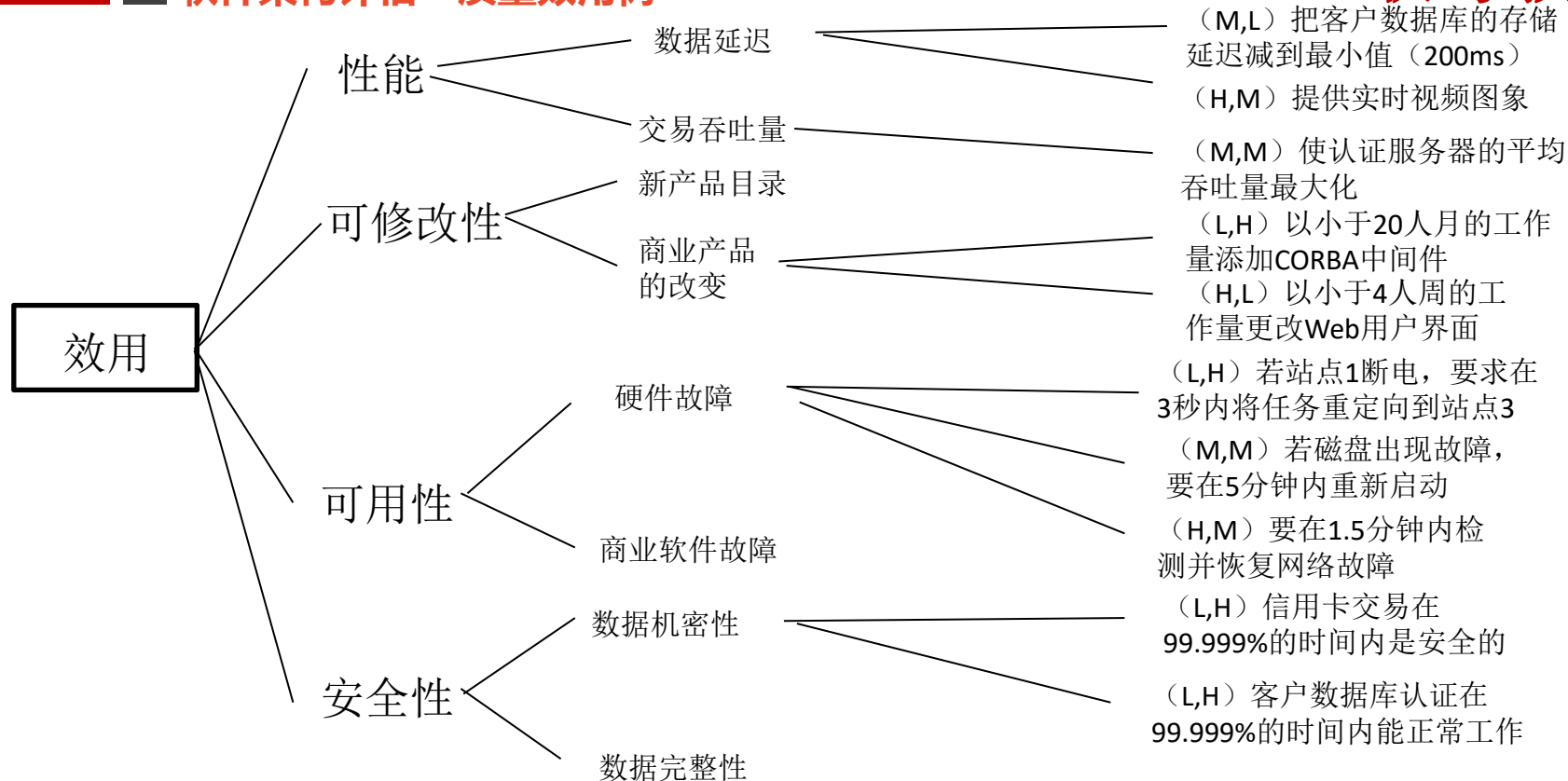




系统架构设计师

DESIGNER: 王川林
软件架构设计



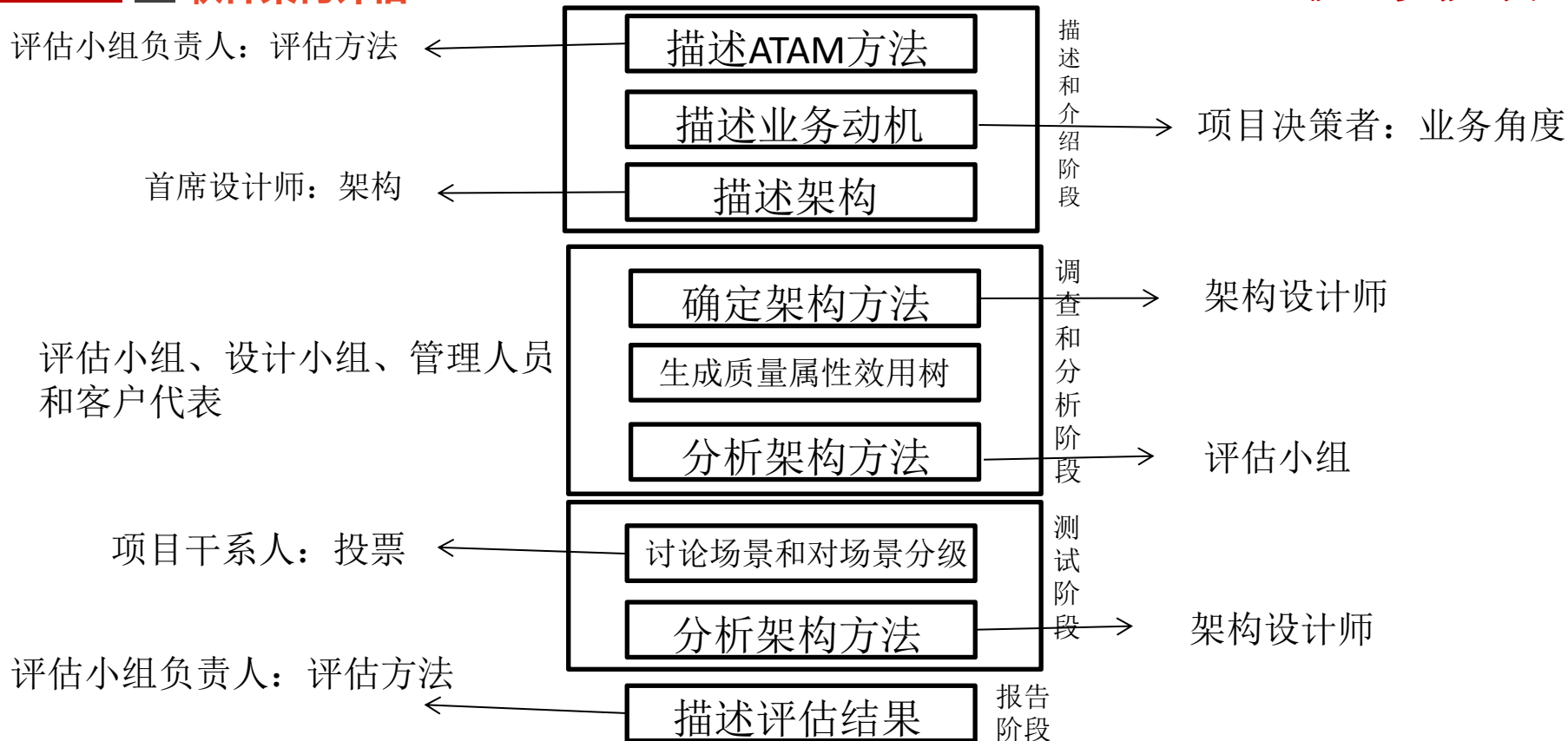


基于场景的评估技术

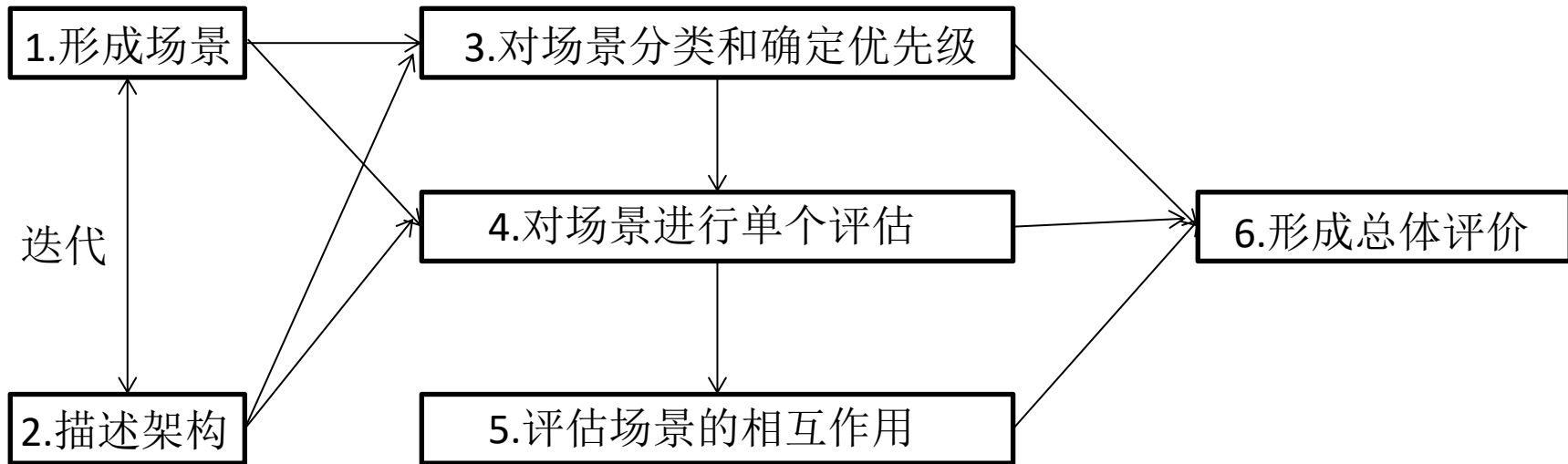
代表性的评估技术有ATAM架构权衡分析法和SAAM软件架构分析法,CBAM成本效益分析法三种方式。

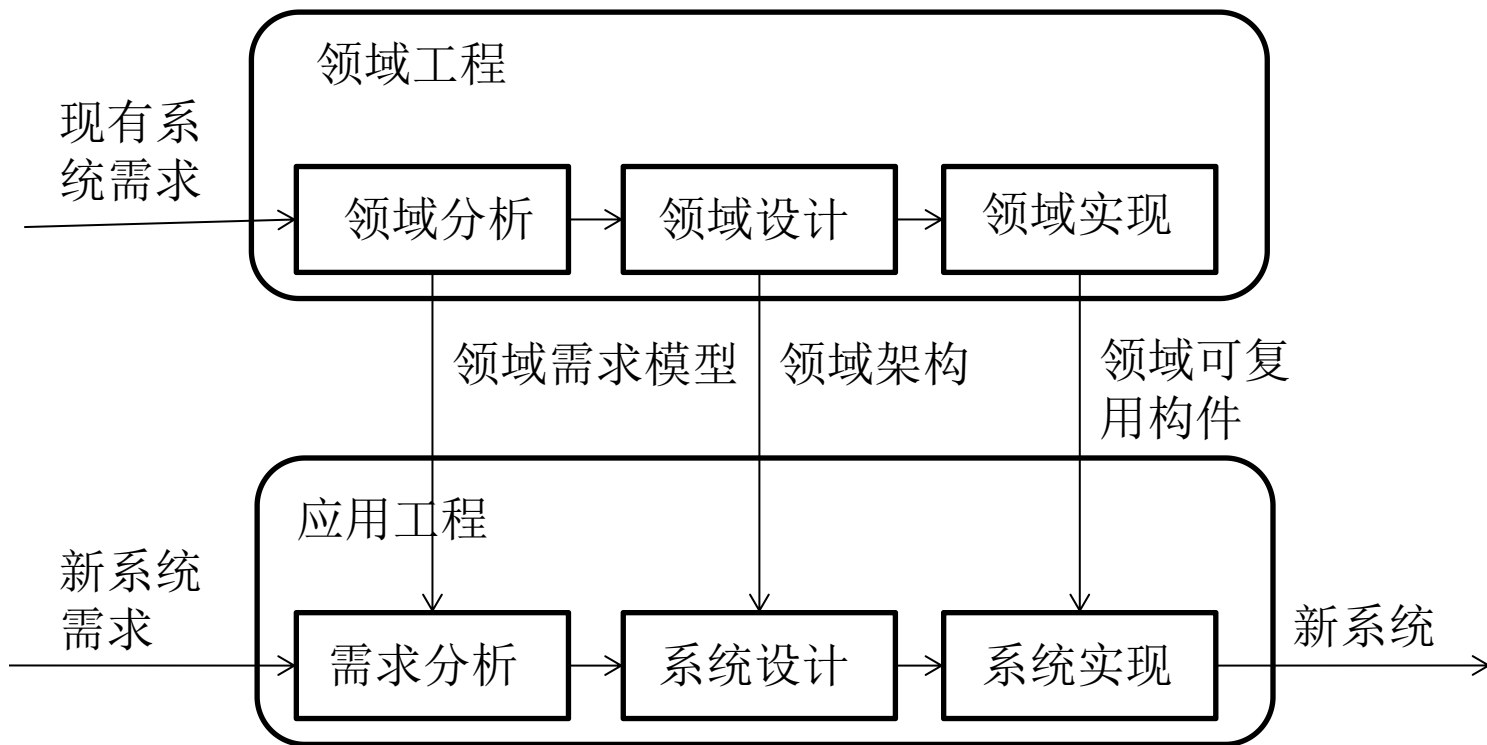
ATAM是在SAAM的基础上发展起来的,SAAM考察的是软件体系结构单独的质量属性,而ATAM则提供从多个竞争的质量属性方面来理解软件体系结构的方法。使用ATAM不仅能看到体系结构对于特定质量目标的满足情况,还能认识到在多个质量目标间权衡的必要性。

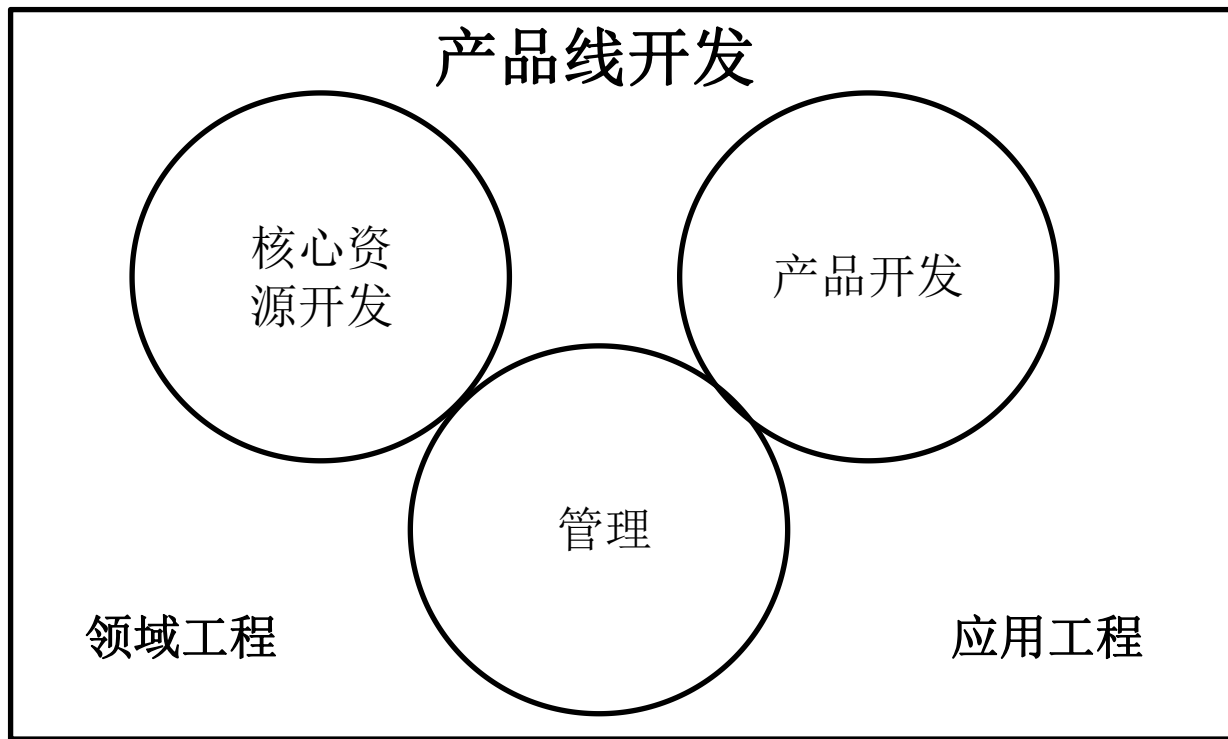
质量属性会为系统的项目干系人带来一些收益。CBAM帮助项目干系人根据其投资收益率来选择架构策略。CBAM可以用做ATAM方法的补充,在ATAM的评估结果上对架构的经济性进行评估。

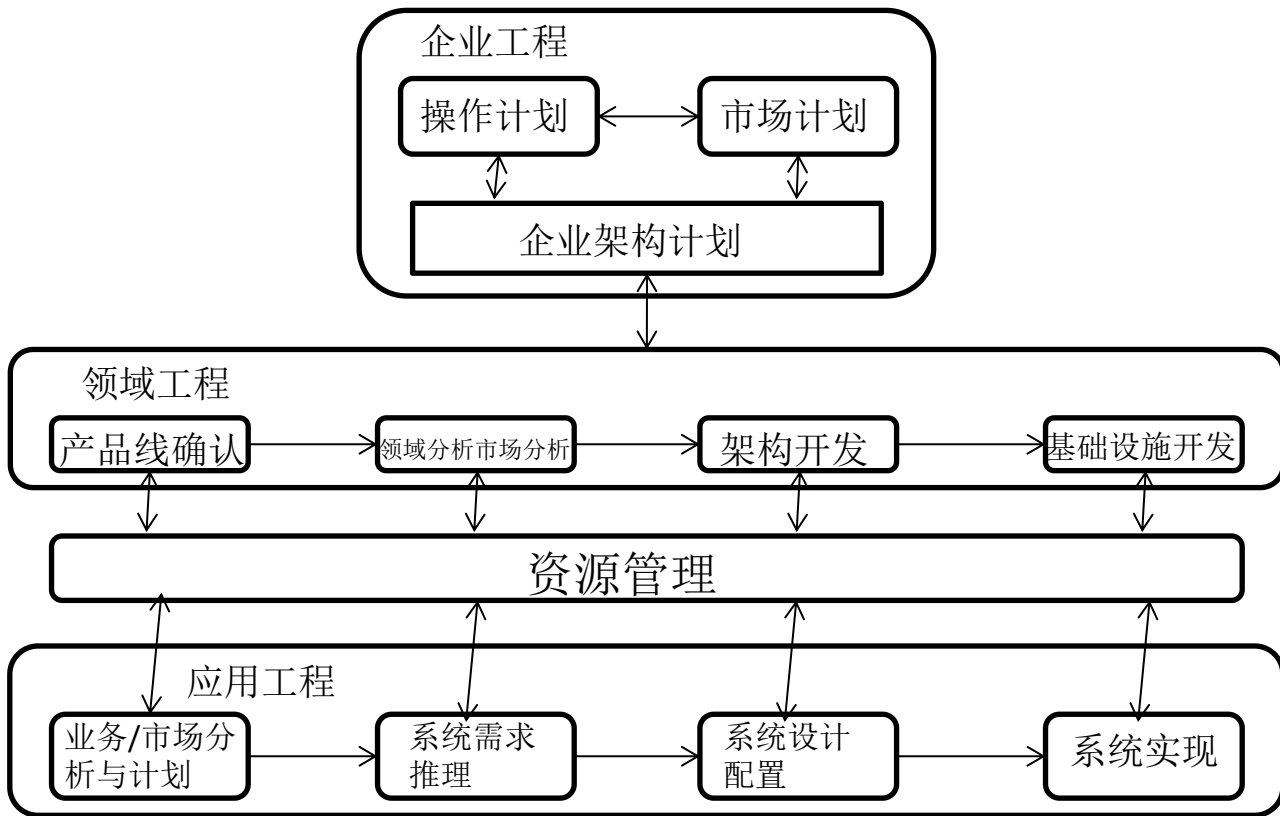


- 整理场景
- 对场景进行求精
- 确定场景的优先级
- 分配效用
- 形成“策略-场景-响应级别”的对应关系
- 确定期望的质量属性响应级别的效用
- 计算各架构策略的总收益
- 根据受成本限制影响的投资报酬率选择架构策略









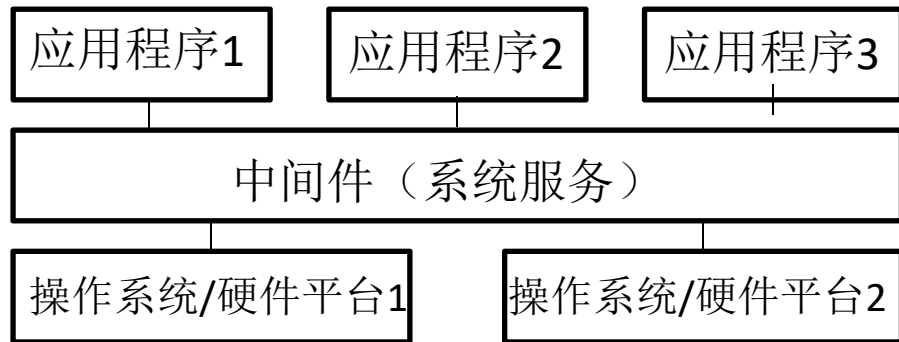
	演化方式	革命方式
基于现有产品	基于现有产品架构设计产品线的架构，经演化现有构件，开发产品线构件	核心资源的开发基于现有产品集的需求和可预测的、将来需求的超集
全新产品线	产品线核心资源随产品新成员的需求而演化	开发满足所有预期产品线成员的需求的核心资源

- 将现有产品演化为产品线
- 用软件产品线替代现有产品集
- 全新软件产品线的演化
- 全新软件产品线的开发

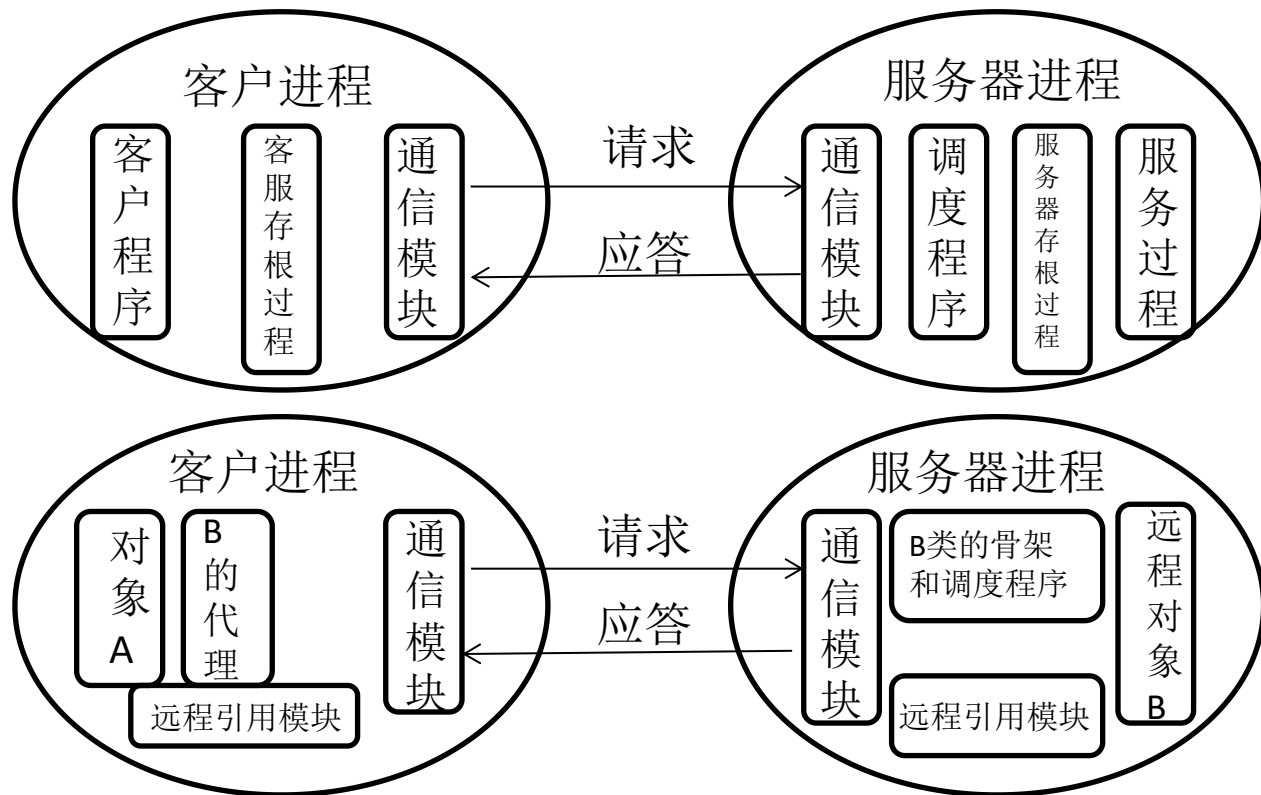
- 设立独立的核心资源小组
- 不设立独立的核心资源小组
- 动态的组织结构

- 对该领域具备长期和深厚的经验
- 一个用于构建产品的好的核心资源库
- 好的产品线架构
- 好的管理（软件资源、人员组织、过程）支持

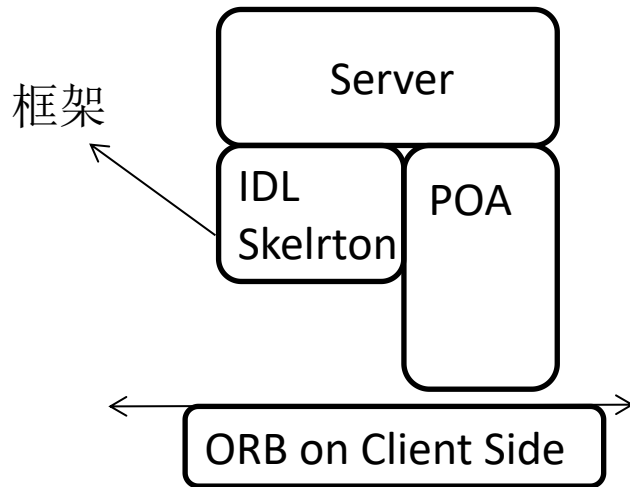
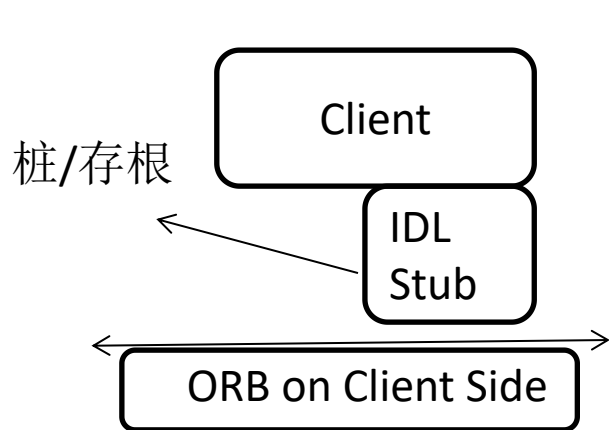
中间件是一种独立的系统软件或服务程序，可以帮助分布式应用软件在不同的技术之间共享资源

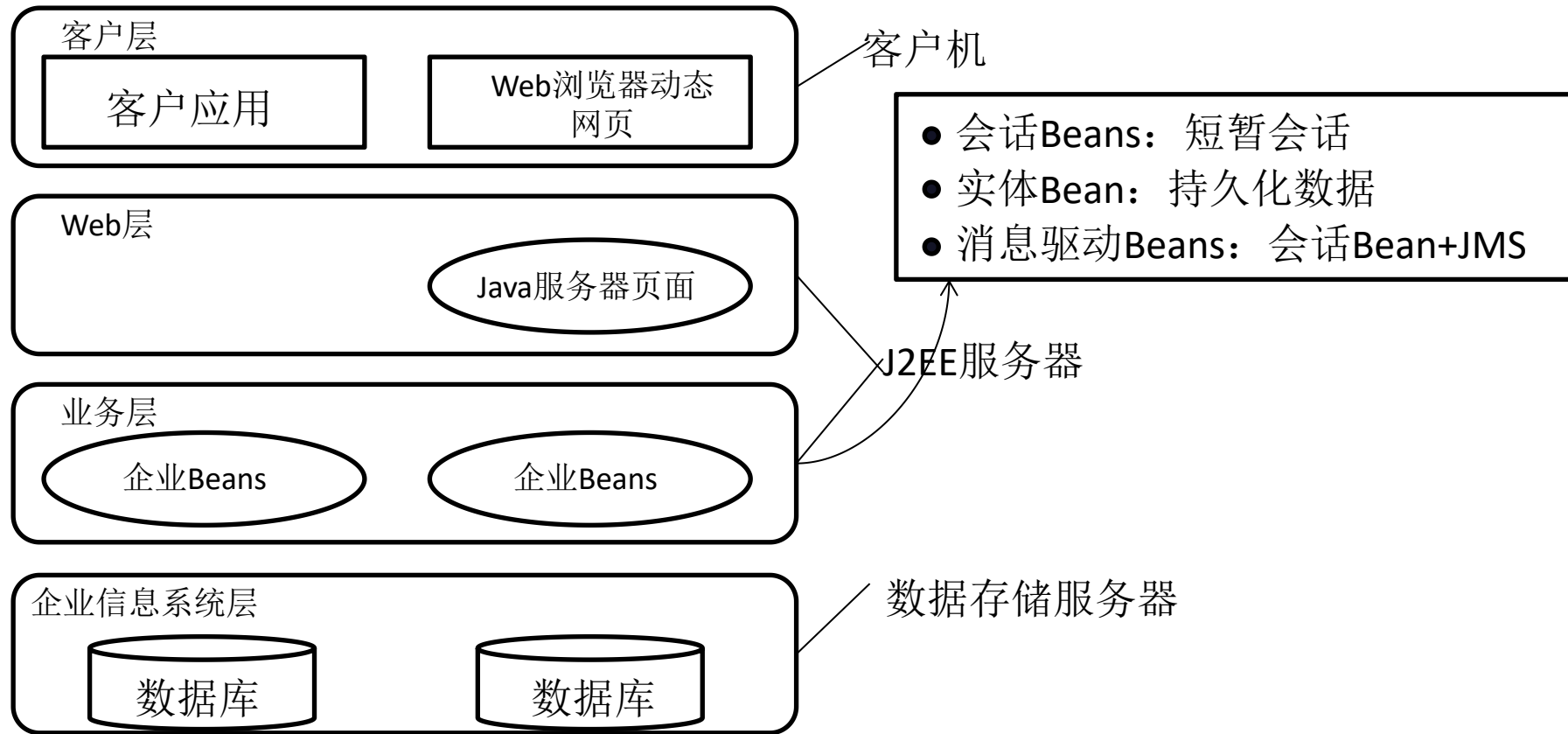


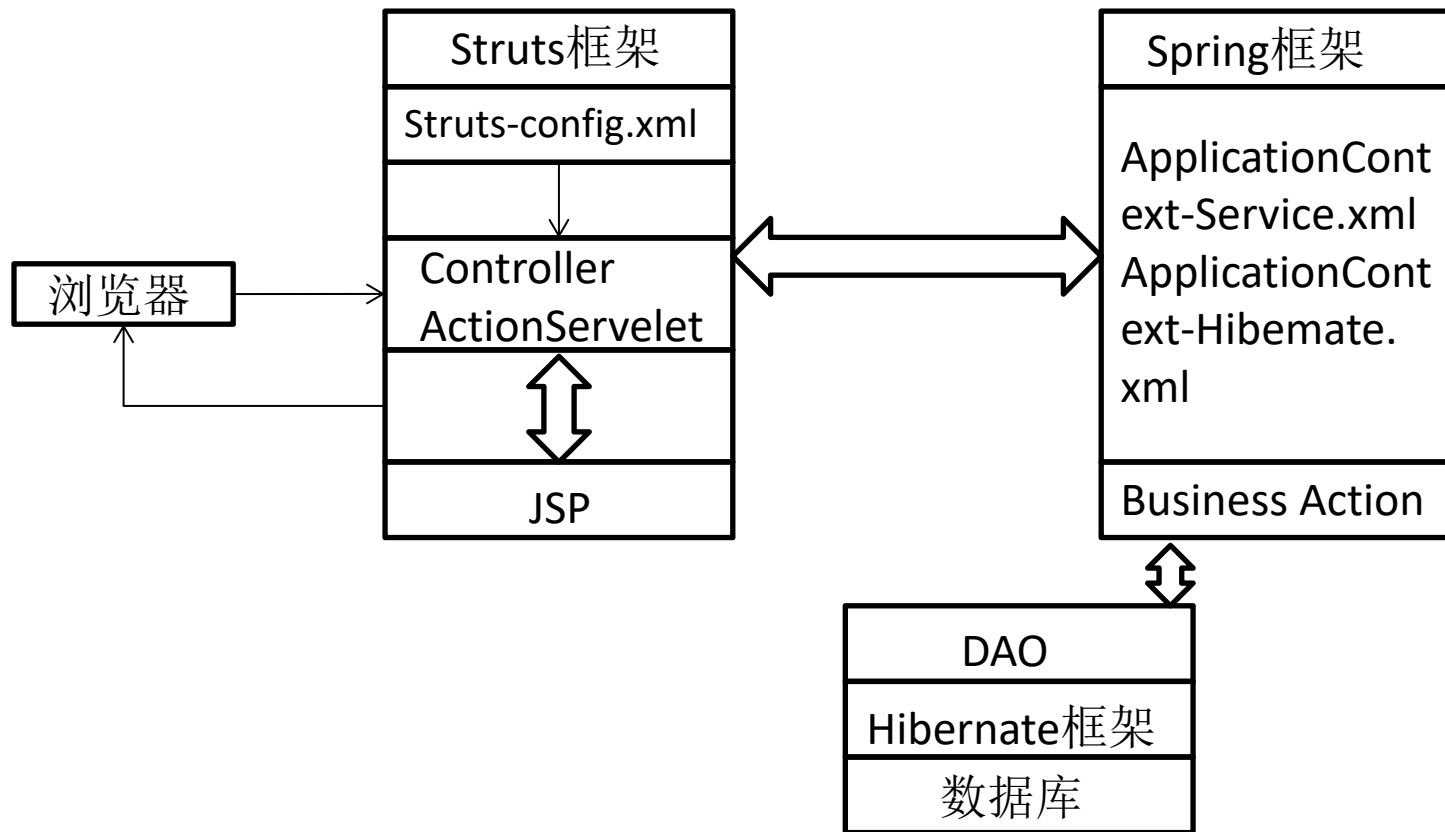
- 负责客户机与服务器之间的连接和通信，以及客户机与应用层之间的高效率通信机制
- 提供应用层不同服务之间的互操作机制，以及应用层与数据库之间的连接和控制机制
- 提供多层构架的应用开发和运行的平台，以及应用开发框架，支持模块化的应用开发
- 屏蔽硬件、操作系统、网络和数据库的差异
- 提供应用的负载均衡和高可用性、安全机制与管理功能，以及交易管理机制，保证交易的一致性
- 提供一组通用的服务去执行不同的功能，避免重复的工作和使应用之间可以协作



- 远程过程调用
- 对象请求代理
- 远程方法调用
- 面向消息的中间件
- 事务处理监控器







Model（模型）是应用程序中用于处理应用程序数据逻辑的部分。通常模型对象负责在数据库中存取数据

View（视图）是应用程序中处理数据显示的部分。通常视图是依据模型数据创建的。

Controller（控制器）是应用程序中处理用户交互的部分。通常控制器负责从视图读取数据，控制用户输入，并向模型发送数据

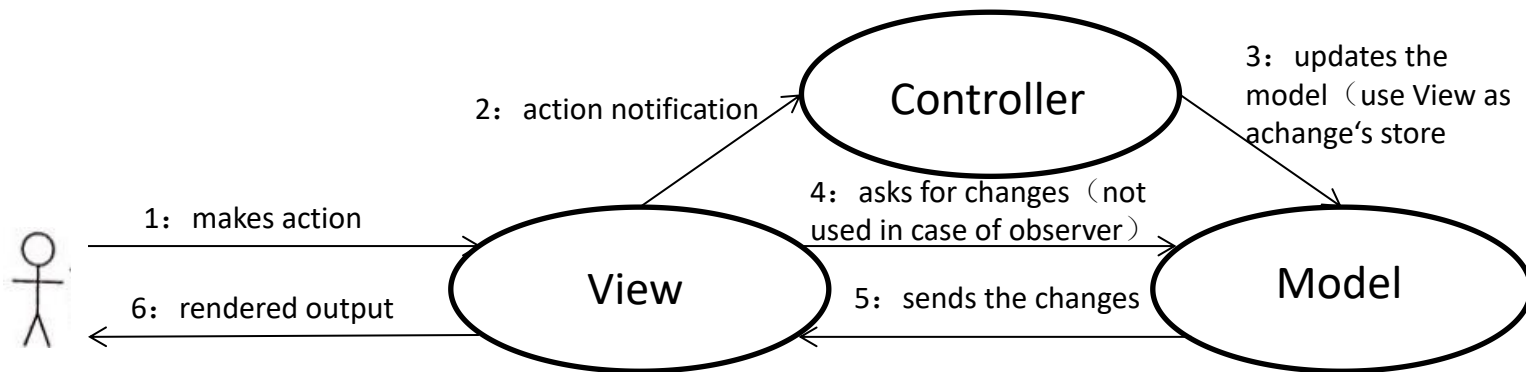
J2EE体系结构中：

视图（**View**）：JSP

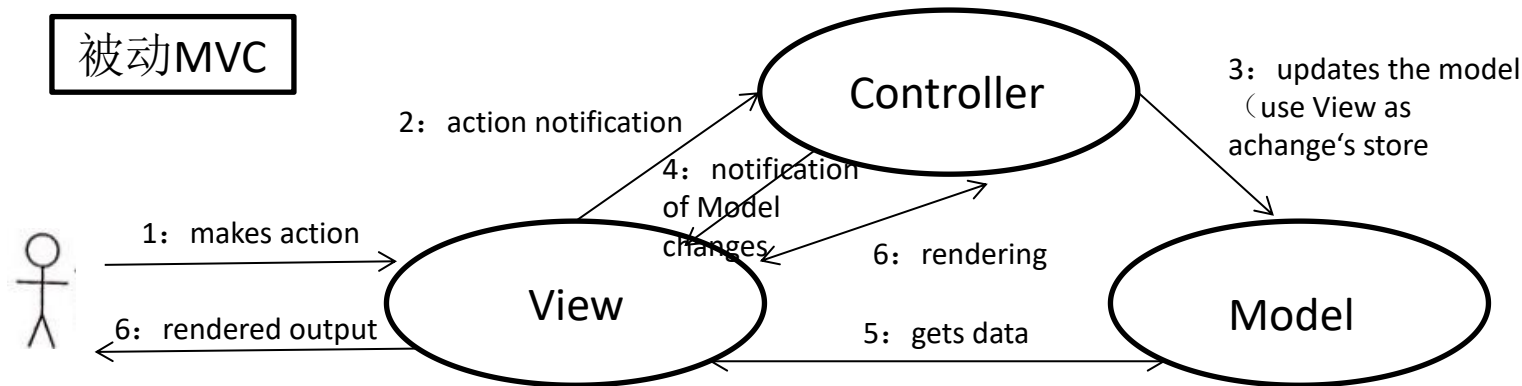
控制（**Controller**）：Servlet

模型（**Model**）：Entity Bean、Session Bean

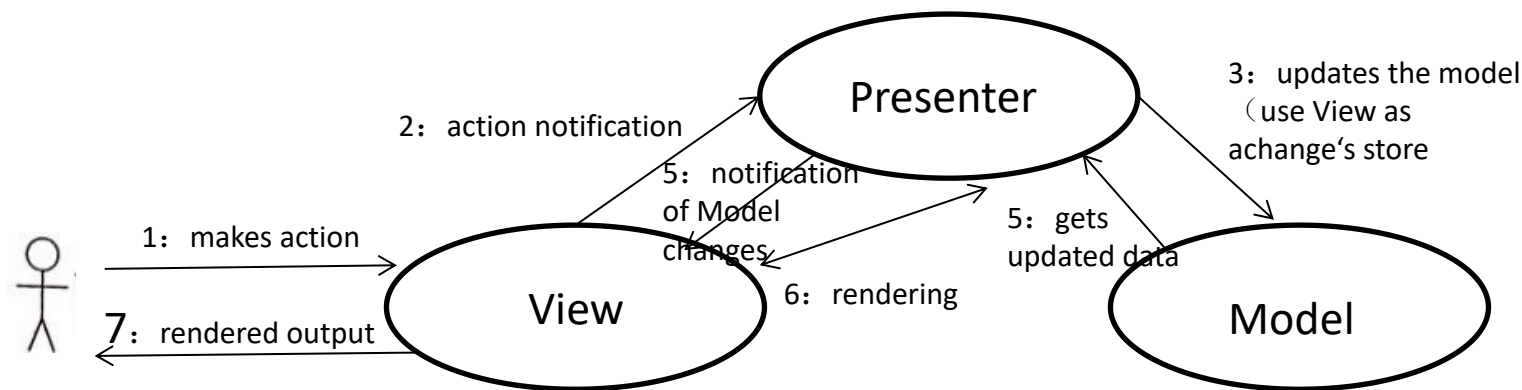
主动MVC

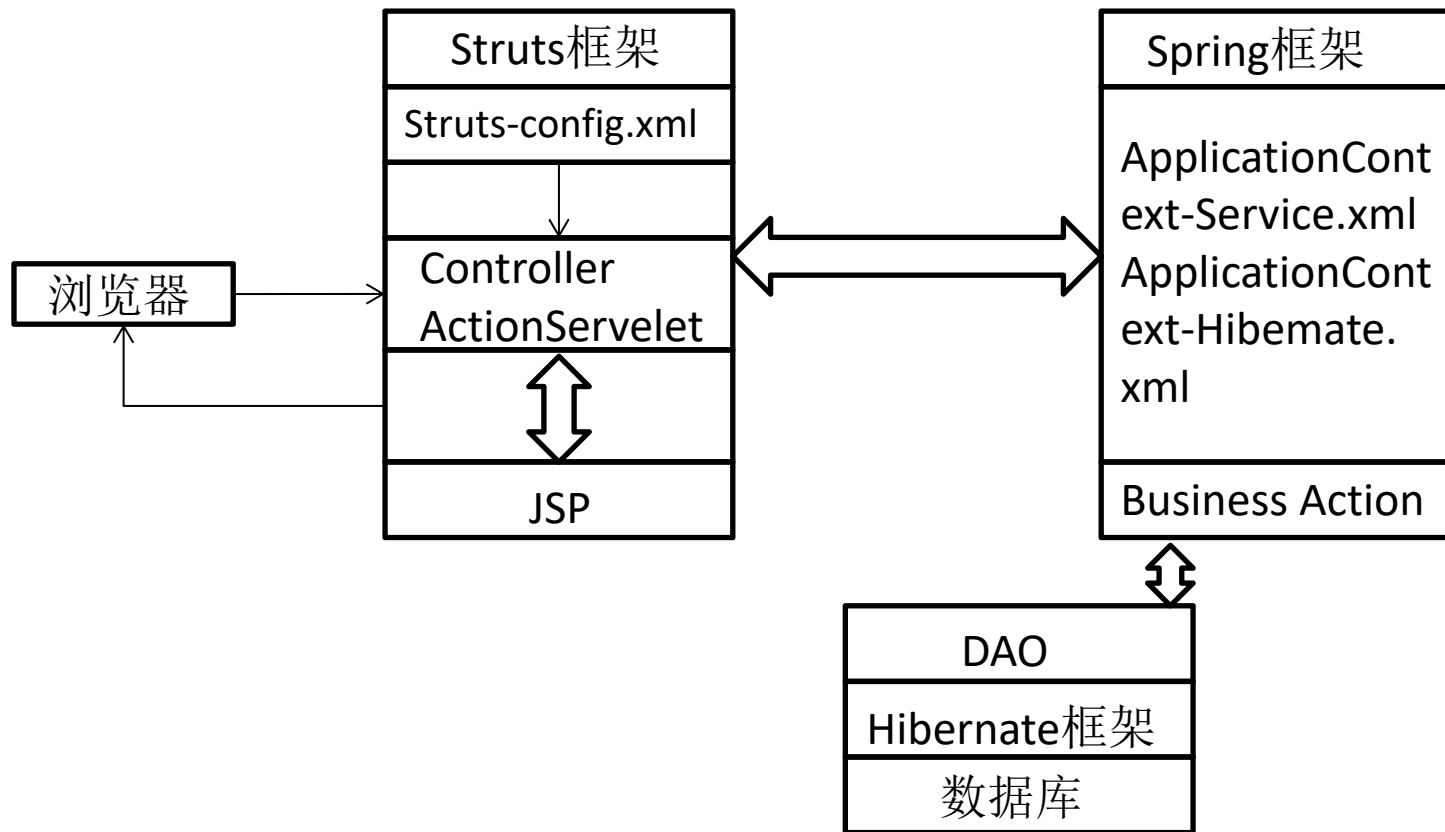


被动MVC



- MVP是MVC的变种
- MVP实现了V与M之间的解耦（V不直接使用M，修改V不会影响M）
- MVP更好的支持单元测试（业务逻辑在P中，可以脱离V来测试这些逻辑；可以将一个P用于多个V，而不需要改变P的逻辑
- MVP中V要处理界面事件，业务逻辑在P中，MVC中界面事件由C处理

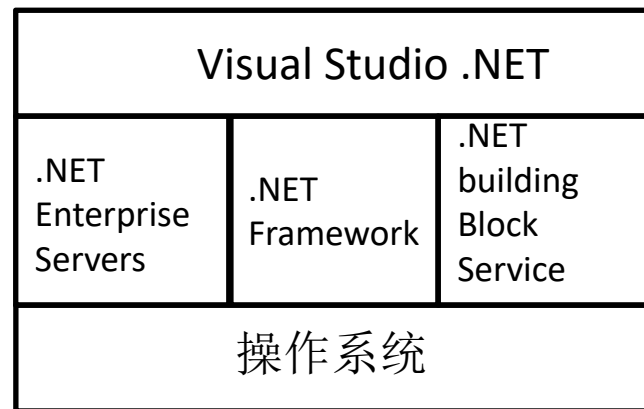
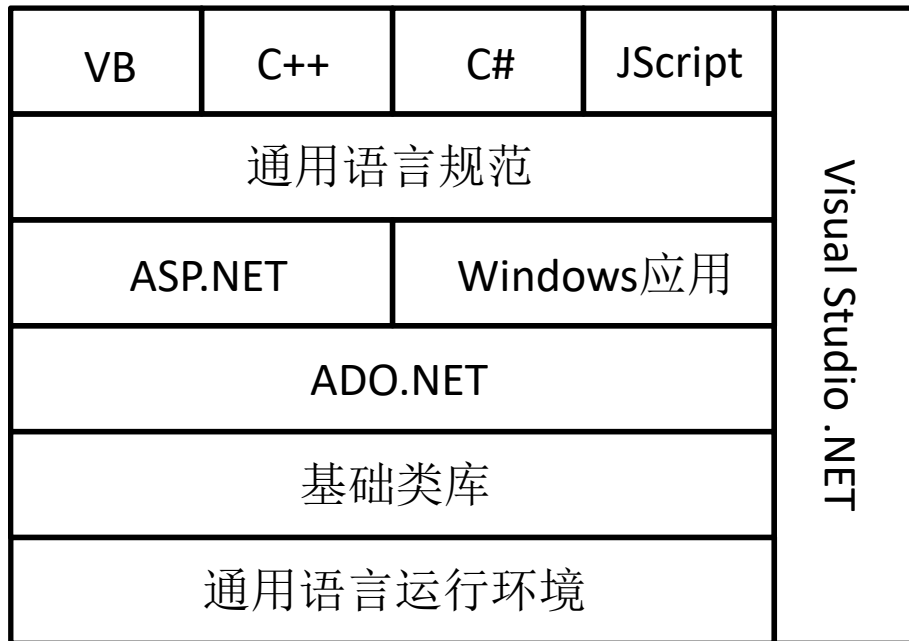




Struts是一个基于J2EE平台的MVC框架，主要采用Servlet和JSP技术来实现。在Struts中，M由实现业务逻辑的JavaBean构成，C由ActionServlet和Action来实现，V由一组JSP文件构成

Spring通过RMI或Web Service远程访问业务逻辑，允许自由选择和组装各部分功能，还提供和其他软件集成的接口。**Spring**本身是个容器，管理构件的生命周期、构件的组态。依赖注入等，并可以控制构件在创建时以原型或单例模式来创建

Hibernate是一个对象关系映射框架，提供了Java对象到数据库表之间的直接映射，它对JDBC进行了非常轻量级的对象封装，使得Java程序员可以使用对象编程思维来操作数据库。在Hibernate中，ORM机制的核心是一个XML文件，该文件描述了数据库模式是怎么与一组Java类绑定在一起的



WINDOWS本身的安全漏洞，使得.NET的安全性不如J2EE。同时，在应用服务器的选择上，.NET只能用IIS，安全性、稳定性难以保证；而J2EE有更多的选择，可以在诸多遵循标准的厂商所提供的应用程序服务器中，选择最符合需要、成本最低、而且又被认为是最佳的平台。

J2EE最大的优势是跨平台，
.NET最大的优势是入门容易和性能较高。

- JVM与CLR
- 对多层分布式应用的支持
- 安全性
- 应用程序的部署
- 可移植性
- 外部支持

J2EE作为一种开放的规范，从一开始就得到了众多厂商的支持，IBM、BEA、HP、Oracle等在J2EE的实施上都有较大的投入。

二者都支持多层分布式应用的标准架构。

在移植性方面，.NET支持跨语言，J2EE支持跨平台。

Visual Stdio.NET是其唯一的开发工具。



基于 JavaEE 平台的基础功能服务构建应用系统时，(41) 可用来集成遗产系统。

- A. JDBC、JCA 和 Java IDL
- B. JDBC、JCA 和 JMS
- C. JDBC、JMS 和 Java IDL
- D. JCA、JMS 和 Java IDL

【答案】D

【解析】

JDBC (Java Data Base Connectivity, java 数据库连接) 是一种用于执行 SQL 语句的 Java API, 可以为多种关系数据库提供统一访问, 它由一组用 Java 语言编写的类和接口组成。JDBC 提供了一种基准, 据此可以构建更高级的工具和接口, 使数据库开发人员能够编写数据库应用程序。

JCA 标准化连接子是由 J2EE 1.3 首先提出的, 它位于 J2EE 应用服务器和企业信息系统 (EIS) 之间, 比如数据库管理、企业资源规划 (ERP)、企业资产管理 (EAM) 和客户关系管理 (CRM) 系统。不是用 Java 开发的企业应用或者在 J2EE 框架内的应用都可以通过 JCA 连接。JCA 是在 javax.resource 包和它的子包 (cci, spi 和 spi.security) 中定义的。

(JCA 的缩写形式也用于 Java cryptography API 的缩写。)

JMS 是 Java 对消息系统的访问机制, 但它本身并不实现消息。JMS 支持点对点分发的消息队列, 也支持多个目标订阅的消息主题。当消息发布给一个主题的适合, 消息就会发送给所有那个主题的订阅者。JMS 支持各种消息类型 (二进制、流、名-值表、序列化的对象和文本)。通过声明与 SQL 的 WHERE 相近的句段, 可以建立消息的过滤器。

Java IDL 即 idltojava 编译器就是一个 ORB, 用来在 Java 语言中定义、实现和访问

CORBA 对象。Java IDL 支持的是一个瞬间的 CORBA 对象, 即在对象服务器处理过程中有效。实际上, Java IDL 的 ORB 是一个类库而已, 并不是一个完整的平台软件, 但它对 Java IDL

应用系统和其他 CORBA 应用系统之间提供了很好的底层通信支持, 实现了 OMG 定义的 ORB 基本功能。



DESIGNER:王川林
软件架构设计



THANK YOU