

线上问题排查平台

动态日志

陈潼



陈潼
美团点评Android资深工程师

2015年加入美团点评，先后负责代码静态检查、网络层优化、动态日志等基础设施开发工作。

目录

01 线上排查面临的问题

02 动态日志功能及基本原理介绍

03 动态日志遇到的技术挑战

04 总结与回顾

偶现的线上问题



客户端不能换肤，提示网络有问题

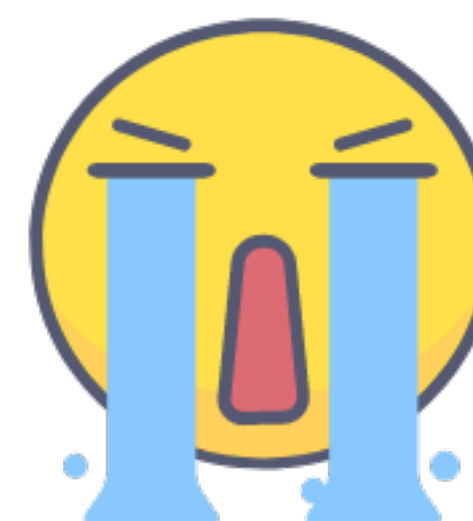
毫无头绪的线上崩溃



反混淆后堆栈

原始崩溃堆栈

```
java.lang.OutOfMemoryError: OutOfMemoryError thrown while trying to throw OutOfMemoryError; no stack trace available
```



咱们 try catch?

现有方案

尝试复现

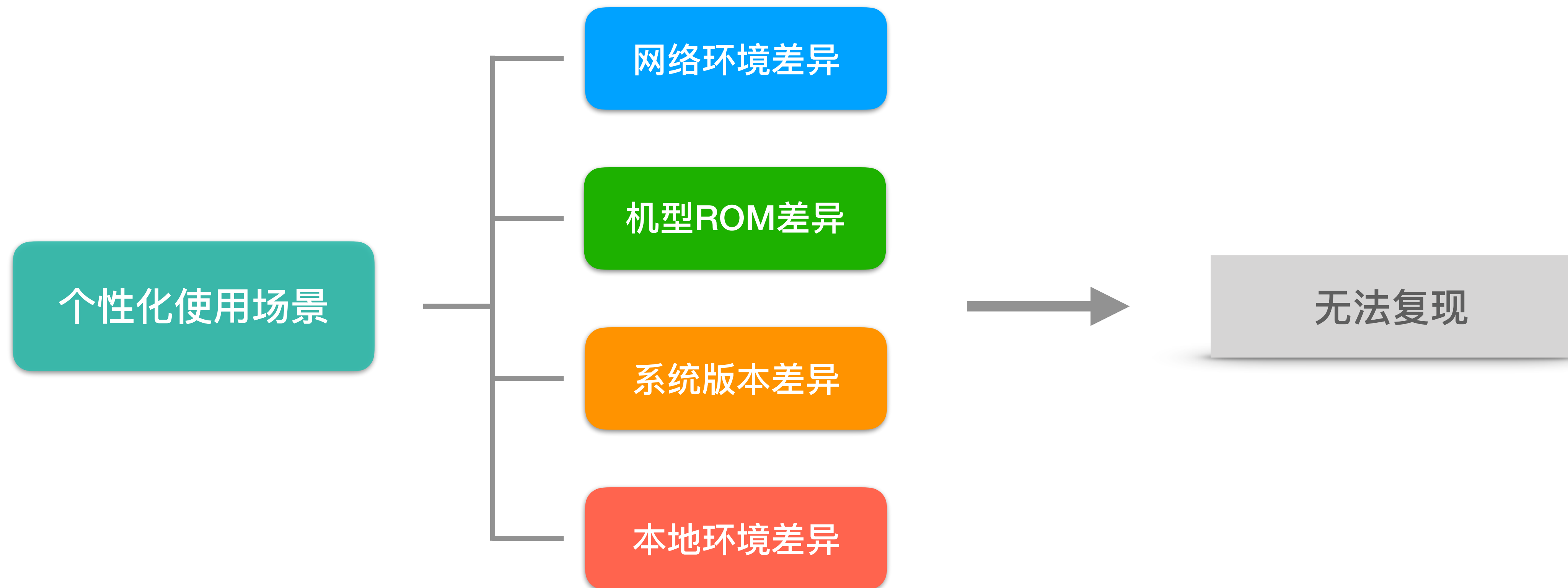
发临时包

在线调试

手动埋点

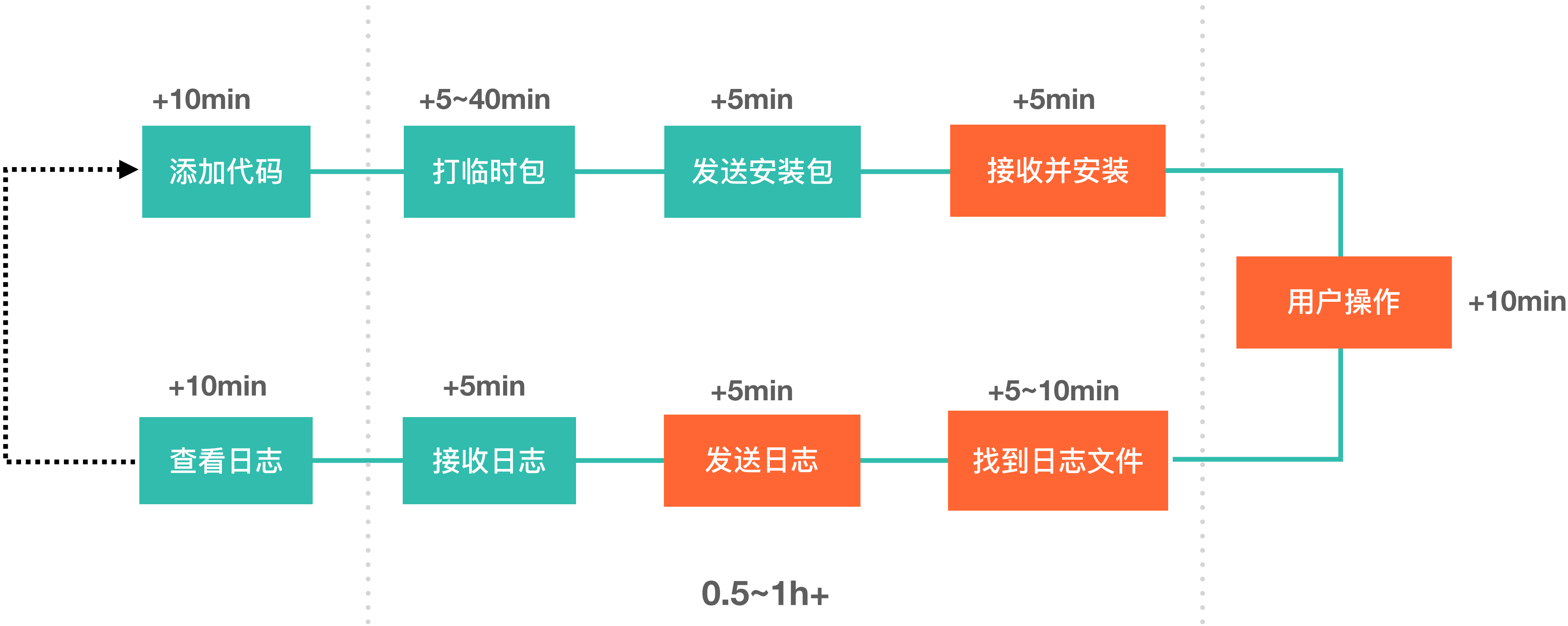
现有方案

尝试复现



现有方案

发临时包



现有方案

在线调试

网络稳定性要求高

保持长连

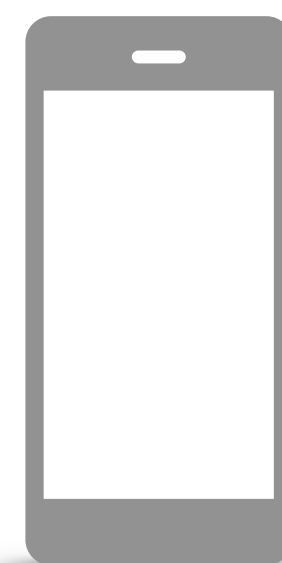


服务器集群

专线打通

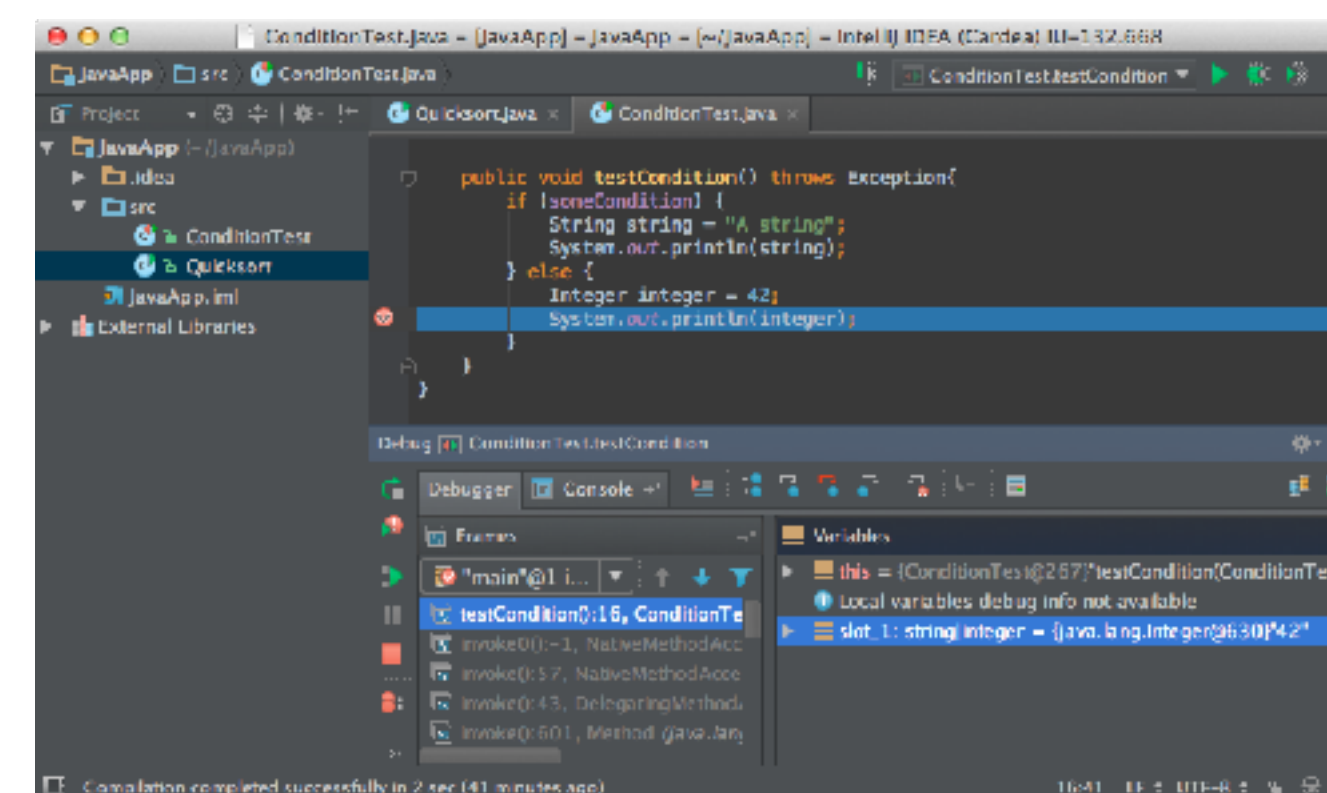
持续沟通

难以接受



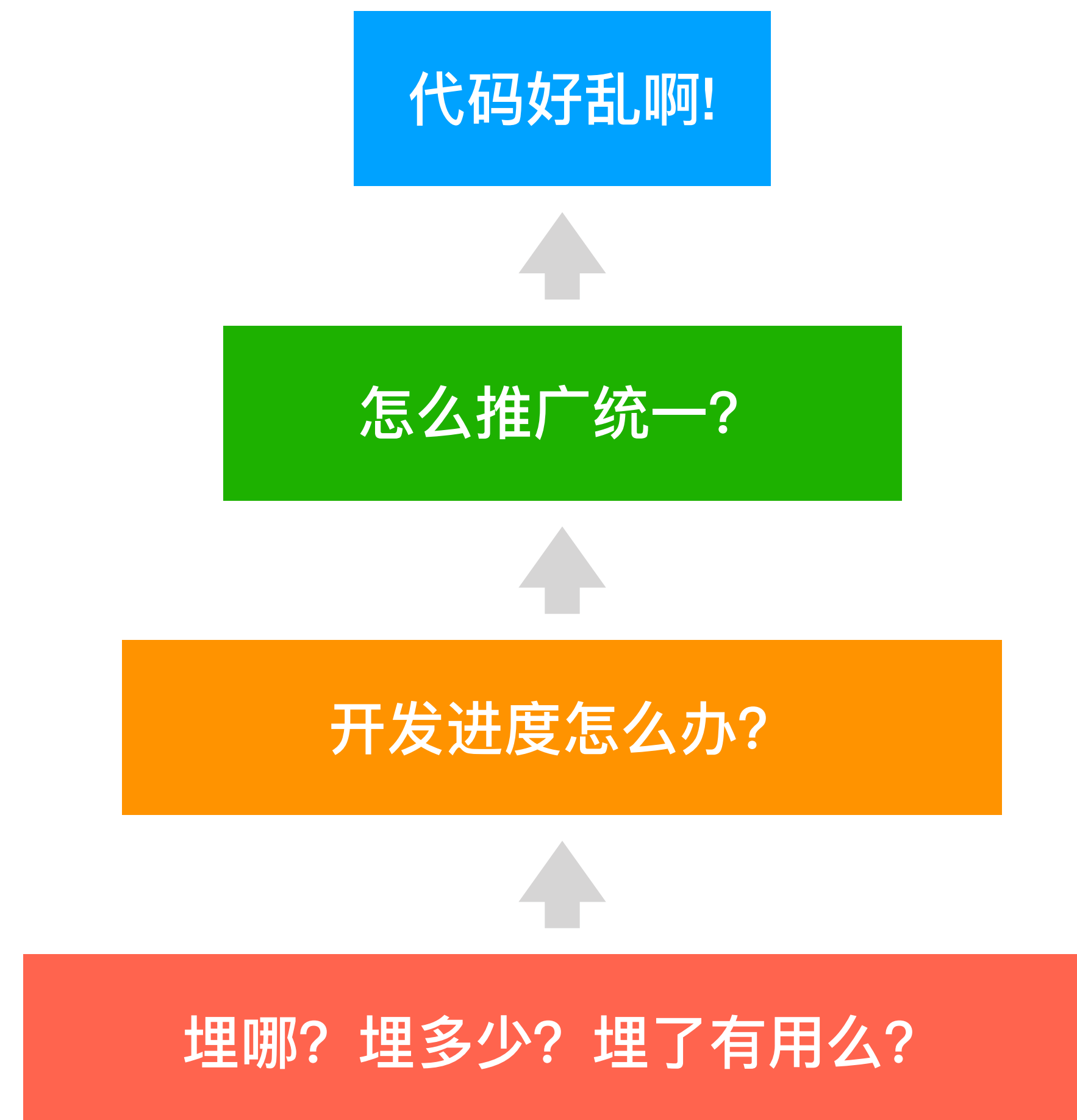
方法内联消失

变量名混淆



现有方案

手动埋点

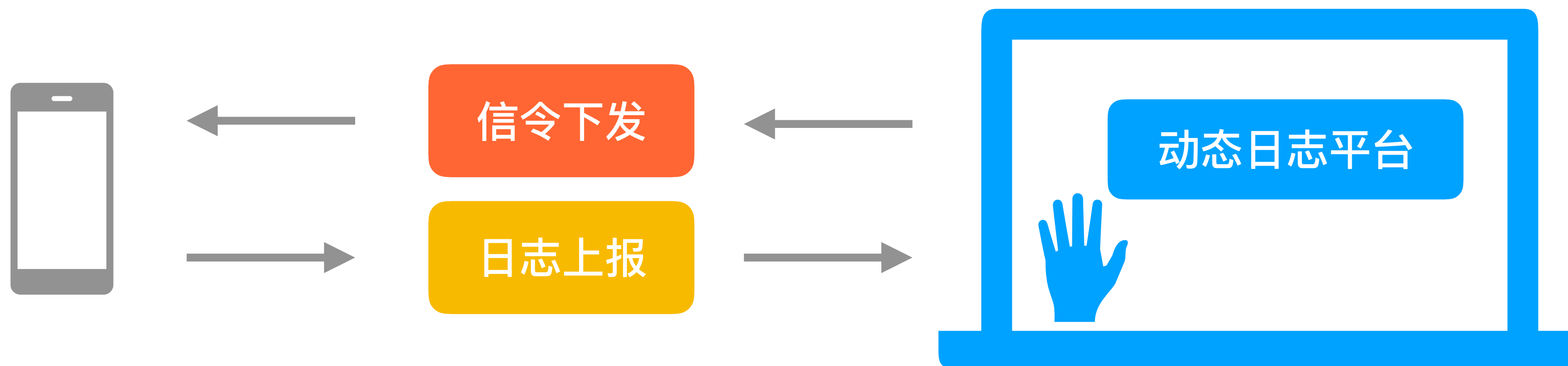


我们想要什么？



动态日志

总体介绍



- 配置简单
- 下发迅速
- 上报提醒
- 查看方便

问题分类与两大功能



功能1：方法执行路径

01 全覆盖自动埋点：方法执行路径记录

自动将所有方法执行路径记录，方便RD通过方法执行路径直接确定没有头绪的问题或锁定范围

方法执行路径记录

原理介绍

```
public int fab(int index) {  
    if (Holmes.isEnable(.....)) {  
        return Holmes.invoke(.....);  
    } else {  
        if (index == 1 || index == 2) {  
            return 1;  
        } else {  
            return fab(index - 1) + fab(index - 2);  
        }  
    }  
}
```



插桩

DB

❖ 构建期自动插桩

- 全面覆盖，内联方法也能搞定
- 不干扰原有代码

❖ DB存储

- 精准查询，减少流量消耗
- 自定义空间大小，循环使用

方法执行路径记录

配置下发

规则1 历史方法执行路径回捞

?

自动关闭

回捞类型：

指定回捞

全部回捞

* 类名：

▼

请填写短类名，可自动匹配

* 方法名：

▼

* 向前（条数）：

0

* 向后（条数）：

500

线程相关：

所有线程

当前线程

主线程

- ✓ 支持精准回捞，省流量
- ✓ 支持全部回捞，看全局

方法执行路径记录

日志上报

元数据 >>

历史结果集:

I

II

III

IV

V

方法签名:

多个关键字请用空格隔开

线程: ☒ 全选

☒

main

☒

statistic.bus

☒

AsyncTask #1

☒

AsyncTask #2

☒

productor

☒

consumer

☒

Thread-1801

☒

xm_im_sdk_receive_message_0

搜索

清空

共 500 条数据, 当前是第1-200条

<

1

2

3

>

- 2017-08-04 10:41:40 [main](#) com.sankuai.meituan.setting>AboutMeituanActivity.onResume()
- 2017-08-04 10:41:40 [main](#) com.sankuai.android.spawn.base.BaseActivity.onResume()
- 2017-08-04 10:41:40 [main](#) com.meituan.android.common.performance.utils.ActivityUtil\$ActivityLifecycleCallbacksWrapper.onActivityResumed(android.app.Activity)
- 2017-08-04 10:41:40 [main](#) com.meituan.android.common.performance.utils.ActivityUtil\$ActivityLifecycleCallbacksWrapper.logAction(android.app.Activity,java.lang.String)
- 2017-08-04 10:41:40 [main](#) com.meituan.android.common.performance.utils.ActivityUtil.logAction(java.lang.String)
- 2017-08-04 10:41:40 [main](#) com.meituan.android.common.performance.statistics.fps.FpsStatisticsManager.getInstance()
- 2017-08-04 10:41:40 [main](#) com.meituan.android.common.performance.statistics.fps.FpsStatisticsManager.statisticsStart()
- 2017-08-04 10:41:40 [main](#) com.meituan.android.common.performance.statistics.fps.FpsStatistics.start()
- 2017-08-04 10:41:40 [main](#) com.meituan.android.common.performance.statistics.fps.FpsInfo.getInstance()

功能2：动态下发代码

02 多时机精准排查：动态下发代码

对目标用户下发代码，动态执行一段代码并将结果上报，用以确定问题

动态下发代码

为什么选择了LUA

运行时库小

支持调用
Java

精简易懂

动态下发代码

LUA示例

```
local t = luajava.bindClass("android.widget.Toast")
local d = t:makeText(getContext(), "Hello World", 0);
d:show();|
```

动态下发LUA

3种时机任意编程

立即

方法开始前

方法结束后

方法开始前执行

原理介绍

```
public int fab(int index) {
    if (Holmes.isEnable(.....)) {
        return Holmes.invoke(.....);
    } else {
        if (index == 1 || index == 2) {
            return 1;
        } else {
            return fab(index - 1) + fab(index - 2);
        }
    }
}
```

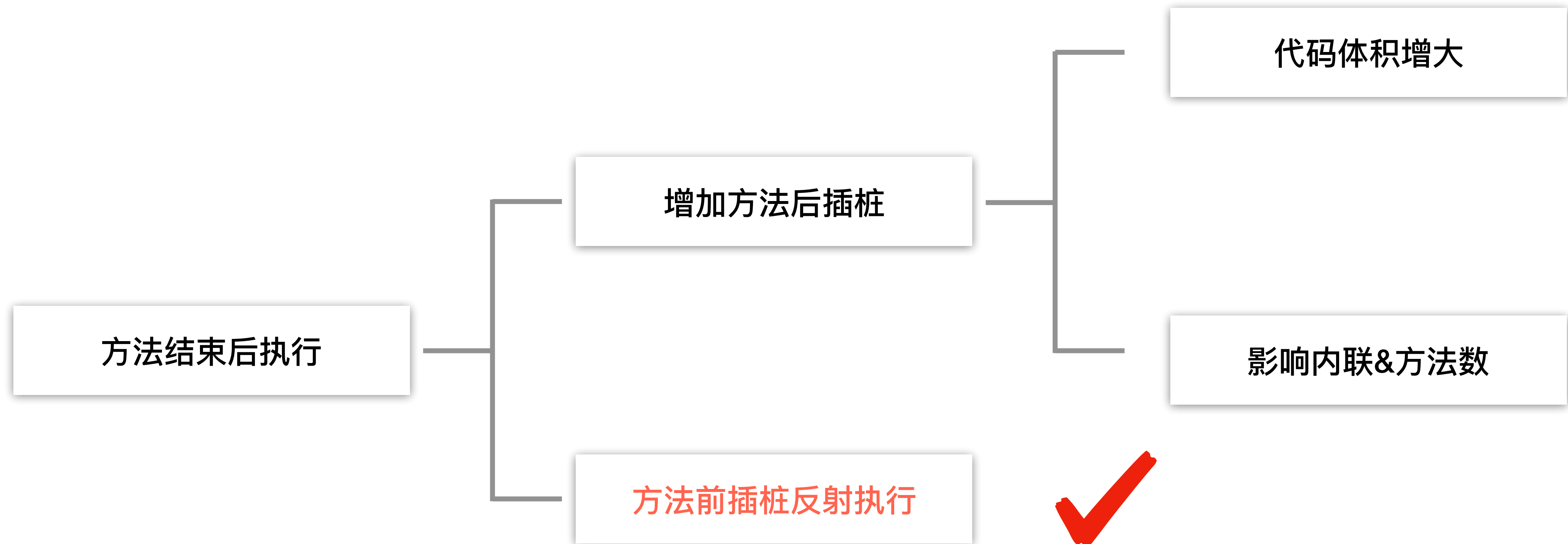
插桩

入参 + 当前对象

LUA

方法结束后执行

面临的问题



方法结束后执行

最终实现

```
public int fab(int index) {  
    if (Holmes.isEnable(.....)) {  
        return Holmes.invoke(.....);  
    } else {  
        if (index == 1 || index == 2) {  
            return 1;  
        } else {  
            return fab(index - 1) + fab(index - 2);  
        }  
    }  
}
```

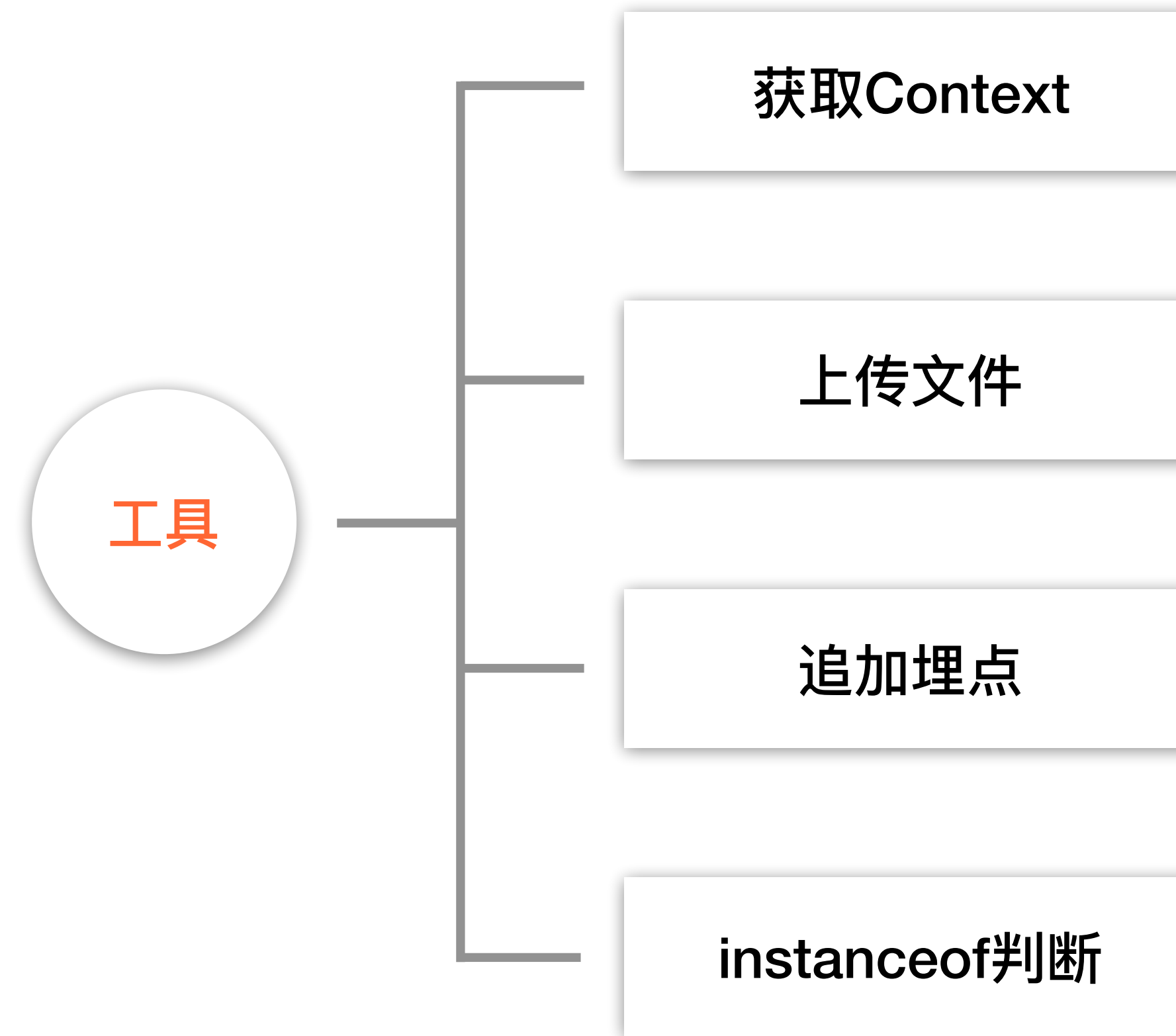
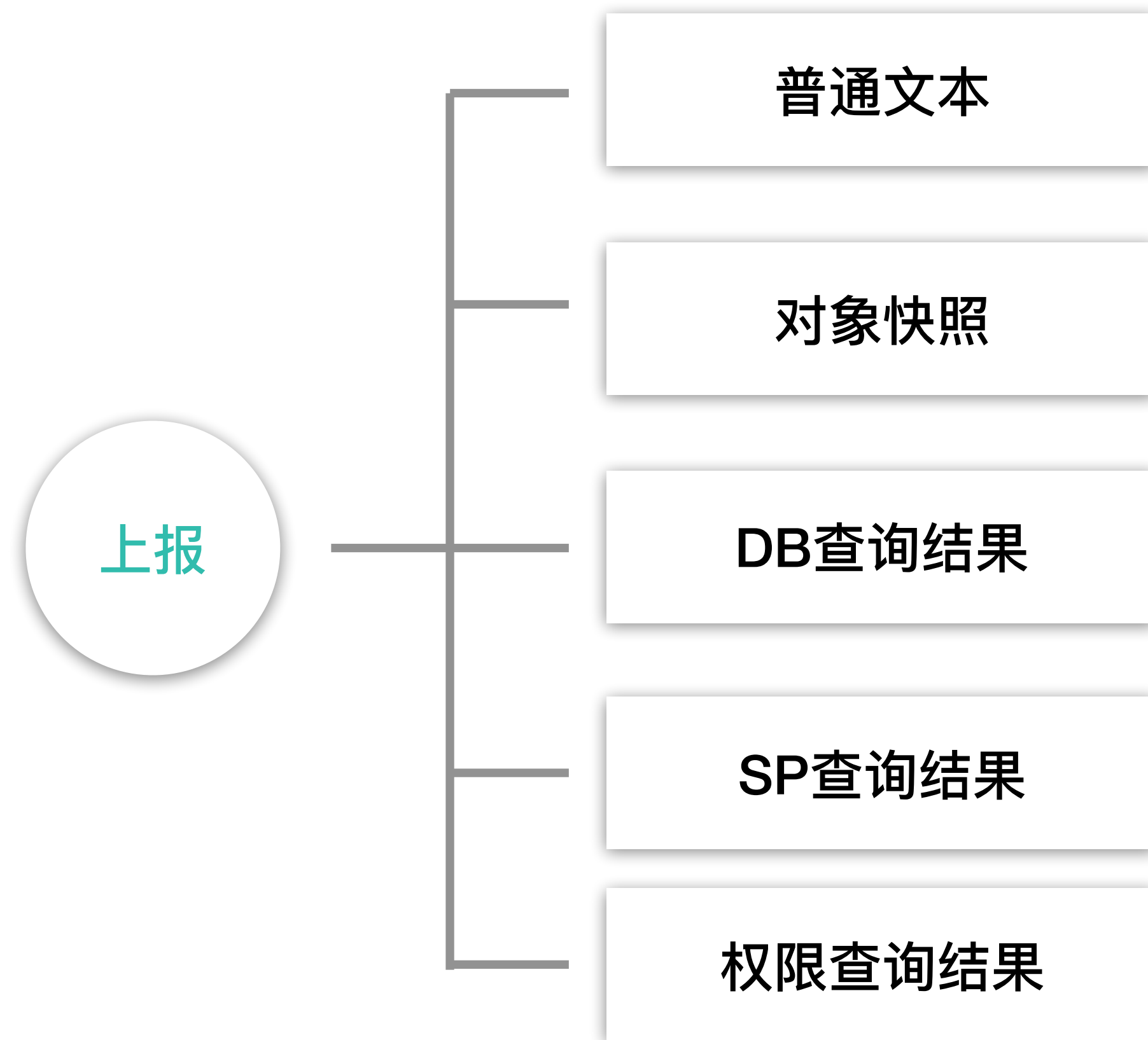
插桩

入参 + 当前对象 + 返回值

LUA

LUA用途

内置API，一行代码搞定



LUA用途

直接调用Java方法

静态方法调用

```
local t = luajava.bindClass("android.widget.Toast")
local d = t.makeText(getContext(), "Hello World", 0);
d.show();|
```

实例方法调用

```
local instance = luajava.newInstance("com.meituan.LuaSample", 365)
instance.execute(1, 2)
```

动态下发LUA

配置下发

规则1 动态下发Lua



自动关闭 ▾



执行时机：

立即执行



lua脚本：

立即执行

方法开始前执行

方法结束后执行

动态下发LUA

配置下发

规则1 动态下发Lua



自动关闭 ▾



执行时机：

方法开始前执行



* 类名：



请填写短类名，可自动匹配

* 方法名：



lua脚本：

lua脚本

[Lua使用介绍](#)

动态下发LUA

日志上报

权限名 ▾	状态 ▾
android.permission.SYSTEM_ALERT_WINDOW	✓ 开启
android.permission.READ_SETTINGS	✗ 关闭
android.permission.READ_PHONE_STATE	✓ 开启
com.meizu.flyme.push.permission.RECEIVE	✗ 关闭
android.permission.READ_USER_DICTIONARY	✓ 开启
android.permission.READ_CONTACTS	✓ 开启
android.permission.READ_CALENDAR	✓ 开启
android.permission.READ_SMS	✓ 开启
com.sankuai.common.PERMISSION	✓ 开启
android.permission.WAKE_LOCK	✓ 开启

动态下发LUA

日志上报

执行线程: `main-1`

▼ `this`: `com.sankuai.meituan.setting>AboutMeituanActivity`

`clickCount`: `int` = 0

`MetaChannel`: `java.lang.String` = 'undefined'

`updateClickable`: `boolean` = true

`isActive`: `boolean` = false

`mStateSaved`: `boolean` = false

`mThemeld`: `int` = 2131493062

`mCreated`: `boolean` = false

`mNextCandidateRequestIndex`: `int` = 0

`mReallyStopped`: `boolean` = true

`mRequestedPermissionsFromFragment`: `boolean` = false

`mResumed`: `boolean` = false

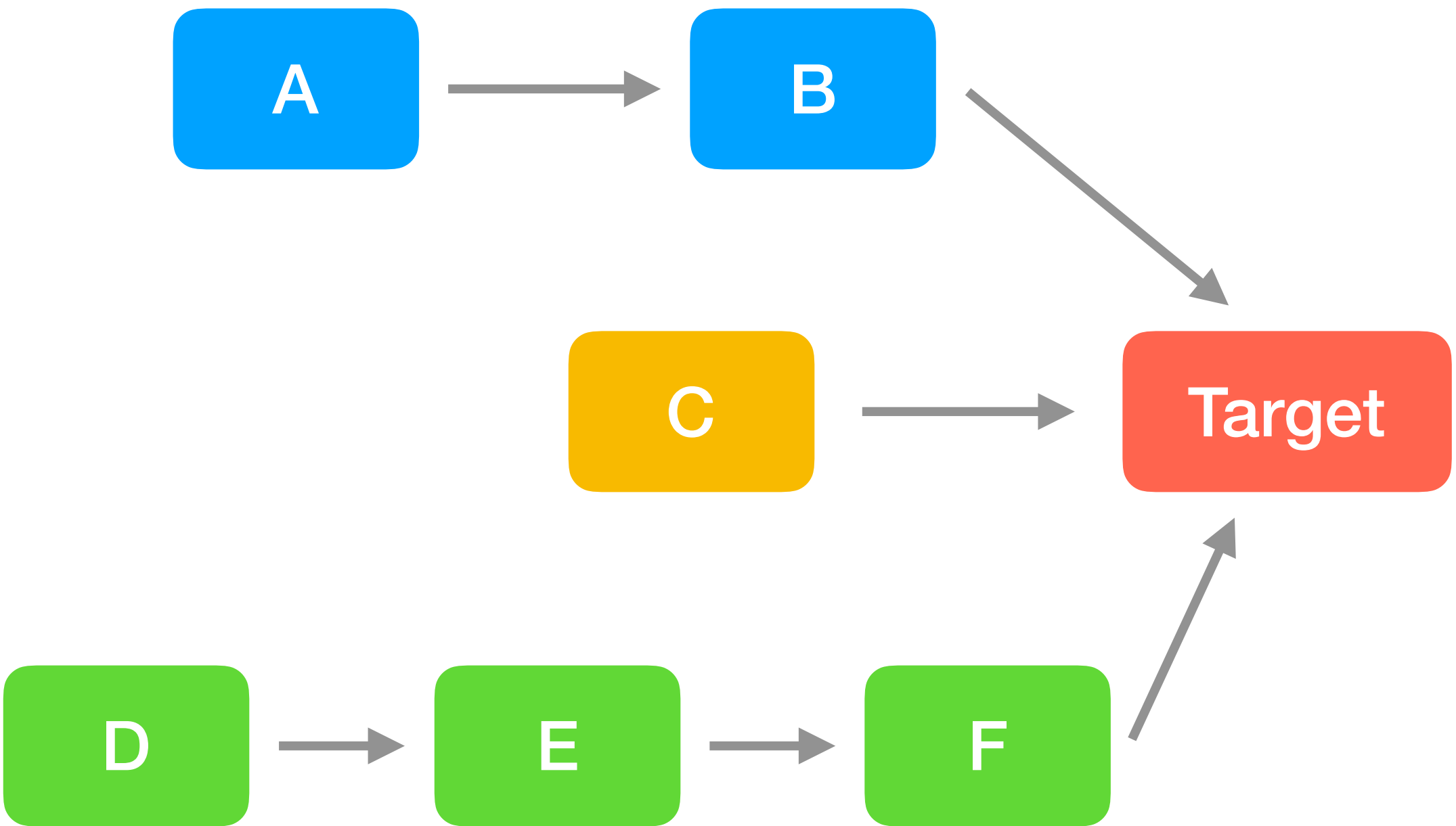
动态下发LUA

日志上报

id ↕	seq ↕	method_number ↕	process_id ↕	thread_id ↕	thread_name ↕	version_name ↕	time ↕
1	1	12d792a1a89723210314409a9df3f062	15870	4788	AsyncTask #1	8.8	1512631869481
2	2	1b4c0fedba29658bd3eb6e478180e171	15870	4788	AsyncTask #1	8.8	1512631869482
3	3	e2ebe3a1527239f142155f28655ea91f	15870	4788	AsyncTask #1	8.8	1512631869482
4	4	b6417f41c2645b733119d7e9b12874f6	15870	4788	AsyncTask #1	8.8	1512631869482
5	5	4d69245191c68b9994715d3af34cb8a9	15870	4788	AsyncTask #1	8.8	1512631869482
6	6	a792092e81aeb580cc019d741e84891d	15870	4788	AsyncTask #1	8.8	1512631869482
7	7	c94291bb8ce3517926c142b0f02bb76d	15870	4788	AsyncTask #1	8.8	1512631869482
8	8	8cf5ba9c88144a7f2b6d800ae32f1354	15870	4788	AsyncTask #1	8.8	1512631869482

动态下发LUA

多种调用支持堆栈筛选



根据方法堆栈筛选

```
com.meituan.passport.UserCenter.isLogin(UserCenter.java:210)
com.sankuai.meituan.init.StatisticsEnvironment.getUid(StatisticsEnvironment.java:90)
com.meituan.android.common.statistics.channel.ChannelManager.getEnvironment(ChannelManager.java:233)
com.meituan.android.common.statistics.channel.ChannelManager.getDefaultEnvironment(ChannelManager.java:128)
com.meituan.android.common.statistics.channel.Channel.getAllEnvironment(Channel.java:168)
com.meituan.android.common.statistics.channel.Channel$1.run(Channel.java:1384)
android.os.Handler.handleCallback(Handler.java:815)
android.os.Handler.dispatchMessage(Handler.java:104)
android.os.Looper.loop(Looper.java:194)
android.os.HandlerThread.run(HandlerThread.java:61)
```

221

```
com.meituan.passport.UserCenter.isLogin(UserCenter.java:210)
com.sankuai.meituan.init.StatisticsEnvironment.getLoginType(StatisticsEnvironment.java:100)
com.meituan.android.common.statistics.channel.ChannelManager.getEnvironment(ChannelManager.java:238)
com.meituan.android.common.statistics.channel.ChannelManager.getDefaultEnvironment(ChannelManager.java:128)
com.meituan.android.common.statistics.channel.Channel.getAllEnvironment(Channel.java:168)
com.meituan.android.common.statistics.channel.Channel$1.run(Channel.java:1384)
android.os.Handler.handleCallback(Handler.java:815)
android.os.Handler.dispatchMessage(Handler.java:104)
android.os.Looper.loop(Looper.java:194)
android.os.HandlerThread.run(HandlerThread.java:61)
```

216

```
com.meituan.android.base.knb.impl.EnvironmentImpl.getUserId(EnvironmentImpl.java:96)
com.sankuai.meituan.android.knb.KNBWebCompatDelegate.getUserId(KNBWebCompatDelegate.java:708)
com.sankuai.meituan.android.knb.KNBWebCompatDelegateV2Impl.wrapUrl(KNBWebCompatDelegateV2Impl.java:1074)
com.sankuai.meituan.android.knb.KNBWebCompatDelegateV2Impl$6.run(KNBWebCompatDelegateV2Impl.java:679)
android.os.Handler.handleCallback(Handler.java:815)
android.os.Handler.dispatchMessage(Handler.java:104)
android.os.Looper.loop(Looper.java:194)
android.app.ActivityThread.main(ActivityThread.java:5714)
```

6

技术难点

难点1：方法执行路径记录

遇到的挑战

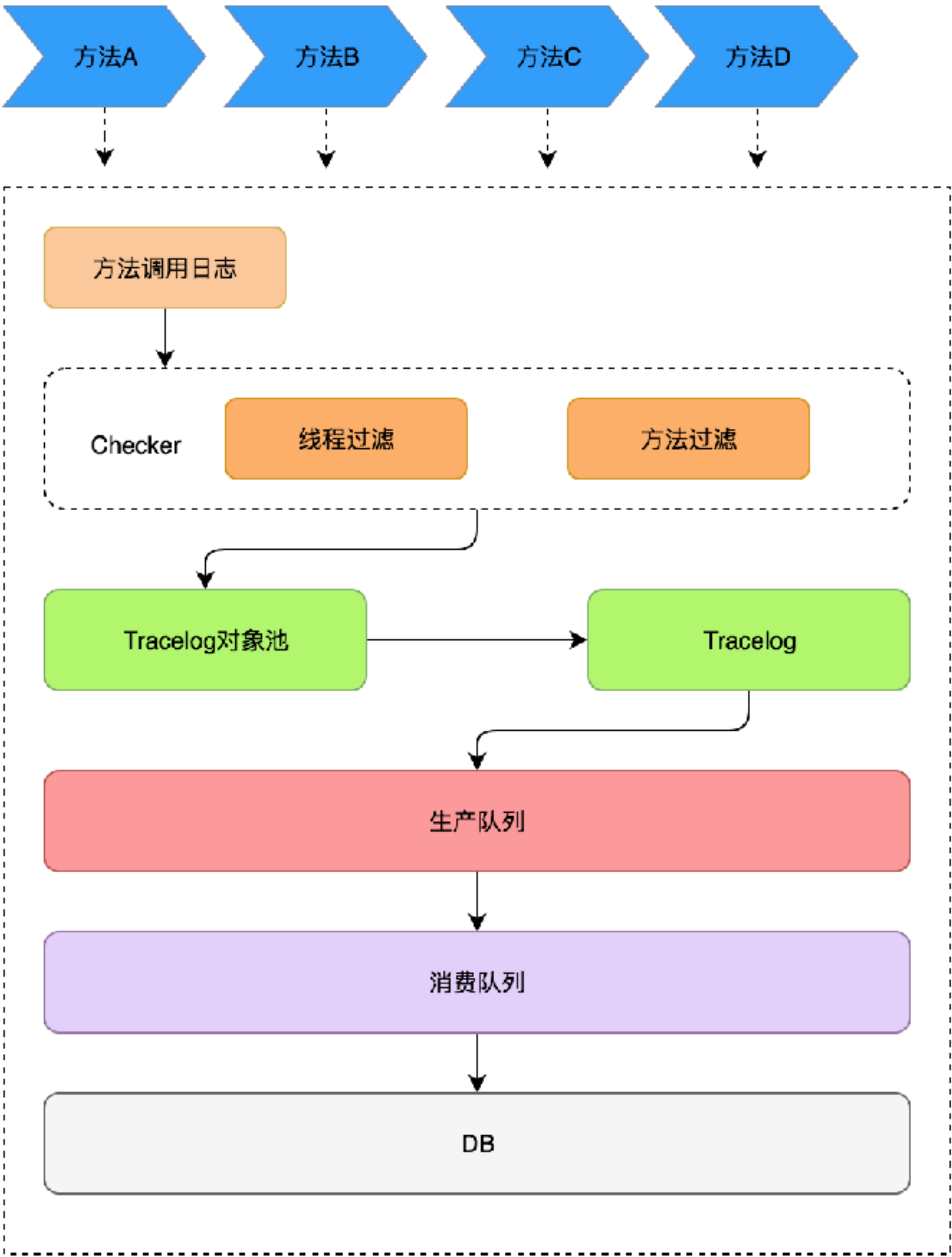
- 与常规手动埋点比，写入数据发生量级变化

数据量大

- ▶ 进入美团首页已经执行11万方法
- ▶ 1ms调用近300个方法

难点1：方法执行路径记录

工作流程图



- ❖ 插桩过滤
- ❖ 动态识别机制
- ❖ 对象池方案

难点1：方法执行路径记录

优化效果

Name	Incl Cpu Time %	Incl Cpu Time	Calls+Recurs Calls/Total	Excl Cpu Time %	Excl Cpu Time	Incl Real Time %	Incl Real Time	Excl Real Time %	Excl Real Time
96 android.view.ThreadedRenderer.draw (Landroid/view/View;Landroid/view/View...	3.6%	554.998	317+0	0.0%	0.000	0.6%	6716.481	0.0%	0.000
97 com.bumptech.glide.load.engine.DecodeJob\$SourceWriter.write (Ljava/io/File;)Z	3.6%	553.089	68+0	0.0%	0.000	0.2%	2653.485	0.0%	0.000
98 com.meituan.android.common.analyse.mtanalyse.Analyzer\$EventReportTask....	3.6%	552.960	1+0	0.0%	0.000	0.5%	5709.861	0.0%	0.000
99 com.meituan.android.common.analyse.mtanalyse.Analyzer\$EventReportTask....	3.6%	552.960	1+0	0.0%	0.000	0.5%	5709.861	0.0%	0.000
100 com.meituan.android.common.holmes.db.DBHandler.insert (Ljava/util/Collection;)V	3.6%	551.015	27+0	0.1%	11.942	0.4%	4905.871	0.0%	11.942
101 com.meituan.android.common.holmes.db.TraceDBManager.insert (Ljava/util/...	3.6%	551.015	27+0	0.0%	0.000	0.4%	4905.871	0.0%	0.000
102 com.meituan.android.common.holmes.trace.Producer\$2.run ()V	3.6%	551.015	27+0	0.0%	0.000	0.4%	4905.871	0.0%	0.000
▼ Parents									
2 java.util.concurrent.ThreadPoolExecutor.runWorker (Ljava/util/concurrent/...	100.0%	551.015	27/27			100.0%	4905.871		
▼ Children									
self	0.0%	0.000				0.0%	0.000		
101 com.meituan.android.common.holmes.db.TraceDBManager.insert (Ljav...	100.0%	551.015	27/27			100.0%	4905.871		
103 com.sankuai.meituan.common.net.nvnetwork.NVUserAgentInterceptor.interc...	3.5%	547.179	26+0	0.0%	0.000	0.7%	7380.752	0.0%	0.000
104 com.sankuai.meituan.common.net.nvnetwork.NVDpMapInterceptor.intercept...	3.5%	546.930	26+0	0.0%	0.000	0.7%	7369.983	0.0%	0.000
105 com.sankuai.meituan.common.net.nvnetwork.NVUIDInterceptor.intercept (L...	3.4%	526.838	26+0	0.0%	0.000	0.6%	6951.670	0.0%	0.000
106 com.bumptech.glide.load.resource.bitmap.BitmapEncoder.encode (Lcom/bu...	3.4%	520.606	60+0	0.0%	0.000	0.2%	2430.810	0.0%	0.000
107 com.bumptech.glide.load.resource.bitmap.BitmapEncoder.encode (Ljava/lan...	3.4%	520.606	60+0	0.0%	0.000	0.2%	2430.810	0.0%	0.000
108 com.sankuai.meituan.common.net.nvnetwork.NVAntiCrawlerInterceptor.inter...	3.4%	519.983	26+0	0.0%	0.000	0.6%	6894.607	0.0%	0.000
109 com.bumptech.glide.load.engine.DecodeJob.decodeFromSourceData (Ljava/...	3.4%	519.310	137+0	0.0%	0.000	0.4%	4118.670	0.0%	0.000
110 com.meituan.robust.PatchProxy.isSupport ([Ljava/lang/Object;Ljava/lang/Obj...	3.4%	518.825	500+0	1.1%	172.608	0.3%	3622.434	0.0%	172.608
111 android.graphics.Bitmap.compress (Landroid/graphics/Bitmap\$CompressFor...	3.3%	516.769	59+0	0.0%	0.000	0.2%	2420.773	0.0%	0.000
112 android.graphics.Bitmap.nativeCompress ([Ljava/io/OutputStream;[B)Z	3.3%	516.769	59+0	3.2%	497.416	0.2%	2420.773	0.0%	497.416

Name	Incl Cpu Time %	Incl Cpu Time	Calls+Recurs Calls/Total	Excl Cpu Time %	Excl Cpu Time	Incl Real Time %	Incl Real Time	Excl Real Time %	Excl Real Time
0 java.lang.Thread.run ()V	41.7%	6444.688	107+0	0.1%	15.440	68.8%	773545.523	0.0%	15.440
1 java.util.concurrent.ThreadPoolExecutor\$Worker.run ()V	40.0%	6188.517	98+0	0.0%	0.000	59.1%	664390.556	0.0%	0.000
2 java.util.concurrent.ThreadPoolExecutor.runWorker (Ljava/util/concurrent/Threa...	40.0%	6188.517	98+0	0.0%	0.000	59.1%	664390.556	0.0%	0.000
▼ Parents									
1 java.util.concurrent.ThreadPoolExecutor\$Worker.run ()V	100.0%	6188.517	98/98			100.0%	664390.556		
▼ Children									
self	0.0%	0.000				0.0%	0.000		
12 java.util.concurrent.FutureTask.run ()V	41.7%	2581.120	153/167			4.6%	30439.964		
25 com.sankuai.meituan.activity.modules.UpdateDialog\$1\$1.run ()V	21.2%	1312.533	1/1			1.5%	9816.760		
81 com.sankuai.meituan.retrofit2.ClientCall\$1.run ()V	9.7%	601.635	12/12			1.2%	7995.045		
102 com.meituan.android.common.holmes.trace.Producer\$2.run ()V	8.9%	551.015	27/27			0.7%	4905.871		
150 java.util.concurrent.ThreadPoolExecutor.getTask ()Ljava/lang/Runnable;	5.1%	315.262	418/418			79.0%	524808.492		
209 com.meituan.android.common.holmes.trace.Producer\$1.run ()V	3.3%	206.565	1/1			3.7%	24255.098		
293 com.meituan.android.common.holmes.trace.Checker\$1.run ()V	2.0%	126.344	25/25			0.2%	1215.501		
332 com.sankuai.meituan.activity.modules.DelayRequests.run ()V	1.8%	111.568	1/1			0.1%	938.185		
493 java.util.concurrent.ScheduledThreadPoolExecutor\$ScheduledFutureTa...	1.1%	67.961	13/13			0.2%	1141.274		
589 com.sankuai.meituan.activity.modules.IndexWindows\$1.run ()V	0.8%	51.783	2/2			0.1%	919.806		
847 com.sankuai.meituan.activity.modules.AsyncTaskModule\$1.run ()V	0.5%	31.035	1/1			0.0%	320.485		
858 android.os.AsyncTask\$SerialExecutor\$1.run ()V	0.5%	30.115	6/6			2.6%	17485.404		
895 java.lang.UNIXProcess\$3.run ()V	0.5%	28.203	35/35			0.1%	517.242		
925 com.sankuai.meituan.init.PushInitManager\$1.run ()V	0.4%	26.606	1/1			0.0%	203.567		
960 com.sankuai.meituan.init.PushInitManager\$1.run ()V	0.4%	24.890	22/22			0.4%	2720.776		

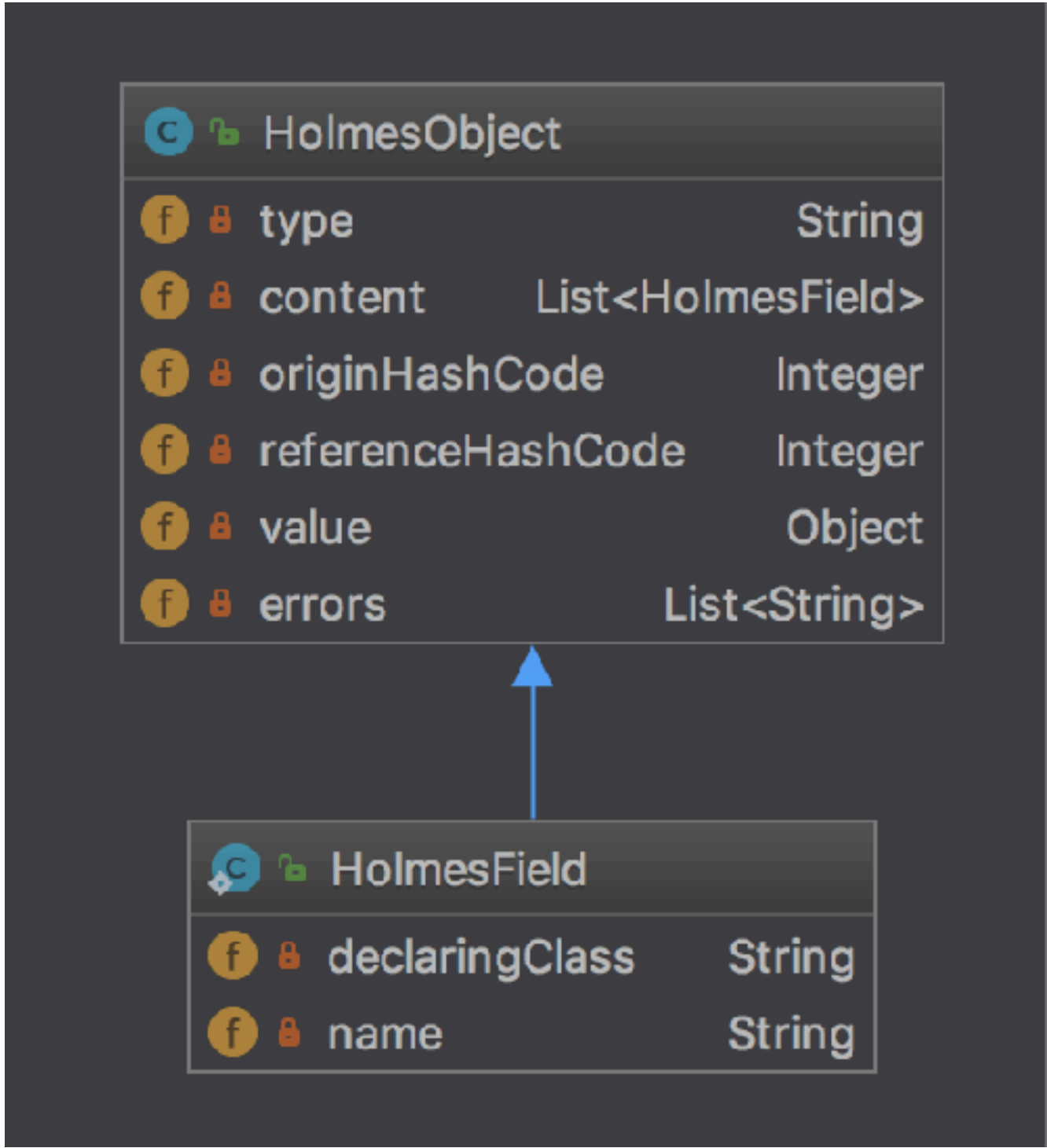
难点2: 对象快照

方案选择



难点2: 对象快照

自定义对象上报



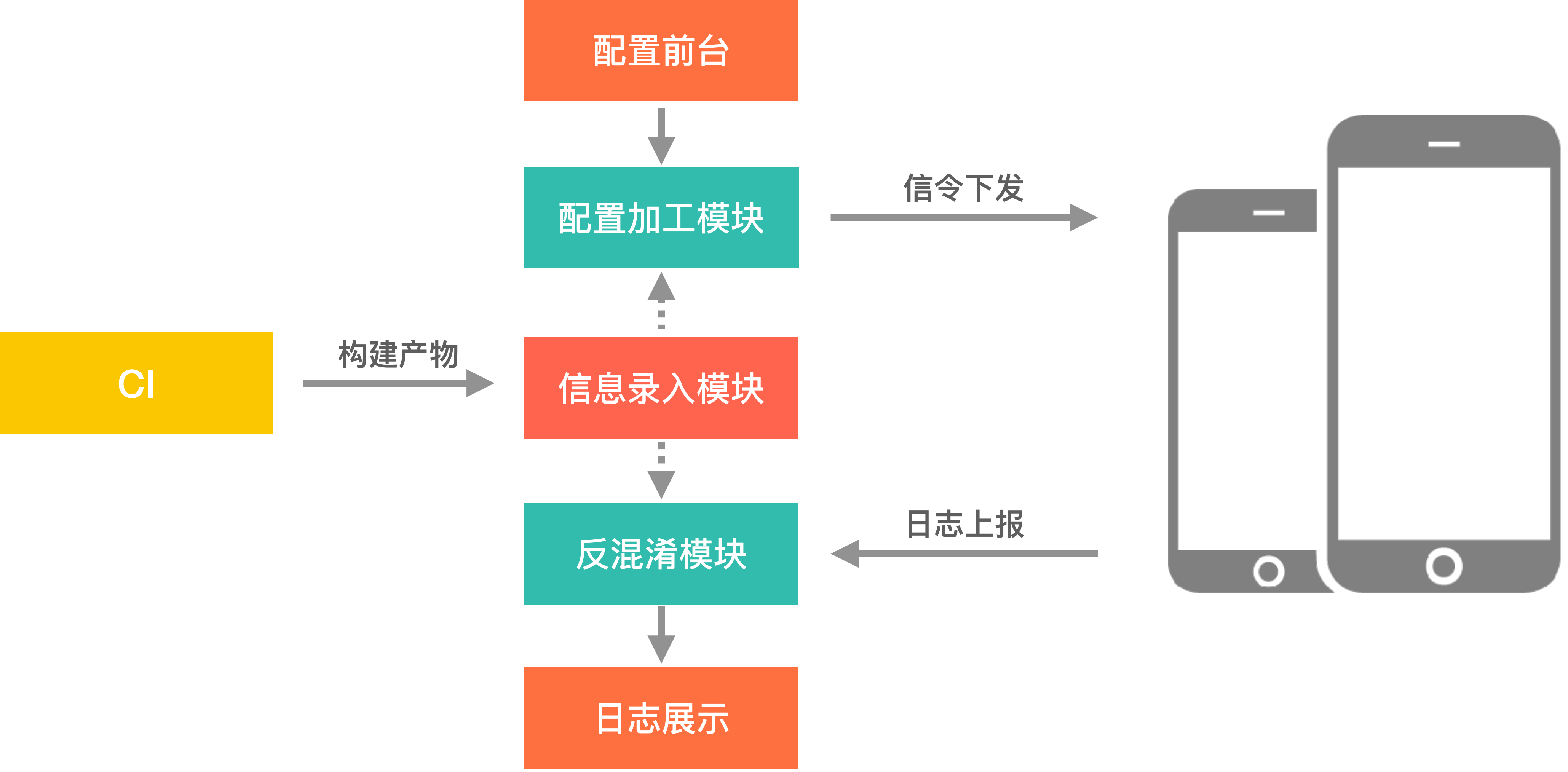
执行线程: [main-1](#)

▼ **this:** com.sankuai.meituan.setting>AboutMeituanActivity

- clickCount: int = 0
- MetaChannel: java.lang.String = 'undefined'
- updateClickable: boolean = true
- isActive: boolean = false
- mStateSaved: boolean = false
- mThemeld: int = 2131493062
- mCreated: boolean = false
- mNextCandidateRequestIndex: int = 0
- mReallyStopped: boolean = true
- mRequestedPermissionsFromFragment: boolean = false
- mResumed: boolean = false

难点3：易用性系统设计

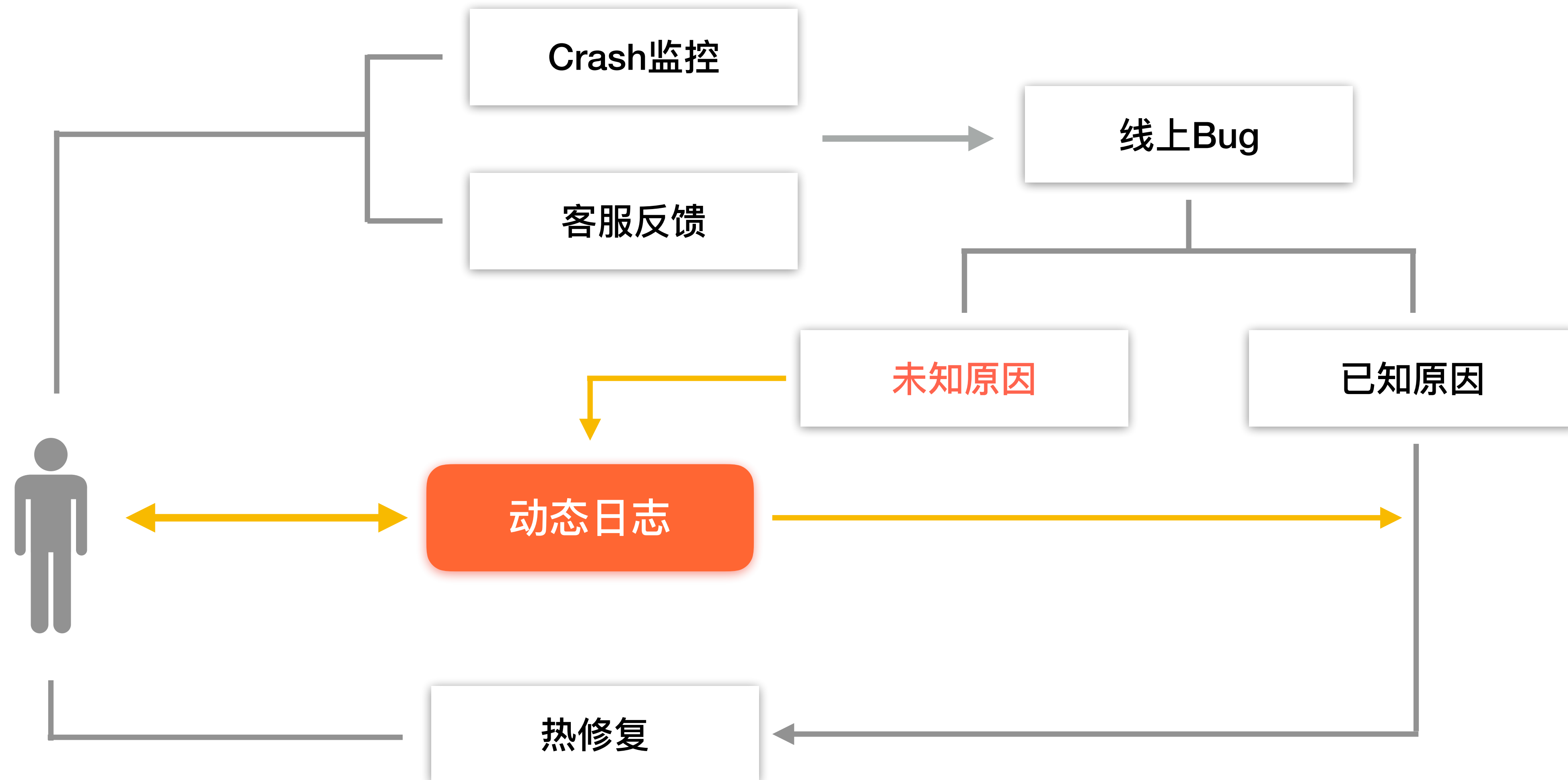
开发体系联动



总结与回顾

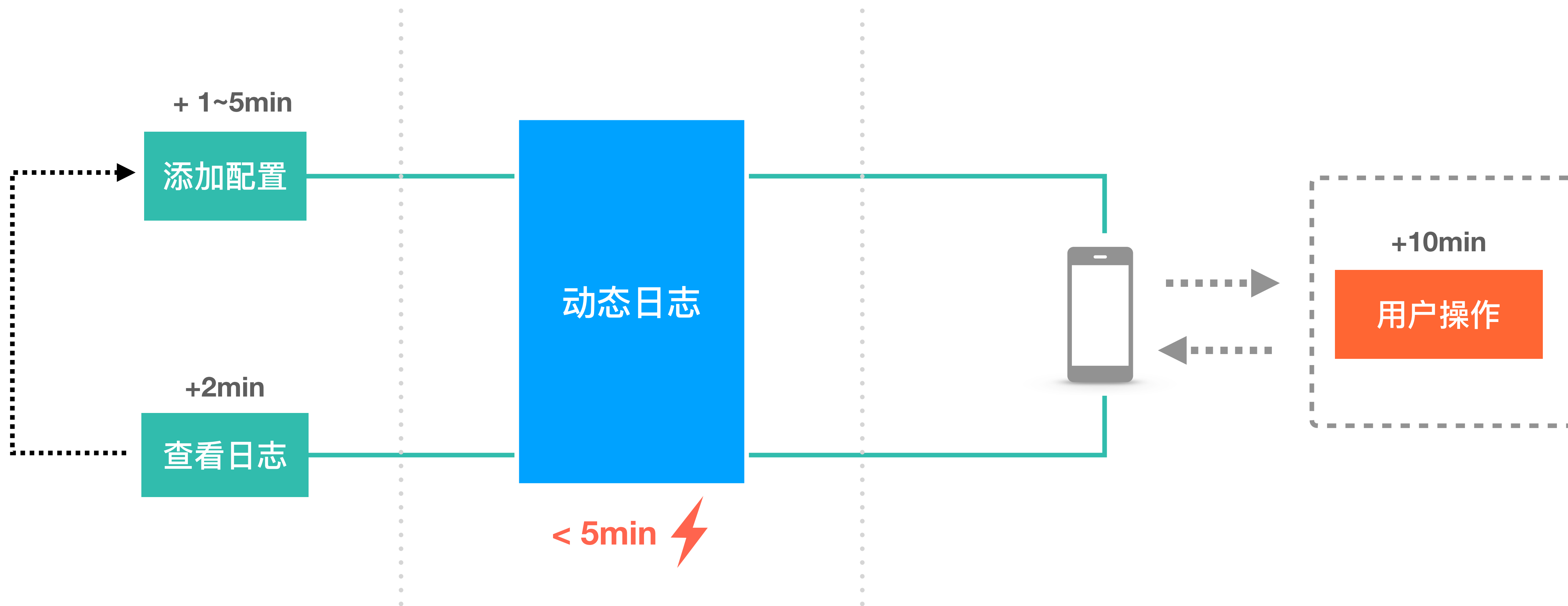
动态日志

线上排查体系



动态日志

效率提升



Q&A



扫码关注美团点评技术团队公众号
获取最 IN 的技术资讯

