Bài 13: Spring Basic

Giảng viên: Nguyễn Quang Huy
Email: HuyNQ12@topica.edu.vn

Differences between Java EE and Java SE

- Java technology is both a **programming language** and a **platform**
  - The Java **programming language** is a **high-level object-oriented language** that has a particular **syntax** and **style**
  - A Java **platform** is a particular **environment** in which Java programming language applications run
- There are four platforms of the Java programming language
  - Standard Edition (Java SE)
  - Enterprise Edition (Java EE)
  - Micro Edition (Java ME)
  - JavaFX

The Java Programming Language Platforms

❑ Standard Edition

- provides core functionality
- defines basic types and objects, high-level classes that are used for networking, security, database access, graphical user interface (GUI) development, and XML parsing
- consists of a virtual machine, development tools, deployment technologies, and other class libraries and toolkits

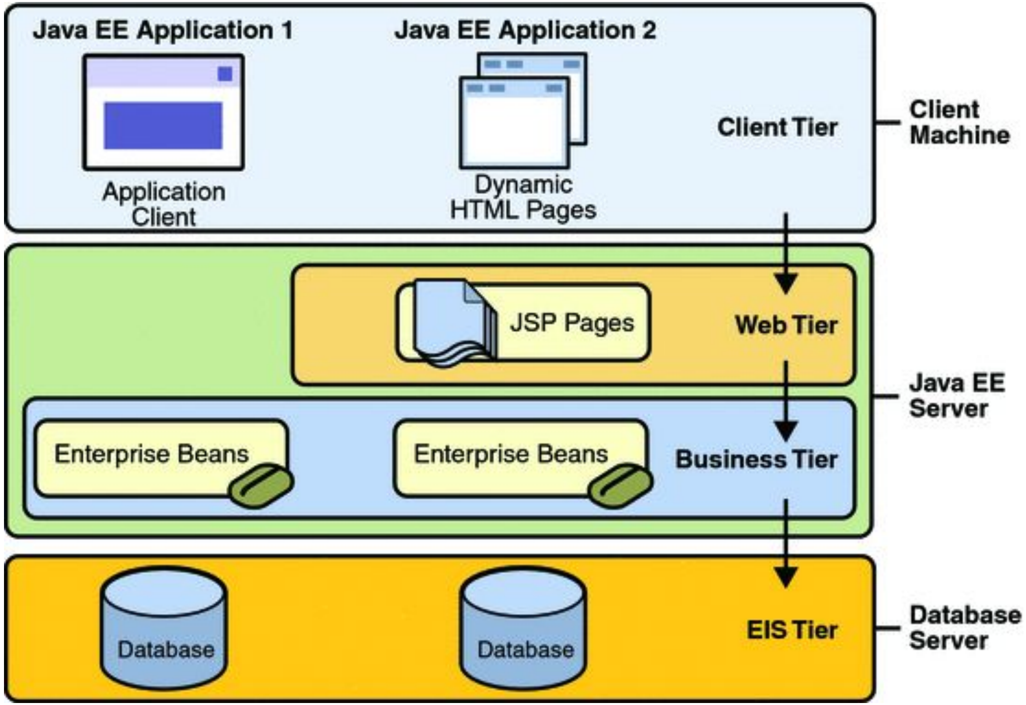
❑ Enterprise Edition

- built on top of the Java SE platform
- provides an API and runtime environment for developing and running large-scale, multi-tiered, scalable, reliable, and secure network applications

The Java Programming Language Platforms (continue)

- ❑ Micro Edition
  - ○ provides an API and a small-footprint virtual machine for running Java programming language applications on small devices, like mobile phones
  - ○ a subset of the Java SE API, along with special class libraries useful for small device application development
  - ○ often clients of Java EE platform services
- ❑ JavaFX
  - ○ a platform for creating rich internet applications using a lightweight user-interface API
  - ○ use hardware-accelerated graphics and media engines to take advantage of higher-performance clients and a modern look-and-feel as well as high-level APIs for connecting to networked data sources
  - ○ may be clients of Java EE platform services

Java EE - Distributed Multitiered Applications



Java EE - Security

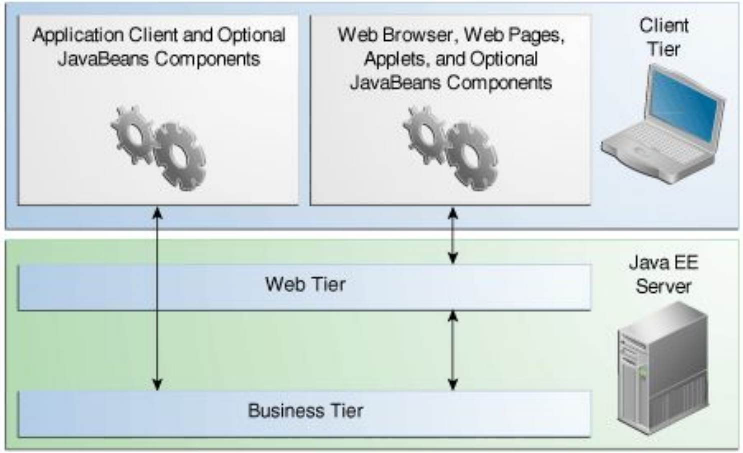
- ❑ Enables security constraints to be defined at **deployment time**
- ❑ Applications portable to a wide variety of security implementations by shielding application developers from the complexity of implementing security features
- ❑ Provides standard declarative access control rules that are defined by the developer and interpreted when the application is deployed on the server
- ❑ Provides standard login mechanisms so that application developers do not have to implement these mechanisms in their applications

Java EE - Components

- Java EE applications are made up of components
- A Java EE component is a self-contained functional software unit that is assembled into a Java EE application with its related classes and files and that communicates with other 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.
    - EJB components (enterprise beans) are business components that run on the server
- Java EE components are **assembled** into a Java EE application, they are **verified** to be well formed and in compliance with the Java EE specification, and they are **deployed** to production, where they are **run and managed** by the Java EE server

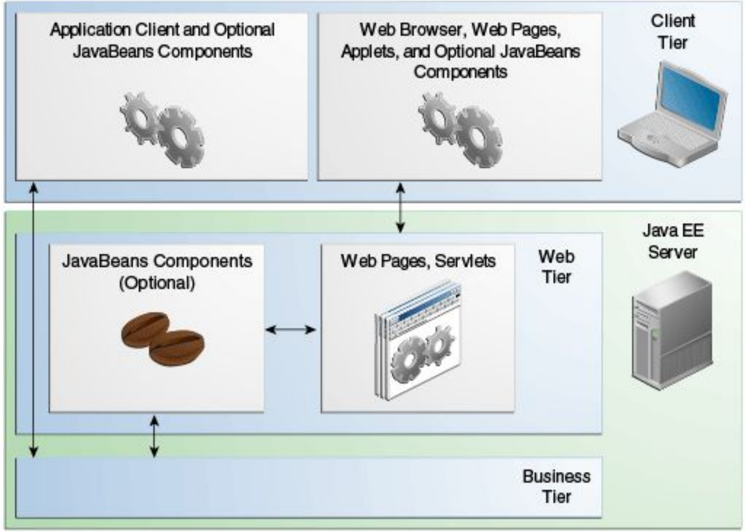
Java EE - Clients

- **Web Clients** consists of two parts
  - Dynamic web pages containing various types of markup language (HTML, XML, and so on), which are generated by web components running in the web tier
  - A web browser, which renders the pages received from the server
- **Application Clients**
  - provides a way for users to handle tasks that require a richer user interface than can be provided by a markup language
- Applets
  - a small client application that executes in the Java virtual machine installed in the web browser


Application Client and Optional JavaBeans Components
Web Browser, Web Pages, Applets, and Optional JavaBeans Components
Client Tier
Java EE Server
Web Tier
Business Tier

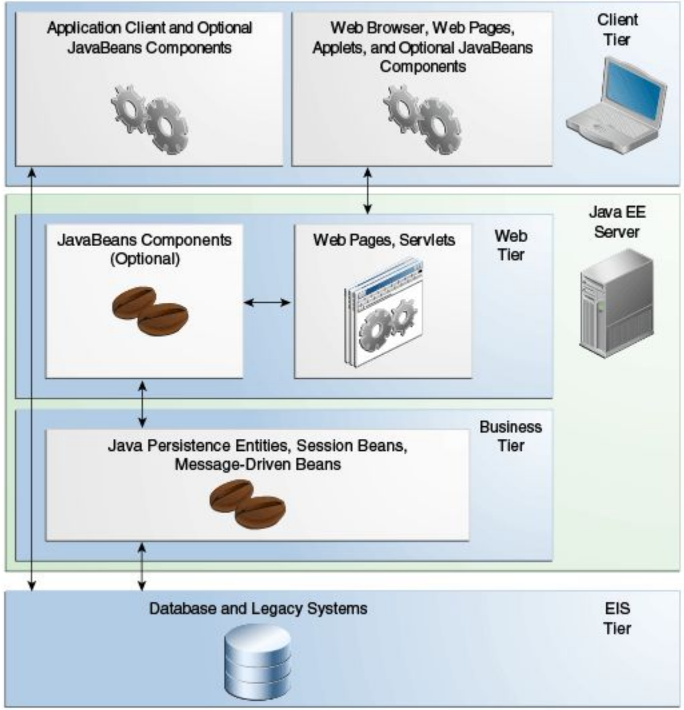
Java EE - Web Components

- Are either **servlets** or **web pages** created using **JavaServer Faces** technology and/or **JSP** technology (JSP pages)
  - Servlets are Java programming language classes that dynamically process requests and construct responses
  - JSP pages are text-based documents that execute as servlets but allow a more natural approach to creating static content
  - JavaServer Faces technology builds on servlets and JSP technology and provides a user interface component framework for web applications
- Might include a JavaBeans component to manage the user input and send that input to enterprise beans running in the business tier for processing


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 EE Server
Business Tier

Java EE - Business Components

- Business code, which is logic that solves or meets the needs of a particular business domain such as banking, retail, or finance, is handled by enterprise beans running in either the business tier or the web tier


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 EE Server
Java Persistence Entities, Session Beans, Message-Driven Beans
Business Tier
Database and Legacy Systems
EIS Tier

Java EE - Enterprise Information System Tier

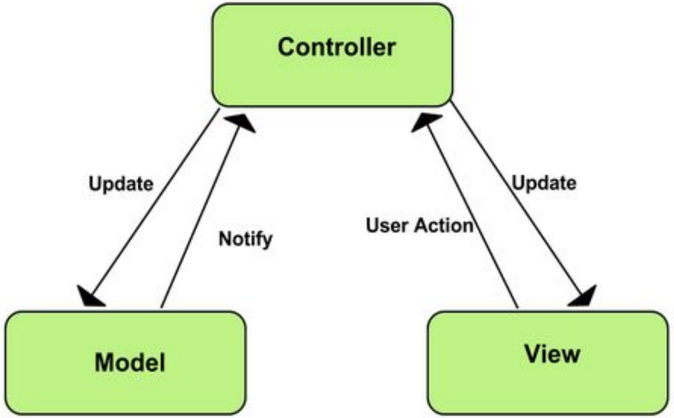
The enterprise information system tier handles EIS software and includes enterprise infrastructure systems, such as enterprise resource planning (ERP), mainframe transaction processing, database systems, and other legacy information systems. For example, Java EE application components might need access to enterprise information systems for database connectivity.

What’s MVC?

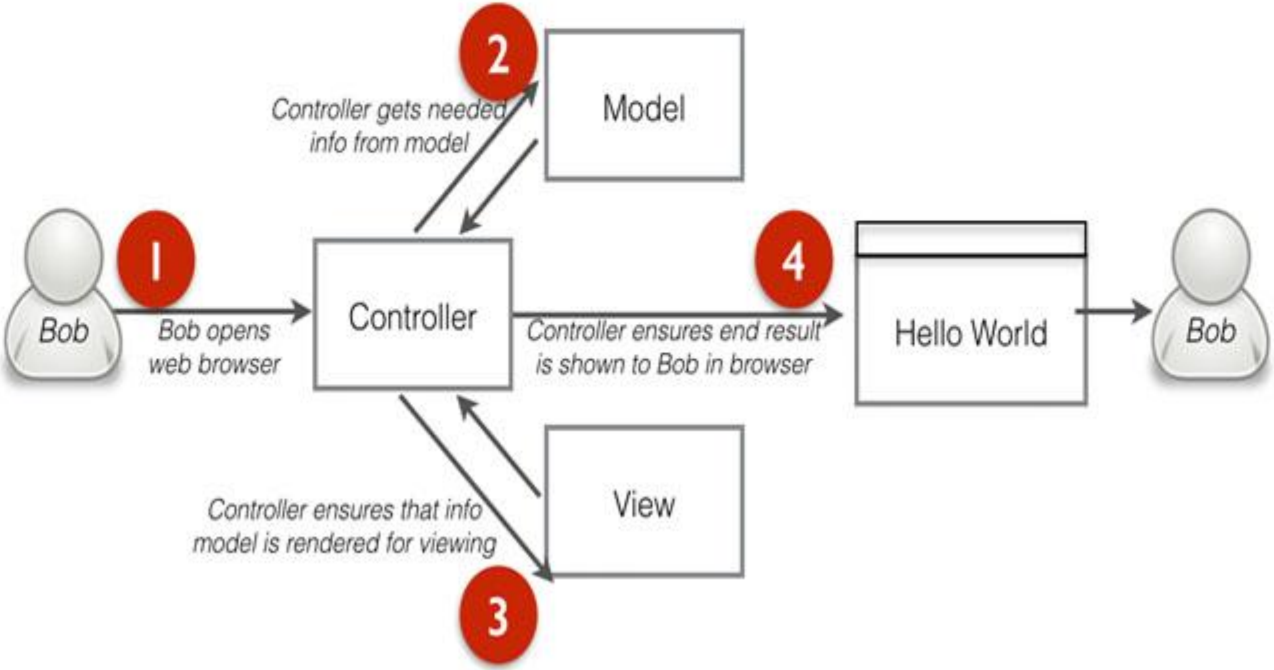
MVC pattern

Stand for Model-View-Controller

- Model
  - represents an object or JAVA POJO carrying data
  - can also have logic to update controller if its data changes
- View
  - represents the visualization of the data that model contains
- Controller
  - acts on both model and view
  - controls the data flow into model object and updates the view whenever data changes
  - keeps view and model separate



What’s WEB MVC?


2
Controller gets needed info from model
Model
1
Bob
Bob opens web browser
Controller
4
Controller ensures end result is shown to Bob in browser
Hello World
Bob
Controller ensures that info model is rendered for viewing
View
3

Q&A

Task

- Build an Vietnamese - English dictionary application use Java EE
  - Business requirements:
    - Allow user search by type in textbox then show result in table
    - Allow admin import data for dictionary by upload new file
    - Allow search by “tiếng Việt có dấu hoặc không dấu”
  - Technical requirement
    - Use GlassFish
    - Dictionary store in MySQL
  - Bonus: use **JavaServer Faces**, use **Java Persistence API (JPA)**, use & explain **Contexts and Dependency Injection for Java EE**, use **Java Authorization Contract for Containers**