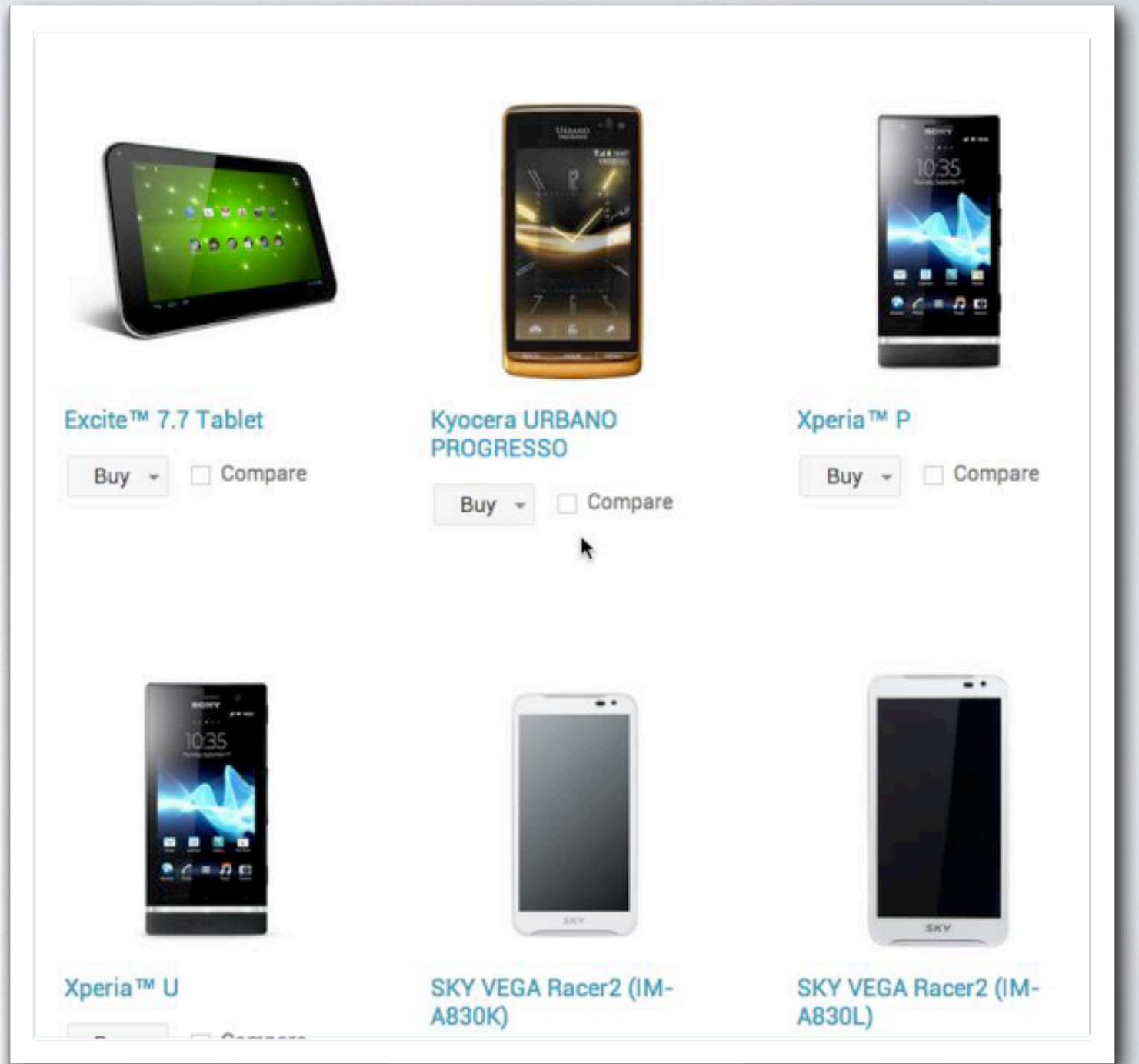
Calaba.sh

Cross-Platform,
Automated Acceptance Testing
for Mobile Apps

Mobile Challenges



Mobile Challenges

- Many devices, screens, resolutions, CPUs, OS'es,...
- Varying Conditions (network, location, rotation, settings.)

- In practice, often
  - a manual process: repetitive, expensive, low coverage
  - insufficient device converage
  - regressions occur

Automated Testing Philosophy

- We recognize value in BDD (and TDD)
- You provide business readable specifications
- Those specifications can be executed as automated acceptance tests
- Communication and shared language

Example

```
Feature: Login
  As a user I'll be able to add wordpress hosted blogs.

  Scenario: Invalid login to WordPress.com blog
    Given I am about to login
    When I enter invalid credentials
    Then I am presented with an error message

Scenario: Add a WordPress.com blog
  Given I am on the Welcome Screen
  When I add the WordPress.com blog
  Then I should be logged in
```

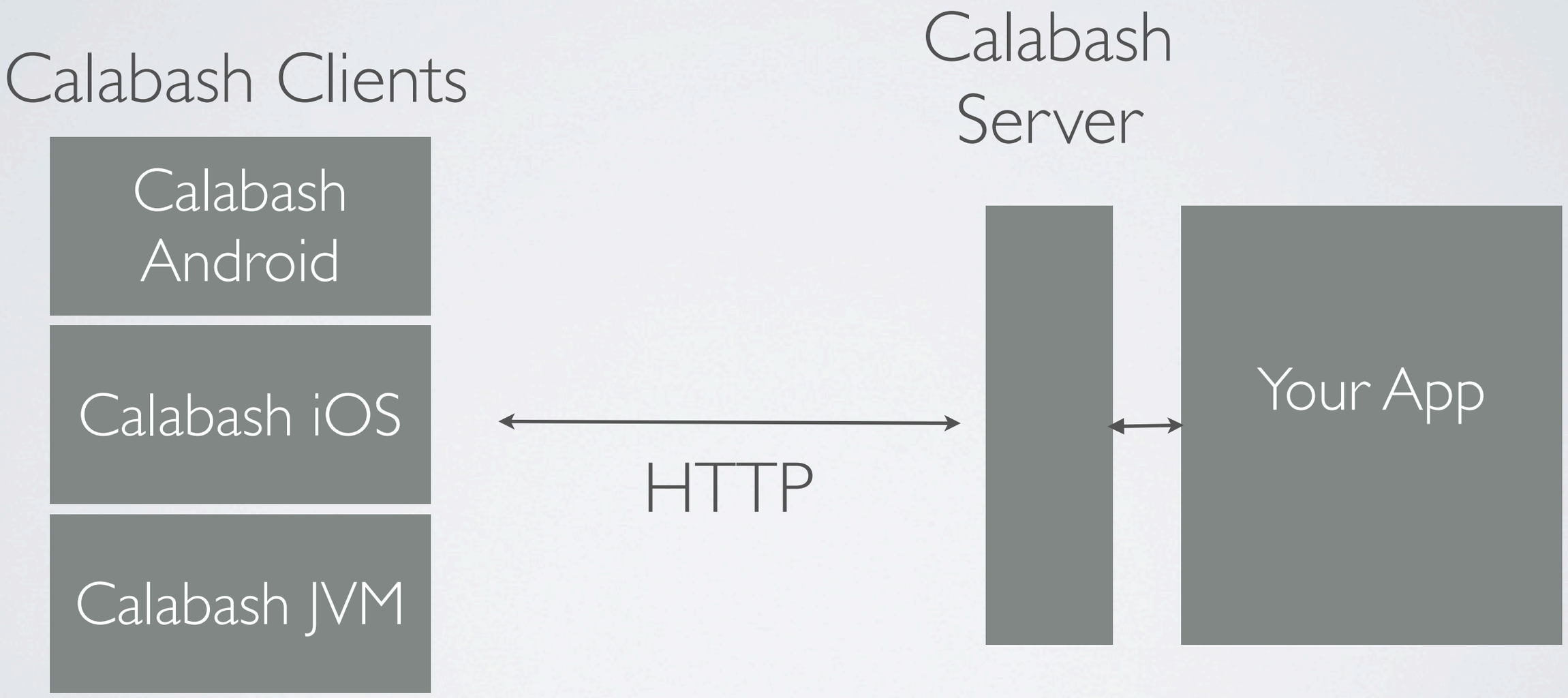
What is Calabash?

- Tool for BDD-style acceptance testing, consisting of two projects:
  - Calabash Android: https://github.com/calabash/calabash-android
  - Calabash iOS: https://github.com/calabash/calabash-ios
- Core development by LessPainful, but Open-Source and Free (Eclipse Public License)
- Written in Objective C, Java and Ruby (and Clojure)
  - Test written in Ruby but other languages are easy (e.g. Java/JVM is WIP).

Features

- Native and hybrid apps for Android and iOS
- BDD via the Cucumber tool
- Cross-platform testing: maximize sharing when developing the same (or similar) apps on multiple platforms.
- Runs on physical devices as well as simulators/emulators
- Value-added options via LessPainful (e.g. support)

Architecture



Calabash Android

- Instrumentation-based
  - but supports dynamic, interactive test-development
- Has the full power of Robotium, and adds additional methods for easing automation
- Web-support based on a shared (small) JavaScript library: calabash-js

Calabash iOS

- App is built for testing by linking a static lib (calabash-server)
  - Uses a combination of UIAutomation and other private APIs
- Supports interactive test-development
- Based on Frank, but changes some core parts
  - view selection engine (web, reflective actions, predicates, visibility)
  - touch synthesis engine allows generating of arbitrary touch gestures (extensible)
- Will be developed in coordination with Calabash Android
- Will be componentized in coordination with Frank

Example

```
Feature: Login
  As a user I'll be able to add wordpress hosted blogs.

  Scenario: Invalid login to WordPress.com blog
    Given I am about to login
    When I enter invalid credentials
    Then I am presented with an error message

Scenario: Add a WordPress.com blog
  Given I am on the Welcome Screen
  When I add the WordPress.com blog
  Then I should be logged in
```

IOS

VIDEO EXAMPLE

https://github.com/LessPainful/mobilesummit2012/raw/master/ios_example_wp_login.mov

ANDROID

VIDEO EXAMPLE

https://github.com/LessPainful/mobilesummit2012/raw/master/android_sample.mov

Future for Calabash

- Unify APIs cross Android and iOS.
- Improve hybrid support - low-level, CSS/XPath selectors, visibility $\Longrightarrow$ High-level webdriver-like API.
- Official Java clients (JVM really).
- Unification and cross-pollination with all the other cool projects.

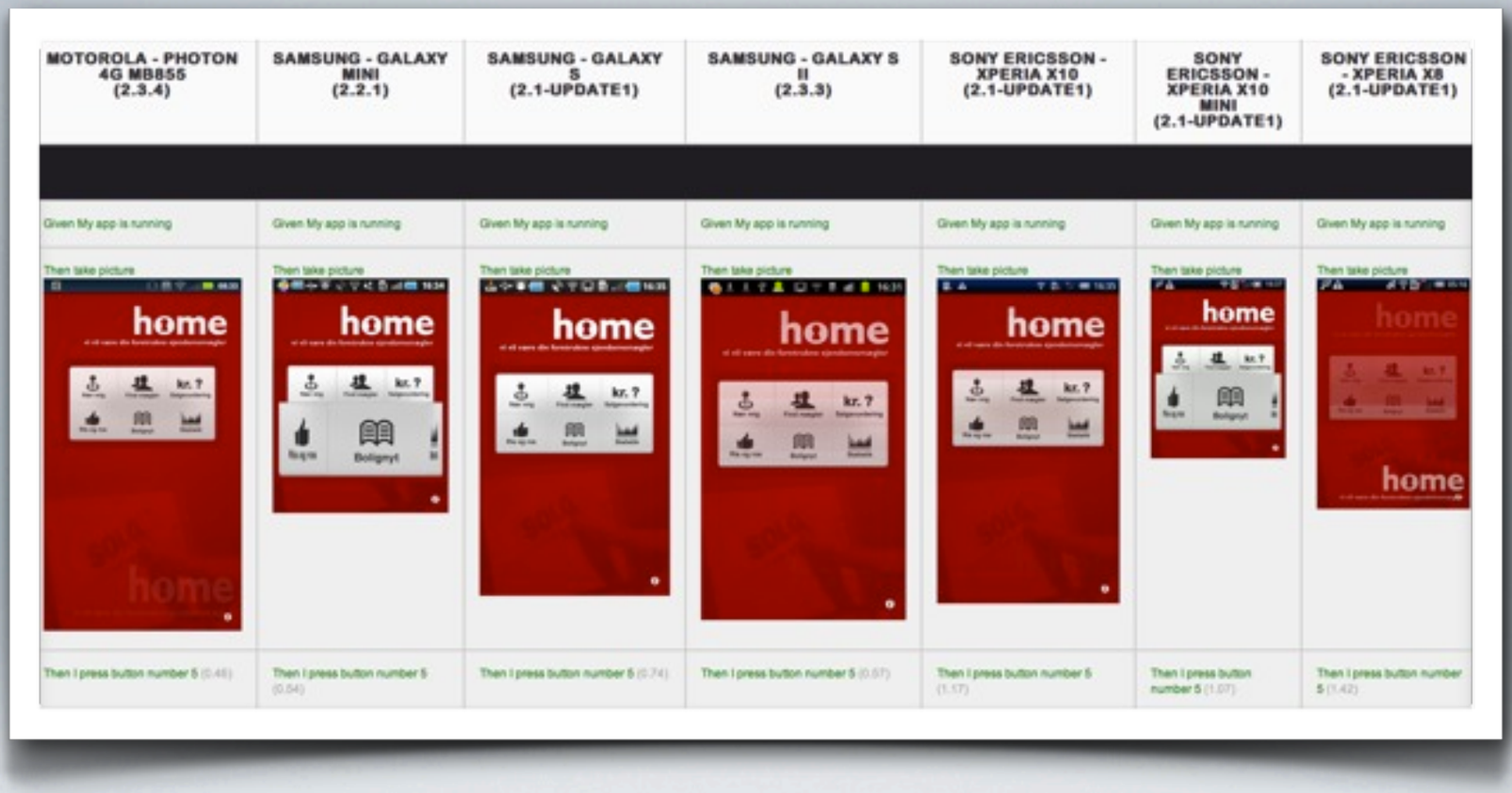
Mobile Test Lab

www.lesspainful.com

LessPainful

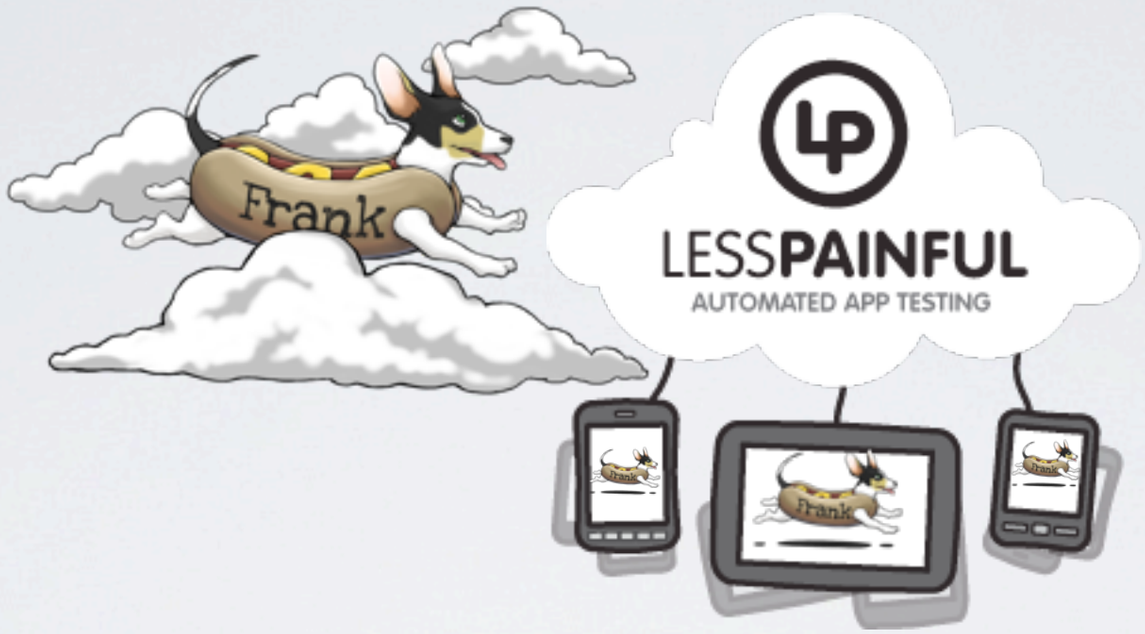
- Support plans for Calabash + Training
- Hosted test environment (shared or private)
  - Hosted authentic (non jailbroken) devices
    - tests in parallel
  - Visual comparative test reports
  - Continuous integration support
- Exposure to differnet conditions.
  - OS versions, languages, network throttling,...

Visual Test Reports



https://www.lesspainful.com/instant/result/ffxxcdtwxjjoctowngtedfamdwppezscnsuoimmi

Announcement



- As of today, LessPainful fully embraces Frank
  - Full commercial support option - same level as Calabash
  - Execution of Frank tests in our hosted environments (private beta)
- Special thanks to Pete Hodgson from Thoughtworks!

Questions?

Making App Testing
Less Painful

http://www.lesspainful.com