互联网大规模高可用高并发微服务架构构建之道

The Microservices Architecture

作者：孙玄

00 个人简介

- 转转公司首席架构师
- 转转公司技术委员主席
- 前58集团技术委员主席
- 前58集团高级系统架构师
- 前百度资深研发工程师
- 毕业于浙江大学
- 擅长架构设计、大数据、机器学习等技术领域
- 架构之美〔beautyArch〕公众号作者
- 多次代表公司对外分享交流

00 对外技术交流

QCon 全球软件开发大会

DTCC 2016中国数据库技术大会 DATABASE TECHNOLOGY CONFERENCE CHINA 2016

Artificial Intelligence CONFERENCE

SDCC 中国软件开发者大会 Software Developer Conference China

TOP1 技术型企业案例研究智库

全球互联网技术大会 GLOBAL INTERNET TECHNOLOGY CONFERENCE

WOT World Of Tech

SACC

ArchSummit 全球架构师峰会

PROGRAMMER 程序员

A2M

目录

01

微服务拆分之道

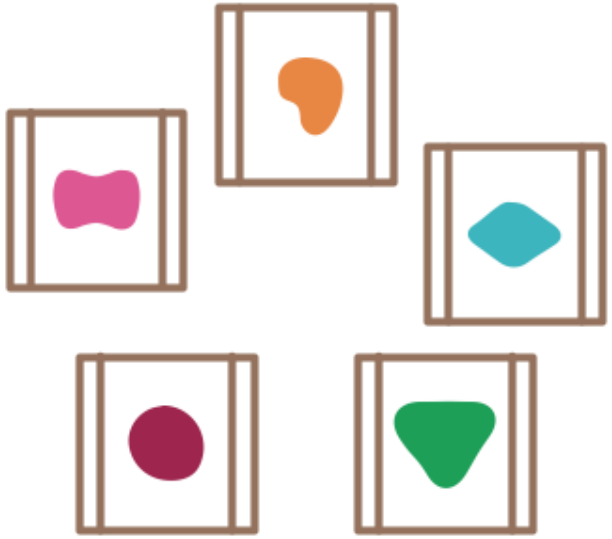
How to split Microservices

01微服务拆分之道

Microservices

common characteristics of this architectural style

by James Lewis and Martin Fowler



In short, the microservice architectural style is an approach to developing a single application as a **suite of small services**, each **running in its own process** and communicating with lightweight mechanisms, often an HTTP resource API. These services are **built around business capabilities** and **independently deployable** by fully automated deployment machinery. There is a **bare minimum of centralized management** of these services, which may be written in different programming languages and use different data storage technologies.

-- James Lewis and Martin Fowler

01微服务拆分之道

1.业务时间 事前 | 事中 | 事后

2.营 | 1.下单 扣减库存 销售 | 回复/结算

2.基于业务模型(库存，积分，物流 ——>

01微服务拆分之道

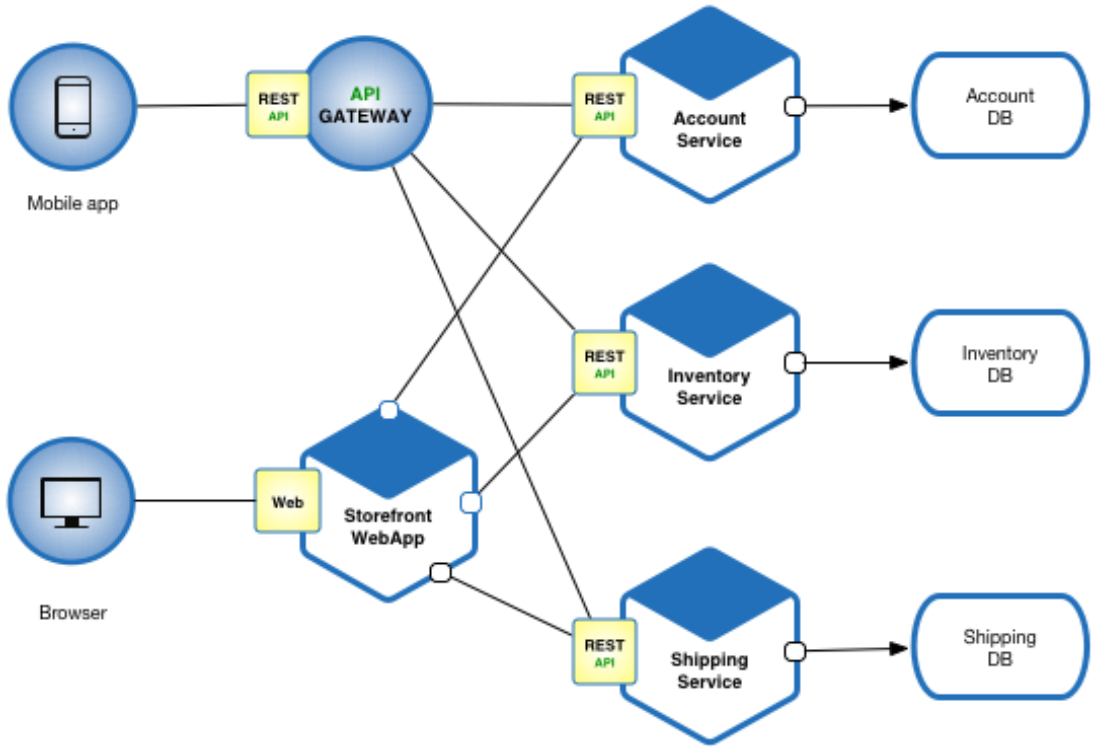
01微服务拆分之道

案例

- e-commerce application.
- It takes orders from customers, verifies inventory and available credit, and ships them.

微服务架构

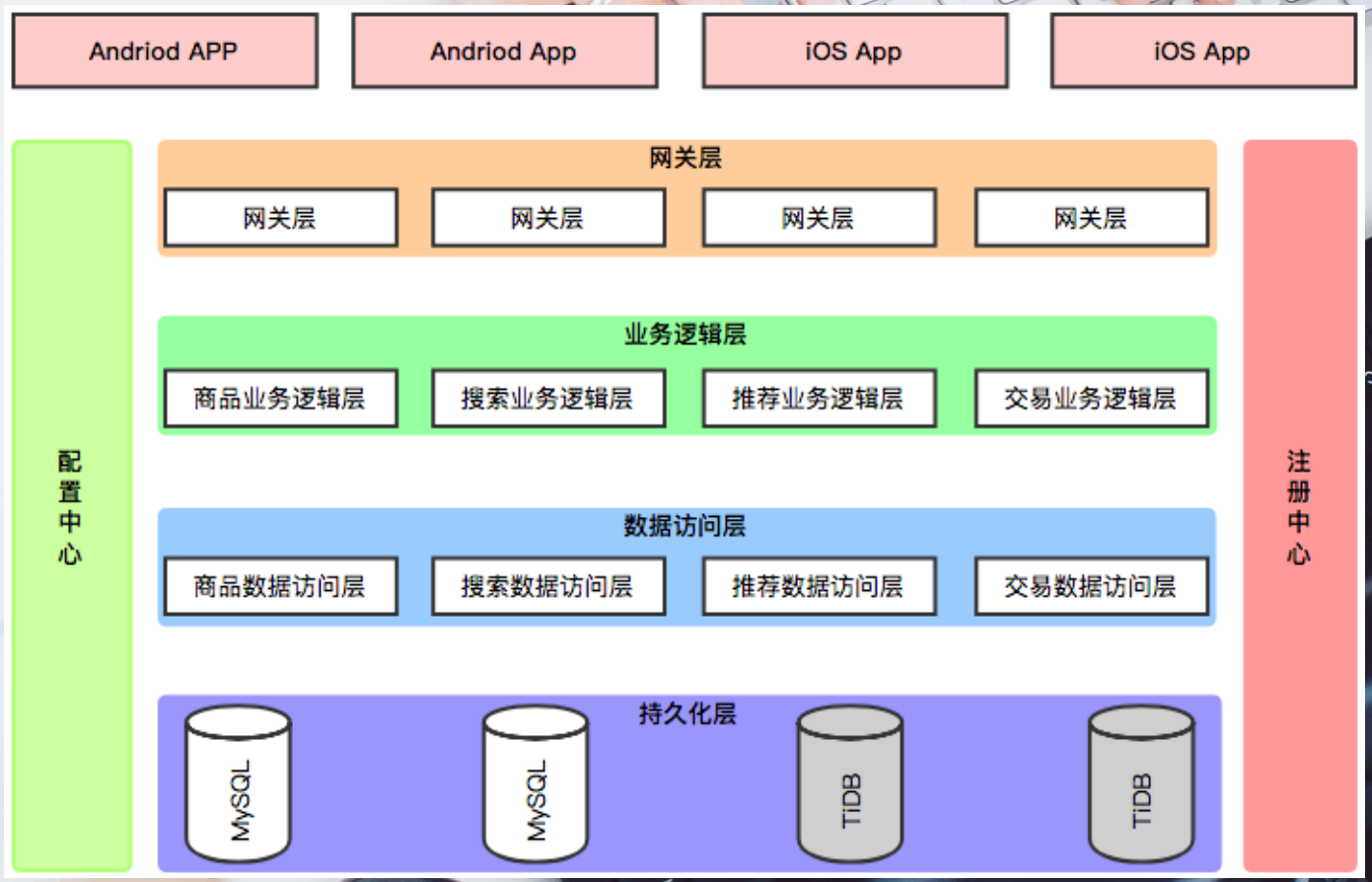
- The application consists of a set of services.



01微服务拆分之道

二手交易平台微服务架构

- 网关层
  - 1个
- 业务逻辑层
  - 多个
- 数据访问层
  - 多个
- DB / Cache
  - 多个



01微服务拆分之道

本质

- 2个维度拆分
- 业务架构
- 组织架构

02

微服务架构应用场景

Application of Microservice architecture

02微服务架构应用场景

目的

- 项目快速迭代
- 项目持续交付

02微服务架构应用场景

适用场景

- 需求层面
- 性能层面
- 数据一致性层面

03

微服务常用典型技术架构深度剖析

Microservices typical architecture mode

03微服务常用典型技术架构深度剖析

典型技术架构

链式
架构

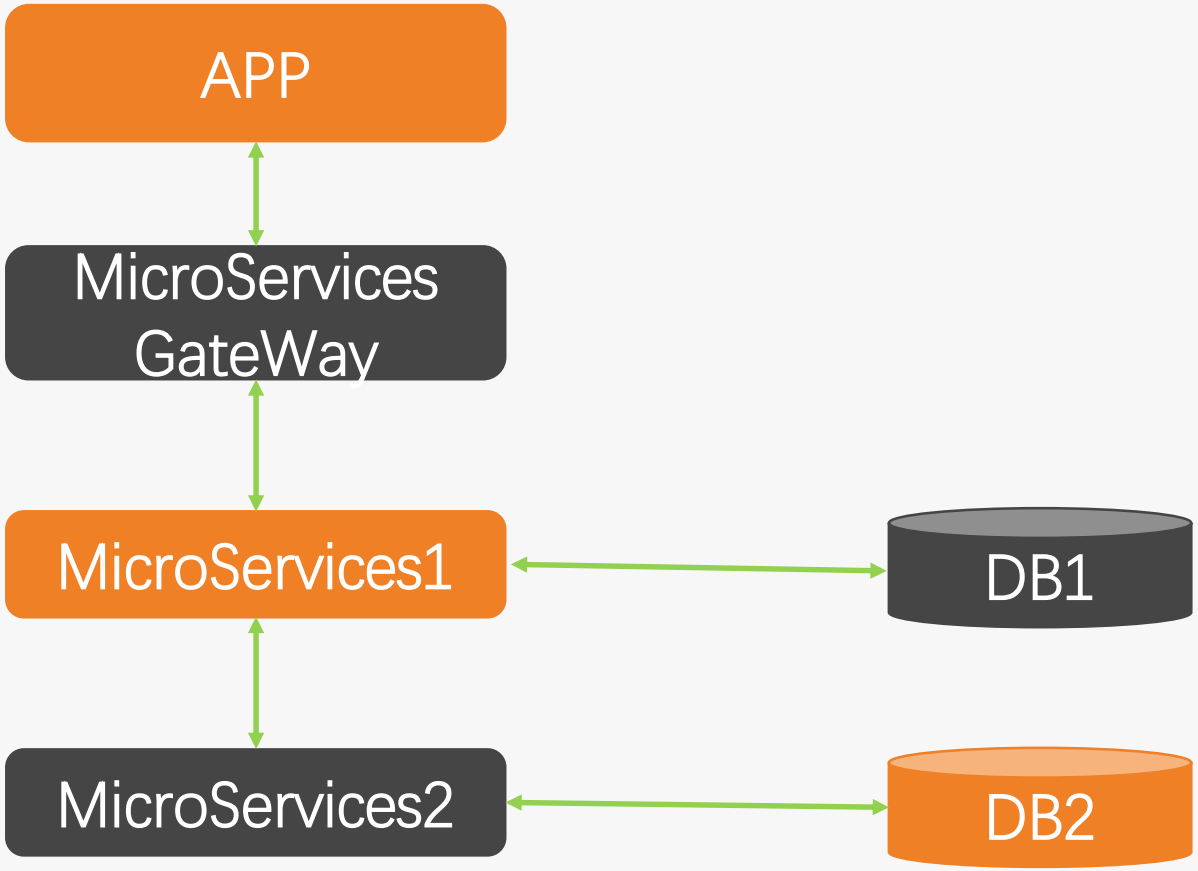
聚合器
架构

数据共享
架构

异步消息
架构

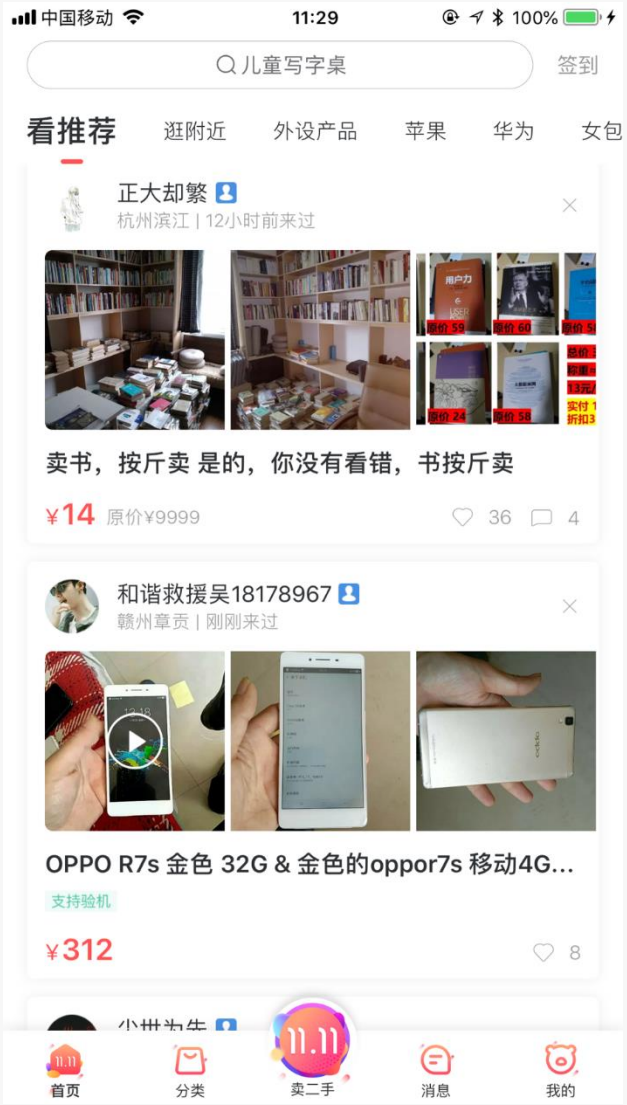
03微服务常用典型技术架构深度剖析

链式架构

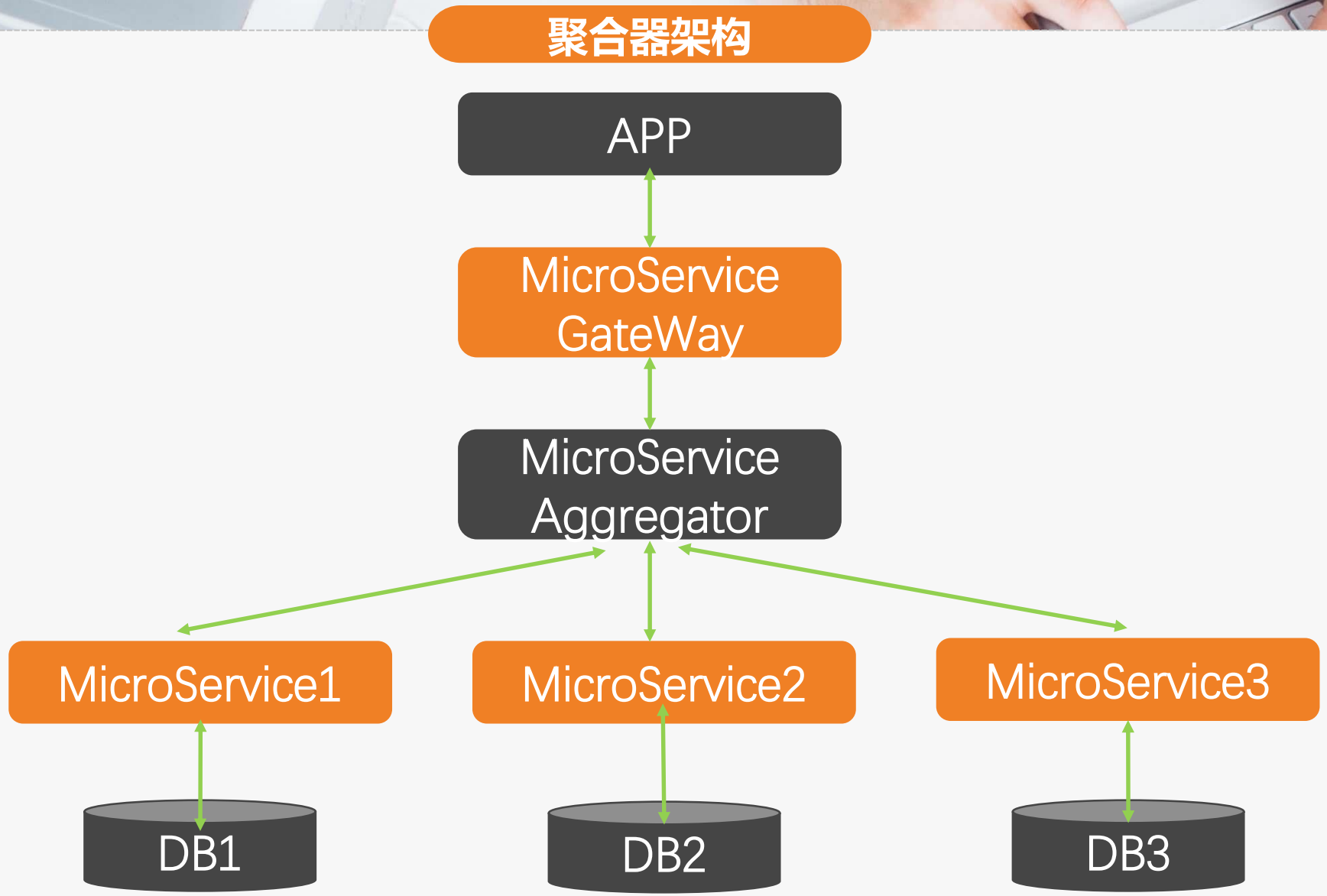


03微服务常用典型技术架构深度剖析

聚合器架构

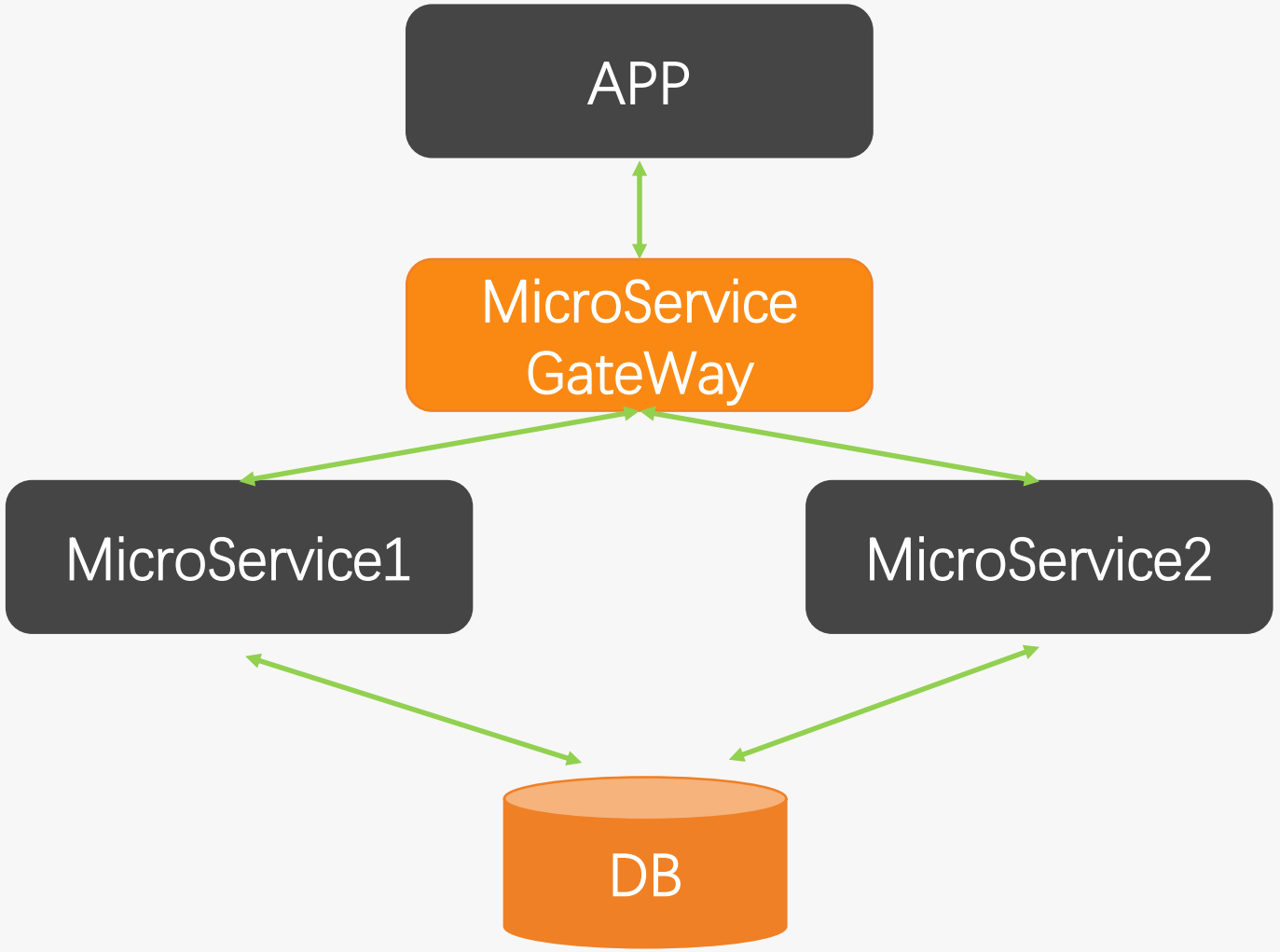


03微服务常用典型技术架构深度剖析

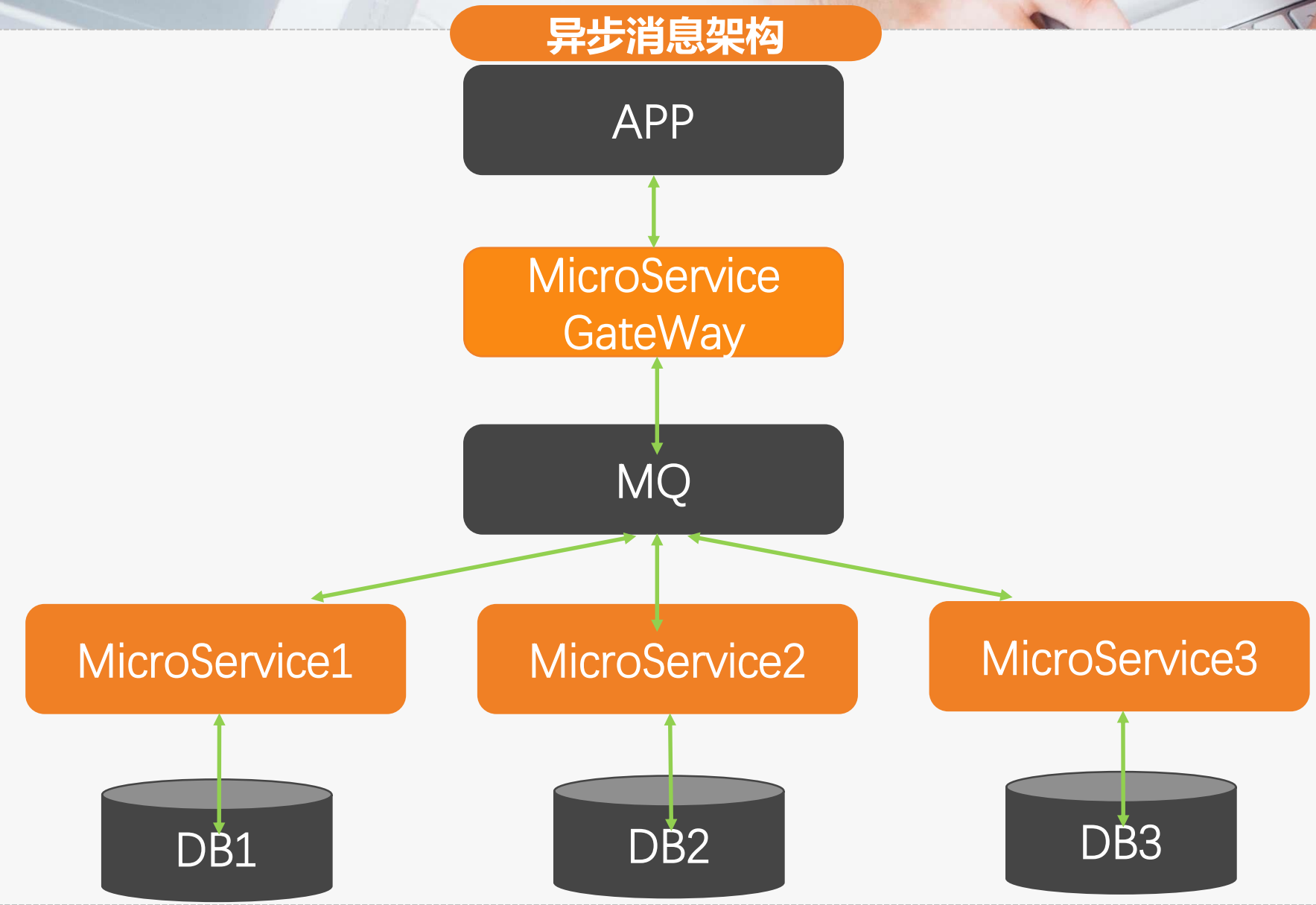


03微服务常用典型技术架构深度剖析

数据共享架构



03微服务常用典型技术架构深度剖析



微服务架构设计模式

实际项目使用

聚合器架构设计

异步消息架构设计

数据共享架构设计

04

微服务架构互联网案例演进

Real case of Microservices

04-1

微服务架构1.0

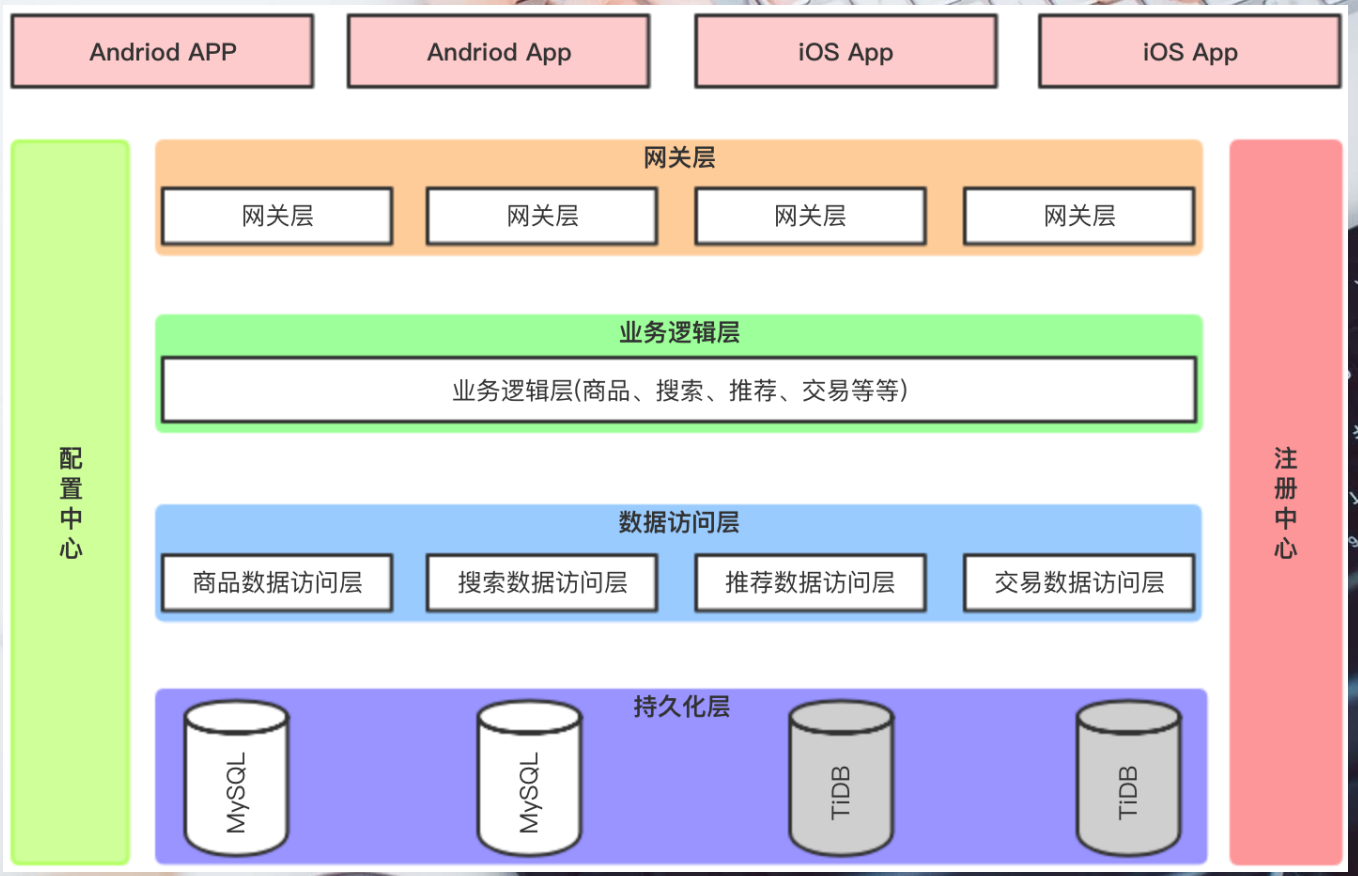
The Version One Of Microservices architecture

04-1微服务架构1.0

04-1微服务架构1.0

微服务架构1.0

- 网关层
  - 1个
- 业务逻辑层
  - 1个
- 数据访问层
  - 多个
- DB / Cache
  - 多个



04-1 微服务架构1.0

微服务架构1.0

- 存在问题
  - 业务逻辑层
    - 粒度粗
      - 所有业务逻辑耦合在一个物理进程内部
    - 迭代效率低下

怎么办?

垂直方向拆

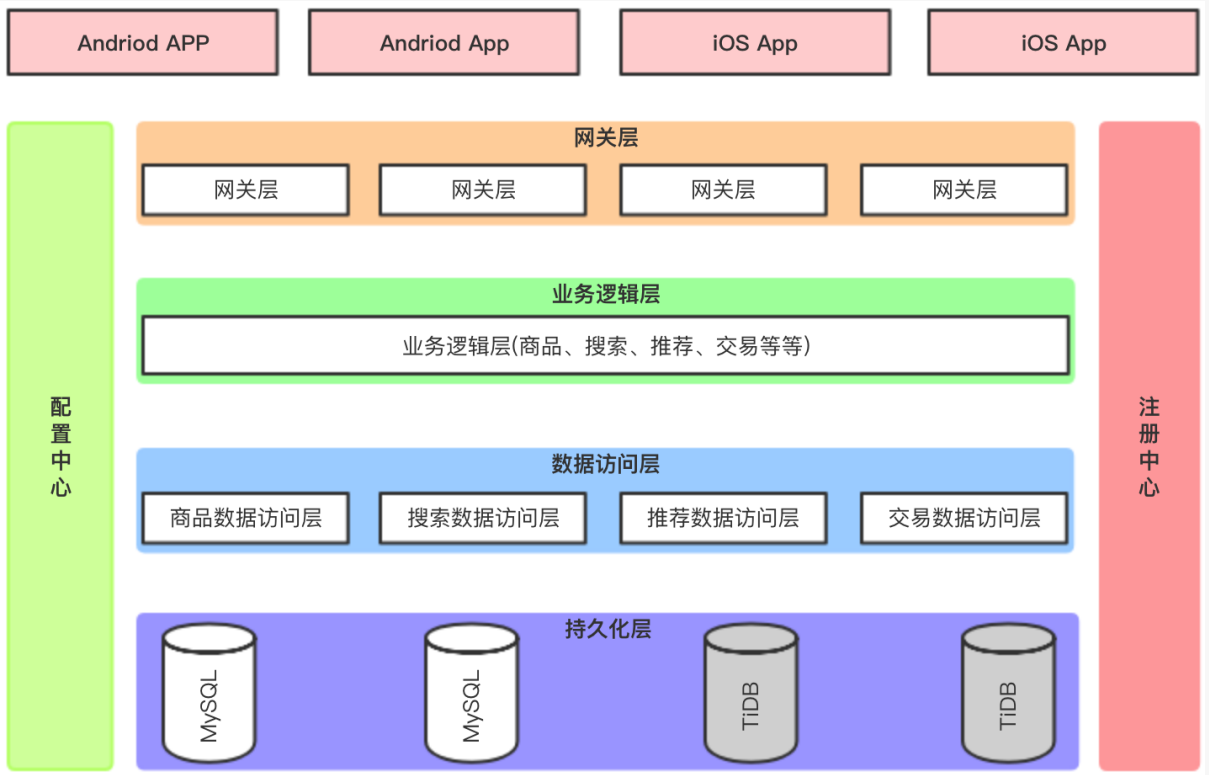
04-2

微服务架构2.0

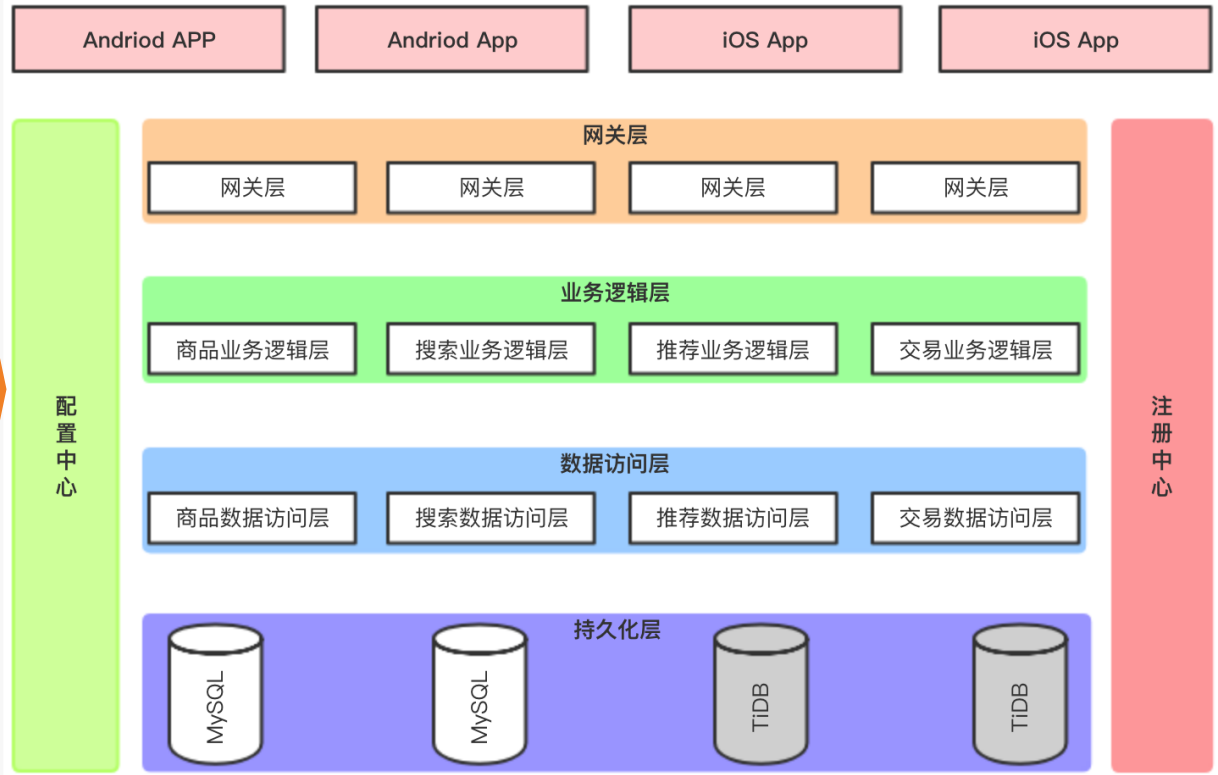
The Version Two Of Microservices architecture

04-2 微服务架构2.0

微服务架构1.0



微服务架构2.0



04-2 微服务架构2.0

微服务架构2.0

- 存在问题
  - 公共逻辑层
    - 组件化
      - 引用Jar包
    - 服务化
      - 下沉为独立服务，提供兼容接口

怎么办？

继续水平方向拆

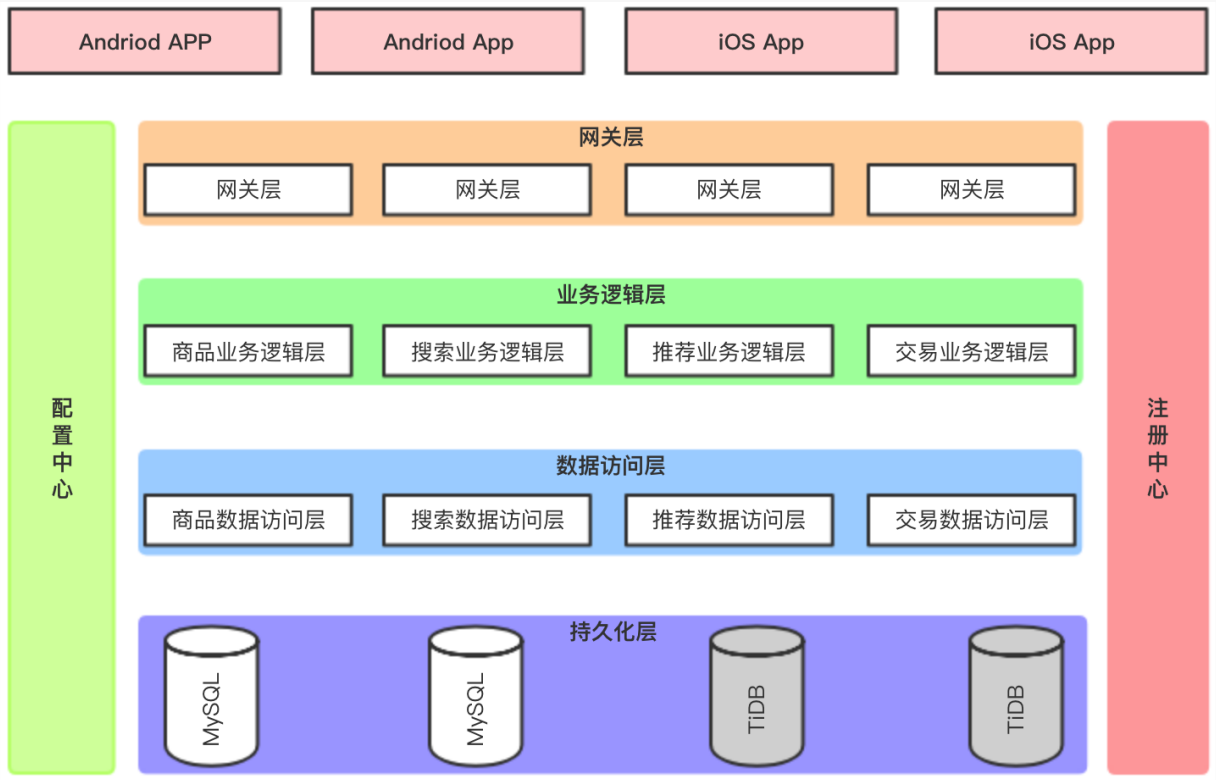
04-3

转转微服务架构3.0 & Service Mesh

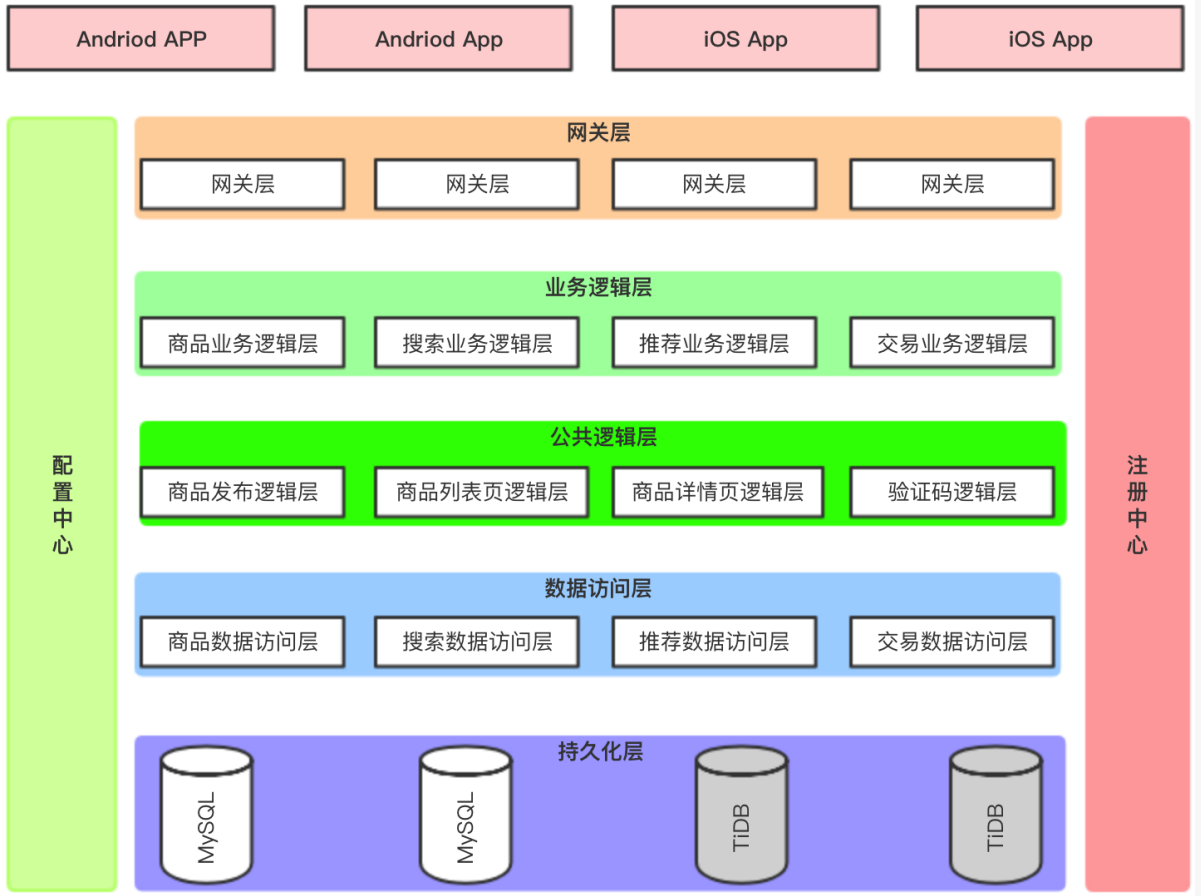
The Version Three Of Microservices architecture AND Service Mesh

04-3 微服务架构3.0

微服务架构2.0



微服务架构3.0



04-3 微服务架构3.0

微服务架构3.0

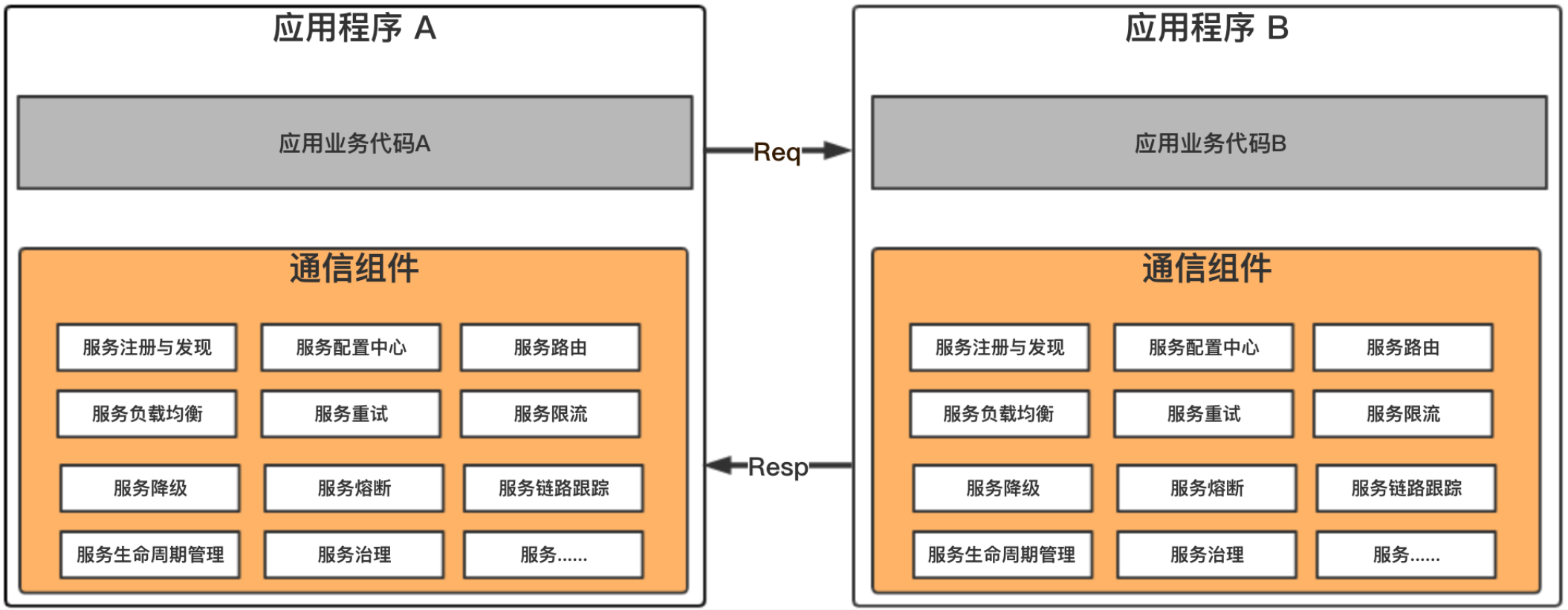
- 抽象成公共逻辑层
  - 发布业务逻辑服务
  - 商品列表页逻辑服务
  - 商品详情页逻辑服务
  - 验证码逻辑服务
  - Push消息逻辑服务
  - ……

架构还有哪些痛点？

04-3 微服务架构3.0

微服务架构痛点之一

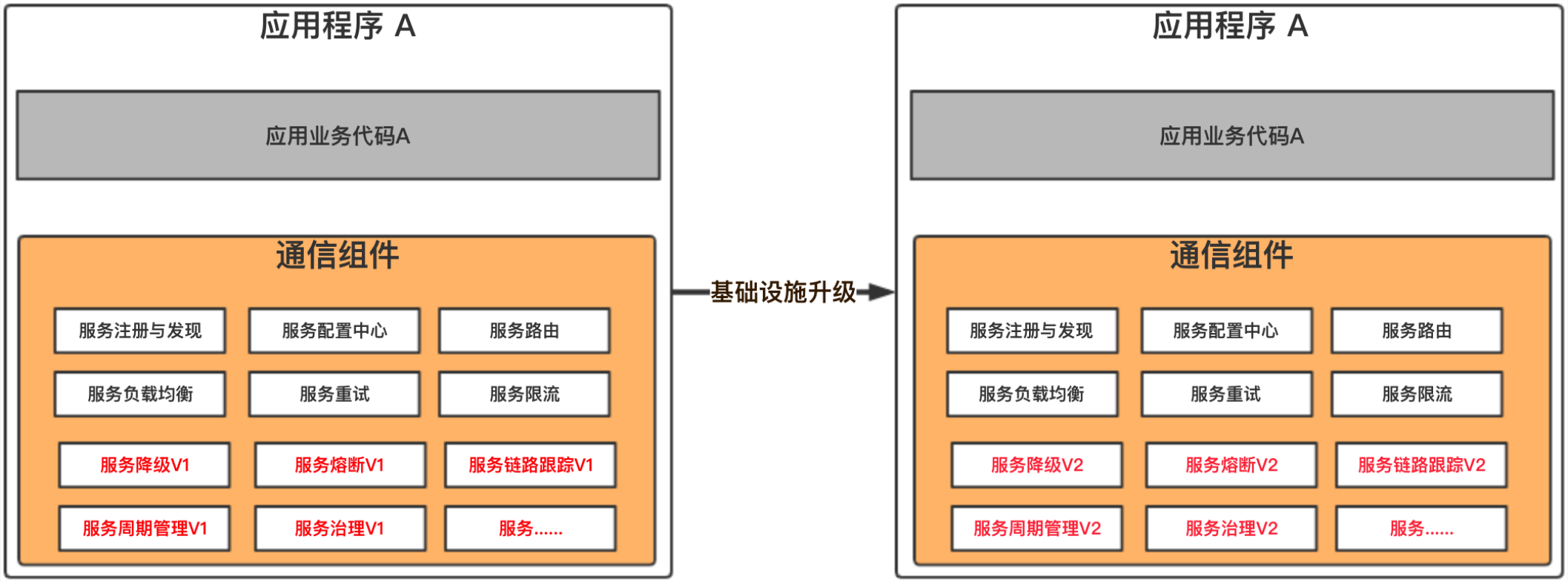
- 业务关注服务间“通信”
  - 业务迭代速度变慢



04-3微服务架构3.0

微服务架构痛点之二

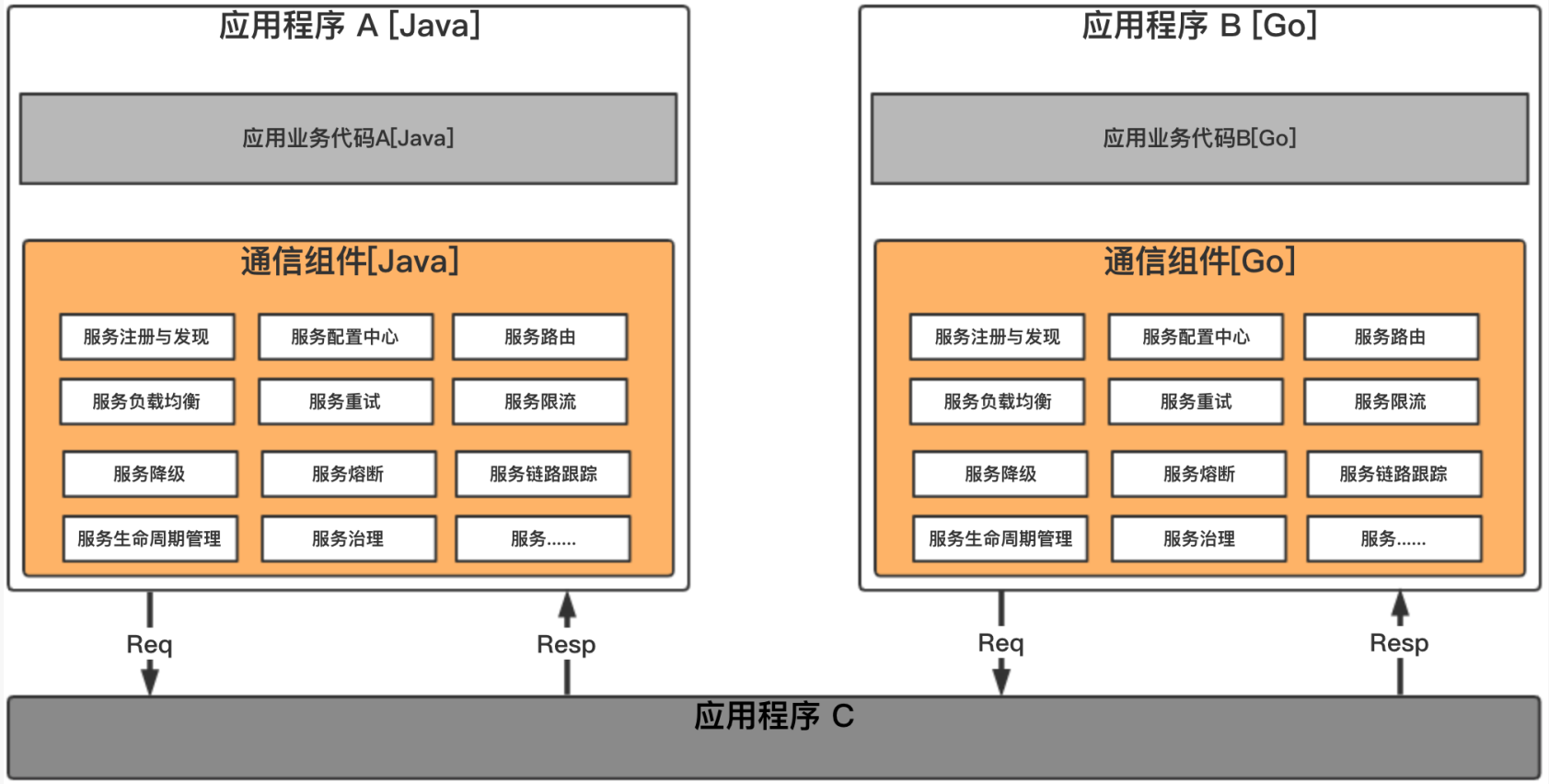
- 基础设施组件升级困难
  - 影响基础设施团队的交付能力和交付速度



04-3 微服务架构3.0

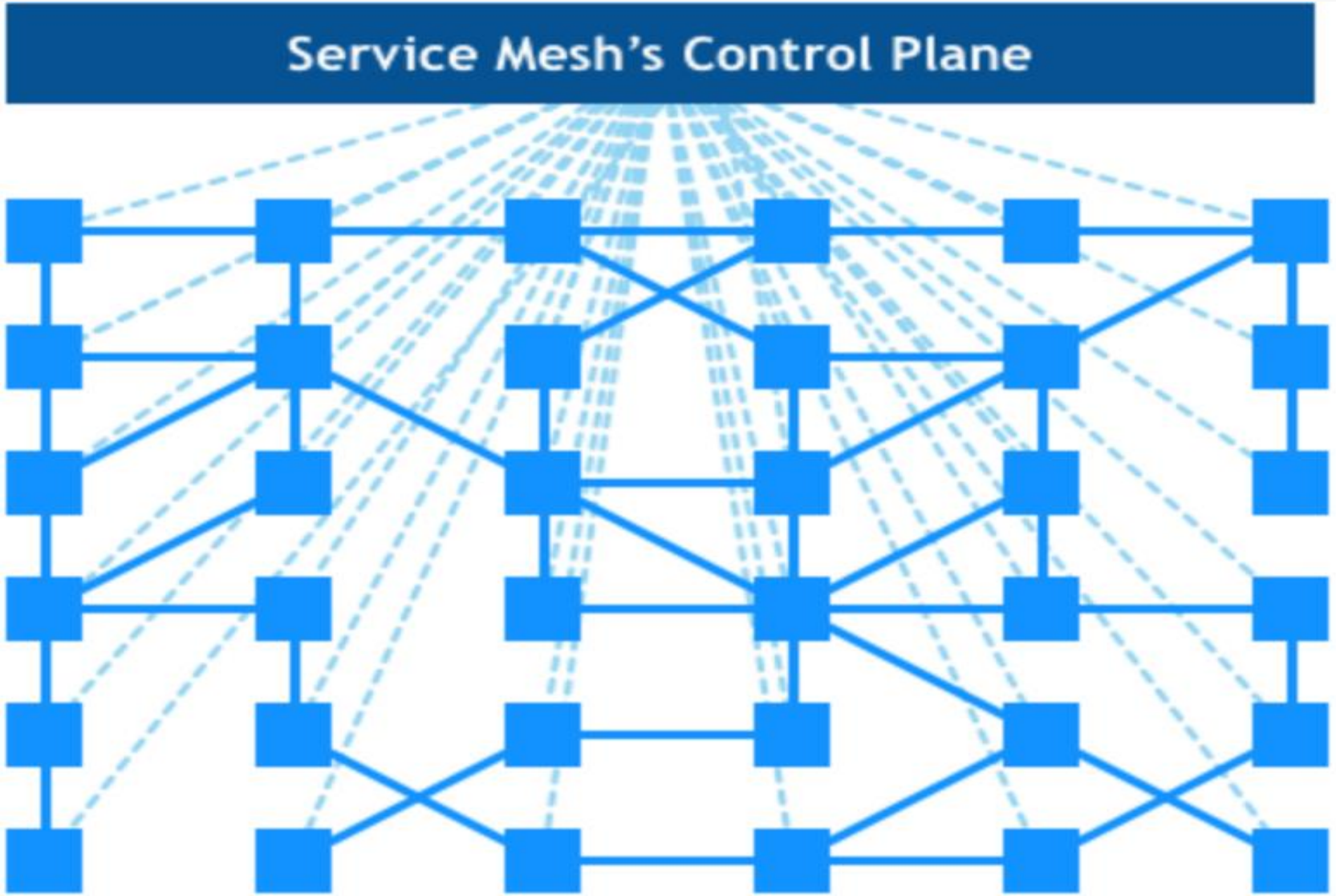
微服务架构痛点之三

- 多语言通信问题
  - 业务每种语言一套基础设施，成本大



04-3 微服务架构3.0

微服务架构演进方向



04-3 微服务架构3.0

Mesh架构实践

- Service Mesh
  - 最早开发Linkerd的Buoyant公司提出，并在内部使用
  - 2016年09月29日第一次公开使用
  - 2017年初，Service Mesh进入国内技术社区视野

Mesh架构实践

- Service Mesh[Willian Morgan(Linkerd的CEO)]定义

WHAT IS A SERVICE MESH?

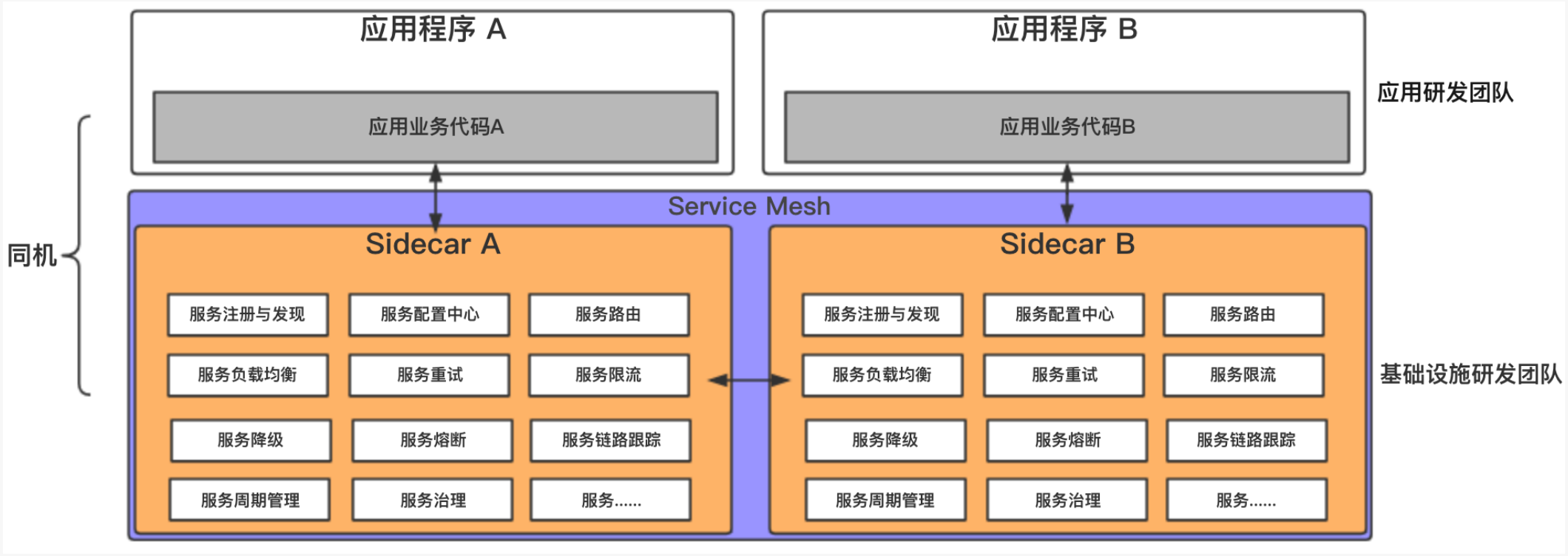
A service mesh is a dedicated infrastructure layer for handling service-to-service communication. It's responsible for the reliable delivery of requests through the complex topology of services that comprise a modern, cloud native application. In practice, the service mesh is typically implemented as an array of lightweight network proxies that are deployed alongside application code, without the application needing to be aware. (But there are variations to this idea, as we'll see.)

服务网格是一个**基础设施层**，用于处理服务间通信。云原生应用有着复杂的服务拓扑，服务网格负责在这些拓扑中**实现请求的可靠传递**。在实践中，服务网格通常实现为一组**轻量级网络代理**，它们与应用程序部署在一起，而**对应用程序透明**。

04-3 微服务架构3.0

Mesh架构实践

- Service Mesh架构



05

微服务架构99.999%高可用解决之道

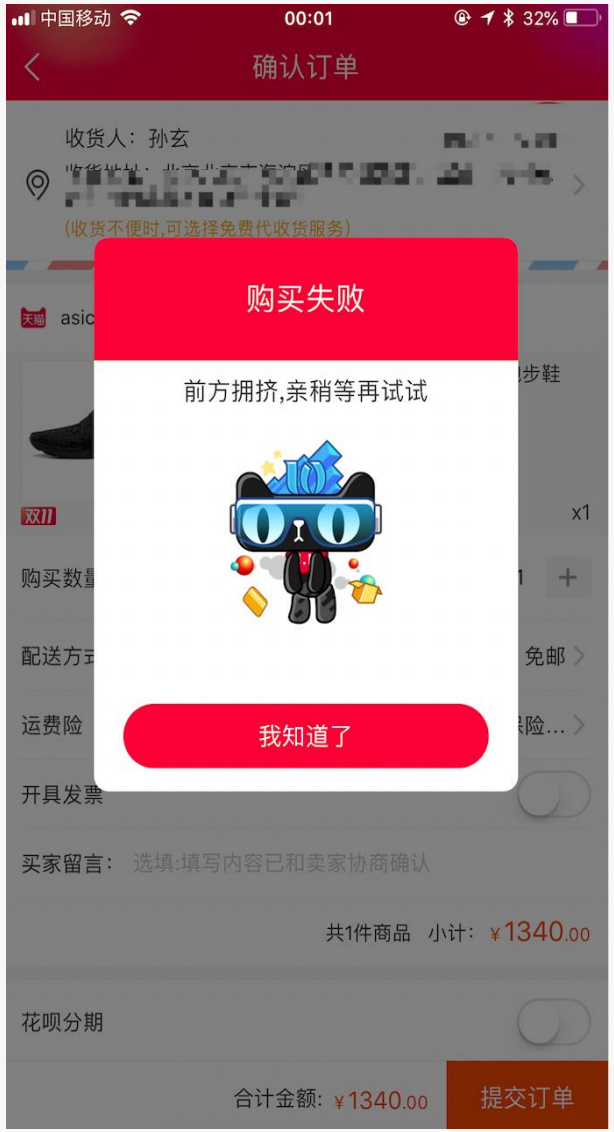
How can Microservice achieve 99.999% high availability?

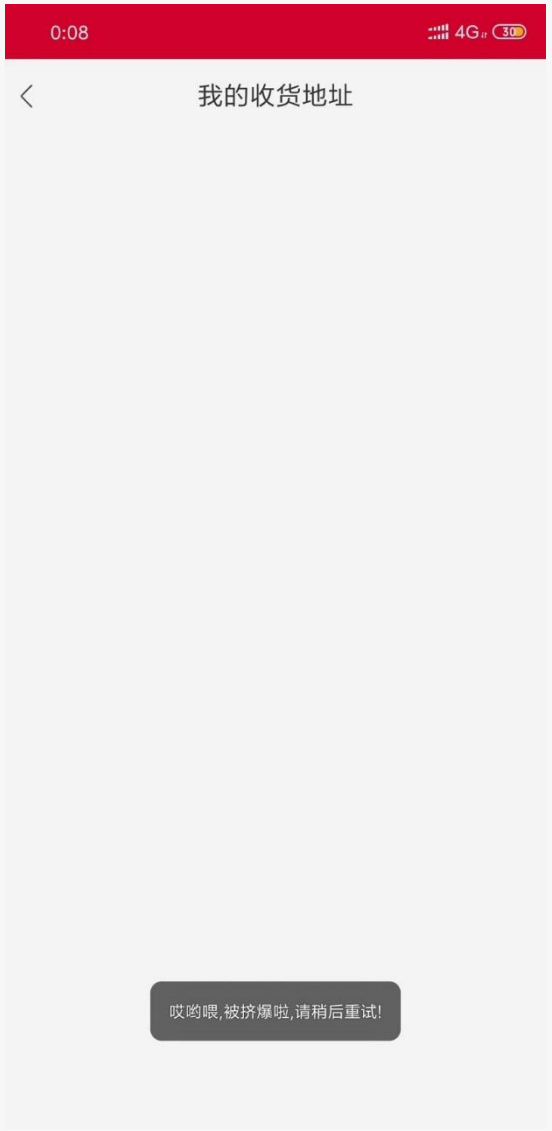
05微服务架构99.999%高可用解决之道

服务突发大流量微博案例

05微服务架构99.999%高可用解决之道

服务突发大流量双11案例





05微服务架构99.999%高可用解决之道

微服务高可用设计手段

1 无状态化

服务冗余

2 负载均衡

幂等设计

3 异步化设计

超时机制

4 服务限流降级熔断

数据复制/缓存/Sharding

5 架构拆分

05微服务架构99.999%高可用解决之道

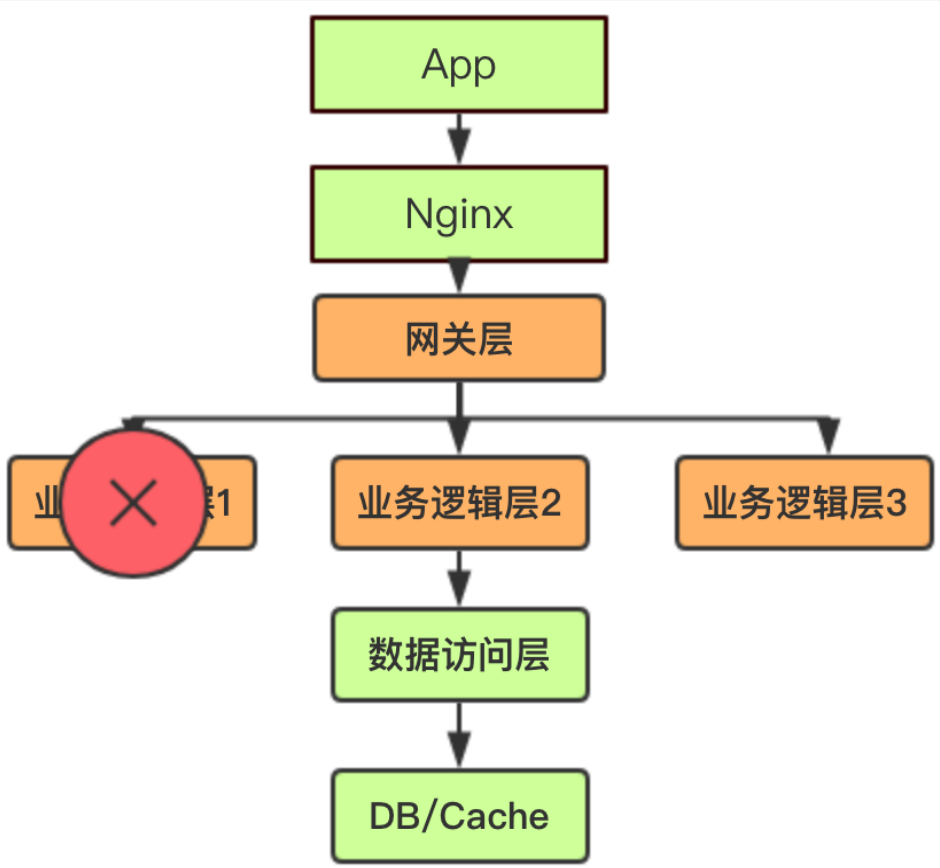
微服务高可用设计手段

05微服务架构99.999%高可用解决之道

广义负载均衡

- 完整的故障处理恢复机制
  - 故障自动发现
  - 故障服务自动摘除
    - 服务熔断机制
  - 请求自动重试
  - 服务恢复
  - 服务恢复自动发现

水平分层架构案例

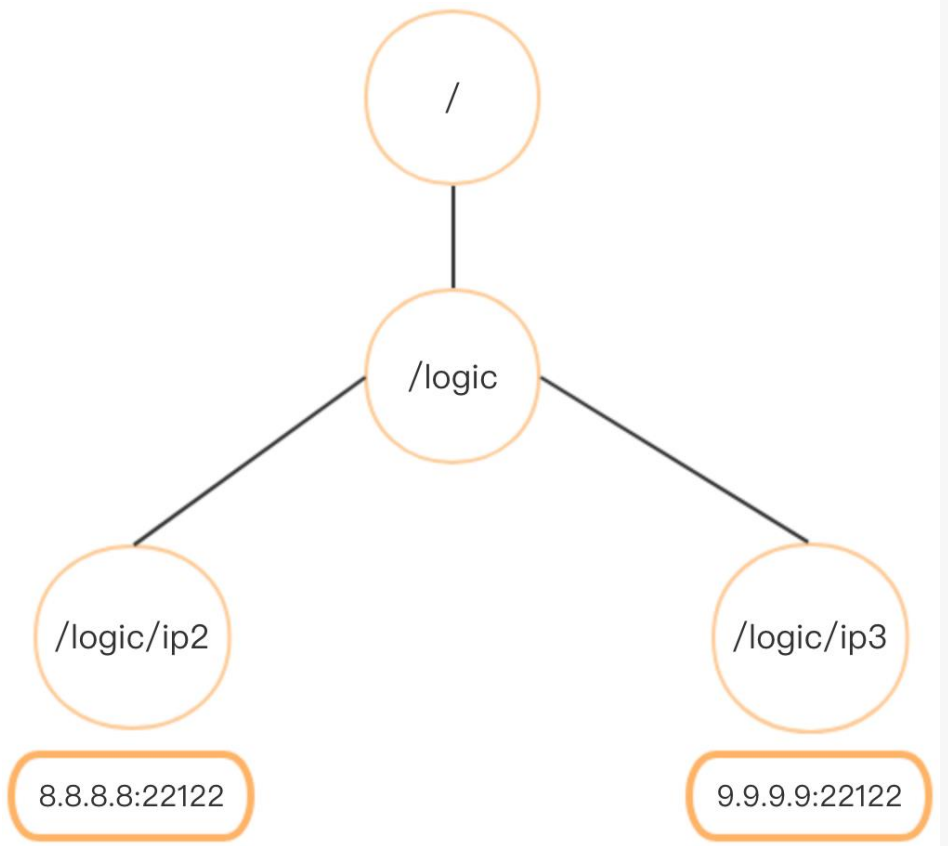


05微服务架构99.999%高可用解决之道

水平分层架构案例

- 业务逻辑层1故障
  - 谁来发现
    - 网关层
    - 注册中心
      - ZooKeeper
      - etcd
      - Consul
      - ……
  - 如何发现
  - 能否发现一切问题
- 服务熔断机制
  - Netflex OSS Hystrix
  - 熔断后恢复机器

ZK核心构成



要点回顾

01
微服务架构拆分之道

02
微服务架构应用场景

03
微服务常用典型技术架构深度剖析

04
微服务架构互联网案例演进

05
微服务架构99.999%高可用解决之道

THANK YOU

I love you more than I've ever loved any woman. And I've waited longer for you than I've waited for any woman. Love you like the warm wind between the mountains, like the rain in the summer, all the time.

作者：孙玄

欢迎关注本人公众号 “架构之美”