北京郵電大學

分布计算环境

北京邮电大学计算机学院

北京郵電大學

Chapter 4

面向构件的分布计算环境

主要内容

- 基于构件的软件体系结构
- J2EE/Java EE
- EJB
- 轻量级框架和EJB3.0

基本概念

- 三个基本概念：
  - 框架 — 对问题的部分解决，是让用户集成构件的架构
  - 构件(组件) — 构件是软件的基本单元， 既足够小，以便于维护，又应足够大，以使之具有功能，可以被打包和使用
  - 对象总线 — 是一种机制，使得构件和框架能够调用分布式环境中的另一构件或框架的服务

主要内容

- 框架
- 构件
- 对象总线
- 构件模型

框架的引出

- 基于过程（函数）的应用程序：
  - 包含大量互相调用的过程
  - 将这些过程拼装在一起形成一个应用程序所需的逻辑关系被分散到应用程序的各个部分，不能由任一函数所把握
- 基于对象的应用程序——通过重用类库中的对象达到代码重用
  - 真正实现了信息隐藏和数据抽象的承诺
  - 各种应用程序分享细颗粒的对象（类库）
  - 对象仍然是被动的，仍然需要一种结构（控制流）将它们连接到一起

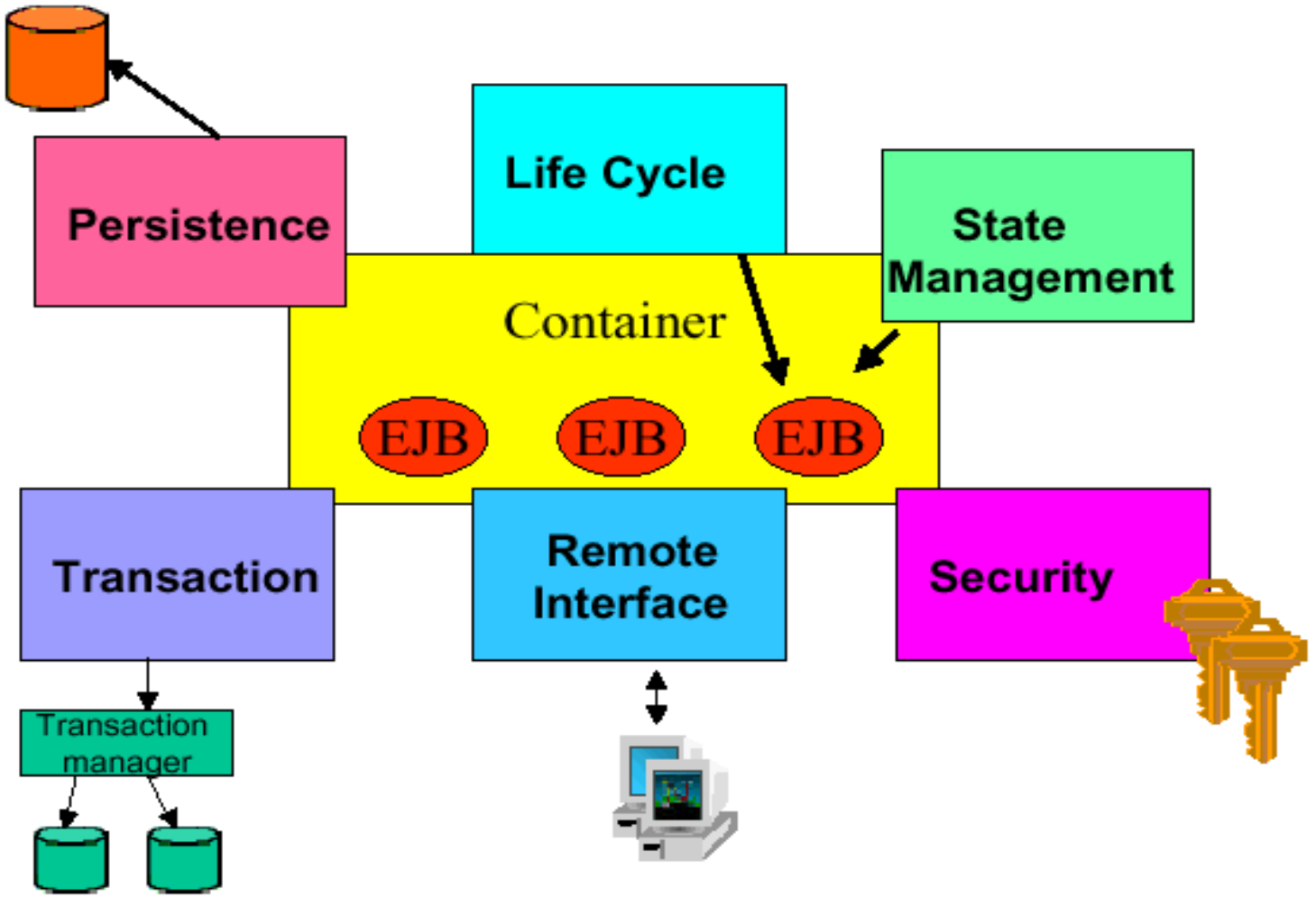
框架的引出（续）

- 问题：对象只构成应用程序的一部分，完全不能把握应用程序的结构（控制流）
- 过程和类库的固有缺点：被动性。它们都需要一种能提供控制流的结构
- 实际的情况 ：大量的应用程序，特别是同一领域中的应用程序，分享相似的结构。这些结构并没有经过通常的面向对象技术而得到重用。例如：
  - 各种面向对象的分布式应用系统，其远程通信、服务对象的创建激活去激活等生命周期管理、服务对象引用的发布和获取、安全事务控制等等功能的工作机制基本一致，可以共享

框架

- 框架把握了某个领域内全部问题集解决方案的不变部分，开发者必须向框架添加变化部分代码以把握其动作，使该应用程序成为所在领域内特定的应用程序
  - 框架提供统一的总设计。它也提供对所有应用程序有用的功能（如接口、存储等）
  - 框架是对相似应用程序集合的一个部分解决方案。由领域内的专家所设计，并已编码和调试
  - 开发者用这不完整的解决方案加上必要的代码以建立一个完整的应用程序（20%：80% 功能 ）

例：EJB 服务器


Persistence
Life Cycle
State Management
Container
EJB
EJB
EJB
Transaction
Remote Interface
Security
Transaction manager

主要内容

- 框架
- 构件
- 对象总线
- 构件模型

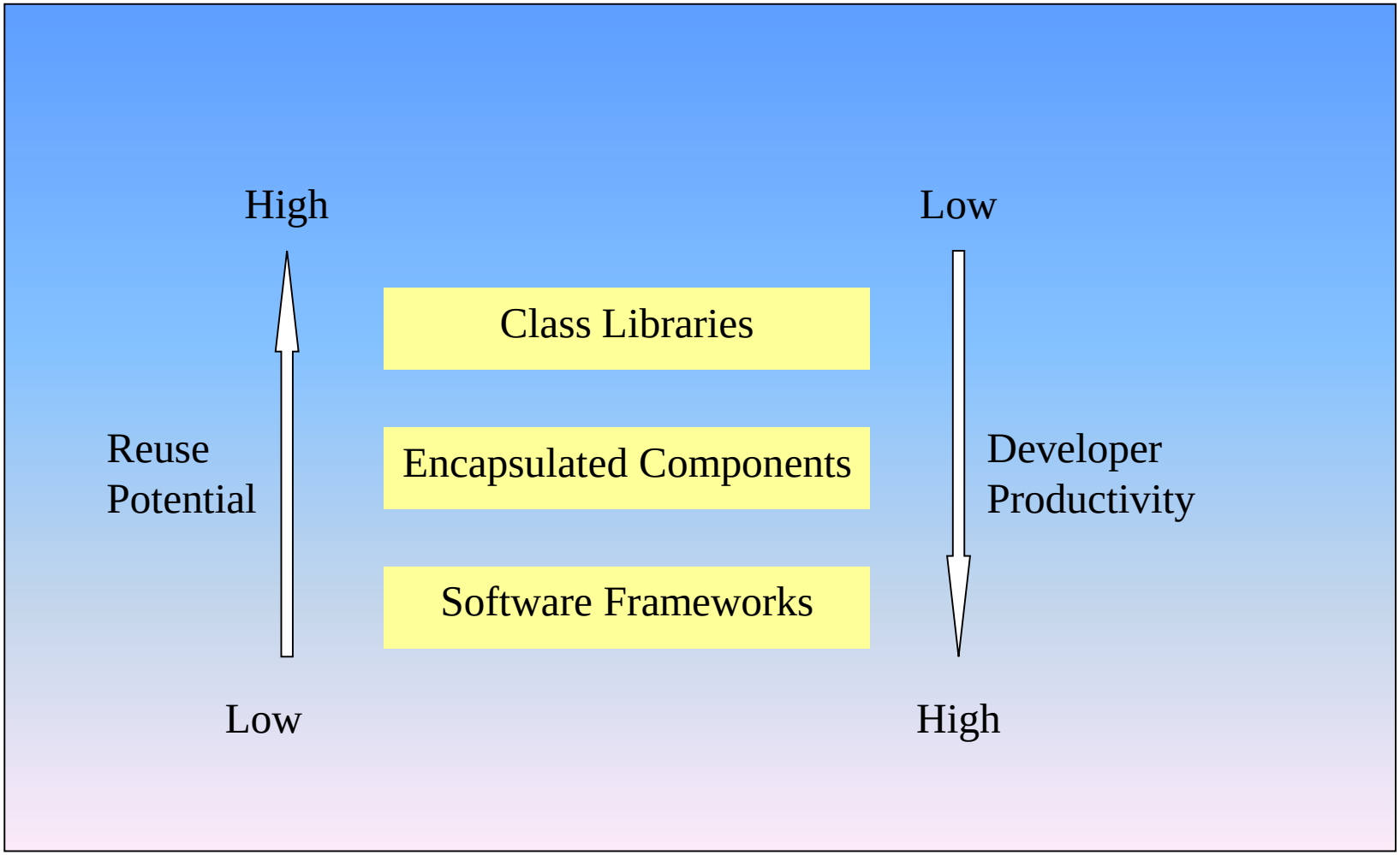
软件构件

- 构件定义 （业界尚没有一个统一的标准）
- “ 软件构件是一个接近独立的、可被替换的软件单元，它实现一个已定义好的体系结构上下文中的一个明确的功能，同时遵循并提供一组接口的物理实现。”
  - “接近独立”，指一个构件与其他的构件不相关，在不同的上下文中，一个构件可以与不同的构件互相合作，以满足应用的需要
  - “可被替换”，是指一个构件可被任何一个实现相同接口的构件替换（e. g. 技术更成熟、性能更好），并支持系统的升级，可以独立地改进系统的某一部分
  - “一个已定义好的体系结构上下文中”，是指构件代表设计和组成一个系统的一个基本构造块，它可以运行的环境是定义好的
  - “明确功能”，一个构件应是逻辑和物理内聚的，它是一个更大系统的一个有意义的成分。构件不是一个任意的组

构件之间的通讯

- 通过数据库间接进行
- **TCP/IP**
- **RPC**
- **ORB、RMI**
- **……**

广义构件的粒度及其可重用性



构件与对象的比较：不同点

- 构件技术是从面向对象技术中发展起来的，但它又不同于对象技术。在构件技术中，既可用面向对象的技术来实现，也可不用面向对象的技术来实现。对象技术在构件技术中不是必须的。开发者可能使用任何非面向对象语言和环境来实现构件。而且，构件可能包括遗留的系统，这些遗留的系统可能被包装起来以跟基于构件的软件相互操作
- 在面向对象技术中，继承、封装和多态性是OO技术的三个基本特征，但在构件技术中一般只突出封装的特性，即构件是独立可交付的软件单元
- 典型地，一个构件提供比对象更为强大的功能。对象和对象类型是用于应用开发的基本组成部分。开发者指定对象类型来包括关键的域和技术概念。一个应用可能包括几百个对象类型。相反，一个应用可能只包括几个构件，一个构件可能包括多个对象。构件提供了一个在对象级以上的一个组成部分。构件可看成是一个组织对象的一个方法

主要内容

- 框架
- 构件
- 对象总线
- 构件模型

对象总线

- 构件提供软件单元
- 框架把构件拼装在一起建立许多应用程序
- 对象总线把构件和框架的能力扩展到网络，使数百万独立的软件单元在异构环节下无缝地交互操作
- 对象总线的核心服务集：
  - 为总线上的所有对象提供通用服务（对象服务），扩充对象总线的功能
- 最著名的对象总线：CORBA（ORB） 、DCOM

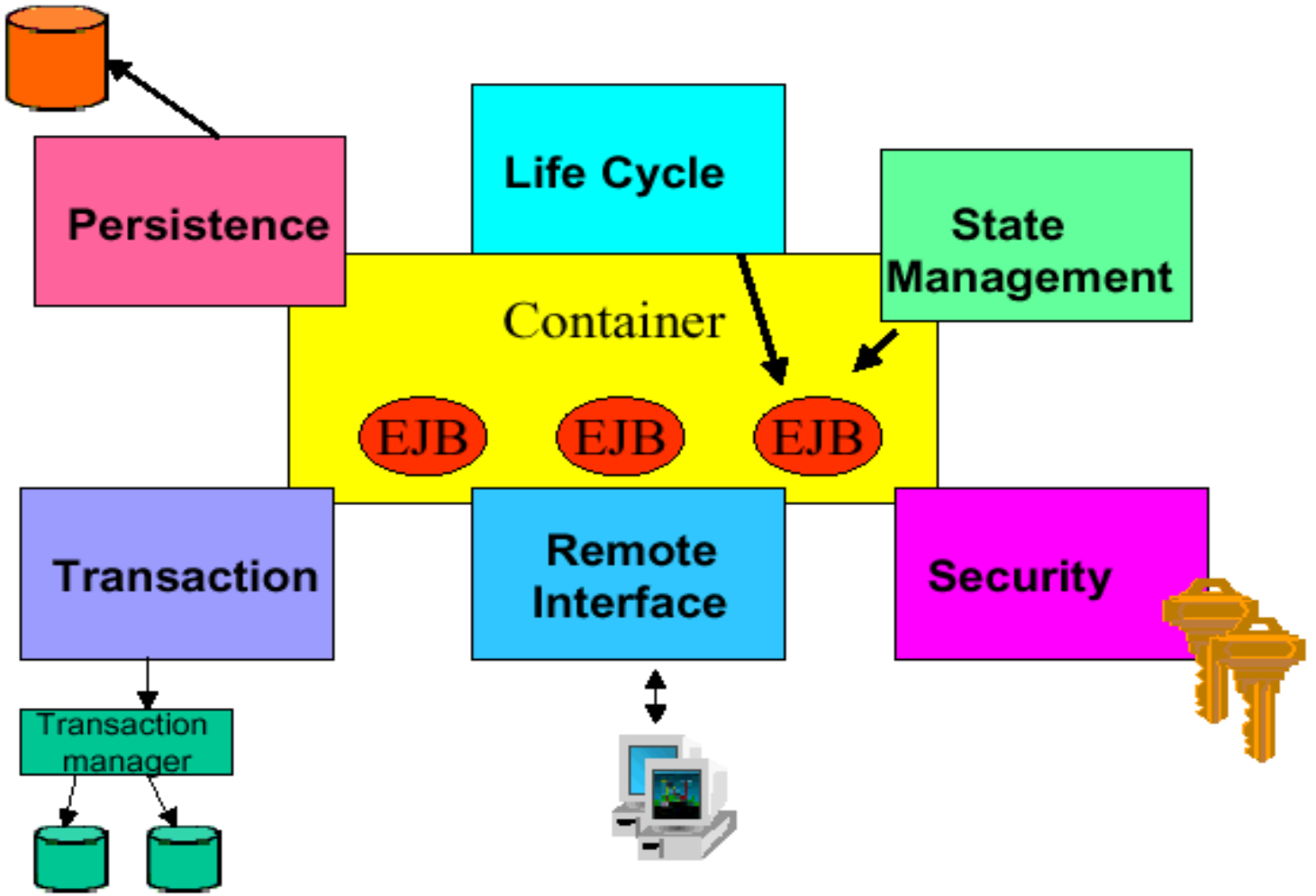
主要内容

- 框架
- 构件
- 对象总线
- 构件模型

构件模型

- 构件模型 = 构件 + 容器
  - 构件：具有可重用特性的基本软件部件
  - 容器：用于存放和安排构件，实现构件间的交互，并提供一系列可访问的管理服务，只要构件符合容器要求（规范），容器就可以让构件使用相应的管理服务
    - ➔容器提供的管理服务例
      - 生命周期管理、查询定位、安全服务、事务服务
      - 不同协议层的通信支持：远程通信、数据库访问
- 构件模型定义构件和容器的基本结构，制定构件和容器的基本接口，以及构件与构件之间、构件与容器之间交互的机制
  - 一般，构件需要实现容器要求的API才能在容器中运行（重量级框架）

EJB构件与 容器


Persistence
Life Cycle
State Management
Container
EJB
EJB
EJB
Transaction
Remote Interface
Security
Transaction manager

构件模型(续)

- 面向分布计算的构件模型中，不同容器间的构件可以交互，其交互的通信功能由容器提供，所以这种容器的功能可理解为：“基于构件的软件体系结构”上下文中的“框架+（对象）总线”

- 主流构件模型
  - CORBA的构件模型、EJB、COM（ DCOM / COM+ ）、Spring……

- 现在常用的框架一词，如Spring框架中的框架，可以理解为是一种构件模型

重量级框架

- 典型代表EJB 1/2
- 开发的系统基本需要放置在一个容器系统中进行运行，并需要实现容器要求的接口
- 容器在实例化化业务对象后，传给业务对象上下文，而业务对象本身要通过JNDI手段来定位或者pull出其他资源或者业务对象
- 这些容器因为基本针对大型企业应用，所以体积庞大，占用资源，内在服务多，启动比较慢
- 开发需要遵从的规则比较多，开发效率也比较低，很大一部分时间都用在了Deploy、Run这样的过程上，调试和测试比较困难

轻量级框架

- 典型代表：Spring
- 以依赖注入（Dependency Injection）为代表的解耦合模式，可以让构件不去依赖容器（运行环境）的API
- 轻量级容器通过反向控制（Inversion of Control）让容器具有主动权，去管理插进来的组件，只要组件是符合标准的，就可以被轻量级容器管理
- 构件以POJO（Plain Old Java Object）的形式存在，只要你有Java.exe就可以运行它，不需要容器就可以实现测试行为

小结

- 基于构件的软件体系结构：构件、对象总线和框架共同组成了软件应用程序
  - 软件框架以基于构件的软件体系结构为组装蓝图，以可复用软件构件为组装预制块，支持组装式软件复用
    - ➔描述的是系统整体设计格局，表现不同系统的高层共性
    - ➔为构件提供结构，使软件工程从工程化编程及类库再前进了一步
  - 构件是可重用的软件部件
  - 构件和框架通过对象总线与别的构件和框架连接
    - ➔通过对象总线，使跨越不同异构环境上的应用程序开发成为可能。

小结（2）

- 构件模型：构件模型定义构件和容器的基本结构，制定构件和容器的基本接口，以及构件与构件之间、构件与容器之间交互的机制
  - 构件模型=构件 + 容器
  - 主流构件模型：CORBA（构件模型部分）、EJB、COM（DCOM / COM+）、Spring
- 重量级VS轻量级框架
  - 此上下文中的“框架”相当于“构件模型”
    - 此类术语的理解离不开上下文

主要内容

- 基于构件的软件体系结构
- J2EE/Java EE
- EJB
- 轻量级框架和EJB3.0

为什么需要J2EE/Java EE

- 在企业级应用开发中所面临的而且必须解决的问题：
  - 分布式
  - 可移植
  - 旧系统集成支持
  - 面向Web
  - 可装配
  - 事务性、安全性
  - 可伸缩、可扩展、易维护

J2EE/Java EE是什么

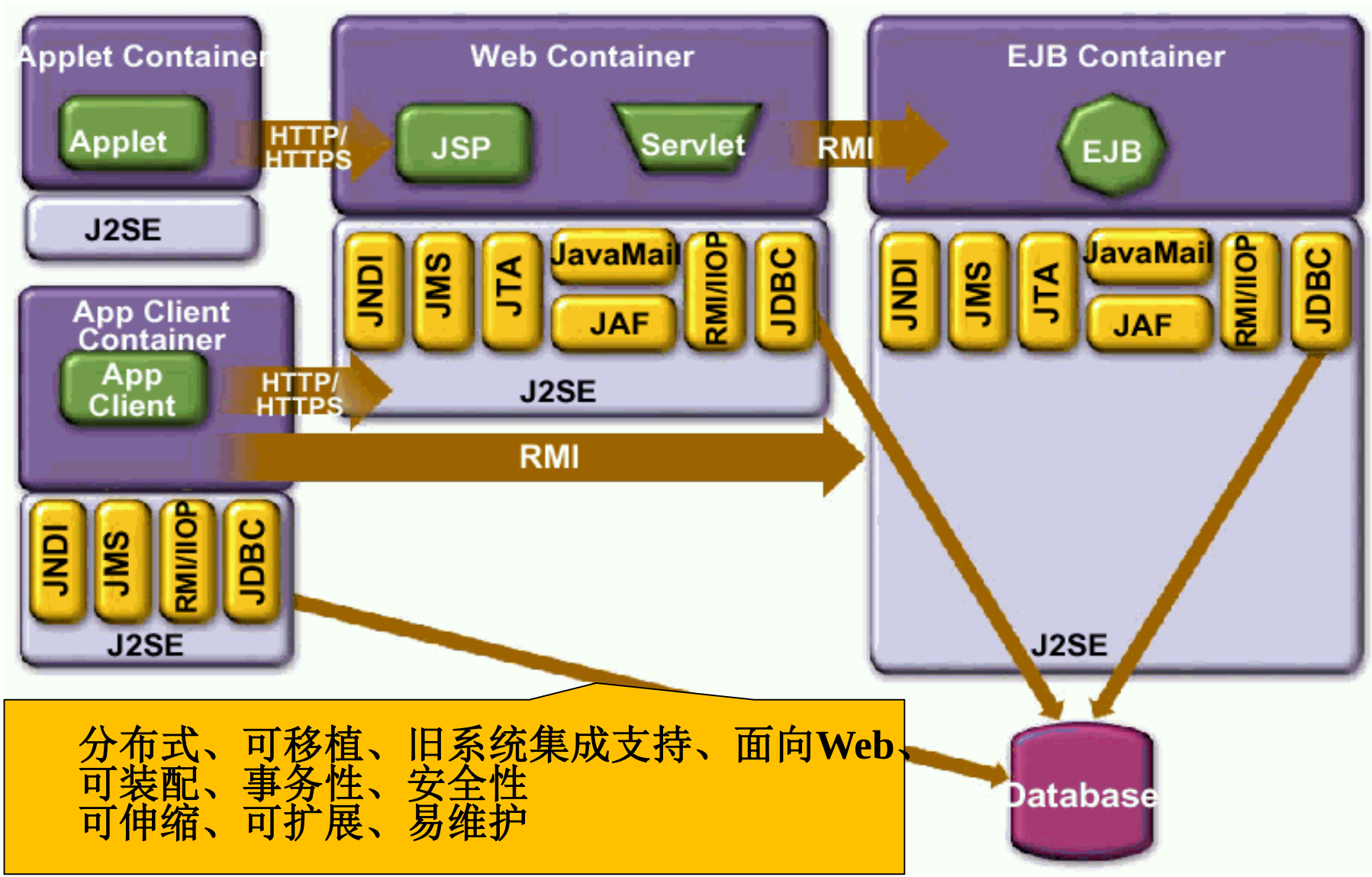
■J2EE (Java 2 Platform, Enterprise Edition)是一套使用Java进行企业级Web应用开发的事实上的工业标准。EJB 1.0 1998年发布

■J2EE是一种平台规范，该平台提供了一套基于组件的方法来设计、开发、装配及部署N层结构的、面向Web的，以服务器为中心的企业级应用。

- J2EE平台提供了多层分布式的应用模型、重新利用组件的能力、统一安全的模式以及灵活的处理控制能力。

■2005年8月， Java升级到1.5版，SUN的伙伴们将J2EE 1.5改名为java EE 5(Java Enterprise Edition)，以前J2EE版本还是称为J2EE。

总体架构


Applet Container
Applet
J2SE
HTTP/
HTTPS
Web Container
JSP
Servlet
RMI
EJB Container
EJB
JNDI
JMS
JTA
JavaMail
JAF
RMI/IIOP
JDBC
J2SE
App Client
Container
App
Client
HTTP/
HTTPS
RMI
J2SE
分布式、可移植、旧系统集成支持、面向Web、
可装配、事务性、安全性
可伸缩、可扩展、易维护
Database

Java EE

- J2EE 1.5改名为Java EE 5（2005年）
  - 以前J2EE版本还是称为J2EE
  - 目的是让大家清楚J2EE只是Java企业应用
- Java EE不再象以前那样只注重大型商业系统的开发，而是更关注小到中型系统的开发，简化这部分系统开发步骤
  - 加入Annotations （注解/标注）
  - 减弱业务核心组件对J2SE/J2EE版本的依赖
- 08年Java EE 6，13年Java EE 7，2017年 Java EE 8
- 2018年3月，开源组织Eclipse基金会宣布，Java EE（Enterprise Edition）被更名为Jakarta EE

体系结构

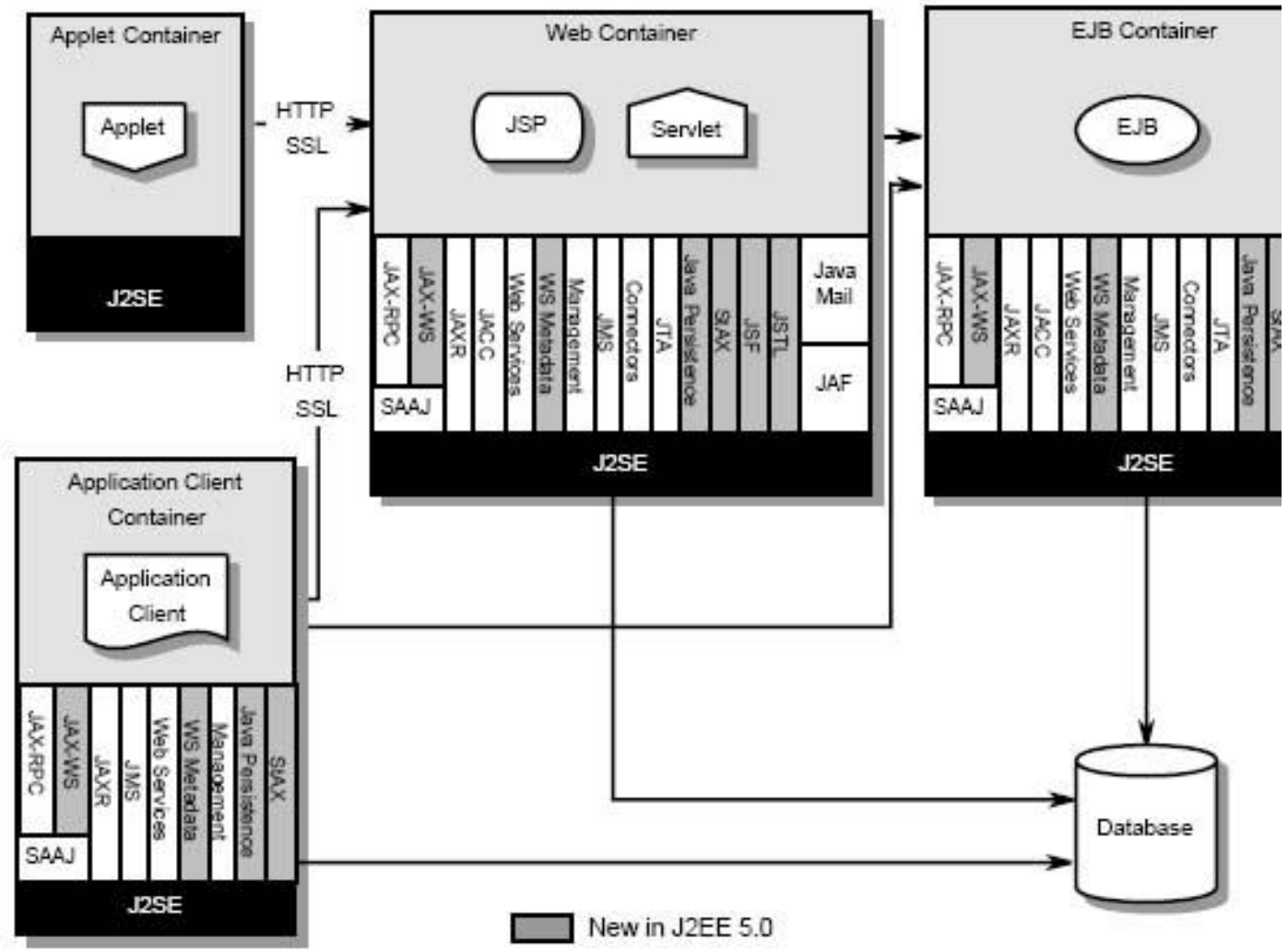
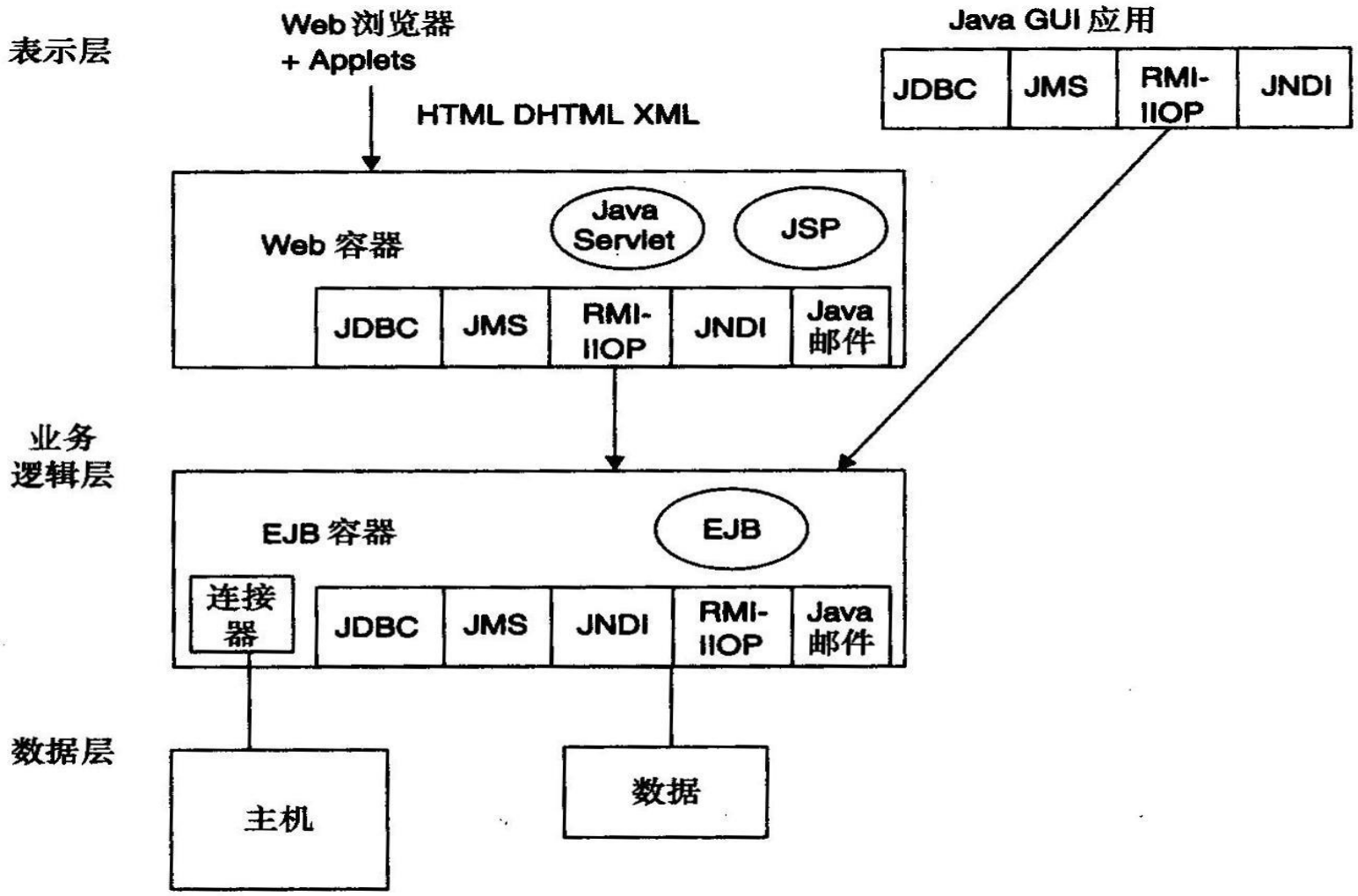


Figure J2EE.2-1 J2EE Architecture Diagram

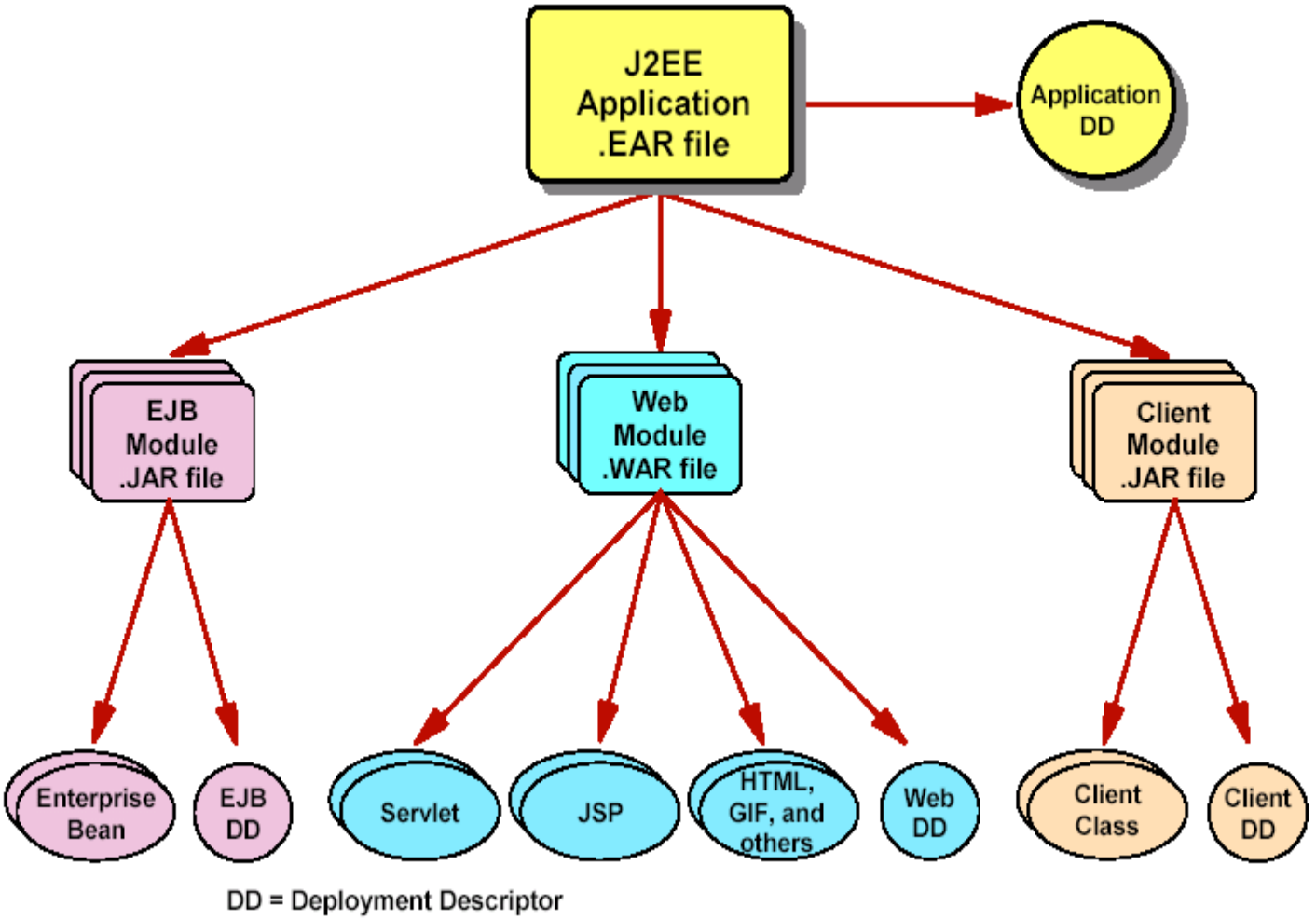
基本平台技术

- **应用构件（组件）**：由开发人员实现，构成应用系统，运行在Java EE平台上，即运行在各种容器中
  - 客户端构件：Applets，Client Applications
  - 服务端构件：Web构件(Servlets，JSPs)，EJBs
- **服务**：Java EE应用构件所使用的功能，由平台提供商实现，分为：
  - Service API（开发时使用）
    - JNDI、JDBC、JTA、Java Mail Service API ……
  - 运行时服务:生命周期、事务、安全、持久性、资源管理
- **通信**：支持协作构件之间的通信，由Container提供

通信技术例


表示层
Web 浏览器
+ Applets
HTML DHTML XML
Java GUI 应用
JDBC
JMS
RMI-IIOP
JNDI
Web 容器
Java Servlet
JSP
JDBC
JMS
RMI-IIOP
JNDI
Java 邮件
业务逻辑层
EJB 容器
EJB
连接器
JDBC
JMS
JNDI
RMI-IIOP
Java 邮件
数据层
主机
数据

J2EE Packaging

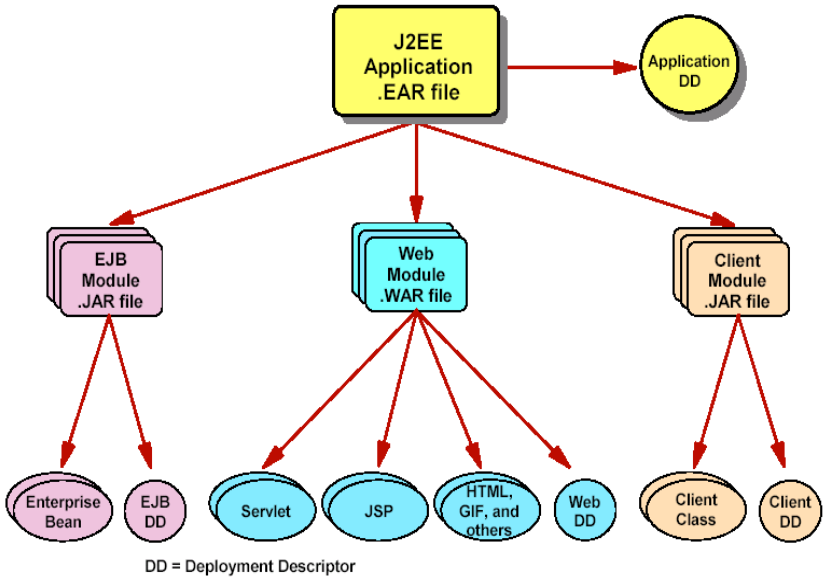


Deployment Descriptors

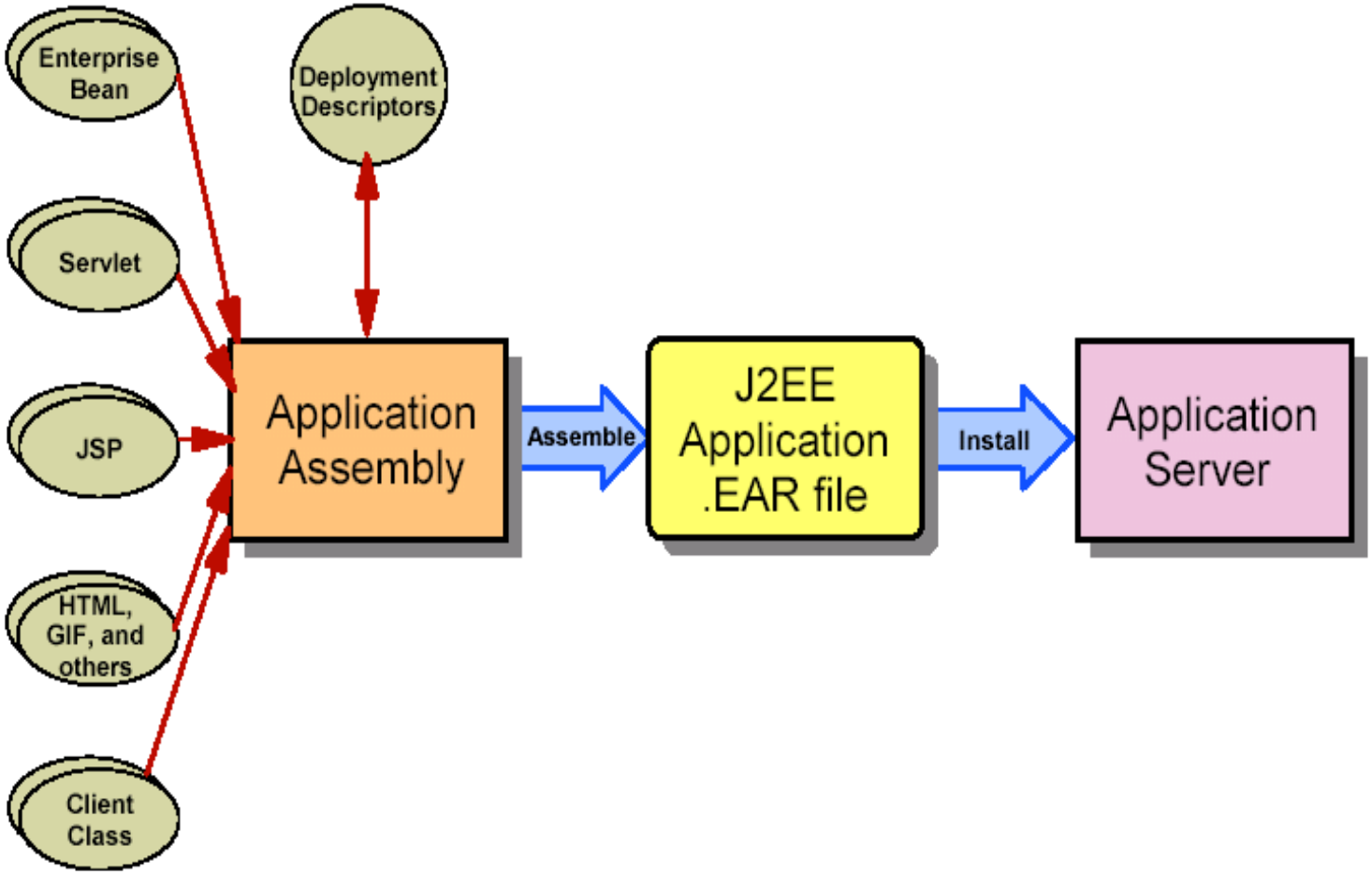
- XML格式的文件，用于描述：
  - 模块中所包含的组件
  - 模块所需要的环境（如安全性控制）
- 每个模块或ear文件都有一个DD
- 可以由厂商提供的工具自动生成，并进行可视化的编辑
- 可以手工创建和编辑

例：EAR Deployment Descriptors

- The application.xml：
  - 定义了EAR文件中包含的所有模块：
    - EJB jars
    - WARs
    - Application Client jars
  - 定义安全性角色



应用程序的组装与安装


Enterprise Bean
Deployment Descriptors
Servlet
JSP
HTML, GIF, and others
Client Class
Application Assembly
Assemble
J2EE Application .EAR file
Install
Application Server

Java EE 技术支持产品示例

- 表现层框架
  - Struts/web work/**Struts2/ JSF**/tapestry(组件)
- 数据持久化层
  - ORM类：EJB(CMP|**JPA**)/**Hibernate**/toplink/openJPA/JDO/Apach OJB
  - SQL定制类: iBatis（）
- **集成框架：EJB | Spring |JDON|Nanning|PicoContainer ---GLUE ws**
  - Web容器：**Tomcat|Resin|Jetty**
  - Web容器+EJB容器：**JBoss（开源）| webLogic** | webSphere |
- 集成开发环境
  - **Eclipse，MyEclipse，NetBeans，** JBuilder，Jcreator, IDEA

J2EE/Java EE 小结

- 支持多层应用开发模型
- 将实现多层结构应用的工作分为两部分
  - 系统服务由平台提供
  - 开发者关注于商业逻辑和表示逻辑的实现
- Write Once, Run Anywhere
  - J2EE是一个开放标准
  - Java的特性
- 丰富的部署时定制特性：如安全性限制、资源
- 明确划分任务与责任
- 系统可扩充性
- 灵活的安全性模型
- 逐步关注中小型应用的开发