sourcemind

Advanced Java and Spring course

Traditional Java Web Applications

Course agenda

- Traditional Java Web Applications
- Spring Fundamentals
- Spring Boot
- Spring Web
- Spring Security
- Deployment

sourcemind

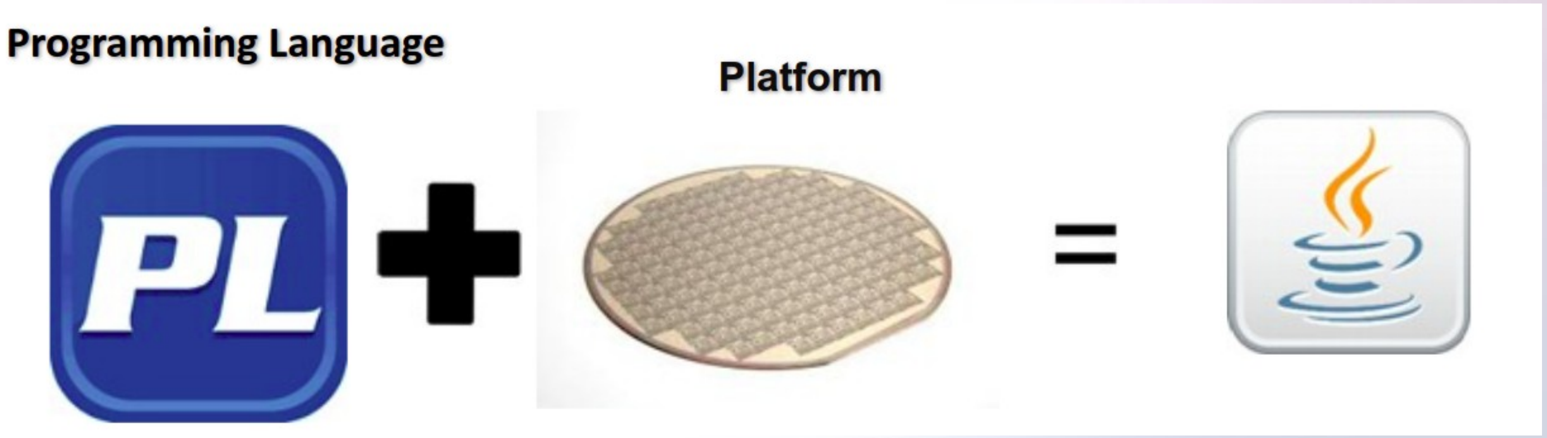
Lesson agenda

- Java platforms
- Basic web architecture
- JEE platform and its components
- Servlet and its implementation
- Tomcat
- Java Server Pages
- JDBC, JPA annotations
- Connection pooling

sourcemind

Java

Java is a programming language and a platform.



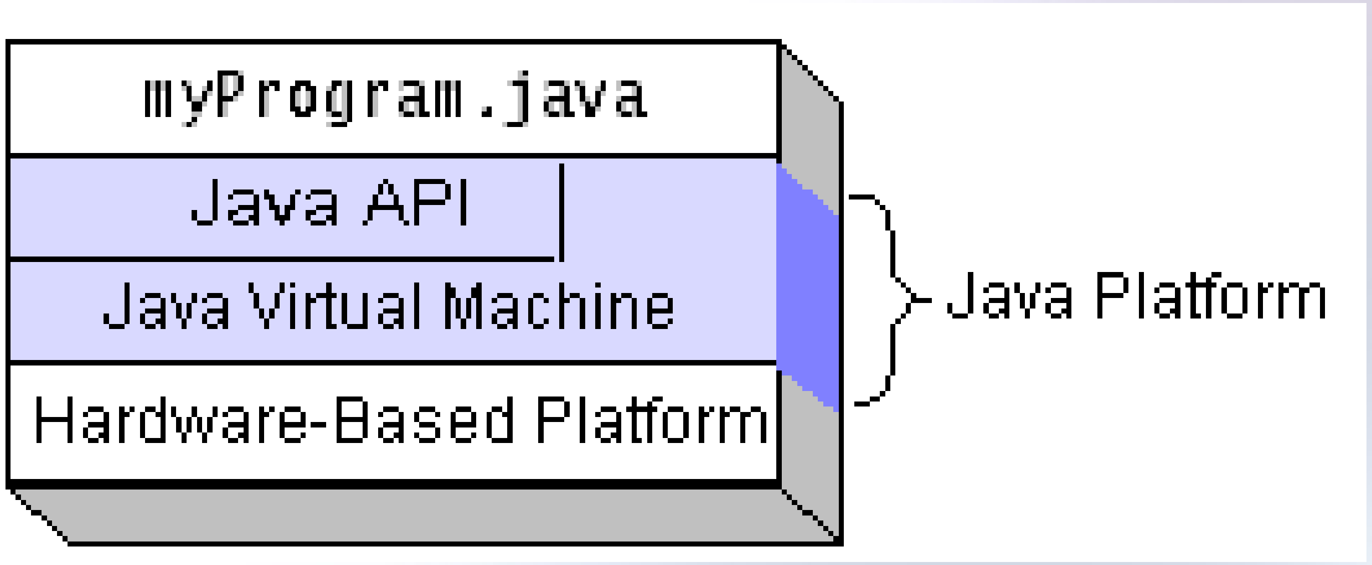
Famous platforms

sourcemind

The Java platform

The Java platform has two components:

- The Java Virtual Machine (JVM)
- The Java Application programming interface (Java API)



Java platforms

There are four platforms of the Java programming language:

- Java Platform, Standard Edition (Java SE)
- Java Platform, Enterprise Edition (Java EE)
- Java Platform, Micro Edition (Java ME)
- Java Platform, Effects (Java FX)

- Java Card
- PersonalJava(discontinued)

Java SE

It has concepts for developing software for:

- Desktop based (standalone) applications
- CUI (command user interface) applications
- GUI (graphical user interface) applications
- Database Interaction applications
- Distributed applications (RMI)
- XML parsing applications

Java EE

It has concepts to develop software for:

- Web applications (Servlets/JSPs/JSF)
- Enterprise applications (EJB)
- Interoperable applications (Webservices)

These applications are called high-scale applications. Some examples are:- banking and insurance-based applications.

Java ME

It has concepts to develop software for consumer electronic devices means embedded systems, like mobile and electronic level applications

sourcemind

Java FX (Eff=F, ects=X)

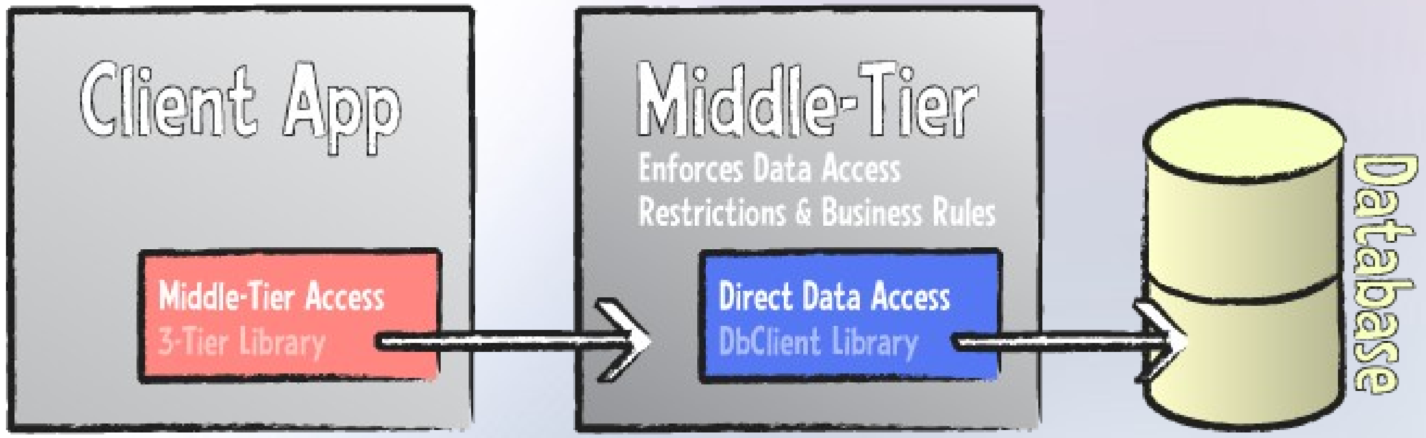
It provides concepts for developing rich internet applications with more graphics and animations.

It's an extension concept to swing applications of Java SE and it's included as part of Java SE software.

sourcemind

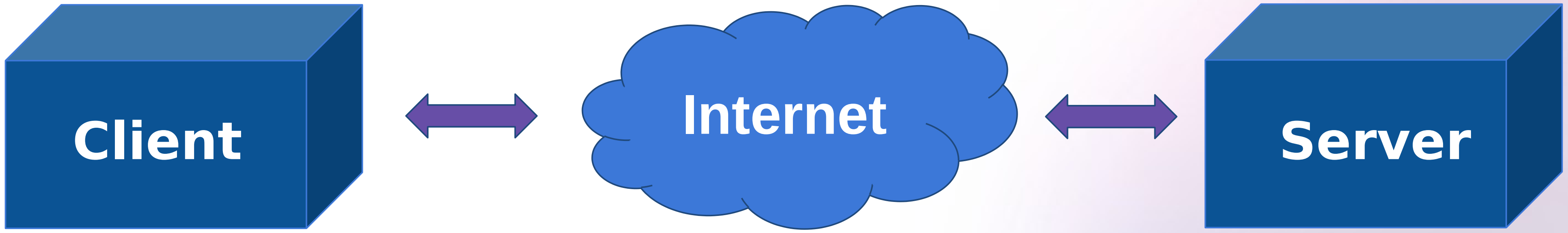
Overview of Enterprise Applications

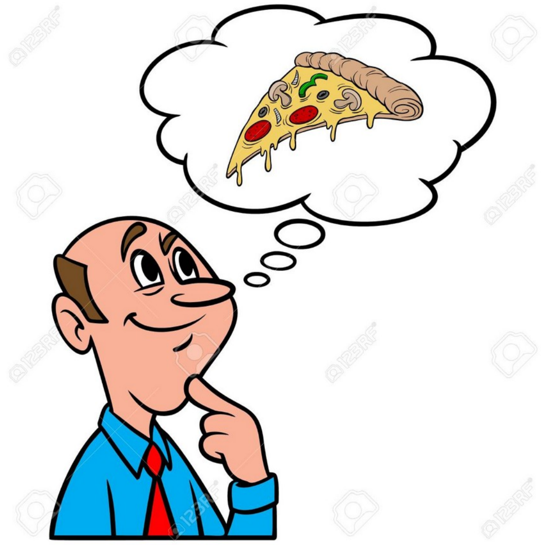
- Large-scale
- Scalable
- Reliable
- Secure
- Multi-tiered
    - Client tier
    - Middle tier
    - Data tier

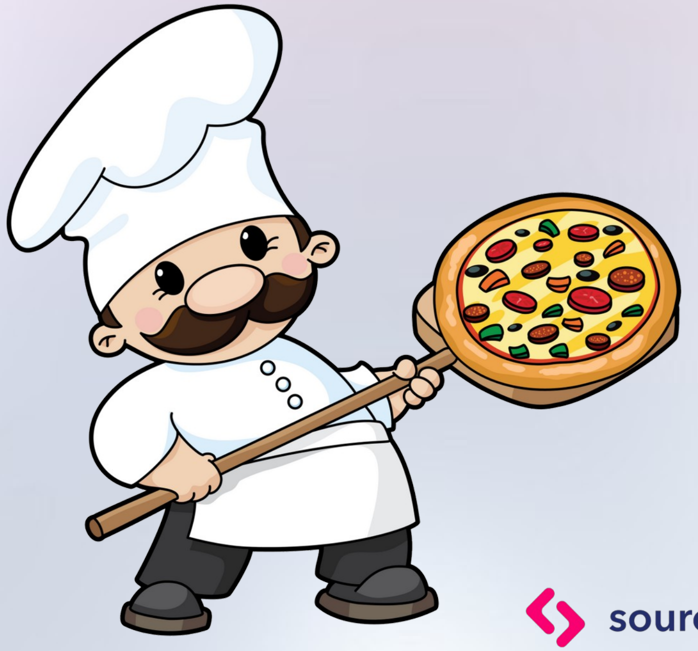


sourcemind

Basic Web Architecture



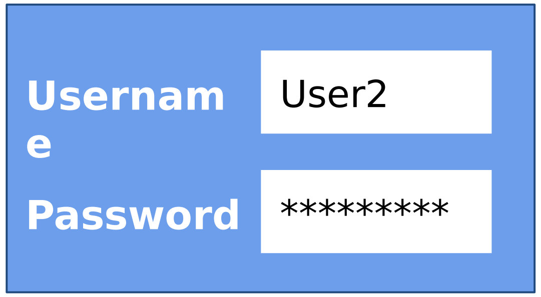




sourcemind

Web Application (3-Tier Application)

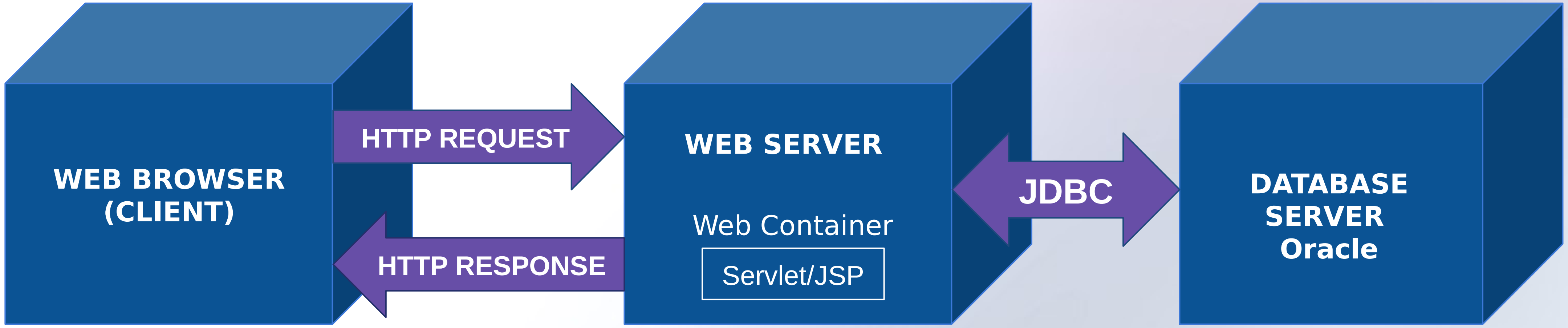
Login.html



Usernam e	User2
Password	*********

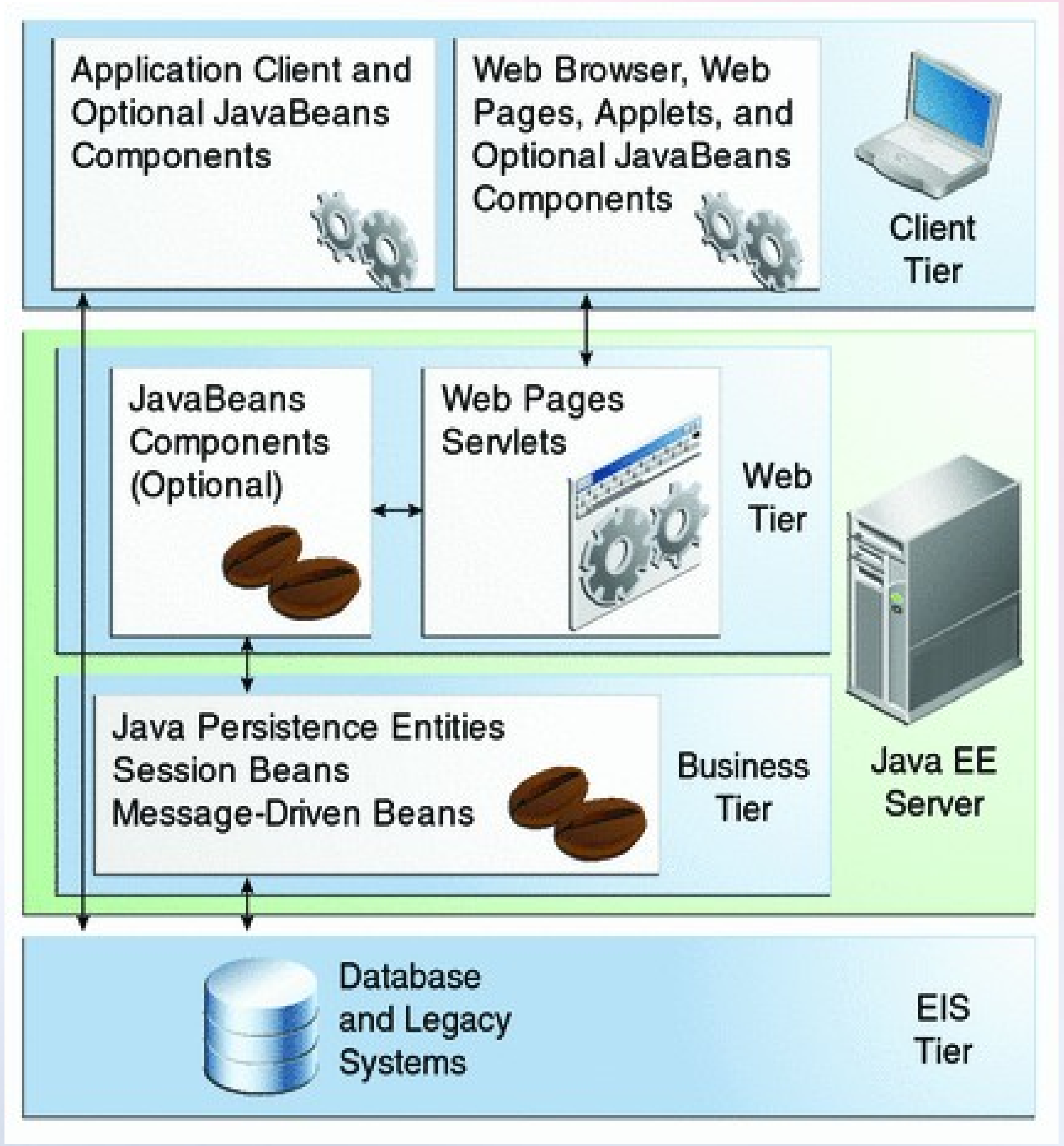
Login Table

Username	Password
User1	Password1
User2	Password2
User3	Password3



JEE components

- Application clients and applets are components that run on the client.
- Java Servlet, JavaServer Faces, and JavaServer Pages (JSP) technology components are web components that run on the server.
- Enterprise JavaBeans (EJB) components are business components that run on the server.


Application Client and Optional JavaBeans Components
Web Browser, Web Pages, Applets, and Optional JavaBeans Components
Client Tier
JavaBeans Components (Optional)
Web Pages Servlets
Web Tier
Java Persistence Entities Session Beans Message-Driven Beans
Business Tier
Java EE Server
Database and Legacy Systems
EIS Tier

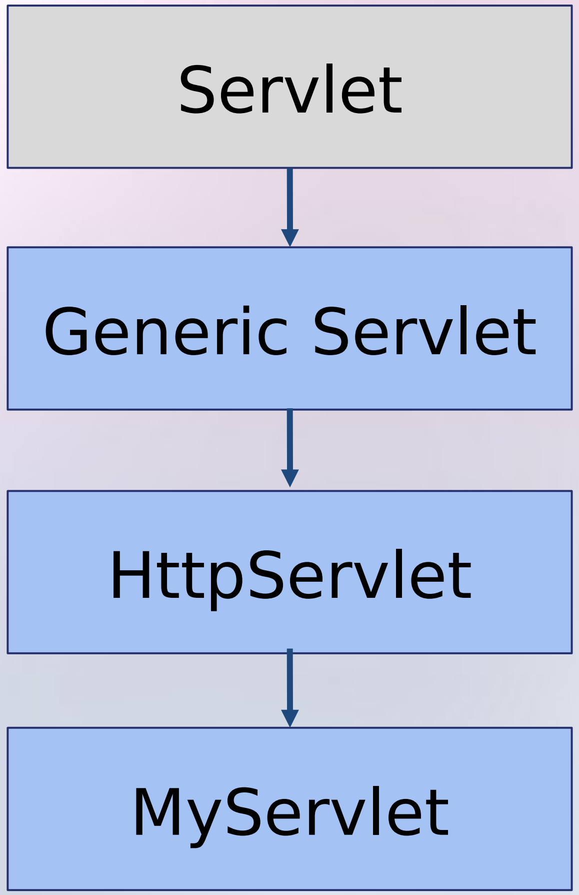
sourcemind

Servlet implementation

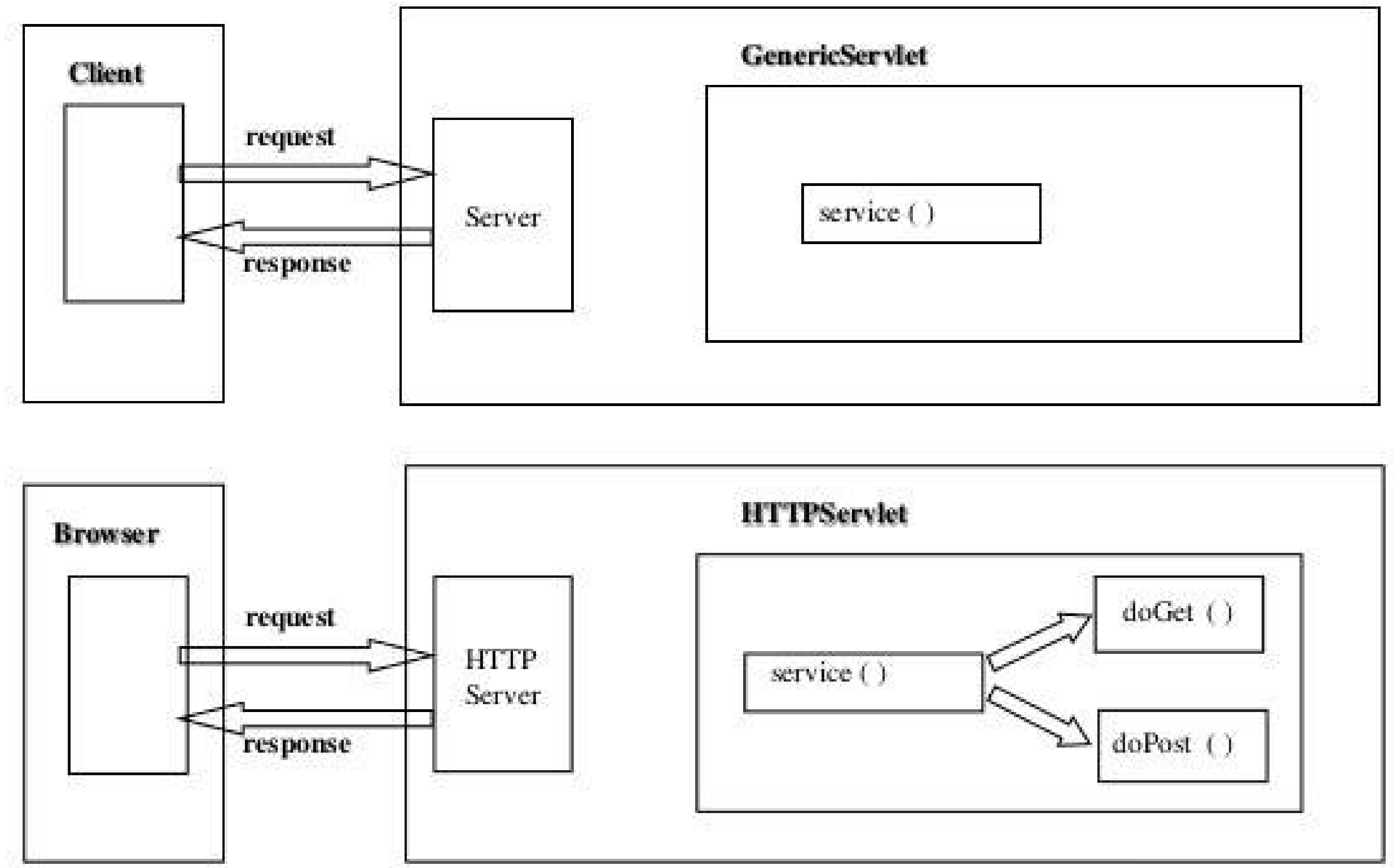
- javax.servlet
- javax.servlet.http

Generic Servlet

```
public abstract void service(ServletRequest request, ServletResponse response)
         throws ServletException, java.io.IOException
```



Generic Servlet vs. HTTP Servlet



sourcemind

GenericServlet methods

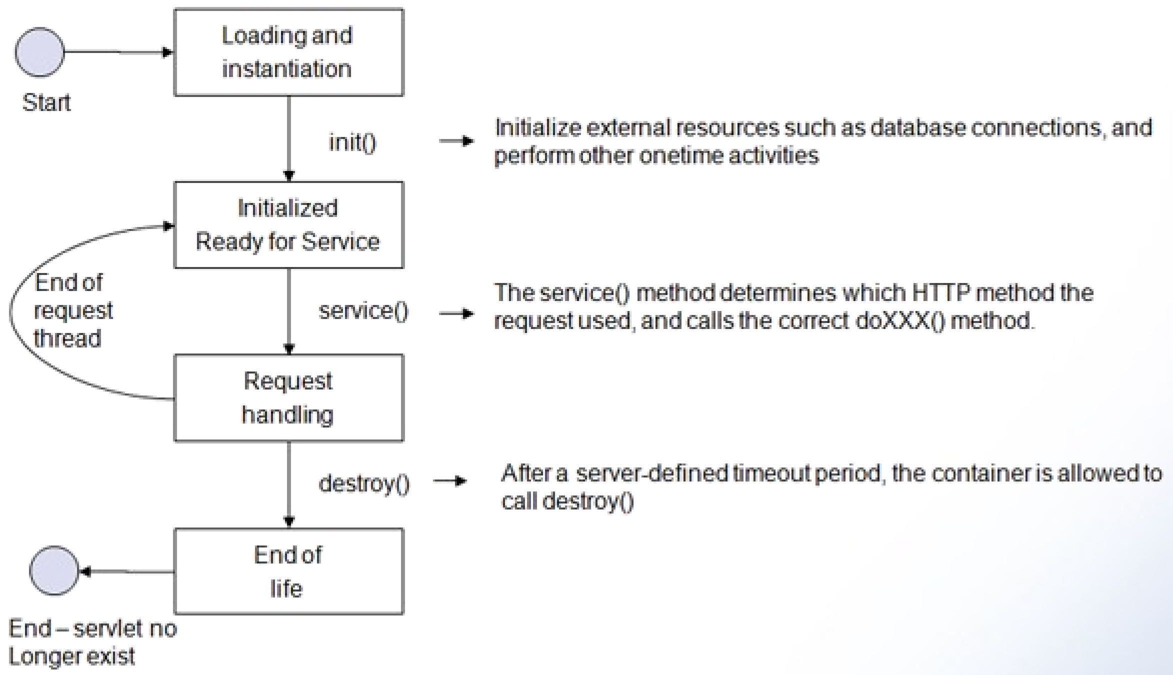
Methods	**Description**
void init(), void **init(*ServletConfig*)**	This method initializes the Servlet object.
abstract *void* **service(*ServletRequest, ServletResponse*)**	This method puts the Servlet in the service.
void **destoy()**	This method destroy the Servlet object.
String **getServletInfo()**	This method gets the Servlet associated information.
ServletConfig **getServletConfig()**	This method gets the ServletConfig object.
String **getInitParameter(*String str*)**	This method returns the value of a parameter named *str*
Enumeration **getInitParameterNames()**	This method returns all the parameter names associated with the Servlet.
ServletContext **getServletContext()**	This method returns an object of ServletContext.
String **getServletName()**	This method returns the name of this Servlet object.

HttpServlet methods

Methods	**Description**
void **doGet(*HttpServletRequest* req, *HttpServletResponse* res)**	This method allows a Servlet to handle the ***get*** request.
void **doPost(*HttpServletRequest* req, *HttpServletResponse* res)**	This method allows a Servlet to handle the ***post*** request.
void **doPut(*HttpServletRequest* req, *HttpServletResponse* res)**	This method allows a Servlet to handle the **put** request.
void **doTrace(*HttpServletRequest* req, *HttpServletResponse* res)**	This method allows a Servlet to handle the ***trace*** request.
void **doHead(*HttpServletRequest* req, *HttpServletResponse* res)**	This method allows a Servlet to handle the **HTTP *head*** request.
void **doDelete(*HttpServletRequest* req, *HttpServletResponse* res)**	This method allows a Servlet to handle the ***delete*** request.
void **doOptions(*HttpServletRequest* req, *HttpServletResponse* res)**	This method allows a Servlet to handle the ***options*** request.

sourcemind

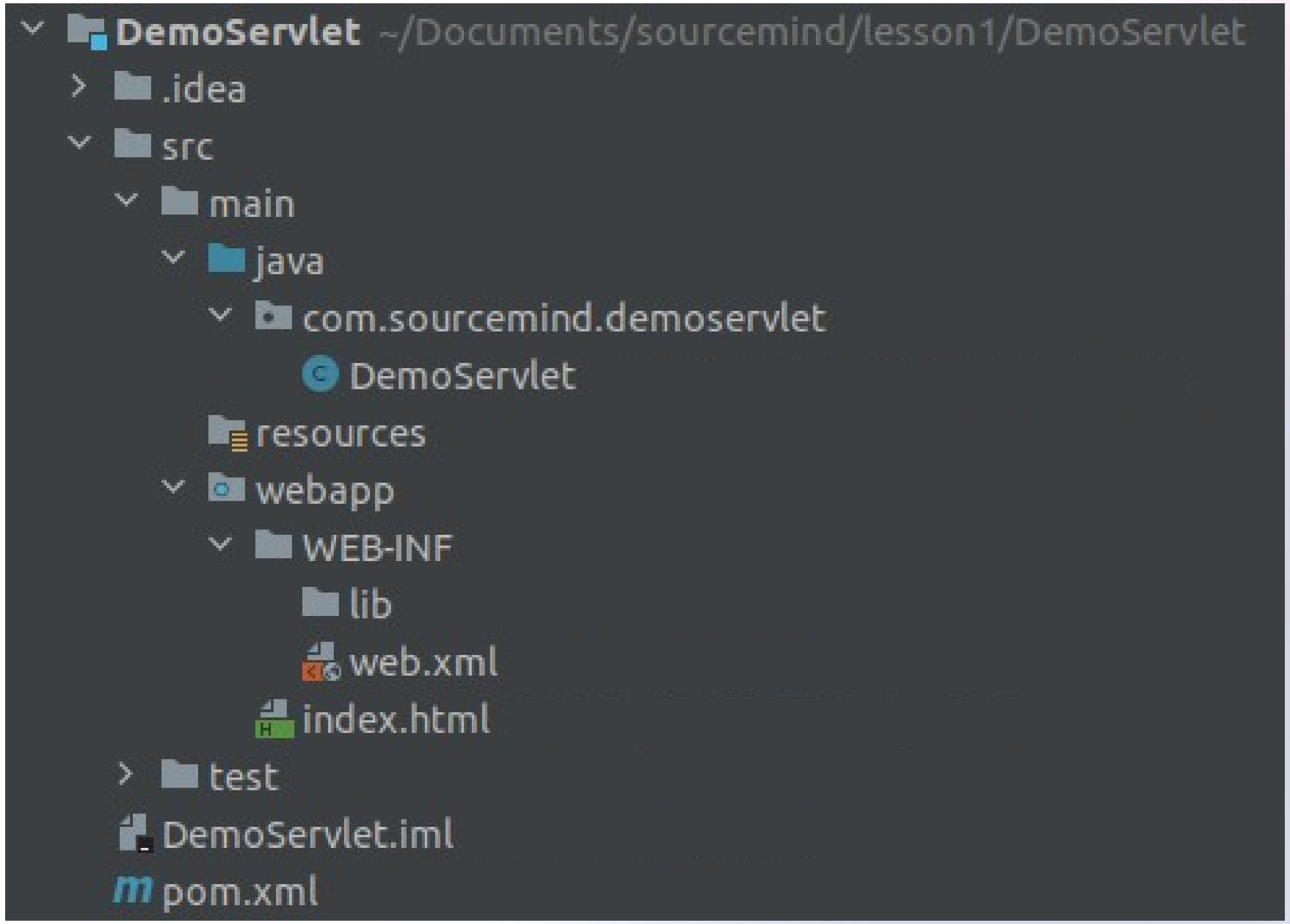
Life cycle of Servlet


Start
Loading and instantiation
init()
Initialize external resources such as database connections, and perform other onetime activities
Initialized Ready for Service
End of request thread
service()
The service() method determines which HTTP method the request used, and calls the correct doXXX() method.
Request handling
destroy()
After a server-defined timeout period, the container is allowed to call destroy()
End of life
End – servlet no Longer exist

- Loading of Servlet
- Creating instance of Servlet
- Invoke init() once
- Invoke service() repeatedly for each client request
- Invoke destroy()

sourcemind

My first servlet project structure



```
DemoServlet ~/Documents/sourcemind/lesson1/DemoServlet
  .idea
  src
    main
      java
        com.sourcemind.demoservlet
          DemoServlet
      resources
      webapp
        WEB-INF
          lib
          web.xml
        index.html
    test
  DemoServlet.iml
  pom.xml
```

sourcemind

index.html and web.xml

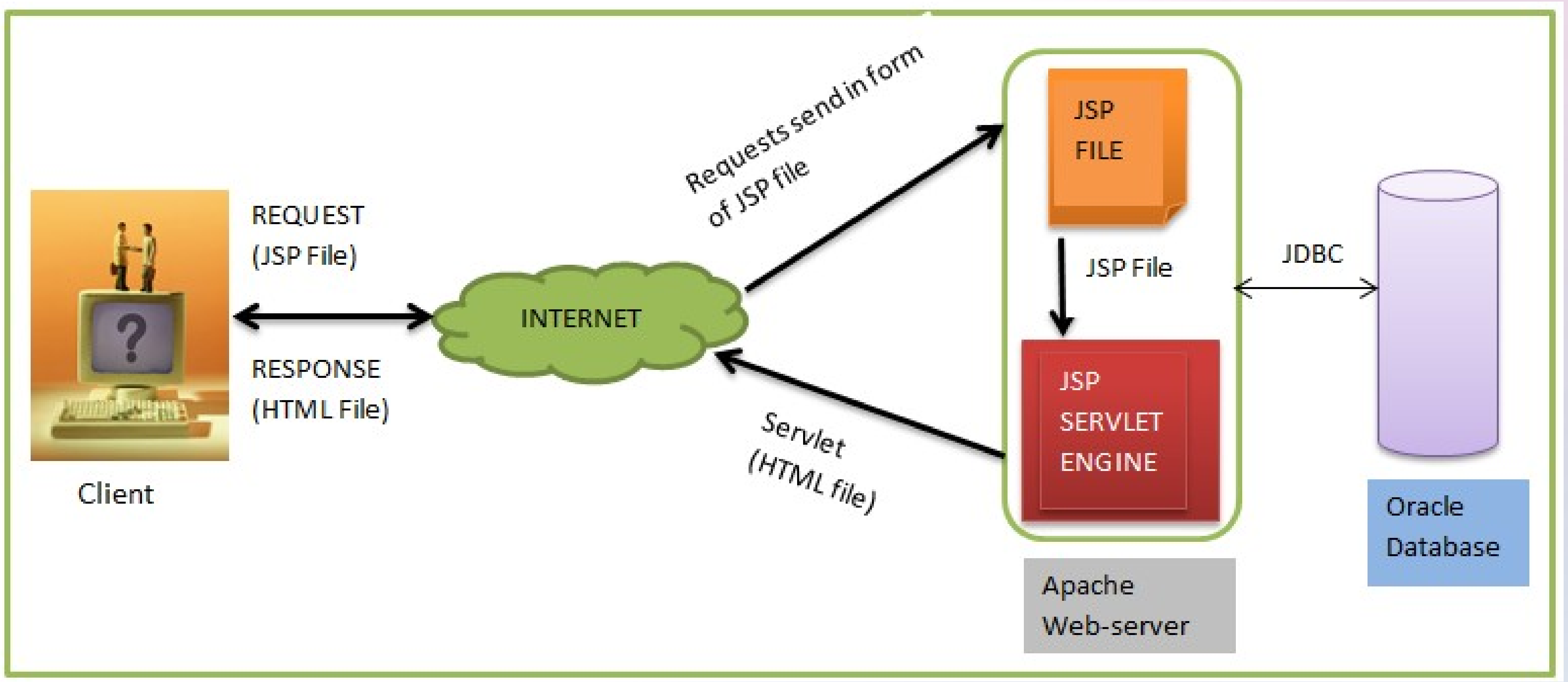
```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Title</title>
  </head>
  <body>
    <h1>Hello World!</h1>
    <br/>
    <a href="hello-servlet">Hello
Servlet</a>
  </body>
</html>
```

```xml
<?xml version="1.0" encoding="UTF-8"?>
<web-app ...>
  <servlet>
    <servlet-name>demoServlet</servlet-name>
    <servlet-class>
          com.sourcemind.demoservlet.DemoServle
    t
              </servlet-class>
            </servlet>
            <servlet-mapping>

    <servlet-name>demoServlet</servlet-name>
              <url-pattern>/hello-servlet</url-
    pattern>
            </servlet-mapping>
          </web-app>
```

sourcemind

Java Server Pages (JSP)



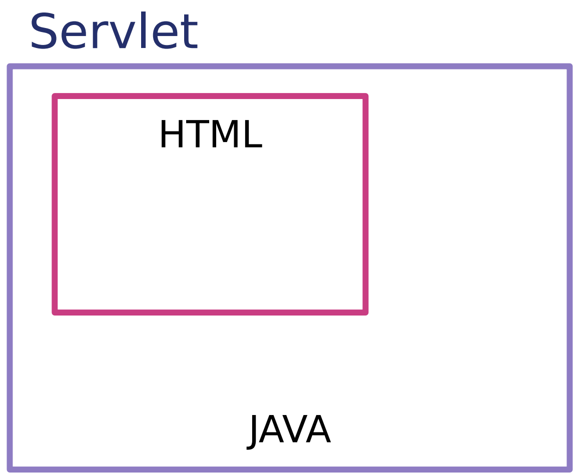
sourcemind

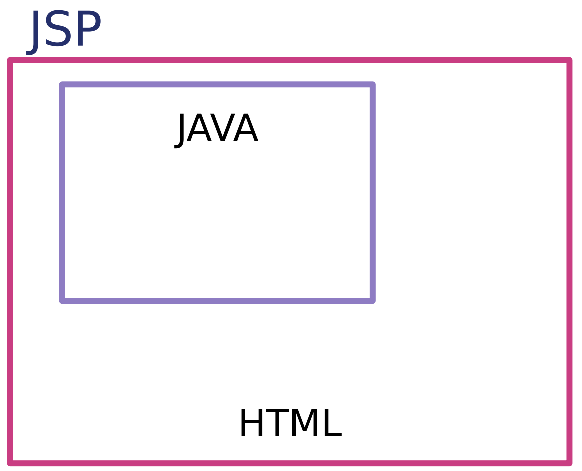
JSP life cycle

1. **Translation**: When a regular code file is written with HTML and JSP tags, and the file extension is “.jsp”, it is translated into a servlet file.
2. **Compilation**: In this step, a servlet file previously available in the step is created into a servlet class file.
   Ex: emp_jsp.java into emp_jsp.class
   This compilation generally happens during deployment.
3. **Loading**: The class created gets loaded into the container by class loader.
4. **Initiation**: An instance of this class, an object is created. The container can manage one or more instances as per the need.
5. **Initialization**: jspInit() method is called by the container and the initialization takes place.
6. **Processing**: All the services and requests are processed. _jspService(req,resp) is invoked.
7. **Destroy**: In this phase, jspDestroy() is called and the instance created is destroyed as all the actions needed are processed and done.

sourcemind

Servlet vs JSP





Servlets

- Servlet is a Java program which supports HTML tags too.
- Generally used for developing business layer(the complex computational code) of an enterprise application.
- Servlets are created and maintained by Java developers.

JSP

- JSP program is a HTML code which supports java statements too. To be more precise, JSP embed java in html using JSP tags.
- Used for developing presentation layer of an enterprise application
- Frequently used for designing websites and used by web developers.

sourcemind

My first JSP file

Rename .html -> .jsp

```
<html>
  <body>
    Hello Sourcemind students!
    Current time is: <%= new java.util.Date() %>
  </body>
</html>
```

JSP file explained

```
<%-- JSP comment --%>
<html>
  <head>
    <title>Just a title</title>
  </head>
  <body>
    Hello Sourcemind students!
    Current time is: <%= new
java.util.Date() %>
  </body>
</html>
```

JSP Comments: start with a tag <%– and end with –%>

Basic HTML tags

JSP Expressions: start with a tag <%= and end with %>

sourcemind

JSP tags

- Declaration tag: <%! Declaration %>
  - Variables declaration
  - Methods declaration
- Expression tag: <%= Expression %>
  - Expression of values
  - Expression of variables
- Directives tag: <%@ directive name [attribute_name="value" attribute_name="value" ....]%>
  - Page Directive: <%@page import="java.io.*"%>
  - Include Directive: <%@include file="index.jsp"%>
- Action tag: <jsp: action … />
  - <jsp:include … />, <jsp:forward … />, <jsp:useBean … /> …

sourcemind

JSP tags

- Scriptlet: <% Executable Java code %>

Sample JSP code

```
<H2> Sample JSP </H2>
<% myMethod();%>
```

Translated Servlet code

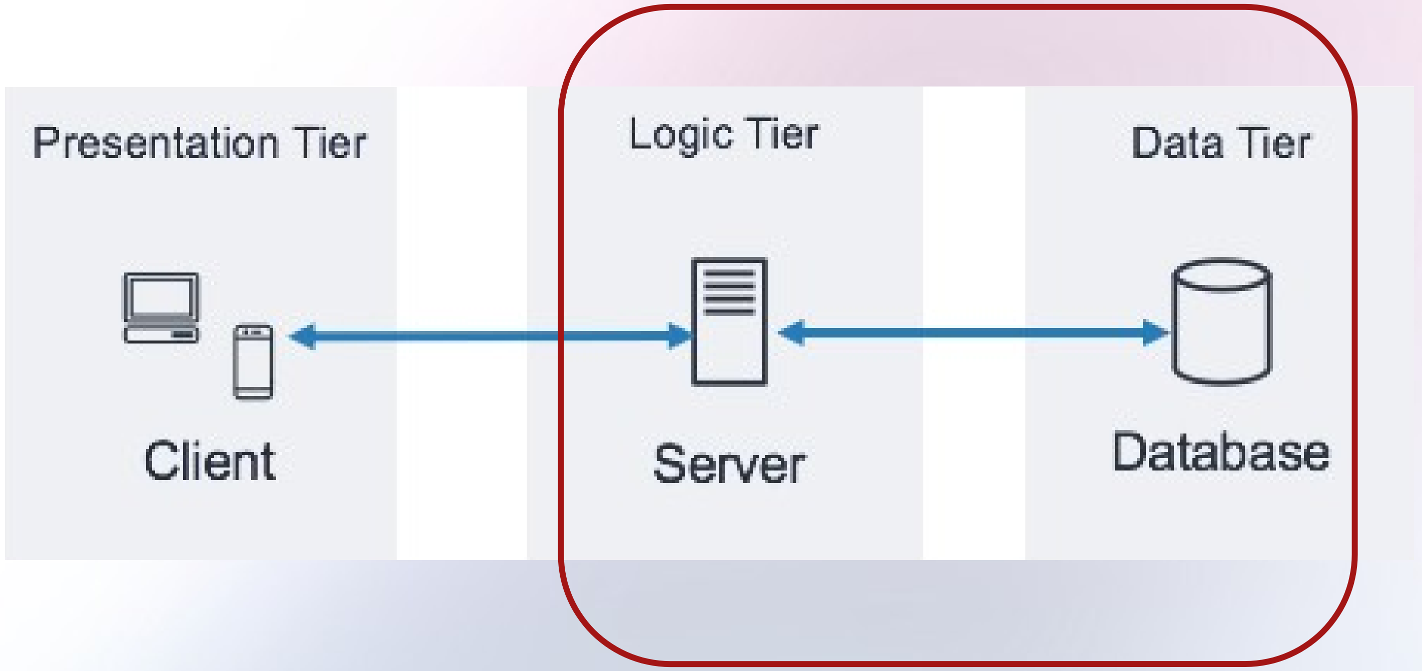
```
public void _jspService(    HttpServletRequest request,
HttpServletResponse response)
    throws ServletException, IOException {
    response.setContentType("text/html");
    HttpSession session = request.getSession();
    JspWriter out = response.getWriter();
    out.println("<H2> Sample JSP </H2>");
    myMethod();
}
```

JSP Implicit Objects

Object	Class
out	javax.servlet.jsp.JspWriter
request	javax.servlet.http.HttpServletRequest
response	javax.servlet.http.HttpServletResponse
session	javax.servlet.http.HttpSession
application	javax.servlet.ServletContext
exception	javax.servlet.jsp.JspException
page	java.lang.Object
pageContext	javax.servlet.jsp.PageContext
config	javax.servlet.ServletConfig

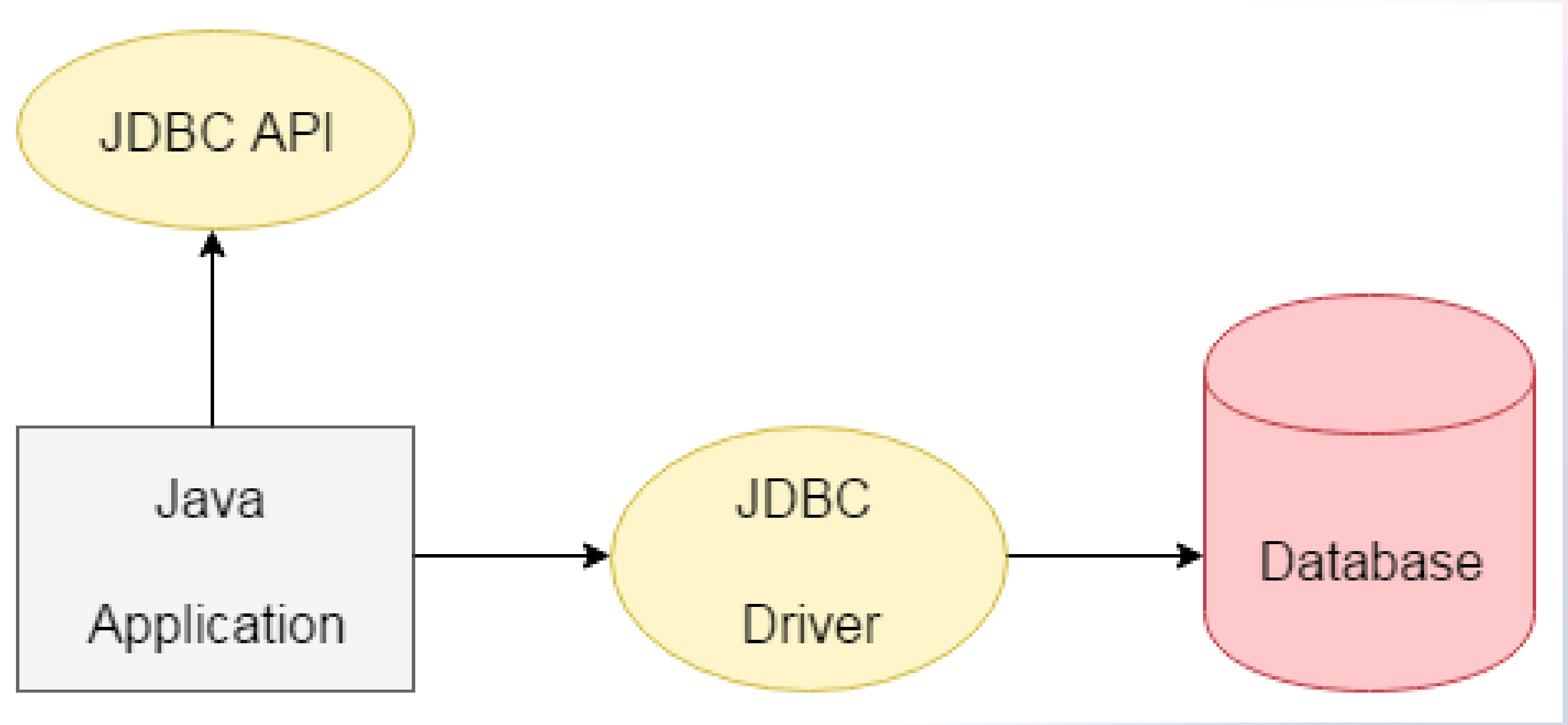
Database Connectivity

- JDBC
  Java Database Connectivity
- ORM
  Object–relational mapping
- JPA
  Java Persistence API



sourcemind

Java Database Connectivity



java.sql package

JDBC with 5 steps

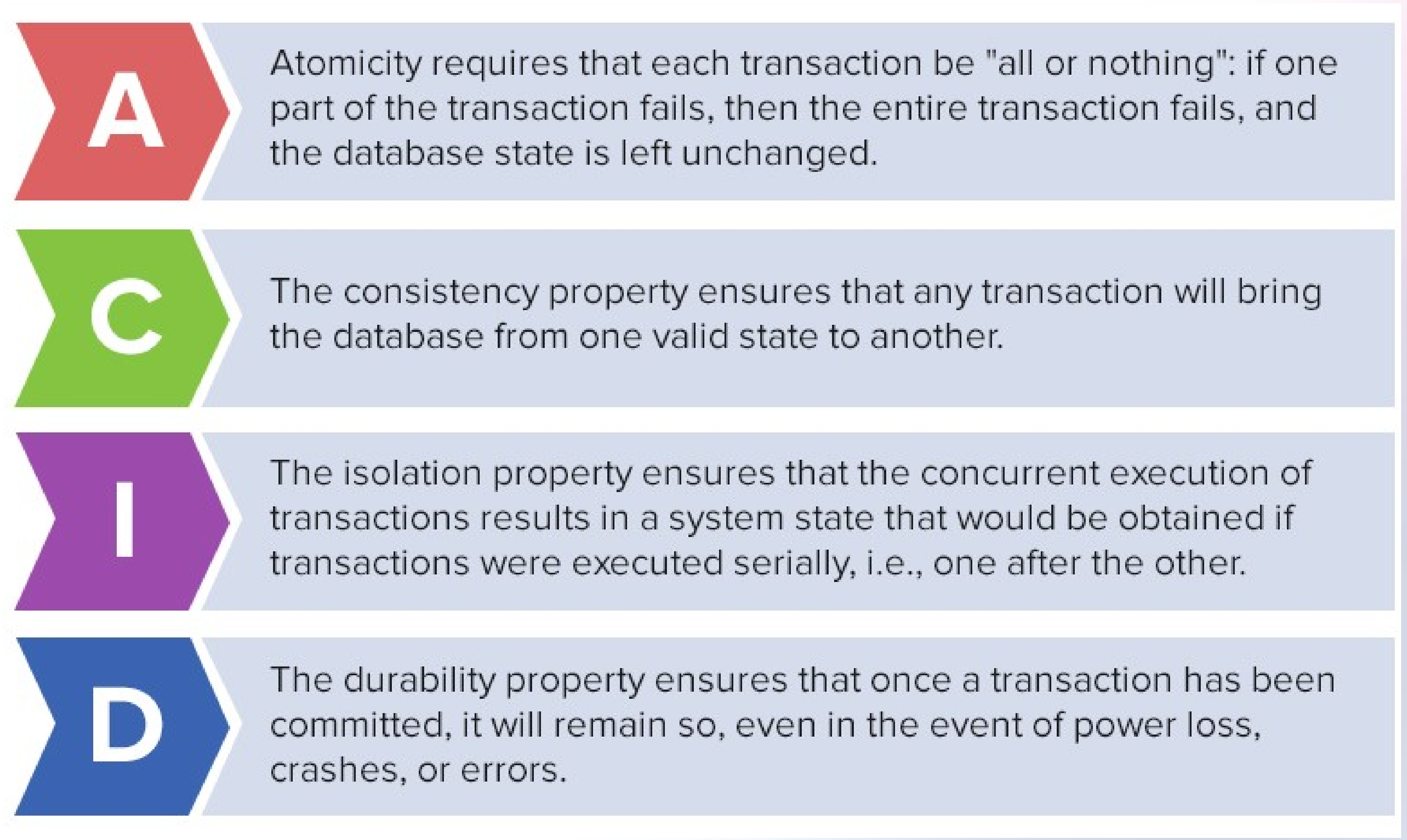
There are 5 steps to connect any java application with the database using JDBC.

- Register the Driver class
- Create connection
- Create statement
- Execute queries
- Close connection

The list of popular interfaces and classes of JDBC API

- Driver interface
- Connection interface
- Statement interface
- PreparedStatement interface
- CallableStatement interface
- ResultSet interface
- ResultSetMetaData interface
- DatabaseMetaData interface
- RowSet interface
- DriverManager class
- Blob class
- Clob class
- Types class

Transaction Management in JDBC

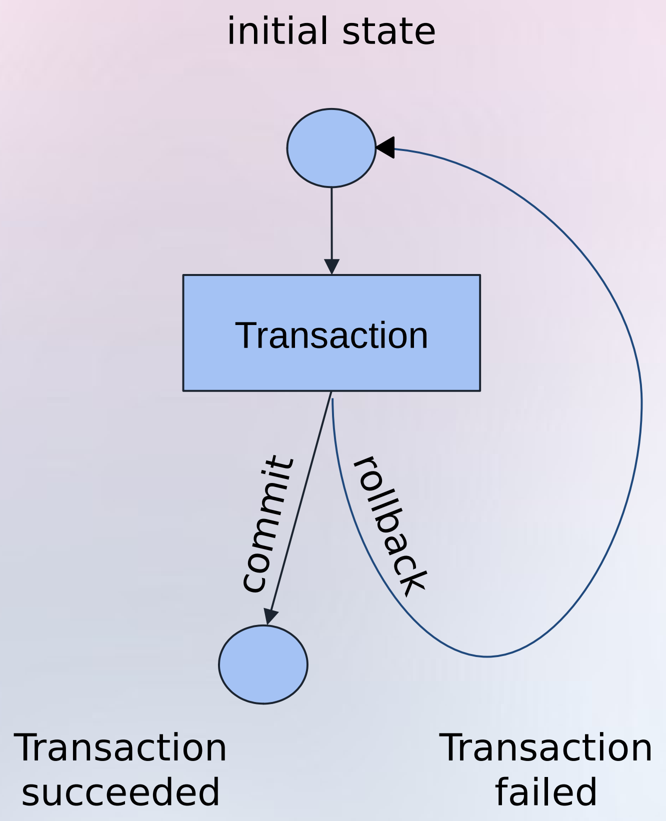


A Atomicity requires that each transaction be "all or nothing": if one part of the transaction fails, then the entire transaction fails, and the database state is left unchanged.

C The consistency property ensures that any transaction will bring the database from one valid state to another.

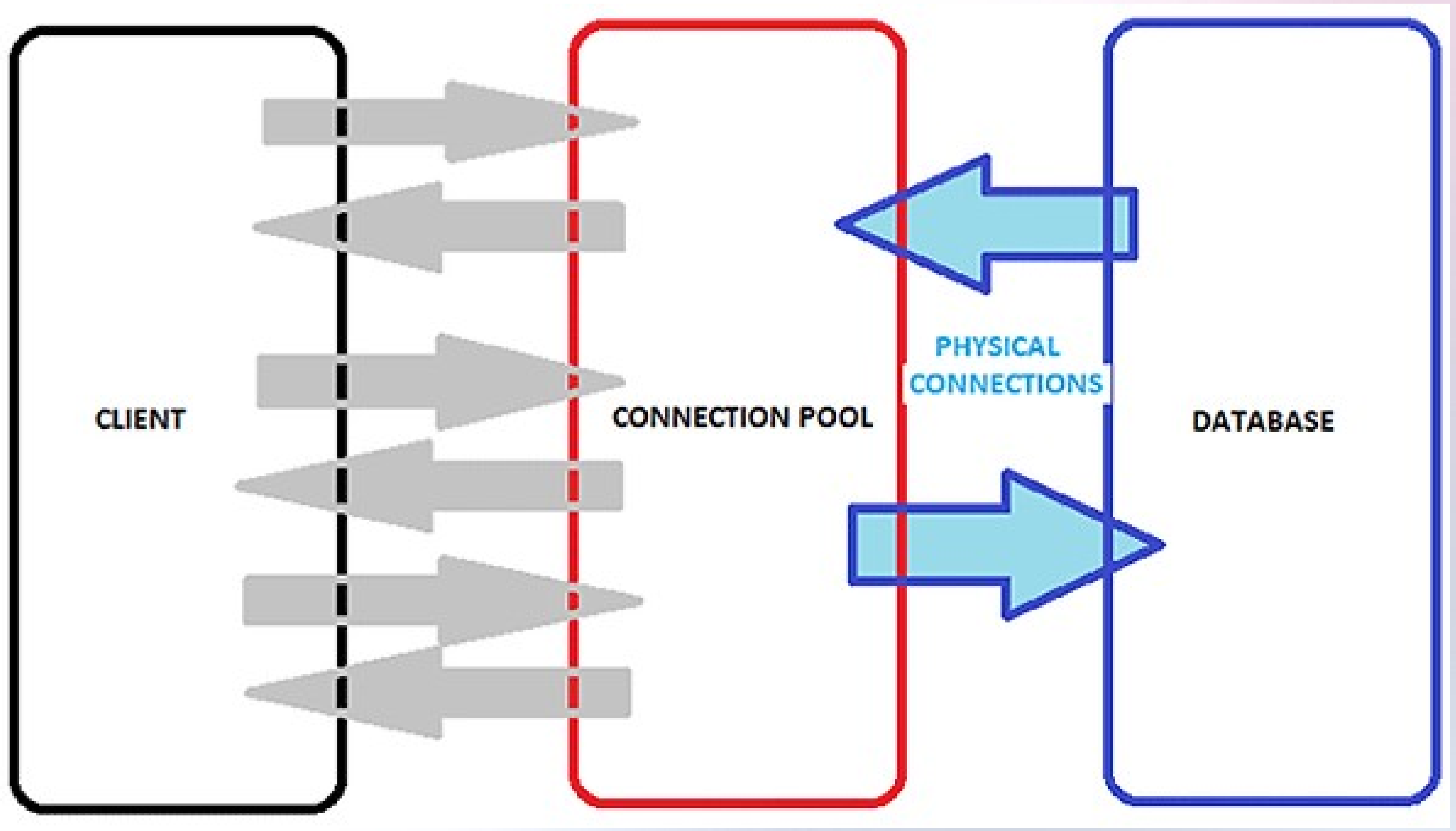
I The isolation property ensures that the concurrent execution of transactions results in a system state that would be obtained if transactions were executed serially, i.e., one after the other.

D The durability property ensures that once a transaction has been committed, it will remain so, even in the event of power loss, crashes, or errors.

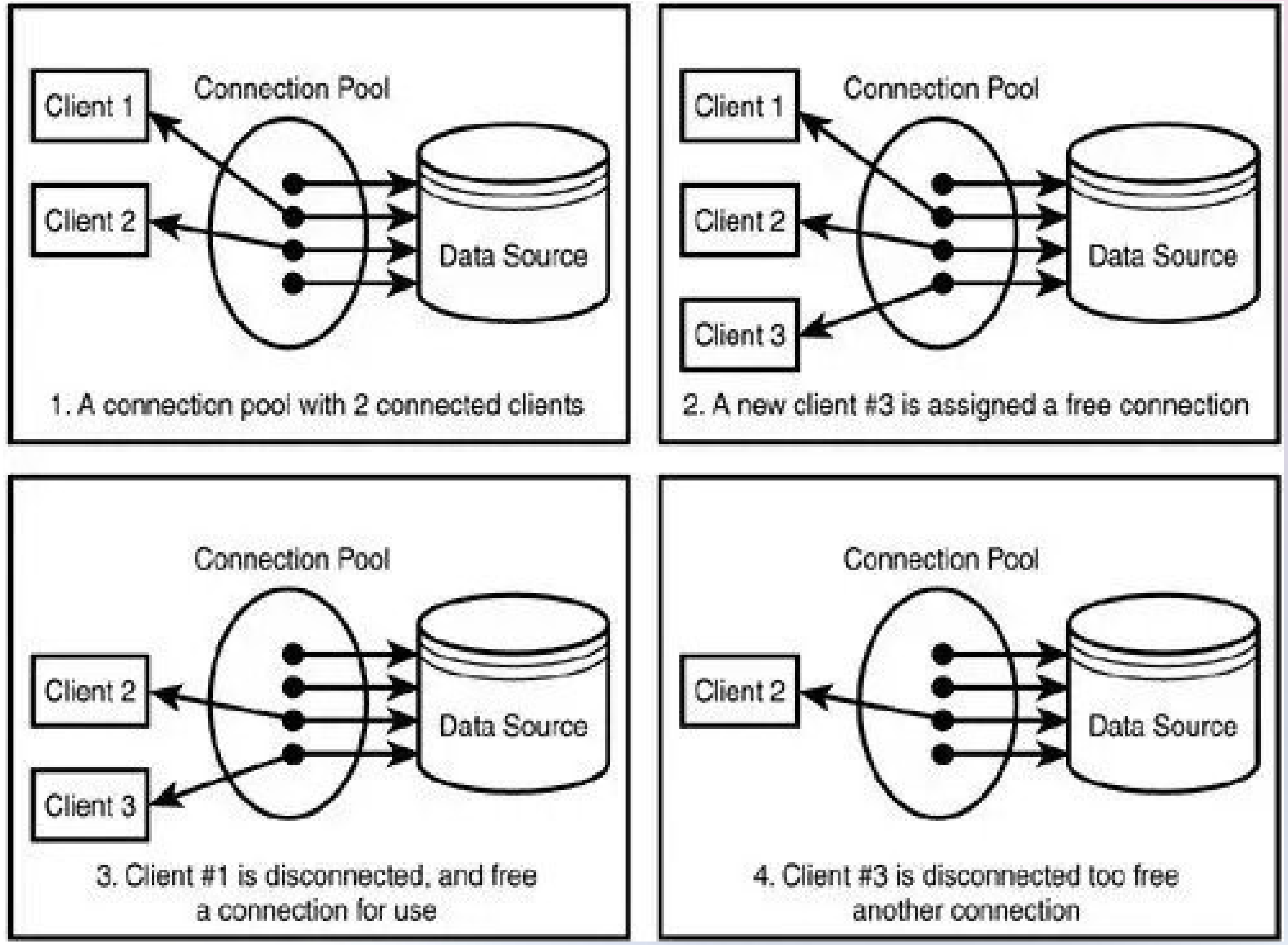


sourcemind

Connection Pooling

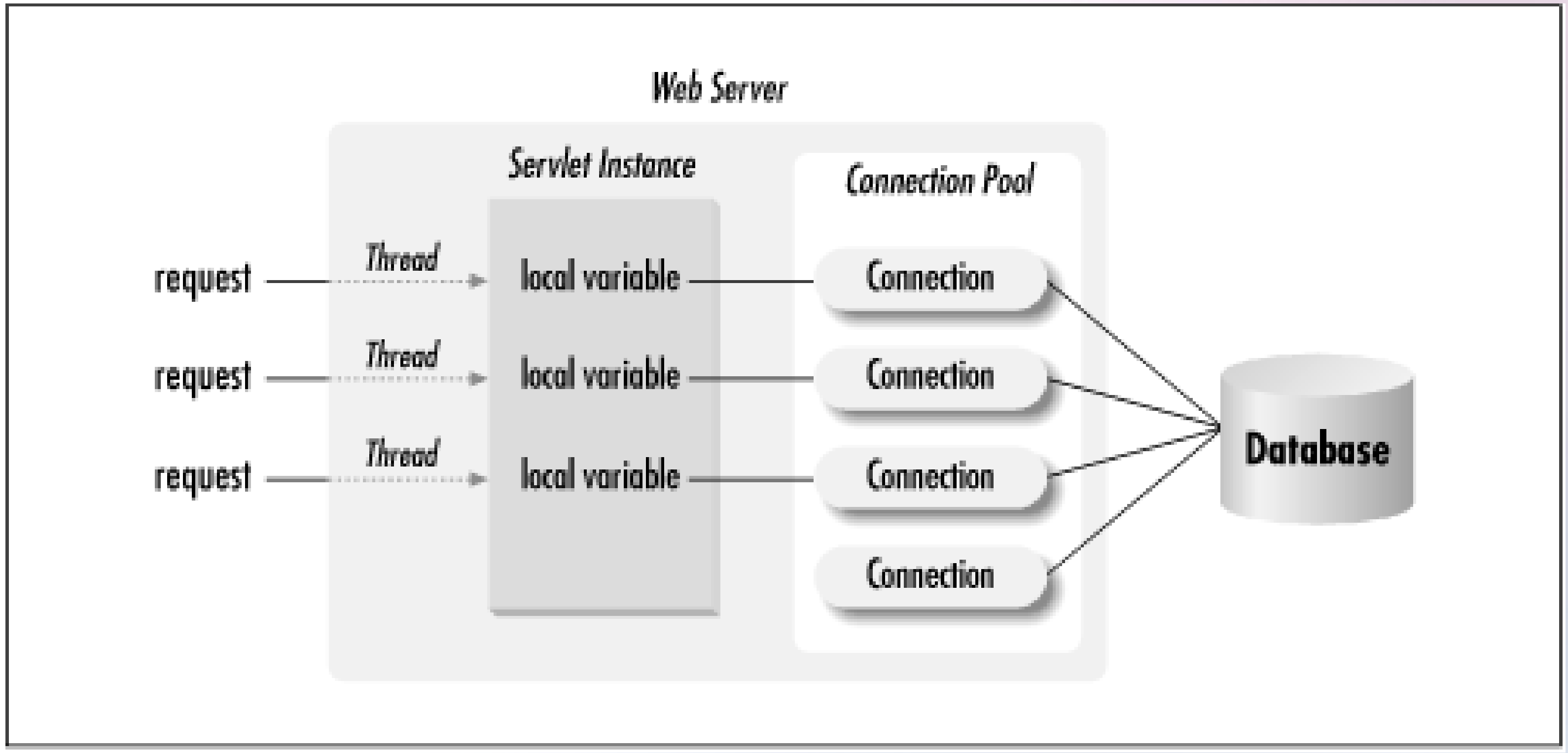


Connection Pooling


Connection Pool
Client 1
Client 2
Data Source
1. A connection pool with 2 connected clients
Connection Pool
Client 1
Client 2
Client 3
Data Source
2. A new client #3 is assigned a free connection
Connection Pool
Client 2
Client 3
Data Source
3. Client #1 is disconnected, and free a connection for use
Connection Pool
Client 2
Data Source
4. Client #3 is disconnected too free another connection

sourcemind

Connection Pooling


Web Server
Servlet Instance
Connection Pool
request
Thread
local variable
Connection
request
Thread
local variable
Connection
request
Thread
local variable
Connection
Connection
Database