

红包狂欢背后的秘密

手Q春节红包后台架构实践

周海发
腾讯

SPEAKER INTRODUCE

周海发 腾讯

- 2011年加入腾讯，一直从事 IM 基础系统开发和运营，先后参与过 PTLogin 统一登录和消息漫游存储改造项目，连续三年参与并负责手Q春节红包后台系统架构设计，在海量分布式高性能系统设计方面积累了一些经验。



SPEAKER
ArchSummit 2017 ShenZhen



TABLE OF
CONTENTS 大纲

- **红包产品简介**
- 红包系统架构设计演进
- 红包核心功能设计介绍
 - 派奖策略设计
 - 通用地图任务方案设计
 - 配置系统设计
 - 实时采集方案设计
- 红包架构设计策略总结

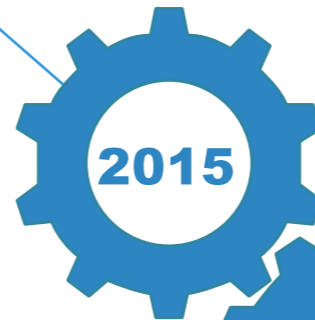
红包产品发展历程

李晨
20:00派发红包



企业&明星红包

参与用户数1.54亿
派发价值30亿红包



2015



2016



2017

刷一刷红包

参与用户数3.08亿
派发22.34亿个红包



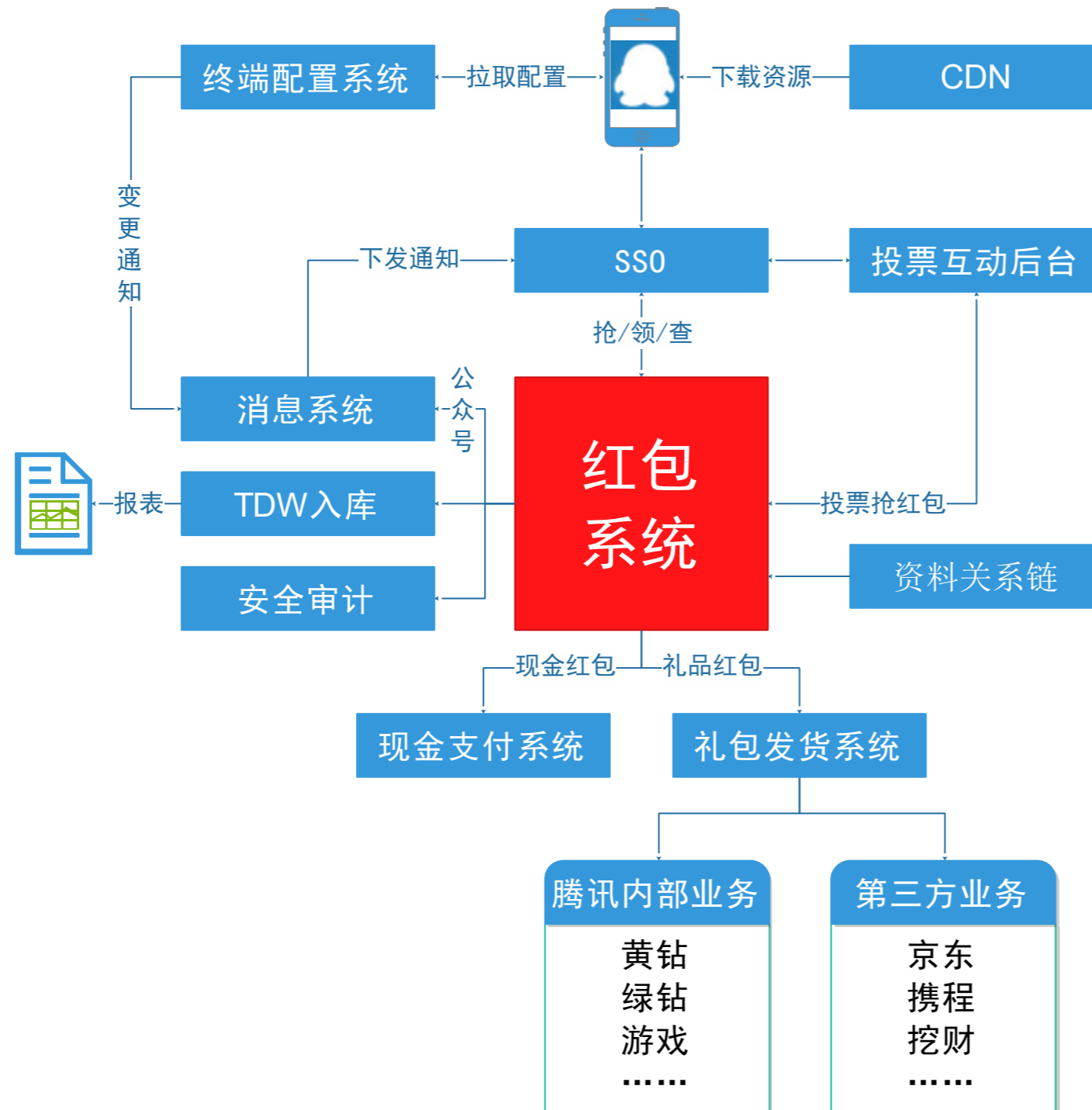
LBS+ AR& 刷一刷红包

参与用户数3.42亿
派发2.5亿现金红包和价值30亿卡券红包

TABLE OF
CONTENTS 大纲

- 红包产品简介
- **红包系统架构设计演进**
- 红包核心功能设计介绍
 - 派奖策略设计
 - 通用地图任务方案设计
 - 配置系统设计
 - 实时采集方案设计
- 红包架构设计策略总结

红包后台架构及设计原则



- 预处理充分应用
 - ✓ 静态资源闲时下载
 - ✓ 现金发放预处理，提前分拆订单
 - ✓ 静态地图数据预处理构建
- 前端保护后端
 - ✓ 尽量缩减往后端透传的流量，每一层优先做预判和检测逻辑
 - ✓ 接入层保护红包系统，限定客户端请求
 - ✓ 红包系统保护CFT、礼包系统，实时控制发放速度
- 架构简单可靠、异步化
 - ✓ 主逻辑灵活伸缩
 - ✓ 重试模块异步保证可靠请求

架构演进|2015-企业&明星红包

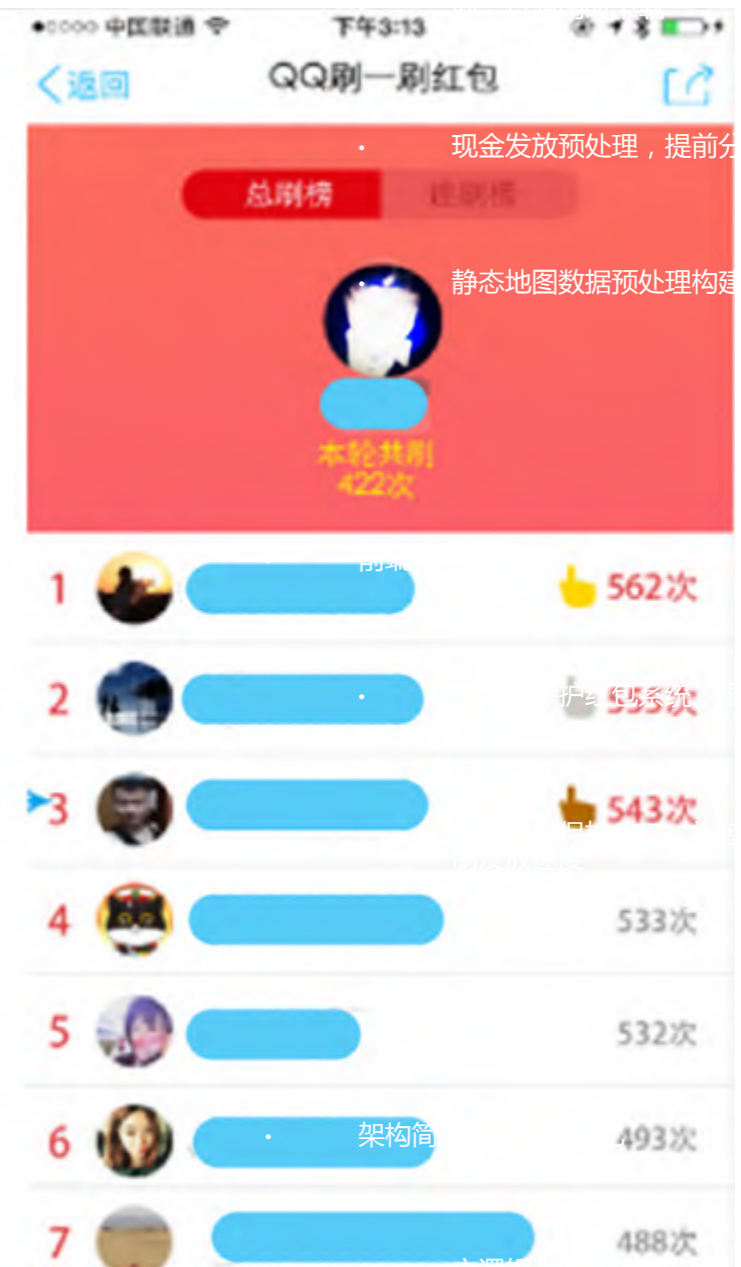


架构演进|2015-企业&明星红包

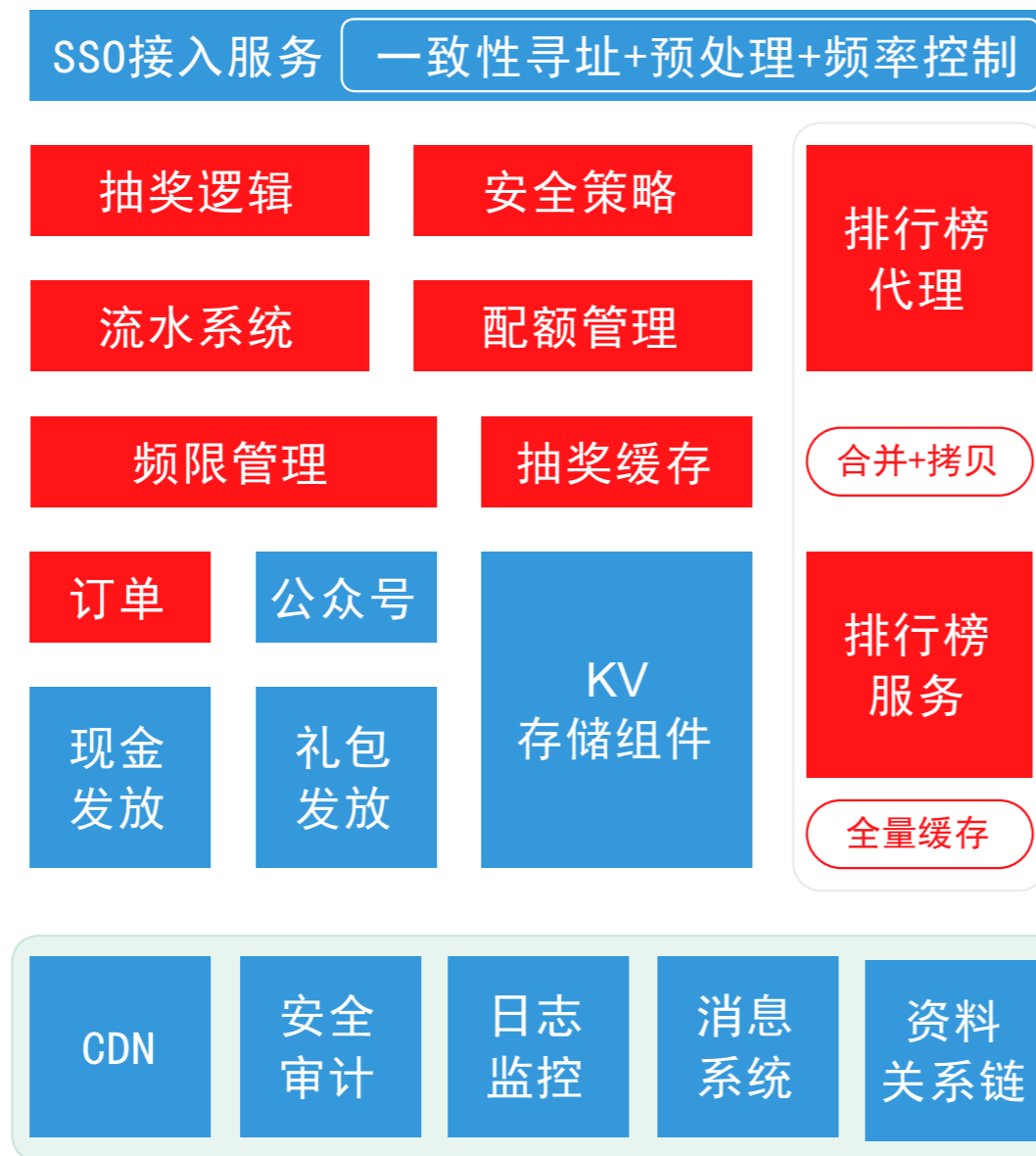


- 无经验,时间紧
 - ✓ 立项到上线不足2个月
 - ✓ 20种虚拟礼包接入
- 多部门协作
 - ✓ 跨4个BG,涉及9个部门
- 突发异常应对
 - ✓ 后端业务容量参差不齐
 - ✓ 部分业务缺乏海量运营经验
- 奖品有效发放
 - ✓ 如何保证不多发少发
 - ✓ 如何保证高中奖率
 - ✓ 如何保证整场活动持续派发
- 安全防刷
 - ✓ Uin频率限制
 - ✓ 每种奖品个数限制
 - ✓ 安全打击和流水审计

架构演进|2016-刷一刷红包

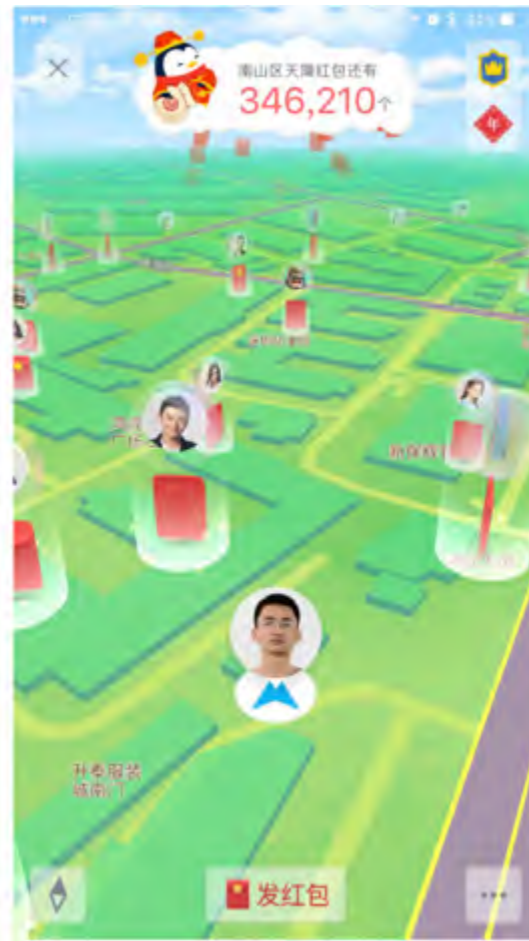


架构演进|2016-刷一刷红包

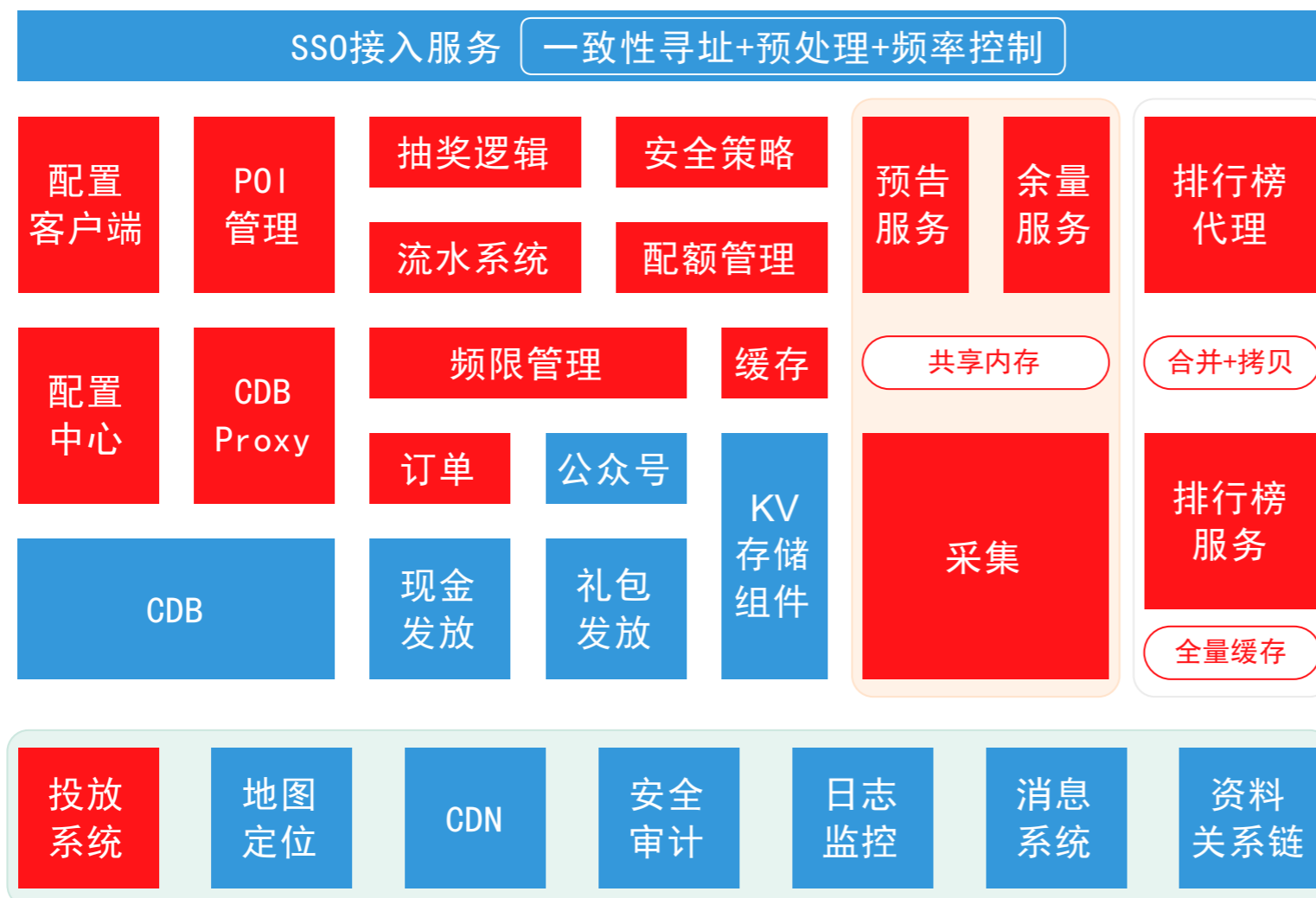


- 海量高并发
 - ✓ 预估抽奖峰值800w/s
- 快速扩容支持
 - ✓ 约20G订单数据
- 突发异常应对
 - ✓ 后端业务性能及容量参差不齐
 - ✓ 部分业务缺乏海量运营经验
- 好友间刷次数实时排行
 - ✓ 读扩散or写扩散
 - ✓ 存储如何设计
 - ✓ 堆排

架构演进|2017-LBS+AR天降红包



架构演进|2017-LBS+AR天降红包

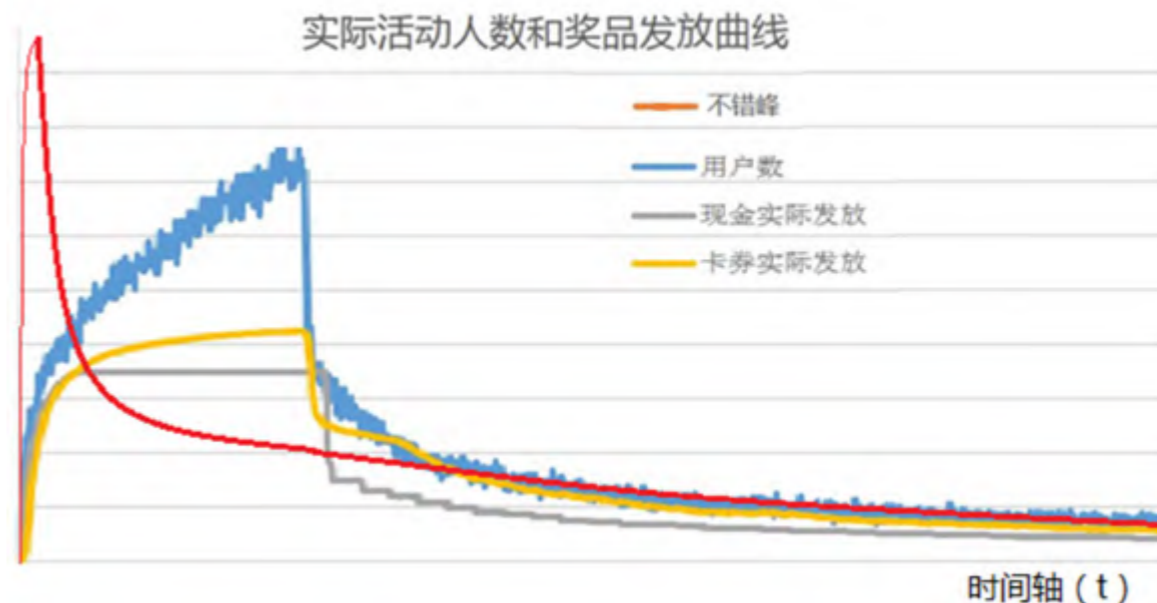
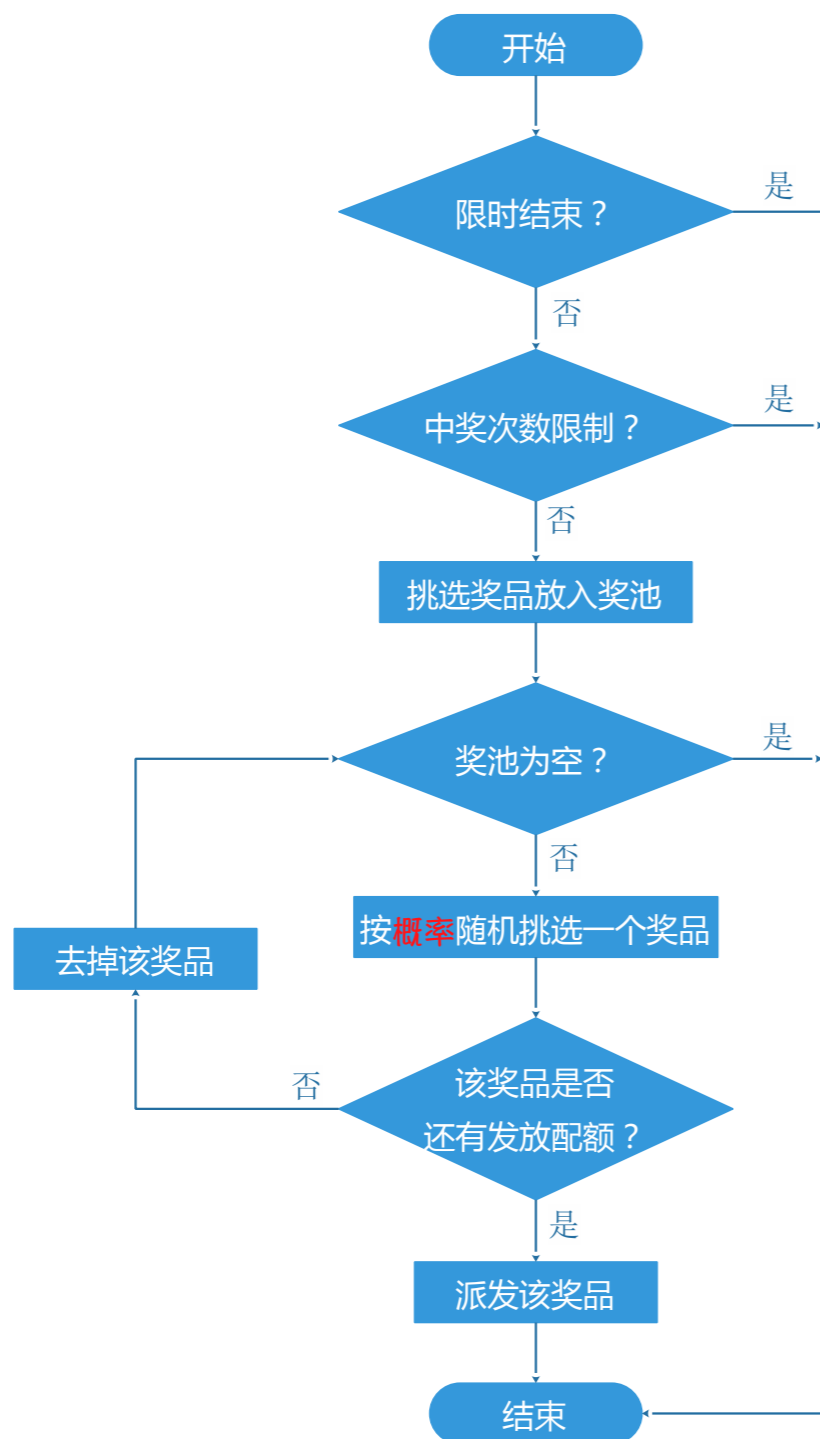


- 地图任务数据管理
 - ✓ 千万级POI数据
 - ✓ Set化管理, 投放灵活
 - ✓ 预构建缓存, 双Buffer设计, 免网络流量高速读取
 - ✓ 地区红包拆分规则
- 实时余量统计
 - ✓ 地区×任务×奖品多维度统计
 - ✓ 无状态, 余量展示问题
- 历史记录割接
 - ✓ 数据库分表, 批量割接, 快速入库, 主服务无阻塞

TABLE OF
CONTENTS 大纲

- 红包产品简介
- 红包系统架构设计演进
- **红包核心功能设计介绍**
 - **派奖策略设计**
 - 通用地图任务方案设计
 - 配置系统设计
 - 实时采集方案设计
- 红包架构设计策略总结

红包系统业务模型



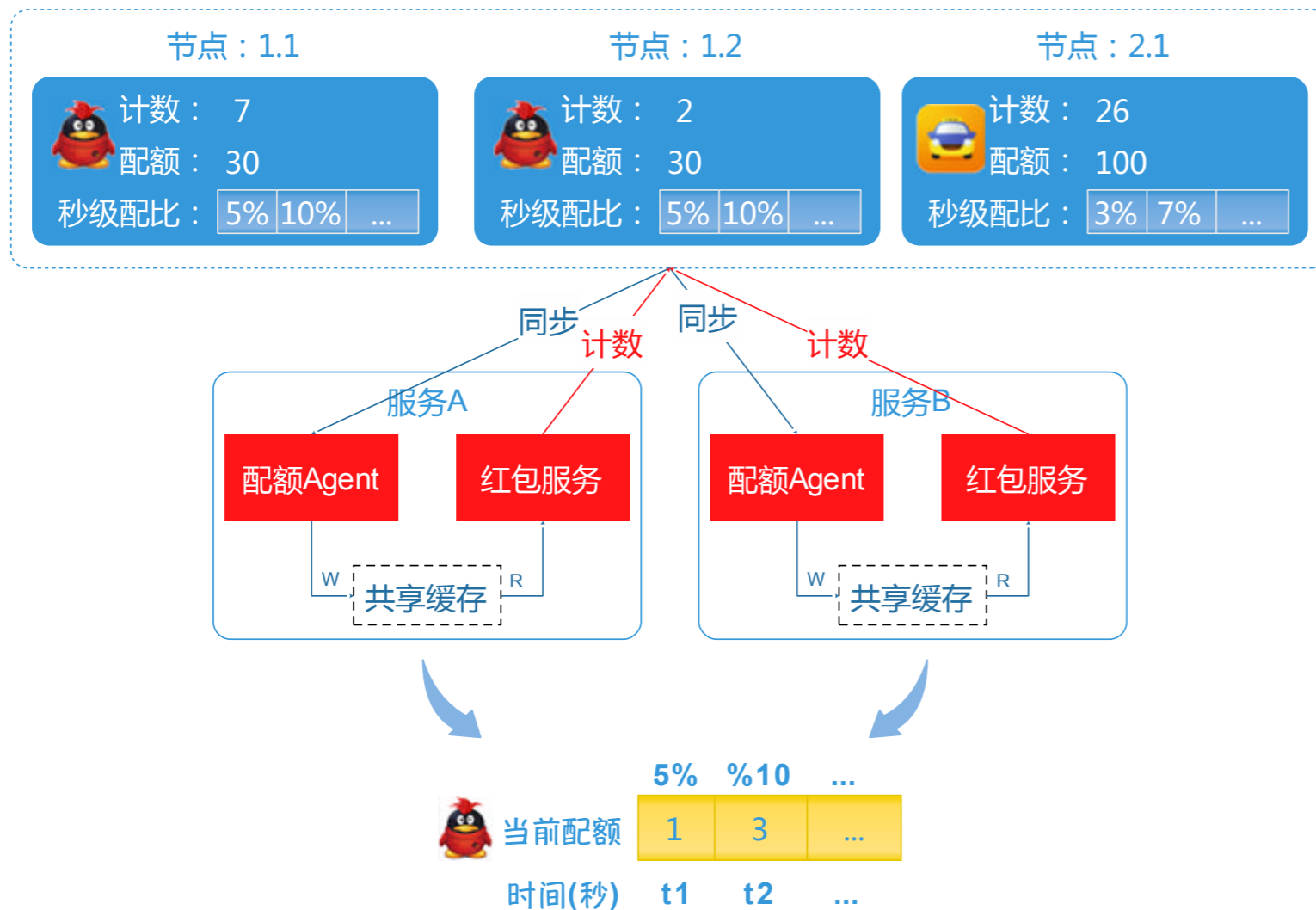
■ 派发策略

- ✓ 瞬间峰值大，后端顶不住
- ✓ 错峰有可能导致后来用户抢不到
- ✓ 快速响应业务故障的能力
- ✓ 结合预演数据提取业务模型，评估请求量，准备资源
- ✓ 评估合理的错峰时间，保持高中奖率

■ 抽奖算法

- ✓ 奖品发放速率控制
- ✓ 奖品发放比例实时可调
- ✓ 尽量保证奖品都派发出去

灵活可靠的奖品派发策略



- Grocery分布式计数器
 - ✓ 灵活拆分和配置，支持不同数目和限制的物品，**快速响应产品物品变更需求**
- 灵活的配额策略
 - ✓ 奖品发送速度精确到秒
 - ✓ 10s快速全量更新奖品全局配置，保护后端服务
- 毫秒级限速控制
 - ✓ 本地缓存配置，根据机器数预先分配上限
 - ✓ 物品id+时间戳为key在本地拆分配额，确保往后端请求均匀

TABLE OF
CONTENTS 大纲

- 红包产品简介
- 红包系统架构设计演进
- **红包核心功能设计介绍**
 - 派奖策略设计
 - **通用地图任务方案设计**
 - 通用地图任务方案设计
 - 实时采集方案设计
- 红包架构设计策略总结

通用地图任务方案设计



- 简化为平面的静态缓存设计
 - ✓ 默认地图标尺
 - ✓ 用户坐标到格子的简单计算
 - ✓ 格子-POI点-任务的存储
 - ✓ 主逻辑全量缓存，加速查找

- 基于任务的通用投放策略
 - ✓ POI随人口热力图/区域切分
 - ✓ 抽象任务和POI的对应关系
 - ✓ 不同任务的投放策略，奖品数目自由配置

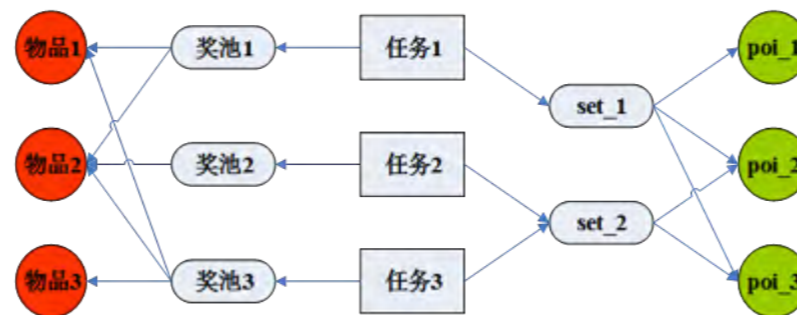
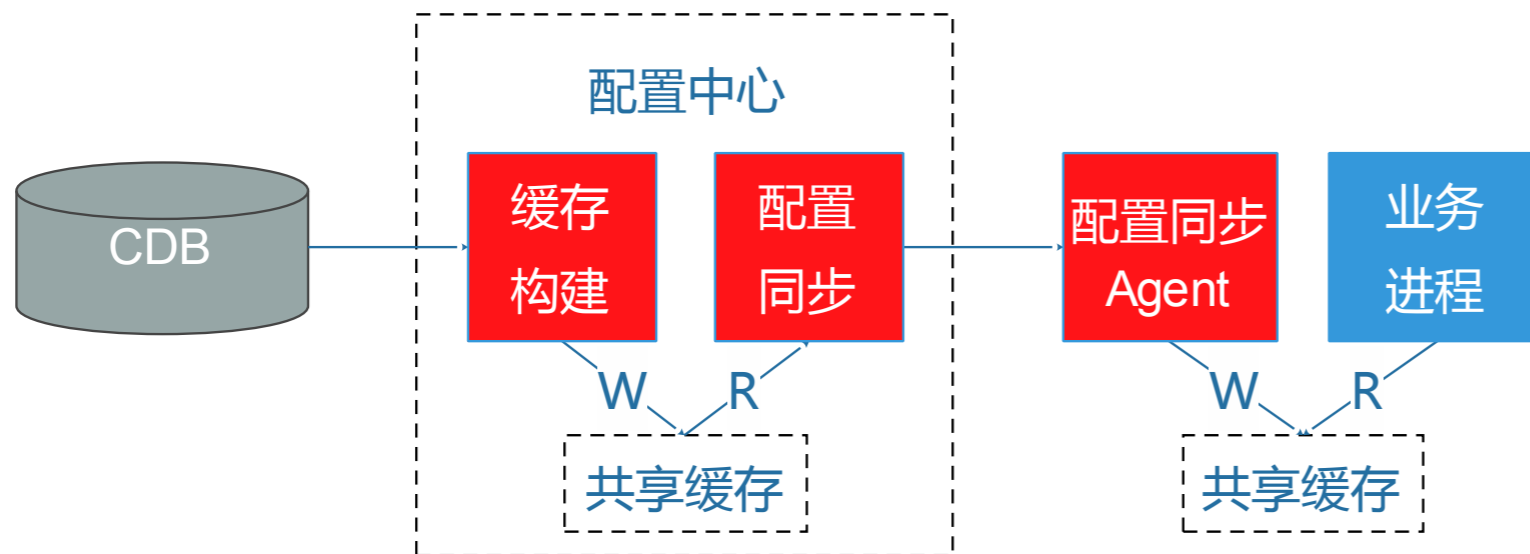


TABLE OF
CONTENTS 大纲

- 红包产品简介
- 红包系统架构设计演进
- **红包核心功能设计介绍**
 - 派奖策略设计
 - 通用地图任务方案设计
 - **配置系统设计**
 - 实时采集方案设计
- 红包架构设计策略总结

配置系统设计



- 配置缓存预先构建
 - ✓ 配置的阅读量远高于写量
 - ✓ 地图及任务投放数据全量有序缓存，加速访问
 - ✓ 配置数据存在关联，采用双Buffer设计，配置可无锁更新和使用
 - ✓ 统一配置操作接口

DoubleBuffer

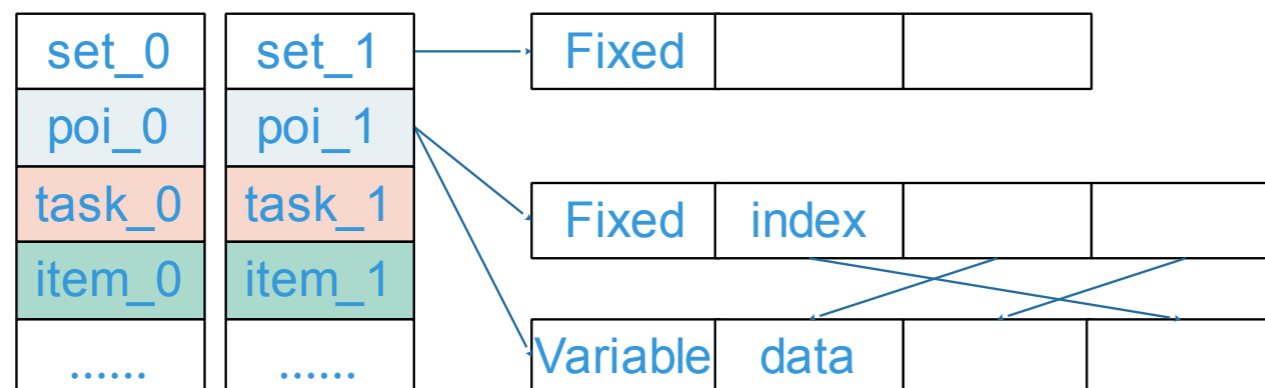
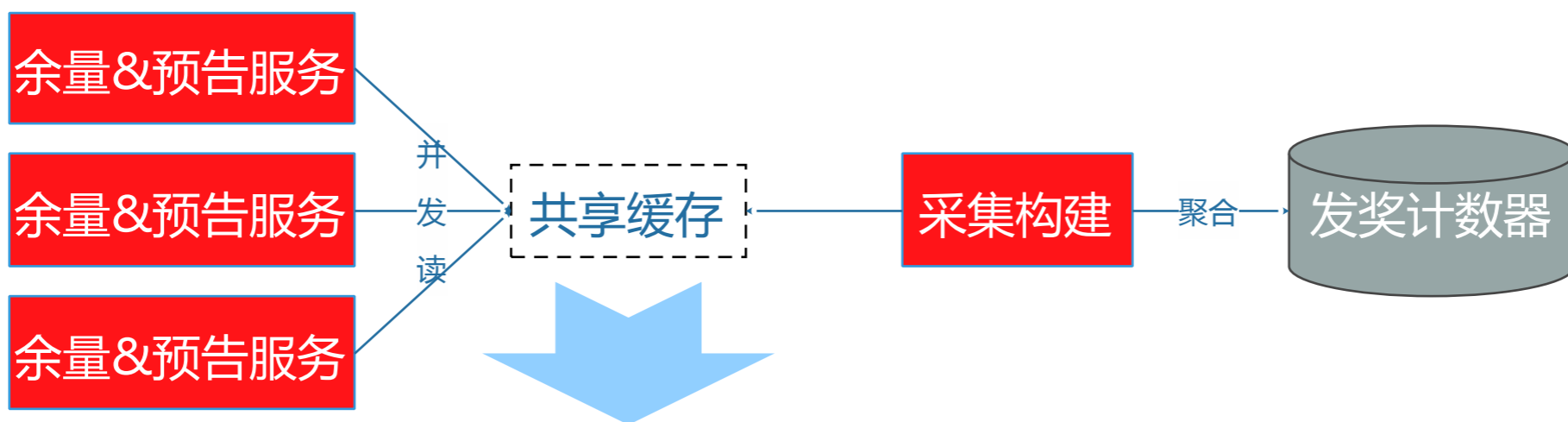


TABLE OF
CONTENTS 大纲

- 红包产品简介
- 红包系统架构设计演进
- **红包核心功能设计介绍**
 - 派奖策略设计
 - 通用地图任务方案设计
 - 配置系统设计
 - **实时采集方案设计**
- 红包架构设计策略总结

实时采集方案设计



Adcode	Task	Item
0	0	100
0	0	101
0	1024	100
0	1024	101
440305	1000	100
440305	1000	101
440305	1001	100
440306	1000	100
440307	1001	101

- 采用读写分离，共享缓存解耦
- 读服务可平行扩展，不受存储层限制
- 多维度统计，聚合采集，减少包扩散
- 预构建本地缓存，提升检索性能
- 海量查询不会影响抽奖系统

TABLE OF
CONTENTS 大纲

- 红包产品简介
- 红包系统架构设计演进
- 红包核心功能设计介绍
 - 派奖策略设计
 - 通用地图任务方案设计
 - 配置系统设计
 - 实时采集方案设计
- **红包架构设计策略总结**

过载保护及柔性策略

重点：要提前做好预案！！

部门	模块	柔性策略	当前状态	影响
即通平台部	SSO	后台调整客户端抽奖/随机上报间隔	支持	降低刷红包请求
	ptlogin	调整中奖比例，减少领用请求	支持	可能部分奖品派发不出去
	Grocery	SSO调整中空红包比例	支持	可能部分奖品派发不出去
应用综合部	公众号	关闭或调整中奖比例	支持	用户收不到中奖通知
增值渠道部	异步发货	调整中奖比例	支持	延迟发货
	MP			
财付通	财付通	调整中奖比例	支持	可能部分现金红包派发不出去
支付平台部	H5立即领用（装扮类）	调整中奖比例	支持	可能部分个性礼包派发不出去
第三方	H5立即领用	调整中奖比例或禁用第三方跳转	支持	可能部分卡券礼包派发不出去

有效发放及安全策略

事前检查

实时保障

事后校验

奖品发放

- ✓ IP访问权限限制
- ✓ 机器环境操作日志
- ✓ 红包生效后校验
- ✓ 现金预处理以防多发

- ✓ 订单去重机制防止重复请求
- ✓ 订单服务多机部署

- ✓ 系统内日志流水对账
- ✓ 系统外日志流水对账（财付通和增值）

业务安全

- ✓ 白名单用户参与
- ✓ 每次活动之前按照之前安全逻辑动态刷新

- ✓ 定位器识别
- ✓ 信令侧SSO通道保证
- ✓ 用户号码，设备imei号频率和领取上限限制
- ✓ 号码户口/IP所在地/LBS所在地几者的跳变的检测

- ✓ 中奖记录审计和封号处理

大型活动通用架构设计原则

架构

安全

活动存储

- ✓ 写数据保证最终一致性
- ✓ 完善的日志，对账功能

- ✓ 完善的日志，对账功能

活动逻辑

- ✓ 静态数据提前构建高效缓存
- ✓ 读写分离，读不要求强一致性
- ✓ 热点数据预处理
- ✓ 业务流程异步化
- ✓ 通用任务配置机制
- ✓ 配置系统快速处理异常问题

- ✓ 用户频率，数目限制
- ✓ 恶意用户打击策略

接入系统

- ✓ 活动结果预处理
- ✓ 容量和柔性策略考虑充分，保护后端

- ✓ 用户恶意攻击应对

前端

- ✓ 活动资源预加载
- ✓ 前端请求间隔/次数可配置
- ✓ 有合理的错峰逻辑和产品展现形式

- ✓ 用户检测和上报

如何确保大型活动成功

- ✓ 后台代码各部分交叉 Review
- ✓ 协议响应时间测试
- ✓ 模拟真实请求全链路压测
- ✓ 核心视图和关键属性整理
- ✓ 定制秒级监控告警
- ✓ 告警标准和负责人梳理

质量

工具

监控

- ✓ 内存dump/秒起检测完备
- ✓ 关键操作脚本/页面化
- ✓ 活动前需要做的check工具脚本化

提升红包系统内部的质量和可运营性，确保万无一失

- ✓ 各种突发事件的**正交完备性预案讨论**，明确各个角色职责
- ✓ **预案手册打印**出来，人手一份
- ✓ 运维和值班同学架构/预案培训

周边系统梳理

应急预案

演练和复盘

- ✓ 安排红包相关的各个**参与部门联合架构Review**
- ✓ QQ后台系统相关准备
- ✓ 活动期间信息的传递流程
- ✓ **构建真实用户请求模型**
- ✓ 应急预案和柔性策略演练
- ✓ 每一轮演练明确需要演练的目标，结束后解决所有可疑点

梳理全系统的架构，过载保护方案，明确各类应急预案并反复演练，确保万无一失

写在最后

- 红包架构设计演进以及活动类系统设计原则
- 具体场景具体分析，从体验的角度理清关键逻辑
- 必须以等量的请求/存储模拟压测
- 不要相信运气，要相信概率

THANKS!