

# Android底座、组件打包

## 传统打包方式的痛点

1. 版本号需要手动维护，如果需要发布正式版本，需要人工在原来的版本号上进行修改，这种方式在组件数量较少的情况下没什么问题，但是当组件数量多，维护将会耗时耗力，而且容易出错。
2. 如果一个组件A需要发布，那么需要去寻找，是否有其他组件依赖了A，比如找到了B、C依赖A，那么BC也需要进行发布。BC发布完成，需要去寻找是否有组件依赖了BC....
3. 发布的过程需要人手动去执行，步骤繁琐。

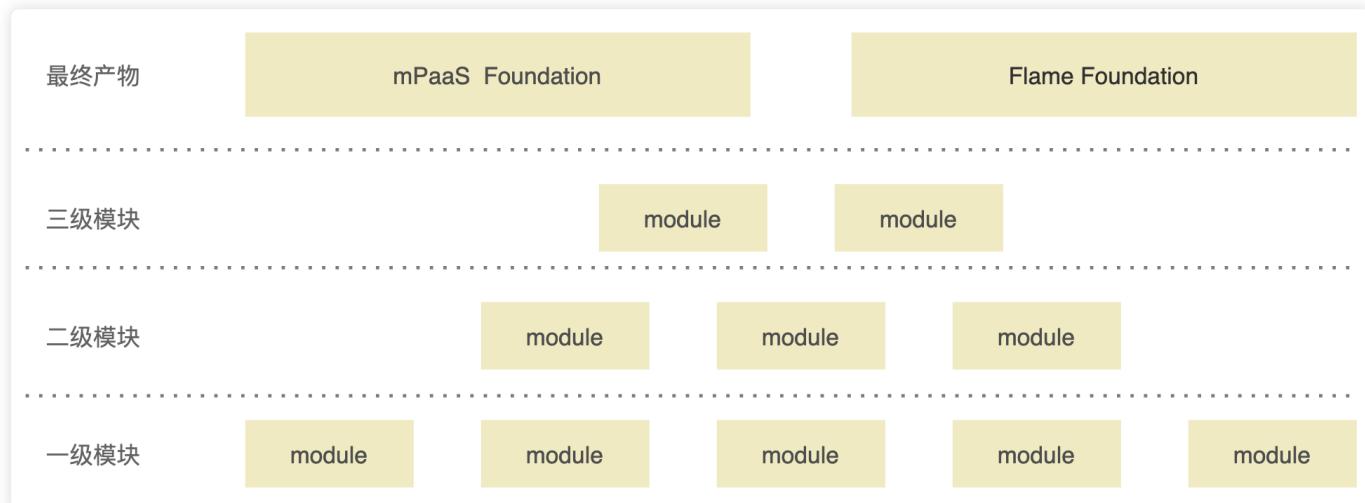
## 优点

1. 无需手动维护组件的版本号，无需关心组件上传的过程，由脚本控制，不会出错。
2. 组件的正式版本，通过Jenkins在云端打包，打包之前通过git同步代码，因此可以保证最终产物的一致性。
3. 开发者不介入打包的流程，需要做的是保证自己代码无误，然后推送到远程的master分支即可。Jenkins接收到master分支的push event后，自动执行打包任务。
4. 打包的开始、成功、失败等事件，通过Jenkins插件发消息到钉钉群。

## 构建环境

1. 支持Mac OS、Windows、Linux
2. JDK、Android SDK
3. 支持Jenkins远程构建

## 底座、组件框架图



# 最终产物

## 本地仓库

本地maven仓库以文件夹的形式存在，作为开发、测试阶段使用，不限制上传频率，对远程的maven仓库没有影响

## 远程仓库

对应正式环境，上传到远程maven仓库的代码，必须经过测试，确认无误才可上传。

## 发布流程图

