

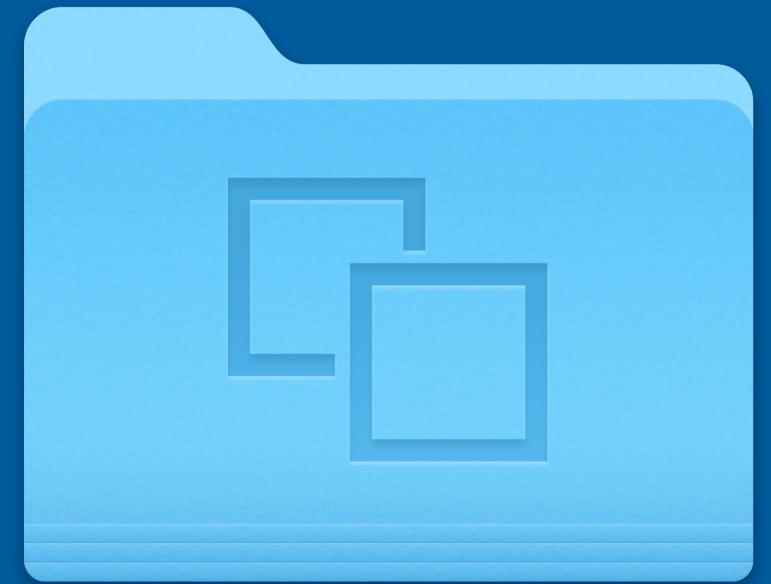
" I HAVE A FRAMEWORK IDEA "  
REPEAT LESS. SHARE MORE.

FABIO MILANO - @IAMFABIOMILANO

I HAVE AN APP IDEA.  
- SOMEONE, SOMEWHERE.

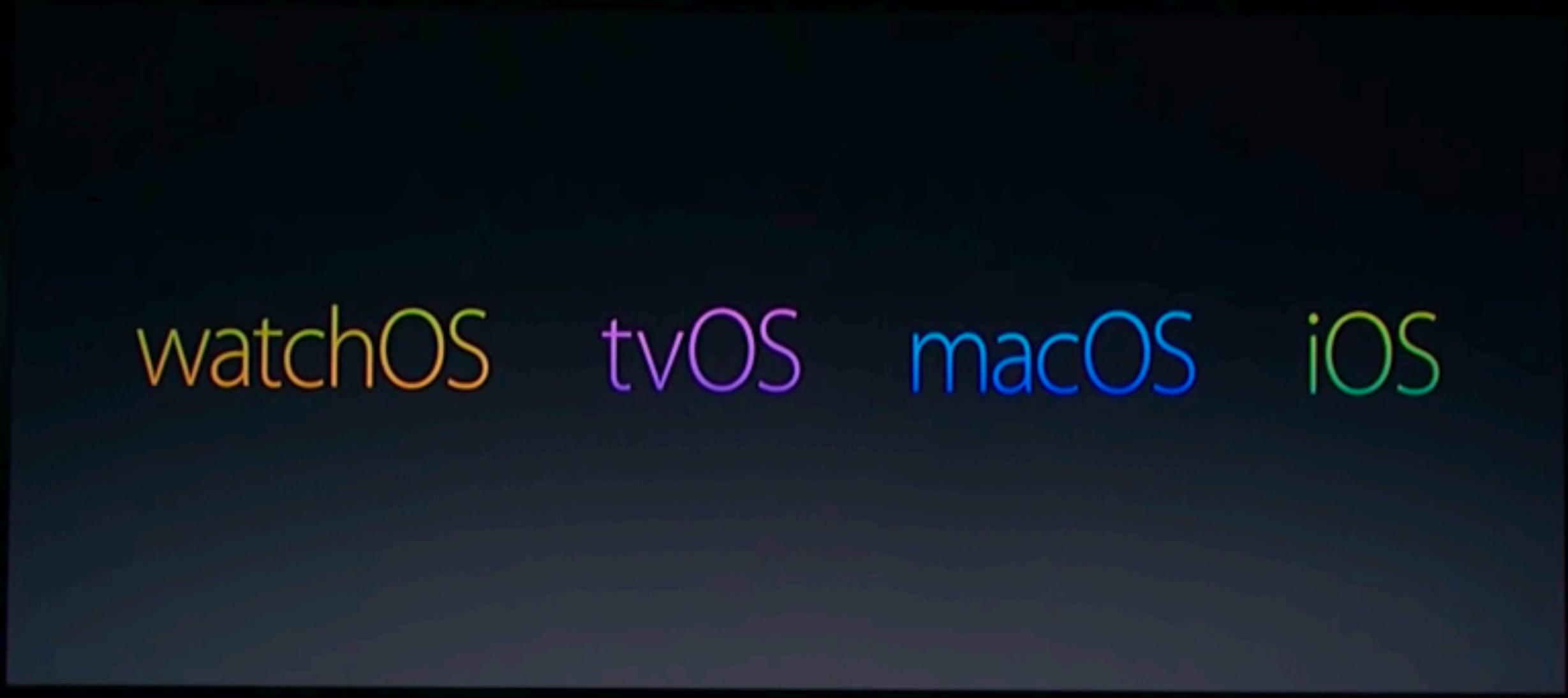
# WHAT IS A FRAMEWORK?

# WHAT IS INSIDE A FRAMEWORK?



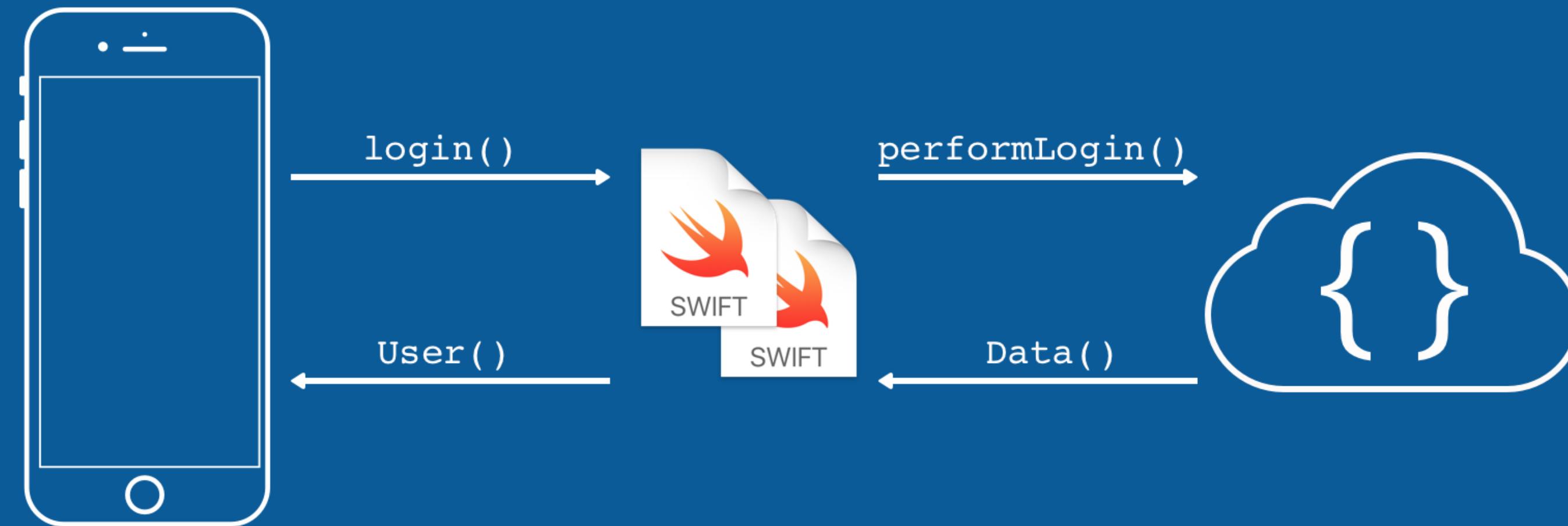
...

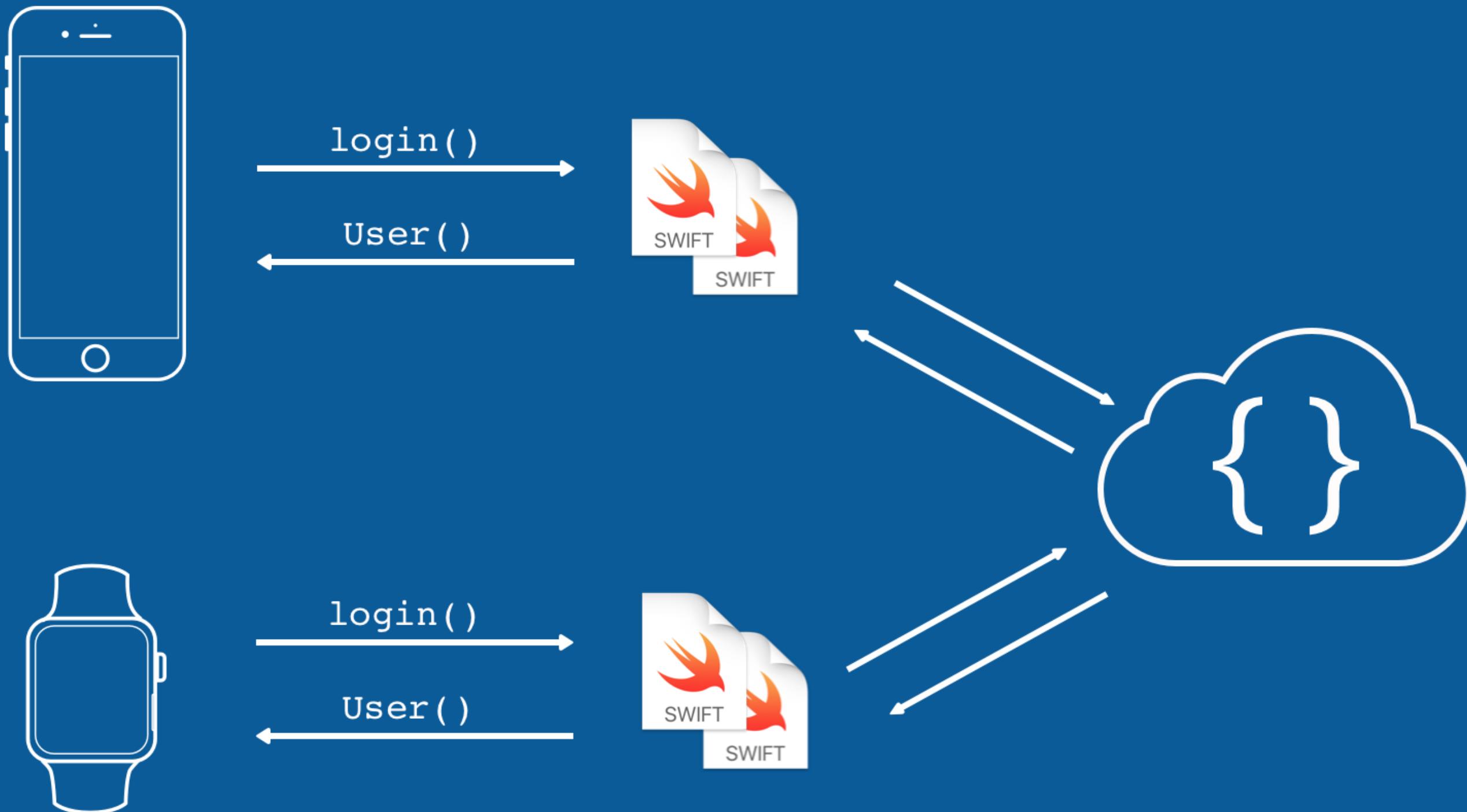
# WHY FRAMEWORKS?

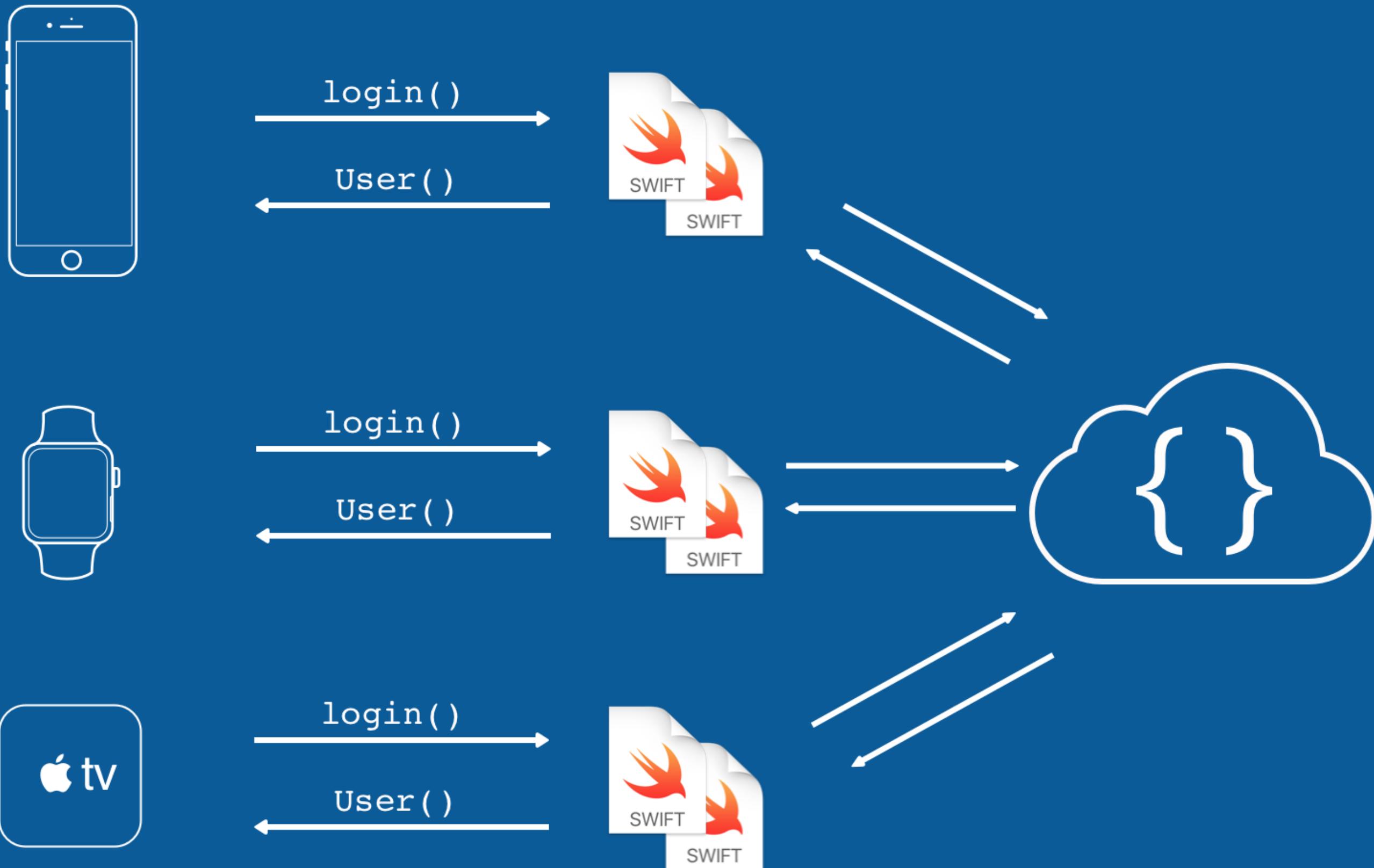


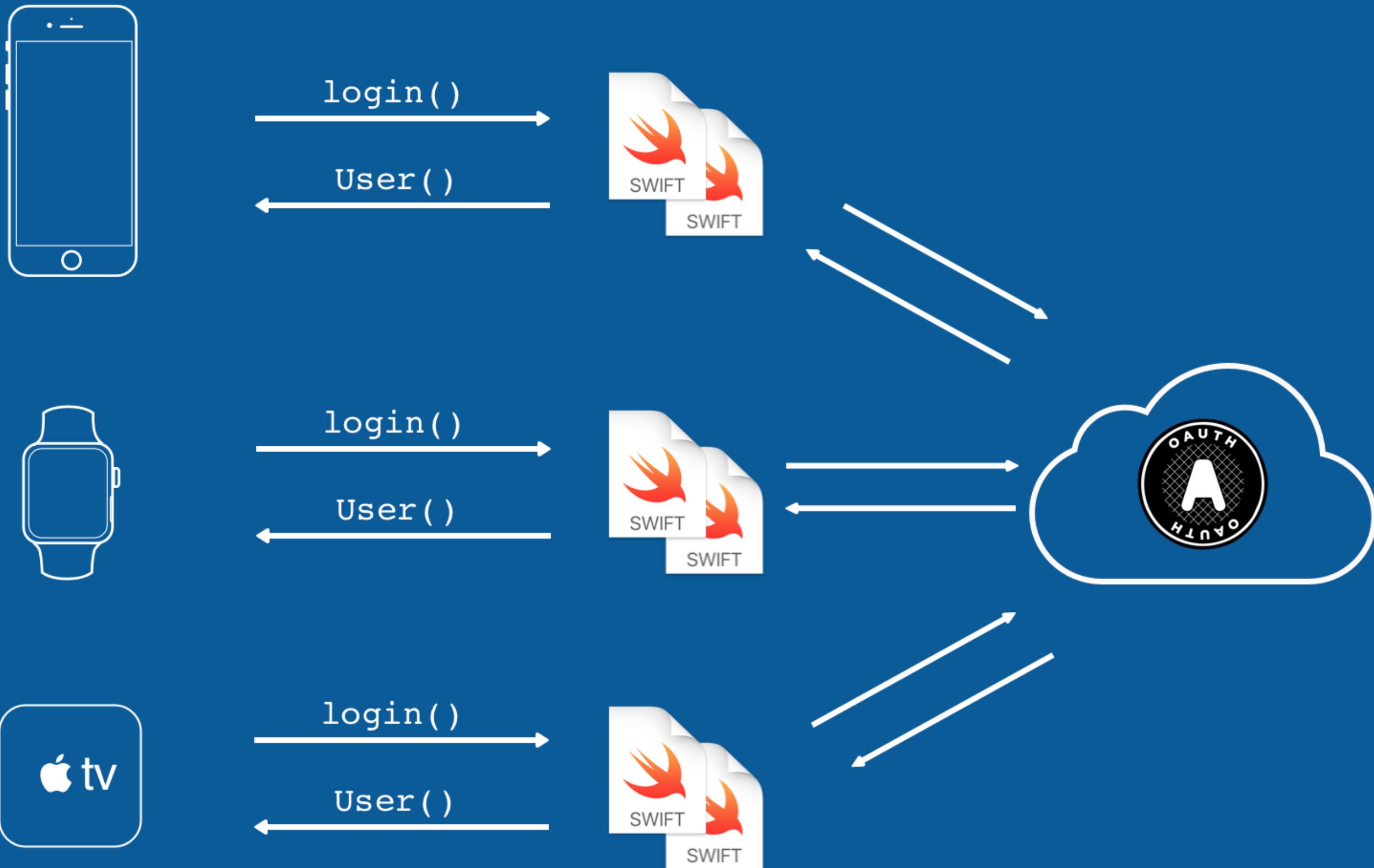
watchOS   tvOS   macOS   iOS

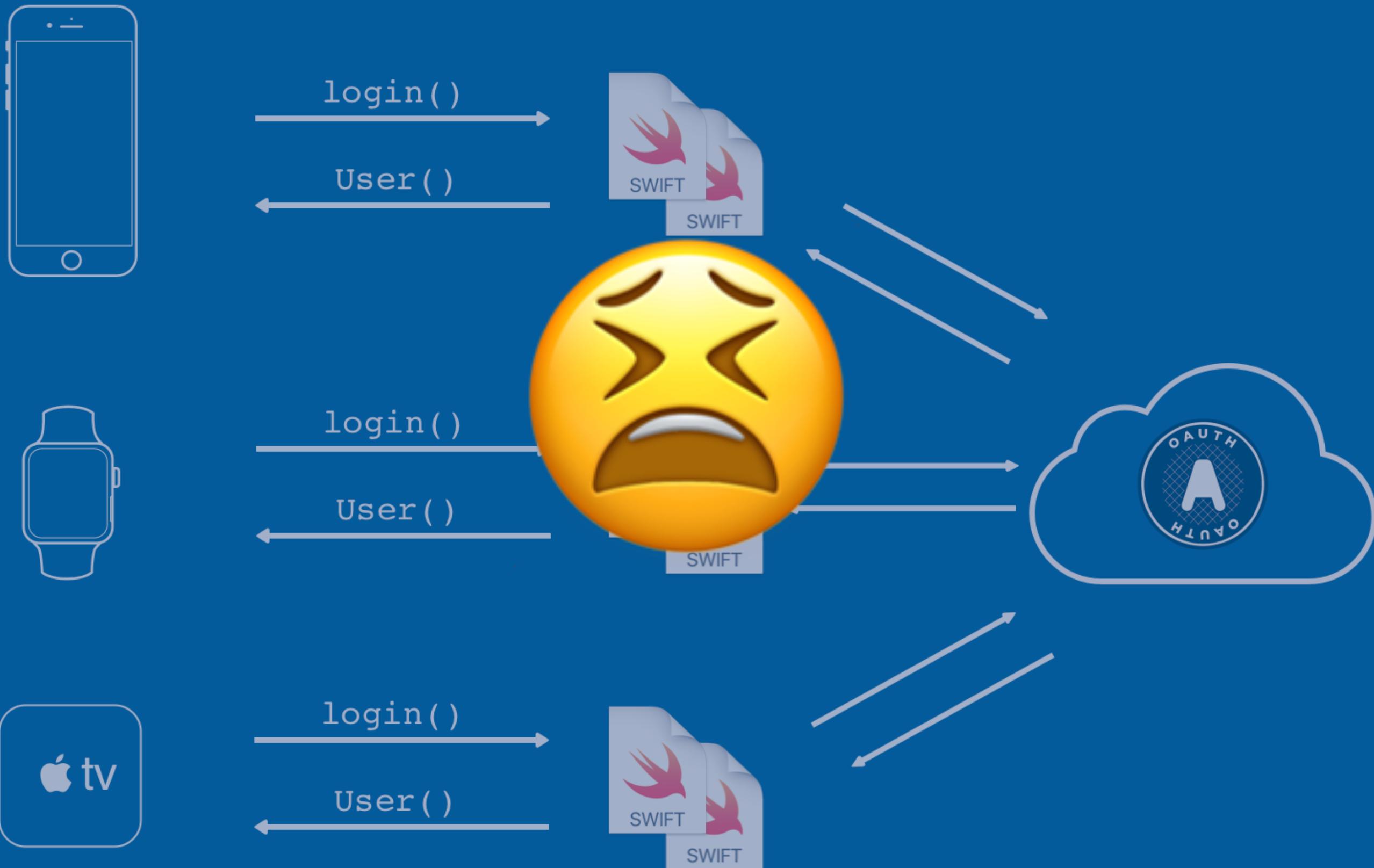


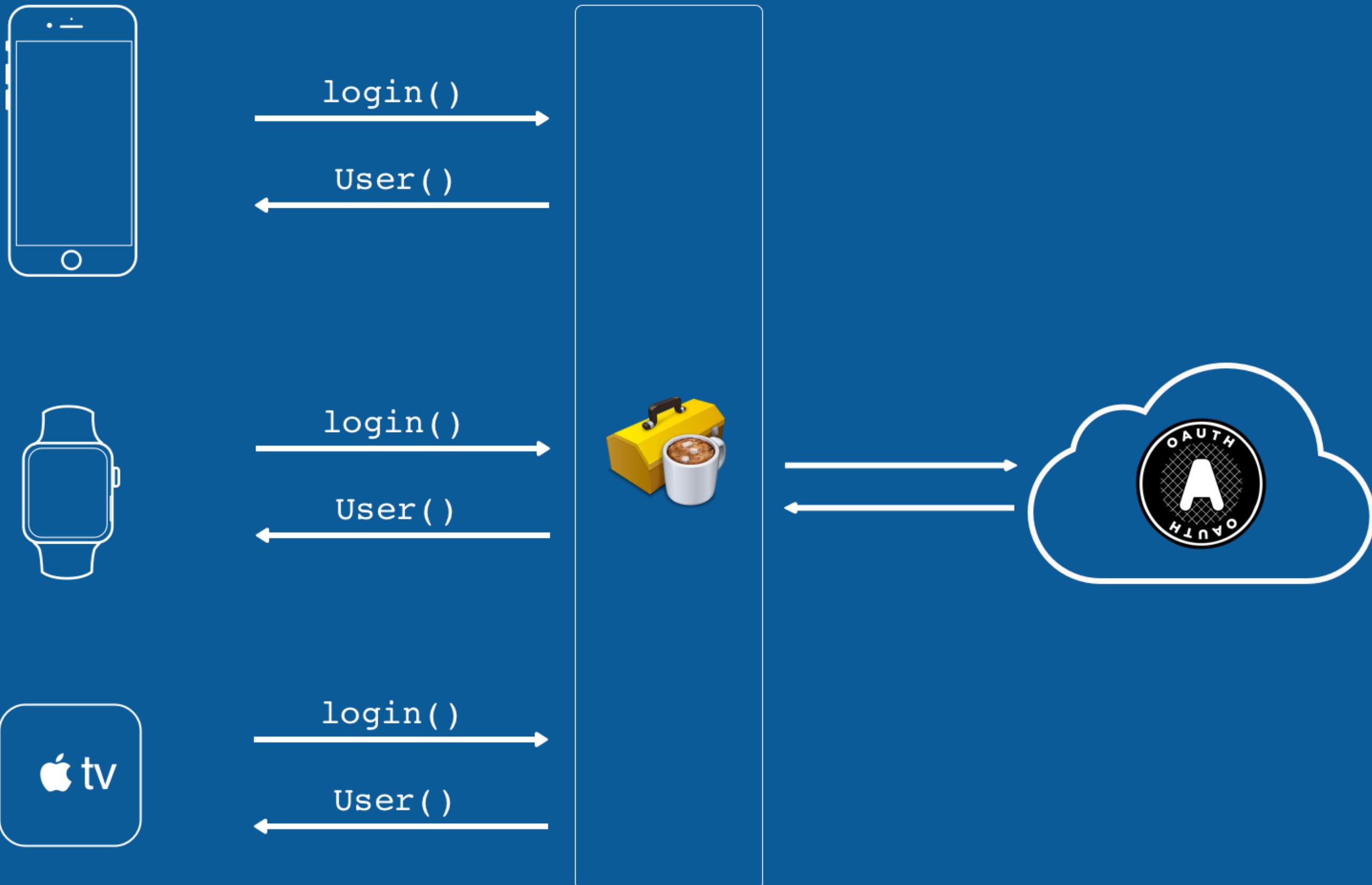




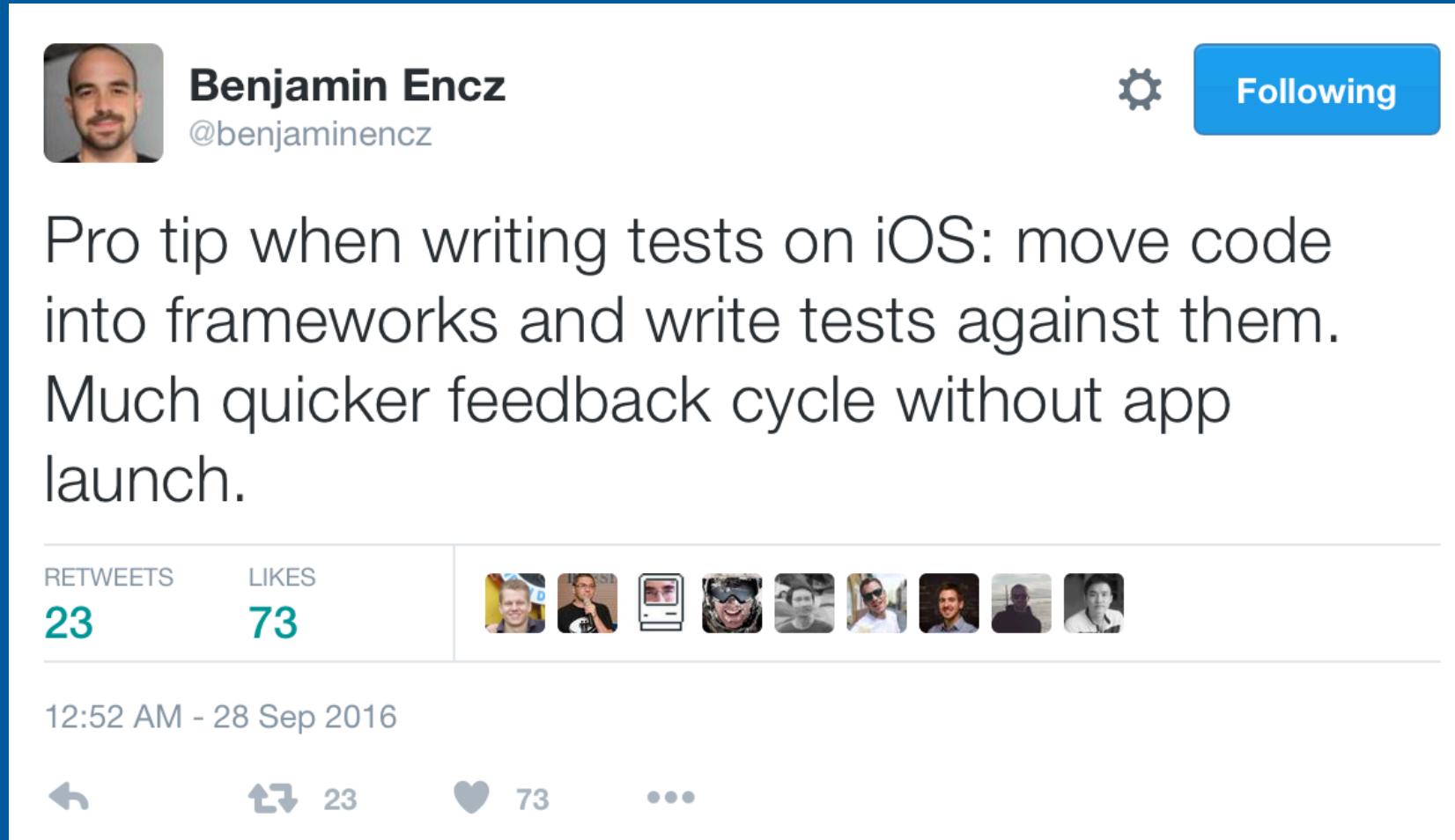








# WHY FRAMEWORKS?



Benjamin Encz (@benjaminencz)

Following

Pro tip when writing tests on iOS: move code into frameworks and write tests against them. Much quicker feedback cycle without app launch.

RETWEETS 23 LIKES 73

12:52 AM - 28 Sep 2016

23 73

# WITH A FRAMEWORK



**SHARING IS  
CARING**

## > SEPARATION OF CONCERNS

## > SEPARATION OF CONCERNS \*

\* IT MIGHT AFFECT APP LAUNCH TIME (USE DYLD\_PRINT\_STATISTICS FOR STATS).

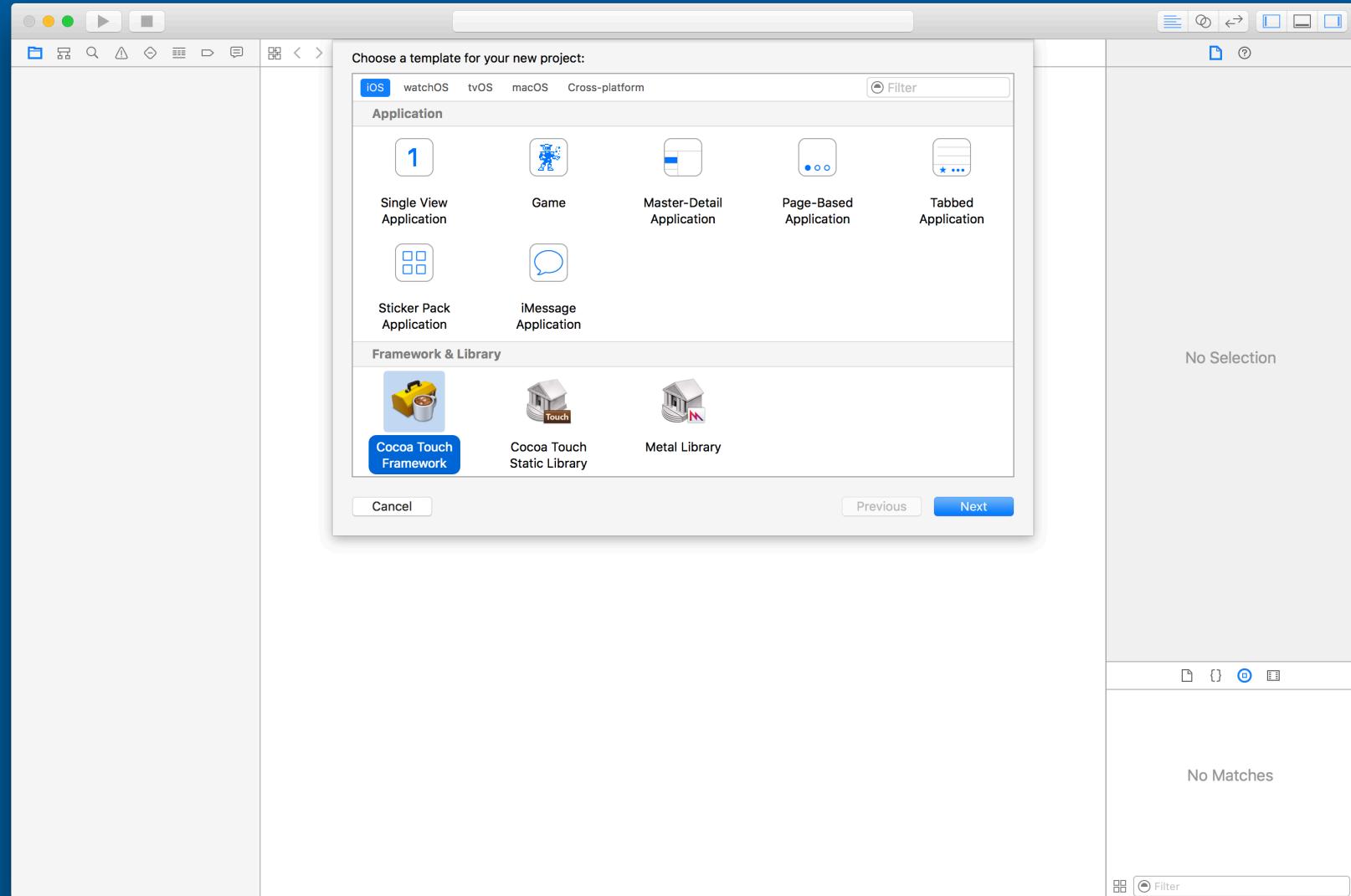
- > SEPARATION OF CONCERNS
- > ENCAPSULATION

- > SEPARATION OF CONCERNS
- > ENCAPSULATION
- > MAINTAINABILITY

# CREATE A FRAMEWORK



# CREATE A FRAMEWORK



# THE FOLDER STRUCTURE

# THE FOLDER STRUCTURE

example-package-awesome-library

  └── Sources

    └── AwesomeStruct.swift

    └── AwesomeClass.swift

    └── AwesomeProtocol.swift

  └── Tests

    └── AwesomeTests

      └── AwesomeClassTest.swift

# DONE!



**..ALMOST..**

**WHAT DO WE WANT?**

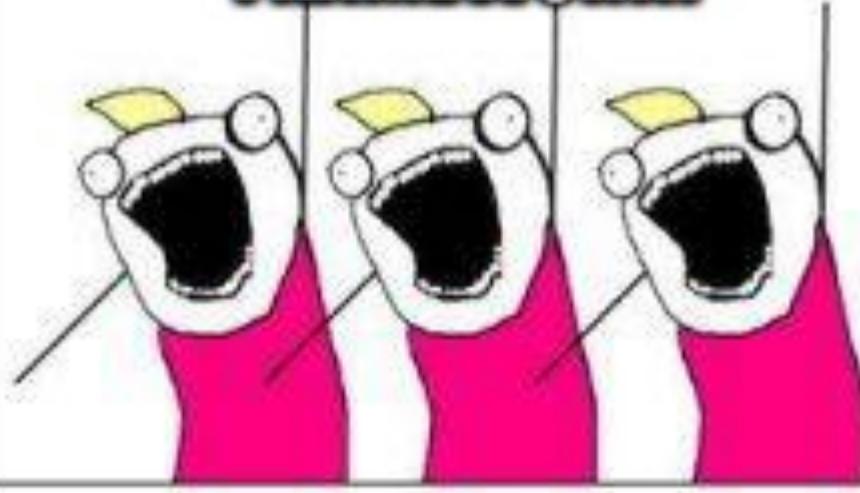


**WHEN DO WE WANT IT?**



[imgflip.com](http://imgflip.com)

**BUILD A MULTI-PLATFORM  
FRAMEWORK!**

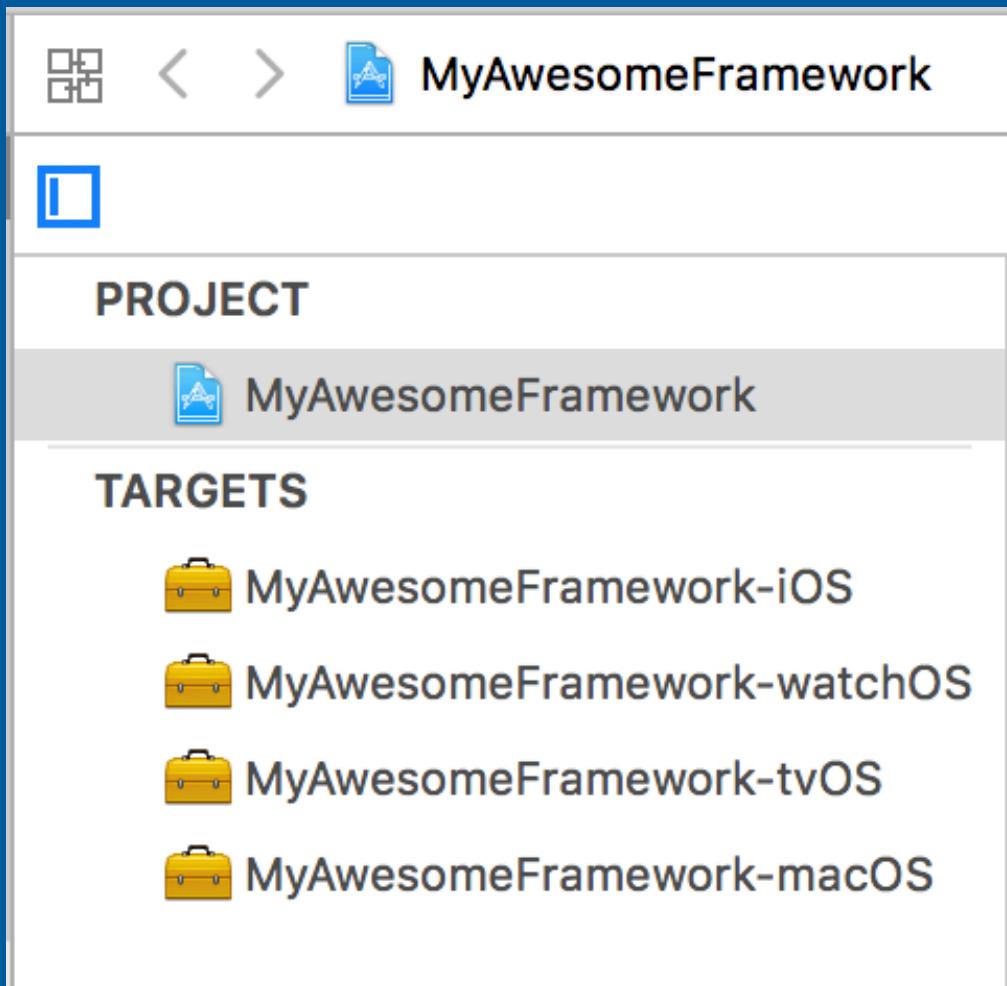


**AT LEAST UNDERSTAND  
HOW TODAY!**

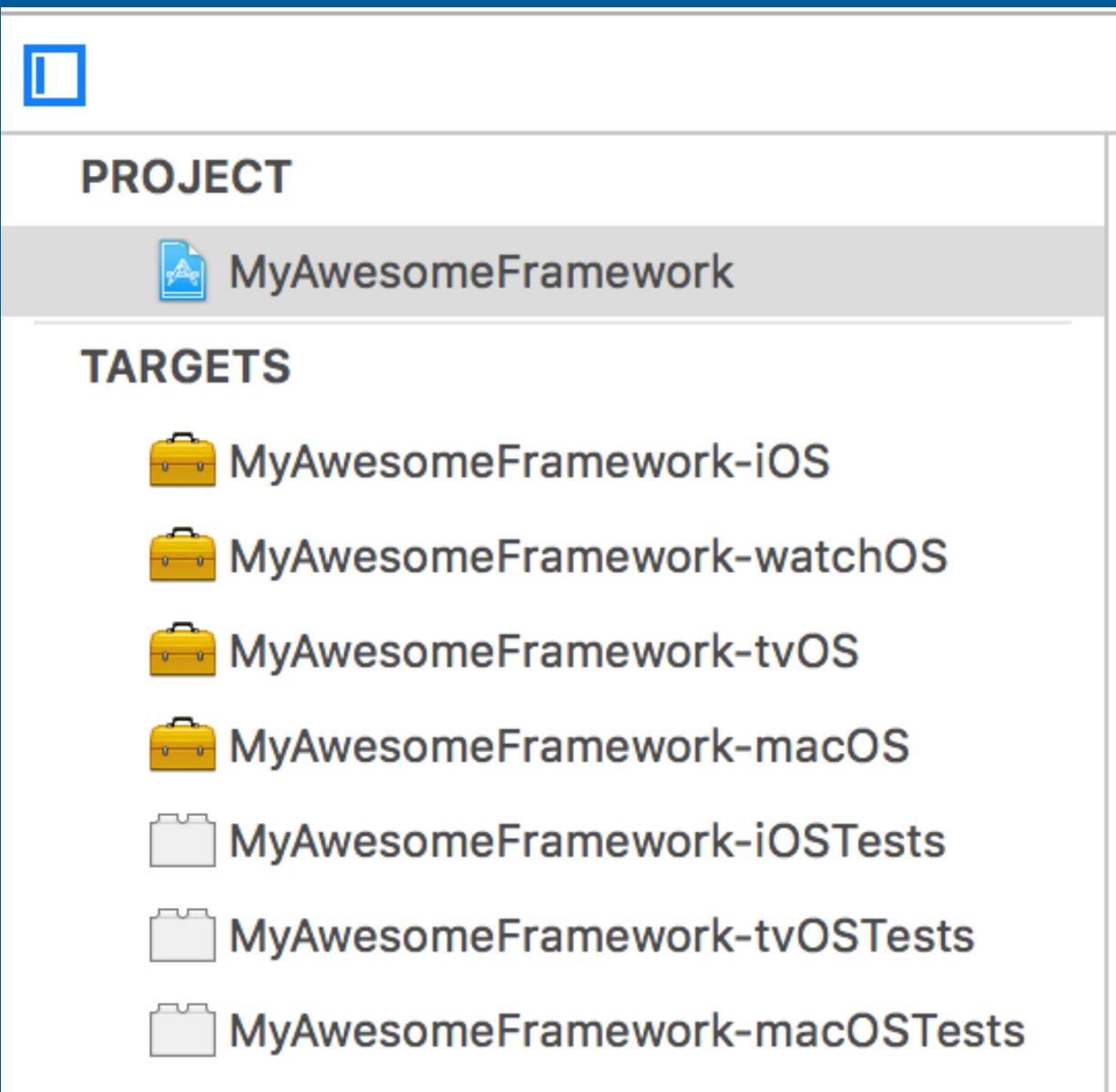


# CREATE FRAMEWORK

## THE PLATFORMS



# MULTI PLATFORM TESTS





## PROJECT



MyAwesomeFramework

## TARGETS



MyAwesomeFramework-iOS



MyAwesomeFramework-watchOS



MyAwesomeFramework-tvOS



MyAwesomeFramework-macOS



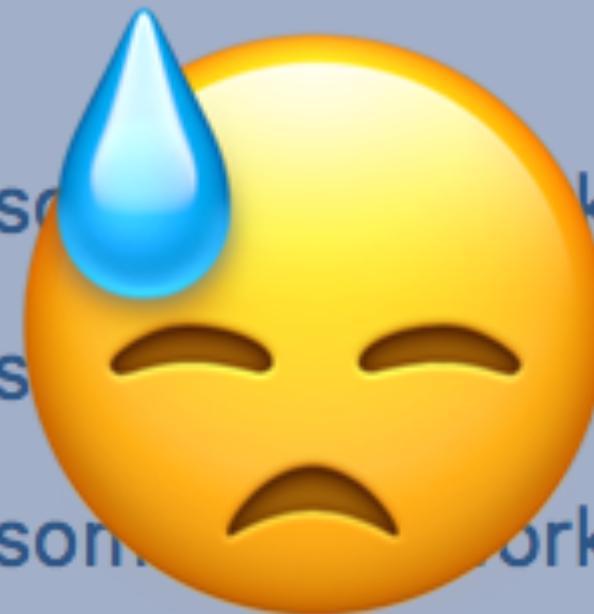
MyAwesomeFramework-iOSTests



MyAwesomeFramework-tvOTests



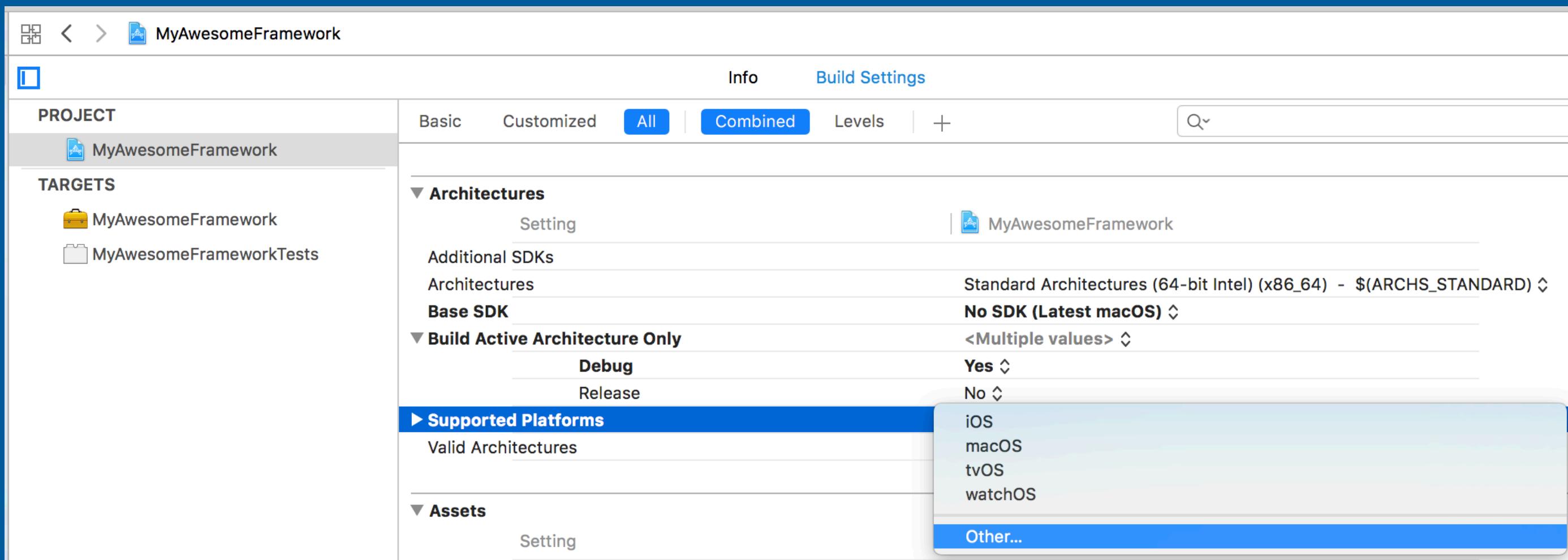
MyAwesomeFramework-macOTests





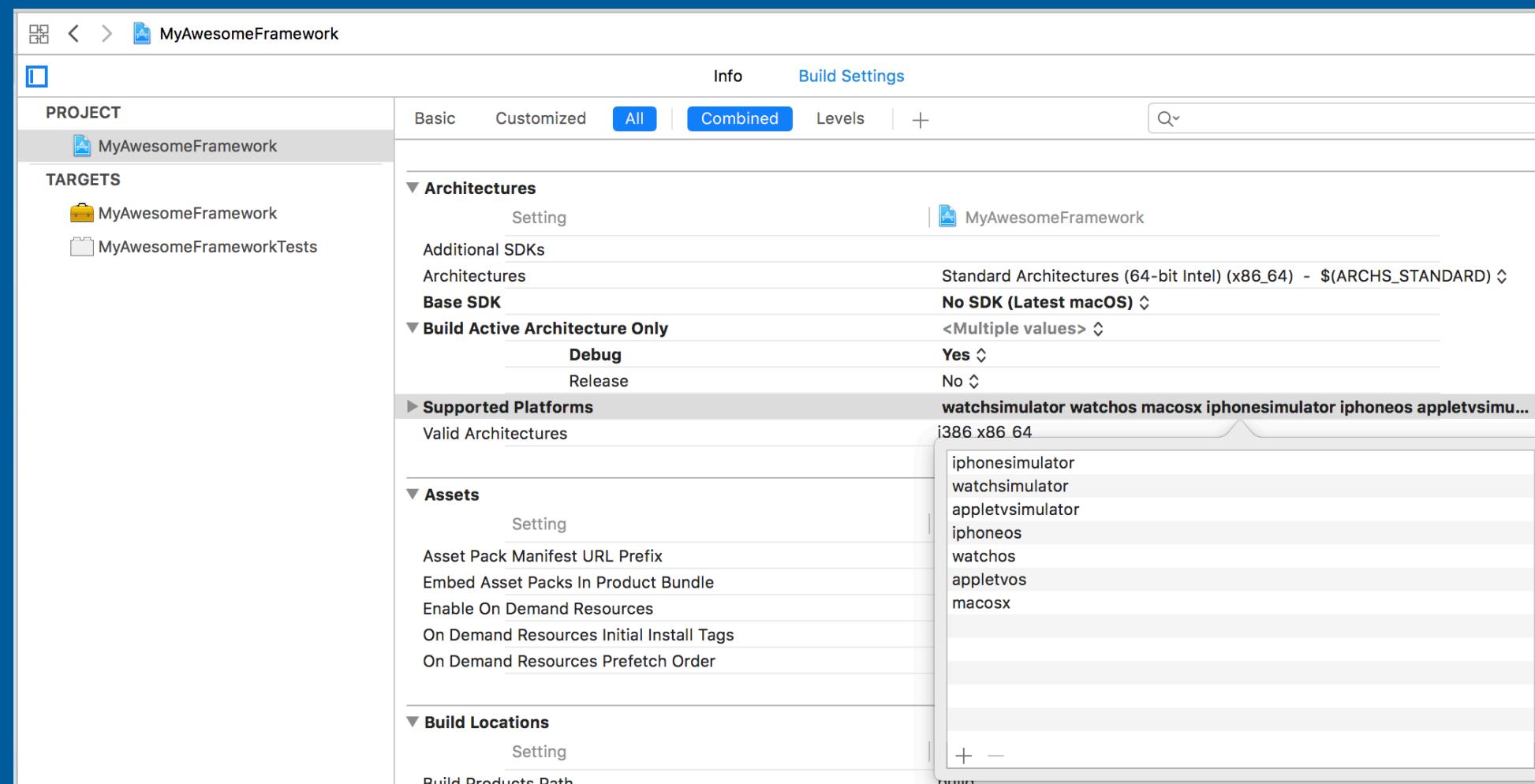
# THE BUILD SETTINGS

## SUPPORTED PLATFORMS



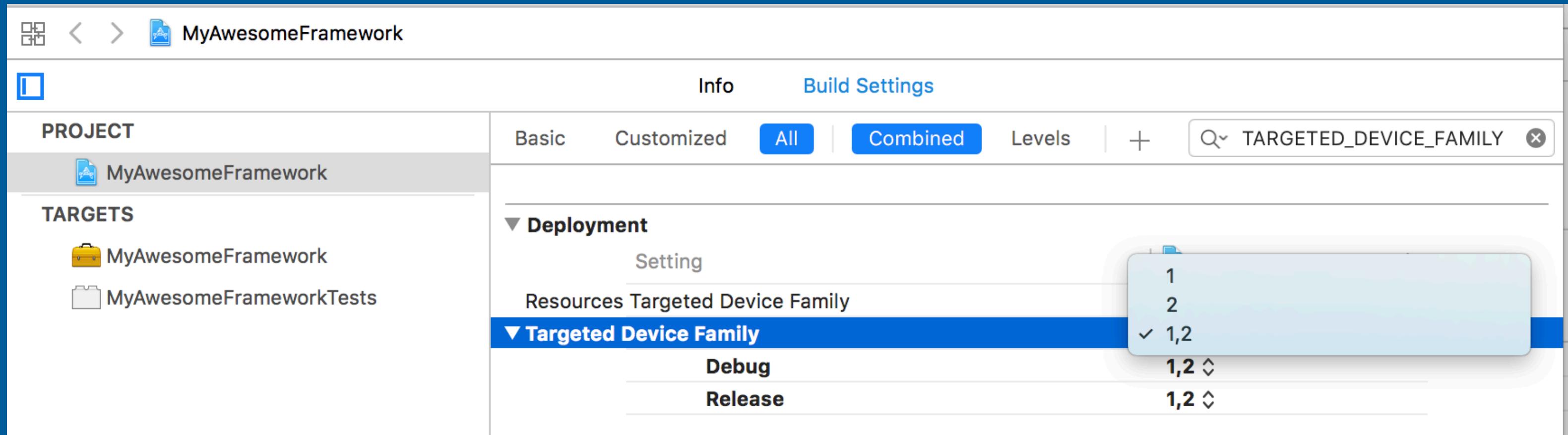
# THE BUILD SETTINGS

## SUPPORTED PLATFORMS



# THE BUILD SETTINGS

## TARGETED DEVICE FAMILY



# THE .xccconfig FILE



Choose a template for your new file:

ios watchOS tvOS macOS

Filter

Resource



Rich Text File



SceneKit Particle System File



SceneKit Scene File



Settings Bundle



SpriteKit Action



SpriteKit Particle File



SpriteKit Scene



SpriteKit Tile Set



Sticker Catalog



Strings File

Other



Empty



Assembly File



Configuration  
Settings File



PCH File



Shell Script

Cancel

Previous

Next

# THE STRANGE CASE OF TARGETED\_DEVICE\_FAMILY

// Config.xcconfig

TARGETED\_DEVICE\_FAMILY = 1,2,3,4

 Info Build Settings

**PROJECT**

 MyAwesomeFramework

**TARGETS**

 MyAwesomeFramework

 MyAwesomeFrameworkTests

**Deployment Target**

macOS Deployment Target  ▾

**Configurations**

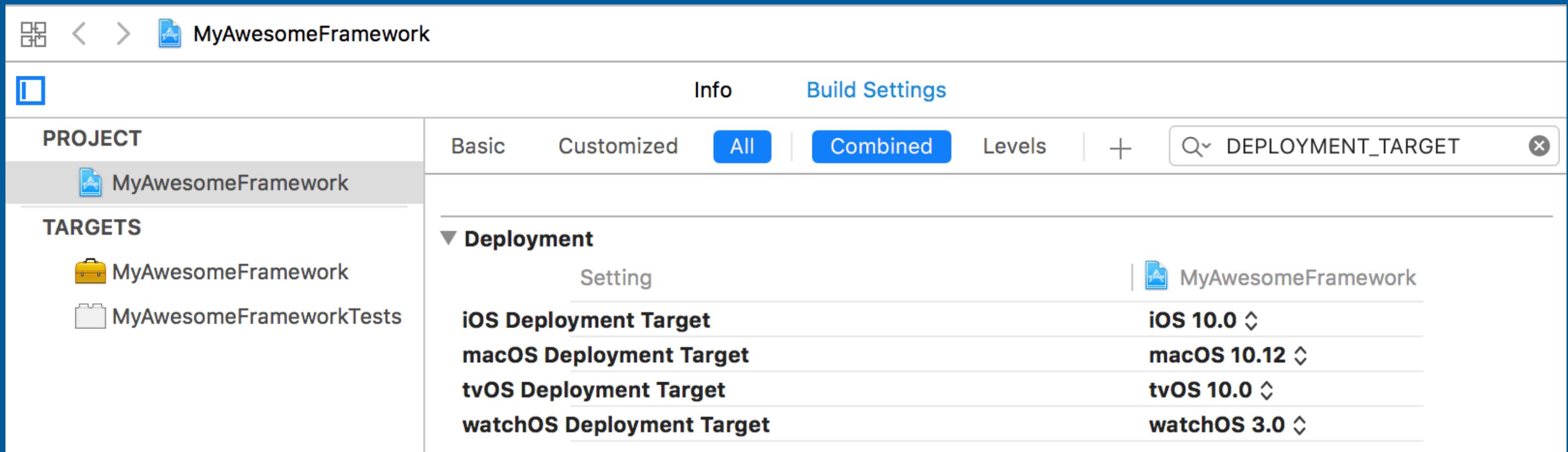
Name	Based on Configuration File
<b>Debug</b>	1 Configuration Set
 MyAwesomeFramework	MyAwesomeConfig ▾
 MyAwesomeFramework	None ▾
 MyAwesomeFrameworkTests	None ▾
<b>Release</b>	1 Configuration Set
 MyAwesomeFramework	MyAwesomeConfig ▾
 MyAwesomeFramework	None ▾
 MyAwesomeFrameworkTests	None ▾

+ -

Use  ▾ for command-line builds

# THE BUILD SETTINGS

## THE DEPLOYMENT TARGET



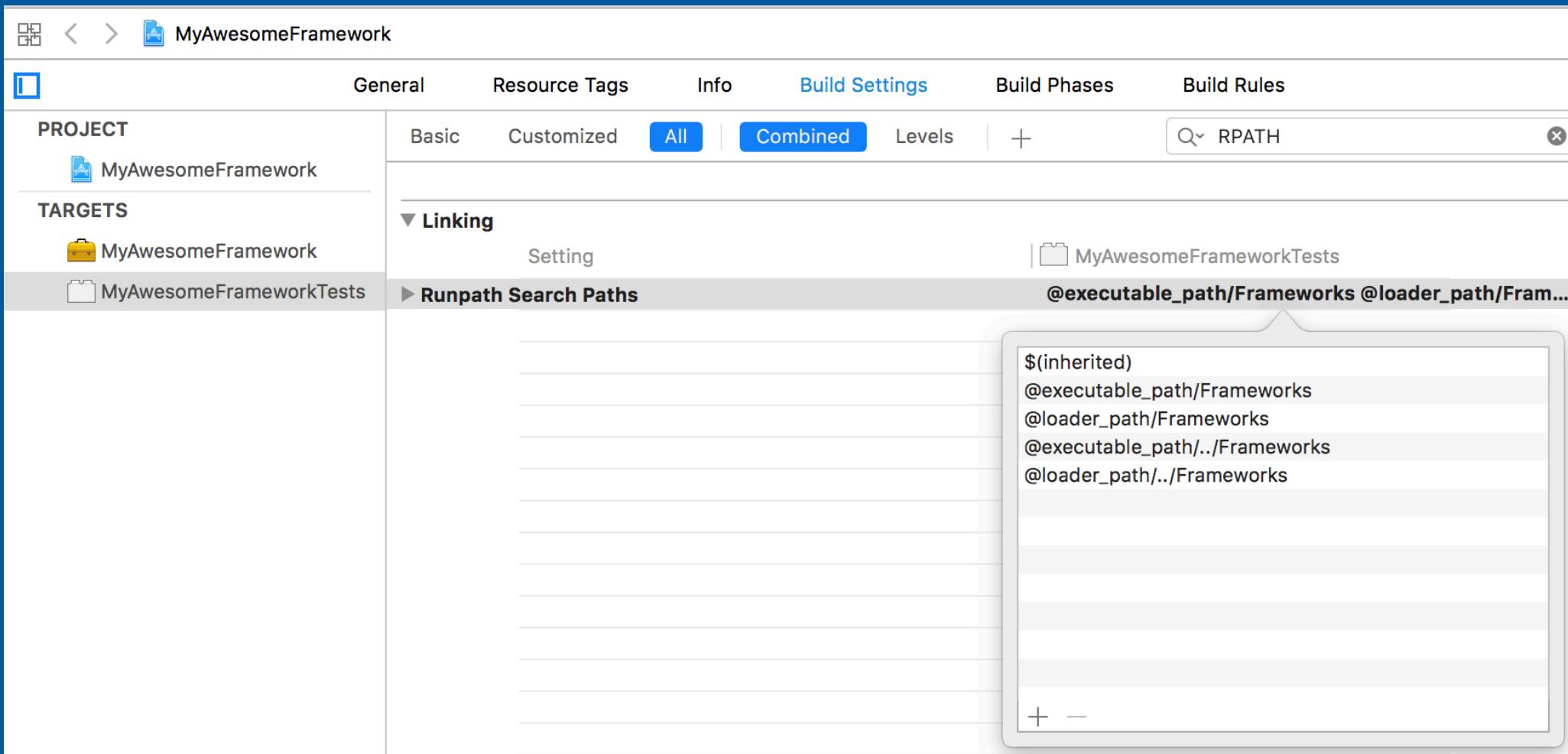
The screenshot shows the Xcode build settings interface for a project named "MyAwesomeFramework". The left sidebar lists the project and its targets: "MyAwesomeFramework" (selected) and "MyAwesomeFrameworkTests". The main area is titled "Build Settings" and shows the "All" tab selected. Under the "Deployment" section, there are four dropdown menus for deployment targets:

Setting	Value
iOS Deployment Target	iOS 10.0
macOS Deployment Target	macOS 10.12
tvOS Deployment Target	tvOS 10.0
watchOS Deployment Target	watchOS 3.0

```
/**  
Authorize by requesting a device code.  
  
- Parameters:  
  - completion: The completion block.  
    - String: The device code to use to finish the authorization flow.  
  
  - see: https://tools.ietf.org/html/draft-denniss-oauth-device-flow-00  
*/  
public func authorizeViaDeviceCode(completion: (String) -> Void) throws { ... }  
  
#if os(iOS) || os(OSX)  
  
import WebKit  
  
/**  
Authorize via standard OAuth 2.0 authorization flow.  
  
- Parameters:  
  - completion: The completion block.  
    - WKWebView: The web page where the user can authorize the application.  
*/  
public func authorize(completion: (WKWebView) -> Void) throws { ... }  
#endif
```

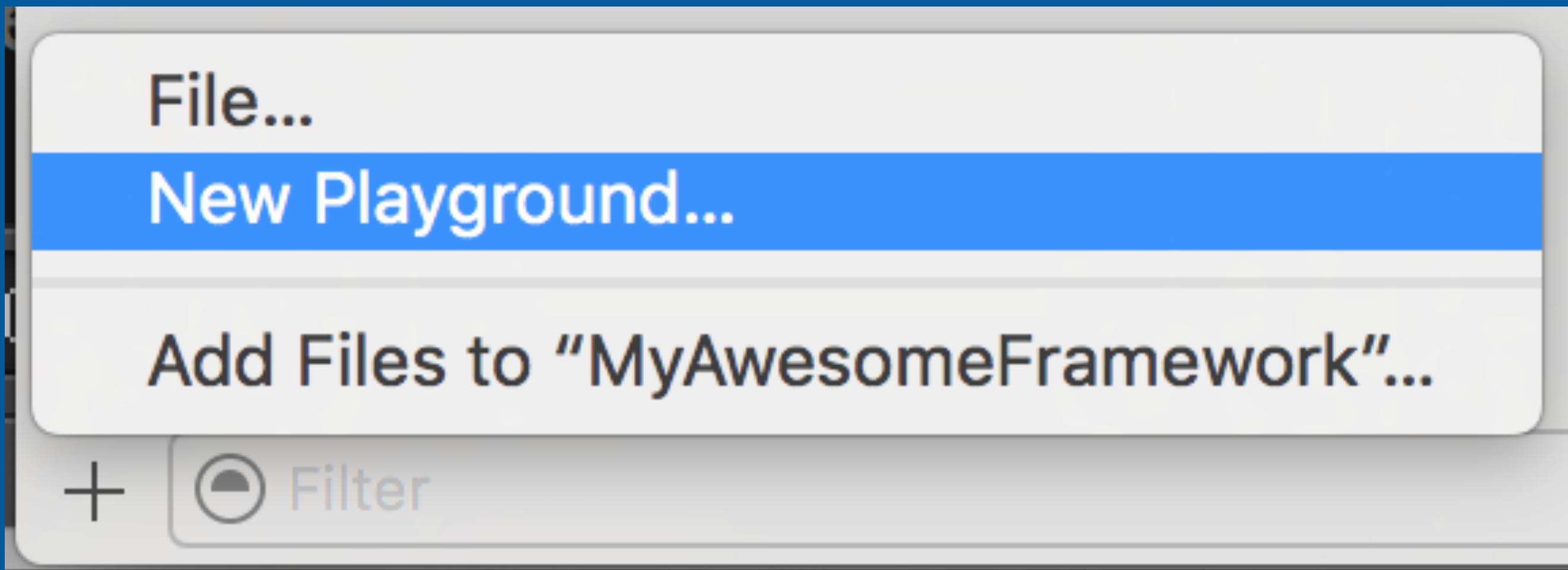
# THE BUILD SETTINGS

## RUN PATH SEARCH PATHS (A.K.A rpath)



# PLAYGROUND:

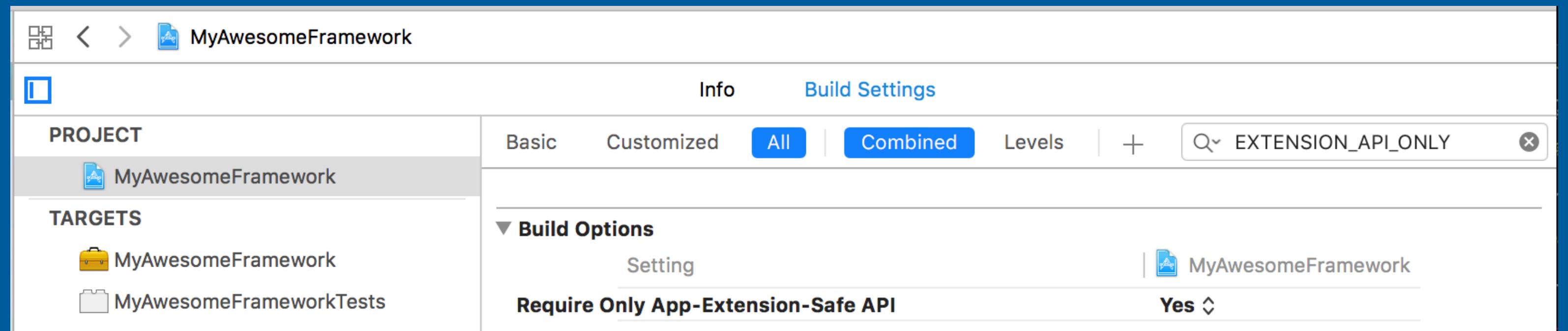
## A PLACE WHERE PEOPLE CAN PLAY

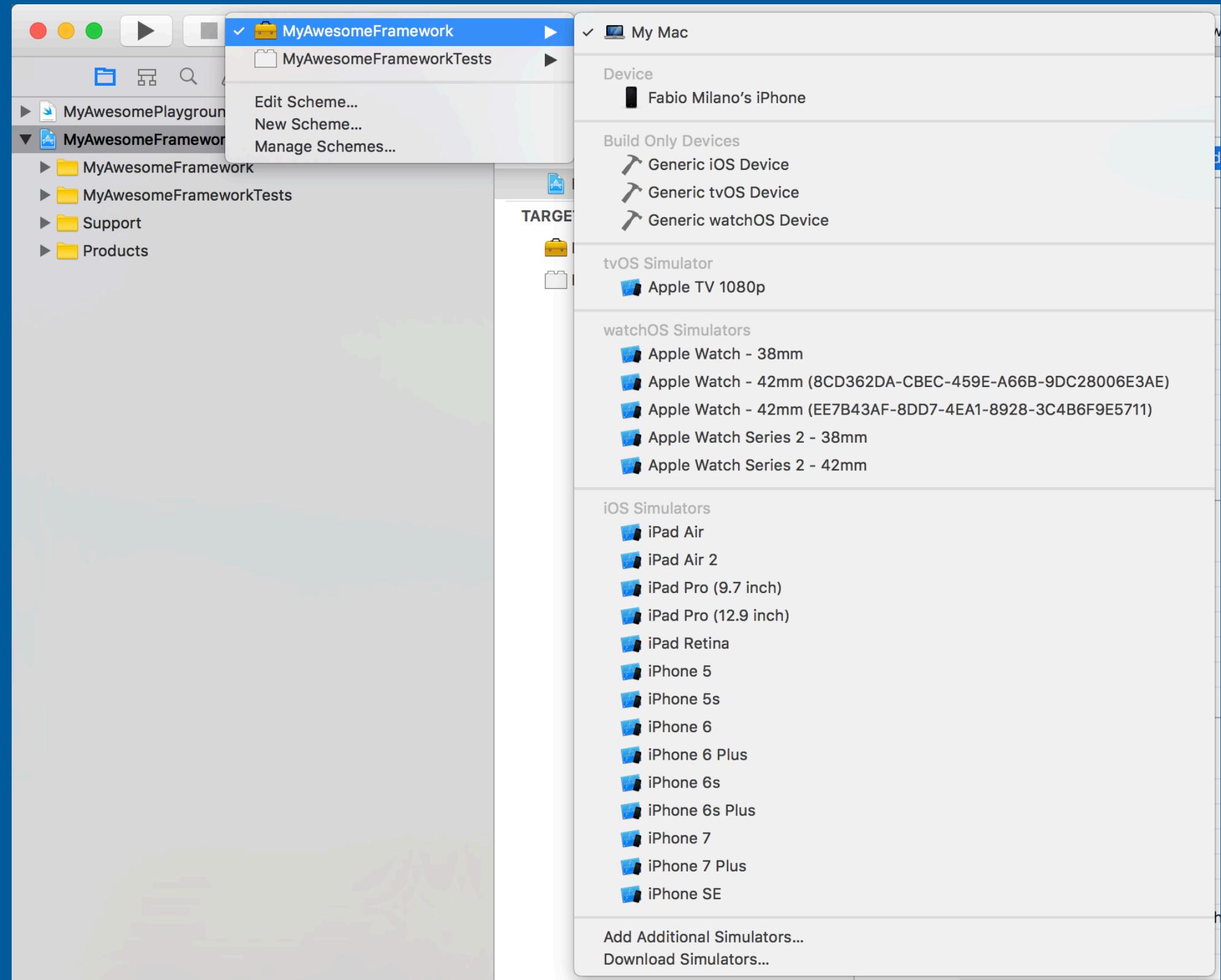


A screenshot of a Swift playground window titled "MyAwesomePlayground". The code in the playground is as follows:

```
1 //: Playground - noun: a place where people can play
2
3 import MyAwesomeFramework
4
5 if MyAwesomeStruct().awesome {
6     print("#yatusabes")
7 } else {
8     print("disappointing")
9 }
```

The output pane shows the result of running the code, which is "#yatusabes\n".





# DEPENDENCY MANAGERS

# COCOAPODS

cocoapods



Fabio Milano  
@iamfabiomilano

Finally here 🤘🤘 @CocoaPods #socks



# COCOAPODS - HOW

pod spec create MyAwesomeKit

# COCOAPODS - PODSPEC FILE

```
Pod::Spec.new do |spec|
  spec.name          = 'MyAwesomeKit'
  spec.version       = '0.1.0'
  spec.license        = { :type => 'MIT' }
  spec.homepage      = 'https://pragmaticconference.com'
  spec.authors       = { 'You' => 'you@me.com' }
  spec.summary        = 'Awesome code that works for your platform.'
  spec.source         = ....
  spec.source_files  = ....
  spec.framework     = 'SystemConfiguration'

  spec.ios.deployment_target = '10.0'
  spec.watchos.deployment_target = '3.0'
  spec.tvos.deployment_target = '10.0'
  spec.osx.deployment_target = '10.12'

  spec.dependency "Result", "~> 2.1"
end
```

# CARTHAGE



Autocreate schemes

Autocreate Schemes Now

Show Scheme

Container

Shared

- |                                     |                         |
|-------------------------------------|-------------------------|
| <input checked="" type="checkbox"/> | MyAwesomeFramework      |
| <input checked="" type="checkbox"/> | MyAwesomeFrameworkTests |

- |                                                                                                                |                                     |
|----------------------------------------------------------------------------------------------------------------|-------------------------------------|
|  MyAwesomeFramework project | <input checked="" type="checkbox"/> |
|  MyAwesomeFramework project | <input type="checkbox"/>            |



Edit...

Close

# CARTHAGE

```
# Cartfile

github "ReactiveCocoa/ReactiveCocoa" # GitHub.com
github "https://enterprise.local/ghe/desktop/git-error-translations" # GitHub Enterprise

git "https://enterprise.local/desktop/git-error-translations2.git"
```

# CARTHAGE

```
# Cartfile.private  
  
github "Quick/Quick"  
github "Quick/Nimble"
```

carthage update

# xcconfig FILE ..AND CARTHAGE

```
// Carthage.xcconfig

#include "MyAwesomeConfig.xcconfig"

FRAMEWORK_SEARCH_PATHS[sdk=macosx*] = $(SRCROOT)/Carthage/Build/Mac/ $(inherited)
FRAMEWORK_SEARCH_PATHS[sdk=iphone*] = $(SRCROOT)/Carthage/Build/iOS/ $(inherited)
FRAMEWORK_SEARCH_PATHS[sdk=watch*] = $(SRCROOT)/Carthage/Build/watchOS/ $(inherited)
FRAMEWORK_SEARCH_PATHS[sdk=appletv*] = $(SRCROOT)/Carthage/Build/tvOS/ $(inherited)
```

MyAwesomeFramework

Info Build Settings

PROJECT MyAwesomeFramework

TARGETS MyAwesomeFramework MyAwesomeFrameworkTests

Deployment Target

macOS Deployment Target 10.12

Configurations

Name	Based on Configuration File
Debug	1 Configuration Set
MyAwesomeFramework	Carthage ▾
MyAwesomeFramework	None ▾
MyAwesomeFrameworkTests	None ▾
Release	1 Configuration Set
MyAwesomeFramework	Carthage ▾
MyAwesomeFramework	None ▾
MyAwesomeFrameworkTests	None ▾

+

Use Release for command-line builds

# SWIFT PACKAGE MANAGER (SPM)

# SPM - THE PACKAGE FILE<sup>1</sup>

```
import PackageDescription

let package = Package(
    name: "SwiftBackend",
    dependencies: [
        .Package(url: "https://github.com/PerfectlySoft/Perfect-HTTPServer.git",
                 majorVersion: 2, minor: 0
        ),
        .Package(url: "https://github.com/PerfectlySoft/Perfect-PostgreSQL.git",
                 majorVersion: 2, minor: 0)
    ]
)
```

<sup>1</sup> EXAMPLE FROM PERFECT: SERVER-SIDE SWIFT PROJECT.

```
$ swift build
```

```
$ swift test
```

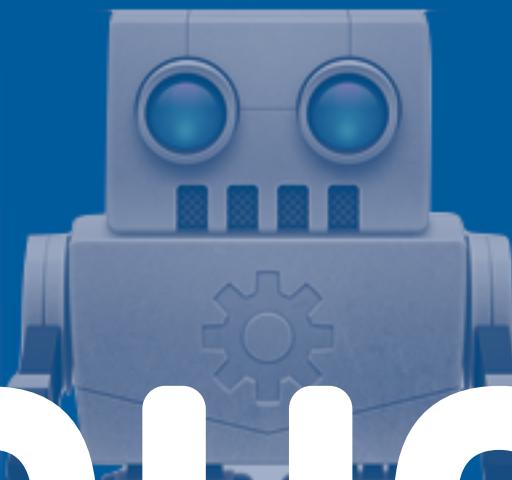
# SPM... AND XCODE

swift package generate-xcodeproj

**BEING A  
LIBRARY**

# CONTINUOUS INTEGRATION

Circleci



# CONTINUOUS INTEGRATION

› CATCHING PROBLEMS QUICKLY, EASILY AND EARLY



# CONTINUOUS INTEGRATION

- › CATCHING PROBLEMS QUICKLY, EASILY AND EARLY
  - › ENHANCING COLLABORATION



# CONTINUOUS INTEGRATION

- › CATCHING PROBLEMS QUICKLY, EASILY AND EARLY
  - › ENHANCING COLLABORATION
  - › BROADENING TEST COVERAGE

Circleci

# TRAVIS CI



```
# .travis.yml

language: objective-c
osx_image: xcode8

script:
- xcodebuild test -workspace MyAwesomeFramework.xcworkspace -scheme MyAwesomeFramework-Mac
- xcodebuild test -workspace MyAwesomeFramework.xcworkspace -scheme MyAwesomeFramework-iOS -sdk iphonesimulator
- xcodebuild test -workspace MyAwesomeFramework.xcworkspace -scheme MyAwesomeFramework-tvOS -sdk appletvsimulator
- xcodebuild build -workspace MyAwesomeFramework.xcworkspace -scheme MyAwesomeFramework-watchOS -sdk watchsimulator
```

```
# .travis.yml

language: objective-c
osx_image: xcode8

env:
- ACTION=test PLATFORM=Mac DESTINATION='platform=OS X'
- ACTION=test PLATFORM=iOS DESTINATION='platform=iOS Simulator,name=iPhone 6S'
- ACTION=build PLATFORM=watchOS DESTINATION='platform=watchOS Simulator,name=Apple Watch - 38mm'
- ACTION=test PLATFORM=tvOS DESTINATION='platform=tvOS Simulator,name=Apple TV 1080p'

script:
- set -o pipefail && xcodebuild -scheme MyAwesomeFramework -destination "$DESTINATION" $ACTION | xcpretty
```



**DANGER**

# DANGER FILE<sup>2</sup>

```
# Sometimes it's a README fix, or something like that - which isn't relevant for
# including in a project's CHANGELOG for example
not_declared_trivial = !(github.pr_title.include? "#trivial")
has_app_changes = !git.modified_files.grep(/Source/).empty?

# Make it more obvious that a PR is a work in progress and shouldn't be merged yet
warn("PR is classed as Work in Progress") if github.pr_title.include? "[WIP]"

# Warn when there is a big PR
warn("Big PR") if git.lines_of_code > 500

# Changelog entries are required for changes to library files.
no_changelog_entry = !git.modified_files.include?("Changelog.md")
if has_app_changes && no_changelog_entry && not_declared_trivial
  fail("Any changes to library code need a summary in the Changelog.")
end

# Added (or removed) library files need to be added (or removed) from the
# Carthage Xcode project to avoid breaking things for our Carthage users.
added_swift_library_files = git.added_files.grep(/Source.*\.swift/).empty?
deleted_swift_library_files = git.deleted_files.grep(/Source.*\.swift/).empty?
modified_carthage_xcode_project = !(git.deleted_files.grep(/Moya\.xcodeproj/).empty?)
if (added_swift_library_files || deleted_swift_library_files) && modified_carthage_xcode_project
  fail("Added or removed library files require the Carthage Xcode project to be updated.")
end
```

<sup>2</sup> EXAMPLE FROM [HTTPS://GITHUB.COM/MOYA/MOYA/BLOB/MASTER/DANGERFILE](https://github.com/Moya/Moya/blob/master/dangerfile)

# DANGER

DangerCI commented

+

	<b>1 Error</b>
	Please include a CHANGELOG entry. You can find it at <a href="#">CHANGELOG.md</a> .
	Please provide a summary in the Pull Request description

	<b>1 Warning</b>
	The file <a href="#">dangerfile_import_plugin.rb</a> does not pass <code>bundle exec danger plugins lint</code> . We want high coverage, as user documentation is auto-generated from it.

	<b>1 Message</b>
	@dangermcshane is not a member of the Danger organisation, would you like an invitation? It's optional, and is part of the <a href="#">Moya Community Continuity</a> .

Generated by [danger](#)



0 . 1 . 0

1 . 0 . 0

# RECAP

# RECAP

> FRAMEWORK OR NOT FRAMEWORK?

# RECAP

- > FRAMEWORK OR NOT FRAMEWORK?
- > PLATFORM MATTERS

# RECAP

- › FRAMEWORK OR NOT FRAMEWORK?
  - › PLATFORM MATTERS
- › AUTOMATE. AUTOMATE. AUTOMATE!

# RECAP

- › FRAMEWORK OR NOT FRAMEWORK?
  - › PLATFORM MATTERS
- › AUTOMATE. AUTOMATE. AUTOMATE!
  - › OPEN SOURCE

# THANK YOU!