



# 实施微服务架构

## 关键技术

演讲人: 孙玄

# 关于我

58集团技术委员会主席

58集团高级系统架构师

转转架构算法部负责人

百度高级工程师

毕业于浙江大学

代表公司多次对外分享

# 关于我





# 分享要点



微服务架构

① 特点

② 架构设计模式及实践案例

③ 数据一致性实践

# 微服务架构

## ● 是什么？

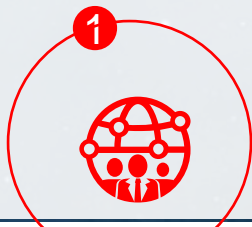
In short, the microservice architectural style is an approach to developing a single application as a **suite of small services**, each **running in its own process** and communicating with lightweight mechanisms, often an HTTP resource API. These services are **built around business capabilities** and **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

## Microservices

common characteristics of this architectural style

by James Lewis and Martin Fowler



# 特点



## 微服务架构



微服务粒度



独立进程



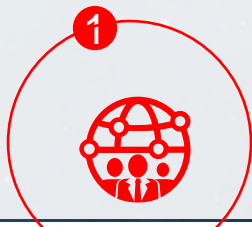
围绕业务建模



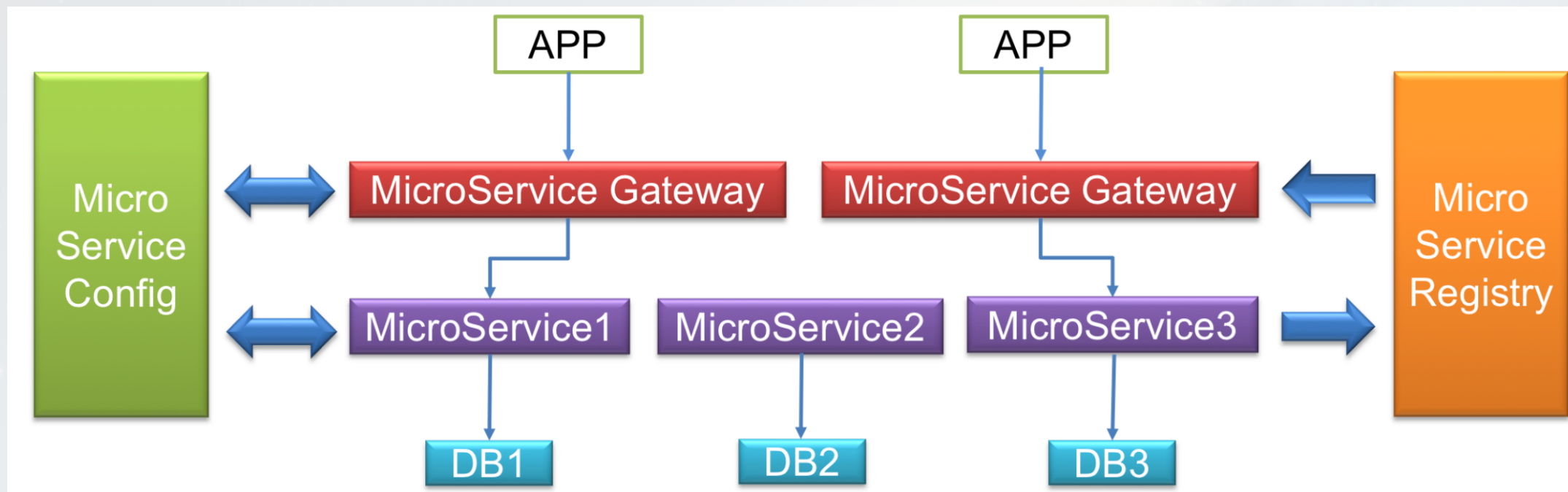
轻量级通信



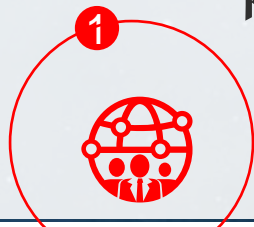
去中心化管理



# 典型微服务架构



网关、微服务、数据存储、注册中心、配置中心





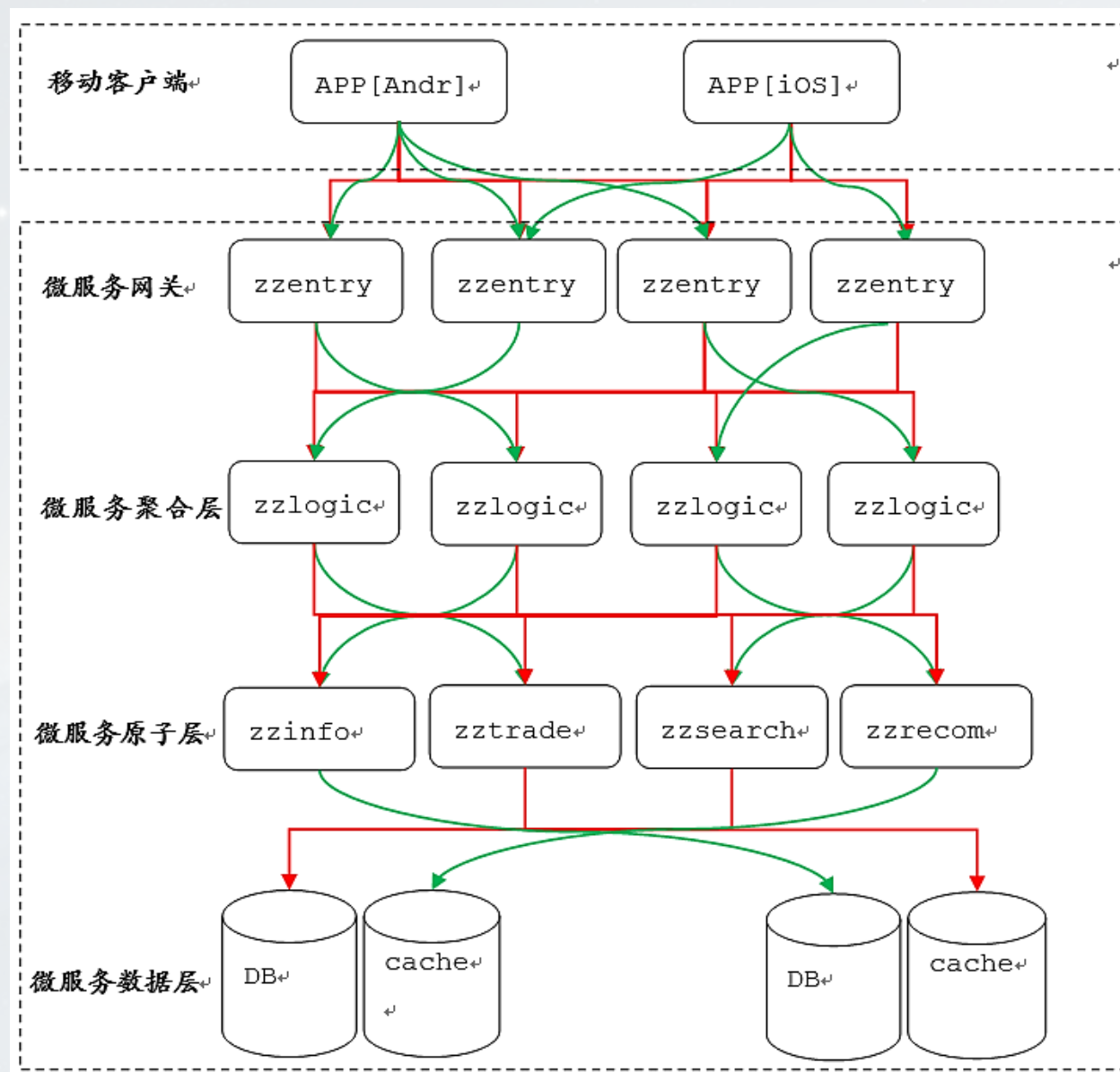
# 二手交易平台（转转）微服务架构





# 总体架构设计

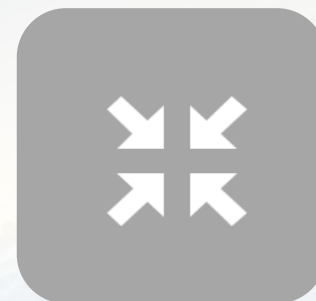
- ✓ 网关
- ✓ 聚合层
- ✓ 原子层
- ✓ 数据层
- ✓ 注册中心
- ✓ 配置中心



# 架构设计模式及实践案例



链式  
设计模式



聚合器  
设计模式



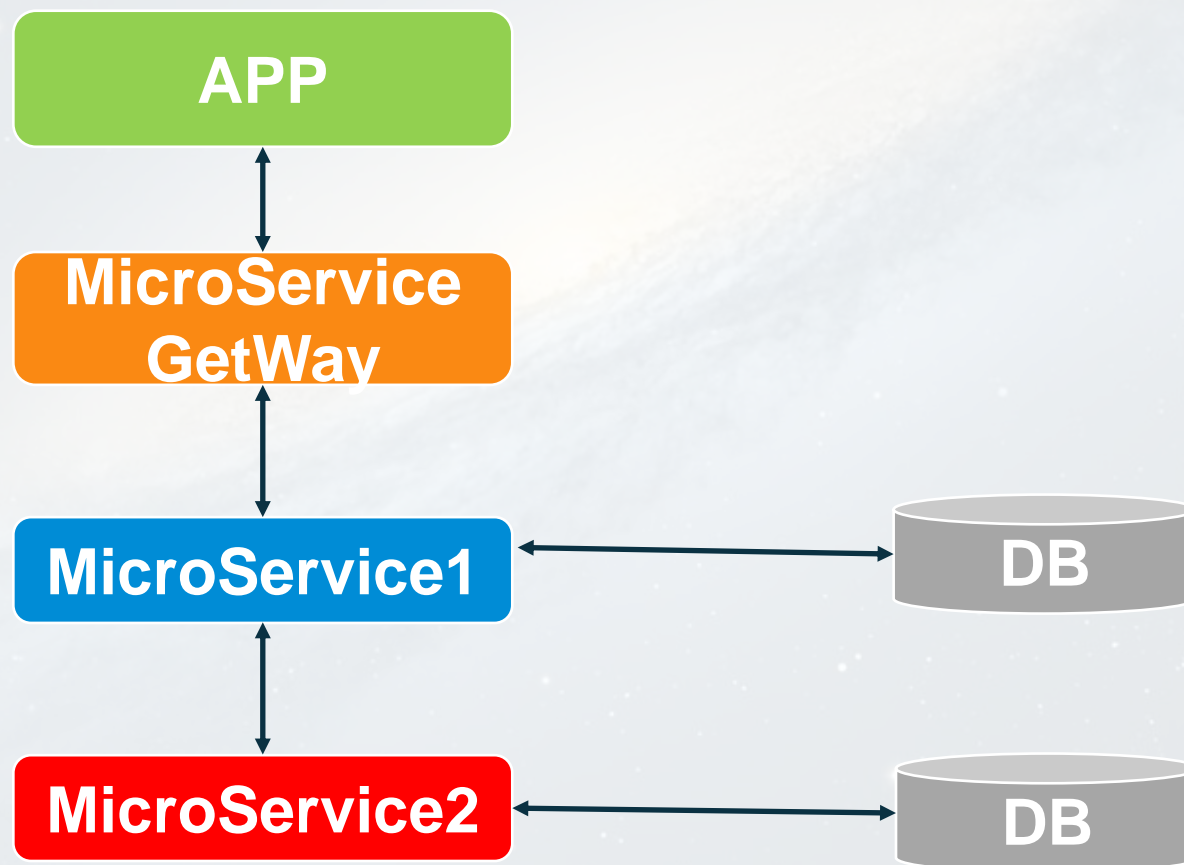
数据共享  
设计模式



异步消息  
模式

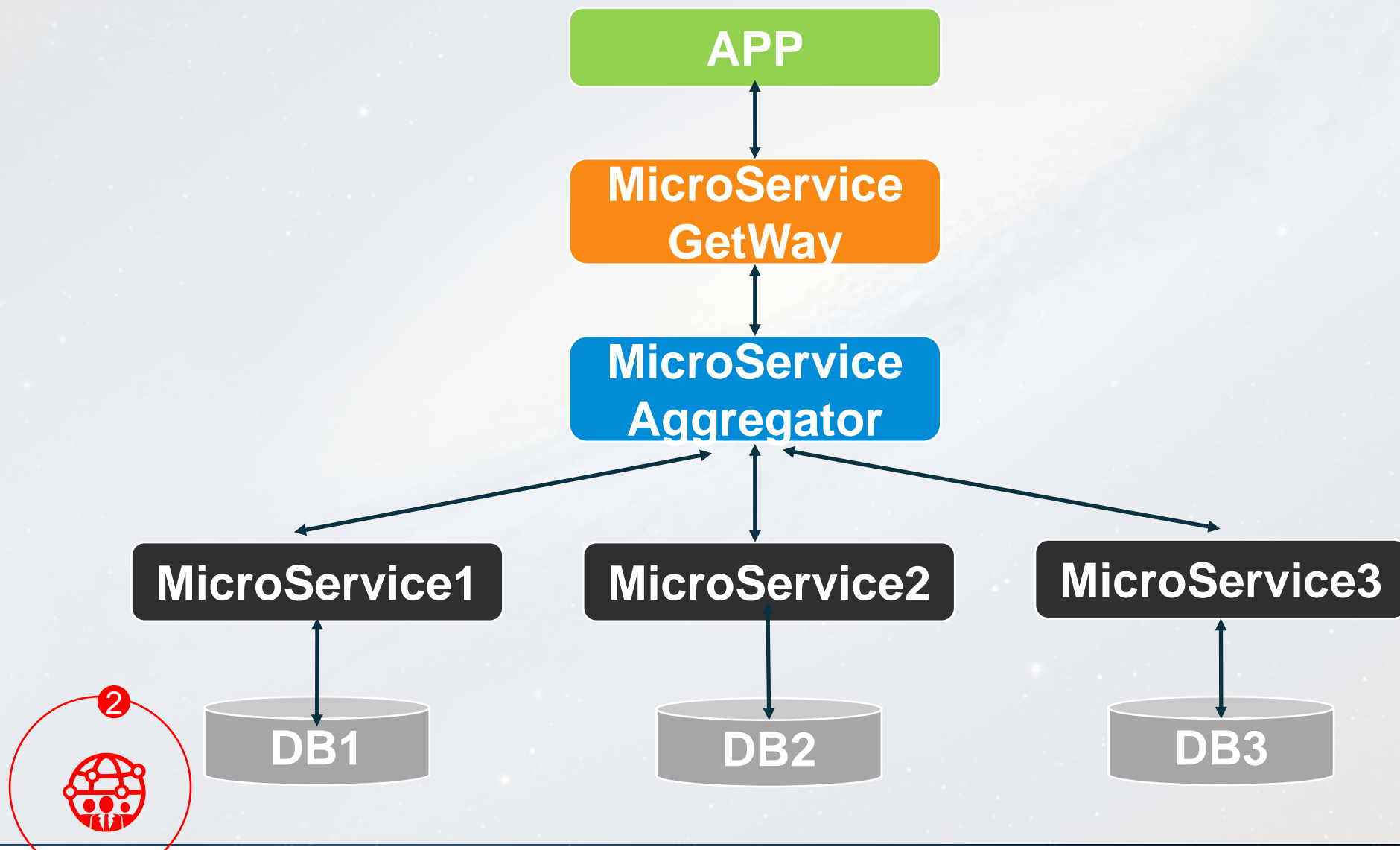


# 链式设计模式

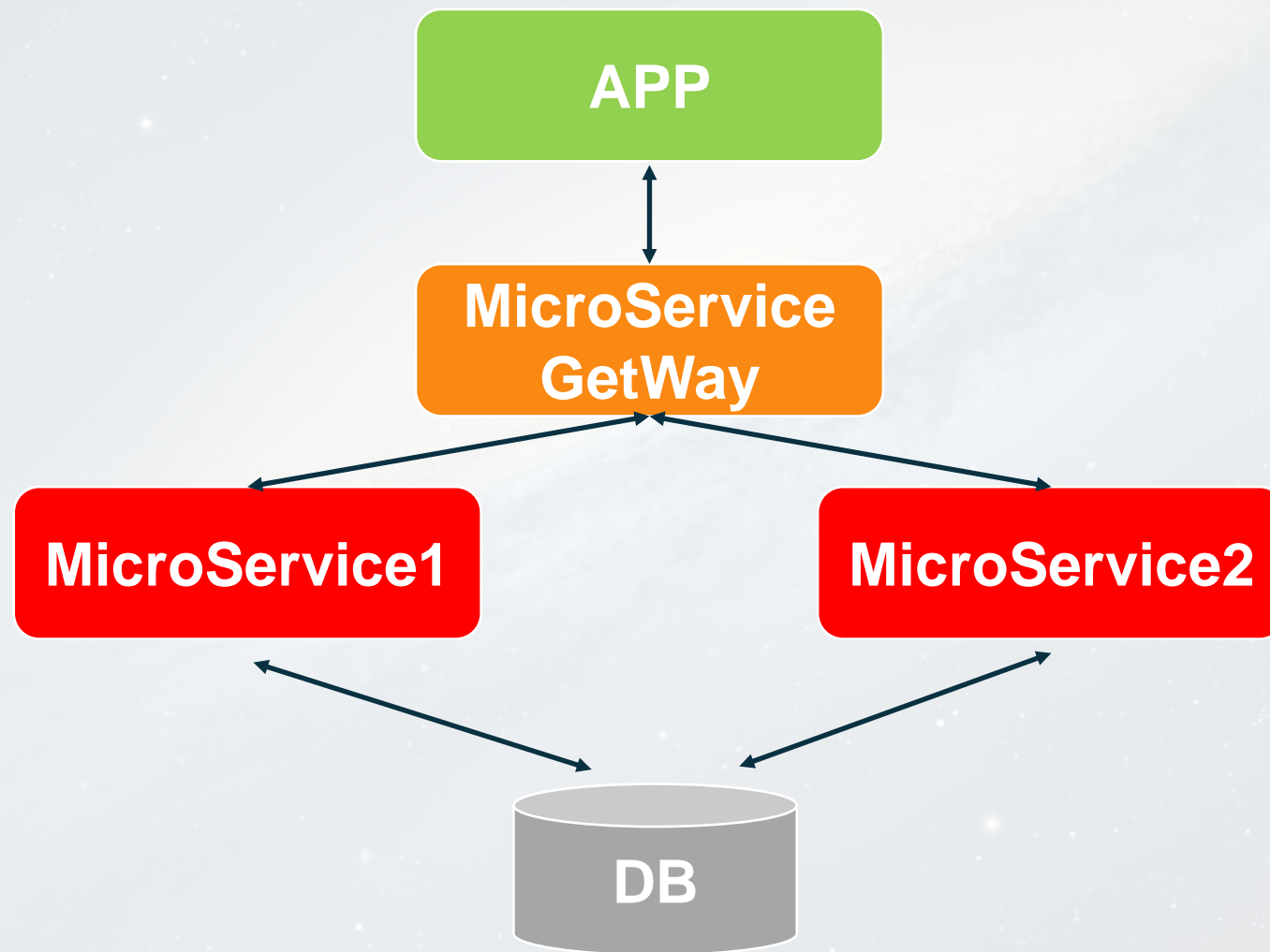




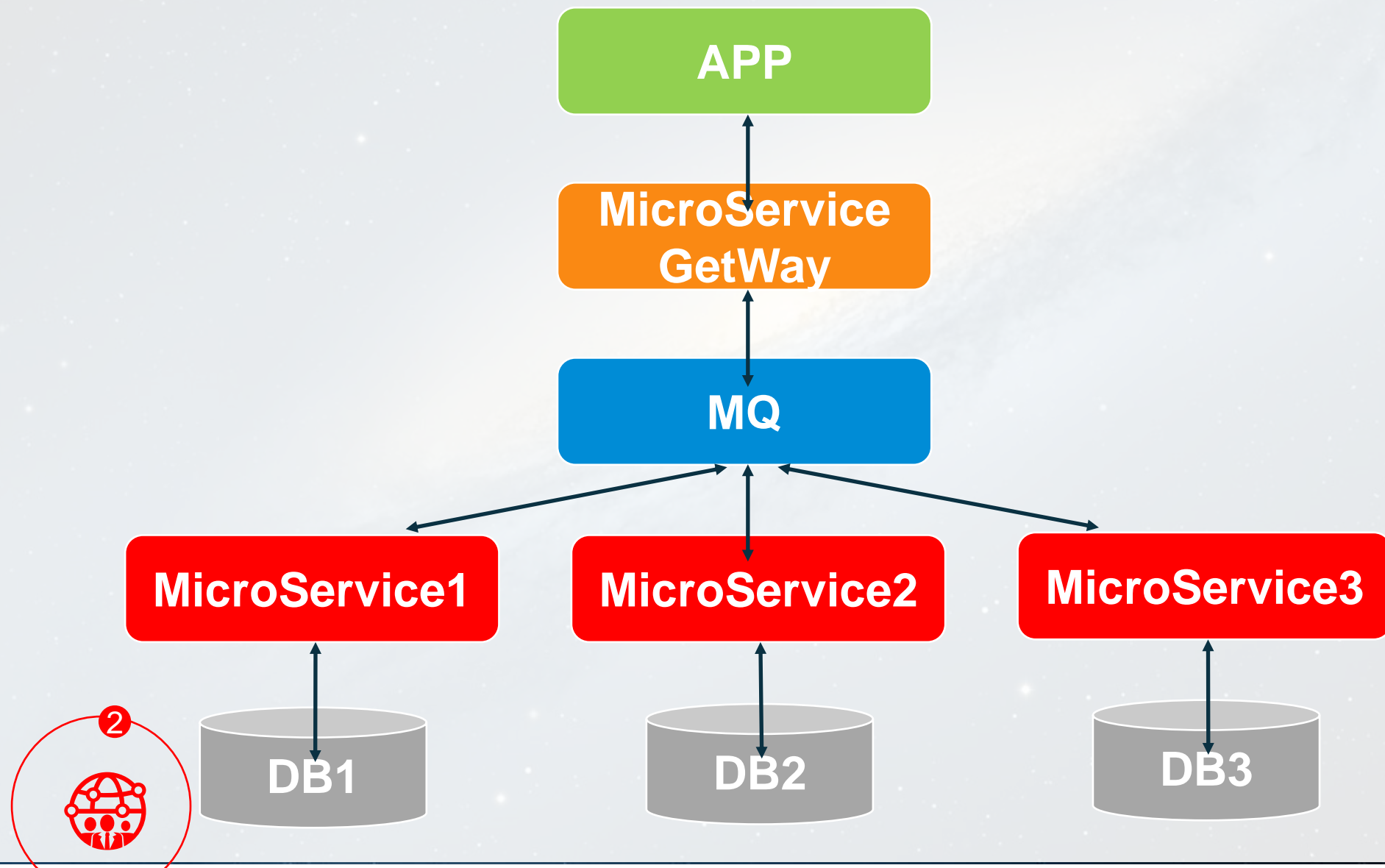
# 聚合器设计模式



# 数据共享设计模式



# 异步消息设计模式





# 微服务架构设计模式

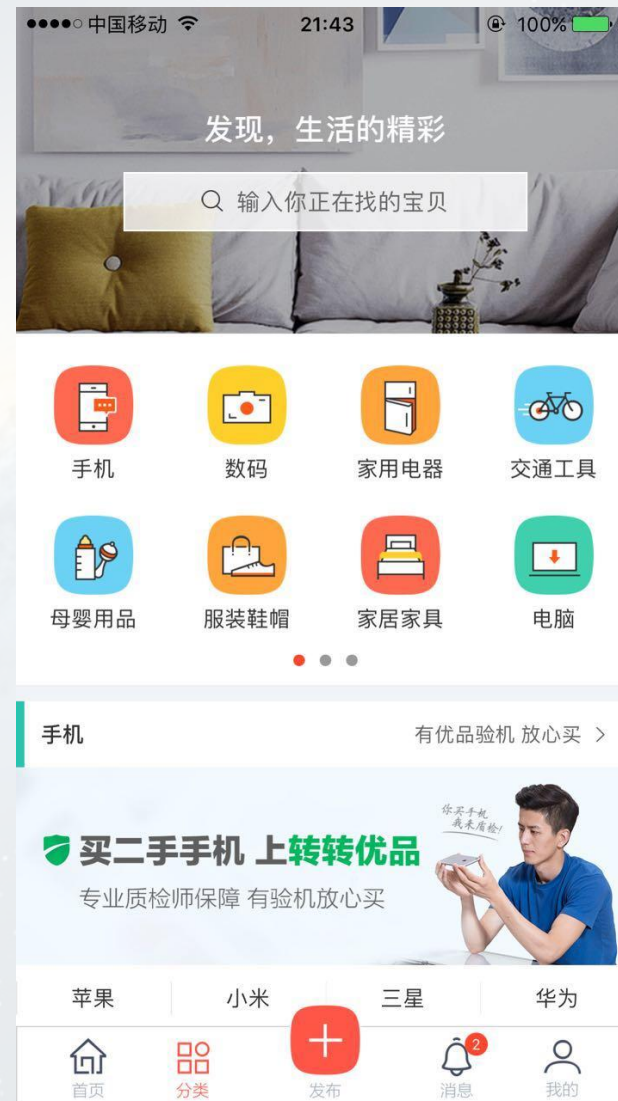
实际项目使用

- ✓ 聚合器设计模式
- ✓ 异步消息设计模式
- ✓ 数据共享设计模式

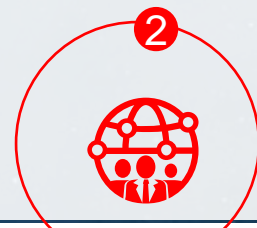
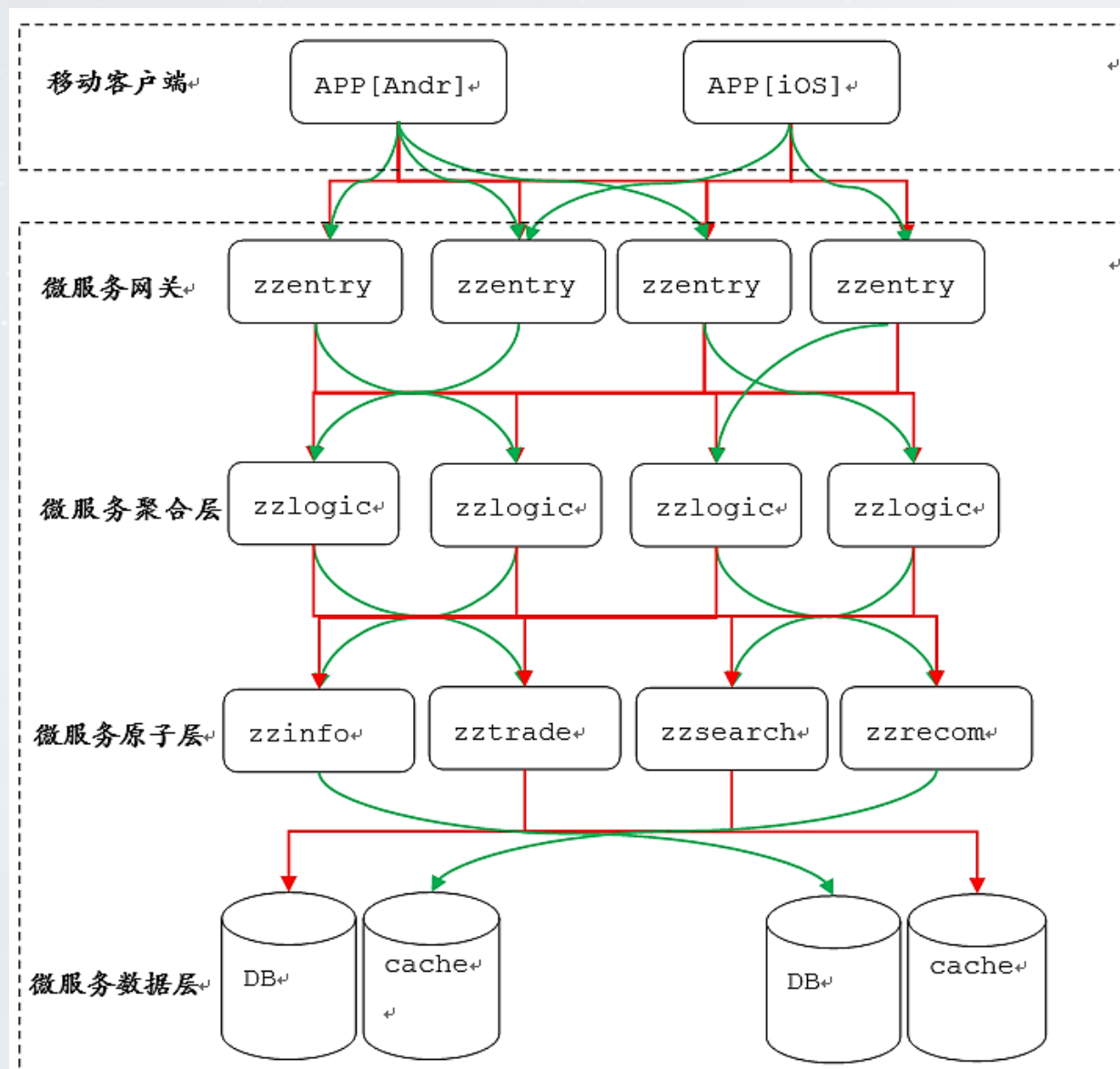


# 二手交易平台实践案例一『聚合器模式』

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



# 实践案例一『聚合器设计模式』





# Feed系统实践案例二『异步消息模式』



musicm

修改个人资料

- 文章
- 相册
- HOHO
- 分享
- 投票
- 测试
- 礼物
- 宠物
- 应用大厅

+ 添加 设置

2012年03月02日: “又到福来day, 各位周末愉快~”

2012啦! 新年快乐!

发布

表情 手机也能发hoho了, 快来试试吧! waphi.baidu.com 还能输入140字

好友动态

全部 文章 相册 分享 HOHO 设置



solaryf发表了新 **文章** 下午的时光 1小时前

坐在阳台书桌边上, 泡上一杯竹叶青, 吃点小饼干, 翻看新买的一摞书: 《数学的诱惑: 日常生活中的数字游戏》, 《你以为的就是你...

☒ 阅读全文 ☒ 分享 ☒ 评论(0)



一框知天下在 通辽吧 中发帖 **【公告】** 通辽吧垃圾帖子回收站: 需要删除的帖子请此处举报 6小时前



一框知天下在 贴吧合并吧 中发帖 **【申请合并】** 申请将内蒙古通辽吧合并到通辽吧 7小时前

2



# Feed系统实践案例二『异步消息模式』

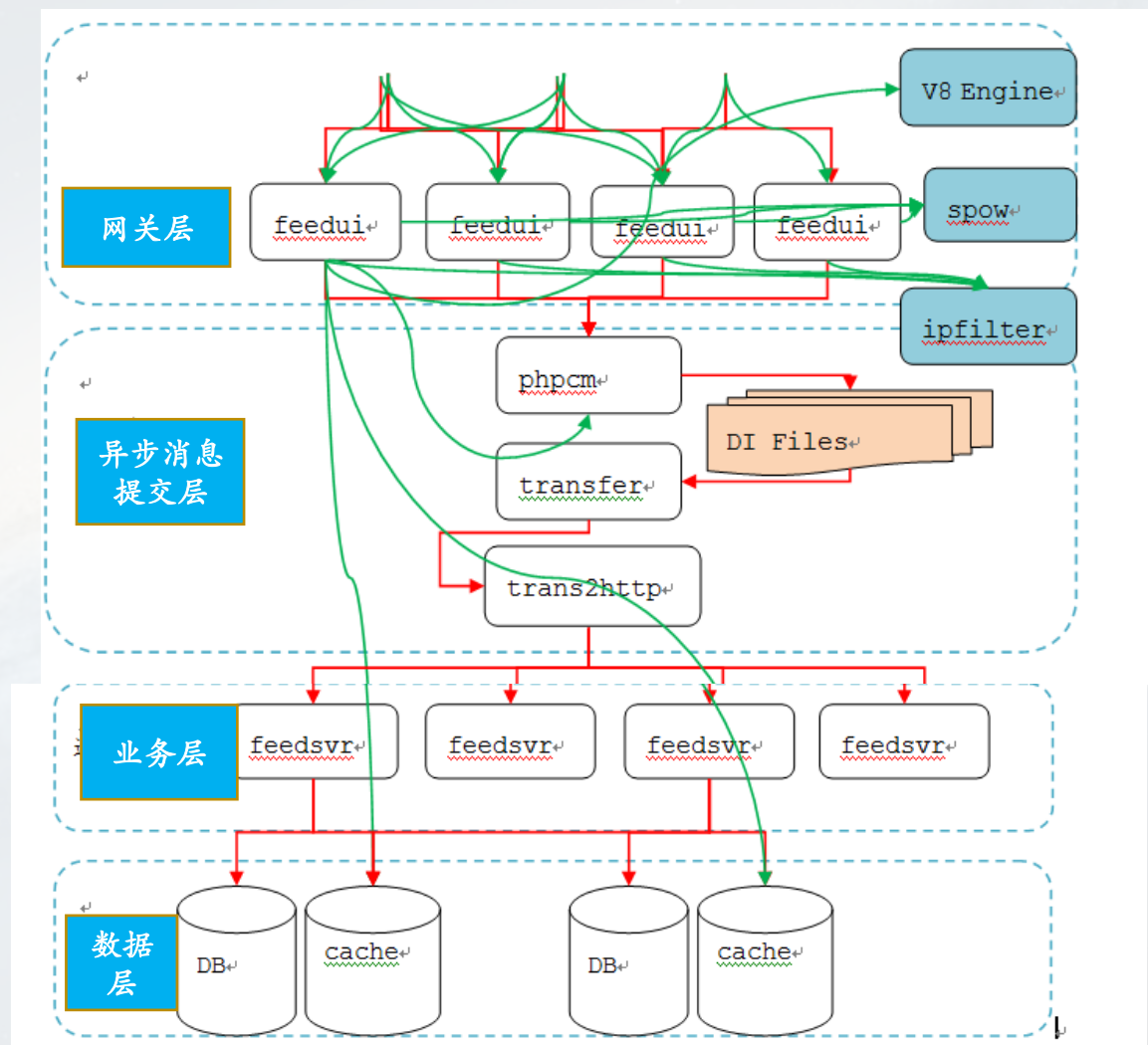
## ✓ Feed系统微服务架构

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



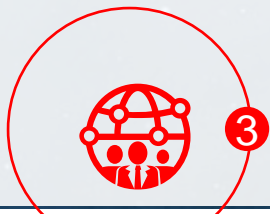
# 数据一致性保证

微服务  
架构不  
是银弹

- ✓ 分布式系统复杂性高
- ✓ 可靠性难度大
  - 微服务多、调用失败
- ✓ 数据一致性难度高



目标：数据最终一致性





# 数据一致性保证

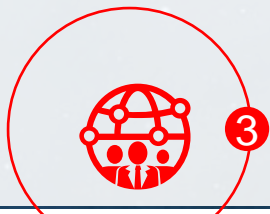
## ✓ 微服务不同数据库

- Local Transaction Not available

## ✓ 微服务性能

- 2PC/3PC Not available
- 同步

怎么办？！



# 数据一致性保证

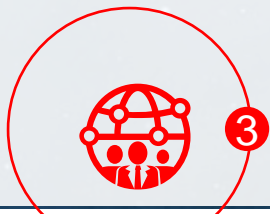
## 业务场景

✓ 异步调用

-MQ

✓ 同步调用

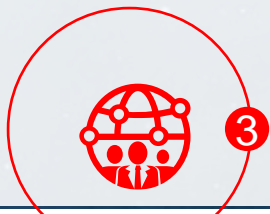
-如何做



# 数据一致性保证

基于异步补偿

分布式事务



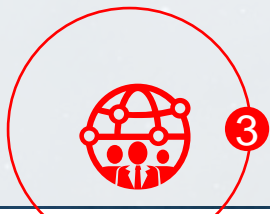
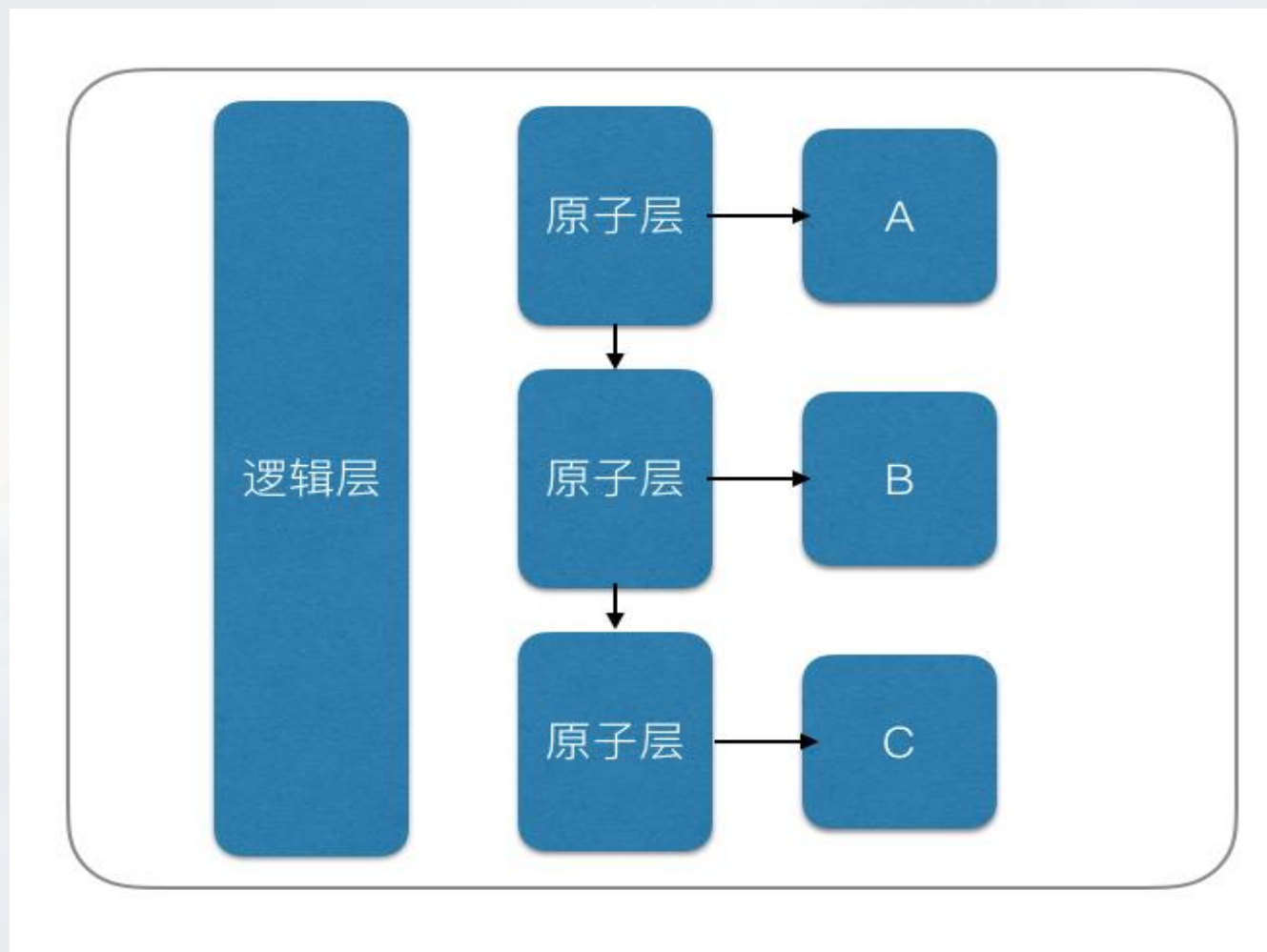
# 数据一致性保证

## ✓ 业务场景

- 同步业务的一致性保证
- 不同微服务
  - A/B/C不同库

## ✓ 聚合层实现事务语义策略

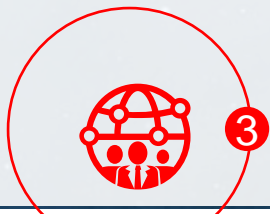
- 记录业务调用链
- 异常时沿调用链反向补偿





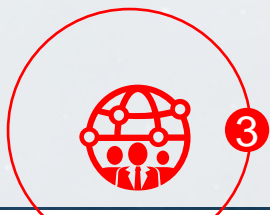
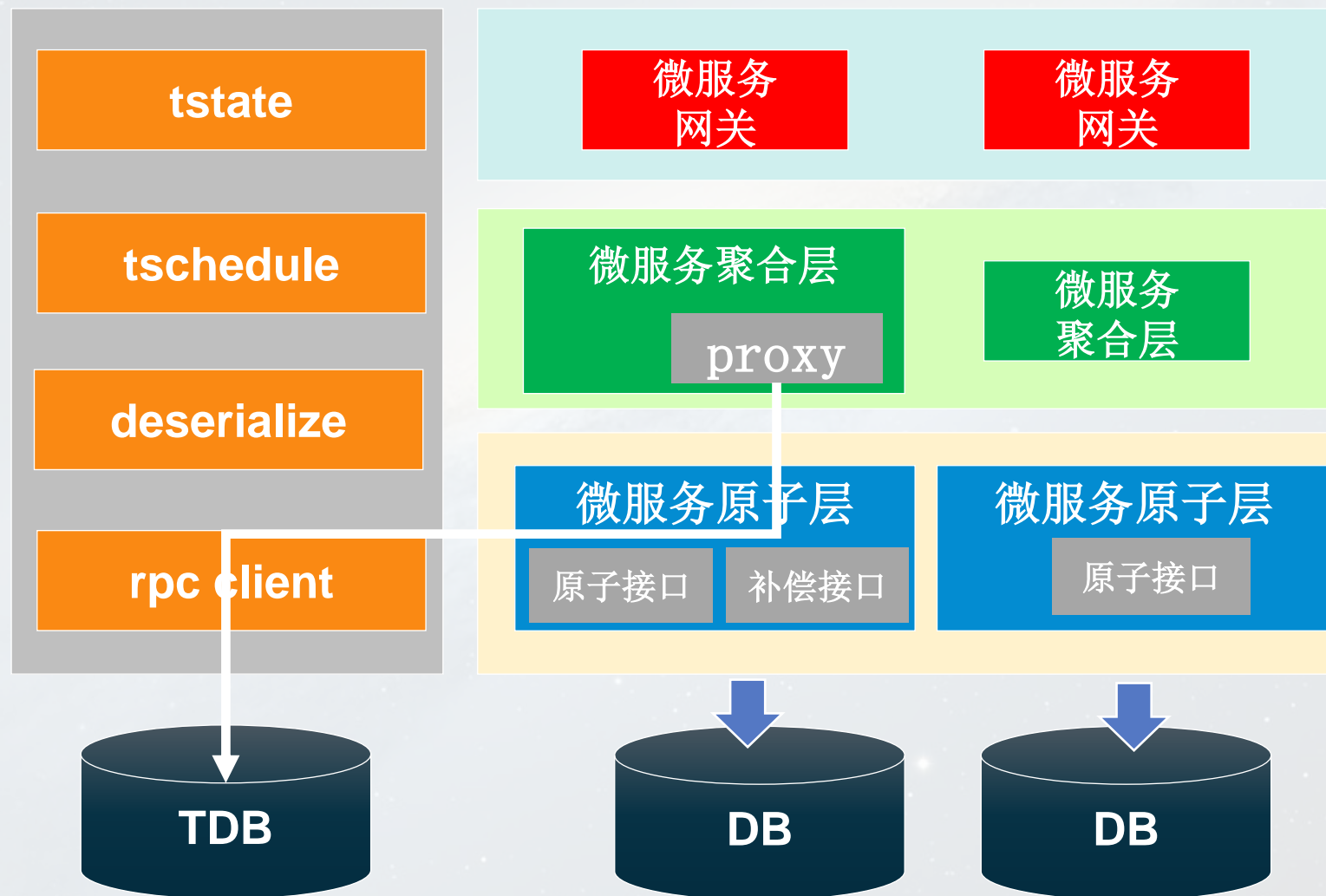
# 基于异步补偿分布式事务

## 架构设计关键点



# 基于异步补偿分布式事务

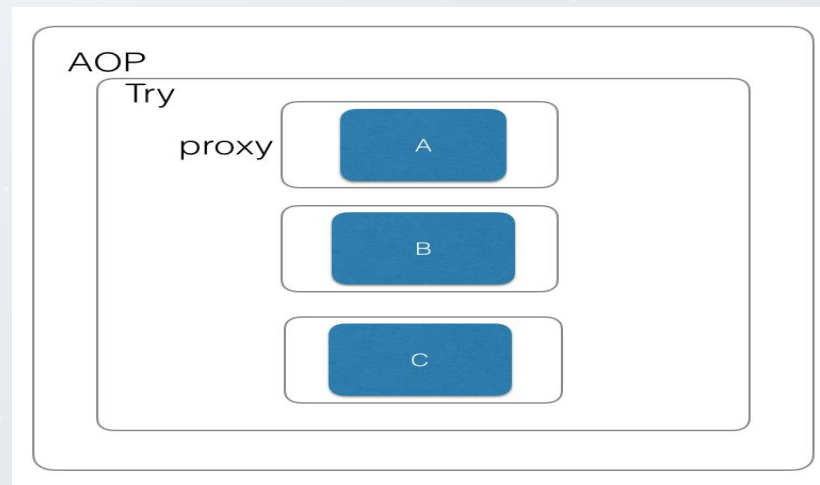
## 架构设计



# 基于异步补偿分布式事务

## 聚合层Proxy

- ✓ 基于方法，在方法名加注解标注补偿方法名
  - @Compensable(cancelMethod= “cancelRecord” )
- ✓ 聚合层调用原子层之前，通过代理记录当前调用请求参数
- ✓ 如果业务正常，调用完成后，当前方法的调用记录存档或删除
- ✓ 如果业务异常，查询调用链回滚



# 基于异步补偿分布式事务

## 微服务原子层

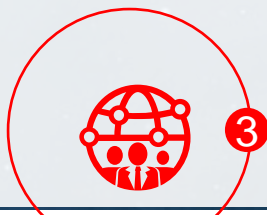
- ✓ 正常原子接口
- ✓ 补偿幂等接口

微服务原子层

补偿接口

微服务原子层

原子接口





# 基于异步补偿分布式事务

## 分布式事务补偿服务

### ✓ 事务组表

- 事务组状态

txid	state
------	-------

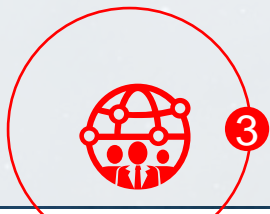
### ✓ 事务调用组

- 记录事务组内的每一次调用以及相关参数

txid	actionid	callmethod	paramtypes	params
------	----------	------------	------------	--------

### ✓ 补偿策略

- 调用执行失败，修改事务组状态
- 异步执行补偿



# 基于异步补偿分布式事务

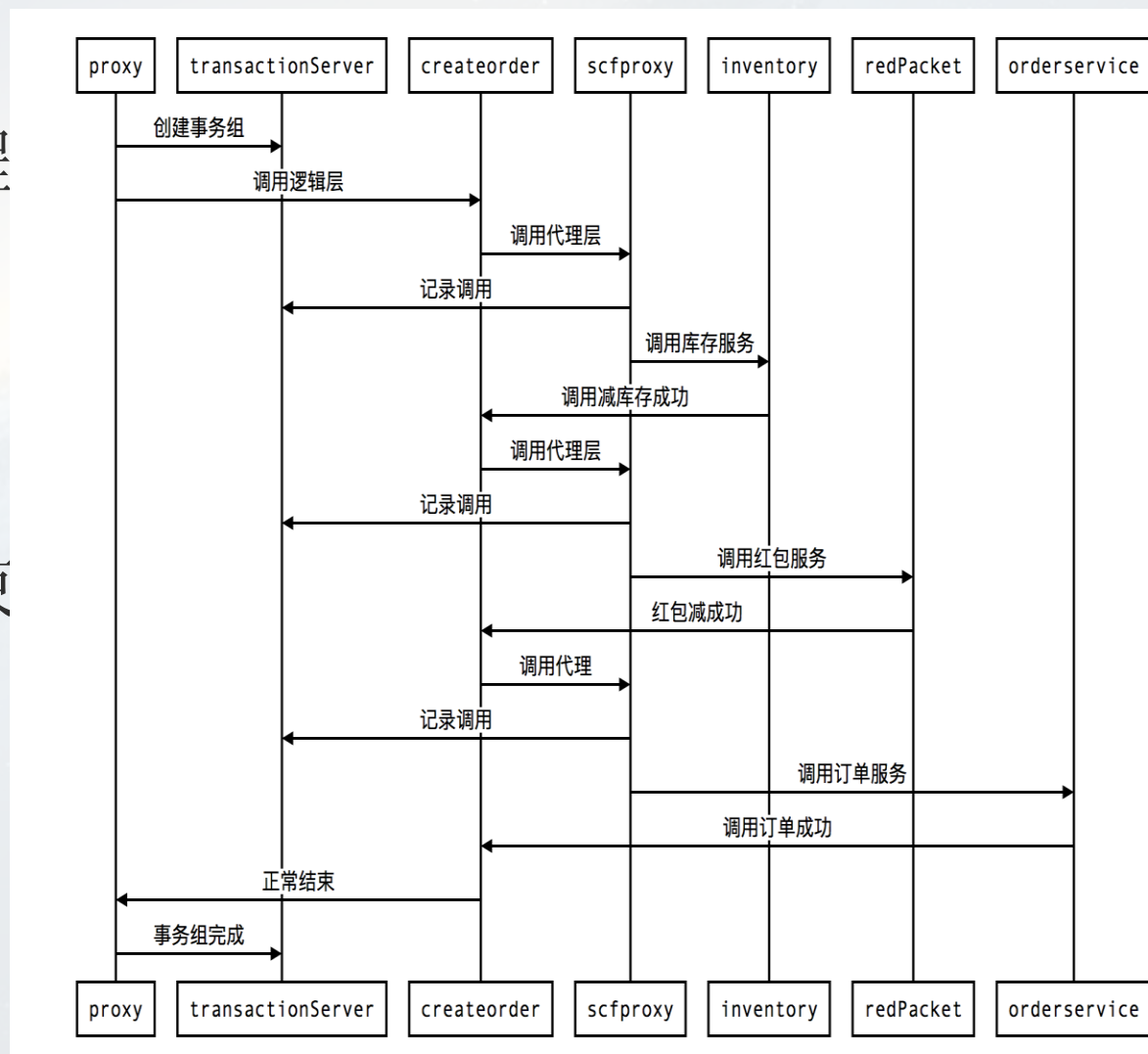
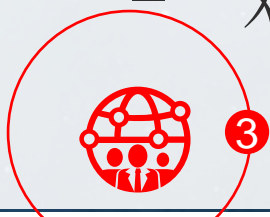
## 案例

二手交易创建订单事务组正常流程

✓ 锁库存->减红包->创建订单

✓ 代理层透明记录调用请求参数

- 记录事务域的开始与结束
- 在所有远程调用成功时
- 对业务逻辑不做侵入

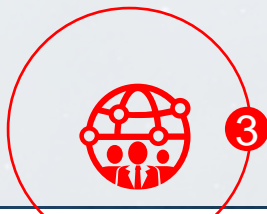
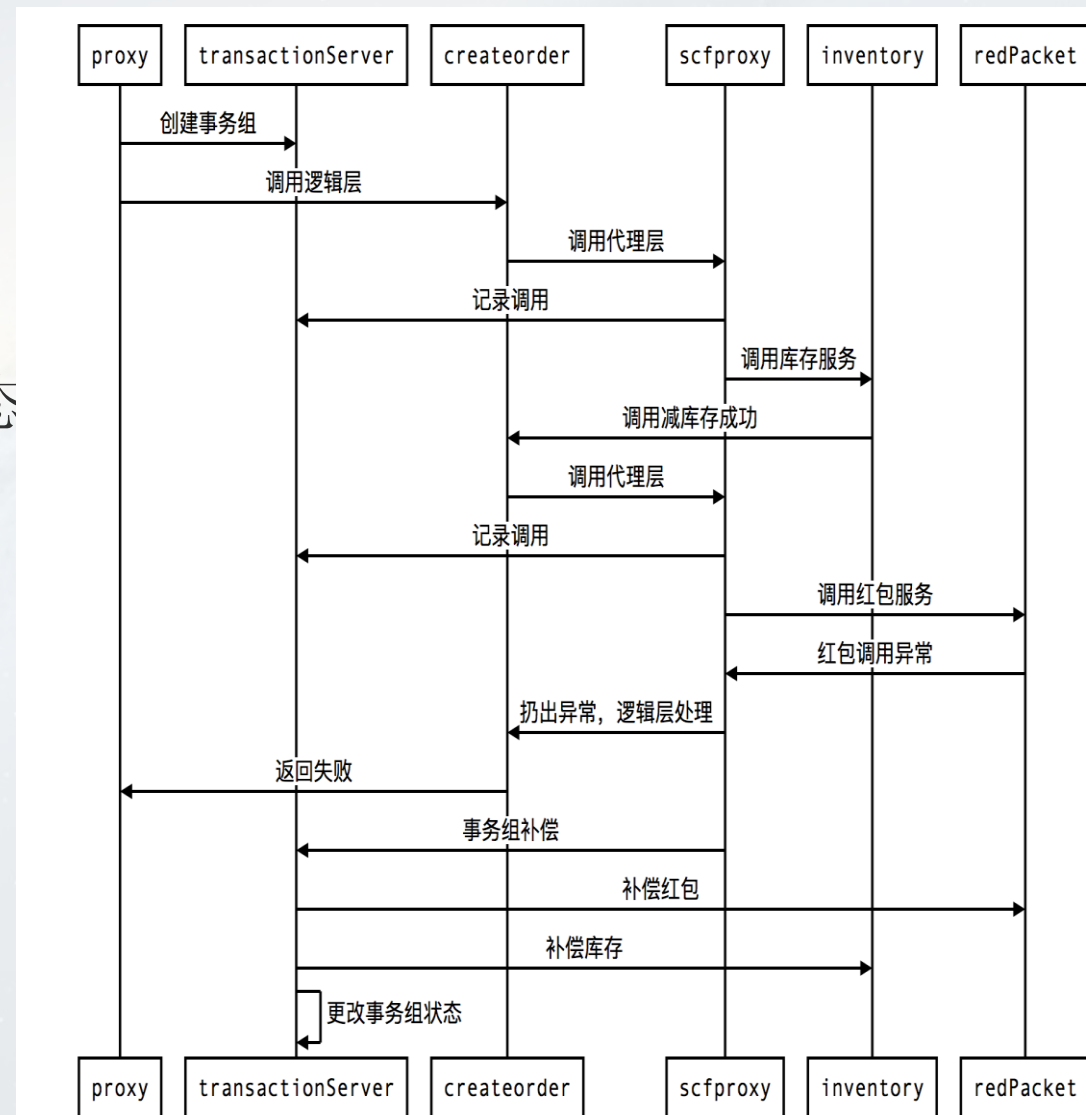


# 基于异步补偿分布式事务

## 案例

### ✓ 二手交易创建订单事务组异常流程

- 微服务原子层失败，代理更改事务组状态
- 微服务业务正常执行
- 事务补偿服务异步执行补偿



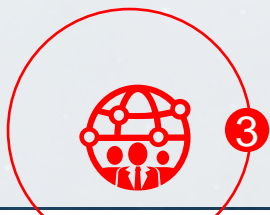
# 基于异步补偿分布式事务

## 分布式事务本身补偿

✓ 基于日志

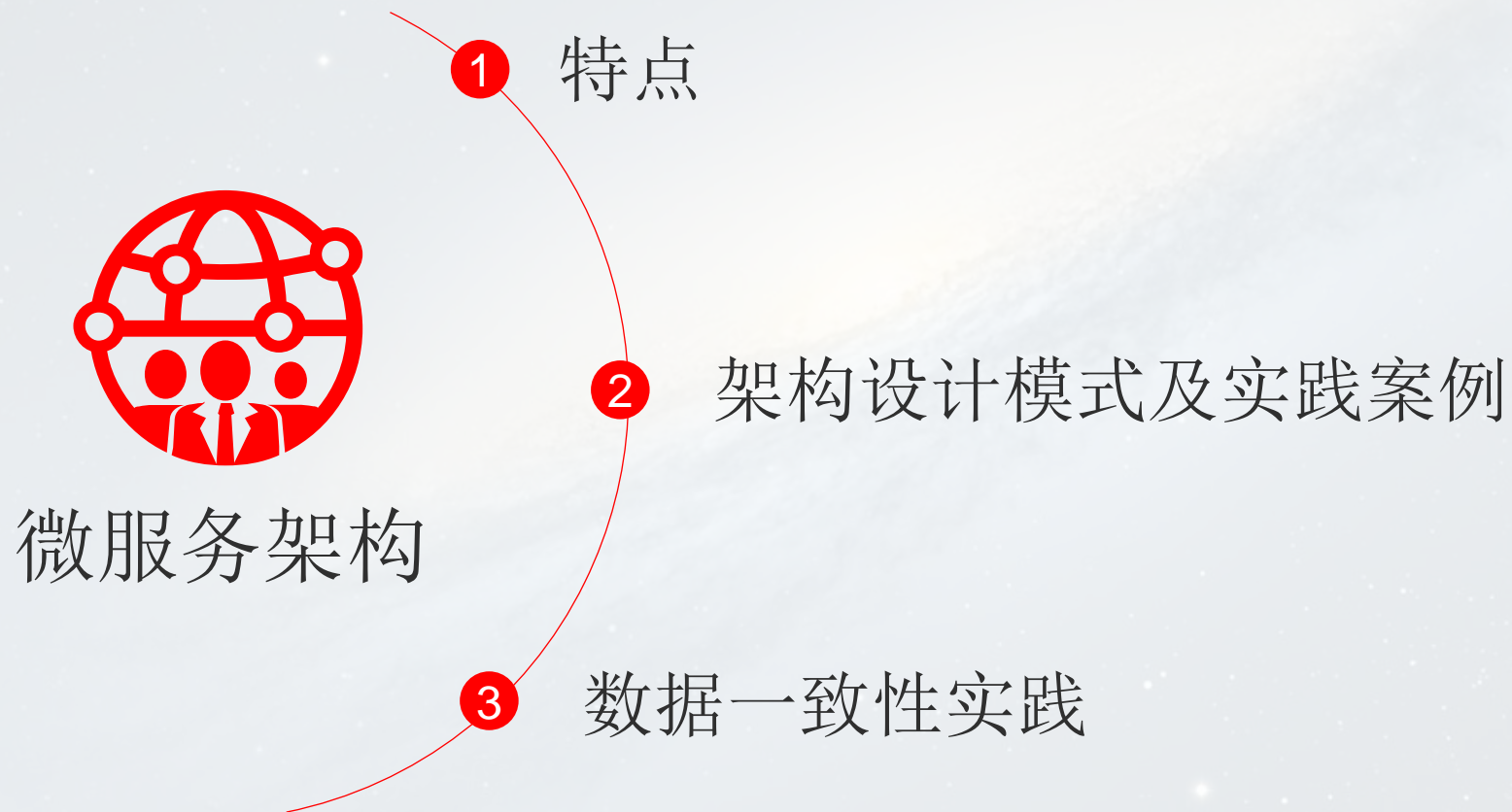
✓ 报警

✓ 人工接入





# 分享要点



欢迎关注本人公众号“架构之  
美”

Q & A

