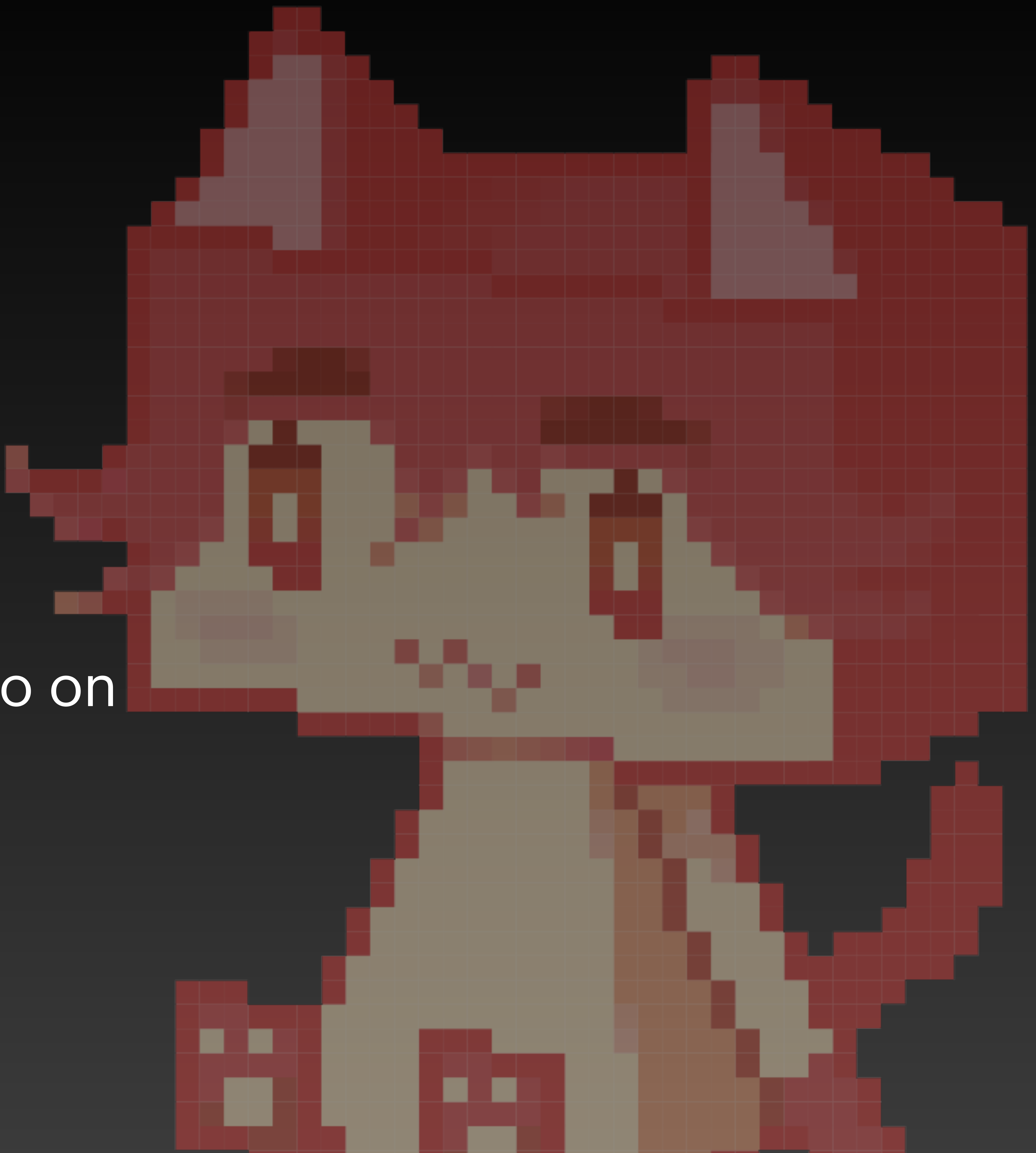
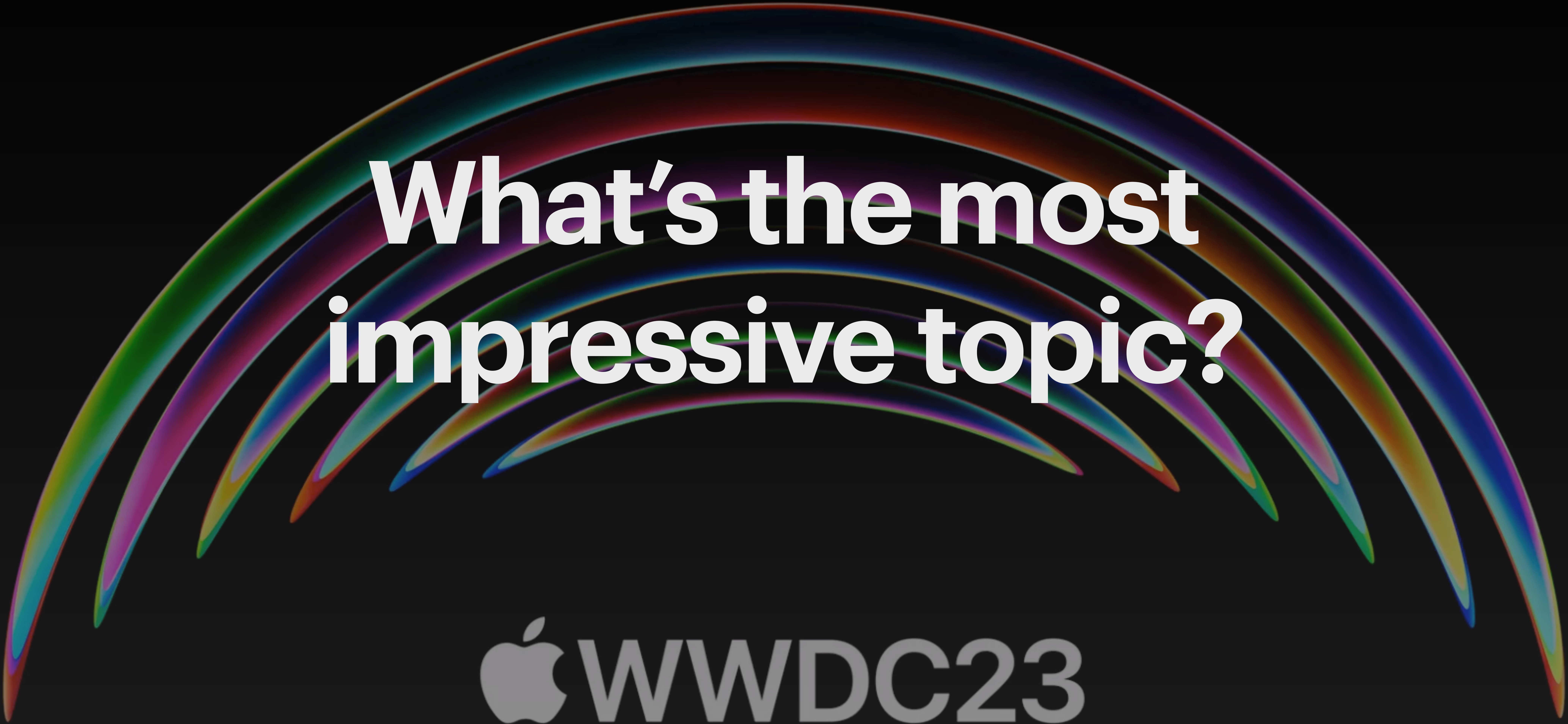
Hi!

- giginet (Twitter/GitHub)
- LINE Corp Senior Engineer
    - Developer Experience Team
- Tech Adviser for Money Forward
- Core Contributor
    - fastlane/Carthage/XcodeGen and so on
- 👾



Thank you for this great venue!!!

WWDC23



What’s the most impressive topic?

WWDC23

Mergeable Library

Deep dive into Mergeable Library

Tokyo iOS Meetup 2023/6/26

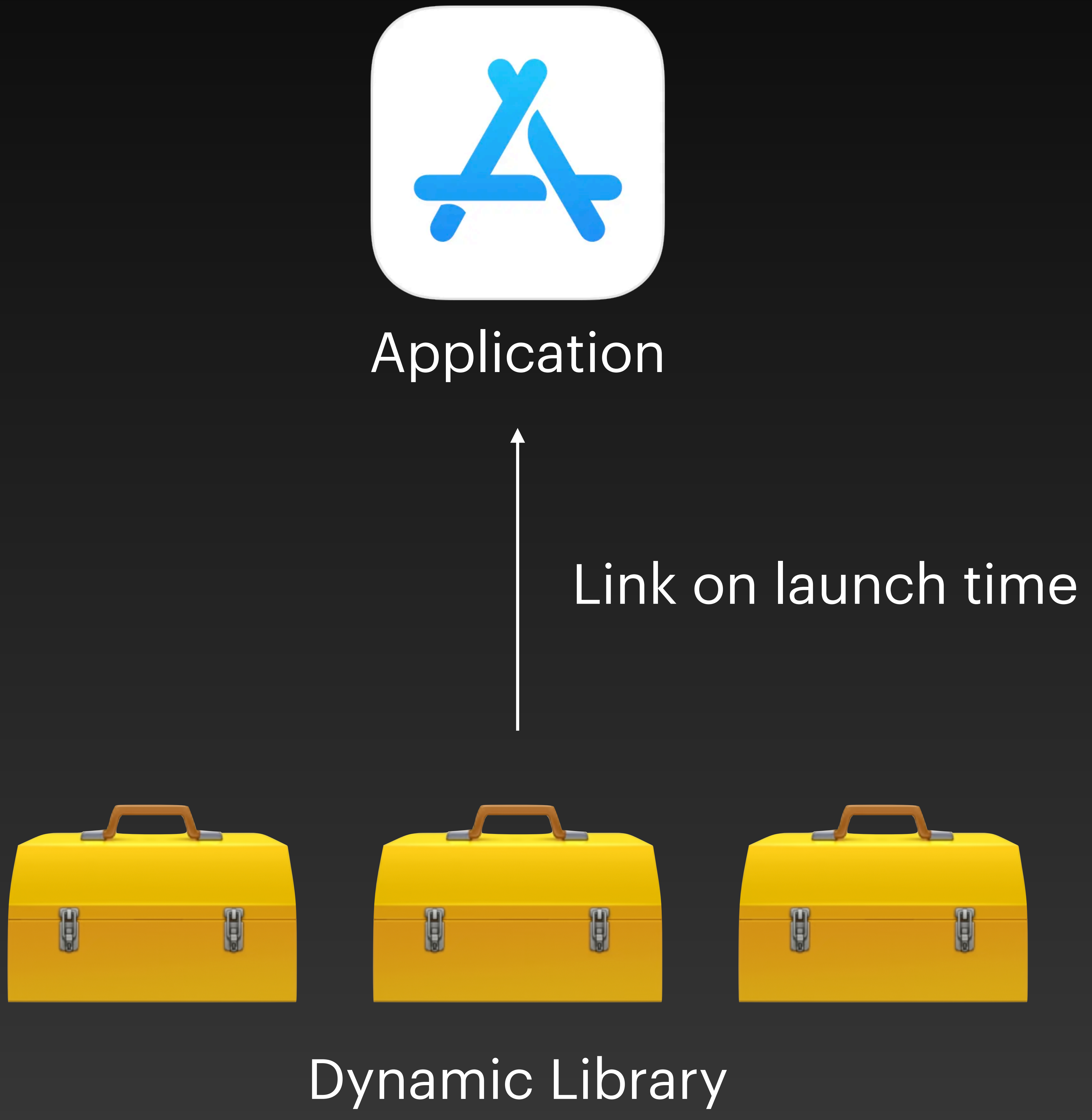
@giginet

In this talk

- Look back on linking type
- Learn about **Mergeable Library**
  - Meet Mergeable Library
    - https://developer.apple.com/videos/play/wwdc2023/10268/
  - Configuring your project to use mergeable libraries
    - https://developer.apple.com/documentation/Xcode/configuring-your-project-to-use-mergeable-libraries
- Look further at the behavior

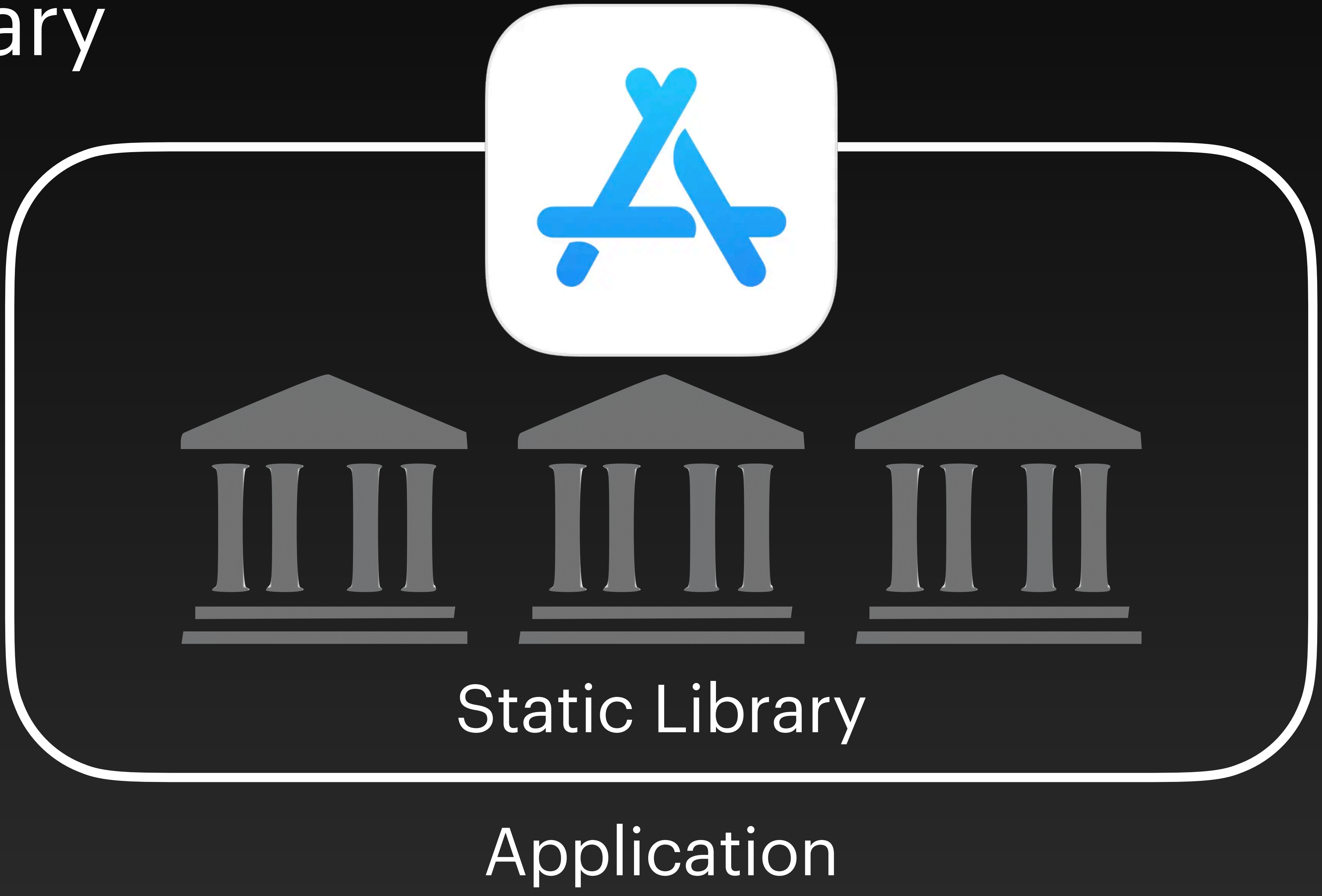
Dynamic linking

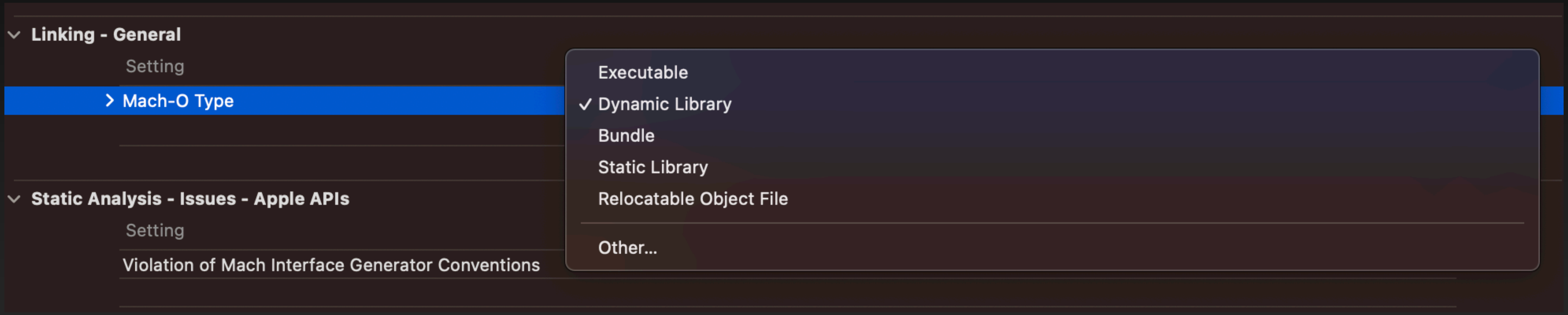
- Application links frameworks dynamically
- Link on **app launch time**
  - 🟢Pros
    - Reducing build duration
    - Work fine with Xcode Preview
  - ❌Cons
    - Overhead of launch time
    - Increasing app size



Static linking

- Embed symbols into an application binary
  - Link on **build time**
  - 🟢Pros
    - No overhead on launch time
    - Easy to reduce app size
  - ❌Cons
    - Build duration is longer
    - Some IDE features don’t work(Xcode Preview)





Linking - General

Setting

Mach-O Type

Static Analysis - Issues - Apple APIs

Setting

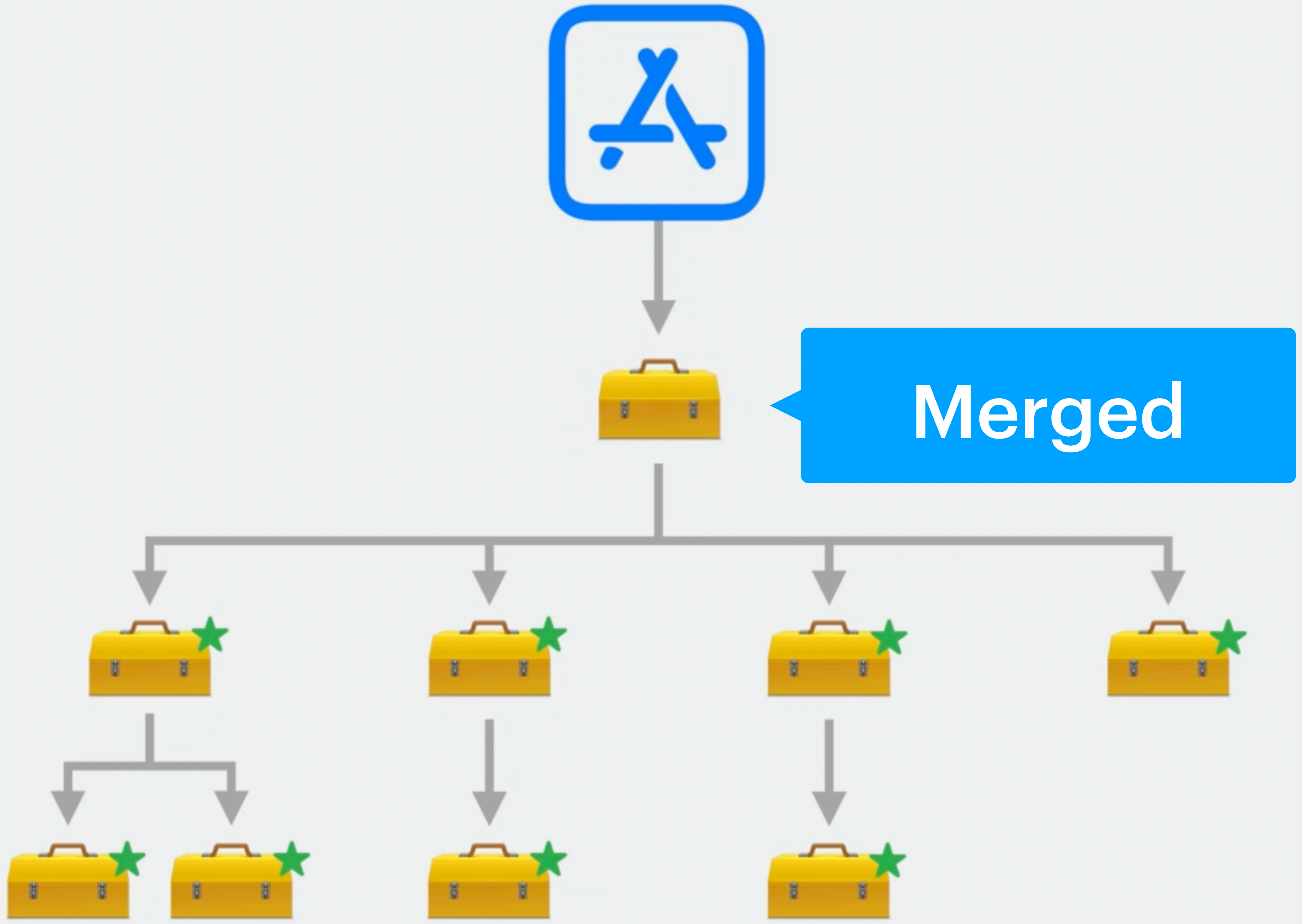
Violation of Mach Interface Generator Conventions

In the past

- The dilemma between Dynamic Linking VS Static Linking
  - For developers, Dynamic Linking is better
    - Faster build duration
    - IDE features work fine
  - For end-users, Static Linking is better
    - Smaller app binary size
    - Faster launch time

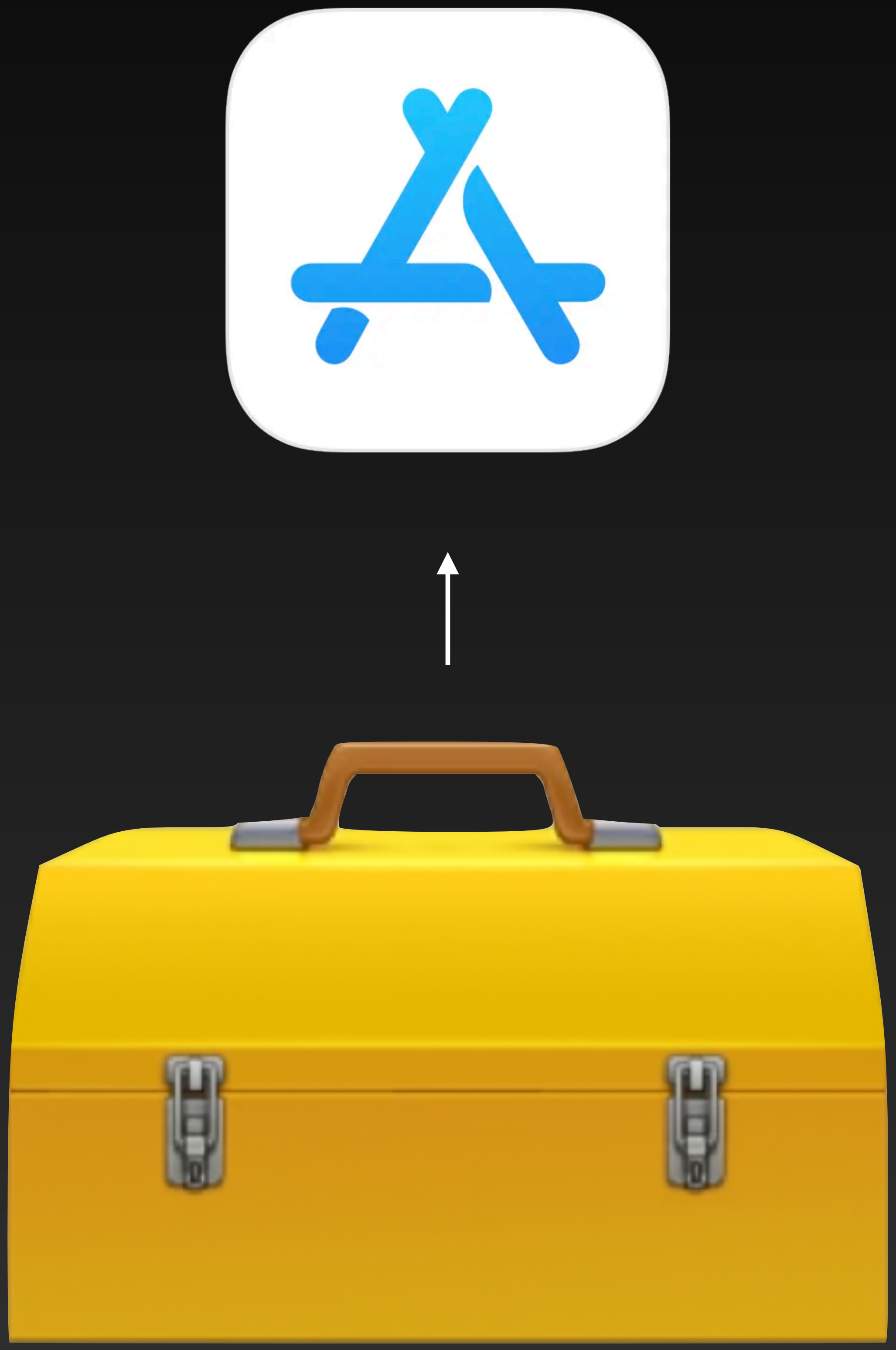
Mergeable Library

- New build system feature with Xcode 15
  - Merge dependencies into one library
    - Reducing linking overhead
  - Switch framework types by a build configuration automatically
    - Debug: Dynamic Linking
    - Release: Static Linking



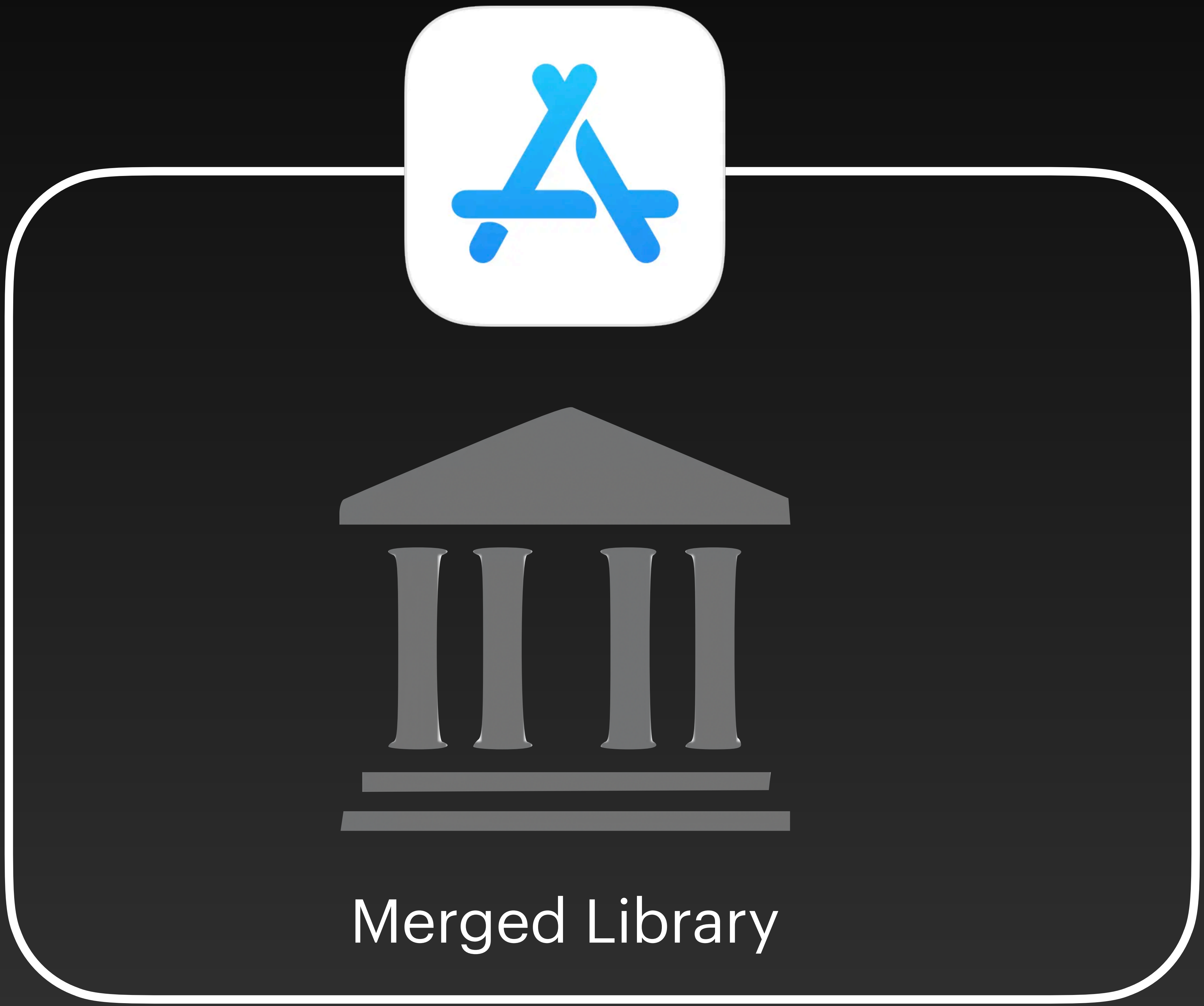
Merged

Debug



Merged Library

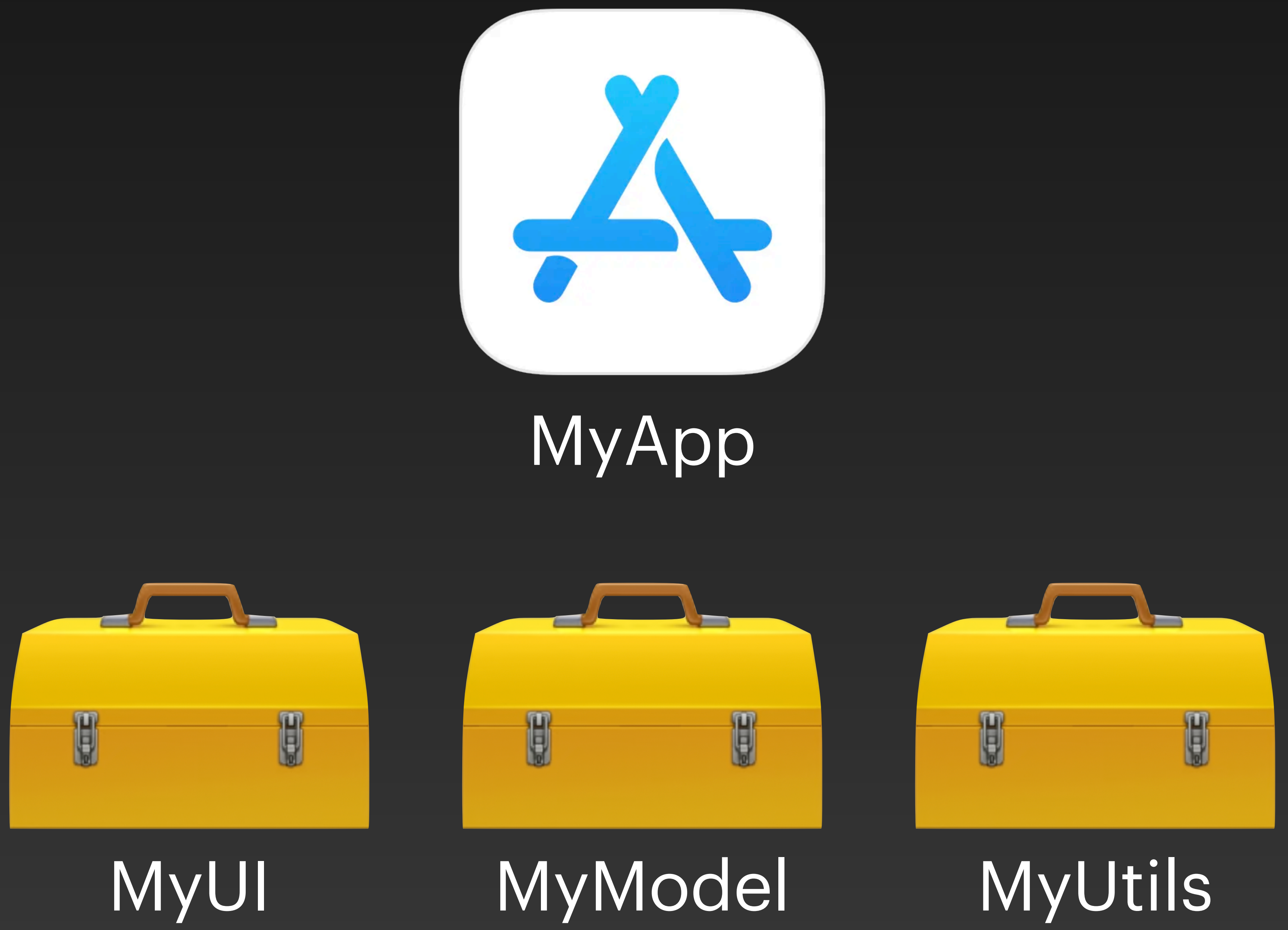
Release

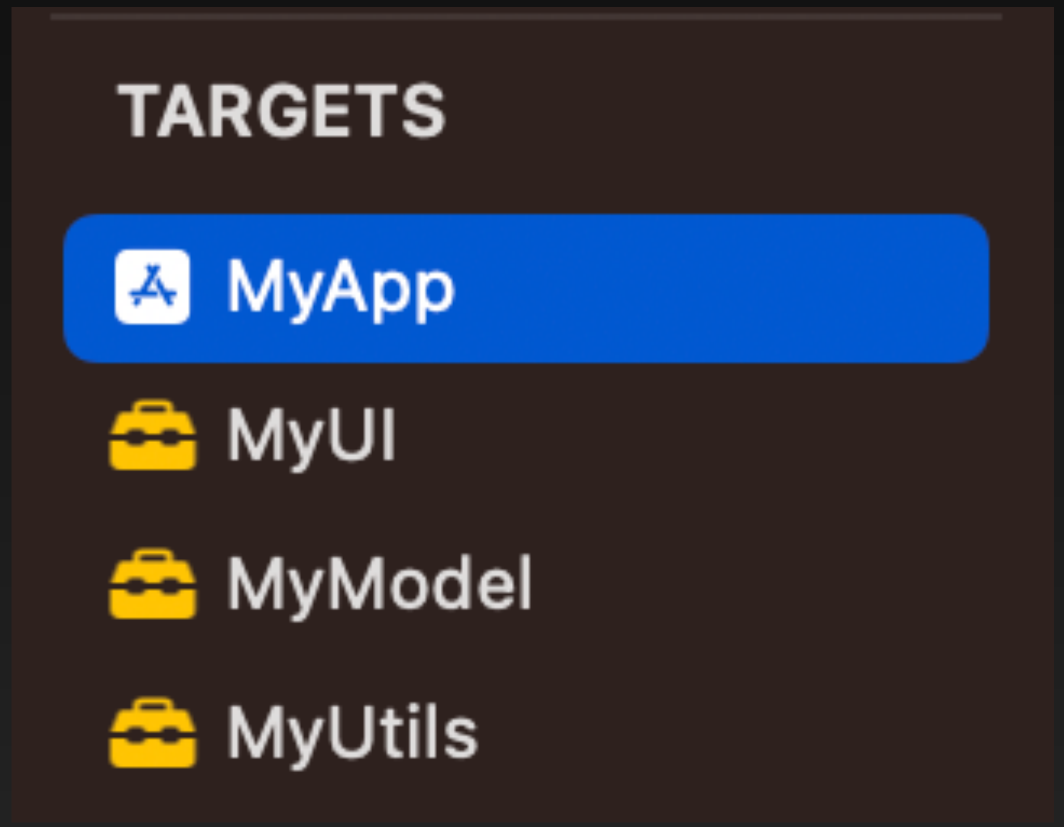


Merged Library

Sample Project

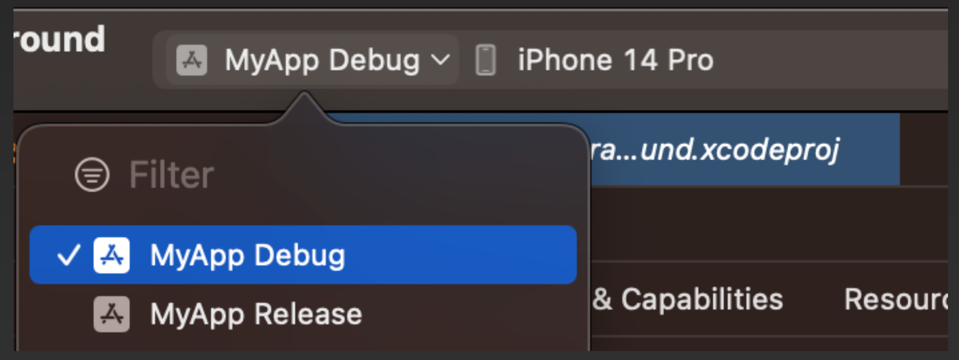
- Tried on Xcode 15 beta 2
- Application with three libraries





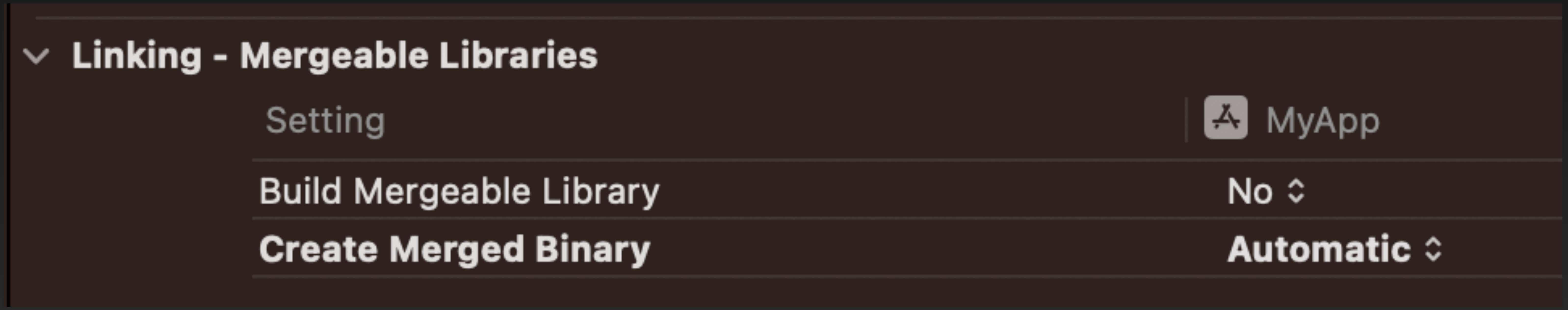
TARGETS

- MyApp
- MyUI
- MyModel
- MyUtils



Enabling Mergeable Library

- Just set "***Create Merged Binary***" of the application target(MyApp)
- Linking > Mergeable Libraries > Create Merged Binary > **Automatic**

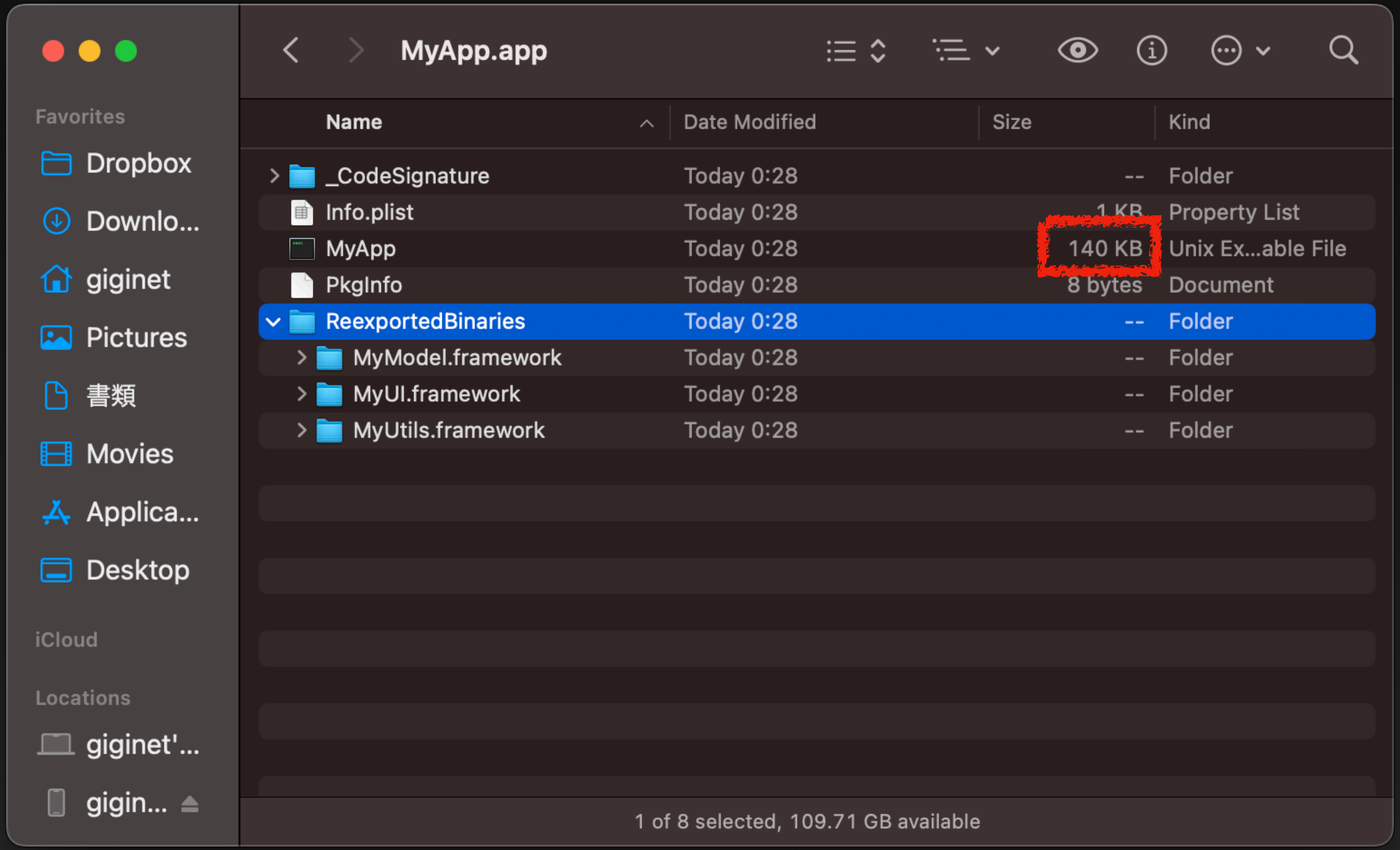


Linking - Mergeable Libraries

Setting	MyApp
Build Mergeable Library	No ⌃⌄
Create Merged Binary	**Automatic** ⌃⌄

Debug Configuration

- All merged frameworks are re-exported into the app bundle
    - *MyApp.app/ReexportedBinaries/*.framework*
    - All frameworks are dynamically linked on development

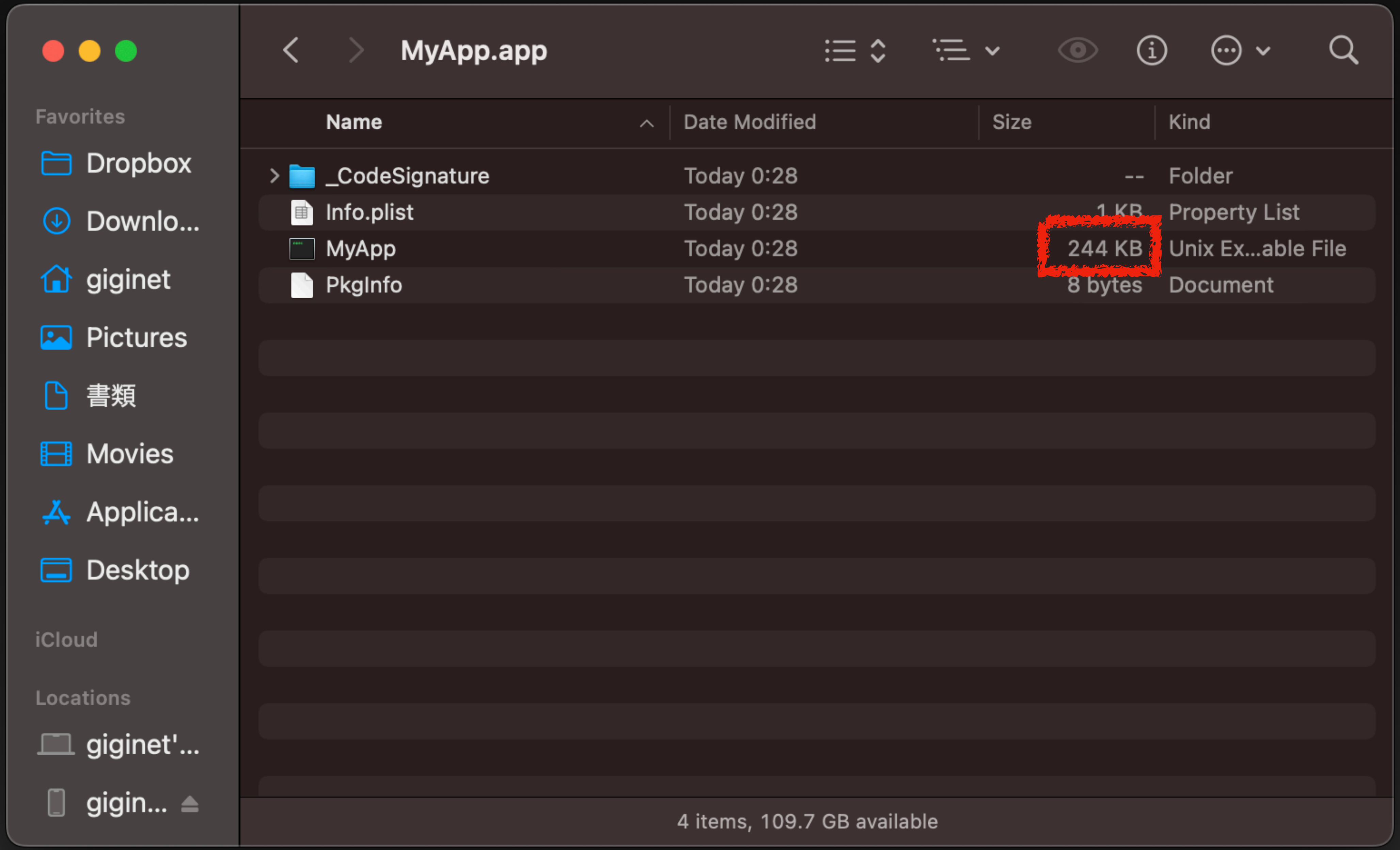


Name	Date Modified	Size	Kind
_CodeSignature	Today 0:28	--	Folder
Info.plist	Today 0:28	1 KB	Property List
MyApp	Today 0:28	140 KB	Unix Ex...able File
PkgInfo	Today 0:28	8 bytes	Document
ReexportedBinaries	Today 0:28	--	Folder
MyModel.framework	Today 0:28	--	Folder
MyUI.framework	Today 0:28	--	Folder
MyUtils.framework	Today 0:28	--	Folder

1 of 8 selected, 109.71 GB available

Release Configuration

- All frameworks are embedded into app binary
  - Binary size is increased than dynamically one



Name	Date Modified	Size	Kind
_CodeSignature	Today 0:28	--	Folder
Info.plist	Today 0:28	1 KB	Property List
MyApp	Today 0:28	244 KB	Unix Ex...able File
PkgInfo	Today 0:28	8 bytes	Document

4 items, 109.7 GB available

How to embed symbols

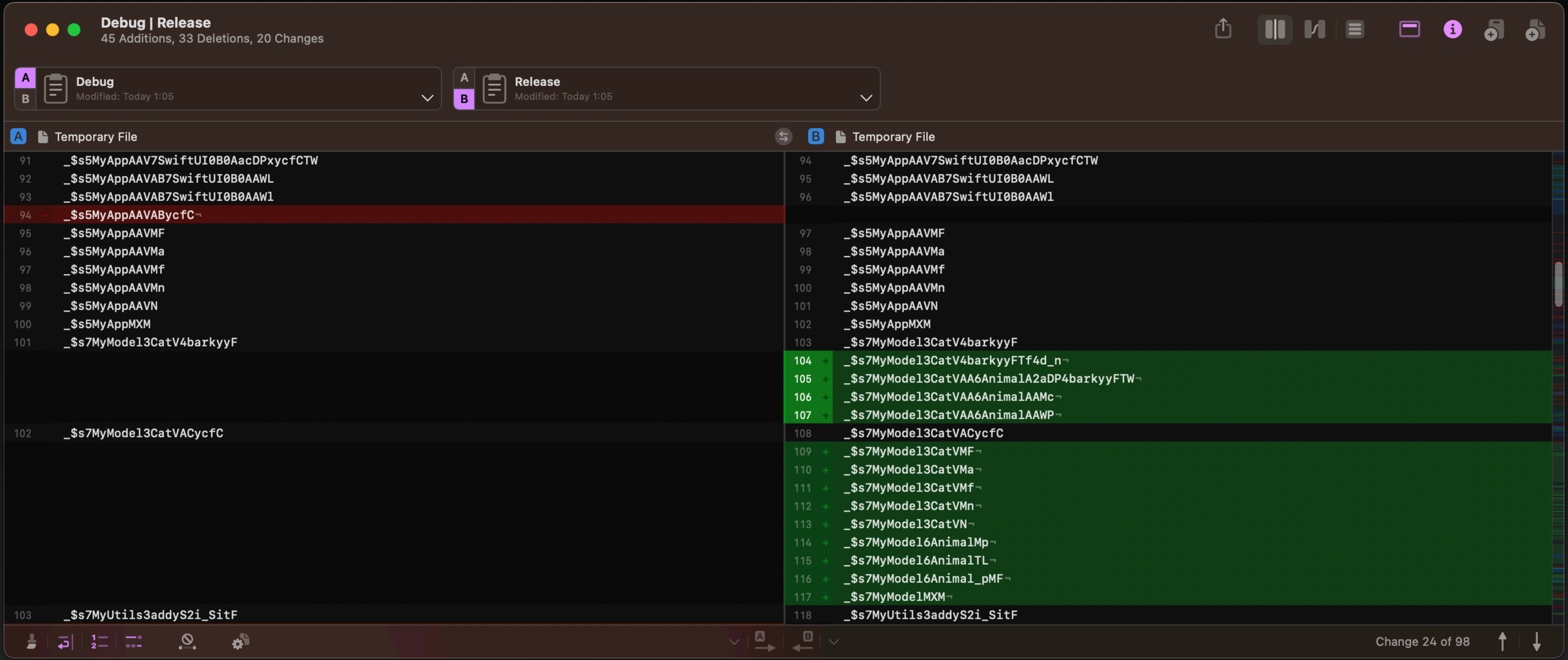
- Example : A struct in **MyModel** framework

```
import Foundation

public protocol Animal {
    func bark()
}

public struct Cat: Animal {
    public init() {
    }

    public func bark() {
        print("meow")
    }
}
```

Dump app binaries

- Dump all symbols in app binary
  - Using **nm** command

```
nm --format=just-symbols --size-sort Release-iphonesimulator/MyApp.app/MyApp
```



Debug | Release
45 Additions, 33 Deletions, 20 Changes

```
  _$s5MyAppAAV7SwiftUI0B0AacDPxycfCTW
  _$s5MyAppAAVAB7SwiftUI0B0AAWL
  _$s5MyAppAAVAB7SwiftUI0B0AAWl
- _$s5MyAppAAVABycfC
  _$s5MyAppAAVMF
  _$s5MyAppAAVMa
  _$s5MyAppAAVMf
  _$s5MyAppAAVMn
  _$s5MyAppAAVN
  _$s5MyAppMXM
  _$s7MyModel3CatV4barkyyF
  _$s7MyModel3CatVACycfC
  _$s7MyUtils3addyS2i_SitF
```

```
  _$s5MyAppAAV7SwiftUI0B0AacDPxycfCTW
  _$s5MyAppAAVAB7SwiftUI0B0AAWL
  _$s5MyAppAAVAB7SwiftUI0B0AAWl
  _$s5MyAppAAVMF
  _$s5MyAppAAVMa
  _$s5MyAppAAVMf
  _$s5MyAppAAVMn
  _$s5MyAppAAVN
  _$s5MyAppMXM
  _$s7MyModel3CatV4barkyyF
+ _$s7MyModel3CatV4barkyyFTf4d_n
+ _$s7MyModel3CatVAA6AnimalA2aDP4barkyyFTW
+ _$s7MyModel3CatVAA6AnimalAAMc
+ _$s7MyModel3CatVAA6AnimalAAWP
  _$s7MyModel3CatVACycfC
+ _$s7MyModel3CatVMF
+ _$s7MyModel3CatVMa
+ _$s7MyModel3CatVMf
+ _$s7MyModel3CatVMn
+ _$s7MyModel3CatVN
+ _$s7MyModel6AnimalMp
+ _$s7MyModel6AnimalTL
+ _$s7MyModel6Animal_pMF
+ _$s7MyModelMXM
  _$s7MyUtils3addyS2i_SitF
```

Debug(140KB)

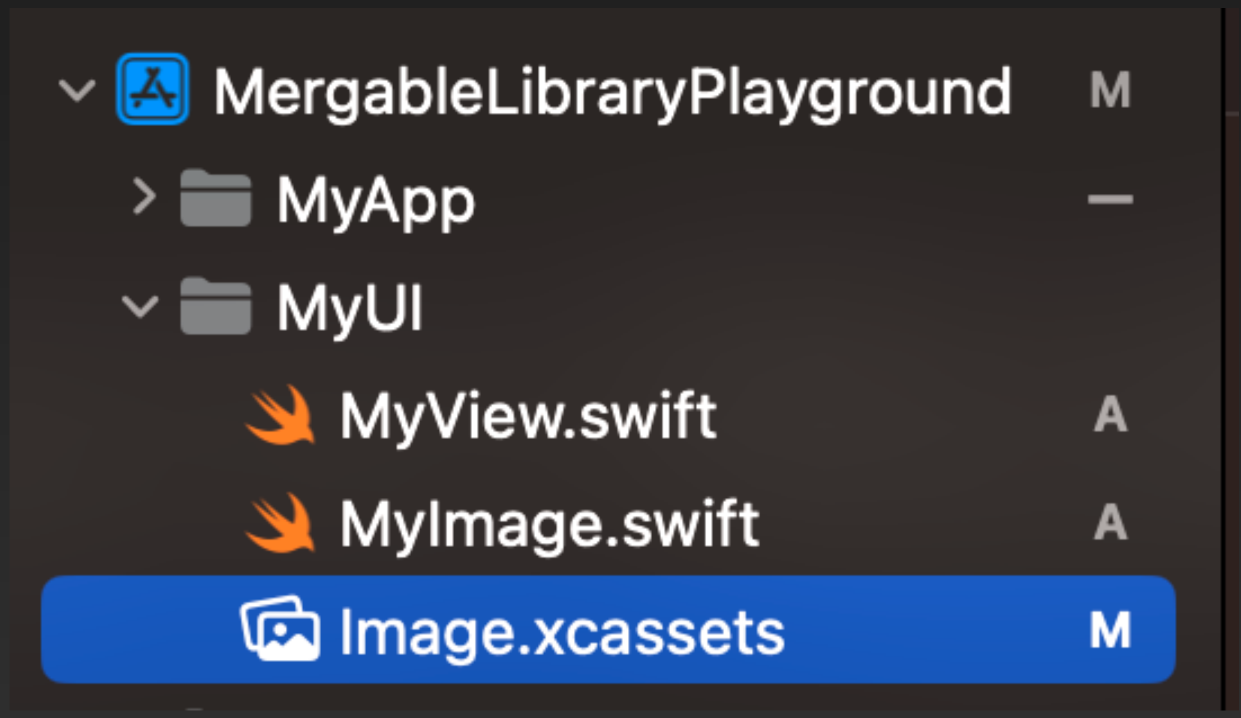
Release(244KB)

Resource bundle support

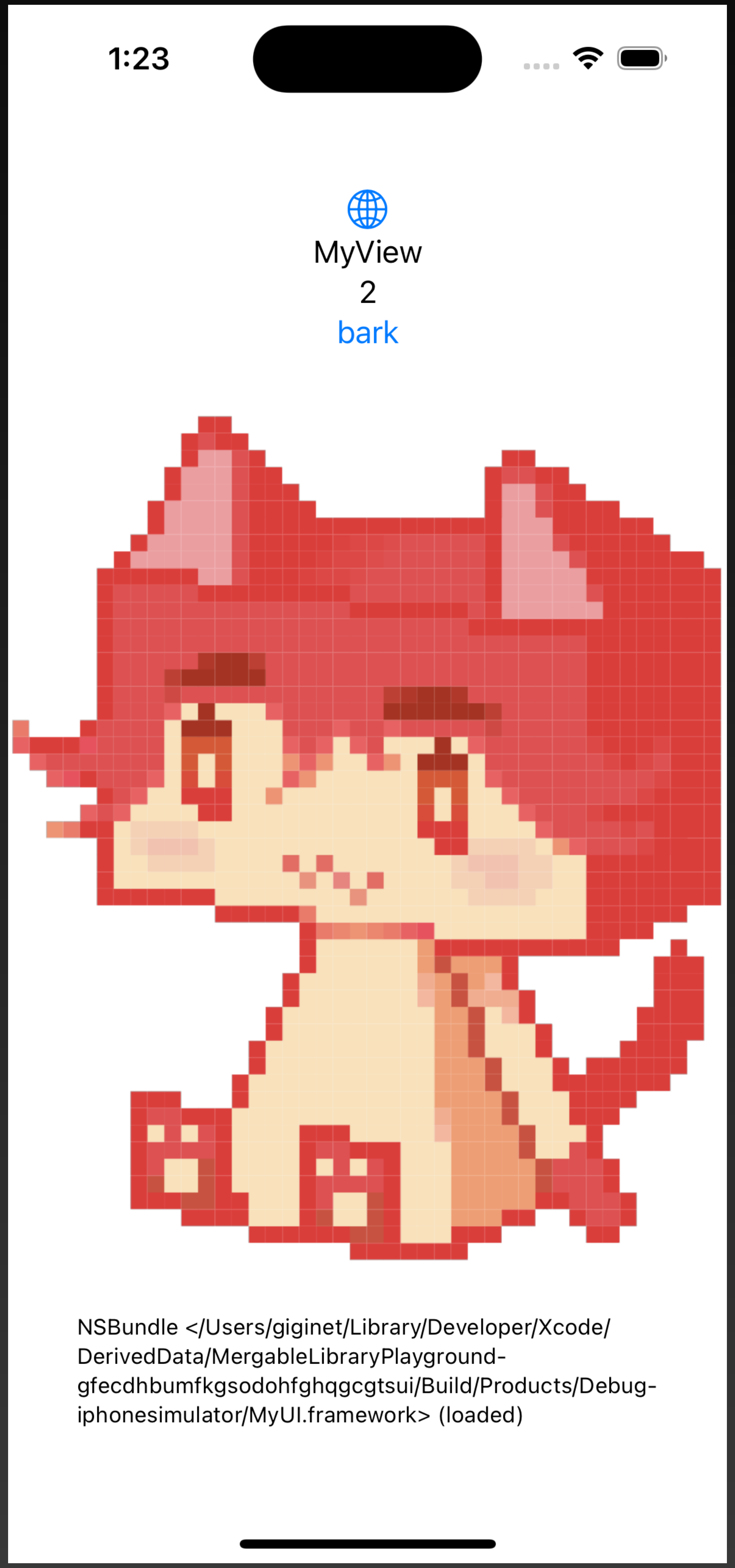
- Runtime lookup with Mergeable Libraries works fine
  - It’s mentioned in the WWDC session
- However, Resource bundle with Static Frameworks cause unexpected behavior

Resources in MyUI framework

```
import Foundation
import SwiftUI

private let bundle = Bundle(for: ResourceFinder.self)

private class ResourceFinder {}

public struct MyImage: View {
    public init() { }

    public var body: some View {
        Image("giginet", bundle: bundle)
        Text(bundle.description)
            .font(.caption)
    }
}
```



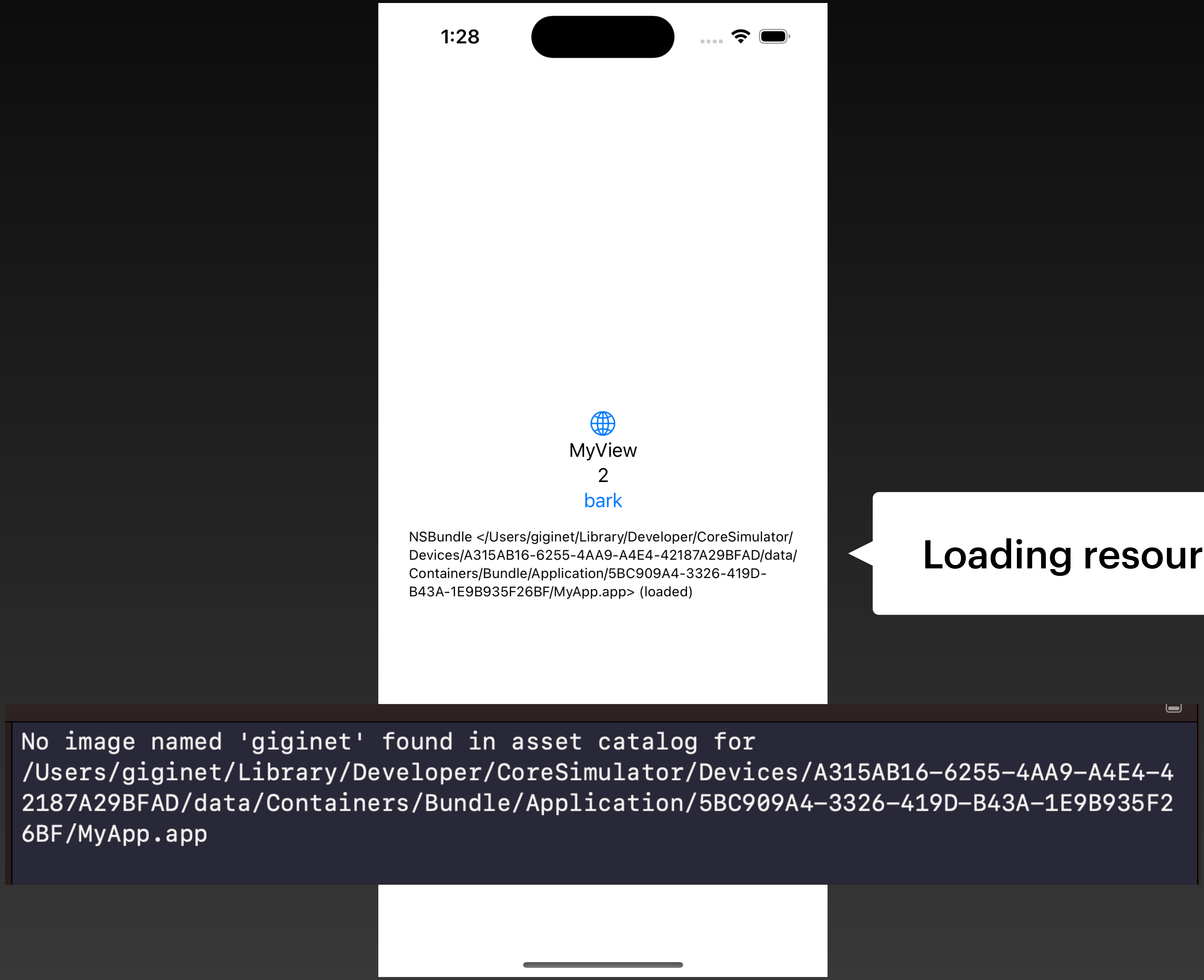
Debug Configuration



```
NSBundle </Users/giginet/Library/Developer/Xcode/
DerivedData/MergableLibraryPlayground-
gfecdhbumfkgsodohfghqgcgtsui/Build/Products/Debug-
iphonesimulator/MyUI.framework> (loaded)
```

Loading resources from MyUI’s bundle ✅

Release Configuration



MyView

2

bark

NSBundle </Users/giginet/Library/Developer/CoreSimulator/Devices/A315AB16-6255-4AA9-A4E4-42187A29BFAD/data/Containers/Bundle/Application/5BC909A4-3326-419D-B43A-1E9B935F26BF/MyApp.app> (loaded)

```
No image named 'giginet' found in asset catalog for
/Users/giginet/Library/Developer/CoreSimulator/Devices/A315AB16-6255-4AA9-A4E4-4
2187A29BFAD/data/Containers/Bundle/Application/5BC909A4-3326-419D-B43A-1E9B935F2
6BF/MyApp.app
```

Loading resources from main bundle 💥

Resource bundle with Mergeable Library

- Dynamic lookup works fine with Mergeable Library
    - But Static Frameworks can‘t get their bundles
    - It causes implicit unexpected behavior
- Need to consider best practice using Mergeable Library for frameworks with resources🤔

Recap

- **Mergeable Library** is the best feature available on Xcode 15
- It works fine in most cases. Let’s enable it now!!!
- Some runtime features may expect unexpected behavior
  - runtime behaviors would be changed by build configurations

Resources

- Meet Mergeable Library
  - https://developer.apple.com/videos/play/wwdc2023/10268/
- Configuring your project to use mergeable libraries
  - https://developer.apple.com/documentation/Xcode/configuring-your-project-to-use-mergeable-libraries

Thank you for your attention