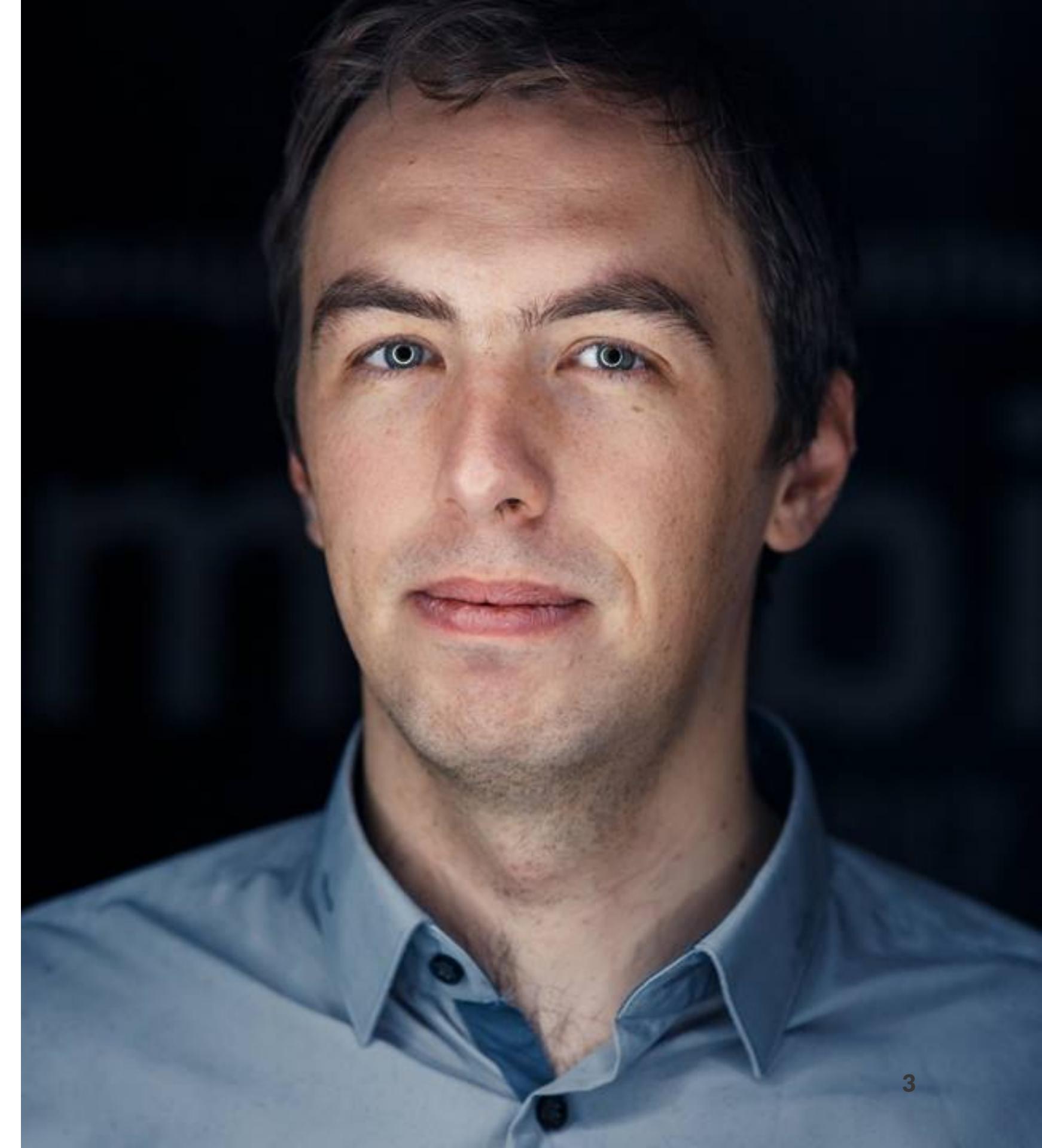


Mobile: DevOps or not DevOps

Такие чудеса,
Гораций, не
снились нашим
DevOps-ам.



My self



My self

- Vladimir Ivanov



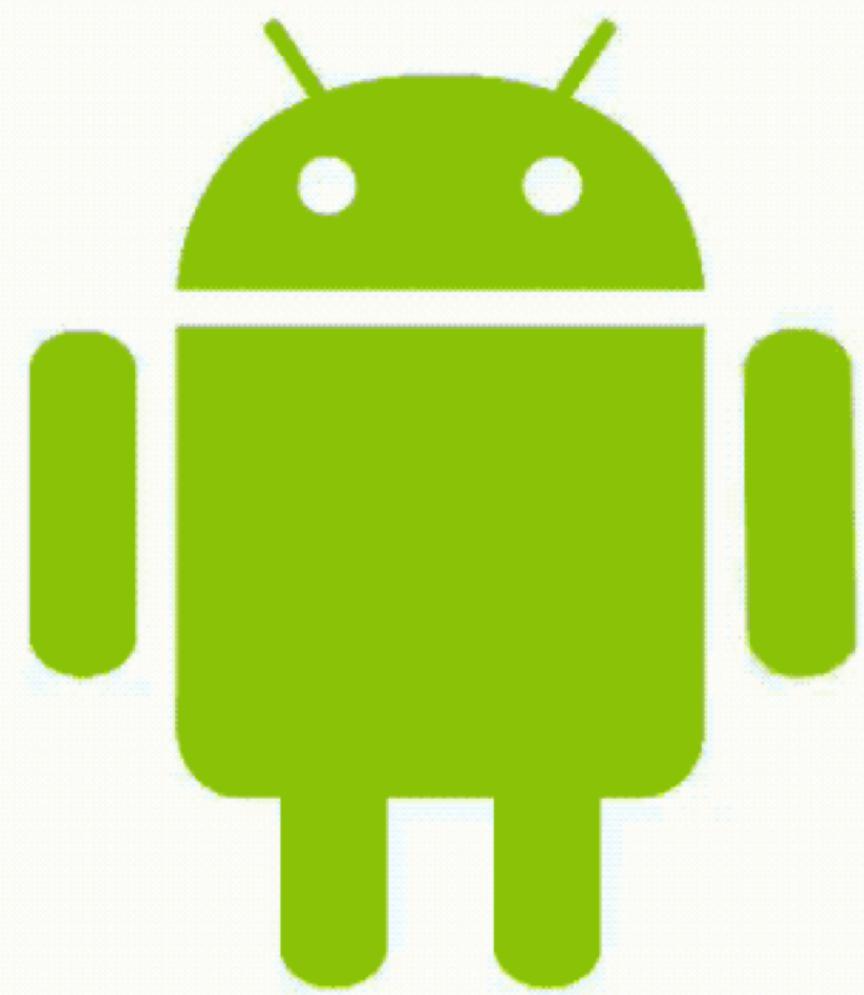
My self

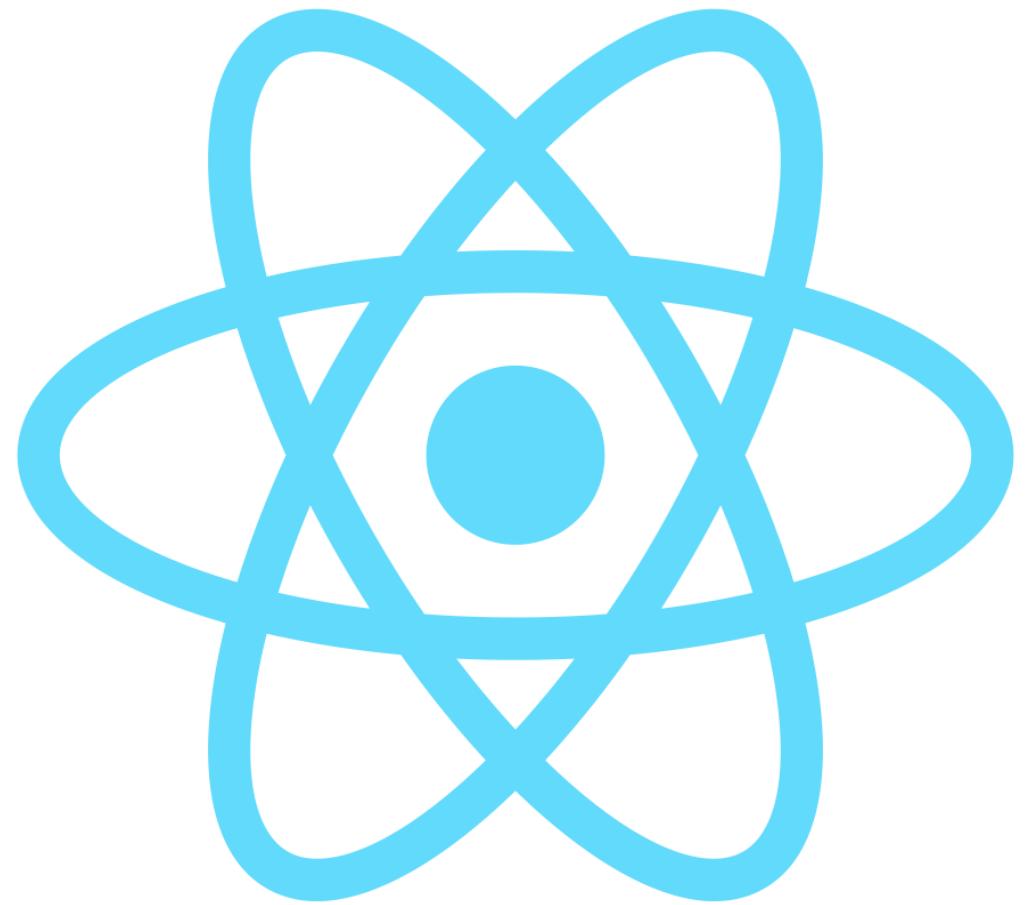
- Vladimir Ivanov
- EPAM Systems





Mobile Landscape





DevOps Practices

DevOps Practices

- Continuous integration

DevOps Practices

- Continuous integration
- Continuous delivery

DevOps Practices

- Continuous integration
- Continuous delivery
- Secured process

DevOps Practices

- Continuous integration
- Continuous delivery
- Secured process
- Frequent releases(Canary/BG deployments)

DevOps Practices

- Continuous integration
- Continuous delivery
- Secured process
- Frequent releases(Canary/BG deployments)
- Automated testing

DevOps Practices

- Continuous integration
- Continuous delivery
- Secured process
- Frequent releases(Canary/BG deployments)
- Automated testing
- Monitoring

DevOps Practices

- Continuous integration
- Continuous delivery
- Secured process
- Frequent releases(Canary/BG deployments)
- Automated testing
- Monitoring
- Automated rollback

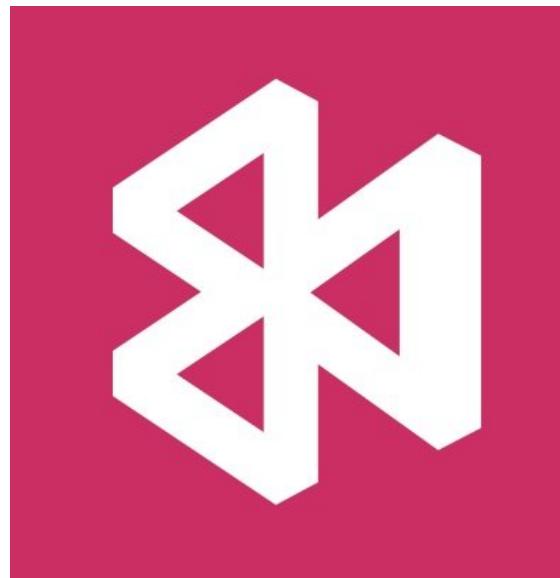
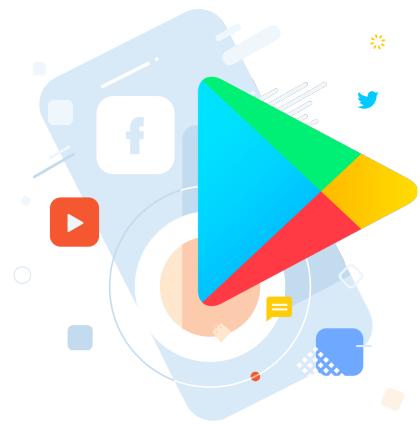
Alena



Continuous Integration

Deploy





Test Fairy



Rollouts

Rollouts

- Internal testing, Alpha and Beta channels in Google Play

Rollouts

- Internal testing, Alpha and Beta channels in Google Play
- Internal testing, Public Testing in TestFlight

Rollouts

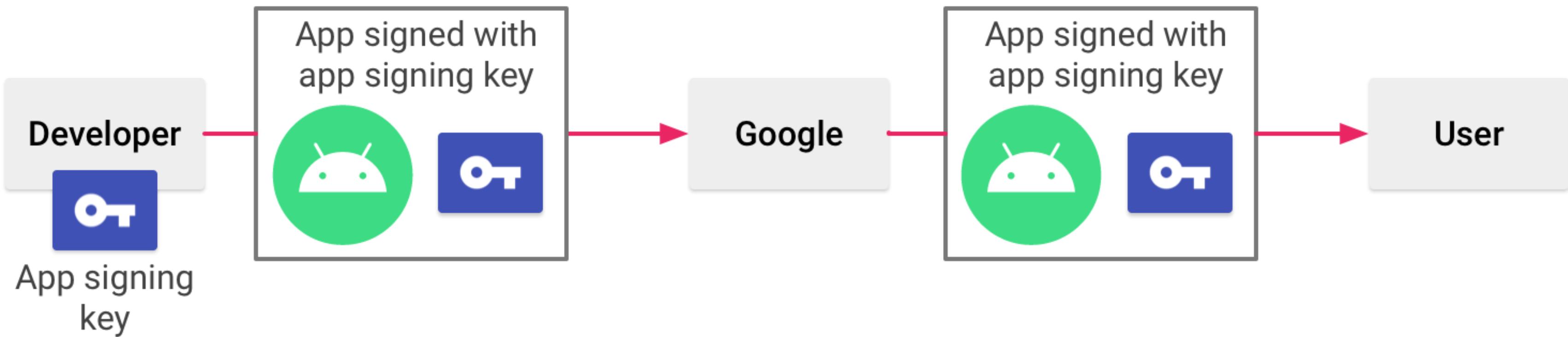
- Internal testing, Alpha and Beta channels in Google Play
- Internal testing, Public Testing in TestFlight
- Public releases support staged rollouts

sign

Secured process



android



Android Sign

Android Sign

- Keystore(Java Keystore)

Android Sign

- Keystore(Java Keystore)
- Store pass

Android Sign

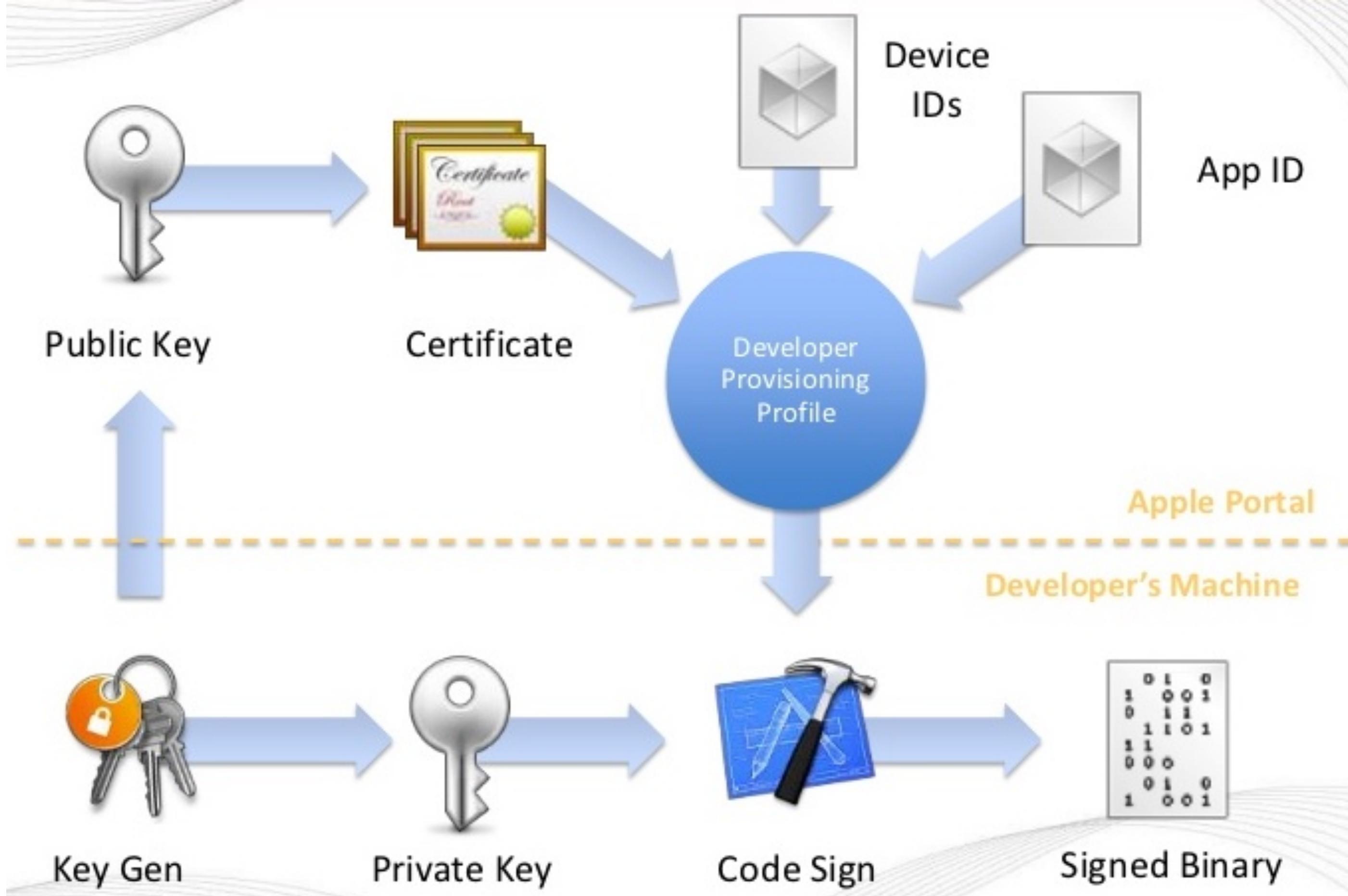
- Keystore(Java Keystore)
- Store pass
- Key alias

Android Sign

- Keystore(Java Keystore)
- Store pass
- Key alias
- Key pass



ios





DevOps Practices

- Continuous integration 
- Continuous delivery 
- Secured process 
- Frequent releases(Canary/BG deployments)
- Automated testing
- Monitoring
- Automated rollback

Review

GP used to take 3-4 hours

AppStore used to take 2 weeks

GP takes 3-4 ~~hours~~ days

AppStore used to take 2 ~~weeks~~ days

DevOps Practices

- Continuous integration 
- Continuous delivery 
- Secured process 
- Frequent releases(Canary/BG deployments) 
- Automated testing
- Monitoring
- Automated rollback

e2e tests



Issues e2e tests

Issues e2e tests

- Takes a lot of time

Issues e2e tests

- Takes a lot of time
- Requires devices

How to cope with devices?

How to cope with devices?

- Buy our own(expensive)

How to cope with devices?

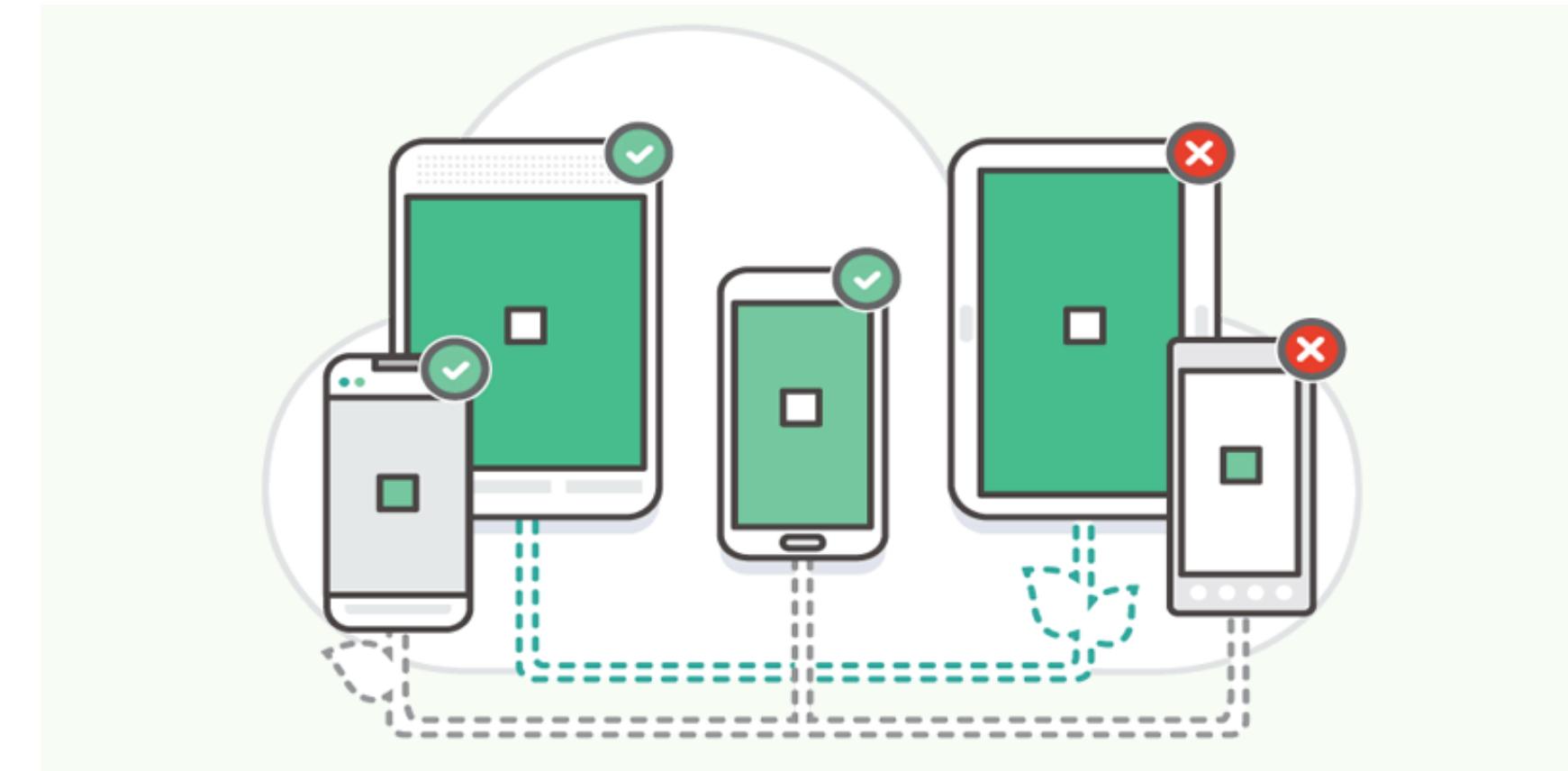
- Buy our own(expensive)
- Test on Emulators(bad quality)

How to cope with devices?

- Buy our own(expensive)
- Test on Emulators(bad quality)
- Device Farms

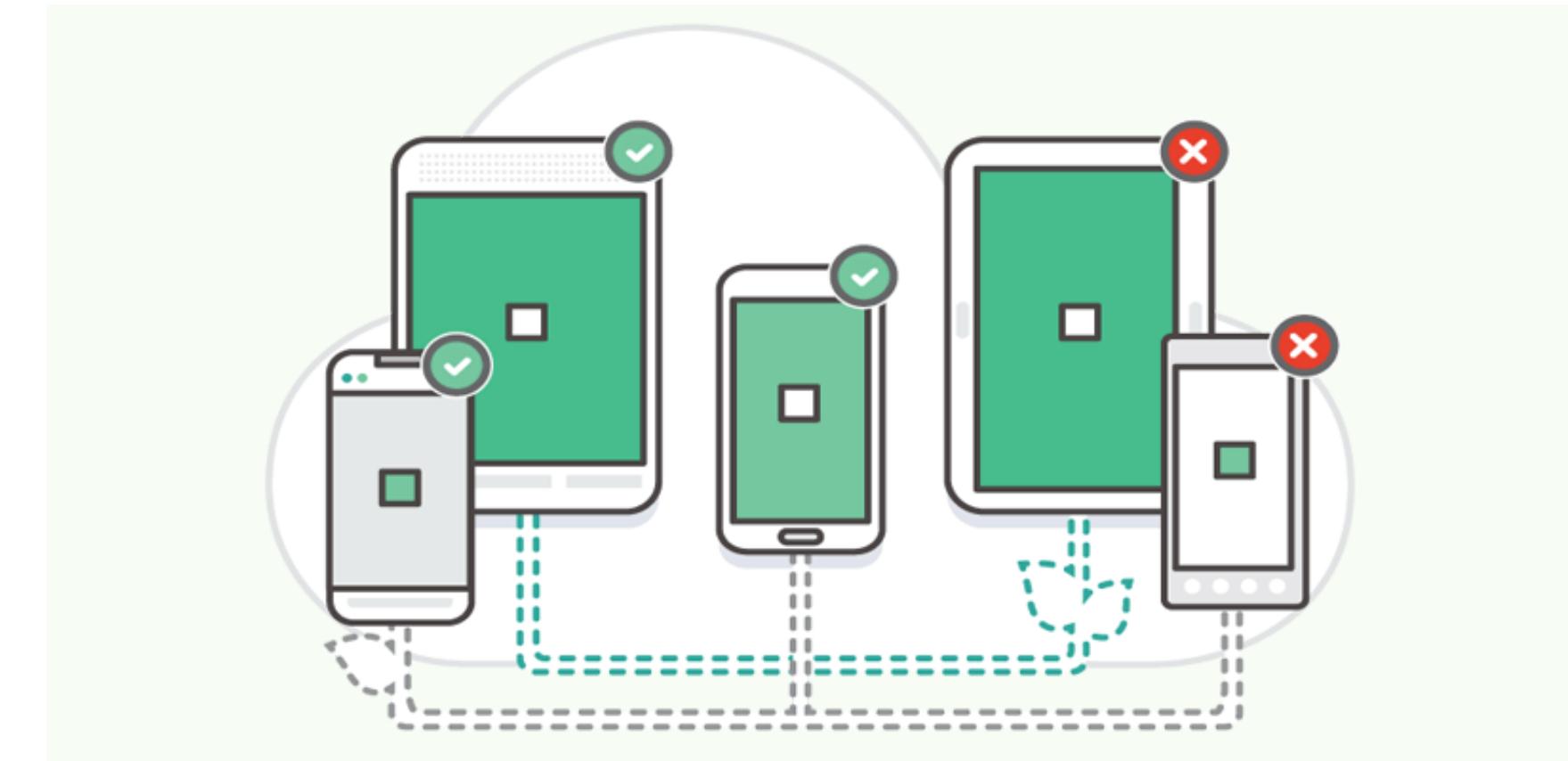


Device Farms



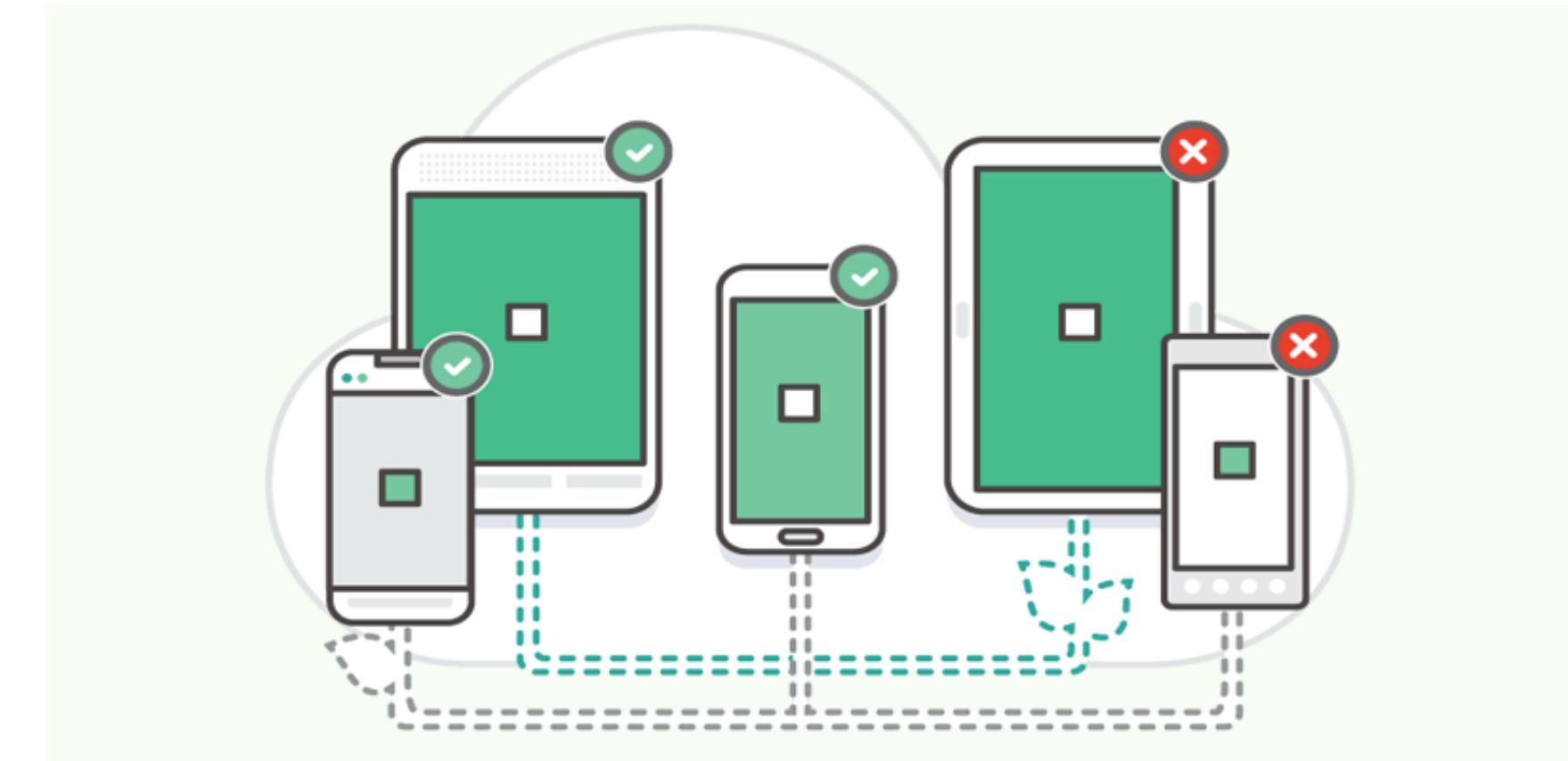
Device Farms

- AWS Device Farm



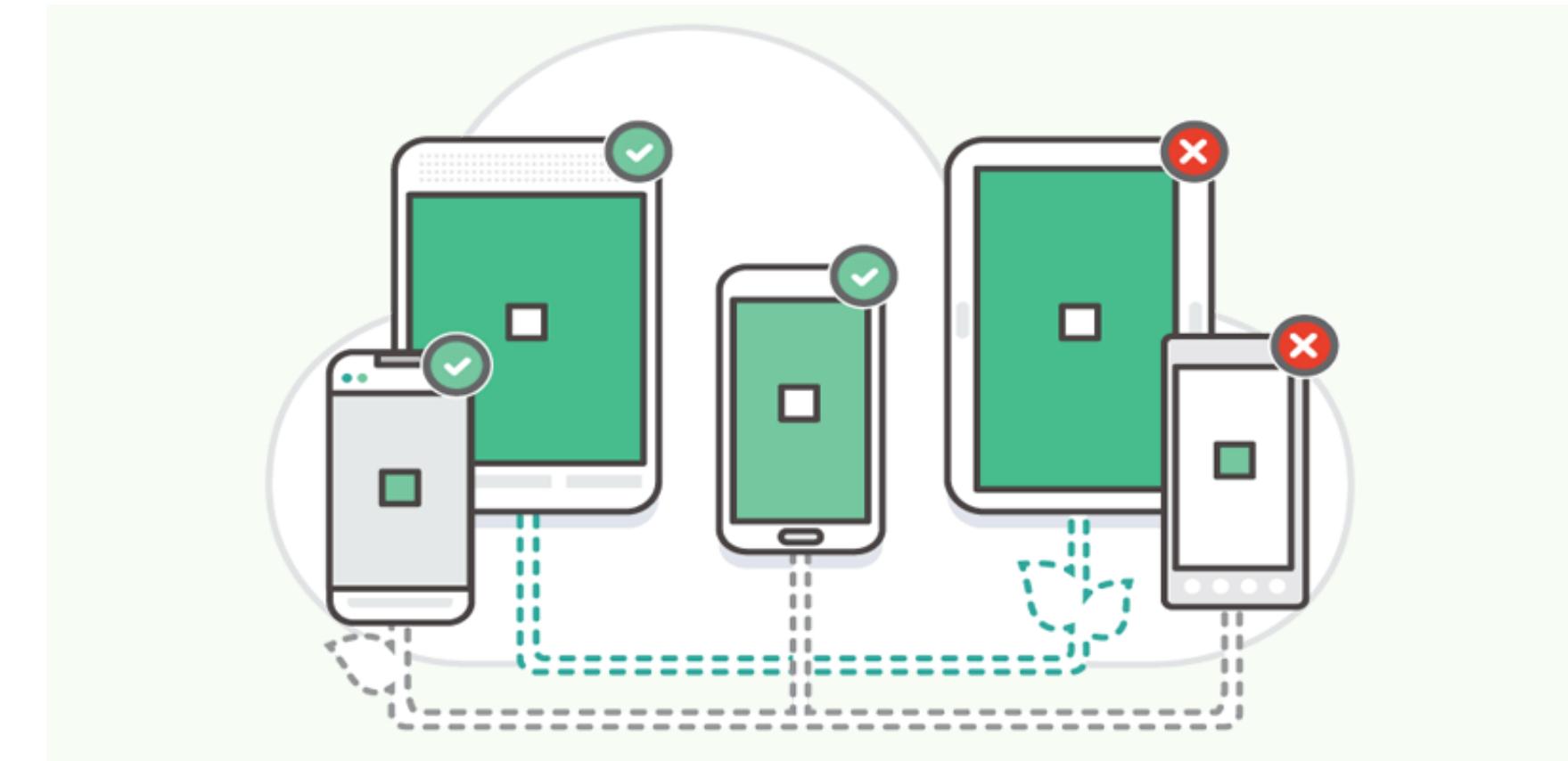
Device Farms

- AWS Device Farm
- Firebase device farm



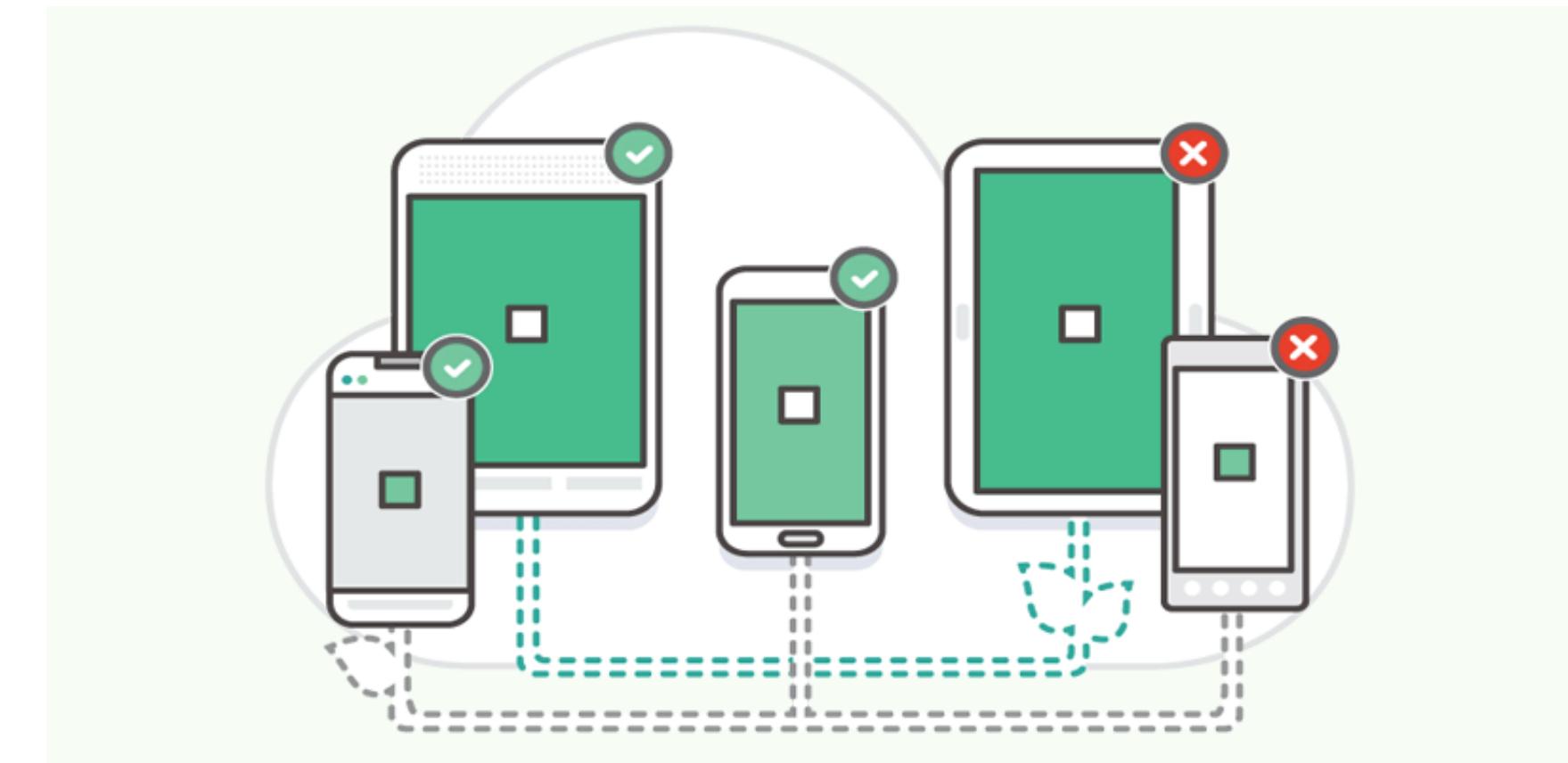
Device Farms

- AWS Device Farm
- Firebase device farm
- Azure Device Farm



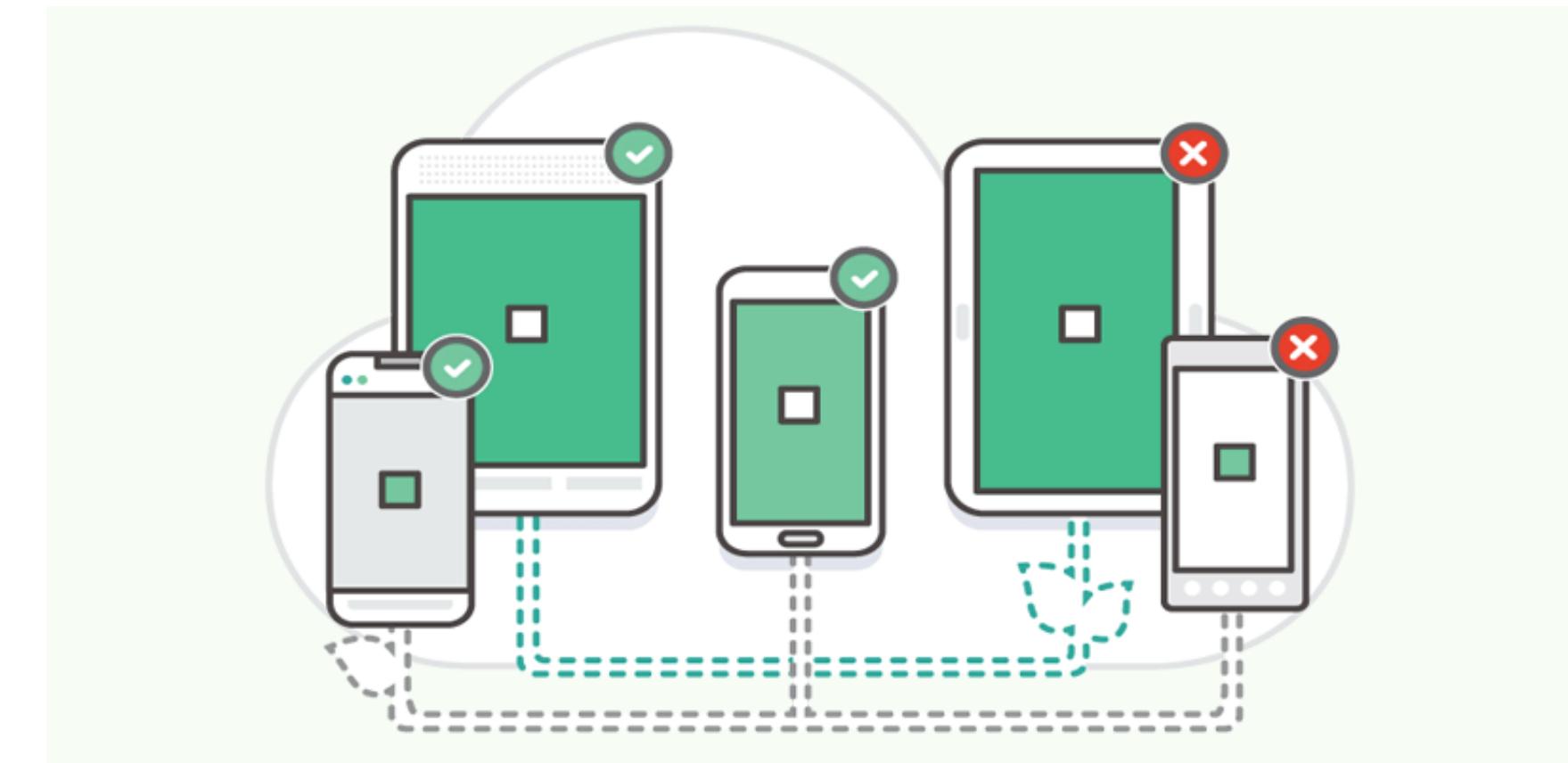
Device Farms

- AWS Device Farm
- Firebase device farm
- Azure Device Farm
- Others



Device Farms

- AWS Device Farm
- Firebase device farm
- Azure Device Farm
- Others
- Your own



Building your own

What would you need

What would you need

- Web Server

What would you need

- Web Server
- Logs Storage

What would you need

- Web Server
- Logs Storage
- Devices

What would you need

- Web Server
- Logs Storage
- Devices
- Media to connect to the devices(add, xcrun)

Easy part...

Easy part...

- List of devices

```
$ adb devices
```

```
$ xcrun simctl list
```

```
snash — [~] — git - TISN — 93x23
iPad 2 (com.apple.CoreSimulator.SimDeviceType.iPad-2)
iPad Retina (com.apple.CoreSimulator.SimDeviceType.iPad-Retina)
iPad Air (com.apple.CoreSimulator.SimDeviceType.iPad-Air)
iPad Air 2 (com.apple.CoreSimulator.SimDeviceType.iPad-Air-2)
iPad Pro (9.7-inch) (com.apple.CoreSimulator.SimDeviceType.iPad-Pro--9-7-inch-)
iPad Pro (12.9-inch) (com.apple.CoreSimulator.SimDeviceType.iPad-Pro)
Apple TV 1080p (com.apple.CoreSimulator.SimDeviceType.Apple-TV-1080p)
Apple Watch - 38mm (com.apple.CoreSimulator.SimDeviceType.Apple-Watch-38mm)
Apple Watch - 42mm (com.apple.CoreSimulator.SimDeviceType.Apple-Watch-42mm)
Apple Watch Series 2 - 38mm (com.apple.CoreSimulator.SimDeviceType.Apple-Watch-Series-2-38mm)
Apple Watch Series 2 - 42mm (com.apple.CoreSimulator.SimDeviceType.Apple-Watch-Series-2-42mm)
== Runtimes ==
iOS 10.3 (10.3 - 14E269) (com.apple.CoreSimulator.SimRuntime.iOS-10-3)
tvOS 10.2 (10.2 - 14W260) (com.apple.CoreSimulator.SimRuntime.tvOS-10-2)
watchOS 3.2 (3.2 - 14V243) (com.apple.CoreSimulator.SimRuntime.watchOS-3-2)
== Devices ==
-- iOS 10.3 --
    Shashi-iPhone-Simulator (C8C050F3-062D-4444-A888-BBEC783E31BD) (Booted)
-- tvOS 10.2 --
-- watchOS 3.2 --
== Device Pairs ==
shashi@Shashikants-MacBook-Pro:~
```

Easy part...

Easy part...

- List of devices

```
$ adb devices
```

```
$ xcrun simctl list
```

Easy part...

- List of devices

```
$ adb devices  
$ xcrun simctl list
```

- Installation and uploading files

```
$ adb push selfie.png /sdcard0/Downloads  
$ adb install -r myCoolApp.apk  
$ xcrun altool --upload-app --type ios --file "path/  
to/application.ipa" --username "YOUR_ITMC_USER" --  
password "YOUR_ITMC_PASSWORD"
```

Complex thing

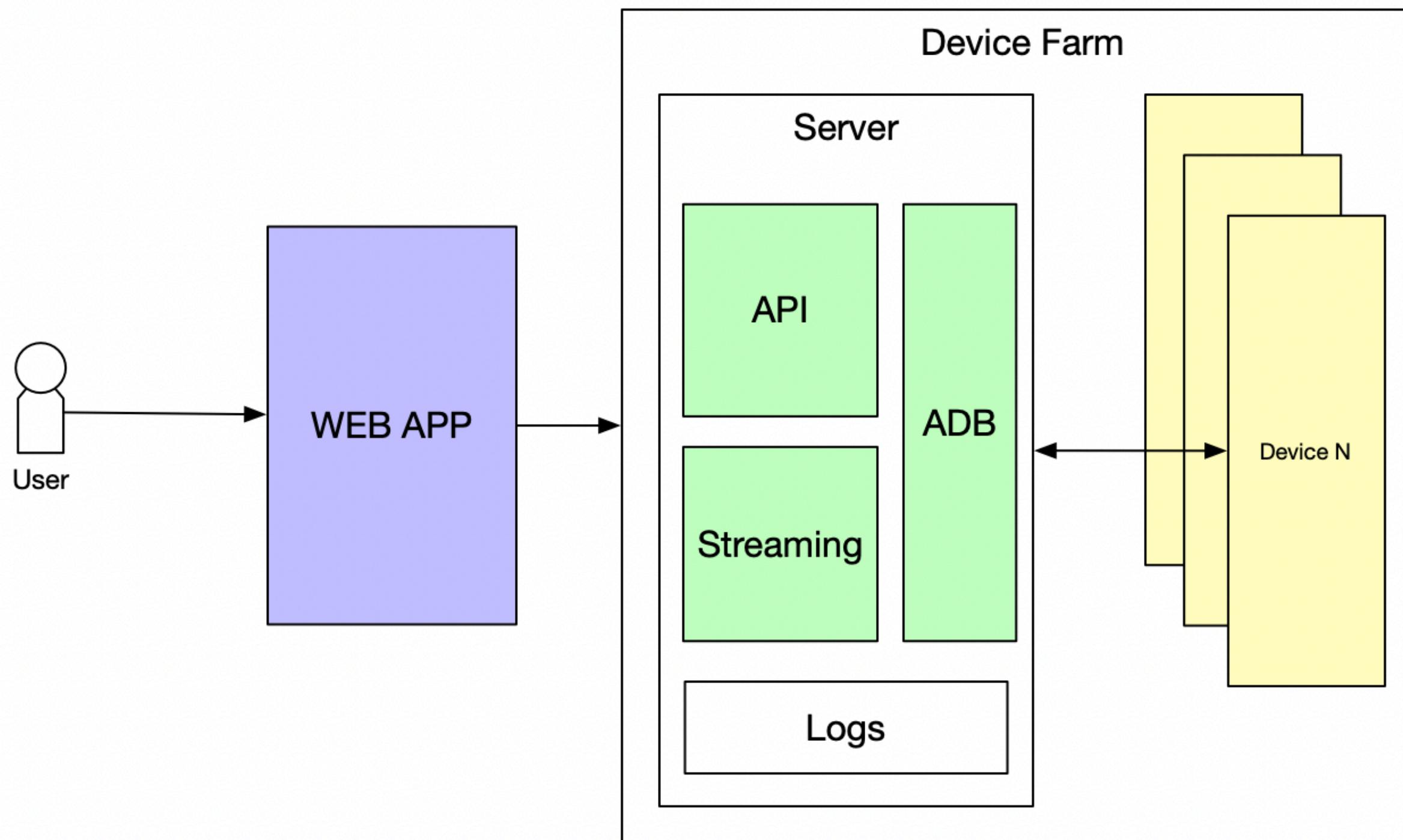
Complex thing

- Streaming

Android: adb, scrcpy² tool

iOS: xcrun

² <https://blog.rom1v.com/2018/03/introducing-scrcpy/>



In EPAM

mobilefarm iPhone 6s Android Devices iOS Devices Metrics

10:38 PM 100% 10:38 PM 100%

IRON DESERT Calculator Messages Utilities

Photos Camera Weather Maps

Videos Wallet Notes Clock

Reminders Stocks iTunes Store App Store

Home iBooks News Settings

Health

Phone Mail Safari Music

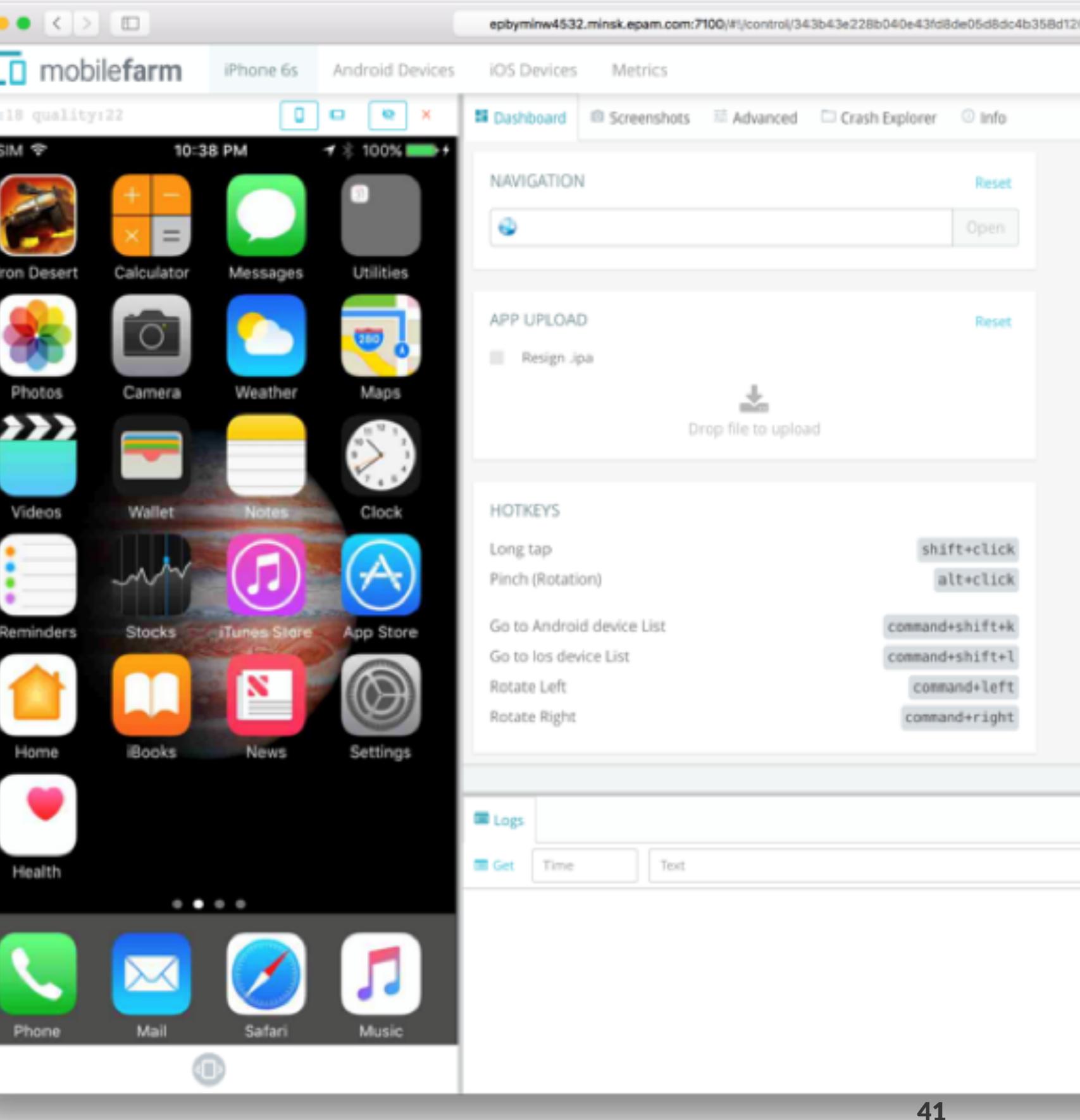
NAVIGATION Open Reset

APP UPLOAD Resign .ipa Drop file to upload

HOTKEYS

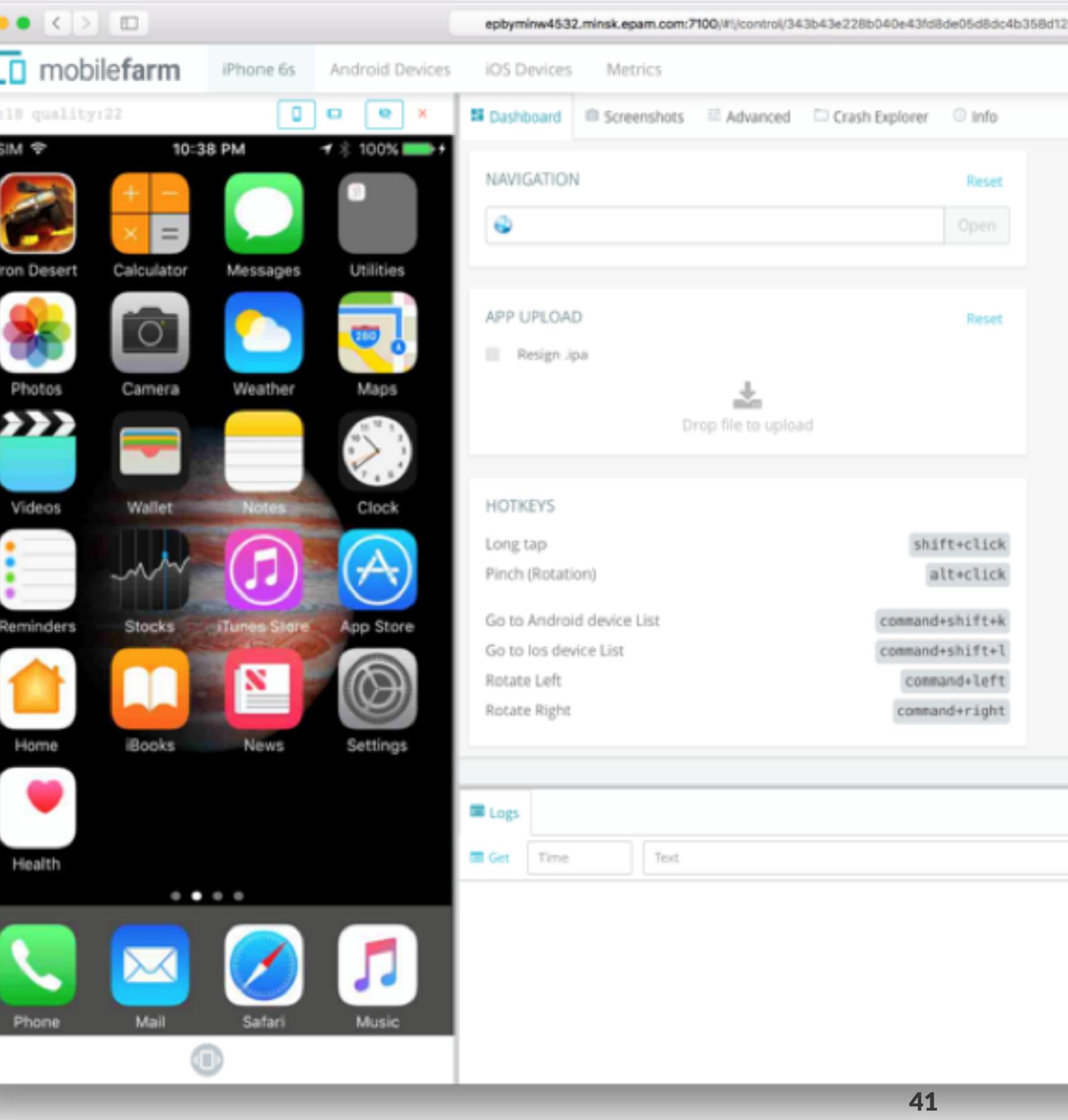
- Long tap shift+click
- Pinch (Rotation) alt+click
- Go to Android device List command+shift+k
- Go to iOS device List command+shift+l
- Rotate Left command+left
- Rotate Right command+right

Logs Get Time Text



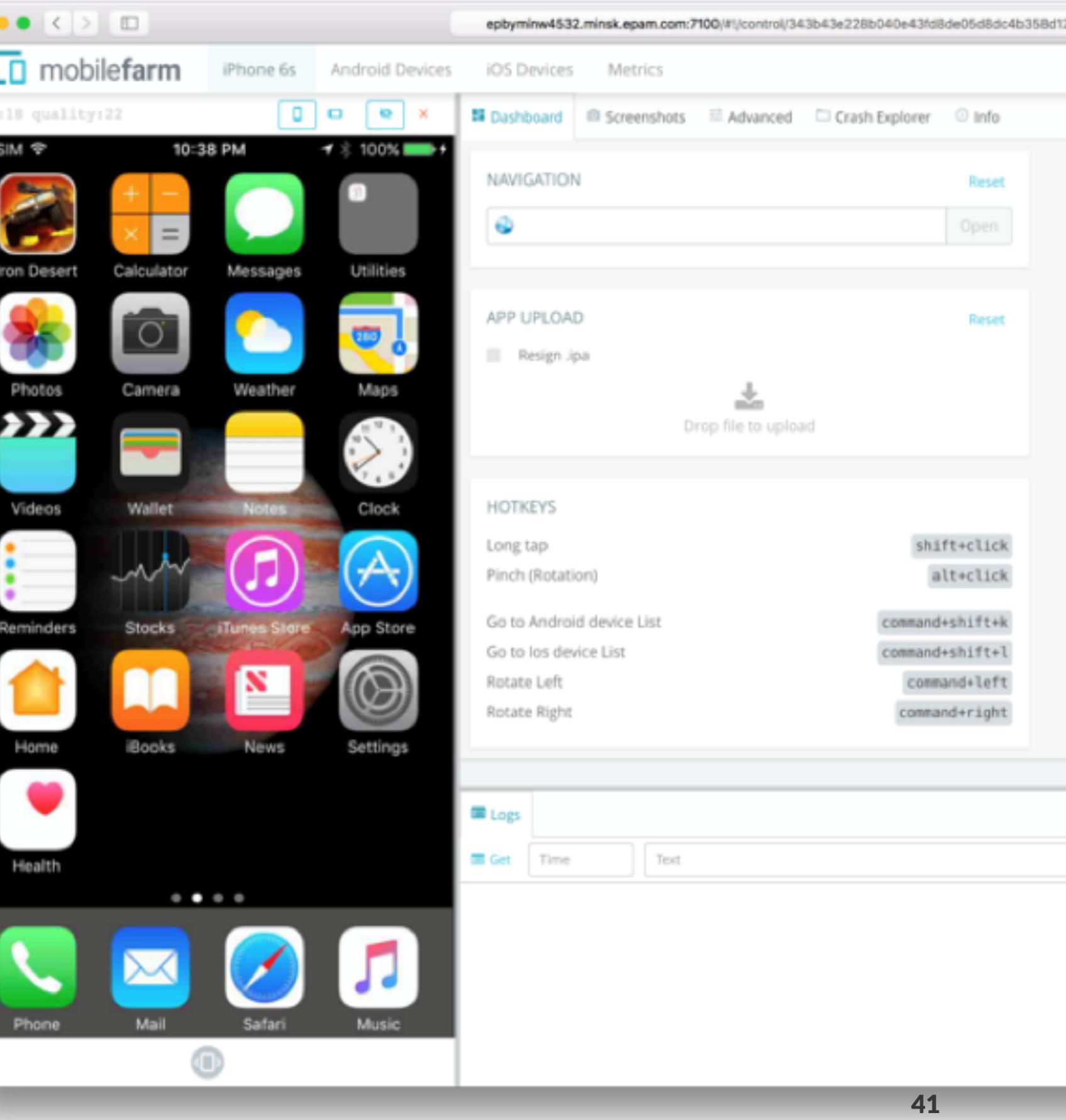
In EPAM

- Mobile farm for Mobile CC



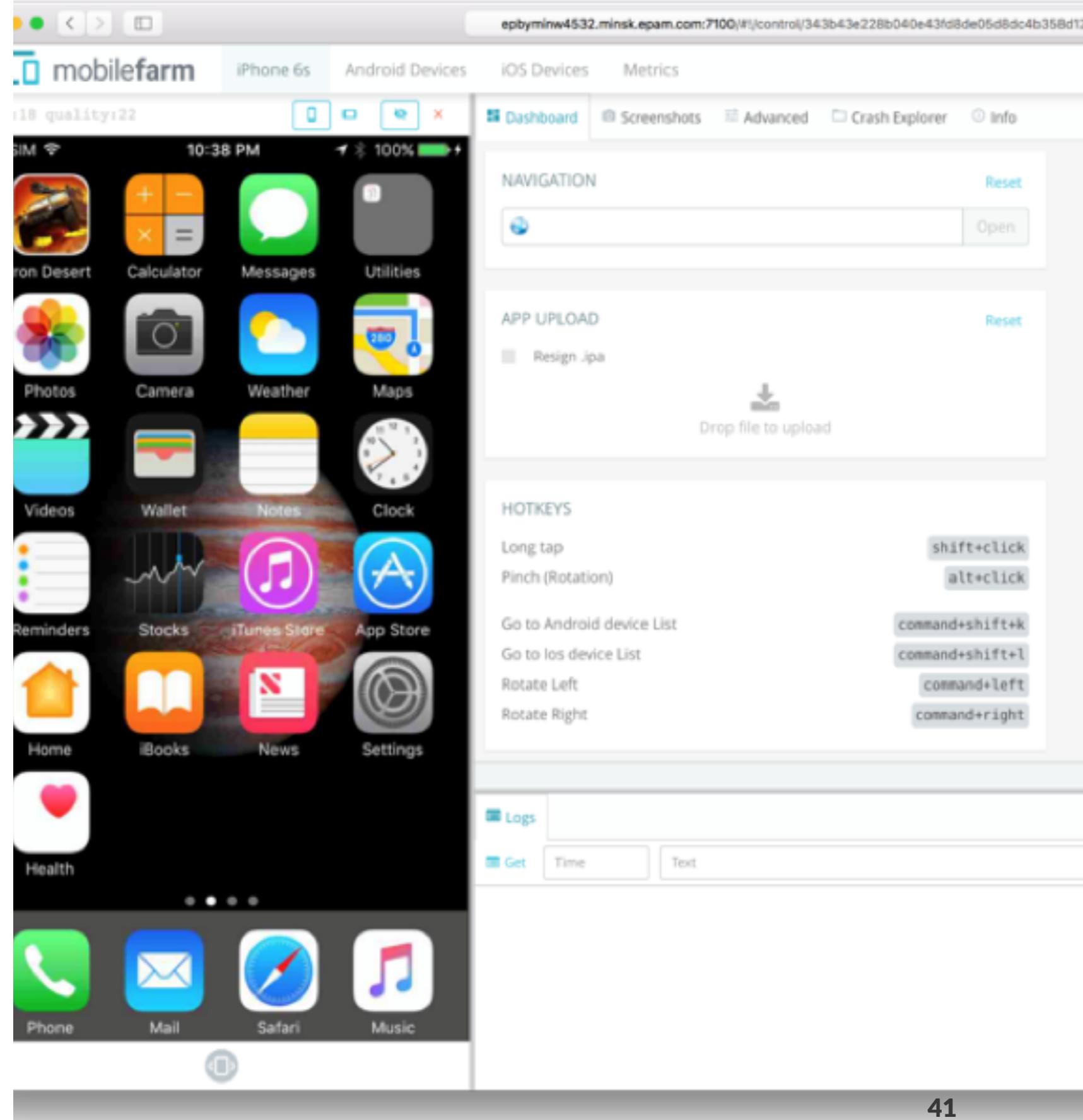
In EPAM

- Mobile farm for Mobile CC
- Free for the projects at the moment



In EPAM

- Mobile farm for Mobile CC
- Free for the projects at the moment
- Plans for commercial offerings



DevOps Practices

- Continuous integration 
- Continuous delivery 
- Secured process 
- Frequent releases(Canary/BG deployments) 
- Automated testing 
- Monitoring
- Automated rollback

Observability starts with simple questions:

- How would you know if your system was unhealthy?
- How would you know if your users were having a bad experience?
- In the event there was a problem how would you isolate the cause?

@robmea

Observability

Observability

- Mobile apps run on the user devices and deployed through stores which make them unmanageable

Observability

- Mobile apps run on the user devices and deployed through stores which make them unmanageable
- However observability is still in place

Crash Reporting and Analytics

Crash Reporting and Analytics

- Your own(with help of ACRA)

Crash Reporting and Analytics

- Your own(with help of ACRA)
- Firebase Analytics

Crash Reporting and Analytics

- Your own (with help of ACRA)
- Firebase Analytics
- AppCenter

Crash Reporting and Analytics

- Your own (with help of ACRA)
- Firebase Analytics
- AppCenter
- Sentry

Crash Reporting and Analytics

- Your own(with help of ACRA)
- Firebase Analytics
- AppCenter
- Sentry
- Bugsnag

What can be gathered

What can be gathered

- Free device memory

What can be gathered

- Free device memory
- Free process memory

What can be gathered

- Free device memory
- Free process memory
- Device, OS Version

What can be gathered

- Free device memory
- Free process memory
- Device, OS Version
- Active users

What can be gathered

- Free device memory
- Free process memory
- Device, OS Version
- Active users
- Session lengths

What can be gathered

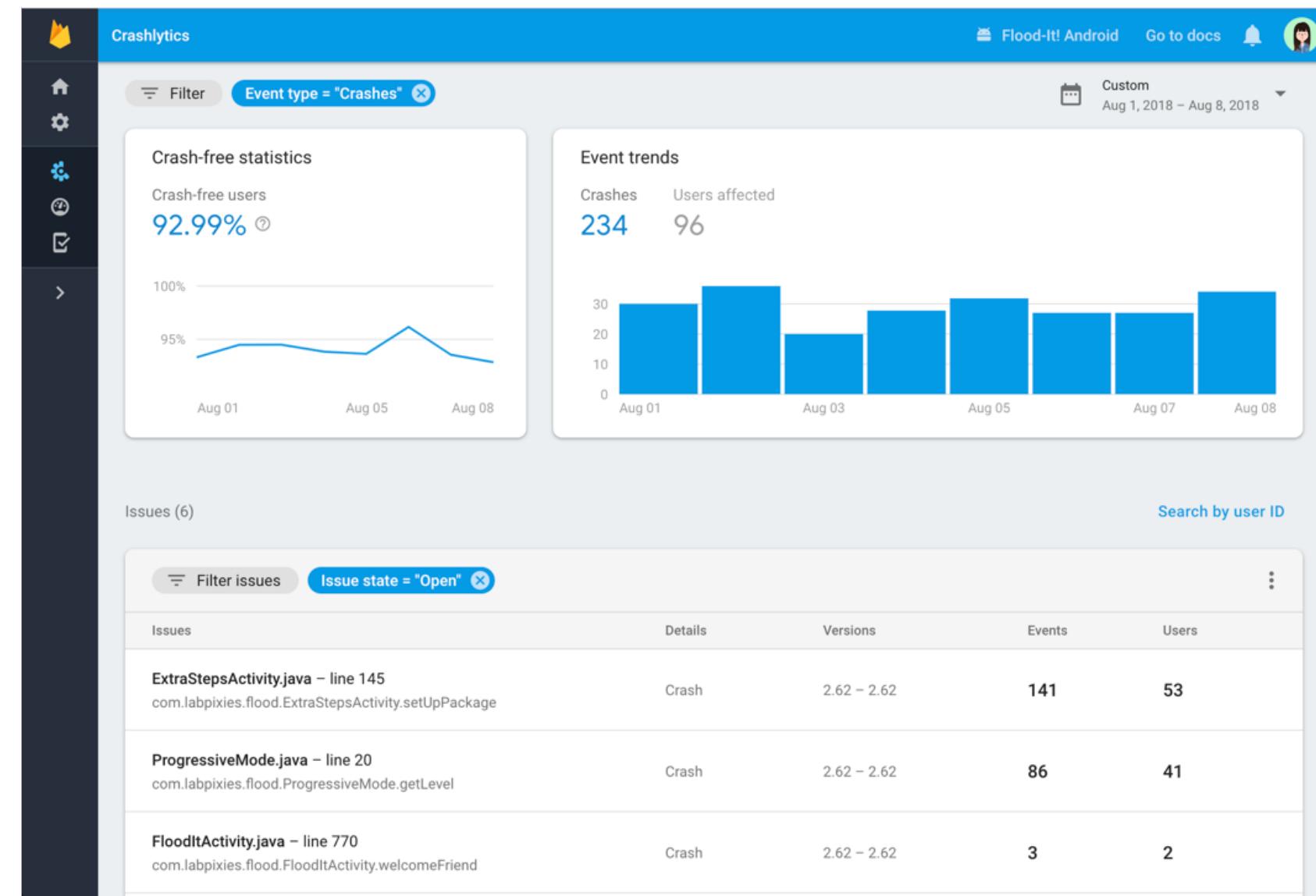
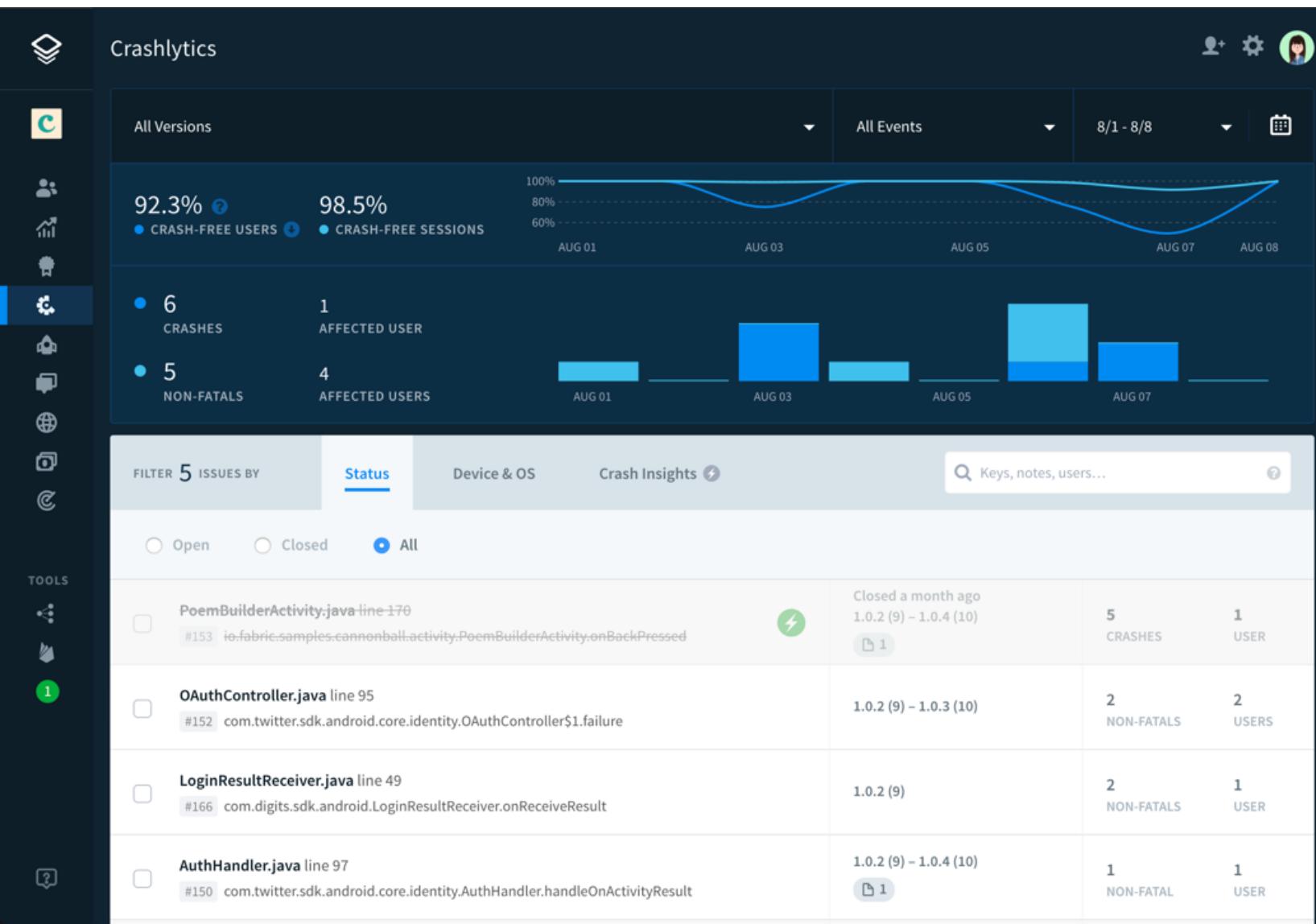
- Free device memory
- Free process memory
- Device, OS Version
- Active users
- Session lengths
- Thread dump

What can be gathered

- Free device memory
- Free process memory
- Device, OS Version
- Active users
- Session lengths
- Thread dump
- Stacktrace

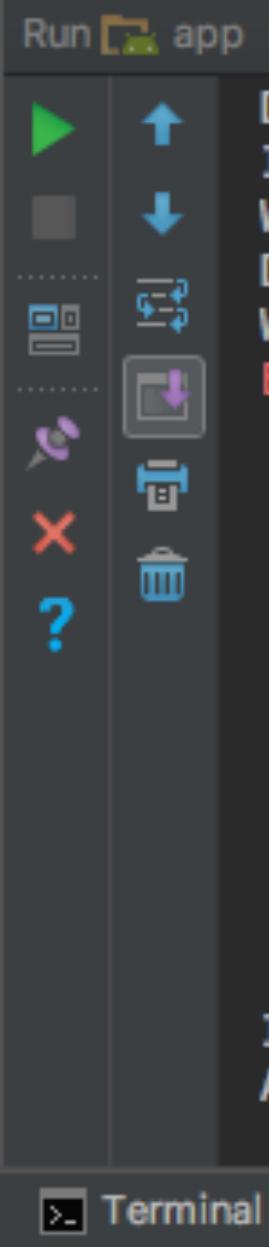
What can be gathered

- Free device memory
- Free process memory
- Device, OS Version
- Active users
- Session lengths
- Thread dump
- Stacktrace
- Many more



Martin

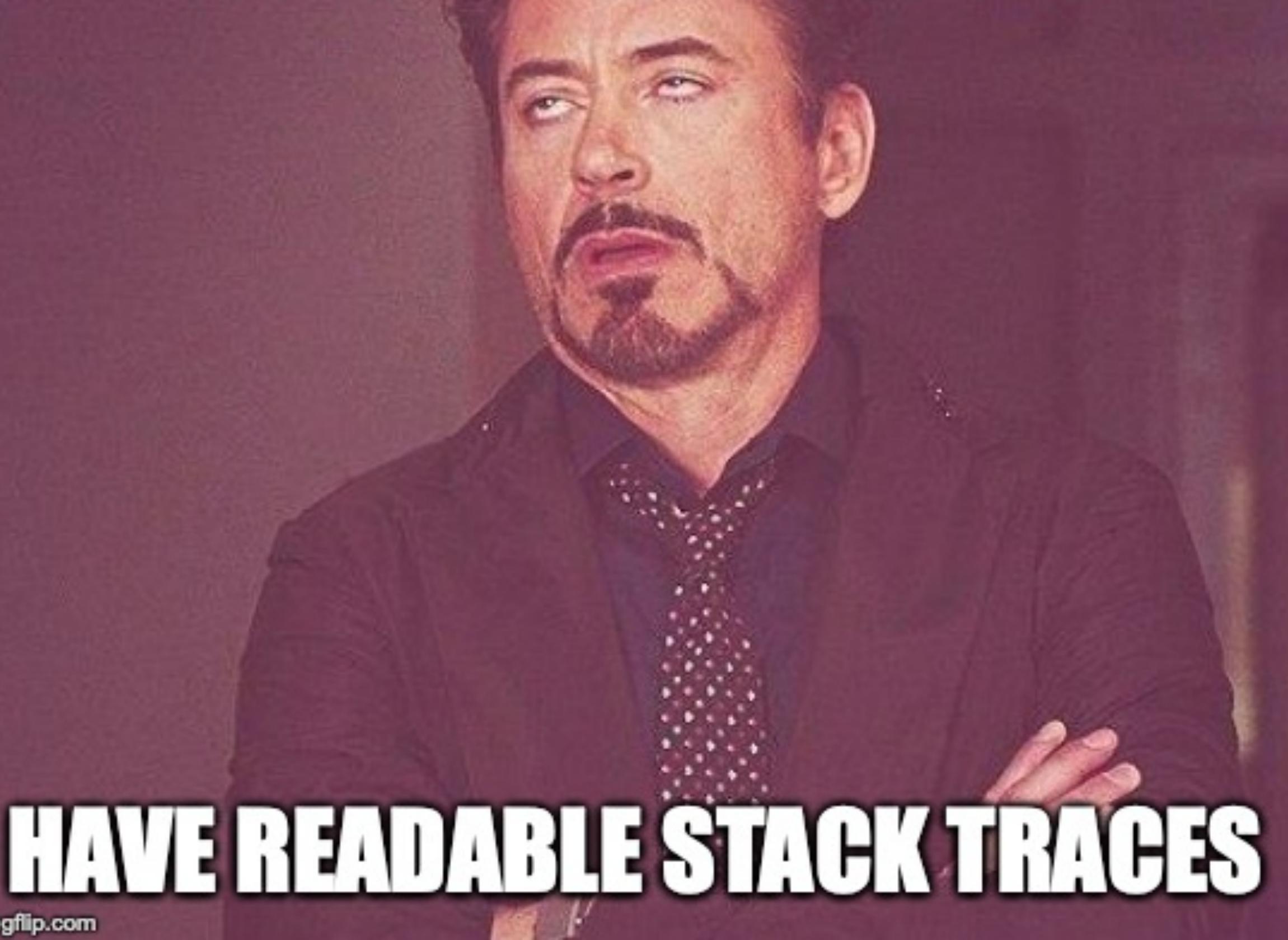




Terminal Build 6: Logcat 4: Run TODO

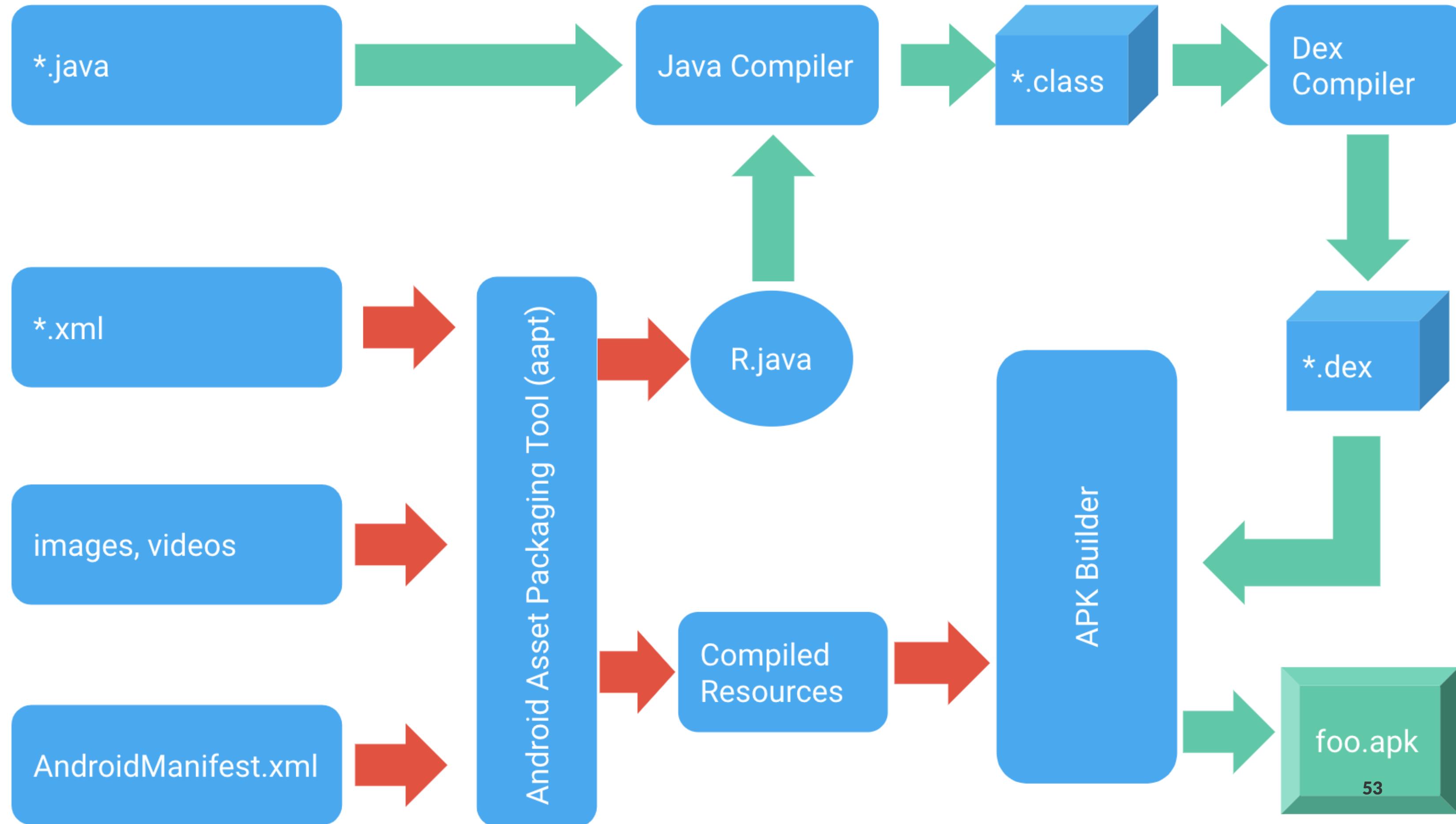
Compilation/Obfuscation, my ass

WHEN I HEAR I CAN'T



HAVE READABLE STACK TRACES









Home

CUSTOMIZE



Pins appear here



App Engine



Compute Engine



Kubernetes Engine



Cloud Functions



Cloud Run

STORAGE

Bigtable

API APIs

Requests (requests/sec)

⚠ No data is available for the selected time frame.

5:30

6 PM

1.0

0.8

0.6

0.4

0.2

0

[Go to APIs overview](#)

Google Cloud Platform status

All services normal

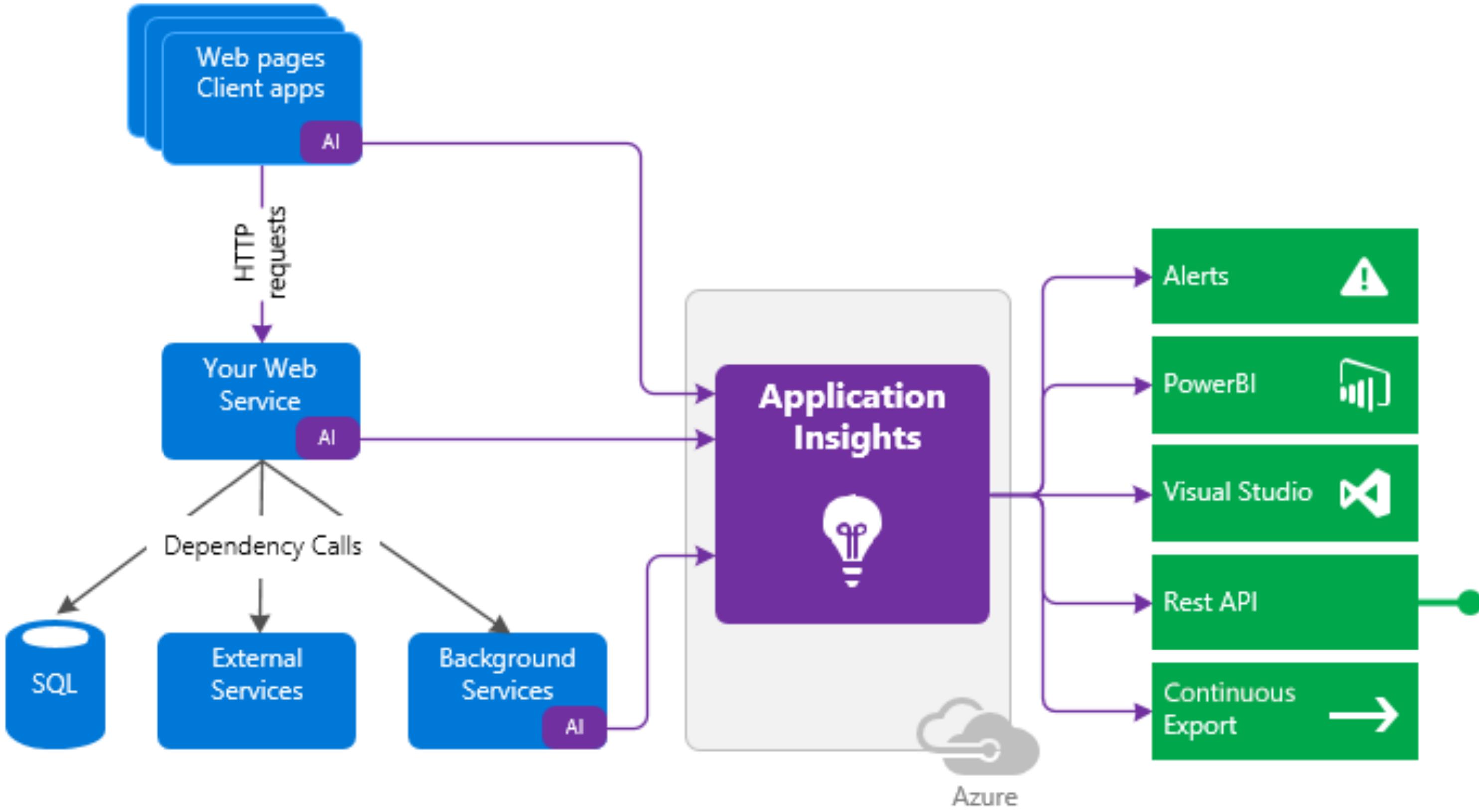
[Go to Cloud status dashboard](#)

Error Reporting

No sign of any errors. Have you set up Error Reporting?

[Learn how to set up Error Reporting](#)

AppCenter



DevOps Practices

- Continuous integration 
- Continuous delivery 
- Secured process 
- Frequent releases(Canary/BG deployments) 
- Automated testing 
- Monitoring 
- Automated rollback

Unfortunately

No rollbacks of unsuccessful releases

You need to build a new version and deploy it
instead



Possible solutions

Possible solutions

- Feature flags and server configs

Possible solutions

- Feature flags and server configs
- Usage of React-Native or other js based technologies

DevOps Practices

- Continuous integration 
- Continuous delivery 
- Secured process 
- Frequent releases(Canary/BG deployments) 
- Automated testing 
- Monitoring 
- Automated rollback 

Caveats

Hardware

Hardware

- Android can be built anywhere

Hardware

- Android can be built anywhere
- iOS - only on Mac machines



Long buildtimes

How to fight long build times

How to fight long build times

- Identifying slow code parts

How to fight long build times

- Identifying slow code parts
- Caching dependencies

How to fight long build times

- Identifying slow code parts
- Caching dependencies
- Modularization

How to fight long build times

- Identifying slow code parts
- Caching dependencies
- Modularization
- Build systems hacks

How to fight long build times

- Identifying slow code parts
- Caching dependencies
- Modularization
- Build systems hacks
- And more¹⁰

¹⁰<https://medium.com/@joshgare/8-tips-to-speed-up-your-swift-build-and-compile-times-in-xcode-73081e1d84ba>

Caching

iOS

iOS

- CocoaPods

iOS

- CocoaPods
- Carthage

iOS

- CocoaPods
- Carthage
- SwiftPM

iOS

iOS

- CocoaPods - source code based, own build spec

iOS

- CocoaPods - source code based, own build spec
- Carthage - dynamic frameworks

iOS

- CocoaPods - source code based, own build spec
- Carthage - dynamic frameworks
- SwiftPM - built by Apple

iOS

- CocoaPods - source code based, own build spec
- Carthage - dynamic frameworks
- SwiftPM - built by Apple
 - no resources support

iOS

- CocoaPods - source code based, own build spec
- Carthage - dynamic frameworks
- SwiftPM - built by Apple
 - no resources support
 - single language per package

iOS

- CocoaPods - source code based, own build spec
- Carthage - dynamic frameworks
- SwiftPM - built by Apple
 - no resources support
 - single language per package
 - problems having binary deps

Android

Android

- Gradle

Android

- Gradle
- Bazel/BUCK for companies with BIG apps

Caching Gradle dependencies might decrease build times. Give it a try if you haven't already.

Setup

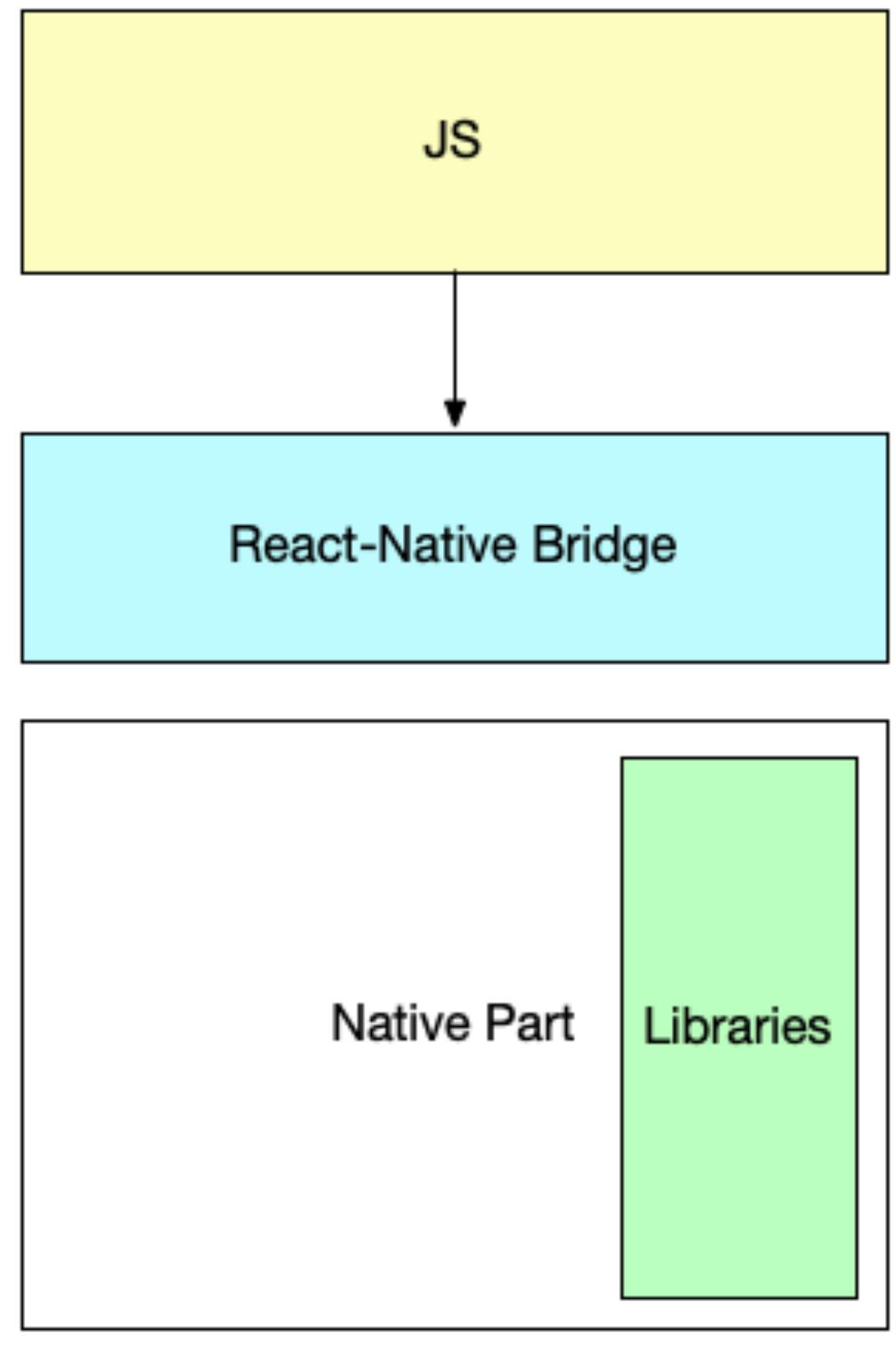
1. Put the Cache:Pull step after the Git clone step
2. Put the Cache:Push step to the very end of the Workflow

If you use v1.0 or newer of the Cache steps and the latest Gradle Runner step then that's all you have to do.

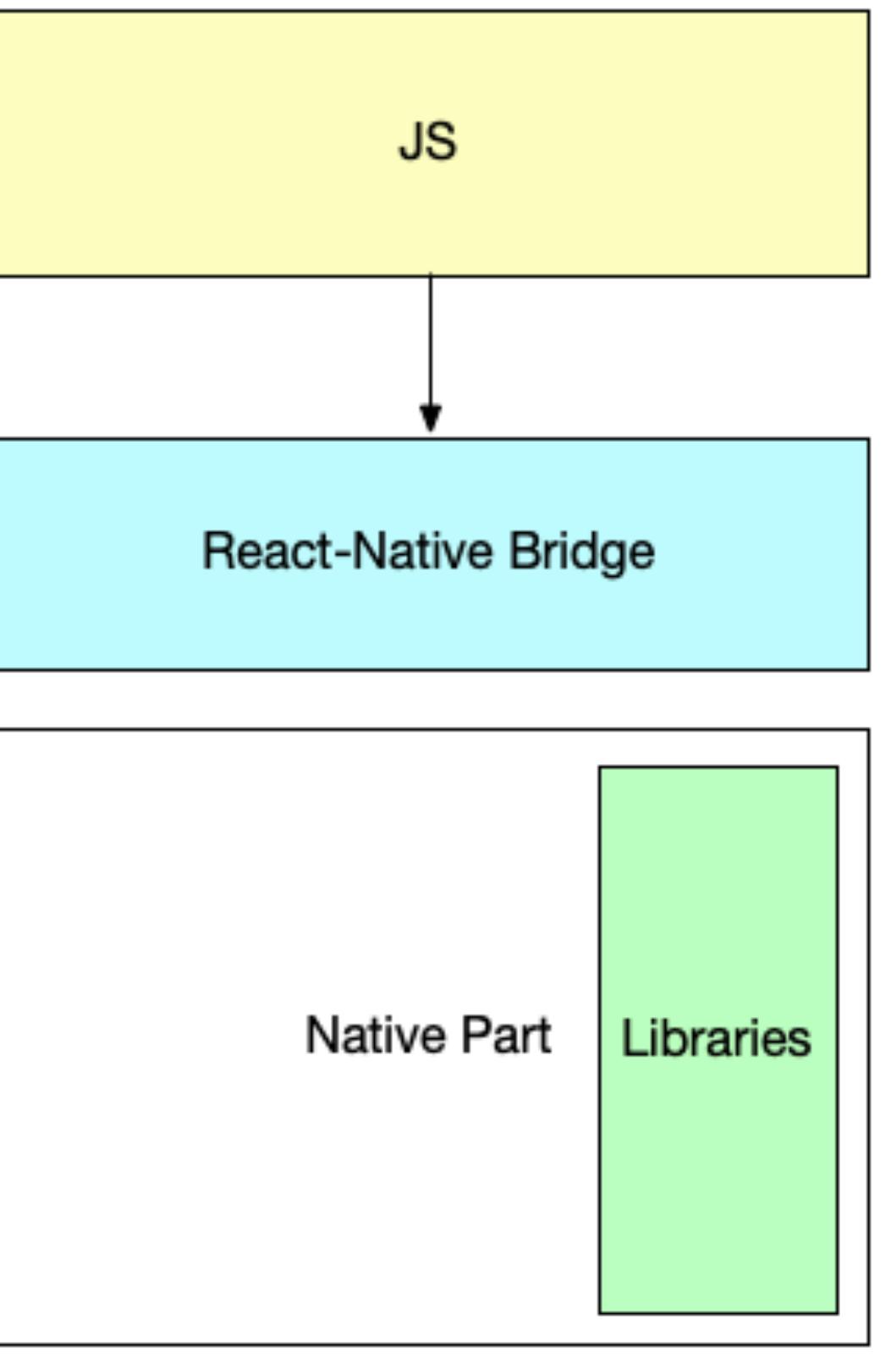
React-Native doesn't benefit from caching

But does benefit from building two apps
within a single run

React Native Android Application



React Native iOS Application



- Checkout the code

- Checkout the code
- Install the dependencies

- Checkout the code
- Install the dependencies
- Compile Android

- Checkout the code
- Install the dependencies
- Compile Android
- Compile iOS

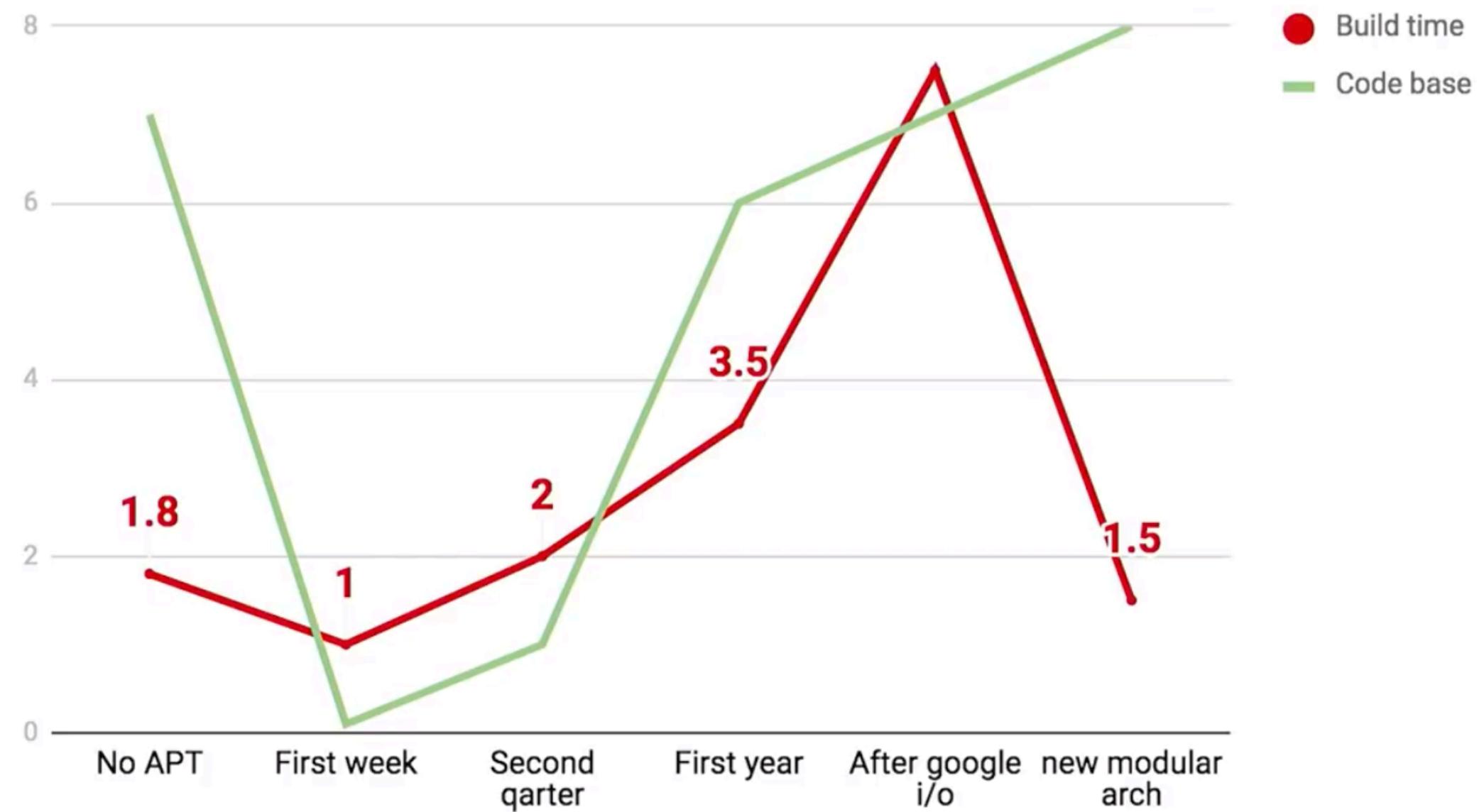
- Checkout the code
- Install the dependencies
- Compile Android
- Compile iOS
- Sign both

- Checkout the code
- Install the dependencies
- Compile Android
- Compile iOS
- Sign both
- Deploy both

- Checkout the code
- Install the dependencies
- ~~Compile Android~~
- ~~Compile iOS~~
- ~~Sign both~~
- ~~Deploy both~~

Modularization

Clean build time minutes



How to start

```
./gradlew clean assembleDebug --scan
```

Gradle

Gradle

- Takes 1 minute to configure 1300 modules

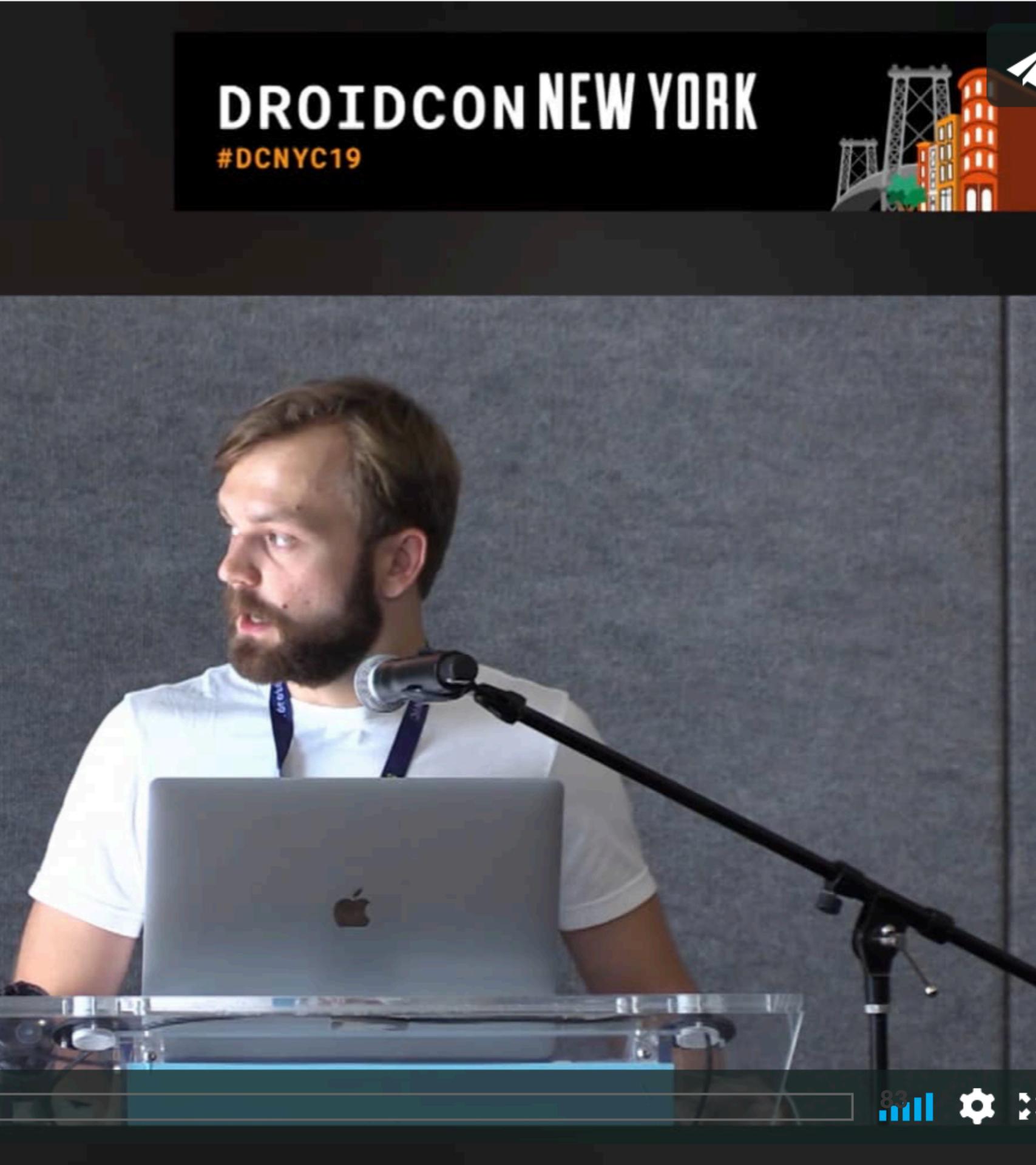
Gradle

- Takes 1 minute to configure 1300 modules
- Due to IO and unlimited immutability

Bazel

See the talk of Artem Zinnatullin¹

¹<https://www.droidcon.com/media-detail?video=362742329>



Your CI/CD pipeline should support it all!

Approaches

Approaches

Approaches

- Install Jenkins/TeamCity/Whatever else

Approaches

- Install Jenkins/TeamCity/Whatever else
- Spawn a Jenkins VM/Container in a Cloud

Approaches

- Install Jenkins/TeamCity/Whatever else
- Spawn a Jenkins VM/Container in a Cloud
- Use SaaS

Local Jenkins



Local Jenkins

- Free



Local Jenkins

- Free
- Great flexibility



Local Jenkins

- Free
- Great flexibility
- Agents system



Local Jenkins

- Free
- Great flexibility
- Agents system
- Decent plugin system



Local Jenkins

- Free
- Great flexibility
- Agents system
- Decent plugin system
- For iOS: just install an agent on a Mac



Local Jenkins

- Free
- Great flexibility
- Agents system
- Decent plugin system
- For iOS: just install an agent on a Mac
- But no containerization out-of-the-box



Local Jenkins

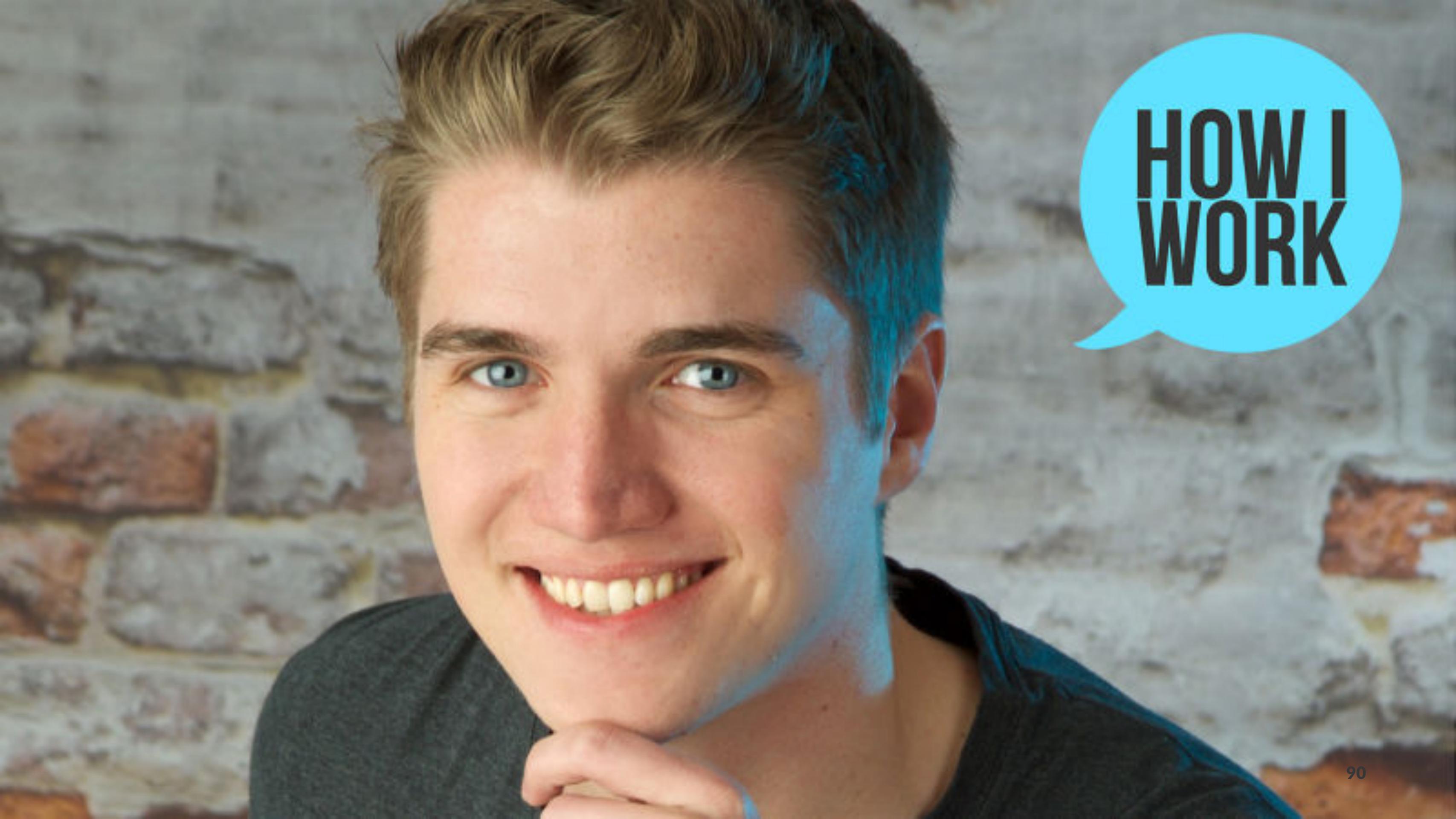
- Free
- Great flexibility
- Agents system
- Decent plugin system
- For iOS: just install an agent on a Mac
- But no containerization out-of-the-box
- All support burden is on your shoulders



It should be managed

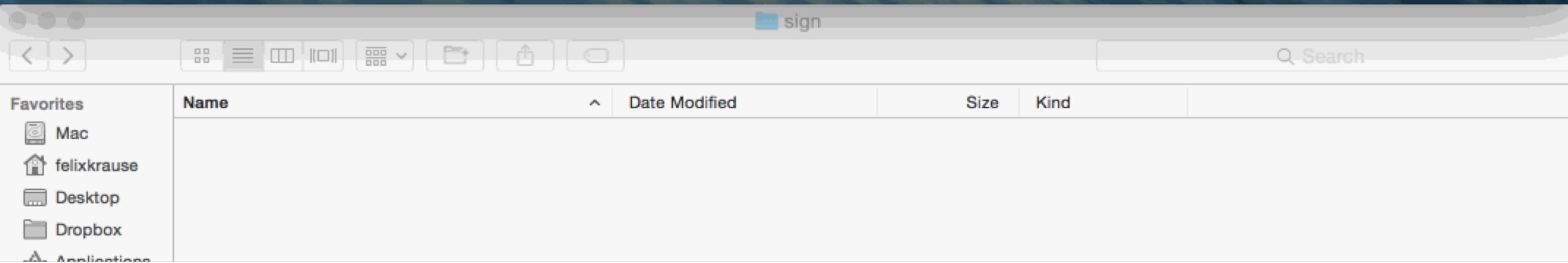


fastlane



**HOW I
WORK**

Felixs-MacBook-Pro-Retina:sign felixkrause\$



Pros

- ✨ More than 400 integrations
- 📖 100% open source under the MIT license
- 🛠️ Runs on your machine, it's your app and your data
- 💻 Supports iOS, Mac, and Android apps
- 🔧 Extendable

When



When

- For education purposes and student projects



When

- For education purposes and student projects
- For companies which already use Jenkins



When

- For education purposes and student projects
- For companies which already use Jenkins
- For security paranoids(debatable)



Jenkins in Cloud

Jenkins in Cloud

- Solves the containerization issue

Jenkins in Cloud

- Solves the containerization issue
- Support burden is partially decreased

Jenkins in Cloud

- Solves the containerization issue
- Support burden is partially decreased
- Unclear how to solve Mac issue

3rd party services

3rd party services

3rd party services

- GitLab CI

3rd party services

- GitLab CI
- Circle CI

3rd party services

- GitLab CI
- Circle CI
- Nevercode

3rd party services

- GitLab CI
- Circle CI
- Nevercode
- App Center

3rd party services

- GitLab CI
- Circle CI
- Nevercode
- App Center
- Bitrise

GitLab CI/Circle CI

GitLab CI/Circle CI

- Very basic support(mac machines, xcode, gradle)

GitLab CI/Circle CI

- Very basic support(mac machines, xcode, gradle)
- Integration with the parent tool

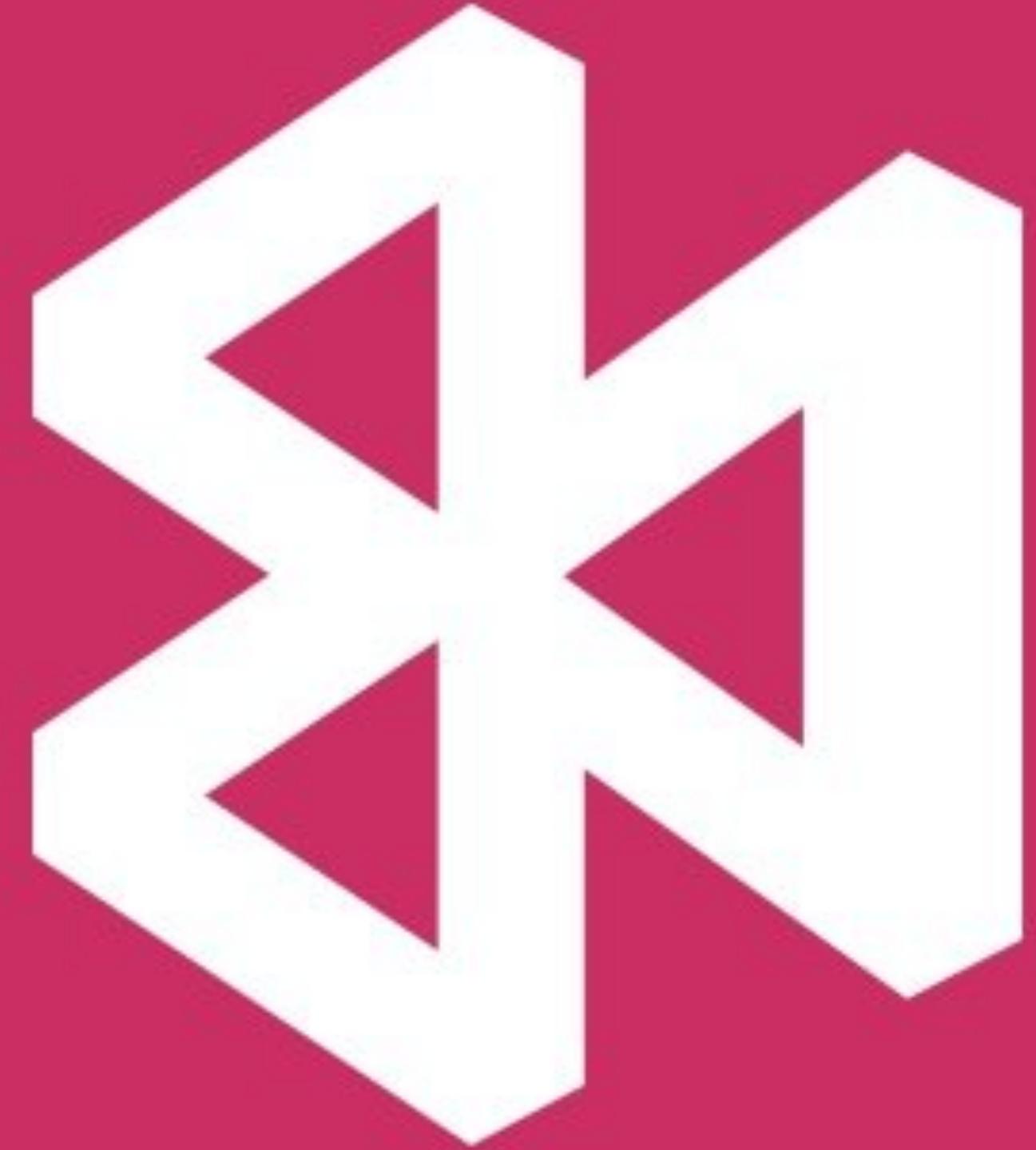
GitLab CI/Circle CI

- Very basic support(mac machines, xcode, gradle)
- Integration with the parent tool
- Yaml editor as an interface

GitLab CI/Circle CI

- Very basic support(mac machines, xcode, gradle)
- Integration with the parent tool
- Yaml editor as an interface
- Almost no mobile specific involved

App Center



App Center - Azure DevOps

App Center - Azure DevOps

- Part of the integrated environment(former TFS)

App Center - Azure DevOps

- Part of the integrated environment(former TFS)
- Distribution destinations - GP, TestFlight, Internal

App Center - Azure DevOps

- Part of the integrated environment(former TFS)
- Distribution destinations - GP, TestFlight, Internal
- Bitbucket, Github, VSTS, GitLab(!)

App Center - Azure DevOps

- Part of the integrated environment(former TFS)
- Distribution destinations - GP, TestFlight, Internal
- Bitbucket, Github, VSTS, GitLab(!)
- Configuration as Code 

App Center - Azure DevOps

- Part of the integrated environment(former TFS)
- Distribution destinations - GP, TestFlight, Internal
- Bitbucket, Github, VSTS, GitLab(!)
- Configuration as Code 
- Mobile Apps as first class citizens 

App Center - Azure DevOps

- Part of the integrated environment(former TFS)
- Distribution destinations - GP, TestFlight, Internal
- Bitbucket, Github, VSTS, GitLab(!)
- Configuration as Code 
- Mobile Apps as first class citizens 
- Cloud based 

App Center - Azure DevOps

- Part of the integrated environment(former TFS)
- Distribution destinations - GP, TestFlight, Internal
- Bitbucket, Github, VSTS, GitLab(!)
- Configuration as Code 
- Mobile Apps as first class citizens 
- Cloud based 
- Allows for Machine Pools(which solves Mac issue)

App Center - Azure DevOps

- Part of the integrated environment(former TFS)
- Distribution destinations - GP, TestFlight, Internal
- Bitbucket, Github, VSTS, GitLab(!)
- Configuration as Code 
- Mobile Apps as first class citizens 
- Cloud based 
- Allows for Machine Pools(which solves Mac issue)
- Webhooks

App Center - Azure DevOps

App Center - Azure DevOps

- Support for ad-hoc git servers 

App Center - Azure DevOps

- Support for ad-hoc git servers 
- SonarQube Support - 

App Center - Azure DevOps

- Support for ad-hoc git servers 
- SonarQube Support - 
- Local debug - 

App Center - When

App Center - When

- You already have Azure DevOps/Azure subscription

App Center - When

- You already have Azure DevOps/Azure subscription
- You're hosted in Bitbucket/Github

App Center - When

- You already have Azure DevOps/Azure subscription
- You're hosted in Bitbucket/Github
- You only want apps distribution solution

/ØNEVERCODE

Nevercode

Nevercode

- Mobile Centric CI/CD

Nevercode

- Mobile Centric CI/CD
- Distribution destinations - App Store Connect, Google Play, HockeyApp, Crashlytics, TestFairy

Nevercode

- Mobile Centric CI/CD
- Distribution destinations - App Store Connect, Google Play, HockeyApp, Crashlytics, TestFairy
- Bitbucket, GitHub or GitLab

Nevercode

- Mobile Centric CI/CD
- Distribution destinations - App Store Connect, Google Play, HockeyApp, Crashlytics, TestFairy
- Bitbucket, GitHub or GitLab
- Cloud based 

Nevercode

- Mobile Centric CI/CD
- Distribution destinations - App Store Connect, Google Play, HockeyApp, Crashlytics, TestFairy
- Bitbucket, GitHub or GitLab
- Cloud based 
- Mobile Apps as first class citizens 

Nevercode

- Mobile Centric CI/CD
- Distribution destinations - App Store Connect, Google Play, HockeyApp, Crashlytics, TestFairy
- Bitbucket, GitHub or GitLab
- Cloud based 
- Mobile Apps as first class citizens 
- Webhooks 

Nevercode

Nevercode

- Configuration as Code 

Nevercode

- Configuration as Code ✗
- Pricy ✗

Nevercode - When

Nevercode - When

- Flutter apps

Bitrise

Bitrise

- Mobile Centric CI/CD

Bitrise

- Mobile Centric CI/CD
- Distribution destinations - GP, TestFlight, TestFairy, App Center, Whatever

Bitrise

- Mobile Centric CI/CD
- Distribution destinations - GP, TestFlight, TestFairy, App Center, Whatever
- Bitbucket, Github, Custom

Bitrise

- Mobile Centric CI/CD
- Distribution destinations - GP, TestFlight, TestFairy, App Center, Whatever
- Bitbucket, Github, Custom
- Configuration as Code 

Bitrise

- Mobile Centric CI/CD
- Distribution destinations - GP, TestFlight, TestFairy, App Center, Whatever
- Bitbucket, Github, Custom
- Configuration as Code 
- Mobile Apps as first class citizens 

Bitrise

- Mobile Centric CI/CD
- Distribution destinations - GP, TestFlight, TestFairy, App Center, Whatever
- Bitbucket, Github, Custom
- Configuration as Code 
- Mobile Apps as first class citizens 
- Cloud based 

Bitrise

- Mobile Centric CI/CD
- Distribution destinations - GP, TestFlight, TestFairy, App Center, Whatever
- Bitbucket, Github, Custom
- Configuration as Code 
- Mobile Apps as first class citizens 
- Cloud based 
- Webhooks

Bitrise

Bitrise

- Support for ad-hoc git servers 

Bitrise

- Support for ad-hoc git servers 
- Support CI job triggers on push to Any branch 

Bitrise

- Support for ad-hoc git servers 
- Support CI job triggers on push to Any branch 
- SonarQube Support with a community extension 

Bitrise

- Support for ad-hoc git servers 
- Support CI job triggers on push to Any branch 
- SonarQube Support with a community extension 
- Local debug - 

Bitrise

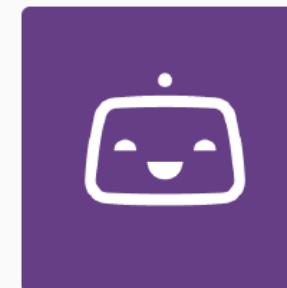
- Support for ad-hoc git servers 
- Support CI job triggers on push to Any branch 
- SonarQube Support with a community extension 
- Local debug - 
- Open source



Search or jump to...

/

Pull requests Issues Marketplace Explore



Bitrise

Continuous Integration and Continuous Delivery for mobile apps

📍 Hungary

🔗 https://www.bitrise.io

✉️ letsconnect@bitrise.io

Report abu...

Repositories 124

People 4

Pinned repositories

bitrise

Bitrise runner CLI - run your automations on your Mac or Linux machine -

Go 421 79

bitrise.io

Bitrise.io public issues

56 19

bitrise-steplib

New Bitrise StepLib

Go 66 131

devcenter

CSS 118 148

build.issues

Issue / bug tracker for build related issues

3 3

bitrise-blog

HTML 2 3

108

Bitrise

Bitrise

- Rather slow 

Bitrise

- Rather slow ✗
- No Flutter support(yet) ✗

Bitrise

- Rather slow ✗
- No Flutter support(yet) ✗
- Flutter support(already) ✓

Services Comparison

App Center

Simplicity 

Device Cloud 

Crash Reporting 

Difficult customization  No conf as service 

Funny jokes

Nevercode

Flutter 

Build cache 

No Xamarin 

No conf as service 

Funny jokes

Bitrise

Flexible 

Build cache 

Different inf. stacks 

Open Source 

Slow 



DevOps Practices

- Secured process 
- Continuous integration 
- Continuous delivery 
- Frequent releases(Canary/BG deployments) 
- Automated testing 
- Monitoring 
- Automated rollback 

Summary

Summary

- Mobile is a serious business requiring automation

Summary

- Mobile is a serious business requiring automation
- There are some caveats and issues which are overcomeable

Summary

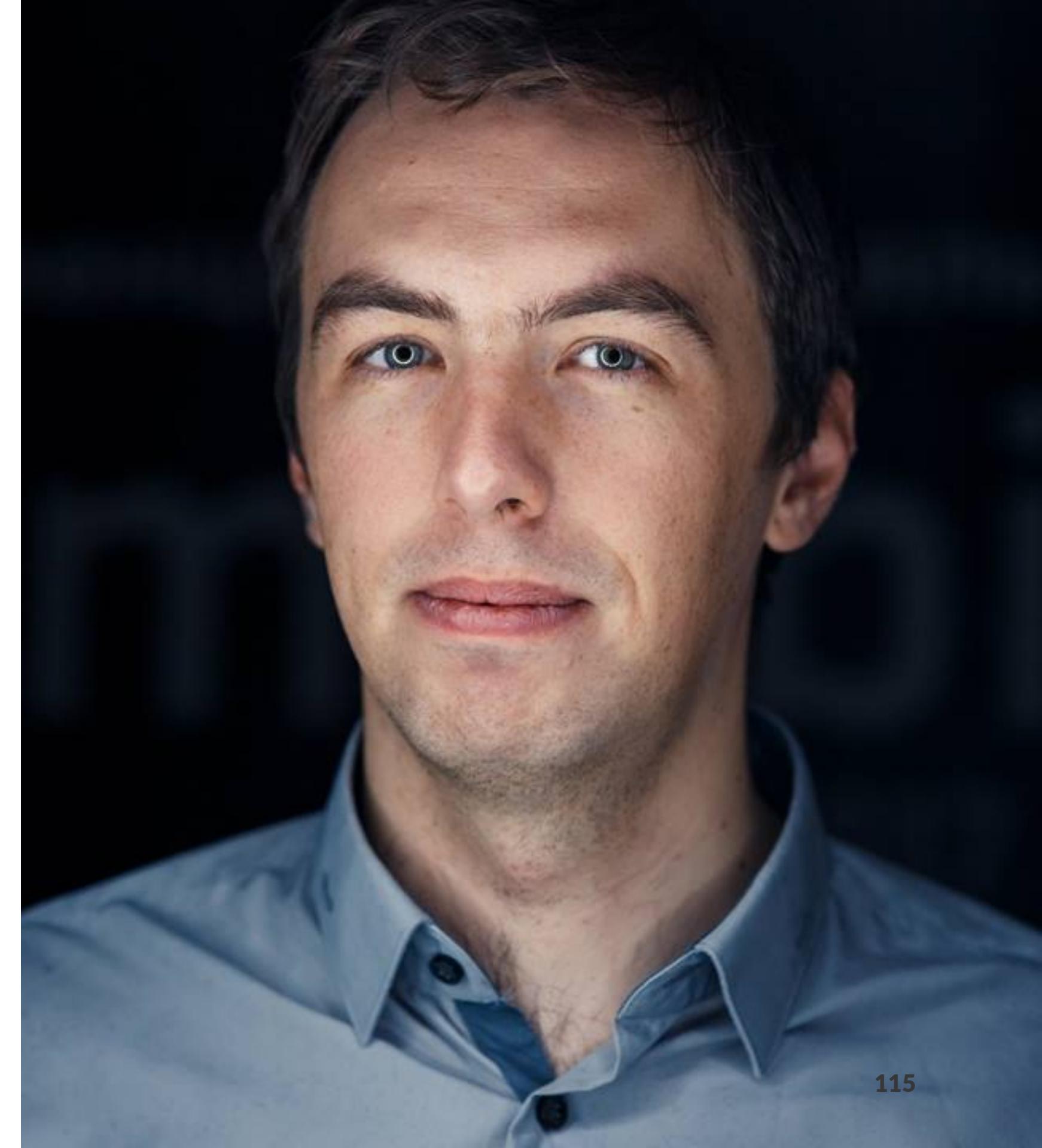
- Mobile is a serious business requiring automation
- There are some caveats and issues which are overcomeable
- Except rollbacks



<epam>



Links



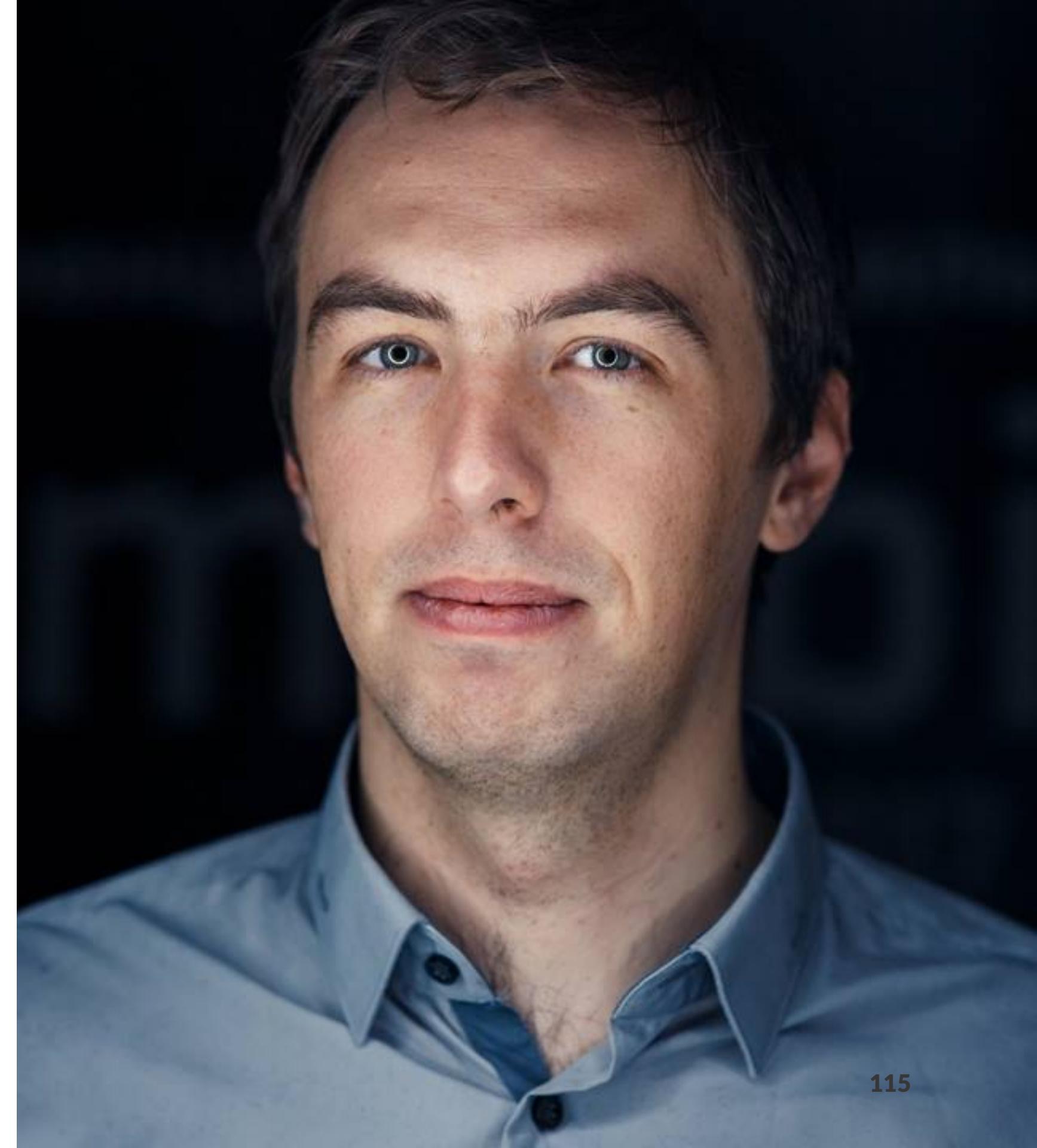
Links

- Twitter, telegram: @vvsevolodovich



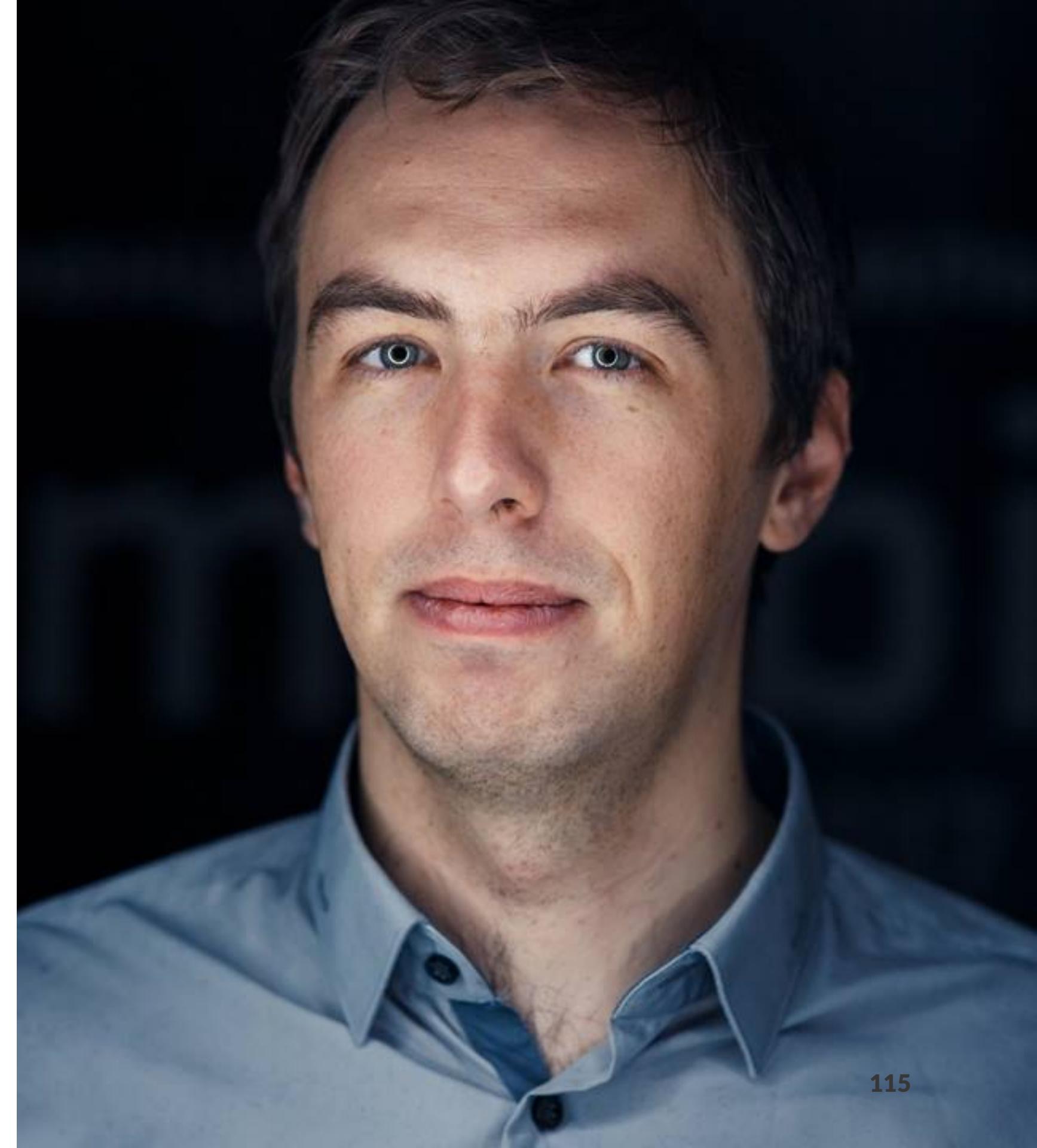
Links

- Twitter, telegram: @vvsevolodovich
- Talks: <http://speakerdeck.com/vlivanov>



Links

- Twitter, telegram: @vvsevolodovich
- Talks: <http://speakerdeck.com/vlivanov>
- Email: Vladimir_Ivanov4@epam.com



Links

- Twitter, telegram: @vvsevolodovich
- Talks: <http://speakerdeck.com/vlivanov>
- Email: Vladimir_Ivanov4@epam.com
- <https://mobiusconf.com/>

