

微服务架构设计与实践

总体架构篇



孙玄@58集团

关于我

✓ 58集团技术 **QCon** 全球软件开发大会 主席



✓ 58集团高级 **SDCC** 中国软件开发大会 架构师
Software Developer Conference China



✓ 转转架构师 **PROGRAMMER** 程序员 负责人



✓ 百度高级工程师

✓ 毕业于浙江大学

✓ 代表公司多次对外分享

✓ 企业内训&公开课



关于我

企业内训

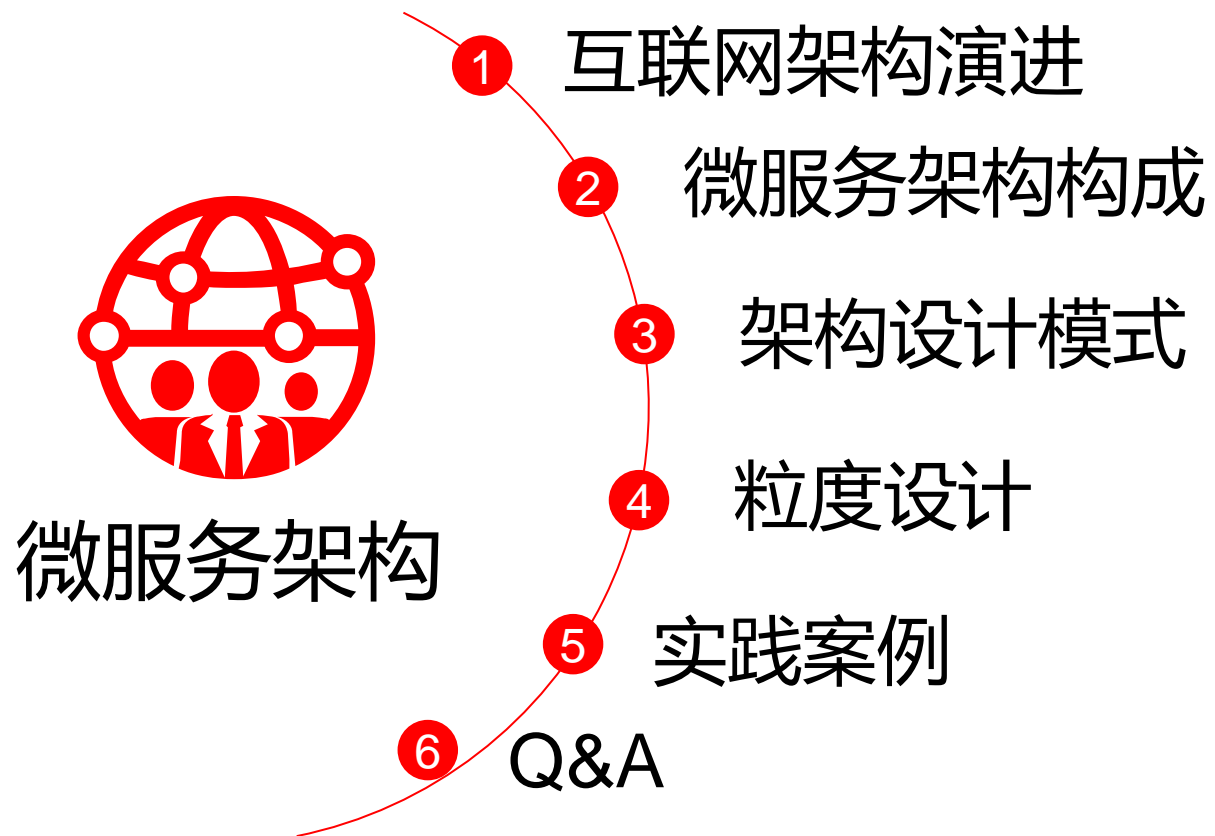
- ✓ 华为
- ✓ 中航信
- ✓ 平安
- ✓ 银联
- ✓ 华泰证券
- ✓ 思科

- ✓ 云南电力
- ✓ 深信服
- ✓ 新华社
- ✓ 民生银行
- ✓ 招商银行
- ✓

公开课

- ✓ 北京
- ✓ 上海
- ✓ 深圳
- ✓ 广州
- ✓ 成都
- ✓

分享要点



互联网架构演进

Monoliths
架构

水平分层
架构

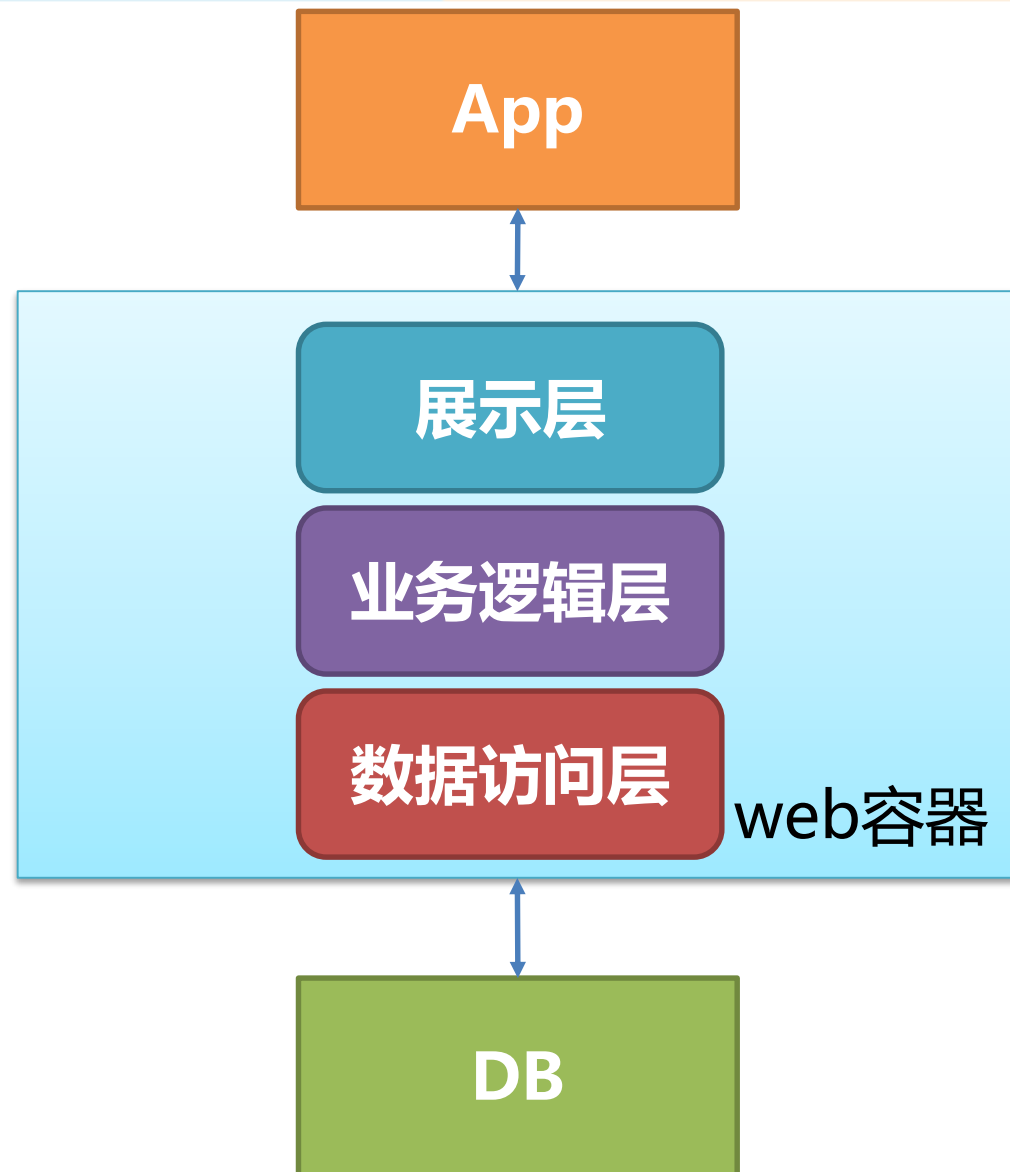
SOA
架构

微服务
架构



互联网架构演进

- Monoliths



互联网架构演进

- Monoliths

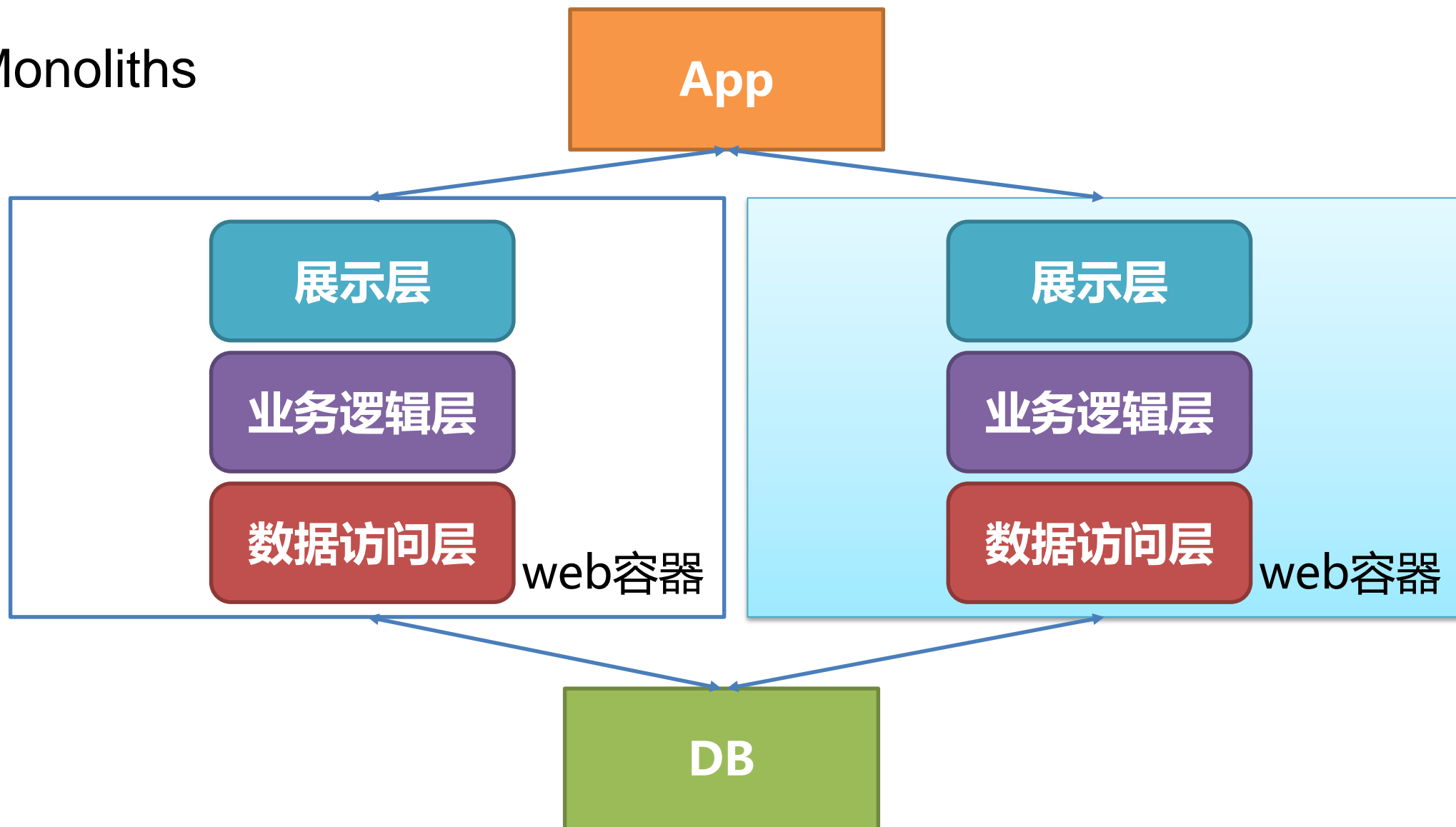
缺点

- ✓ 系统耦合性高
- ✓ 开发效率低
- ✓ 技术选项单一
- ✓ 扩展性差



互联网架构演进

- Monoliths



互联网架构演进

- 水平分层

解决Monoliths架构问题

低耦合

高内聚

开发效率高

扩展性高



互联网架构演进

- 水平分层

- 分层设计原则



互联网架构演进

- 水平分层



互联网架构演进

● 水平分层

过多

- ✓ 请求路径长
- ✓ 平均响应延迟高
- ✓ 定位问题复杂
- ✓ 运维成本增加



互联网架构演进

- 水平分层



✓ Monoliths



互联网架构演进

● 水平分层

适中

- ✓ 三层 (MVC)
- ✓ 四层 (接入层、业务逻辑层、数据访问层、数据层)
- ✓ 五层 (接入层、**异步提交层**、业务逻辑层、数据访问层、数据层)



互联网架构演进

- 水平分层

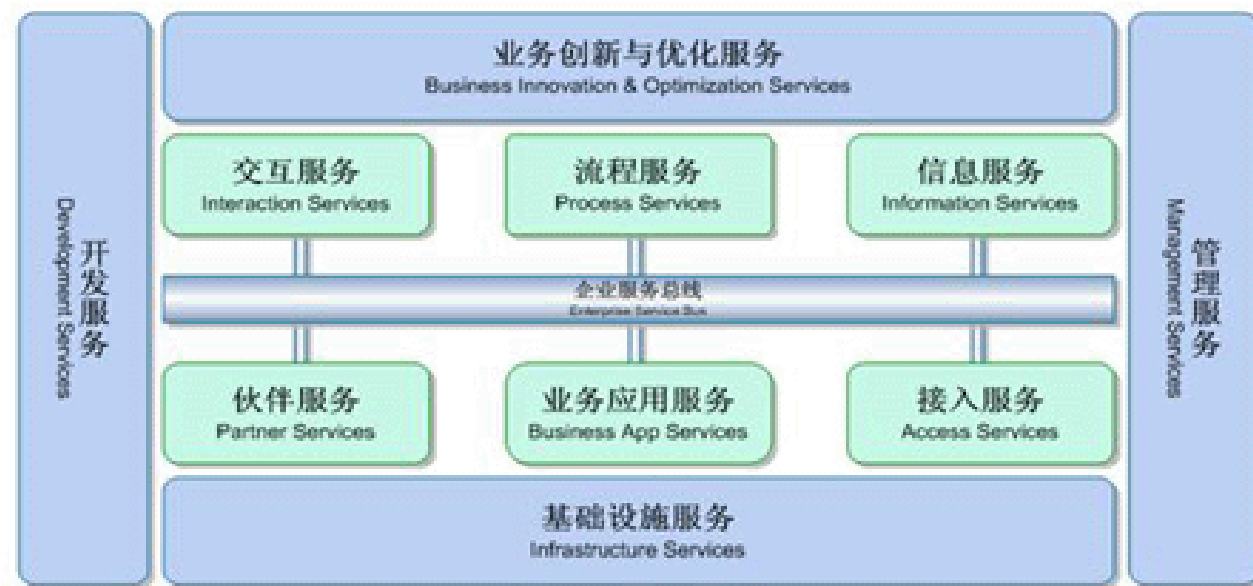
缺点

✓ 每层粒度较粗



互联网架构演进

- SOA



互联网架构演进

- SOA

缺点

- ✓ 每个服务Monoliths
- ✓ 对ESB依赖严重



互联网架构演进

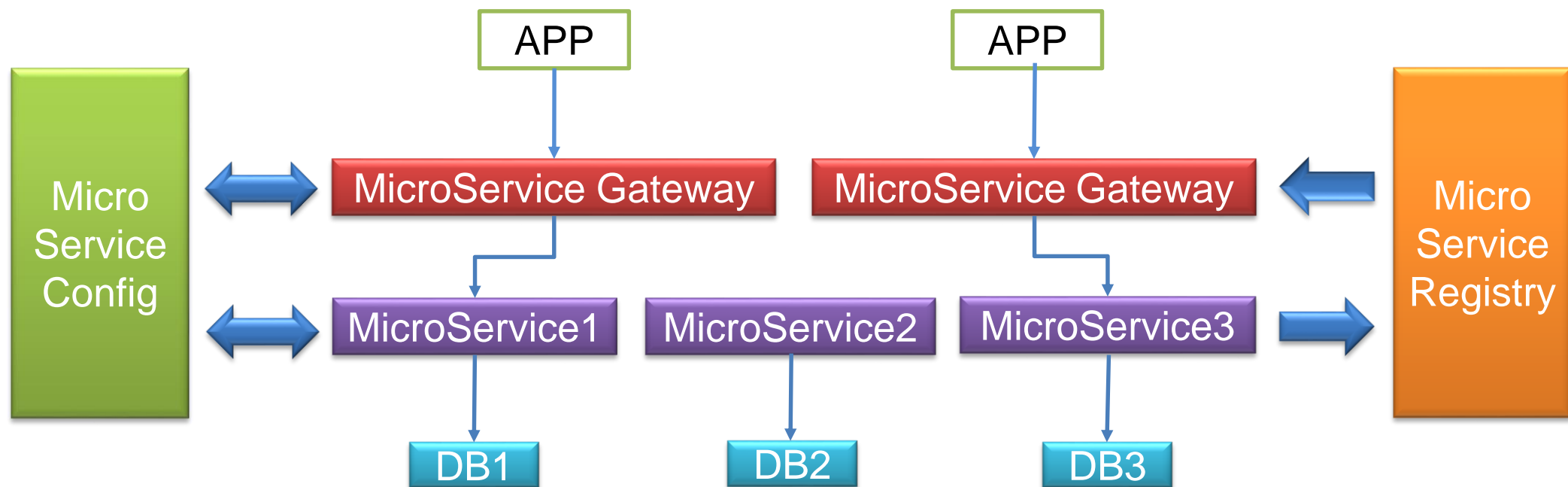
- 微服务

解决水平
分层/SOA
问题

- ✓ 水平分层
- ✓ 每层微服务继续拆分



微服务架构



微服务架构构成



微服务架构



微服务网关层



微服务层



数据层



微服务注册中心



微服务配置中心



微服务发现

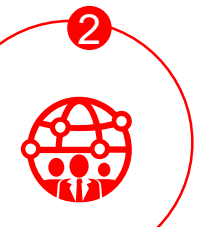


微服务架构构成

● 微服务网关层



- ✓ 海量用户接入管理
- ✓ 请求合法性和安全校验
- ✓ 负责简单的antispam
 - IP、Token过滤
 - 访问频率控制
 - 黑白名单等
- ✓ 请求路由
- ✓ 请求回包

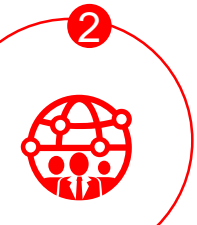


微服务架构构成

● 微服务层



- ✓ 数据访问处理
- ✓ 业务逻辑处理

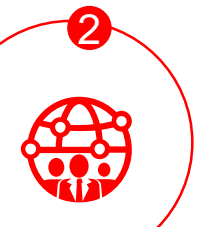


微服务架构构成

● 数据层



- ✓ 持久化存储（MySQL、MongoDB等）
- ✓ 缓存加速（Memcached、Redis等）

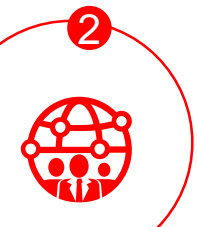


微服务架构构成

● 注册中心

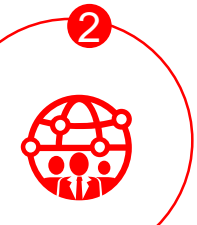
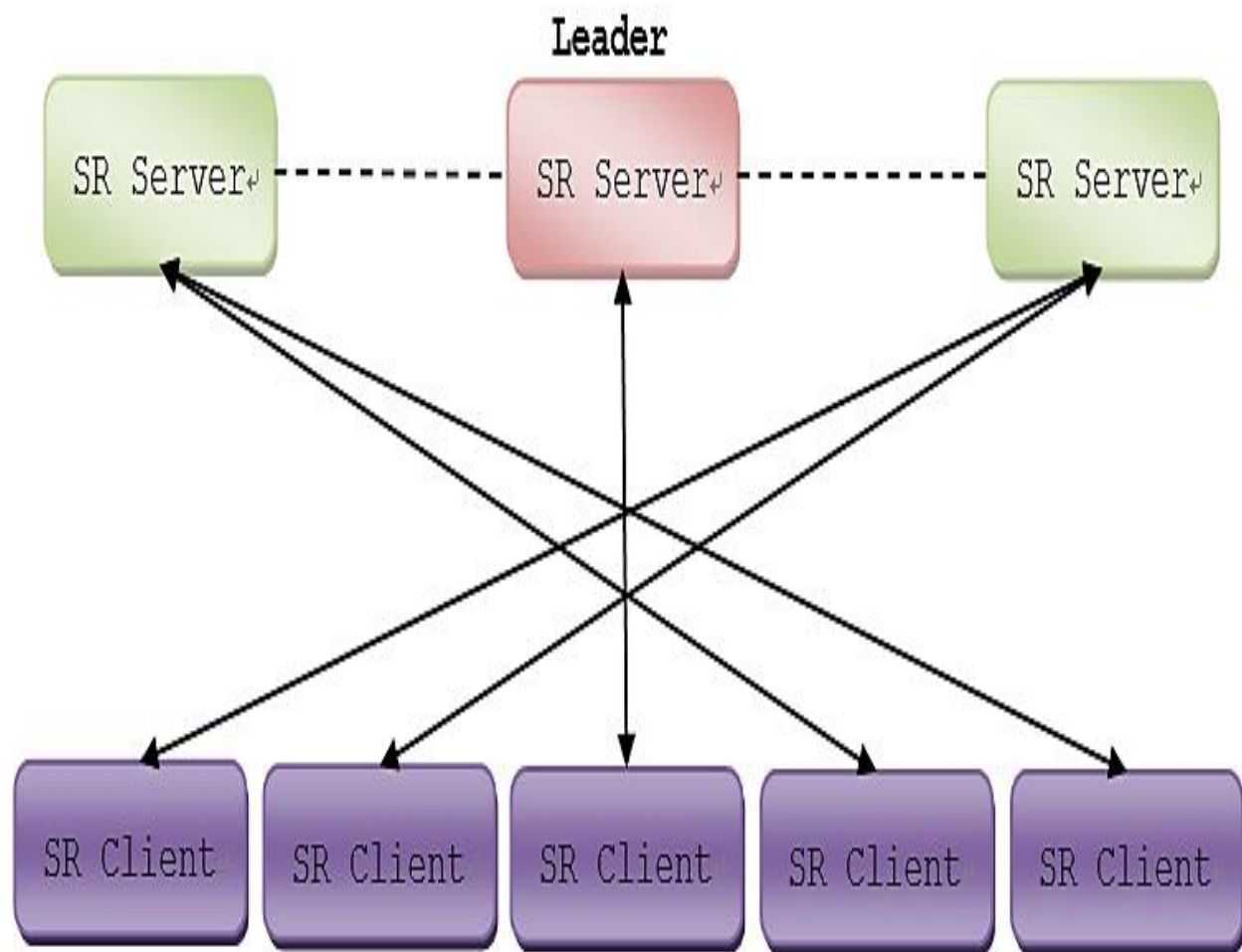


- ✓ 微服务层Host
- ✓ 微服务层Port
- ✓ 数据层Host
- ✓ 数据层Port
- ✓ 配置中心高可用



微服务架构构成

• 注册中心



微服务架构构成

● 配置中心

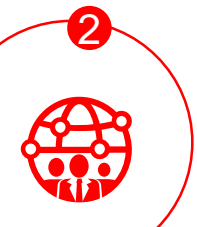
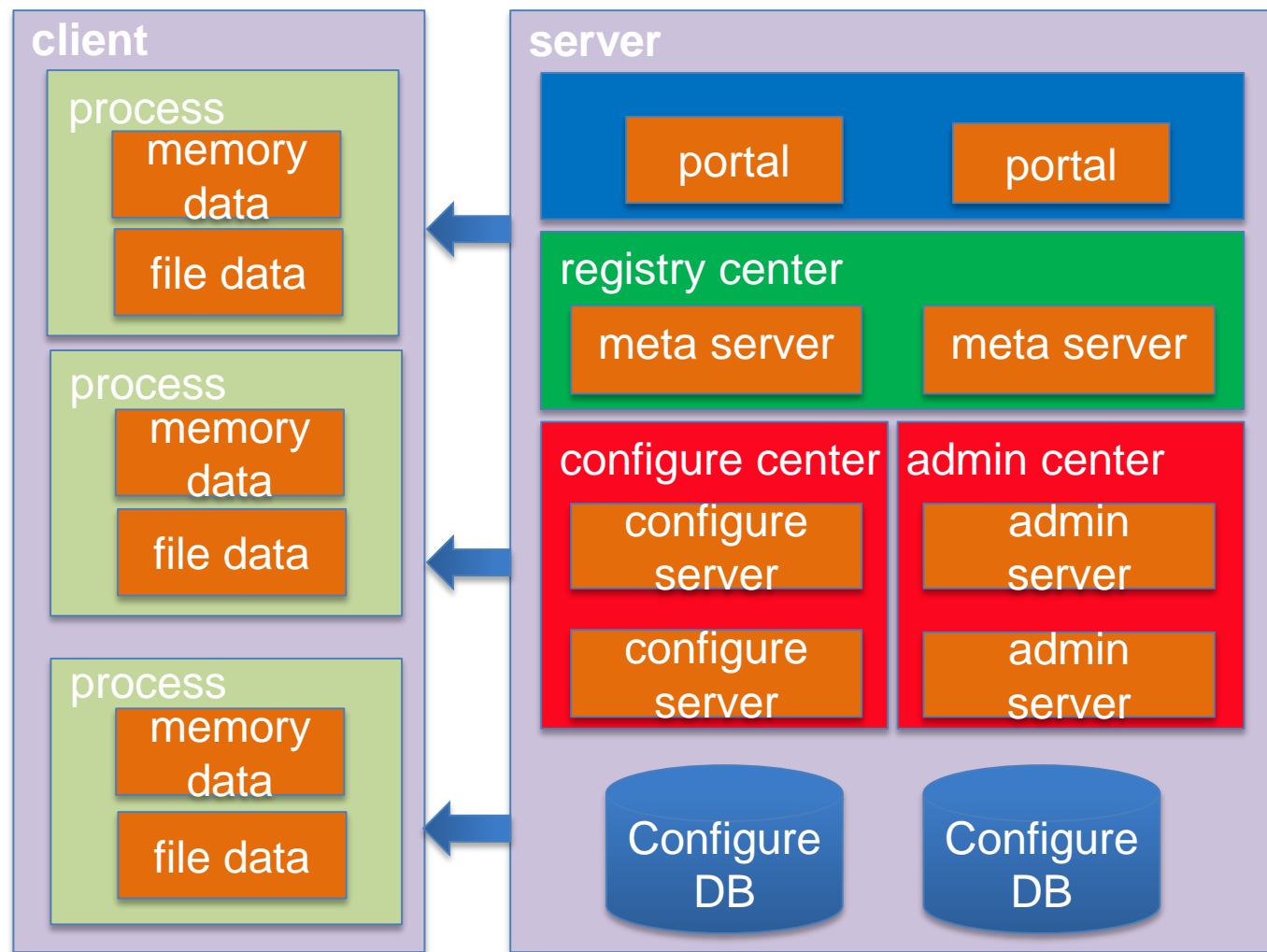


- ✓ 每层服务配置[KV]统一管理
- ✓ 配置更新实时同步
- ✓ 微服务配置项热加载
- ✓ 配置项最终一致性
- ✓ 配置中心高可用、高可靠



微服务架构构成

● 配置中心

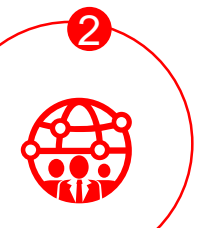


微服务架构构成

● 微服务路由

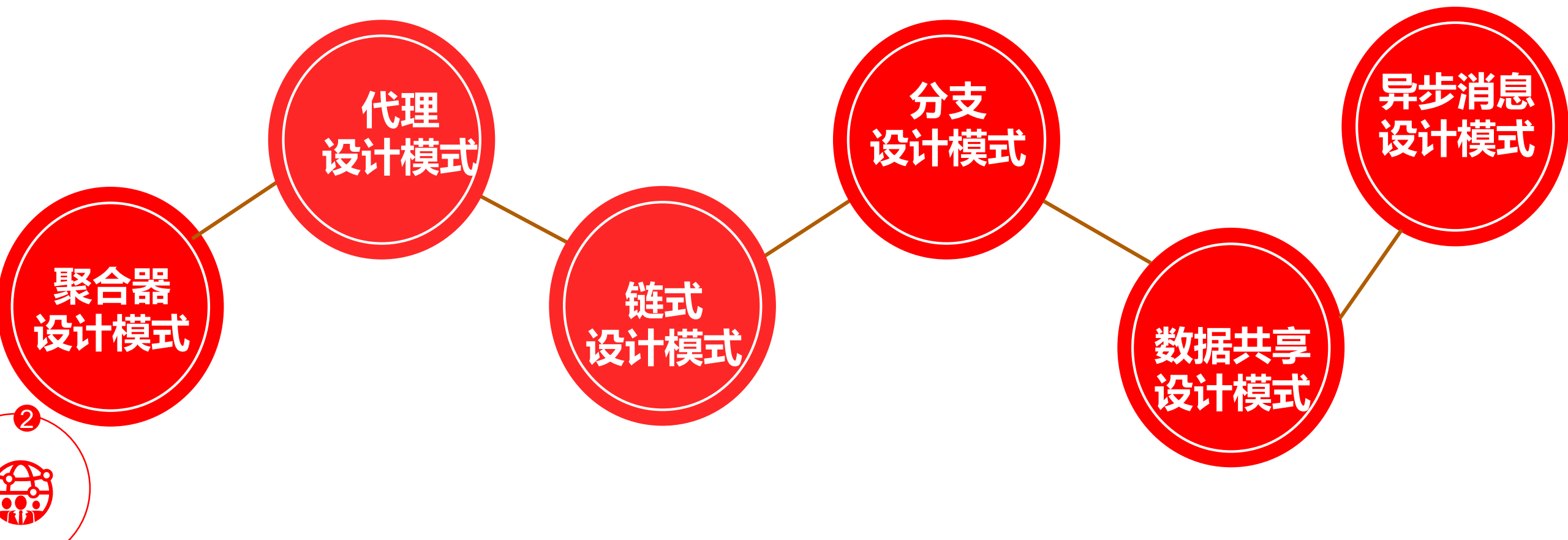


- ✓ 网关层获取微服务层注册信息
- ✓ 网关层请求路由到微服务层
- ✓ 微服务层获取数据层注册信息
- ✓ 微服务层请求路由到数据层



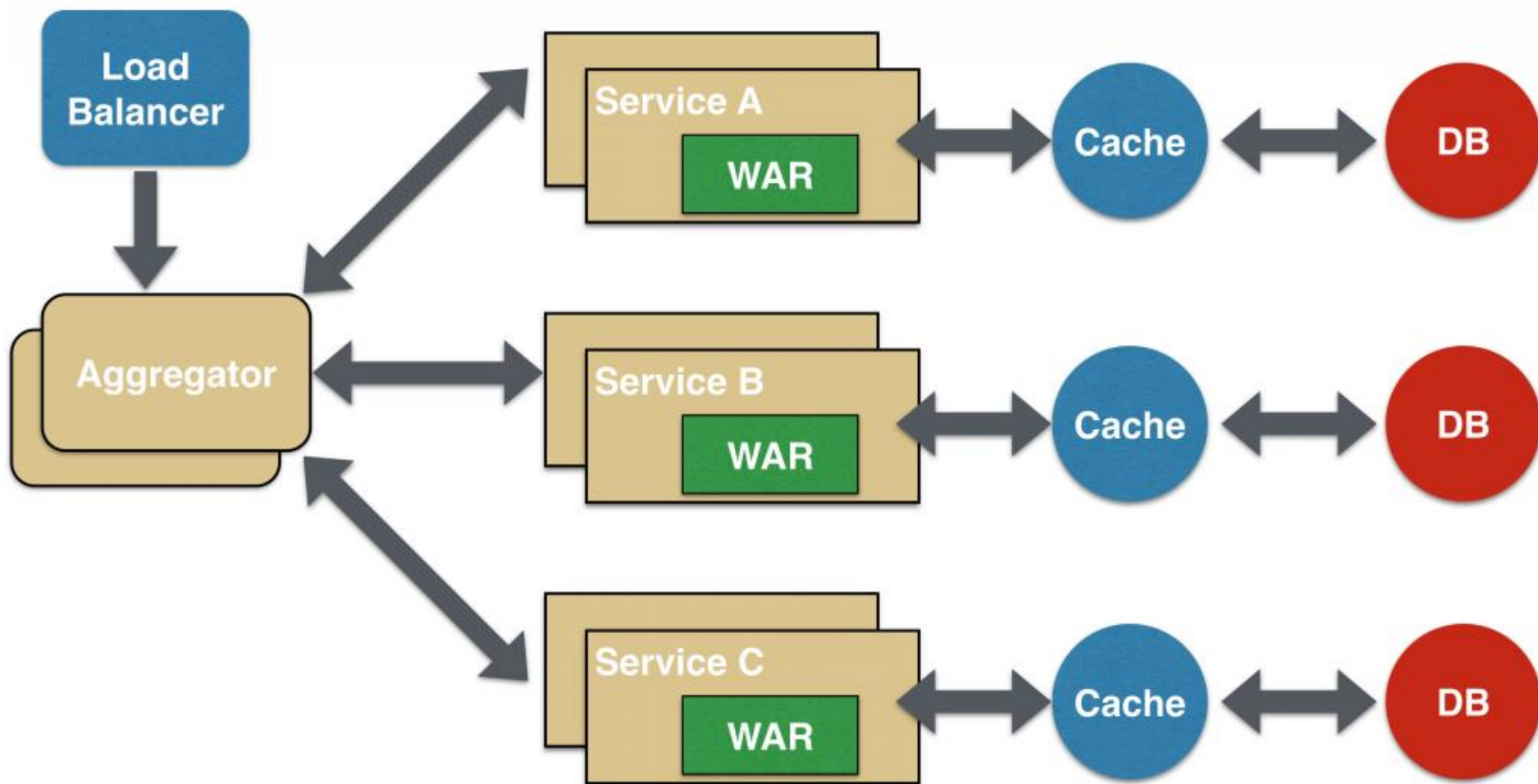
微服务架构设计模式

微服务架构六种设计模式



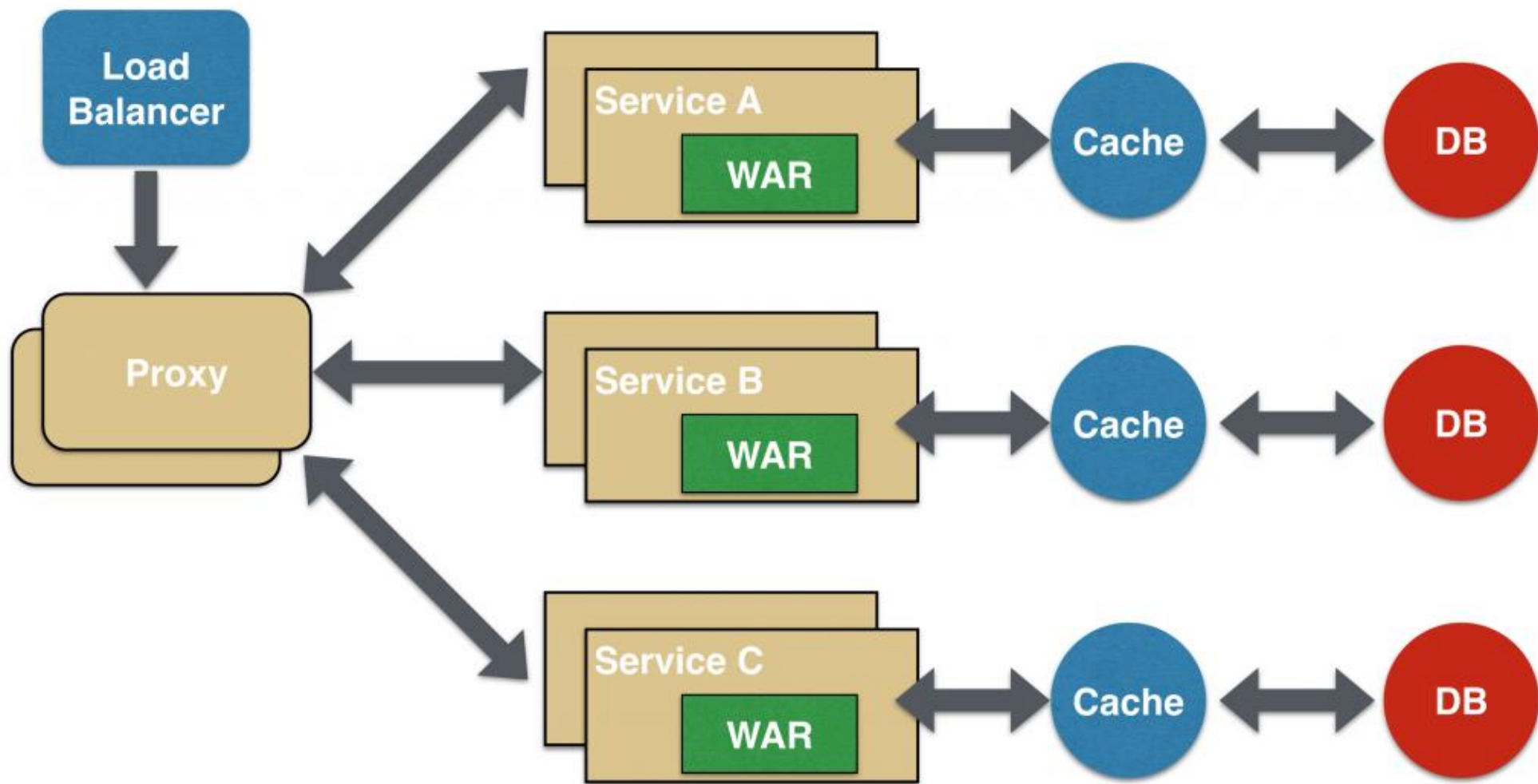
微服务架构设计模式

✓ 聚合器设计模式



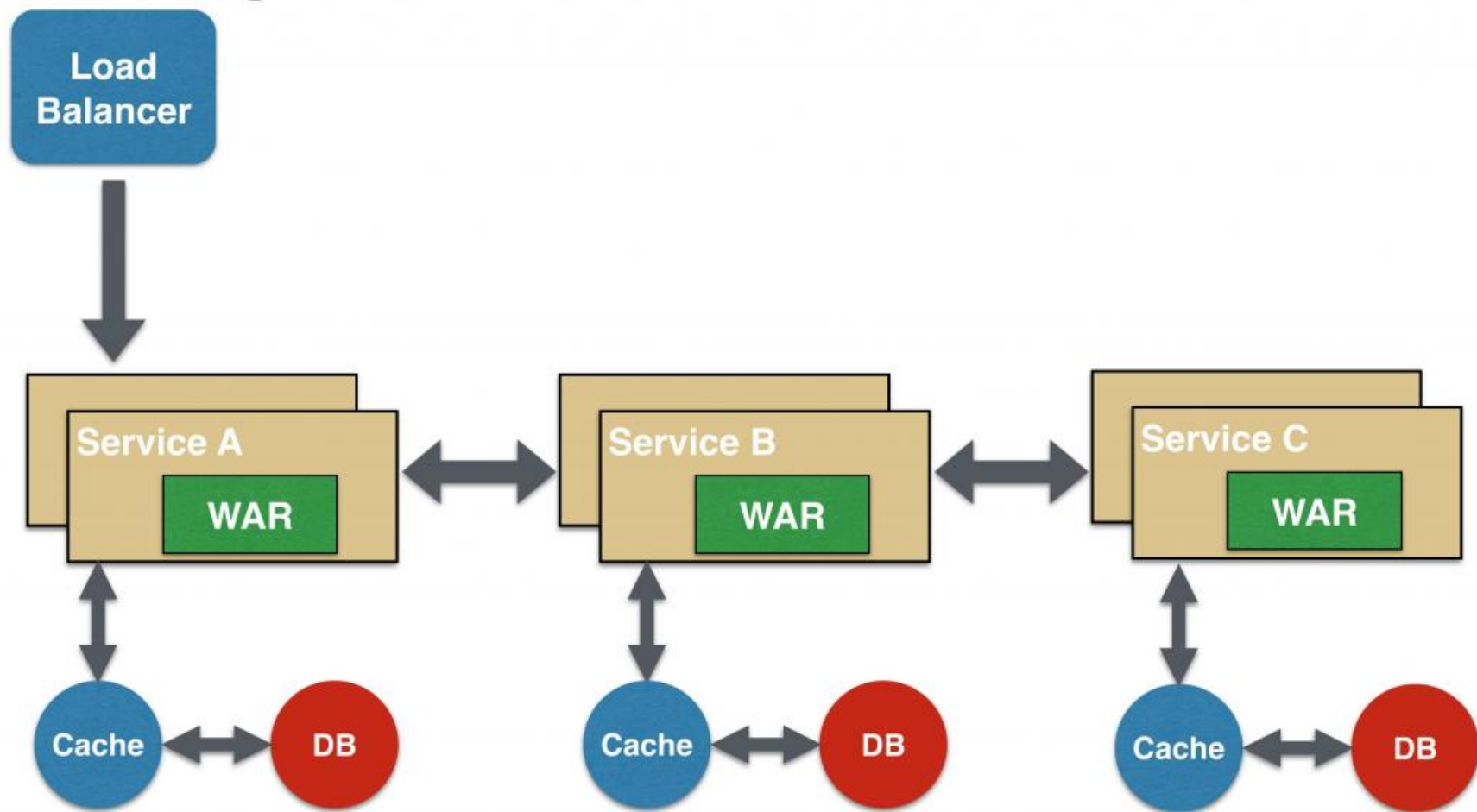
微服务架构设计模式

✓ 代理设计模式



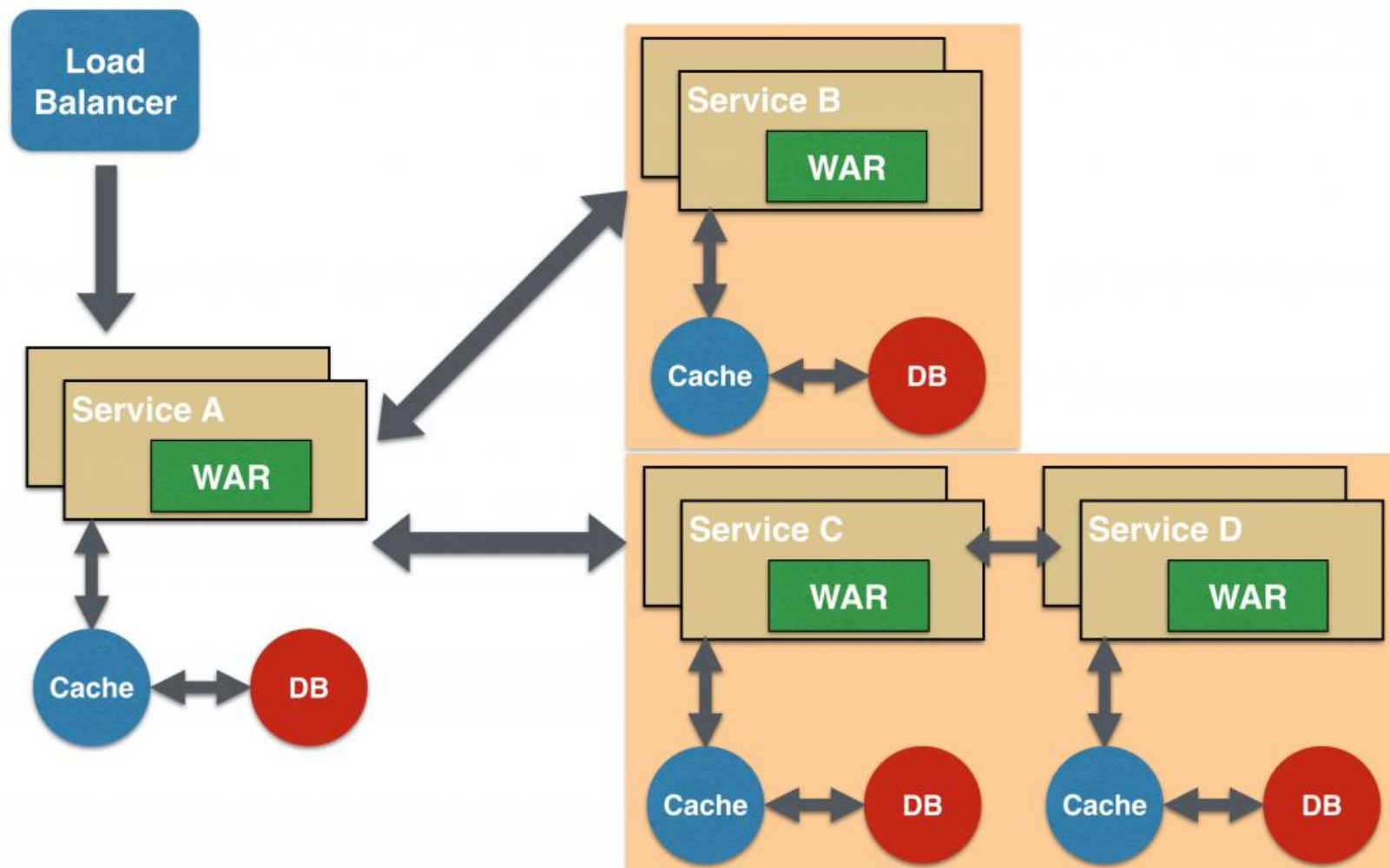
微服务架构设计模式

✓ 链式设计模式



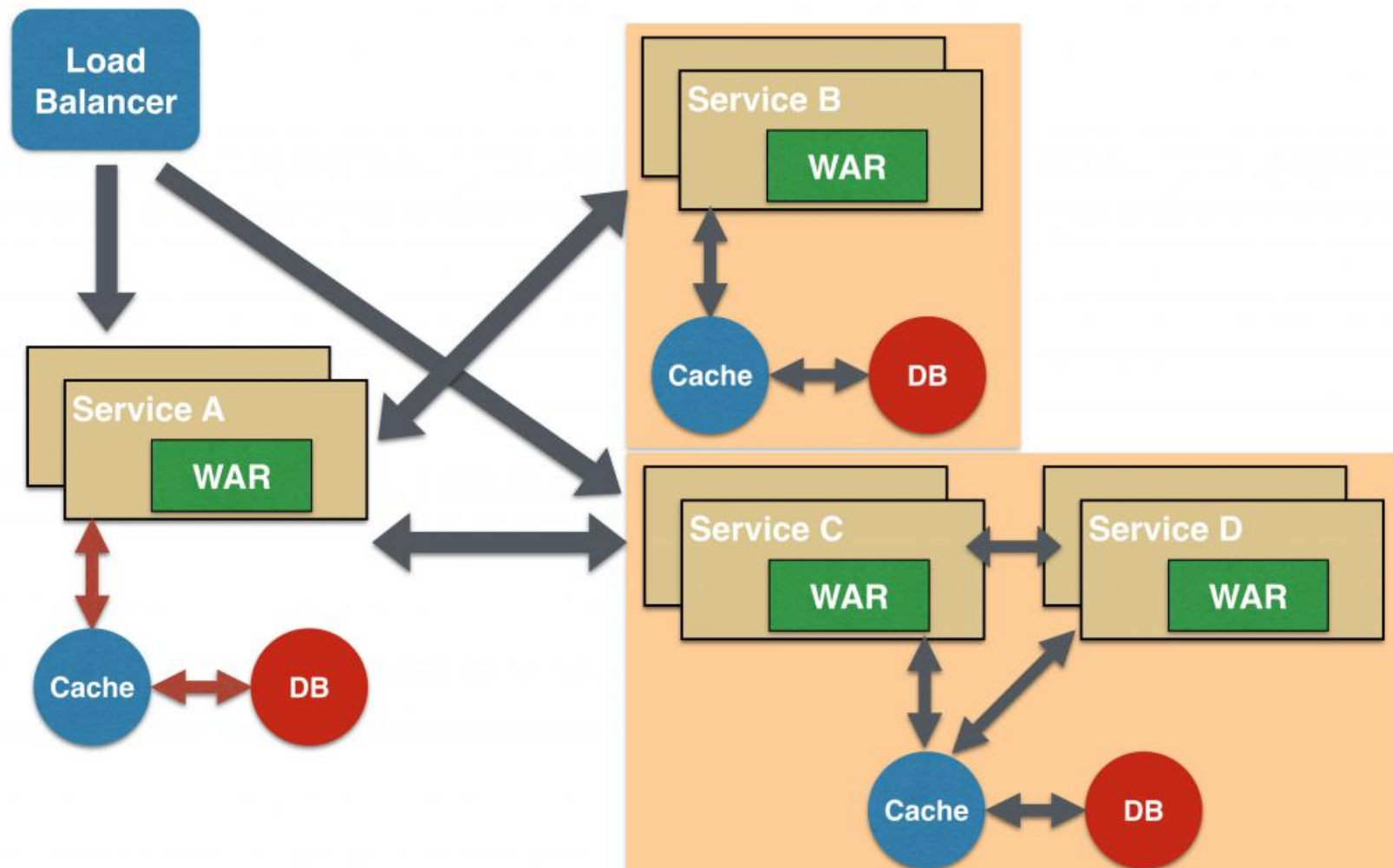
微服务架构设计模式

✓ 分支设计模式



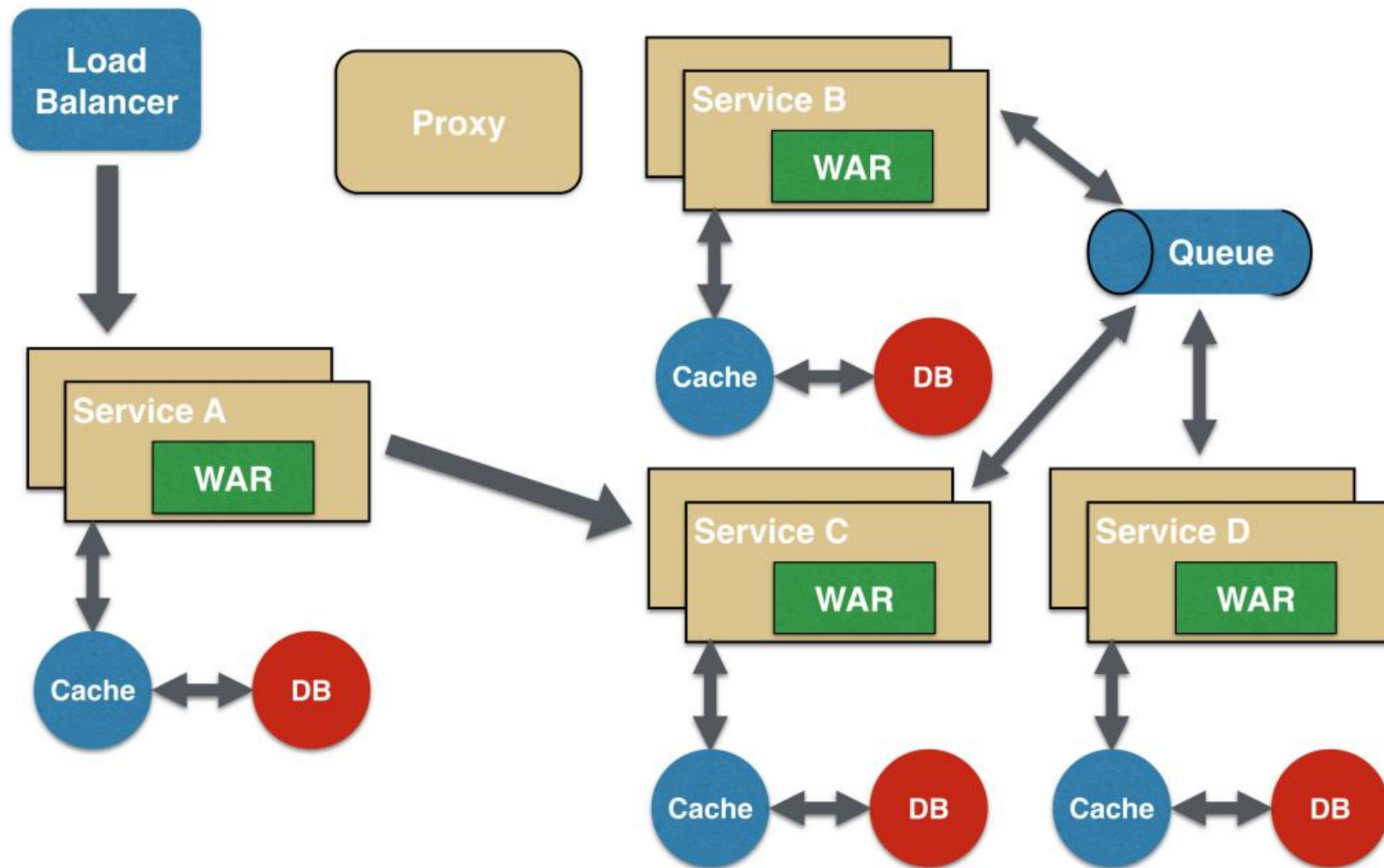
微服务架构设计模式

✓ 数据共享设计模式



微服务架构设计模式

✓ 异步消息设计模式



微服务架构设计模式

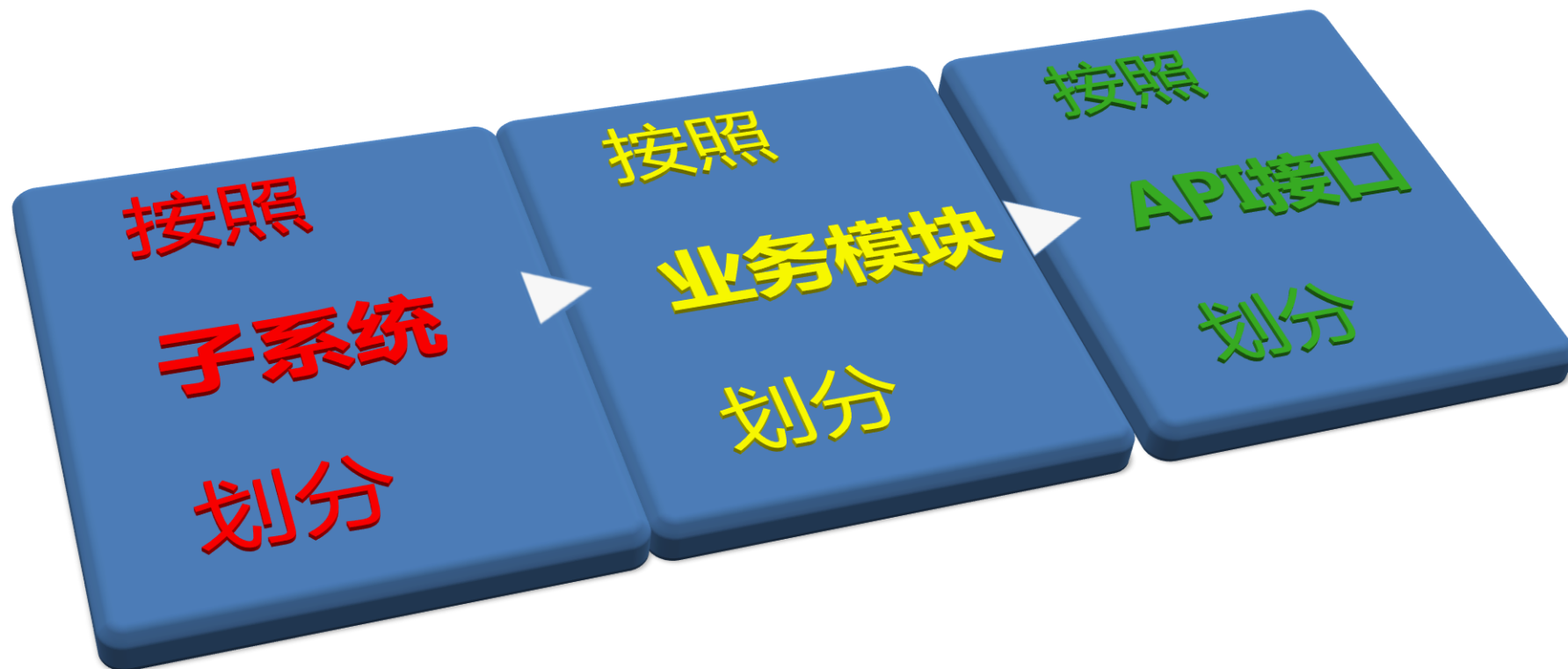
✓ 设计模式

实际项目
最常用

- ✓ 聚合器设计模式
- ✓ 异步消息设计模式
- ✓ 数据共享设计模式



微服务粒度设计



微服务粒度设计



子系统 粒度

- ✓ 粒度粗
- ✓ 类似于SOA
- ✓ 公司多个业务（每个业务一个子系统）



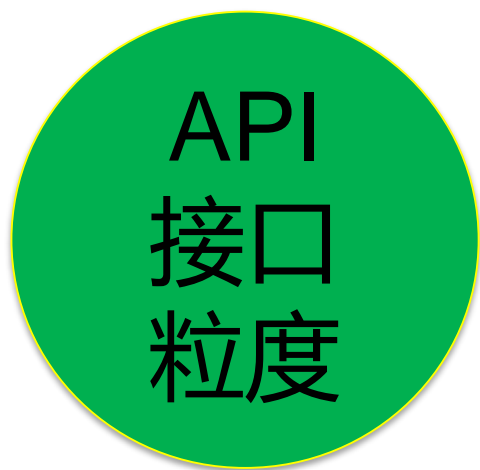
微服务粒度设计

业务
模块
粒度

- ✓ 粒度适中
- ✓ 业务系统按照功能划分
 - （商品、交易、搜索、推荐等）



微服务粒度设计



- ✓ 粒度细
- ✓ 业务模块多个API接口
 - （商品查询API、商品更新API等）



微服务粒度设计

实际
项目
常用

✓ 业务模块粒度

2



实践案例一[聚合器微服务架构设计模式]

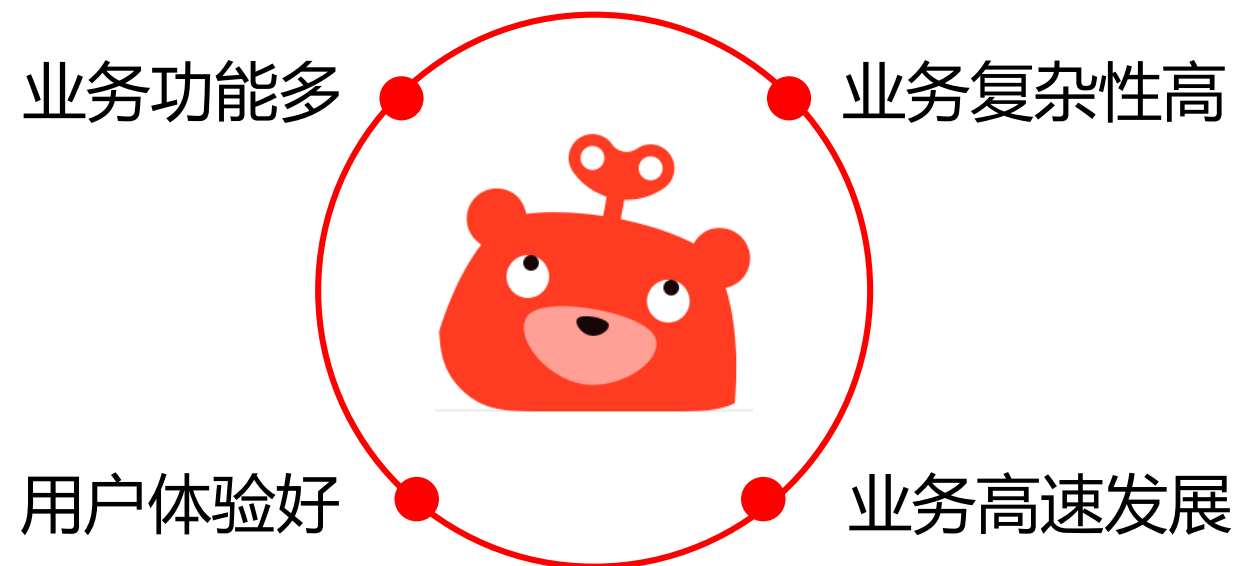


使用原因-二手交易平台功能

- ✓ 发布商品
- ✓ 分类搜索
- ✓ 关键词搜索
- ✓ 推荐商品
- ✓ 消息中心
 - 私信、留言
- ✓ 个人中心等等

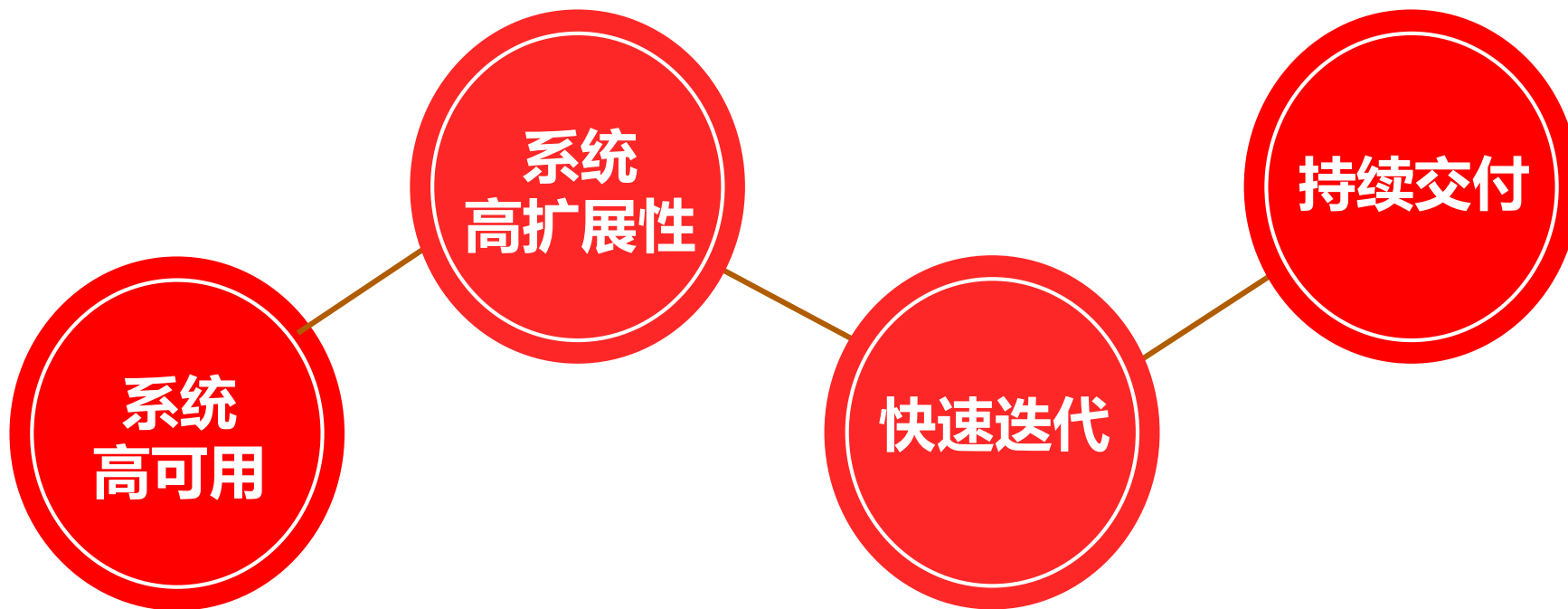


使用原因

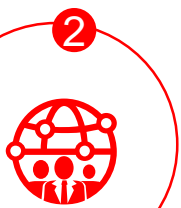


使用原因

微服务架构特点&二手交易平台特点



转转使用微服务架构比较合适



演进-如何设计



整体设计

- ✓ 水平分层
- ✓ 垂直业务拆分



每层设计

- ✓ 微服务-业务单元垂直拆分
- ✓ 无状态化
- ✓ 独立进程、部署、运维

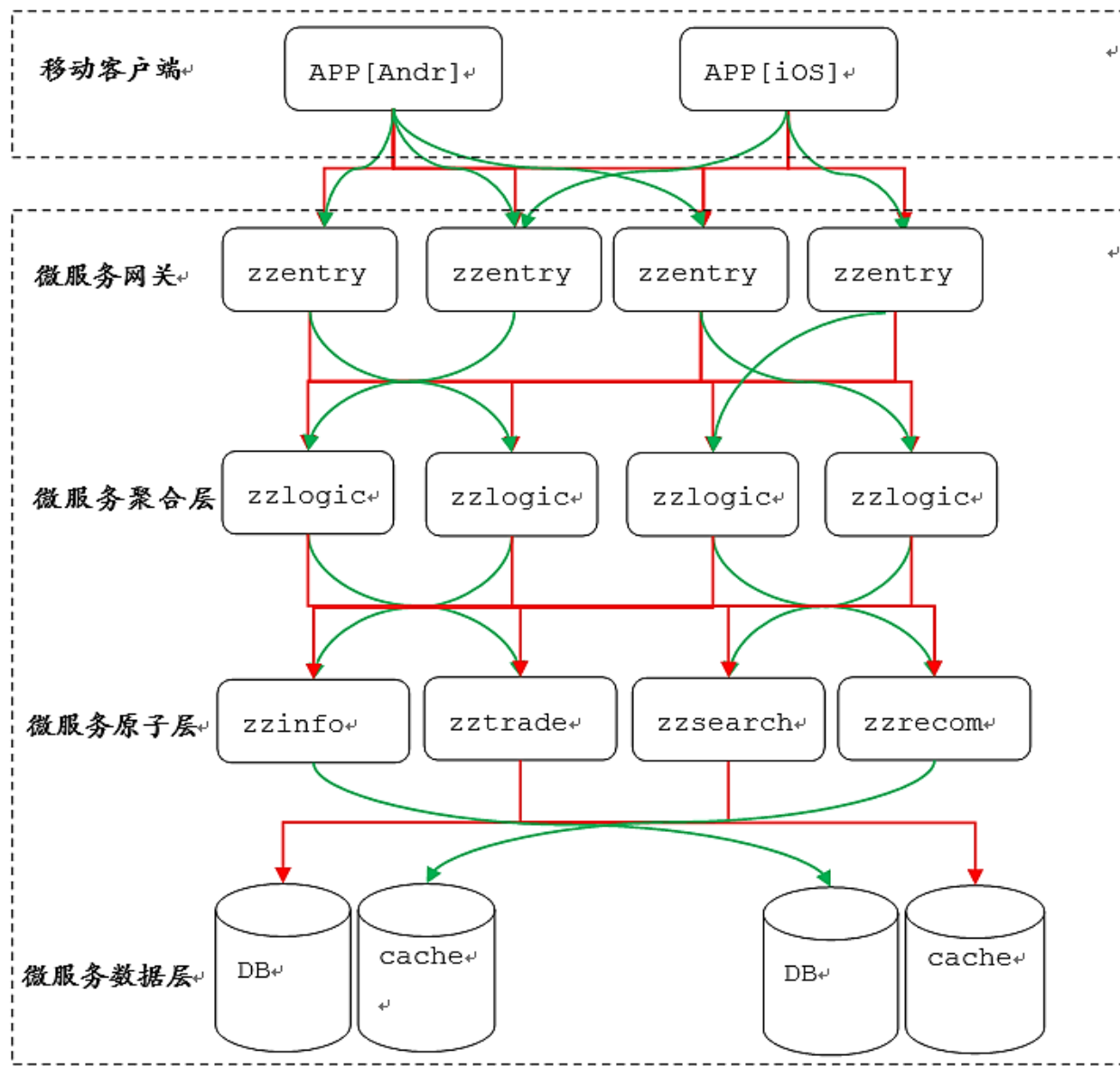


高可用

- ✓ 冗余
- ✓ 自动恢复

演进-总体架构设计

- ✓ 微服务网关
- ✓ 微服务聚合层
- ✓ 微服务原子层
- ✓ 微服务数据层
- ✓ 轻量级通信
 - HTTP
 - RPC
- ✓ 去中心化管理
 - 开发语言-java
- ✓ 微服务注册
- ✓ 微服务发现



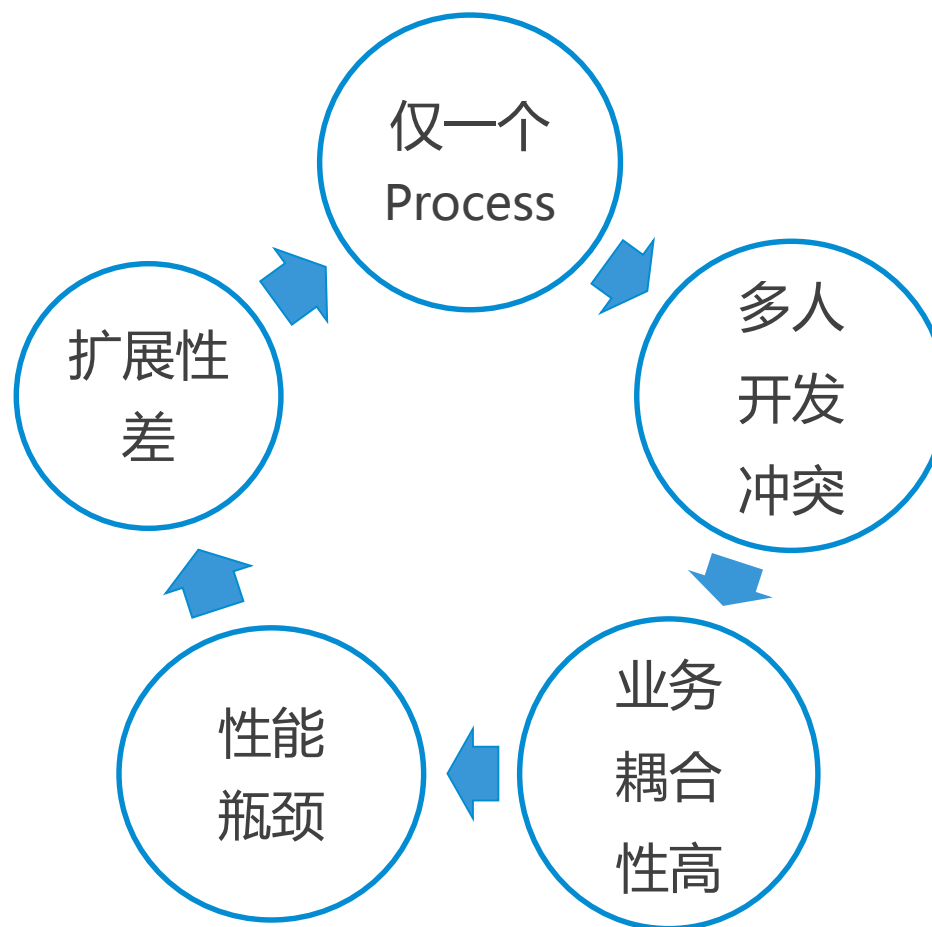
演进-存在问题

二手交易特点

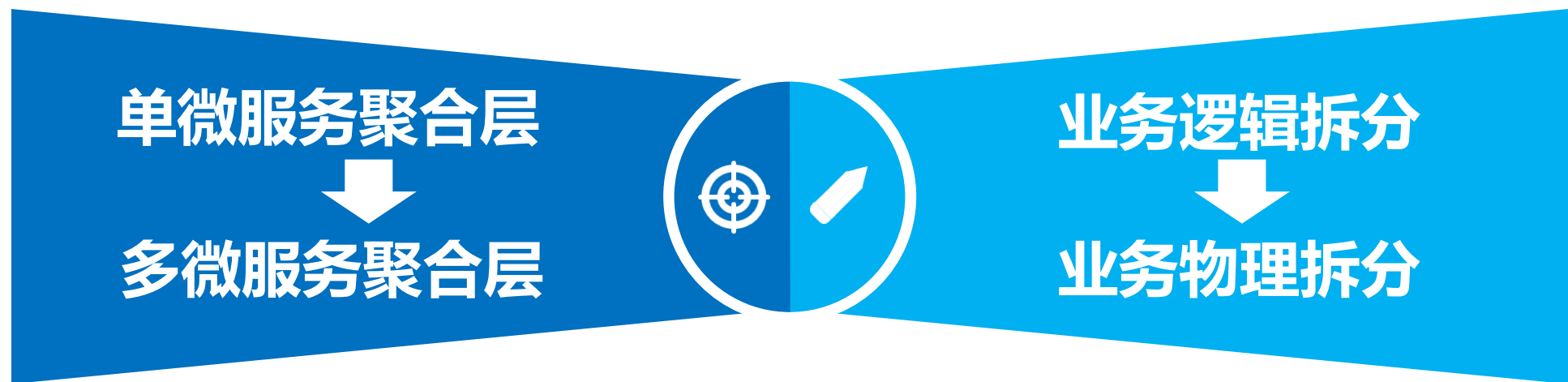
- ✓ 业务发展快
- ✓ 业务越来越复杂

微服务聚合层

- ✓ 开发瓶颈集中
-微服务聚合层



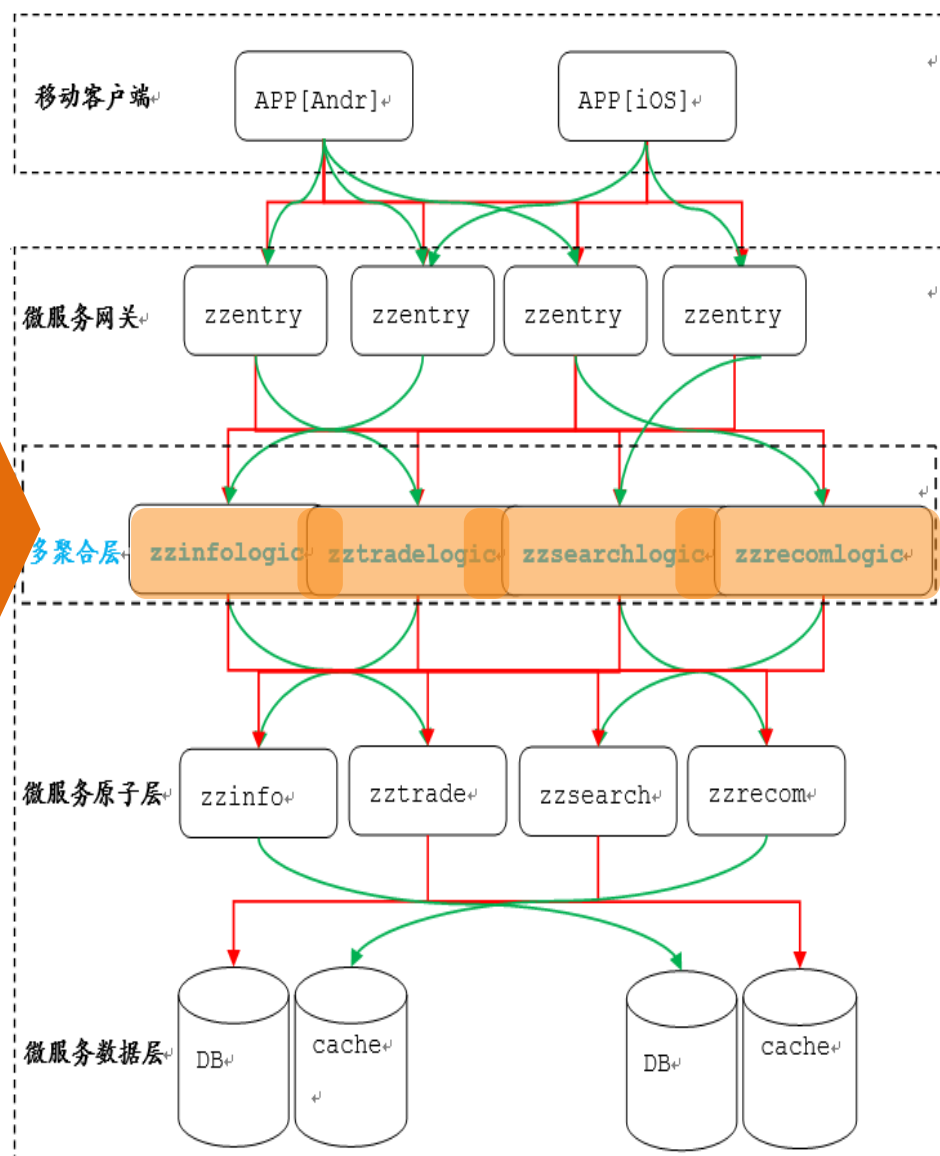
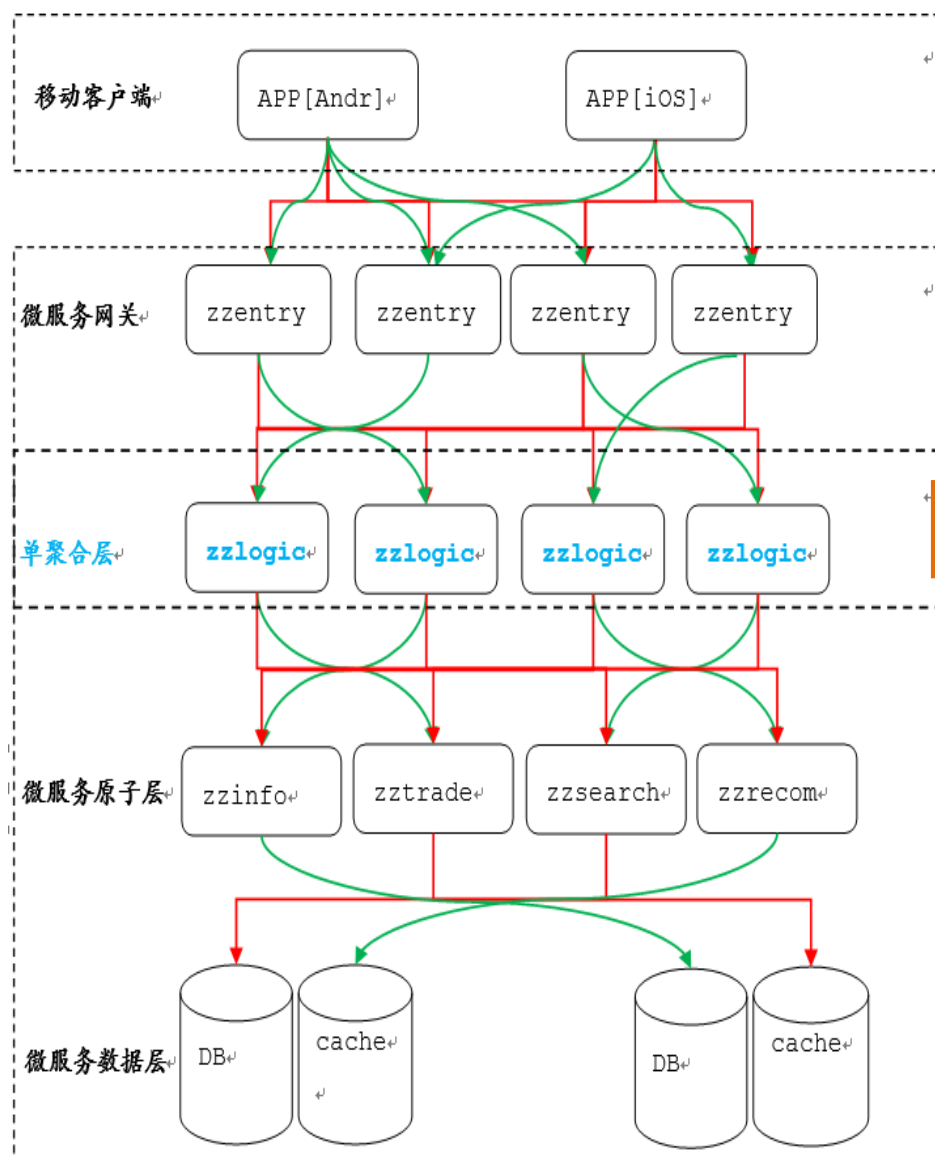
演进



转转业务功能领域模型拆分



演进



演进-多微服务聚合层优点

独立

- ✓ 进程
- ✓ 开发
- ✓ 部署
- ✓ 运维

高效

- ✓ 快速迭代
- ✓ 持续交付



实践案例二[异步消息微服务架构设计模式]

✓ 百度Feed系统



The screenshot displays the Baidu Feed system interface. On the left is a user profile for 'musicmi' with a '修改个人资料' (Edit Profile) link and a sidebar menu containing: 文章 (Articles), 相册 (Albums), HOHO, 分享 (Share), 投票 (Vote), 测试 (Test), 礼物 (Gifts), 宠物 (Pets), and 应用大厅 (App Hall). The main content area shows a post input section with a date/time stamp '2012年03月02日: “又到福来day, 各位周末愉快~”', a text box containing '2012啦! 新年快乐!', and a '发布' (Post) button. Below this is a '好友动态' (Friends' Activity) section with tabs for '全部' (All), '文章' (Articles), '相册' (Albums), '分享' (Share), and 'HOHO'. The feed lists three posts: 1) 'solaryf' posted an '文章' (Article) titled '下午的时光' (Afternoon Time) 1 hour ago, with content about sitting on a balcony and reading books. 2) '一框知天下在 通辽吧' posted an '公告' (Announcement) about deleting posts 6 hours ago. 3) '一框知天下在 贴吧合并吧' posted an '申请合并' (Merge Application) 7 hours ago. Each post includes interaction links like '阅读全文' (Read Full Text), '分享' (Share), and '评论' (Comment).



实践案例二[异步消息微服务架构设计模式]

✓ 百度Feed系统

百度
Feed
系统

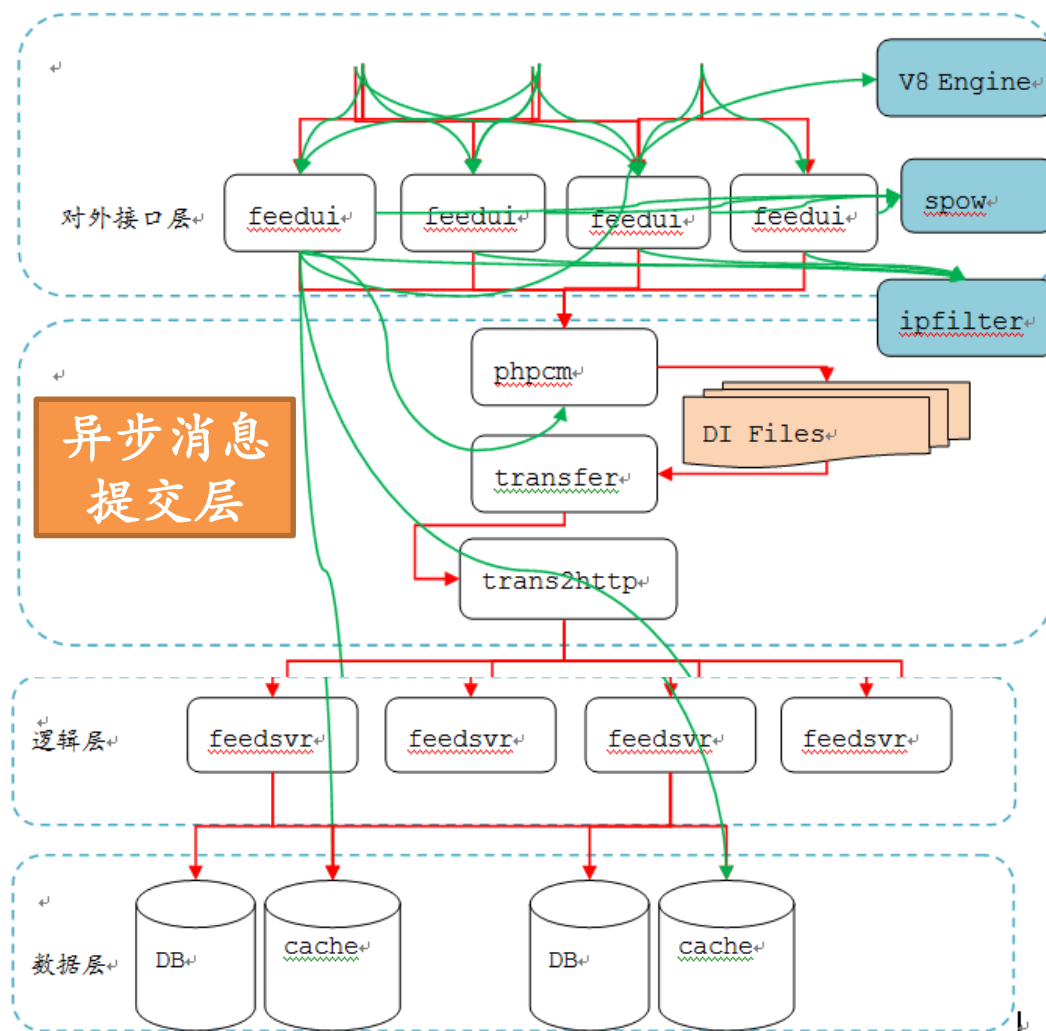
- ✓ 用户发feed
- ✓ 获取好友的feed
- ✓ 聚合好友的feed
- ✓ 按照feed发布时间倒序展现
- ✓ Push or Pull



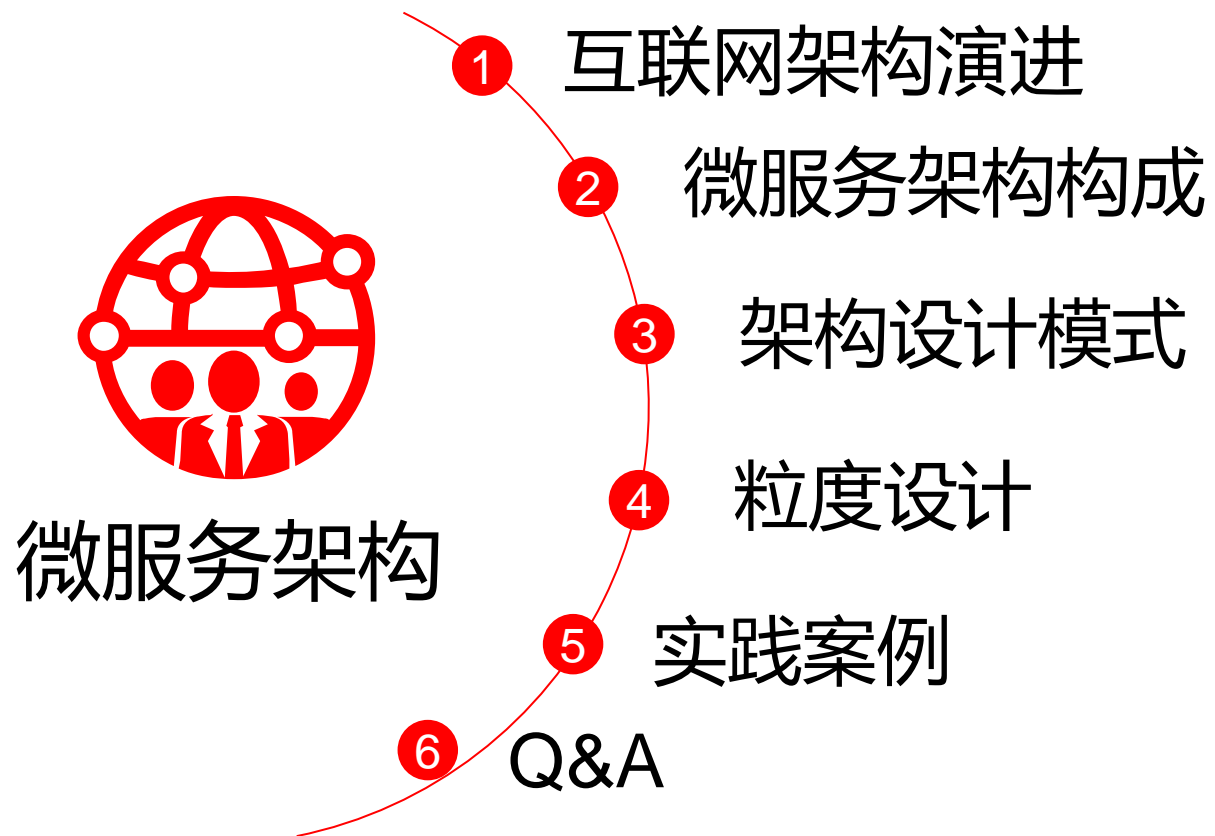
实践案例二[异步消息微服务架构设计模式]

✓ 百度Feed系统微服务架构

- 微服务网关层
- 异步消息提交层
- 微服务业务处理层
- 微服务数据层



要点回顾



欢迎关注本人公众号 “架构之美”



Thanks!

让生活更简单

