

Java EE Architecture

Goal

- What is an enterprise Application
- Requirements for an Enterprise Application
- Layered Application design
- Tiered Architecture
- Distributed Computing
- Java EE Framework and its Services
- Java EE in detail

Enterprise Application

- An enterprise means a business organization
- Enterprise applications are those software applications that facilitate various activities in an enterprise
- An enterprise application has the following challenges and complexity:
 - Diversity of information needs
 - Information is created and used in different ways
 - Complexity of business process
 - Involves complex information capture, processing and sharing
 - Diversity of applications
 - Applications are built using different architecture and technologies

Requirements for an Enterprise Application

- **Programming productivity**
 - The applications should be developed and deployed as quickly as possible
- **Reliability and availability**
 - Downtime is fatal to a business
 - The applications must provide guarantee of business transaction
 - Business process should proceed completely and accurately
- **Security**
 - The application must be safe from illegal access and information theft
- **Scalability**
 - Should be able to handle large number of clients
 - Should ensure effective use of system resources
- **Integration**
 - Should be compatible with existing applications

Layers and Tier-s

- **Layers** are the way to logically break code into components and **tiers** are the physical nodes to place the components on.

Layers in application

- An application contains
 - Presentation layer
 - Business layer
 - Data layer

Presentation tier

The top-most level of the application is the user interface. The main function of the interface is to translate tasks and results to something the user can understand.



Logic tier

This layer coordinates the application, processes commands, makes logical decisions and evaluations, and performs calculations. It also moves and processes data between the two surrounding layers.



Data tier

Here information is stored and retrieved from a database or file system. The information is then passed back to the logic tier for processing, and then eventually back to the user.



Presentation Layer

- Presentation layer components implement the functionality required to allow users to interact with the application.
 - **User interface (UI) components.**
 - provide the mechanism for users to interact with the application. They format data and render it for display, and acquire and validate data entered by users.
 - **UI process components.**
 - To help synchronize and orchestrate user interactions, it can be useful to drive the process using separate UI process components. This prevents the process flow and state management logic from being hard-coded into the UI elements themselves, and allows you to reuse the same basic user interaction patterns in other user interfaces.

Business layer

- Business layer components implement the core functionality of the system, and encapsulate the relevant business logic
- **Business Layer Components**
 - **Application façade**
 - **Business components**
 - **Business workflows**
 - **Business entity components**

Data Layer

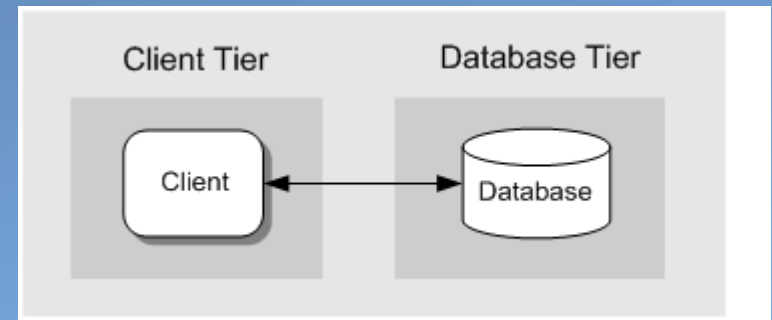
- Data layer components provide access to data that is hosted within the boundaries of the system, and data exposed by other back-end systems
 - **Data access components**
 - **Data helper and utility components**
 - **Service agents**

Tiers

- *Tiers* represent the physical separation of the presentation, business, services, and data functionality of your design across separate computers and systems.
- Common tiered design patterns are (one-tier?) **two-tier**, **three-tier**, and **n-tier**.
 - if 3 layers are present in one system - 1-tier
 - if 3 layers are present in two systems - 2-tier
 - if 3 layers are present in three system - 3-tier
 - if 3 layer are present in n system - n-tier

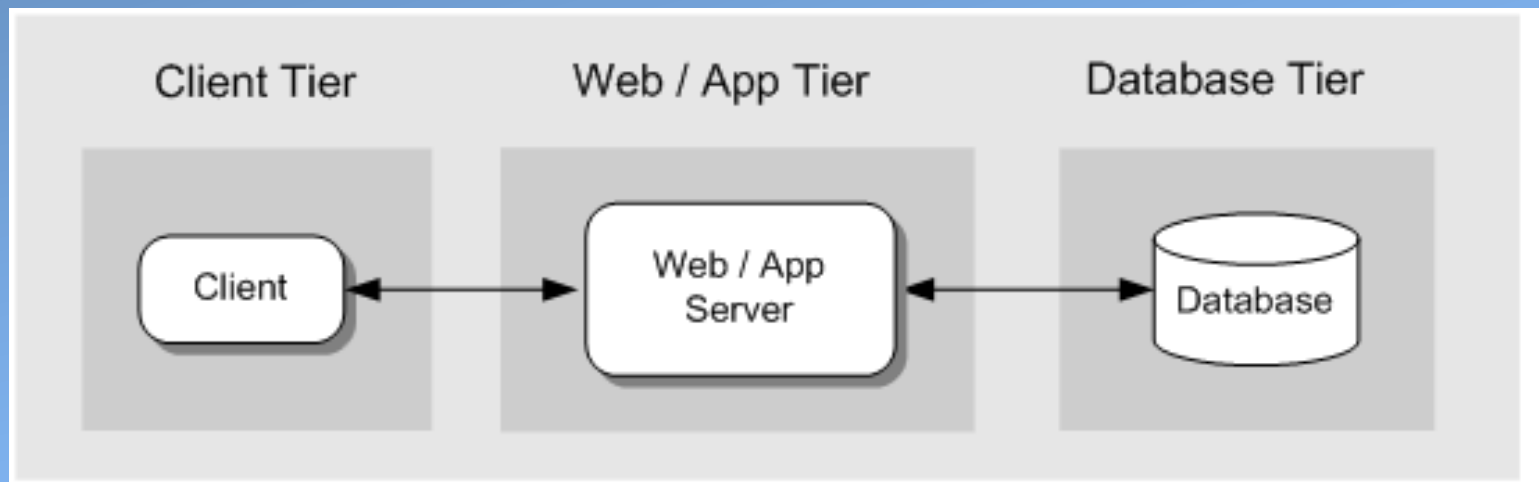
Two-Tier

- The two-tier pattern represents a basic structure with two main components, a client and a server.
- The client and server may exist on the same machine, or may be located on two different machines.
- This tier contains the presentation layer logic and any required business layer logic.
- The Web application communicates with a separate machine that hosts the database tier, which contains the data layer logic.



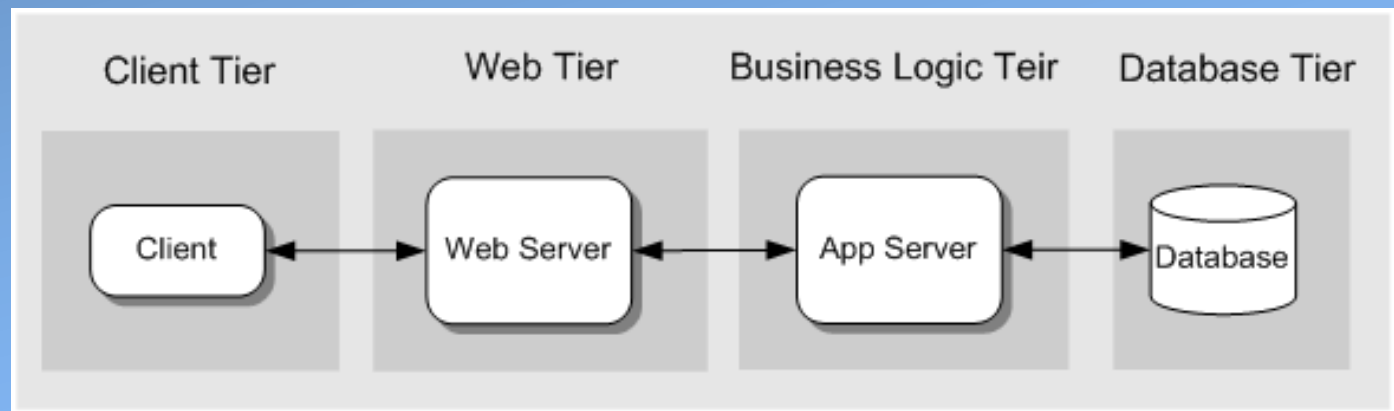
Three-Tier

- In a three-tier design, the client interacts with application software deployed on a separate server, and the application server interacts with a database that is also located on a separate server.
- This is a very common pattern for most Web applications and Web services



N-Tier

- In this scenario, the Web server (which contains the presentation layer logic) is physically separated from the application server that implements the business logic.
- This usually occurs for security reasons, where the Web server is deployed within a perimeter network and accesses the application server located on a different subnet through a firewall.
- It is also common to implement a firewall between the client and the Web tier



How do these tiers relate to
Java EE?

Java EE Platform

- **JavaEE is one of the best solutions so far to meet the requirements of today's Enterprise**
- **JavaEE platform is a distributed application – server environment for n-tier applications.**
- **JavaEE is a Java Environment providing**
 - **A set of Java Extension APIs to build applications**
 - **A runtime infrastructure for hosting and managing applications**

JavaEE Architecture

- JavaEE is a Container centric architecture
- JavaEE platform includes one or more **Containers**

(A container is a runtime to manage components developed according to the API specification, and to provide access to the JavaEE APIs)

JavaEE Architecture

- JavaEE is a Container centric architecture
- JavaEE platform includes one or more **Containers**

(A container is a runtime to manage components developed according to the API specification, and to provide access to the JavaEE APIs)

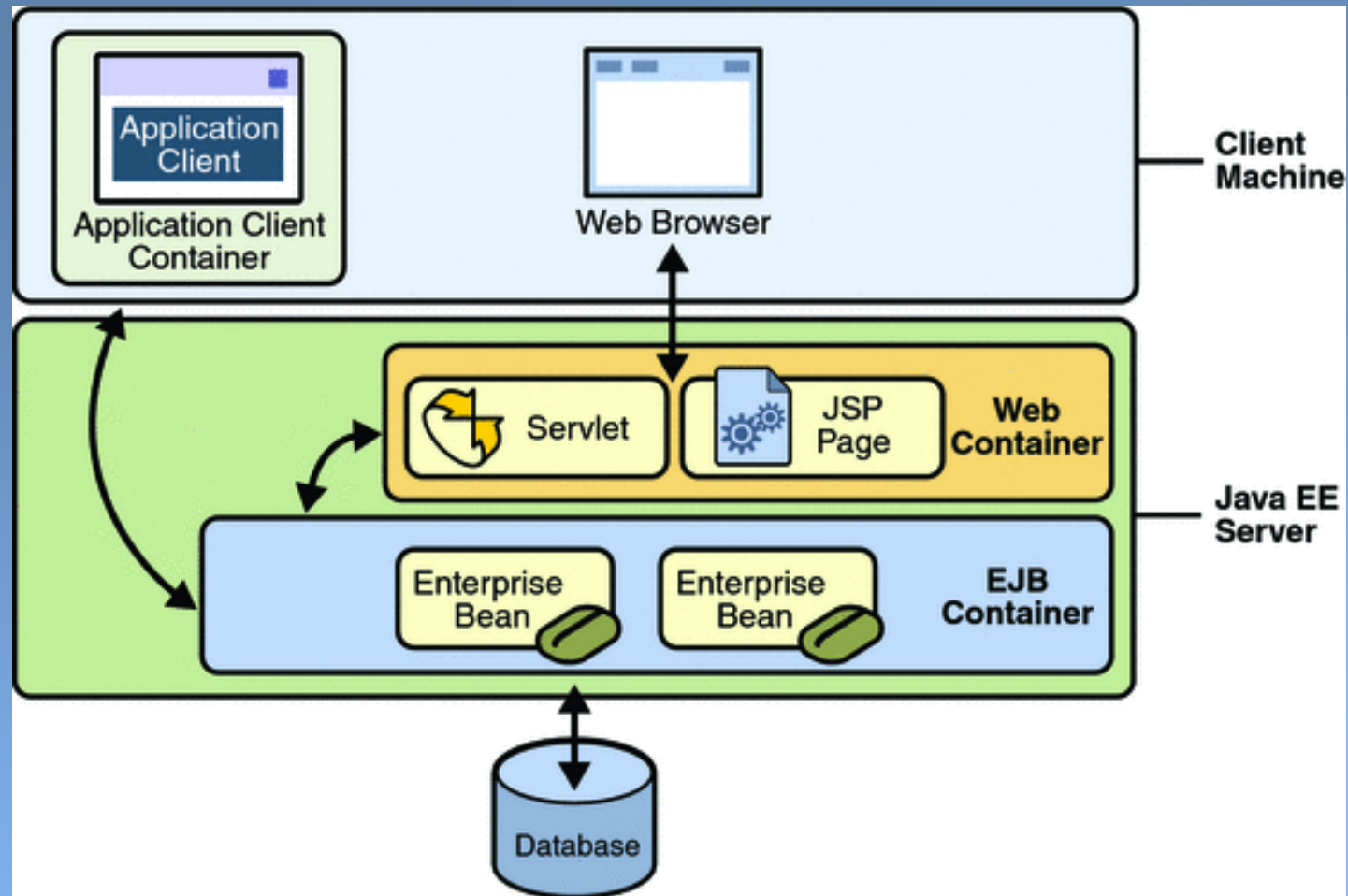
Container Architecture

- Developers are required to provide
 - **Application Components**
 - Servlets, JSP Pages, EJBs etc
 - **Deployment Descriptor**
 - An XML file that describes the application

Container Types

- **Java EE server:** The runtime portion of a Java EE product. A Java EE server provides EJB and web containers.
 - **Enterprise JavaBeans (EJB) container:** Manages the execution of enterprise beans for Java EE applications. Enterprise beans and their container run on the Java EE server.
 - **Web container:** Manages the execution of JSP page and servlet components for Java EE applications. Web components and their container run on the Java EE server.
- **Application client container:** Manages the execution of application client components. Application clients and their container run on the client.
- **Applet container:** Manages the execution of applets. Consists of a web browser and Java Plug-in running on the client together.

Container Types



Container Architecture

- The container Provides
 - **Component Contract**
 - A set of APIs specified by the container that an application component is required to extend or implement
 - **Container Service API**
 - Additional services provided by the container required for applications e.g JDBC, JTS, JNDI, JMS etc.
 - **Declarative Services**
 - Container interposes these services based on the deployment descriptor of the application component e.g Transaction Security etc.
 - **Other Container Services**
 - Life cycle management of components
 - Resource pooling like Object pooling and Connection Pooling
 - Populating JNDI namespace based on the Deployment names associated with EJBs
 - Populating JNDI namespace with objects necessary for utilizing container service APIs

JavaEE Technologies

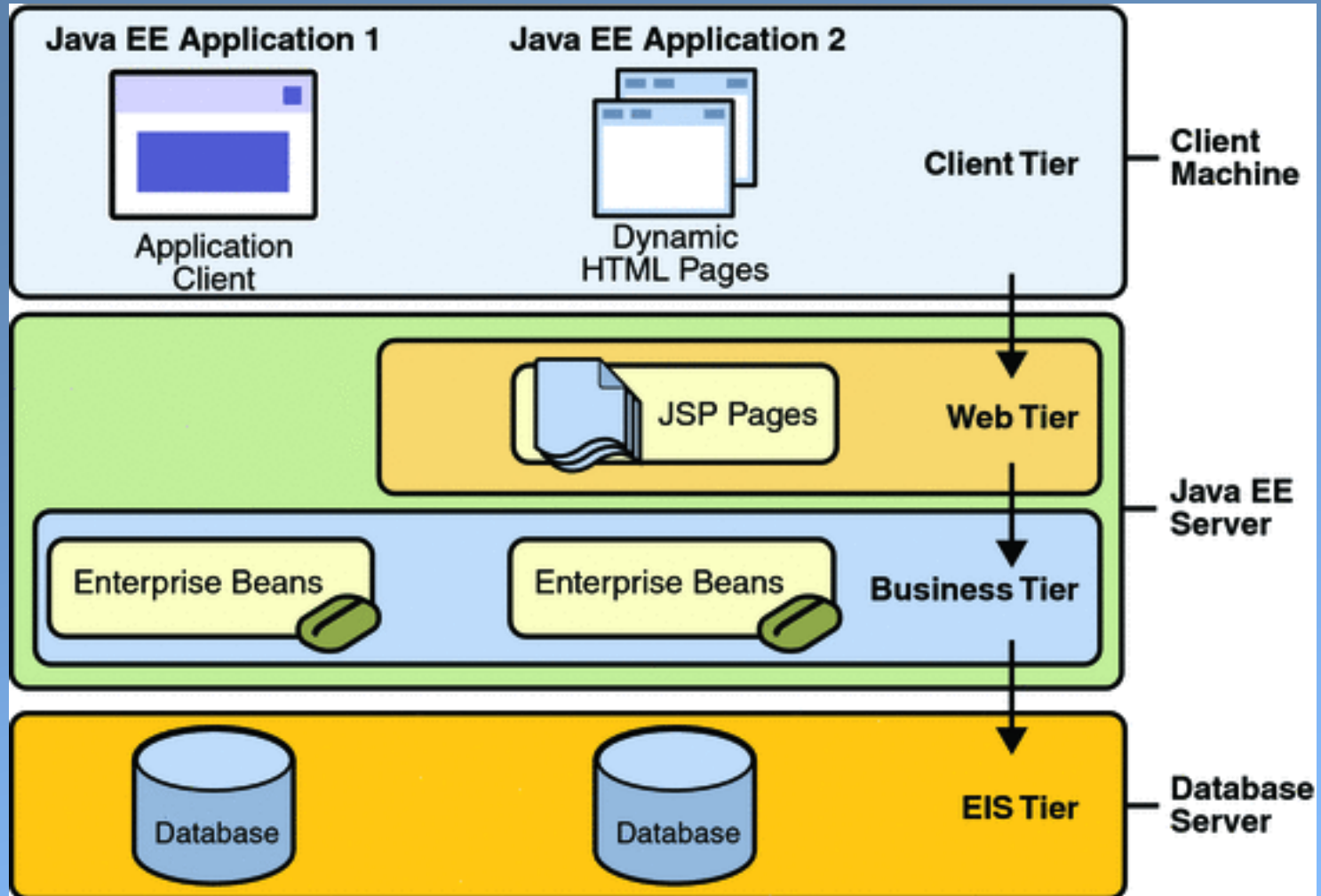
- The Component Technologies
 - Web Components
 - Servlets, JSP Pages
 - Enterprise JavaBean Components
 - Session Beans
 - Entity Beans
 - Message Driven Beans
- Service Technologies
 - JDBC, JTA, JNDI, JMS, JavaMail, JAAS
- Communication Technologies
 - Internet Protocols
 - HTTP, TCP/IP, SSL
 - Remote Object Protocols
 - RMI, RMI-IIOP
 - JavaIDL

- The Java EE platform uses a distributed multitiered application model for enterprise applications.
- Application logic is divided into components according to function, and the various application components that make up a Java EE application are installed on different machines depending on the tier in the multitiered Java EE environment to which the application component belongs.

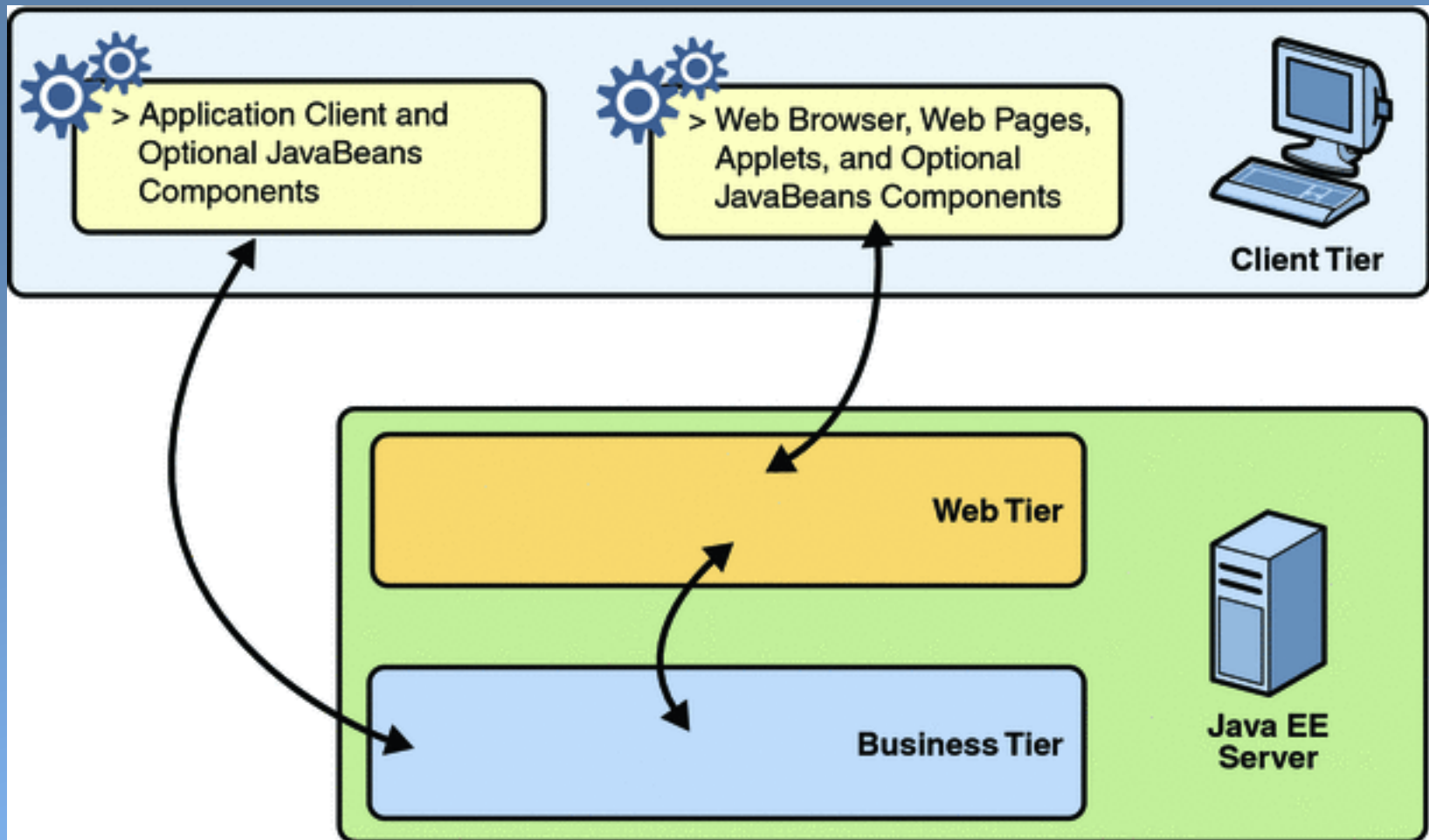
Distributed Multitiered Applications

- In a multitiered java application
 - Client-tier components run on the client machine.
 - Web-tier components run on the Java EE server.
 - Business-tier components run on the Java EE server.
 - Enterprise information system (EIS)-tier software runs on the EIS server.

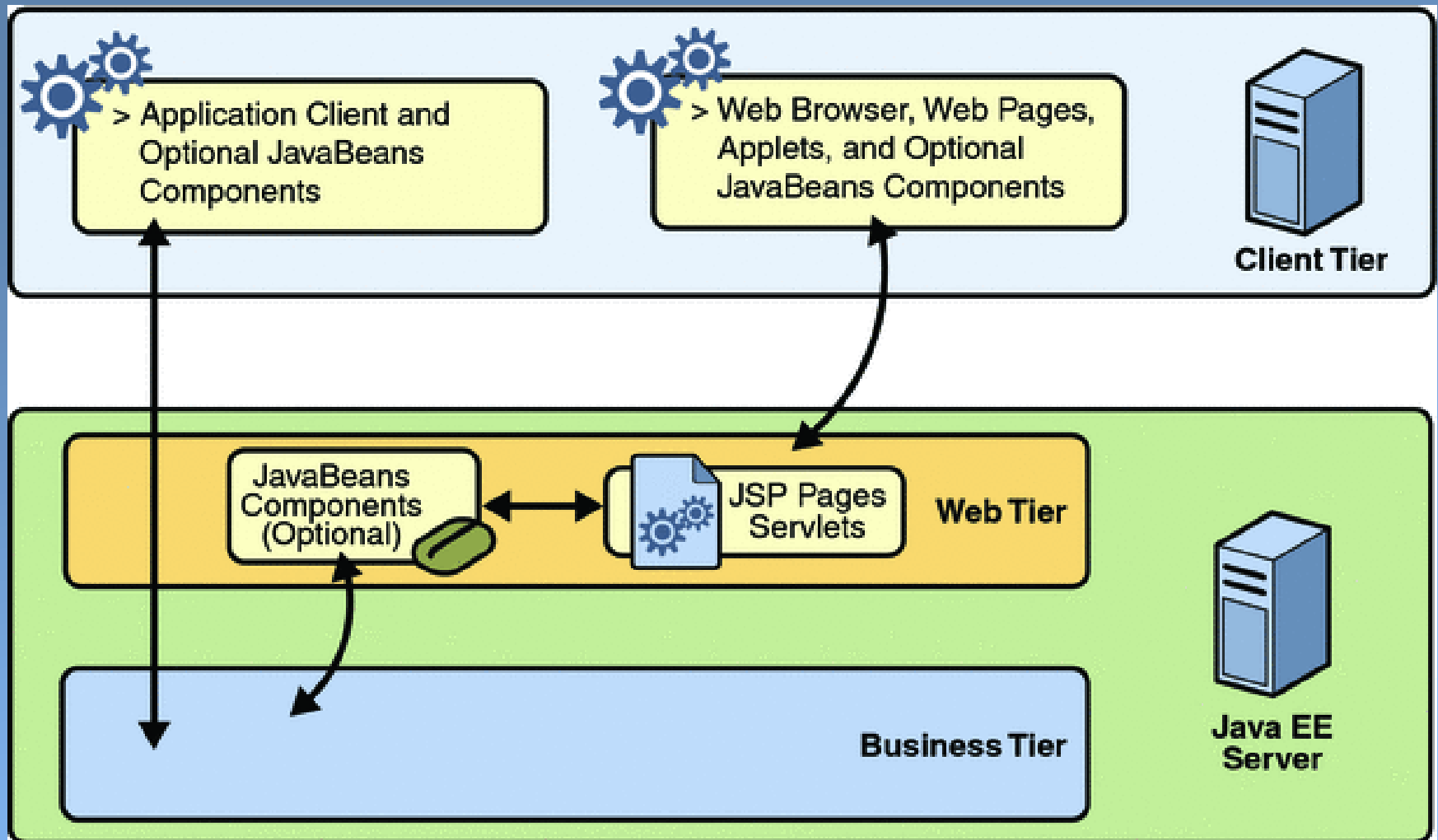
Distributed Multitiered Applications



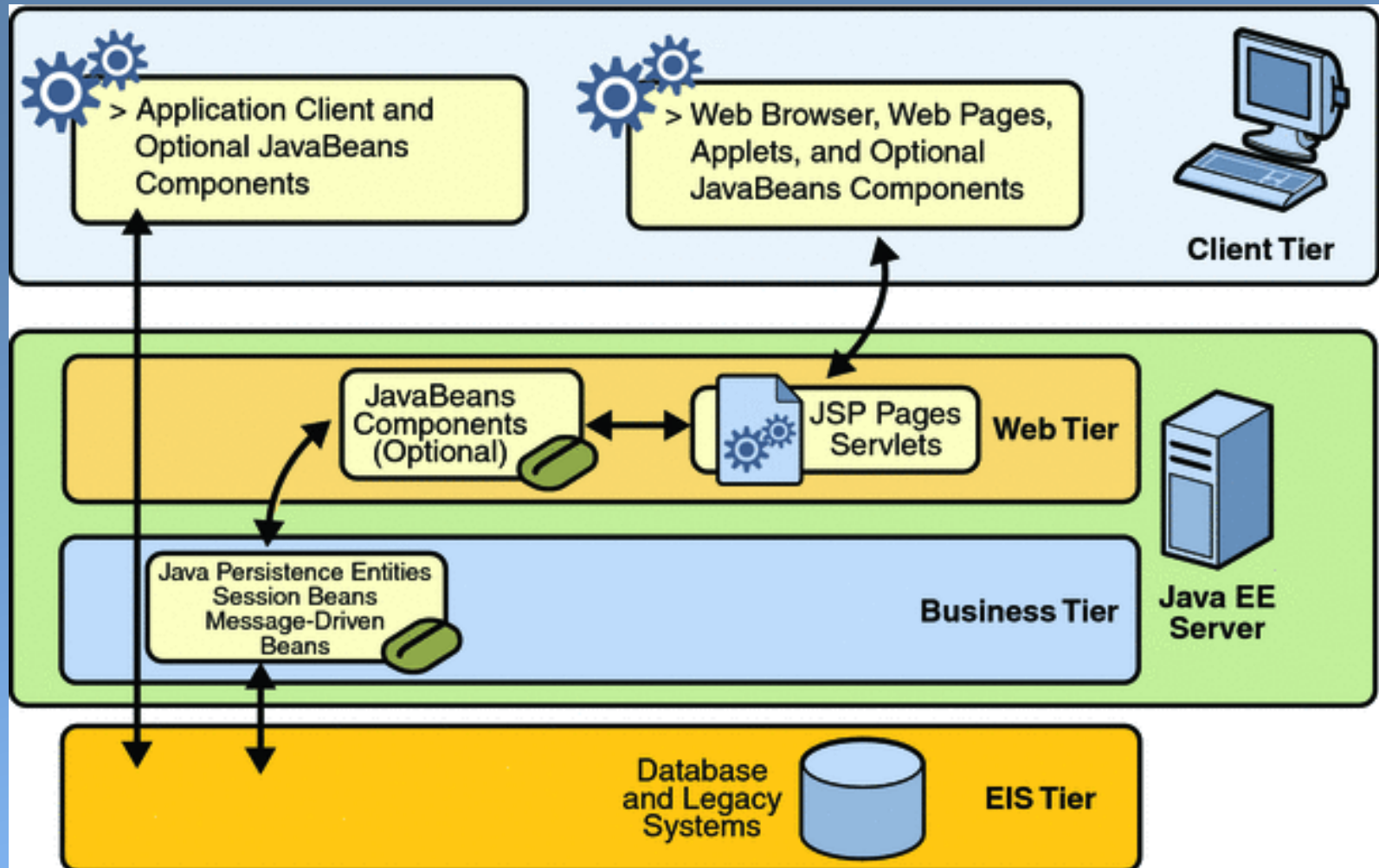
Java EE Server Communications



Web Tier and Java EE Applications



Business and EIS Tiers



Any Question ?



Thank You