

ThoughtWorks®

FOSDEM 17

Future of Mobile Automation, Appium Steals it

Agenda

- Mobile Element Inspectors
- Need for Speed
- Handling Multiple Simulators
- Wearables/tvOS Support
- StarDriver Enterprise
- Mobile First to AI First

ThoughtWorks®

Inspectors



Inspectors

- ❑ WebDriverAgent Inspector
 - ❑ Support only for iOS 9.4+ devices
 - ❑ Not for Android Apps
- ❑ UIAutomatorViewer
 - ❑ Work's perfectly fine for Android Platform
 - ❑ Doesn't support Multi Window searches for API21+
- ❑ Macaca Inspector
 - ❑ Inspects both iOS and Android Apps
 - ❑ Doesn't render fields in UIAutomation format
- ❑ Appium Desktop Inspector
 - ❑ Still in development
 - ❑ Supports both iOS and Android Apps
 - ❑ Integrated with Appium Server
 - ❑ Interactive GUI

Appium Desktop Inspector

The screenshot displays the Appium Desktop Inspector interface, which is used for inspecting mobile applications. The interface is divided into several panels:

- Mobile App View (Left):** Shows a mobile application interface with a green header bar containing "Back" and "Login" buttons. The main content area features the "VODQA" logo, a text input field with the value "admin", a green password input field with masked characters "*****", and a blue "LOG IN" button. The bottom of the screen shows a standard Android navigation bar.
- Source Panel (Middle):** Displays the HTML structure of the application. The tree shows a sequence of Android widget classes: `<android.widget.FrameLayout>`, `<android.widget.LinearLayout>`, `<android.widget.FrameLayout>`, `<android.widget.FrameLayout>`, `<android.view.View>`, `<android.view.View>`, `<android.view.View>`, `<android.widget.ImageView>`, `<android.widget.EditText>`, `<android.widget.EditText>` (highlighted), `<android.view.View>`, `<android.view.View>`, and `<android.widget.Button>`. Below the tree, the "Attributes" section lists properties for the selected `<android.widget.EditText>` element, including `index: 2`, `text:`, `class: android.widget.EditText`, `package: com.vodqareactive`, `content-desc: password`, `checkable: false`, `checked: false`, `clickable: true`, `enabled: true`, `focusable: true`, `focused: false`, and `scrollable: false`. To the right of the attributes are buttons for "Tap Element", "Enter keys", and "Send Keys".
- Elements Panel (Right):** Shows the raw HTML code of the application. The code includes a DOCTYPE declaration, an HTML tag, a head section, and a body section. The body contains a `<div id="root">` element, which includes a `<script>` tag and a `<div id="root">` element. The selected `<div id="root">` element is highlighted in the code.
- Styles Panel (Bottom Right):** Displays the CSS styles for the selected element. The "body" style is selected, showing properties like `color: #222`, `font-size: 16px`, `border: 1px solid #ccc`, and `padding: 10px`. A visual representation of the element's box model is shown, with dimensions of 874 x 778. The "Properties" section lists various CSS properties and their values, including `background-color`, `border-color`, `border-style`, `border-width`, `color`, `cursor`, and `display`.

ThoughtWorks®

**NEED FOR
SPEED™**
NO LIMITS

Drive Faster



NFS

- Run Tests Faster than before
 - Dockerize Android Environment to scale up and scale down emulators in seconds
 - Speed up tests on iOS Platform by improving locator finding strategies, etc
 - Parallel execution of tests on iOS sims
 - Multiple server sessions from single node instance

ThoughtWorks®



Handle Multiple Sims

Handling Multiple Simulators

- Existing Limitations:
 - Xcode supports only one simulator at a time
- Existing Solutions which Appium evaluates:
 - **Project Hydra** - Python Wrapper to run tests in multiple sims parallelly
 - Doesn't have access to Core Sim APIs
 - Depends on less active Facebook's XCtool Project
 - **FBSimulatorControl**
 - OS X Library for managing and manipulating iOS Sims
 - **BluePill**
 - iOS test runner to run tests on multiple simulators parallelly



ThoughtWorks®

Wearables/tvOS Support

Wearables/tvOS Support

- Android Wear Apps are possible to test using many existing frameworks
- Apple Watch and tvOS apps are still a concern for many
- Facebook's WebDriverAgent works with tvOS and OSX apps but not supported yet.
- Cross Platform video apps developed using you.i Engine can be automated using Appium for tvOS & Android TV
 - <https://github.com/YOU-i-Labs/appium-youiengine-driver>

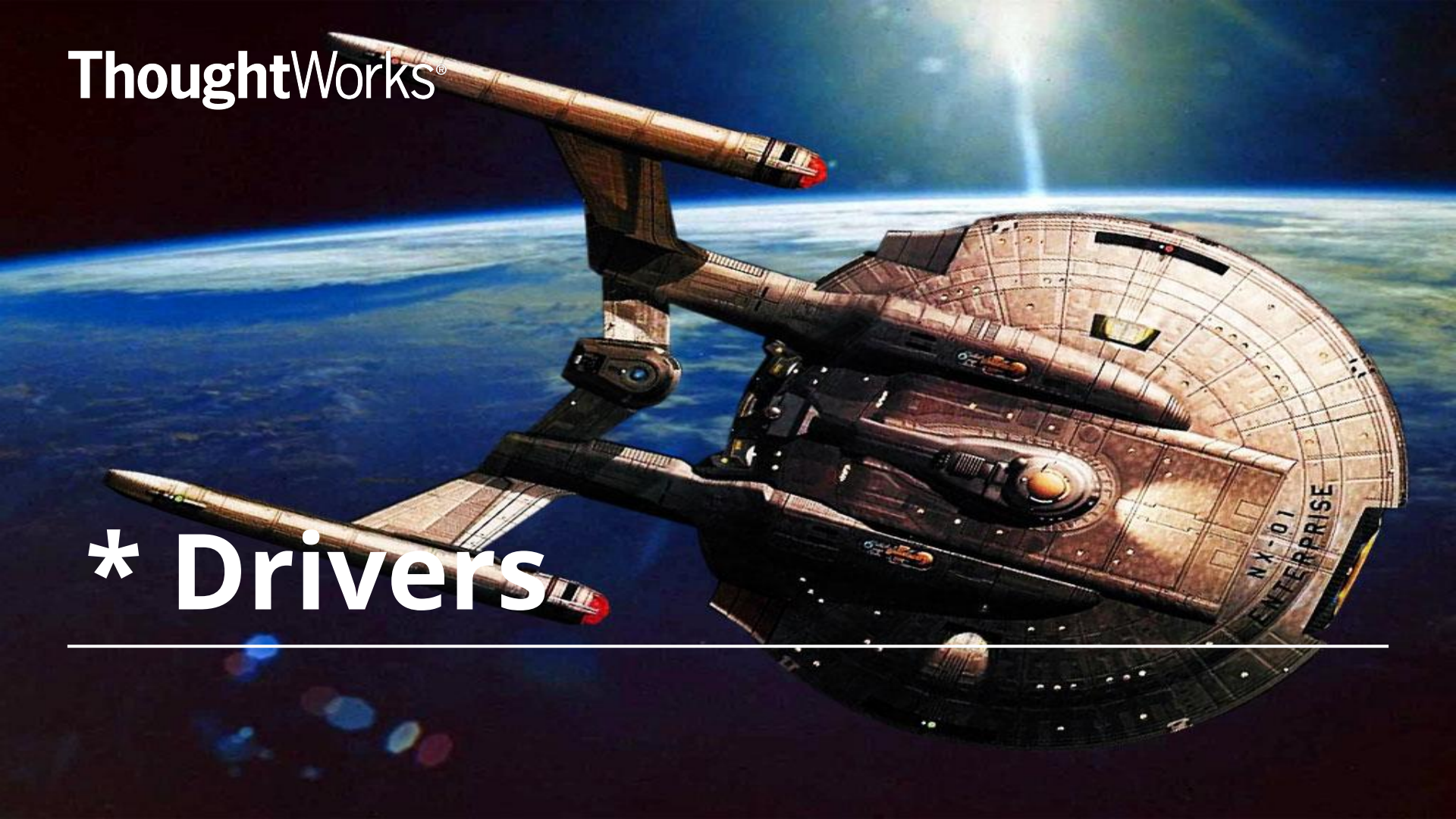


Move towards *Driver Enterprise

*Automate Anything & Everything - *driver*

ThoughtWorks®

* Drivers



* Drivers

- Enable Automation for everything
- Generalise W3C Spec
- Become locked in an external cycle of writing code to automate new platforms while convincing vendors of old platforms to adopt the standard
- Appium will be a standard interface for array of *Drivers
- *Drivers will be a node wrapper written on top of any existing technical solutions to automate any crazy things.
- *Drivers might be watchDriver, IOTDriver, UMPDriver etc

ThoughtWorks®

Mobile first to AI first





THANK YOU

Srinivasan Sekar

ThoughtWorker

Appium Member

sekars@thoughtworks.com

ThoughtWorks®