



Automated Testing with Appium



Automated Testing



<http://appium.io/docs/en/about-appium/intro/>



Automated Testing

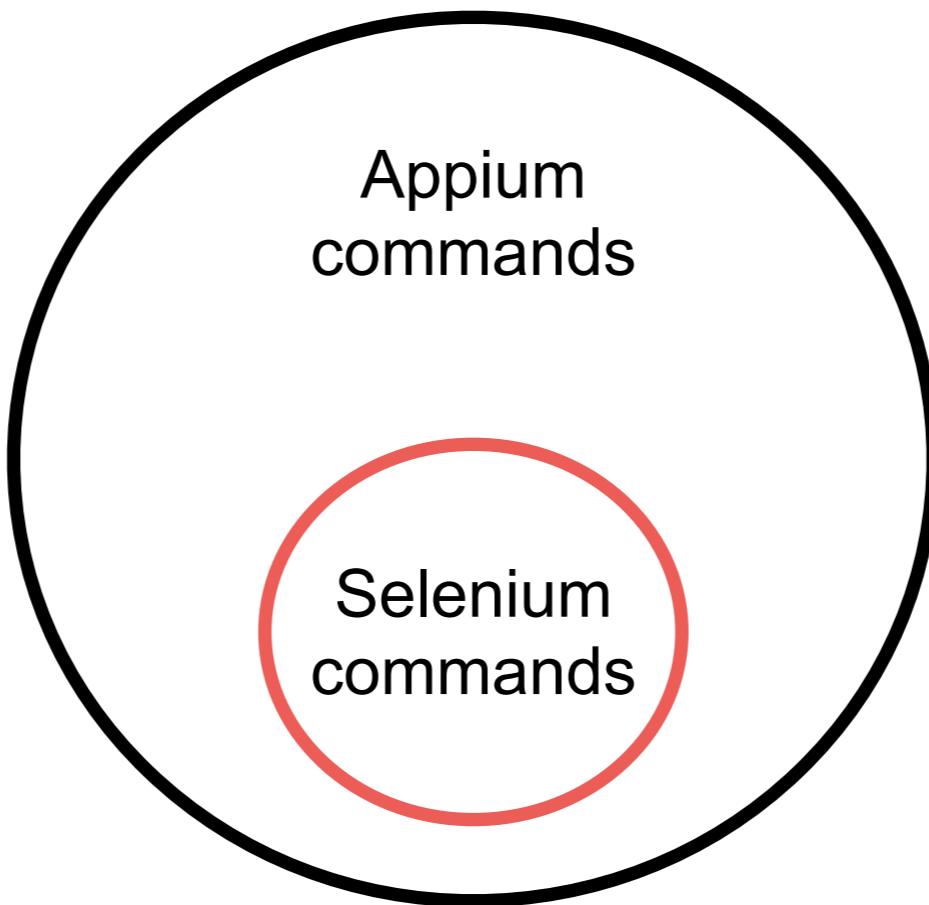


<https://appium.github.io/appium/docs/en/2.0/>

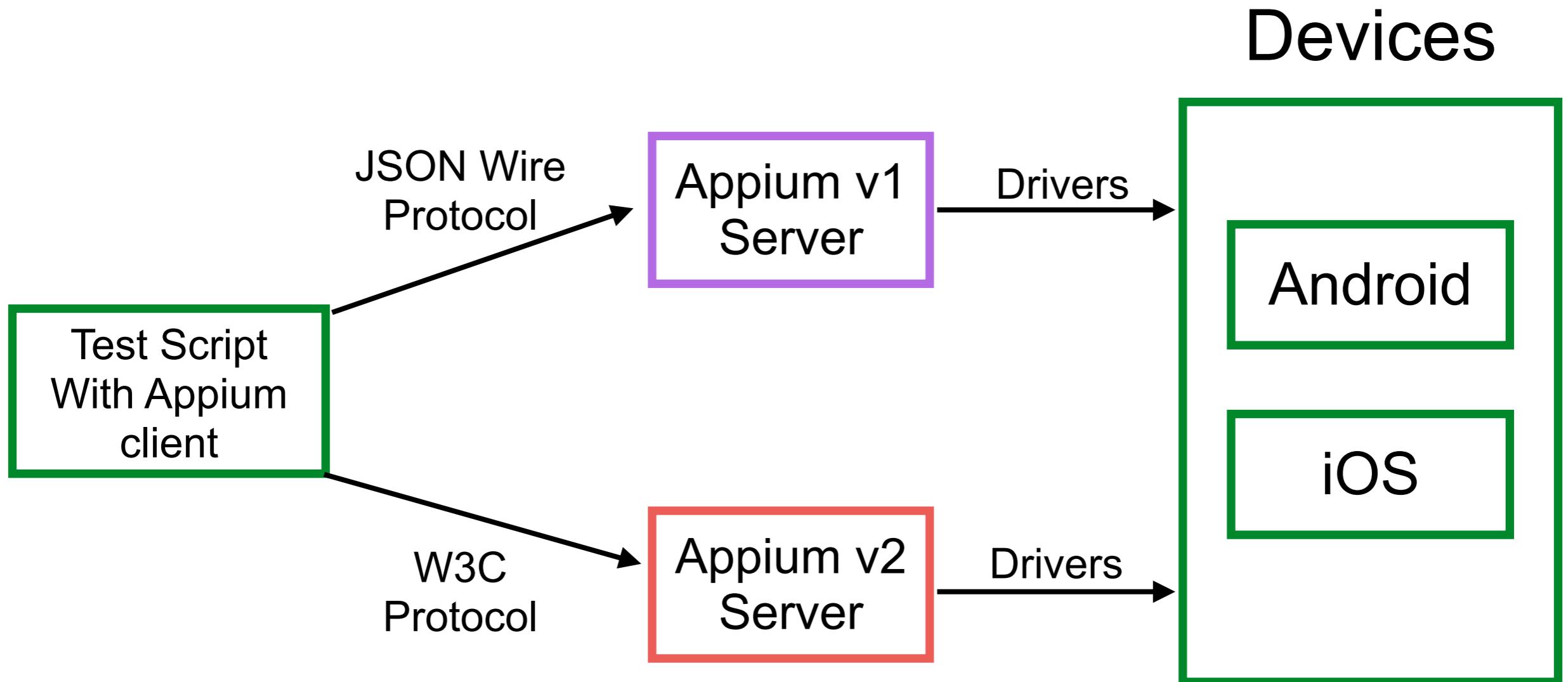


Appium

Automation tool for mobile app
Selenium for mobile
High compatibility with selenium



Architecture



Appium Tools

Appium Server via npm

Appium Server GUI/Desktop (deprecated)

Appium Inspector

<http://appium.io/docs/en/about-appium/getting-started/?lang=en#installing-appium>



Appium Clients

Ruby
Python
Java
C#

Robotframework + Appium Library

<http://appium.io/docs/en/about-appium/appium-clients/index.html>



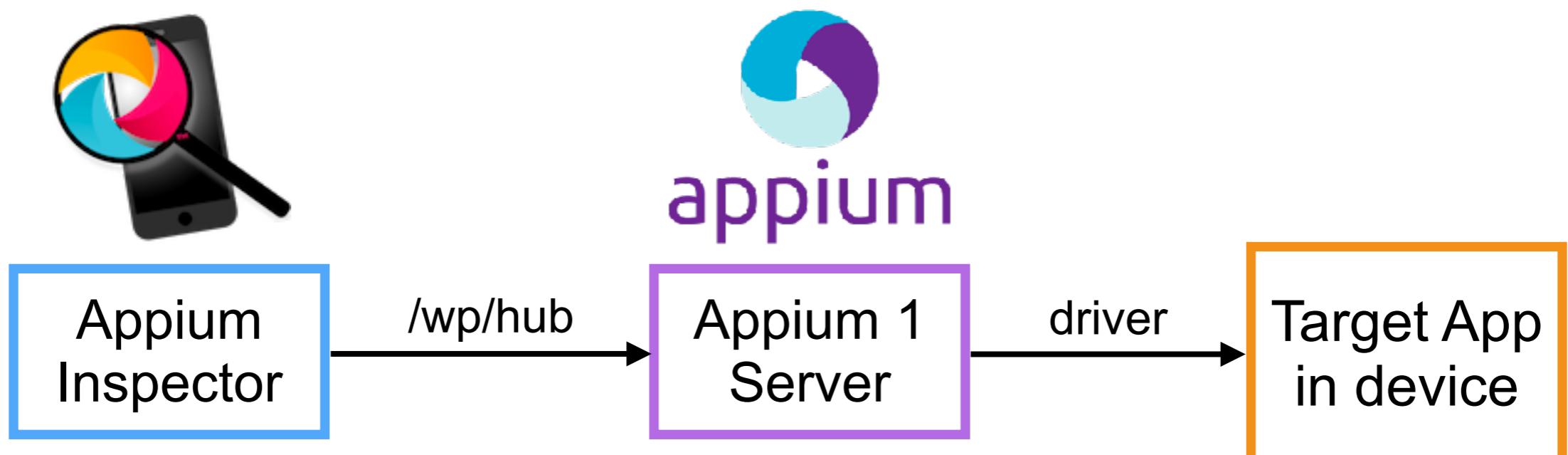
Appium Drivers

| Platform | Driver | Platform Versions | Appium Version |
|----------|--------------|-------------------|----------------|
| iOS | XCUITest | 9.3+ | 1.6.0+ |
| | UIAutomation | 8.0 to 9.3 | All |
| Android | Espresso | ?+ | 1.9.0+ |
| | UiAutomator2 | ?+ | 1.6.0+ |
| | UiAutomator | 4.3+ | All |
| Mac | Mac | ?+ | 1.6.4+ |
| Windows | Windows | 10+ | 1.6.0+ |



Let's start with Appium

Appium doctor (android, iOS)



Steps to run

Appium doctor

Start server

Provide target apps (api, ipa/app)

Appium Inspector



Appium Doctor



Appium Doctor

Check before start with Appium...

```
$npm install -g @appium/doctor  
$appium-doctor
```

Android
env

iOS
env

<https://github.com/appium/appium/tree/master/packages/doctor>



Appium Doctor

\$appium-doctor

```
info AppiumDoctor Appium Doctor v.1.12.1
info AppiumDoctor ### Diagnostic for necessary dependencies starting ####
info AppiumDoctor ✓ The Node.js binary was found at: /Users/somkiat/.volta/tools/image/node
info AppiumDoctor ✓ Node version is 14.20.0
info AppiumDoctor ✓ Xcode is installed at: /Applications/Xcode.app/Contents/Developer
info AppiumDoctor ✓ Xcode Command Line Tools are installed in: /Applications/Xcode.app/Con
info AppiumDoctor ✓ DevToolsSecurity is enabled.
info AppiumDoctor ✓ The Authorization DB is set up properly.
WARN AppiumDoctor ✖ Carthage was NOT found!
info AppiumDoctor ✓ HOME is set to: /Users/somkiat
WARN AppiumDoctor ✖ ANDROID_HOME is NOT set!
WARN AppiumDoctor ✖ JAVA_HOME is NOT set!
WARN AppiumDoctor ✖ adb could not be found because ANDROID_HOME is NOT set!
WARN AppiumDoctor ✖ android could not be found because ANDROID_HOME is NOT set!
WARN AppiumDoctor ✖ emulator could not be found because ANDROID_HOME is NOT set!
WARN AppiumDoctor ✖ Bin directory for $JAVA_HOME is not set
info AppiumDoctor ### Diagnostic for necessary dependencies completed, 7 fixes needed. ####
info AppiumDoctor
info AppiumDoctor ### Diagnostic for optional dependencies starting ####
WARN AppiumDoctor ✖ opencv4nodejs cannot be found.
info AppiumDoctor ✓ ffmpeg is installed at: /usr/local/bin/ffmpeg. ffmpeg version 5.1 Copy
he FFmpeg developers
WARN AppiumDoctor ✖ mjpeg-consumer cannot be found.
WARN AppiumDoctor ✖ idb and idb_companion are not installed
info AppiumDoctor ✓ applesimutils is installed at: /usr/local/bin/applesimutils. Installed
```



Check environments

| Android | iOS |
|-----------------------------|---------------------------|
| APPIUM_HOME | APPIUM_HOME |
| ANDROID_HOME | Xcode |
| JAVA_HOME | Xcode command line tools |
| adb, emulator, apk analyzer | ios-deploy, applesimutils |
| NodeJS | NodeJS |



List of android's devices

\$adb devices



List of iOS's devices

\$xcrun xctrace list devices

```
== Simulators ==
Apple TV Simulator (15.2) (9666637B-AD71-47CD-932D-DE7BA9096F46)
Apple TV 4K (2nd generation) Simulator (15.2) (2B6A15B3-4A03-41B5-B6C6-D7727DCA94D8)
Apple TV 4K (at 1080p) (2nd generation) Simulator (15.2) (EC5B5310-9DE7-4CB2-8AE2-172FC885DB8C)
iPad (9th generation) Simulator (15.2) (03B84395-1A2B-4759-B01A-171CD4F8F796)
iPad Air (4th generation) Simulator (15.2) (6389A3CA-7805-452D-9110-5B3796A9D9BB)
iPad Pro (11-inch) (3rd generation) Simulator (15.2) (40E522A5-14CE-453C-A54C-29D096F19236)
iPad Pro (12.9-inch) (5th generation) Simulator (15.2) (1E6E63EF-FA56-4E31-8082-13B0C664AD33)
iPad Pro (9.7-inch) Simulator (15.2) (9C7181D9-73D6-48BF-8972-80A73ADB2EA9)
iPad mini (6th generation) Simulator (15.2) (01F8F0AB-1AF5-4FBE-9587-B7713F7BBC6F)
iPhone 11 Simulator (15.2) (1E75E325-57EC-4D6D-85BF-9958B0A9B158)
iPhone 11 Pro Simulator (15.2) (7D50F81C-3521-447F-A79C-8613C6C58FEF)
iPhone 11 Pro Max Simulator (15.2) (FB336CEE-A241-43D1-8704-B0632EA9FD28)
iPhone 12 Simulator (15.2) (5325CF3D-9576-462A-A110-49A5D76665FE)
```

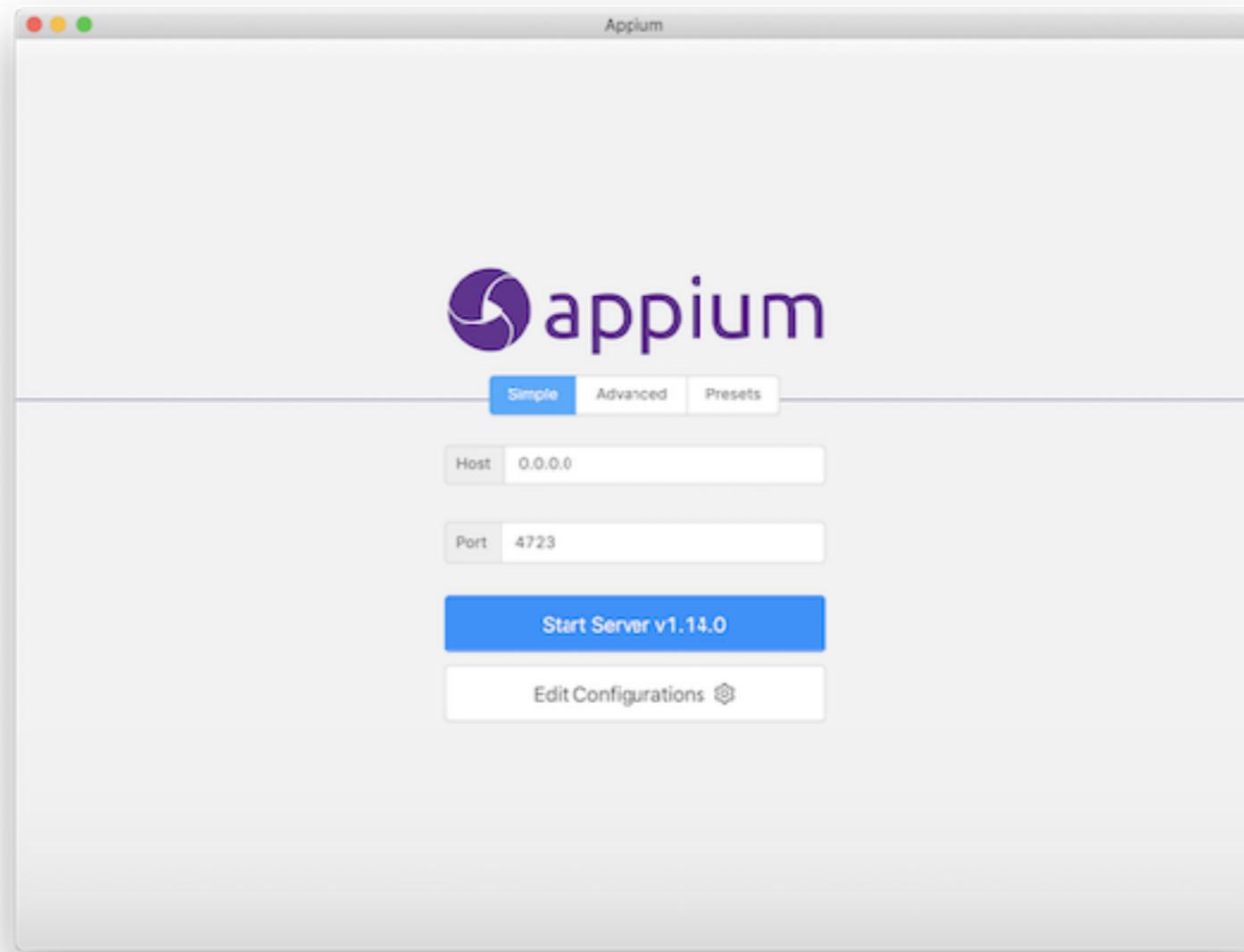


Appium Server



Appium Server Desktop

DEPRECATED !!!



<https://github.com/appium/appium-desktop>



Appium Server

Install via npm (required Node.js)

```
$npm install -g appium  
$appium
```

<https://github.com/appium/appium>

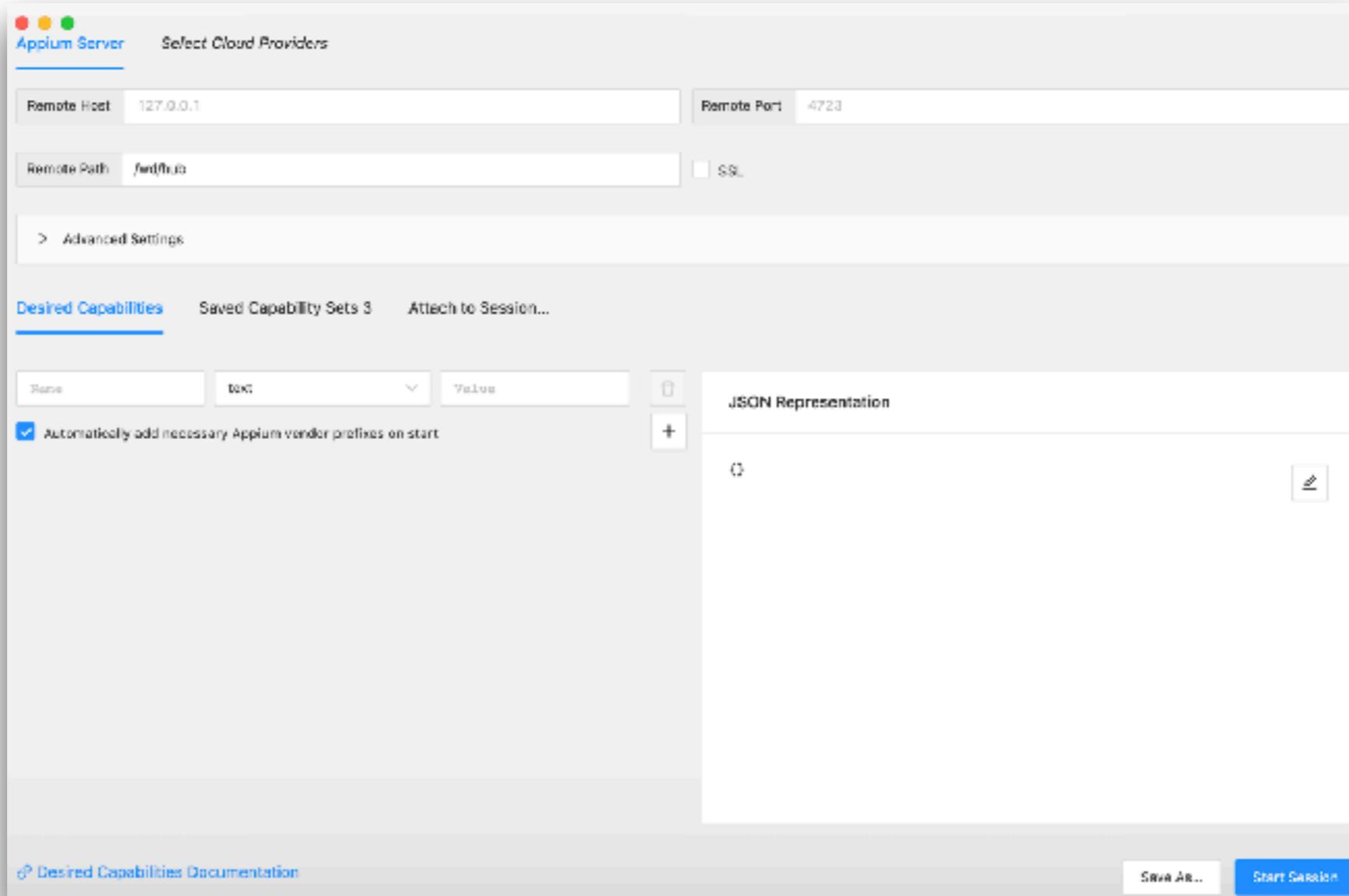


Appium Inspector



Appium Inspector

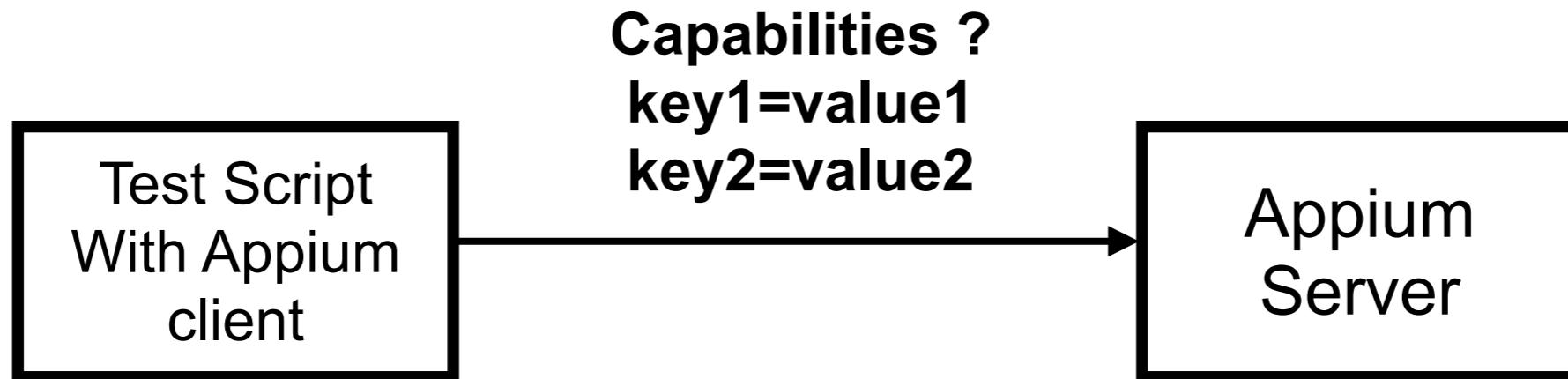
GUI inspector for mobile apps



<https://github.com/appium/appium-inspector>



Create a session with capabilities



<http://appium.io/docs/en/writing-running-appium/caps/index.html>



Required capabilities

| Name | Description |
|-----------------|-------------------------------------|
| platformName | Platform to automate (iOS, Android) |
| platformVersion | Version of platform |
| deviceName | Type of device to automate |
| app | Path to your app |

<https://appium.io/docs/en/2.1/guides/caps/>



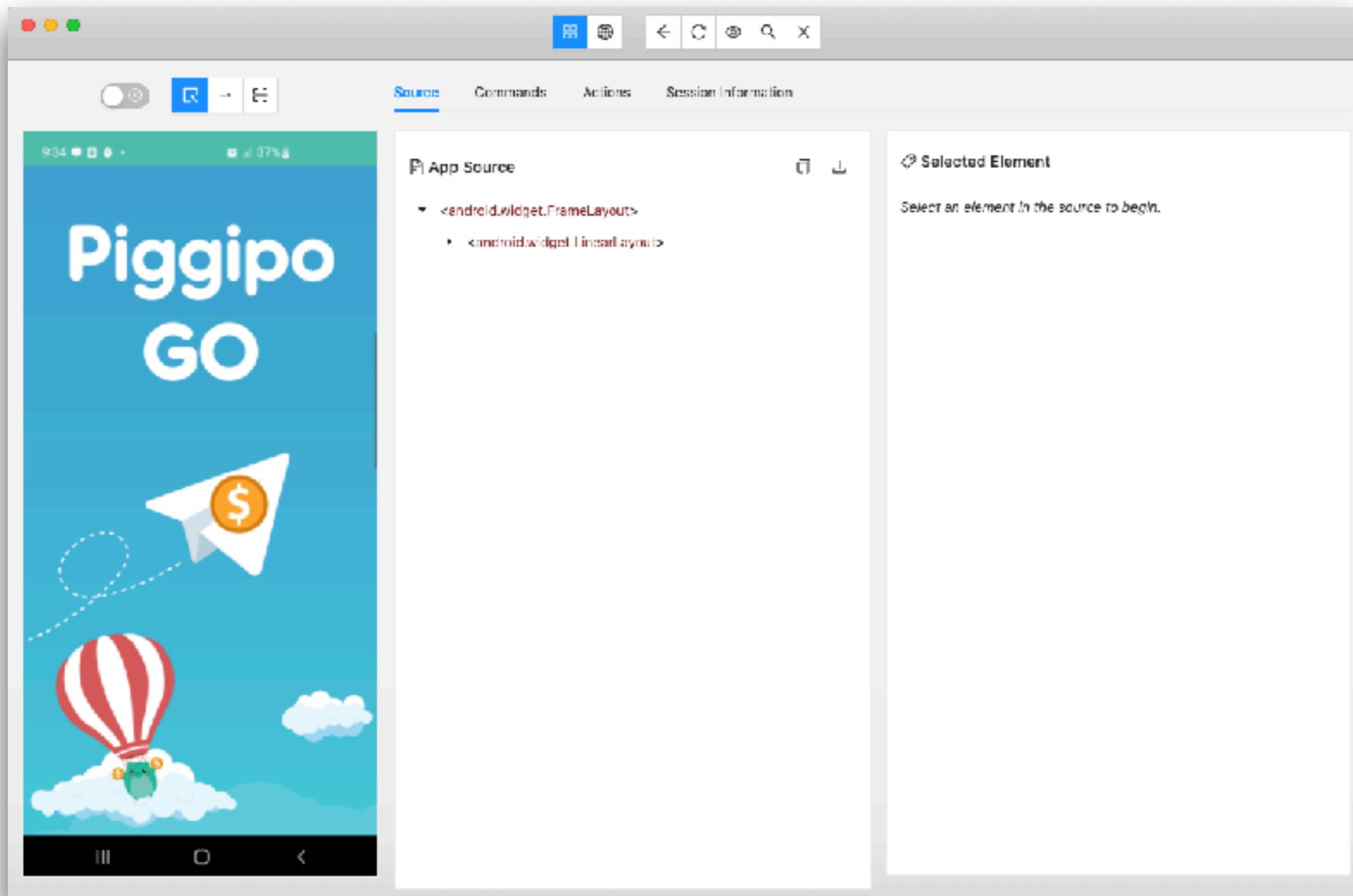
Session capabilities for android

```
{  
  "platformName": "Android",  
  "automationName": "UiAutomator2",  
  "app": "Path to APK file",  
  "deviceName": "ID/Name of target device"  
}  
?
```

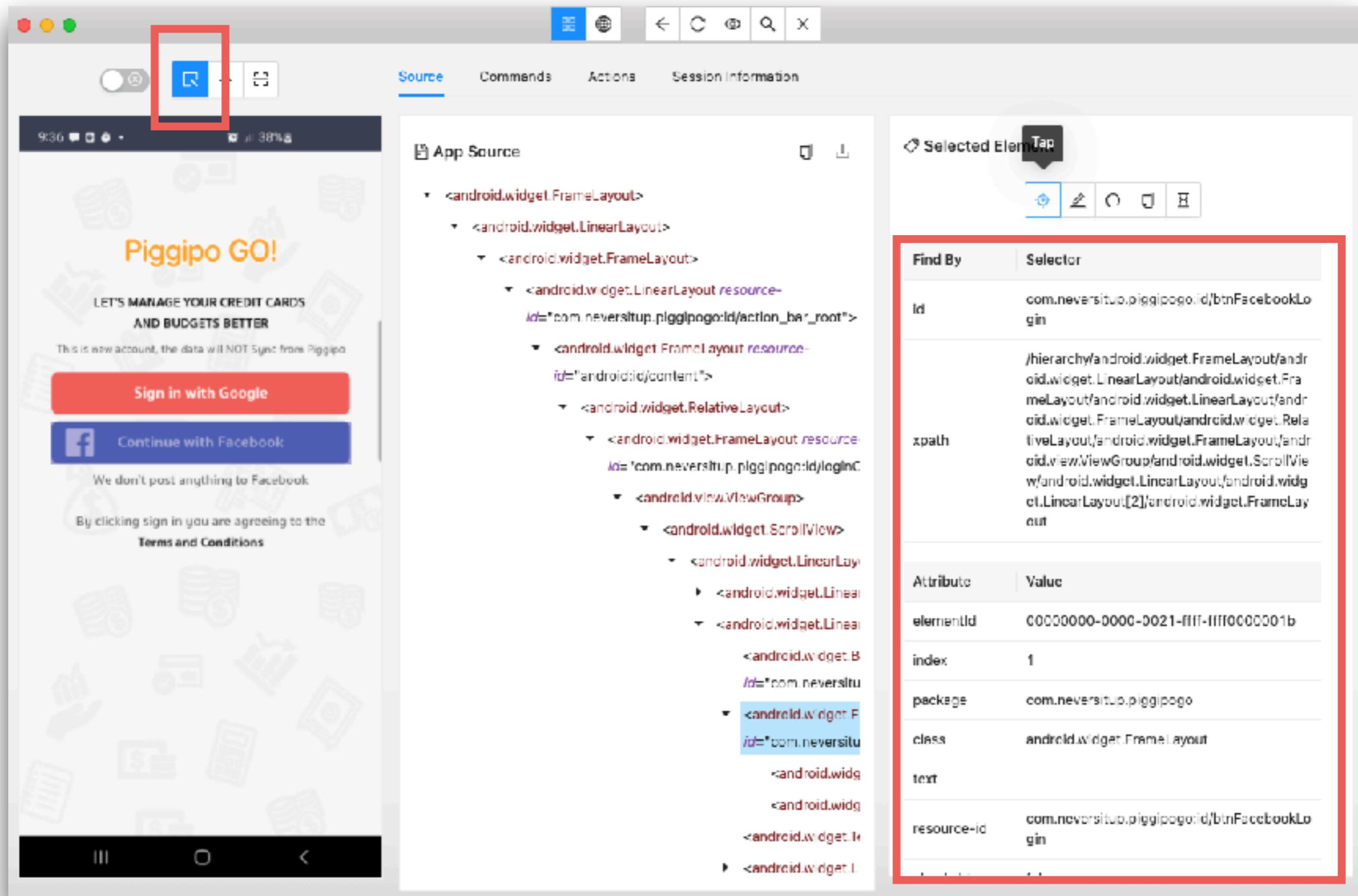
<https://appium.io/docs/en/drivers/android-uiautomator2/>



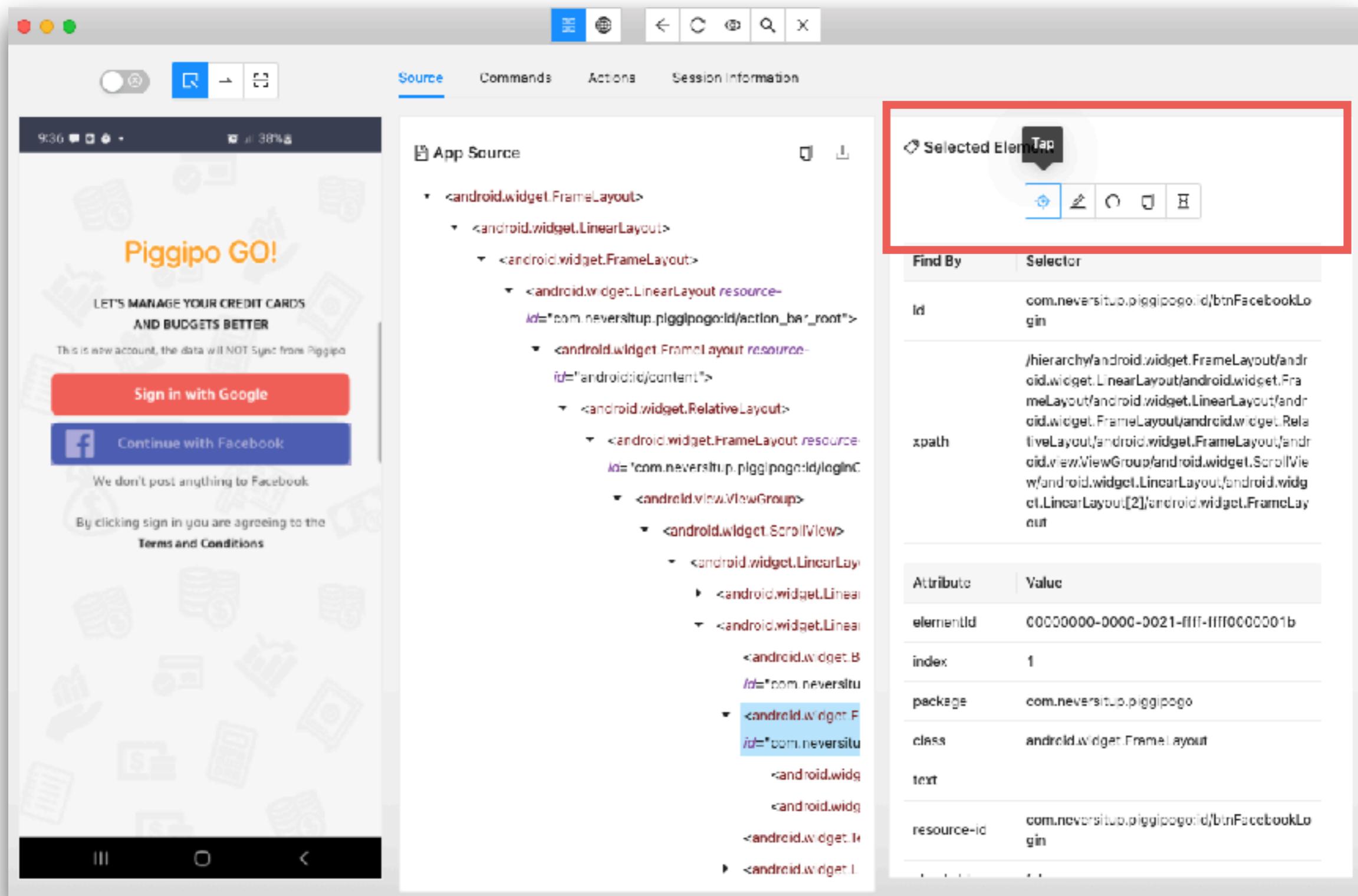
Appium Inspector



Inspect Element



Simulate actions on element



Code Example

The screenshot shows a mobile application session information screen. The top navigation bar includes icons for Source, Commands, Actions, and Session Information. The Session Information section displays the following details:

| | | |
|-------------------------|--------------------------|-------|
| Session Length | 00:02:13 | |
| platform | LINUX | |
| Session Details | webStorageEnabled | false |
| | takesScreenshot | true |
| Currently Active App ID | com.neversleep.piggipogo | |

A red box highlights the "Start this Kind of Session with Code" section, which contains sample code for the Robot Framework. The code uses the Appium robot client and includes settings and variables for running the session.

```
# This sample code uses the Appium robot client
# pip install robotframework-appiumlibrary
# Then you can paste this into a file and simply run with robot
#
# more keywords on: https://github.com/robotframework-appiumlibrary/AppiumLibrary

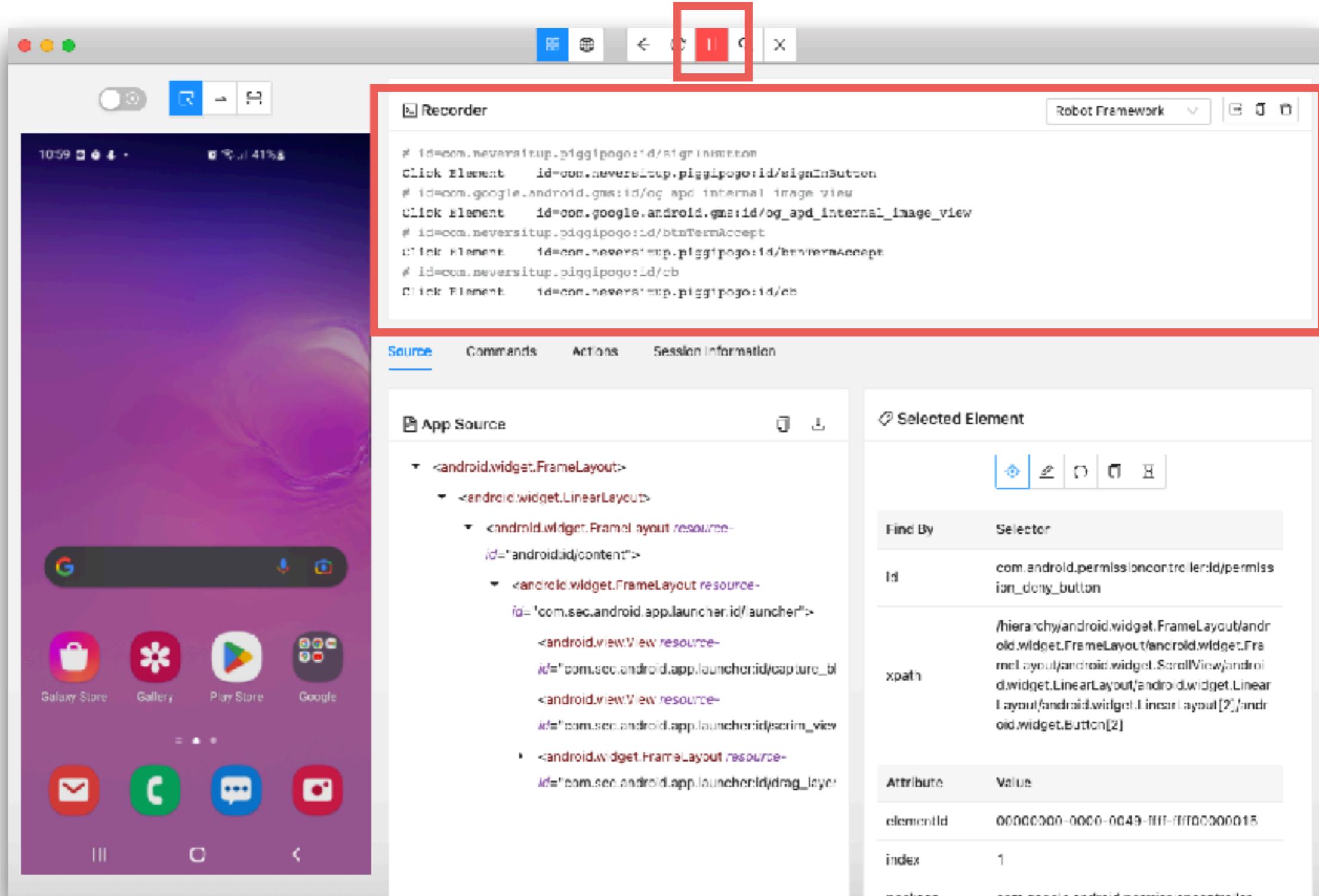
*** Settings ***
Library          AppiumLibrary
Test Teardown    Quit Application
Suite Teardown   Close Application

*** Variables ***
${DEVMODE}        http://127.0.0.1:4723/wd/hub
${platformName}   Android
${appium:app}      /Users/akonkitan/data/alide/appium/workshop/piggipogo_1.2.21.apk
${appium:deviceName} R58M16QKSJK
${appium:automationName} UIAutomator2
${appium:ensureWebviewsHavePages} True
${appium:nativeWebScreenshot} True
```

The right side of the interface shows a dropdown menu for "Robot Framework" with other options like JS - WD (Promise), JS - Webdriver.io, JS - Oxygen HQ, Java - JUnit, Python, Ruby, and Robot Framework.



Try to record test script



Session capabilities for iOS

```
{  
  "platformName": "Android",  
  "automationName": "XCUITest",  
  "app": "Path to IPA file",  
  "platformVersion": "14.5"  
  "deviceName": "ID/Name of target device"  
}
```

<https://appium.io/docs/en/drivers/ios-xcuitest/index.html>



Finding and Using Elements

Isolated from UI !!



Locator Strategies

| Strategy | Description |
|-------------------------|---|
| Accessibility ID | iOS = accessibility-id Android = content-desc |
| ID | iOS = name Android = resource-id |
| Name | Name of element |
| XPath | Pattern of element in XML (not recommended) Absolute path vs Relative path ? |
| Class name | Don't |

<http://appium.io/docs/en/commands/element/find-elements/index.html#selector-strategies>



Good Locators

Unique
Descriptive
Resilient

Shorter in length (maintainability)



XPath

XML Path Language
Query language used to identify tag in XML
Appium builds XML representation of app

XPath is powerful but very dangerous



Benefits of XPath

Find any element that exists

Find elements by using complex criteria



Cons of XPath

XPath selectors can be brittle

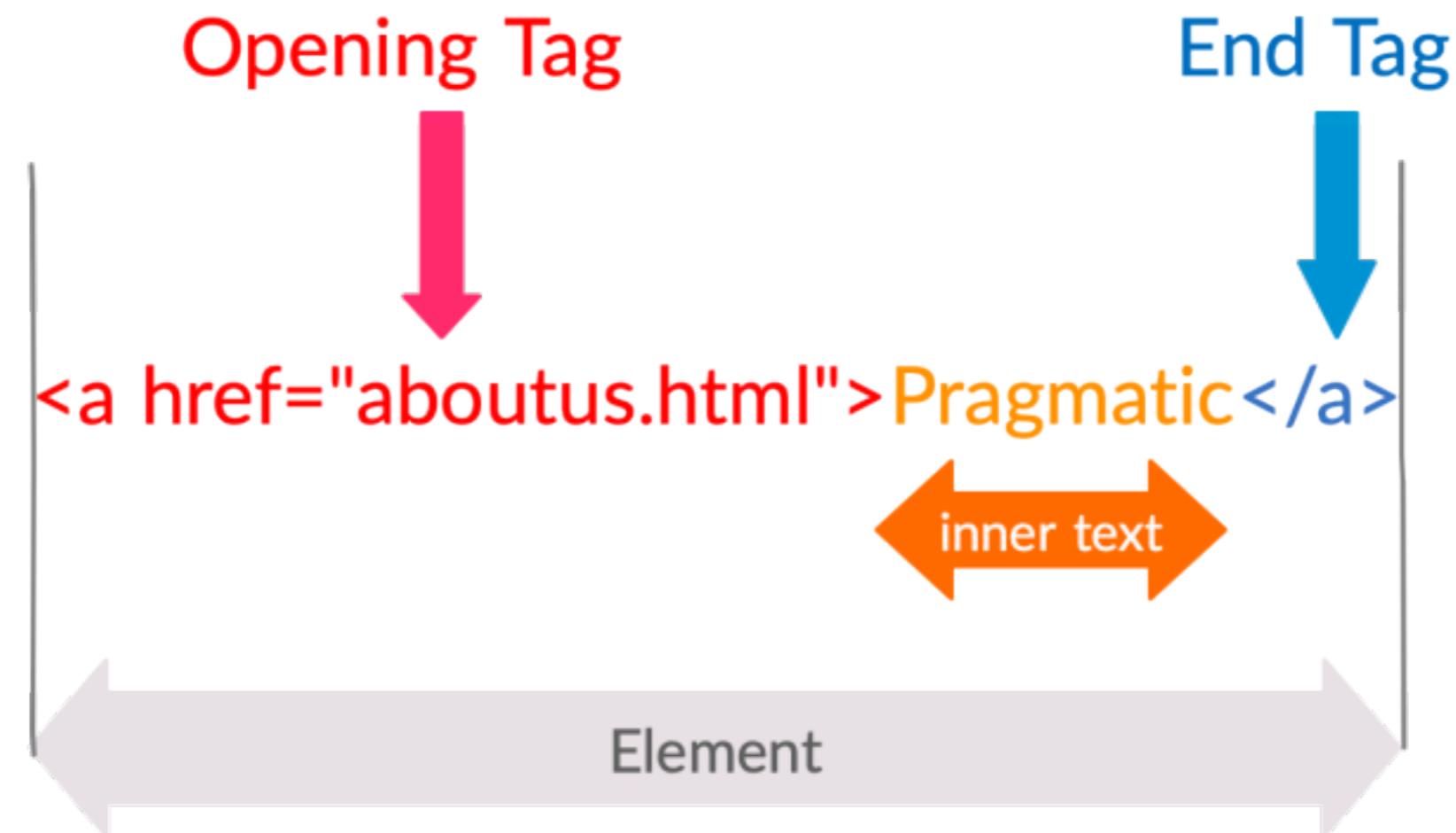
Slow

Broken test !!

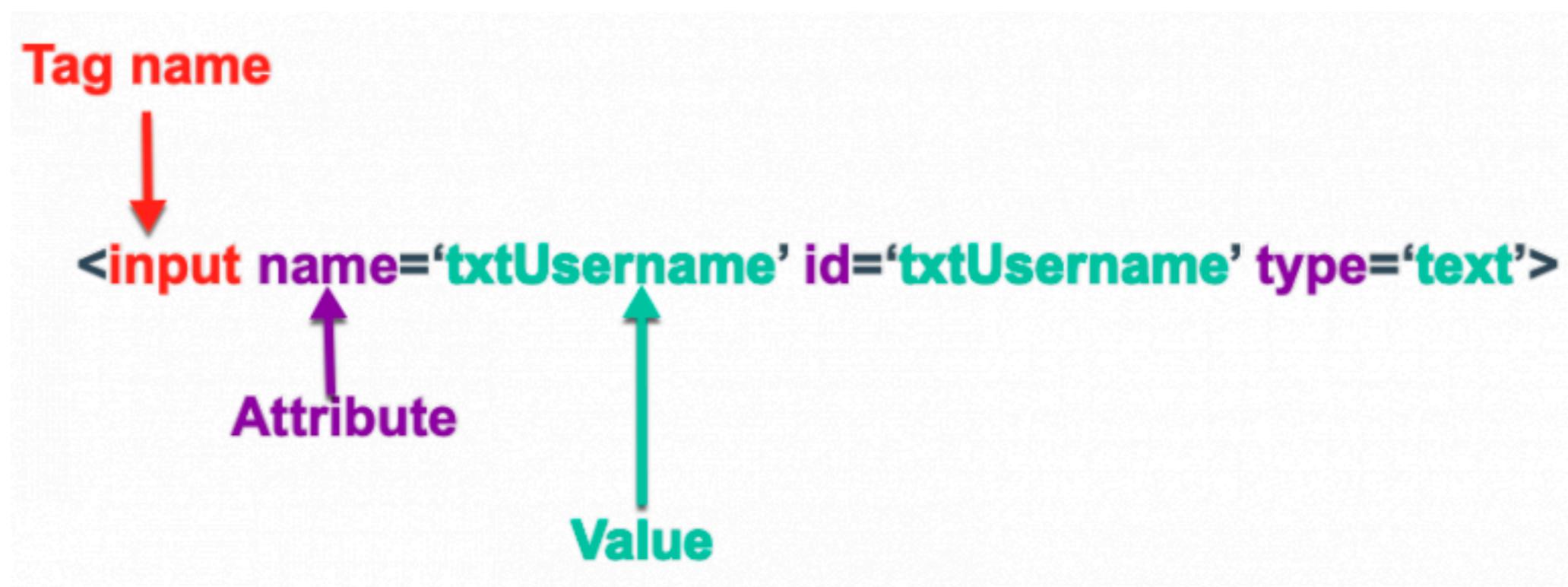
Avoid brittle selectors by rely on unique tag attribute



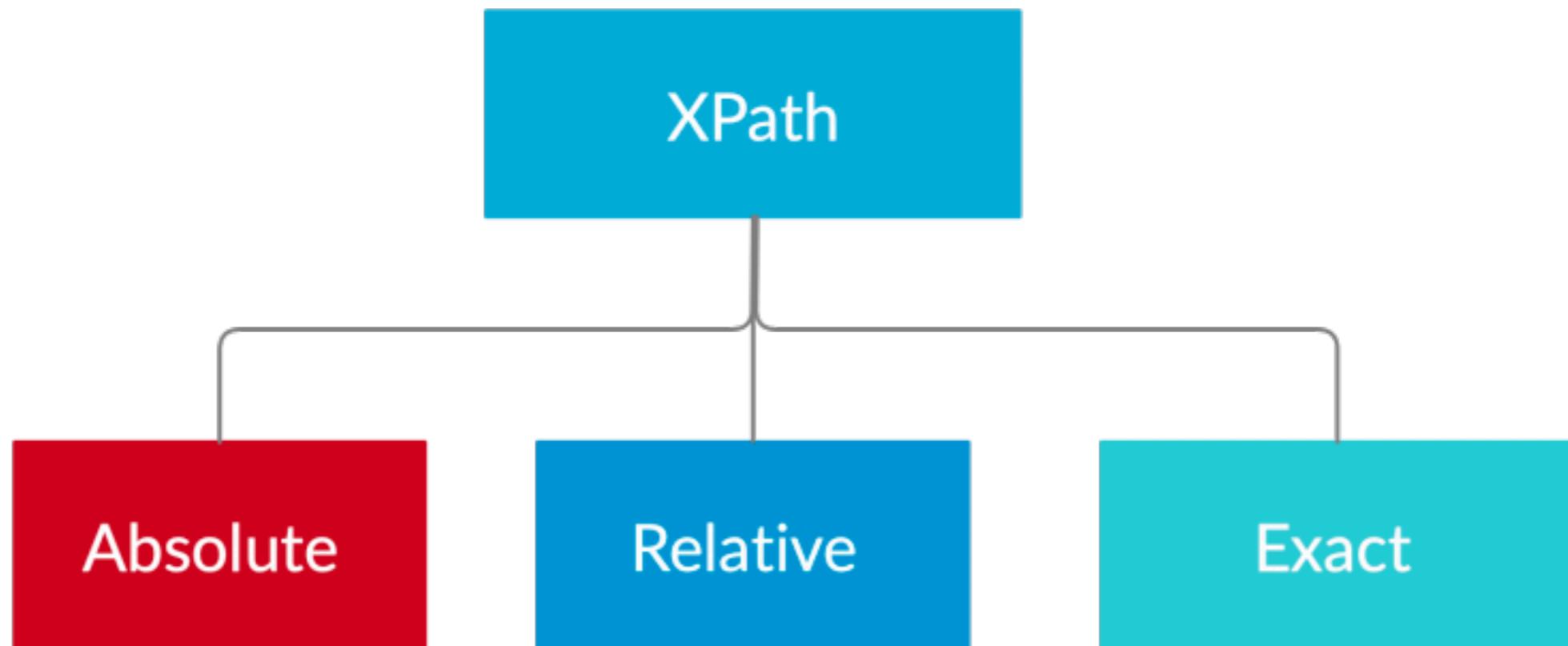
XPath



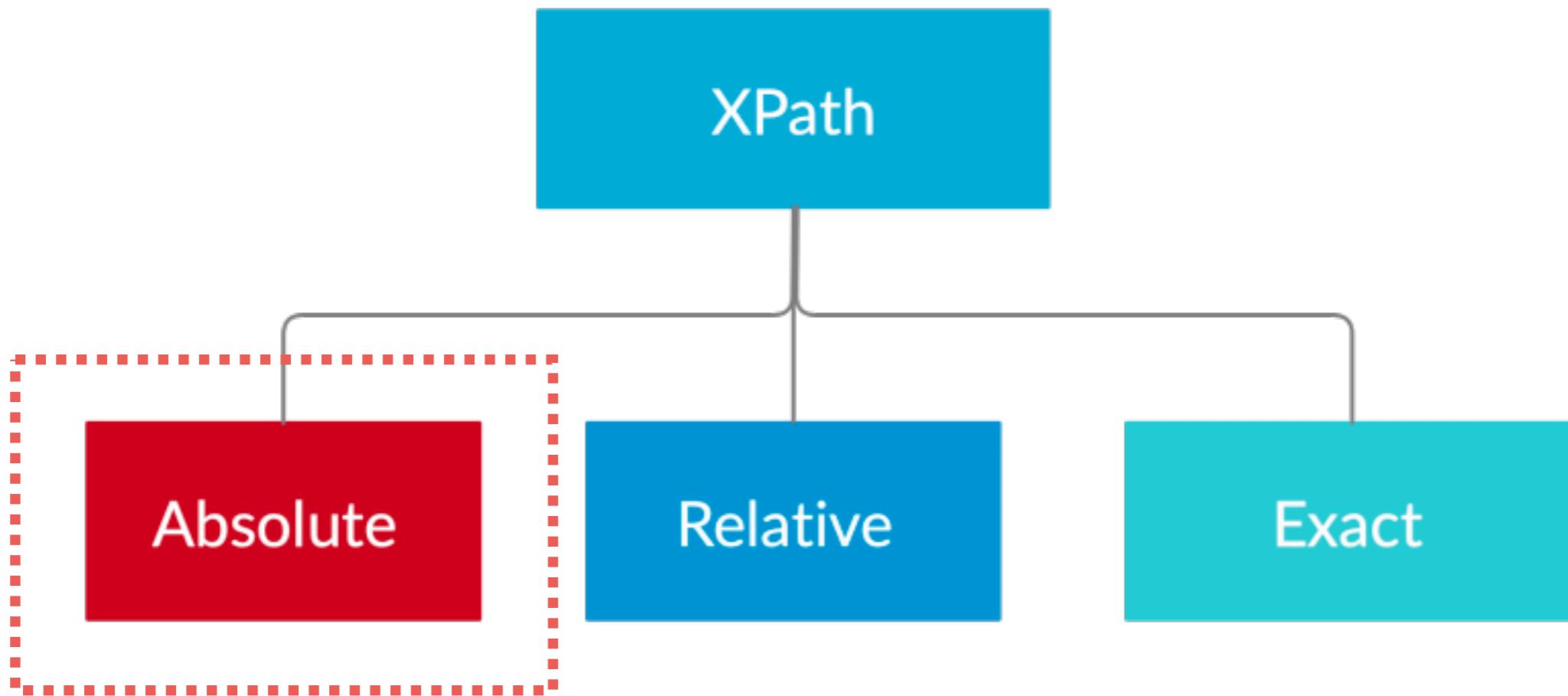
XPath



Types XPath



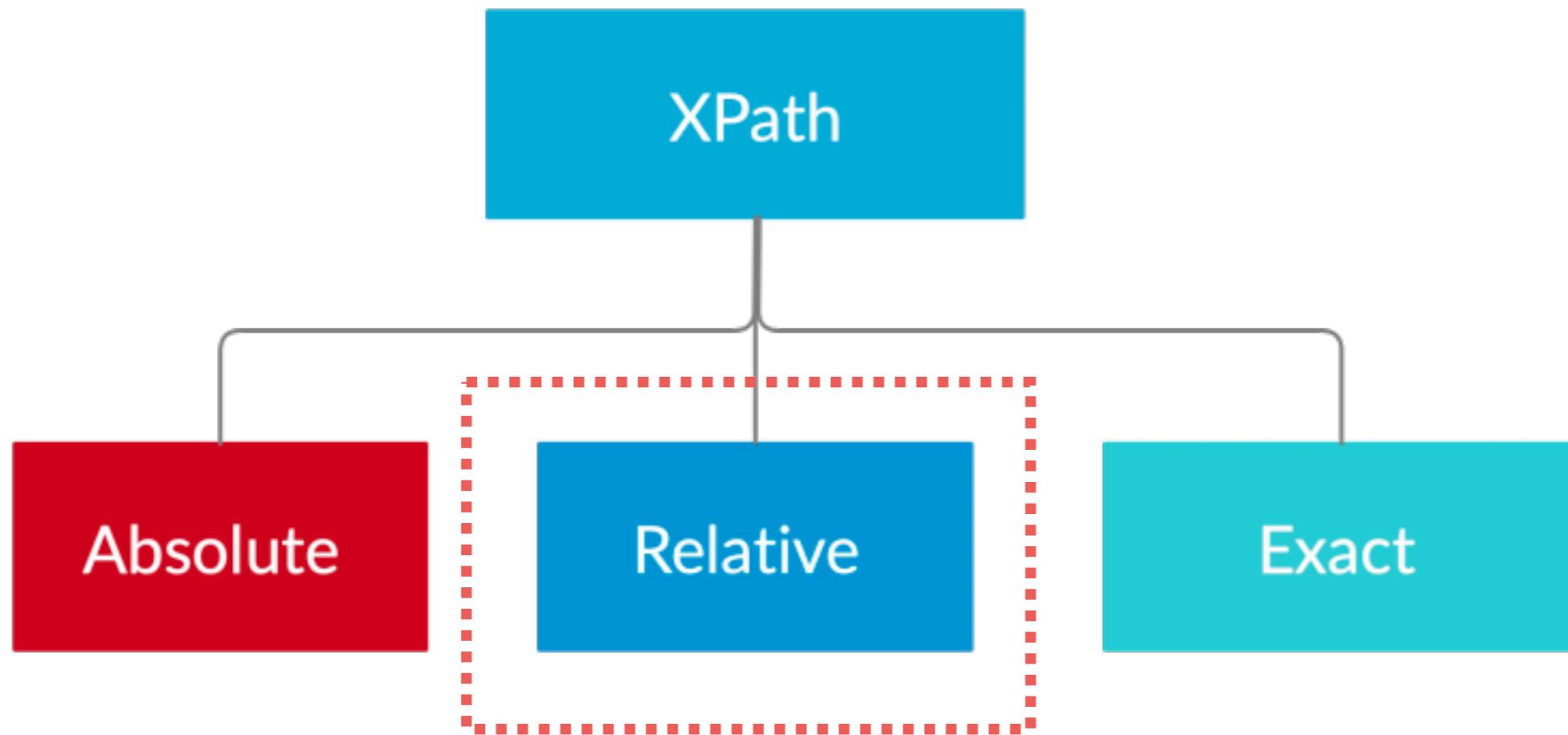
Don't use !!



`/html/body/div[1]/div/div[2]/form/div[2]/input`



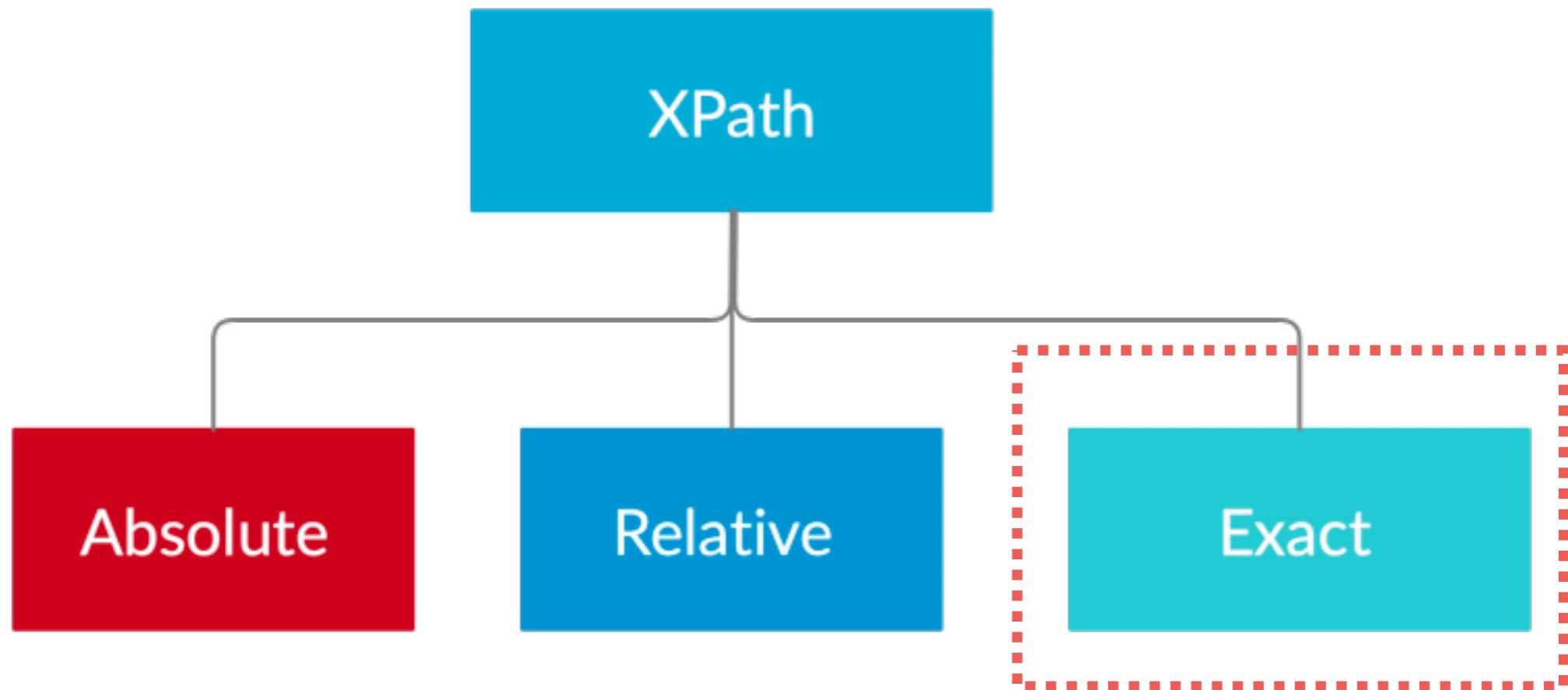
Relative



```
//div[@id='divUsername']/input  
//form/div[@id='divUsername']/input  
//form/*/input
```



Exact



```
//div[@class='datevalue currmonth']//span[./text()='2']
```



**Absolute faster than Relative
But shortest is better**



Element Interactions



Element Interactions

| Command | Description |
|------------|---|
| click() | Tab an element |
| sendKeys() | Enter keystrokes into an input field |
| clear() | Clear an input field |
| getText() | Retrieve the text displayed from a field or label |

<http://appium.io/docs/en/commands/element/find-elements/index.html#selector-strategies>



Waiting for Elements



Waiting ...

Static waits (sleep) !!

Explicit waits

Wait until the element is found before proceeding
the next step

Wait Until Element Is Visible

Wait Until Page Contains

Wait Until Page Contains Element

Wait Until Page Does Not Contain

Wait Until Page Does Not Contain Element



Learn to read Appium logs

```
[HTTP] {}
[AndroidUiautomator2Driver@0c13 (97cf7a81)] Driver proxy active, passing request on via HTTP proxy
[debug] [AndroidUiautomator2Driver@0c13 (97cf7a81)] Matched '/session/97cf7a81-37ce-4ac4-bd05-4d029612237e/source' to command name 'getPageSource'
[debug] [AndroidUiautomator2Driver@0c13 (97cf7a81)] Proxying [GET /session/97cf7a81-37ce-4ac4-bd05-4d029612237e/source] to [GET http://127.0.0.1:8200/session/fe20a07f-45ed-4502-91f5-05df40569d7e/source] with no body
[debug] [AndroidUiautomator2Driver@0c13 (97cf7a81)] Got response with status 200: {"sessionId":"fe20a07f-45ed-4502-91f5-05df40569d7e","value":"<?xml version='1.0' encoding='UTF-8' standalone='yes' ?>\r\n<hierarchy index=\"0\" class=\"hierarchy\" rotation=\"0\" width=\"1080\" height=\"1857\">\r\n  <android.widget.FrameLayout index=\"0\" package=\"com.amazonaws.devicefarm.android.referenceapp\" class=\"android.widget.FrameLayout\" text=\"\" checkable=\"false\" checked=\"false\" clickable=\"false\" enabled=\"true\" focusable=\"false\" focused=\"false\" long-clickable=\"false\" password=\"false\" scrollable=\"false\" selected=\"false\" bounds=\"[0,0][1080,1857]\" displayed=\"true\">\r\n    <android.widget.LinearLayout index=\"0\" package=\"com.amazonaws.devicefarm.android.referenceapp\" class=\"android.widget.LinearLayout\" text=\"\" checkable=\"false\" checked=\"false\" clickable=\"false\" enabled=\"true\" focusable=\"false\" focused=\"false\" long-clickable=\"false\" password=\"false\" scrollable=\"false\" selected=\"false\" bounds=\"[0,0][1080,1857]\" displayed=\"true\">\r\n      <androi...
[AndroidUiautomator2Driver@0c13 (97cf7a81)] Replacing sessionId fe20a07f-45ed-4502-91f5-05df40569d7e with 97cf7a81-37ce-4ac4-bd05-4d029612237e
[HTTP] <-- GET /session/97cf7a81-37ce-4ac4-bd05-4d029612237e/source 200 51 ms - 26619
[HTTP]
[HTTP] --> POST /session/97cf7a81-37ce-4ac4-bd05-4d029612237e/element
```

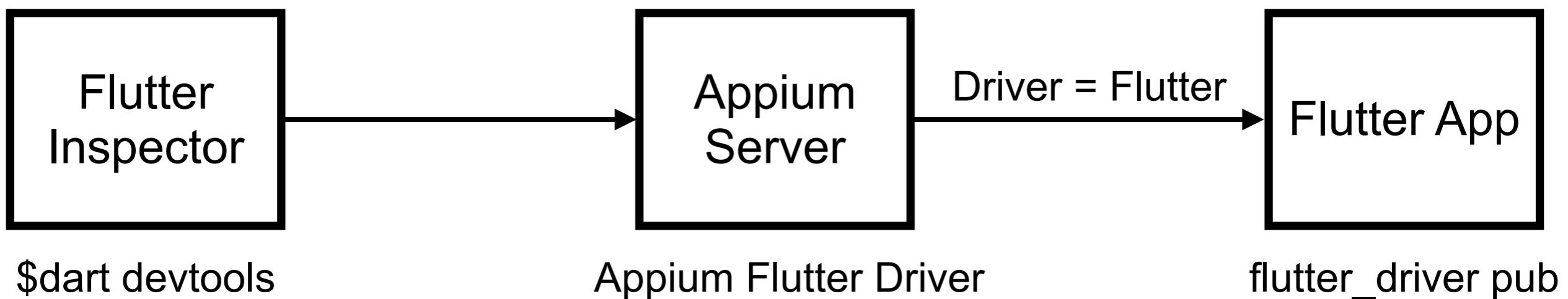


Appium for Flutter App



Appium with Flutter

Install via npm (required Node.js)



<https://github.com/appium/appium-flutter-driver>



Appium Server

Install Appium Flutter Driver

```
$appium driver install  
--source=npm appium-flutter-driver
```

<https://github.com/appium/appium-flutter-driver>



List of Appium driver

\$appium driver list

```
✓ Listing available drivers
- uiautomator2@2.29.2 [installed (npm)]
- flutter@2.0.0 [installed (npm)]
- xcuitest [not installed]
- mac2 [not installed]
- espresso [not installed]
- safari [not installed]
- gecko [not installed]
- chromium [not installed]
```

\$appium driver install xcuitest



Flutter App

pubspec.yaml

```
dev_dependencies:  
  flutter_driver:  
    sdk: flutter  
  
  flutter_test:  
    sdk: flutter
```

main.dart

```
import 'package:flutter_driver/flutter_driver_extension.dart';  
  
void main() {  
  
  enableFlutterDriverExtension();  
  
  runApp(DevicePreview(  
    enabled: !kReleaseMode,  
    builder: (context) => const MyApp(),  
  ));  
}
```

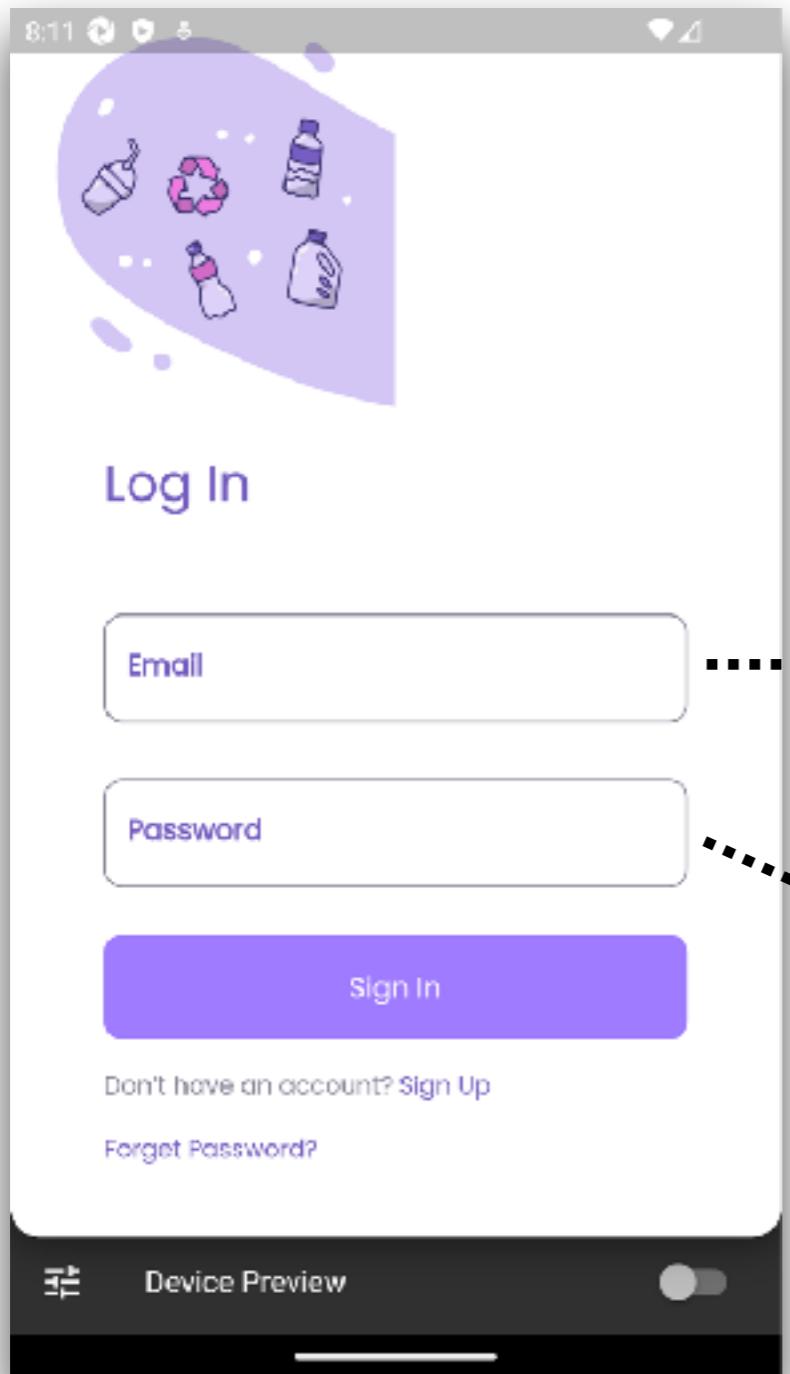


Flutter App Locator

Key (default)
Text
Semantics
Tooltip
Type



Flutter App Locator



```
TextField(  
  key: Key('email_key'),
```

```
TextField(  
  key: Key('password_key'),
```



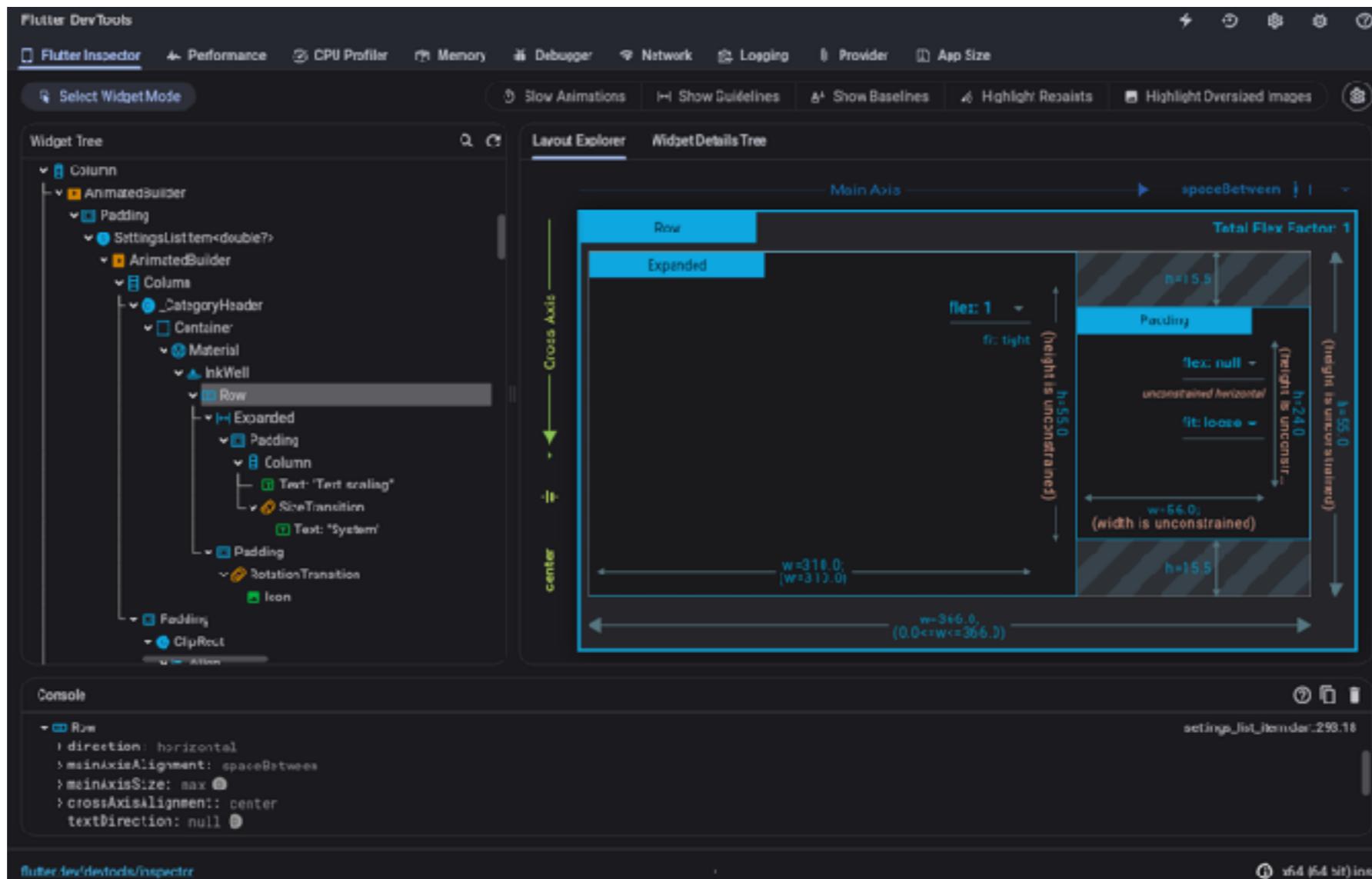
Start Flutter App

\$flutter run



Flutter Inspector

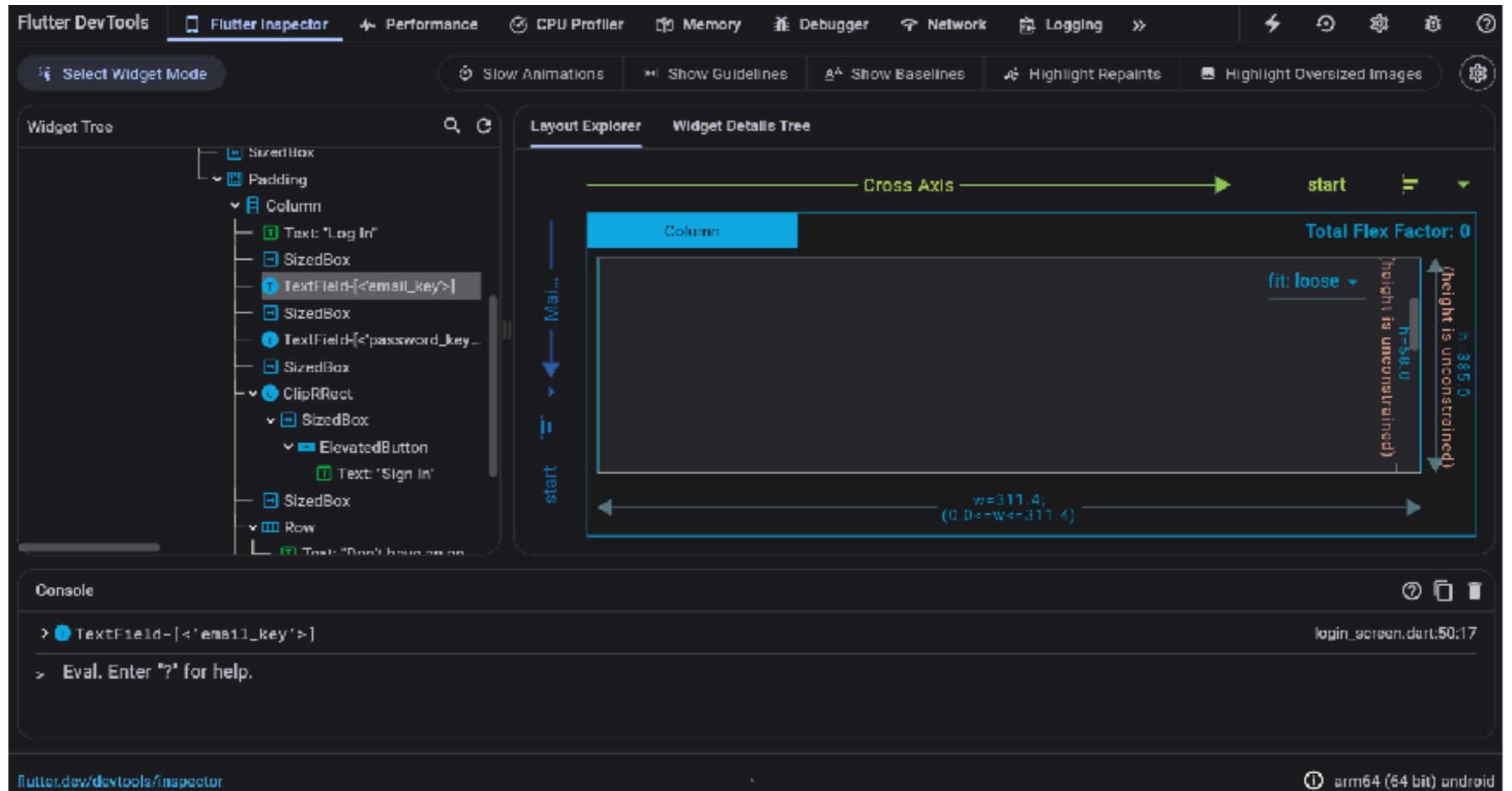
\$dart devtools



<https://docs.flutter.dev/tools/devtools/inspector>



Flutter Inspector



AppiumFlutter Library

Working with Robot framework

```
$pip install -U robotframework-appiumflutterlibrary
```

<https://github.com/igortavtib/robotframework-appiumflutterlibrary>



Write test case

*** Settings ***

Library AppiumFlutterLibrary

*** Variables ***

```
 ${REMOTE_URL}    http://127.0.0.1:4723
 ${platformName}   Android
 ${app}           ${CURDIR}/../../demo-app/flutter.apk
 ${deviceName}     id/name
 ${automationName} flutter
```

*** Test Cases ***

Test case name

```
  Open Application    ${REMOTE_URL}
    ...  platformName=${platformName}
    ...  app=${app}  deviceName=${deviceName}
    ...  automationName=${automationName}
```

```
  Wait For Element  key=email_key
  Click Element    key=email_key
  Input Text       key=email_key  demo@admin.com
```

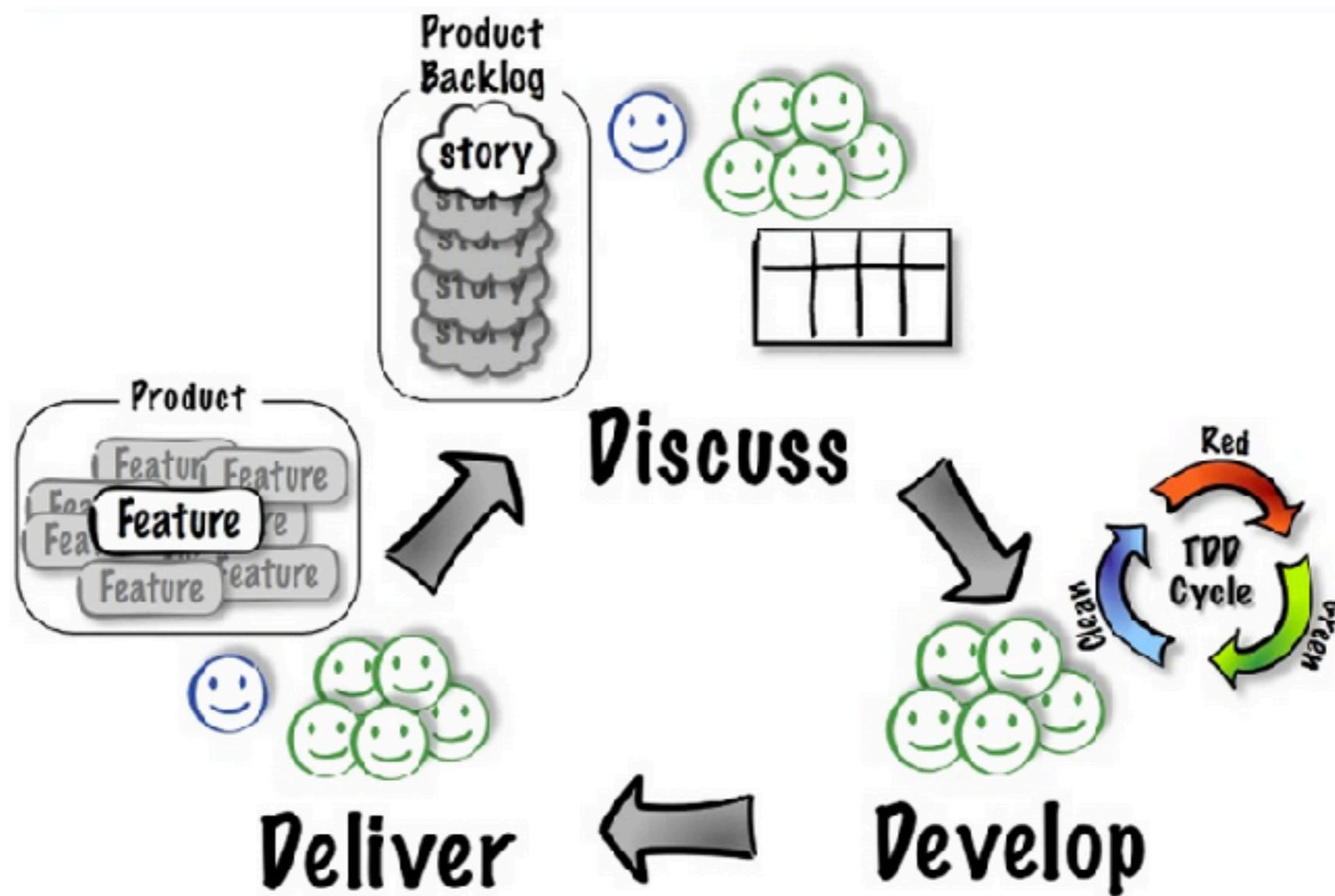
<https://igortavtib.github.io/robotframework-appiumflutterlibrary/AppiumFlutterLibrary.html>



THINK before coding



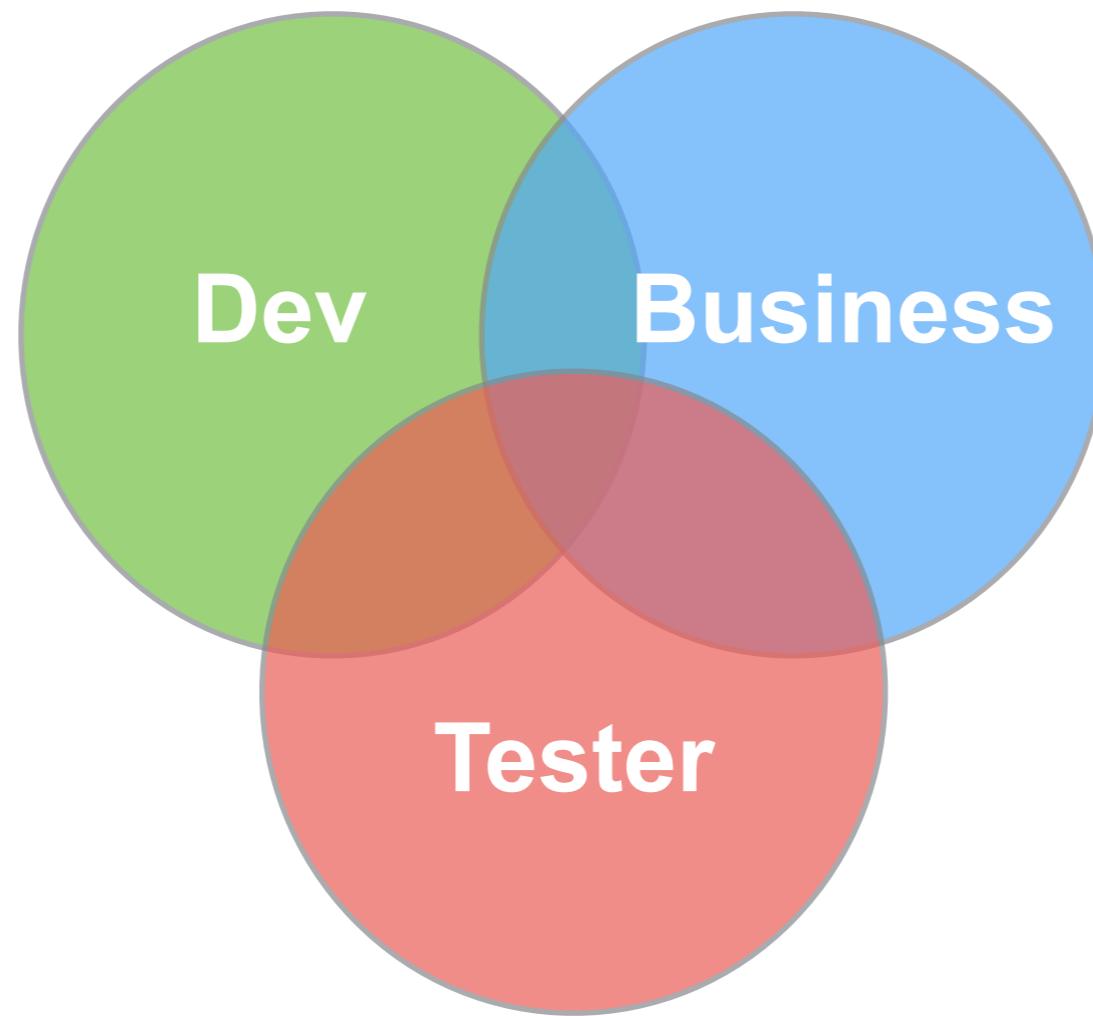
Acceptance Test-Driven Development



(Model developed with Pekka Klärck, Bas Vodde, and Craig Larman.)



Acceptance Test-Driven Development



Acceptance Tests

=

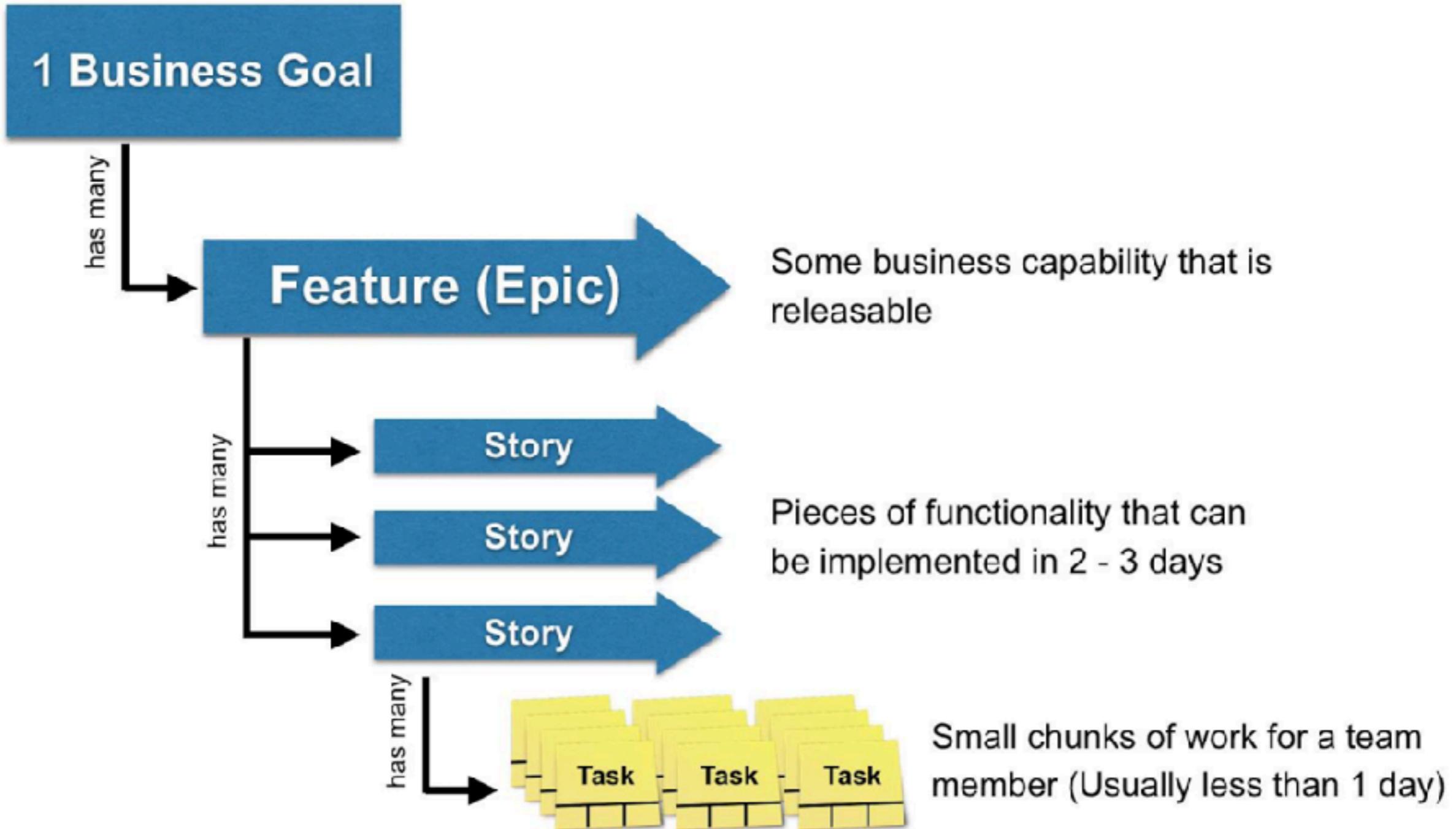
Business Criteria

+

Examples (data)



Work break down



Iterative and incremental process

Feature 1

Time



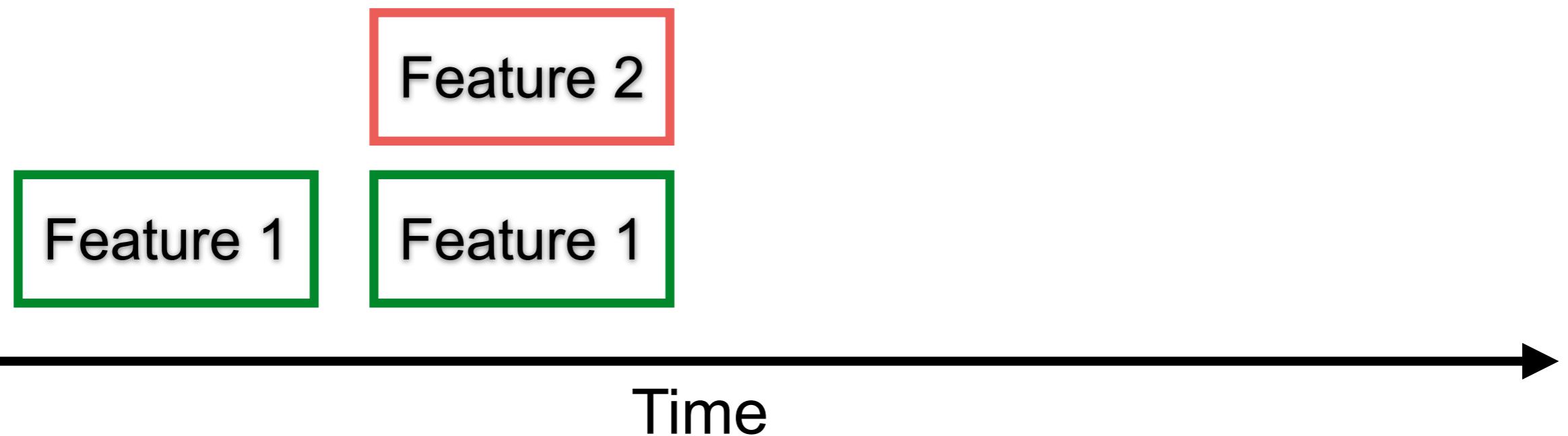
Iterative and incremental process

Done = coded and tested



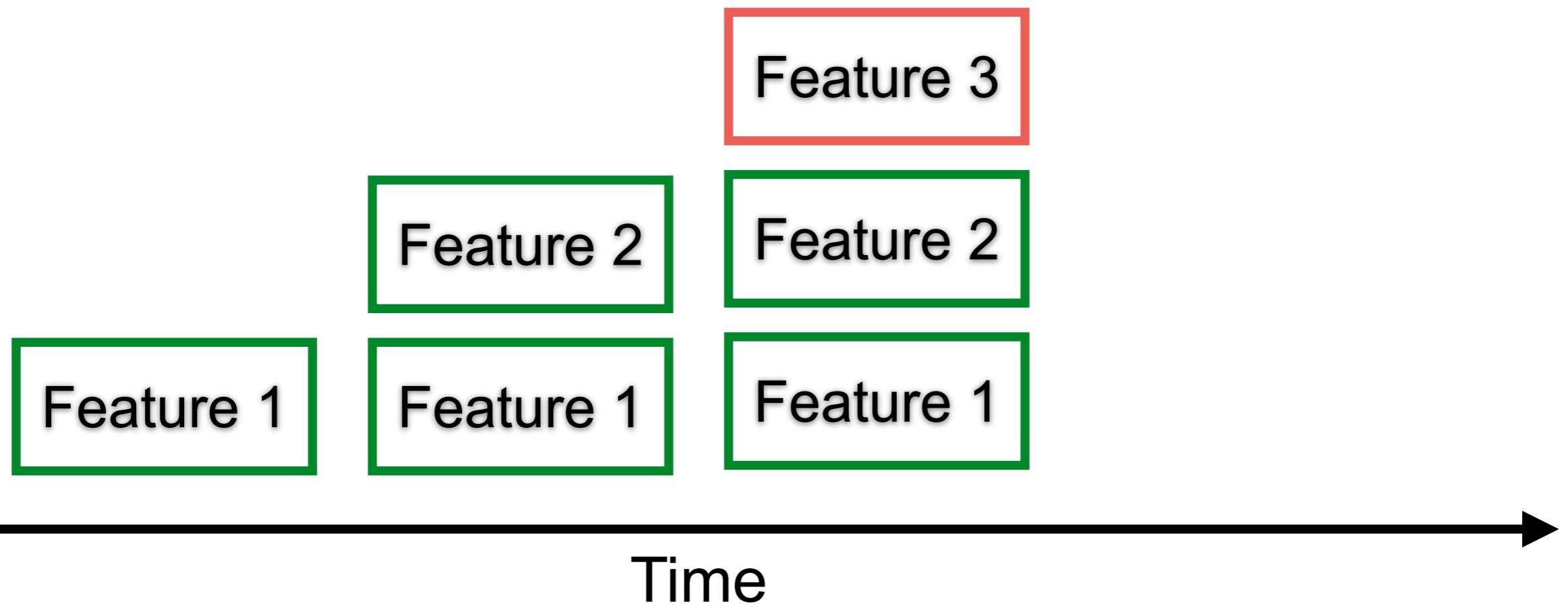
Iterative and incremental process

Done = coded and tested



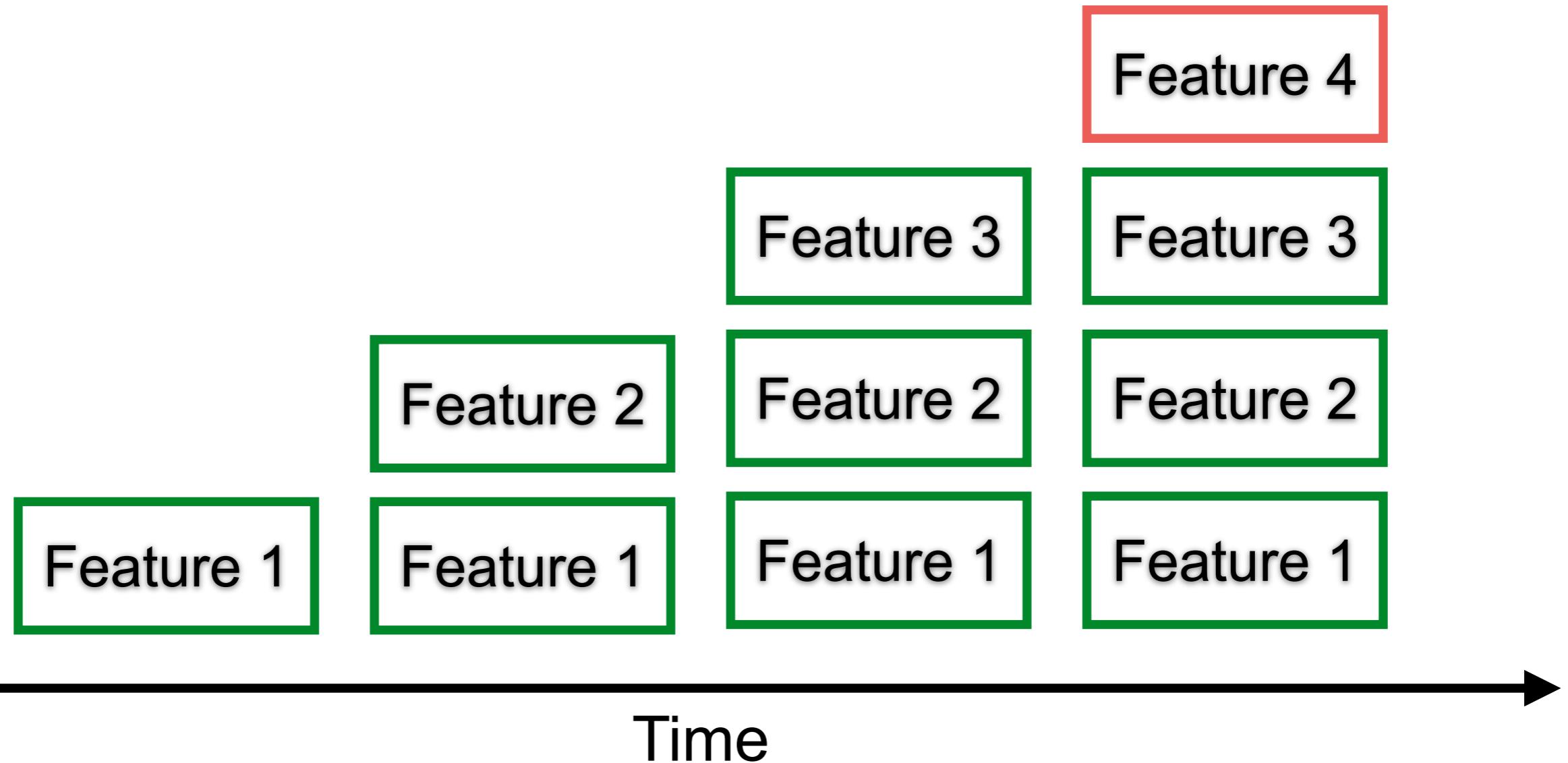
Iterative and incremental process

Done = coded and tested



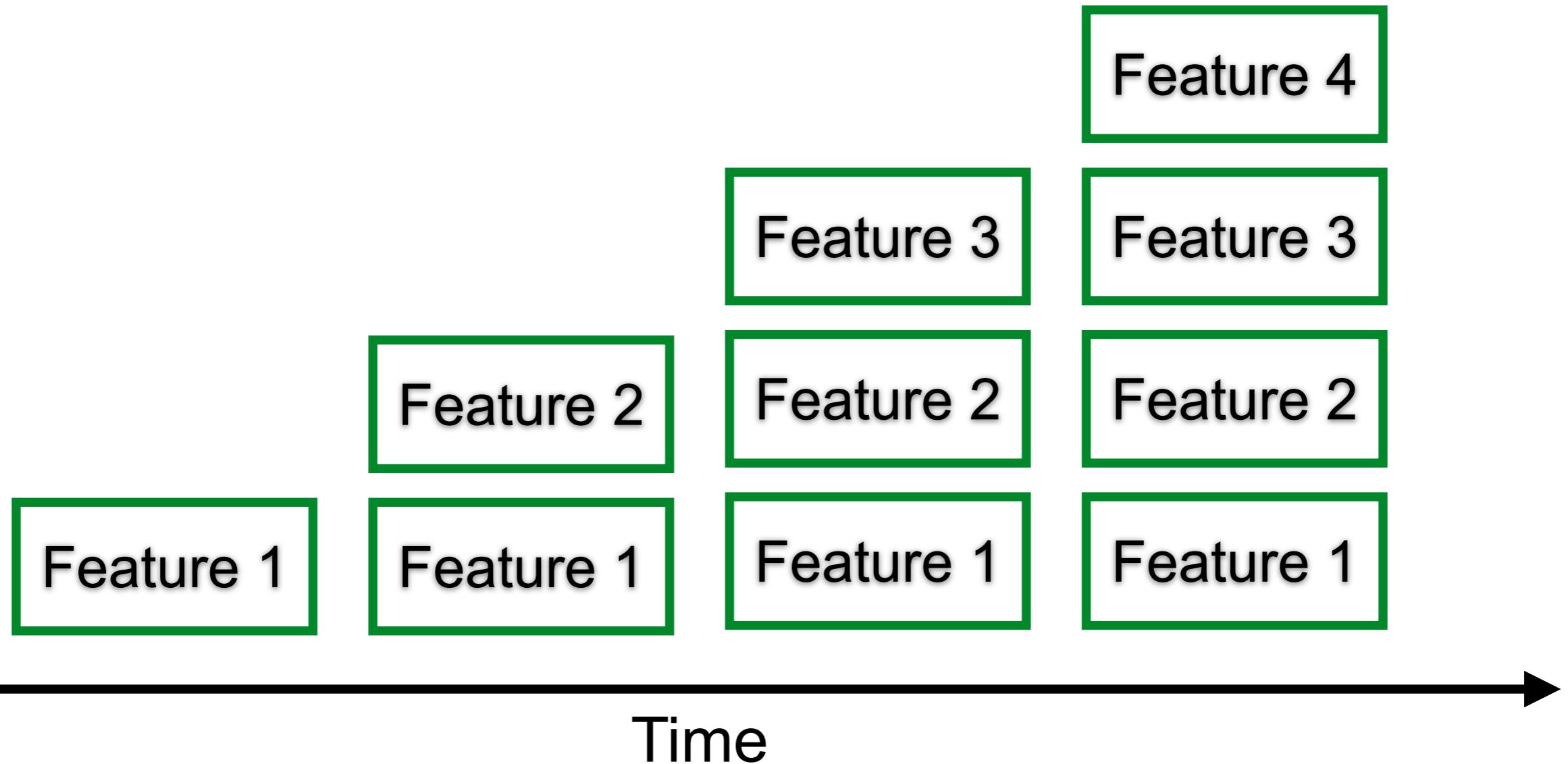
Iterative and incremental process

Done = coded and tested



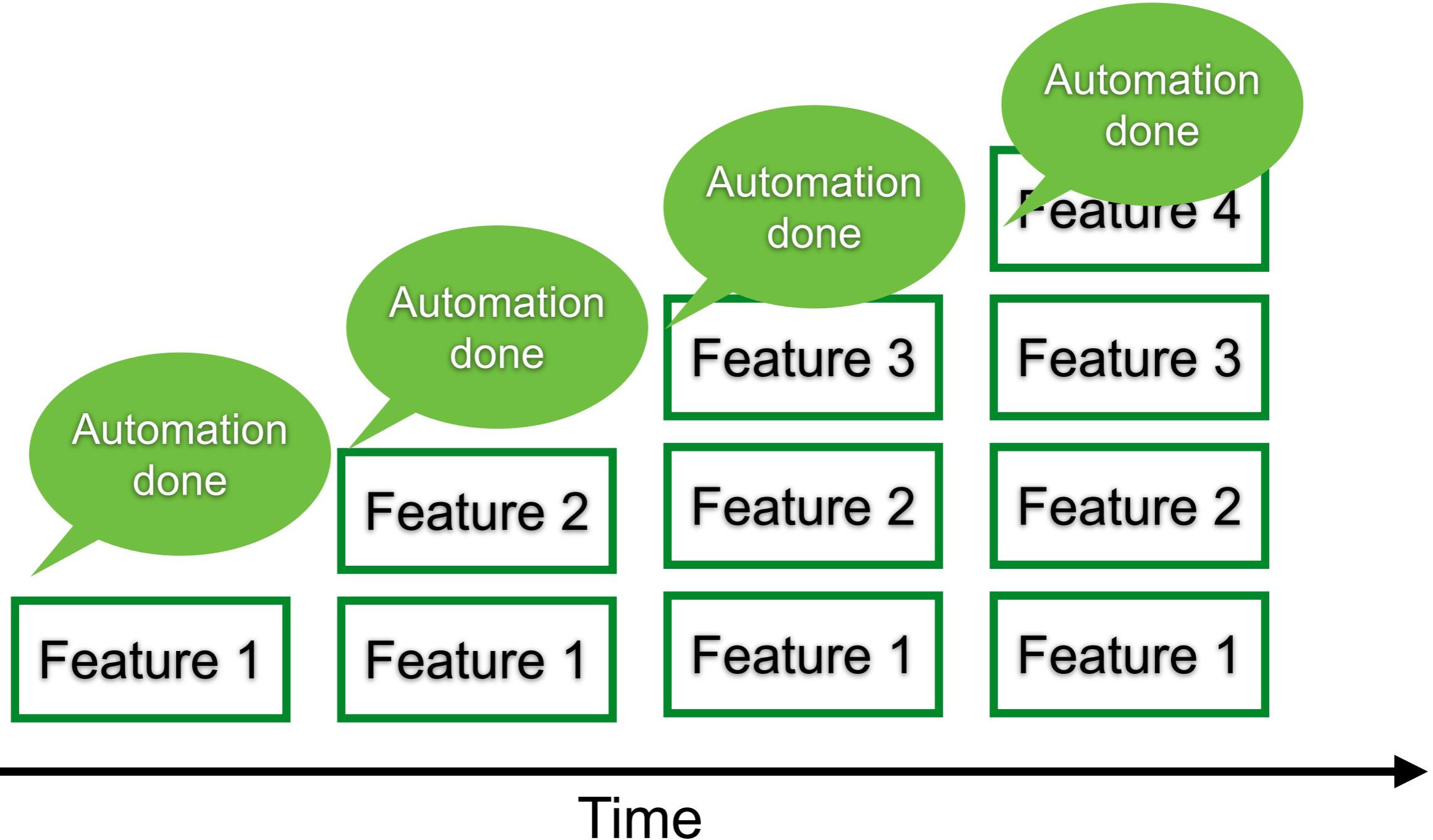
Iterative and incremental process

Done = coded and tested



Iterative and incremental process

Done = coded and tested



Automation feedback

Easier over time ?

Time spent on maintenance ?

Test find regression bugs ?



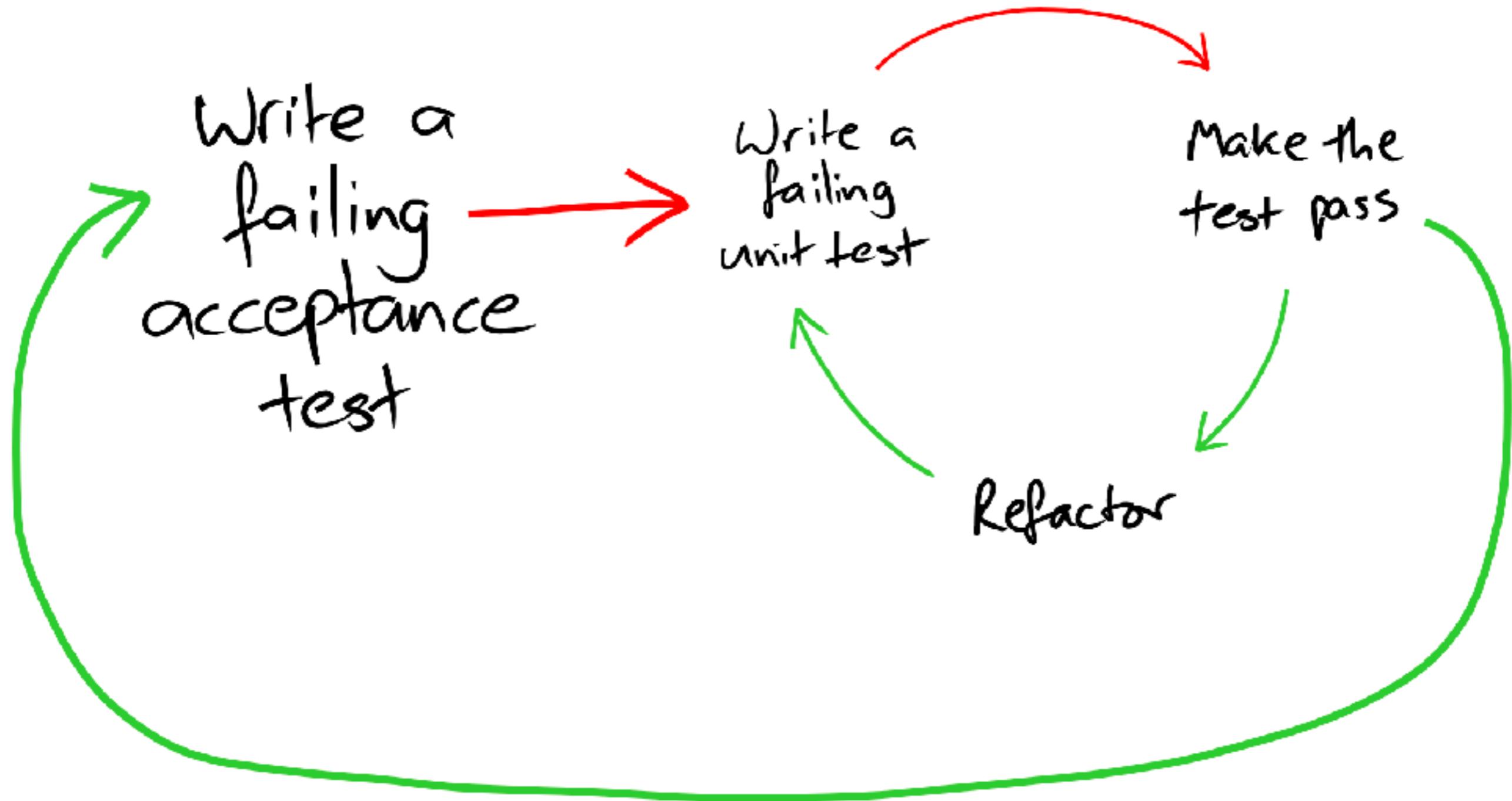
Start with simple



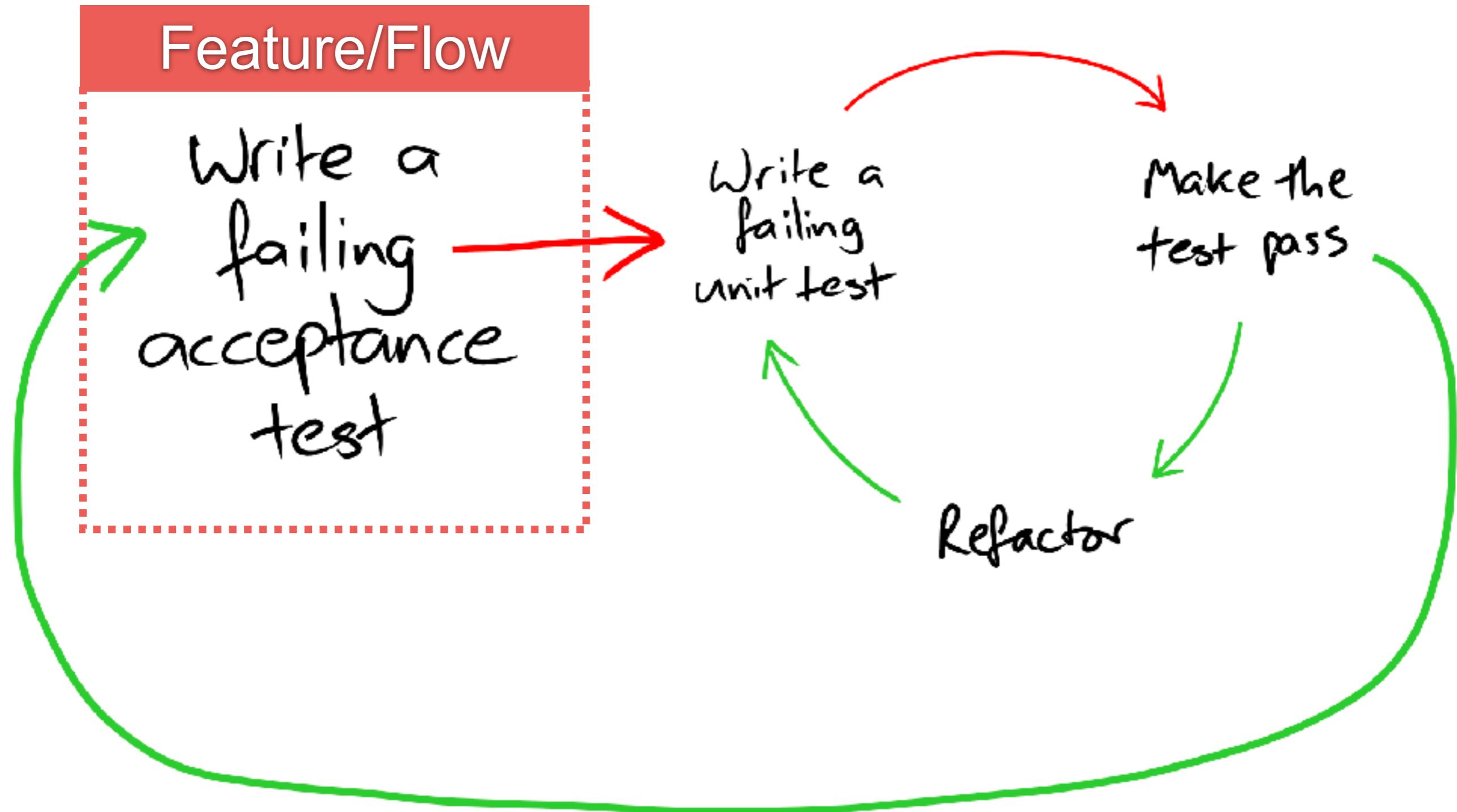
Use **feedback** to improve



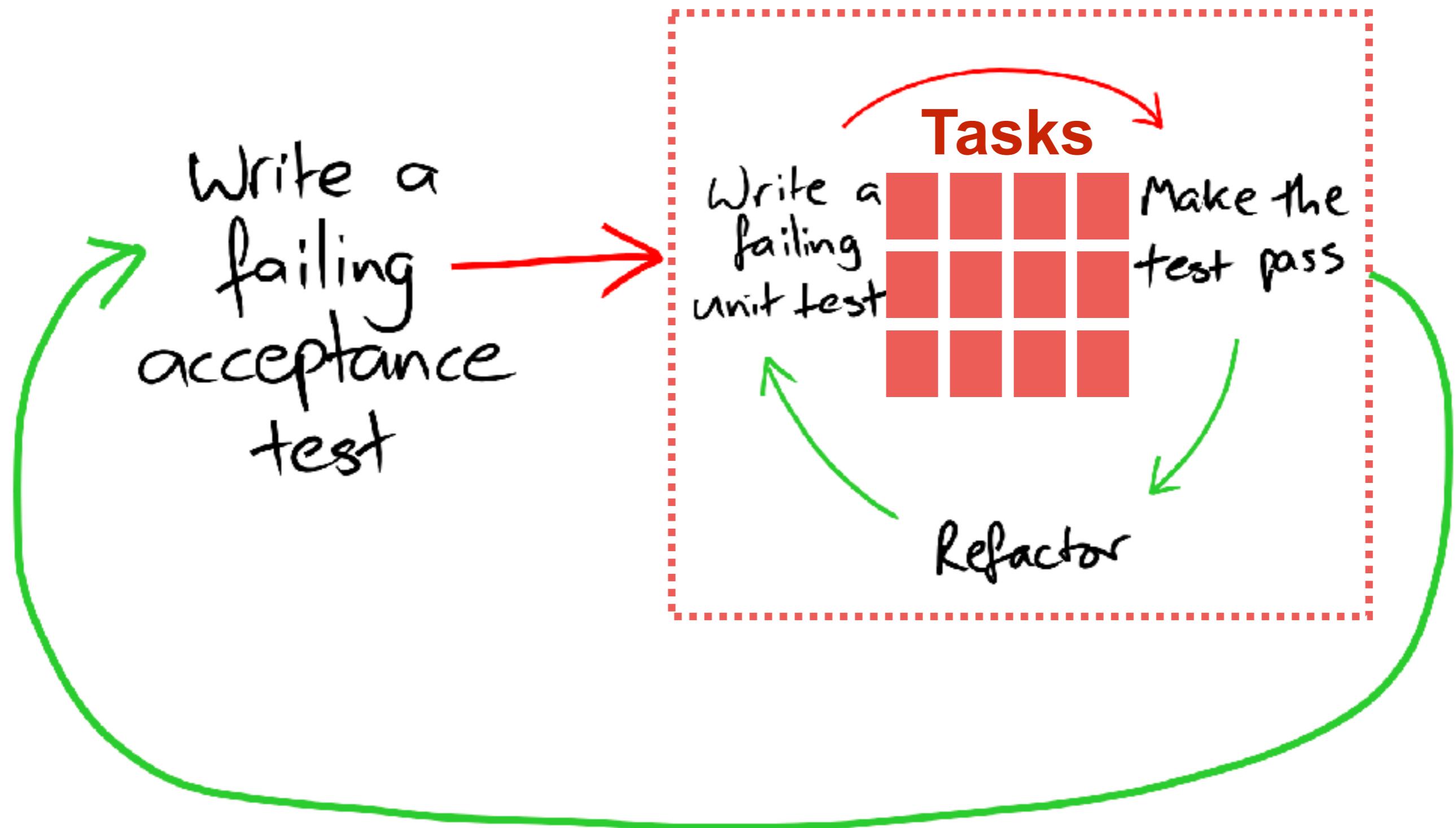
Outside-in develop/test



Outside-in develop/test



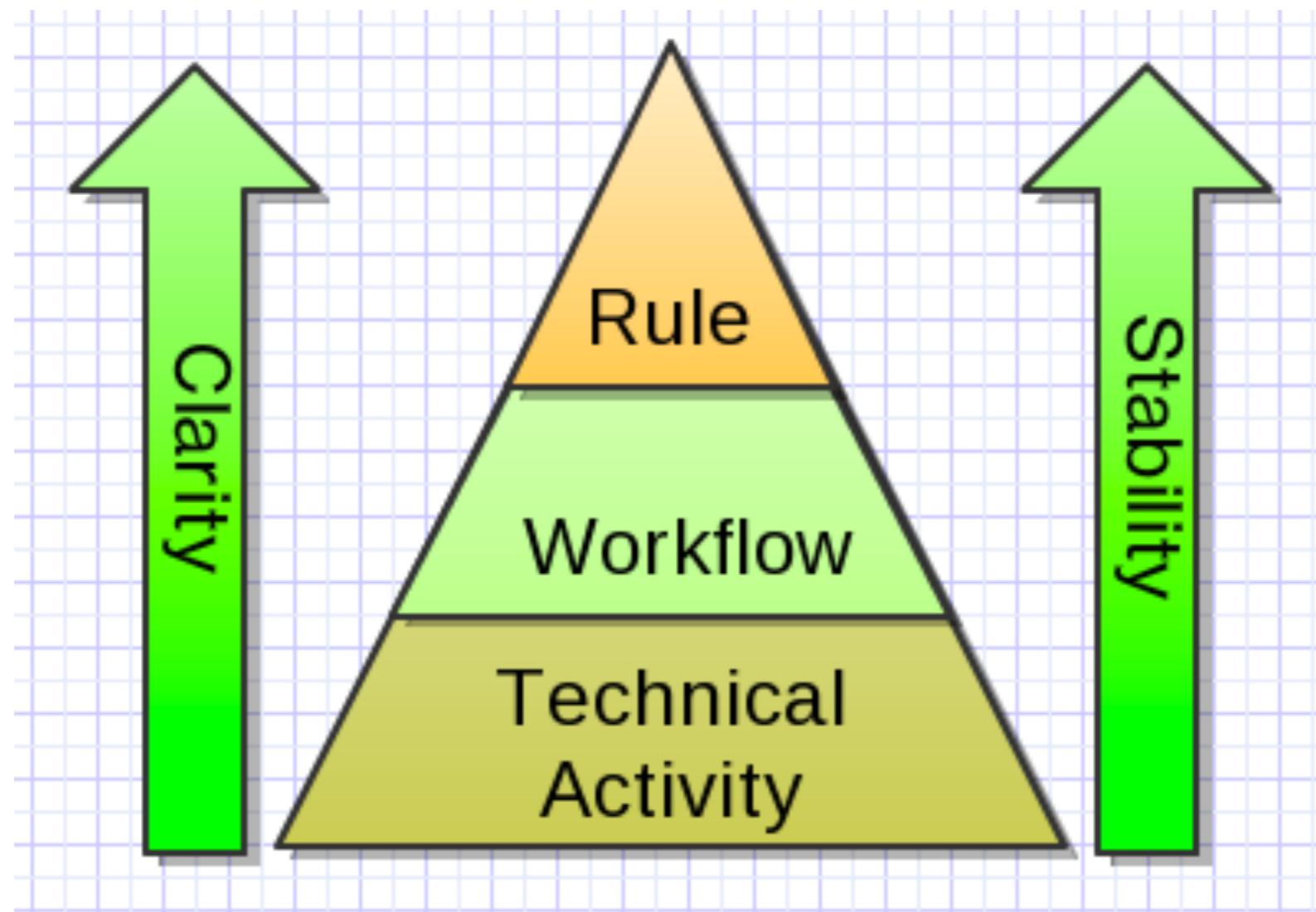
Outside-in develop/test



UI Test Automation



3 levels of UI test automation



3 levels of UI test automation

Business rule/functionality level

what is this test demonstrating or exercising

User interface workflow level

what does a user have to do to exercise the functionality through the UI

Technical activity level

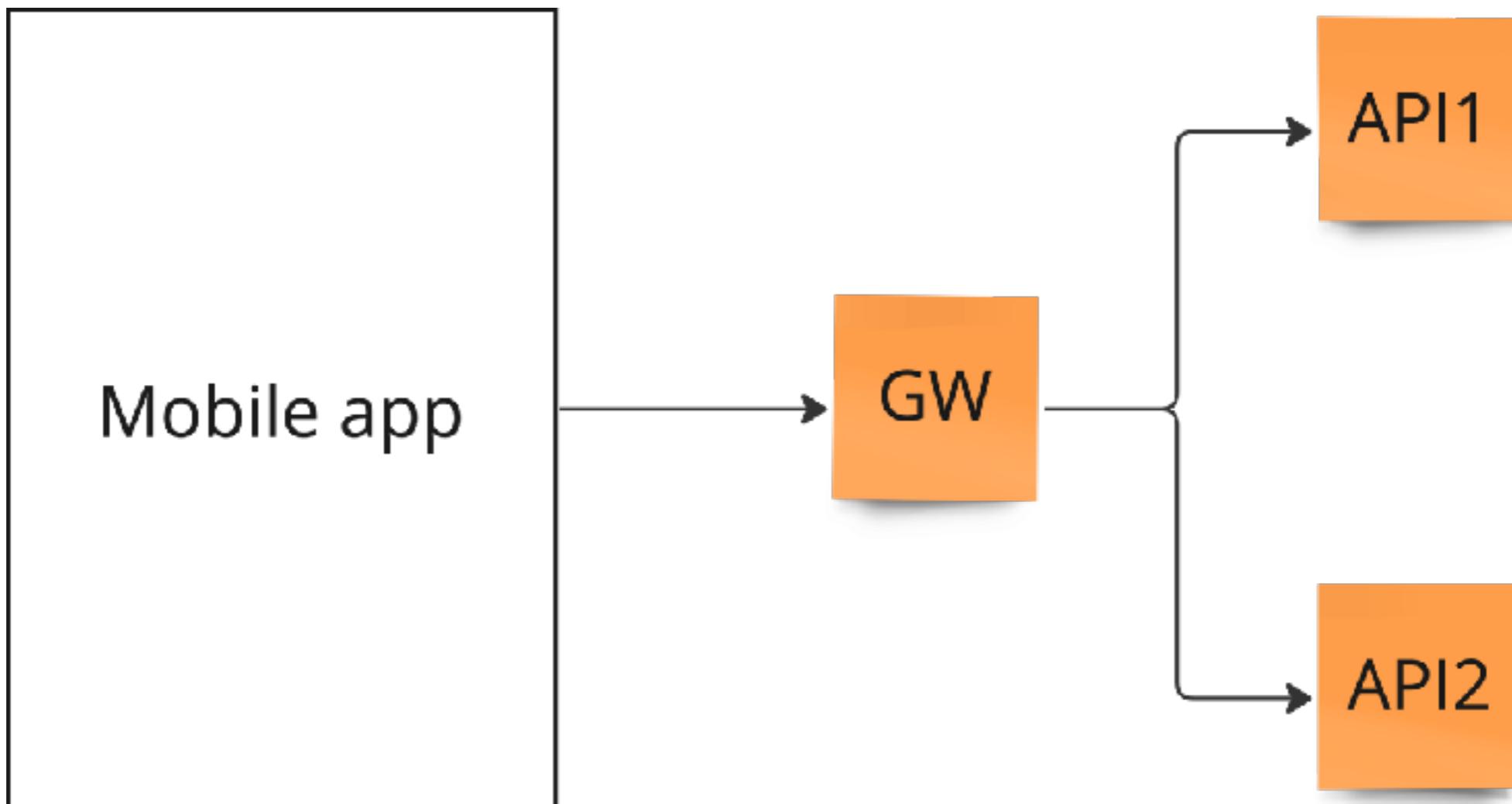
what are the technical steps required to exercise the functionality



Architecture !!



Architecture of System



Test strategy ?

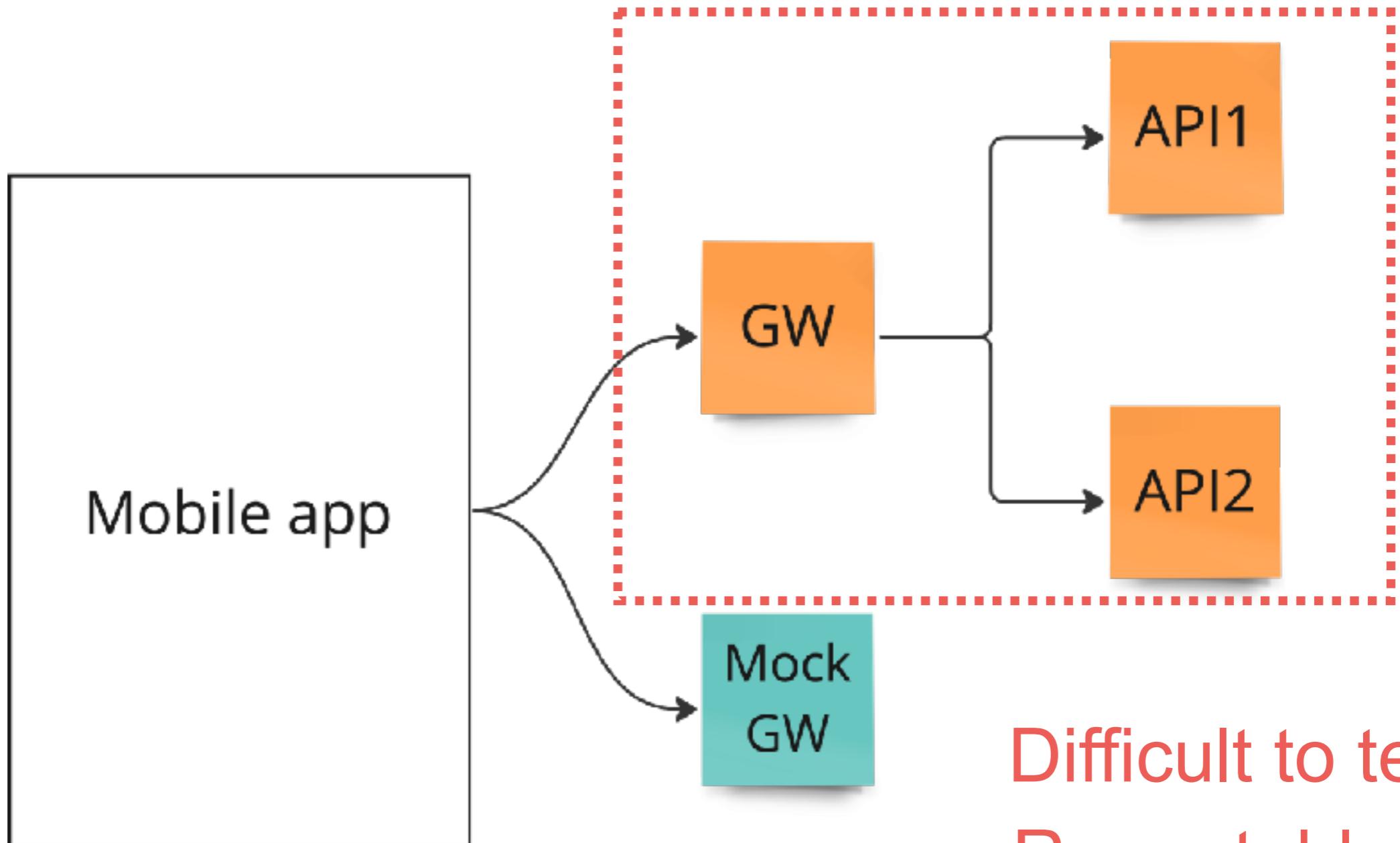


Good Tests ?

Fast
Isolate
Repeatable
Self-verify
Timely



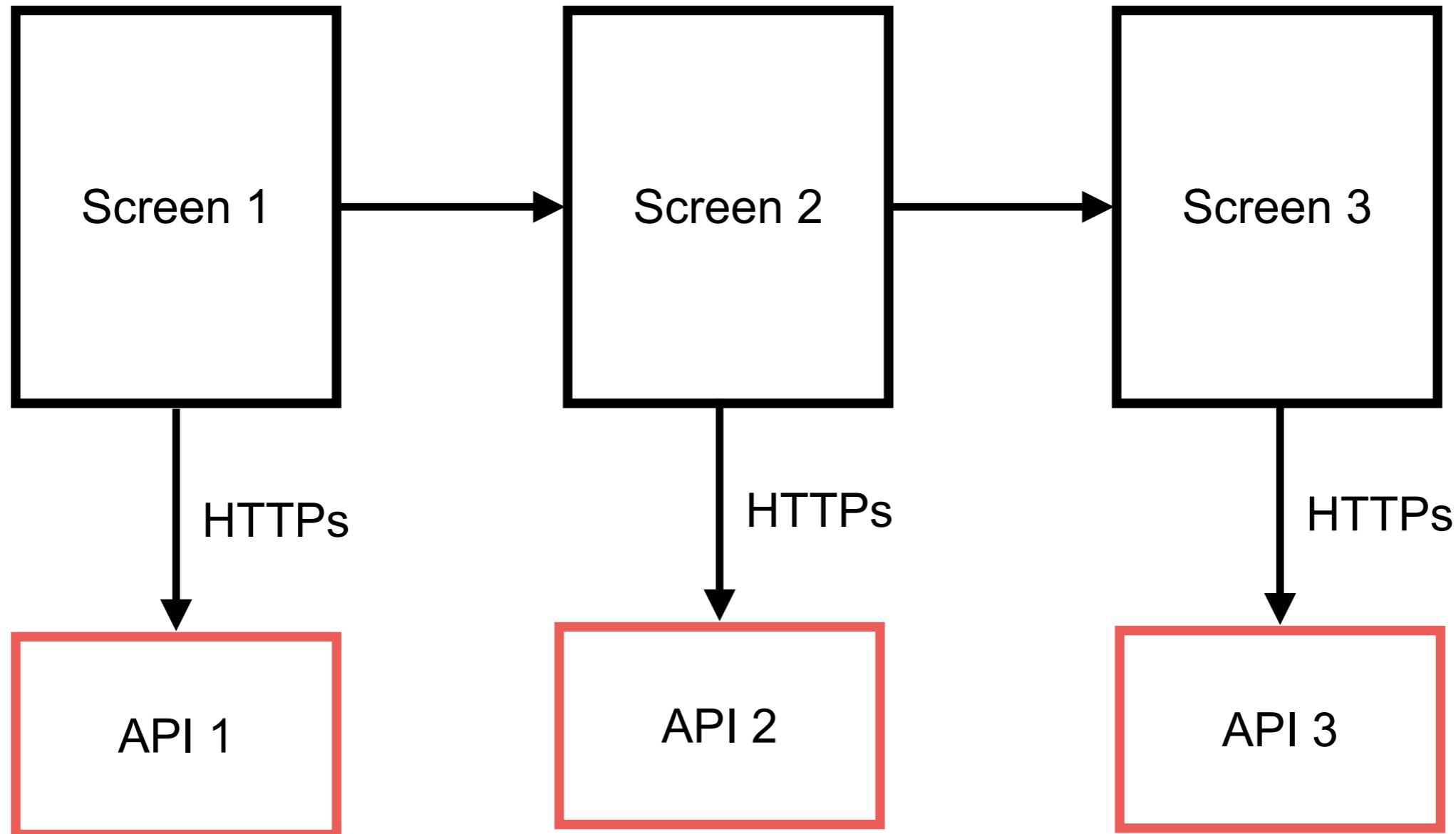
Integration Testing



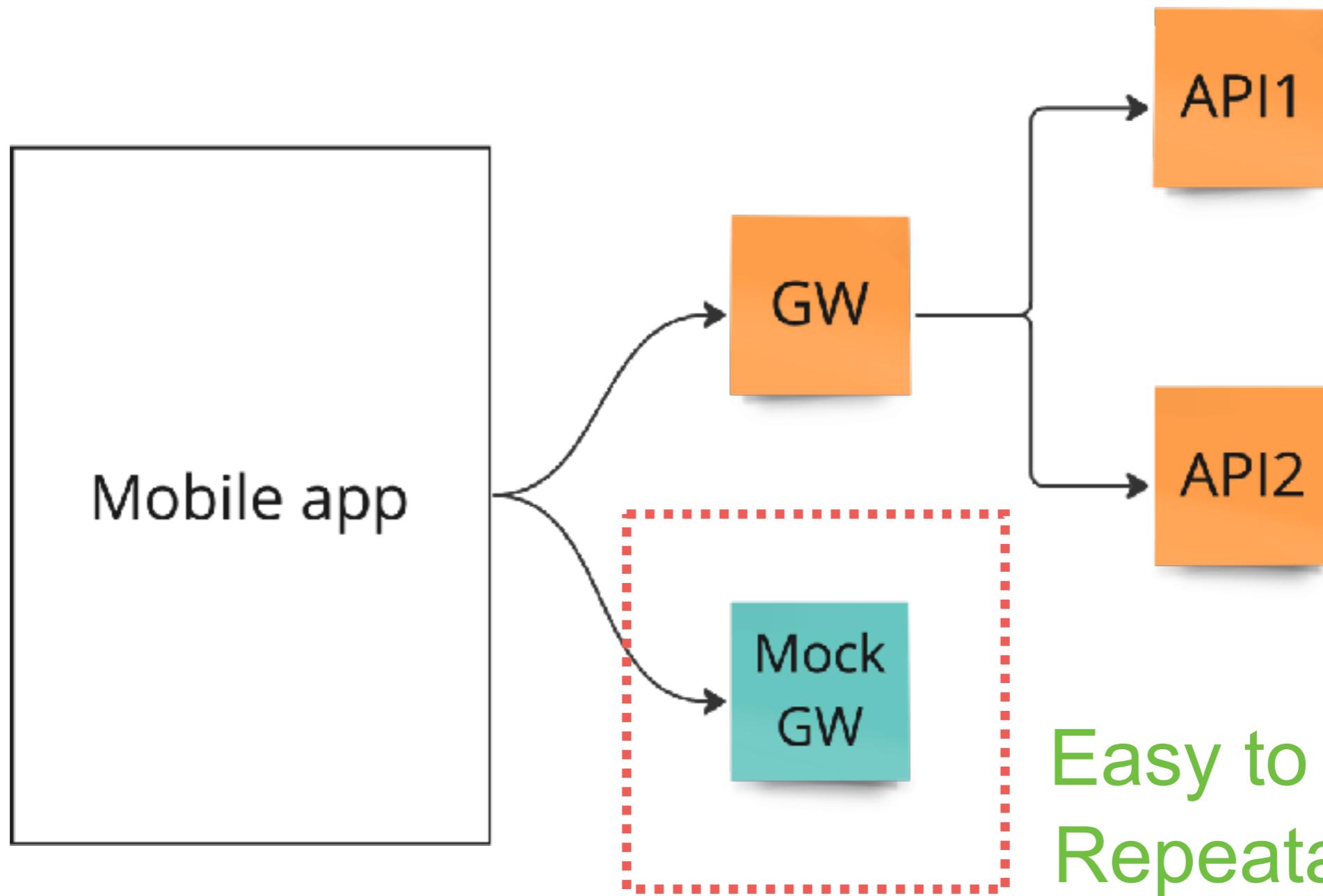
Difficult to test
Repeatable
Reliable



Integration Testing



Component Testing



Easy to test
Repeatable
Reliable



Mock Server (HTTPs)

Stubby4Node

<https://github.com/mrak/stubby4node>

MountBank

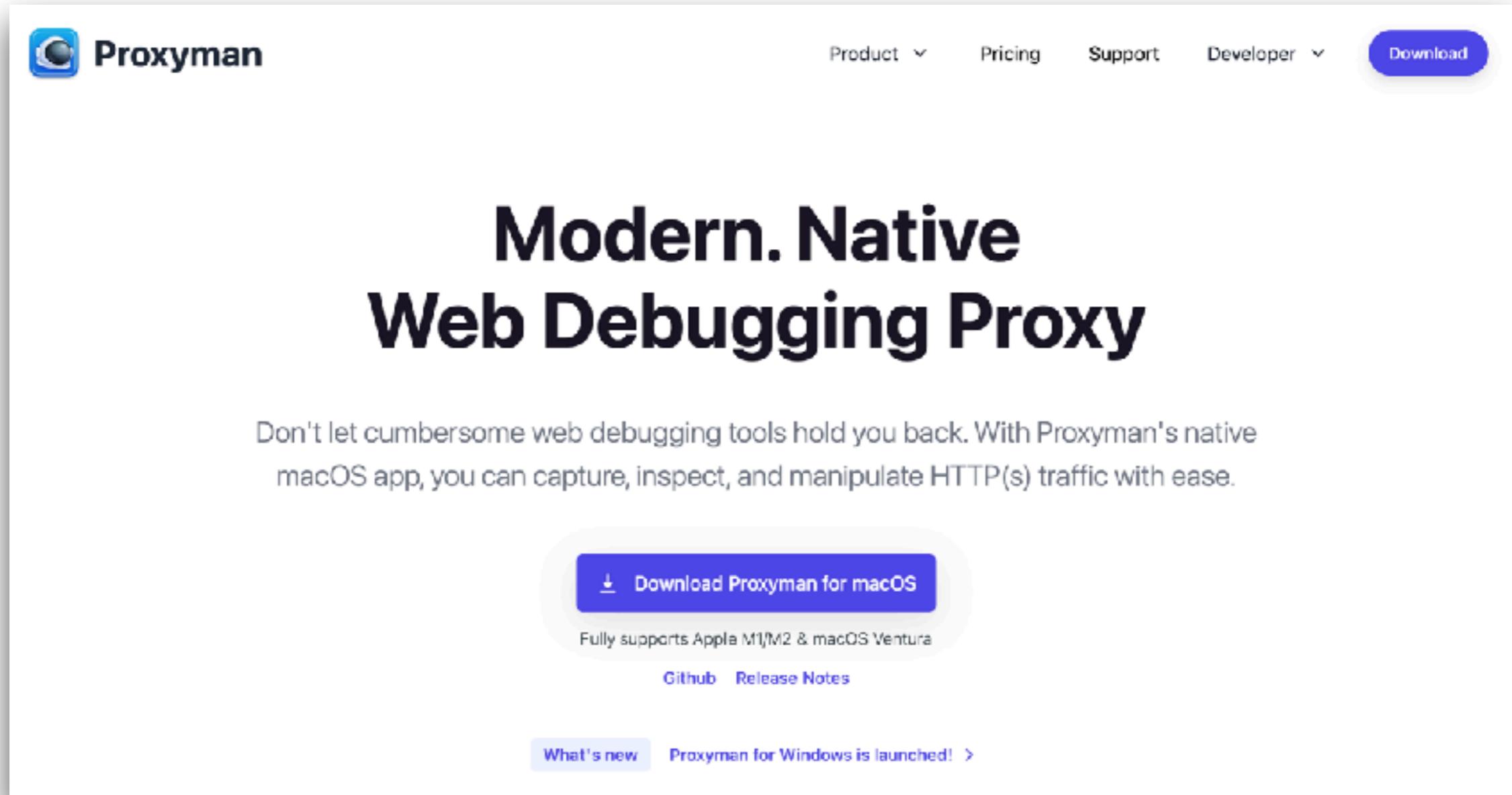
<https://www.mbtest.org/>

WireMock

<https://wiremock.org/>



Capture traffic from Mobile app

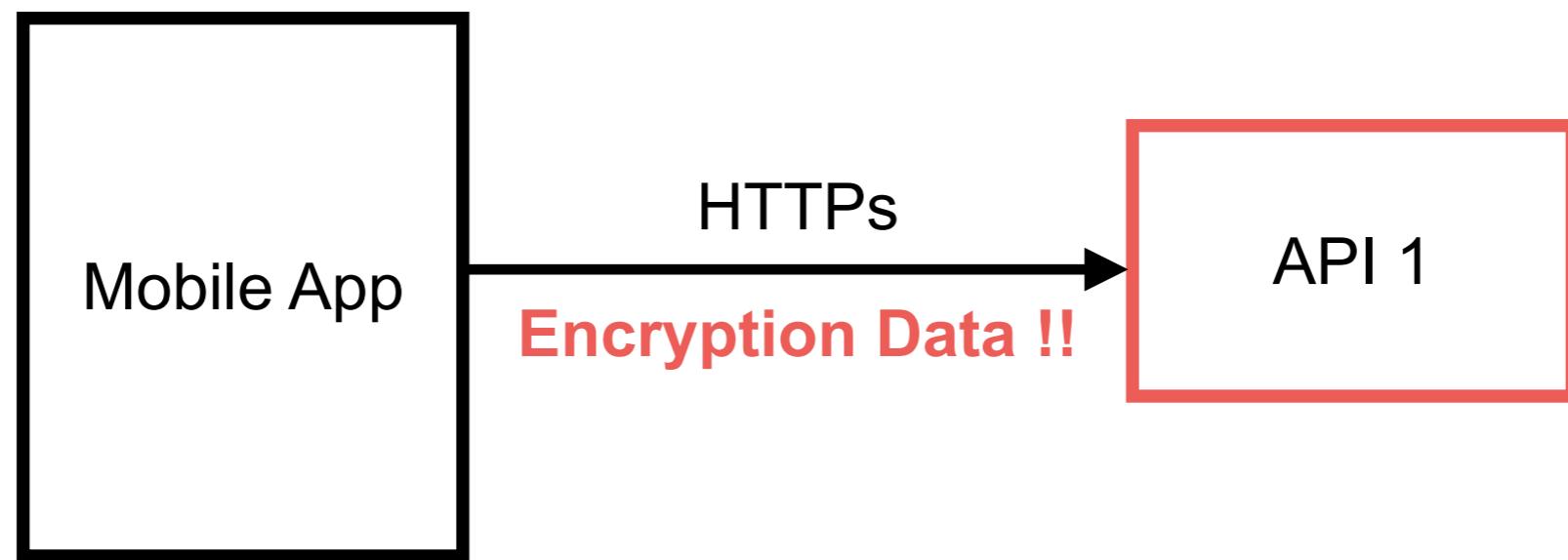


The screenshot shows the Proxyman website homepage. At the top left is the Proxyman logo (a blue square with a white gear icon). To its right are navigation links: Product (with a dropdown arrow), Pricing, Support, Developer (with a dropdown arrow), and a blue "Download" button. Below the navigation is a large, bold title: "Modern. Native Web Debugging Proxy". Underneath the title is a descriptive paragraph: "Don't let cumbersome web debugging tools hold you back. With Proxyman's native macOS app, you can capture, inspect, and manipulate HTTP(s) traffic with ease." Below this text is a prominent blue "Download Proxyman for macOS" button with a download icon. Underneath the button, it says "Fully supports Apple M1/M2 & macOS Ventura". Below that are links to "Github" and "Release Notes". At the bottom of the main content area are two small buttons: "What's new" and "Proxyman for Windows is launched! >".

<https://proxyman.io/>



Encryption Data !!



Manage App's States

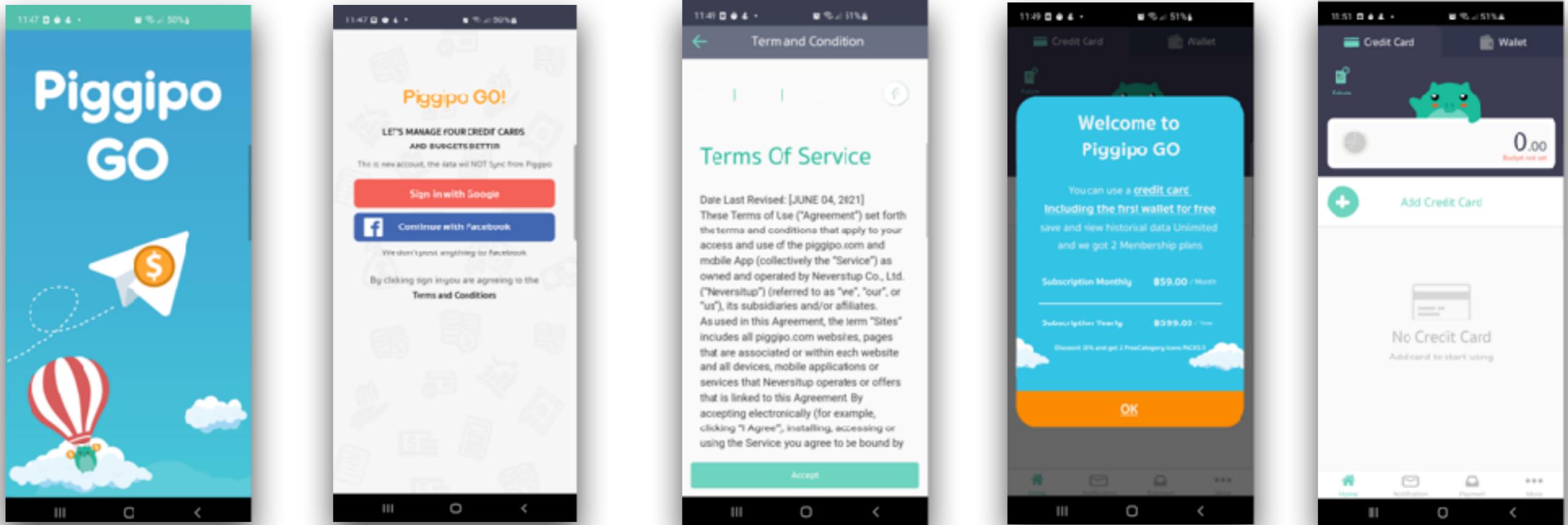


Manage App's States

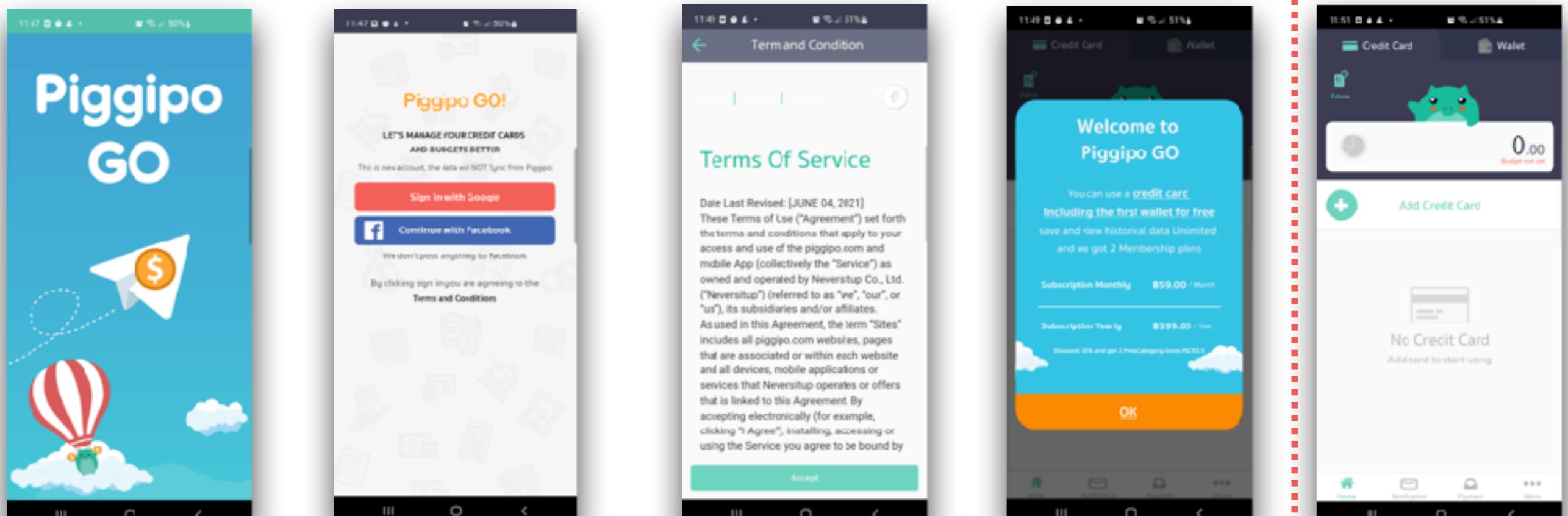
Easy for test
Reduce test steps



Manage App's States



Manage App's States



Manage App's States

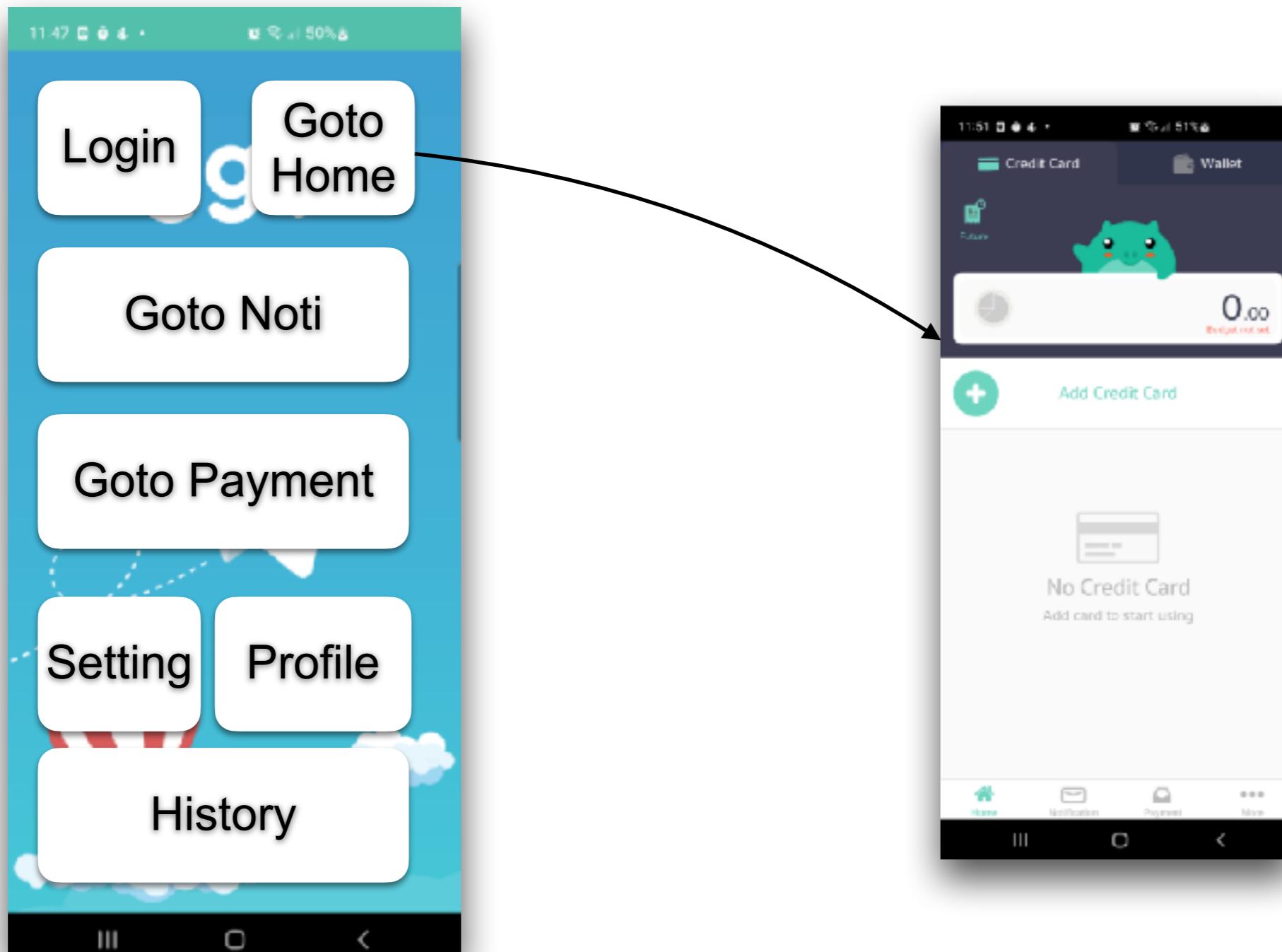
Use only in test build of the app

Screen with on-click link to all area of app

Link can even populate data



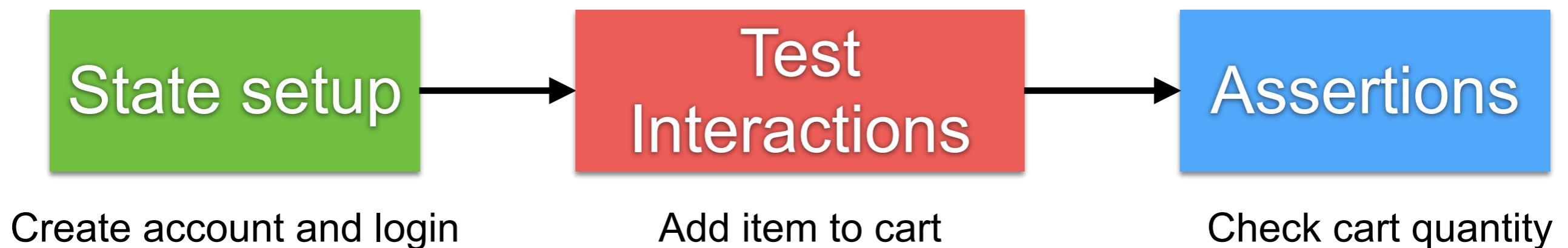
Manage App's States



State vs. Assertions

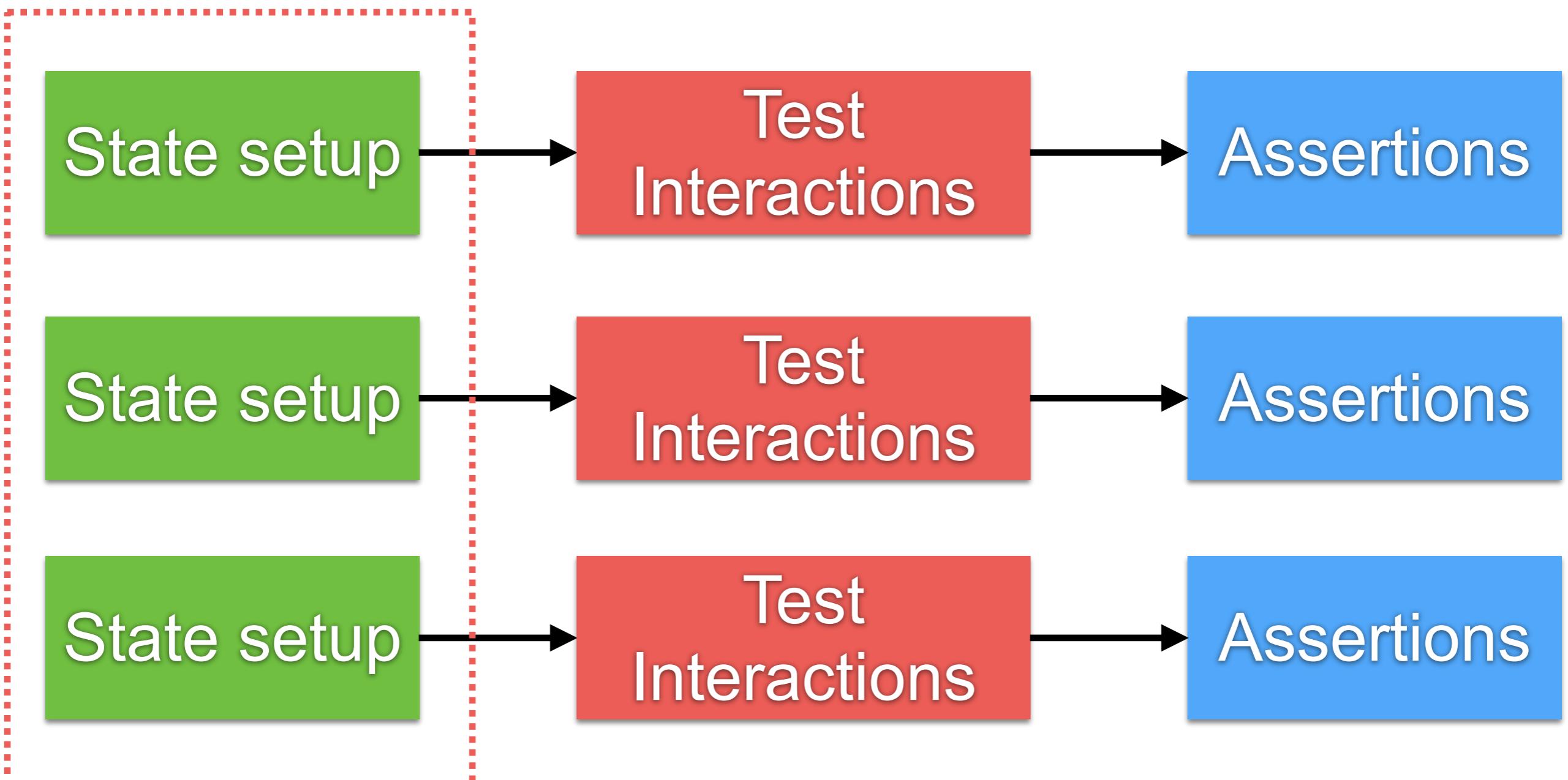


State vs. Assertions



State vs. Assertions

Not too much time !!



Working with Robot Framework



<https://robotframework.org/>





<https://github.com/serhatbolsu/robotframework-appiumlibrary>



Install Appium Library

```
$pip install -U robotframework
```

```
$pip install -U robotframework-appiumlibrary
```

<https://robotframework.org/>



Appium Library keywords

AppiumLibrary

Search X

Keywords (115) +

- Activate Application
- Background App
- Background Application
- Capture Page Screenshot
- Clear Text
- Click A Point
- Click Button
- Click Element
- Click Element At Coordinates
- Click Text
- Close All Applications
- Close Application
- Delete File
- Drag And Drop
- Element Attribute Should Match

Library version: 2.0.0
Library scope: GLOBAL

Introduction

AppiumLibrary is a Mobile App testing library for Robot Framework.

Locating or specifying elements

All keywords in AppiumLibrary that need to find an element on the page take an argument, either a `locator` or a `WebElement`. `locator` is a string that describes how to locate an element using a syntax specifying different location strategies. `WebElement` holds a `WebElement` instance, which is a representation of the element.

Using locators

By default, when a locator is provided, it is matched against the key attributes of the particular element. If the `name` attribute is `id` for all elements and locating elements is easy using just the `id`. For example:

```
Click Element    id=my_element
```

New in AppiumLibrary 1.4, `id` and `xpath` are not required to be specified, however `xpath` should start with `//` to indicate an absolute locator as explained below.

<https://serhatbolsu.github.io/robotframework-appiumlibrary/AppiumLibrary.html>



Locator Elements

| Strategy | Example | Description |
|------------------|--|-----------------------------------|
| identifier | Click Element / identifier=my_element | Matches by @id attribute |
| id | Click Element / id=my_element | Matches by @resource-id attribute |
| accessibility_id | Click Element / accessibility_id=button3 | Accessibility options utilize. |
| xpath | Click Element / xpath=//UITableView/UIATableCell/UIAButton | Matches with arbitrary XPath |
| class | Click Element / class=UIAPickerWheel | Matches by class |
| android | Click Element / android=UiSelector().description('Apps') | Matches by Android UI Automator |
| ios | Click Element / ios=.buttons().withName('Apps') | Matches by iOS UI Automation |
| nsp | Click Element / nsp=name=="login" | Matches by iOSNsPredicate |
| chain | Click Element / chain=XCUIElementTypeWindow[1]/* | Matches by iOS Class Chain |
| css | Click Element / css=.green_button | Matches by css in webview |
| name | Click Element / name=my_element | Matches by @name attribute |

<https://serhatbolsu.github.io/robotframework-appiumlibrary/AppiumLibrary.html>



Write your tests



Make it Work



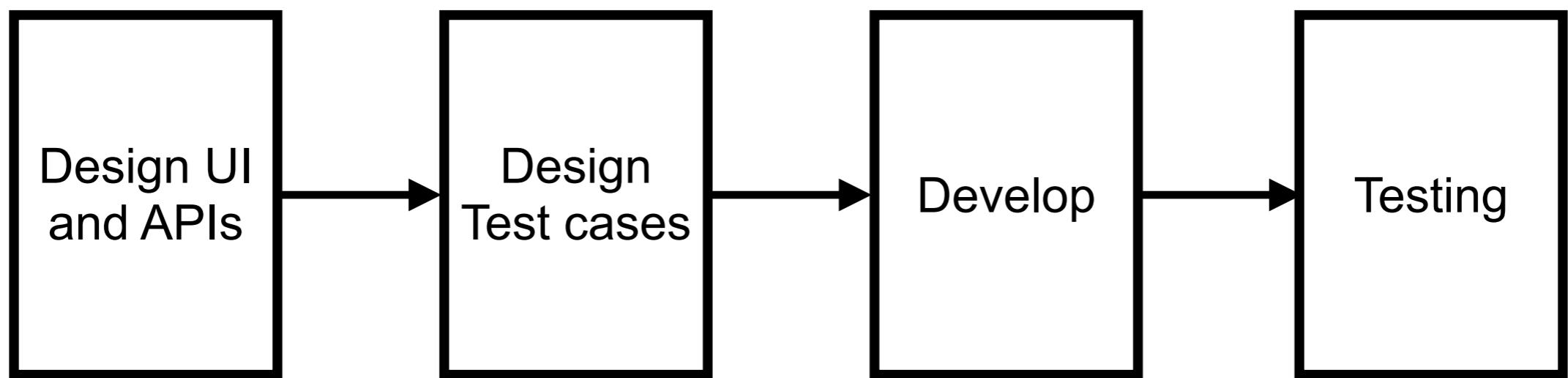
Make it Right



Test-First vs Test-Last ?



Improve Process

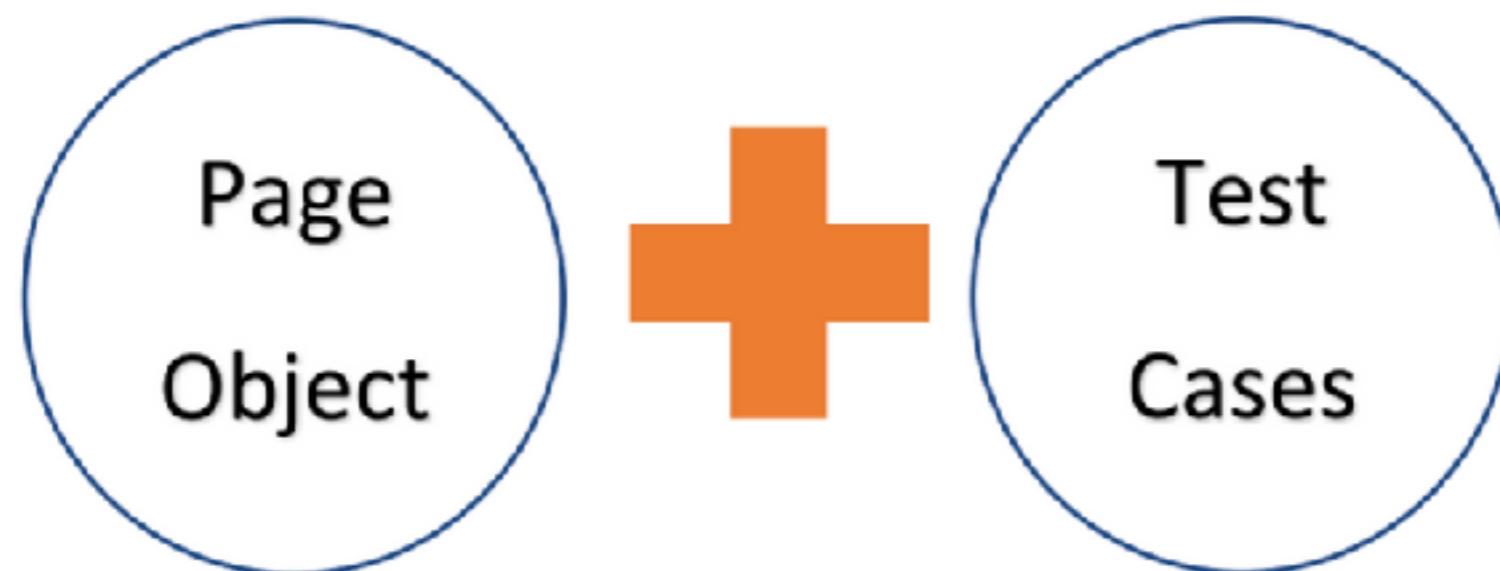


Page Object Pattern

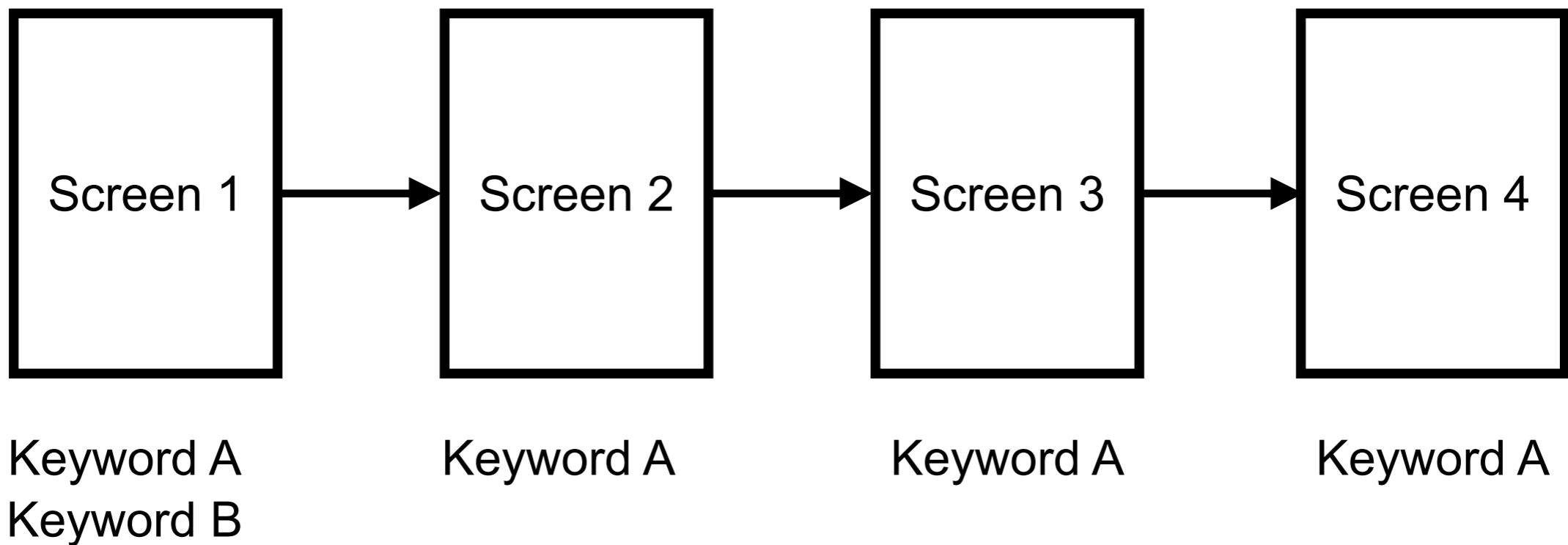


Page Object Pattern

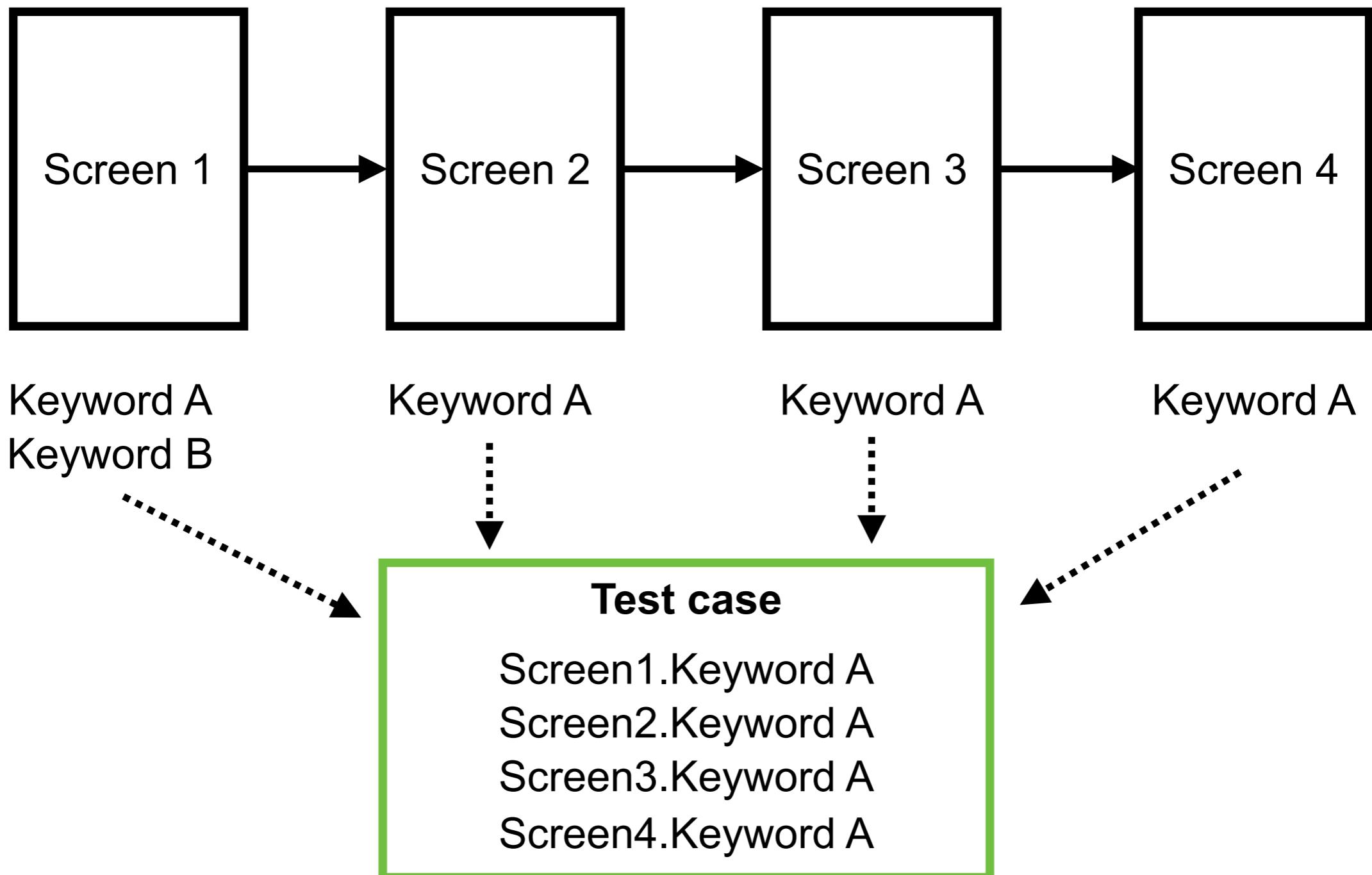
Maintainable test cases
Reduce code duplication
Working as a team



Use Page Object Pattern



Use Page Object Pattern

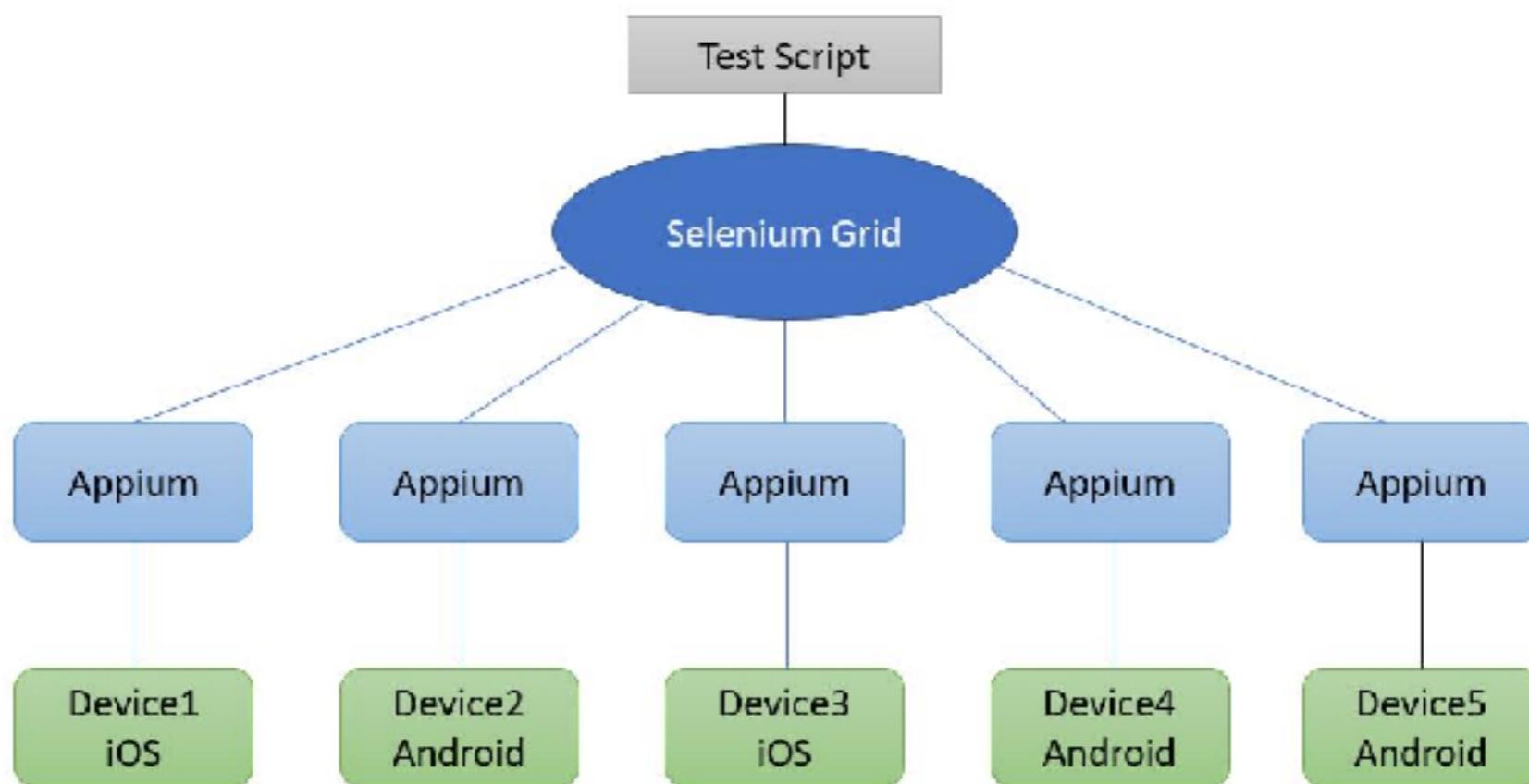


Make it Fast



Appium Grid

Proxy server that allow to run parallel tests
on multiple devices simultaneously



<https://appium.io/docs/en/2.1/guides/grid/>



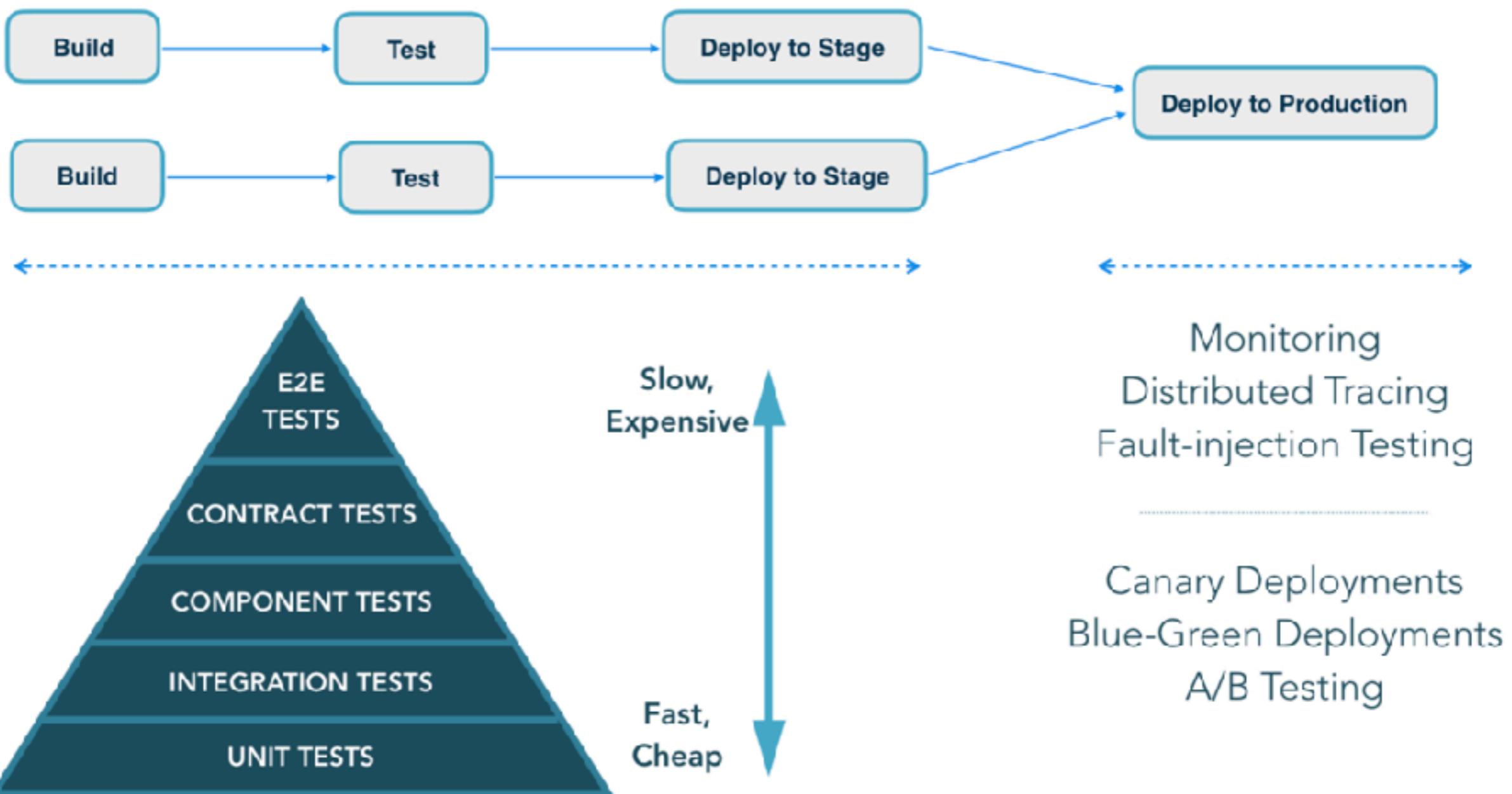
Continuous Integration



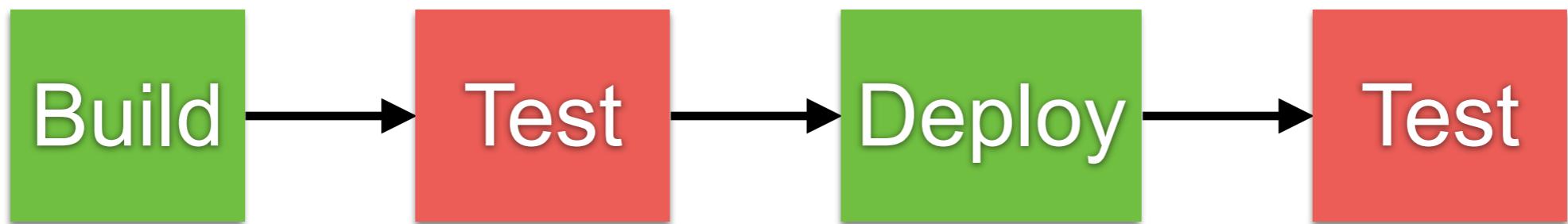
Continuous integration



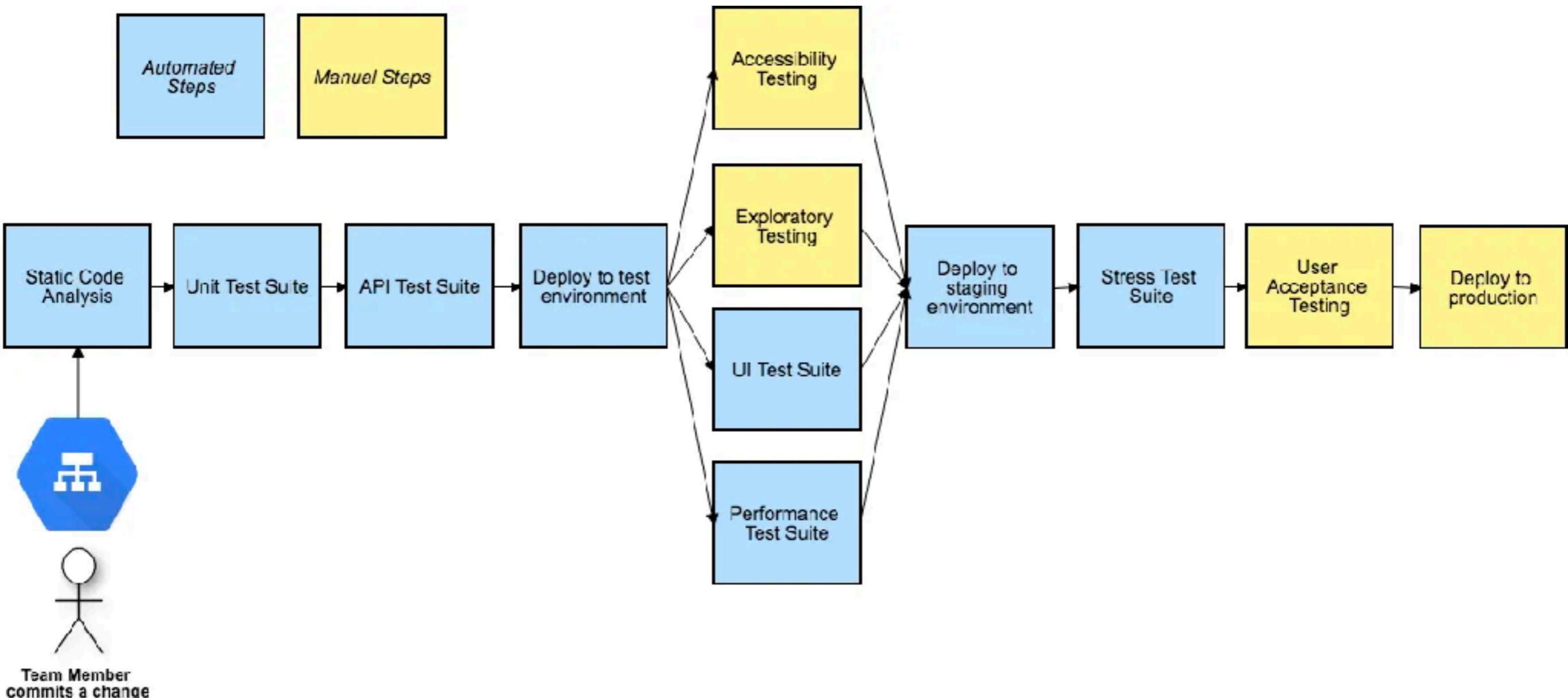
Test strategy



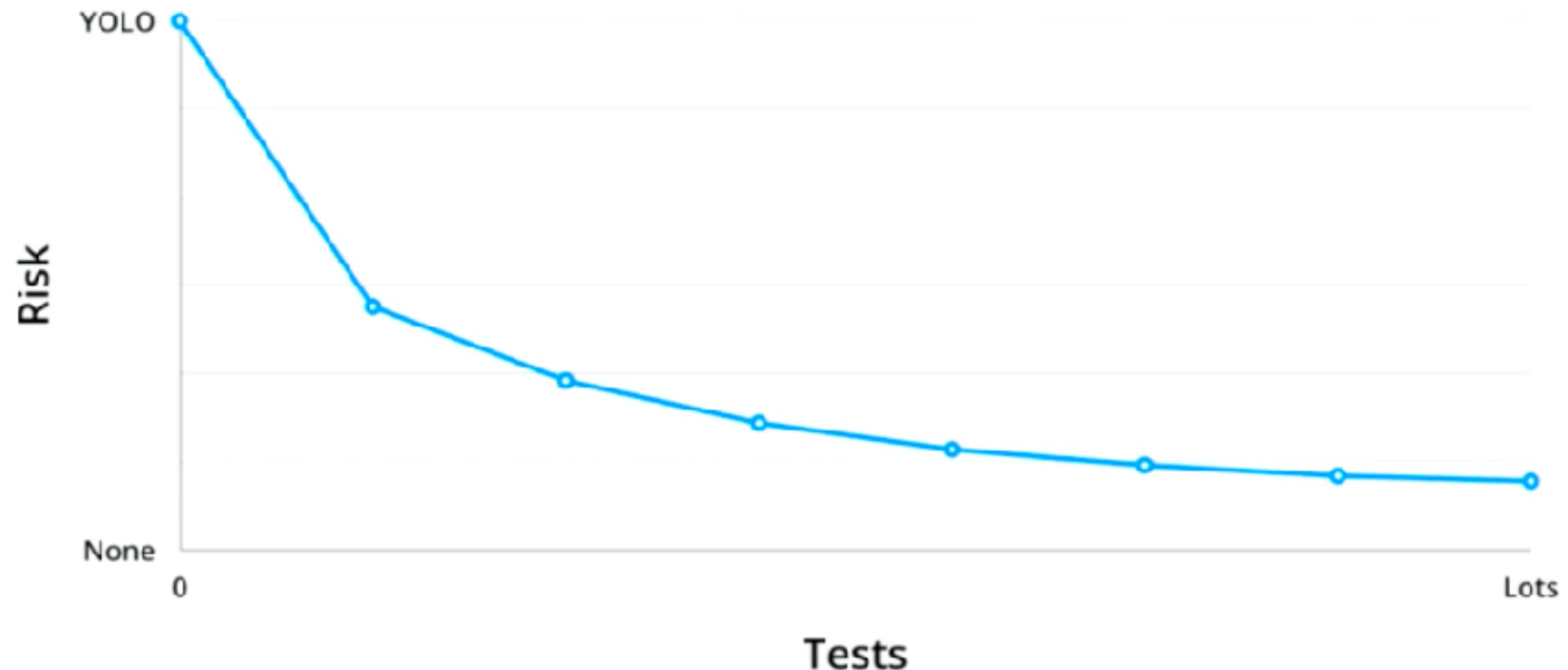
Design your pipeline



Design your pipeline/process



Reduce risk with tests



Q/A

