COS30041 Creating Secure and Scalable Software

Lecture 01B – Introduction to Java EE

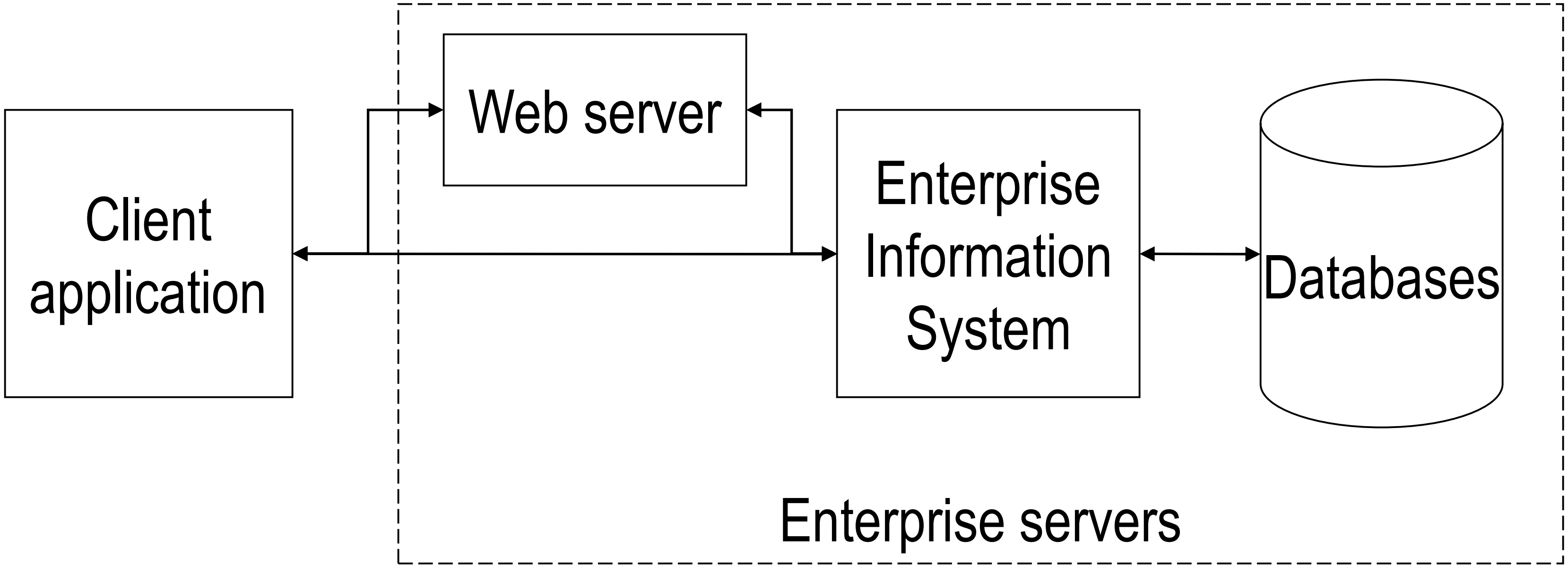
SWINBURNE UNIVERSITY OF TECHNOLOGY

Learning Objectives

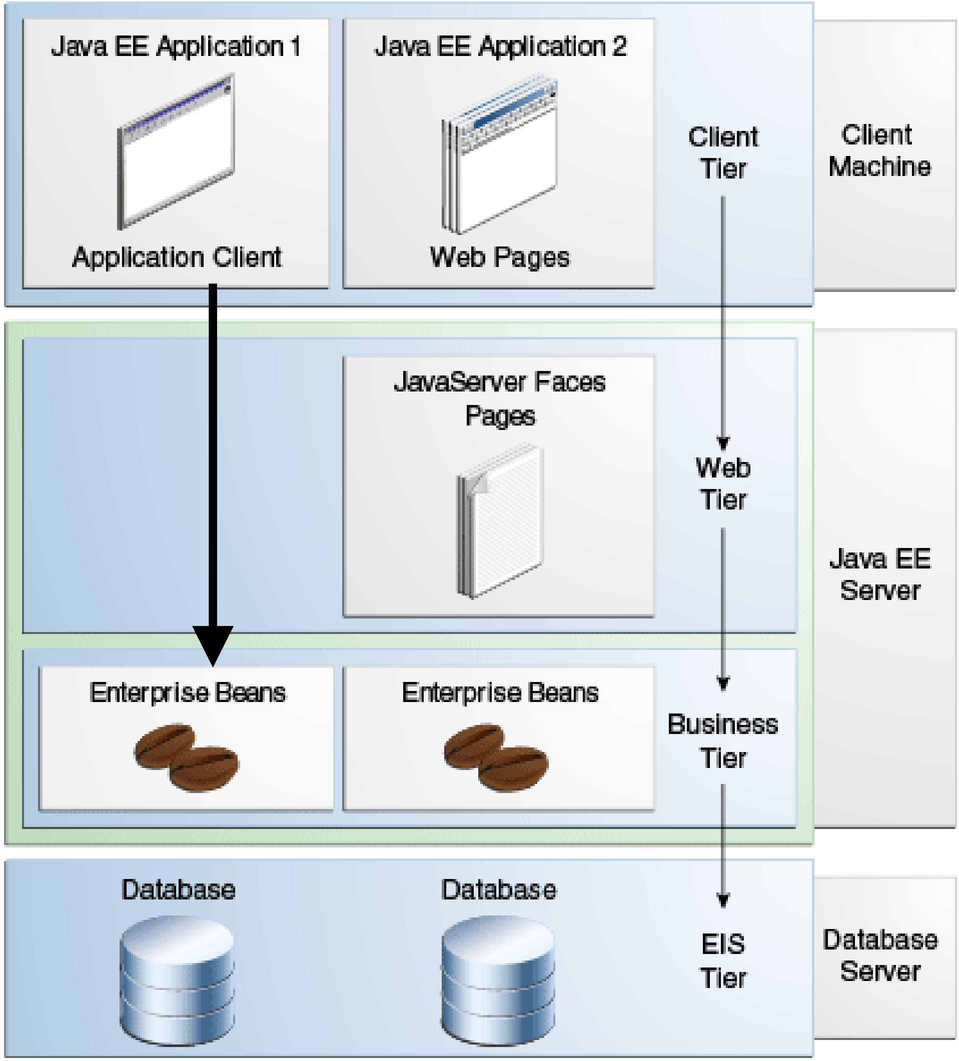
- After studying the lecture material, you will be able to
  - □ Understand and describe what Java EE is
  - □ Explain why you want to use Java EE to develop your Enterprise application
  - □ Explain the 3-tier and 4-tier architecture frameworks of Java EE
  - □ Explain why there is a need to separate the following tiers
    - □ Presentation tier
    - □ Business tier
    - □ Data tier

Java Enterprise Edition, Java EE

- Collection of APIs and specifications
  - □ for developing, deploying, and managing multi-tier server-centric applications
- Large-scale business systems
- Distributed environment



Java EE – A Framework (3-Tier / 4-Tier)



Note: Adapted from Fig.1-1 from [JEE7T]

Java EE – Major challenges

- Scalability & Performance
- Security
- Integration
- Development Productivity
- Maintenance
- Portability

Background

- A programming model for building enterprise systems using Java
- Use with different underlying vendor implementations of an enterprise system infrastructure
- Various Java EE or J2EE
  - □ Java EE 7 (Jun 2013)
  - □ Java EE 6 (Dec 2009) – EJB version 3.1 [*still to find suitable free ebook*]
  - □ Java EE 5 (2006) – EJB version 3.0 [free ebook is available]
  - □ J2EE 1.4 (2003) – EJB version 2.1
  - □ J2EE 1.3 (2002) – EJB version 2.0

Java EE Technologies

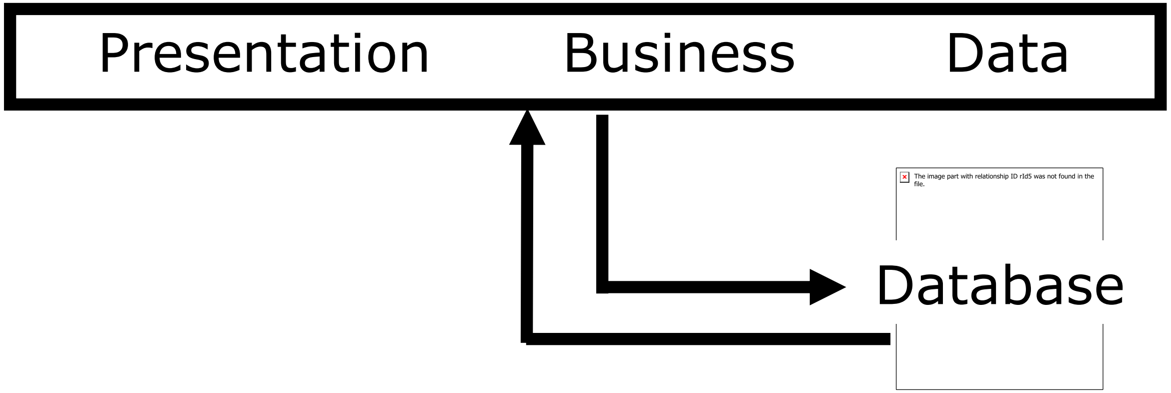
- For Web Tier – Handling web presentation
  - JavaServer Faces
- For Business Tier – Handling business logic
  - Session Bean* (an Enterprise JavaBean, EJB)
  - Message Driven Bean*, MDB (an EJB)
  - Java Message Services*, JMS [used by MDB]

Java EE Technologies (cont'd)

- For Data Tier – Handling data storage (persistence)
  - Java Persistence API* (starting from Java EE 5.0, EJB 3.0)
  - Entity Bean* (an EJB) (J2EE 1.4 or earlier)
- For communication infrastructure
  - Java RMI over IIOP (RMI–IIOP)* - underlying technology for remote objects
  - Java DataBase Connectivity (JDBC)* - underlying technology for database updates
- …

Enterprise Architecture – Good old days

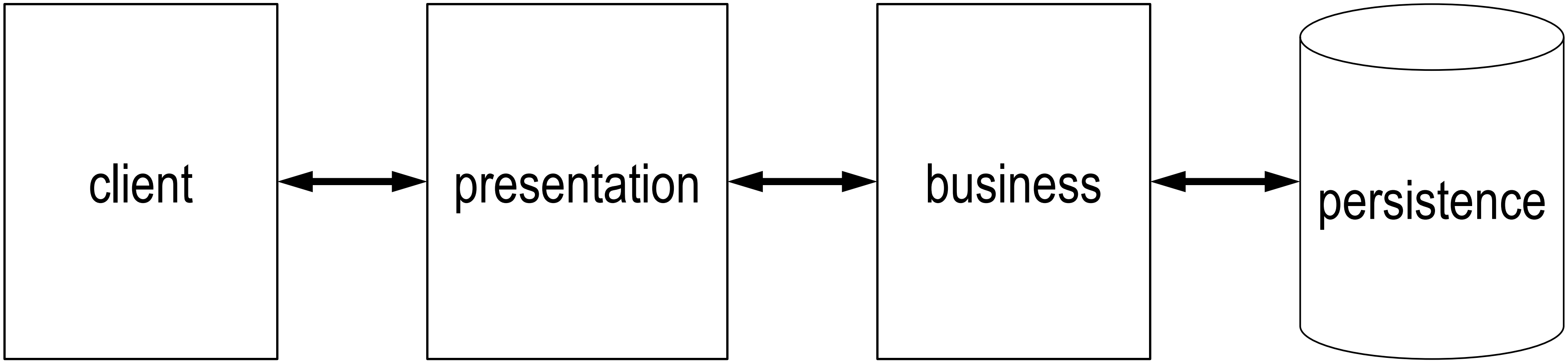
- Use monolithic architectures and ***thick clients***
  - □ installation & deployment issues
  - □ difficult to modify


Presentation
Business
Data
The image part with relationship ID rId5 was not found in the file.
Database

© Swinburne University of Technology

Enterprise Architecture – Modern style

- Use different tiers to hold different responsibilities
- Tiers use components to increase modularity
  - □ Developer productivity
  - □ Code reuse
  - □ Maintenance



Enterprise Architecture (cont'd)

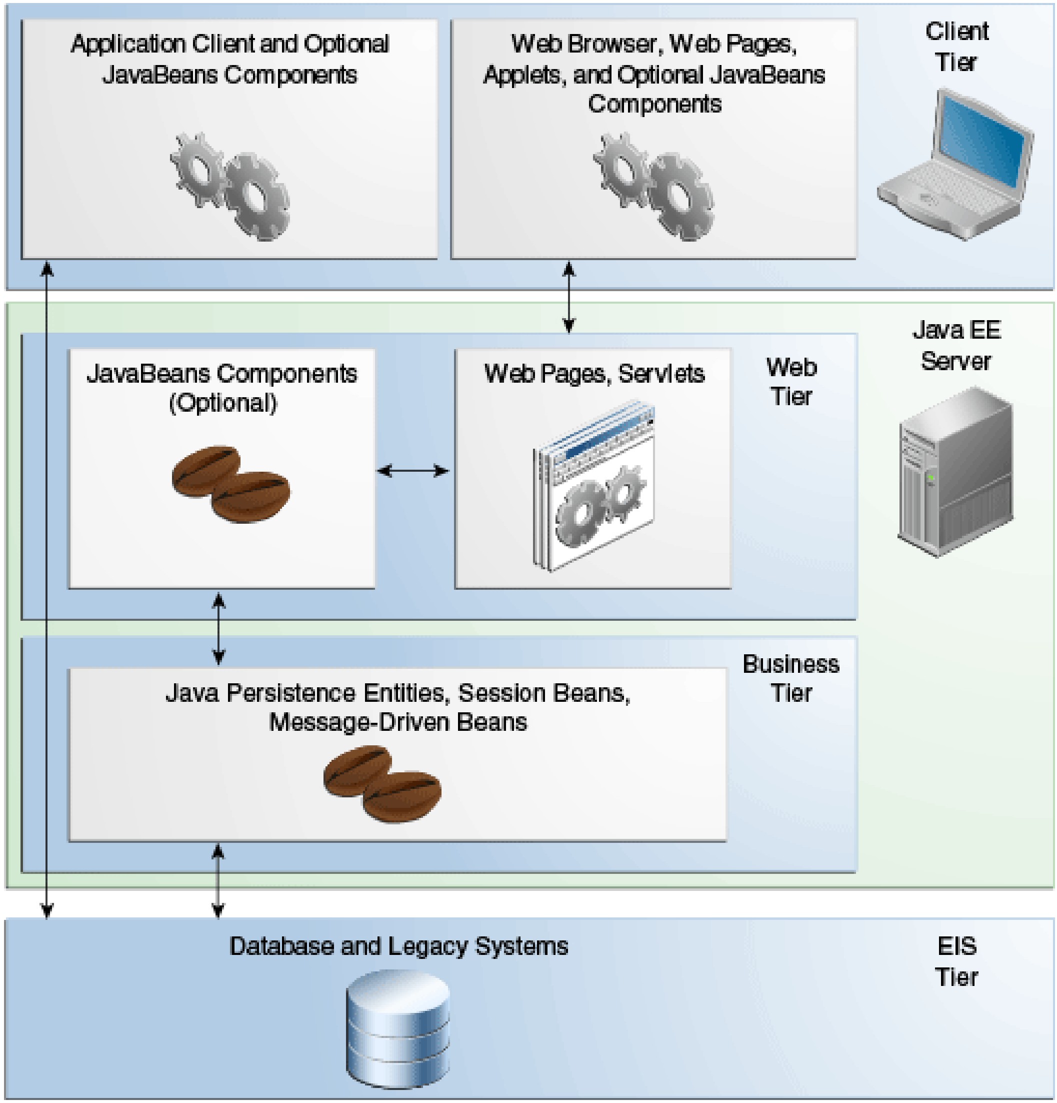
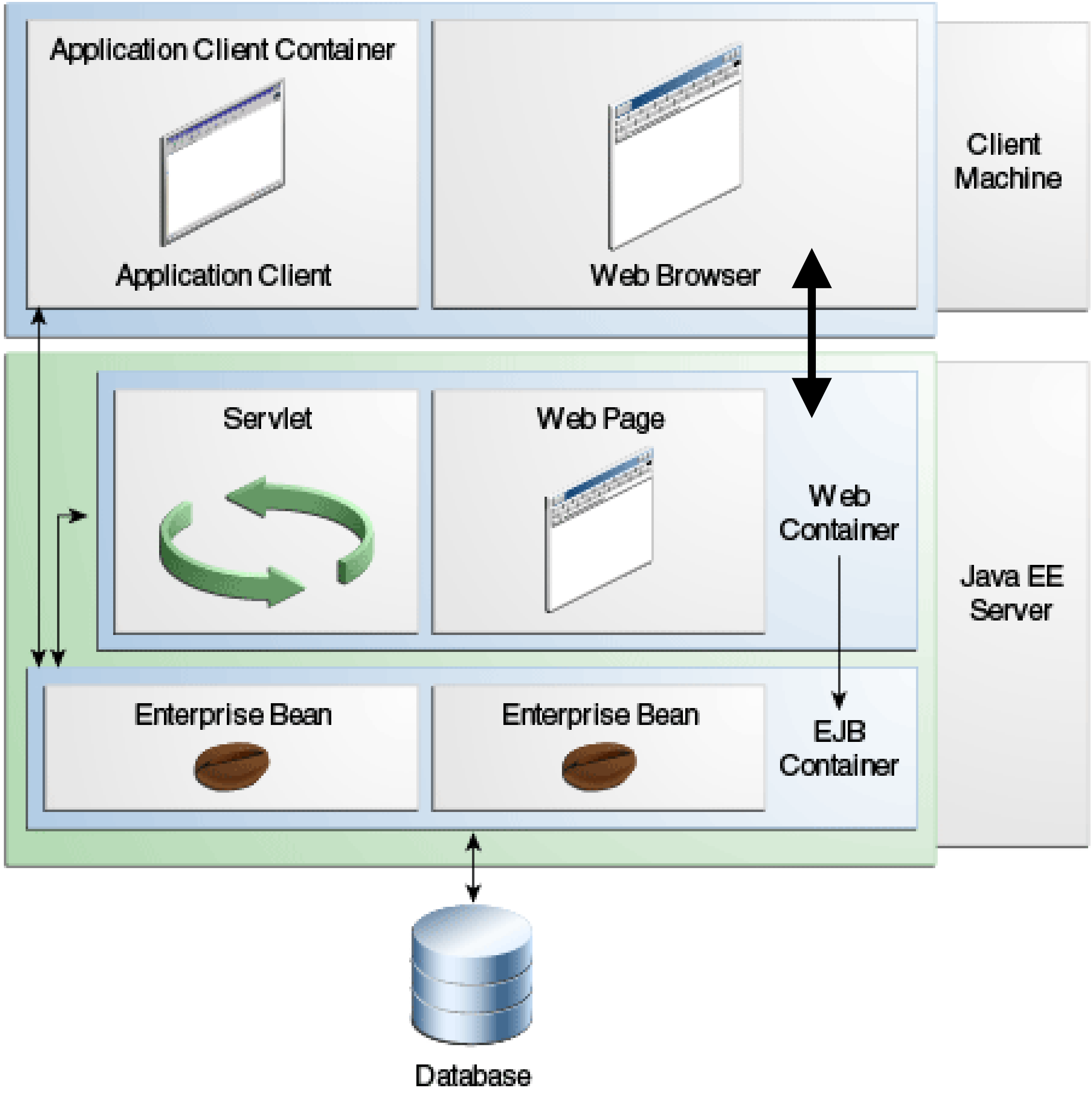

Application Client and Optional JavaBeans Components
Web Browser, Web Pages, Applets, and Optional JavaBeans Components
Client Tier
Java EE Server
JavaBeans Components (Optional)
Web Pages, Servlets
Web Tier
Business Tier
Java Persistence Entities, Session Beans, Message-Driven Beans
Database and Legacy Systems
EIS Tier

Fig.1-4 from [JEE7T]

Enterprise Architecture (cont'd)



Note: Adapted from Fig.1-5 from [JEE7T]

Components & Containers

Component

- An application level software unit

Container

- A system level “entity” that provides runtime support of all Java EE components
- It controls and runs the components inside it

Components & Containers (cont'd)

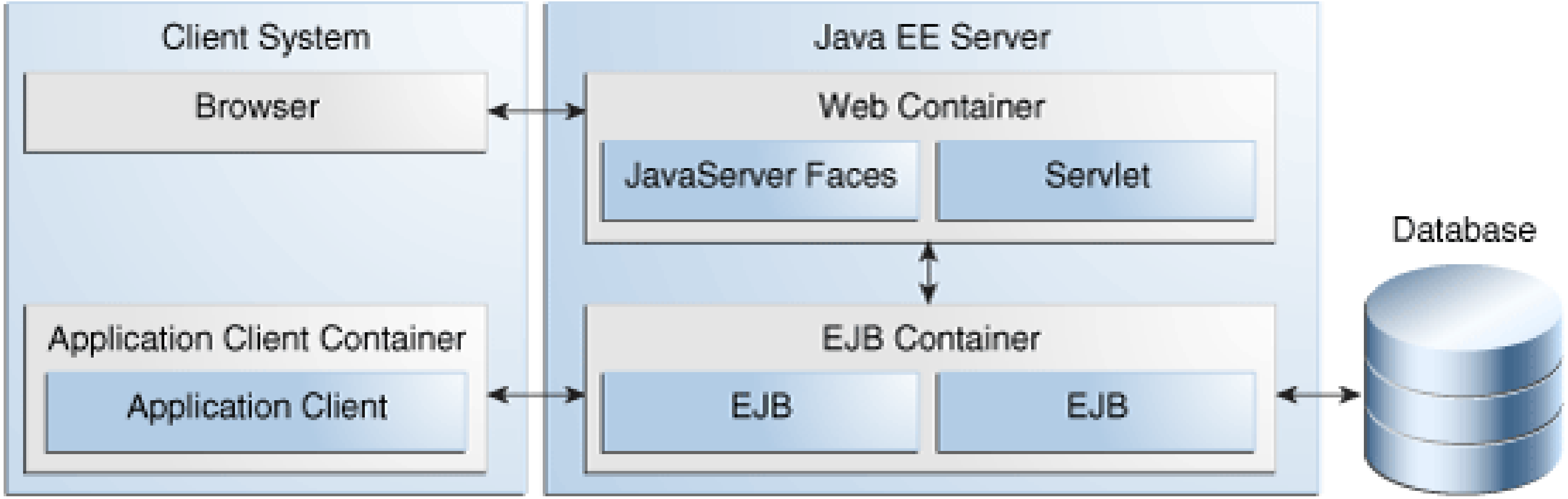


Fig.1-6 from [JEE7T]

References

- [JEE7T] E. Jendrock et al. (2014) *The Java EE 7 Tutorial*, Oracle – Chapter 1 Overview
  - https://docs.oracle.com/javaee/7/tutorial/index.html
- *Designing Enterprise Applications with the J2EE Platform*
  - http://java.sun.com/blueprints/guidelines/designing_enterprise_applications_2e/DEA2eTOC.html