

# 淘宝开放平台订单推送架构演进

朱天峰

2017.06

# 数据同步的两种实现

数据库之间的同步



简单的业务计算

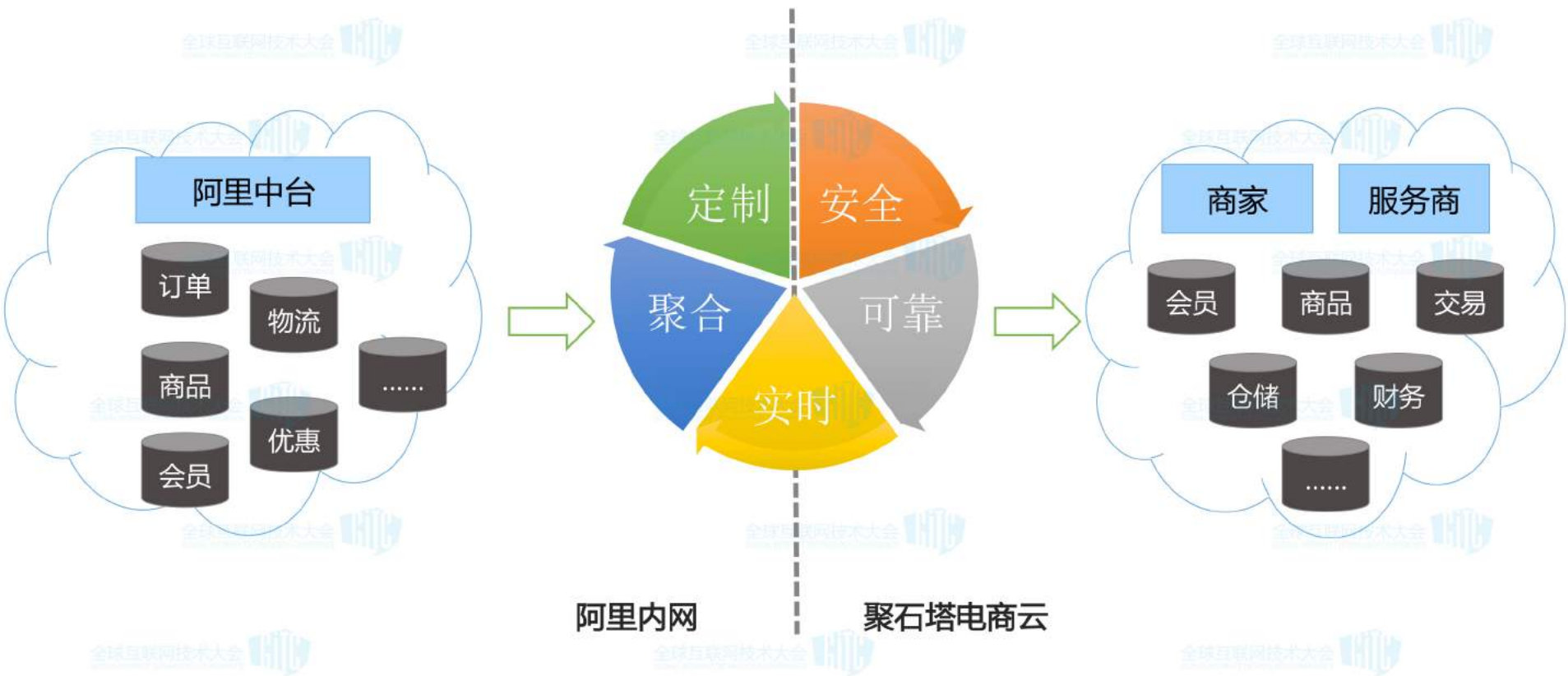
V.S.

应用之间的同步

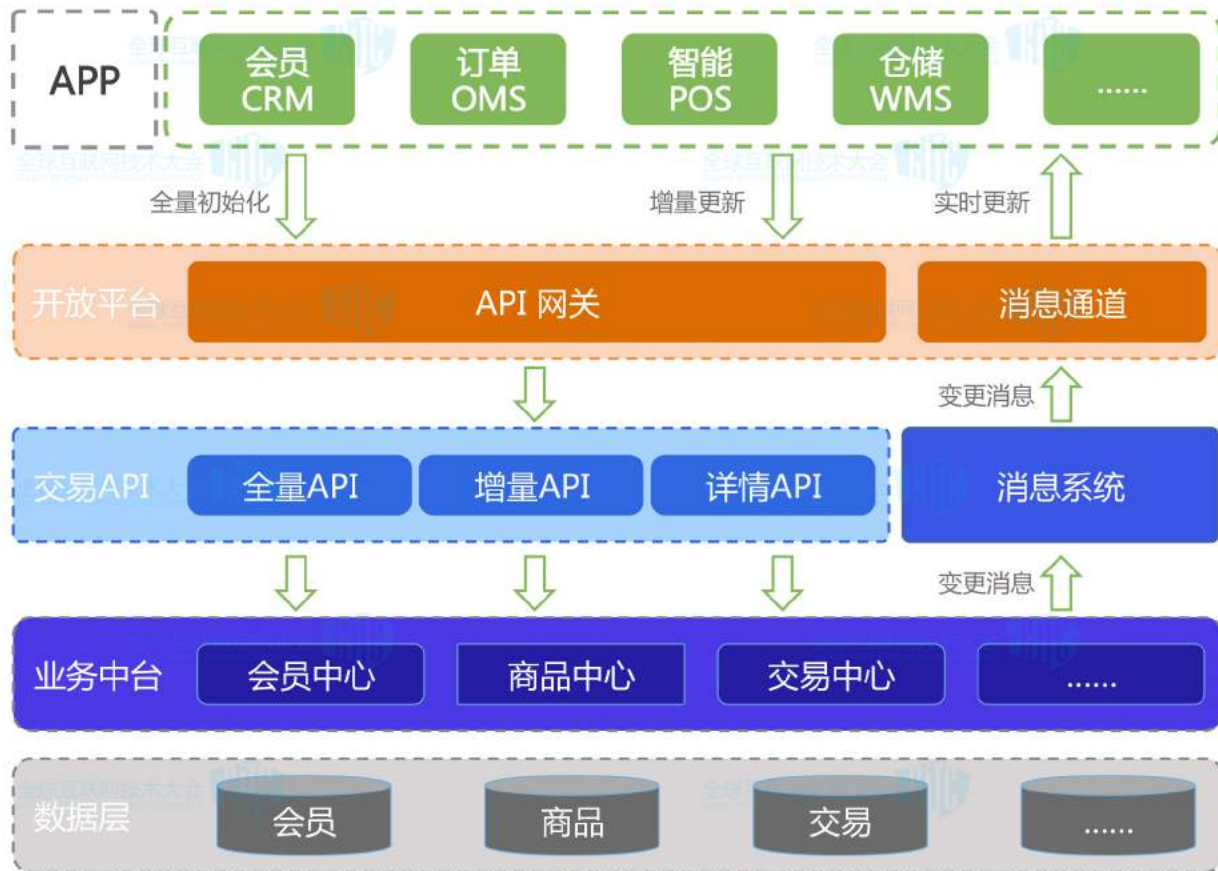


复杂的业务场景

# 淘宝公网订单同步的诉求



# 基于公网API的订单同步方案



## 全量数据同步：

全量API + 详情API

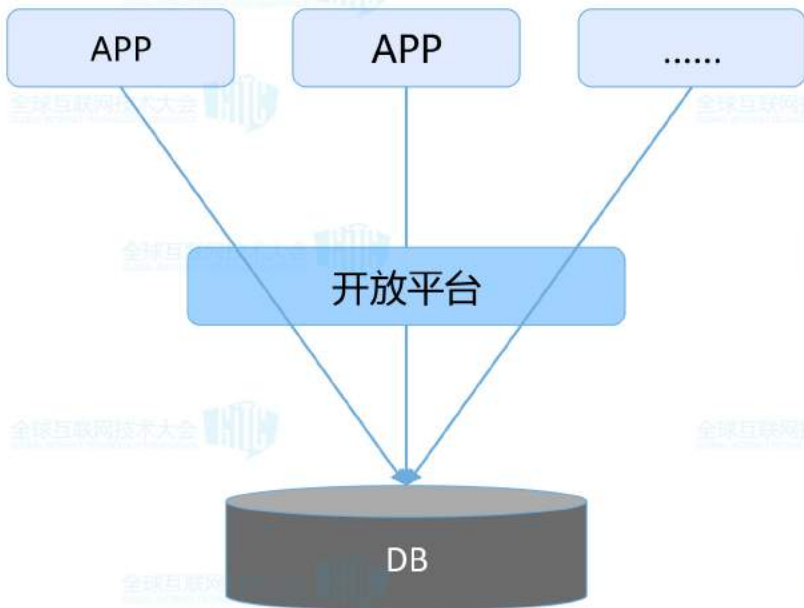
## 增量数据同步：

增量API + 详情API

## 实时数据同步：

消息通知 + 详情API

# 基于公网API的订单同步方案 —— 存在的问题



## 问题：

分页处理不当引起数据不一致

热点数据查询DB超时

交易异地多活，DB同步延迟

大促交易接口限流

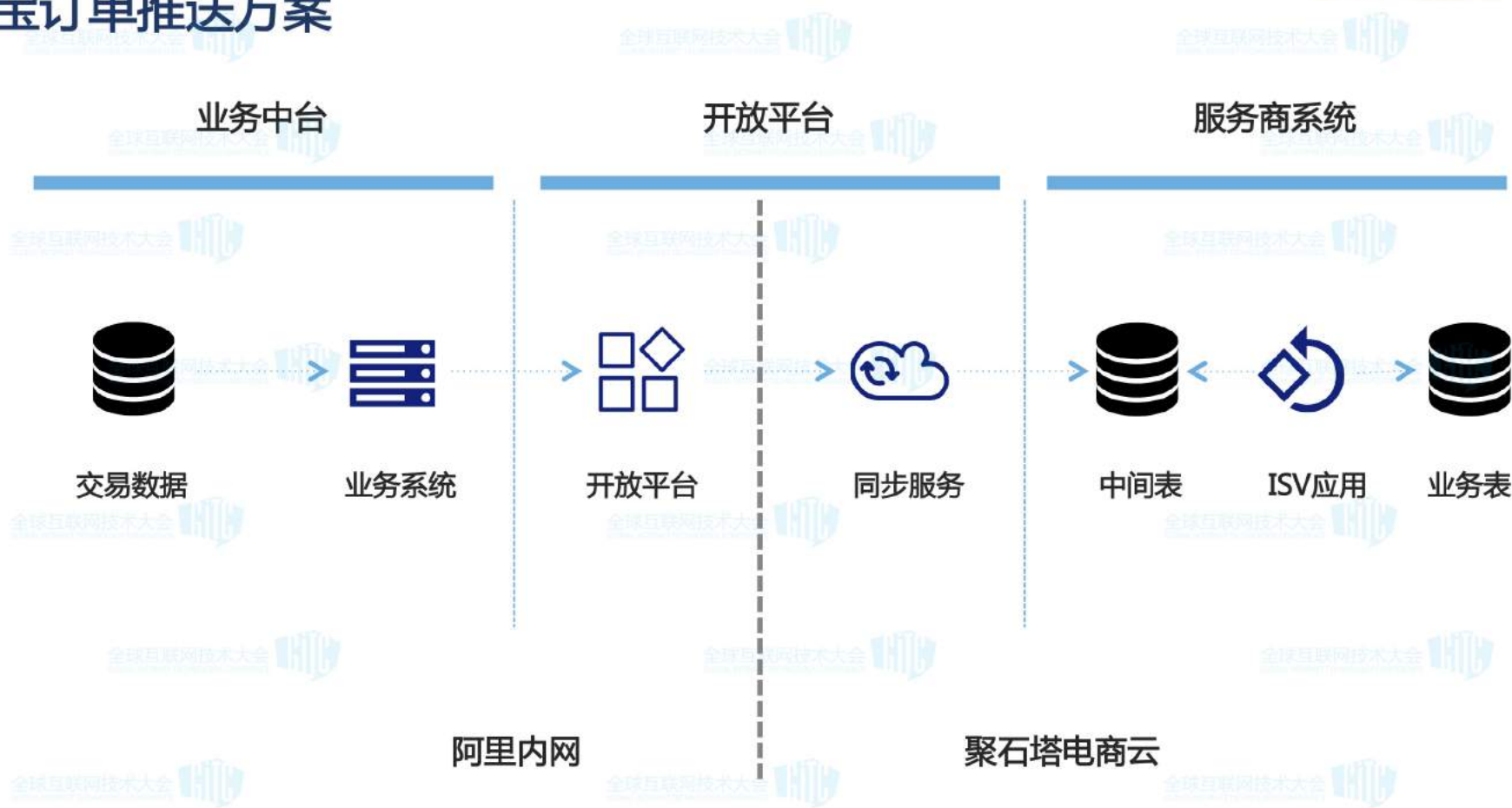


技术成本高，重复建设



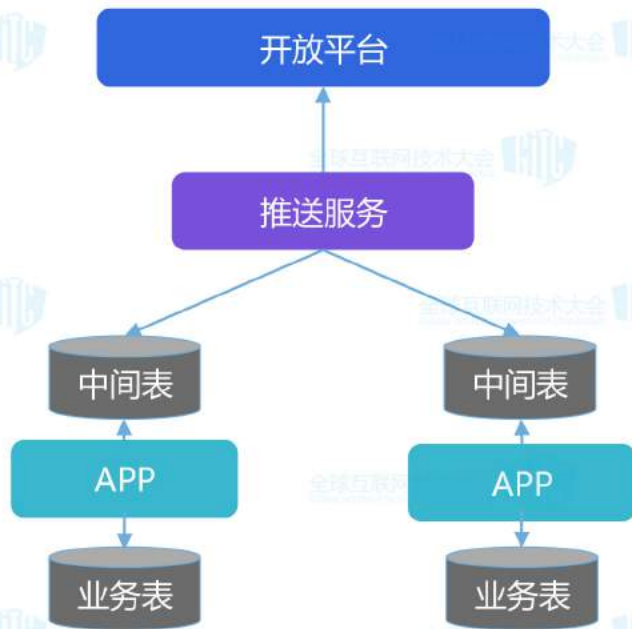
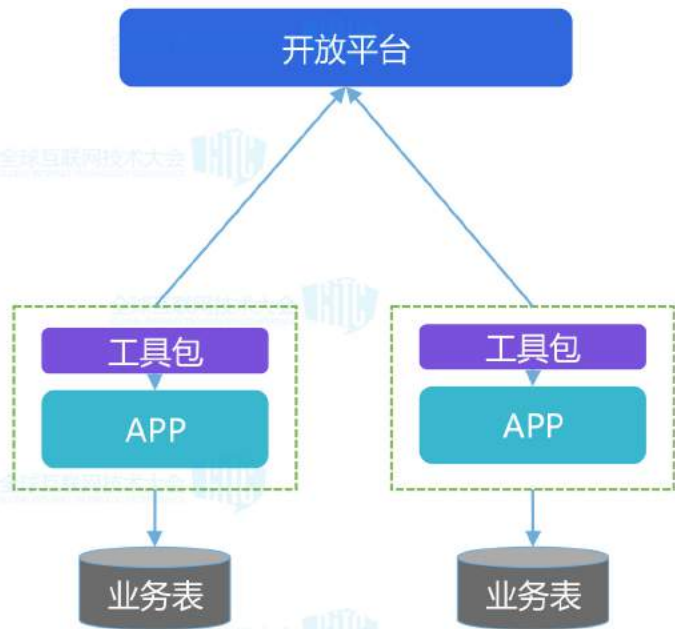
订单延迟，丢单

# 淘宝订单推送方案





# 订单推送服务1.0&2.0



维护成本高：问题排查，版本升级，开发者接入等

集中式部署：RDS连接数，资源隔离等

# 订单推送服务 —— 中间表设计

## 关键字段

业务核心字段，用于查询过滤

## 系统字段

jdp\_modified避免分页查询漏单

jdp\_hashcode乐观锁，减少DB操作

## 大字段

详情API返回数据(JSON)

后台配置需要返回的字段

只需要一份SqlMap

避免DDL锁表影响业务

关键字段

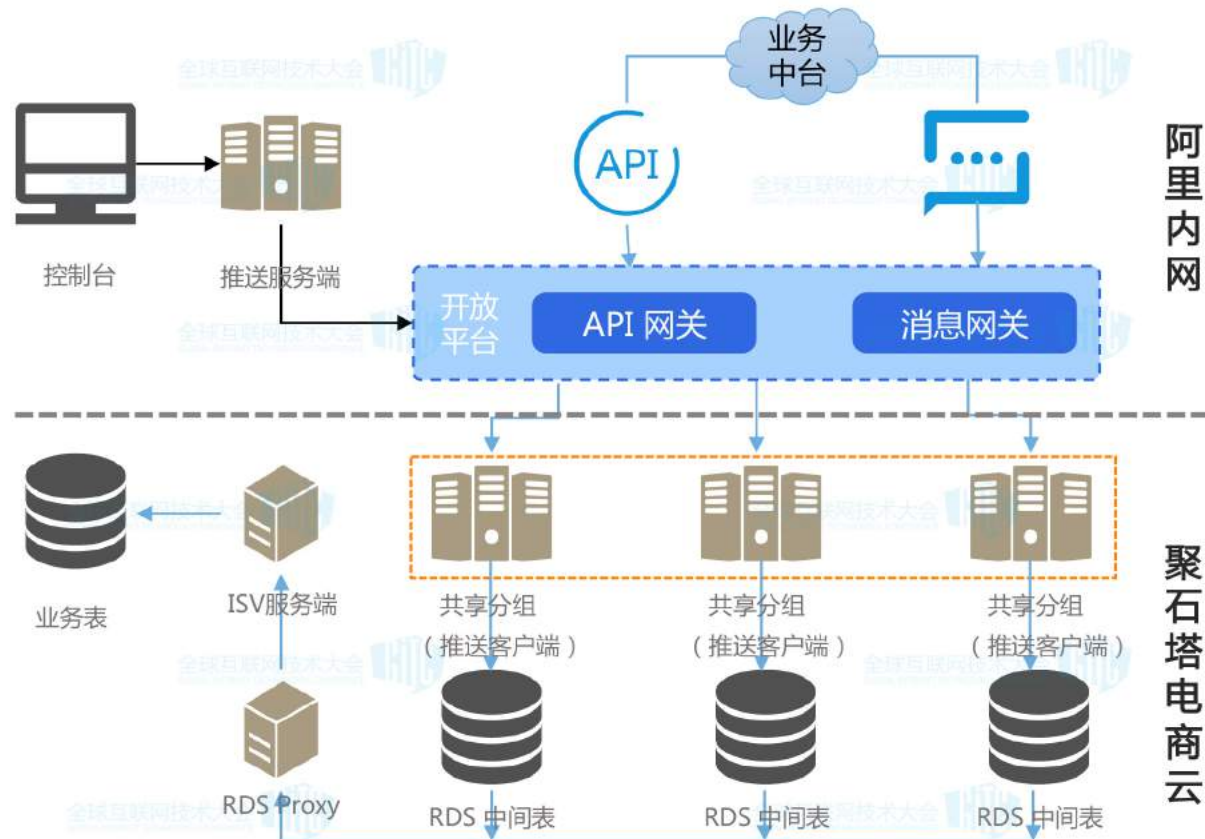
系统字段

大字段

名称	类型	是否索引	说明
tid	NUMBER	Y	交易ID
status	VARCHAR	Y	交易状态
type	VARCHAR	Y	交易类型
seller_nick	VARCHAR	Y	卖家昵称
buyer_nick	VARCHAR		买家昵称
created	DATETIME	Y	交易创建时间
modified	DATETIME	Y	交易修改时间
jdp_created	DATETIME	Y	数据入库时间
jdp_modified	DATETIME	Y	数据最后修改时间
jdp_hashcode	VARCHAR		用来做数据校验的字段
jdp_response	MEDIUMTEXT		API返回的整个JSON BODY



# 订单推送服务3.0



## 共享分组 + 资源隔离

按应用纬度分组  
多机房容灾  
独享RDS

## 消息网关

数据实时同步  
降低轮训DB的压力

## API网关

订单详情数据  
提供对单服务

# 订单推送服务 —— API对单任务

## 为什么需要对单？

源消息缺失

用户授权失效补数据

RDS不可用导致消息过期

.....

## 对系统的要求

商家数大，提高作业效率

后端资源和用户RDS资源有限，如何降低资源的使用

## 如何对单？

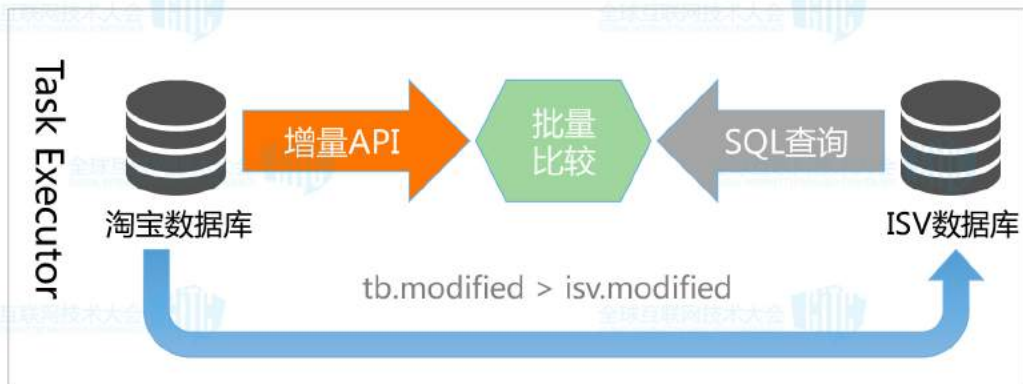
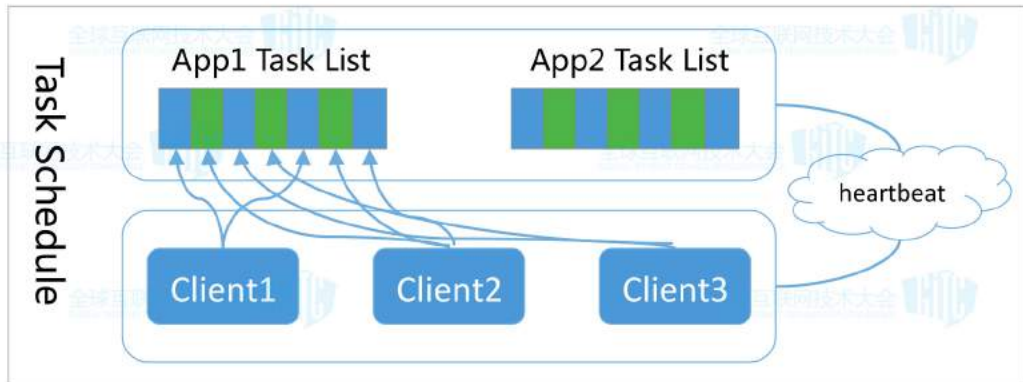
分布式任务调度，多机器对账

每隔15分钟进行一次对账，平衡业务的压力

断点继续

大卖家自动切片，避免DB超时

数据批量比较，减少RDS查询次数



# 订单推送服务3.0 —— 资源隔离

## 消息隔离

每个应用独立一个逻辑队列

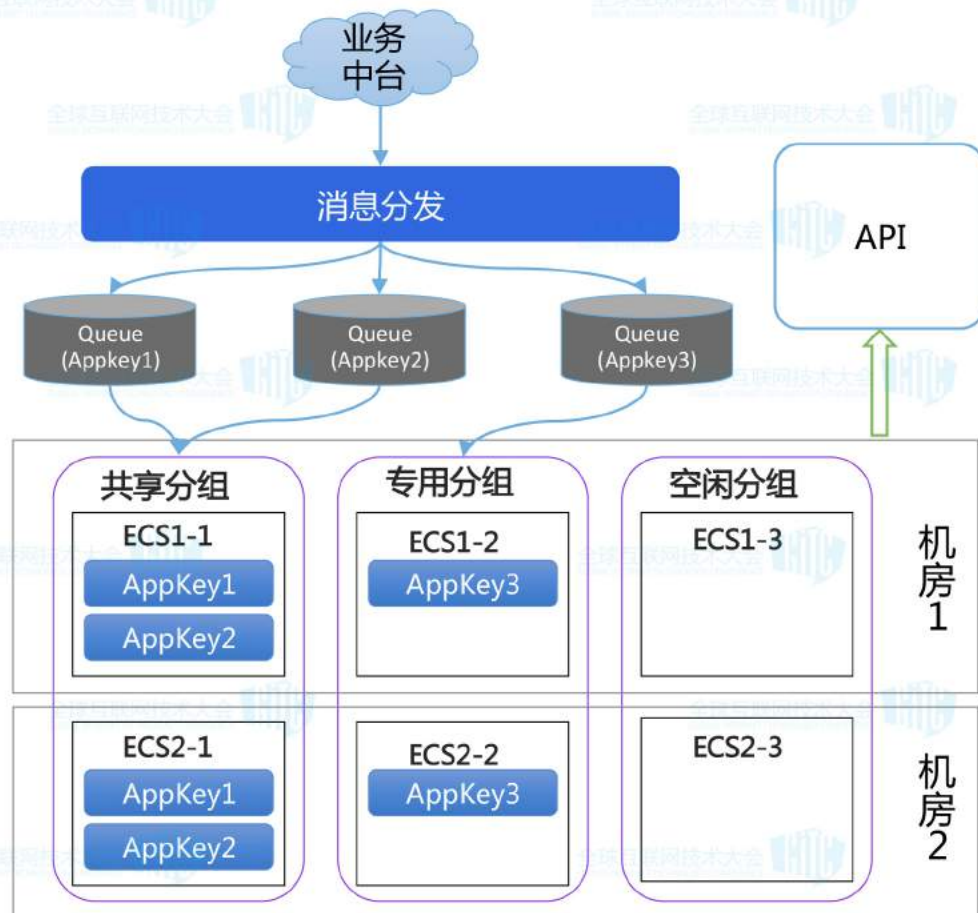
## ECS资源隔离

每个应用只会分配到一个分组

## 系统资源隔离

消息处理线程，对单任务线程，API处理线程，RDS连接池

## 慢RDS隔离



# 订单推送服务3.0 —— 存在的问题



交易API流量

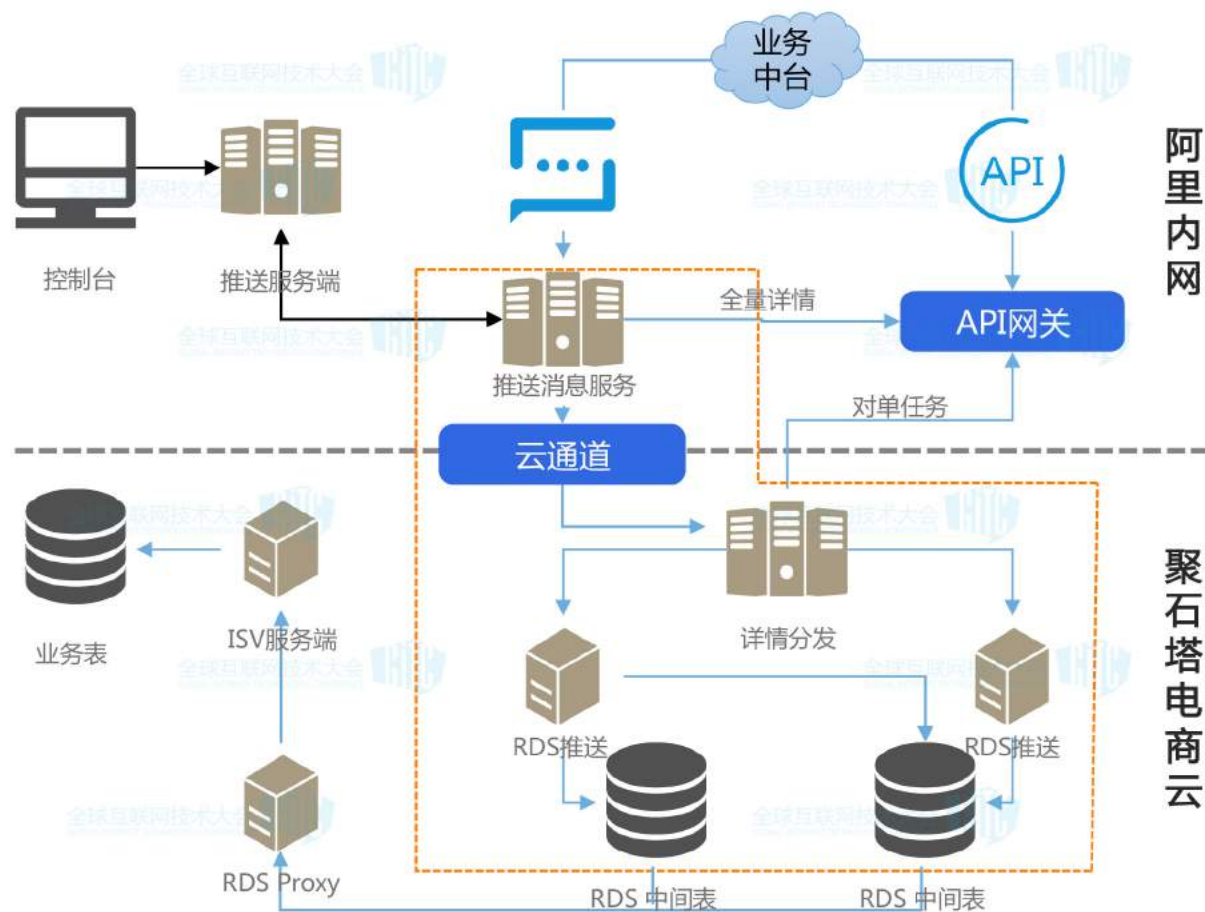


单APP处理性能



资源成本

# 订单推送服务4.0



## 推送消息服务端

接收订单变更消息  
获取交易全量详情  
寻找路由关系推送到聚石塔服务端

## 云通道

内外双向的RPC通讯

## 分发详情

订单数据分发

## RDS推送

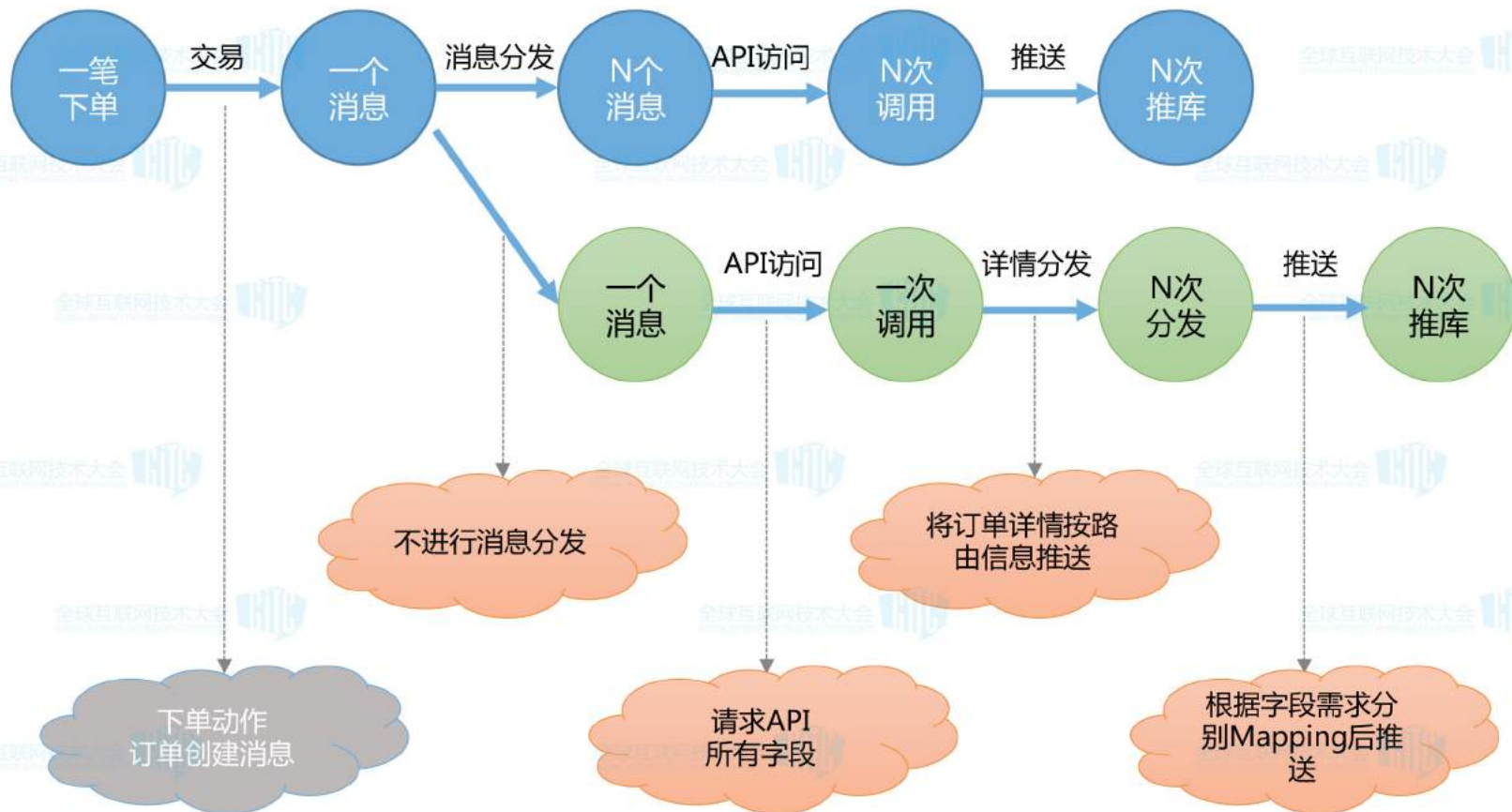
负责RDS数据写入

## API网关

对单任务



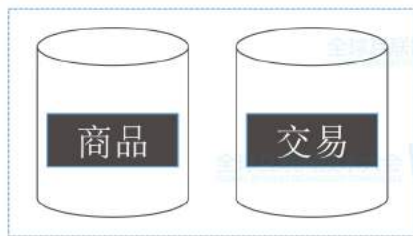
## 订单推送服务4.0 —— API详情复用



# 订单推送服务4.0 —— 处理效率



Queue  
Appkey



Queue  
Message Type

## 消息处理

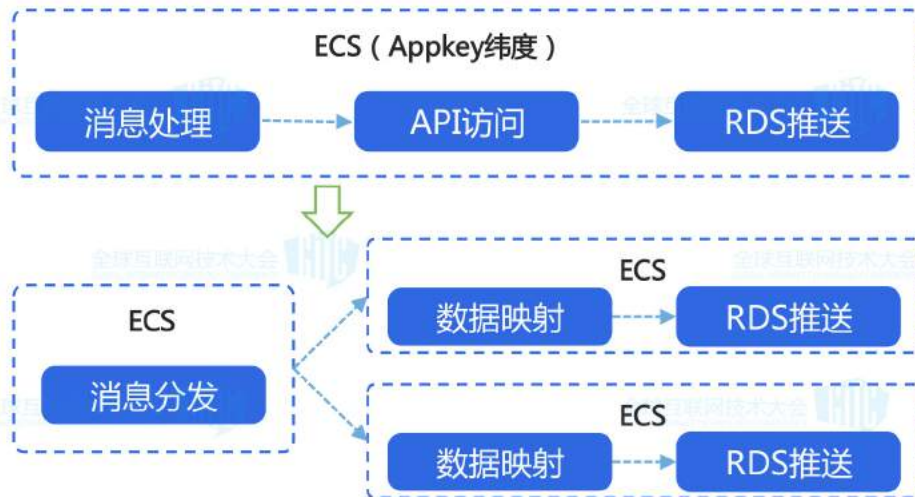
按应用纬度消息处理改造成按照消息类型处理  
商品和交易隔离处理，互不影响  
可优先处理交易消息

## 最小任务单元

提升机器利用率

解决APP集群瓶颈问题

RDS压力大，不可用不再对APP集群影响



# 订单推送服务4.0 —— 资源成本



硬件

消息分发：消息不再按订阅APP分发，消息系统TPS呈百万级降低  
最小任务单元：单个ECS资源利用率最大化

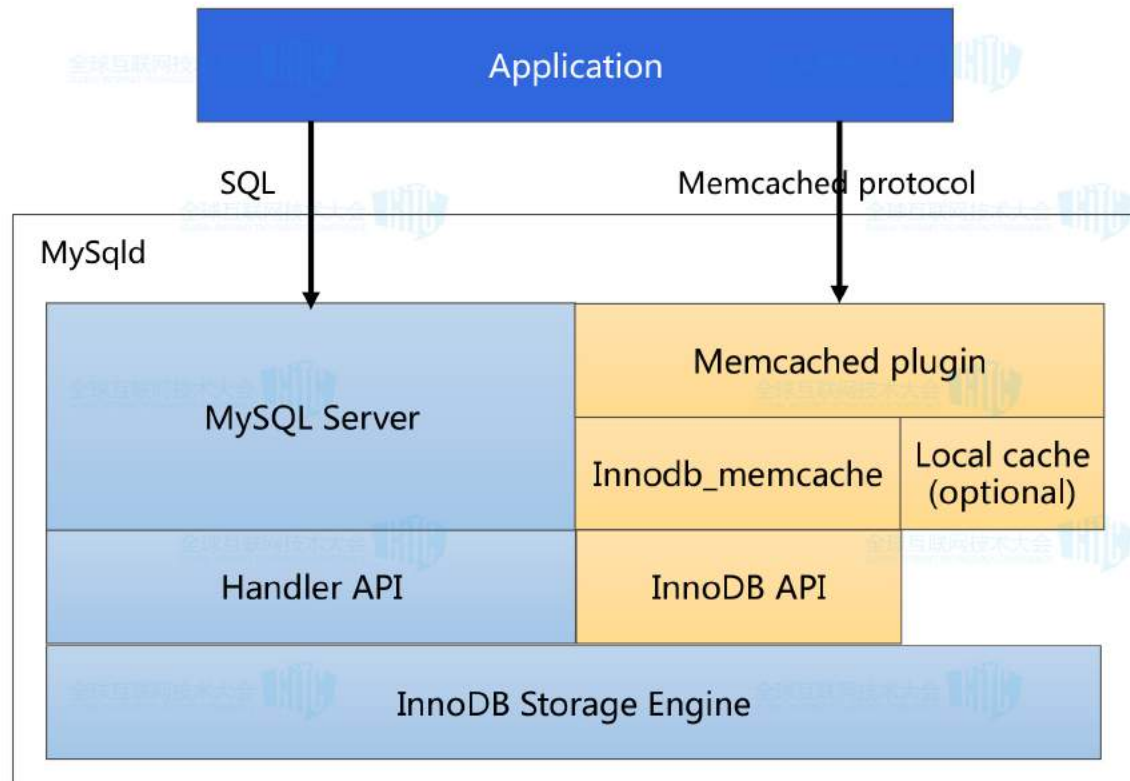


带宽

API详情复用，一次变更消息只存在一次交易详情的公网访问

# 交易API优化

## Mysql Memcached Plugin



## 业务代码优化

本地缓存

最小访问原则

跨单元合并查询

分页查询使用UseHasNext

# 订单履行链路监控

## 为什么需要链路监控？

订单处理链路长

某一环节出问题，不能快速定位

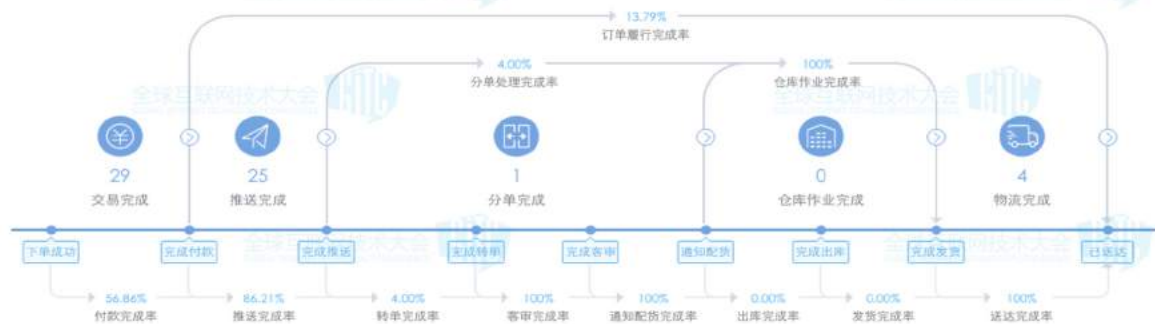
## 监控

订单处理效率

订单完成率

单笔订单链路详情

今日完成率



订单查询

订单ID: 33470848800000000

查询

订单详情

Id: 33470848800000000

商家昵称: 测试官方旗舰店

订单状态







更多阿里技术干货  
关注“阿里技术”官方微信公众号

# Thanks



更多技术交流