QCon 全球软件开发大会
INTERNATIONAL SOFTWARE DEVELOPMENT CONFERENCE

BEIJING 2017

QQ空间百亿级流量广告系统海量服务实践

腾讯： 冯启航

主办方 Geekbang. 极客邦科技 InfoQ

内容提要

1、Qzone广告业务背景介绍
2、架构建设过程
　　设计思路
　　一些实际问题和解决
3、ROI提升经验分享
　　实用于内部海量流量ROI优化的 实验+策略+技术
4、小结杂谈

一、QQ空间广告业务背景

QBOSS(Qzone business opreator support system)

业务指标

广告系统：QBOSS(Qzone business opreator support system)
最早定位于Qzone业务支撑系统

1、QQ空间等产品接入广告渠道 数百

2、300亿/日的海量访问 运算，日常峰值40w/s

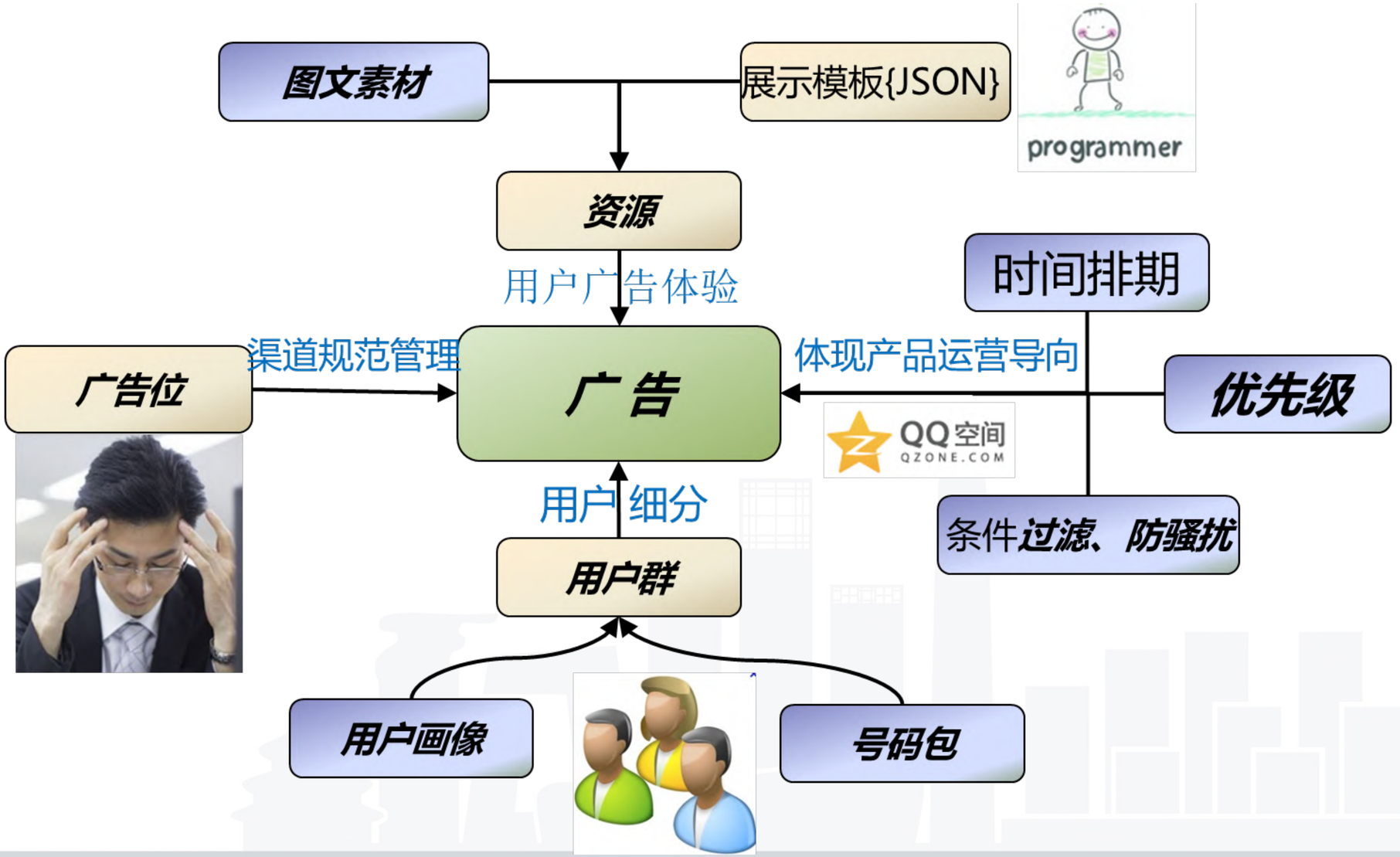
3、性能要求高：大部分渠道广告计算耗时要在50ms内

二、海量服务基础架构建设

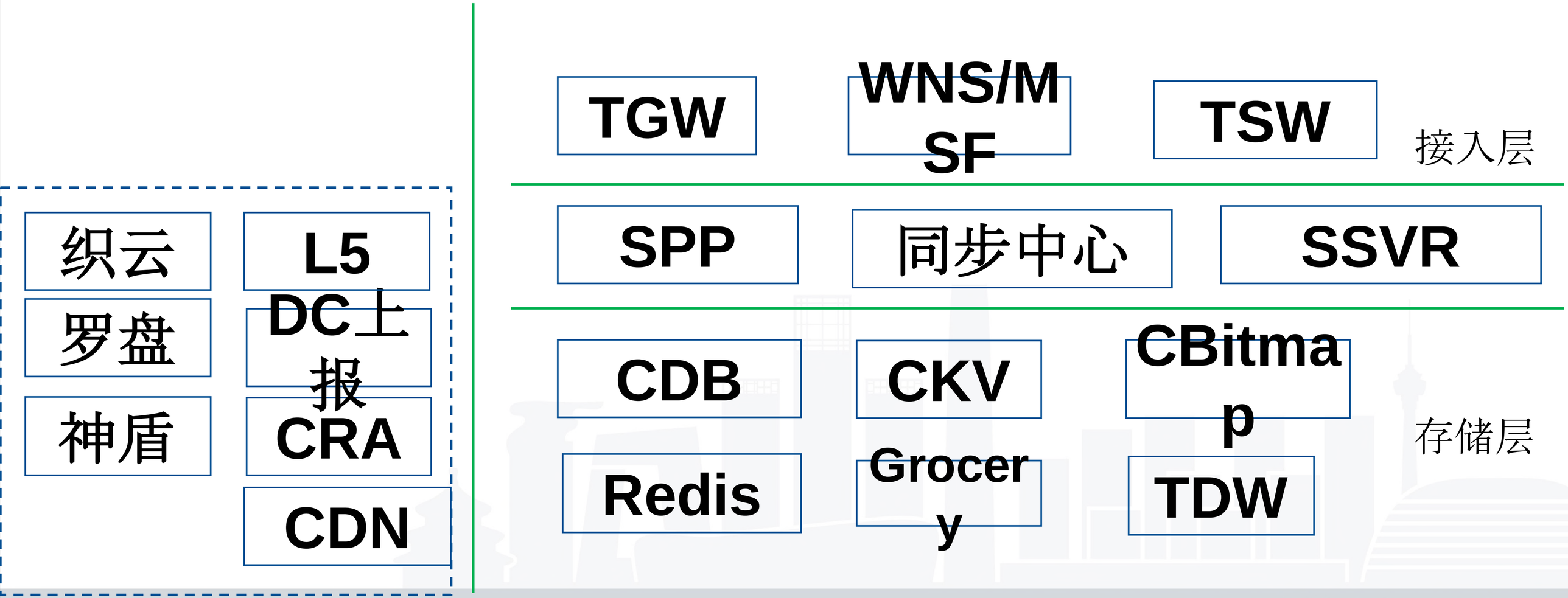
初期要实现的简单但核心的功能

1、渠道
2、防止骚扰，限制广告曝光次数，重复点击等
3、定投: 人群属性；号码包；地域；网络环境...
4、播放端业务和广告投放逻辑脱离耦合
5、广告排期

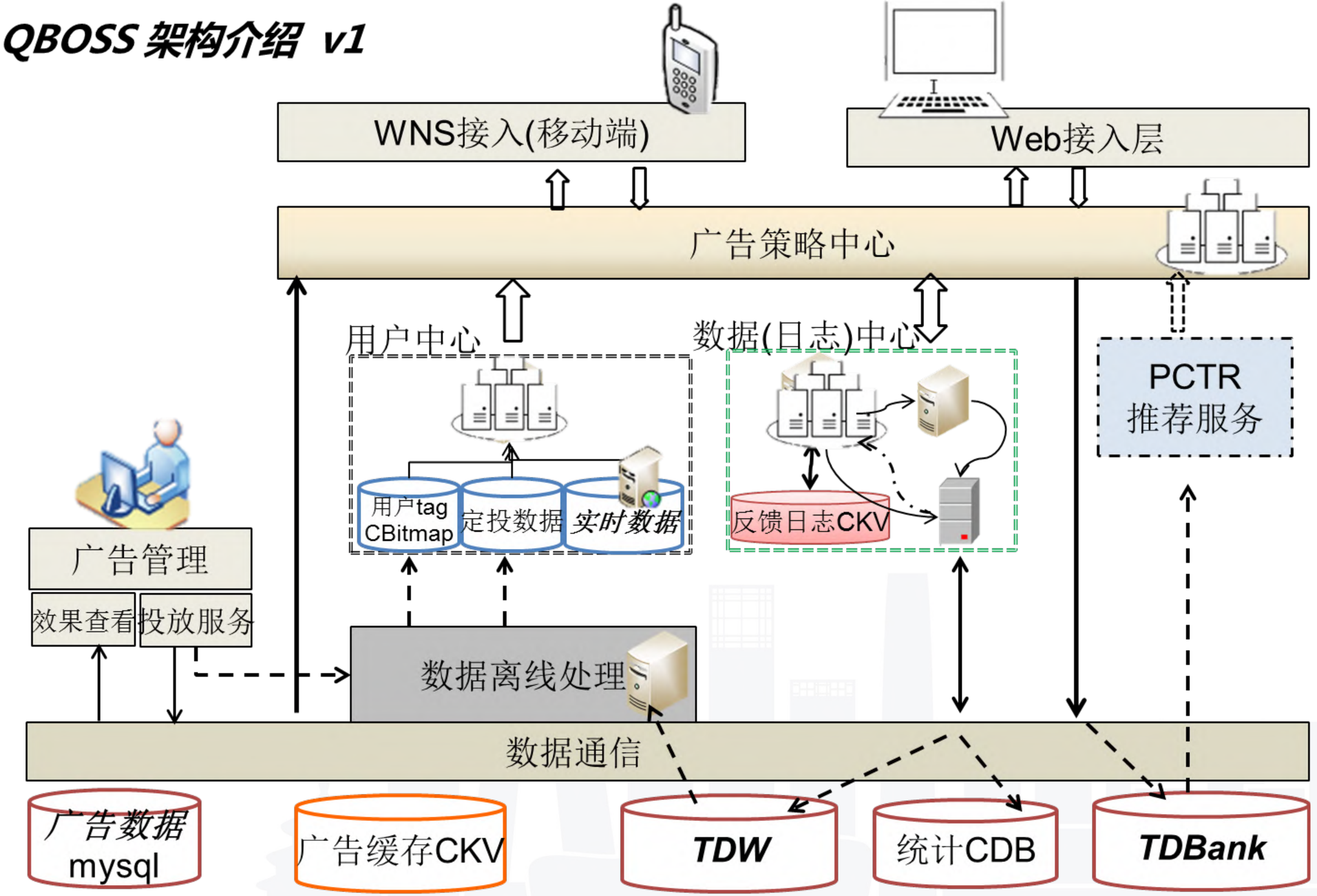
定义秩序


图文素材
展示模板{JSON}
programmer
资源
用户广告体验
时间排期
渠道规范管理
广告位
广 告
体现产品运营导向
优先级
QQ空间
QZONE.COM
用户 细分
条件过滤、防骚扰
用户群
用户画像
号码包

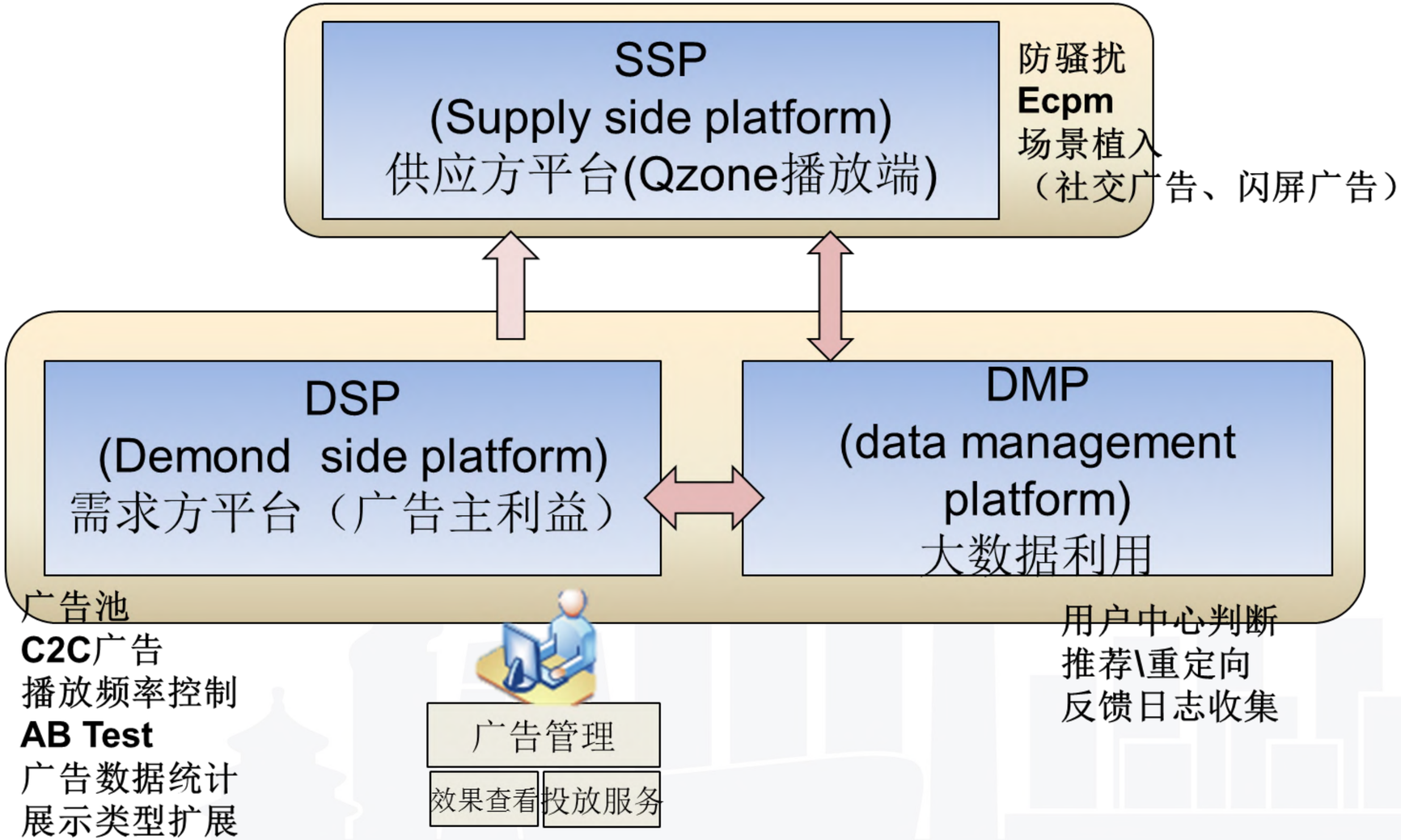
腾讯通用技术积累


织云
罗盘
神盾
L5
DC上报
CRA
CDN
TGW
WNS/MSF
TSW
接入层
SPP
同步中心
SSVR
CDB
CKV
CBitmap
Redis
Grocery
TDW
存储层

QBOSS 架构介绍 v1



策略中心-广告系统的中控

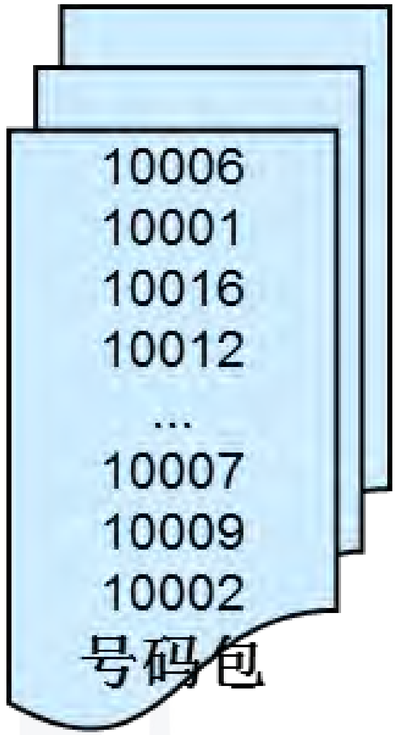


用户中心服务-DMP

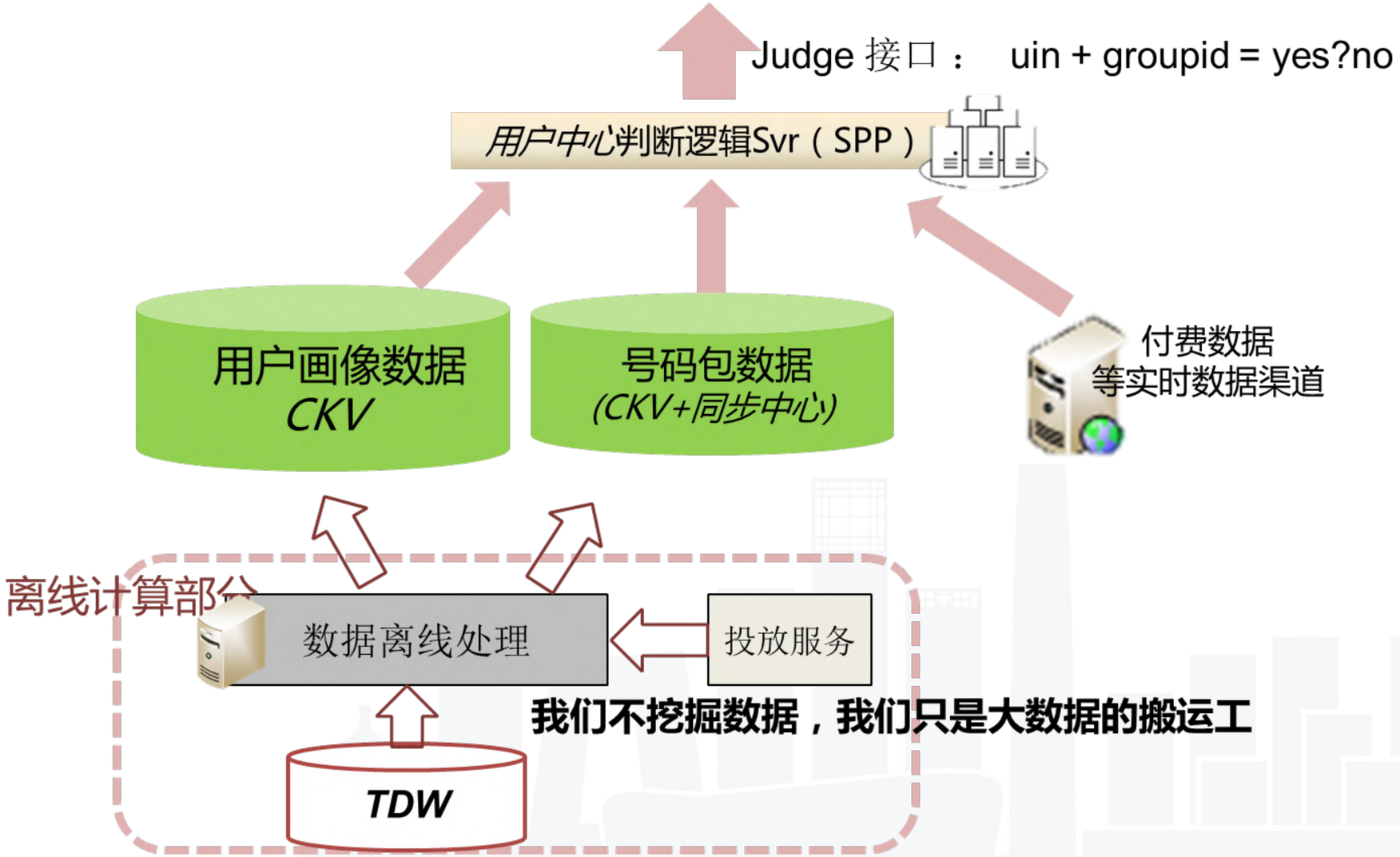
- 黄钻属性
- 基本特征
- Qzone基础活跃
- 空间基础功能PC写操作
- 空间基础功能手机写操作
- 手机游戏类
- PC游戏类
- 电台用户属性
- 黄钻用户生命周期标签
- 黄钻特权偏好
- 黄钻特权偏好(QBoss神盾联合出品)

已经选择的条件

✖	总共登录天数	不活跃	~	不活跃
✖	是否装扮特权活跃	是		
✖	关系链活跃	是		
✖	是否黄钻	是		
✖	相册操作次数	不活跃	~	不活跃



用户中心服务起始架构


Judge 接口： uin + groupid = yes?no
用户中心判断逻辑Svr（SPP）
用户画像数据
CKV
号码包数据
(CKV+同步中心)
付费数据
等实时数据渠道
离线计算部分
数据离线处理
投放服务
我们不挖掘数据，我们只是大数据的搬运工
TDW

用户中心服务面临的问题分析

1、建设一个优秀的DMP(数据管理平台)

tag格式 ：年龄、性别、装扮活跃、登录时间次数。。

tag来源多，要开放

tag数据更新频率不一，有的可以离线缓存，有的需要实时访问

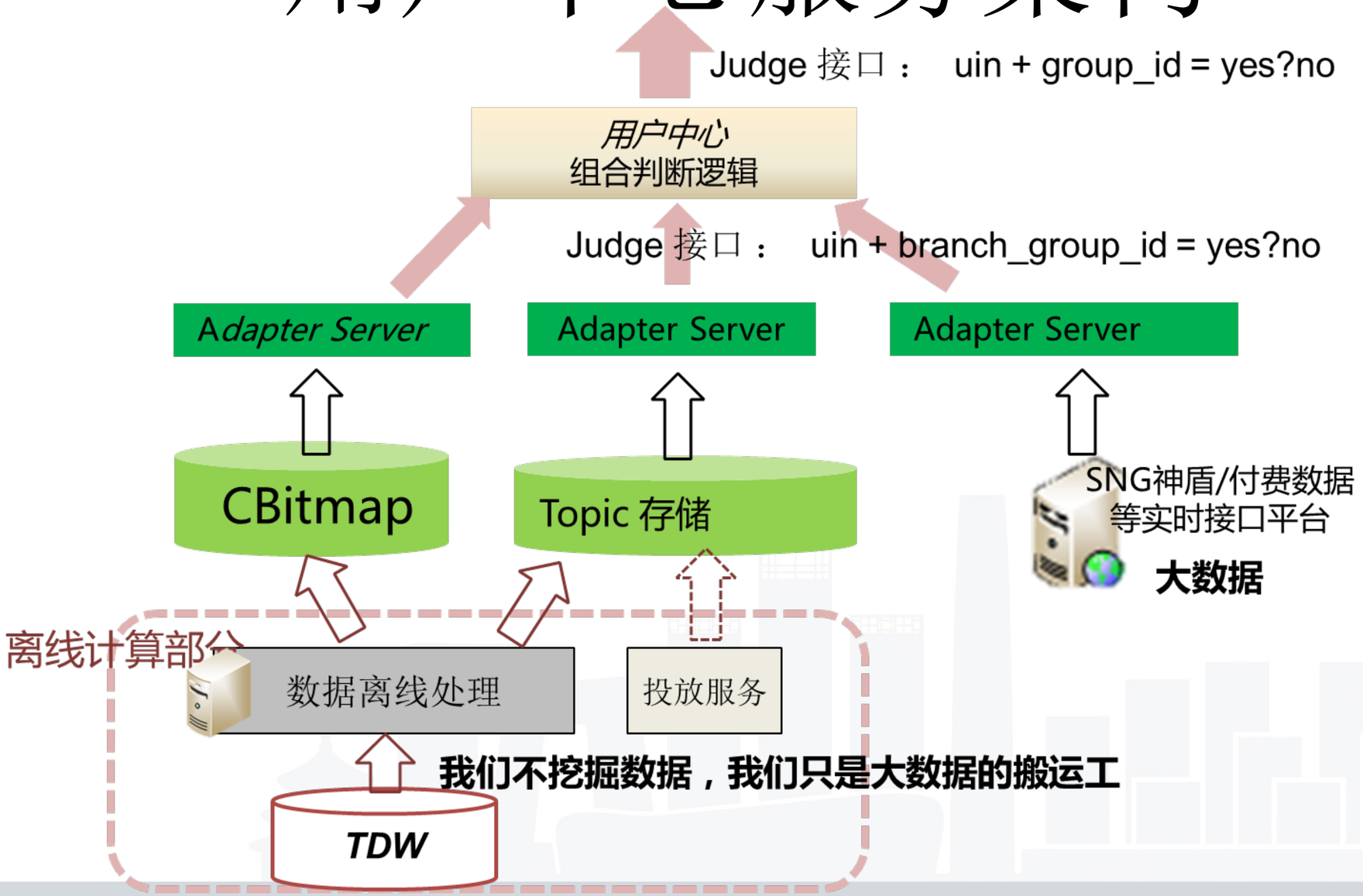
2、投放体验

3、成本

开发效率

内存存储成本

用户中心服务架构

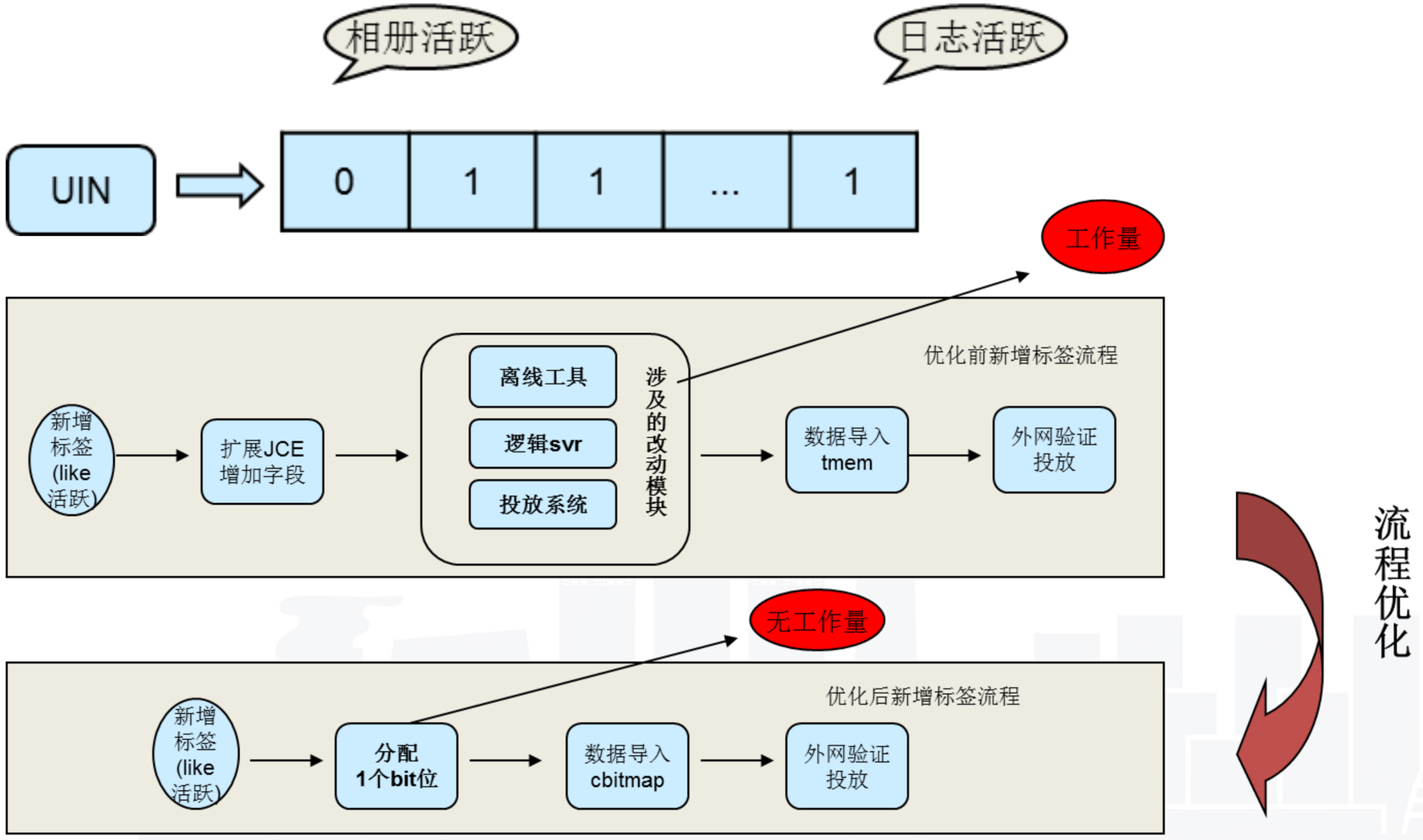

Judge 接口： uin + group_id = yes?no
用户中心
组合判断逻辑
Judge 接口： uin + branch_group_id = yes?no
Adapter Server
Adapter Server
Adapter Server
CBitmap
Topic 存储
SNG神盾/付费数据
等实时接口平台
大数据
离线计算部分
数据离线处理
投放服务
我们不挖掘数据，我们只是大数据的搬运工
TDW

问题1：用户画像tag建设效率

```
struct tUser
{

        0 require string user_id;
        1 optional byte age;
        2 optional byte gender;
        3 optional byte photo_user;
        4 optional byte twwiter_user;
        5 optional byte bvip;
        6 optional byte vip_level;
        7 optional unsigned vip_expire_tm
        ....
}
```

问题1：用户画像tag建设效率



问题2：适配层数据来源

优点：

1、腾讯内部各个大的DMP业务的接入，插件式高效率的开放出去

2、一些实时更新的数据源（用户的增值服务属性）也可以轻松纳入DMP管理

问题3：号码包定投也是个技术活

最早的存储选择nosql型的ckv，设计：

Key （userid）=》 val（[group id A，group id B，group id C]）

优点：

线上服务单次查询，满足批量定投请求

缺点：

1、投放难度很大，大量的投放操作耦合写用户数据操作，造成外网服务波动。上亿的用户号码包等上N小时，接受不了。

2、存储回收操作麻烦，内存成本持续上涨

问题3：号码包定投也是个技术活

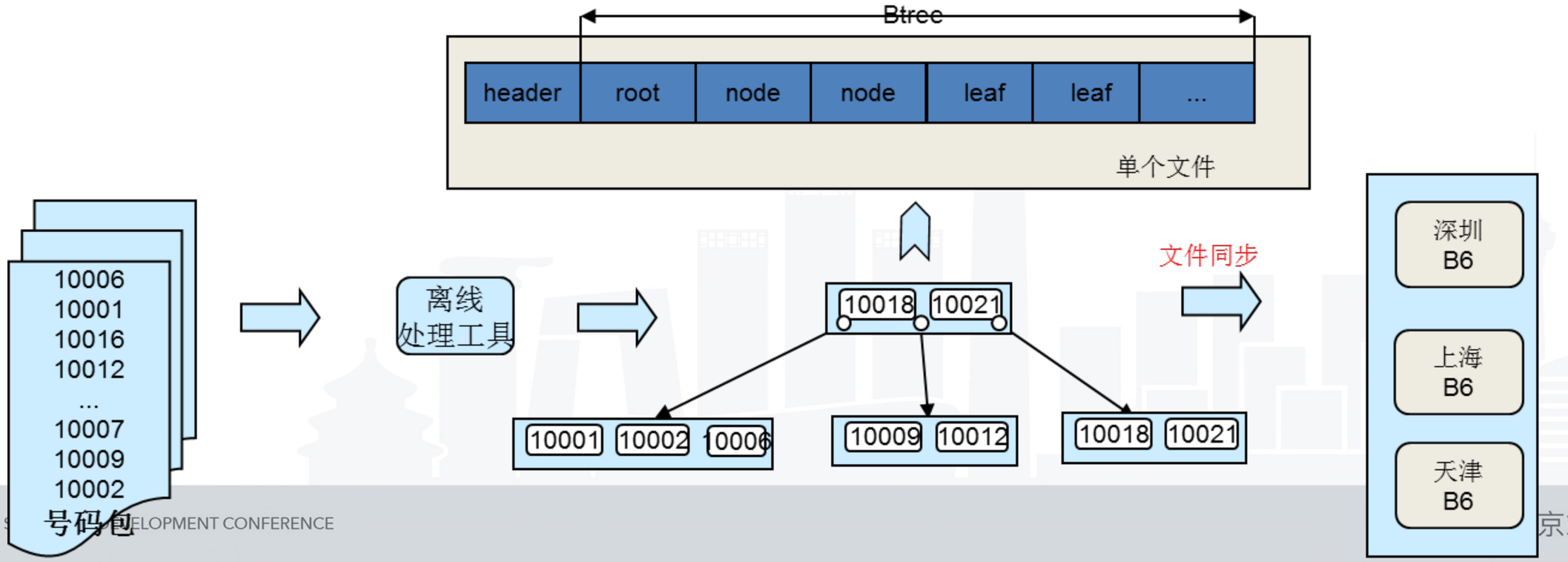
解决方法：

在Qzone自研的存储利器：好友参与系统.适当改造出适合号码包的topic存储

号码包（topic）　　group id =》 ([uin ,uin ,uin..])

把顺序的号码包，根据存储设计结构，用离线工具处理成btree树。

并用tmpfs 使得每个btree文件就是可见的号码包。 几分钟完成投放



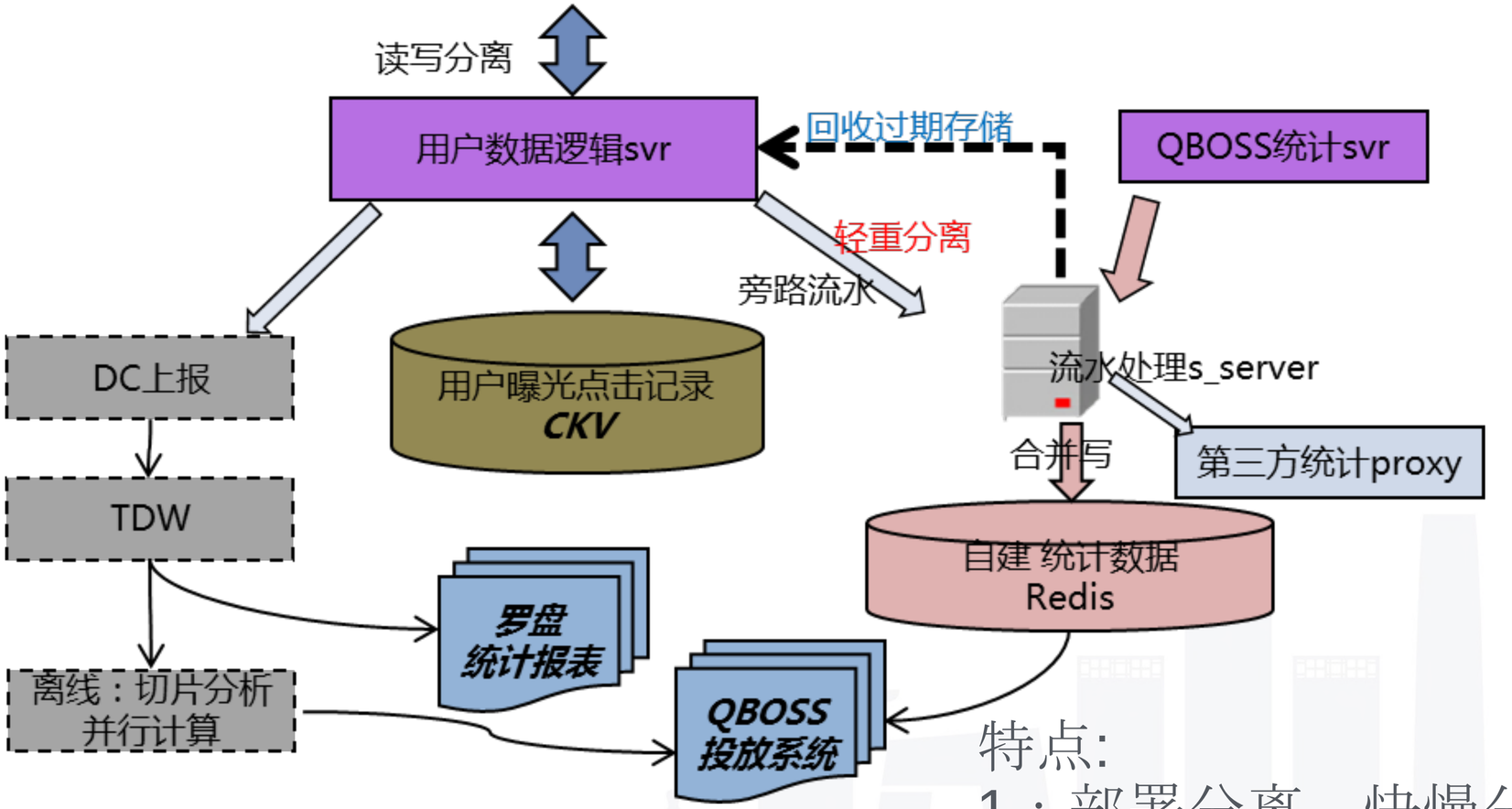
数据中心服务难点

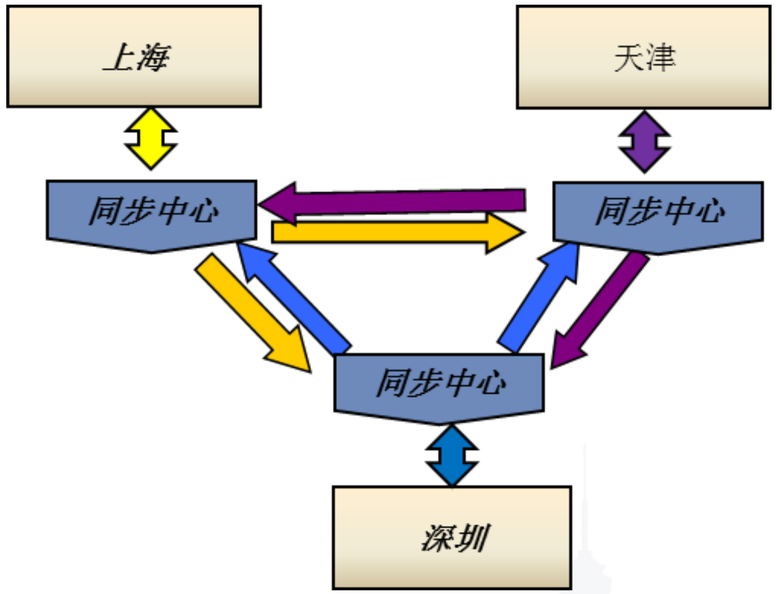
1、读多写多模型 ，且读写访问量最高近20w/s ，性能要求也高

2、异地服务接入，数据同步问题，量大了容易堵塞

3、日志数据均在内存里，CKV成本持续上涨需要节制

数据中心服务架构剖析





特点:

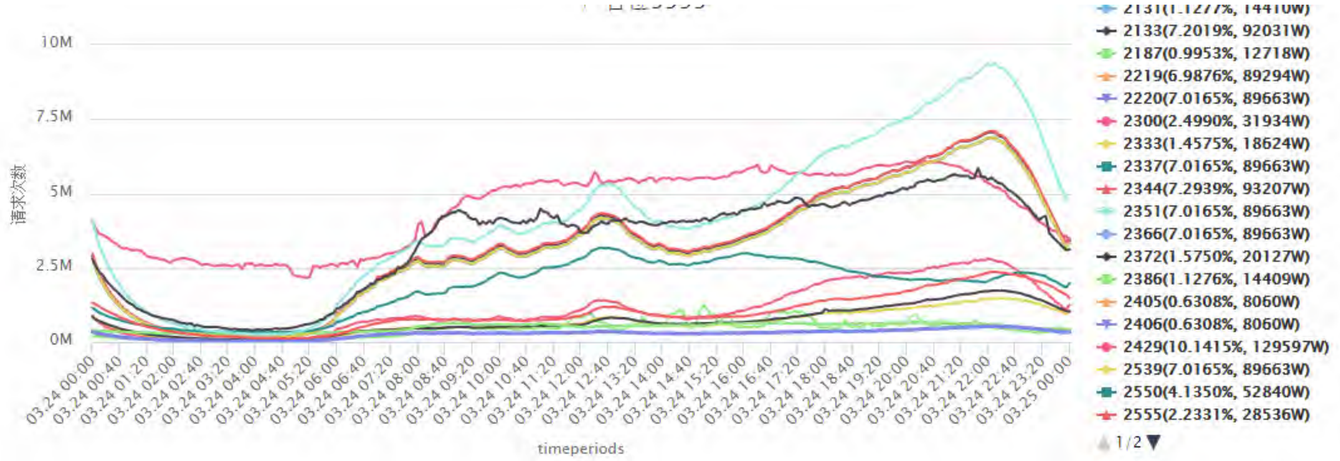
1：部署分离，快慢分离 回收存储、统计、监控

2：双通道统计保证数据准确

3：防重试避堵塞的异地同步三地写三地读服务

平台海量服务运营能力

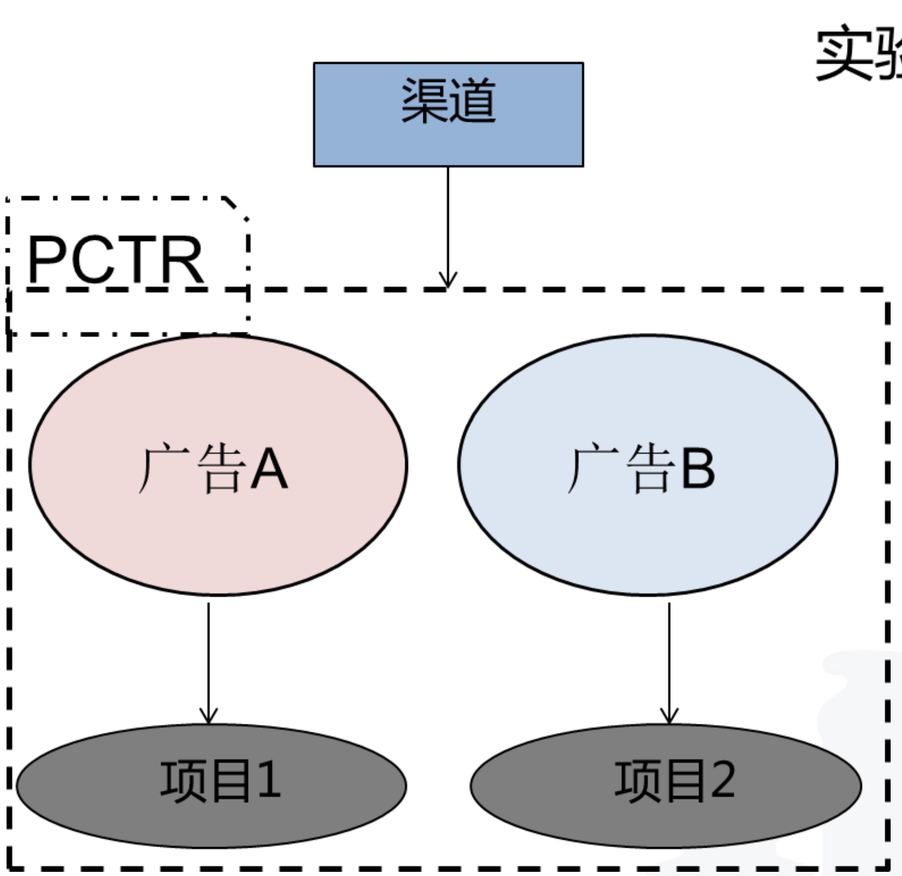
1、监控能力
接口级别监控； 用户端延时监控；
广告数据监控；
2、服务质量，容量
动态变化的负载预警
高性能增强容量buffer
各种分离部署减少毛刺
3、容灾
L5负载均衡
SET管理
支持城市级别容灾（天津大爆炸）
重在细节,找短板



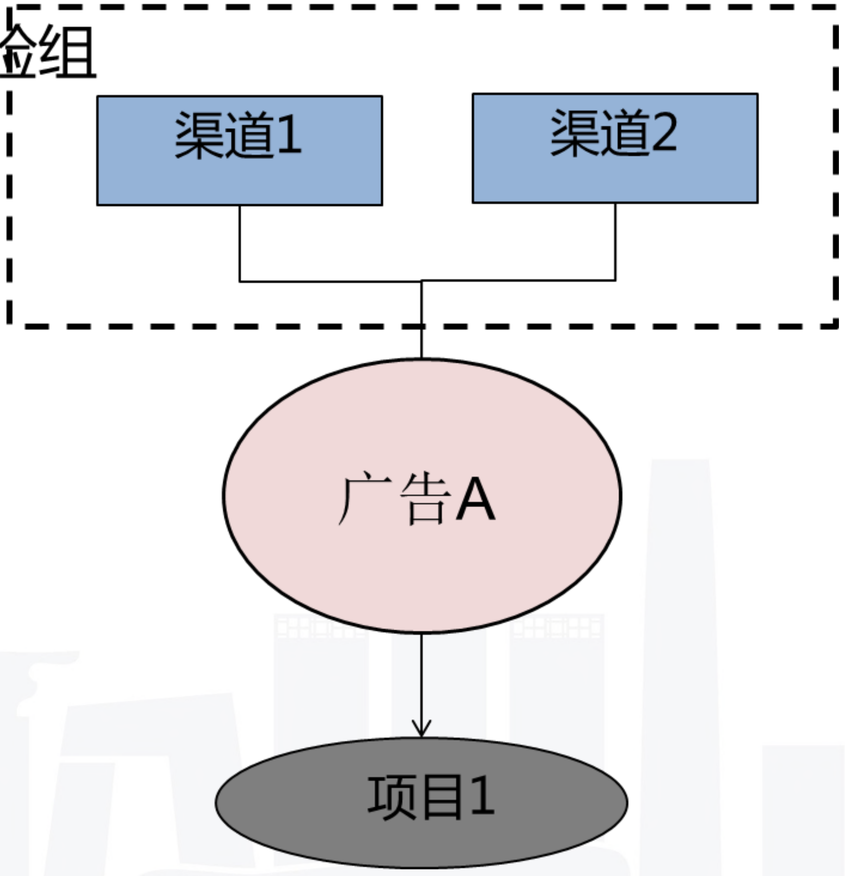
三、广告系统ROI优化之路

ROI提升100%实践，走一条符合“国情”的道路

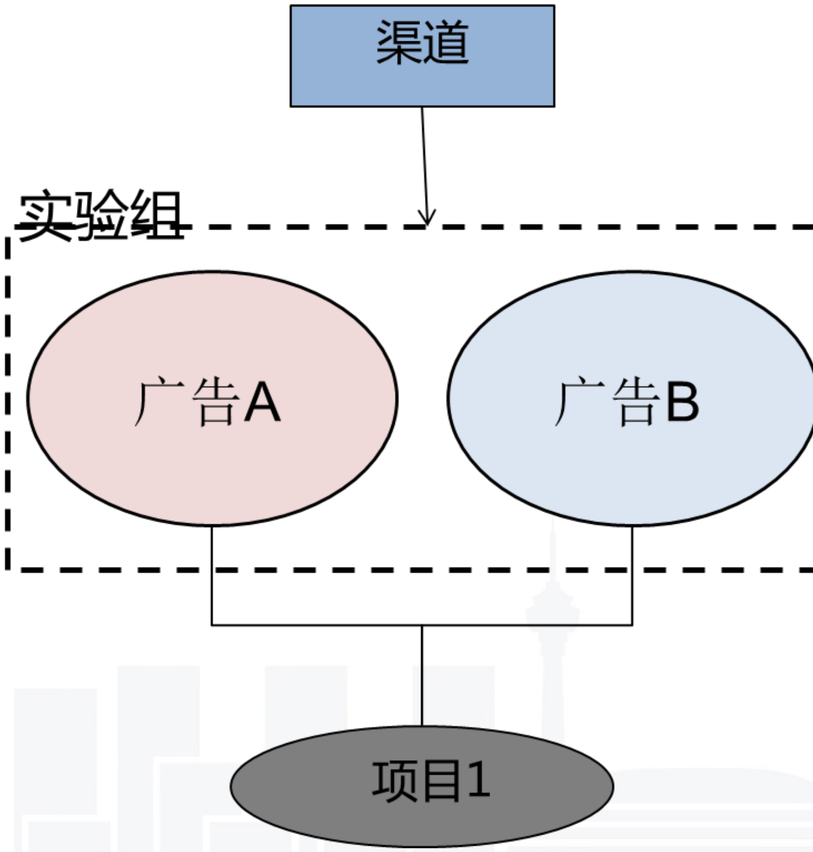
ROI提升实践



方案A



方案B



方案C

ROI提升实践

另外一些很重要的功能点

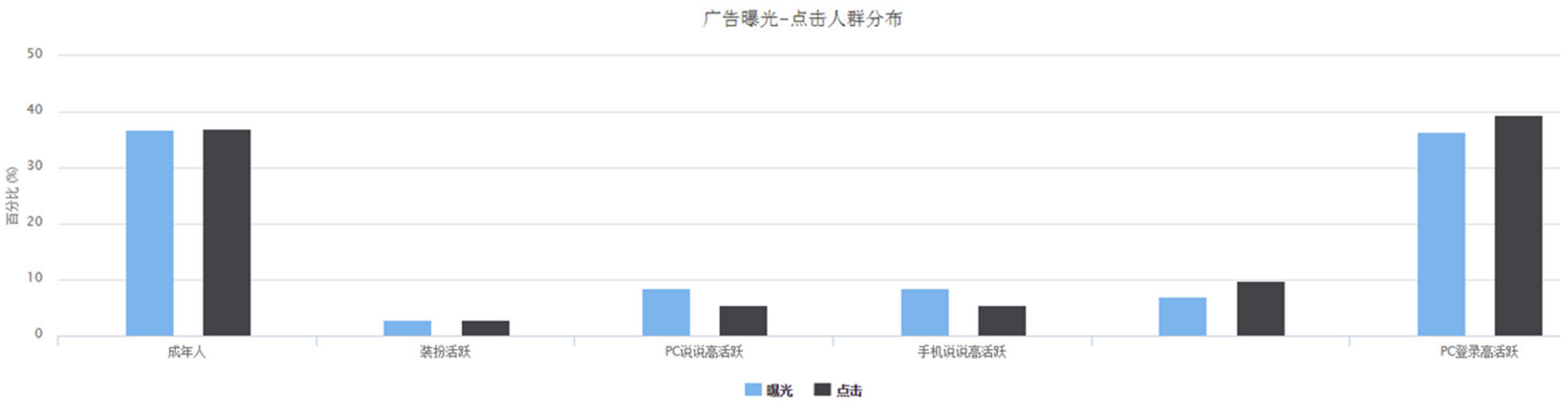
负反馈 ：大数据分析如果某渠道用户看7次都不点，或者三天内重复关闭，这类用户的兴趣点很高。除非定投或者平台强策略，建议不要再短期内继续给他曝光。

频率限制 ：让流量在长达一天甚至更长的周期去平稳消耗。还能起到保护目标站点的作用，对转化率的提升真的有作用。

ROI提升实践

QBOSS人群分析统计

开始时间: 20160105 0000　结束时间: 20160105 2358　广告ID: 96484　选择标签　展示数据



标签	曝光用户数	曝光用户占比	点击用户数	点击用户占比
成年人	9260134	36.79%	21833	36.95%
装扮活跃	693941	2.76%	1707	2.89%
PC说说高活跃	2133758	8.48%	3242	5.49%
手机说说高活跃	2153019	8.55%	3235	5.47%
	1776833	7.06%	5798	9.81%
PC登录高活跃	9155477	36.37%	23280	39.39%

四、小结


QBOSS
动态运营
大胆合理的技术选型-敏捷开发
先抗住再优化
联调工具，wiki建设
接入管理
容量分配管理
标准准入流程
权限管理
定期review资源消耗
web层\svr层两种接入方式
扩展能力
接口设计应用tlv协议
业务 / 广告 解耦
独特的“展示类型”DIY
全网调度
模块化
SET标准化
量化服务能力
平滑扩容/缩容
L5负载均衡
定期演习/异地容灾
立体监控
模块接口调用监控
自研系统指标监控
监控收入、活跃结果数据
邮件\短信\MOA多种触达手段
大系统小做
SOA-解耦
专业的事情交给专业团队
灰度发布
CC/ars发布系统
功能上线按渠道\用户纬度灰度
柔性可用
全系统梳理关键路径
逐层设计容错柔性策略
质量管理
代码review提交
免测
单元测试
用例回归
物理容灾
同城多IDC部署
投放系统部署多套
轻重分离
CAP应用
结合容错特点设计一致性要求
成熟的同步中心，异地同步

系统架构设计心得

1、技术价值一定要体现在业务上
2、职责识别、抽象能力分清大方向
3、大系统小做。KISS原则
4、抽象一个层，可以解决很多复杂问题
5、善于在存储上做性能提升，例如redis的成功
6、低成本，一定要更重视去业务场景上去挖
7、无状态的服务，负载均衡、一致性、有利于容灾和维护
8、异步、快慢、读写各种分离部署
9、架构设计前依赖数据经验推演决策；上线后一定要用数据验证。

Thanks!

INTERNATIONAL SOFTWARE DEVELOPMENT CONFERENCE

主办方 Geekbang. 极客邦科技 InfoQ

联系邮箱:fqihang@qq.com