



QCon

www.qconferences.com





Hybrid应用的质量保证实践

By 陈晔 (Monkey陳曄曄)



重点

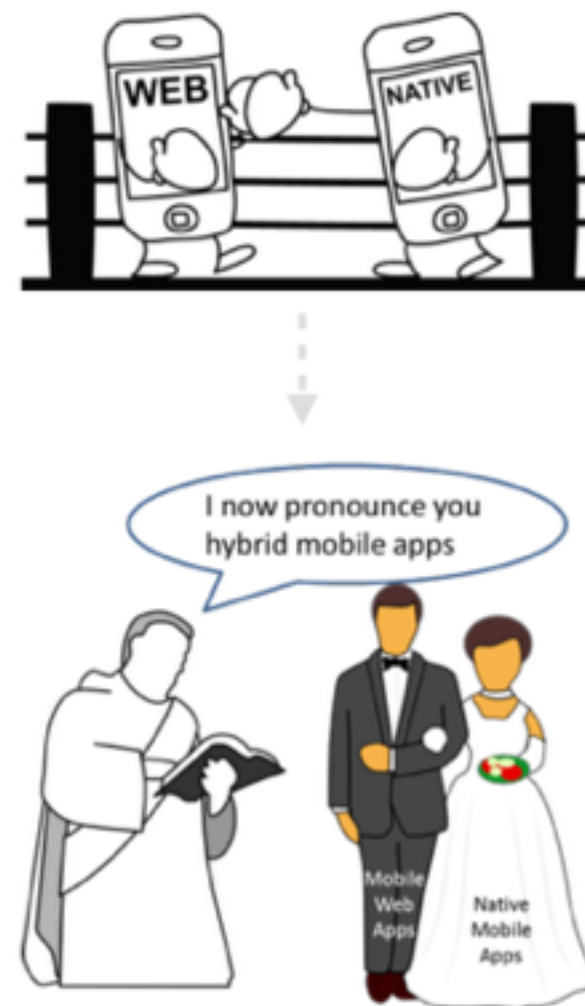


- Hybrid在支付宝中的应用场景
- 基于自定义H5容器的应用开发
- 研发过程中的测试
- 线下线上数据巡检
- 结合业务的Hybrid应用自动化平台
- 持续集成

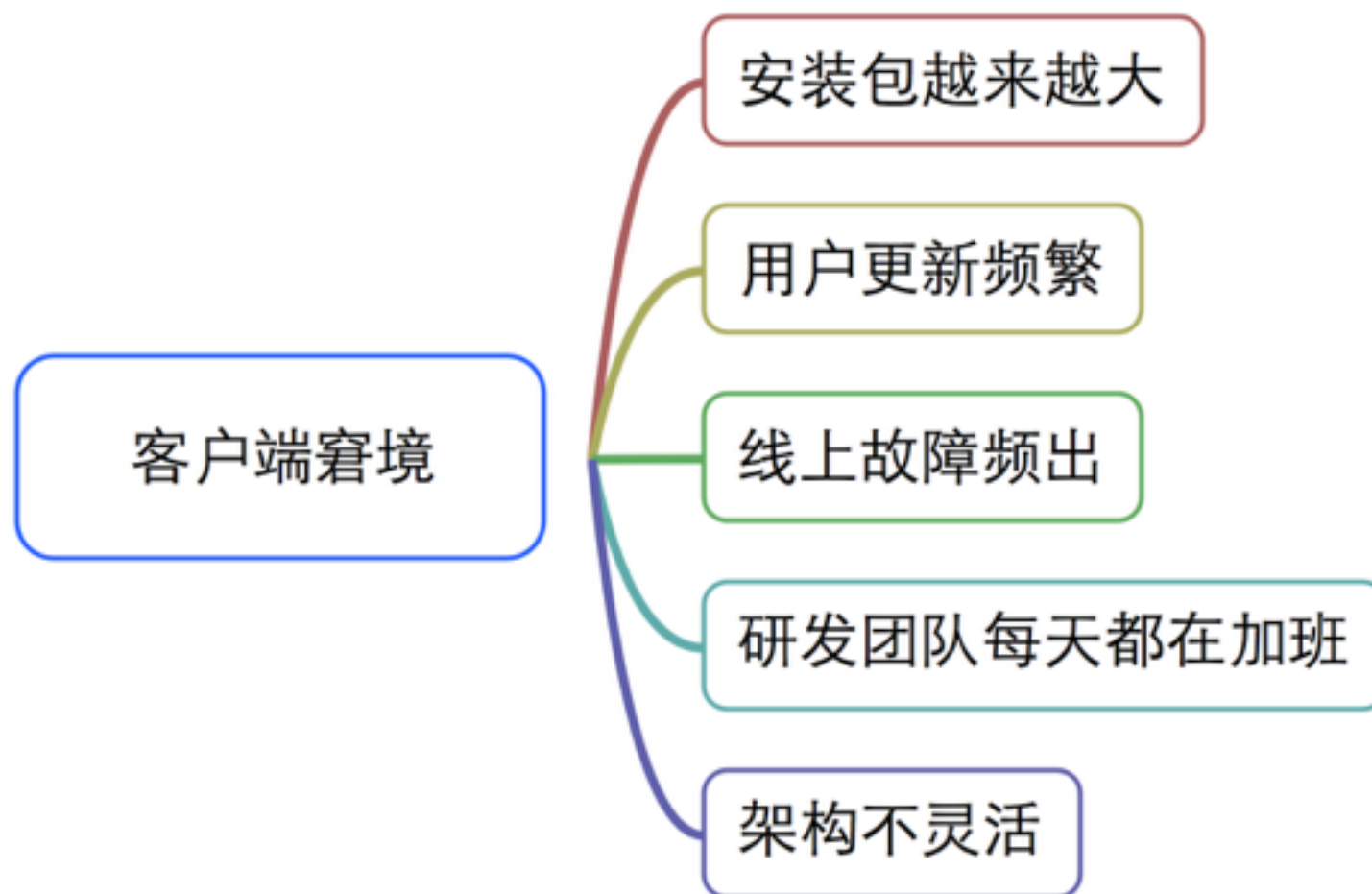


重新认识 Hybrid

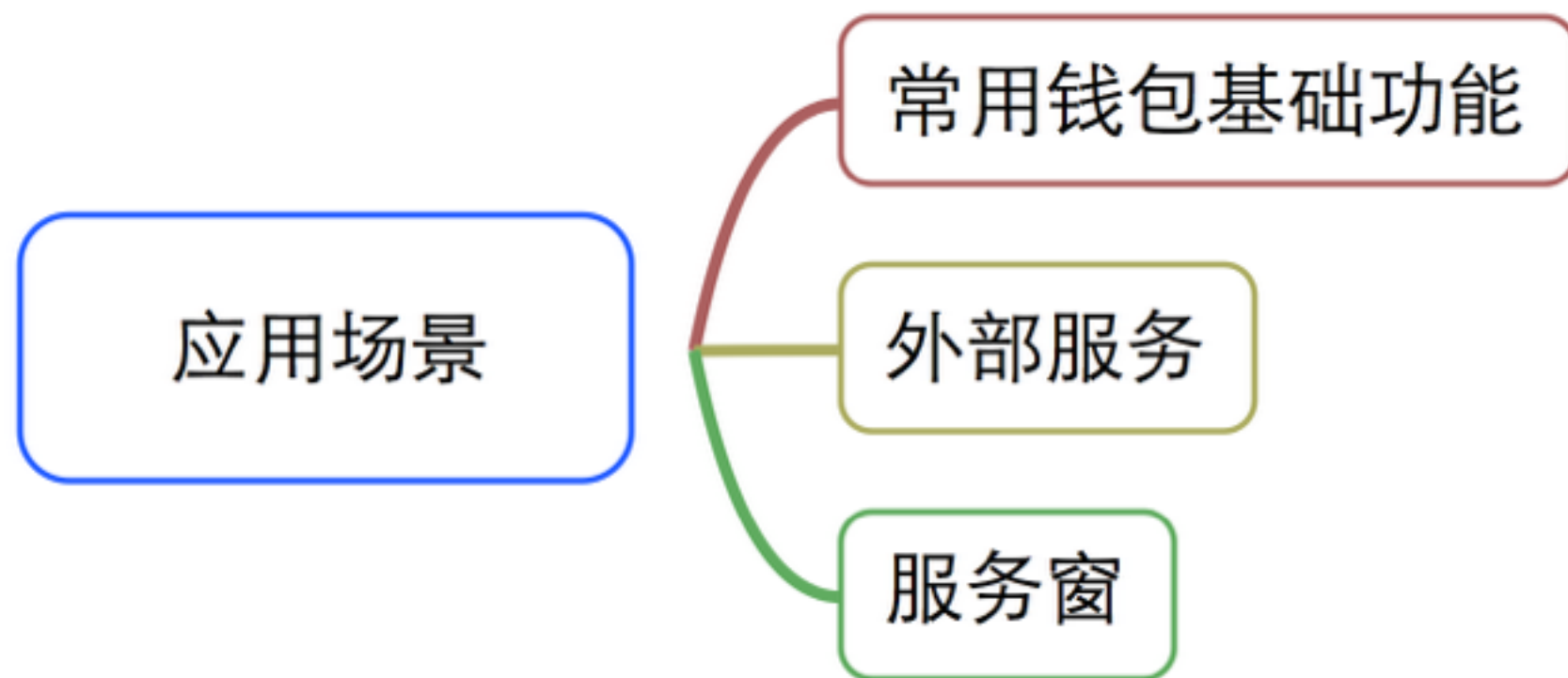
	Native	H5
性能	✓	
硬件调用能力	✓	
安全	✓	
敏捷开发		✓
发布		✓
跨平台		✓
可访问性		✓



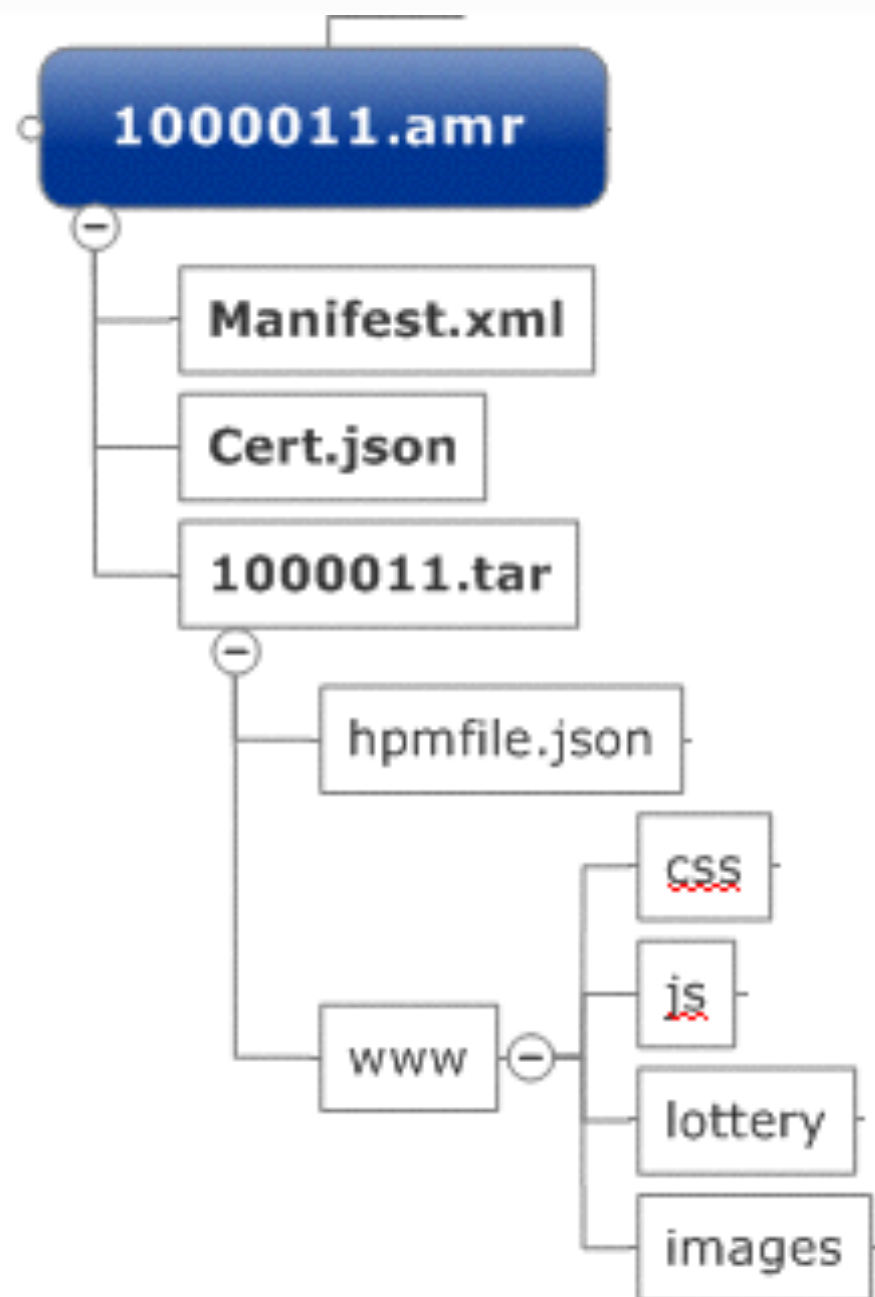
客户端窘境



应用场景



钱包的推包机制 (amr)



根目录

应用中心配置文件

H5app安全加签文件

资源tar包

资源配置文件

资源文件

一把利剑——H5 container



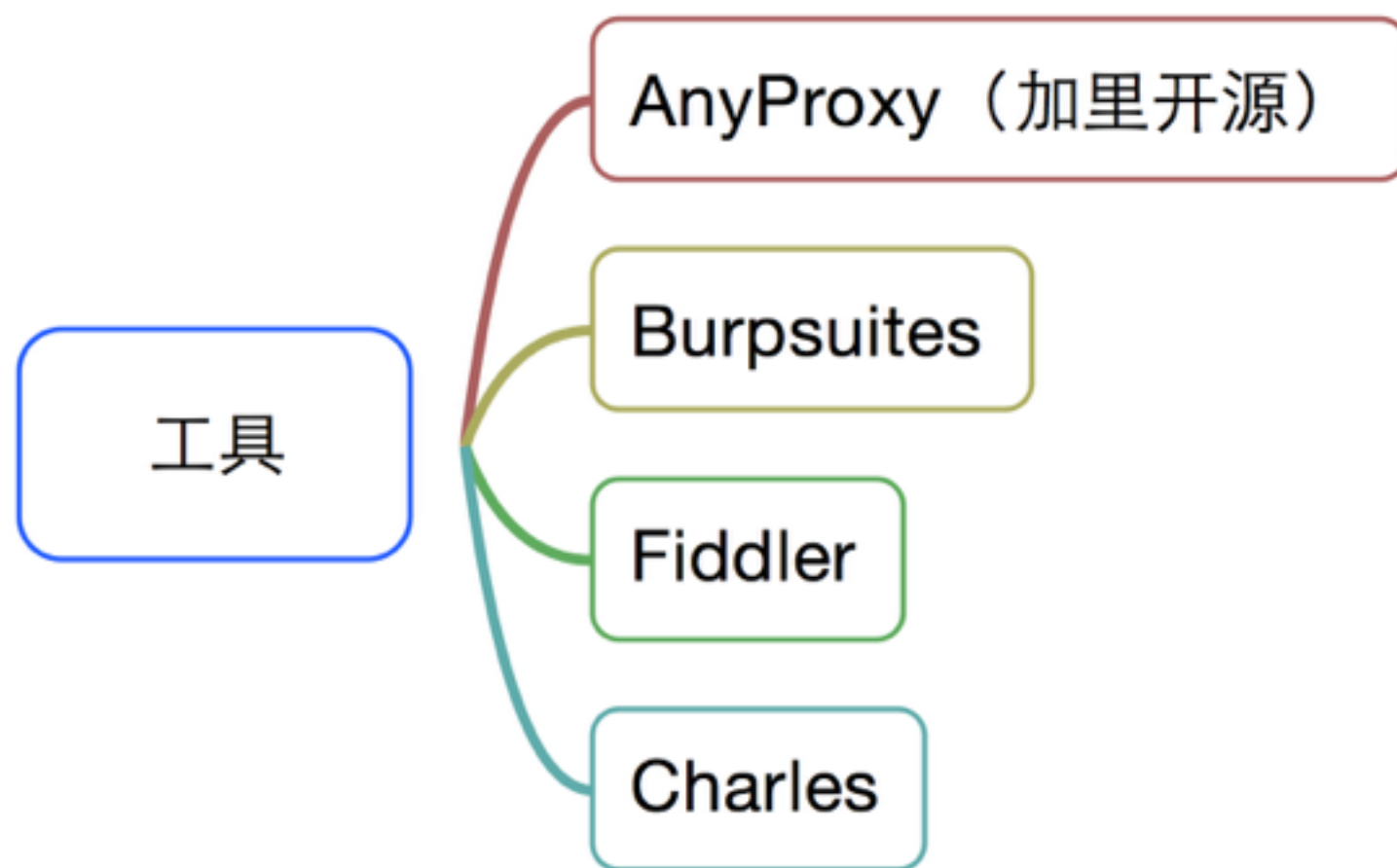
容器功能



研发过程中的测试



借助工具



AnyProxy简述



```
var proxy = require("anyproxy");

//create cert when you want to use https features
//please manually trust this rootCA when it is the first time
!proxy.isRootCAFileExists() && proxy.generateRootCA();

var options = {
  type      : "http",
  port      : 8001,
  hostname  : "localhost",
  rule      : require("path/to/my/ruleModule.js"),
  dbFile    : null, // optional, save request data to a
  webPort   : 8002, // optional, port for web interface
  socketPort : 8003, // optional, internal port for web
  webConfigPort : 8088, // optional, internal port for web
  throttle  : 10, // optional, speed limit in kb/s
  disableWebInterface : false, //optional, set it when you d
  silent    : false, //optional, do not print anything i
};
new proxy.proxyServer(options);
```

实际数据



测试场景	接口消耗时间	实际消耗时间	接口消耗流量	测试场景	接口消耗时间	实际消耗时间	接口消耗流量
首次启动	<input type="text"/>	<input type="text"/>	<input type="text"/>	非首次	<input type="text"/>	<input type="text"/>	<input type="text"/>

前端体验指标



- 图片加载时间
- 超时时间
- 界面整体加载时间
- 接口流量数据大小
- 减少白屏时间

又一大利器——hpmsim (伙爪)



原理

1. 安装Xcode
command line
tools

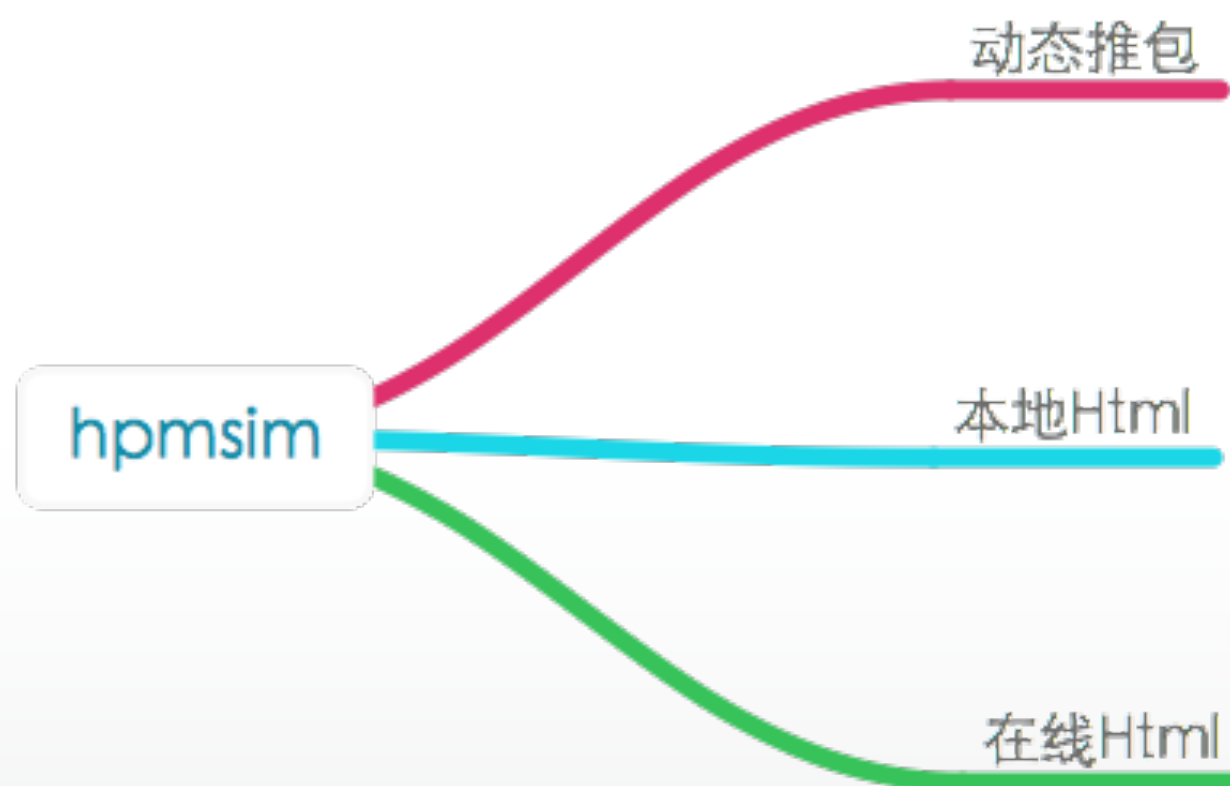
2. 创建hpmsim模拟器

3. 启动hpmsim

4. 安装并启动
支付宝钱包

5. url启动H5App

hpmsim



Sample



```
MonkeydeMacBook-Pro:AndroidLibrary2 monkey$ hpm sim -u http://www.baidu.com
```

```
# - hpm sim 功能目前属于beta版。报错信息请反馈给 @爪爪
```

```
# - 必须安装 Xcode6 以上！内网急速下载地址：http://h5.alipay.net/portal/article/151
```

```
# launch hpmsim success! D558B885-E170-4E2A-9403-1ACCB9292541
```

```
# Portal_8.6.app is exist
```

```
# install Portal_8.6.app success!
```

```
# xcrun simctl openurl booted 'alipays://platformapi/startapp?appId=20000067&url=http%3A%2F%2Fwww.baidu.com'
```

可配置的一个应用设置



iOS Simulator - hpmsim -

Carrier 12:52 AM

< Settings

支付宝钱包

WEB远程调试

Weinre



Server

ux.alipay-inc.com

Port

9119

容器调试面板



WEBVIEW自动化测试

开关



监测插件控制面板(修改生效需要重新启动)

总开关



控制面板



内存查看开关



性能监控
支持自动化

设置

内存监控

应用环境

远程调试

结合 hpmsim

Hybrid非功能测试关注点



- 界面各元素控件加载时间（容器埋点反馈）
- 每个功能场景的对应接口的流量消耗（通过代理实现）
- 每个功能场景的对应接口消耗的时间数据（通过代理实现）
- Hybrid容器加载的时间（容器反馈）

以上都需要在2G， 2.5G， 3G， wifi网络下进行测试

Hybrid应用线上巡检



- 对Hybrid调用的服务进行主动的定期巡检
 - 主动巡检接口是否通畅
 - 报警机制
 - 定期邮件报告



巡检历史

开始日期 2014-10-01 结束日期 2015-04-14 查看



110%

100%

90%

80%

70%

60%

50%

40%

30%

20%

10%

Dec 22

Dec 29

Jan

Jan 12

Jan 19

Jan 26

Feb

Feb 09

Feb 16

Feb 23

Mar

Mar 09

Mar 16

Mar 23

Mar 30

Apr

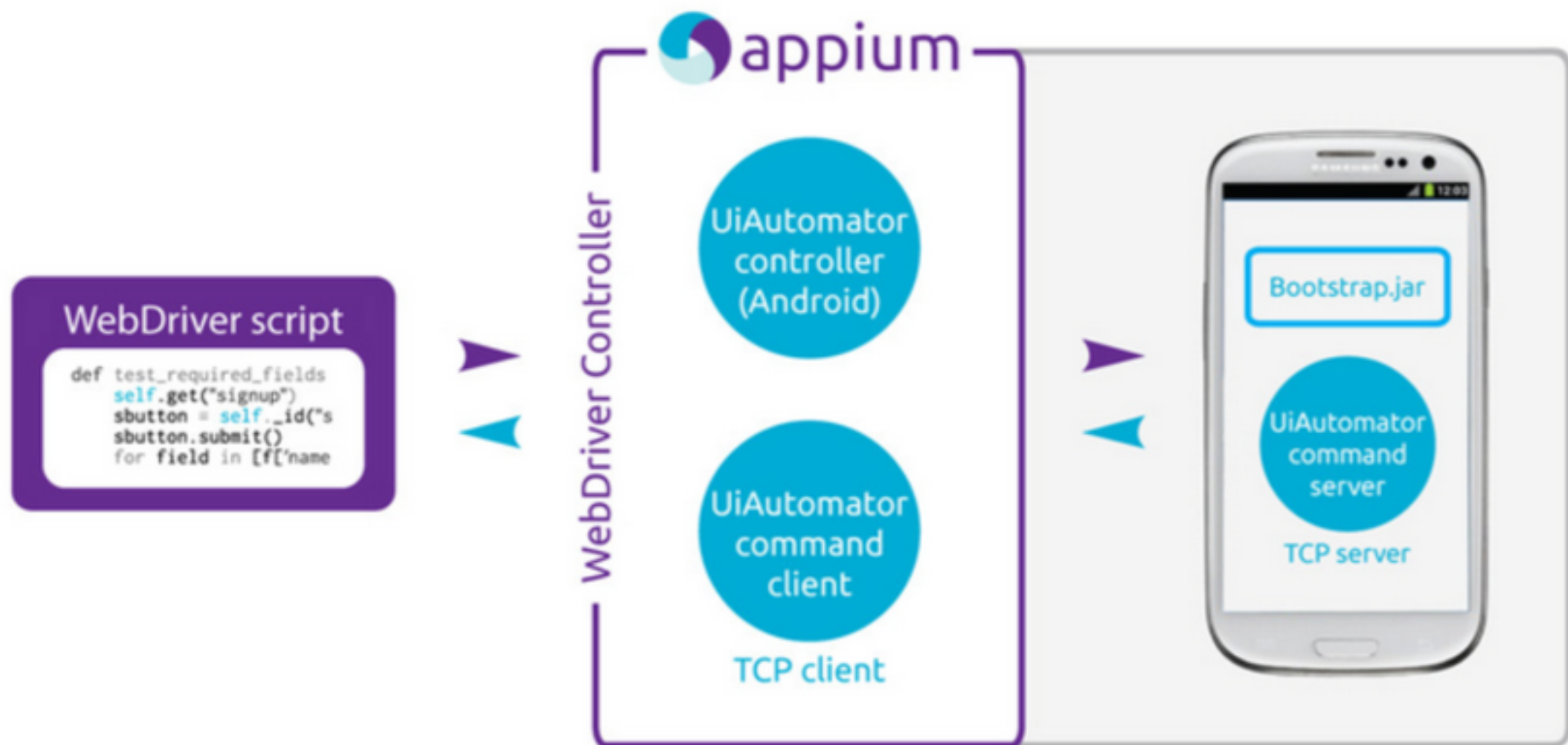
Apr 1



Hybrid应用自动化难点



新宠框架Appium



Hybrid应用自动化新宠——Appium



劣势

?

优势



Hybrid界面自动化方案



- 通过容器的指定alipays://***appId=***&url=*** 来进行指定的界面的跳转，这样避免了业务流程中的不稳定因素
- 通过截图对比的方式进行结果的验证，避免了Hybrid 中很多细节的元素抓取不到的问题
- 通过部分对比，全图对比以及部分相反对比保证了结果的准确度

自动化获取性能数据



- 获取数据有：

- 对应Case的网络消耗流量 (Anyproxy)
- 对应Case的api消耗时间 (Anyproxy)
- 应用占用的数据增长 (自己编写App监控, 缓存大小 + 代码大小 + 数据大小)
- Cpu数据、内存数据 (adb)
- 对应应用层面的流量 (自己编写的App监控)

应用大小监测核心代码



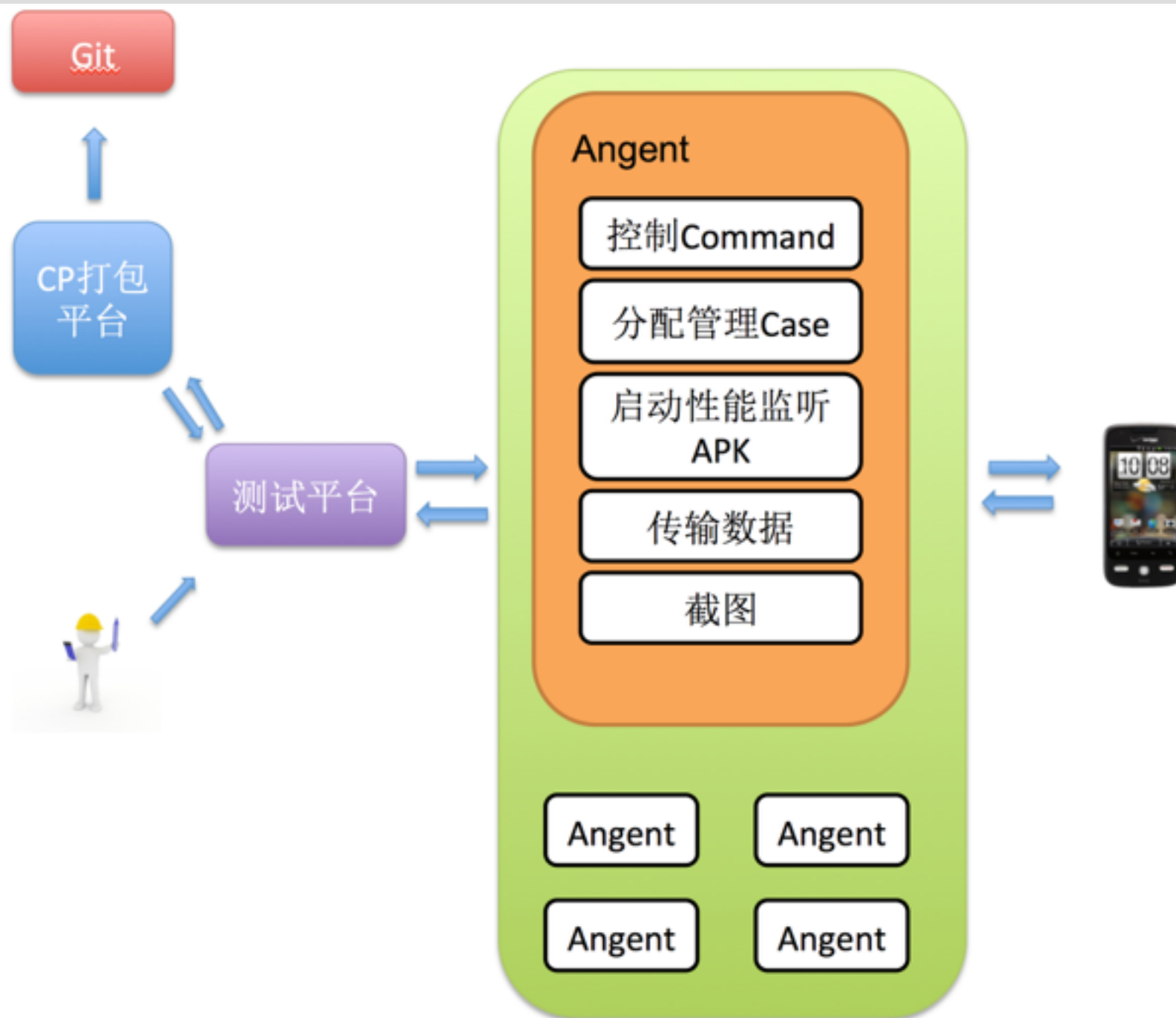
```
@SuppressWarnings("NewApi")
public void queryPackageSize(String pkgName) throws Exception {
    if (pkgName != null) {
        PackageManager pm = getPackageManager();
        try {
            Method getPackageSizeInfo = pm.getClass().getDeclaredMethod(
                "getPackageSizeInfo", String.class, int.class,
                IPackageStatsObserver.class);

            getPackageSizeInfo.invoke(pm, "com.android.chrome",
                Process.myUid() / 100000, new PkgSizeObserver());
        } catch (Exception ex) {
            Log.e(TAG, "NoSuchMethodException");
            ex.printStackTrace();
            throw ex;
        }
    }
}
```

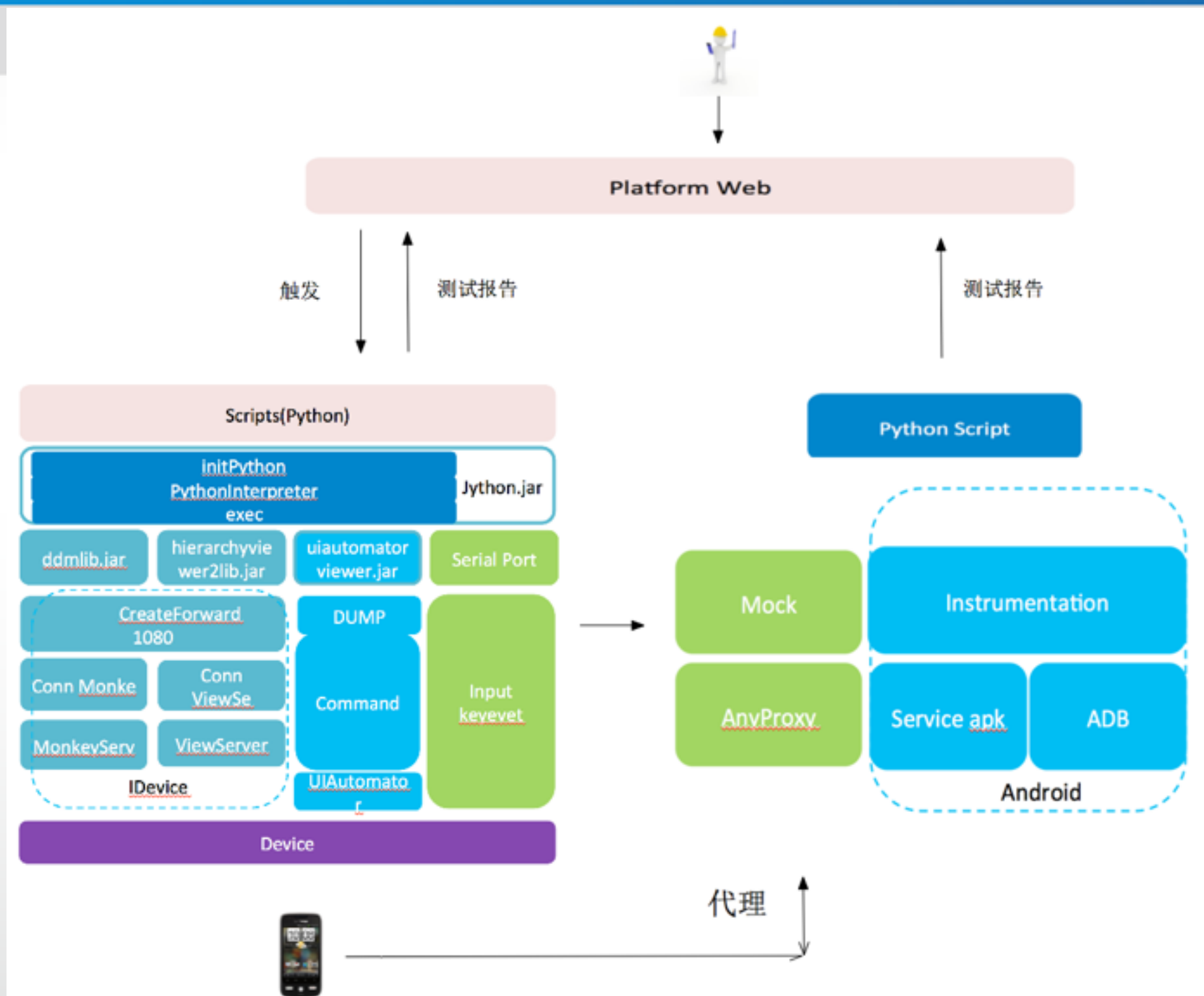

获取流量核心代码

```
public void getAppTrafficList() {
    PackageManager pm = getPackageManager();
    List<PackageInfo> pinfos = pm
        .getInstalledPackages(PackageManager.GET_UNINSTALLED_PACKAGES
            | PackageManager.GET_PERMISSIONS);
    for (PackageInfo info : pinfos) {
        String[] permissions = info.requestedPermissions;
        if (permissions != null && permissions.length > 0) {
            for (String permission : permissions) {
                if ("android.permission.INTERNET".equals(permission)) {
                    int uId = info.applicationInfo.uid;
                    long rx = TrafficStats.getUidRxBytes(uId);
                    long tx = TrafficStats.getUidTxBytes(uId);
                    if (rx < 0 || tx < 0) {
                        continue;
                    } else {
                        Log.e(info.packageName.toString() + "Traffic",
                            (rx + tx) / 1000 + "kb");
                    }
                }
            }
        }
    }
}
```

持续集成



细分框架



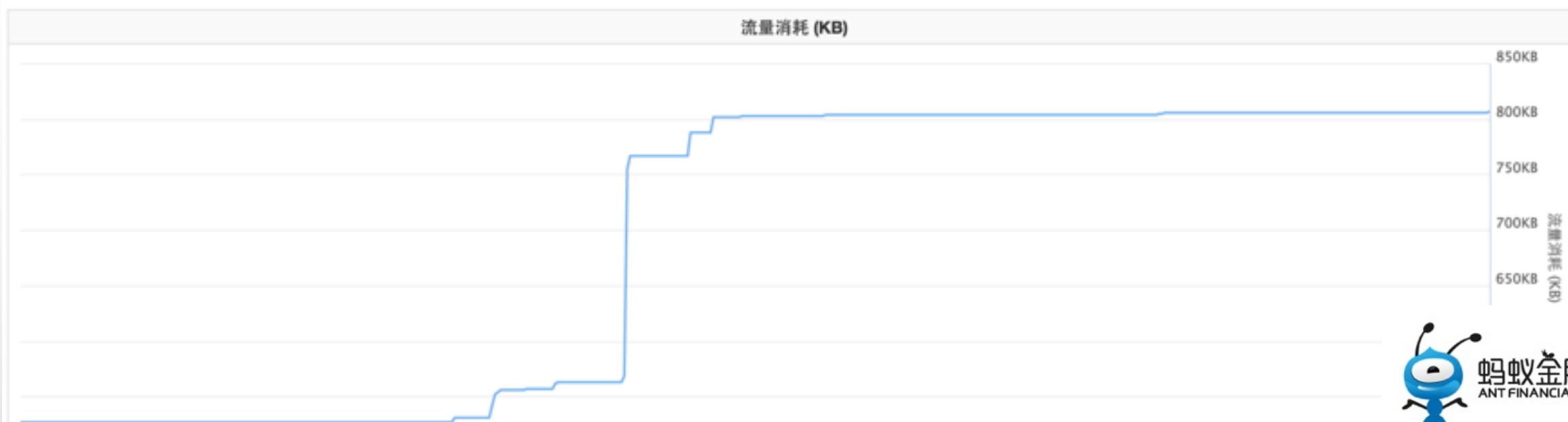
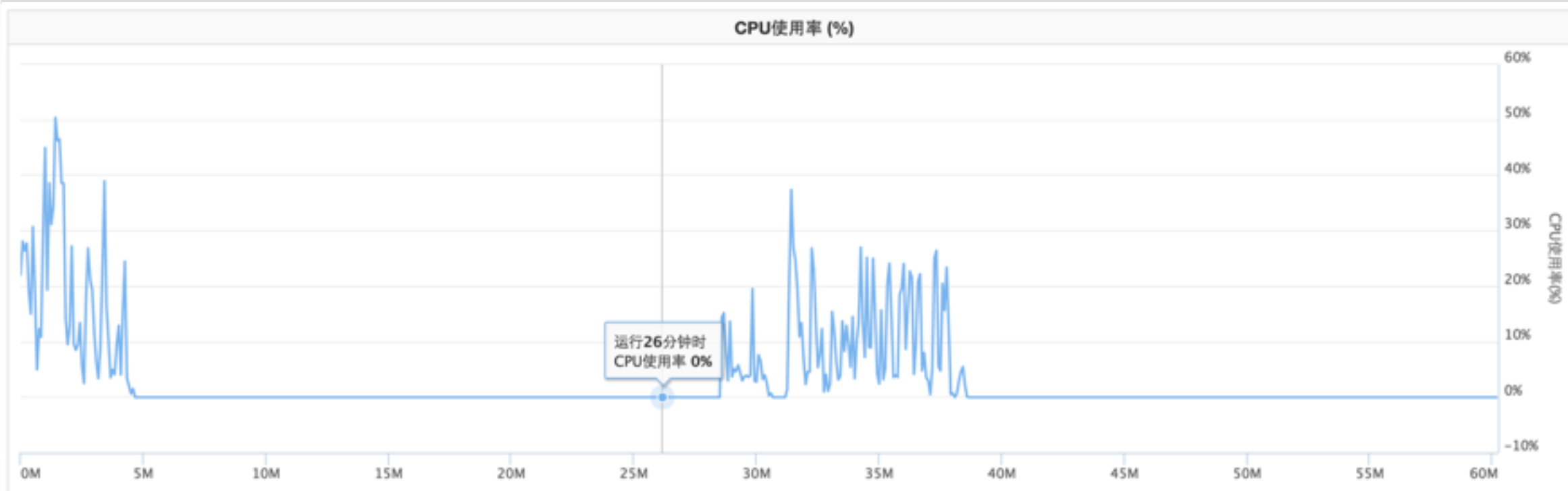
我们来看下结果



应用流量消耗 (3G)

被测界面	首次启动	非首次启动
引导界面	<input type="text"/>	<input type="text"/>
成功登录	<input type="text"/> b	<input type="text"/> b
声波支付	<input type="text"/> b	<input type="text"/> b
扫码 (多次轮询)	<input type="text"/> b	轮询的时候按照 <input type="text"/> /s 增长直到成功
进入账单详情	<input type="text"/> b	<input type="text"/> b
退款	<input type="text"/>	<input type="text"/>
加载 3 个页面的账单	<input type="text"/> b	<input type="text"/> kb
登出	<input type="text"/>	<input type="text"/>
签约	<input type="text"/> b	/
提升签约额度	<input type="text"/> kb	<input type="text"/> kb
数据罗盘	<input type="text"/> kb	<input type="text"/> kb

我们来看下结果



Demo



Problems @ Javadoc Declaration Search Console Progress LogCat Call Hierarchy

<terminated> StartPerformaceMonitor [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java

应用网络消耗流量: 6779kb

应用缓存占用量: 6458):kb应用数据占用量: 0kb应用代码占用量11kb

cpu占用率1.0%

vss内存占用率753648.0kb

rss内存占用率68952.0kb

本次功能使用了12个API 时间消耗342ms 流量消耗128kb

应用网络消耗流量: 6779kb

应用缓存占用量: 6458):kb应用数据占用量: 0kb应用代码占用量11kb

cpu占用率2.0%

vss内存占用率756328.0kb

rss内存占用率78676.0kb

关于应用专项测试详情

- 专项测试详细PPT可见: <http://pan.baidu.com/s/1kTJzAmj>

QA



- 微博: Monkey陳曄曄
- 花名: 知衡



Monkey

中国

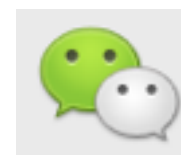


扫一扫上面的二维码图案，加我微信

蚂蚁金服
ANT FINANCIAL



@InfoQ



infoqchina

软件
正在改变世界!

个人介绍



- 移动互联网测试布道者
- 在各个企业以及高校推广正确的测试理念
- 任蚂蚁金服上海无线测试开发
- 主要负责无线技术在业务上的落地以及负责支付宝钱包非功能性测试
- 著有《大话移动应用测试》（清华大学出版社）