

JWelder

Thomas Neumann
Portals & Marketplaces



JWelder

The JWelder Use Case Architecture

Design

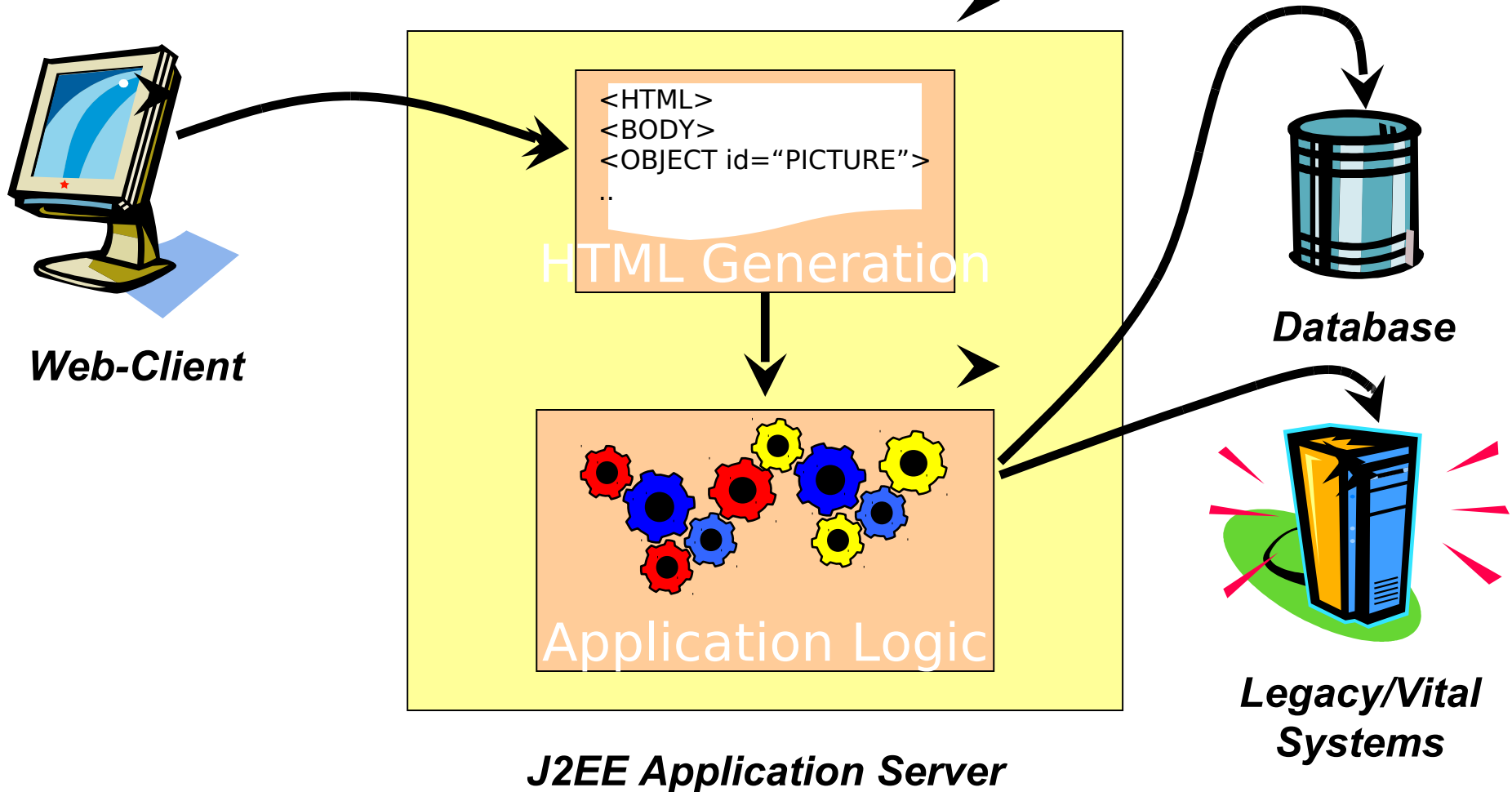
Using JWelder

Experiences & Future Enhancements



The JWelder Use Case

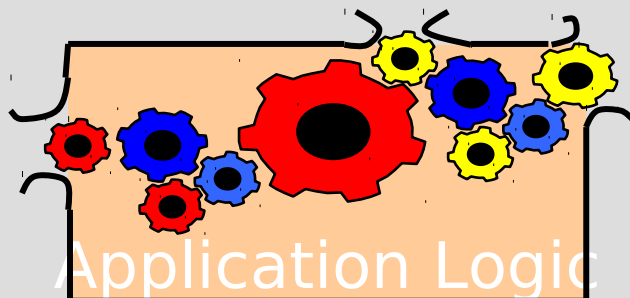
JWelder Sample Scenario



Problems With

EJB Nets

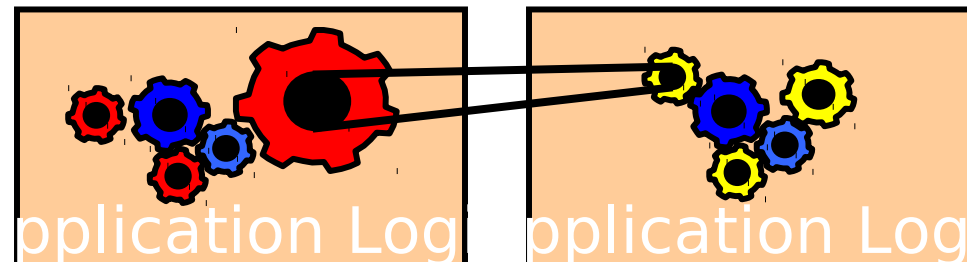
- Problems
 - Strong dependency
 - Heavy communication
- Drawbacks
 - Makes the evolution of the system harder
 - Makes it less scaleable
 - Makes it hard to test



Solution to the EJB

Nets

- Provide the system into Business Components
- Provide different communication models for interaction between Business Components



What is JWelder good for?

JWelder provides a software architecture for EJB based systems

- A system is divided into Business Components (subsystems)
 - Business Components are loosely coupled with a software bus

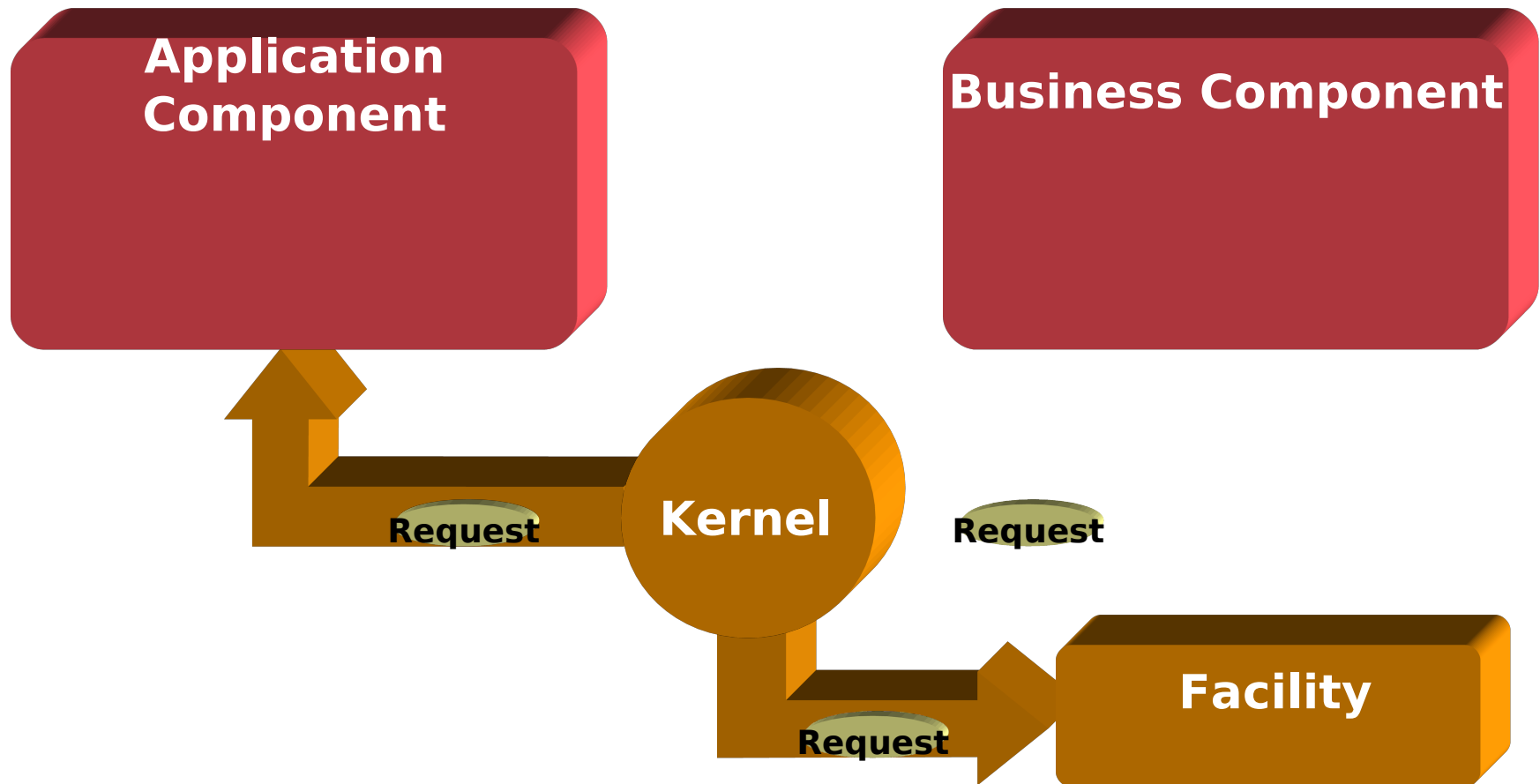
JWelder provides Frameworks to easily build Business Components based on EJBs

- JWelder provides all the enabling and infrastructure for EJB development
- These base classes enable rapid and fast development of EJB based Business Components

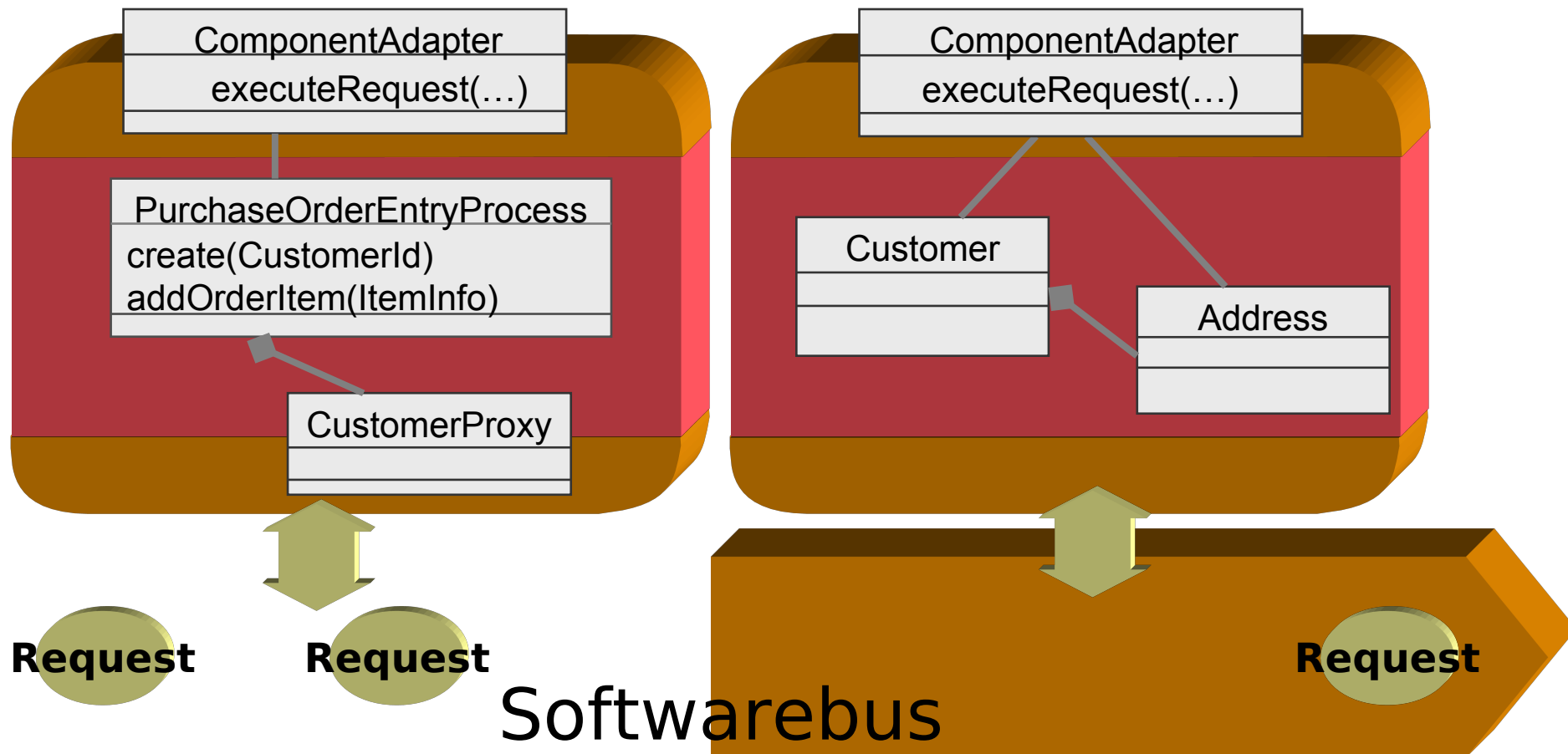


Architecture

Logical Overview

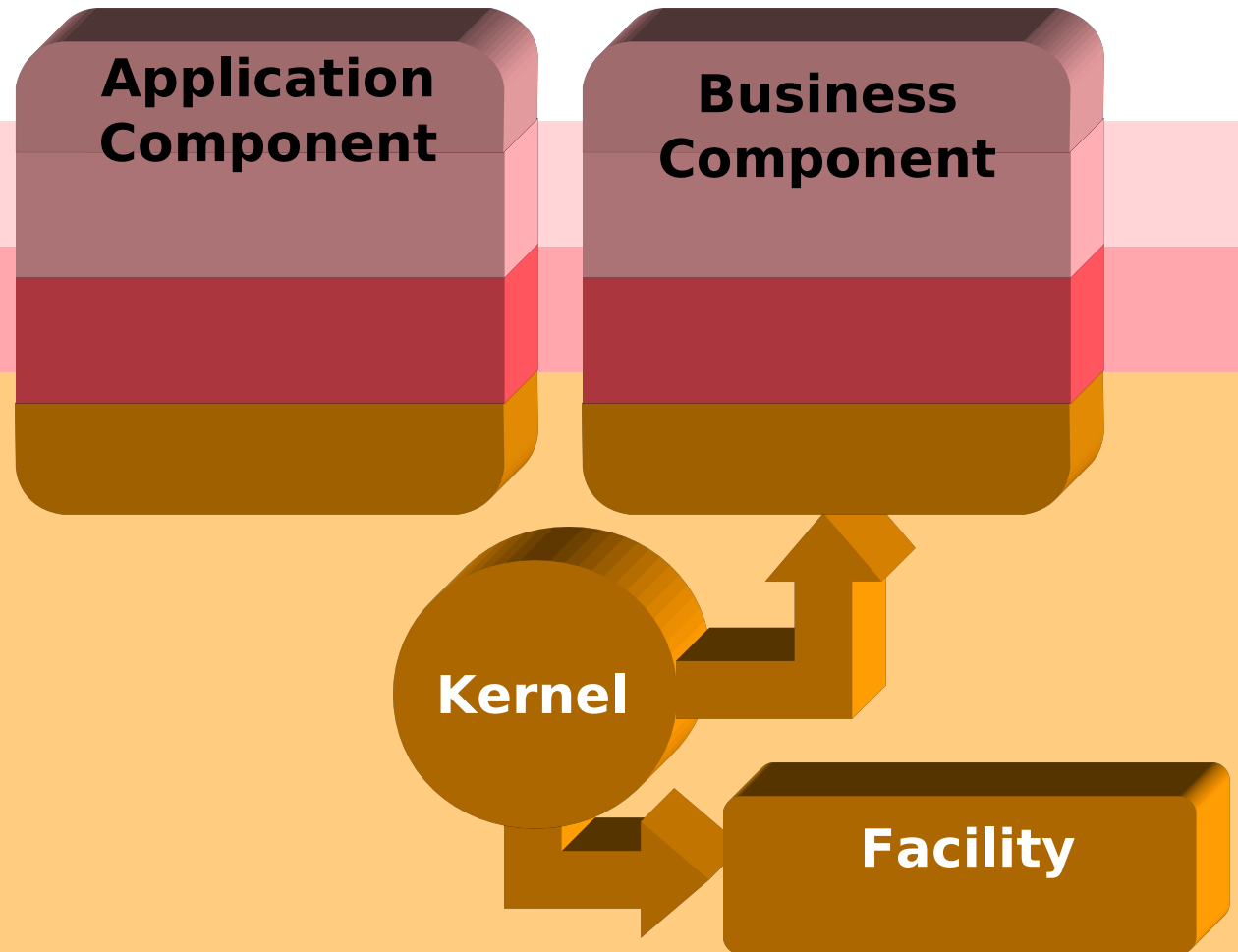


Connecting Components with a Softwarebus



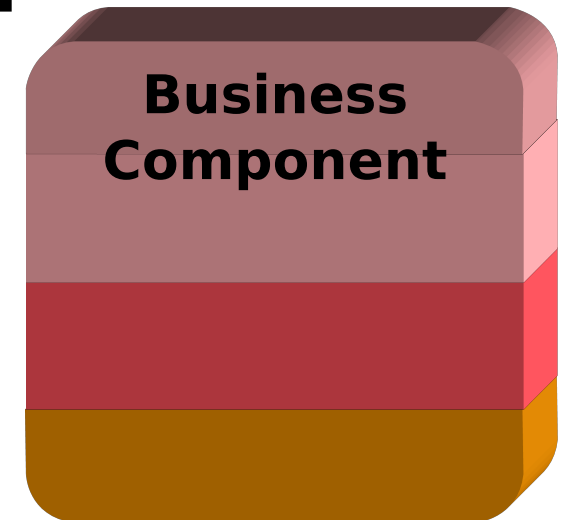
Implementation Overview

Business
Component
Business
Component
Framework
Business
Component
System



Business Component

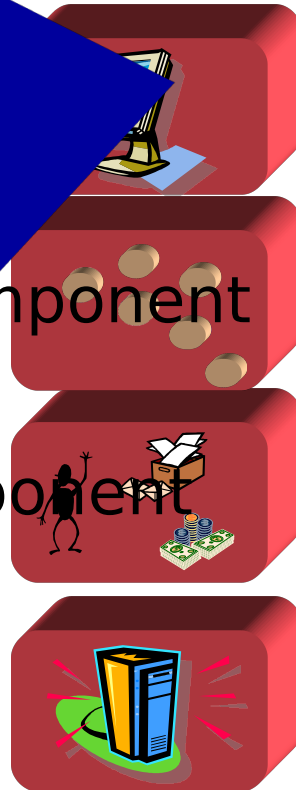
- One Business Component covers one subsystem
- Each Business Component belongs to a category
- Business Components are interconnected via the software bus
- Each Business Component is made of one to many EJBs
- EJBs may be custom developed or bought from third parties



Of Business

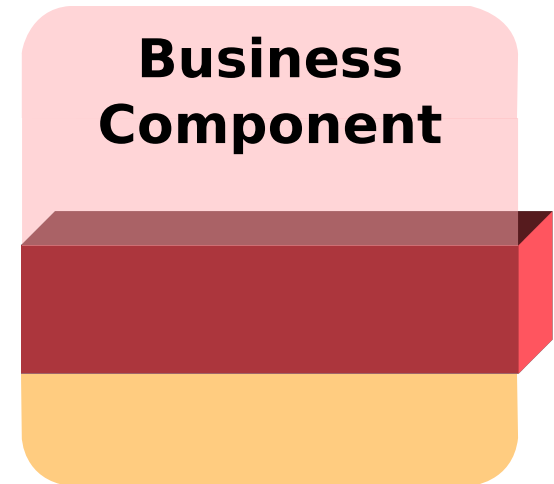
Component

- Application Component
- Business Process Component
- Business Entity Component
- Adapter Component



Business Component Framework

- Each category may be supported by a Business Component Frameworks
- List of categories is not finished - there may be more categories in the future.



Business Component System

- Connecting Business Components
 - Integrate different Business Components to one system
 - Provide infrastructure services
 - Implement different communication models

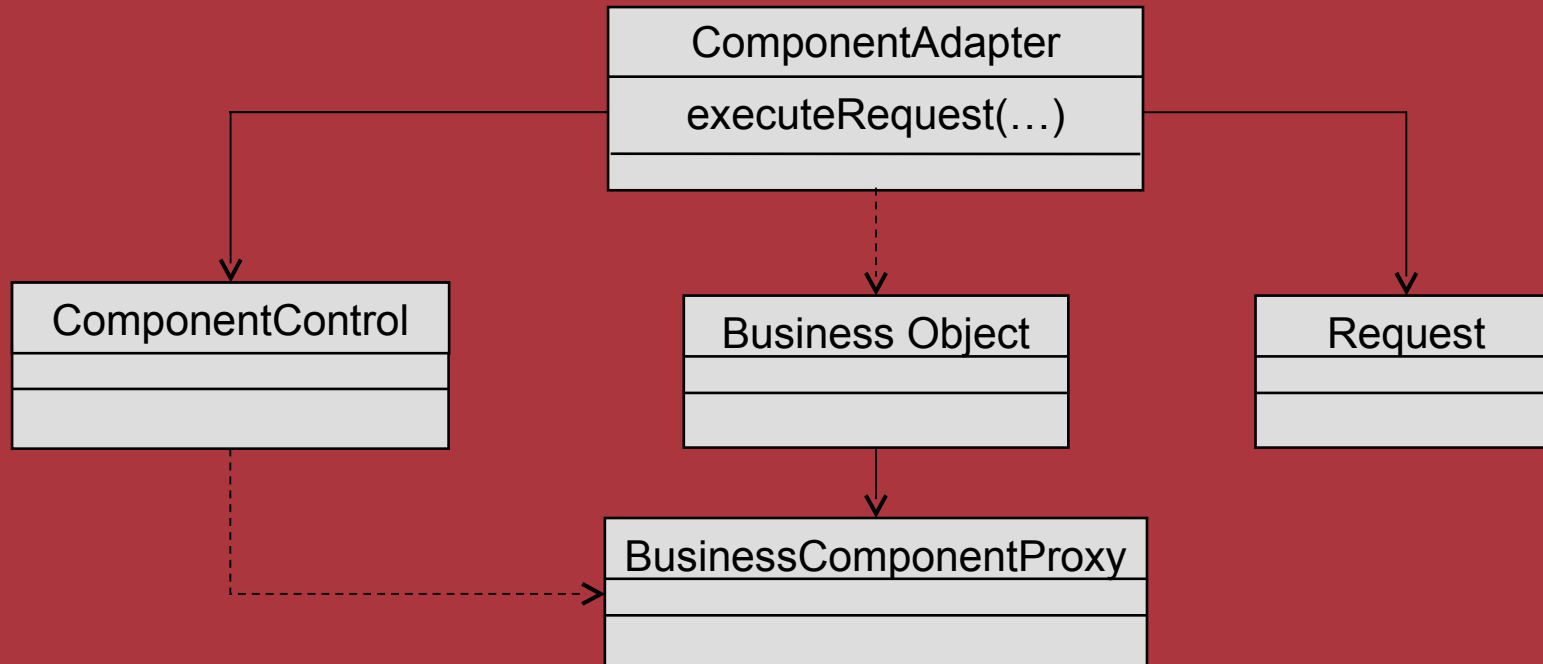




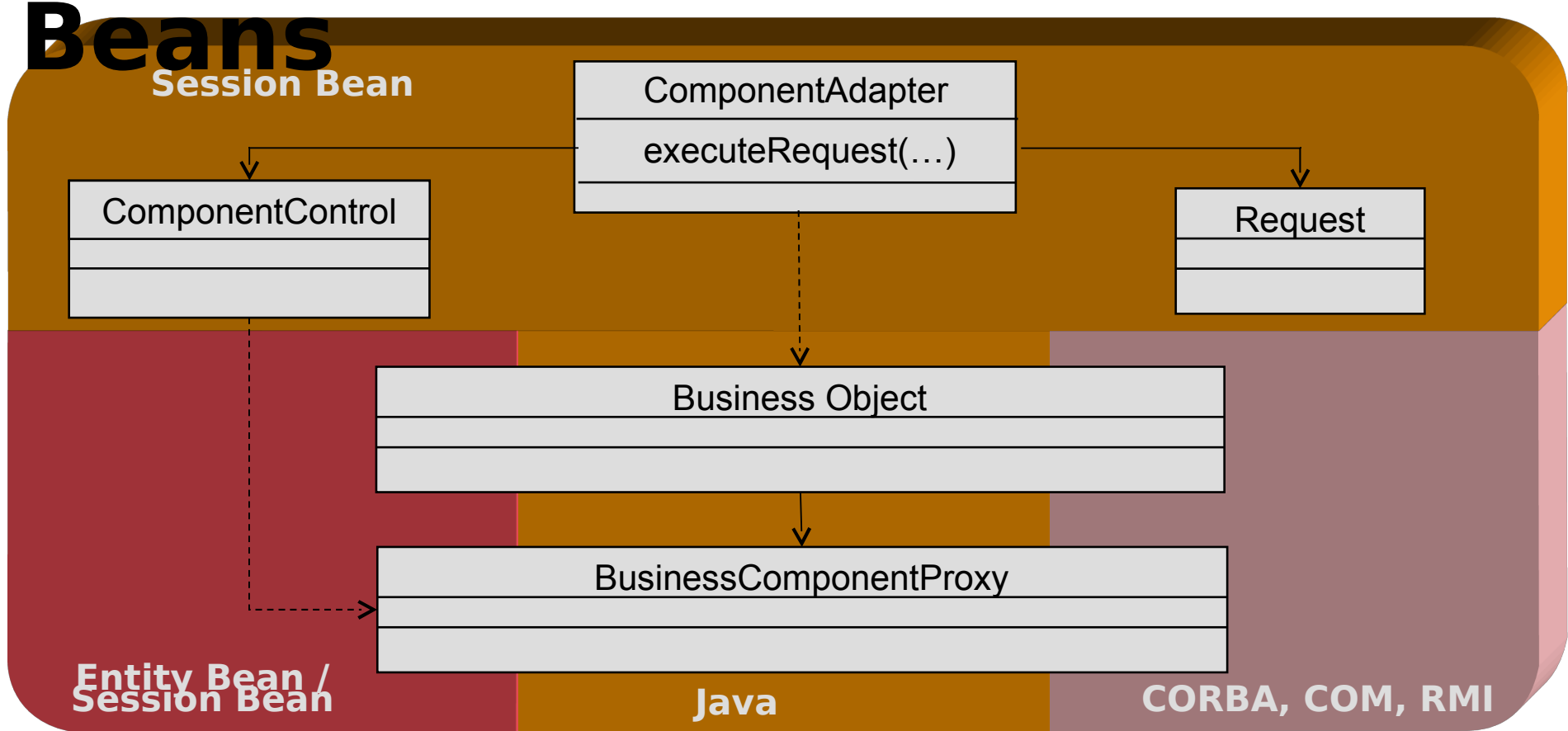
Design

First Class Objects of a Component

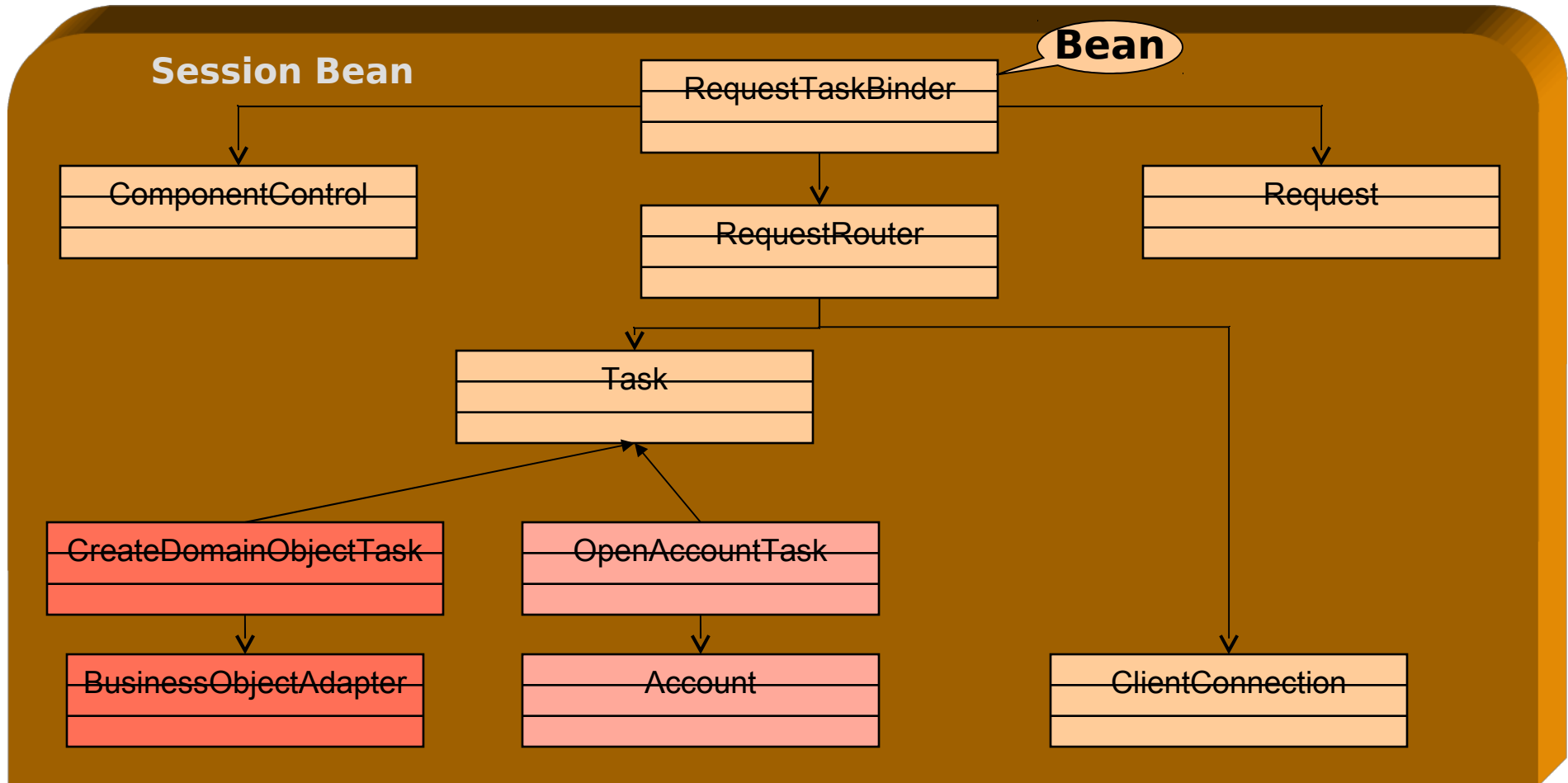
Business Component



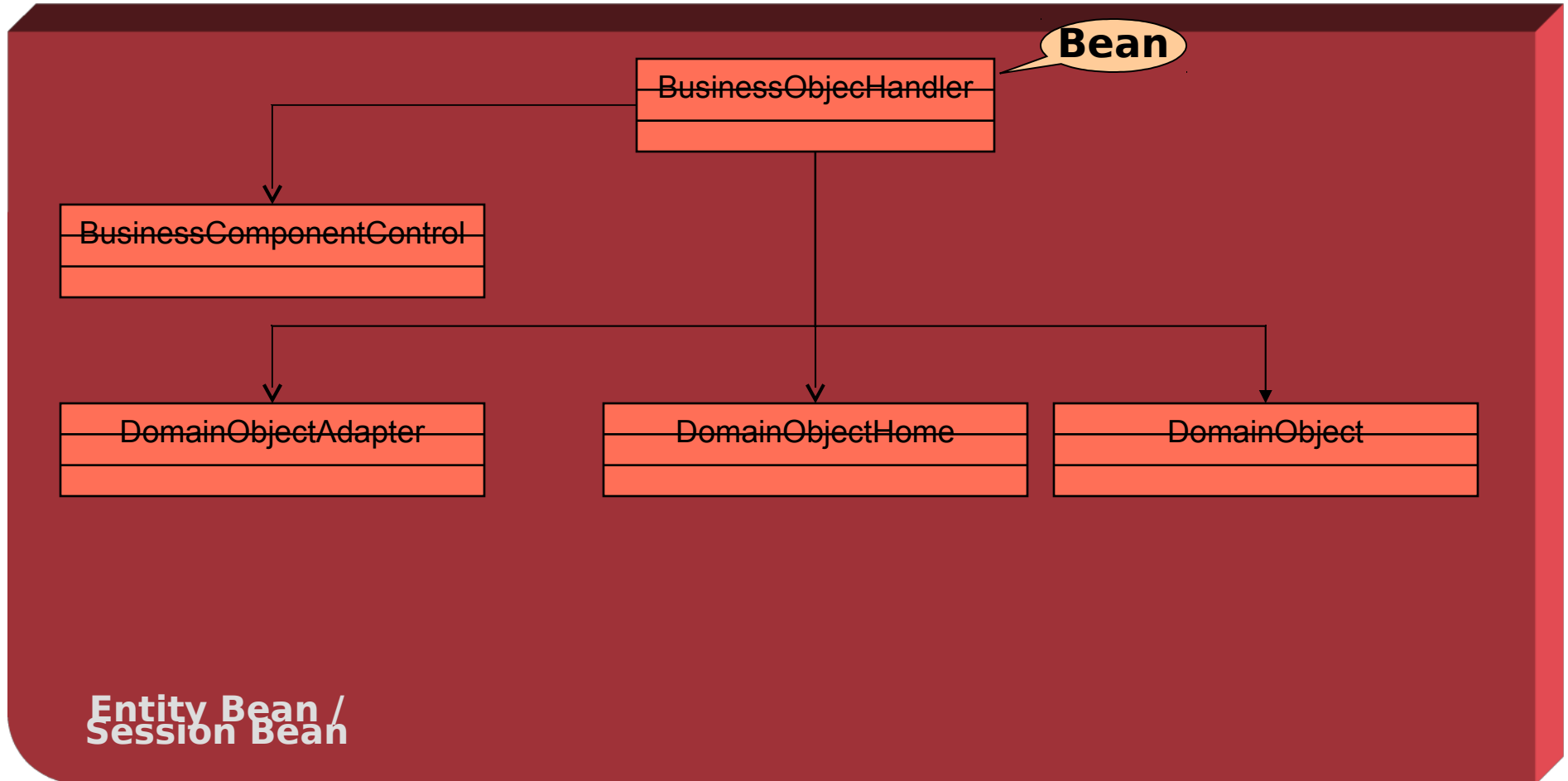
Mapping on Enterprise Java Beans



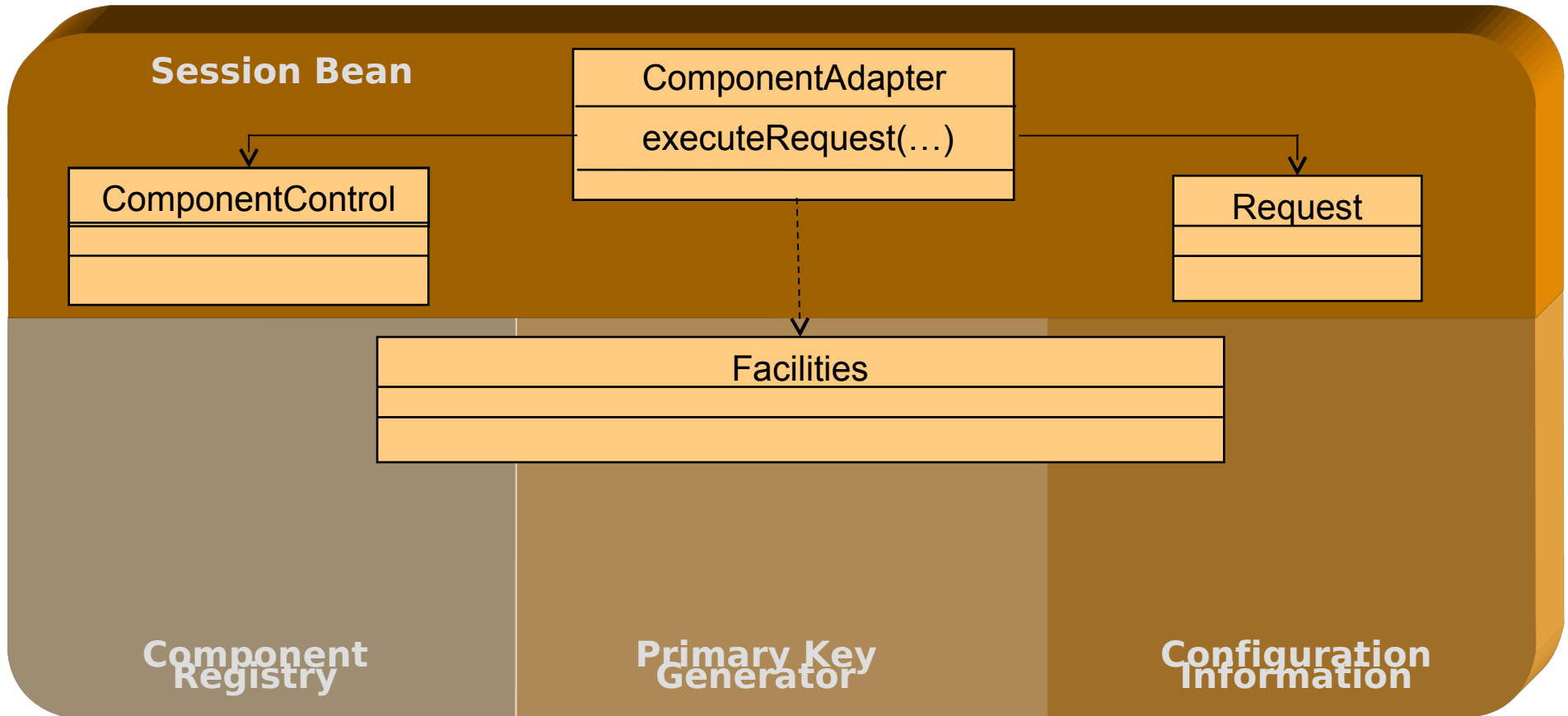
Component Adapter



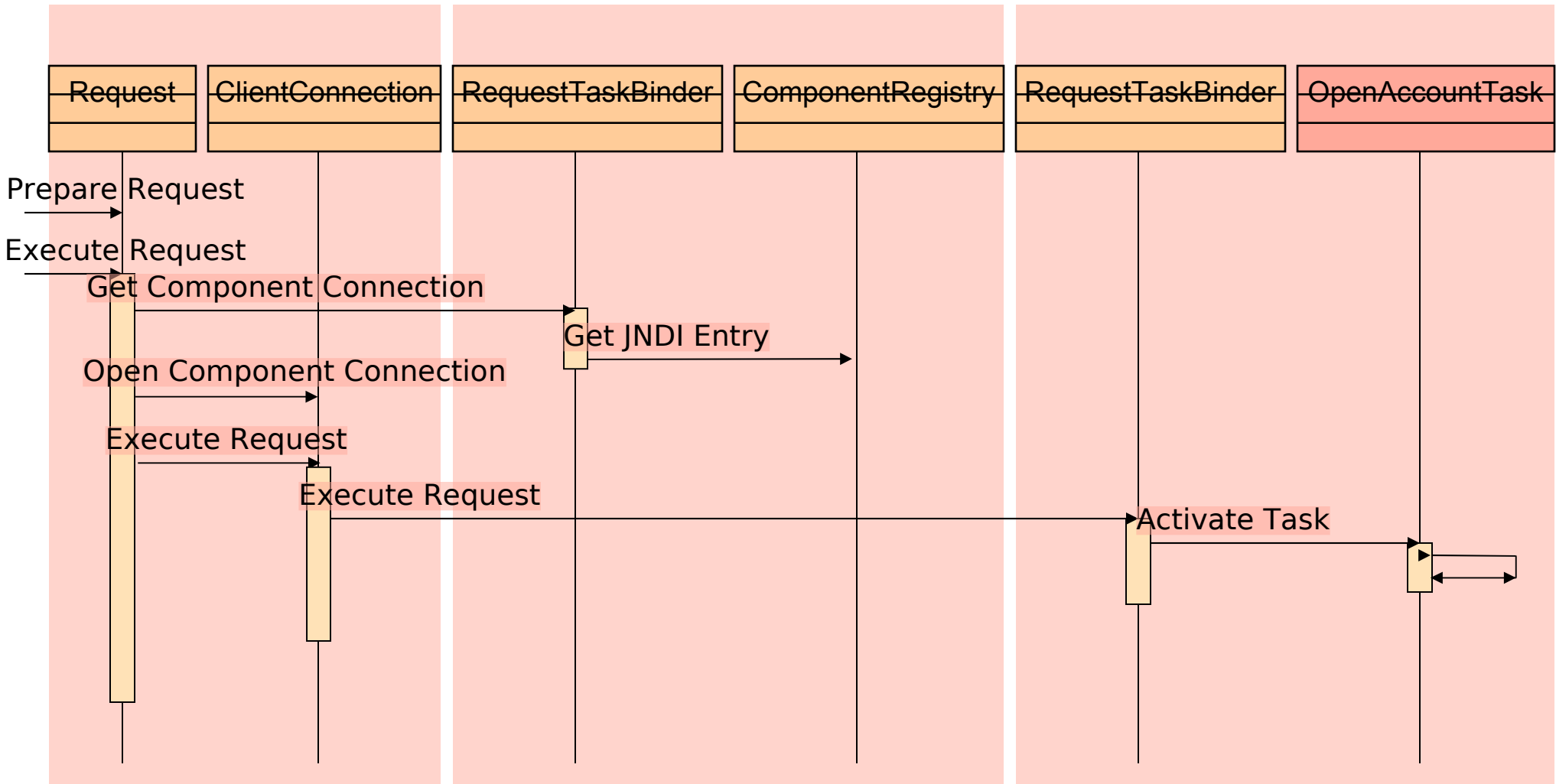
Business Object



Kernel Component



Execution of a Request



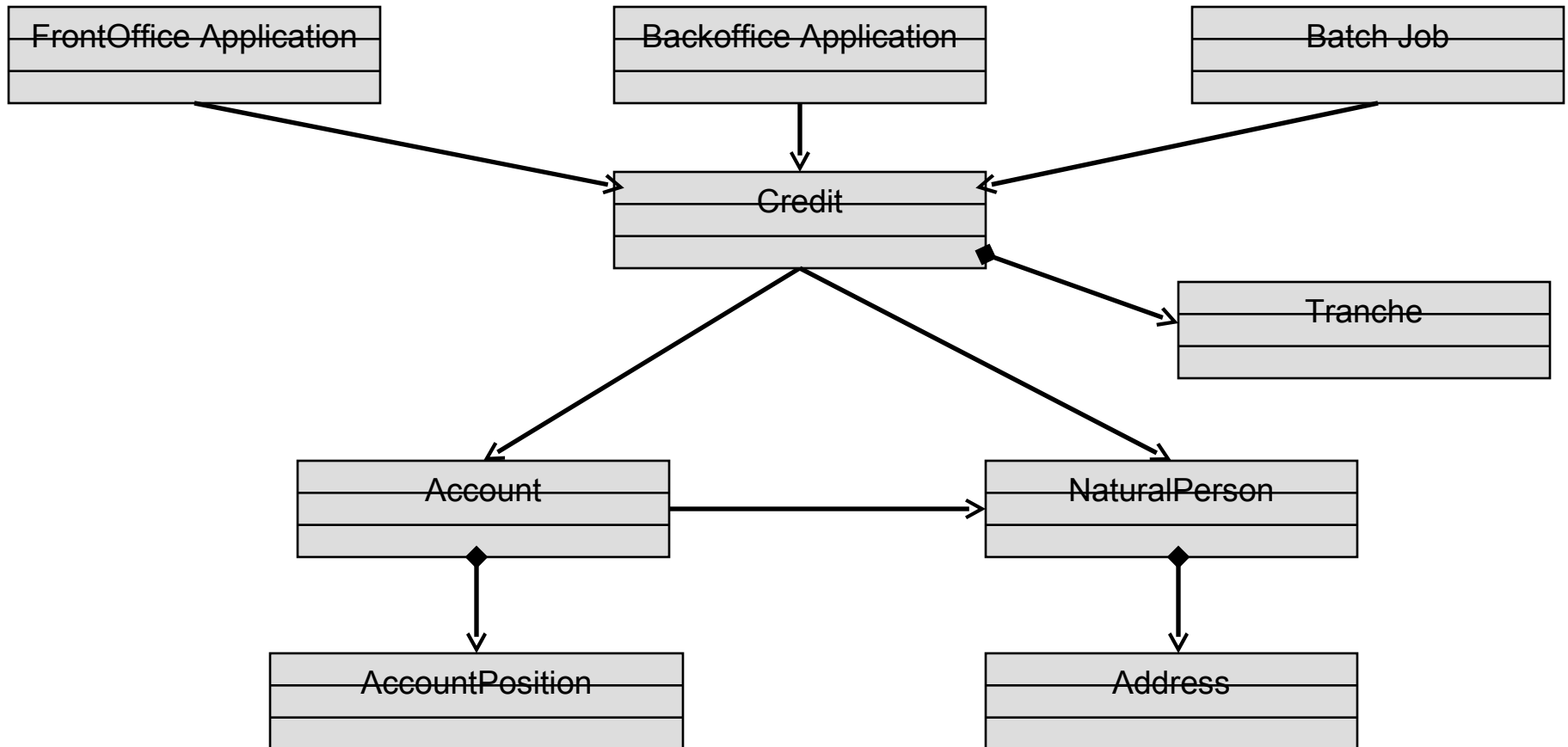
How to use JWelder?

- Identify autonomous business concepts / business components
 - Identify services provided by these business components
 - Identify categories of business components
 - Provide a framework for each category
 - Implement or reuse business component frameworks
 - Implement or reuse business components
 - integrate business components to the overall system.



Using JWelder

Sample System



Sample Component Based Application

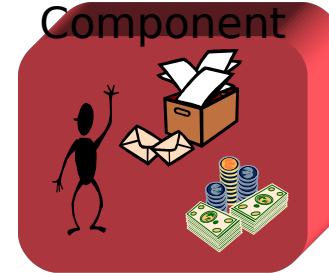
Front Office
Application-
C



Account
C



Credit
Component



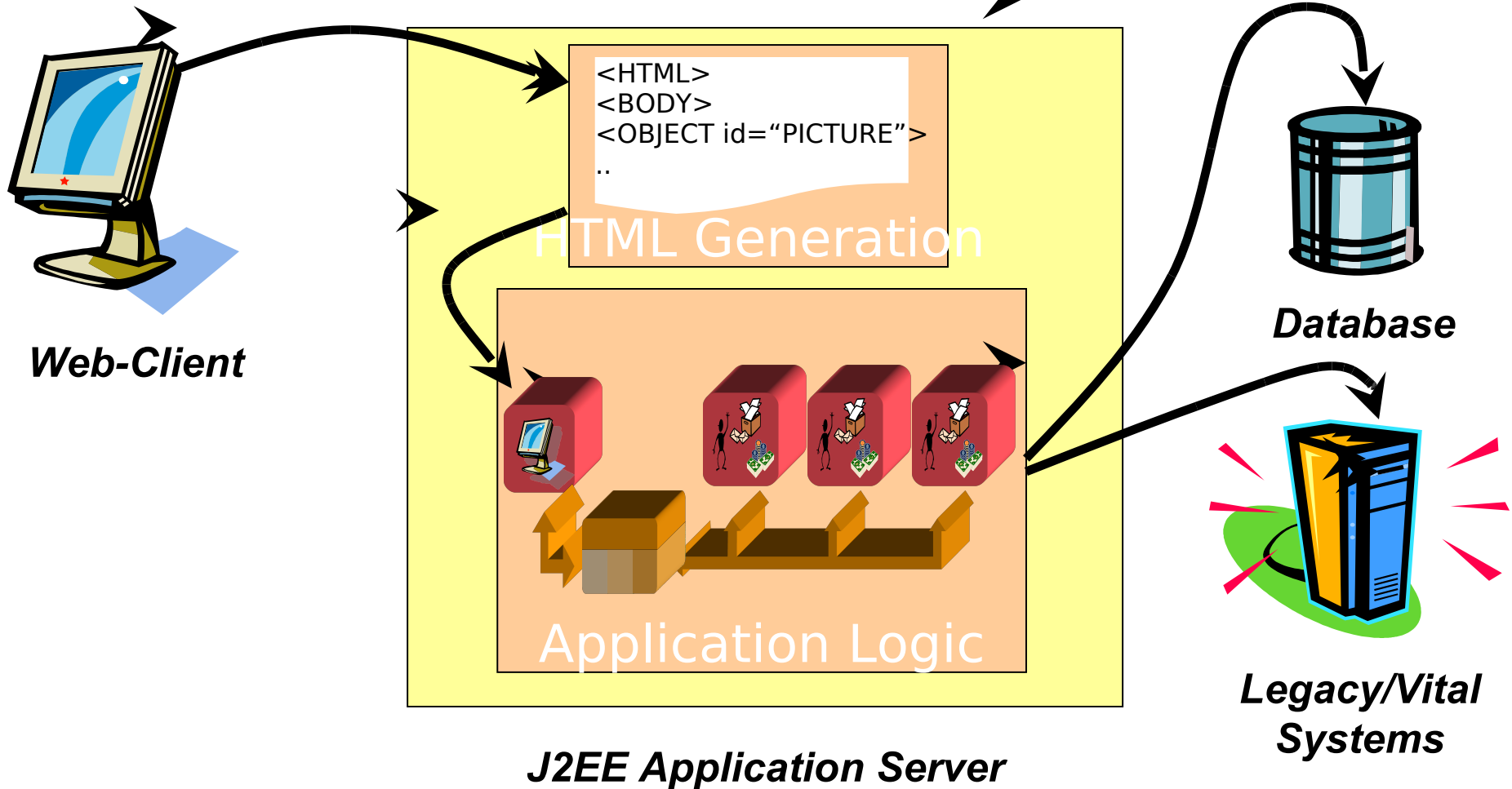
Client Component



Kernel



JWelder Sample



Experiences & Future Enhancements

Experiences Made

- Used Application Server
 - BEA Weblogic Server, IBM Websphere Advanced Edition, Gemstone
- Stress Test
 - Hardware: COMPAQ Notebook 250 MB Pentium II/ 300 MHz Hauptspeicher
 - Load: 100 concurrent HTTP Sessions, average response time 4 sec/HTTP Request
- Projects implemented with JWelder
 - Mobile Show Case
 - Insurance Show Case
 - Subitop Prototype

Future Enhancements

- Application Component Framework
- Publish & Subscribe Communication Model

Everythings Unclear - Further Questions ?

