

if(kakao)2022

# 카카오뱅크 iOS 프로젝트의 모듈화 여정

Tuist를 활용한 모듈 아키텍처 설계 사례

안정민 **damon.ahn**  
카카오뱅크

# 발표를 통해 기대하는 것

- 카카오뱅크 iOS 프로젝트의 진행 내용 공유
- 현재 작업중인 모듈 아키텍처 설계 공유

프로젝트 여정

개발 생산성 개선 방법 알아보기

카카오뱅크 모듈 아키텍처 설계

프로젝트 여정

개발 생산성 개선 방법 알아보기

카카오뱅크 모듈 아키텍처 설계

# 프로젝트 여정

# 아키텍처

- 장점
  - 데이터 바인딩으로 UI와 Model 간의 동기화



MVVM

# 아키텍처

- 장점
  - 데이터 바인딩으로 UI와 Model 간의 동기화
- 단점
  - 비대해지는 ViewModel 구조
  - 정형화되지 않은 Routing



# 아키텍처

# 아키텍처

- 장점
  - 명확한 템플릿
  - 프로토콜 지향 프로그래밍 가능
  - 응집도 높은 코드
  - Routing 기능



# 아키텍처

- 장점
  - 명확한 템플릿
  - 프로토콜 지향 프로그래밍 가능
  - 응집도 높은 코드
  - Routing 기능



# 아키텍처

- 장점
  - 명확한 템플릿
  - 프로토콜 지향 프로그래밍 가능
  - 응집도 높은 코드
  - Routing 기능
- 단점
  - 많은 파일 생성



# 아키텍처

- 장점
  - 명확한 템플릿
  - 프로토콜 지향 프로그래밍 가능
  - 응집도 높은 코드
  - Routing 기능
- 단점
  - 많은 파일 생성



# 프로젝트 규모

\_\_\_\_\_ MVVM \_\_\_\_\_

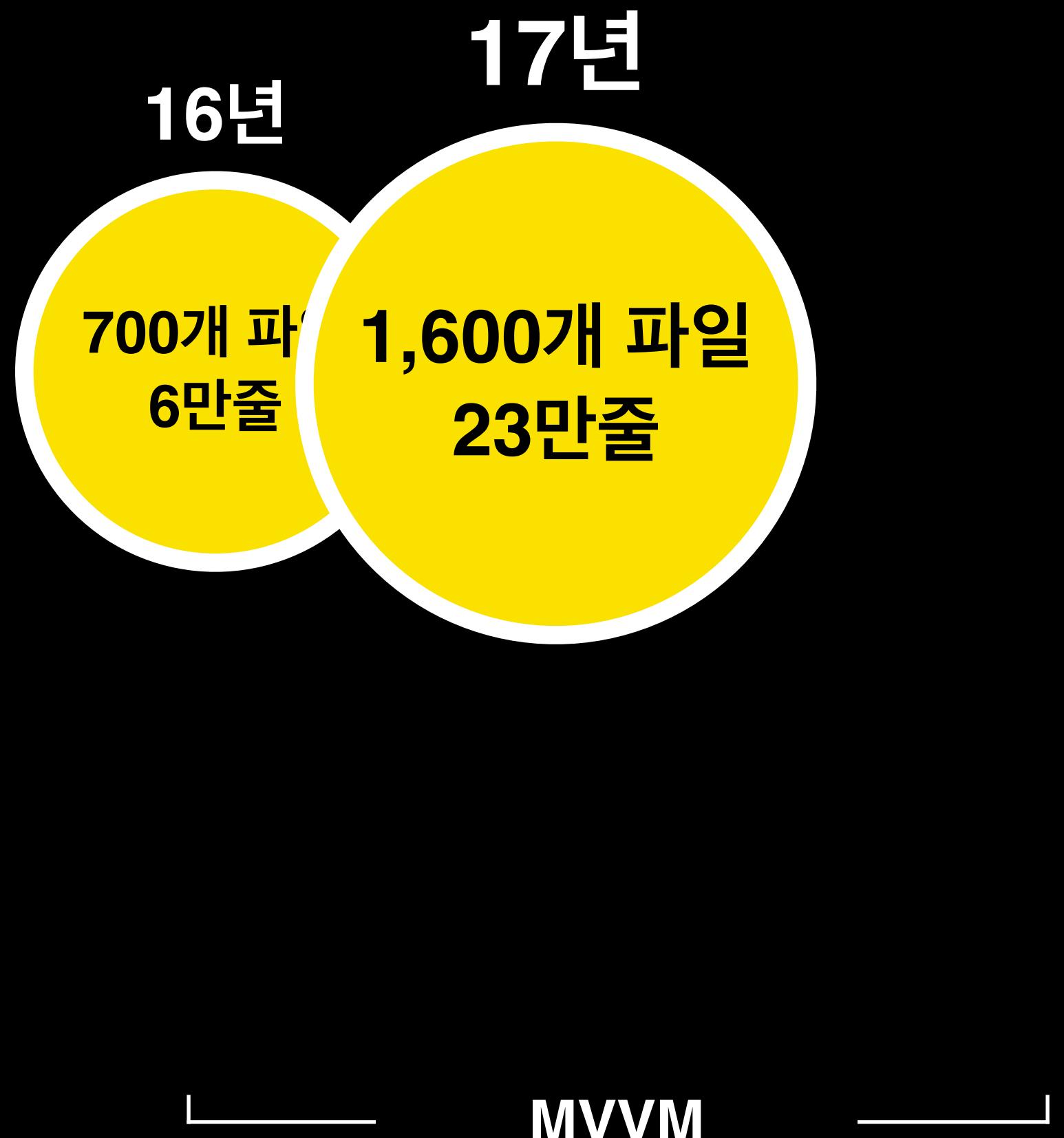
# 프로젝트 규모

16년

700개 파일  
6만줄

\_\_\_\_\_ MVVM \_\_\_\_\_

# 프로젝트 규모

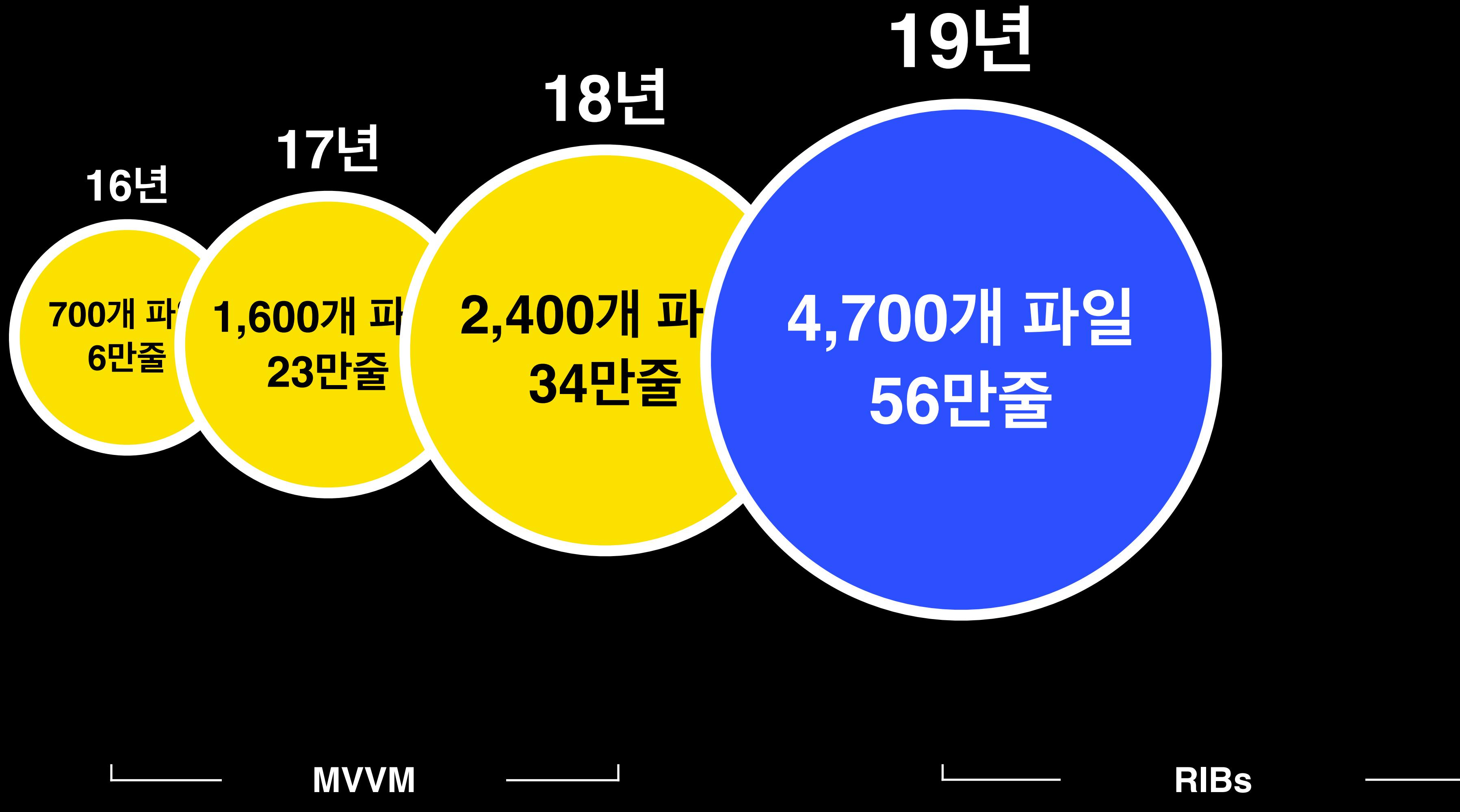


# 프로젝트 규모



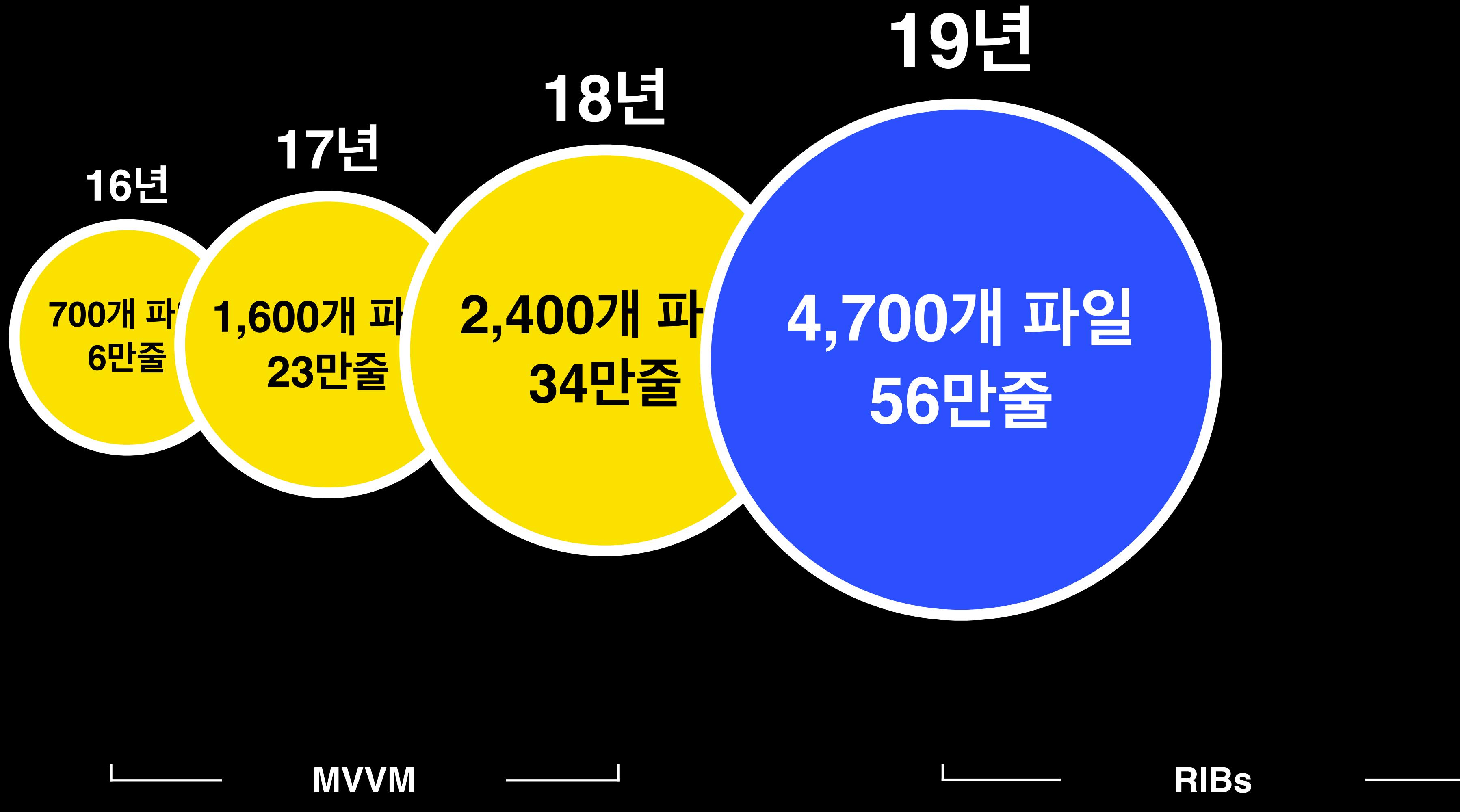
MVVM

# 프로젝트 규모



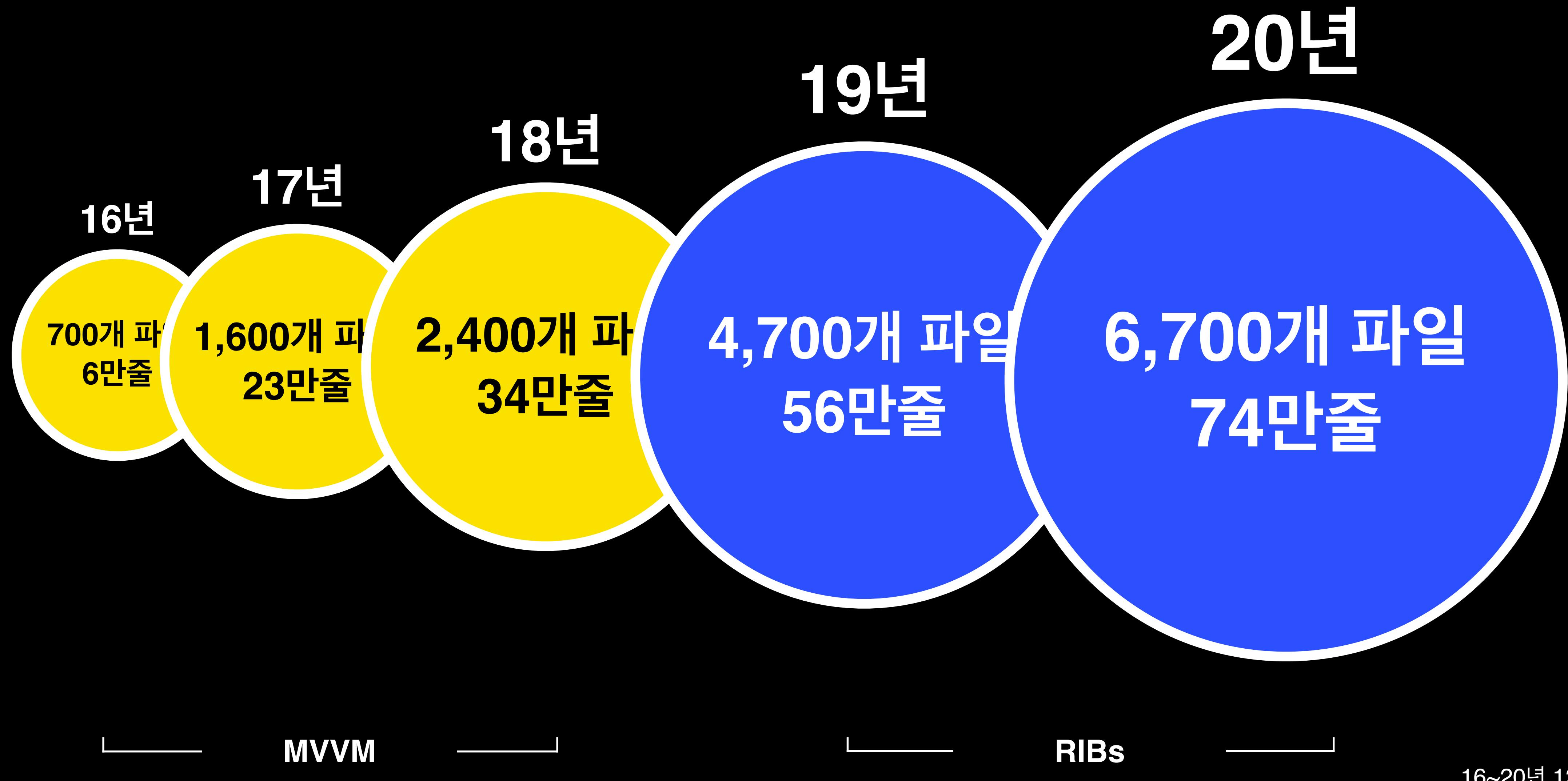
16~20년 12.31 기준

# 프로젝트 규모



16~20년 12.31 기준

# 프로젝트 규모





**11,000개 파일  
104만 줄**

2022.08.31 기준



# 결합도 증가

손쉽게 코드 접근

# 결합도 증가

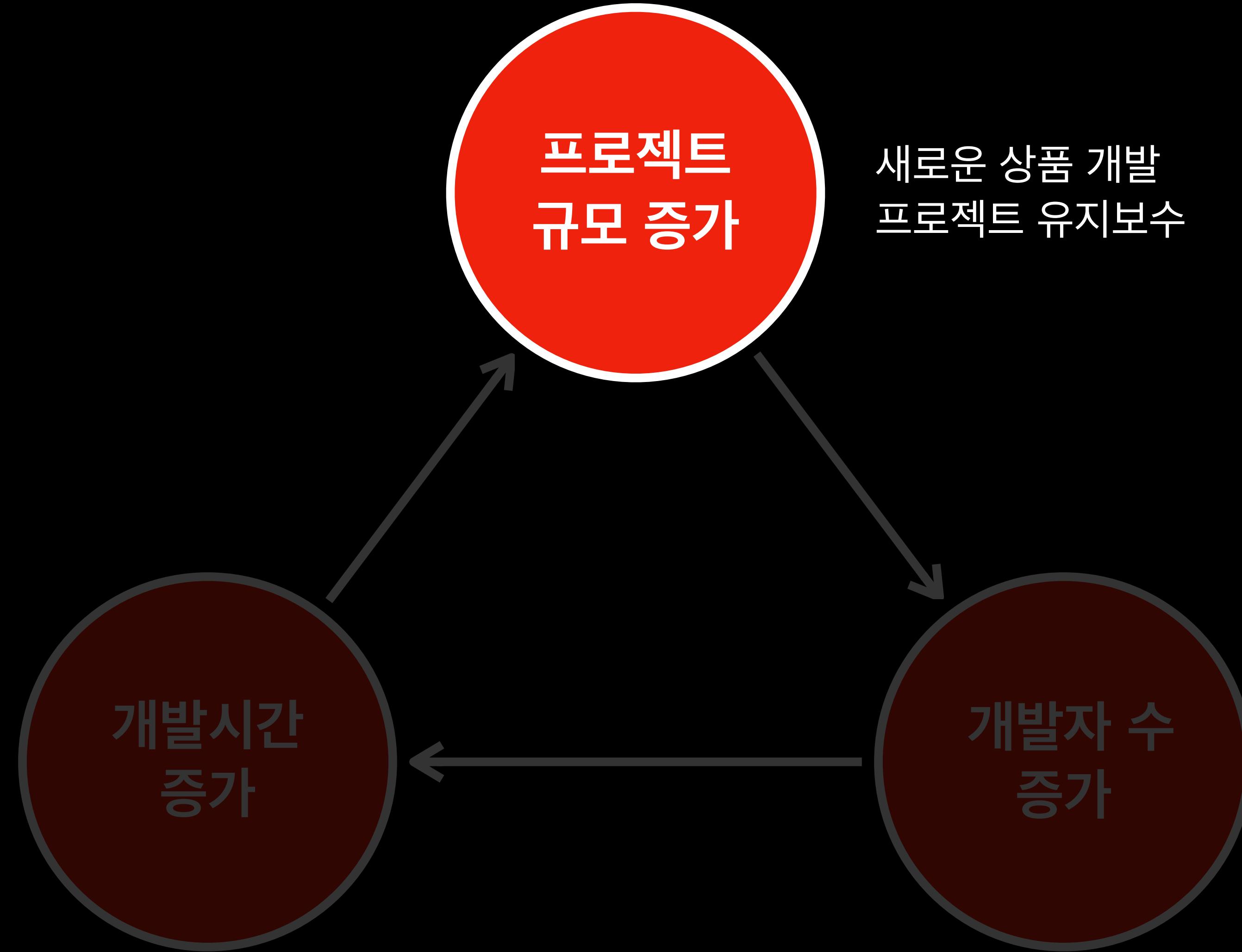
손쉽게 코드 접근

# 결합도 증가      빌드 문제

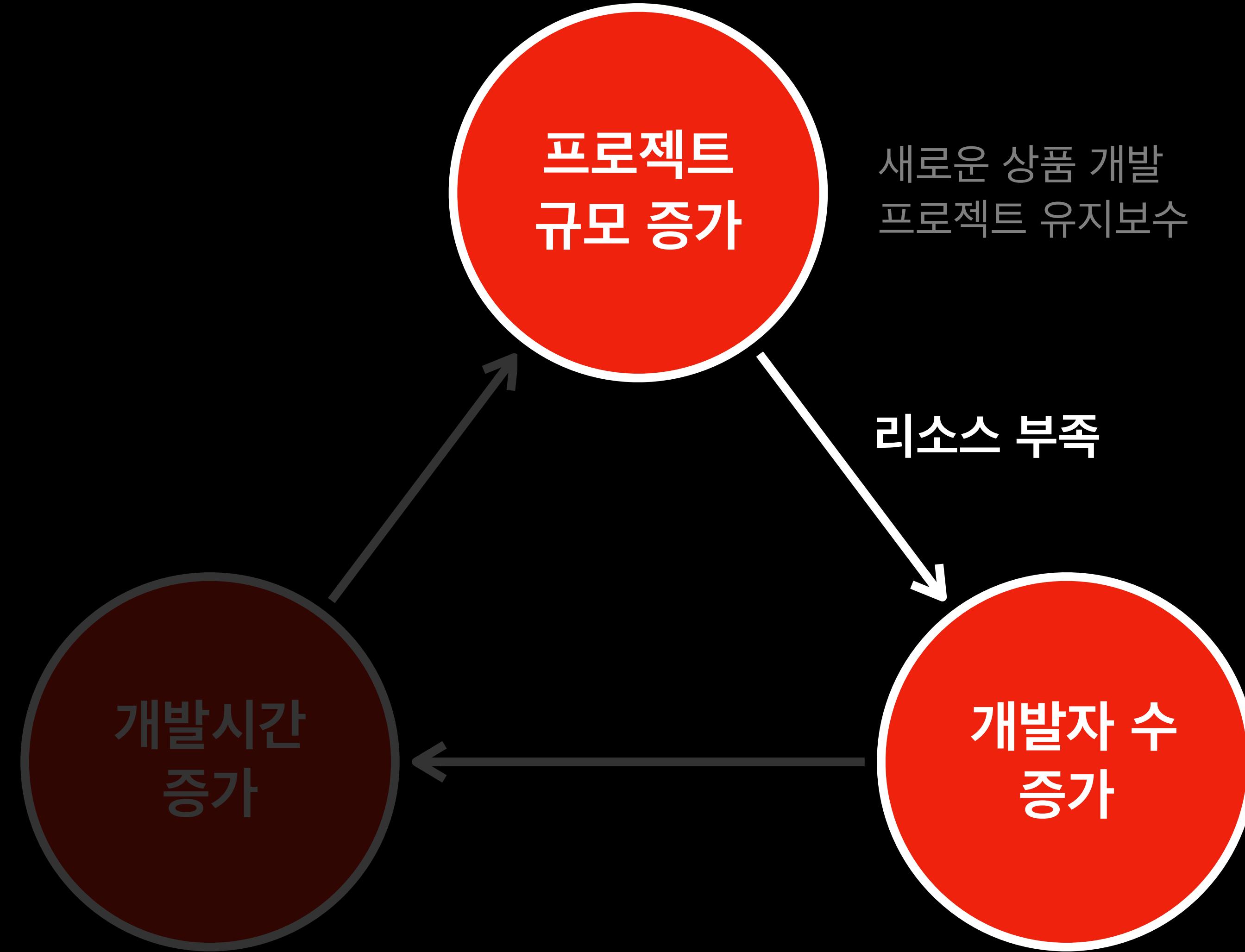
손쉽게 코드 접근

빌드 시간 증가, 빌드 실패

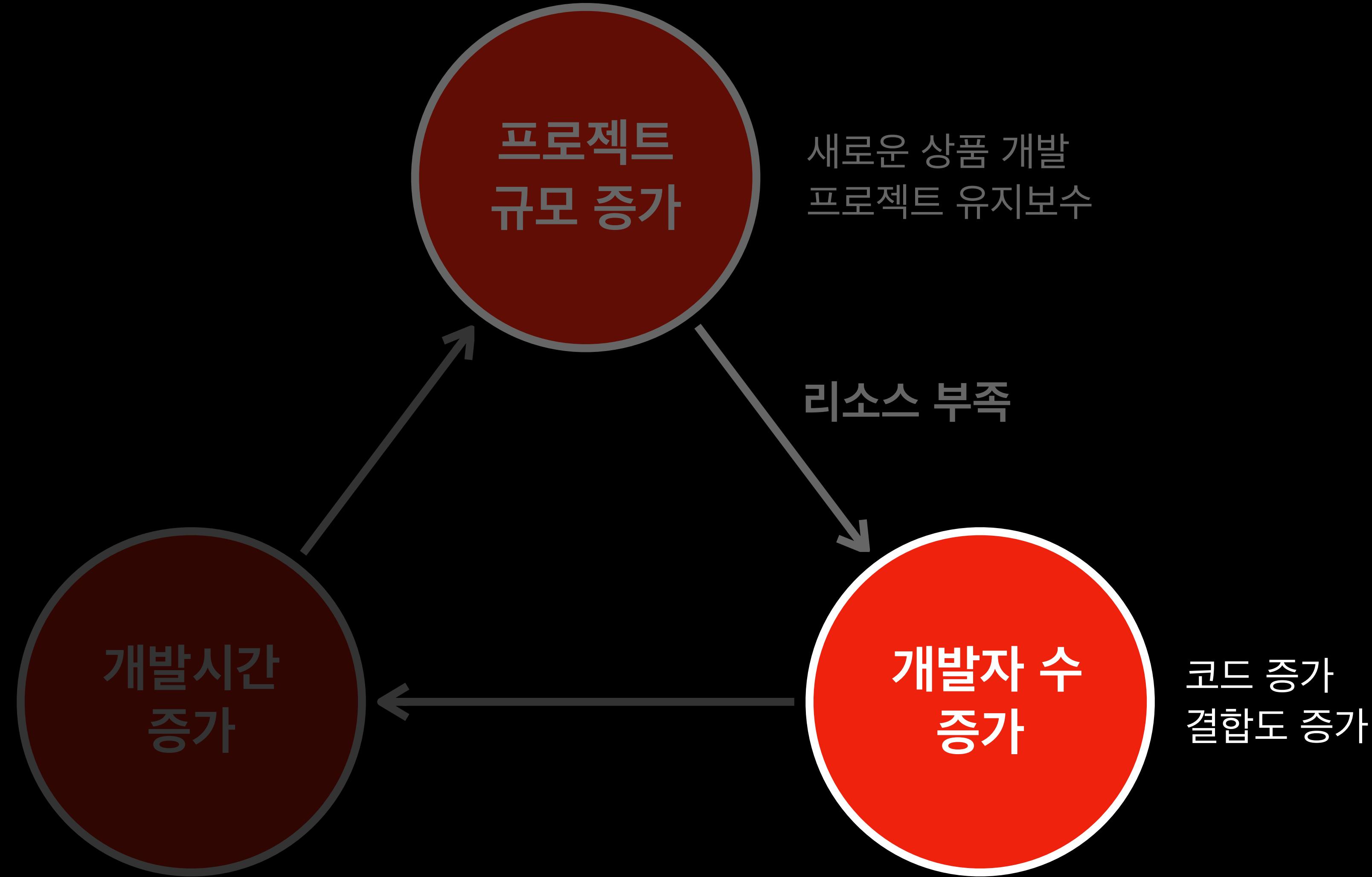
# 프로젝트 규모



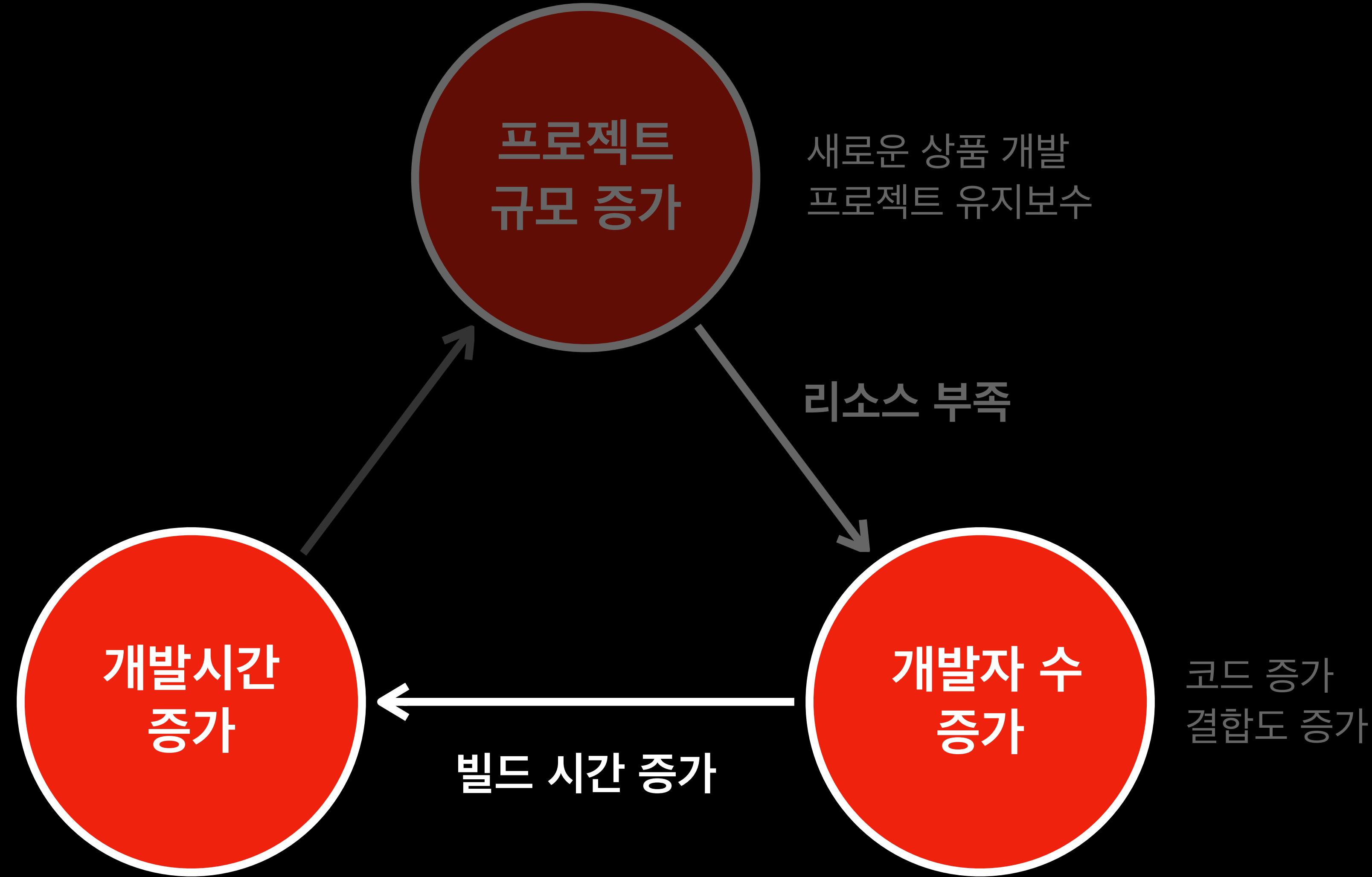
# 프로젝트 규모



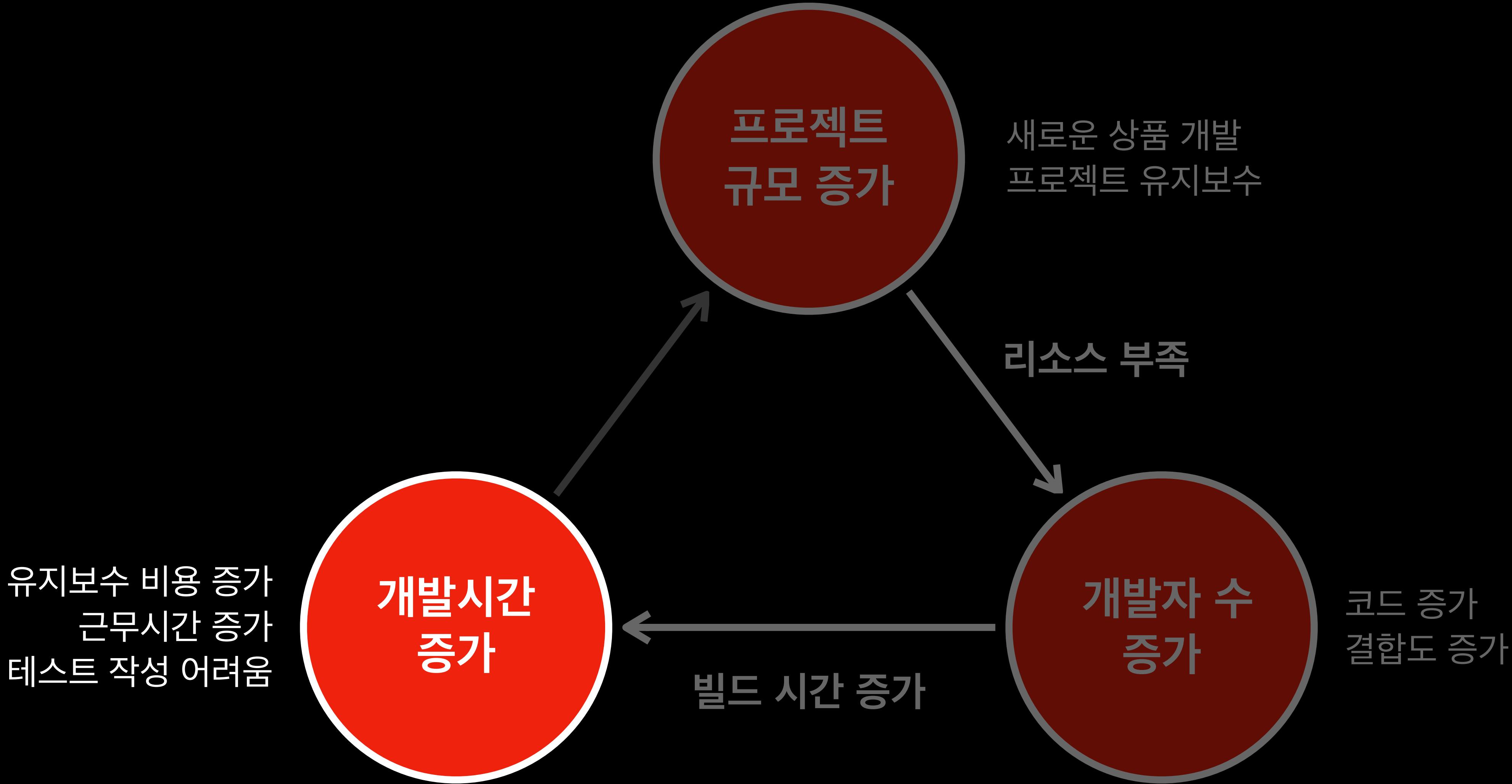
# 프로젝트 규모



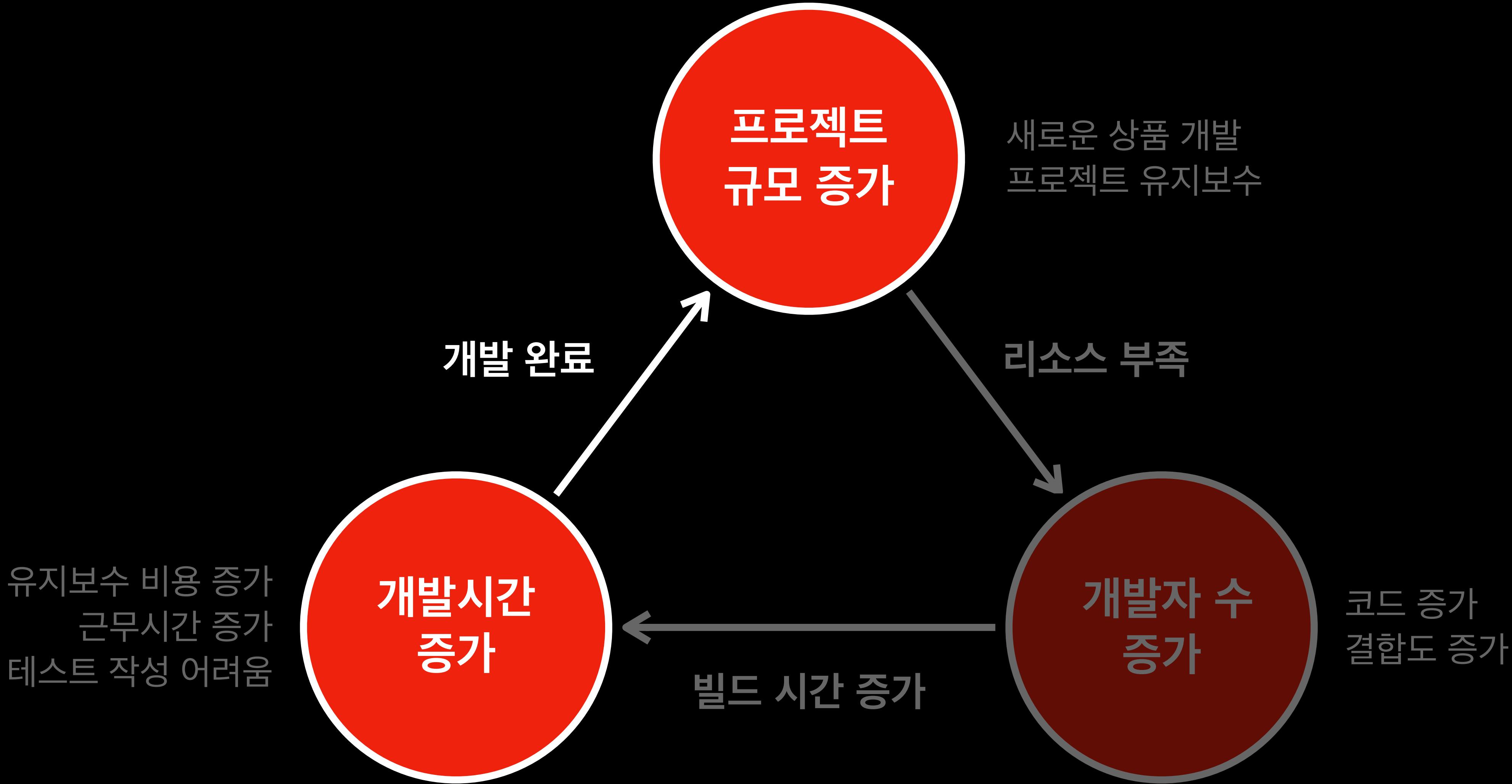
# 프로젝트 규모



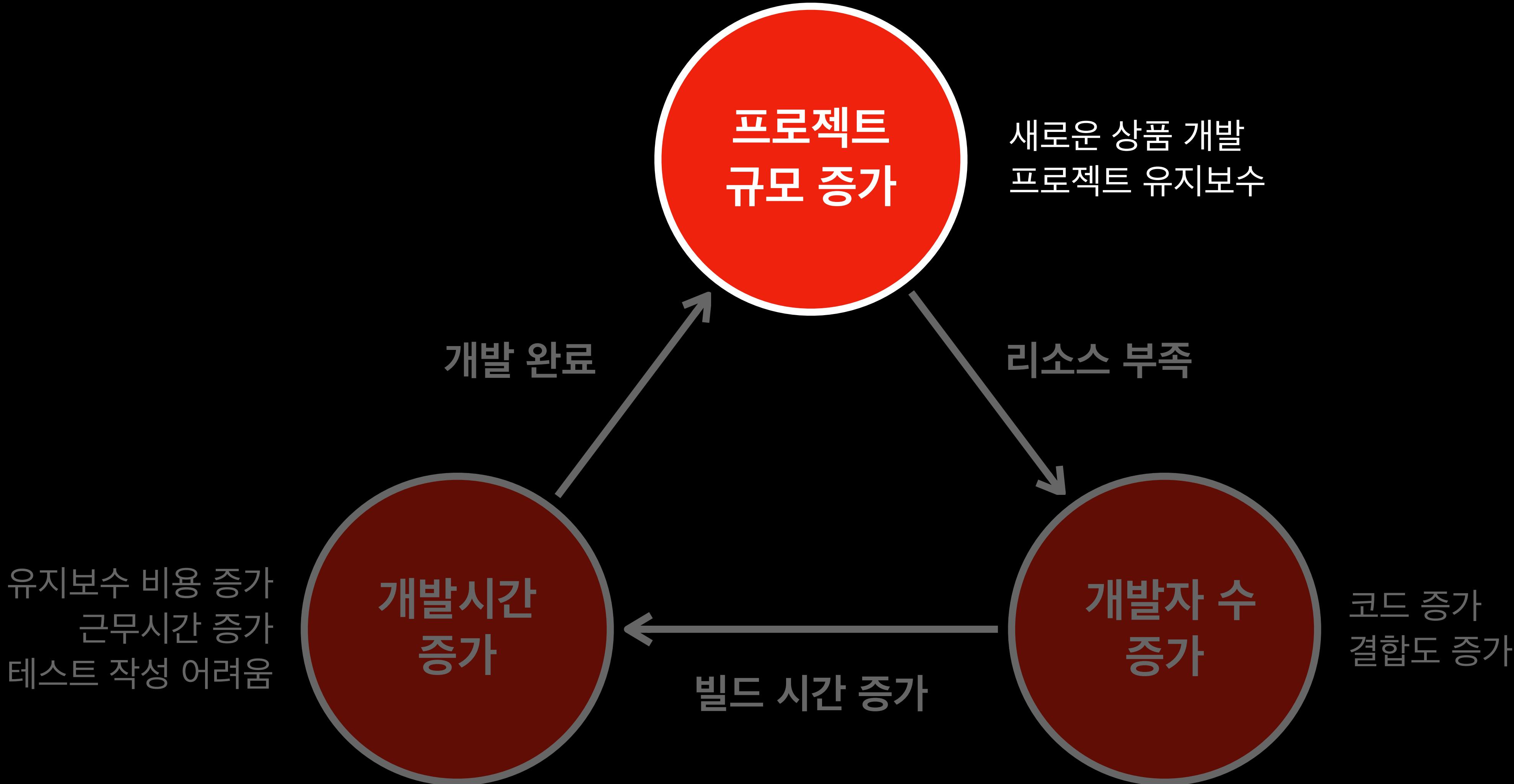
# 프로젝트 규모



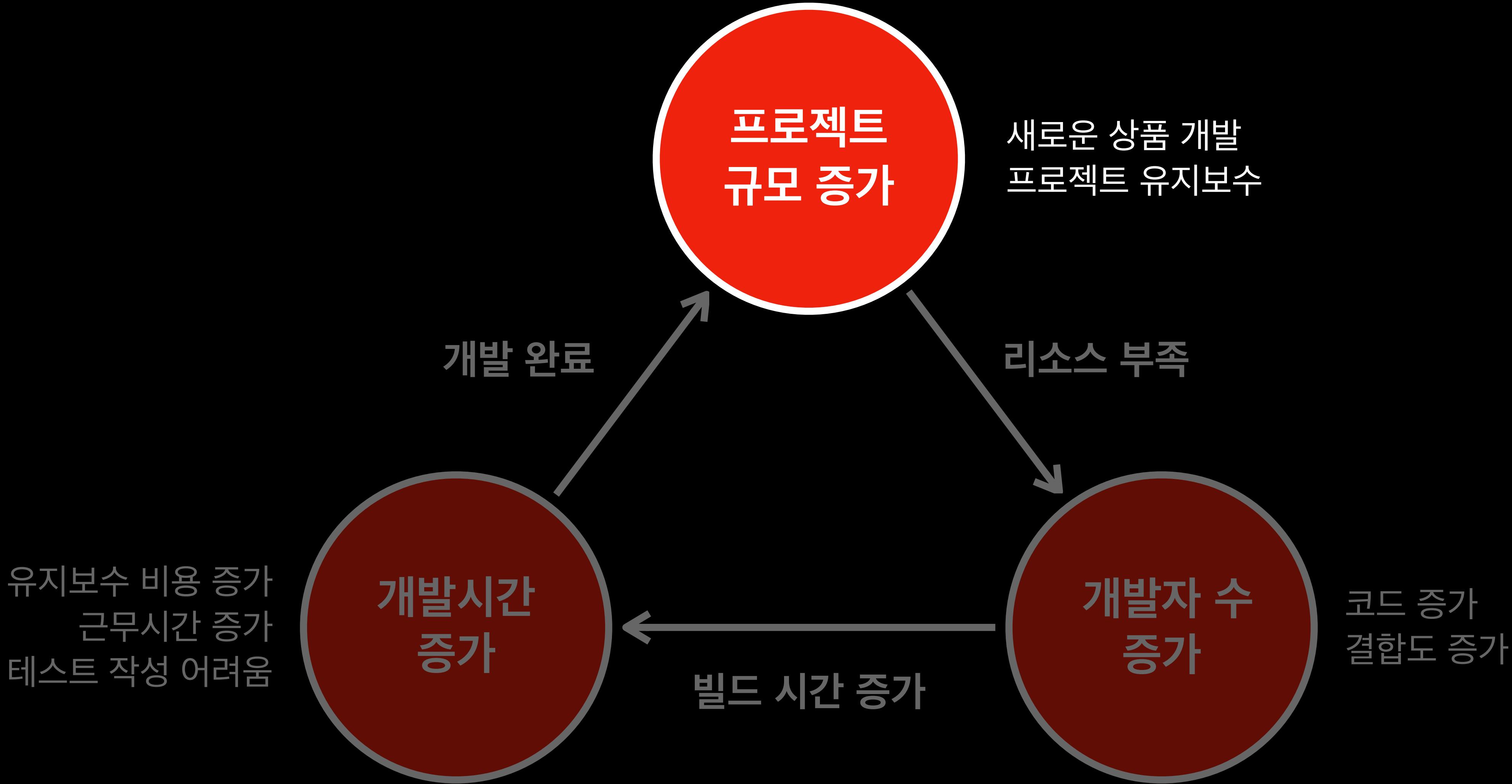
# 프로젝트 규모



# 프로젝트 규모



# 프로젝트 규모



# 모듈화

모듈로 개발할 수 있다면 해결할 수 있지 않을까?

# 프로젝트 규모

# 프로젝트 규모

17년

3개 모듈

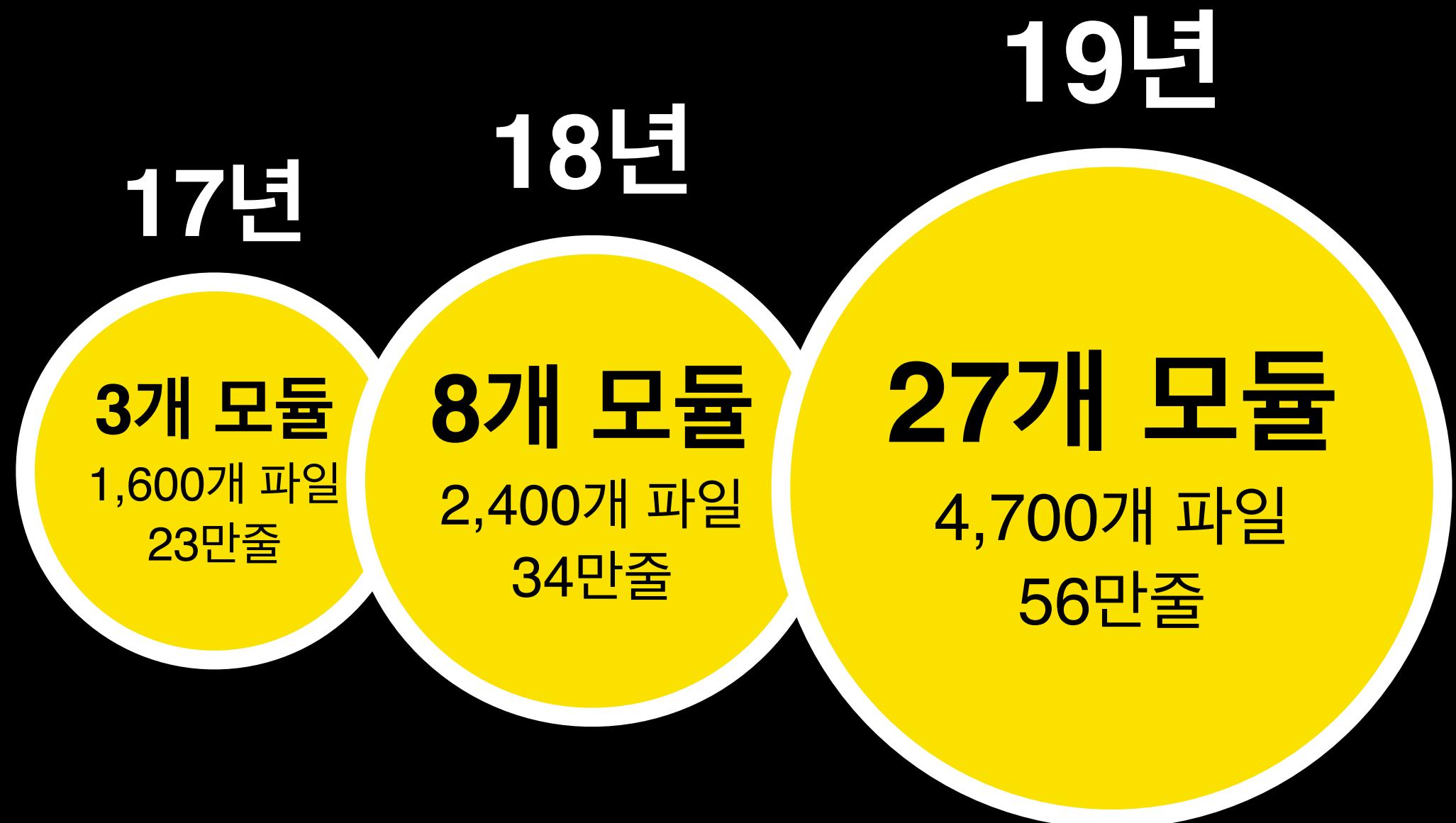
1,600개 파일

23만줄

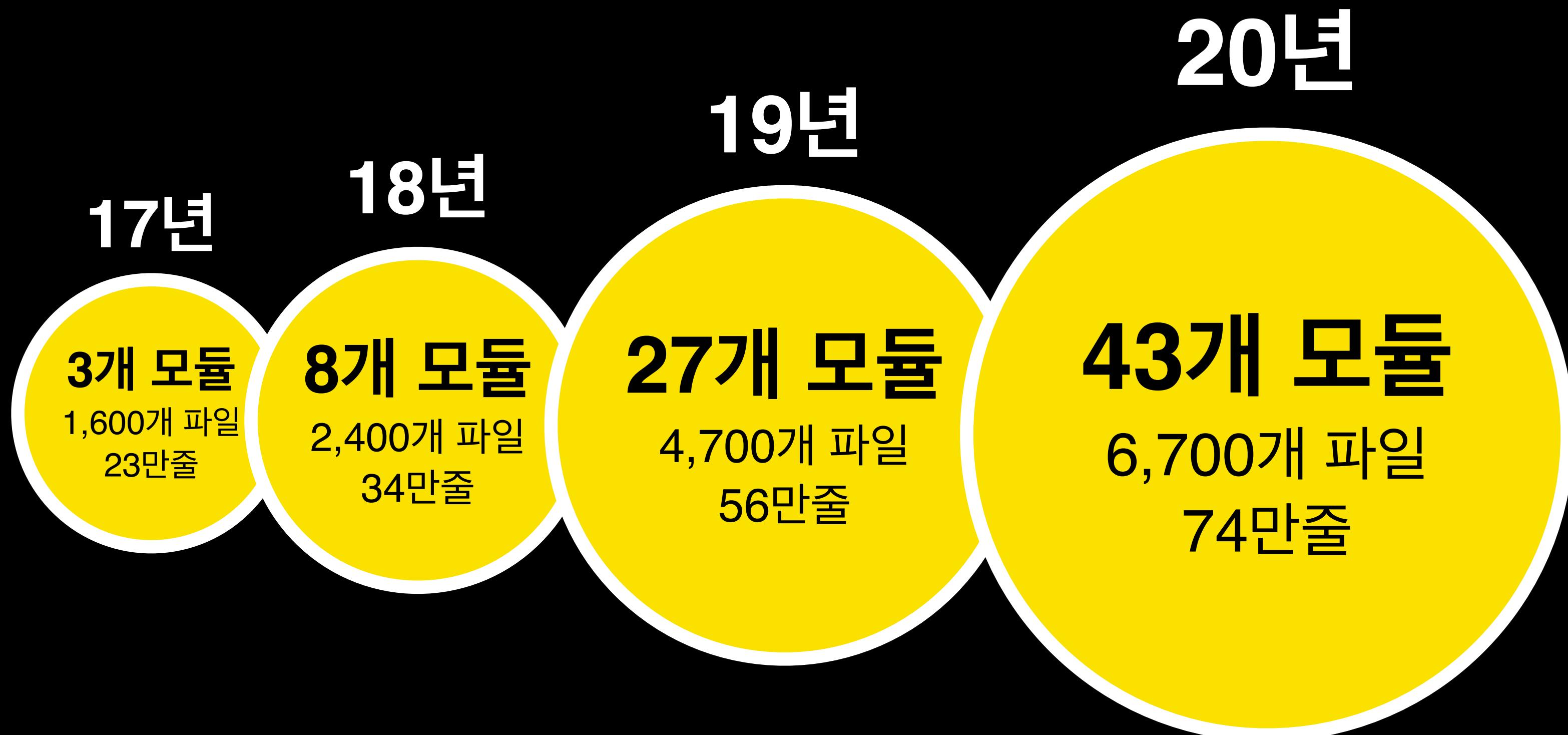
# 프로젝트 규모



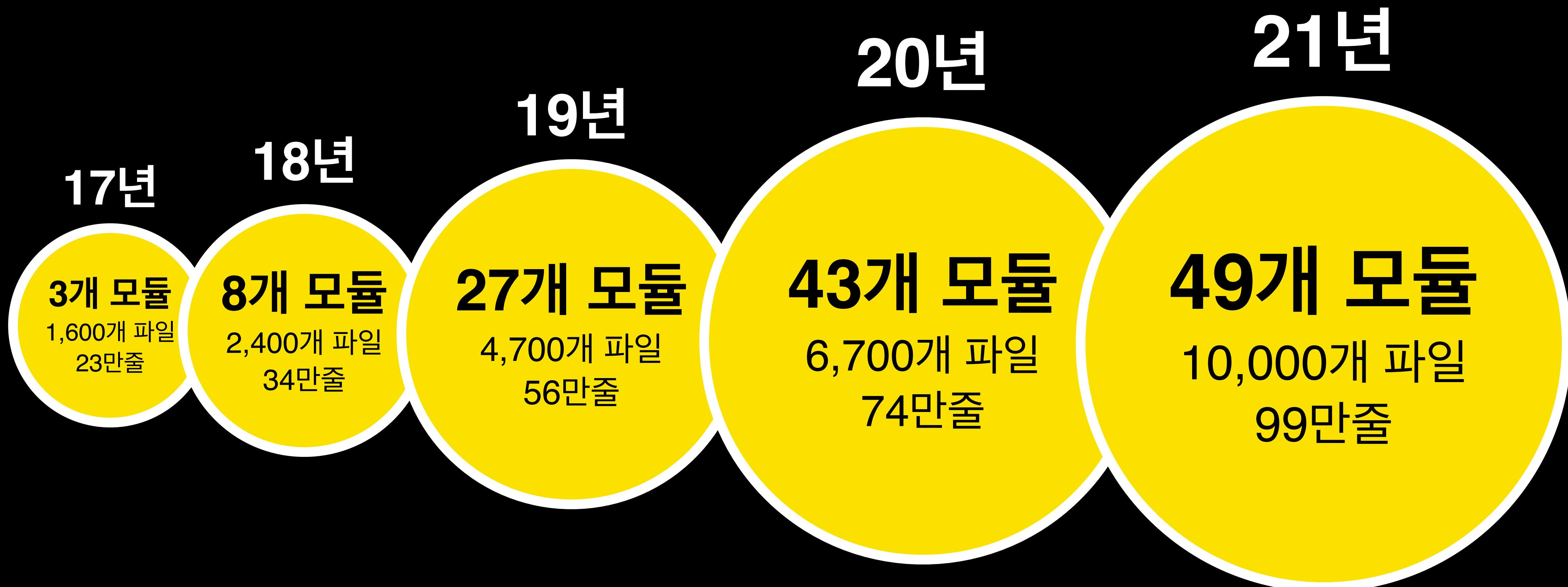
# 프로젝트 규모



# 프로젝트 규모

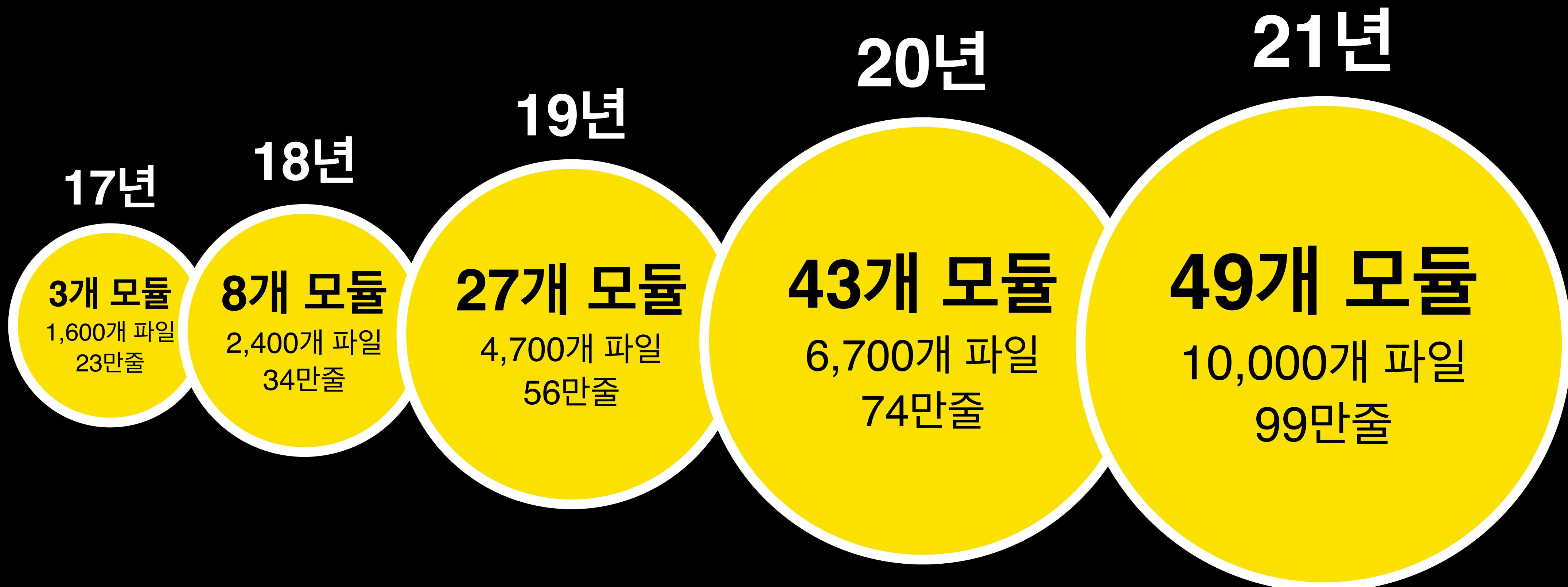


# 프로젝트 규모



17~21년 12.31 기준

# 프로젝트 규모



17~21년 12.31 기준

## PROJECT

 Application

## TARGETS

 Application

## Frameworks, Libraries, and Embedded Content

Name	Filters	Embed
 AdSupport.framework	Always Used	 Do Not Embed ▾
 Alamofire.xcframework	Always Used	 Embed & Sign ▾
 AVFoundation.framework	Always Used	 Do Not Embed ▾
 Combine.framework	Always Used	 Do Not Embed ▾
 Core.framework	iOS	 Embed & Sign ▾
 CryptoKit.framework	Always Used	 Do Not Embed ▾
 Feature.framework	iOS	 Embed & Sign ▾
 FeatureAuth.framework	iOS	 Embed & Sign ▾
 FeatureDeposit.framework	iOS	 Embed & Sign ▾
 FeatureLoan.framework	iOS	 Embed & Sign ▾
 FirebaseAnalytics.xcframework	Always Used	 Do Not Embed ▾
 FirebaseCore.xcframework	Always Used	 Do Not Embed ▾
 FirebaseCoreDiagnostics.xcframework	Always Used	 Do Not Embed ▾
 FirebaseCrashlytics.xcframework	Always Used	 Do Not Embed ▾
 FirebaseInstallations.xcframework	Always Used	 Do Not Embed ▾
 GoogleAppMeasurement.xcframework	Always Used	 Do Not Embed ▾
 GoogleDataTransport.xcframework	Always Used	 Do Not Embed ▾
 GoogleUtilities.xcframework	Always Used	 Do Not Embed ▾
 JavaScriptCore.framework	Always Used	 Do Not Embed ▾

## PROJECT

 Application

## TARGETS

 Application

## Frameworks, Libraries, and Embedded Content

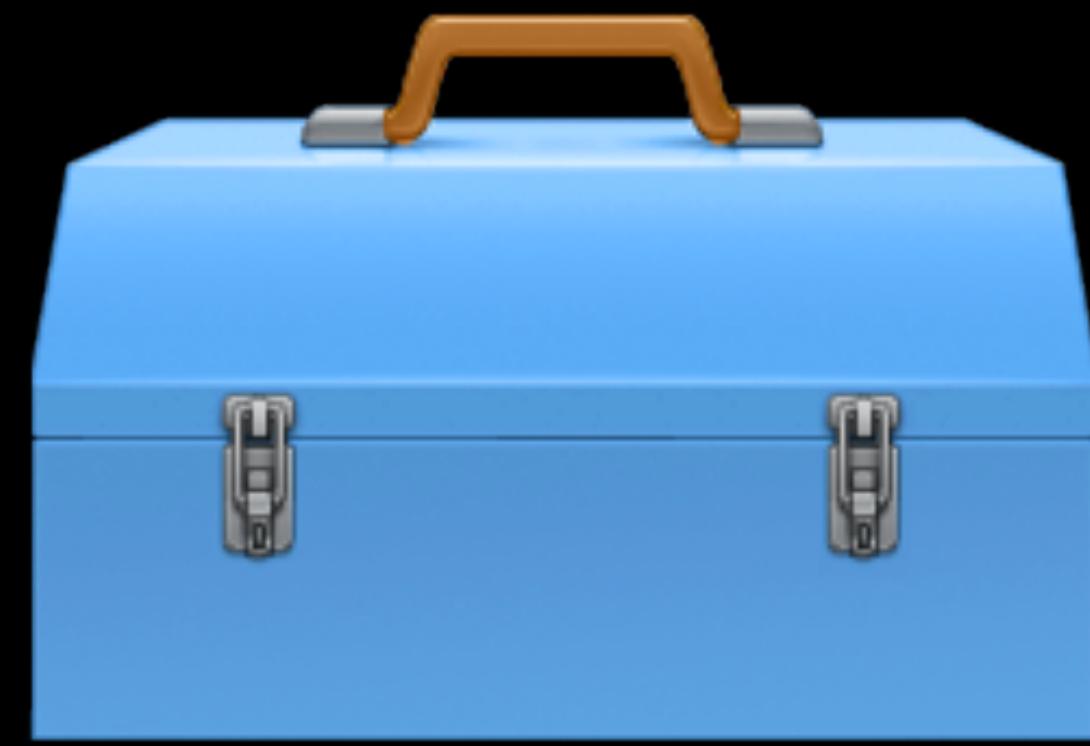
Name	Filters	Embed
 AdSupport.framework	Always Used	 Do Not Embed ▾
 Alamofire.xcframework	Always Used	 Embed & Sign ▾
 AVFoundation.framework	Always Used	 Do Not Embed ▾
 Combine.framework	Always Used	 Do Not Embed ▾
 Core.framework	iOS	 Embed & Sign ▾
 CryptoKit.framework	Always Used	 Do Not Embed ▾
 Feature.framework	iOS	 Embed & Sign ▾
 FeatureAuth.framework	iOS	 Embed & Sign ▾
 FeatureDeposit.framework	iOS	 Embed & Sign ▾
 FeatureLoan.framework	iOS	 Embed & Sign ▾
 FirebaseAnalytics.xcframework	Always Used	 Do Not Embed ▾
 FirebaseCore.xcframework	Always Used	 Do Not Embed ▾
 FirebaseCoreDiagnostics.xcframework	Always Used	 Do Not Embed ▾
 FirebaseCrashlytics.xcframework	Always Used	 Do Not Embed ▾
 FirebaseInstallations.xcframework	Always Used	 Do Not Embed ▾
 GoogleAppMeasurement.xcframework	Always Used	 Do Not Embed ▾
 GoogleDataTransport.xcframework	Always Used	 Do Not Embed ▾
 GoogleUtilities.xcframework	Always Used	 Do Not Embed ▾
 JavaScriptCore.framework	Always Used	 Do Not Embed ▾

# 도구 사용

복잡한 설정 관리

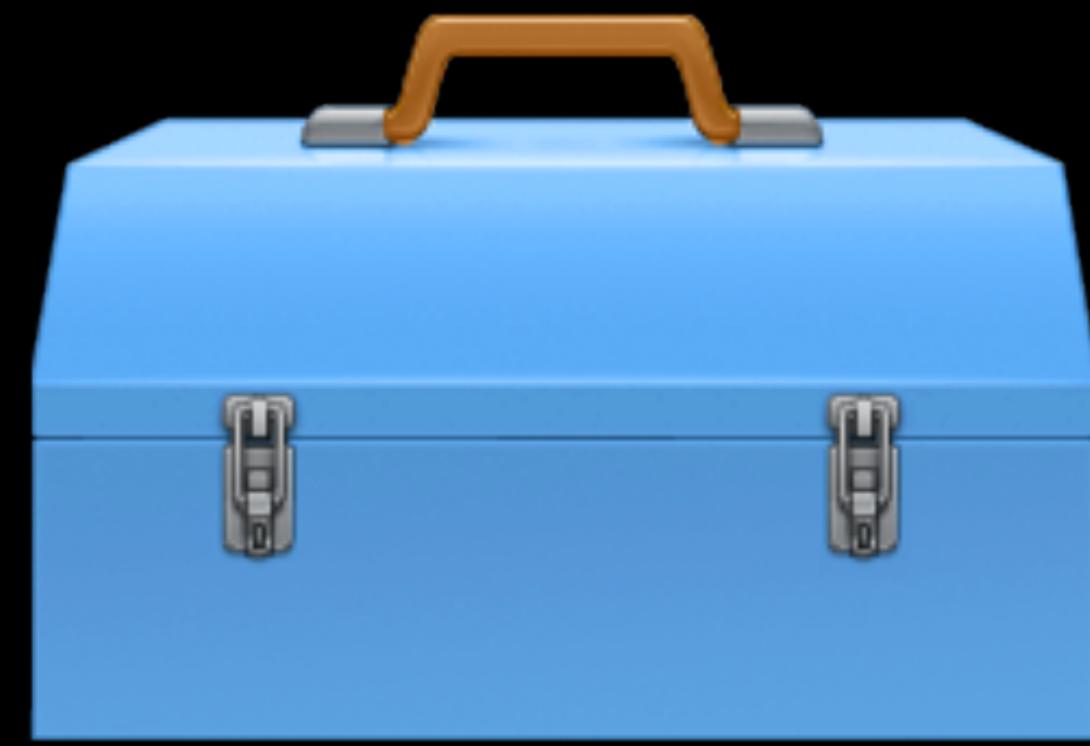
# 템플릿

프로젝트 생성 및 규격화



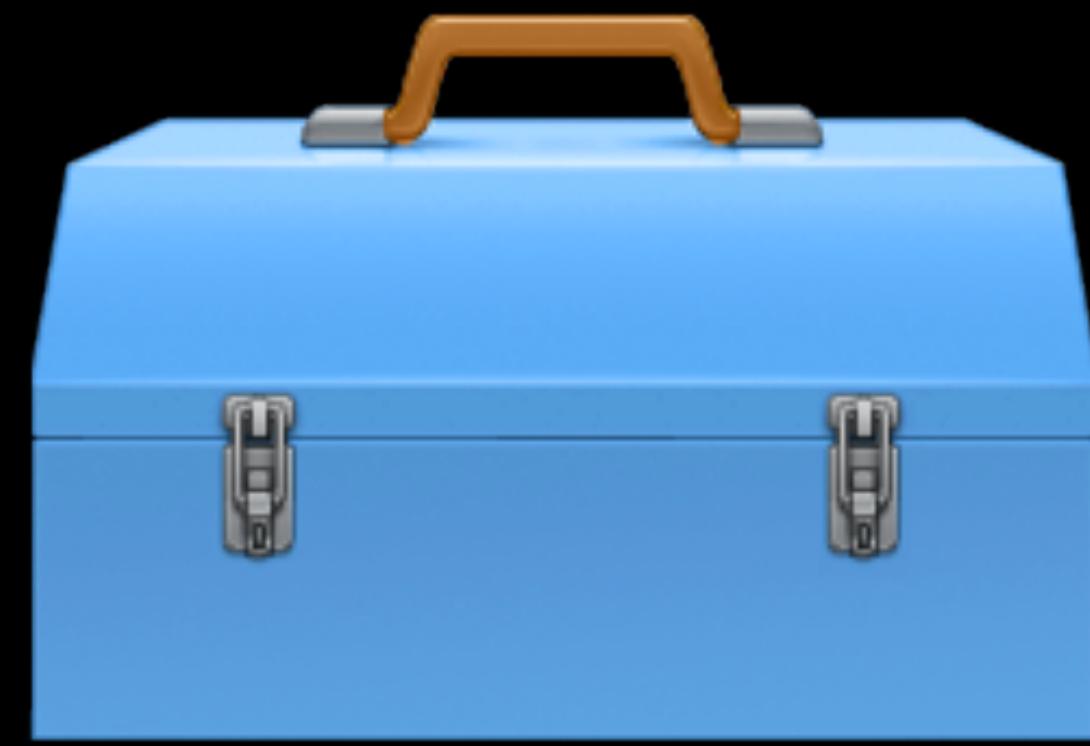
# 모듈 110개

2022.08.31 기준



# 모듈 110개

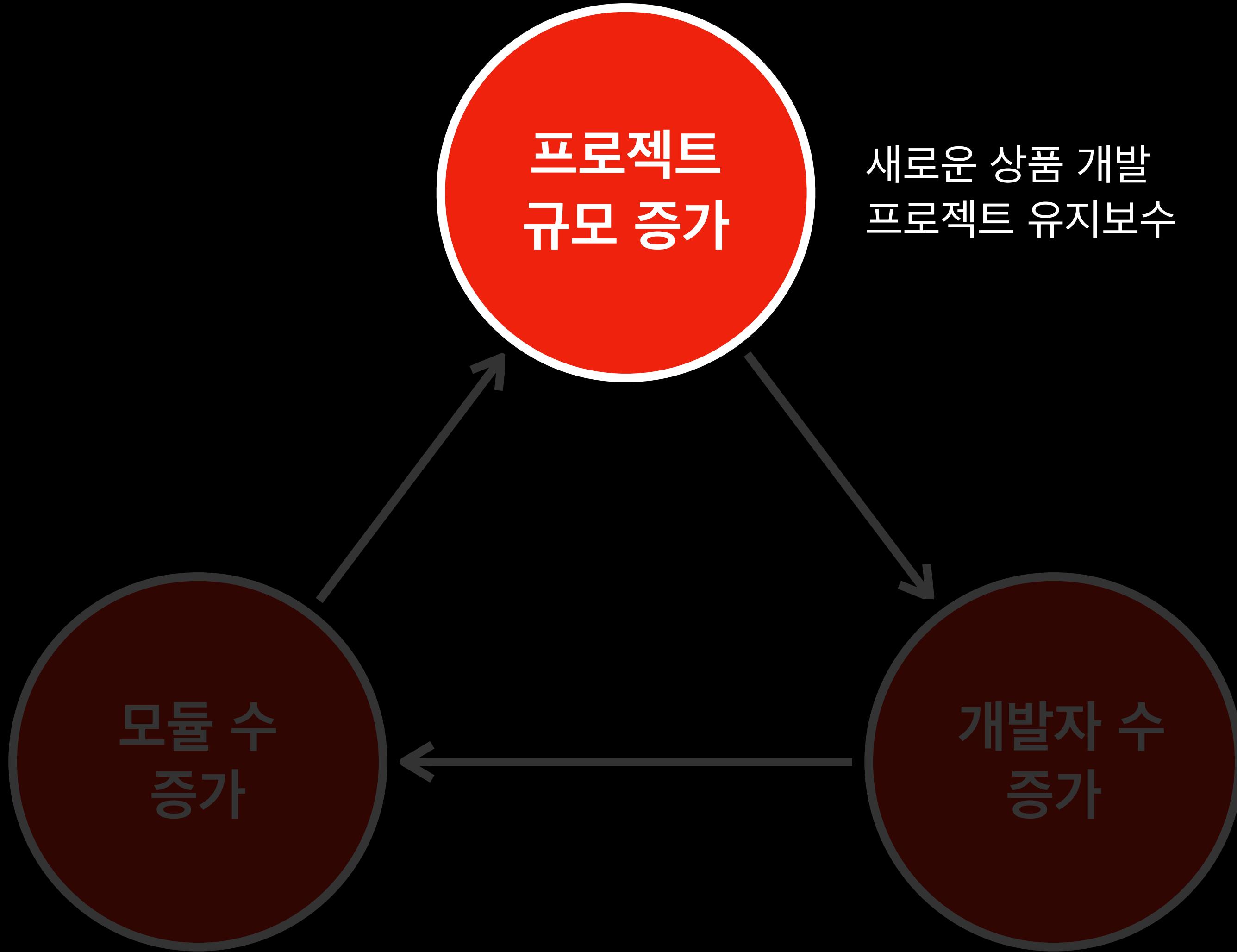
2022.08.31 기준



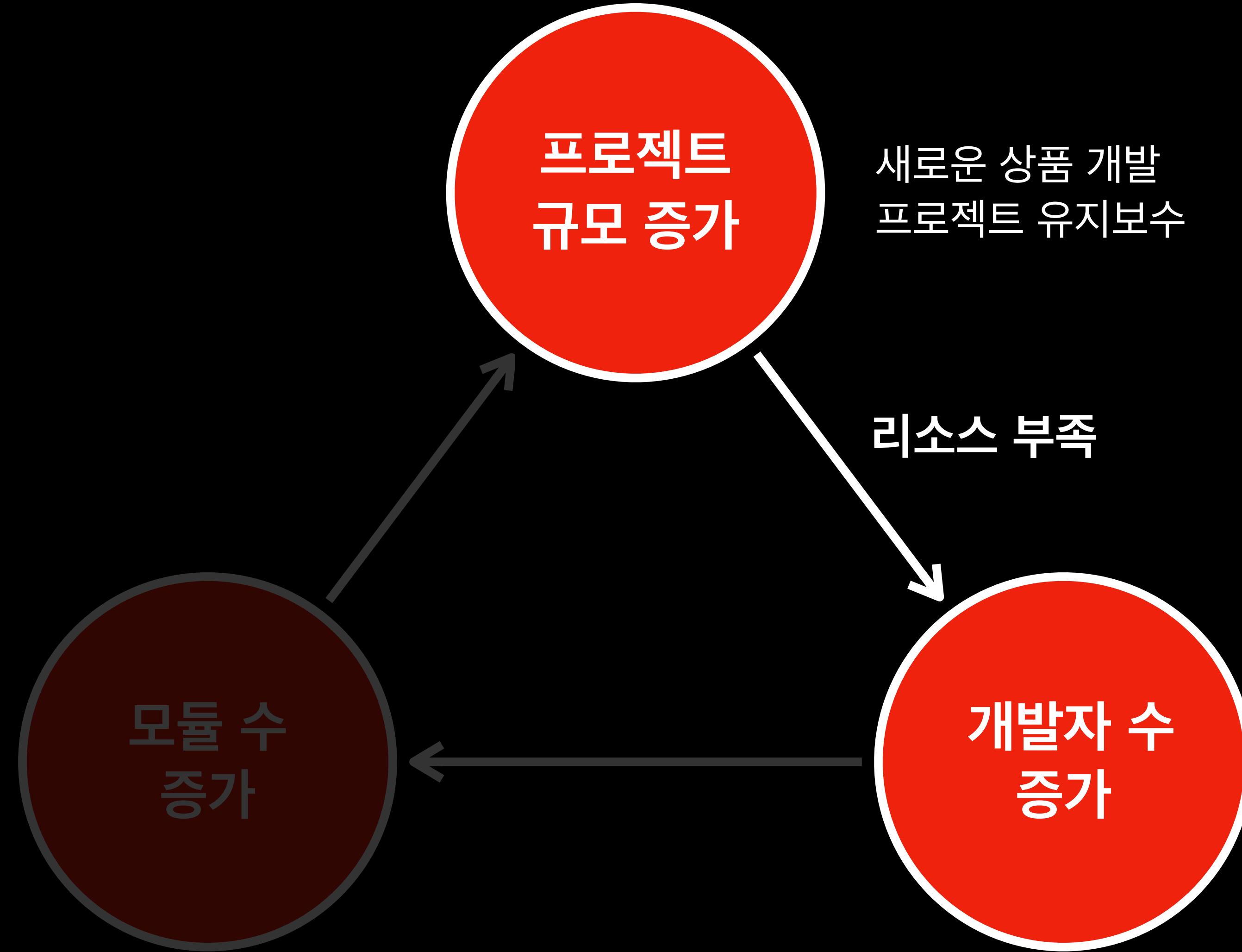
# 모듈 110개

2022.08.31 기준

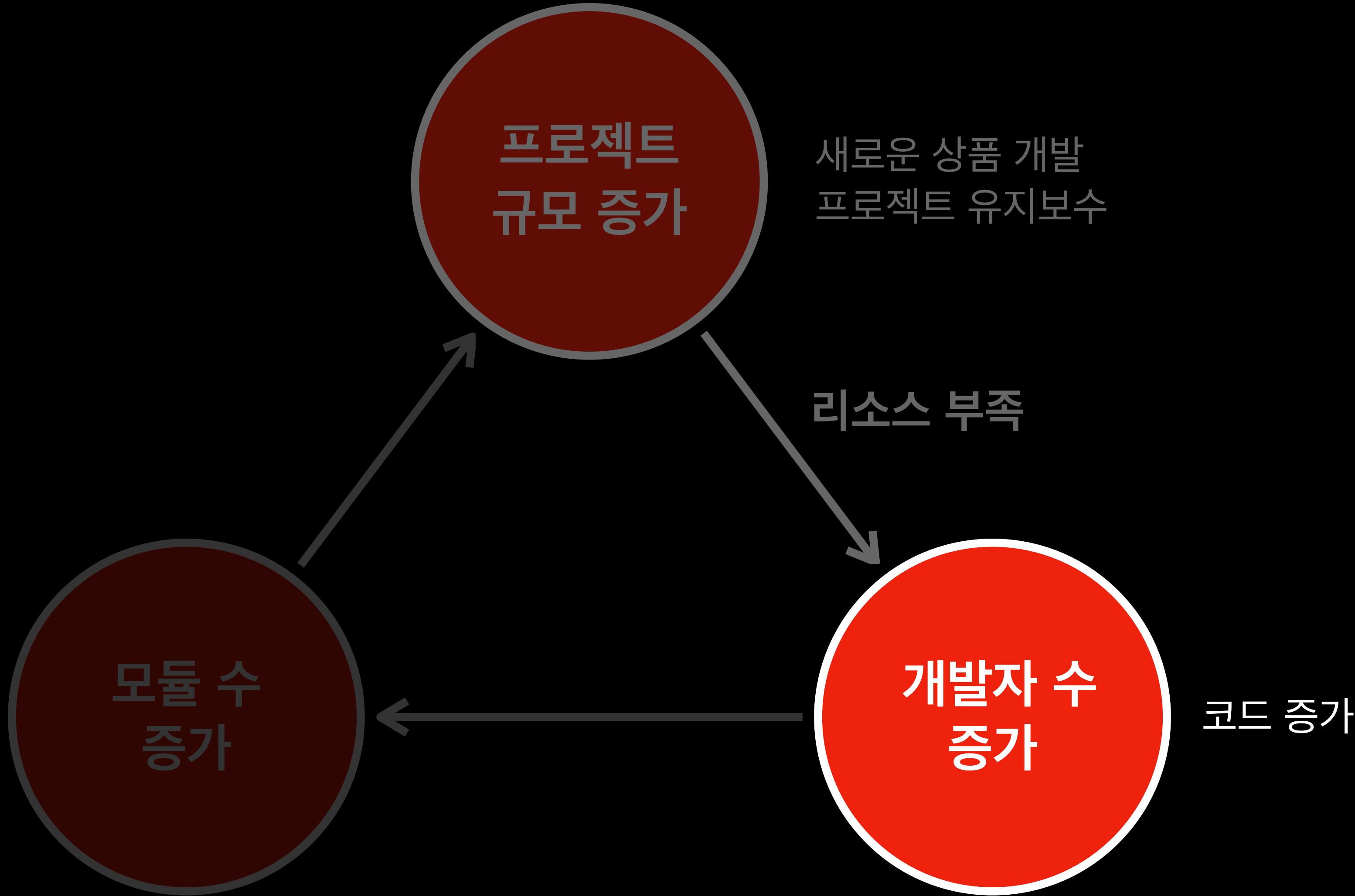
# 프로젝트 규모



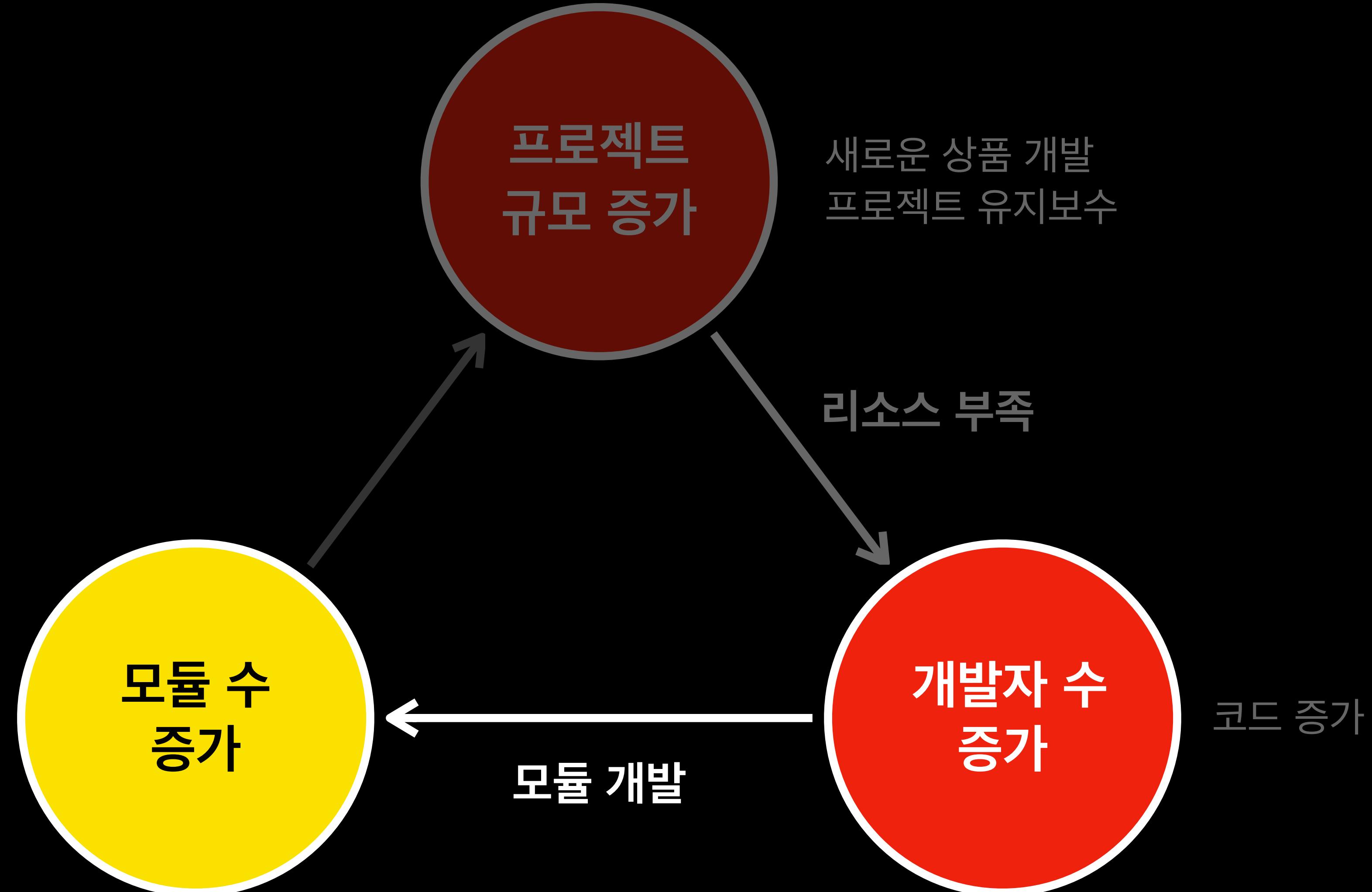
# 프로젝트 규모



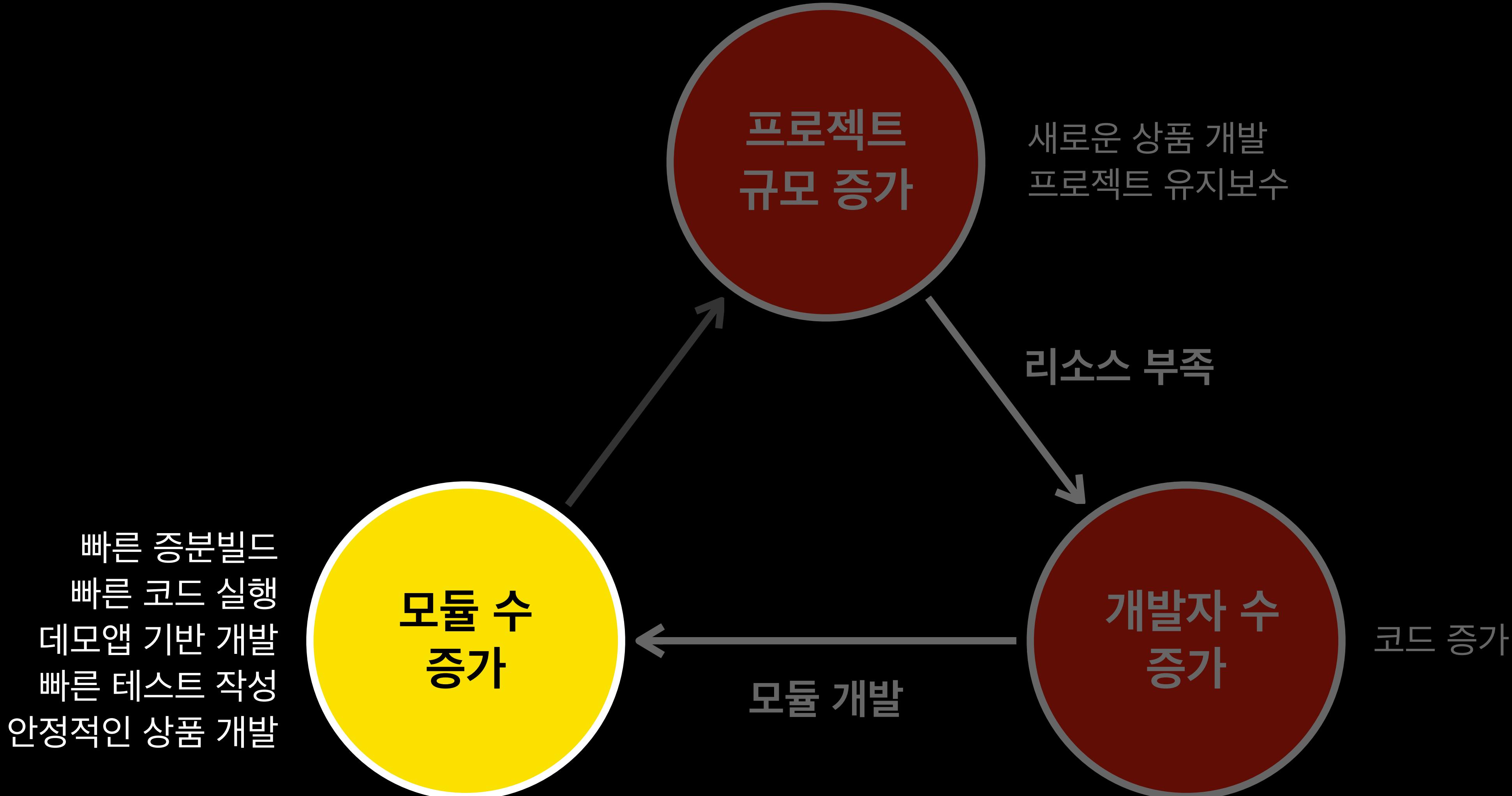
# 프로젝트 규모



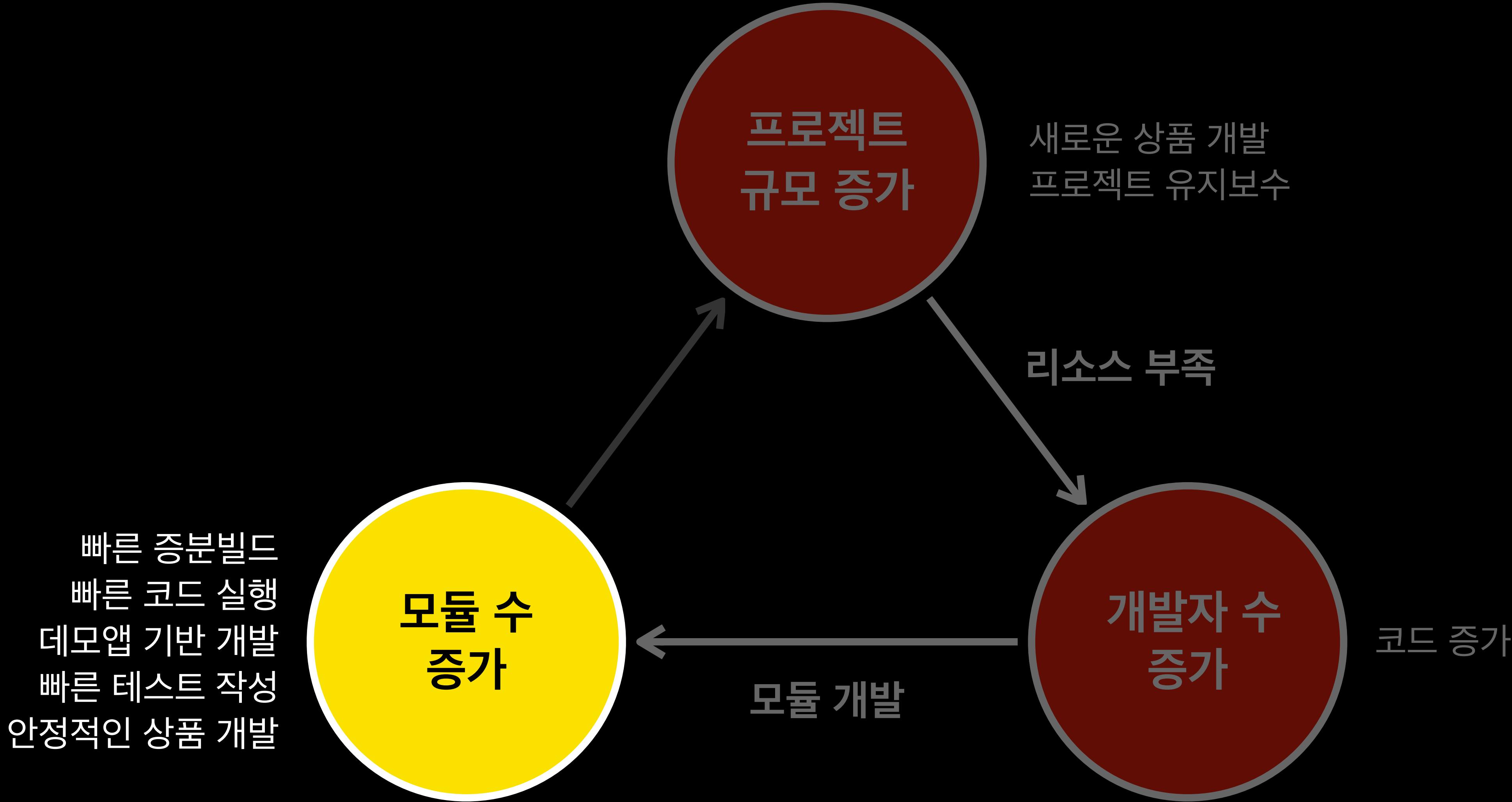
# 프로젝트 규모



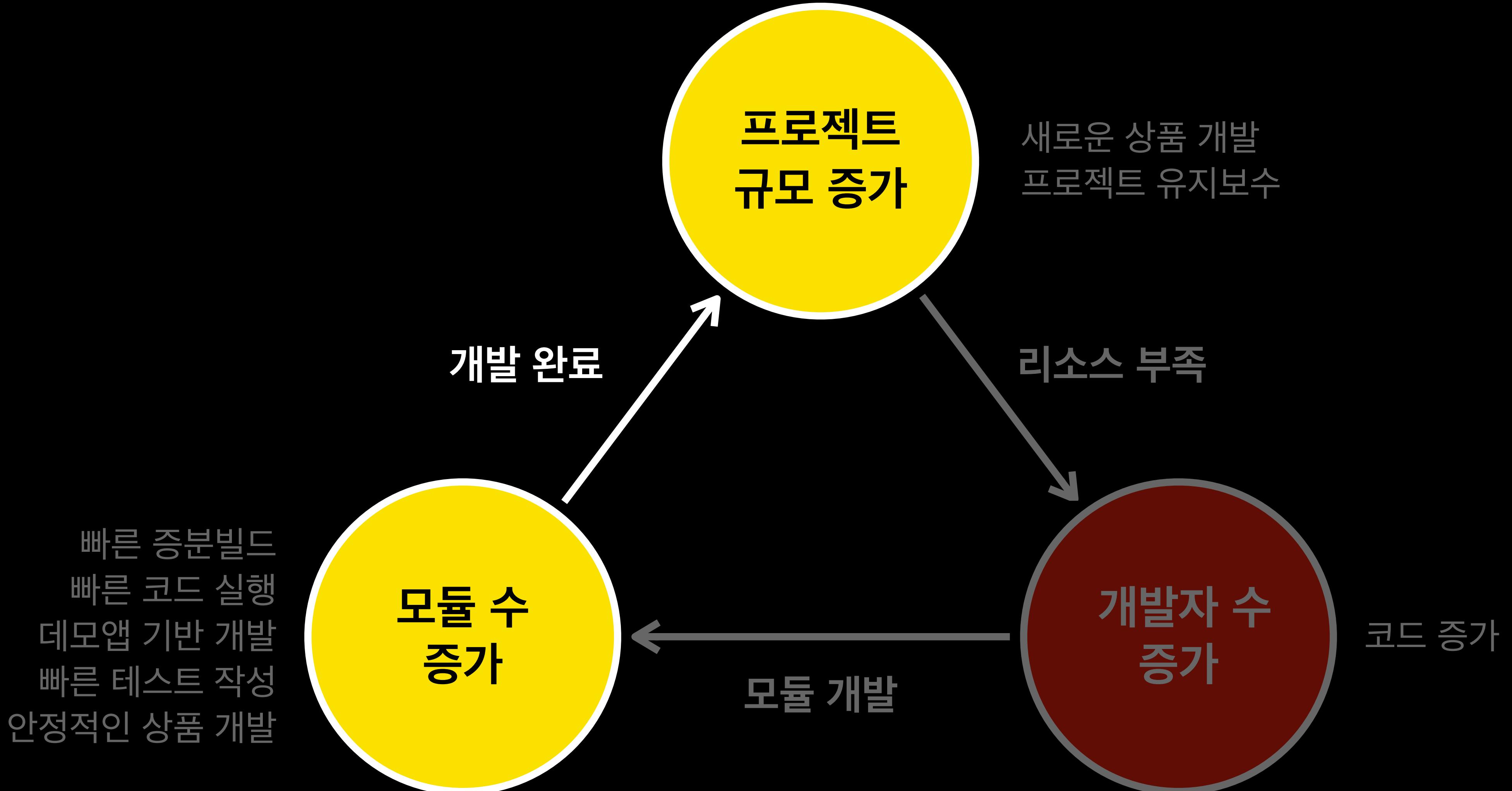
# 프로젝트 규모



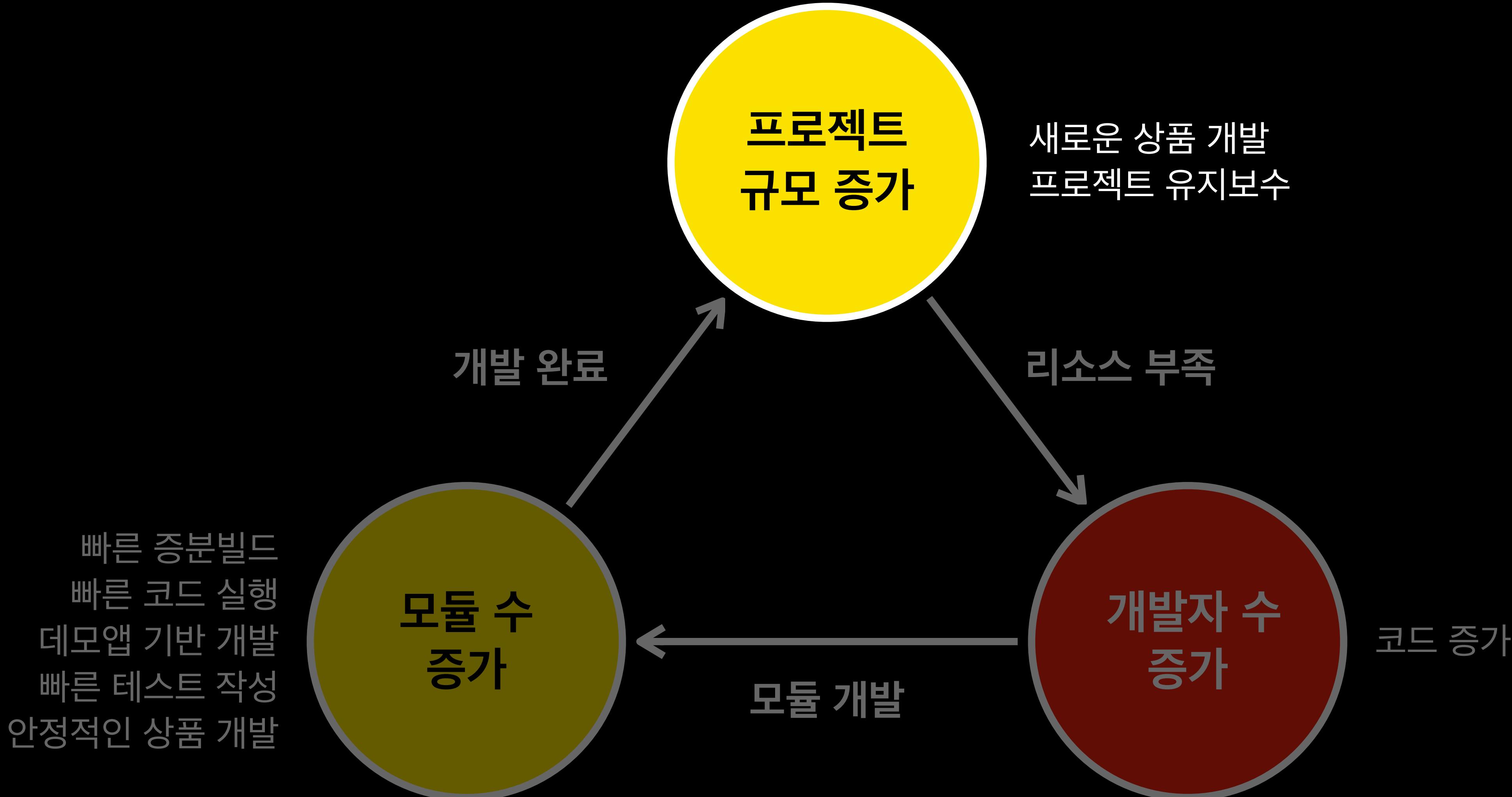
# 프로젝트 규모



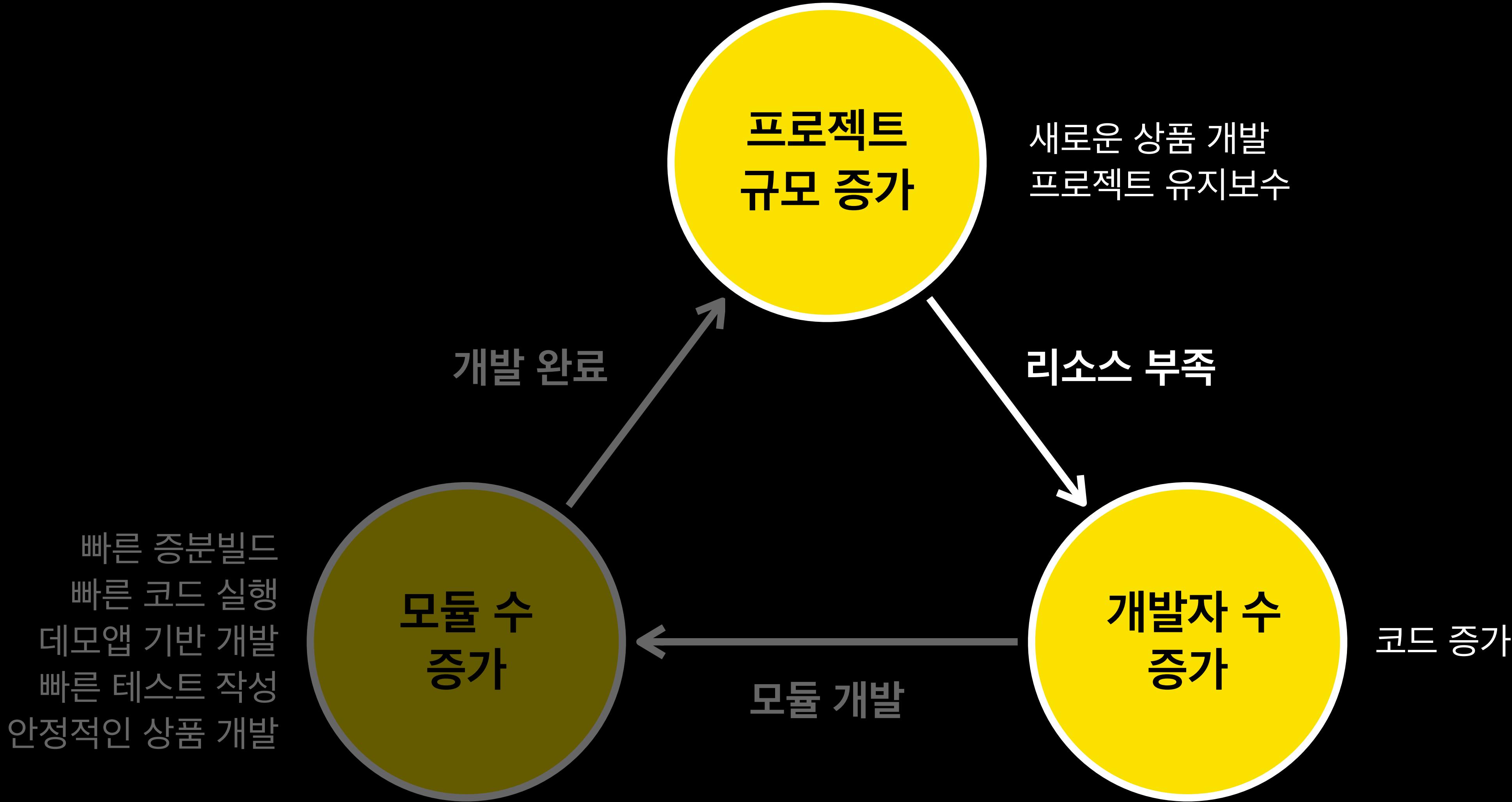
# 프로젝트 규모



# 프로젝트 규모



# 프로젝트 규모





# 모듈 조립

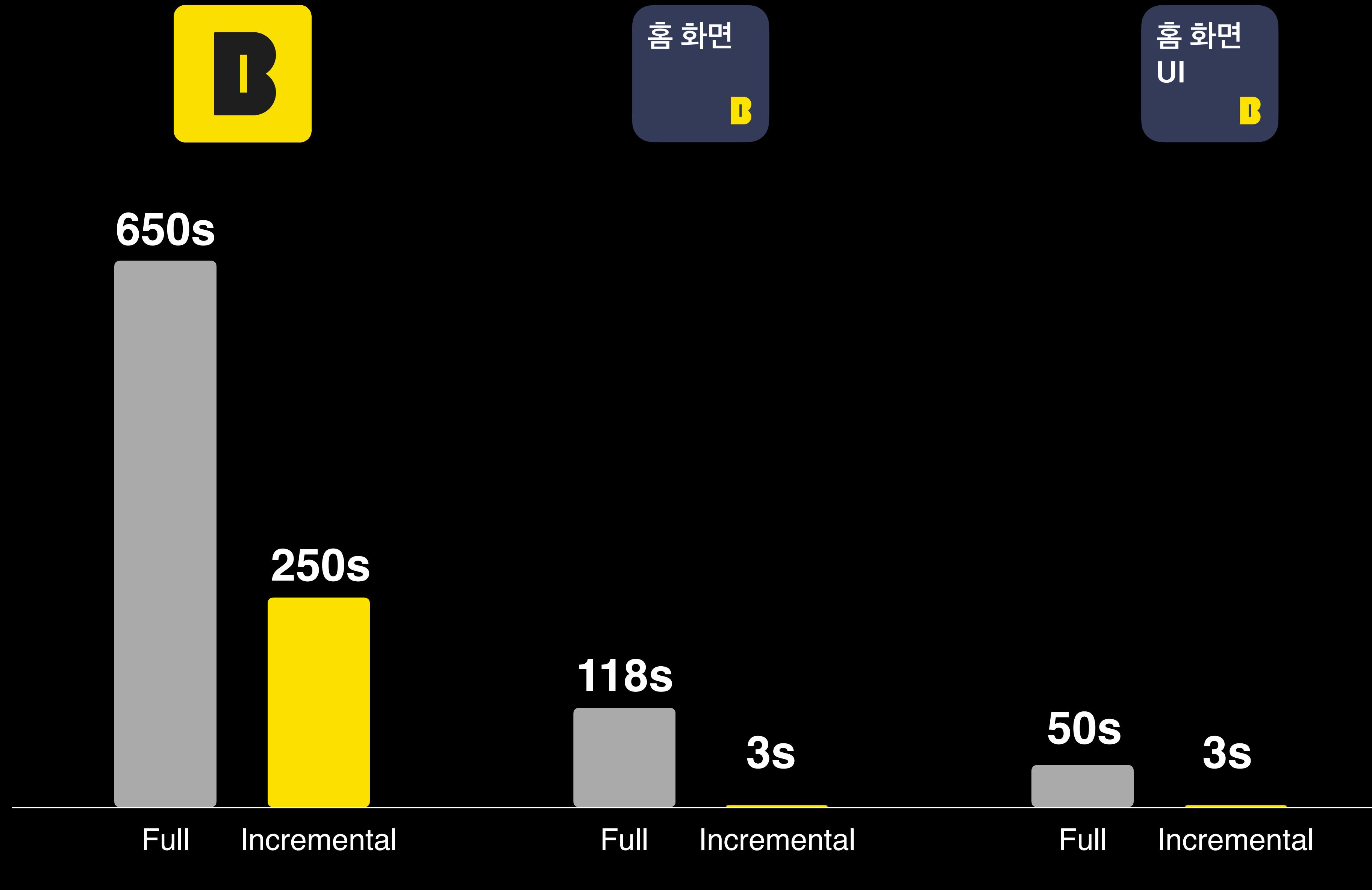
안정적인 기능, 상품 개발

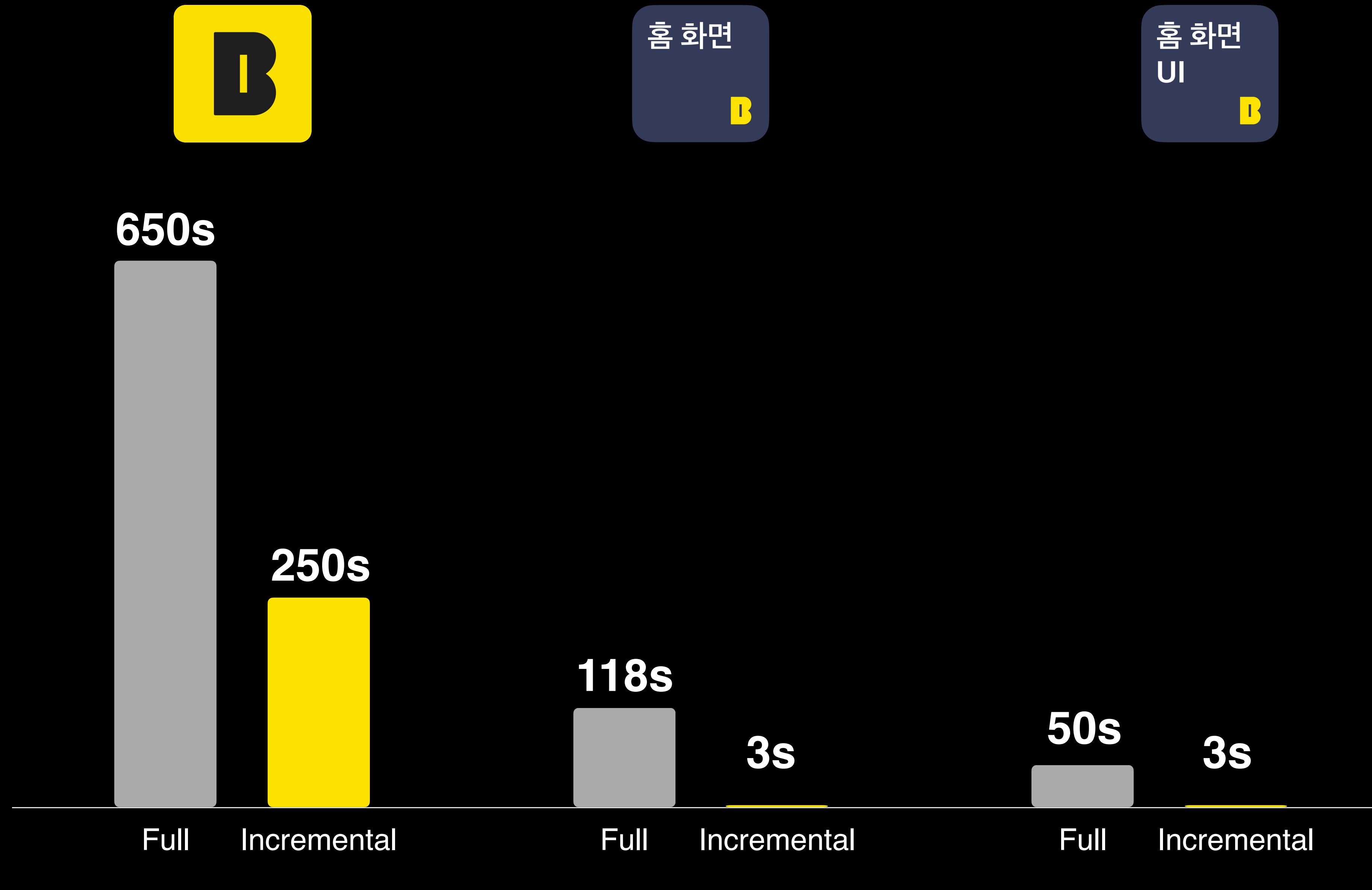
# 모듈 조립

안정적인 기능, 상품 개발

# 빠른 빌드

UI, 비즈니스 로직 개발 및 실행



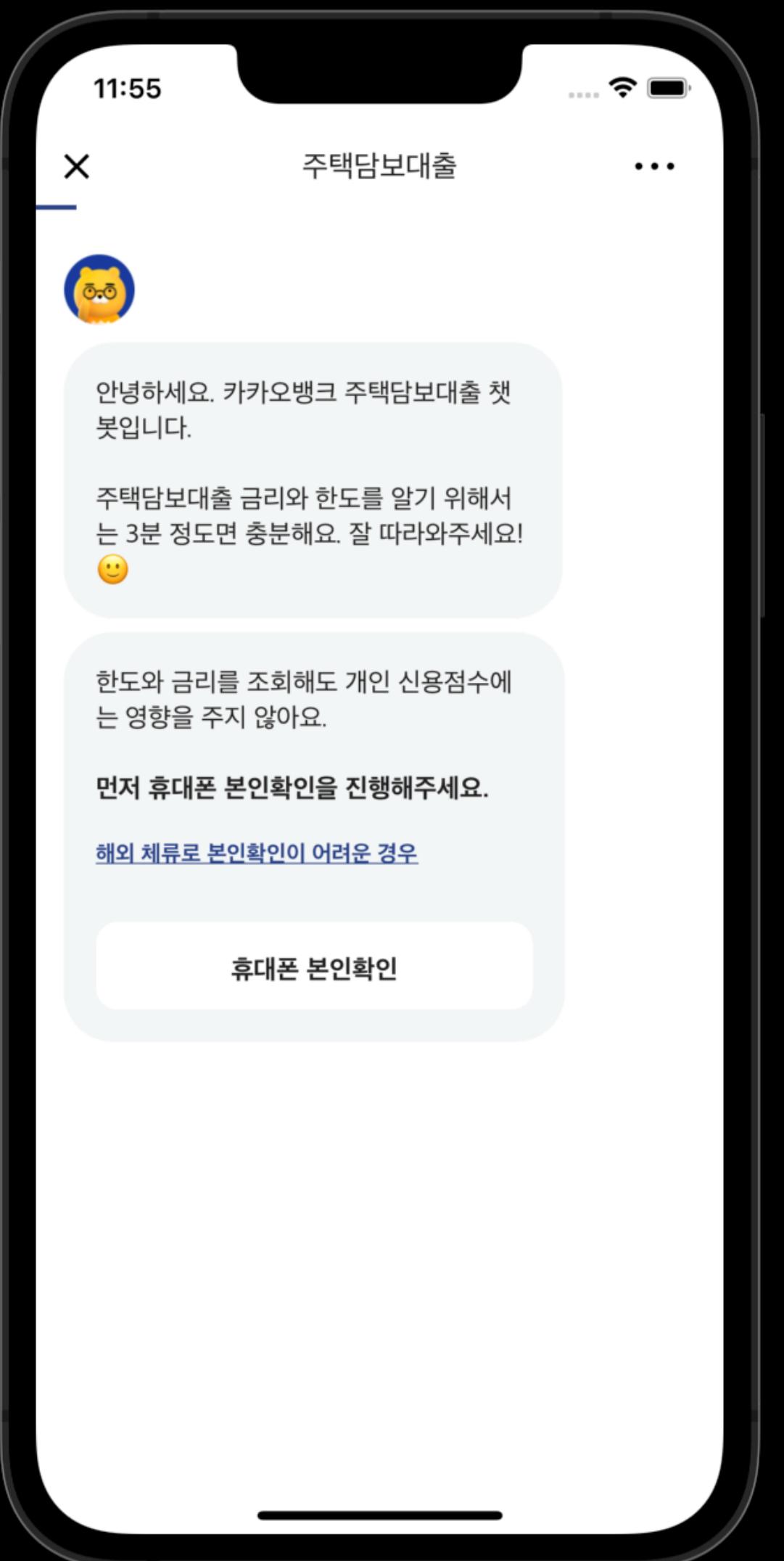


프로젝트 여정

개발 생산성 개선 방법 알아보기

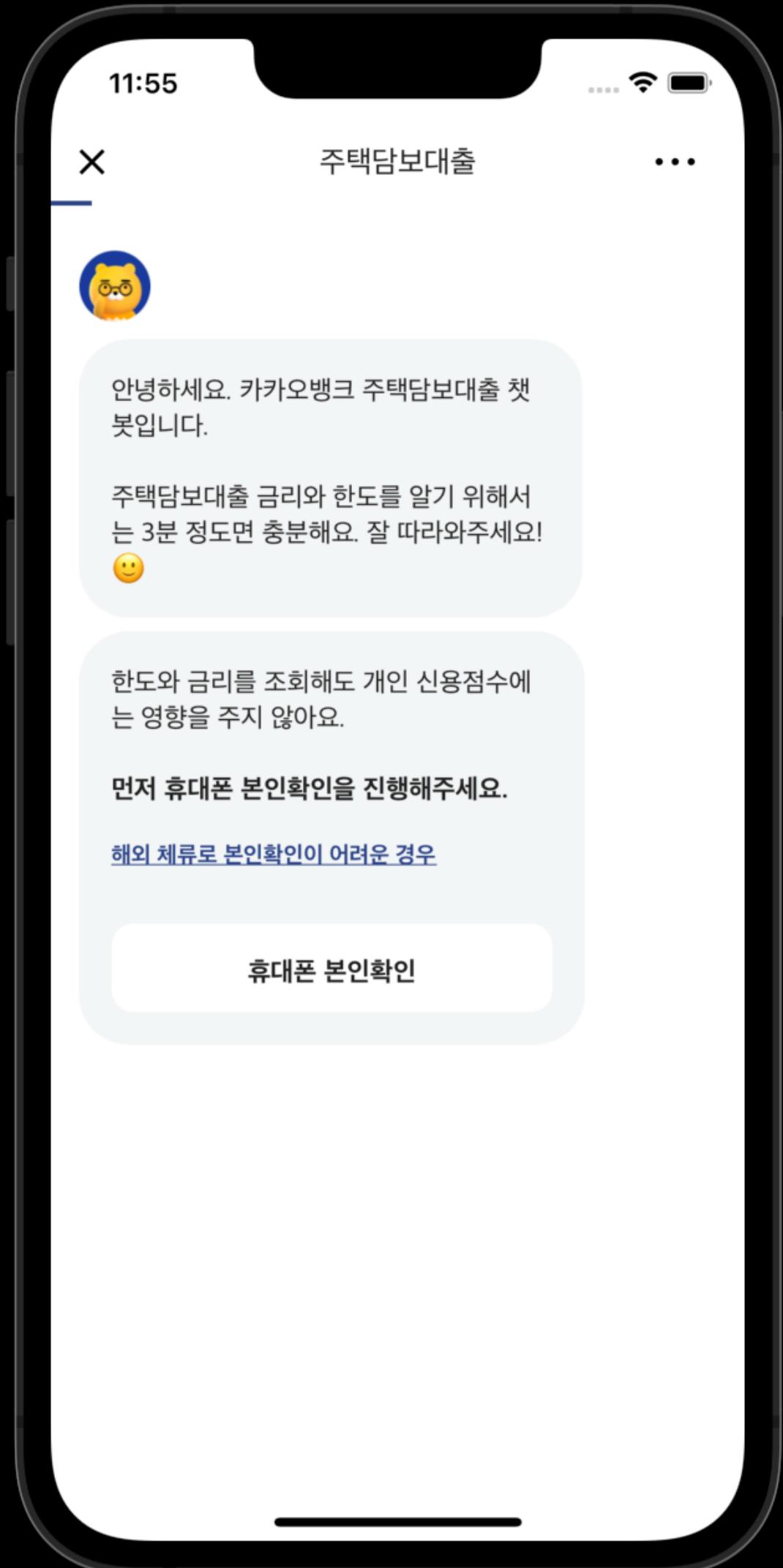
카카오뱅크 모듈 아키텍처 설계

# 상품에 필요한 기능



# 상품에 필요한 기능

UI

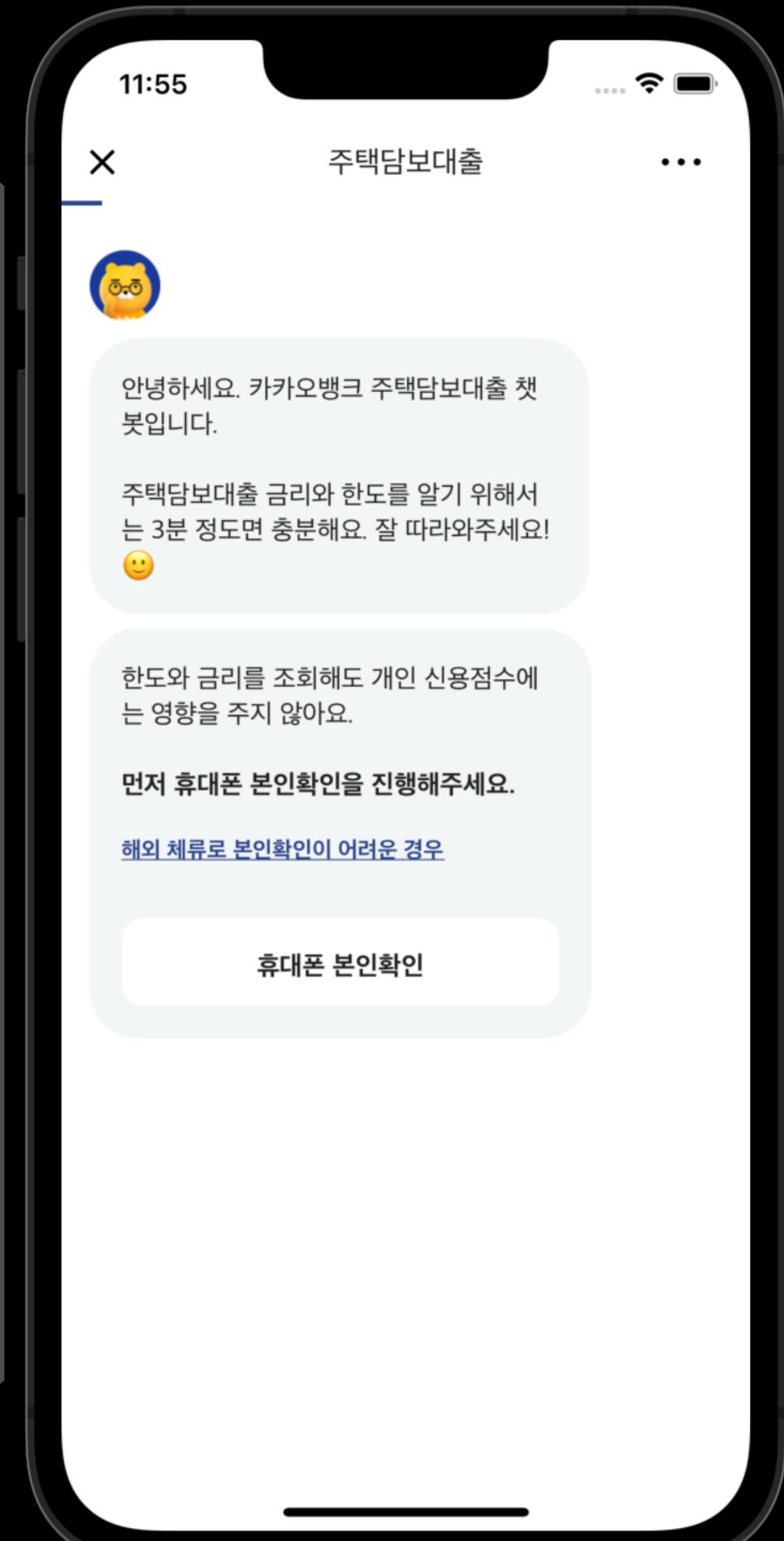


Business

# 상품에 필요한 기능

UI

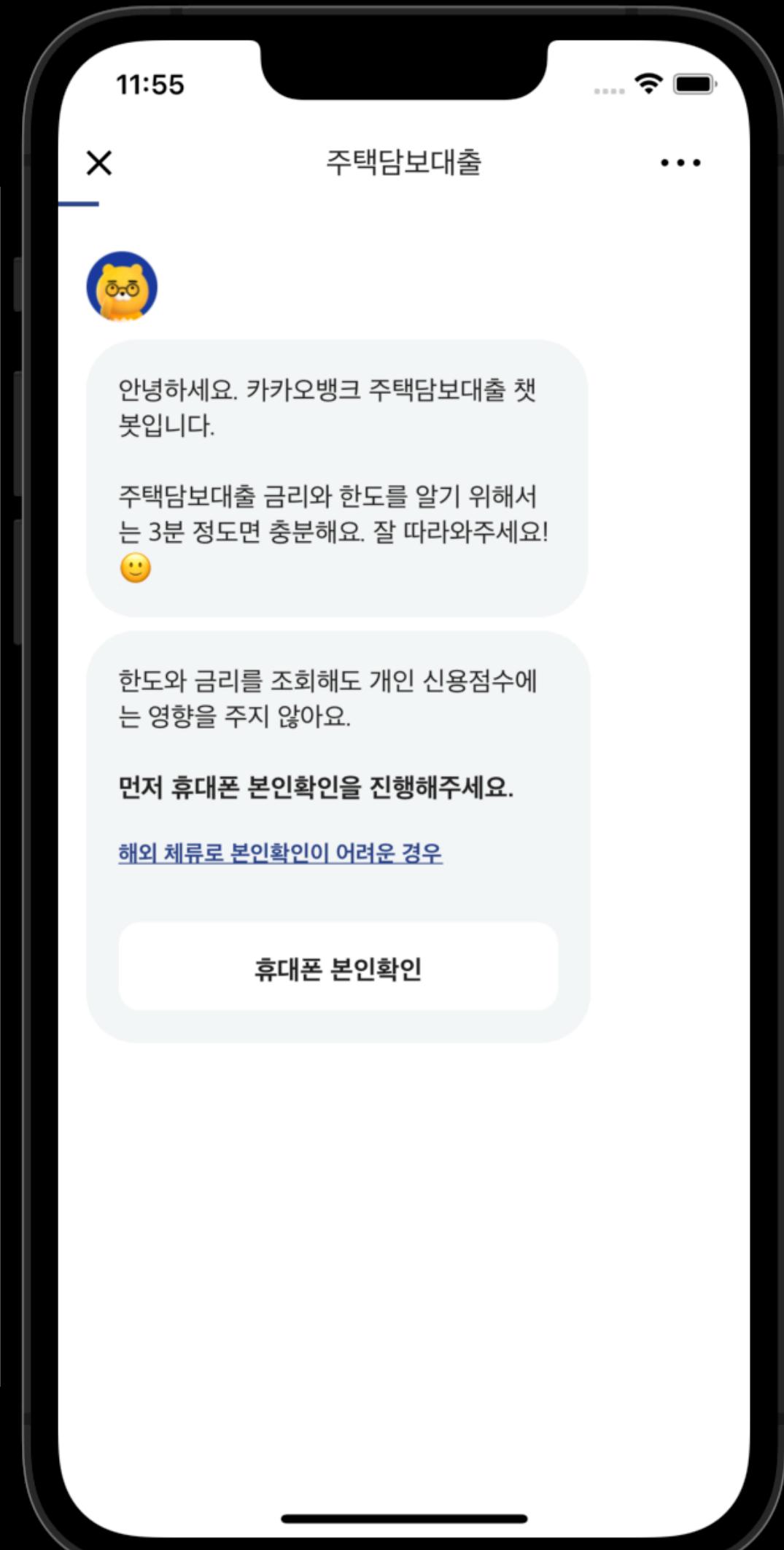
Business



# 상품에 필요한 기능

UI

Business



# 상품에 필요한 기능

UI

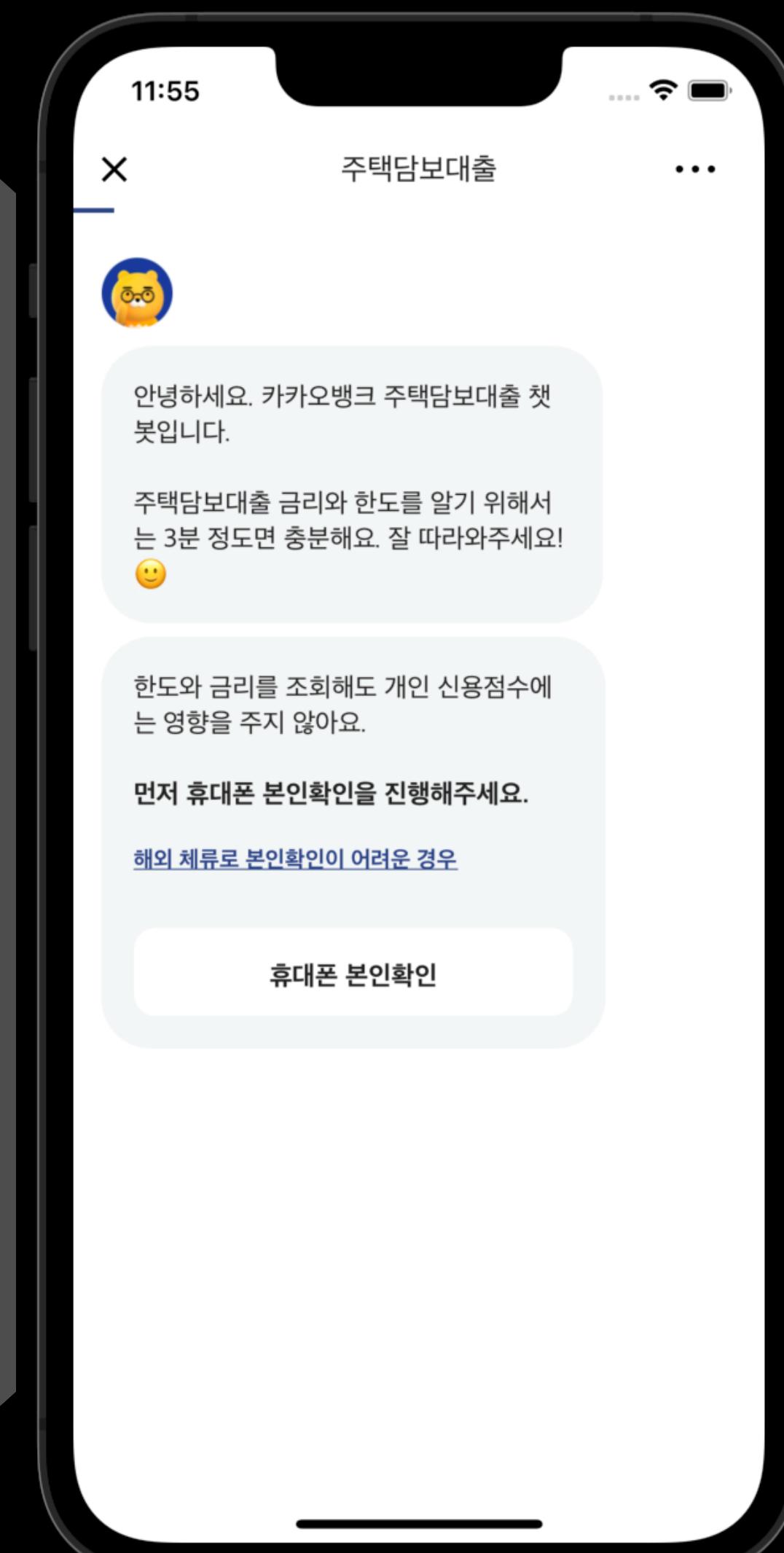
UI  
Component

UIHelper

Popup

Resource

Business



# 상품에 필요한 기능

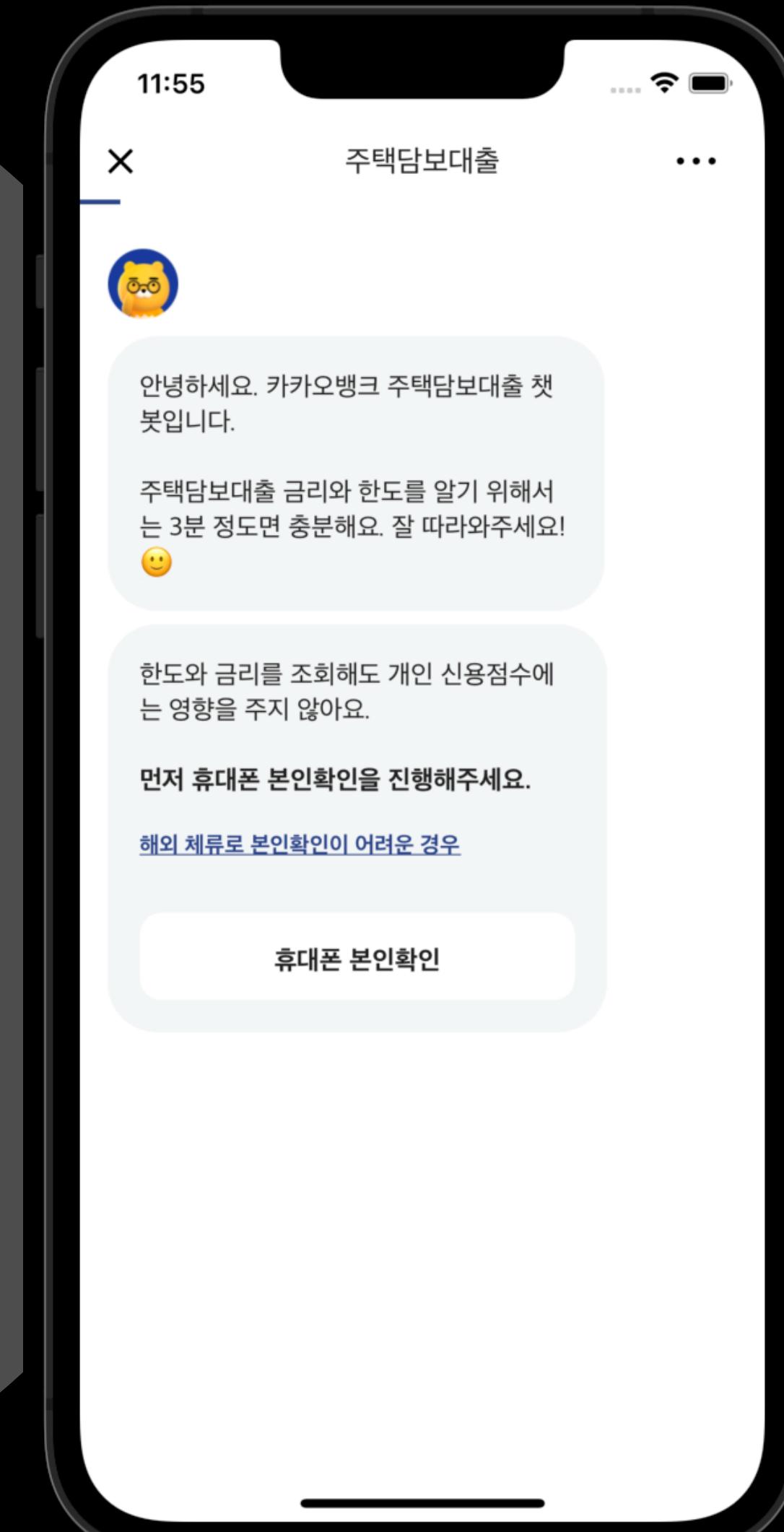
UI

UI  
Component

UIHelper

Popup

Resource



Business

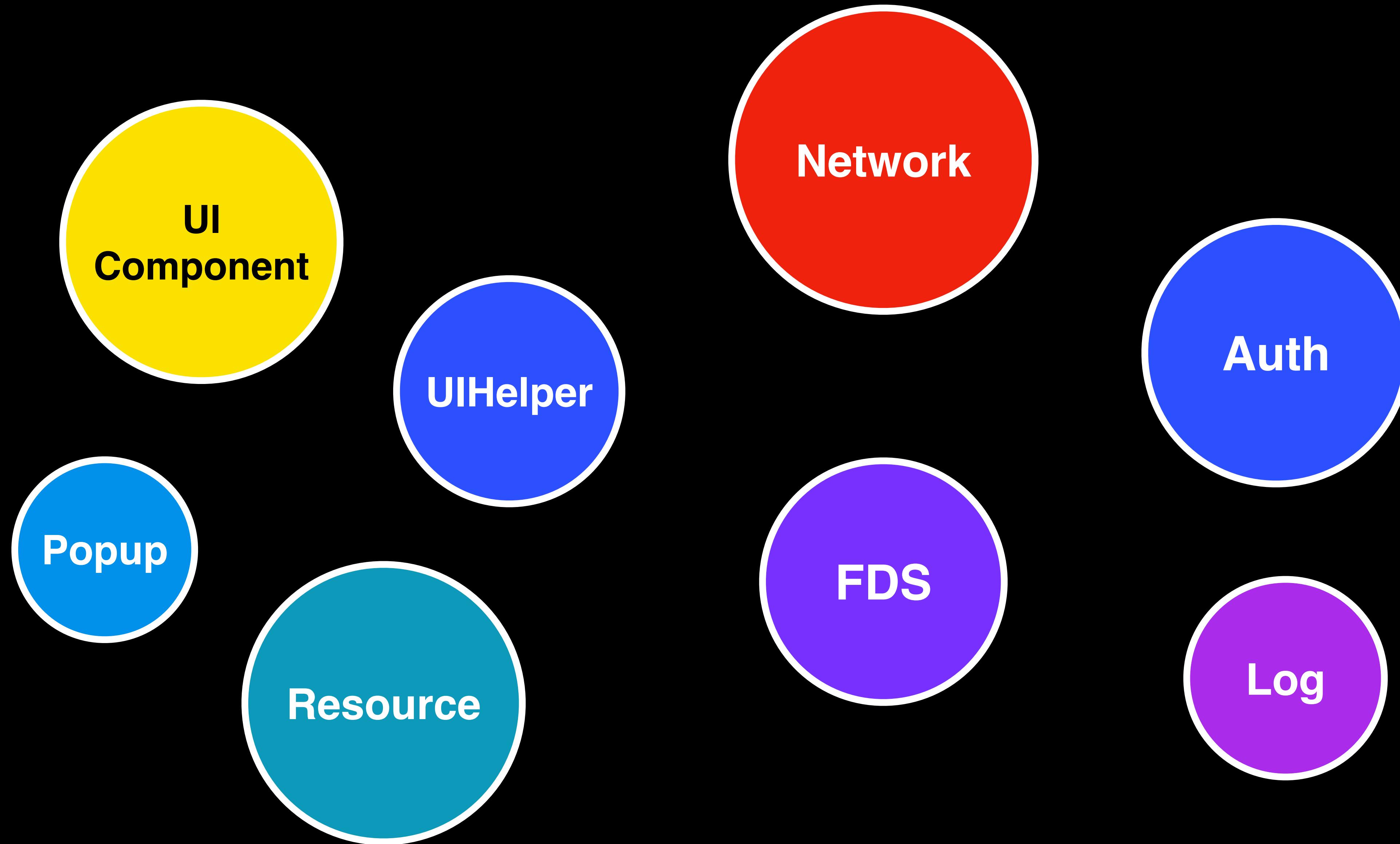
Network

Auth

FDS

Log

# 상품에 필요한 기능



# 모듈 분리가 어려운 이유

# 모듈 분리가 어려운 이유

- 프로젝트, 프레임워크를 수동 생성 및 관리

# 모듈 분리가 어려운 이유

- 프로젝트, 프레임워크를 수동 생성 및 관리

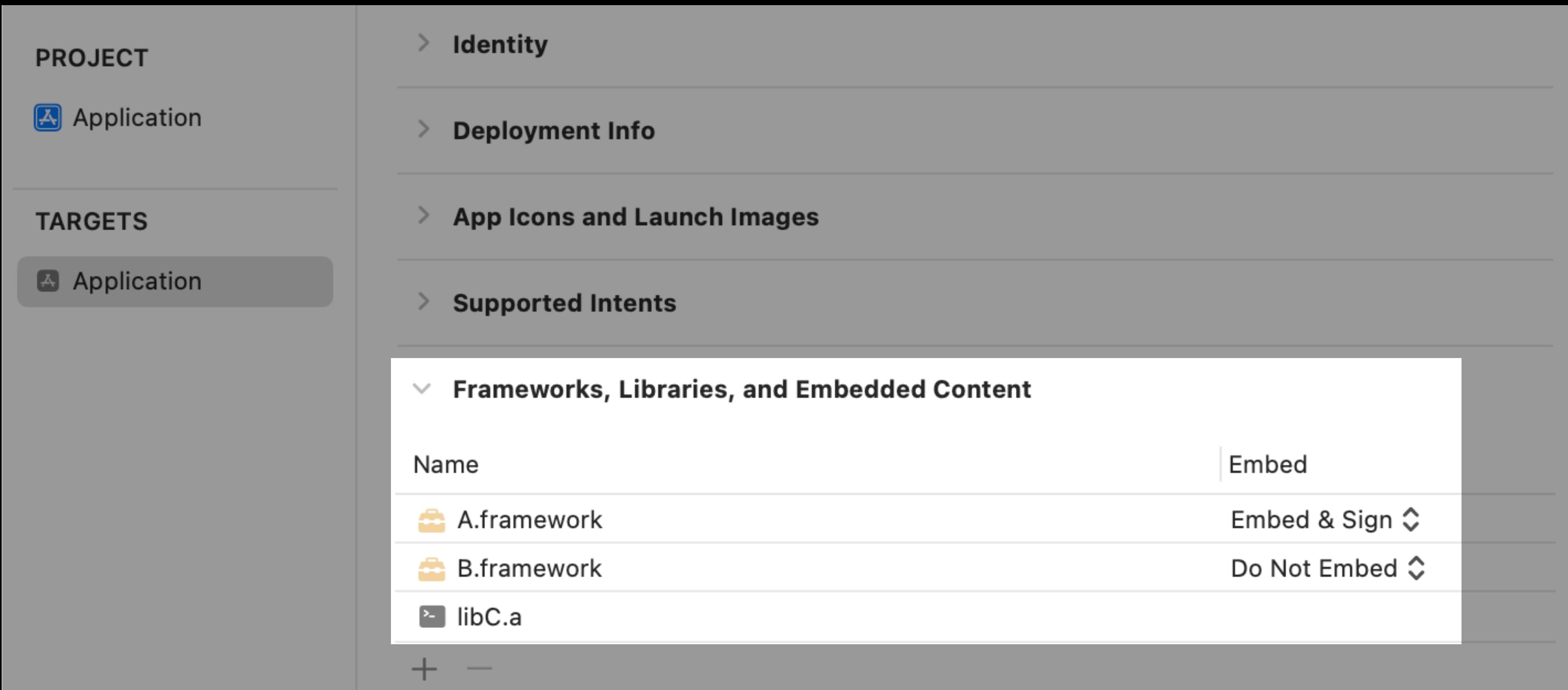
The screenshot shows the Xcode interface with the Project Navigator open. On the left, under PROJECT, there is an Application target. Under TARGETS, there is also an Application target, which is highlighted with a grey background. In the main pane, under the 'Frameworks, Libraries, and Embedded Content' section, there is a table:

Name	Embed
A.framework	Embed & Sign ▾
B.framework	Do Not Embed ▾
libC.a	

At the bottom of the table, there are '+' and '-' buttons for adding or removing frameworks.

# 모듈 분리가 어려운 이유

- 프로젝트, 프레임워크를 수동 생성 및 관리



# 모듈 분리가 어려운 이유

- 프로젝트, 프레임워크를 수동 생성 및 관리

The screenshot shows the 'Frameworks, Libraries, and Embedded Content' section of the Xcode project settings. The 'PROJECT' tab is selected at the top left, and the 'Identity' tab is selected under it. The main area displays three entries:

Name	Embed
A.framework	Embed & Sign ▾
B.framework	Do Not Embed ▾
libC.a	

Below the table, there is a toolbar with a '+' button and a '-' button.

# 모듈 분리가 어려운 이유

- 프로젝트, 프레임워크를 수동 생성 및 관리

The screenshot shows the 'Frameworks, Libraries, and Embedded Content' section of the Xcode project settings. The 'PROJECT' tab is selected at the top left, and the 'Identity' tab is selected under it. The main area displays three entries:

Name	Embed
A.framework	Embed & Sign ▾
B.framework	Do Not Embed ▾
libC.a	

Below the table, there is a toolbar with a minus sign and a plus sign.

# 모듈 분리가 어려운 이유

- 프로젝트, 프레임워크를 수동 생성 및 관리
  - 애플리케이션에서 동적 프레임워크를 **Embed** 하지 않으면 런타임 에러 발생

```
dyld: Library not loaded: @rpath/ModuleB.framework/ModuleB  
Referenced from: /private/var/containers/Bundle/Application/D00C1442-17A1-4D39-9E94-1C4EF18DB688/  
Application.app/Application  
Reason: image not found
```

# 모듈 분리가 어려운 이유

- 프로젝트, 프레임워크를 수동 생성 및 관리
  - 애플리케이션에서 동적 프레임워크를 Embed 하지 않으면 런타임 에러 발생
  - 정적 프레임워크/라이브러리 사용시 코드가 중복 복사되는 문제

Class AFNetworking is implemented in both /private/var/containers/Bundle/Application/D00C1442-17A1-4D39-9E94-1C4EF18DB688/Application.app/Frameworks/ModuleA.framework/ModuleA (0x10000000) and /private/var/containers/Bundle/Application/D00C1442-17A1-4D39-9E94-1C4EF18DB688/Application.app/Frameworks/ModuleB.framework/ModuleB (0x10000000). One of the two will be used.  
Which one is undefined.

# 모듈 분리가 어려운 이유

- SwiftPM 제약사항
  - Configuration은 Debug, Release 만 제공
  - 여러 환경의 Configuration 설정은 어려움
    - Develop, Test, Stage, Production 등

```
.target(  
    name: "FeatureAuth",  
    swiftSettings: [  
        .define("DEBUG", .when(configuration: .debug)),  
        .define("RELEASE", .when(configuration: .release))  
    ])
```

# 모듈 분리가 어려운 이유

- SwiftPM 제약사항
  - Configuration은 Debug, Release 만 제공
  - 여러 환경의 Configuration 설정은 어려움
    - Develop, Test, Stage, Production 등

```
.target(  
    name: "FeatureAuth",  
    swiftSettings: [  
        .define("DEBUG", .when(configuration: .debug)),  
        .define("RELEASE", .when(configuration: .release))  
    ])
```

# 모듈 분리가 어려운 이유

- 오픈소스, 참고자료 등 부족



어려운 부분을 해결해줄 수 있는 도구가 있다면?

# 프로젝트 생성 도구 조건

# 프로젝트 생성 도구 조건

- Swift 언어로 Manifest 작성 가능할 것
- 프레임워크, 라이브러리의 Embed 설정 및 의존성 관리
- 조건에 따라 프로젝트, 워크스페이스 동적 생성
- 의존성 그래프 생성 및 가시화





# 프로젝트 생성 도구 조건

- 프레임워크, 라이브러리의 Embed 설정 및 의존성 관리
  - 동적 프레임워크, 정적 라이브러리 구분하여 처리

# 프로젝트 생성 도구 조건

- 프레임워크, 라이브러리의 Embed 설정 및 의존성 관리

The screenshot shows the 'Frameworks, Libraries, and Embedded Content' section of the Xcode project settings. The sidebar on the left shows 'PROJECT' and 'TARGETS' sections, with 'Application' selected under both. The main pane lists frameworks and libraries with their embedding options:

Name	Embed
A.framework	Embed & Sign ▾
B.framework	Do Not Embed ▾
libC.a	

At the bottom of the list, there are '+' and '-' buttons for adding or removing items.

# 프로젝트 생성 도구 조건

- 프레임워크, 라이브러리의 Embed 설정 및 의존성 관리

The screenshot shows the 'Frameworks, Libraries, and Embedded Content' section of the Xcode project settings. The sidebar on the left shows 'PROJECT' and 'TARGETS' sections, with 'Application' selected under TARGETS. The main pane lists three frameworks: 'A.framework' (Embed & Sign), 'B.framework' (Do Not Embed), and 'libC.a'. There are '+' and '-' buttons at the bottom for managing dependencies.

Name	Embed
A.framework	Embed & Sign
B.framework	Do Not Embed
libC.a	

# 프로젝트 생성 도구 조건

- 프레임워크, 라이브러리의 Embed 설정 및 의존성 관리

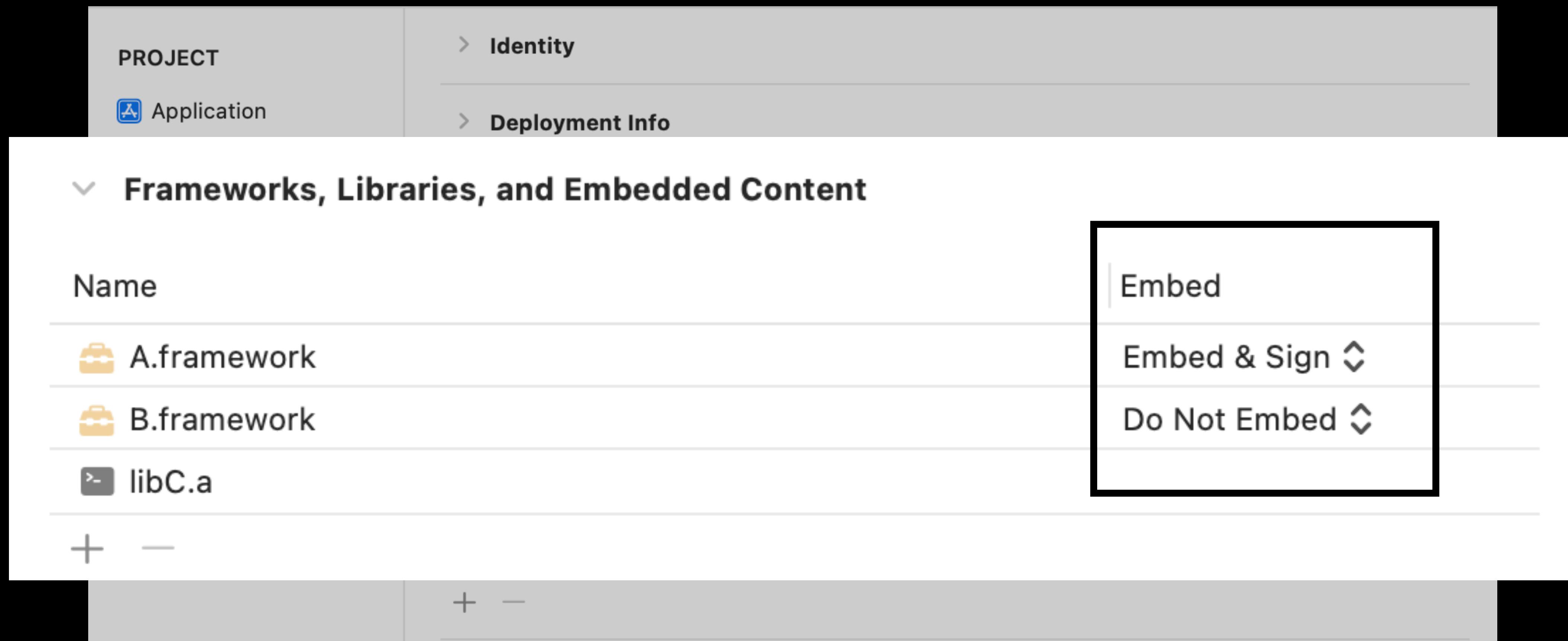
The screenshot shows the Xcode Project Navigator with the 'Application' tab selected. The 'Identity' section is open, revealing the 'Deployment Info' section. Under 'Frameworks, Libraries, and Embedded Content', there is a table listing three items: 'A.framework' (Embed & Sign), 'B.framework' (Do Not Embed), and 'libC.a'. At the bottom of the list are '+' and '-' buttons for adding or removing items.

Name	Embed
A.framework	Embed & Sign ▾
B.framework	Do Not Embed ▾
libC.a	

+ -

# 프로젝트 생성 도구 조건

- 프레임워크, 라이브러리의 Embed 설정 및 의존성 관리



# 프로젝트 생성 도구 조건

- 프레임워크, 라이브러리의 Embed 설정 및 의존성 관리

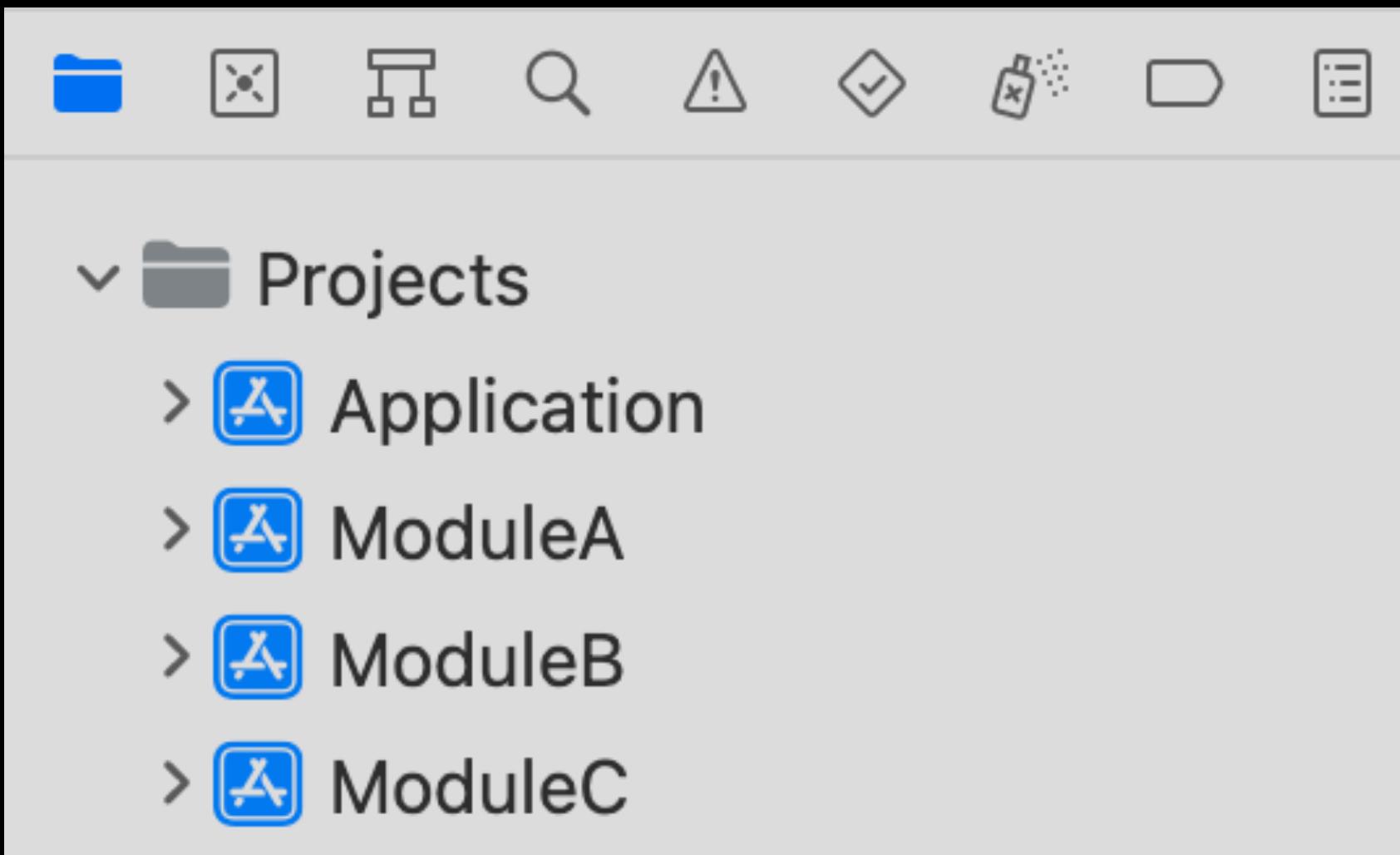
The screenshot shows the Xcode Project Navigator with the 'Application' tab selected. The 'Identity' and 'Deployment Info' sections are visible above the main list. The main list is titled 'Frameworks, Libraries, and Embedded Content'. It contains three entries: 'A.framework', 'B.framework', and 'libC.a'. To the right of each entry is a dropdown menu with three options: 'Embed', 'Embed & Sign', and 'Do Not Embed'. The 'Embed' option is highlighted with a black border.

Name	Embed Options
A.framework	Embed, Embed & Sign, Do Not Embed
B.framework	Embed, Embed & Sign, Do Not Embed
libC.a	Embed, Embed & Sign, Do Not Embed

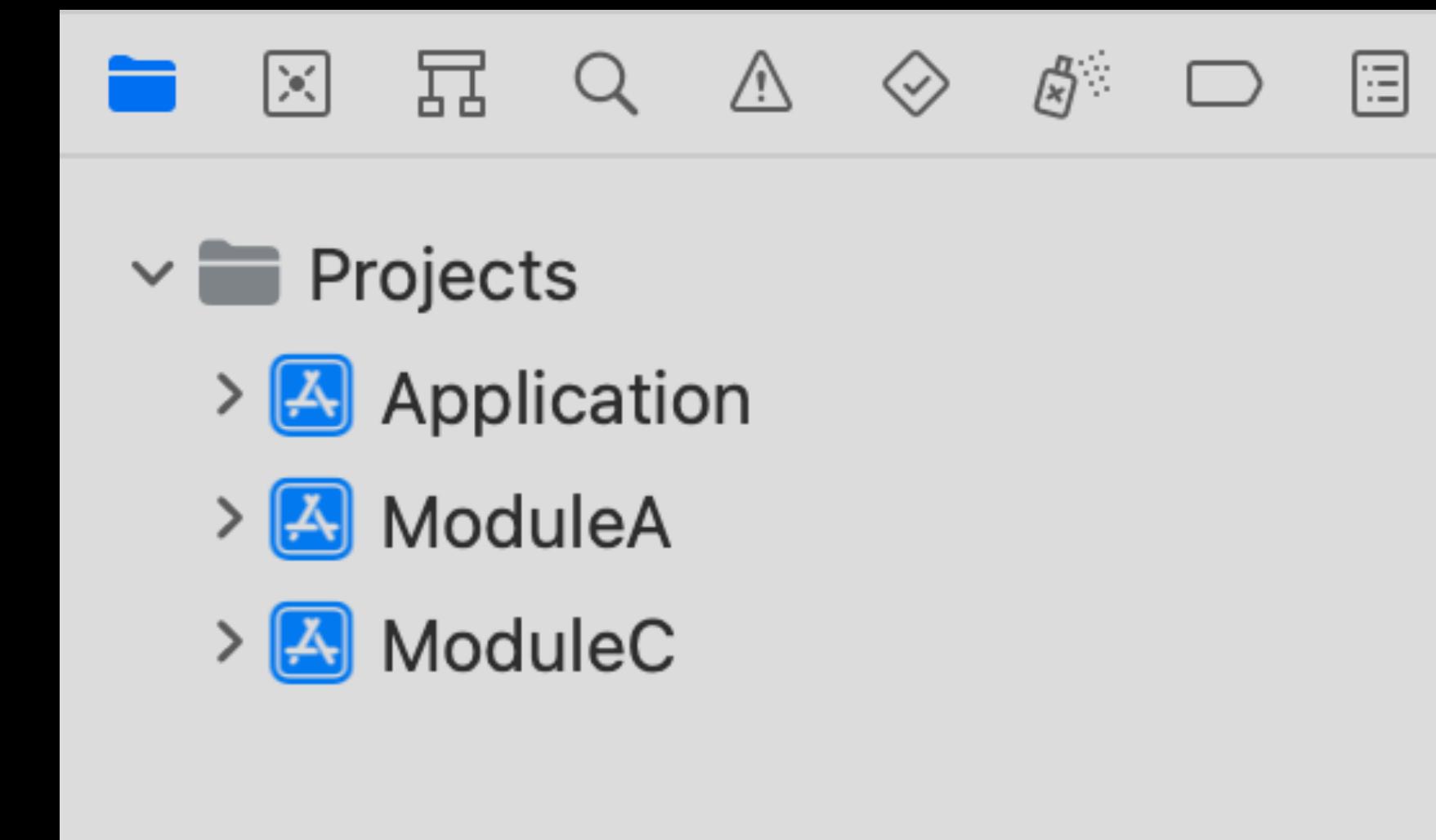
# 프로젝트 생성 도구 조건

- 조건에 따라 프로젝트, 워크스페이스 생성 제어
  - 환경변수를 이용하여 조건문으로 분기 처리 가능
  - 특정 프로젝트를 기준으로 워크스페이스 구성

EXCEPT\_MODULEB 환경변수가 없는 경우



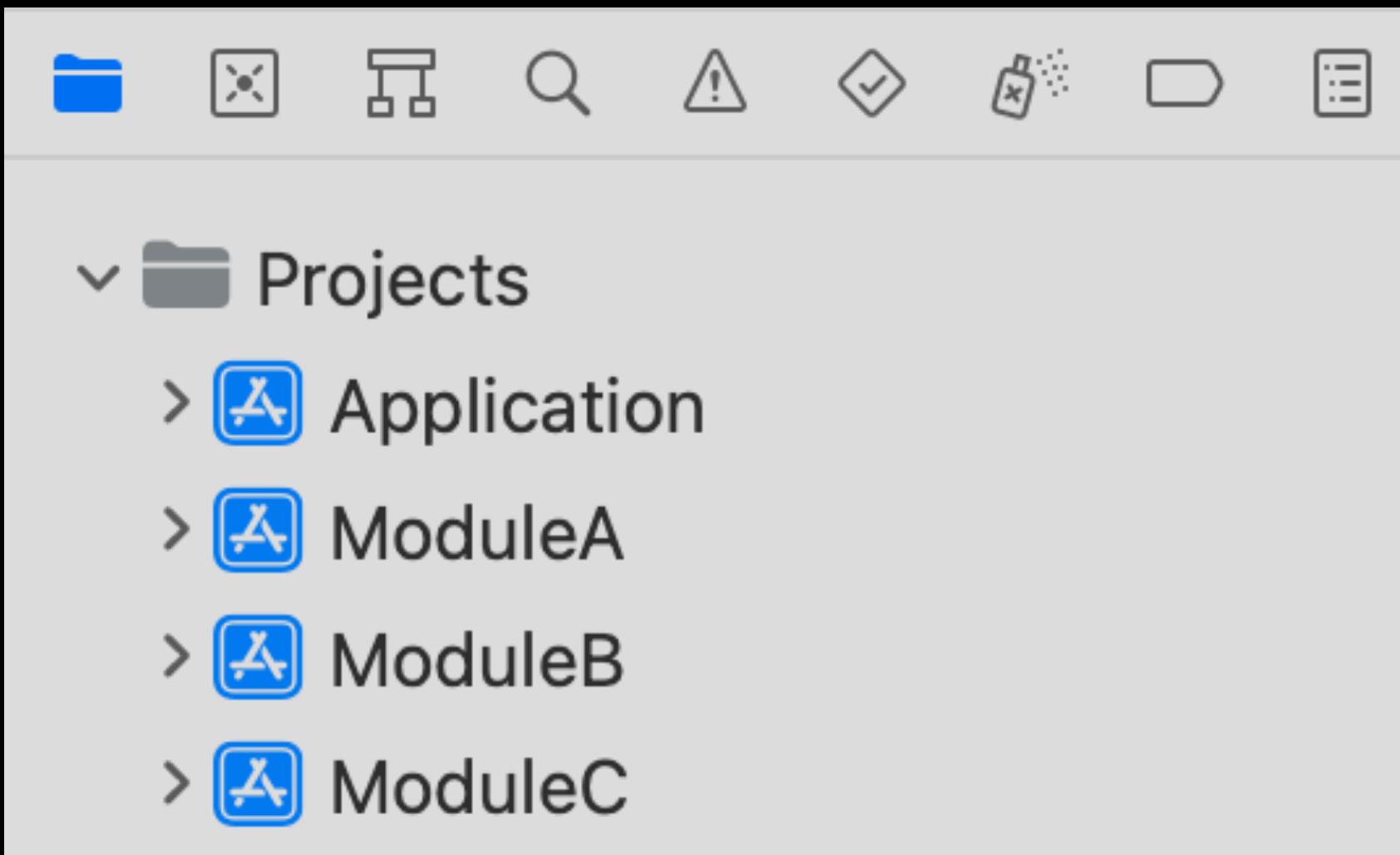
EXCEPT\_MODULEB 환경변수 값이 Y인 경우



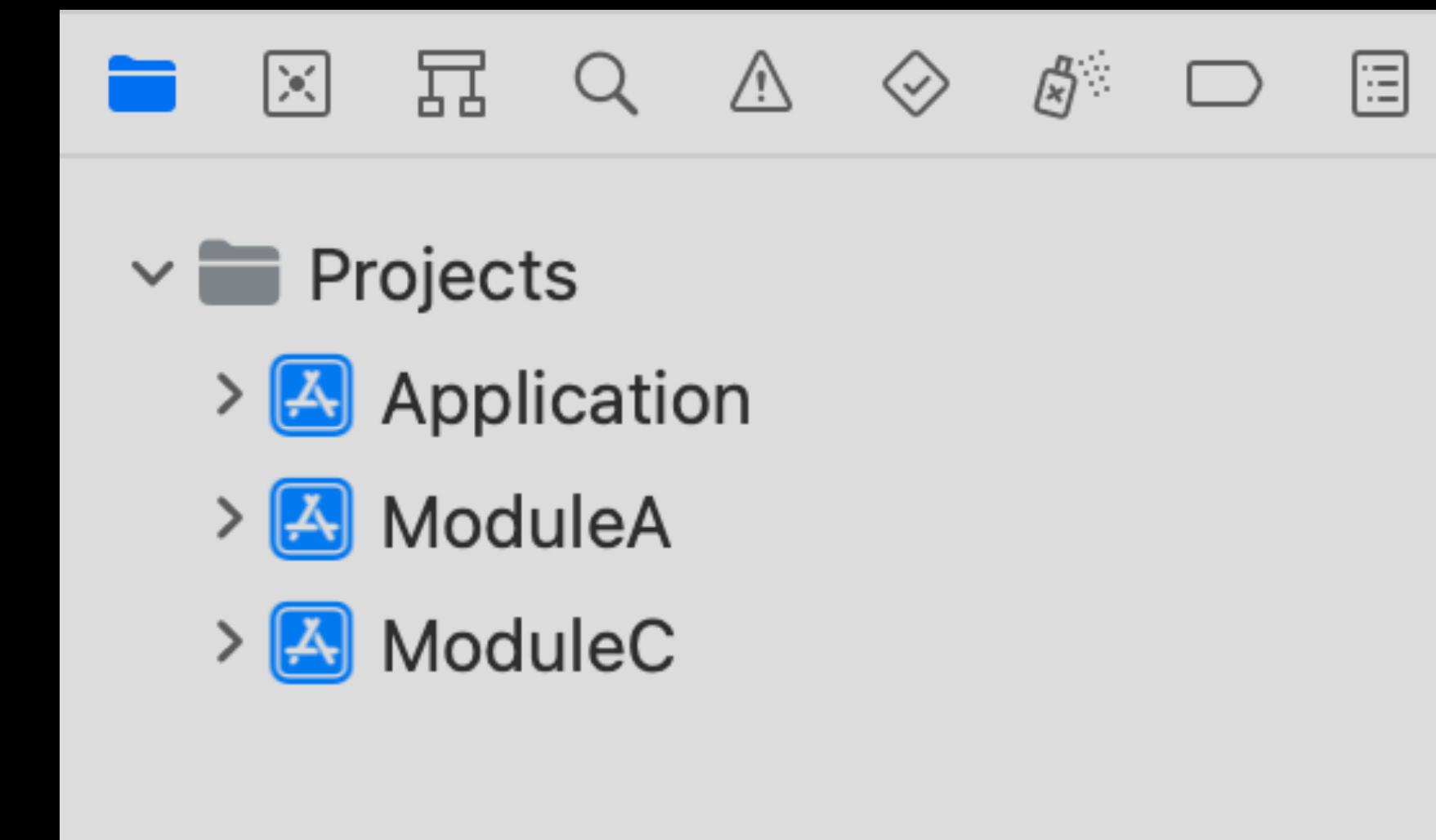
# 프로젝트 생성 도구 조건

- 조건에 따라 프로젝트, 워크스페이스 생성 제어
  - 환경변수를 이용하여 조건문으로 분기 처리 가능
  - 특정 프로젝트를 기준으로 워크스페이스 구성

EXCEPT\_MODULEB 환경변수가 없는 경우

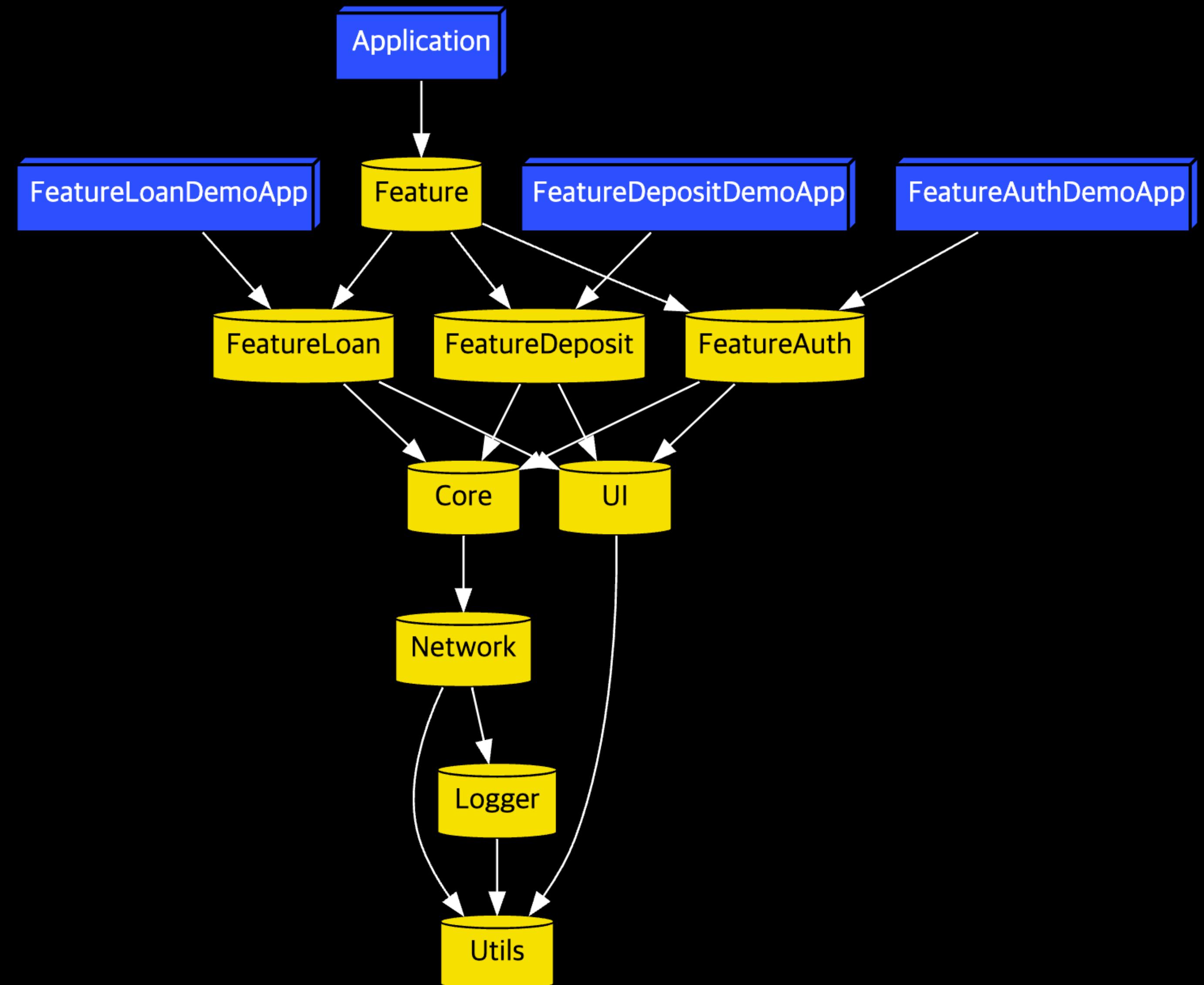


EXCEPT\_MODULEB 환경변수 값이 Y인 경우



# 프로젝트 생성 도구 조건

- 모듈의 의존성 그래프 생성 및 가시화
  - 전체 의존성 그래프 확인으로 확장 가능한 구조를 설계 가능



# 프로젝트 생성 도구 조건 비교

버전에 따라 가변적일 수 있습니다.

	Tuist	XcodeGen
Manifest 사용 가능 언어	Swift	YAML
템플릿 사용	가능	가능
의존성 그래프 생성	가능	가능
프레임워크, 라이브러리 의존성 관리	가능	가능
환경변수 사용	가능	가능
조건에 따라 프로젝트 동적 생성	가능	불가

# 프로젝트 생성 도구 조건 비교

버전에 따라 가변적일 수 있습니다.

	Tuist	XcodeGen
Manifest 사용 가능 언어	Swift	YAML
템플릿 사용	가능	가능
의존성 그래프 생성	가능	가능
프레임워크, 라이브러리 의존성 관리	가능	가능
환경변수 사용	가능	가능
조건에 따라 프로젝트 동적 생성	가능	불가

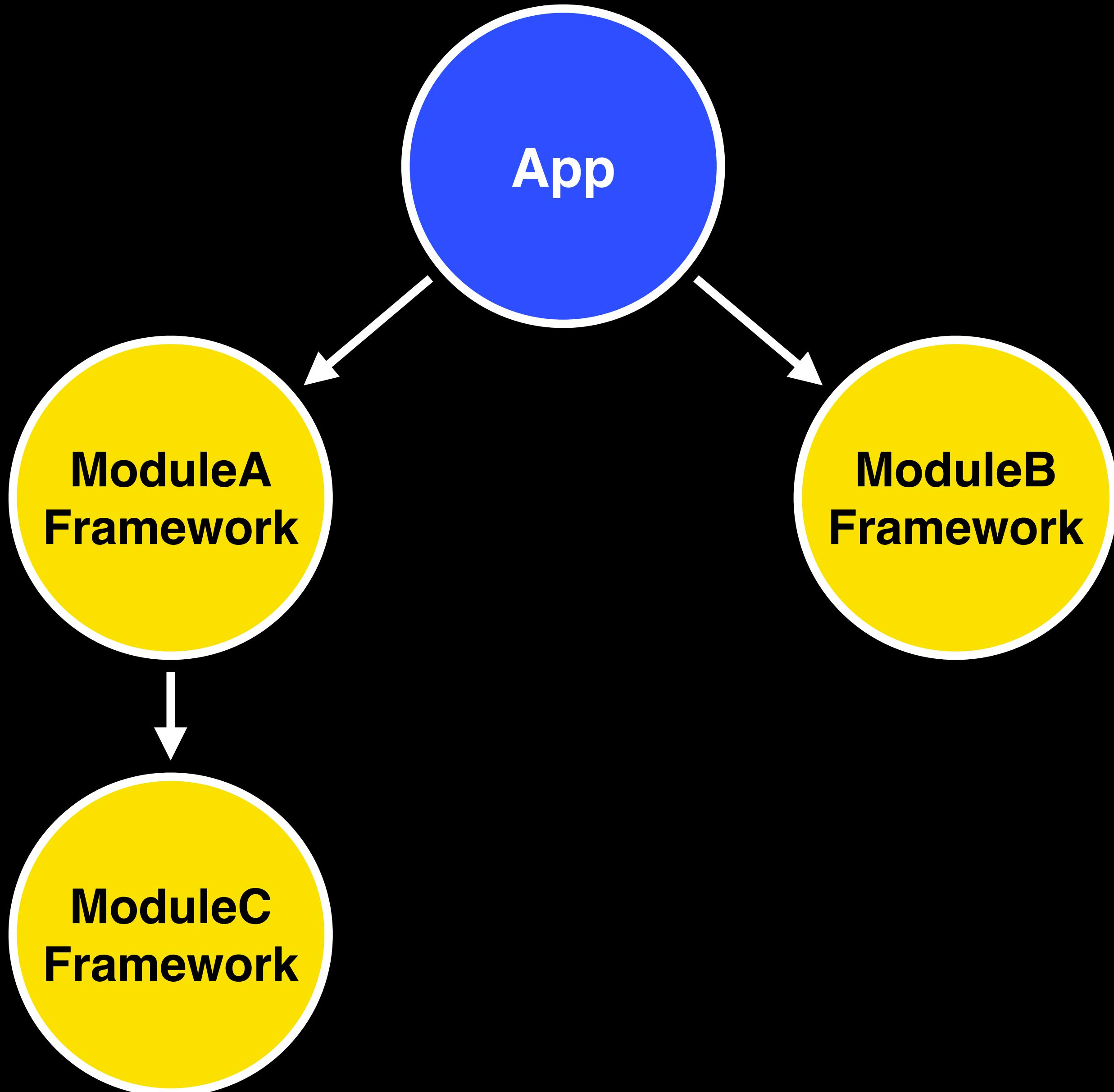
# 프로젝트 생성 도구 조건 비교

버전에 따라 가변적일 수 있습니다.

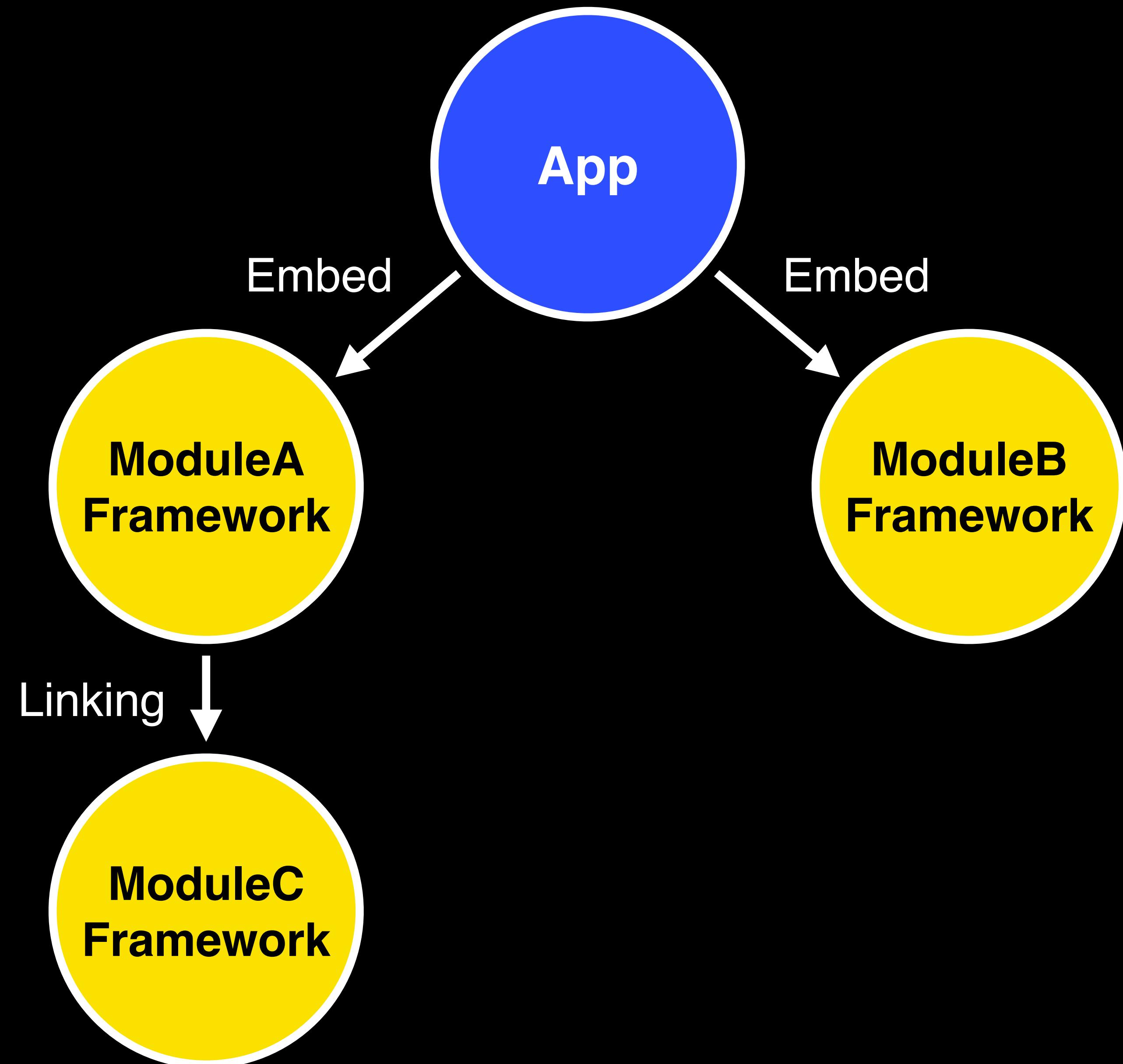
	Tuist	XcodeGen
Manifest 사용 가능 언어	Swift	YAML
템플릿 사용	가능	가능
의존성 그래프 생성	가능	가능
프레임워크, 라이브러리 의존성 관리	가능	가능
환경변수 사용	가능	가능
조건에 따라 프로젝트 동적 생성	가능	불가

# Tuist를 이용한 프로젝트 관리

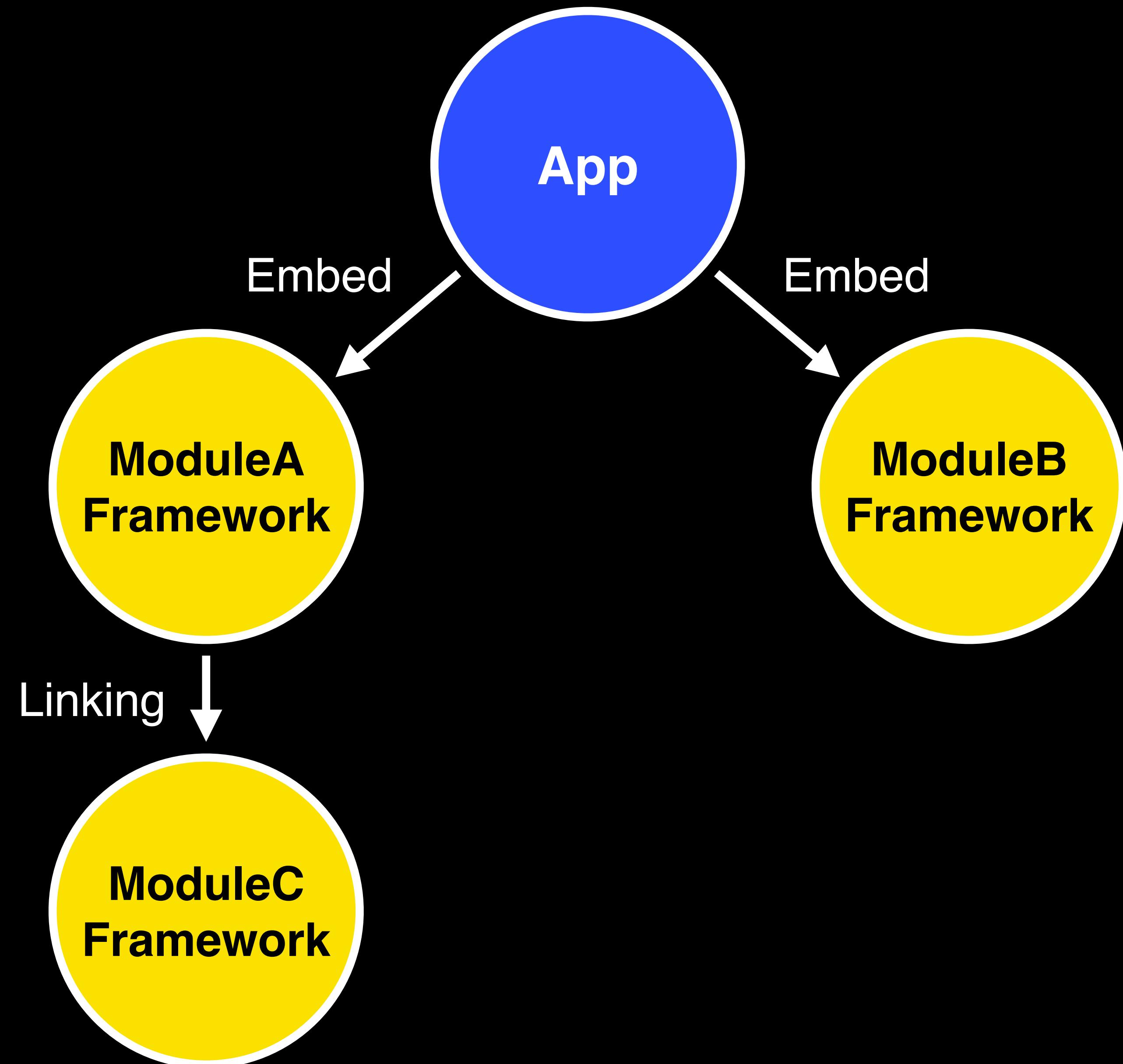
# 프로젝트 관리



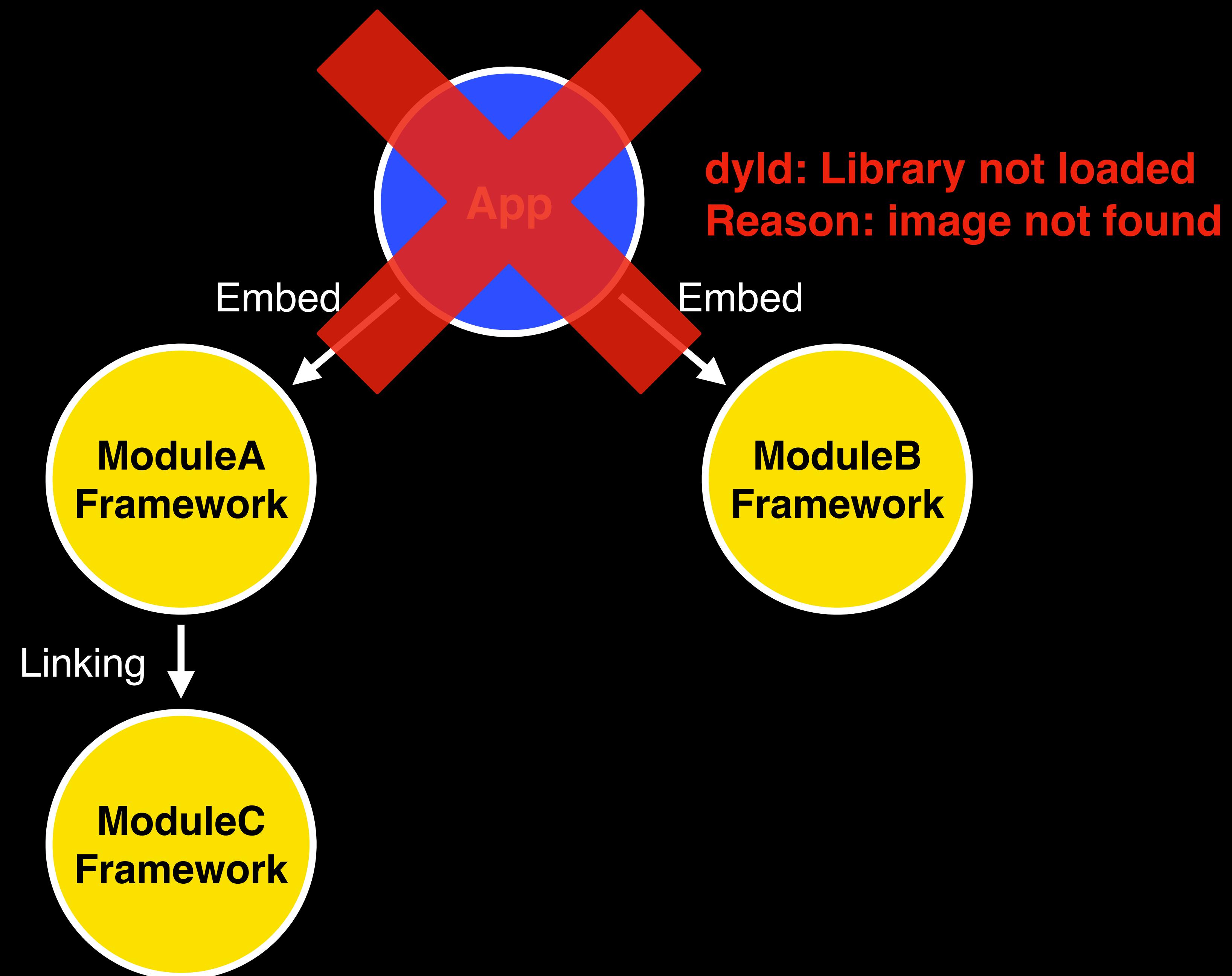
# 프로젝트 관리



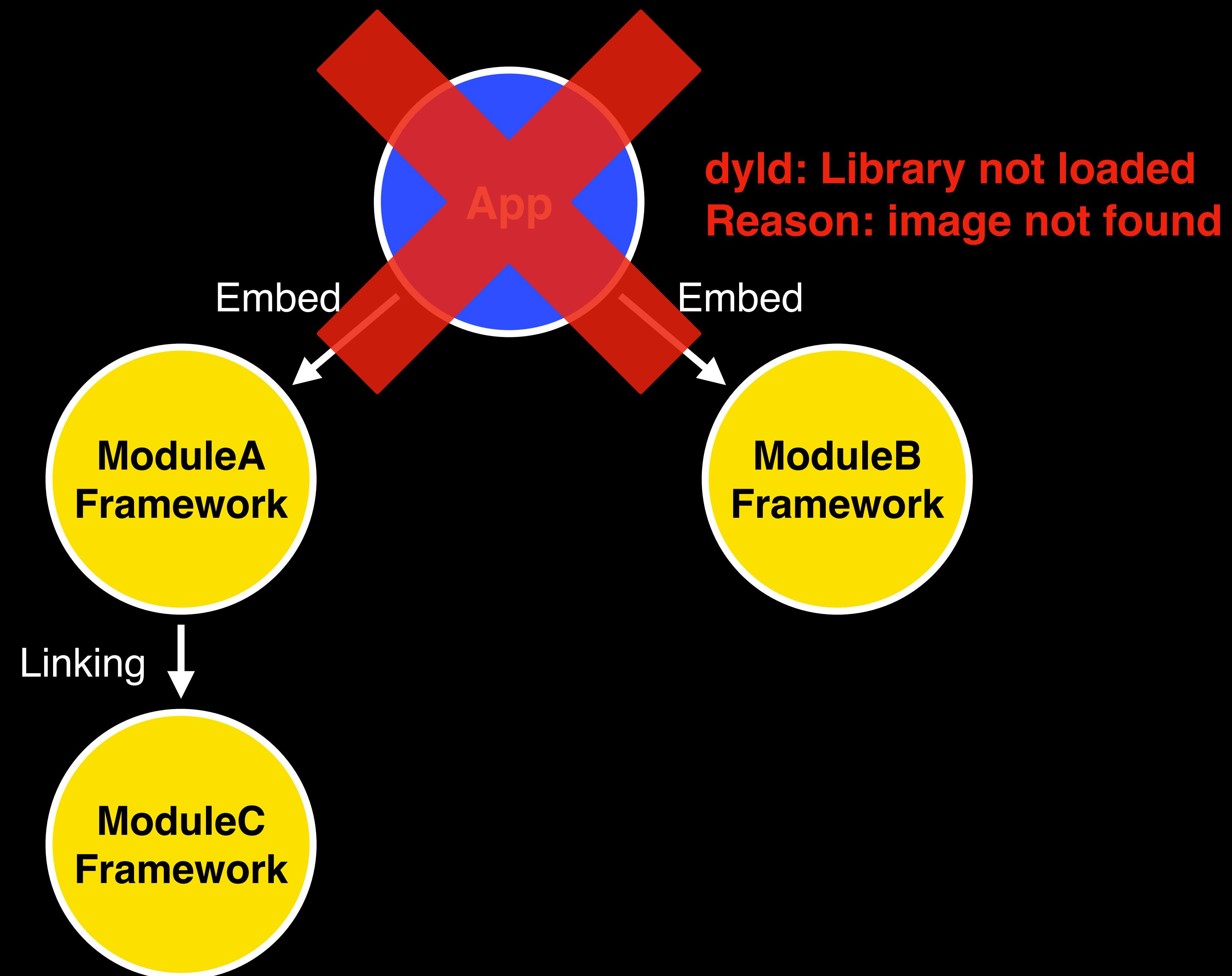
# 프로젝트 관리



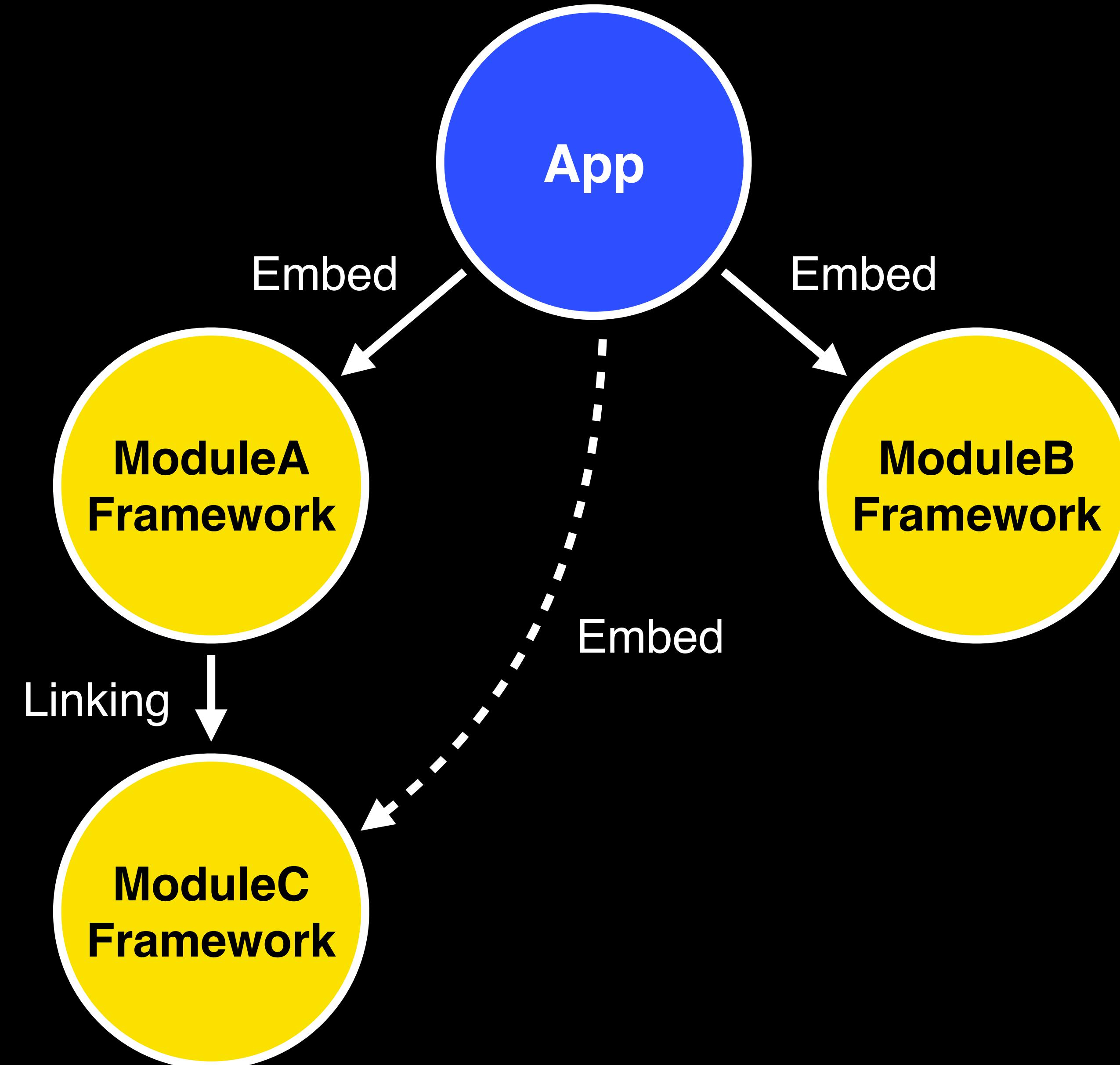
# 프로젝트 관리



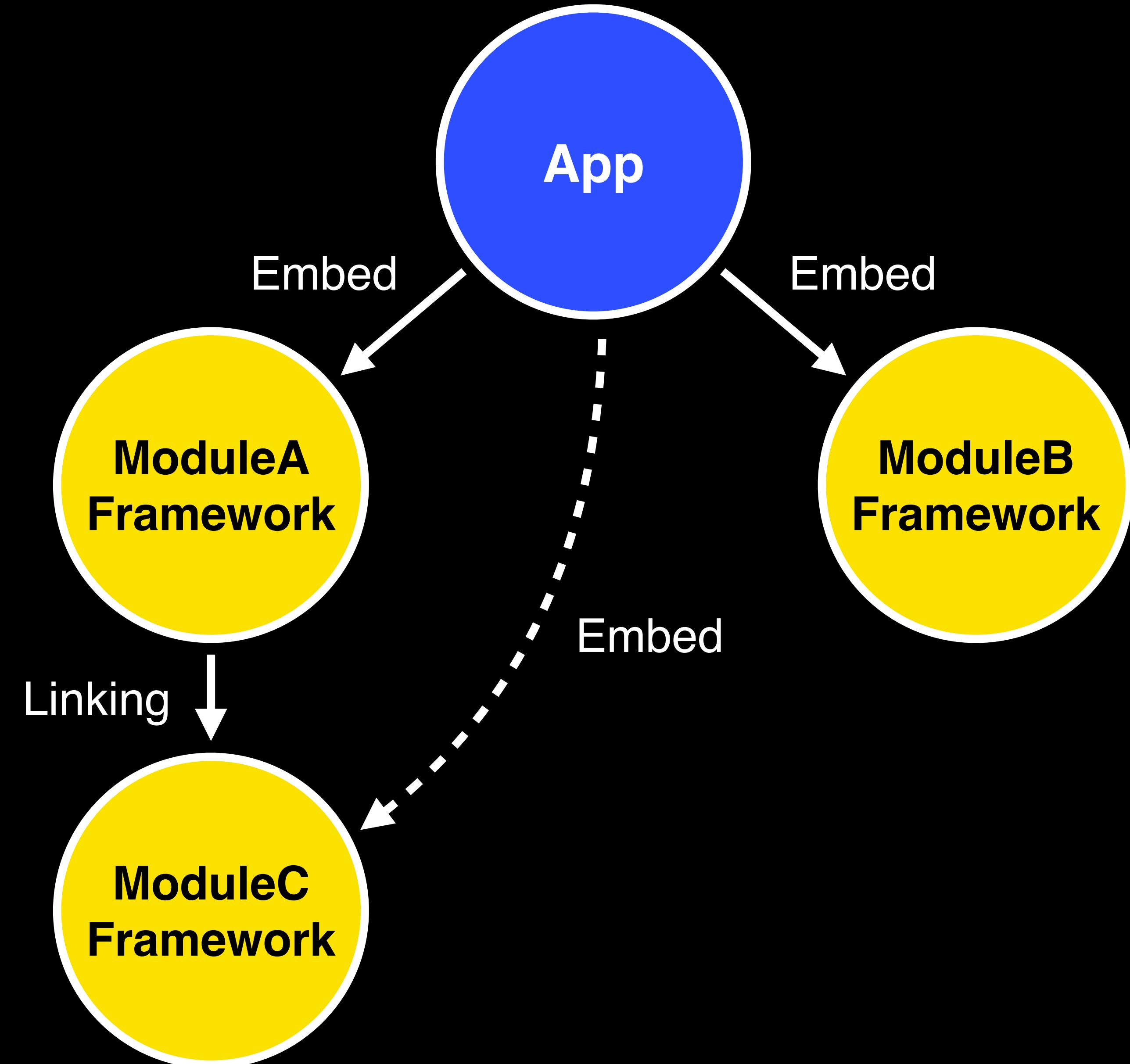
# 프로젝트 관리



# 프로젝트 관리



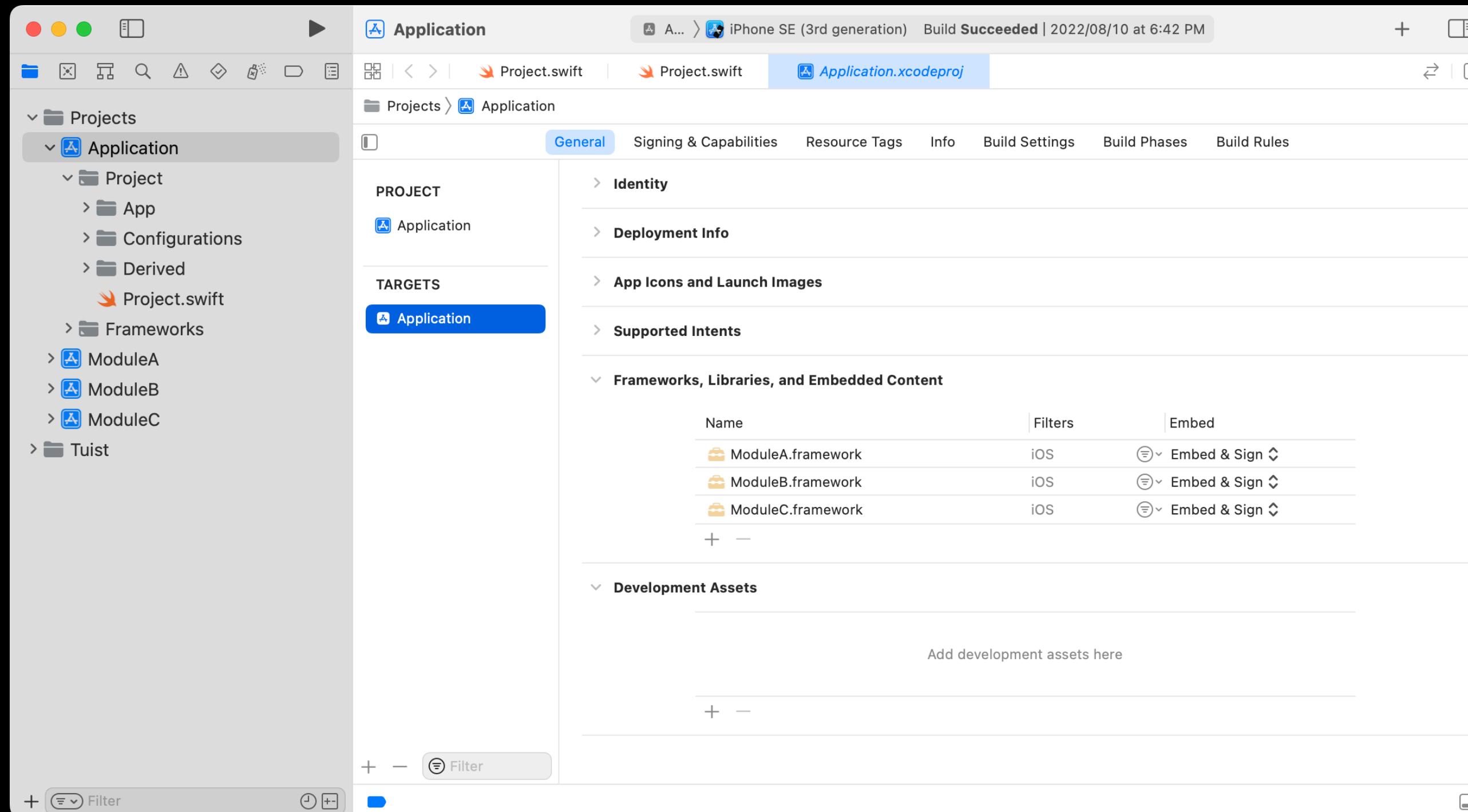
# 프로젝트 관리



# 프로젝트 관리

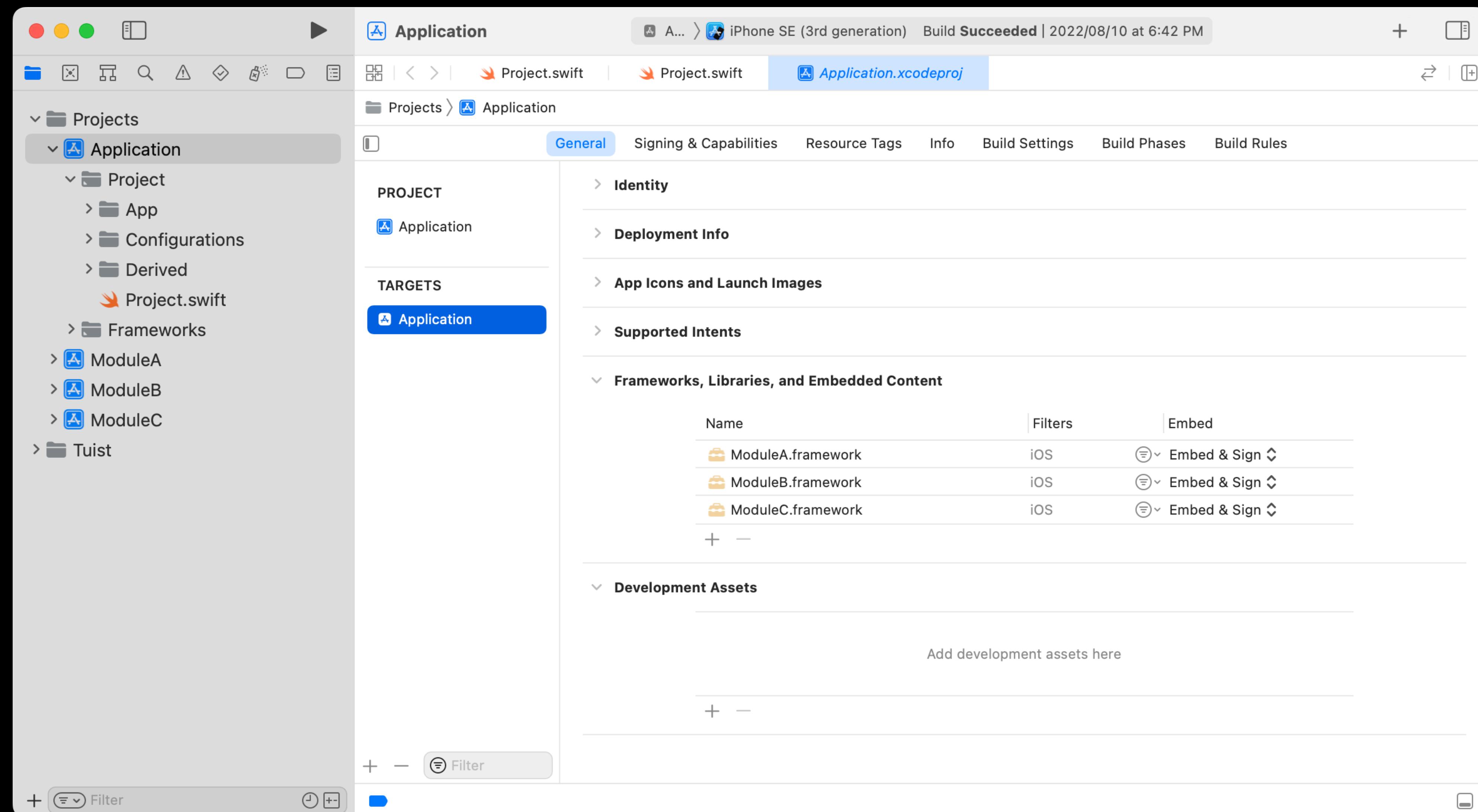
```
let targets: [Target] = [  
    Target(  
        name: "Application",  
        platform: .iOS,  
        product: .app,  
        bundleId: "com.kakaobank.app",  
        sources: ["App/Sources/**"],  
        resources: ["App/Resources/**"],  
        dependencies: [  
            .ModuleA,  
            .ModuleB,  
        ],  
        settings: .settings(configurations: XCConfig.example)  
    )  
]
```

# 프로젝트 관리

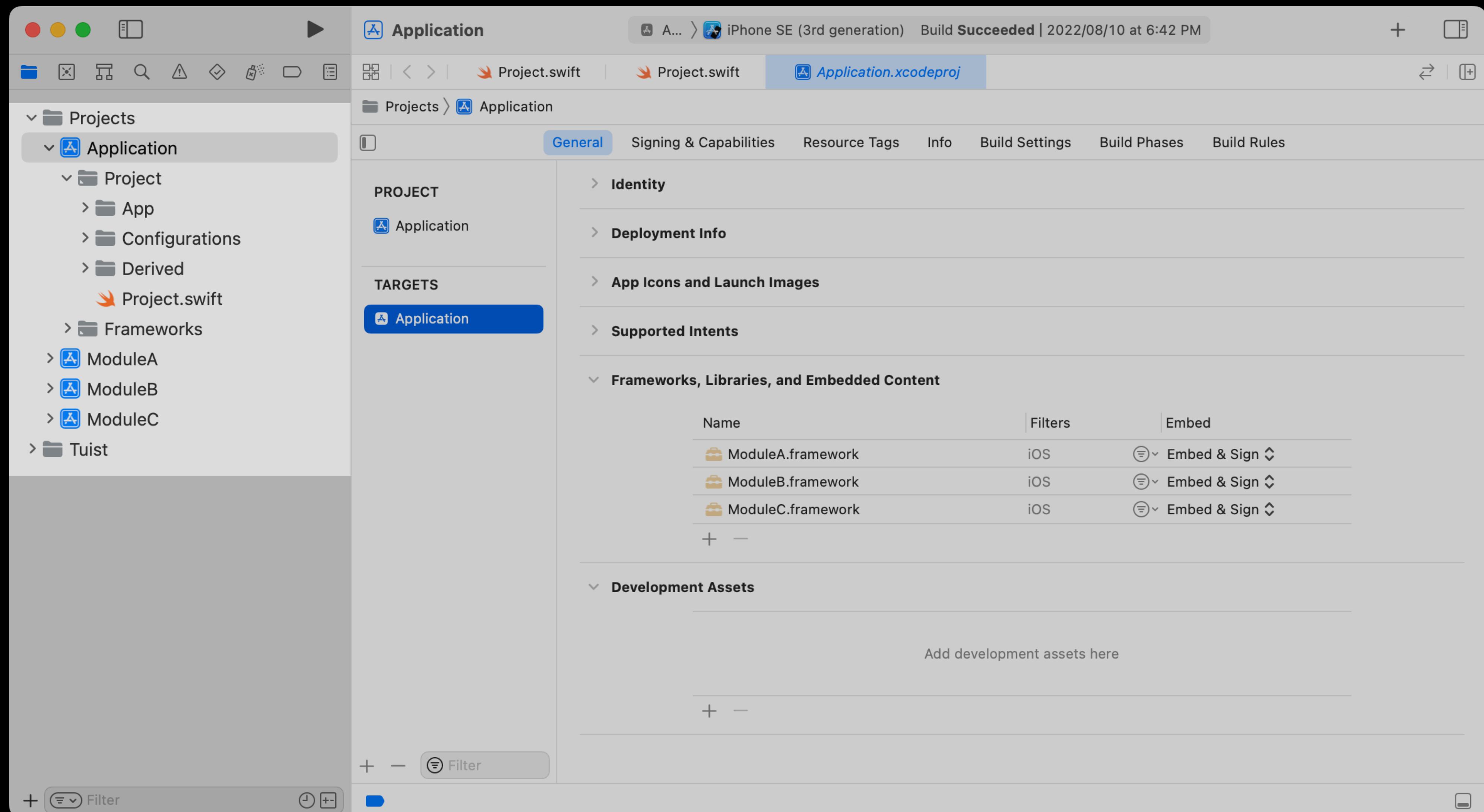


```
let targets: [Target] = [
    Target(
        name: "Application",
        platform: .iOS,
        product: .app,
        bundleId: "com.kakaobank.app",
        sources: ["App/Sources/**"],
        resources: ["App/Resources/**"],
        dependencies: [
            .ModuleA,
            .ModuleB,
        ],
        settings: .settings(configurations: XCConfig.example)
    )
]
```

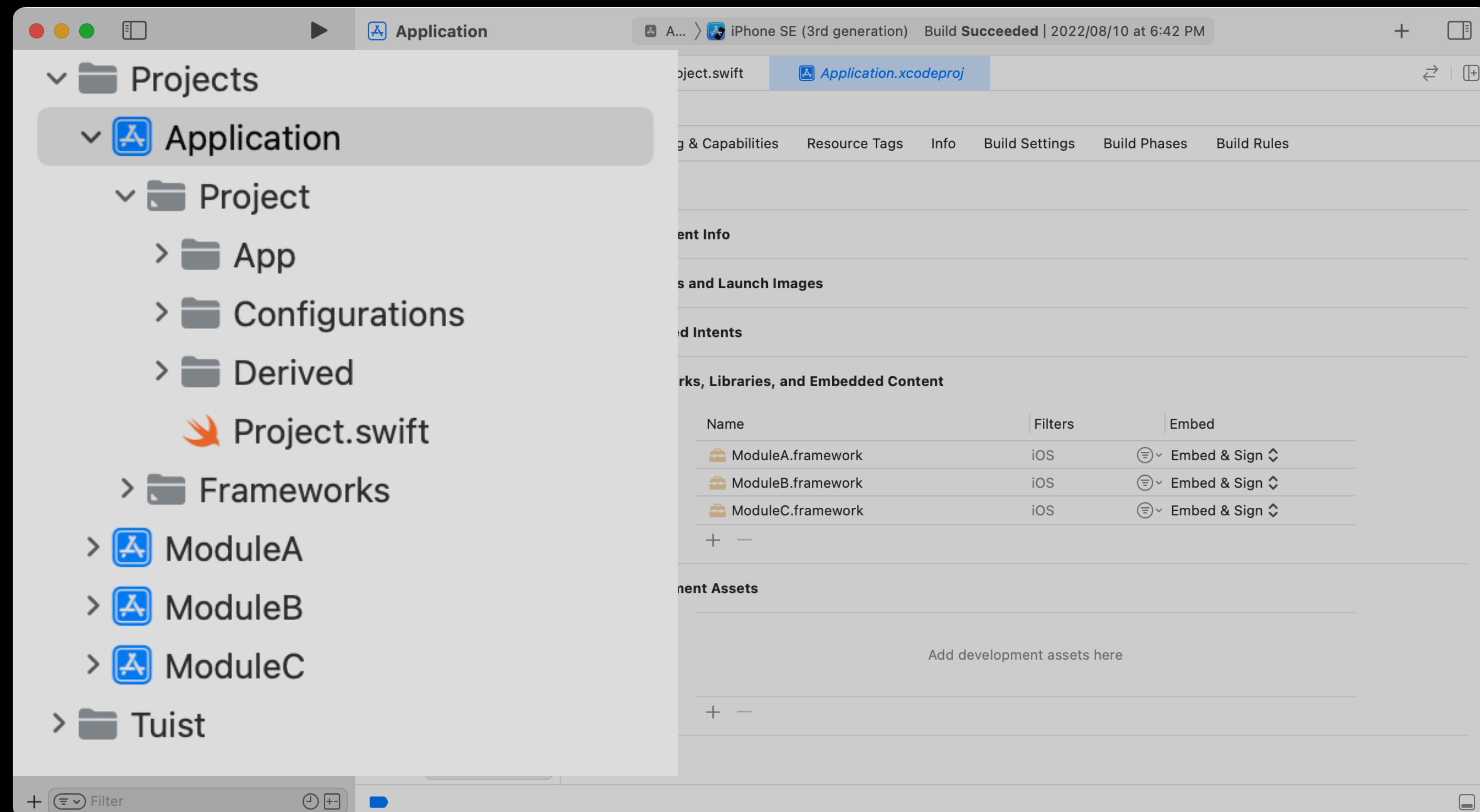
# 프로젝트 관리



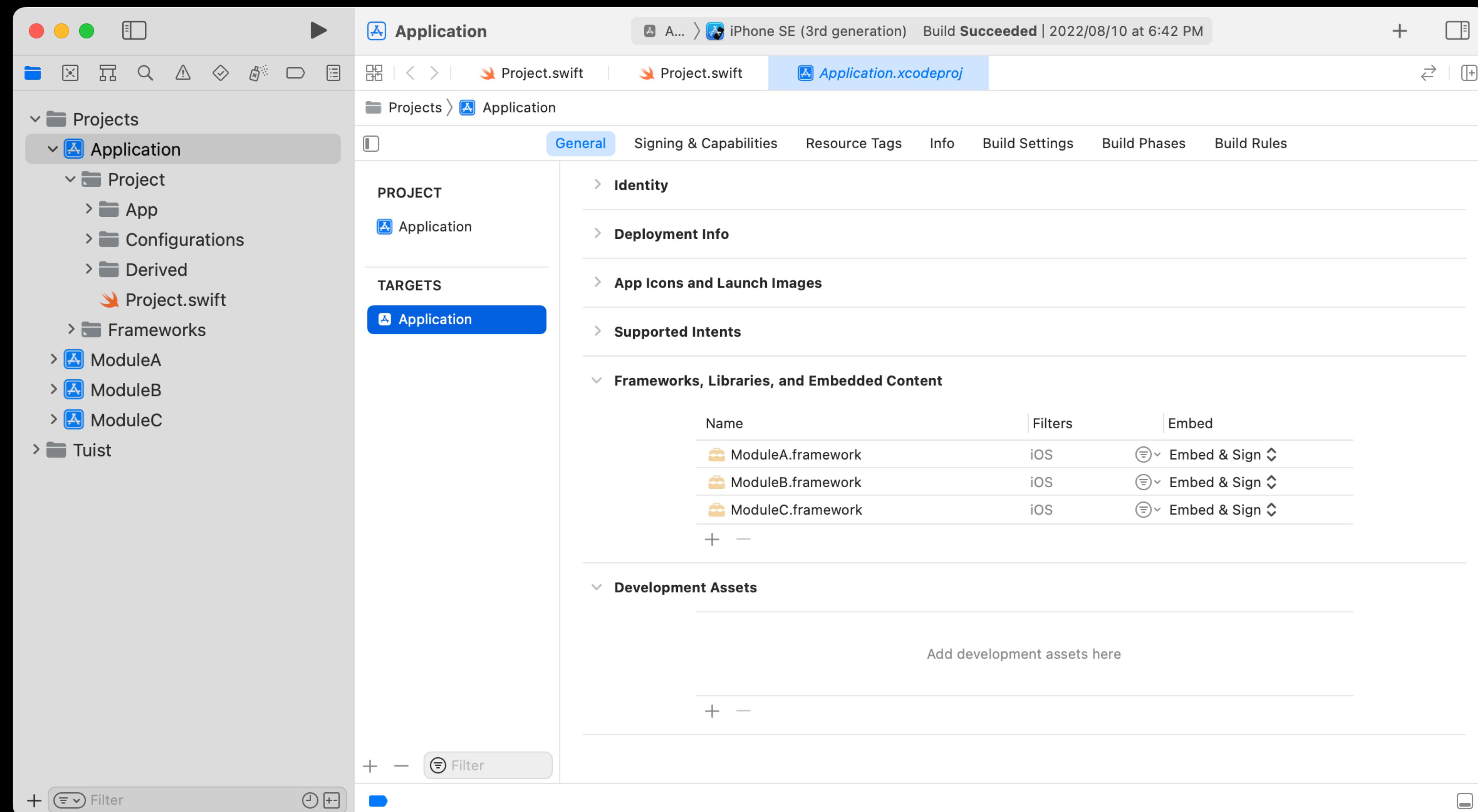
# 프로젝트 관리



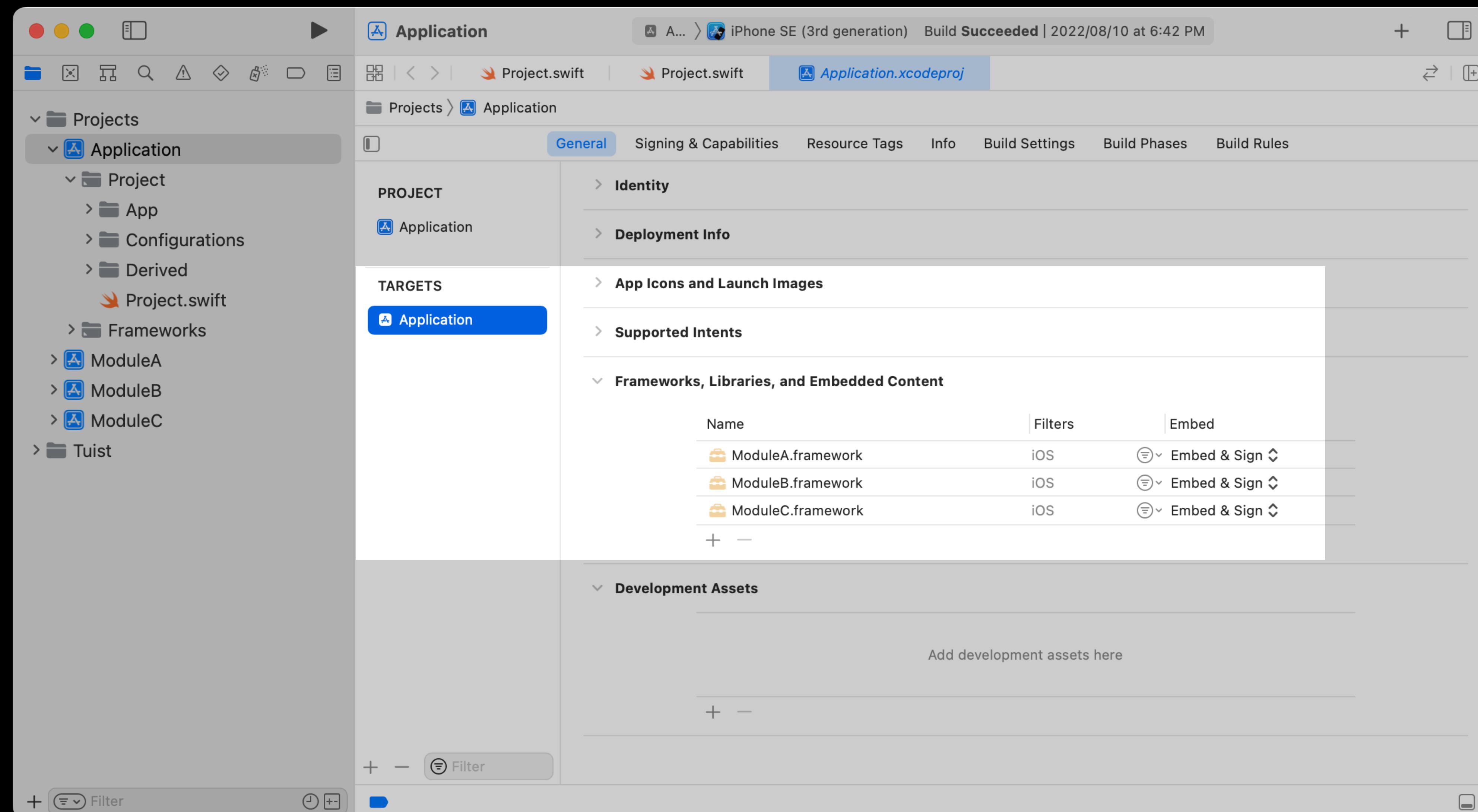
# 프로젝트 관리



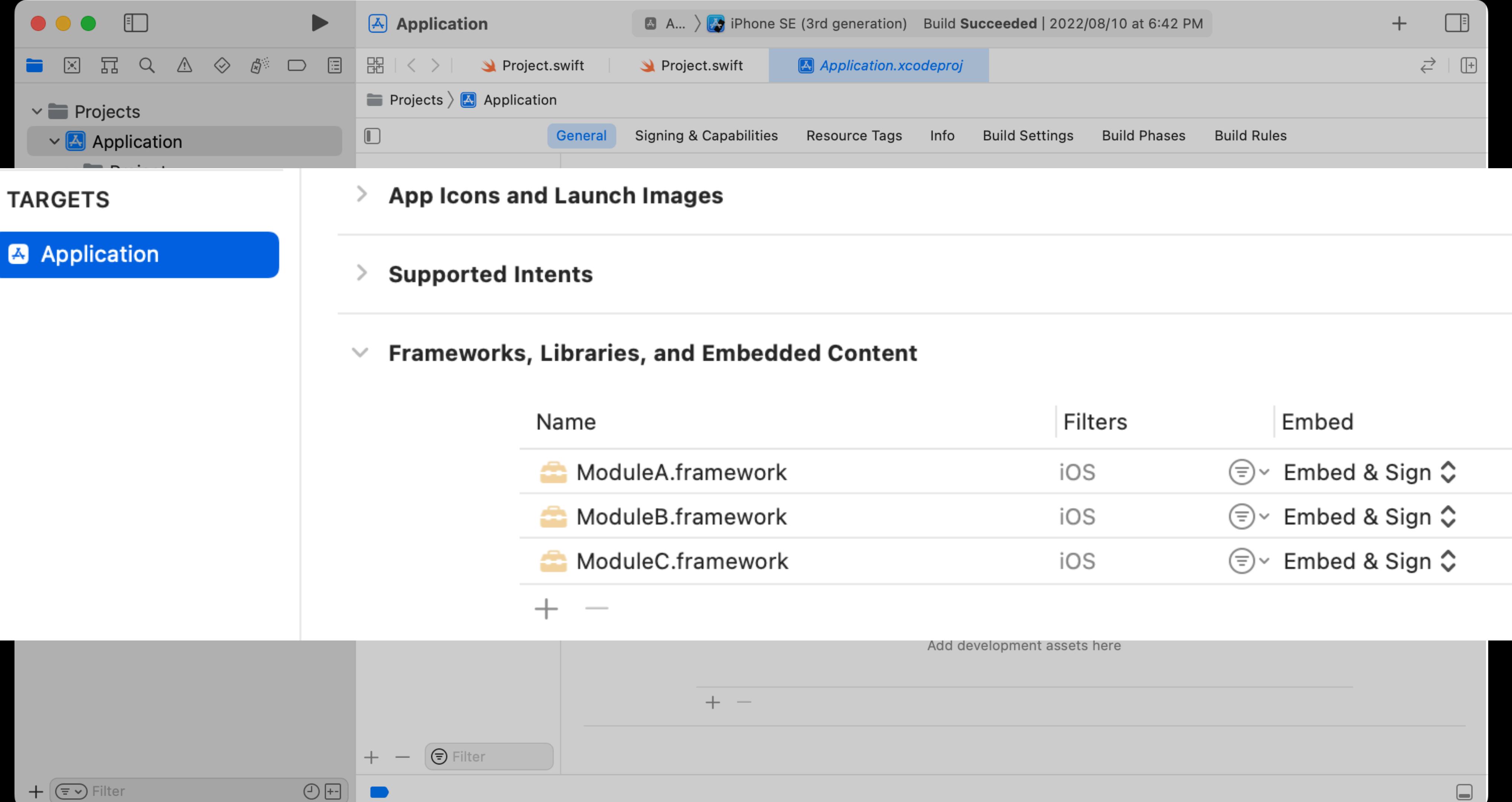
# 프로젝트 관리



# 프로젝트 관리



# 프로젝트 관리



The screenshot shows the Xcode interface with the following details:

- Toolbar:** Standard Mac OS X-style toolbar with icons for file operations.
- Project Navigator:** Shows the project structure with "Projects" expanded, and "Application" selected.
- Search Bar:** Displays the path "A... > iPhone SE (3rd generation)" and the message "Build Succeeded | 2022/08/10 at 6:42 PM".
- Tab Bar:** Contains tabs for "Application", "Project.swift", "Project.swift", and "Application.xcodeproj", with "Application.xcodeproj" currently active.
- Table of Contents:** Lists sections: "TARGETS", "App Icons and Launch Images", "Supported Intents", and "Frameworks, Libraries, and Embedded Content".
- Frameworks Table:** A table showing three frameworks:

Name	Filters	Embed
ModuleA.framework	iOS	Embed & Sign
ModuleB.framework	iOS	Embed & Sign
ModuleC.framework	iOS	Embed & Sign
- Asset Catalog:** An empty asset catalog area with the placeholder text "Add development assets here".
- Bottom Bar:** Includes buttons for adding (+), removing (-), filtering, and a search icon.

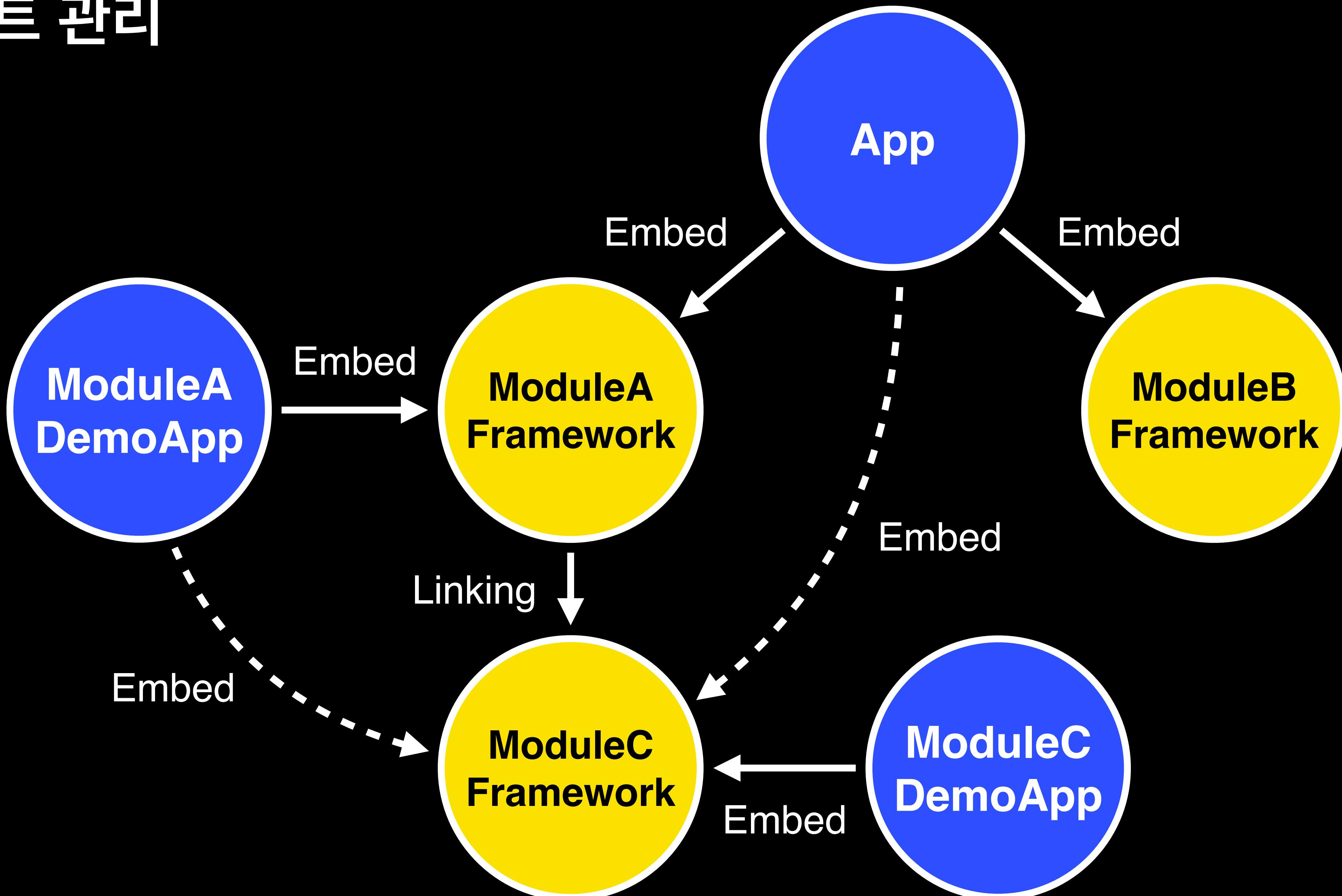
# 프로젝트 관리

The screenshot shows the Xcode interface with the following details:

- Toolbar:** Standard Mac OS X-style toolbar with icons for file operations.
- Project Navigator:** Shows the project structure under "Projects". The "Application" target is selected, indicated by a blue bar at the top of its row.
- Central Area:** The "General" tab is selected in the "Application" target settings.
- Targets Section:** The "Application" target is selected.
- Frameworks, Libraries, and Embedded Content:** This section is expanded, showing three frameworks:

Name	Filters	Embed
ModuleA.framework	iOS	Embed & Sign
ModuleB.framework	iOS	Embed & Sign
ModuleC.framework	iOS	Embed & Sign
- Bottom Bar:** Includes buttons for adding (+), removing (-), filtering, and a search field.

# 프로젝트 관리



# 프로젝트 관리

TUIST로 시작하는 환경변수 사용

일반적인 로직처럼 작성 가능

```
$ export TUIST_BANK_EXCEPT_MODULEB=Y
```

```
$ tuist generate
```

```
let env = ProcessInfo.processInfo.environment
let isExceptModuleB = (env["TUIST_BANK_EXCEPT_MODULEB"] ?? "") == "Y"
let targets: [Target] = [
    Target(
        name: "Application",
        platform: .iOS,
        product: .app,
        bundleId: "com.kakaobank.app",
        sources: ["App/Sources/**"],
        resources: ["App/Resources/**"],
        dependencies: [
            [.ModuleA],
            isExceptModuleB
            ? []
            : [.ModuleB]
        ].flatMap { $0 },
        settings: .settings(configurations: XConfig.example)
    )
]
```

# 프로젝트 관리

TUIST로 시작하는 환경변수 사용

일반적인 로직처럼 작성 가능

```
$ export TUIST_BANK_EXCEPT_MODULEB=Y
```

```
$ tuist generate
```

```
let env = ProcessInfo.processInfo.environment
let isExceptModuleB = (env["TUIST_BANK_EXCEPT_MODULEB"] ?? "") == "Y"
let targets: [Target] = [
    Target(
        name: "Application",
        platform: .iOS,
        product: .app,
        bundleId: "com.kakaobank.app",
        sources: ["App/Sources/**"],
        resources: ["App/Resources/**"],
        dependencies: [
            [.ModuleA],
            isExceptModuleB
            ? []
            : [.ModuleB]
        ].flatMap { $0 },
        settings: .settings(configurations: XConfig.example)
    )
]
```

# 프로젝트 관리

TUIST로 시작하는 환경변수 사용  
일반적인 로직처럼 작성 가능

```
$ export TUIST_BANK_EXCEPT_MODULEB=Y  
$ tuist generate
```

```
let env = ProcessInfo.processInfo.environment
let isExceptModuleB = (env["TUIST_BANK_EXCEPT_MODULEB"] ?? "") == "Y"
let targets: [target] = [
    Target(
        name: "Application",
        platform: .iOS,
        product: .app,
        bundleId: "com.kakaobank.app",
        sources: ["App/Sources/**"],
        resources: ["App/Resources/**"],
        dependencies: [
            [.ModuleA],
            isExceptModuleB
            ? []
            : [.ModuleB]
        ].flatMap { $0 },
        settings: .settings(configurations: XConfig.example)
    )
]
```

# 프로젝트 관리

The screenshot shows the Xcode interface with the Project Navigator on the left and the Editor on the right.

**Project Navigator:**

- Projects folder
  - Application (selected)
  - Project
    - App
    - Configurations
    - Derived
    - Project.swift
  - Frameworks
  - ModuleA
  - ModuleC
  - Tuist

**Editor - Application.xcodeproj (General tab):**

Build log: A... > iPhone SE (3rd generation) Build Succeeded | 2022/08/10 at 6:42 PM

File tabs: Application, Project.swift, Project.swift, Application.xcodeproj (selected)

Table of Contents:

- General
- Signing & Capabilities
- Resource Tags
- Info
- Build Settings
- Build Phases
- Build Rules

**PROJECT**

- Application

**TARGETS**

- Application

**Frameworks, Libraries, and Embedded Content**

Name	Filters	Embed
ModuleA.framework	iOS	Embed & Sign
ModuleC.framework	iOS	Embed & Sign

**Development Assets**

# 프로젝트 관리

The screenshot shows the Xcode interface for managing a project named "Application".

**Project Navigator:** On the left, under the "Projects" section, the "Application" target is selected, highlighted with a blue background.

**Editor Area:** The main area displays the "General" tab of the "Application" target settings.

**PROJECT Section:** Contains the "Application" target configuration.

**TARGETS Section:** Contains the "Application" target configuration.

**Frameworks, Libraries, and Embedded Content:** A table showing embedded frameworks:

Name	Filters	Embed
ModuleA.framework	iOS	Embed & Sign
ModuleC.framework	iOS	Embed & Sign

**Development Assets:** A section for managing development assets.

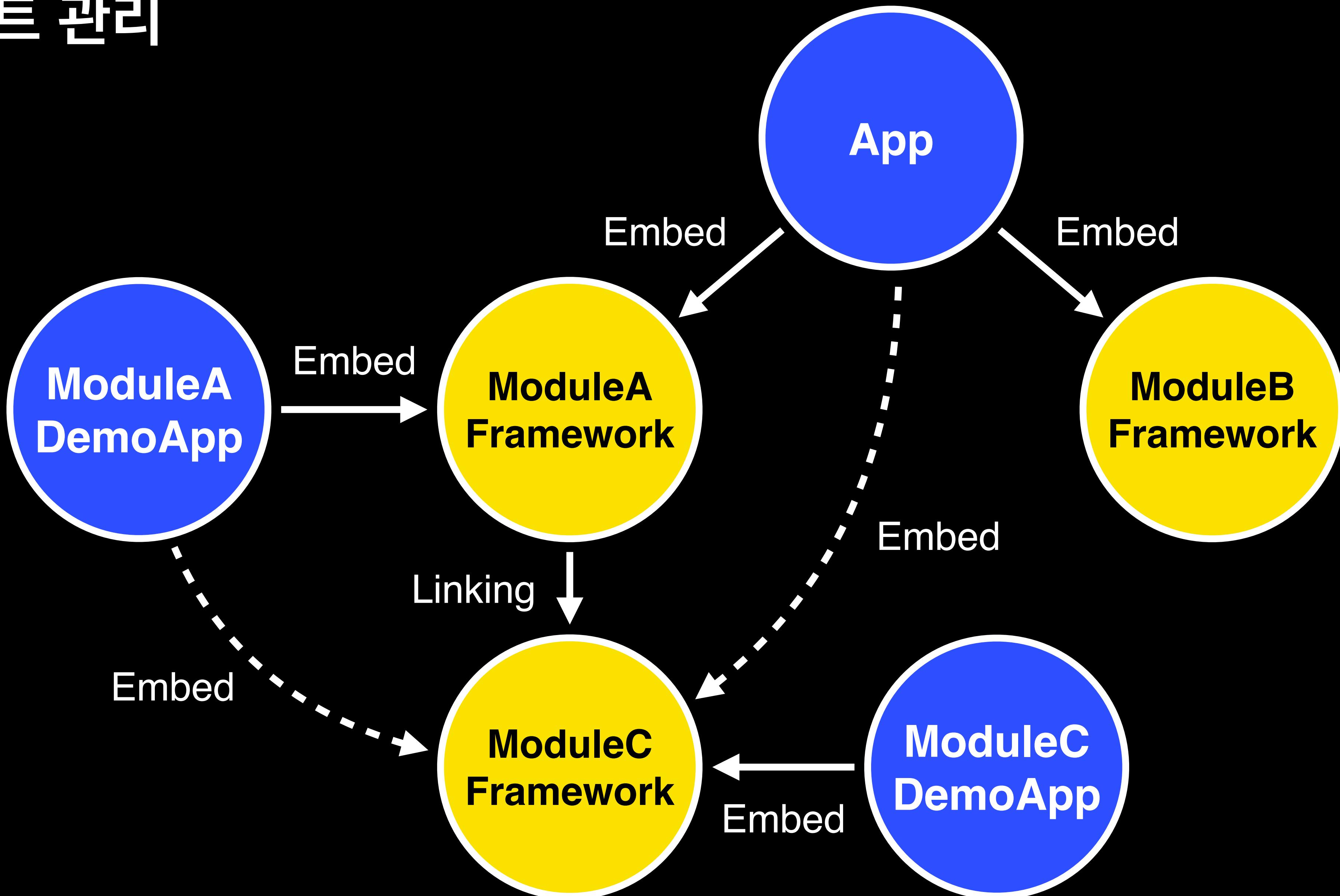
# 프로젝트 관리

The screenshot shows the Xcode interface with the following details:

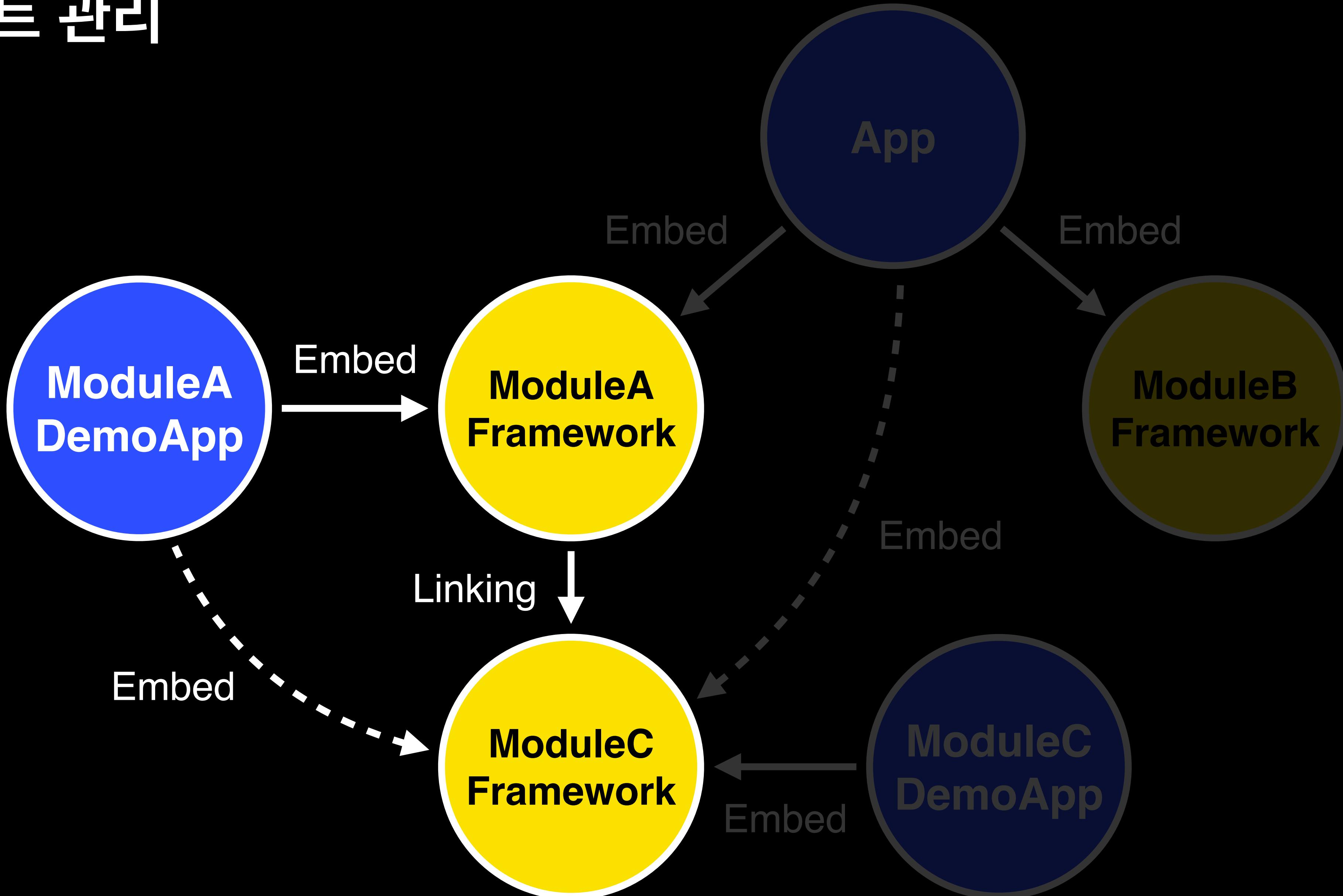
- Project Navigator:** On the left, under the "Projects" section, the "Application" project is selected, highlighted with a blue background.
- Build Log:** At the top center, it shows "iPhone SE (3rd generation) Build Succeeded | 2022/08/10 at 6:42 PM".
- File Navigator:** Below the build log, there are two entries: "Project.swift" and "Application.xcodeproj", with "Application.xcodeproj" currently selected.
- General Tab:** In the main content area, the "General" tab is selected for the "Application" target. It displays sections for Identity, Deployment Info, App Icons and Launch Images, and Supported Intents.
- Frameworks, Libraries, and Embedded Content:** A table view shows the following entries:

Name	Filters	Embed
ModuleA.framework	iOS	Embed & Sign
ModuleC.framework	iOS	Embed & Sign
- Bottom Navigation:** At the very bottom, there are buttons for adding (+), filtering, and saving (-).

# 프로젝트 관리

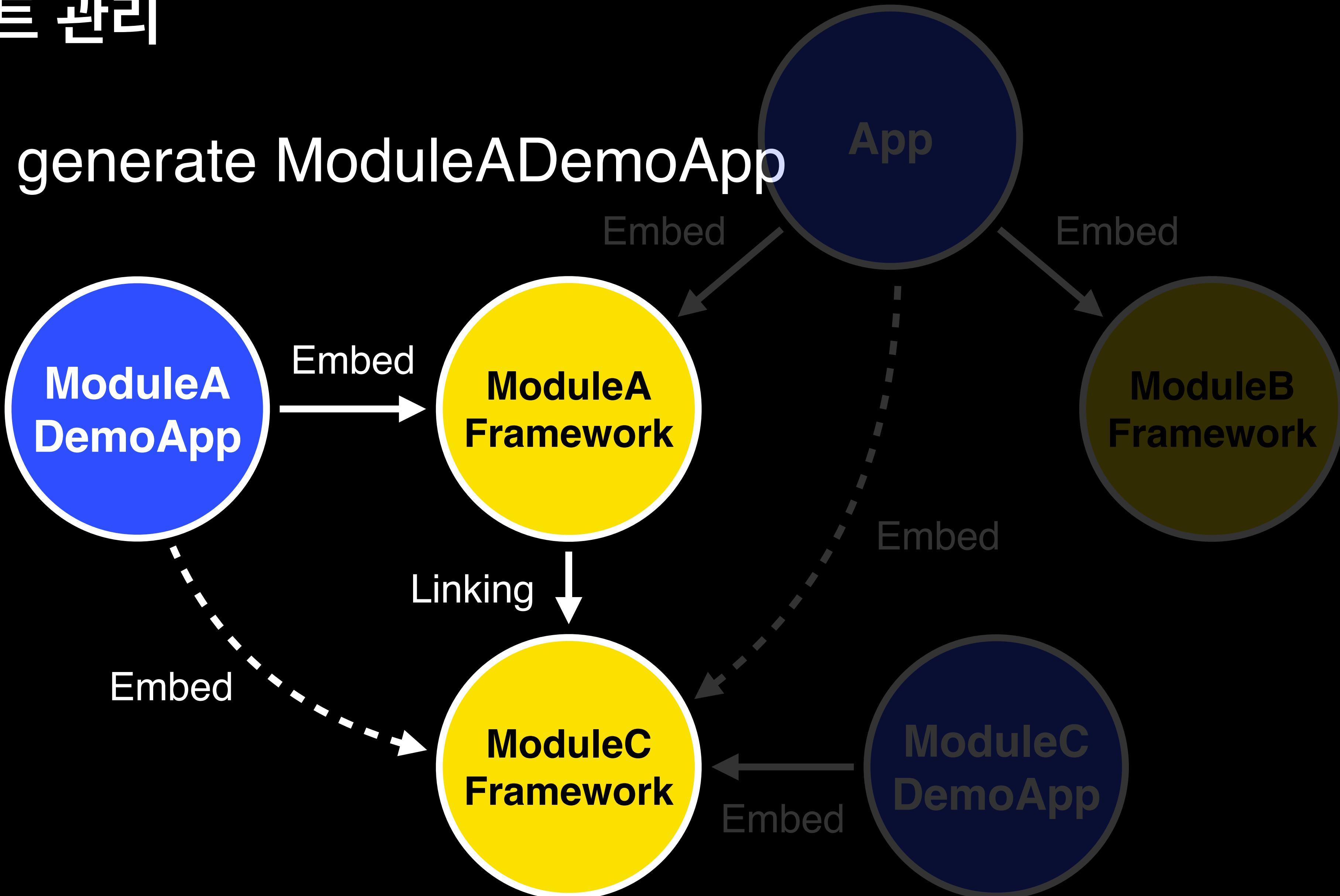


# 프로젝트 관리



# 프로젝트 관리

\$ tuist generate ModuleADemoApp



# 프로젝트 관리

The screenshot shows the Xcode interface for managing a project named "ModuleA".

**Project Navigator:** On the left, under "Projects", "ModuleA" is selected. The project structure includes "Project", "App", "Configurations", "Derived", "Sources", and "Project.swift". Other items like "Frameworks" and "Tuist" are also listed.

**Detail Area:** The main area displays the "General" tab for the "ModuleA" target. The "TARGETS" section lists "ModuleA" and "ModuleADemoApp".

**Frameworks, Libraries, and Embedded Content:** This section contains a table with two entries:

Name	Filters	Embed
ModuleA.framework	iOS	Embed & Sign
ModuleC.framework	iOS	Embed & Sign

**Development Assets:** A placeholder message "Add development assets here" is shown.

# 프로젝트 관리

The screenshot shows the Xcode interface for managing a project named "ModuleA".

**Project Navigator:** On the left, under "Projects", "ModuleA" is selected. The project structure includes "Project", "App", "Configurations", "Derived", "Sources", and "Project.swift". Other items like "Frameworks" and "ModuleC" are also listed.

**Detail Area:** The main area displays the "General" tab of the "ModuleA.xcodeproj" settings.

- PROJECT:** Shows "ModuleA" as the target.
- TARGETS:** Shows "ModuleADemoApp" as the selected target.
- Identity:** Configuration for the target.
- Deployment Info:** Deployment configuration.
- App Icons and Launch Images:** App icon and launch image settings.
- Supported Intents:** Supported intents for the app.
- Frameworks, Libraries, and Embedded Content:** Lists frameworks embedded in the build.

Name	Filters	Embed
ModuleA.framework	iOS	Embed & Sign
ModuleC.framework	iOS	Embed & Sign
- Development Assets:** Placeholder for development assets.

# 프로젝트 관리

The screenshot shows the Xcode interface with the following details:

**Project Navigator (Left):**

- Projects folder
  - ModuleA
    - Project
      - App
      - Configurations
      - Derived
      - Sources
      - Project.swift (selected)
    - Frameworks
    - ModuleC
  - Tuist

**Build Phases Tab (Right):**

The "ModuleA.xcodeproj" tab is selected in the top navigation bar.

Build Phases tab content:

- Signing & Capabilities
- Resource Tags
- Info
- Build Settings
- Build Phases (selected)
- Build Rules

**Identity**

**Deployment Info**

**App Icons and Launch Images**

**Supported Intents**

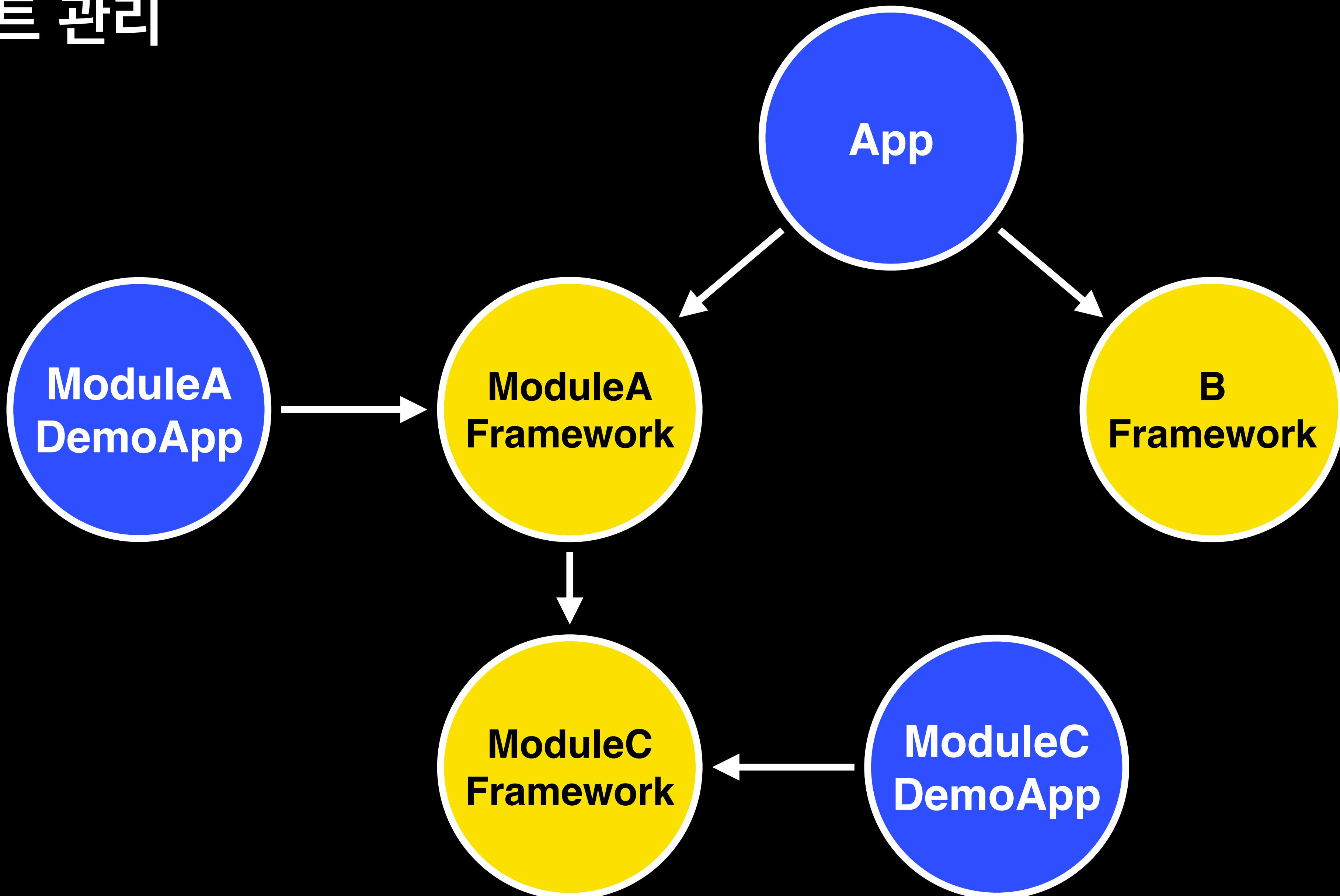
**Frameworks, Libraries, and Embedded Content**

Name	Filters	Embed
ModuleA.framework	iOS	Embed & Sign
ModuleC.framework	iOS	Embed & Sign

**Development Assets**

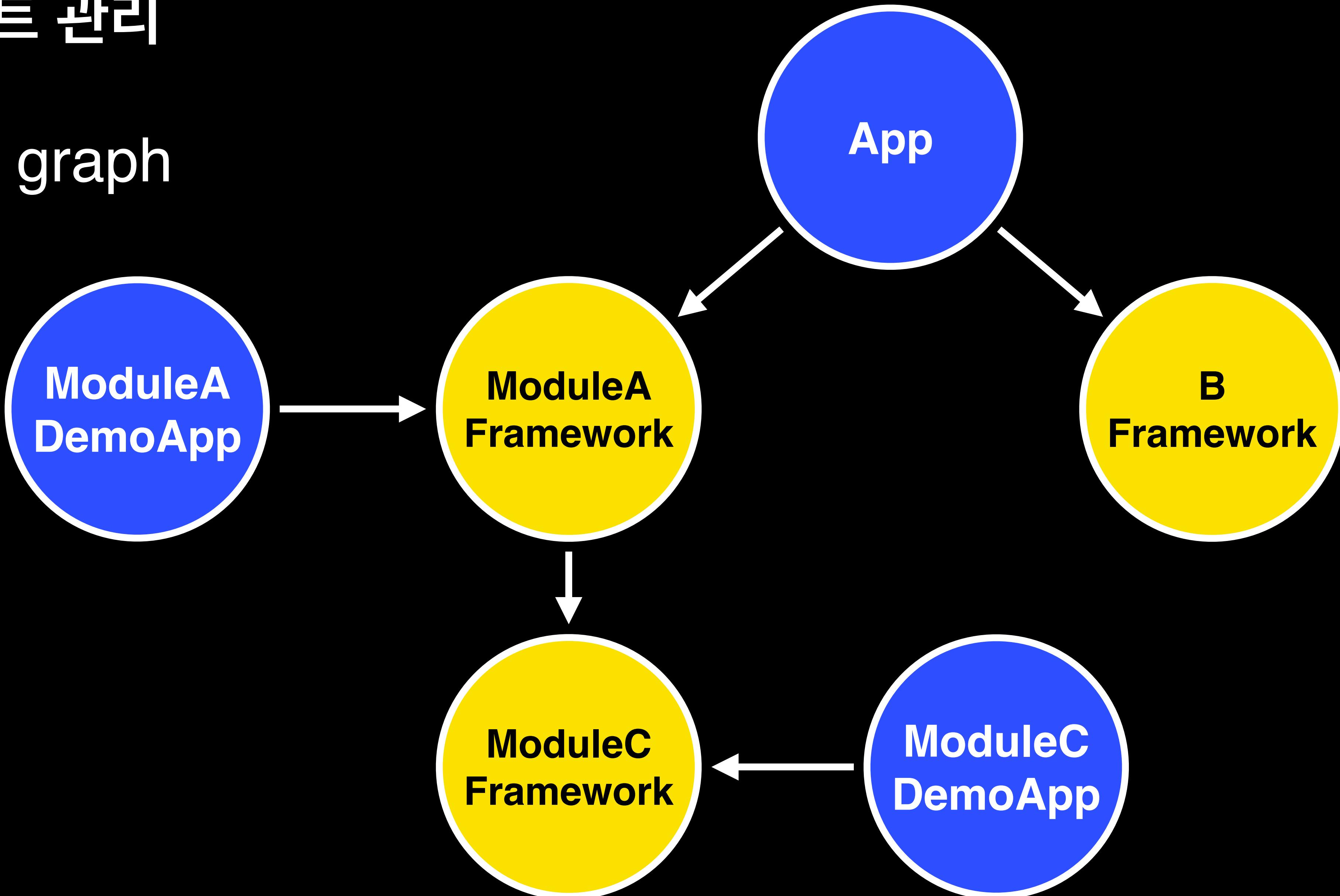
Add development assets here

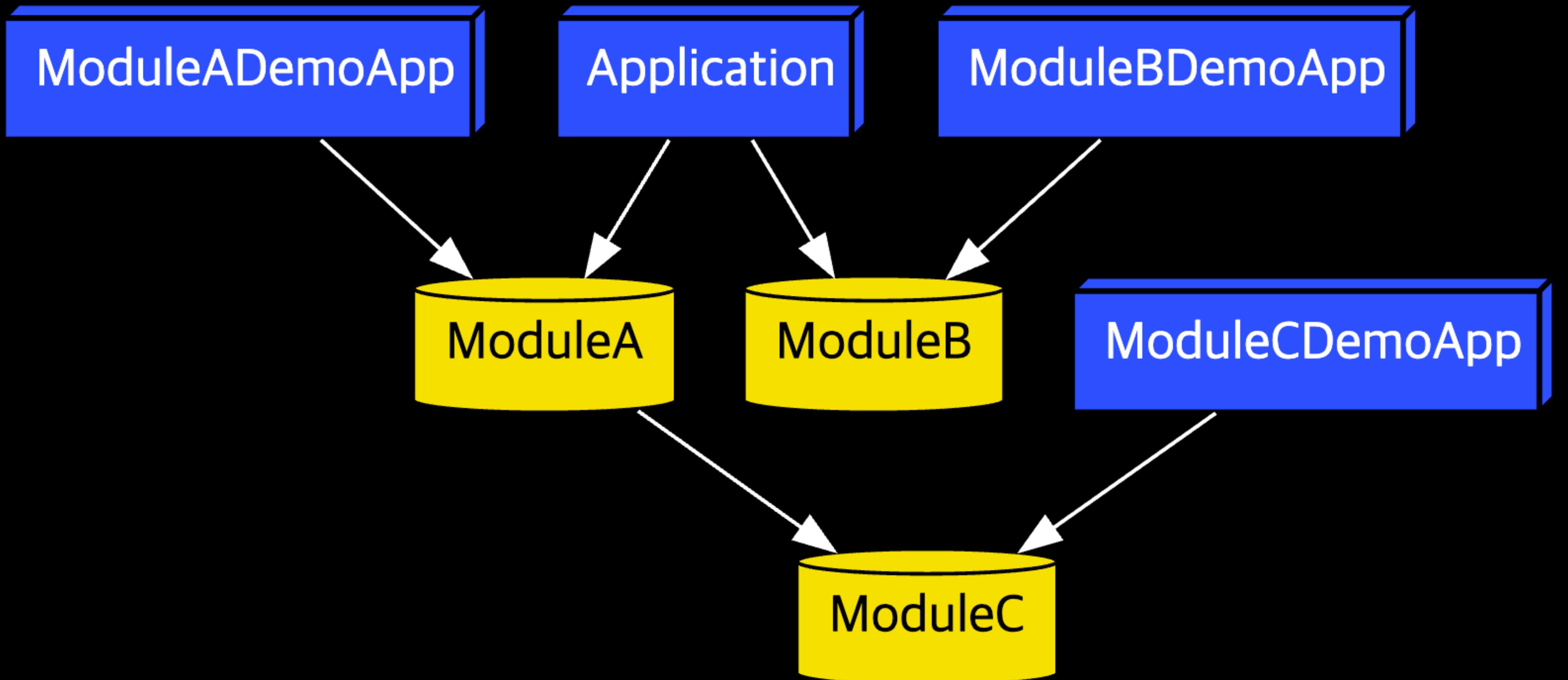
# 프로젝트 관리

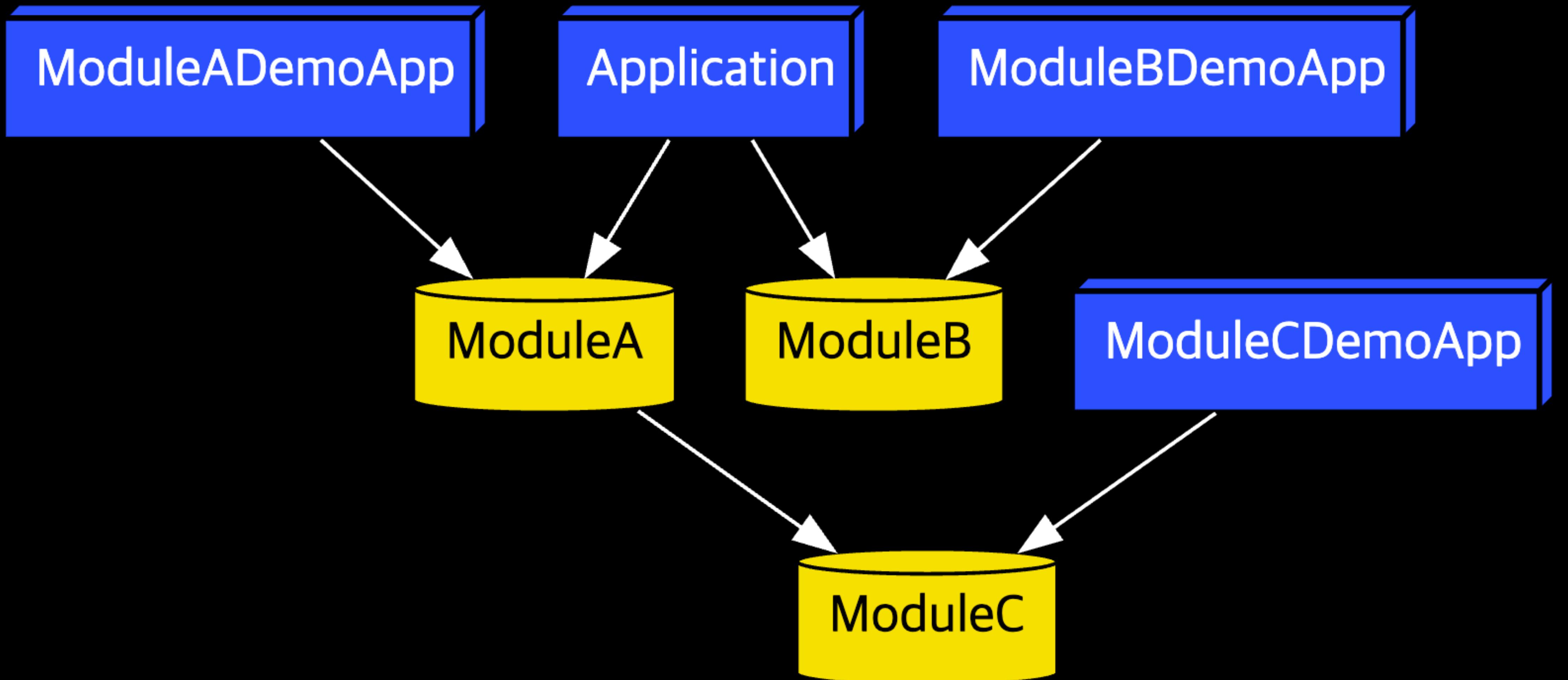


# 프로젝트 관리

\$ tuist graph



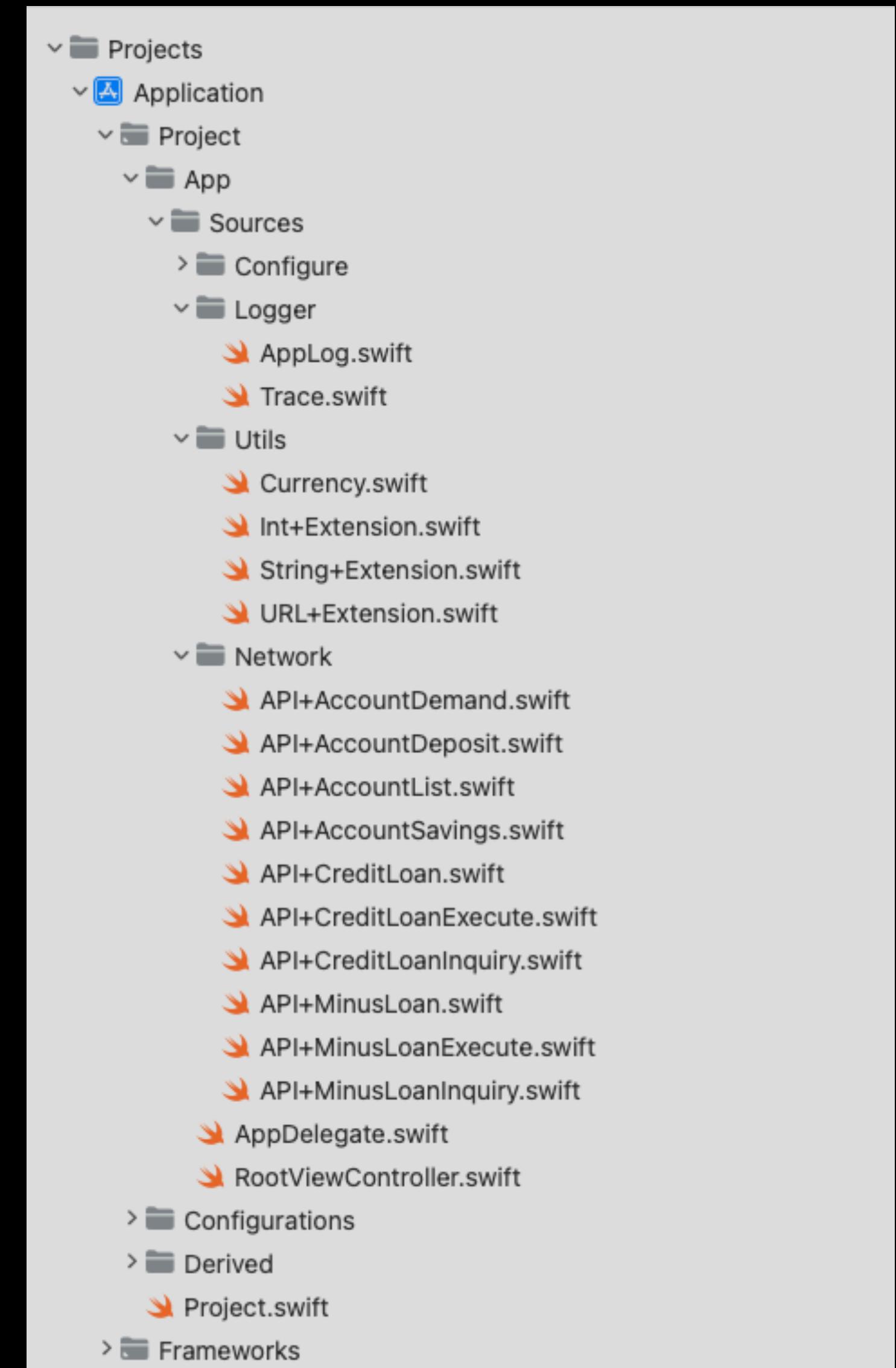




# 모듈 분리 시작해보기

# 모듈 분리 시작 순서

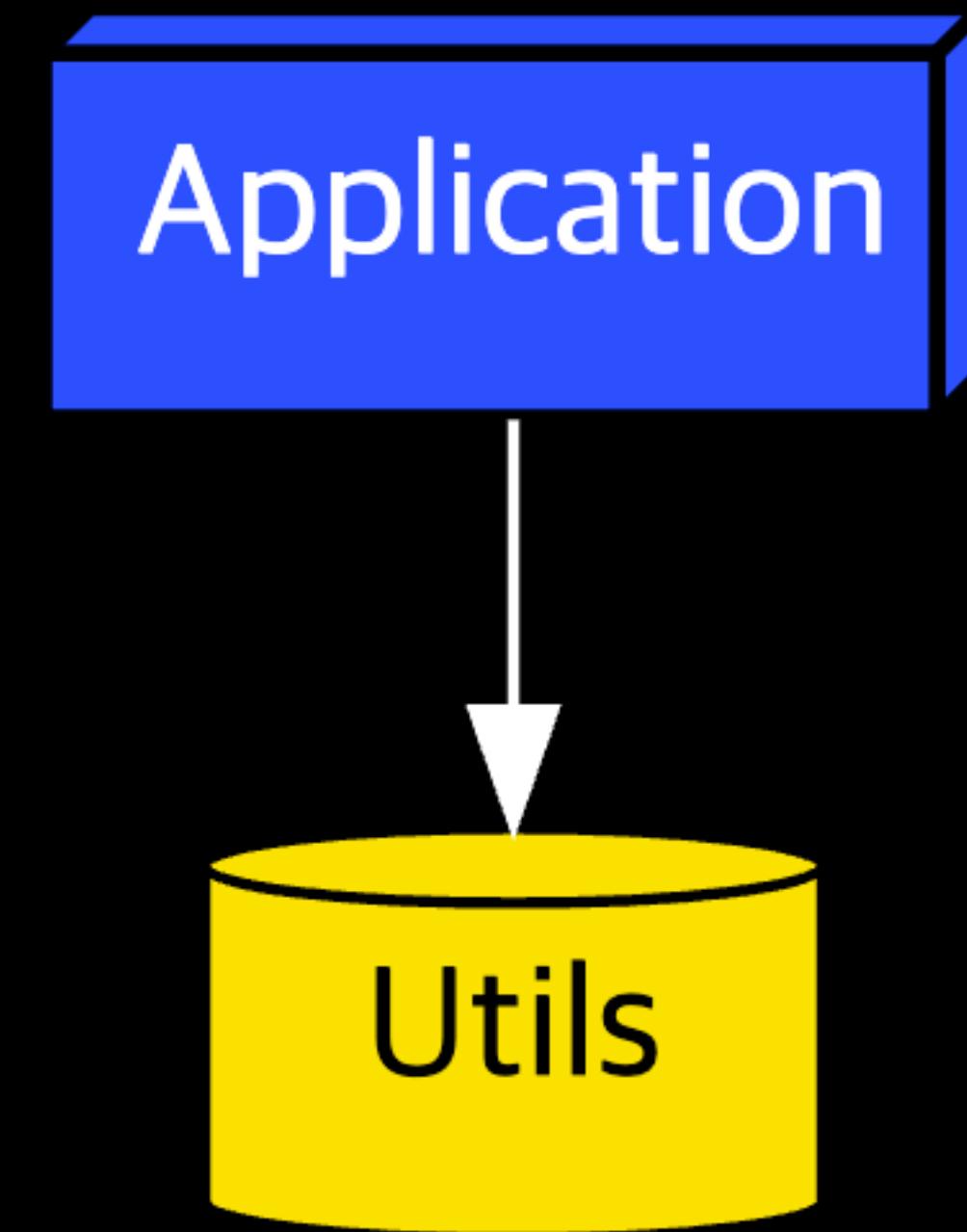
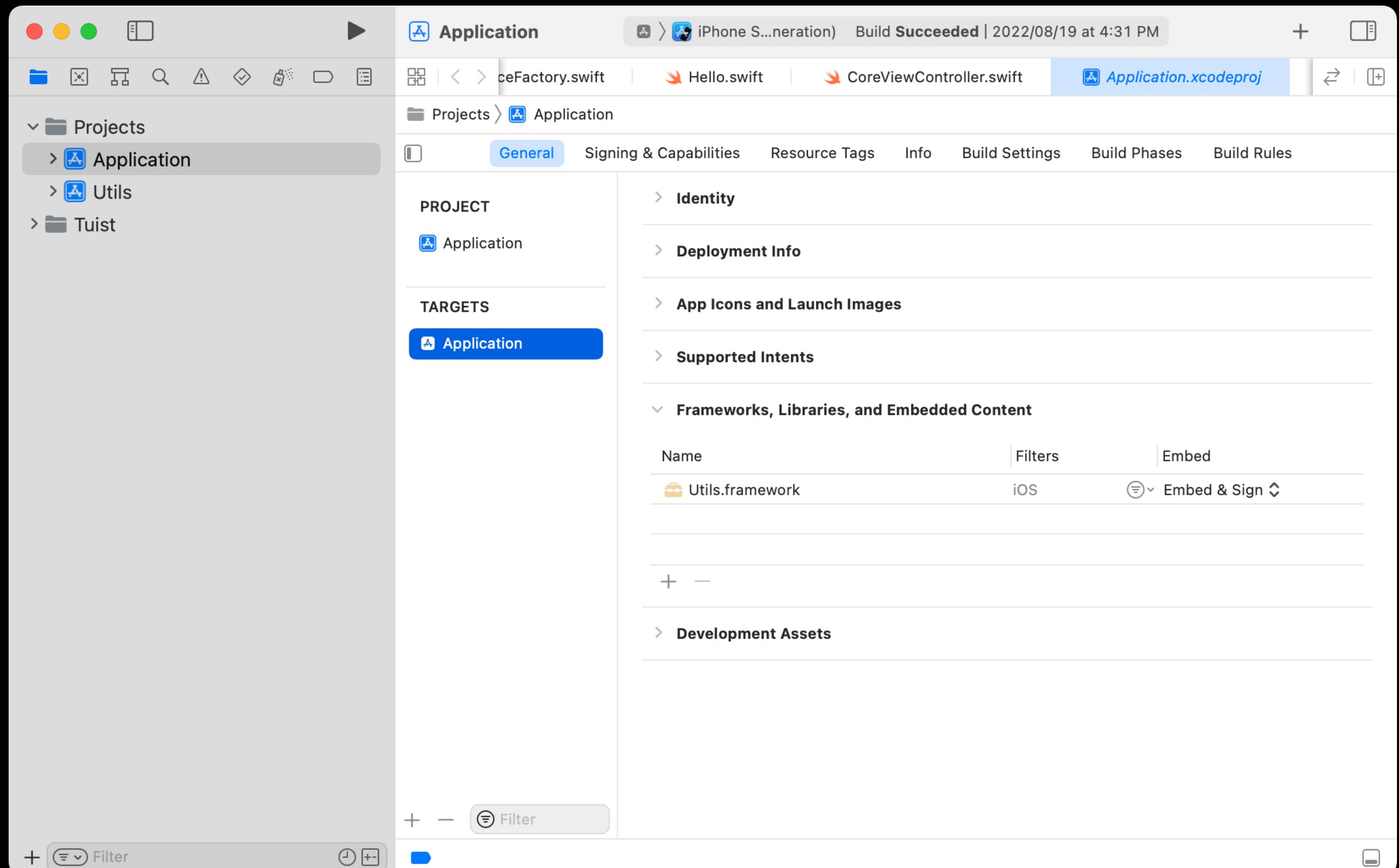
- 코드 정리
  - 유ти리티, 네트워크, 보안, 환경설정, 로그, UI 등의 폴더, 파일로 묶어서 정리



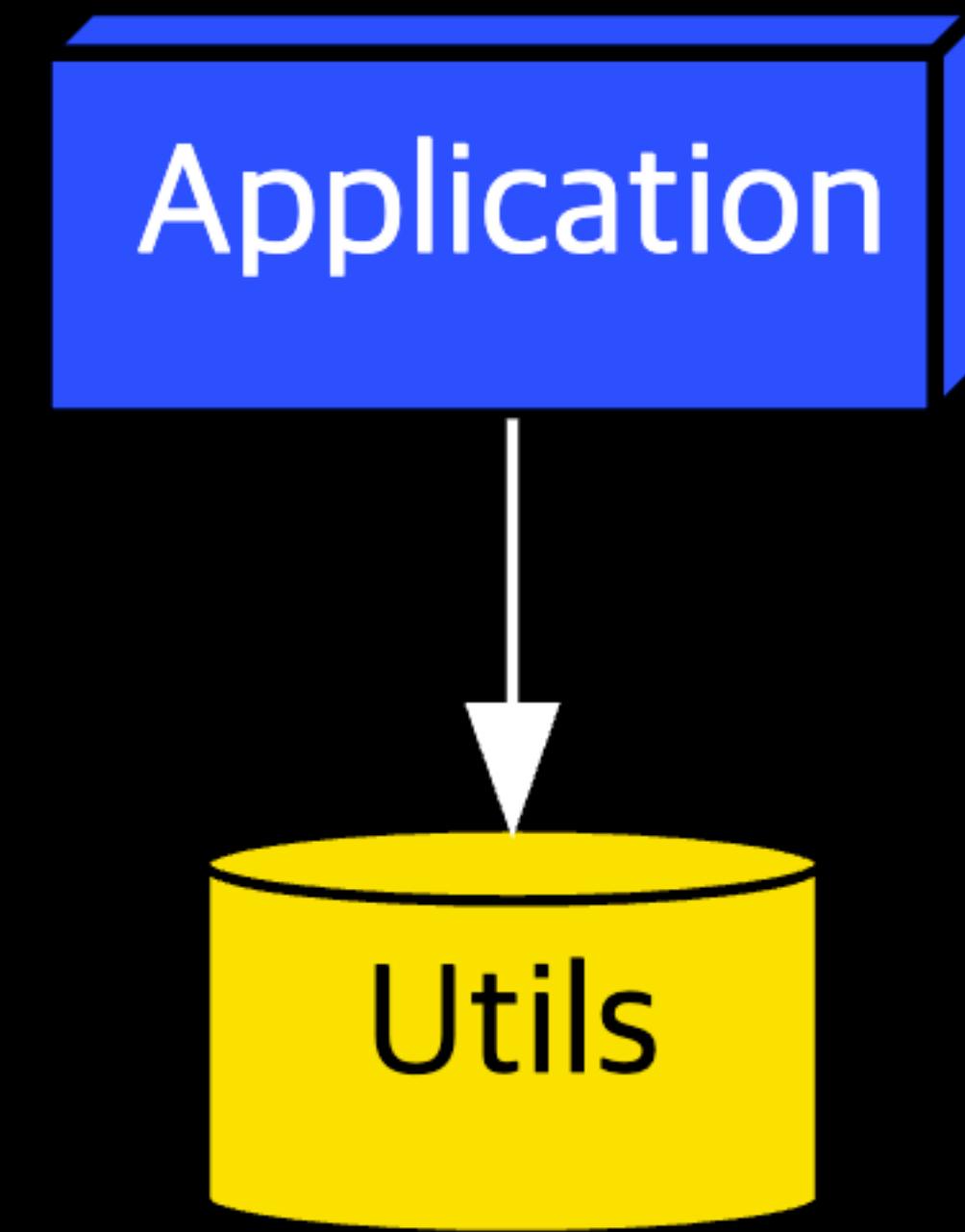
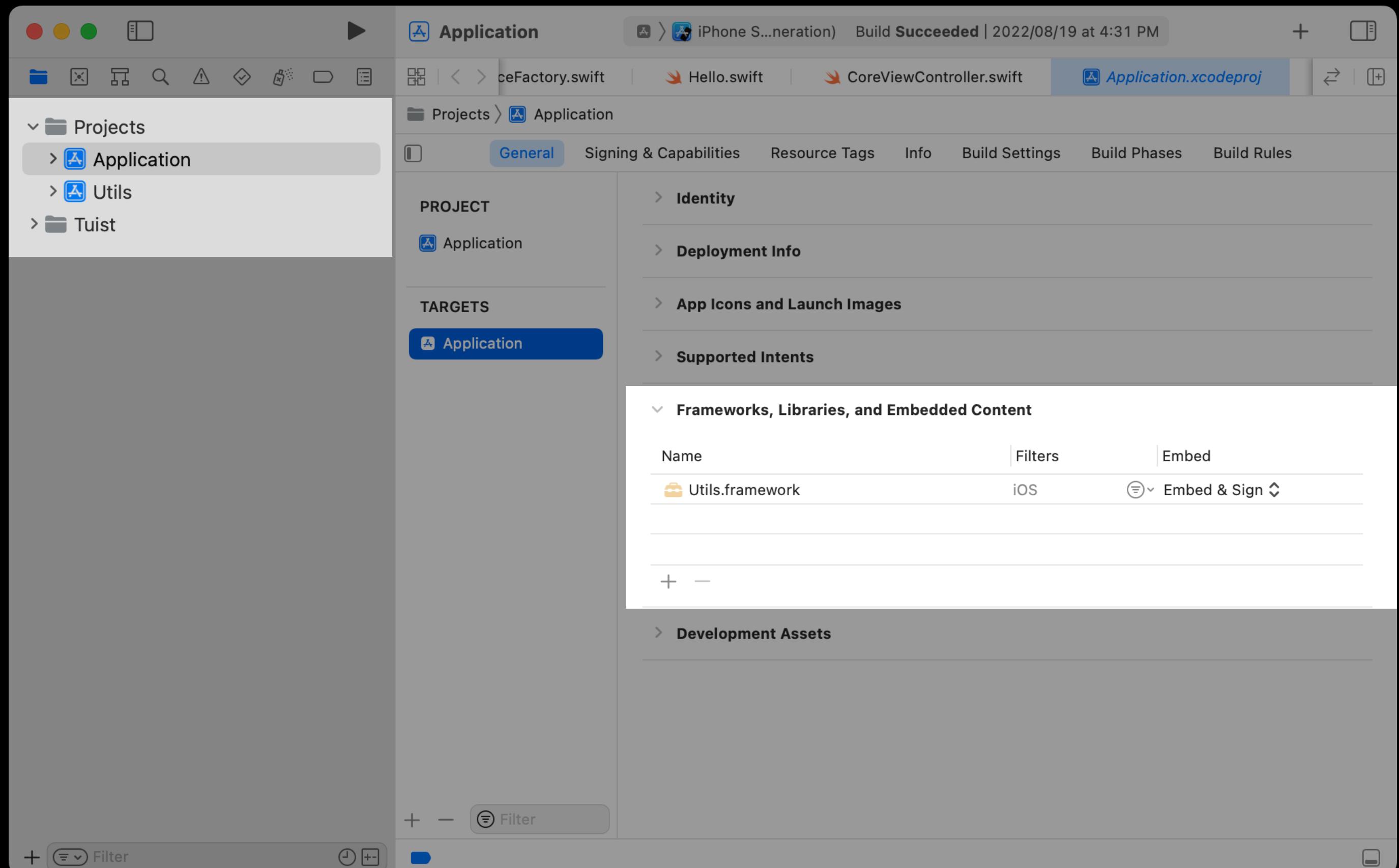
# 모듈 분리 시작 순서

- Tuist를 활용하여 프로젝트 생성
  - 의존성이 적은 코드부터 새로운 프로젝트로 단계별 이전

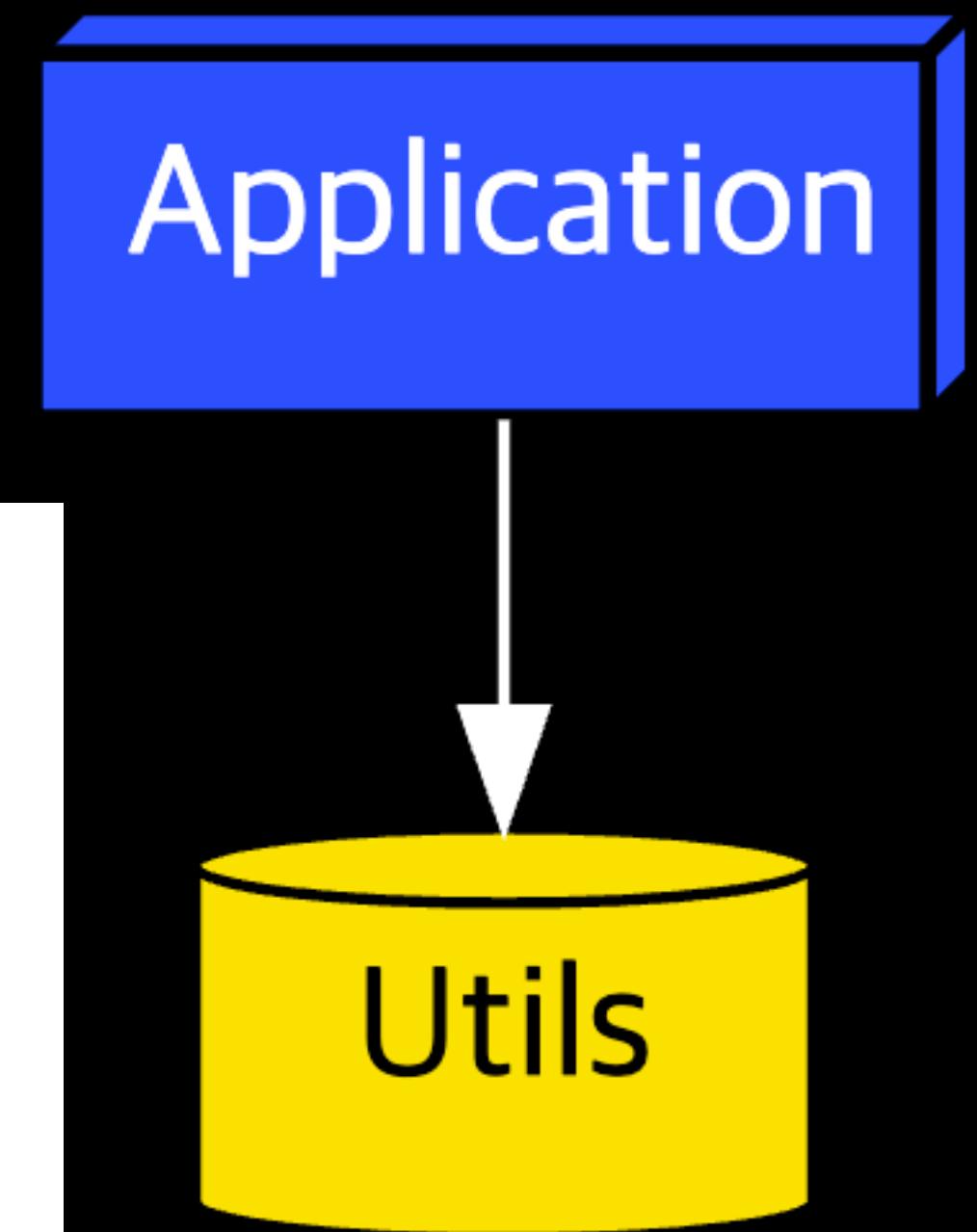
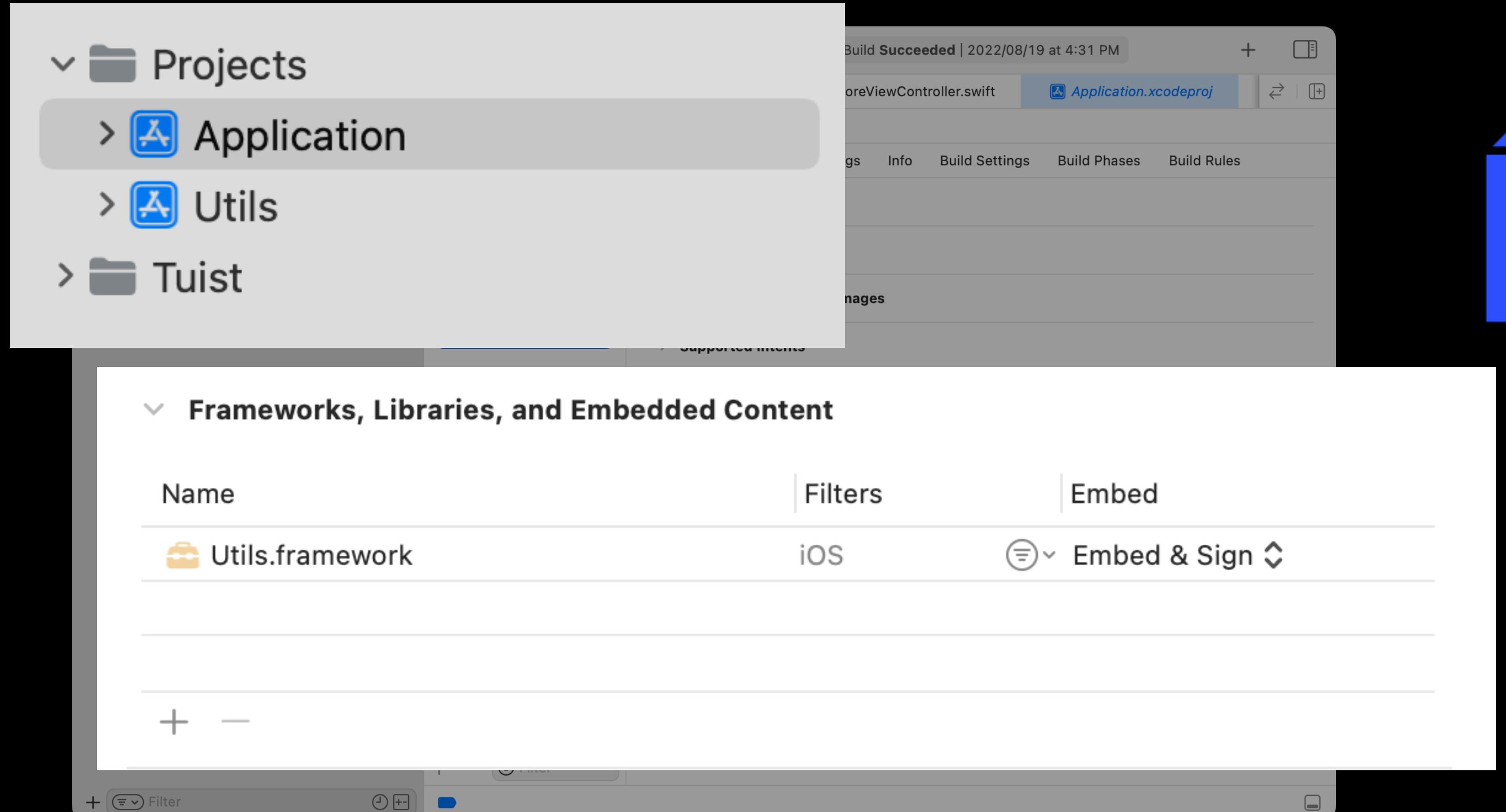
# 모듈 분리 작업



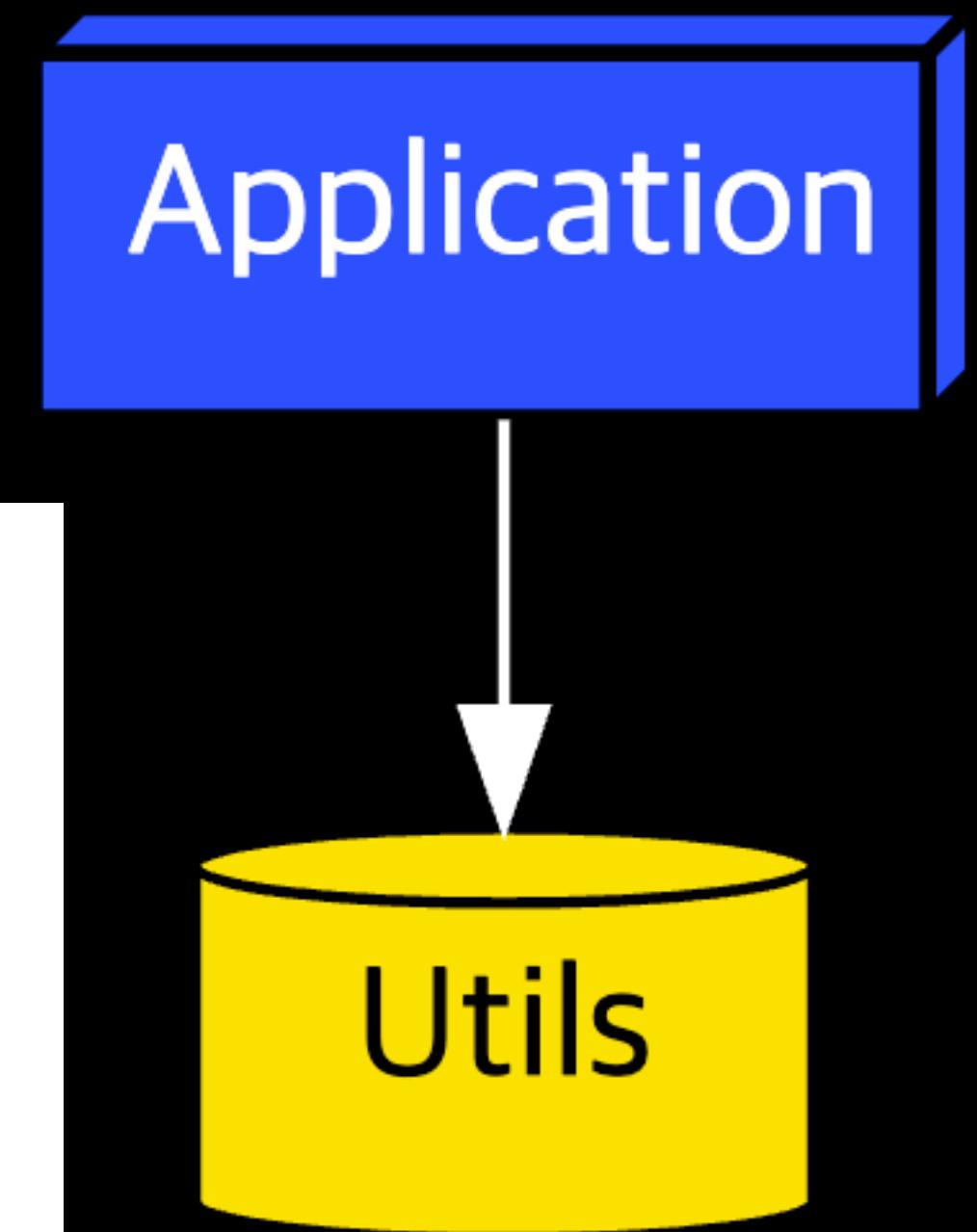
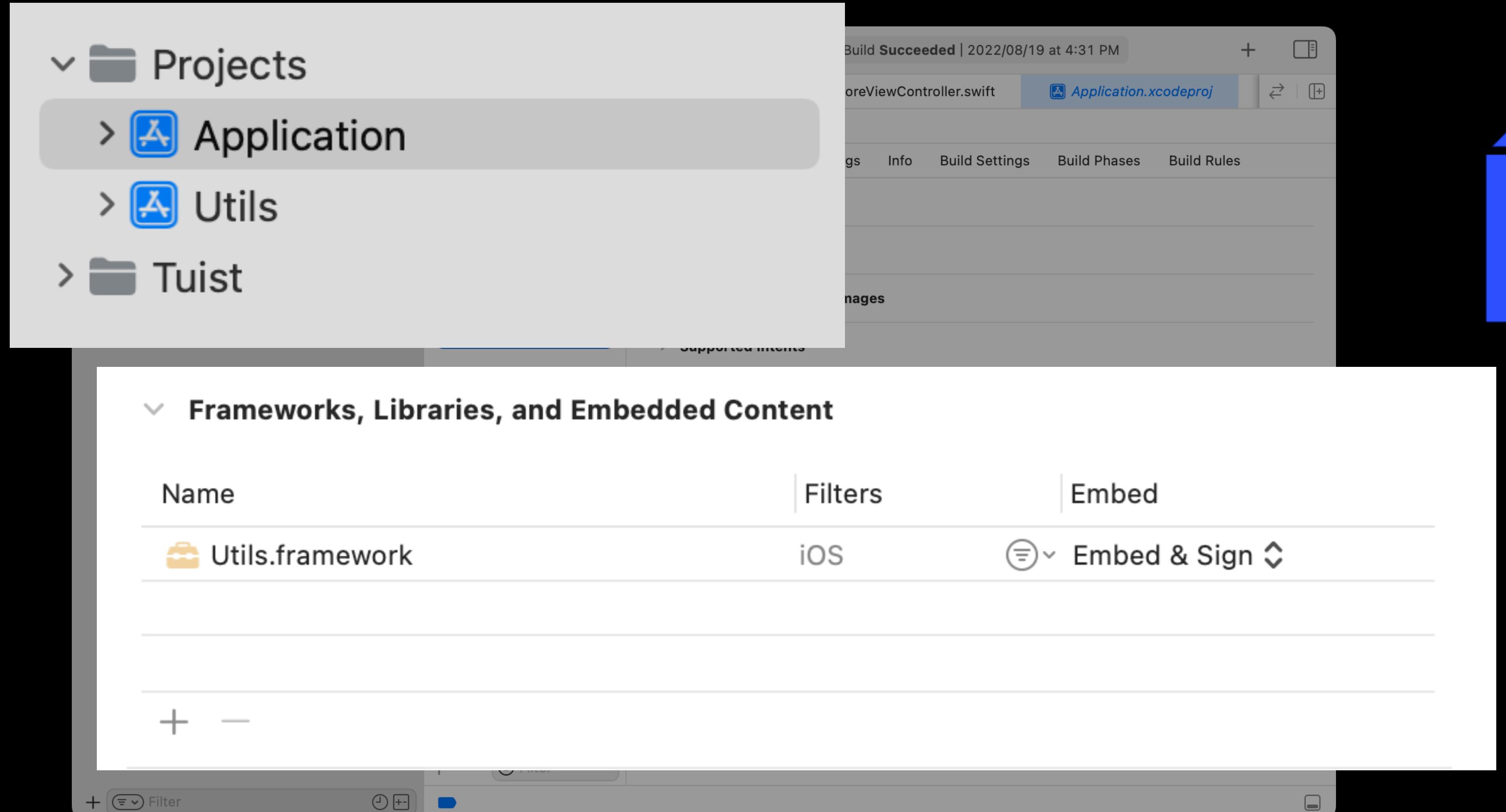
# 모듈 분리 작업



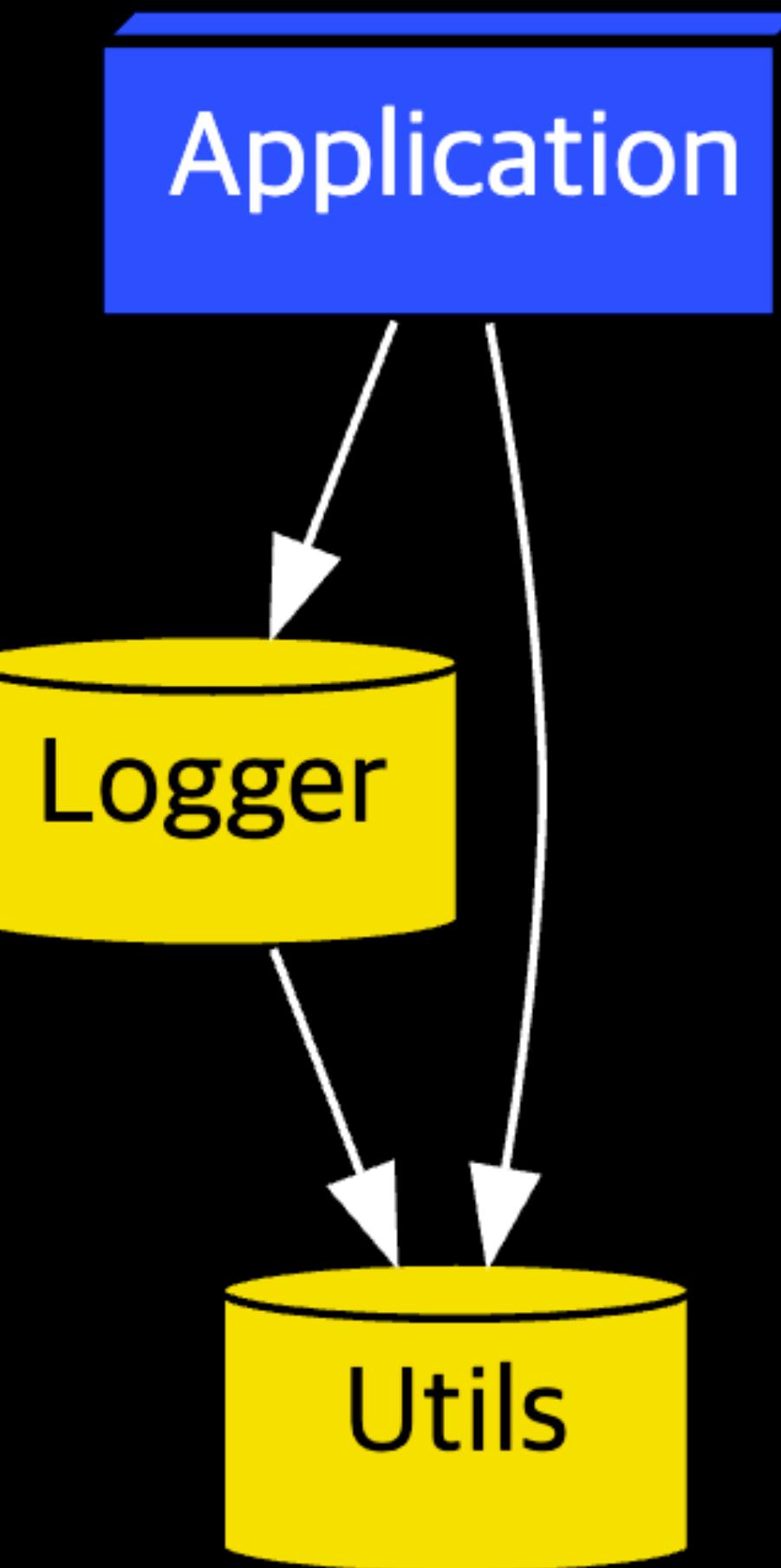
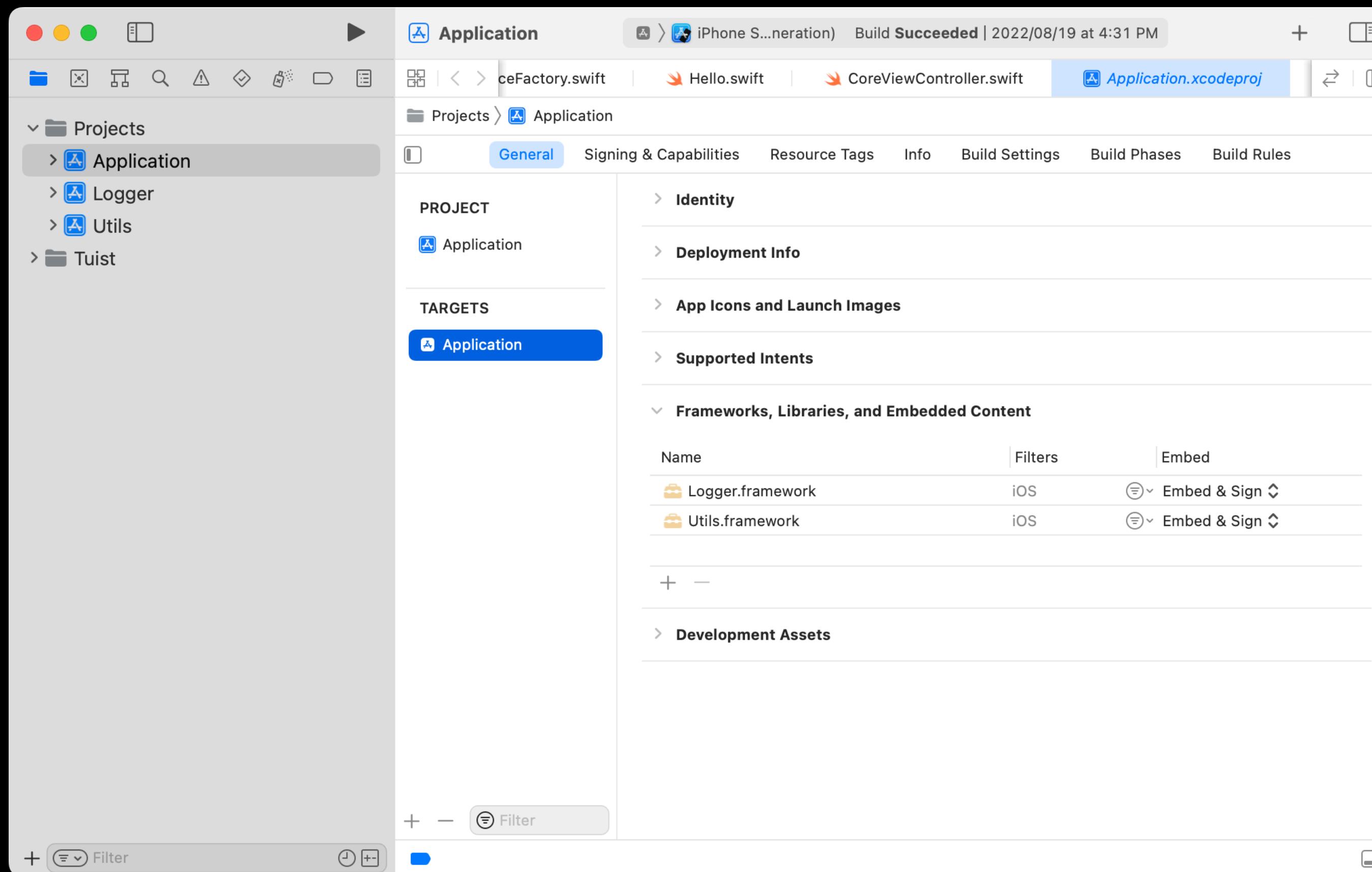
# 모듈 분리 작업



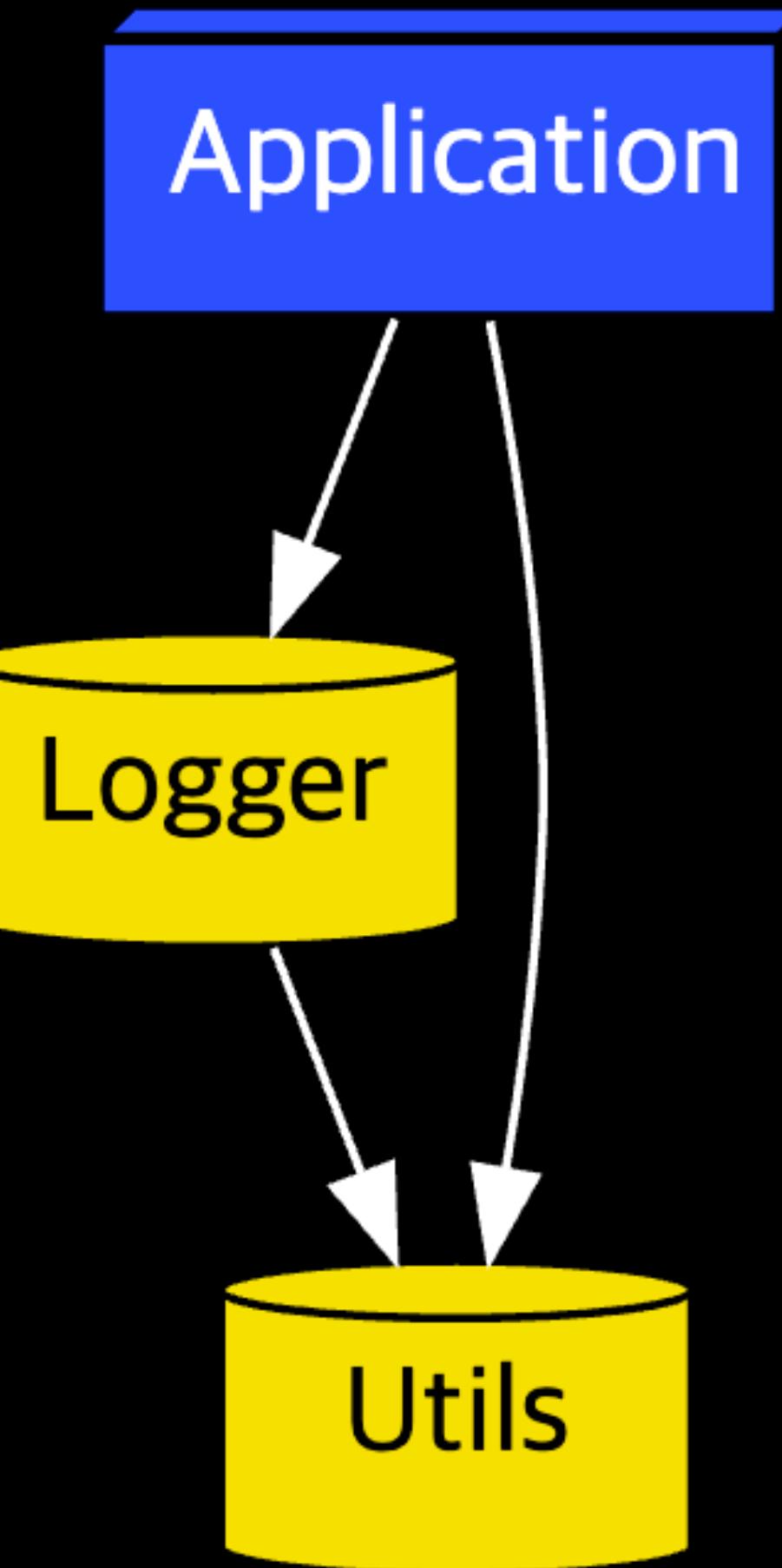
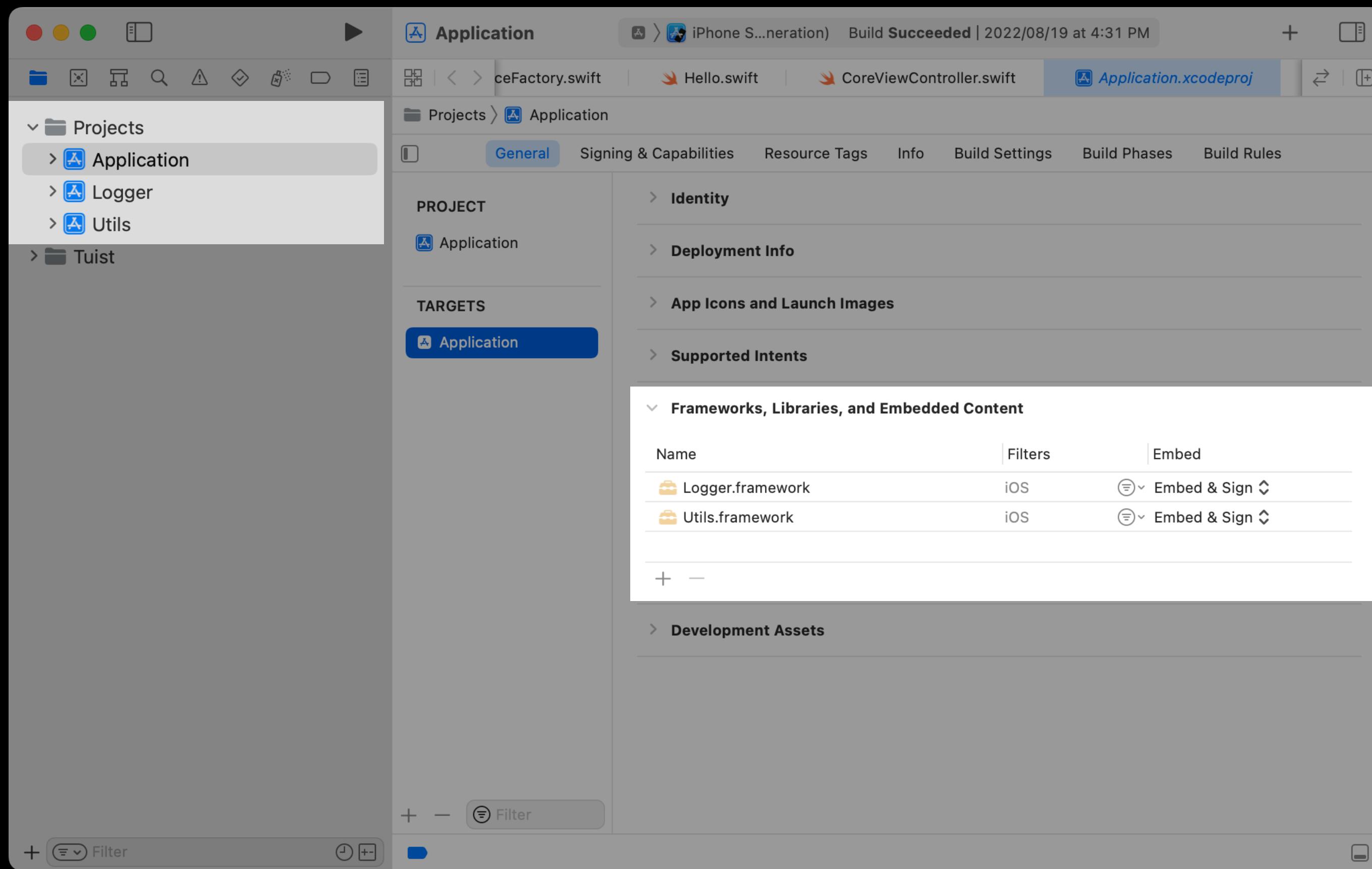
# 모듈 분리 작업



# 모듈 분리 작업

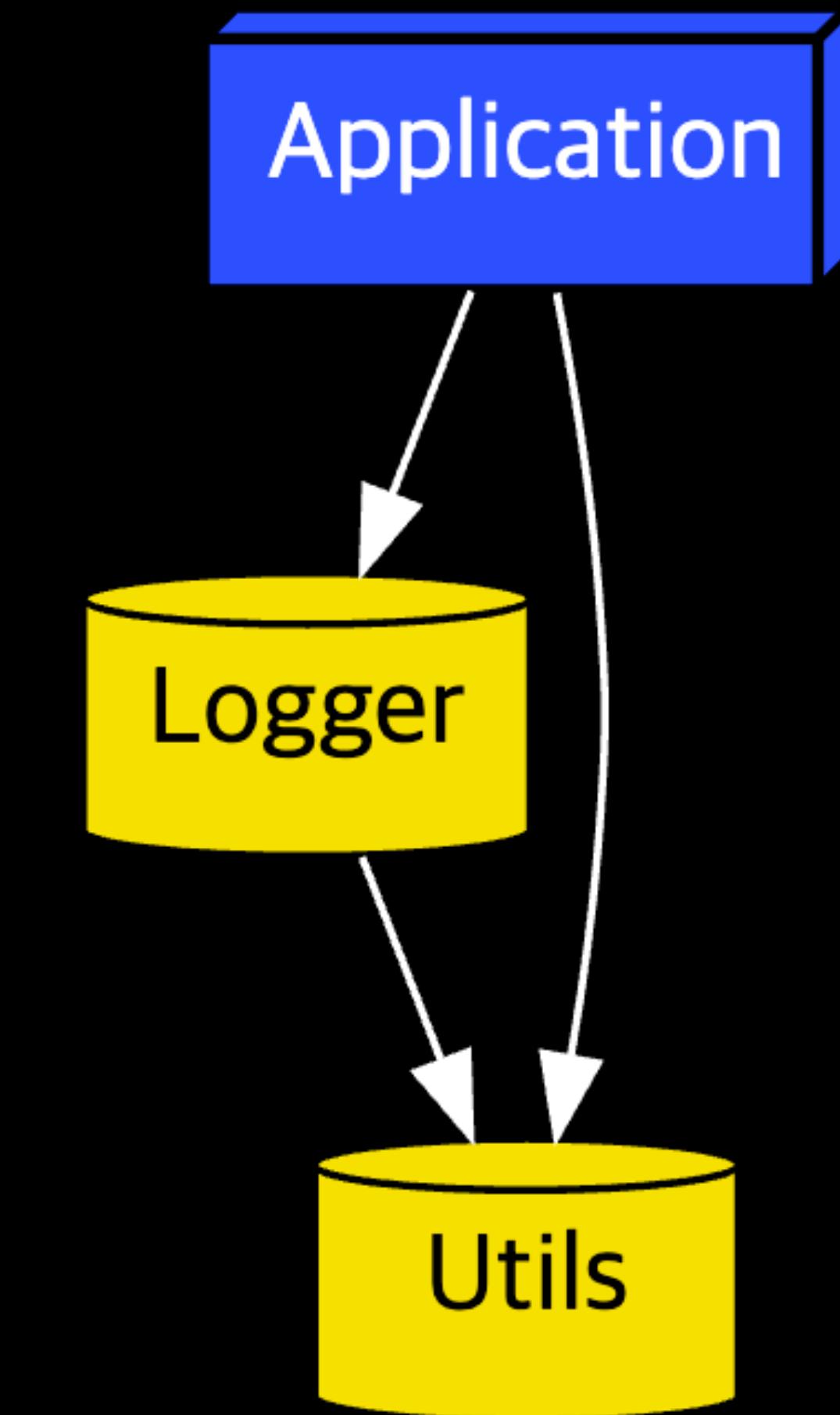


# 모듈 분리 작업

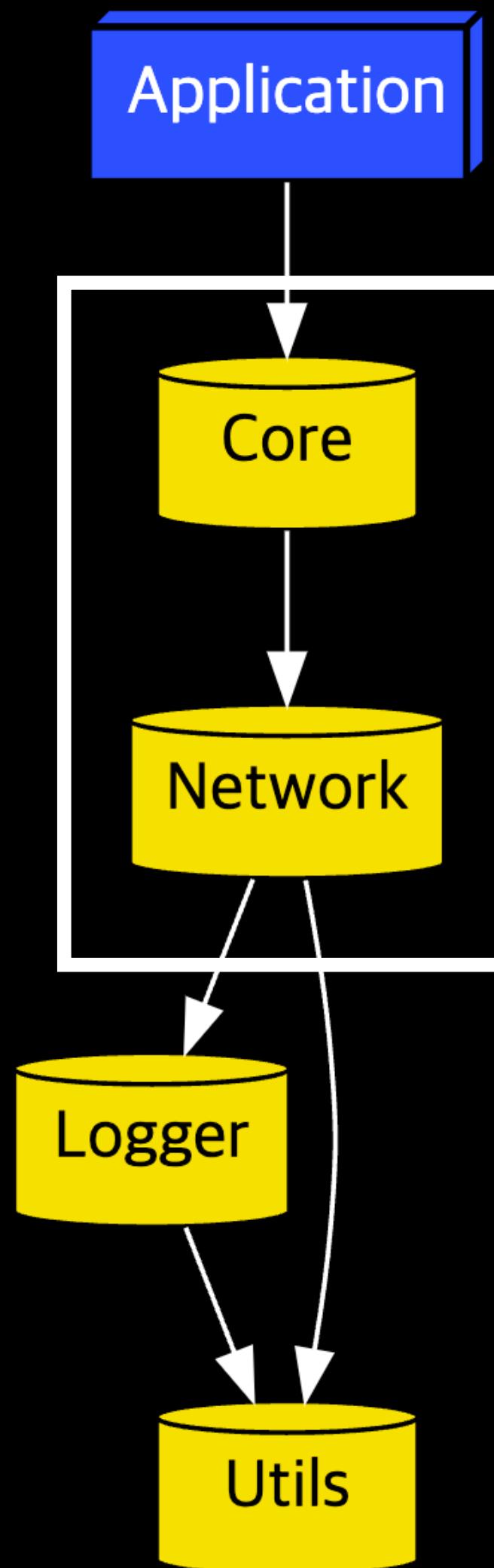
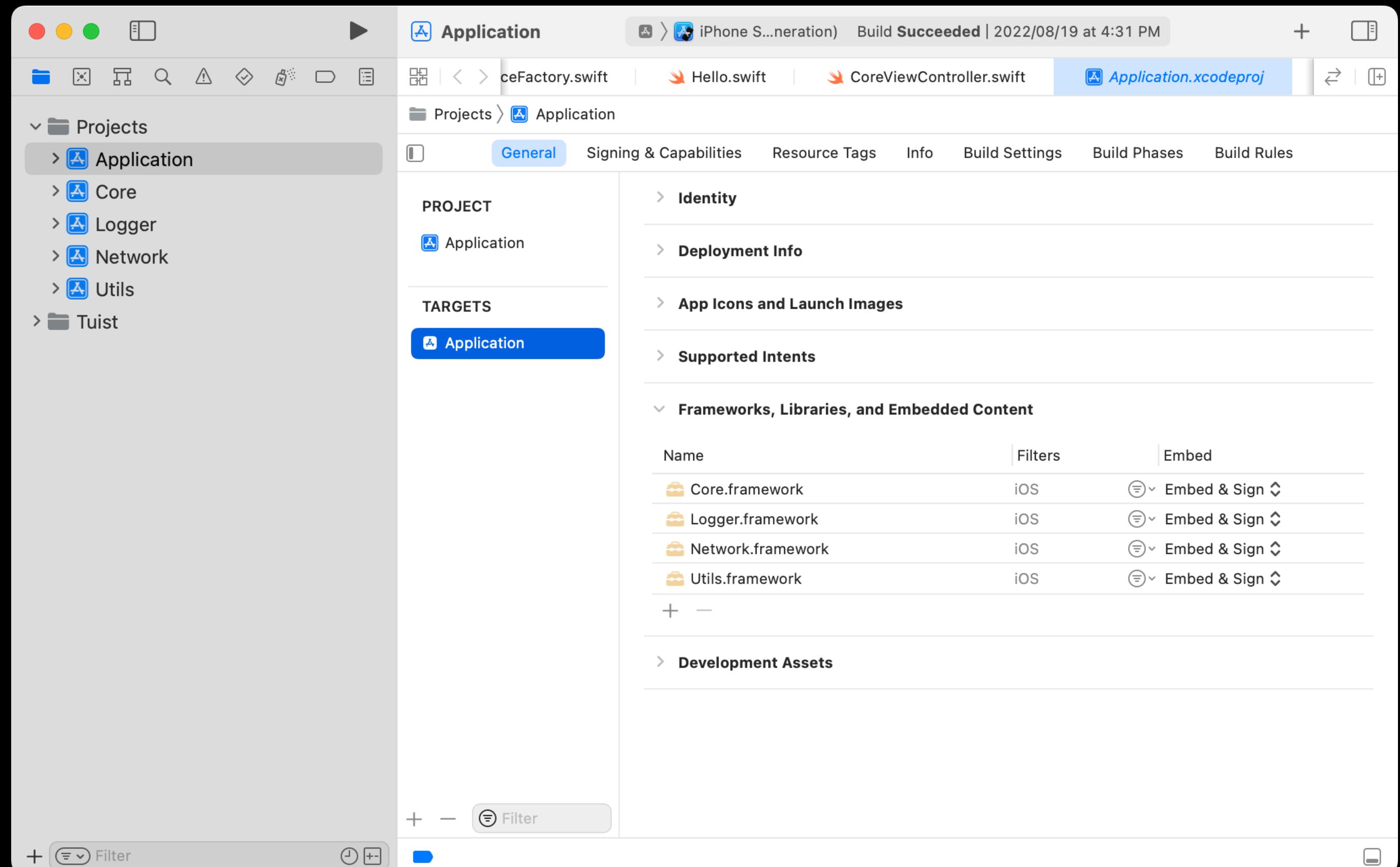


# 모듈 분리 작업

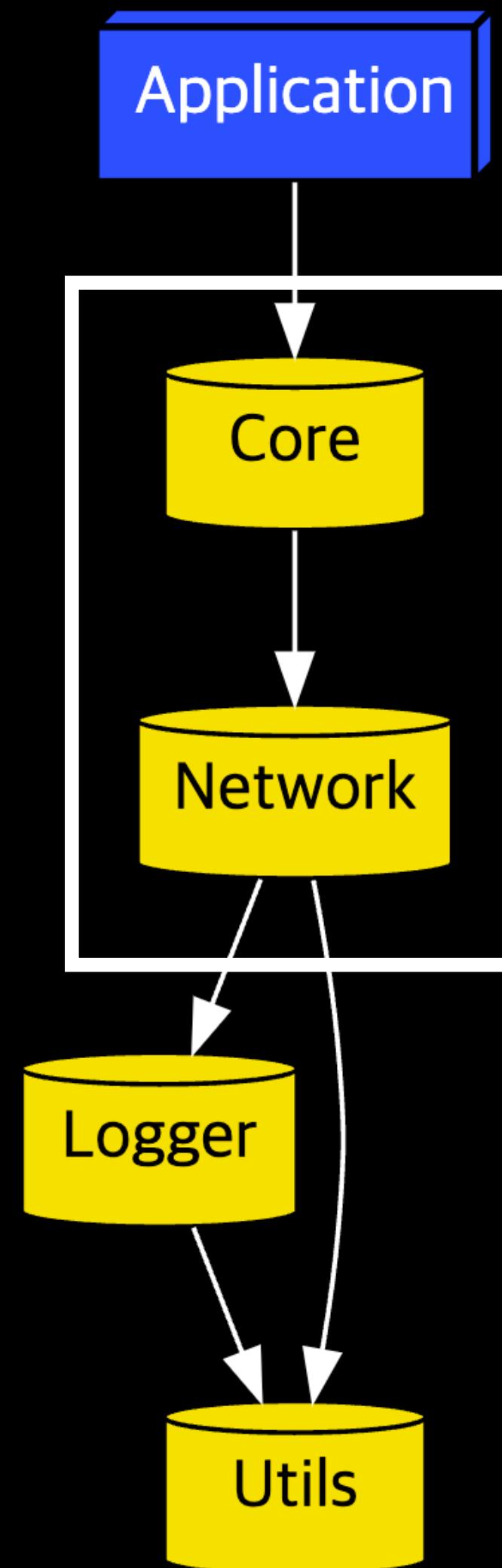
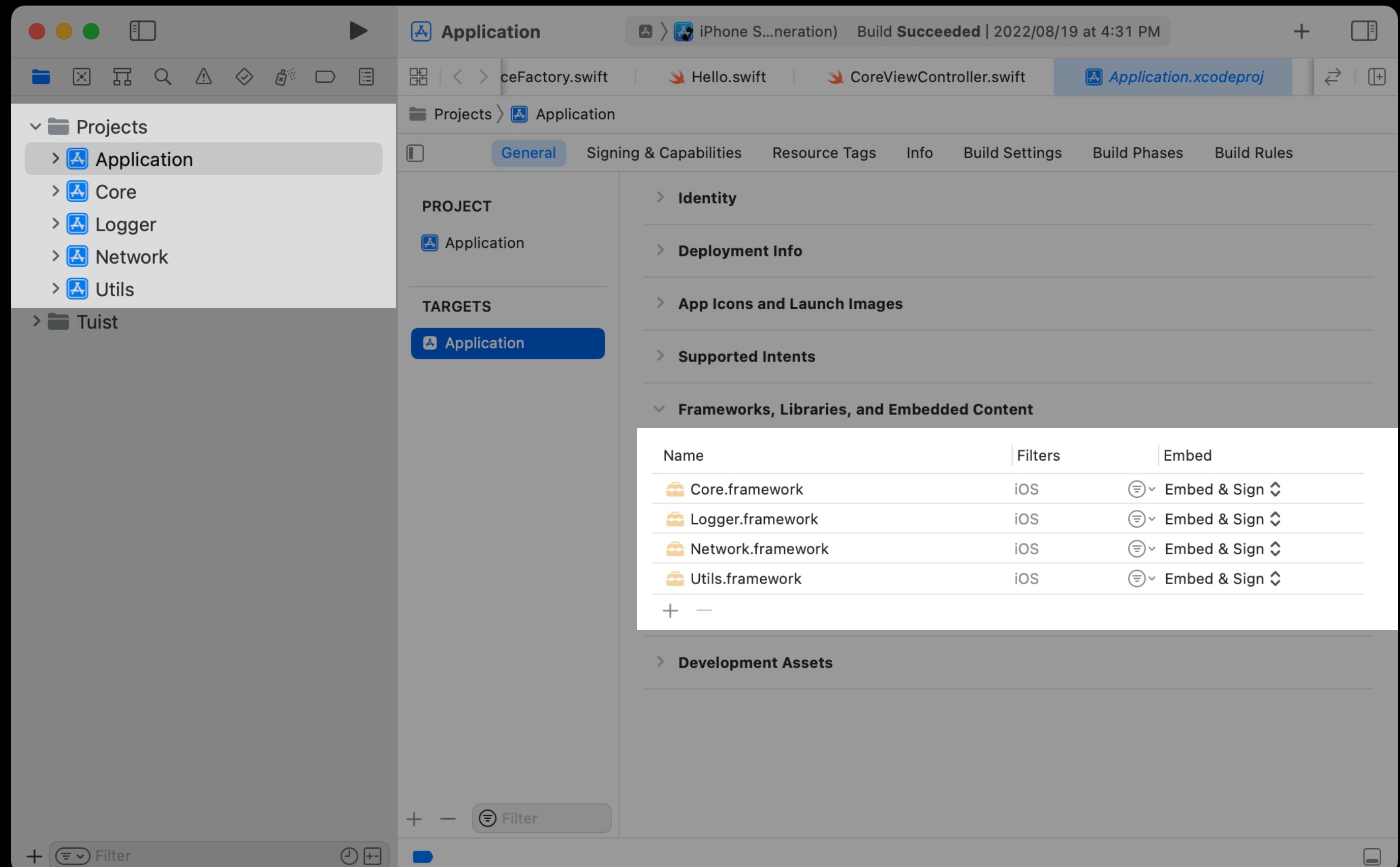
The screenshot shows the Xcode interface. On the left, the Project Navigator displays three projects: Application, Logger, and Utils. The Application project is selected. In the center, the Editor area shows the build log: "Build Succeeded | 2022/08/19 at 4:31 PM" and the file "MoreViewController.swift". Below the log are tabs for "Logs", "Info", "Build Settings", "Build Phases", and "Build Rules". The bottom tab bar has "Application" selected. On the right, the Utilities area shows the "Frameworks, Libraries, and Embedded Content" section. It lists two frameworks: "Logger.framework" and "Utils.framework", both of which are embedded and signed for iOS. There are also "+" and "-" buttons for adding or removing frameworks.



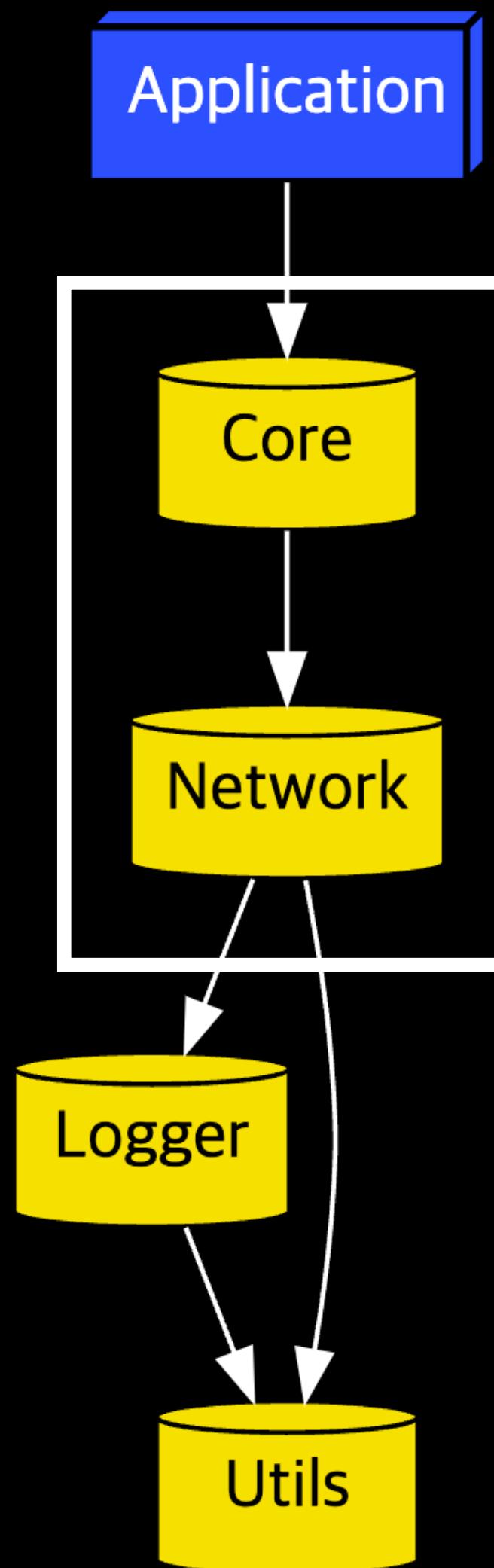
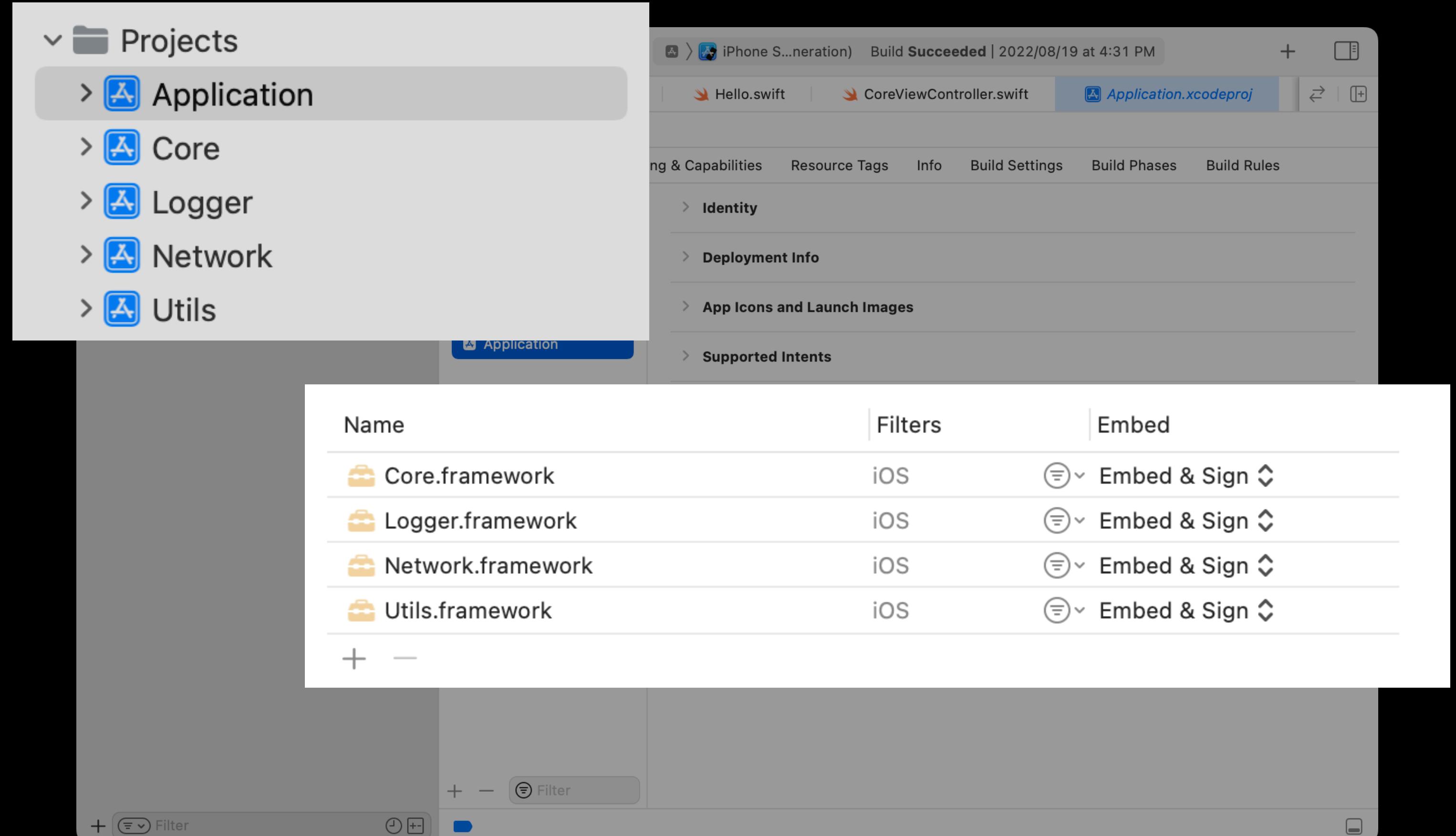
# 모듈 분리 작업



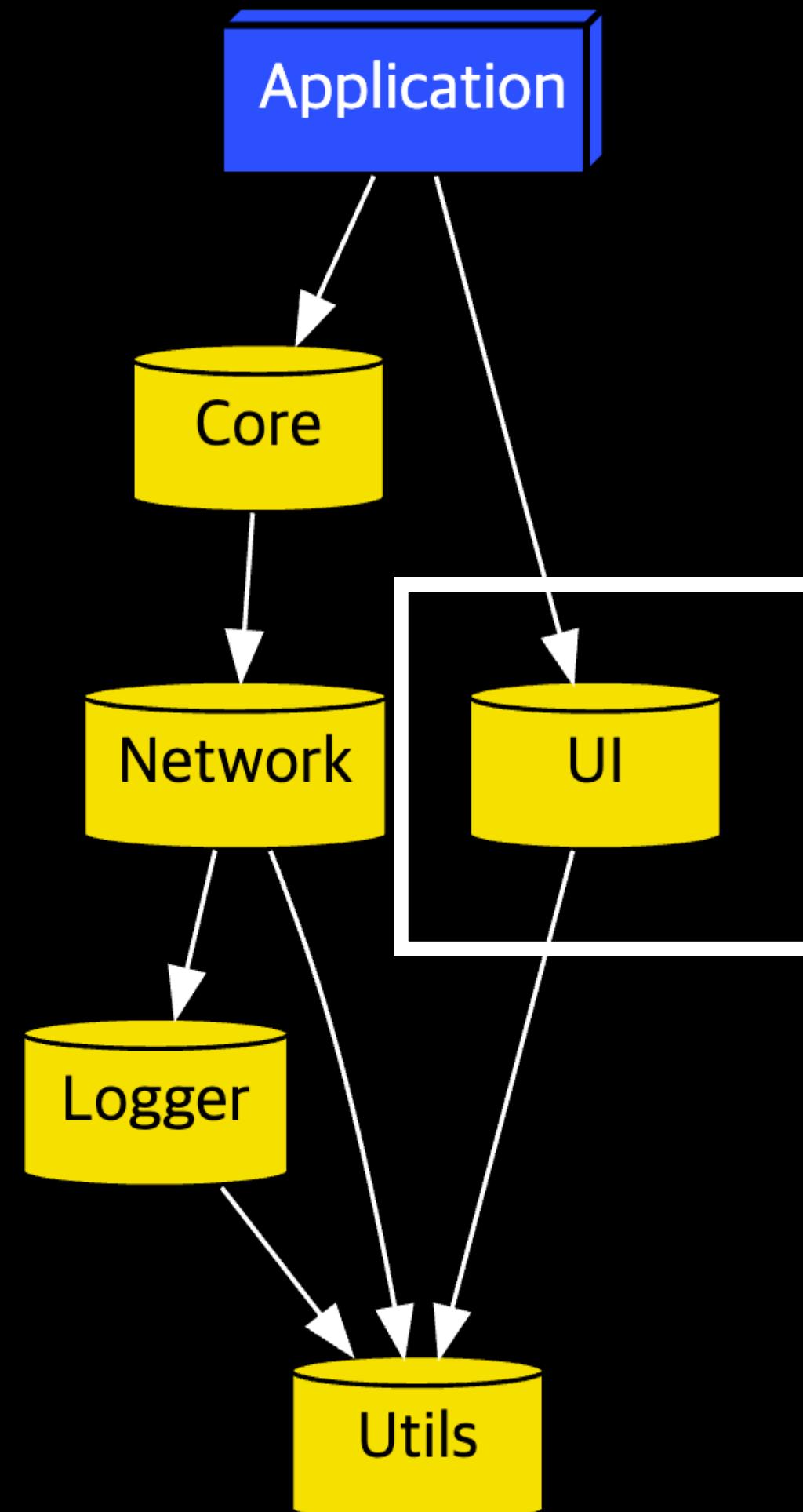
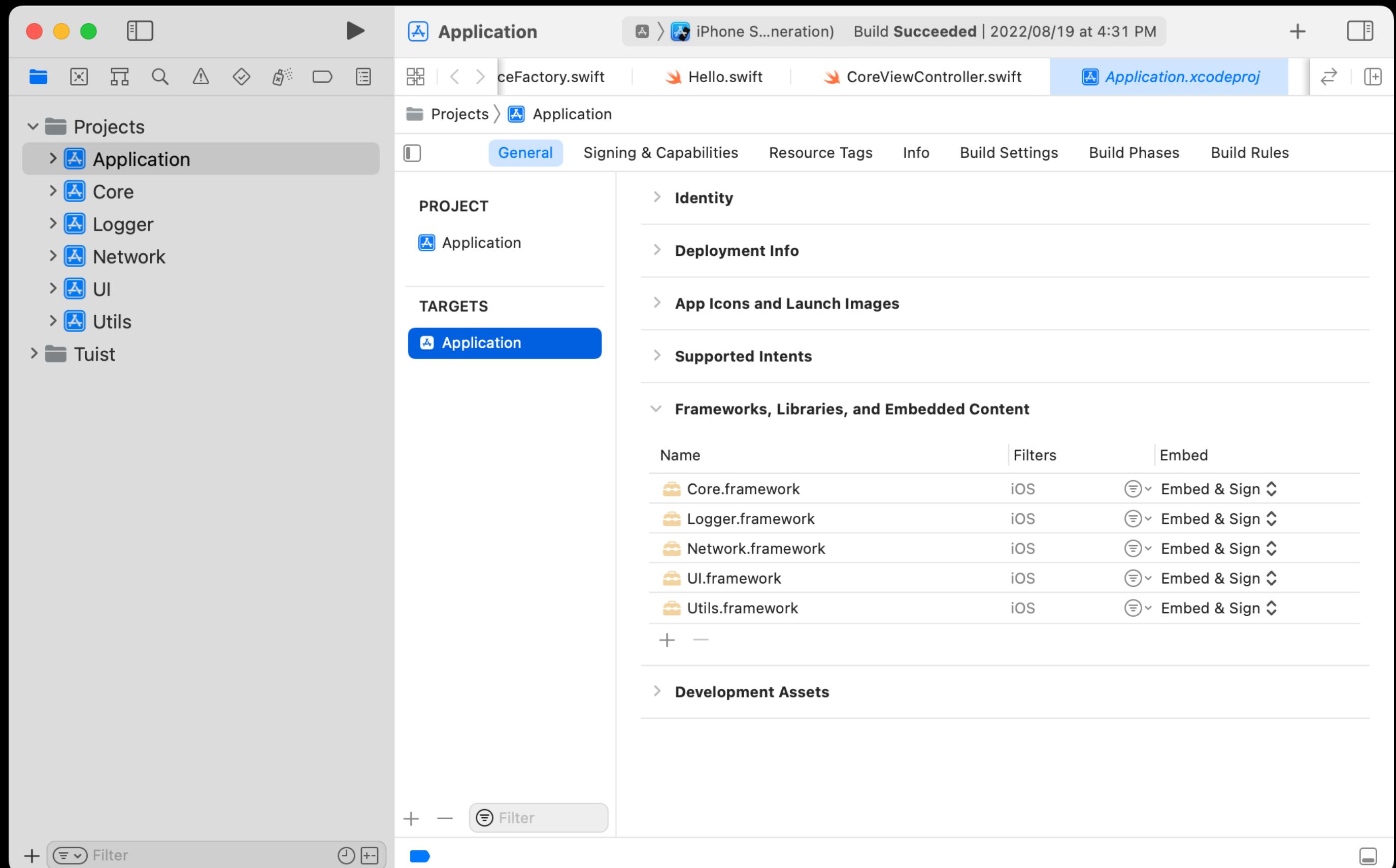
# 모듈 분리 작업



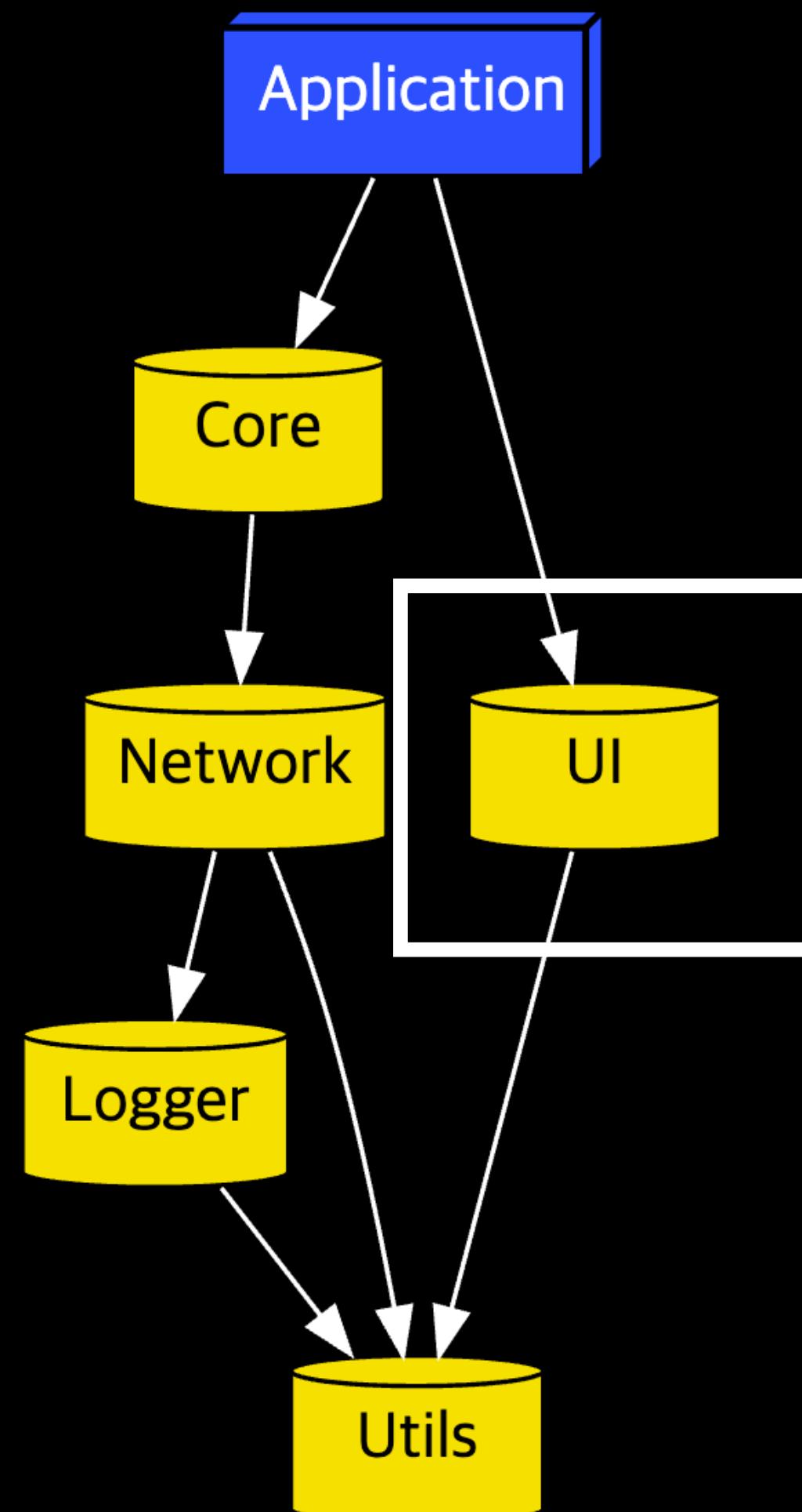
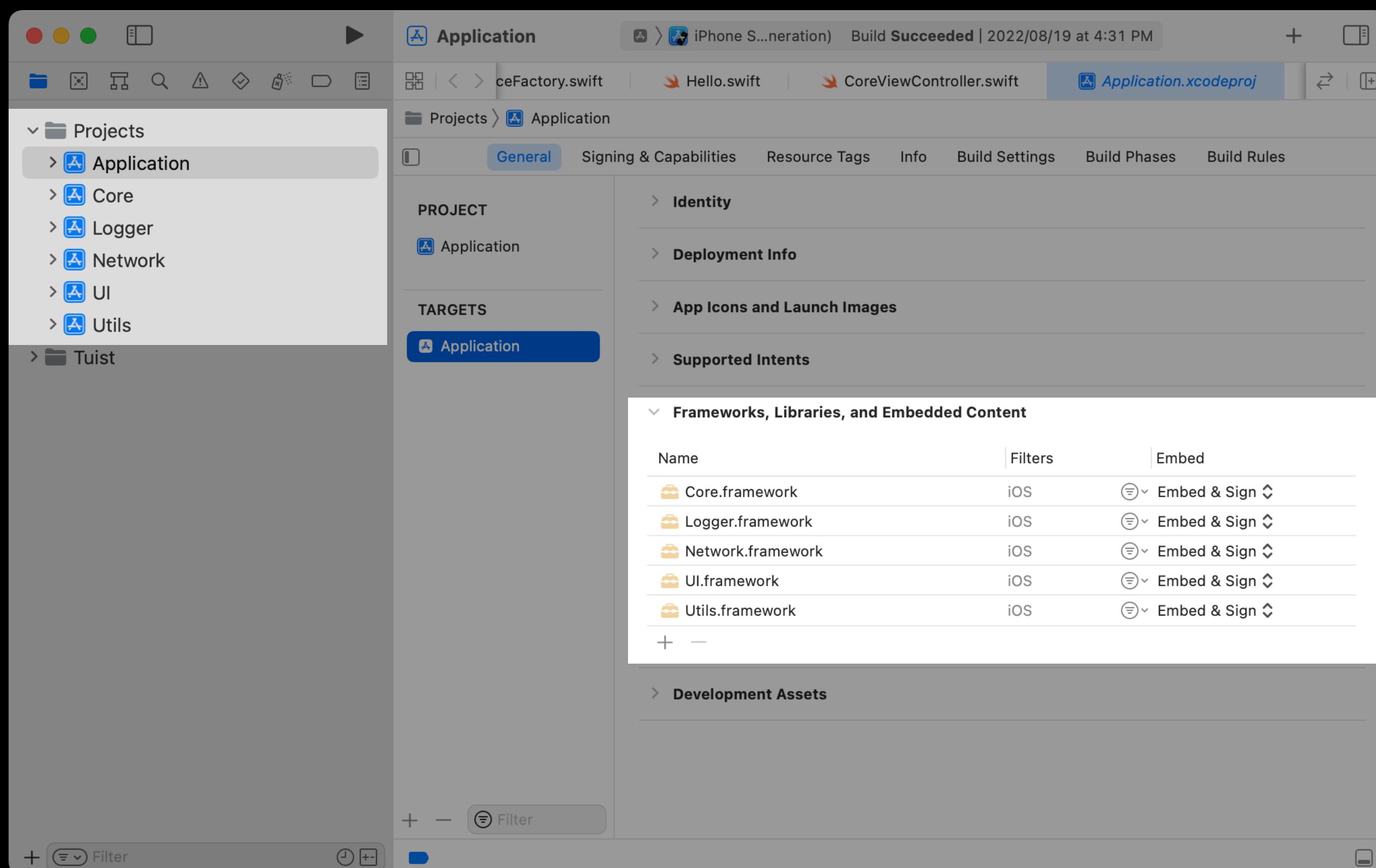
# 모듈 분리 작업



# 모듈 분리 작업



# 모듈 분리 작업

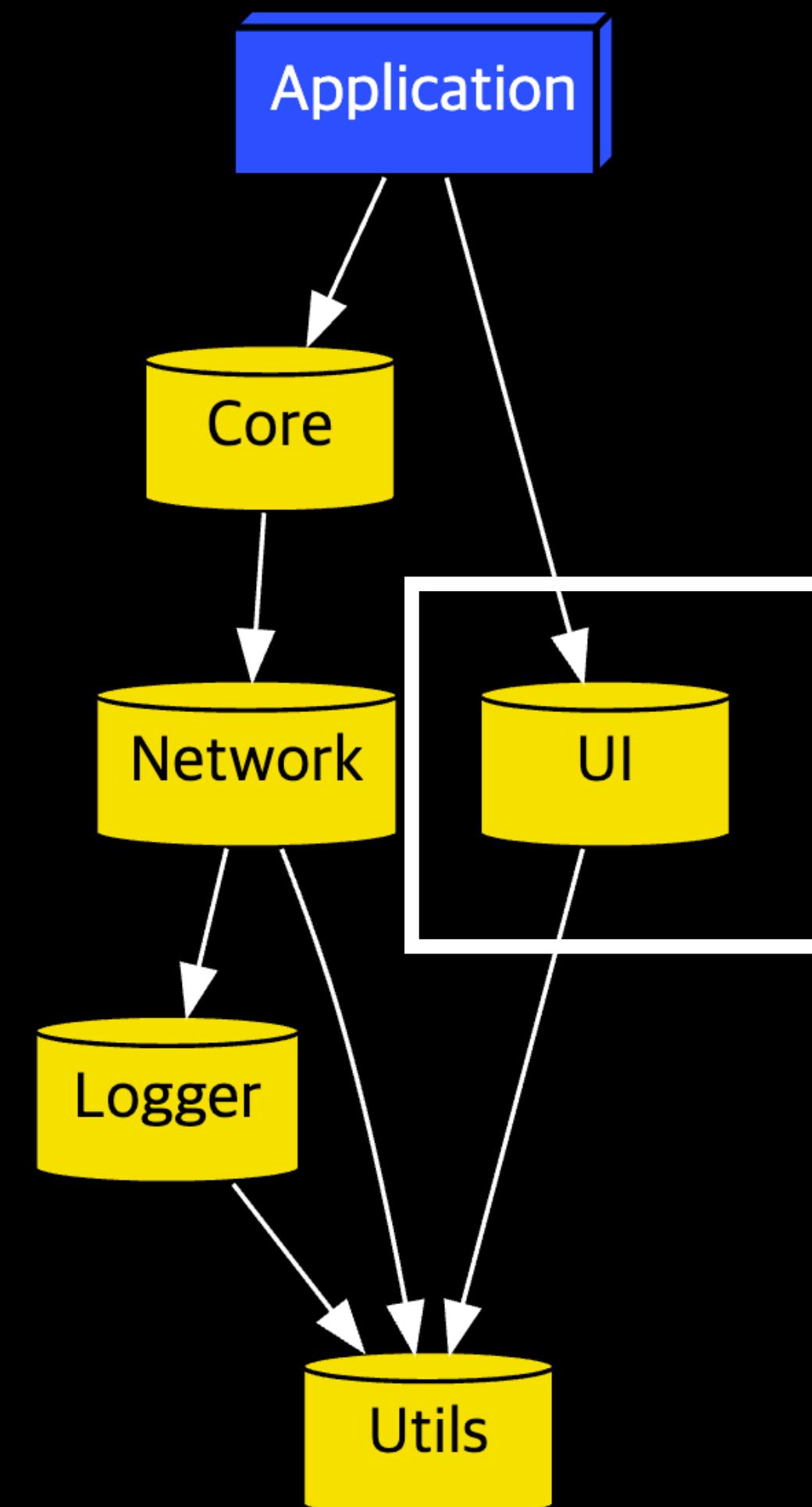


# 모듈 분리 작업

The screenshot shows the Xcode interface with the following details:

- Projects:** Application, Core, Logger, Network, UI, Utils.
- Application Project View:** Shows files Hello.swift and CoreViewController.swift, and tabs for Identity, Deployment Info, App Icons and Launch Images, Supported Intents, and Frameworks, Libraries, and Embedded Content.
- Frameworks, Libraries, and Embedded Content:** A table listing five frameworks:

Name	Filters	Embed
Core.framework	iOS	Embed & Sign
Logger.framework	iOS	Embed & Sign
Network.framework	iOS	Embed & Sign
UI.framework	iOS	Embed & Sign
Utils.framework	iOS	Embed & Sign

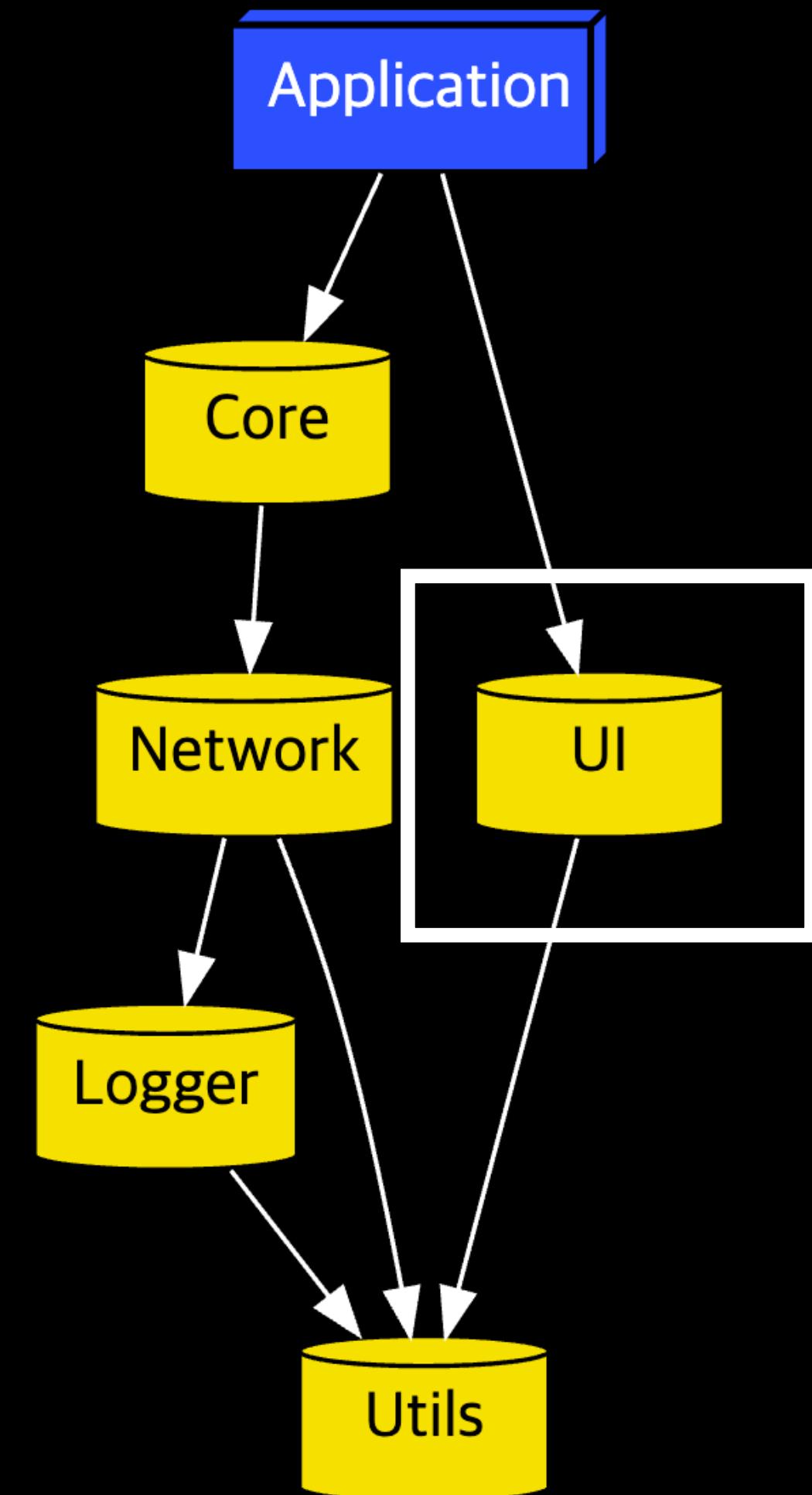


# 모듈 분리 작업

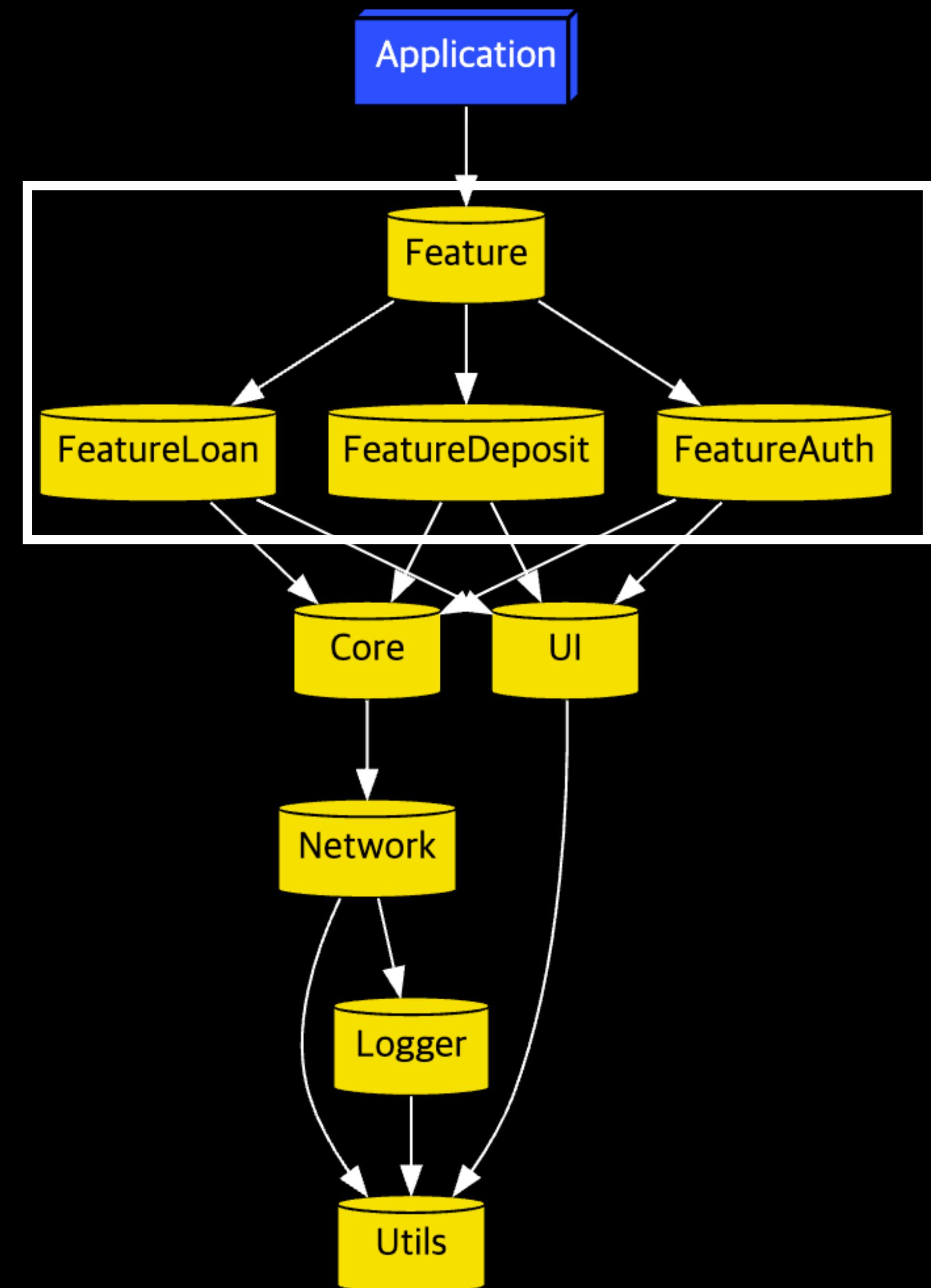
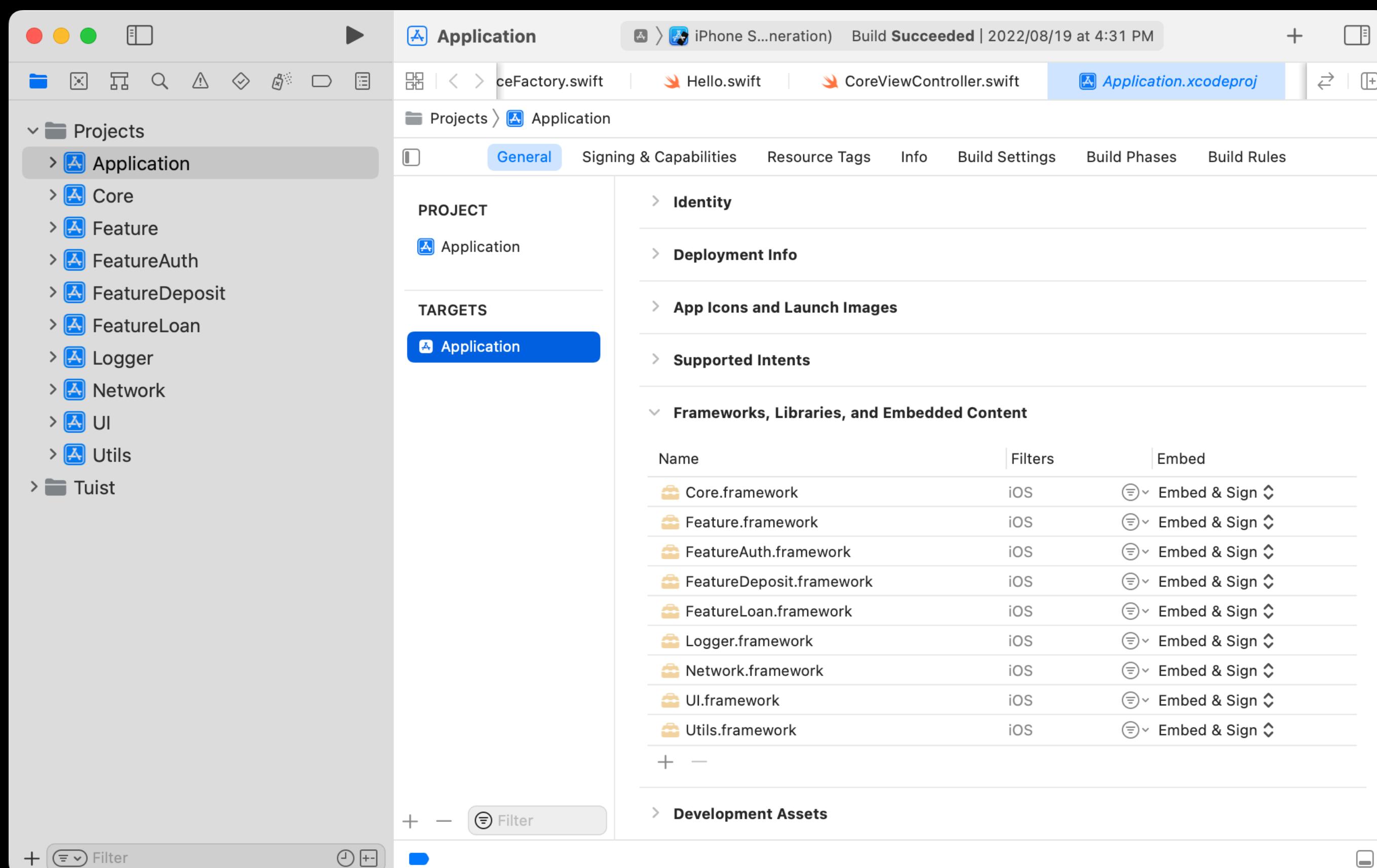
The screenshot shows the Xcode interface with the following details:

- Projects:** Application, Core, Logger, Network, UI, Utils.
- Application Project View:** Shows files Hello.swift and CoreViewController.swift, and tabs for Identity, Deployment Info, App Icons and Launch Images, Supported Intents, and Frameworks, Libraries, and Embedded Content.
- Frameworks, Libraries, and Embedded Content:** A table listing five frameworks:

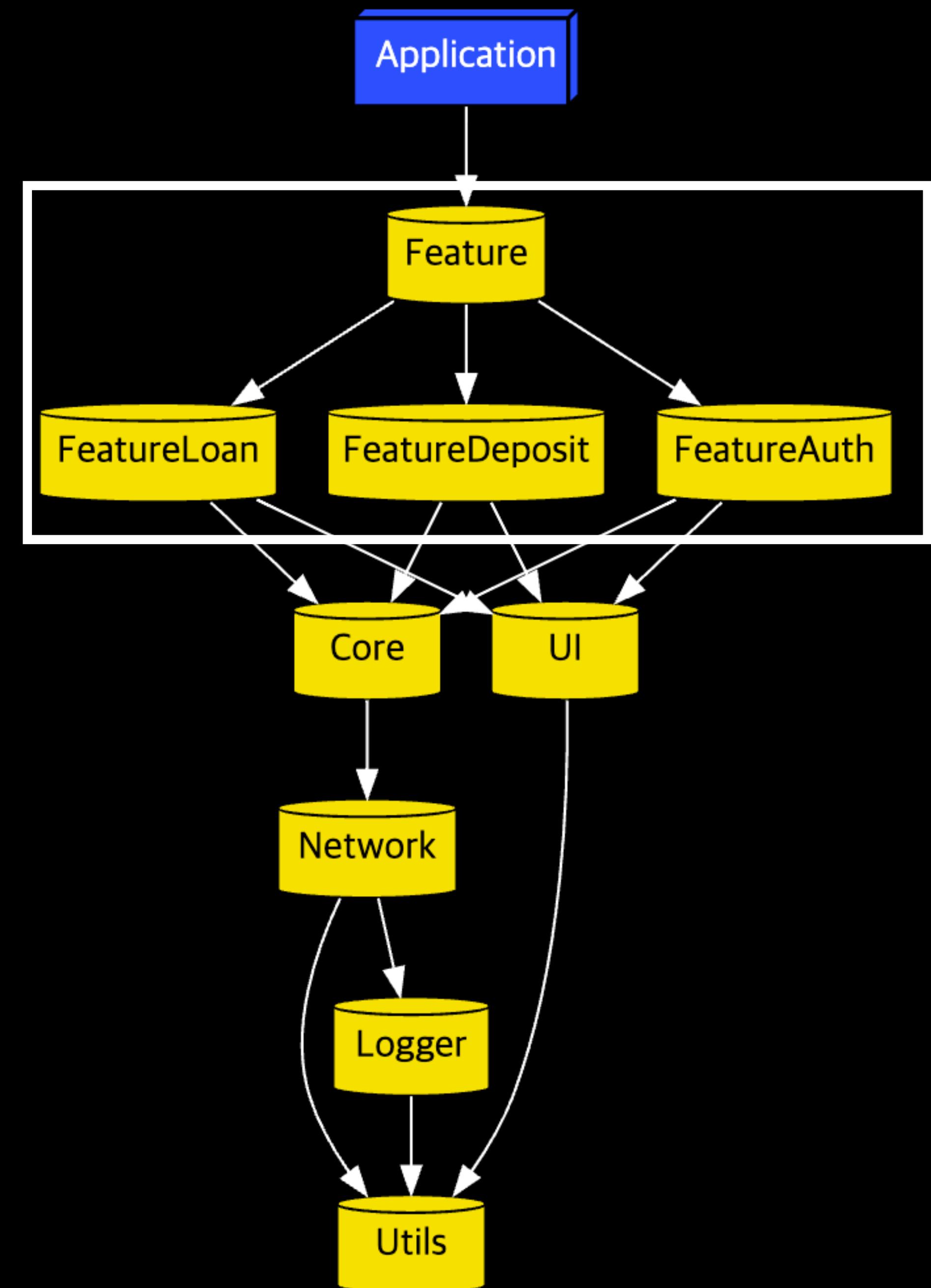
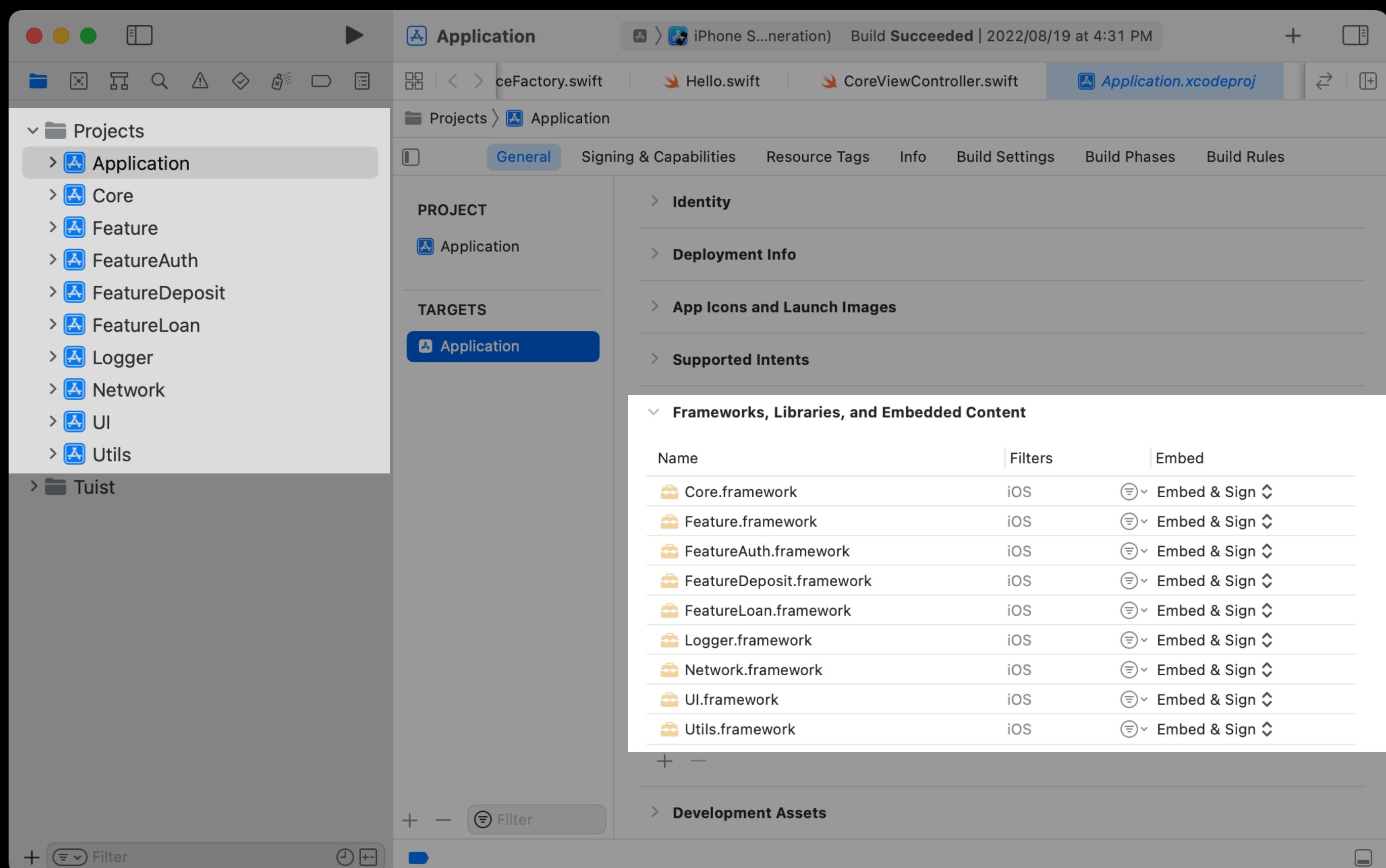
Name	Filters	Embed
Core.framework	iOS	Embed & Sign
Logger.framework	iOS	Embed & Sign
Network.framework	iOS	Embed & Sign
UI.framework	iOS	Embed & Sign
Utils.framework	iOS	Embed & Sign



# 모듈 분리 작업



# 모듈 분리 작업

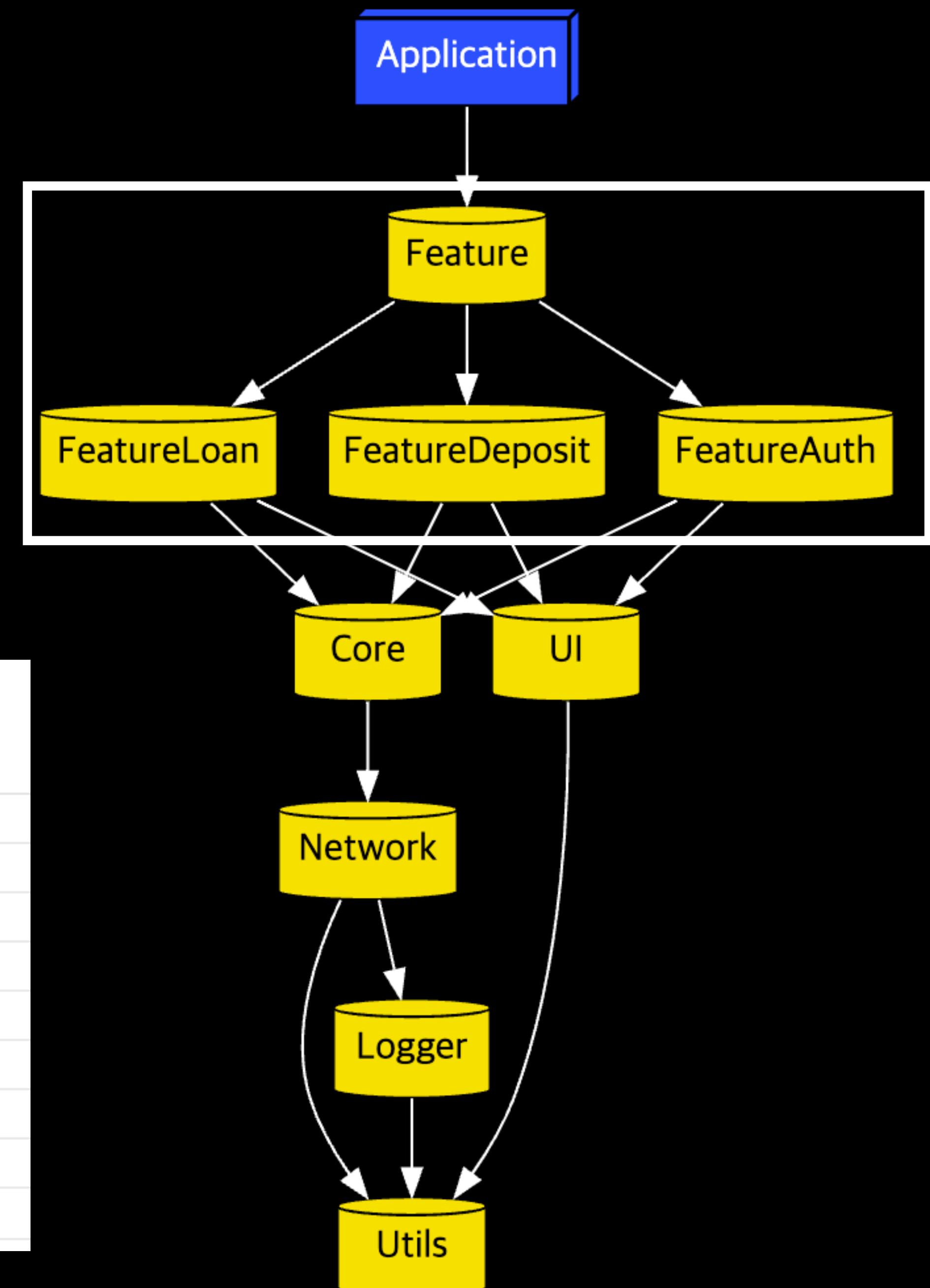


# 모듈 분리 작업

The screenshot shows the Xcode interface with the following details:

- Projects:** Application, Core, Feature, FeatureAuth, FeatureDeposit, FeatureLoan, Logger, Network, UI, Utils.
- Application Project Details:** iPhone S...eration, Build Succeeded | 2022/08/19 at 4:31 PM. Files: Hello.swift, CoreViewController.swift, Application.xcodeproj.
- Build Settings:** Identity, Deployment Info, App Icons and Launch Images, Supported Intents, Frameworks, Libraries, and Embedded Content.
- Frameworks, Libraries, and Embedded Content Table:**

Name	Filters	Embed
Core.framework	iOS	Embed & Sign
Feature.framework	iOS	Embed & Sign
FeatureAuth.framework	iOS	Embed & Sign
FeatureDeposit.framework	iOS	Embed & Sign
FeatureLoan.framework	iOS	Embed & Sign
Logger.framework	iOS	Embed & Sign
Network.framework	iOS	Embed & Sign
UI.framework	iOS	Embed & Sign
Utils.framework	iOS	Embed & Sign

# 모듈 분리 작업

The screenshot shows the Xcode interface with a project structure on the left and a build log on the right.

**Projects:**

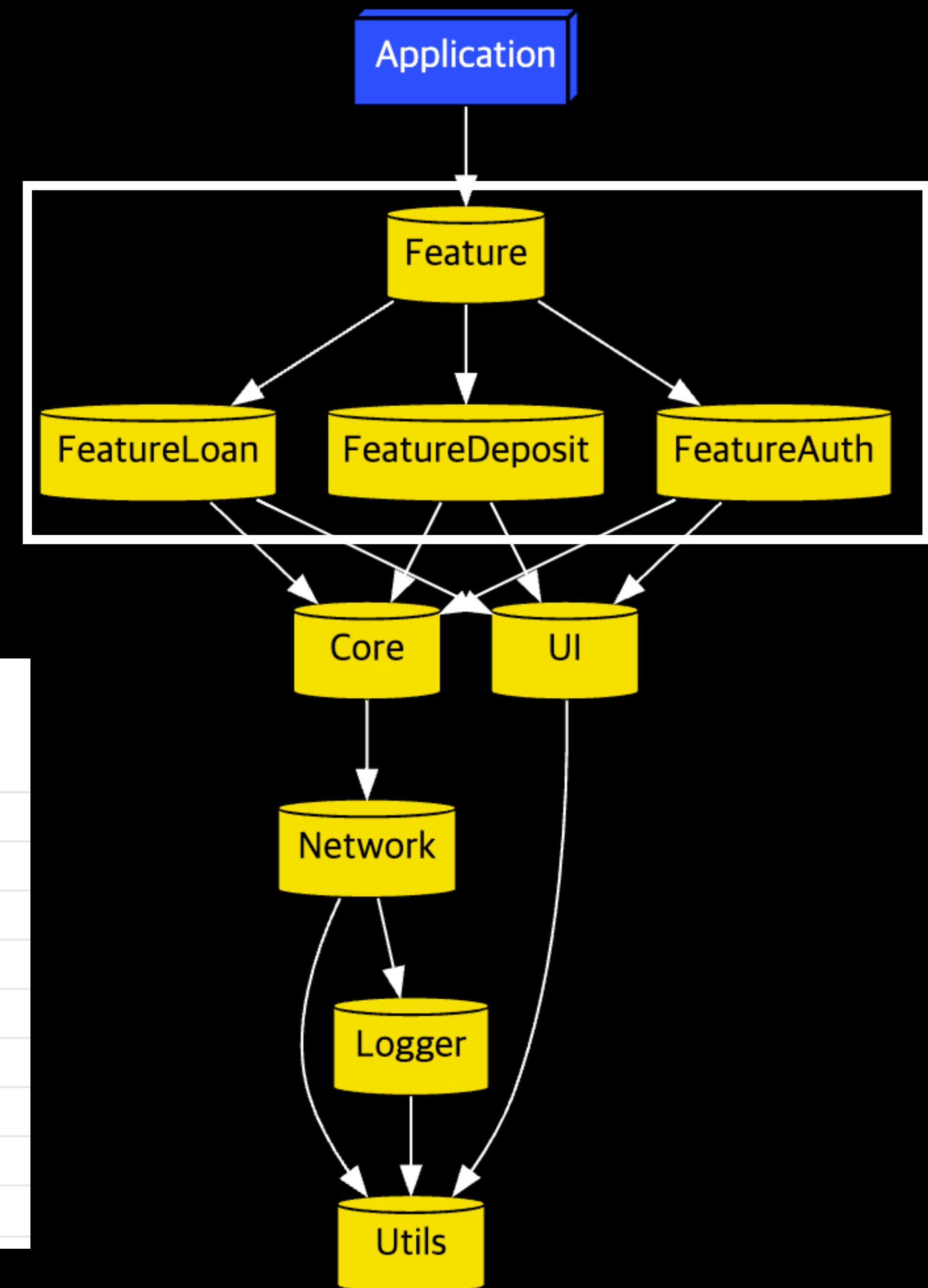
- > Application
- > Core
- > Feature
- > FeatureAuth
- > FeatureDeposit
- > FeatureLoan
- > Logger
- > Network
- > UI
- > Utils

**Frameworks, Libraries, and Embedded Content:**

Name	Filters	Embed
Core.framework	iOS	Embed & Sign
Feature.framework	iOS	Embed & Sign
FeatureAuth.framework	iOS	Embed & Sign
FeatureDeposit.framework	iOS	Embed & Sign
FeatureLoan.framework	iOS	Embed & Sign
Logger.framework	iOS	Embed & Sign
Network.framework	iOS	Embed & Sign
UI.framework	iOS	Embed & Sign
Utils.framework	iOS	Embed & Sign

**Build Log:**

```
iPhone S...eration) Build Succeeded | 2022/08/19 at 4:31 PM  
Hello.swift | CoreViewController.swift Application.xcodeproj  
ng & Capabilities Resource Tags Info Build Settings Build Phases Build Rules  
> Identity  
> Deployment Info  
> App Icons and Launch Images  
> Supported Intents  
Frameworks, Libraries, and Embedded Content
```

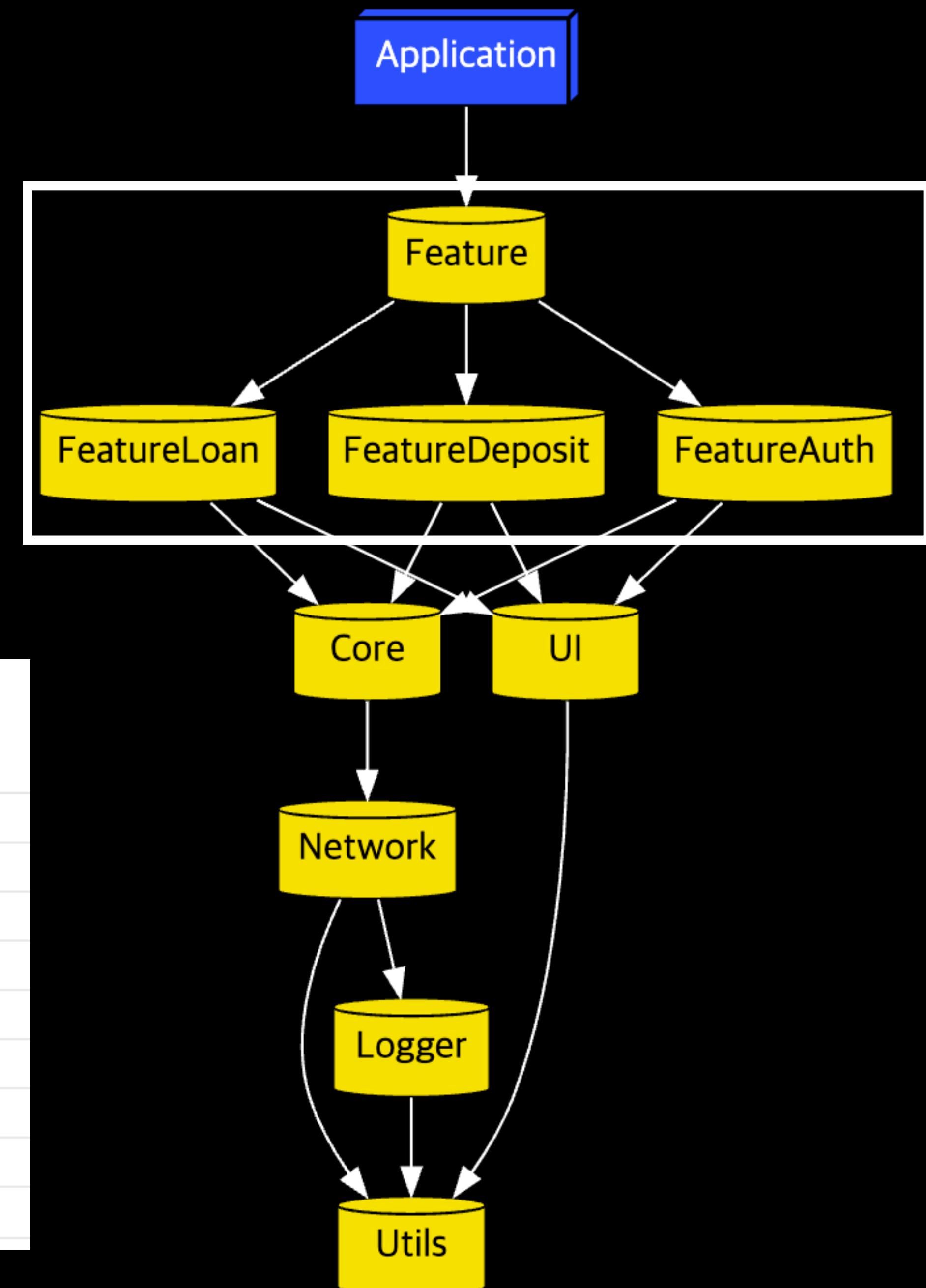


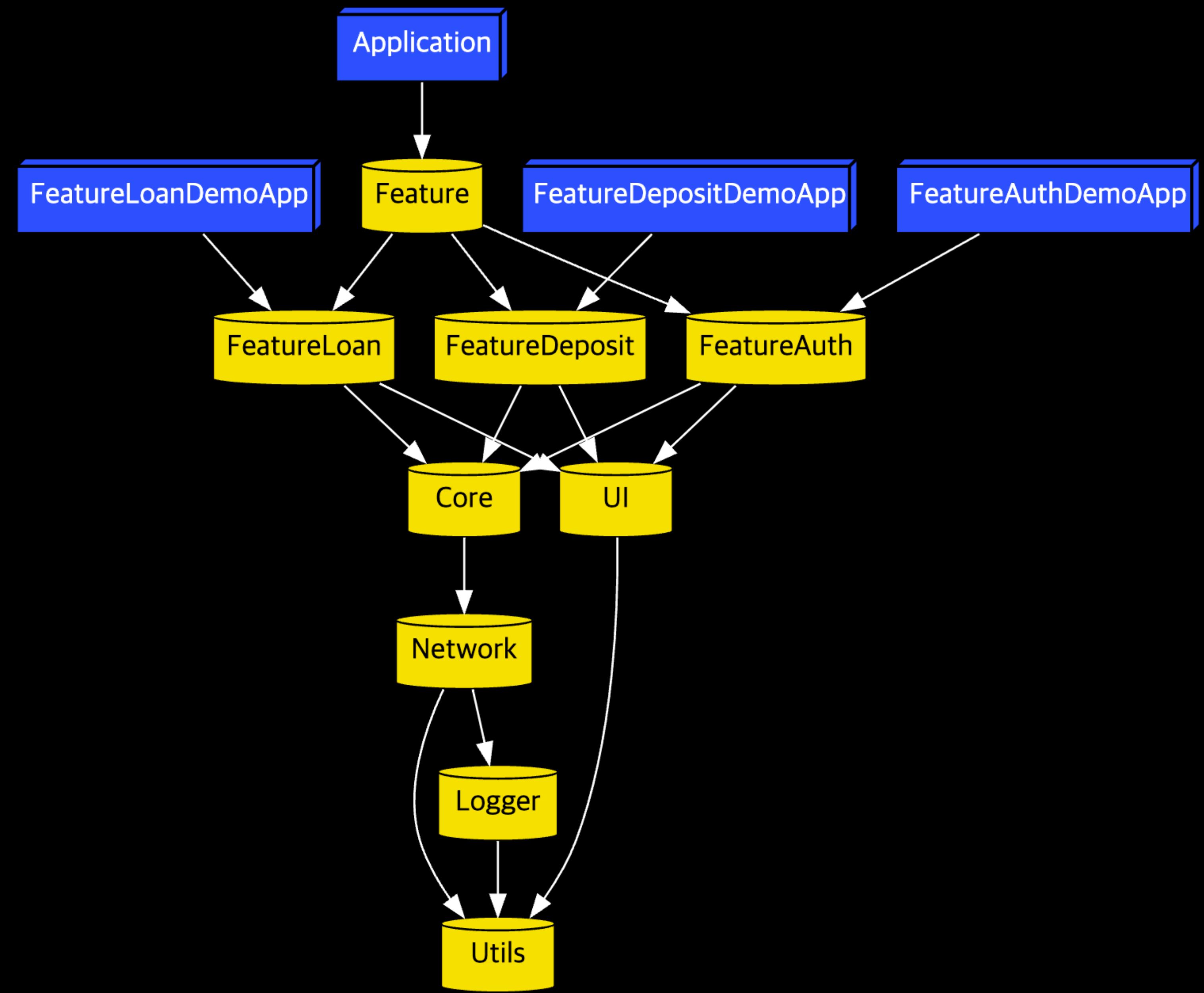
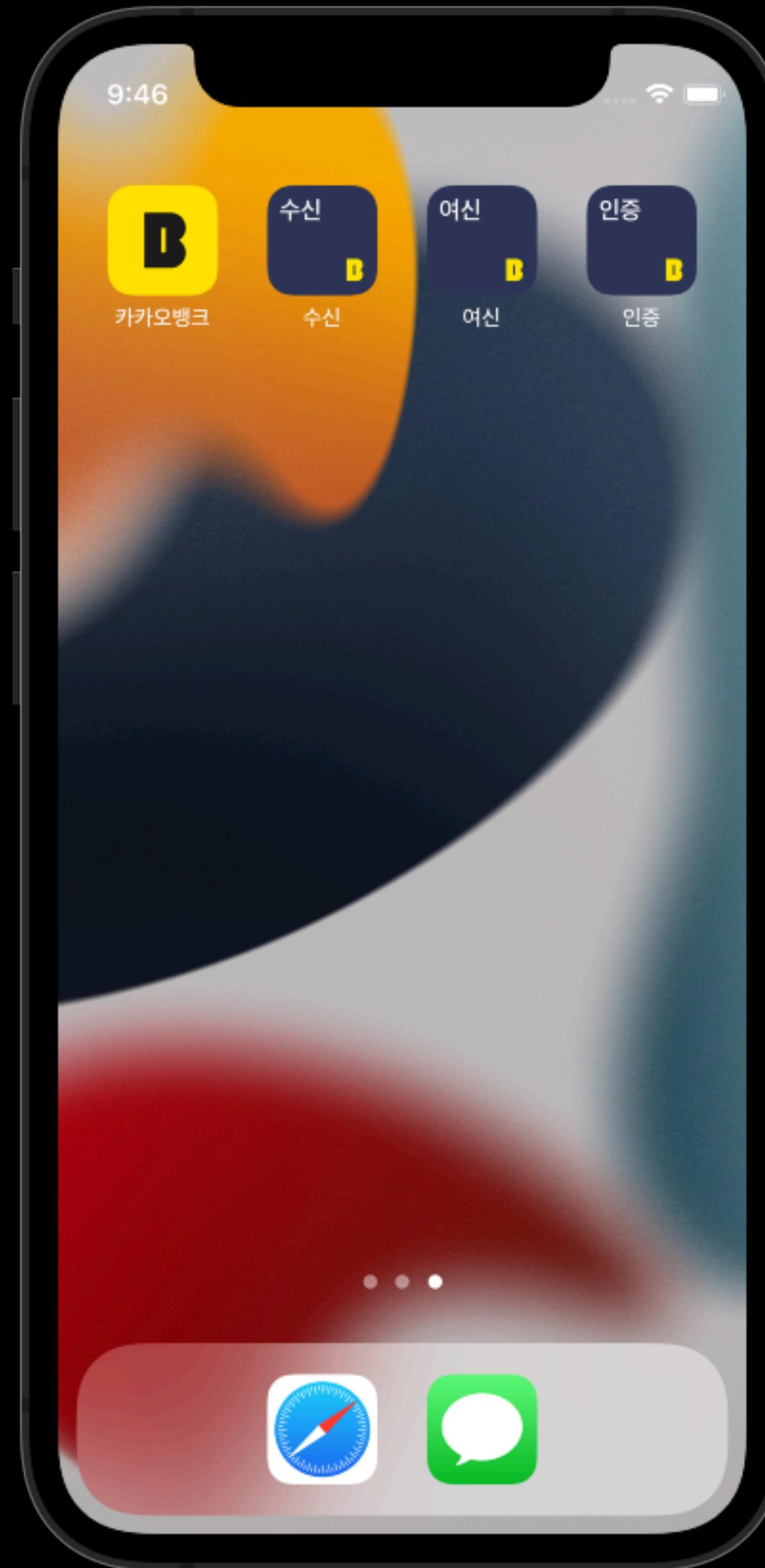
# 모듈 분리 작업

The screenshot shows the Xcode interface with the following details:

- Projects:** Application, Core, Feature, FeatureAuth, FeatureDeposit, FeatureLoan, Logger, Network, UI, Utils.
- Application Project:** iPhone S...eration, Build Succeeded | 2022/08/19 at 4:31 PM. The file list includes Hello.swift and CoreViewController.swift.
- Build Settings:** Identity, Deployment Info, App Icons and Launch Images, Supported Intents, Frameworks, Libraries, and Embedded Content.
- Frameworks, Libraries, and Embedded Content:** A table listing frameworks with their filters and embed options.

Name	Filters	Embed
Core.framework	iOS	Embed & Sign
Feature.framework	iOS	Embed & Sign
FeatureAuth.framework	iOS	Embed & Sign
FeatureDeposit.framework	iOS	Embed & Sign
FeatureLoan.framework	iOS	Embed & Sign
Logger.framework	iOS	Embed & Sign
Network.framework	iOS	Embed & Sign
UI.framework	iOS	Embed & Sign
Utils.framework	iOS	Embed & Sign





프로젝트 여정

개발 생산성 개선 방법 알아보기

카카오뱅크 모듈 아키텍처 설계

# Core

Core

Common

CodeTable

Error

Environment

Security

E2E

FDS

Keychain

Logger

Trace

AppLog

Firebase

Utility

Currency

Foundation  
Extension

Network

APIs

APIBase

# Core

Core

Common

CodeTable

Error

Environment

Security

E2E

FDS

Keychain

Logger

Trace

AppLog

Firebase

Utility

Currency

Foundation  
Extension

Network

APIs

APIBase

# UI

UI

Keyboard

Toast

Color

RichString

UIComponent

Helper

Resource

Indicator

# Feature

## Features

수신

입출금통장

정기예금

정기적금

여신

주택담보대출

전월세보증금대출

신용대출

카드

체크카드

미니카드

제휴

해외송금

해외계좌송금

WU빠른해외송금

고객센터

인증

홈

이체

내 신용정보

Core

UI

# Feature

## Features

수신

입출금통장

정기예금

정기적금

여신

주택담보대출

전월세보증금대출

신용대출

카드

체크카드

미니카드

제휴

해외송금

해외계좌송금

WU빠른해외송금

고객센터

인증

홈

이체

내 신용정보

Core

UI

# 카카오뱅크 모듈 아키텍처 설계



# 카카오뱅크 모듈 아키텍처 설계



# 카카오뱅크 모듈 아키텍처 설계



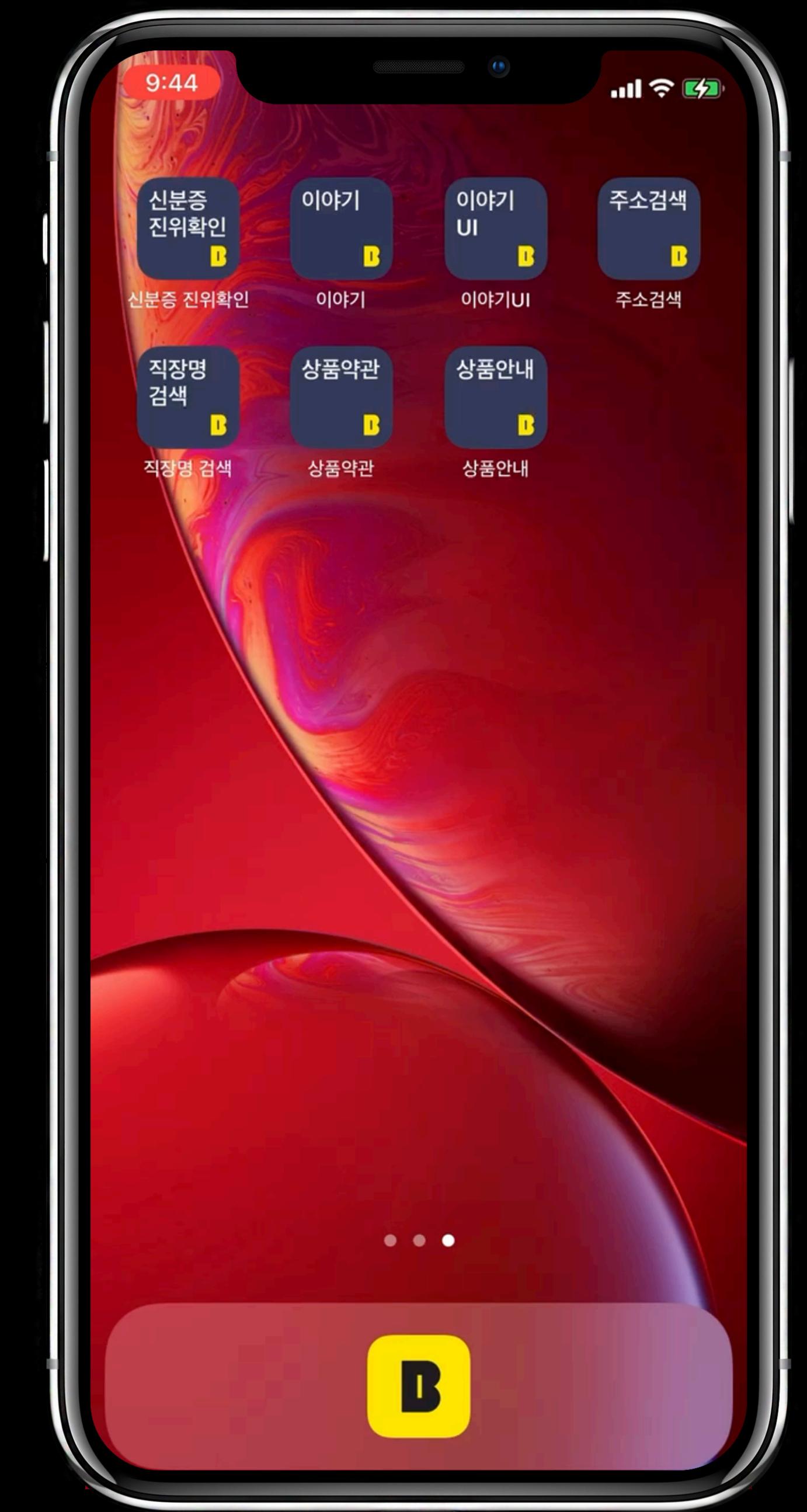
# 카카오뱅크 모듈 아키텍처 설계

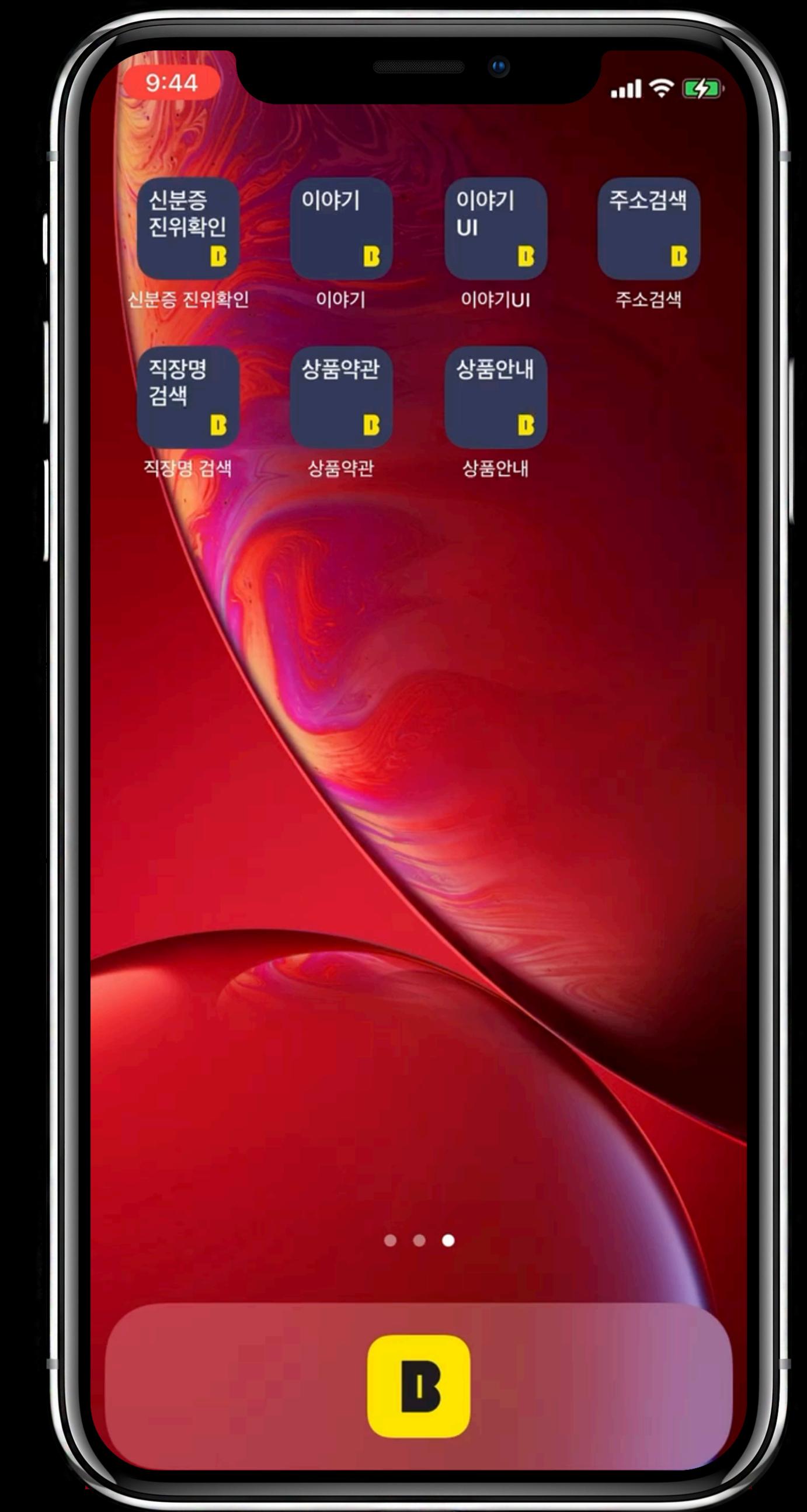


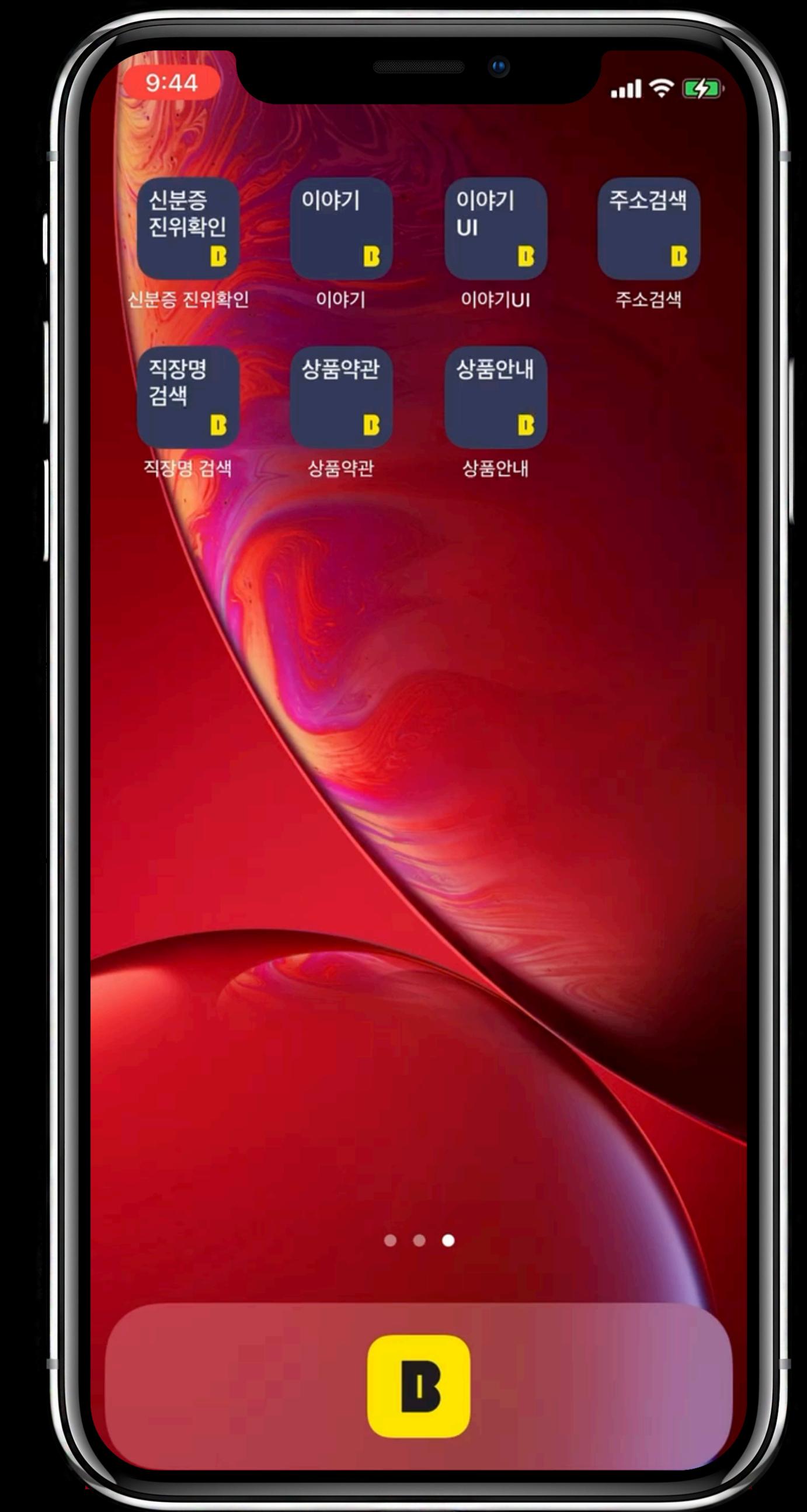
# 카카오뱅크 모듈 아키텍처 설계











모듈 아키텍처 설계로 얻은 이점

# 모듈 아키텍처 설계로 얻은 이점

- 전체 의존관계 파악 가능
  - 코드의 의존관계 파악 및 해당 성격에 맞는 모듈로 이전하여 응집도 높임

# 모듈 아키텍처 설계로 얻은 이점

- 전체적인 의존관계 파악 가능
- 증분 빌드 활용

# 모듈 아키텍처 설계로 얻은 이점

- 전체적인 의존관계 파악 가능
- 증분 빌드 활용
  - Tuist로 생성된 모듈의 데모앱 활용
    - InjectionIII, Lookin, RevealApp, Xcode Preview 등 UI 개발 도구를 활용한 UI 개발
    - 빠른 비즈니스 로직 개발, 실행 및 확인 가능



# 모듈 아키텍처 설계로 얻은 이점

- 전체적인 의존관계 파악 가능
- 증분 빌드 활용
  - Tuist로 생성된 모듈의 데모앱 활용
    - InjectionIII, Lookin, RevealApp, Xcode Preview 등 UI 개발 도구를 활용한 UI 개발
    - 빠른 비즈니스 로직 개발, 실행 및 확인 가능



# 모듈 아키텍처 설계로 얻은 이점

- 전체적인 의존관계 파악 가능
- 증분 빌드 활용
  - Tuist로 생성된 모듈의 데모앱 활용
  - 빨라진 테스트 사이클

# 모듈 아키텍처 설계로 얻은 이점

- 전체적인 의존관계 파악 가능
- 빠른 증분 빌드 활용
- 코드에서 모듈로 관점 변경
  - 안정적인 모듈 구축
  - 모듈 사용시 예상 가능한 동작
  - 새로운 상품과 기능을 빠르고 쉽게 작성
  - 프로젝트가 안정적인 구조로 개선

# 모듈 아키텍처 설계로 얻은 이점

- 전체적인 의존관계 파악 가능
- 빠른 증분 빌드 활용
- 코드에서 모듈로 관점 변경
  - 안정적인 모듈 구축
  - 모듈 사용시 예상 가능한 동작
  - 새로운 상품과 기능을 빠르고 쉽게 작성
  - 프로젝트가 안정적인 구조로 개선

앞으로 해야 할 것

# 앞으로 해야 할 것

- 아키텍처 설계 문서
- 모듈의 세분화
- 모듈의 문서화
- 데모 애플리케이션의 지속적인 관리

# 앞으로 해야 할 것

- 아키텍처 설계 문서
- 모듈의 세분화
- 모듈의 문서화
- 데모 애플리케이션의 지속적인 관리

# 앞으로 해야 할 것

- 아키텍처 설계 문서
- 모듈의 세분화
- 모듈의 문서화
- 데모 애플리케이션의 지속적인 관리

# 앞으로 해야 할 것

- 아키텍처 설계 문서
- 모듈의 세분화
- 모듈의 문서화
- 데모 애플리케이션의 지속적인 관리

# 앞으로 해야 할 것

- 아키텍처 설계 문서
- 모듈의 세분화
- 모듈의 문서화
- 데모 애플리케이션의 지속적인 관리

# 앞으로 해야 할 것

- 아키텍처 설계 문서
- 모듈의 세분화
- 모듈의 문서화
- 데모 애플리케이션의 지속적인 관리

# 앞으로 해야 할 것

- 아키텍처 설계 문서
- 모듈의 세분화
- 모듈의 문서화
- 데모 애플리케이션의 지속적인 관리

# 앞으로 해야 할 것

- 아키텍처 설계 문서
- 모듈의 세분화
- 모듈의 문서화
- 데모 애플리케이션의 지속적인 관리

마치며

마치며

마치며