

# **Introduction to Web-Based Java Applications (PRJ301)**

**Kiều Trọng Khánh**

# Prerequisites

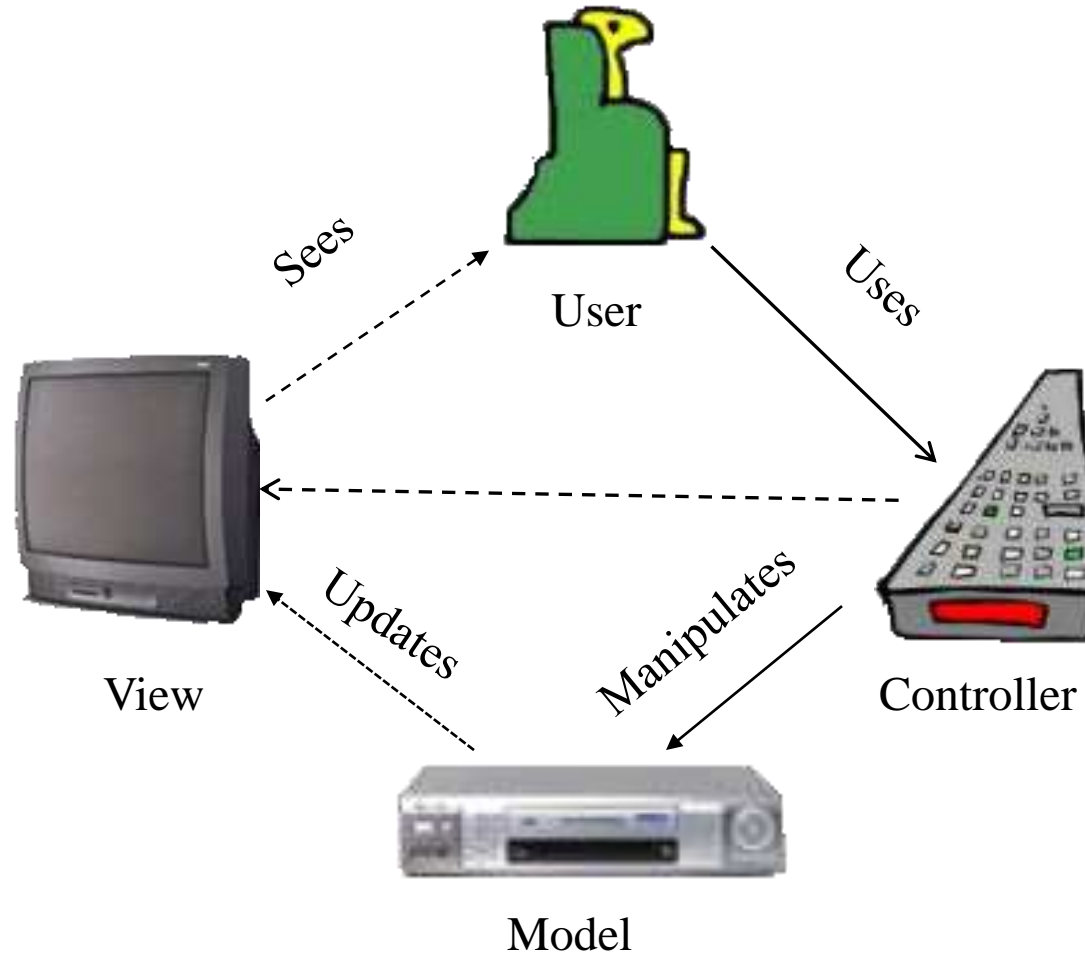
- **Completed:**
  - DBI202 (Database Systems)
  - PRO192 (Object Oriented Programming)

# Course Objectives

- This course **explores the features** of JavaEE (J2EE)
  - Understand the **core technologies** of **Java web** programming (Servlets, JSP, JavaBeans, Custom Tags, Filtering)
  - Understand and be able to **apply MVC architecture** for the web **combining with framework** (Struts 2)
  - Develop a **Web Application** (Servlets, JSP)

# MVC Design Pattern

## Model – View – Controller



**This is a MVC Model**

# Course Objectives

- The MVC architecture is applied in web application



# Course Objectives

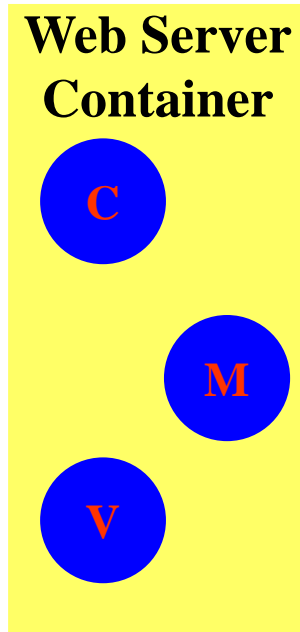
- The MVC architecture is applied in web application

**Web Server  
Container**



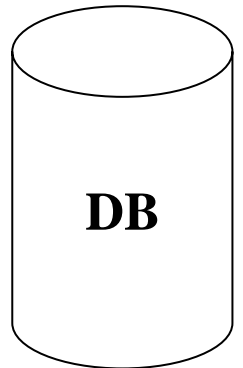
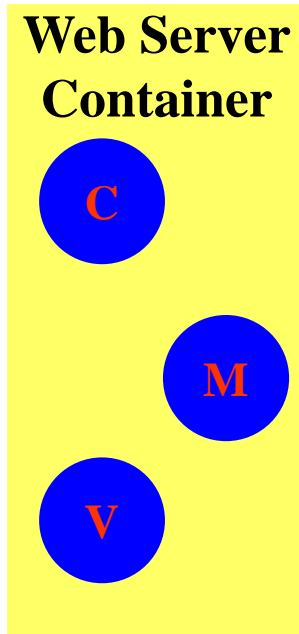
# Course Objectives

- The MVC architecture is applied in web application



# Course Objectives

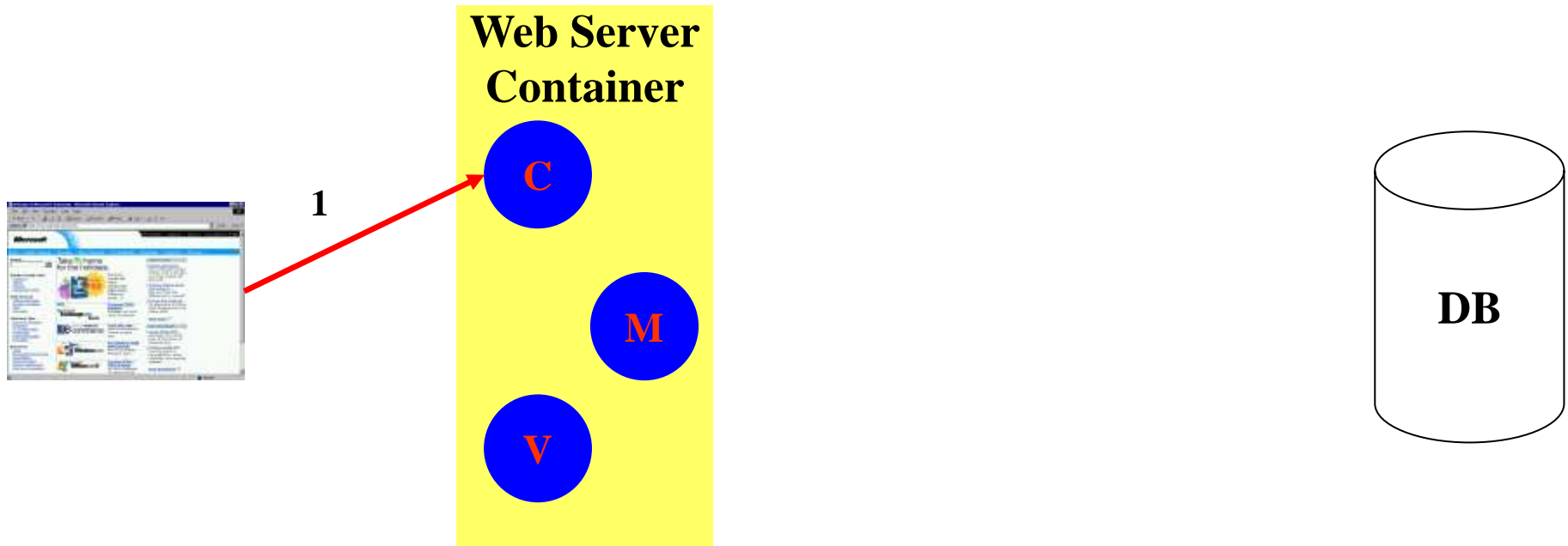
- The MVC architecture is applied in web application





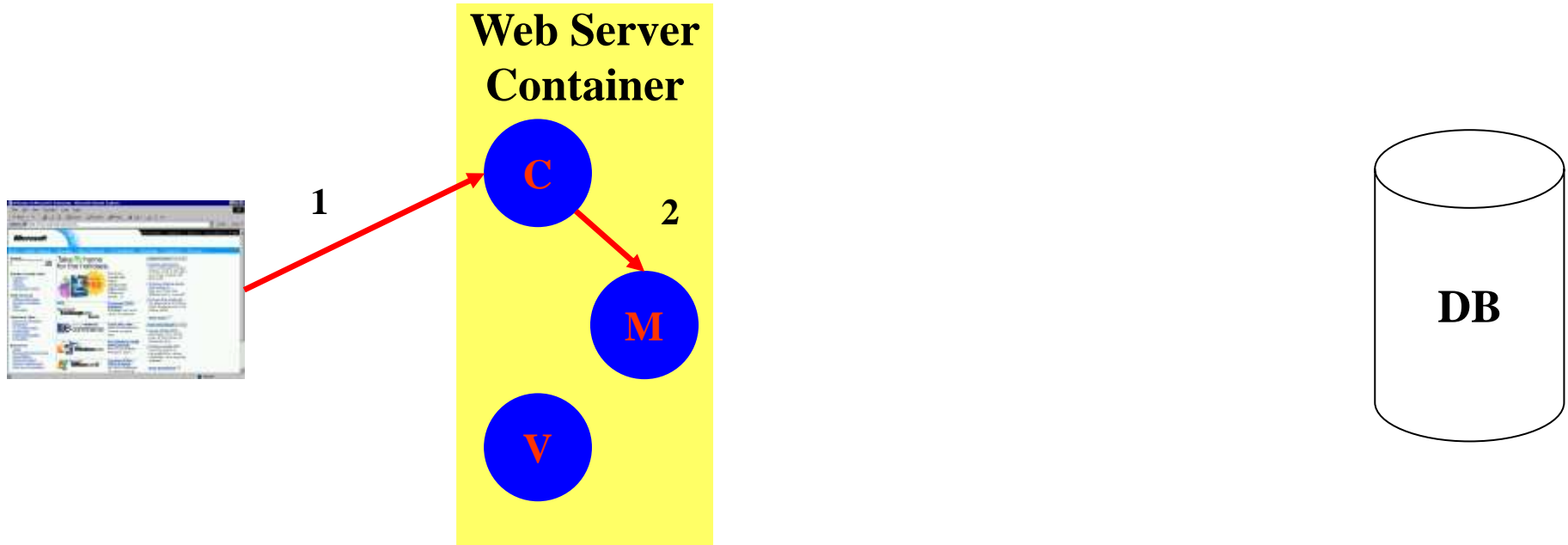
# Course Objectives

- The MVC architecture is applied in web application



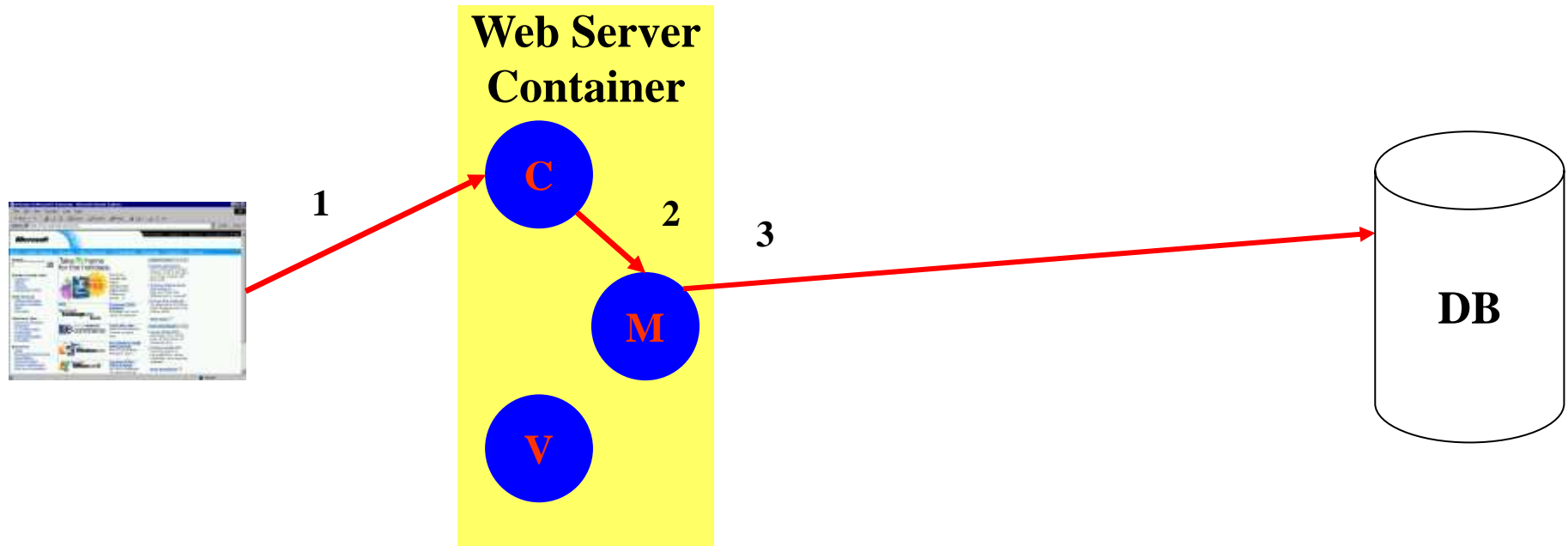
# Course Objectives

- The MVC architecture is applied in web application



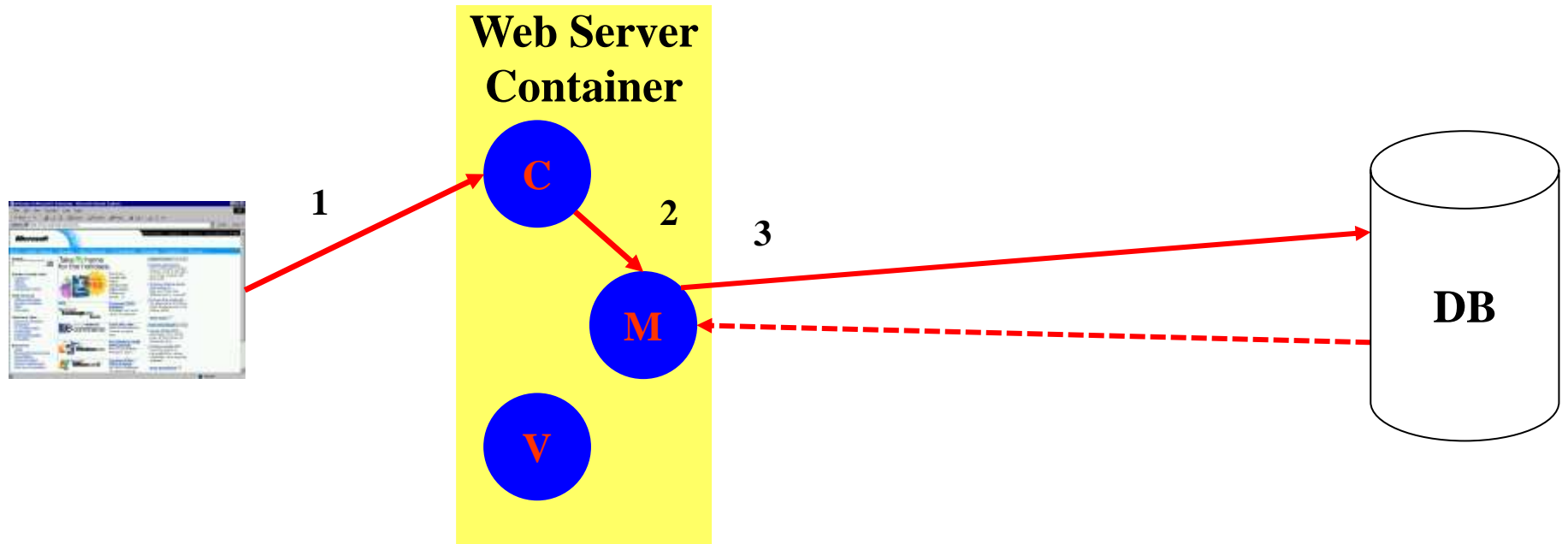
# Course Objectives

- The MVC architecture is applied in web application



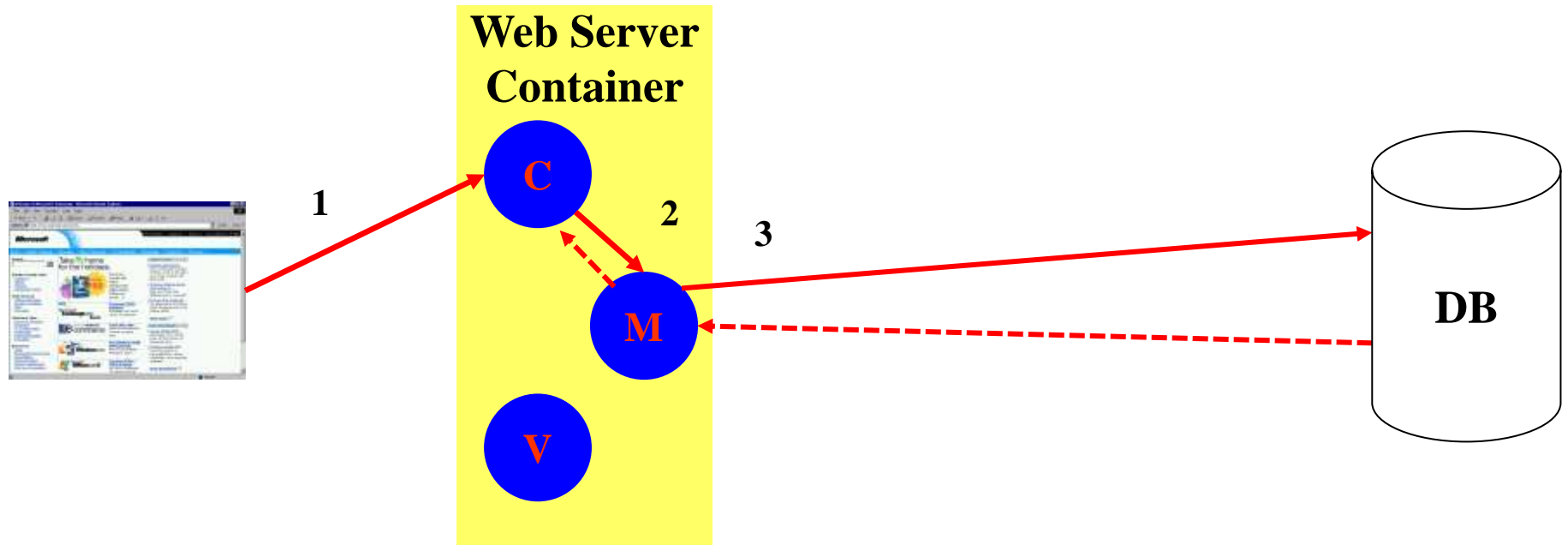
# Course Objectives

- The MVC architecture is applied in web application



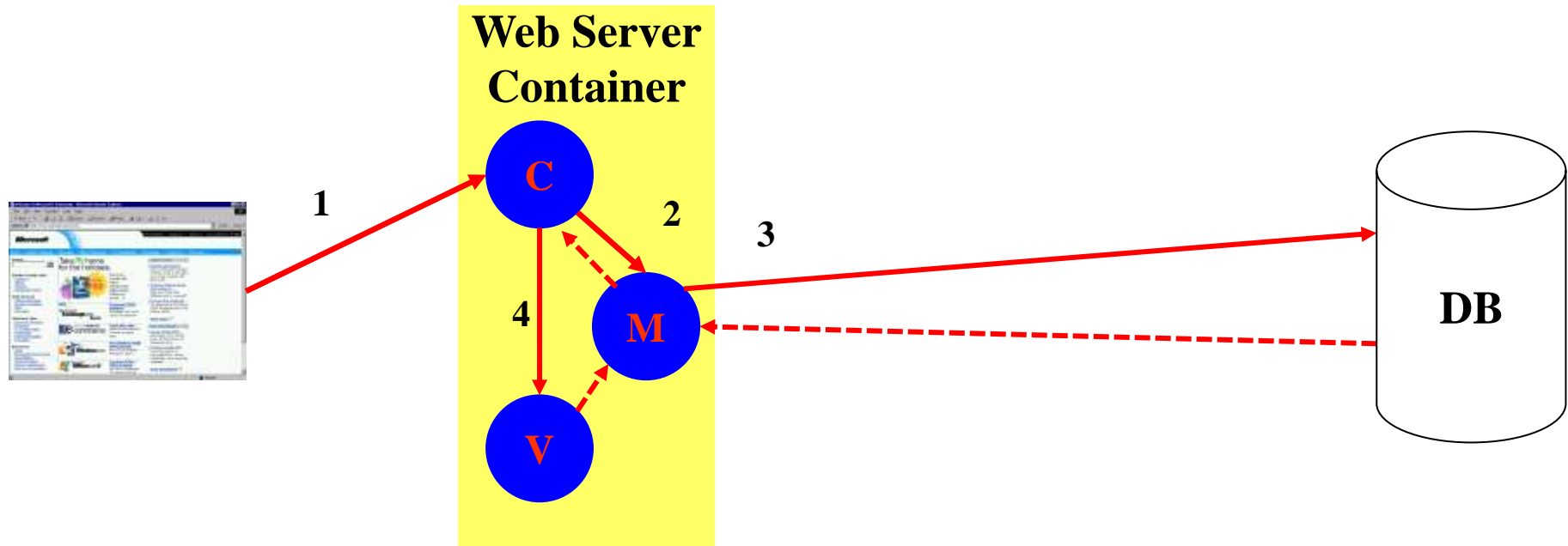
# Course Objectives

- The MVC architecture is applied in web application



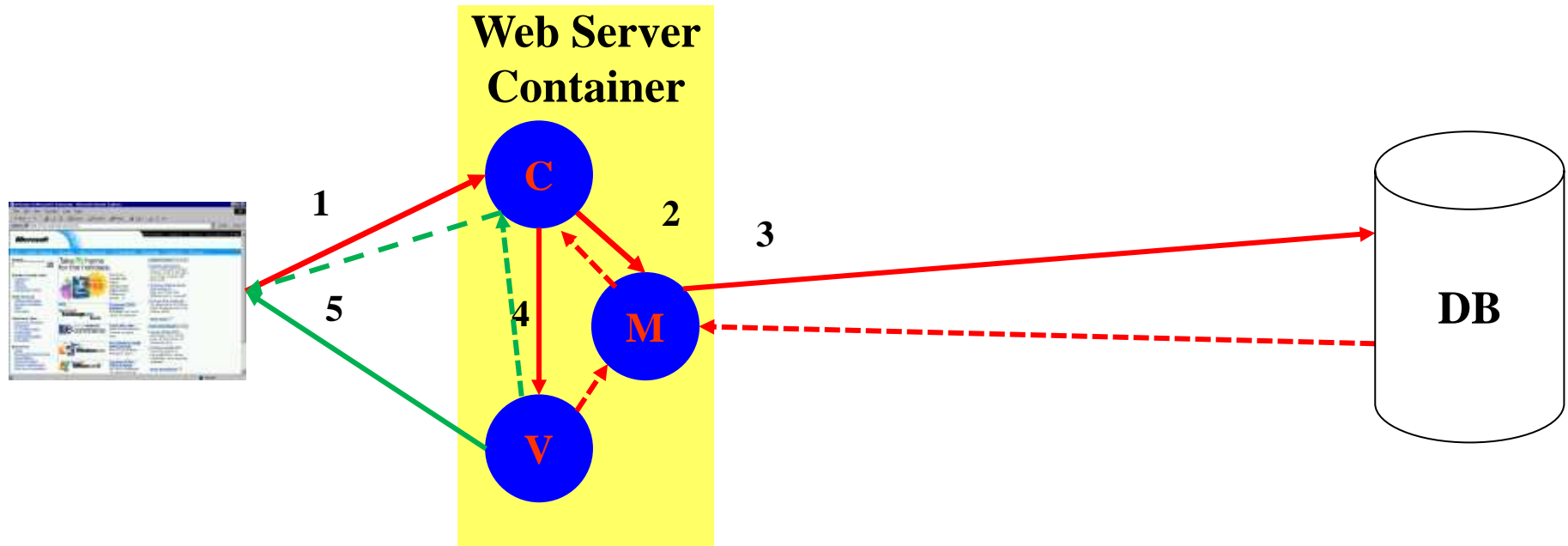
# Course Objectives

- The MVC architecture is applied in web application

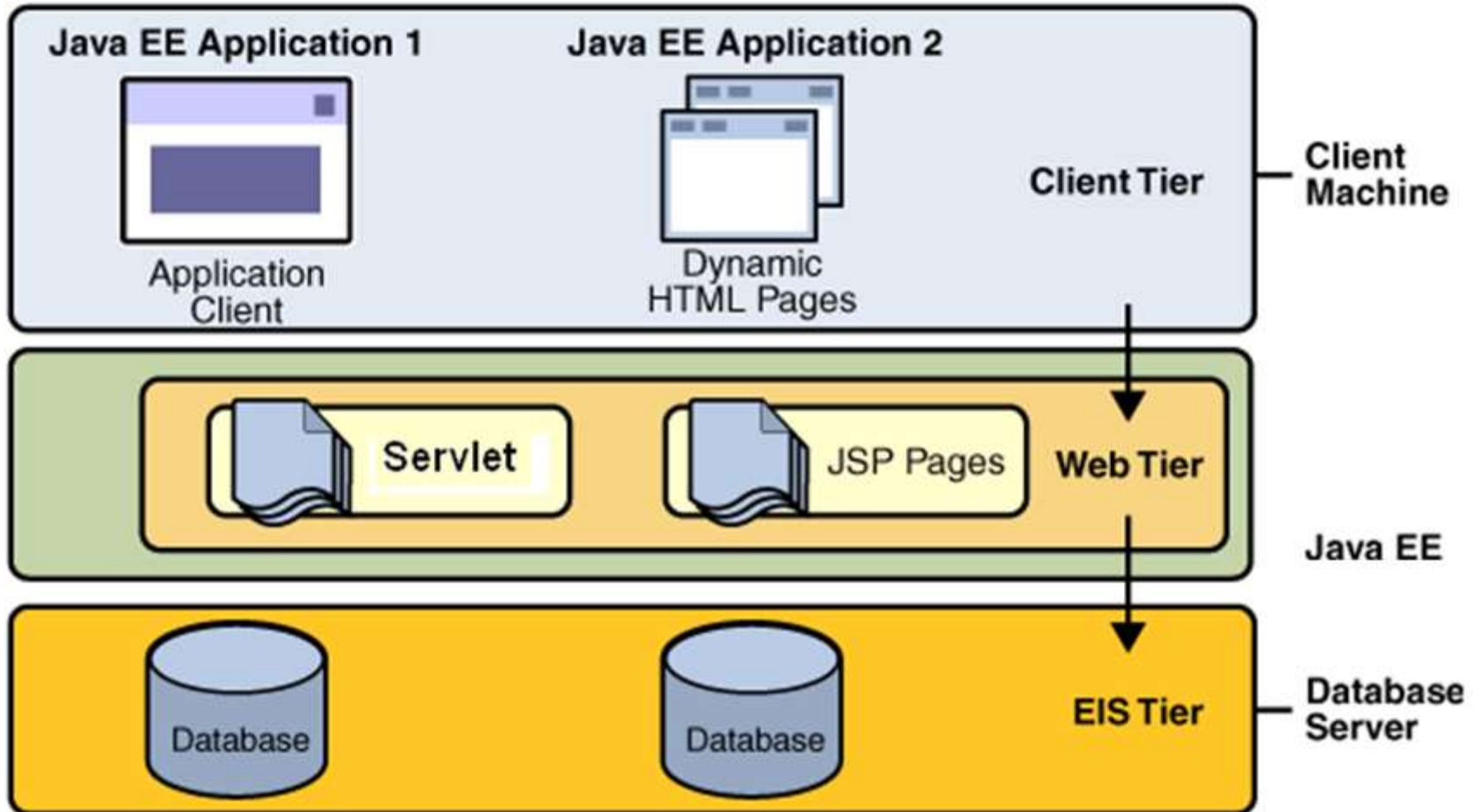


# Course Objectives

- The MVC architecture is applied in web application



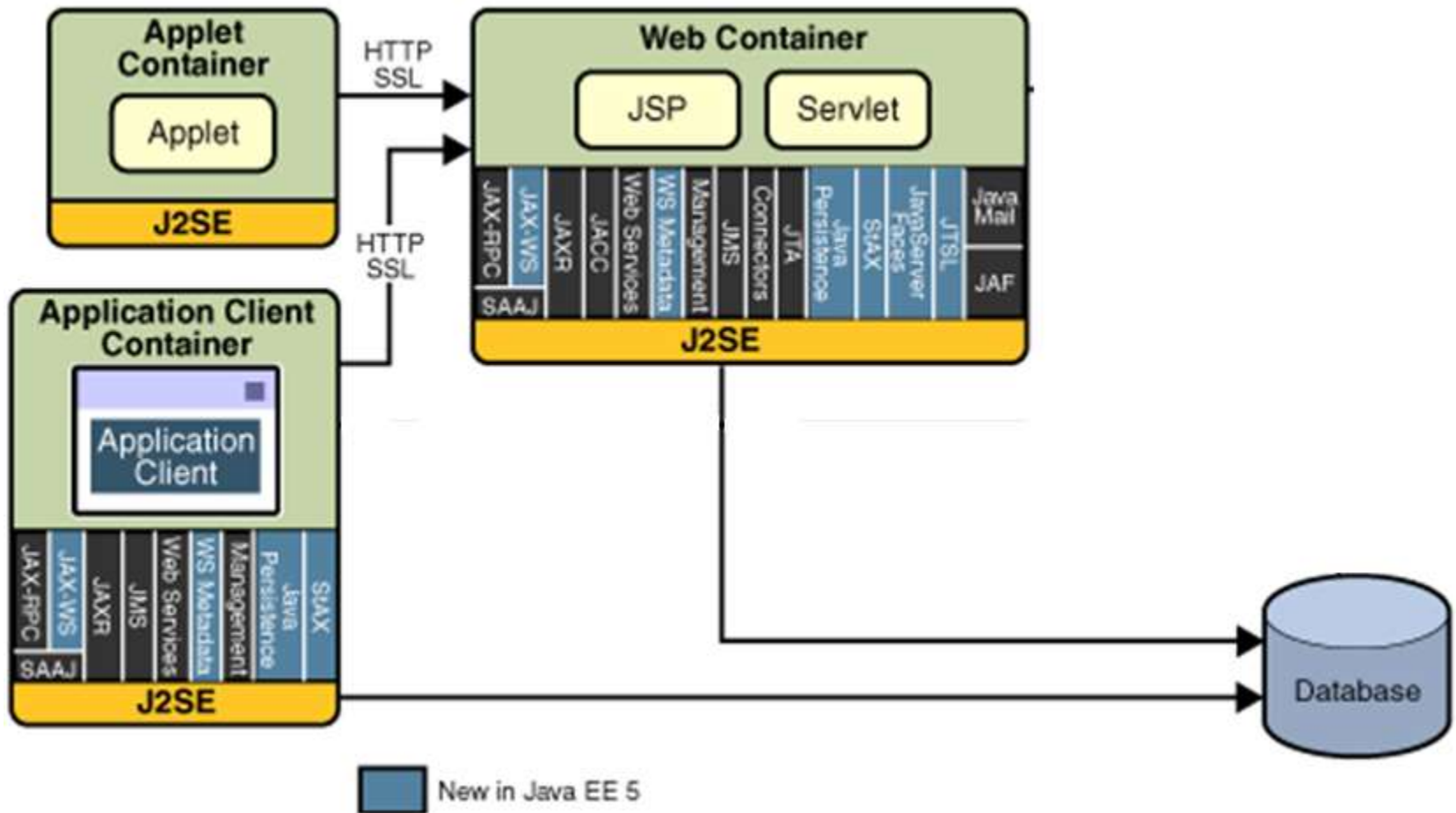
# Course Objectives





# Course Objectives

- **J2EE 1.4/ JavaEE5/ JavaEE6 Platform API**



# Course Objectives

- J2EE/JavaEE Technologies

## Java EE 6 APIs

Enterprise JavaBeans Technology

Java Servlet Technology

JavaServer Faces Technology

JavaServer Pages Technology

JavaServer Pages Standard Tag Library

Java Persistence API

Java Transaction API

Java API for RESTful Web Services

Managed Beans

Contexts and Dependency Injection for the Java EE Platform (JSR 298)

Dependency Injection for Java (JSR 330)

Bean Validation

Java Message Service API

Java EE Connector Architecture

JavaMail API

Java Authorization Contract for Containers

Java Authentication Service Provider Interface for Containers

## Java EE 6 APIs in the Java Platform, Standard Edition 6 and 7

Java Database Connectivity API

Java Naming and Directory Interface API

JavaBeans Activation Framework

Java API for XML Processing

Java Architecture for XML Binding

SOAP with Attachments API for Java

Java API for XML Web Services

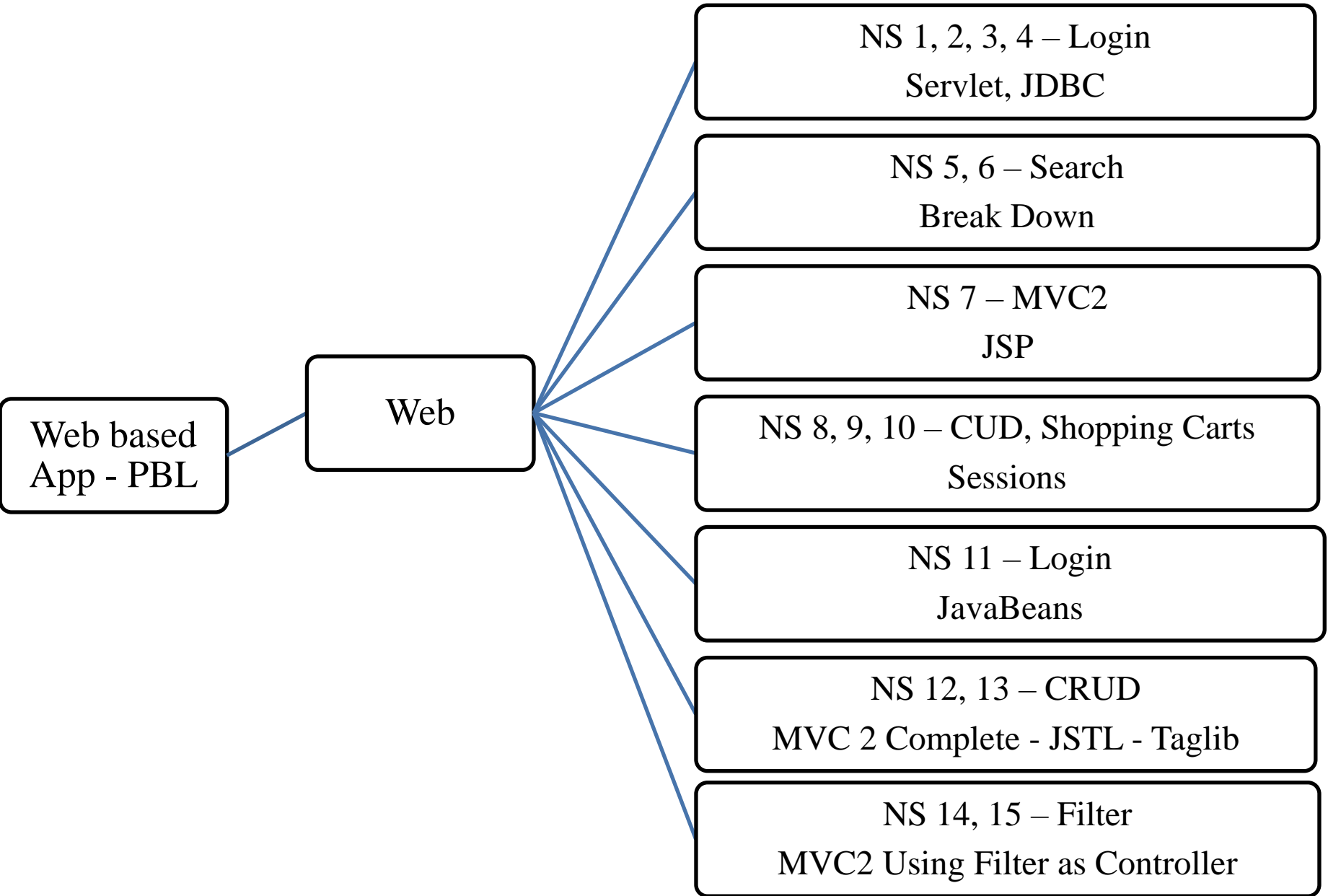
Java Authentication and Authorization Service

# Course Description

- 1. Servlets Model**
- 2. Web Application Interacting with Database**
- 3. Web Application & Web Container**
- 4. Java Server Pages (JSP)**
- 5. Session Management**
- 6. JavaBeans**
- 7. JSP Tag Libraries – Custom Tags**
- 8. Filters**

# Course Plan

- See course plan on LMS



# Course Plan

- See course plan on LMS
  - Agenda
1. The Servlet Model
  2. Web Application Interacting with Database
  3. Web Application & Web Container
  4. Java Server Pages (JSP)
  5. Session Management, Session & Listener (self-study)
  6. JavaBeans
  7. JSP Tag Libraries – Custom Tags
  8. Filters
  9. *Practical test*
  10. *Project Presentation*

# Materials/ References

- **Required Textbook**

- Online Text book: Nicholas S. Williams, 2014, Professional Java® for Web Applications, Wrox Press
- <http://library.books24x7.com/toc.aspx?bookid=62587>

- **Required References**

- <http://java.sun.com/docs/books/tutorial/jdbc/>
- [https://docs.oracle.com/cd/B14099\\_19/web.1012/b14017/filters.htm](https://docs.oracle.com/cd/B14099_19/web.1012/b14017/filters.htm)

- **References**

- <http://java.sun.com/>
- Fan Page: <https://www.facebook.com/TrongKhanh.Kieu/>

# Learning Environments

- JDK 8 (<https://www.oracle.com/java/technologies/downloads/>)
  - **Recommend:** JDK 8 Update 172/321  
(<https://www.oracle.com/java/technologies/javase/javase8u211-later-archive-downloads.html>)
- JDK 8 Documentation
- J2EE 1.4/JavaEE7 Core Patterns
- **NetBeans IDE 8.2/13 with supporting JavaEE**  
(<https://netbeans.apache.org/> )
- **Bundle Tomcat 8.0.27/ 10.0.18** (<http://tomcat.apache.org/> )
- **DBMS: MS. SQL Server 2017/2019**  
(<http://www.microsoft.com/sqlserver>)
- **Browser: Internet Explorer  $\geq$  8.x**
- **Driver Type 4/6 for MS. SQL Server:**  
**sqlserver4.jar/sqlserver42.jar**  
(<http://lms-undergrad.fpt.edu.vn/mod/resource/view.php?id=4566> )
- **Team Viewer for supporting**

# Course Rules

- **How to conduct**
  - Prepare contents of the next session/ topic at home
  - **Following lessons in classrooms (take note yourself to understand) and *review previous session in lecture on class every day (penalty marks on workshops)***
  - Completing chapter assessments in time and Quizzes (via LMS)
- **Communication**
  - Class
  - Interchange by FU-HCM LMS, Forum
  - Discussing actively in your teams and in classrooms
  - Free to question and answer
- **Others**
  - Off phone
  - Use laptops under teacher's instruction (No game, no chat in class)



# Evaluation Strategy

- Must attend more than 80% of contact hours (if not, not allow to take exam).
- **Evaluating**
  - **02 Progress Test (Q)** 10 %
  - **02 Workshop (Lab)** 10 %
  - **01 Assignment (Prj)** 40 %
  - **01 Practical Exam (P)** 20 %
  - **Final Exam (FE)** 20 %
- **Total score**
  - 10% (Q) + 10% (Lab) + 40% (Prj) + 20% (P) + 20% (FE)
- **Pass**
  - Total score  $\geq 5$  and Final Examination  $\geq 4$  (of 10)
  - **Every components > 0**
- **Retake** only the Final Exam when not passed

# How to study

- This course is **complex knowledge** (however, it's **attractive and exciting**), so you need to keep tight grip on it
  - **Read**
    - On the books to get the general concept
    - Reference, study, collection from internet, your classmates, forum ...
  - **Attend lectures**
    - Listens, understand, then make your own notes
    - Give your explanation about some topic in lectures
    - Ask questions
    - Give some examples that are not existed in your book
    - Practice all the exercises, demo to make your sense
  - **After classes**
    - Discuss your classmate in directly, on forum
    - Do the lab, assignments to submit via CMS, and do more exercises
    - Build your teams in yourselves to support together in studying

→ **Make question to deposit marks for components in progress**

# How to exam/test

- This course is **required** following rules, so you **need to focus and practice** your exercises and homework in try your best everyday (**meet deadline and submit on LMS**)
  - **Progress Tests (if students violate, they take 0 marks)**
    - No books and No conversations
  - **Practical Exam (if students violate, they take 0 marks)**
    - **Obey requirements and lecturer's recommended in his lecture**
    - No internets, No emails, No chats, No conversation
    - Not copy or paste from available/previous code.
    - All are try it yourselves **manually**
    - ... Nothing else
    - You do only work with Netbeans IDE tools and DBMS. (**without configuring svn**)
  - **Workshop/Assignment (if students violate, they take 0 marks)**
    - **Obey requirements and lecturer's recommended in his lecture**
    - Not copy (copy code, contents, style)
    - Submission of all source code **does not delete anything**

# Academic policy

- Cheating, plagiarism and breach of copyright are serious offenses under this Policy.
  - **Cheating**
    - Cheating during a test, making a project, or an exam is construed as talking, peeking at another student's paper, or any other clandestine method of transmitting information.
    - Cheating during in making lab and assignment as copy source code, copy style, same meaning in progress, ...
    - **To verify, student's code in project can be deleted, then he/she must be retyped to make the program running correctly. Or/And make new required functions (Otherwise, he/she takes 0 marks for his/her project/quiz)**
  - **Plagiarism**
    - Plagiarism is using the work of others without citing it; that is, holding the work of others out as your own work.
    - **To verify, student must be described functionality dataflow and how it work (Otherwise, he/she takes 0 marks for his/her project)**
  - **Breach of Copyright**
    - If you photocopy a textbook without the copyright holder's permission, you violate copyright law.

# Enjoy the Course

- Be enthusiastic about the material because it is interesting, useful and an important part of your training as a software engineer.
- Our job is to help you learn and enjoy the experience.
- We will do our best but we need your help.
- So, let's all have fun together with Web-Based Java Applications!!!

Q & A