



Appium剖析

阿里移动质量中心
字白



- Robotium
- UIAutomator
- UIAutomation
- Selenium
- WebDriver
- Selendroid



Appium特点

- 跨平台和应用类型

支持iOS、Android、FirefoxOS 三个系统平台

支持Native App、Hybird App、Web App 三种应用类型

- 支持多语言和技术框架

Python, Java, C#, Objective-C, JavaScript, PHP, Ruby, C#, Perl, ...



open source

- 非侵入式

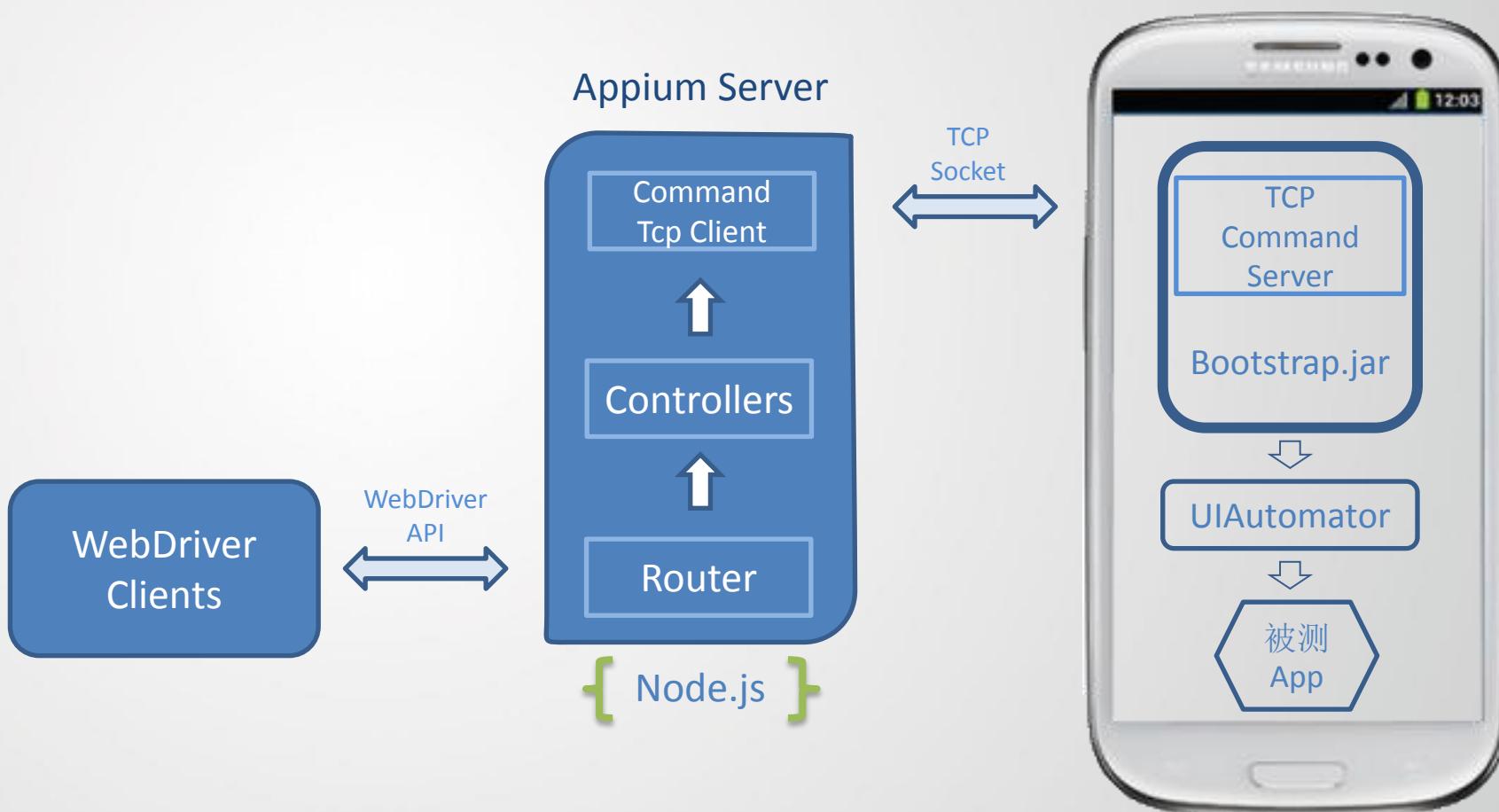
无需添加SDL，无需修改代码，无需重新编译

- 易于分布式部署

基于WebDriver的HTTP Restful API 的 C/S 结构，天生分布式



Appium UIAutomator架构图





1. Server初始化URI -> Controller的映射

```
rest.get('/wd/hub/session/:sessionId?/context', controller.getCurrentContext);  
rest.post('/wd/hub/session/:sessionId?/context', controller.setContext);
```

2. 初始化Command -> WebDriver URI的映射关系。

```
GET_CURRENT_CONTEXT -> ('GET', '/session/$sessionId/context')  
SWITCH_TO_CONTEXT -> ('POST', '/session/$sessionId/context')
```

3. Post 'desired capabilities' 对象到Server的URI: /session

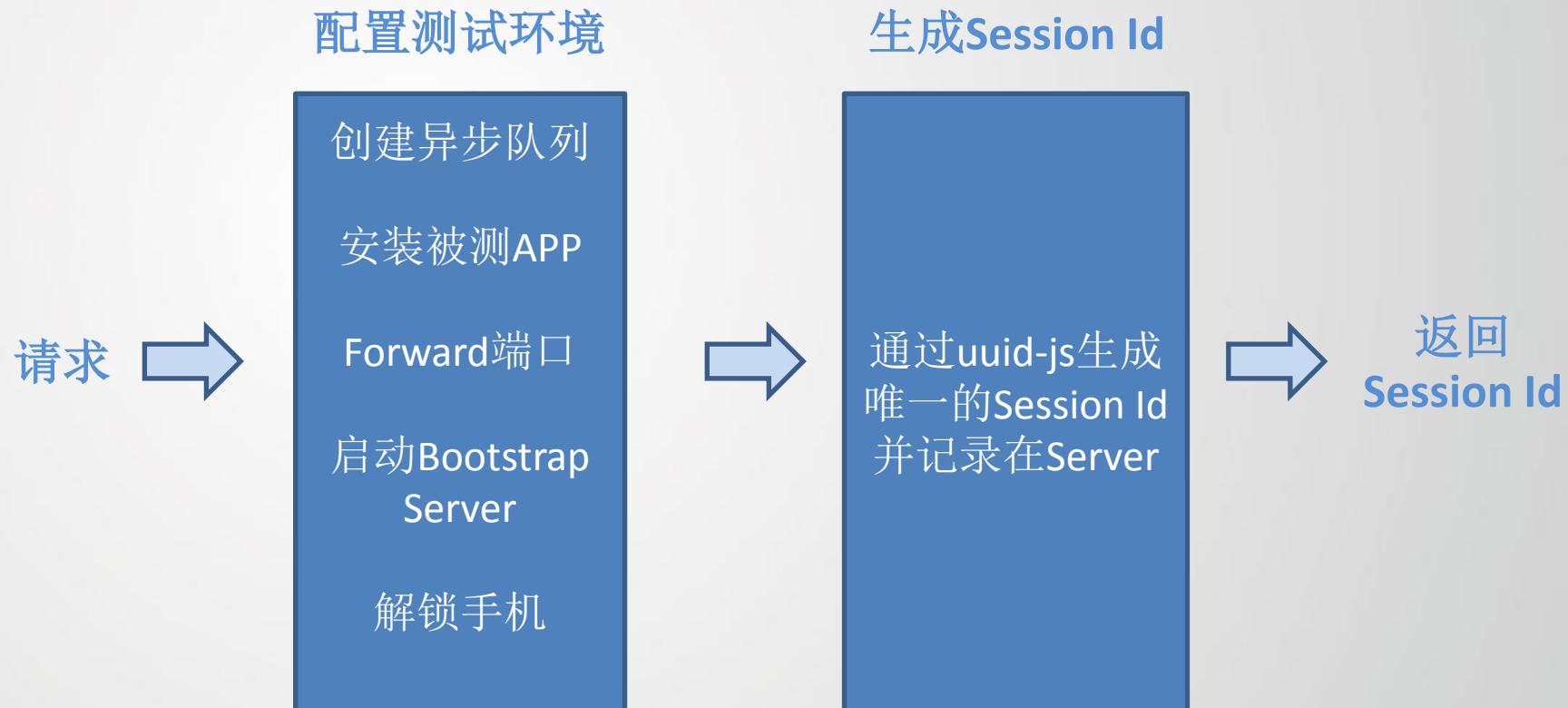


Desired capabilities 示例

```
desired_caps = {
    'platformName': 'Android',
    'platformVersion': '4.0.4',
    'deviceName': 'V889F',
    'appPackage': 'com.alibaba.mqc.mqcdemoapp',
    'appWaitPackage': 'com.alibaba.mqc.mqcdemoapp',
    'app': '/path/to/MQC.apk',
    'newCommandTimeout': 30,
    'automationName': 'Appium'
}
```



Server收到请求后，





Selenium Driver

ID = "id"

XPATH = "xpath"

LINK_TEXT = "link text" (UIAutomator Driver 不支持)

PARTIAL_LINK_TEXT = "partial link text" (UIAutomator Driver 不支持)

NAME = "name"

TAG_NAME = "tag name"

CLASS_NAME = "class name"

CSS_SELECTOR = "css selector" (UIAutomator Driver 不支持)

Appium Driver

IOS_UIAUTOMATION = '-ios uiautomation'

ANDROID_UIAUTOMATOR = '-android uiautomator'

ACCESSIBILITY_ID = 'accessibility id'



Step by step 寻找Element

Client 发送查找一个Element的请求。

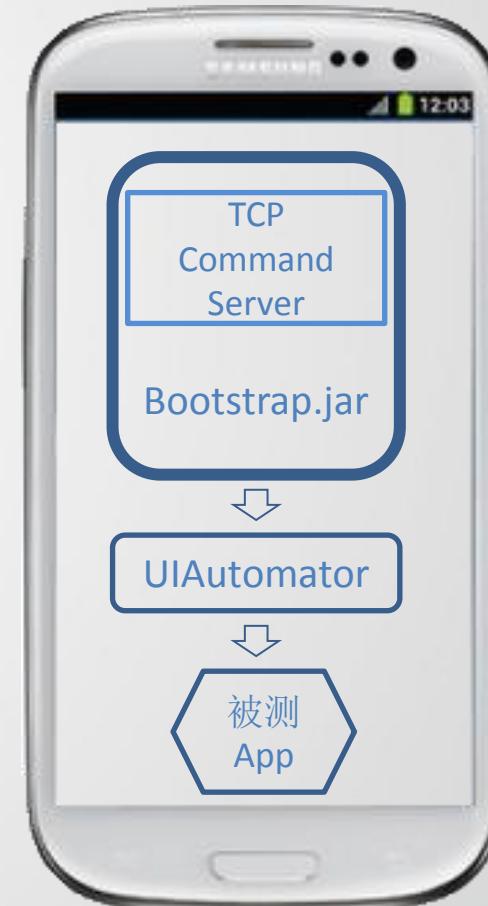
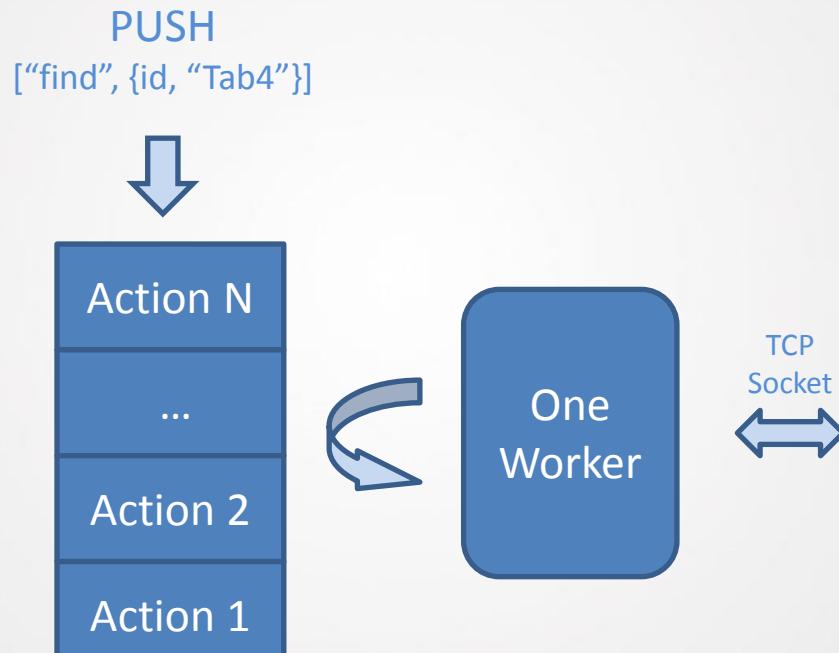
URI: /wd/hub/session/8012de97-5d25-44e0-ba1d-1dad1e89db1f/element
POST Data:

```
{  
  "using": "id",  
  "sessionId": "8012de97-5d25-44e0-ba1d-1dad1e89db1f",  
  "value": "Tab4"  
}
```



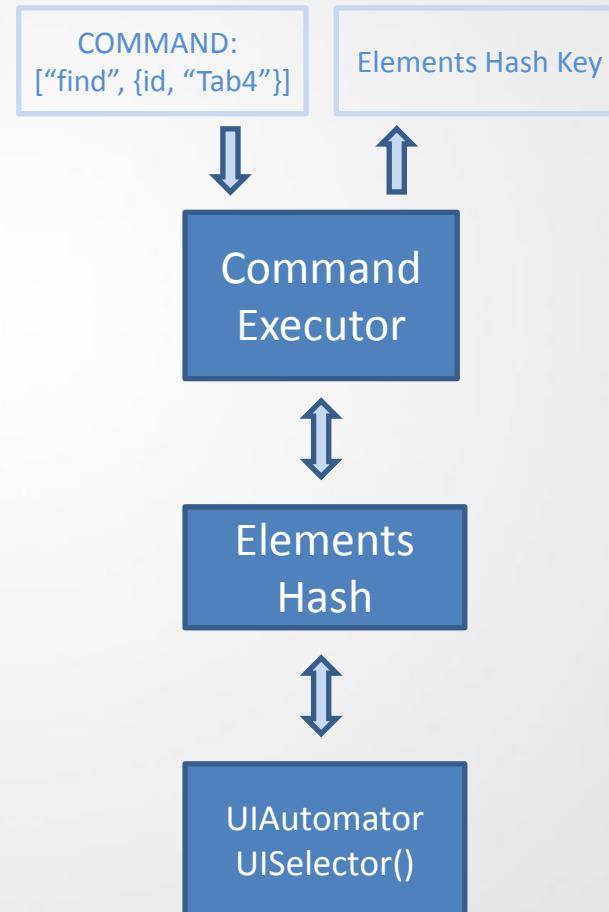
Step by step 寻找Element

Server 收到请求后，经过router：





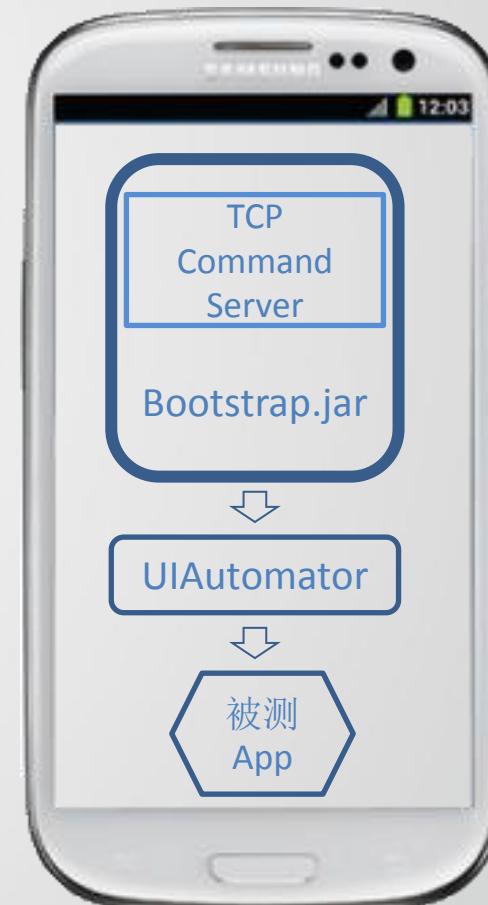
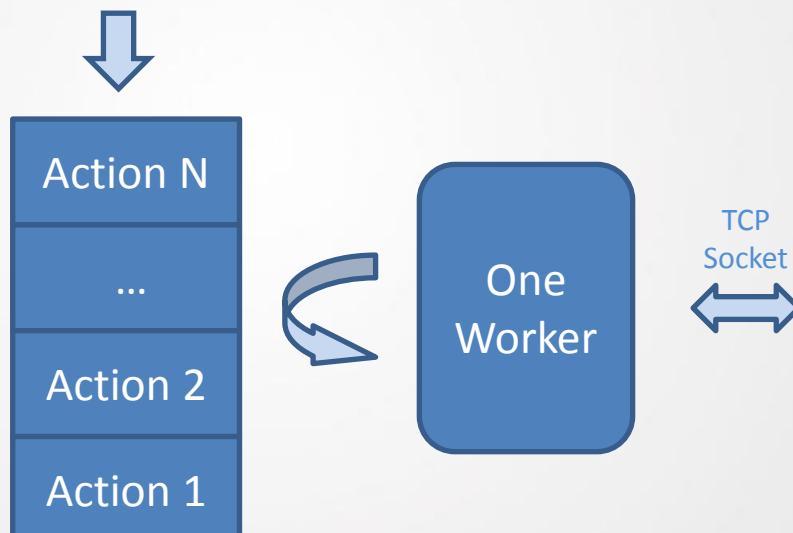
Bootstrap.jar怎么工作的？





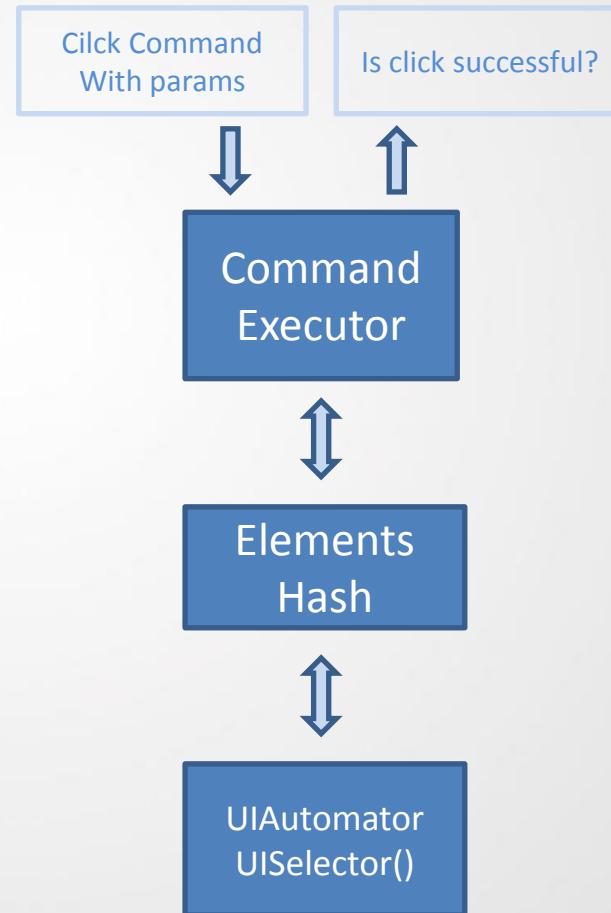
Step by step Click it

PUSH
["element:click", {elementId:
elementId}]



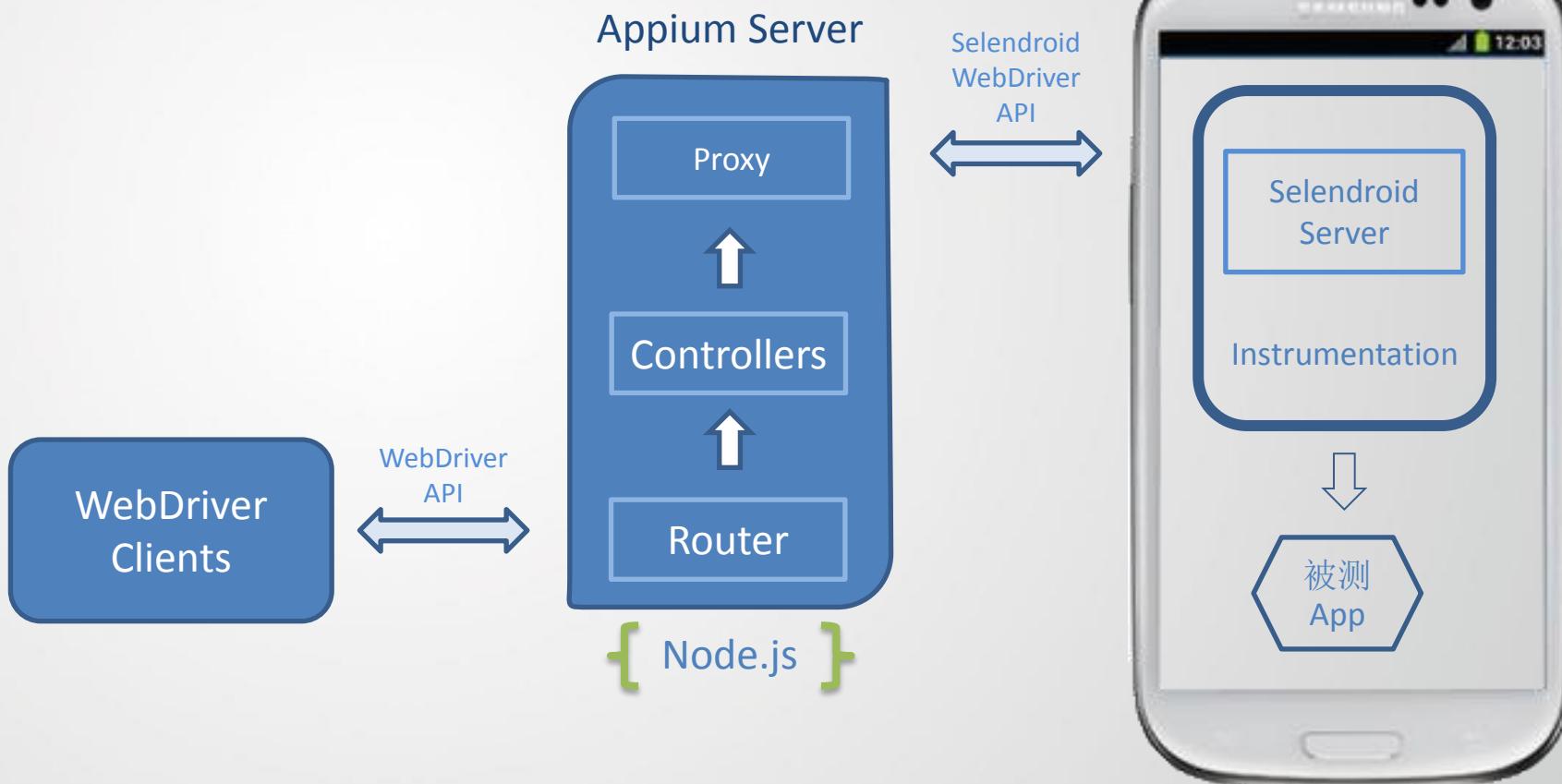


Bootstrap.jar 工作流



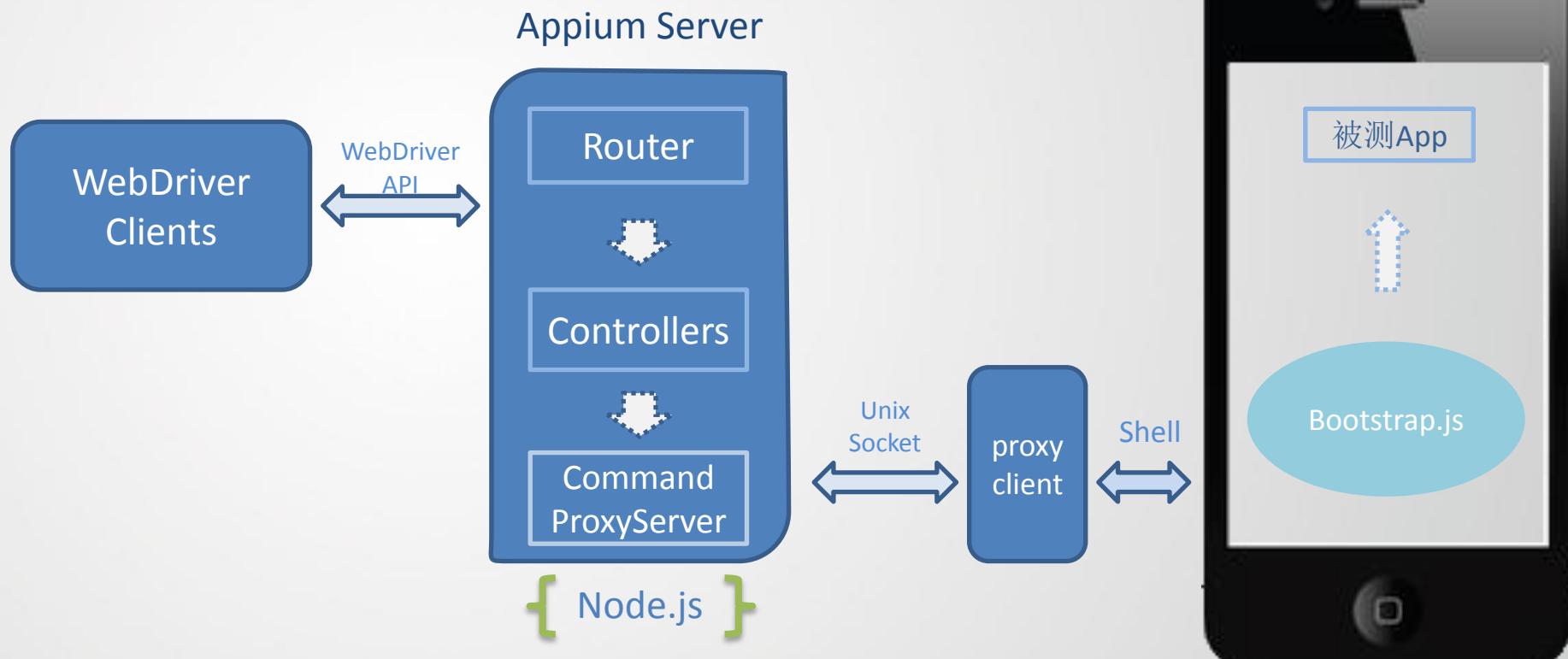


Appium Selendroid 架构图





Appium的IOS架构图



performTaskWithPathArgumentsTimeout



1. 使用 WebDriver 封装App操作接口，通过HTTP API使多语言的客户端支持成为可能。
2. 异步队列。根据业务需求，将大量操作请求，通过队列的形式逐步分发执行，防止压力过大。
3. 在不同的设备上进行交互式的执行的方法。
4. unlock_apk、io.appium.settings、
io.appium.android.ime



1. 多条件组合查找

```
def stable_find_element( resId, name, className ):
    element = None
    if resId is not None:
        try:
            element = driver.find_element(by=By.ID, value=resId)
        except:
            pass

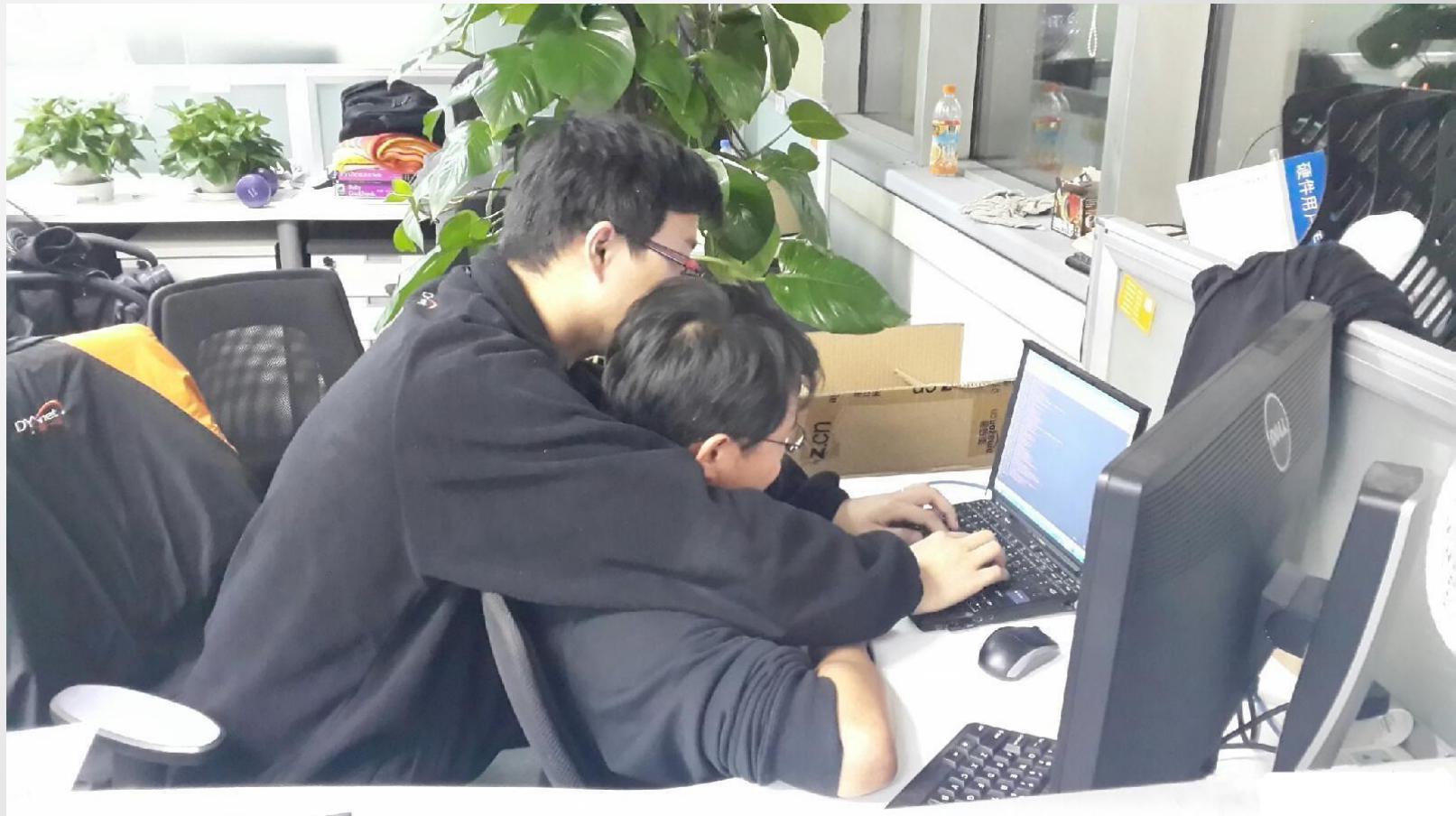
    if element is None and className is not None:
        try:
            element = driver.find_element(by=By.CLASS_NAME, value=className)
        except:
            pass

    if element is None and name is not None:
        try:
            element = driver.find_element(by=By.NAME, value=name)
        except:
            pass

    return element
```



2. 把开发同学变成好朋友





Appium 存在的问题

- 稳定性稍显不足

appium 脚本在运行时不十分稳定。

- 代码不够清晰、简洁

工程模块非常多，但是组织不够清晰，代码编写不够简洁。

- 文档欠缺

帮助文档不健全，文档查找麻烦。

- 多机支持不够简单

one server To one client 架构导致多设备支持略麻烦。



- 代码重构
- Apple Watch 和 iOS 8.4 支持
- 集成DroidDriver，弃用Selendroid和UIAutomator
- 桌面版Appium更新
- 3D游戏支持



MQC可以提供的支持

- 提供Appium的文档。

Appium介绍、如何搭建Appium、怎么编写Appium脚本、怎么开始测试

- 扩展Appium的功能

目前支持自定义截图、多账号登录等功能。支持用户其他的需求，并提高Appium的稳定性。

- 解答问题和疑惑

如果有Appium的问题，可以在我们的答疑群里提问。



MQC简介

