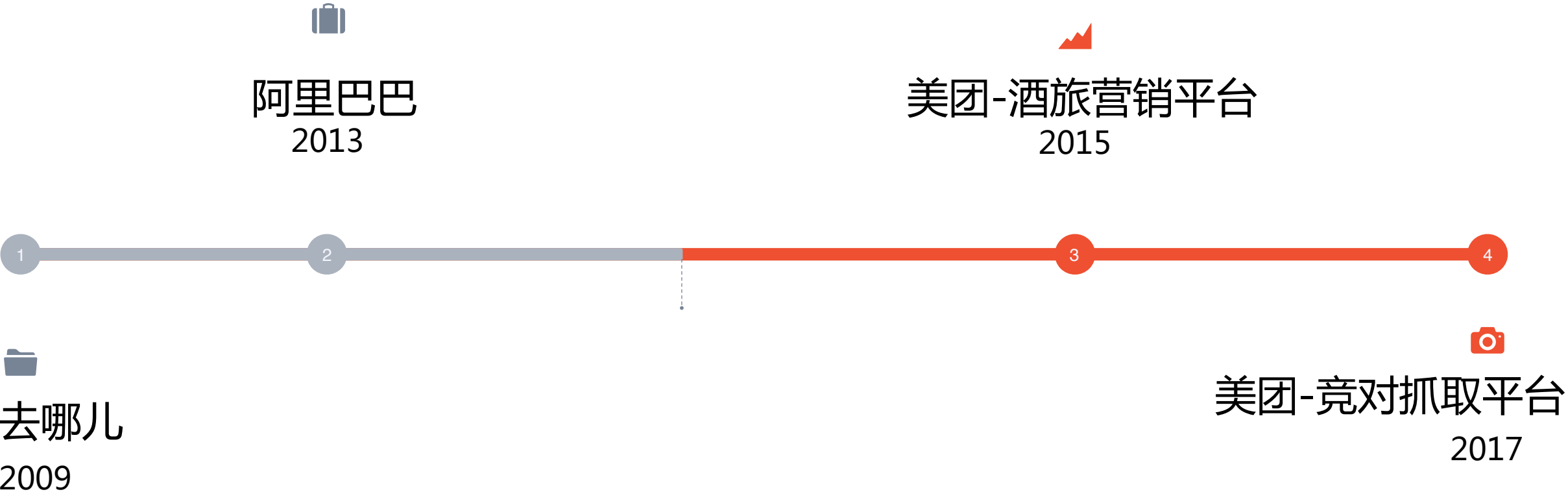




2017秋季目标职级P3-3答辩

酒旅事业群平台研发组-孙鲁宁

个人简介



主要业绩

- 主导酒店促销系统到支撑酒旅全业务线多维度营销平台的发展演变
- 带领团队获得酒旅优秀项目奖5次
- 营销平台获得技术专利1次
- 担任美团技术沙龙讲师1次
- 培养新人晋升15人次
- 担任晋升评委3次
- 影响美团外卖、金融、猫眼营销平台化建设

酒旅营销平台

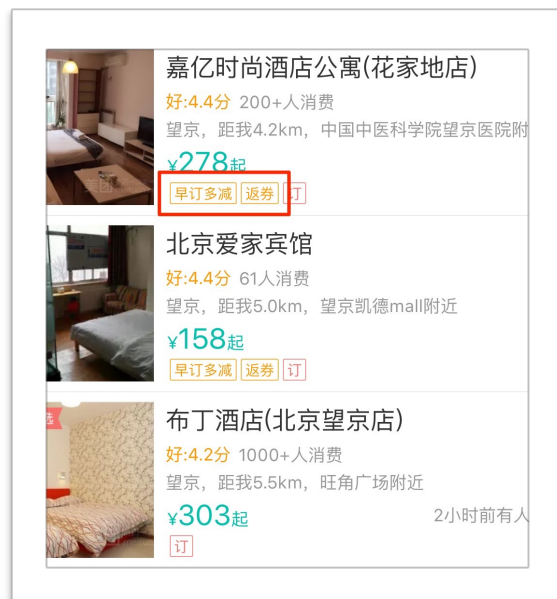
支持酒旅15个业务的40+个子系统调用
峰值QPS 50000,TP999 小于50ms
负责20个子系统的架构设计与运维

酒旅抓取平台

平台化架构演进
团队建设
新人培养

营销业务

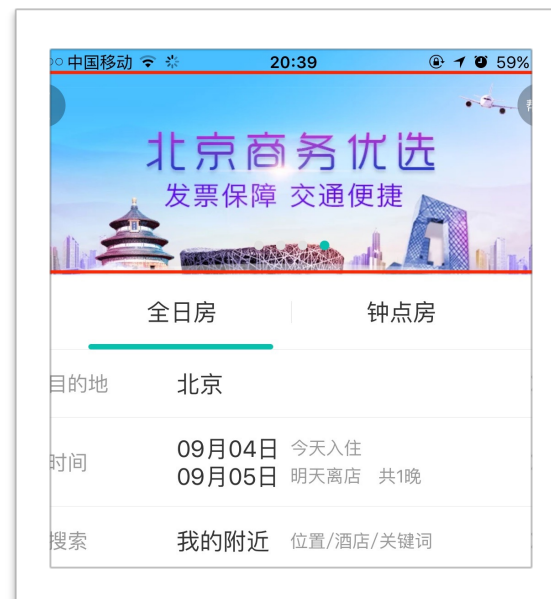
通过高效连接，与客户建立长期关系，争夺认知，获得利润。



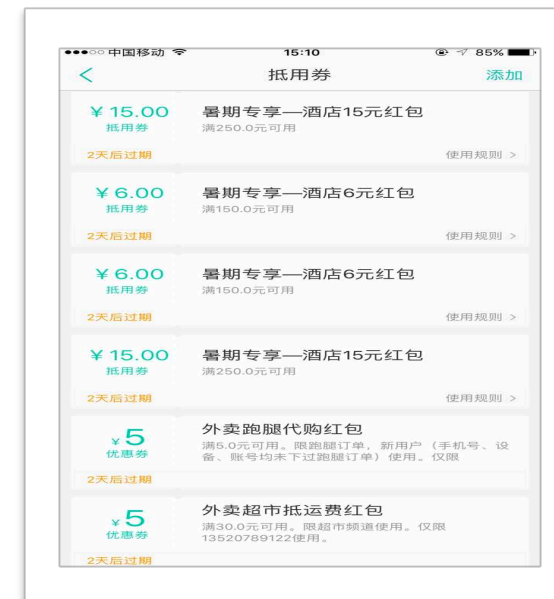
促销
引导决策



触发
流失用户挽回



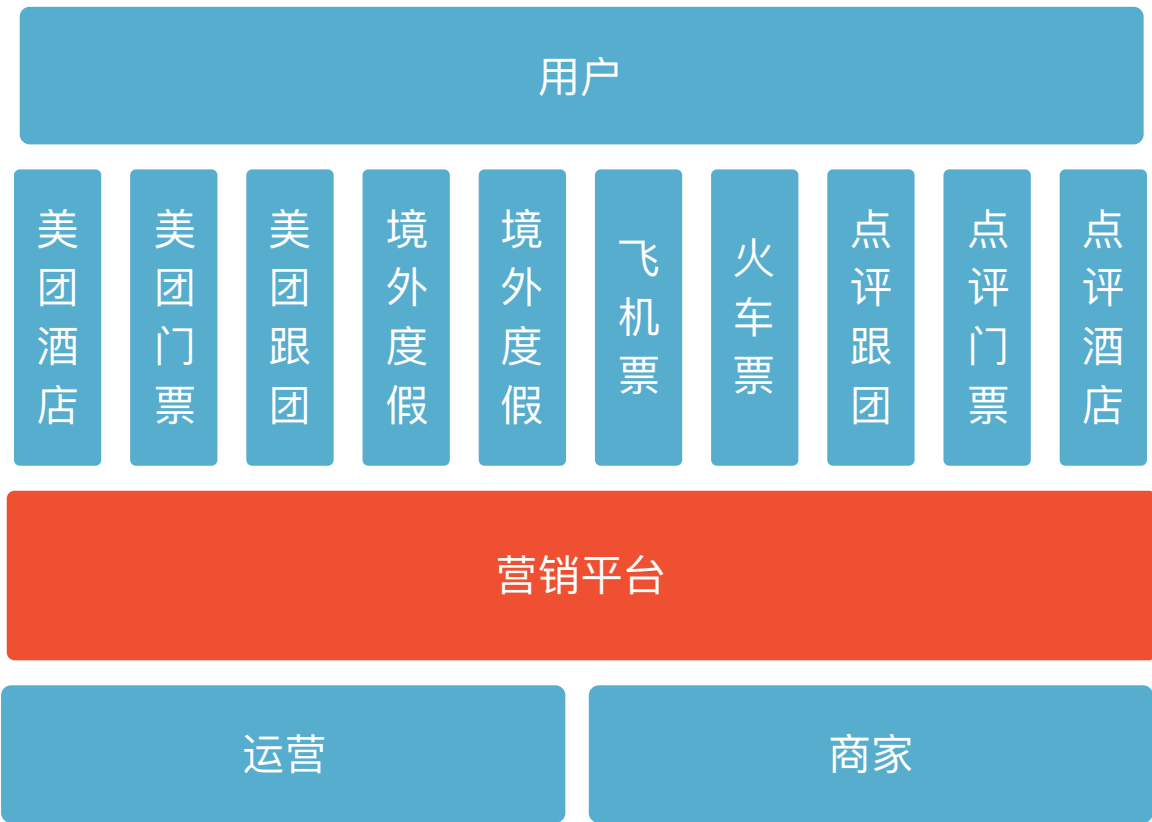
广告
潜在客户挖掘



红包
强化意向

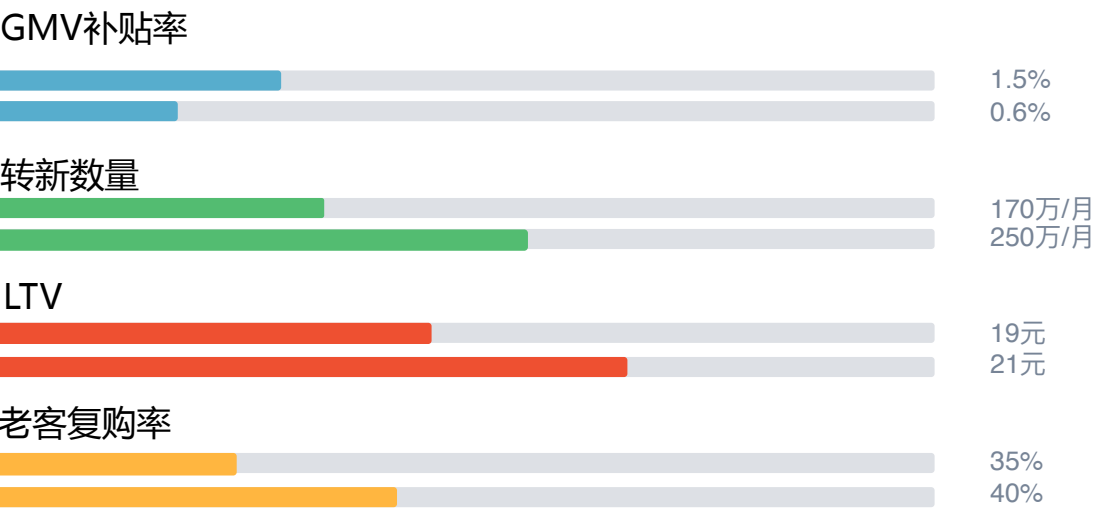
营销平台

连接两亿用户与几十万商家

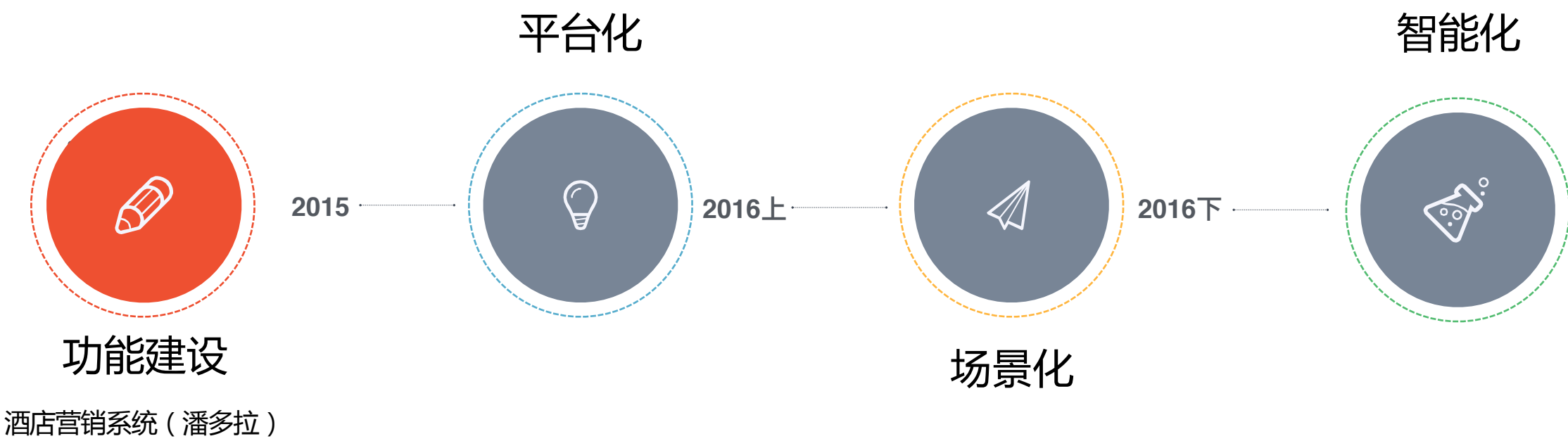


为平台 商家营销提供整合营销方案

帮助平台提升供需匹配效率
帮助商家提升综合收益



营销平台演进



功能建设阶段-挑战

01. 高并发

峰值QPS 4.5W

节假日相比平日5倍以上的流量增长

- ✔ 支持酒旅15个业务线C+B+E全系统调用，系统支撑整个酒旅所有的查询量.

02. 高性能

C端调用TP999小于80ms

- ✔ 业务极度复杂，200+营销规则。10+外部依赖

03. 高可用

可用性9999,年故障时间小于52分钟

- ✔ 20W+在线商家活动 1W+ 运营活动
- ✔ 下单QPS 100+



1.高并发 问题

上游40+系统。日访问量5亿次。峰值QPS 5W
节假日相比平日5倍以上的流量增长

- 流量增长迅猛 扩容依赖中间件
- 资源浪费 高配的节点支持运营编辑
- 部署影响服务可用性
- 责任不明确 对接口可用性缺乏负责人



1.高并发 解决

架构

- 消息驱动
- 读写分离
- 内存是新的存储

环境

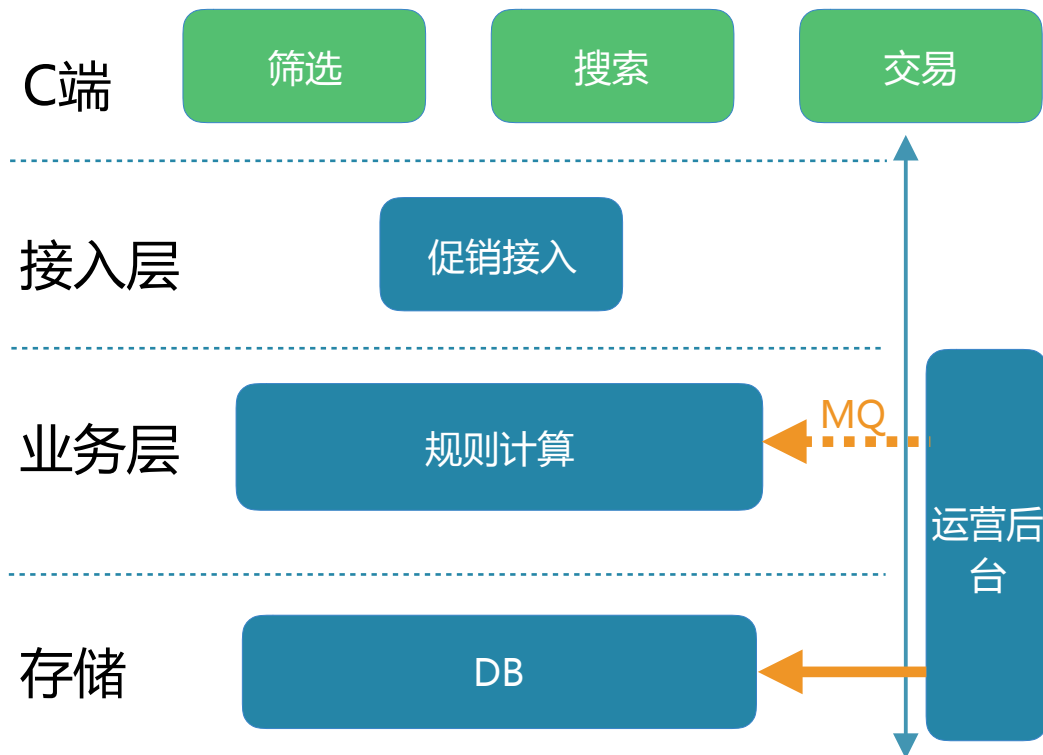
- 与业务方同机房部署 避免跨机房调用
- 虚拟机避免跟资源消耗大的服务部署同一台宿主机
- 网络优化 关注网络耗时

组织

- 明确系统的负责人
- 与存储团队支持维稳小组

链路

- 区分降级与不可降级的服务
- 不可降级服务报警阈值容量一半设置



功能建设阶段-挑战

1.高并发 效果

QPS 5W
TP999 <50MS

- 读写分离降低资源使用：120台 —> 80台
- 内存替代缓存中间件提升扩容速度:1小时 —> 5分钟
- 服务化拆分：服务无单点，水平扩展

功能建设阶段挑战

2.高性能 问题

上游40+系统。日访问量5亿次。峰值QPS 5W

海量计算

20亿次计算/请求

200规则 * 20000活动

* 500 deal = 20亿

GC频繁

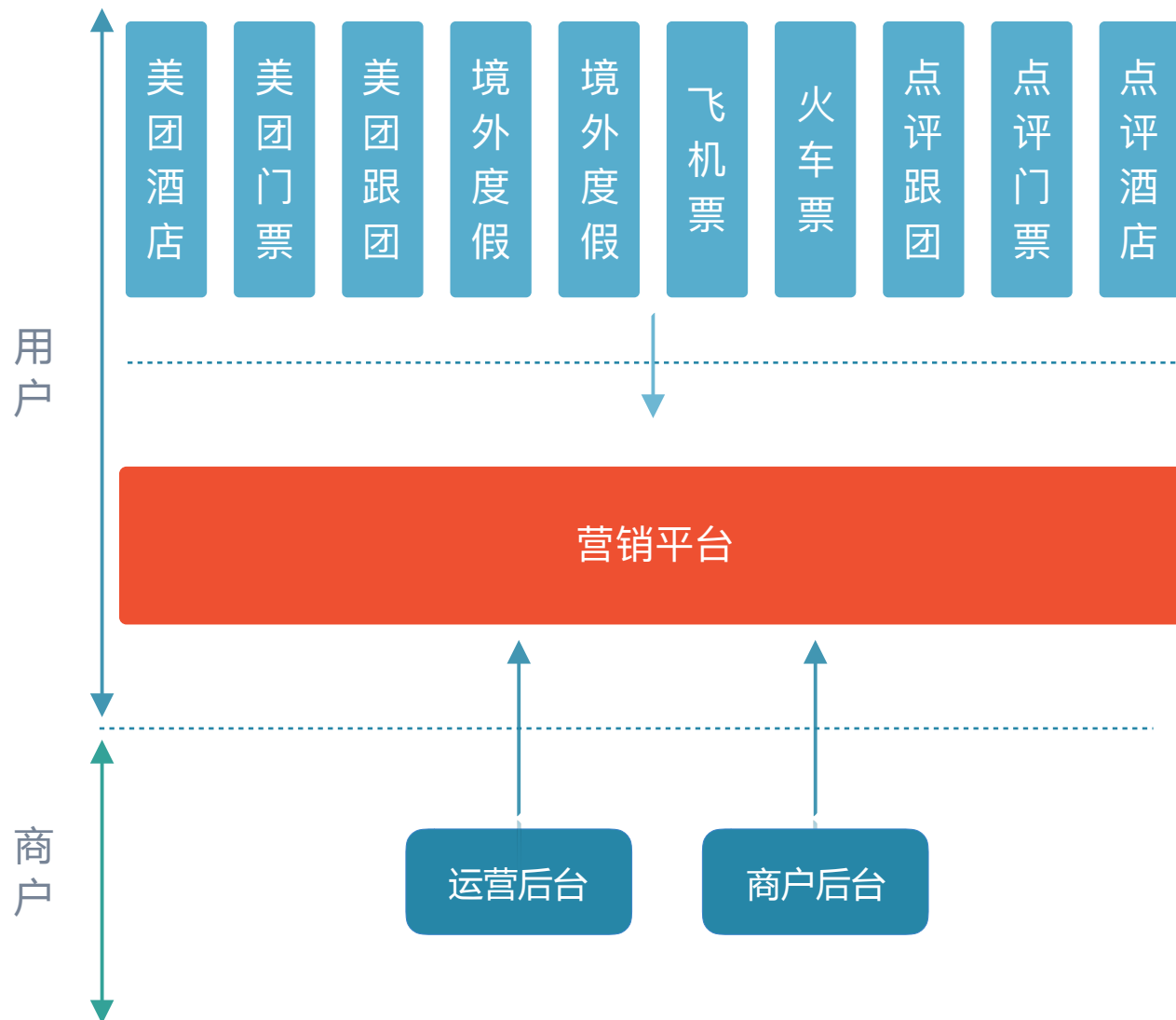
8次/分钟

大量的计算造成CPU

使用率很高。

新建对象频繁造成

GC的负担也很高



功能建设阶段挑战

2.高性能 解决

数据结构

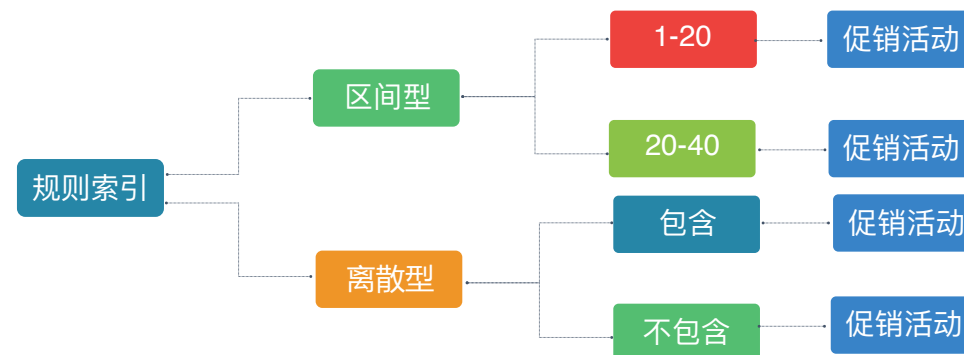
算法

- ➔ 建立规则索引 计算量减少70倍
- ➔ 避免新建对象 GC频率减少8倍
- ➔ 优化算法 CPU使用率减少6倍

• 索引结构

区间型 价格 ≤ 值 < 元

离散型 所选入住城市

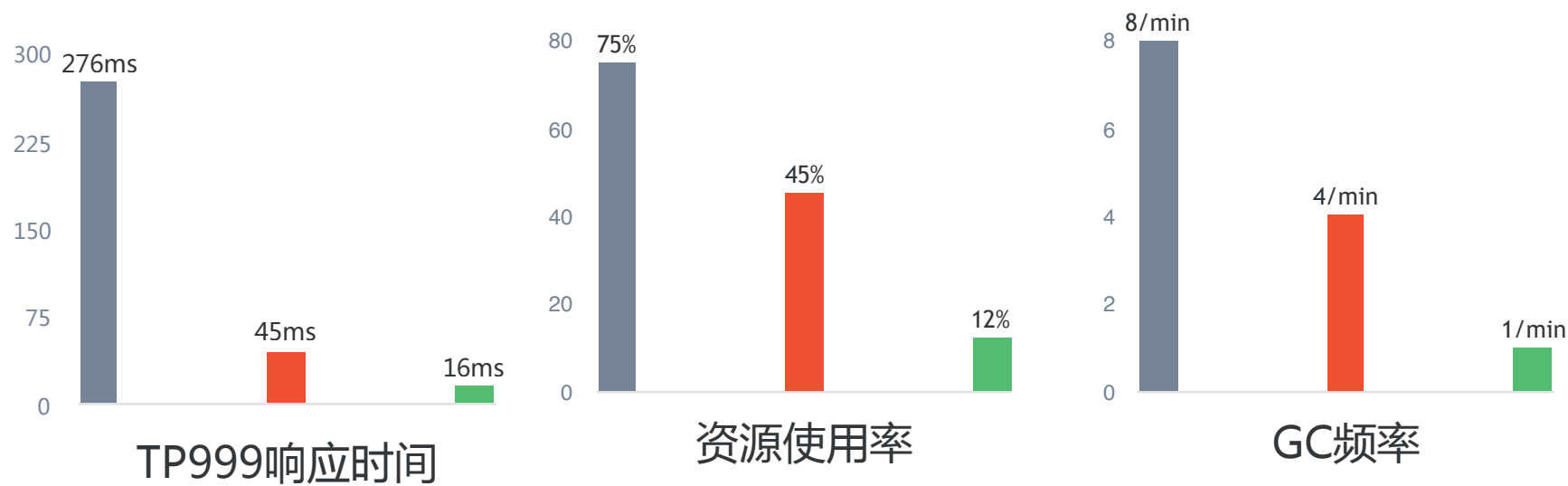


• 算法模型



功能建设阶段挑战

2.高性能 效果



第一版

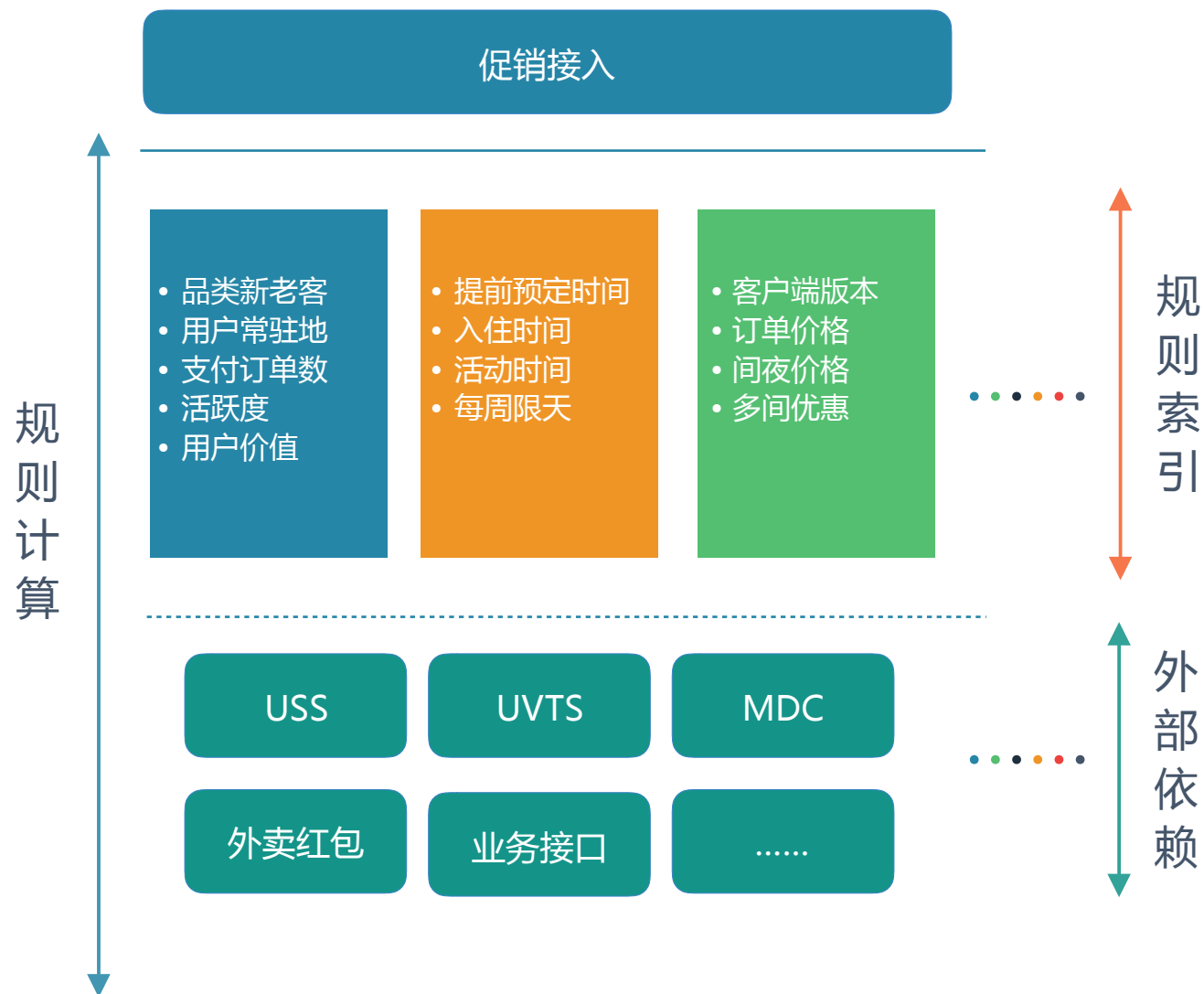
第二版

第三版

3.高可用 问题

可用性9999,年故障时间小于52分钟

- 规则组合复杂
- 不稳定的外部依赖
- 排查问题频繁



3.高可用 解决

监控

- cat falcon
- 核心业务指标

熔断

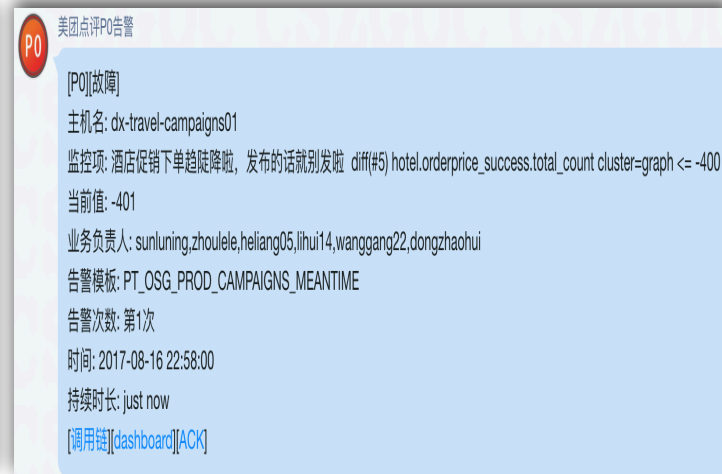
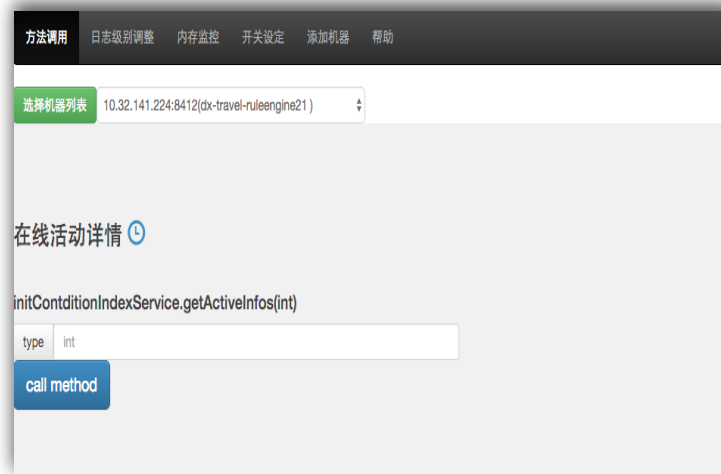
- 超时
- 开关
- Hystrix

工具

- 运营问题排查系统
- 开发问题提排查系统

流程

- 发布周知
- 灰度检查



3.高可用 效果

0

外部依赖故障次数

- 线上问题发现：1个小时 —> 1分钟
- 线上问题排查：2个小时 —> 1分钟
- 运营常见问题：人工处理 —> 自动化

技术设计总结

数据结构

- 业务映射到数学模型上
- 简化业务逻辑的表达
- 实现了优化和一致性约束

工具

- 线上方法调用 内存占用查看
- Hystrix组件 隔断不稳定依赖



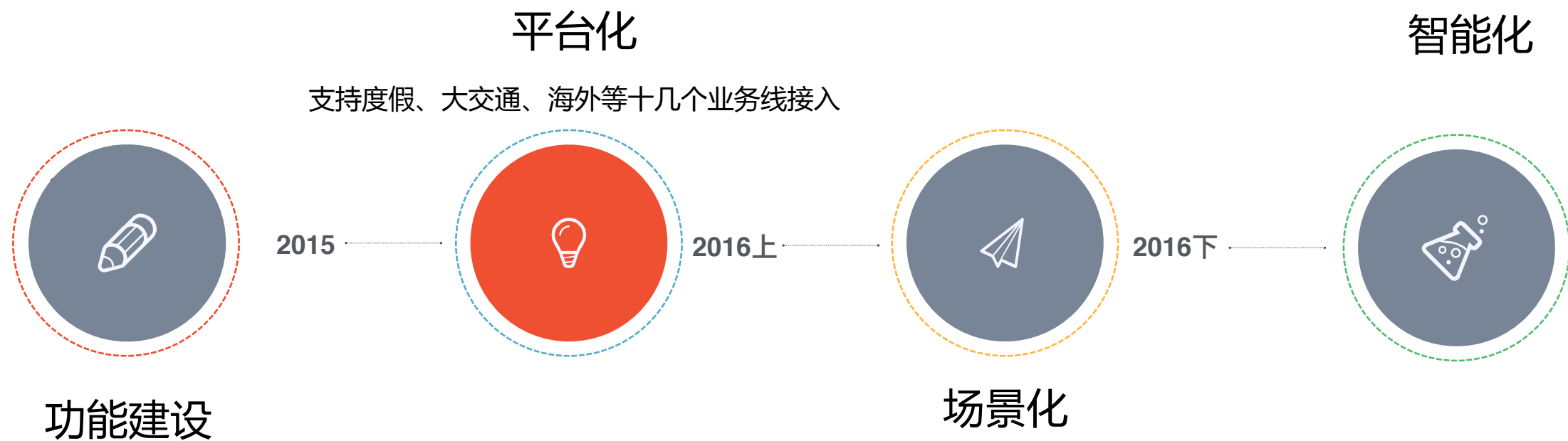
算法

- 分析问题确定本质
- 倒过来看问题

架构

- 读写分离 提升性能资源、使用率
- 全内存架构 扩容及时
- 服务化 提升复用性和可用性

营销平台演进



平台化挑战

业务模型不一致

- 产品模型不一致
- 业务规则多变
- 优惠规则多样

需求玩法不一致

- 酒店每间夜减
- 度假每成人减

业务发展不一致

- 住宿领跑各个业务

接入复杂

- 后端10PD
- 前端5PD



目标

简化新业务接入速度

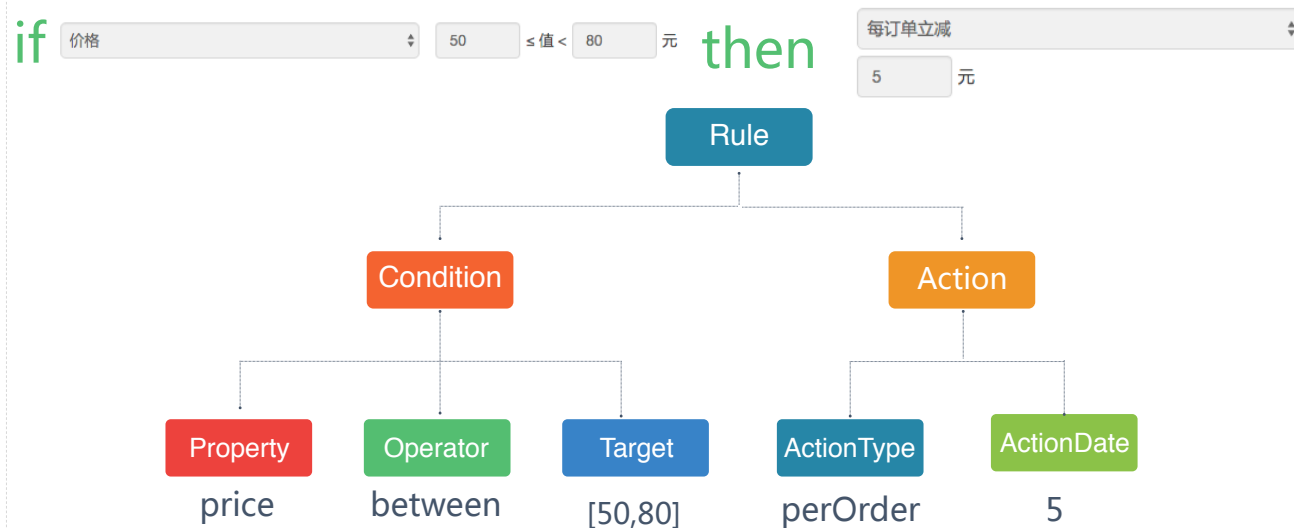
如何让使用营销的平台最小化成本接入

能力复用

已有能力如何快速复用到其他业务线

平台化挑战

1.业务模型不一致 解决



解决

- 抽象规则元信息
- 打通系统各个环节 元信息共享

平台化挑战

1.业务模型不一致 解决



潘多拉规则管理平台

规则

Search... Q

ROOT

部门

产品类型

促销类型

索引类型

PPRelation

运算符相关

Target相关

属性相关

Action相关

条件相关

规则相关

规则模板

ID: ID 部门: 输入名称搜索 产品类型: 输入名称搜索 Action: 输入名称搜索

名称: 名称 描述: 描述 更新时间: 操作人: 操作人

Q 查找 清空 刷新

| ID | 部门 | 产品类型 | Action | 条件组 | 名称 | 描述 | 创建时间 | 更新时间 | 操作人 |
|----|----|------|--------|-------------------------------|----------|--------|---------------------|---------------------|---|
| 1 | 1 | 1 | 1 | 酒店预付-酒店范围 酒店预付-用户范围 酒店预付-产品范围 | 立减活动规则 | 立减活动规则 | 2016-06-24 18:17:32 | 2016-07-07 14:04:15 | 查看 编辑 删除 |
| 2 | 1 | 1 | 1 | 酒店预付-订单 酒店预付-用户范围 | 酒店预付满减规则 | 满减活动规则 | 2016-07-04 15:05:35 | 2016-07-07 14:08:42 | daimao 查看 编辑 删除 |
| 3 | 1 | 1 | 2 | 酒店预付-用户范围 | 返抵用券规则 | 返抵用券 | 2016-07-04 15:14:05 | 2016-07-04 15:14:05 | daimao 查看 编辑 删除 |
| 4 | 1 | 1 | 2 | 酒店预付-订单 | 满赠抵用券规则 | 满赠抵用券 | 2016-07-04 15:16:09 | 2016-07-04 15:16:09 | daimao 查看 编辑 删除 |
| 5 | 1 | 1 | 9 | 酒店预付-酒店范围 酒店预付-用户范围 酒店预付-产品范围 | 折扣活动规则 | 折扣活动 | 2016-07-04 15:17:28 | 2016-07-07 14:57:35 | daimao 查看 编辑 删除 |



解决

- 建立规则开放平台创建规则元信息
- 搭建规则开放平台 允许业务自创规则

平台化挑战

2.业务快速接入 解决

配置化

- 业务方使用那些规则业务灵活配置
- 新配置上线零发布

标准化

- 元数据配置标准化；统一后台管理
- 接口标准化；15个业务线共用一个接口 减少业务方学习成本 减少开发运维成本

潘多拉规则管理平台

规则

Search...

ROOT

部门

产品类型

促销类型

索引类型

PPRelation

运算符相关

Target相关

属性相关

Action相关

条件相关

规则相关

规则

规则模板

ID: ID 部门: 输入名称搜索 产品类型: 输入名称搜索 Action: 输入名称搜索

名称: 名称 描述: 描述 更新时间: 操作人: 操作人

查找 清空 刷新

| ID | 部门 | 产品类型 | Action | 条件组 | 名称 | 描述 | 创建时间 | 更新时间 | 操作人 | |
|----|----|------|--------|-------------------------------------|----------|--------|---------------------|---------------------|--------|----------|
| 1 | 1 | 1 | 1 | 酒店预付-酒店范围 酒店预付-用户范围 酒店预付-产品范围 | 立减活动规则 | 立减活动规则 | 2016-06-24 18:17:32 | 2016-07-07 14:04:15 | | 查看 编辑 删除 |
| 2 | 1 | 1 | 1 | 酒店预付-订单 酒店预付-用户范围 | 酒店预付满减规则 | 满减活动规则 | 2016-07-04 15:05:35 | 2016-07-07 14:08:42 | daimao | 查看 编辑 删除 |
| 3 | 1 | 1 | 2 | 酒店预付-用户范围 | 返抵用券规则 | 返抵用券 | 2016-07-04 15:14:05 | 2016-07-04 15:14:05 | daimao | 查看 编辑 删除 |
| 4 | 1 | 1 | 2 | 酒店预付-订单 | 满赠抵用券规则 | 满赠抵用券 | 2016-07-04 15:16:09 | 2016-07-04 15:16:09 | daimao | 查看 编辑 删除 |
| 5 | 1 | 1 | 9 | 酒店预付-酒店范围 酒店预付-用户范围 酒店预付-产品范围 | 折扣活动规则 | 折扣活动 | 2016-07-04 15:17:28 | 2016-07-07 14:57:35 | daimao | 查看 编辑 删除 |

3.能力复用 解决

平台化意识

- 关注架构设计能力复用能力，评估业务接入成本。宁可多做一点，减少业务方开发量。

平台化设计

- 能力下沉。避免功能为只为某个业务服务
- 系统自运营能力考核

潘多拉规则管理平台

Search...

Q

ROOT

部门

产品类型

促销类型

索引类型

PPRelation

运算符相关

Target相关

属性相关

Action相关

条件相关

规则相关

规则

规则模板

规则

ID: ID

部门: 输入名称搜索

产品类型: 输入名称搜索

Action: 输入名称搜索

名称: 名称

描述: 描述

更新时间:

操作人: 操作人

Q 查找

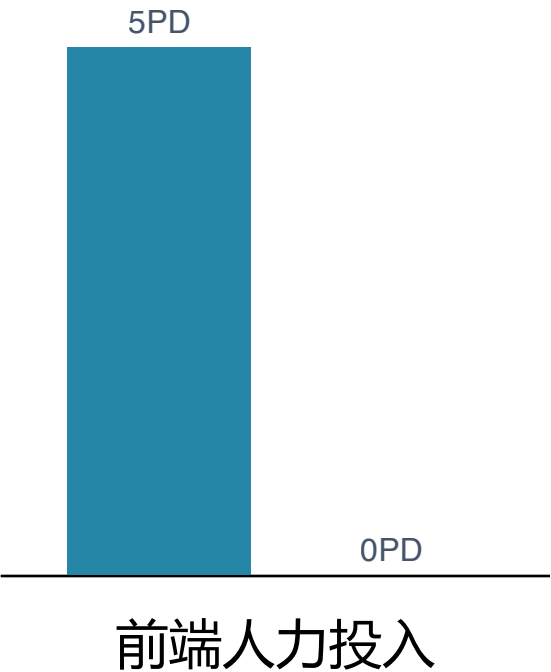
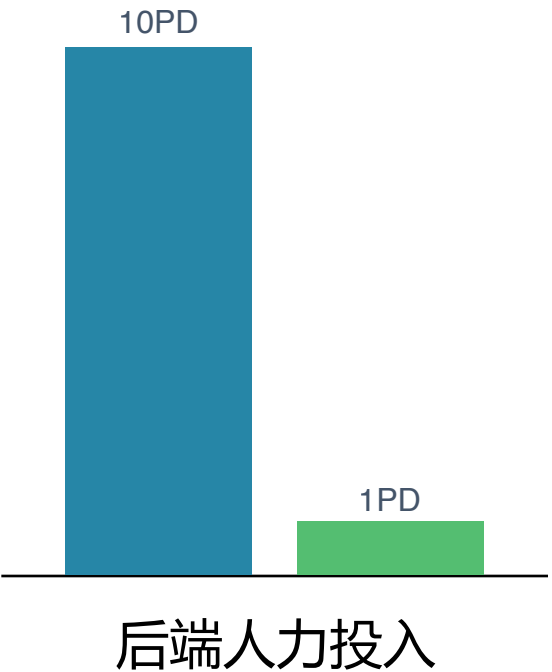
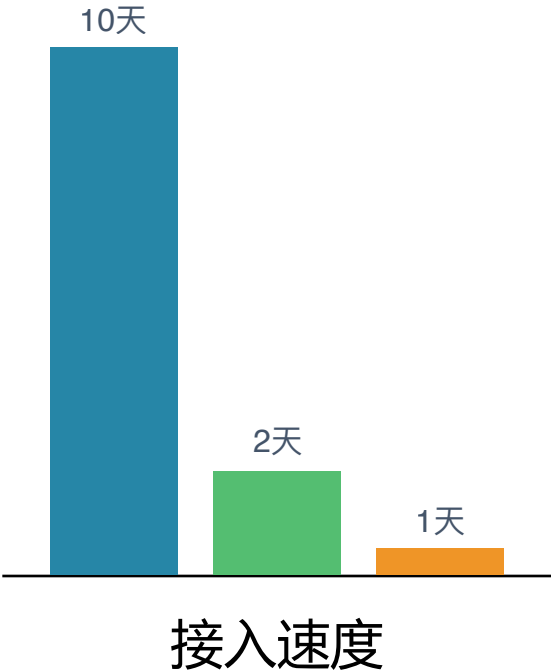
清空

刷新

| ID | 部 门 | 产 品 类 型 | Action | 条件组 | 名称 | 描述 | 创建时间 | 更新时间 | 操作人 | |
|----|--------|------------------|--------|-------------------------------------|----------|--------|---------------------|---------------------|--------|--|
| 1 | 1 | 1 | 1 | 酒店预付-酒店范围 酒店预付-用户范围 酒店预付-产品范围 | 立减活动规则 | 立减活动规则 | 2016-06-24 18:17:32 | 2016-07-07 14:04:15 | | 查看 编辑 删除 |
| 2 | 1 | 1 | 1 | 酒店预付-订单 酒店预付-用户范围 | 酒店预付满减规则 | 满减活动规则 | 2016-07-04 15:05:35 | 2016-07-07 14:08:42 | daimao | 查看 编辑 删除 |
| 3 | 1 | 1 | 2 | 酒店预付-用户范围 | 返抵用券规则 | 返抵用券 | 2016-07-04 15:14:05 | 2016-07-04 15:14:05 | daimao | 查看 编辑 删除 |
| 4 | 1 | 1 | 2 | 酒店预付-订单 | 满赠抵用券规则 | 满赠抵用券 | 2016-07-04 15:16:09 | 2016-07-04 15:16:09 | daimao | 查看 编辑 删除 |
| 5 | 1 | 1 | 9 | 酒店预付-酒店范围 酒店预付-用户范围 酒店预付-产品范围 | 折扣活动规则 | 折扣活动 | 2016-07-04 15:17:28 | 2016-07-07 14:57:35 | daimao | 查看 编辑 删除 |

平台化挑战

平台化效率提升

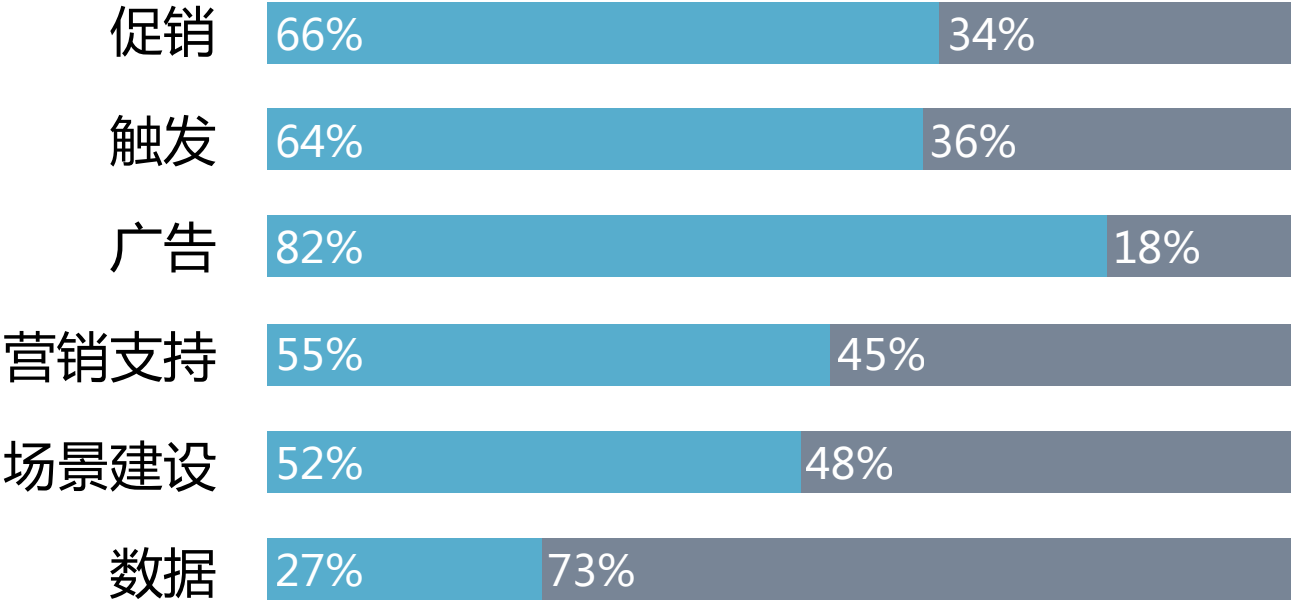
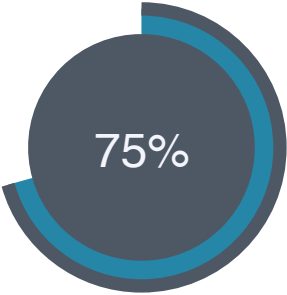


平台化挑战

平台化 业务支持



业务线功能覆盖率



■ 已接入
■ 待接入

平台化设计总结

模型抽象

- 提取业务共性 建立规则模型
- 平台化业务共建

标准化

- 标准化接口 语义统一 减少业务学习成本
- 降低开发运维成本



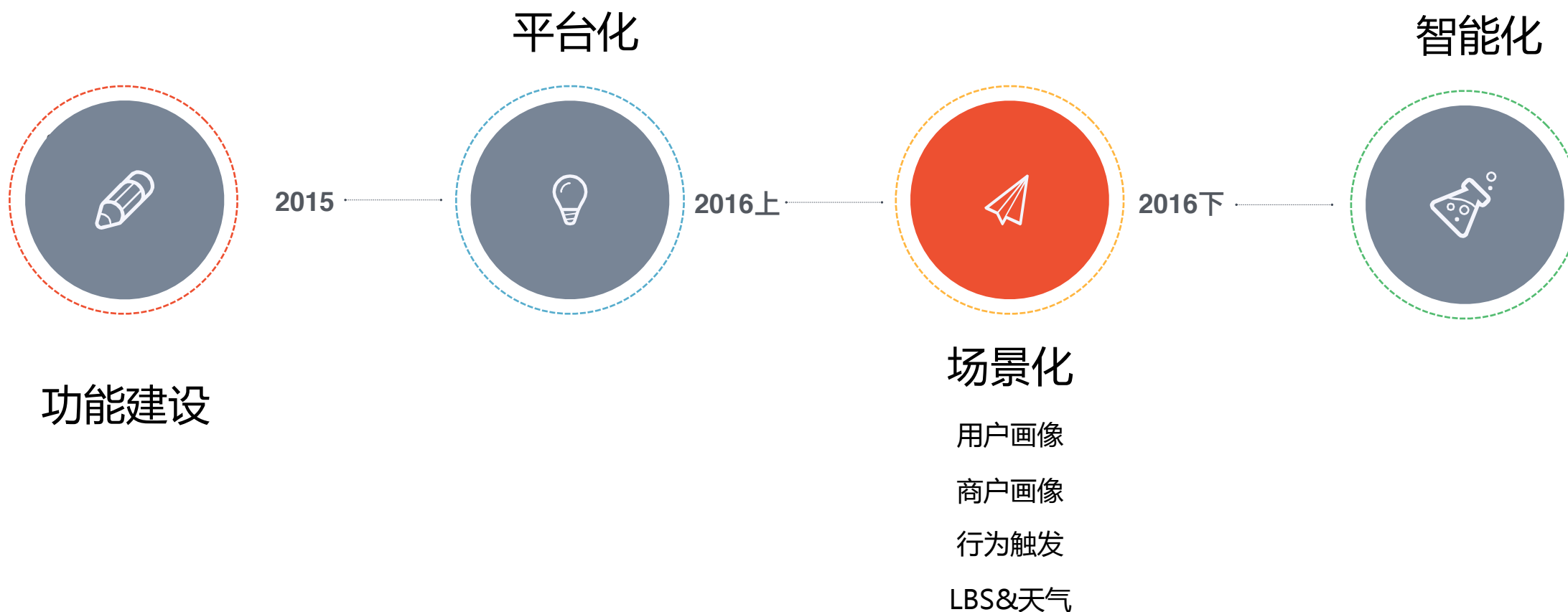
配置化

- 业务配置化，减少开发成本
- 业务接入0成本

意识

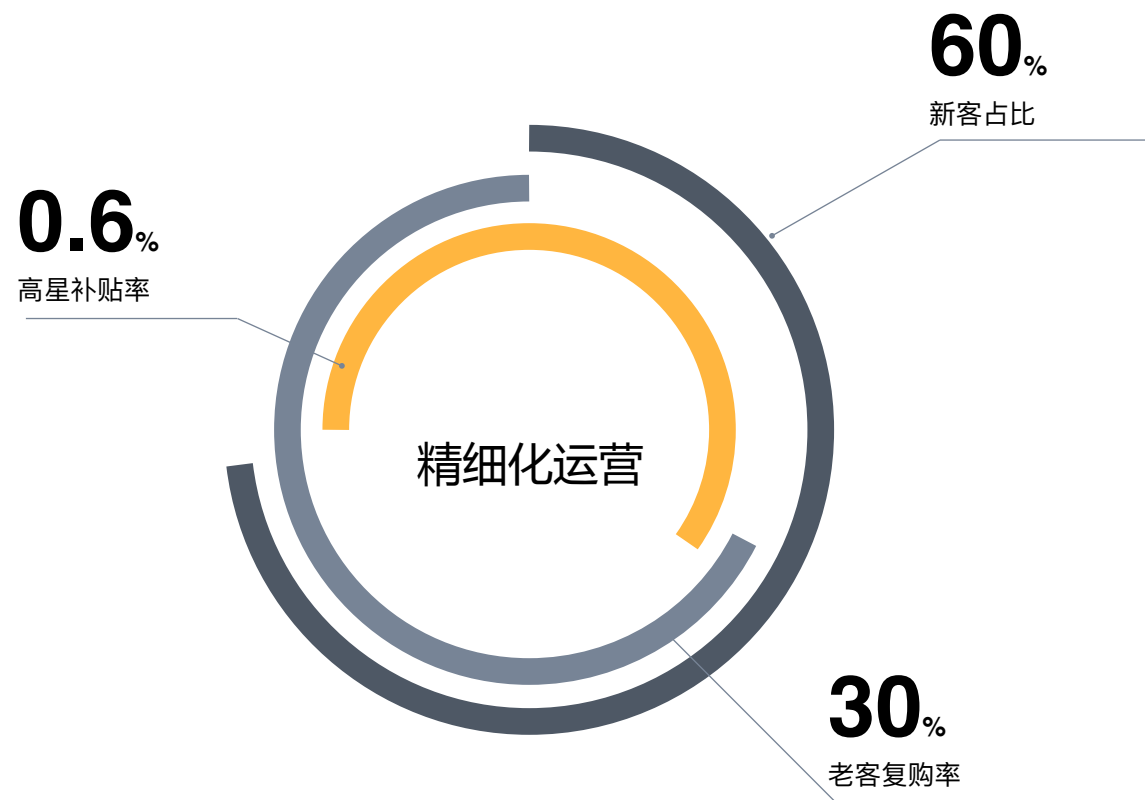
- 平台化架构意识
- 关注业务数据，帮助业务提升营销能力
- 系统自运营能力

营销平台演进



场景化建设背景

千篇一律的营销的方式不适合对资金的充分使用



业务目标差异

- 新客抓质量 老客抓复购

成本控制

- 单店成本商家出，跨店成本平台出
- 本异地差异化补贴
- 用户质量差异化补贴

玩法扩展

- 天气营销
- 商圈营销
- 生日关怀



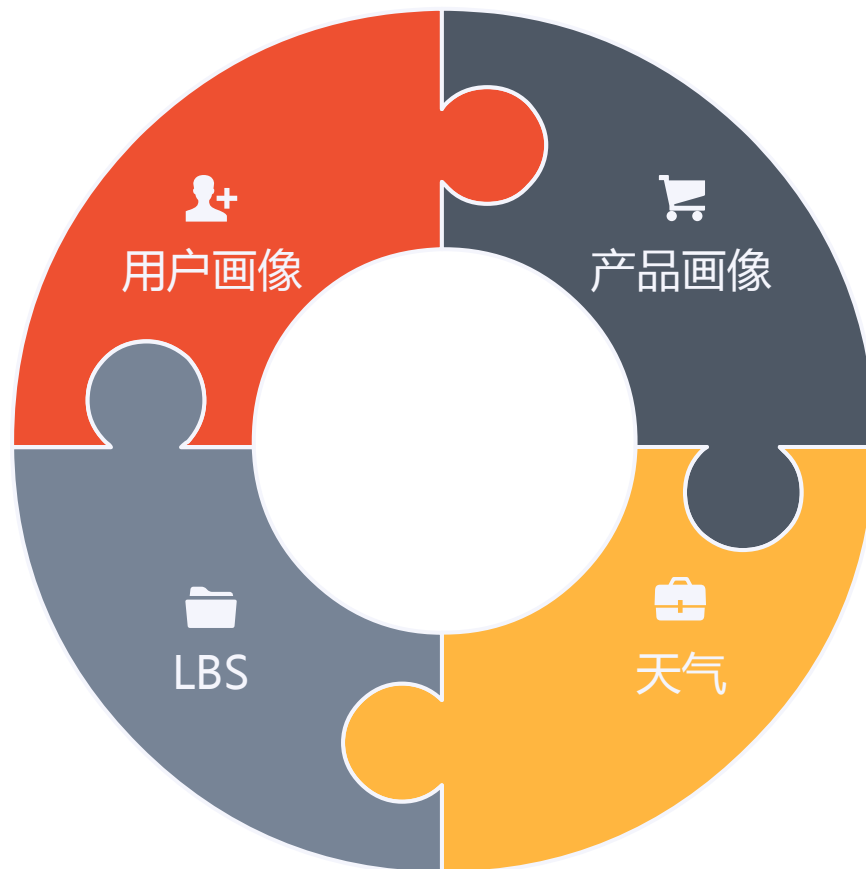
场景化建设目标

用户画像

- 品类新老客
- 流式倾向用户
- 高价值用户

LBS

- 商圈触发
- 景区触发
- 商家POI附近



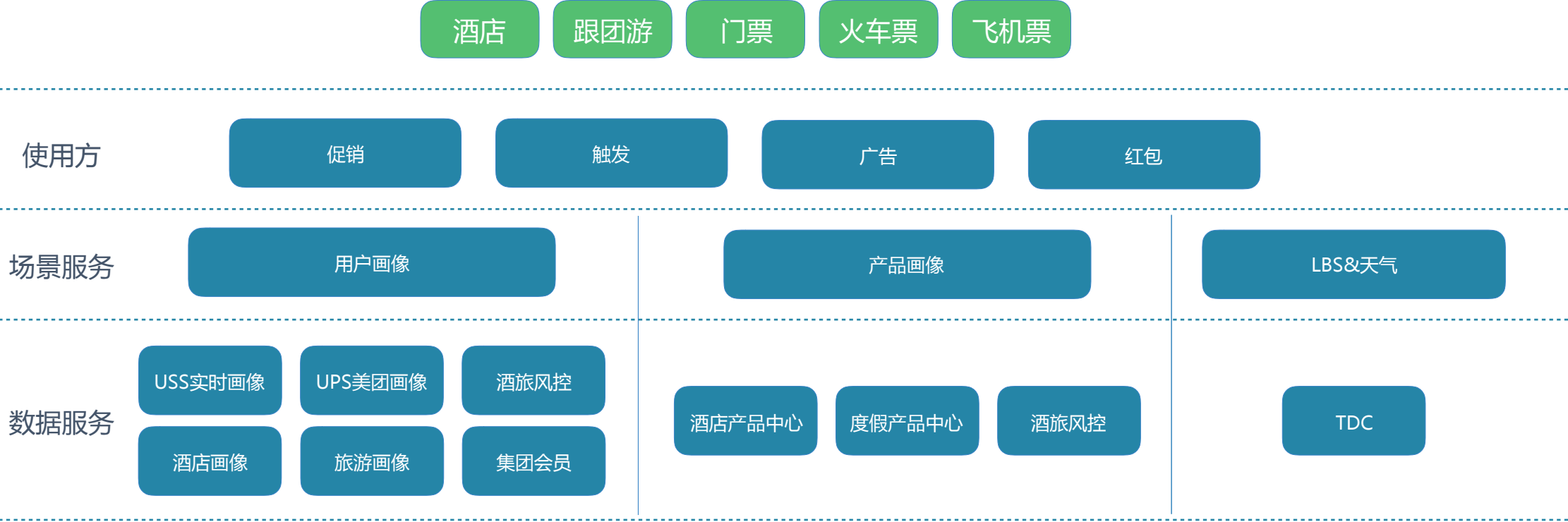
产品画像

- 产品类型
- 产品利润
- 产品毛利
- 产品属性 (品牌 所在地)

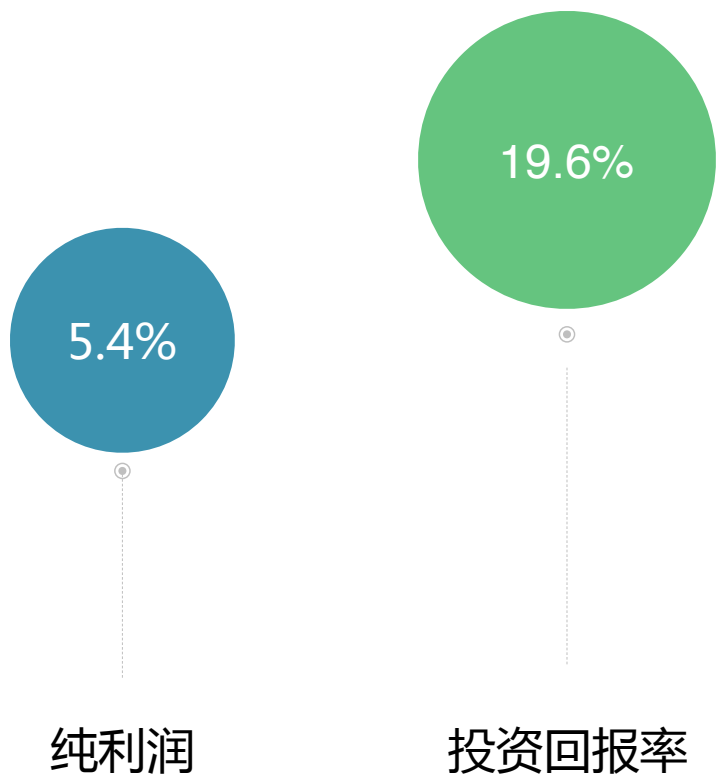
天气

- 天气关怀
- 场景广告

场景化架构图



场景化 成果

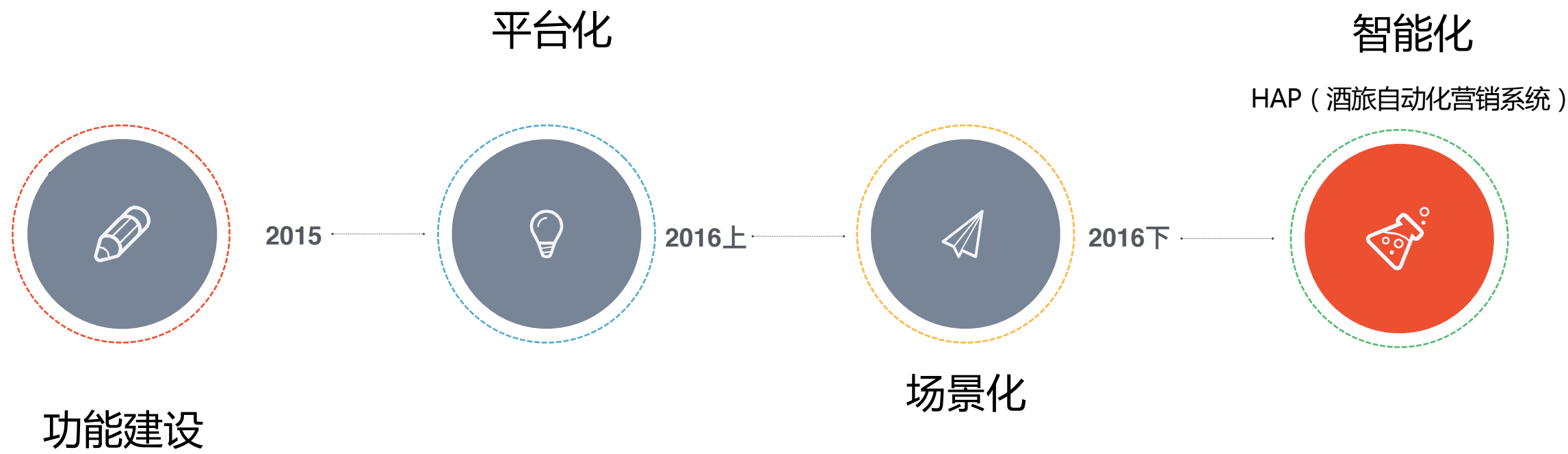


用户分层补贴

150W/天 纯利润

通过区分画像对用户进行差异化补贴

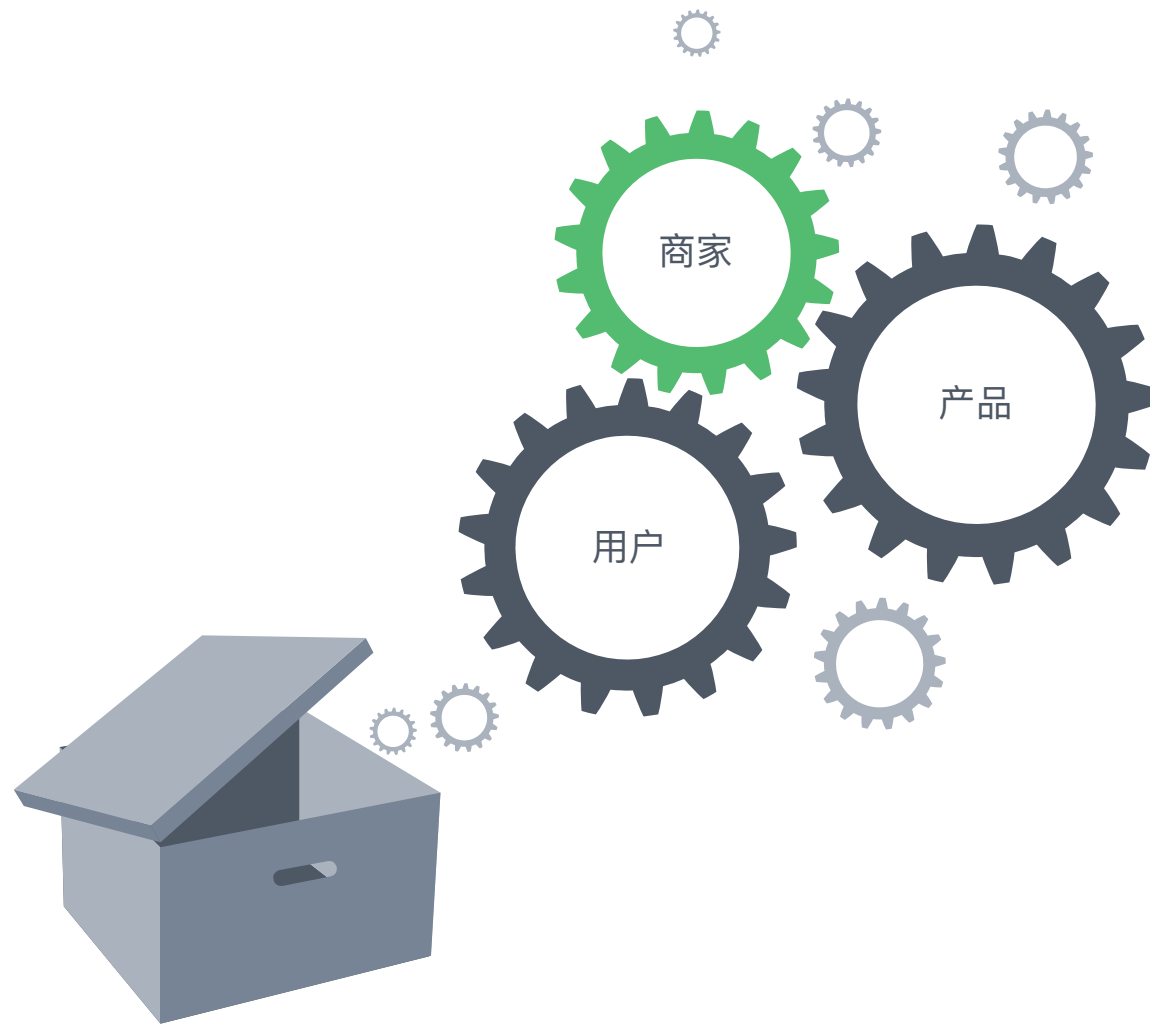
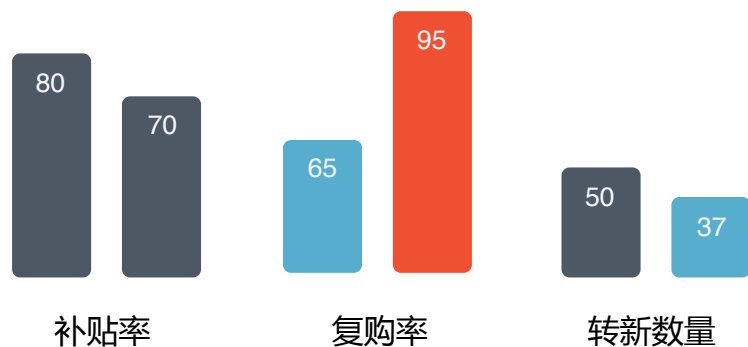
营销平台演进



营销智能化

营销投放**自动化**，提升运营效率

营销策略**智能化**，提升费用效率、流量效率



营销智能化思路

01. 建立假设

提出运营想法或者改进建议

02. 定义指标

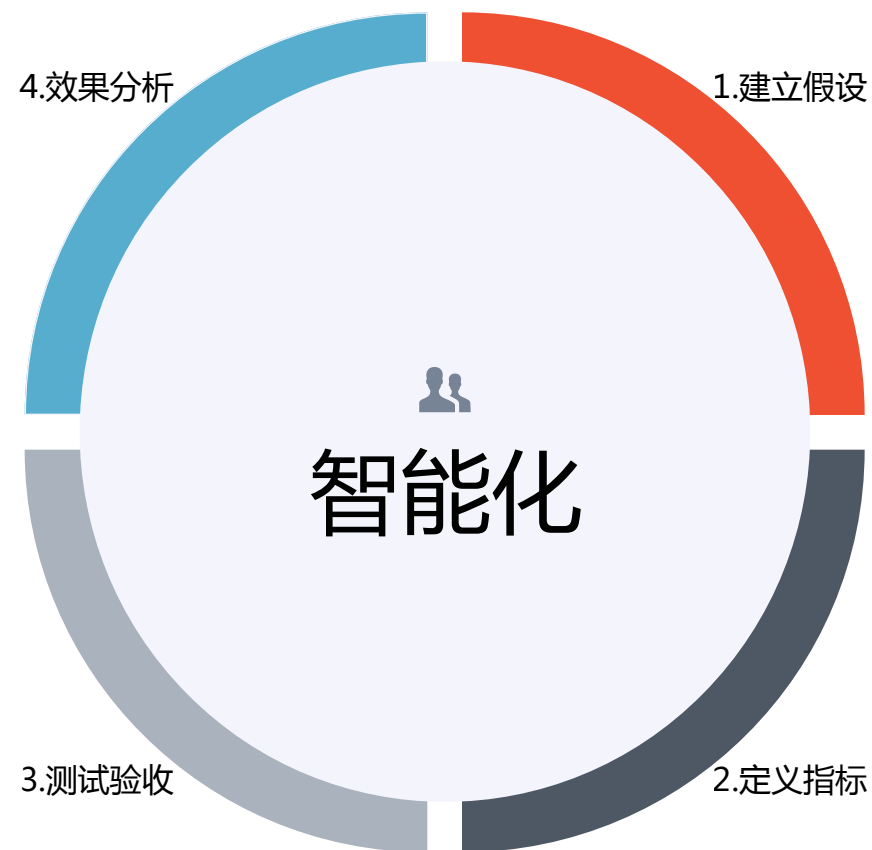
评估受该变更影响的用户范围，
确定考核指标和预期目标

03. 测试验收

检测执行合理性确保实验正确
配置，并评估测试数据

04. 效果分析

根据测试数据分析评估影响和收
益，并决定是否扩量



营销智能化-任务拆解



用户价值分层系统

复用场景化建设中的用户画像
丰富用户画像规则



资金管理系统

营销资金管理在线化 系统化



产品价值分层系统

复用场景化建设中的用户画像
丰富产品画像规则



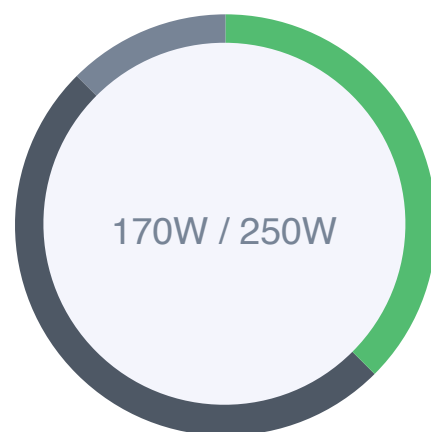
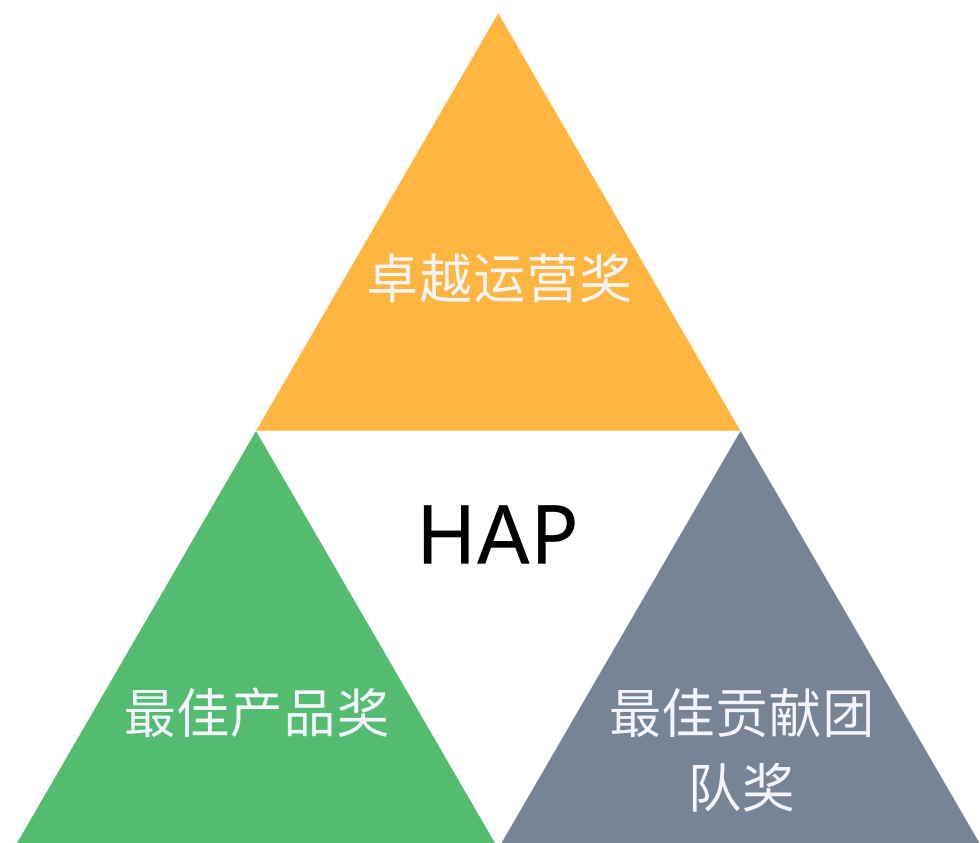
效果评估系统

量化业务指标
效果评估在线化 实时化

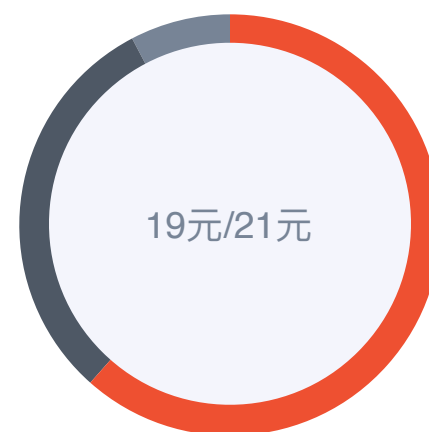
营销智能化整体架构



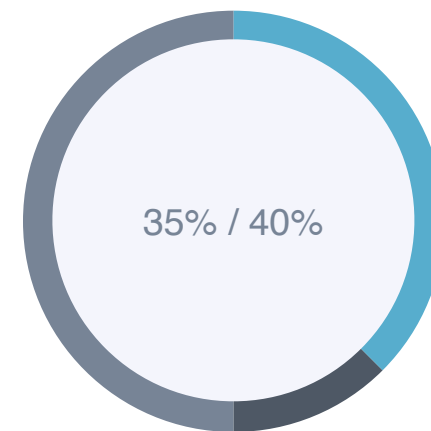
营销智能化结果



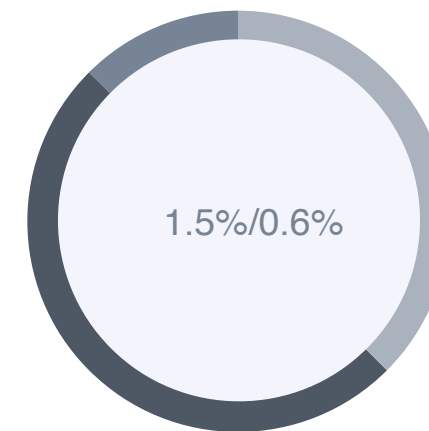
月转新数量



转新当月毛利



老客复购率



补贴率

总结

高并发



集群QPS 5W
TP999 小于80ms

高可用



服务可用性99.99%
服务可降级 依赖可熔断
完善监控报警 快速定位解决问题

可扩展



架构简单清晰
系统间低耦合
服务无状态，水平扩展，快速升级

可复用



系统功能可复用
低维护成本
提升开发效率

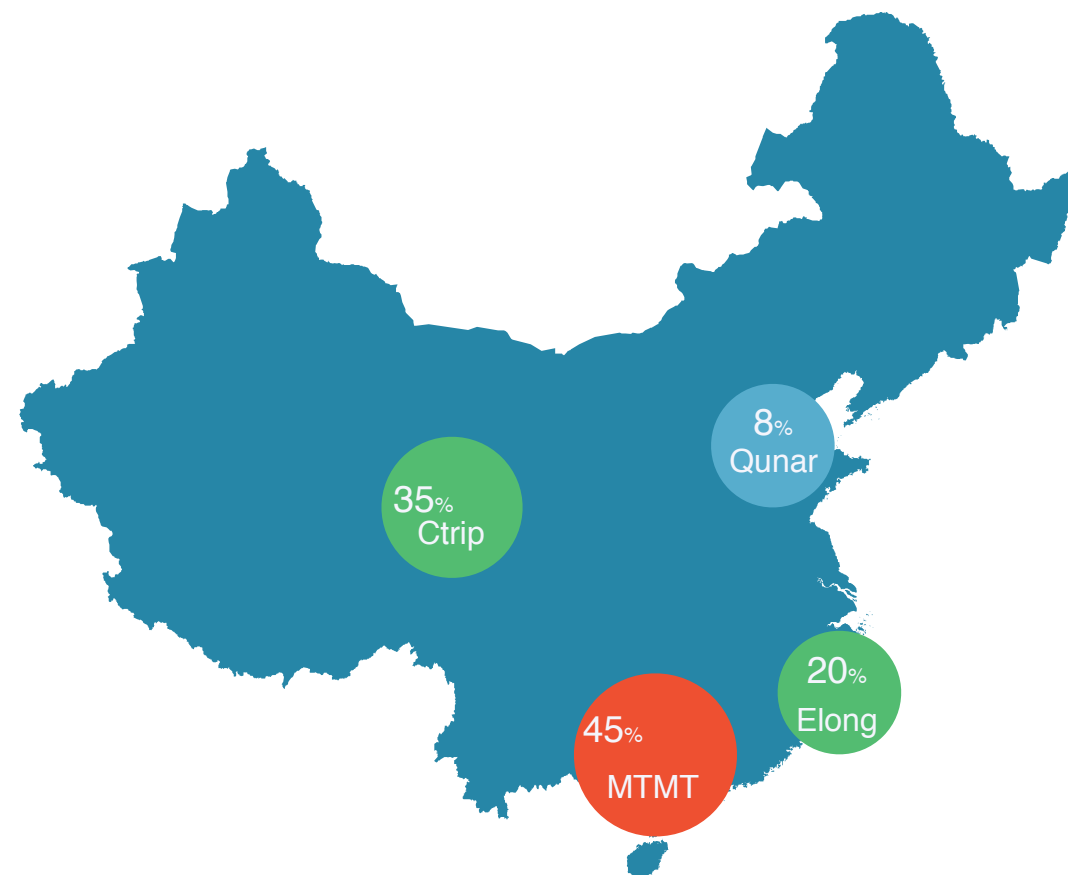
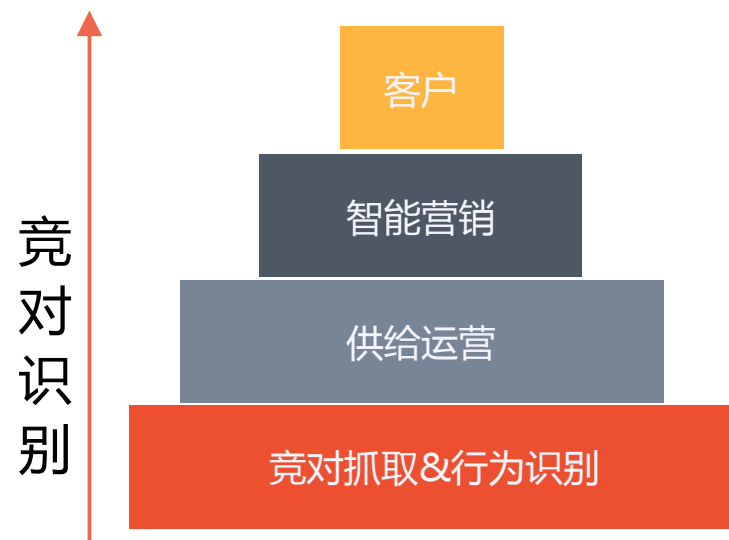
平台化



业务接入1PD
开放共建
能力复用

未来规划

抓取平台建设——最好的防守是进攻



抓取平台业务目标

短期指标

抓取能力提升1倍
多渠道容灾



多渠道抓取能力
登录抓取能力

抓取能力



代理扩展标准化
提供代理容量监控

代理能力

长期指标

容灾能力提升3倍
交付速度从月到天
抓取调度智能化



6个以上的代理渠道
扩展能力从天到小时

代理能力



清洗配置化
数据解析配置化
抓取能力配置化

配置化



抓取调度系统建设

智能化

抓取平台技术目标

短期指标

服务可用性999

提升系统复用性60%



完善报警，早发现问题
服务集群化，无单点

可用性



建立业务抽象，提复用
批处理流式化，提时效

架构

长期指标

运维效率提升2倍

问题排查缩短到分钟



完善运营问题排查工具

工具



抓取验证在线化
多规则并行验证

效率



抓取中间结果可视化

可视化

个人成长



个性化

优先业务需求
需求抽象,做平台化建设



平台化

沉淀业务共性



优先级

资源有限,先做重要的事。
形成一套高效的排期管理机制



沟通反馈
了解运营迫切需要的是什么
系统那里做的不好



结果导向
从业务拿到结果,激励开发



架构先行
了解业务规划
架构为业务铺路



个人影响力



促销大讲堂
了解业务
分享技术



跨BG交流分享
交流学习架构经验
自动化促销实现



订单大讲堂
促销对订单流程影响



美团技术沙龙
提升外部影响力

酒旅促销系统潘多拉规则引擎系统分享

负责人: 周乐乐
讲师: 周乐乐 (酒店旅游事业群 - 技术研发部)
时间: 2016-07-27 周三 10:30 - 11:30
地点: 三亚厅 (望京研发园-3层-F座-南)
名额: 15人

潘多拉规则抽象及管理平台搭建分享

负责人: 代茂
讲师: 代茂 (酒店旅游事业群 - 技术研发部)
时间: 2016-08-03 周三 10:33 - 11:33
地点: 昆明厅 (望京研发园-2层-G座-北)
名额: 15人



人才培养

研究职级标准和能力模型或TC评价考虑团队核心KP能力

- 制定培养计划，缺什么补什么。

定期review，1V1沟通

- 计划内容包括：目标、内容、时间、参与者、结果。

充分发挥组内所有人员的特长和搭配

- 李辉 – 细节 周乐乐 – 算法 代茂 – 组件 何樑 – 工程

培养每个人的owner意识

- 每个人都有质疑系统架构的权利，并且每个人都可以提出合理化建议并实施。

团队晋升通过 **15人次**

1. 李辉顺利晋升P3-1
2. 代茂顺利晋升P2-3
3. 乐乐顺利晋升P2-3 同期成绩第一名
4. 逸帆顺利晋升P2-3
5. 何樑顺利晋升P2-2 同期成绩第一名

总结



营销平台架构设计

主导多个高性能 平台化系统设计
QPS0000 TP99小于0ms

01

架构能力



平台化系统设计

平台化系统将系统价值最大化
数据结构与业务解耦设计
性能问题排查与解决

02

方法论建设



系统按时交付

每日站会 及时发现问题
项目总结 避免问题再次发生

03

执行力

结合业务规划系统架构

系统可复用 提升交付速度
沟通业务愿景 对未来可扩展

04

技术规划



组织跨BG技术交流

组织促销技术大讲堂
组织酒旅与外卖的技术交流
美团技术沙龙讲师

05

知识传播



团队晋升通过率80%

发挥特长 优势互补
制定人才培养计划

06

人才培养



Q:为什么要做酒旅的促销系统 与平台促销系统的差别是？

A:贴近业务特性

在用户维度上 接入按照品类新老客、房型偏好

在产品维度上 接入限制品类 限制单价 限制总价

在时间维度上 接入按照提前预定时间 入住时间

Q:CPU使用率是如何下降的？

A:先分析CPU使用率为什么高

- 1 需要计算200个规则之间的交集

- 2 交集运算造成了大量的中间对象 YGC频繁

通过建立规则不包含逻辑 将交集运算改为对所有在线活动的移除。避免交集计算 和中间对象的产生。所以使用率下降。

Q:为什么不用ES drools

A: 配置的是价格大于 这样的规则信息。我们并没有产品具体的价格，所以在场景上我们就不适合用ES。

drools 优化比较复杂。我们的核心模型就2个 代码几百行 维护成本比较低 而且可以针对性的做优化 比如规则的优先级，规则的缓存，规则分组，后期做平台化改造更容易。

对比和drools的性能我们的性能会更好

Q:平台化的具体实施细节

A:

- 1 建立对活动的抽象 定义rule 和 action
- 2 规则匹配计算与rule类型相关 跟具体的规则无关
- 3 外部接口描述传入相当于一个规则的ID 通过ID获取规则的类型等配置信息 规则可随意增加
- 4 打通各个系统 规则元信息共享
- 5 建立规则开放平台 让业务RD开发规则

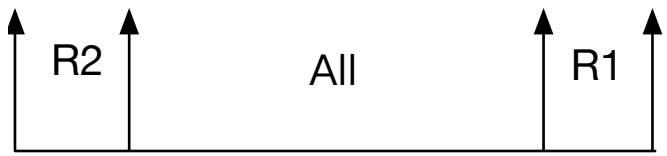
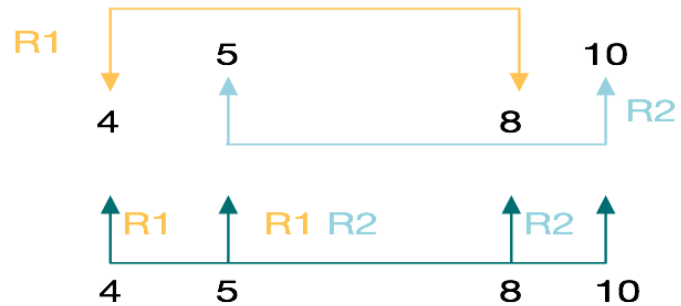
```
struct Property{  
    1:i32          code;          ##属性编码  
    2:string       value;        ##属性值  
    3:optional string desrc;     ##附加描述信息  
}  
  
struct ActiveRequestUnit{  
    1:list<Property> properties;  #属性集合  
    2:optional i64   rowId;      #行标号,表示行id  
    3:required i32   requestTypeCode; #业务方类型  
}
```

Q:订单退款流程怎么与促销关联的。

A: 我们关心的是订单的 支付成功 消费成功 退款 三个状态，我们跟业务约定这三个状态值，通过接口或者消息的形式获取状态变更状态，变更促销的状态。

算法优化

| 规则 | 抽象 |
|--------|----------------------------------|
| 价格大于5 | $(5, +\infty]$ |
| 价格小于5 | $[-\infty, 5)$ |
| 价格不等于5 | $[-\infty, 5) \cup (5, +\infty]$ |



个人成长



个性化

优先业务需求
需求抽象,做平台建设



平台化

沉淀业务共性
能力复用



优先级

资源有限，先做重要的事。
高效的排期管理机制



沟通反馈

了解业务需要的是什么
系统那里做的不好



结果导向

从业务拿到结果，激励开发



架构先行

了解业务规划
架构为业务铺路