Bài 1: Tổng quan về

# Phát triển Web với Java EE

# Mục tiêu bài học

- - Giới thiệu Java EE
- - Mô hình web service trên Java EE
- - Web application,components, và Web container
- - Cấu hình ứng dụng Web
- - Giới thiệu về mẫu thiết kế MVC
- - Một số web application framework

# Mở đầu

- Sự phát triển như vũ bão của công nghệ thông tin, nhất là mạng Internet đã khiến cho mô hình lập trình ứng dụng thay đổi rất nhiều.
- Các chương trình cần phải tương tác được với người dùng, chia sẻ tài nguyên, kết nối từ xa, phân tán dữ liệu … Với những yêu cầu trên mô hình khách/chủ (client-server) đã ra đời và tỏ ra rất hiệu quả trong thời gian dài

# Mở đầu

- Tuy nhiên cả máy khách và máy chủ ngày càng trở nên quá tải bởi độ phức tạp và yêu cầu của người dùng. Từ đó phát sinh mô hình phát triển ứng dụng đa tầng (multi-tier).
- Mục tiêu là làm cho máy khách trở nên gọn nhẹ, dễ cấu hình. Tất cả mã nguồn lõi, cài đặt, xử lý đều thực hiện trên máy chủ, do đó chương trình trở nên dễ quản lý, các máy khách luôn được phục vụ với phiên bản chương trình mới nhất
- Web là một ví dụ điển hình nhất của mô hình ứng dụng đa tầng
- Mô hình ứng dụng đa tầng đáp ứng được nhu cầu về mặt tốc độ, bảo mật, cũng như sự đáng tin cậy của ứng dụng

# Java EE là gì

- Java EE là viết tắt của Java Platform, Enterprise Edition là nền tảng tiêu chuẩn và mở để xây dựng, phát triển các ứng dụng doanh nghiệp lớn, bao gồm: ứng dụng mạng, web, đa tầng, phân tán… J2EE là mở rộng của J2SE
- Các phiên bản
  - J2EE 1.2 (December 12, 1999)
  - J2EE 1.3 (September 24, 2001)
  - J2EE 1.4 (November 11, 2003)
  - Java EE 5 (May 11, 2006)
  - Java EE 6 (December 10, 2009)
  - Java EE 7 (May 28, 2013)

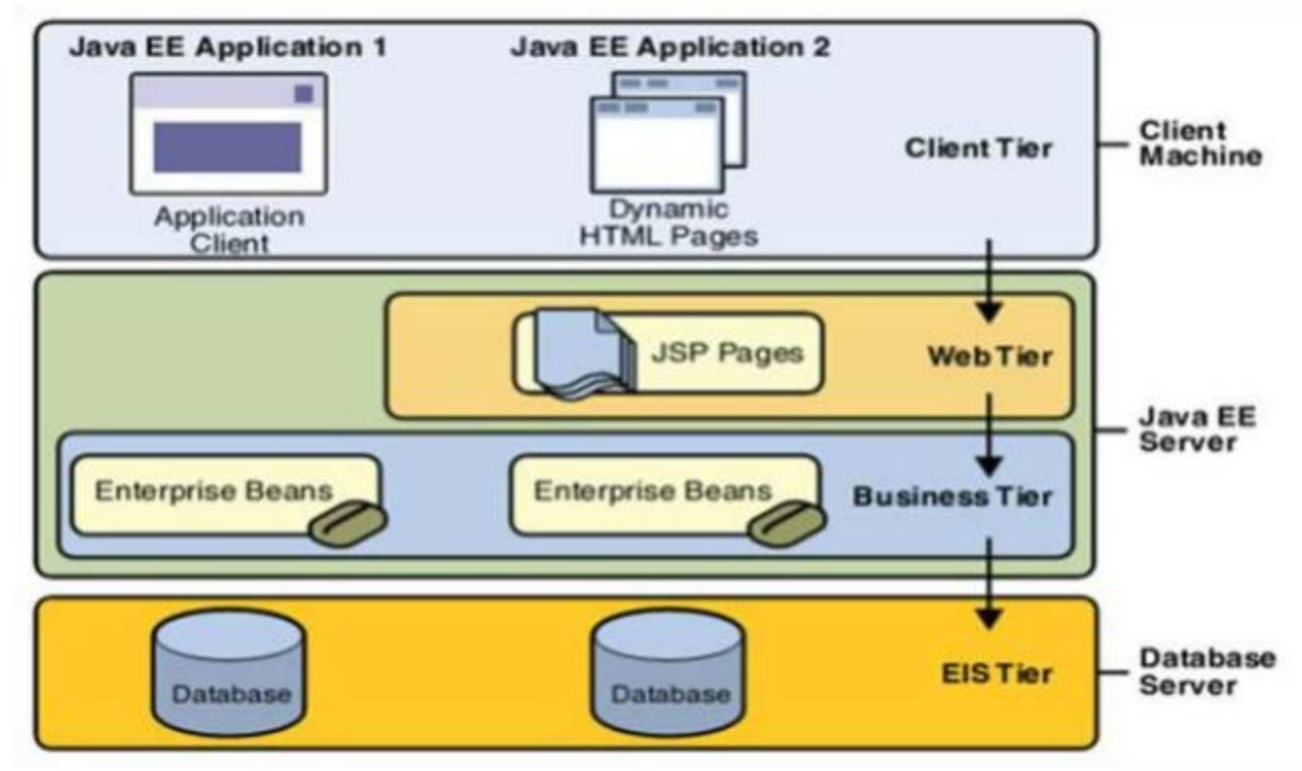
# Java EE là gì

- J2EE cung cấp các API cho việc phát triển ứng dụng nhằm:
  - Giảm thời gian phát triển ứng dụng
  - Giảm độ phức tạp của ứng dụng
  - Tăng hiệu suất ứng dụng

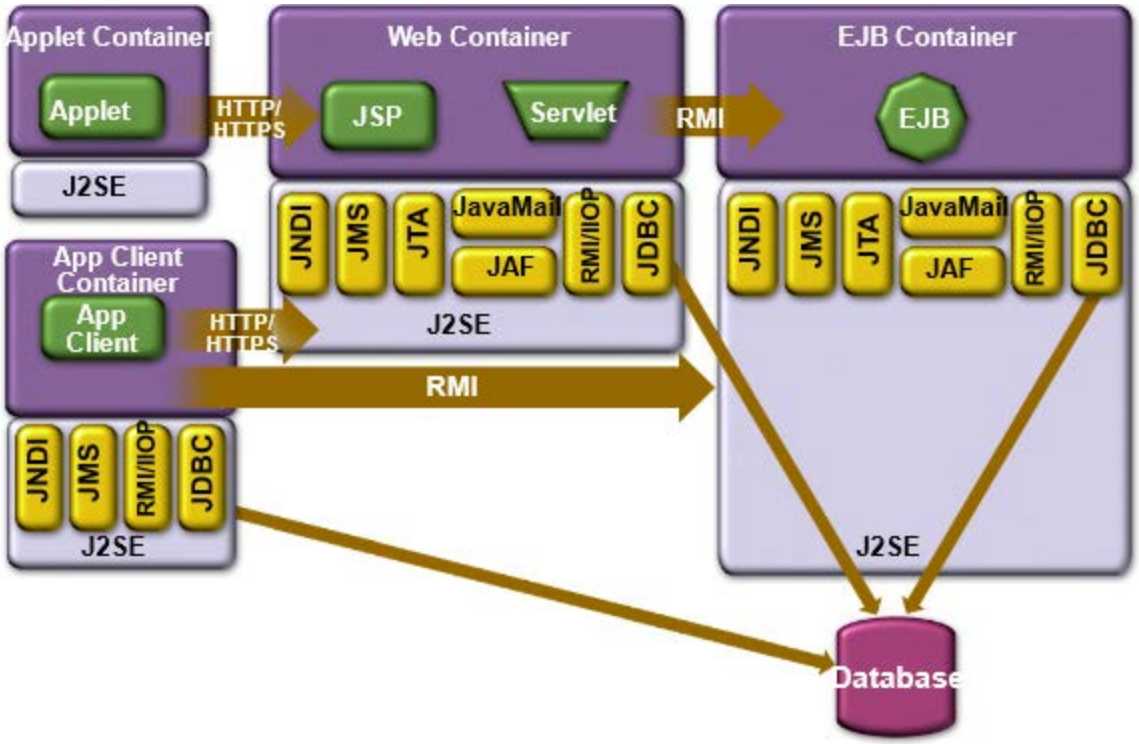
# Java EE là gì

- Là kiến trúc ứng dụng đa tầng với ưu điểm:
  - Khả năng mở rộng
  - Khả năng truy cập
  - Khả năng quản lý
- Mô hình kiến trúc chia làm 2 tầng:
  - Tầng trình diễn
  - Tầng nghiệp vụ

# Các thành phần Java EE



# Java EE Container



# Java EE Container

- Container cung cấp các dịch vụ :
  - Dịch vụ bảo mật (security service)
  - Dịch vụ giao dịch (transaction service)
  - Dịch vụ JNDI (JNDI lookup service)

# Java EE Container

- Java Application – component này là 1 chương trình standalone chạy bên trong Application Client container. Application Client container cung cấp những API hỗ trợ cho messaging, remote invocation, database connectivity và lookup service. Application Client container đòi hỏi những API sau: J2SE, JMS, JNDI, RIM-IIOP và JDBC. Container này được cung cấp bởi nhà cung cấp application server
- Applet – Applet component là java applet chạy bên trong Applet container, chính là web browser có hỗ trợ công nghệ Java. Applet phải hỗ trợ J2SE API.

# Java EE Container

- Servlet và JSP – đây là Web-based component chạy ở bên trong Web container, được hỗ trợ bởi Web Server. Web container là một môi trường run-time cho servlet và jsp. Web Container phải hỗ trợ những API sau: J2SE, JMS, JNDI, JTA, JavaMail, JAF, RIM-IIOP và JDBC. Serlet và JSP cung cấp một cơ chế cho việc chuẩn bị, xử lý, định dạng nội dung động
- Enterprise JavaBean (EJB) – EJB component là business component chạy bên trong EJB container. EJB component là phần nhân, cốt lõi của ứng dụng J2EE. EJB container cung cấp các dịch vự quản lý transaction, bảo mật, quản lý trạng thái, quay vòng tài nguyên (resource pooling). EJB container phải hỗ trợ những API sau: J2SE, JMS, JNDI, JTA, JavaMail, JAF, RIM-IIOP và JDBC.

# Web components

- Web components: Servlet hoặc JSP (cùng với JavaBean và custom tags)
- Các web components chạy trên 1 web container
- Các web container phổ biến: Tomcat, Resin
- Web container cung cấp các dịch vụ hệ thống (system services) cho các Web components
  - Request dispatching, security, và quản lý vòng đời

# Web Application

- Web Application là 1 gói triển khai, gồm:
  - Web components (Servlet và JSP)
  - Tài nguyên tĩnh như images
  - Helper classes (sử dụng bởi web components)
  - Thư việc Libraries
  - Deployment desciptor (web.xml)
- Web Application có thể được tổ chức thành:
  - Phân cấp các thư mục và files ( dạng chưa đóng gói)
  - File *.WAR: dạng đóng gói ( bên trong có phân cấp ), sử dụng khi muốn triển khai trên một remote machine

# Cấu hình ứng dụng web:

- Thông số cấu hình được đặc tả trong file web.xml (Web Applications Deployment Descriptor)

# Cấu hình ứng dụng web:

- Tất cả tài liệu XML cần có prolog

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app
 xmlns="http://java.sun.com/xml/ns/javaee"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-
 instance"
 xsi:schemaLocation=http://java.sun.com/xml/n
 s/javaee
 http://java.sun.com/xml/ns/javaee/web-
 app_3_0.xsd   version="3.0">
```

# Alias Path

- Khi 1 Servlet container nhận 1 request, nó cần biết Web component nào trong ứng dụng Web sẽ xử lý request này.
  - Thực hiện map URL path trong request tới 1 Web component
- Alias Path có thể có 2 dạng
  - /alias-string (cho servlet) hoặc
  - /*.jsp (cho JSP)

# Chú ý

- Phân biệt chữ hoa-thường
- Phải đúng thứ tự như sau:
  - icon, display-name, description, distributable
  - context-param, filter, filter-mapping
  - listener, servet, servlet-mapping, session-config
  - mime-mapping, welcome-file-list
  - error-page, taglib, resource-env-ref, resource-ref
  - security-constraint, login-config, security-role
  - env-entry, ejb-ref, ejb-local-ref

# Tham số khởi tạo và ngữ cảnh (Context and Initialization Parameters)

- Biểu diễn ngữ cảnh trong ứng dụng
- Có thể được dùng chung bởi các Web components trong cùng 1 file WAR

```
<web-app>
      ...
      <context-param>
          <param-name>
                javax.servlet.jsp.jstl.fmt.localizationContext
          </param-name>
          <param-value>messages.BookstoreMessages</param-value>
      </context-param>
          ...
</web-app>
```

# Event Listeners

- Nhận các events trong vòng đời của servlet

```
<listener>
      <listener-class>
          listeners.ContextListener
      </listener-class>
</listener>
```

# Filter Mappings

- Chỉ ra filters nào được áp dụng cho request nào, và theo thứ tự nào

```
<filter>
        <filter-name>OrderFilter<filter-name>
        <filter-class>filters.OrderFilter<filter-class>
</filter>
<filter-mapping>
        <filter-name>OrderFilter</filter-name>
        <url-pattern>/receipt</url-pattern>
</filter-mapping>
```

# Error Mappings

- Ánh xạ (Map)
  - mã trạng thái (status code) trả về trong 1 HTTP response do có 1 ngoại lệ Java
  - với 1 Web resource (ví dụ các trang error page)

```
<error-page>
 <exception-type>exception.OrderException</exception-
   type>
 <location>/errorpage.html</location>
</error-page>
```

# References

- Sử dụng khi có web components muốn tham chiếu tới các tài nguyên (database), môi trường.
- Ví dụ: khai báo 1 tham chiếu tới data source

```
<resource-ref>
        <res-ref-name>jdbc/BookDB</res-ref-name>
        <res-type>javax.sql.DataSource</res-type>
        <res-auth>Container</res-auth>
</resource-ref>
```


DEMO
Cấu hình web.xml

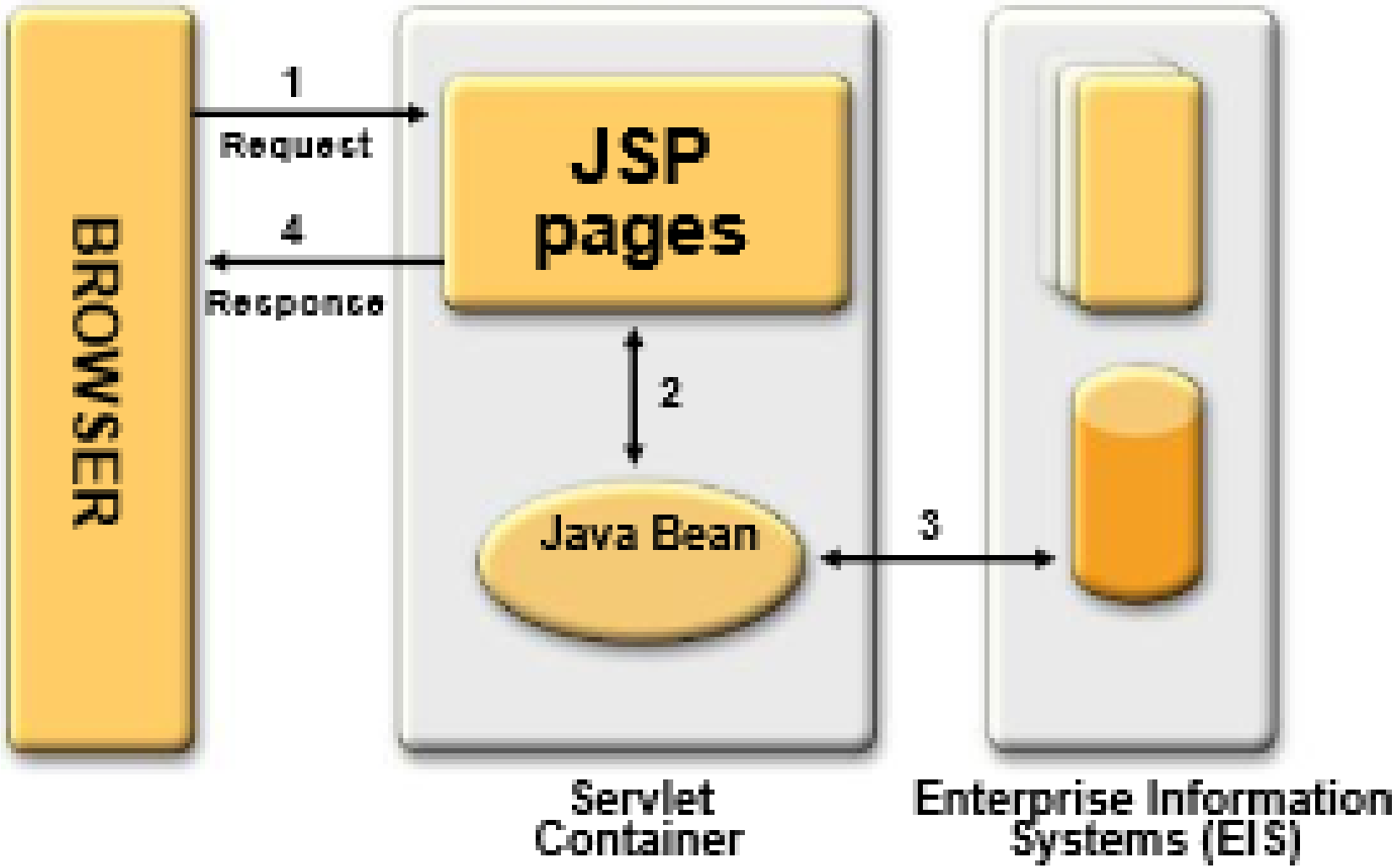
# Mẫu thiết kế phần mềm (Design Pattern)

- Một design pattern là một giải pháp chung cho một vấn đề thông thường trong công nghiệp phát triển phần mềm
- Đưa ra mô tả và cách giải quyết vấn đề trong các trường hợp khác nhau
- Design Pattern là cách giải quyết vấn đề được chuẩn hóa cho việc phát triển ứng dụng
- Một số Design Pattern
  - OOP
  - Connection pooling
  - Observer pattern
  - MVC
  - …

# Quá trình phát triển của kiến trúc ứng dụng Web

- No MVC
- MVC Model 1 (Page-centric)
- MVC Model 2 (Servlet-centric)
- Web application frameworks
  - Struts
- Standard-based Web application framework
  - JavaServer Faces (JSR-127)

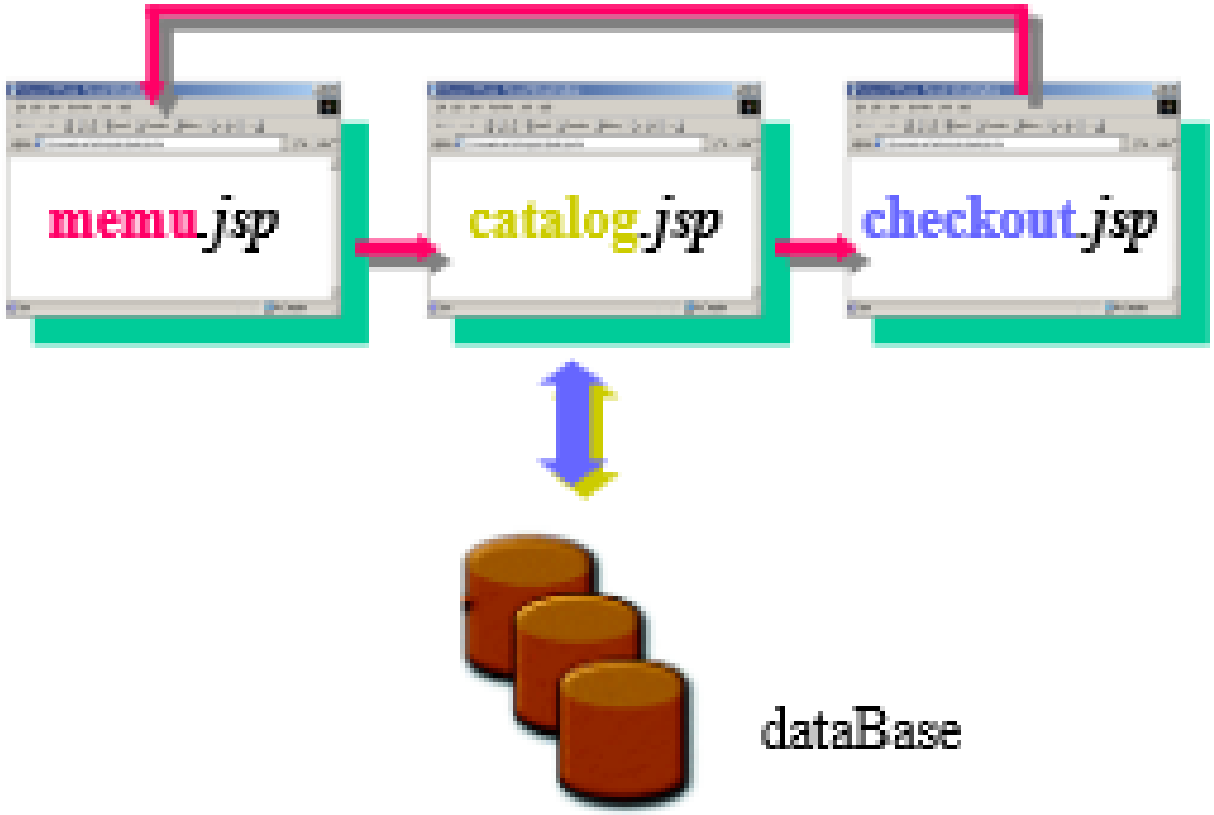
# MVC 1: Page Centric



# MVC 1: Page Centric

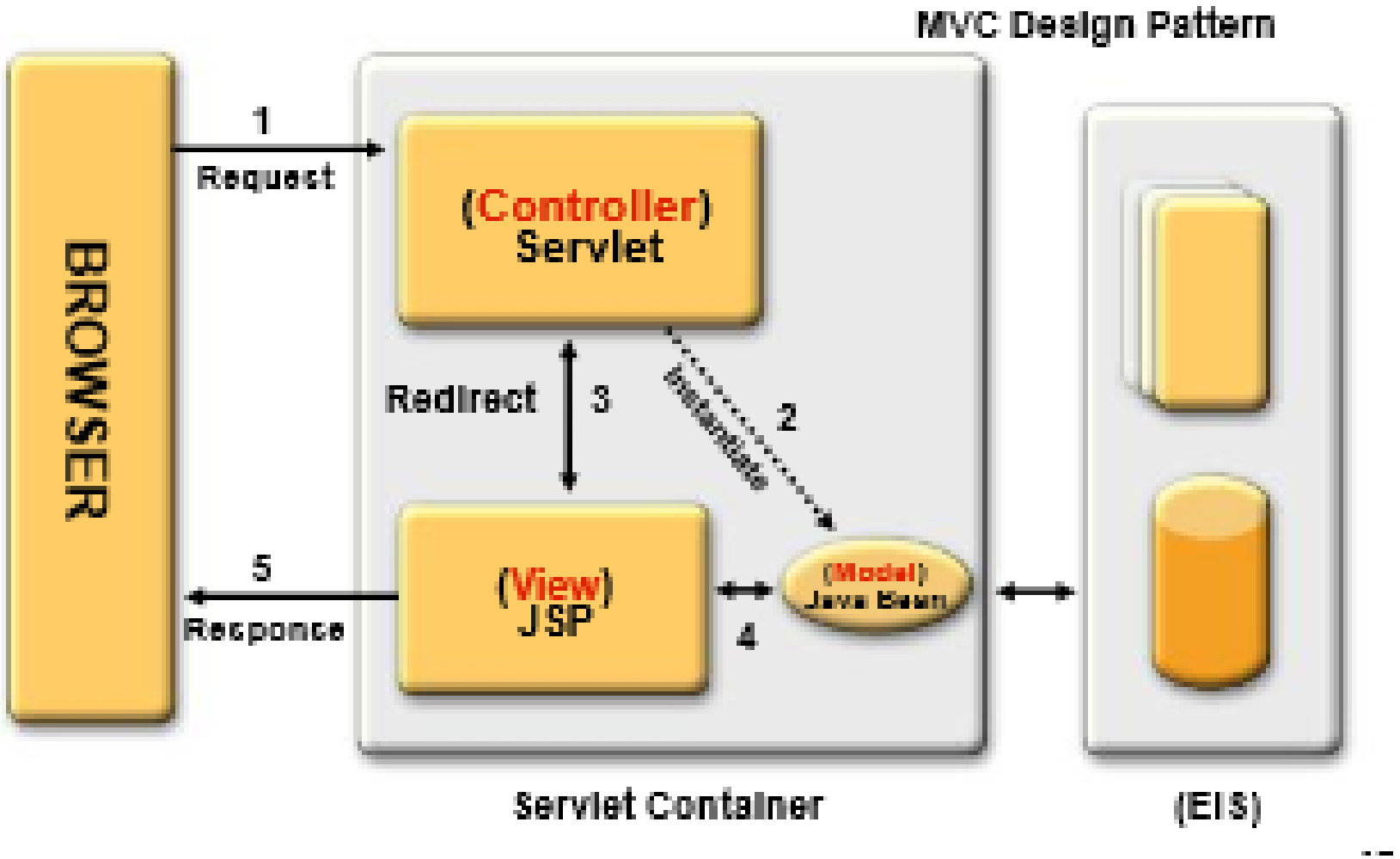
- Bao gồm 1 loạt các trang JSP có liên hệ chặt chẽ với nhau
  - Các trang JSP xử lý tất cả: presentation, control, và business process
- Business process logic và control được CODE CỨNG trong các trang JSP
  - Dưới dạng JavaBeans, scriptlets, expression
- Chuyển trang được thực hiện
  - Khi user click vào 1 liên kết. Ví dụ: <A HERF="find.jsp>
  - Qua hành động submit form. Ví dụ: <FORM ACTION="search.jsp">

# MVC 1: Page Centric




page–centric catalog application

# MVC 2: Server Centric



# MVC 2: Server Centric

- Nếu muốn biểu diễn các trang JSP khác nhau, tùy theo dữ liệu nhận được?
  - Riêng JSP với JavaBeans và custom tags (Model 1) chưa xử lý tốt được
- Giải pháp
  - Sử dụng đồng thời Servlet và JSP (Model 2)
  - Servlet xử lý request gửi tới, xử lý 1 phần dữ liệu, thiết lập các beans, forward kết quả cho 1 trong nhiều trang JSP nào đó

# MVC 2: Server Centric

- JSP chỉ được sử dụng để biểu diễn kết quả (presentation)
  - Xử lý điều khiển (Control) thực hiện bởi servlets
- Servlet hoạt động như một gatekeeper
  - Cung cấp các services thông dụng, như authentication, authorization, login, error handling, ...
- Servlet hoạt động như một central controller
  - Quyết định logic phù hợp để xử lý các request, sẽ gửi request đến những nơi nào, ...
  - Thực hiện việc điều hướng (redirecting)

# Web Application Framework

- Dựa trên kiến trúc MVC Model 2
- Hầu hết các ứng dụng Web phải cung cấp các chức năng
  - Nhận (receive) và gửi tiếp (Dispatching) HTTP requests
  - Gọi các phương thức từ tầng model
  - Tổng hợp và chọn ra các views trả về cho client
- Cung cấp các classes và interfaces cho lập trình viên sử dụng/mở rộng

# Web Application Framework

- Phân tách tầng presentation và các business logic thành các components
- Cung cấp 1 điểm điều khiển trung tâm
- Cung cấp các tính năng mở rộng
- Dễ dàng kiểm thử unit (unit-testing) và bảo trì
- Nhiều công cụ hỗ trợ
- Ổn định
- Có cộng đồng hỗ trợ mạnh mẽ
- Đơn giản hóa chế độ đa ngôn ngữ (internationalization)
- Đơn giản hóa việc validate đầu vào

# Web Application Framework

- Frameworks đang phát triển mạnh mẽ
- JSP/Servlets vẫn còn khó sử dụng
- Frameworks định nghĩa các components chuẩn, cho phép tái sử dụng.
- Frameworks còn chỉ rõ cách thức phối hợp các components trong 1 ứng dụng

# Một số Web Application Frameworks

- Apache Struts I and II
- Spring Framework MVC
- JavaServer Faces
- Echo
- Tapestry
- Wicket
- …

# XIN CẢM ƠN!