如何构建 App 基础体验保障体系

一套小思维框架的使用

蘑菇街 - 银时 负责蘑菇街 App 基础体验

三个事

介绍一套小思维框架

该思维框架在 App 基础体验保障中的使用

我们在基础保障体验中做过的以及要做的事

三个事

介绍一套小思维框架

该思维框架在 App 基础体验保障中的使用

我们在基础保障体验中做过的以及要做的事

SolarCity

Power forever.

不可再生能源储量有限，依目前的使用情况将在可预见的时间内被耗尽

很长一段时间内不再需要担心能源短缺

提升可持续能源转化率，成为大众生活的主要能源

在火星生活

大众主要设备支持可持续能源作为能量输入源

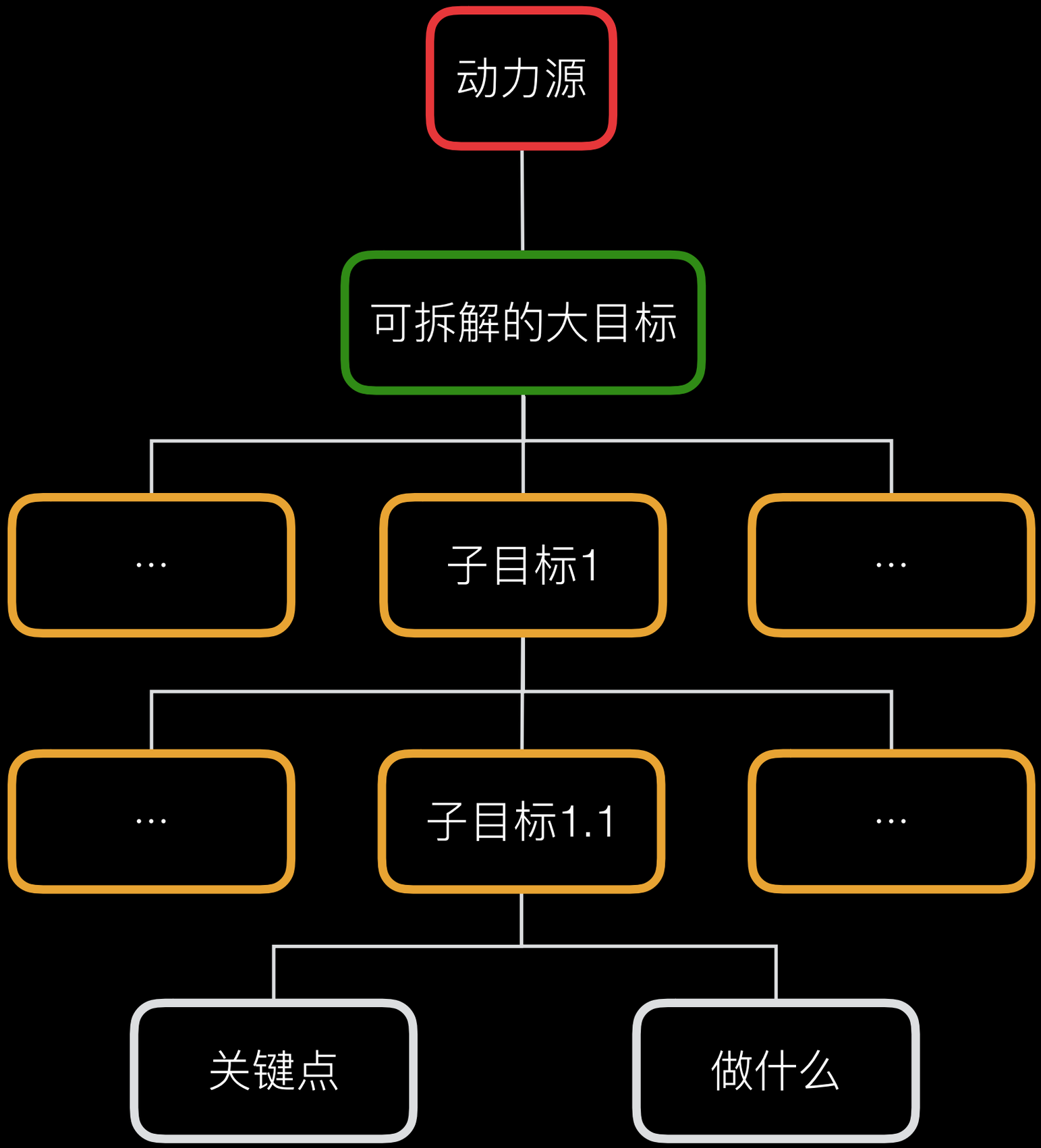
…

可回收的火箭

…

着陆稳定
机身完好无损
失败后不会殃及无辜

多次减速软着陆方案


动力源
可拆解的大目标
…
子目标1
…
…
子目标1.1
…
关键点
做什么

小测试

我要开一家水果店

我要加薪

小思维框架的好处

对于如何达成目标心里有谱
确保没有重要的部分被遗漏
知道当前的整体状态
知道目前在做些什么
知道弱项和一下阶段的方向

讲三个事

介绍一套小思维框架

该思维框架在 App 基础体验保障中的使用

我们在基础保障体验中做过的以及要做的事

该思维框架在 App 基础体验保障中的使用

动力源

该思维框架在 App 基础体验保障中的使用

互联网红利没了
用户对 App 的热情下降
只剩增加用户使用 App 的时间一条路

该思维框架在 App 基础体验保障中的使用

互联网红利没了
用户对 App 的热情下降
只剩增加用户使用 App 的时间一条路

可拆解的大目标

该思维框架在 App 基础体验保障中的使用

互联网红利没了
用户对 App 的热情下降
只剩增加用户使用 App 的时间一条路

给线上用户提供优质的基础体验

该思维框架在 App 基础体验保障中的使用

互联网红利没了
用户对 App 的热情下降
只剩增加用户使用 App 的时间一条路

给线上用户提供优质的基础体验

App 呈现在用户面前时，基础体验已得到足够的保证

一小部分线上用户先用起来，验证真实场景下的基础体验

正式使用过程中，发生基础体验问题时可以被快速发现和修复

该思维框架在 App 基础体验保障中的使用

互联网红利没了
用户对 App 的热情下降
只剩增加用户使用 App 的时间一条路

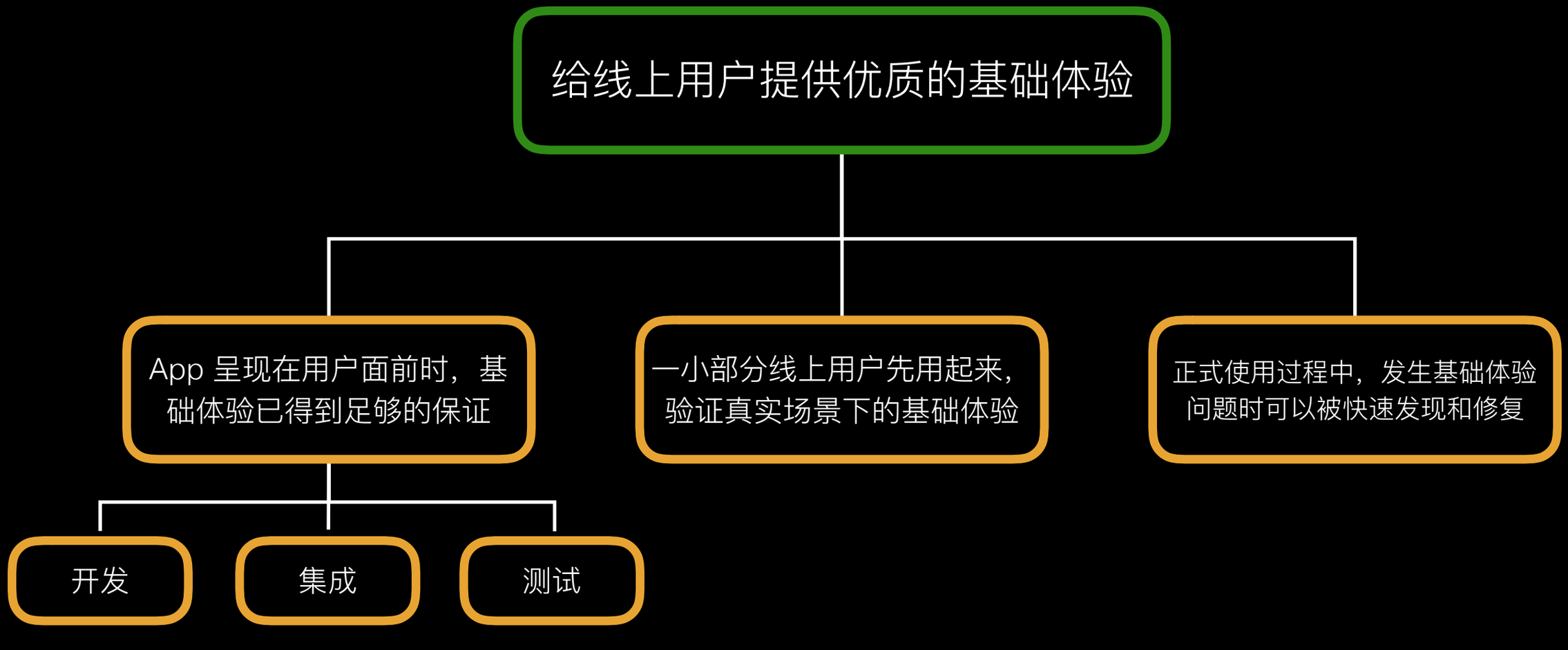
给线上用户提供优质的基础体验

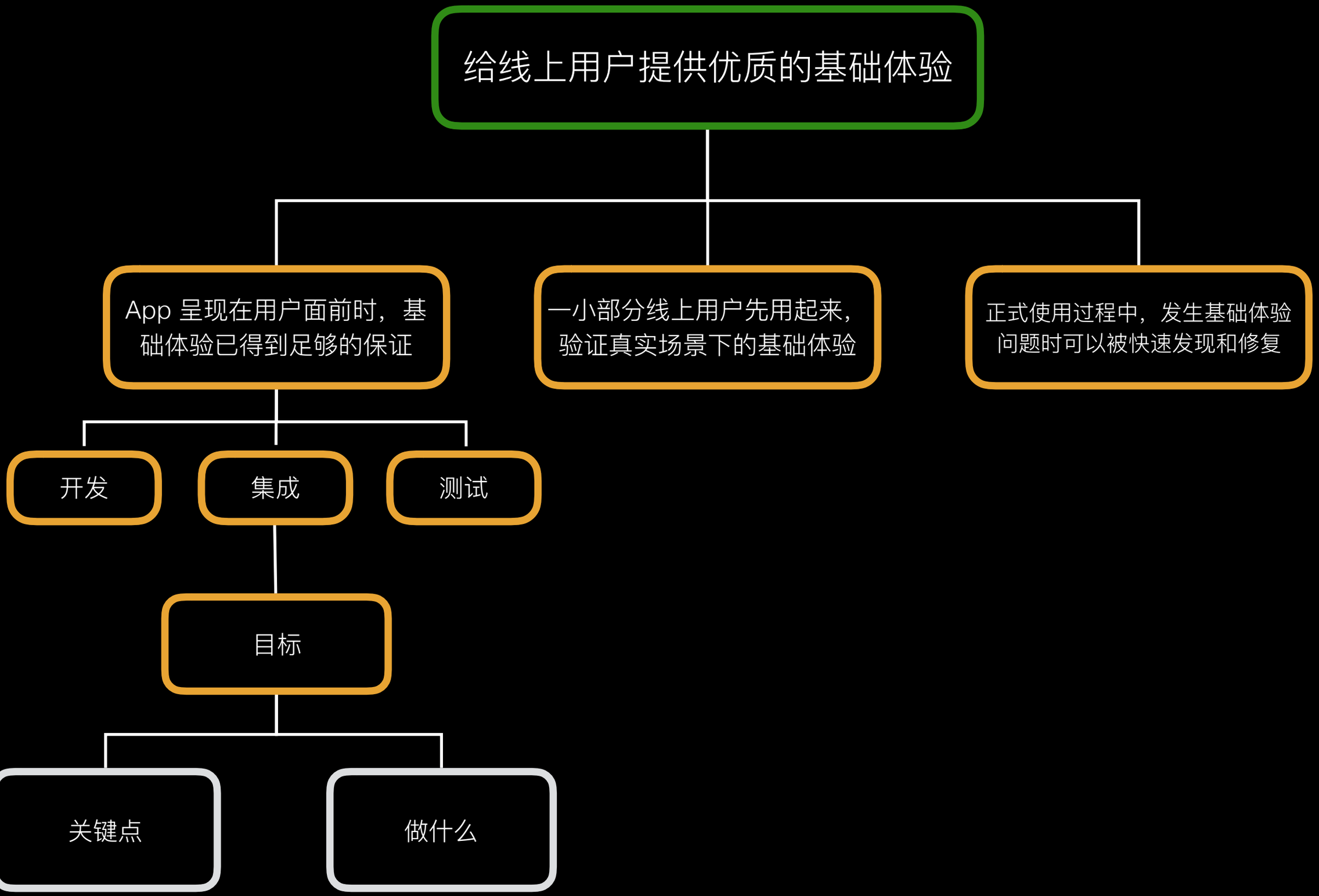
App 呈现在用户面前时，基础体验已得到足够的保证

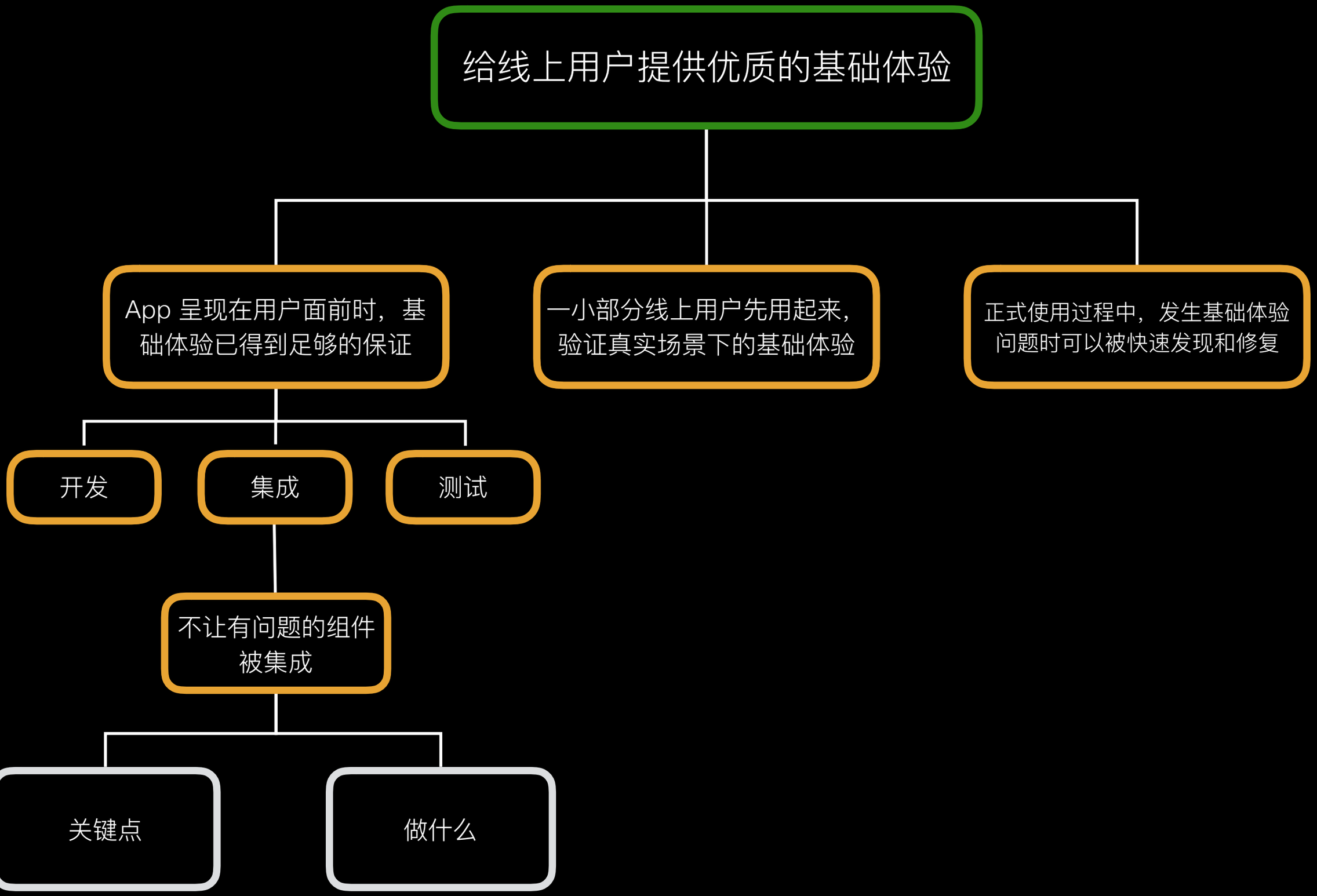
- 开发
- 集成
- 测试

一小部分线上用户先用起来，验证真实场景下的基础体验

正式使用过程中，发生基础体验问题时可以被快速发现和修复


给线上用户提供优质的基础体验
App 呈现在用户面前时，基础体验已得到足够的保证
一小部分线上用户先用起来，验证真实场景下的基础体验
正式使用过程中，发生基础体验问题时可以被快速发现和修复
开发
集成
测试


给线上用户提供优质的基础体验
App 呈现在用户面前时，基础体验已得到足够的保证
一小部分线上用户先用起来，验证真实场景下的基础体验
正式使用过程中，发生基础体验问题时可以被快速发现和修复
开发
集成
测试
目标
关键点
做什么


给线上用户提供优质的基础体验
App 呈现在用户面前时，基础体验已得到足够的保证
一小部分线上用户先用起来，验证真实场景下的基础体验
正式使用过程中，发生基础体验问题时可以被快速发现和修复
开发
集成
测试
不让有问题的组件被集成
关键点
做什么

给线上用户提供优质的基础体验

App 呈现在用户面前时，基础体验已得到足够的保证

一小部分线上用户先用起来，验证真实场景下的基础体验

正式使用过程中，发生基础体验问题时可以被快速发现和修复

开发

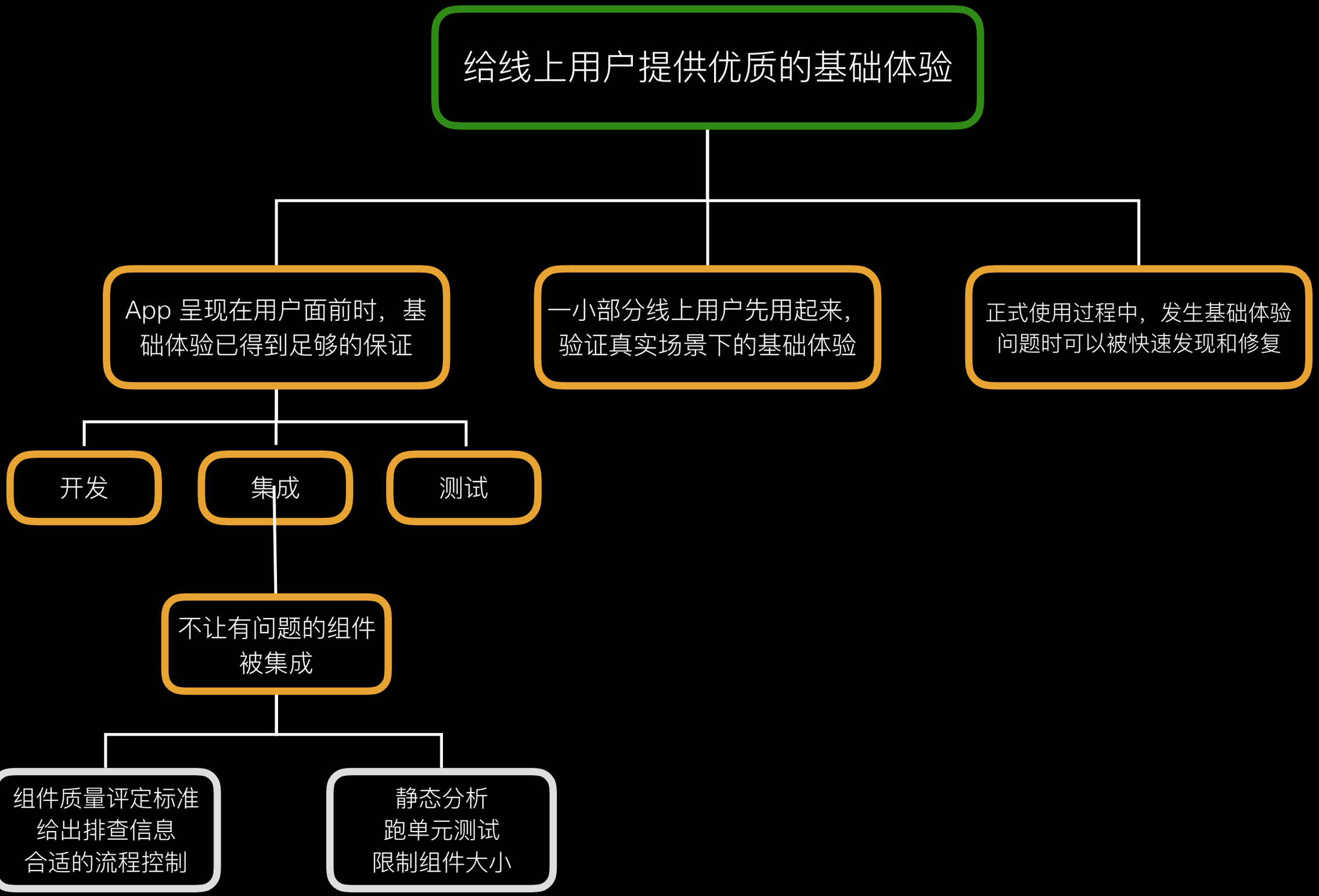
集成

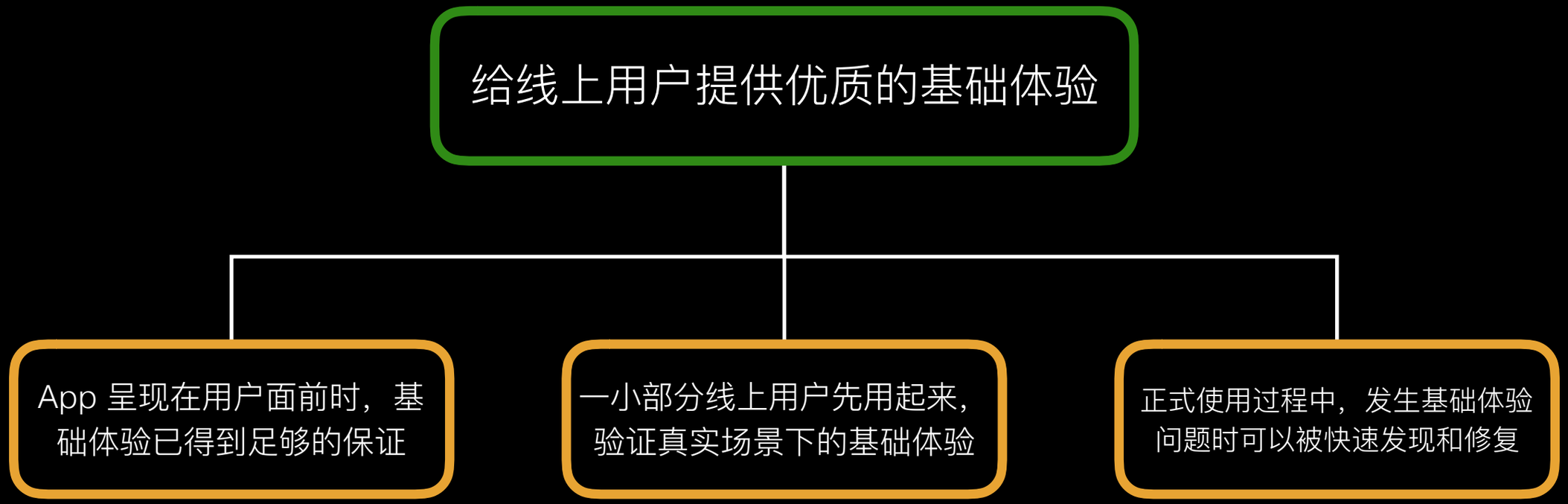
测试

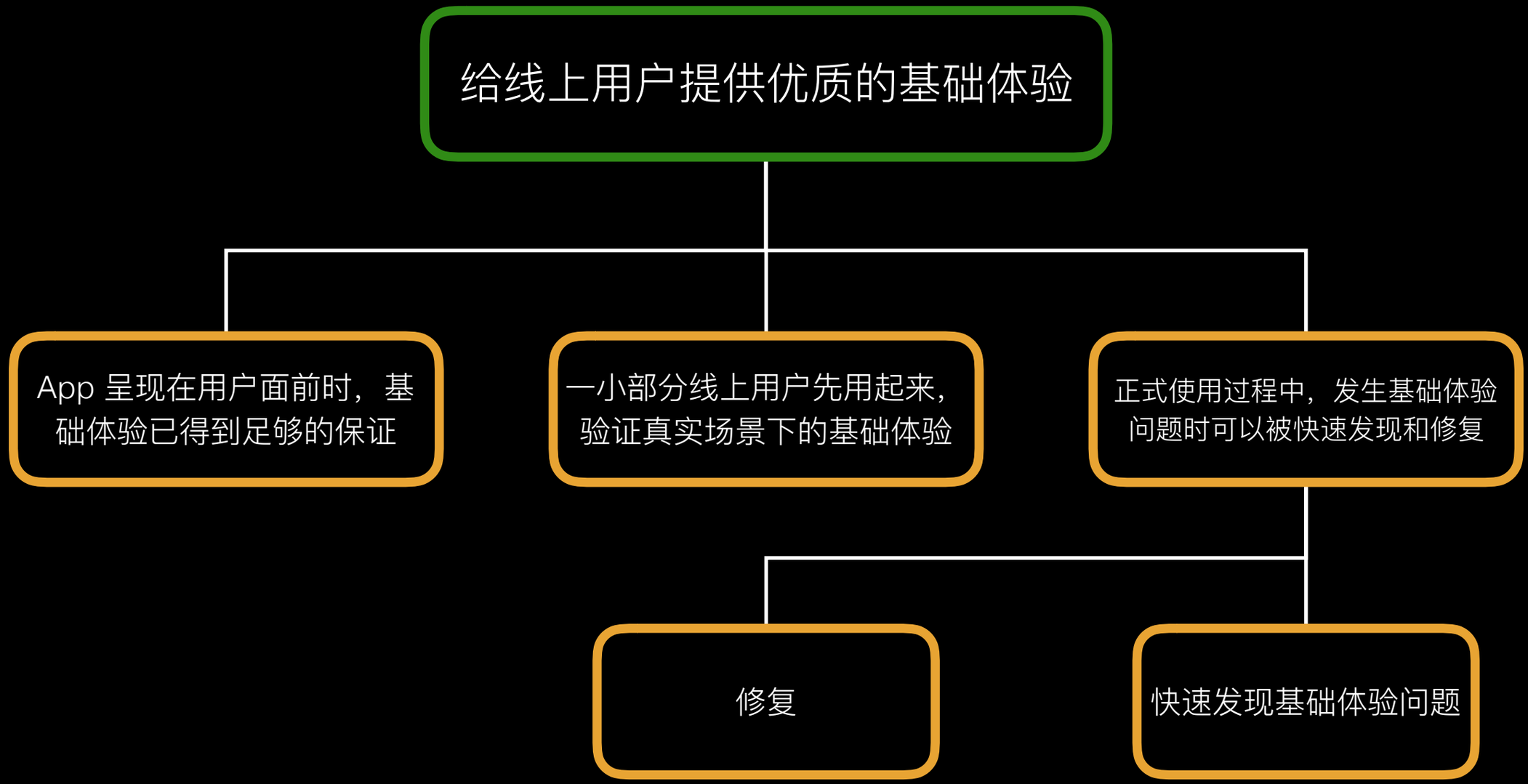
不让有问题的组件被集成

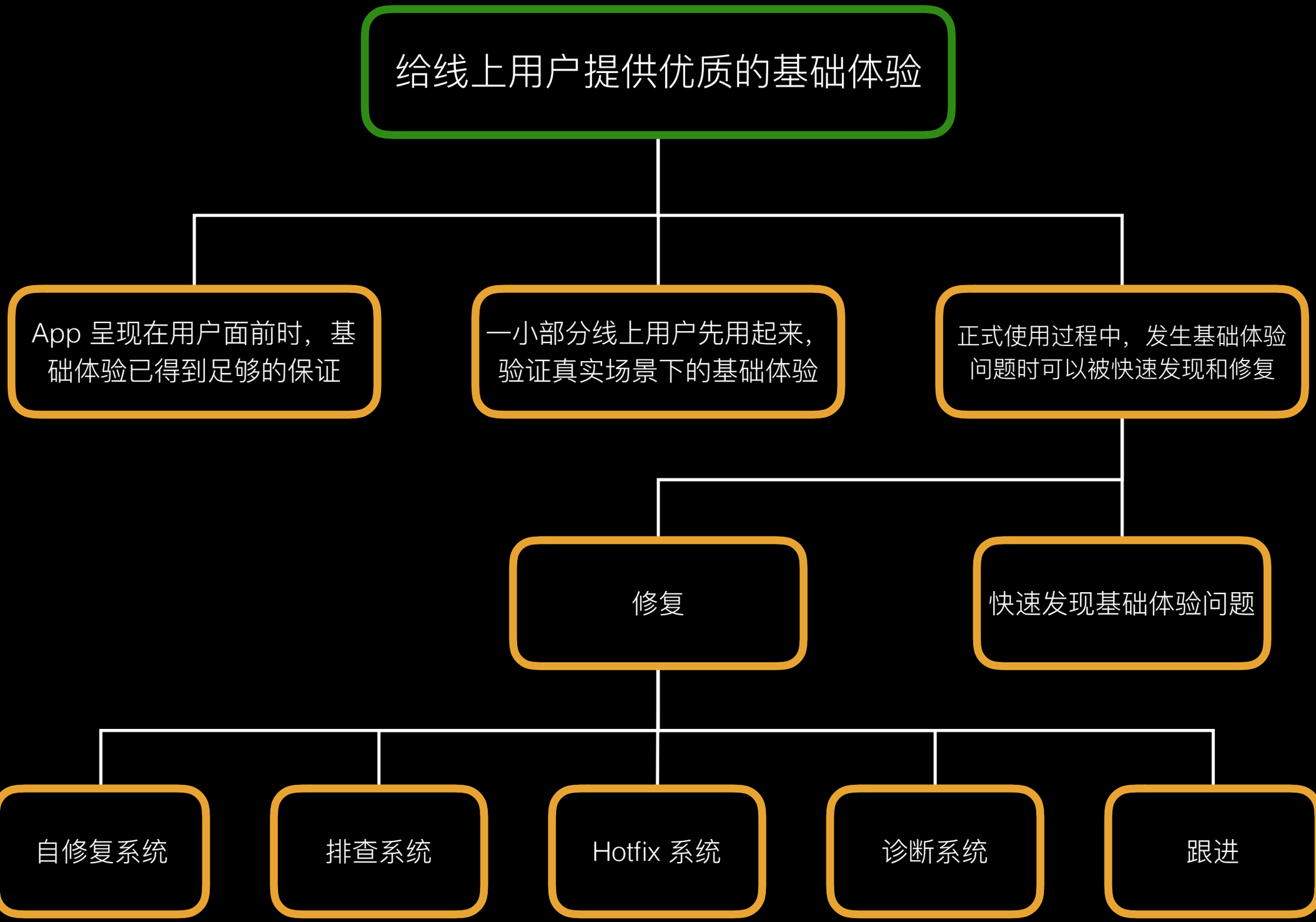
组件质量评定标准
给出排查信息
合适的流程控制

做什么


给线上用户提供优质的基础体验
App 呈现在用户面前时，基础体验已得到足够的保证
一小部分线上用户先用起来，验证真实场景下的基础体验
正式使用过程中，发生基础体验问题时可以被快速发现和修复
开发
集成
测试
不让有问题的组件被集成
组件质量评定标准
给出排查信息
合适的流程控制
静态分析
跑单元测试
限制组件大小


给线上用户提供优质的基础体验
App 呈现在用户面前时，基础体验已得到足够的保证
一小部分线上用户先用起来，验证真实场景下的基础体验
正式使用过程中，发生基础体验问题时可以被快速发现和修复


给线上用户提供优质的基础体验
App 呈现在用户面前时，基础体验已得到足够的保证
一小部分线上用户先用起来，验证真实场景下的基础体验
正式使用过程中，发生基础体验问题时可以被快速发现和修复
修复
快速发现基础体验问题


给线上用户提供优质的基础体验
App 呈现在用户面前时，基础体验已得到足够的保证
一小部分线上用户先用起来，验证真实场景下的基础体验
正式使用过程中，发生基础体验问题时可以被快速发现和修复
修复
快速发现基础体验问题
自修复系统
排查系统
Hotfix 系统
诊断系统
跟进

App 呈现在用户面前时，基础体验已得到足够的保证

一小部分线上用户先用起来，验证真实场景下的基础体验

正式使用过程中，发生基础体验问题时可以被快速发现和修复

- 修复
  - 自修复系统
  - 排查系统
  - Hotfix 系统
  - 诊断系统
  - 跟进
- 快速发现基础体验问题

App 呈现在用户面前时，基础体验已得到足够的保证

一小部分线上用户先用起来，验证真实场景下的基础体验

正式使用过程中，发生基础体验问题时可以被快速发现和修复

- 修复
  - 自修复系统
  - 排查系统
    - 目标
      - 关键点
      - 做什么
  - Hotfix 系统
  - 诊断系统
  - 跟进
- 快速发现基础体验问题

App 呈现在用户面前时，基础体验已得到足够的保证

一小部分线上用户先用起来，验证真实场景下的基础体验

正式使用过程中，发生基础体验问题时可以被快速发现和修复

- 修复
  - 自修复系统
  - 排查系统
    - 提供相关信息帮助定位问题原因
      - 关键点
      - 做什么
  - Hotfix 系统
  - 诊断系统
  - 跟进
- 快速发现基础体验问题

App 呈现在用户面前时，基础体验已得到足够的保证

一小部分线上用户先用起来，验证真实场景下的基础体验

正式使用过程中，发生基础体验问题时可以被快速发现和修复

- 修复
  - 自修复系统
  - 排查系统
    - 提供相关信息帮助定位问题原因
      - 信息尽量全且相关
        使用体验好
      - 定向分析系统
        日志系统
  - Hotfix 系统
  - 诊断系统
  - 跟进
- 快速发现基础体验问题

几个问题

如何表示目前的状态?

如何知道目前在做什么?

如何知道进度?

如何知道各个子项目本周做了什么?

App 呈现在用户面前时，基础体验已得到足够的保证

一小部分线上用户先用起来，验证真实场景下的基础体验

正式使用过程中，发生基础体验问题时可以被快速发现和修复

- 修复
  - 自修复系统
  - 排查系统
    - 提供相关信息帮助定位问题原因
      - 信息尽量全且相关
      - 使用体验好
      - 定向分析系统
      - 日志系统
  - Hotfix 系统
- 快速发现基础体验问题
  - 诊断系统
  - 跟进

App 呈现在用户面前时，基础体验已得到足够的保证

一小部分线上用户先用起来，验证真实场景下的基础体验

正式使用过程中，发生基础体验问题时可以被快速发现和修复

修复

快速发现基础体验问题

自修复系统

排查系统

Hotfix 系统

诊断系统

跟进

提供相关信息帮助定位问题原因

信息尽量全且相关
使用体验好

定向分析系统
日志系统

App 呈现在用户面前时，基础体验已得到足够的保证

一小部分线上用户先用起来，验证真实场景下的基础体验

正式使用过程中，发生基础体验问题时可以被快速发现和修复

修复

快速发现基础体验问题

自修复系统

排查系统

Hotfix 系统

诊断系统

跟进

提供相关信息帮助定位问题原因

信息尽量全且相关
使用体验好

定向分析系统
日志系统

- iOS 捕获 OOM 信息

  Progress:75%

  难点

  - OOM 判定方案和实现
  - 获取可用信息方便定位问题原因
  - 搭建监控、排查平台，有专人跟进

  状态

  - 客户端 OOM 检测已实现，930 上线
  - 监控平台搭建完成，待正式上线后看效果

  参考

  - code.facebook.com/posts/1146930688654547/reducing-fooms-in-the-facebook-ios-app/

- 建立无侵入监测机制

  Progress:50%

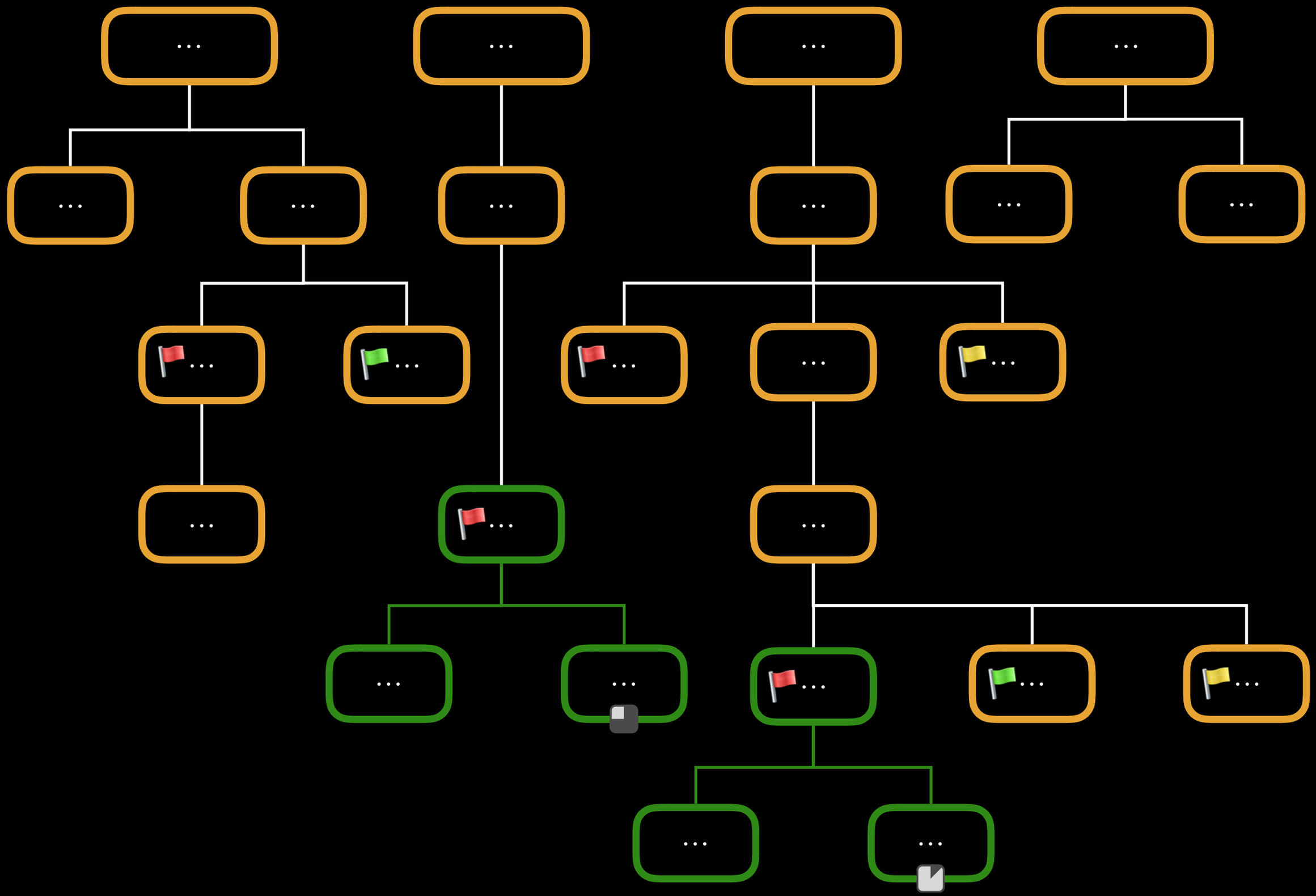
  难点

  - 保证数据的准确性
  - 尽量无侵入
  - 没有性能损耗

  状态

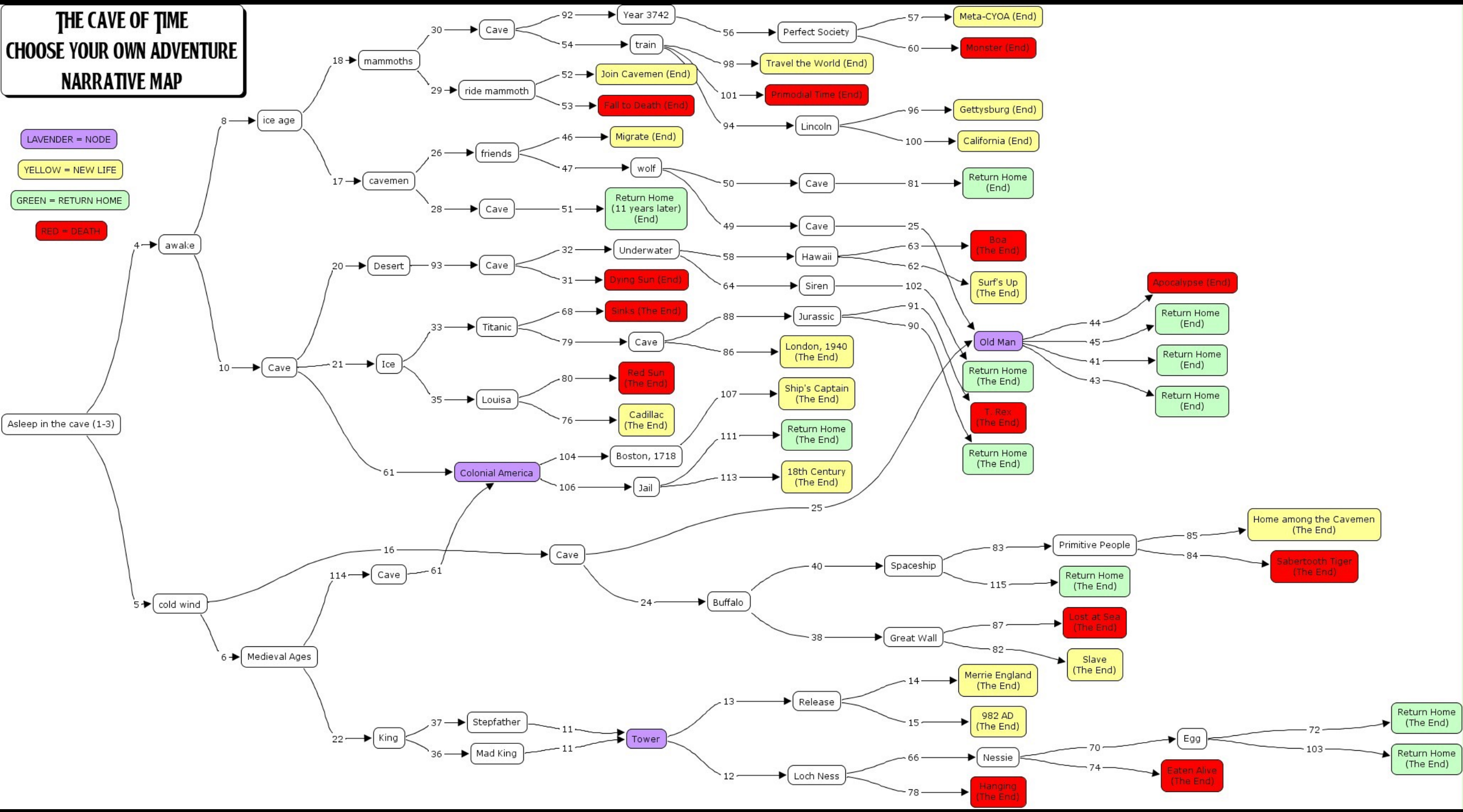
  - iOS 页面加载时间检测 ✅
  - Android 页面加载时间检测
  - iOS 启动耗时优化 ✅
  - Android 启动耗时优化 ✅
  - iOS 卡顿检测 ✅

结果就像这样

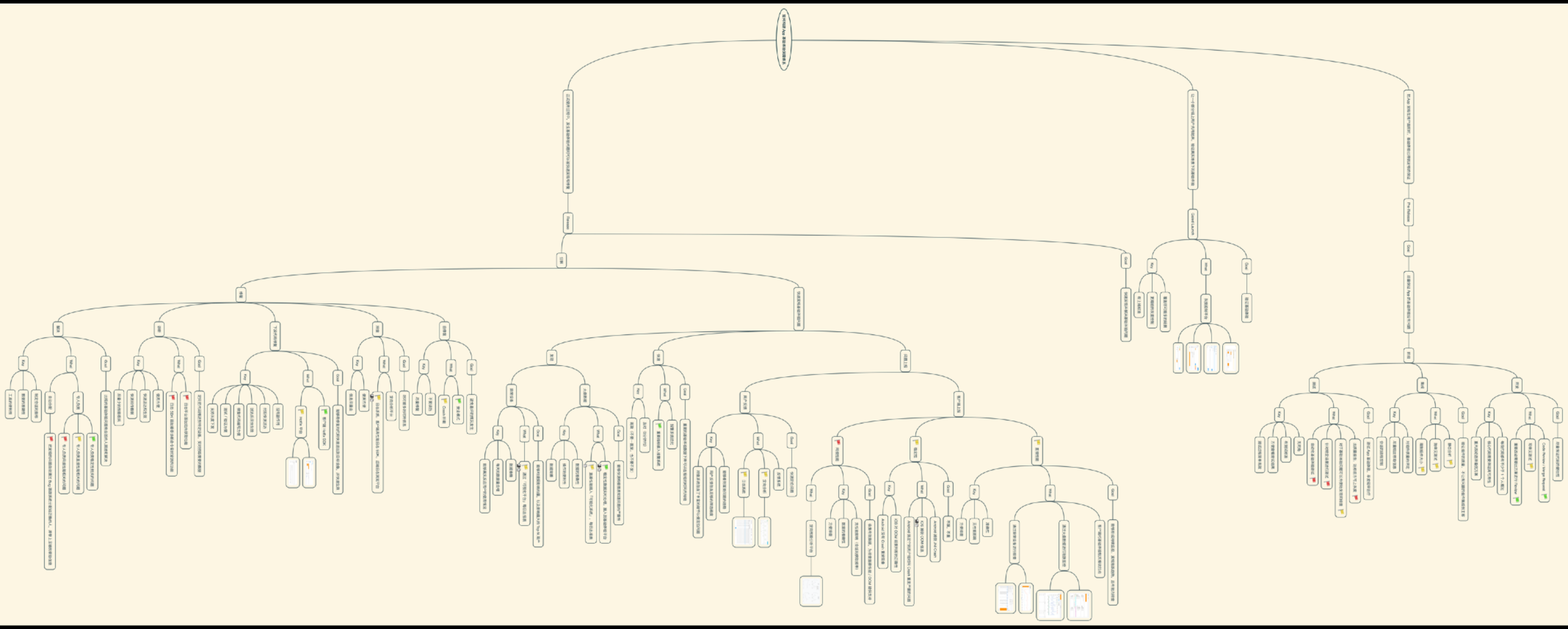


实际上它长这样

Just Kidding



实际上它长这样



讲三个事

介绍一套小思维框架

该思维框架在 App 基础体验保障中的使用

我们在基础保障体验中做过的以及要做的事

做过的事：异常设备体系

建立异常设备体系的目的

了解异常设备的量和异常程度

对问题设备进行针对性排查

通过点的问题来解决面的问题

异常设备体系监控维度

稳定性

Crash

OOM

安全模式

性能

网络

图片

启动

异常设备报表

iOS OOM

报表等级: B0　负责人:dubai　编辑报表

日期	2017-02-10	--	2017-03-09		
设备id	属于	多个值用^隔开			
设备名称	属于	多个值用^隔开			
错误次数	大于				
平台	属于	多个值用^隔开			
app版本	属于	多个值用^隔开			
系统版本	属于	多个值用^隔开			

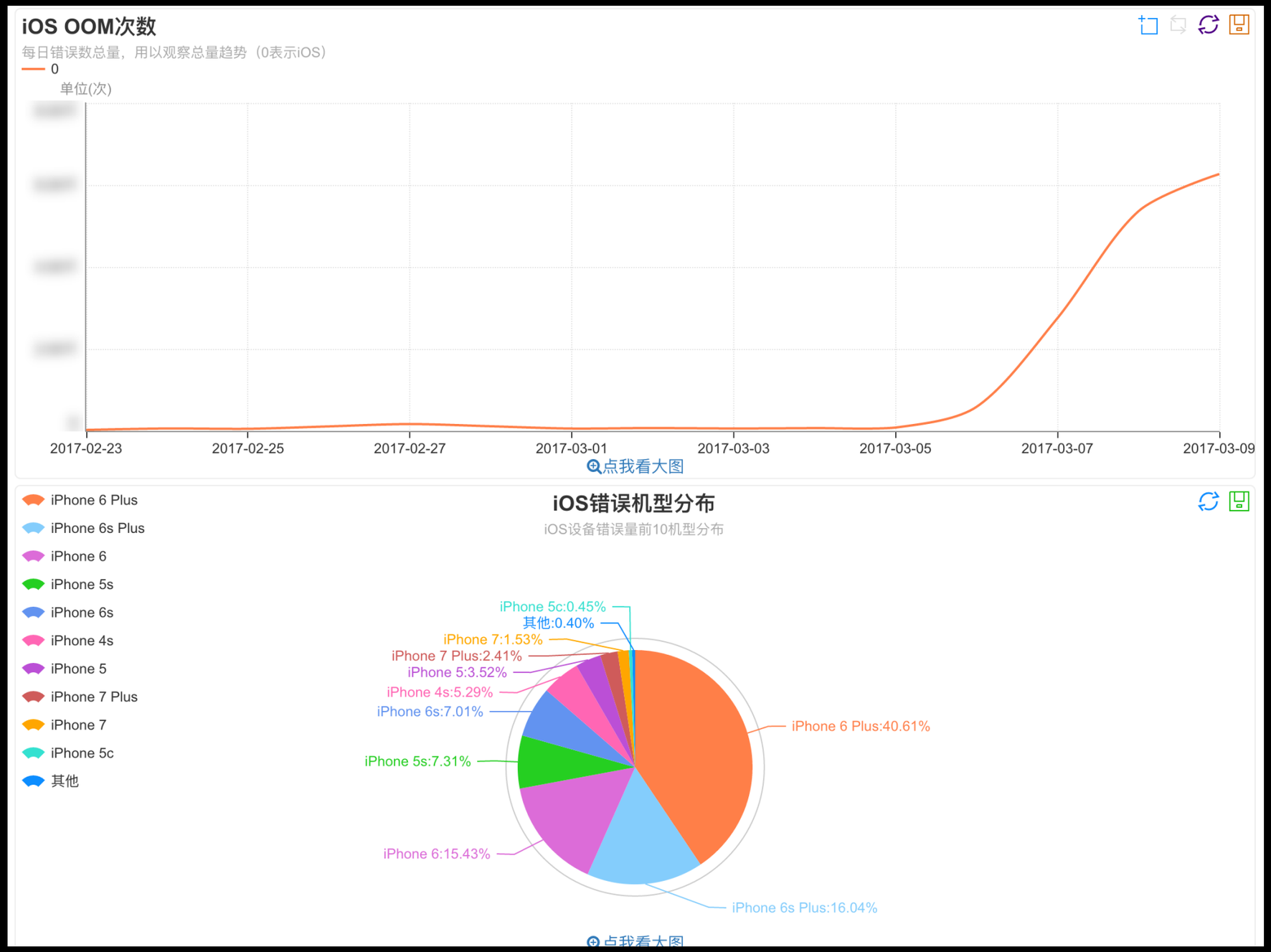
查询

iOS OOM详细信息（昨日数据，数据量: [redacted]）　工具箱>　下载

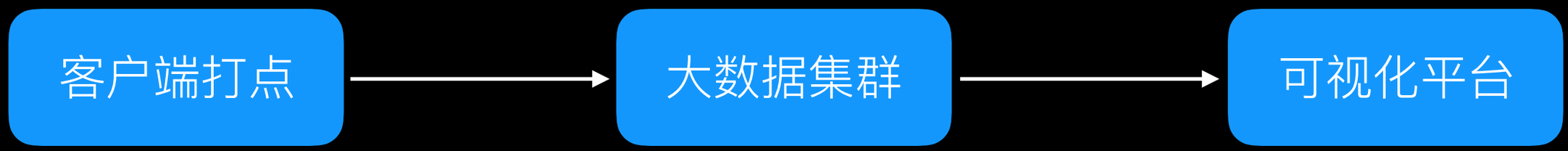
日期	平台	设备id	app版本	设备名称	系统版本	错误次数
2017-03-09	0	[redacted]	9.3.0.12226	iPhone 6s Plus	10.0.1	[redacted]
2017-03-09	0	[redacted]	9.3.0.12226	iPhone 6s	10.1.1	[redacted]
2017-03-09	0	[redacted]	9.3.0.12226	iPhone 6s Plus	10.2	[redacted]
2017-03-09	0	[redacted]	9.3.0.12226	iPhone 6 Plus	10.1.1	[redacted]
2017-03-09	0	[redacted]	9.3.0.12226	iPhone 5c	10.2	[redacted]
2017-03-09	0	[redacted]	9.3.0.12226	iPhone 6 Plus	9.3.3	[redacted]
2017-03-09	0	[redacted]	9.2.0.11023	iPhone 7	10.2.1	[redacted]
2017-03-09	0	[redacted]	9.3.0.12226	iPhone 6 Plus	8.4	[redacted]
2017-03-09	0	[redacted]	9.3.0.12226	iPhone 6 Plus	9.3.2	[redacted]
2017-03-09	0	[redacted]	9.3.0.12226	iPhone 6s Plus	10.2	[redacted]

10　1 / [redacted]　共 [redacted] 条数据

异常设备报表


iOS OOM次数
每日错误数总量，用以观察总量趋势（0表示iOS）
0
单位(次)
2017-02-23
2017-02-25
2017-02-27
2017-03-01
2017-03-03
2017-03-05
2017-03-07
2017-03-09
点我看大图
iPhone 6 Plus
iPhone 6s Plus
iPhone 6
iPhone 5s
iPhone 6s
iPhone 4s
iPhone 5
iPhone 7 Plus
iPhone 7
iPhone 5c
其他
iOS错误机型分布
iOS设备错误量前10机型分布
iPhone 5c:0.45%
其他:0.40%
iPhone 7:1.53%
iPhone 7 Plus:2.41%
iPhone 5:3.52%
iPhone 4s:5.29%
iPhone 6s:7.01%
iPhone 5s:7.31%
iPhone 6:15.43%
iPhone 6 Plus:40.61%
iPhone 6s Plus:16.04%
点我看大图

异常设备报表制作流程



报表制作一个工程师搞定

异常设备体系难点

对「异常」的定义

排查效率不够高

要优化到什么程度不太好确定

要做的事：性能体系

能够知道发生了性能问题

能够知道问题的影响面和严重程度

能够方便排查问题

性能体系难点

定义「异常」

定位到具体的问题原因

无侵入的实现

性能体系细分

直接性能

- 启动
- 页面加载
- 卡顿
- 耗电量
- 磁盘占用
- 网络
- 图片

间接性能

- CPU
- 内存
- 网络请求数
- GC
- 线程
- IO

如何发现性能问题?

给设备打分

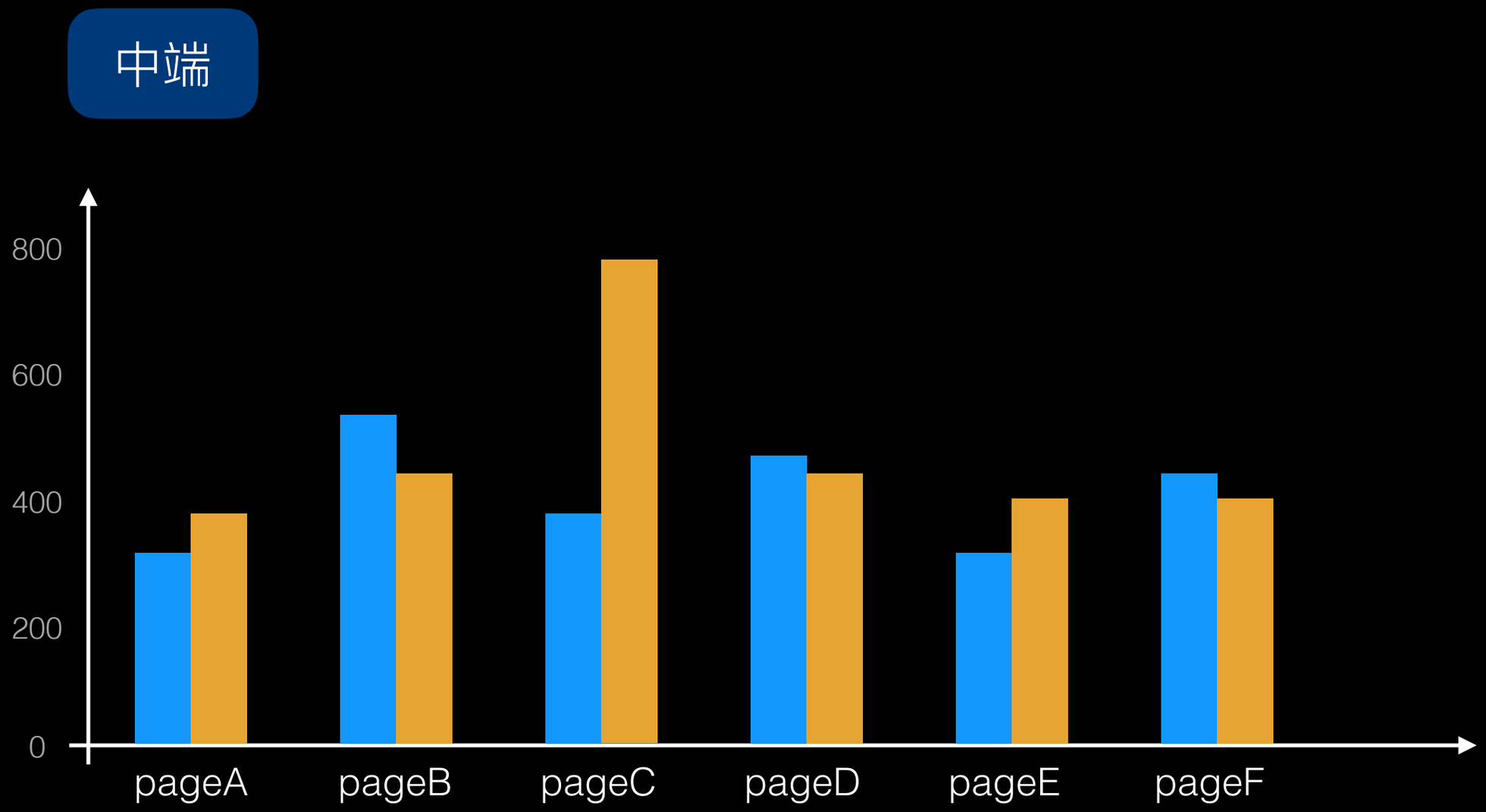
横向对比

设置临界值

给设备打分

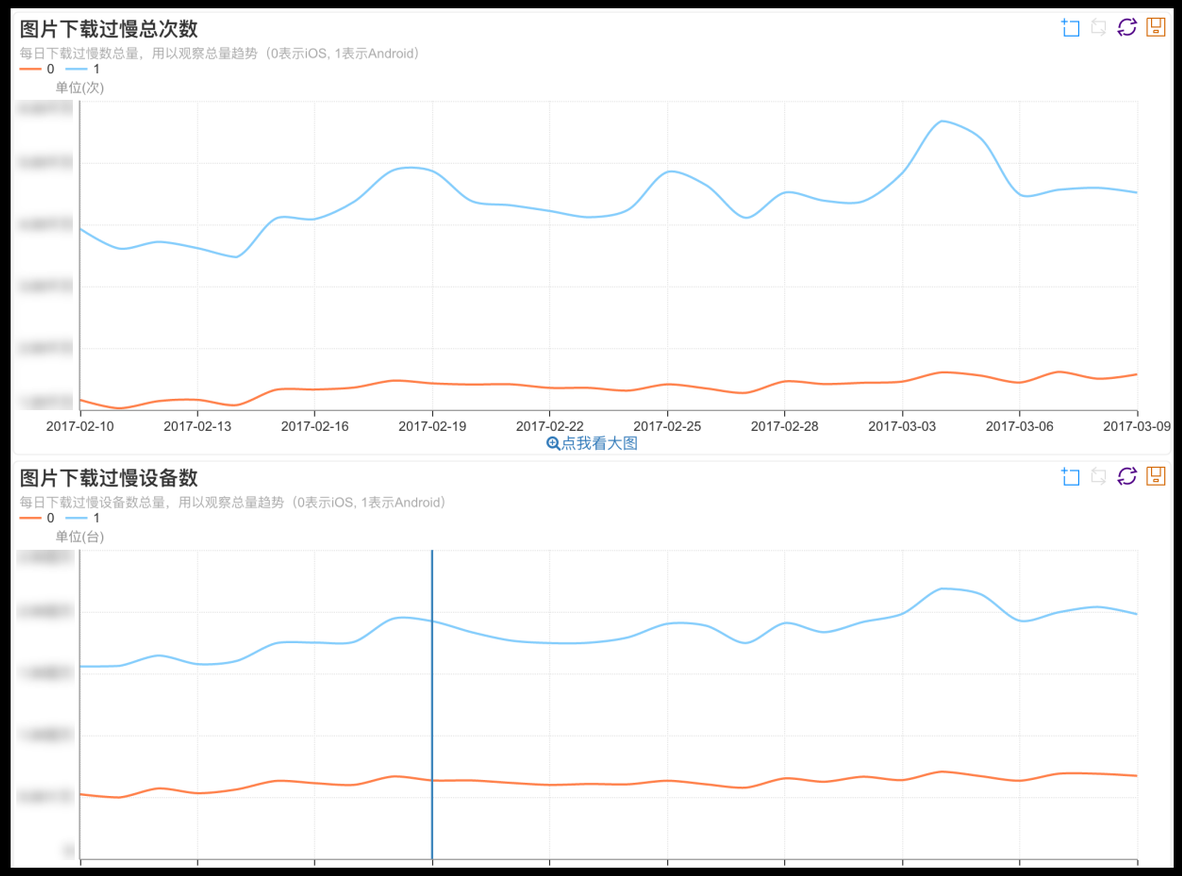
Score = DeviceInfo + Benchmark()

横向对比



设置临界值


图片下载过慢
报表等级: B0
负责人:xiaoqiang
编辑报表
日期
2017-03-09
--
2017-03-09
平台
属于
多个值用^隔开
did
属于
多个值用^隔开
APP版本
属于
多个值用^隔开
设备名称
属于
多个值用^隔开
系统版本
属于
多个值用^隔开
网络类型
属于
多个值用^隔开
错误次数
大于
查询
图片下载过慢的详细信息（昨日数据，数据量: 条）
工具箱›
下载
日期
平台
did
APP版本
设备名称
系统版本
网络类型
错误次数
2017-03-09 0 9.1.0.10185 iPhone7,2 8.1.3 4
2017-03-09 1 9.3.0.4501 GT-N5100 16 4
2017-03-09 1 9.3.0.4501 GT-S7278 16 4
2017-03-09 0 9.3.0.12226 iPhone8,2 10.2.1 4
2017-03-09 1 9.3.0.4501 Le+X620 23 4
2017-03-09 0 9.2.2.11498 iPhone6,2 8.1 3
2017-03-09 1 9.3.0.4501 vivo+Y51A 22 2
2017-03-09 1 9.2.0.3827 vivo+Y23L 19 1
2017-03-09 1 9.3.0.4501 GT-I9128V 16 4
2017-03-09 1 9.2.0.3827 MI+3 19 1
10
1
/
共 条数据

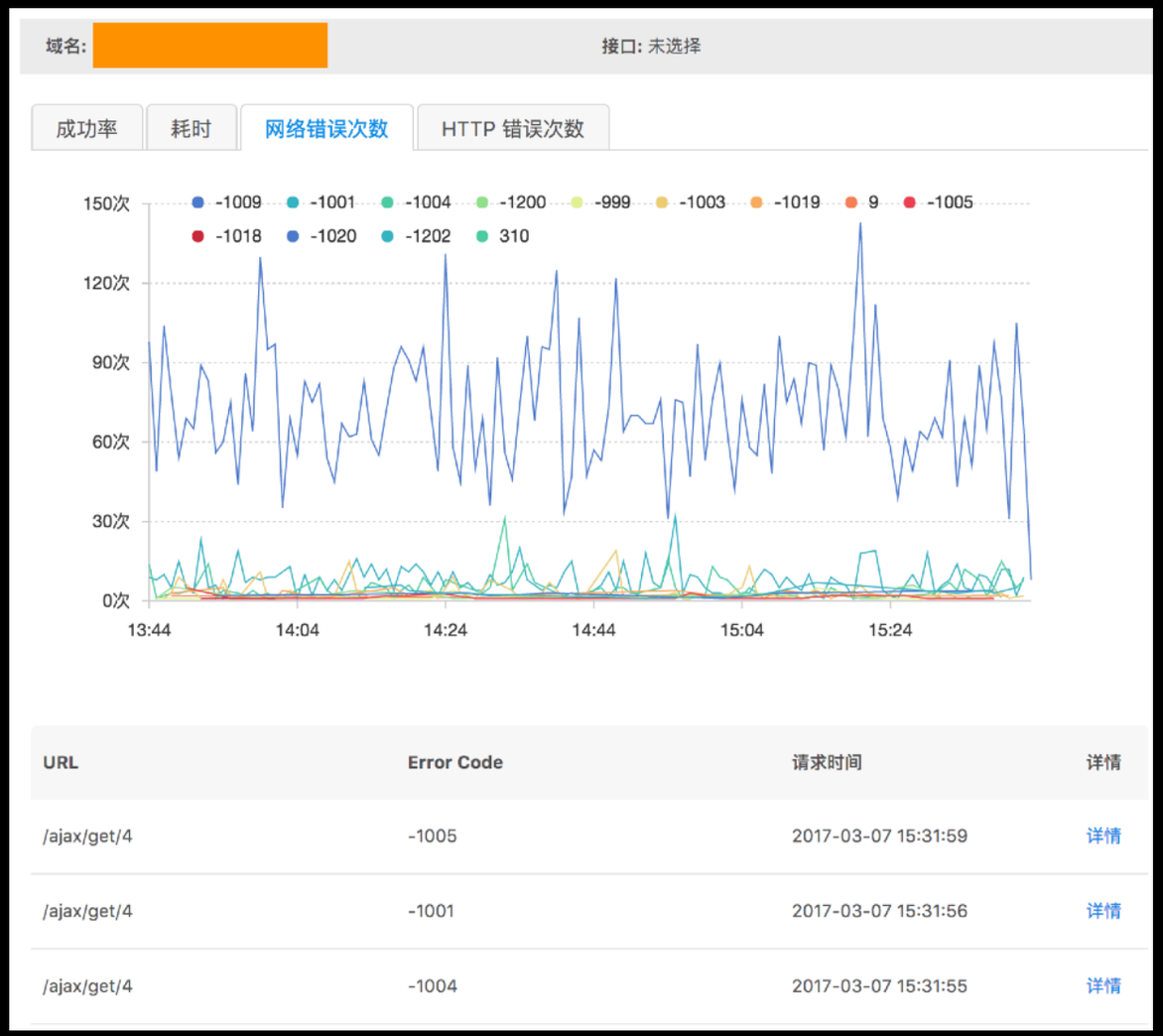

图片下载过慢总次数
每日下载过慢数总量，用以观察总量趋势（0表示iOS, 1表示Android）
0
1
单位(次)
2017-02-10
2017-02-13
2017-02-16
2017-02-19
2017-02-22
2017-02-25
2017-02-28
2017-03-03
2017-03-06
2017-03-09
点我看大图
图片下载过慢设备数
每日下载过慢设备数总量，用以观察总量趋势（0表示iOS, 1表示Android）
0
1
单位(台)

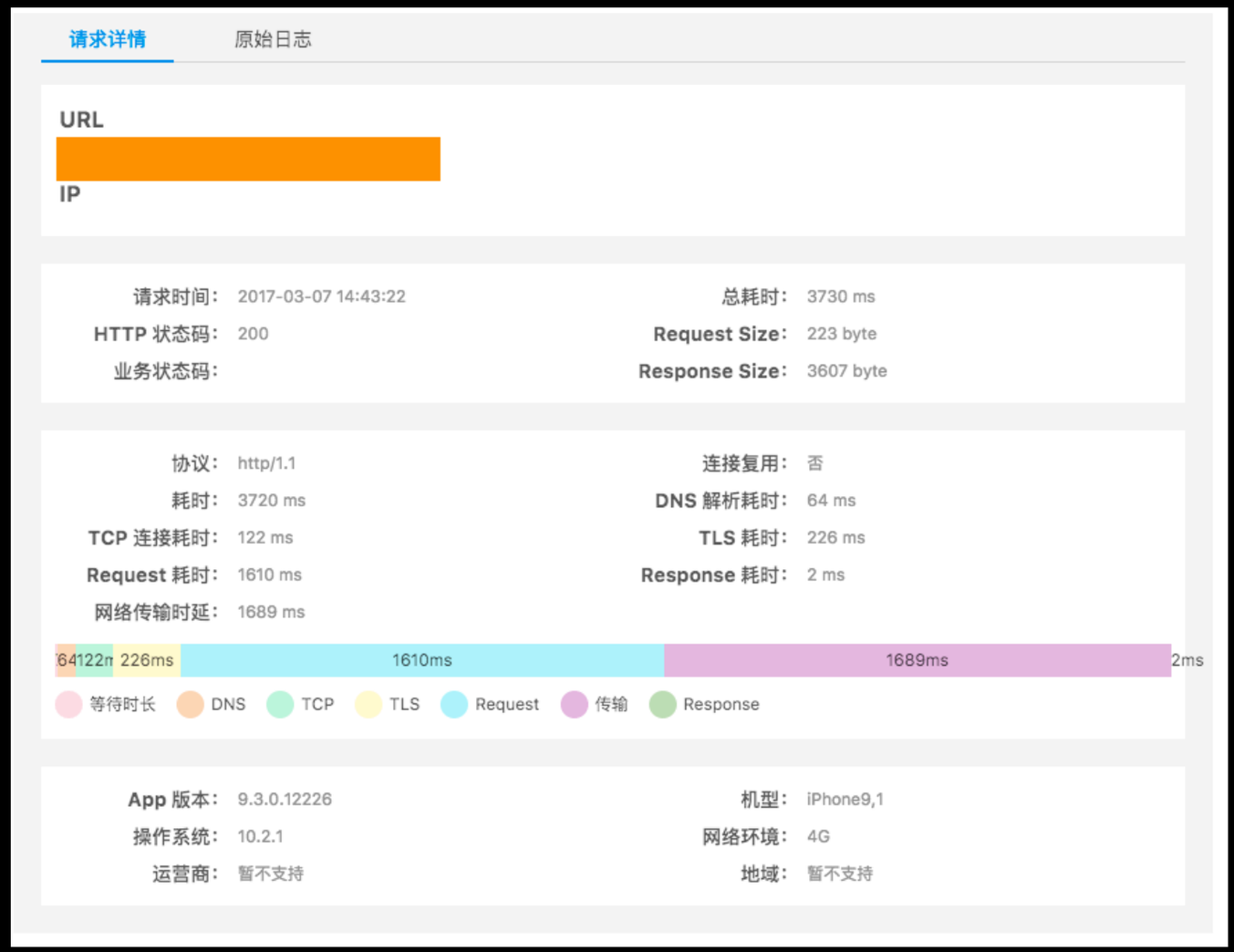
如何排查问题?

更精确的上下文

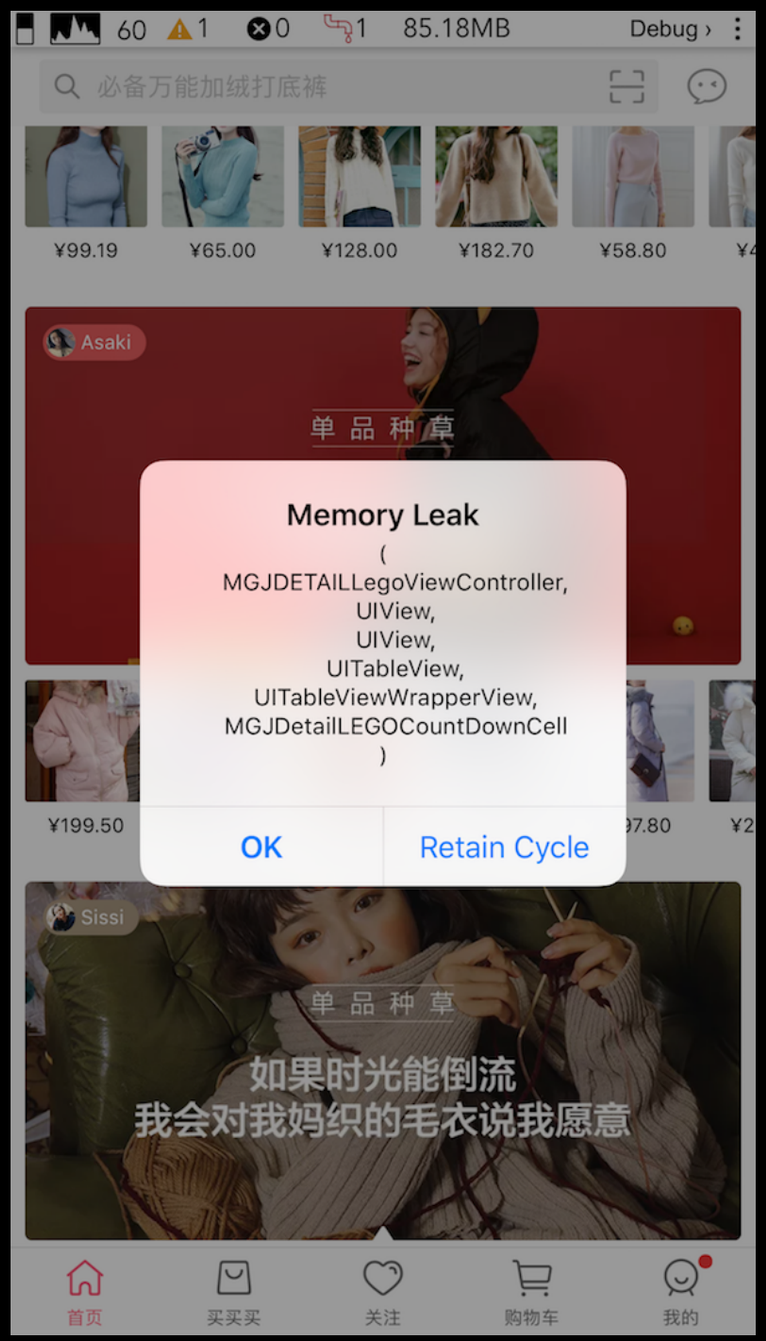
间接性能

更精确的上下文

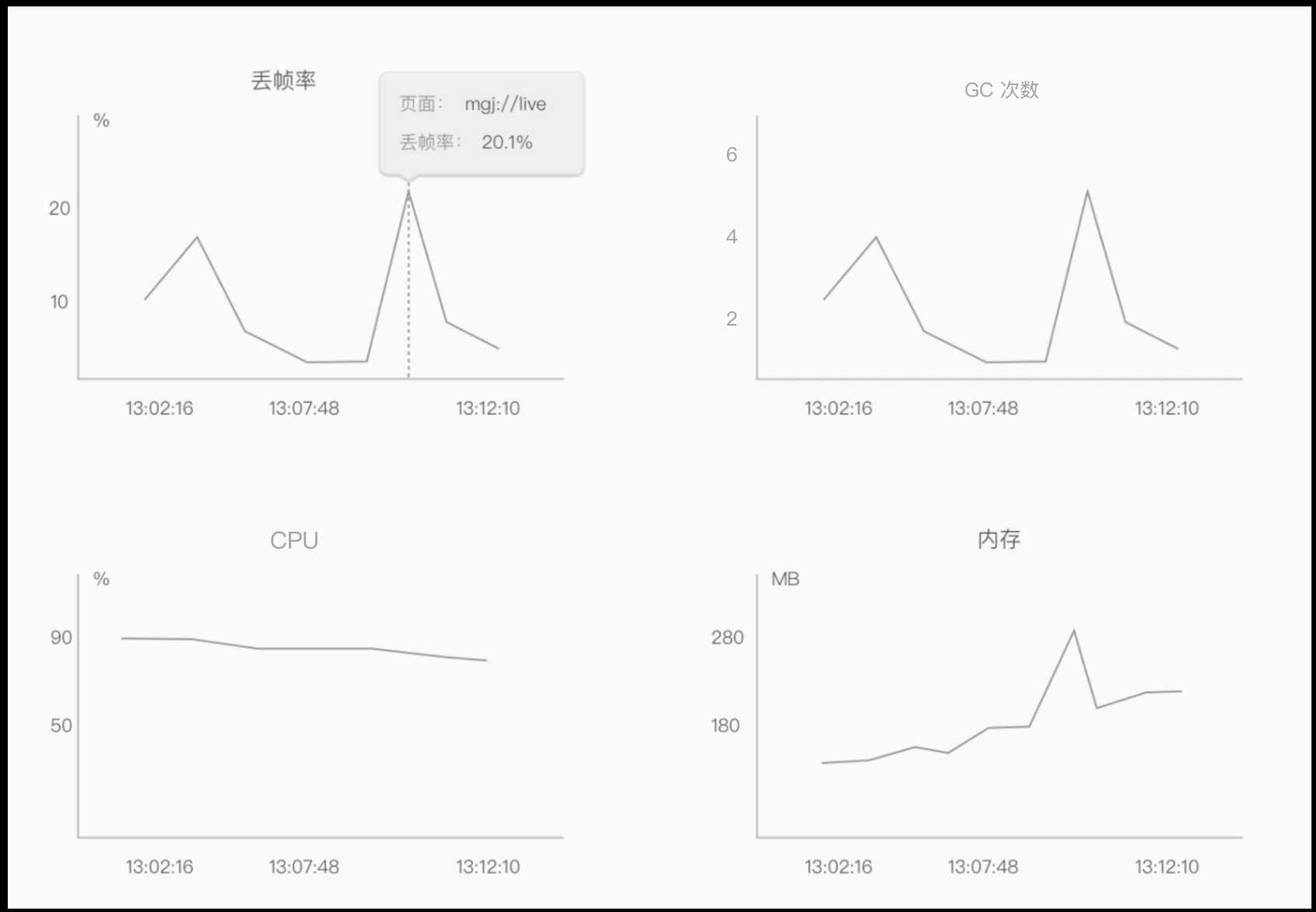

域名:
接口: 未选择
成功率
耗时
网络错误次数
HTTP 错误次数
150次
120次
90次
60次
30次
0次
-1009
-1001
-1004
-1200
-999
-1003
-1019
9
-1005
-1018
-1020
-1202
310
13:44
14:04
14:24
14:44
15:04
15:24
URL
Error Code
请求时间
详情
/ajax/get/4
-1005
2017-03-07 15:31:59
详情
/ajax/get/4
-1001
2017-03-07 15:31:56
详情
/ajax/get/4
-1004
2017-03-07 15:31:55
详情


请求详情
原始日志
URL
IP
请求时间: 2017-03-07 14:43:22
HTTP 状态码: 200
业务状态码:
总耗时: 3730 ms
Request Size: 223 byte
Response Size: 3607 byte
协议: http/1.1
耗时: 3720 ms
TCP 连接耗时: 122 ms
Request 耗时: 1610 ms
网络传输时延: 1689 ms
连接复用: 否
DNS 解析耗时: 64 ms
TLS 耗时: 226 ms
Response 耗时: 2 ms
64
122m
226ms
1610ms
1689ms
2ms
等待时长
DNS
TCP
TLS
Request
传输
Response
App 版本: 9.3.0.12226
操作系统: 10.2.1
运营商: 暂不支持
机型: iPhone9,1
网络环境: 4G
地域: 暂不支持

更精确的上下文


60 1 0 1 85.18MB Debug ›
必备万能加绒打底裤
¥99.19
¥65.00
¥128.00
¥182.70
¥58.80
Asaki
单品种草
Memory Leak
(
MGJDETAILLegoViewController,
UIView,
UIView,
UITableView,
UITableViewWrapperView,
MGJDetailLEGOCountDownCell
)
OK
Retain Cycle
¥199.50
Sissi
单品种草
如果时光能倒流
我会对我妈织的毛衣说我愿意
首页
买买买
关注
购物车
我的

间接性能


丢帧率
%
20
10
页面： mgj://live
丢帧率： 20.1%
13:02:16
13:07:48
13:12:10
GC 次数
6
4
2
13:02:16
13:07:48
13:12:10
CPU
%
90
50
13:02:16
13:07:48
13:12:10
内存
MB
280
180
13:02:16
13:07:48
13:12:10

性能体系整体还处于摸索中

三个事

介绍一套小思维框架

该思维框架在 App 基础体验保障中的使用

我们在基础保障体验中做过的以及要做的事

谢谢

One More Thing