

MTSC2017

第三届中国移动互联网测试开发大会

iOS 远程真机调试

百度MTC 项光特

场景



自动化脚本测试遇到问题：

- 网络原因导致的页面元素加载延迟
- 设备分辨率导致的页面元素无法直接点击

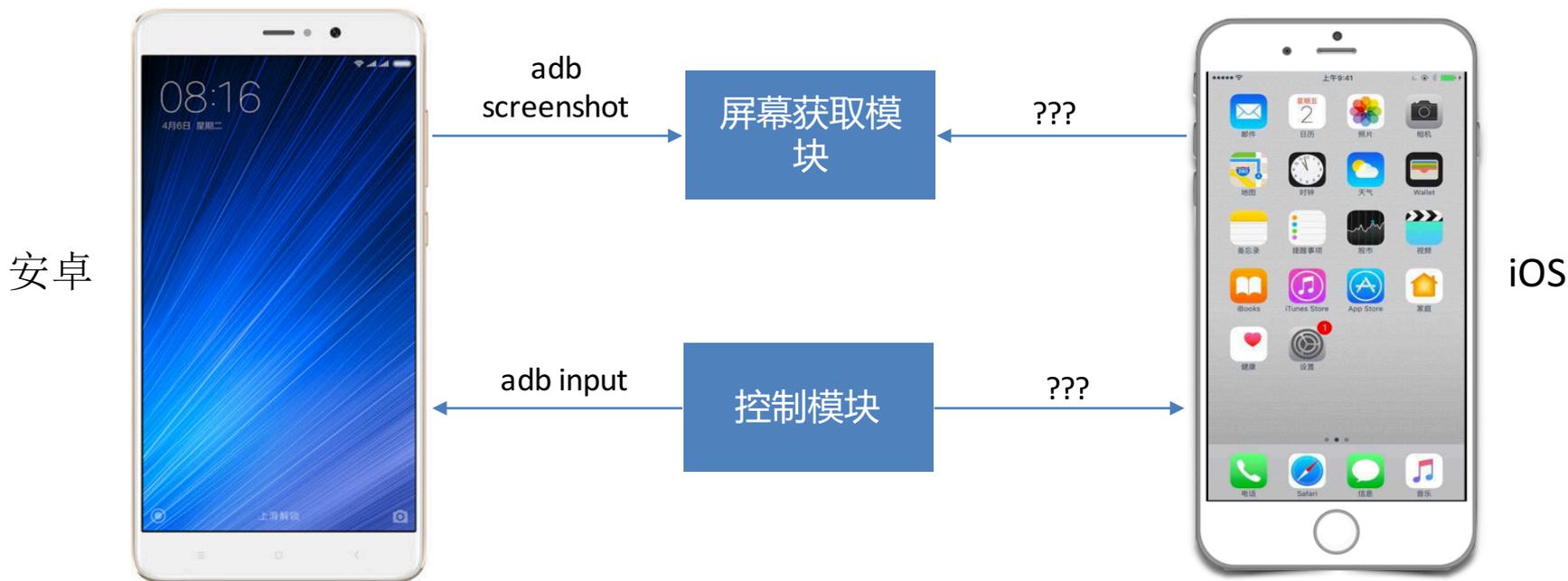
这些问题我们需要通过观察实际手机界面情况来修改脚本
我们希望有工具能远程的方式实时查看和操作iOS设备



待解决问题

关键功能：

- 从宿主机实时获取当前设备屏幕内容
- 从宿主机发送触控事件给设备

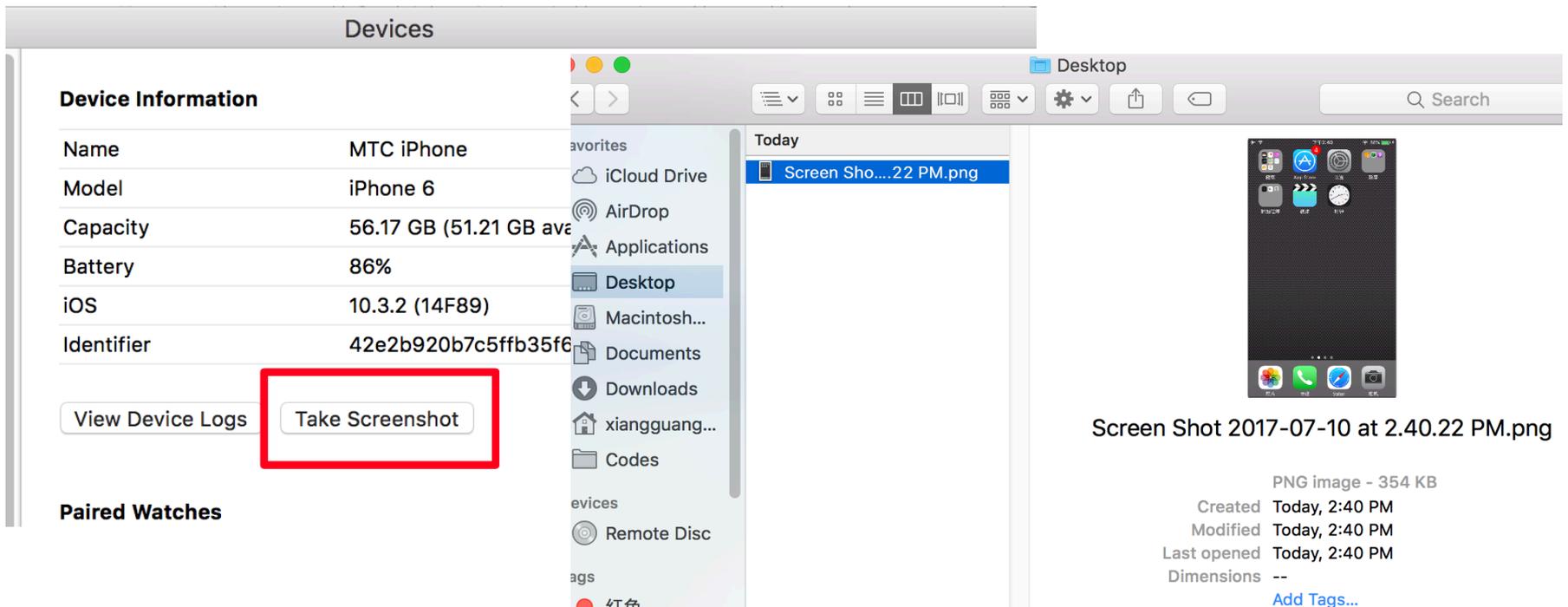




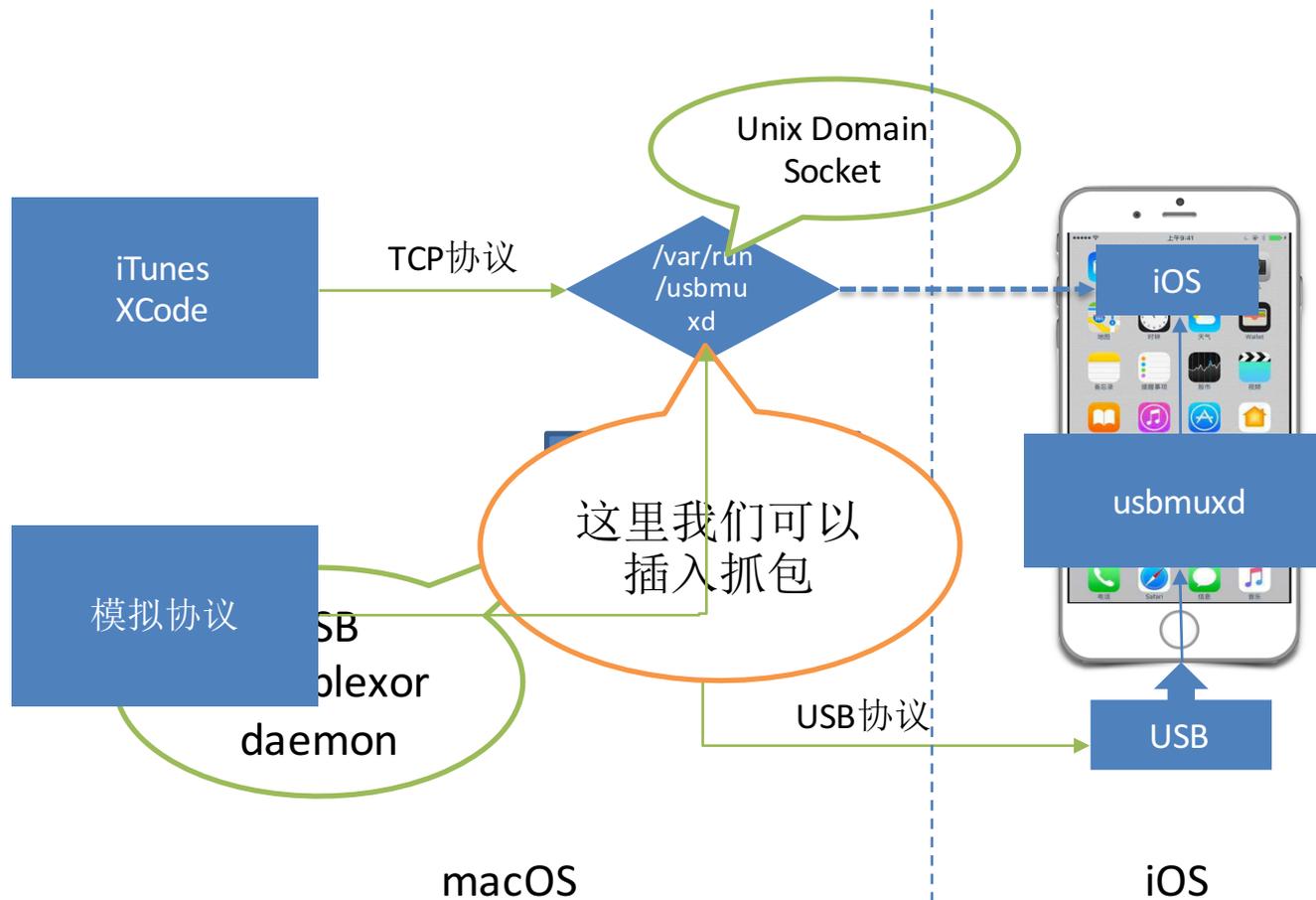
屏幕获取模块

XCode工具Devices截图功能

XCode自带的Devices工具提供了获取设备屏幕截图的功能。
点击按钮后，宿主机桌面会出现一个设备当前屏幕截图的图片文件。



iOS设备通讯协议



libimobiledevice开源工具

- libimobiledevice是一个根据以上协议实现开源工具集合，里面常用的工具如下：

ideviceinfo

获取设备状态信息

ideviceinstaller

安装APP

idevicebackup

备份设备

idevicescreenshot

获取屏幕截图

idevicesyslog

获取日志

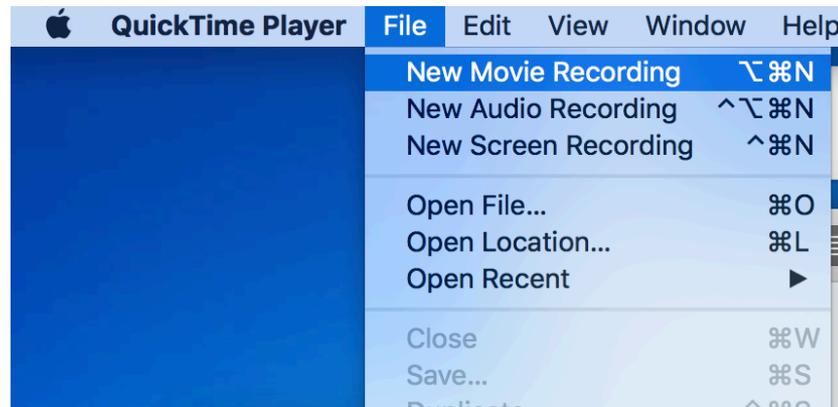
idevicescreenshot

- 根据屏幕大小和显示内容的不同, idevicescreenshot可以在0.2~0.5秒内返回一台iPhone的屏幕截图
- 综合下来可以获取到每秒2~5帧流畅度的画面

```
sh-3.2$ time idevicescreenshot -u 42e2b9
Screenshot saved to screenshot-2017-07-1
real    0m0.292s
user    0m0.014s
sys     0m0.014s
```

QuickTime Player的iOS屏幕录制

- macOS 10.10以后，系统内置的QuickTime Player支持iOS8以上的设备屏幕录制。
- 可以选择已连接的iOS设备，以每秒30帧的速率获取屏幕内容。



ios-minicap工具

- ios-minicap是openstf提供的一个模拟QuickTime Player类似操作实现的开源工具。
- 通过从视频流里截取关键帧的方式，可以达到每秒返回20~30张屏幕实时截图。



屏幕内容获取解决方案

方案1

idevicescreenshot

速率：2~5帧

限制：无

方案2

ios-minicap

速率：20~30帧

限制：每台mac仅有一台设备可用

比较下来，我们最终综合了两种解决方案：

- 优先选用ios-minicap解决方案
- 无法使用ios-minicap时回退到idevicescreenshot解决方案



控制模块

控制模块

- 我们遇到的问题：
 - 无官方工具
 - 沙盒机制
 - 墓碑式后台机制

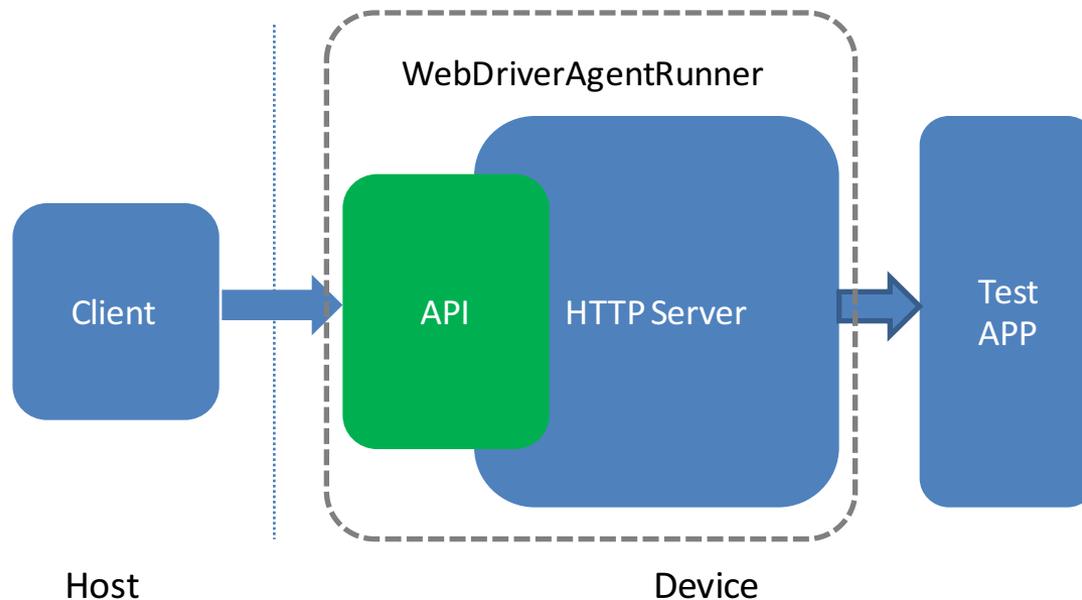


WebDriverAgent测试框架

- WebDriverAgent是Facebook推出的一个基于XCUITest的开源测试框架。
- XCUITest是Apple随XCode7发布以来引入的一套UI测试框架，并在XCode8中做了较多的升级与改进。
- XCUITest能做到启动后进入后台，并在后台启动目标测试APP，并与之进行交互。

WebDriverAgent测试框架

- WebDriverAgent在设备端额外启动一个HTTP Server
- 提供用于元素查找定位及元素级别的操作的API
- 外部程序通过API对APP进行测试



私有API

- 私有 API 是指苹果未公开的一些方法
- 私有 API 可以实现一些开放 API 不能实现的效果，功能强大，效果非凡。
- 可以通过在设备使用RuntimeBrowser等工具导出私有API的Header文件

XCTestGenerator

XCTestGenerator Class Reference

Inherits from	NSObject
Declared in	XCTestGenerator.h

可以根据需要生成不同类型的屏幕触控事件。

+ `sharedGenerator`

- `doubleTapAtPoint:orientation:handler:`

- `pinchInRect:withScale:velocity:orientation:handler:`

- `pressAtPoint:forDuration:liftAtPoint:velocity:orientation:name:handler:`

- `pressAtPoint:forDuration:orientation:handler:`

- `rotateInRect:withRotation:velocity:orientation:handler:`

- `tapAtPoint:orientation:handler:`

- `tapWithNumberOfTaps:numberOfTouches:inRect:orientation:handler:`

- `twoFingerTapInRect:orientation:handler:`

MTCRunner后台操作APP

- MTCRunner是一个XCUI Test测试APP
- 借鉴了WebDriverAgent的工作方式，通过web server的方式暴露操作接口
- 通过私有API XCEventGenerator实现了设备级别的触控、长按、滑动等操作
- 通过iproxy工具把接口映射到宿主机
- 宿主机通过curl命令的方式实现对设备的控制

```
curl -X POST -d '{"x":0.5,"y":0.5}' http://localhost:8000/tap
```

MTCRunner

- 一些额外的功能：

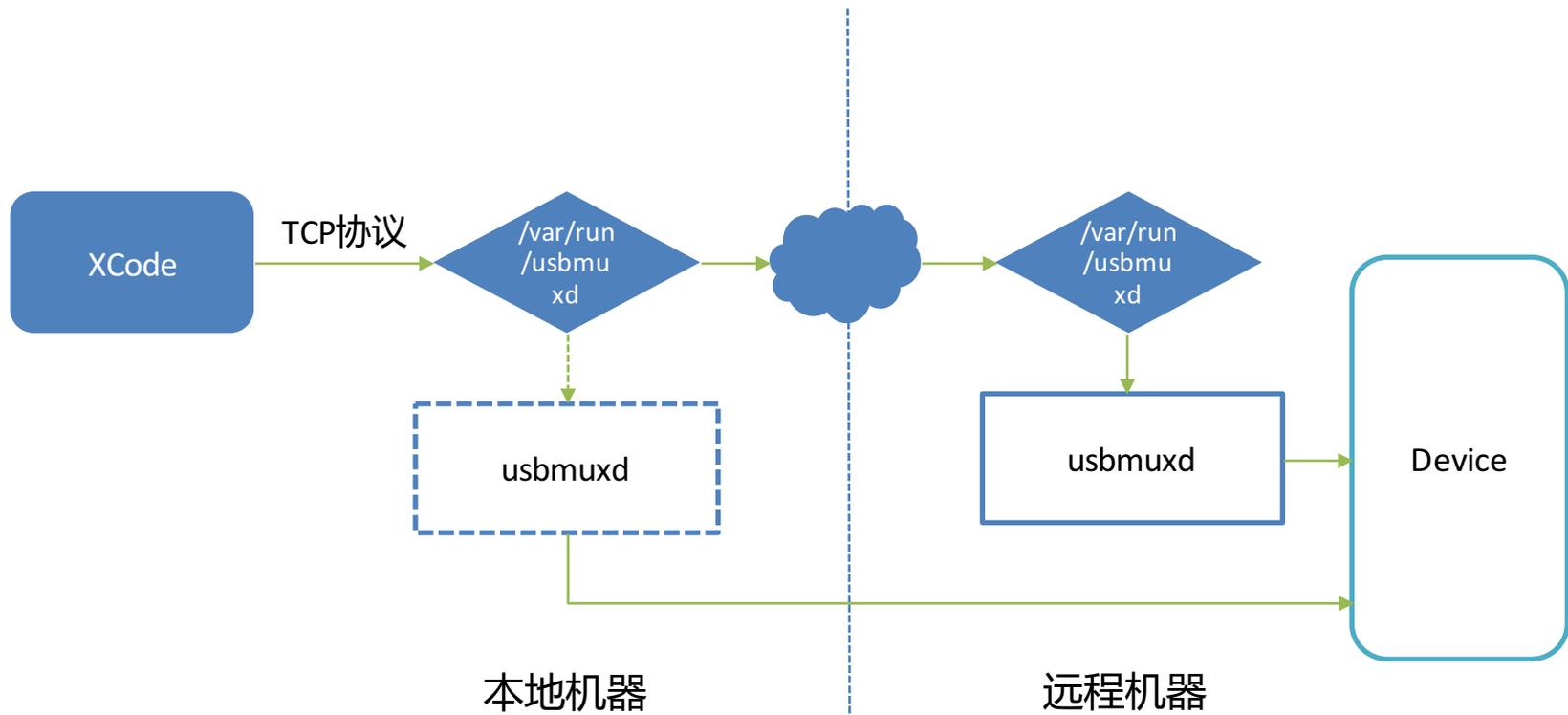
实时CPU使用率

实时内存使用率

App的启动与终止

Monkey操作

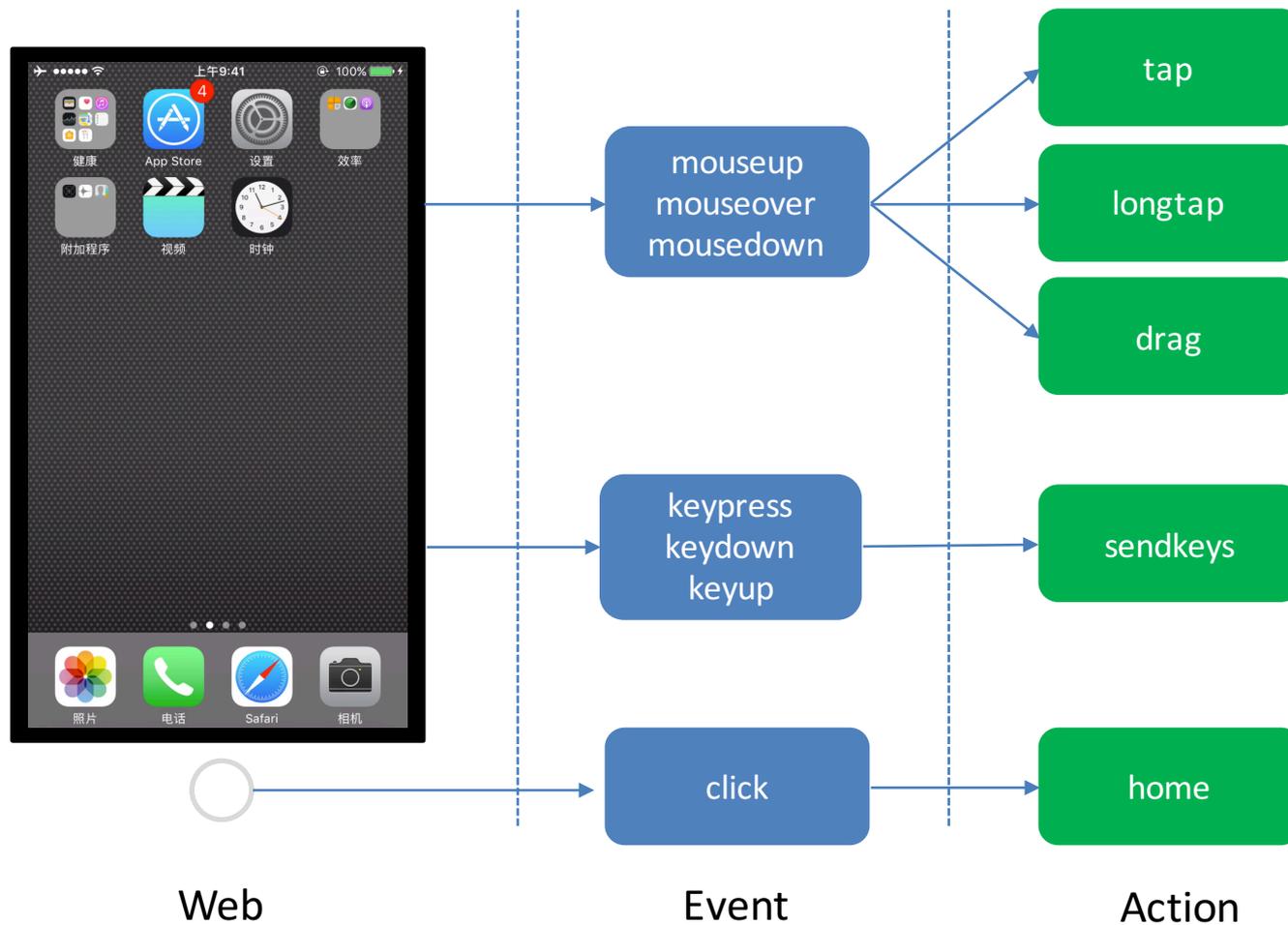
远程XCode调试



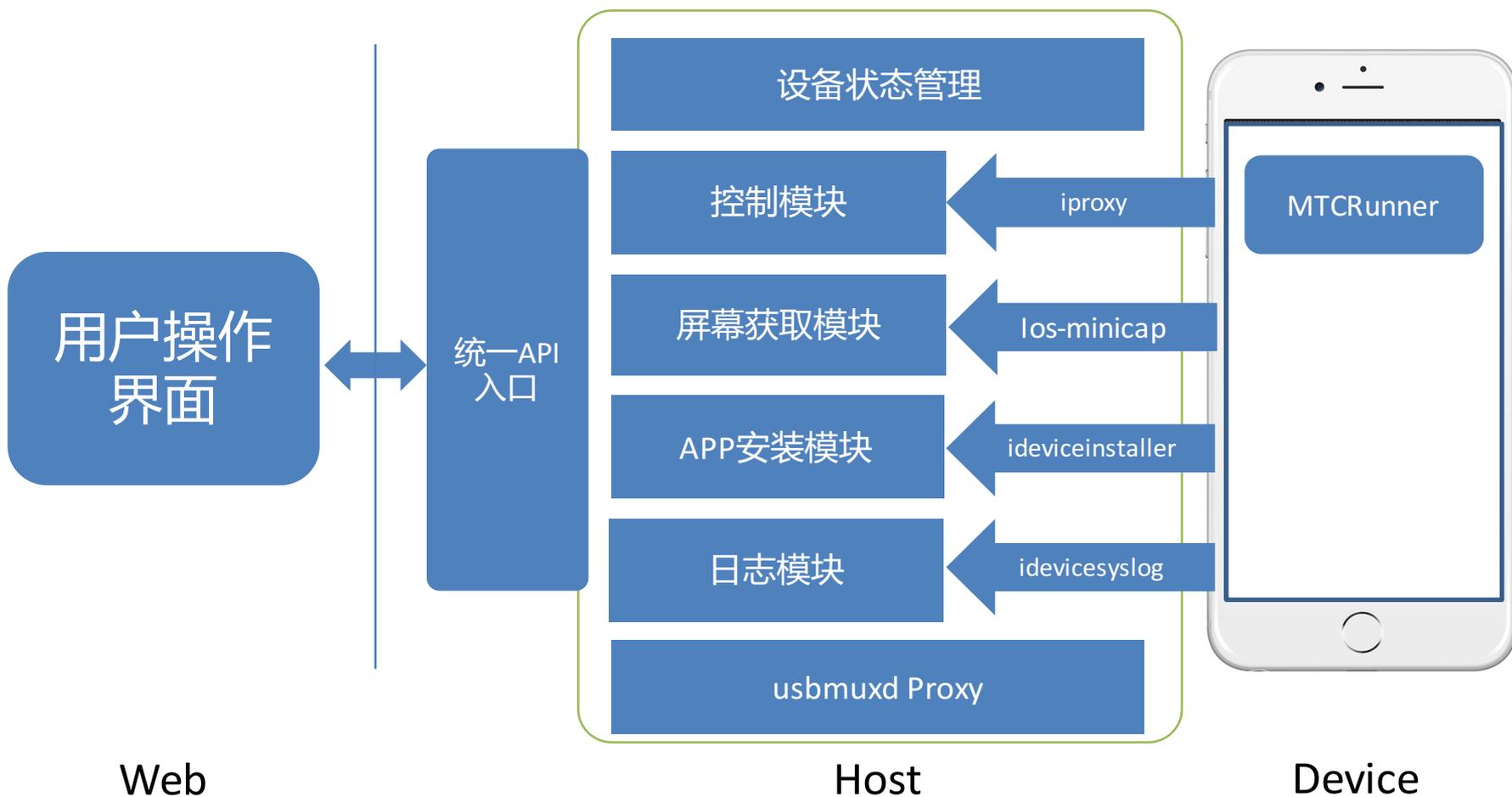


整体解决方案

操作界面



远程真机调试整体解决方案



MTC远程真机调试内部DEMO效果

Baidu MTC
管理控制台
产品服务
服务支持
mtcsuper VIP

[移动App测试](#) / [远程真机调试](#) / [iOS设备调试](#)

iPhone 6



+ 安装应用 📷 截图 使用中

安装本地IPA文件



设备信息-iPhone 6

版本: 10.0.2 分辨率: 750x1334

RAM: 1G 主频: 1.4GHz

CPU: 2核 Apple A8

设备UDID: 42e2b920b7c5ffb35f6b6e4c18a73308b86e311a

📄 日志log ⚙️ 设置

All
Time
Service
Pid
LogSource
Message
开始获取
清空

Severity	Time	Service	Pid	LogSource	Message
Debug	2017-02-12T12:00:41.000Z	kernel(AppleBiometricSensor)	0	log	Status={0x80, 0x20, 00, 0x10, 00, 00, 00, 0x11, 0x1, 0x1, 00, 0x4, 0xe9, 0x2, 0x9, 00}
Notice	2017-02-12T12:00:41.000Z	securityd	103	log	cert[0]: NonEmptySubject = (leaf)[] > 0
Notice	2017-02-12T12:00:41.000Z	securityd	103	log	cert[0]: NonEmptySubject = (leaf)[] > 0
Debug	2017-02-12T12:00:41.000Z	kernel(AppleBiometricSensor)	0	log	exit
Debug	2017-02-12T12:00:41.000Z	kernel(AppleBiometricSensor)	0	log	entry
Debug	2017-02-12T12:00:41.000Z	kernel(AppleBiometricSensor)	0	log	exit
Debug	2017-02-12T12:00:41.000Z	kernel(AppleBiometricSensor)	0	log	mask=0xFF, head=false

依赖

- macOS10.10以上的mac设备
- iOS9以上iOS设备，无需越狱
- iOS手机数据线
- Apple的开发者证书

实现效果

- 远程以20 ~ 30帧的速率查看设备当前屏幕内容
- 远程通过鼠标点击模拟触摸操作，操作延迟感在200毫秒以内
- 远程使用xcode直接进行调试

参考资料

- libimobiledevice: <https://github.com/libimobiledevice/libimobiledevice>
- ios-minicap: <https://github.com/openstf/ios-minicap>
- WebDriverAgent: <https://github.com/facebook/WebDriverAgent>
- Appium XCUITest Driver: <https://github.com/appium/appium-xcuitest-driver>
- The iPhone Wiki: <https://www.theiphonewiki.com/>
- iPhone Development Wiki: <http://iphonedevwiki.net/>
- iOS Runtime Header: <https://github.com/nst/iOS-Runtime-Headers>

MTSC2017

第三届中国移动互联网测试开发大会

Thanks!