

Freeline

蚂蚁聚宝Android秒级编译方案

SPEAKER

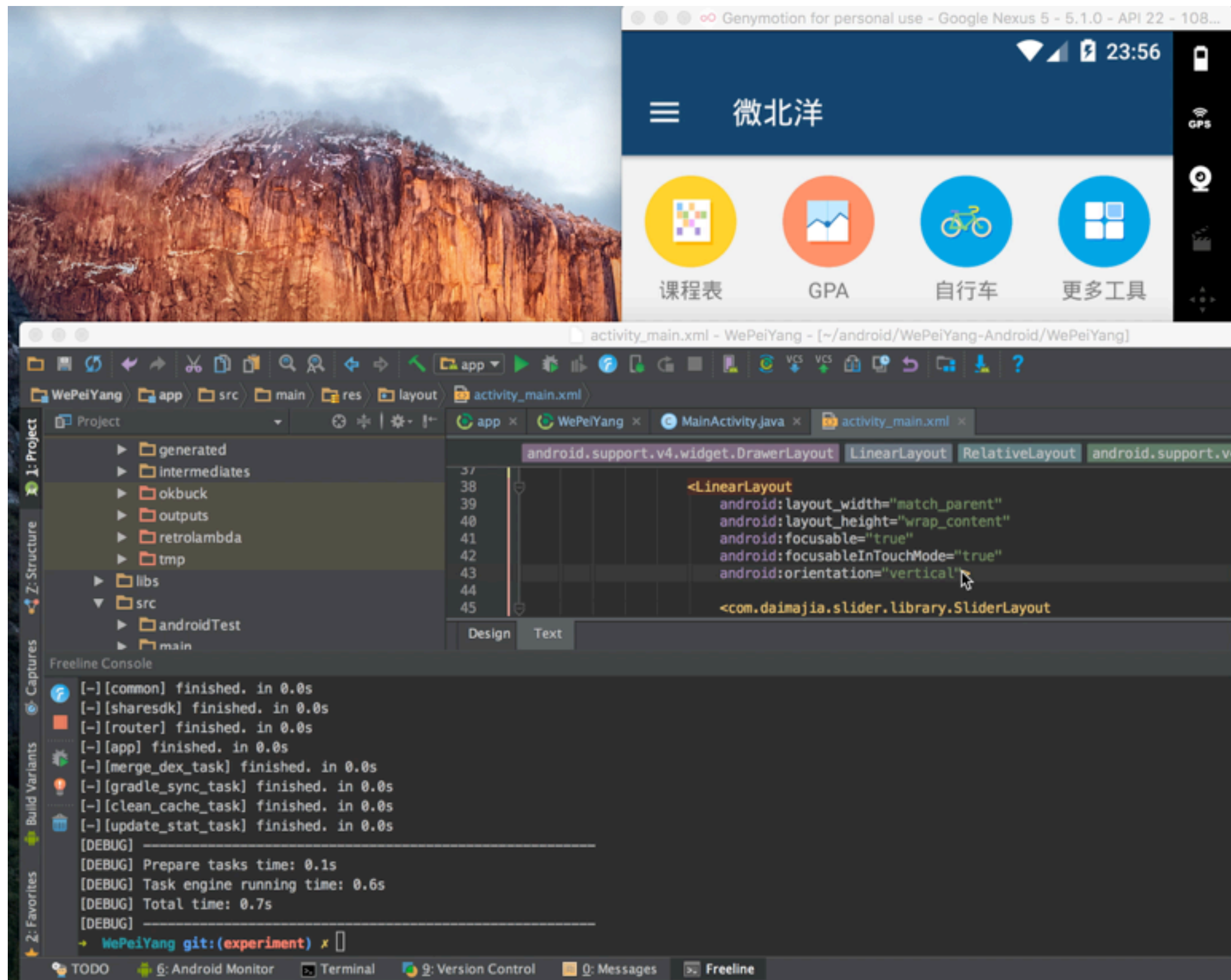
何嘉文(弦影)

个人介绍



- 蚂蚁金服高级技术专家，蚂蚁聚宝基础架构负责人，7年一线开发,先后于UC,微信任技术骨干
- 亿级客户端架构及性能优化经历
- 专注领域：终端存储，网络通讯，性能，基础架构

Freeline是什么?



Buck

多任务并发

多级缓存

```
$ buck build buck
[-] PARSING BUILD FILES...FINISHED 0.6s
[+] BUILDING...2.3s
  |=> IDLE
  |=> //src/com/facebook/buck/graph:graph... 1.8s (running javac[1.8s])
  |=> //src/com/facebook/buck/util:exceptions... 1.9s (running javac[1.8s])
  |=> //src/com/facebook/buck/util/concurrent:concurrent... 1.8s (running javac)
  |=> //third-party/java/aosp:aosp... 2.1s (running javac[2.0s])
  |=> //src/com/facebook/buck/util/hash:hash... 1.8s (running javac[1.8s])
  |=> //src/com/facebook/buck/util/environment:environment... 1.8s (running jav)
  |=> //src/com/facebook/buck/util/unit:unit... 0.6s (running javac[0.6s])
  |=> //src/com/facebook/buck/dalvik:dalvik_stats_tool... 1.8s (running javac[1])
  |=> //src/com/facebook/buck/util/collect:collect... 0.6s (running javac[0.6s])
  |
```

A

全量构建利器

B

入侵性强，接入成本高
需要安装全量包
以module为单位做增量

LayoutCast

优势：

基于Dex、Res动态替换
国内首个基于动态替换的开源编译方案

缺陷：

资源全量替换，没有实现增量
资源不支持新增
没有利用缓存，并发
反射生成public文件性能低
不支持4.x

Instant-Run

优势：

谷歌官方方案
通过HACK Method 实现代码增量替换

缺陷：

性能低
资源全量替换，慢
增量模式下不支持debug
多module工程结构支持不够完善

Freeline核心思想

多任务并发

多级缓存

增量范围最小化

懒加载

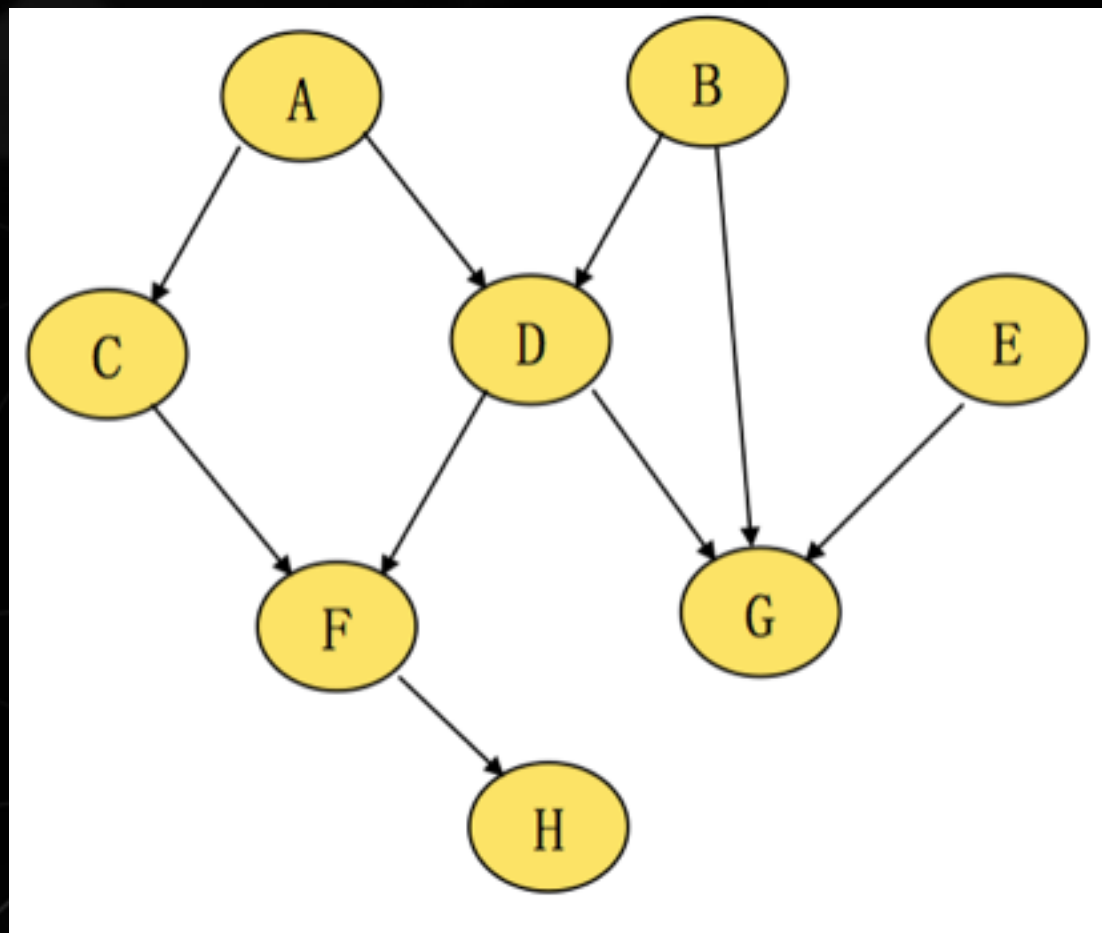
基于长链接无安装式运行期动态替换

基线对齐触发机制

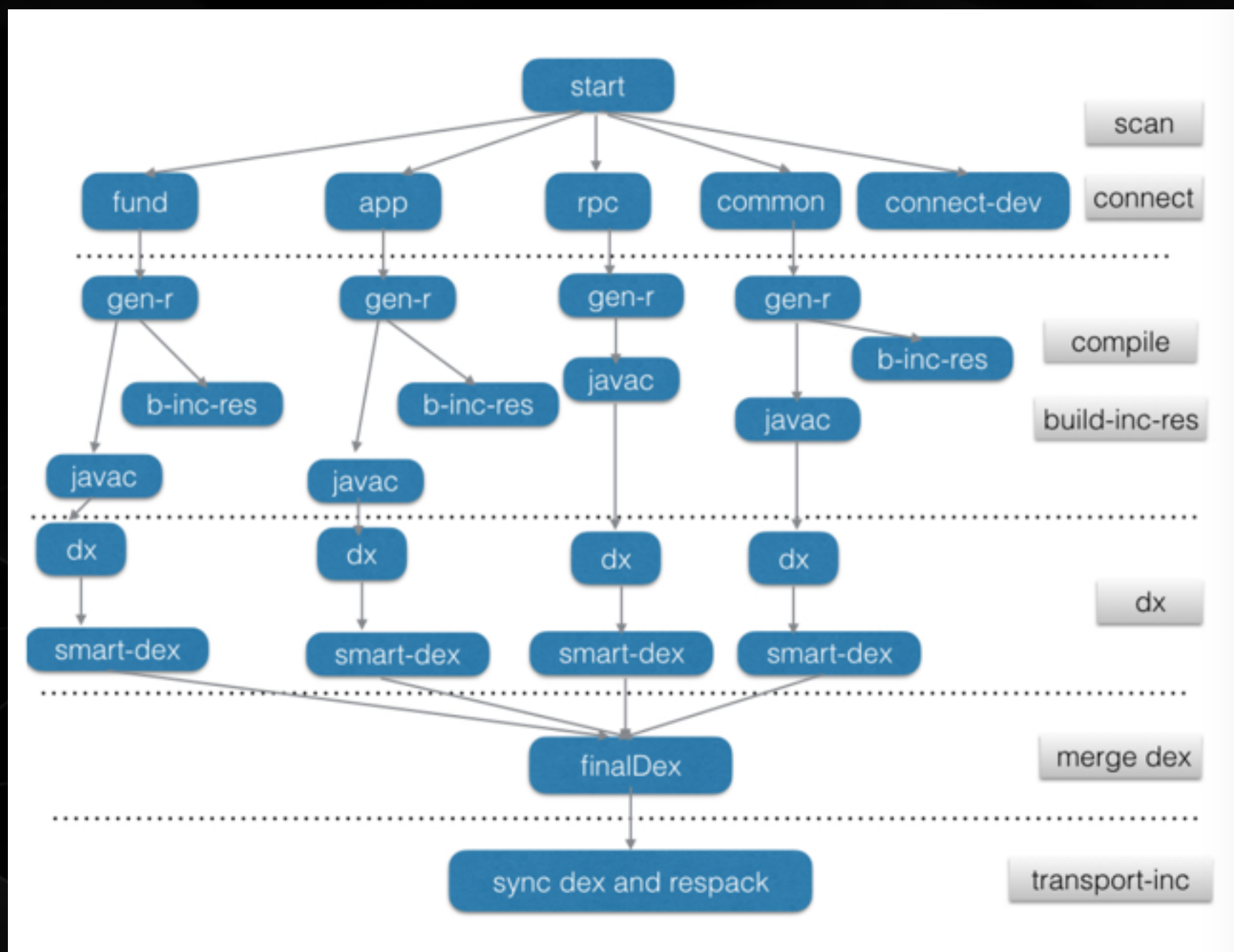
可调试

独立进程

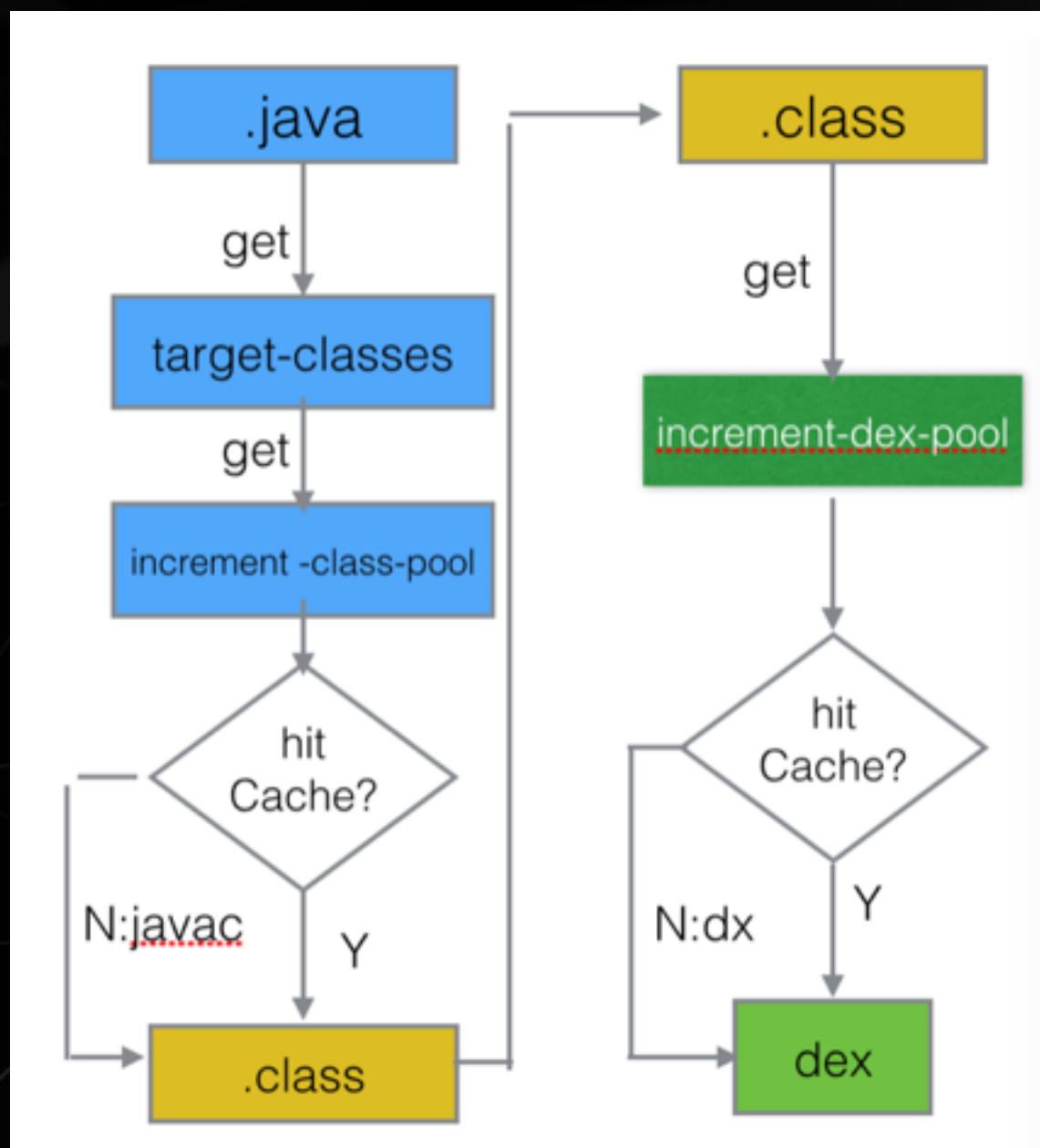
多任务并发 (1)



多任务并发 (2)

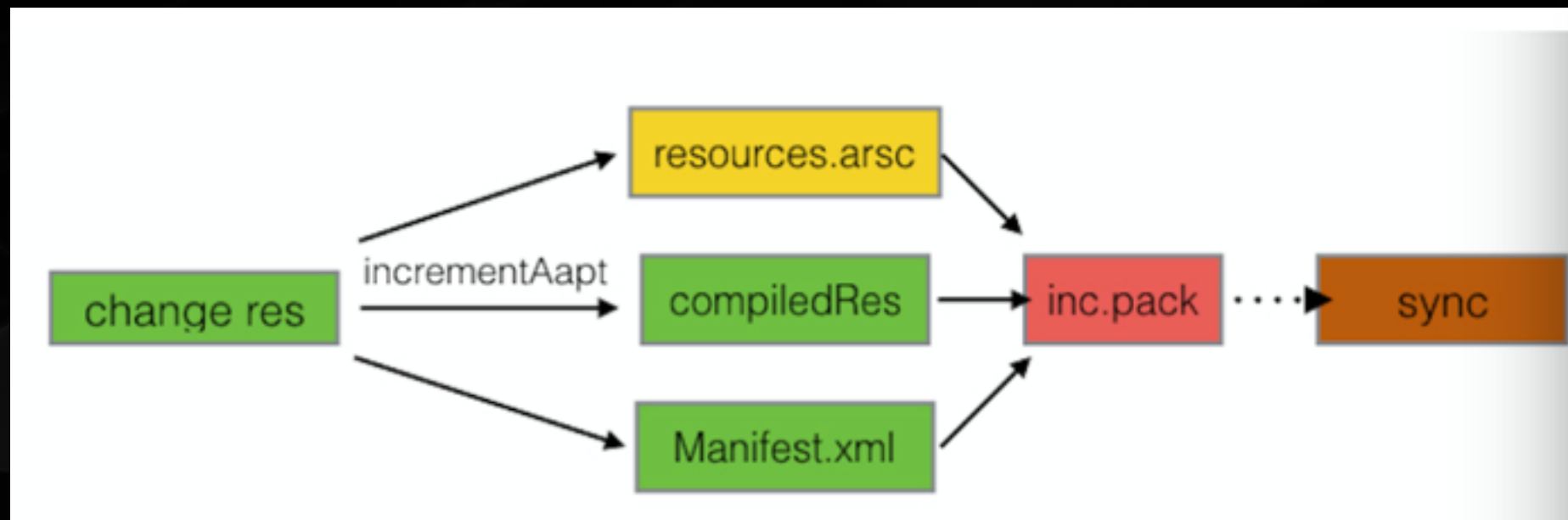


多级缓存-代码

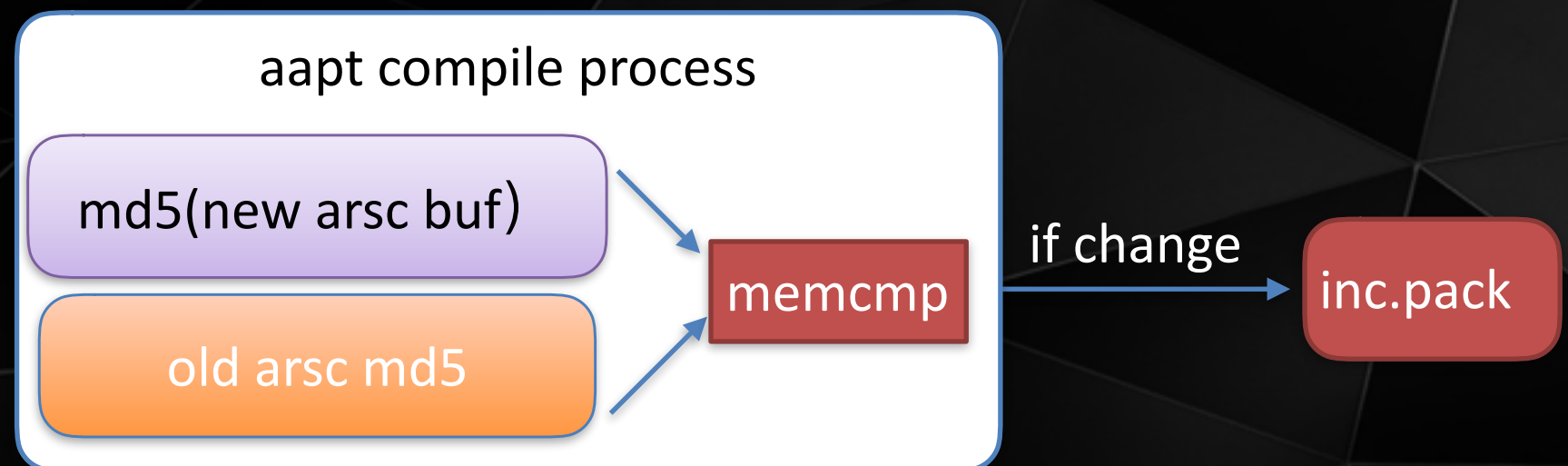


多级缓存-资源

一. 单次修改，单次清除



二. resource.arsc 缓存



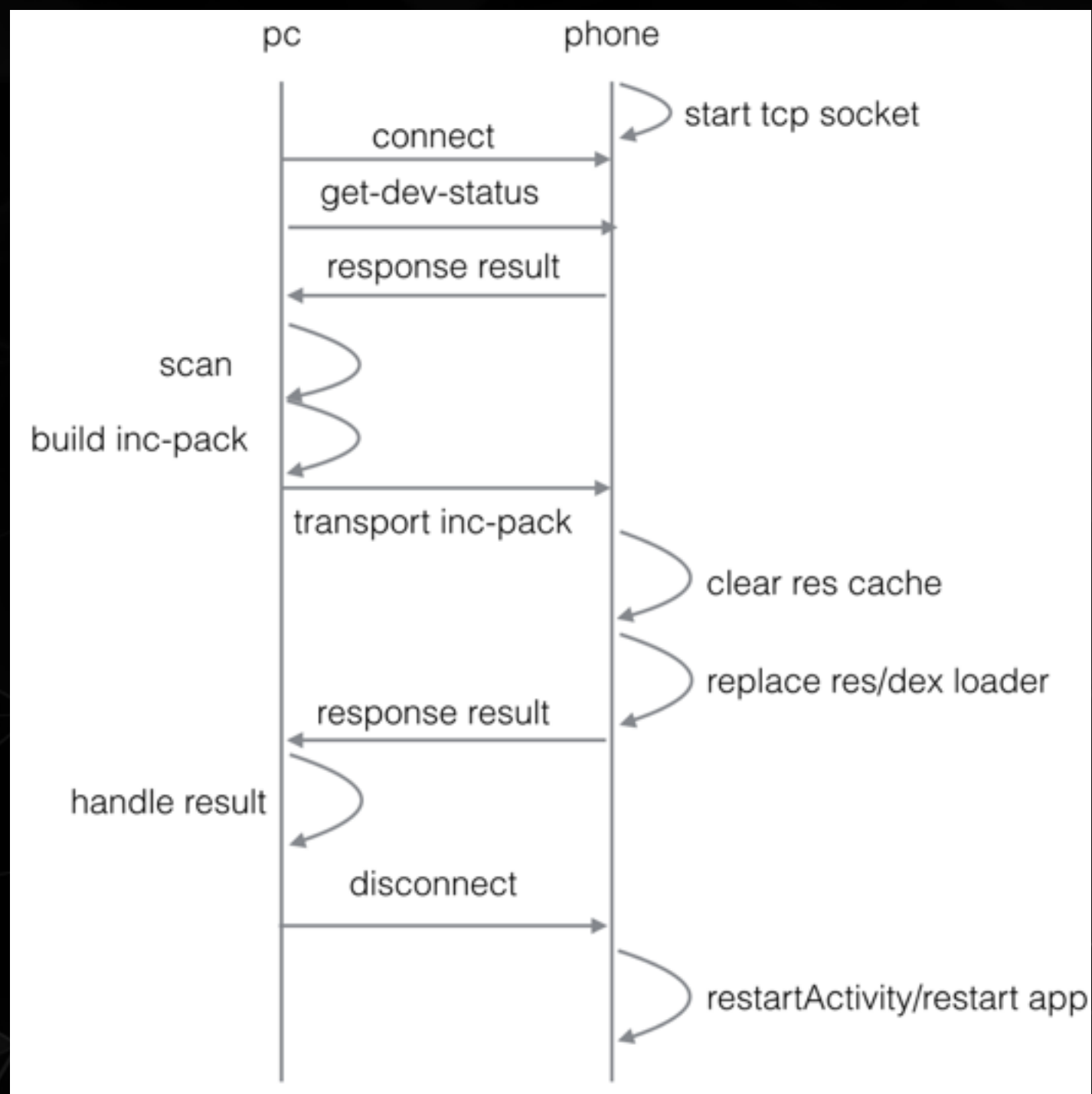
懒加载与可调试性

懒加载的实现

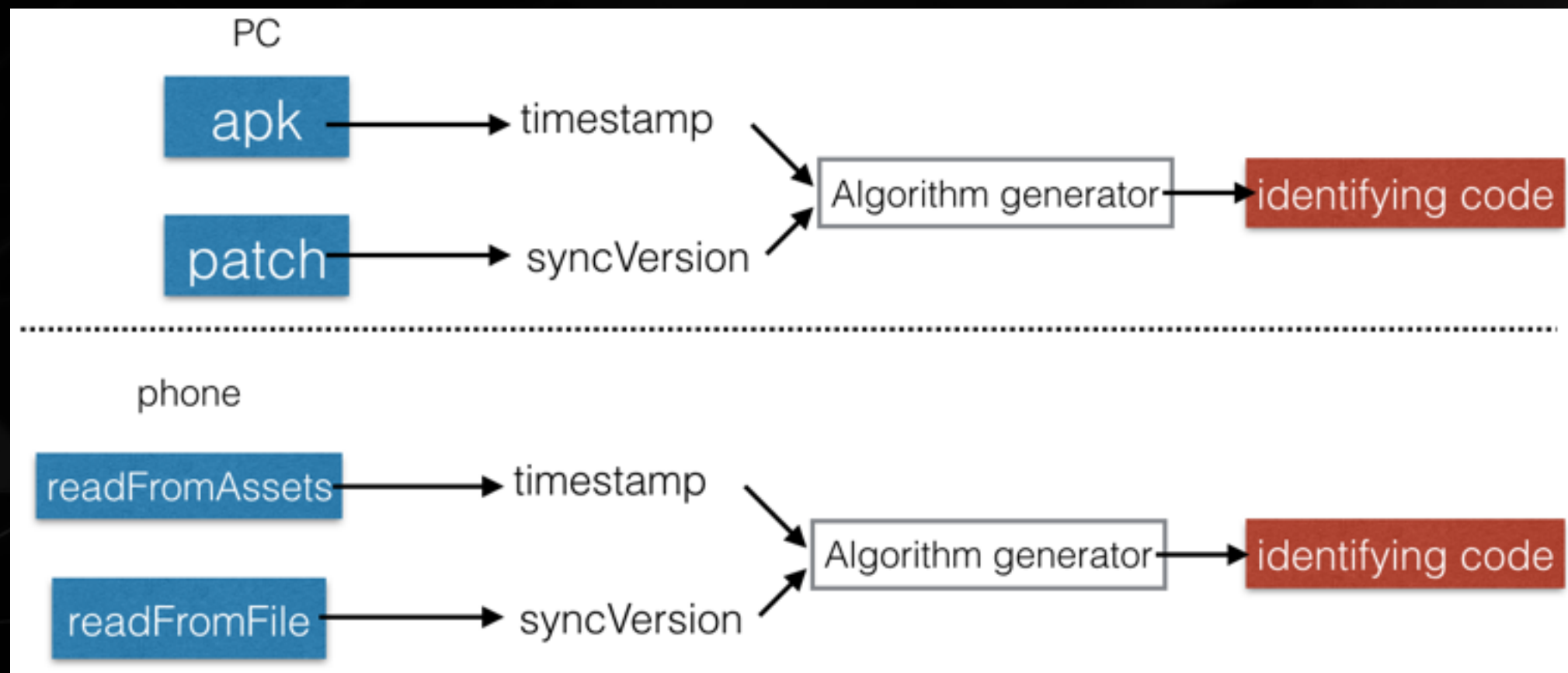
R.java 的编译时机

可调试怎么实现

基于长连接无安装式动态替换



基线对齐



进程隔离

隔离目的

隔离范围

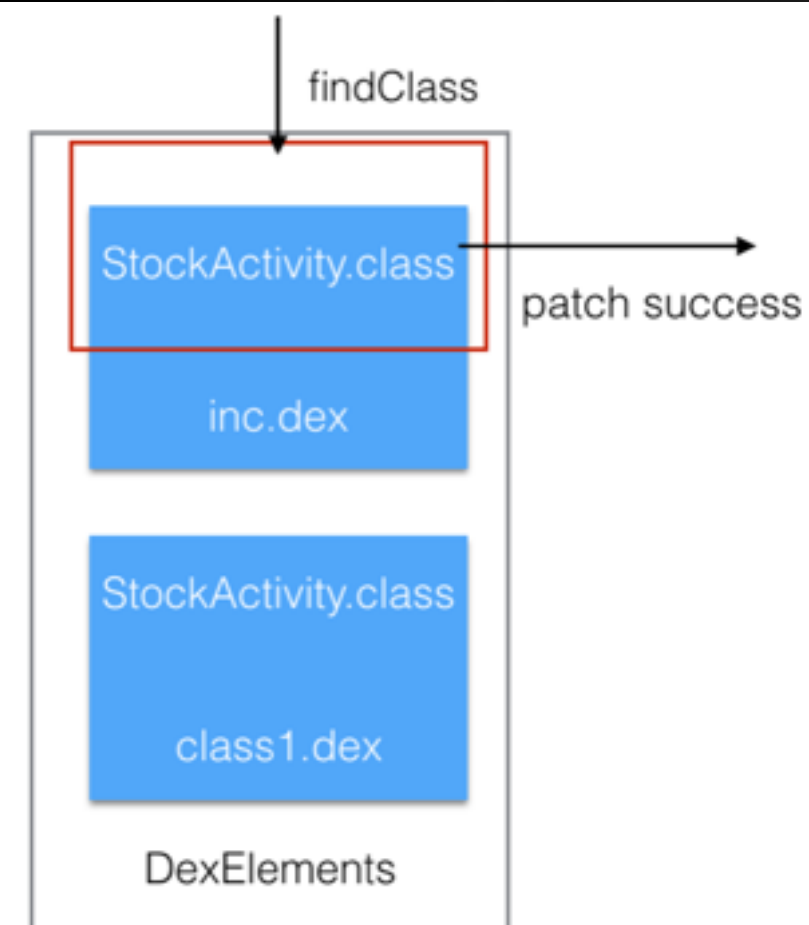
socket 建连接，传输，基线对齐

技术原理-代码增量

```
> | DexPathList.java > M findClass(String name, List<Throwable> suppressed)
String Matching

*/
public Class findClass(String name, List<Throwable> suppressed) {
    for (Element element : dexElements) {
        DexFile dex = element.dexFile;

        if (dex != null) {
            Class clazz = dex.loadClassBinaryName(name, definingContext, suppressed);
            if (clazz != null) {
                return clazz;
            }
        }
    }
    if (dexElementsSuppressedExceptions != null) {
        suppressed.addAll(Arrays.asList(dexElementsSuppressedExceptions));
    }
    return null;
}
```

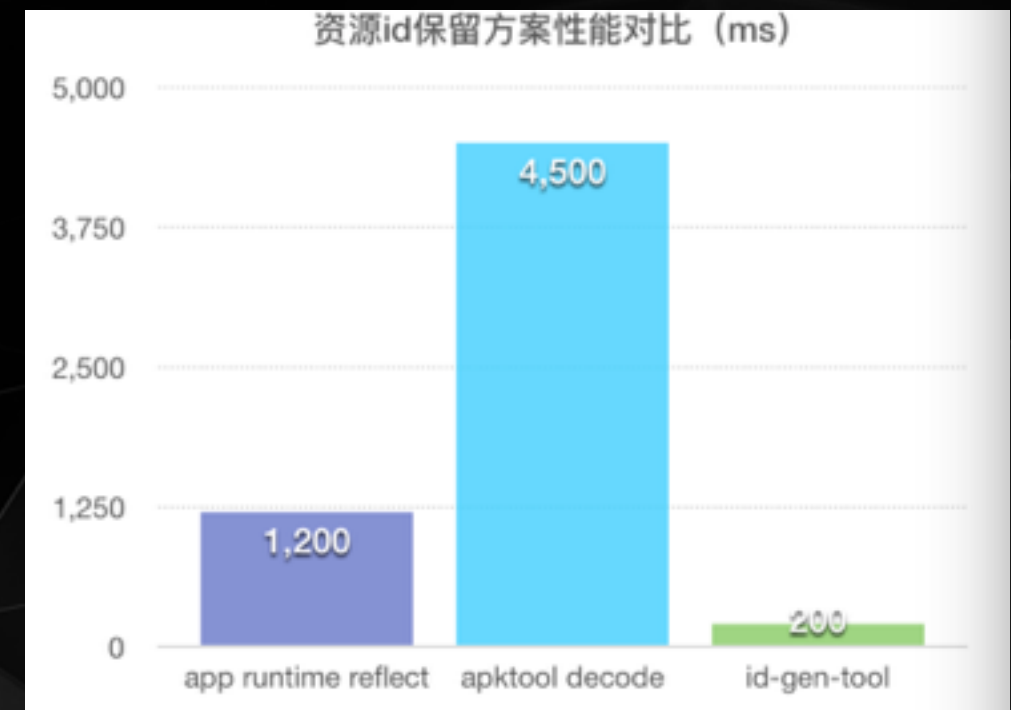
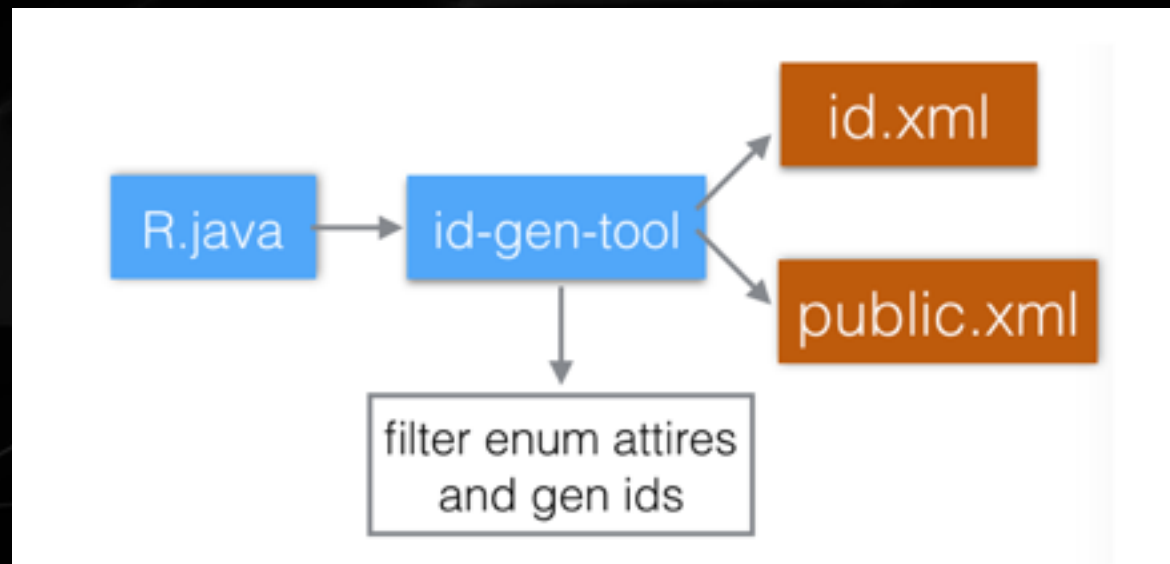


技术原理-资源增量-技术难点

- 1.增量包资源id怎么兼容基线包资源id?
- 2.怎么样高效构建出仅仅包含变更集合的资源包?
- 3.怎么样在手机端让上面构建的增量包生效?

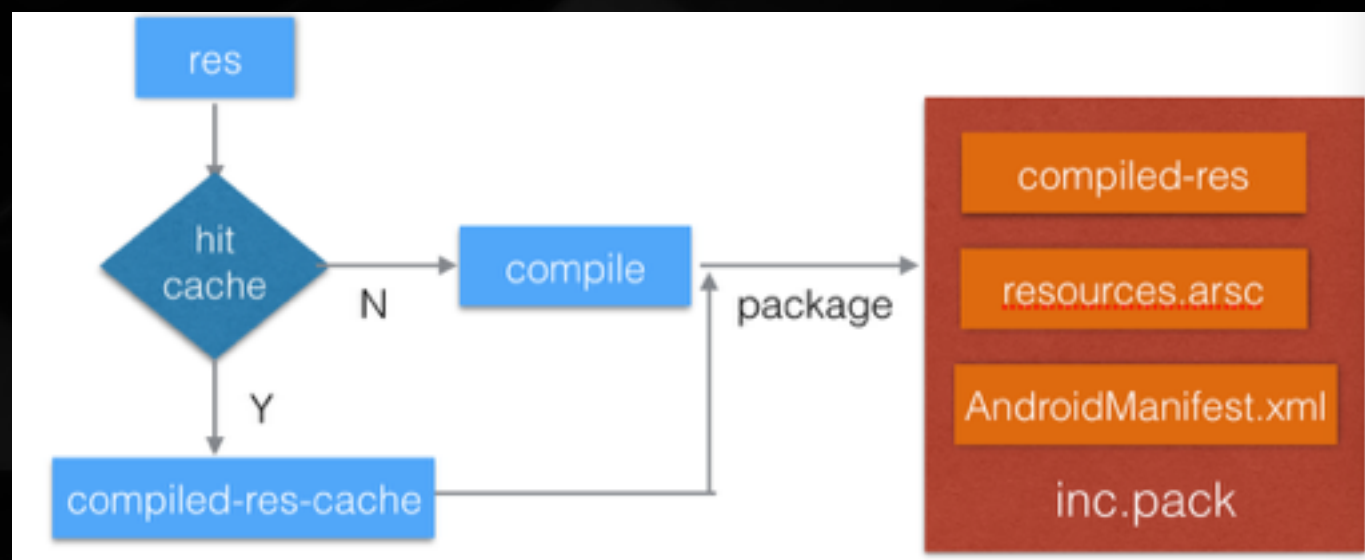
技术原理-资源增量

增量包资源id怎么兼容基线包资源id?



技术原理-资源增量

怎么样高效构建出仅仅包含变更集合的资源包？



```
1  <  >  aapt4.4.2  aapt4.4.2  aapt  Package.cpp  No Selection
2  nd  Q String Matching
3
4  1  /*
5  2  bool processFile(Bundle* bundle, ZipFile* zip,
6  3  const sp<AaptGroup>& group, const sp<AaptFile>& file)
7  4  {
8  5  const bool hasData = file->hasData();
9  6  // fprintf(stderr, "processFile file %s \n", (file->getPath().string()));
10 7
11 8  if(gChangeList){
12 9  if (!strstr(file->getPath().string(), "resources.arsc") && !strstr(file->getPath().string(), "AndroidManifest.xml") && !strstr(gChangeList, file->getPath().string())){
13 10  fprintf(stderr, "find change file %s \n", (file->getPath().string()));
14 11  return true;
15 12  }
16 13  }
17 14  }
```



图4 Android资源打包工具的执行过程

技术原理-资源增量-Aapt流程优化

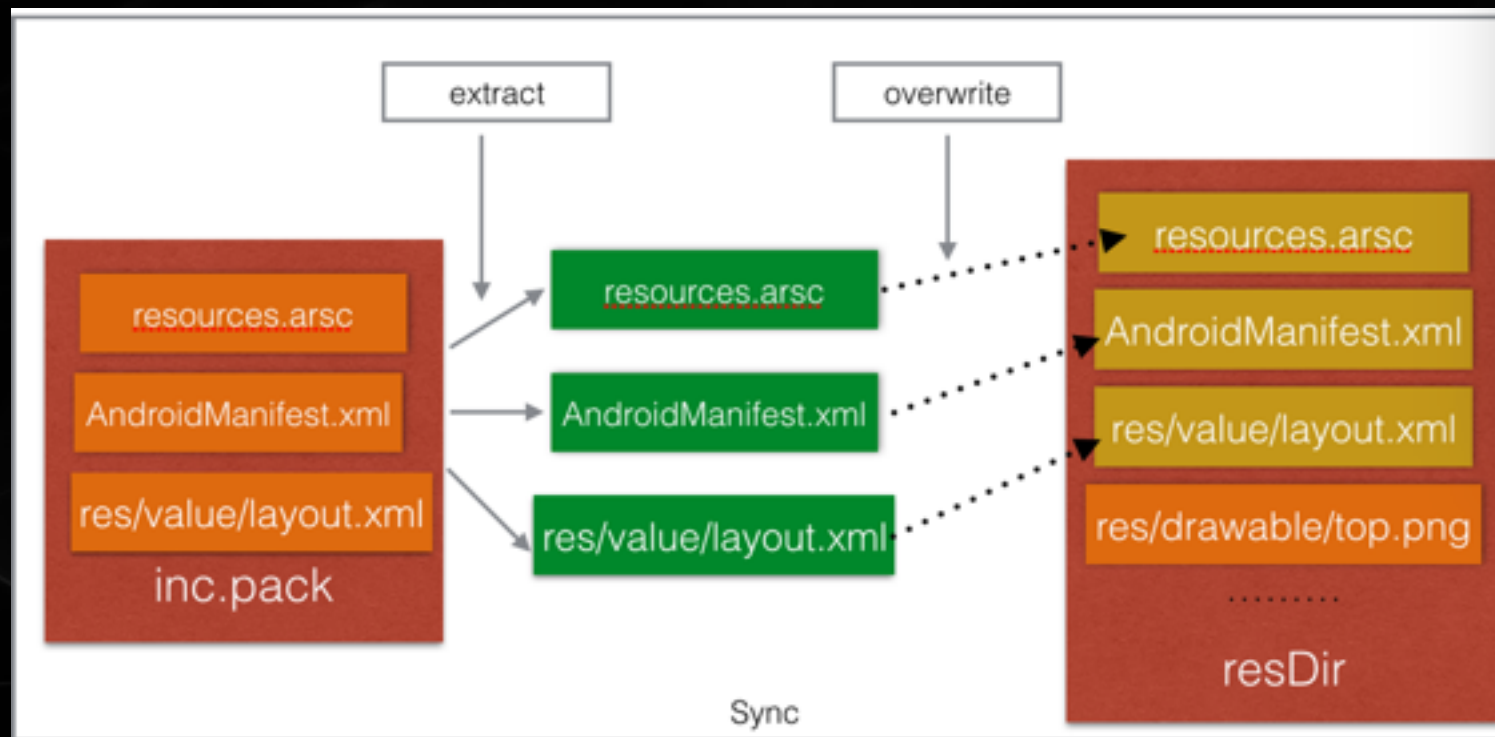
full-build

```
--- command 6 (verbose=0 force=1):  
Freeline trace:: slurp up res last 34.786000 ms  
Freeline trace:: makeFileResources all resource. Time Elapsed: 2518.108000 ms  
Freeline trace:: compile values. Time Elapsed: 250.162000 ms  
Freeline trace:: makeFileResources for color and menu. Time Elapsed: 0.225000 ms  
Freeline trace:: generate all bag attributes and assign indices. Time Elapsed: 62.970000 ms  
Freeline trace:: compile all xml. Time Elapsed: 376.342000 ms  
Freeline trace:: flatten gen resources.arsc Time Elapsed: 59.713000 ms  
Freeline trace:: gen r file last 24.591000 ms  
Freeline trace:: APK Bundling. Time Elapsed: 179.372000 ms  
Freeline trace:: build pack last 3543.390000 ms in total
```

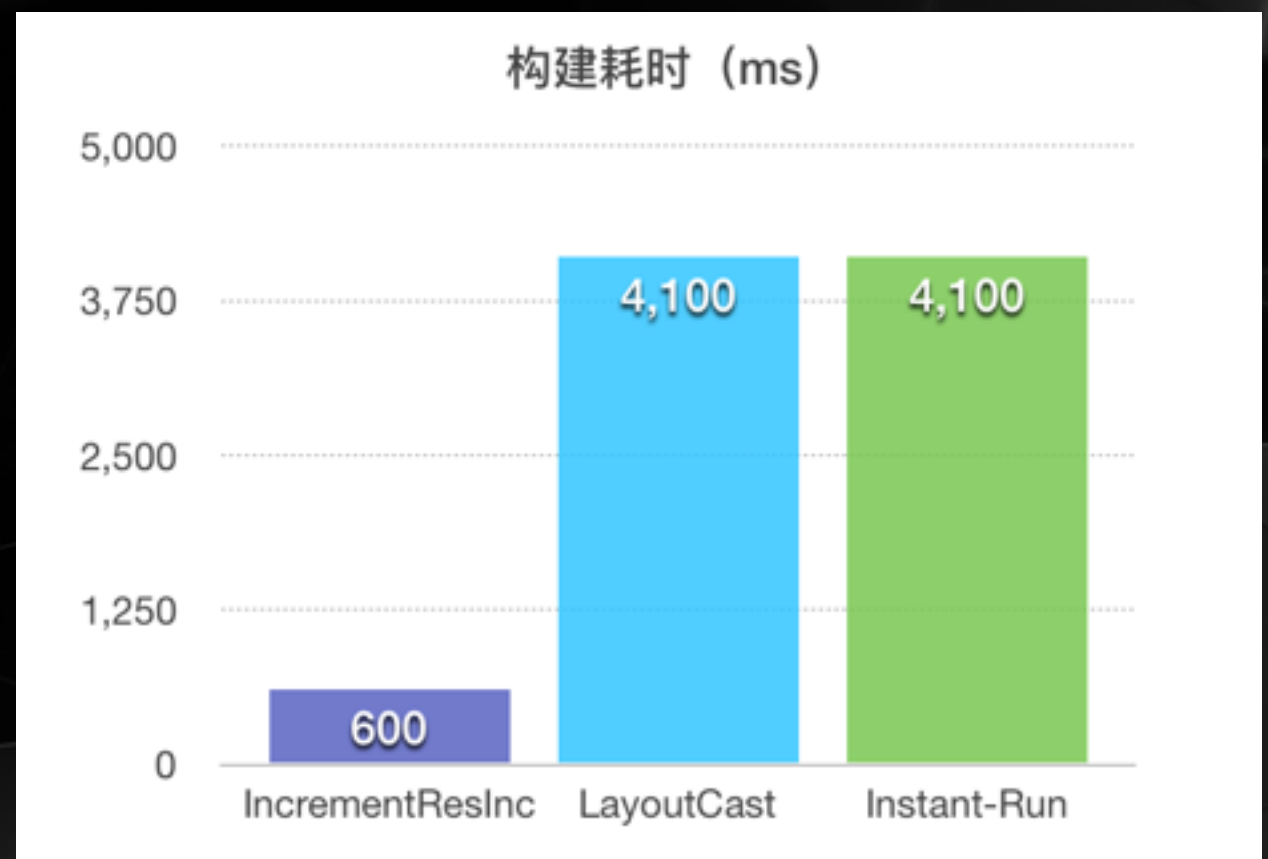
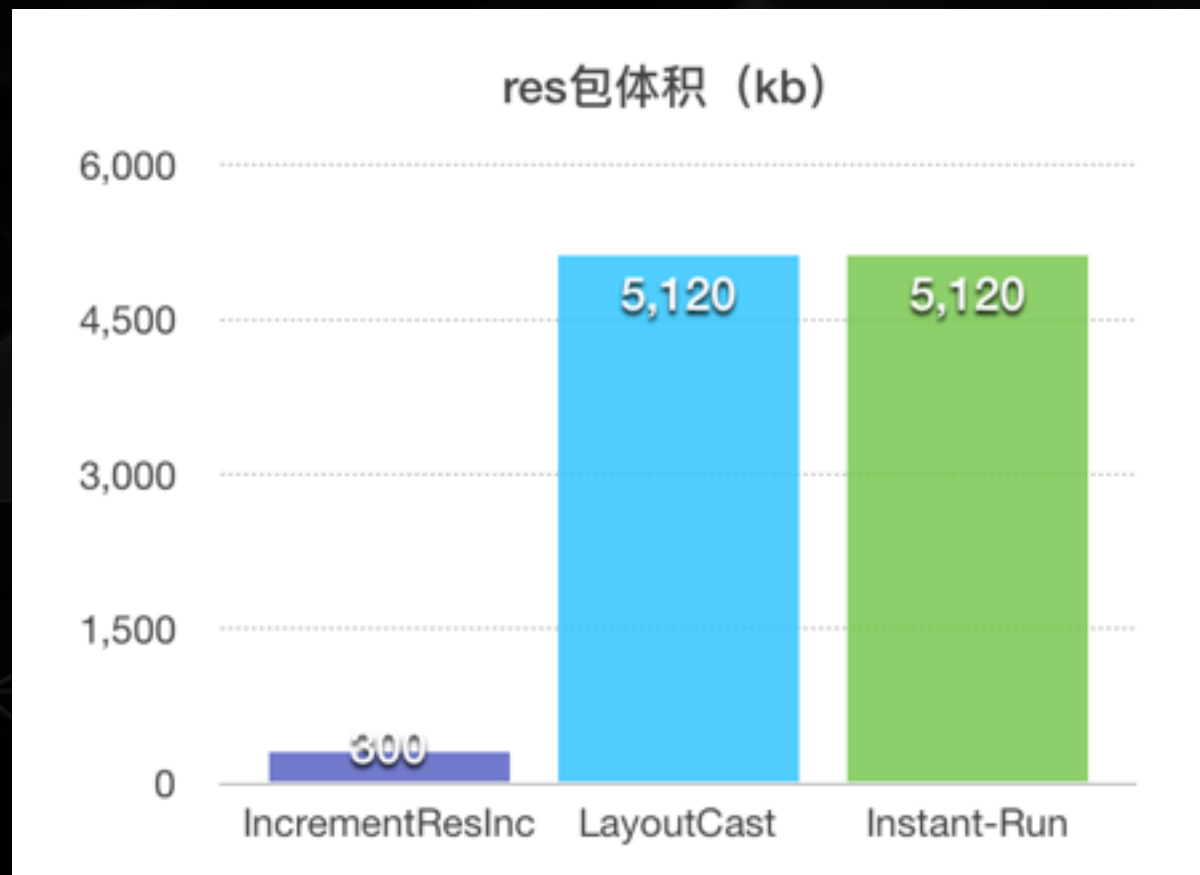
inc-build

```
--- command 6 (verbose=0 force=1):  
Freeline trace:: slurp up res last 34.009000 ms  
Freeline trace:: makeFileResources all resource. Time Elapsed: 14.814000 ms  
Freeline trace:: compile values. Time Elapsed: 237.380000 ms  
Freeline trace:: makeFileResources for color and menu. Time Elapsed: 0.234000 ms  
Freeline trace:: generate all bag attributes and assign indices. Time Elapsed: 64.001000 ms  
Freeline trace:: compile all xml. Time Elapsed: 32.840000 ms  
Freeline trace:: assign id for name res/layout/fragment_mywealth_home_2.xml  
Freeline trace:: flatten gen resources.arsc Time Elapsed: 62.738000 ms  
Freeline trace:: gen r file last 32.472000 ms  
Freeline trace:: APK Bundling. Time Elapsed: 11.518000 ms  
Freeline trace:: build pack last 312.600000 ms in total
```


技术原理-资源增量-手机端生效 (1)



技术原理-资源增量-手机端生效-数据对比



技术原理-资源增量-手机端生效-OverLay

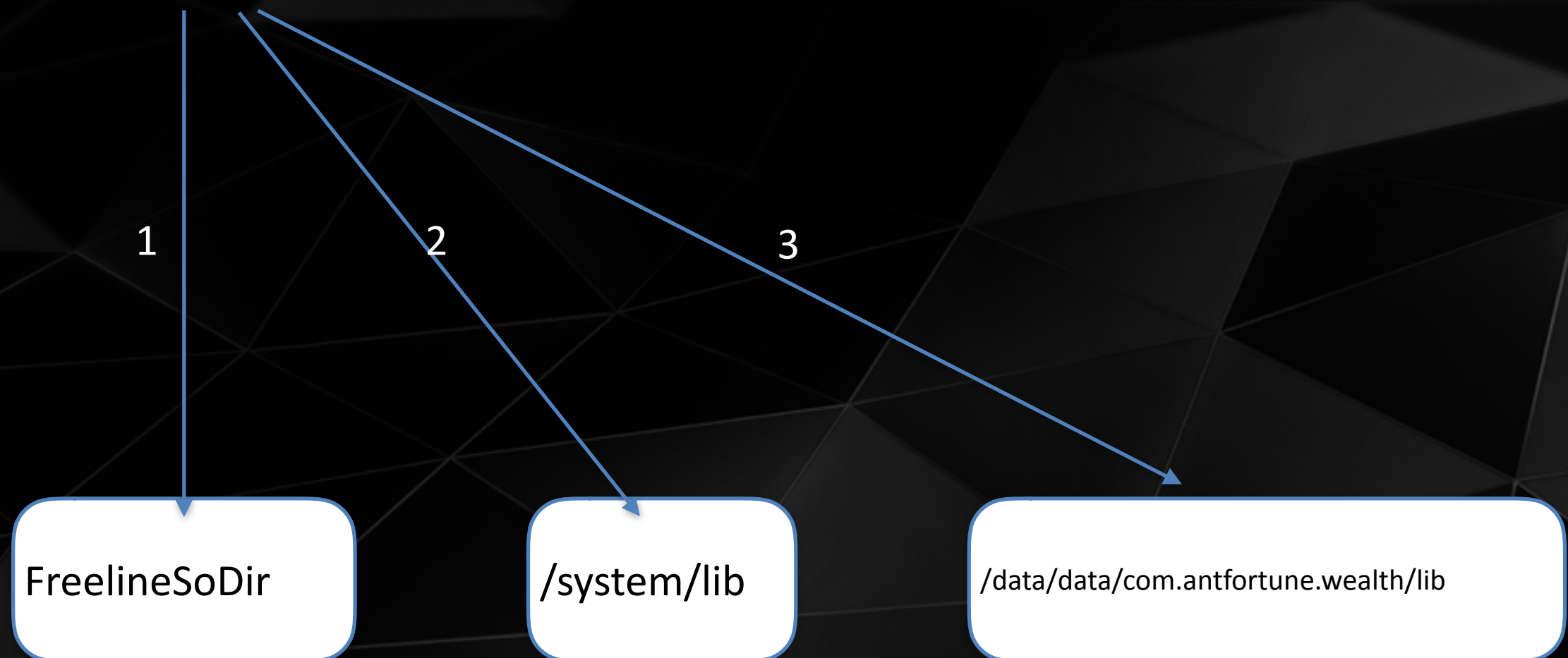
1.只能实现修改，不支持新增

2.包体大小随着修改范围越来越大

技术原理-so动态替换

nativeLibraryDirectories / nativeLibraryPathElements

find so



技术细节

ASM 修改class

R.class 去final

```
final class A {  
    public static int a = 1;  
    public static int b = 2;  
}
```

android:vmSafeMode

数据对比-兼容性

兼容性

| 手机端 | ANDROID 2.0~4.0 | ANDROID5.0~6.0 |
|-------------------|-----------------|----------------|
| LayoutCast | × | √ |
| Buck | N/A | N/A |
| Android studio2.0 | × | √ |
| Freeline | √ | √ |

| PC端 | LINUX/MAC | WINDOWS |
|-------------------|-----------|---------|
| LayoutCast | √ | √ |
| Buck | √ | × |
| Android studio2.0 | √ | √ |
| Freeline | √ | √ |

数据对比-性能

Test with **50w line** code and **40mb** resources, Environment:(macbook 2015 + galaxy note4)

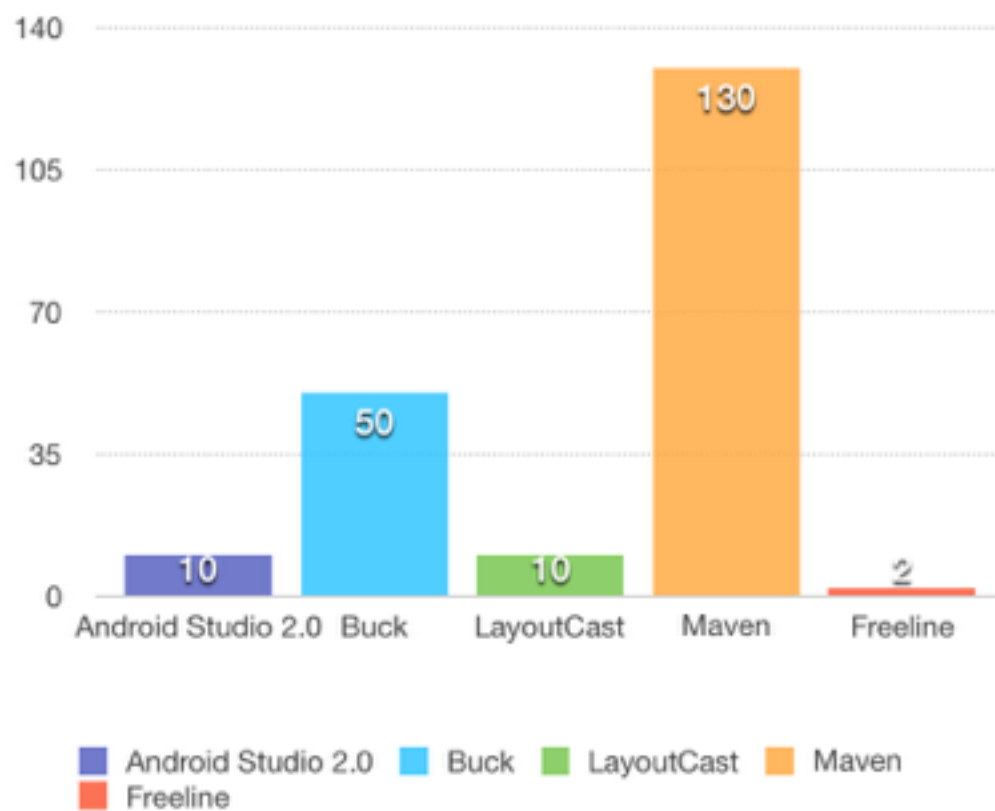
code change build + install

| COMPLIE SCHEME | CODE CHANGE(SECOND) |
|--------------------|---------------------|
| Android Studio 2.0 | 10 |
| Buck | 50 |
| LayoutCast | 10 |
| Maven Full Build | 130 |
| Freeline | 2 |

resource change build + install

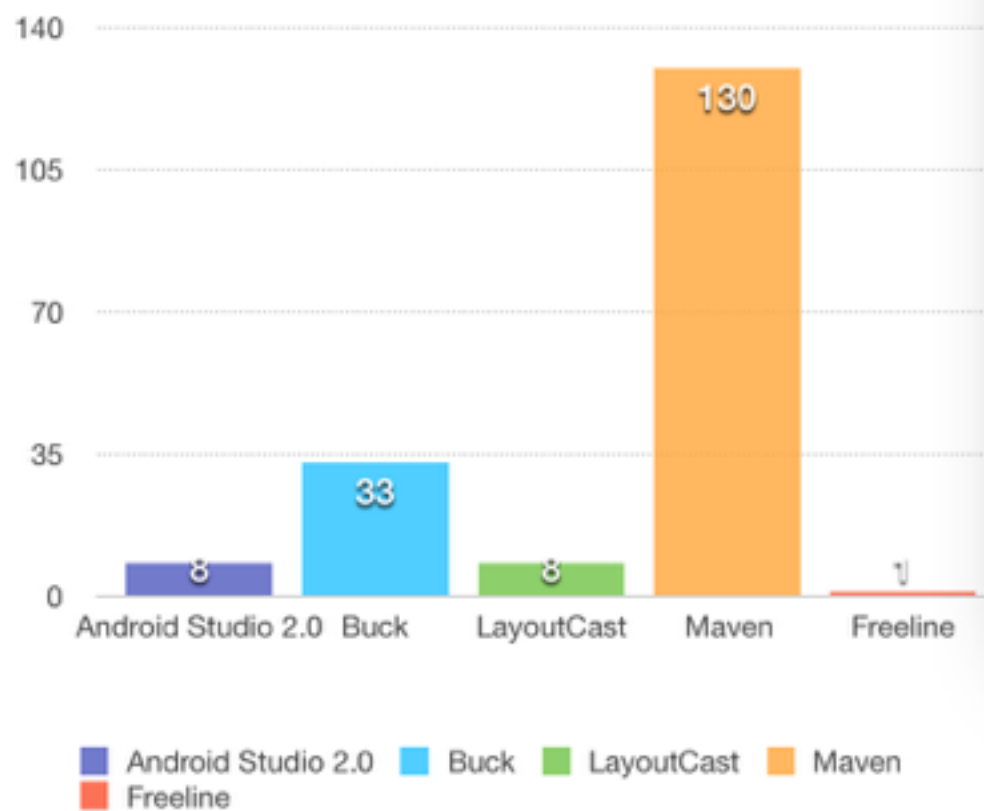
| COMPLIE SCHEME | RES CHANGE(SECOND) |
|--------------------|--------------------|
| Android Studio 2.0 | 8 |
| Buck | 33 |
| LayoutCast | 8 |
| Maven Full Build | 130 |
| Freeline | 1 |

Code Change



Promote **60** times

Code Change



Promote **> 100** times

接入说明-仅需3步

一.配置project-level的build.gradle, 加入freeline-gradle的依赖:

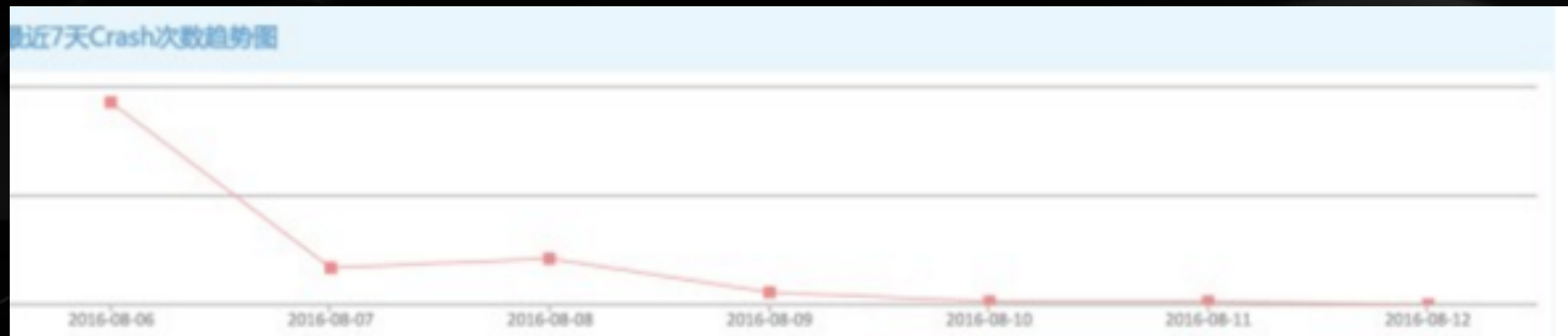
```
buildscript {  
    repositories {  
        jcenter()  
    }  
    dependencies {  
        classpath 'com.antfortune.freeline:gradle:0.7.3'  
    }  
}
```

二.在你的主module的build.gradle中, 应用freeline插件的依赖:

```
apply plugin: 'com.antfortune.freeline'
```

三.运行一次 './gradlew initFreeline -Pmirror'

应用场景-线上热修复



应用场景-业务A/B-Test-实际案例

背景

进B方案之后，将提问和发布观点按钮独立开来，看看是否能提高提问内容数量。
如果提问内容数量增加，是否会对交流区互动产生比较大的影响

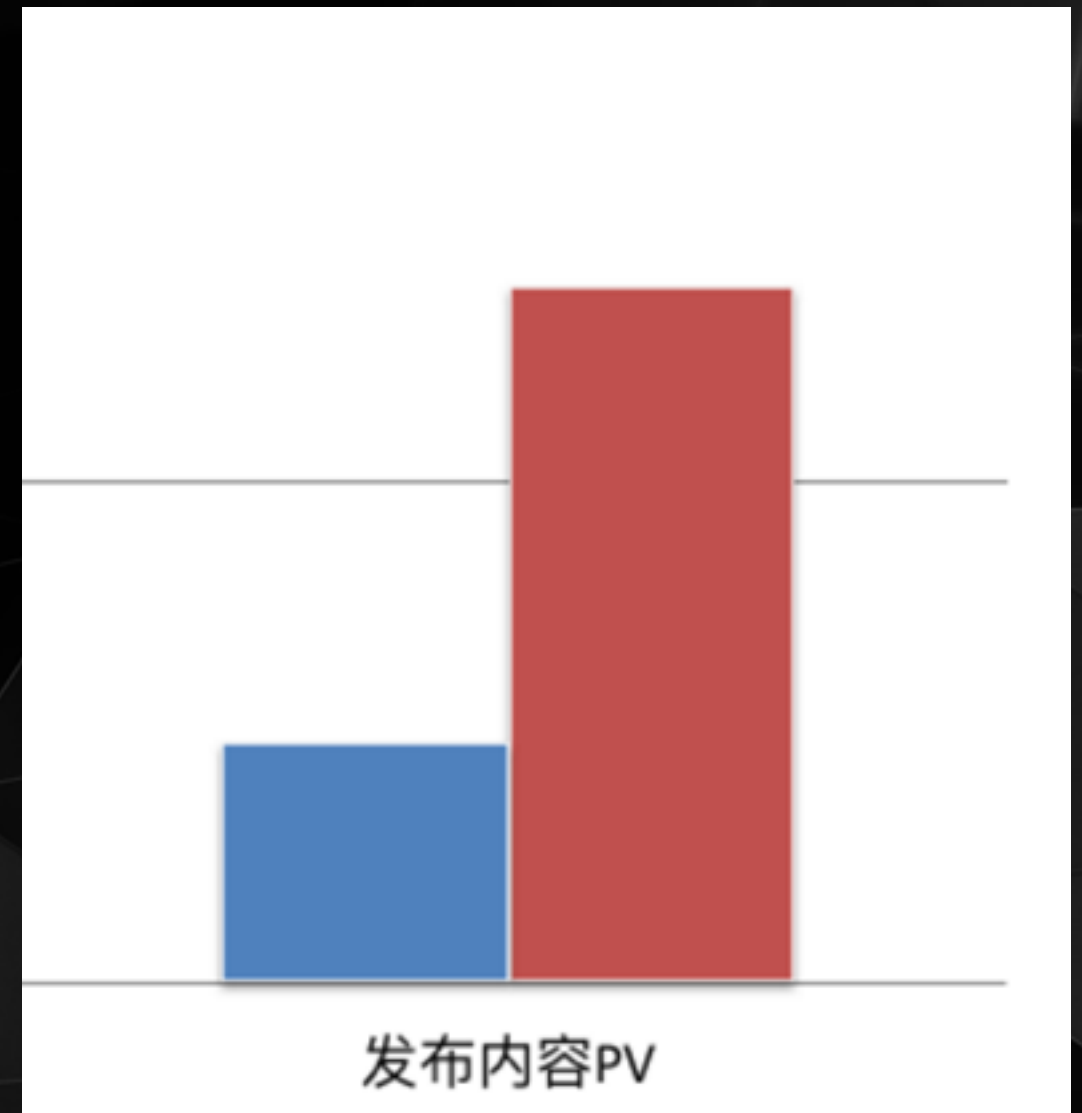


应用场景-业务A/B-Test-实际案例

1天开发

2天内完成测试+推送

7天内完成数据回流提供给产品
决策



release及后续计划

已支持:

Annotation (ButterKnife, AndroidAnnotation, Dragger)

Android studio 插件

Retrolambda

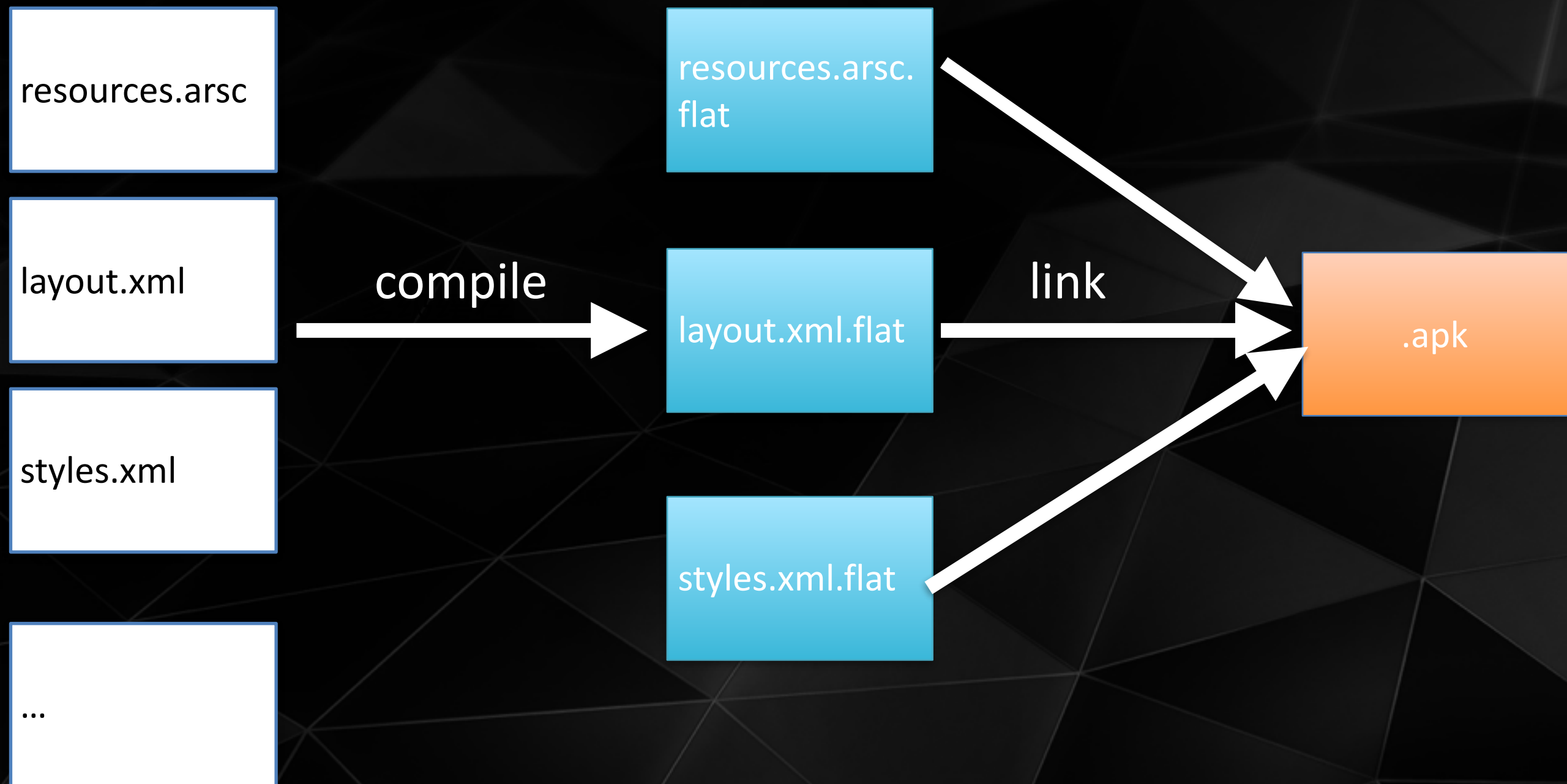
后续计划:

Databinding

稳定性/兼容性持续提升

Aapt2.0优化后引入

资源编译大趋势—Aapt2.0

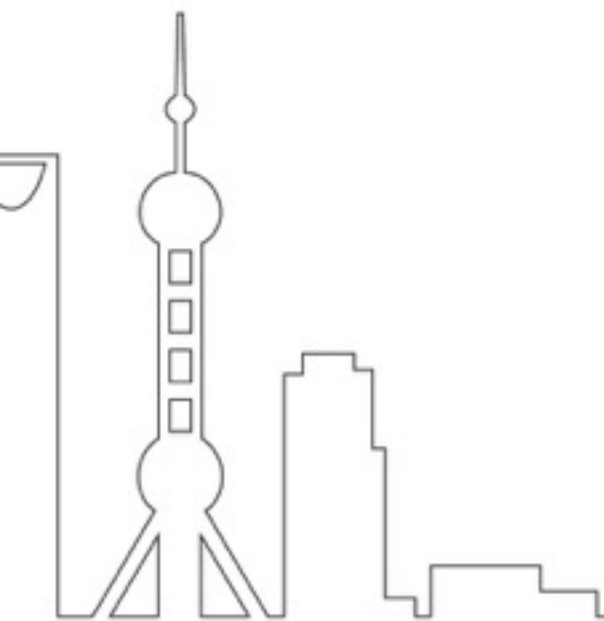


开源及反馈方式

开源地址: <https://github.com/alibaba/freeline>

官方群





Thanks!

International Software Development Conference