

互联网自动化测试云平台构建经验分享

王东峰

2015/05

目录

01

构建自动化测试平台

02

自动化测试远程执行

03

浏览器兼容性测试

04

统一测试管理平台

05

测试桩的构建与使用

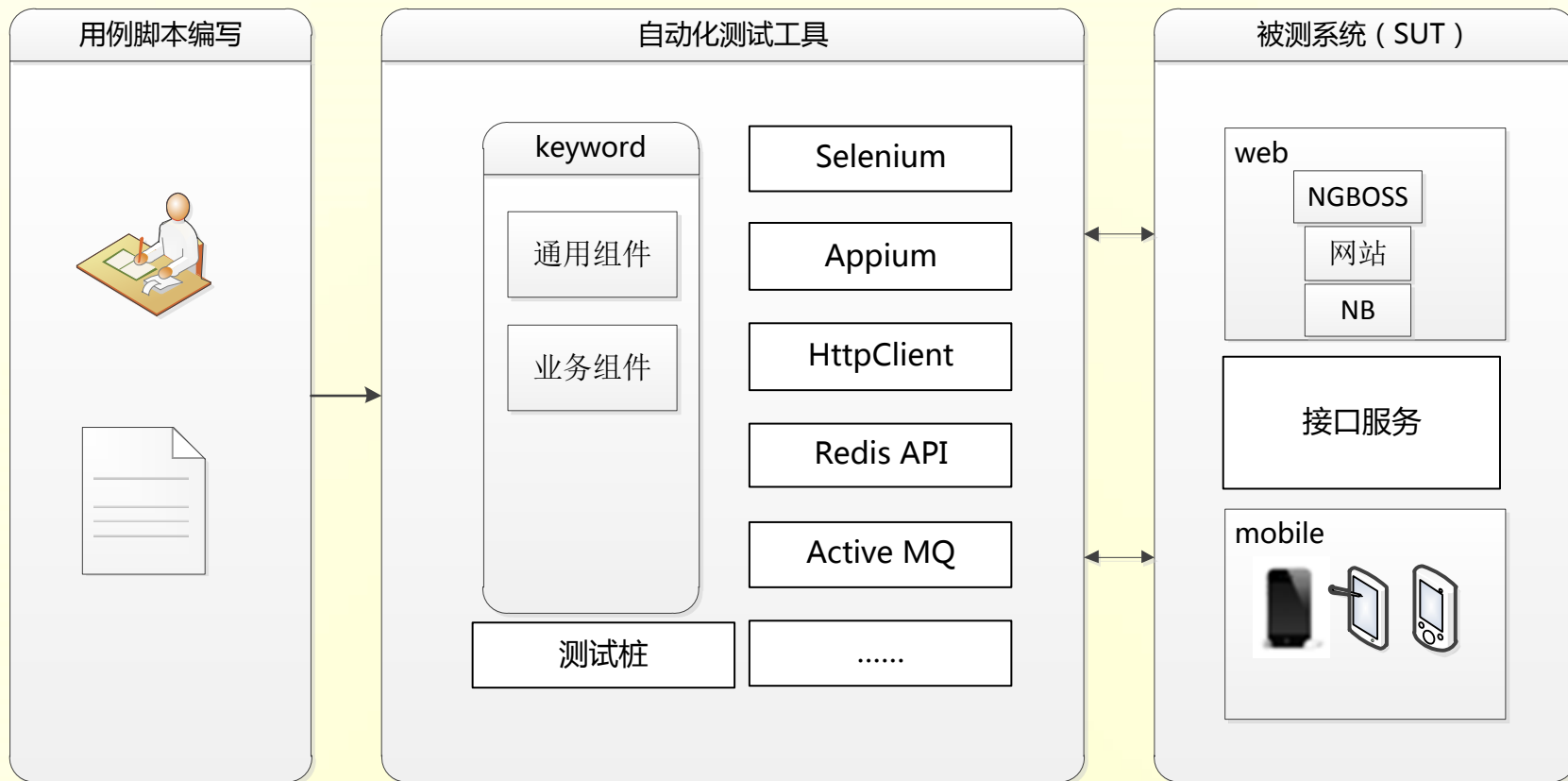
06

移动自动化测试

07

自动化项目实践

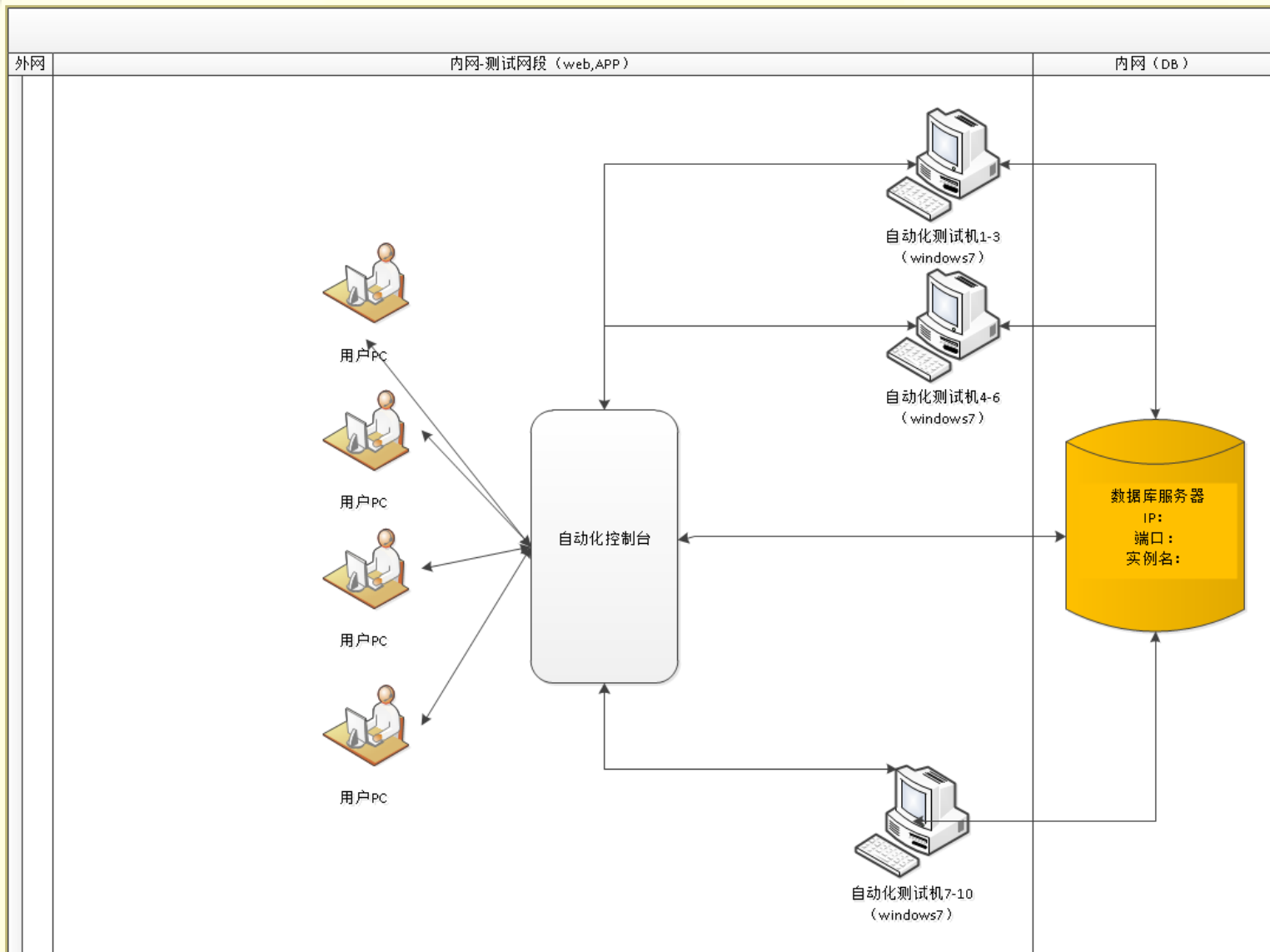
01 构建自动化测试平台



01 构建自动化测试平台

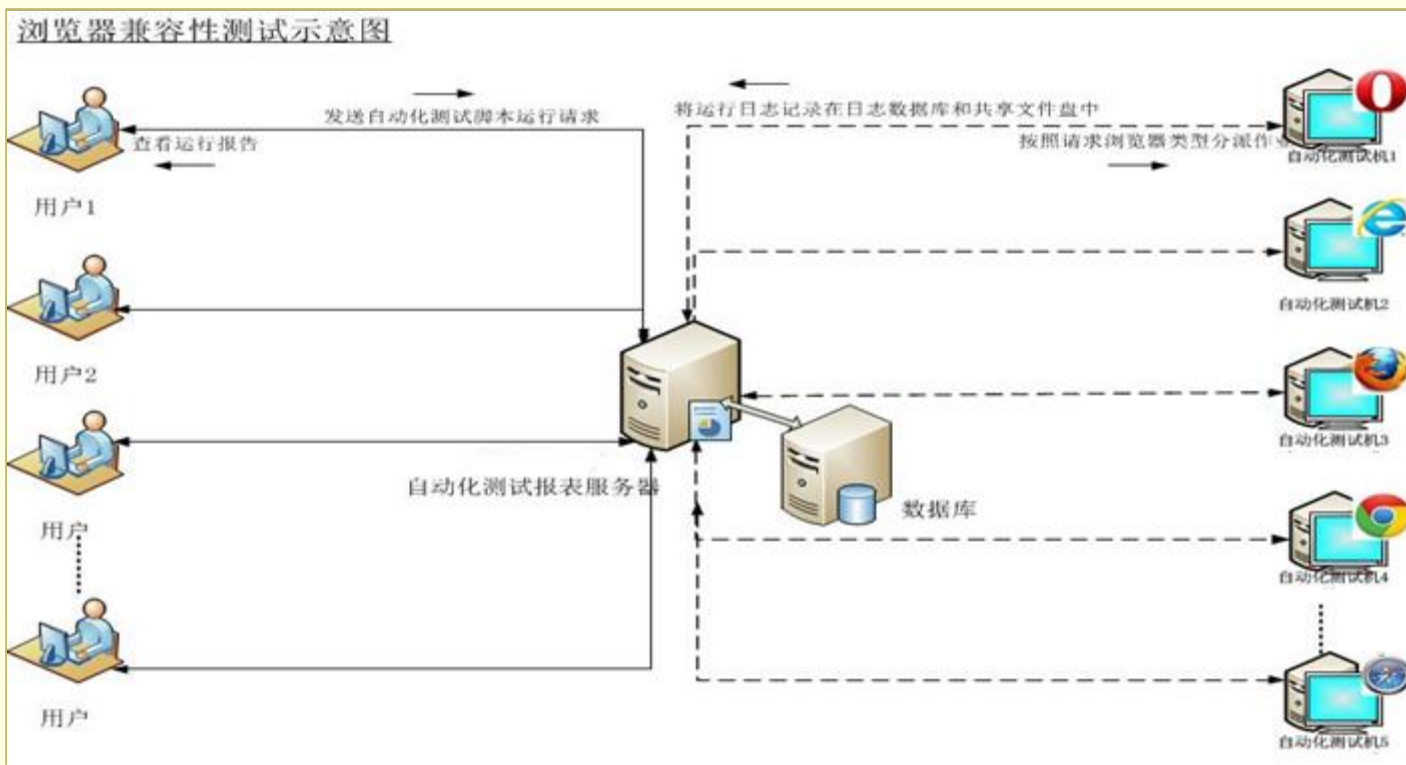
- 关键字驱动(Keyword)
- 实现Keyword共享
- 日志收集
- WEB页面测试基于WebDriver方式实现
- 支持对LINUX的操作，
- 多种数据库，redis等中间件的访问等
- 提供远程批量执行自动化用例、定时任务、邮件通知等功能
- Selenium、Appium、HttpClient等开源的测试包和框架

02 自动化测试远程执行

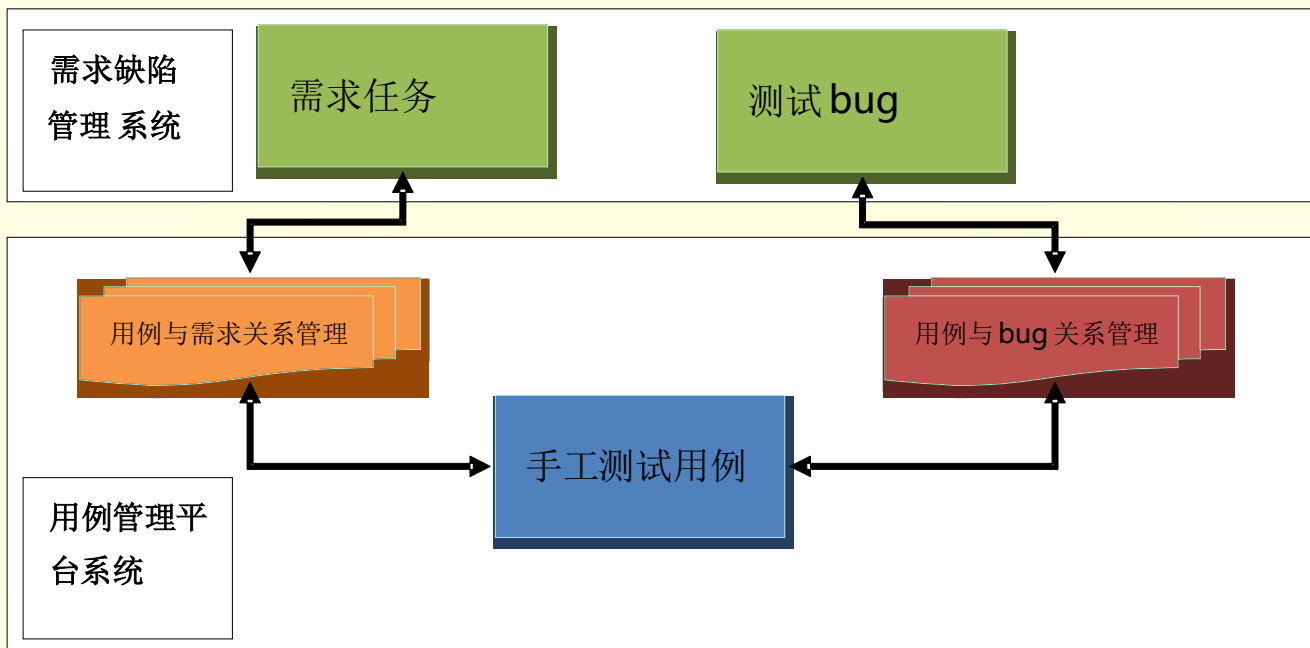


03 浏览器兼容性测试

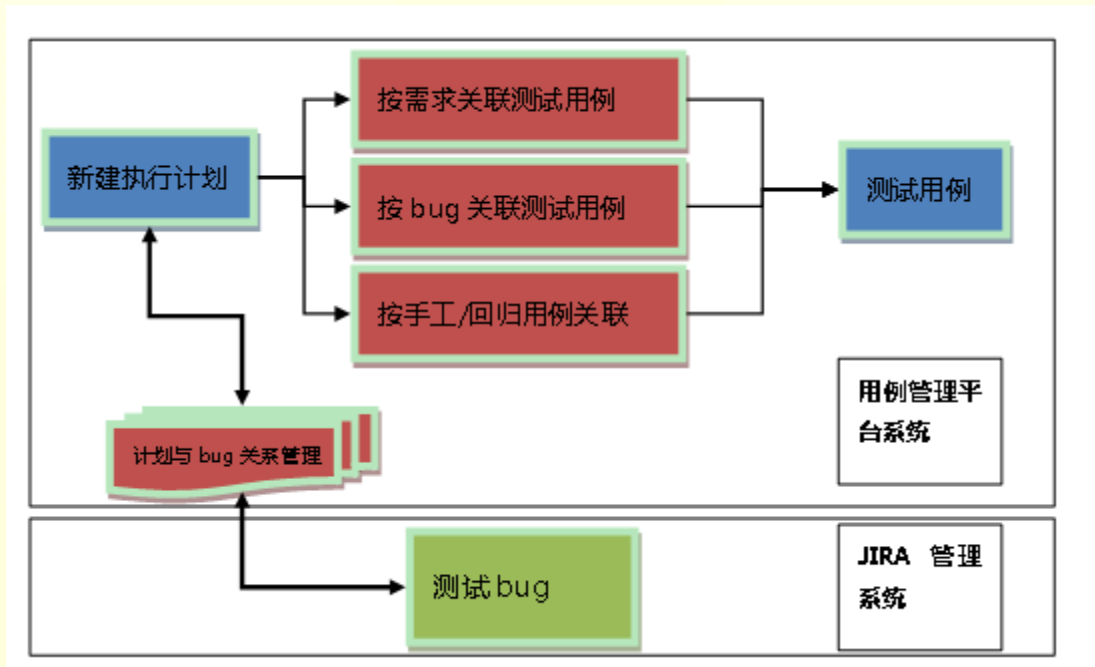
集群部署-支持浏览器兼容性测试



04 统一测试管理平台

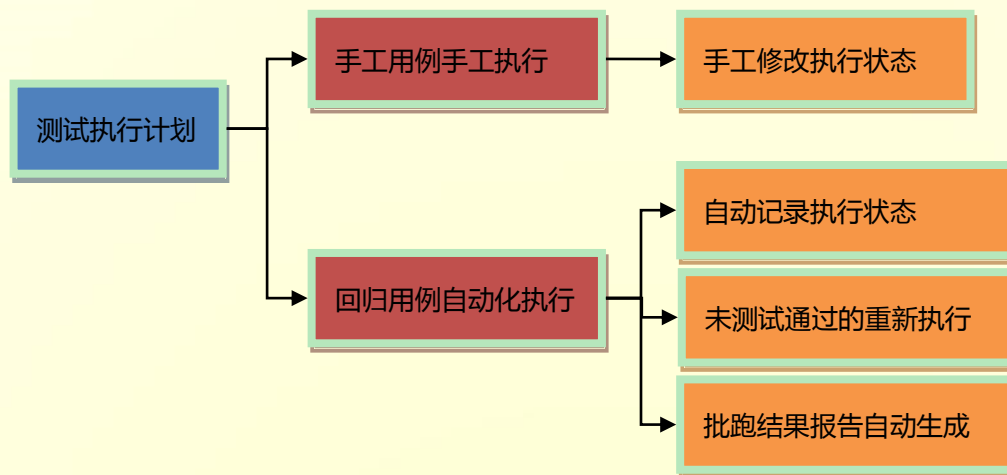


04 统一测试管理平台



04 统一测试管理平台

- 1、自动化用例支持本地执行和远程执行
- 2、手工用例直接调用自动化用例执行
- 3、与CI继承，自动构建，自动执行
- 4、使用开源工具jacoco统计测试覆盖率（手工测试+自动化测试）
- 5、分析代码覆盖率，针对未覆盖到的代码补充用例场景，提高代码覆盖率



05 测试桩的构建与使用

测试桩注册中心的由来：

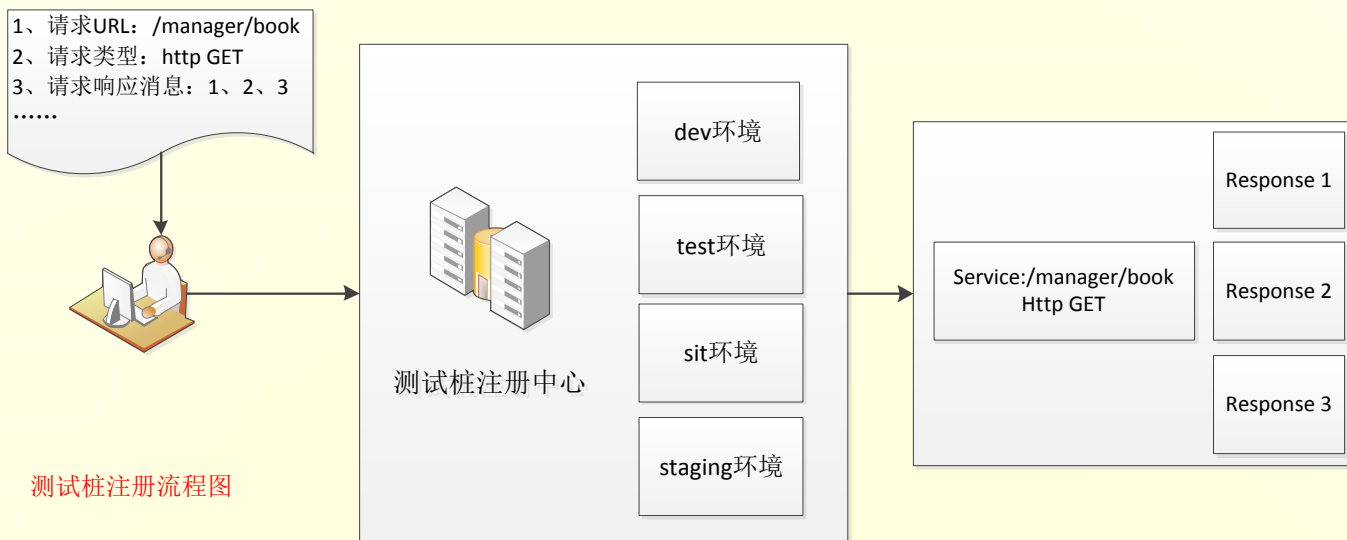
- 1、平台对接需求成为主流趋势，
- 2、互联网产品系统架构决定，跨系统的联调和测试

测试桩技术架构：

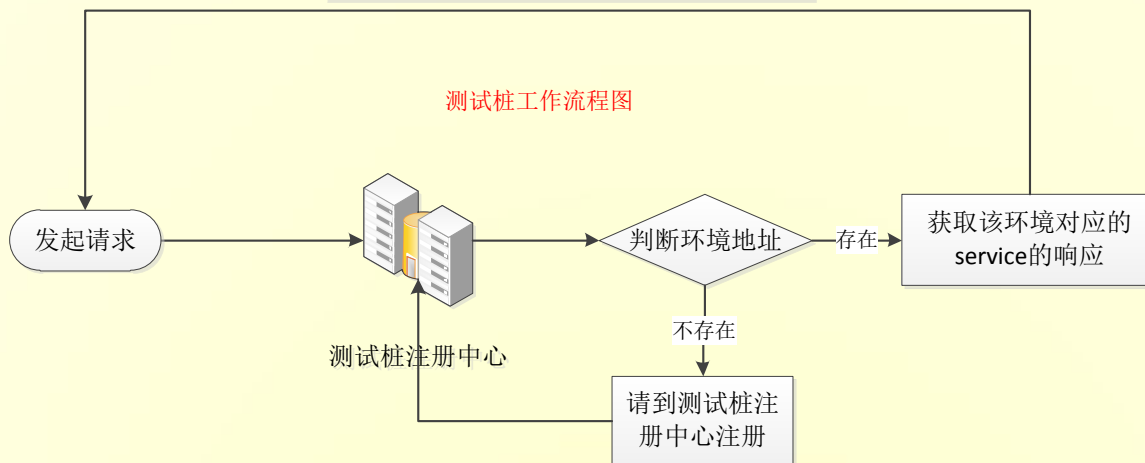
- 1、service层采用java/golang语言，根据URL动态生成桩服务，实现参数解析和传递，并根据测试环境映射形成各个测试环境对应的桩服务
- 2、注册中心层实现桩服务的统一注册，管理和http响应消息的分发

测试桩注册和工作流程图：

05 测试桩的构建与使用



测试桩注册流程图



TAP移动终端应用-技术和架构

· Appium设计理念

- 使用webdriver协议与用户交互
- Xpath抽象定位
- 底层调用各自平台的自动化技术，类selenium的架构

* Appium架构

- TAP在4.1以上使用uiautomator,4.1以下使用selendroid
- Android上使用instrumentation和uiautomator两套技术
- ios使用uiautomation
- 同时还支持firefoxOS，并可扩展其他平台

TAP移动终端应用-技术和架构

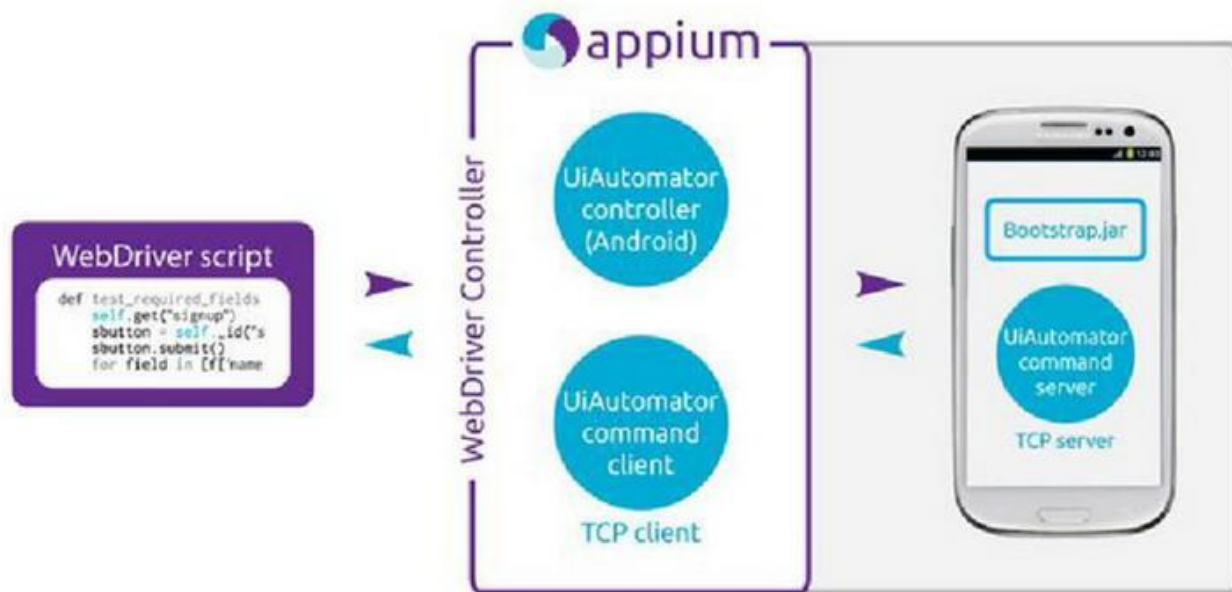
* 流程举例

- Webdriver脚本执行element.send_keys
- 发送到remote webdriver

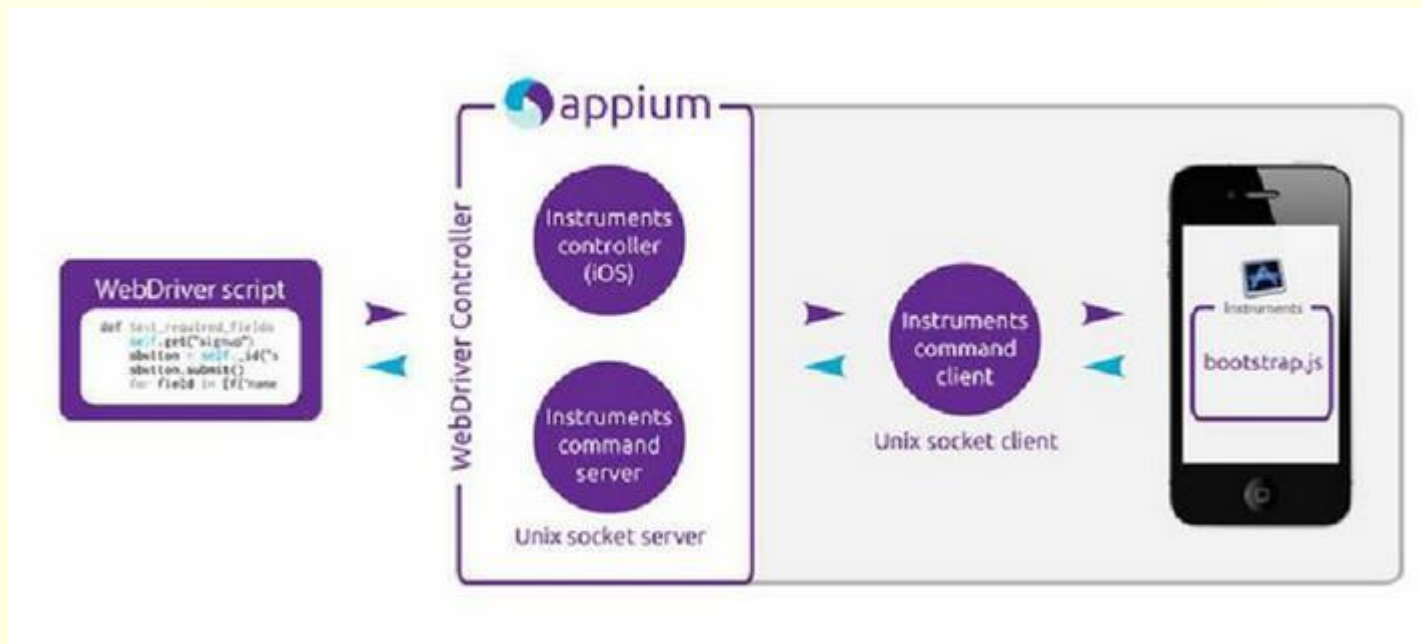
/wd/hub/session/xxx/keys

- pc处理请求并转发给android和iphone
- Bootstrap.jar调用uiautomator
- Uiautomator调用setText方法

TAP在android上的架构



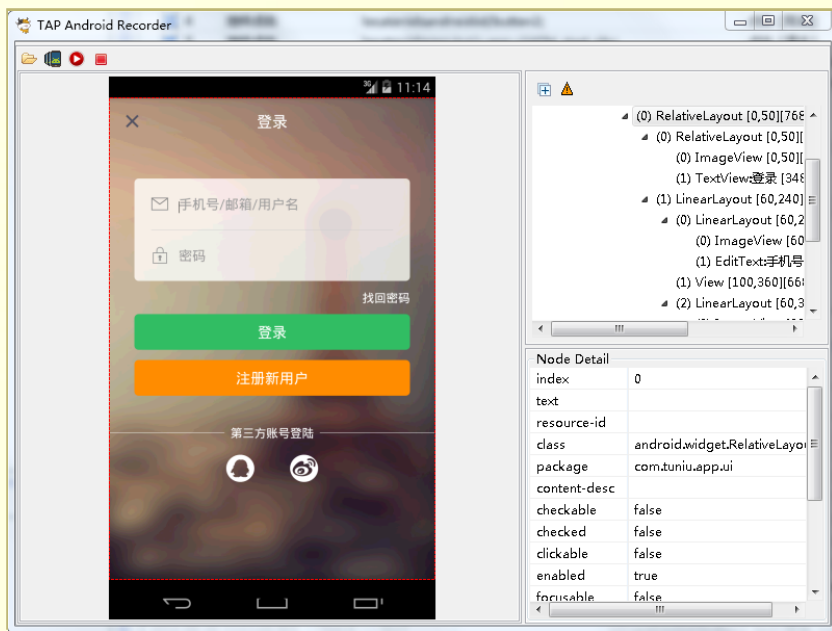
TAP在IOS上的架构



TAP移动终端应用-移动应用

* 录制脚本

- 通过Appium实现PC端与移动终端通信，完成app安装
- 启动录制界面，实现PC与移动终端屏幕同步



TAP移动终端应用-移动应用

- 完成一个步骤的录制后的移动终端界面：



07 自动化项目实践

