**动态加载Apk-插件化技术(动态代理方案)**

1. 什么是插件化
2. 主App(宿主App)加载插件apk的实现
3. 每个业务组件模块形成一个独立的Apk, 然后通过主App动态加载部署业务组件模块Apk的一种方案
4. 插件化的优点好处
5. 业务组件解耦,能够实现业务组件模块的热插拔
6. 更改产品迭代模式,可分为主App和次Apk(动态加载业务组件模块)
7. 改善产品更新过程,可以在不影响用户的情况下实现业务组件模块更新以及重要Bug修复
8. 减轻主App的内存和CPU占用,提高应用的性能.
9. 插件化的思想

动态加载Apk的主要思想是:主App是被系统(PMS)安装,被系统(AMS)调用,整个过程都是由系统提供的,而插件Apk并非一个真正的Apk,只是一个打包成Apk的一个组件模块,因为它并非被系统安装调用.简言之,需要讲插件Apk看成一个”非Apk”文件,只是一个结构比较复杂的压缩打包成Apk格式的文件.调用插件即用某种特殊技术手段打开文件并执行其相关代码.

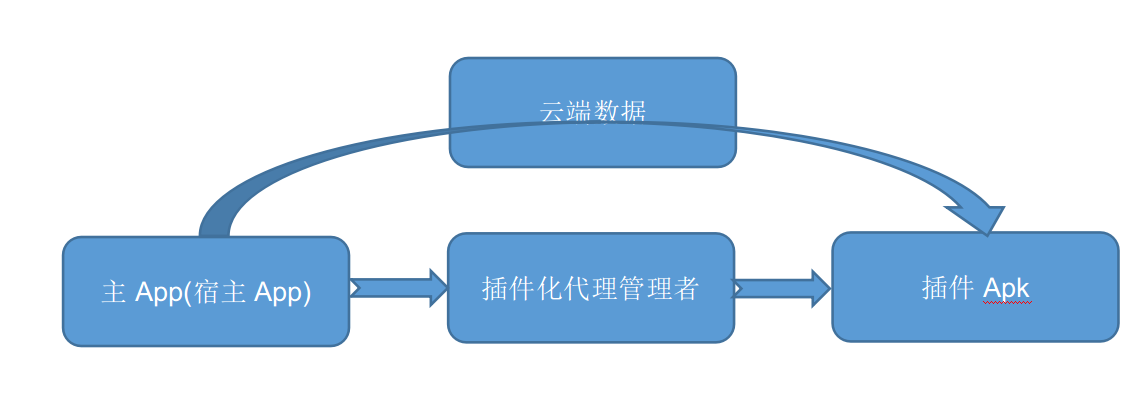
1. 插件化的步骤-分析主App

1.主APp打包完成解压后,会有dex,images,xml,asset等类型文件

2.Dex靠PathClassLoader加载运行

3.图片以及xml等资源依靠Resources&AssetManager加载管理

1. 插件化的实现流程



1. 插件化的代码实现步骤

1.创建DexClassLoader加载插件化Apk相关代码

2.创建Resources&AssetManager来加载插件化Apk的资源

3.管理插件Apk里的组件(如Activity)的生命周期

4.通过代理模式实现对插件Apk里面组件的管理