

# Appium 错误排查与日志分析



扫码试看/订阅

《移动端自动化测试实战》视频课程

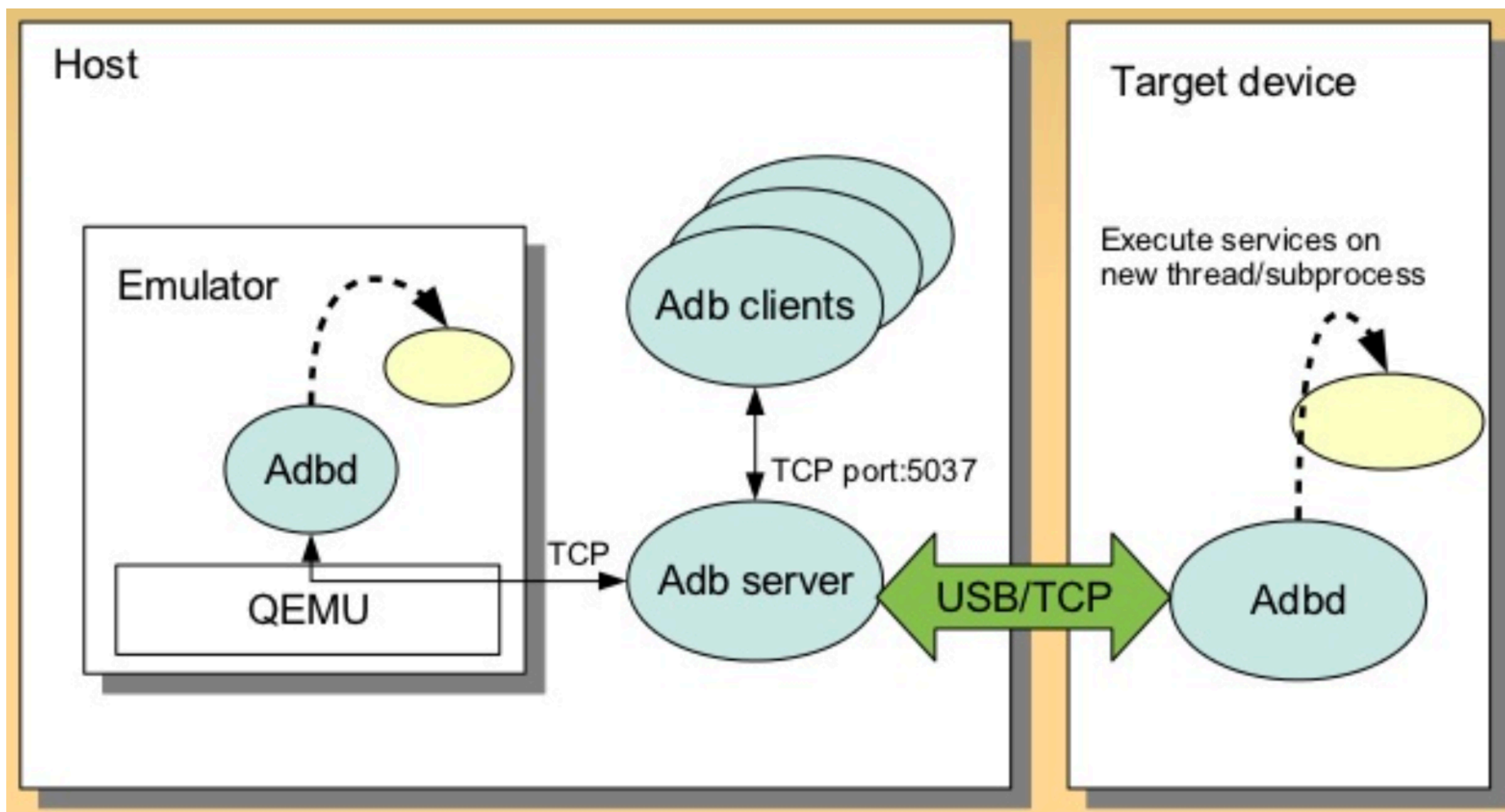
# Appium Android 自动化流程分析

- `appium -g <log file path>`
- shell 命令
- WebDriver 请求
- ChromeDriver 日志

# 调试分析方法

- Appium Log
  - 清晰记录了所有的请求和结果以及底层的一些错误堆栈
- 分析界面数据
  - 使用 getPageSource 获取界面的完整 DOM 结构
  - 利用 XPath 获取所有匹配的元素
- 脚本外调试
  - 借助于 curl 命令从脚本外探测界面数据

# adb 知识与排错



# adb 排错

- adb client: 普通的 adb 命令, 使用 shell 脚本代替
- adb server:
  - pc 上的一直开启的 server 进程
  - adb 命令通过 adb 协议与 adb server 通讯
  - 通过代理转发结合 capability

# Appium 原理与 JSONWP 协议分析

# WebDriver 协议与手工模拟

- WebDriver 协议基础知识要懂
- session\_id 获取
  - `session_id=`curl http://127.0.0.1:4723/wd/hub/sessions | awk -F\" '{print $8}'``
- element id 获取
  - `curl -X POST http://127.0.0.1:4723/wd/hub/session/$session_id/elements --data-binary '{"using":"xpath","value":"//*[@class=\"android.widget.Toast\"]}' -H "Content-Type: application/json; charset=UTF-8"`
- 元素属性获取
  - `curl http://127.0.0.1:4723/wd/hub/session/$session_id/element/$element_id/attribute/text`
- 元素动作
  - `curl -X POST http://127.0.0.1:4723/wd/hub/session/$session_id/element/$element_id/click`



# Appium 源码分析

# 所有源代码

- appium server
  - Node.js 系列包
  - adb shell
  - ChromeDriver
- 底层引擎
  - Uiautomator Java
  - WDA
  - selenium
- appium client: Python、Java

# Appium 二次封装

# 自定义 Appium Server

- git clone
- npm install

# 重新编译 Uiautomator

- git clone
- Gradle



扫码试看/订阅

《移动端自动化测试实战》视频课程