Remote iOS Device Server

Scaling iOS testing

Nikolai Abalov

badoo

badoo tech

Hello, my name is

Nikolai Abalov

Senior Automation Engineer at Badoo

Previously Senior Developer and Automation Engineer at 2GIS for 3 years

Co-creator of Winium.Mobile

From Siberia ❄

github.com/NickAb
@NickAb

Agenda

1. Introduction
2. Parallel tests on iOS
3. Device Server
4. Results & next steps
5. Tips & Tricks — how to apply it at your scale
6. Recap

Introduction

Our automation stack

- calabash-ios
- cucumber-ruby
- ruby
- WebDriverAgent
- parallel_cucumber

What we had

- 1300+ end-to-end tests
- 35+ hours of cpu-time
- 1.5 hours wall-clock time
- 10+ Pull Requests per day to the app repo
- 30+ Test runs per day
- Extra projects on the same infrastructure
- 1500+ cpu-time hours per day

What we want

- Increase trust in automation
- Fast and reliable feedback

What we want

- 30 minutes wall-clock
- Stability
- Scalability
- Easy maintenance and development
- Simulators & physical devices

How to get there

- Delete ⅔ of tests

Questions?

badoo

badoo tech

But really

- Filter tests based on change set, coverage, components
- Speed up and stabilise tests
- Heavily parallelise test runs
- Provide scalable infrastructure to run tests

Parallel tests on iOS

Test parallelisation

$1300_{\text{tests}} / 1_{\text{simulator}} = 40_{\text{hours}}$

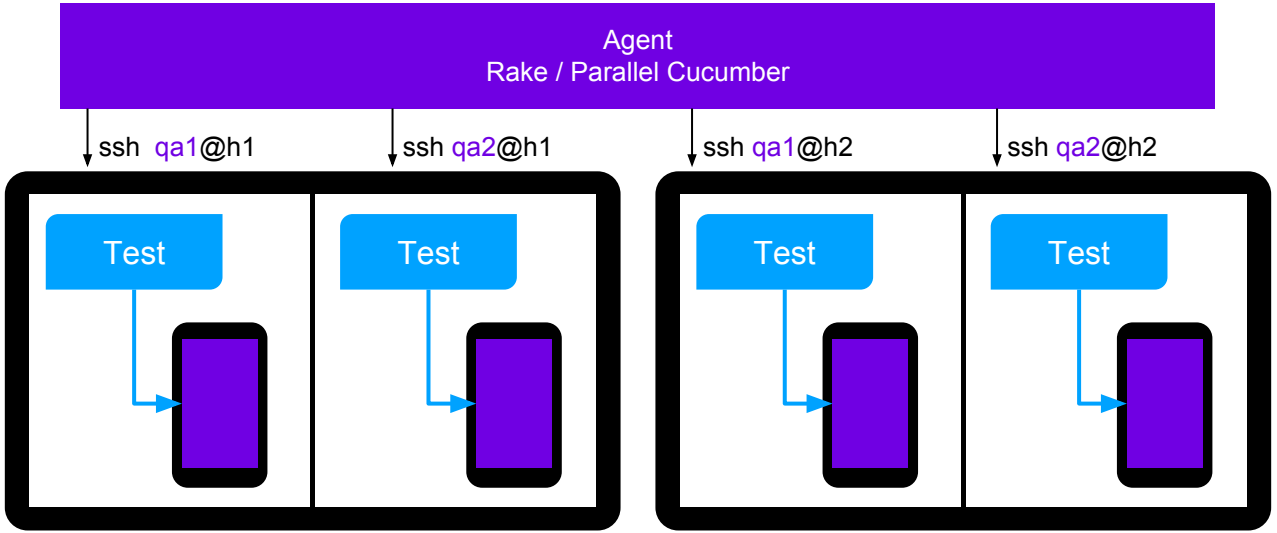
$1300_{\text{tests}} / X_{\text{simulators}} = 0.5_{\text{hours}}$

$X = 80_{\text{simulators}}$

Options

- 3rd party cloud
  - SauceLabs / TestObject
  - Xamarin Test Cloud
  - AWS Device Farm
- In-house

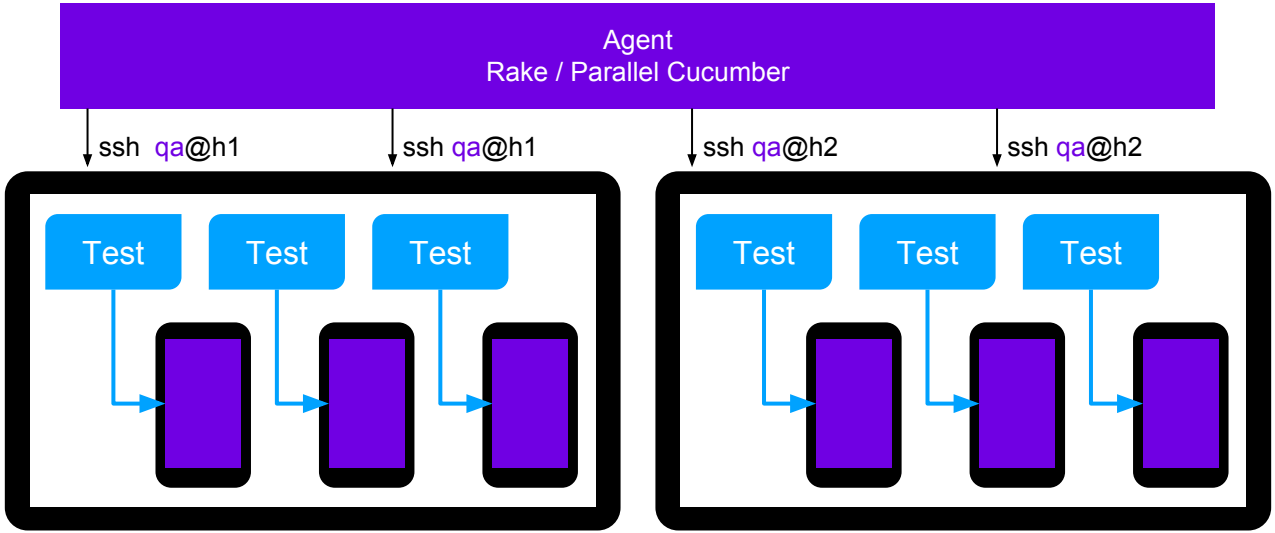
Multiple users, simulator per user



macOS,
Xcode,
Ruby,
…

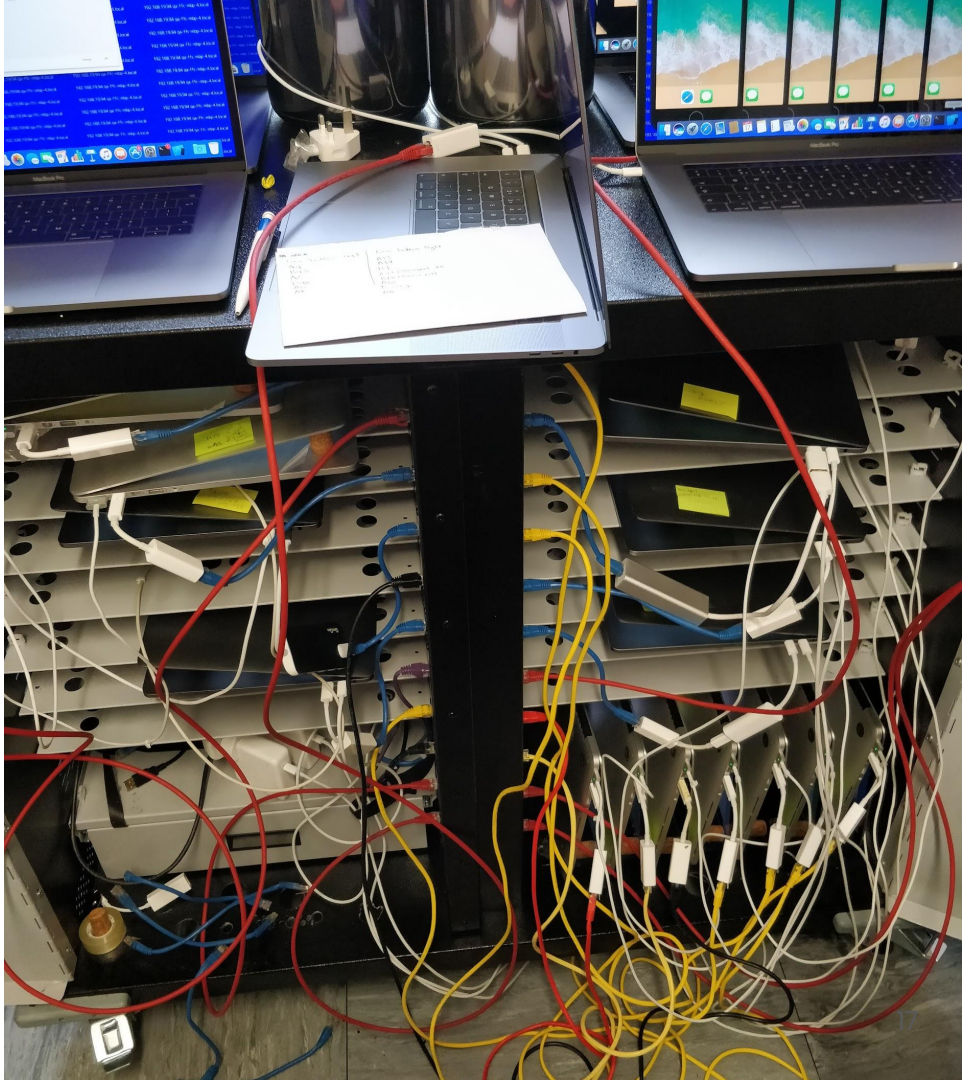
[iOS Calabash in parallel on one host](#) by Tim Baverstock

Single user, multiple simulators



macOS,
Xcode,
Ruby,
…

[iOS Parallel Automation: run faster than fast] by Viktar Karanevich
[Turning iOS End to End Testing up to 11] by Lawrence Lomax

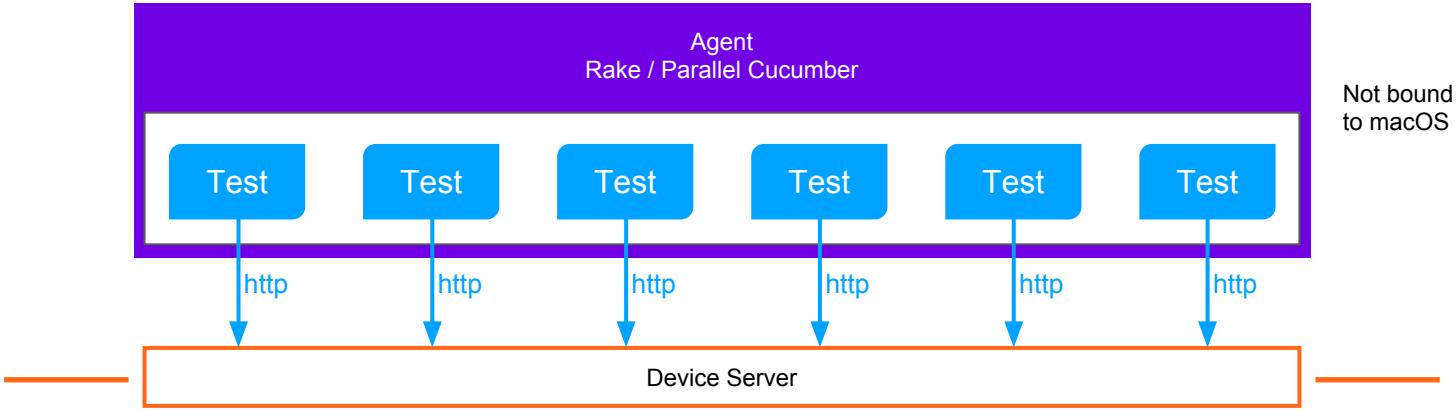


Issues with current model

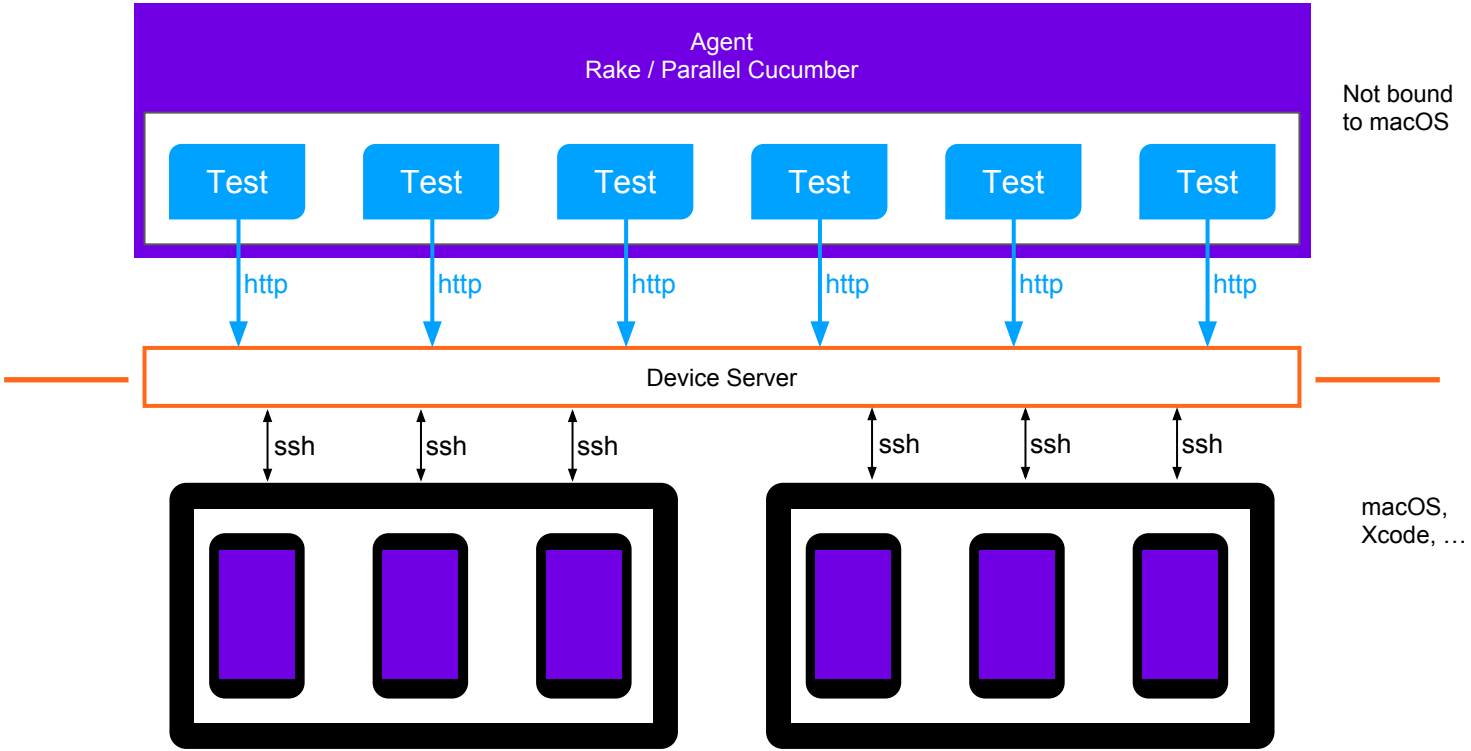
- Tight coupling of test code and infrastructure
- Hard to scale
- Tests are bound to simulator host machine
- Too easy to mess with simulator internals
- No sharing of resources between test jobs on CI

New Model

New model


Agent
Rake / Parallel Cucumber
Test
Test
Test
Test
Test
Test
http
http
http
http
http
http
Device Server
Not bound
to macOS

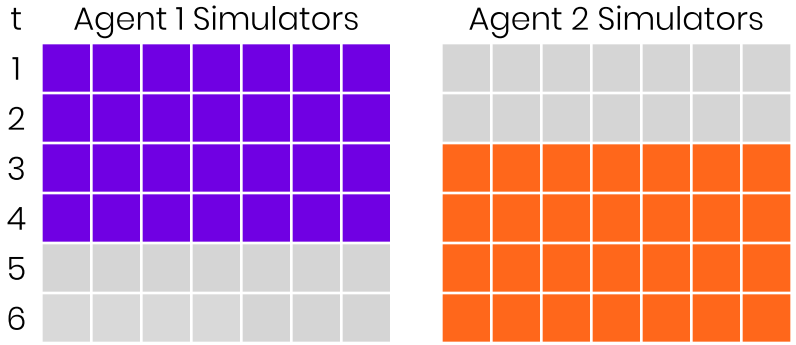
New model

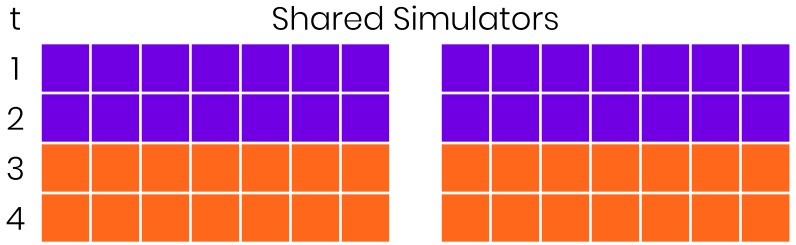

Agent
Rake / Parallel Cucumber
Test
Test
Test
Test
Test
Test
http
http
http
http
http
http
Device Server
ssh
ssh
ssh
ssh
ssh
ssh
Not bound
to macOS
macOS,
Xcode, …

New model

- Separation of concerns
- Well-defined interface
- Cheaper test environments
- Easy sharing of resources between test jobs on CI

Why move? Sharing



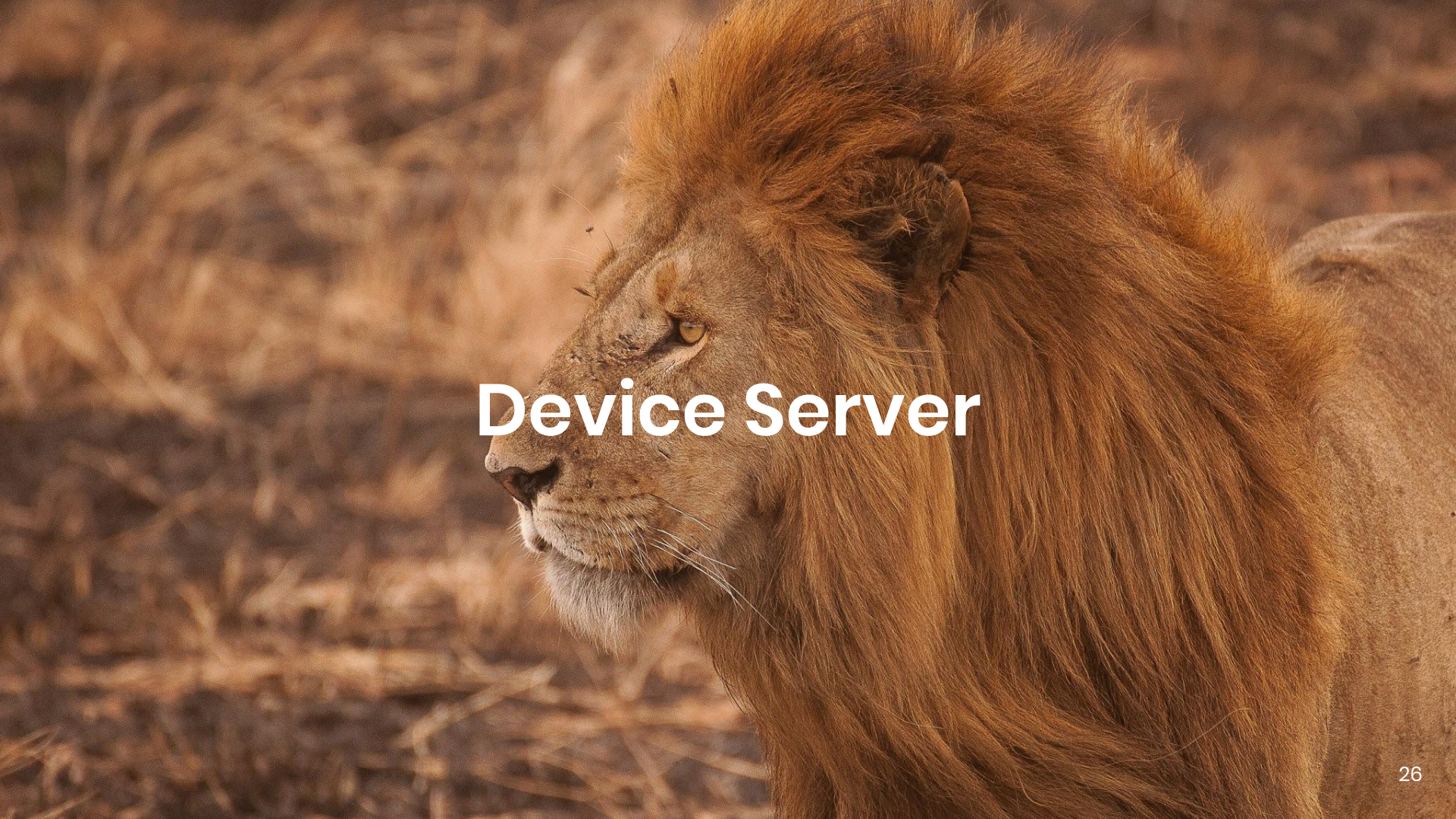


More reasons to move

- Black boxing
- No more messing with internals
- No more xcrun simctl, fbsimctl, idevice*
- Easier updates

Why not Selenium Grid?

- Existing code
- Custom actions
- Flexibility and fast prototyping


Device Server

Tools

Previous

- xcrun simctl
- fbsimctl
- WebDriverAgent
- ideviceinstaller

Current

- fbsimctl
- WebDriverAgent

fbsimctl as a server

```
$ fbsimctl $udid boot -- listen --http 9999 -- shutdown

listen started: Http: 9999 Hid: No stdin: false


$ curl http://localhost:9999/B98A0678-3724-4BF5-9A4C-E84E06DDA1DD/list

{"status":"success","subject":[{"event_type":"discrete","timestamp":1523370322,"subje
ct":{"arch":"x86_64","state":"Shutdown","model":"iPhone 6","name":"iPhone
6","os":"iOS
11.2","udid":"B98A0678-3724-4BF5-9A4C-E84E06DDA1DD"},"event_name":"list"}],"events":[
]}
```

fbsimctl over http – exposed actions

- approve
- clearKeychain
- config
- diagnosticQuery
- diagnostic
- hid
- install
- launch
- list
- listApps
- open
- record
- relaunch
- screenshot
- search
- setLocation
- tap
- terminate
- uninstall
- upload

fbsimctl over http – missing actions

- boot
- create
- delete
- erase
- shutdown
- launch_xctest
- list_device_sets

fbsimctl over http – missing actions

- boot
- create
- delete
- erase
- shutdown
- launch_xctest
- list_device_sets

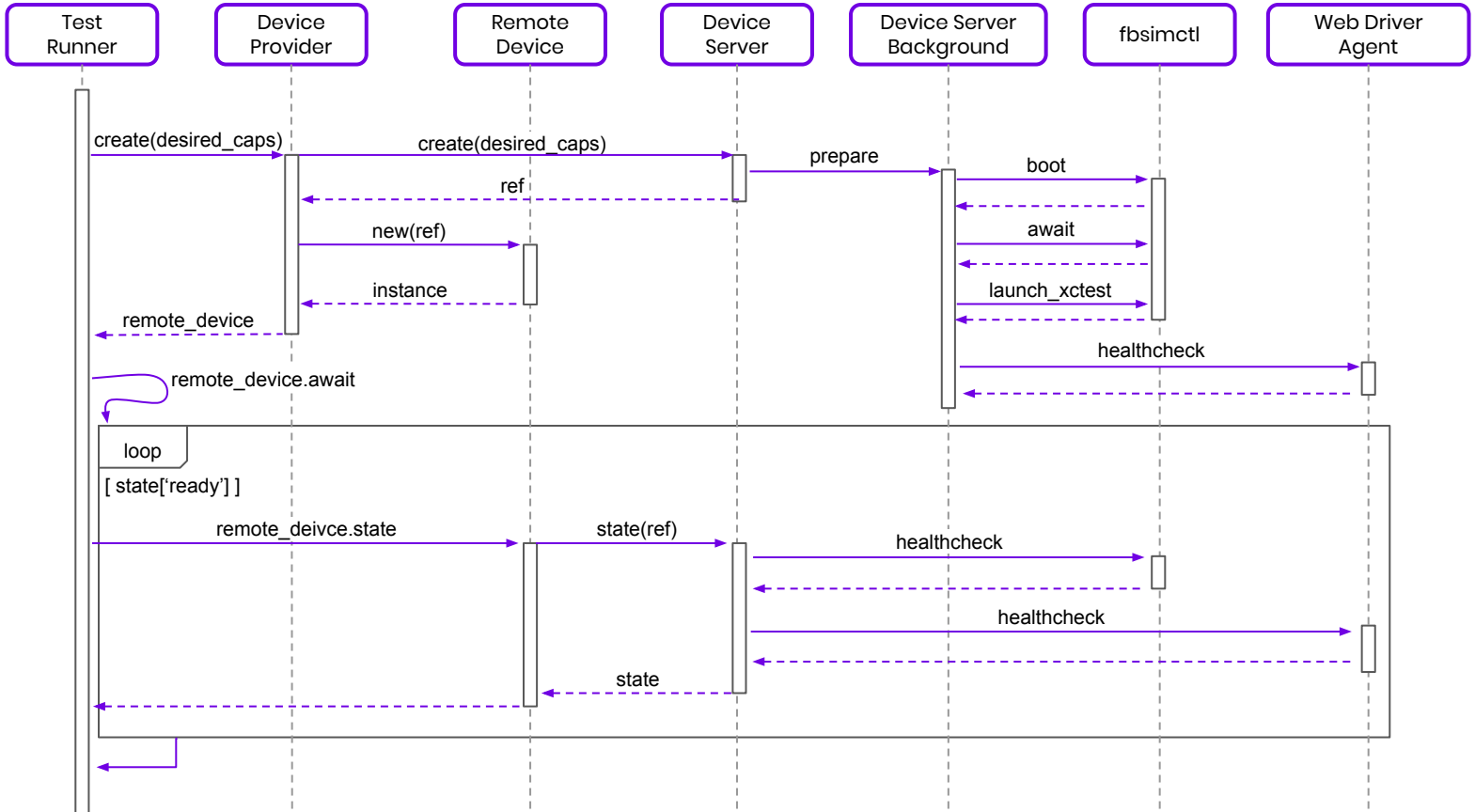
Additional requirements

- Prepare and reset simulators on demand
- Start WebDriverAgent on simulators
- Hide away satisfying desired requirements
- Extra actions
- Pool simulators

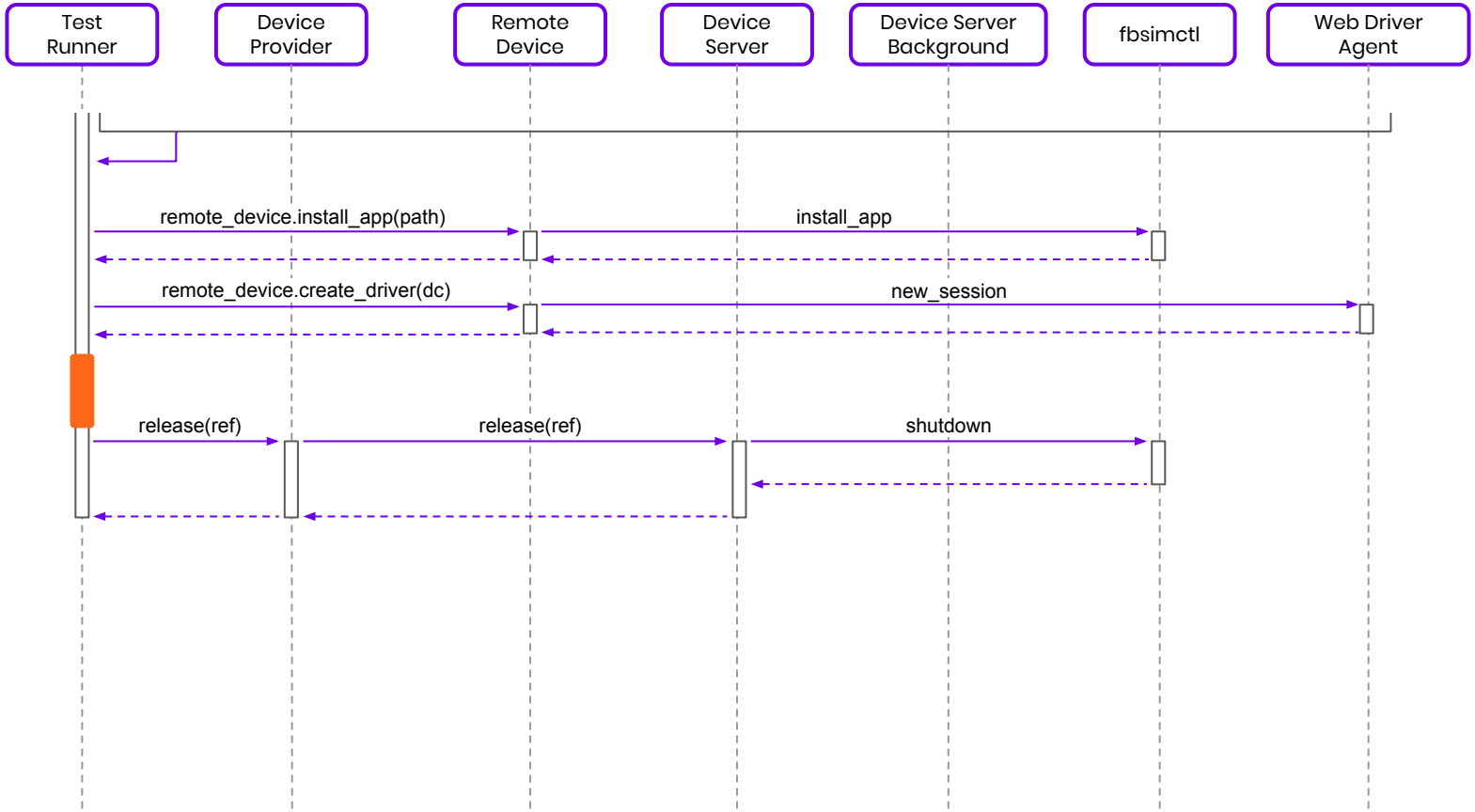
Remote Device API (Subset)

```
class RemoteDevice
  ...
  def reset(timeout: 60); end
  def clear_safari_cookies; end
  ...
  def app_installed?(bundle_id); end
  def create_driver(caps); end
  ...
  def video_acquire; end
  def video_start; end
  def video_stop; end
  def last_crash_log; end
```

Device Server model



Device Server model



Results & Next Steps

Results

- 30 → 60 machines hosting simulators
- 1.5 hours → 0.5 hours
- Decoupled tests and infrastructure
- Eased updates
- Simplified test runner

Next Steps

- Open-source
- Remove ssh

Tips & Tricks

Simulators per host

MacBook Pro

4 Cores Intel Core i7 @ 2.8 GHz,

16 GB RAM

6 simulators

£ 2,349.00

1 simulator = £ 391.50

Mac Pro

12 Cores Intel Xeon E5-2697 @ 2.7 GHz,

64 GB RAM

18 simulators (3 VMs × 6 simulators)

£ 5,879.00

1 simulator = £ 326.61

ESXi = £ 0.00

Resetting Simulator

Option 1: Straightforward

```
fbsimctl ${UDID} shutdown -- erase -- boot
```

Full boot ~ 18 seconds

Resetting Simulator

Option 2: Not so straightforward

```
cd $(fbsimctl list_device_sets)
fbsimctl ${UDID} shutdown
rm -rf ${UDID}
cp -rp ${UDID}_BACKUP ${UDID}
fbsimctl ${UDID} boot
```

Full boot ~ 8 seconds

Reset Cookies

```
$ fbsimctl B98A0678-3724-4BF5-9A4C-E84E06DDA1DD list_apps | grep MobileSafari

Bundle Name: MobileSafari | ID: com.apple.mobilesafari | Install Type system | ⤸
Container /Users/qa/Library/Developer/CoreSimulator/Devices/ ⤸
B98A0678-3724-4BF5-9A4C-E84E06DDA1DD/data/Containers/Data/Application/ ⤸
22D18402-3D7B-4F28-88CF-98935328B0FB

$ rm -f /Users/qa/Library/Developer/CoreSimulator/Devices/ ⤸
B98A0678-3724-4BF5-9A4C-E84E06DDA1DD/data/Containers/Data/Application/ ⤸
22D18402-3D7B-4F28-88CF-98935328B0FB/Library/Cookies/Cookies.binarycookies
```

Setting Up Simulator Host

```
# enable on-screen keyboard
defaults write com.apple.iphonesimulator ConnectHardwareKeyboard -bool false

# disable location
defaults write com.apple.iphonesimulator LocationMode "3101"

# turn off chrome
defaults write com.apple.iphonesimulator ShowChrome -bool false

# EnableKeyboardSync
# GraphicsQualityOverride
# OptimizeRenderingForWindowScale
# SlowMotionAnimation
```

Setting Up Simulator Host

```
# enable on-screen keyboard
defaults write com.apple.iphonesimulator ConnectHardwareKeyboard -bool false

# disable location
defaults write com.apple.iphonesimulator LocationMode "3101"

# turn off chrome
defaults write com.apple.iphonesimulator ShowChrome -bool false

# EnableKeyboardSync
# GraphicsQualityOverride
# OptimizeRenderingForWindowScale
# SlowMotionAnimation
```

Setting Up Simulator Host

```
# enable on-screen keyboard
defaults write com.apple.iphonesimulator ConnectHardwareKeyboard -bool false

# disable location
defaults write com.apple.iphonesimulator LocationMode "3101"

# turn off chrome
defaults write com.apple.iphonesimulator ShowChrome -bool false

# EnableKeyboardSync
# GraphicsQualityOverride
# OptimizeRenderingForWindowScale
# SlowMotionAnimation
```

Setting Up Simulator Host

```
# enable on-screen keyboard
defaults write com.apple.iphonesimulator ConnectHardwareKeyboard -bool false

# disable location
defaults write com.apple.iphonesimulator LocationMode "3101"

# turn off chrome
defaults write com.apple.iphonesimulator ShowChrome -bool false

# EnableKeyboardSync
# GraphicsQualityOverride
# OptimizeRenderingForWindowScale
# SlowMotionAnimation
```

Managing Infrastructure

- 64 hosts to manage + 6 TeamCity agents
- xcversion https://github.com/KrauseFx/xcode-install
- Ansible playbook to update fbsimctl, xcode, deploy configs, etc
- Ansible to remove outdated simulators after updates

Managing Infrastructure

```
# do once on a storage host
xcversion install 9.2 --no-install --no-clean
xcversion simulators --install="iOS 11.0" --no-install

ls ~/Library/Caches/XcodeInstall/

cd ~/Library/Caches/XcodeInstall/
python -m SimpleHTTPServer
```

Managing Infrastructure

```
# repeat for each work host
xcversion install 9.2 --url=http://$host:$port/Xcode_9.2.xip

curl http://$host:$port/com.apple.pkg.iPhoneSimulatorSDK….dmg --output ⤸
     ~/Library/Caches/XcodeInstall/com.apple.pkg.iPhoneSimulatorSDK….dmg
xcversion simulators --install="iOS 11.0"
```

Recap

Stability, reliability, speed

Separation of concerns

Know-Hows

Thank you! Questions?
Nikolai Abalov
@NickAb
badoo
badoo tech