Calabash

Automated functional testing for your iOS applications
to avoid Breaking Bad when delivering to your customers

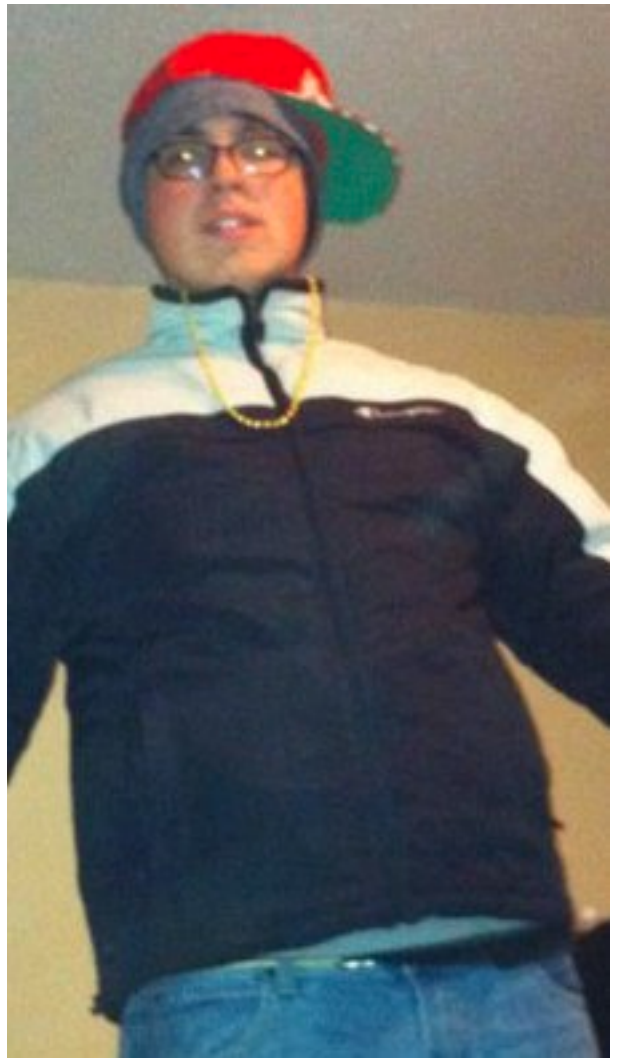
Kyle Roberts
123kyle.roberts@gmail.com
@TheKyKy123

Jeff Roberts
jeff@nimbleNogginSoftware.com
@JeffBNimble

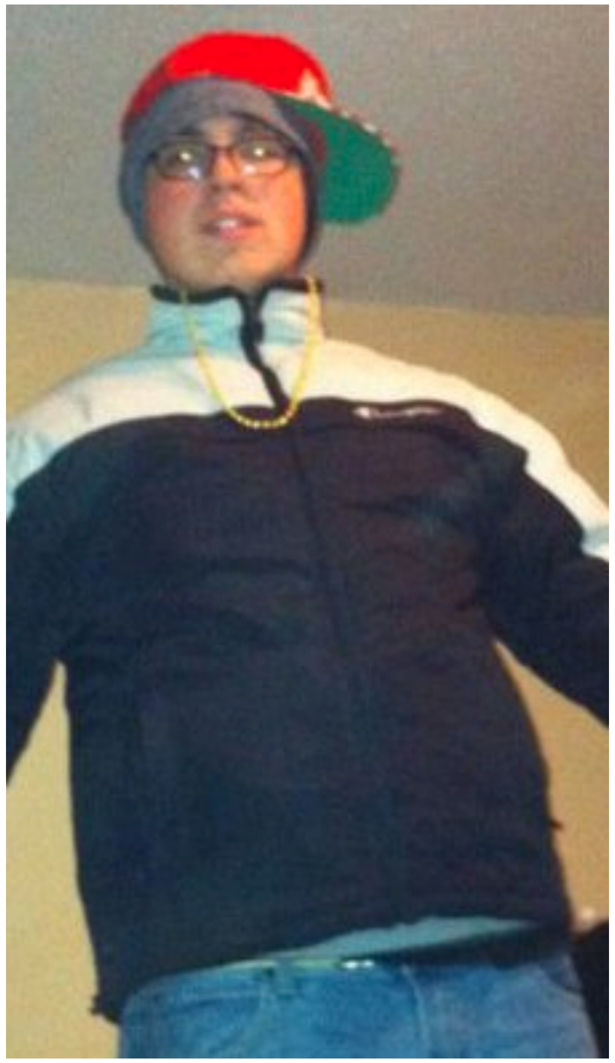
Agenda

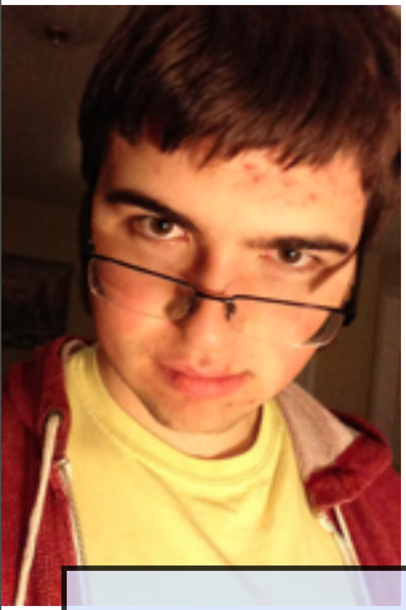
1. What is functional testing and how is it different than other types of automated testing?
2. What is Calabash and how do you integrate it into your iOS apps?
3. Writing and running tests with Calabash

“Kyle wrote an app yo!”



And he wants to test it yo!





Before he sends it to his customers, yo!

Smoke Test

Make sure:

1. App launches correctly (doesn't crash)
2. I enter the app after swiping on the Home screen
3. I can scroll the list of dead drops up and down
4. I can select any dead drop and then tap the map to open it
5. I can tap the pin and see the name of the Dead Drop
6. I can can tap on the header to return to the Dead Drop List

Ship it!

Techno Warning!!!

+

=

RegEx

(C)1998 Halibut Stuff - stuff.halibut.com

Regular Expression

```
/h[a4@](([c<][k\|<])|([k\|<])|(x))\s+((d)|\
([t\+]h))[3ea4@]\s+p[1l][a4@]n[3e][t\+]/i
```

Types of Testing

- Unit Testing
- Integration Testing
- Functional Testing

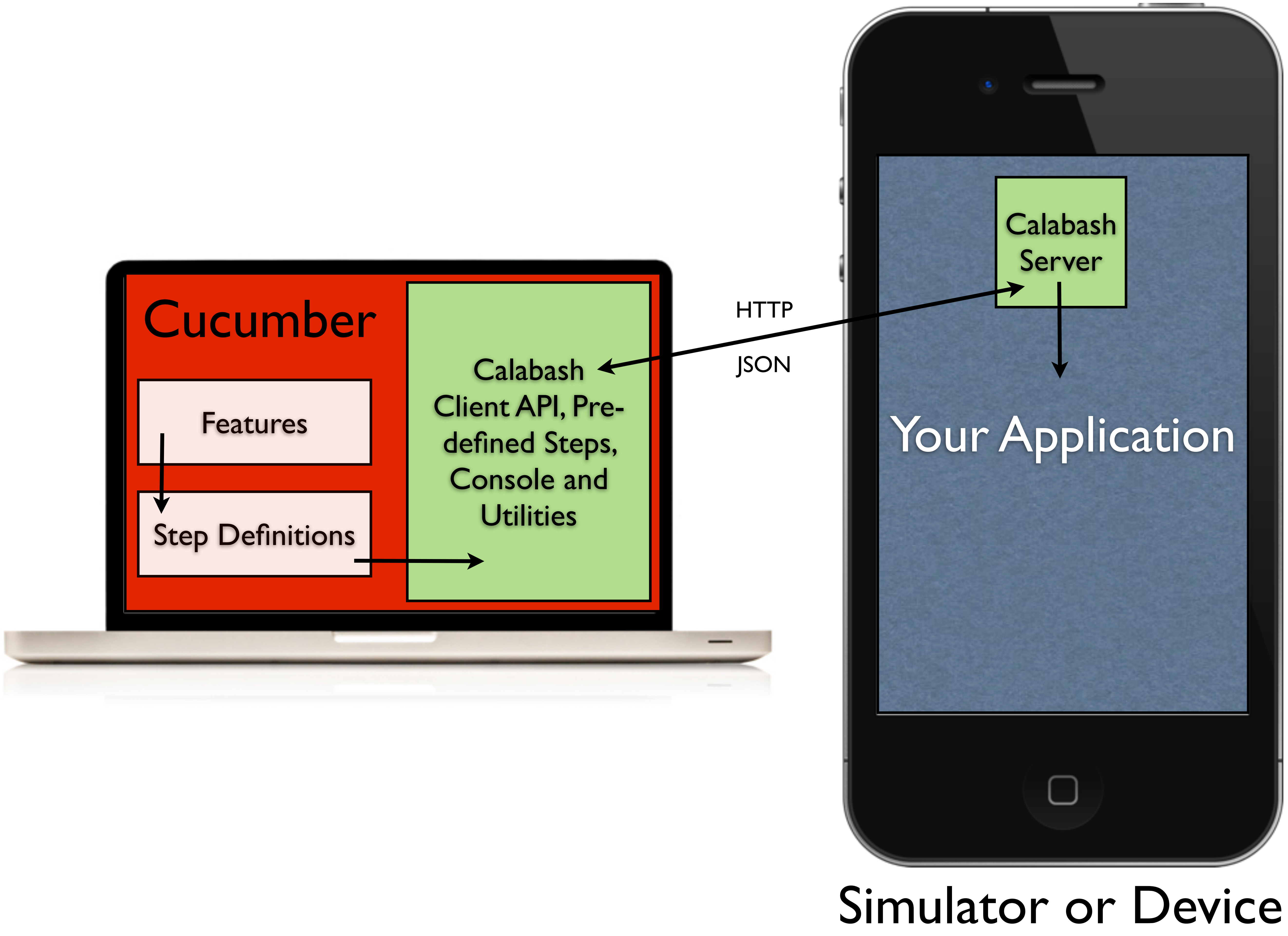
Calabash

Android

iOS

Calabash

- Automated functional testing
- Mobile apps
- Based upon Cucumber
- Works on iOS and Android
- Open Source/Free
- Originally written by LessPainful
- Managed/maintained by Xamarin


Cucumber
Features
Step Definitions
Calabash Client API, Pre-defined Steps, Console and Utilities
HTTP
JSON
Calabash Server
Your Application
Simulator or Device

Features:

Structure

Feature: Launching the application
  As Dead Drop Spy
  I want to be able to successfully launch
  and navigate into the application
  So that I can enjoy using it without any problems

Scenario: Successfully launching the application

Scenario: Successfully navigating past the Home Screen and into the application

Features
Scenarios: Given, When, Then, And

Scenario: Successfully launching the application
  Given *I have launched the application*
  Then *I should see the Home Screen*

Scenario: Successfully navigating past the Home Screen and into the application
    Given *I can see the Home Screen*
    When *I swipe up on "HomeScreen"*
    Then *I should see the Dead Drop List*
    And *I should see the Dead Drop Mini Map*

Step Definitions

```
Given(/^I have launched the application$/) do
  wait_for() {
    element_exists("view")
  }
end

Then(/^I (can|should) see the Home Screen$/) do | c |
  wait_for() {
    element_exists("view marked:'HomeScreen'")
  }
  setCurrentScreen("#{HOME}")
end
```

Instrumenting your application

```
@implementation HomeViewController

- (void)viewDidLoad {
    [super viewDidLoad];

     self.view.accessibilityIdentifier = @"HomeScreen";
}

@end
```

Calabash

Predefined Steps

- **Screenshots:** Then take picture
- **Touching/Tapping:** Then I touch the "login" button
- **Entering Text:** Then I fill in "placeholder" with "text to write"
- **Waiting:** Then I wait to see "text or label"
- **Swiping:** Then I swipe down on "someView"
- **Pinching:** Then I pinch to zoom in
- **Scrolling:** Then I scroll down
- **Device Rotation:** Then I rotate device left
- **Assertions:** Then I don't see the "someView"

Calabash
Ruby iOS API

- Use within step definitions
- Test via Calabash Console
- Querying, touching, swiping, pinching, rotating, scrolling, entering text, waiting, asserting