

# 网络计算模式

---



秦 臻

电子科技大学信息与软件工程学院



# 课程说明

☞ 上课时间:

◆ 周三, 第9,10,11节

◆ 周五, 第7,8节

☞ 课件下载: 微信群



# 课程说明

## ∞ 成绩构成:

- ◆ 平时40分 (考勤、分组讨论)
- ◆ 期末60分 (考试: 笔试)

## ∞ 联系方式:

- ◆ 沙河校区主楼 (中) 422
- ◆ 邮箱: qinzhen@uestc.edu.cn



# 第1章 概述

---



## 网络计算模式



# ● 课程主要内容

- 概述
- 企业计算
- 网格计算和云计算
- P2P网络、CDN网络和物联网
- 社会计算



# 网络计算的发展历程

## 企业计算

- ◆ Stand-alone、dbase单机系统、C/S架构、B/S架构；企业应用集成；中间件服务。

## 网格计算和云计算

- ◆ 网格计算；OGSA开发网络服务体系架构；云计算基础；Google云计算；HDFS文件系统。

## P2P网络、CDN网络和物联网

- ◆ P2P网络概念及分类；分布式哈希表；CDN内容发布网络；物联网及其架构。



# 网络计算的发展历程

## 社会计算（分组讨论）

- ◆ 社会网络概念；社会计算概念；社会计算相关算法；社会计算应用。



# 第2章

## 企业计算

---

### 网络计算模式





# ● 课程主要内容

- 概述
- 企业计算
- 网格计算和云计算
- P2P网络、CDN网络和物联网
- 社会计算



# 企业计算

## ∞ 什么是企业计算?

- ◆ **企业计算(Enterprise Computing)**是稍时髦较好听的名词，**主要是指企业信息系统**，如ERP软件(企业资源规划)、CRM软件(客户关系管理)、SCM软件(供应链管理，即物流软件)，银行证券软件，财务软件，电子商务/政务(包括各种网站)，数据仓库，数据挖掘，商务智能等企业信息管理系统。
- ◆ 是以实现大型组织内部和组织之间**信息共享和协同工作**为主要需求而形成的网络计算技术，其核心是**client/server计算模型**和相关的中间件等技术。



# 企业计算

## ∞ Stand-alone、dbase单机系统

Data Logic	数据逻辑 ( 实体、数据 )	Persistent
		Temporary
Business Logic	业务逻辑 ( 做事情 )	
User Logic	用户逻辑 ( 使用者 )	

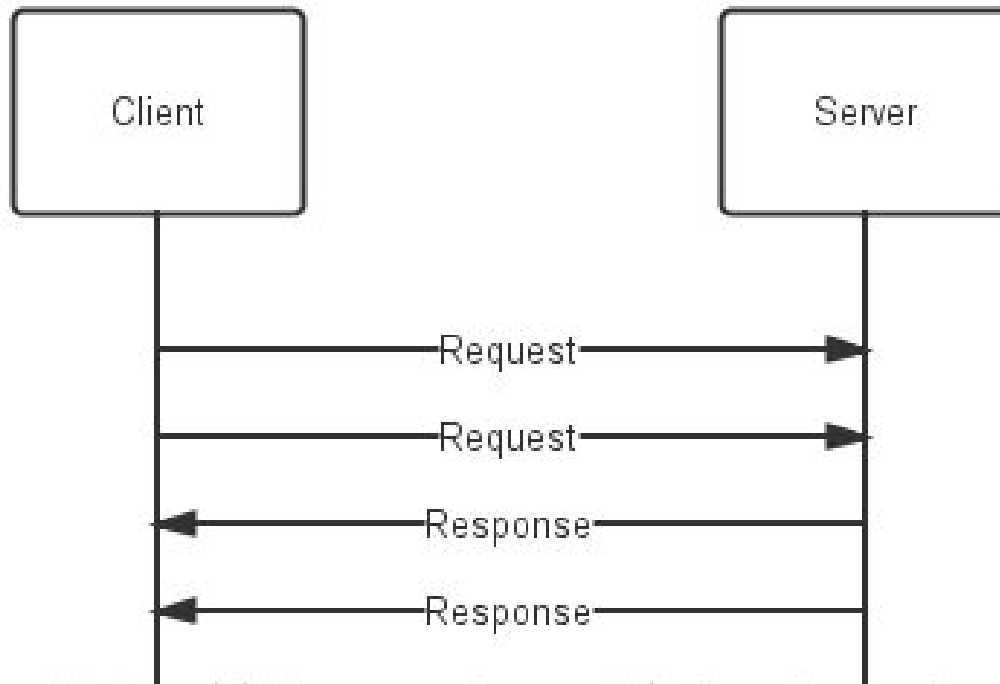
- ◆ **dbase**: 第一个在微型计算机上被广泛使用的关系型数据库管理系统 (DBMS) 。



# 企业计算

## ☞ Client/Server计算模型

- ◆ 有多个客户端;
- ◆ 至少有一个服务器;



# 企业计算

## 2-tier C/S、Fat-client架构

Data Logic	Server ( DBMS )	Pros : ( 1 ) 界面丰富 ( 2 ) 数据共享存储  Cons : ( 1 ) 维护成本高 ( 一旦修改客户端全要修改 )
Business Logic	Client	
User Logic		

- ◆ **比 Standalone 系统能提供更高的性能**，因为客户端和服务端将应用的处理要求分开，同时又共同实现其处理要求，对客户端程序的请求实现“分布式应用处理”。
- ◆ 服务器为多个客户端应用程序管理数据，而客户端程序发送请求和分析从服务器接收的数据，这是一种**胖客户机(Fat-Client)**，**瘦服务器(Thin-Server)**的网络计算模式。



# 企业计算

## ∞ 2-tier C/S、Fat-client架构的问题

- ◆ **系统的可靠性有所降低。** 一个客户机/服务器系统是由各自独立开发、制造和管理的各种硬件和软件的混合体，其内在的可靠性不如单一的、中央管理的大型机或小型机，出现问题时，很难立即获得技术支持和帮助。
- ◆ **维护费用较高。** 尽管这种应用模式在某种程度上提高了生产效率，由于客户端需要安装庞大而复杂的应用程序，当网络用户的规模达到一定的数量之后，系统的维护量急剧增加，因而维护应用系统变得十分困难。



# 企业计算

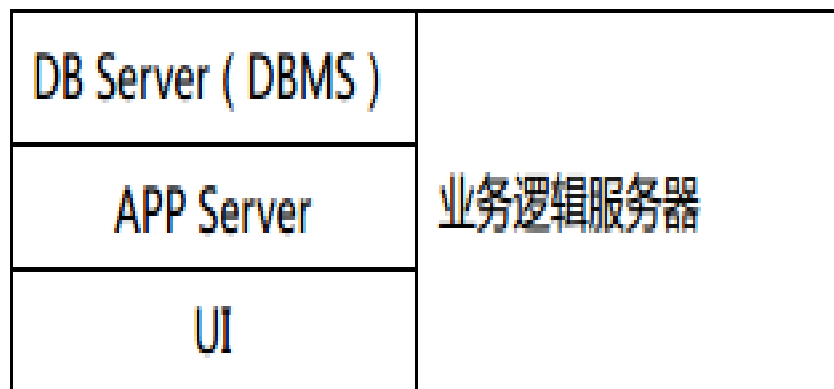
## ∞ 2-tier C/S、Fat-client架构的问题

- ◆ **系统资源的浪费。** 随着客户端的规模越来越大，对客户机资源的要求也越来越高。尽管硬件不断更新，但新的操作系统和新的应用软件的不断出现，使得用户对硬件的更新仍然跟不上软件更新的速度。
- ◆ **系统缺乏灵活性。** 客户机/服务器需要对每一应用独立地开发应用程序，消耗了大量的资源，但胖客户机的计算模式却仍然满足不了日益增长的应用的需要。
- ◆ 应对上述问题，网络计算模式结合动态计算，**从两层模式扩展到N层模式。**



# 企业计算

## ∞ 3-tier C/S架构、N-tier B/S架构



◆ 在三层模式中，Web 服务器既作为一个浏览服务器，又作为一个应用服务器，在这个中间服务器中，可以将整个应用逻辑驻留其上，而只有表示层存在客户机上。**这种结构被称之为“瘦客户机”**。这种结构中，无论是应用的HTML页还是Java Applet，都是运行时刻动态下载的，只需随机地增加中间层的服务(应用服务器)，即可满足扩充系统的需要。





# 企业计算

## ∞ 3-tier C/S架构的主要优势

- ◆ 利用单一的访问点，可以在任何地点访问站点的数据库；
- ◆ 对于各种信息源，无论是文本还是图形都采用相同的界面；
- ◆ 所有的信息，无论其基于的平台，都可以用相同的界面访问；
- ◆ 可跨平台操作；
- ◆ 减少整个系统的成本；
- ◆ 维护升级较方便；
- ◆ 具有良好的开放性；
- ◆ 系统的可扩充性良好；
- ◆ 进行严密的安全管理；
- ◆ 系统管理简单，可支持异种数据库，有很高的可用性。



# 企业计算

## ∞ 3-tier C/S架构、N-tier B/S架构

DB Server ( DBMS )	业务逻辑服务器
APP Server	
UI	



Data Logic	Server
Business Logic	
Presentation Logic	
User Logic	Browser

- ◆ 三层 C/S 结构是将应用功能分成表示层、功能层和数据层三部分。与二层 C/S 结构相比，其程序的可维护性要好得多，但是其业务处理所需的数据要从服务器传给客户机，所以系统的性能需要考虑此因素造成的影响。



# 企业计算

## ∞ 3-tier C/S架构、N-tier B/S架构

DB Server ( DBMS )	业务逻辑服务器
APP Server	
UI	



Data Logic	Server
Business Logic	
Presentation Logic	
User Logic	Browser

◆ **B/S架构维护和升级方式简单。**目前，软件系统的改进和升级越来越频繁，B/S 架构的产品明显体现着更为方便的特性。无论用户的规模有多大，有多少分支机构都不会增加任何维护升级的工作量，所有的操作只需要针对服务器进行。如果是异地，只需要把服务器连接专网即可，实现远程维护、升级和共享。



# 企业计算

随着急剧的市场变化以及不断调整的企业战略,流程的变化、业务的整合几乎成了中国企业的家常便饭。这样的变化一方面给企业在IT的投入带来巨大的浪费,另一方面也将提供IT服务的企业带入巨大的窘境。有企业家曾发过这样的抱怨: **“十几年来,公司的战略几经调整,组织、流程乃至业务更是不断地在变化之中,有时候我也知道这是企业发展中的一个阶段,但没办法,这是事实。问题是我们的每一次调整,都会给我们的IT部门带来巨大的麻烦。”**



# 企业计算

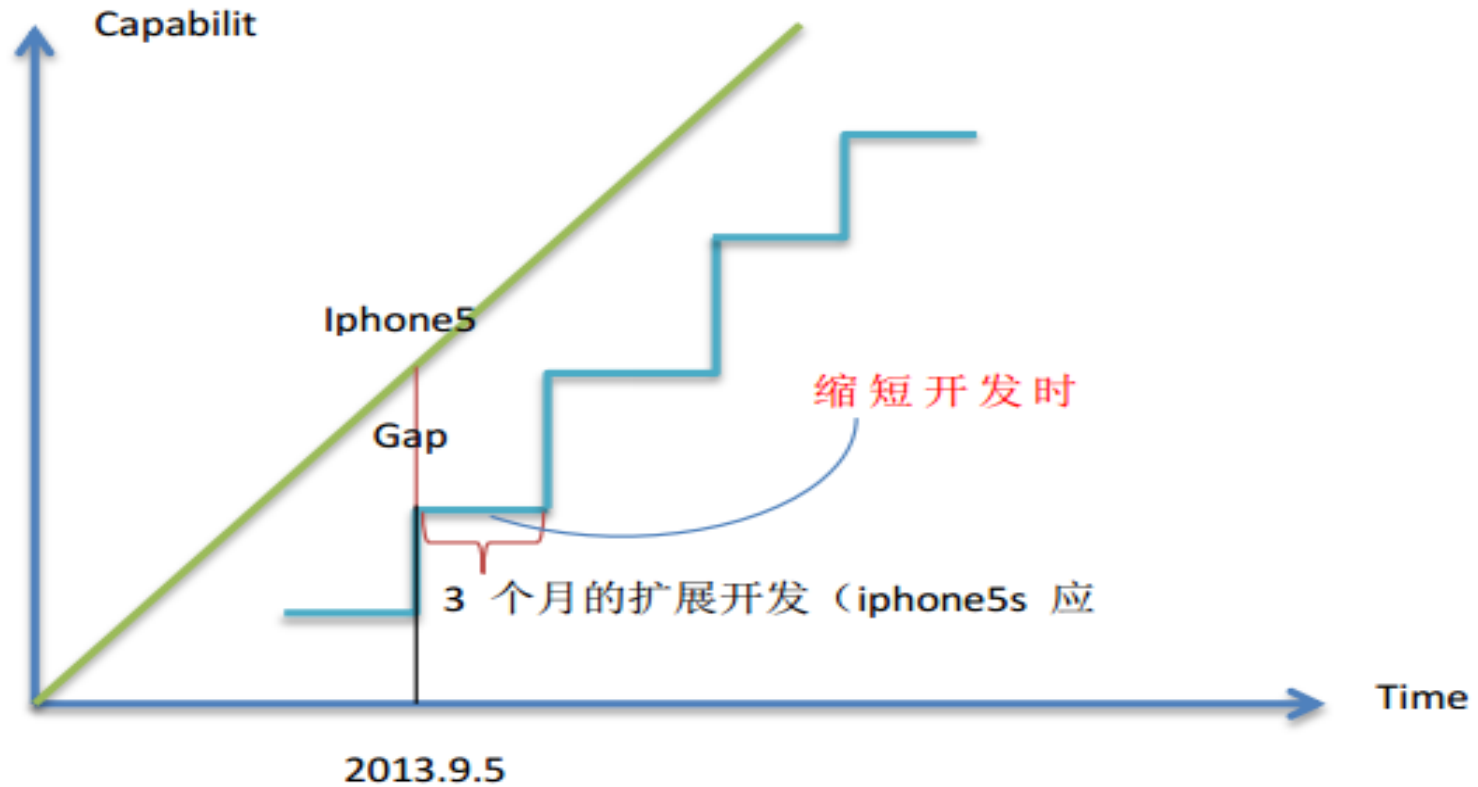
## ∞ IT鸿沟

- ◆ **IT鸿沟**指的是当业务不断发展的时候，现有的IT支持平台的发展不能满足业务的需要，而当IT支持平台发展到可以支持业务的需要的时候，业务又有新的需求和发展，这样业务与IT支持之间持续存在一个鸿沟。
- ◆ 存在IT鸿沟的一个重要的原因就是企业中的信息孤岛问题。**信息孤岛**指相互之间在功能上不关联互助、信息不共享互换以及信息与业务流程和应用相互脱节的计算机应用系统。



# 企业计算

## IT鸿沟



缩短开发时间的方法：提高重用性！



# 企业计算

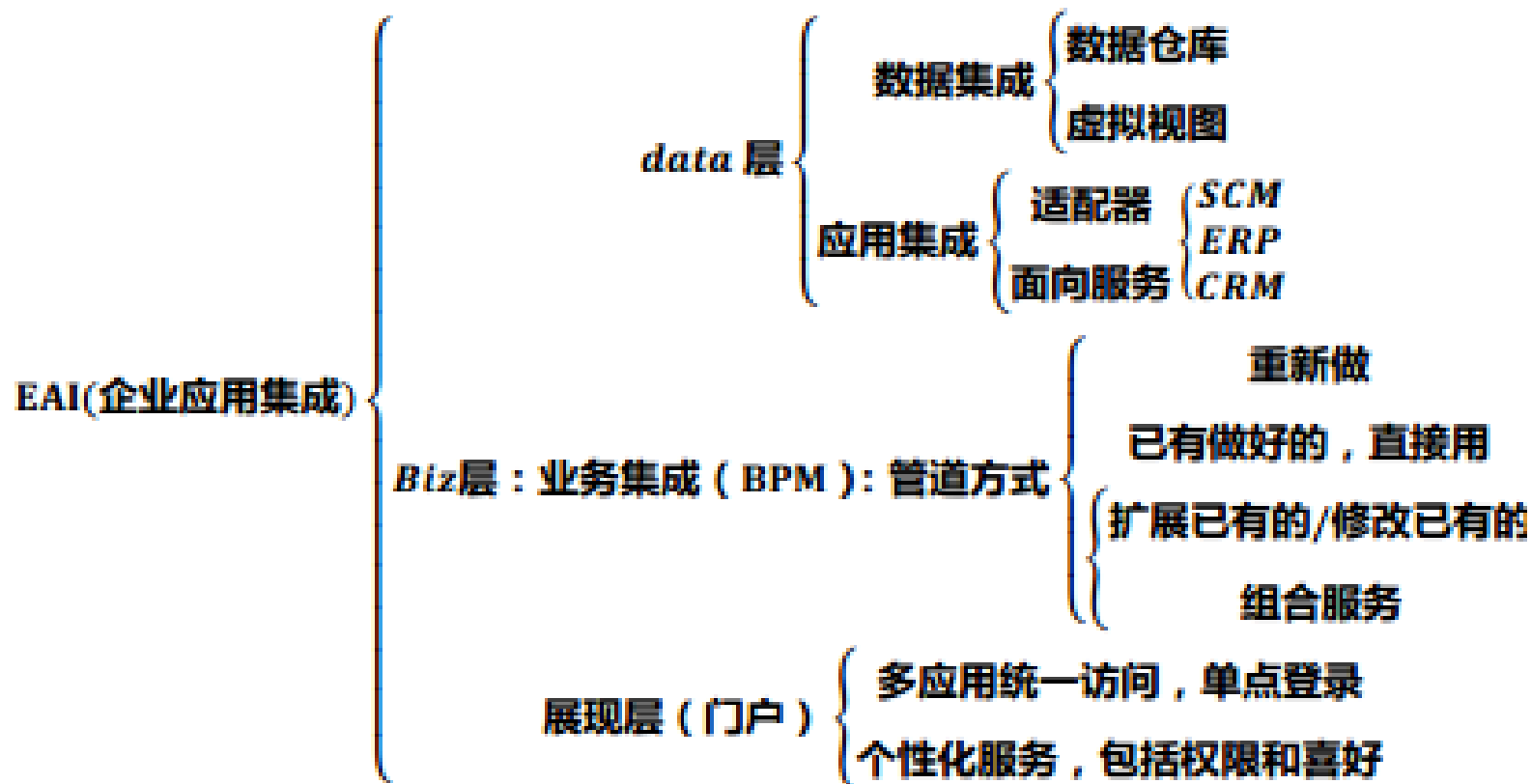
## ∞ 提高重用性（针对公共服务、共性服务）

- ◆ 与数据库有关
- ◆ 与分布式有关
- ◆ 与安全可靠有关



# 企业计算

## ∞ IT鸿沟的解决办法——EAI（企业应用集成）





# 企业计算

## ∞ IT鸿沟的解决办法——EAI（企业应用集成）

- ◆ EAI(Enterprise Application Integration)企业应用集成，具体到技术层面上的划分，我们认为一套完整的 EAI 技术层次体系应该包括**应用接口层**，**应用整合层**，**流程整合层**和**用户交互层**四个大的层面。
- ◆ EAI 技术层次体系最下面的一层是**应用接口层**，它要解决的是应用集成服务器与被集成系统之间的连接和数据接口的问题。
- ◆ 再往上去就是**应用整合层**，它要解决的是被集成系统的数据转换问题，通过建立统一的数据模型来实现不同系统间的信息转换。



# 企业计算

## ∞ IT鸿沟的解决办法——EAI（企业应用集成）

- ◆ 应用整合层之上是**流程整合层**，它将不同的应用系统连接在一起，进行协同工作，并提供商业流程管理的相关功能，包括流程设计、监控和规划，实现业务流程的管理。
- ◆ 到了最上端的**用户交互层**，则是为用户在界面上提供一个统一的信息服务功能入口，通过将内部和外部各种相对分散独立的信息组成一个统一的整体，保证了用户既能够从统一的渠道访问其所需的信息，也可以依据每一个用户的要求来设置和提供个性化的服务。



# 企业计算

## 企业应用集成——应用接口层

- ◆ EAI要解决的问题是独立应用系统之间的连接，传统的应用系统之间的连接方式包括了：CORBA，SOCKET 通讯，HTTP，FTP 等；数据库系统之间常见的连接规范包括：开放数据库互连（ODBC），java 数据库连接（JDBC）。上述这些规范在企业应用系统或数据库系统之间传统的点对点的连接中得以广泛应用。但是由于这些系统之间的连接是通过上述连接接口实现的，没有一定的规范和标准，使得在今后新系统的加入和旧系统的移植过程中，就会产生接口兼容性问题。而在EAI的应用接口层，主要是通过适配器技术将原有数据库系统、应用系统和原有网络服务组件封装起来，实现系统之间的互通互联。



# 企业计算

## 企业应用集成——应用接口层

- ◆ **适配器**是 EAI 厂商或产品厂商为了解决系统之间的连接而开发的、可重用的、统一的接口，通过该接口每一个应用系统仅需要与业务整合平台相连，而不需要与每个与之交互的应用系统相连。
- ◆ **适配器的引入主要有以下优点：**为独立的系统之间提供重用的、统一的接口；支持事务管理、安全性和连接管理与应用系统连接；增加企业数据资源的共享程度；适配器封装了对数据操作的缓冲区，可以提高对数据批处理访问的效率；增强了基于组件或面向对象开发模块的网络应用的功能。
- ◆ **适配器一般可分为四类：**即企业应用系统适配器、技术标准类适配器、主机系统适配器和自行开发适配器。



# 企业计算

## 企业应用集成——应用整合层

- ◆ 应用整合层是 EAI 技术层次体系中的核心层次，该层次是连接业务流程管理层和应用接口层的桥梁。数据信息在业务流程中的流转以及在各个应用系统之间的交互必须建立在数据源和数据目的地都能理解该数据信息的基础之上。**应用整合层定义了能为数据产生源、数据处理地、数据投送地都能理解的信息处理规范方式、方法和规则，包括：数据格式定义、数据转换和消息路由。**
- ◆ **数据格式定义通常有三种方式：**使用特定行业已经存在的标准(如：SQL,Oracle,Sybase)；使用XML作为数据格式的统一描述语言；用户自定义格式(txt,Excel)。



# 企业计算

## 企业应用集成——应用整合层

- ◆ **数据转换**是应用整合层的重要组成部分，它是指将不同的信息格式和语法重新转换成能被目标应用系统所理解的数据格式和语义的整合技术。数据转换包括两个层面的内容，即**数据格式转换**和**数据语义转换**。
- ◆ **消息路由**建立在已有数据格式定义和数据转换基础之上，通过消息中间件技术能够动态的识别和理解从源应用发出的消息并且把它发向特定的应用系统。



# 企业计算

## 企业应用集成——流程整合层

- ◆ 业务流程分企业内部业务流程和企业外部业务流程。因此，企业需要实现的业务流程集成分为三部分：**内部业务流程集成**，**外部业务流程集成**，**内外部业务流程集成**。企业的内部业务流程集成是由通过（Business Process Management, BPM）实现的，外部的业务流程集成是通过（Business-to-Business, B2B）实现的，内外部业务流程集成是通过BPM和B2B共同实现的。BPM实现了一个平台两种业务流程模式（业务流程自动化和人员介入工作流）的融合能力，包括**流程建模**、**流程监控**、**工作流**等方面。



# 企业计算

## 企业应用集成——流程整合层

- ◆ **流程建模**是对业务模型进行集中设计并对模型进行管理。流程建模使用活动图形和常用结构图等模型结构来设计业务模型，用图像显示方式建造内部和外部不同组织的控制流模型，实现层次控制和流程再利用。**流程建模**可以集成企业应用系统、主机系统、数据库、应用服务器、员工工作流程和企业接入业务伙伴系统，同时为业务分析师提供了可视的、易懂的程序自动化解决方案。EAI 平台提供可视化的流程建模工具，包括图形化的流程设计和管理界面，流程运行时可视化的维护管理和监控分析界面。





# 企业计算

## 企业应用集成——流程整合层

- ◆ **流程监控**是对系统中的核心业务流程进行分级和分类（分级可以按照业务的优先级来划分，分类可以按照业务类型进行归类）的管理。使用预定义的图形格式，监查特定流程实例，对业务流程的动态执行过程实时跟踪，并对业务流程执行情况（如业务流程执行成功、失败、回滚）进行统计。通过流程监控，从业务的角度来分析业务流程，为高层提供决策支持。流程监控还提供通知和预警机制，能够更有效地控制业务流程的运行。



# 企业计算

## 企业应用集成——流程整合层

工作流是指那些需要人工进行干预的业务流程。工作流的实现是通过基于可视模型，管理和实现以人为本的业务流程的不断精简，通过设计、部署、管理并维护工作流，可分阶段部署工作流，不用在控制工作流的效益实现前就对所有业务流程重新设计，并把工作流覆盖到企业的各个方面。



# 企业计算

## 企业应用集成——用户交互层

- ◆ **用户交互层**是 EAI 与用户实现人机交互在表示层面上的扩展。涉及的内容包括展示内容的集成（门户应用）、单点登陆（Single Sign On）、用户统一管理、用户认证授权的管理等。现今很多 EAI 产品都提供了对用户集成这几方面内容的支持。



# 企业计算

## 企业应用集成 (Enterprise Application Integration)

- ◆ 应用接口层、应用整合层、流程整合层、用户交互层。
- ◆ EAI是利用**通用的中间件 (Middleware)** 融合了企业已有的应用软件、商业封装式应用软件以及新代码三方面的功能。通过建立底层结构，来联系横贯整个企业的异构系统、应用、数据源等，完成在企业内部的ERP、CRM、SCM、数据库、数据仓库，以及其他重要的内部系统之间无缝地共享和交换数据的需要。
- ◆ EAI使得企业众多信息系统都与一个**由中间件组成的底层基础平台**相连接，各种“应用孤岛”、“信息孤岛”通过各自的“适配器”（可以理解成一个转接口）连接到一个总线上，然后再通过一个消息队列实现各个应用之间的交流。



# 企业计算

## ☞ 中间件 (Middleware)

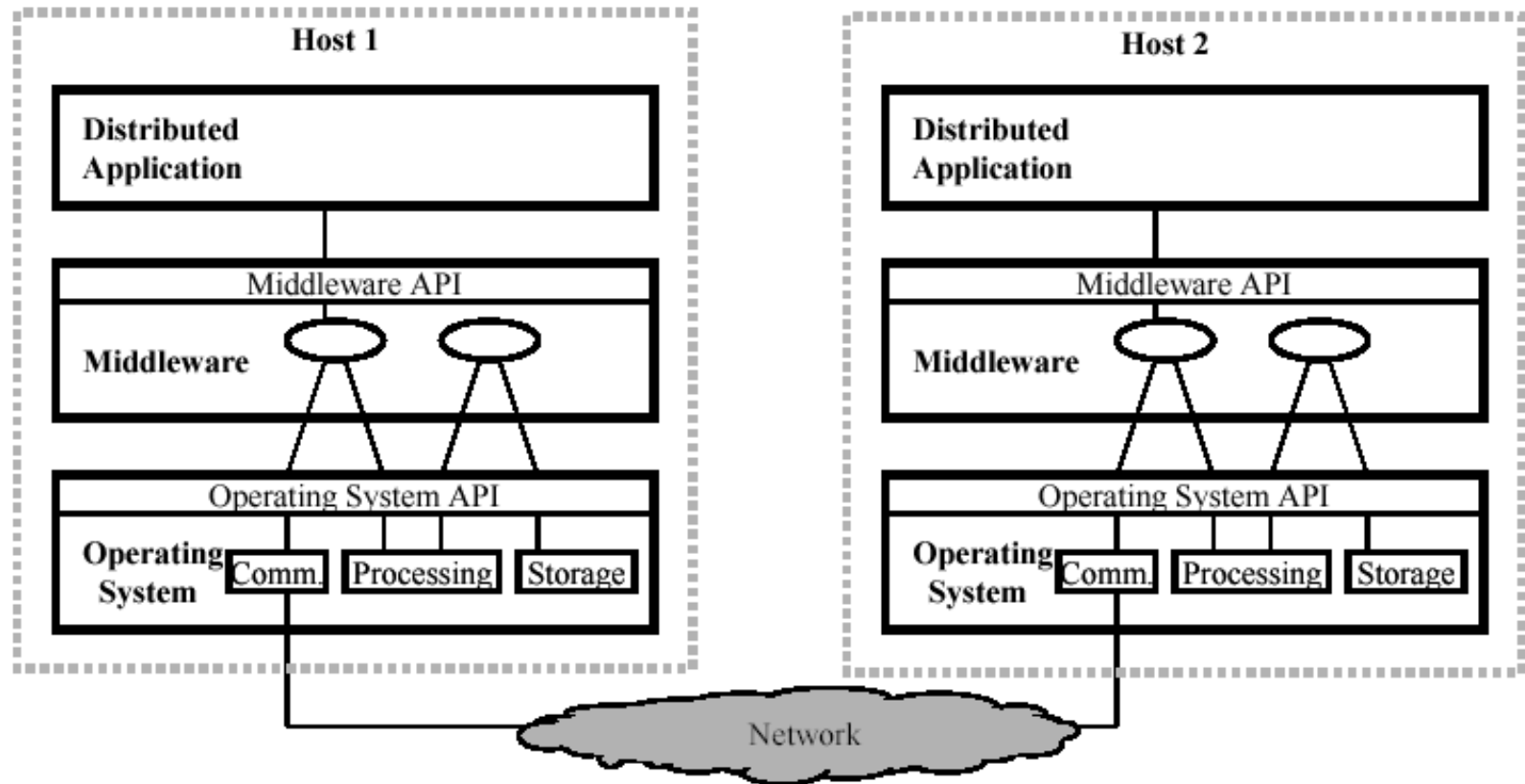


Figure 1: Middleware Layer in Context



# 企业计算

## ∞中间件 (Middleware)

- 1 开放系统
- 2 互操作性
- 3 中间件
- 4 远地过程调用及其增强
- 5 分布式计算环境



# 企业计算

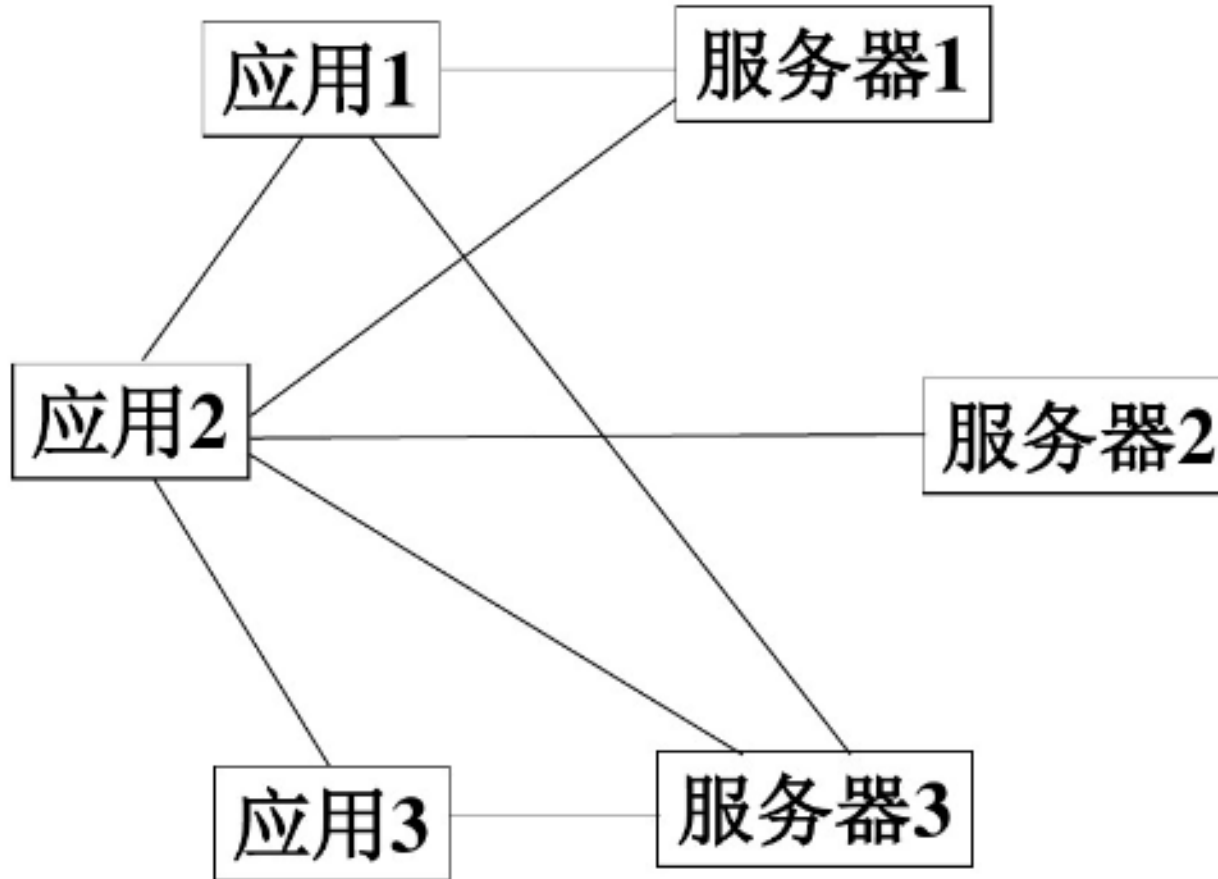
## ∞中间件 (Middleware)

- ◆ 中间件位于C/S计算模型的操作系统、网络和数据库之上，应用软件的下层，是连接两个独立应用程序或独立系统的软件；
- ◆ 中间件为上层的应用软件提供运行与开发的环境，帮助用户灵活、高效地开发和集成复杂的应用软件；
- ◆ 分布式应用软件可借助中间件在不同的技术之间共享资源，相连接的系统即使它们具有不同的接口，但通过中间件相互之间仍能交换信息；
- ◆ 中间件屏蔽了底层分布式环境的复杂性和异构性，简化了分布式应用程序的开发，使分布式应用程序的健壮性、可扩展性、可重用性更好。



# 企业计算

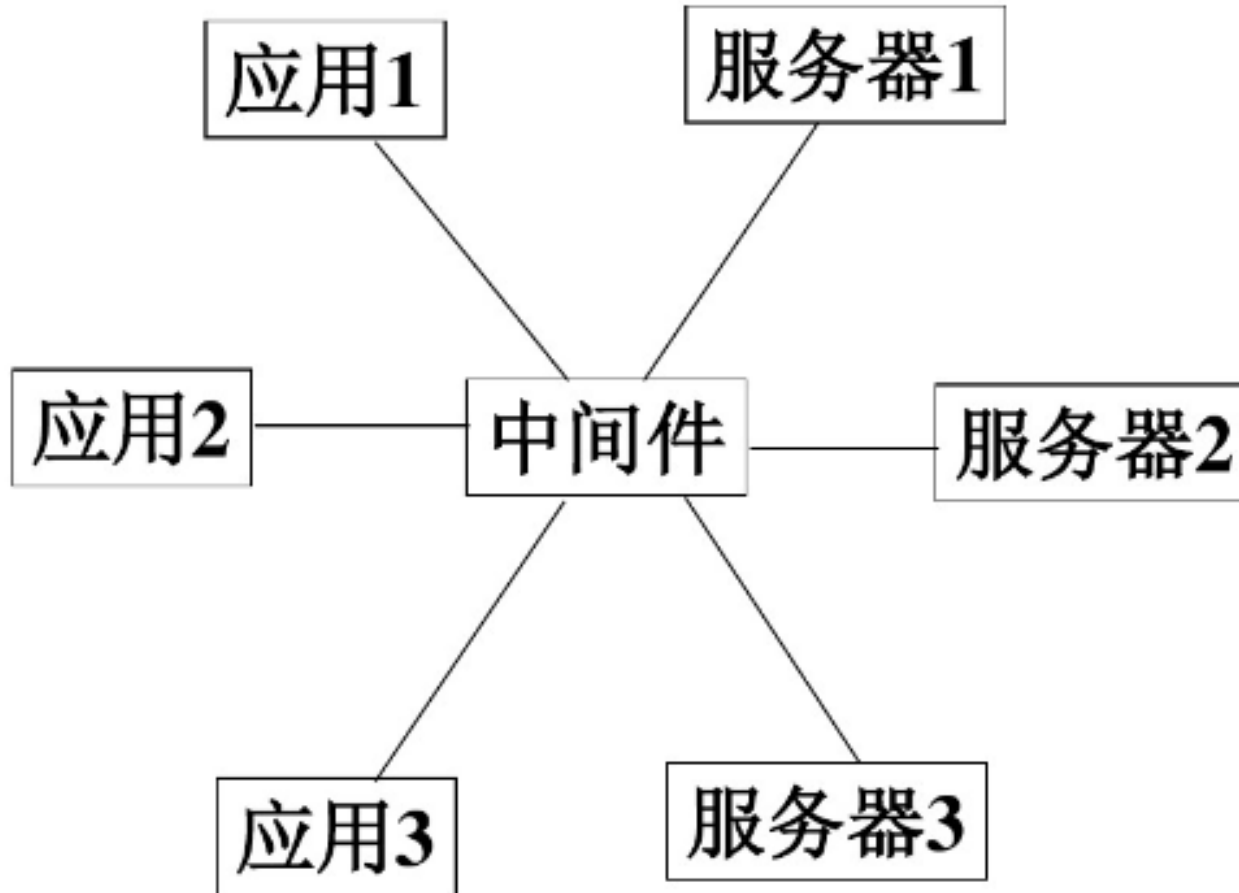
## ☞ 中间件 (Middleware)





# 企业计算

## ☞ 中间件 (Middleware)



# 企业计算

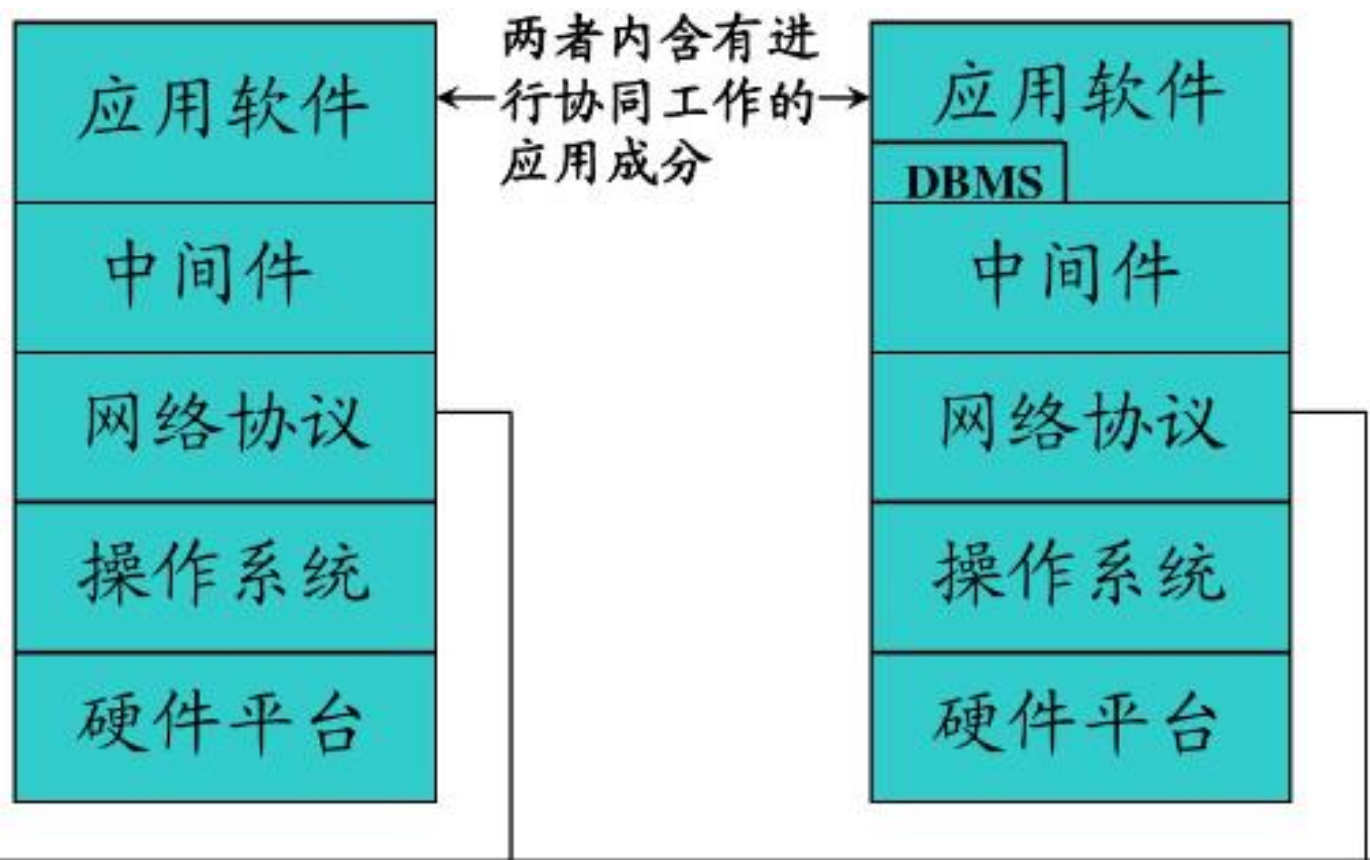
## ∞中间件定义

- ◆ 中间件 (Middleware) 是一种**软件**，处于**系统软件**（操作系统和网络软件）与**应用软件**之间，它能使处于应用层中的各应用成分之间实现跨网络的协同工作（也就是**互操作**），这是运行各应用软件之下所涉及的“**系统结构、操作系统、通信协议、数据库和其他应用服务**”各不相同。



# 企业计算

## ☞ 中间件定义



中间件定义



# 企业计算

## ☞ 中间件所提供的最基本的系统服务

- ◆ **通信服务**：用户屏蔽低层网络通信接口，为高层应用提供可靠的消息通信和远程访问；
- ◆ **名字服务**：提供全局的名字服务空间，屏蔽低层实体分布的具体物理位置；
- ◆ **存储服务**：提供全局的一致数据访问，集成低层分布的局部文件系统和数据库；
- ◆ **分布式事务服务**：提供分布式事务处理，包括事务管理和调度、负载均衡和失败恢复等；
- ◆ **安全服务**：提供存取访问控制，在中间件层次保证分布式系统的安全性。



# 企业计算

## ∞中间件的作用

- ◆ 提供了**通道**功能，可保证应用**系统数据**的可靠传输；
- ◆ 起着**桥梁**作用，达成**异构应用资源**之间的互联互通；
- ◆ 是分布式应用集成**开发框架**，促成企业分布式应用的**有机集成**；
- ◆ 作为**平台**，可有效实现分布应用的**分阶段**部署；
- ◆ 起着**过滤网**作用，可有力保护**关键信息**的安全。



# 企业计算

## ∞中间件的优越性

- ◆ **企业应用方面：** 缩短应用开发周期、减少项目开发风险、应用程序质量可维护、增加产品吸引力；
- ◆ **应用程序方面：** 透明地同其他应用程序交互、与运行平台提供的网络通信服务无关、具有良好的可用性和可靠性、具有良好的可扩展性。



# 企业计算

## ∞中间件技术的主流标准

- ◆ **DNA (Distributed Internet Applications) 标准。**微软在推出Windows2000系列操作系统平台的基础上，扩展了分布式计算模型以及改造Back Office系列服务器端分布计算产品后发布的分布式计算体系结构和规范。
- ◆ **DNA架构由三层组成：**表示层，可以由浏览器组成的客户，也可以是具有有一定商业逻辑处理能力的胖客户；商业逻辑层，由COM/COCOM+/COM等组件组成；数据层，可以是数据库，也可以是外部应用程序。



# 企业计算

## ☞ 中间件技术的主流标准

- ◆ **CORBA (Common Object Request Broker Architecture)**  
标准。它是OMG (Object Management Group) 组织基于众多开放系统, 制订的一种标准的面向对象应用程序体系规范。
- ◆ **CORBA标准主要分为三个层次:** 对象请求代理, 使得在分布式环境下的对象之间能够透明的发送请求和接收响应; 对象服务, 为使用 and 实现对象而提供的基本对象集合, 这些服务 (事件服务、生命周期服务、关系服务等) 应独立于应用领域; 公共设施, 是能被多个应用共享的一系列服务。





# 企业计算

## ☞ 中间件技术的主流标准

- ◆ **J2EE (Java 2 Platform, Enterprise Edition) 标准。**它是Sun公司把企业用户、厂商及技术专家聚集到JCP (Java Community Process) 开发的企业级Java API标准，以此来简化在瘦客户级环境下的应用开发。
- ◆ **J2EE核心是一组技术规范与指南**，其中所包含的各类组件、服务架构及技术层次，均有共同的标准及规格，让各种依循J2EE架构的不同平台之间，存在良好的兼容性，解决过去企业后端使用的信息产品彼此之间无法兼容，企业内部或外部难以互通的窘境。



# 企业计算

性能	CORBA	J2EE	DNA
基本组件模型	CORBA组件	EJB	COM/DCOM
通信协议	IDP	RM I/IDP	ORPC
接口描述	DL	Java	M DL
调用方式	发现服务	JND I	注册表
跨平台型	好	好	差 (限Windows)
网络通信	好	好	一般
可重用性	好	好	好
事务处理	好	一般	一般
安全服务	好	好	一般
目录服务	好	一般	一般
软件开发商支持	少	多	较多
可扩展性	好	好	一般
开发工具	多	多	Microsoft Dev Studio



# 企业计算

## ☞ J2EE优势

- ◆ **保留现有IT资产：**基于J2EE平台的产品几乎能够在任何操作系统和硬件配置上运行，现有的操作系统和硬件也能被保留使用；
- ◆ **高效的开发：**J2EE允许公司把一些通用的、很繁琐的服务端任务交给中间供应商去完成，相应地缩短了开发时间；
- ◆ **支持异构环境：**基于J2EE的应用程序不依赖任何特定操作系统、中间件、硬件；
- ◆ **可伸缩性：**基于J2EE平台的应用程序可允许多台服务器集成部署到各种操作系统上，可达数千个处理器实现可高度伸缩的系统；
- ◆ **稳定的可用性：**可选择鲁棒性好的操作系统如Sun Solaris、IBM OS/390。



# 企业计算

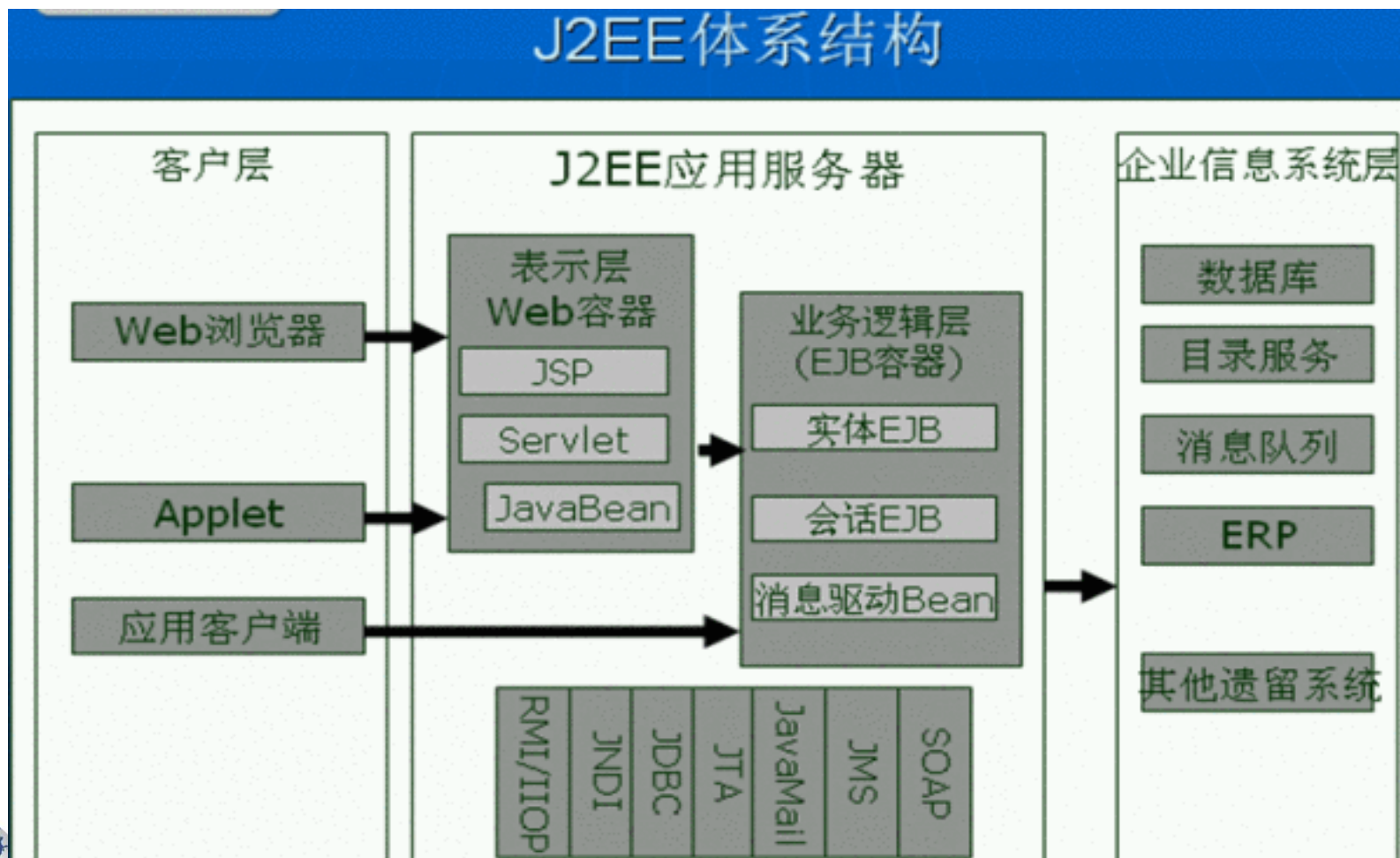
## ☞ J2EE体系结构

- ◆ Sun设计J2EE的初衷正是为了解决两层C/S模式的弊端，在传统模式中，客户端担当了过多的角色而显得臃肿，在这种模式中，第一次部署的时候比较容易，但难于升级或改进，可伸展性也不理想，而且经常基于某种专有的协议（通常是某种数据库协议），使得重用业务逻辑和界面逻辑非常困难。**J2EE 的多层企业级应用模型将两层化模型中的不同层面切分成许多层**，一个多层化应用能够为不同的每种服务提供一个独立的层。



# 企业计算

## ☞ J2EE体系结构——四层结构



# 企业计算

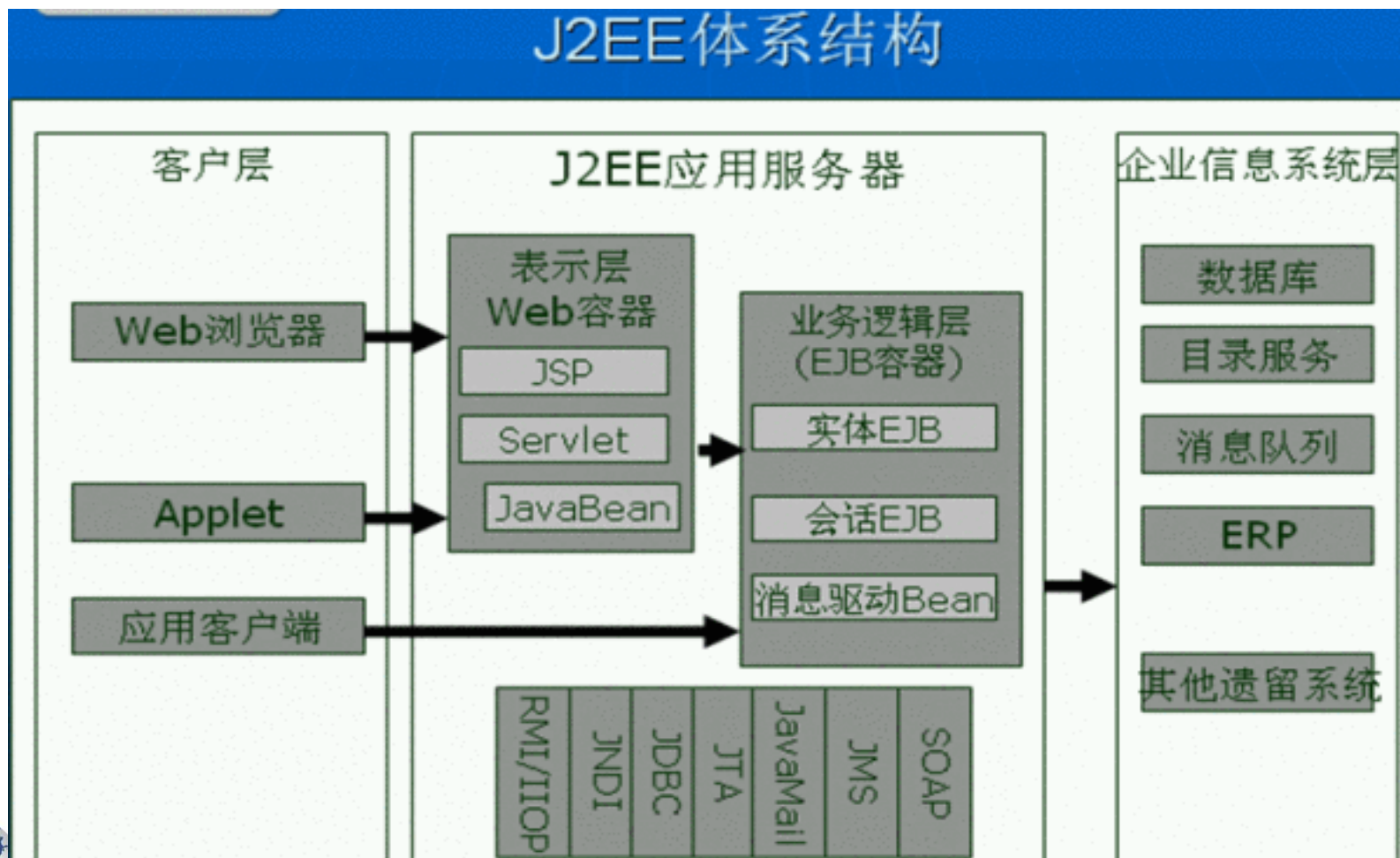
## ☞ J2EE体系结构——四层结构

- ◆ J2EE使用多层的分布式应用模型，应用逻辑按功能划分为组件，  
**各个应用组件根据他们所在的层分布在不同的机器上。**包括：
- ◆ 运行在客户端机器上的客户层组件；
- ◆ 运行在J2EE服务器上的Web层组件；
- ◆ 运行在J2EE服务器上的业务逻辑层组件；
- ◆ 运行在EIS服务器上的企业信息系统（Enterprise information system）层软件。



# 企业计算

## ☞ J2EE体系结构——四层结构



# 企业计算

## ☞ J2EE体系结构——四层结构

- ◆ **客户层组件**：可以是基于web方式的，也可以是基于传统方式。
- ◆ **Web层组件**：可以是JSP 页面或Servlets，还可能包含某些JavaBean对象来处理用户输入，并把输入发送给业务层上处理。
- ◆ **业务层组件**：会话bean 表示与客户端程序的临时交互，当客户端程序执行完后，会话bean和相关数据就会消失；实体bean表示数据库的表中一行永久的记录，当客户端程序中止或服务器关闭时，就会有潜在的服务保证实体bean的数据得以保存；消息驱动bean 结合了会话bean和JMS的消息监听器的特性，允许一个业务层组件异步接收JMS 消息。
- ◆ **信息系统层**：处理企业信息系统软件。





# 企业计算

## ☞ J2EE 体系结构

- ◆ 支持以组件化的方法开发、部署、管理多层结构的分布式企业应用；
- ◆ 提供一种统一的、开放标准的多层平台；
- ◆ 组件依赖系统级实体——容器（Container），容器为组件提供特定的服务；
- ◆ 容器提供：生命周期管理、组件的部署、为组件的运行指派线程，不同类型容器提供不同的服务；
- ◆ 许多组件行为都可以再部署时以声明的方式定制。



# 企业计算

## ☞ J2EE体系——13种核心技术规范

- ◆ JDBC(Java Database Connectivity)
- ◆ JNDI(Java Name and Directory Interface)
- ◆ EJB(Enterprise JavaBean)
- ◆ RMI(Remote Method Invoke)
- ◆ Java IDL/CORBA
- ◆ JSP
- ◆ Java Servlet
- ◆ XML
- ◆ JMS

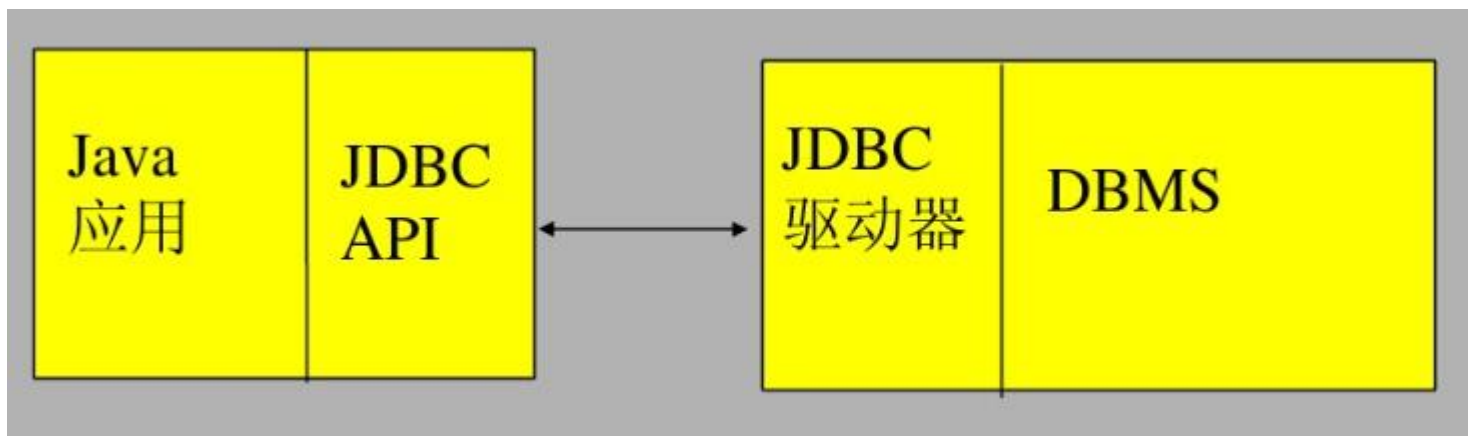
...



# 企业计算

## ☞ J2EE体系——JDBC

- ◆ **JDBC(Java Database Connectivity)** : API为访问不同数据库提供了统一的路径, JDBC开发者屏蔽了一些细节问题。另外, JDBC对数据库的访问也具有平台无关性。



# 企业计算

## ☞ J2EE体系——JNDI

- ◆ **JNDI(Java Name and Directory Interface):** API 被用于执行名字和目录服务。它提供了一致的模型来存取和操作企业级的资源DNS和LDAP(Lightweight Directory Access Protocol), 本地文件系统, 或应用服务器中的对象。



# 企业计算

## ∞J2EE体系——EJB

- ◆ **EJB(Enterprise JavaBean):** J2EE技术之所以赢得广泛重视的原因之一就是EJB。它提供了一个框架来开发和实施分布式业务逻辑，由此很显著的简化了具有可伸缩性和高度复杂的企业级应用程序的开发。EJB规范定义了EJB组件在何时如何与它们的容器进行交互作用。容器负责提供公用的服务，例如目录服务，事务管理，安全性，资源缓冲池以及容错性。但这里值得注意的是，EJB并不是实现J2EE的唯一路径。正是由于J2EE的开放性，使得所有的厂商能够以一种和EJB平行的方式来达到同样的目地。



# 企业计算

## ∞J2EE体系——RMI

- ◆ **RMI(Remote Method Invoke):** 远程方法请求, RMI协议调用远程对象上的方法。它使用了序列化的方式在客户端和服务端之间传递数据。RMI是一种被EJB使用的更底层的协议。

## ∞J2EE体系——Java IDL/CORB

- ◆ **Java IDL/CORBA (通用对象请求代理架构是软件构建的一个标准)** : 在Java IDL的支持下,开发人员可以将Java和CORBA集成在一起。他们可以创建Java对象并使之可在CORBA ORB中展开,或者他们还可以创建Java类并和其它ORB一起展开的CORBA对象客户。后一种方法提供了另外一种途径,通过它Java可以被用于将你的新的应用程序和旧的系统集合在一起。



# 企业计算

## ∞J2EE体系——JSP

- ◆ **JSP:** JSP页面由HTML代码和嵌入其中的Java代码组成。服务器在页面被客户端所请求以后对这些Java代码进行处理,然后将生成的HTML页面返回给客户端浏览器。

## ∞J2EE体系——Java Servlet

- ◆ **Java Servlet:** Servlet 是一种小型的Java程序, 它扩展了web服务器的功能。作为一种服务器的应用, 当被请求时开始执行, 这和CGI Perl脚本很相似。Servlet提供的功能大多和JSP类似, 不过实现的方式不同。JSP通常是大多数的HTML代码中嵌入少量的Java代码, 而servlet全部由java写成并且生成HTML。



# 企业计算

## ∞J2EE体系——XML

- ◆ **XML (Extensible Markup Language)** : 可扩展标记语言是一种可以用来定其它标记语言的语言。它被用来在不同的商务过程中共享数据。XML的发展和java是相互独立的, 但是, 它和java具有的相同目标是平台独立性。

## ∞J2EE体系——JMS

- ◆ **JMS (Java Message Service)** : MS是用于和面向对象消息的中间件相互通信的应用程序接口。它既支持点对点的域, 又支持发布/订阅类型的域, 并且提供了下列类型的支持: 消息传递, 事务型消息的传递, 一致性消息和具有持久性的订阅者支持。JMS还提供了另一种方式来对新系统和旧后台系统相互集成。





# 企业计算

## ☞ J2EE体系——JTA、JTA、JavaMail、JAF

- ◆ **JTA (Java Transaction APIXML)**: JTA定义了一种标准API，应用程序由此可以访问各种事务监控。
- ◆ **JTS**: JTS 是一个组件事务监视器，为应用服务器、资源管理器、独立的应用以及通信资源管理器提供服务。
- ◆ **JavaMail** : JavaMail是用于存取邮件服务器的API，它提供了一套邮件服务器的抽象类。不仅支持SMTP服务器，也支持IMAP服务器。
- ◆ **JAF(JavaBeans Activation Framework)**: JavaMail利用JAF来处理MIME编码的邮件附件。MIME的字节流可以被转换成java对象，大多数应用都可以不需要直接使用JAF。

