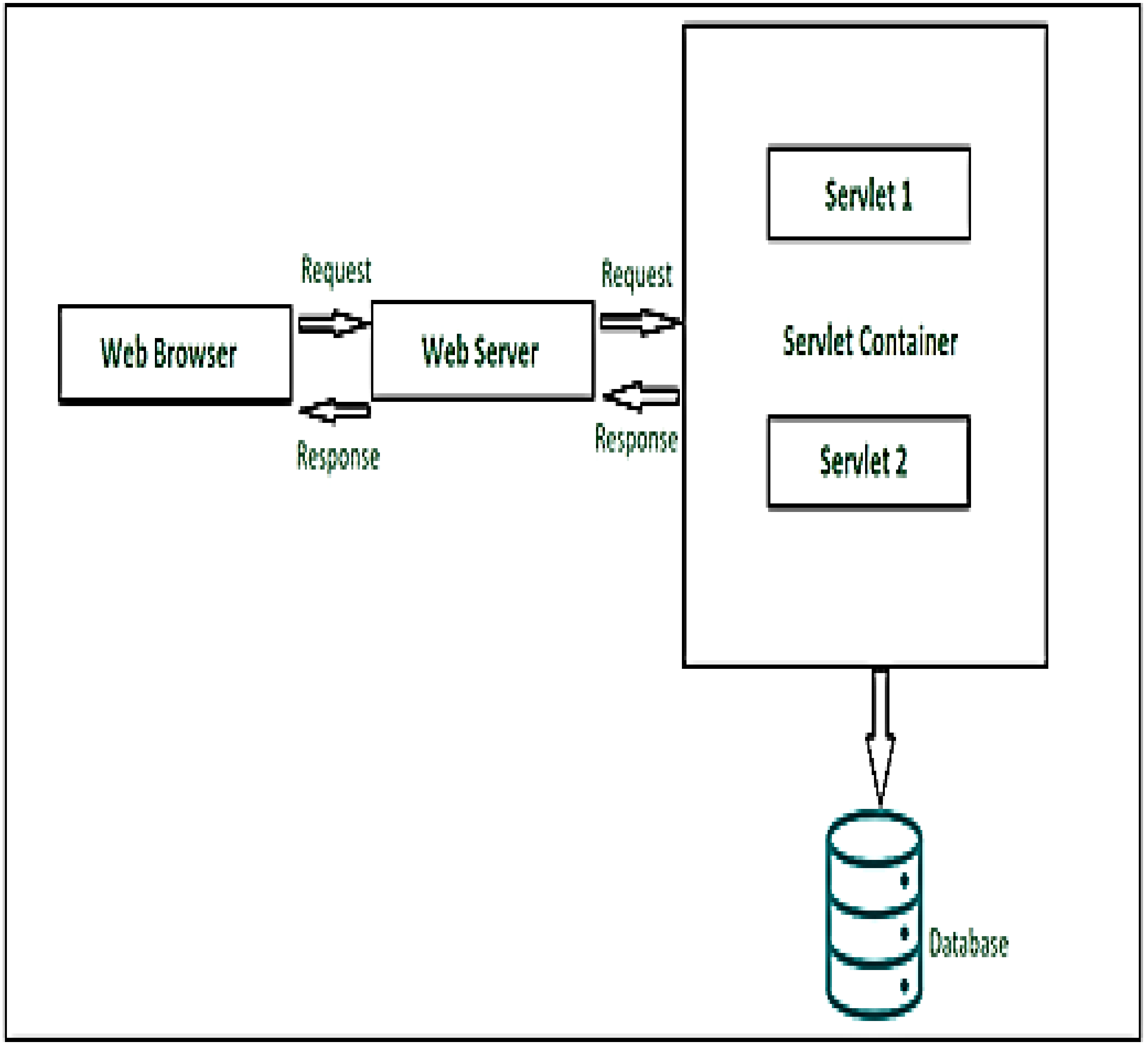
Appache Tomcat Server

- **Java Servlet Container and Web Server.**
- **Developed by the Apache Software Foundation.**
- **Implements Java Servlet and JavaServer Pages (JSP) specifications.**
- **Primarily used for running Java-based web applications.**

- **Key Features of Tomcat**
  - Servlet Container:
    - Handles HTTP requests and processes servlets.
  - Web Server:
    - Serves dynamic Java web applications.
  - Cross-Platform:
    - Runs on Linux, Windows, macOS, and other platforms supporting Java.Open-Source: Free to use under the Apache License.

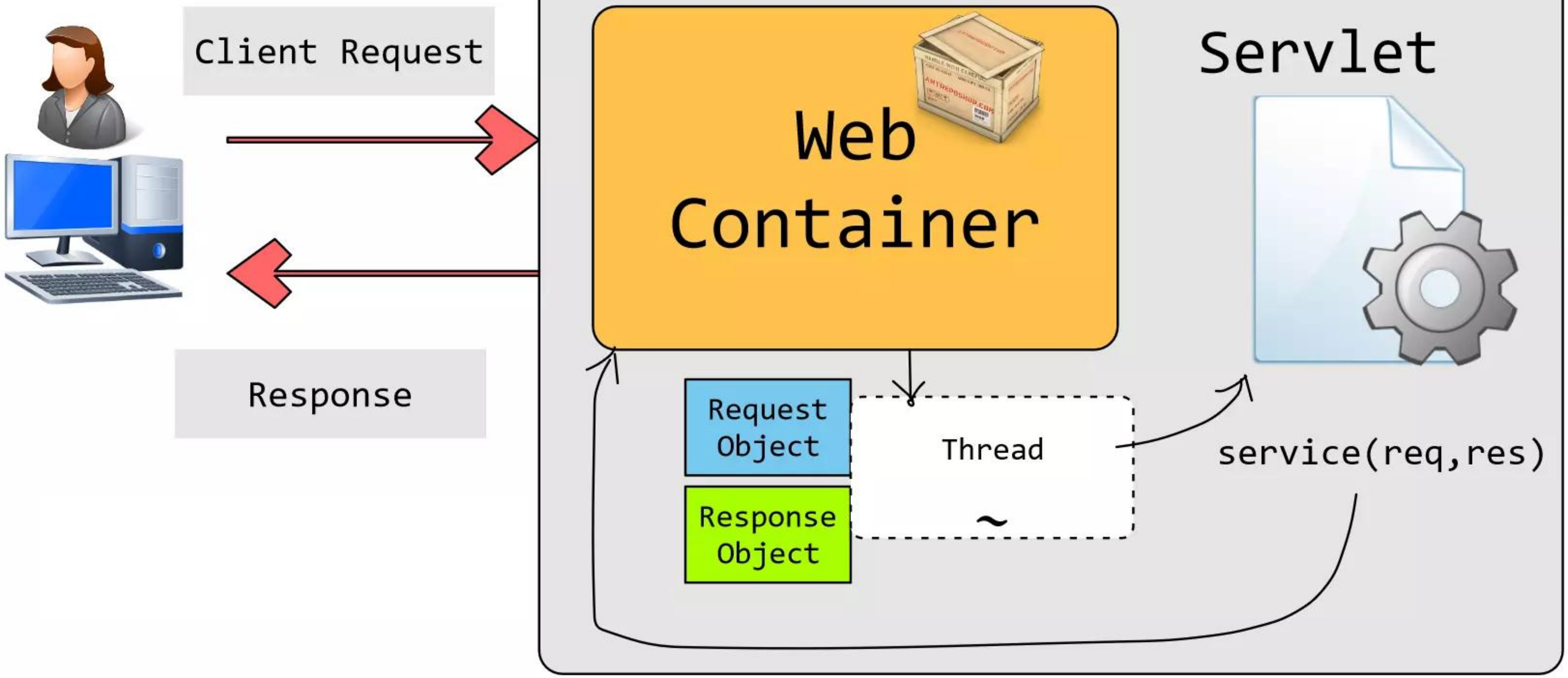

Apache
Software Foundation
Tomcat X
Sitesbay.com
X = server version
conf
bin
webapps
lib
Used for configure Server
Uesd for start and stop Server
Used for deploy application
contains "servlet-api.jar"

Types of Servlets



- **Servlet as a Web Component:**
  - A servlet is a web component deployed on a server that creates dynamic web pages.
- **Servlet API:**
  - Provides a set of classes and interfaces to handle HTTP requests and responses.
- **Servlet Classes:**
  - GenericServlet: A protocol-independent servlet.
- **HttpServlet:**
- **An HTTP-specific servlet used for handling HTTP requests.**

Web Server



Client Request

Response

Web Container

Servlet

Request Object

Response Object

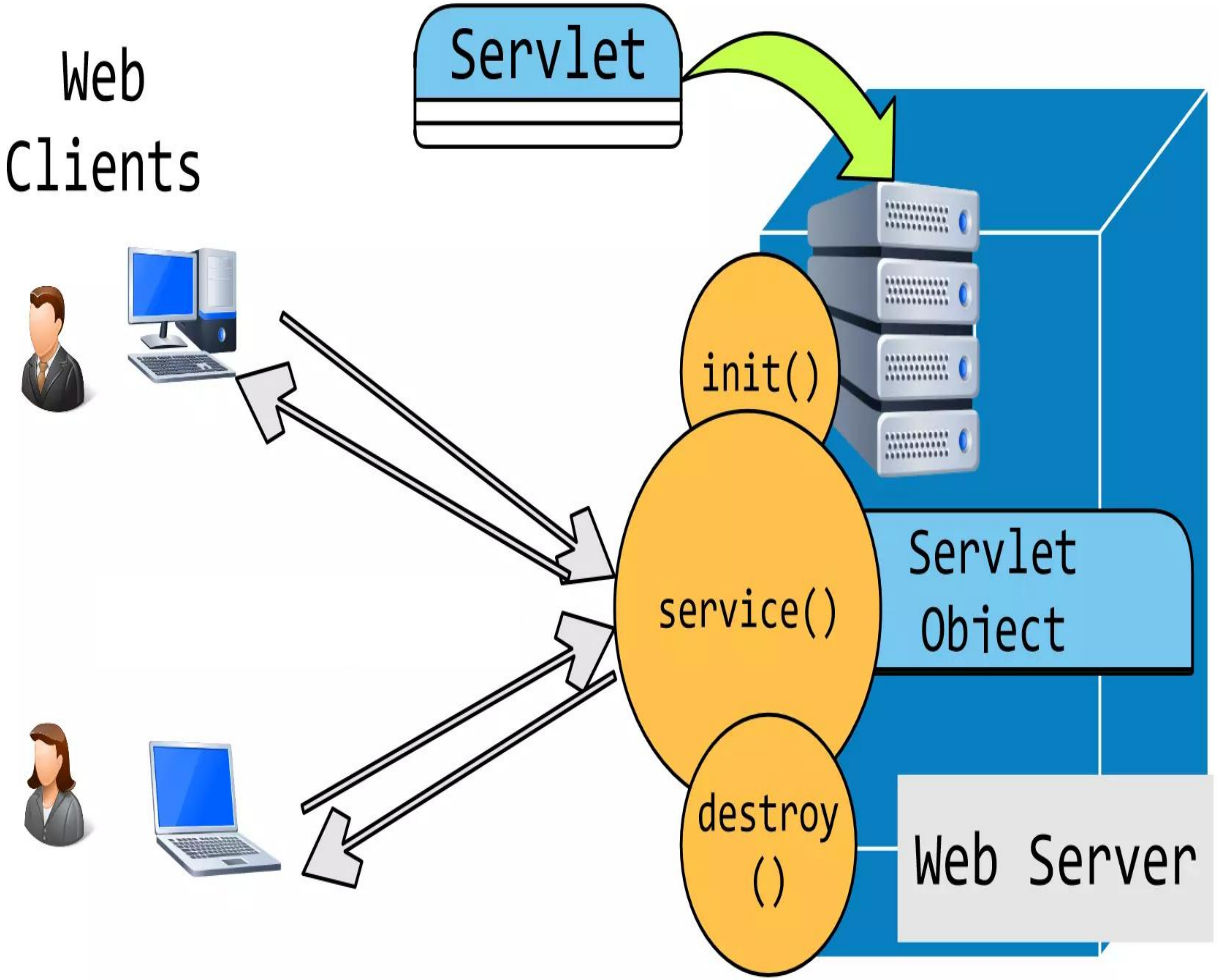
Thread

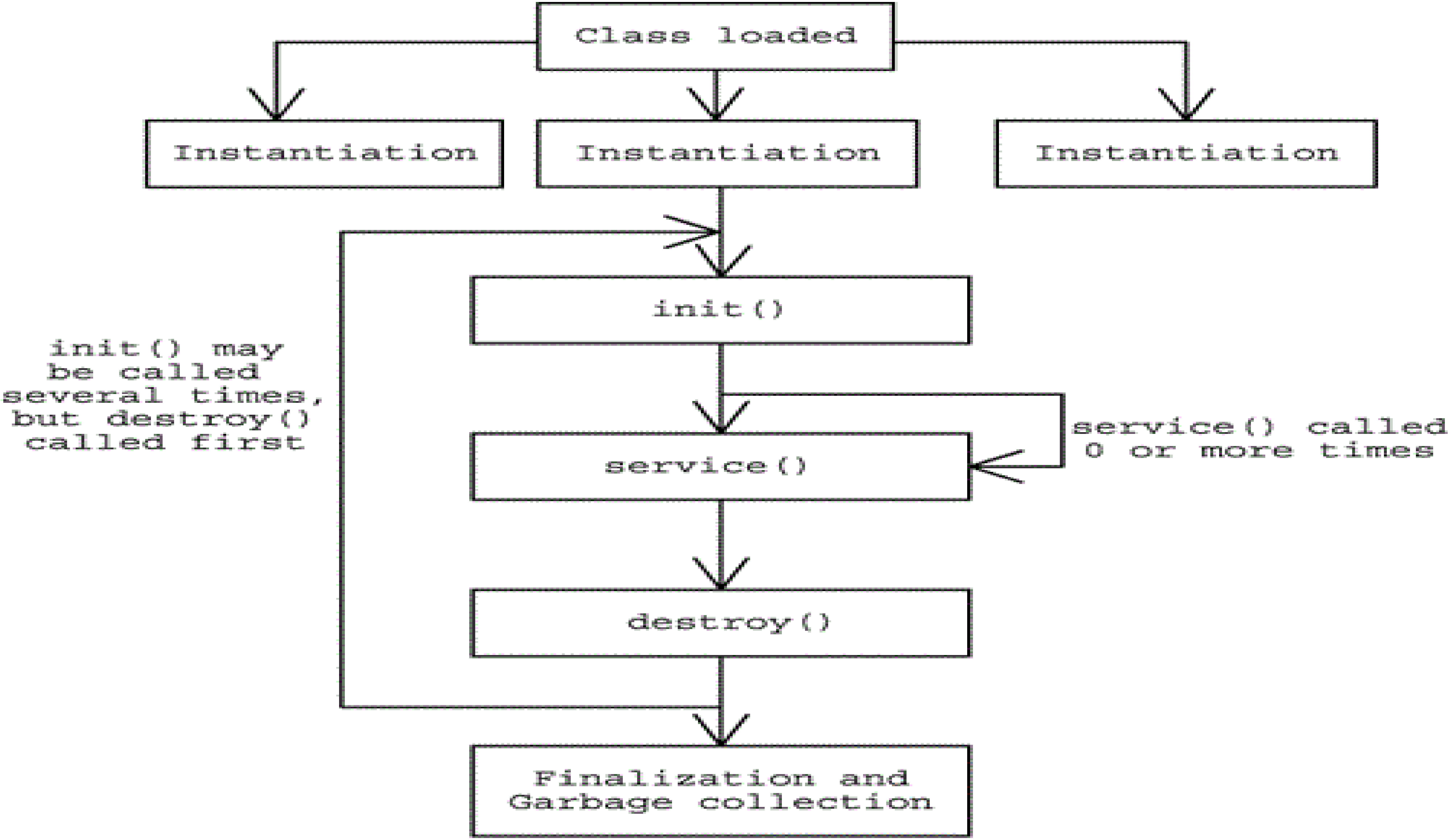
~

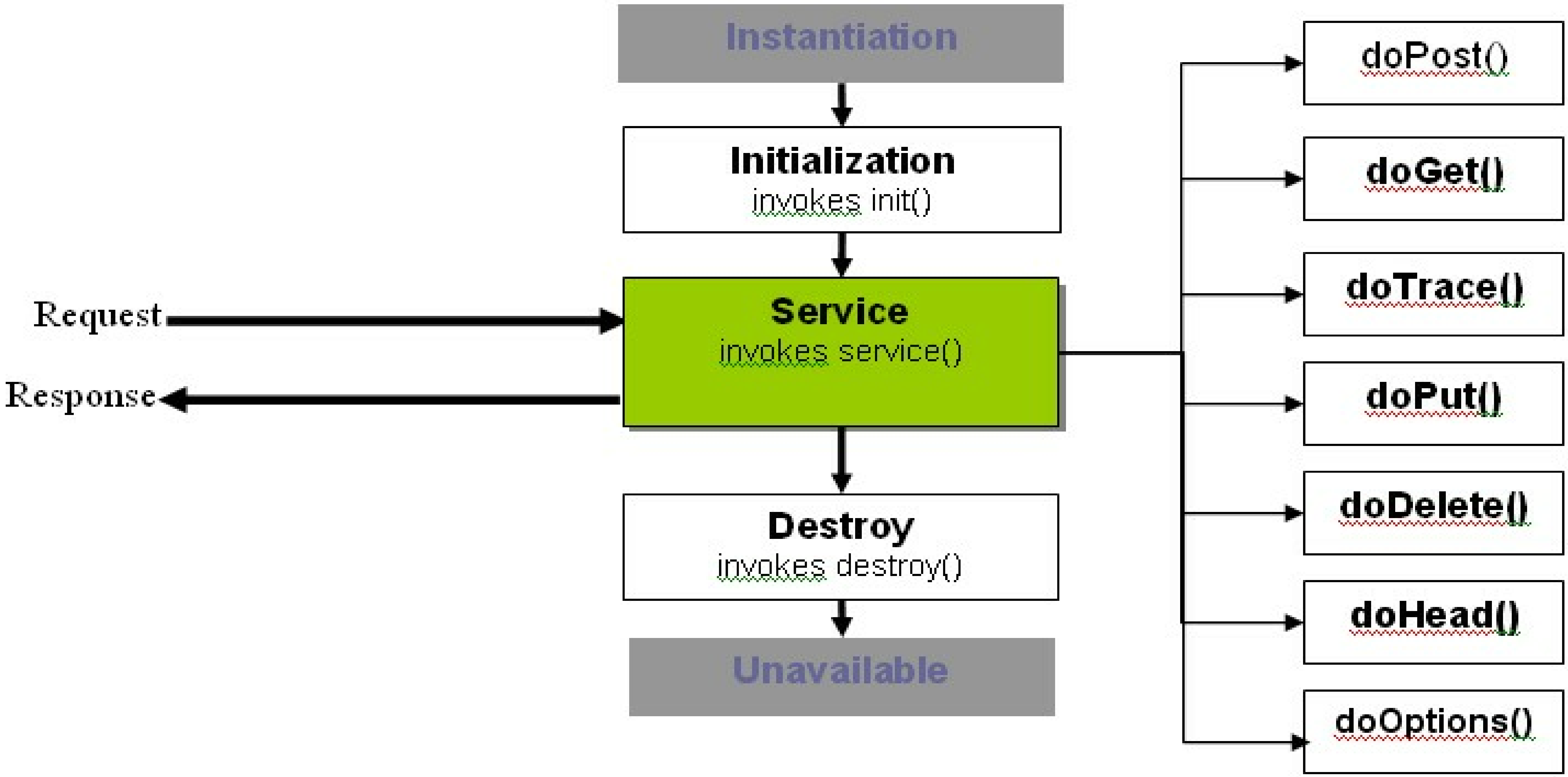
service(req,res)

Servlet Lifecycle

- **Initialization:**
  - The servlet is loaded into memory by the web container. The init() method is called once.
- **Request Handling:**
  - For each incoming request, the service() method is called, processing the request and generating a response.
- **Destruction:**
  - When the servlet is no longer needed, the destroy() method is called to clean up resources.

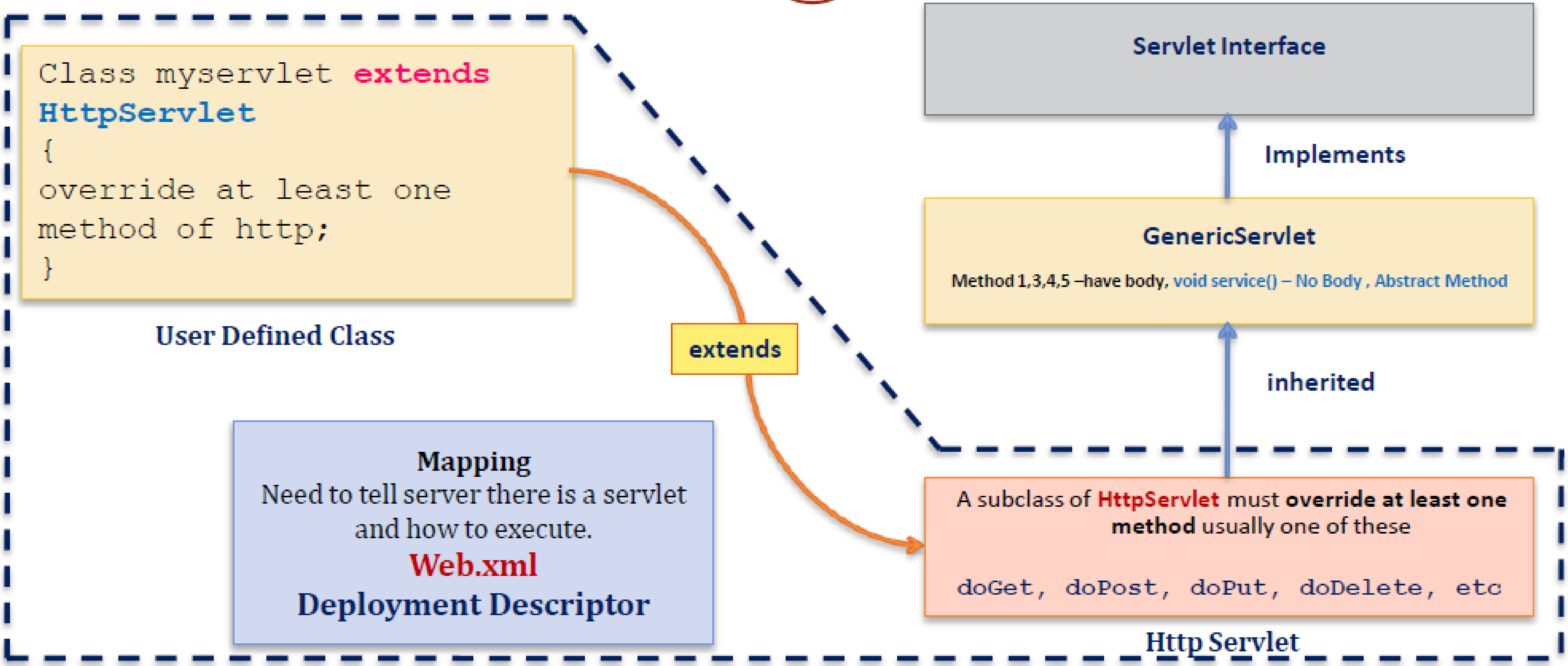



Class loaded
Instantiation
Instantiation
Instantiation
init()
init() may be called several times, but destroy() called first
service()
service() called 0 or more times
destroy()
Finalization and Garbage collection


Instantiation
Initialization
invokes init()
Request
Response
Service
invokes service()
Destroy
invokes destroy()
Unavailable
doPost()
doGet()
doTrace()
doPut()
doDelete()
doHead()
doOptions()

Create Servlet extends HttpServlet class

13



```
Class myservlet extends
HttpServlet
{
override at least one
method of http;
}
```

User Defined Class

Method 1,3,4,5 –have body, void service() – No Body , Abstract Method

A subclass of **HttpServlet** must **override at least one method** usually one of these

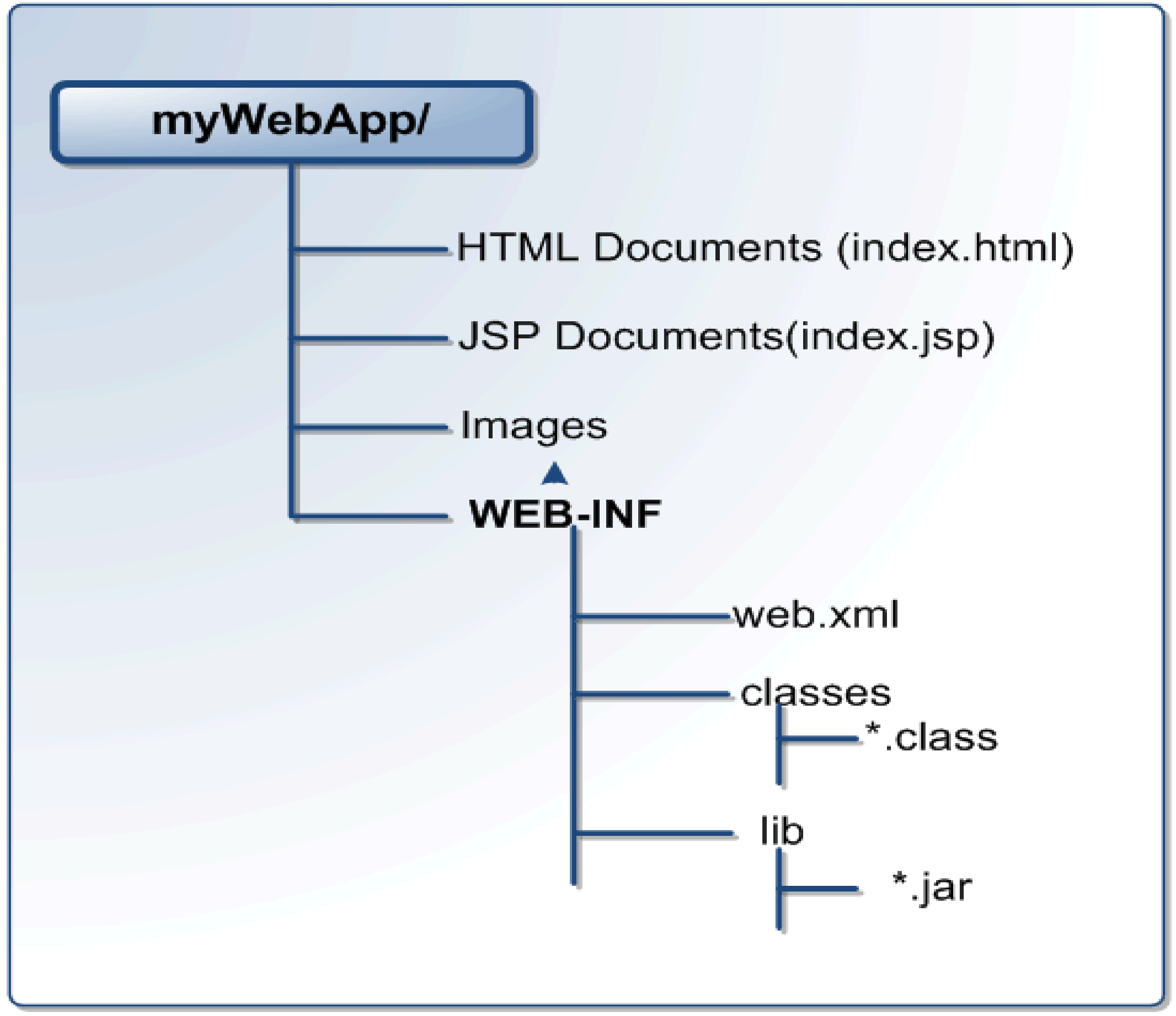
`doGet, doPost, doPut, doDelete, etc`

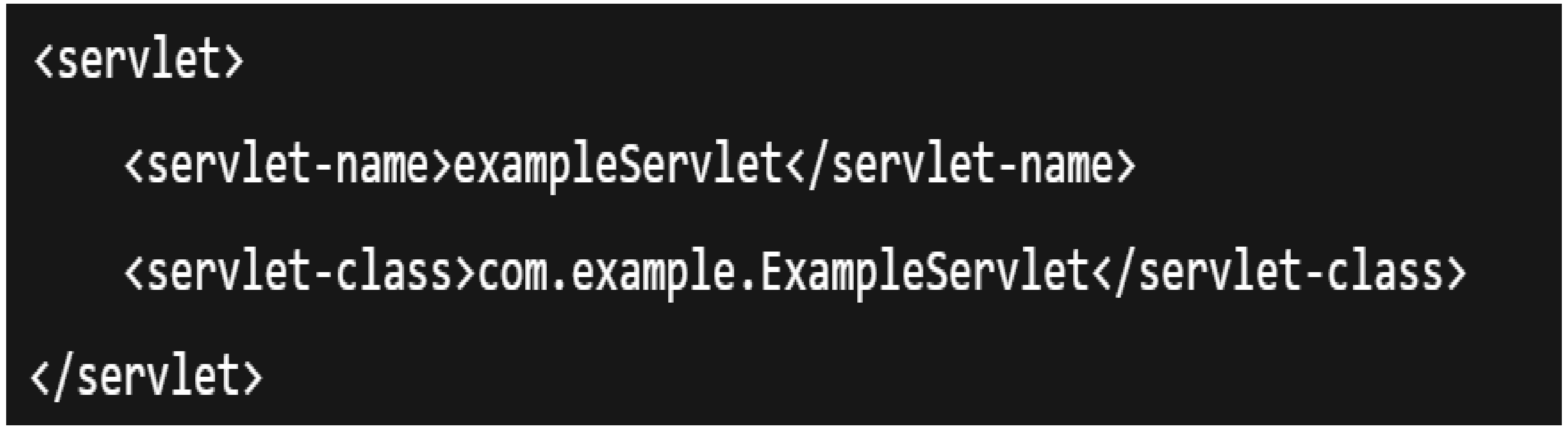
Http Servlet

Mapping
Need to tell server there is a servlet and how to execute.
Web.xml
Deployment Descriptor

Web.xml

- web.xml is a Deployment Descriptor used to configure Java Web Applications in Servlet containers (e.g., Tomcat, Jetty).
- It is an XML file that defines the settings and configurations for servlets, filters, listeners, and other components in a web application.

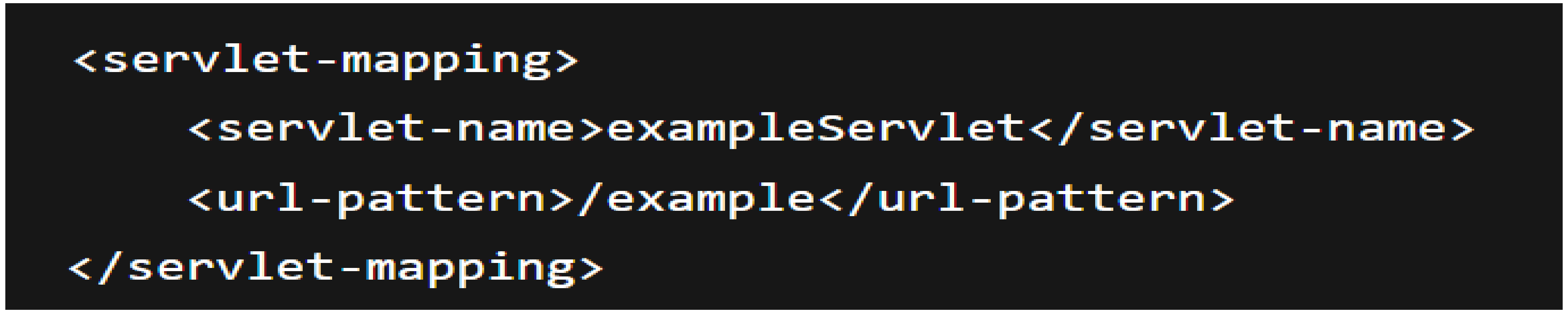




```
<servlet>
    <servlet-name>exampleServlet</servlet-name>
    <servlet-class>com.example.ExampleServlet</servlet-class>
</servlet>
```

Servlet Configuration

Servlet Mapping



```
<servlet-mapping>
    <servlet-name>exampleServlet</servlet-name>
    <url-pattern>/example</url-pattern>
</servlet-mapping>
```

Benefits of web.xml

Centralized Configuration:

- All servlet, filter, and listener configurations are stored in one location.

Easy Maintenance:

- Any changes to servlets, filters, security, or error handling are quickly made in web.xml.

Compatibility:

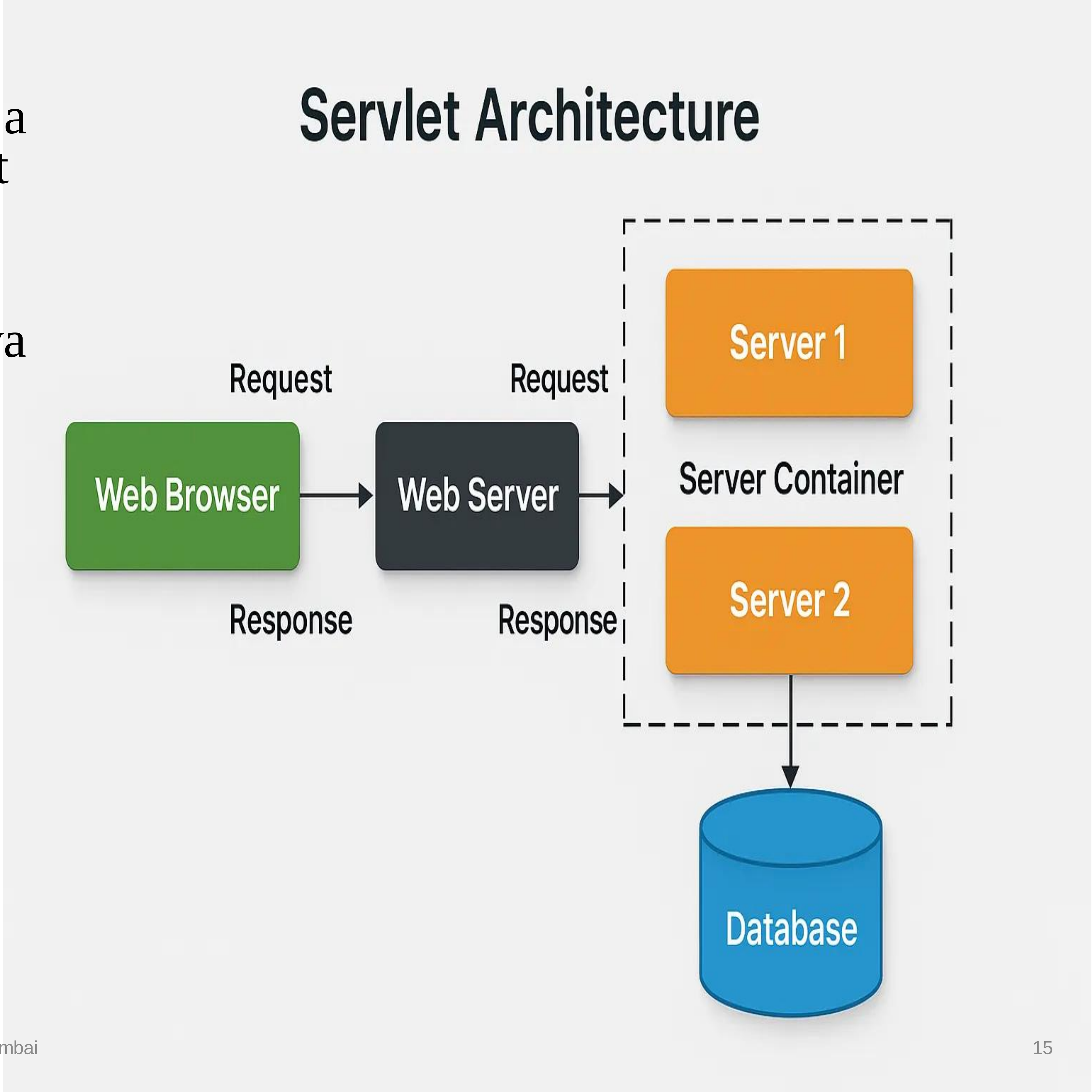
- Ensures compatibility with various servlet containers and versions of Java EE.

Legacy Support:

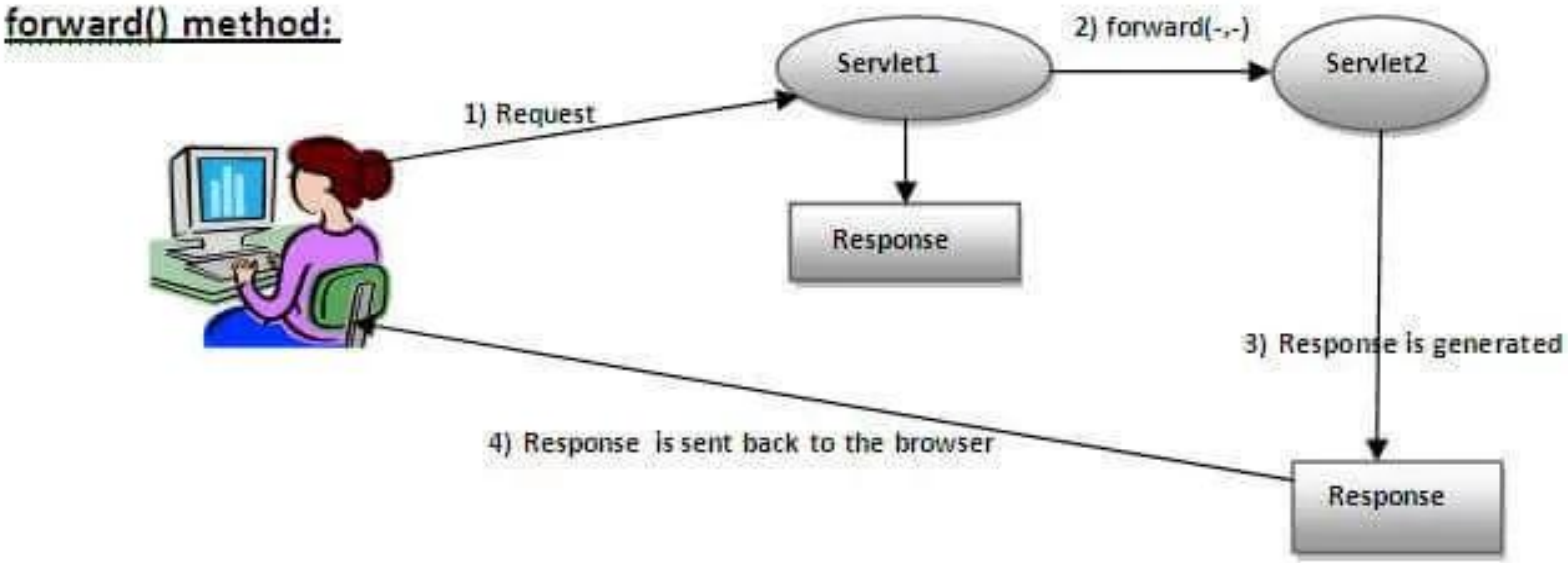
- Although Servlet 3.0 introduced annotations, web.xml remains essential for certain configurations and for backward compatibility.

Web Application and Servlet Collaboration

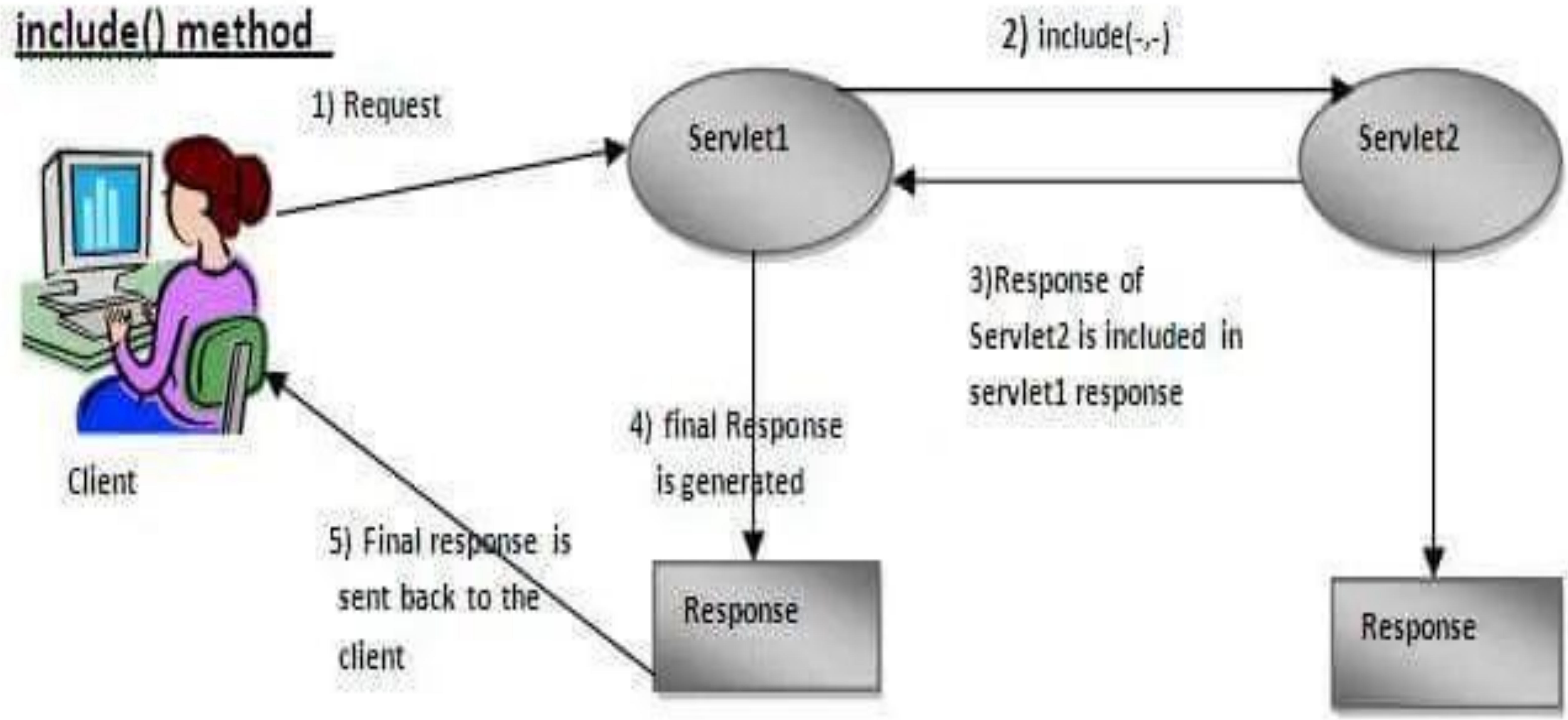
- **Web Application:**
  - A web application is an application that runs on a web server and can be accessed over the internet using HTTP or HTTPS.
  - It consists of components like Servlets, JSP (Java Server Pages), Filters, HTML, CSS, and JavaScript.
- **Servlet Collaboration**:
  - RequestDispatcher:
    - For forwarding requests to another servlet or JSP.
  - sendRedirect():
    - Redirects the client to a different URL.
-



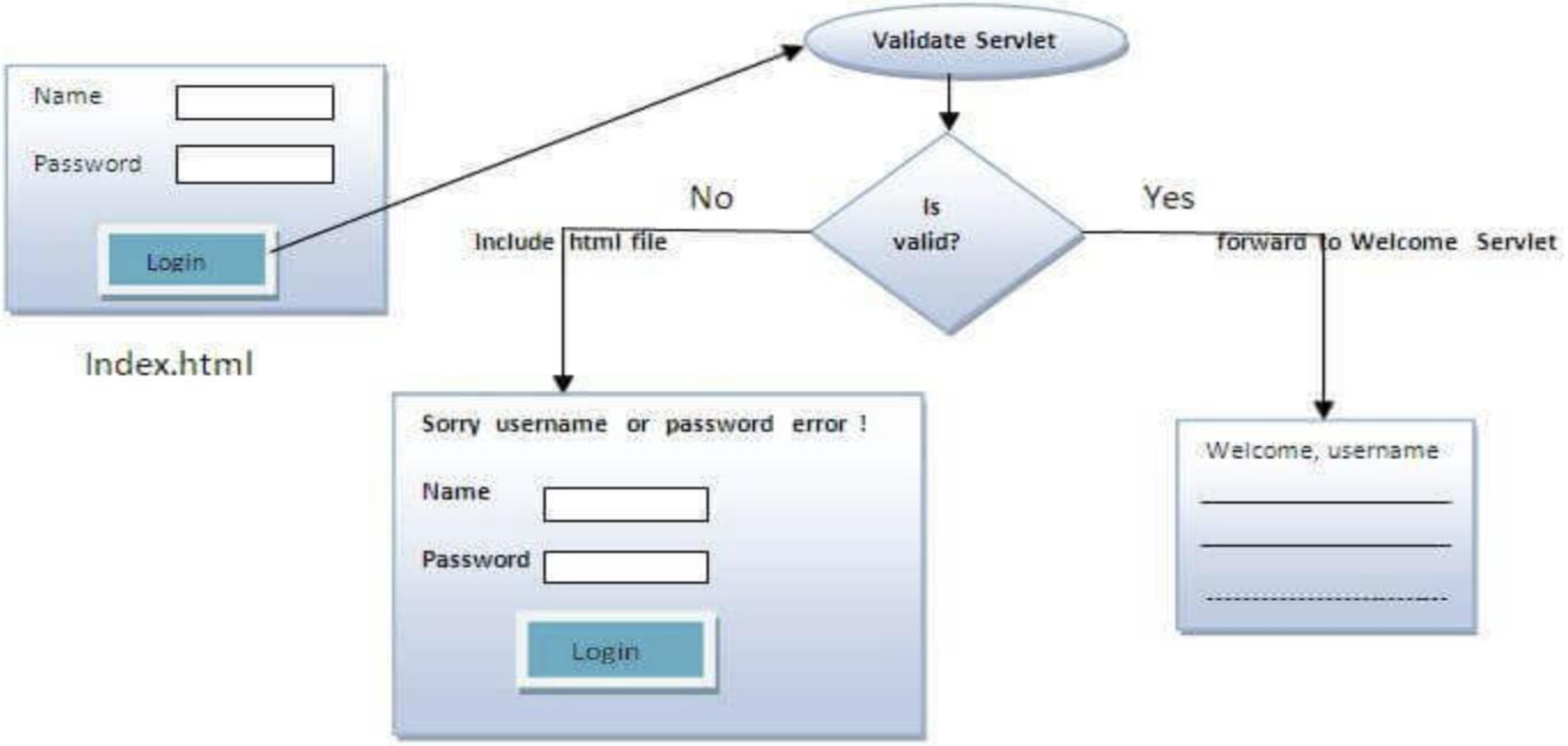
RequestDispachter



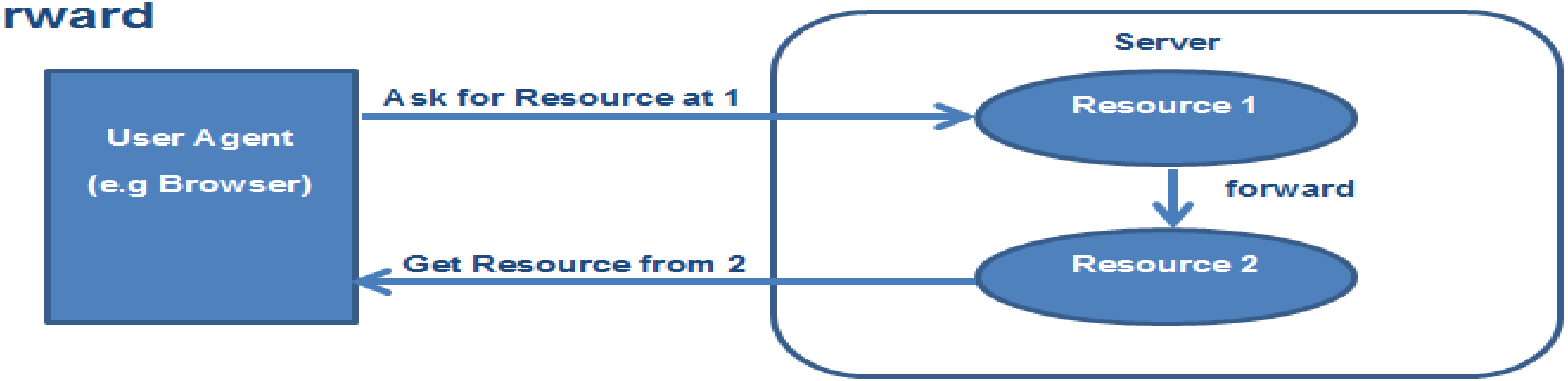
RequestDispachter


include() method
1) Request
2) include(-,-)
Servlet1
Servlet2
3)Response of Servlet2 is included in servlet1 response
4) final Response is generated
Client
5) Final response is sent back to the client
Response
Response

Example of include() & forward()



Forward



Server

User Agent

(e.g Browser)

Ask for Resource at 1

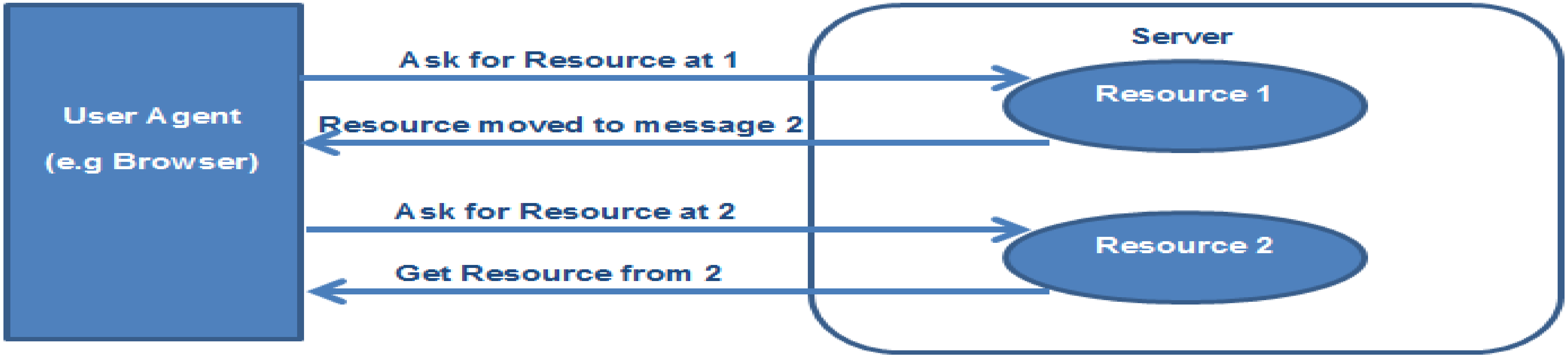
Resource 1

forward

Resource 2

Get Resource from 2

SendRedirect



Server

User Agent

(e.g Browser)

Ask for Resource at 1

Resource 1

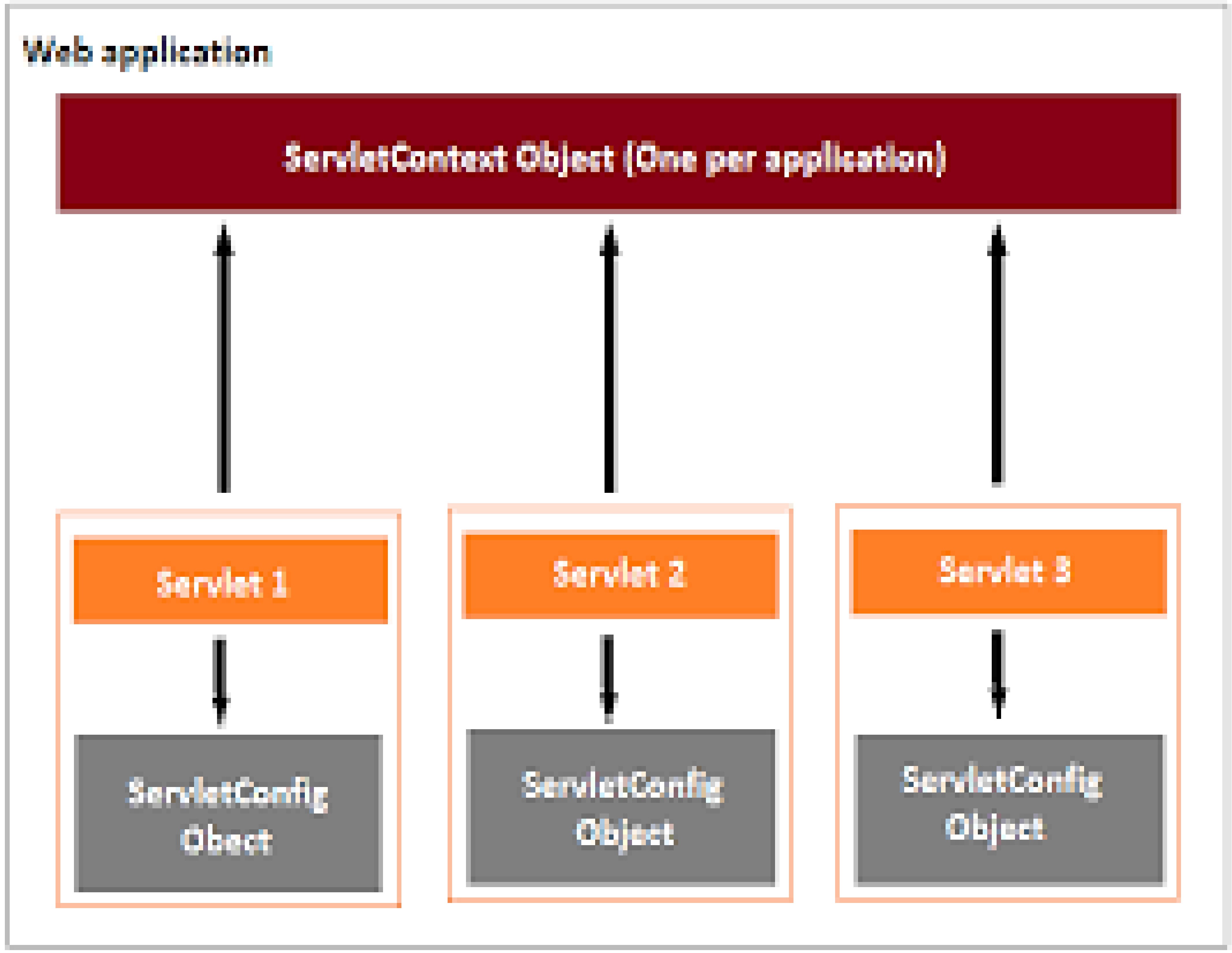
Resource moved to message 2

Ask for Resource at 2

Resource 2

Get Resource from 2

Servlet Config and Context



ServletConfig Interface:

Provides configuration information for the servlet (e.g., initialization parameters).

ServletContext Interface:

Provides context for the entire web application. It can be used to share data between servlets in the same application.

Thank You

FOR ALL YOUR ATTENTION