There is more in Appium Than you think

Rajdeep Varma

badoo

Uncover Secrets Of Your App - With EspressoDriver

Rajdeep Varma

badoo



Rajdeep Varma

mail.rajvarma@gmail.com

rajdeepv

@varmarajdeep

MagicLab

badoo

bumble

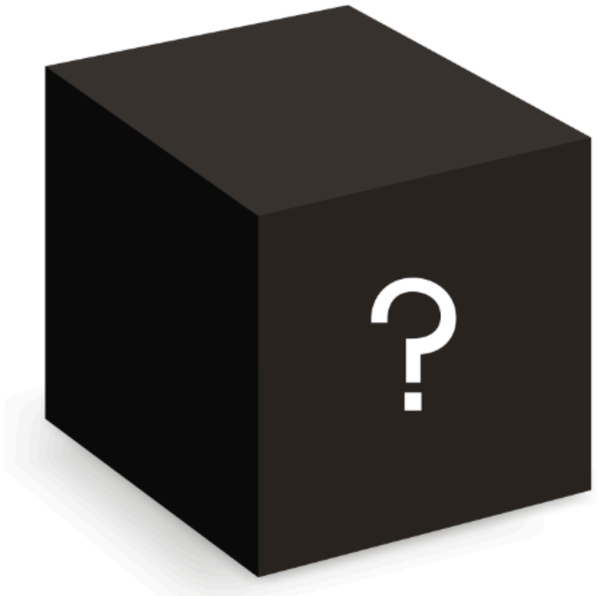
Lumen

CHAPPY






THERMOS.



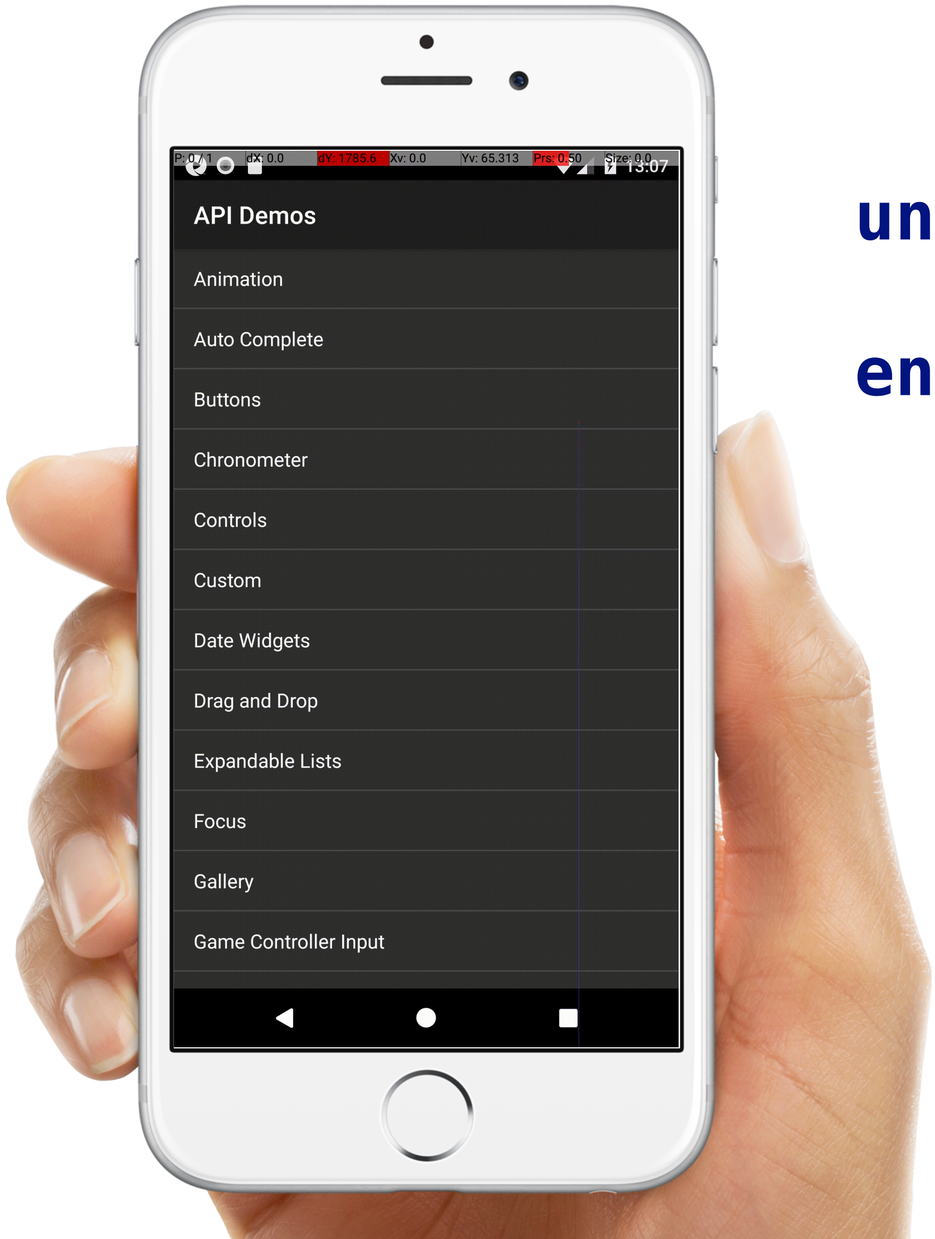
No Access to Information



Black Box with Appium

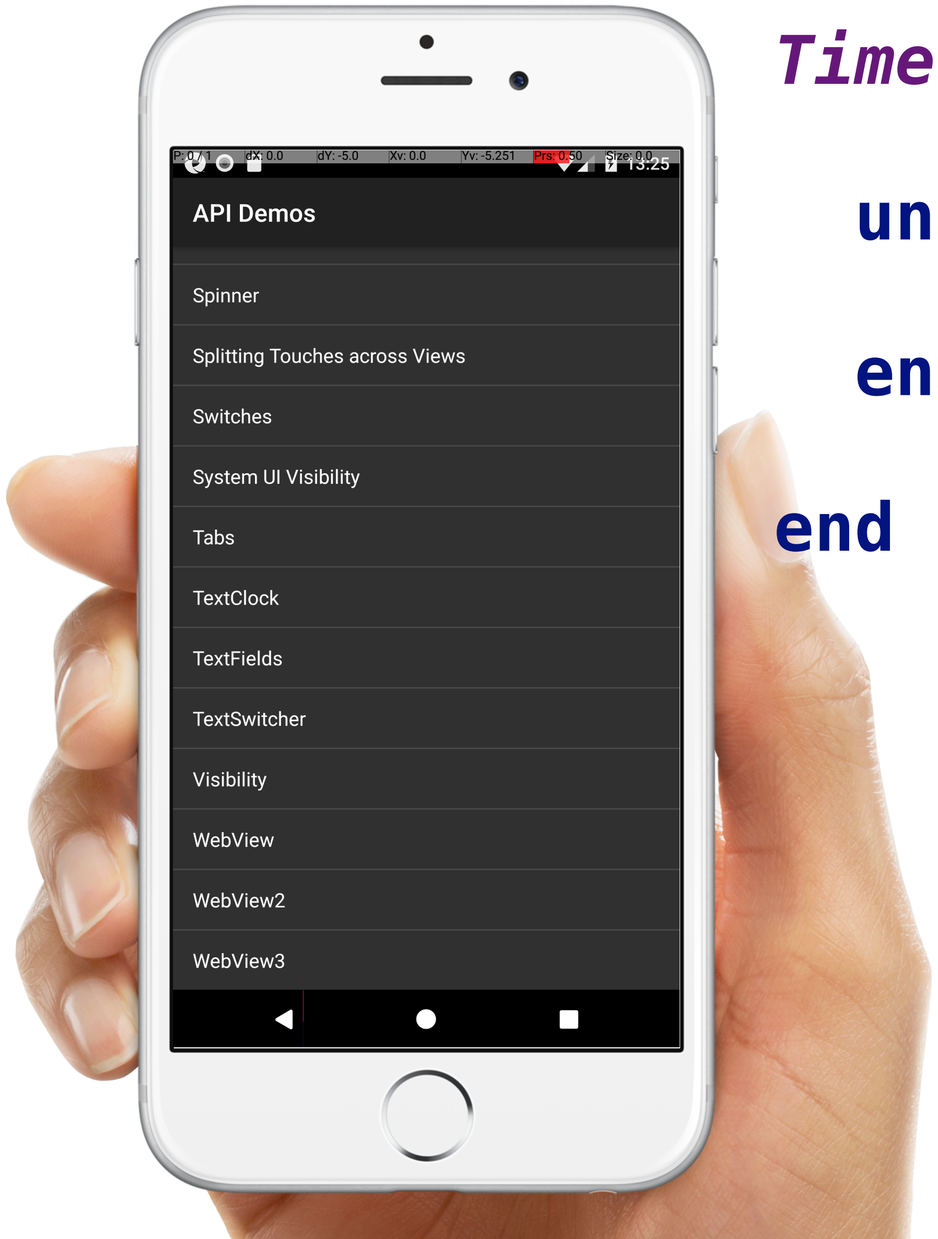
- Impossible to automate cases
- Slow/inefficient tests

Scroll




```
until element_exists?('Item 420') do
  scroll_down
end
```

Scroll

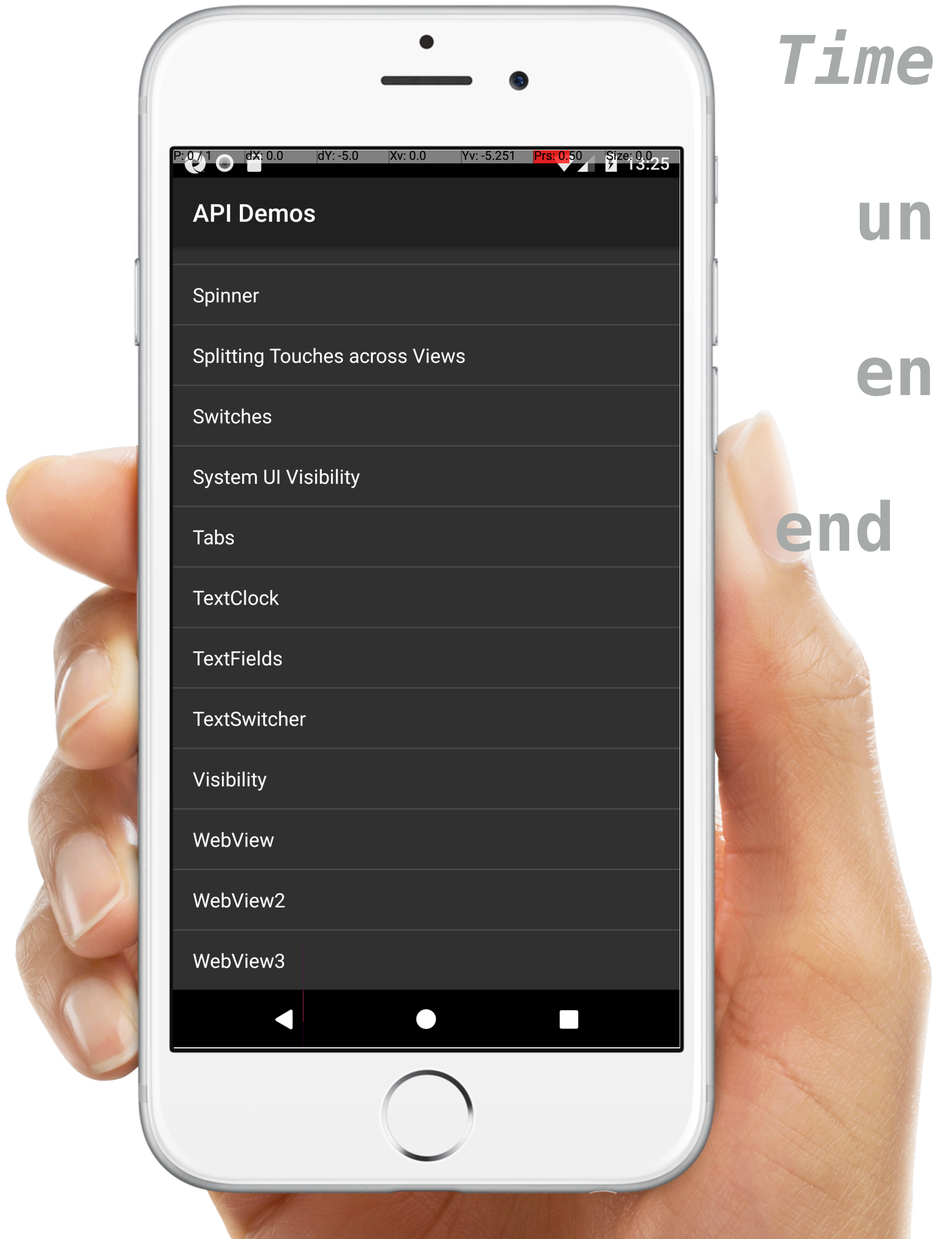



```
Timeout::timeout(30) do

  until element_exists?('Item 420') do
    scroll_down
  end

end
```

Scroll

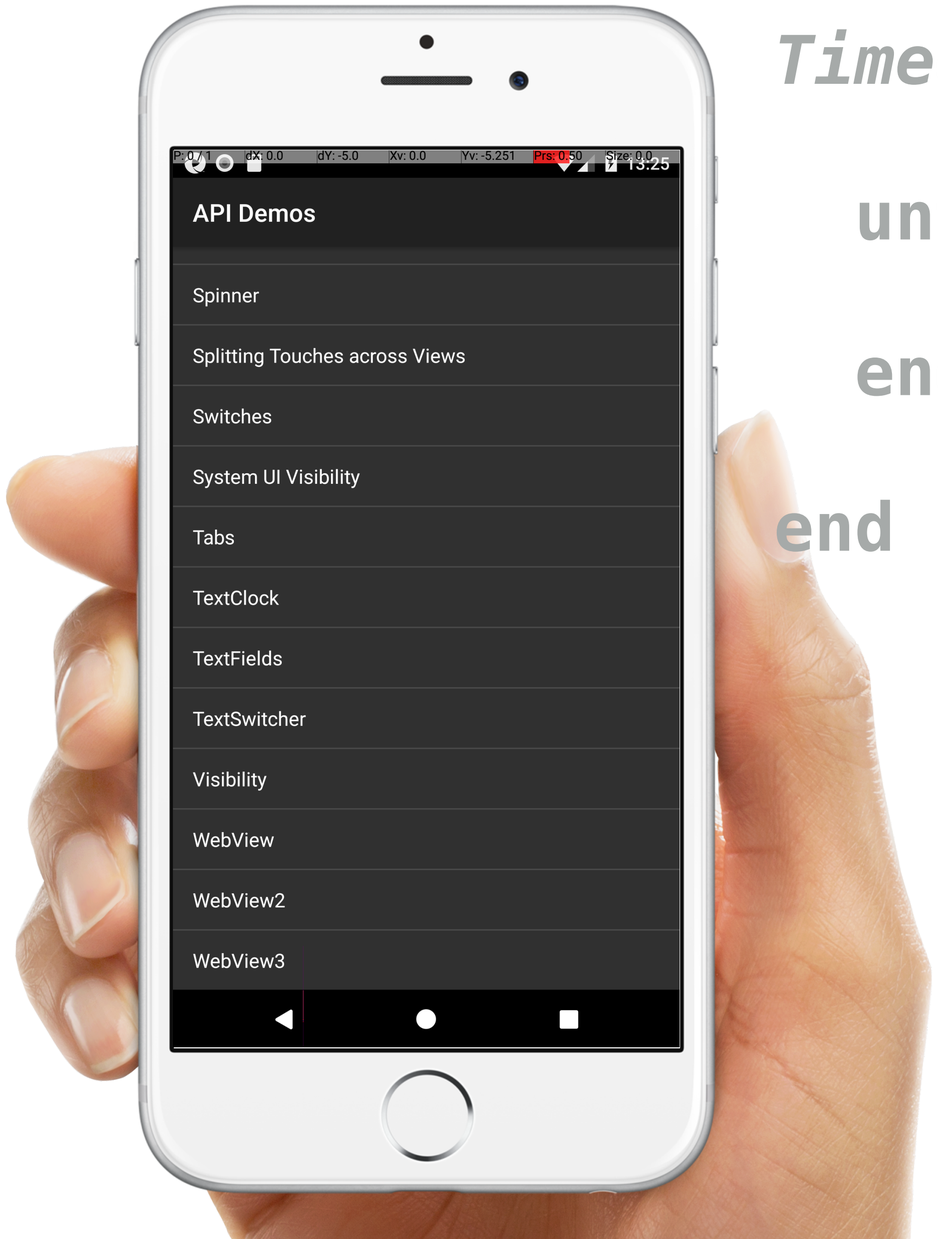



```
Timeout::timeout(30) do

  until element_exists?('Item 420') do
    scroll_down
  end

end
```

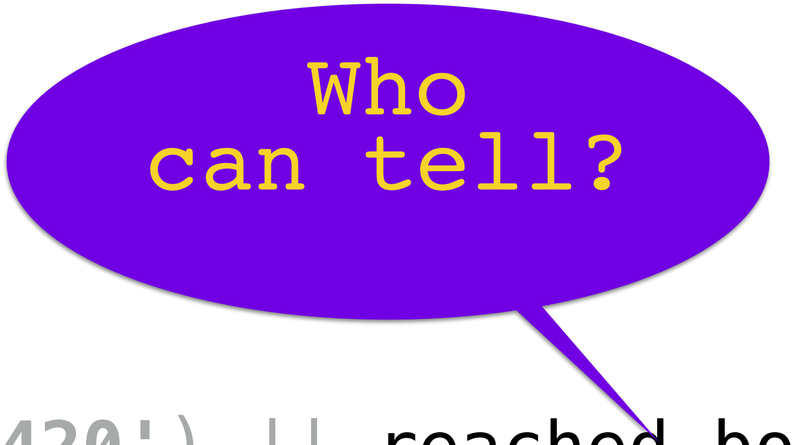
Scroll




```
Timeout::timeout 😒 do
  until element_exists?('Item 420') do
    scroll_down
  end

end
```

```
until element_exists?('Item 420') || reached_bottom? do
    scroll_down
end
```




```
until element_exists?('Item 420') || reached_bottom? do
    scroll_down
end
```

SurfaceControl
SurfaceControl.Builder
SurfaceControl.Transaction
SurfaceView
TextureView
TouchDelegate
VelocityTracker
View
View.AccessibilityDelegate
View.BaseSavedState
View.DragShadowBuilder
View.MeasureSpec
ViewAnimationUtils
ViewConfiguration
ViewDebug
ViewGroup
ViewGroup.LayoutParams
ViewGroup.MarginLayoutPar...

canScrollVertically

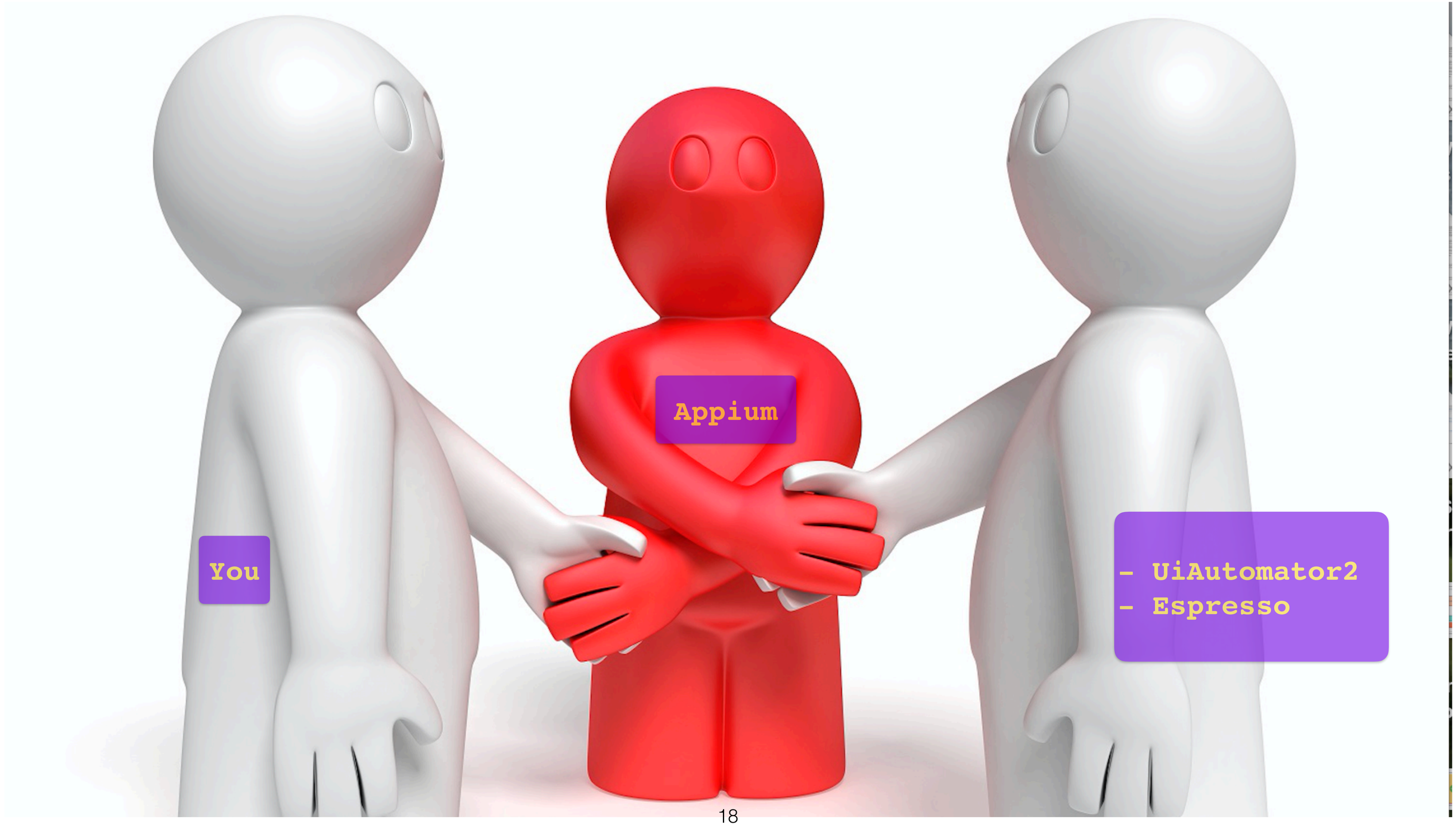
Added in API level 14

```
public boolean canScrollVertically (int direction)
```

Check if this view can be scrolled vertically in a certain direction.

Parameters	
direction	**int**: Negative to check scrolling up, positive to check scrolling down.

Returns	
boolean	true if this view can be scrolled in the specified direction, false otherwise.


You
Appium
- UiAutomator2
- Espresso

UiAutomator2

😞 Black Box

😃 Can Automate across apps

Espresso

😃 White Box

😞 Can Automate only app under test

How to utilise this?

How to overcome this?

Espresso

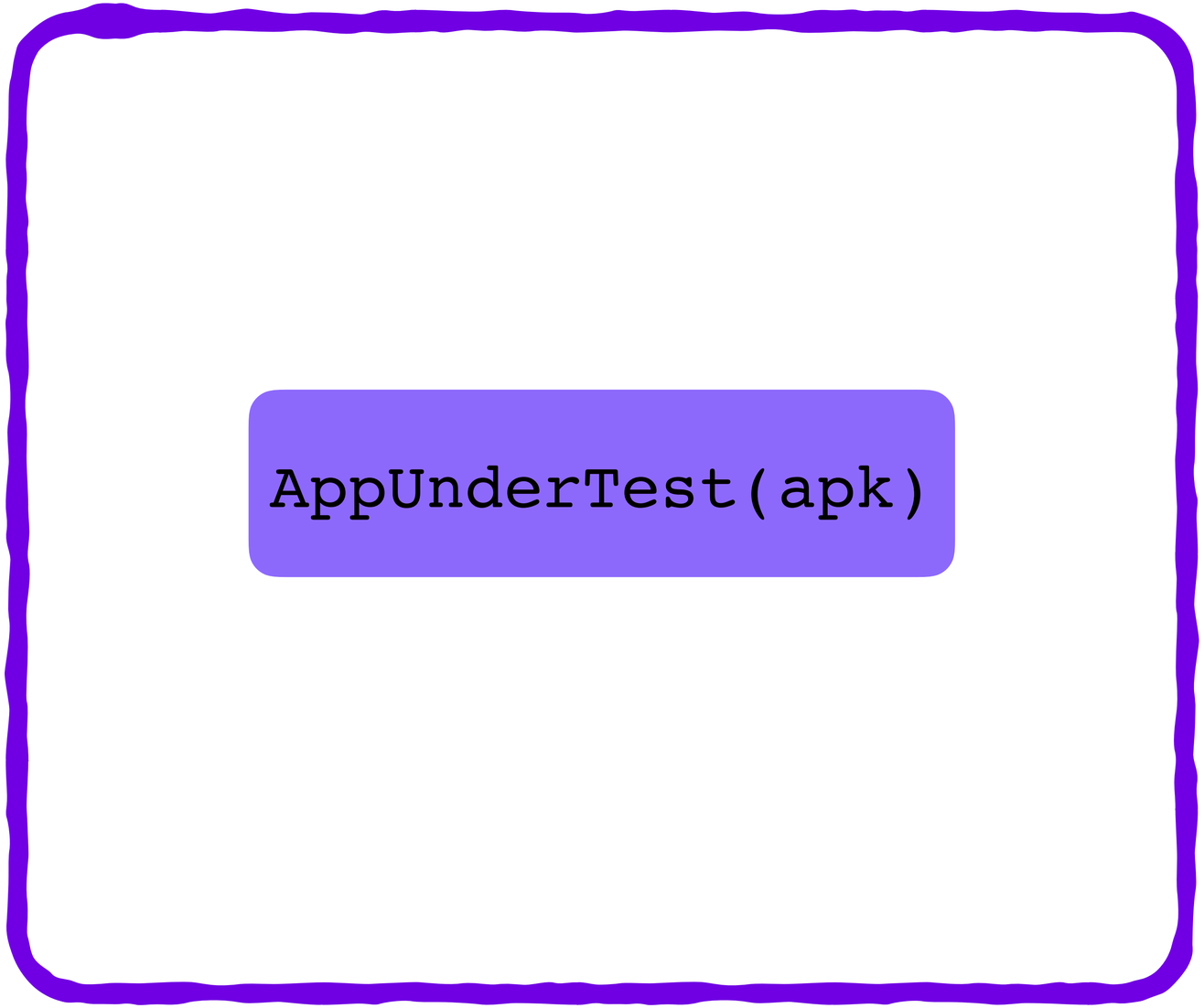
😃 White Box

😞 Can Automate only app under test

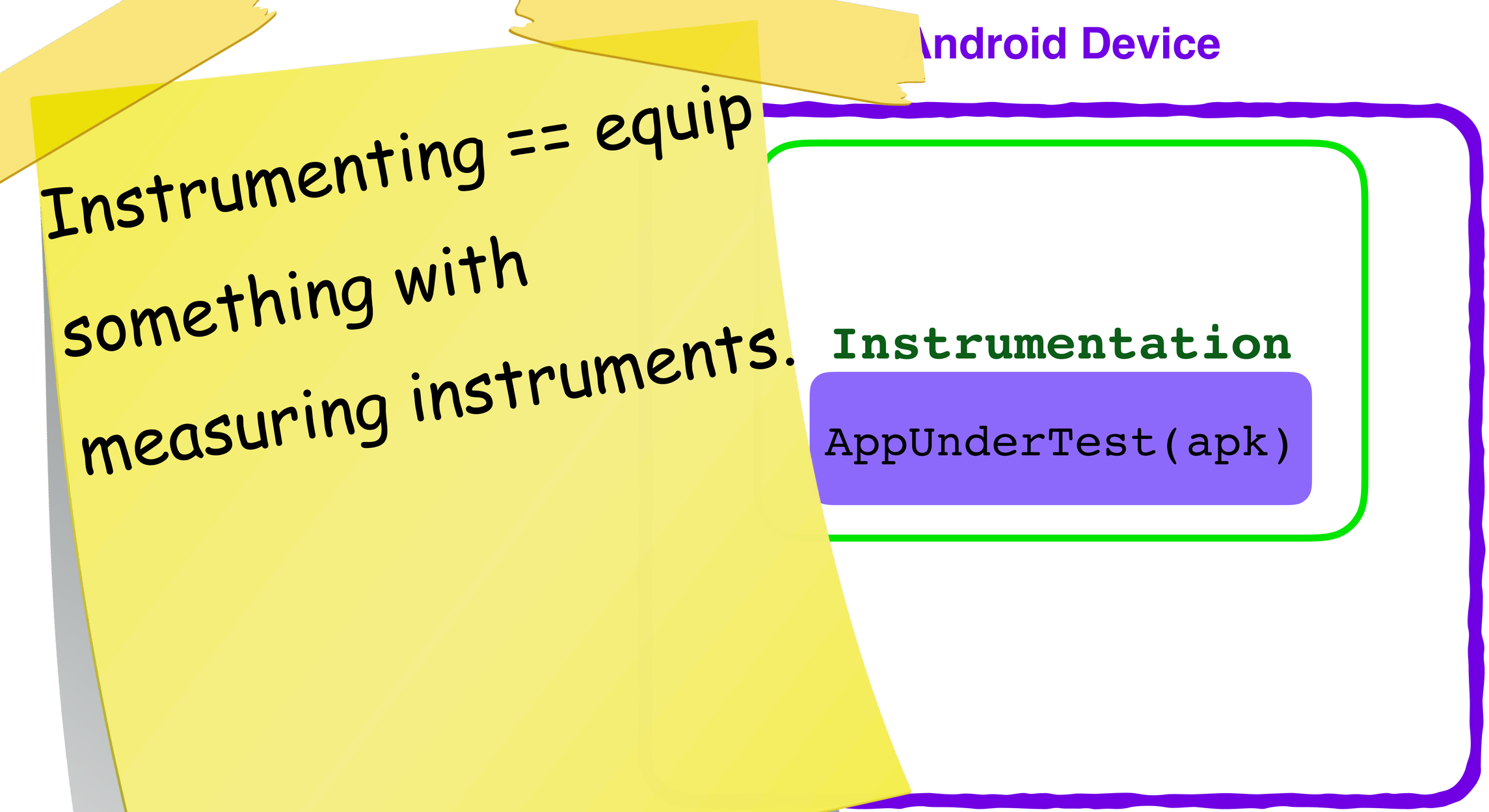
Demo

How?

Android Device



Instrumenting == equip something with measuring instruments.



Instrumentina == equip

sc

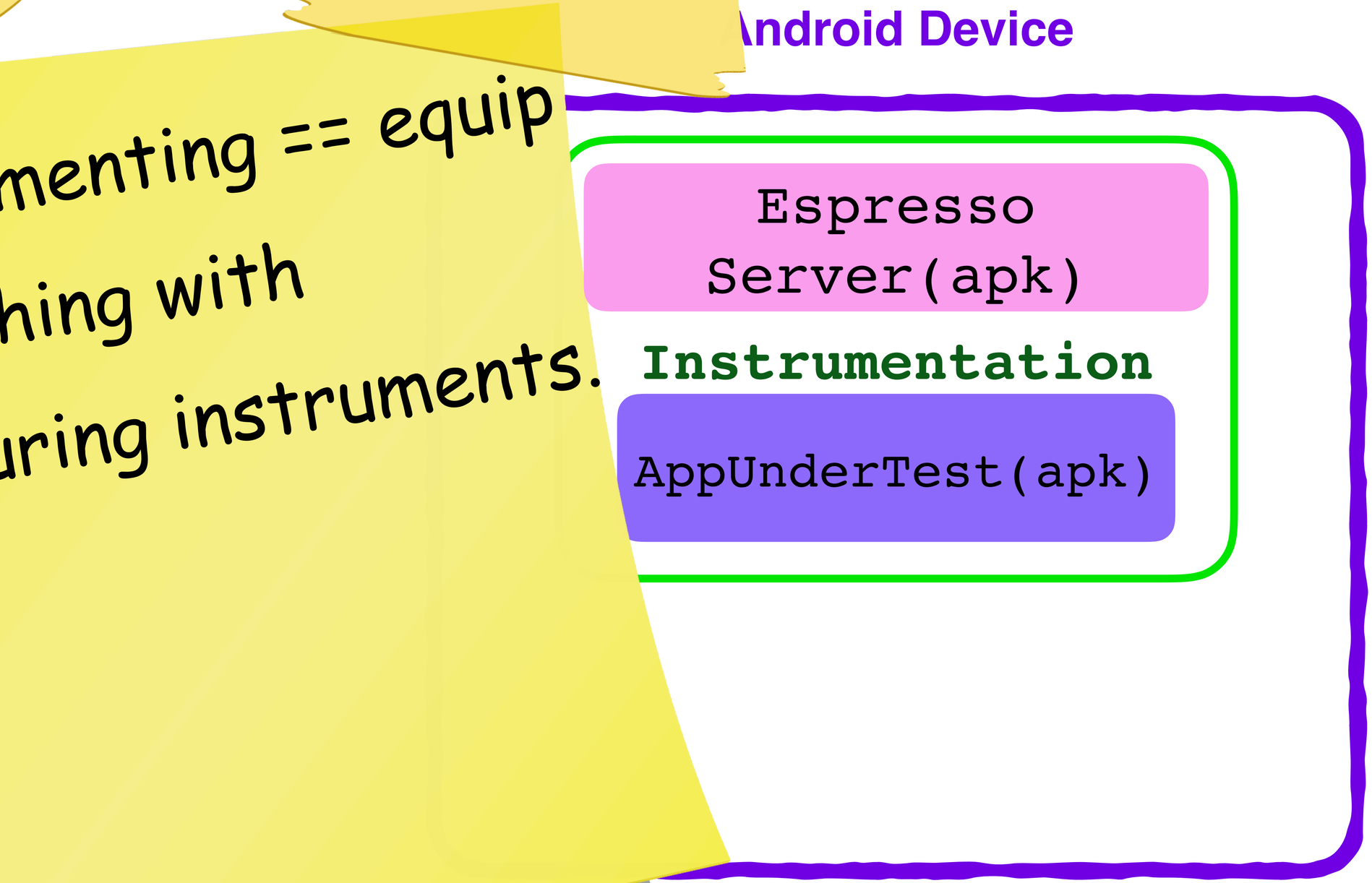
m

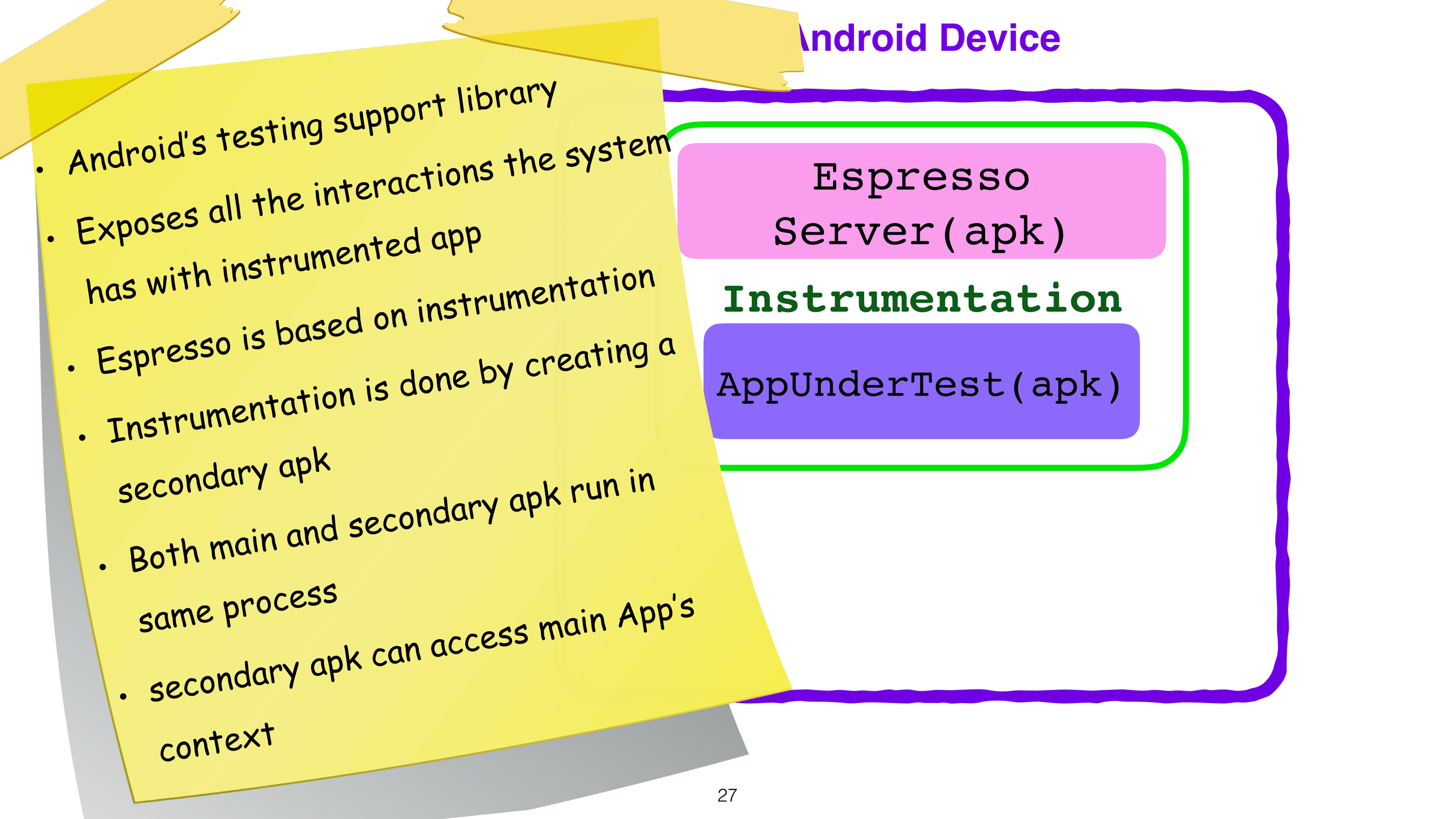


on

)

Instrumenting == equip something with measuring instruments.




ndroid Device
Espresso
Server(apk)
Instrumentation
AppUnderTest(apk)
Android's testing support library
Exposes all the interactions the system has with instrumented app
Espresso is based on instrumentation
Instrumentation is done by creating a secondary apk
Both main and secondary apk run in same process
secondary apk can access main App's context

```
[HTTP] --> POST /wd/hub/session

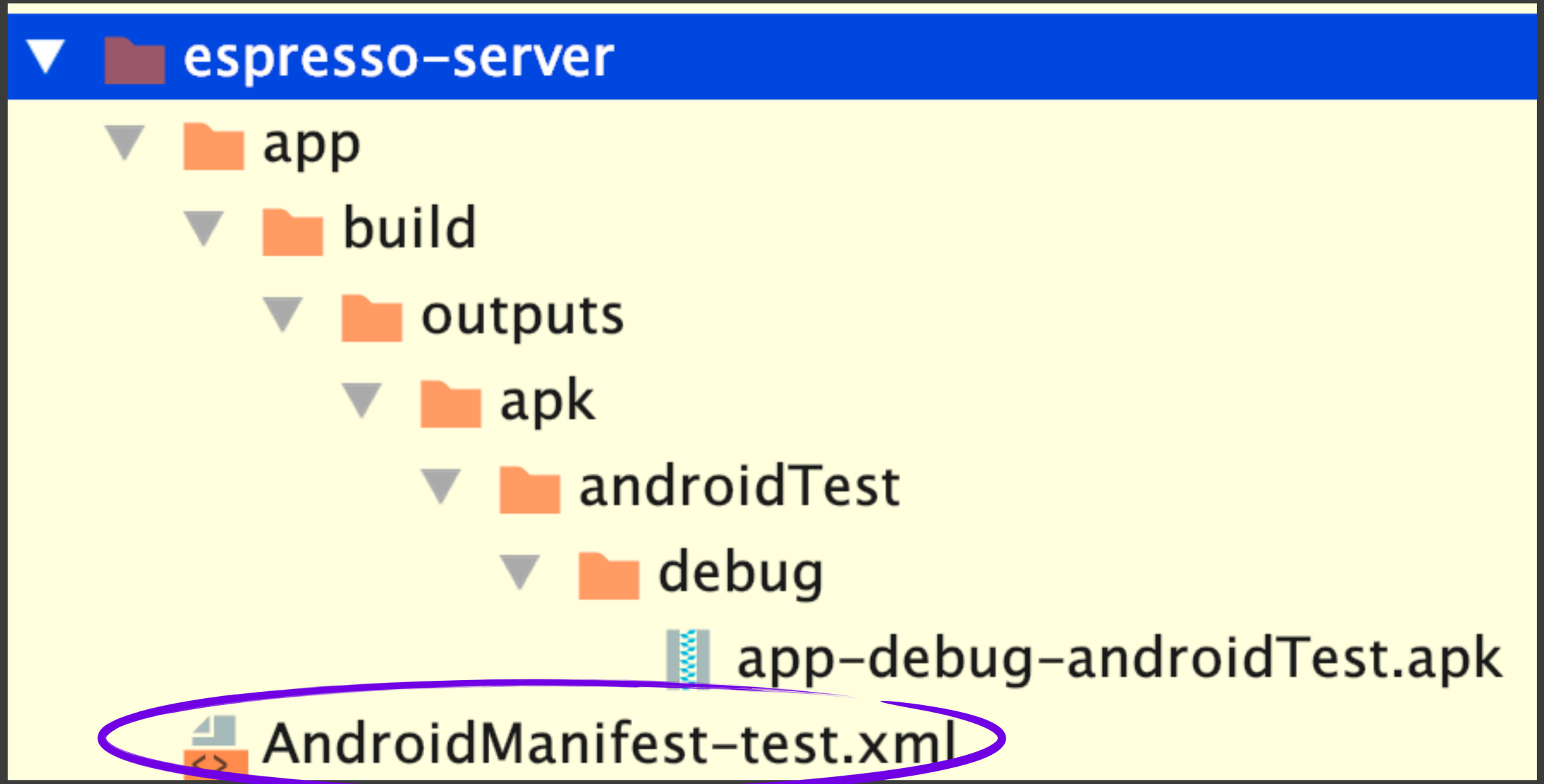

{
  "desiredCapabilities": {
    "platformName": "Android",
    "deviceName": "emulator-5554",
    "app": "app-debug.apk",
    "automationName": "espresso"
  }
}
```

```
> apkanalyzer -h manifest print app-debug.apk


<manifest
    ...
    package="com.hfad.messanger">
    ...
</manifest>
```

```
> apkanalyzer -h manifest print app-debug.apk

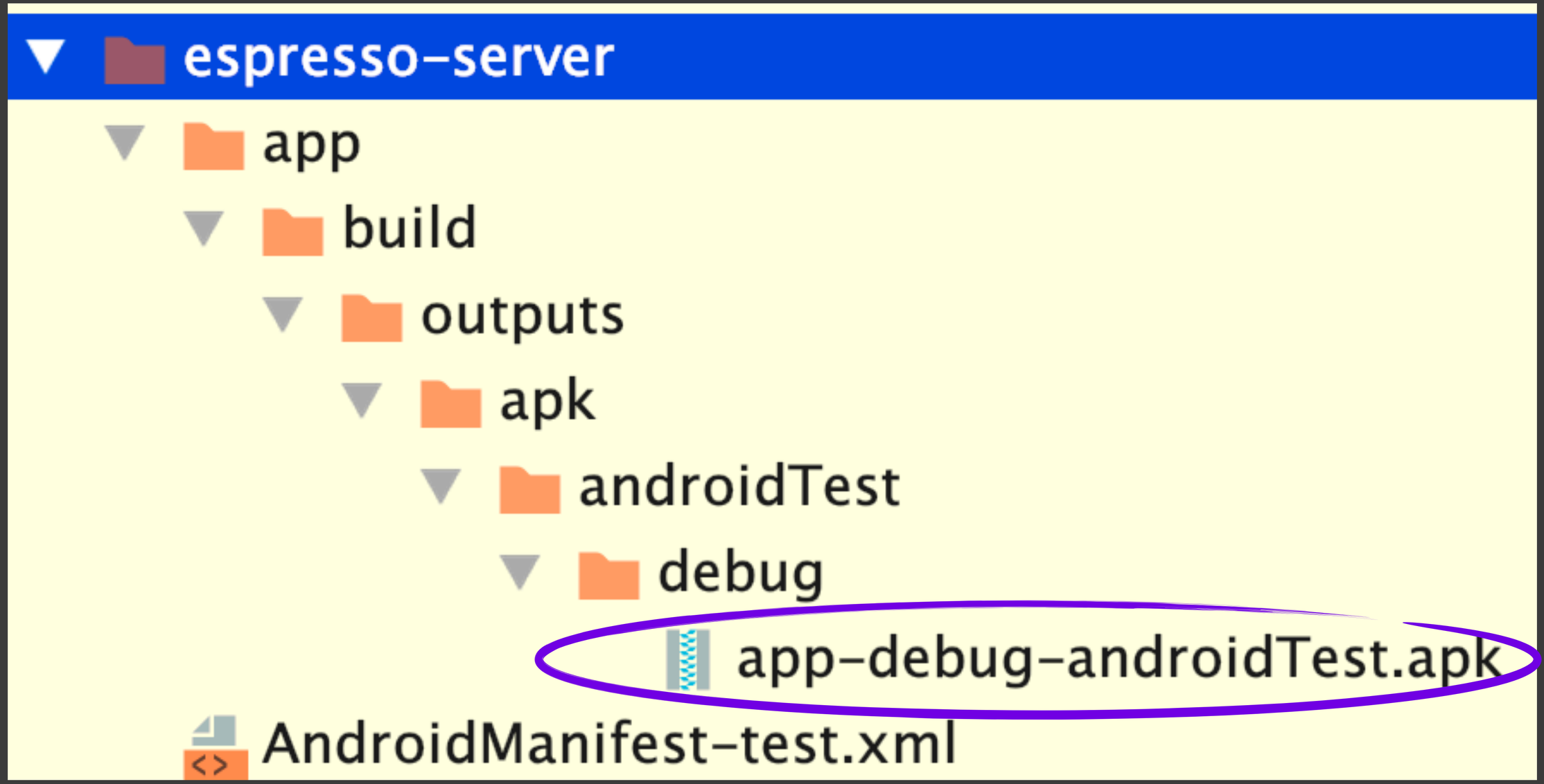

<manifest
    ...
    package="com.my.app">
    ...
</manifest>
```


espresso-server
app
build
outputs
apk
androidTest
debug
app-debug-androidTest.apk
AndroidManifest-test.xml

```
> aapt2 link -o <tmp_dir>/
AndroidManifest.xml.apk —manifest <path_to>/
AndroidManifest.xml --rename-manifest-package
io.appium.espressoserver.test —rename-
instrumentation-target-package com.my.app -I ~/
Library/Android/sdk/platforms/android-28/
android.jar
```

Generates AndroidManifest.xml.apk

```
<manifest
    ...
    <instrumentation
        android:targetPackage="com.my.app"/>
    ...
</manifest>
```

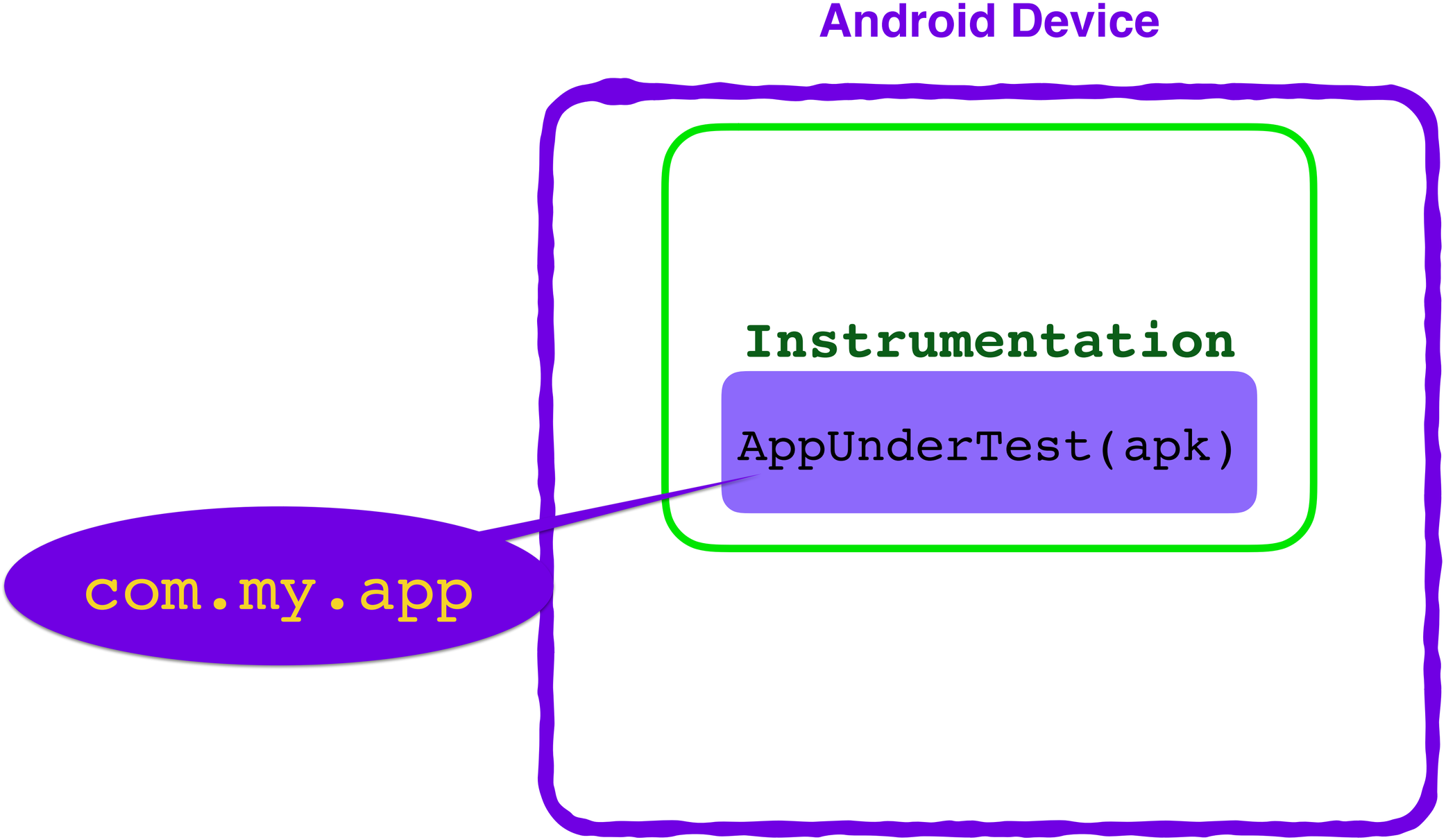

espresso-server
app
build
outputs
apk
androidTest
debug
app-debug-androidTest.apk
AndroidManifest-test.xml

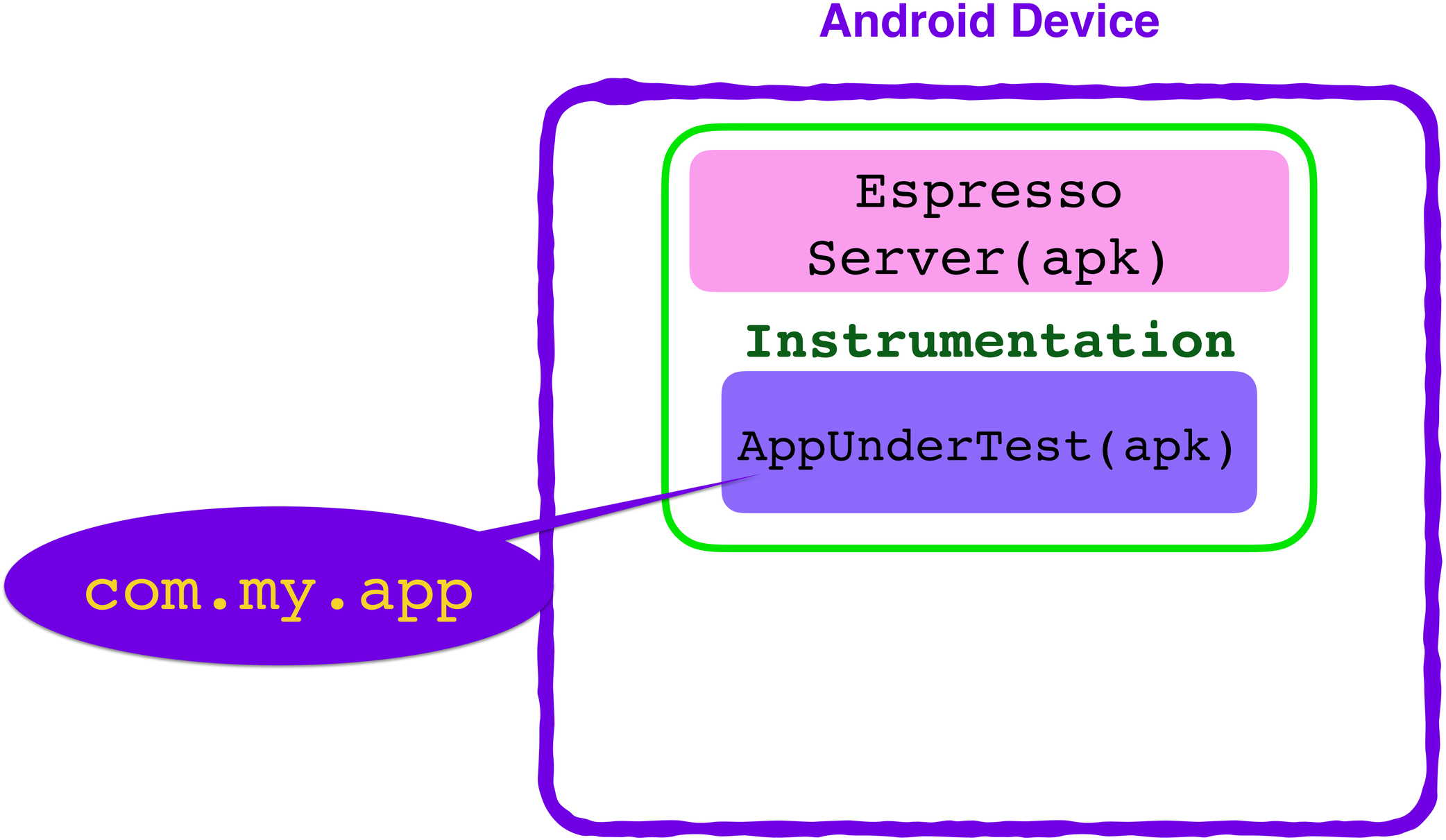
```
> unzip AndroidManifest.xml.apk
    extract AndroidManifest.xml from
    AndroidManifest.xml.apk

> aapt remove app-debug-androidTest.apk
AndroidManifest.xml

> aapt add app-debug-androidTest.apk
AndroidManifest.xml

resign app-debug-androidTest.apk with same key
as app under test
```


Android Device
Instrumentation
AppUnderTest(apk)
com.my.app


Android Device
Espresso
Server(apk)
Instrumentation
AppUnderTest(apk)
com.my.app

Android Device

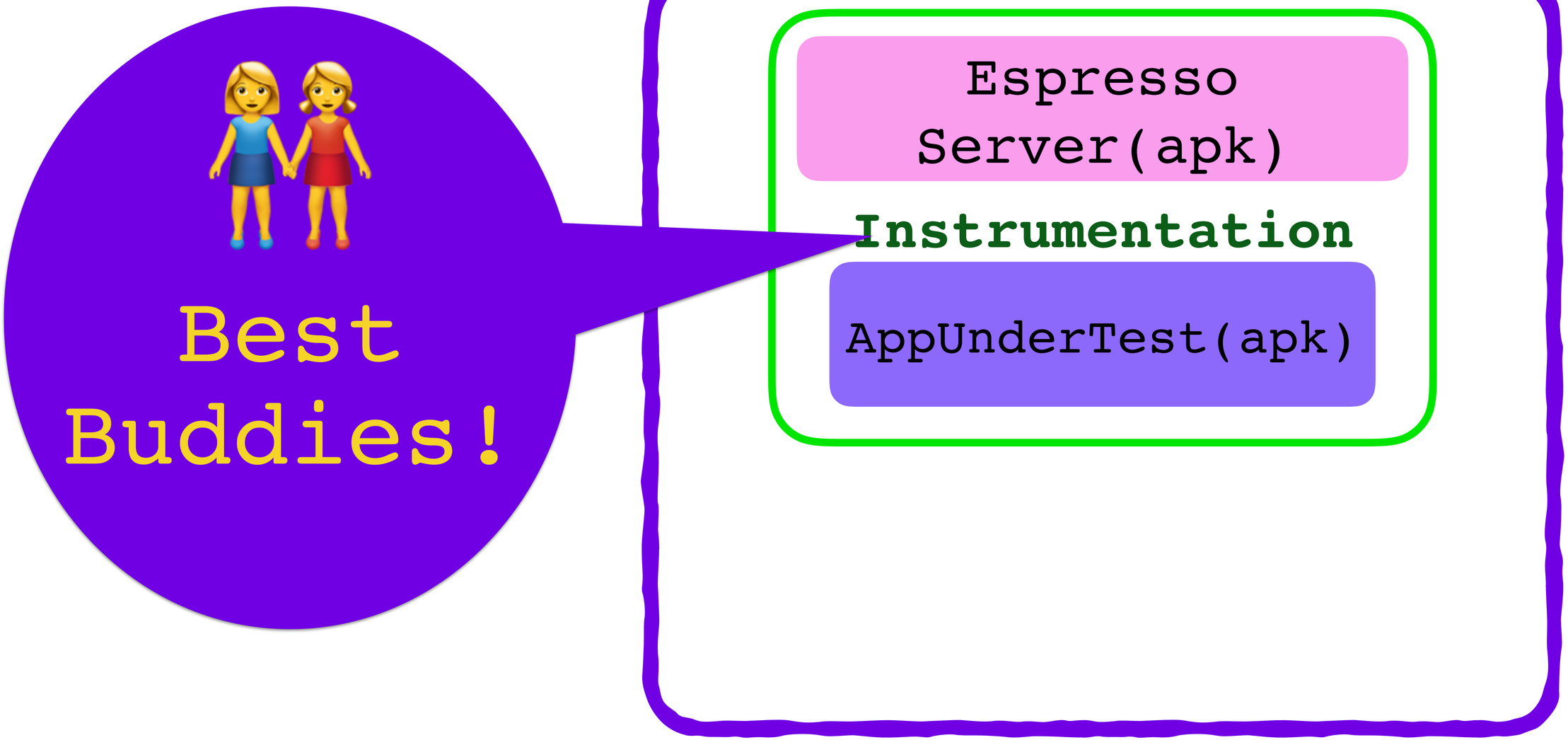
Espresso Server Manifest

```
<manifest
    ...
   <instrumentation
    android:label="Tests for io.appium.espressoserver"
  android:name="androidx.test.runner.AndroidJUnitRunner"
    android:targetPackage="com.my.app"
    android:handleProfiling="false"
    android:functionalTest="false" />
    ...
</manifest>
```

Android Device

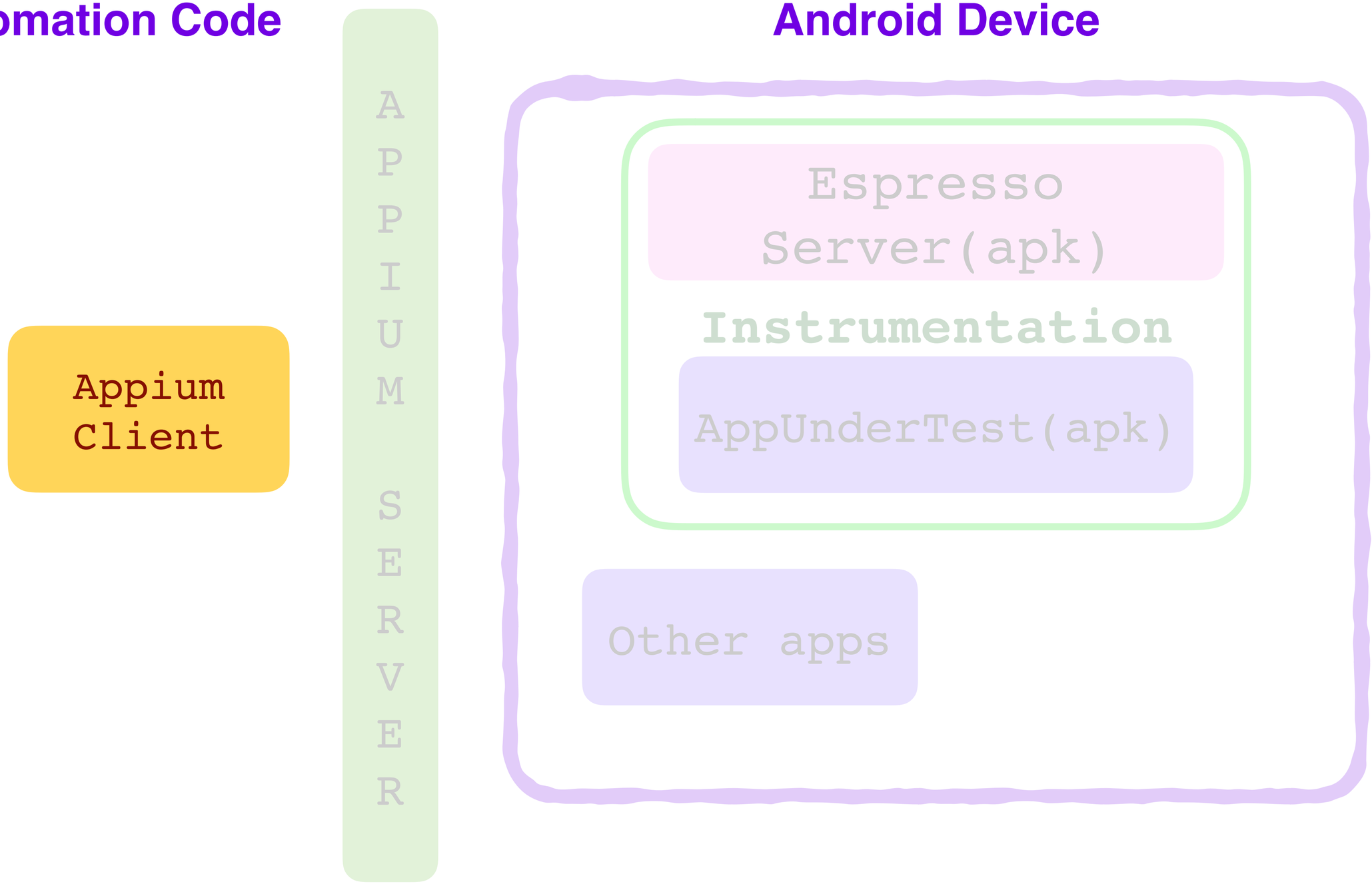
Espresso Server Manifest

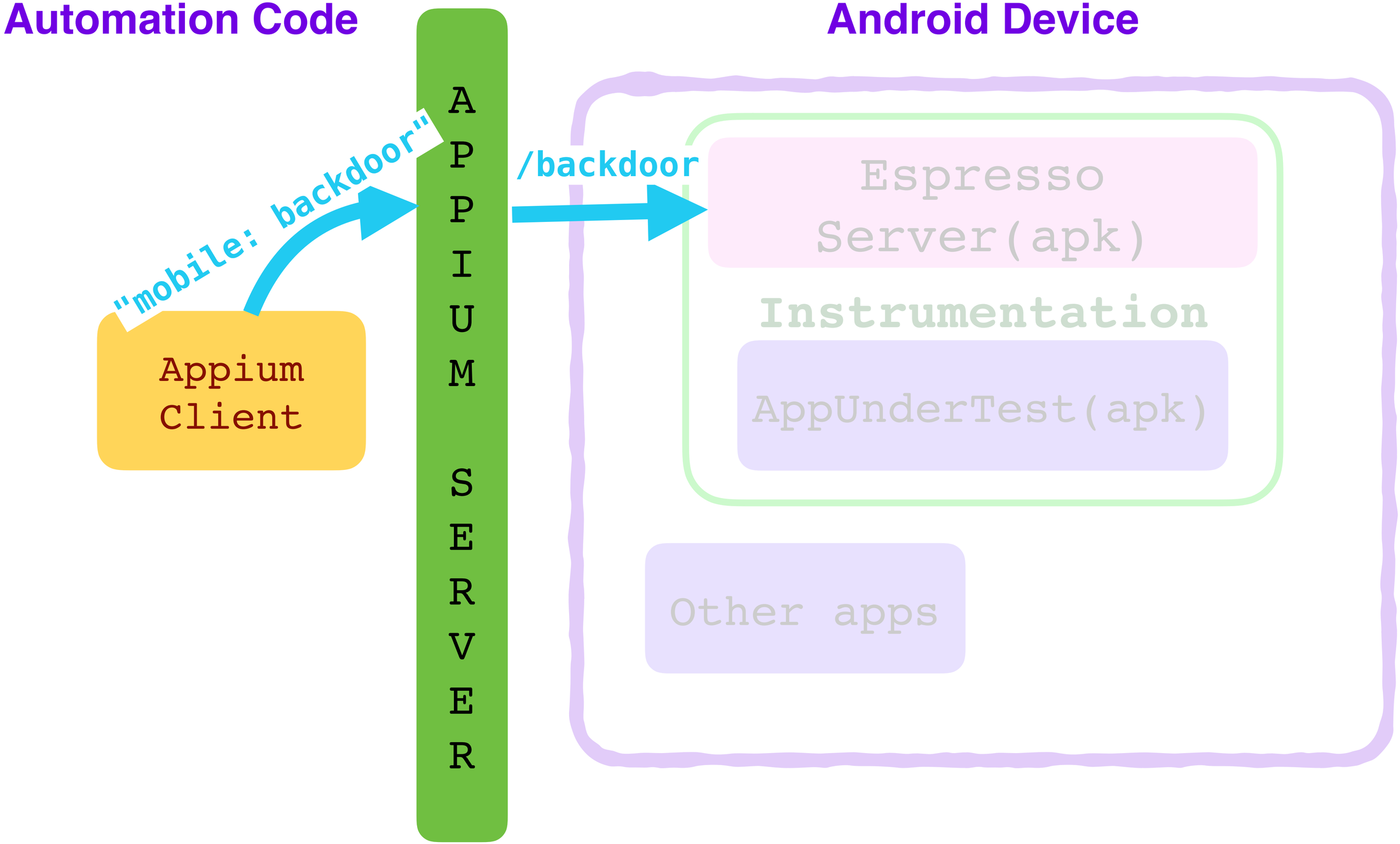
```
<manifest
    ...
    <instrumentation
     android:label="Tests for io.appium.espressoserver"
   android:name="androidx.test.runner.AndroidJUnitRunner"
     android:targetPackage="com.my.app"
     android:handleProfiling="false"
     android:functionalTest="false" />
    ...
</manifest>
```

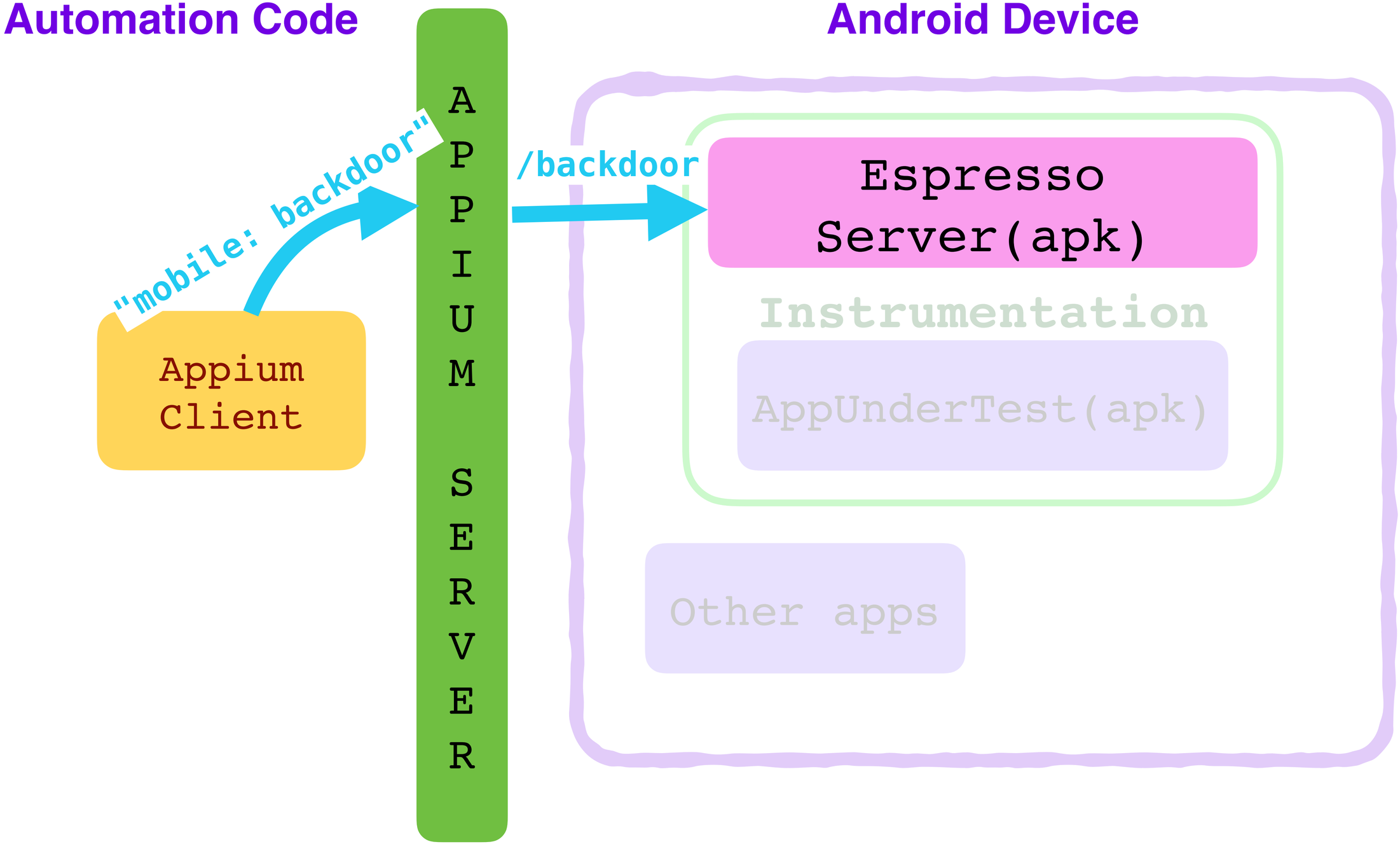

Android Device
Espresso
Server(apk)
Instrumentation
AppUnderTest(apk)
Best
Buddies!

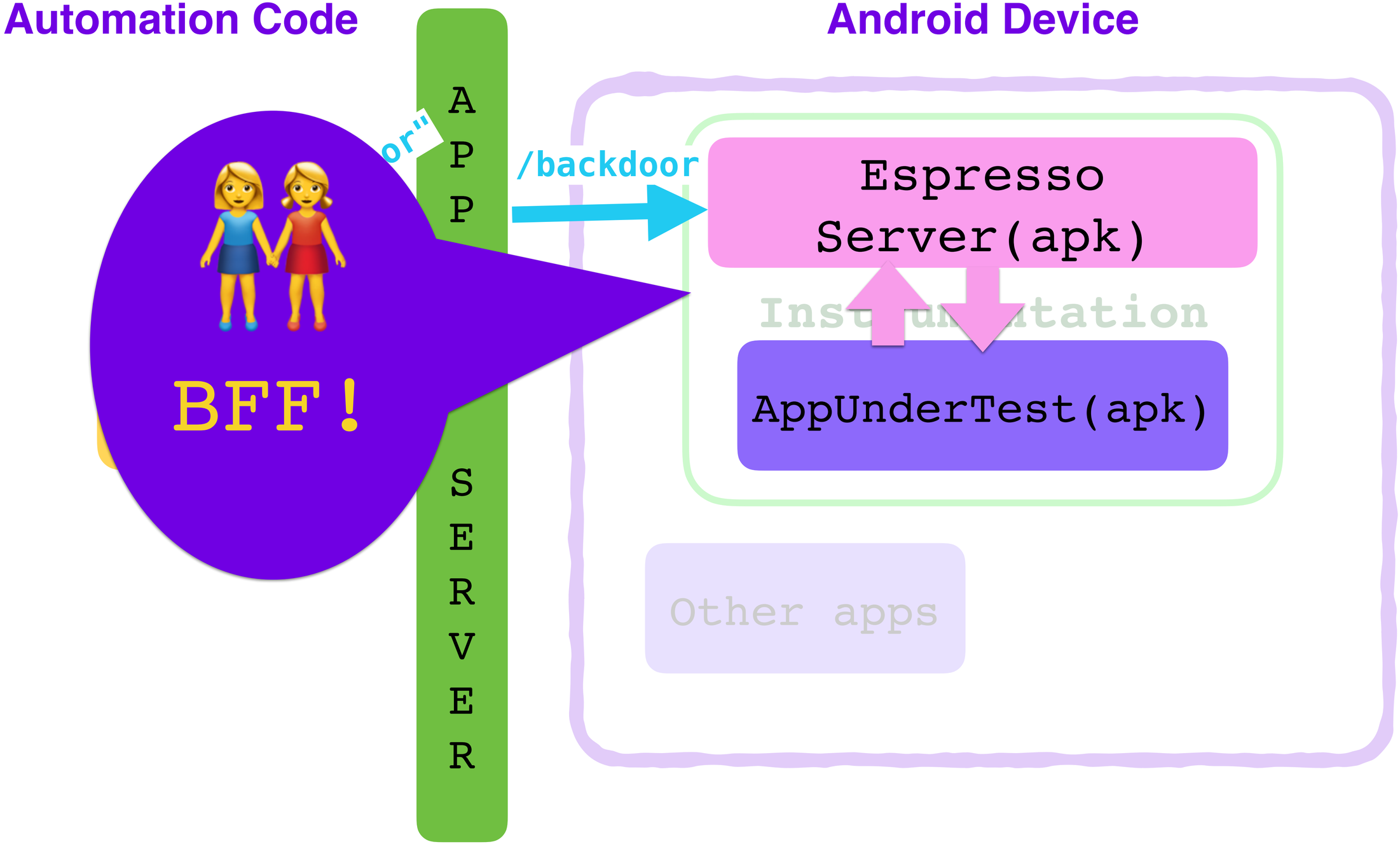
Automation Code

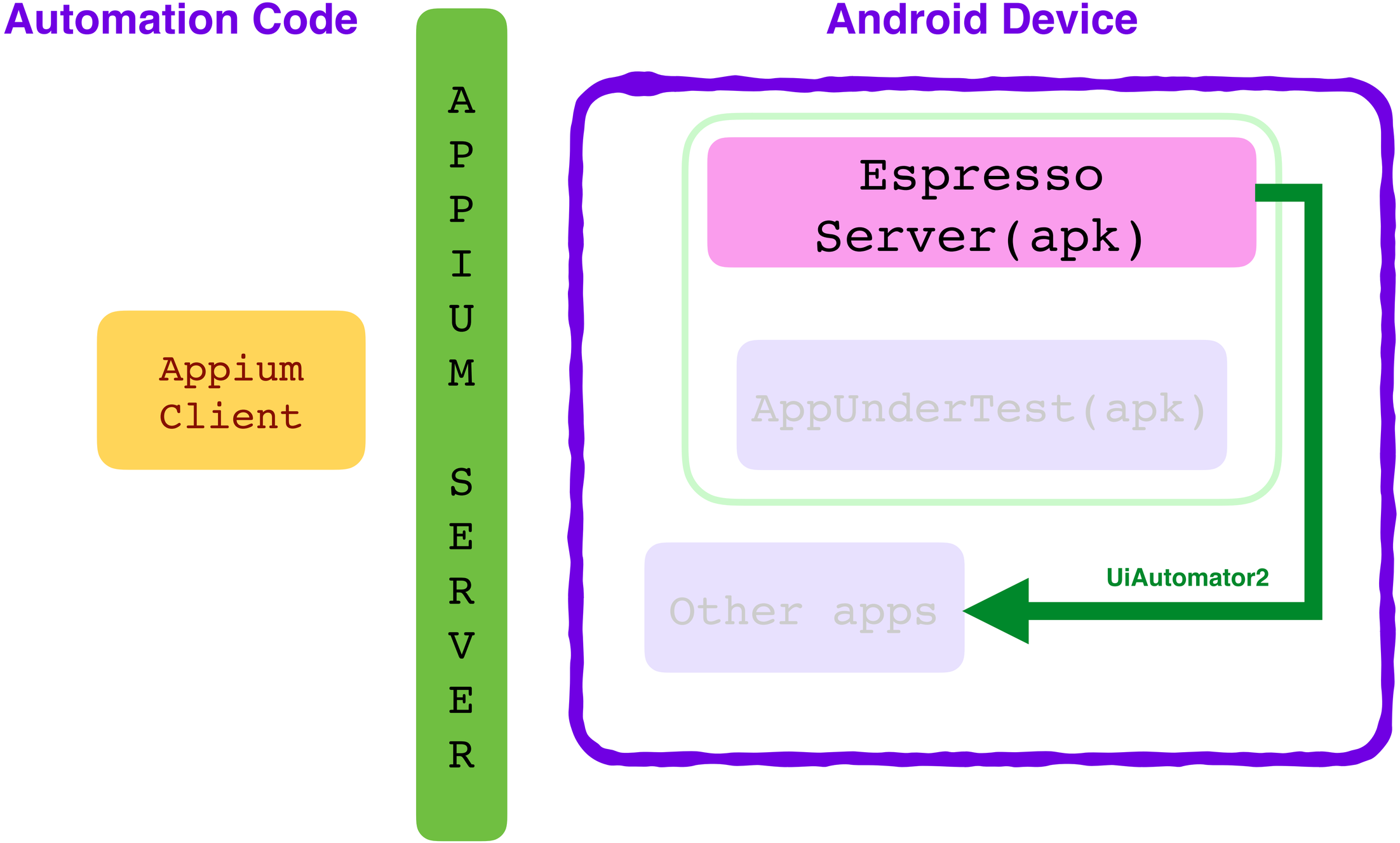
Android Device

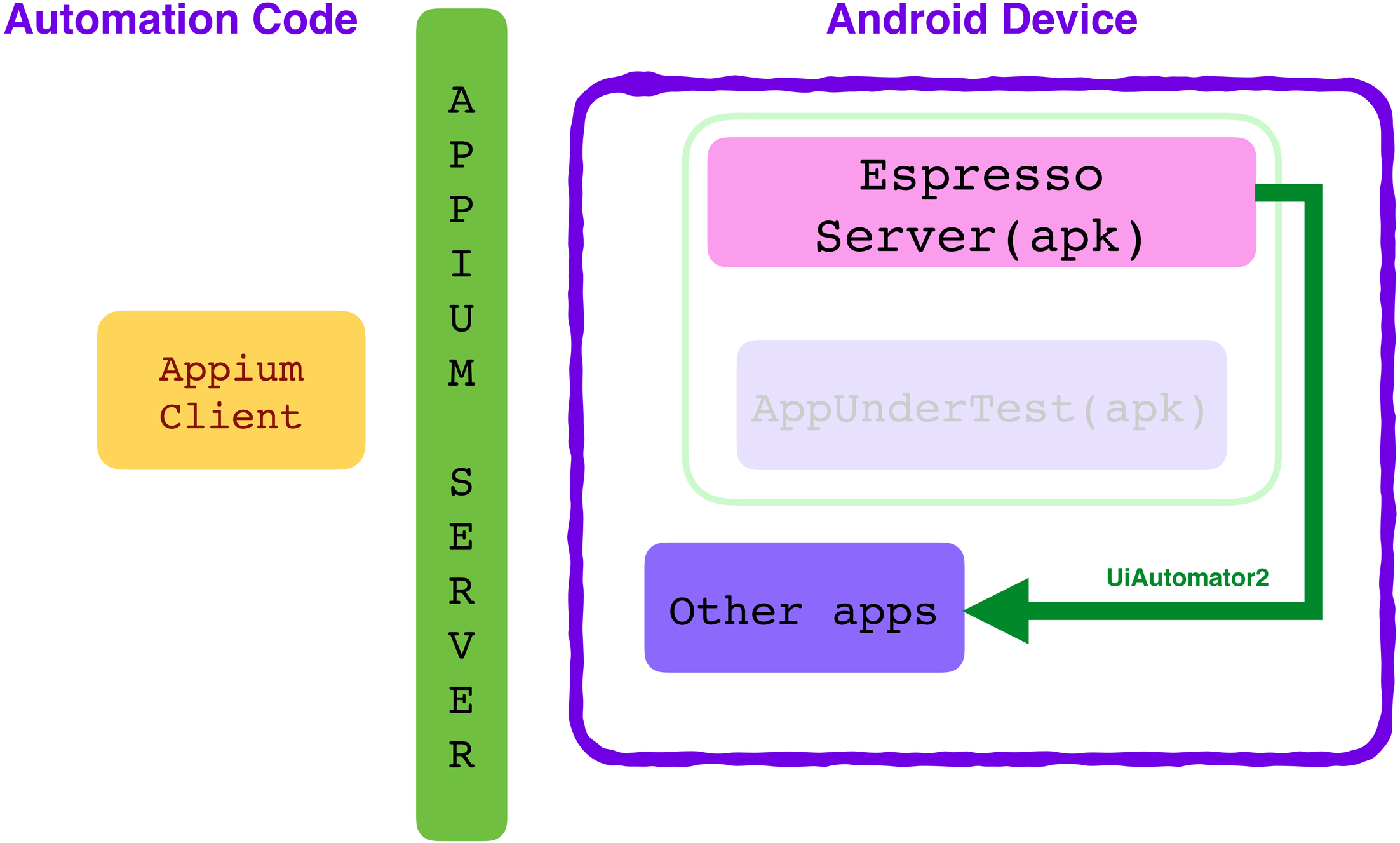



Automation Code
Android Device
"mobile: backdoor"
Appium Client
APPIUM SERVER
/backdoor
Espresso Server(apk)
Instrumentation
AppUnderTest(apk)
Other apps

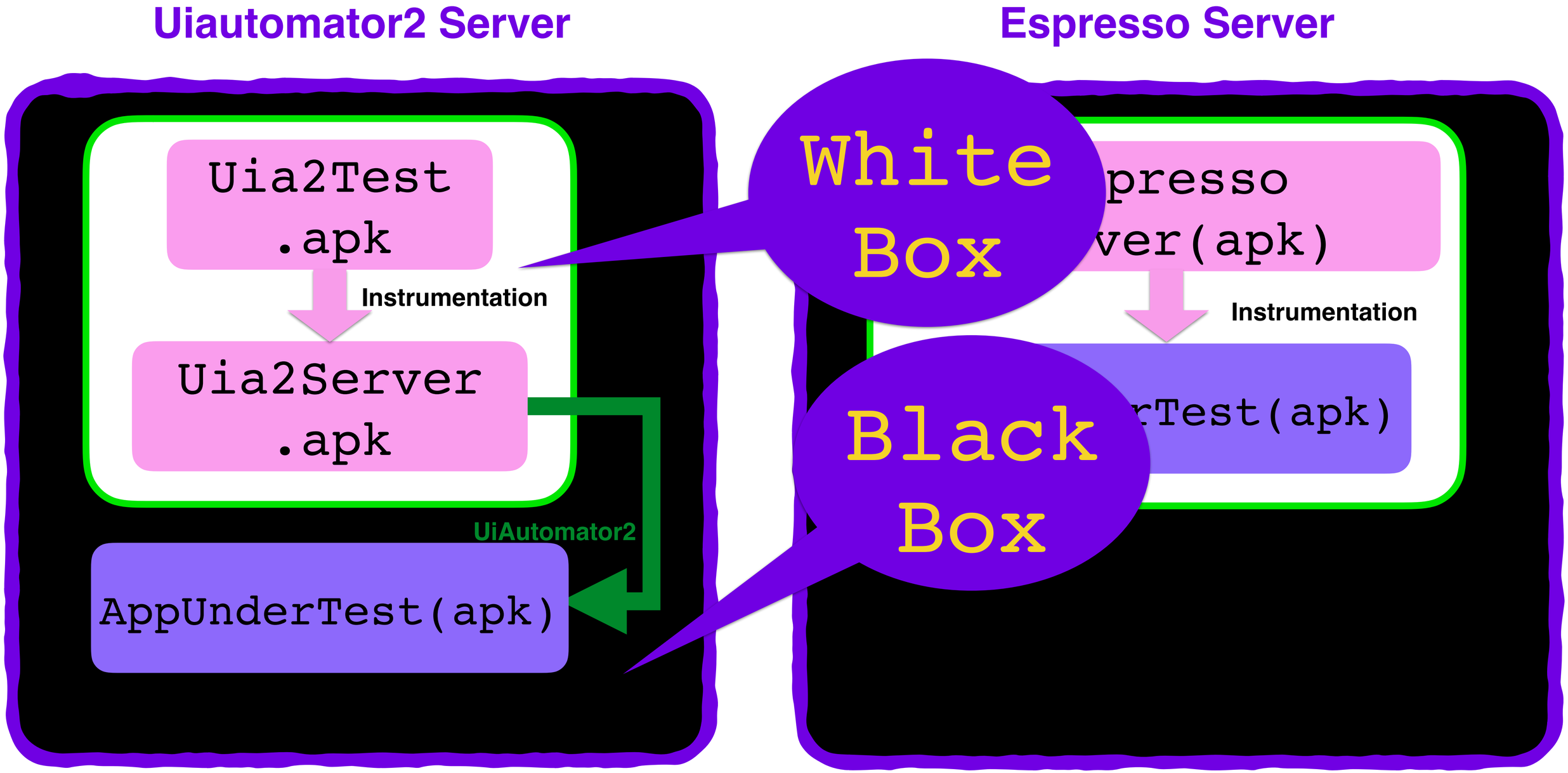

Automation Code
Android Device
"mobile: backdoor"
Appium Client
APPIUM SERVER
/backdoor
Espresso Server(apk)
Instrumentation
AppUnderTest(apk)
Other apps


Automation Code
Android Device
APP
SERVER
or"
/backdoor
Espresso
Server(apk)
Instrumentation
AppUnderTest(apk)
Other apps
BFF!


Automation Code
Android Device
Appium Client
APPIUM SERVER
Espresso Server(apk)
AppUnderTest(apk)
Other apps
UiAutomator2


Automation Code
Android Device
APPIUM SERVER
Appium Client
Espresso Server(apk)
AppUnderTest(apk)
Other apps
UiAutomator2

Appium UiAutomator Server
VS
Appium Espresso Server


Uiautomator2 Server
Espresso Server
Uia2Test
.apk
Instrumentation
Uia2Server
.apk
UiAutomator2
AppUnderTest(apk)
White
Box
Black
Box
Instrumentation

Appium ElementBackdoor (View Functions)

- Espresso element is stored as **`android.view.View`** object on server side
- Can call public methods on these views
- Can be called in a chain
- Can be passed arguments

Other types of backdoors

1. `target: 'element'`
2. `target: 'activity'`
3. `target: 'application'`

Other types of backd

1. target: 'el
2. target: 'activity'
3. target: 'application'

need to
define custom
method in Activity Or
Application class

References

- https://github.com/rajdeepv/appium_backdoors
- https://developer.android.com/reference/android/app/Instrumentation.html
- https://developer.xamarin.com/guides/testcloud/calabash/working-with/backdoors/

Thank you!

Rajdeep Varma

 mail.rajvarma@gmail.com

 rajdeepv

 @varmarajdeep

badoo

Btw, we are hiring



badoo_tech

© 2019