**一、什么是自动化测试**

自动化测试是把以人为驱动的测试行为改成机器执行，通过精心设计的测试用例，由机器按照测试用例的执行步骤对应用进行自动操作，然后输出结果，由测试人员进行比较。自动化测试可以极大的节省人力、时间和硬件资源，提高测试效率。

大白话就是: 写好自动操作app的脚本，运行起来，输出结果，结果正确，app正常测试没问题。

**二、移动自动化测试的流程**

1. 通过id、name、xpath等方法定位页面元素
2. 对定位到的页面元素执行相应的操作
3. 对操作后出现的结果和预期结果做一个比较

**三、移动自动化测试的优点**

1. 对程序员来说，在提交代码之前，可以先进行回归测试，这样能提高代码质量，加快代码上线速度。
2. 自动化测试能够很方便的自动完成安装/卸载、启动/运行、UI适配等枯燥的手工测试环节，节省很多时间，在准确编写测试脚本的前提下，能够帮助测试人员完成更多要求。
3. 脚本和框架一般在iOS和Android平台是通用的，同一测试脚本适用于所有Android机型，可快速完成兼容性测试。
4. 可以模拟产生数据。通过一些重复的操作，产生一些简单的数据。能够保存截图、输出log、输出错误信息。
5. 可以长时间运行，并保证数据的正确性，避免人工测试的人为错误。
6. Android可以通过局域网，多设备多脚本多case连续测试。
7. 可以借助透视宝对应用性能进行全面诊断。比如，用户提供安装有透视宝SDK的apk/ipa的安装包，就可以对用户的APP进行一次整体测试，产生一份包括崩溃、内存泄漏、网络监控和UI性能测试的具有洞察力的详细报告。目前云智慧的很多客户在产品发布之前都要先绑定透视宝SDK，之后在 testin、百度云测、腾讯优测等平台进行测试之后才评估是否使用。
8. 不需要重新编译APP。

**四、移动自动化测试的缺点**

1. 需要借助Appium工具来录制脚本，而Appium是第三方的开源工具，需要布置环境。
2. 可通过Xcode自带的Instruments或者Android自带的UIAutomator工具录制脚本，要会简单使用xcode和eclipse。
3. 录制成功的脚本代码需要重新编写，来输出结果、进行截图、输出错误信息等，需要会编写简单的python或者JS脚本。
4. 每次要测试一个新app的时候，都要重新设计测试用例，重新设计输出规范和记录内容，重新生成标准化的测试报告。
5. 需要对结果进行逻辑处理，需要人工比较、分析。
6. 没办法手机录脚本，需要配合客户端。
7. 框架维护不易，复用率低。

**五、移动测试自动化框架(Android)**

1. Appium (IOS/Android)
   * 开源、跨平台的自动化测试工具，适用于测试原生或者混合型的移动app，支持ios、Android和FirefoxOS平台，利用它，开发者可以利用测试代码完全访问后端api和数据库。Appium是无须SDK和编译就可以对原生应用进行测试的。
   * 可以直接在PHP 、Python、Ruby、C#、Clojure、Java、Objective-c、JavaScript、以及Perl等语言中编写测试脚本。
2. Colabash (IOS/Android)
   * 支持Cucumber，开源免费，可进行多方位测试：截屏 手势识别 实际功能代码等
3. Robolectric (Android)
   * 基本上摆脱了模拟器测试的老套路的单元测试框架，可以解压Android Sdk，还能直接对应用进行测试
4. App Grader (Android)
   * App Grader是以色列uTest的，比较少人知道，但是它能为众多的Android开发者提供非常专业的意见参考
   * 开发者可以将自己开发的应用和其他同类型应用就图形、功能及其其他地方进行比较，之后还会推出ios

5…

**六、Appium**

**1. 我选择的是Appium，为什么呢?**

1. 同时支持android、ios、混合app
2. 支持多语言，PHP 、Python、Ruby、C#、Clojure、Java、Objective-c、JavaScript、以及Perl
3. 能与大多数主流开发测试工具集成，可是使用任何测试框架驱动Appium
4. 不会要求测试人员修改或者重新编译待测应用
5. 有一个活跃社区

**2. Appium简介**

* Appium 是由Sauce Labs 提供支持的一款自动化测试框架，针对原生(native)、混血(hybrid)、和移动Web应用，使用WebDriver协议来驱动ios和Android
* Appium的设计理念是测试原生应用，不应该要求用户引入额外的SDK或者重新编译应用，另外Appium应该能与测试人员喜欢的测试实践、测试框架、测试工具一起使用。
* Appium 旨在用户可以通过任何语言以及任何测试框架去自动化测试任何移动应用，另外通过测试代码可以访问后端的API和DB

**3. Appium的要求**

针对Anroid:

Mac OSX 10.7+ 或者 Windows 7+ 或者 Linux   
Android SDK >= 16 (SDK<16 使用Selendroid 模式)

**4. Appium概念**

**1.服务端/服务器架构**

appium的核心其实是一个暴露了一系列REST API的server。

这个server的功能其实很简单：监听一个端口，然后接收由client发送来的command。翻译这些command，把这些command转成移动设备可以理解的形式发送给移动设备，然后移动设备执行完这些command后把执行结果返回给appium server，appium server再把执行结果返回给client。   
在这里client其实就是发起command的设备，一般来说就是我们代码执行的机器，执行appium测试代码的机器。狭义点理解，可以把client理解成是代码，这些代码可以是java/ruby/python/js的，只要它实现了webdriver标准协议就可以。

这样的设计思想带来了一些好处:   
1. 可以带来多语言的支持   
2. 可以把server放在任意机器上，哪怕是云服务器都可以；（是的，appium和webdriver天生适合云测试）

**2. Session**

自动化测试总是基于会话内容来实现的，客户端与服务端初始化会话方式，具体到不同的库有区别，但它们最终都会发送一个post/session 请求到服务器，携带一个名为Desired Capabilities的JSON对象，服务器将会自动化会话并响应一个Session ID用于发送后续命令

**3. Desired Capabilities**   
Desired Capabilities是一组发送到服务器的键和值(map and hash),告诉服务器测试人员期望建立的自动化会话类型，这里有各种类型的属性，可以用来改变服务器在测试期间的行为。例如，可以设置platformName属性值为Android，以便告诉Appium我们期望Android会话而不是IOS

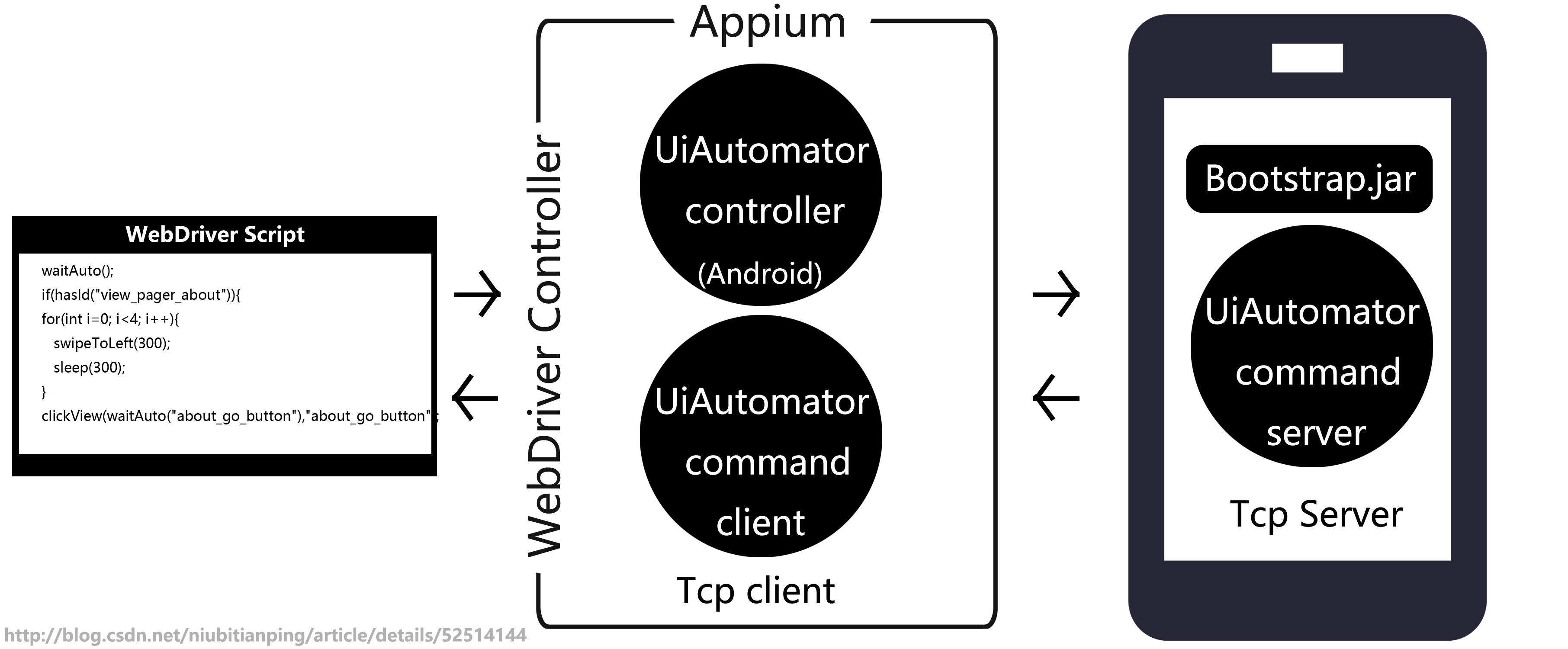
**4. Appium Server**   
Appium是用Node.js写的服务器,它可以通过源码构建和安装或者直接通过NPM安装，这里不建议用命令安装，外国网你懂的，下一篇文章教大家安装。

**5. Appium Clients**   
Appium提供了Java、Ruby、Python、PHP、JavaScript、和C#语言的客户端库，都是基于WebDriver协议扩展的，当使用Aoppium时候，测试人员可以使用他们来代替标准的WebDriver客户端，可以在

**6. Appium.app 和 Appium.exe**   
Apppium提供了Appium server的GUI封装下载，它们包含了运行Appium server所需要的都东西，所以不用担心node，它们还携带了Inspector，用于检查windows应用撑血的结构，但是Windows下用不了，我这里用 Android Device Tool。

**5. Appium的原理**

Appium基于WebDriver，利用Bootstrap.jar，最后通过UiAutomator的命令，实现app的自动化测试，如图。



图解析：

* 左边是WebDriver script 是我写的测试脚本
* 中间是Appium服务，Appium启动一个服务(默认4723)，与Selenium-WebDriver测试框架类似，Appim支持标准的WebDriver JSONWireProtocol，它提供一套Web服务，Appiumm Server接收WebDriver标准请求，解析请求内容，调用对应的框架响应操作，例如：脚本发送一个单击按钮请求给Appium Server
* Appium server 会把请求转发给中间件Bootstrap.jar，它使用java编写的，安装在手机上，Bootstrap接收Appium命令，最终通过调用UiAutomator的命令来实现。
* 执行结果由Bootstrap返回给Appium Server
* Appium还用到了Chromedriver来支持基于WebDriver的测试。

**七、什么是黑盒白盒测试**

* **黑盒测试**：已知产品的功能设计规格，可以进行测试证明每个实现了的功能是否符合要求。并没有通过检查程序的源代码或者很清楚地了解该软件的源代码程序具体是怎样设计的。测试人员通过输入他们的数据然后看输出的结果从而了解软件怎样工作。
* **白盒测试**：是通过程序的源代码进行测试而不使用用户界面。这种类型的测试需要从代码句法发现内部代码在算法，溢出，路径，条件等等中的缺点或者错误，进而加以修正。

二者最大的区别应该就是测试对象不一样，白盒测试主要针对的是程序代码逻辑，黑盒测试主要针对的是程序所展现给用户的功能，简单的说就是白盒测试后台程序，黑盒测试前台展示功能