

手机京东Crash自动分析处理系统

王永杰 · 京东



王永杰

- 国防科大学生，北交大硕士
- 首届GMTC大会特邀嘉宾
- 平台产品研发部技术专家
- 架构委员会主任架构师
- 手机京东开放平台负责人



1 背景介绍

2 系统架构

3 关键实践

4 未来方向



- **定义:**

影响用户数/DAU

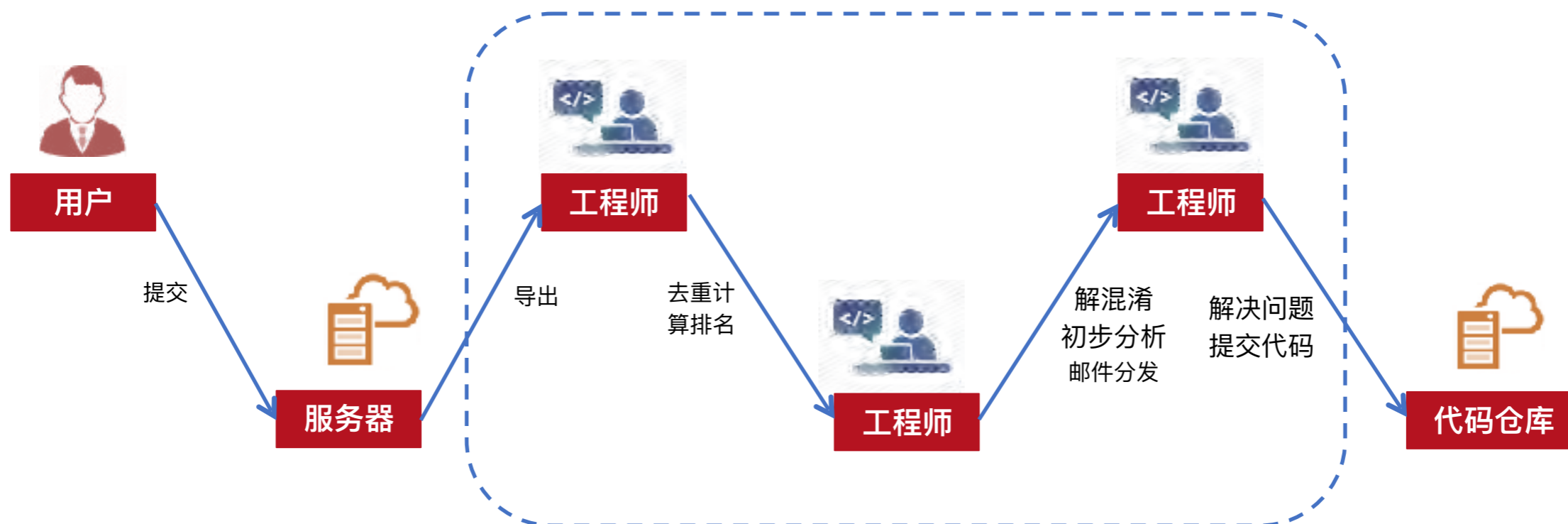
Crash数/启动次数

- **指标:**

百分位 => 千分位 => 万分位



人工处理流程



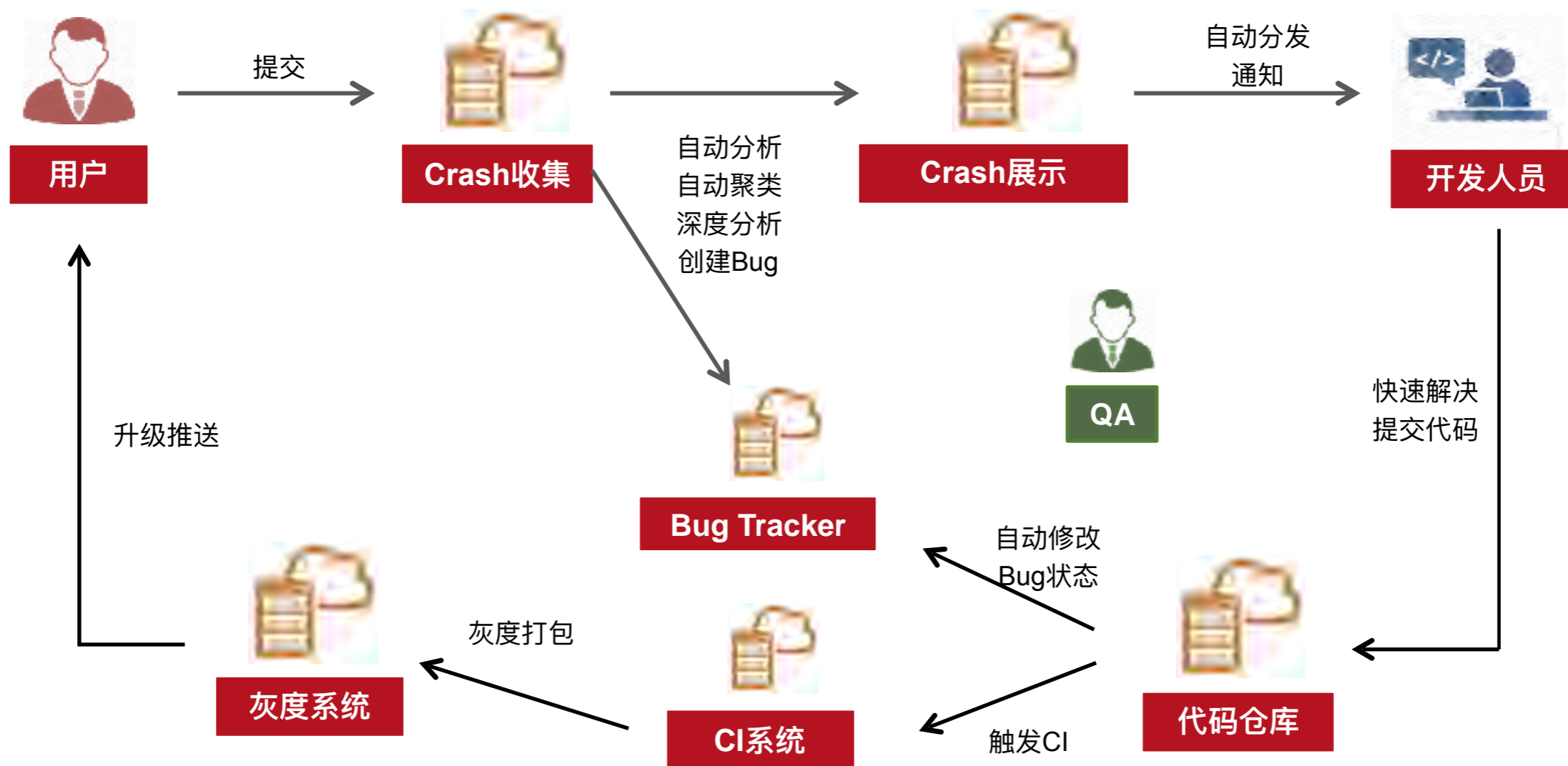
- 人工统计
- 人工分析
- 人工分派
- 灰度监控
- 日常监控

问题

- 繁琐
- 效率
- 遗漏
- 实时性
- 问题跟踪
- 统计
- 工程师
- 创造性
- 疑难问题
- 机器
- 不知疲倦
- 算力
- 不惧繁琐



自动化流程



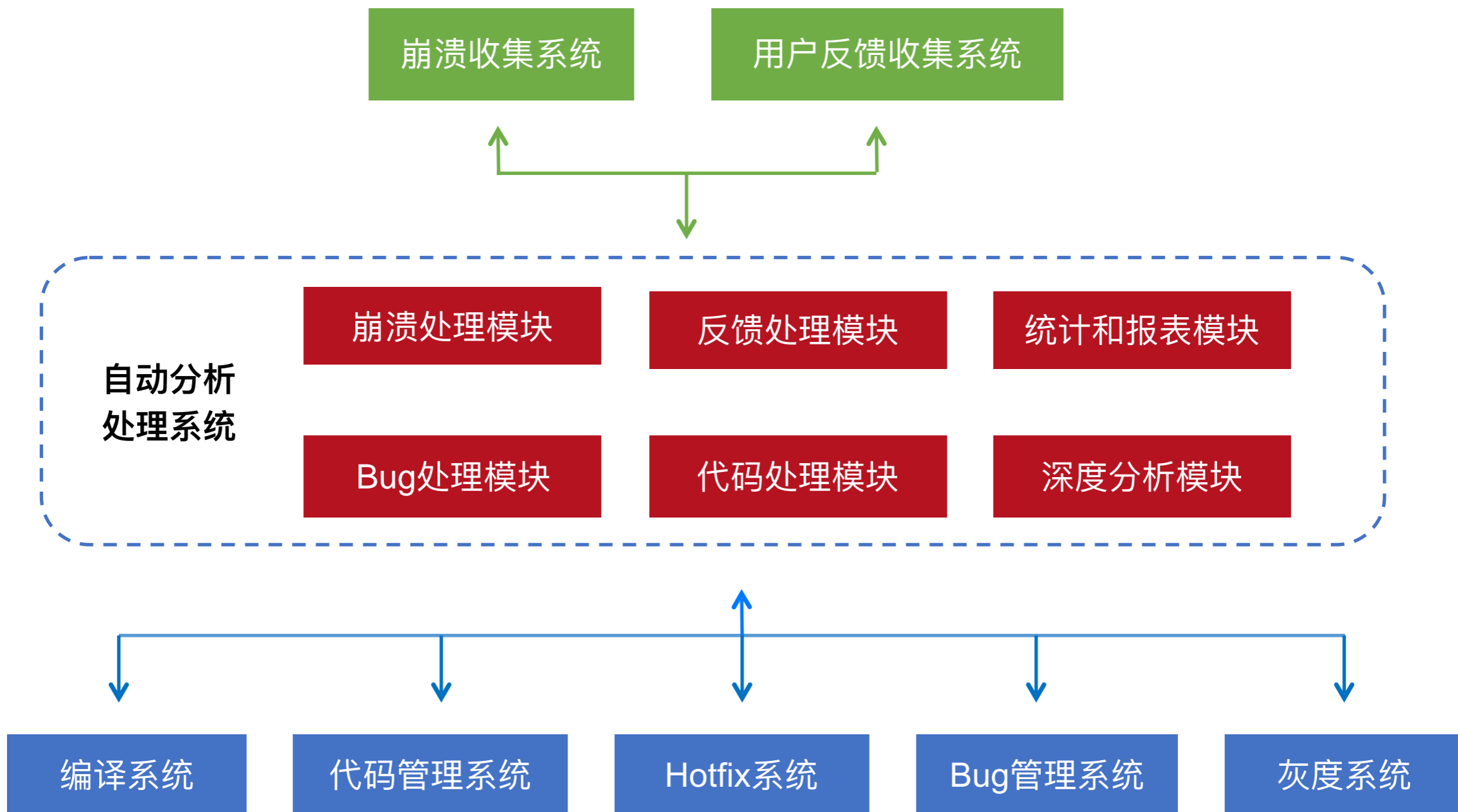


1 背景介绍

2 系统架构

3 关键实践

4 未来方向





总体架构





1 背景介绍

2 系统架构

3 关键实践

4 未来方向



崩溃分析结果-预览



Crash information:

Type: **java.lang.NullPointerException**
Description: Attempt to invoke interface method 'void java.util.List.add(int, java.lang.Object)' on a null object reference
Client Version: 5.4.0
Build number: 38386
Source: AndroidManifest.xml:android.support.design.widget.BottomNavigationView\$1.onNavigationItemSelectedListener:63
Branch or Tag: origin/V5.4.0
Stack:

代码和分支等信息

```
com.jingdong.com.MainActivity.toListActivity:(void onActivityResult(int,int,android.content.Intent)) [63]  
android.app.Activity.dispatchActivityResult(16602)  
android.app.ActivityThread.deliverResults(3774)  
android.app.ActivityThread.handleSendResult(3021)  
android.app.ActivityThread.access$1400(168)  
android.app.ActivityThread$H.handleMessage(47)  
android.os.Handler.dispatchMessage(47)  
android.os.Looper.loop(150)  
android.app.ActivityThread.main(5665)  
java.lang.reflect.Method.invoke(-2)  
com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(799)  
com.android.internal.os.ZygoteInit.main(689)
```

高亮出错文件、函数和行号

经过复原的崩溃堆栈

[More information details](#)

原始崩溃信息链接



崩溃分析结果-代码信息



Code:

用色彩格式化出错代码上下文

```
245     }
246
247     // String prop = ...;
248     // if (TextUtils.isEmpty(property)) {
249     //     property = ...;
250     // }
251
252     try {
253         ensureResourcesInjected(activity);
254         mBase.callActivityOnCreate(activity, bundle);
255         afterCallActivityOnCreate(activity);
256     } catch (Exception e2) {
257         // e2.printStackTrace();
258         if (!e2.toString().contains("android.content.res.Resources") || e2.toString().contains("OutOfMemoryError")) {
259             e2.printStackTrace();
260         }
261         HandleResourceNotFound(activity, bundle, e2);
262     }
263 }
```

高亮出错代码行

出错代码上下文行号

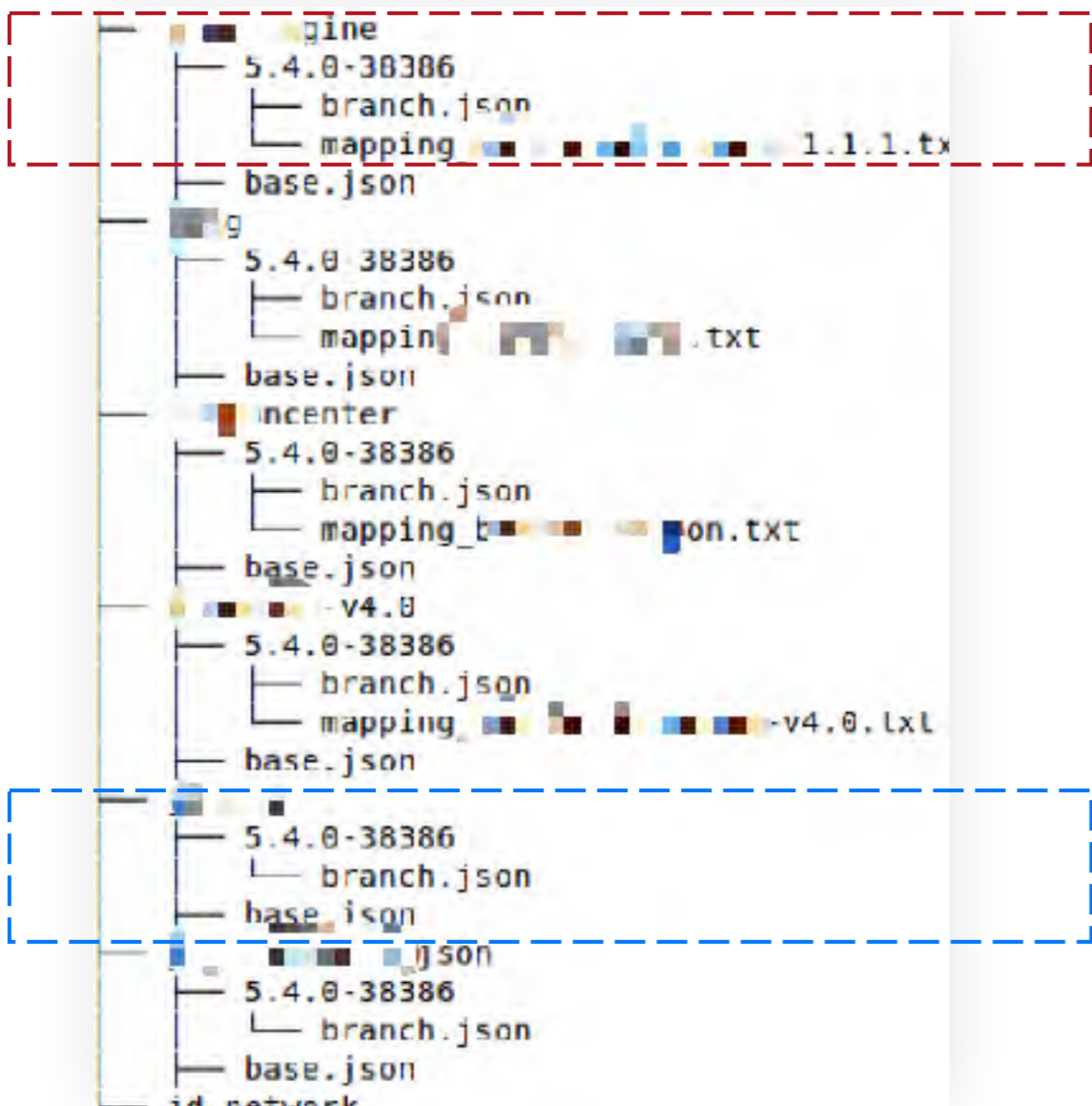


多库还原

- 主工程
- 插件
- jar包
- AAR
- Maven



多库还原配置



```
{  
  "branchOrTag": "VERSION"  
}
```

版本配置

```
com.jingdong.aa ↔ com.jingdong.Test1
```

Mapping文件

主工程/插件/Jar/Aar包 (混淆)

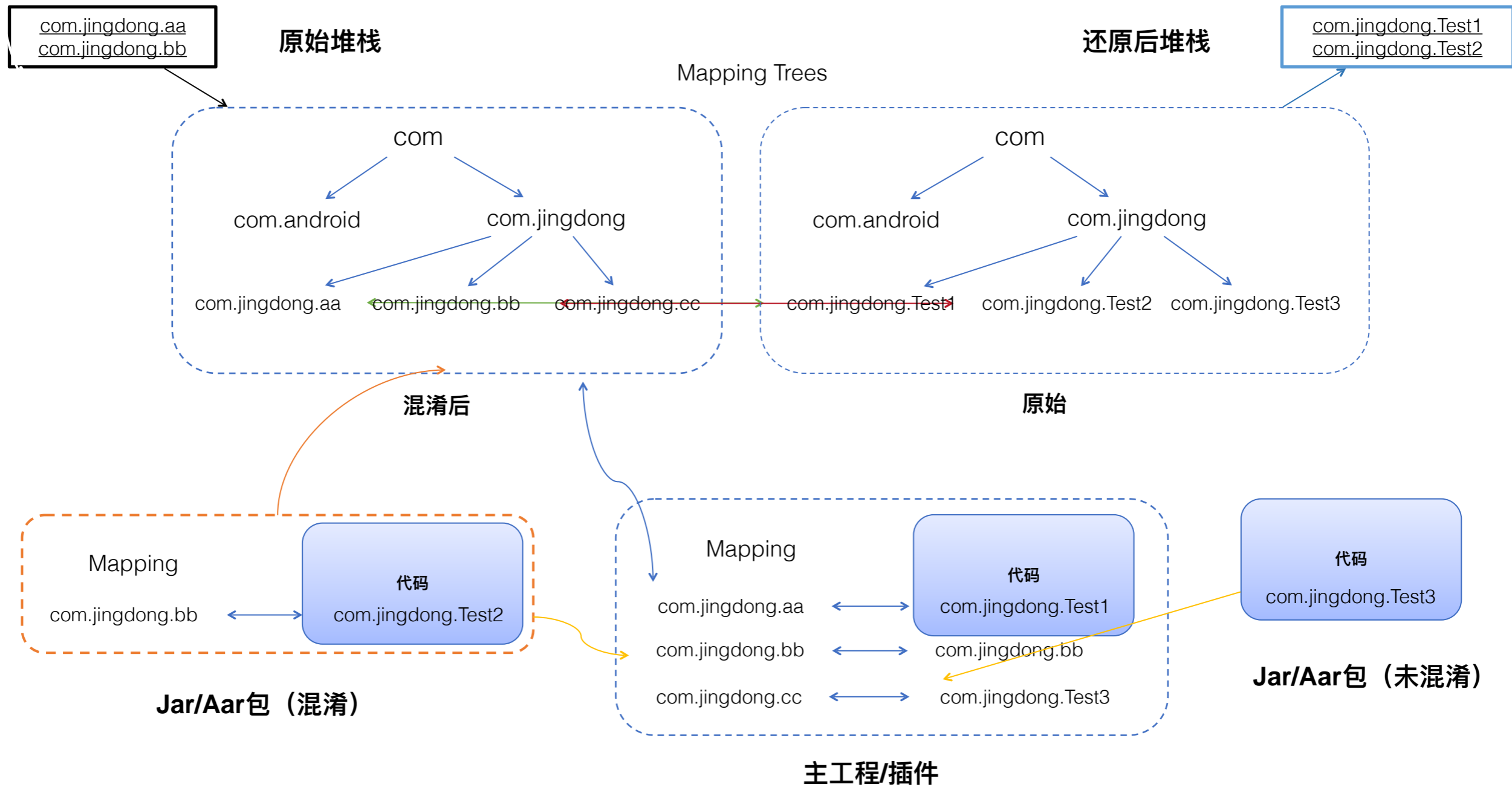
```
{  
  "branchOrTag": "VERSION"  
}
```

版本配置

Aar包 (未混淆)



多库还原过程



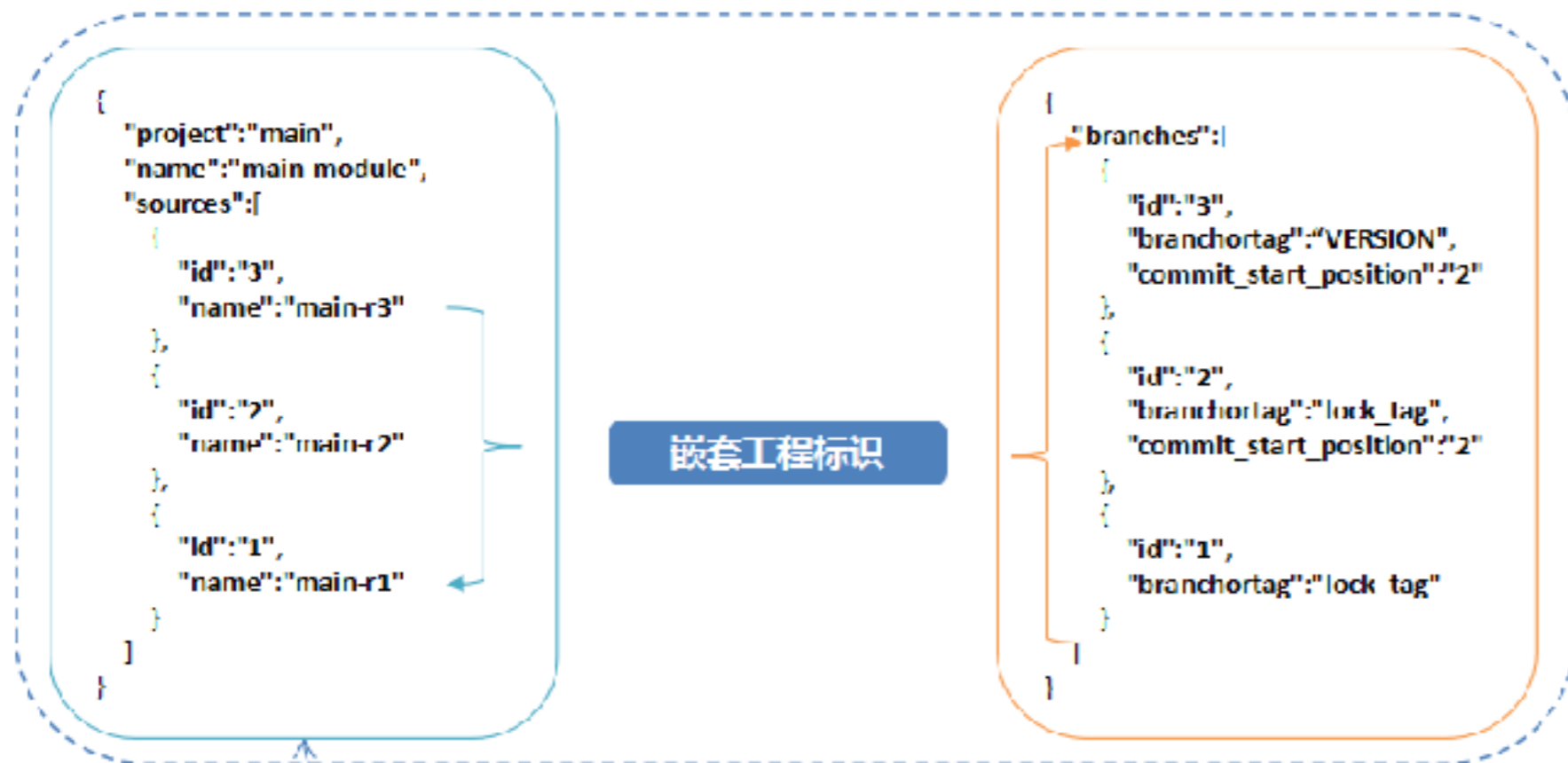


- **多代码库的问题：** 代码库迁移、拆分
- **文件路径变更：** 源代码修改了目录或名称。

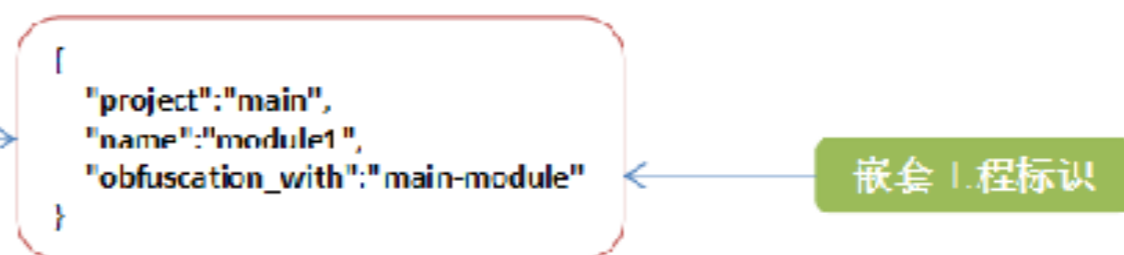


代码库的配置方式

多代码仓库

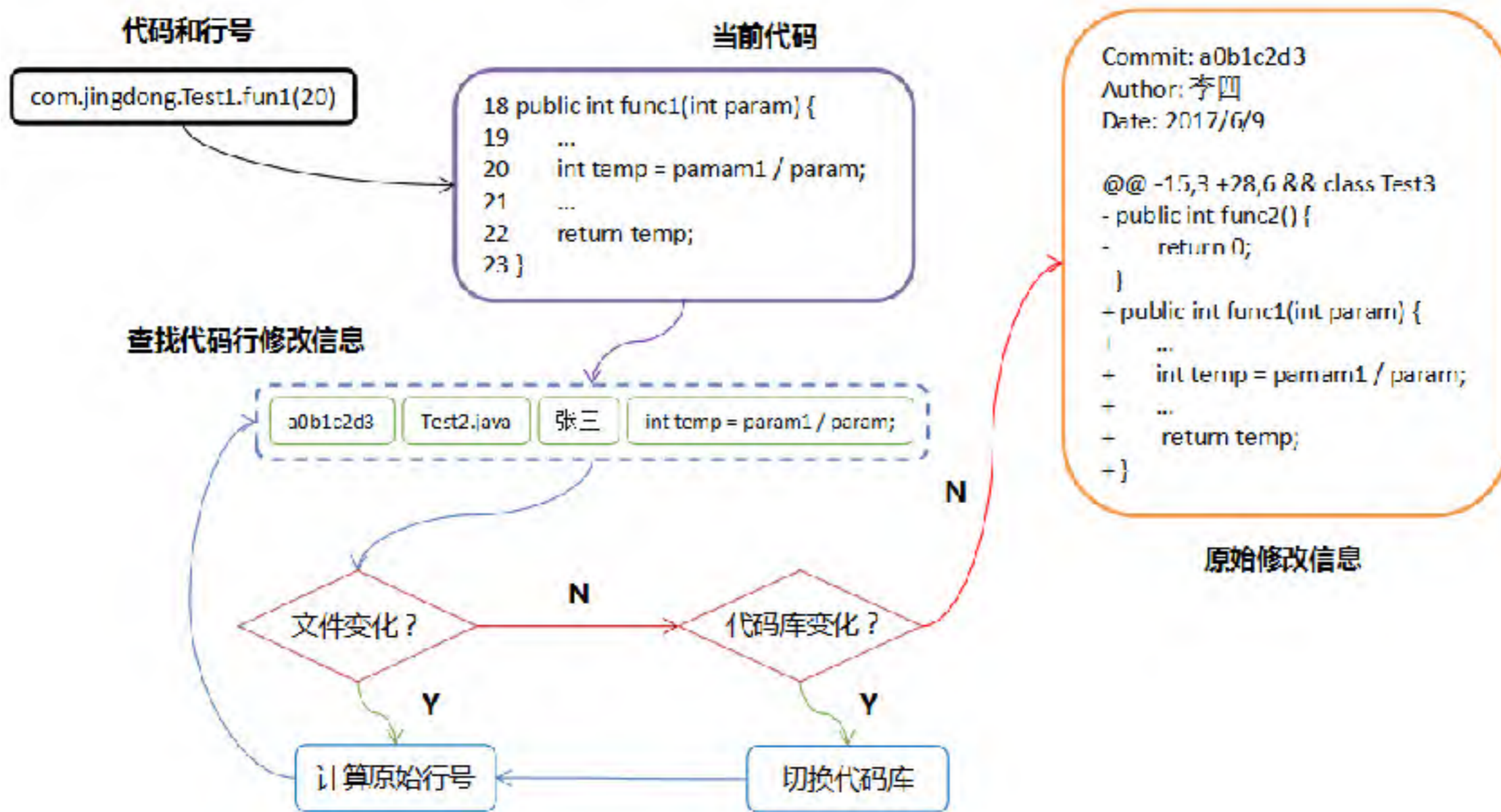


单代码仓库





代码追踪过程





1 背景介绍

2 系统架构

3 关键实践

4 未来方向



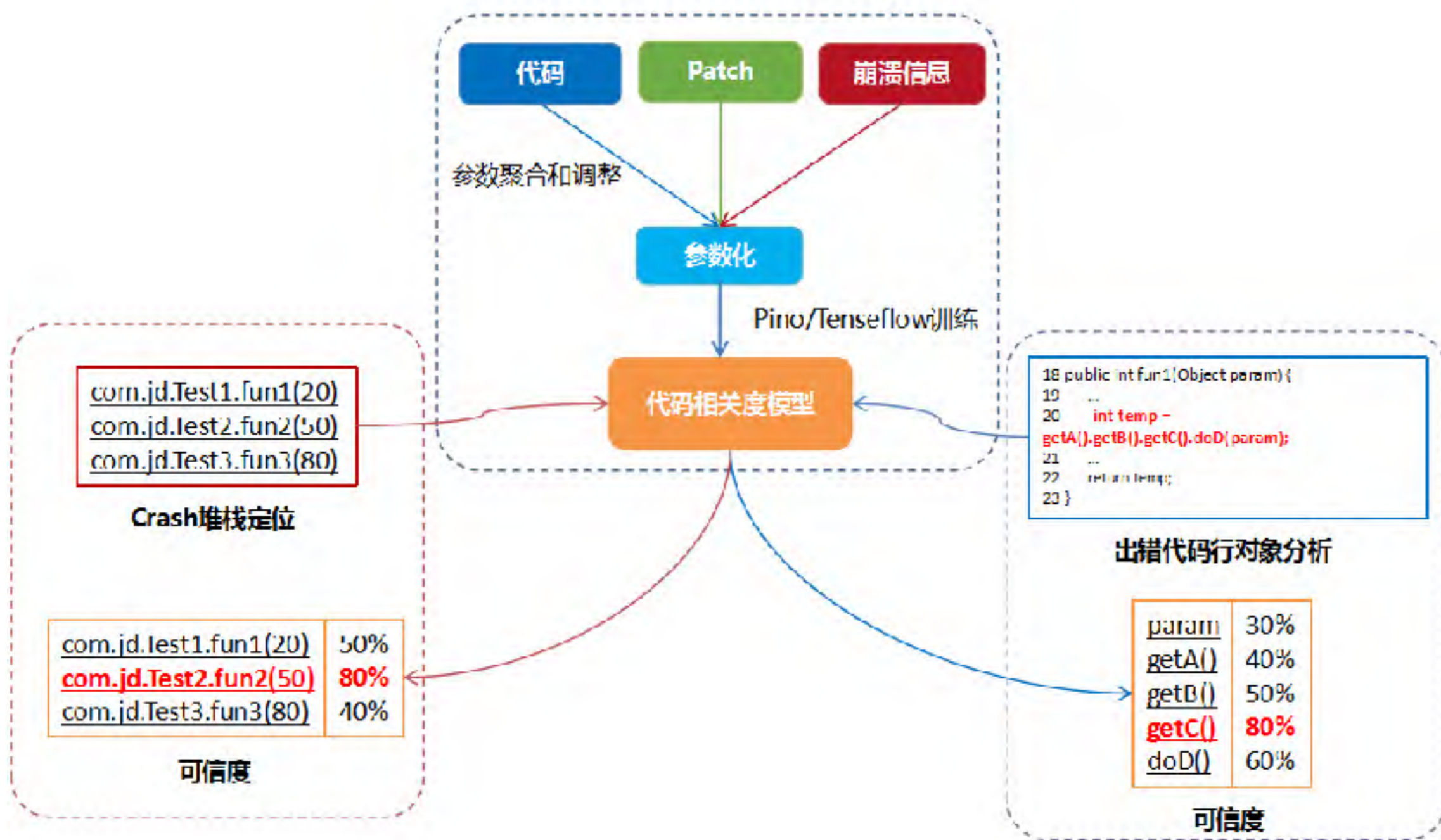
自动化



智能化



Crash深度分析





- **模型建立:**

通过机器学习对代码, patch和崩溃信息, 建立代码相关度模型。

- **Crash分析:**

Crash堆栈定位

出错代码行对象分析



THANKS!

GMTC
全球移动技术大会2017