

项目研发管理

经验总结

袁剑 2020年

目录

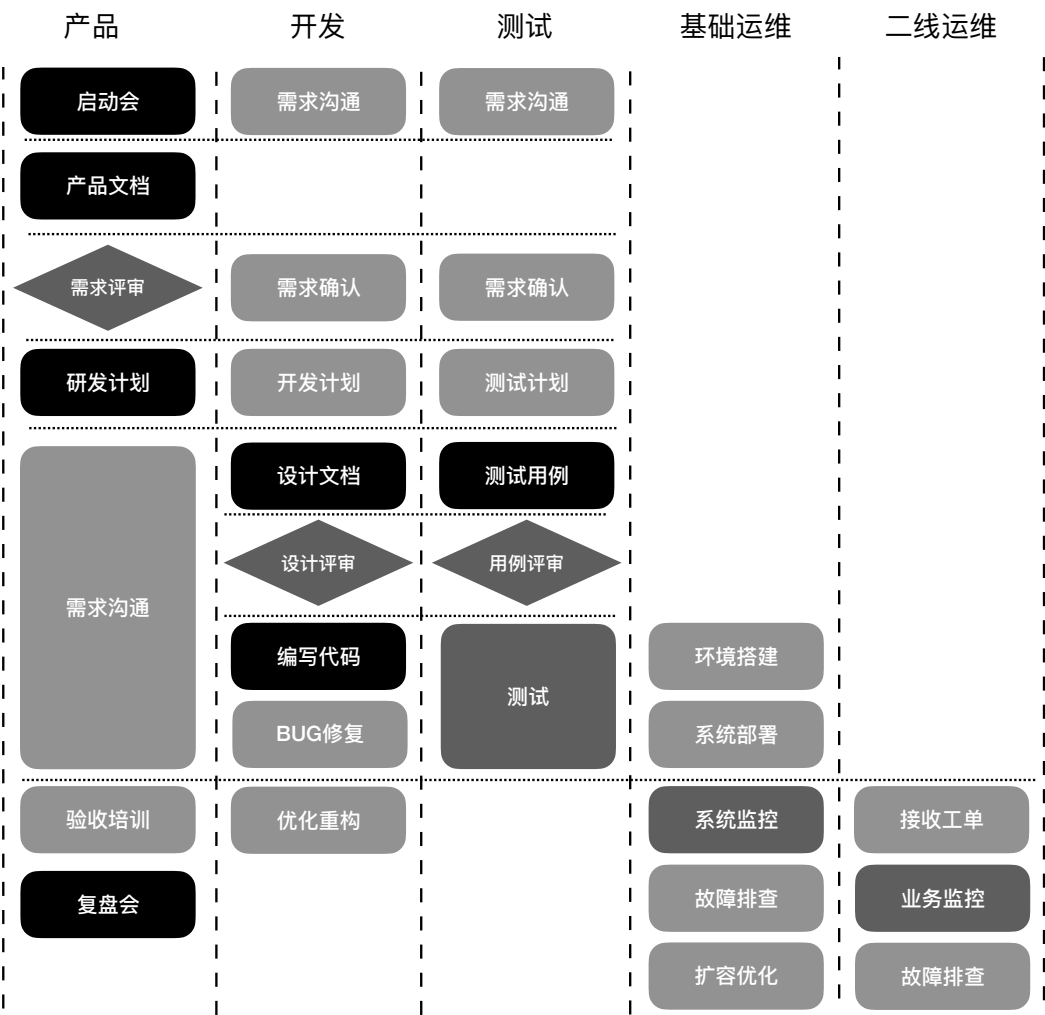
- 质量保证
- 进度把控
- 问题应对

质量保证

- 流程
 - 研发流程，上线流程
- 规范
 - 编码规范，源码管理规范
- 架构
 - 开发框架和技术栈
- 检查和监控

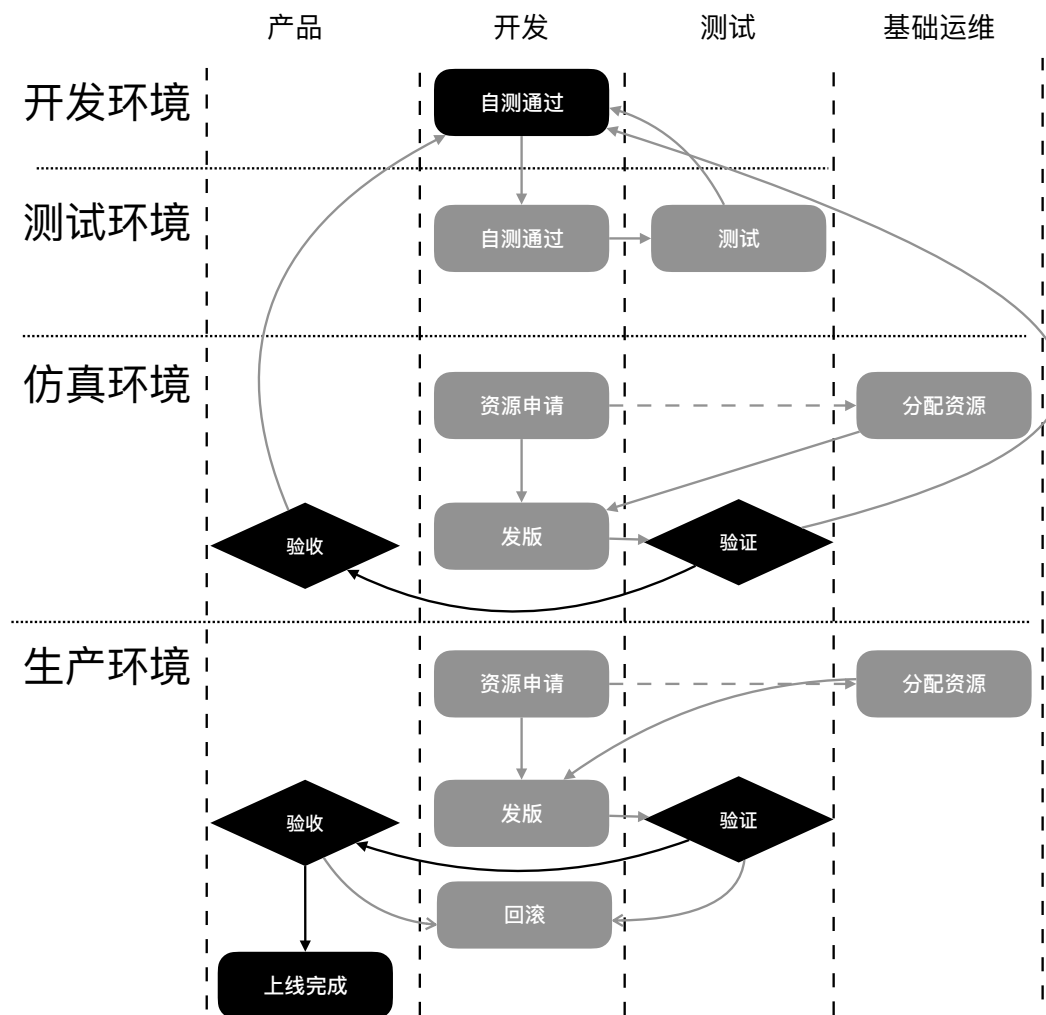
质量保证 - 研发流程

- 目的：明确岗位职责
- 重点：
 - 需求评审
 - 研发计划
 - 设计评审



质量保证 - 上线流程

- 目的：防范系统上线风险，确保系统稳定
- 重点：
 - 自测通过
 - 仿真和生产环境双重验证



质量保证 - 编码规范

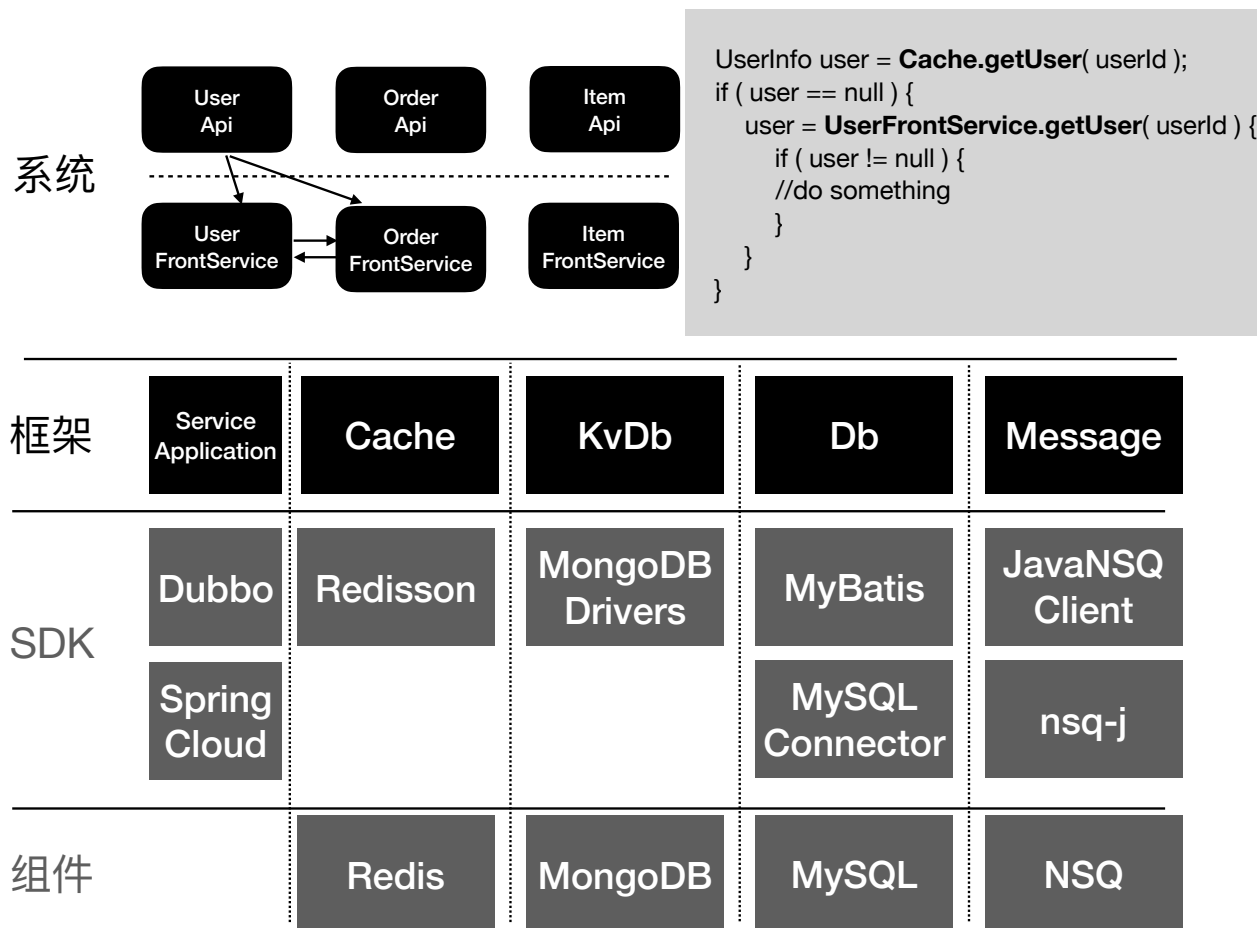
- 目的：提高协作效率
- 重点：
 - 命名：见名知意，遵循语言规范
 - 注释：有意义的注释，每个public方法都应该有
 - 类：职责单一
 - 方法：只做一件事
 - DRY, KISS, SOLID, 设计模式

质量保证 - 源码管理规范

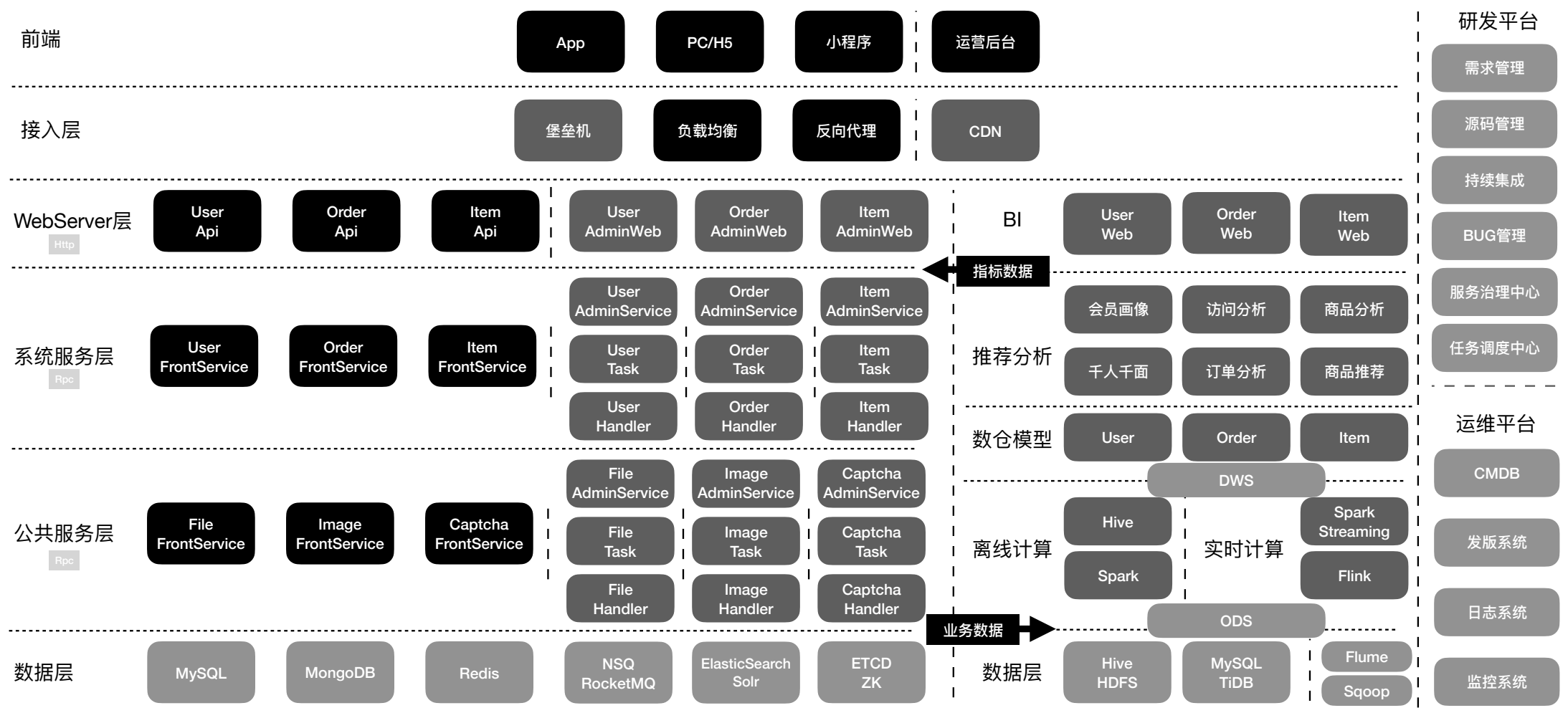
- 目的：提高研发效率
- 重点：
 - 环境：开发（Dev），测试（QA），仿真（STG），生产（PRD）
 - 工具：代码管理（Gitlab），Maven私服（Nexus），持续集成（Jenkins）
 - 分支管理：master（上线），feature（新功能/BUG），release（发布），hotfix（紧急修复）

质量保证 - 开发框架

- 目的：提高开发效率
- 重点：
 - 选择熟悉的组件（服务框架，数据库，缓存，消息队列）
 - 封装对组件的调用
 - 封装常用的工具类
- 关键：有效的技术支持



质量保证 - 技术架构概览



质量保证 - 检查和监控

- 目的：提前发现系统隐患，降低生产故障率
- 重点：
 - CodeReview
 - 尽可能覆盖每个commit
 - 集成打包时通过SonarQube进行静态分析
 - 压力测试
 - Api接口100%覆盖
 - 报警监控
 - 基础指标：CPU占用率，内存占用率，服务连接数
 - 性能指标：响应时间平均值、P95值、P99值
 - 业务指标：交易下单率、成功率，报表实时率

进度把控

- 制定计划
- 推进措施
- 平衡资源与成本

进度把控 - 制定计划

- 目的：达成共识，明确责任
- 重点：
 - 明确里程碑和时间点
 - 安排任务到天，单个任务不能超过3天

序号	系统模块	功能	任务	任务类型	产品负责人	开发负责人	测试负责人	优先级	开始日期	完成日期	验收日期	上线日期
1	支付平台	平台对接	支付宝对接	调研	张三	李四	王五	1	2020/05/01	2020/05/03	-	-
2	支付平台	平台对接	支付宝对接	文档	张三	李四	王五	1	2020/05/04	2020/05/06	-	-
3	支付平台	平台对接	支付宝对接	开发	张三	李四	王五	1	2020/05/07	2020/05/10	-	-
4	支付平台	平台对接	支付宝对接	联调	张三	李四	王五	1	2020/05/11	2020/05/12	2020/05/13	2020/05/25

进度把控 - 推进措施

- 目的：降低沟通成本，及时发现和解决问题
- 行动：
 - 启动会，复盘会
 - 每日早会，每周例会
 - 周报，月报

进度把控 - 平衡资源与成本

- 目的：当资源不足时最大化降低成本
- 解决思路：
 - 明确项目类型和核心诉求：复利型项目可以适当牺牲成本
 - 把握关键资源：核心系统、模块自研，其它系统、模块外包
 - 协调资源：分期实现，寻求兄弟团队和领导帮助

问题应对

- 系统问题
 - 性能，稳定性，安全性
- 延期风险
- 线上故障

问题应对 - 性能

- 基本原则
 - 动静分离
 - 读写分离
- 缓存
 - 主动充缓存；两级缓存：分布式、本地
- 业务操作
 - 能异步就不同步
- 其它
 - 按需获取数据
 - 批量操作
 - 减少I/O

问题应对 - 稳定性

- 冗余
 - 不允许存在单点
 - 可以快速扩容节点
- 限流熔断
 - 合理设置一定时间范围内的应用、服务连接数限制
 - 快速失败：对于超过一定时间阈值的请求直接丢弃
- 重试
 - 合理设置超时时间、重试次数、重试节点数

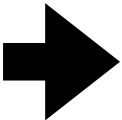
问题应对 - 安全性

- 数据安全
 - 敏感数据不可逆加密存储，关键数据混淆（遮蔽/变形/收缩）存储
 - 后台页面操作行为，生产环境登录、操作行为记录并审计
 - Api接口请求响应数据加密传输，图片、视频数据添加盲水印
- 账户安全
 - 登录管理：异地登录提醒，更换手机号、城市后强制重新登录
 - 资产变动：支付密码验证
 - 异常监控：频繁登录（IP/手机号）报警，频繁交易报警（IP/手机号/商品）
- 网络安全
 - 生产环境管控：入口服务仅开放必要端口（80/443），所有服务禁止访问外网
 - 办公环境管控：禁止直连生产环境，禁止通过堡垒机上传、下载文件
 - DDoS/CC攻击：采购高防IP服务

问题应对 - 延期风险

- BUG日清
 - 增加**每日BUG例会**，逐个确认BUG原因、负责人、需要的协助事项及解决时间
- 需求变化频繁
 - 设置**需求截至时间点**，该时间点以后的需求作为优化需求，如：1.0.1，1.0.2
- 某团队不给力
 - 原团队继续的同时将此团队负责的功能拆分给其它团队或寻求外包协助

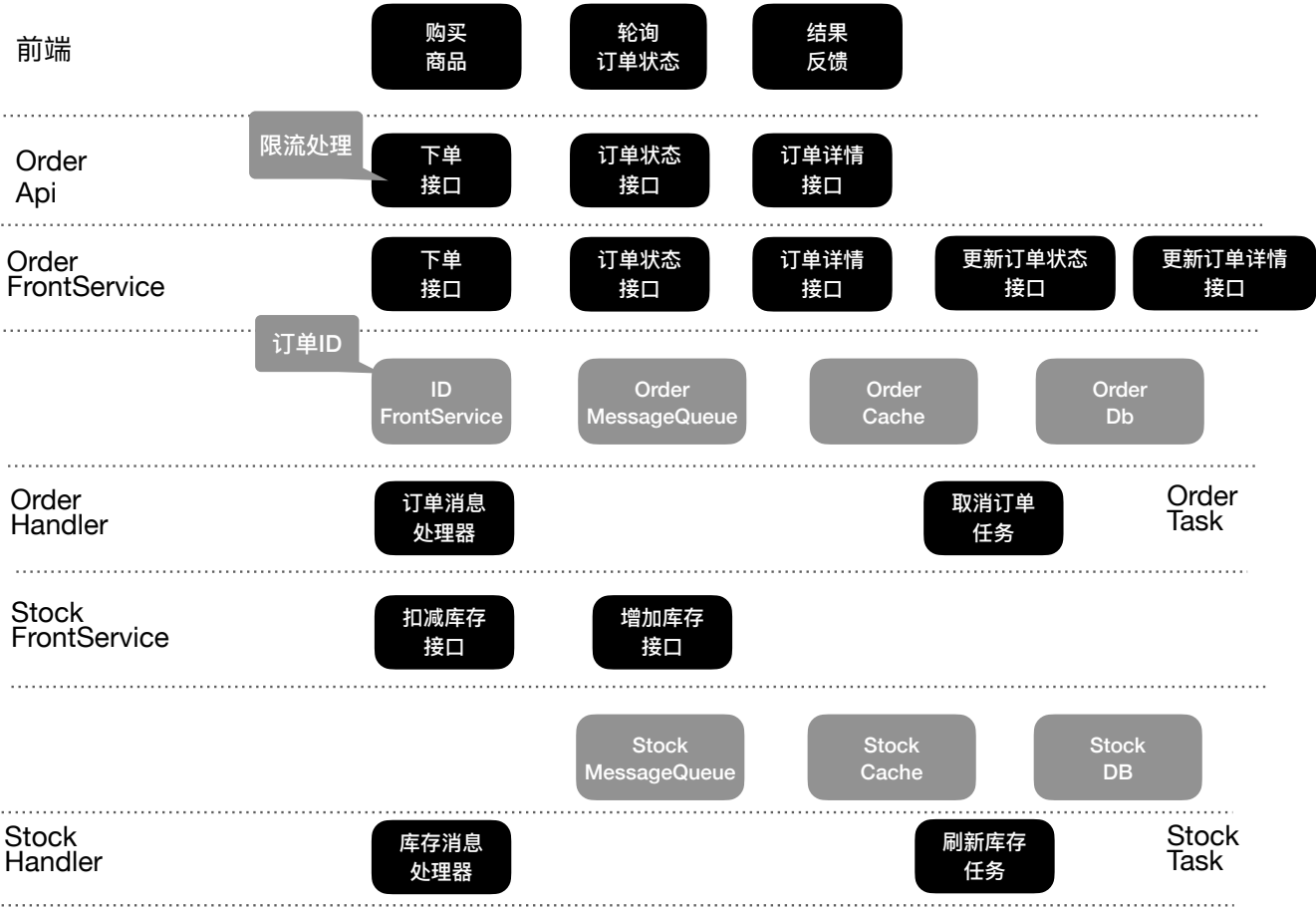
问题应对 - 线上故障

- 目的：减少业务损失
 - 现象：
 - 流量突增
 - 应用连接数突增
 - 应用CPU占用率突增
 - 应用内存占用率突增
 -
- 
- 高频IP封禁
 - 高频API接口限流
 - 扩容（节点，CPU，内存）
 - 排查应用依赖的数据库、缓存组件
 - 排查应用日志
 - 排查近期上线的模块
 -

谢谢

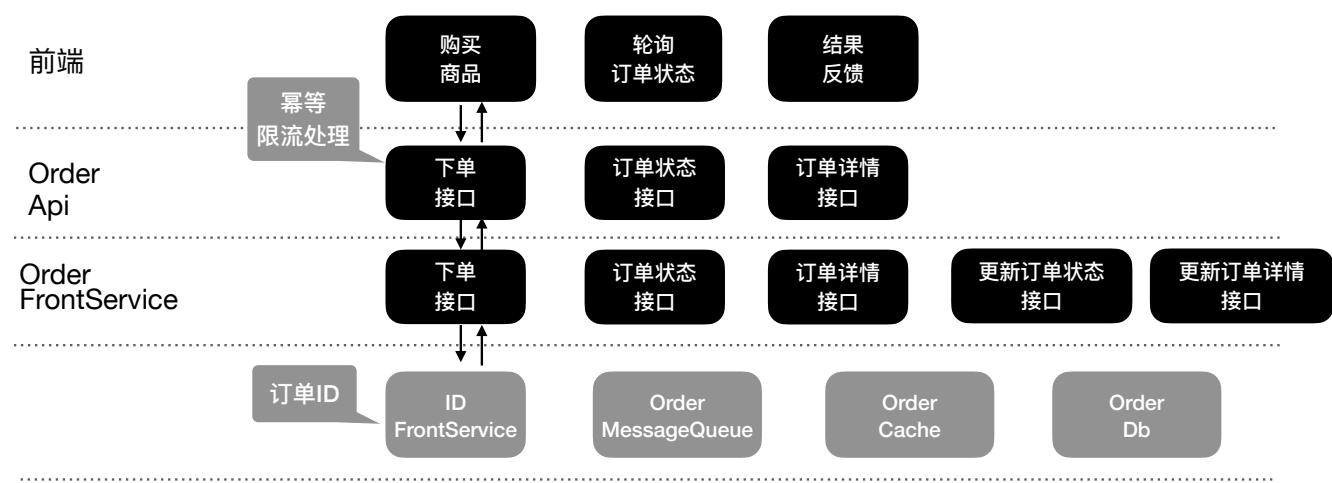
附 - “秒杀”类问题解决方案

- 影响：高并发导致系统不稳定
- 解决方案：
 - 限流（请求次数/时间）
 - 缓存（主动充）
 - 削峰（异步处理）



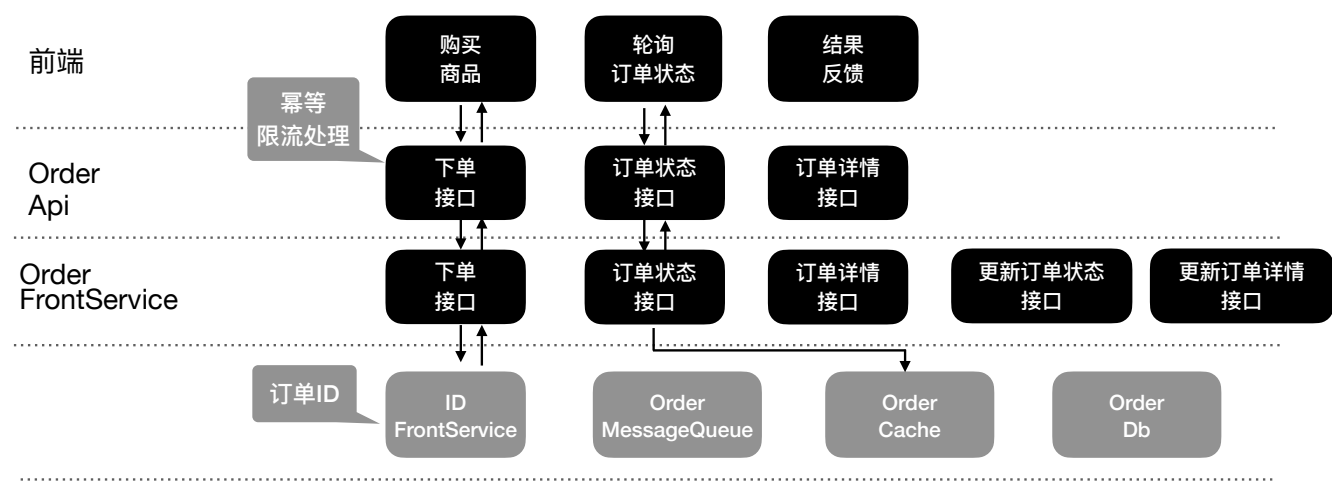
附 - “秒杀”类问题解决方案

第一步：下单时返回订单ID



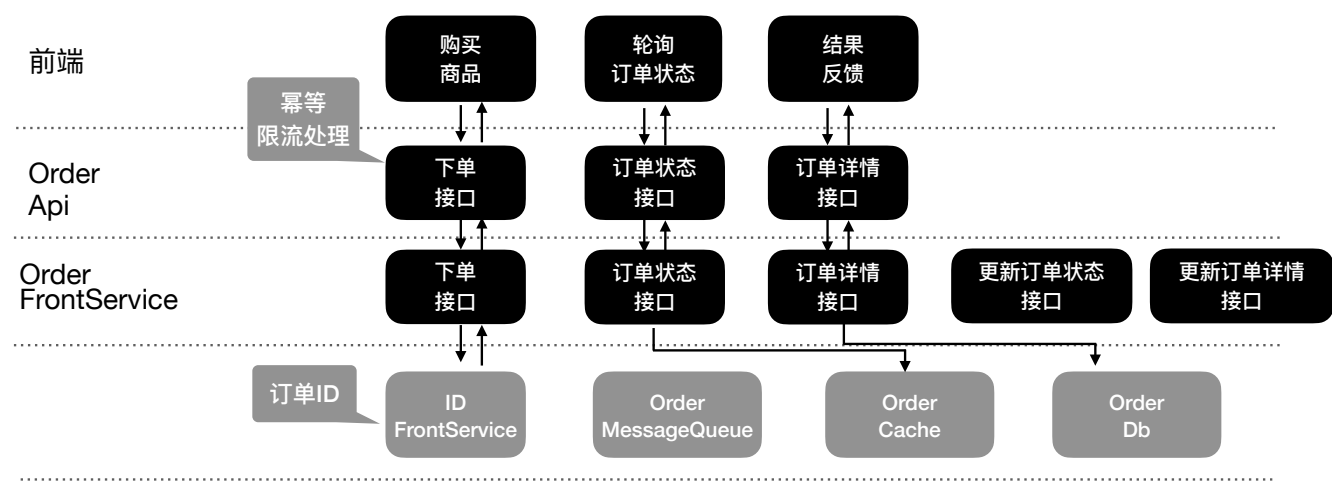
附 - “秒杀”类问题解决方案

第二步：定时轮询订单状态



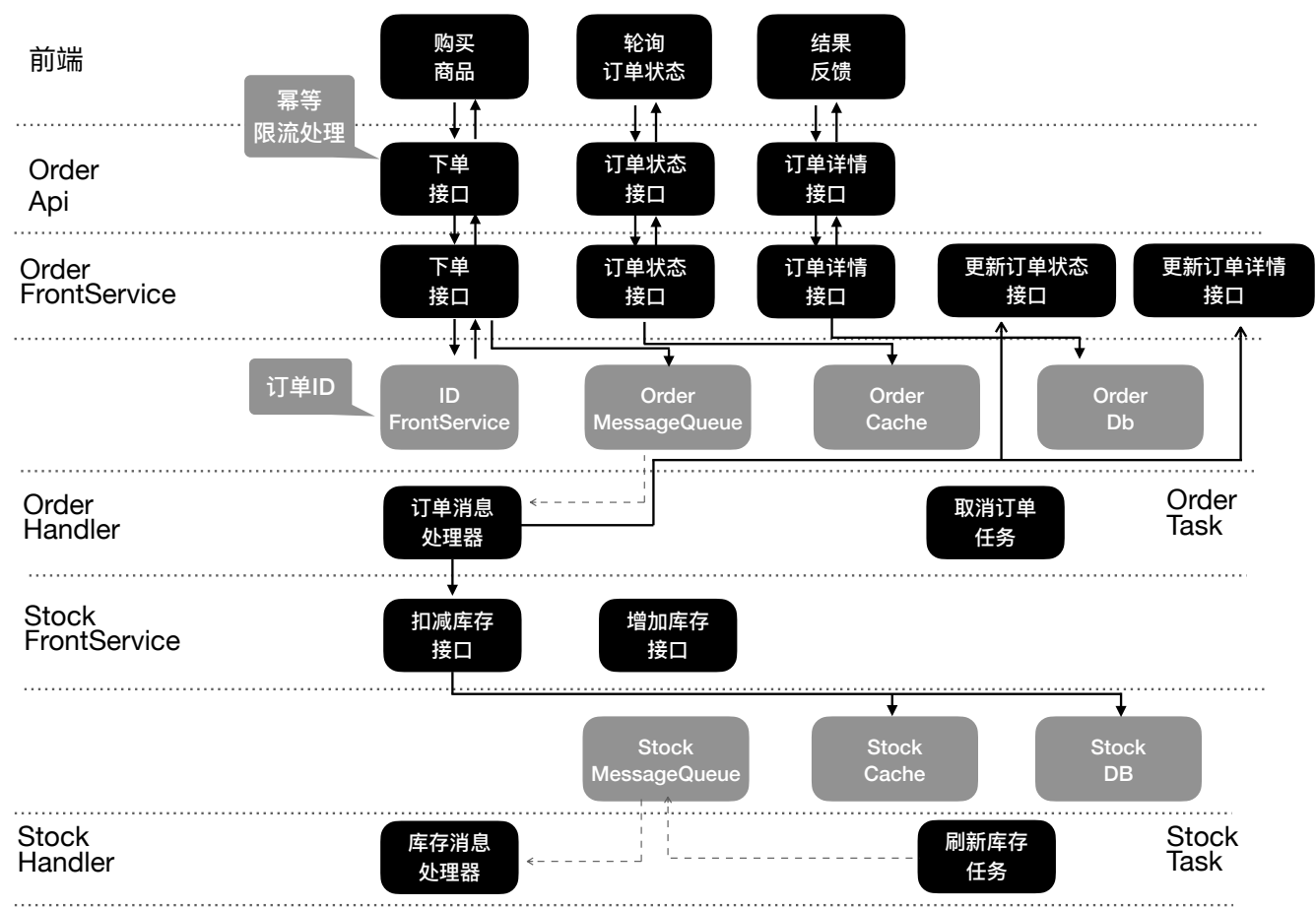
附 - “秒杀”类问题解决方案

第三步：获取订单详情



附 - “秒杀”类问题解决方案

第四步：异步处理下单请求



附 - “秒杀”类问题解决方案

第五步：取消未支付订单

