

如何实现高并发高可用的选购系统：唯品会选购系统架构设计实践

张广平



促进软件开发领域知识与创新的传播



关注InfoQ官方微信
及时获取ArchSummit
大会演讲视频信息



全球软件开发大会 [北京站]

2017年4月16-18日 北京·国家会议中心

咨询热线: 010-64738142



全球架构师峰会 2016 [深圳站]

2017年7月7-8日 深圳·华侨城洲际酒店

咨询热线: 010-89880682



资深架构专家

2014年加盟唯品会，作为唯品会电商平台应用架构负责人，负责唯品会电商平台应用架构管理工作，主持公司架构评审运作；主持多个公司战略级项目的架构设计和支持工作；唯品会核心系统重构总架构师。



开发经理&Tech Leader

2014年前在eBay工作接近10年，负责eBay商城支付平台开发管理工作，PaaS平台架构师，电商平台系统Tech Leader



大纲

- 一．关于唯品会
- 二．唯品会系统历史问题和技术演变
- 三．唯品会基础架构体系
- 四．选购业务系统架构设计思路

关于唯品会

唯品会

一家专门做特卖的网站



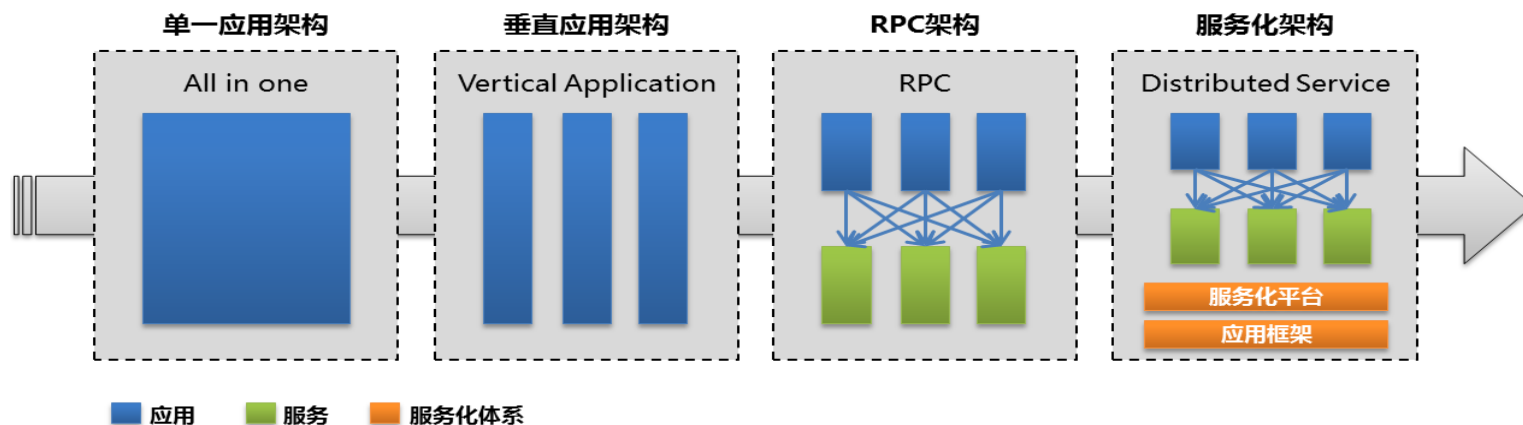
精选品牌

深度折扣

限时限量

唯品会应用业务系统历史问题和技术演变

- **业务系统**缺少长期规划，**扩展困难**
- 系统没有使用**统一框架**，效率低下
- 业务**边界不清**、责任不明确、耦合严重
- **数据冗余**严重、数据模型定义不严谨
- 系统**可靠性低**、治理能力弱、运维成本高
- **服务颗粒度太粗**，复用性差
- **容量扩展**问题



唯品会基础框架体系

唯品会基础框架



唯品会基础框架 - 服务化框架OSP

高性能

- 基于Thrift与Netty
- 单机数万QPS
- 高效二进制协议
- TCP长连接，异步传输，流式处理
- 海量并发连接

分布式与扩展性

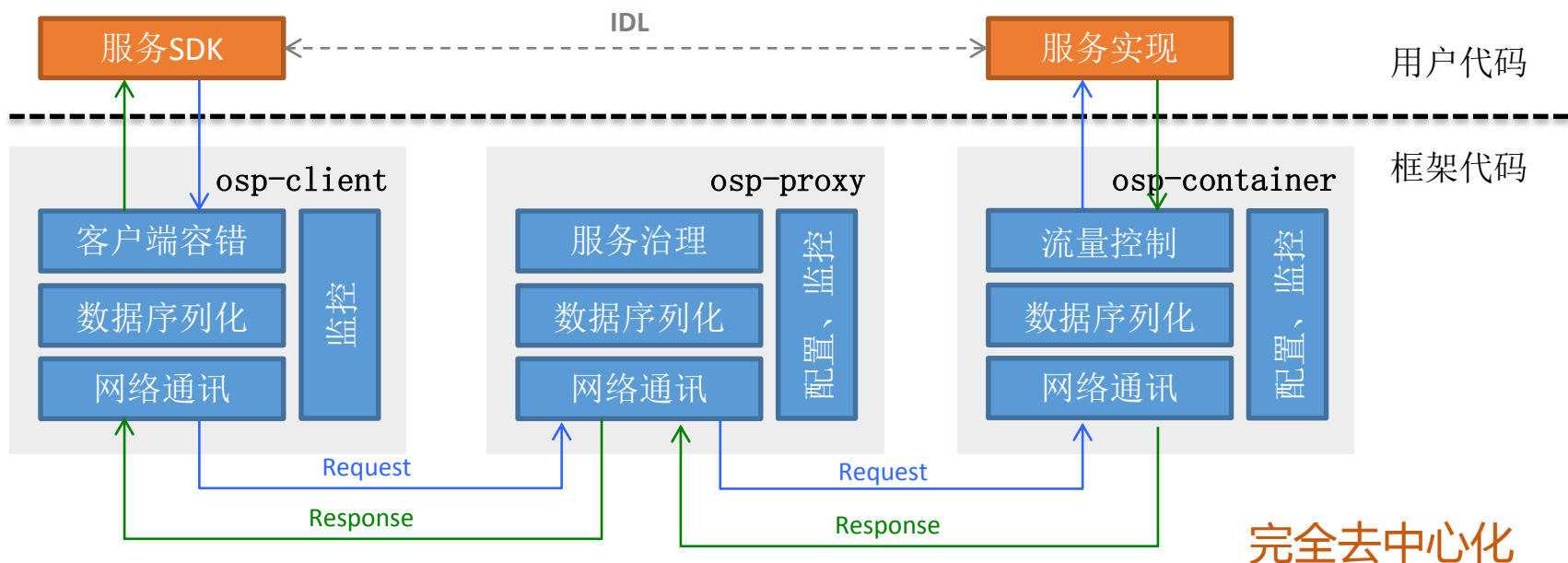
- 协议无状态
- 服务的自动注册与发现
- 不依赖中心化的Load Balancer

智能服务路由

- 服务上线/下线
- 机房选择
- 灰度发布
- 读写分离、优雅降级

容错机制

- 超时控制
- 容错重试、熔断隔离
- 服务限流、优雅停止



Java: 长链接, PHP: 短链接

长链接

唯品会日志框架Mercury

海量日志消息的记录，告警与分析



IT运维 / 监控中心人员

- 快速故障告警和问题定位
- 把握应用性能和容量评估
- 提供可追溯的性能数据



应用开发人员

- 定位线上服务性能瓶颈
- 持续优化代码和SQL
- 帮助快速解决线上问题



应用管理人员

- 全方位把握应用整体拓扑结构
- 定位全网应用瓶颈
- 帮助优化关键业务



唯品会选购业务系统设计

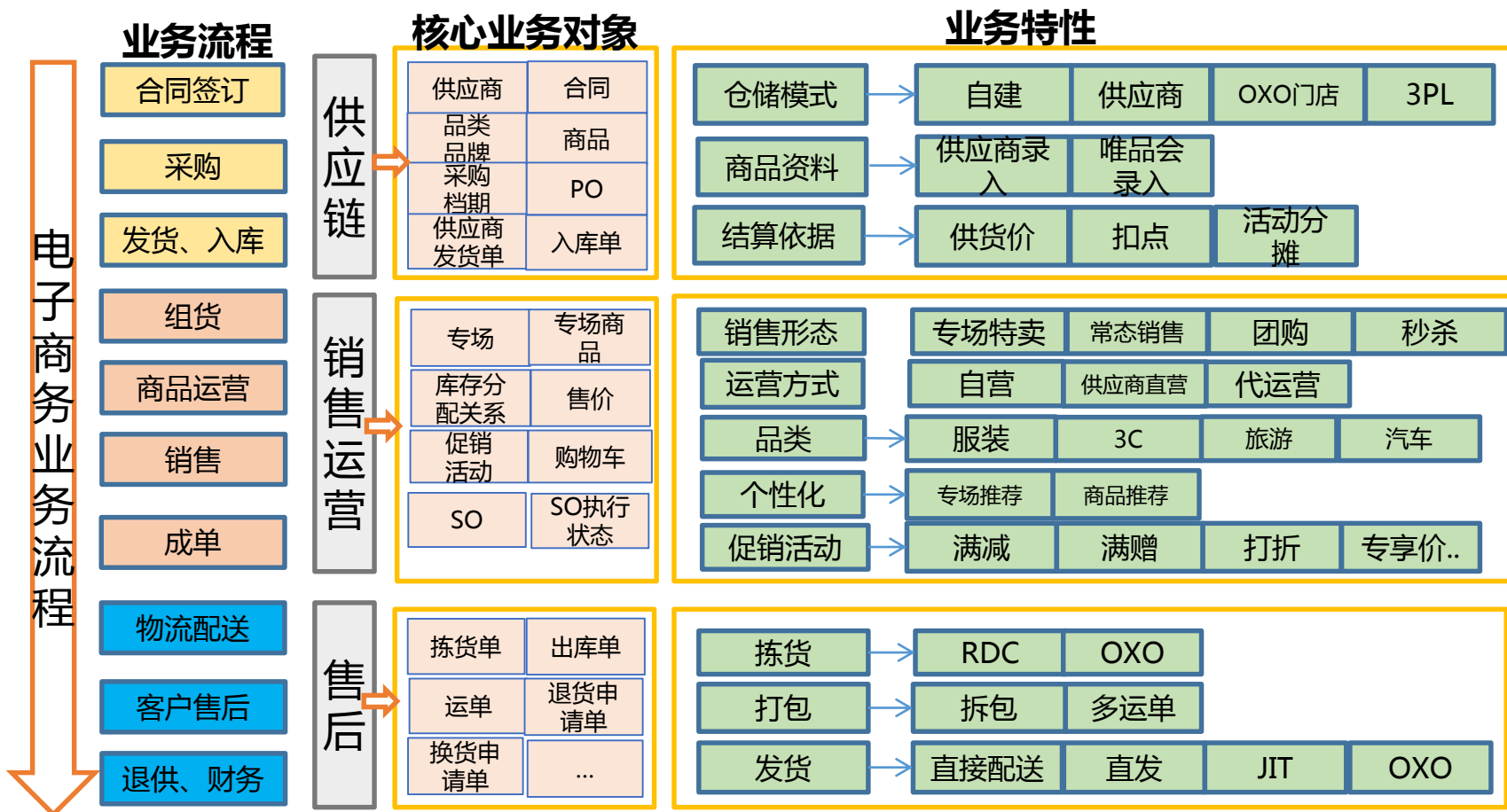
选购线业务介绍

- 选购业务流程以及主要用例

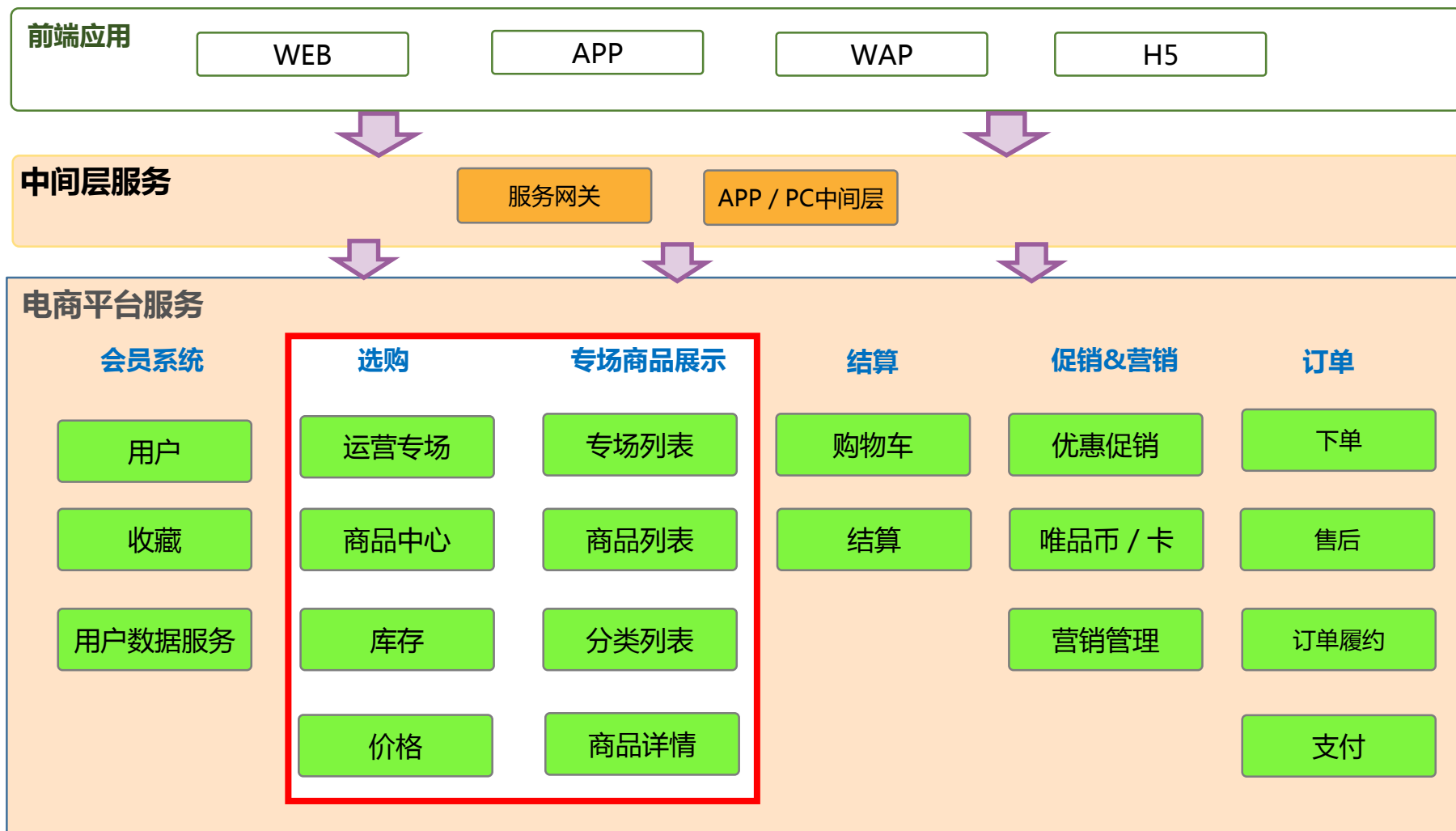
选购系统设计

- 系统扩展模型
- 整体设计思路
- 业务系统划分
- 系统服务化
- 系统异步交互改造
- 系统不一致性改造
- 优化数据访问，改善性能
- 移动中间层服务改造

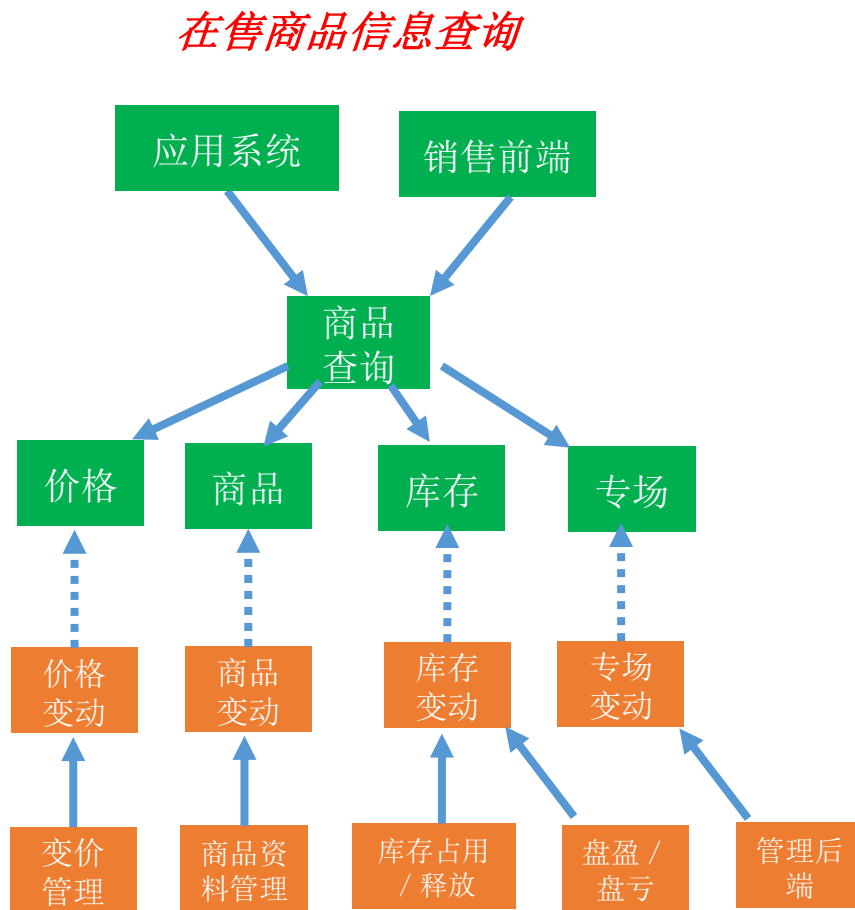
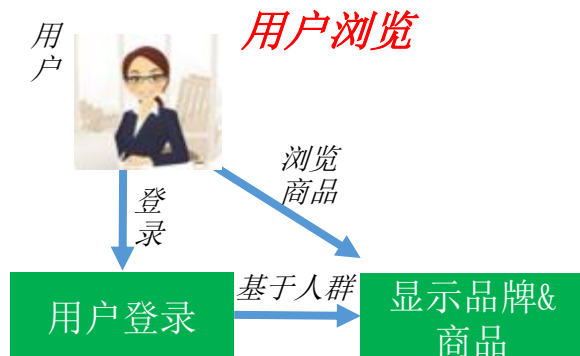
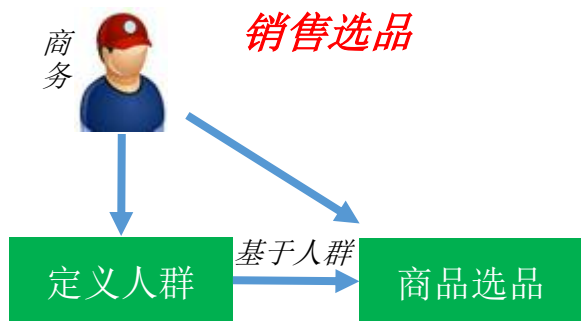
唯品会业务流程图



唯品会电商系统逻辑架构图



选购业务主要用例

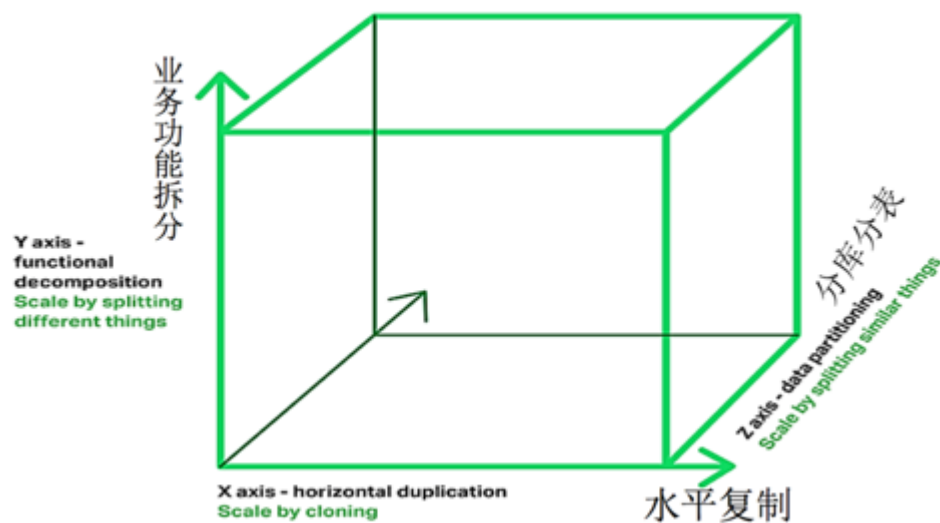


系统如何设计？

系统设计原则

- ✓ 业务驱动
- ✓ 高可用性
- ✓ 高可扩展性和伸缩性
- ✓ 低成本
- ✓ 高性能
- ✓ 安全性

系统扩展模型



整体架构设计思路

- 系统扩展性

- ✓ 业务逻辑拆分，合理系统建模和逻辑拆分
- ✓ 系统分层设计，增加系统可伸缩性
- ✓ 服务化解耦，增加系统重用性
- ✓ 系统间通讯增加异步处理，减少同步处理，改造时间敏感的定时任务
- ✓ 优化移动APP接入，引入服务网关

- 系统可用性

- ✓ 减少数据冗余，降低不一致性
- ✓ 多机房部署容灾

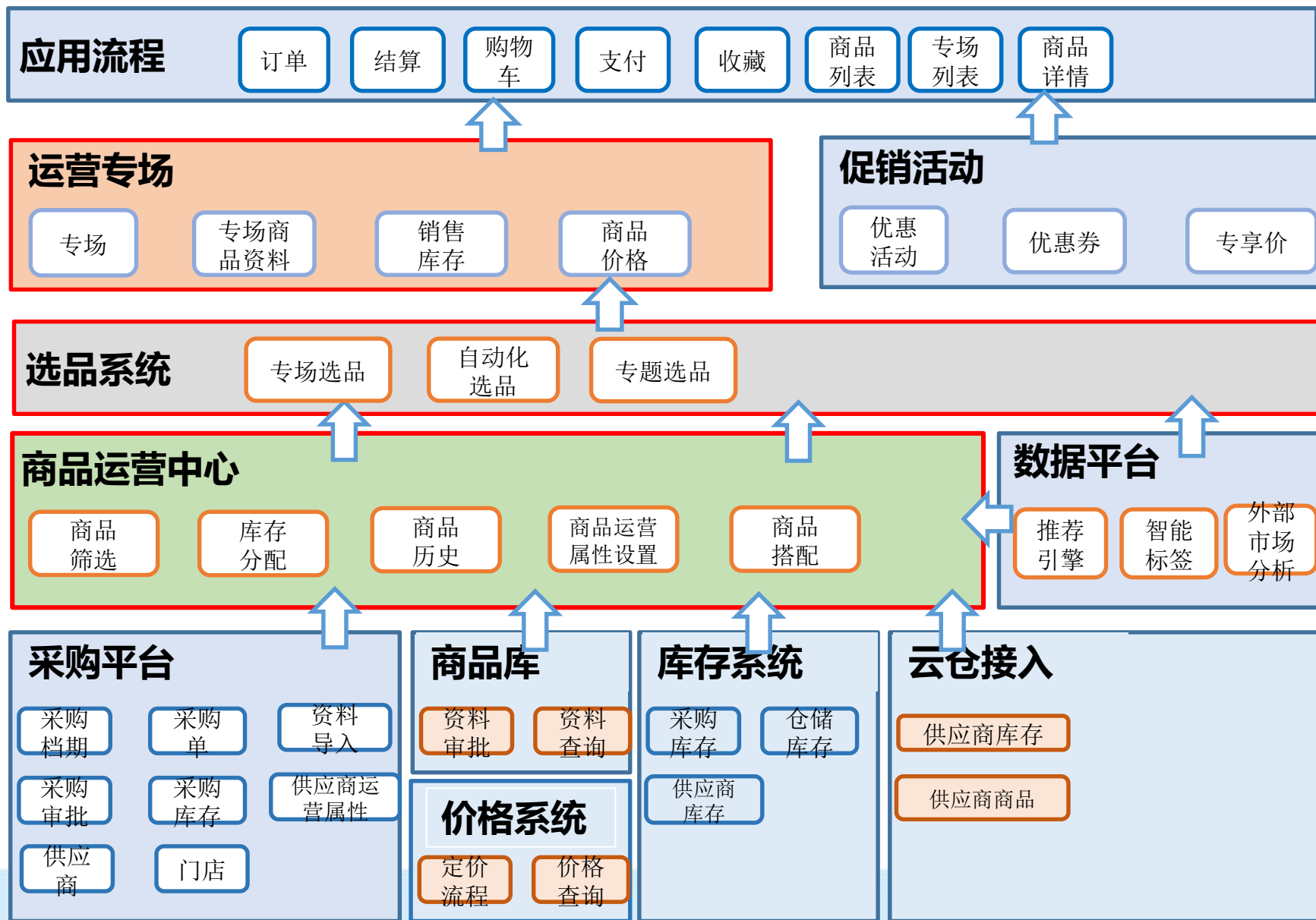
- 系统性能

- ✓ 优化数据访问，合理使用缓存

- 系统标准化

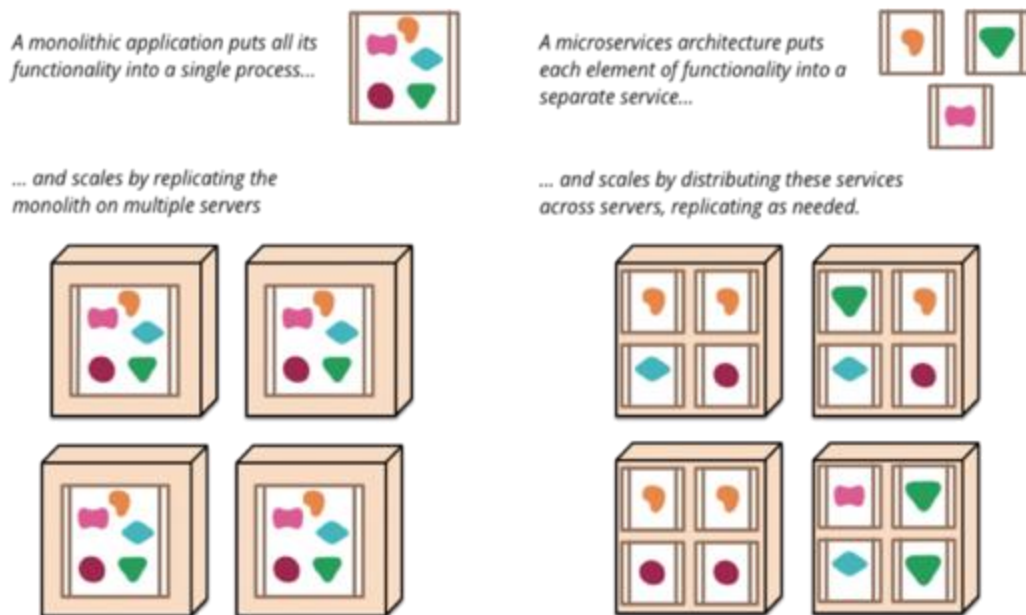
- ✓ 统一数据共享标准，明确各数据共享场景及规范
- ✓ 统一框架治理

选购线业务系统划分



系统服务化解耦

- ◆ 基于统一服务化框架OSP，改写业务系统
- ◆ 引入服务分层，基础服务、聚合服务、流程服务、中间层服务，规划服务分工，改善服务重用性和扩展性
- ◆ 优化服务调用链，减少依赖，提高服务可用性



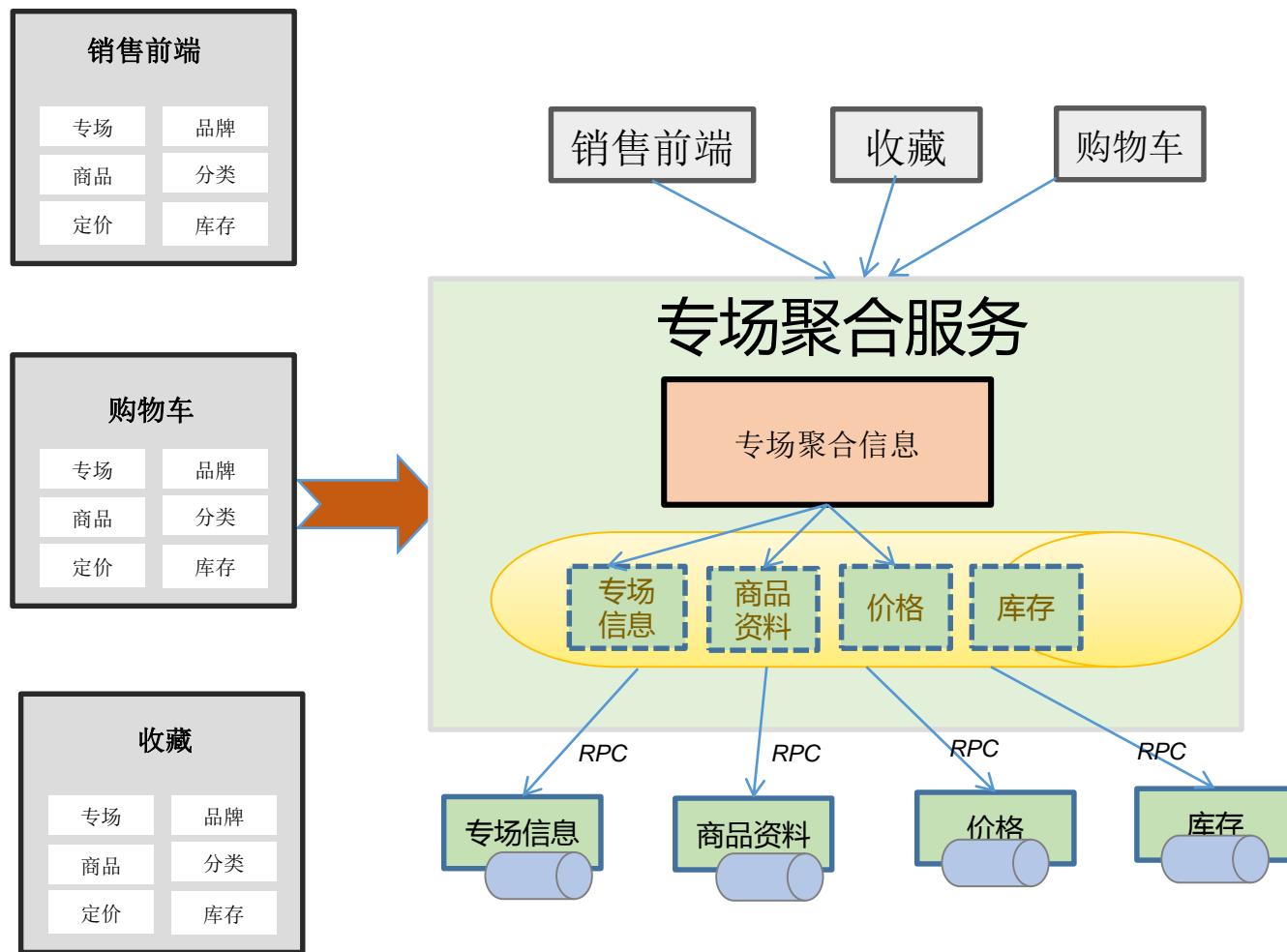
系统服务化原则



服务类型：

服务类型	注意事项	实例
中间层服务	<ul style="list-style-type: none">✓ 中间层调用对应业务流程服务✓ 中间层不负责具体业务逻辑	移动中间层、不同业务系统之间dispatch服务
业务流程服务	<ul style="list-style-type: none">✓ 避免直接访问数据库✓ 流程服务不能调用其它流程服务✓ 粗粒度服务	生成订单、取消订单、购物车操作等。
聚合服务	需要情况下创建聚合服务	专场查询聚合服务、价格聚合服务
业务基础服务	<ul style="list-style-type: none">✓ 细粒度服务，可重用，灵活组合新的流程服务	比如商品查询 / 发布、订单查询、品牌查询、品类查询等

服务化用例：专场商品信息查询服务优化



优点：

1、通过统一的专场聚合服务提供给各个应用系统，统一处理商品信息查询，简化各个应用系统逻辑

2、缓存更加集中处理，减少了RPC的调用次数，节省资源，大幅降低底层基础服务的压力。

3、各种性能加速机制在这个架构上得到了充分的发挥。如：异步并发读取缓存，异步并发调用底层服务，缓存预热机制。

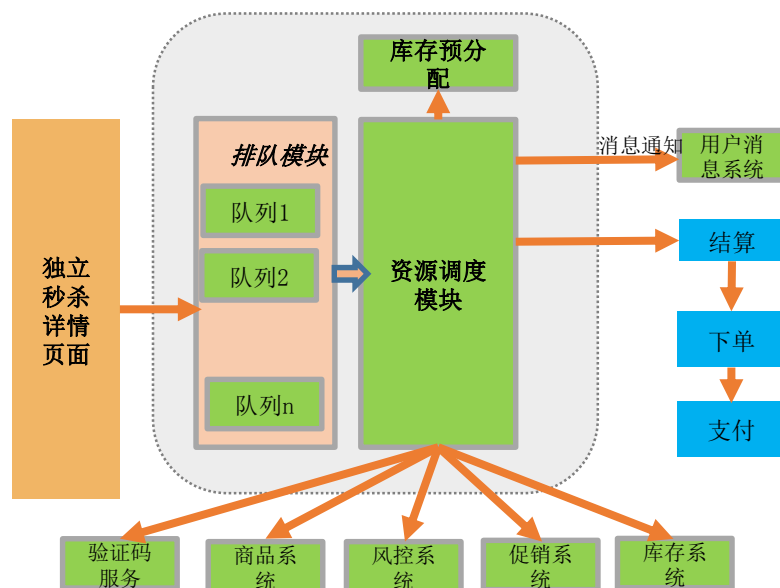
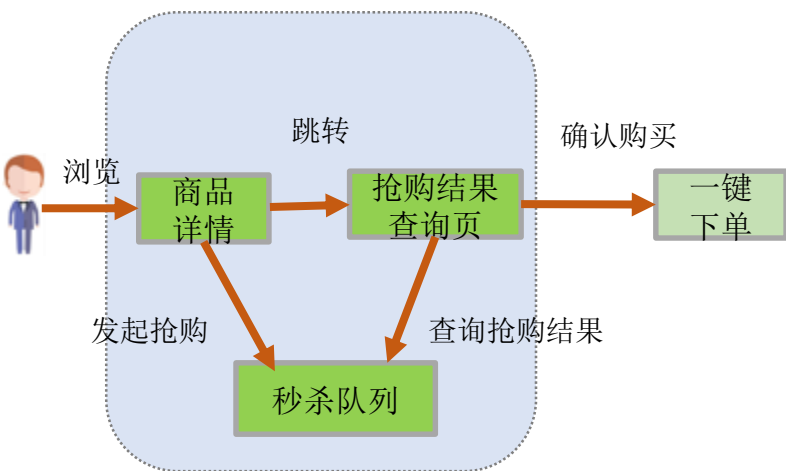
专场系统重构前后服务划分对比



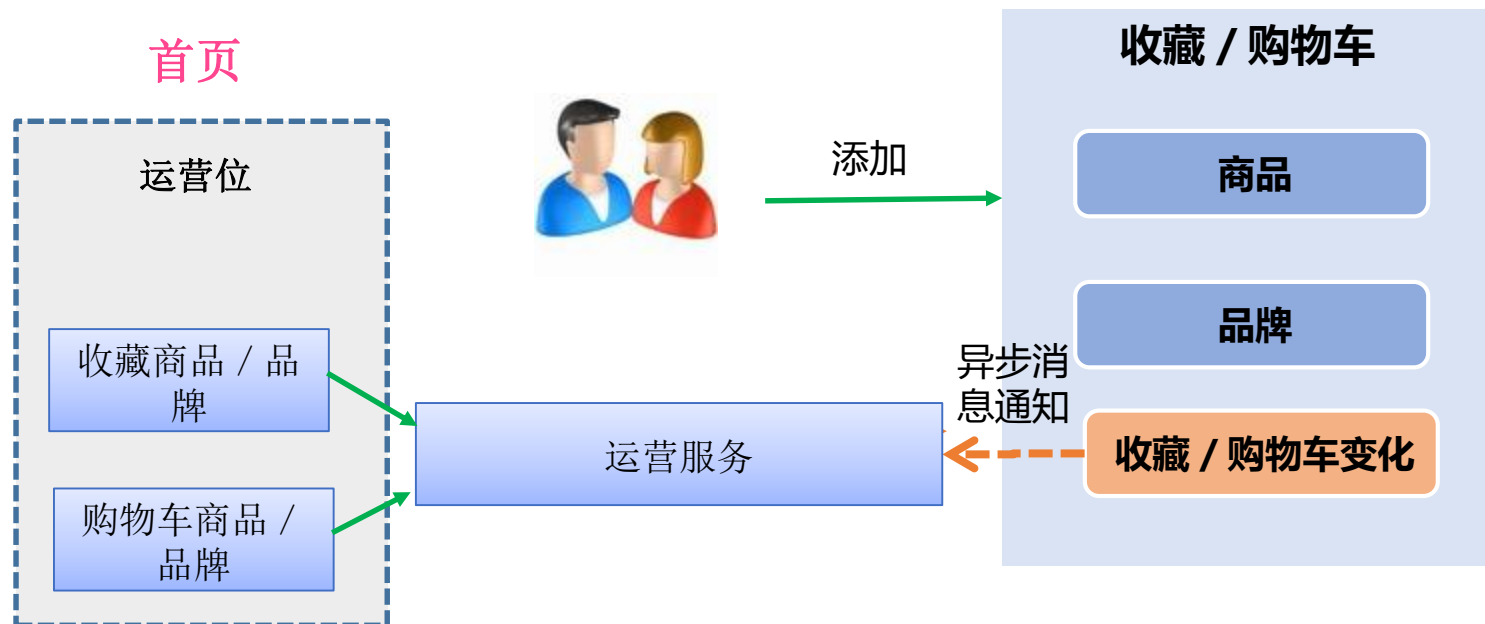
系统间增加异步处理，减少同步处理

- ◆ 通过统一异步消息框架改造异步消息处理、统一治理
- ◆ 重定义业务模块，主业务流程和非主要流程分开，非实时或者强关联业务通过异步调用，实现系统间解耦
- ◆ 数据一致性通过定时补偿实现

异步化用例1：秒杀业务流程异步化处理



系统间异步处理用例 - - - 首页显示个人收藏商品



首页上增加用户个性化感兴趣商品，增加收藏商品服务负担，怎么办？

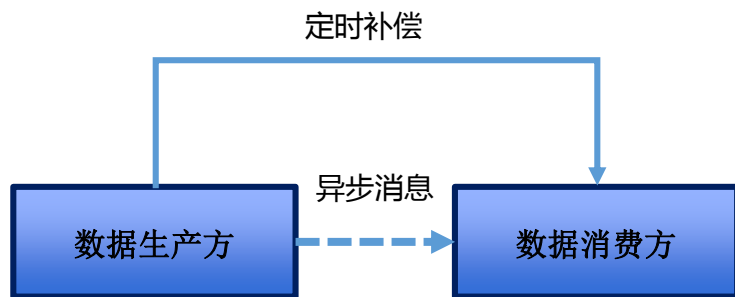
- ◆ 通过异步消息通知方式，同步数据到运营服务模块
- ◆ 运营服务收集用户感兴趣商品信息，减少首页对收藏服务的访问

系统间不一致性改造

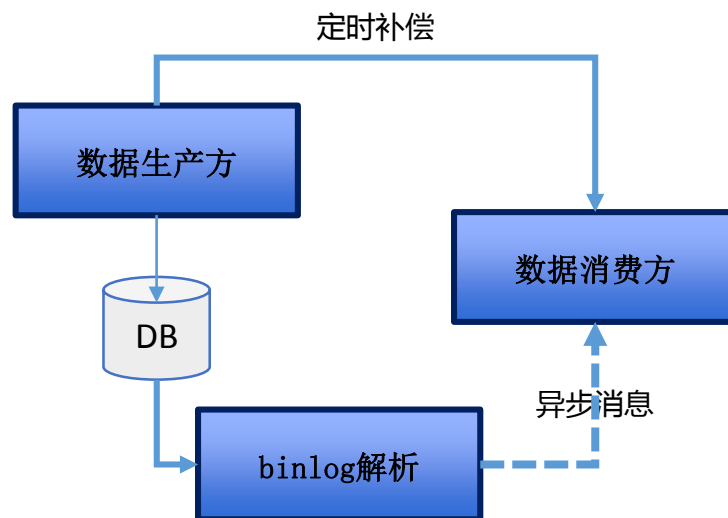
系统拆分后，如何保证分布式事务？

- ◆ 适度业务拆分，减少关键数据冗余
- ◆ 独立系统间共用数据，各个系统引用唯一对象标识
- ◆ 最终一致性，异步消息通知，定时补偿
- ◆ 基于数据库binlog的自研数据推送产品VDP

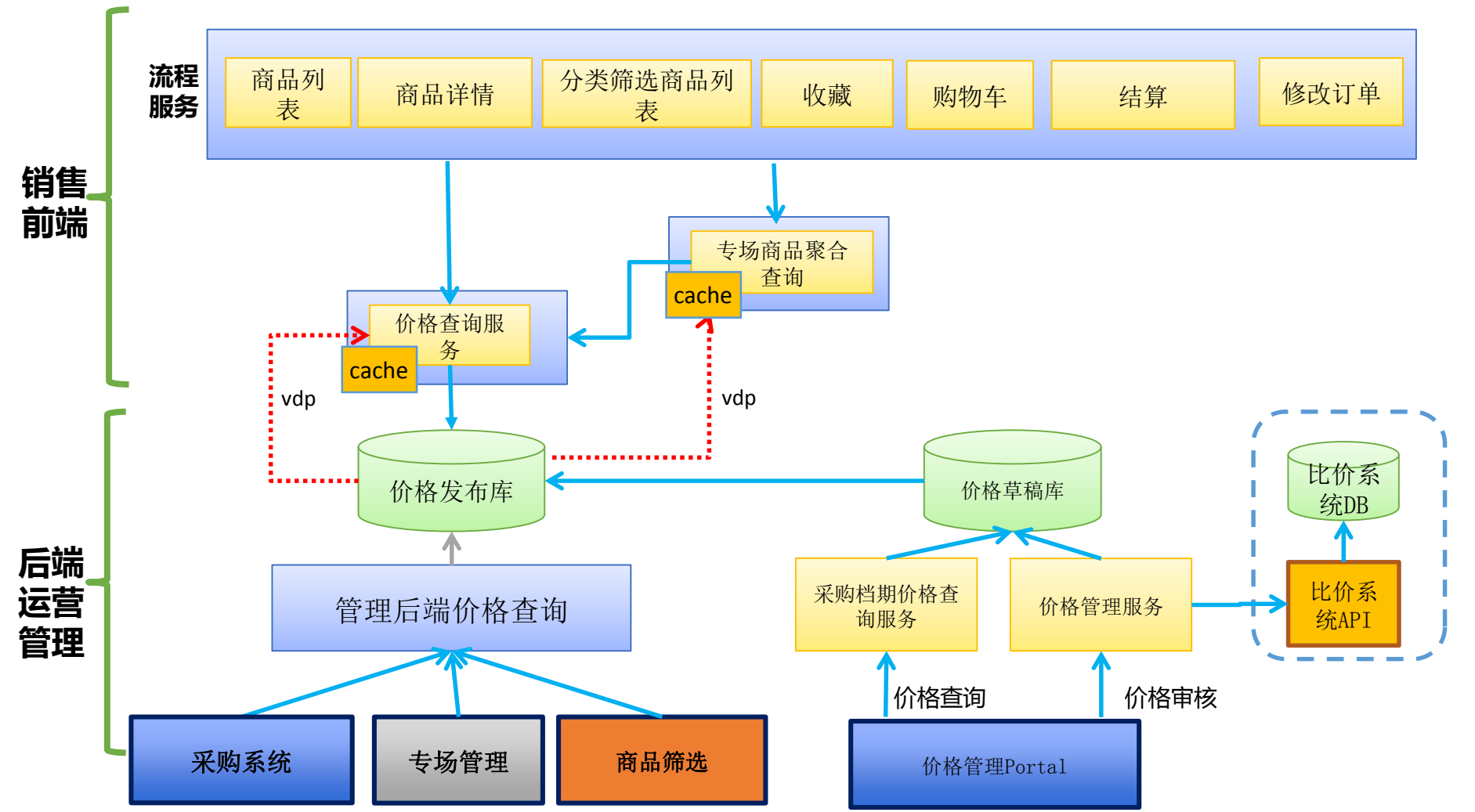
基于消息的数据同步方案



基于binlog的数据同步方案



数据一致性设计用例 - - - 价格变更



优化数据访问

- ◆访问量大的数据库做读写分离
- ◆访问量和数据量都很大的数据库，通过分库分表
- ◆合理利用缓存提高访问性能，结合本地缓存和分布式缓存
- ◆数据库有能力支撑时，尽量不要引入缓存
- ◆不同业务域数据库做分区隔离
- ◆数据库必须做归档
- ◆重要数据配置备库

优化数据访问用例：批量查询多个商品库存



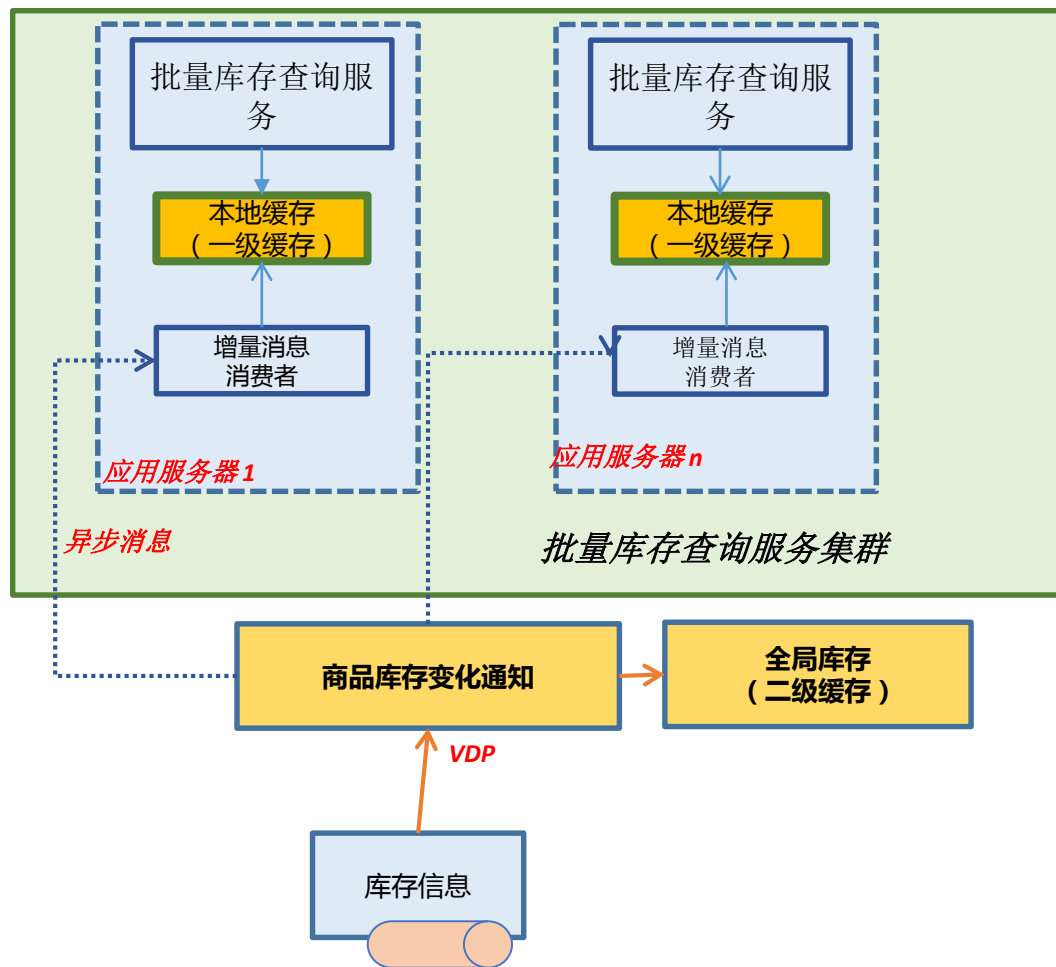
优化数据访问用例 - - - 批量库存查询

背景:

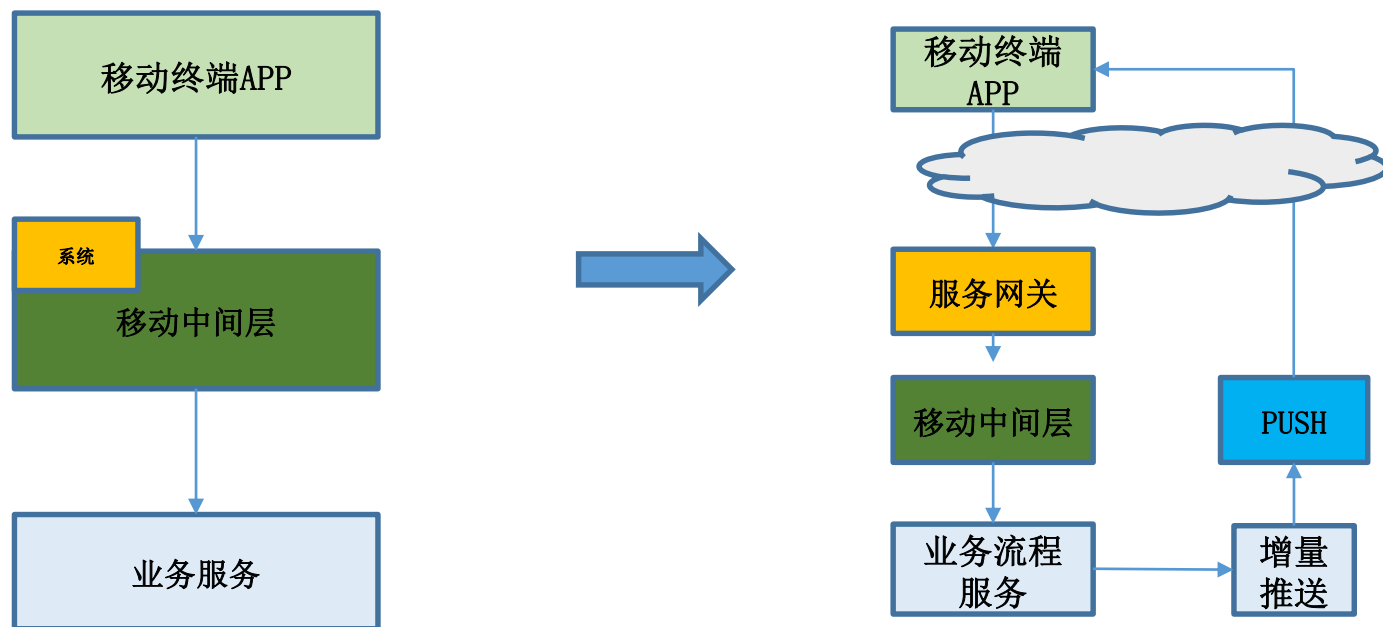
- ✓ 商品数量多，性能要求高，数据库或分布式缓存查询难以满足性能要求
- ✓ 库存变化频繁

解决方案:

- ✓ 通过本地缓存保存全量库存信息
- ✓ 库存变化通过数据推送器vdp触发异步消息
- ✓ 应用服务器消费库存变动消息并同步本地缓存
- ✓ 批量查询服务直接查询本地缓存获取库存信息
- ✓ 部署发布，系统扩容或者定时库存对数时，通过二级缓存，避免数据库过载访问

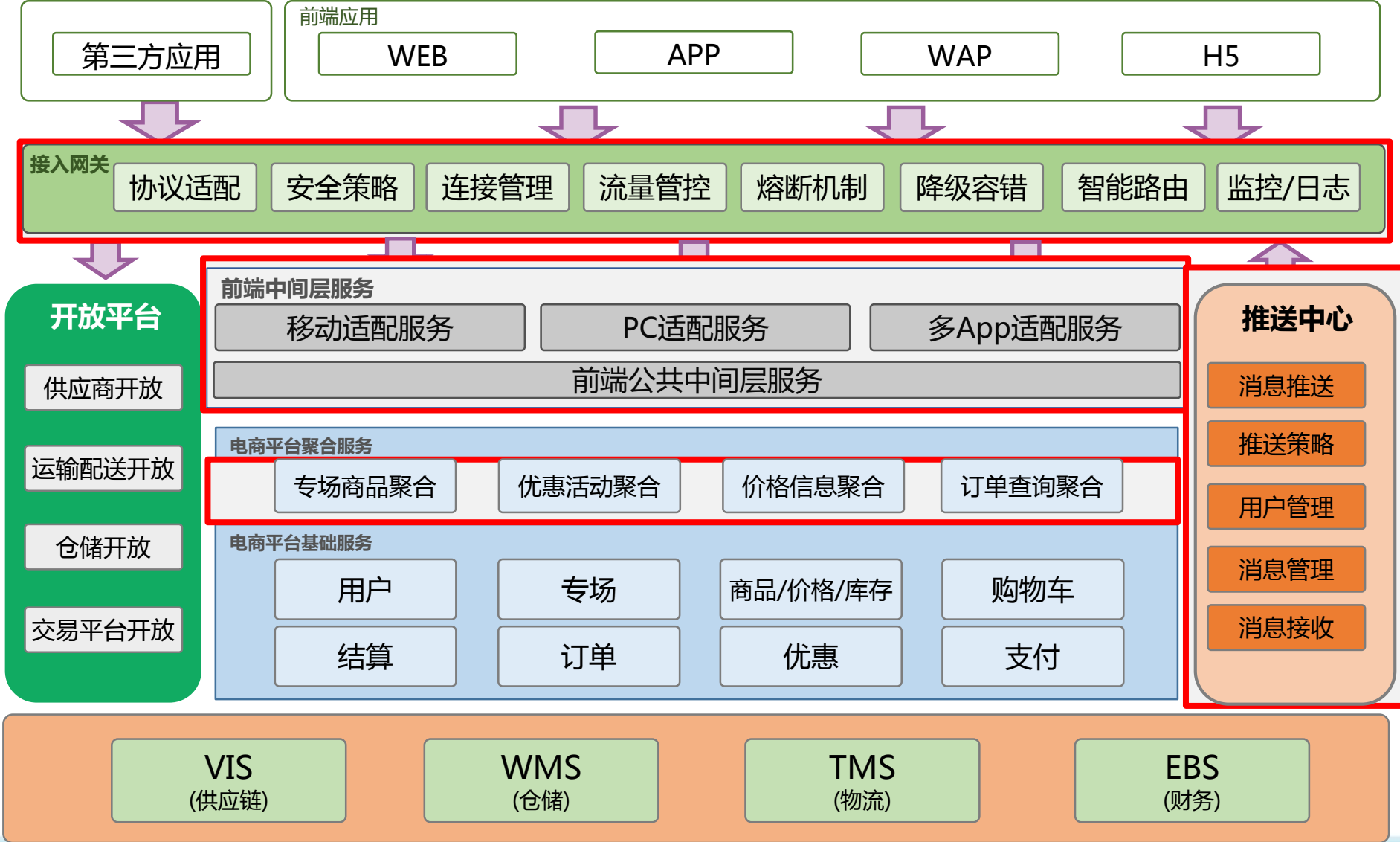


移动中间层改造思路



- ◆ 引入独立服务网管，服务网管负责公共逻辑，如协议适配、安全策略、流量管理、熔断机制、降级容错、智能路由、监控
- ◆ 中间层业务逻辑下沉，简化中间层业务逻辑
- ◆ 由业务流程服务提供对应聚合服务
- ◆ 增量数据通过推送中心推送到移动APP

前端中间层改造



系统整体架构



THANKS



[北京站]

主办方 **Geekbang** & **InfoQ**
极客邦科技