

## 1 多包

manifest中 meta - data + 占位符，一行代码

gradle中 + productFlavors渠道号，可直接复制

程序中 + LZAnalysis.java，app入口调用

```
LZAnalysis.activate(this, "user1");
```

gradle一键签名 + 打包，时间 = 单包时间 \* 包数。每次更新重打一次，数据收集、分析、关键词更改等均不用修改。

优点：客户程序员操作简单

缺点：客户服务器需要存储多包

## 2 动态打包

程序中 + LZDynamic.java，app入口调用

```
LZDynamic.activate(this, "user1");
```

下载页面中调用python程序，在原包基础上动态打包，原包可是经过签名和混淆的包，对原包保证无任何影响，打包时间几毫秒

```
import zipfile
zipped = zipfile.ZipFile(your_apk, 'a', zipfile.ZIP_DEFLATED)
empty_channel_file = "META-INF/
lzchannel_{channel}".format(channel=your_channel)
zipped.write(your_empty_file, empty_channel_file)
```

有两种细分方案：

a. 客户将动态打包程序部署在客户的服务器上。

优点：操作更简单，速度快，省存储

缺点：对客户程序猿要求会部署，但依旧较简单大部分程序猿一学就会，我们也可远程登录直接帮用户部署。

b. 客户将原包交给我们，小白下载时从客户页面跳转到我们的服务器上下载。

优点：客户极度省事儿

缺点：流量大，会产生成本