

中间件技术基础

徐悦甡

计算机科学与技术学院

西安电子科技大学





- 从分布式软件到中间件
- 对象与构件
- 中间件技术发展历程
- 中间件类型
- 中间件提供的支撑



□ 软件面临的问题

■ 软件正变得无处不在，同时软件所面临的挑战也正在日益加剧

- 复杂度**高**
- 开发周期**长**
- 可靠性保证**难**

■ 本课程关注**分布式软件系统的开发支撑**

- 随着网络与通信技术的发展，分布式软件的应用越来越广泛，分布式软件在计算机软件应用领域扮演着非常重要的角色
- 分布式软件一般比集中式软件规模大、复杂，是软件开发复杂性的集中体现



□ 分布式软件

- **网络环境**：一群通过网络互相连接的处理系统，每个处理节点由处理机硬件、操作系统及基本通信软件等组成
- **分布式软件**：运行在**网络环境**中的软件系统
- 两种典型的应用途径
 - 将分布式软件系统看作直接反映了现实世界中的分布性
 - 用于改进某些应用程序的运行性能
 - 单机或者串行已不能满足性能的要求



□ 当前的分布式软件通常基于**客户机/服务器**模型

■ Client/Server

□ 客户机提出对信息或服务的请求，服务器提供这种信息或服务

■ 应用举例

➤ 电子邮件客户程序与服务程序

➤ FTP客户程序与服务程序

➤ 执行调用语句的子程序与实现函数/方法体的子程序

□ 强调客户端与服务端的分离

■ 分布式软件系统中，客户端与服务端的分离带来很多好处

- 更好的支持平台无关性
- 更好的可扩展性

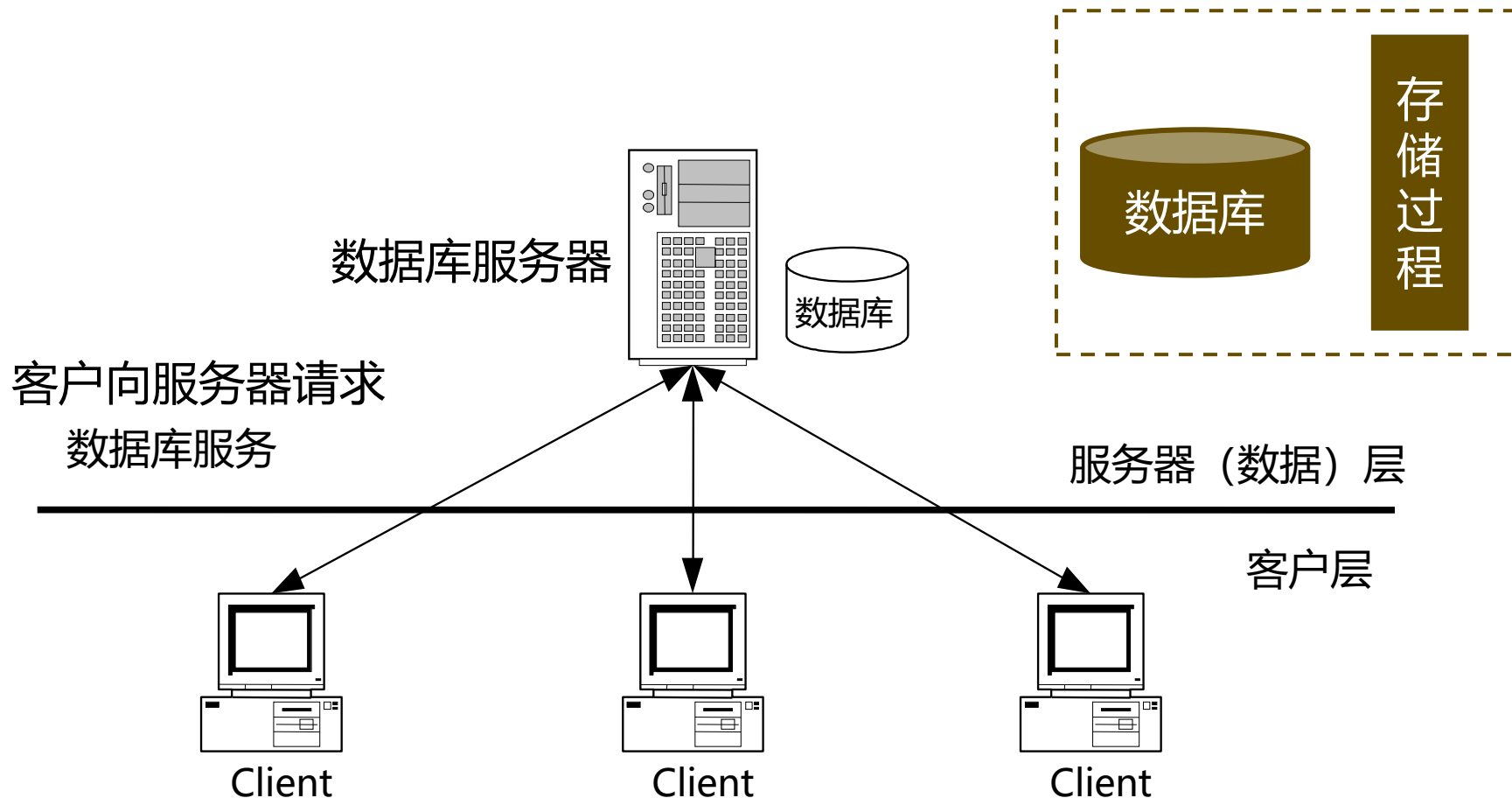
■ 客户端与服务端的分离使得软件系统更复杂

- 开发人员不得不分别编写客户端和服务端应用程序，并力求保持两者的一致性
- 软件系统的调试、部署、维护更加困难
- 需考虑更多的可靠性、安全性、性能等软件质量因素

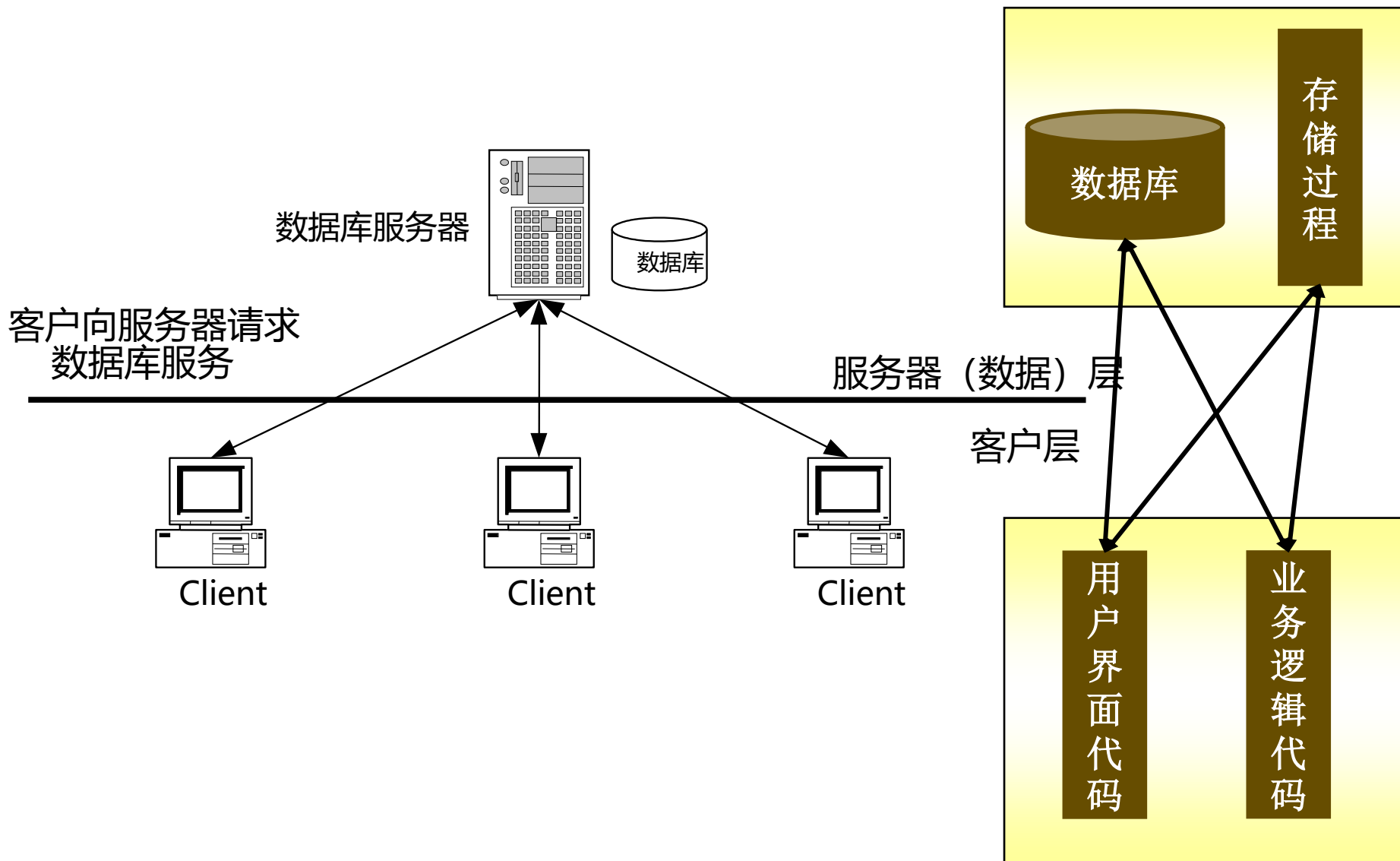
传统的客户机/服务器结构



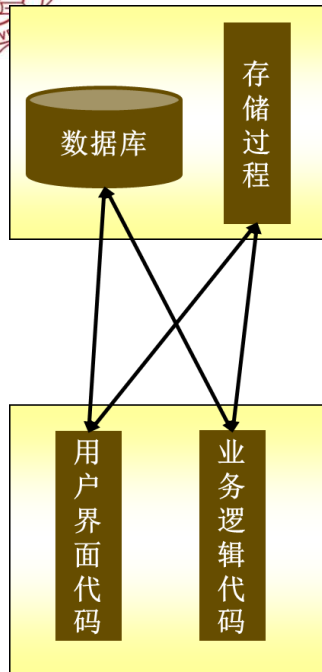
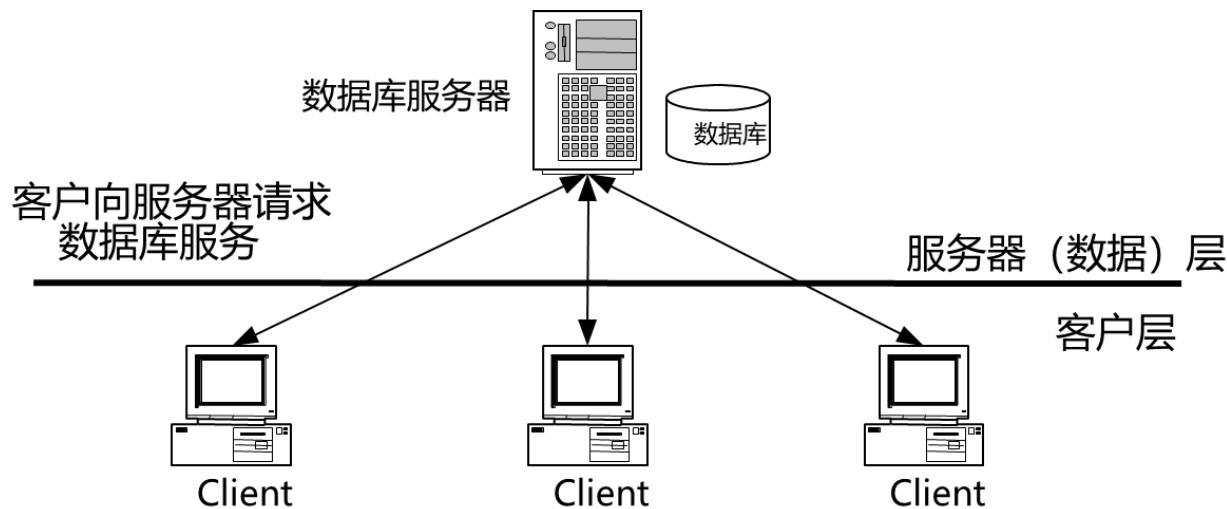
西安电子科技大学
XIDIAN UNIVERSITY



2层结构的缺陷



2层结构的缺陷



❑ 缺点

■ 客户端的负担仍比较重

- 仍然需要客户端进行较复杂的数据处理

■ 客户端的可移植性不好

- 处理复杂必然牵涉更多的移植性问题
- 每个客户端上都要安装数据库驱动程序

■ 系统的可维护性不好

- 客户端包含过多的商业逻辑
- 商业逻辑与人机交互界面交织在一起

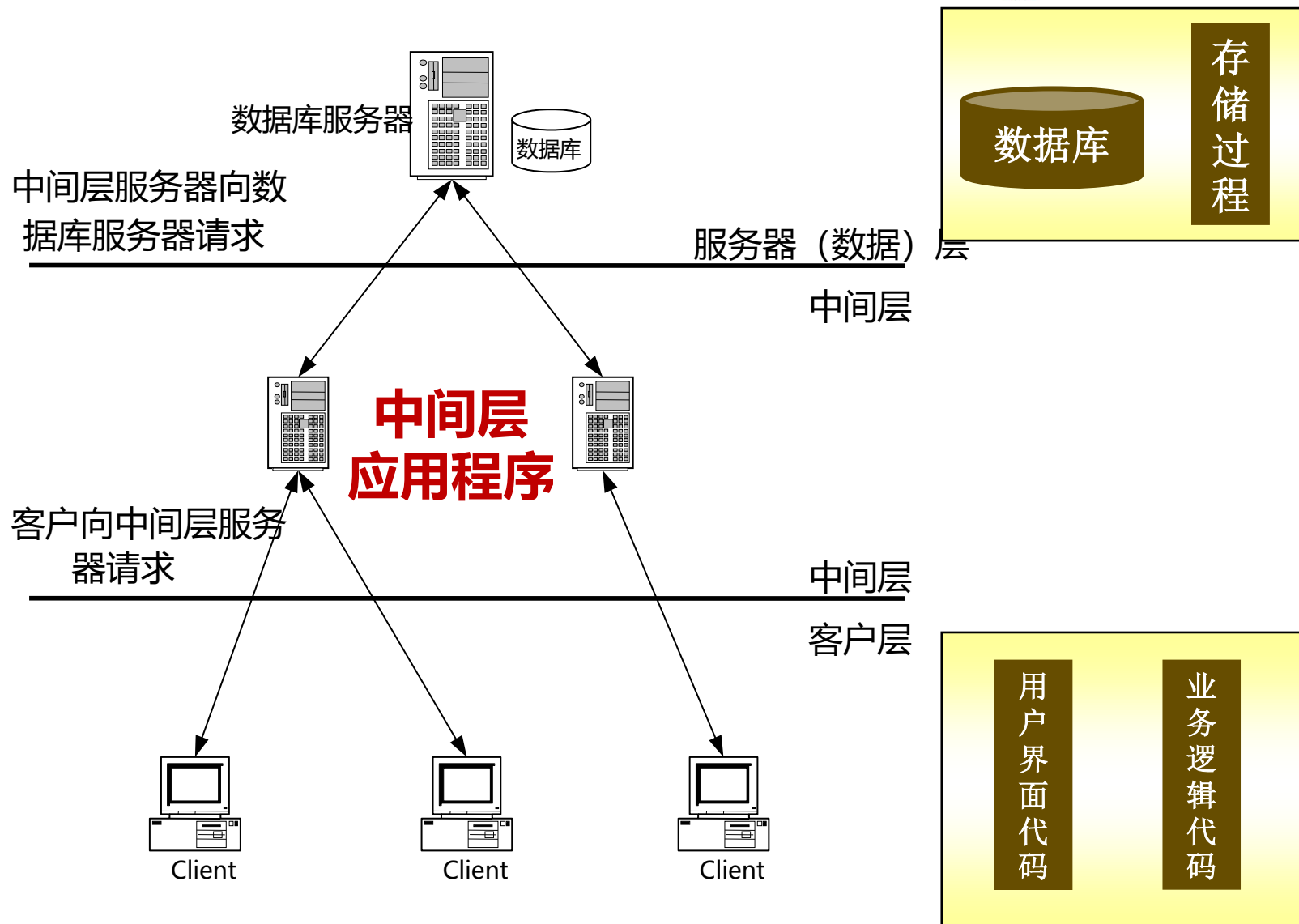
■ 数据的安全性不好

- 需要更合理的工作分配 3 层或多层结构

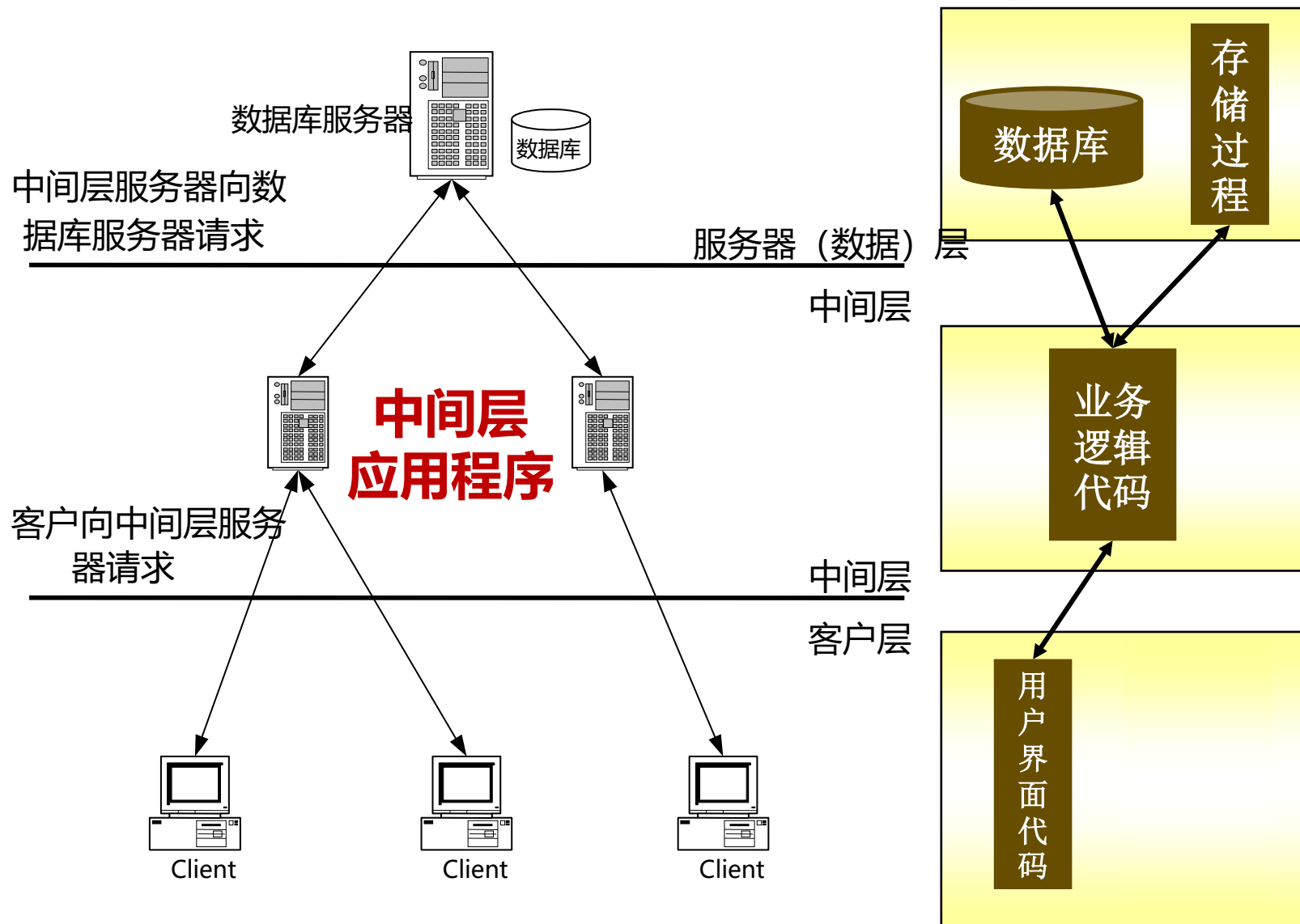
典型的3层结构



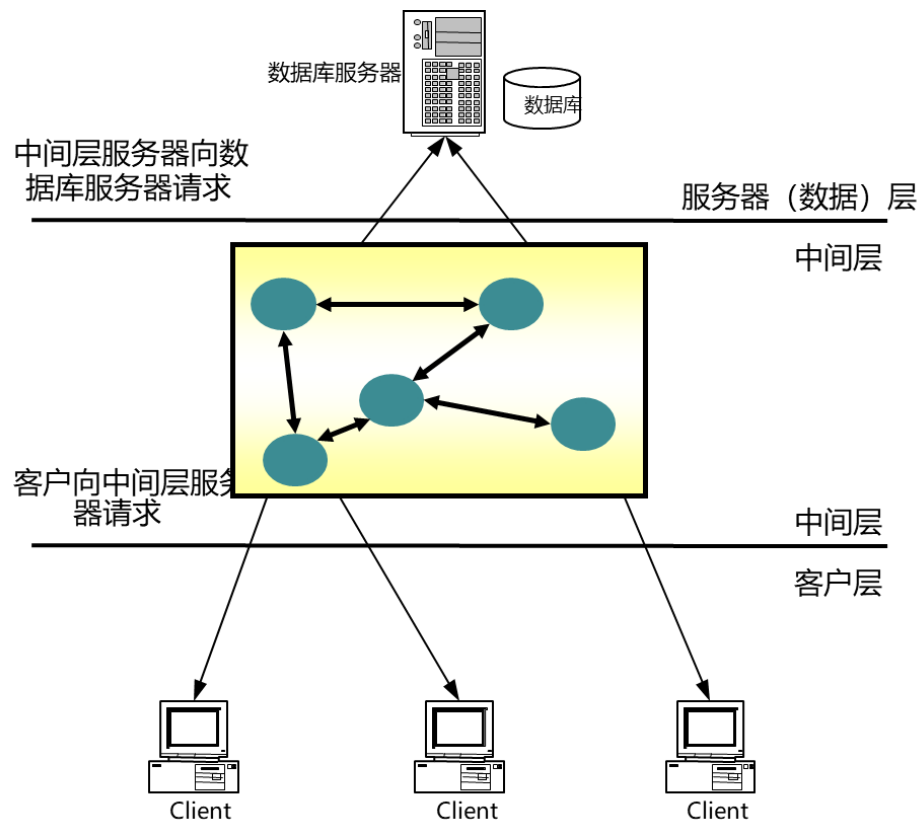
西安电子科技大学
XIDIAN UNIVERSITY



典型的3层结构



3层结构 vs. 2层结构



□ 优点

- 减轻了客户端的负担
- 提高了客户端的可移植性
- 增强了系统的可维护性
- 增强了数据的安全性
- 更好的性能与可伸缩性
- 大量的中间层中间件平台提供丰富的**系统级服务**
 - 使得开发人员可以以更少的工作量开发出更复杂、可靠、高效的软件系统



- 从分布式软件到中间件
- 对象与构件
- 中间件技术发展历程
- 中间件类型
- 中间件提供的支撑

□ 传统对象的关注点

- 封装和通过继承对实现进行重用
- 封装提供了一种将对象实现细节与其他对象屏蔽开的严格方法
 - 可以大大缓解在面向过程系统中较突出的维护问题
- 继承提供了一种重用对象实现的简便方法

□ 分布式环境要求更好的可插入性

- 要求另一层次上的封装，只需暴露公用接口
- 不太关注于直接重用代码，而是要求能够利用远程所实现的服务。

□ 对象 → 构件（分布式对象）



□ 相同点

- OO (Object Oriented) 的精髓在于对象维护自身的状态并通过消息进行通信
 - 这对于分布式计算环境是非常理想的

□ 不同点

- 很多分布式对象体系结构都实现了与语言的无关性
- 分布式对象最重要的方面在于
 - 分布式对象可以在任何粒度上透明的使用, 也就是说, 无需考虑位置与实现
- 分布式对象比远程过程调用 (Remote Method Invocation) 有更多的优点



□ 什么是软件构件

- **构件**指系统中可以明确辨识的构成成分
- **软件构件**指软件系统中具有一定意义的、相对**独立**的构成成分，是可以被**重用**的软件实体

□ 产业工程化、工业化的模式

- 符合标准的零部件（构件）生产与基于标准零部件的产品生产（组装）
- 复用是必需的手段，而构件是核心和基础

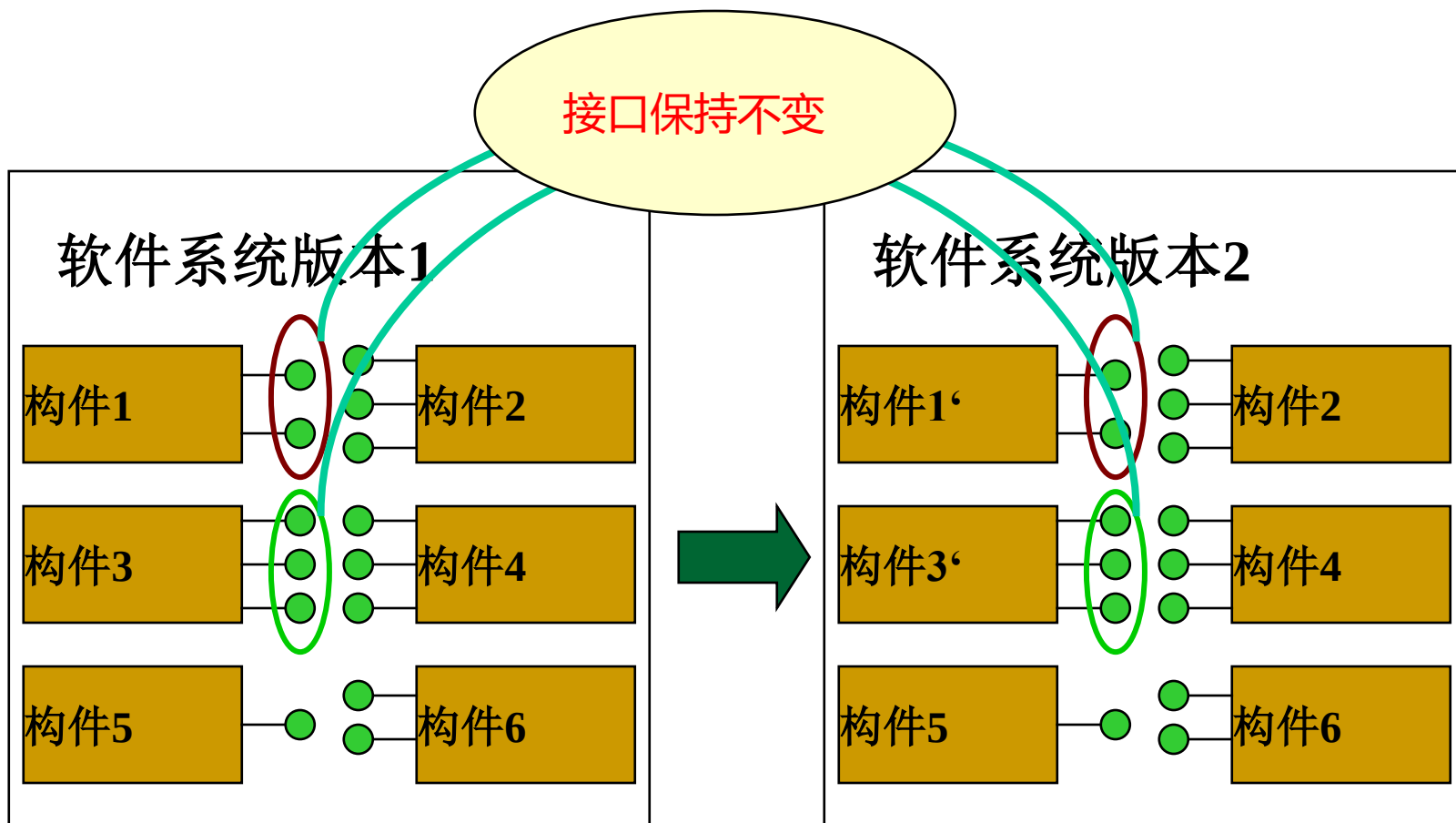
□ 软件构件提供了**软件重用**的基本支持

- 构件是一个严格定义的可插入单元
 - 构件一般是基于对象实现的，但也可以不作为对象实现
- 构件将封装运用到了极限
 - 构件通过封装来隐藏构件的实现以达到
 - 构件的实现语言是未知的：
 - 一个Java客户不会感觉到所使用的构件是由C++实现的
 - 构件的物理位置是未知的：
 - 一个VB客户不会感觉到所使用的构件是运行在相同的进程内（使用DLL），还是在同一机器的不同进程内，甚至是位于不同机器上

构件化软件系统的升级方式



西安电子科技大学
XIDIAN UNIVERSITY





- 从分布式软件到中间件
- 对象与构件
- 中间件技术发展历程
- 中间件类型
- 中间件提供的支撑

中间件出现的动机



西安电子科技大学
XIDIAN UNIVERSITY

■ 中间件出现的驱动力主要来自软件开发过程碰到的严重问题

- 业务需求不断的变化、系统不断增加、流程更复杂、系统越来越不堪重负，总结为 “**软件危机**” ；
- 解决软件危机的过程中，发展起来了软件工程、新的程序设计方法、中间件等**软件开发方法学**、**软件开发技术**与**软件平台**。

出现背景：**软件危机**

- 60年代中期前：通用硬件相当普遍，软件却是**为某个具体应用编写**；软件的编写组织**模式极不规范**，被称为 “**软件作坊**”

导致

- ✓ 对软件开发**成本**和**进度**的**估计不准确**；
- ✓ **用户**对完成的软件系统**不满意**；
- ✓ 软件常常不可维护，**经常修改**；
- ✓ 软件**没有**适当的**文档资料**；
- ✓ **软件成本**在总成本中所占比例**逐年上升**

软件开发所需的**高成本**
同产品的**低质量**之间
存在着尖锐矛盾，导致了
软件危机(Software Crisis)

□ 中间件的动机与目标

■ 屏蔽异构性

- 异构性表现在计算机的软硬件之间的异构性，包括**硬件**（CPU 和指令集、硬件结构、驱动程序等），**操作系统**（不同操作系统的 API 和开发环境）、**数据库**（不同的存储和访问格式）等；
- 高级语言依赖于特定的编译器和操作系统 API 来编程，因此软件必须依赖于开发和运行环境。

■ 实现互操作

- 不同软件之间在不同平台之间不能移植，或者移植非常困难；
- 网络协议和通信机制的不同，这些系统之间还不能有效地相互集成。

■ 软件复用

- 尽可能多地凝练共性并复用以提高软件开发效率和质量，通过中间件通过提供简单、一致、集成的开发和运行环境，简化分布式系统的设计、编程和管理

中间件出现的动机

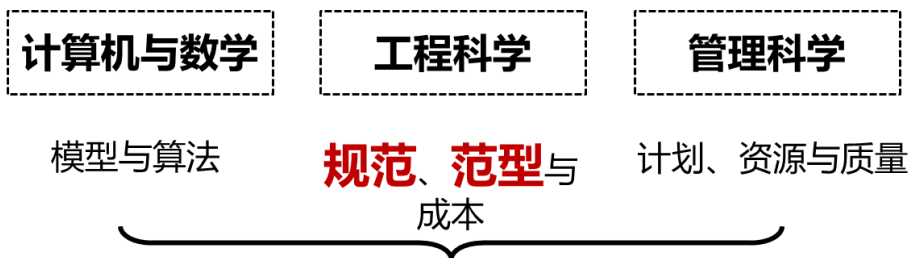


西安电子科技大学
XIDIAN UNIVERSITY

■ 解决方案

在软件开发60余年的历史中，解决软件问题的方案可以总结为三个方面：

➢ 工程方法 → **软件工程**； 软件技术 → **软件开发技术**； 软件平台 → **中间件**



中间件？

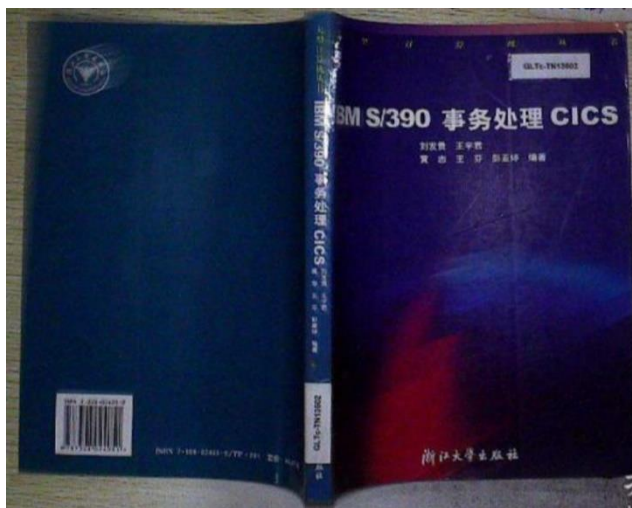


中间件发展历程

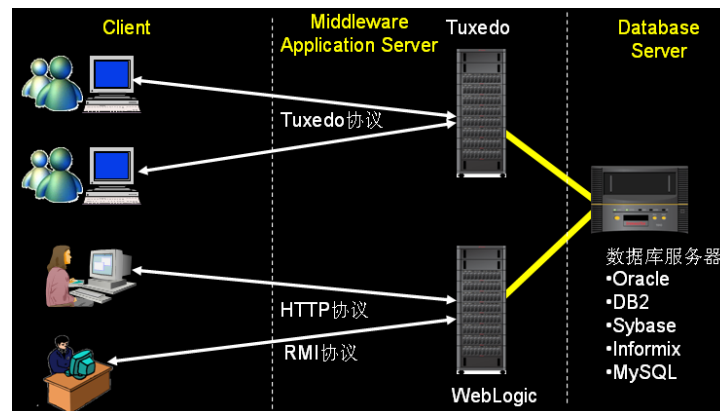


西安电子科技大学
XIDIAN UNIVERSITY

- 1968年 IBM 发布 CICS (Customer Information Control System) 交易事务控制系统，**应用软件**与**系统服务**分离，中间件技术萌芽，因为 CICS 不在分布式环境下，所以一般不将这时的 CICS 正式作为中间件



- 一般将1990 年诞生于 AT&T 公司的 BELL 实验室的Tuxedo 系统作为中间件诞生标志，解决了**分布式交易事务控制**问题，中间件开始成为网络应用的基础设施，**事务 (交易) 中间件**诞生。



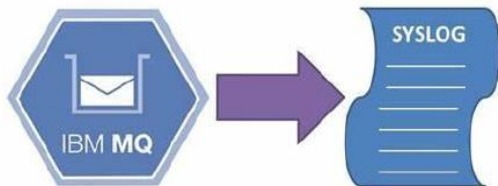
中间件发展历程



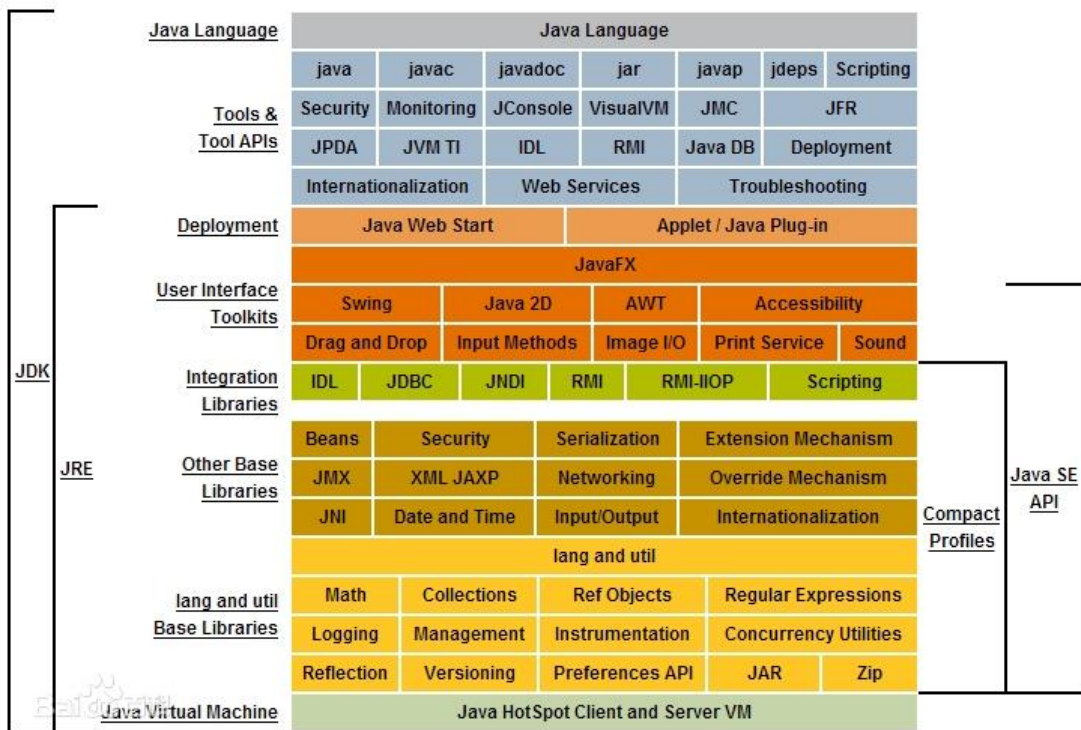
西安电子科技大学
XIDIAN UNIVERSITY

- 1994 年 IBM 发布消息队列服务 MQ 系列产品 解决分布式系统异步、可靠、传输的通讯服务问题，**消息中间件**诞生。

IBM Middleware User Community



- 1995 年，JAVA 语言诞生，JAVA 提供了跨平台的通用的**网络应用服务**，成为今天中间件的核心技术。特别是 J2EE 发布以来，JAVA 从一个编程语言，演变为网络应用架构，成为应用服务平台的事实标准。**应用服务器中间件**，成为中间件技术的集大成者。
- 也间接加速了 CORBA 的衰落



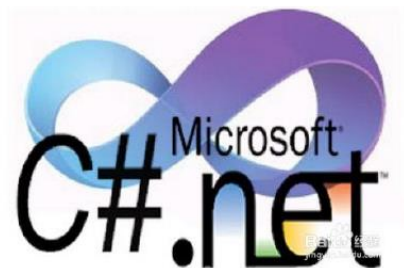
中间件发展历程



西安电子科技大学
XIDIAN UNIVERSITY

- 2001 年，微软发布 .NET，中间件演变为 .NET 和 JAVA 两大技术阵营。但由于 .NET 还不是一个完全开放的技术体系。IBM/ORACLE/SAP 等都成为 JAVA 阵营的支持者，因此，中间件往往不包括 .NET 中间件体系。

.NET – A unified platform



- 面向服务体系结构（Service-oriented Architecture, SOA）的出现，引领了现代中间件开发的新潮流。
- 1996 Gartner 提出 SOA



□ 中间件概念

■ IDC（国际数据公司）：

- 中间件是一种独立的系统软件或服务程序，分布式应用软件借助这种软件在不同的技术之间共享资源，中间件位于客户机服务器的操作系统之上，管理计算资源和网络通信

■ 我国定义：

- 中间件是指网络环境下处于操作系统、数据库等系统软件和应用软件之间的一种起连接作用的**分布式软件**，主要解决异构网络环境下分布式应用软件的互连与互操作问题，提供标准接口、协议，屏蔽实现细节，提高应用系统易移植性。（梅宏院士）

□ 中间件特征

■ 平台性

- “平台”：能够独立运行并自主存在，为其所支撑的上层系统和应用提供运行所依赖的环境。
- 中间件是必须独立存在，是运行时刻的系统软件。
- 比如：Eclipse、IntelliJ、Struts Hibernate 等不是中间件，J2EE 应用服务器则为中间件。

■ 支撑性

- 模型支撑、结构支撑、互操作支撑、开发方法支撑
- 模型支撑：通过抽象层度更高的构件模型，实现具备更高结构独立性、内容自包含性和业务完整性的**可复用构件**
- **结构支撑**：结构松散化，即，将完整分离功能描述和功能实现，以及使用者和提供者
- **互操作支撑**：交互过程标准化
- **开发方法支撑**：应用系统的构建方式由代码编写转为主要通过服务间的快捷组合及编排，完成更为复杂的业务逻辑的按需提供和改善。

■ 复用性

- 软件复用技术发展历程，可以总结为：不断提升抽象级别，扩大复用范围。

SCA: Service Component Architecture
SDO: Service Data Objects

复用对象	复用范围
子程序	一个可执行程序内复用，静态开发期复用
构件(DLL,COM等)	系统内复用，动态运行期复用
企业对象组件 (DCOM,.NET,EJB等)	企业网络内复用，不同系统之间复用
服务 (如Web Service,SCA/SDO, 微服务)	不同企业之间，动态可配置

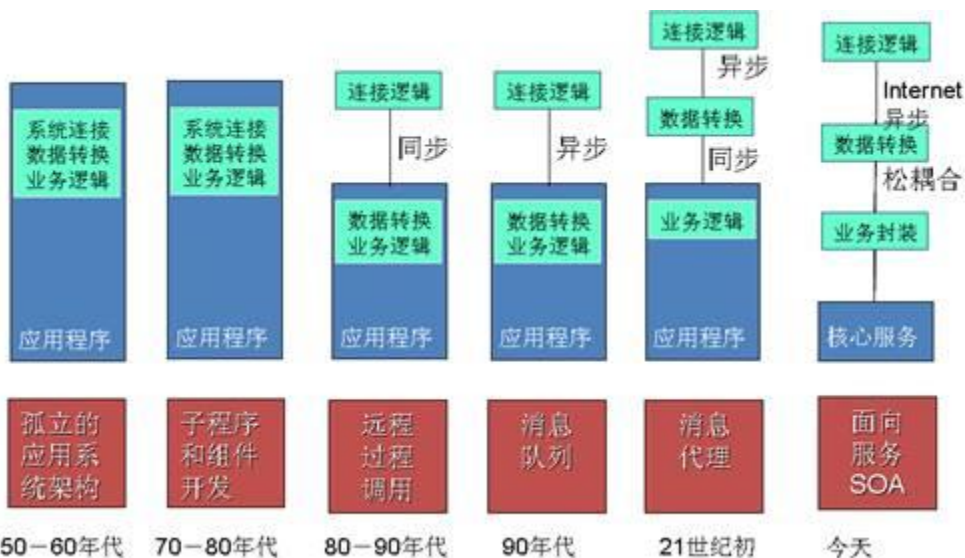
中间件概念与特征



西安电子科技大学
XIDIAN UNIVERSITY

■ 解耦性

➤ 软件之中核心三部分：网络/系统连接、数据操作、业务逻辑



- **网络/系统连接解耦**：消息中间件
- **数据操作解耦**：数据访问中间件、事务（交易）中间件
- **业务逻辑解耦**：SOA与微服务

■ 互操作性

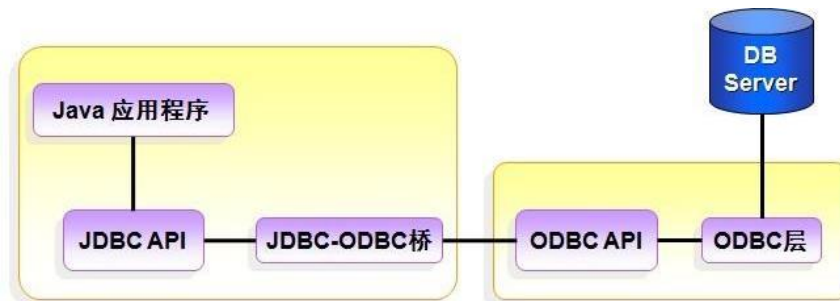
- **访问互操作**：通过标准化的 API 实现了同类系统之间的调用互操作。
 - 存在的问题：访问互操作依赖于特定的访问协议，如EJB使用RMI，CORBA使用IIOP 等。
- **连接互操作**：SOA通过标准的、支持 Internet、与操作系统无关的SOAP协议实现了连接互操作。
 - SOAP: Simple Object Access Protocol



- 从分布式软件到中间件
- 对象与构件
- 中间件技术发展历程
- 中间件类型
- 中间件提供的支撑

□ 中间件

- 在操作系统与应用系统之间的一层软件，为分布式应用的开发、部署、运行与管理提供支持。
- 主要的中间件类型（分类方法不一）包括：
- 1. 数据访问中间件
 - 为了建立数据应用资源互操作的模式
 - 对异构环境下的数据库实现连接或文件系统实现连接的中间件
 - 代表性：



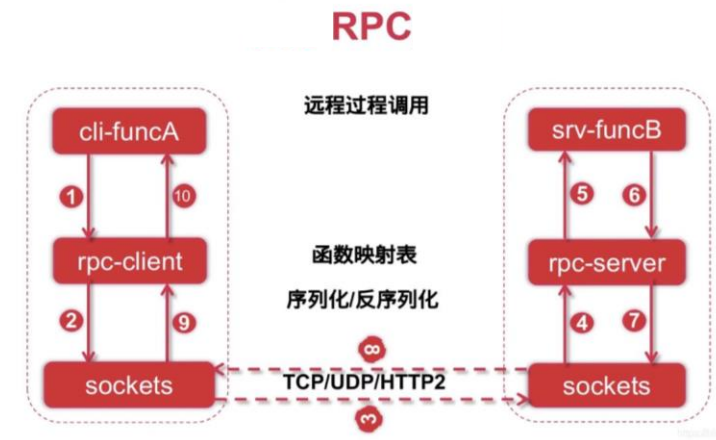
中间件类型



西安电子科技大学
XIDIAN UNIVERSITY

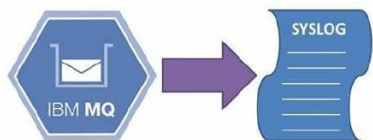
■ 2. 远程过程调用中间件

- Remote procedure call middleware
- 程序员方便地编写客户端应用程序
- 调用位于远端服务器上的过程
- 代表性：

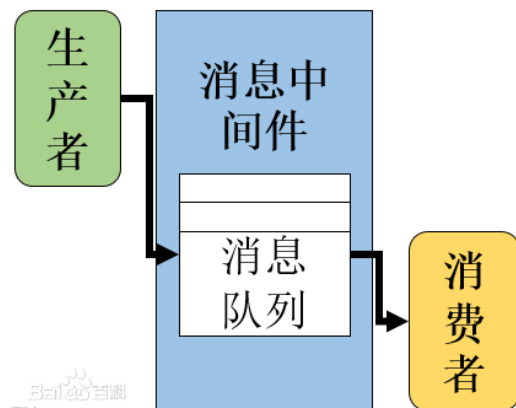


■ 3. 消息中间件

- 用来屏蔽各种平台及协议之间的特性
- 进行**相互（异步）通信**
- 实现应用程序之间的协同



RabbitMQ



中间件类型



西安电子科技大学
XIDIAN UNIVERSITY

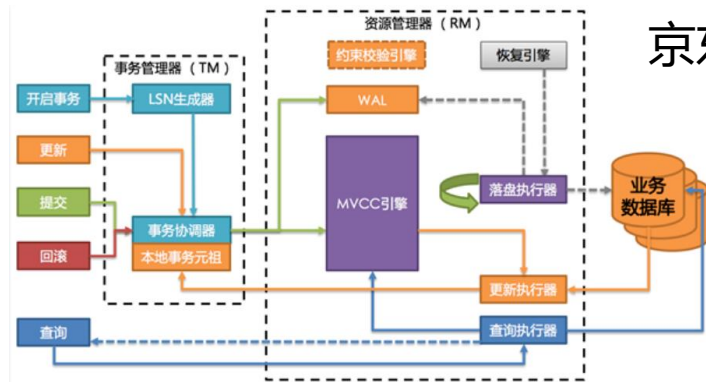
■ 4. 事务（交易）中间件

- 在分布、异构环境下
- 提供保证交易完整性和数据完整性的一种环境平台

ORACLE®
FUSION MIDDLEWARE
TUXEDO

SEATA

Seata: Simple Extensible Autonomous Transaction Architecture



■ 5. 构件中间件（分布式对象中间件）

- 在分布、异构的网络计算环境中
- 将各种分布对象有机地结合在一起
- 完成系统的快速集成
- 实现对象重用

□ 在中间件研究的基础上

- 人们开始考虑将各种中间件的功能集成在一起，现有中间件多以**5. 集成中间件**的形式出现
- 集成中间件在应用中的表现形式主要为**应用服务器**
 - 集成中间件典型地为三层/多层结构的分布式软件系统提供各种开发支撑
 - 因为三层结构的分布式软件的核心为中间层，因此支撑主要集中在对中间层开发的支撑上

□ 目前应用最广泛的**集成中间件**有三类

- 基于OMG (Object Management Group, 对象管理组织) **CORBA**规范的集成中间件
- 基于**Sun/Oracle J2EE** (Java2 Enterprise Edition, Java2企业版) 规范的集成中间件
- 基于微软**.NET**架构的集成中间件

中间件类型



西安电子科技大学
XIDIAN UNIVERSITY

IBM WebSphere

The screenshot shows the IBM WebSphere Integrated Solutions console. The left sidebar contains a navigation tree with categories like Servers, Applications, Services, Resources, Security, Environment, System administration, Users and Groups, Monitoring and Tuning, Troubleshooting, Service integration, and UDDI. The main content area is titled 'Global security' and includes sections for Administrative security, Application security, and Java 2 security. The 'Global security' section contains a 'Security Configuration Wizard' and a 'Security Configuration Report' button. The 'Administrative security' section has a checkbox for 'Enable administrative security' and links to 'Administrative user roles', 'Administrative group roles', and 'Administrative authentication'. The 'Application security' section has a checkbox for 'Enable application security'. The 'Java 2 security' section has a checkbox for 'Use Java 2 security to restrict application access to local resources' and a checkbox for 'Warn if applications are granted custom permissions'. The 'User account repository' section has a 'Realm name' field and a 'Current realm definition' dropdown set to 'Local operating system'. There are 'Apply' and 'Reset' buttons at the bottom of the configuration area.

ORACLE®
WebLogic Server

The screenshot shows the Oracle WebLogic Server Administration Console. The top navigation bar includes 'Home', 'Log Out', 'Preferences', 'Record', 'Help', and a search bar. The main content area is titled 'Summary of Servers' and includes a 'Configuration' tab and a 'Control' tab. A message states: 'All changes have been activated. No restarts are necessary. Server created successfully.' Below this is a 'Summary of Servers' section with a table showing the status of servers. The table has columns for Name, Cluster, Machine, State, Health, and Listen Port. The servers listed are AdminServer(admin), Server-0, and Server-1, all in the cluster-0 cluster. The State column shows AdminServer(admin) is RUNNING, while Server-0 and Server-1 are SHUTDOWN. The Health column shows AdminServer(admin) is OK. The Listen Port column shows AdminServer(admin) is 7001, and Server-0 and Server-1 are 7002. There are 'New', 'Clone', and 'Delete' buttons for each server row. The bottom of the console shows a 'Find' field with 'TODO' and a 'Done' button.

Name	Cluster	Machine	State	Health	Listen Port
AdminServer(admin)			RUNNING	OK	7001
Server-0	cluster-0		SHUTDOWN		7002
Server-1	cluster-0		SHUTDOWN		7002



- 从分布式软件到中间件
- 对象与构件
- 中间件技术发展历程
- 中间件类型
- 中间件提供的支撑

中间件提供的支撑



西安电子科技大学
XIDIAN UNIVERSITY

□ 支撑

- 提供构件运行环境 → 屏蔽异构性
- 提供互操作机制 → 互操作
- 提供公共服务 → 软件复用



□ 支撑

■ 提供构件运行环境

■ 提供互操作机制

■ 提供公共服务

■ 中间件一般通过构件容器为构件提供基本的运行环境，具体功能包括

- 管理构件的**生命周期**
- 管理构件的**实例**
- 管理构件的**元信息**等

□ 支撑

- 提供构件运行环境

- 提供互操作机制

- 提供公共服务

- 集成中间件都提供了很强的**高层通信协议**

 - 屏蔽节点的物理特性以及各节点在处理器、操作系统等方面的异构性

- 基于中间件的互操作支持，开发人员在开发与调用分布式对象时，均不需自己编写处理底层通信的代码

■ 高层通信协议

- 远程过程调用 (Remote Procedure Call, RPC)
- IIOP(Internet Inter-ORB Protocol)
- DCOM通信协议 (Distributed Component Object Model)
- JRMP (Java Remote Messaging Protocol)
- RMI (Remote Method Invocation)

- 共同特征就是帮助应用程序完成**编组**与**解组**等跨越网络通信的底层工作

- 实现远程过程/方法调用中间件的功能

中间件提供的支撑



西安电子科技大学
XIDIAN UNIVERSITY

□ 支撑

- 提供构件运行环境
 - 提供互操作机制
 - 提供公共服务
-
- 现有集成中间件将早期各种中间件中针对分布式软件的通用支持集成于一身
 - 以**公共服务**的形式提供给应用程序
 - 公共服务又称为**系统级服务**
 - 指由中间件（应用服务器）实现的、应用程序使用的软件系统中共性程度高的功能成分
 - 由中间件而非应用程序实现
 - 应用程序中通常会调用其实现的**共性**功能
-
- 中间件提供的主要公共服务包括：
 - **命名服务**
 - **事务服务**
 - **安全服务**
 - **持久性服务、消息服务、分布式垃圾回收服务、资源管理服务**等



- 孟宪福. 分布式对象技术及其应用. 清华大学出版社, 2015
- 李文军, 周晓聪, 李师贤. 分布式对象技术. 机械工业出版社, 2004
- 奉继承. 浅析深究什么是中间件?
<http://blog.vsharing.com/fengjicheng/A1059912.html>, 2010



感谢!