

# 美团配送：从零到千万量级的实 时物流平台架构实践

@郑跃强

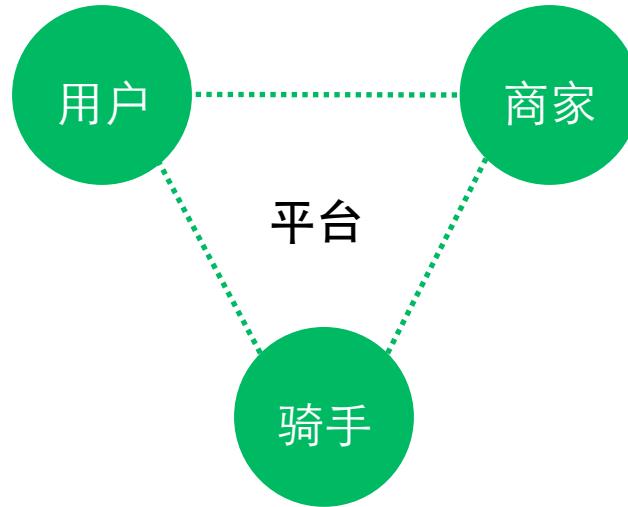
# 个人简介



2011年中科大 研究生毕业



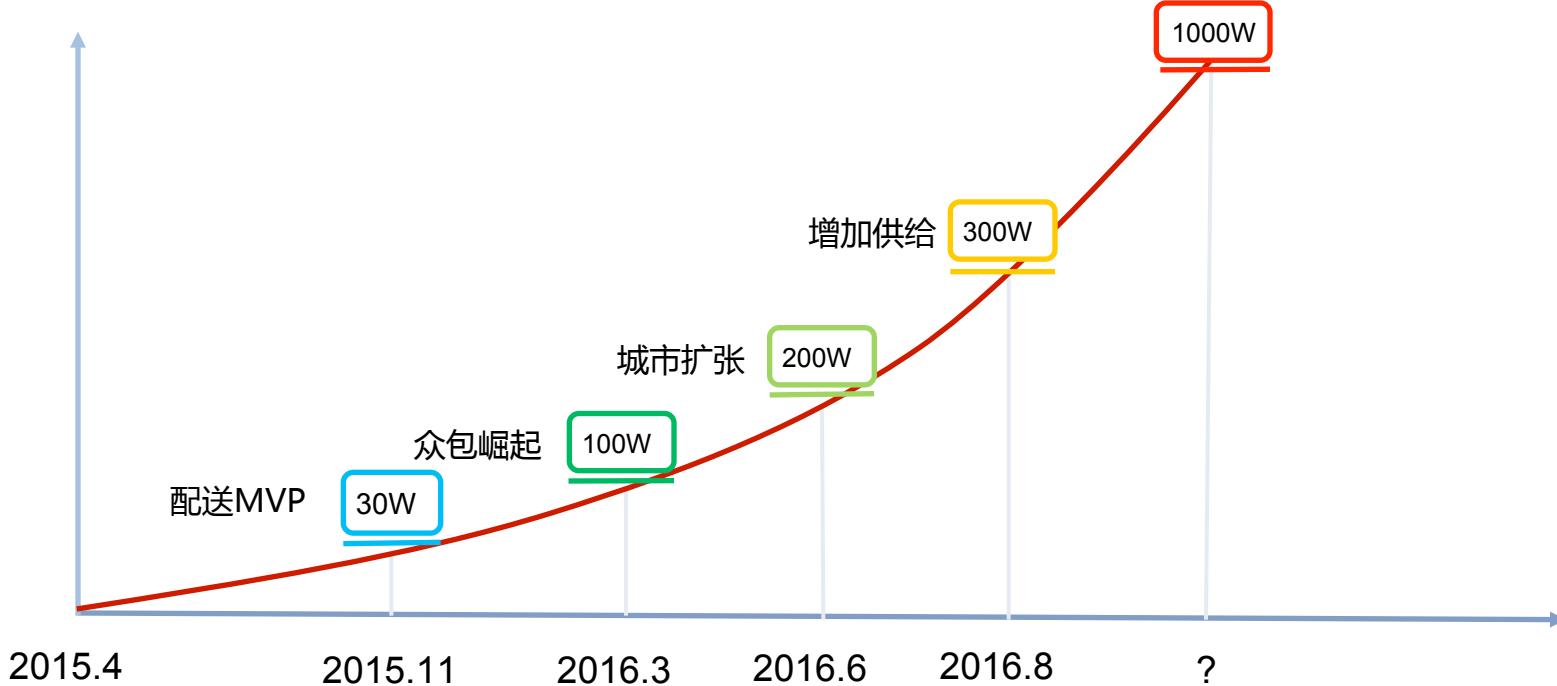
- 配送业务发展历程
- 技术架构演进介绍
- 系统稳定性面临的挑战
- 我们的解决之道



- 四方利益
- 三条业务线

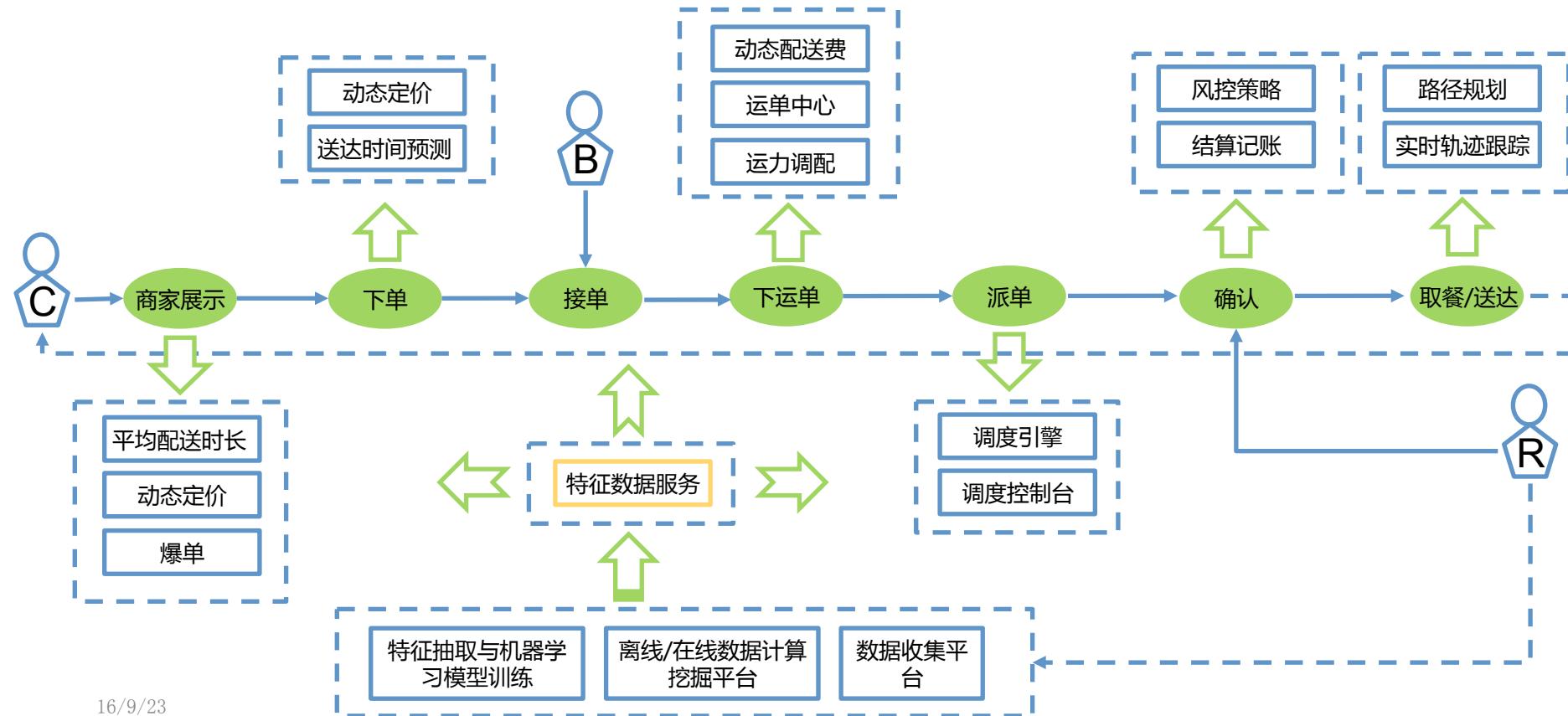
- ✓ 自营
- ✓ 加盟
- ✓ 众包

# 配送业务发展历程



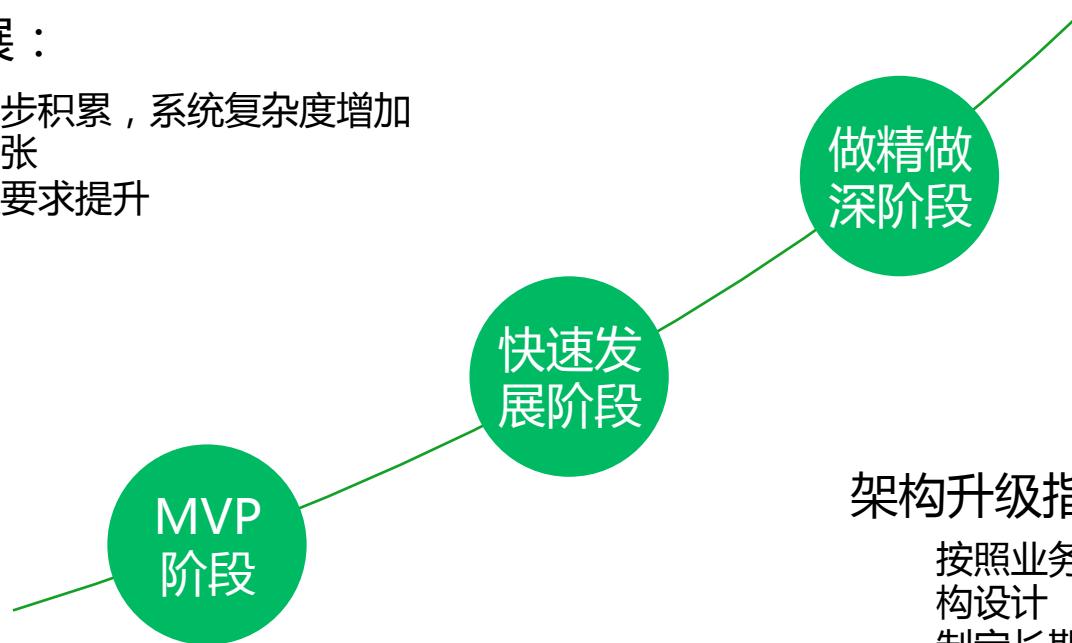
- 配送业务发展历程
- 技术架构演进介绍
- 系统稳定性面临的挑战
- 我们的解决之道

# 配送业务流程下的技术应用概览



## 业务发展：

需求逐步积累，系统复杂度增加  
团队扩张  
稳定性要求提升



## 架构升级指导原则：

按照业务发展阶段选择合适的架构设计  
制定长期的架构升级路线，逐步实施，小步迭代

## 业务特点：

- 业务复杂度低，快速验证业务
- 团队人员较少
- 需求稳定性差

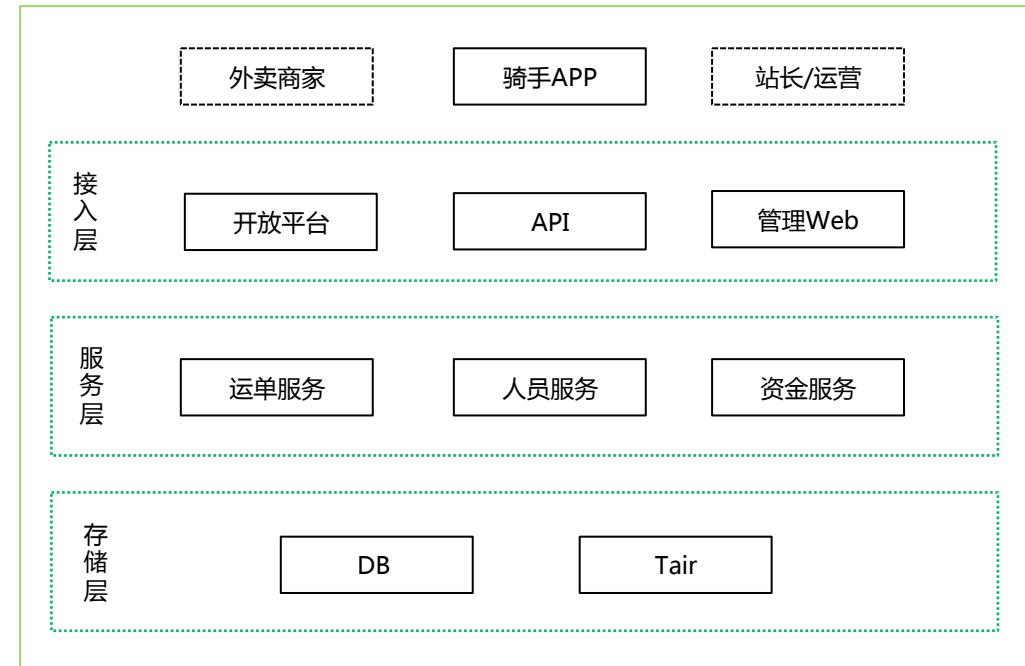
## 架构设计：

- 系统层面解决有无问题
- 快速搭建验证业务的最小系统

# 配送架构演进-配送1.0



- 简单分层
- 2周内搭建出可用版本



## 业务特点：

- 需求迭代速度快，业务越来越复杂
  - 一个模块功能越来越多，越来越难维护
- 团队扩张，细分小团队
  - 多个团队在一个模块上面开发影响效率
- 对系统稳定性要求越来越高
  - 业务规模变大，对系统稳定性要求变高

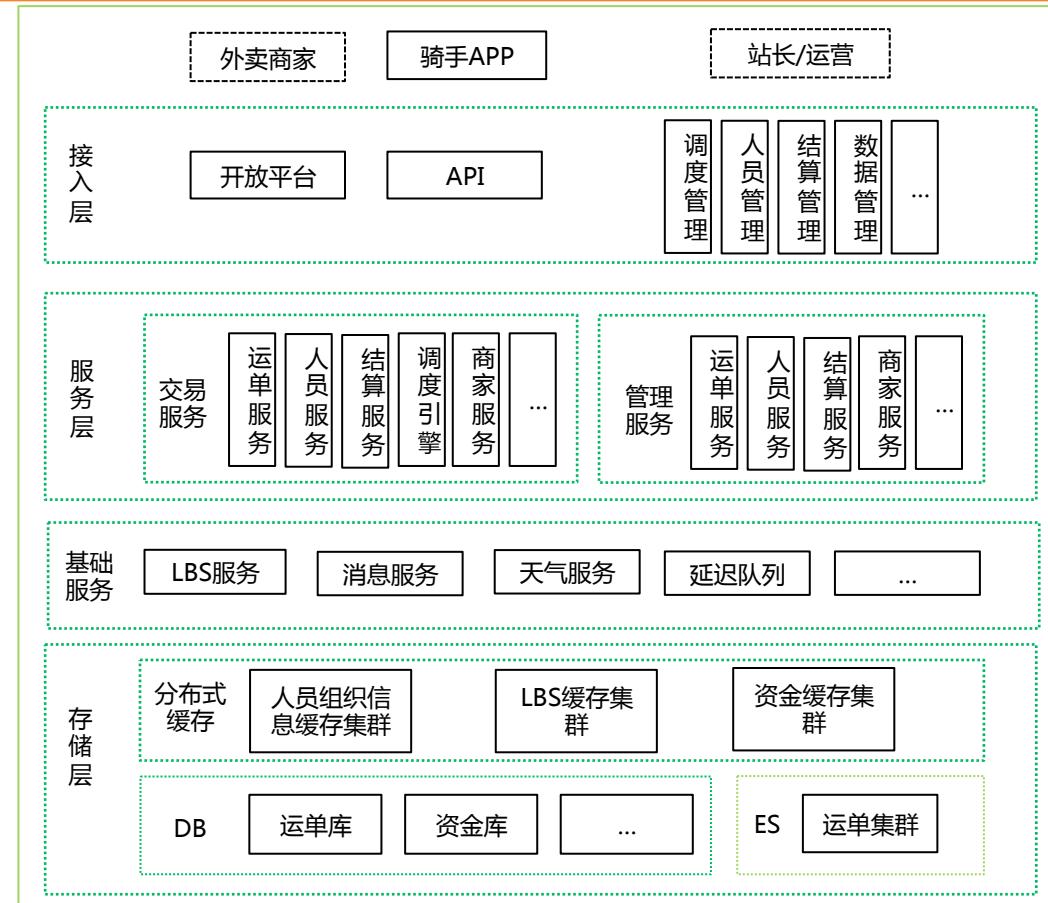
## 架构设计：

- 拆分：按业务拆分模块
- 复用：抽取公共模块、业务服务和基础服务分离、组件化
- 解耦：消息中间件等
- 容量相关架构升级
  - 缓存
  - DB等

# 配送架构演进-演进2.0



- 业务垂直拆分
  - 交易服务和管理服务拆开
  - 公共基础服务抽取
- 
- 分布式缓存：按业务拆分
  - DB：按业务拆分
  - 搜索引擎



## 业务特点：

- 按业务域粒度划分服务越来越臃肿
- 某些功能有平台化需求
- 对稳定性要求更高

## 架构设计：

- 大系统小做
  - 高频、核心模块拆分独立服务
- 业务模块分层
  - 业务服务层 + 基础业务服务层
- 自动化运维
  - 配送服务应急平台
- 容量相关架构升级
  - 数据资源
  - 计算资源

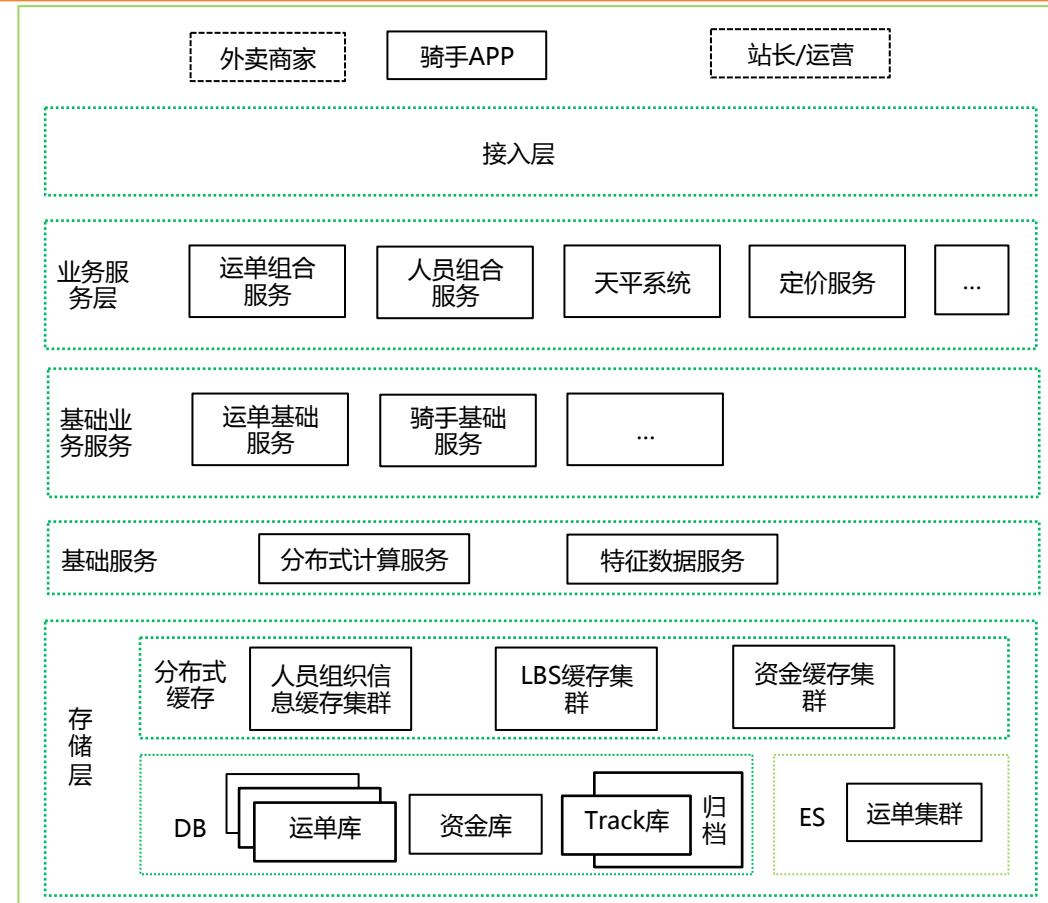
# 配送架构演进-演进3.0



- 服务层拆分：基础业务服务层、业务服务层
- 大系统小做：高频、核心模块单独成服务

单一服务职责

- DB：散库散表 / 归档
- 计算瓶颈：分布式计算服务



背景：

- 日运单：150W+
- 主库写QPS：1.1K+
- 主库读QPS：10K+
- 单表规模：1.5亿+

目标：为了支撑更高的单量(500w)

解决方法：分库分表

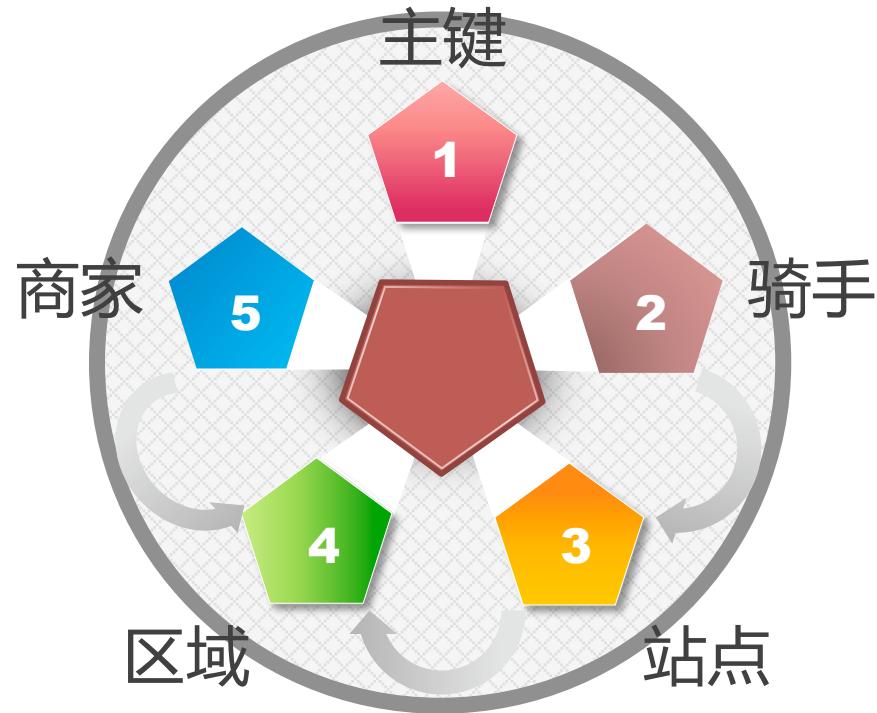
## 难点分析：

- 查询场景多且复杂
- 众包和专送两种配送业务模式
- 管理端存在复合及关联查询
- 数据一致性问题

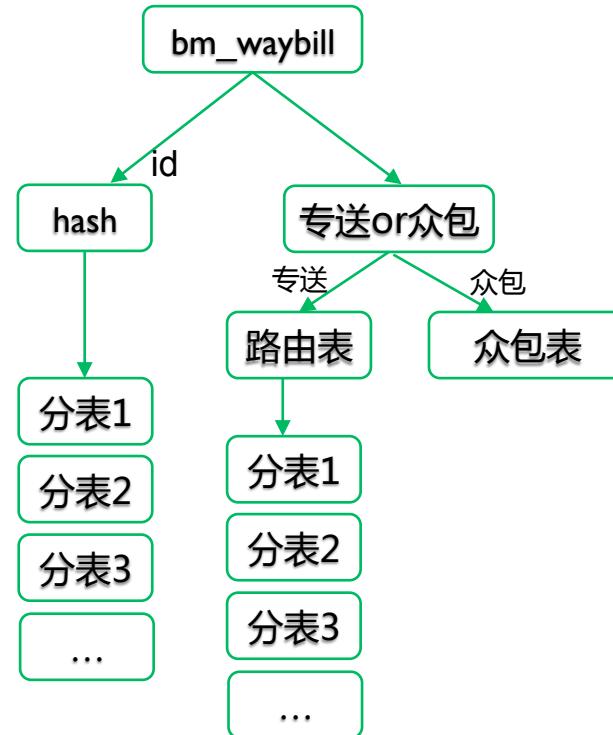
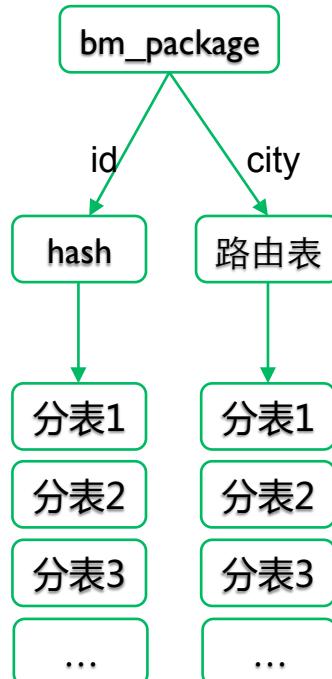
## 拆表原则：

- 维度越少越好
- 支撑各个场景查询

维度选择？



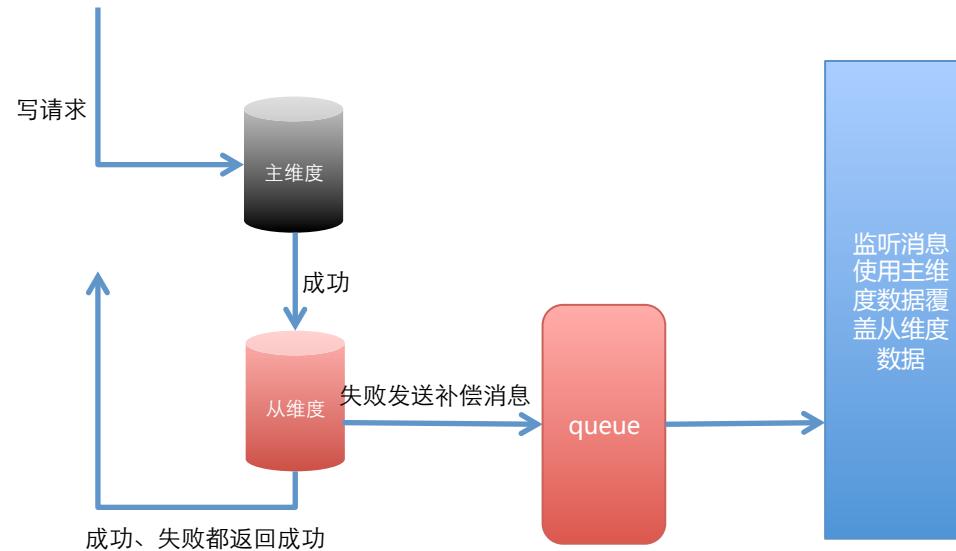
## 如何划分维度？



## 路由表如果解决数据均匀问题？

- 热点城市单独一张表
- 非热点数据通过订单分布数据手动指定

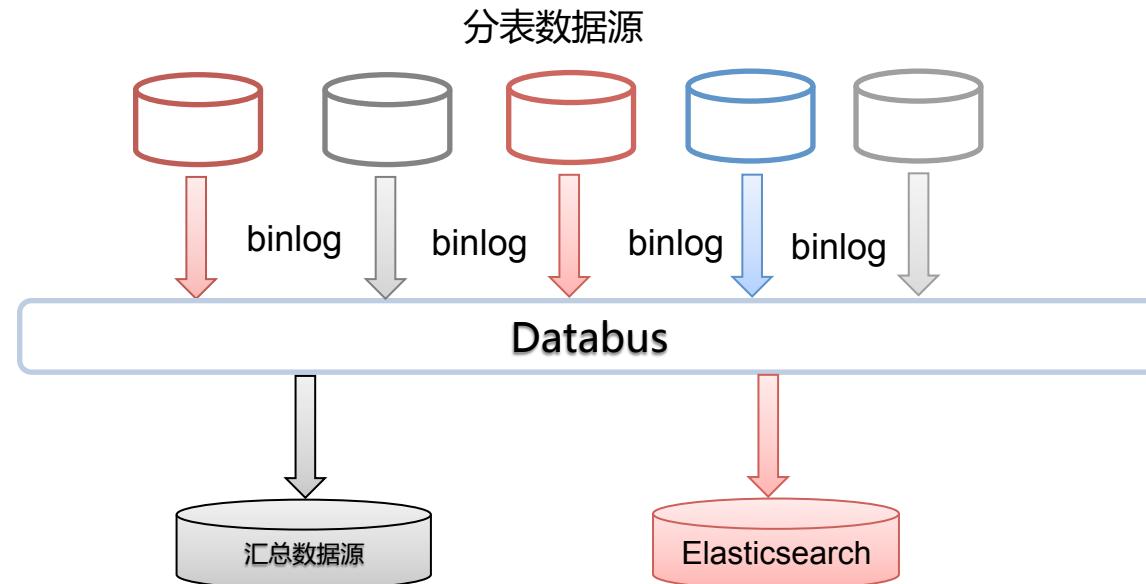
## 如何解决多个维度数据一致性问题？



## 如何解决管理端复杂查询？

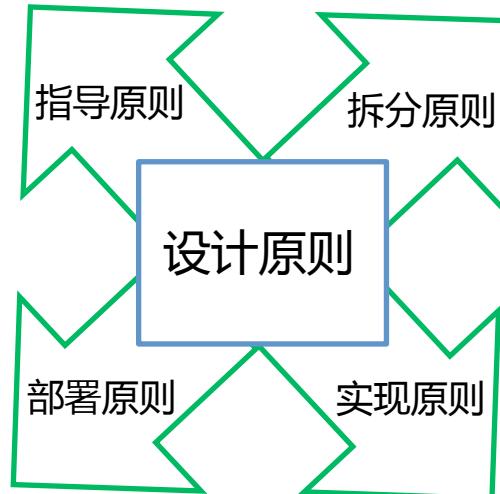
数据异构：

- 索引异构
- 数据异构



# 架构设计总结-设计原则

- 一切以稳定为中心
- 不过度设计
- 架构尽可能简单、清晰



- 按业务域拆分系统
- 按功能场景拆分系统
- 根据非功能需求拆分系统

- 服务可回滚
- 集群容错，避免单点
- 多机房部署

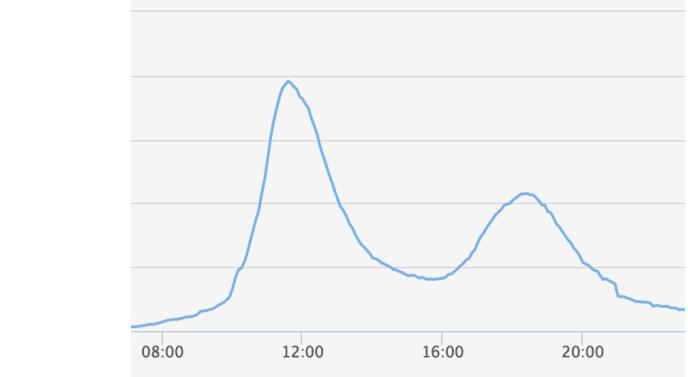
- 服务抽象化
- 低耦合
- 服务可治理

- 配送业务发展历程
- 技术架构演进介绍
- 系统稳定性面临的挑战
- 我们的解决之道

# 挑战

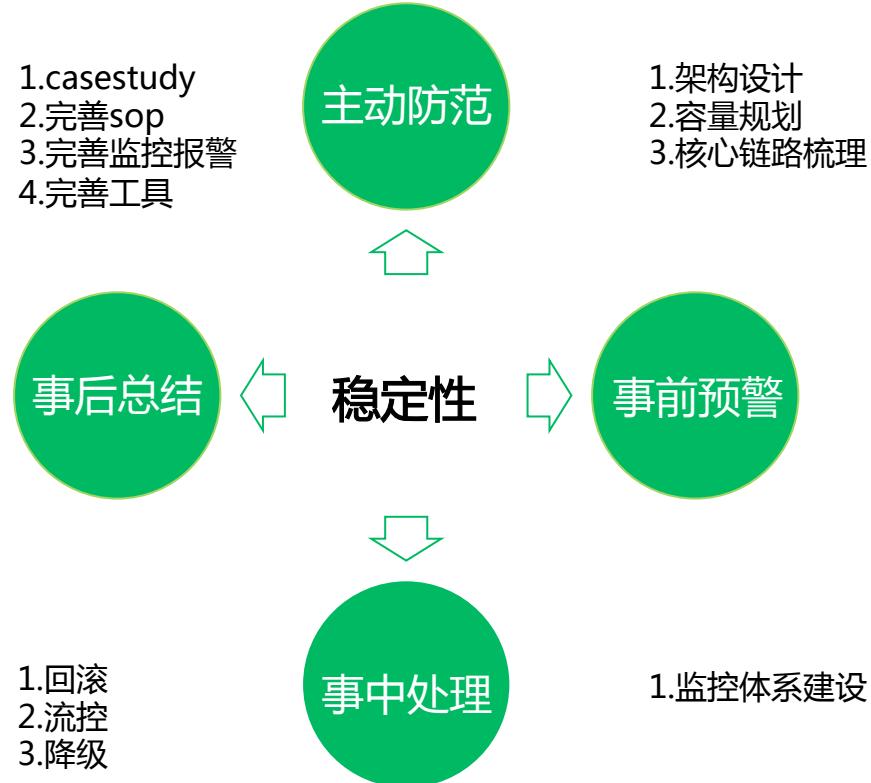


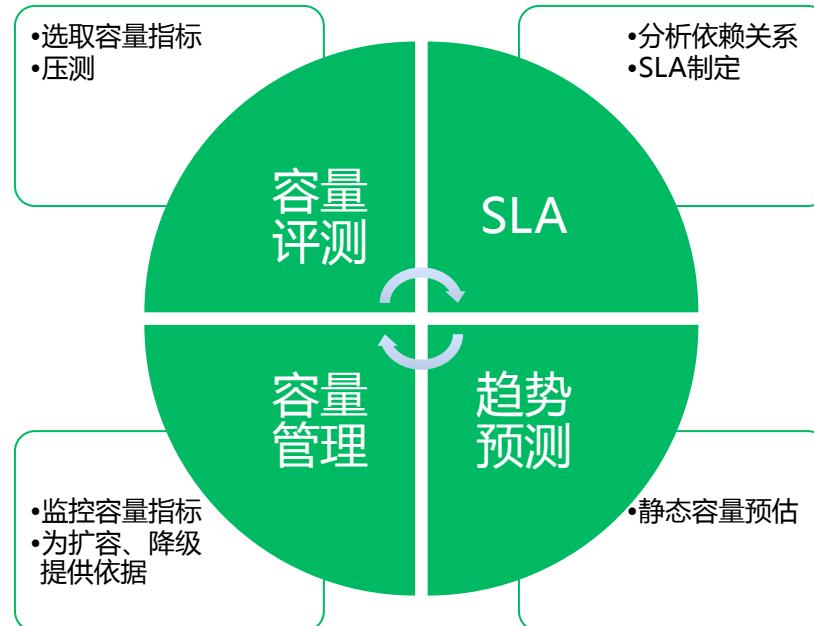
- 业务产品迭代快  
39个APP版本, 平均两个礼拜一个版本
- 业务发展迅猛
- 业务尖峰明细
- 业务时效性强  
一秒内派出200单
- 系统复杂性高



- 配送业务发展历程
- 技术架构演进介绍
- 系统稳定性面临的挑战
- 我们的解决之道

# 解决之道

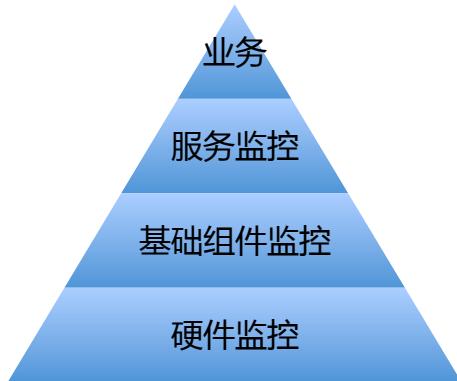




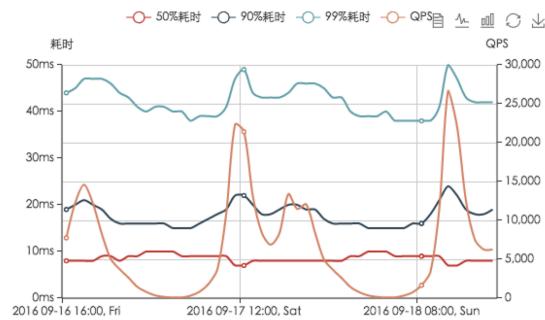
## 核心链路梳理 关键点

- 区分核心流程和非核心流程
  - 非核心流程可降级
- 依赖关系梳理
  - 跨业务域调用尽量异步
  - 同步调用要设置超时时间
  - 异步调用要设置队列长度
  - 非核心功能尽量异步、可降级

## 一、全面性 4层监控体系建设



- 业务指标监控：运单环比情况监控
- 服务监控：业务服务接口性能、JVM等
- 基础组件监控：DB、缓存、消息中间件等
- 硬件监控：CPU、网络、链接、IO等



## 二、准确性

### (1)个性化OVS设置

- cpu.busy 60%
- net.bytes.in 40M
- ...

### (2)业务监控

根据统计值来预测阈值

## 三、及时性

按不同报警等级选择不同的报警通道：短信 --> 大象 --> 邮件

## ➤ 回滚

线上问题90%是因为上线导致，所以首先要判断回滚是否可以解决问题

预计上线时间	2016-09-18 16:00
Code Review同事	<b>安全隐藏</b>
上线服务	fas  finance_account  保证线上已有方法能正常调用不被拦截器拦截即可
上线流程	1. 以下顺序没有绝对依赖关系 2. finance_account 3. finance_account client 4. fas
上线负责人	<b>安全隐藏</b>
回滚方案	fas 回滚到#630552d  fas client 回滚到 #e3f4eeb  account回滚到#0a27865  account_client回滚到#2190ce8
版本	
相关改动	配合QA增加自动化测试iface，并增加拦截器，本次修改的方法均不允许线上调用
知会的 QA/PM	
紧急程度	一般
自己测试代码逻辑	是

## ➤ 降级

使系统临时承担更大的流量压力

具体策略：

- 关闭影响性能非关键业务功能
- 屏蔽非关键业务的入口
- 页面静态化
- 使用缓存数据牺牲数据一致性
- 放弃后端的补偿机制以减少调用链时间等

➤ 流控  
人工流控

外卖平台

按业务线降级

配送平台

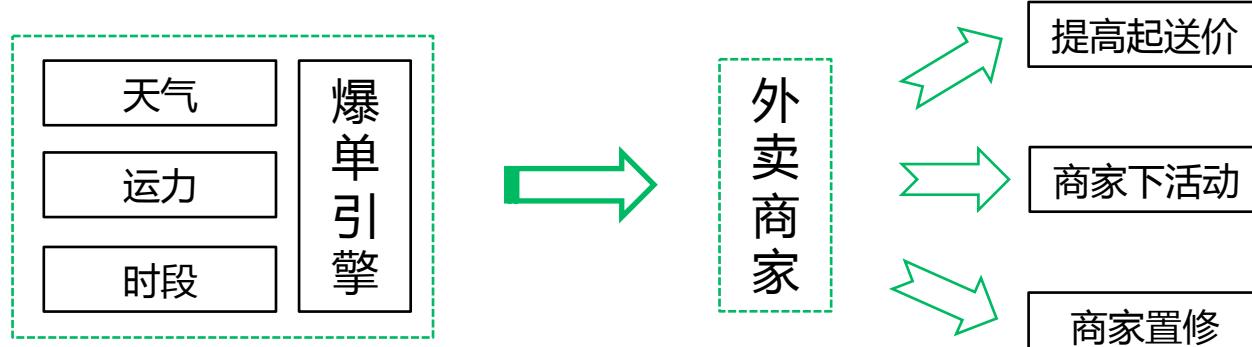
骑手限流



- 按类型
- 按城市
- 按区域
- 按骑手尾号

## ➤ 流控

自动流控 – 智能爆单



# 事后总结

## ➤ CaseStudy

5W2H(What、Who、Where、When、Why、How、How much)

## ➤ 完善SOP

标准化操作流程，缩短故障时间  
 标准化上线流程，避免故障发生

## ➤ 完善工具

工欲善其事，必先利其器

三 班马服务应急平台 <small>● 首页反馈</small>							欢迎你, <small>郭敬强</small>	
常用工具	降级管理	缓存降级	集群状态	监控预警	客户端工具	API工具	服务化工具	QA工具
<small>banma banma_banma_admin banma_banma_admin_dispatch banma_banma_api banma_banma_api_xiniu banma_banma_biz_proxy banma_banma_dbus_sync banma_banma_open banma_banma_service_auth_server banma_banma_service_business_server banma_banma_service_cache_server banma_banma_service_callback_server</small>								
<small>banma_banma_service_data_server banma_banma_service_deliveryarea_admin_server banma_banma_service_deliveryarea_server banma_banma_service_dispatch_engine banma_banma_service_dispatch_galaxy_server banma_banma_service_doctor_server banma_banma_service_finance_admin_server banma_banma_service_finance_pool_server</small>								
<small>banma_banma_service_finance_server banma_banma_service_lbs_server banma_banma_service_message_server banma_banma_service_operation_admin_server banma_banma_service_operation_server banma_banma_service_package_admin_server banma_banma_service_package_server banma_banma_service_package_sync_server banma_banma_service_prediction_server</small>								
<small>banma_banma_service_rider_admin_server banma_banma_service_rider_server banma_banma_service_router_server banma_banma_service_staff_admin_server banma_banma_service_staff_server banma_banma_service_upstream_server banma_banma_service_weather_admin_server banma_banma_task</small>								
降级级别	降级分类	功能类别	降级状态	降级开关				
全部	全部	全部	全部	全部				
总计3条记录 <small>● 搜索</small>								
序号	降级开关	降级级别	所属项目	功能类别	业务描述	降级影响	当前状态	
1	tair_degrade_to_medis	二级降级(影响可接受)	banma	FUNC_KV	部分人员组织接口从tair切换到redis	部分人员组织接口从tair切换到redis	<small>未降级</small>	<small>恢复</small> <small>回滚</small>
2	kv_to_db_degrade	二级降级(影响可接受)	banma	FUNC_KV	部分人员组织接口在读cache失败后回滚到db	部分人员组织接口在读cache失败后回滚到db，降级后，回滚到db将受影响	<small>未降级</small>	<small>恢复</small> <small>回滚</small>
3	tair_fail_read_medis	三级降级(影响较小)	banma	FUNC_KV	tair失败后读redis降级	降级后，读tair失败将不会读redis	<small>未降级</small>	<small>恢复</small> <small>回滚</small>



谢谢聆听！