

# 蘑菇街 App 的性能与稳定性实践

蘑菇街 - 银时

# 原则

能发现



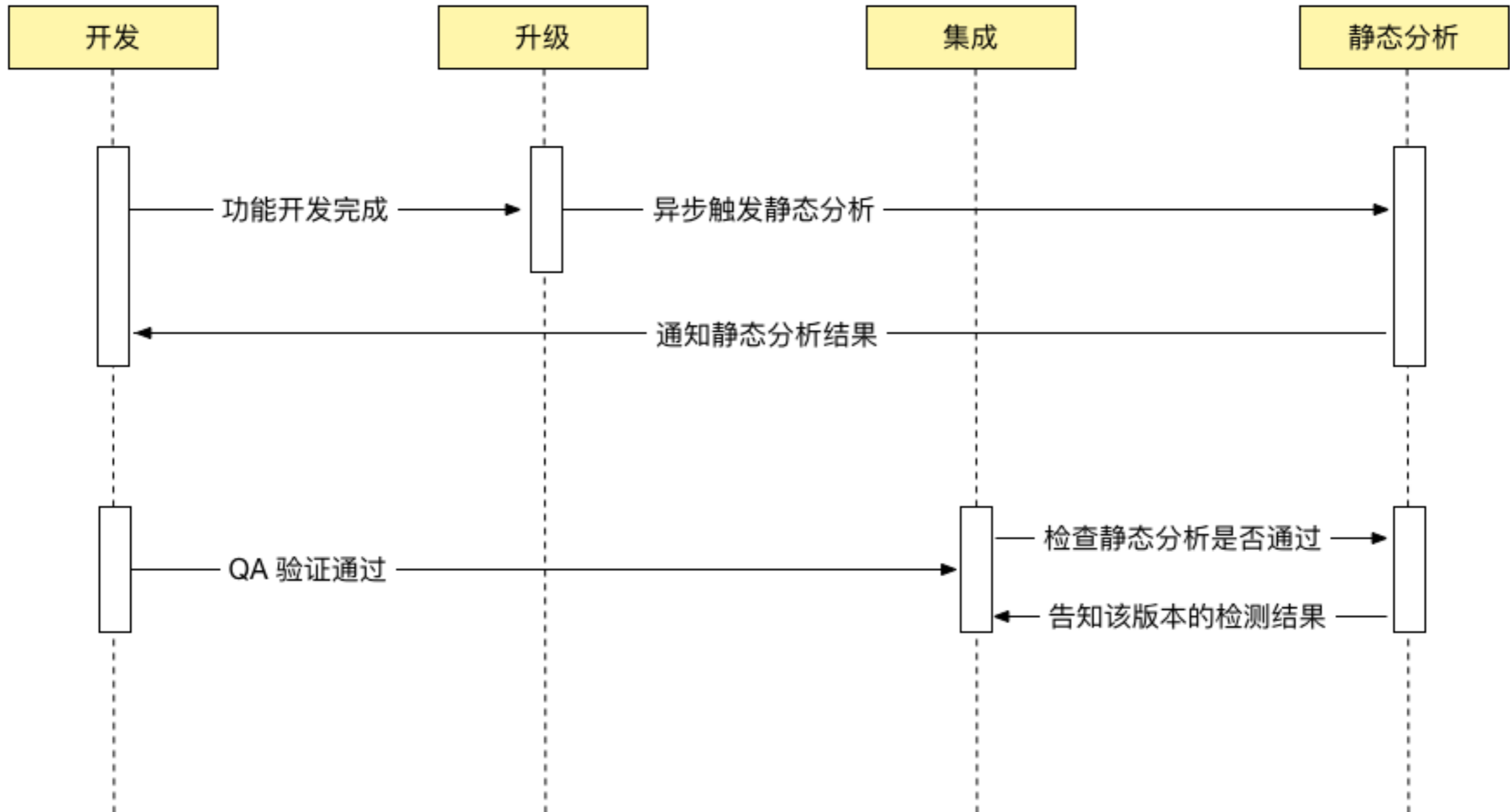
早发现，早治疗



监控体系

# 稳定性

线下



# 稳定性

线下

静态分析结果页

mogujie-iphone ▾

[release/v8.1.1](#)  
[release/v8.1.0](#)  
[develop](#)

组件名	负责人	错误数	版本号	操作
WU/WebViewController		0	0.1.16	<a href="#">查看详情</a>
WU/RequestManagerFactory		0	0.1.13	<a href="#">查看详情</a>
WU/Request		0	0.2.11	<a href="#">查看详情</a>
WU/URLResponse		0	0.0.3	<a href="#">查看详情</a>
WU/URLRequest		0	0.1.13	<a href="#">查看详情</a>
WU/URLSession		0	1.2.1	<a href="#">查看详情</a>
WU/WebViewFactory		0	0.0.9	<a href="#">查看详情</a>
WU/WebView		0	0.0.7	<a href="#">查看详情</a>
WU/WebView		0	0.0.13	<a href="#">查看详情</a>
WU/WebViewFactory		0	0.1.1	<a href="#">查看详情</a>
WU/WebView		0	0.1.3	<a href="#">查看详情</a>
WU/WebView		0	0.0.58	<a href="#">查看详情</a>

静态分析结果页

1.2.10 ▾

file	code	type
WU/RequestManagerFactory.java:76	METHOD:MWPCContext Call.syncContext() ERROR:object returned by mwpContext.getRequest() could be null and is dereferenced at line 76	NULL_DEREFERENCE
WU/RequestManagerFactory.java:13	METHOD:Context ApplicationGetter.getContext() ERROR:Context class android.content.Context may leak during method com.mogujie.mwpsdk.util.ApplicationGetter.getContext: Static field com.mogujie.mwpsdk.util.ApplicationGetter.context l-> Leaked class android.content.Context	CONTEXT_LEAK
WU/RequestManagerFactory.java:20	METHOD:void ApplicationGetter.setContext(Context) ERROR:Context class android.content.Context may leak during method com.mogujie.mwpsdk.util.ApplicationGetter.setContext: Static field com.mogujie.mwpsdk.util.ApplicationGetter.context l-> Leaked class android.content.Context	CONTEXT_LEAK
WU/ResponseInterceptor.java:50	METHOD:void MGJResponseInterceptor\$MGJResponseValve.invoke(PipelineContext) ERROR:object payload last assigned on line 49 could be null and is dereferenced at line 50	NULL_DEREFERENCE
WU/ResponseInterceptor.java:53	METHOD:void MGJResponseInterceptor\$MGJResponseValve.invoke(PipelineContext) ERROR:object payload last assigned on line 52 could be null and is dereferenced at line 53	NULL_DEREFERENCE

# 稳定性

线上

## 监控

- 报警
- Crash 平台

之前

写死阈值，超过报警

- 3 分钟 Crash 超过 N 次
- HTTP 请求成功率低于 N%

## 修复

- 安全模式
- Hotfix

现在

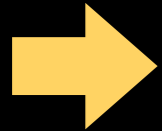
与前一天对比，结合标准差  
支持降级到内部 IM  
专人轮值

# 稳定性

线上

## 监控

- 报警
- Crash 平台



## 修复

- 安全模式
- Hotfix

### Crash日报

#### 昨日Android 数据概览

类目	Crash	Catch
发生次数	1000	500
影响用户数	500	250
3次以上用户数	100	50
Crash率	0.1%	0.05%

#### 昨日Crash状态更新

Jira地址	版本	Crash地址	责任人	状态	备注
--------	----	---------	-----	----	----

#### 昨日iOS 数据概览

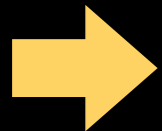
类目	Crash	Catch
----	-------	-------

# 稳定性

线上

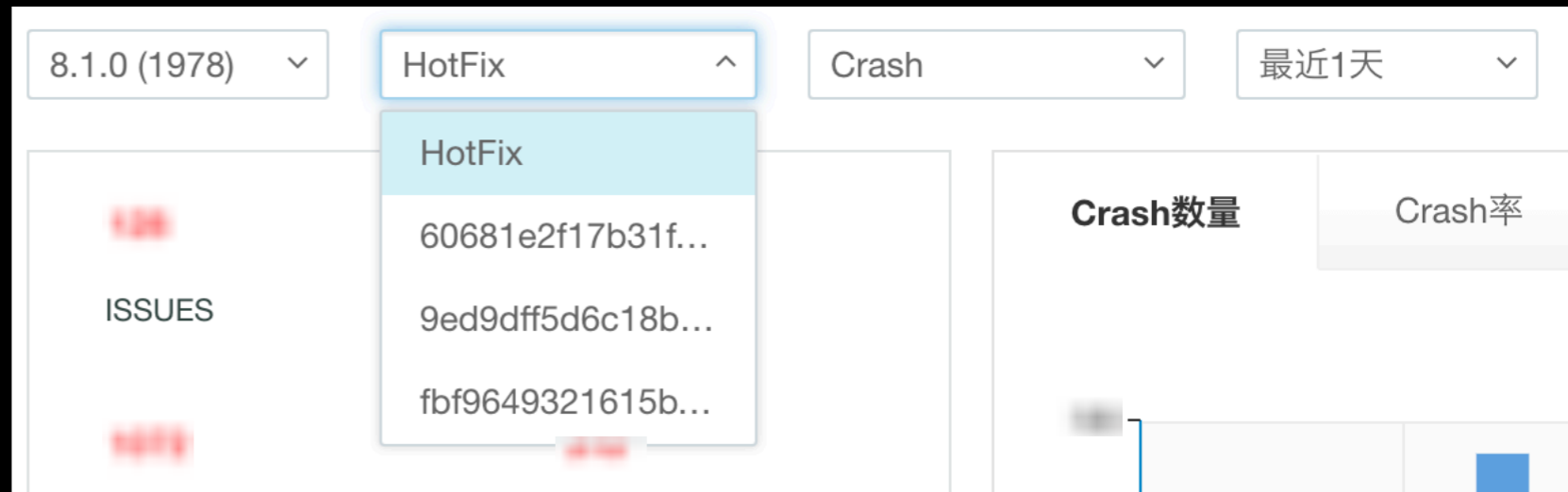
## 监控

- 报警
- Crash 平台



## 修复

- 安全模式
- Hotfix



### Hotfix生效分析

所有版本 2016-10-08

2016-10-07: 所有版本总人数为 1,200,000

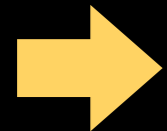
Hash	下载人数
af6464d33202d9789dba03bdfc2284ba	
2e82edb3795c1f1480323d80ed618739	

# 稳定性

线上

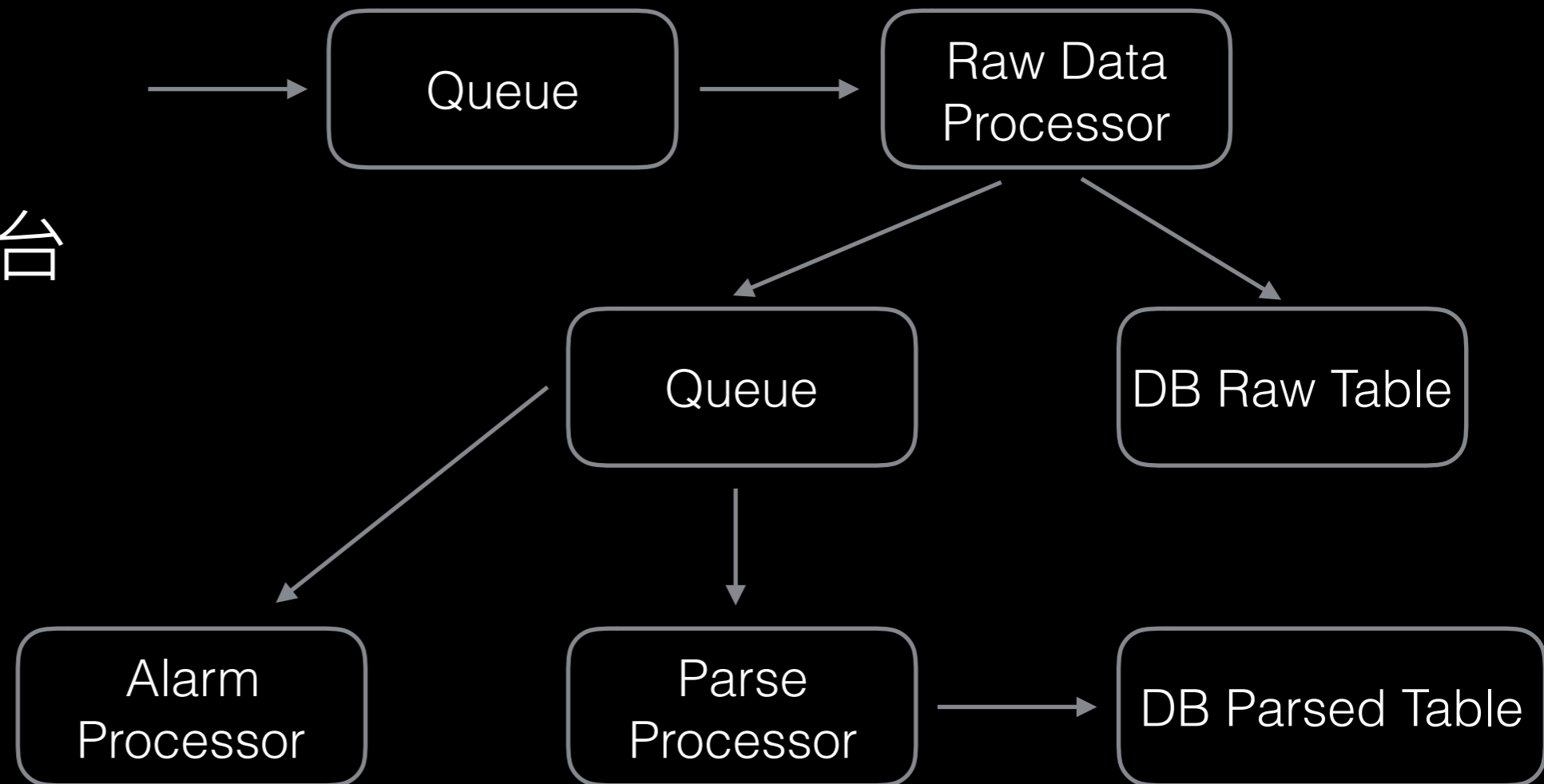
## 监控

- 报警
- Crash 平台



## 修复

- 安全模式
- Hotfix





# 稳定性

线上

## 监控

- 报警
- Crash 平台

## 修复

- ➔ • 安全模式
- Hotfix

短时间内连续 3 次 Crash



清除沙箱文件，恢复到初始状态

# 稳定性

线上

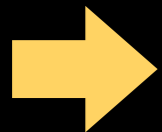
## 监控

- 报警
- Crash 平台

hack classloader / JSPatch

## 修复

- 安全模式
- Hotfix



# 性能

## 体验



# 哪些场景用户会觉得不爽



闪退

打开慢

滑动不流畅

耗电

网络不畅/出错

图片加载慢/出不来

流量大

# 细化为指标

FPS

CPU

卡顿

冷启动时间

内存泄漏

Exception

图片请求

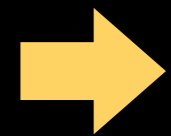
网络请求

# 如何获得这些性能指标

## FPS

```
func handleDisplayLink() {  
    var deltaTime = currentTimestamp - self.lastTimestamp  
    var currentFPS = 1 / deltaTime  
    var currentURL = MGJAnalytics.sharedInstance().currentURL  
    self.fpsData[currentURL].append(currentFPS)  
    self.lastTimestamp = currentTimestamp  
}
```

# 做的一些优化



打点库

线程治理

内存泄漏治理

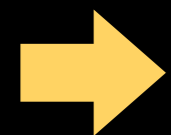
过度绘制治理

图片域名收敛

1. 写文件 -> 写数据库
2. 先写后发再删 -> 先发再写后删
3. 更精细的后端控制

# 做的一些优化

打点库



线程治理

内存泄漏治理

过度绘制治理

图片域名收敛

120 -> 80



# 做的一些优化

60+

打点库

线程治理

➔ 内存泄漏治理

过度绘制治理

图片域名收敛

- 只在灰度开启
- 对 Act 添加弱引用，在 destroy 时做标记，每隔 2 分钟 GC，每隔 5 秒钟检查，5 分钟后不该活着的还活着，标记为泄漏
- 结合 UV，来判断泄漏

# 做的一些优化

一个点被绘制多次

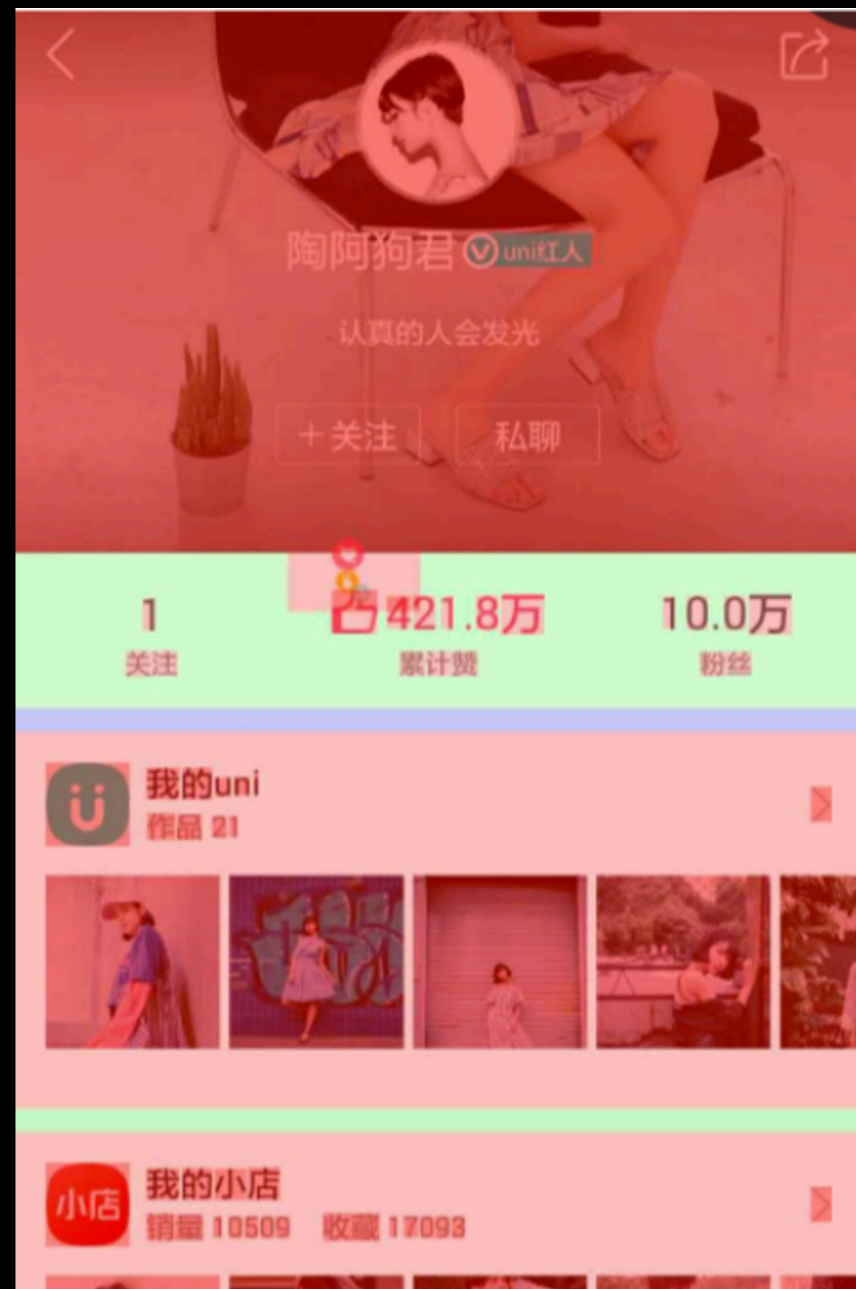
打点库

线程治理

内存泄漏治理

➔ 过度绘制治理

图片域名收敛



# 做的一些优化

打点库

线程治理

内存泄漏治理

过度绘制治理

 图片域名收敛

单域名的情况不要使用 HTTP/1.1

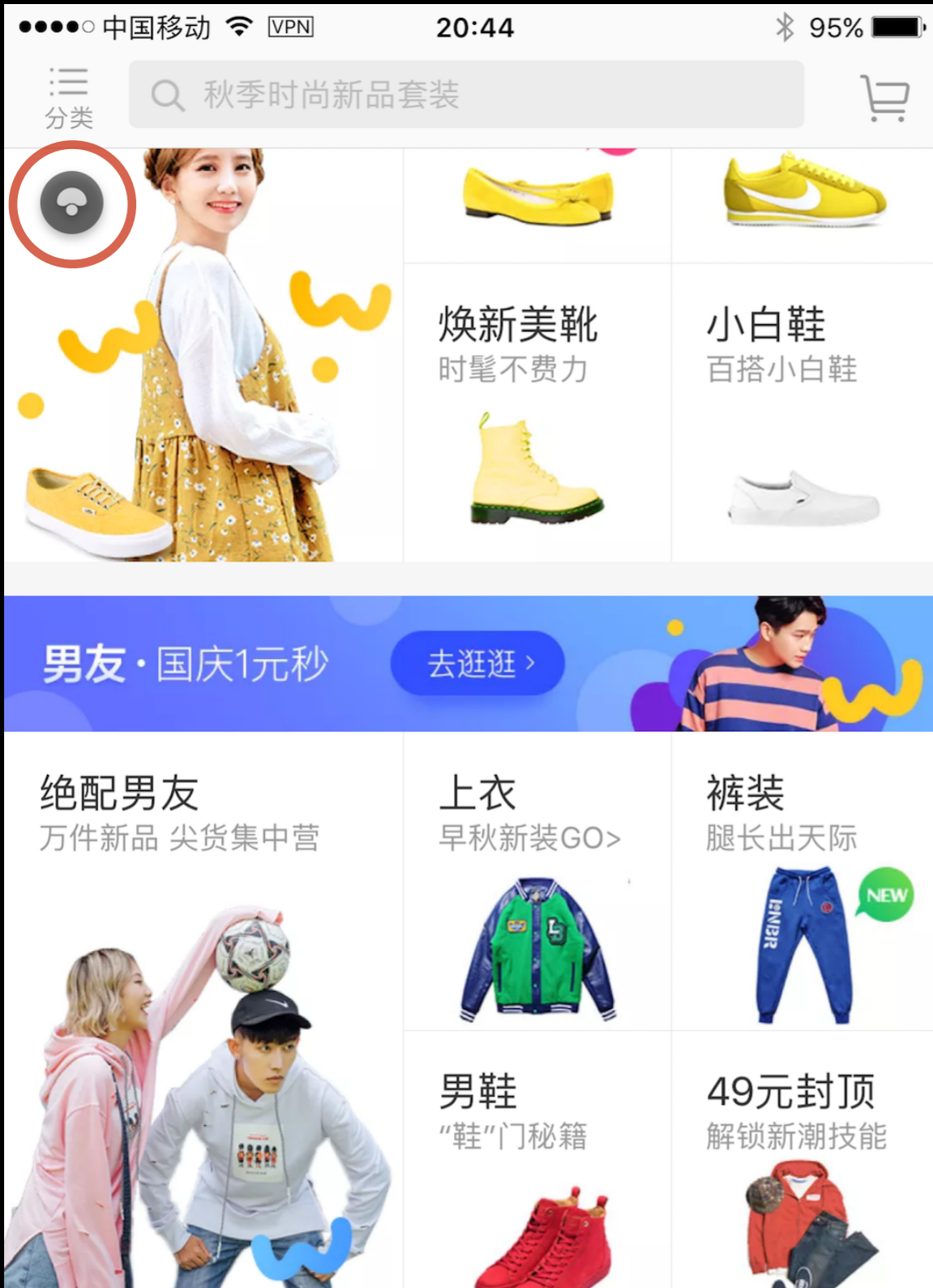
# AppMate

## 问题

不方便知道开发阶段的性能状况  
没有人对开发阶段的性能把关

阿里百川·小蘑菇

# AppMate



# AppMate

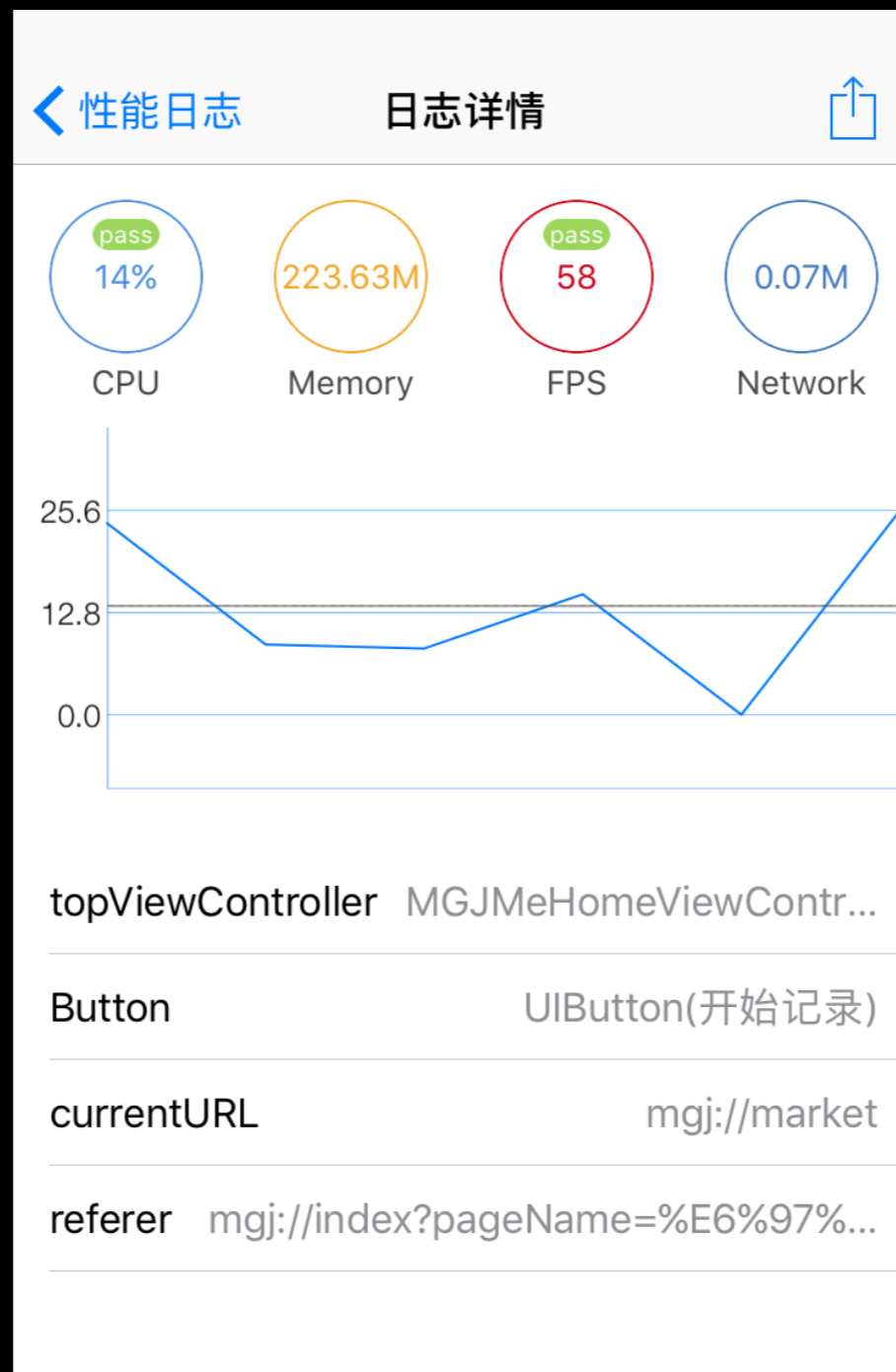


- 图片大小超过限制
- 网络异常/错误
- 捕获到的异常
- ...

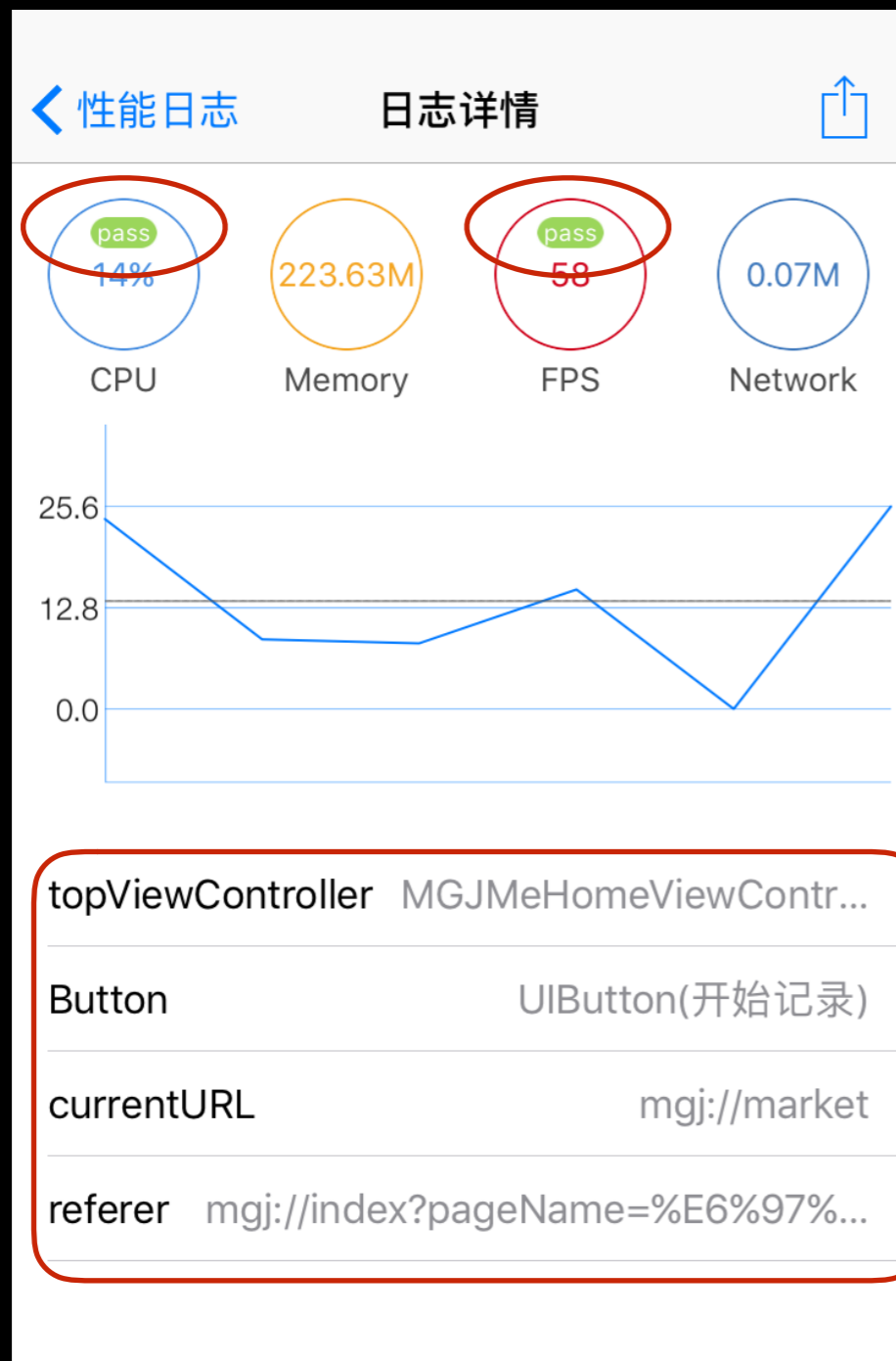


MGJLogWarning

# AppMate



# AppMate



- 对特定指标进行测试，并保存数据
- 对特定指标显示是否通过
- 每个节点都可以展示相关信息
- 数据可以导出

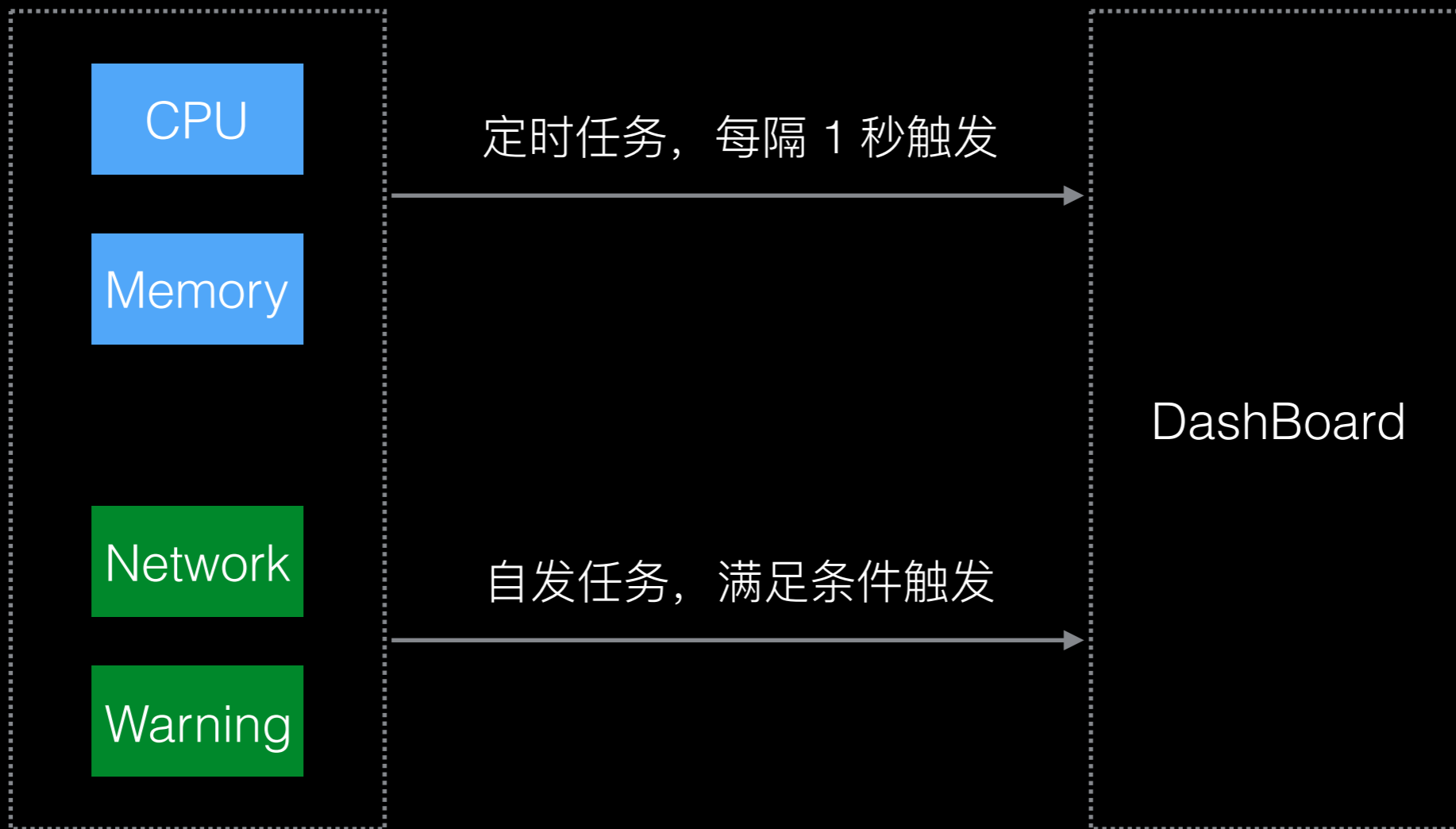


# AppMate

LogProvider

LogItem

LogConsumer



# AppMate

开发：每次打开 App 都要重新开启，好麻烦

# AppMate



# AppMate

QA: 测业务忙不过来, 数据看起来也不方便

# AppMate

陷入了思考



# AppMate

如果每次打开 App 都自动记录性能数据  
并且自动发送数据到 Server  
并且能够在浏览器查看实时数据  
并且能够查看历史性能数据

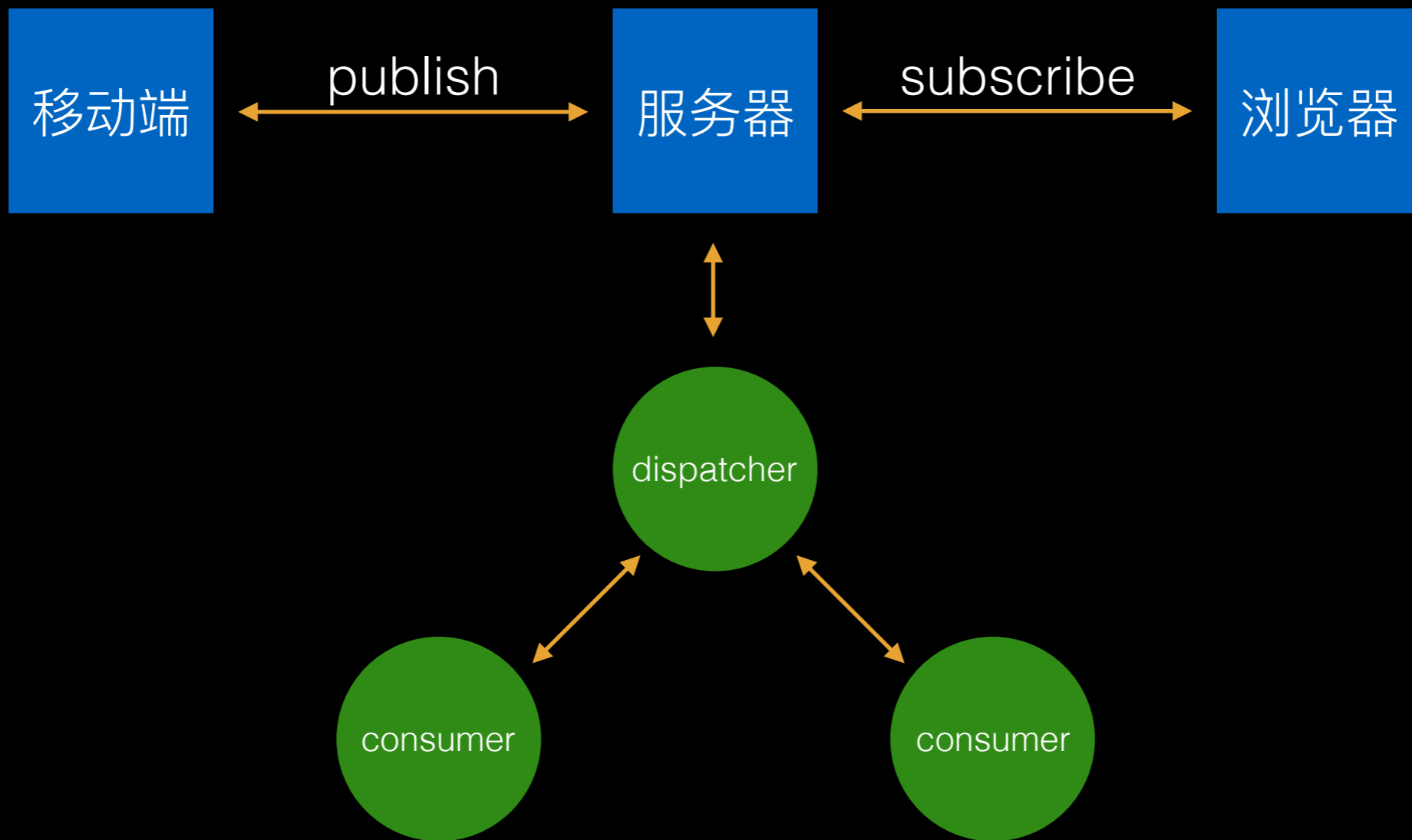
# AppMate

## AppRoad

在移动端和浏览器之间修建一条路，让双方可以实时互通

实验中...

# AppMate





# AppMate

```
{
  "publish": {
    "topic": "performance.cpu:0f890831fa6",
    "payload": {
      "value": 10.08,
      "common": {
        "device": "iPhone6,1",
        "system": "10.0.2"
      }
    }
  },
  "subscribe": {
    "topic": "performance.cpu:0f890831fa6"
  },
  "unsubscribe": {
    "topic": "performance.cpu:0f890831fa6"
  }
}
```

# AppMate

AppMate

## Sessions

Filter Device ID



2016-10-16

查找

Device ID	Platform	Start Time	Duration	Active
0F9A31-DC1568-A10972-081356	iOS	10/16 14:08	185 sec	<span style="color: green;">●</span>
0f8297836cd0823	Android	10/16 14:06	102 sec	<span style="color: green;">●</span>
B39A31-0F1078-A10972-081356	iOS	10/16 11:05	237 sec	<span style="color: gray;">●</span>
0F9A31-05801B-29087D-081356	iOS	10/16 14:08	73 sec	<span style="color: gray;">●</span>
0890387bcd90fa8	Android	10/16 10:21	803 sec	<span style="color: gray;">●</span>
7073519057d6c0a	Android	10/16 10:08	31 sec	<span style="color: gray;">●</span>
B39A31-0F1078-A10972-081356	iOS	10/16 10:04	237 sec	<span style="color: gray;">●</span>
0F9A31-05801B-29087D-081356	iOS	10/16 09:43	73 sec	<span style="color: gray;">●</span>

# AppMate

## AppMate

Performance

Overview

CPU

Memory

FPS

Network

Leaks

Cold Start

Warnings

Exceptions

Image too Large

Image Load Slow

Network Error

概览



指标	数值	基准值
CPU	10%	8%
Memory	170 MB	150 MB
FPS	59	55
Network	10.43 MB	N/A
Leaks	1	0
Cold Start	4.39	2
Warnings	3	0

# AppMate

AppMate

Performance

Overview

CPU

Memory

FPS

Network

Leaks

Cold Start

Warnings

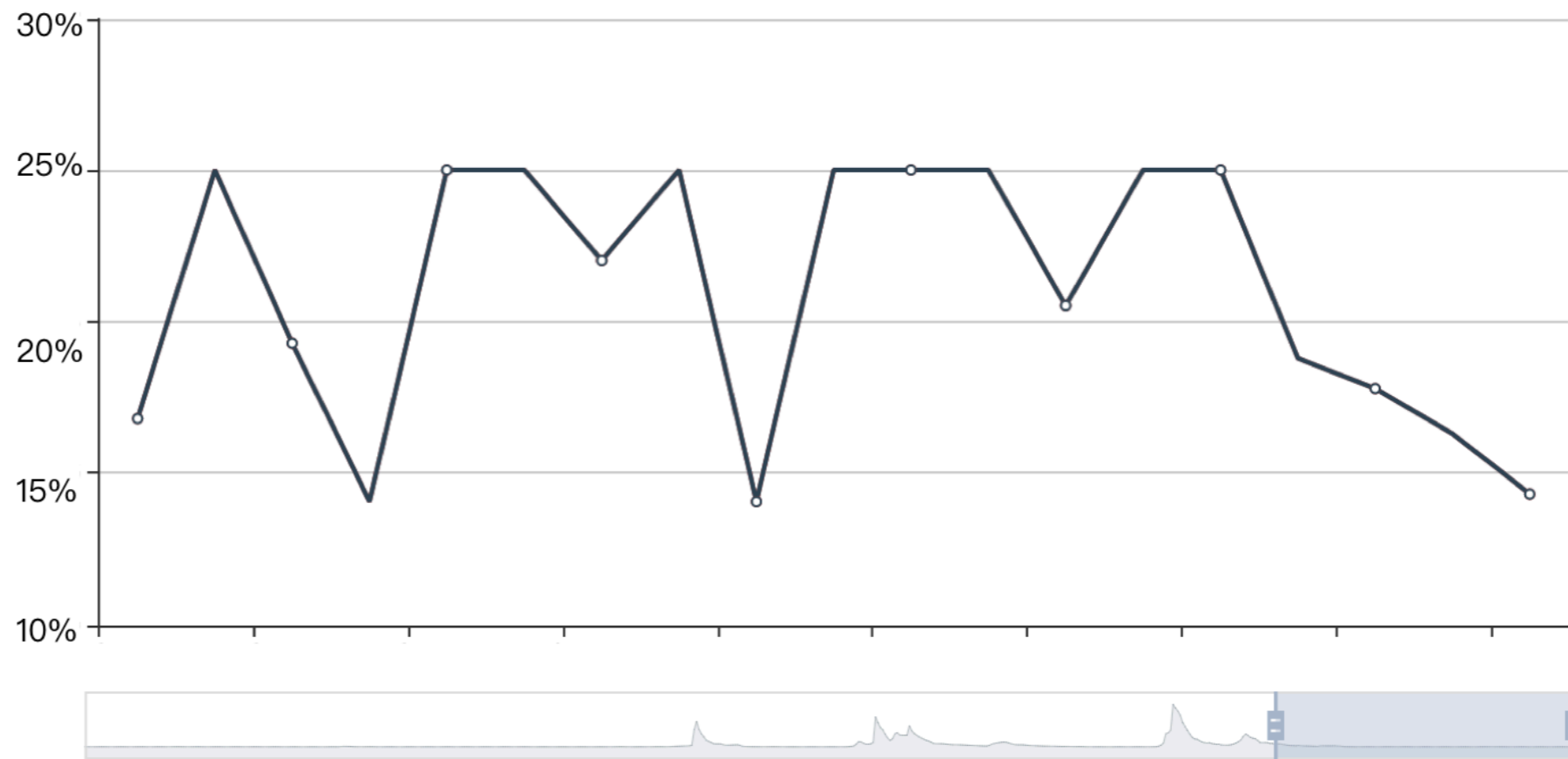
Exceptions

Image too Large

Image Load Slow

Network Error

CPU



时间

数值

附加属性

10:09:05

18%

page: mgj://index referer: mgj://init topVC: MGJIndexViewCon...

# AppMate

还可以做什么？

浏览器成为了手机的扩展终端

- 在浏览器里查看/过滤日志
- 在浏览器里对数据进行格式化输出
- ...

通过浏览器向手机发送指令

- 获取/修改 配置中心的值
- 获取/修改 NSUserDefaults 的值
- 获取/修改 沙箱内容
- ...

# 异常设备体系

## 问题

只有用户反馈时才知道发生了什么问题，被动  
不知道线上异常用户的量  
缺少对「异常」的定义

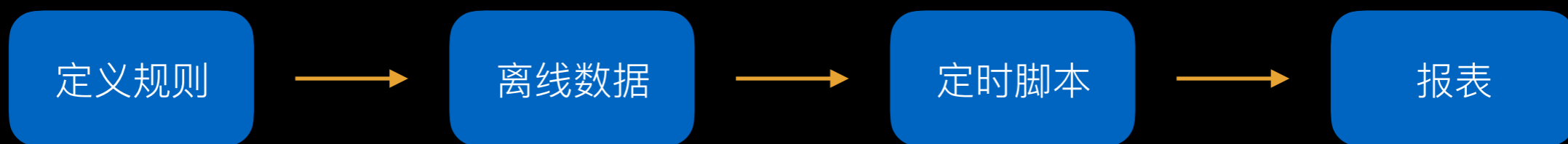
# 异常设备体系

解决

根据异常模型筛选数据

# 异常设备体系

## 实现





# 异常设备体系

## 实现

1	Crash超过2次		%
2	前台crash超过0次		%
3	网络请求失败超过5次		%
4	接口业务逻辑错误超过5次		%
5	启动时间[最快]超过2s		%
6	启动时间[最慢]超过5s		%
7	启动时间[平均]超过4s		%
8	流量累积[3G/4G]超过100M		%
9	流量累积[Wifi]超过300M		%
10	图片大小超过阈值10次		%
11	图片下载失败超过10次		%
12	图片下载慢(大于1秒)超过0次		%
13	图片下载慢(2G)超过0次		%
14	图片下载慢(3G)超过0次		%
15	图片下载慢(4G)超过0次		%
16	图片下载慢(WIFI)超过0次		%
17	图片下载慢(大于1秒)超过5次		%
18	图片下载慢(2G)超过5次		%
19	图片下载慢(3G)超过5次		%
20	图片下载慢(4G)超过5次		%
21	图片下载慢(WIFI)超过5次		%
22	开启安全模式		%
23	总设备数		

网络请求失败超过5次( )	862305036026326	失败	次	Mogujie Android	8.1.1.2249
	861980030669270	失败	次	Mogujie Android	8.1.1.2249
	861084038998151	失败	次	Mogujie Android	8.1.1.2249
	861374030097403	失败	次	Mogujie Android	8.1.1.2249
	99000848666889	失败	次	Mogujie Android	8.1.1.2249
	861374032509330	失败	次	Mogujie Android	8.1.1.2249
	99000830349543	失败	次	Mogujie Android	8.1.1.2249
	869271026883915	失败	次	Mogujie Android	8.1.1.2249
	99000694808006	失败	次	Mogujie Android	8.1.1.2249
	99000856904658	失败	次	Mogujie Android	8.1.1.2249
	861735031314833	失败	次	Mogujie Android	8.1.1.2249
	861084035156514	失败	欠	Mogujie Android	8.1.1.2249
	861374034861465	失败	欠	Mogujie Android	8.1.1.2249
	99000830155398	失败	欠	Mogujie Android	8.1.1.2249

# 异常设备体系

## 问题

不直观

不方便进行历史数据对比

不方便快速发现 / 定位问题

# 异常设备体系

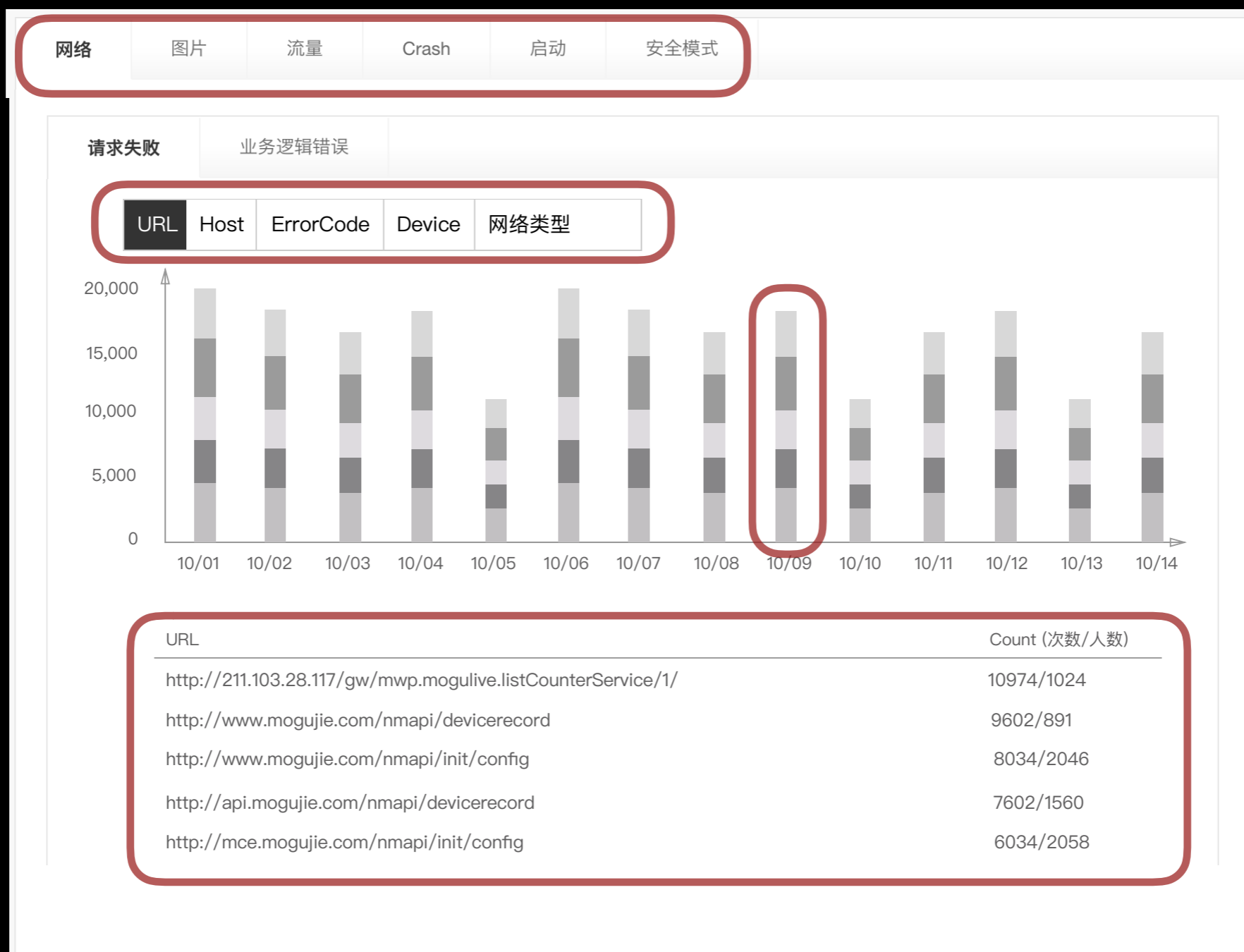
解决

图表化，落地到 Web 端

进行中...

# 异常设备体系

## 解决



# 定向分析

## 问题

知道了某个人 / 设备有异常，怎么办？

# 定向分析

实时:  用户名  自动刷新: 开(每秒)

## 🔑 关键指标 (今日)

HTTP请求成功率	HTTP响应耗时	Socket连接成功率	Socket连接耗时	Socket数据包耗时	Crash次数
100%	309ms	100%	254ms	213ms	0次

全部	页面路径	<b>HTTP请求</b>	Socket请求	用户操作	进出页面	Crash/Exception	<a href="#">刷新</a>
----	------	---------------	----------	------	------	-----------------	--------------------

输入搜索信息, 按回车

日期	类型	请求返回码	业务返回码	响应时间	操作
2016-10-09 10:01:47	N	200	1001	257	<a href="#">详情</a>
2016-10-09 10:01:47	N	200	-1	252	<a href="#">详情</a>
2016-10-09 10:01:47	N	200	1001	218	<a href="#">详情</a>
2016-10-09 10:01:47	N	200	1001	268	<a href="#">详情</a>
2016-10-09 10:01:47	N	200	1001	255	<a href="#">详情</a>
2016-10-09 10:01:47	N	200	1001	212	<a href="#">详情</a>
2016-10-09 10:01:47	N	200	1001	686	<a href="#">详情</a>

# 定向分析

## 特点

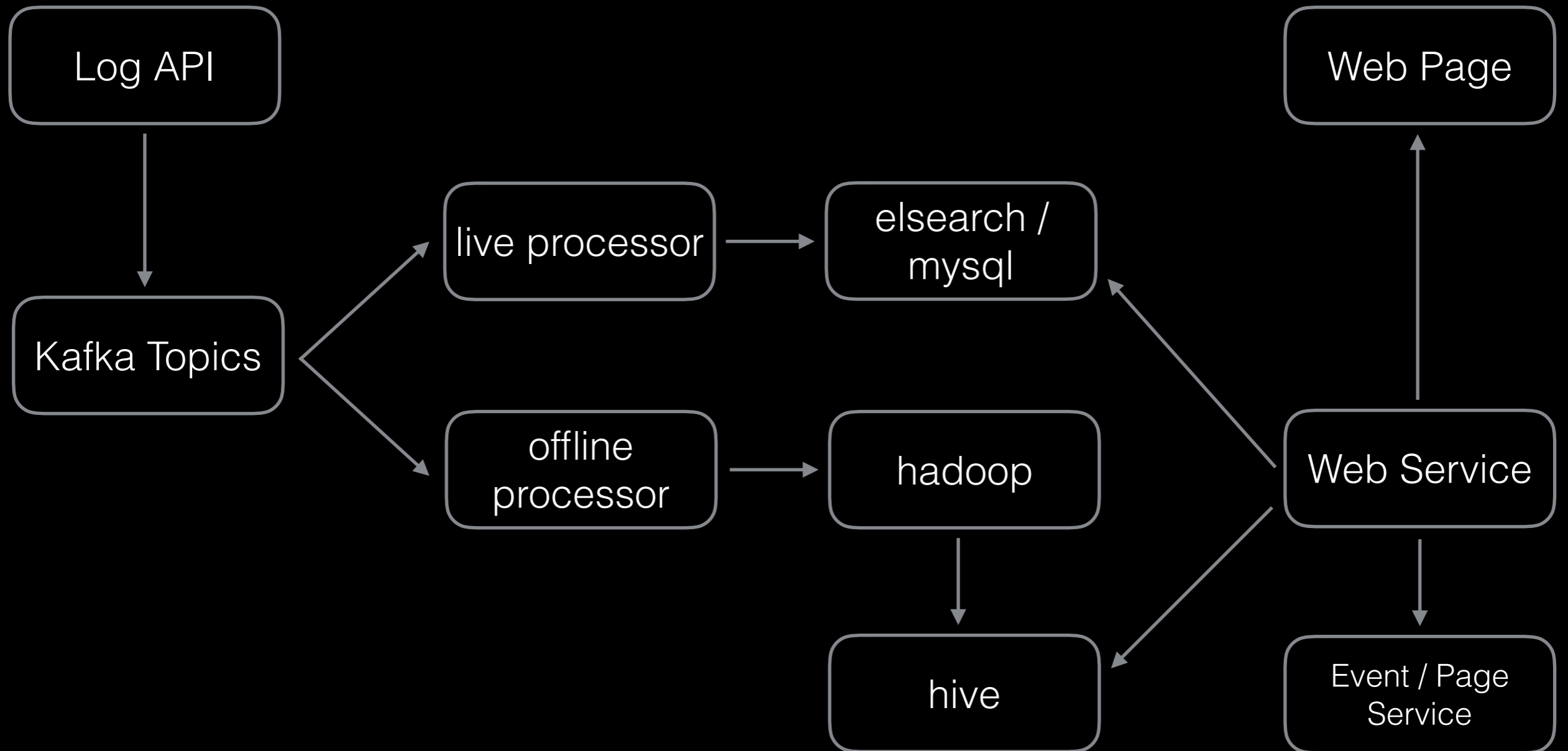
使用方便

对小仙小侠号默认实时

可以结合多个维度来看问题

可以结合事件的名字来更直观地了解

# 定向分析





# What's Next

拿到更多 / 更准的性能数据

建立更加完善的线上性能监控系统（可视化）

更加有针对性的优化

自动化性能检测与提单系统

Thanks

Q & A