

MTSC2017

第三届中国移动互联网测试

主办方: TesterHome

让数据为质量服务

王胜 2018-07

MTSC2018

第四届中国移动互联网测试开发大会

TesterHome

IT大咖说

TesterHome



- 2014年加入百度搜索公司质量部

- 搜索公司质量工程部大数据专项负责人

- 主要工作：

- ✓质量数据平台

- ✓质量数据治理

- ✓基于大数据的质量场景挖掘

MTSC2018

第四届中国移动互联网测试开发大会

- 之前供职于腾讯电商

- 腾讯电商（易讯，拍拍，团购）



- 部门对QA在产品质量构建提出的新要求

- 基于大数据的泛质量场景诉求：

商业推广风险挖掘，定位排查，行为分析，性能分析等等

- 质量闭环（效率）：

数据充分利用，以帮助质量活动提升效率和工具优化

- 平台化（成本）：

更多的QA，更低成本的使用

MTSC2018
第四届中国移动互联网测试开发大会

有数据



用户数据

! 用户行为
! 用户画像
!



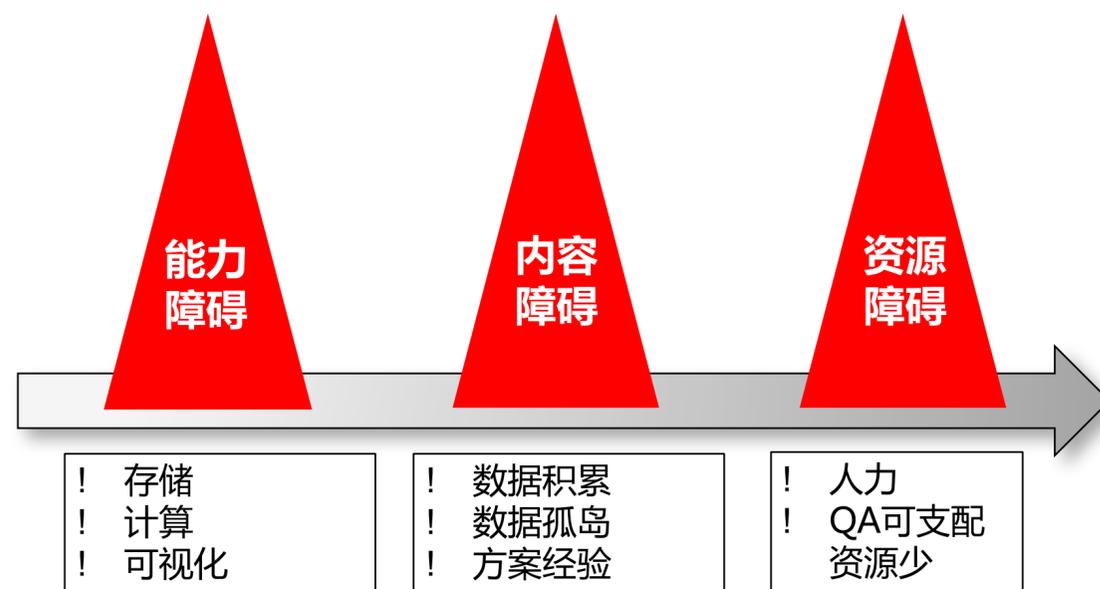
业务数据

! 营销数据
! 系统间调用日志
!



质量数据

! 需求, 设计, FQGH
! ODT, Bug
!



有诉求



测试监控



质量度量



竞品分析



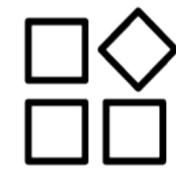
定位排查



性能分析



行为分析



灰度分析



风险挖掘



QUKHST



质量数据平台

(一)



质量数据地图

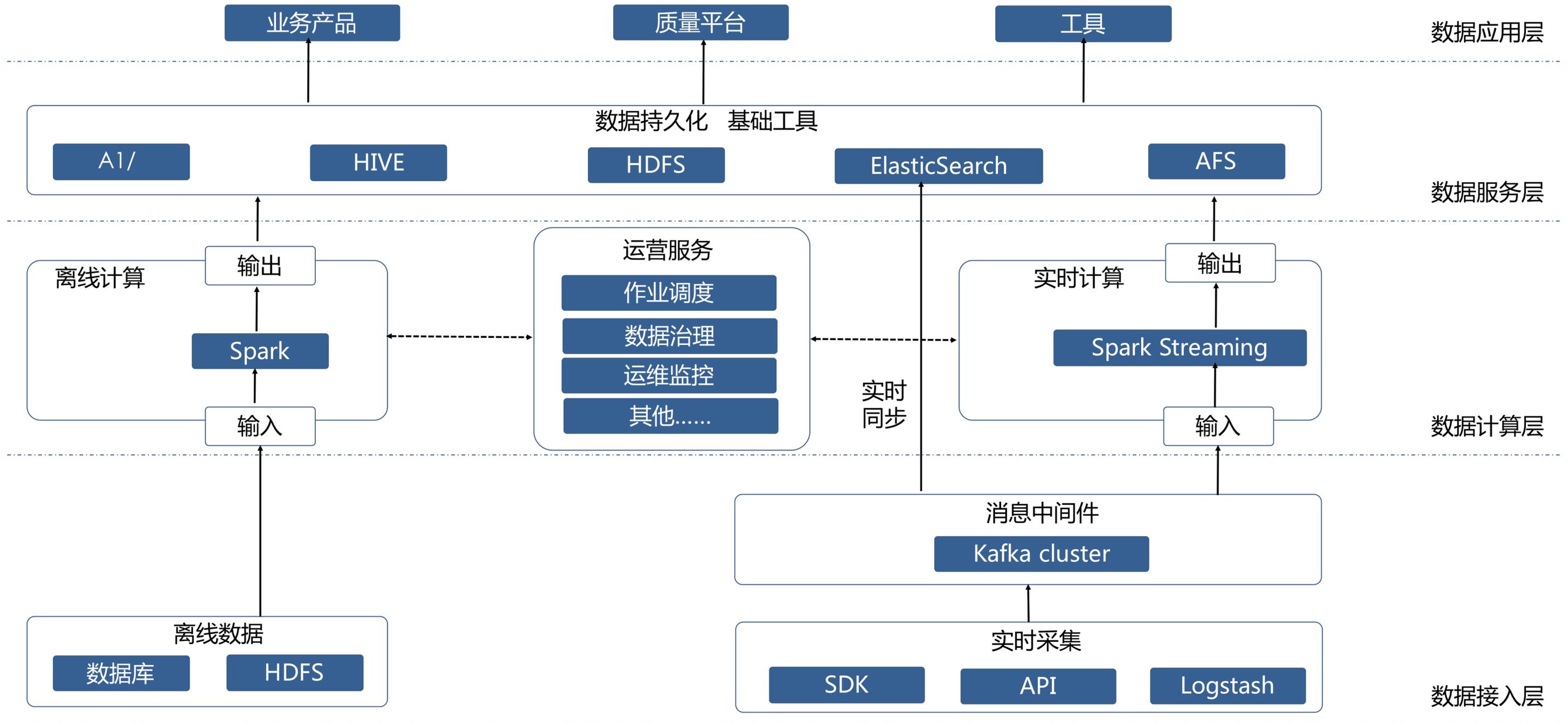
(二)



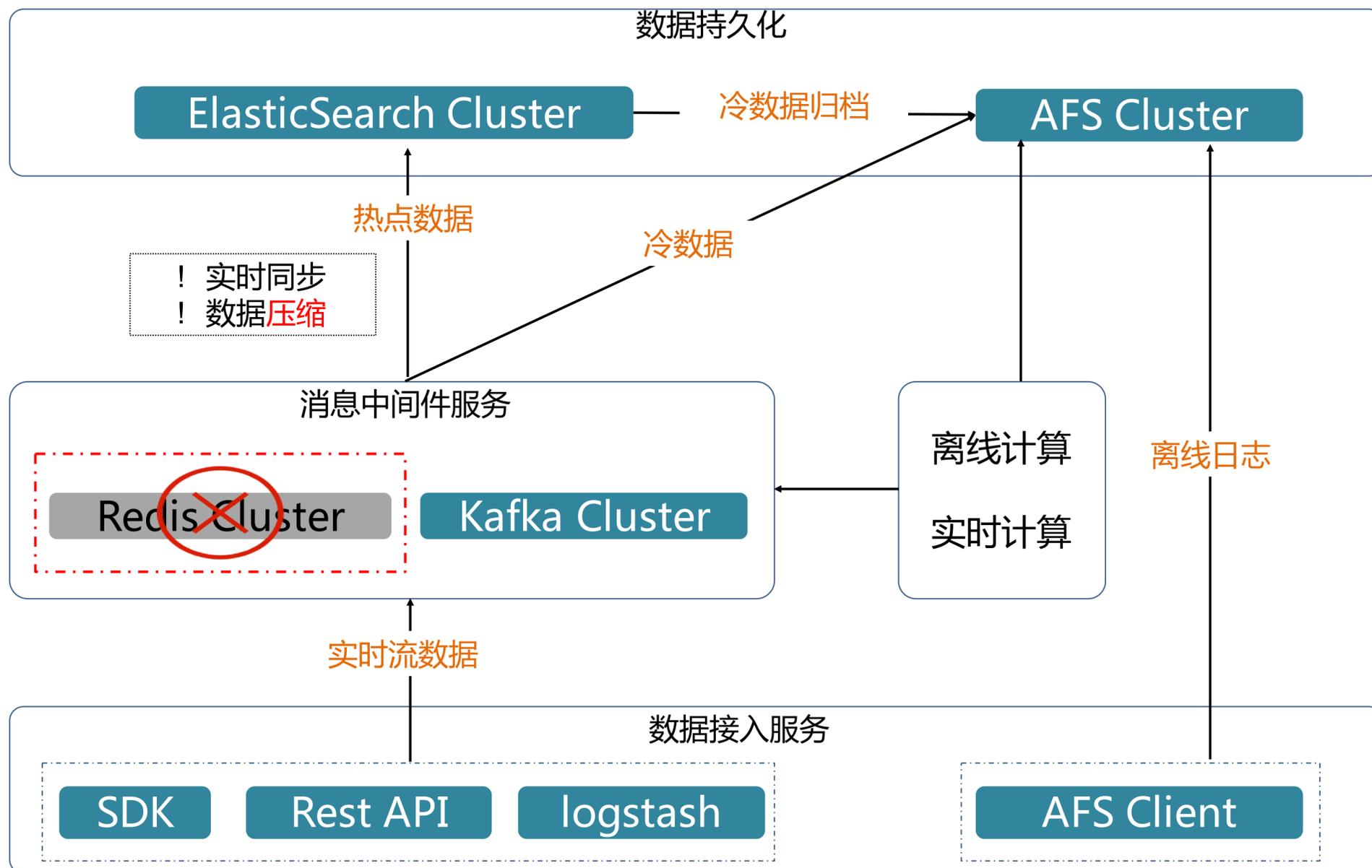
数据应用案例

(三)

MTSC2018
第四届中国移动互联网测试开发大会



背景：1、高性能、高可靠要求；2、更少的资源存更多的数据

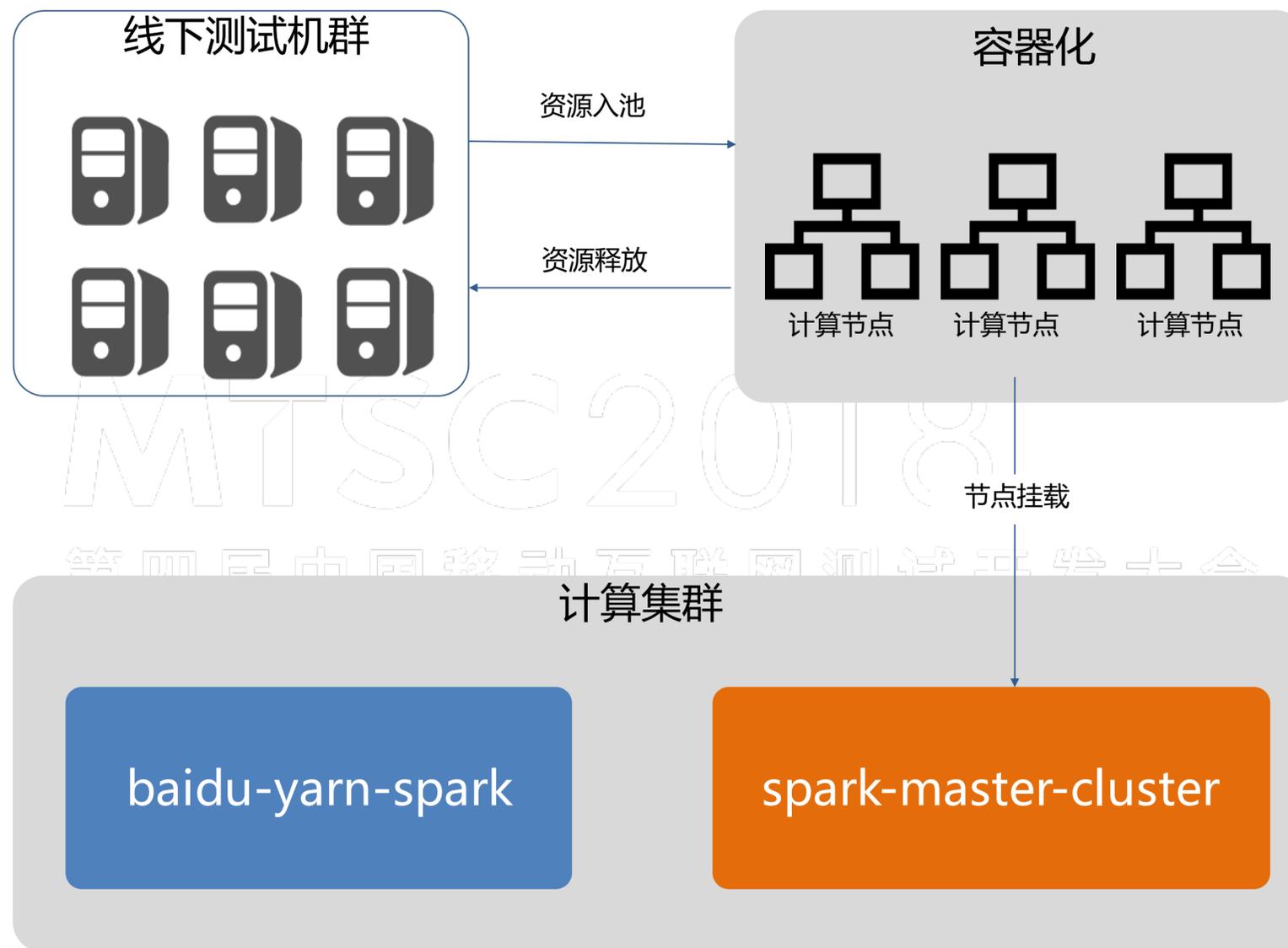


- ! 数据限流**
缓解HT的写入QPS压力
- ! 冷热分离**
保证热点数据检索性能的同时，缓解HT存储压力
- ! 数据压缩**
根据HT索引规则，对非索引字段进行4B69压缩



测试机群特点：

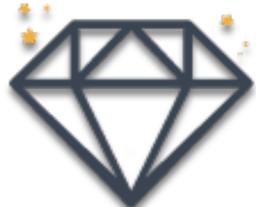
- 数量：5000+；
- 非测试时间，利用率低



资源入池标准：

- cpu idle > 90%
- mem idle > 30%
- 22 : 00~09 : 00

- 对满足条件的线下测试机群，利用容器化技术，动态构建spark计算节点
- 满足计算诉求的同时，降低了资源成本并提升了资源的利用率



质量数据平台

(一)



质量数据地图

(二)



数据应用案例

(三)

MTSC2018
第四届中国移动互联网测试开发大会

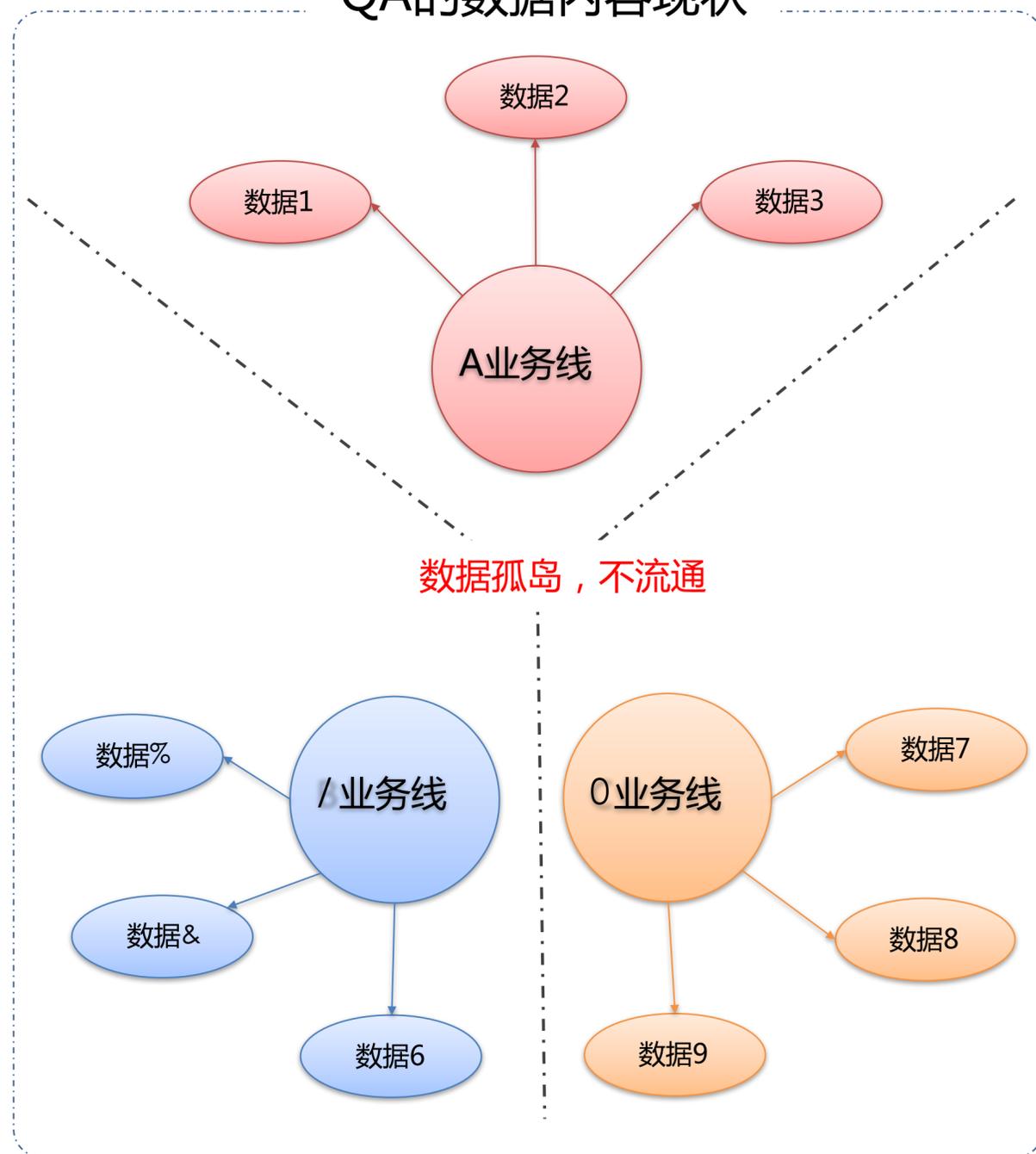


	需求	编码	提测	测试	预发/灰度	发布	线上
story	✓						
bug	✓	✓	✓	✓	✓		✓
code		✓					
Case	✓	✓		✓			
覆盖率		✓	✓	✓	✓		
行为数据				✓	✓		✓
服务调用				✓	✓		✓
CRASH				✓	✓		✓
打包/编译/部署			✓			✓	
灰度					✓		
泛反馈							✓

注：

- 1、case包括单元测试，功能测试，UI自动化，接口自动化用例等
- 2、覆盖率包括UT覆盖率，功能覆盖率，各种自动化覆盖率等

QA的数据内容现状



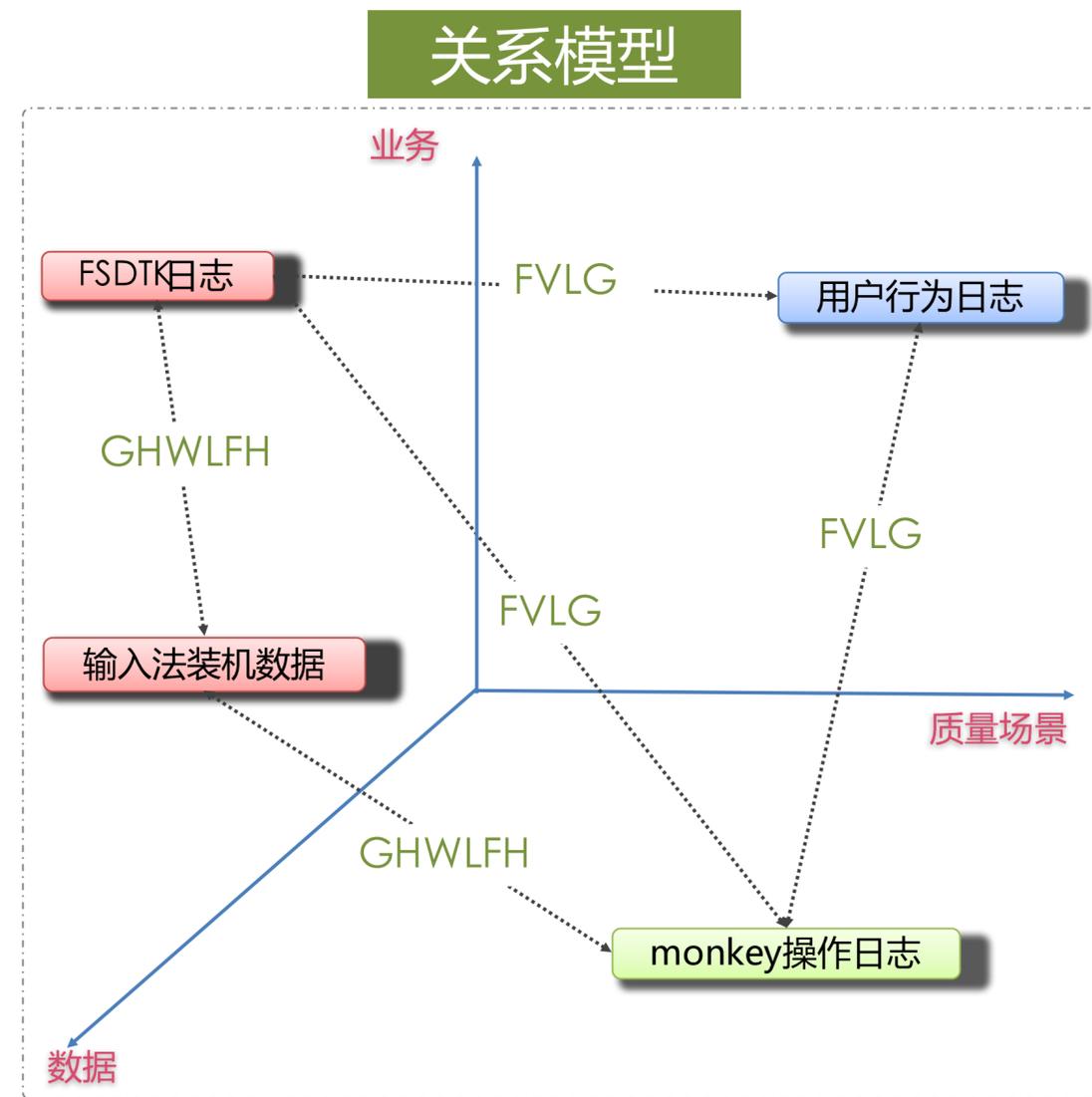
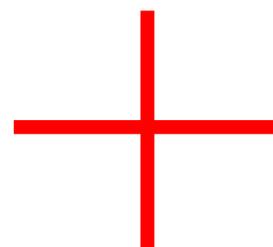
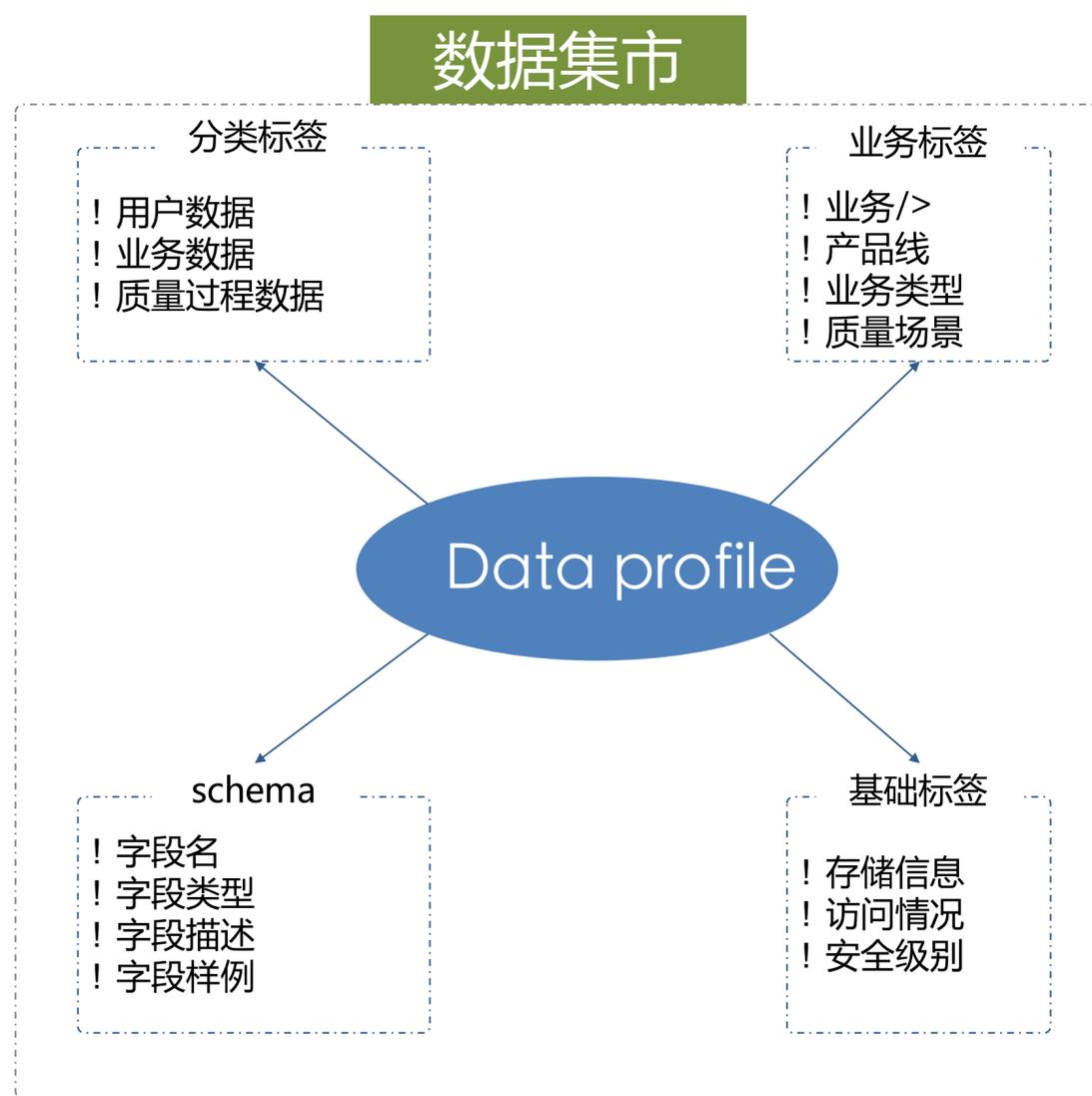
问题分析

- ! 数据流通差:**
业务线QA的数据相对独立, 不流通或流通效率低
- ! 信息不对称:**
构建场景找不到数据或者不知道去哪里找
- ! 数据积累少:**
很多有价值的数据没有沉淀下来

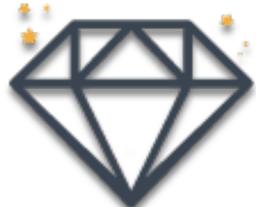
质量数据地图

目标

- ! 打破使用阻力, 提升流通效率:**
安全前提下, 打通数据流通的壁垒
- ! 构建关联模型, 快速高效使用**
更快速找到数据, 更高效的使用数据
- ! 数据资产沉淀:**
数据从无到有, 从零星到丰富



- ! 通过构建质量数据市集，将数据管起来
- ! 通过构建关系模型，将原本独立的数据节点连接起来，发挥数据联动价值



质量数据平台

(一)



质量数据地图

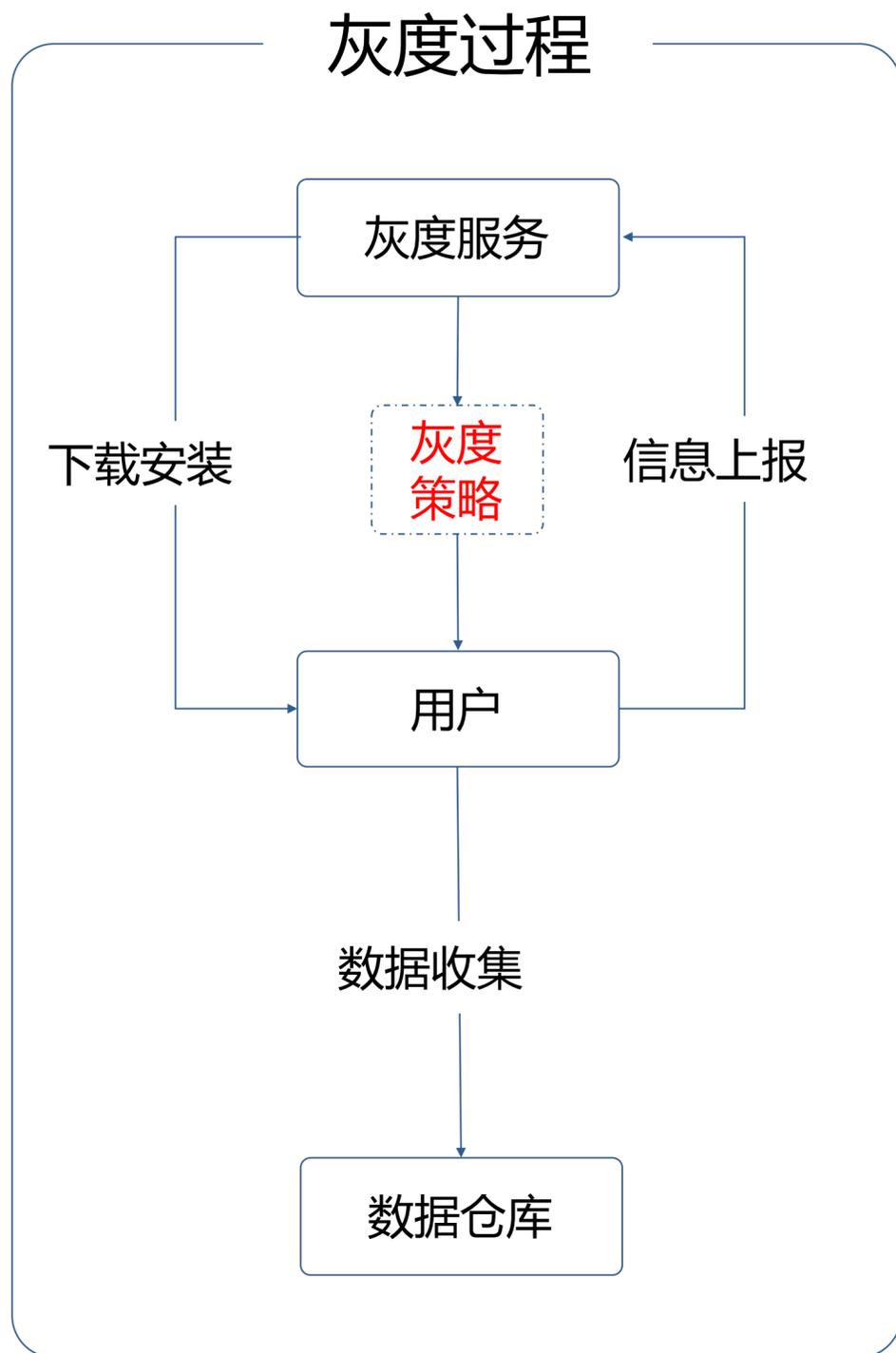
(二)



数据应用案例

(三)

MTSC2018
第四届中国移动互联网测试开发大会



Crash发生的原因分析：

- 内存，网络因素管理错误（内存过低 or 内存泄露）
- 程序逻辑错误（数组越界，堆栈溢出，并发等等）
- 环境问题（设备兼容，地域）

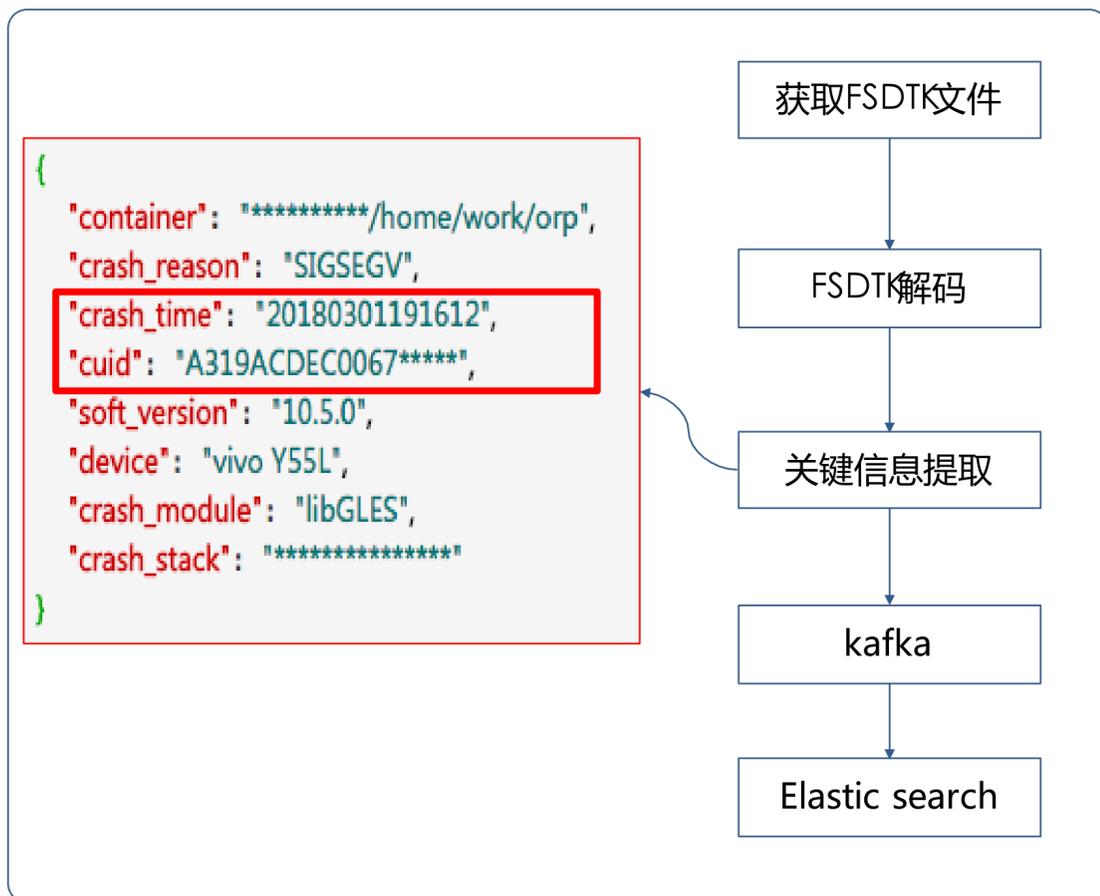
堆栈信息有时无法分析出因**环境问题**引发的crash

解决2个问题

- Crash发生时，用户的环境是什么样子？
- **Crash发生前，用户都干了什么？**

用户行为

OSDTK原文数据



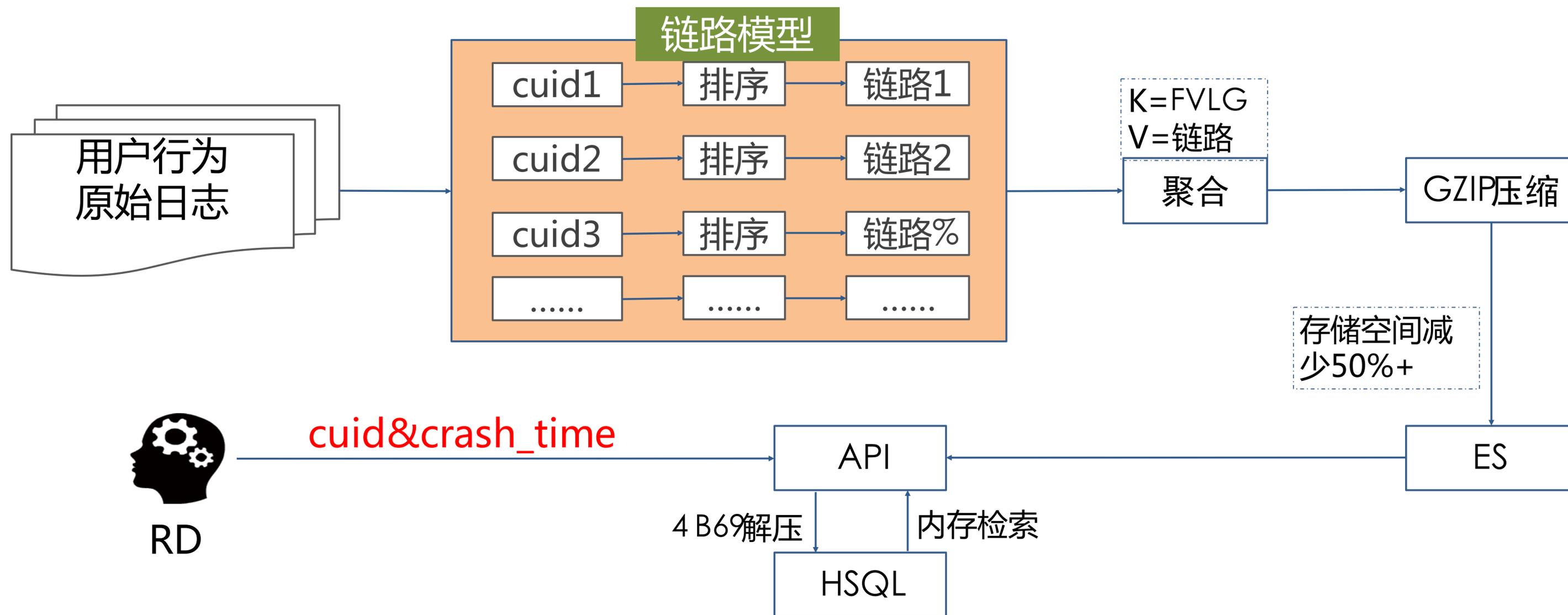
Whoc - Whenc - Wherec

用户行为打点数据

```
{
  "os_name": "android", #操作系统
  "soft_version": "10.4.0.11", #app版本
  "cuid": "D7997DD14428B0*****", #用户唯一标识
  "item": {
    "content": {"type": "1", "skinid": "19570"},
    "id": "227",
    "timestamp": "1521716738281", #用户操作发生的时间
    "type": "0"
  },
  "id": 127, #操作id标识
  "manufacturer": "Xiaomi", #用户终端品牌
  "mode_id": "MIX 2", #用户终端机型
  "os_version": "7.1.1", #用户终端系统版本
  "net_type": "wifi", #用户网络类型
  "province": "北京", #用户所在省份
  "city": "北京" #用户所在城市
  .....
}
```

Whatc





处理步骤：

- 1、以FVLG(用户维度)，做SHGVF计算
- 2、对每个数据集按照操作时间进行排序
- 3、对行为链路进行压缩，以小时粒度构建索引，存储于HT
- % 开发着根据条件查询用户某时间段类的操作链路

```
{  
  "os_name": "android", #操作系统  
  "soft_version": "10.4.0.11", #app版本  
  "cuid": "D7997DD14428B0*****", #用户唯一标识  
  "item": {  
    "content": {"type": "1", "skinid": "19570"},  
    "id": "227",  
    "timestamp": "1521716738281", #用户操作发生的时间  
    "type": "0"  
  },  
  "id": 127, #操作id标识  
  "manufacturer": "Xiaomi", #用户终端品牌  
  "mode_id": "MIX 2", #用户终端机型  
  "os_version": "7.1.1", #用户终端系统版本  
  "net_type": "wifi", #用户网络类型  
  "province": "北京", #用户所在省份  
  "city": "北京", #用户所在城市  
  .....  
}
```



用户操作链路

语言不互通

执行器回放

- ! XLPGQXc
- ! DFULWdJZ
- ! xpath
- ! LG
- ! namec

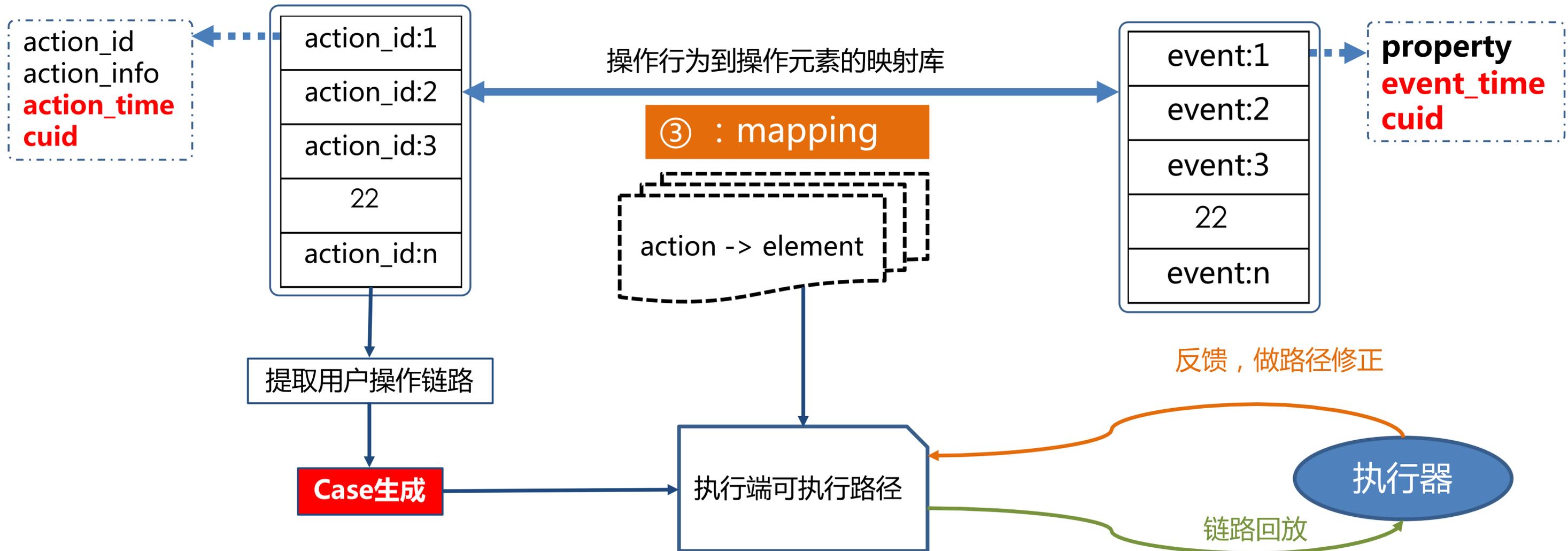
用户行为链路

数据转换

测试执行器语言

①：用户行为日志

②：monkey操作日志



原则：找出覆盖所有行为的最短路径，以提升测试执行效率

1-提取用户操作链路

👤	A	B	C	A	B	C	D	E
👤	A	E	G	A	B	C	B	C

2-起始点切分

A	B	C		
A	B	C	D	E
A	E	G		
A	B	C	B	C

3-环切分

A	B	C		
B	C			
A	B	C		
A	B	C	D	E
A	E	G		

4-去重

A3	B3	C3	D1	E1
A1	E1	G1		

执行器



MTSC2018
第四届中国移动互联网测试开发大会

定义：发生FSDTK的操作链路和操作的控件

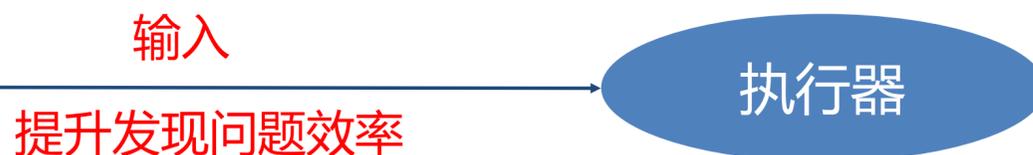
行为链路

操作控件	时间
A	2018-03-20 10:00:00
/	2018-03-20 10:00:03
0	2018-03-20 10:00:07
1	2018-03-20 11:26:07
4	2018-03-20 11:26:08
2	2018-03-20 11:26:12

OSDTK信息

```
{  
  "container": "*****/***/orp",  
  "crash_reason": "SIGSEGV",  
  "crash_time": "20180320100010",  
  "cuid": "A319ACDEC0067CDB5",  
  "soft_version": "10.5.0",  
  "device": "vivo Y55L",  
  "crash_module": "libGLES",  
  "crash_stack": "*****"  
}
```

! 高危链路：A -> B -> C
! 高危控件：0

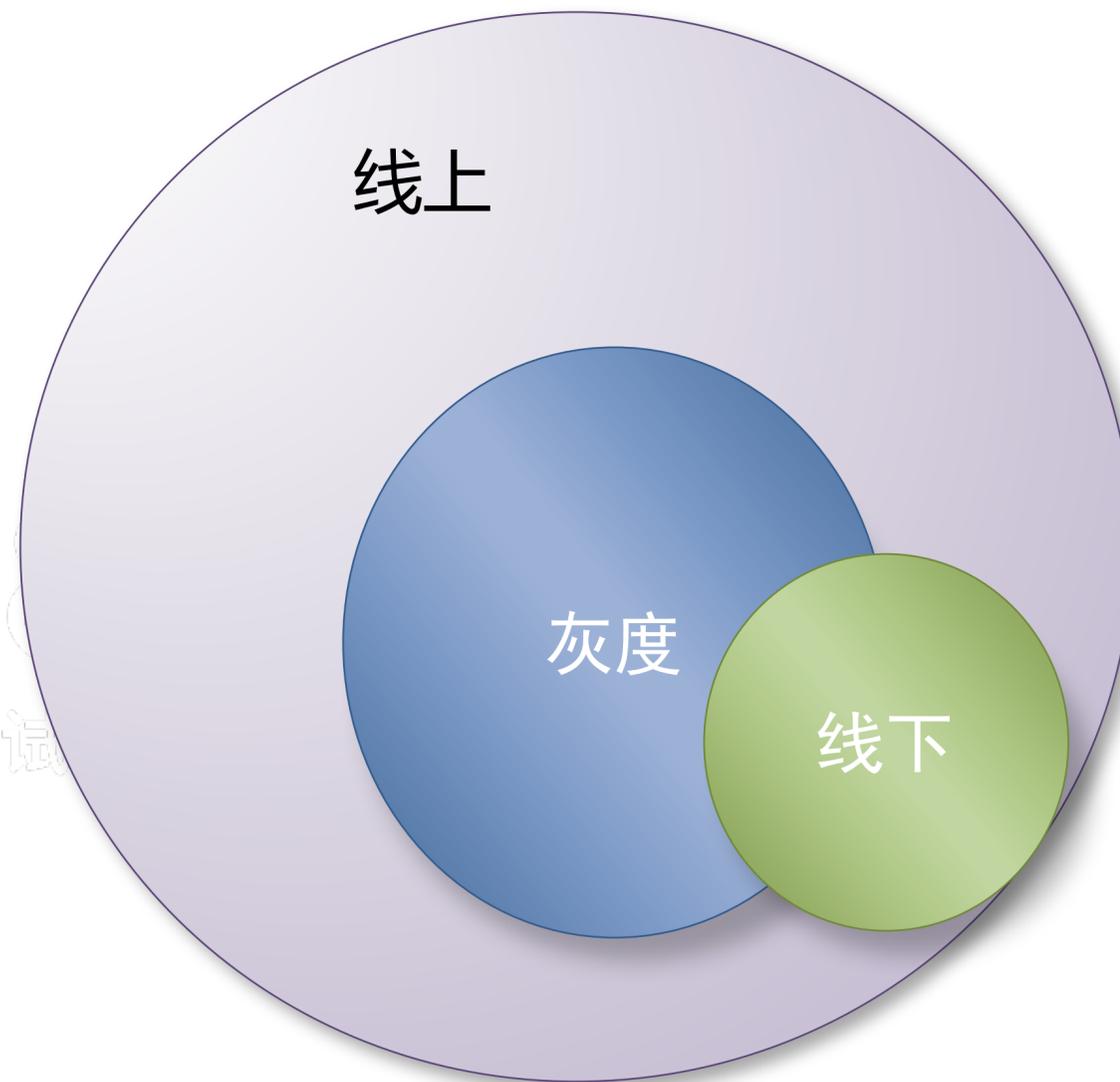




传统线下测试覆盖度衡量方式：

- 白盒覆盖（语句，判定，条件，路径覆盖等等）
- 灰盒覆盖（接口，函数覆盖等等）
- 黑盒覆盖（功能，需求覆盖等等）

MTSC2018
第四届中国移动互联网测试



灰度覆盖率



已覆盖控件集合

全量控件集合



质量
数据
平台

- ! 数据吞吐：100T+/天
- ! 数据存取QPS α 20W+/S



质量
数据
地图

- ! 200+数据元
- ! 800T+质量相关数据沉淀
- ! 数据检索150W次/天



质量
改进
应用

- ! 基于数据的质量改进落地子项70+



谢谢大家



MTSC2018

第四届中国移动互联网测试开发大会

TesterHome

TesterHome

IT大咖说