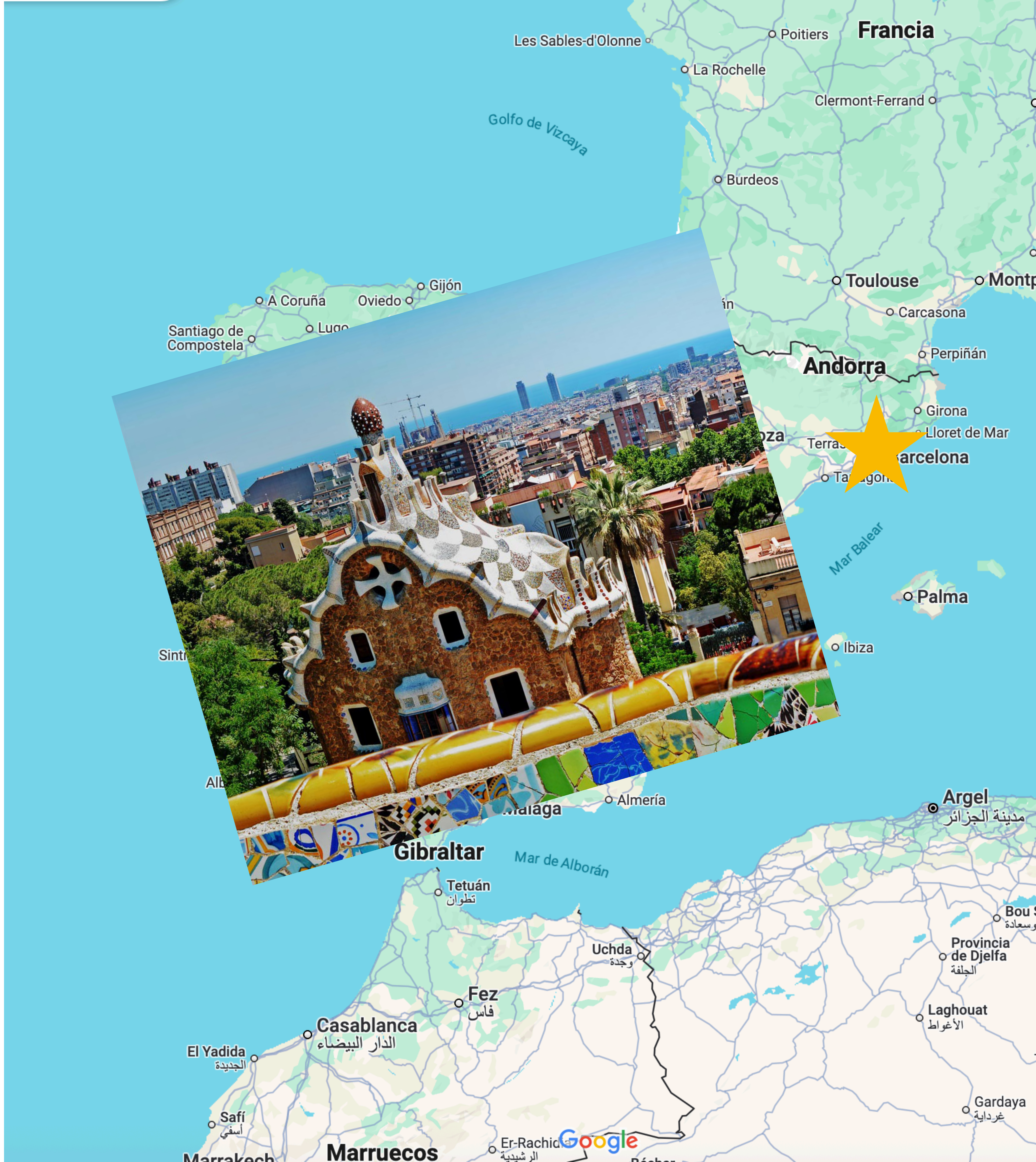
Bazel for iOS devs

About me

- David Collado
- @bitomule
- Tech Lead at Wallapop
- Indie when possible
- Born in Guadalajara, Spain
- Living in Barcelona



About Wallapop

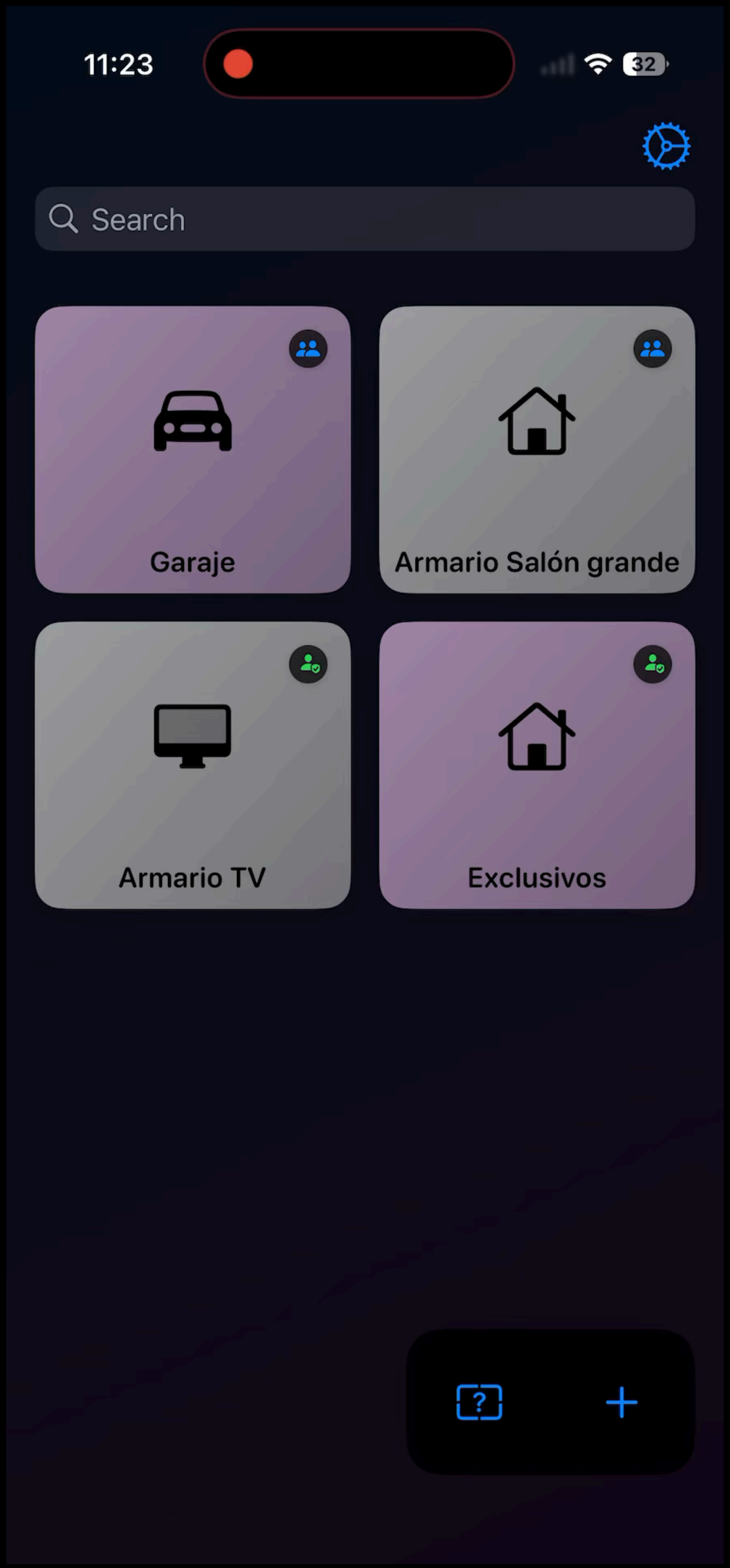
- Leading second hand marketplace in Spain
- > 400 employees
- 40 ~ mobile devs
- 300k lines (99% swift)

My apps

- Connect me to users and problem solving
- Always driven by self/close ones needs
- Have fun

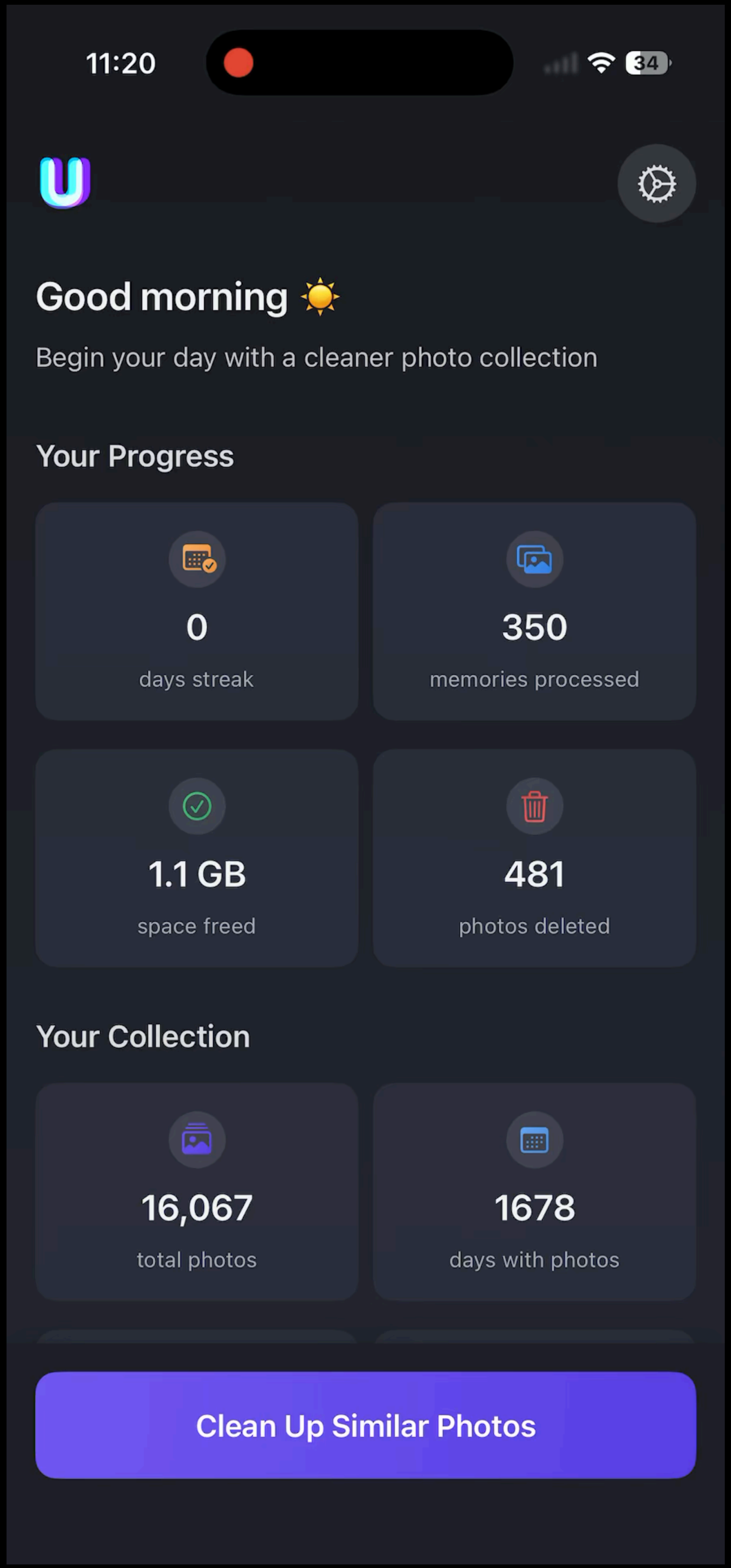
BOXY

- Organize your boxes while moving to a new place
- Give each box a code, find later
- Fully packed iOS citizen



UNDOLLY

- Clean your photo library from similar takes
- One set of photos at a time, insights, 100% on device



Good morning ☀

Begin your day with a cleaner photo collection

Your Progress

0 days streak

350 memories processed

1.1 GB space freed

481 photos deleted

Your Collection

16,067 total photos

1678 days with photos

Bazel for iOS devs

- What's Bazel
- Our history
- What you need to build an iOS App with Bazel
- Resources

What’s Bazel

Open-source build and test tool designed to build software quickly and reliably across multiple platforms and languages

When Bazel

- Builds times are impacting you heavily
- You have lot’s of modules
- You will benefit from more power over your build system

What you get

- Remote cache
- Local cache
- Dependency cache
- Test diffing
- Sandboxed builds
- Queries
- Rules
- SPEEEEED

Our history

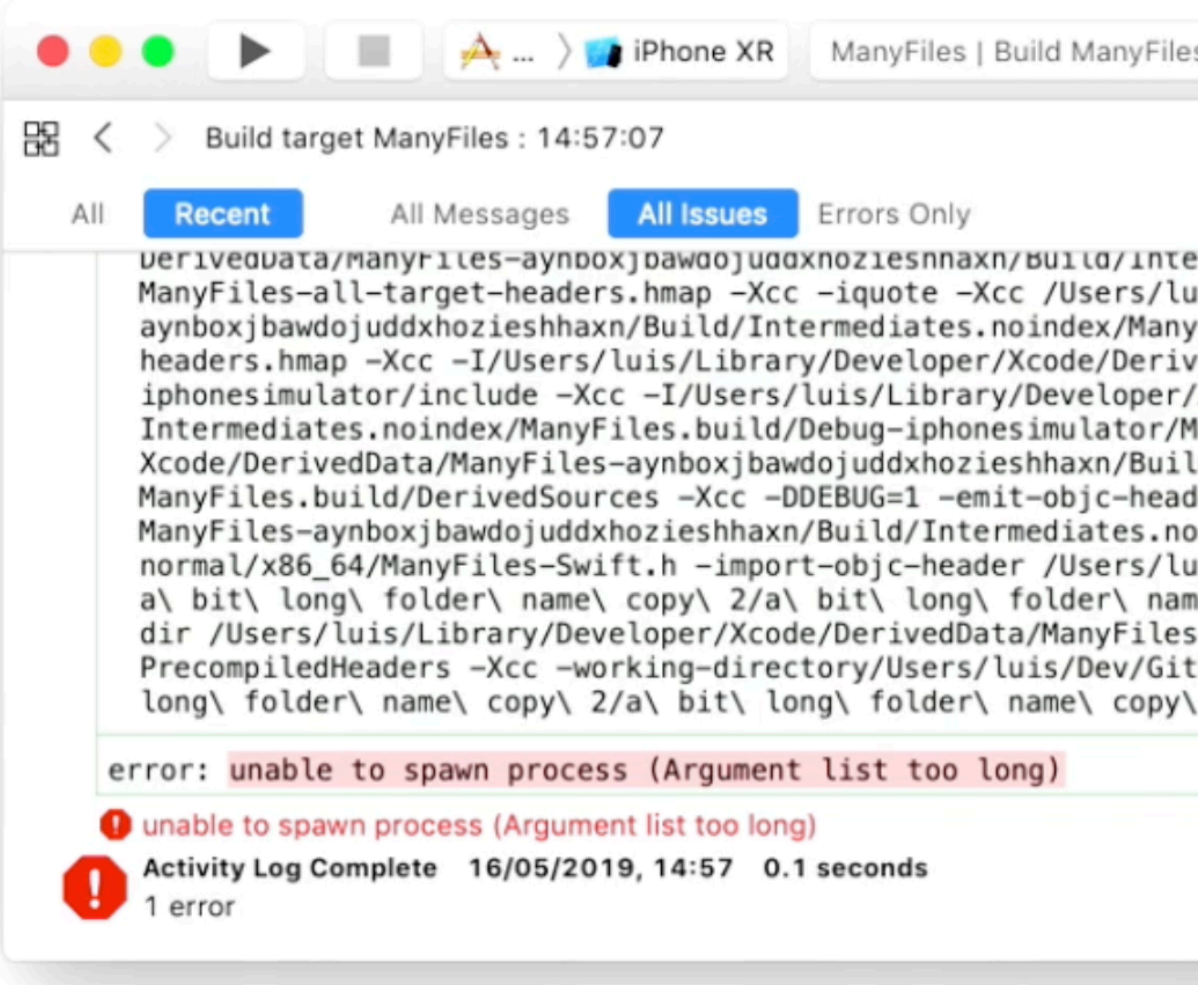
Let’s travel back in time

2017

- Monolith app
- Cocoapods for 3rd party dependencies

Xcode collapse

- Project collapsed because of an Xcode bug
- Hackiest ever solution



iPhone XR
ManyFiles | Build ManyFiles
Build target ManyFiles : 14:57:07
All
Recent
All Messages
All Issues
Errors Only

```
DerivedData/ManyFiles-aynboxjbawdojuddxhozieshhaxn/Build/Inte
ManyFiles-all-target-headers.hmap -Xcc -iquote -Xcc /Users/lu
aynboxjbawdojuddxhozieshhaxn/Build/Intermediates.noindex/Many
headers.hmap -Xcc -I/Users/luis/Library/Developer/Xcode/Deriv
iphonesimulator/include -Xcc -I/Users/luis/Library/Developer/
Intermediates.noindex/ManyFiles.build/Debug-iphonesimulator/M
Xcode/DerivedData/ManyFiles-aynboxjbawdojuddxhozieshhaxn/Buil
ManyFiles.build/DerivedSources -Xcc -DDEBUG=1 -emit-objc-head
ManyFiles-aynboxjbawdojuddxhozieshhaxn/Build/Intermediates.no
normal/x86_64/ManyFiles-Swift.h -import-objc-header /Users/lu
a\ bit\ long\ folder\ name\ copy\ 2/a\ bit\ long\ folder\ nam
dir /Users/luis/Library/Developer/Xcode/DerivedData/ManyFiles
PrecompiledHeaders -Xcc -working-directory/Users/luis/Dev/Git
long\ folder\ name\ copy\ 2/a\ bit\ long\ folder\ name\ copy\

error: unable to spawn process (Argument list too long)
```

unable to spawn process (Argument list too long)

Activity Log Complete **16/05/2019, 14:57** **0.1 seconds**
1 error

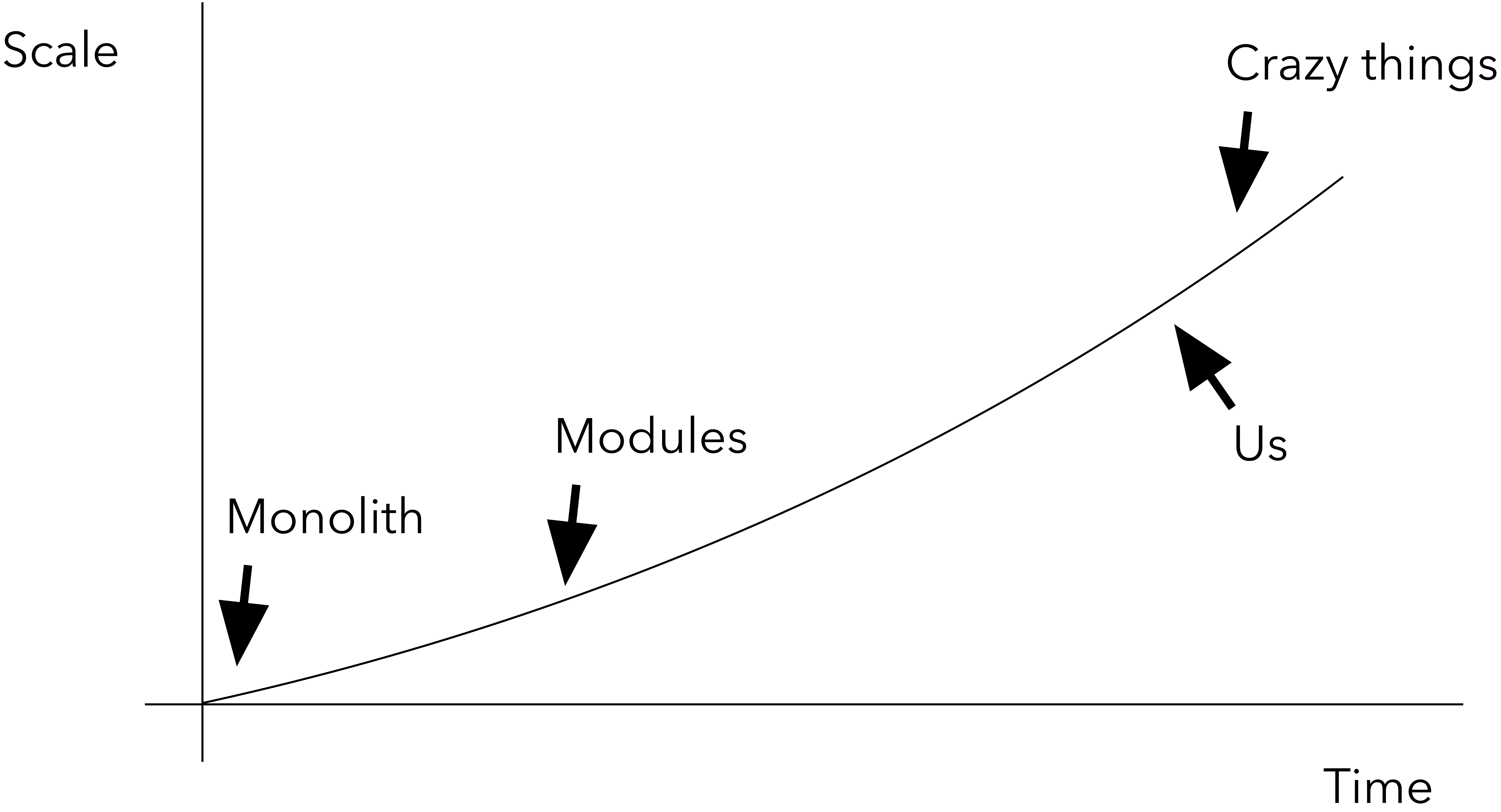
Rename
~/Projects/Wallapop
to ~/w

Spoiler:
Didn’t work for long

It enabled long term plan:
Cocoapods based modules

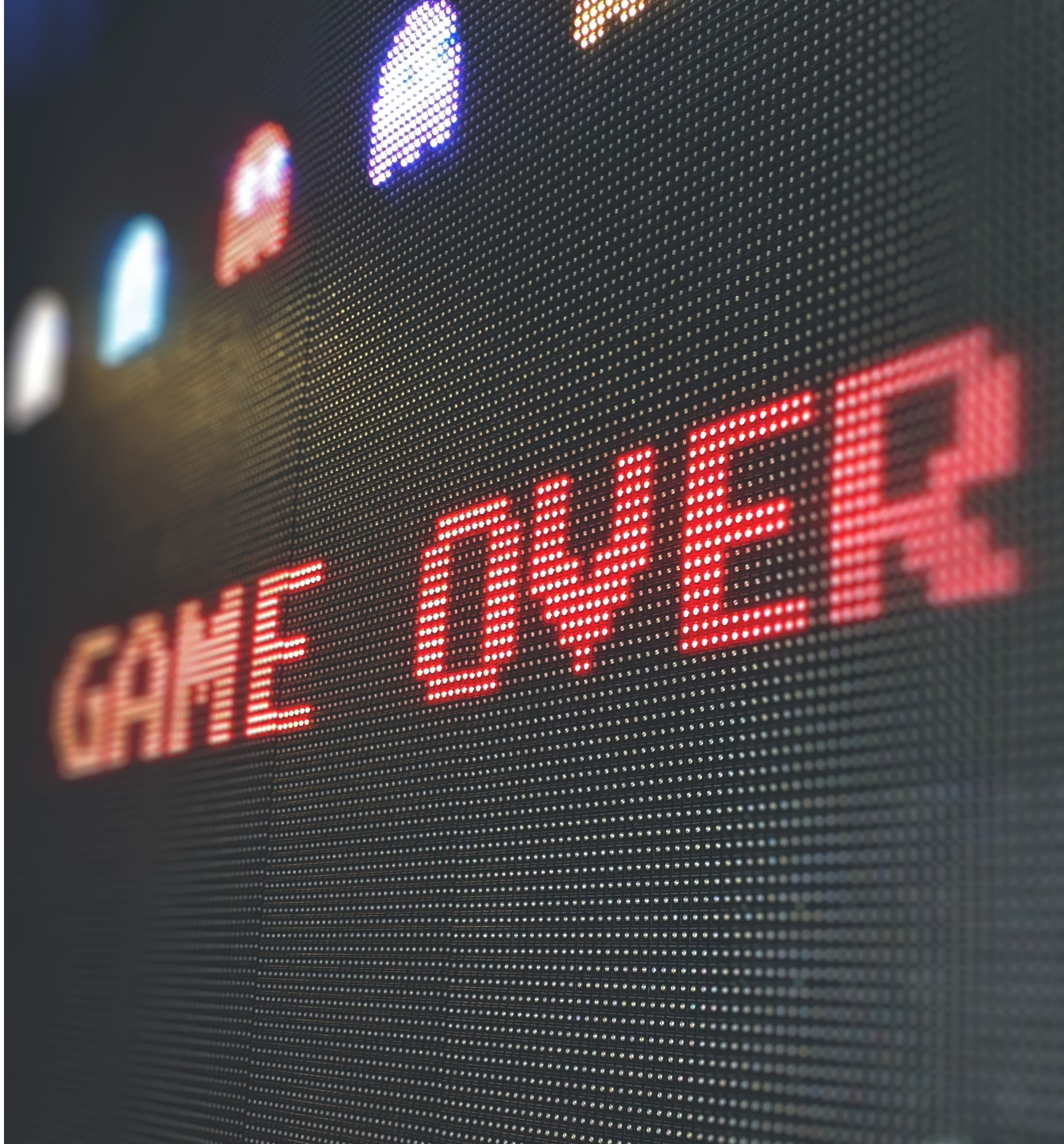
2024

- Proper modules architecture
- 60 modules
- Cocoapods based
- No SPM


Scale
Crazy things
Modules
Us
Monolith
Time

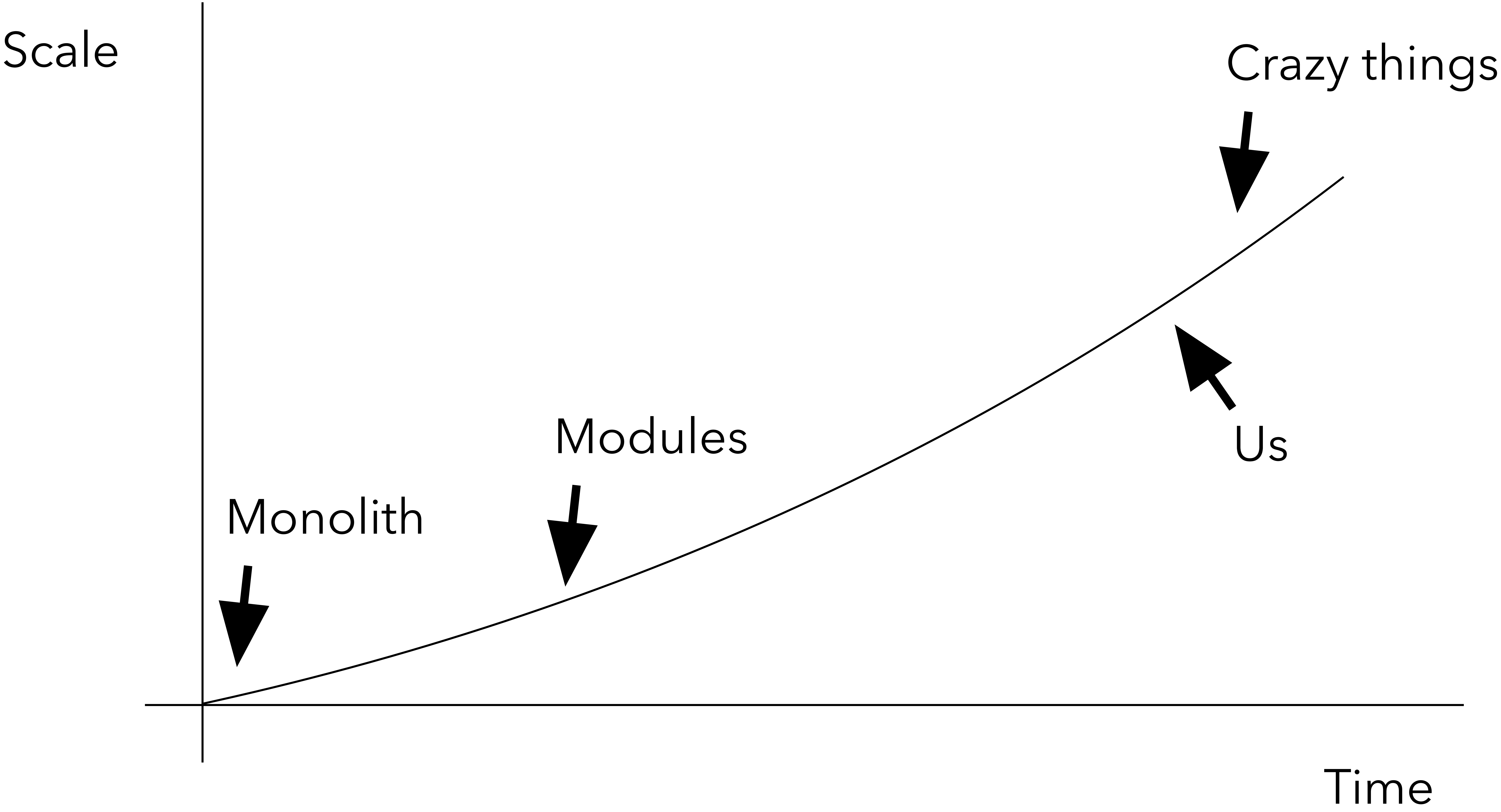
2025

- Proper modules architecture
- Almost 100 modules
- Cocoapods based
- Cocoapods dying



What now

- Almost 100 modules
- 20 iOS developers
- Dedicated platform team


Scale
Crazy things
Us
Modules
Monolith
Time

It may be the right time

The right moment

- Bazel for iOS is now far easier than it was 2 years ago
- Many companies jumping to the same solution
    - Airbnb, Lyft, Google, Spotify, Pinterest, Tinder…

Where we are today

- We have a full app POC
- We are learning and preparing for a full migration during 25/26
- We have data
    - CI time from 30 to 10 minutes
    - Local build times 7 to 3 minutes
- Just first draft

What’s next

- Build tooling around Bazel
- Migrate to Bazel using cocoapods for 3rd party dependencies
- Migrate 3rd party dependencies to SPM

Downsides

- Learning curve
- Maintenance overhead

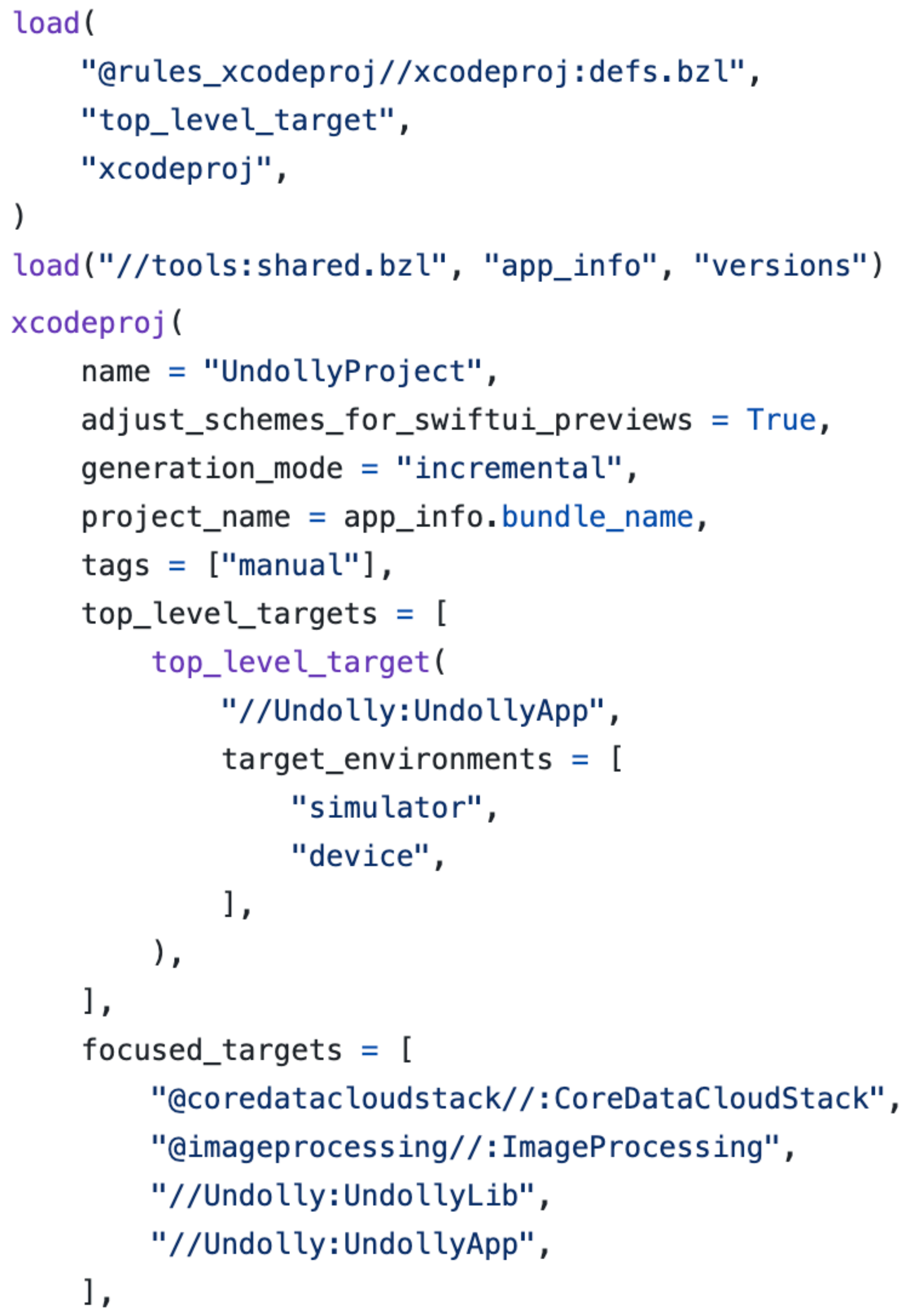
Building your first Bazel iOS app

RULES

- Building blocks
- Build and iOS App, create an Xcode project
- rules_swift, rules_apple, rules_xcodeproj

MODULE

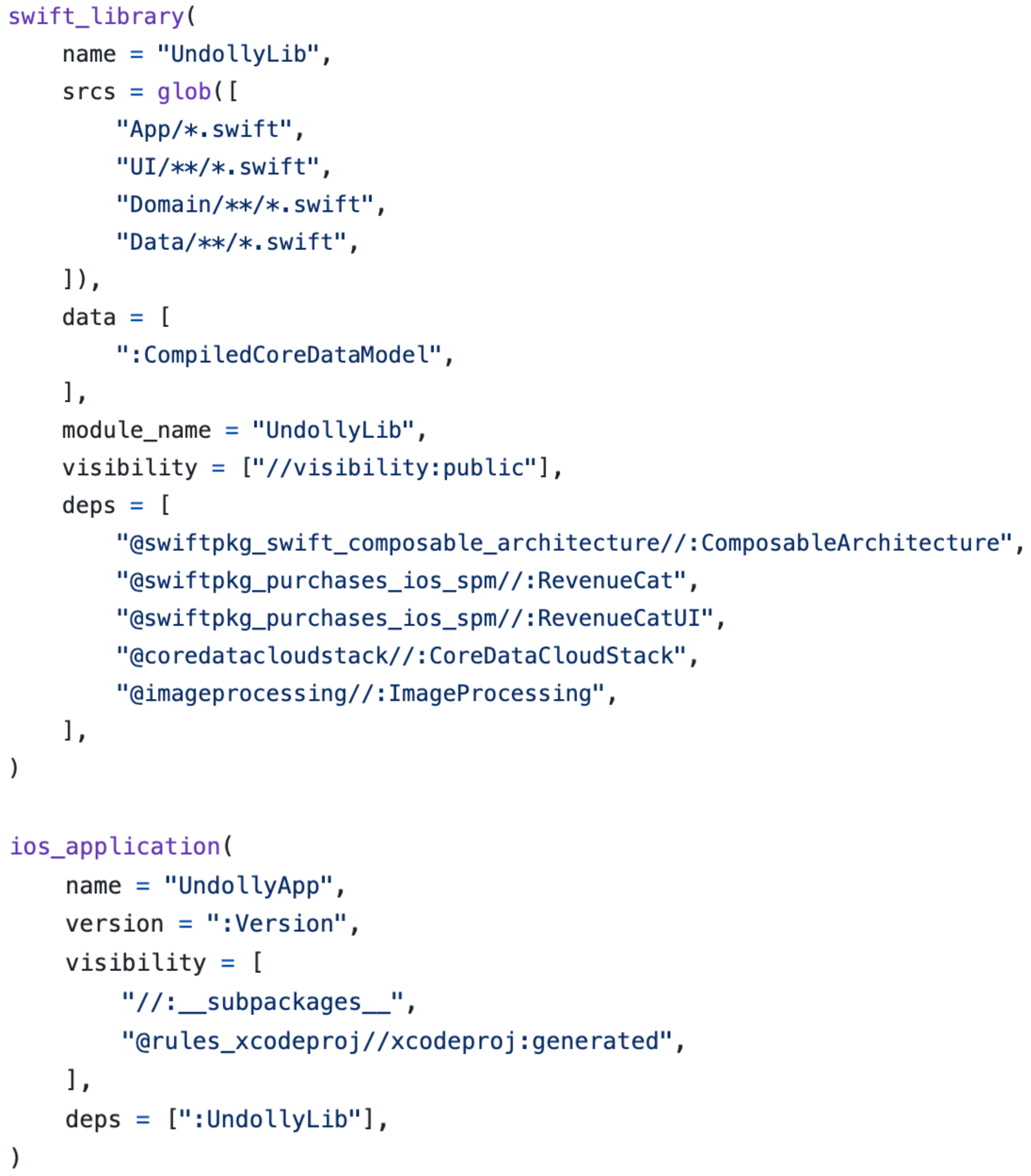
- File at project root
- Defines external dependencies
- Configures tools, versions, repositories…



```
load(
    "@rules_xcodeproj//xcodeproj:defs.bzl",
    "top_level_target",
    "xcodeproj",
)
load("//tools:shared.bzl", "app_info", "versions")
xcodeproj(
    name = "UndollyProject",
    adjust_schemes_for_swiftui_previews = True,
    generation_mode = "incremental",
    project_name = app_info.bundle_name,
    tags = ["manual"],
    top_level_targets = [
        top_level_target(
            "//Undolly:UndollyApp",
            target_environments = [
                "simulator",
                "device",
            ],
        ),
    ],
    focused_targets = [
        "@coredatacloudstack//:CoreDataCloudStack",
        "@imageprocessing//:ImageProcessing",
        "//Undolly:UndollyLib",
        "//Undolly:UndollyApp",
    ],
load(
    "@rules_xcodeproj//xcodeproj:defs.bzl",
    "top_level_target",
    "xcodeproj",
)
load("//tools:shared.bzl", "app_info", "versions")
xcodeproj(
    name = "UndollyProject",
    adjust_schemes_for_swiftui_previews = True,
    generation_mode = "incremental",
    project_name = app_info.bundle_name
```

BUILD

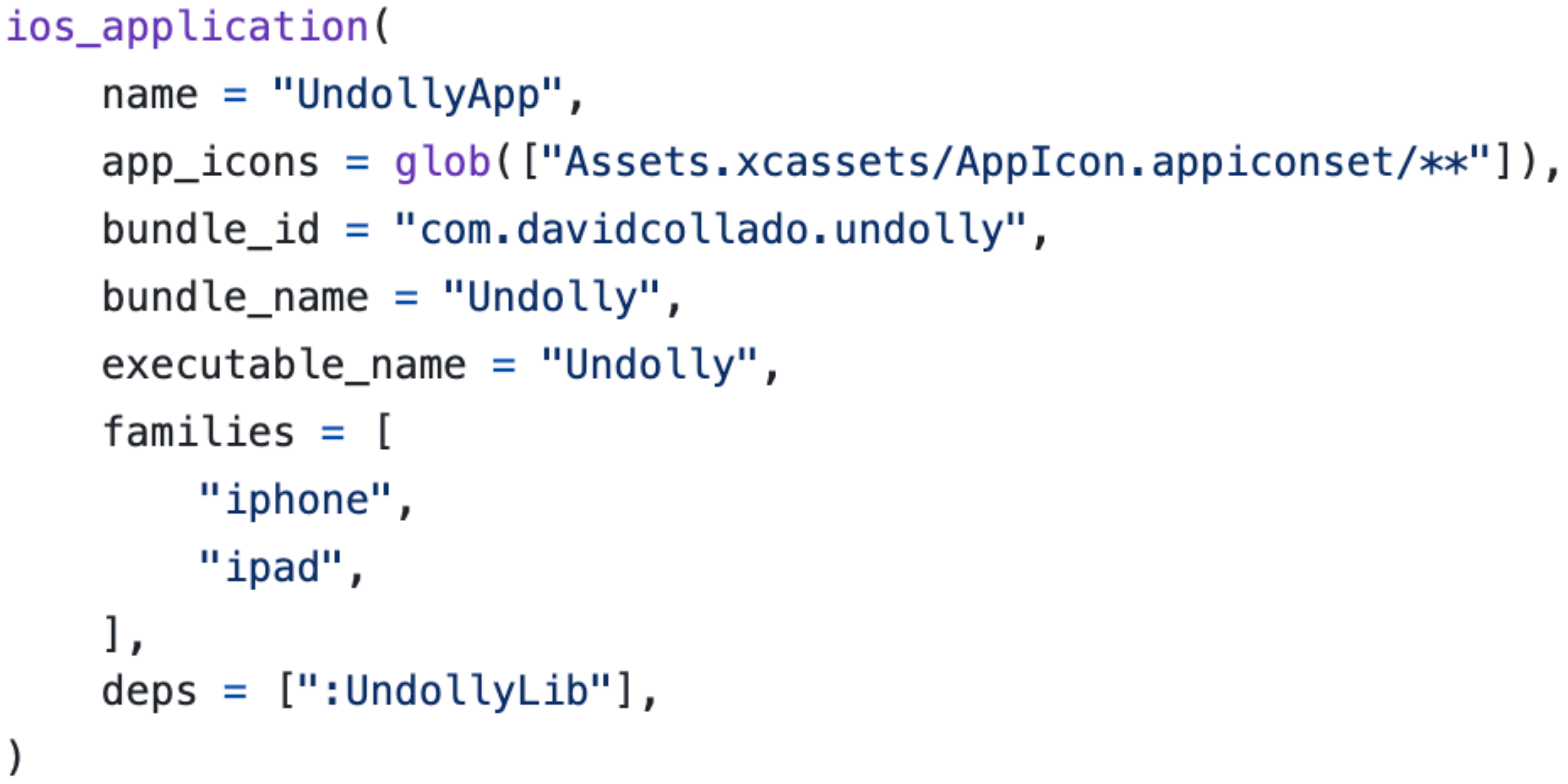
- File at module root
- Defines targets, sources, resources
- Defines dependencies between modules



```
swift_library(
    name = "UndollyLib",
    srcs = glob([
        "App/*.swift",
        "UI/**/*.swift",
        "Domain/**/*.swift",
        "Data/**/*.swift",
    ]),
    data = [
        ":CompiledCoreDataModel",
    ],
    module_name = "UndollyLib",
    visibility = ["//visibility:public"],
    deps = [
        "@swiftpkg_swift_composable_architecture//:ComposableArchitecture",
        "@swiftpkg_purchases_ios_spm//:RevenueCat",
        "@swiftpkg_purchases_ios_spm//:RevenueCatUI",
        "@coredatacloudstack//:CoreDataCloudStack",
        "@imageprocessing//:ImageProcessing",
    ],
)

ios_application(
    name = "UndollyApp",
    version = ":Version",
    visibility = [
        "//:__subpackages__",
        "@rules_xcodeproj//xcodeproj:generated",
    ],
    deps = [":UndollyLib"],
)
```

```
swift_library(
    name = "UndollyLib",
    srcs = glob([
        "App/*.swift",
        "UI/**/*.swift",
        "Domain/**/*.swift",
        "Data/**/*.swift",
    ]),
    deps = [
        "@swiftpkg_swift_composable_architecture//:ComposableArchitecture",
        "@swiftpkg_purchases_ios_spm//:RevenueCat",
        "@swiftpkg_purchases_ios_spm//:RevenueCatUI",
        "@coredatacloudstack//:CoreDataCloudStack",
        "@imageprocessing//:ImageProcessing",
    ],
)
```

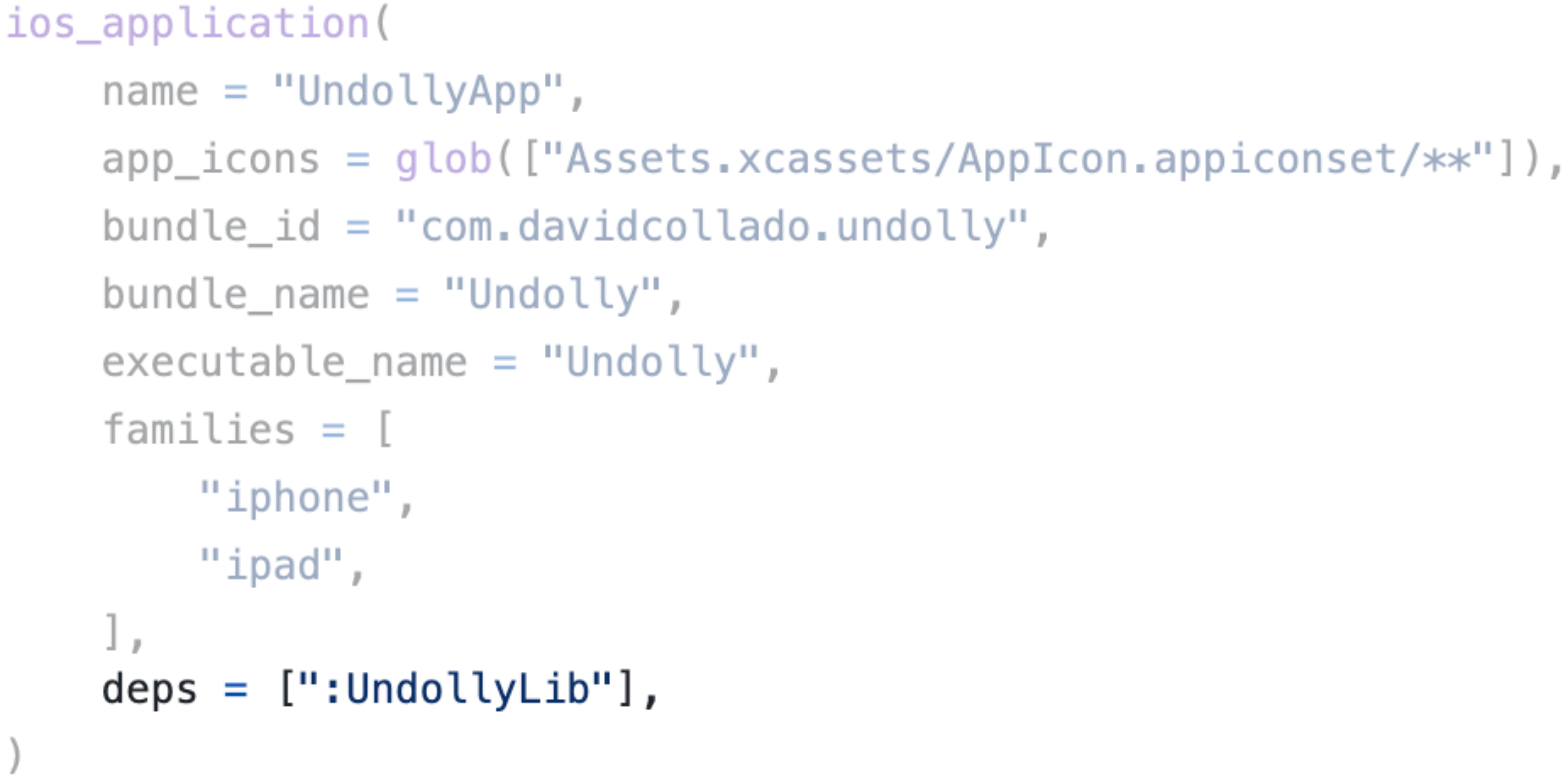


```
ios_application(
    name = "UndollyApp",
    app_icons = glob(["Assets.xcassets/AppIcon.appiconset/**"]),
    bundle_id = "com.davidcollado.undolly",
    bundle_name = "Undolly",
    executable_name = "Undolly",
    families = [
        "iphone",
        "ipad",
    ],
    deps = [":UndollyLib"],
)
```

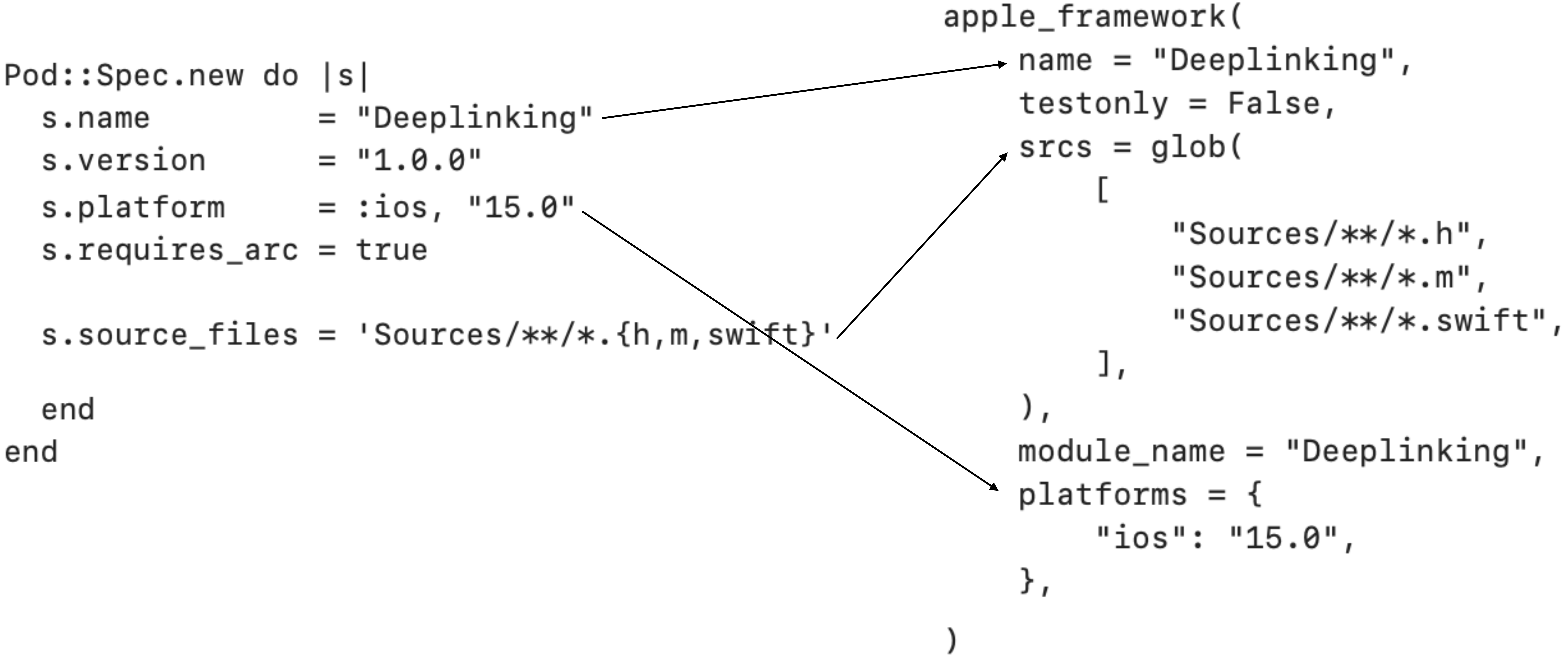
```
ios_application(
    name = "UndollyApp",
    app_icons = glob(["Assets.xcassets/AppIcon.appiconset/**"]),
    bundle_id = "com.davidcollado.undolly",
    bundle_name = "Undolly",
    executable_name = "Undolly",
    families = [
        "iphone",
        "ipad",
    ],
    deps = [":UndollyLib"],
)
```

```
ios_application(
    name = "UndollyApp",
    app_icons = glob(["Assets.xcassets/AppIcon.appiconset/**"]),
    bundle_id = "com.davidcollado.undolly",
    bundle_name = "Undolly",
    executable_name = "Undolly",
    families = [
        "iphone",
        "ipad",
    ],
    deps = [":UndollyLib"],
)
```

```
ios_application(
    name = "UndollyApp",
    app_icons = glob(["Assets.xcassets/AppIcon.appiconset/**"]),
    bundle_id = "com.davidcollado.undolly",
    bundle_name = "Undolly",
    executable_name = "Undolly",
    families = [
        "iphone",
        "ipad",
    ],
    deps = [":UndollyLib"],
)
```



```
ios_application(
    name = "UndollyApp",
    app_icons = glob(["Assets.xcassets/AppIcon.appiconset/**"]),
    bundle_id = "com.davidcollado.undolly",
    bundle_name = "Undolly",
    executable_name = "Undolly",
    families = [
        "iphone",
        "ipad",
    ],
    deps = [":UndollyLib"],
)
```



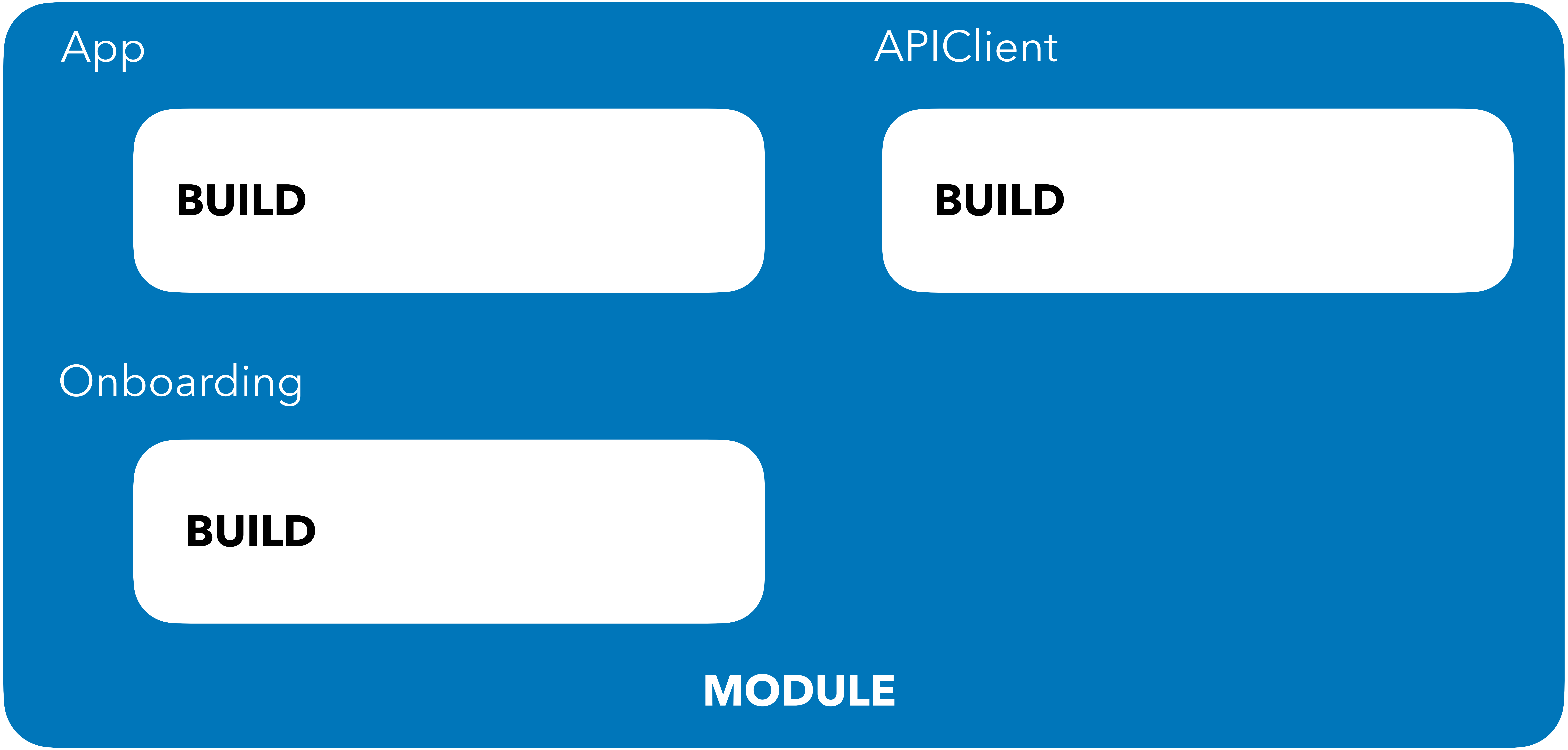
```
Pod::Spec.new do |s|
  s.name         = "Deeplinking"
  s.version      = "1.0.0"
  s.platform     = :ios, "15.0"
  s.requires_arc = true

  s.source_files = 'Sources/**/*.{h,m,swift}'

  end
end
```

```
apple_framework(
    name = "Deeplinking",
    testonly = False,
    srcs = glob(
        [
            "Sources/**/*.h",
            "Sources/**/*.m",
            "Sources/**/*.swift",
        ],
    ),
    module_name = "Deeplinking",
    platforms = {
        "ios": "15.0",
    },
)
```

Example App



What we learned

- Bazel offers speed, reproducibility, and scalability for your iOS builds
- Be cautious on adopting but playful
- iOS support has improved a lot and community may swift from Cocoapods to Bazel

Resources to get started

- bazel.build/install
- github.com/bazelbuild/rules_apple
- github.com/bazelbuild/rules_swift
- github.com/MobileNativeFoundation/rules_xcodeproj

THANK YOU

- You have Bazel experience? Let’s chat!
- Feel free to reach me @bitomule (x, bluesky, mastodon.social)