

# 死丢丢的魅力

The Beauty of Android Studio



longtianlove



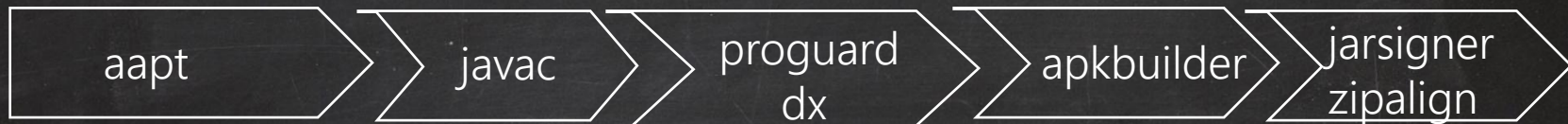
1/ 编译打包流程

2/ 文件结构与安装流程

3/ 增量编译

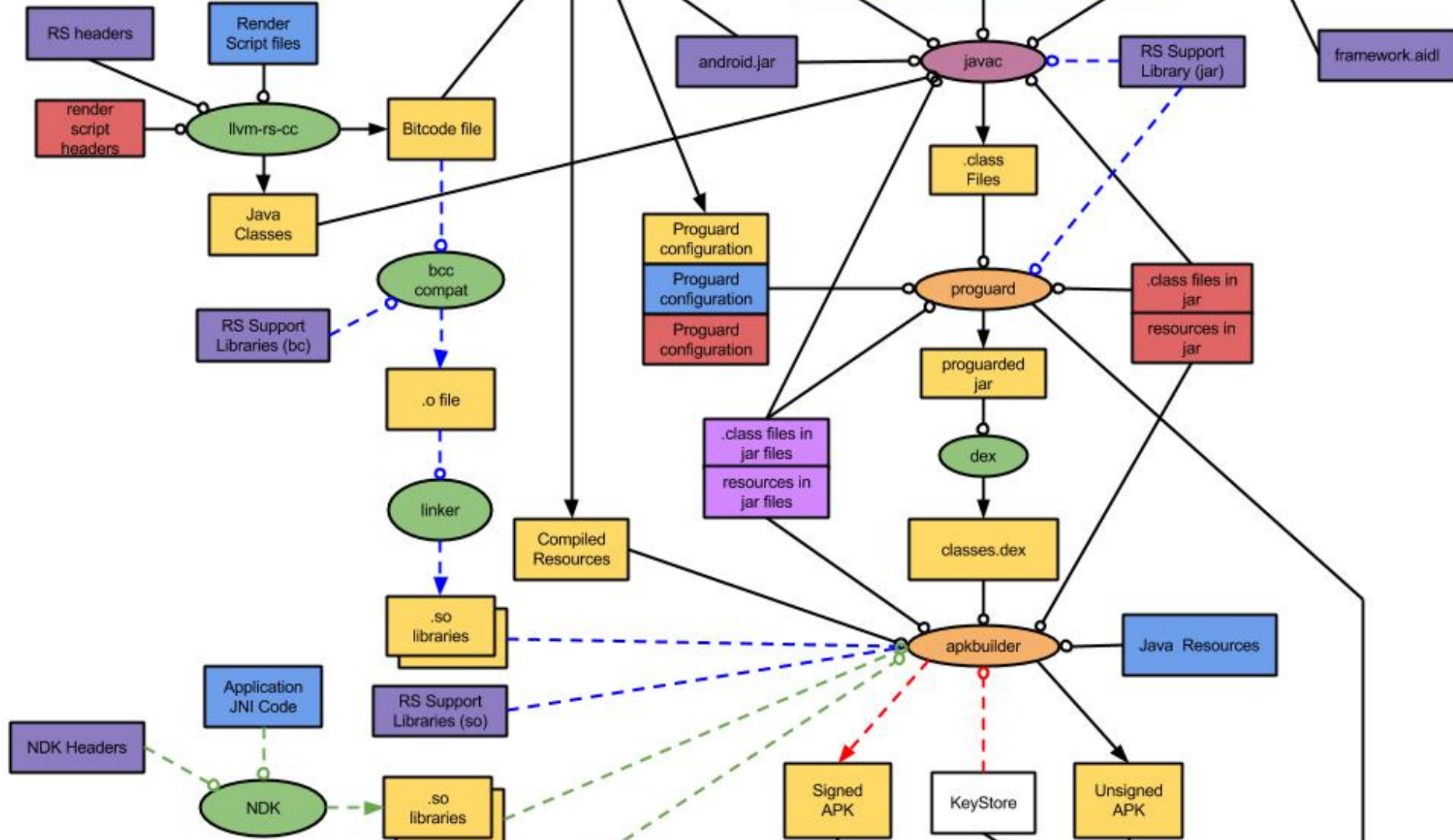
4/ 更新

## 编译打包流程



打包资源文件	编译源文件	转化dex	打包apk	签名、对齐
Resource、Assets、 AndroidManifest.xml、 Android.jar	R.java、Application java code、 android.jar	.class、.jar	资源文件、.dex、 .so	app-release- unaligned.apk
R.java、 resources.ap_	.class	.dex、 mapping file	app-release- unaligned.apk	app-release.apk





# 文件结构

classes.dex

执行者

resources.arsc

存放资源文件和资源ID之间的映射关系。用来根据资源ID寻找资源。Android的开发是分模块的，res目录专门用来存放资源文件，当在代码中需要调用资源文件时，只需要调用findviewbyId()就可以得到资源文件。

AndroidManifest.xml

Android应用向Android系统“自我介绍”的配置文件。注册使用到的组件。还包含一些权限声明以及使用的SDK版本信息等。打包时，进行简单的编译，便于Android系统识别，编译之后的格式是AXML格式。

res/

存放资源文件，这个文件夹下的所有文件都会映射到Android工程的.R文件中，生成对应的ID，访问的时候直接使用资源ID即R.id.filename，res文件夹下可以包含多个文件夹，可以存放动画文件、图像资源、布局文件、一些特征值、color颜色值、尺寸值等

META-INF/

存放应用的签名信息，签名信息可以验证APK文件的完整性。其中CERT.RSA是开发者利用私钥对APK进行签名的签名文件，CERT.SF，MANIFEST.MF记录了文件中文件的SHA-1哈希值。

assets/

存放需要打包到APK中的静态文件。assets不会自动生成对应的ID，访问的时候需要AssetManager类

lib/

存放应用程序依赖的native库文件。根据cpu架构不同，大致分为ARM，ARM-v7a，MIPS，X86等





## copy数据

apk文件被copy到**data/app**目录下。解压apk, 得到dex, 然后优化成**odex**, 存放在**/data/dalvik-cache**。**/data/data/**目录下创建应用程序的数据目录（以应用的包名命名），存放应用数据，如数据库、xml文件、cache、二进制的so动态库等。



## 解析AndroidManifest.xml

将信息写入**/data/system/packages.xml**, 包含: **appId(包名)**, **apk的位置data/app/包名-2.apk**, **本地库位置/data/app-lib/包名-2**, 版本, **userId**, 签名, 权限等



## 校验

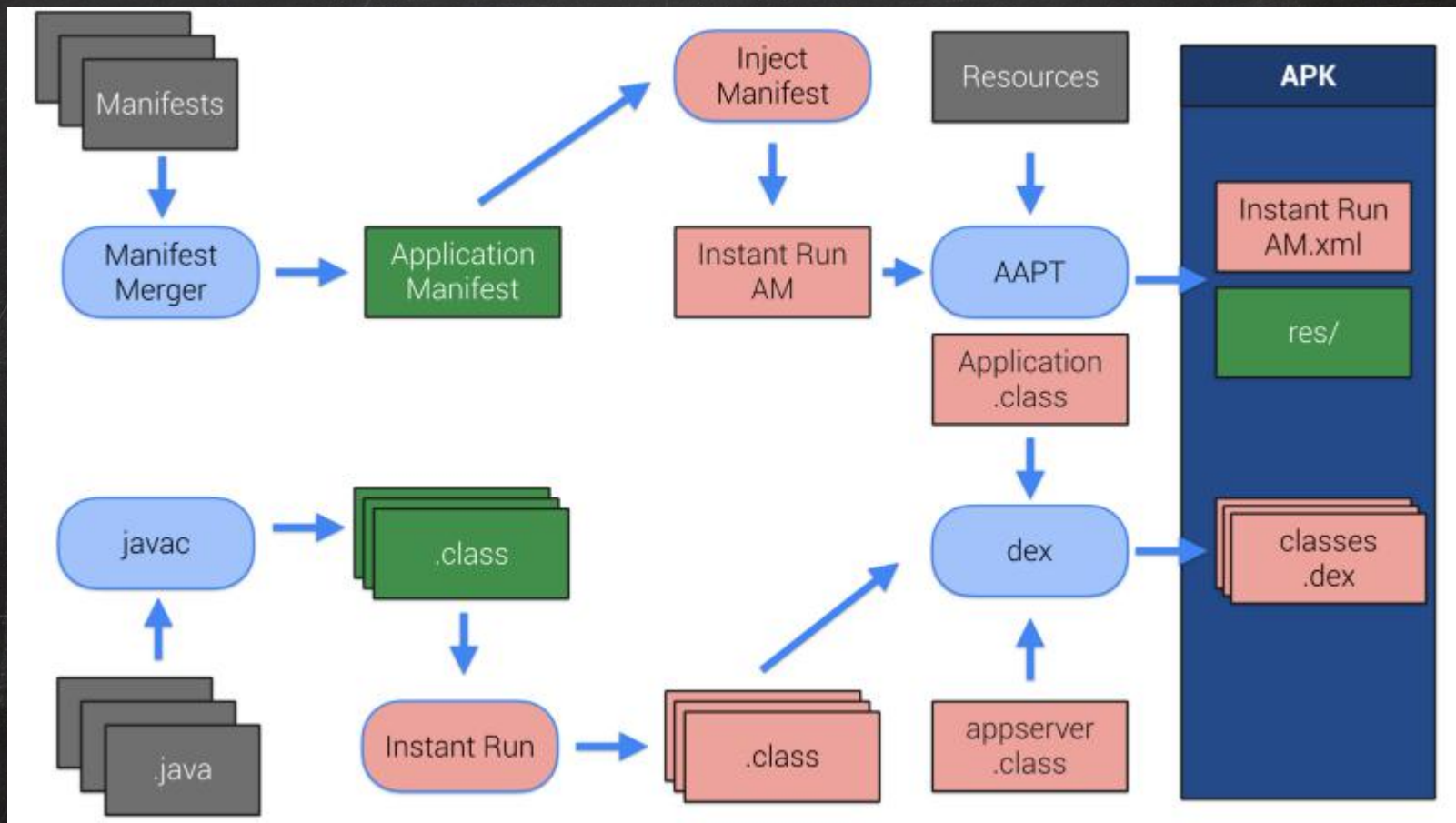
校验APK的签名是否正确，检查APK的结构是否正常，进而解压并且校验APK中的dex文件，确定dex文件没有被损坏后，再把dex优化成odex，使得应用程序启动时间加快，同时在**/data/data**目录下建立于APK包名相同的文件夹，如果APK中有lib库，系统会判断这些so库的名字，查看是否以lib开头，是否以.so结尾，再根据CPU的架构解压对应的so库到**/data/data/包名/lib**下。



## 第一次编译

1. 将instant-run.jar打包进主dex
2. AndroidManifest.xml中application为com.android.tools.fd.runtime.BootstrapApplication
3. 使用asm工具，在每个类中添加\$change，在每个方法前加逻辑
4. 把源代码编译成dex，然后存放到压缩包instant-run.zip中

## 第一次编译

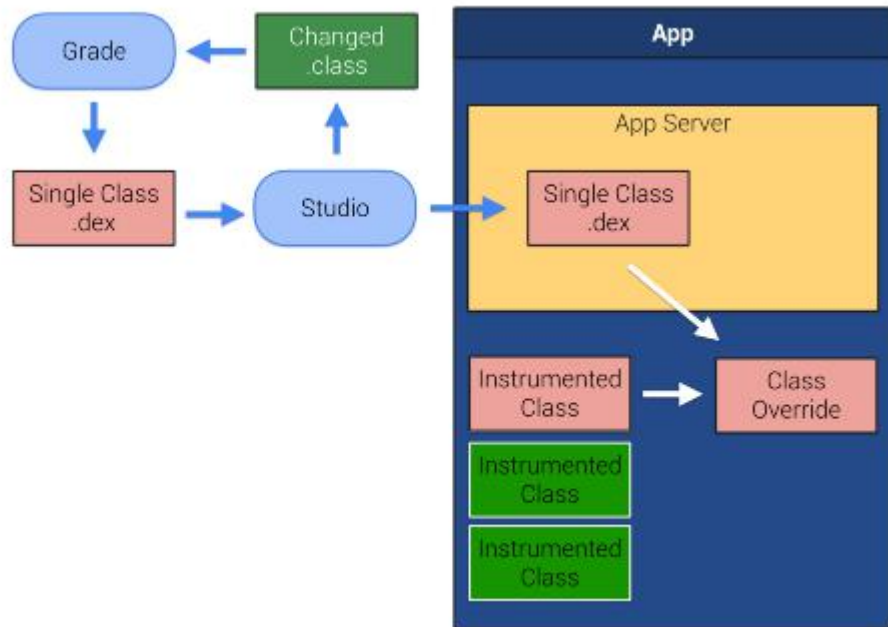




## 运行时

1. 获取更改后资源resource.ap\_的路径
2. 设置ClassLoader
3. 创建apk真实的application
4. 反射替换ActivityThread中的各种Application成员变量
5. 反射替换所有存在的AssetManager对象
6. 调用realApplication的onCreate方法
7. 启动Server，Socket接收patch列表

## 运行时

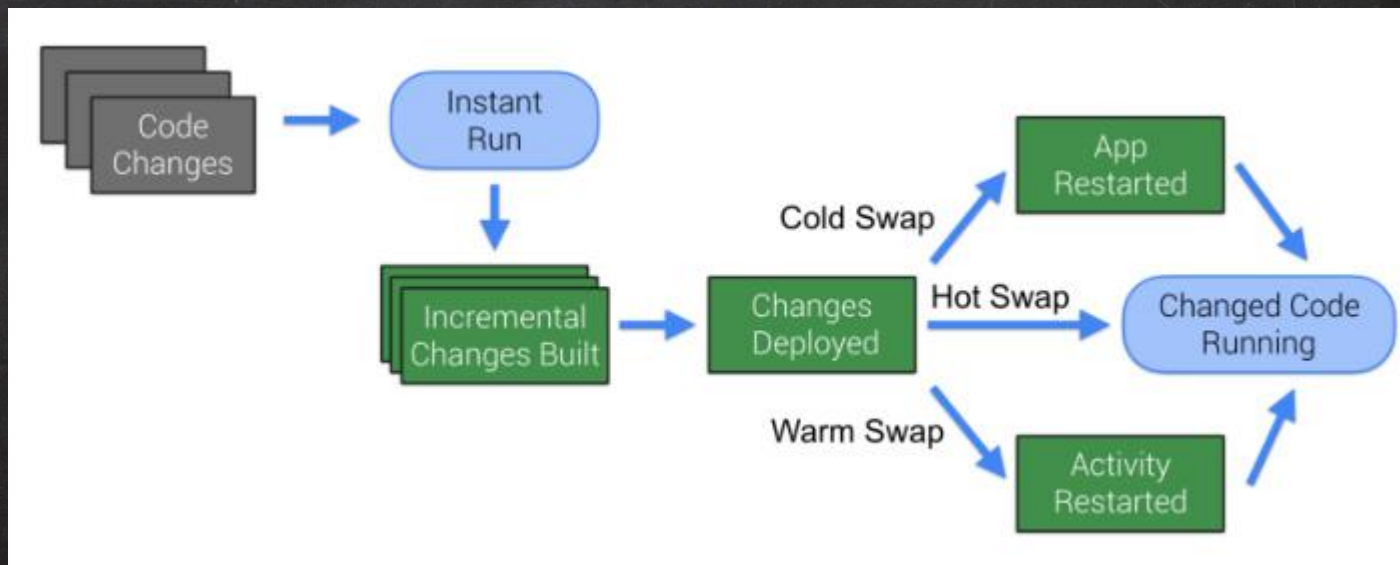




更新

1. 获取更改后资源resource.ap\_的路径
2. 生成AppPatchesLoaderImpl类，记录修改的类列表
3. 打包成patch，通过socket传递给app
4. app的server接收到patch之后，分别按照handleColdSwapPatch、handleHotSwapPatch、handleResourcePatch等待对patch进行处理
5. restart使patch生效





希望英才越来越好！