

# 手机百度iOS技术平台化方案

王聪



2015年6月

# 目录 CONTENTS

**1.** 背景介绍

**2.** 问题分析

**3.** 解决方案

**4.** 目前效果

**5.** 后续扩展

# 背景

连接人与服务

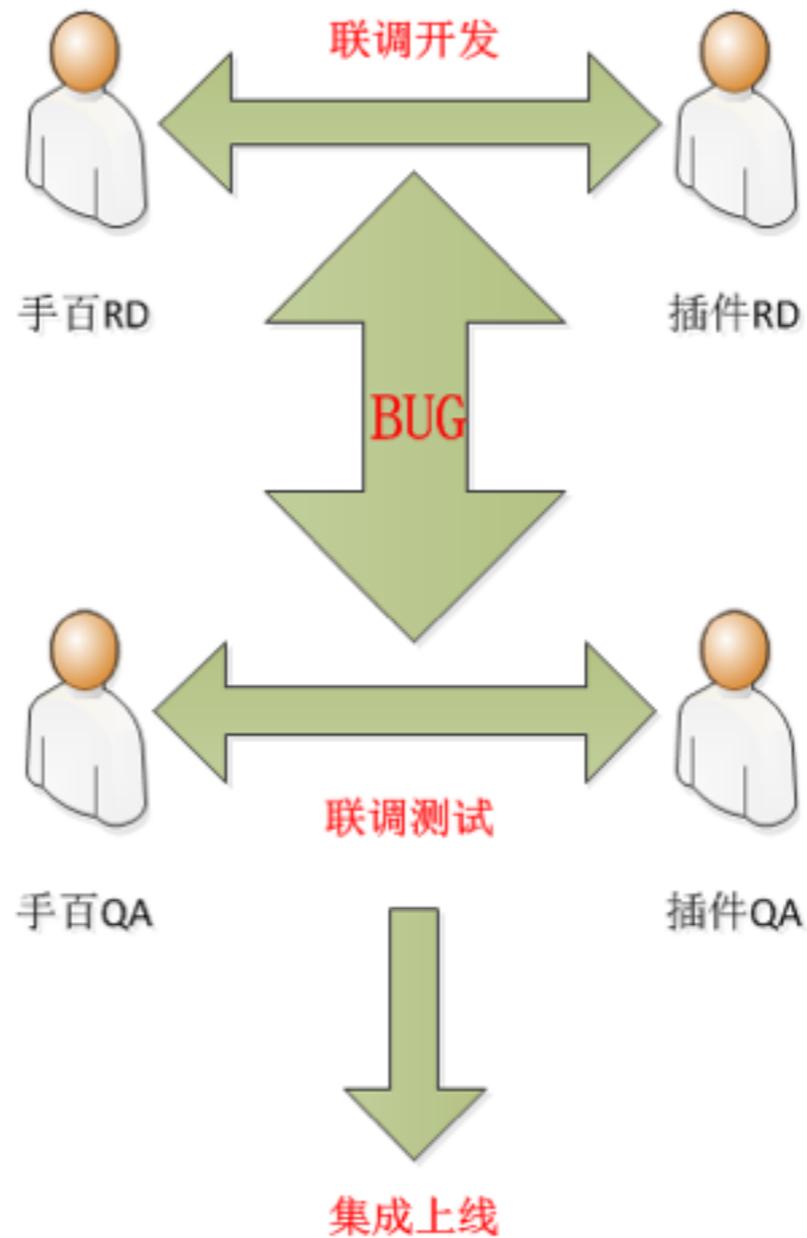
业务井喷式增长

成型业务以插件形式的接入



# 背景

## 传统插件集成方式



## 成本来源

- RD间联调开发
- QA间联调测试
- Bug: 定位Bug原因以及随之带来的频繁打包



如何降低成本，  
提高效率？

# 平台化转变

接口平台化

开发平台

准入平台

# 方案



手机百度平台

# 方案



# 方案

## 壳工程

- 工程配置
- 资源文件
- 手百主体代码打包后的库文件
- 可直接运行
- 安全机制

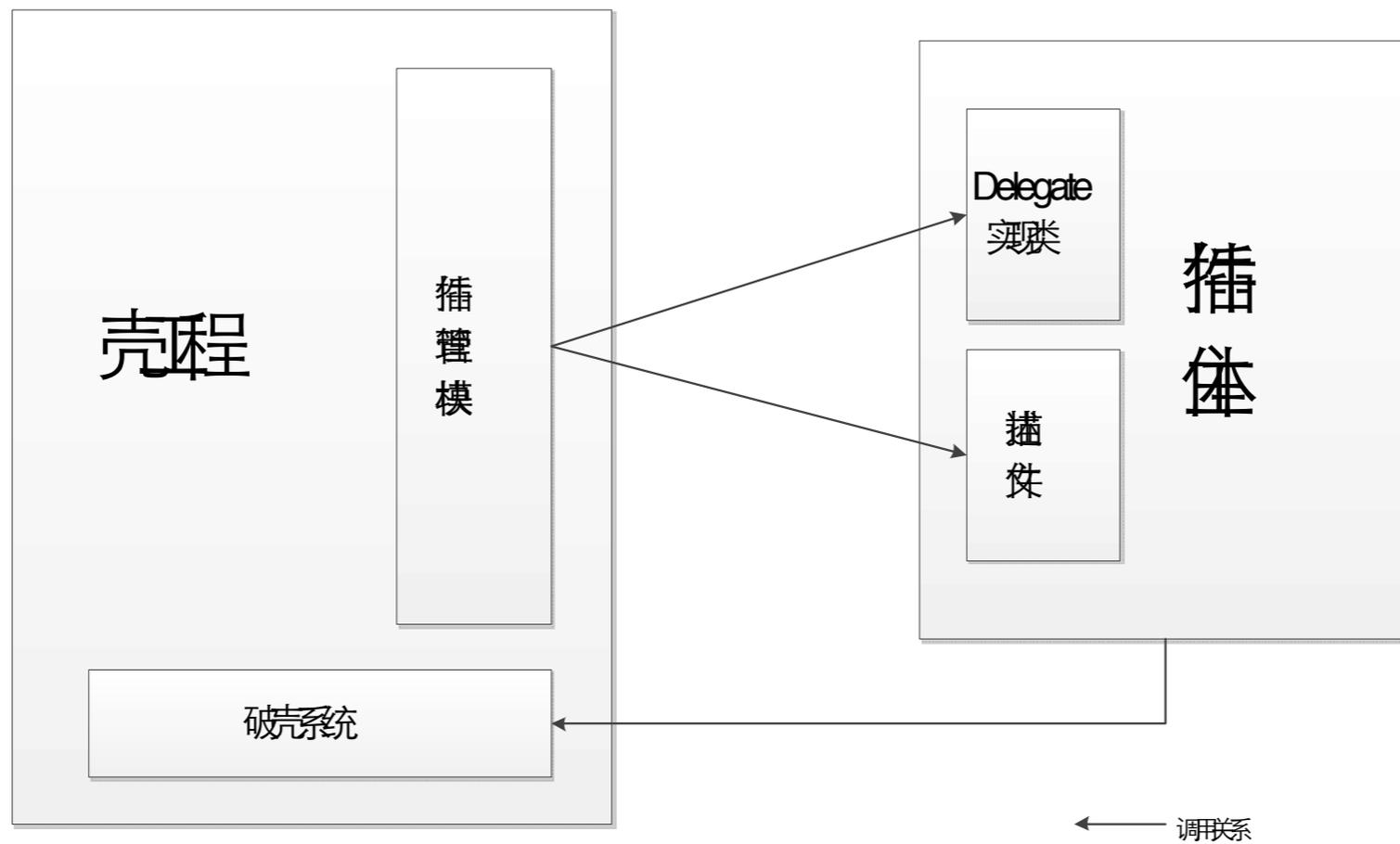
## 破壳系统

- 提供插件调用端能力接口
- 插件管理
- 插件入口自动生成

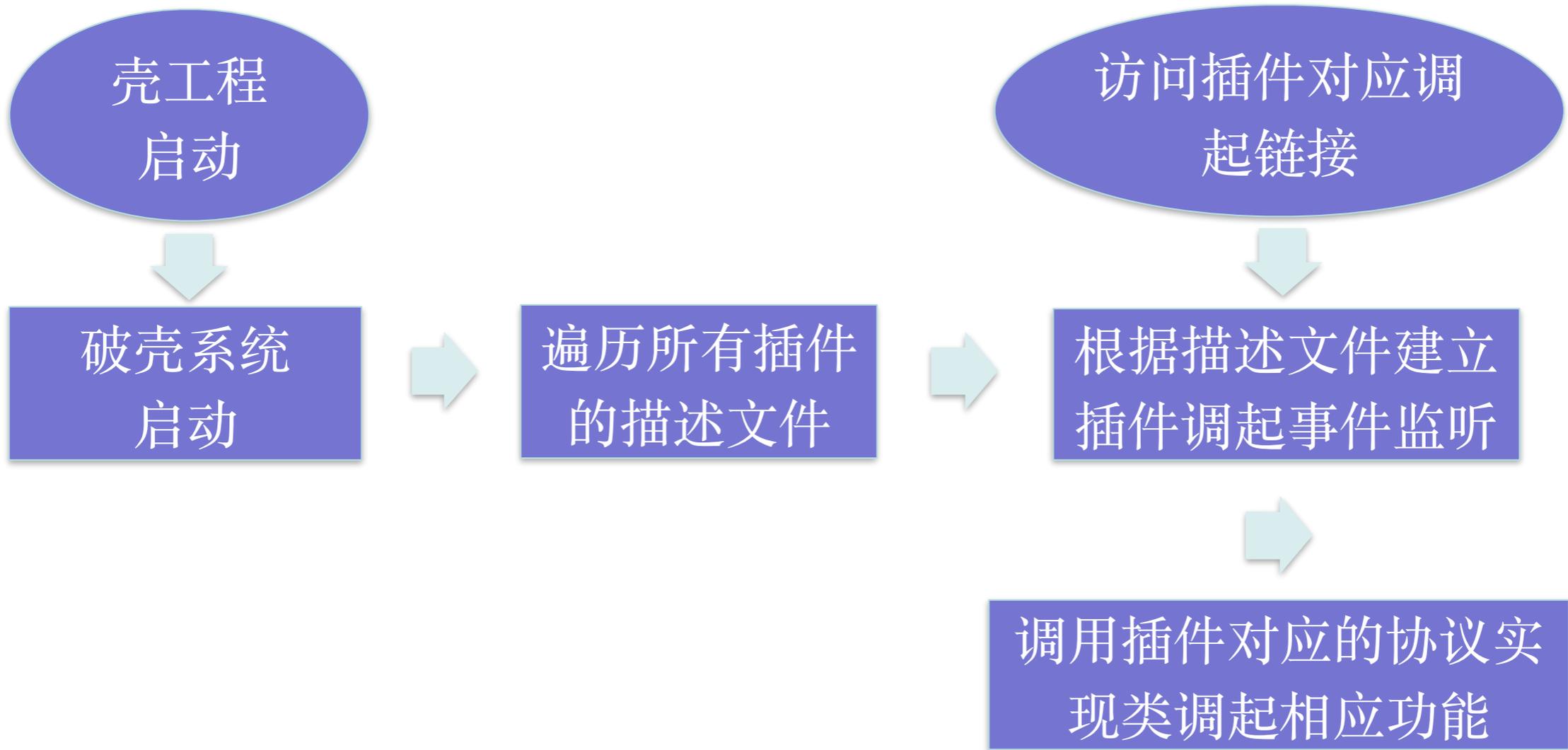
## 插件

- 插件主体
- 指定协议实现类
- 描述文件

# 方案



# 方案



## 实际问题

问题一：第三方库冲突

解决办法：第三方库统一由手百方进行接管，  
插件方在打包时将第三方库剥离

## 实际问题

问题二： 保证壳工程安全性

解决办法： 时间限制、 BundleID验证、 重要  
功能密钥保护

## 实际问题

### 问题三：端能力接口升级

解决办法：破壳接口是实际端能力接口的封装，保持不变，能力升级时修改实际端能力接口，破壳接口不变

# 效果

- **RD间联调开发**
  - 插件方依托壳工程自主开发，无需手百RD参与，省去联调过程
- **QA间联调测试**
  - 分散测试，插件测试托管给插件QA，手百只需后期进行简单准入测试
- **Bug定位及修复**
  - 可以单步调试，方便Bug的定位，修复也几乎不需要手百RD和QA参与

## 效果

- 插件接入流程化，规范化
- 可以快速、并行接入多个插件
- 插件与手百主体之间结构松散，低耦合
- 插件独立开发，与手百没有人员依赖

## 后续

- 增加插件灰度机制
  - 加大插件验证力度
- 自动化校验能力加强
  - 扫描私有API使用情况
  - 扫描系统方法覆盖情况
- 手百自身独立功能插件化
  - 将手百自身功能解耦

谢 谢