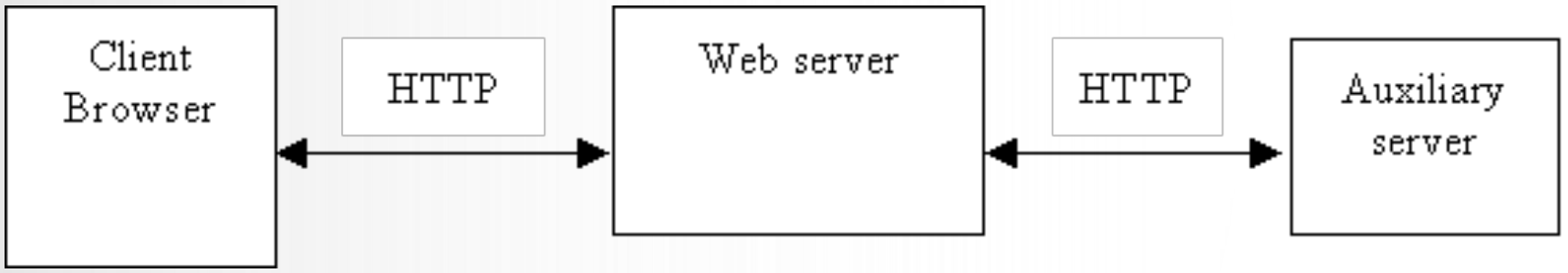
Java web programming

Java technology

- Concept of a *Java Virtual Machine (JVM)*
- Portability
- Three kinds of Java program
  - Applications
  - Applets
  - Servlets

Auxiliary server



Architecture of a Java web application

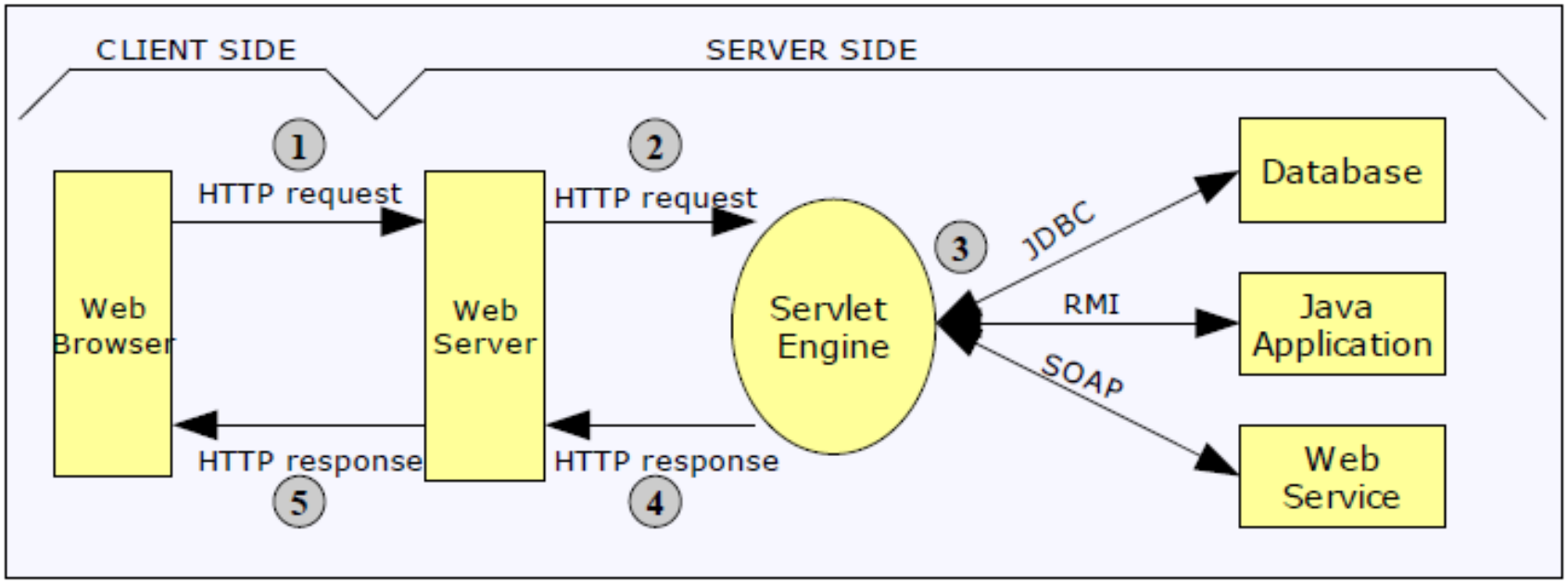


Figure 2.1 - The servlet's role. Adapted from Hall & Brown(2004).

Servlet

- Mini program created when the servlet container first receives a request that maps onto it
- A servlet **services** a request via a thread
- Servlet object continues to exist until container closes down

Typical functions of a servlet

1. Validate data input to it via HTTP request parameters
2. Perform some business function using that data
3. Decide what view the user should next see
4. Generate that view as an HTTP response

Servlet container

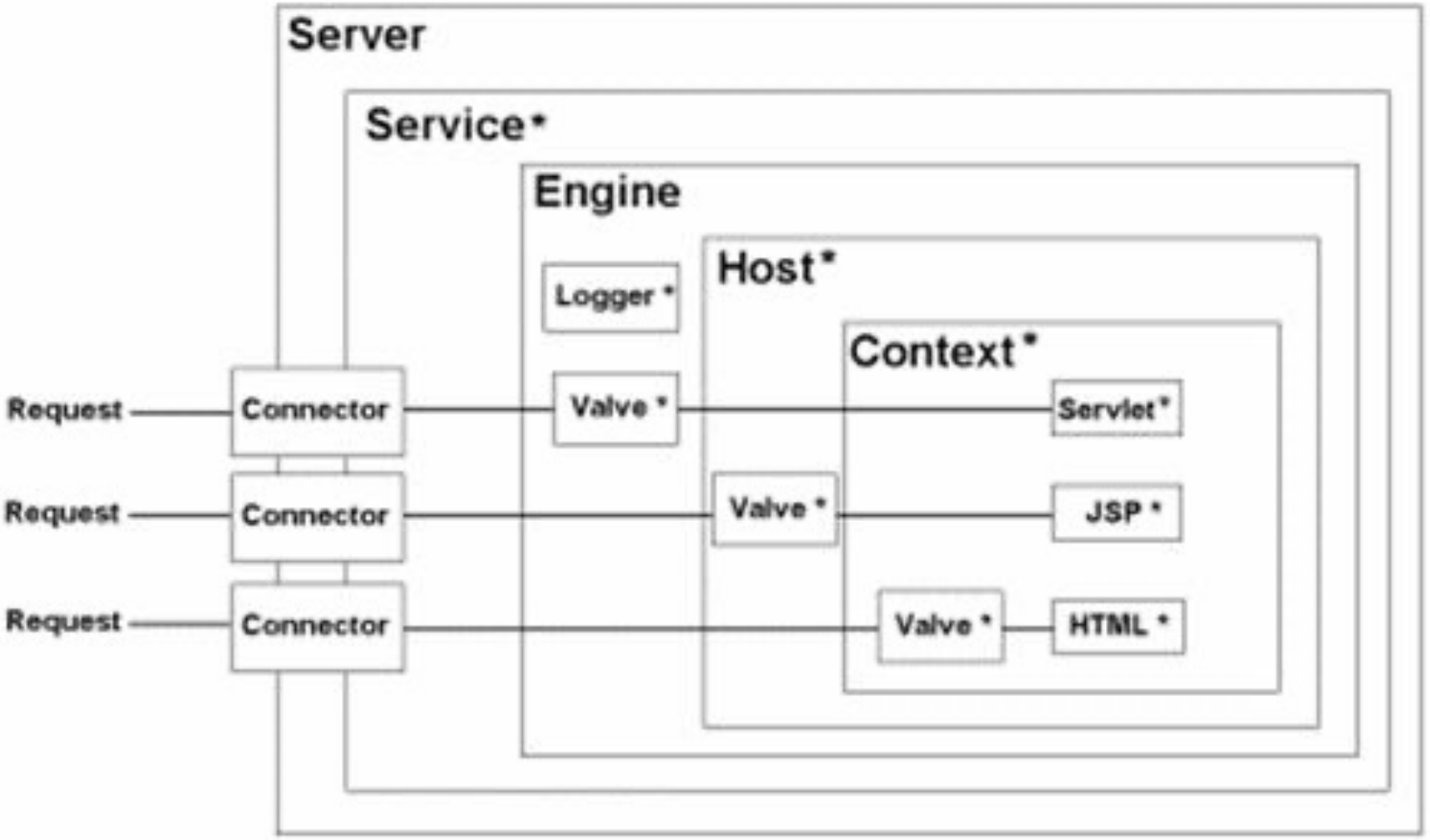
- Program that implements the Java servlet specification (and others)
- Part of the Java Enterprise Edition (Java EE)
- Reference implementation used to be **Tomcat** (an Apache project: http://tomcat.apache.org/)
- Full Java EE reference implementation now is **Glassfish** (http://glassfish.java.net/)
- Other implementations are available:
  - JBoss/WildFly

Functions of servlet container

1. Listen for HTTP requests
2. Decode HTTP request and decide which application and servlet it is intended for
3. Call servlet, passing request and response objects
4. Encode HTTP response and send it

Mapping URLs to servlets

- Consider the URL:
  - www.myserver.com/myapp/myservlet
- Container must break this down
  - www.myserver.com: virtual host
  - /myapp: context or web application
  - /myservlet: address within web application


Server
Service*
Engine
Logger *
Host*
Context*
Request
Connector
Valve *
Servlet*
Request
Connector
Valve *
JSP *
Request
Connector
Valve *
HTML *

Web applications

- A container may run several (independent) web applications (webapps)
- Each must have a WEB-INF directory:
  - web.xml configuration file
  - classes directory
  - lib directory

Brief history of JWP

- Servlets
- JavaServer Pages (JSP)
- Various MVC technologies (including Apache Struts)
- JavaServer Faces (JSF)

Important classes and interfaces 1

- All servlets must implement the *Servlet* interface
- Class *HttpServlet*
  - *init/destroy*
  - *doGet/doPut*
  - Your servlet will derive from this

Important classes and interfaces 2

- 2 parameters to a request handling method
- Class *HttpServletRequest*
  - String param = request.getParameter(name);
- Class *HttpServletResponse*
  - PrintWriter out = response.getWriter();
- Class *HttpSession*
  - Holds data common to related requests

JavaServer Pages (JSP)

- Distinction:
  - servlets: HTML embedded in program
  - JSP: program embedded in HTML
- JSP useful where majority of effort is page design
- Translated automatically into a servlet
  - Retranslated if changed (no need to restart server)
- Can be placed anywhere in a web application
  - but not visible to client if in the WEB-INF directory

JSP elements

- Scriptlets
- Actions
- Directives
- Standard tags
- Custom tags
- Expression language

Scriptlets

- Any Java code between <% … %>
- Expressions
  - <%= name %>
- Declarations
  - <%! String name %>
- DEPRECATED
  - Do not use - not XML
  - Much easier to use JSTL

Actions

- Including other files
  - <jsp:include page="path"/>
  - Request time inclusion
- Accessing beans
  - <jsp:usebean id="beanName" class="package.class" scope="session"/>
  - <jsp:getproperty name="beanName" property="propertyName"/>
  - <jsp:setproperty name="beanName" property="propertyName" value="newValue"/>

Directives

- Page directive
  - <%@page import="package.class"%>
  - <%@page contentType="text/html"%>
  - <%@page errorPage="URL"%>
- Include directive
  - <%@include file="filename"%>
  - Translation time inclusion

Standard tags

- Java Standard Tag Library (JSTL)
  - Taglib directive
    - <%@ taglib prefix="c" uri="http://java.sun.com/jstl/core" %>
  - Core
    - <c:out value="${anExpression}"/>
  - SQL
  - XML
  - Format

Custom tags

- Implement your own tags
- Create a Tag Library Definition (tld) file
- Extend predefined classes
- Specify your library in a @taglib directive
- Use like JSTL tags

Expression language

- Refer to Java Beans and other common classes
- ${expression} can appear as tag attributes or (since JSP 2.0) in the page itself
- Several implicit objects:
  - header
    - ${header["user-agent"]}
  - param
    - ${param['name']}
    - ${param.name}

Deferred evaluation

- EL originally conceived as a read-only notation
  - ${expression} reads value of property
- JSF (see later) also requires ability to update values
- EL therefore added *deferred evaluation of expressions*
  - Syntax: #{expression}