

基于Android App Bundle 动态化方案探索

陈家伟

爱奇艺资深工程师

自我介绍

陈家伟

二零一四年，毕业于南昌大学。

工作轨迹：步步高 -> 平安 -> 至今爱奇艺。

技术领域：Android 动态化，React Native 等，目前主要负责基于Android App Bundle 动态化方案Qigsaw研发和推广。



目录

- Google减少APK体积的发展历程
- 国内插件化发展回顾及原理分析
- Android App Bundle原理介绍
- Qigsaw简介及原理分析

回首Android第一个10年，应用发布方式。

Android应用发布流程

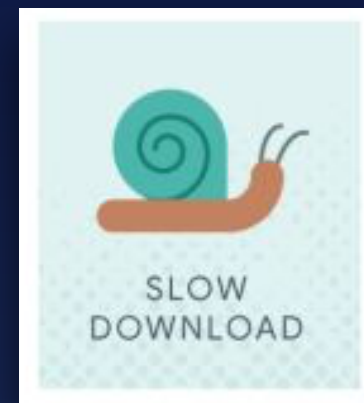


APK

APK

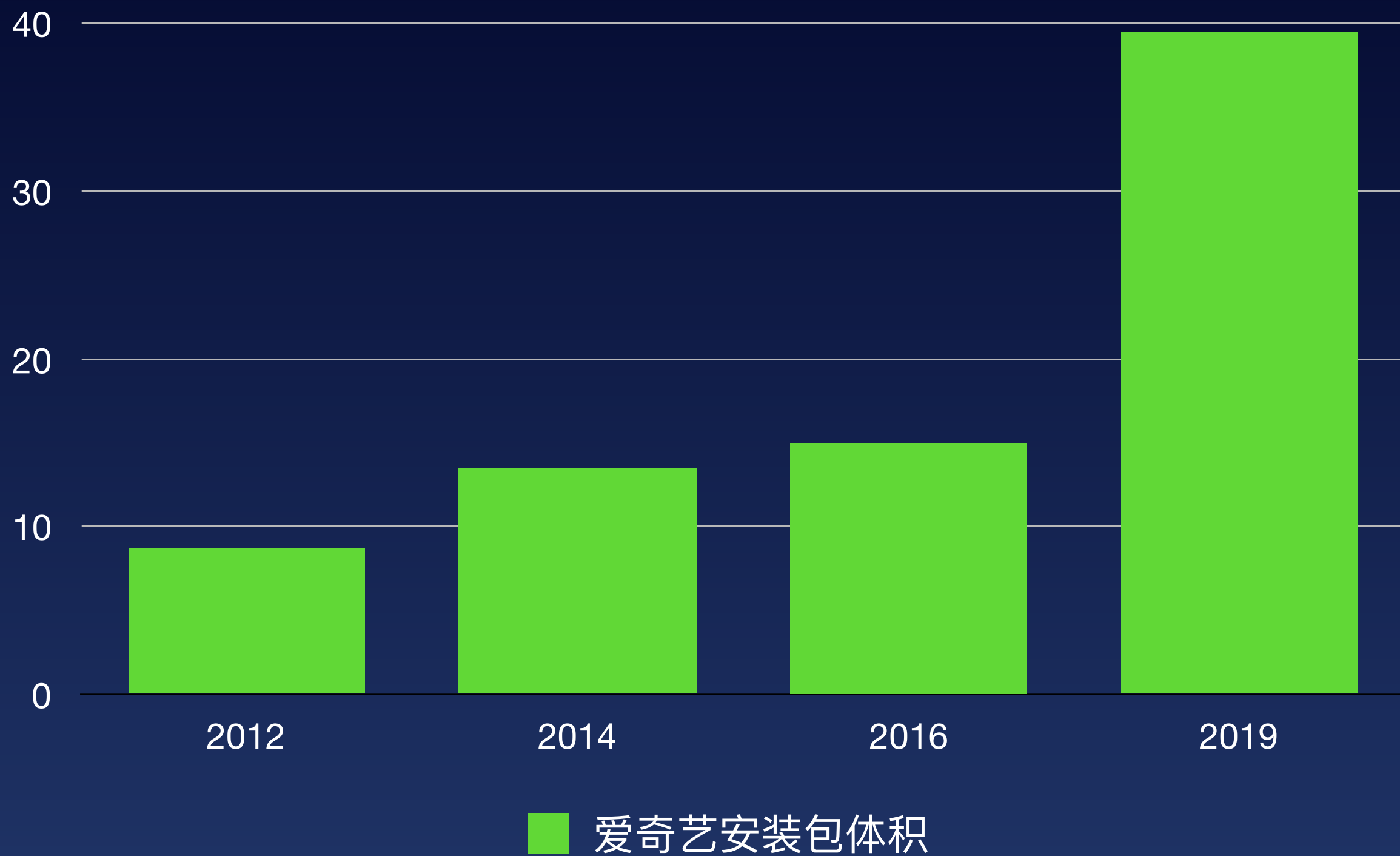
APK

传统发布方式弊端



all native library,
all language,
all density,
all OS
all...

难以忽视的“大”问题



自2012年以来，
应用平均体积增长**5**倍

应用体积越大，安装成功率越低

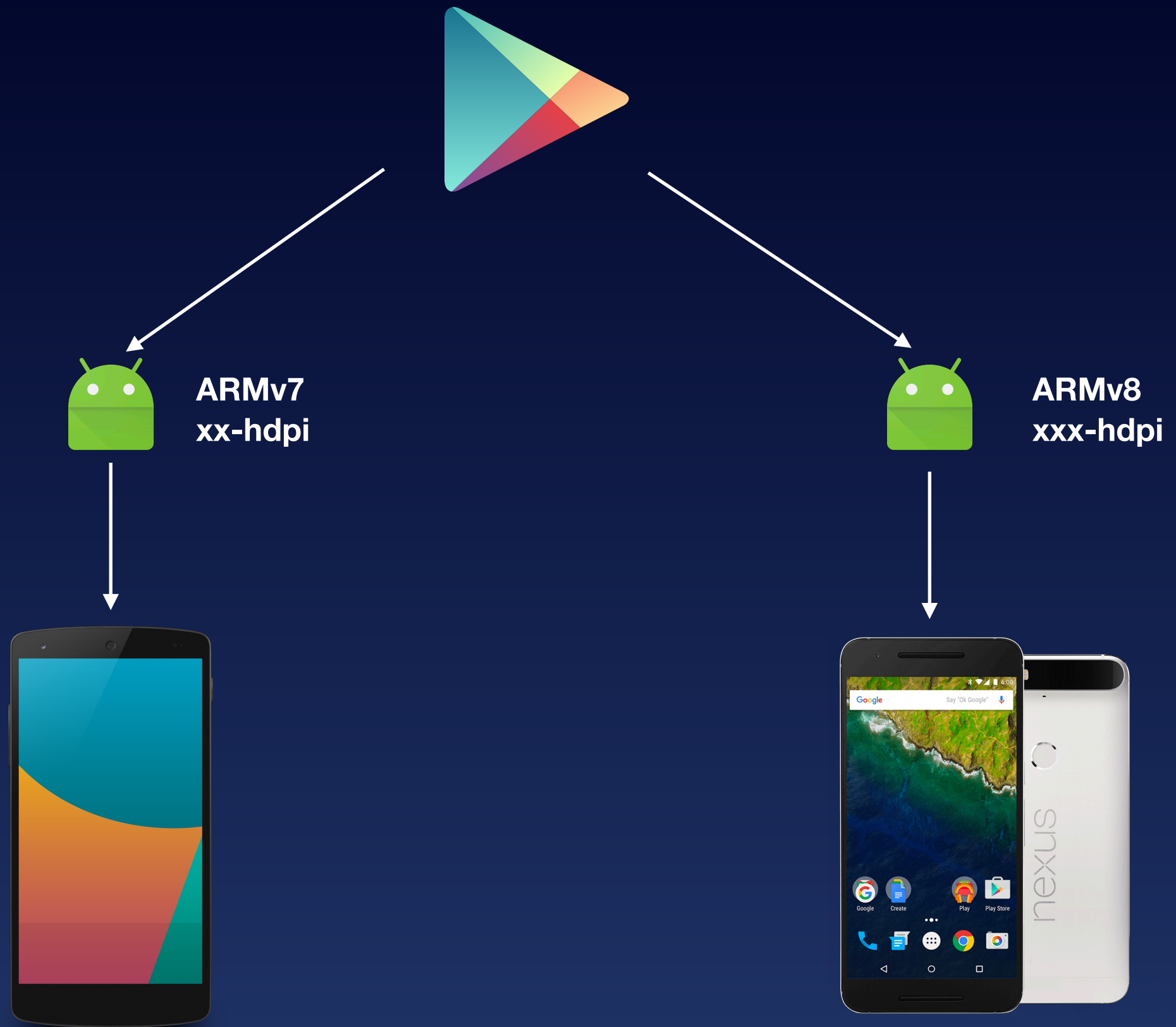


Android 5.0 推出Multiple APK, 旨在减少安装包体积

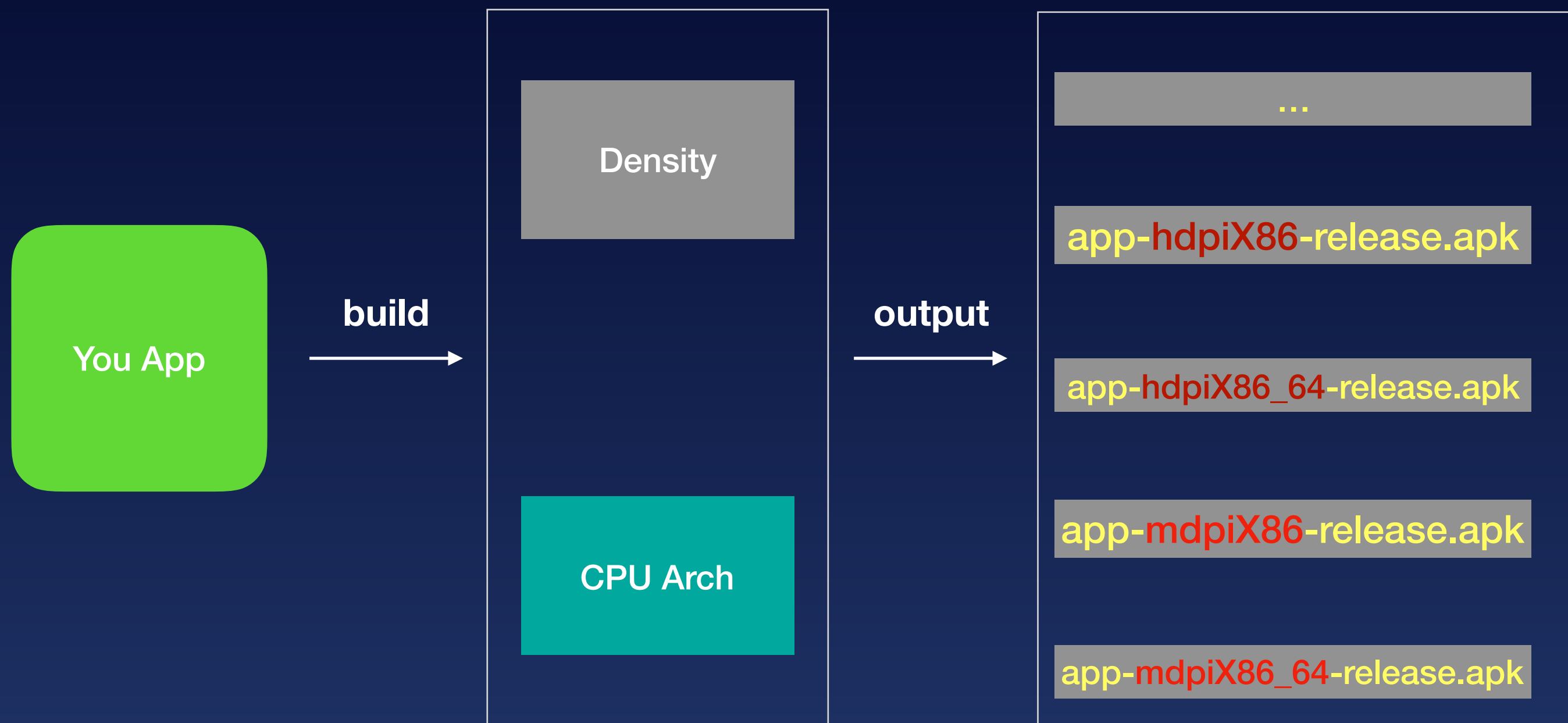
Multiple APK

Multiple APK是Google Play提供一个功能，它允许您的应用针对不同的设备配置发布不同的APKs。

<https://developer.android.com/google/play/publishing/multiple-apks>



```
android {  
    ...  
    splits {  
        // Configures multiple APKs based on screen density.  
        density {  
            ...  
            // Specifies a list of screen densities Gradle should not create multiple  
            APKs for.  
            exclude "ldpi", "xxhdpi", "xxxhdpi"  
        }  
        // Configures multiple APKs based on ABI.  
        abi {  
            ...  
            // Specifies a list of ABIs that Gradle should create APKs for.  
            include "x86", "x86_64"  
            // Specifies that we do not want to also generate a universal APK that  
            includes all ABIs.  
            universalApk false  
        }  
    }  
}
```

Multiple APK 弊端

您可能为每个版本构建
数百个APK，降低迭代
效率。



Multiple APK的“恶果”

开发者宁愿它“胖”，也不愿麻烦。
导致用户设备存在大量未使用过的内容。



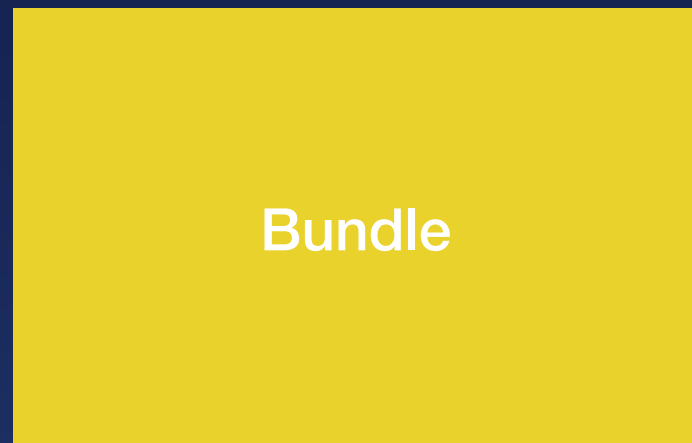
Android App Bundles



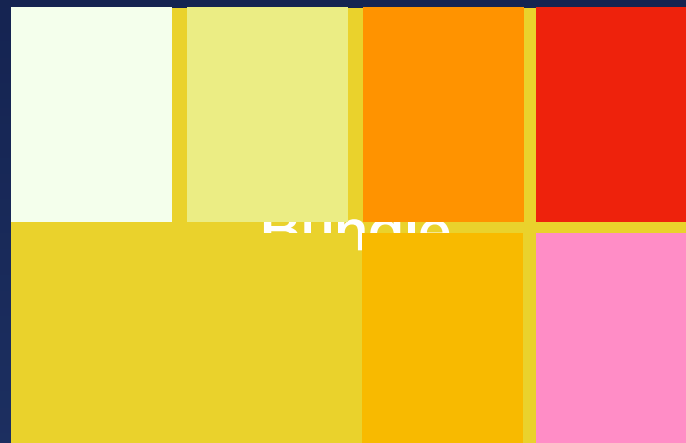
Android App Bundle发布流程



Pixel 2XL

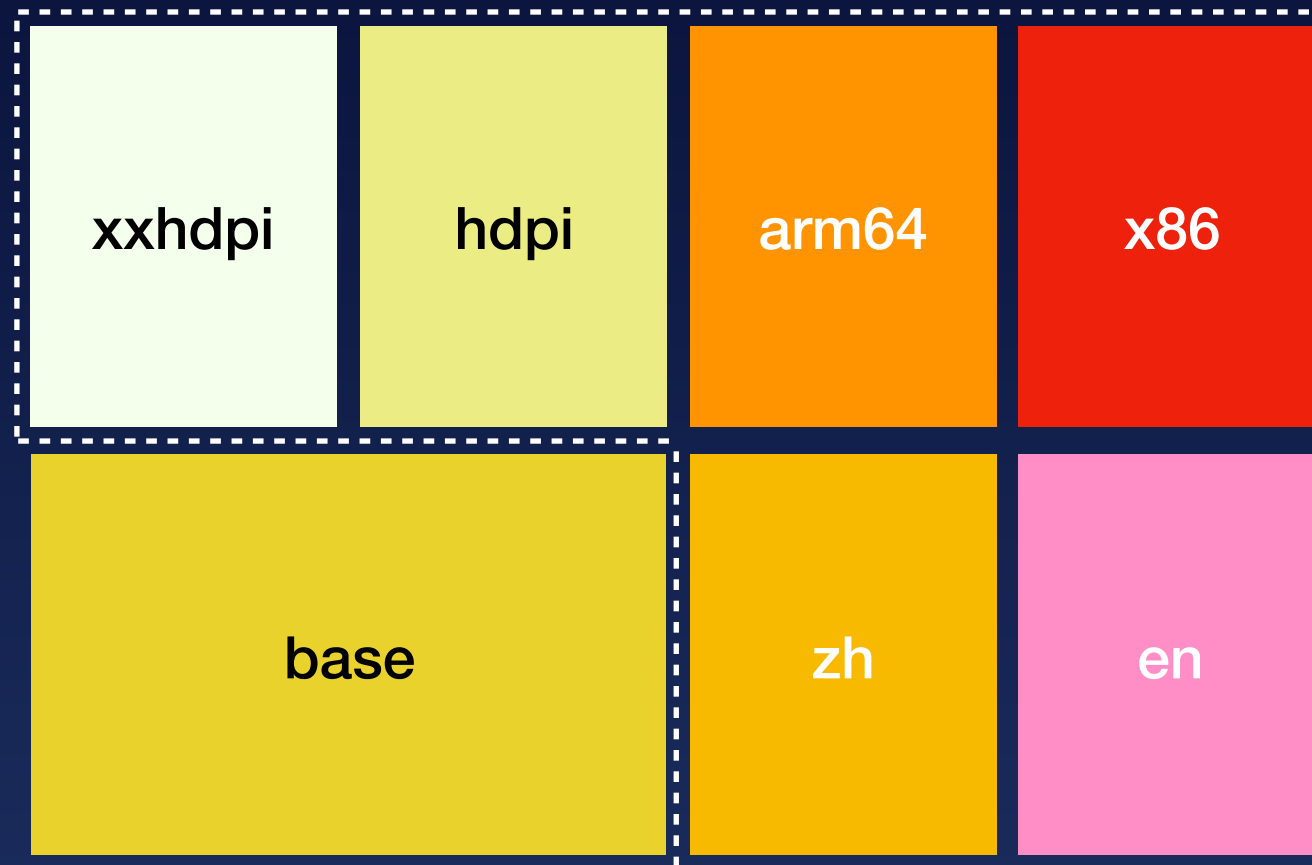


Android App Bundle



Android App Bundle

**Base APK
(shared code)**



**Configuration
APKs**





Pixel 2 XL

en

xxxhdpi

arm64



Dynamic Delivery



Pixel 2 XL (pre L)

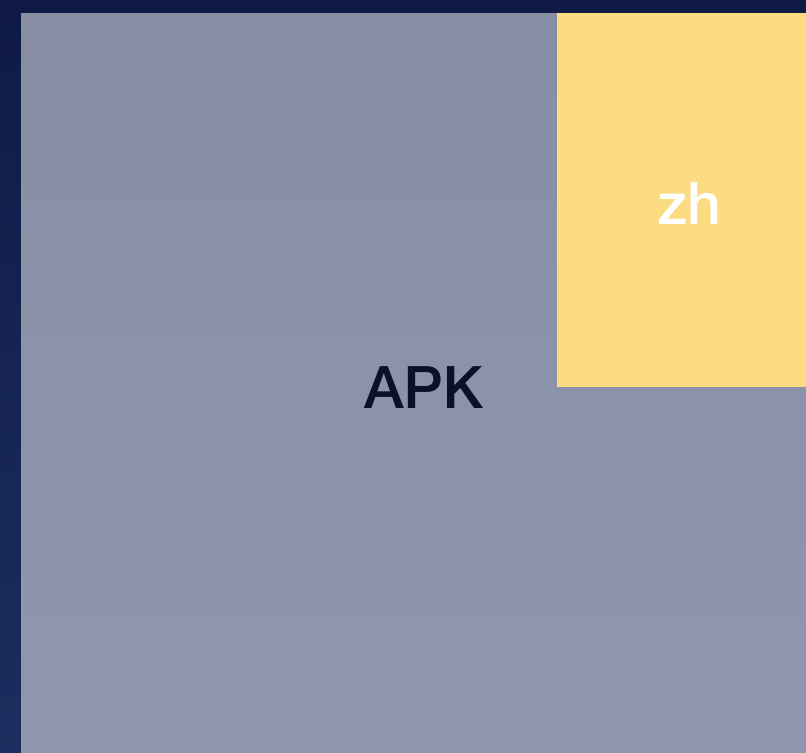
en

xxxhdpi

arm64



Split APKs

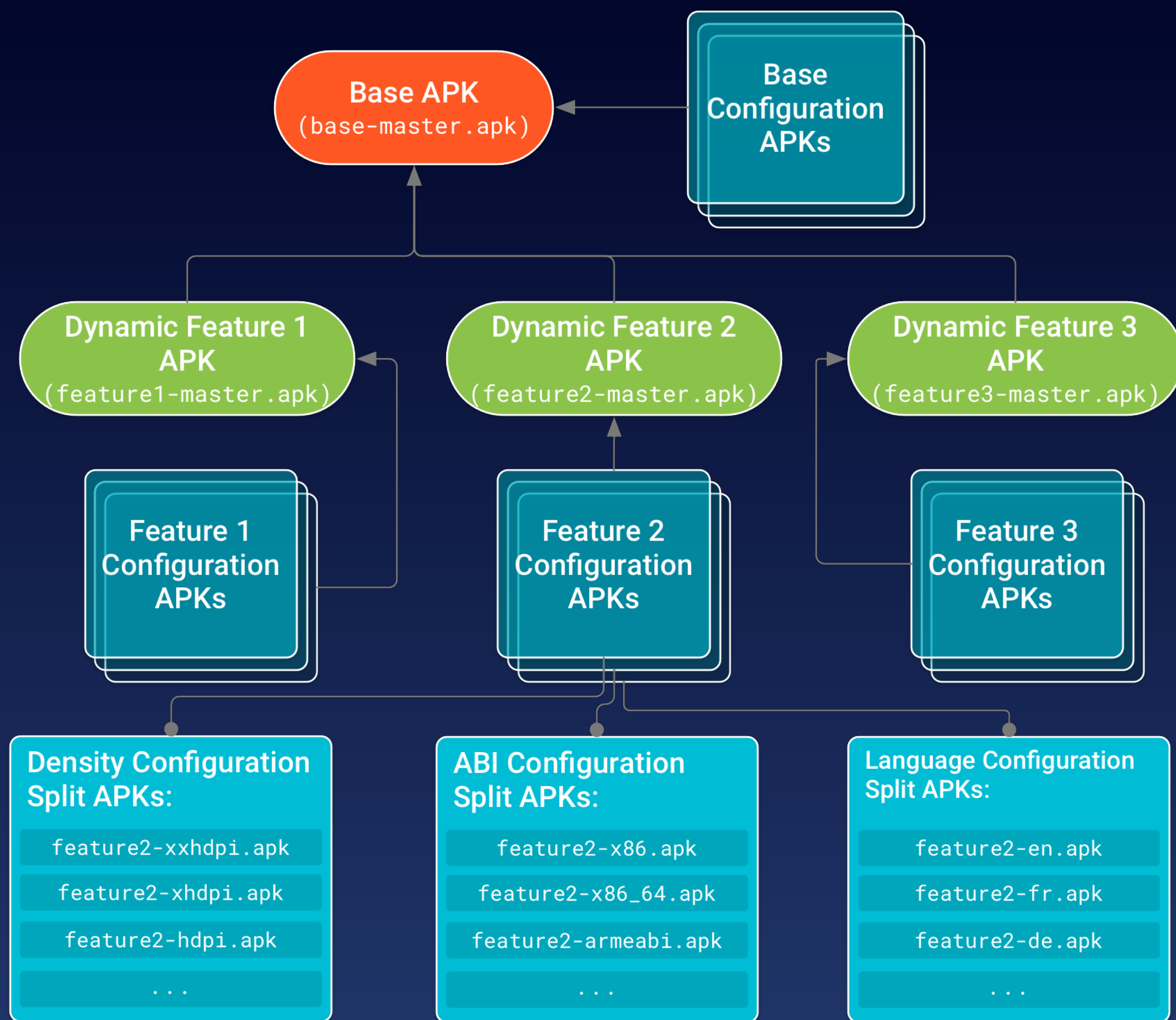


Optimized APK

Dynamic Feature

- 安装时不需要的大型功能(付费高级功能、AR功能等)。
- 针对特定受众群体的功能(商业应用中特定用户下载专属功能)。
- 很少使用的功能(身份验证、信用卡扫描)。





国内插件化发展回顾



AndroidDynamicLoad



DynamicLoadAPK



Neptune



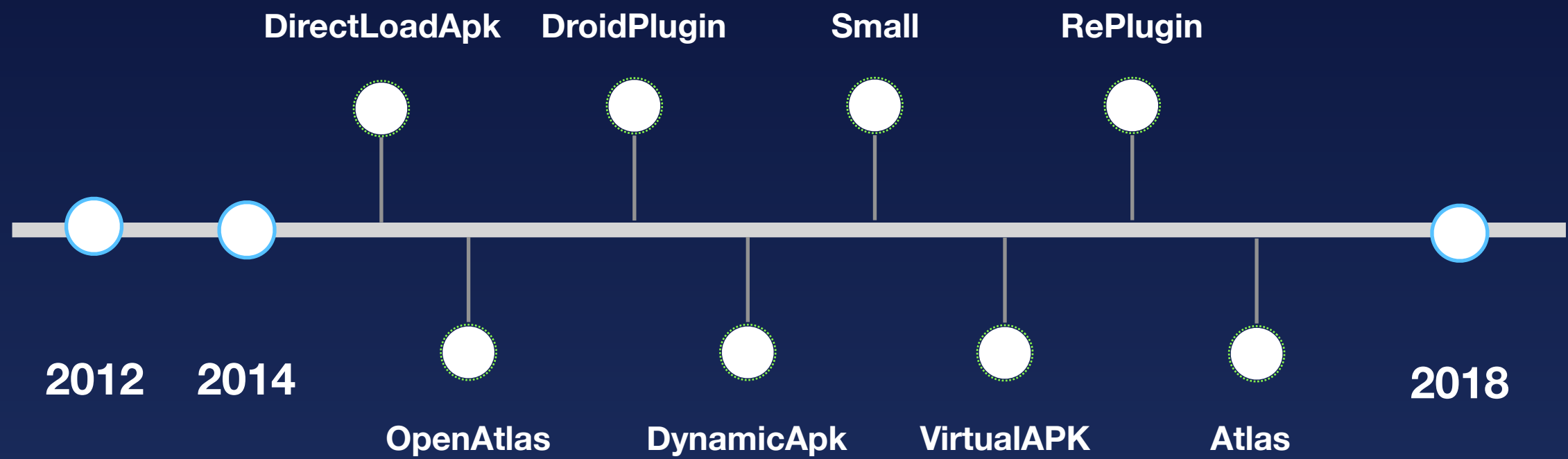
2012



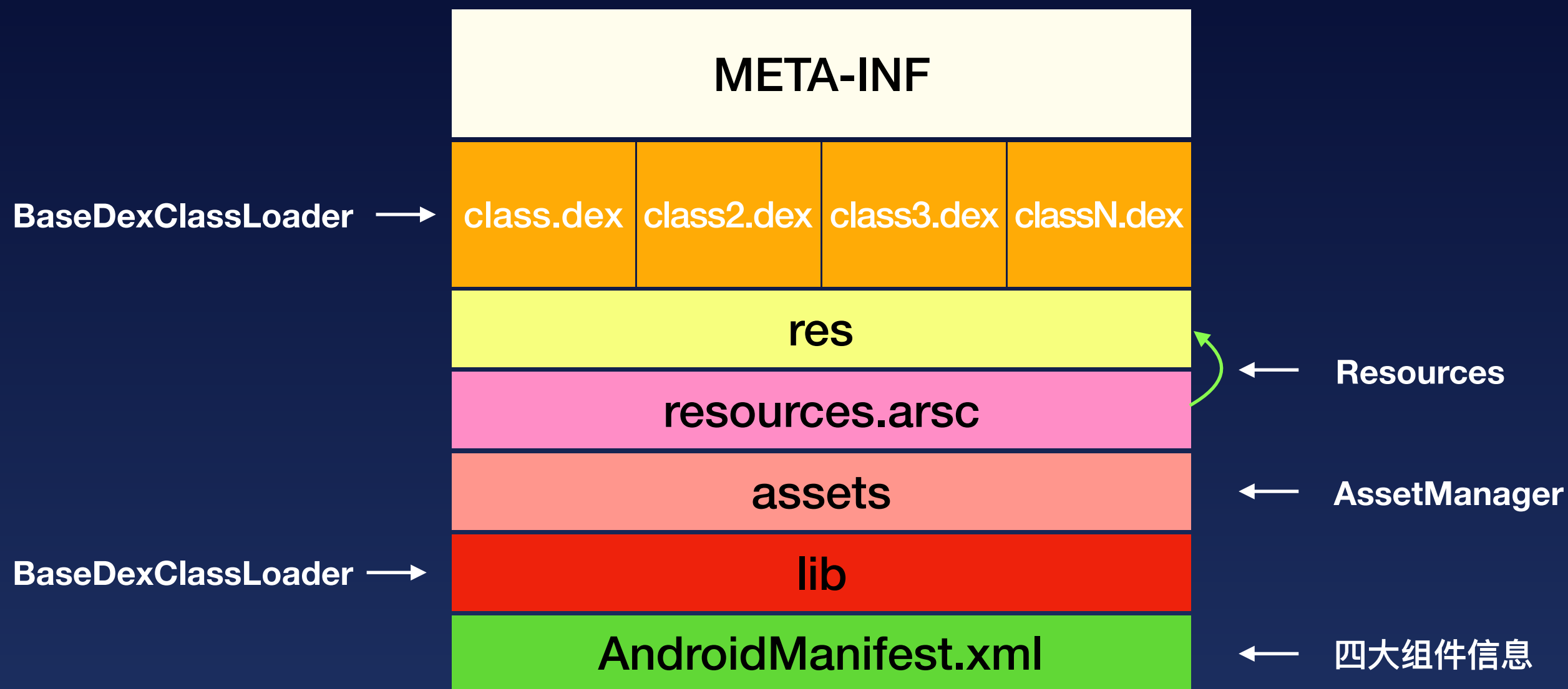
2014



2018

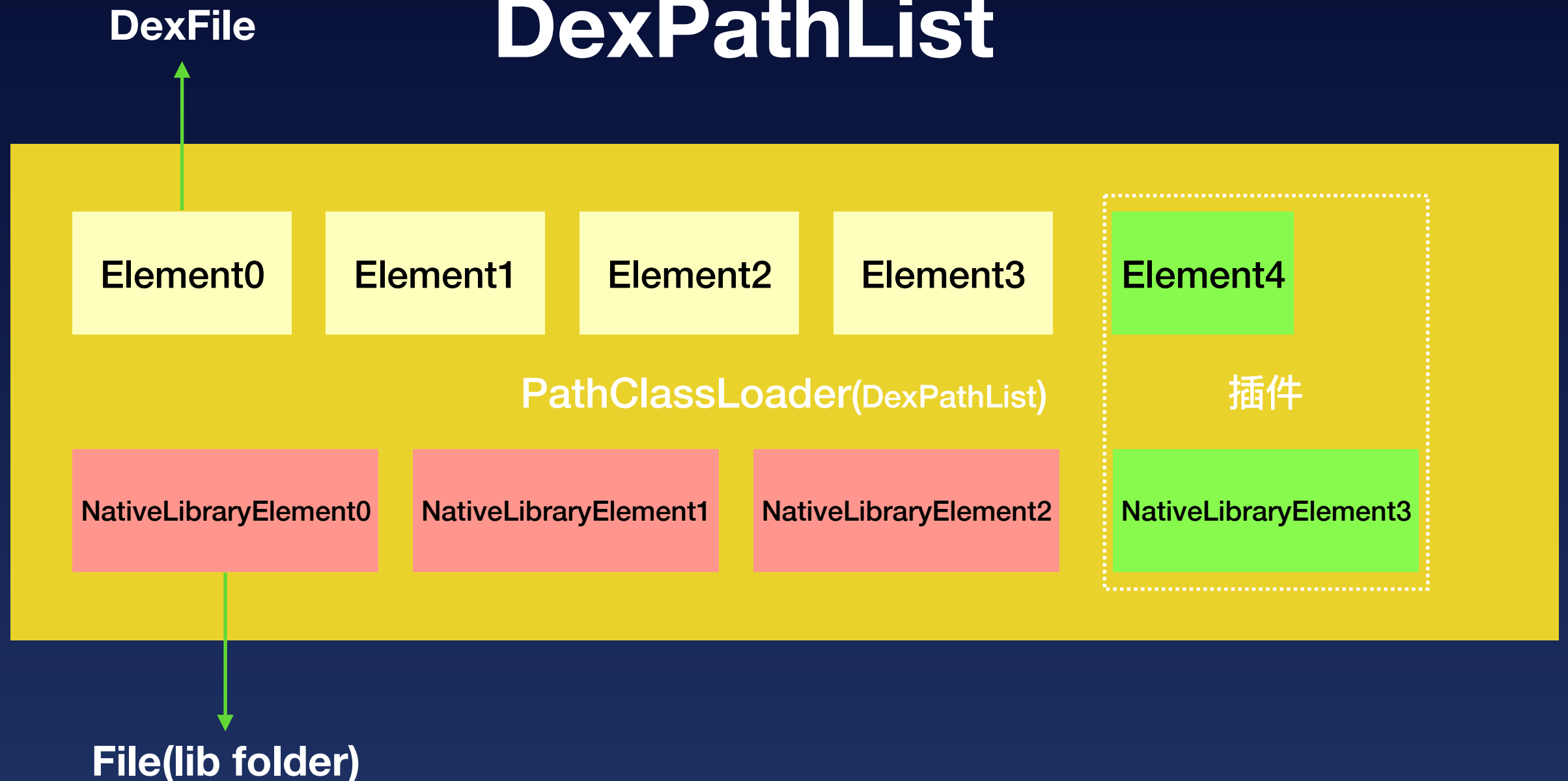


国内插件化原理简介

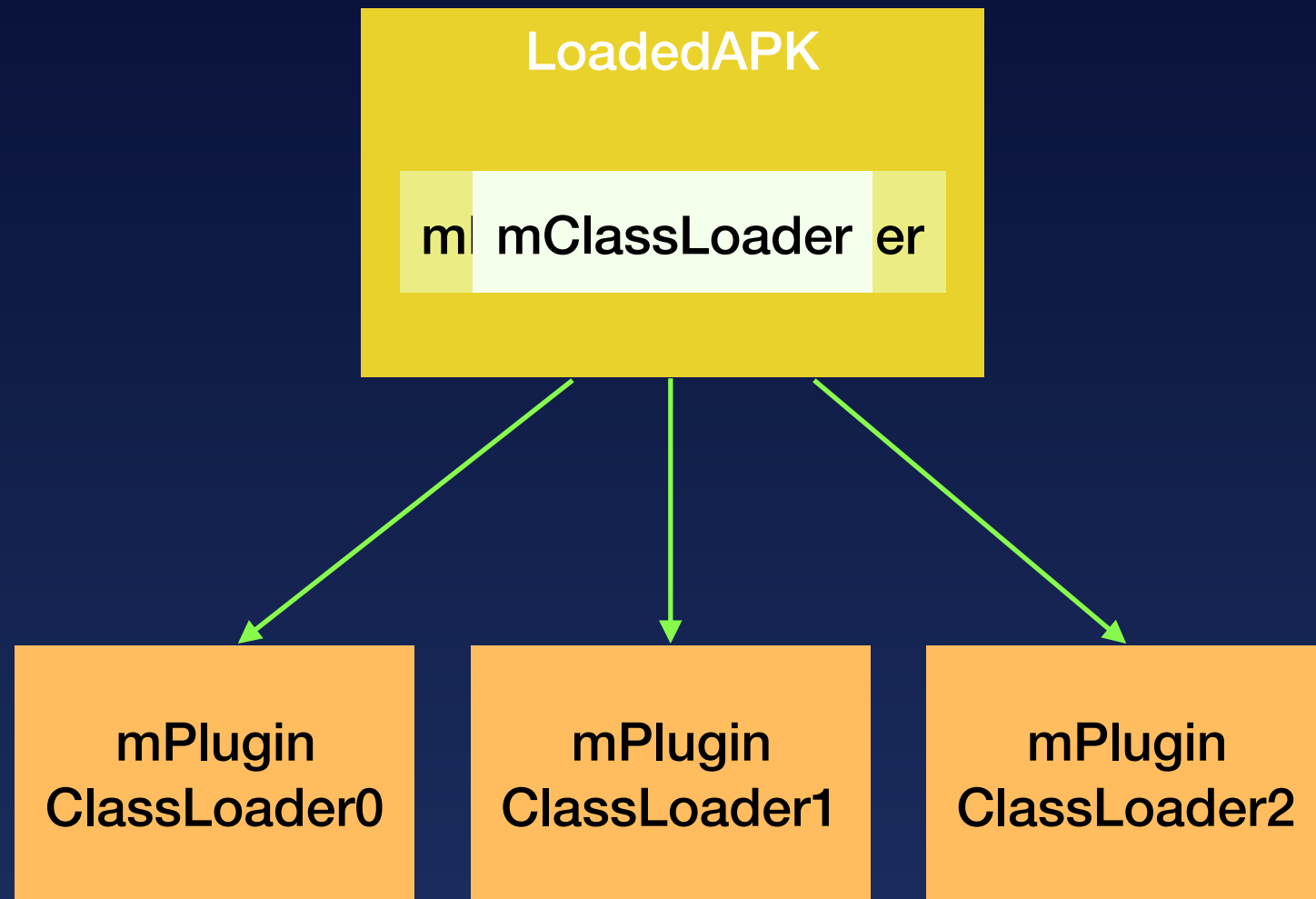


单类加载器

DexPathList



多类加载器



资源加载

P	P	T	T	E	E	E	E
7	F	0	2	0	0	0	1

AAPT | AAPT2

资源加载调用AssetManager#addAssetPath(String)即可

注：Android 5.0之前无法动态增加资源路径

AssetManager中已加载资源路径：

系统资源

/system/framework/framework-res.apk (PP:01)
/system/framework/framework-res-hwext.apk (PP:02)
/product/overlay/frameworkResOverlay.apk (PP:03)

App资源

/data/app/com.iqiyi.test-tX6rCsTDr08d9EOfzYy5ug==/base.apk (PP:7F)

插件资源

/data/user/0/com.iqiyi.test/app_split/plugin.apk (PP:7E)

/system/framework/framework-res.apk (PP:01)

/system/framework/framework-res-hwext.apk (PP:02)

/product/overlay/frameworkResOverlay.apk (PP:03)

/data/user/0/com.iqiyi.test/app_split/plugin.apk (PP:7F)

资源隔离：每个插件拥有独立Resources对象，无法直接访问宿主资源。

/system/framework/framework-res.apk (PP:01)

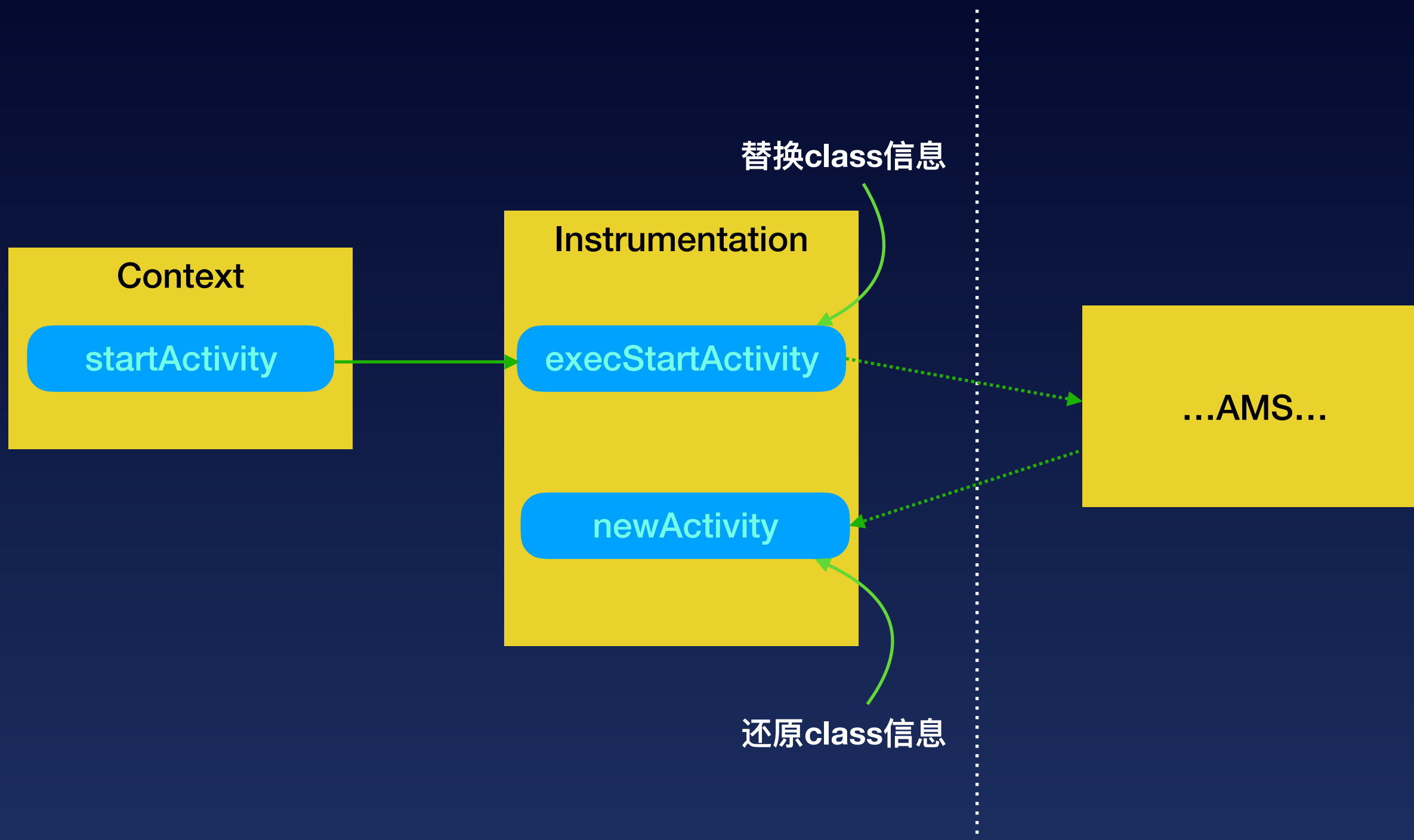
/system/framework/framework-res-hwext.apk (PP:02)

/product/overlay/frameworkResOverlay.apk (PP:03)

/data/app/com.iqiyi.test-tX6rCsTDr08d9EOfzYy5ug==/base.apk (PP:7F)

/data/user/0/com.iqiyi.test/app_split/plugin.apk (PP:7E)

资源分区：插件和宿主共用Resources对象，可互相访问资源。



采用插桩预埋方式启动未在宿主AndroidManifest文件注册的Activity

Android App Bundles原理

Split APKs

Android 5.0

Framework层功能

目的：减少APK体积

特点1：签名相同

特点2：同一应用

Split APKs安装

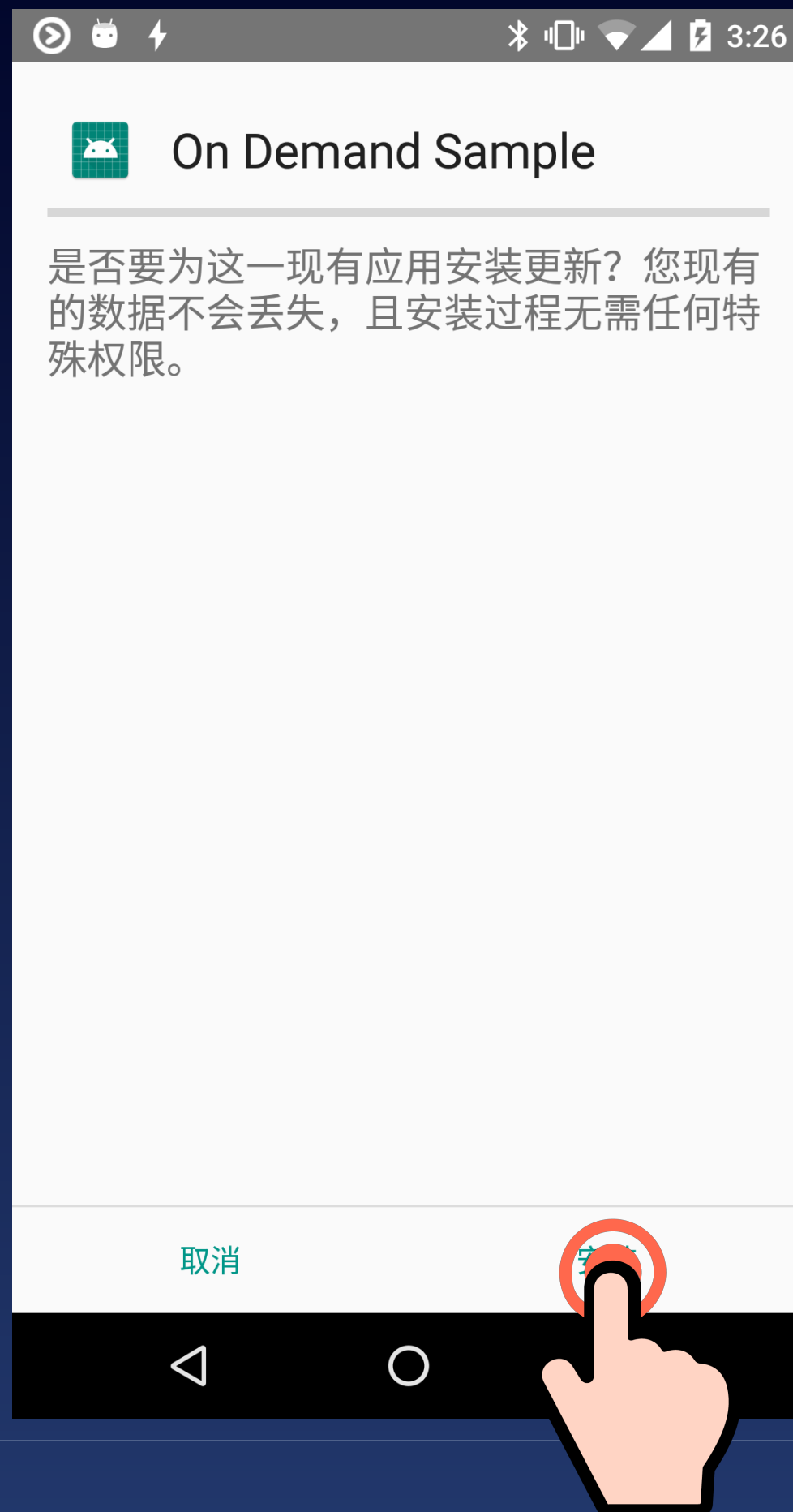
ADB安装

adb install [apk]

adb install-multiple [base-apk, split1-apk]

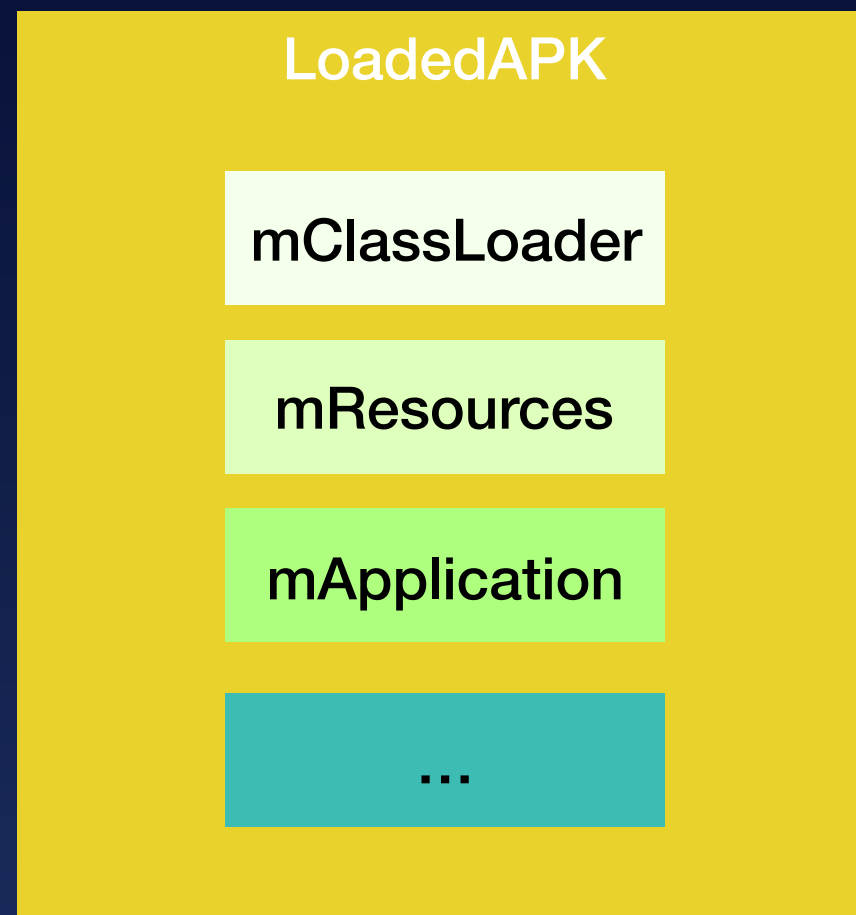
PackageInstaller安装

第三方应用利用PackageInstaller 可在应用运行期间安装Split APKs



系统应用可以静默安装Split APKs
系统版本低于7.0，需重启才能生效

Split APKs加载



LoadedAPK: 用于保存当前已加载APK的本地状态

Split APKs路径集合



```
private void createOrUpdateClassLoaderLocked(List<String> addedPaths) {
```

```
...
```

```
if (mClassLoader == null) {
```

创建PathClassLoader

```
...
```

```
    mClassLoader = ApplicationLoaders.getDefault().getClassLoader(zip,  
        mApplicationInfo.targetSdkVersion, isBundledApp,  
librarySearchPath,  
        libraryPermittedPath, mBaseClassLoader,  
        mApplicationInfo.classLoaderName);
```

```
...
```

```
}
```

```
...
```

```
if (addedPaths != null && addedPaths.size() > 0) {
```

增加Split APKs路径

```
    final String add = TextUtils.join(File.pathSeparator, addedPaths);  
    ApplicationLoaders.getDefault().addPath(mClassLoader, add);  
    // Setup the new code paths for profiling.  
    needToSetupJitProfiles = true;
```

```
}
```

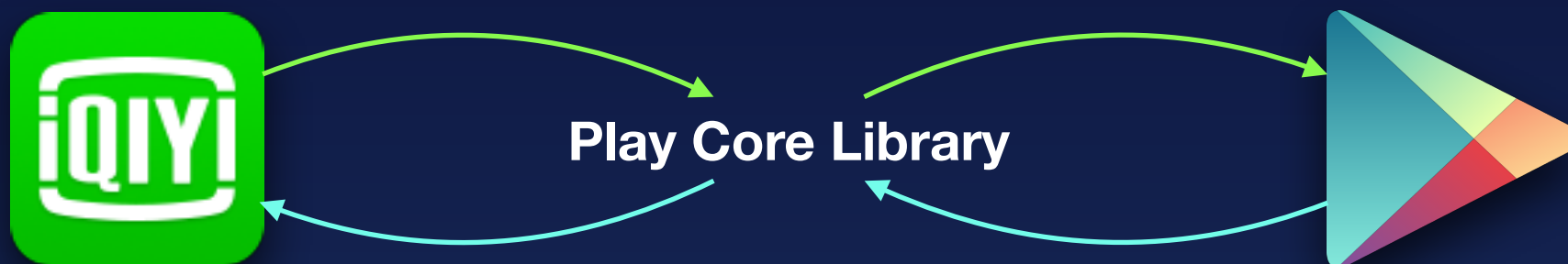
```
...
```

```
}
```



```
public Resources getResources() { Split APKs资源路径数组
    if (mResources == null) {
        final String[] splitPaths;
        try {
            splitPaths = getSplitPaths(null);
        } catch (NameNotFoundException e) {
            throw new AssertionError("null split not found"); 创建Resources
        }
        mResources = ResourcesManager.getInstance().getResources(null,
        mResDir, splitPaths, mOverlayDirs, mApplicationInfo.sharedLibraryFiles,
        Display.DEFAULT_DISPLAY, null, getCompatibilityInfo(),
        getClassLoader());
    }
    return mResources;
}
```

使用*Play Core Library*安装插件



PlayCoreLibrary提供SplitCompat模式让app可立即使用插件。

SplitCompat与国内插件化框架工作原理类似

- 单类加载器方式加载插件代码
- *AssetManager#addAssetPath(String)* 加载资源。



What's Qigsaw?

Qigsaw是利用AAB开发套件，
爱奇艺自研的一套动态组件化方案

Why Qigsaw?

开发者希望插件和基线
能在同一工程协作开发

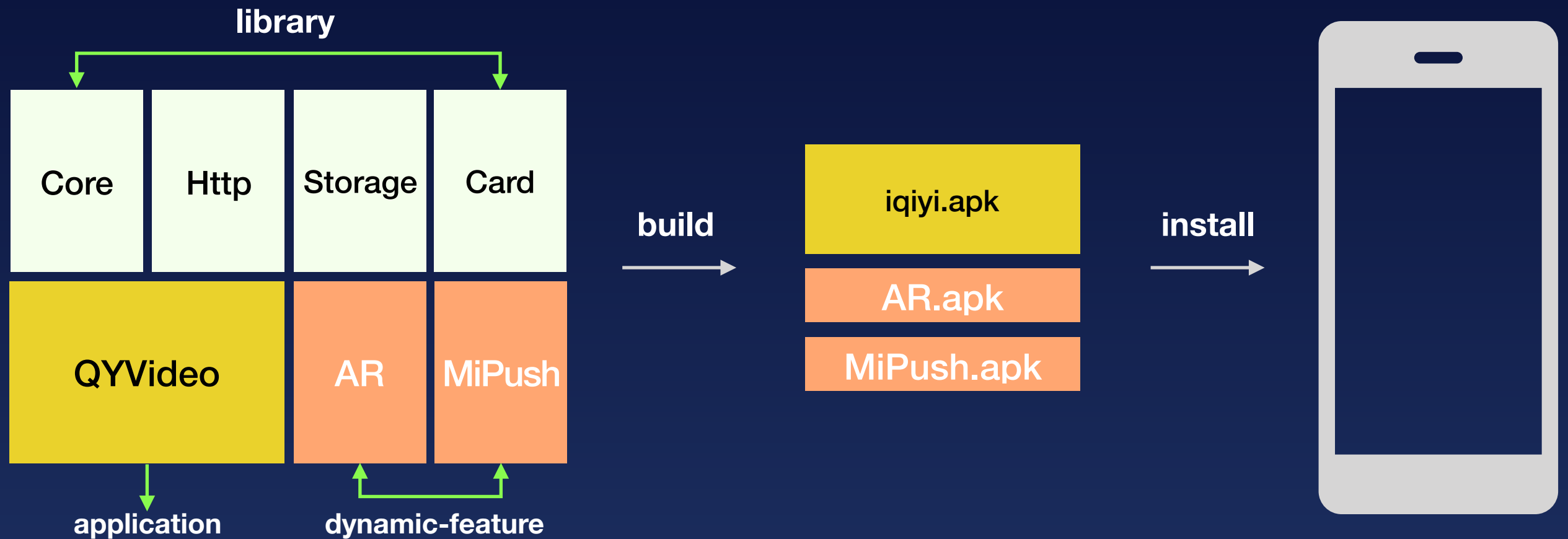
Android P私有API访问限制

利用AAB强大开发套件

便于走向国际

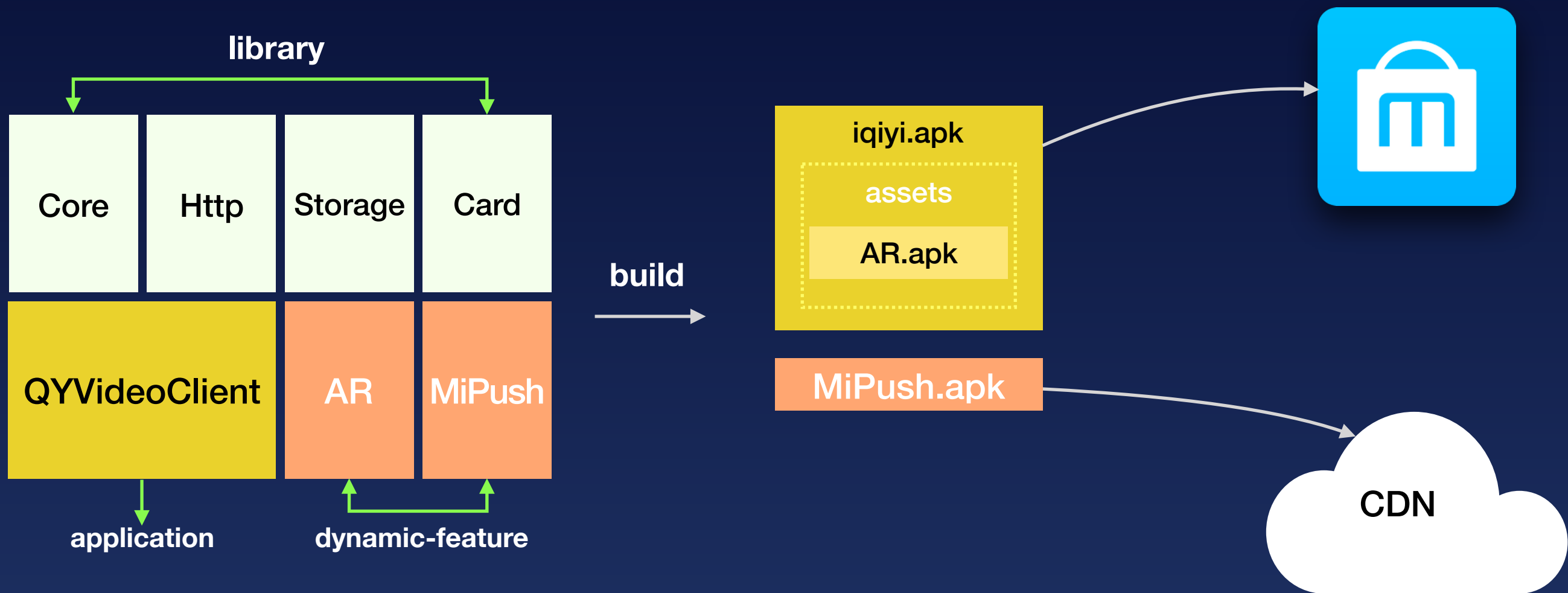
Qigsaw开发初体验

Debug阶段



adb install-multiple [base-apk, split1-apk]

Release阶段



Qigsaw提供打包插件让开发者享受一条龙服务

Qigsaw核心优势

“山寨”*Play Core Library*公开接口实现 开发者阅读官方文档即可愉快开发

1 Hook, 少量私有API访问 (grey list)

利用AAB开发工具 领略极速开发体验

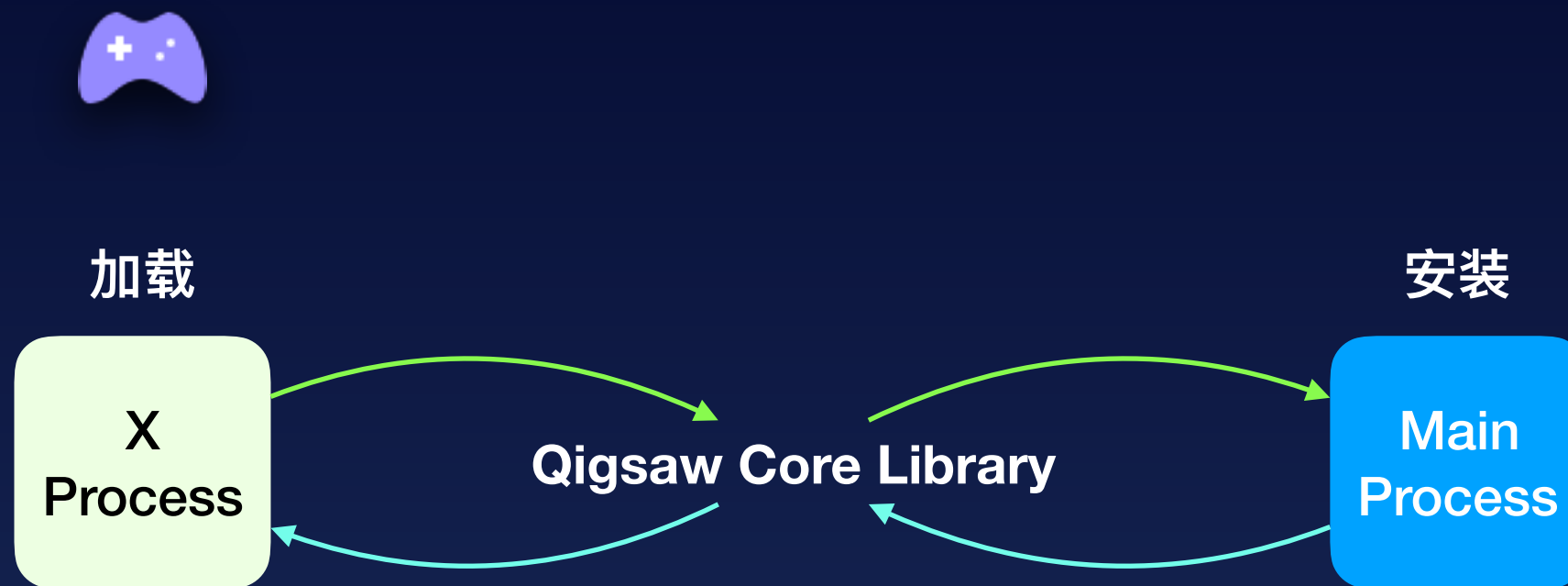
支持Android 4.0及以上
任何进程均可加载插件

国际化应用可无缝切换至AAB

对比其他插件化框架， 有啥优势！？

接入使用超级方便，
用过的朋友都说好！

Qigsaw插件安装加载过程



QigsawCoreLibrary提供SplitCompat模式让进程可立即使用插件。

Qigsaw插件加载

- 代码加载采用单类加载器方式。
- 资源加载使用Qigsaw独有方式加载。

Qigsaw资源加载

```
public class MyActivity extends Activity {  
  
    @Override  
    protected void onCreate(@Nullable Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
    }  
  
    @Override  
    public Resources getResources() {  
        SplitInstallHelper.loadResources(this, super.getResources());  
        return super.getResources();  
    }  
}
```

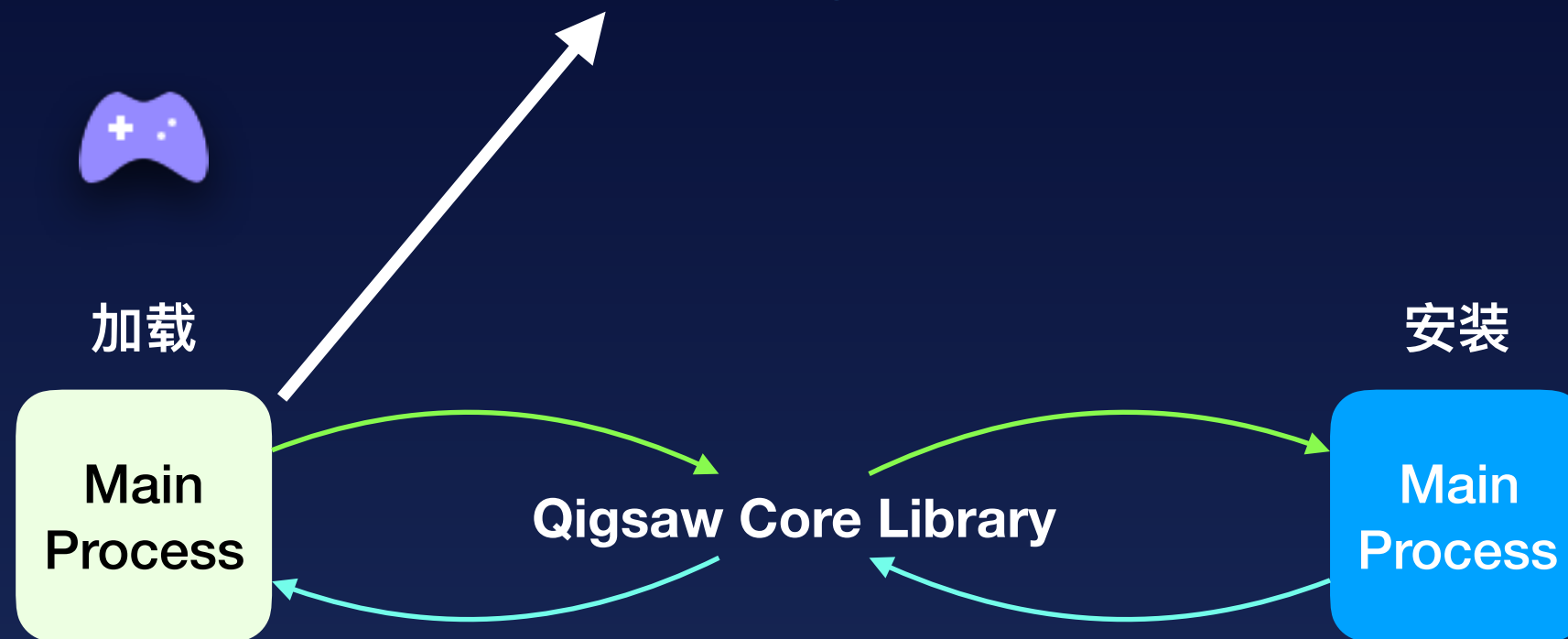
为避免开发者重复写大量模板代码，Qigsaw提供插件在打包过程中插入模板代码。

拓展功能

(一) 多进程支持

GameActivity01 (main process)

GameActivity02 (game process)



哪个进程发起插件安装请求，就在哪个进程加载插件

Unfortunately, App has stopped.

REPORT

OK

插件在“:game”进程未加载导致 ClassNotFoundException

解决方案

(一)

进程启动阶段加载所有已安装插件

Application#attachBaseContext

(二)

Hook PathClassLoader

(二) 续

当出现ClassNotFoundException时

- 判断异常类是否为插件四大组件。
- 如果是，则加载所有已安装但未加载插件。
- 当加载完成后，依然出现类找不到异常。
- 返回空四大组件类防止进程崩溃。

(二) 支持插件Application

在插件启动时创建Application实例，
并调用对应生命周期方法。

(三) 支持插件Provider

在进程启动时候移除插件所有Provider，
在插件启动时再进行Provider安装。

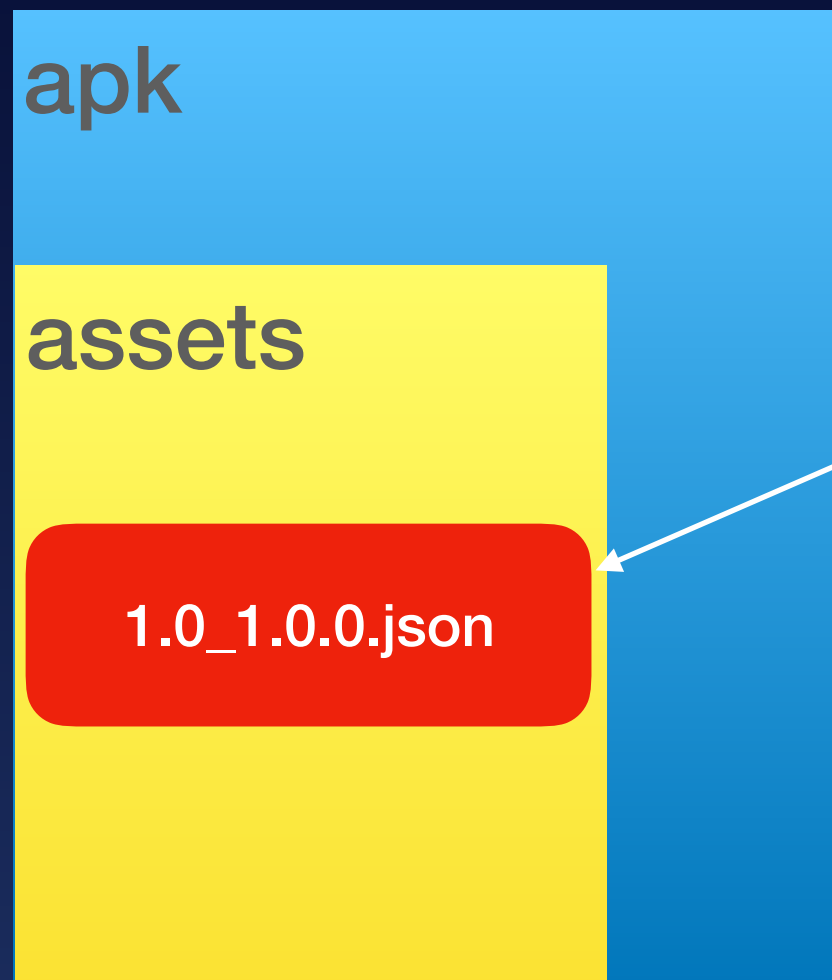
与Tinker融合

通过Tinker Patch下发插件更新



该文件记录插件信息
包括下载地址

```
1  {
2    "qigsawId": "1.0_146959974",
3    "appVersionName": "1.0",
4    "splits": [
5      {
6        "splitName": "java",
7        "url": "assets://java.apk",
8        "builtIn": true,
9        "size": 17082,
10       "version": "1.1@1",
11       "workProcesses": [
12         "silk",
13         ""
```



如果该文件更新了，
那么插件也就更新了。

Tinker开启资源修复即可。

Qigsaw在爱奇艺App的使用



小程序

🔍 发射2048

我的小程序

您还没有添加小程序哦 >

爱奇艺家族

全部 >



商城



奇秀



游戏直播



轻小说

为你推荐

全部 >



看图猜成语



成语小秀才



守望者



发射2048

最近使用



一笔画英雄



戏多多配音社区

登录后小程序数据可多设备同步

立即登录

从19年年初上线至今，
踩过无数坑，经受数亿用户检验



→ 0%

Qigsaw在爱奇艺公司的使用



开源计划



Qigsaw开源交流群



该二维码7天内(6月25日前)有效, 重新进入将更新

Q&A

THANKS

