



TESTINGMIND

Scaling your device lab using cloud solutions

Martin Schneider

#TAS19

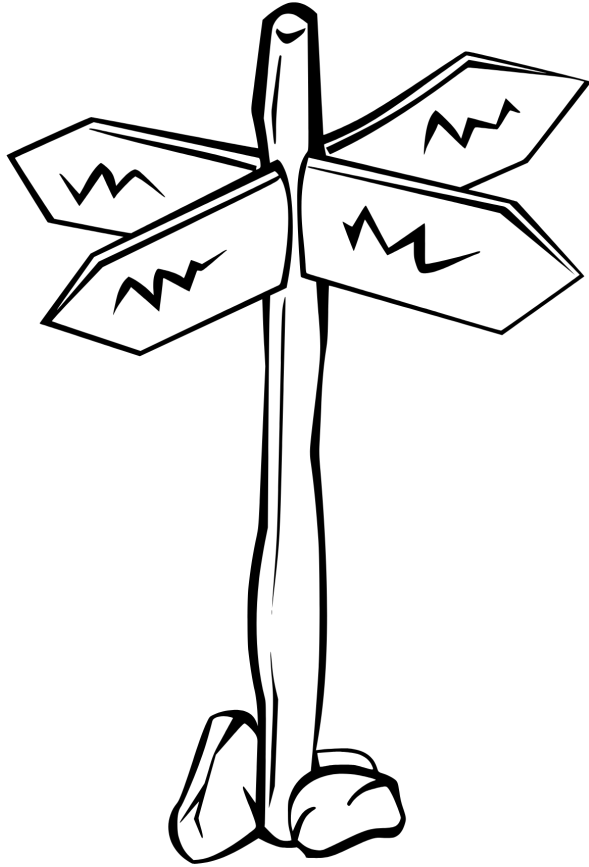
Singapore
July 25th, 2019

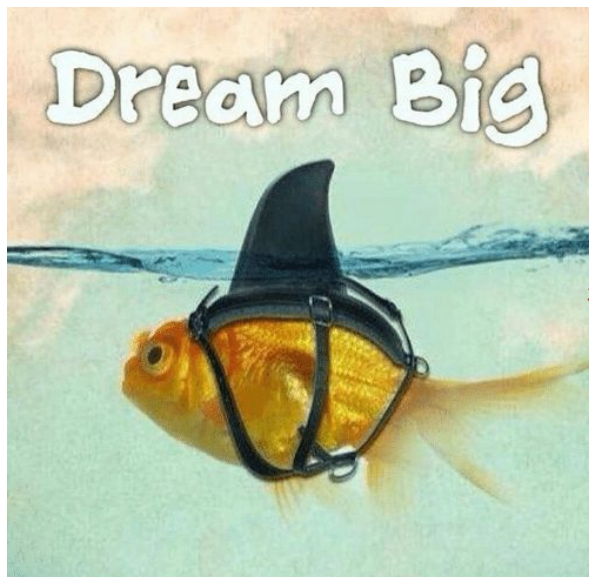
Where are we?

- UI automation PoC
- Some first test cases
- Techstack stable and framework ready
- Initial thoughts about CI
- **Now what next**



Quo vadis?





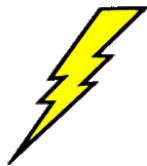
SAUCE LABS



pCloudy.com

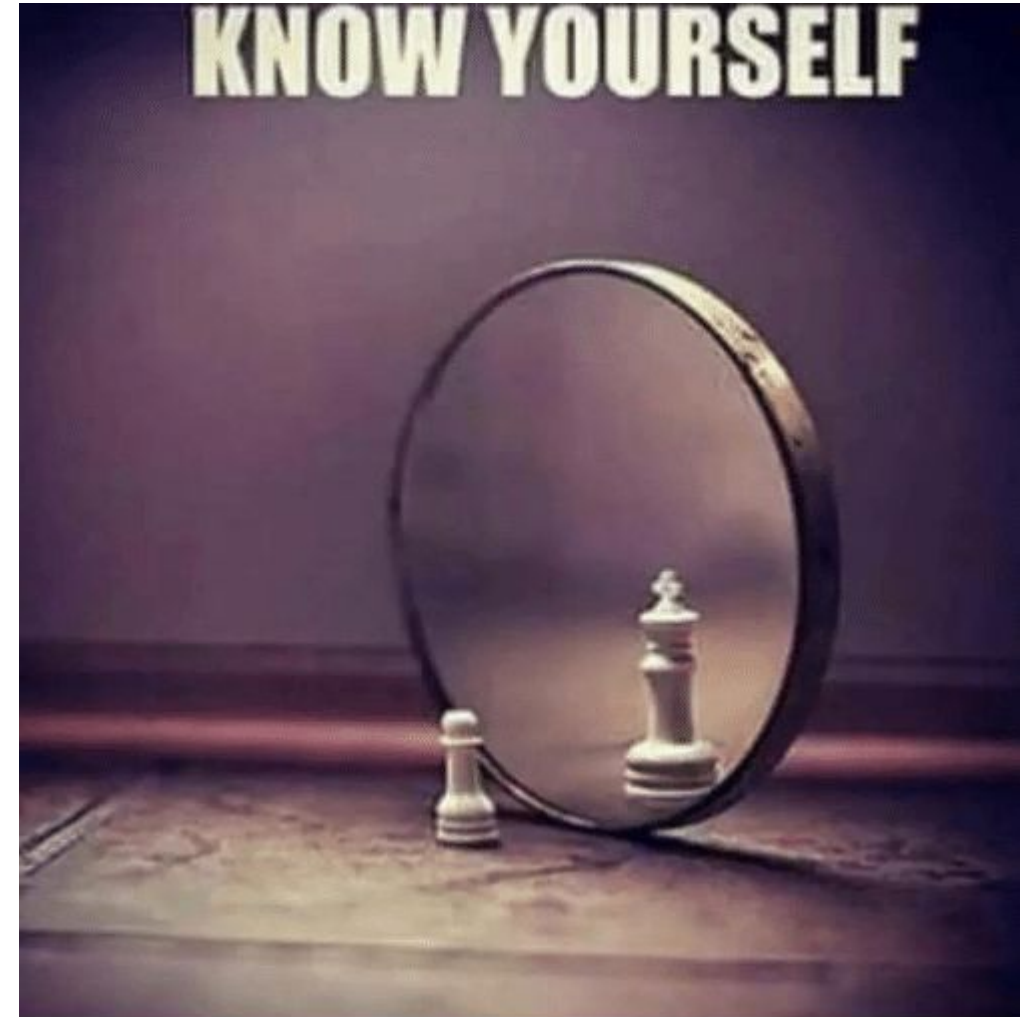
On premise vs. cloud



<ul style="list-style-type: none">● Full control<ul style="list-style-type: none">○ Number of devices○ Types of devices○ Software libraries○ ...	<ul style="list-style-type: none">● No maintenance required● Fixed costs● Support
<ul style="list-style-type: none">● Maintenance<ul style="list-style-type: none">○ replace devices○ fix issues○ infrastructure○ ... 	<ul style="list-style-type: none">● Less flexibility<ul style="list-style-type: none">○ Available devices○ Software update cycle? (Appium, Java)

Selection criteria

Understand your use-case!



Selection criteria

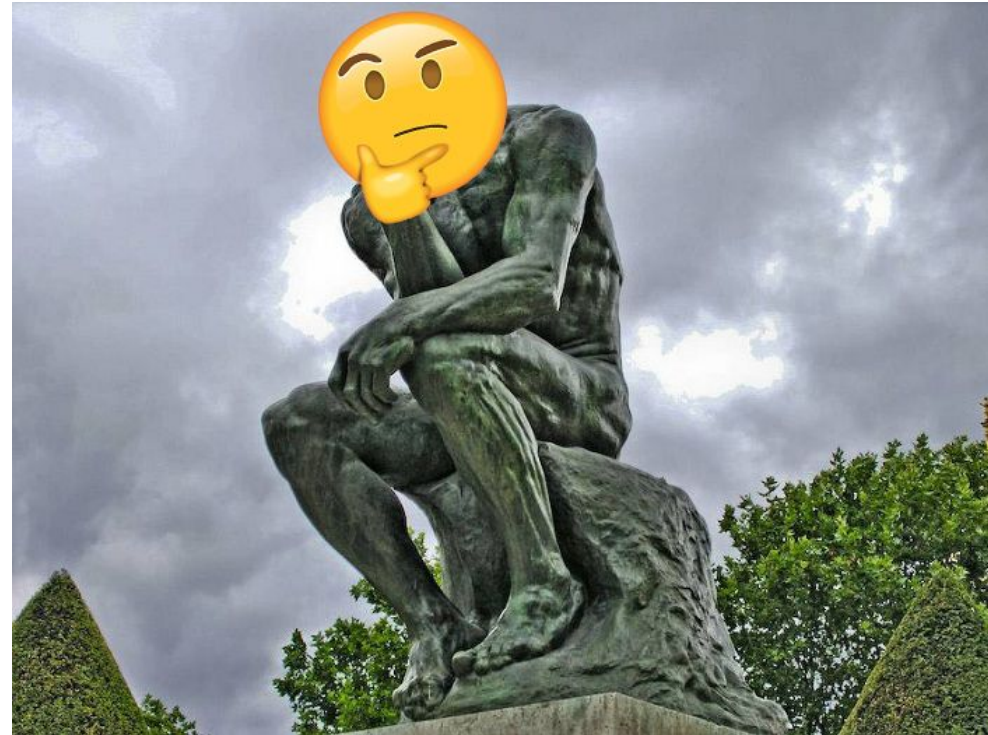
- **Time constraints, execution speed, waiting times**
 - run tests once a day (“nightly tests”) → not time-critical
 - run tests multiple times a day to verify pull requests → highly time-critical
- **Parallelisation and Distribution**
 - How to distribute n tests across m devices efficiently (each test runs once)?
 - How to run n tests on m devices in parallel (each test runs m times)?

Selection criteria

- **API support**
- **Cost and pricing models**
 - per device or per device minutes
- **Future use-cases**
- **Manual testing?**
- **Devices**
 - specific models, manufacturers, OS versions
- **Client- vs. server execution mode**

Disclaimer

My thoughts are my own
and not the views of my
employer.



Two cloud testing approaches

#1: Client-side execution

Step 1: Upload your APP to the cloud

Step 2: Create a WebDriver instance

```
capabilities.setCapability("app", appUrl);  
new AppiumDriver("http://cloud.com/wd/hub",  
capabilities);
```

Step 3: Execute your tests as before

For example, Browserstack, Saucelabs, pCloudy...



Two cloud testing approaches

#2: Server-side execution

Step 1: Upload your APP to the cloud

Step 2: Package and upload your tests to the cloud


Step 3: Tests get executed on the cloud

For example, AWS Devicefarm (public cloud)



Client vs. server-side execution

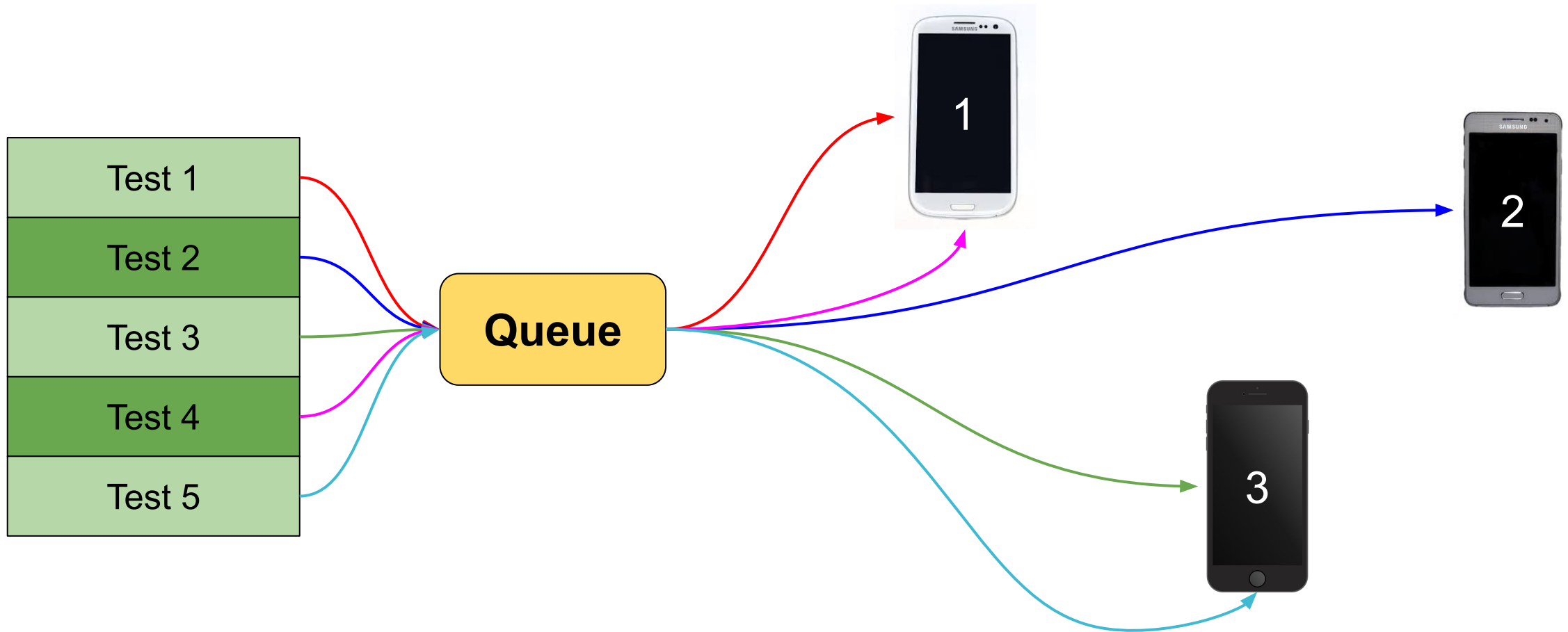


- 
- Easier migration
 - Execution flow remains the same
 - Reporting and CI remain the same
 - More control
 - Software updates on the client side
 - easier to use a mixed/hybrid mode (=use different cloud providers or local + cloud)

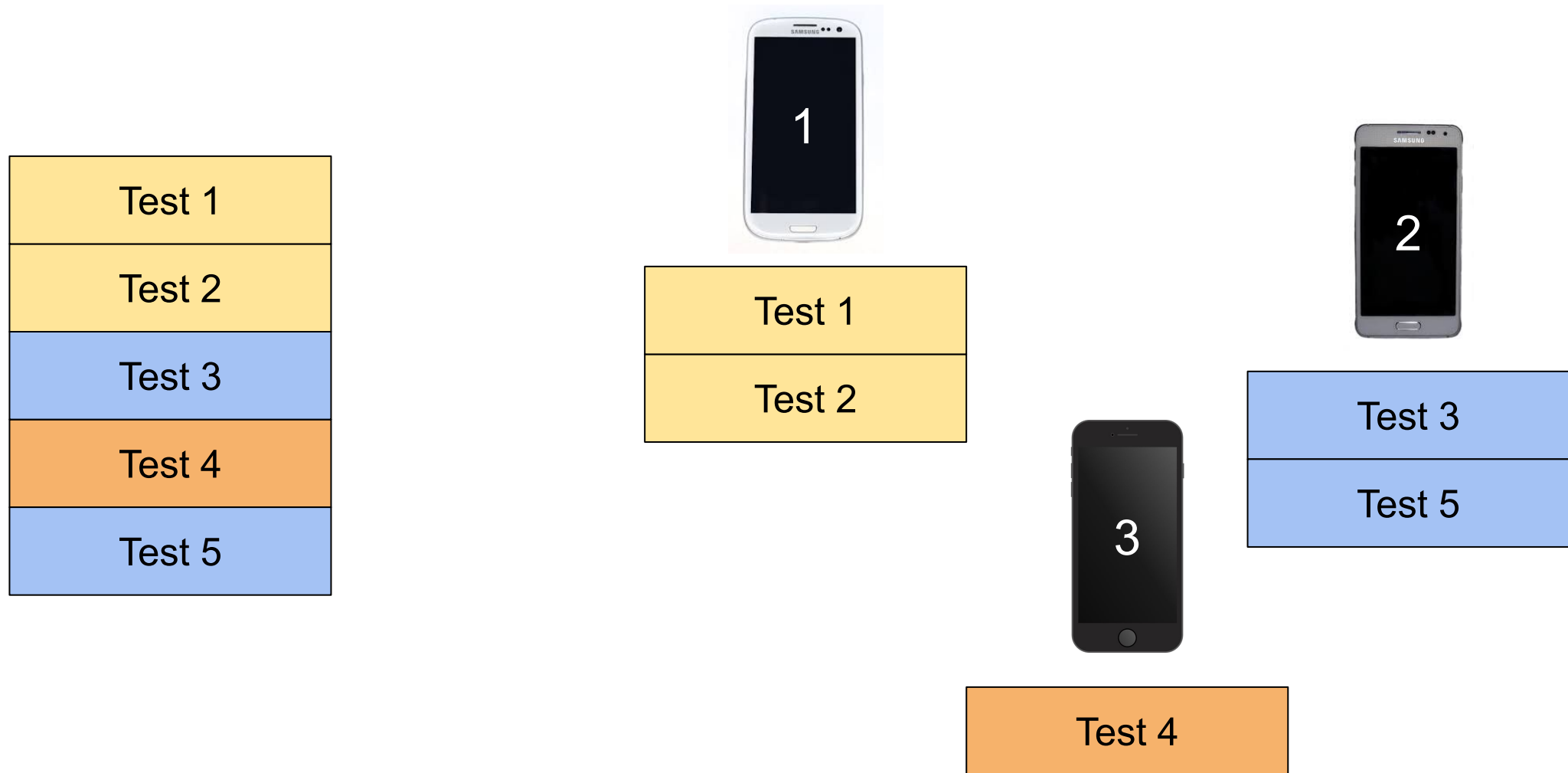
- 
- Network latency (?)

- Performance (?)
-
- Trust and compliance
 - Integration overhead
 - Test results and reporting
 - Network challenges

Test case distribution - Queue



Test case distribution - Chunks



Distribution support

Re: Running test in parallel



Posted by:  RohanD@AWS

Posted on: Mar 28, 2016 10:12 PM

 in response to: [DexRobinson](#)

Hi,

The feature you are asking for ("test sharding") isn't available in AWS Device Farm at this time. We've heard this request from several customers and will look into adding support.

Regards
Rohan

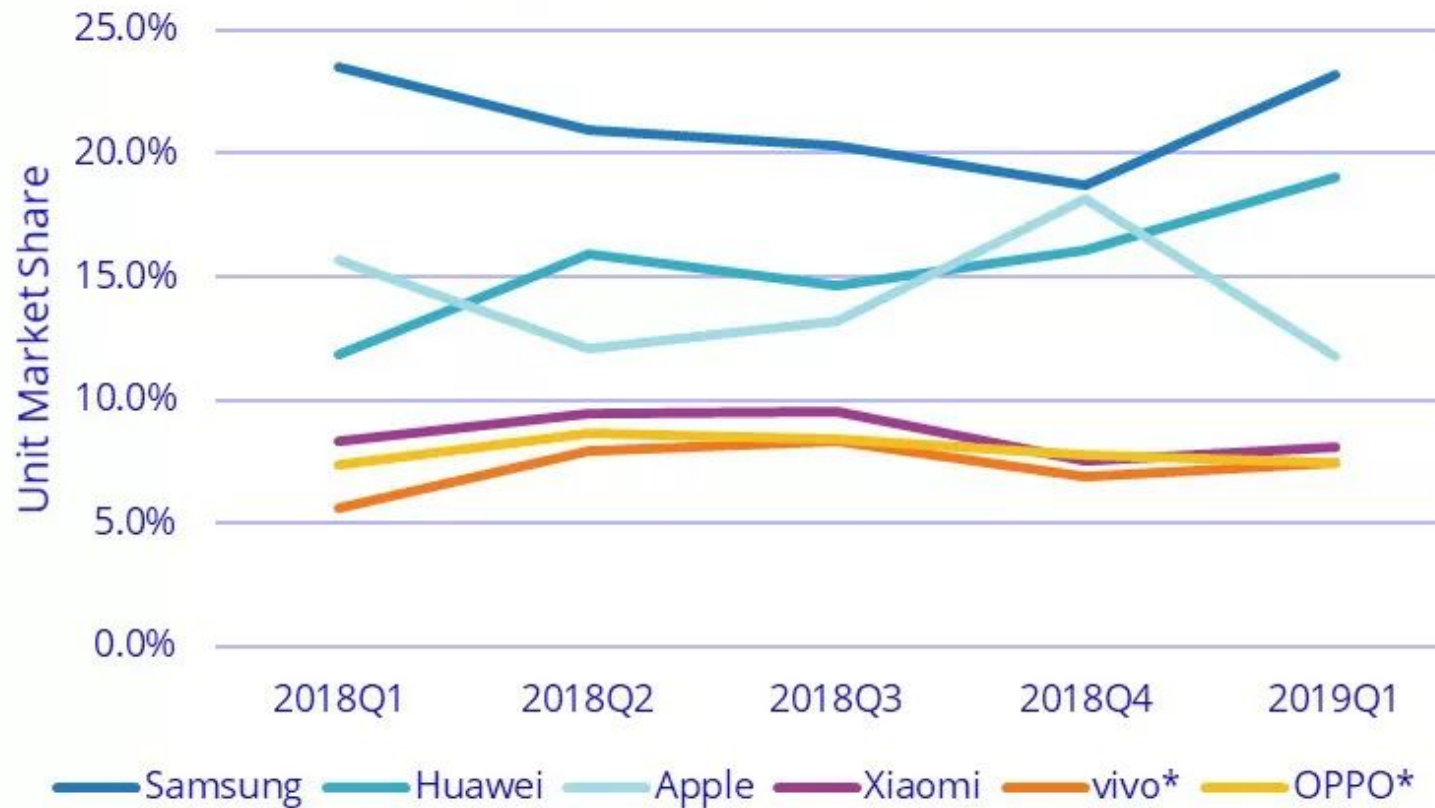
Device models

Device Vendor	Rank	Traffic %
Apple	1	47.23
Samsung	2	30.75
Huawei	3	5.4
Oppo	4	5.25
Xiaomi	5	3.17
Google	6	1.35
Unknown	7	0.91
Sony	8	0.81
LG	9	0.74
Asus	10	0.49

- In Singapore, Apple and Samsung cover 78% of the market (US 79%, UK 84%, Hong Kong 70%, Australia 87% etc.)
- What about Huawei, Oppo, Xiaomi, Google...?

Device models

Worldwide Top 5 Smartphone Companies,
2019Q1 Unit Market Share



Source: IDC 2019

- Huawei (13,3%) surpassed Apple (11,9%) in market share in Q2/2018
- Q1/2019: 19% global market share

Device models

Market specifics

- India: 21% Xiaomi (#2), Apple < 3%
- China: 24% Huawei (#1), Samsung < 3%
- Other local brands: Pakistan (QMobile), Bangladesh (Symphony) etc.

Device models

Availability

- AWS Device Farm doesn't have any Huawei, Xiaomi, Oppo or Vivo models (in their public cloud)
- Browserstack only supports Samsung, Google as well as iPhones (and one OnePlus model)

Sources:

- <https://www.gartner.com/en/newsroom/press-releases/2018-08-28-gartner-says-huawei-secured-no-2-worldwide-smartphone-vendor-spot-surpassing-apple-in-second-quarter>
- <https://deviceatlas.com>
- <http://gs.statcounter.com/vendor-market-share>
- https://www.browserstack.com/list-of-browsers-and-platforms/app_automate
- <http://awsdevicefarm.info>

Demos

We will run the same login test on AWS Devicefarm, Browserstack and locally.



www.justtestlah.qa

Demos

AWS Devicefarm

- package the test configuration into a test spec and upload to AWS
- upload the APK file to AWS
- package our tests (+framework) and upload a zip file to AWS
- select an available device (using device filters)
- execute the tests and process the results

Demos

Browserstack

- upload the APK file to Browserstack
- configure the execution (this includes the device selection) using Capabilities
- execute the tests and process the results

Vision

Test execution

- test definition
- required device type
- priority
- ...

enqueue test



iOS internal

Simulators

Physical



Android internal

Emulators

Physical



External vendors



Thank you!



Martin Schneider

Senior Software Engineer @
Carousell

mart.schneider@gmail.com

<https://twitter.com/martinschneider>

<https://github.com/martinschneider>