



# 探探A/B Test实践

桂勇哲

---

探探  
资深技术专家



# 目录

什么是A/B Test

01

如何实现一个A/B Test平台

02

遇到的问题

03

第一部分

# 什么是A/B Test



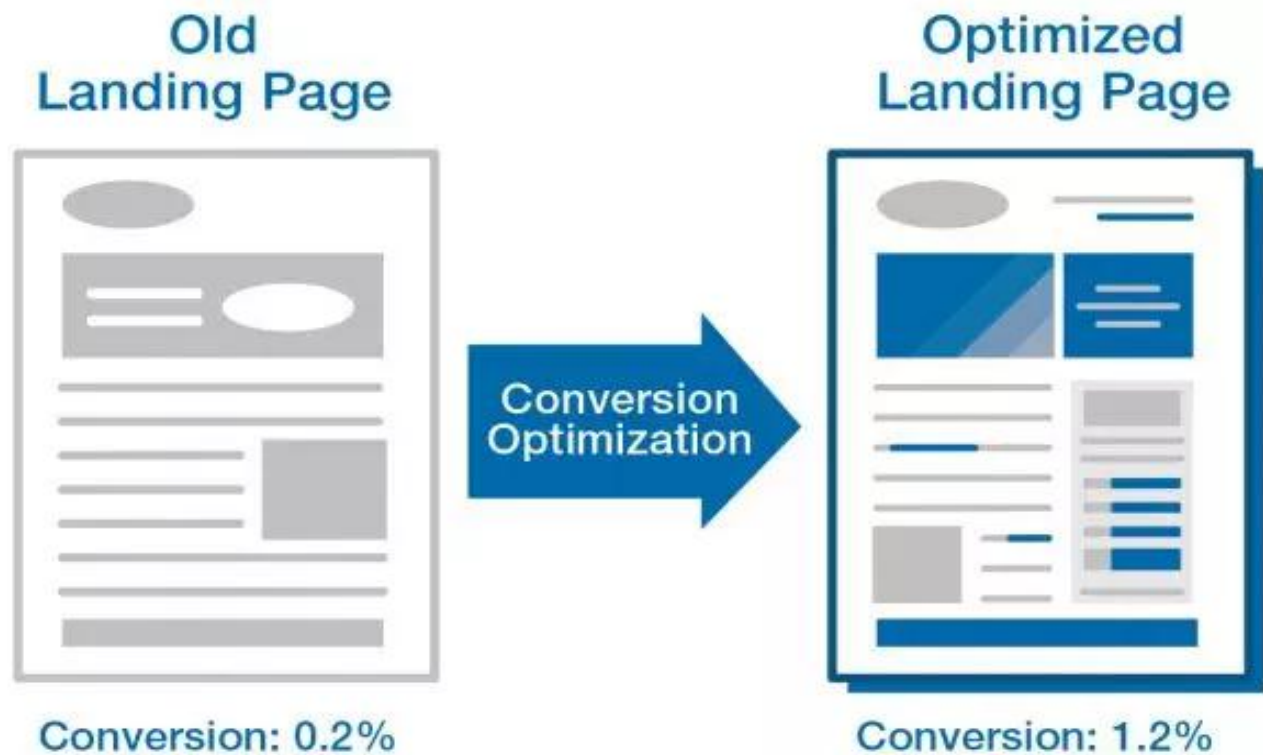
- 2018年 探探App整体改版，是否真的用户体验提升
- 2018年，VIP会员的价格，多少是最合适的
- 2019年，新增闪聊功能以后，用户聊天会变多么
- 2020年，注册流程的修改，是否会导致新用户增加
- 2020年，图片变的更清晰以后，被喜欢的概率会提升么

# 什么是A/B Test

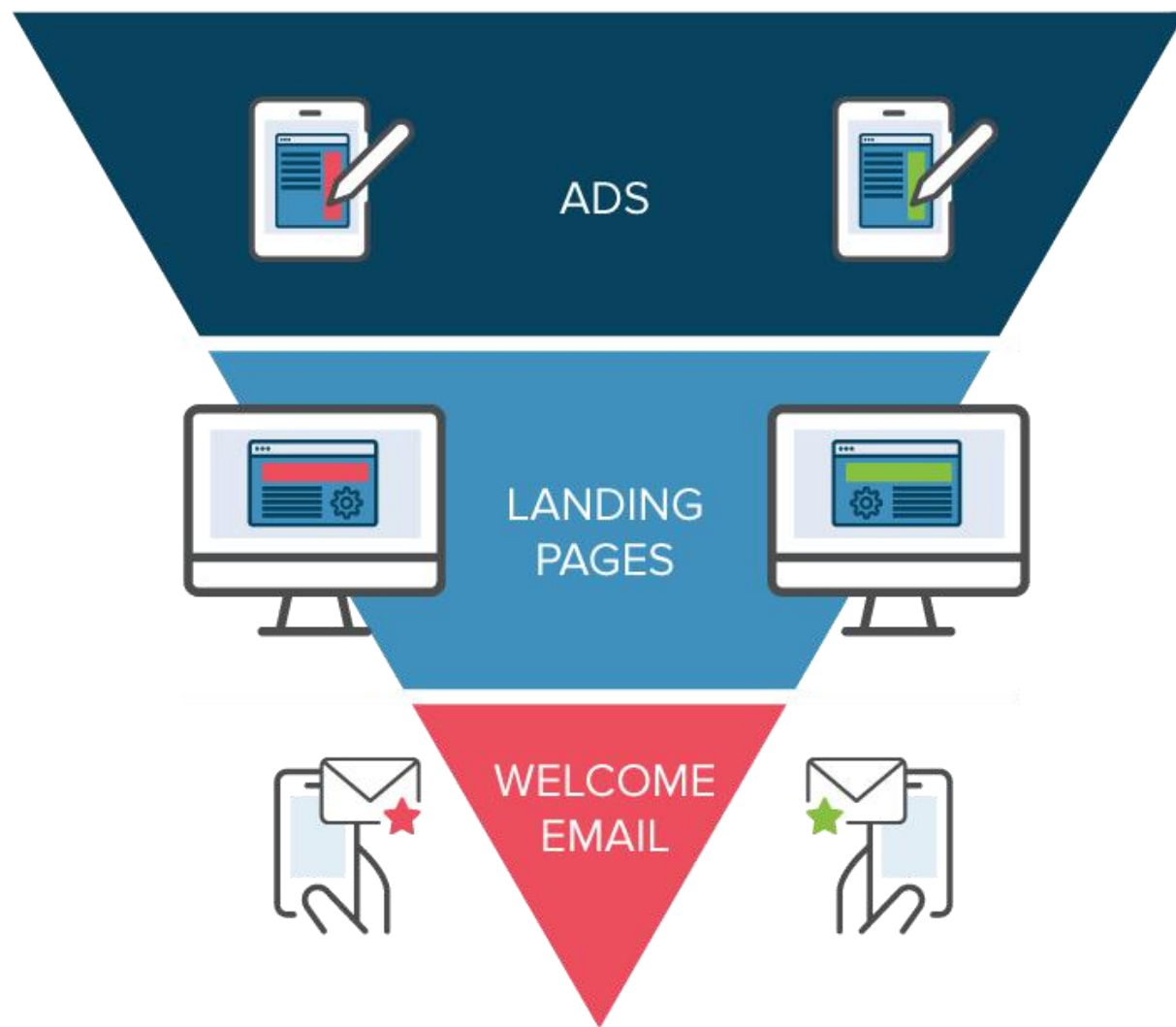
---

- A/B测试是一种随机测试，来对两个变量进行比较
- 基于对照实验与假设检验
- 将用户随机对分不到A、B两个组，比较两个组内用户对表现是否不同
- 基于数据，决策那个方案更好

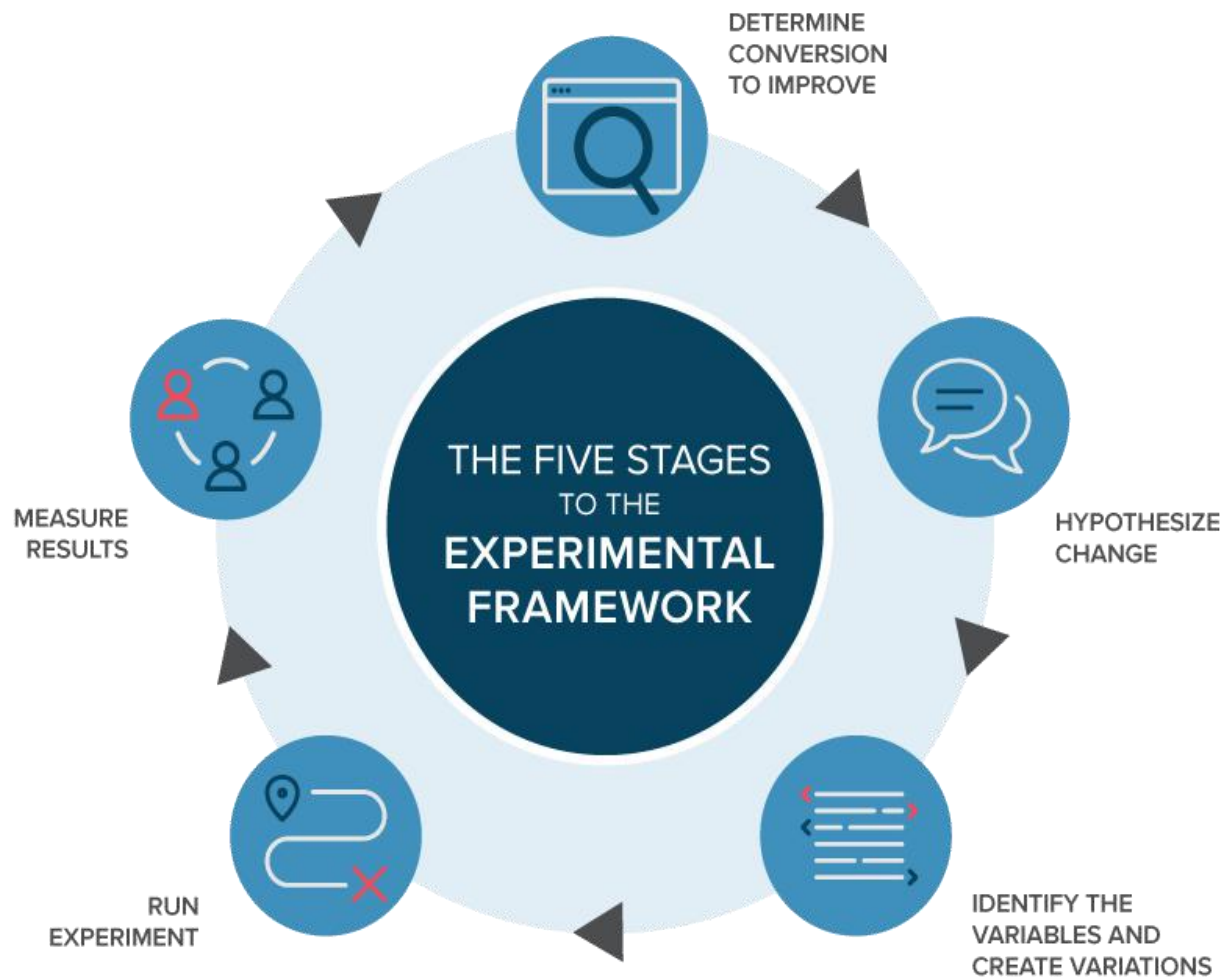
# 什么是A/B Test



# 什么是A/B Test



# 什么是A/B Test





第二部分

# 如何实现A/B Test平台

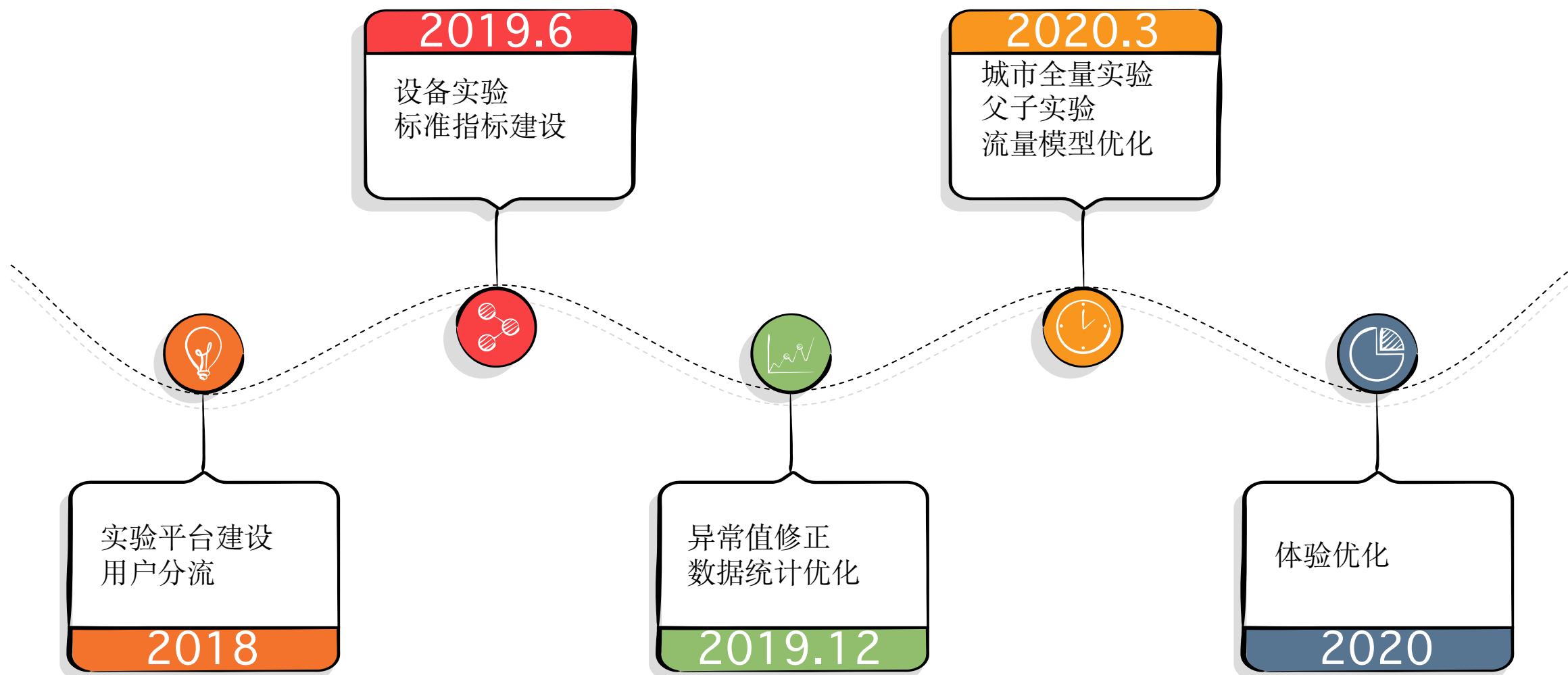


# 如何实现A/B Test平台

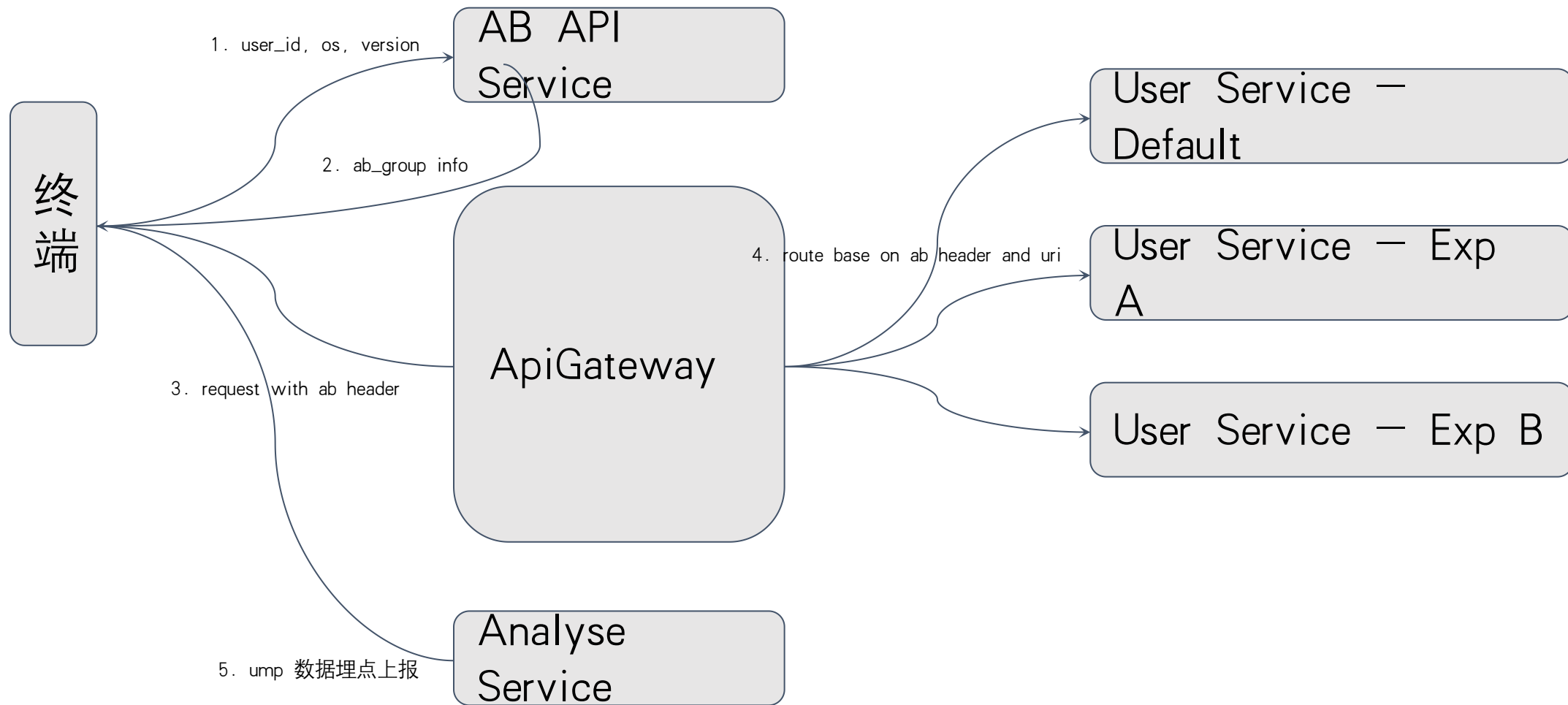
---

- 实验配置与管理
- 流量分发
- 数据分析

# 如何实现A/B Test平台



# 如何实现A/B Test平台



# Why Golang

---

- ApiGateway
  - 作为网关层，需要确保高性能，低延迟
  - Golang 1.5后gc持续改进
- ApiService
  - 需要从多个服务获取用户属性，支持异步和并发
    - goroutine的实现能够很方便的实现
  - 有框架支持实现RESTful API
- Web Admin
  - 支持网站开发，有良好的template支持
  - 支持task，crontab之类的管理
- 工程规范，生态，开发效率



# ApiGateway

```
package v2

import ...

var (
    allowedMethods      = []string{"GET", "PUT", "POST", "DELETE"}
    corsMethodsDelimiter = ","
    corsHeadersDelimiter = ","
)

// GetRouteConfigurations extracts xDS RouteConfigurations from DiscoveryResponse.
func GetRouteConfigurations(resp *xdsapi.DiscoveryResponse) ([]*xdsapi.RouteConfiguration, error) {
    routes := make([]*xdsapi.RouteConfiguration, 0, len(resp.Resources))
    for _, res := range resp.Resources {
        var resource ptypes.DynamicAny
        if err := ptypes.UnmarshalAny(res, &resource); err != nil : nil, err ↗
        route, ok := resource.Message.(*xdsapi.RouteConfiguration)
        if !ok : nil, fmt.Errorf("unexpected resource type: %T in RDS response", resource.Message) ↗
        routes = append(routes, route)
    }
    return routes, nil
}
```

# expr evaluator

```
// Package evaluator evaluates an expression in the form of s-expression
package evaluator

import "errors"

var (
    // ErrNotFound means the unknown string within the expression cannot be Get from neither functions or params
    ErrNotFound = errors.New( text: "neither function not variable found")
    // ErrInvalidResult means the invalid result type expected with the real output
    ErrInvalidResult = errors.New( text: "invalid result type")
)

// Expression stands for an expression which can be evaluated by passing required params
type Expression struct {
    exp sexp
}

// New will return a Expression by parsing the given expression string
func New(expr string) (Expression, error) {
    exp, err := parse(expr)
    if err != nil : Expression{}, err ↗
    return Expression{
        exp: exp,
    }, nil
}

// Eval evaluates the Expression with params and return the real value in the type of interface
func (e Expression) Eval(params Params) (interface{}, error) {
    return e.exp.evaluate(params)
}
```

# web admin

```
<!-- Chosen script -->
{{append . "moreScripts" "premium/js/plugins/chosen/chosen.jquery.js"}}
{{append . "moreStyles" "premium/css/plugins/datapicker/datepicker3.css"}}
{{append . "moreScripts" "premium/js/plugins/datapicker/bootstrap-datepicker.js"}}

{{template "header.html" .}}
<!-- breadcrumb -->
<div class="row wrapper border-bottom white-bg page-heading">
  <div class="col-sm-4">
    <h2> </h2>
    <ol class="breadcrumb">
      <li>
        <a href="{{url "Experiment.Mine"}}">Dashboard</a>
      </li>
      <li>
        <a href="{{url "Experiment.Detail" .exp.ID}}">{{.exp.Name}}</a>
      </li>
      <li class="active">
        <strong>Project detail</strong>
      </li>
    </ol>
  </div>
</div>
```



# 如何实现A/B Test平台

- Golang + revel
- JQuery
- Bootstrap

测试实验demo

Running

基本配置

取样配置

统计分析beta

流量监控beta

实验报告beta

层: 【自动创建】 2.0实验层, 1274-645(user-mod-199)

桶:

终端版本: Android no version limit

IOS no version limit

系统版本: Android no version limit

IOS no version limit

渠道:

不包含: googleplay

语言:

包含: 简体中文

设备:

性别: 全部用户

管理者: guiyongzhe

抄送人员: guiyongzhe

开始时间(优先级): 2020-10-30 13:02:55

运行时间: 2020-10-30 13:02:55

实验可看数时间: -

入组率达标

预计到期时间: -

关闭时间: -

实验ID: 4921

实验改动类型: 无改动实验

是否入组后立即生效: 否

实验阶段: 小流量阶段

配置要求:

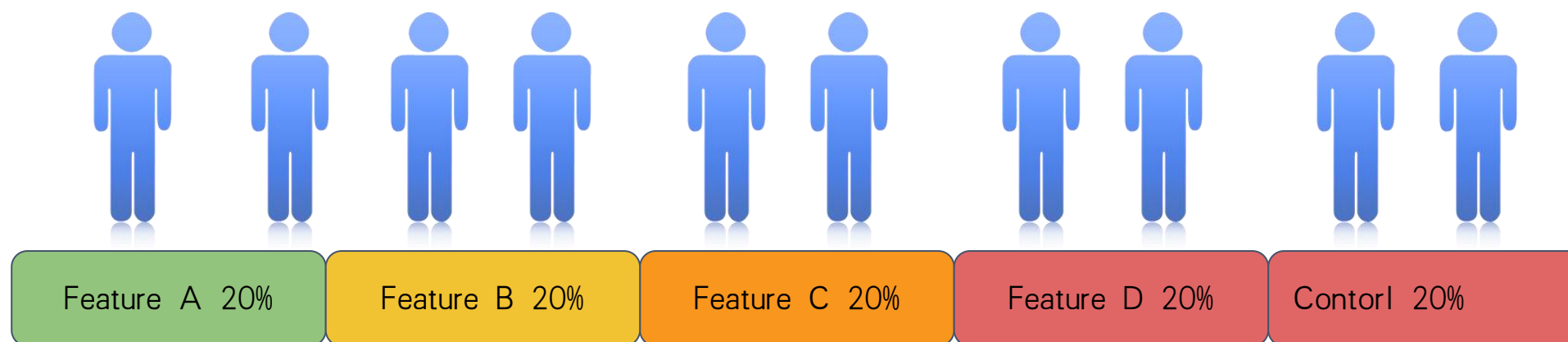
```
(and
  (or (and (eq os "android") (ne affiliate "googleplay")) (ne os "android"))
  (eq language "zh-Hans")
)
```

#	Key	服务端参数	客户端参数
9068	exp_test_demo:test_a		map[]
9069	exp_test_demo:__control		map[]



# 流量分发

# 如何实现A/B Test平台



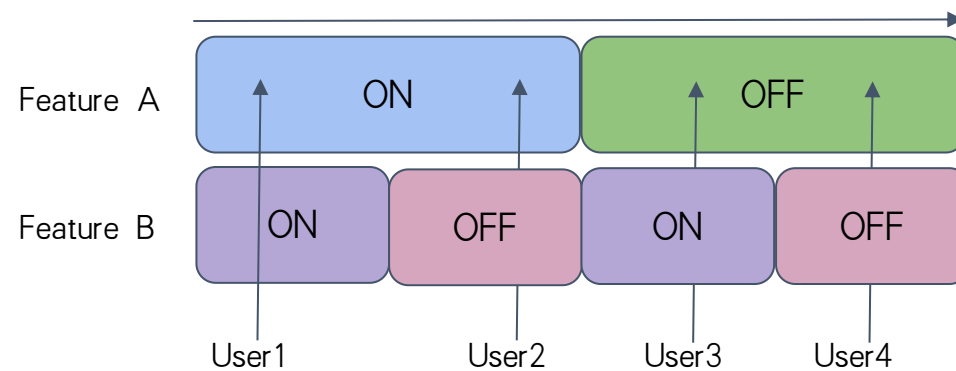
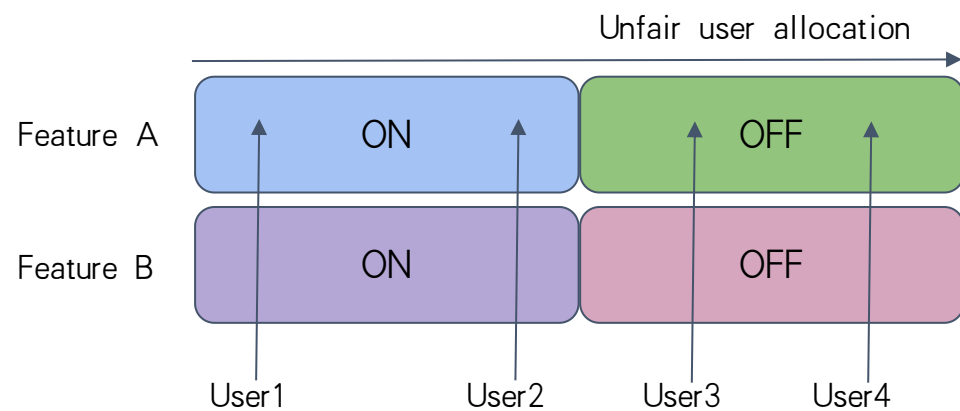
# 重叠实验

## Overlapping Experiment Infrastructure: More, Better, Faster Experimentation

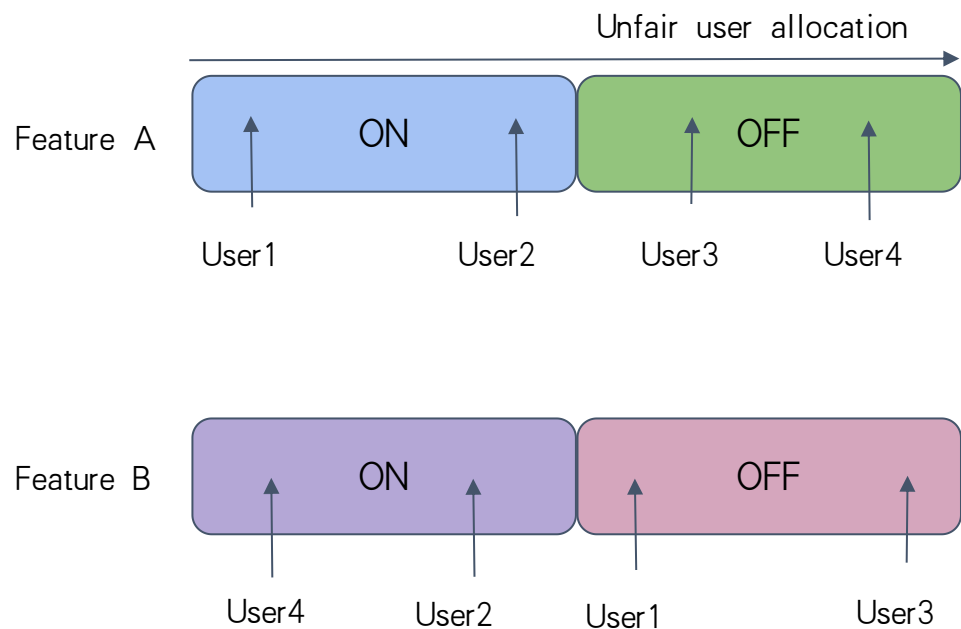
- **更多**: 我们需要系统具备一定的可伸缩性, 以便于能够同时运行更多的实验。与此同时, 也需要一定的灵活性: 不同的实验需要不同的配置和不同的流量来衡量实验在统计意义上的效果显著性。有些实验只需要改变流量的一个子集, 比如日本的流量, 并且需要适当的调整流量大小; 其他实验则可能会改变所有的流量, 并在度量上产生较大的变化, 因此可以在较少的流量上运行。
- **更好**: 无效的实验不应该允许在线上流量上运行。有效但糟糕的实验(如, 线上Bug或无意中产生的非常糟糕的结果)应该能很快被发现并被禁止运行。标准化的实验指标应该对所有实验都是容易获取的, 以便实验的比较是公平的: 如两个实验在计算CTR(点击通过率)等指标时, 应该使用相同的过滤器来去除机器人产生的流量。
- **更快**: 建立一个实验应该既简单又快速, 简单到不需要专业的工程师, 也不需要编写任何代码就就可以做到。评价指标应该很快就能被统计出来, 以便快速评估实验效果。简单的迭代应该可以快速完成。此外, 实验平台除了支持实验进行ABTest, 还应该支持以易于理解的方式进行流量灰度。

# 分层实验模型

Feature A 用户表情优化  
Feature B 附近动态优化



# 分层实验模型



```
2 for (layer : layers) {  
3     val bucket = Math.abs(hash(userId + layer.layerId)) % 1024;  
4     val experiment = getExperiment(layer, bucket);  
5     // use experiment to determine user experience  
}
```

# 流量分发实现

---

```
(and
  (ne os "ios")
  (eq gender "male")
  (beteen version (t_version "2.7.1") (t_version "2.9.1"))
  (in (mod (hash_with_exp2 user_id 5072) 541) (3 5 6 36 38 39 46 60 75 81 87 95 107))
  (ge user_created_time (date_from_today "+0h"))
)
```

- 轻量的动态表达式引擎
- 简单可扩展
- 灵活、对人友好
- 流量分发核心引擎保持稳定

# 流量分发实现

function	example	description
in	(in 1 (1 2))	also suport array like (in (1) ((1)))
between	(between age 18 20)	
overlap	(overlap region (3142 1860))	
and	(and (eq gender "femal") (between age 18 20))	
or	(or (eq gender "male") (gt age 25))	
eq		equal
mod		
t_version		convert type to version
t_time	(t_time "2006-01-02 15:04" "2017-09-09 12:00")	convert type to time, first param must be the layout for the time





# 流量分发实现

---

## 国外实验

```
(or (eq user_source "facebook")  
    (and (eq user_source "mobile") (ne country_code 86)) )
```

## 过滤高级别vip用户

```
(lt user_vip_level 25)
```

## 限制用户注册时间

```
(ge user_created_time (date_from_today "+0h"))
```

# 流量分发实现

---

NewUI实验， 做了第一期后， 期望在一期的实验组上继续迭代。

(and

(in (mod (hash\_with\_exp2 user\_id 5072) 541) (3 5 6 36 38 39 46 60 75 81 107)

(in (mod (hash\_with\_exp2 user\_id 5072) 741) (3 5 6 36 38 39 46 60 75 95 107)))

第三部分

# 遇到的问题

# 一些问题

---

- 幂等性
- 如何实现扩量
- 层应该如何划分?
- 新功能改动过多应该如何处理?
- 跨层的互斥问题应该如何解决?

# 幂等性

---

- 限制版本，用户切换到低版本后是否出组
- 用户地理位置变化，是否会导致出组
- 用户属性，等级等变化，是否会导致用户出组

## 入组条件

```
(in (mod (hash_with_exp2 user_id 5072) 541) (3 5 6 36 38 39 46 60 75 81 107))
```

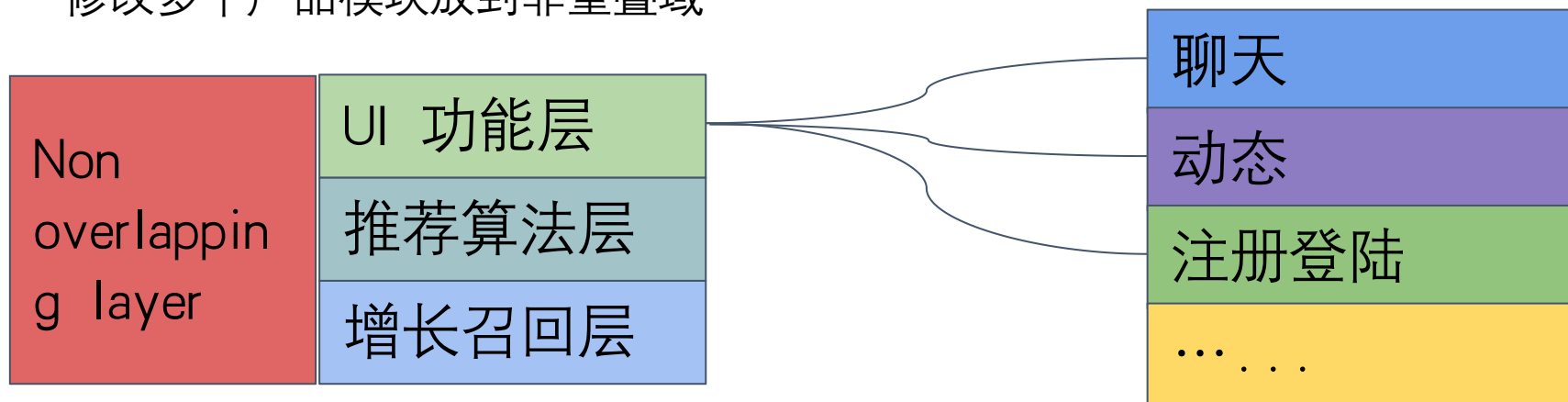
## 限制条件

```
(gt version (t_version "2.7.1"))
```

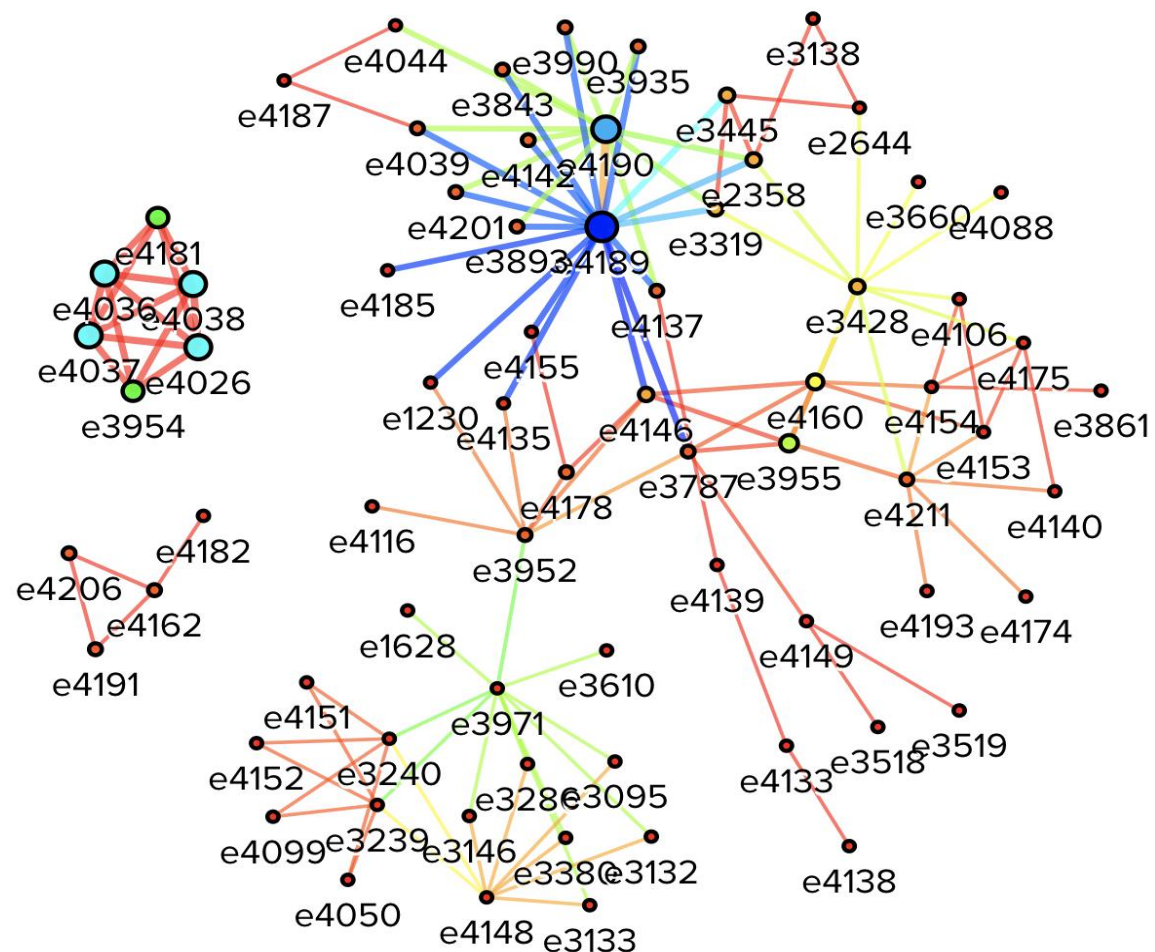
# 层如何划分

## 一个层是多个实验的集合

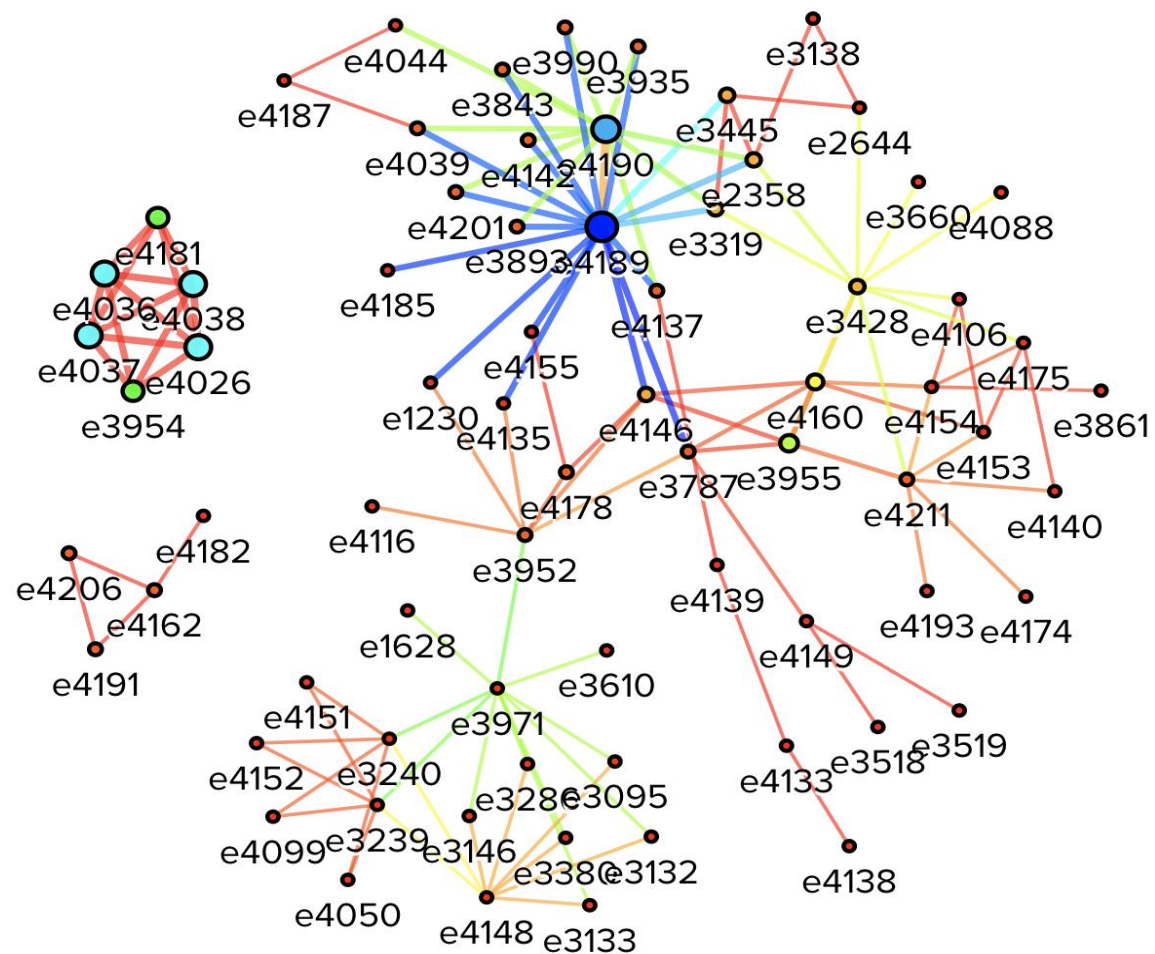
- 互相没有影响的实验可以分到不同的层
- 互相有影响的实验分到同一层
- 按照产品功能边界归属到对应的层
- 修改多个产品模块放到非重叠域



## 层如何划分



## 层如何划分





# 跨层互斥

---

- 当一个用户同时命中多个互斥实验的入组逻辑时
  - 如果已经在某个实验组中，保留在当前实验组，不能进入其他互斥实验
  - 两个实验按照开启时间进行优先级排序，优先进入先开的实验
- 更大程度的复用流量
  - A, B 实验互斥, B、C实验互斥时, A、C实验之间仍然是正交的
  - 可用流量增长了50%
- 实验的关闭，会导致其他互斥实验的入组流量增加
- 当存在某些大流量实验时问题，问题会变严重

# 数据分析

核心指标:

dau	
净增幅(实际):	P值: Null
0.72%	校正P值: Null
统计功效: Null	
净增幅(理论): 0.35%	

Revenue	
净增幅(实际):	P值: Null
-7.00%	校正P值: Null
统计功效: Null	
净增幅(理论): -6.05%	

duration	
净增幅(实际):	P值: 0.82
-0.39%	校正P值: Null
统计功效: 0.06	
净增幅(理论): 0.22%	

vip buy revenue	
净增幅(实际):	P值: Null
110.82%	校正P值: Null
统计功效: Null	
净增幅(理论): 69.94%	

swipe	
净增幅(实际):	P值: 0.04
-5.29%	校正P值: Null
统计功效: 0.53	
净增幅(理论): -5.00%	

Messages	
净增幅(实际):	P值: 0.55
-1.42%	校正P值: Null
统计功效: 0.09	
净增幅(理论): -0.64%	

match uv	
净增幅(实际):	P值: Null
0.89%	校正P值: Null
统计功效: Null	
净增幅(理论): 0.66%	

Match	
净增幅(实际):	P值: 0.67
1.49%	校正P值: Null
统计功效: 0.07	
净增幅(理论): 1.80%	

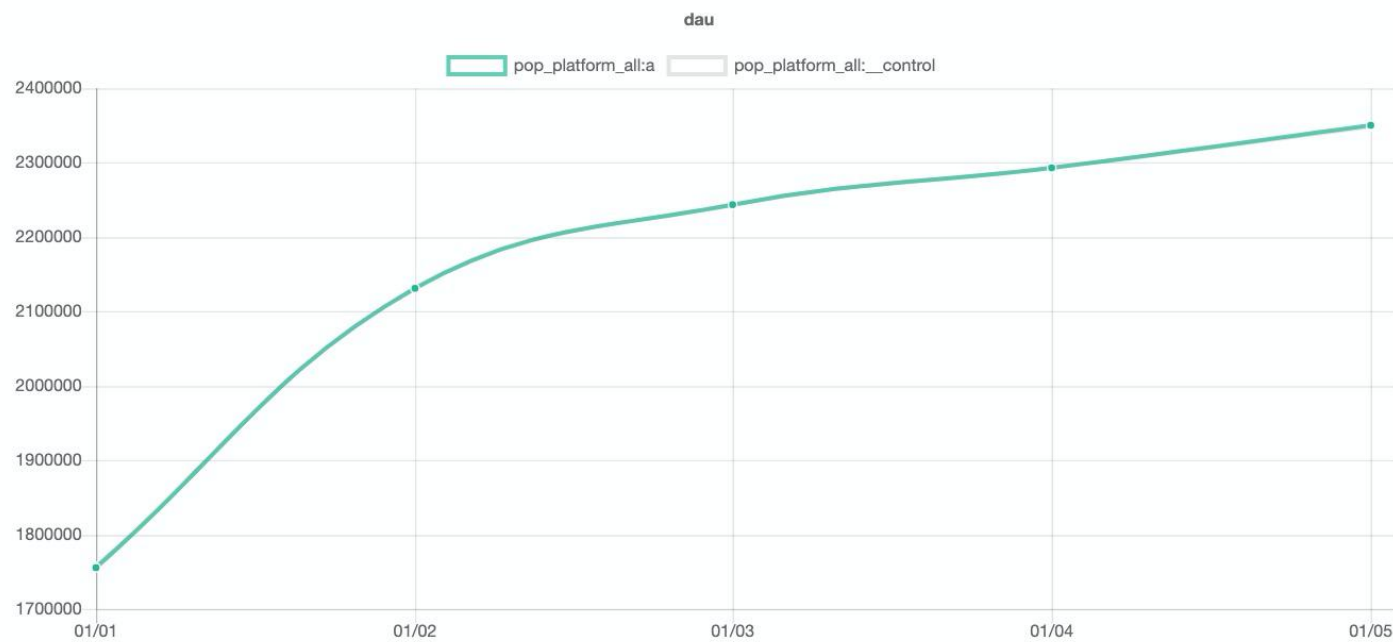
live payment ratio	
净增幅(实际):	P值: Null
-14.41%	校正P值: Null
统计功效: Null	
净增幅(理论): -3.33%	

live dau ratio	
净增幅(实际):	P值: Null
-4.56%	校正P值: Null
统计功效: Null	
净增幅(理论): -2.39%	

Likes	
净增幅(实际):	P值: 0.72
1.54%	校正P值: Null
统计功效: 0.06	
净增幅(理论): 1.01%	

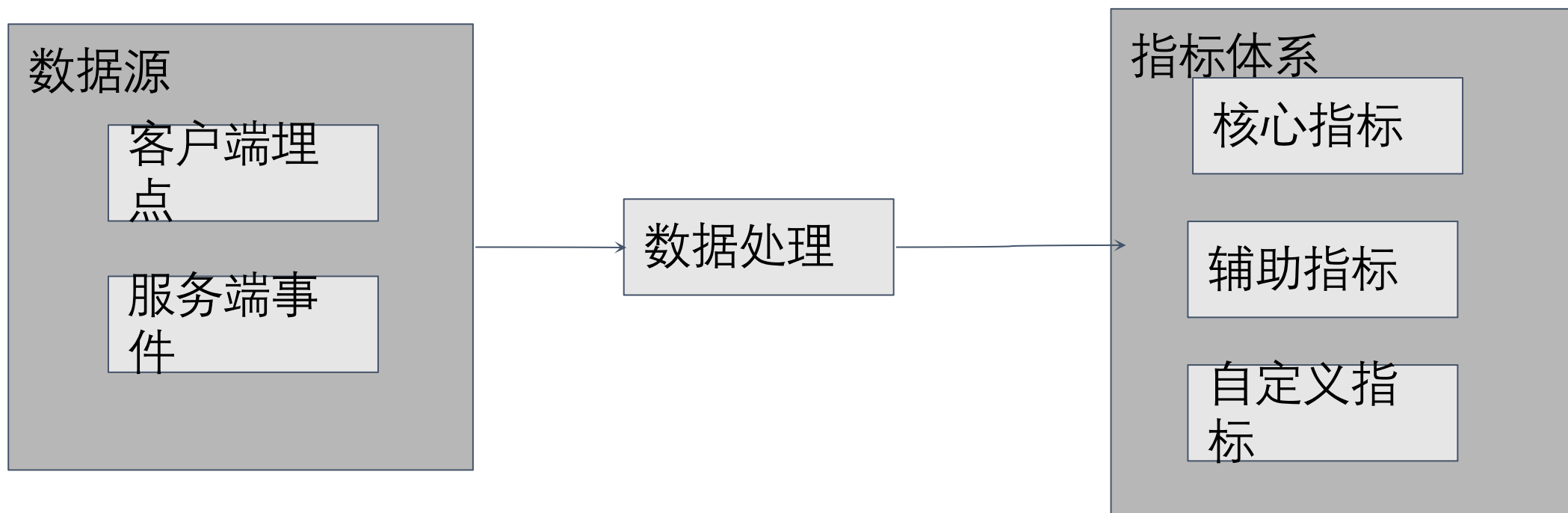
dau retention 1D	
净增幅(实际):	P值: Null
0.91%	校正P值: Null
统计功效: Null	
净增幅(理论): 0.91%	

# 数据分析



- 上线前理论入组增幅: -0.02%
- 上线后实际入组增幅: 0.00%
- 上线后理论入组增幅: -0.01%
- 实际入组置信区间: -0.00 - 0.00
- 指标名称: 日活跃用户
- 指标定义: 每日活跃用户数, 只计算good用户。当日上传过地理位置或更新了last activities。

# 数据分析



- 正确性
- 埋点治理

- 口径一致
- 全局共享
- 提升效率

# 其他问题

---

- 实验的科学性
  - 实验设计与指标选择
- 统计显著性检验
- 异常值剔除
- AA实验耗时，如何让实验更快
- 组间差异如何解决

# TODO

---

- 组件化插件化
- 无缝接入
- 更快的支持产品迭代



The background is a solid orange color. It features several abstract geometric elements: a large, semi-transparent orange circle in the center; a series of thin, concentric white circles on the right side; a white line forming a 'V' shape on the left; and a series of thin, parallel white lines at the bottom left.

**Thank you!**