移动虚拟化：360分身大师那些事

团队：手机卫士
讲师：王云鹏

课程提纲/内容

分身大师及技术架构

基本原理解析

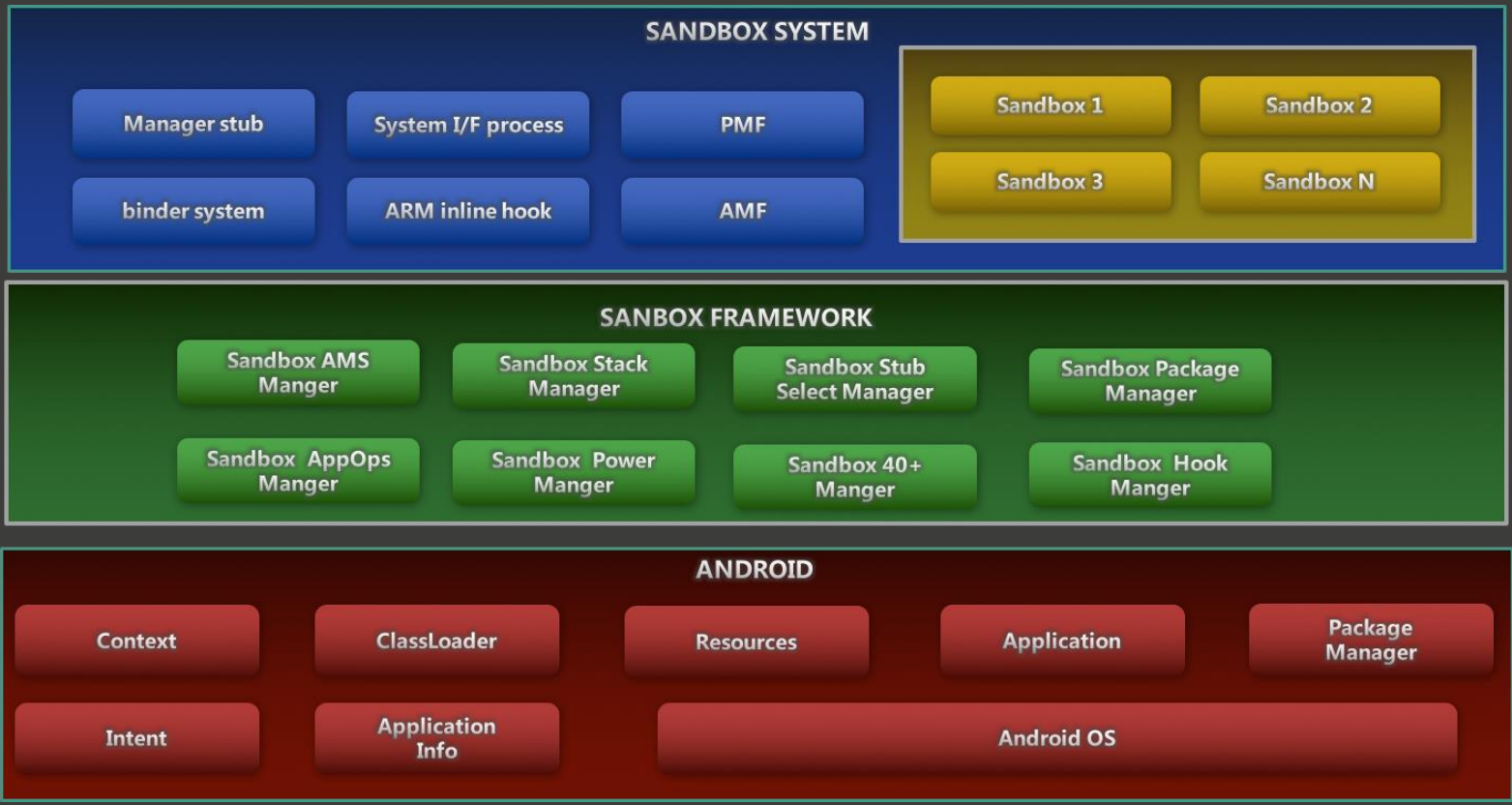
分身大师实战经验

分身大师Xposed方案

分身大师

- 基于沙箱机制打造的Android App
- 内部运行原生Android应用
- 依赖Android的Hook机制
- 轻量级的Android虚拟机

整体技术架构


SANDBOX SYSTEM
Manager stub
System I/F process
PMF
binder system
ARM inline hook
AMF
Sandbox 1
Sandbox 2
Sandbox 3
Sandbox N
SANBOX FRAMEWORK
Sandbox AMS Manger
Sandbox Stack Manager
Sandbox Stub Select Manager
Sandbox Package Manager
Sandbox AppOps Manger
Sandbox Power Manger
Sandbox 40+ Manger
Sandbox Hook Manger
ANDROID
Context
ClassLoader
Resources
Application
Package Manager
Intent
Application Info
Android OS

课程提纲/内容

分身大师及技术架构

基本原理解析

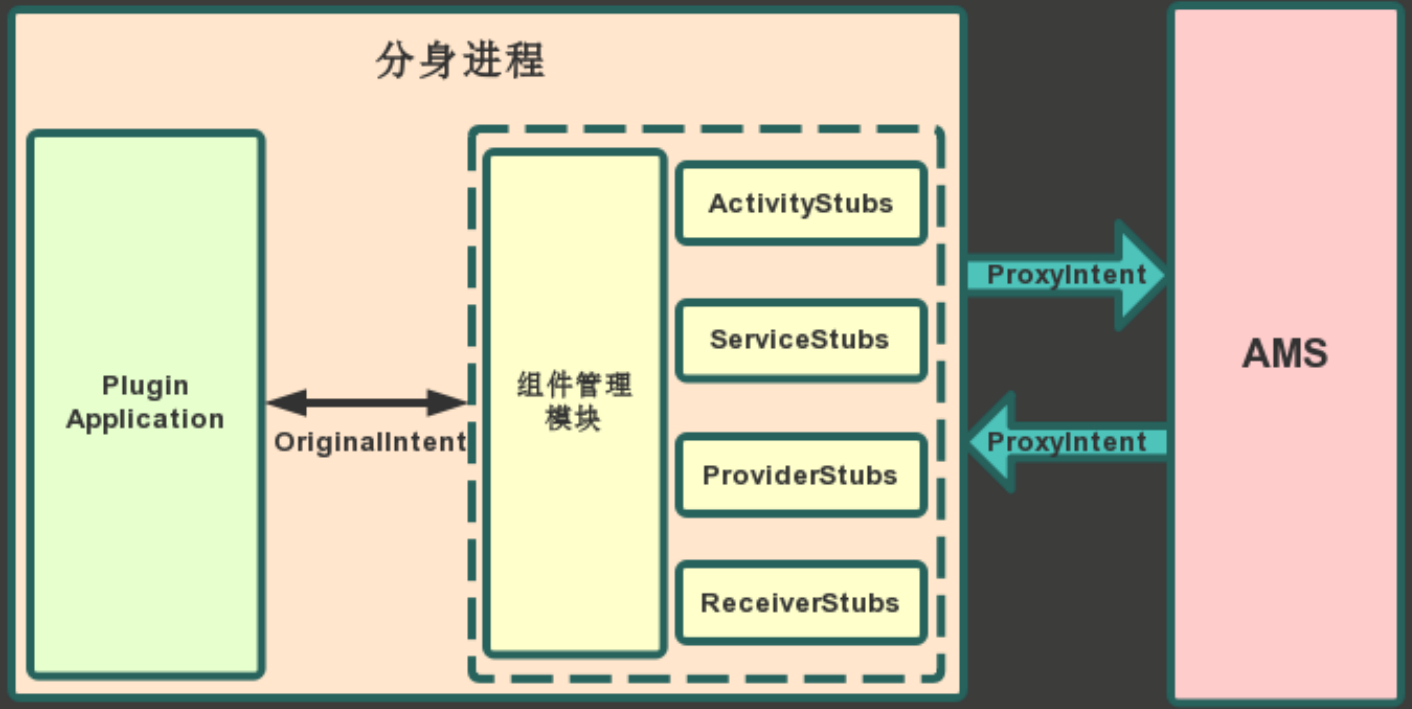
分身大师实战经验

分身大师Xposed方案

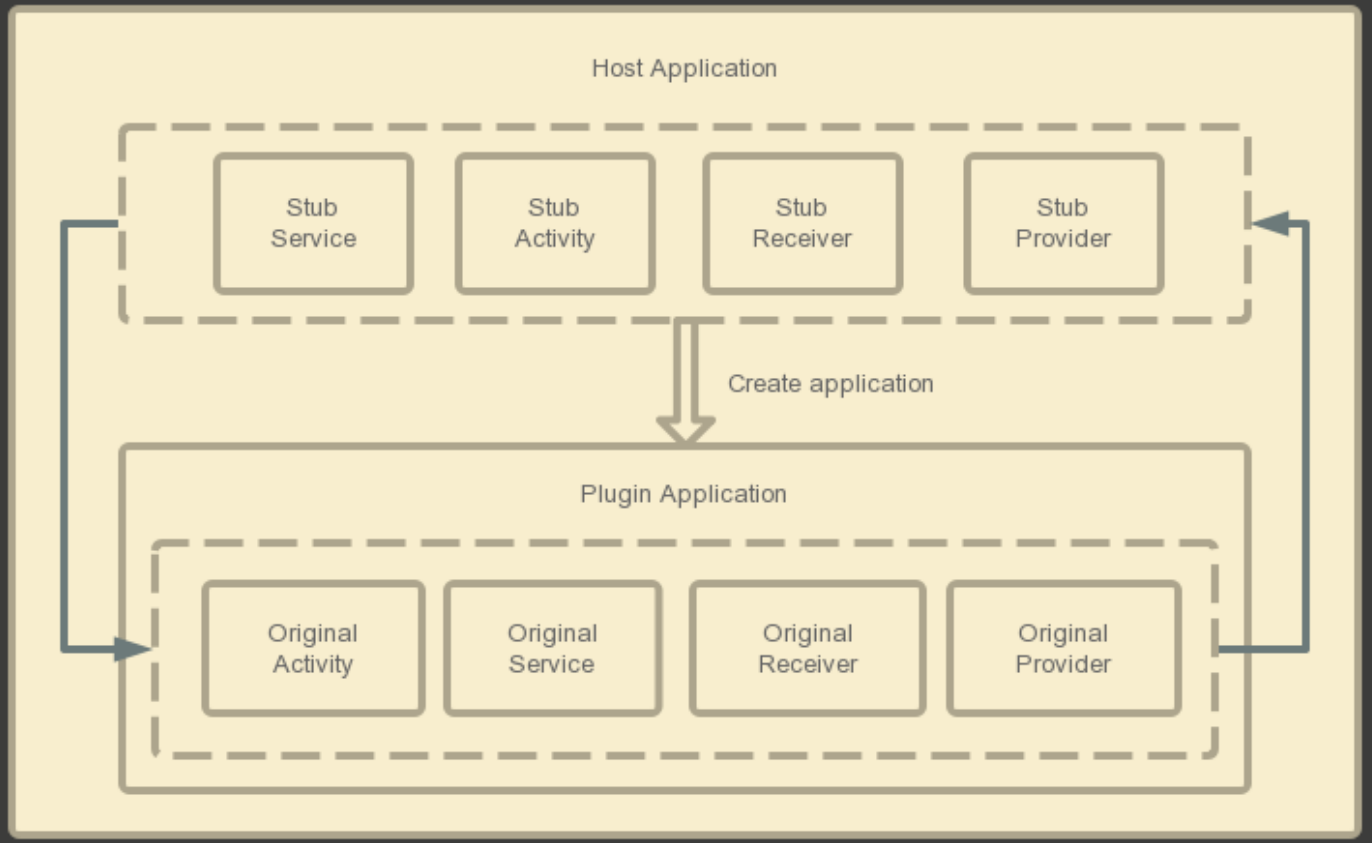
需要解决的问题

- Android 4大组件代理机制
- 初始化Application
- 和系统服务通信(Binder Hook)
- 文件路径重定向(Native Hook)
- 运行Android 4大组件

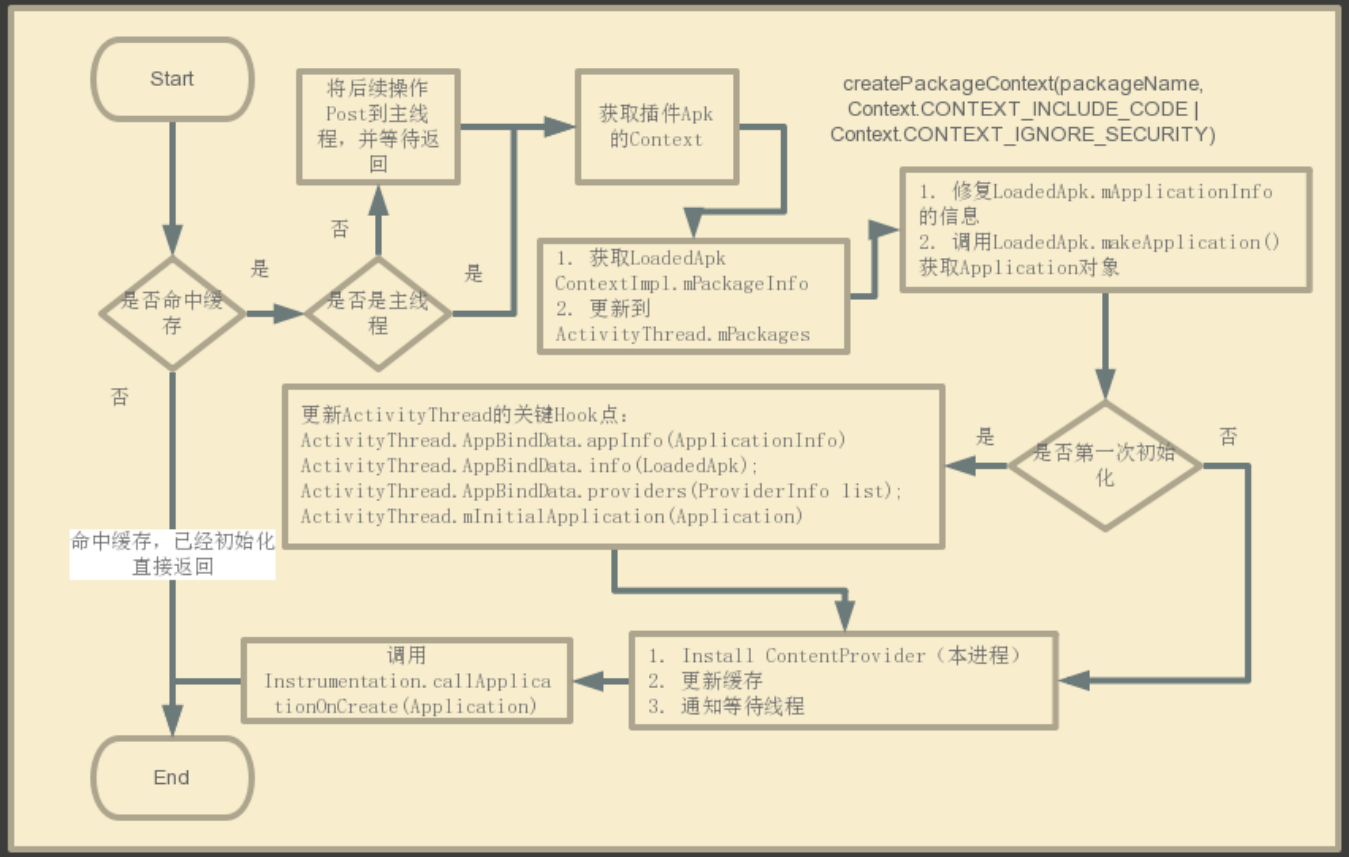
组件代理机制


分身进程
Plugin Application
OriginalIntent
组件管理模块
ActivityStubs
ServiceStubs
ProviderStubs
ReceiverStubs
ProxyIntent
ProxyIntent
AMS

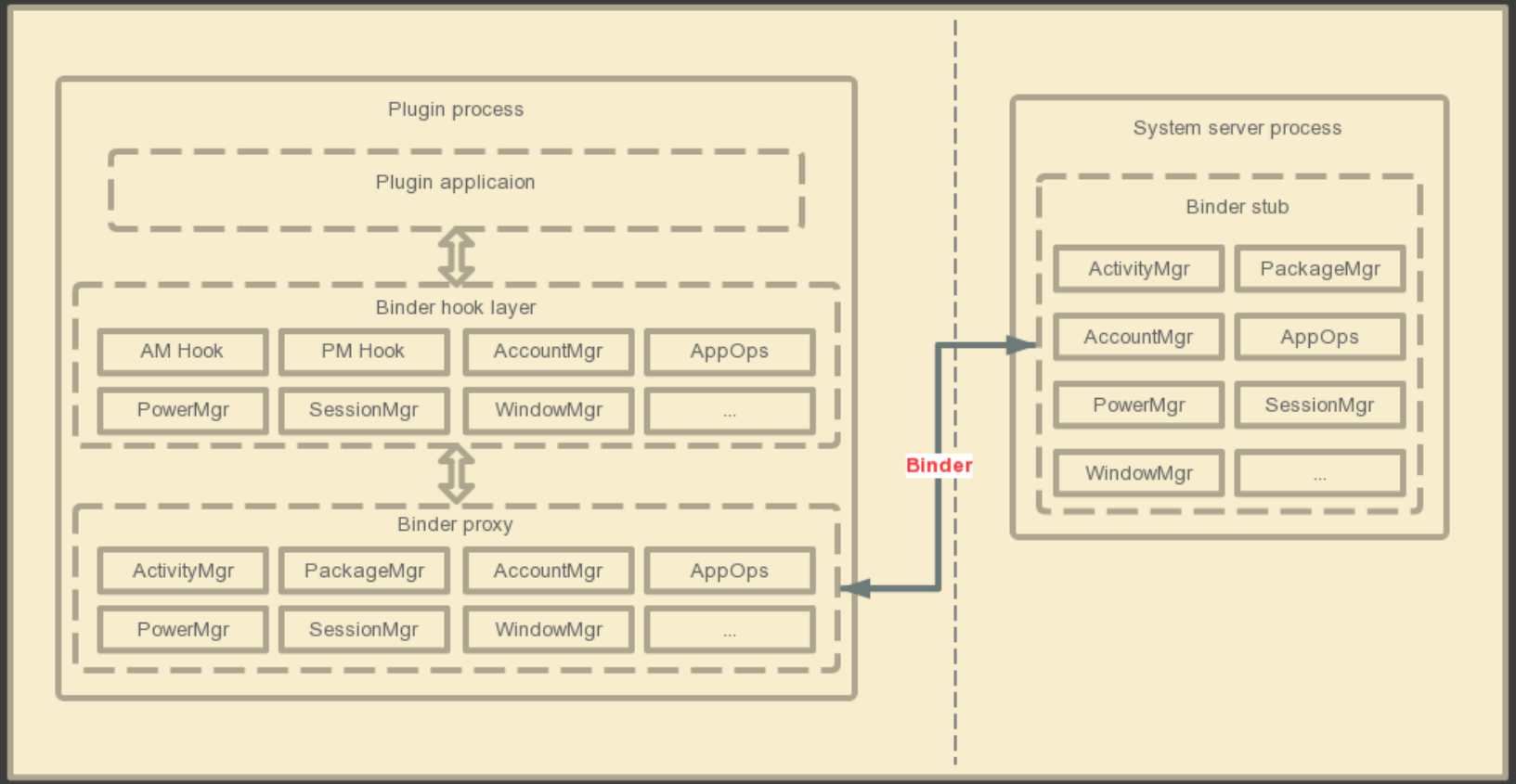
Application初始化



Application初始化流程

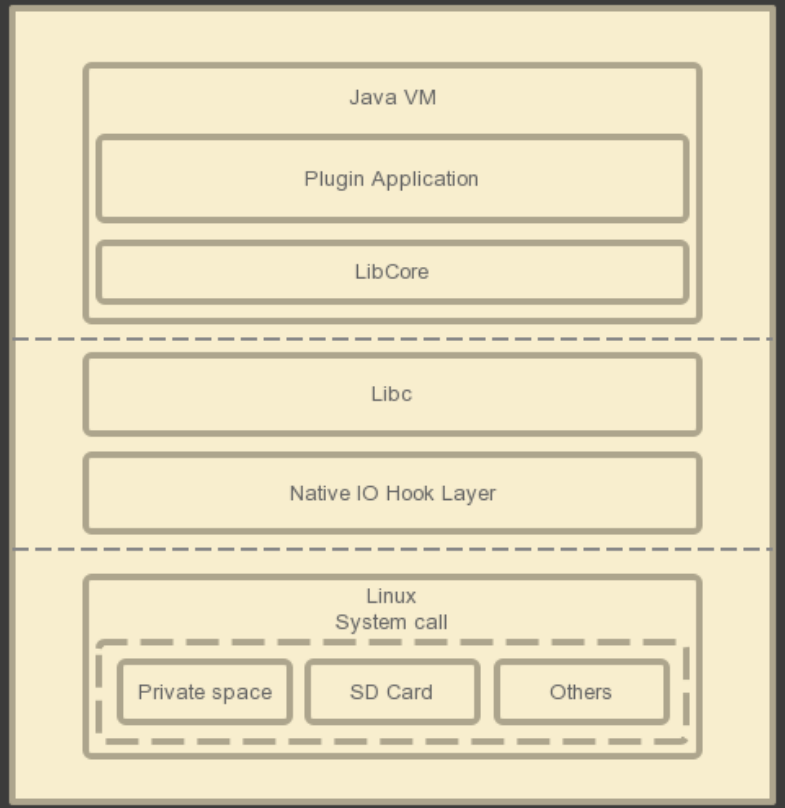


和系统服务通信

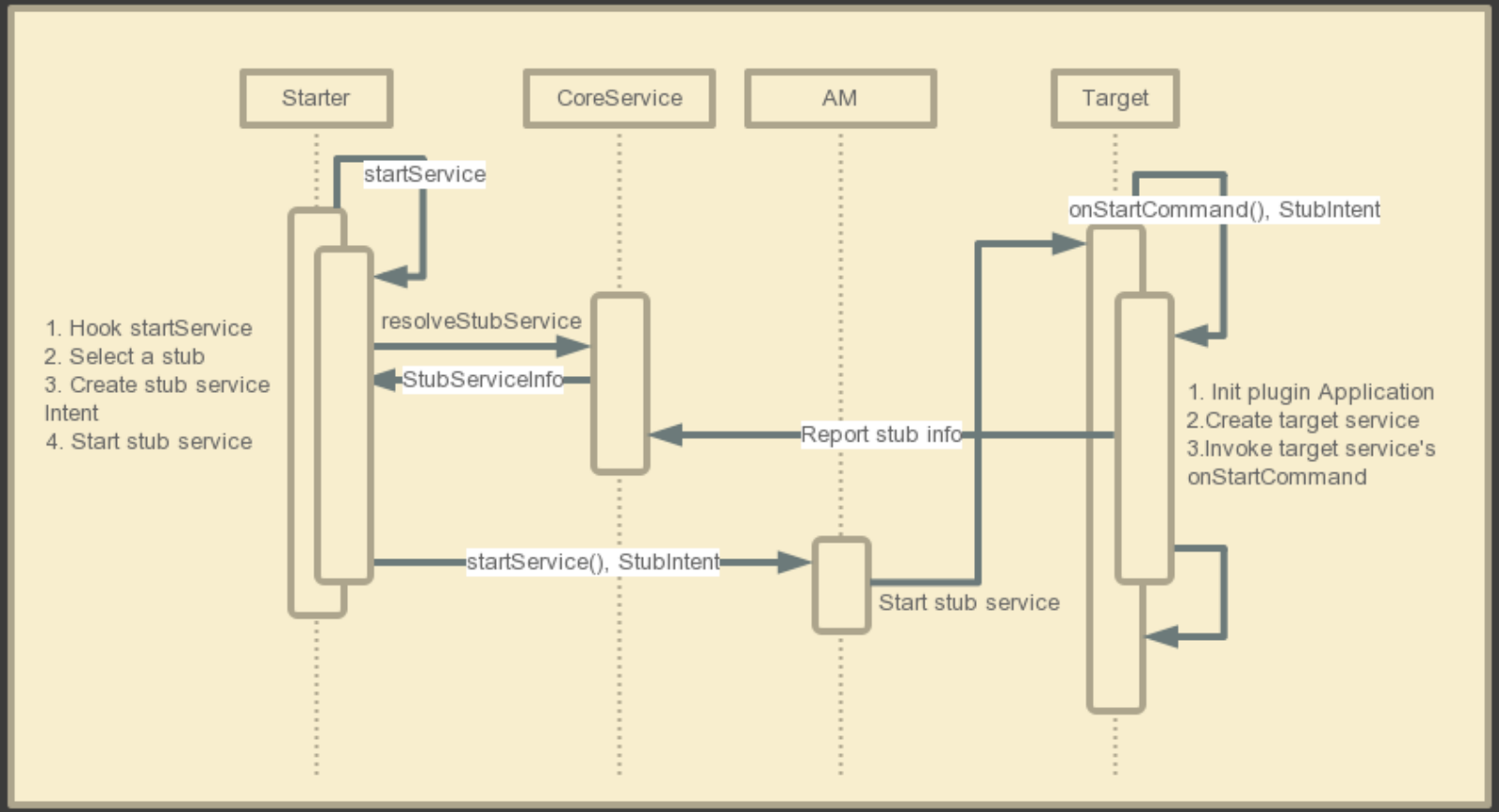

Plugin process
Plugin applicaion
Binder hook layer
AM Hook
PM Hook
AccountMgr
AppOps
PowerMgr
SessionMgr
WindowMgr
...
Binder proxy
ActivityMgr
PackageMgr
AccountMgr
AppOps
PowerMgr
SessionMgr
WindowMgr
...
Binder
System server process
Binder stub
ActivityMgr
PackageMgr
AccountMgr
AppOps
PowerMgr
SessionMgr
WindowMgr
...

文件路径重定向

- 通过Native IOHook实现运行时替换
- SD卡目录隔离
- 与外部应用通信时，路径的正向和反向替换
- 加固类应用DEX目录重定向处理



组件启动流程


Starter
CoreService
AM
Target
startService
onStartCommand(), StubIntent
1. Hook startService
2. Select a stub
3. Create stub service
Intent
4. Start stub service
resolveStubService
StubServiceInfo
Report stub info
1. Init plugin Application
2.Create target service
3.Invoke target service's
onStartCommand
startService(), StubIntent
Start stub service

课程提纲/内容

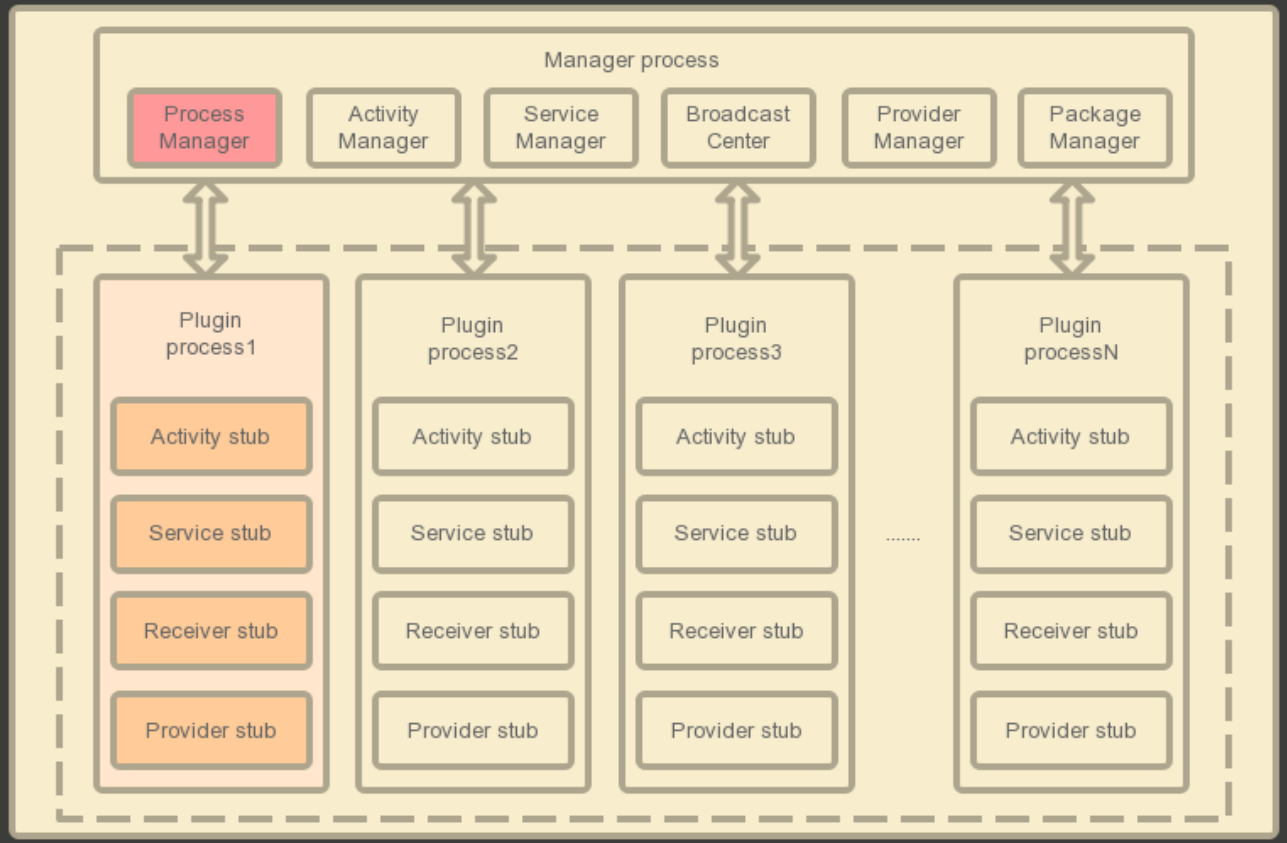
分身大师及技术架构

基本原理解析

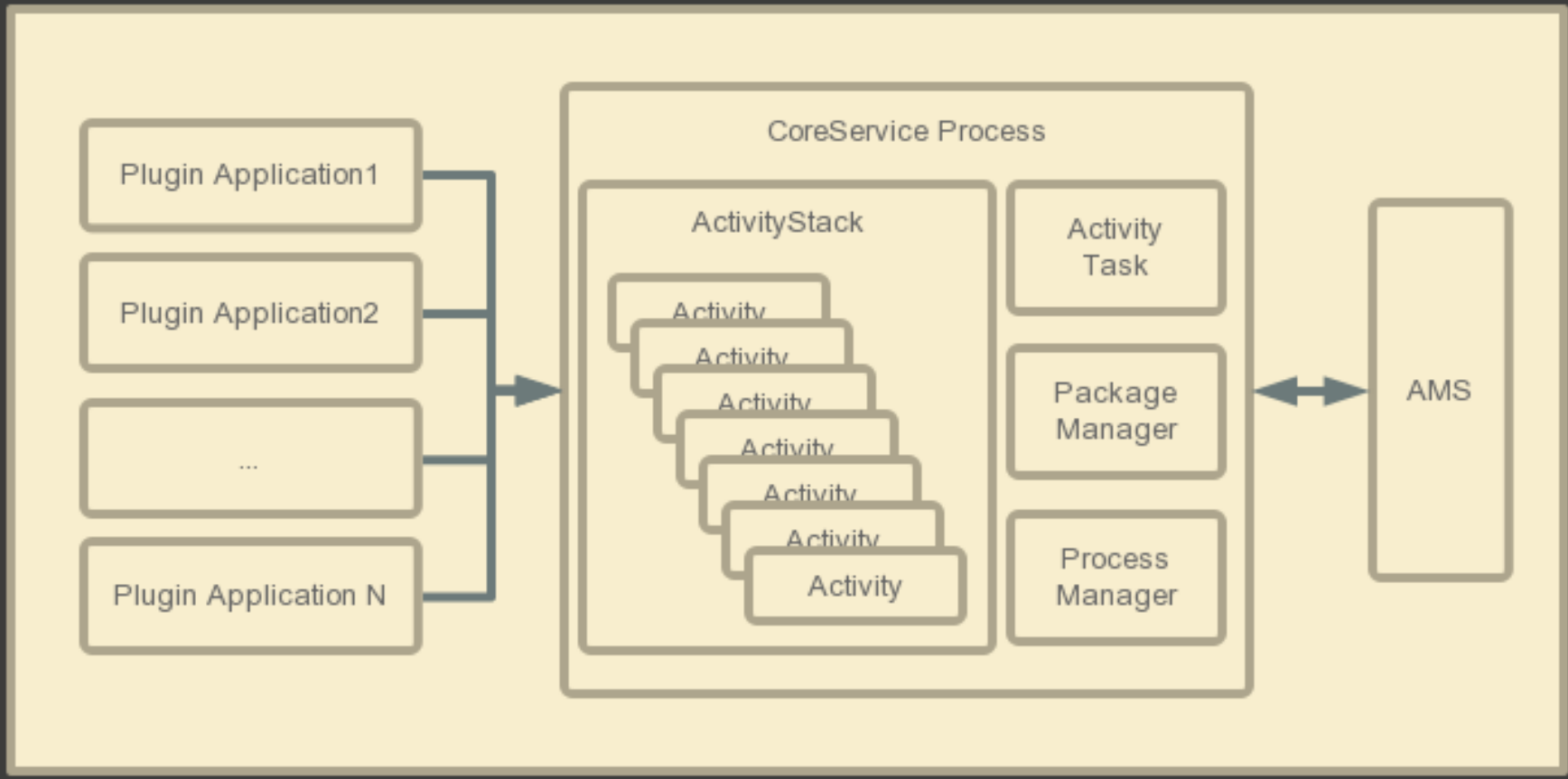
分身大师实战经验

分身大师Xposed方案

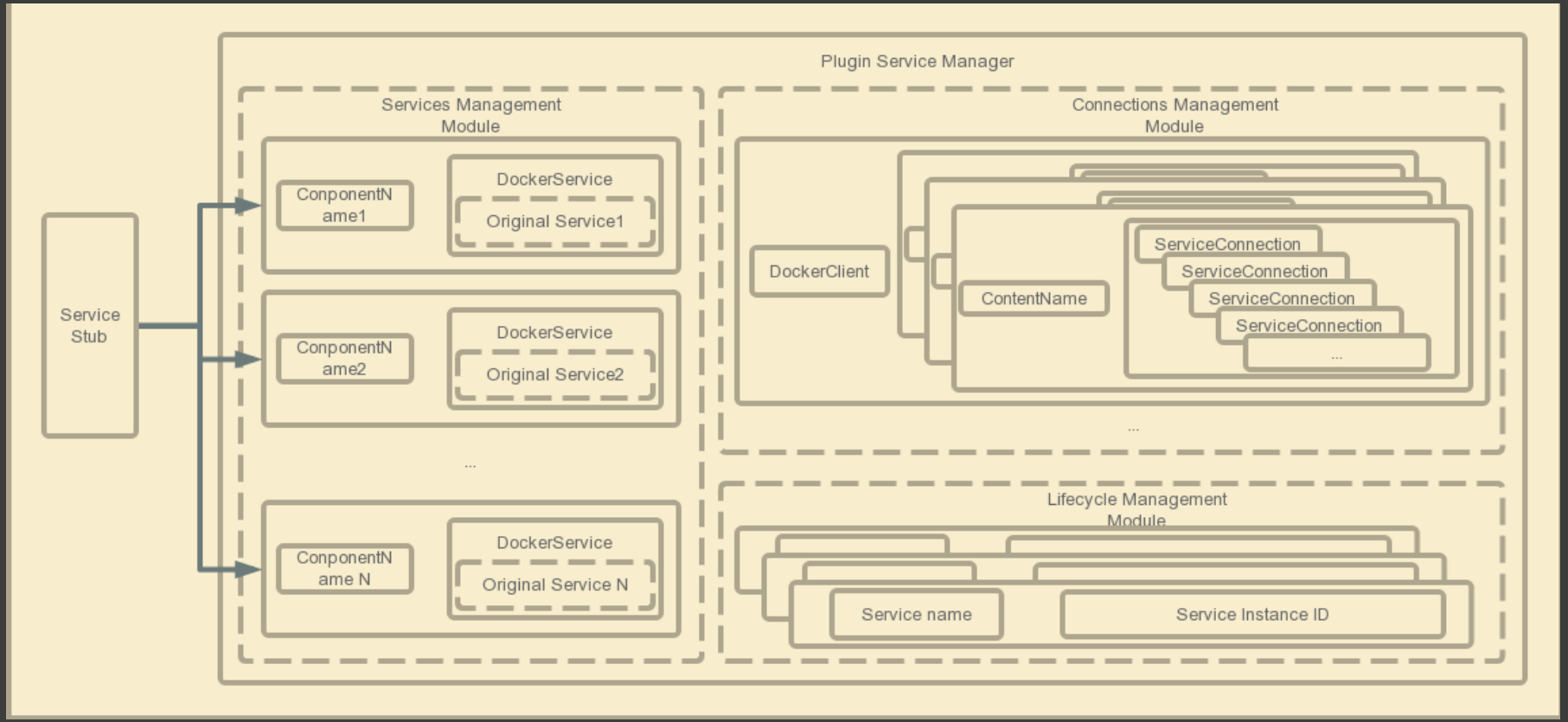
组件管理


Manager process
Process Manager
Activity Manager
Service Manager
Broadcast Center
Provider Manager
Package Manager
Plugin process1
Plugin process2
Plugin process3
Plugin processN
Activity stub
Service stub
Receiver stub
Provider stub
.....

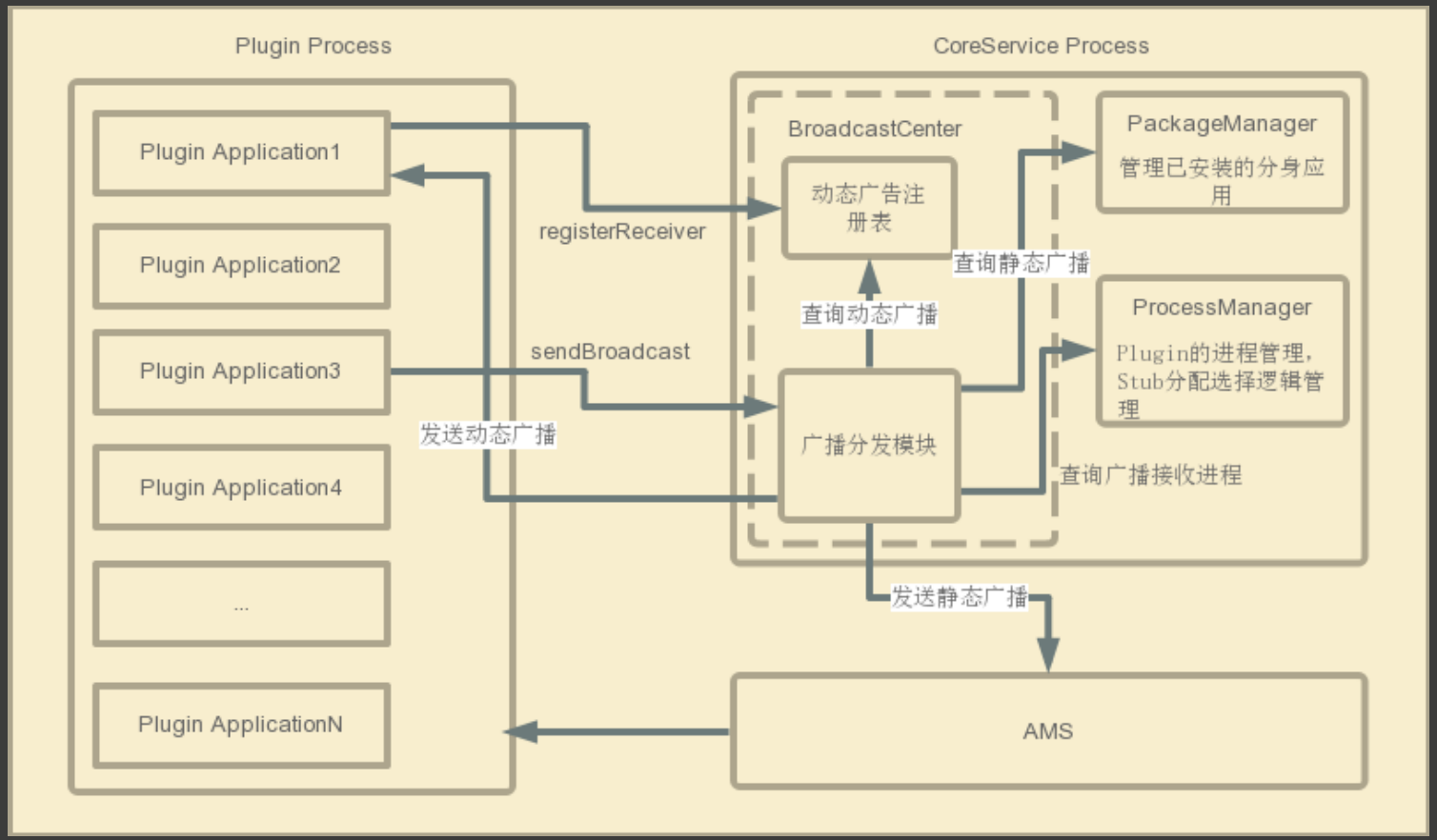
Activity技术方案



Service技术方案



Receiver技术方案



技术挑战

- 需要Hook的点多，适配量巨大
- Android版本不断迭代，权限收紧
- Apk千差万别
- 加固应用方案变更

课程提纲/内容

分身大师及技术架构

基本原理解析

分身大师实战经验

分身大师Xposed方案

技术可行性

- 为什么要支持Xposed
- 如何替代ROOT，提供类似环境
- 支持修改系统的Xposed插件？
- 支持修改3rd App的Xposed插件
- 注入Xposed插件代码到宿主应用进程
- 宿主应用进程中如何提供Xposed插件运行环境

Xposed介绍

- Xposed

  Xposed框架的C++部分（Xposed版的zygote）

- XposedBridge

  Xposed框架的Java部分

- android_art

  定制版的Android Art库（用于Art Hook实现）

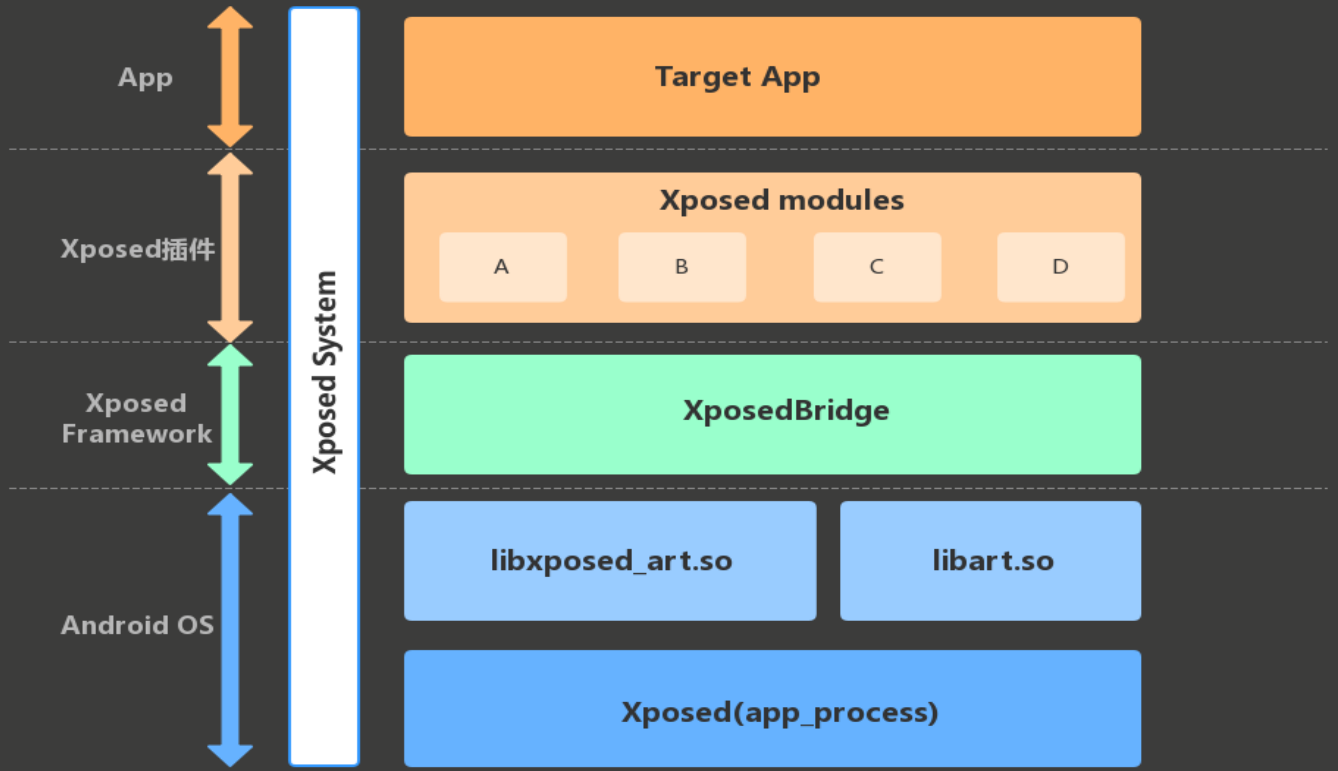
- XposedTools

  辅助编译Xposed和XposedBridge

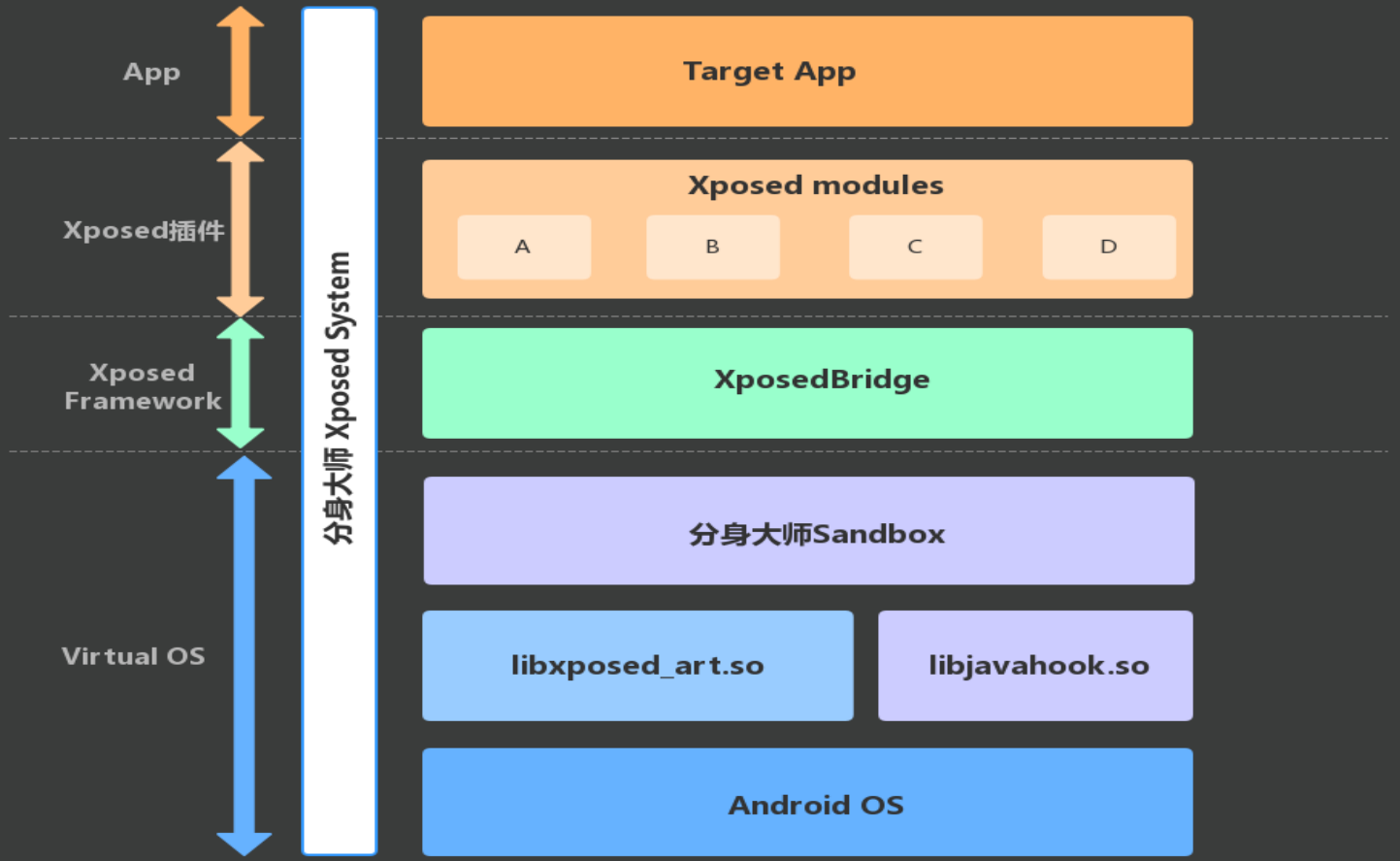
- XposedInstaller

  Xposed的插件管理和功能控制App

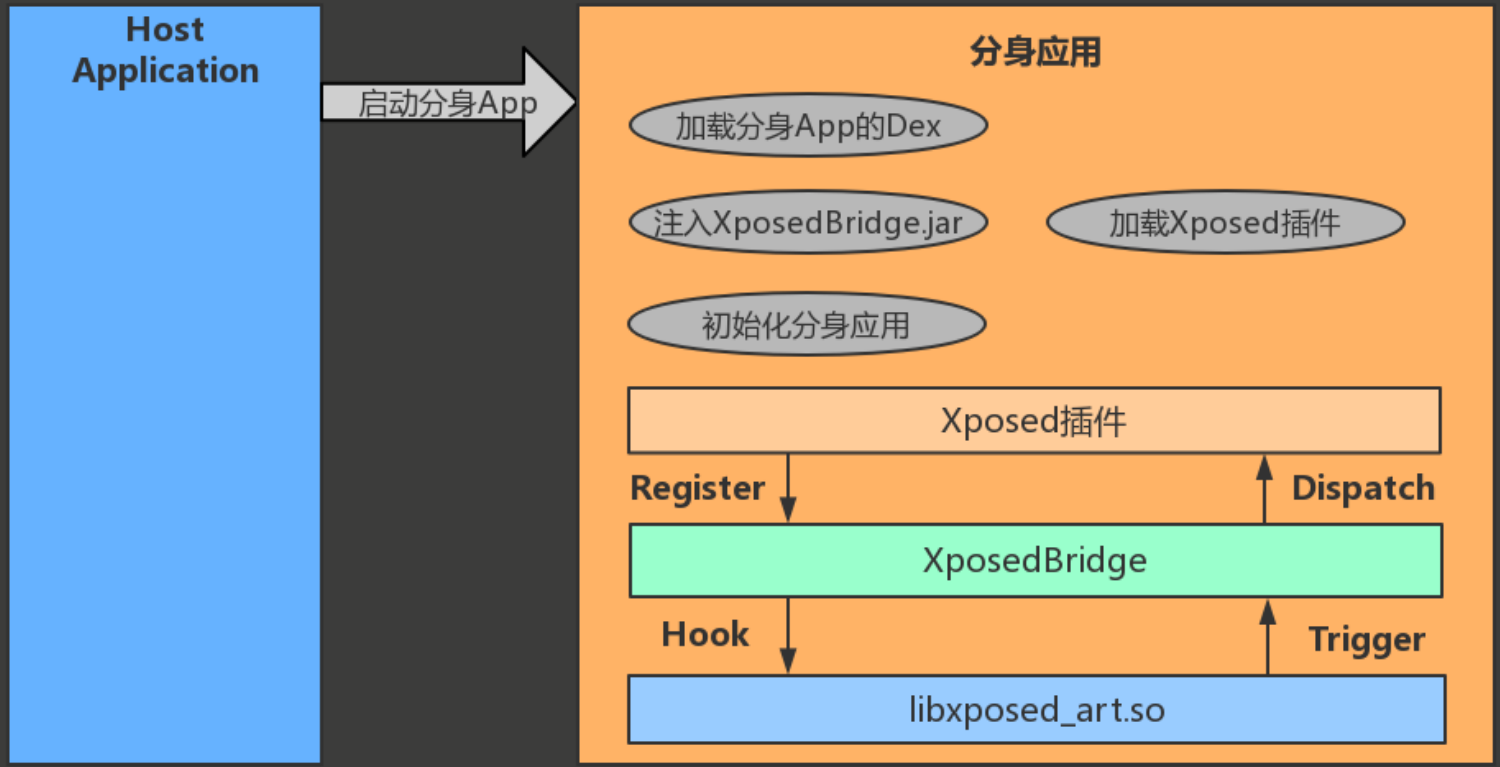
Xposed框架


App
Xposed插件
Xposed Framework
Android OS
Xposed System
Target App
Xposed modules
A
B
C
D
XposedBridge
libxposed_art.so
libart.so
Xposed(app_process)

分身大师Xposed框架



分身大师Xposed框架



分身大师Xposed框架优势

- 免ROOT运行Xposed插件
- 无缝接入原生Xposed插件
- 专业技术团队支持
- Xposed合作站点
  - http://xposed.appkg.com
- 官方开发者网站(敬请期待)
  - wangyunpeng@360.cn

技术交流（干货）：奇卓社（360移动技术微信公众号）

谢 谢！