# 2

# Fundamentals of JEE 7 Technology

# Objectives

After completing this lesson, you should be able to do the following:

- Describe the Java Platform, Enterprise Edition (Java EE) platform
- Define the components of a Java EE application
- Identify the deployment options for a Java EE application
- List the security options available in Java EE applications

# Java Platform, Enterprise Edition

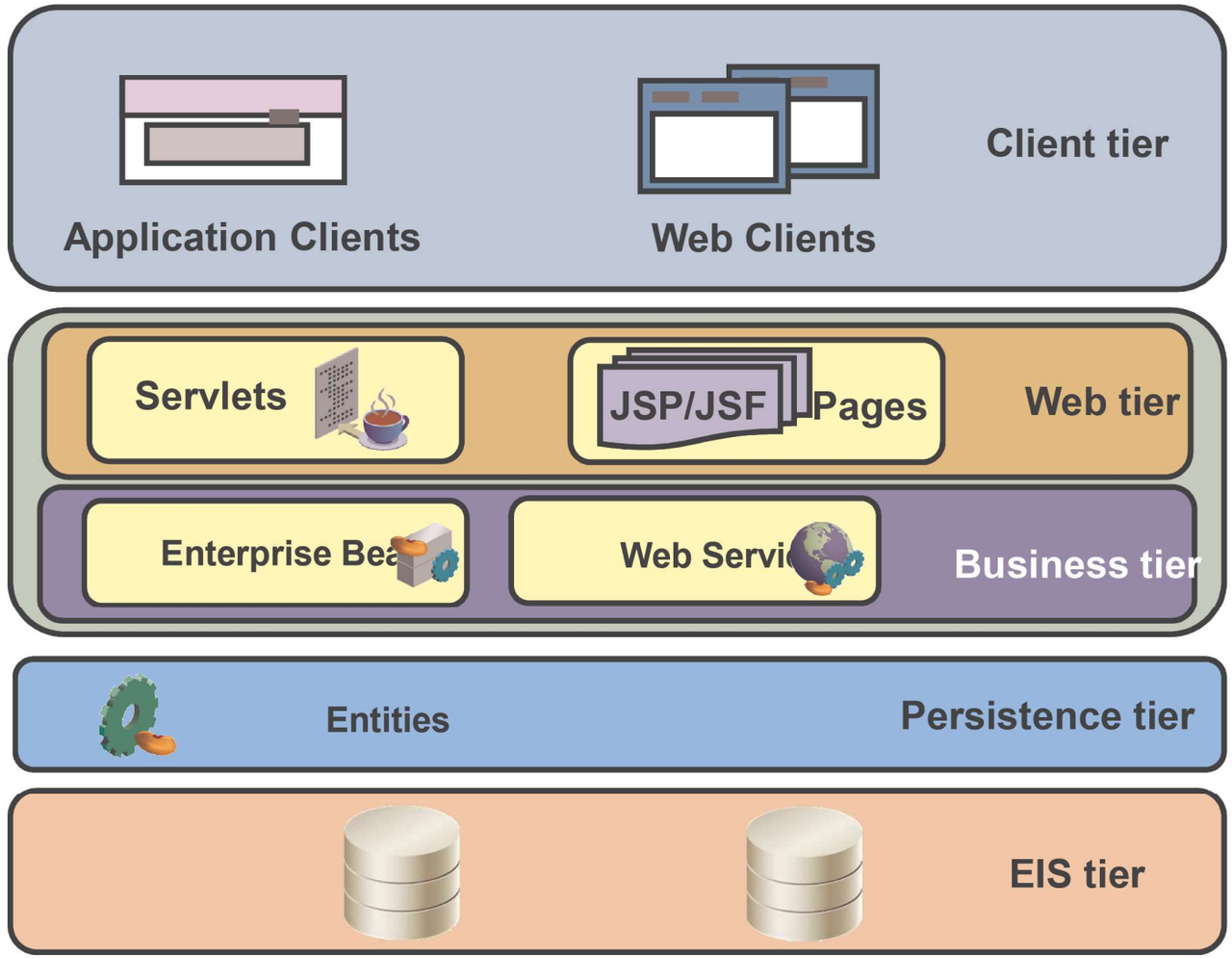
Java Platform, Enterprise Edition (Java EE) is a standard for developing and implementing enterprise wide applications:

- It provides support for multitier applications.
- It is designed to help improve the process of developing, deploying, and implementing enterprise wide applications.

# Java EE Platform

- Multitiered, distributed application model
- Supports component-based Java EE applications
- Distributes the application logic to the appropriate tier on the multitiered architecture

# Distributed Multitiered Applications



# Benefits of the Java EE Platform

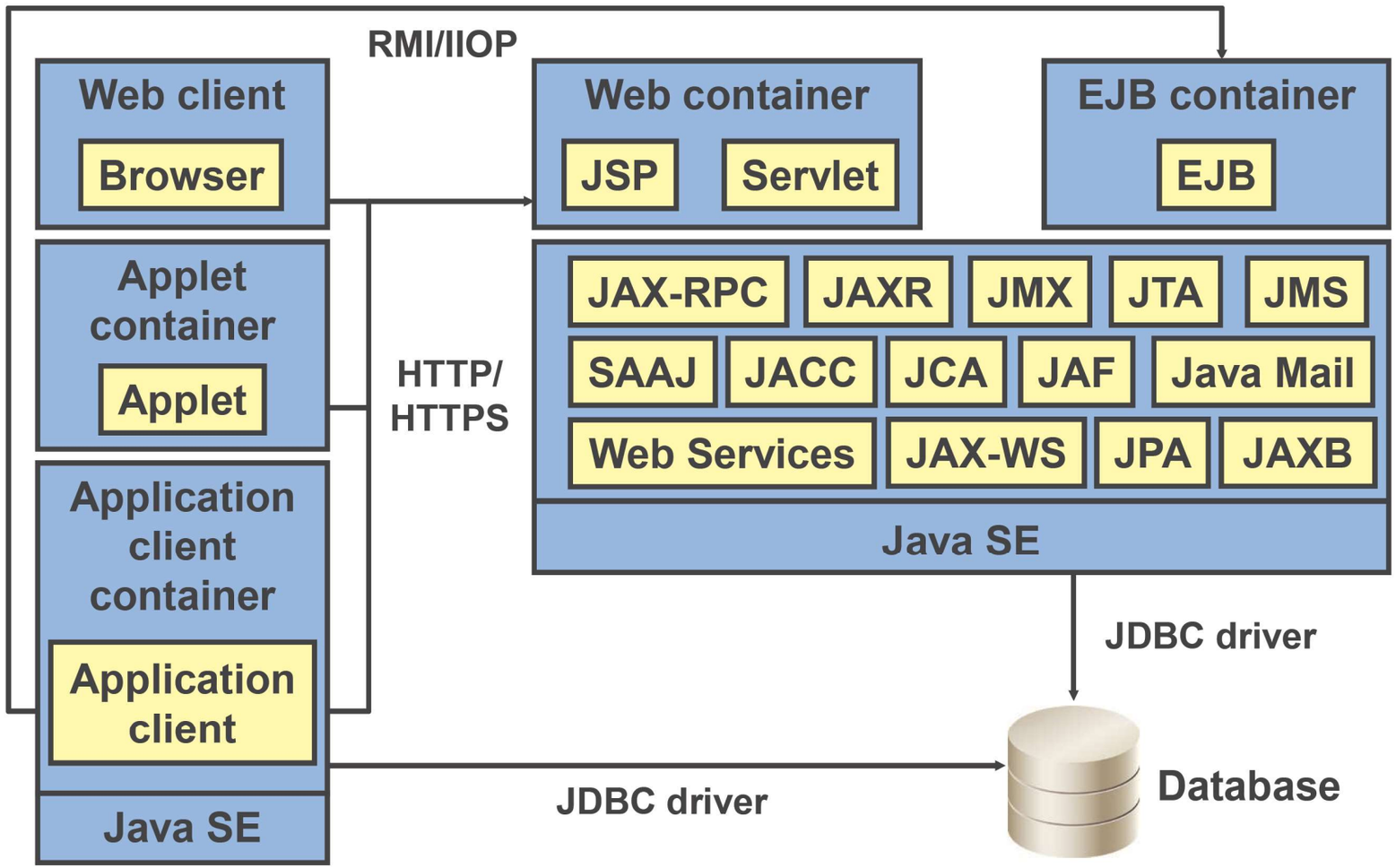
- "Write once, run anywhere" provides simplified component development.
- Multiple server products and vendors support the Java EE standard, thus giving more deployment choices.
- Integration with legacy systems through standard APIs is possible.
- It provides multiple development and design scenarios.
- It allows multiple clients to share server business logic.

# Benefits of the Java EE Platform

Java EE separates development tasks into specific skill areas:

- Web component developers create JSF components.
- Enterprise bean developers and business experts create business logic and rules.
- Application assembler and application deployer teams handle assembly and deployment.

# Java EE Platform and APIs



The Java EE platform provides a single development and design scenario to develop applications.

1. True
2. False

# Java EE Components

- Java EE applications are made up of components. A component:
  - Is an application-level software unit
  - Can be easily updated as business needs change
  - Is reusable
- There are several types of components:
  - Client-side components
  - Web components
  - Business-tier components

# Business-Tier Components

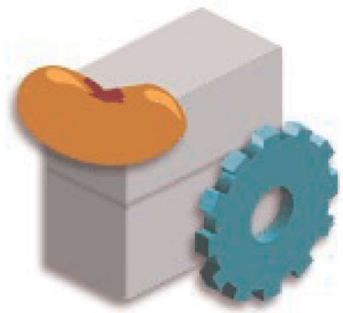
Business-tier components:

- Are implemented by remote components, such as Web services and EJBs
- Handle business logic
- Receive data from client programs
- Encapsulate business data and rules
- Process the data and communicate with the database and the client program
- Can be invoked by the Web-tier components

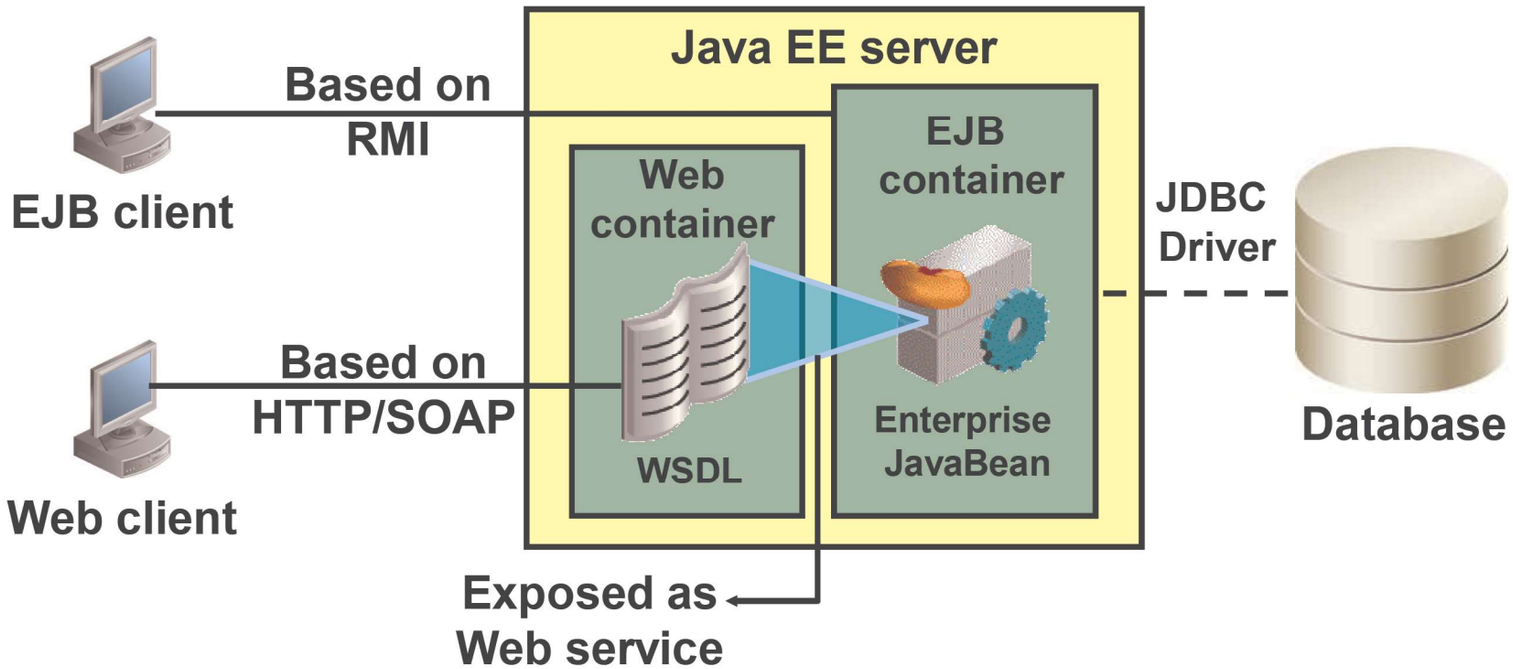
# Enterprise JavaBeans (EJB)

Enterprise JavaBeans:

- Is the server-side component written in Java
- Contains the business logic of an enterprise application
- Is hosted in EJB containers
- Is platform independent
- Provides remote services for clients
- Uses JDBC/JPA to connect to a database

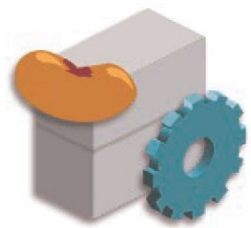


# Enterprise JavaBeans Application Architecture

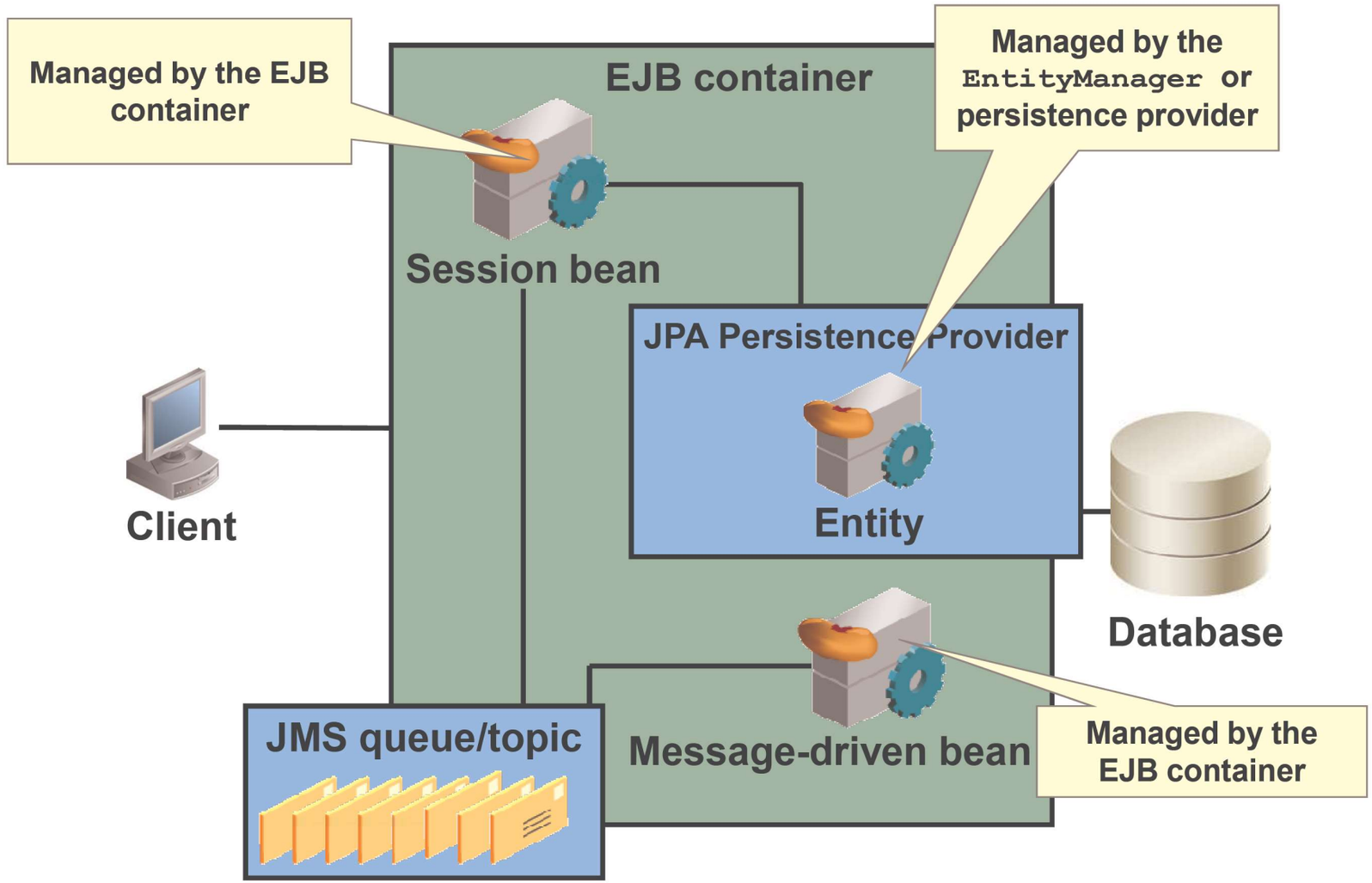


# Types of Enterprise JavaBeans 3.0

- Entity
  - Used to model the persistence part of an application
- Session Bean
  - Invoked by a client to perform a specific business operation
- Message-driven bean
  - Triggered by messages sent to a messaging server, which enables sending asynchronous messages between system components



# Enterprise JavaBeans 3.0 Component Architecture



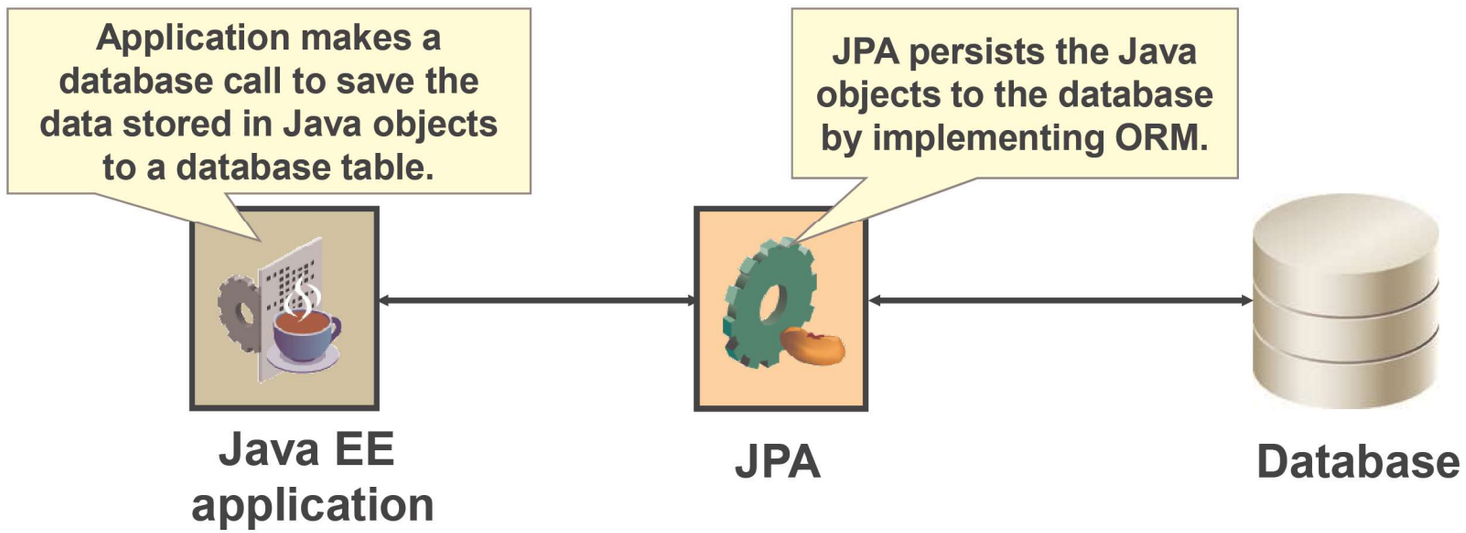
# Java Persistence API (JPA)

The Java Persistence API (JPA) deals with:

- The way relational data is mapped to Java objects (“persistent entities”)
- The way these objects are stored in a relational database so that they can be accessed at a later time
- The continued existence of an entity’s state even after the application that uses it, expires

JPA defines a standard for:

- The ORM configuration metadata
- The EntityManager API
- The JPQL



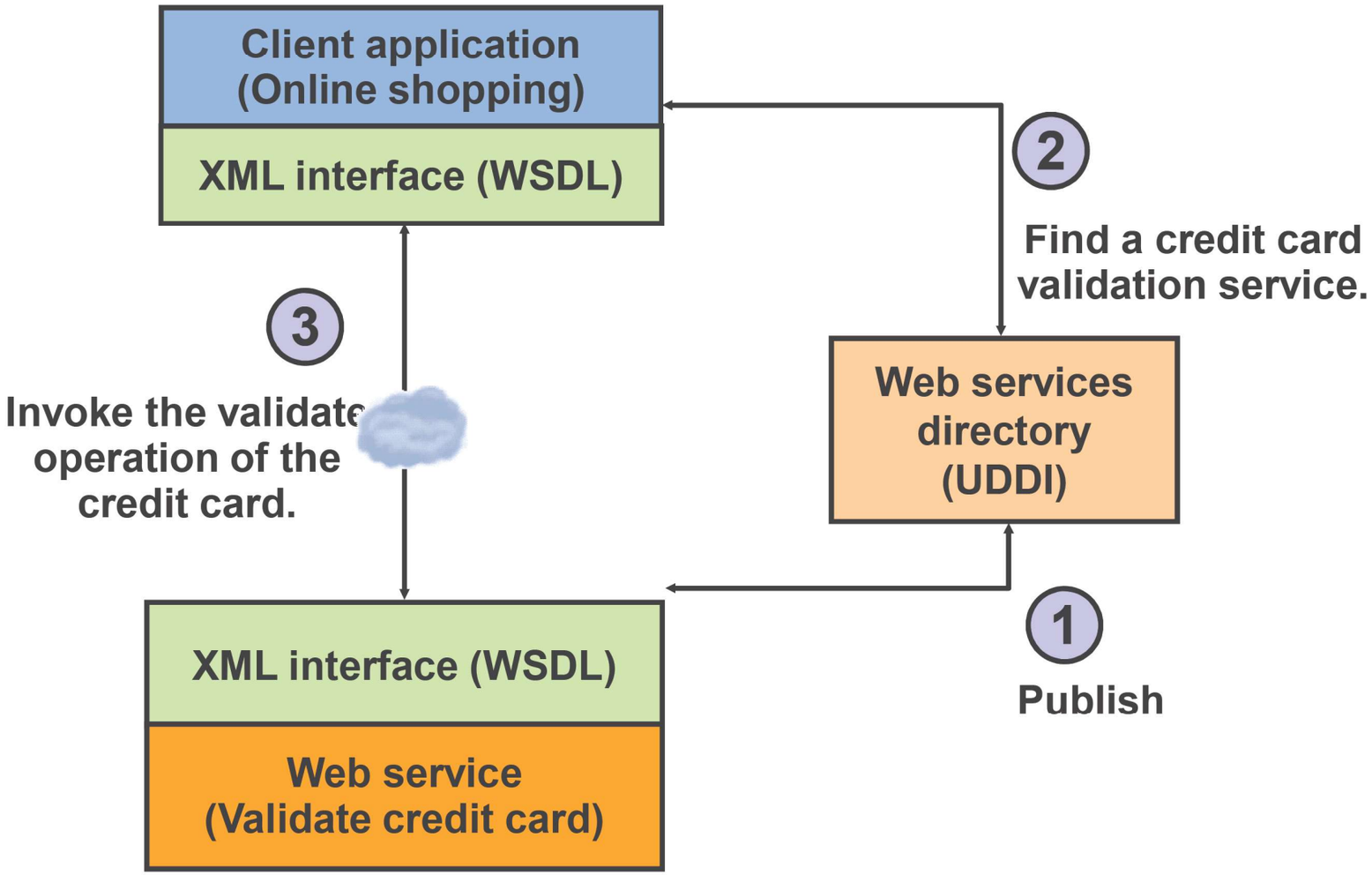
Enterprise Java Beans can be categorized into:

1. Stateless Session Bean
2. Entity
3. Stateful Java Bean
4. Message-Driven Bean

# Web Services

- Web services use open XML-based standards and transport protocols to exchange data with calling clients.
- The Java EE platform provides the XML APIs and tools needed to design, develop, test, and deploy Web services and clients that fully interoperate with other Web services and clients running on Java-based or non-Java-based platforms.

# Java EE Web Services Architecture



# Client-Tier Components

- A Web browser:
  - Is used for a Web-based Java EE application
  - Presents static or dynamic Web pages from Web-tier components
- An application client:
  - Is used for a non-browser-based Java EE application
  - Executes on the client machine
  - Can contain a graphical or command-line interface
  - Accesses middle-tier services
  - Requires installation on the client machine

# Web Applications

Web applications come in many forms:

- Simple static web pages
- Single page applications
- Animated pages with JavaScript
- Interactive pages
- HTML5 games
- Forms to request user data

Examples of a web application:

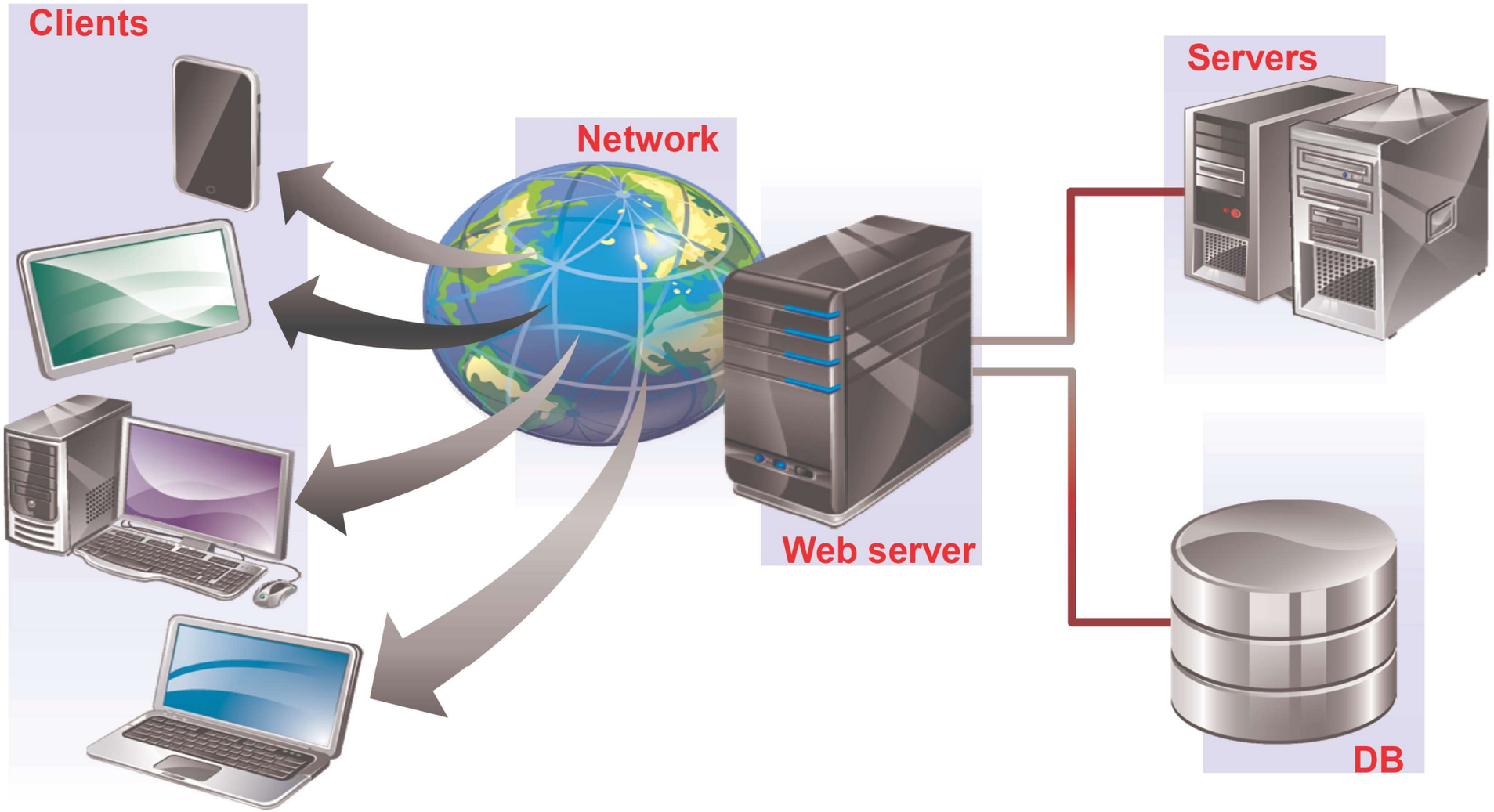
- Weather web page
- The Oracle web portal

# Web Servers

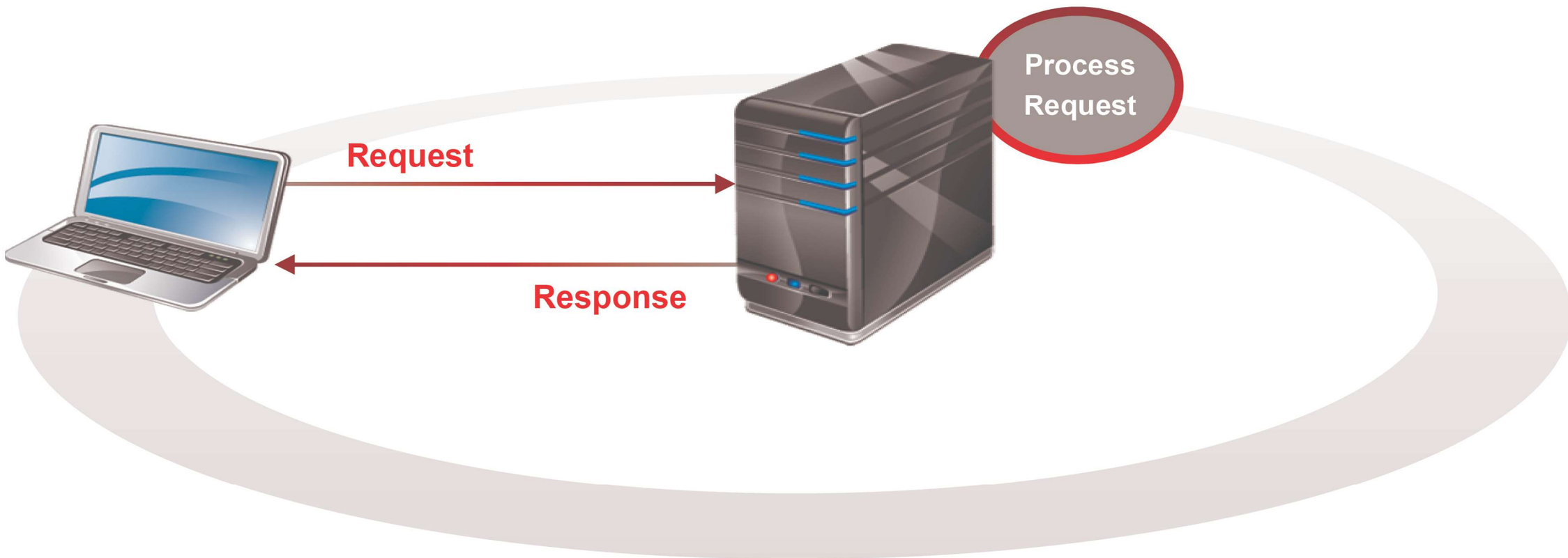
Web applications are usually stored in web servers. They:

- Handle web requests
- Store application files
- Provide access to resources

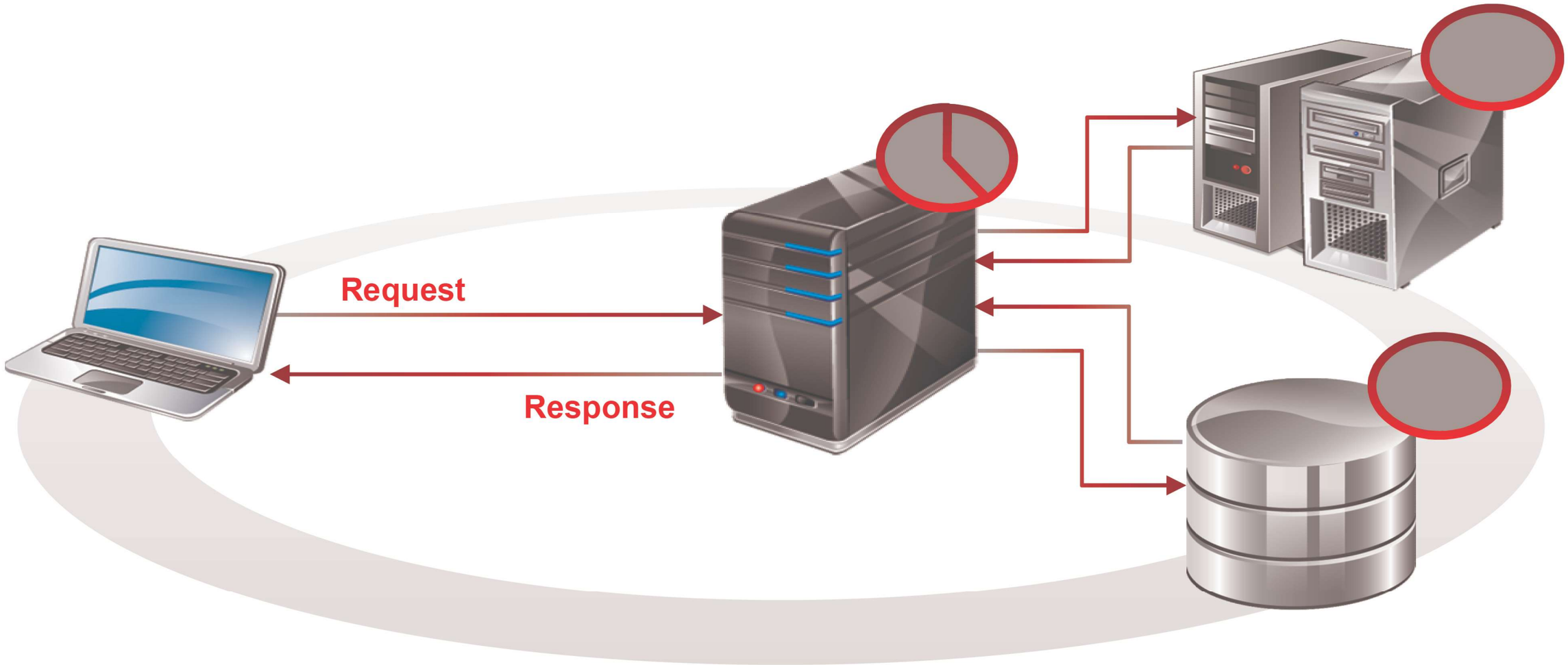
# Web Architecture



# How Web Servers Work



# How Web Servers Work

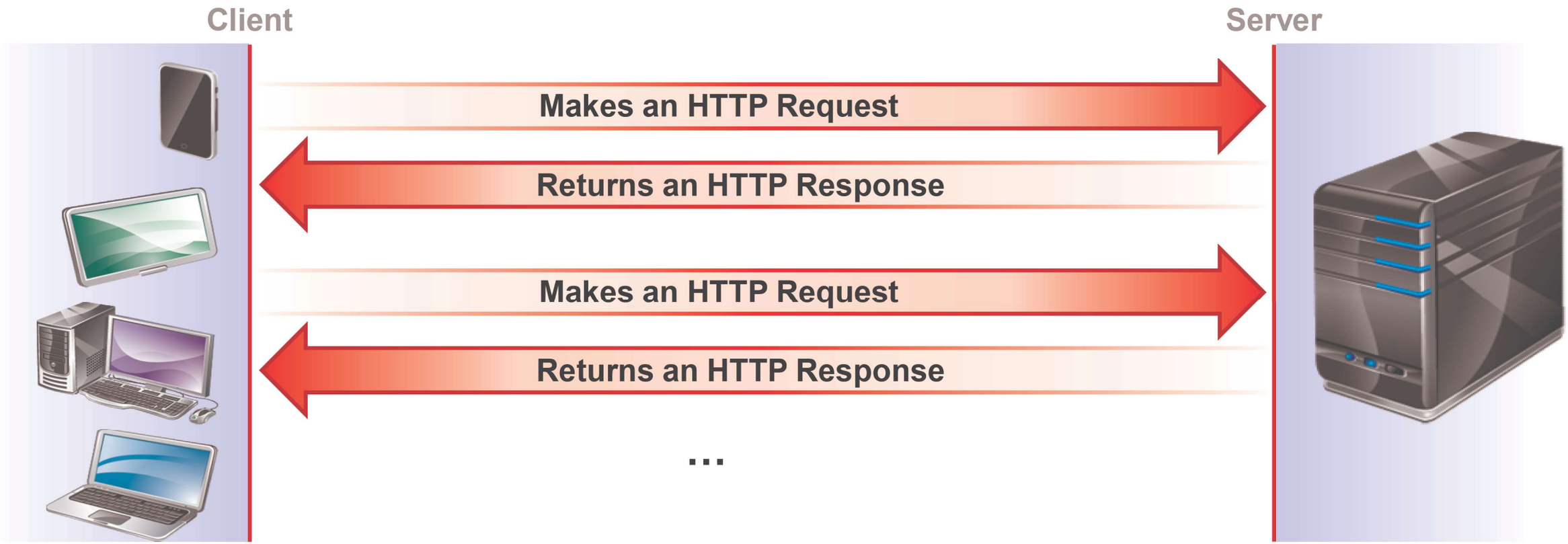


# What Is a Client?

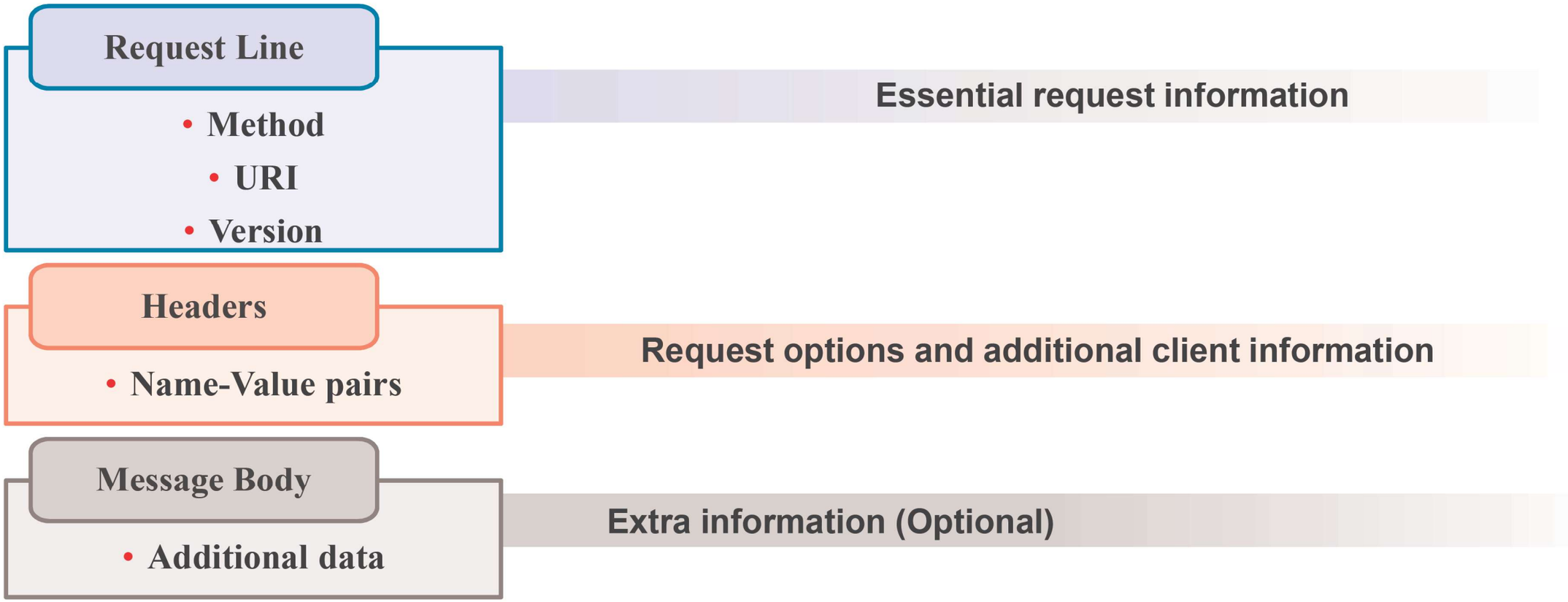
A client is an application that runs on a machine that can make HTTP requests and interpret HTTP responses.

- Web browsers
  - Firefox, Chrome, Internet Explorer, and Safari
- Other applications
  - Weather app on mobiles
  - News readers

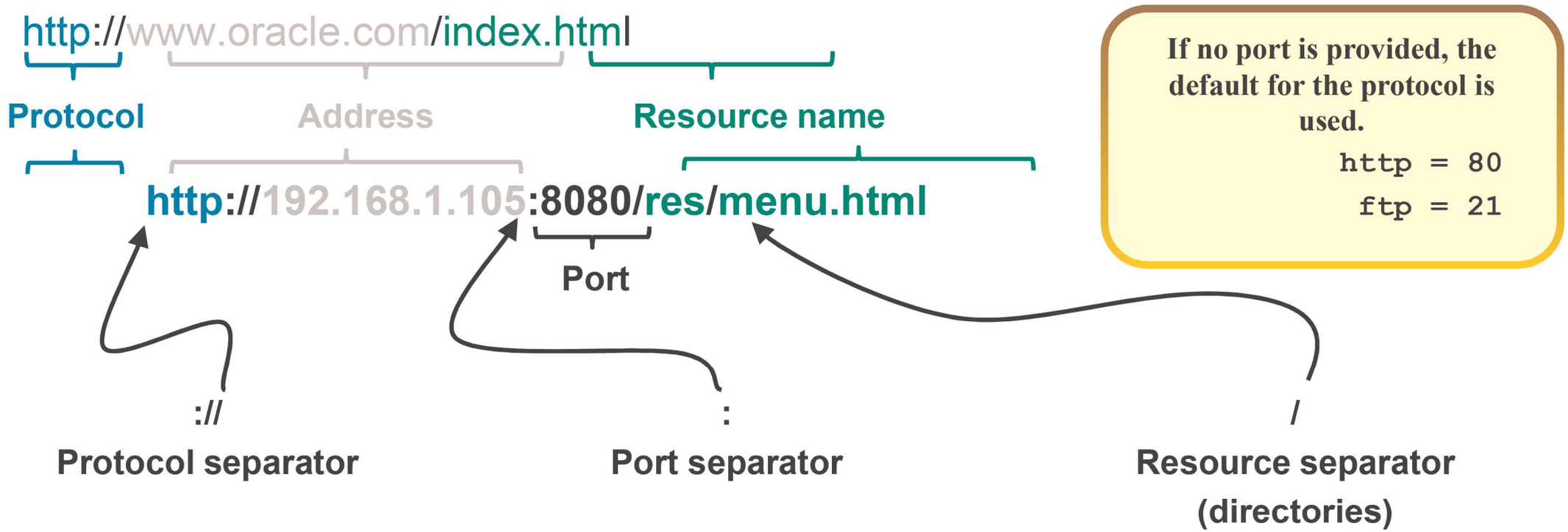
Clients communicate with the server by using the HTTP protocol.



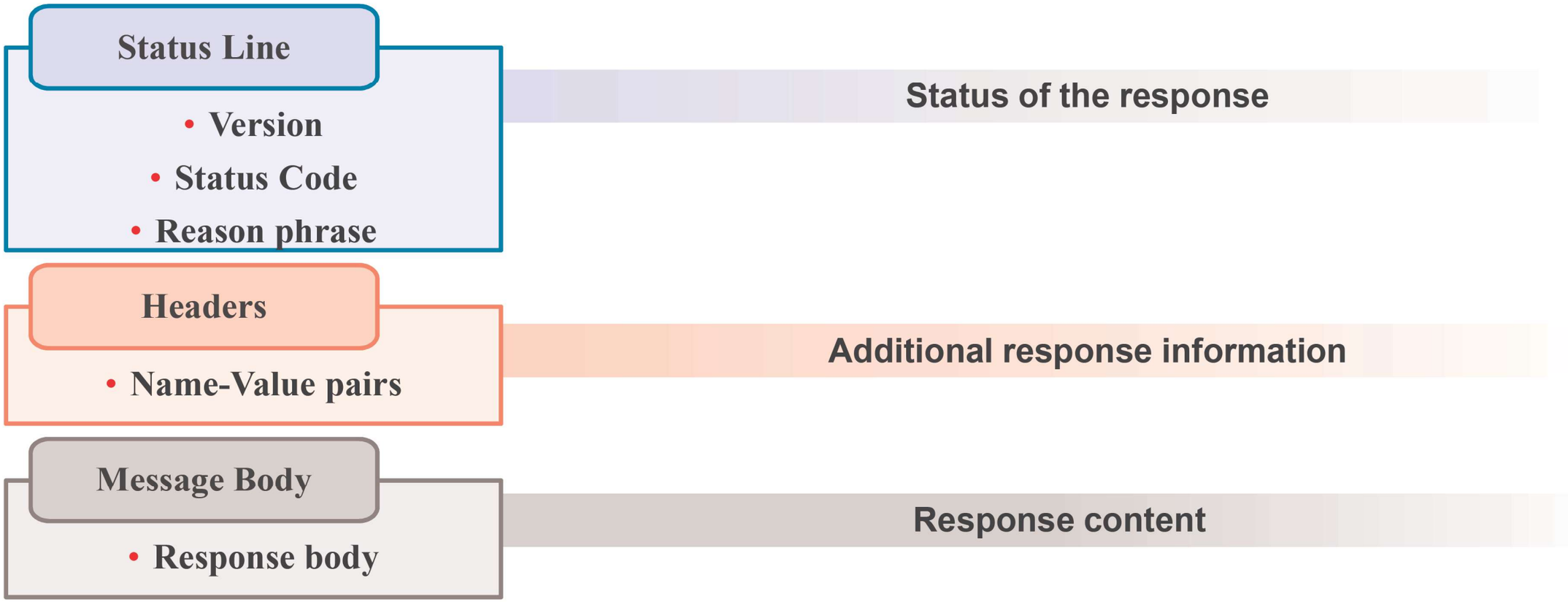
# HTTP Request



# HTTP Request: URL



# HTTP Response



# Response Bodies

A response body contains the content of a resource, including:

- Documents
- Images
- Audio
- Video
- JavaScript files

**The client (web browser) usually knows how to handle or display the contents of the response.**

**The JavaScript code in web applications is run by the web browser on the client machine.**

# Web Applications

## Technical Definition

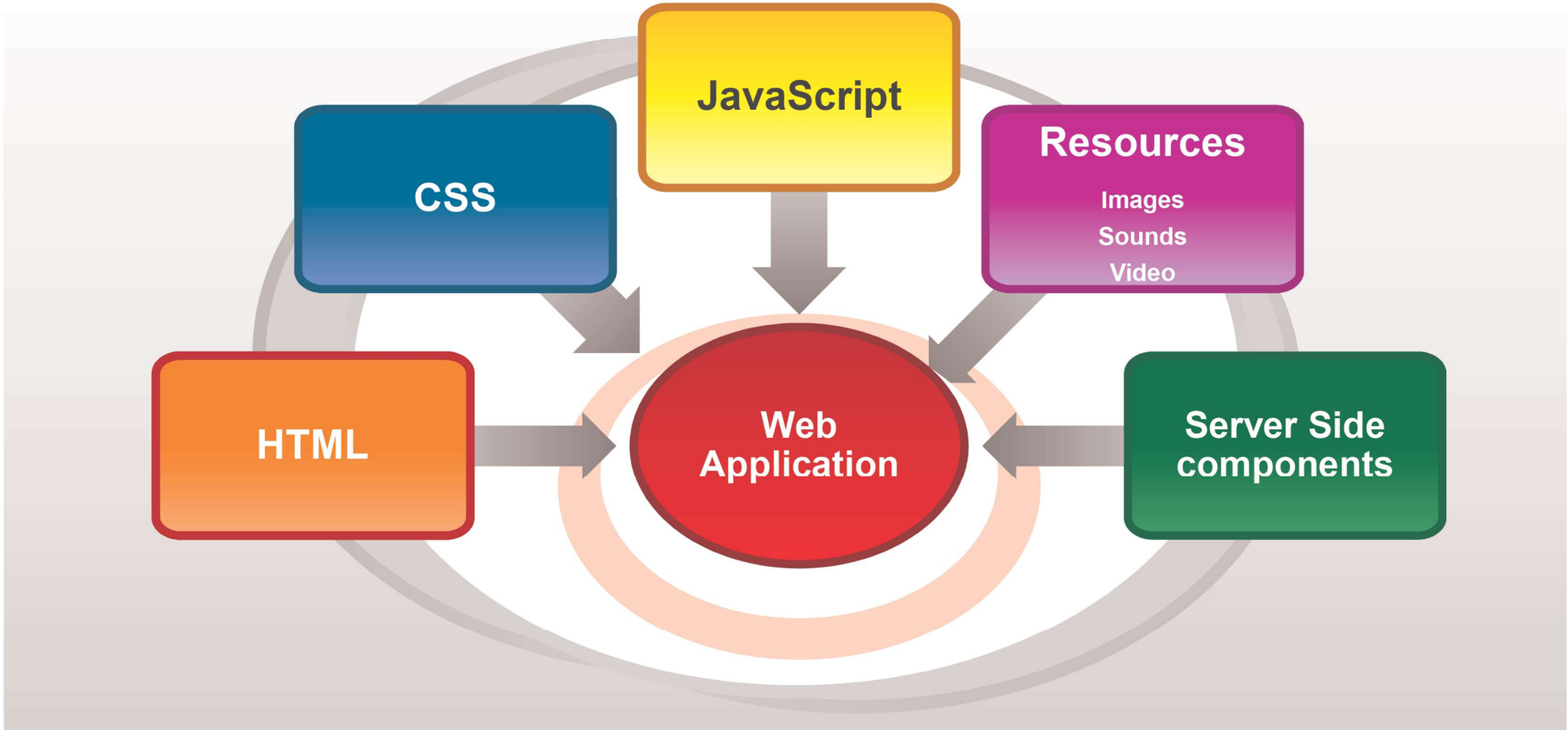
- A collection of files that are stored locally or in a web server that can run on a web browser

## Conceptual Definition

An application that runs on a web browser that:

- Provides the user a solution to a problem
- Is self-contained and focused
- Has a rich user interface
- Uses the capabilities of the user's device

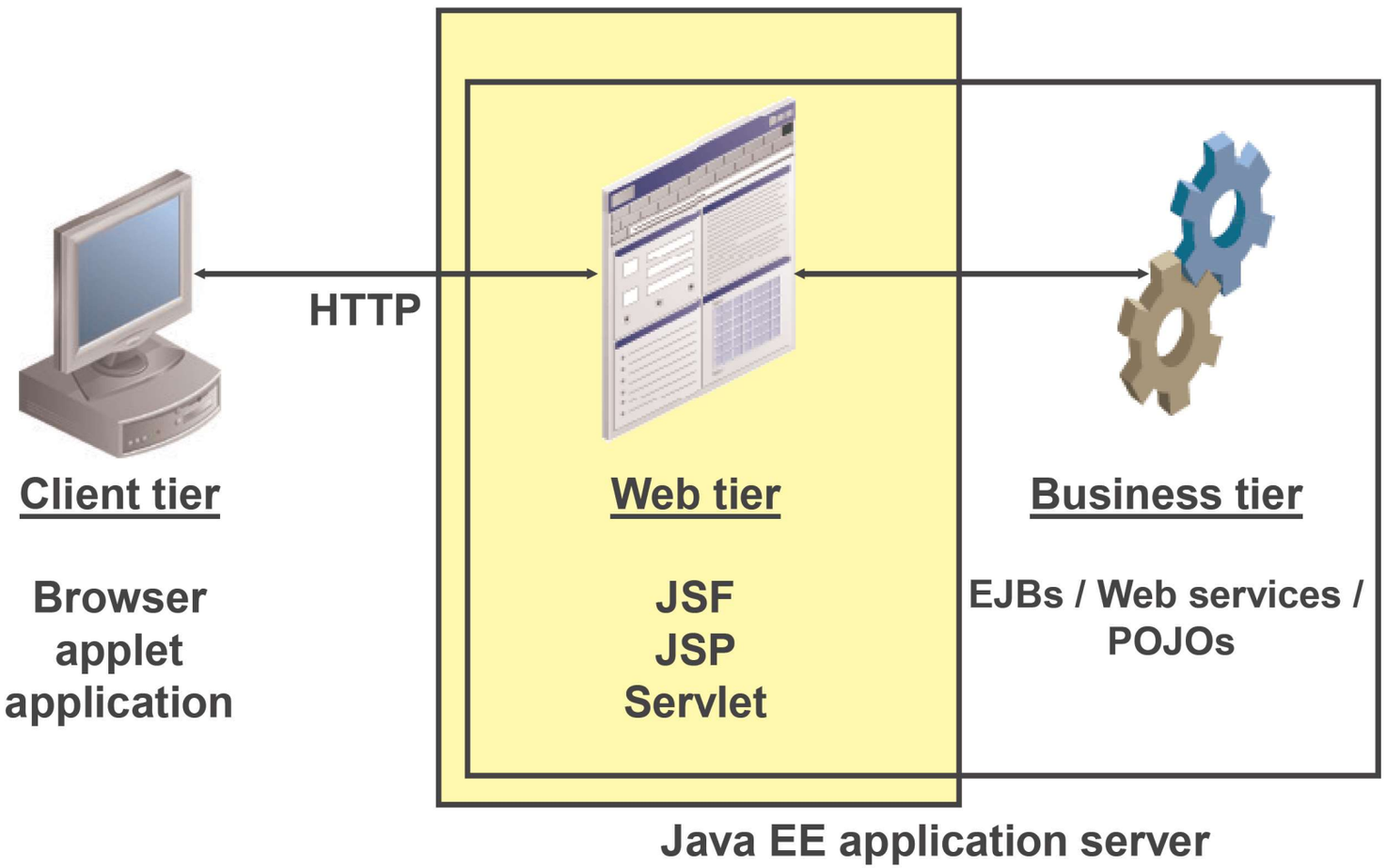
# Web Applications



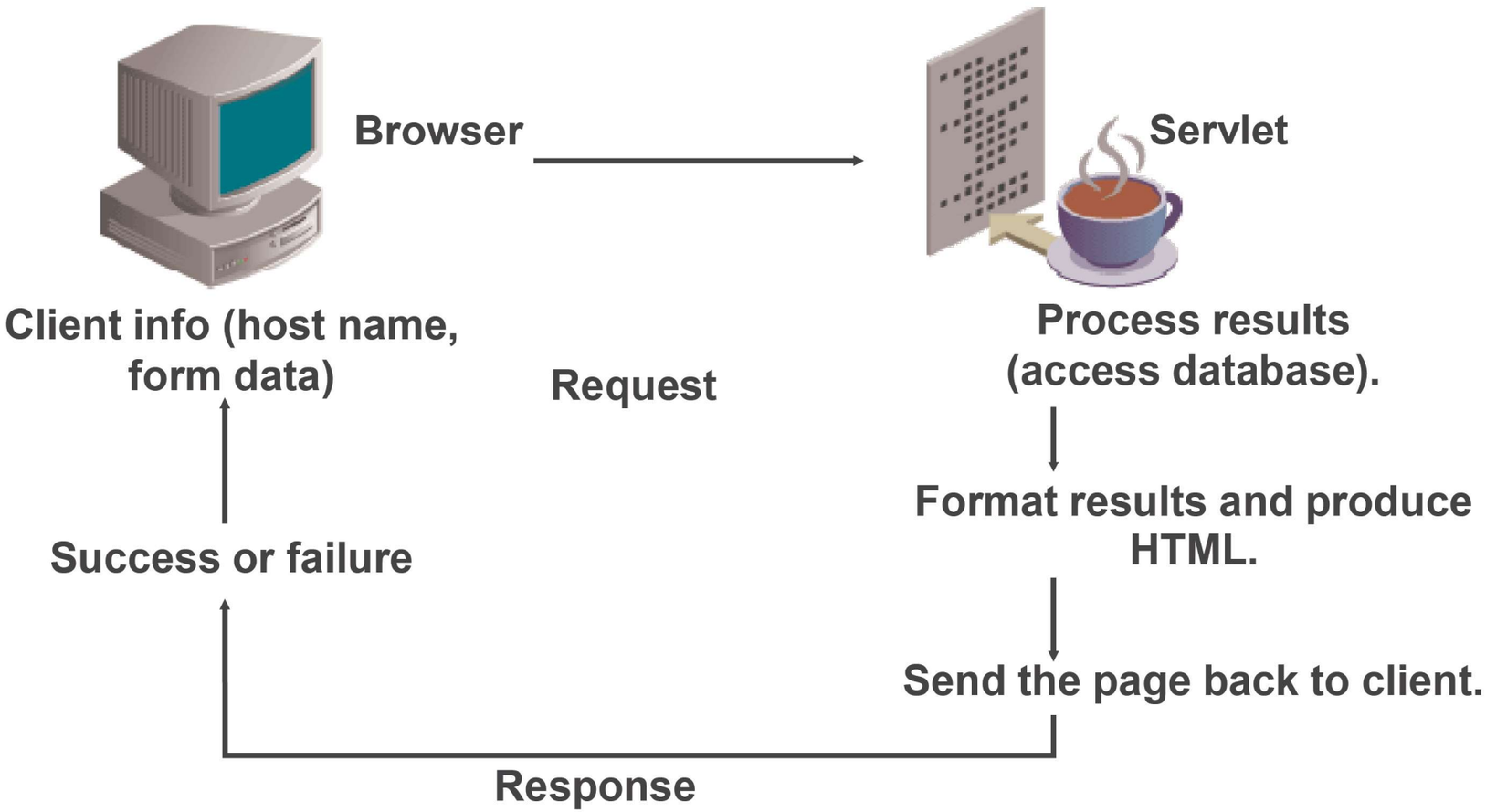
# Java EE Web-Tier Components

- A Web tier may consist of:
  - Java servlets
  - JSPs
  - JSFs
- Servlets and JSPs:
  - Work on a request-response model
  - Generate HTML dynamically
  - Access the business-tier components
  - Handle user-centric events, such as an HREF link or form submission
  - Usually generate visual interfaces, such as a Web page

# Java EE Web Application Architecture



# What Is a Servlet?



# What Is a JavaServer Page?

A JSP:

- Is a text-based document that includes:
  - HTML
  - JSP tags
  - Java code (including calls to JavaBeans and servlets)
- Cleanly separates content creation from presentation logic
- Focuses on rapid development and easy modification of the user interface
- Focuses on presentation

# What Is JavaServer Faces?

A JSF:

- Is a server-side component framework for Web applications
- Implements the Model-View-Controller (MVC) framework
- Provides separation of navigational and data flow
- Is built for rapid application development (RAD) style development

# Web-Tier Components: Summary

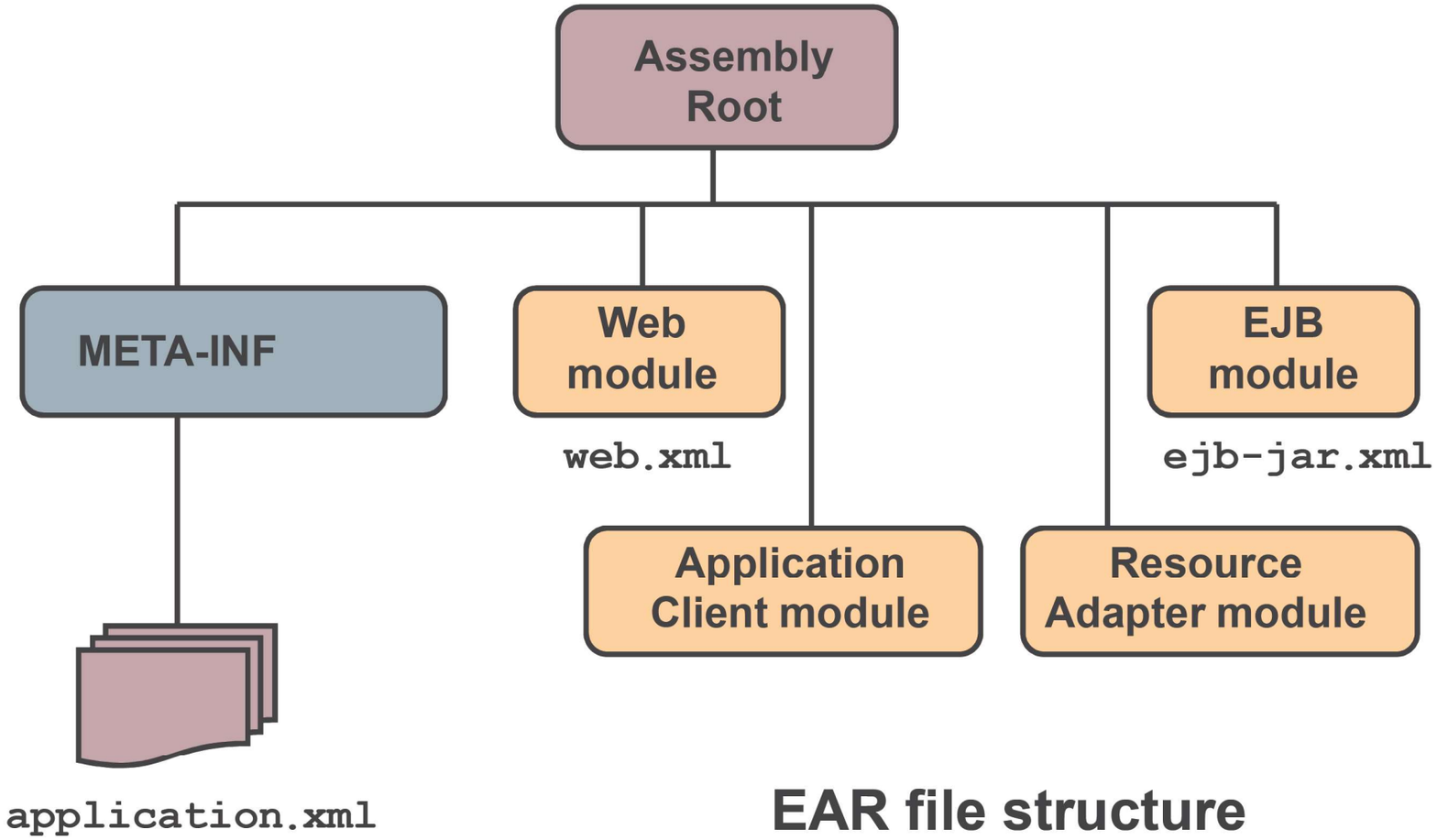
- Web-tier components generate dynamic content.
- Servlets:
  - Extend Web server functionality
  - Are best used for controlling the application's flow
- JSPs:
  - Combine HTML (or other markup) and Java
  - Are best used for presentation logic
- JSFs:
  - Are component based
  - Implement MVC

# Java EE Applications

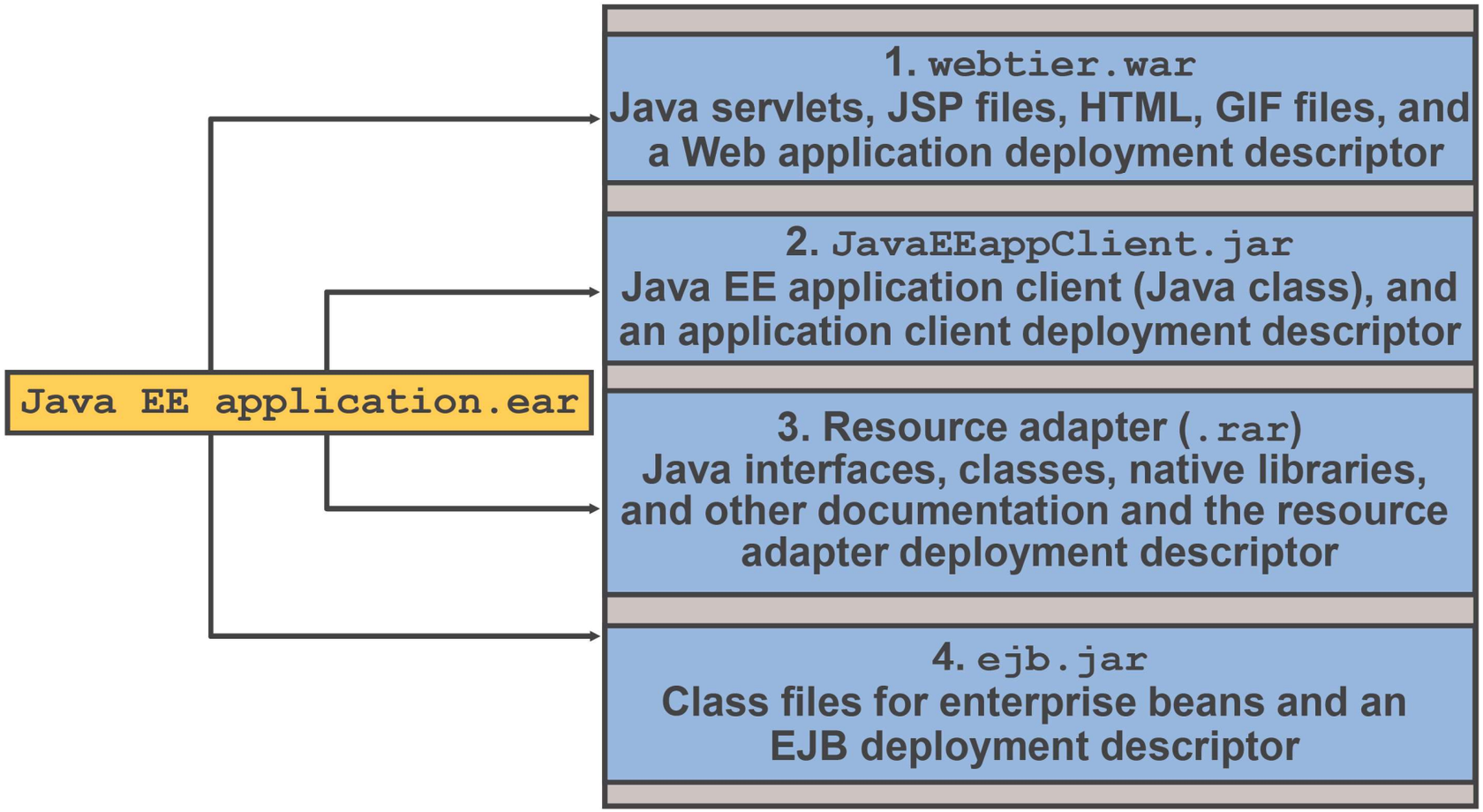
Java EE applications consist of Java EE components and are deployed in the form of modules:

- Web modules contain the user interface: HTML, JSP, and servlets.
- EJB modules contain reusable EJB components.
- Client modules provide access to remote application code.
- Packaging information identifies dependencies between modules.

# Packaging Java EE Applications



# Packaging Java EE



# Security in Java EE Applications

- Security for Java EE application components is provided by their containers.
- A container provides two types of security:
  - Declarative security through deployment descriptors
  - Programmatic security embedded in the application
- Annotations are used to specify information about security within a class file.

# Using Deployment Descriptors for

A deployment descriptor:

- Is an XML document with an `.xml` extension
- Is used by application developers to communicate how security is to be set up for the deployed application
- Enables an application's security structure, including roles, access control and authentication requirements to be expressed in a form external to the application
- Is read at run time by the Java EE server which then acts upon the application, module, or component, accordingly

Identify the components of Java EE Web Tier.

1. Stateless Session Bean
2. Servlet
3. MDB
4. JSP
5. JPA

# Java EE Server

The Java EE server provides:

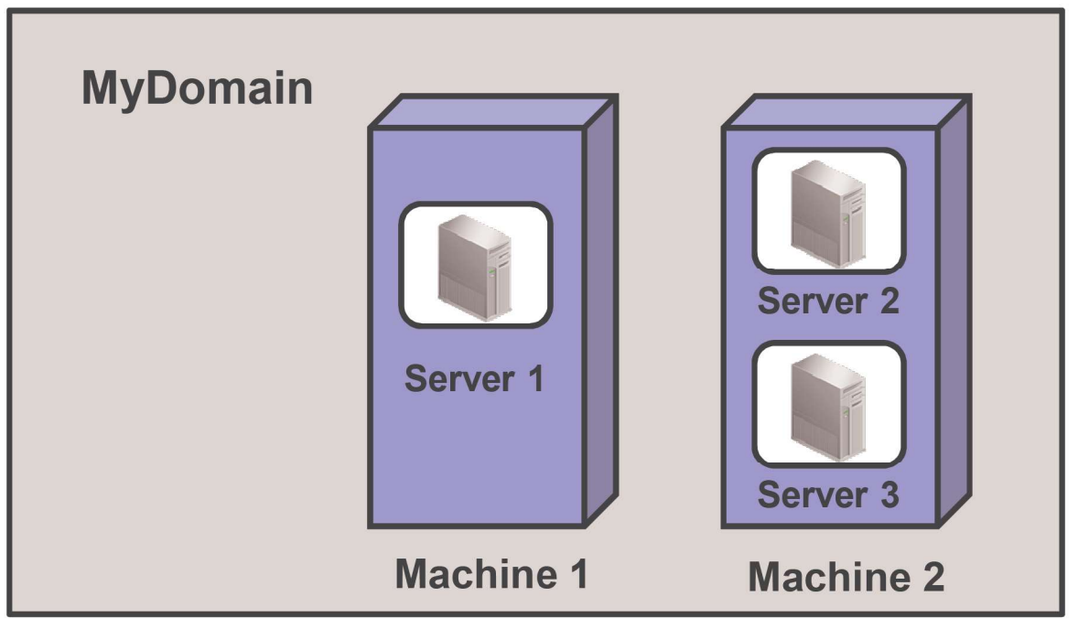
- Containers for each component type of a Java EE application
- System-level services to components:
  - Naming and directory services (JNDI)
  - Security services for Web components and EJBs (JAAS)
  - Transaction architecture (JTA)
  - Remote client connectivity:
    - » Enterprise beans (RMI/IIOP)
    - » Servlet/JSP (HTTP, HTTPS, FTP)

- Is a Java EE server implementation
- Key features:
  - Implements Java EE 1.7 Specification
  - Runs on standard JVM
  - Provides high performance and scalability
  - Is productive for developers to use
  - Is simple to manage and deploy
  - Provides clustering for high availability and failover

# WebLogic Server Domain

A domain:

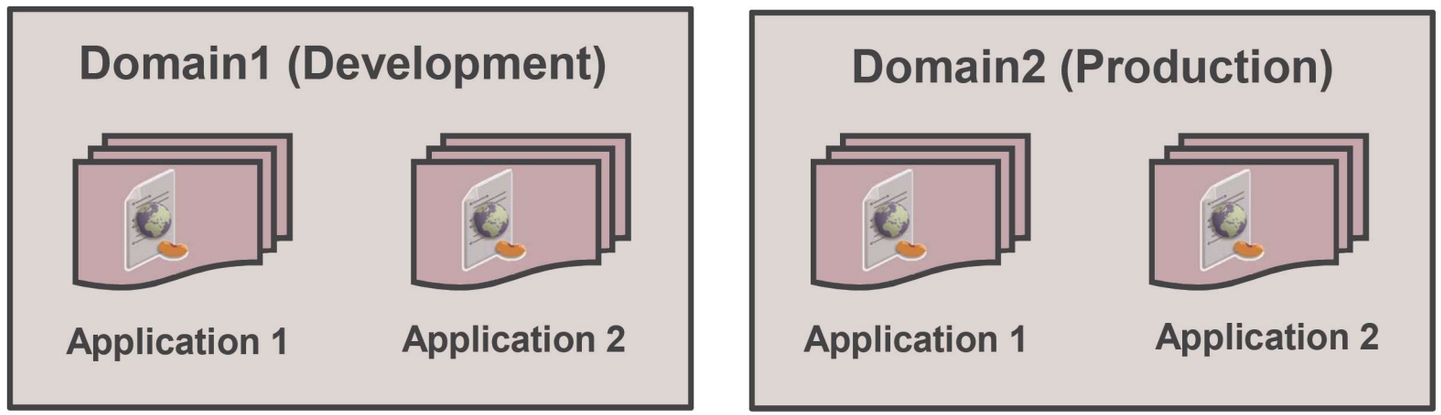
- Is an administrative unit or boundary.
- Allows for a single point of administration for a collection of servers



# WebLogic Server Domain

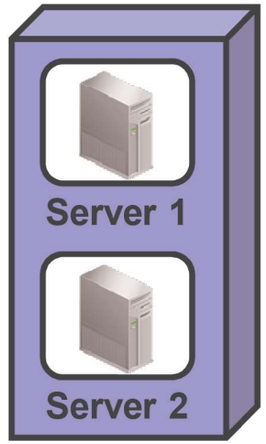
WebLogic Server Domains can be used to separate:

- Development, test, and production applications
- Administration and operational responsibilities
- Organizational or business divisions



# WebLogic Server Servers

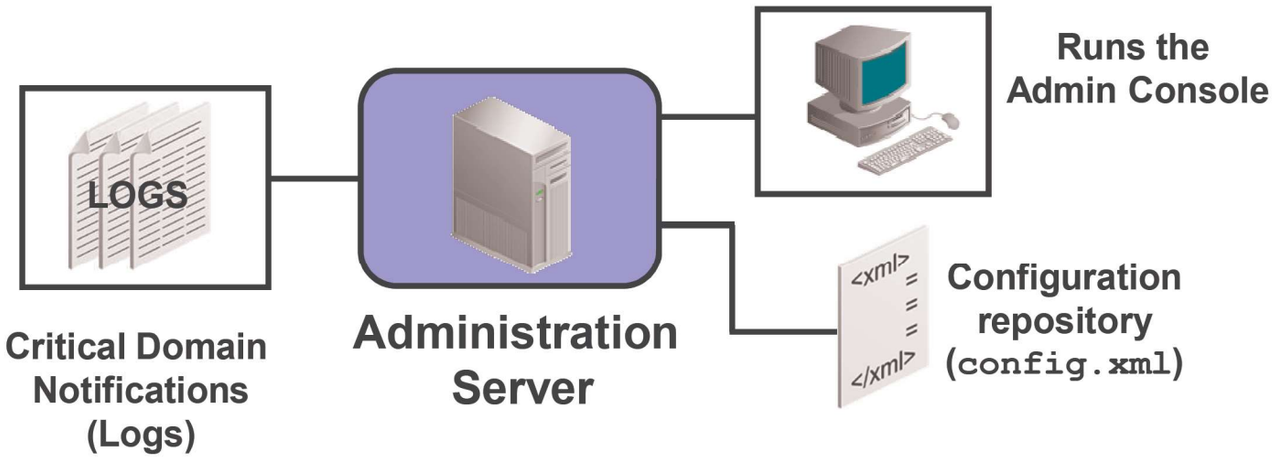
- A server is a WebLogic Server instance executing within a JVM.
- A server:
  - Can be associated with at most one WLS machine
  - Has a dedicated amount of RAM
  - Is multithreaded



# Administration Server

An Administration Server is:

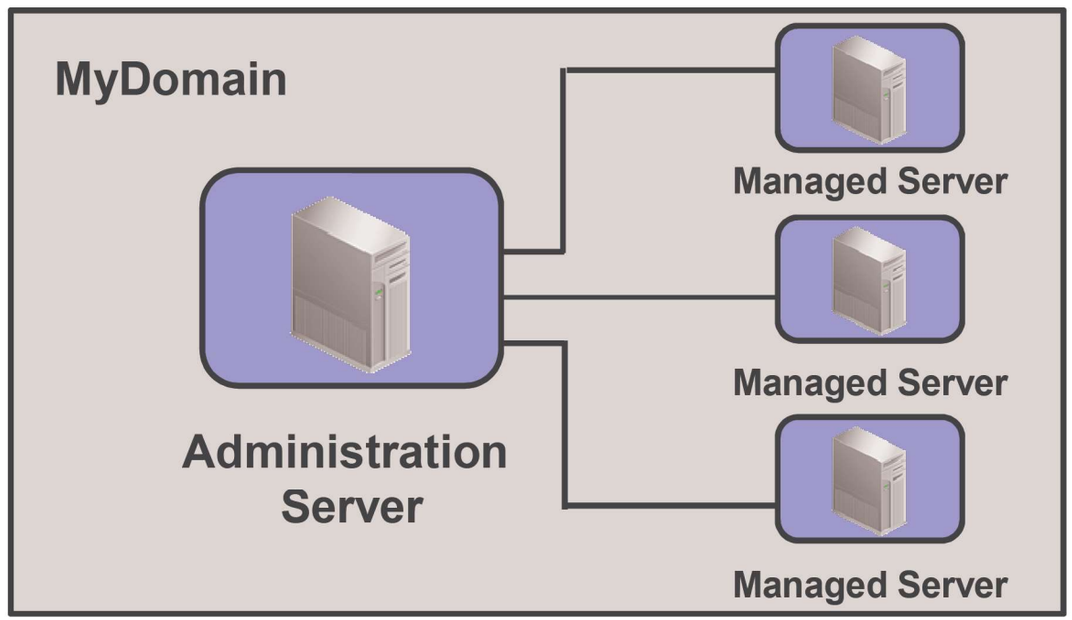
- The central point of control for a domain
- The keeper of the XML configuration repository
- The central source for logging information



# Managed Server

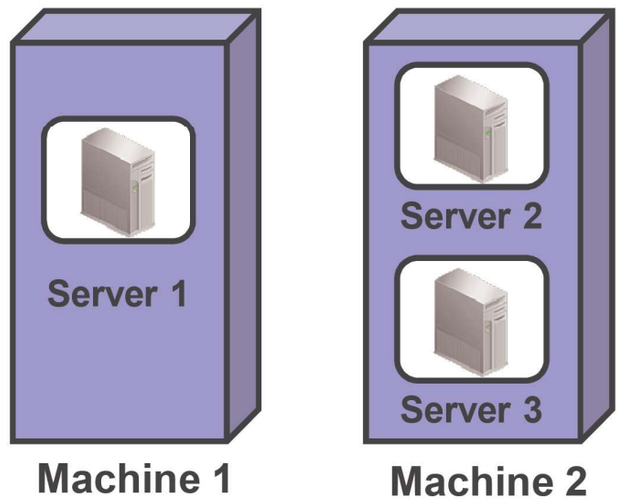
A Managed Server:

- Is an instance of WebLogic Server
- Loads its configuration remotely from an Administration Server



# WebLogic Server Machines

- A machine:
  - Represents the physical piece of hardware that a server resides on
  - Can be a UNIX (Solaris, HP-UX, AIX, Linux) or non-UNIX type (NT, OS-390, Mainframe)
- One or more server instances can reside on a single machine.



# IDE and Java EE

IDE provides:

- Integrated development, deployment, and testing support for Web-tier and business-tier components
- A Java EE framework for rapid development
  - Application Development Framework (ADF) business components
  - Data tags
- Integration with Struts
- UML modeling
- Visual editors for Web clients
- Easy deployment to WebLogic Server

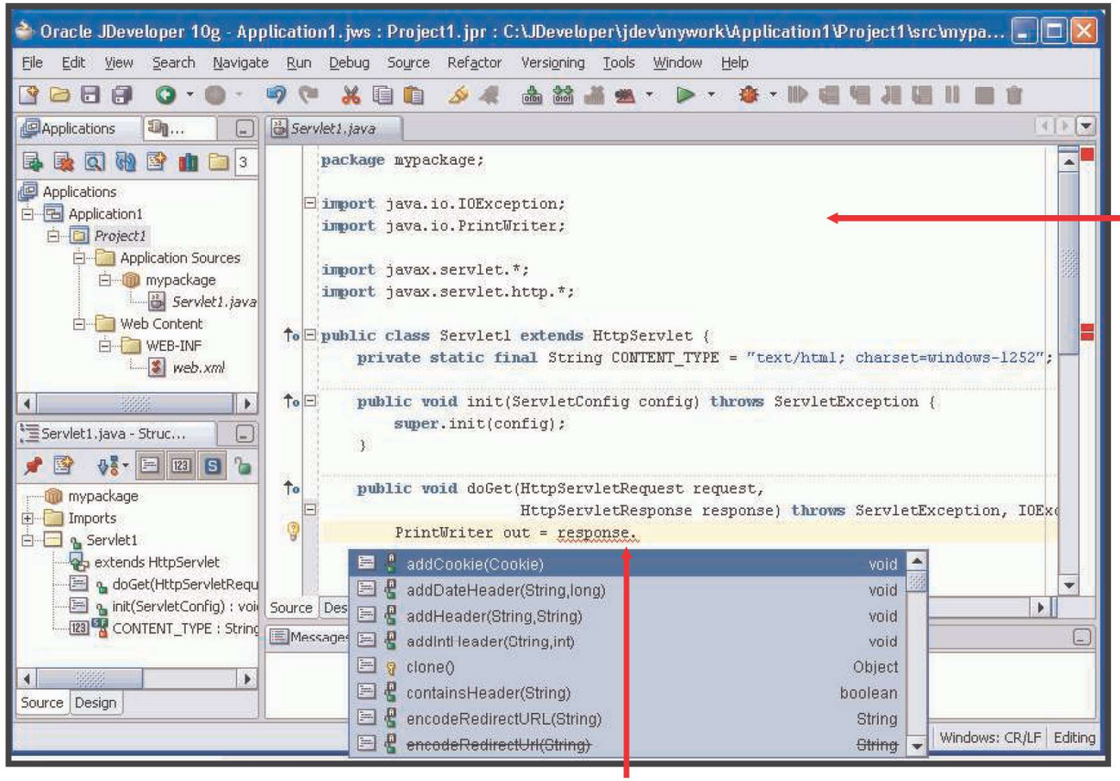
# IDE Environment

Wizards for JSPs, servlets, and EJBs

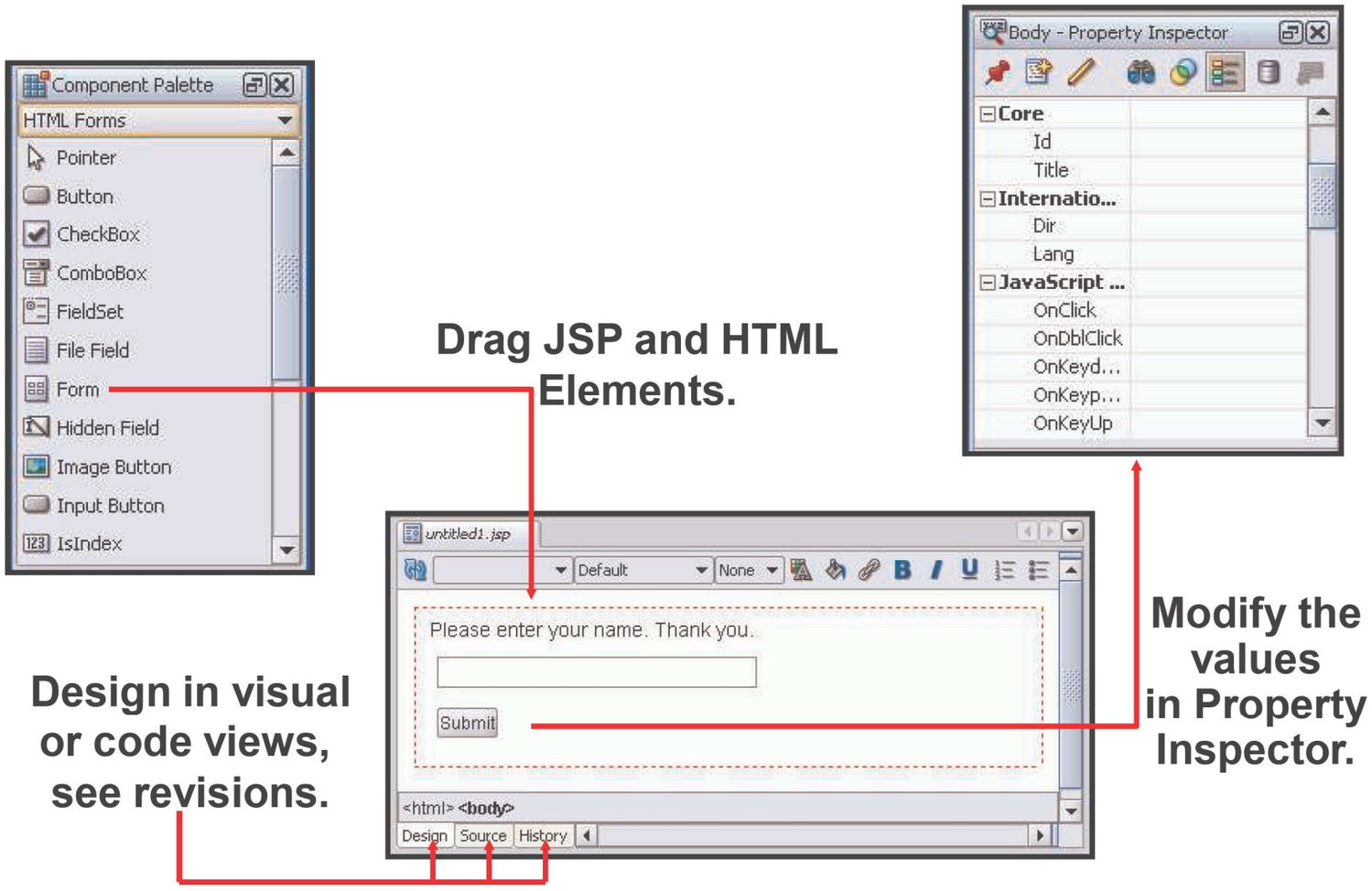
Error checking for HTML and JSPs

EAR, WAR deployment to Java EE server

Customizable Code Editor




Code Insight


Component Palette
HTML Forms
Pointer
Button
CheckBox
ComboBox
FieldSet
File Field
Form
Hidden Field
Image Button
Input Button
IsIndex
Drag JSP and HTML Elements.
Body - Property Inspector
Core
Id
Title
Internatio...
Dir
Lang
JavaScript ...
OnClick
OnDblClick
OnKeyd...
OnKeyp...
OnKeyUp
Modify the values in Property Inspector.
untitled1.jsp
Default
None
Please enter your name. Thank you.
Submit
<html> <body>
Design
Source
History
Design in visual or code views, see revisions.

# Summary

In this lesson, you should have learned how to:

- Java EE is a set of Java technologies that supports end-to-end application development
- Components are the foundation of the Java EE architecture
- Business components (EJBs) are server-side components that contain business logic
- Web components (servlets, JSPs, and JSFs) generate dynamic content

# Summary

In this lesson, you should have learned how to:

- Java EE is a set of Java technologies that supports end-to-end application development
- Components are the foundation of the Java EE architecture
- Business components (EJBs) are server-side components that contain business logic
- Web components (servlets, JSPs, and JSFs) generate dynamic content