

部门业务情况

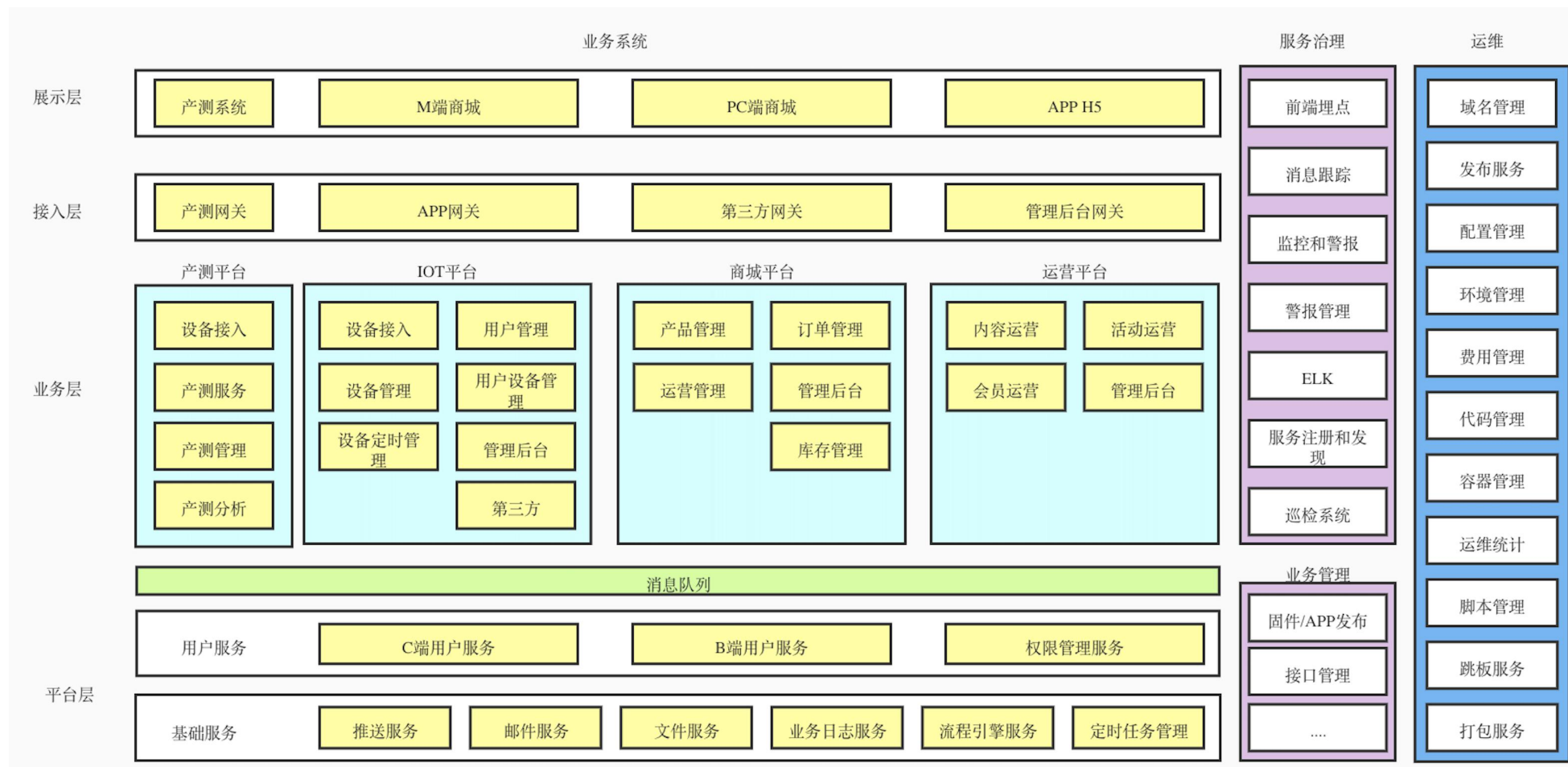
- ▶ IoT平台：单品开发，产测开发、开放平台
- ▶ 商城：商城、商城管理系统
- ▶ 运营：活动运营、运营管理系统、菜谱管理系统
- ▶ 前端：安防B端管理系统，M端商城、PC端商城、运营H5，商城管理系统，运营管理系统，菜谱管理系统
- ▶ 大数据：IoT数据分析，商城数据分析、活动数据分析、推广数据分析
- ▶ 安防：温控器开发、门锁开发
- ▶ 中国区：单品开发、第三方接入

部门业务研发思路

- 公共功能平台化
 - 通过复用，降低开发和维护成本
- 业务开发按业务线划分
 - 降低业务线之间的耦合，减少沟通成本、提高开发效率
 - 提高开发人员业务专业能力
 - 提高开发人员业务意识
 - 能根据业务特性，制定不同的研发策略

业务	研发策略
IoT	向开放平台演进，稳定性达到5个9，扩展性达到非全新单品接入免开发，性能达到千万级的设备和用户
商城	金融级别可靠性，快速开发和迭代（2周完成一次功能迭代上线）
运营	金融级别可靠性，快速迭代开发（每个运营活动独立上线，公共功能组件化）
前端	开发效率（组件化开发）、提高用户体验（资源加载加速、快速响应
大数据	数据分析快速响应、精准化营销（用户分层、分群）、推动业务发展
运维	稳定性、运维效率和代码流转效率

部门业务情况-当前架构



团队建设

► 团队特点:

- 新团队较多 (运营、商城和前端)
- 新人多 (老员工22人, 社招24人, 校招15人)

► 团队建设思路

- 明确分工: 业务方面, 按业务线、按模块、按技术等; 管理方面, 按专利、流程规范、团建活动等
- 流程规范: 新流程规范制定; 已有流程规范整理、宣讲和落实; 核心流程规范增加考核环节
- 基层管理培养: 周会组织规范、项目管理方法、优先级意识、主人翁意识和业务导向意识等
- 项目管理: 前端排期规范、后台排期规范、敏捷迭代意识

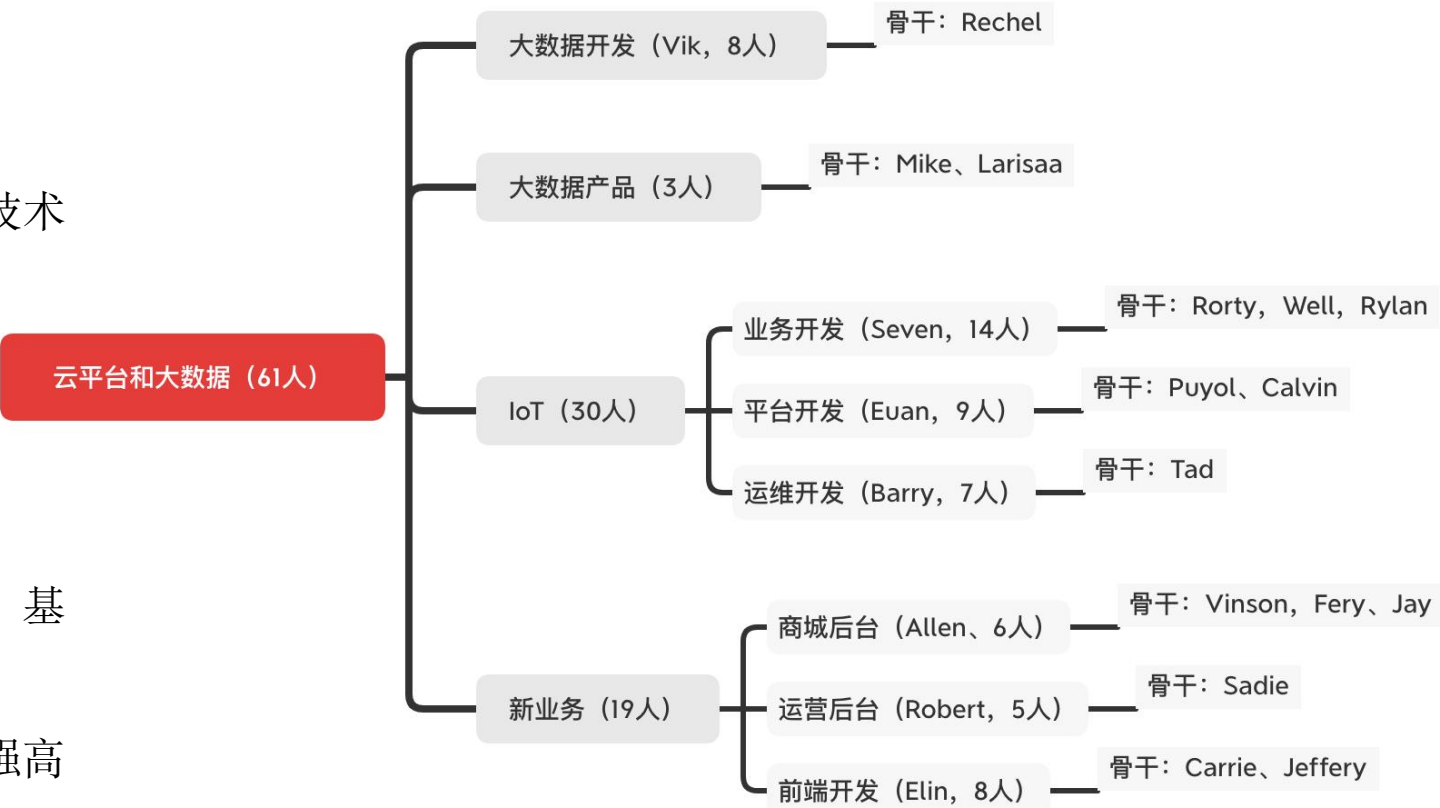
团队建设

► 组织架构和团队情况

- 初步形成业务负责人、小组负责人、技术骨干和普通员工四层梯队
- 高级人才偏少，特别是大数据开发

► 团队规划

- 加强人才培养，如加强技术学习计划、基层管理培养计划，推行骨干培养计)
- 加强人才招聘，推行结构化面试，加强高级人才招聘
- 搭建健康解决方案后台开发团队



业务情况-商城进展

策略和措施:

► 可靠性

- 监控和警报: 严格的策略, 不放过任何一个错误;
- 业务报表: 防止业务错配 (优惠券报表、订单盈利校验等)
- 基础数据审核: 优惠券新增发放、商品价格设定通过钉钉审核审批, 防止设置错误造成损失
- 库存缓存和异步同步: 降低ERP不稳定的影响, 同时保证不超卖
- 物流异步处理: 通过消息队列, 保证最终一致性
- 支付结果轮询任务处理: 通过轮询任务对支付结果进行校验, 保证不掉单、不漏单

► 快速开发和迭代

- 敏捷开发, 流水线作业, 2周一个版本
- 简化设计: 分订单管理、商品管理、营销管理和库存管理4大模块

进展

- 2021年4月份: 商城基础功能, 包括商品展示、购物车、订单、支付和物流
- 2021年5月份: 引入商城营销优惠券功能
- 2021年6月份: 接入信用卡支付
- 2021年7月份: 营销推广活动完成免费试用业务
- 2021年8月份: 一口价商品体系上线
- 2021年9月份: 捆绑销售业务、领券中心业务

业务情况-商城规划

- 可靠性:
 - 引进RabbitMQ技术, 解决订单超时未支付释放库存问题
 - 引起分布式事务框架, 提高数据一致性
 - 新增内部接口网关服务模块, 通过统一签权实现接口安全
- 快速迭代和开发
 - 组件化营销模块中券种适用判定, 新券业务中将由单个人5日的开发工作量减少至1;
 - 模板化订单优惠金额计算, 新券业务中将由单个人3日的开发工作量减少至1;
- 性能
 - 引进redis缓存技术, 缓存商品主要信息, 降低请求时延, 提高客户端体验;
 - 引入DolphinScheduler分布式定时任务技术, 消除服务定时任务之间锁等待;
- 扩展
 - 引入Elasticsearch搜索引擎, 为商城商品搜索提供模糊查询、按需排序;
 - 引入Kafka消息中间件, 将商城订单状态变更推送, 为大数据、运营提供业务支持;

业务情况-运营

策略和措施:

- 可靠性
 - 监控和警报: 严格的策略, 不放过任何一个错误
 - 事件队列: 通过消息队列来传递事件, 解耦模块
 - 分布式事务: 在业务层面实现接口幂等和撤销功能
 - 功能预览: 推送预览 & 菜谱预览
- 快速开发和迭代
 - 活动之间相互独立开发、测试和上线
 - 组件化设计, 如抽奖板块、任务板块
- 安全
 - 推送审核: 根据不同角色和推送范围, 自动选择审核流程

进展

- 2021年3月份: 新运营管理后台上线、美区App增加菜谱版块上线
- 2021年4月份: 菜谱评论、菜谱运营后台一期 (专题菜谱) 功能上线
- 2021年5月份: UGC菜谱功能上线、积分签到上线
- 2021年6月份: 美区菜谱版块2.0上线、Referral二期上线
- 2021年7月份: 菜谱扩区功能上线、推送审核功能上线、组团二期上线、KOL系统上线
- 2021年8月份: UGC大赛功能上线、积分抽奖和兑换优惠券上线

业务情况-运营规划

- 可靠性:
 - 引进RabbitMQ, 确保内部重要消息百分百到达
 - 引起分布式事务框架, 减少事务撤销的开发工作
 - 新增数据检查和半自动化修补功能
 - 搭建风控系统, 检查异常用户, 实现黑名单功能
- 快速迭代和开发
 - 积分任务迁移到通用任务板块, 简化新积分任务开发工作
 - 增加operationActivity模块, 专门负责后续活动的业务, 解耦不同业务
- 性能:
 - 在任务板块引进本地缓存, 缓存任务信息及已完成任务的用户情况, 降低接口处理时延
 - 引进HyperLoglog, 减少banner的UV统计存储空间

业务情况-前端

策略和措施:

► 开发和迭代效率

- 组件化: 公共代码组件化, 自有组件库管理
- 模版: 运营h5页面模版、邮件模版
- 复用: PC端和M端商城重构, 复用业务逻辑代码, 提高维护效率
- 模块化: 运营活动模块化, 独立开发、迭代和上线, 独立开发、迭代和上线

► 用户体验

- 资源懒加载、预加载技术的运用, 降低首屏加载时间
- 合理运用loading交互, 提高人机交互体验
- 静态资源加速: 静态资源通过CDN分发, 减少资源下载时间

► 提高曝光率

- SEO (搜索引擎优化): 使用ssr (服务端渲染) 技术, 合理配置关键字

进展

- 2021年4月份: 商城后台管理系统上线, 包括商品管理、订单管理、用户管理、优惠券管理等基础功能; 新运营管理后台、专题菜谱上线。
- 2021年5月份: 积分签到、Referral一期上线。
- 2021年6月份: PC端商城上线, 包括商品展示、加购物车、下单、支付、订单管理等基础功能; 组团一期、Referral二期上线。
- 2021年7月份: M端商城上线, 配合营销推广活动完成免费试用业务; 组团二期、积分抽奖、KOL系统上线。
- 2021年8月份: PC端&M端商城功能迭代, 包括信用卡支付, 一口价商品, 新客体系等; UGC大赛功能、积分三期上线。

业务情况-前端规划

开发效率:

- 持续整理公共代码成组件, 积累和丰富组件库
- 框架: 开发自有内部后台管理系统框架

用户体验:

- 加载速度: 拆分第三方组件, 充分利用CDN加速

性能:

- 新技术: 引进vue3+vite新框架, 提高页面渲染效率和打包效率

可靠性:

- 埋点和监控: 引进规范化的事件埋点, 实现对运行异常的监控

业务情况-IoT规划

策略和措施:

- 可靠性
 - 监控和警报: 严格的策略, 不放过任何一个错误;
 - 事件队列: 通过消息队列来传递事件, 确保事件的最终一致性
 - 分布式事务: 接口幂等和撤销功能
 - 功能预览: 推送预览 & 菜谱预览
- 快速开发和迭代
 - 活动之间相互独立开发、测试和上线
 - 组件化设计, 如抽奖板块、任务板块
- 安全
 - 推送审核: 根据不同角色和推送范围, 自动选择审核流程

进展

- 2021年4月份: 商城基础功能, 包括商品展示、购物车、订单、支付和物流
- 2021年5月份: xx
- Xx: xx
- Xx: xx

业务情况-IoT规划

- 产品开发29款
 - 空净类14款, 厨电类6款, 健康类5款
- 产品迭代48次
 - 家庭房间需求: VeSyncApp具有家庭房间概念和家庭分享, 成员邀请功能
 - 健康场景1.0: 整合体脂秤, 营养称, 血压计数据, 为用户打造一个全面的健康监测管理平台
 - 山火需求: 用户所在地区山火分布, 提醒功能
- 开放平台建设
 - 完成2款单品SK50和DS03开发上线
 - 接入层tls放在ELB, 释放接入层的加解密压力, 接入能力更强
 - 设备接入协议废弃pid概念, 三方统一使用configModel标识
- 性能优化
 - 用户和设备关系引入缓存, 查询平均时延由3ms降低到1.3ms
 - 安全
 - 数据合规: 所有操作符合数据合规要求

业务情况-平台建设

策略和措施:

- ▶ 可扩展性、可维护性
 - 邮件服务改造, 为商城、运营平台提供统一邮件发送服务
 - 推送服务改造, 为商城、运营、菜谱提供统一、高性能推送服务
 - 用户和第三方平台授权公共服务, 实现快速和第三方云相互授权
 - EDM系统
- ▶ 生产、维护、问题定位效率
 - 消息链路追踪解决方案, 实现调用异常、缓慢快速精准定位
- ▶ 性能
 - Dubbo同可用区调用, 减少服务之间跨可用区调用, 每次调用降低0.1ms的时延

业务情况-平台规划

► 可扩展性、可维护性

- 流程服务化，实现流程功能与业务系统解耦
- 天气数据服务化，实现天气数据功能与业务系统解耦
- 文件服务重构，实现文件管理方案优化和访问控制
- 短信服务化，实现短信功能与业务系统解耦
- PushService、MongoDB合并，实现统一逻辑和管理
- DolphinScheduler分布式定时任务调度平台，实现定时任务平台化管理，提升定时任务高可用性
- 权限管理服务，实现云平台业务系统权限统一管理，为商城后台管理、运营后台管理提供权限控制服务

► 生产、维护、问题定位效率

- API管理平台，实现接口可视化定义、接口测试、接口元数据版本管理等一体化管理，支持迭代Mock、权限、通知功能，实现自动、安全、主动感知能力
- 消息链路追踪系统，迭代报表功能，实现数据可视化，指导系统优化、方案决策

业务情况-运维

- 安全：建立用户数据安全和合规管理流程
 - 用户邮箱、用户地址、健康数据等敏感数据脱敏
 - 用户数据查询系统、日志查询等系统权限管理规范
- 稳定
 - 按照业务线划分，测试环境上，各业务线独立服务器部署，避免资源挤占，提升环境稳定性
 - 联调环境独立AWS账号部署，进行VPC网络隔离，避免影响线上环境
- 效率
 - 前端项目、产测项目接入运维平台统一发布，减少手工操作，提升运维部署效率和上线效率
 - 运维平台增加UAT环境发布，保障工厂测试效率
- 成本
 - 服务器磁盘类型从gp2更换为gp3，存储成本降低20%
 - ELK集群使用ARM架构服务器，成本降低15%
 - 逐步采用saving plan替代预付费实例，提升预付成本的灵活性，避免将来容器化后大量预付费实例不能被使用

业务情况-运维规划


- 安全和稳定
 - 接入AWS WAF，可从网络层面控制流量接入，屏蔽部分恶意请求，提升云端安全性
 - DNS管理平台开发，建立域名管理规范，通过运维平台操作，记录操作记录，一键回滚，提升域名管理效率和安全
 - 外部ELB调整默认路由返回404，避免恶意请求影响云端服务稳定
 - 产测系统迁移到AWS，便于快速扩容，提升系统稳定性
- 效率
 - CMDB平台开发，运维基础资源接入运维平台管理，统一部署规范，避免登录各个AWS账号，降低操作复杂度
 - 推进服务容器化，结合AWS自动伸缩，提高服务扩缩容和回滚效率
- 成本
 - 推进业务服务器改用ARM架构，降低服务器成本
 - 推进服务容器化，提升服务器资源利用率，减少服务器使用量，降低服务器成本

业务情况-健康解决方案规划

- 用户体验:
 - 引入算法工程师，做视频、课程方面的推荐
 - 直播技术和编解码技术，降低视频的时延
- 快速迭代开发
 - 搭建新的平台，独立迭代和演进
 - 敏捷开发和迭代，确保需求快速上线
 - 最小化架构设计，在迭代中进行架构演进
- 安全
 - 付费内容管理调研

团队搭建:

- 由架构师Euan担任健康解决方案架构师
- 继续招聘2个高级后台开发，2个前端开发

A top-down view of a desk with various items. In the top left, a white laptop keyboard is visible. To its right is a row of colorful books standing upright. Below the laptop is a magazine with a blue cover featuring an iPod mini and the text 'iPod и iPod mini Самые популярные iPod-плееры в мире'. In the bottom left, there's a spiral-bound notebook with a pencil and an eraser. In the bottom center, a smartphone with a grid of app icons is placed on top of a magazine. The magazine has logos for Quark, conEdison, and LG, along with some text and circular patterns. The entire scene is dimly lit with a dark overlay.

谢谢