

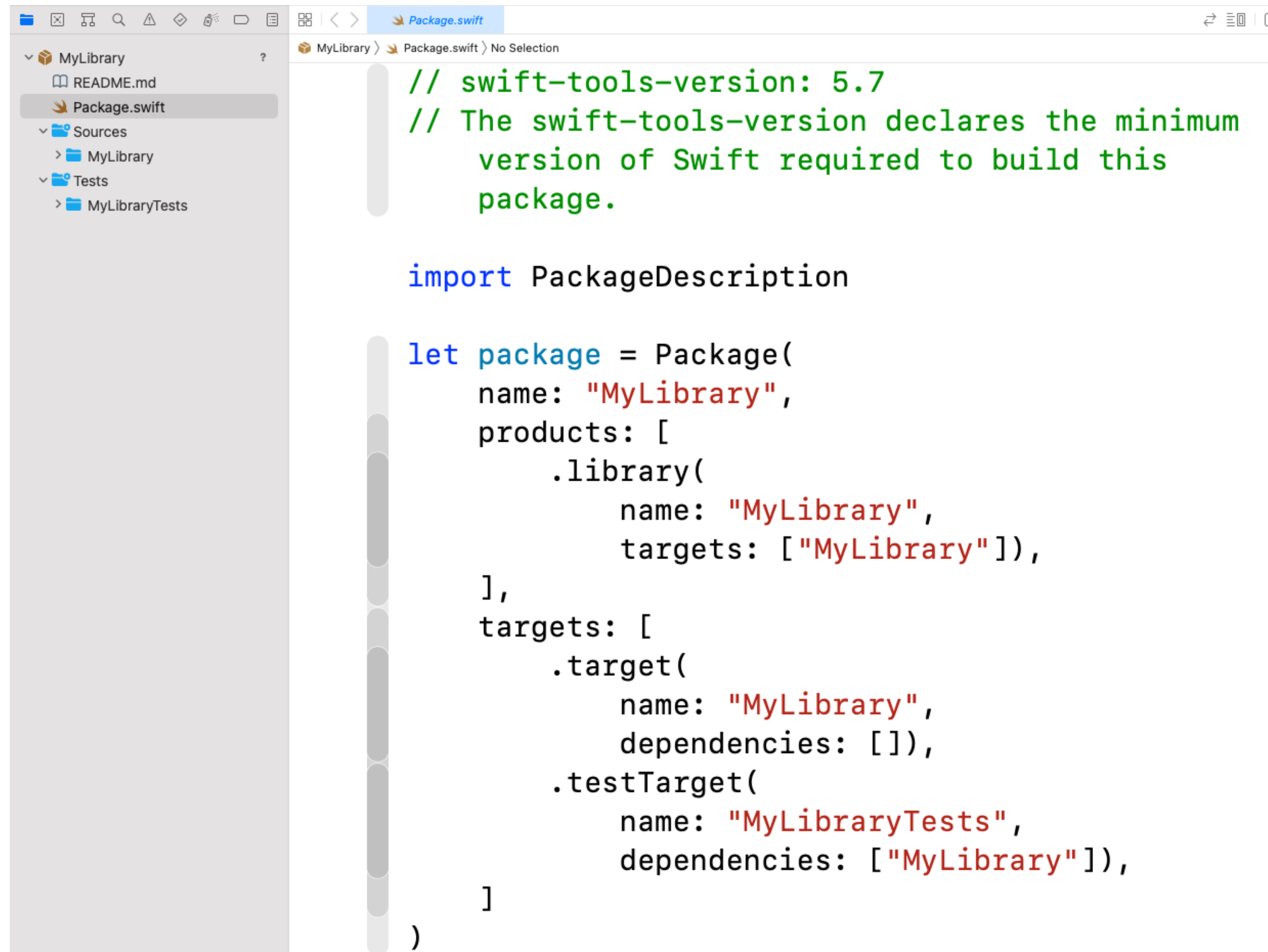
쉽고 재미있는 Swift Package 동적 프레임워크 만들기

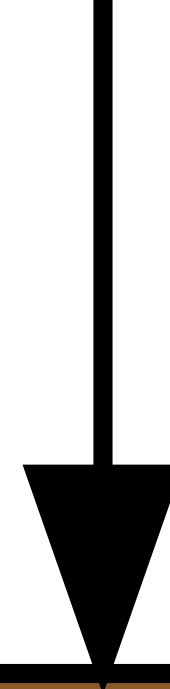
Swift Package

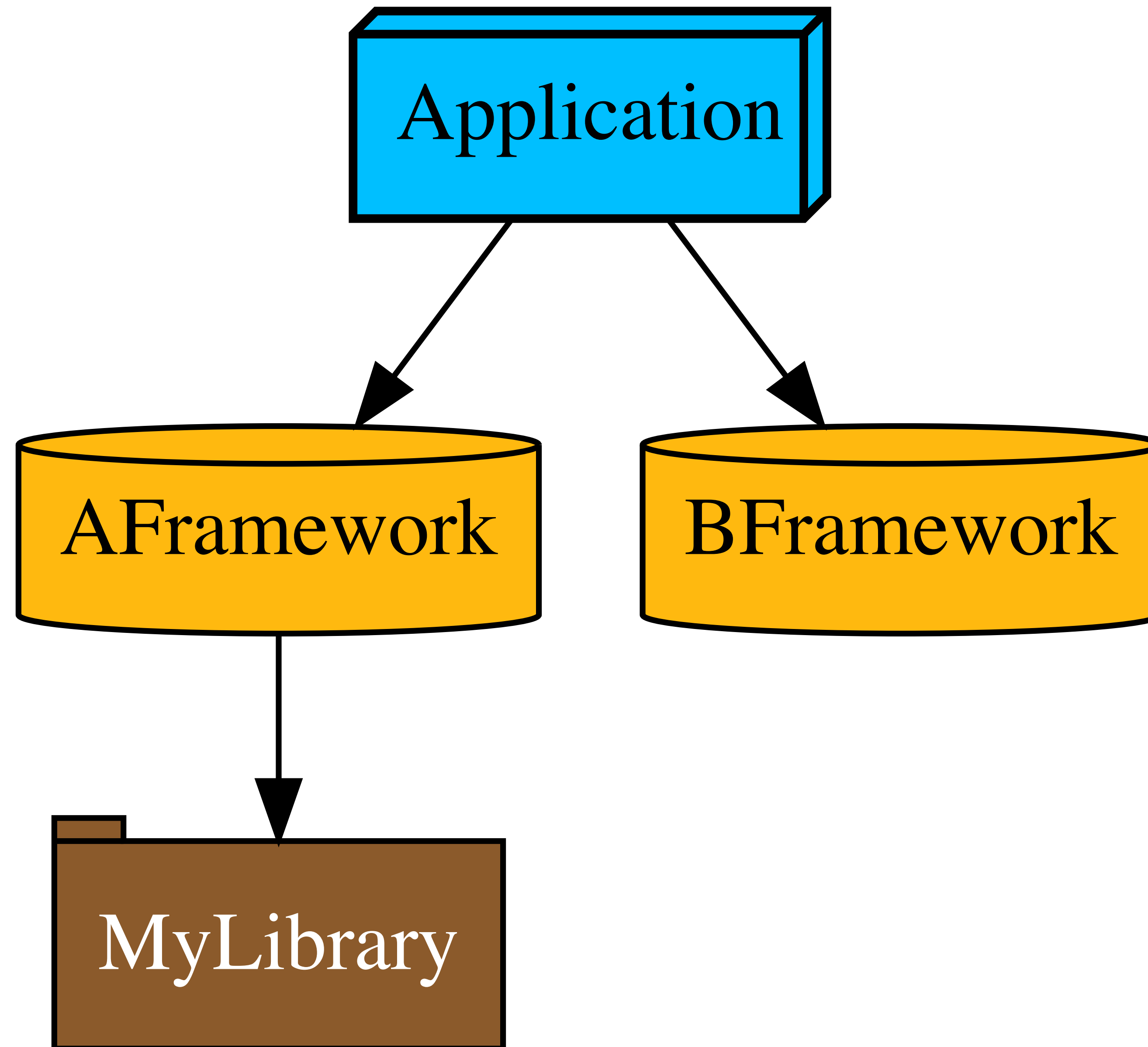


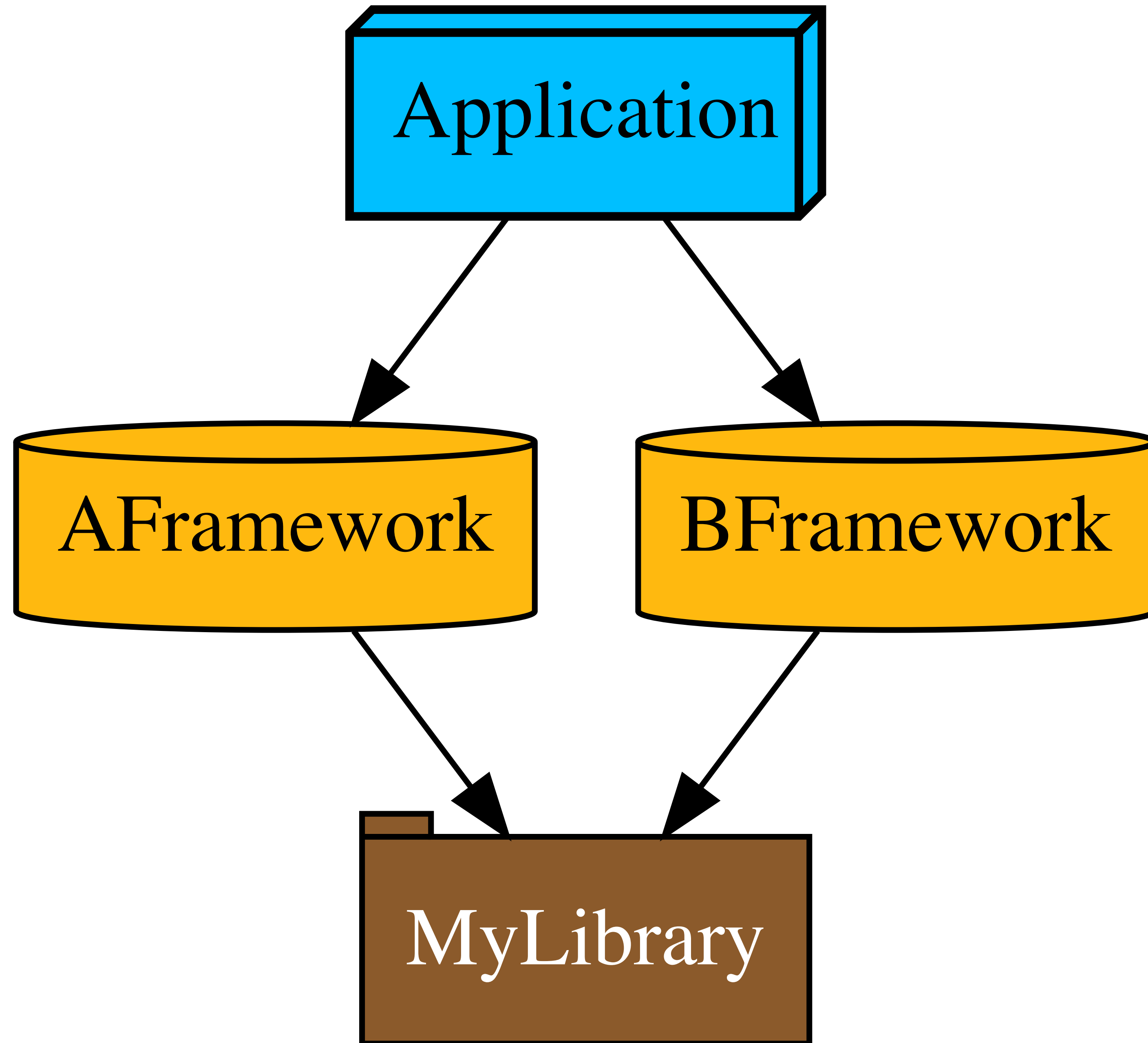
Swift Package

- 기본 Mach-O는 Static Library









MyLibrary

MyLibrary-Package > My Mac

MyLibrary: Ready | Today at 12:51 AM

1

Package.swift

MyLibrary > Package.swift > No Selection

```
// swift-tools-version: 5.7
// The swift-tools-version declares the minimum version of Swift
// required to build this package.

import PackageDescription

let package = Package(
    name: "MyLibrary",
    products: [
        .library(name: "MyLibrary",
            targets: ["MyLibrary"]),
        .library(name: "MyLibrary-Dynamic",
            type: .dynamic,
            targets: ["MyLibrary"]),
    ],
    targets: [
        .target(name: "MyLibrary"),
        .testTarget(name: "MyLibraryTests",
            dependencies: ["MyLibrary"]),
    ]
)
```

Filter

Line: 22 Col: 1

Swift Packages

New Features

- The Swift Package Manager now builds package products and targets as dynamic frameworks automatically, if doing so avoids duplication of library code at runtime. (59931771) (FB7608638)

이제 Swift 패키지 관리자는 런타임에 라이브러리 코드 중복을 방지하는 경우 패키지 제품 및 타겟을 동적 프레임워크로 자동 빌드합니다. (59931771) (fb7608638)

- DeepL 번역

RxSwift-main

RxSwift-Package > Any iOS Device RxSwift-main: Ready | Today at 11:08 PM

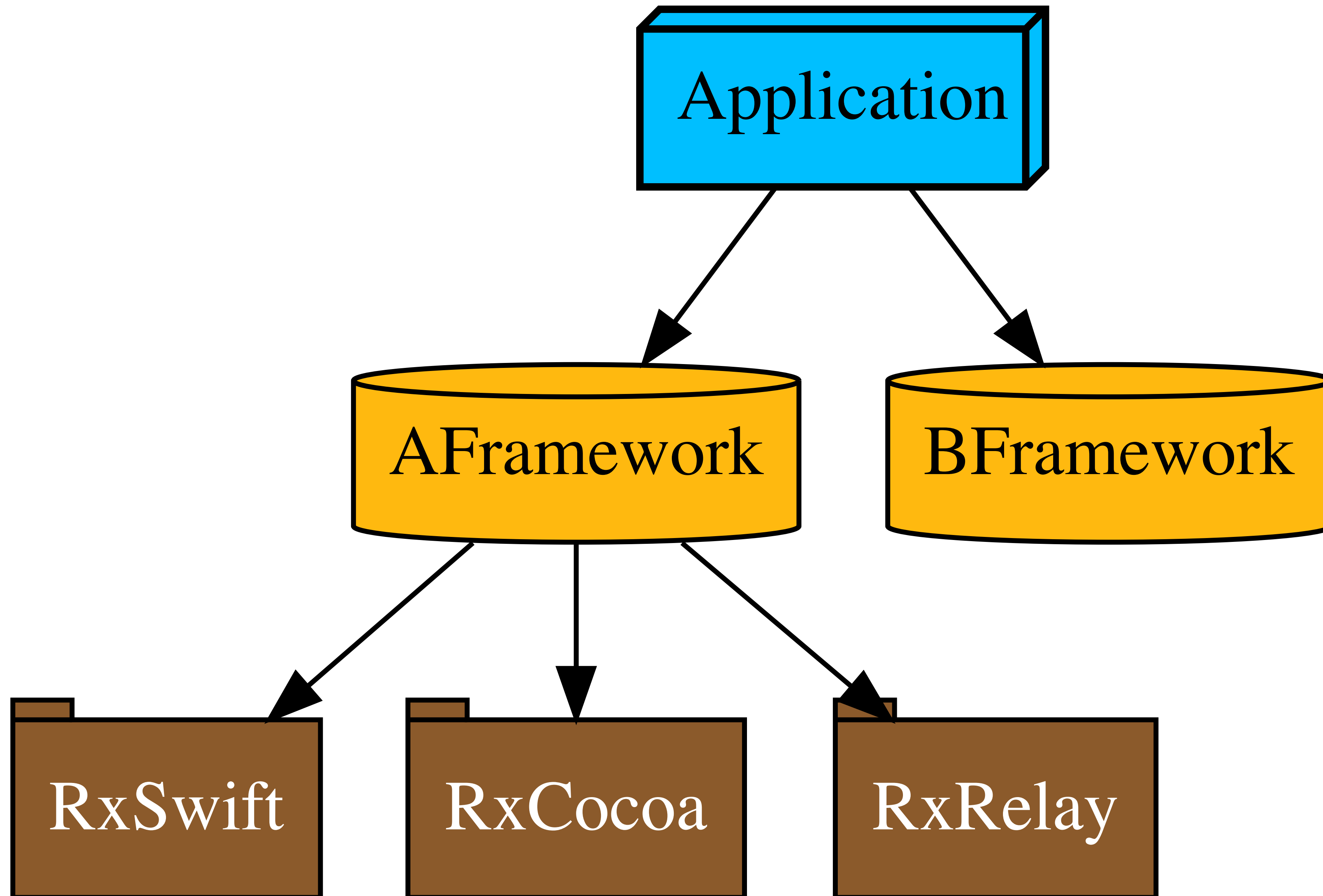
Package.swift

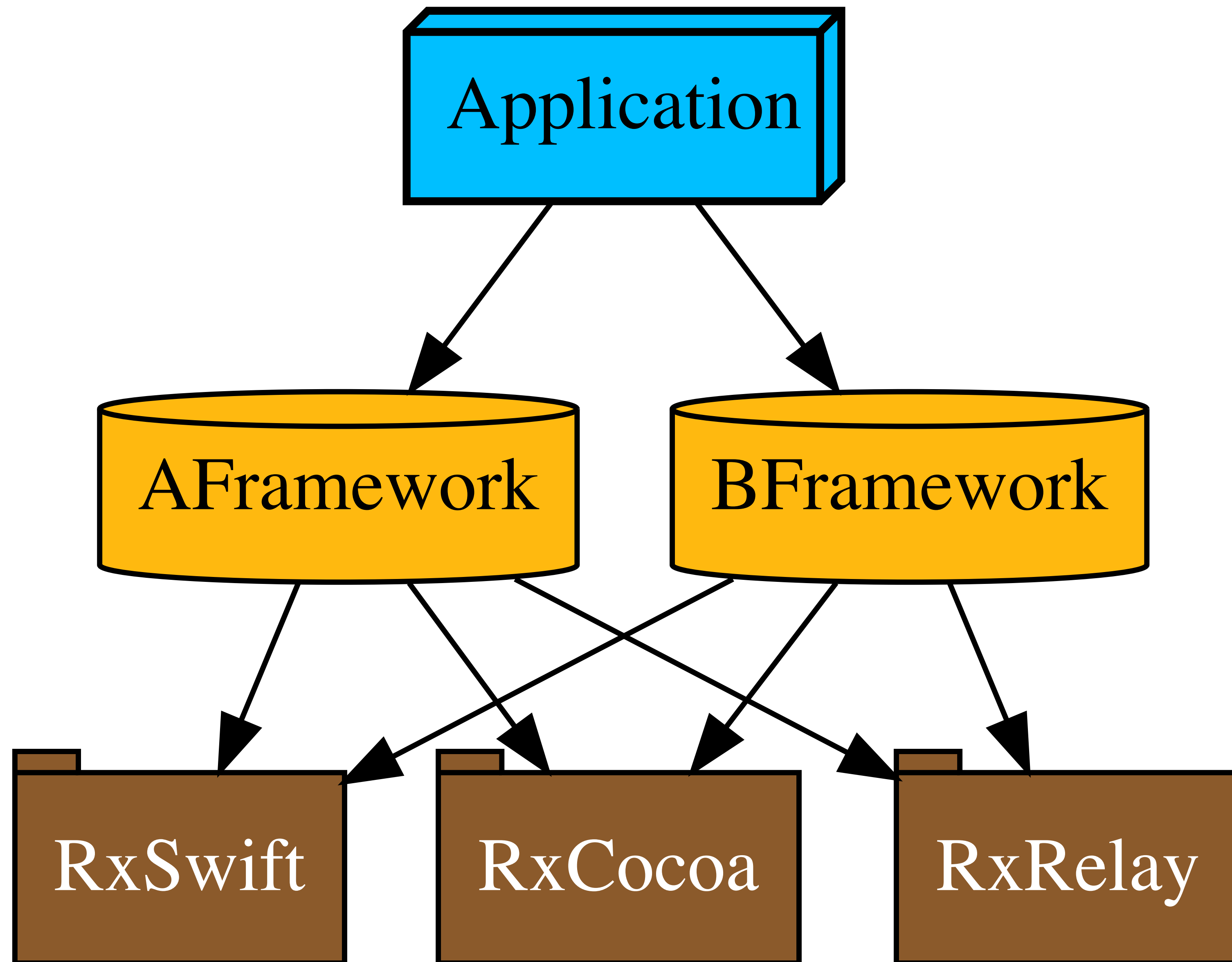
RxSwift-main > Package.swift > No Selection

```
let package = Package(
    name: "RxSwift",
    platforms: [.iOS(.v9), .macOS(.v10_10), .watchOS(.v3), .tvOS(.v9)],
    products: ([
        .library(name: "RxSwift", targets: ["RxSwift"]),
        .library(name: "RxCocoa", targets: ["RxCocoa"]),
        .library(name: "RxRelay", targets: ["RxRelay"]),
        .library(name: "RxBlocking", targets: ["RxBlocking"]),
        .library(name: "RxTest", targets: ["RxTest"]),
        .library(name: "RxSwift-Dynamic", type: .dynamic, targets: ["RxSwift"]),
        .library(name: "RxCocoa-Dynamic", type: .dynamic, targets: ["RxCocoa"]),
        .library(name: "RxRelay-Dynamic", type: .dynamic, targets: ["RxRelay"]),
        .library(name: "RxBlocking-Dynamic", type: .dynamic, targets:
            ["RxBlocking"]),
        .library(name: "RxTest-Dynamic", type: .dynamic, targets: ["RxTest"]),
    ],
    Product.allTests()
    ] as [[Product]]).flatMap { $0 },
    targets: ([
        .target(name: "RxSwift", dependencies: []),
    ],
    Target.rxCocoa(),
    Target.rxCocoaRuntime(),
    r
```

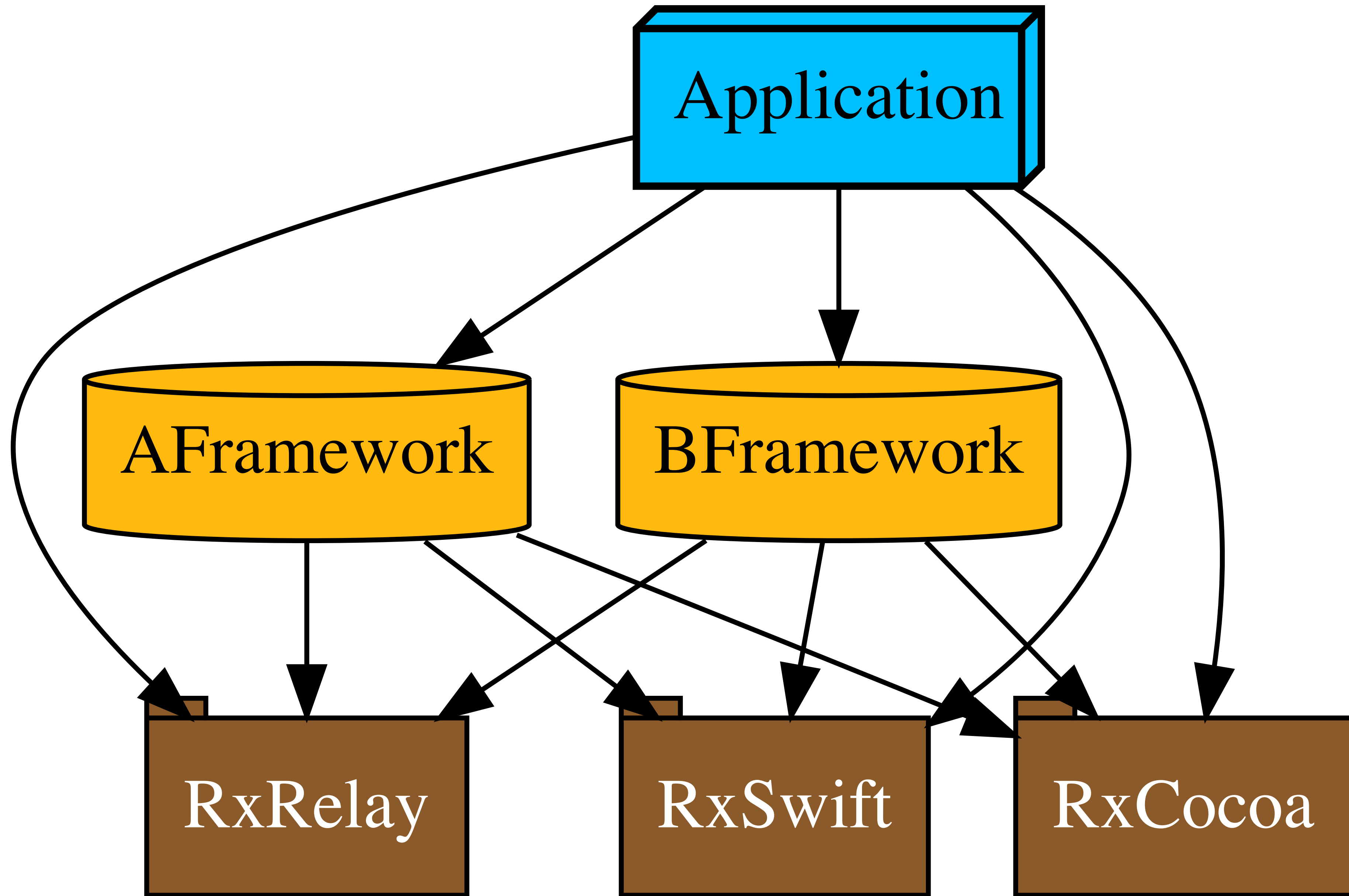
Line: 72 Col: 21

<https://github.com/reactiveX/RxSwift>





최종



끝