

Small: 插件化轻巧之道

林光亮

0x00

诞生

0x01

轻

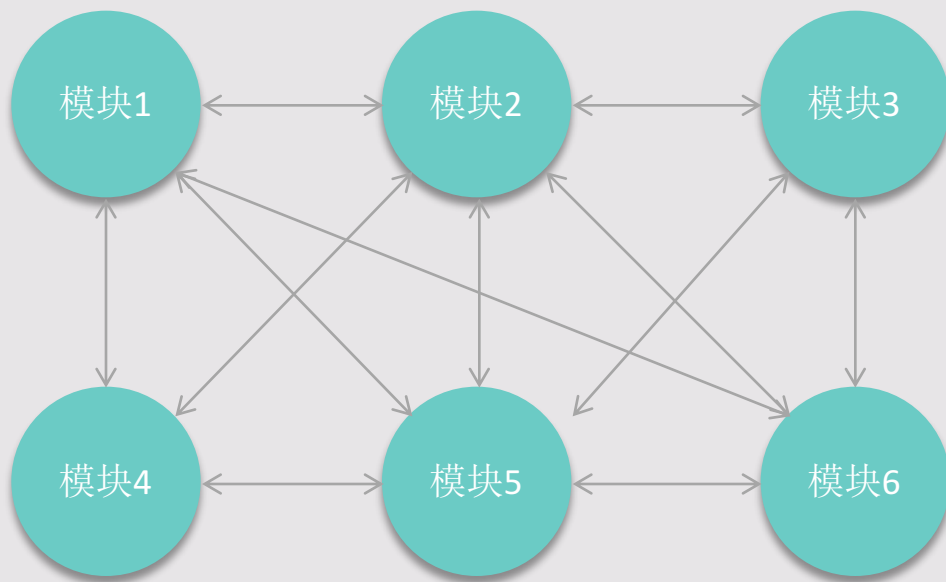
0x02

巧

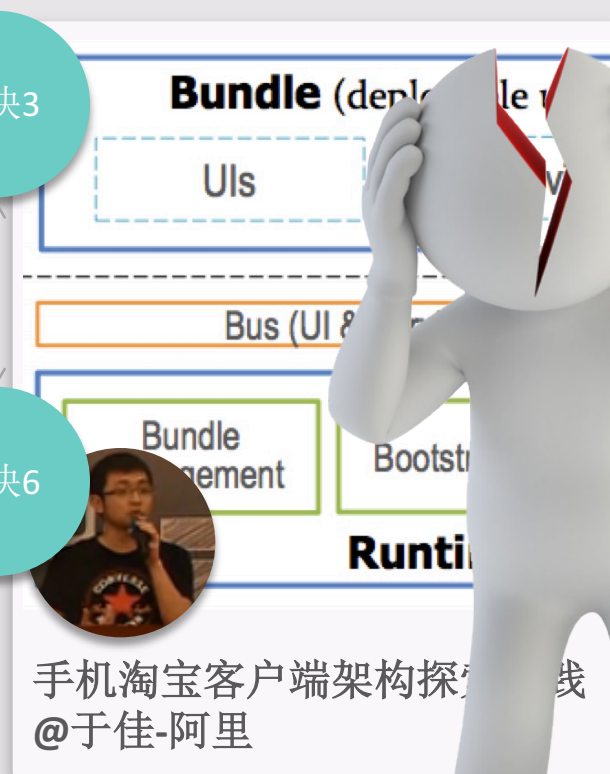
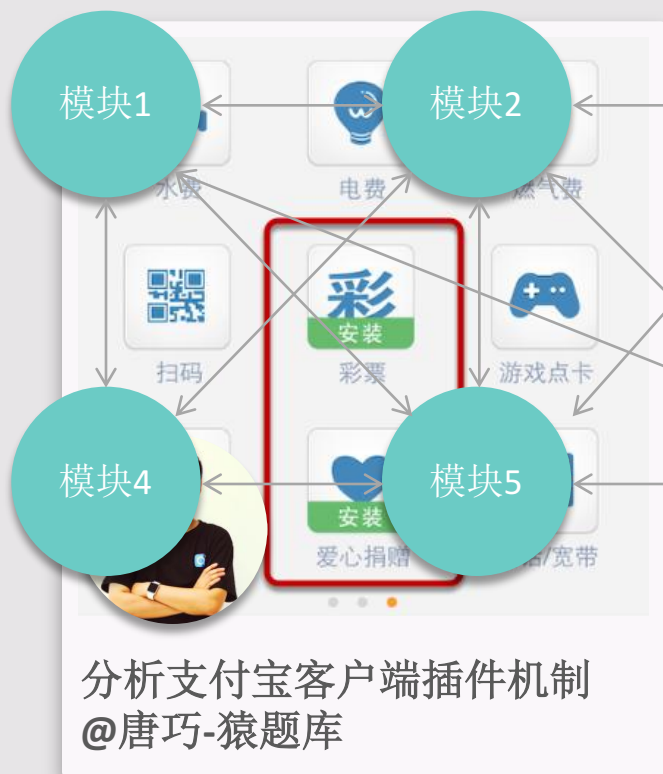
0x03

TODO

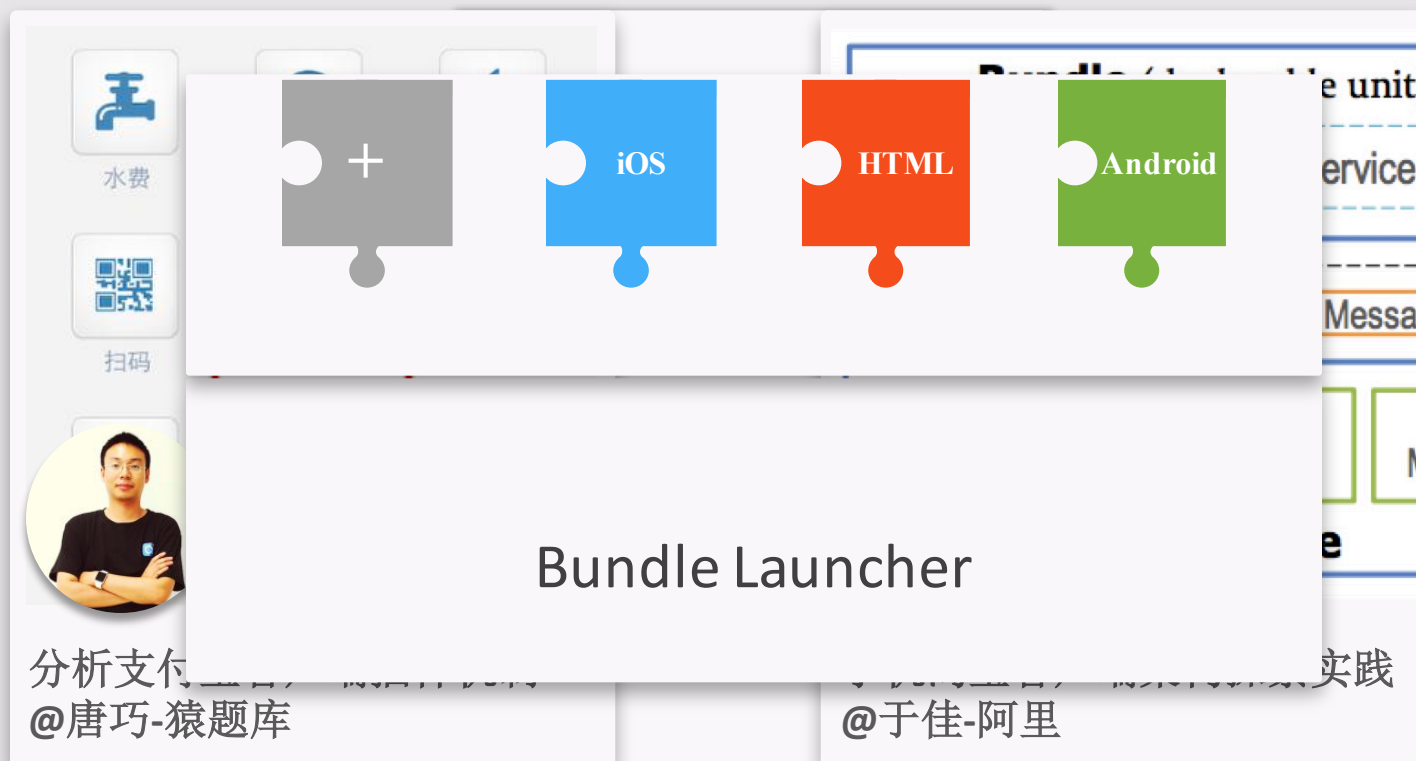
诞生



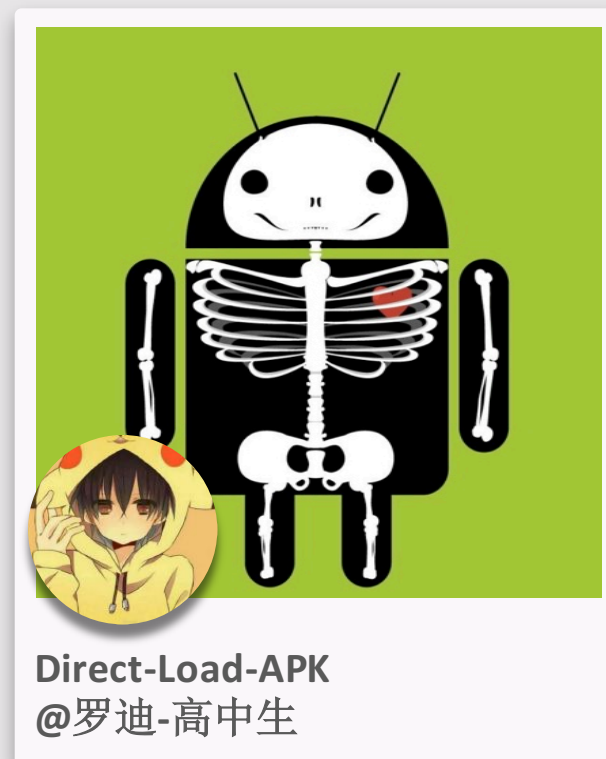
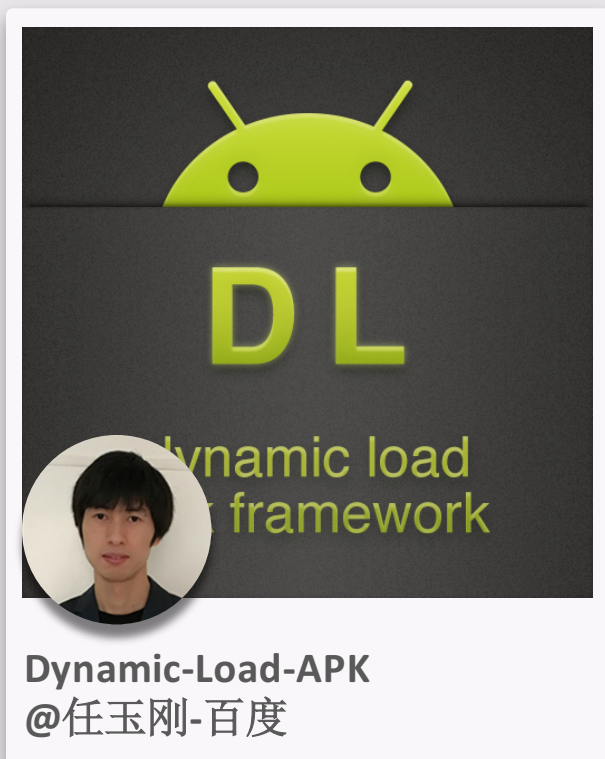
诞生



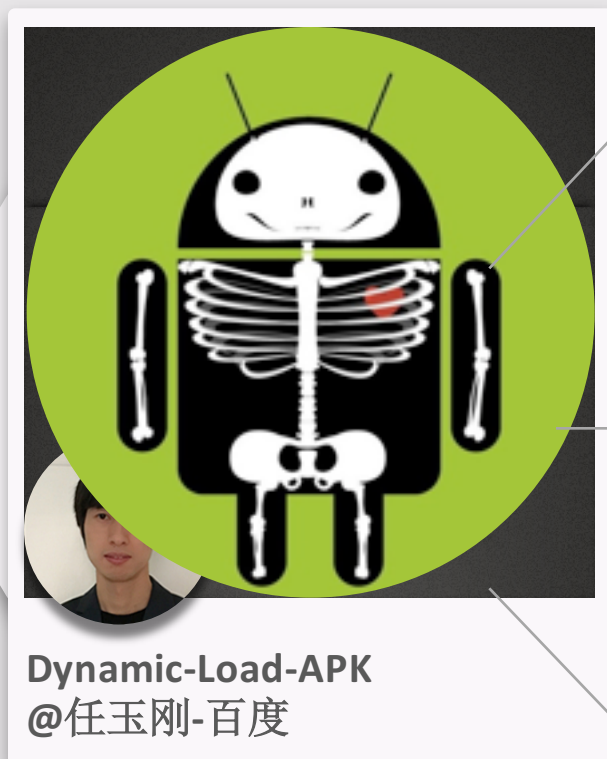
诞生



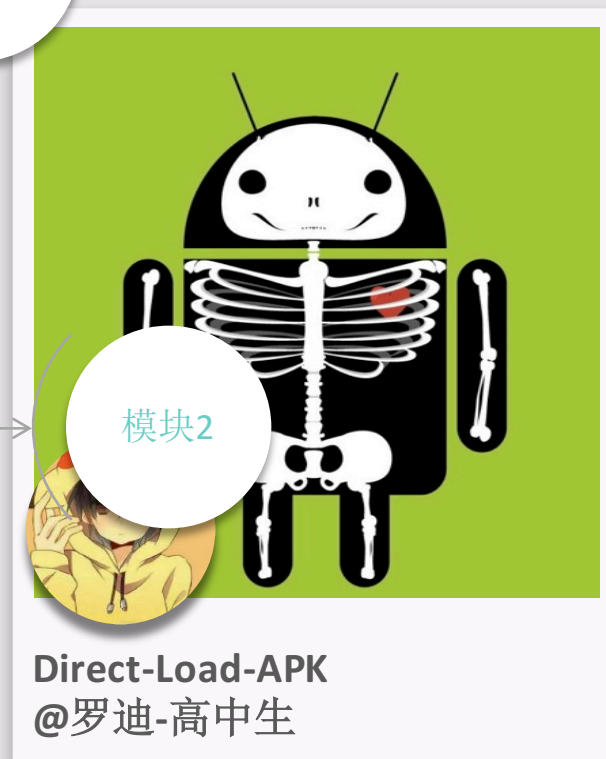
诞生



诞生

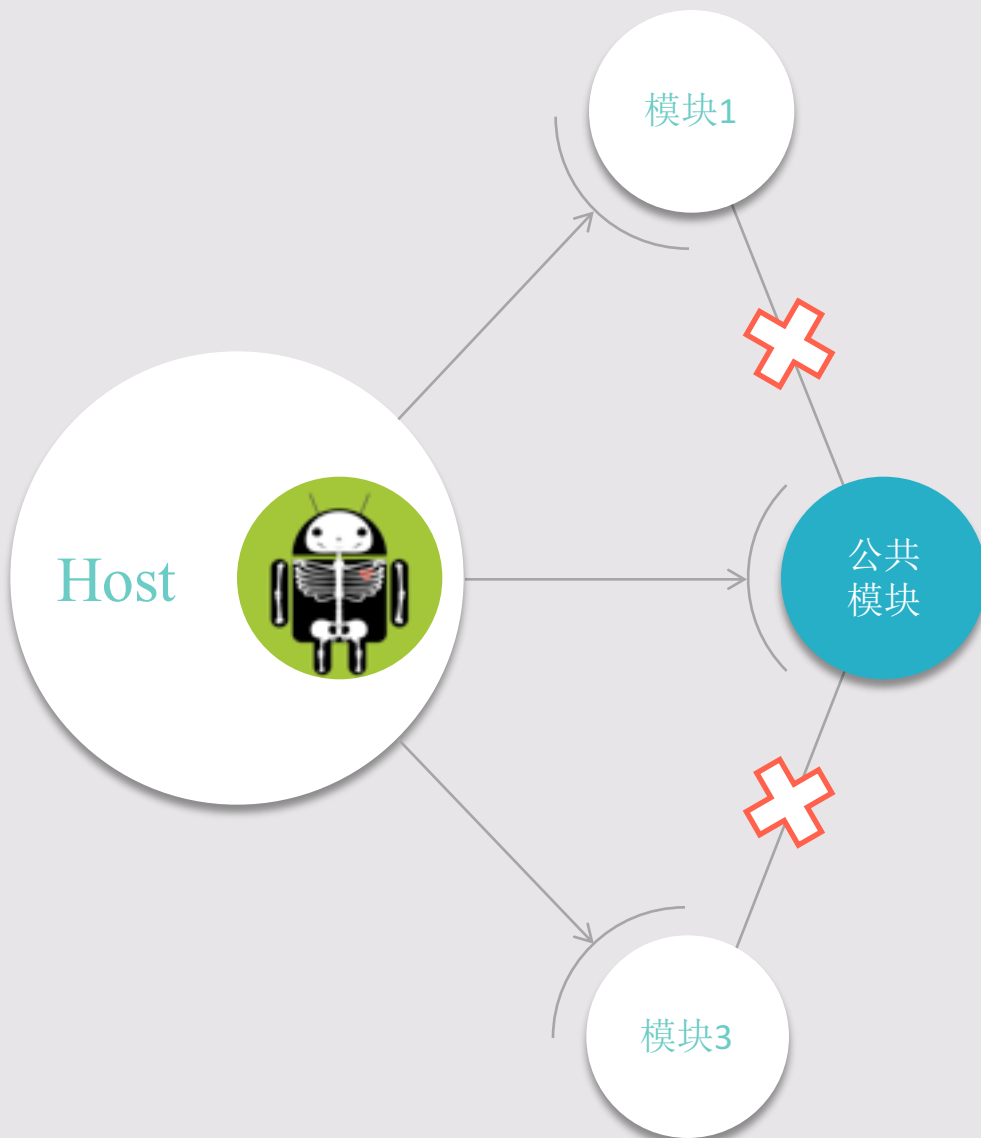


模块1

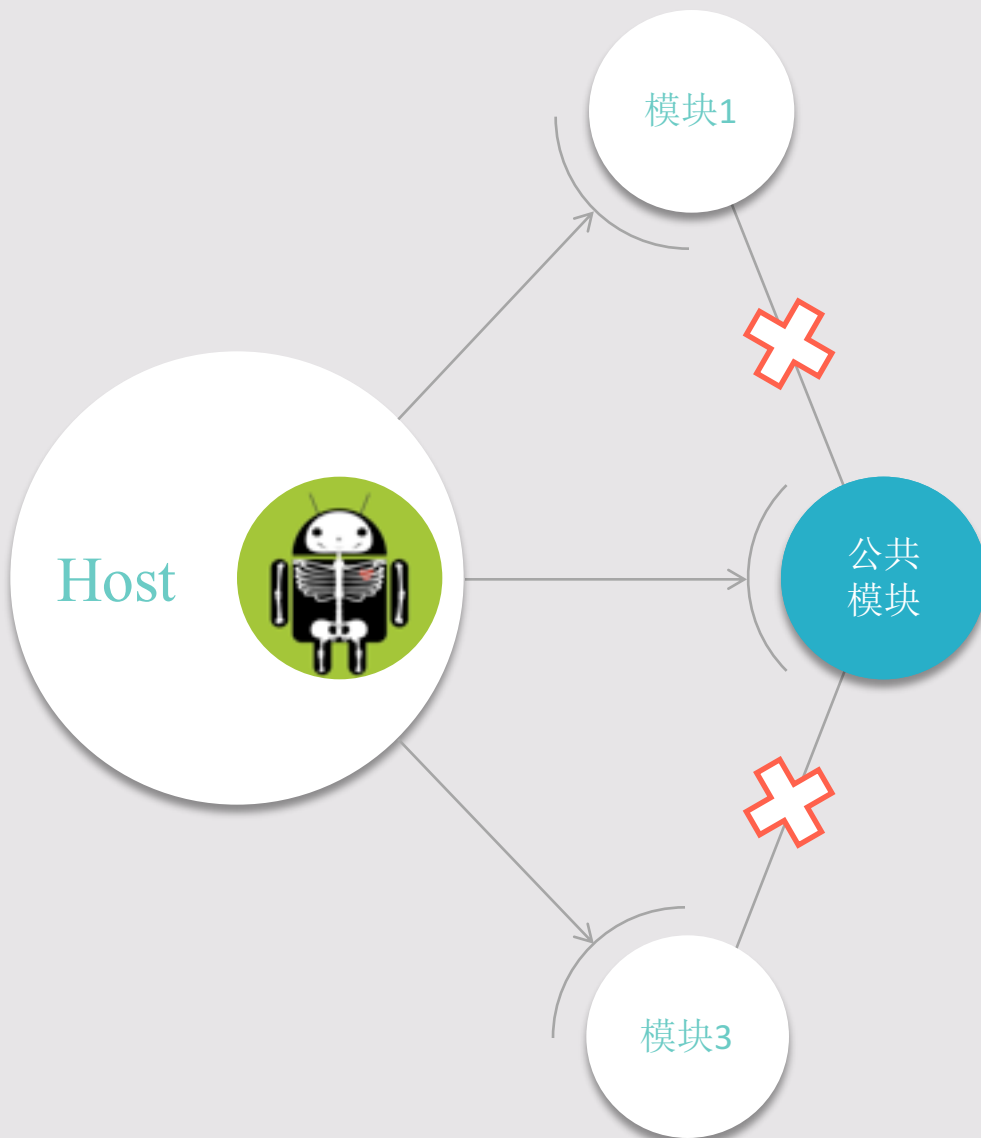


模块3

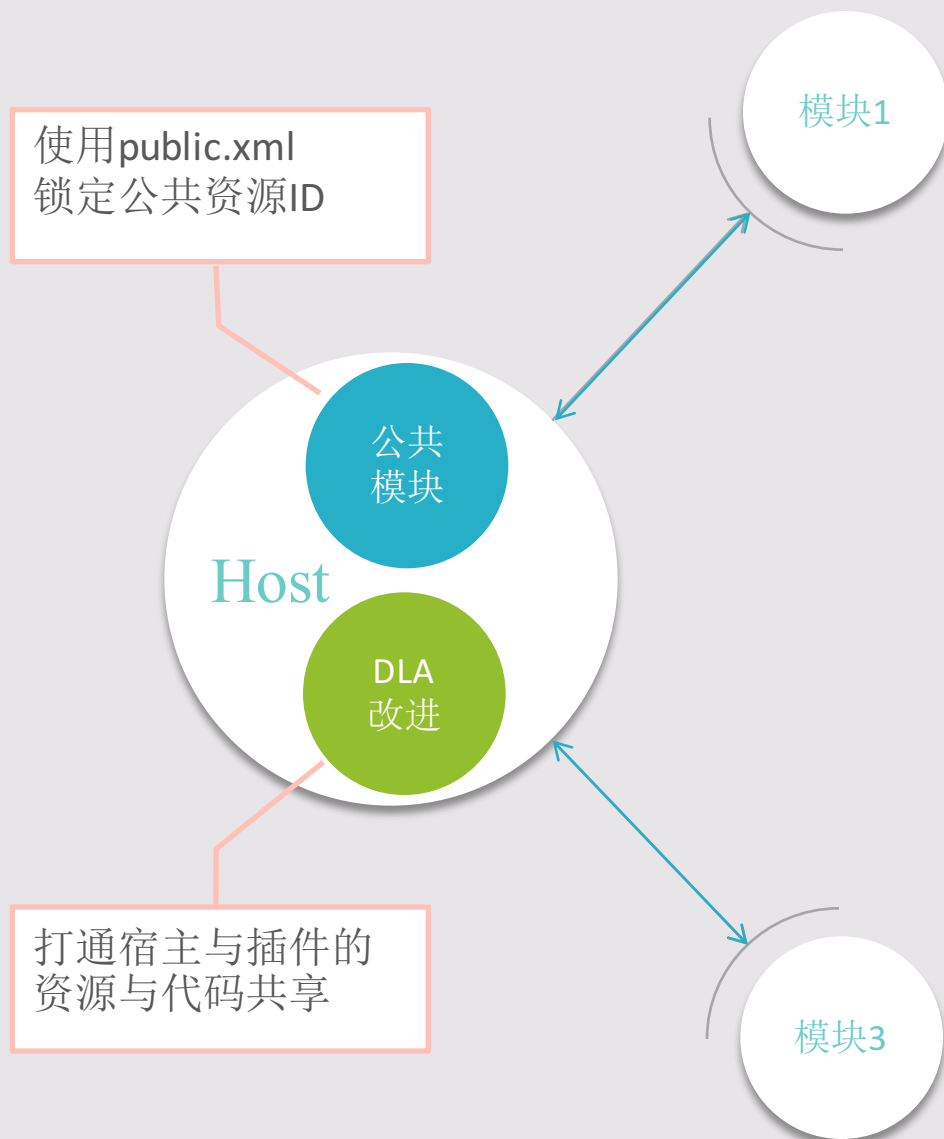
诞生



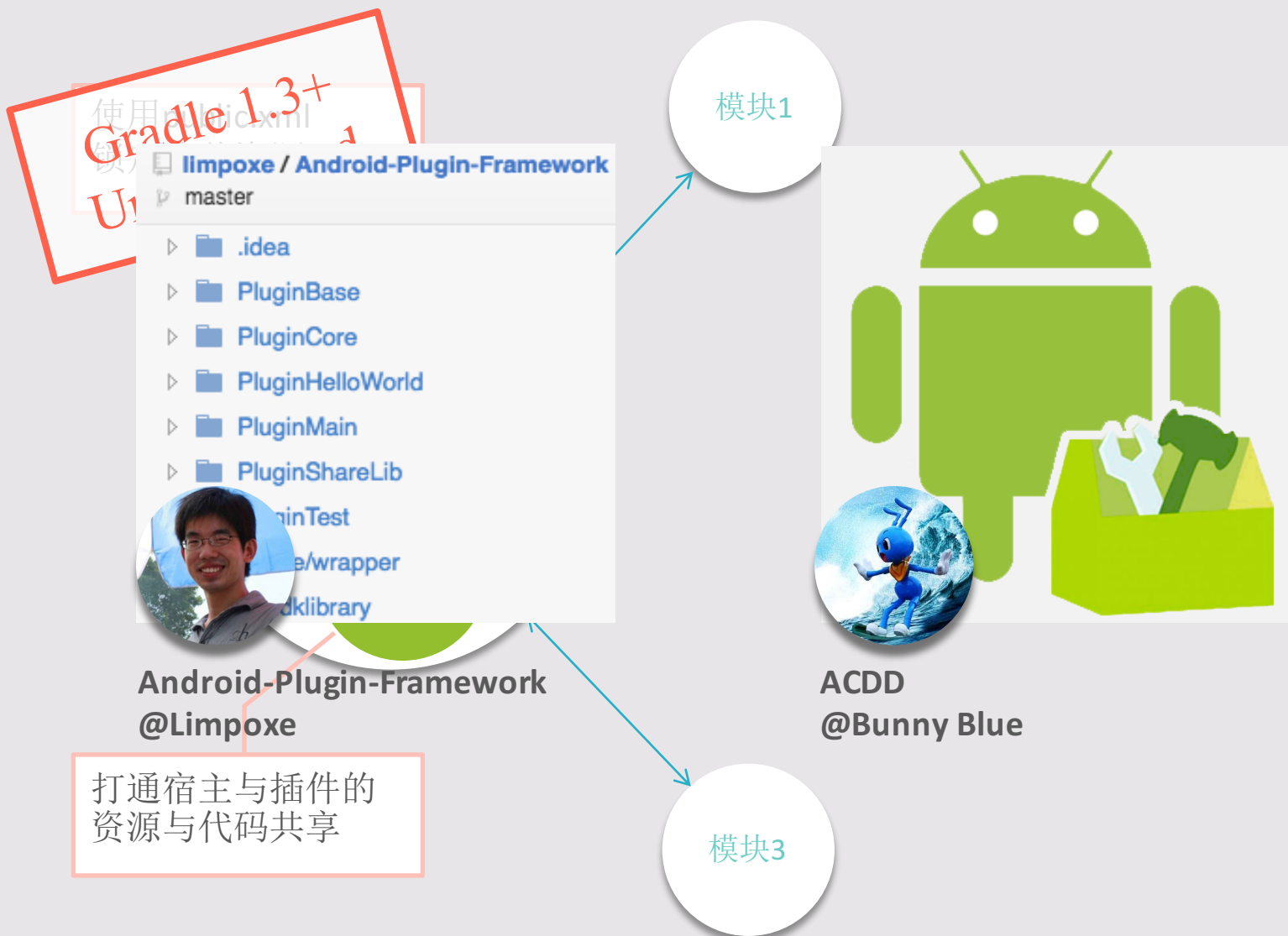
诞生



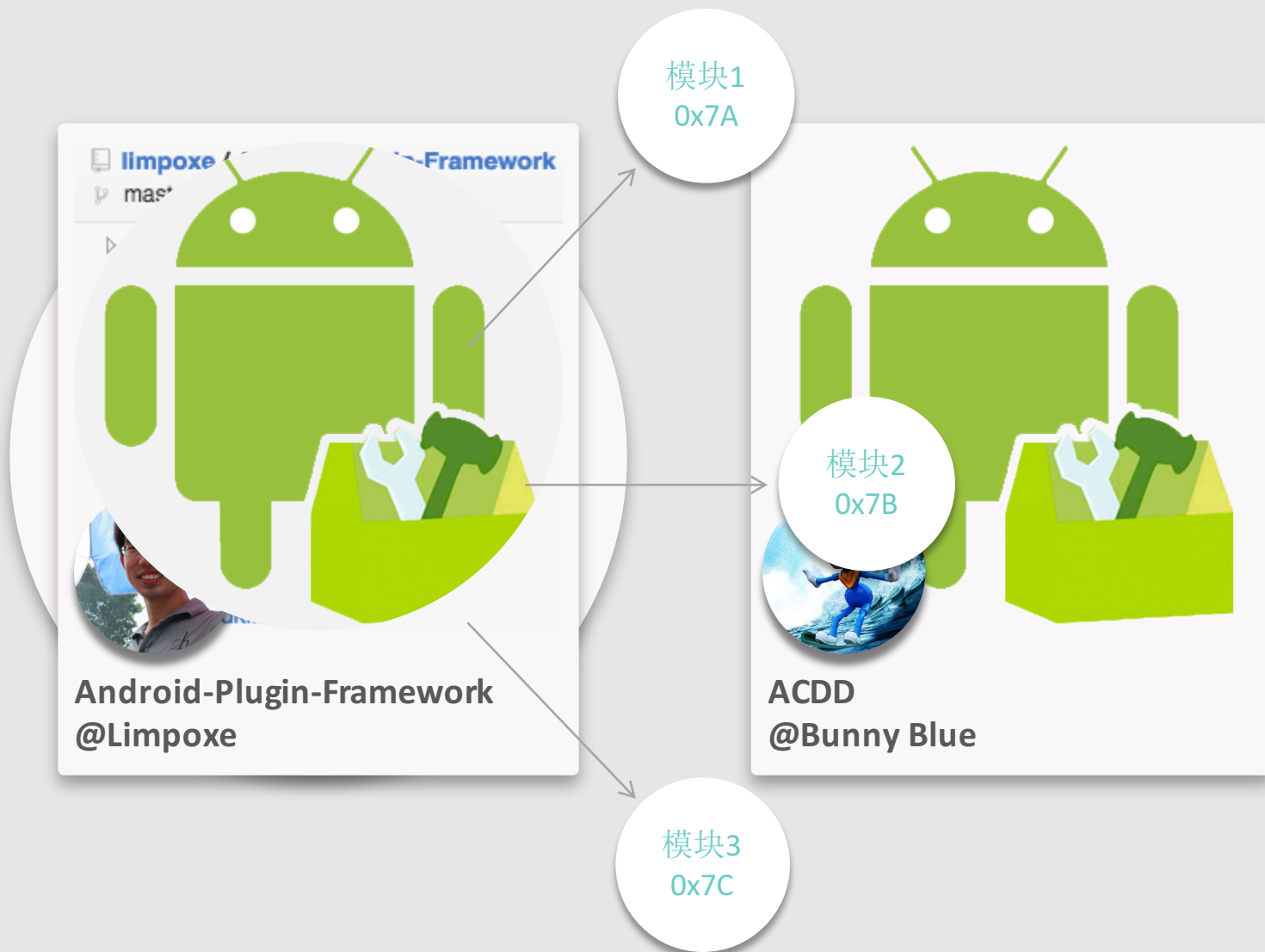
诞生



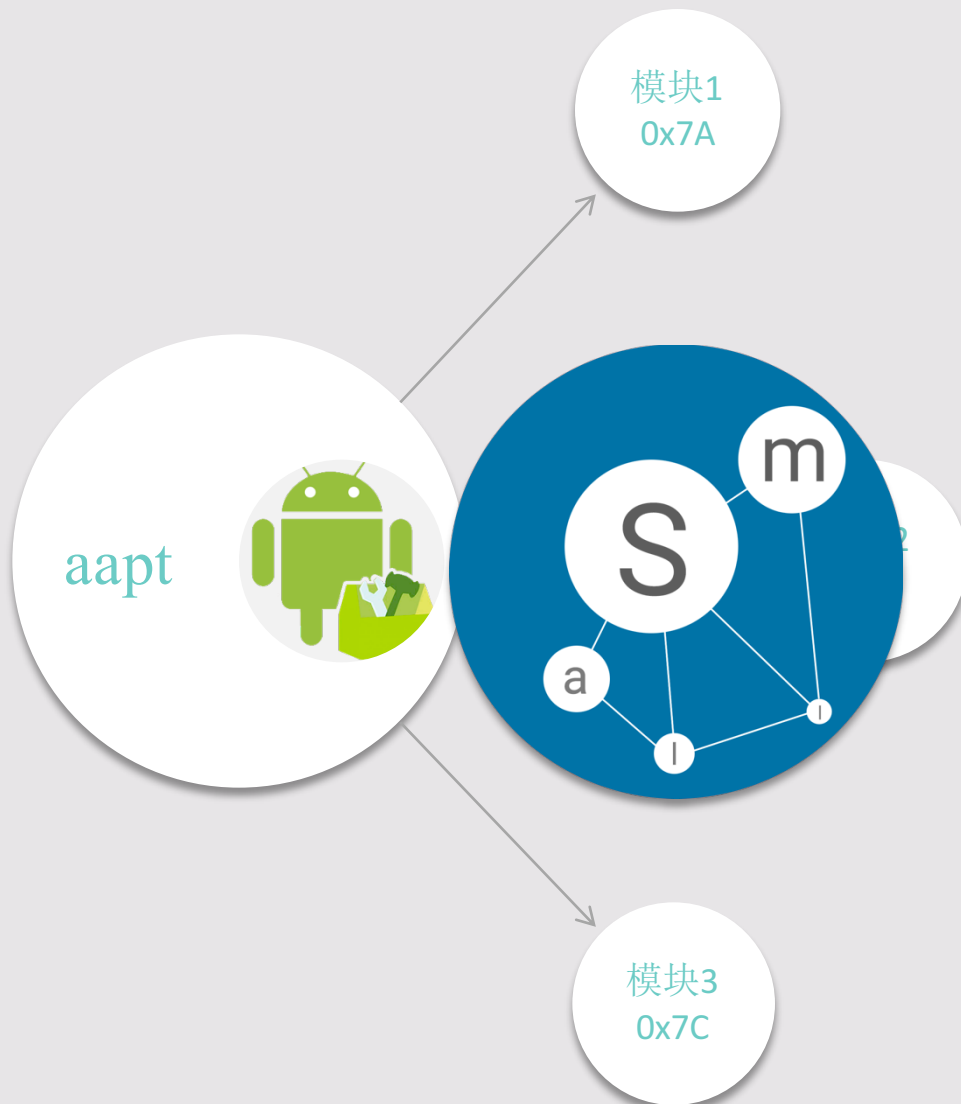
诞生



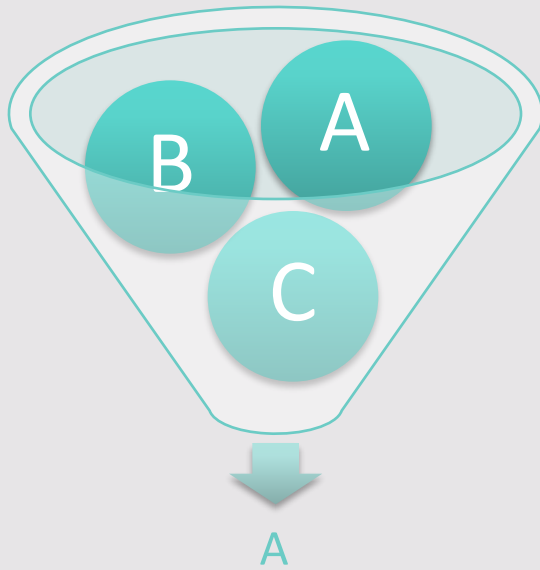
诞生



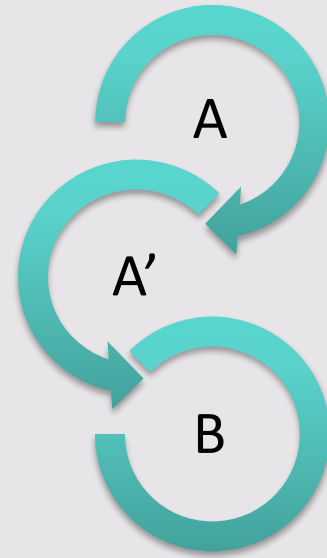
诞生



轻

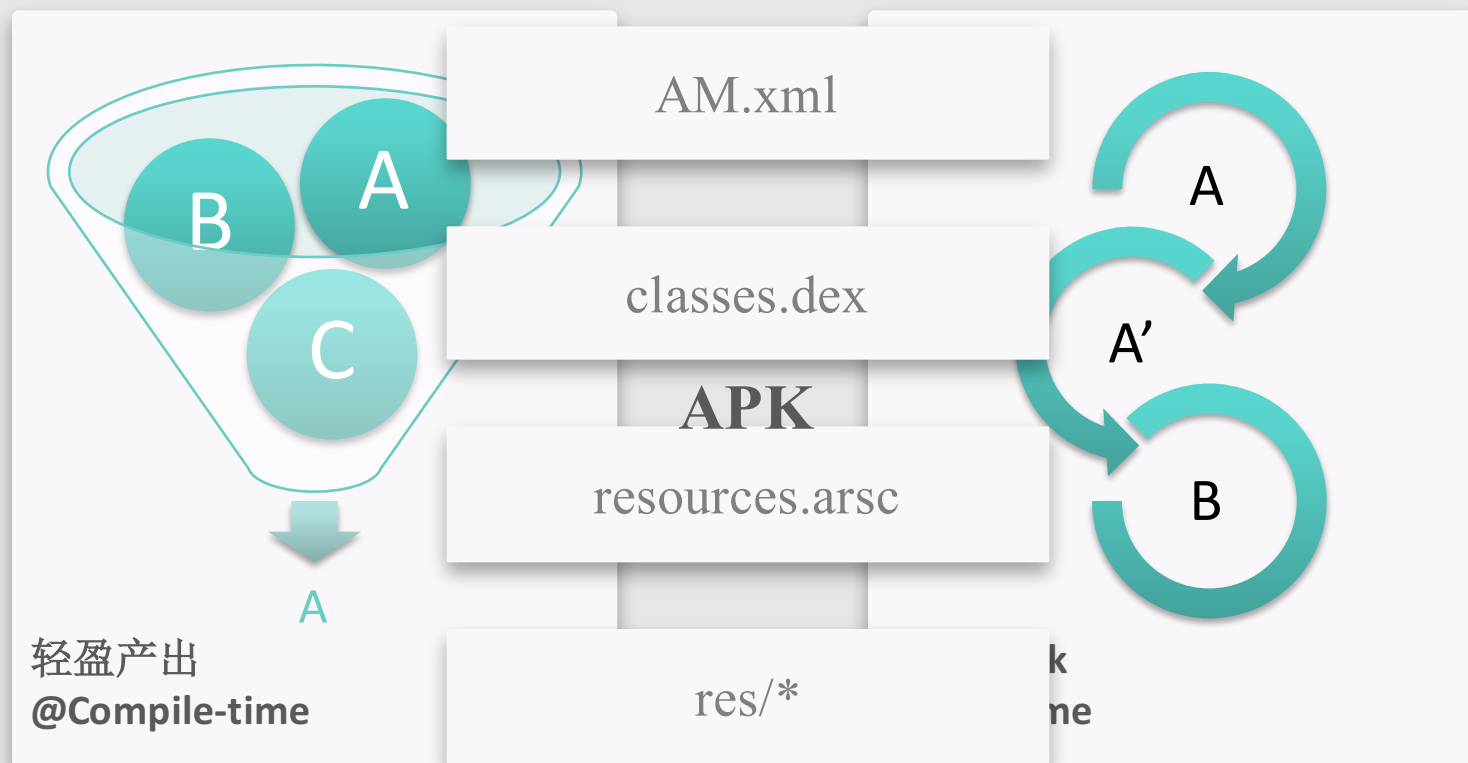


轻盈产出
@Compile-time



轻度Hook
@Run-time

轻/轻盈产出



轻/轻盈产出

拆分粒度/方案

文本

AM\$1

AM.xml

AM\$2

文件(*.jar)

dex\$1

classes.dex

dex\$2

二进制

arsc\$1

resources.arsc

arsc\$2

文件(*.xml/png)

res\$1

res/*

res\$2

轻/轻盈产出

```
0000000: 0200 0c00 4804 0000 0100 0000 0100 1c00 .....H.....
0000010: 8c00 0000 0400 0000 0000 0000 0001 0000 .....
0000020: 2c00 0000 0000 0000 0000 0000 2500 0000 ,.....%...
0000030: 4800 0000 5700 0000 2222 7265 732f 6d69 H...W..."res/mi
0000040: 706d 6170 2d68 6470 692d 7634 2f69 635f pmap-hdpi-v4/ic_
0000050: 6c61 756e 6368 6572 2e70 6e67 0020 2072 launcher.png. r
0000060: 6573 2f6d 6970 6d61 702d 6864 7069 2d76 es/mipmap-hdpi-v
0000070: 342f 6963 5f70 6c75 6769 6e2e 706e 6700 4/ic_plugin.png.
0000080: 0c0c 4c65 6172 6e69 6e67 4172 7363 0006 ..LearningArsc..
0000090: 0650 6c75 6769 6e00 0002 2001 b003 0000 .Plugin... ..
00000a0: 7f00 0000 6e00 6500 7100 0000 7700 6500 ....n.e.t...w.e.
00000b0: 7100 7500 6900 6300 7200 0000 7200 7200 q.u.i.c.k...a.r.
00000c0: 7300 6300 0000 0000 0000 0000 0000 0000 s.c.....
00000d0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000e0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000f0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0000100: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0000110: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0000120: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0000130: 0000 0000 0000 0000 0000 0000 0000 0000 .....
```

resources.arsc

轻/轻盈产出/arsc格式

[illegible]

轻/轻盈产出/arsc格式

hex(LE) 小端代码	struct 数据结构
0200	ResTable_header
0100	ResStringPool_header
0002	ResTable_package
0100	ResStringPool_header
0100	ResStringPool_header
0202	ResTable_typeSpec
0102	ResTable_type

轻/轻盈产出/arsc格式

hex(LE) 小端代码	struct 数据结构
0200	ResTable_header
0100	ResStringPool_header
0002	ResTable_package
0100	ResStringPool_header
0100	ResStringPool_header
0202	ResTable_typeSpec
0102	ResTable_type

[illegible]

轻/轻盈产出/arsc格式

hex(LE) 小端代码		struct 数据结构	
0200		ResTable_header	
资源ID	PP 包ID	TT 类型ID	NNNN 项目ID
0x			
0100		ResStringPool_header	
0100		ResStringPool_header	
0202		ResTable_typeSpec	
0102		ResTable_type	

```
table: {
  package: { id: 0x7F, name: "net.wequick.arsc" },
  strings: [
    "res/mipmap-hdpi-v4/ic_launcher.png",
    "res/mipmap-hdpi-v4/ic_plugin.png",
    "LearningArsc",
    "Plugin"
  ],
  typeStrings: [ "attr", "mipmap", "string", "style" ],
  keyStrings: [
    "ic_launcher", "ic_plugin", "app_name",
    "s_plugin", "AppTheme", "PluginTheme"
  ],
  typeSpecs: [
    01 { types: [ ] },
    02 { types: [ Configs@ic_launcher, Configs@ic_plugin ] },
    03 { types: [ Configs@app_name, Configs@s_plugin ] },
    04 { types: [ Configs@AppTheme, Configs@PluginTheme ] }
  ]
}
```

资源ID
并非实际存在

轻/轻盈产出/arsc分离

想象中最简单的分离方式

```
host (0x7f)
|-- mipmap (02)
|   |-- ic_launcher (0000)
|   `-- ic_plugin (0001)
`-- values
    |-- strings.xml (03)
    |   |-- app_name (0000)
    |   `-- s_plugin (0001)
    `-- styles.xml (04)
        |-- AppTheme (0000)
        `-- PluginTheme (0001)
```

```
plugin (0x7f)
|-- mipmap (02)
|   |-- padding_mipmap_0000
|
|-- values
    |-- strings.xml (03)
    |   |-- padding_string_0000
    |
    `-- styles.xml (04)
        |-- padding_style_0000
```

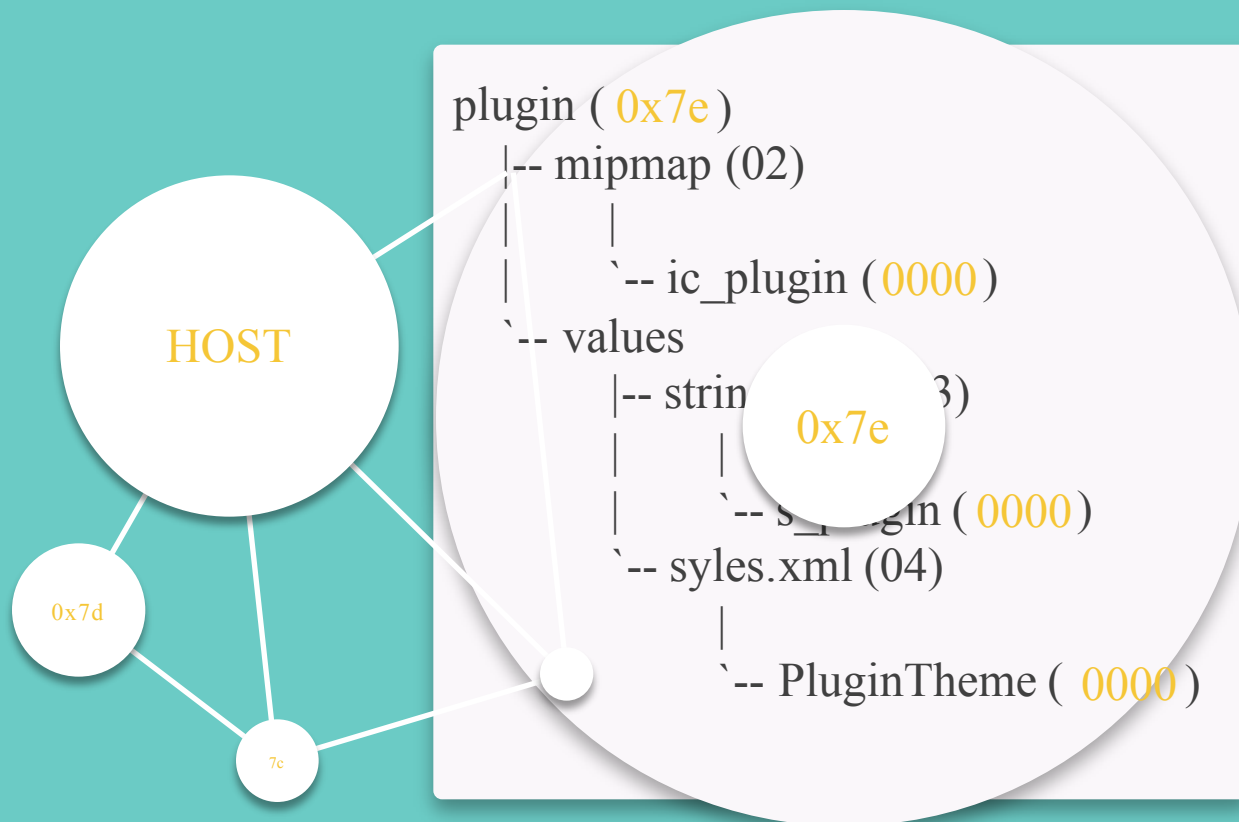
存在问题：必须补齐资源(输出变大)、只能分离一个插件

实践中最极致的分离方式

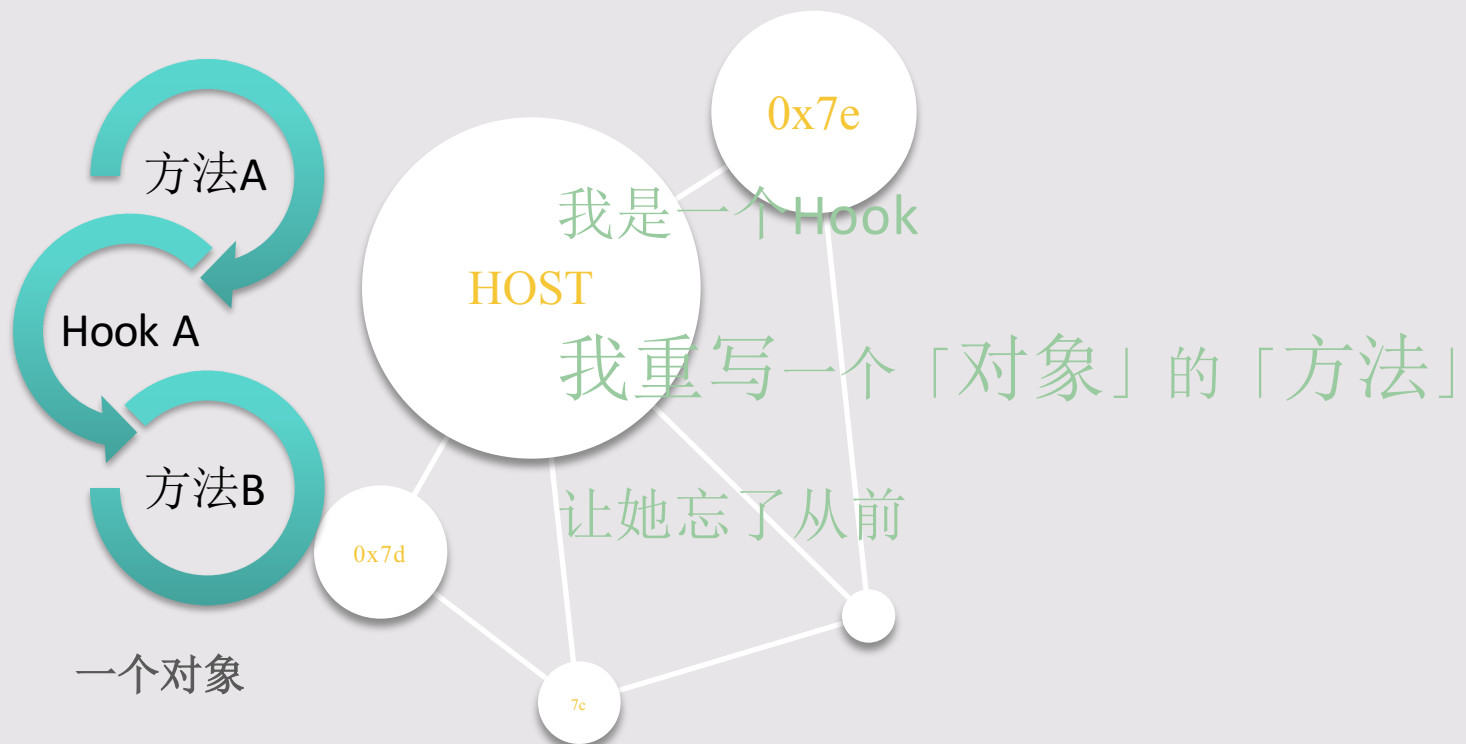
```
host (0x7f)
|-- mipmap (02)
|   |-- ic_launcher (0000)
|
|-- values
|   |-- strings.xml (03)
|       |-- app_name (0000)
|
|-- syles.xml (04)
    |-- AppTheme (0000)
```

```
plugin ( 0x7e)
|-- mipmap (02)
|   |
|   |-- ic_plugin ( 0000)
|-- values
|   |-- strings.xml (03)
|       |
|       |-- s_plugin ( 0000)
|-- syles.xml (04)
    |
    |-- PluginTheme ( 0000)
```

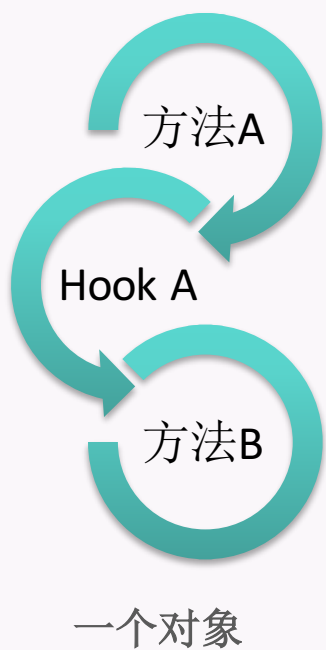
轻/轻盈产出



轻/轻度Hook



轻/轻度Hook



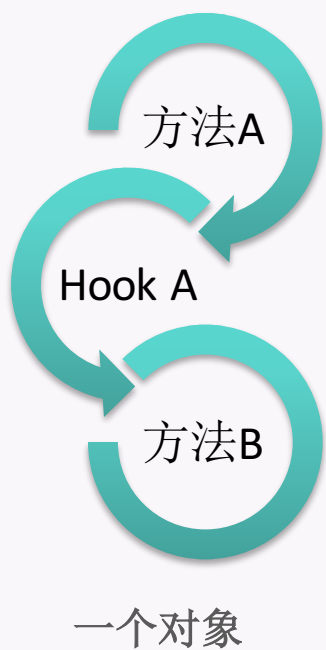
我是一个Hook

我是一个「对象」的「方法」

她静静地坐着

她的方法「重放」

轻/轻度Hook



但是首先

我要找到她

在我的「进程」里

启动

插件Activity

坐着

她的方法向我「开放」

轻/轻度Hook

我的「进程」
com.user.galen

启动
插件Activity

系统进程
system.process

Intent解析
任务栈调度
Activity栈调度

Not Found

我的「进程」
com.user.galen

实际启动
插件Activity

轻/轻度Hook

我的「进程」
com.user.galen

系统进程
system.process

我的「进程」
com.user.galen

捐

:

Not Found

Intent解析
任务栈调度
Activity栈调度

实际启动
插件Activity

轻/轻度Hook

我的「进程」
com.user.galen

系统进程
system.process

我的「进程」
com.user.galen

启动
插件Activity

Hook
伪装宿主

Intent解析
任务栈调度
Activity栈调度

启动
Activity

轻/轻度Hook

Small

Droid Plugin

Android-Plugin-
Framework
ACDD

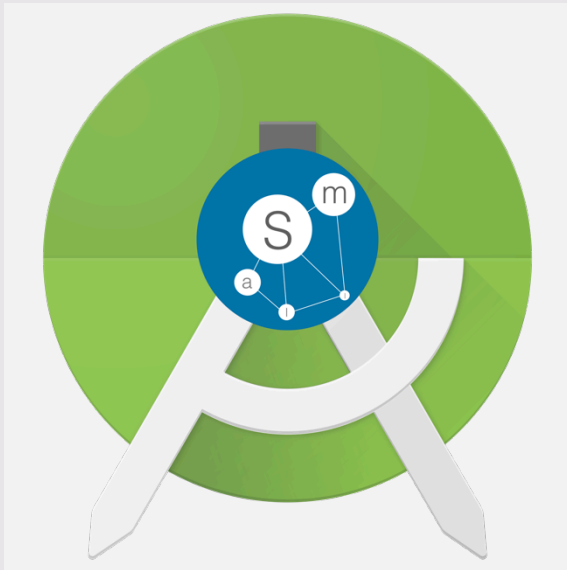
Dynamic-Load-APK
Direct-Load-APK

```
private Activity performLaunchActivity(ActivityClientRecord r, Intent customIntent) {  
    // 创建Activity  
    if (r.activityInfo.targetActivity != null) {  
        component = new ComponentName(r.activityInfo.packageName,  
            r.activityInfo.targetActivity);  
    }  
    java.lang.ClassLoader cl = r.packageInfo.getClassLoader();  
    activity = mInstrumentation.newActivity(  
        cl, component.getClassName(), r.intent);  
  
    // 绑定Context  
    Context appContext = createBaseContextForActivity(r, activity);  
    activity.attach(appContext, this, getInstrumentation(), ...);  
    // 设置主题  
    int theme = r.activityInfo.getThemeResource();  
    if (theme != 0) {  
        activity.setTheme(theme);  
    }  
  
    // 触发onCreate  
    mInstrumentation.callActivityOnCreate(activity, r.state);  
}
```

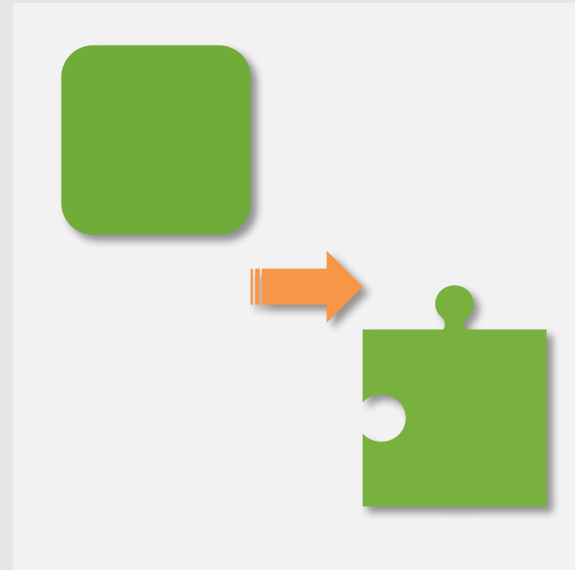
实际启动
插件Activity

轻/轻度Hook

插件方案	代表框架	包数	类加载器 个数	资源管理器 个数	Context
完全隔离	Droid Plugin	1/插件	1/插件	1/插件	1/插件
	Dynamic-Load-APK	1	1/插件	1/插件	1/插件Activity
	Direct-Load-APK	1	1/插件	1/插件	1/插件Activity
宿主插件 两两融合	Android-Plugin- Framework	1	1/插件	1/插件	1/插件Activity
除主题外 完全融合	ACDD	1	1	1	1/插件Activity
完全融合	Small	1	1	1	1



IDE友好
@Debug

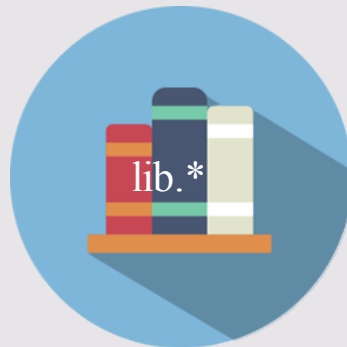
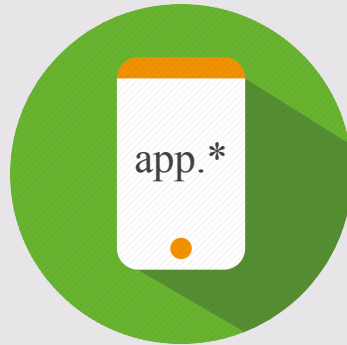


模块变身
@Release

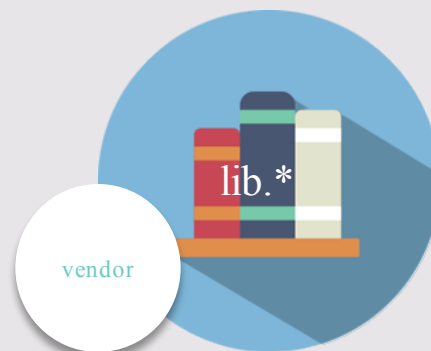
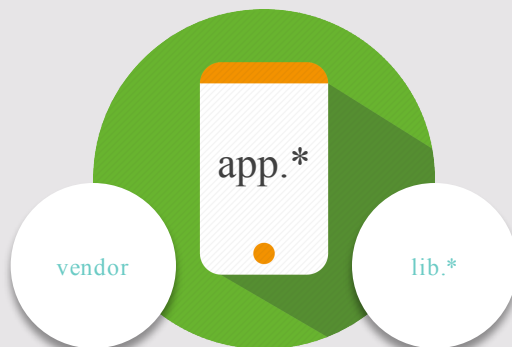
巧/IDE友好



巧/IDE友好



巧/IDE友好



巧/模块变身

模块是开发态，插件是目标态。

模块	开发态	目标态	转换难点
app.*	Application模块 可以依赖其他模块、可以独立运行	带代码、资源的插件	AAR(代码/资源) 剥离
lib.*	Library模块 可以被app.*依赖	带代码、资源的插件	资源ID锁定
[other].*	Application模块 可以独立运行	仅含assets的插件	

巧/模块变身

模块是开发态，插件是目标态。

模块	开发态	目标态	转换难点
app.*	Application模块 可以依赖其他模块、可以独立运行	带代码、资源的插件	<div>AAR</div> (代码/资源)剥离
lib.*	Library模块 可以被app.*依赖	带代码、资源的插件	资源ID锁定
[other].*	Application模块 可以独立运行	仅含assets的插件	

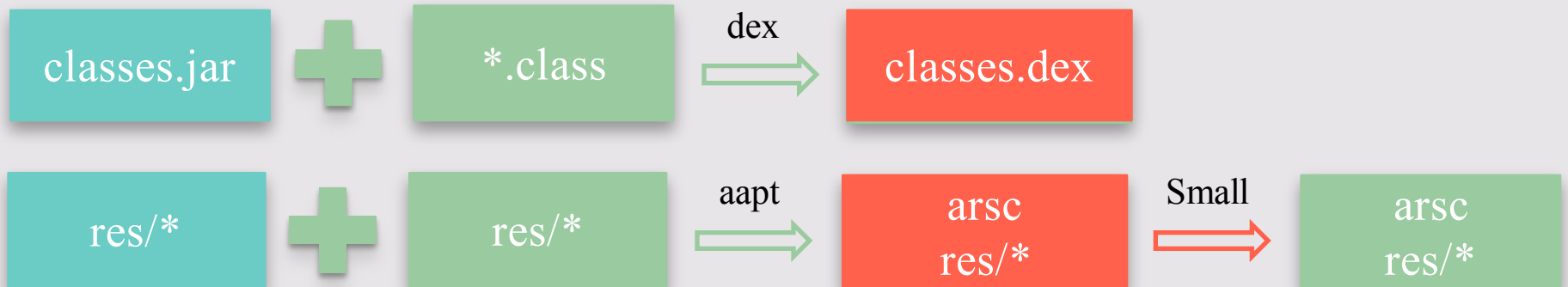
巧/模块变身

AAR

```
dependencies {  
    provided fileTree(dir: 'libs', include: ['*.jar'])  
    compile 'com.android.support:appcompat-v7:23.2.1'  
    compile 'com.android.support:design:23.2.1'  
}
```



app.main
 |-- build/intermediates/exploded-aar
 |-- com.android.support/appcompat-v7/23.2.1



总模块变身





Q & A