

# GOPS

# 全球运维大会

2019  
- AIOps 风向标



深圳站

指导单位：



主办单位：



大会时间：2019年4月12日-13日

大会地址：深圳市南山区圣淘沙大酒店（翡翠店）

# 站在珠峰之巅—淘宝首页百亿级投放系统挑战

陈康贤 2019/04



# 首页场景下的核心挑战

手淘首页要解决什么问题？

淘系巨量DAU下流量的精准匹配问题

需要满足整个阿里生态下，所有业务的分发，以及商业策略的承载

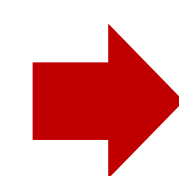
需要

快速地/科学地迭代

&

极端稳定性

# 目录



1

如何快速的迭代

2

如何科学的迭代

3

如何保障首页的高可用

4

Q/A

# 布局动态化—快速迭代

# 快速迭代的主要障碍

超级APP发布模型

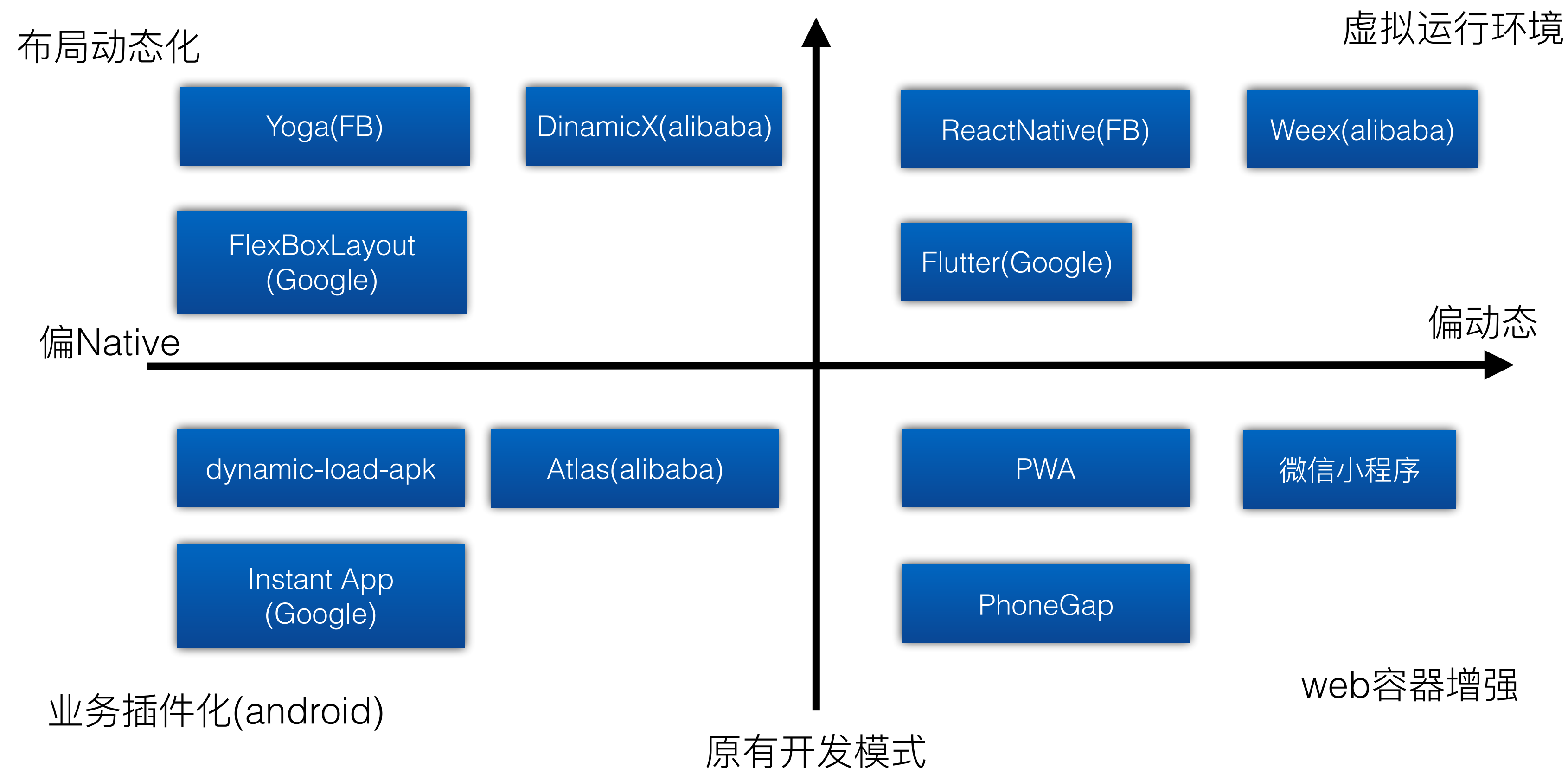
发布周期长

需求响应不灵活



火车模型，固定的“发车”时间，功能的发布取决于是否赶上最近的火车发车时间

# 客户端动态化能带来什么



## 动态化带来的好处

1. 提高迭代效率 — 相对native
  - 研发效率：避免多端开发
  - 更新效率：避免版本依赖
2. 提升体验 — 相对H5
  - 类native体验
  - 功能强大

客户端动态化技术演进

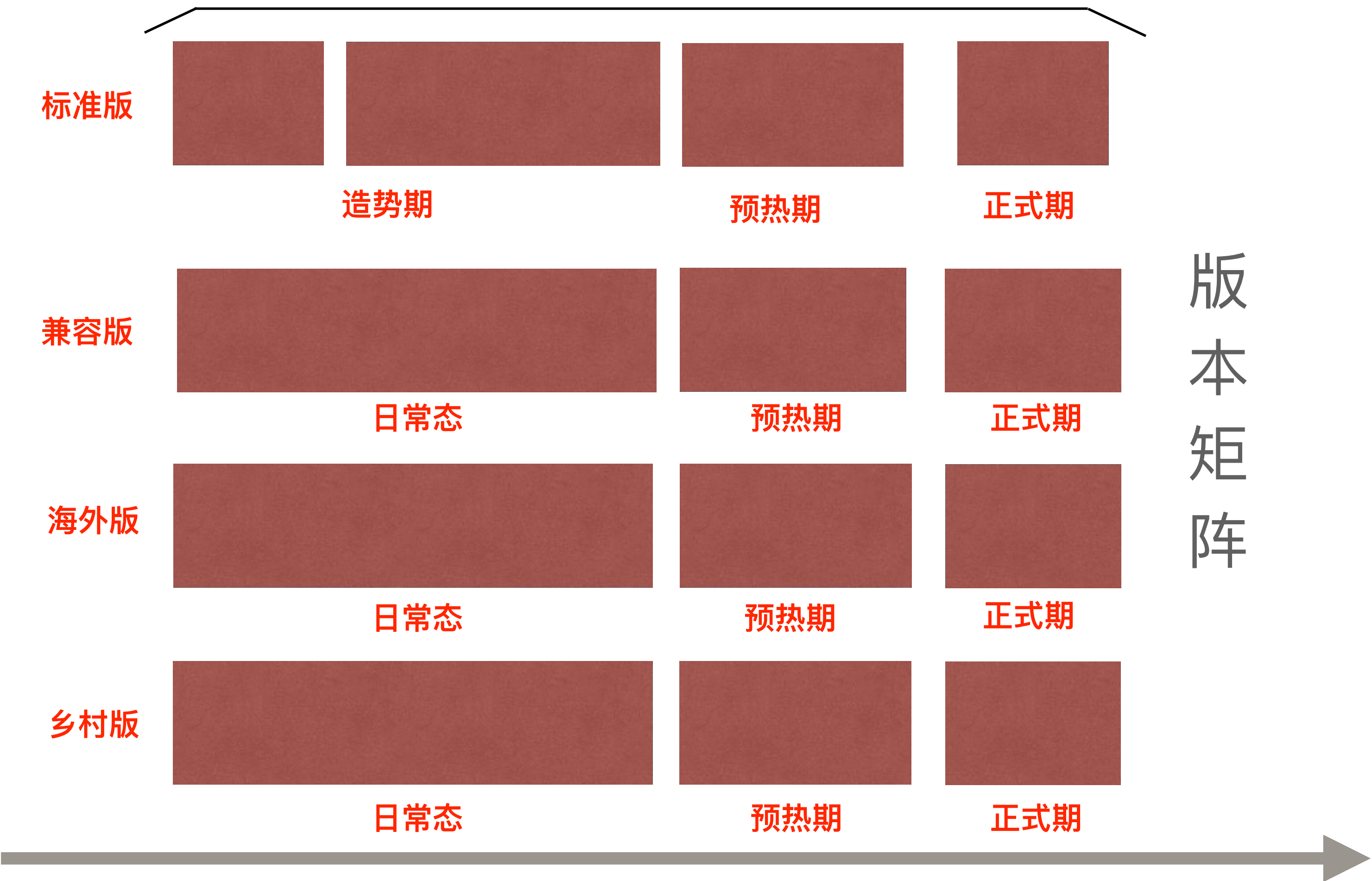
# 动态化技术方案对比

方案	迁移效率	开发效率	更新效率	稳定性	流畅度	适合场景
web容器增强	高	高	高	中	低	跨终端跨APP访问
业务插件化	高	中	中	中	中	模块隔离，动态更新
布局动态化	低	中	高	高	高	极致稳定性，如首页，交易链路
虚拟运行环境	中	高	高	中	中	尝试型新业务，短期运营业务



# 基于布局动态化的多版本首页搭建

大促不同时期的首页



## 多版本首页搭建的挑战

1. 版本页面多
  - 不同人群、地域投不同版本
  - 版本关联AB实验
  - 低版本客户端兼容
  - 低性能机器兼容
2. 同个页面，大促态与日常态UI差异大，需支持页面排期发布
3. 组件投放逻辑复杂
  - 不同人群、不同位置投不同组件
  - 组件关联AB实验
  - 组件支持资源位排期

# 定投体系

投放决策(基于规则引擎的逻辑判断)

AB实验

地理位置

人群

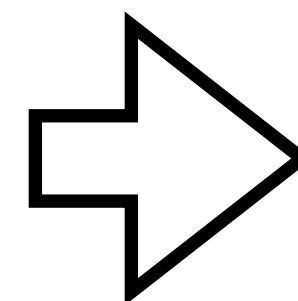
客户端版本

机型/操作系统/设备性能

.....

定投解决几个问题:

- 1.分人群产品策略;
- 2.客户端版本碎片化;
- 3.基于地理位置, 最大化流量利用;



引入的问题:

- 1.流量碎片化;
- 2.分流逻辑负责, 管理成本高;

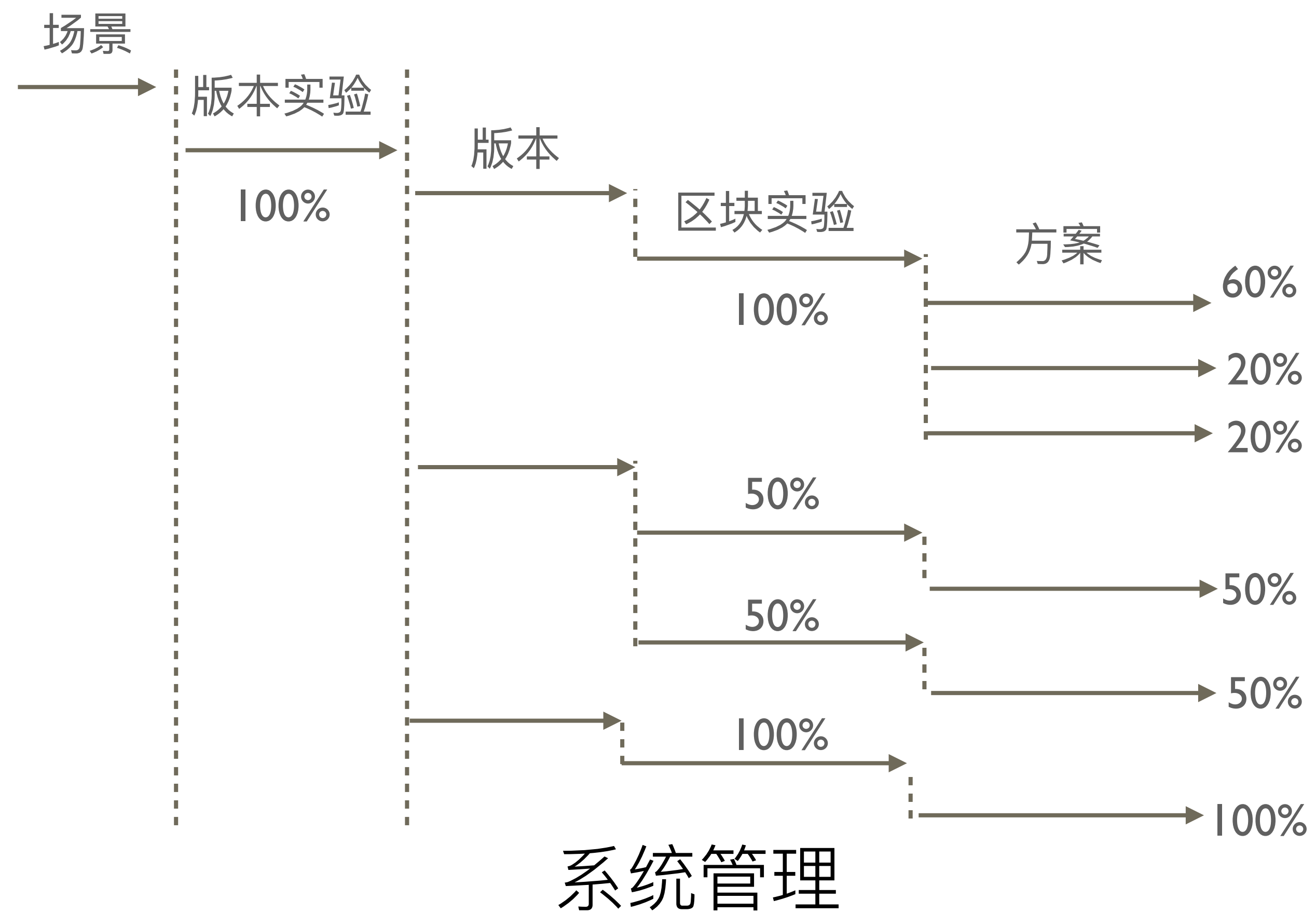
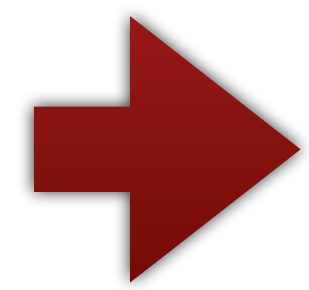
# 流量可视化

首页的分流逻辑复杂

- 1.分流依赖人肉管理，逻辑黑盒，易出问题
- 2.流量不可见，难及时发现问题



人脑管理



# 目录

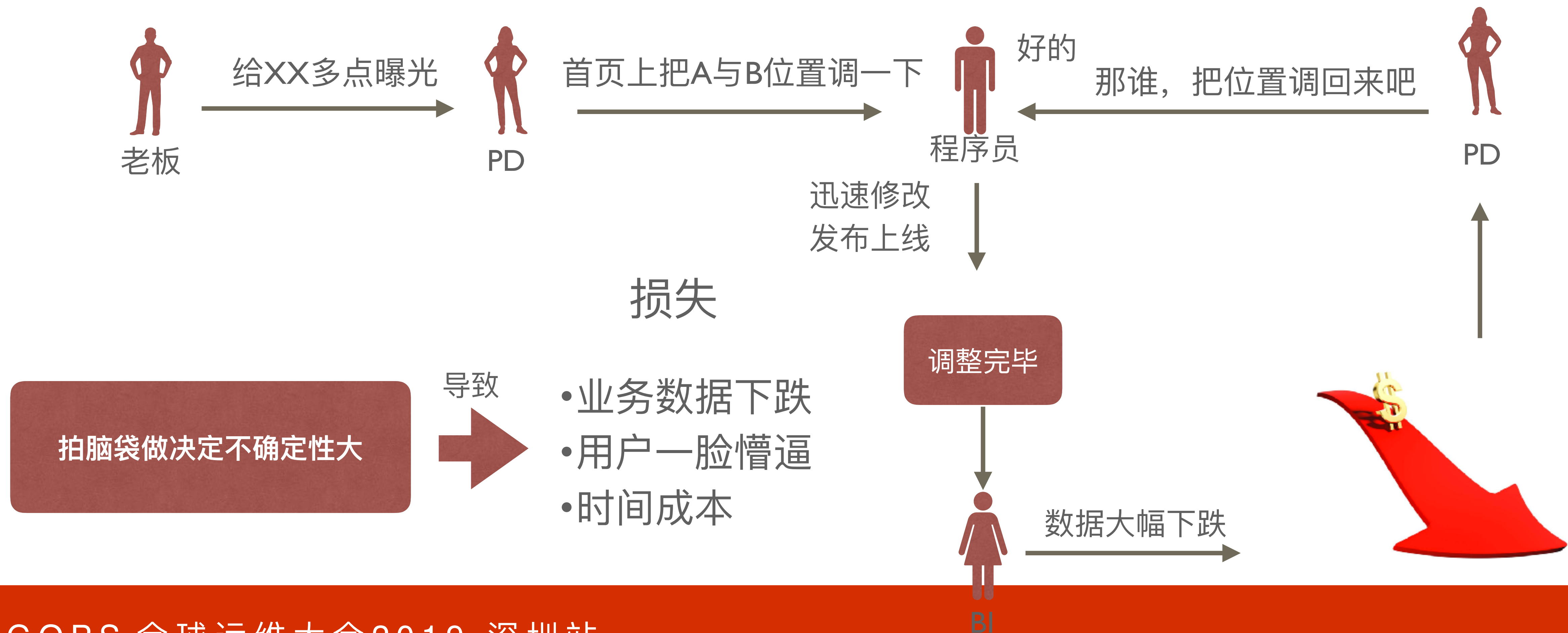
**1** 如何快速的迭代

 **2** 如何科学的迭代

**3** 如何保障首页的高可用

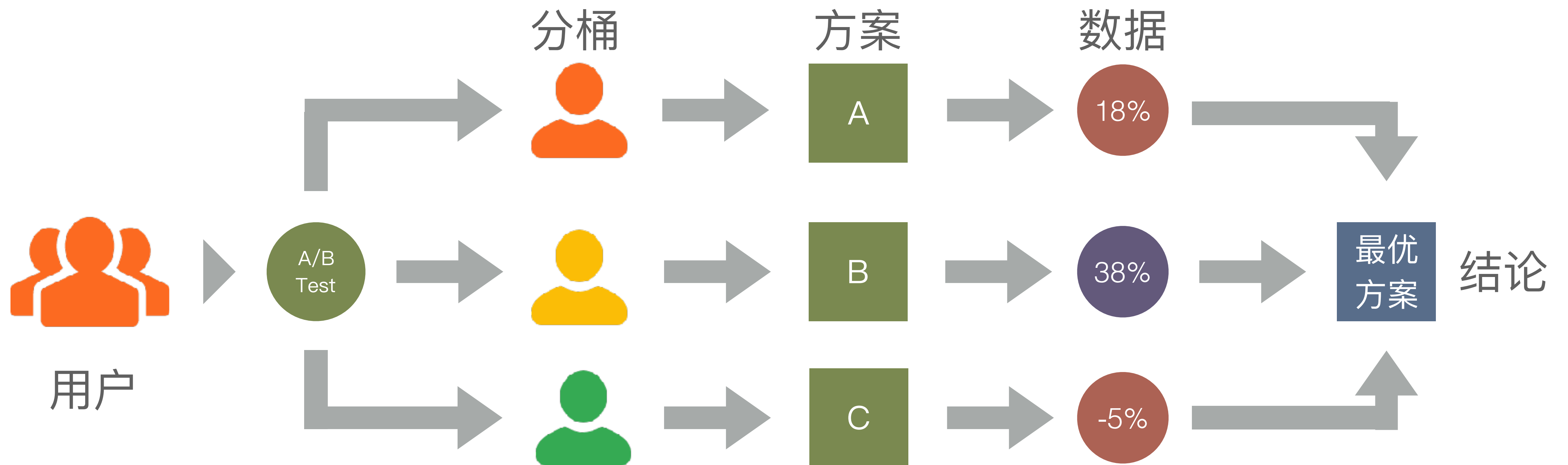
**4** Q/A

# 为何要通过实验驱动产品迭代



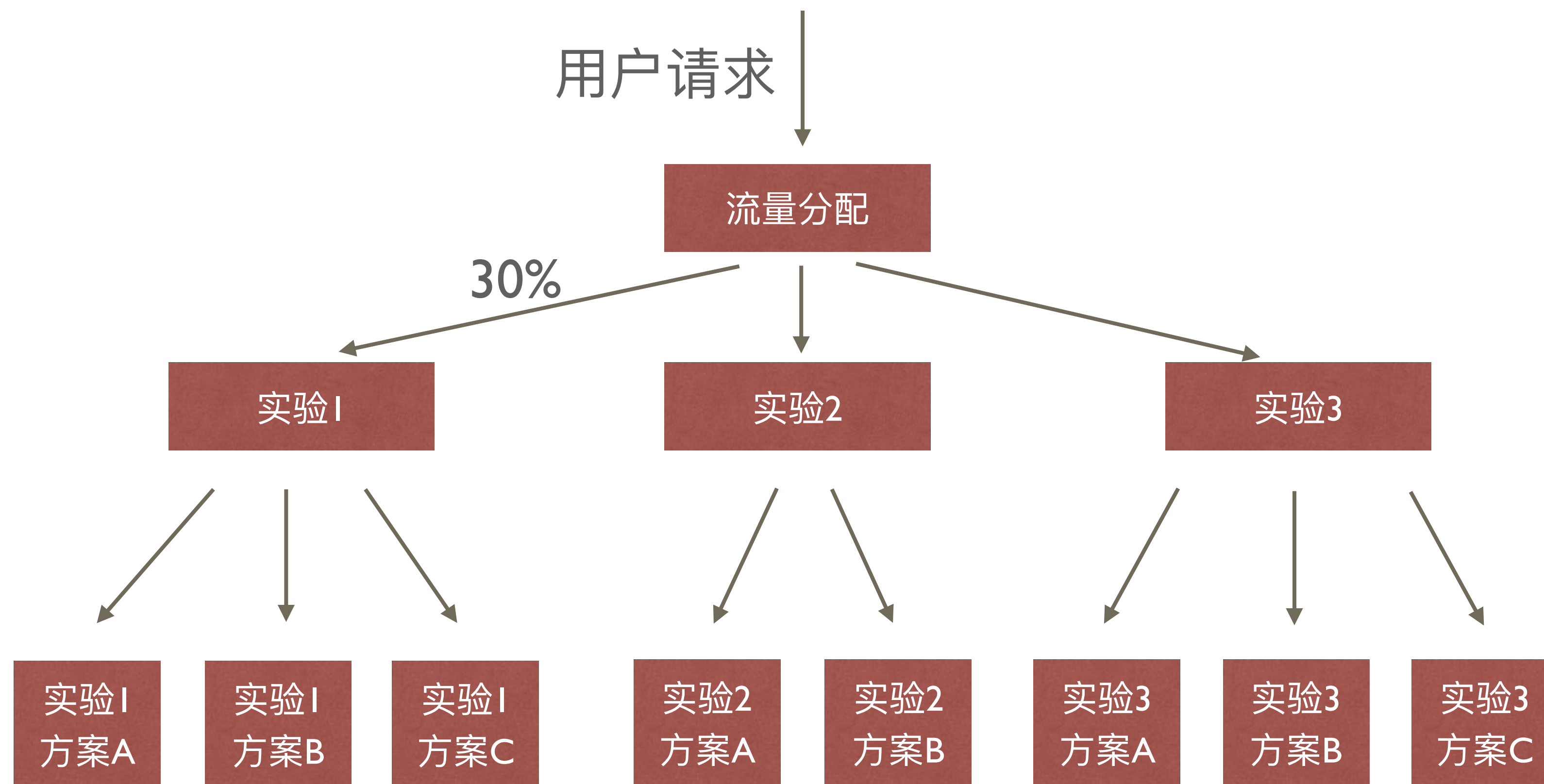


# 正确的迭代方式



- 在产品正式迭代发版之前，为同一个目标制定两个（或以上）方案，将用户流量对应分成几组，在保证每组用户特征相同的前提下，让不同分组用户看到不同的方案设计，根据几组用户的真实反馈，科学的帮助产品进行决策

# 流量不够怎么办



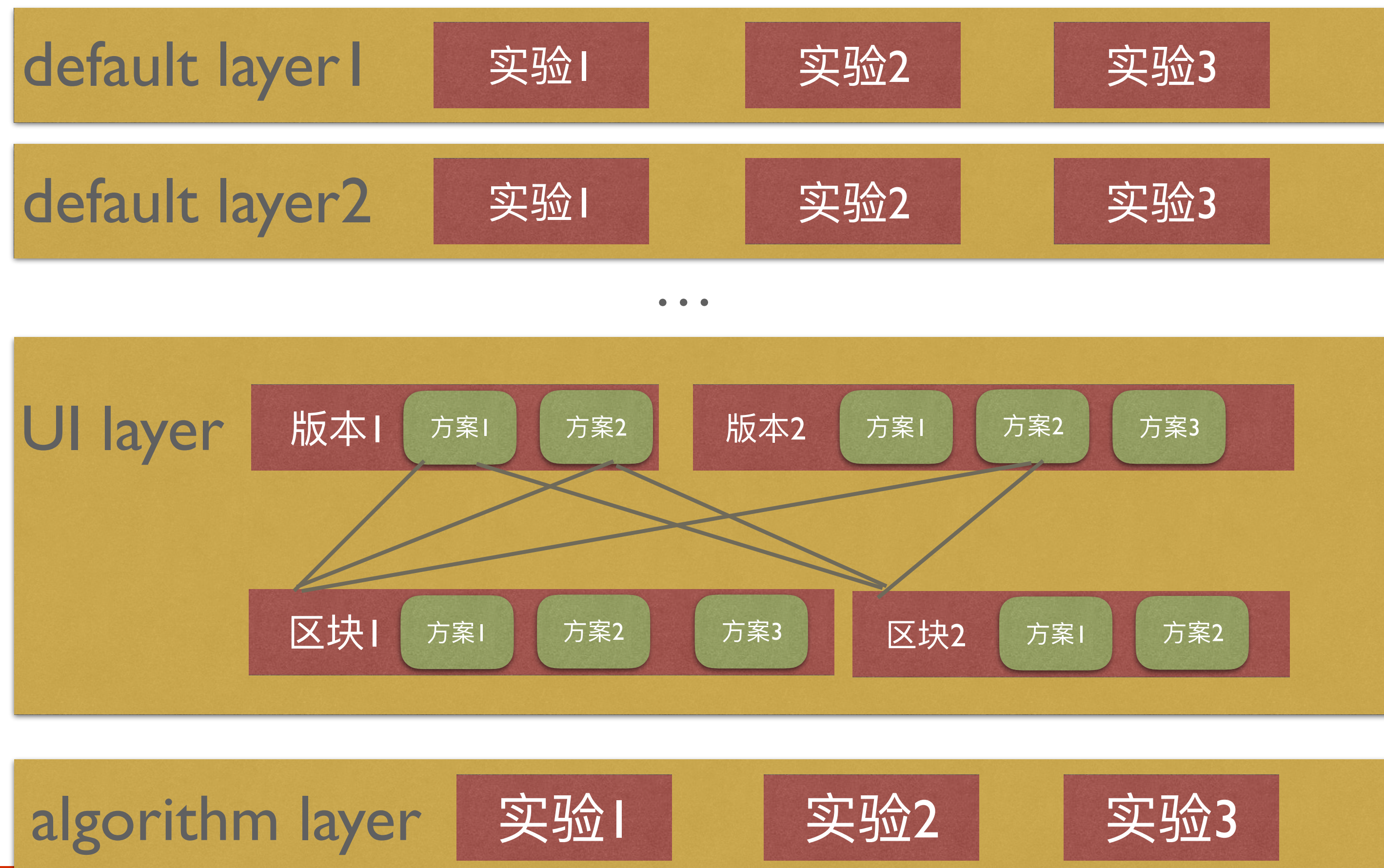
如果有上百个实验怎么办？

同时上？每个实验分到的流量少，实验结果不可信

OR

排队？无法保持高效

# 分层实验框架解决流量饥渴



分层实验，不同层间的实验互相独立，每层命中一个实验，流量到每一层重新打散。

每层几乎独占100%的流量，每次流量可以同时命中不同层间的实验选项。

层与层之间，必须正交

# AB实验中的注意事项

## 一. 样本容量

样本容量越大、置信区间越窄，结果越可信。流量（样本）太小，会导致结果不可信。需要合理最大化的利用流量，为实验各个指标的置信区间计算最小样本容量。

## 二. 条件偏置

经过条件过滤的流量不应该再分配给其他实验，因为这部分流量是被偏置的流量。

## 三. 参数耦合

比如实验A和实验B的流量重叠，如果A的参数会影响到B，那么就会导致B的结果有偏。

## 四. 调整流量

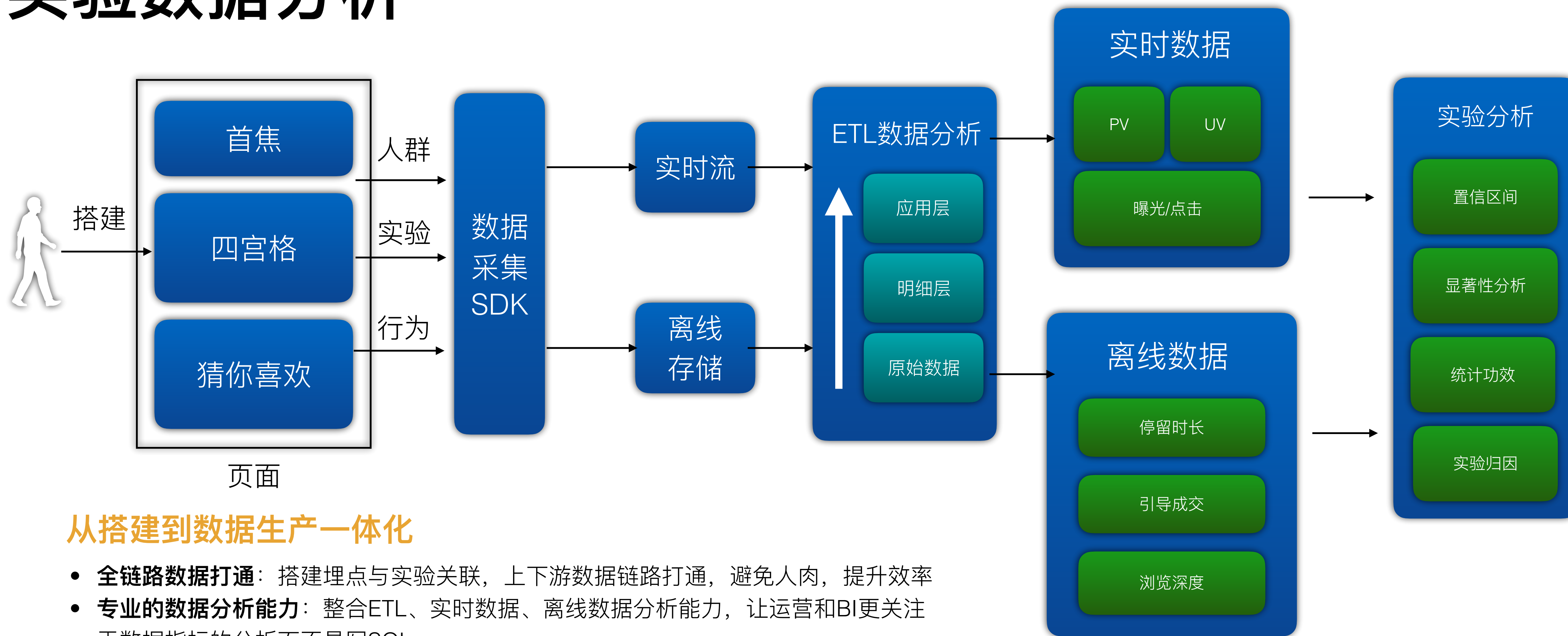
合理使用流量进行实验，尽量不要在实验过程中调整流量。

## 五. 人群跃迁

权益系统跟实验系统的冲突可能导致人群跃迁，计算时需要人工剔除。



# 实验数据分析

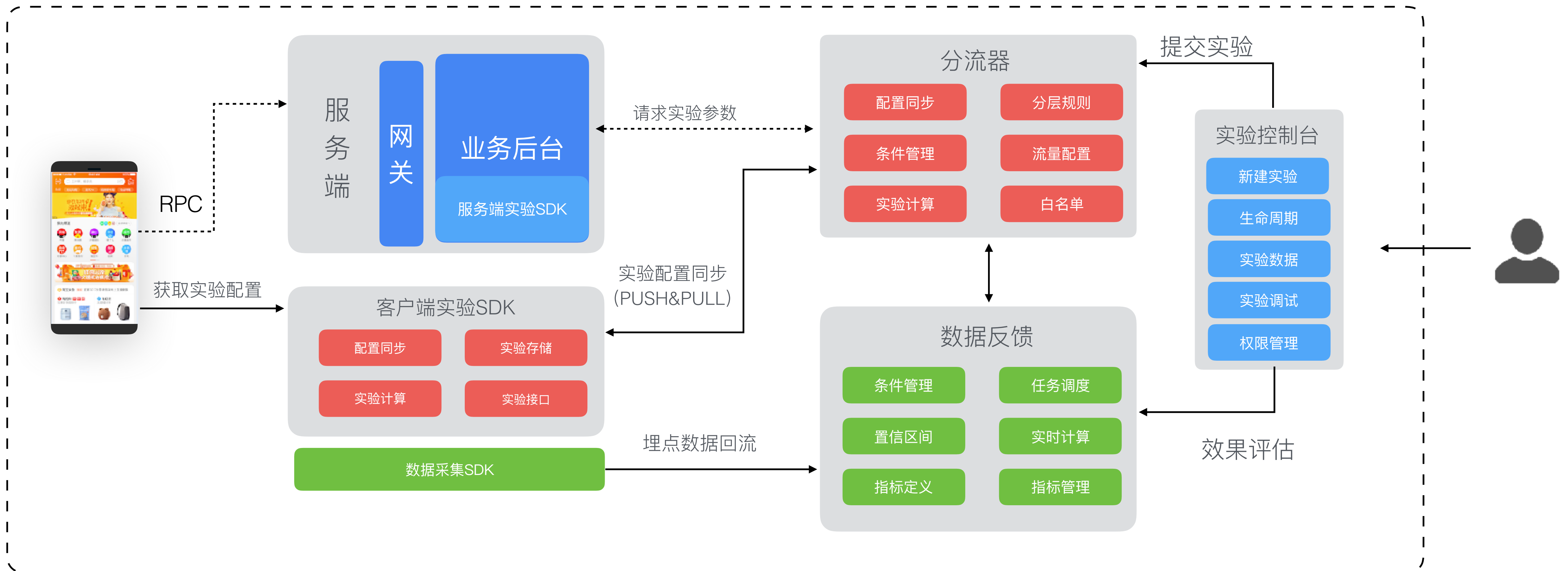


## 从搭建到数据生产一体化

- **全链路数据打通**：搭建埋点与实验关联，上下游数据链路打通，避免人肉，提升效率
- **专业的数据分析能力**：整合ETL、实时数据、离线数据分析能力，让运营和BI更关注于数据指标的分析而不是写SQL
- **实验数据沉淀**：帮助更好的确定实验最小样本数以及进行实验结果归因
- **及时的数据反馈**：常规实验从实验上线就开始产生数据，提升决策效能



# AB实验技术架构



# 目录

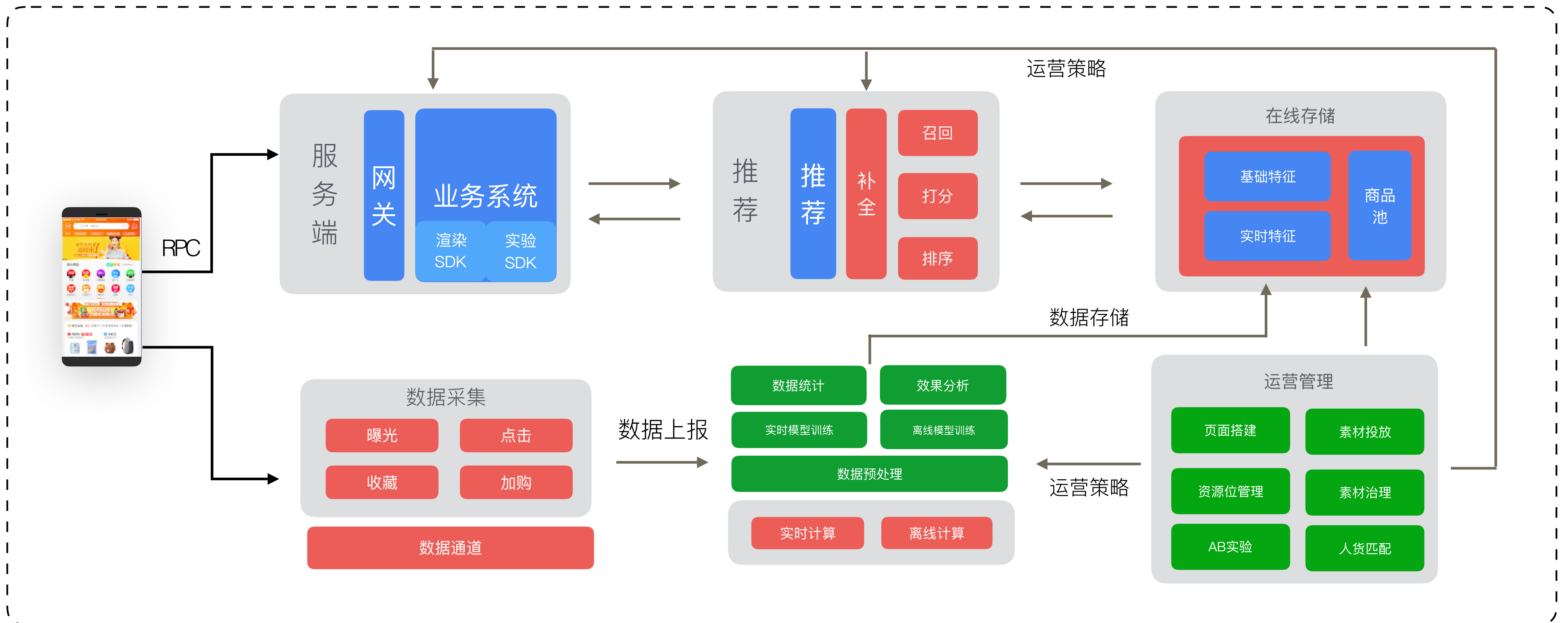
**1** 如何快速的迭代

**2** 如何科学的迭代

 **3** 如何保障首页的高可用

**4** Q/A

# 首页系统架构



# 首页场景下稳定性的挑战&应对

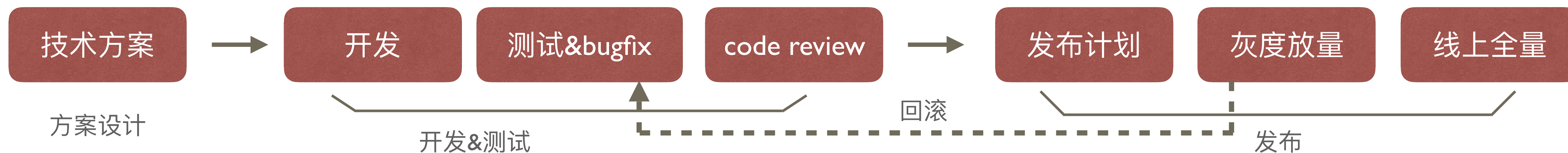
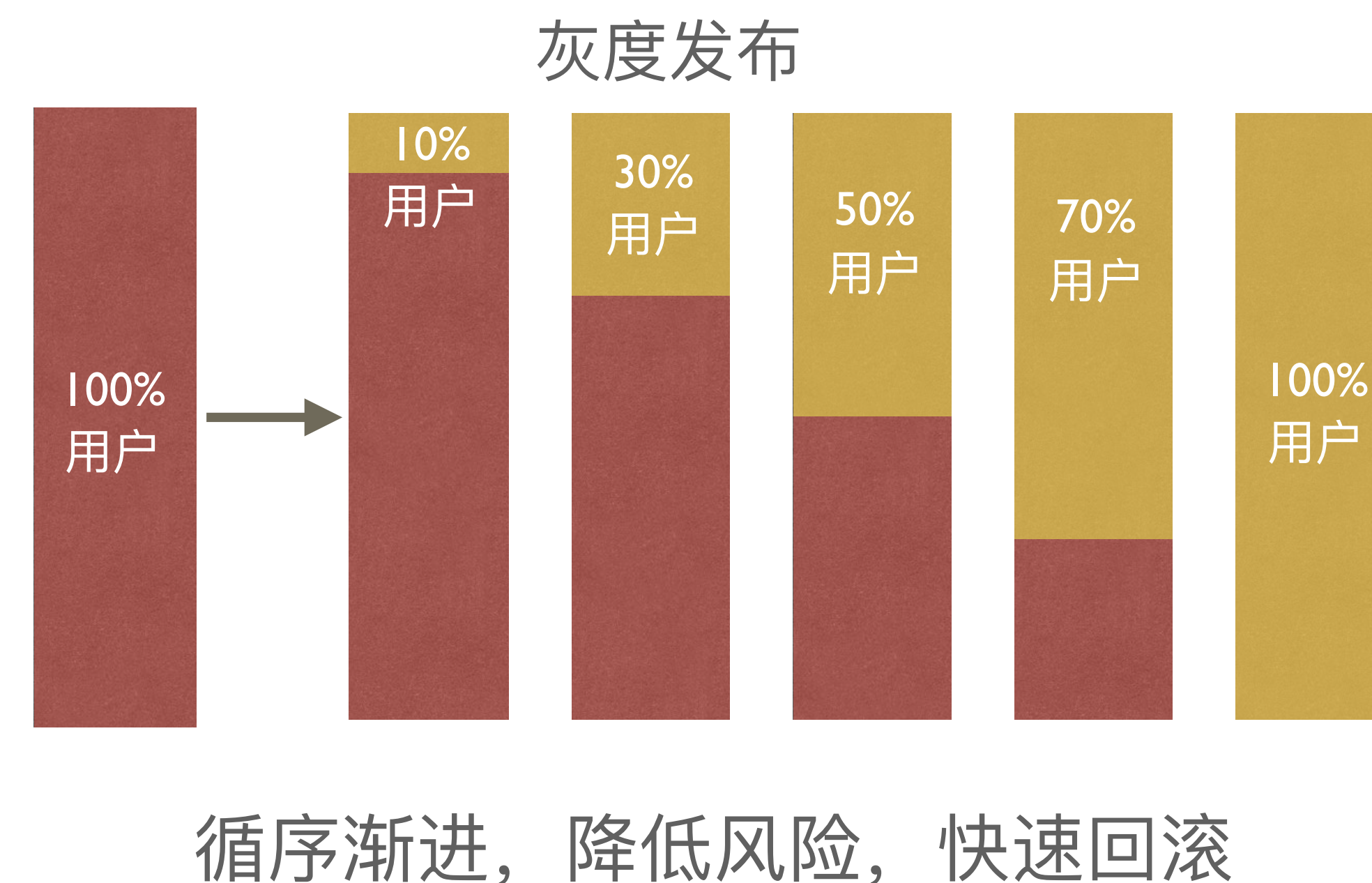
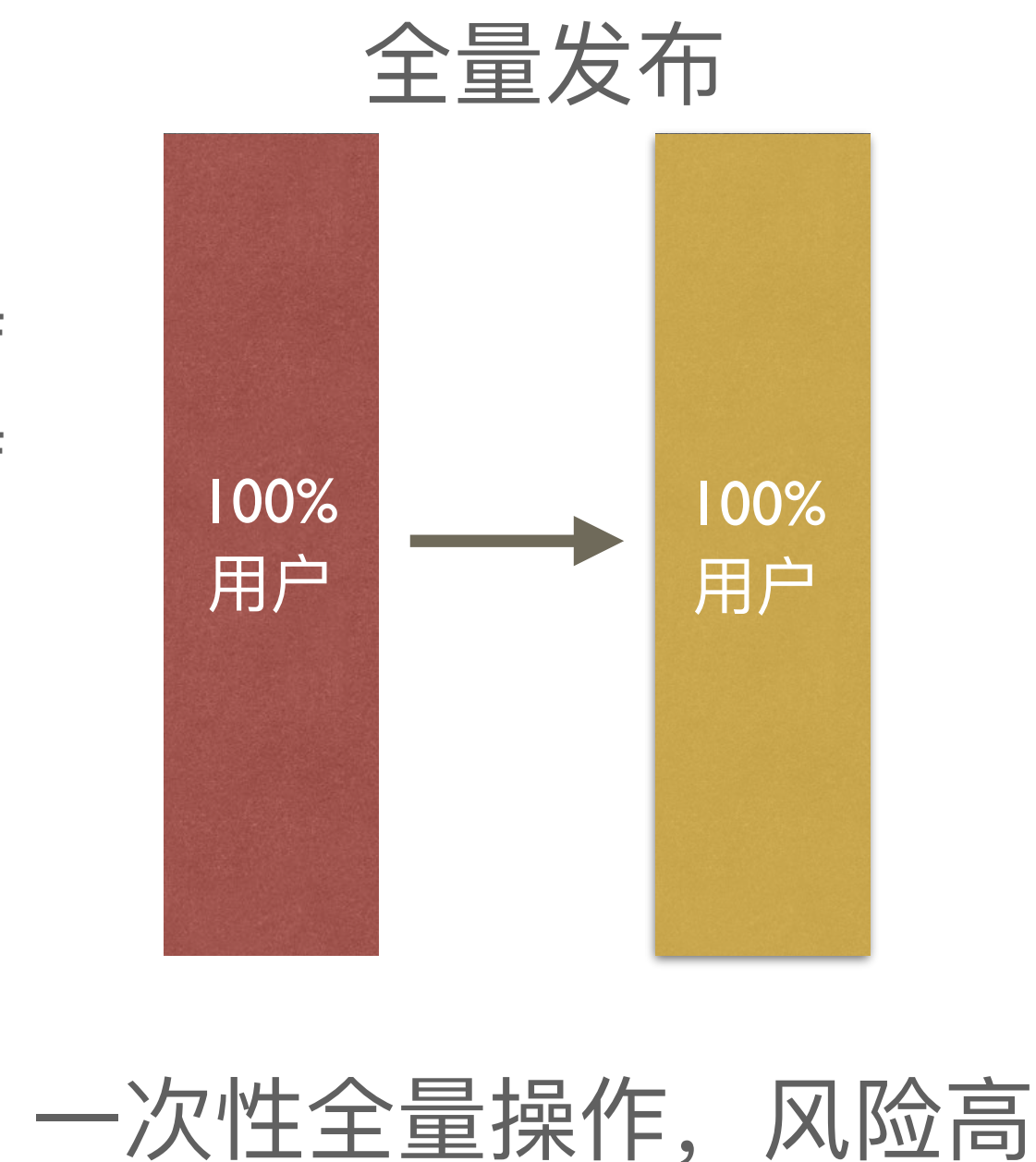
1. 流量巨大，业务迭代速度非常快，变更频繁
  - 业务要快速迭代，但变更不能无序
  - 机制和流程能减少犯错，如code review、发布审核、技术方案review等
  - 灰度发布能及时发现问题，避免问题扩大
2. 上下游依赖复杂，系统的异常和失败是常态
  - 各种异常状态的容灾兜底
  - 发生异常情况下的熔断机制，如超时、限流等
3. 业务逻辑定投条件复杂，融合了版本、位置、机型等信息
  - 监控要细，并且要定期梳理，过滤无效的监控，提升效率
  - 自动化测试系统，提升效率



# 灰度发布控制风险

## 变更管控：

- 日常与紧急变更区别对待
- 代码与配置变更同等对待
- 坚持灰度放量
- 变更可回滚
- 变更可查询
- 加强code review

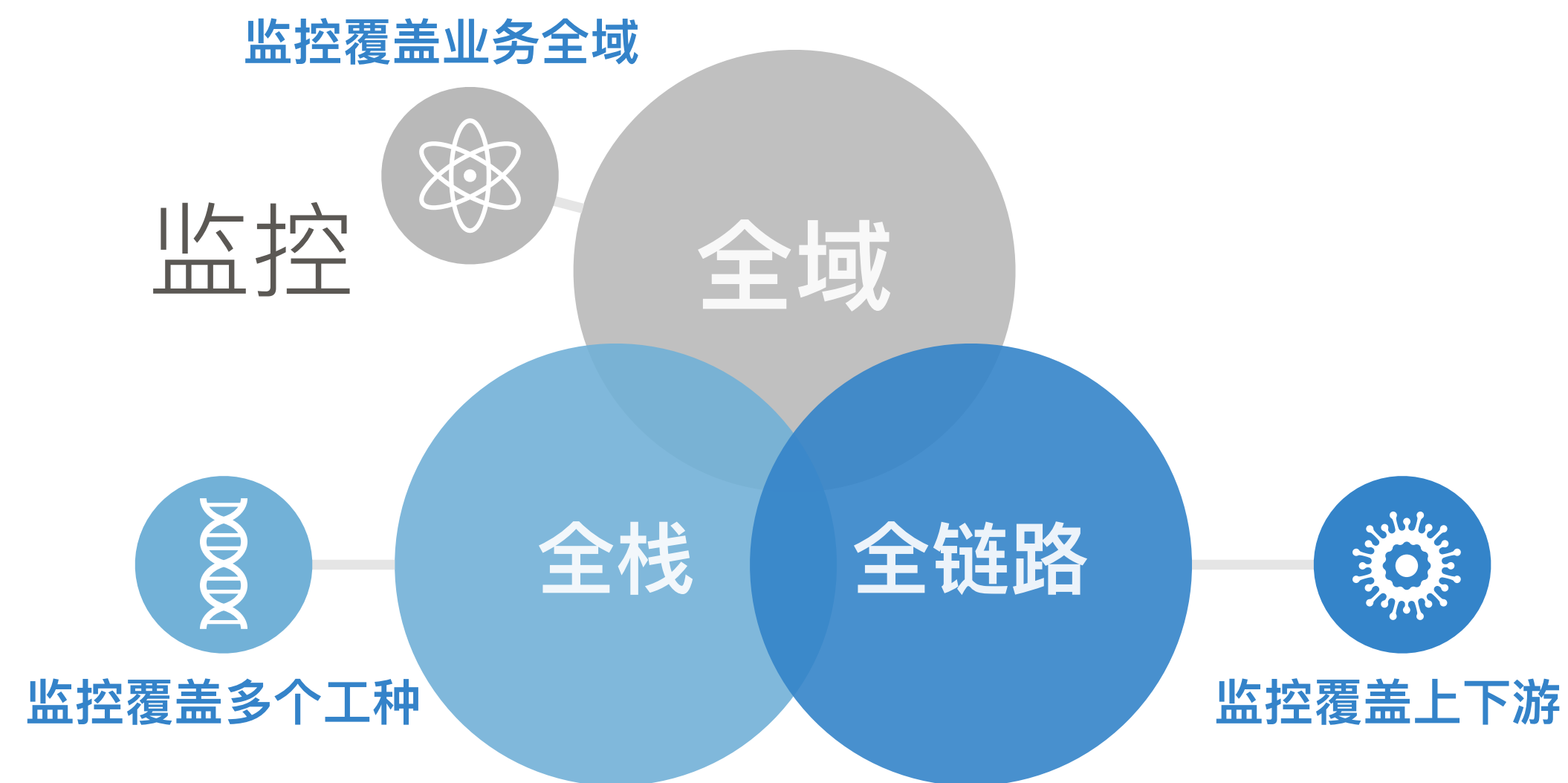




# 容灾策略



# 监控的有效性



诊断



**核心业务定义：**定义核心业务，定期review核心业务监控

**核心业务指标：**系统各个业务的访问流量

**基础数据指标监控：**异常量、线程数、CPU、Load、内存等等

**客户端指标监控：**crash率、请求失败率、响应时长、曝光点击等等

**上下游链路监控：**对下游依赖的响应时间、错误率做监控，以及降级预案

**变更：**全链路变更同步可查询

**异常与异常关联：**增加异常与异常之间的关联

**异常与变更关联：**异常出现的时机，与变更关联起来，帮助定位问题

**异常反馈&标记：**异常处理完后，填写反馈及原因，便于下次更快速的处理

**历史异常记录：**通过历史发生次数及原因定位，自动判别是否需要人工介入

**同时刻其他异常：**便于找到异常之间的关联关系

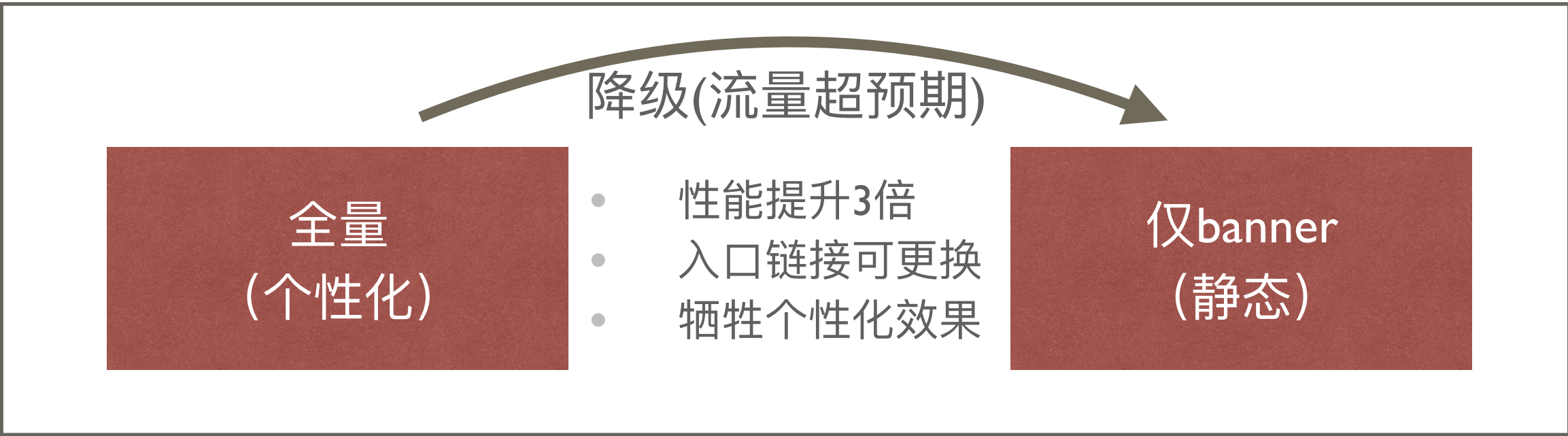
**异常诊断：**基于历史数据，算法自动识别是否需要人工介入，过滤无效信息

# 一次大促保障实践 — 春节大流量方案

挑战：

- 流量远超双十一，模型评估无参考
- 机器资源有限，保障互动入口在各种极端情况下都能透出

主  
方案



降级  
方案



互动入口  
强保障  
(百万级QPS)





Q/A



# Thanks

高效运维社区  
开放运维联盟

荣誉出品



想第一时间看到高效运维社区  
的新动态吗?

