

Jekaterina Gulbinova
Dmitrij Munda

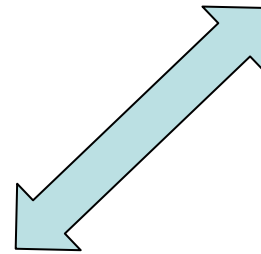


Automated testing of Mobile devices

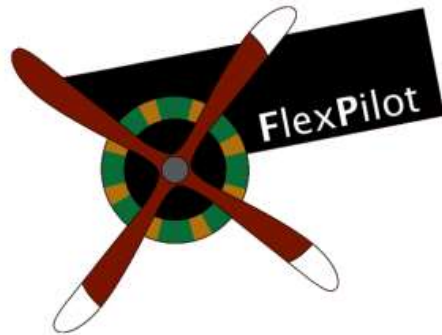
Agenda

- Introduction
- Framework adaptation
- Choice of tools for physical iPad
- Demo:
 - Desktop
 - iPad emulator

Introduction



Flash



maven



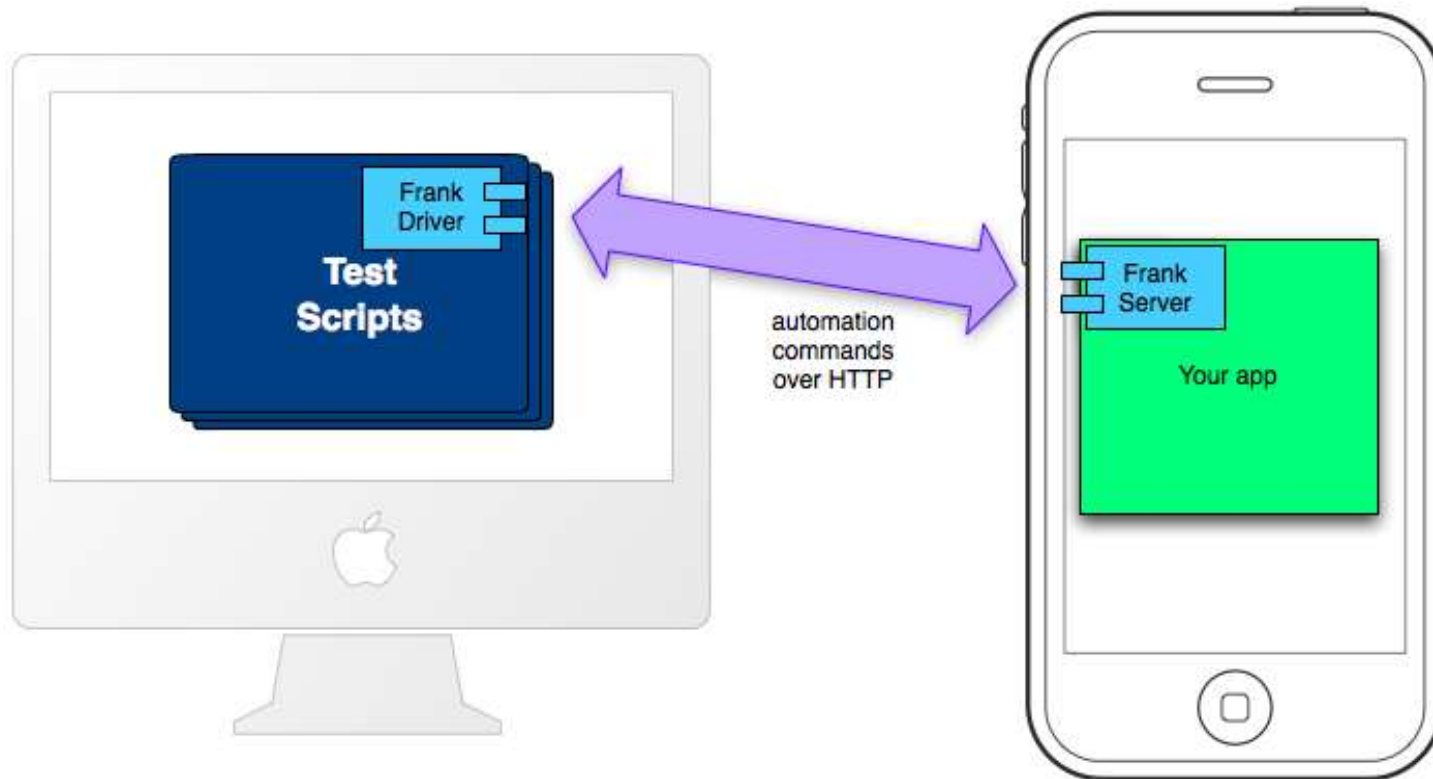
Framework adaptation



Physical iPad automation



Frank Framework



remote driver from within your automated testing framework of choice to send automation commands to the Frank server and interpret the responses sent back

small objective-C server which is compiled into the iOS application you want to automate

Other options

- **FlexMonkey** from GorillaLogics – app
- **Squish** from froglogic – app
- **Ranorex** – no Java, cannot integrate with existing tests
- **KIF** (Keep It Functional) from XING – app
- **IPhoneDriver** - can not run on real devices for mobile safari right now, MacOS only, deprecated
- **ios-driver** – worth investigating

ios-driver

- WebDriver has very clean API that can be used for ios-driver (about to become W3C standard)
- **Java**, Python, Ruby, C# clients
- For mobile web it uses the remote webkit debug protocol.
- The server is written in Java.
- **ios-driver Inspector**
- can use native events (i.e typing text using the keyboard) and web based selectors
- can use its native part to handle alerts

Example of Java code with ios-driver

```
public static void main(String[] args) throws Exception {
    DesiredCapabilities safari =
        IOSCapabilities.iphone("Safari");

    RemoteWebDriver driver = new RemoteWebDriver(new
        URL("http://localhost:4444/wd/hub"), safari);
    driver.get("http://hp.mobileweb.ebay.co.uk/home");

    WebElement search =
        driver.findElement(By.id("srchDv"));

    search.sendKeys("ipad");
    search.submit();
    waitFor(pageTitleToBe(driver, "ipod | eBay Mobile
    Web"));
    driver.quit();
}
```

- Demonstration