Calabash

Cross-Platform Automated UI Testing

Olexandr Leuschenko
Nov 22, 2014

Agenda

- UI Test Automation with Calabash

- Calabash tests as documentation

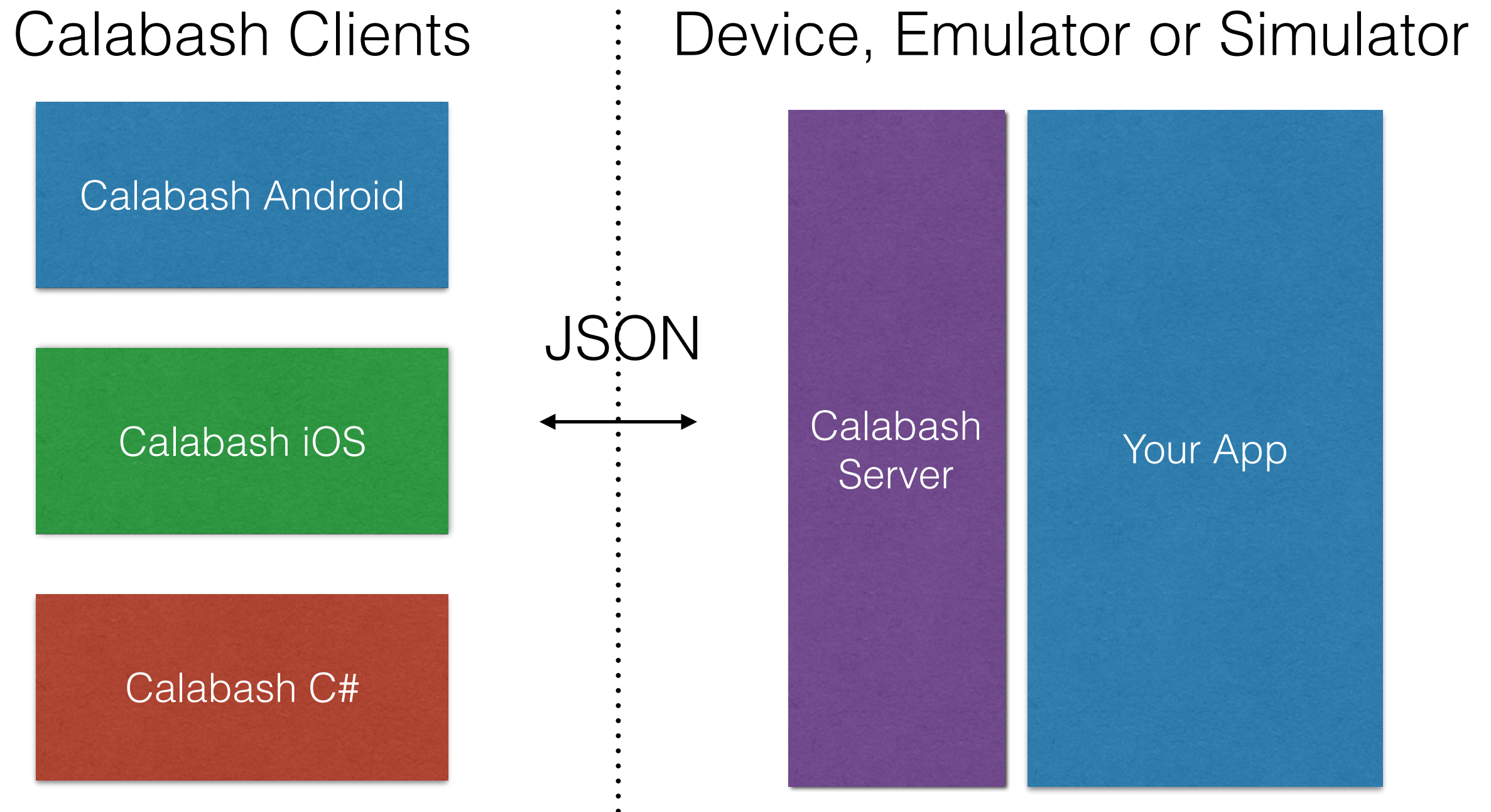- Cross-platform Best Practices

- Bonus

Agenda

- UI Test Automation with Calabash

- Calabash tests as documentation

- Cross-platform Best Practices

- Bonus

What is Calabash?

- UI Test Automation for iOS & Android

- Open-Source (https://github.com/calabash)

- Native and Hybrid apps

- Ruby and C# API

- Behaviour Driven Development (BDD)

High-level architecture

Calabash Clients

Device, Emulator or Simulator

Calabash Android

Calabash iOS

Calabash C#

JSON

Calabash Server

Your App

Calabash Test

Feature

Scenario

Step Interface

iOS Step Implementation

Android Step Implementation

Calabash test

Feature: **UaMobile agenda**

Scenario: Calabash presentation is present in timeline

Given I see a navigation bar titled **"Timeline"**

When I swipe left

Then I should see **"Розклад"**

When I press **"Розклад"**

Then I see a navigation bar titled **"Розклад"**

And I scroll down until I see text **"Тестування мобільних додатків з Calabash"**

Calabash Demo

Calabash Predefined Steps

- Screenshots

- Touching (tapping)

- Entering text

- Waiting

- Back button

- Swipes

- Pinch

- Playback of touch events

- Device orientation

- Assertions

https://github.com/calabash/calabash-ios/wiki/02-Predefined-steps

Agenda

- UI Test Automation with Calabash

- Calabash tests as documentation

- Cross-platform Best Practices

- Bonus

Calabash

- ~~Writable by your boss~~

- Specification

- Readable by all project members

- Improves communication

Calabash tests as documentation

- Brownfield development (legacy applications)

- Greenfield development (new projects)

Brownfield development

Greenfield development

BDD lifecycle

1. Write the Feature

2. Run the Feature (fail!)

3. Create the Step Definitions

4. Explore the Application using the Calabash Console

5. Update Step Definitions

6. Run the Feature

7. Implement the Feature in the Application

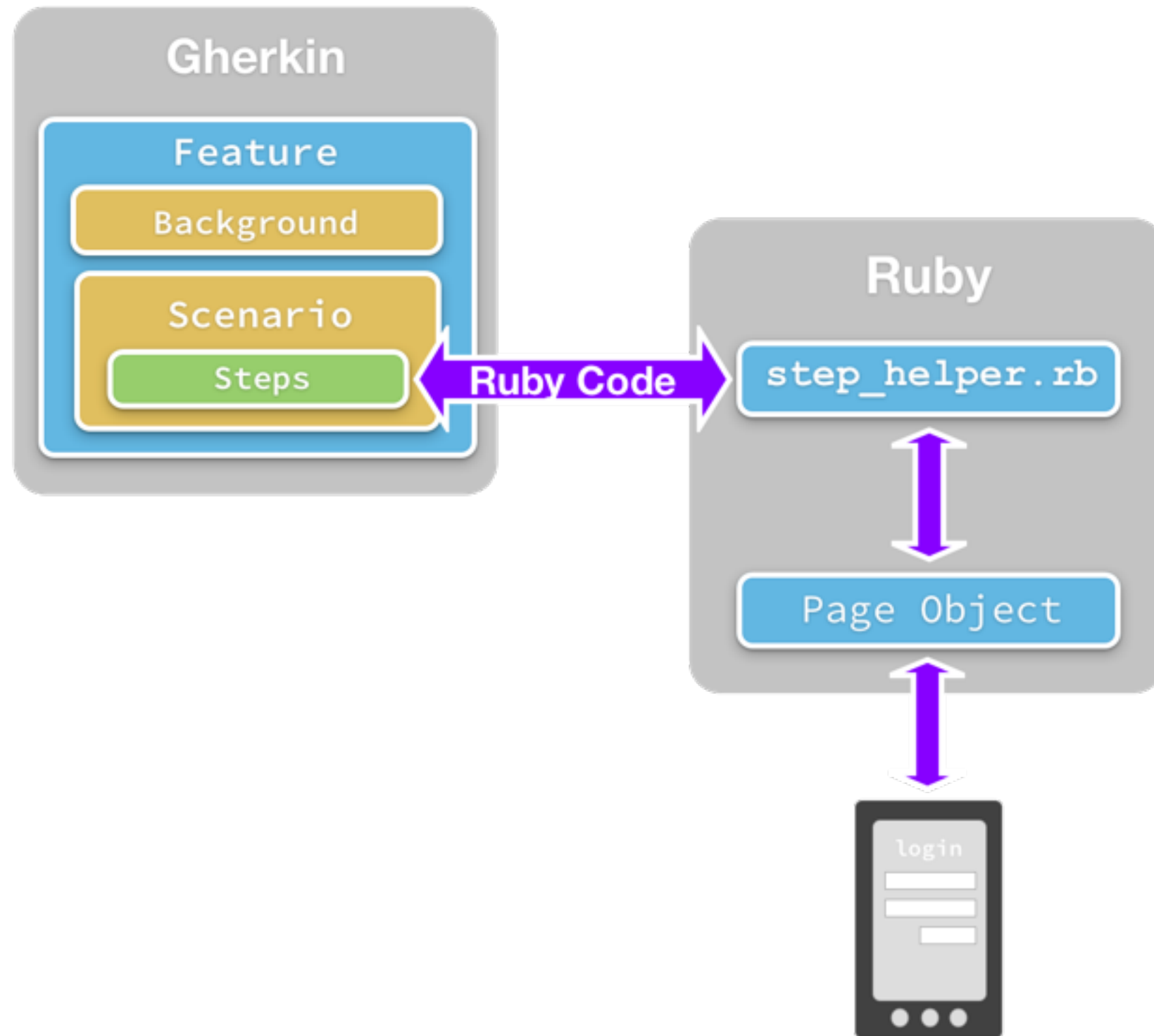# Explore the Application using the Calabash Console

Calabash Console Demo

Agenda

- UI Test Automation with Calabash

- Calabash tests as documentation

- **Cross-platform Best Practices**

- Bonus

Page-Object Pattern

Page-Object Pattern

- Reuse

- Encapsulation

- Readability / Maintenance

Page-Object Pattern Demo

Agenda

- UI Test Automation with Calabash

- Calabash tests as documentation

- Cross-platform Best Practices
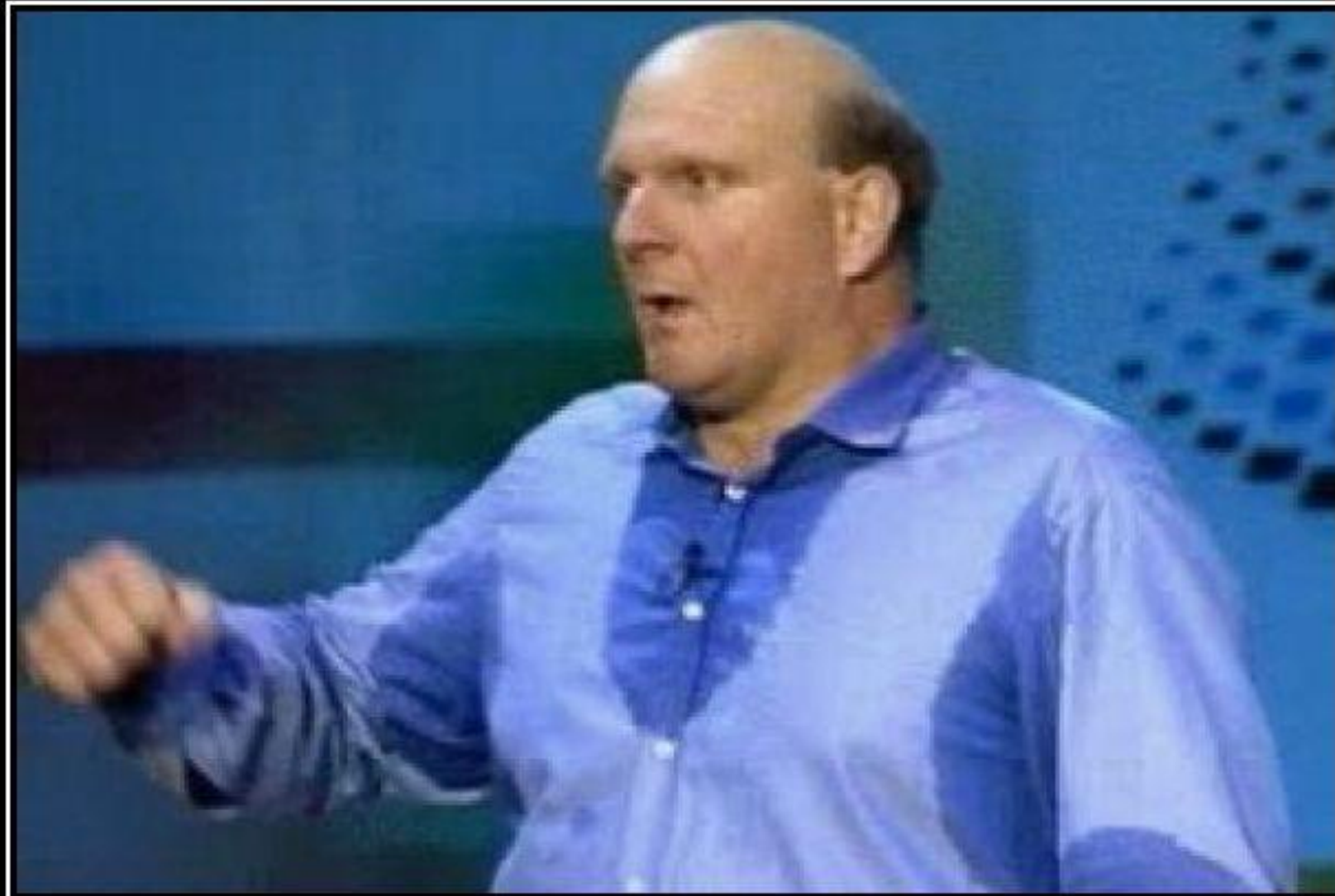
- Bonus

Xamarin.UITest

- **Not** a BDD Framework

- Each test is a method inside a Unit Test Fixture class

- Arrange-Act-Assert pattern

Xamarin.UITest

```
[Test]
public void CreditCardNumberTooShortDisplayErrorMessage ()
{
    // Arrange — set up our queries for the views
    Func<AppQuery, AppQuery> editTextView = c => c.Marked ("CreditCardTextField");
    Func<AppQuery, AppQuery> validateButton = c => c.Marked ("ValidateButtonLabel");
    Func<AppQuery, AppQuery> errorMessage = c => c.Marked ("ErrorMessagesTextField")
        .Text ("Credit card number is too short.");

    // Act
    _app.EnterText (editTextView, new string ('9', 15));
    _app.Tap (validateButton);

    // Assert
    AppResult[] result = _app.Query (errorMessage);
    Assert.IsTrue (result.Any (), "The error message is not being displayed.");
}
```
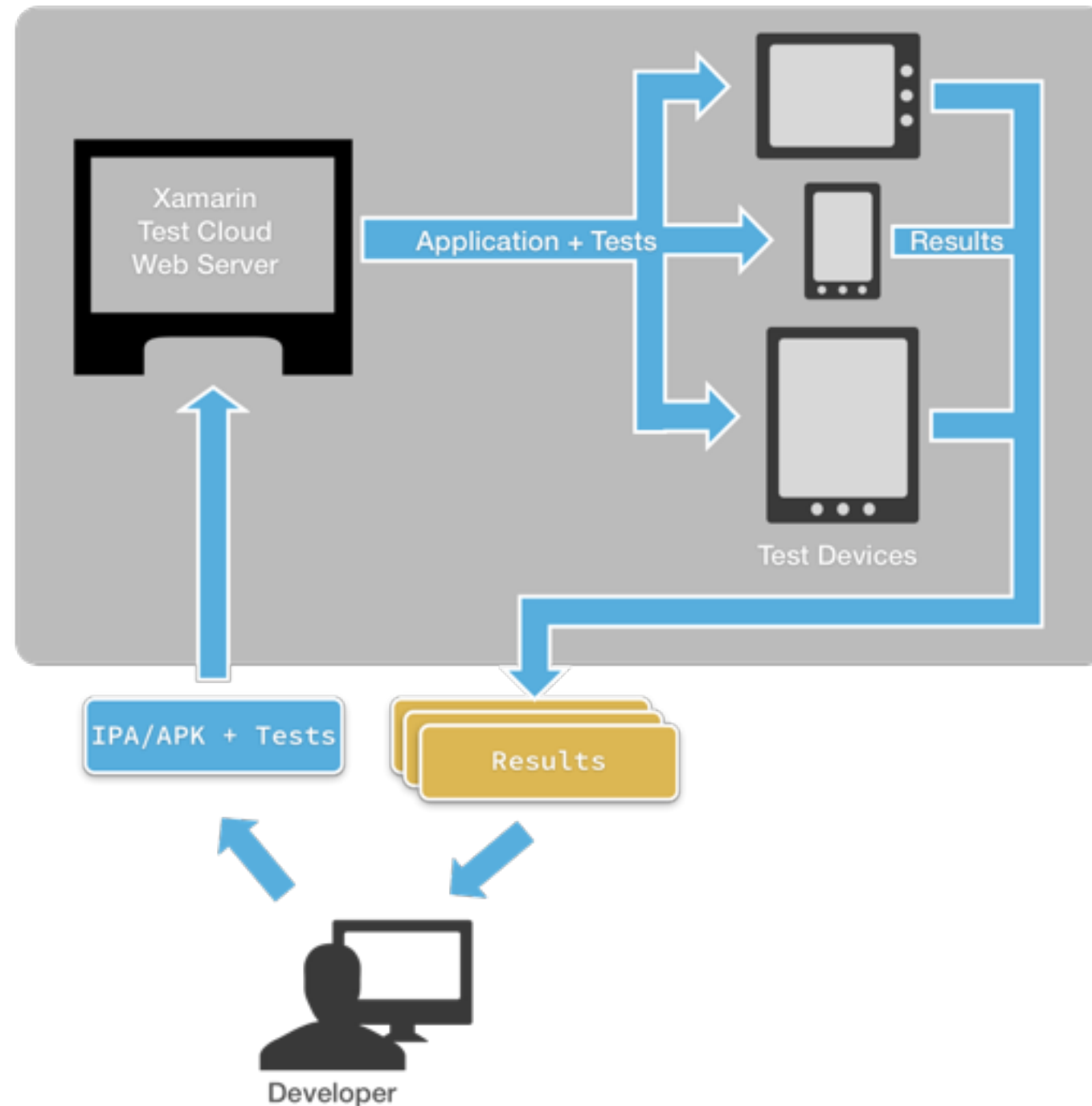
FOUR WORDS

DEVELOPERS DEVELOPERS DEVELOPERS DEVELOPERS

Xamarin Test Cloud

Xamarin Test Cloud

Xamarin Test Cloud

Basic	**Professional**	**Business**	**Enterprise**
$1,000	**$5,000**	**$8,000**	**$12,000**
/month	/month	/month	/month
paid annually	paid annually	paid annually	paid annually
✅ 2 apps	✅ 4 apps	✅ 10 apps	✅ 20 apps
✅ 200 monthly device hours	✅ 1000 monthly device hours	✅ 1600 monthly device hours	✅ 2400 monthly device hours
✅ Email support	✅ 1 Xamarin University seat	✅ 3 Xamarin University seats	✅ 5 Xamarin University seats
	✅ Dedicated customer success manager	✅ Dedicated customer success manager	✅ Dedicated customer success manager
	✅ Priority execution	✅ 4 hours/month of automation consulting	✅ 8 hours/month of automation consulting
		✅ Priority execution	✅ Priority execution
			✅ Enterprise Service Level Agreement

Q & A

@OlexaLe

olexa.le@gmail.com