

# iOS 自动化测试框架选型

陈恒捷

- 引入自动化的目的
- 目前开源的 **iOS** 测试框架分类及对比
- 结合示例项目进行选型

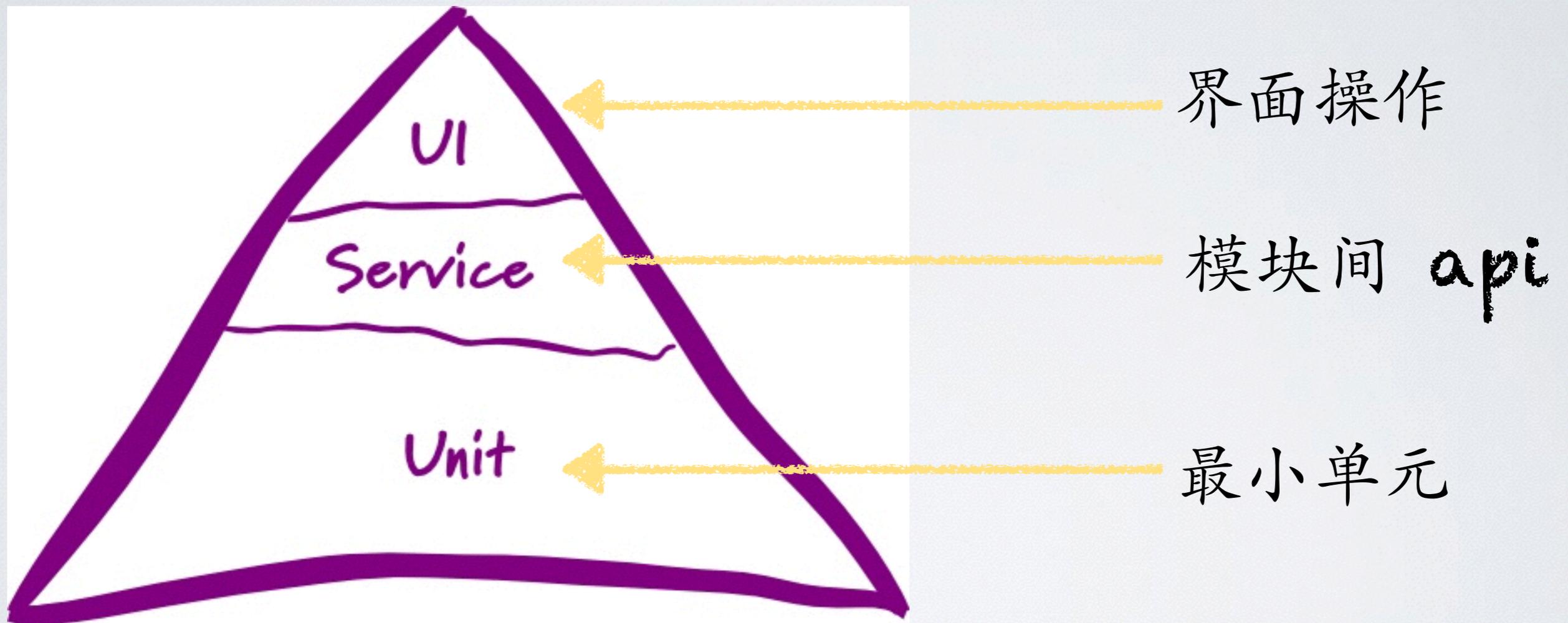
# 引入自动化的目的

降低成本

# 影响自动化成本的因素



# 科学降低成本的方法



根据测试点选择最佳的测试方式

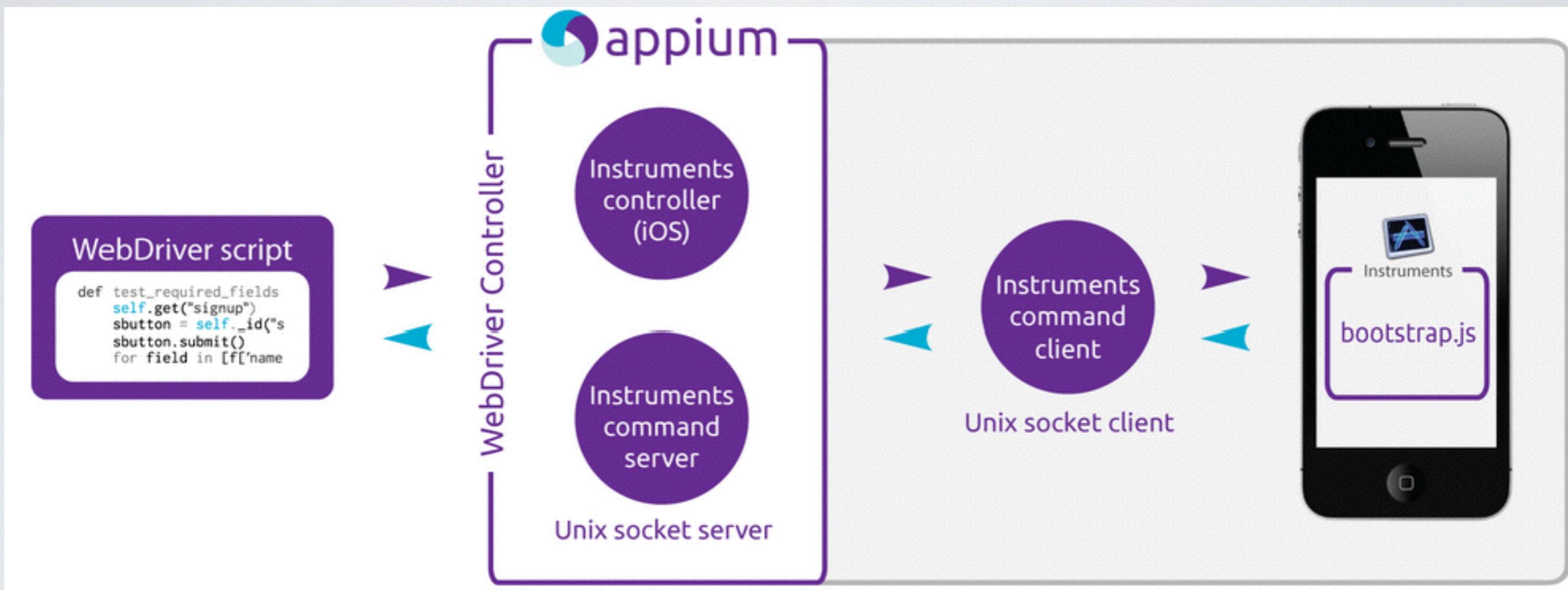
# iOS 测试自动化框架

- UIAutomation
- XCTest/XCUITest
- 注入编译型
- 另类方法

# UIAutomation

- 最早的，苹果官方推出的针对 UI 的自动化测试工具
- 优：可录制，不需要修改源码
- 缺：不能断点调试；从 Xcode 7 出来开始，Apple 不再维护
- 相关框架
  - 扩展型：Tuneup
  - 驱动型：Appium

# Appium 原理



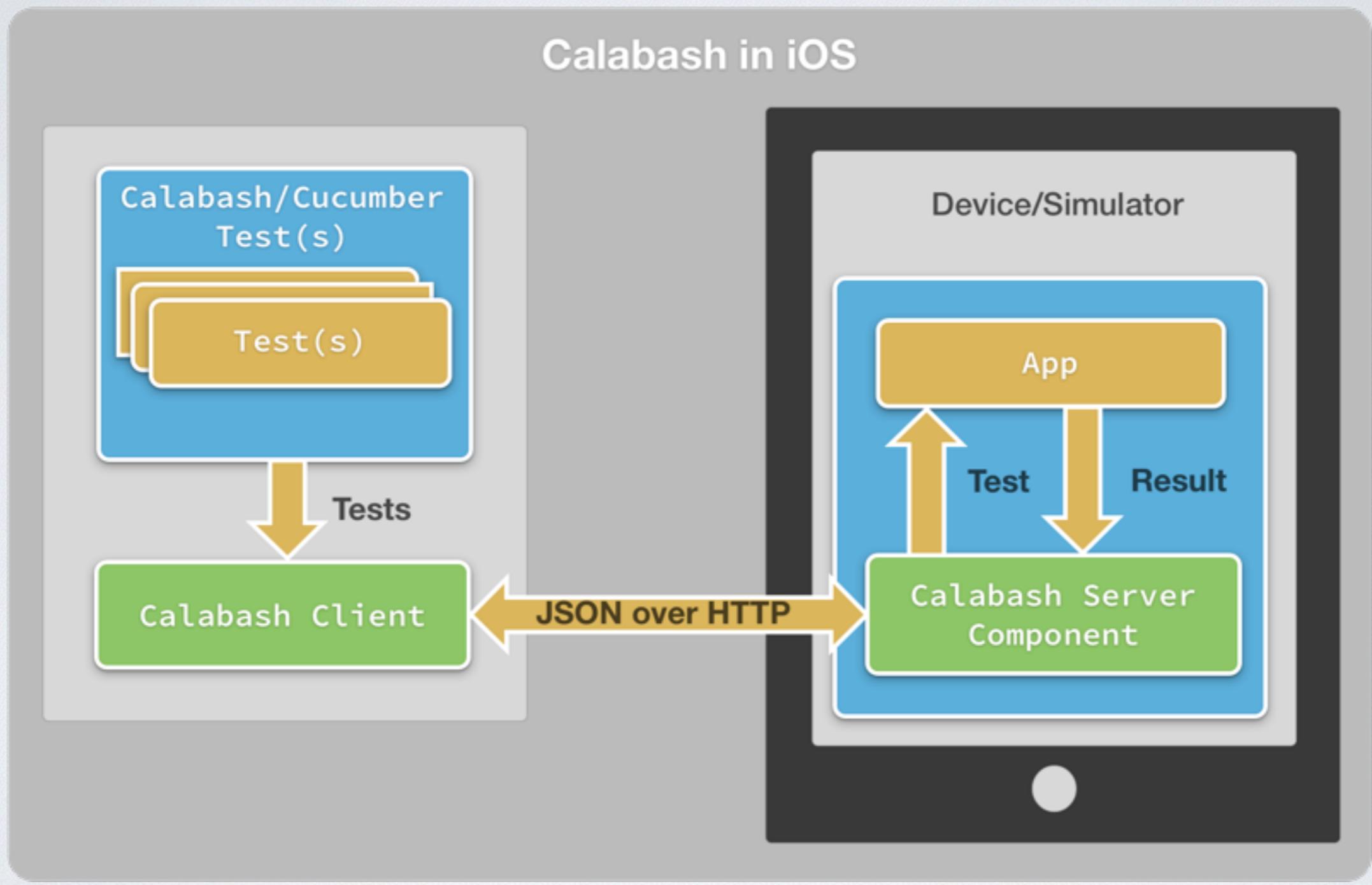
# XCTest

- 苹果在 Xcode 5 加入的单元测试框架，也可通过扩展完成功能自动化
- 优：从应用内部访问，可调用应用内部函数及获取内部状态
- 缺：只支持 Object-C 和 Swift
- 相关框架：
  - 官方：XCUITest
  - 私有 API：KIF
  - Google：EarlGrey

# 注入编译型

- 通过在应用中插码控制应用操作。添加到应用功能的部分多以 `agent` 形式存在。
- 优：在应用内部控制，比较灵活
- 缺：需要源码，灵活度不如 `XCTest`
- 相关框架：
  - `calabash-ios`
  - `Frank`

# Calabash 原理



# 另类方法

- 坐标系+图像识别型
- 优点：可跨应用，不需要源码
- 缺点：仅能检测纯 UI 界面，需越狱
- 相关框架
  - TouchElf 触摸精灵（需越狱）

# 各主要框架简单对比

	Tuneup	Appium	calabash- iOS	KIF	EarlGrey	XCUITest
脚本语言	Javascript	Python等 多种语言	Ruby cucumber	Object- c, swift	Object- c, swift	Object- c, swift
系统支持	iOS 4+	iOS 4+	iOS 7.1+	iOS 5.1+	iOS 8+	iOS 9+
结合单测 使用				✓	✓	✓
需要修改 源码			✓	✓	✓	✓

选型之前，先确认一个  
重要问题

拿得到被测应用源码吗？

没有源码，你的 iOS 自动  
化会很艰难

	有 app 源码	没 app 源码
Tuneup	√	√
Appium	√	√
calabash- ios	√	
KIF	√	
EarlGrey	√	
XCUITest	√	

可以不需要 app 源码的框架  
必须使用 debug 证书进行签名

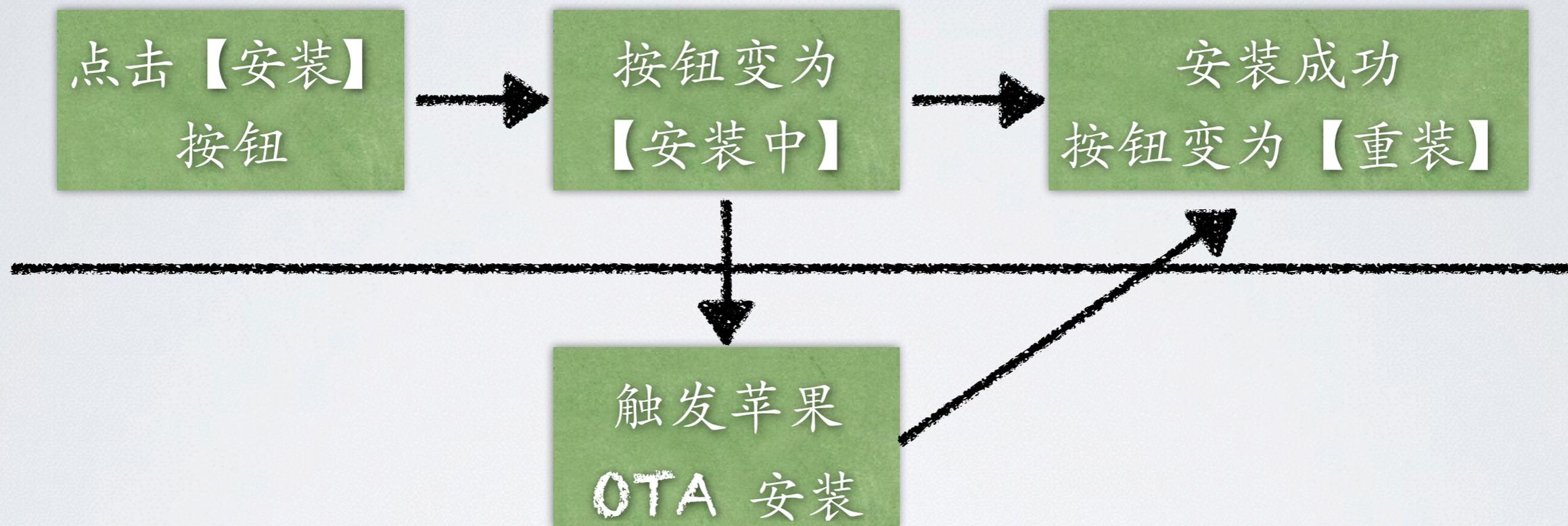
# 有了源码之后

- 能否满足核心业务功能测试点
- 编写、调试、维护、查错成本衡量

# 项目一

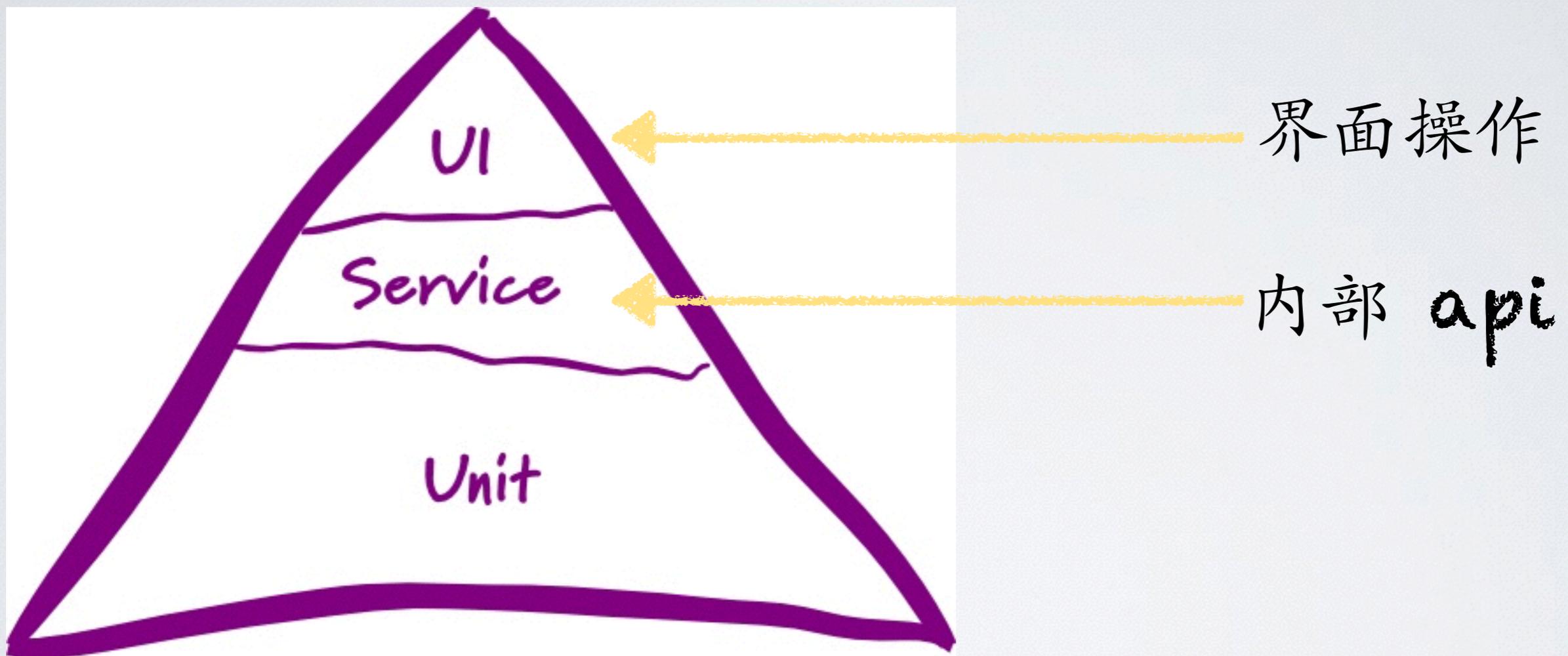
- 应用类型：应用市场
- 支持的系统版本：iOS 7+
- 核心用例：应用安装
- **app** 自动化目的：确认核心功能在每次代码提交后均正常

# 安装流程



应用大小不同，安装时间差异巨大  
iOS 7下，OTA 安装会回到桌面

# 如何抉择



用单元测试的 **Mock** 技术  
代替 **OTA** 安装

# 项目契合度对比

	Tuneup	Appium	calabash- iOS	KIF	EarlGrey	XCUITest
支持Mock 技术				√	√	√
支持 iOS 7+				√		

# 成本评估

(越少越好)

编写成本

★★★★★

调试成本

★★

维护成本

★★

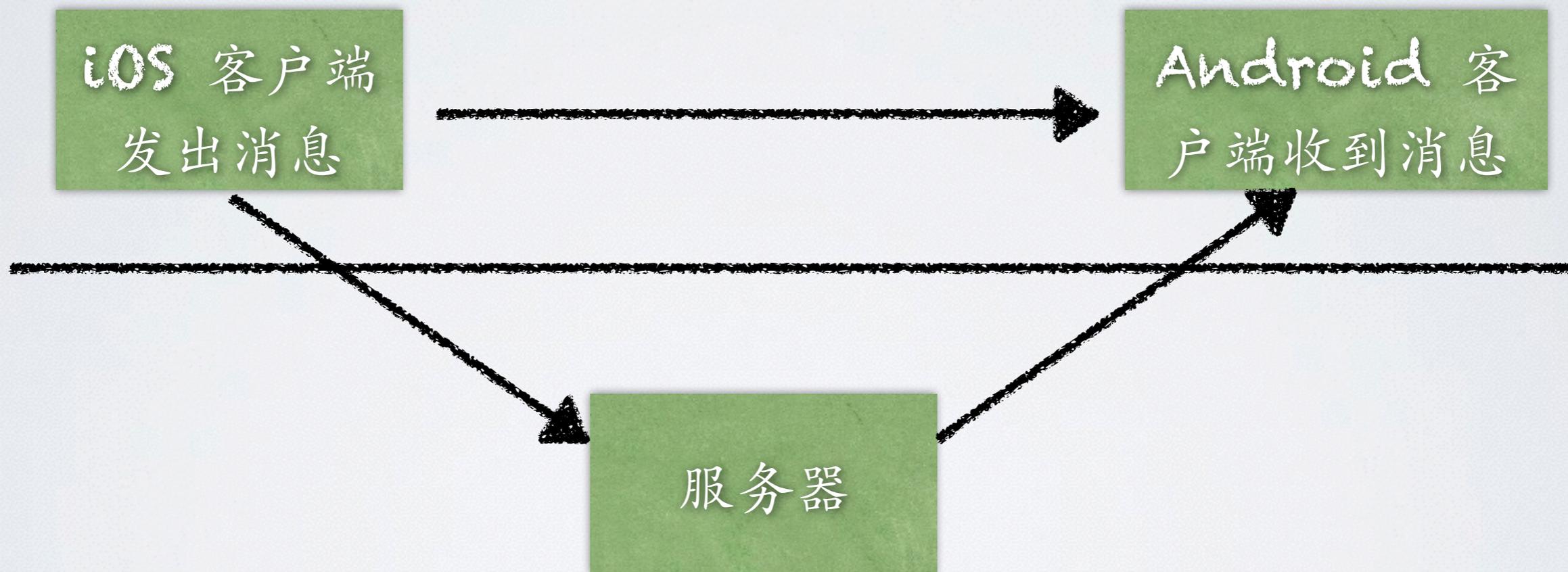
查错成本

★

# 项目二

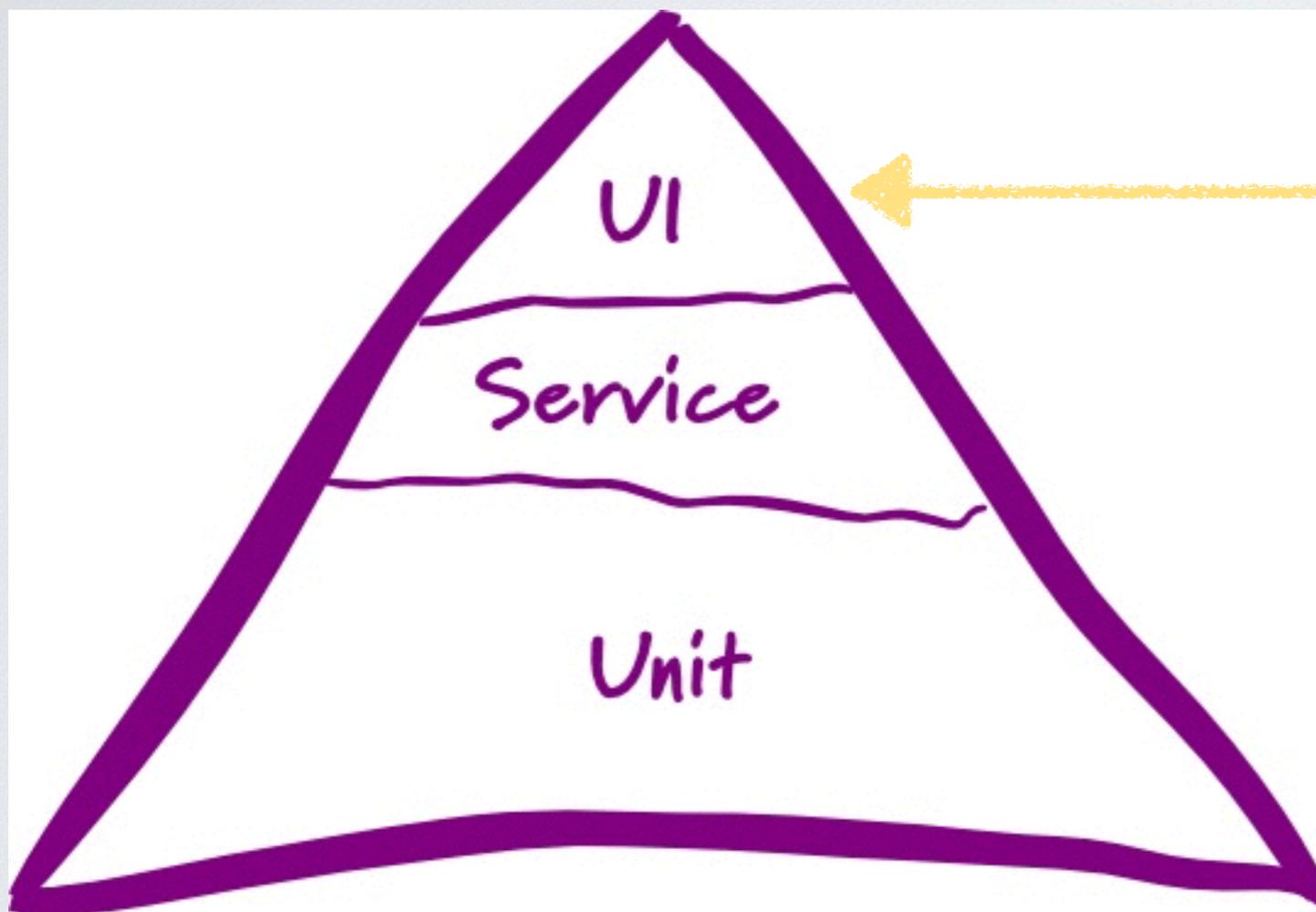
- 应用类型：社交 app
- 支持的系统版本：iOS 7+, Android 4.0+  
(两版本上用户界面基本一致)
- 核心用例：聊天
- 自动化目的：发版前核心使用场景的自动回归测试

# 聊天流程



多设备控制  
多平台

# 如何抉择



界面操作

对于用户故事，直接用 **UI** 更方便

# 项目契合度对比

	Tuneup	Appium	calabash -iOS	KIF	EarlGre y	XCUITes †
支持 Android		✓	✓ (calabash- Android)			
支持多 设备操 作		✓	✓			

# 成本评估

(越少越好)

编写成本

★★

调试成本

★★★

维护成本

★★

查错成本

★★★★★

# 总结

没有最好，只有最合适

NJSD

中国（南京）软件开发者大会  
China (Nanjing) Software Developers Conference

2016

Thank you