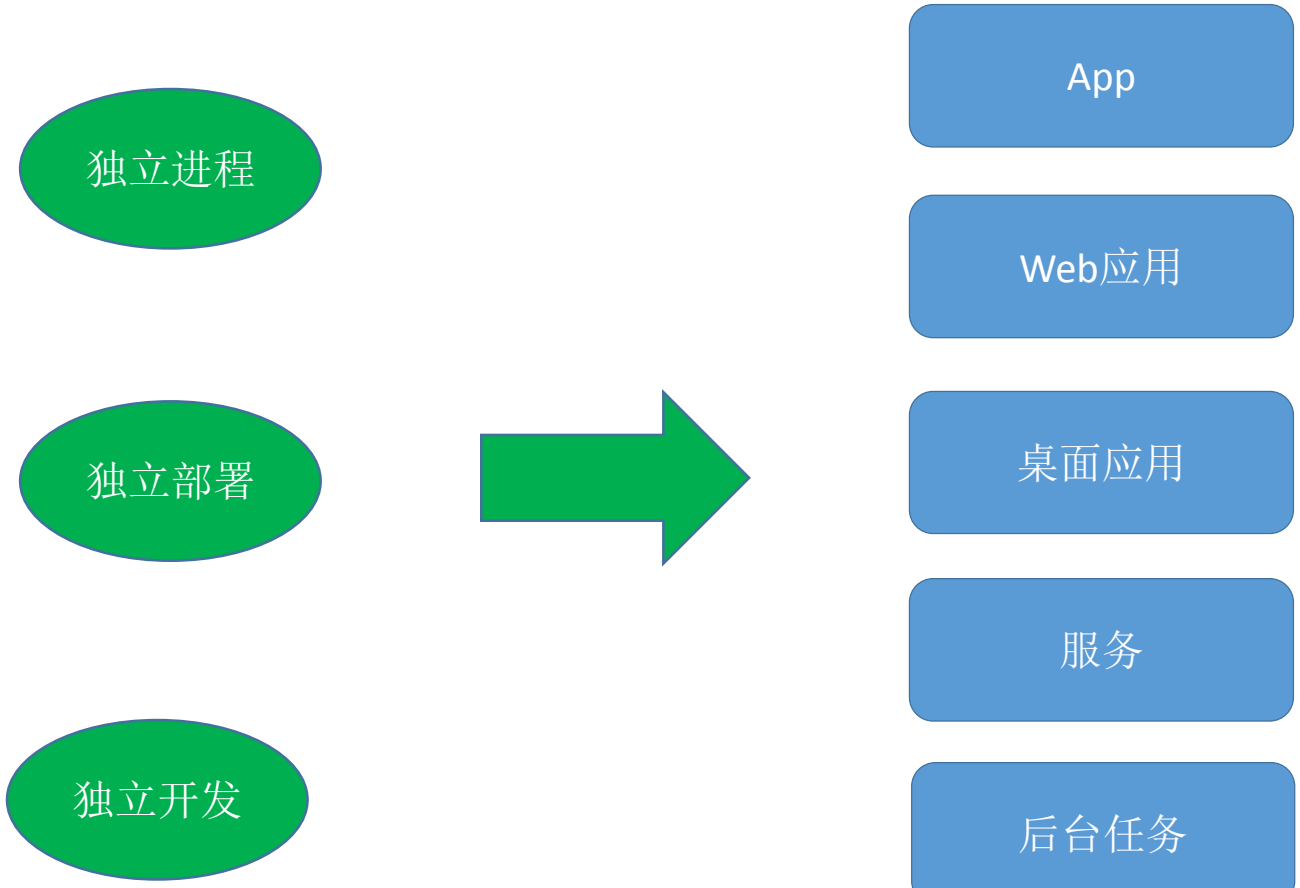
应用架构如何与时俱进

王庆友

目录

什么是应用

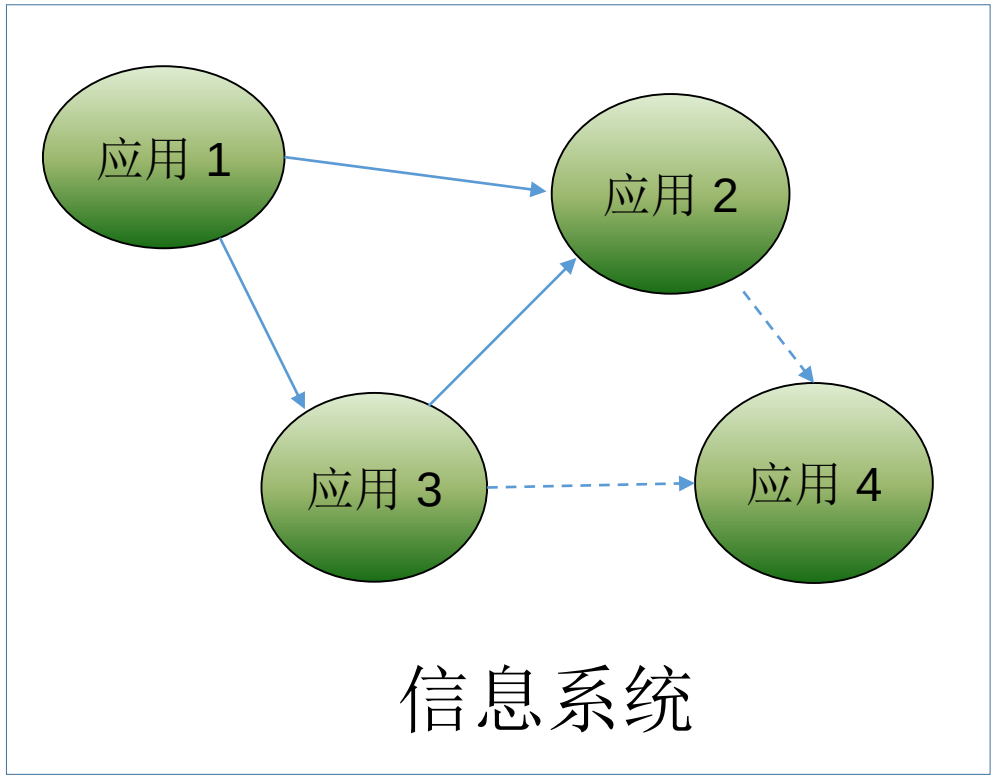
可独立运行的程序代码，提供相对完整的业务功能。



应用架构

把整体系统拆分为多个应用：

- 应用边界划分
- 依赖关系和调用



调用机制
(同步/异步/数据共享)

通讯协议
(Http/WS/Socket)

数据格式
(文本/XML/二进制)

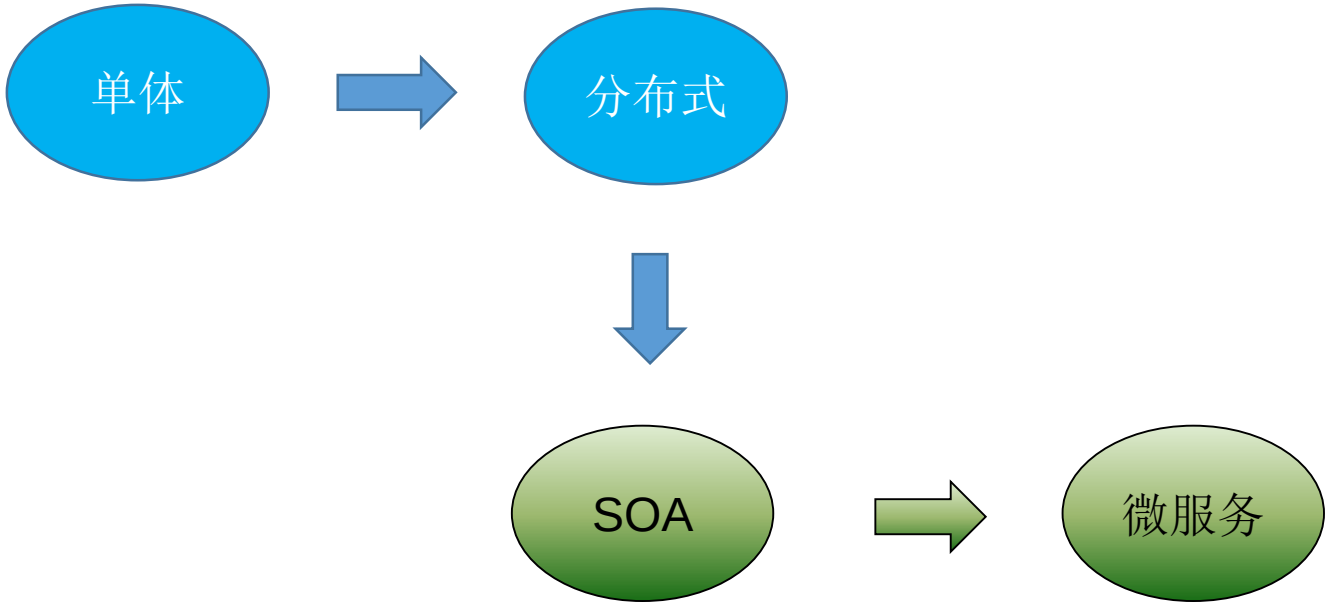
不同架构关系

概念模型
(战略)

逻辑模型
（战术）

物理模型
（装备）

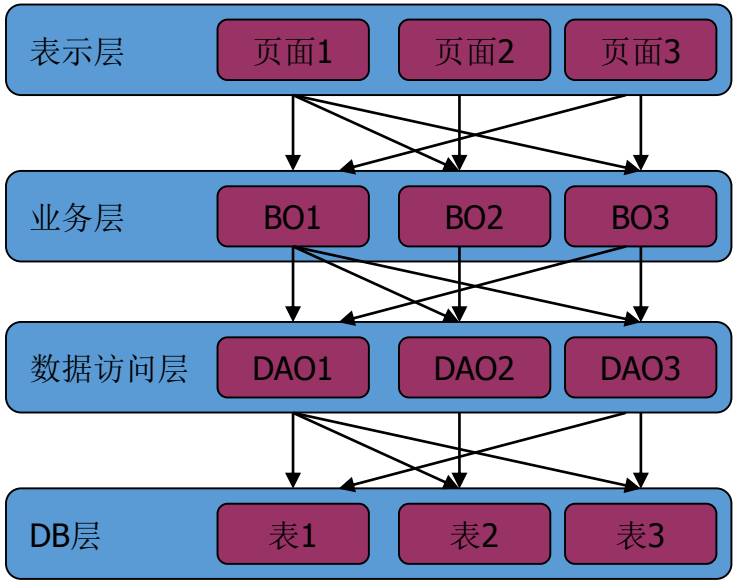
发展过程



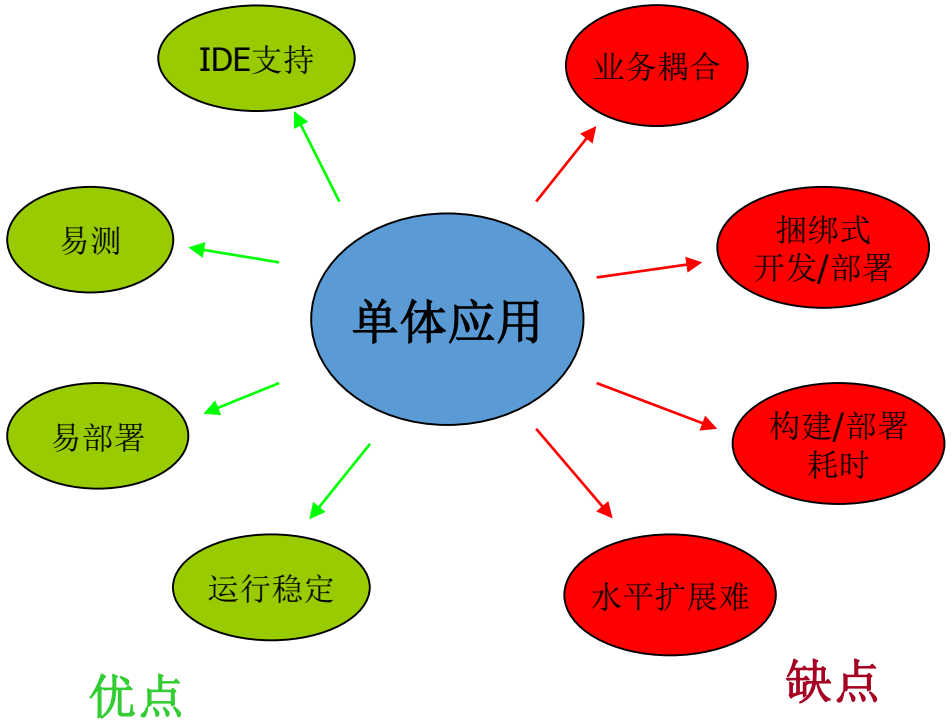
拆分更细，职责更清晰。

目录

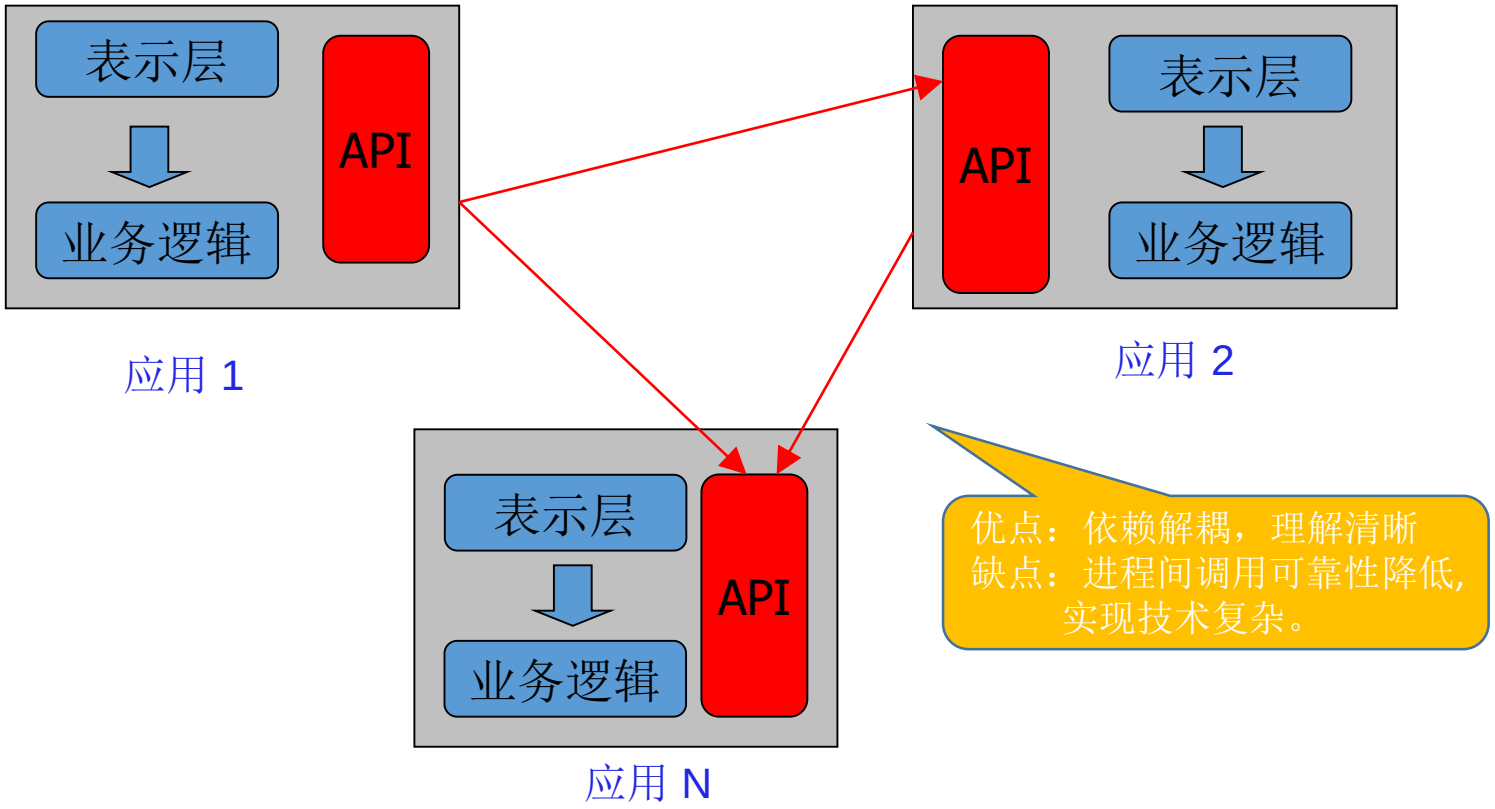
单体架构


表示层
页面1
页面2
页面3
业务层
BO1
BO2
BO3
数据访问层
DAO1
DAO2
DAO3
DB层
表1
表2
表3

水平分层（逻辑）

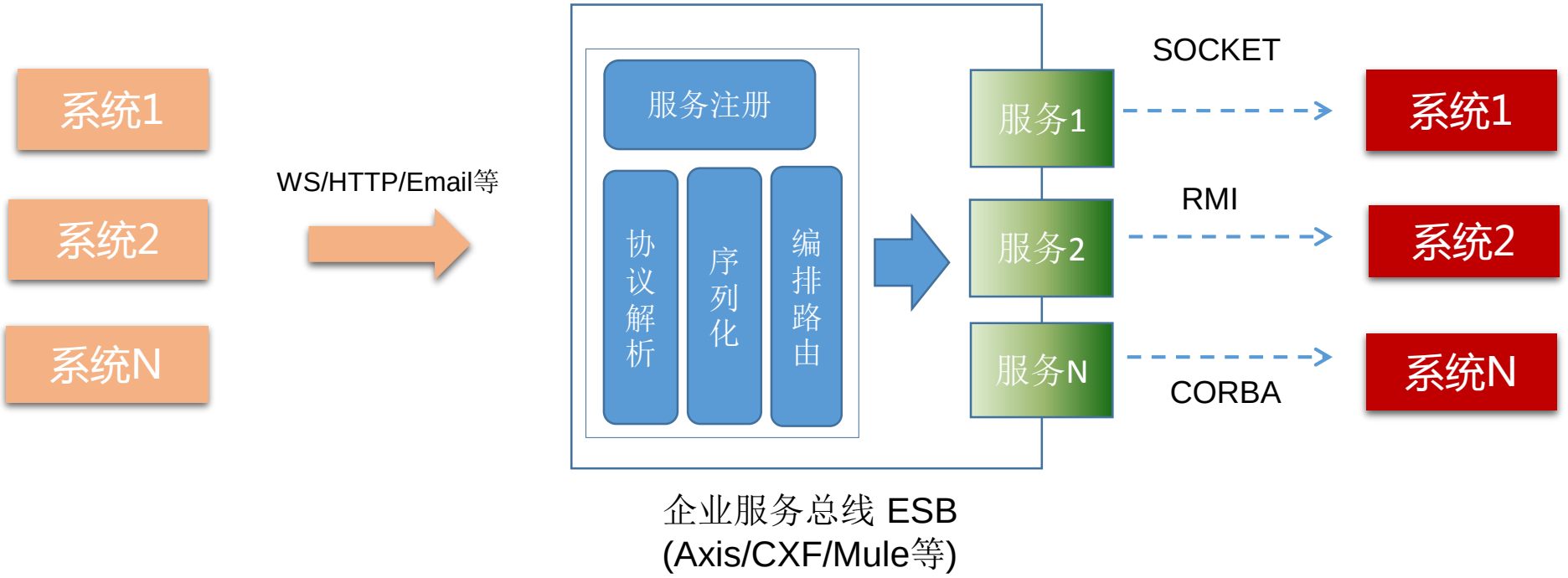

IDE支持
业务耦合
易测
单体应用
捆绑式
开发/部署
易部署
构建/部署
耗时
运行稳定
水平扩展难
优点
缺点

分布式架构



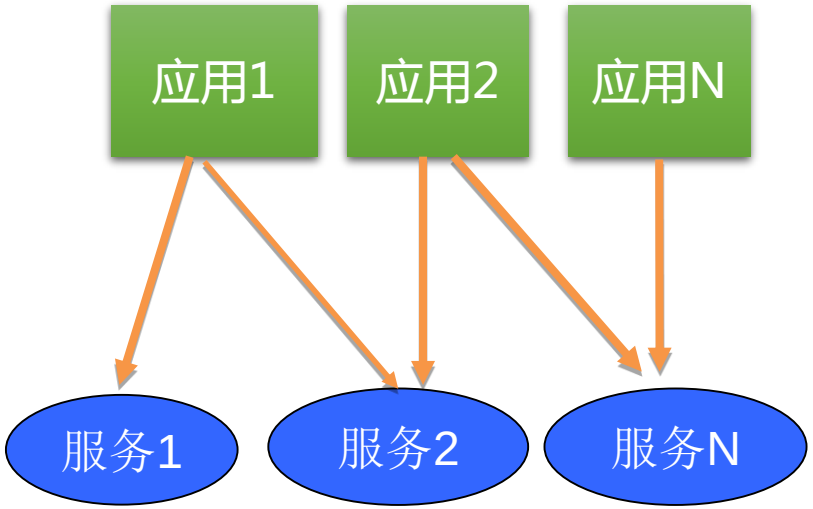
按照业务垂直切分，每个应用是单体架构，通过API互相调用。

传统SOA架构

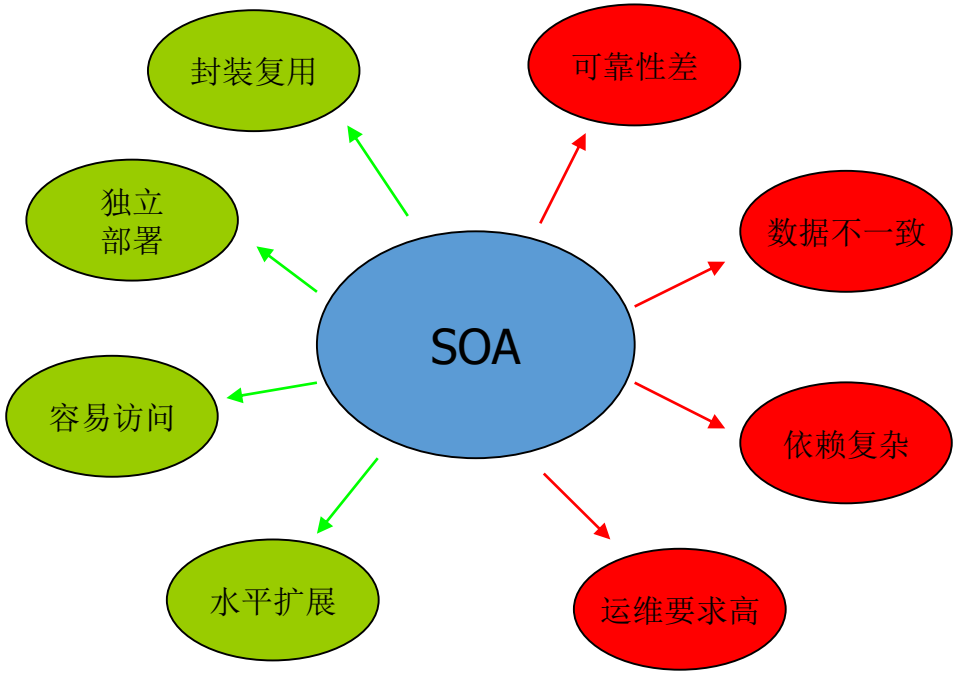


- 重量级通讯，粗粒度接口。
- 集中式总线管理，包含复杂系统和业务逻辑。
- 用于企业异构系统集成，没有大规模流行。

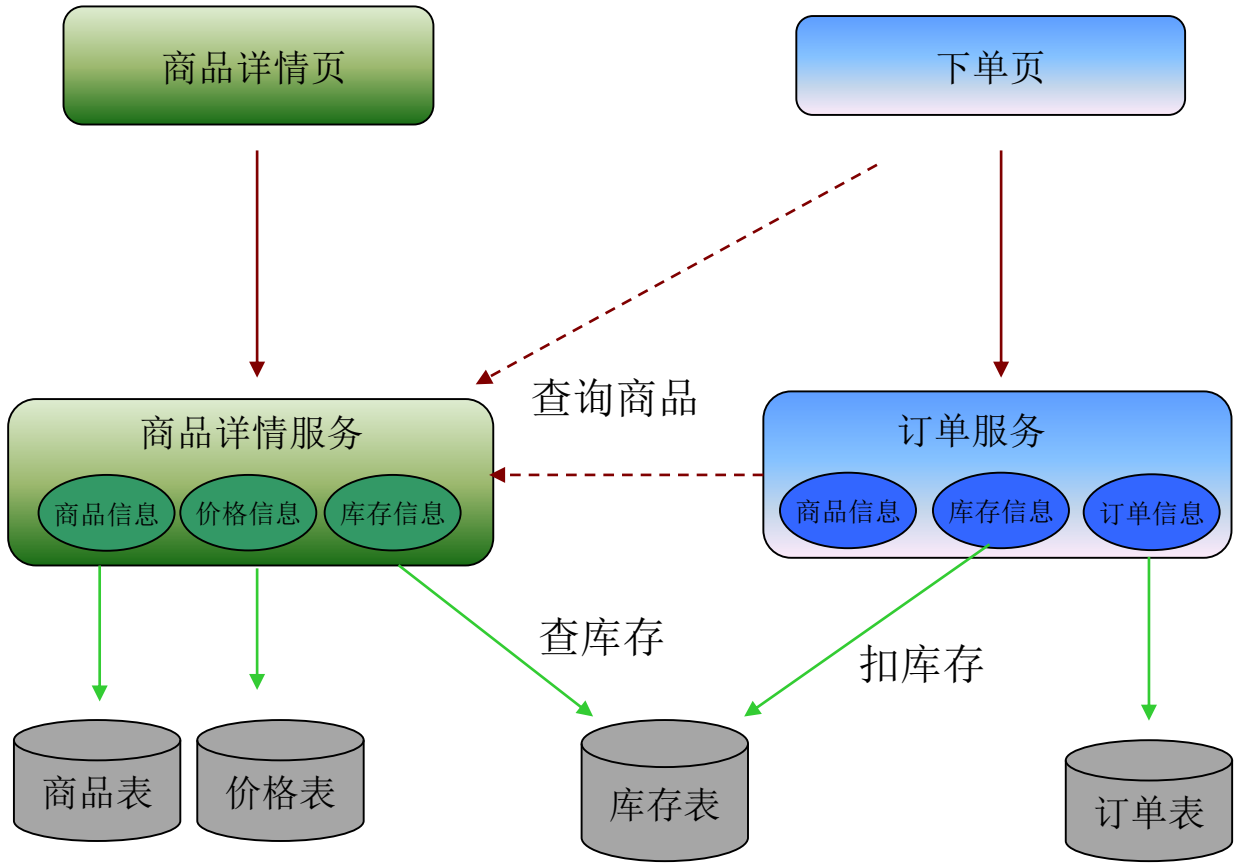
新型SOA



- 轻量级访问
- 核心业务封装
- 独立部署



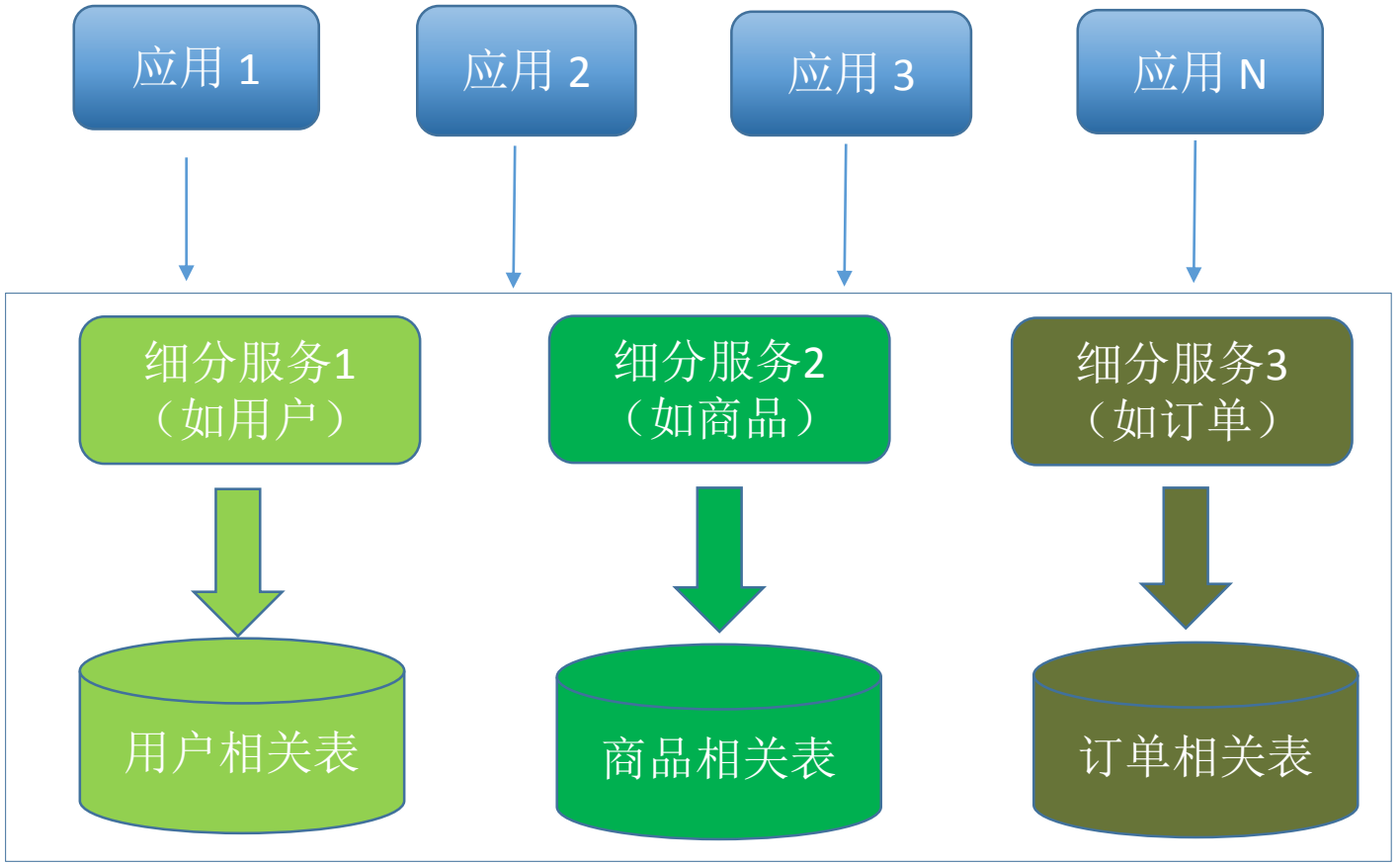
面向业务的服务



面向特定业务系统，提供粗粒度服务。
整体网状依赖(应用/服务/数据多对多）。

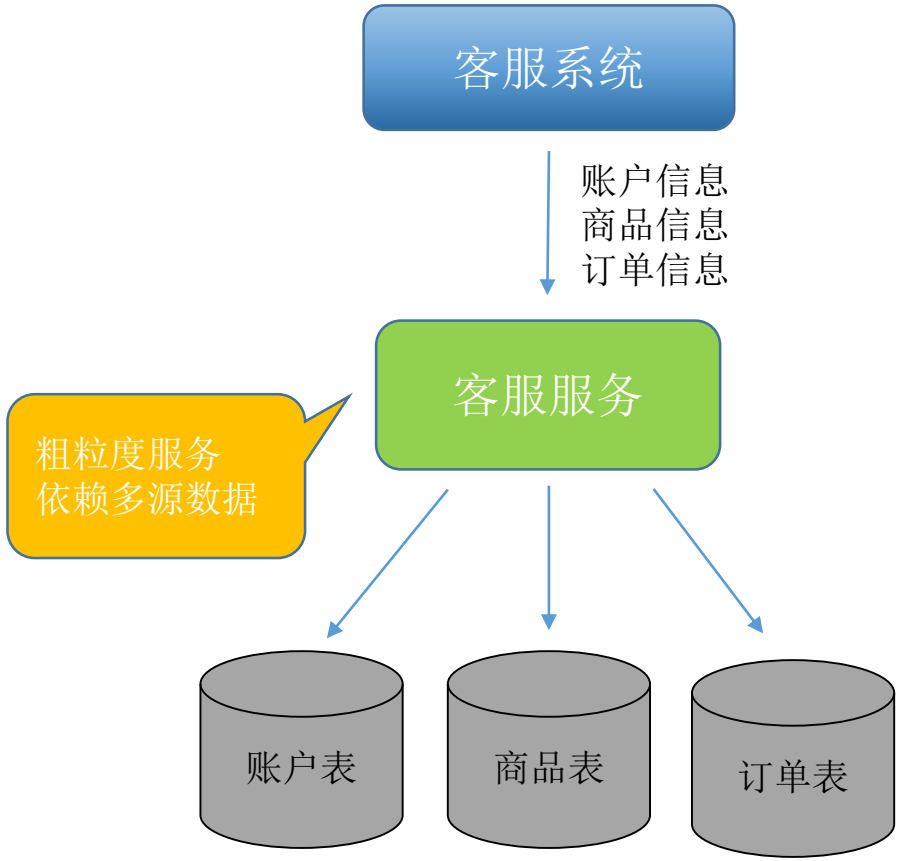
面向主题的服务

面向特定主题/概念/要素构建细分服务。

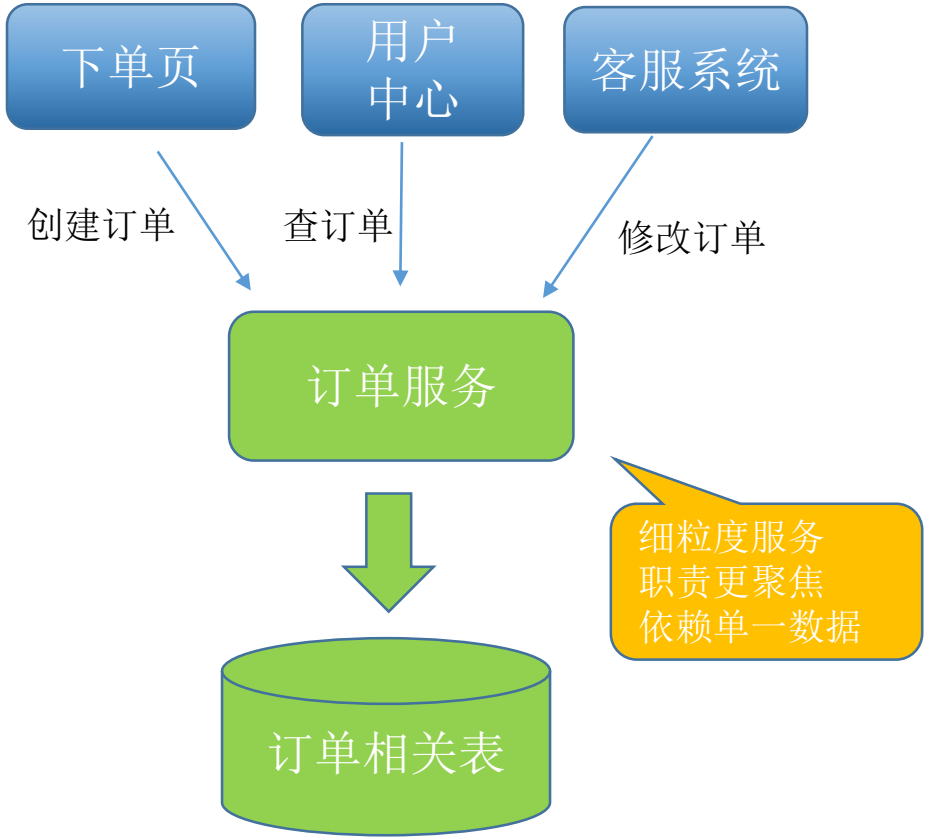


服务粒度比较

面向业务系统



面向细分主题



微服务思想

简单连接

- 轻量级通讯协议
- 简单数据格式
- 无中心节点
- 跨语言互通

分散管理

- 细粒度封装(业务分散)
- 独占式访问数据（数据分散）
- 独立部署(物理资源分散)
- 监控和治理

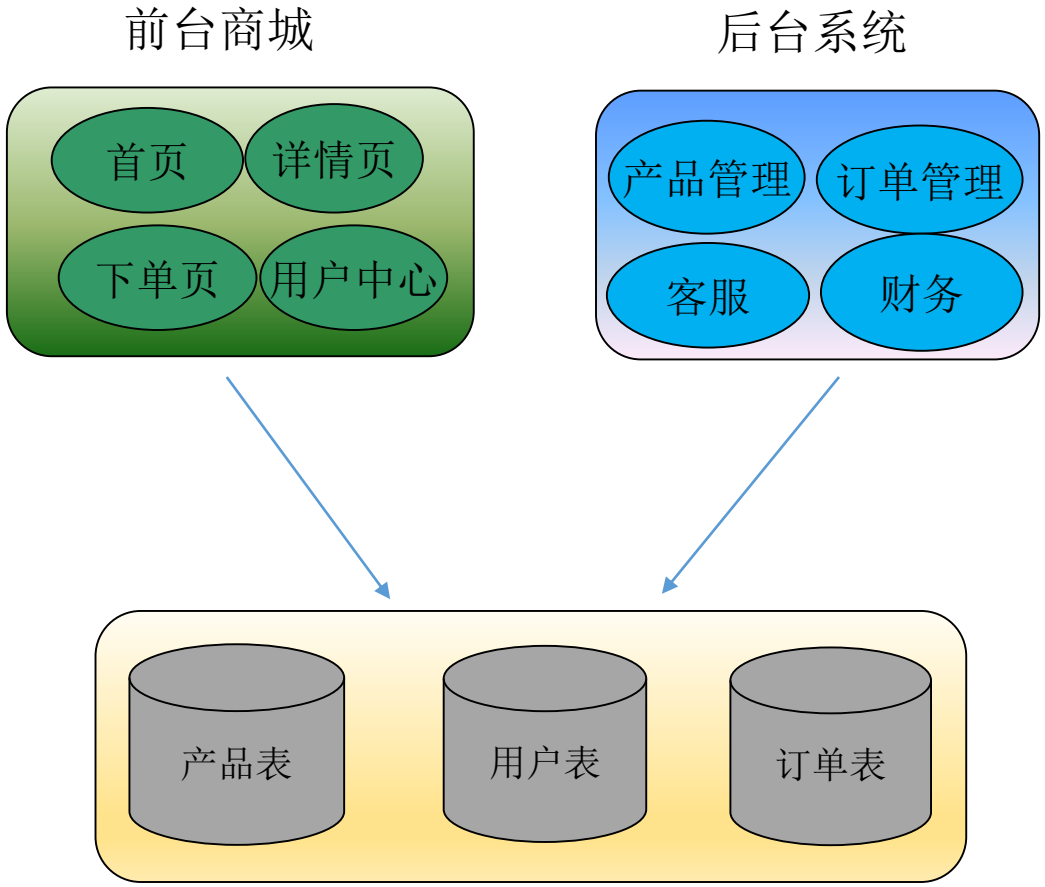
目录

应用架构演化

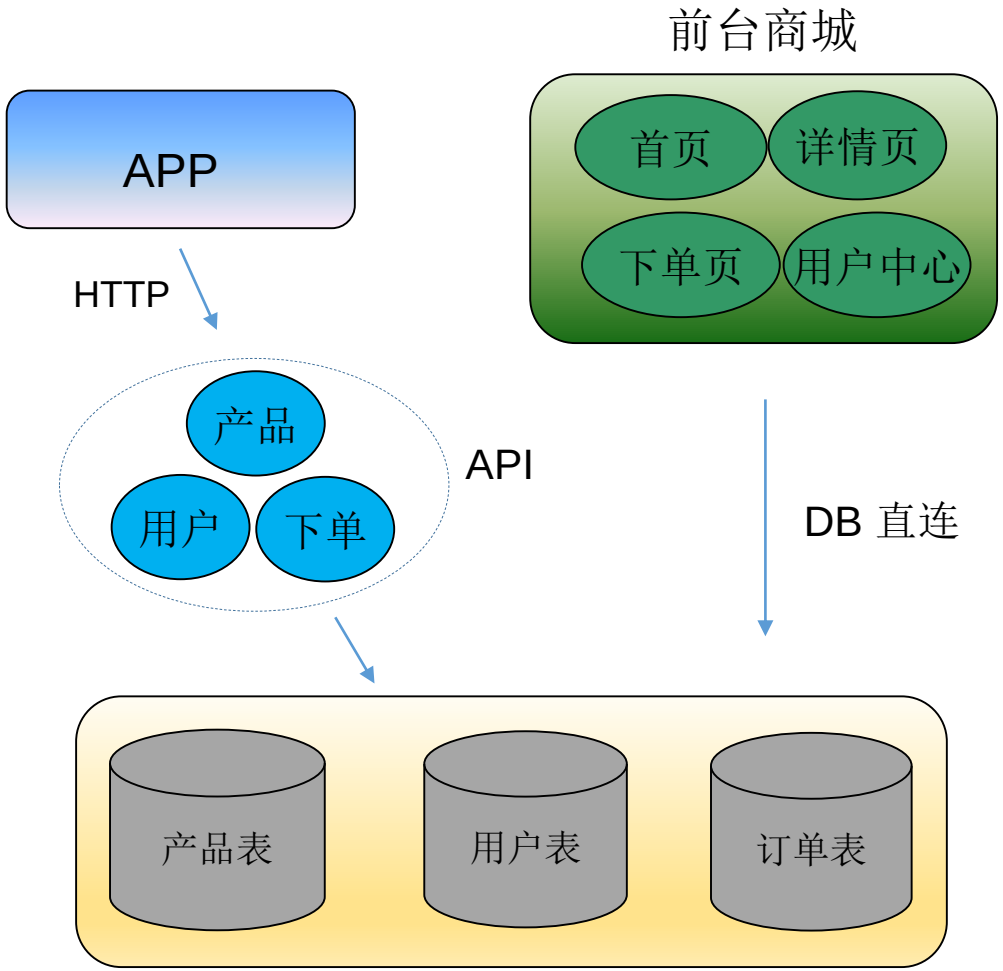
与时俱进

电商为例，介绍应用架构发展过程。

第一个系统-单体架构

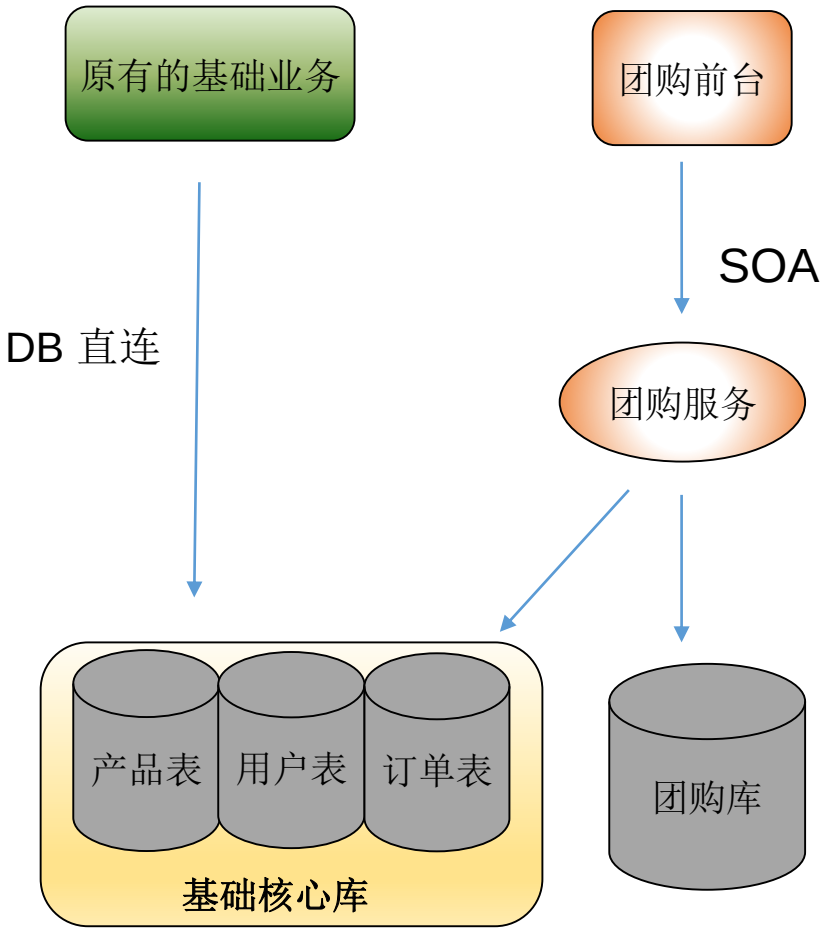


移动时代



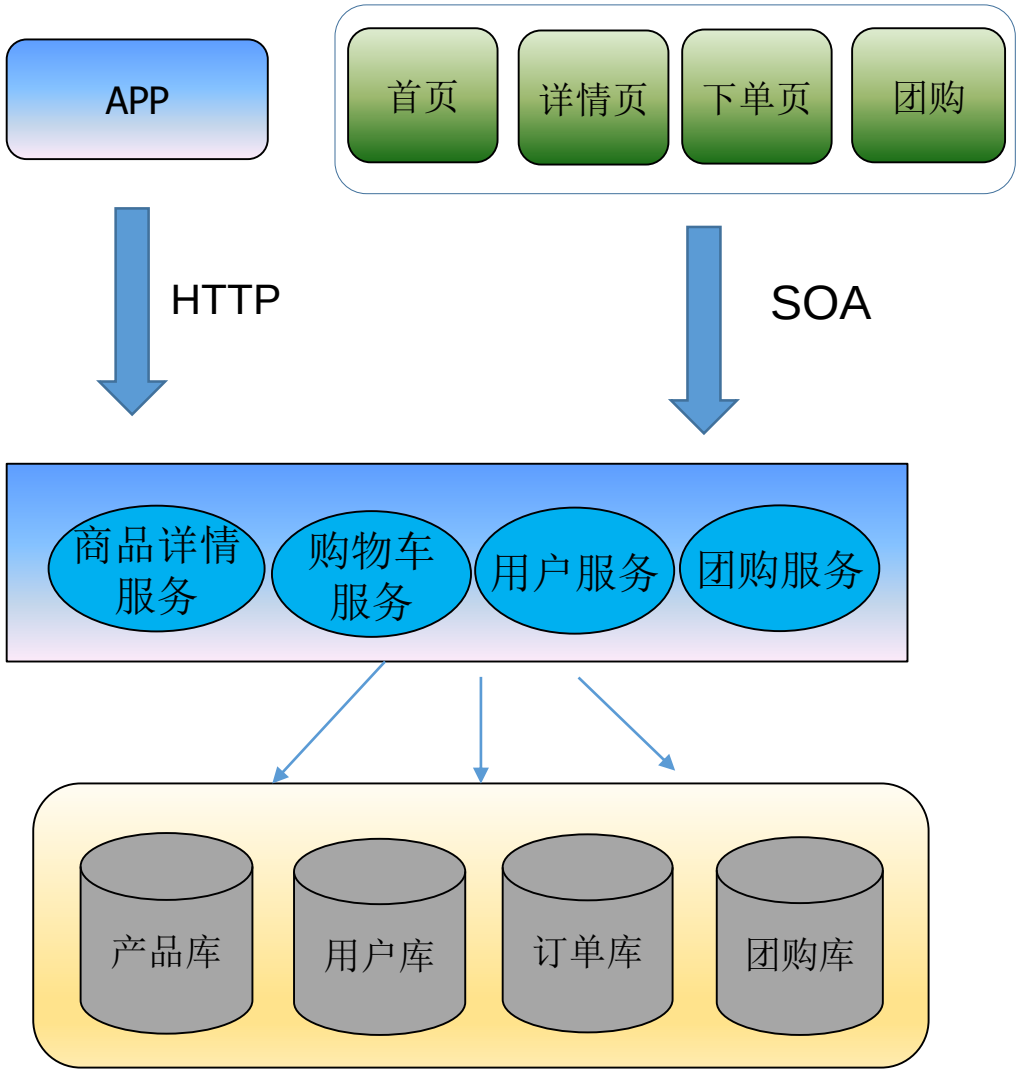
- APP 通过API层访问数据（不推荐在Web端直接提供接口）。
- API引用Web端的现有jar包访问DB（物理代码耦合）。

新业务独立-分布式架构



- 新业务以独立应用方式出现，变成简单的分布式架构。
- 面向新业务，单独构造服务，访问自有库和核心库。
- 该服务同时提供给APP端使用，SOA架构雏形出现。

业务分拆-SOA时代

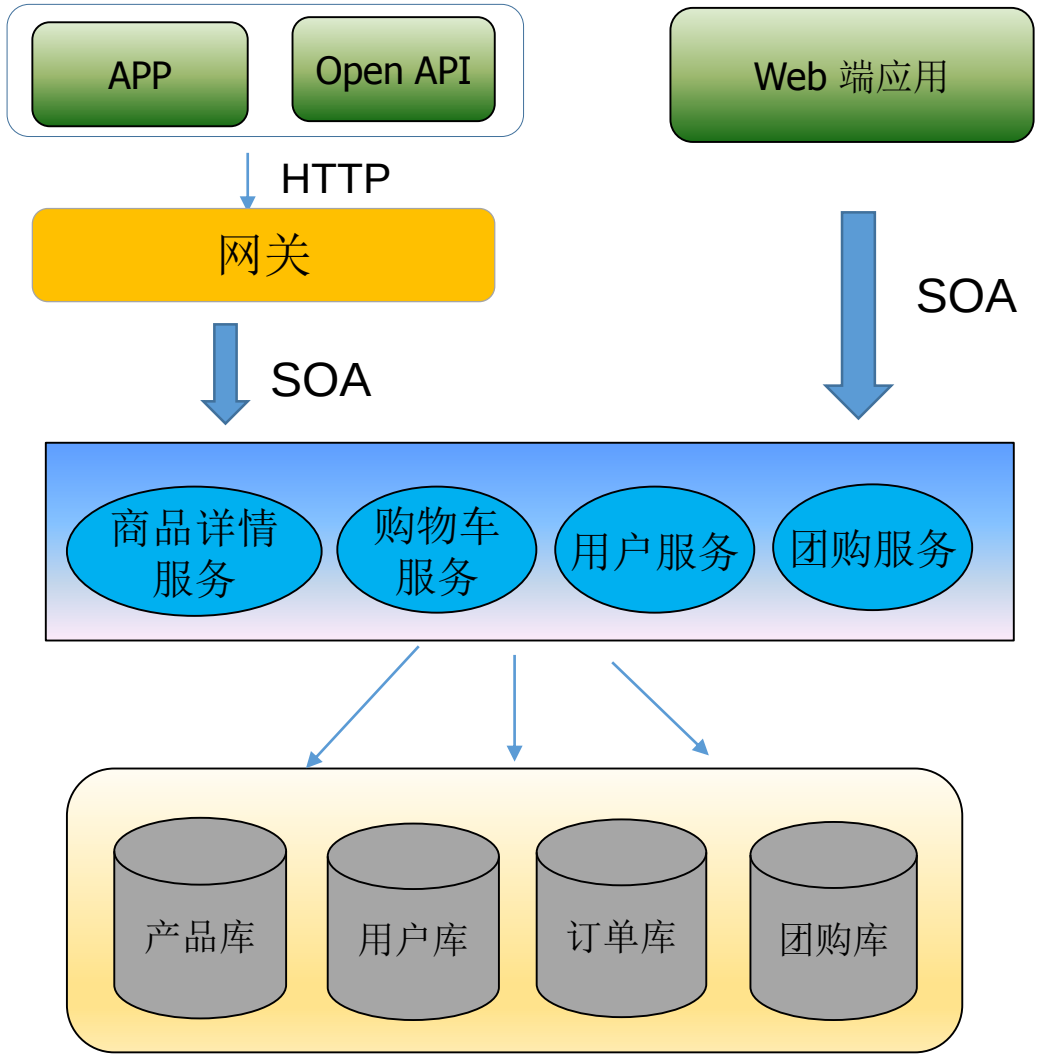


数十人共同维护老的工程，耦合严重，开发效率低下。

解决方案

- 旧的基础业务逐渐分拆，包括应用和数据库，彻底分布式。
- 在原有API基础上，进行服务化改造，同时供PC和APP端访问，打造SOA架构。

全面开放时代

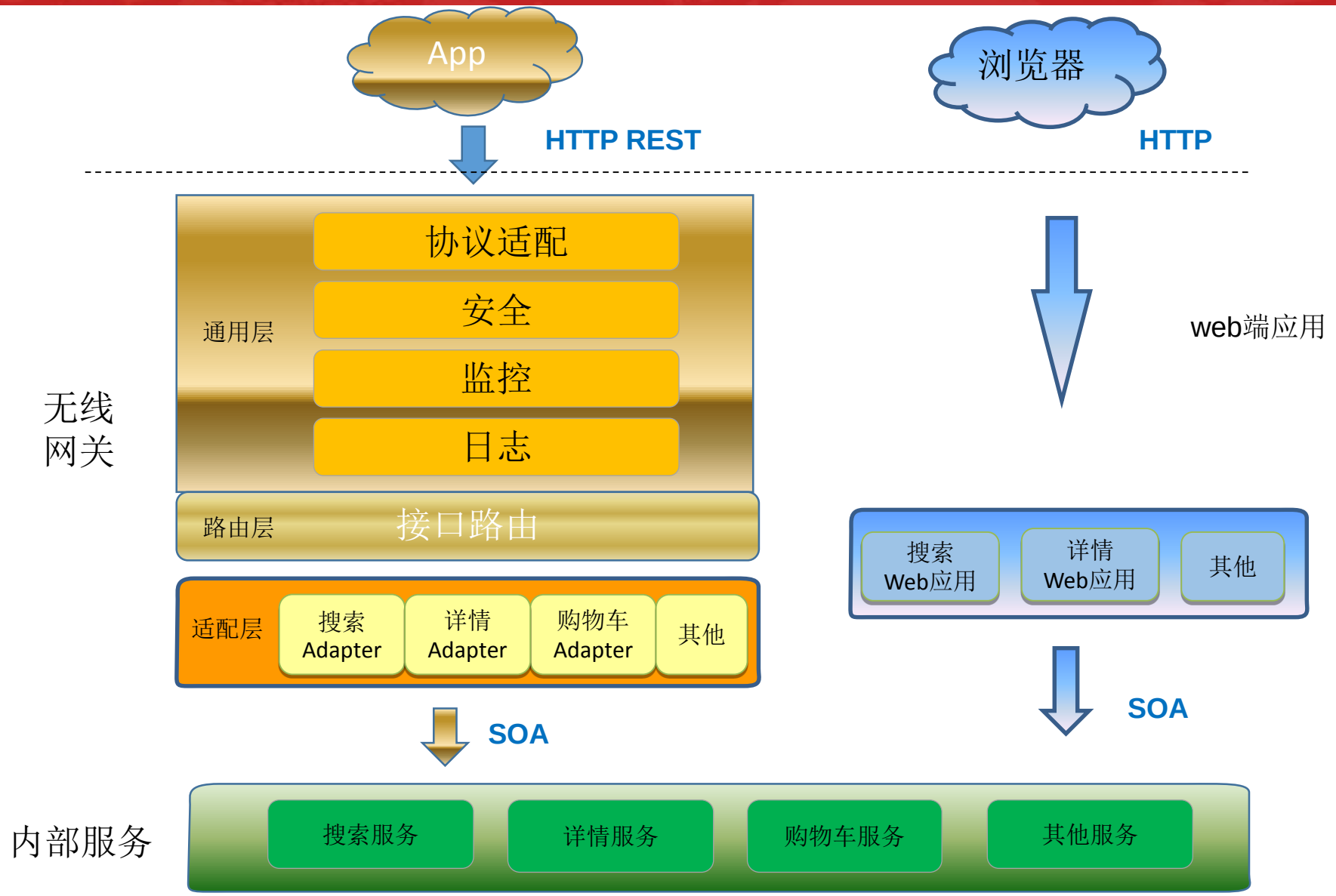


APP端和开放平台非功能性问题突出，包括安全性/性能/可用性，缺乏有效监控。

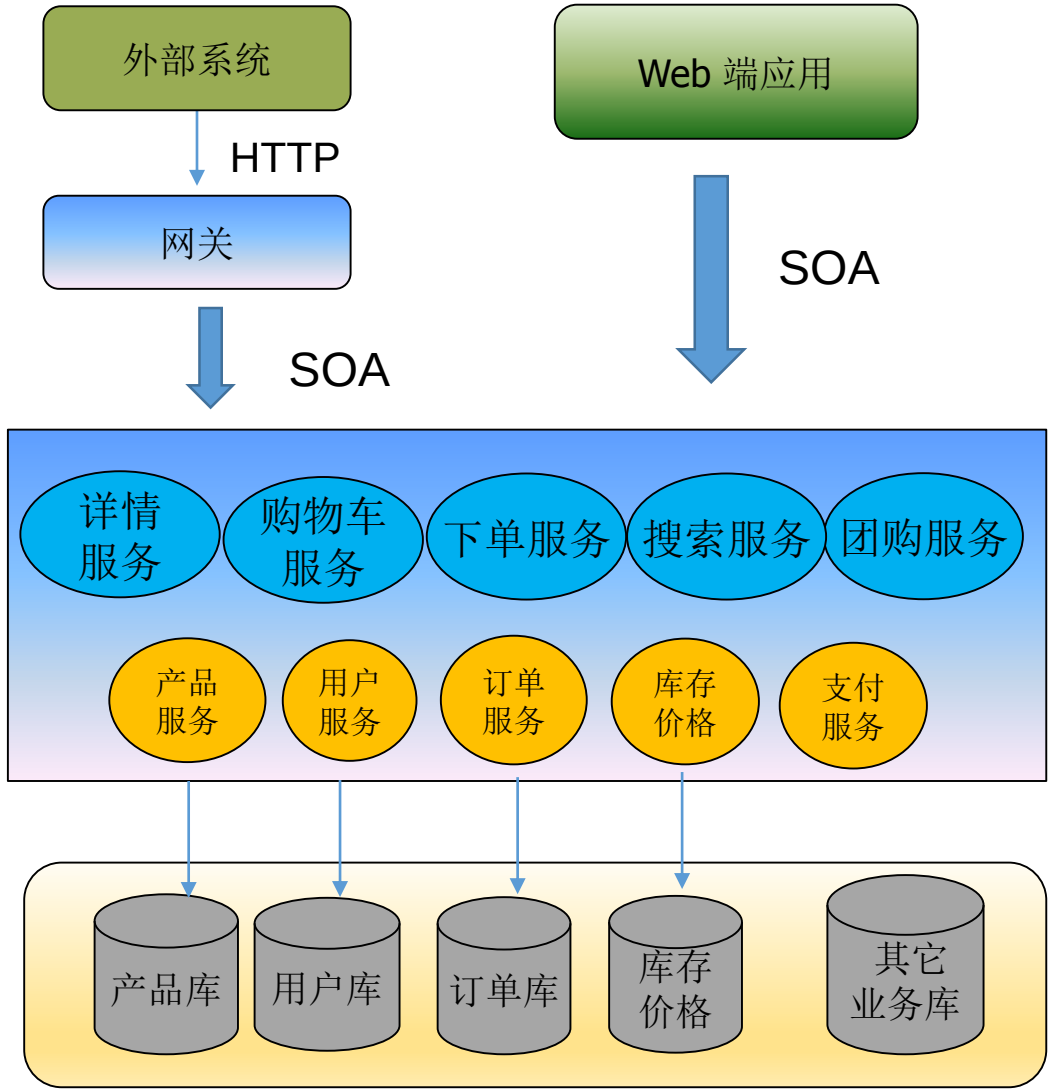
解决方案

- 独立的网关，提供系统级的通用功能和访问路由。
- 内部SOA架构，外部面向API的网关架构。

深入网关架构


App
浏览器
HTTP REST
HTTP
无线
网关
通用层
协议适配
安全
监控
日志
路由层
接口路由
适配层
搜索
Adapter
详情
Adapter
购物车
Adapter
其他
web端应用
搜索
Web应用
详情
Web应用
其他
SOA
SOA
内部服务
搜索服务
详情服务
购物车服务
其他服务

共享业务平台-微服务时代



业务背景

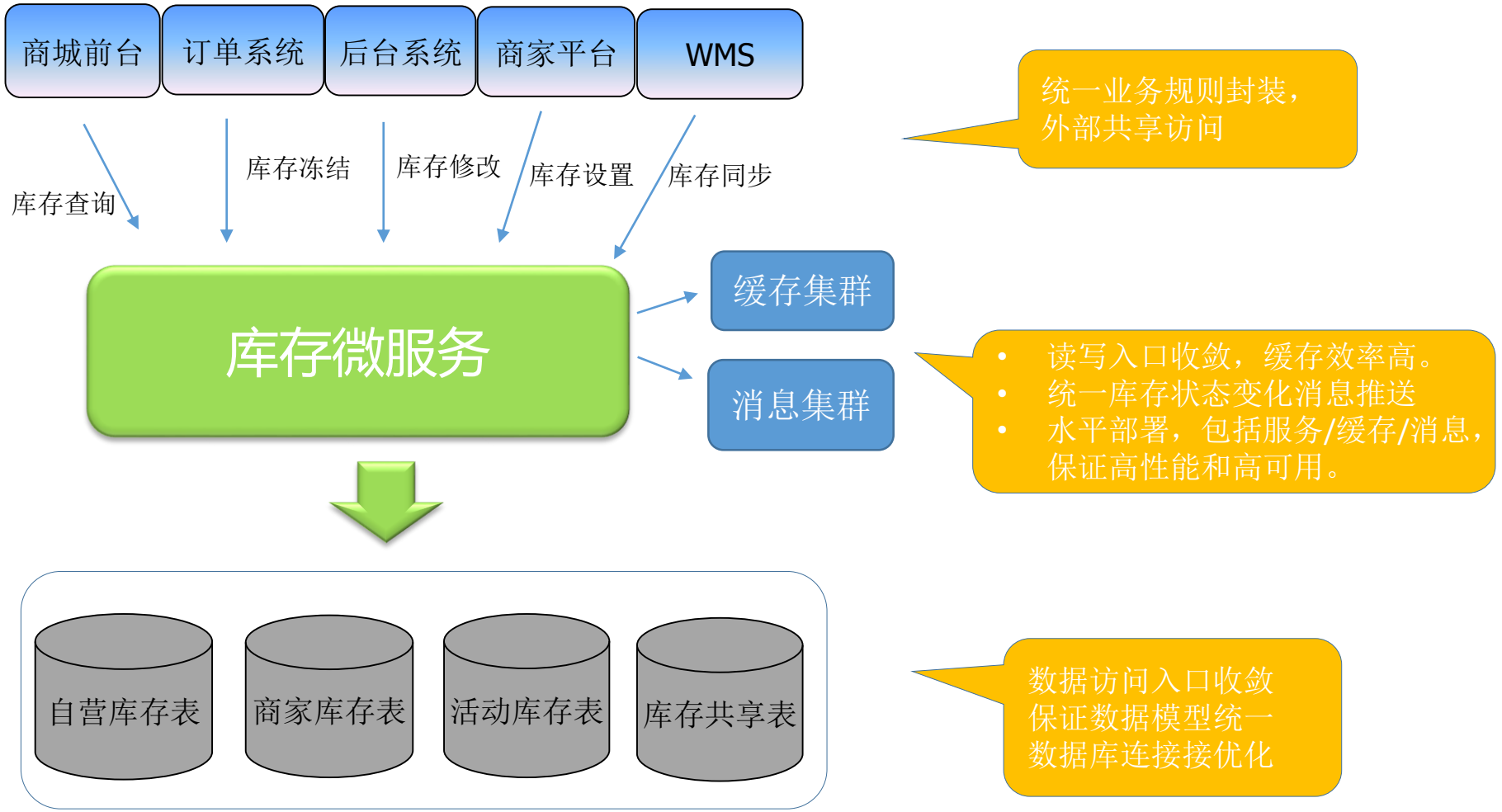
- 业务关系紧密，都涉及产品/订单/库存/价格等主数据。
- 业务规则碎片化，散布在多个系统中。
- 业务量大，要保证主数据访问的可用性和性能。

解决方案

针对主数据，单独构造细分服务，封装业务规则和数据，打造共享业务平台。

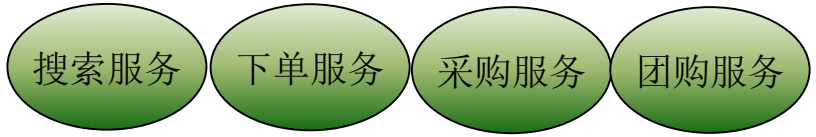
深入微服务（库存例子）

电商库存包括物理库存/虚拟库存/活动库存/共享库存/冻结库存。
可售库存=本地库存（TS-TSF）+虚拟库存（VS-VSF）+兄弟仓库共享库存（TS-TSF）
数十个系统调用，每天调用数亿次，上百台虚拟机支撑。



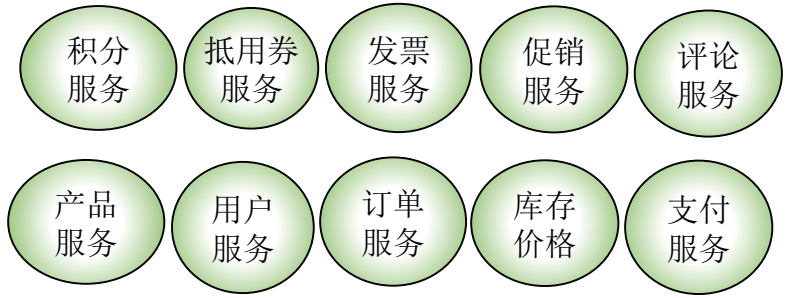
完整的服务体系

业务服务



为上层具体业务系统提供服务，在底层服务基础上，做信息或流程聚合。

共享服务



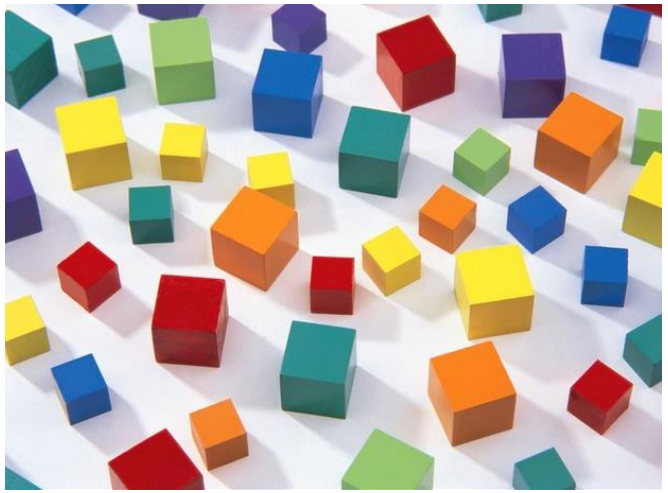
针对主题/要素提供专门服务，封装业务规则和数据资源。

系统服务

提供专门的系统层面服务，和业务无关，封装基础设施的访问。

基于微服务的电商系统架构

架构总结



分

合

分开是为了更好地相聚。

Thanks！

微信公众号：**software-arch**