

The Sandbox

Container Directory



Giftbot
이봉원

Sandbox

Bundle Container

Data Container

Tips

Summary



Sandbox





App Sandboxing



App Sandboxing

Sandboxing your app is a great way to protect systems and users by limiting the privileges of an app to its intended functionality, increasing the difficulty for malicious software to compromise your users' systems

App Sandboxing



App Sandboxing

Sandboxing your app is a **great way to protect systems** and users by limiting the privileges of an app to its intended functionality, increasing the difficulty for malicious software to compromise your users' systems

App Sandboxing



App Sandboxing

Sandboxing your app is a great way to protect systems and users by limiting the privileges of an app to its intended functionality, increasing the difficulty for malicious software to compromise your users' systems

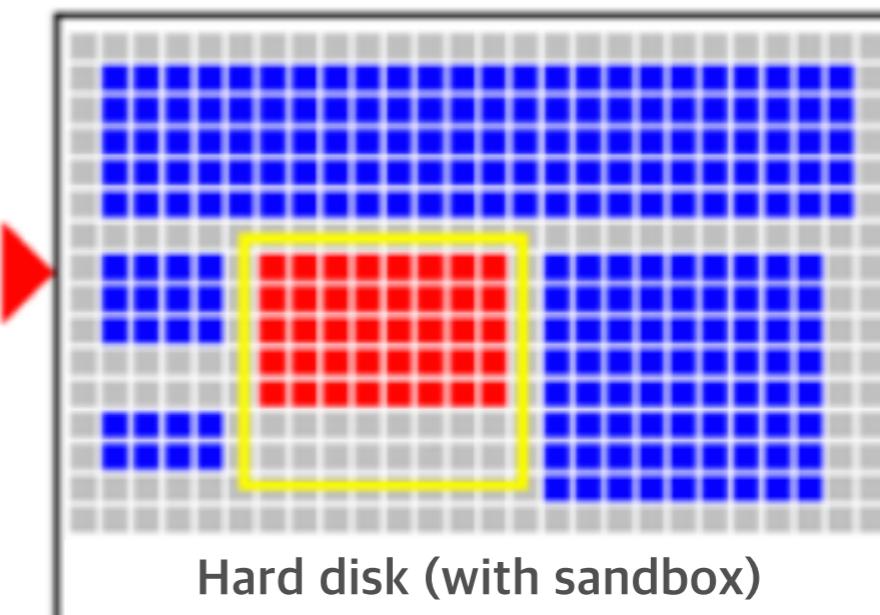
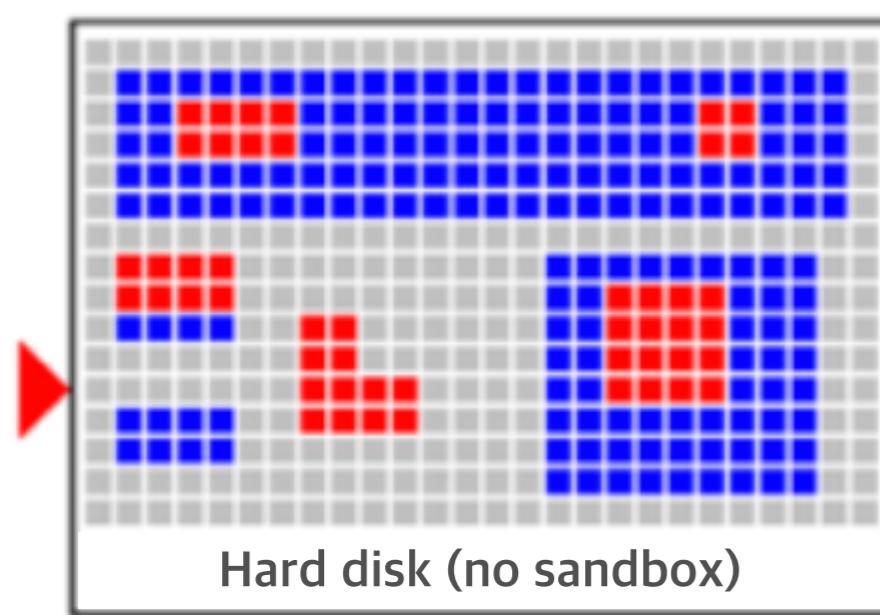
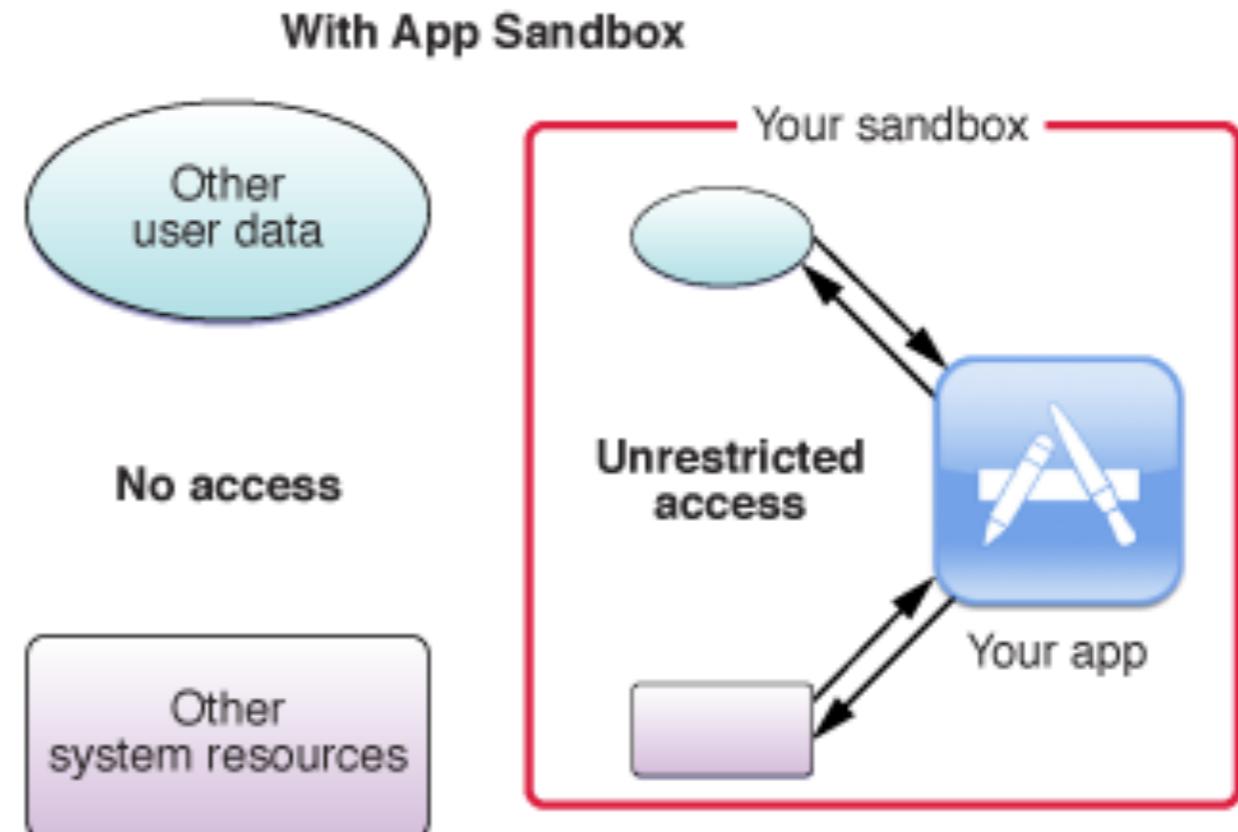
App Sandboxing



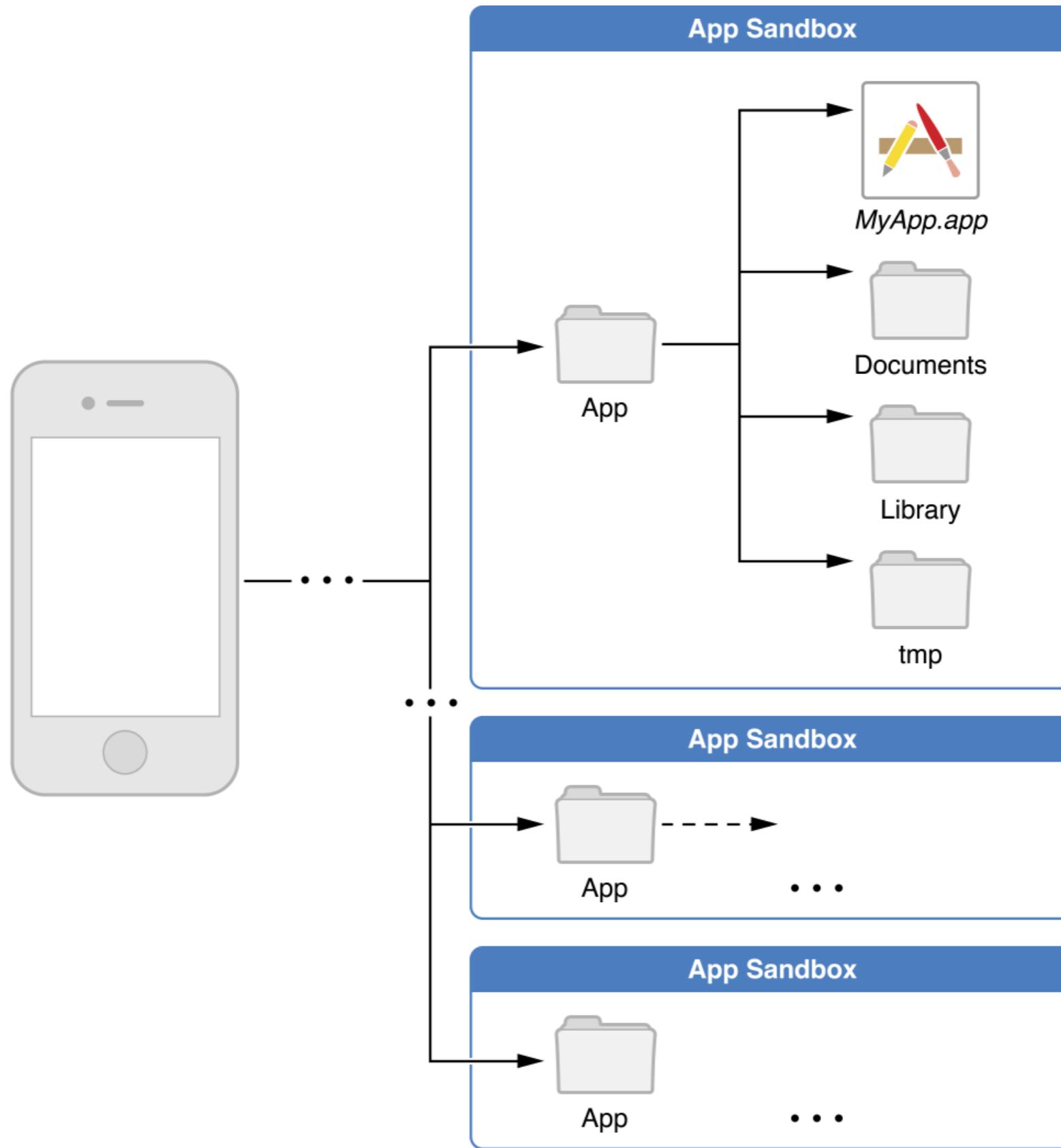
App Sandboxing

Sandboxing your app is a great way to protect systems and users by limiting the privileges of an app to its intended functionality, increasing the difficulty for malicious software to compromise your users' systems

Without vs With Sandbox



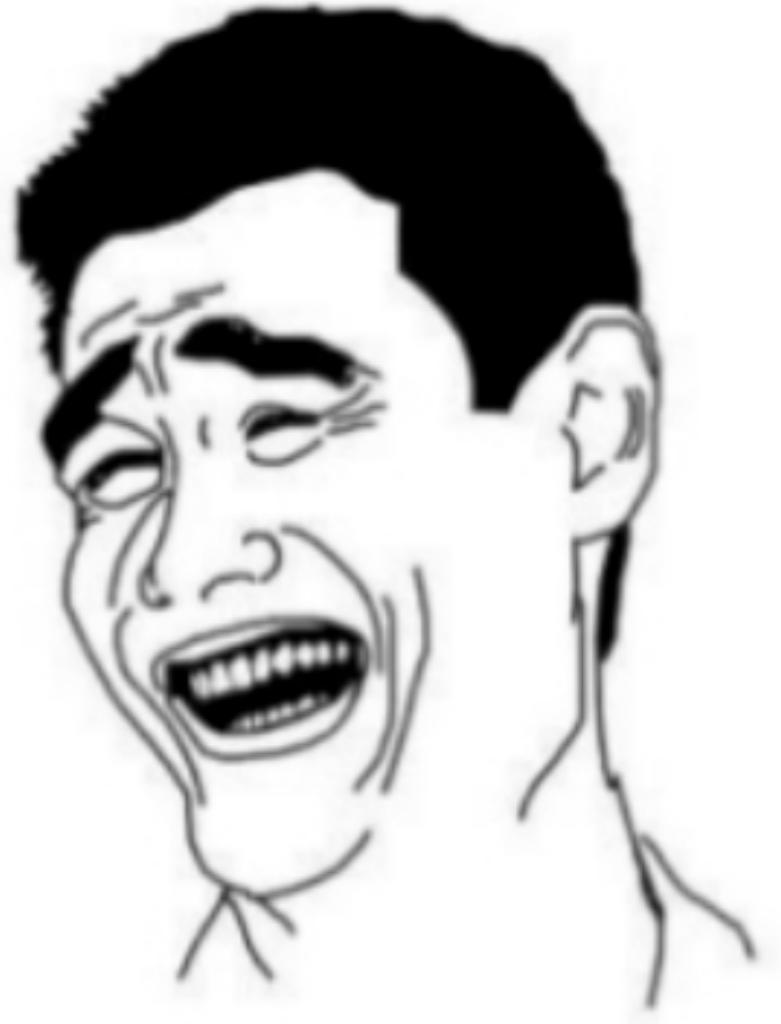
iOS App Sandboxes



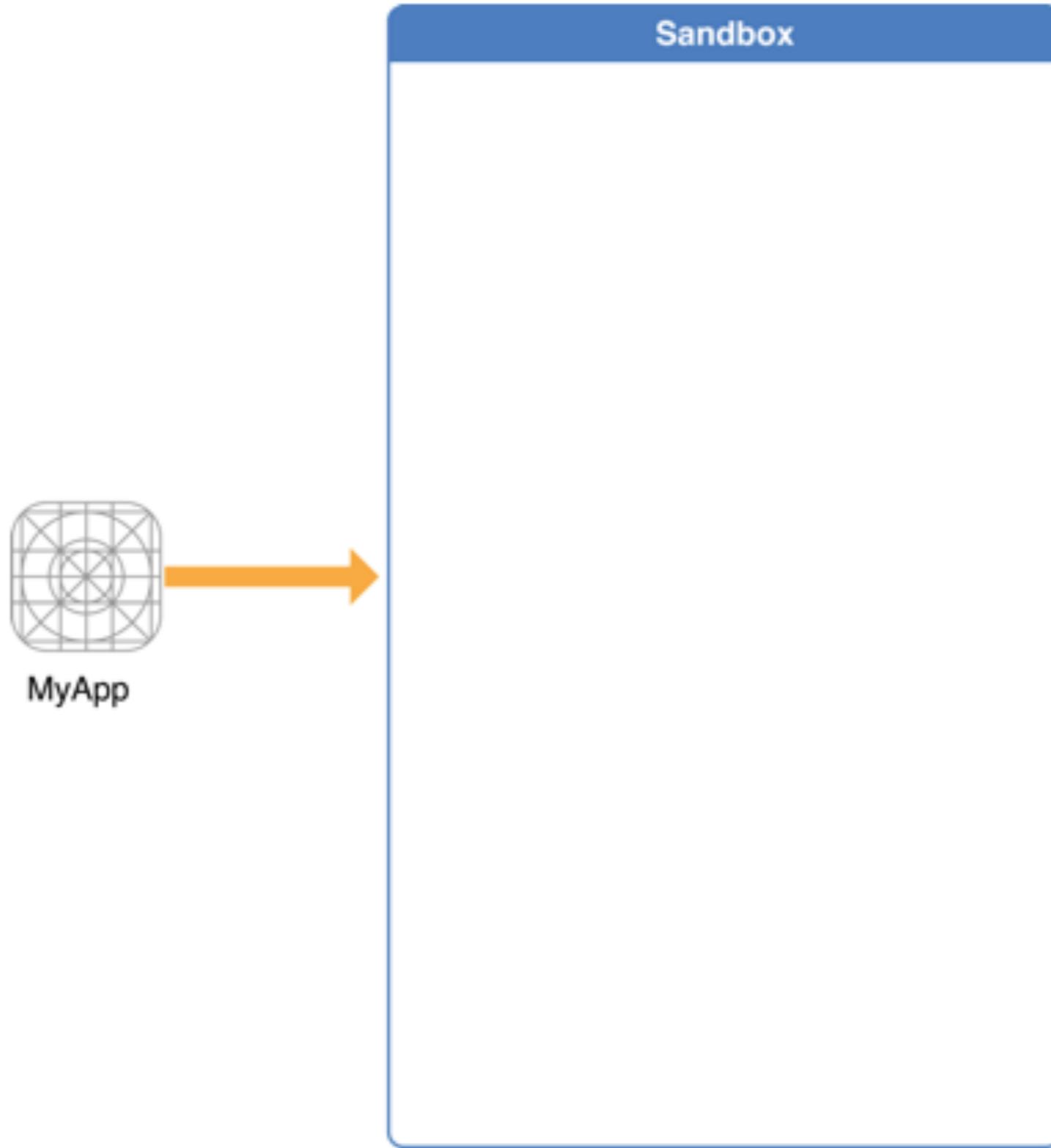


With Great Power Comes Great Responsibility

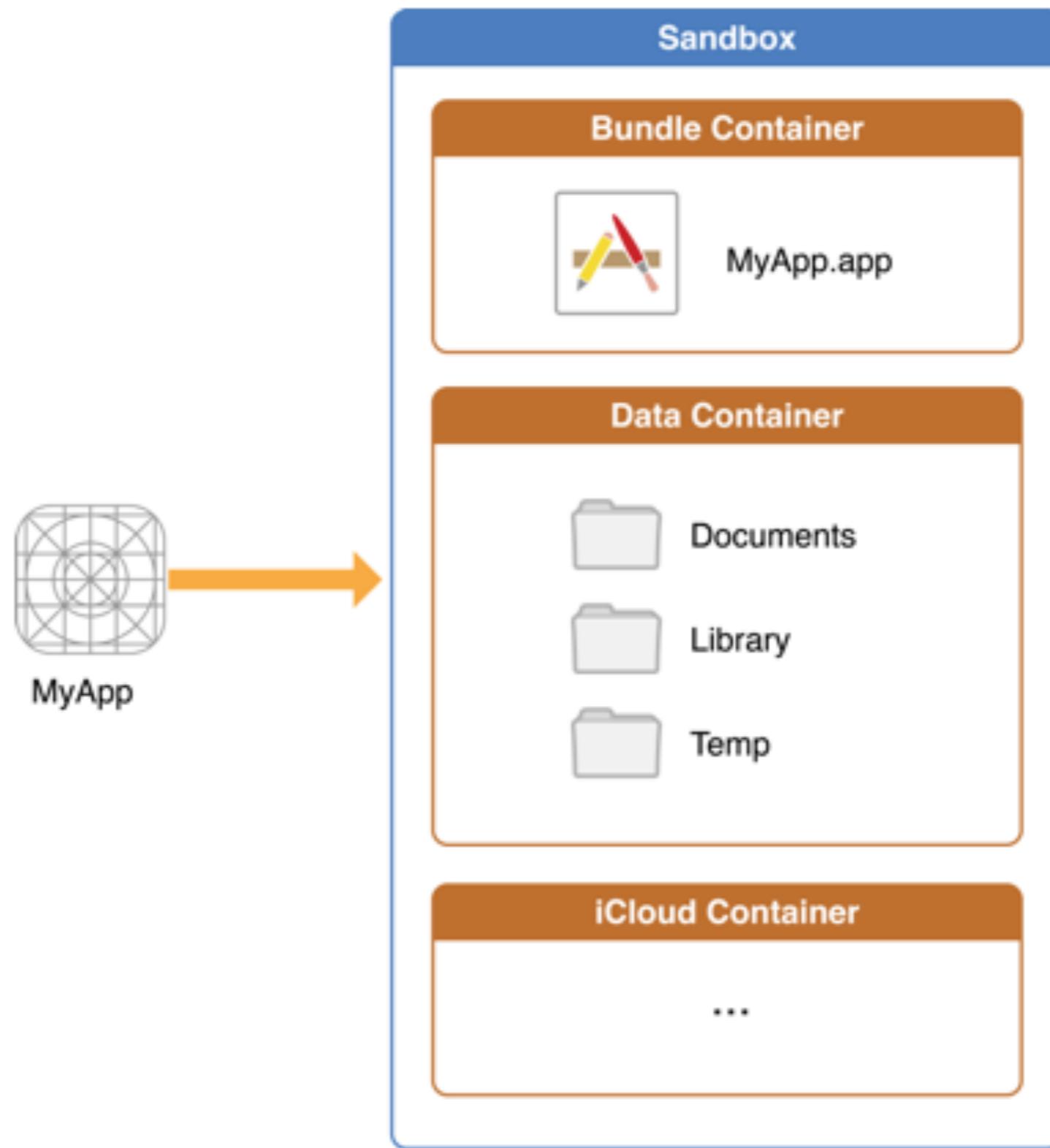
I've no great powers.
So there are no responsibilities



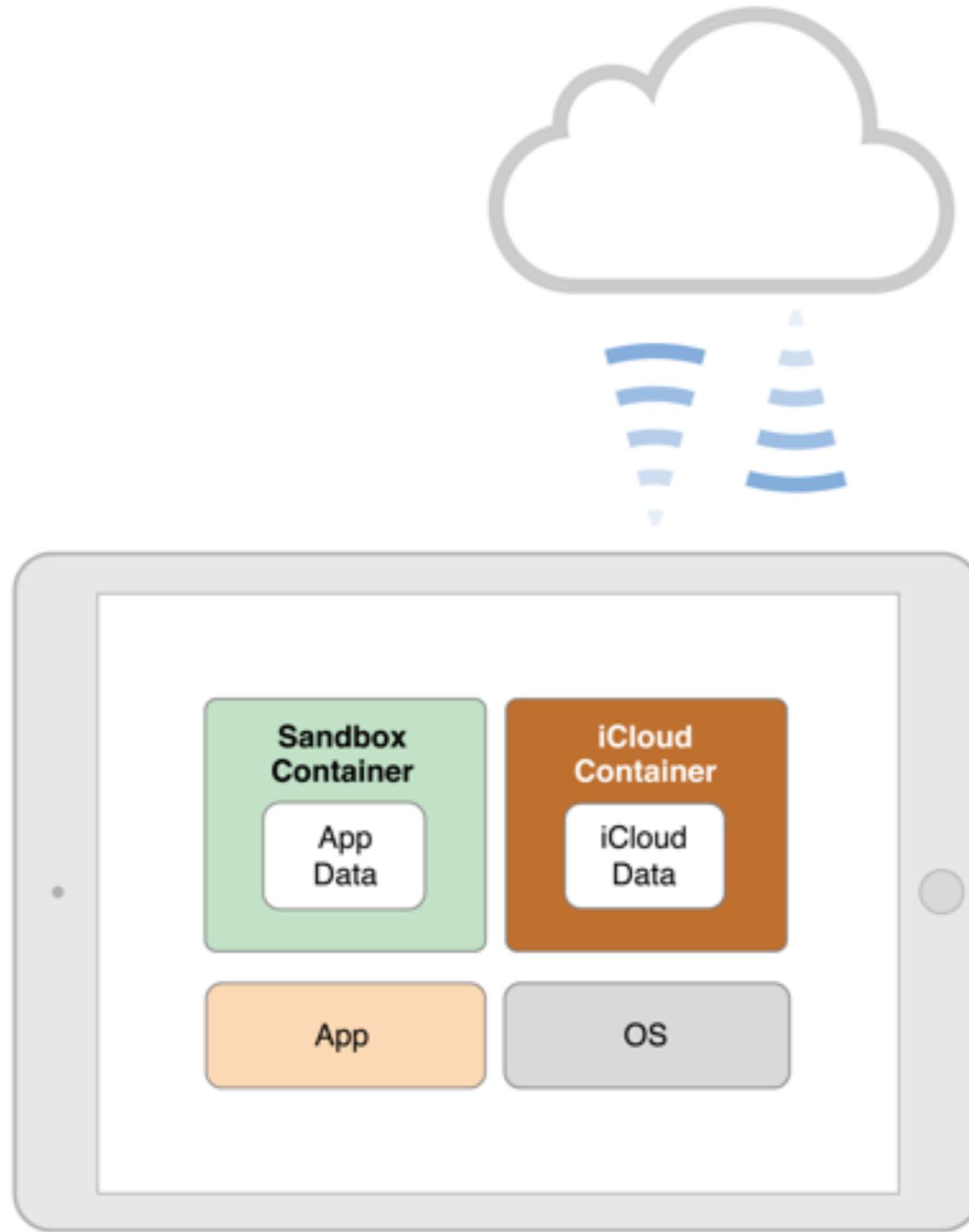
Sandbox Containers



Sandbox Containers

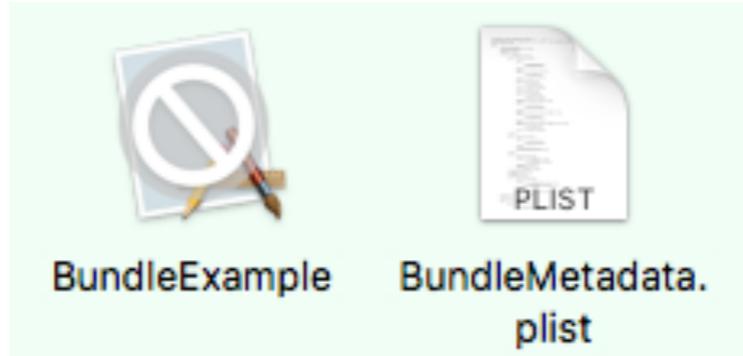


Sandbox Containers



Bundle

Bundle



- ✓ **Read.**
- ✗ **Write, Delete, Backup**

Bundle.main.bundlePath

파일 시스템 내 하나의 디렉토리

실행 가능(Executable) 파일, Info.plist, 각종 Resources (이미지, 사운드, strings 등) 등을 함께 그룹화

Codesign Tool이 Signing 을 위해 Bundle Container 를 불러와 Code Seal 적용

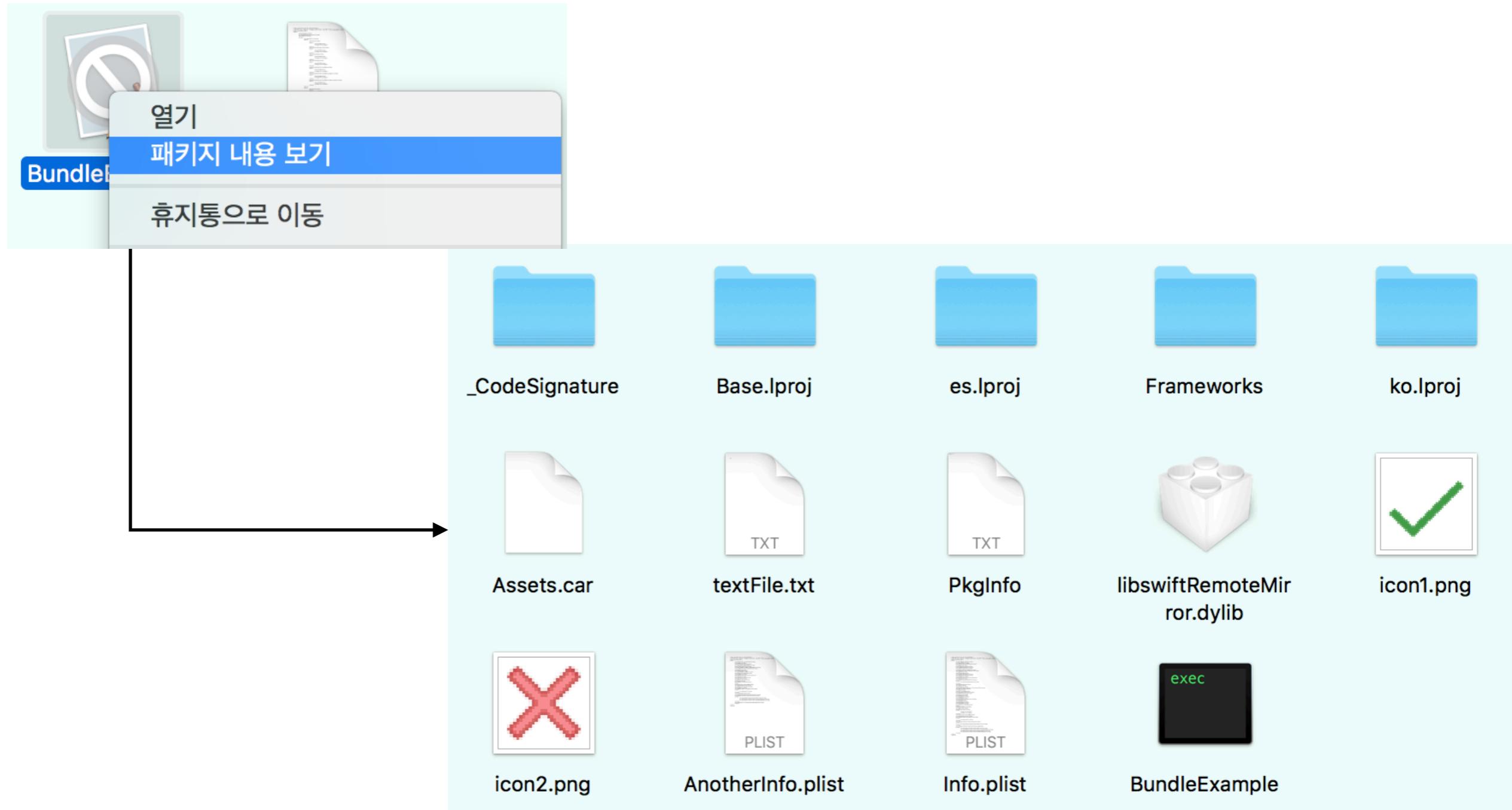
Static Library - Executable 파일에 통합 컴파일

Dynamic Library 및 Framework 는 Frameworks 디렉토리에 존재

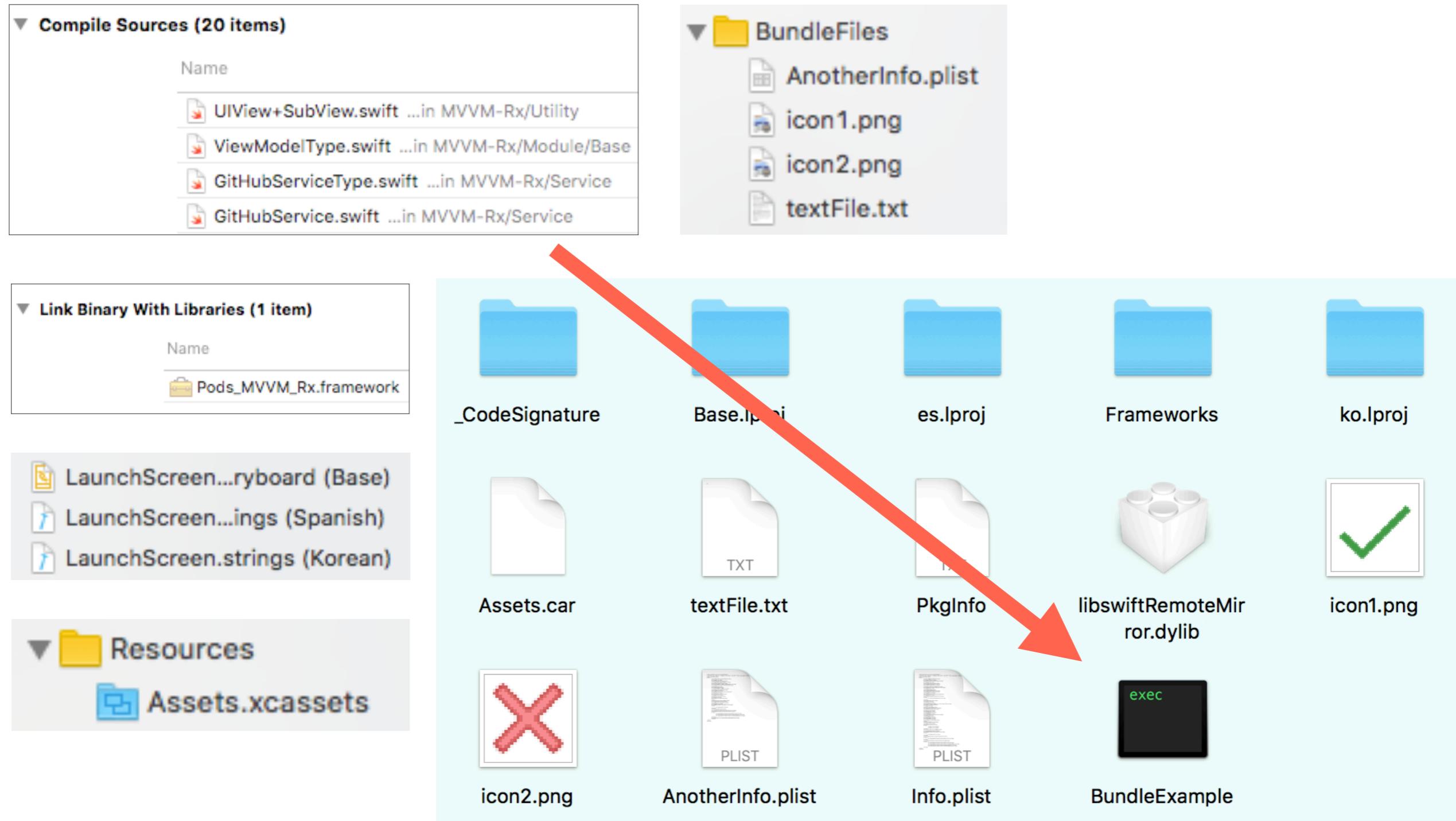
읽기 전용. 수정이 필요한 경우 데이터 컨테이너로 옮겨서 작업

iTunes, iCloud 에 백업되지 않음

Bundle Package



Bundle Package



Bundle Package

▼ Compile Sources (20 items)

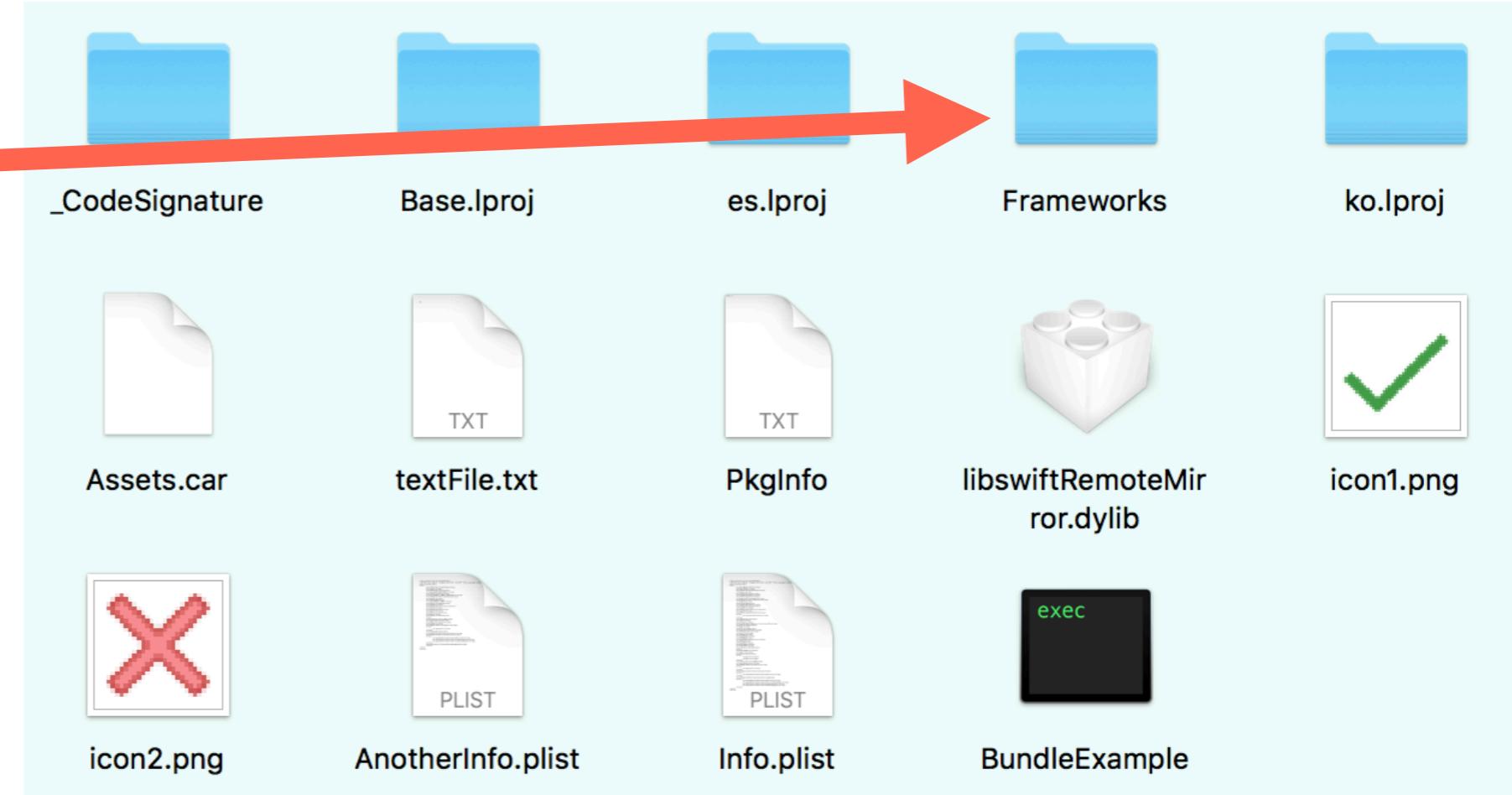
Name
UIView+SubView.swift ...in MVVM-Rx/Utility
ViewModelType.swift ...in MVVM-Rx/Module/Base
GitHubServiceType.swift ...in MVVM-Rx/Service
GitHubService.swift ...in MVVM-Rx/Service

▼ BundleFiles

- AnotherInfo.plist
- icon1.png
- icon2.png
- textFile.txt

▼ Link Binary With Libraries (1 item)

Name
Pods_MVVM_Rx.framework



Bundle Package

▼ Compile Sources (20 items)

Name
UIView+SubView.swift ...in MVVM-Rx/Utility
ViewModelType.swift ...in MVVM-Rx/Module/Base
GitHubServiceType.swift ...in MVVM-Rx/Service
GitHubService.swift ...in MVVM-Rx/Service

▼ BundleFiles

AnotherInfo.plist
icon1.png
icon2.png
textFile.txt

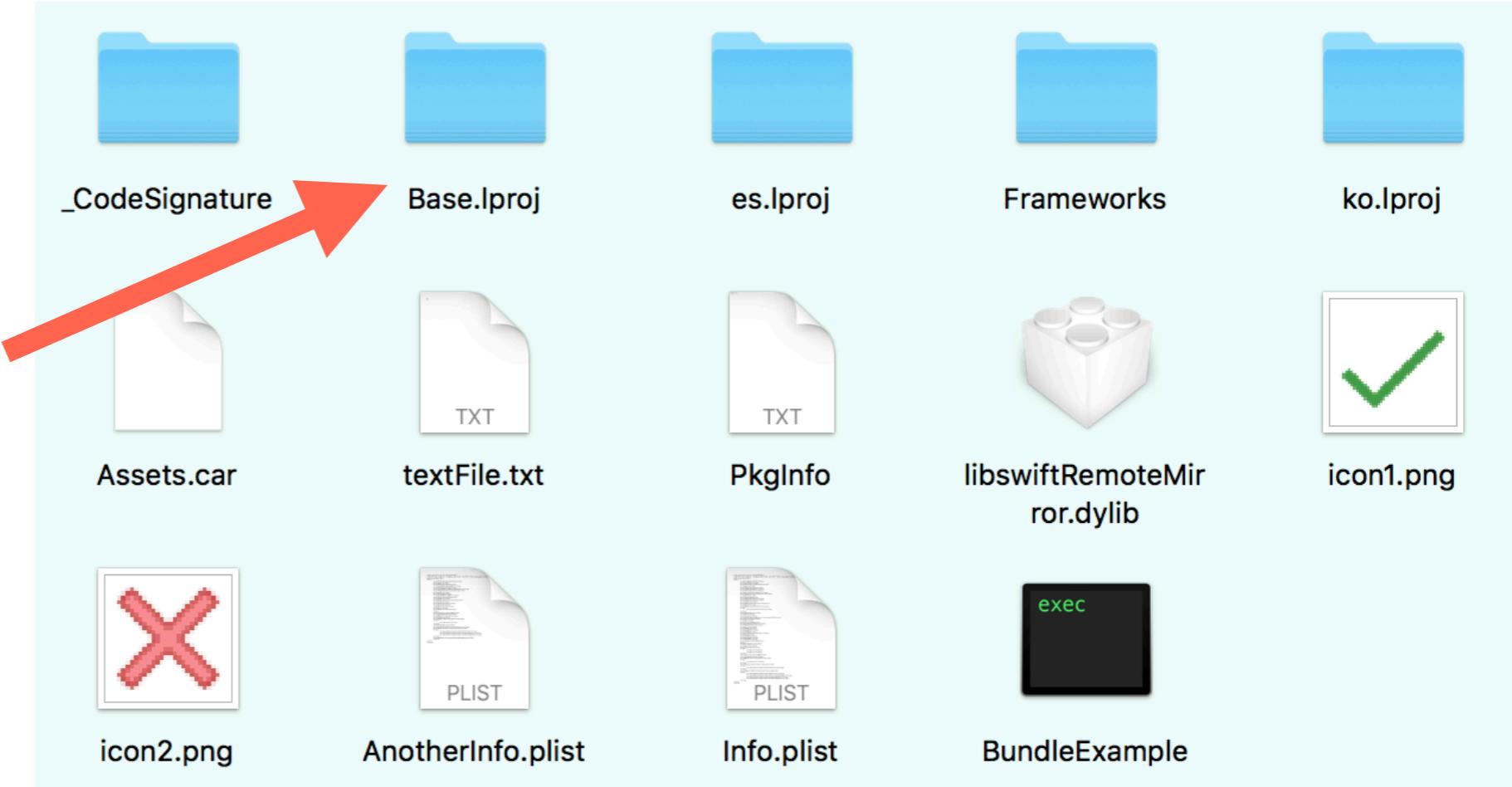
▼ Link Binary With Libraries (1 item)

Name
Pods_MVVM_Rx.framework

LaunchScreen...ryboard (Base)
LaunchScreen...ings (Spanish)
LaunchScreen.strings (Korean)

▼ Resources

Assets.xcassets



Bundle Package

▼ Compile Sources (20 items)

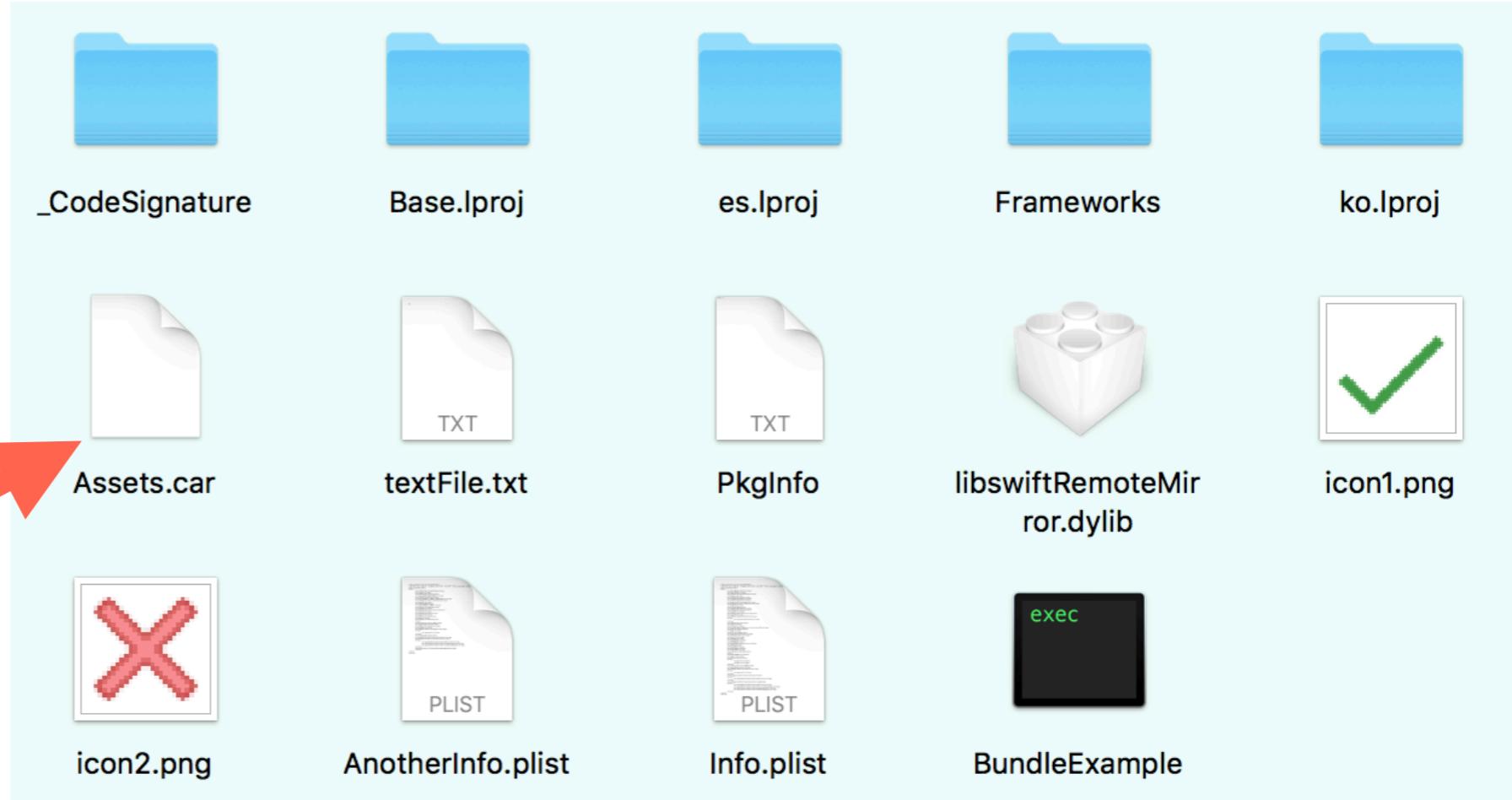
Name
UIView+SubView.swift ...in MVVM-Rx/Utility
ViewModelType.swift ...in MVVM-Rx/Module/Base
GitHubServiceType.swift ...in MVVM-Rx/Service
GitHubService.swift ...in MVVM-Rx/Service

▼ BundleFiles

AnotherInfo.plist
icon1.png
icon2.png
textFile.txt

▼ Link Binary With Libraries (1 item)

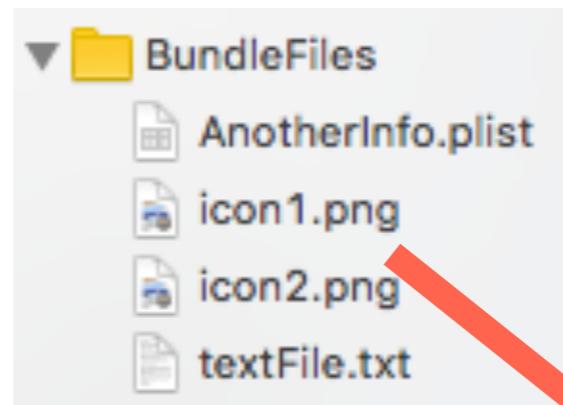
Name
Pods_MVVM_Rx.framework



Bundle Package

▼ Compile Sources (20 items)

Name
UIView+SubView.swift ...in MVVM-Rx/Utility
ViewModelType.swift ...in MVVM-Rx/Module/Base
GitHubServiceType.swift ...in MVVM-Rx/Service
GitHubService.swift ...in MVVM-Rx/Service



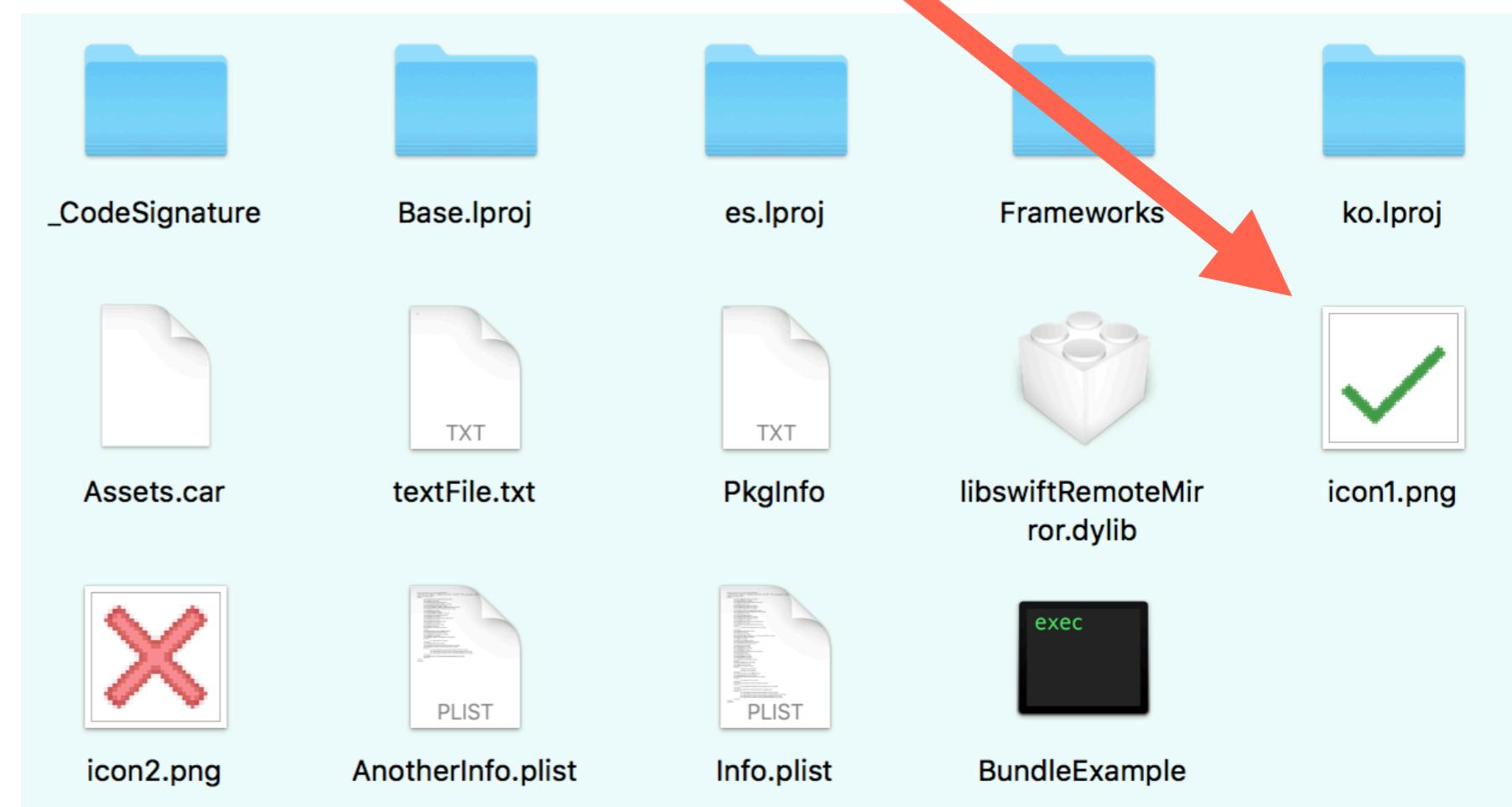
▼ Link Binary With Libraries (1 item)

Name
Pods_MVVM_Rx.framework

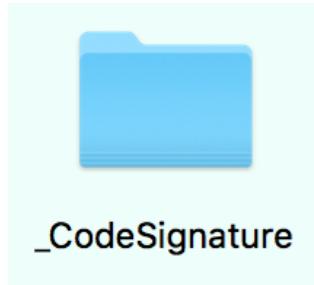
LaunchScreen...ryboard (Base)
LaunchScreen...ings (Spanish)
LaunchScreen.strings (Korean)

▼ Resources

Assets.xcassets



CodeSignature

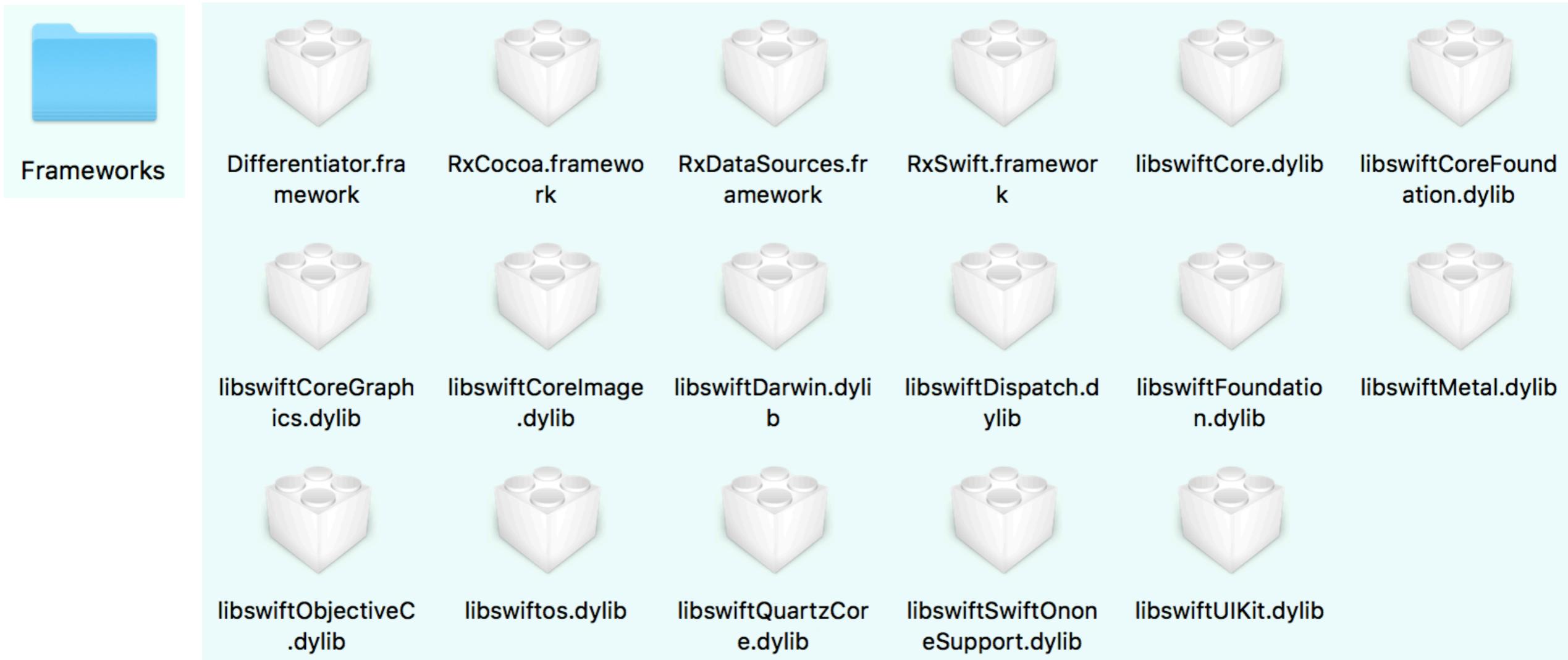


```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
    <key>files</key>
    <dict>
        <key>Assets.car</key>
        <data>8HYvZyfm72T5LbgWuqbr79kzzvI=</data>
        <key>Base.lproj/LaunchScreen.storyboardc/01J-lp-oVM-view-Ze5-6b-2t3.nib</key>
        <data>1F7h7Q0RFM6KAbx3cvJGNLcC1/c=</data>
        <key>Base.lproj/LaunchScreen.storyboardc/Info.plist</key>
        <data>n2t8gsDpfE6XkhG31p7IQJRxTxU=</data>
        <key>Base.lproj/LaunchScreen.storyboardc/UIViewController-01J-lp-oVM.nib</key>
        <data>9PxYe00hOhSoc9Th8ufTin0fbx8=</data>
        <key>Frameworks/Differentiator.framework/Differentiator</key>
        <data>jcX9Sv50DczNLErELHNNue4PhVQ=</data>
    </dict>
</dict>
</plist>
```

```
<key>files2</key>
<dict>
    <key>Assets.car</key>
    <dict>
        <key>hash</key>
        <data>8HYvZyfm72T5LbgWuqbr79kzzvI=</data>
        <key>hash2</key>
        <data>QUiK86mImN+eJcDOVlWqm1mUh+tU+cdFtu//S/rb2Dg=</data>
    </dict>
</dict>
```

Frameworks

- .a** - Static Library (실행 파일에 포함)
- .dylib** - Dynamic Library
- .framework** - (Static / Dynamic) Framework



CocoaPods supports Static Frameworks

Latest release

1.4.0

segiddins released this on 19 Jan · 138 commits to master since this release

Assets

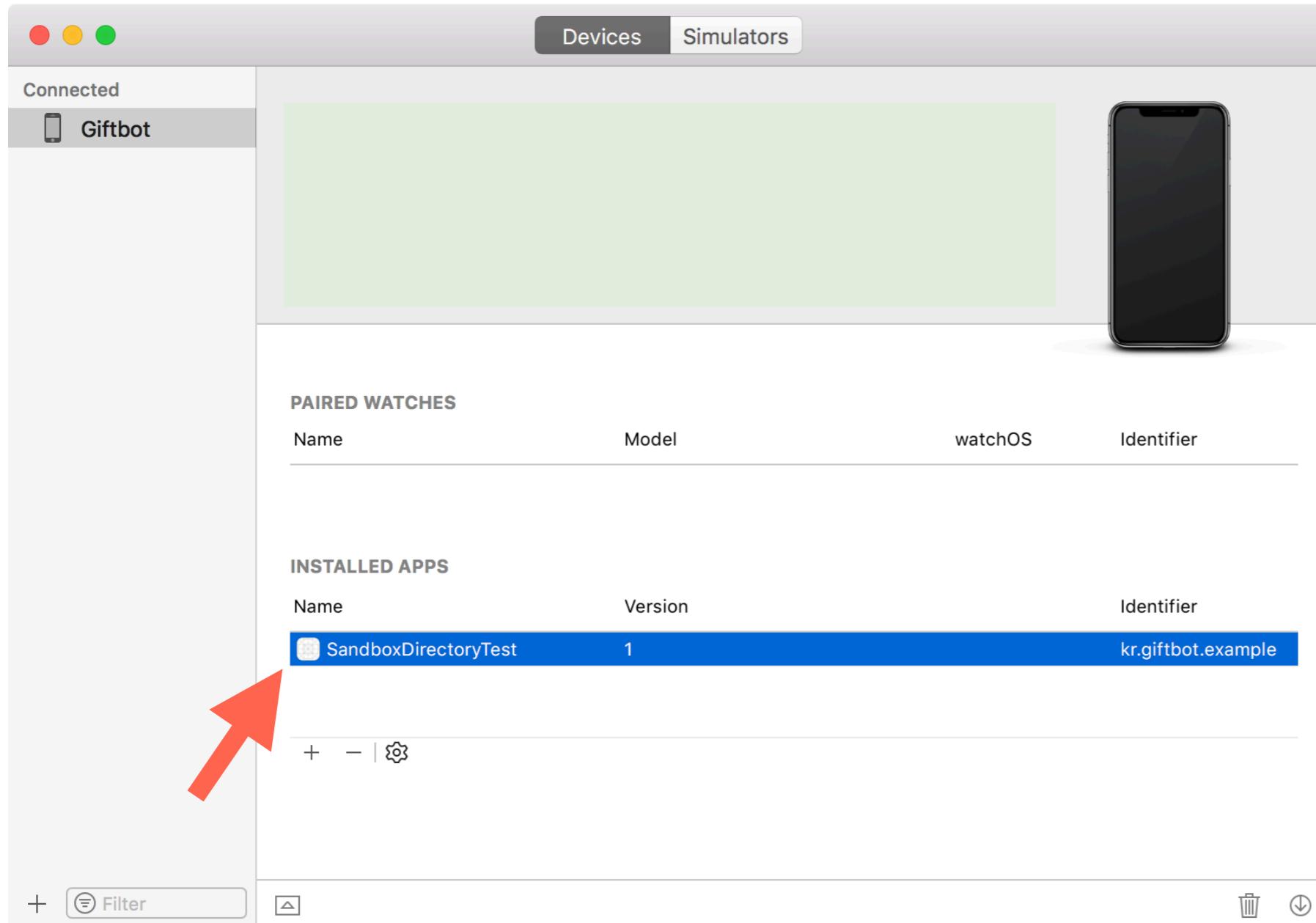
Static Frameworks

Source static frameworks is another powerful new feature of CocoaPods 1.4.0. For the first time, it is possible to distribute sources and build them into a static framework when `use_frameworks!` is specified in the `Podfile`. Because dynamic libraries cannot depend upon static libraries, it now becomes possible to distribute pods that depend upon other static library frameworks, including the many static frameworks that are released as `vendored_frameworks` today.

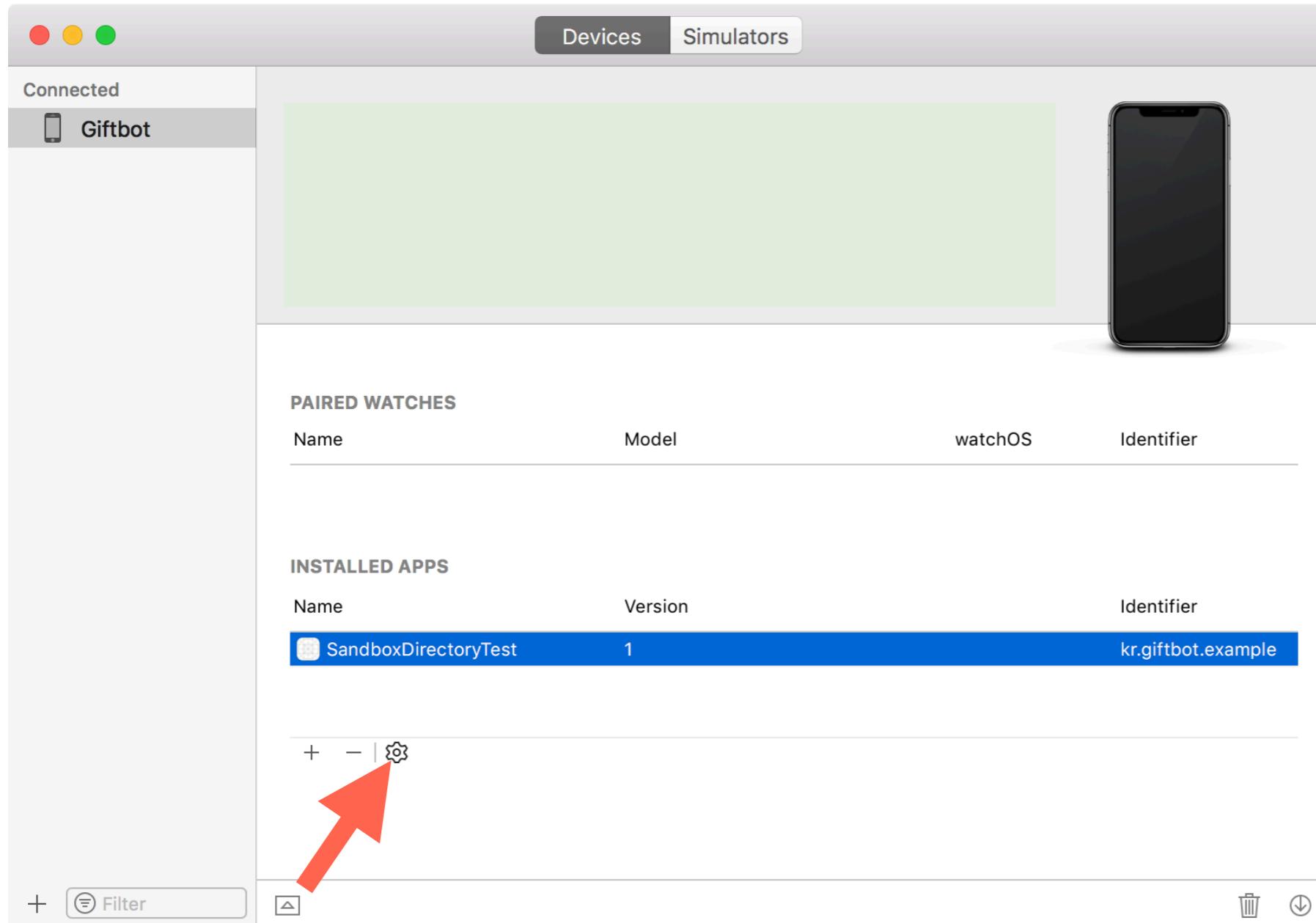


Data

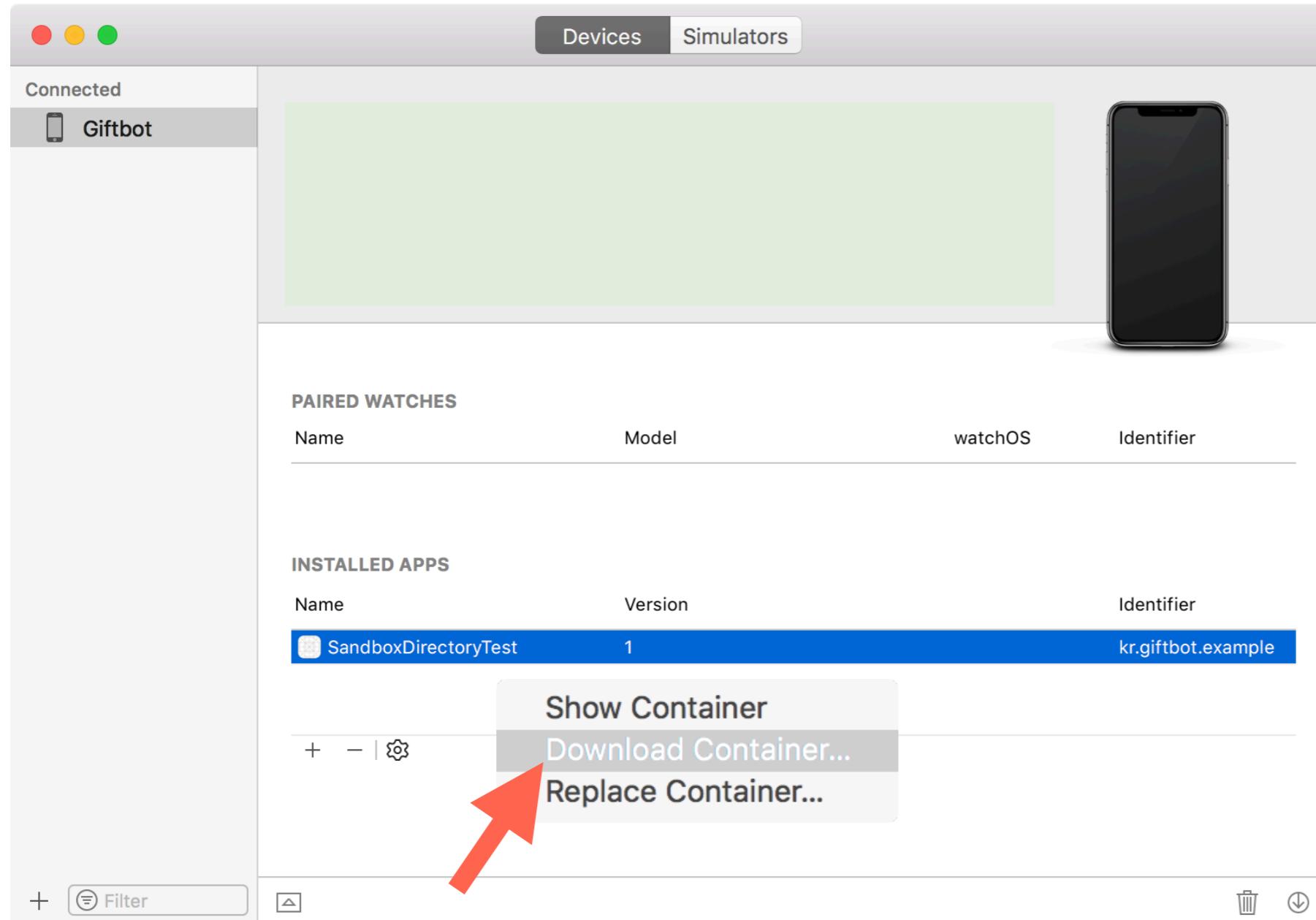
Download DataContainer



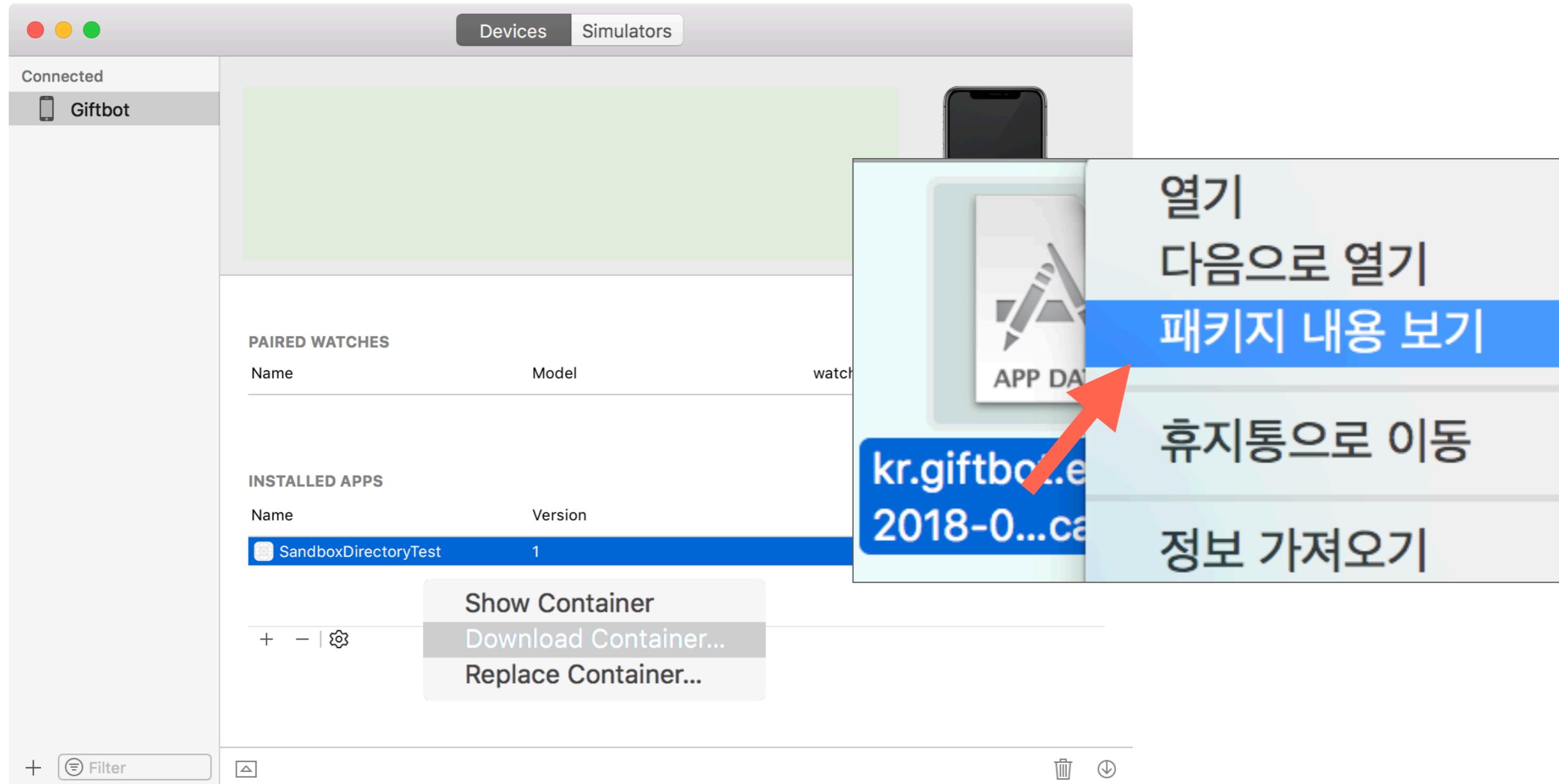
Download DataContainer



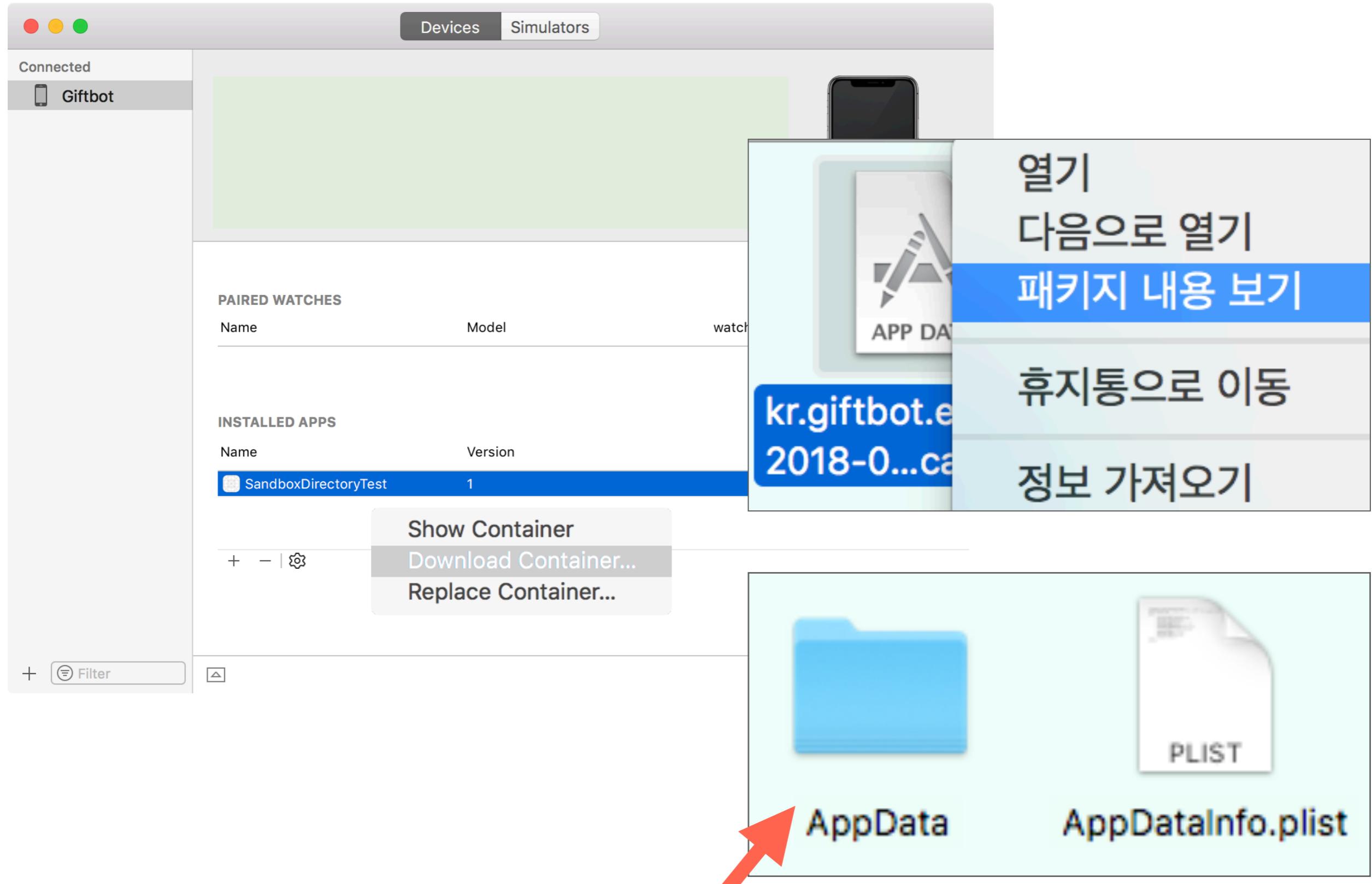
Download DataContainer

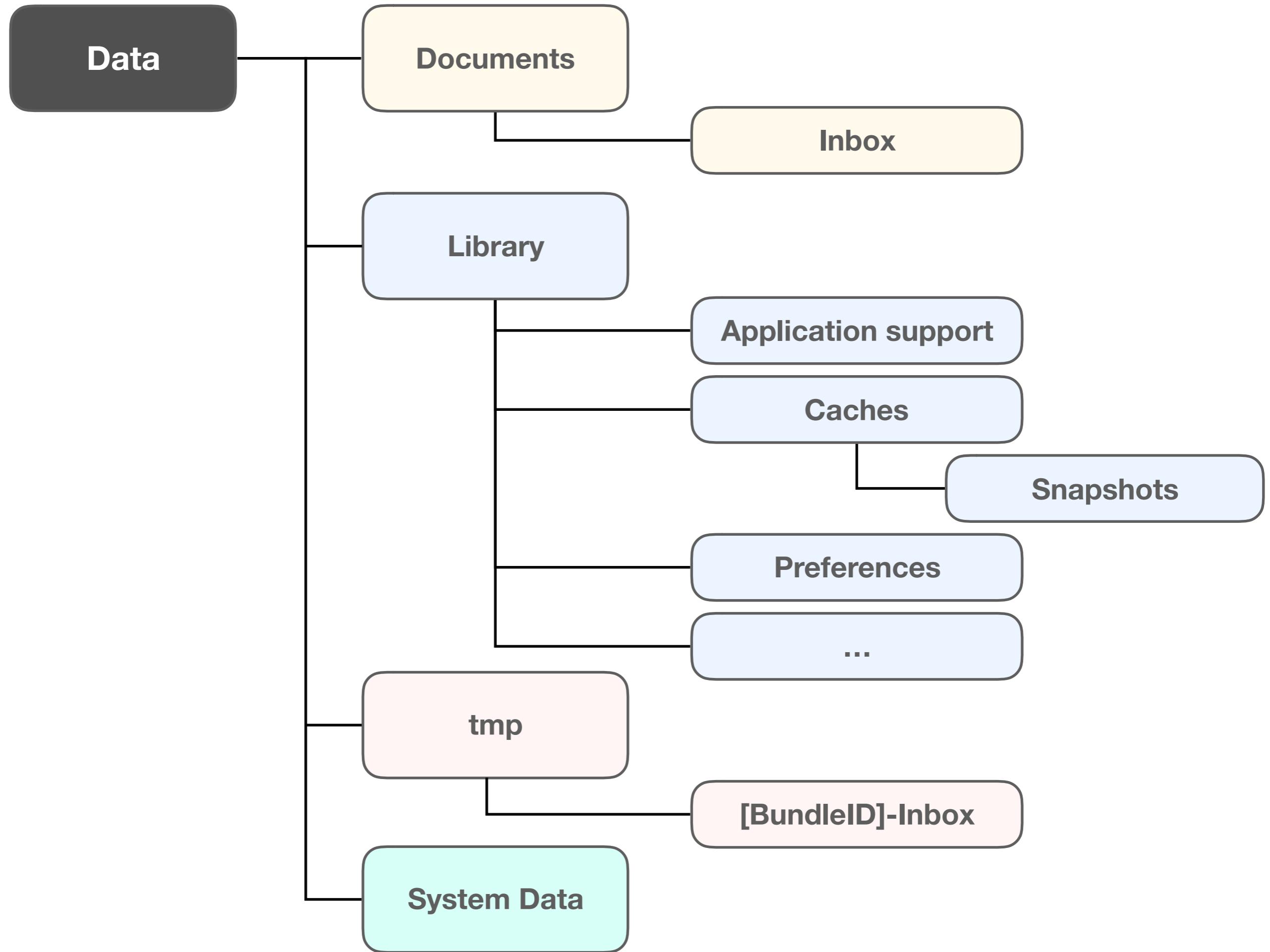


Download DataContainer

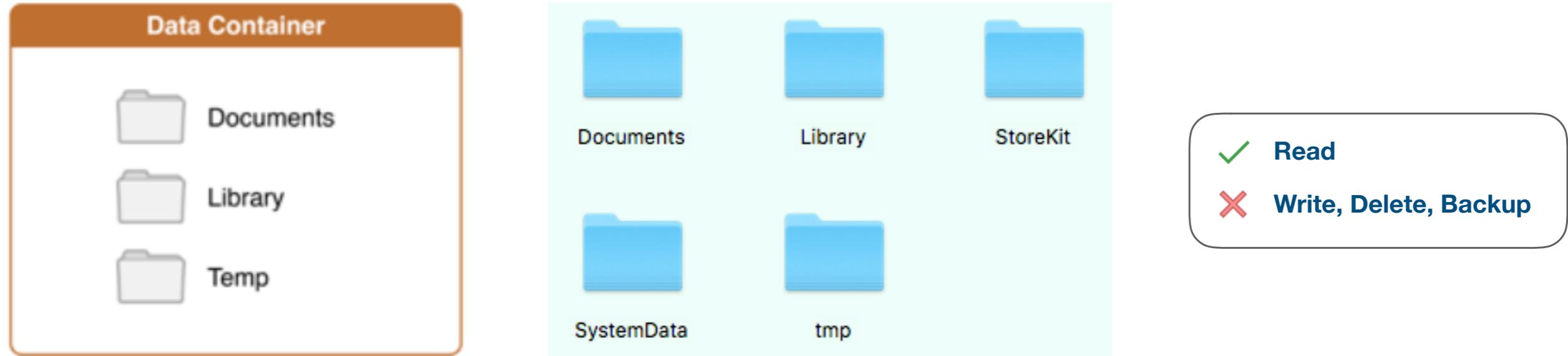


Download DataContainer





Data Container



NSHomeDirectory()

Data Container 홈 디렉토리

기본 디렉토리 - Documents, Library, tmp

사용자가 직접 디렉토리나 파일을 추가할 수 없으며 Documents 같은 서브디렉토리를 통해 관리

StoreKit 등 앱에서 사용하는 데이터에 따라 시스템이 그에 맞는 디렉토리 추가

iOS 11 부터 SystemData 디렉토리 추가 (읽기, 쓰기, 삭제 권한 모두 없음) - 현재 용도 불명확

Documents

- ✓ Read, Write, Backup
- ✗ Delete

[Directory]

- ✓ Read, Write, Delete, Backup
- ✗

[Files]

- `NSHomeDirectory() + "/Documents"`
- `FileManager.default.urls(for: .documentDirectory, in: .userDomainMask)[0]`
- `NSSearchPathForDirectoriesInDomains(.documentDirectory, .userDomainMask, true)[0]`

- 유저가 앱을 통해 생성한 문서나 데이터, 또는 외부 앱을 통해서 전송한 음악, pdf 등의 컨텐츠를 저장
- 설정에 따라 유저가 직접 파일 추가 및 삭제 가능

따라서 유저에 의해 삭제되거나 내용이 변경되어도 무방하고 유저가 다루는 컨텐츠와 관련이 있는 파일들만 저장

- Documents 디렉토리 자체는 삭제 불가

디렉토리 삭제 시도 시 Documents 내의 디렉토리, 파일들만 일괄 삭제

- iTunes, iCloud 에 백업

UIFileSharingEnabled

Information Property List	Dictionary (15 items)
Application supports iTunes file sharing	Boolean YES

The screenshot shows the iPhone Settings screen with the following details:

- Device Information:** Giftbot, 64GB, 90% battery.
- Sharing Options:** 파일 공유 (File Sharing) is selected, indicated by a red arrow.
- Available Apps for Sharing:** A list includes 리스닝드릴, 카카오톡, CamScanner, iMovie, Keynote, Numbers, Pages, and SandboxDirectoryTest (highlighted with a red arrow).
- Sandbox Directory Content:** Shows the contents of the SandboxDirectoryTest directory: Inbox (11MB, 3:35 ago) and text.txt (4KB, 7:50 ago).
- Bottom Navigation:** Includes tabs for 앱 (App), 42.93GB 사용 가능 (42.93GB available), 동기화 (Sync), and 완료 (Done).

Documents/Inbox

✓ **Read, Backup**

✗ **Write, Delete**

[Directory]

✓ **Read, Delete, Backup**

✗ **Write**

[Files]

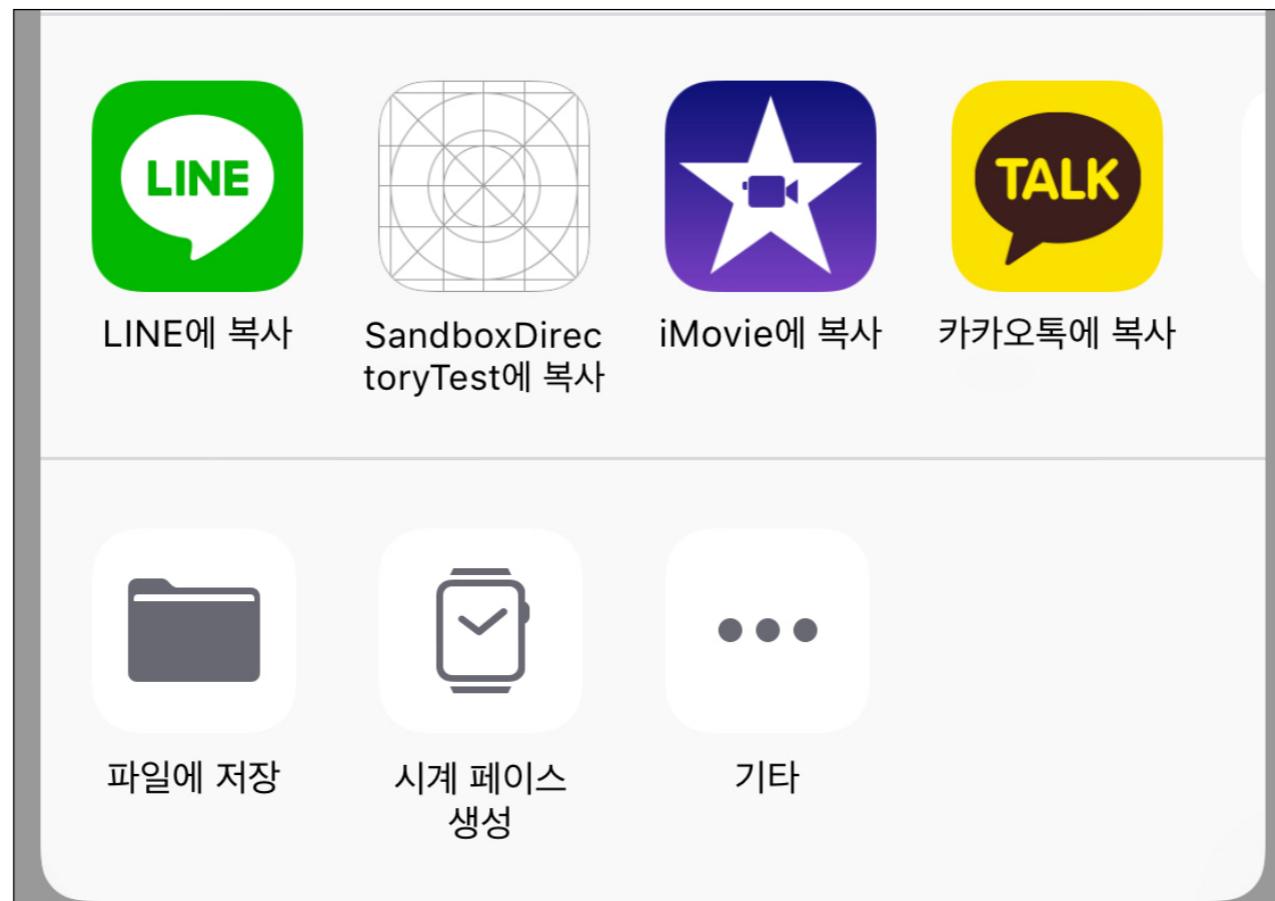
\$DocumentsPath + "Inbox"

- 타 앱을 통해 전송받은 파일이 저장되는 디렉토리 e.g.) 메일 앱 첨부파일 공유 등
- 파일들을 읽거나 삭제할 수는 있지만, 새 파일을 추가하거나 기존 파일 수정 불가
- 타 앱에서 동일 이름 파일 전달 시 덮어쓰기 대신 [file-1.txt, file-2.txt] 처럼 번호 자동 부여하며 새 파일 생성
- iOS 9.0 부터 디렉토리가 한 번 만들어진 이후로는 삭제 불가하며 파일만 삭제 가능. 유저가 직접 생성 불가

Documents 와 동일하게 디렉토리 삭제 시도 시 Inbox 내의 파일들만 일괄 삭제

CFBundleDocumentTypes

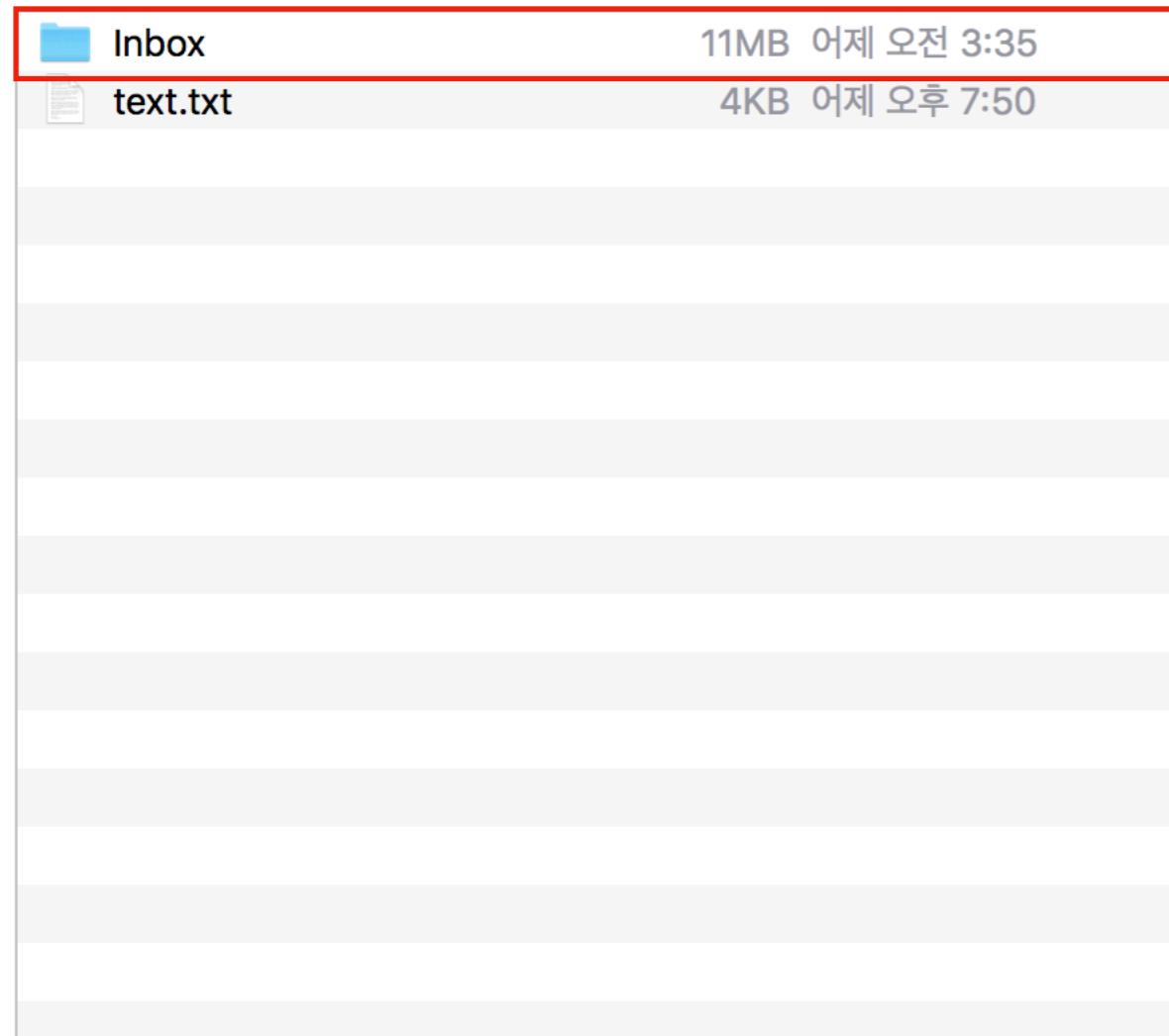
▼ Information Property List	Dictionary	(18 items)
▼ Document types	Array	(1 item)
▼ Item 0 (All Document Types)	Dictionary	(3 items)
Document Type Name	String	All Document Types
Handler rank	String	Default
▼ Document Content Typ...	Array	(1 item)
Item 0	String	public.item



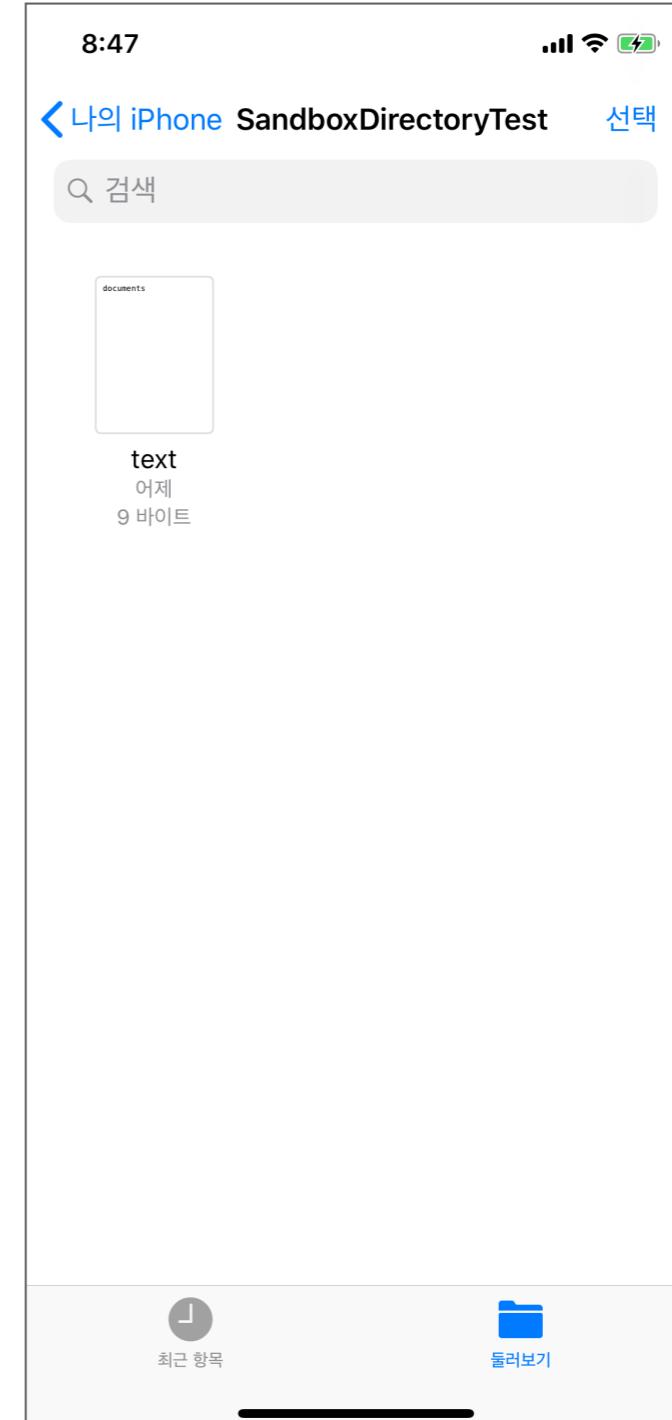
[UIDocumentInteractionController]

iTunes vs Files

SandboxDirectoryTest의 도큐멘트



[iTunes]



[Files]

application(_:open:options:)

```
func application(  
    _ app: UIApplication,  
    open url: URL,  
    options: [UIApplicationOpenURLOptionsKey : Any] = [:]  
) -> Bool {  
    return true  
}
```

url - \$HomePath/Documents/Inbox/test.pdf

sourceApplication - com.apple.mobilemail

openInPlace - false

annotation - nil

tmp/[BundleID]-Inbox

✓ Read, Write, Delete

✗ Backup

[Directory]

✓ Read, Write, Delete

✗ Backup

[Files]

NSTemporaryDirectory() + Bundle.main.bundleIdentifier! + “-Inbox”

- LSRequiresOpeningDocumentsInPlace 키의 값이 NO 일 때

iCloud 드라이브 관련 / Files 앱 / DocumentsPicker 등에서 공유되는 파일들이 저장되는 디렉토리

- Documents/Inbox 와 달리 디렉토리 및 파일을 다루는 제약 없음

- tmp 디렉토리의 속성을 따르므로 백업되지 않으며 시스템에 의해 삭제될 수 있음

LSSupportsOpeningDocumentsInPlace

Key	Type	Value
▼ Information Property List	Dictionary	(17 items)
Supports opening documents in place	Boolean	YES

- NO - 원본 파일에 대한 사본을 Inbox 디렉토리에 생성
- YES - 원본 파일 url 을 그대로 사용. File Coordinator, File Presenter 등을 통한 관리 필요

- NO - 사본 url

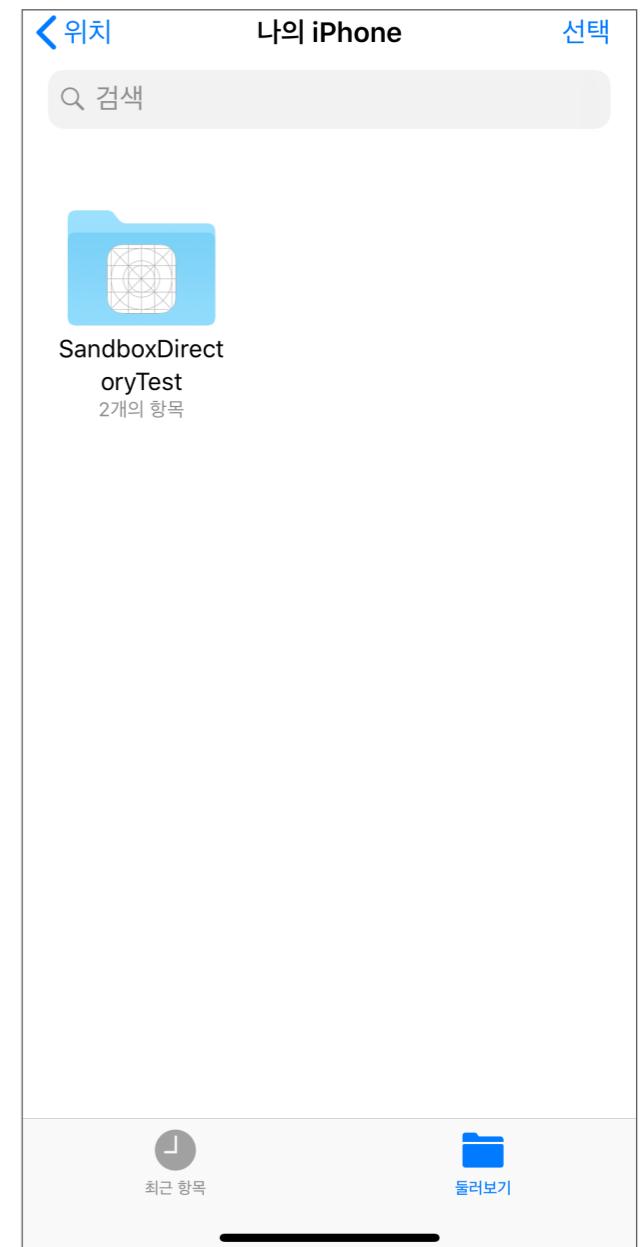
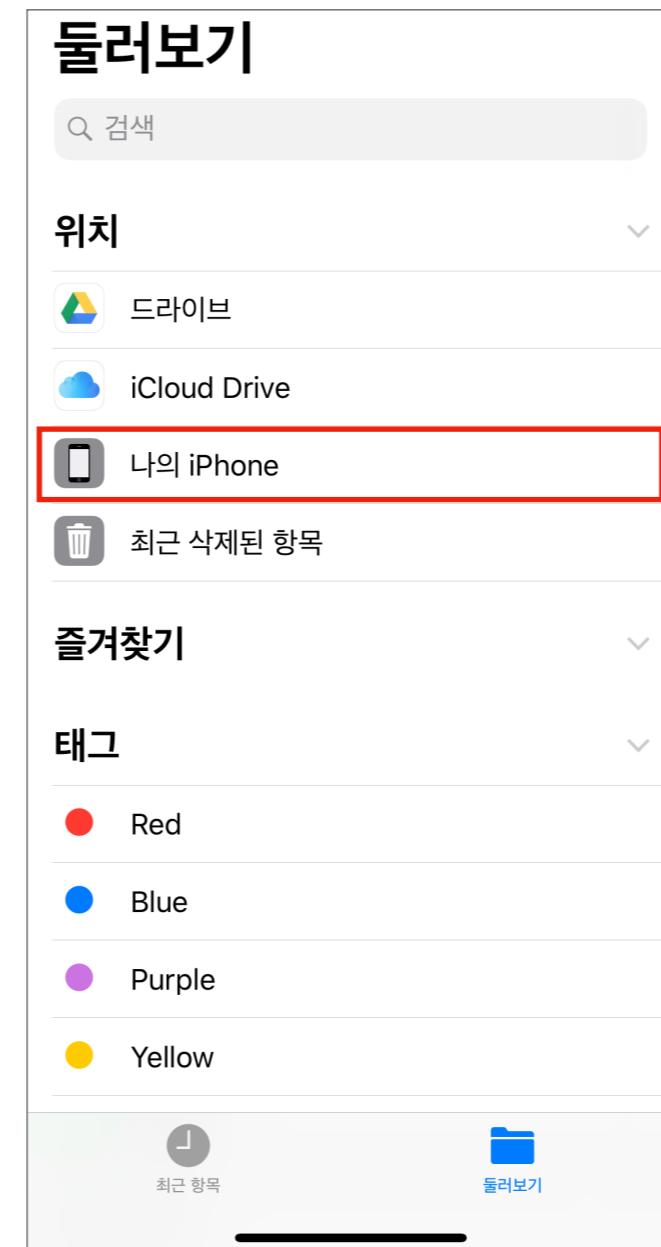
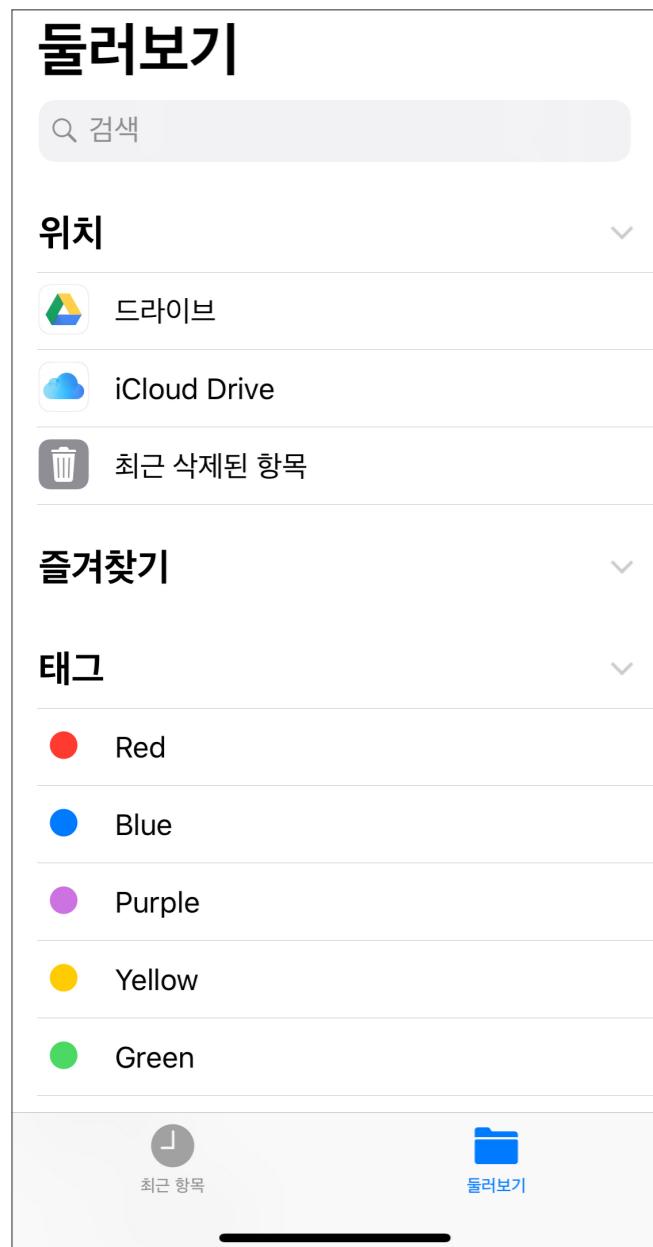
file:///private/var/mobile/Containers/Data/Application/[UUID]/tmp/kr.giftbot.demo-Inbox/file.key

- YES - 원본 url

file:///private/var/mobile/Library/Mobile%20Documents/com~apple~Keynote/Documents/file.key

Files App

- **UIFileSharingEnabled = YES**
- **LSSupportsOpeningDocumentsInPlace = YES**



UIDocumentBrowserViewController

The screenshot displays the `UIDocumentBrowserViewController` interface on an iPad. The left sidebar contains navigation sections: "Locations" (with "MyFileProvider", "iCloud Drive", and "On My iPad" selected), "Favorites" (empty), and "Tags" (listing "Red", "Orange", "Yellow", "Green", "Blue", "Purple", "Gray", "Work", and "Home"). The main content area shows a search bar at the top with placeholder "Search". Below it is a toolbar with a "Create Document" button (containing a plus sign) and a "Select" button. The main view lists documents and folders:

Item Type	Name	Created On	Last Modified	Size
Image	Cooking	May 30, 2017	May 30, 2017	1.8 MB
Folder	Dad's Birthday			16 items
Image	Fish	May 30, 2017	May 30, 2017	832 KB
Image	Flowers	May 30, 2017	May 30, 2017	555 KB
Folder	Hawaii Vacation			8 items
Image	Landscape	May 30, 2017	May 30, 2017	2.4 MB
Image	Stingray	May 30, 2017	May 30, 2017	453 KB

At the bottom, there are two tabs: "Recents" and "Browse".

Library

- ✓ Read, Write, Backup
- ✗ Delete

[Directory]

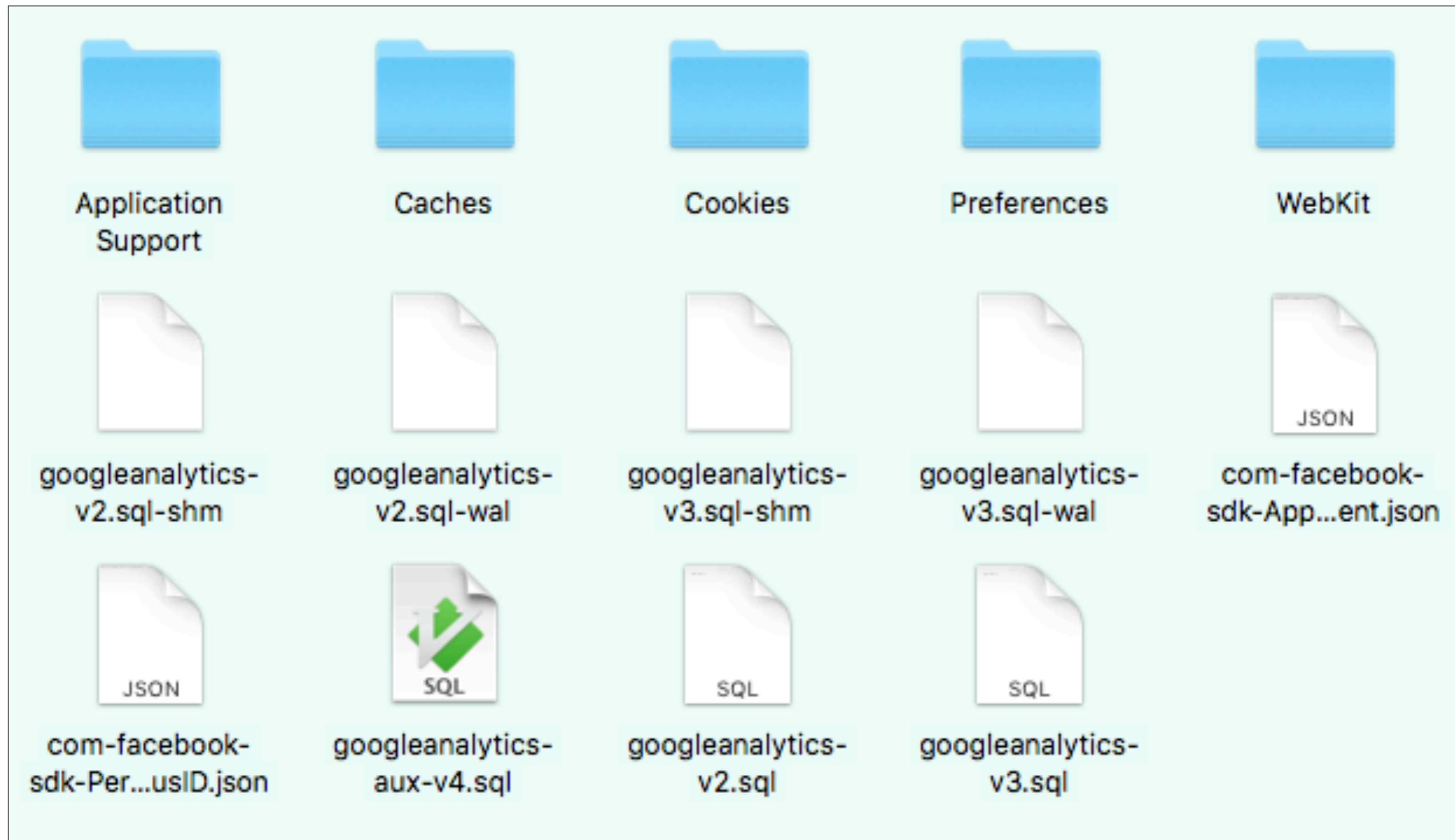
- ✓ Read, Write, Delete, Backup
- ✗

[Files]

- `NSHomeDirectory() + "/Library"`
- `FileManager.default.urls(for: .libraryDirectory, in: .userDomainMask)[0]`
- `NSSearchPathForDirectoriesInDomains(.libraryDirectory, .userDomainMask, true)[0]`

- 유저 데이터 파일 및 임시 파일을 제외한 모든 파일들을 관리
- 유저에게 노출되는 것을 피하고 앱의 기능이나 관리에 필요한 파일 저장
- 주로 서브디렉토리인 Application Support 와 Caches 를 이용하지만 커스텀 디렉토리 사용 가능
- Preference, Cookies, Saved Application State, WebKit 등 필요할 때 시스템에서 자동 생성
- iTunes, iCloud 에 백업

Use subdirectory



Library/Application Support

✓ Read, Write, Delete, Backup
✗

[Directory]

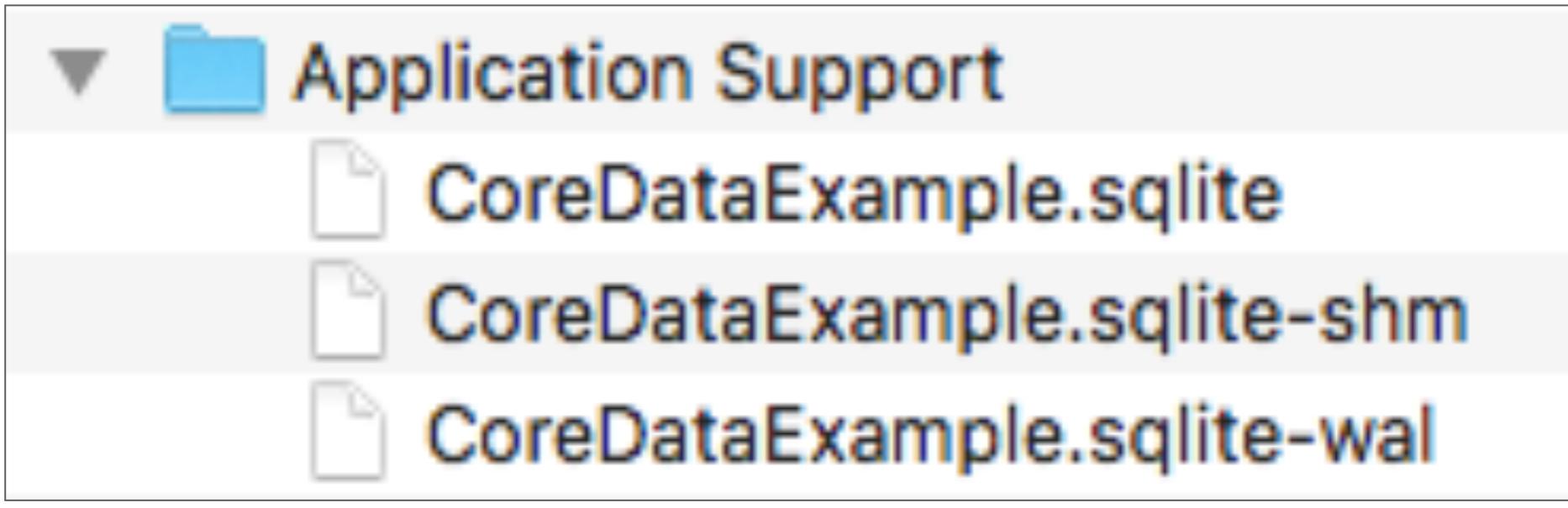
✓ Read, Write, Delete, Backup
✗

[Files]

- `FileManager.default.urls(for: .applicationSupportDirectory, in: .userDomainMask)[0]`
- `NSSearchPathForDirectoriesInDomains(.applicationSupportDirectory, .userDomainMask, true)[0]`

- 앱의 기능 또는 관리를 위해 지속적으로 관리해야 되는 파일 저장
- Documents 와 거의 동일한 속성을 지니며, 유저에 대한 노출 여부에 따라 위치 결정
- BundleID 나 회사명 등의 서브디렉토리를 만들어 관리
- CoreData 기본 저장 경로
- iTunes, iCloud 에 백업

CoreData vs Realm

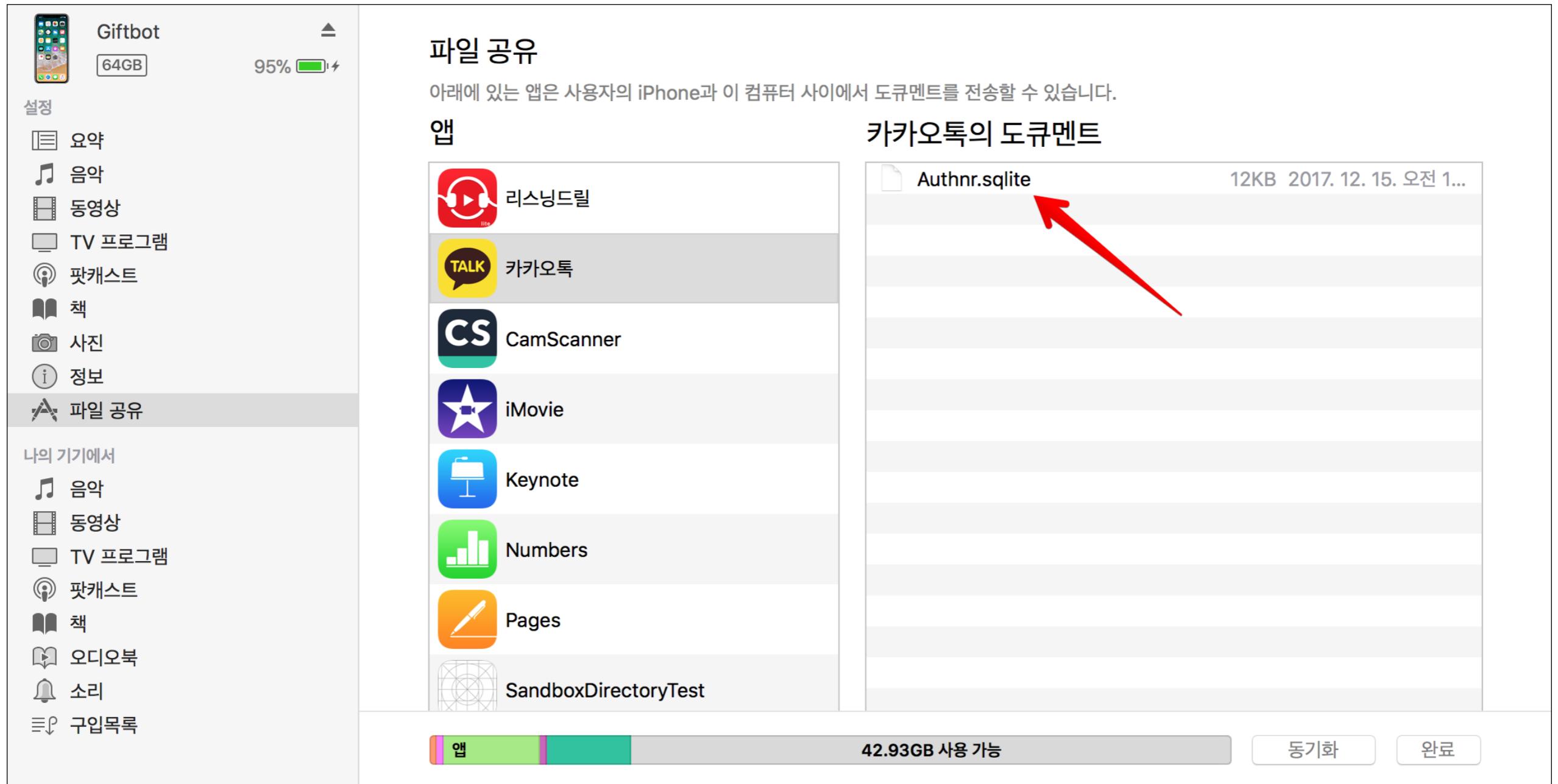


[CoreData]



[Realm]

Authnr.sqlite??



Remove from Documents

데이터베이스 구조 데이터 보기 Pragma 설정 SQL 실행

테이블 생성하기(C) 인덱스 생성하기(I) 테이블 수정하기 테이블 삭제하기

이름	타입	스키마
▼ 테이블 (1)		
keyinfotbl		CREATE TABLE keyinfotbl (aaid BLOB(9) NOT NULL, keyid BLOB NOT NULL, `aaid` BLOB (9) NOT NULL, `keyid` BLOB NOT NULL, signcounter INTEGER `signcounter` INTEGER NOT NULL, prikey BLOB `prikey` BLOB NOT NULL, appid TEXT `appid` TEXT NOT NULL, username TEXT `username` TEXT NOT NULL)
aaid	BLOB (9)	
keyid	BLOB	
signcounter	INTEGER	
prikey	BLOB	
appid	TEXT	
username	TEXT	

인덱스 (0)
뷰 (0)
트리거 (0)

데이터베이스 구조 데이터 보기 Pragma 설정 SQL 실행

테이블: keyinfotbl

aaid	keyid	signcounter	prikey	appid	username
필터	필터	필터	필터	필터	필터

새 레코드 레코드 삭제

0 - 0 of 0 특정 레코드 행으로 가기: 1

The screenshot shows a database management interface with two main panes. The left pane displays the database schema, specifically a single table named 'keyinfotbl' with six columns: 'aaid', 'keyid', 'signcounter', 'prikey', 'appid', and 'username'. The right pane shows the data view for this table, which is currently empty (0 rows). Navigation and search tools are visible at the bottom of the data view.

Library/Caches

- ✓ Read, Write, Delete
- ✗ Backup

[Directory]

- ✓ Read, Write, Delete
- ✗ Backup

[Files]

- `FileManager.default.urls(for: .cachesDirectory, in: .userDomainMask)[0]`
- `NSSearchPathForDirectoriesInDomains(.cachesDirectory, .userDomainMask, true)[0]`

- 앱의 동작 속도 / 데이터 절약 등을 위해 사용하는 공간
- 필요하면 쉽게 재생성 / 다운로드 받을 수 있는 파일이어야 함 e.g.) 잡지, 신문, 지도 정보, json 파일 등
- 디스크 공간이 부족하거나 시스템 복원, 앱 업데이트 등의 경우에 시스템이 자동으로 파일 삭제 가능
- 어느 순간 삭제되어도 앱의 동작에는 영향이 없을 파일만 저장
- 앱이 실행 중에는 삭제되지 않는 것을 보장
- 백업되지 않음

Library/Caches/Snapshots



AdMob



com.apple.WebKit
.Networking



com.apple.WebKit
.WebContent



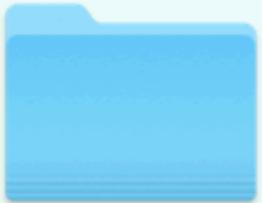
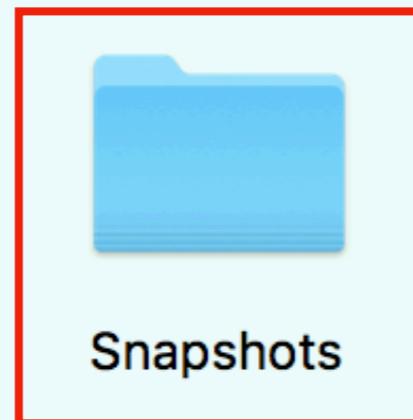
com.crashlytics.d
ata



io.fabric.sdk.ios.d
ata



kr.giftbot.example



WebKit



https_googleads.
g.doubleclick.netstorage

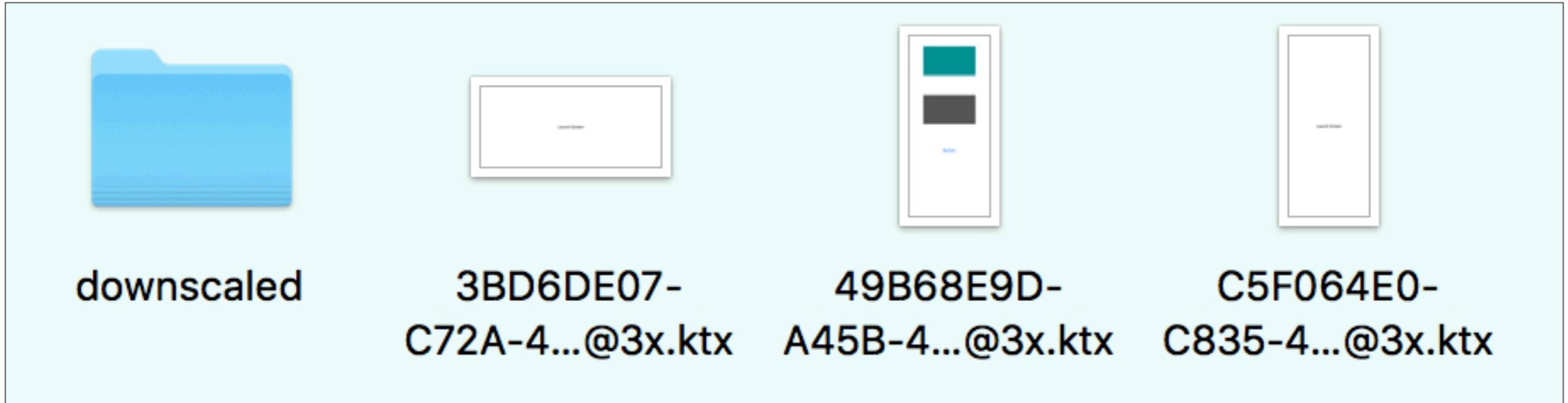


https_googleads.
g.doubleclick.netstorage.shm

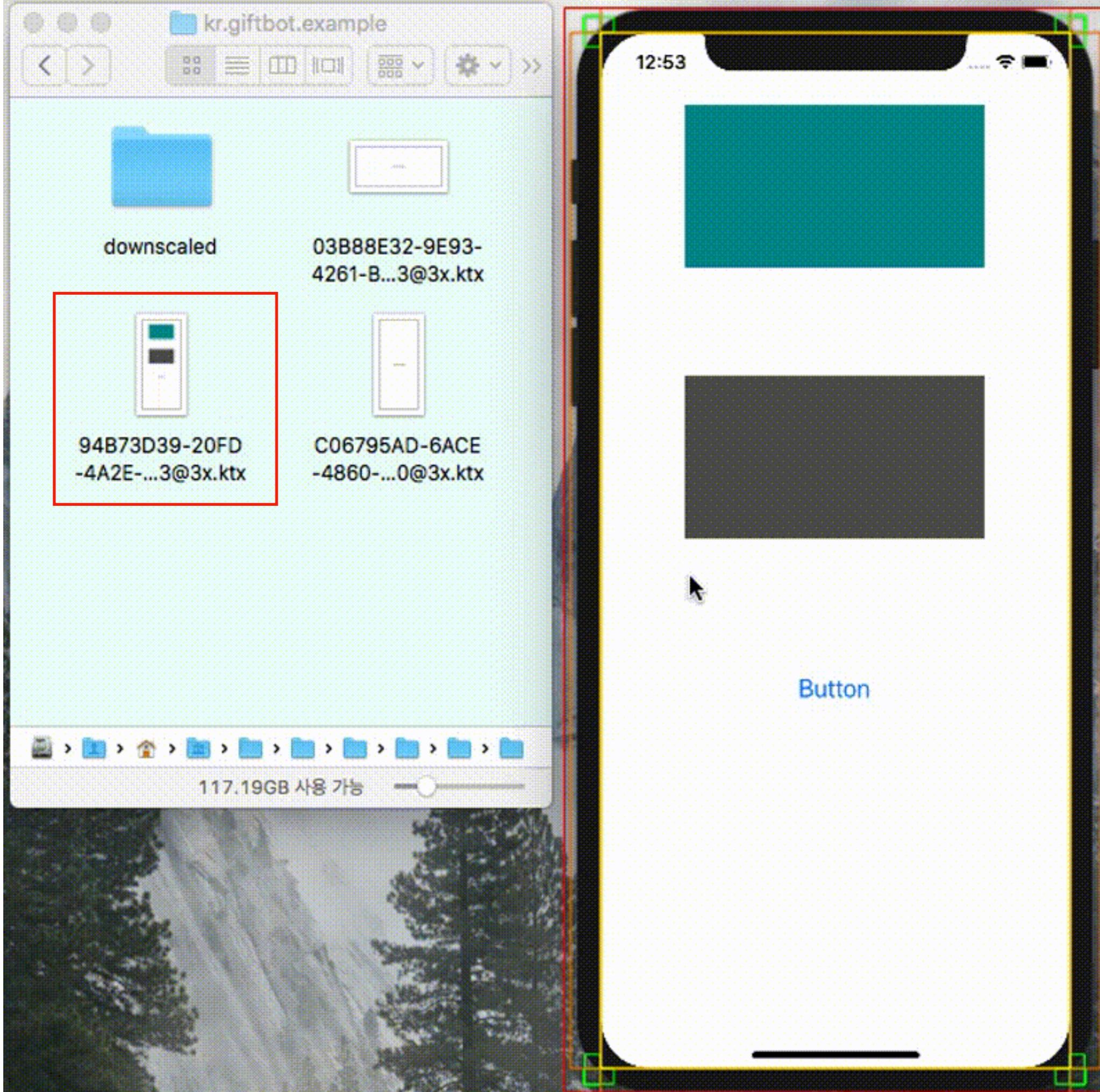


https_googleads.
g.doubleclick.netstorage.rage-wal

Library/Caches/Snapshots



- **applicationDidEnterBackground(_:) 메서드 호출 후** 현재 뷰에 대한 스냅샷 생성
- **Background**에서 **Foreground**로 넘어올 때 스냅샷 이미지 사용
- **Restore Application State** 기능 사용 시, **Suspend** 상태에서 종료되었던 앱이 재실행될 때 사용자에게 앱이 종료되지 않고 계속 실행되고 있었던 느낌을 주기 위해 **Snapshot** 이미지 사용

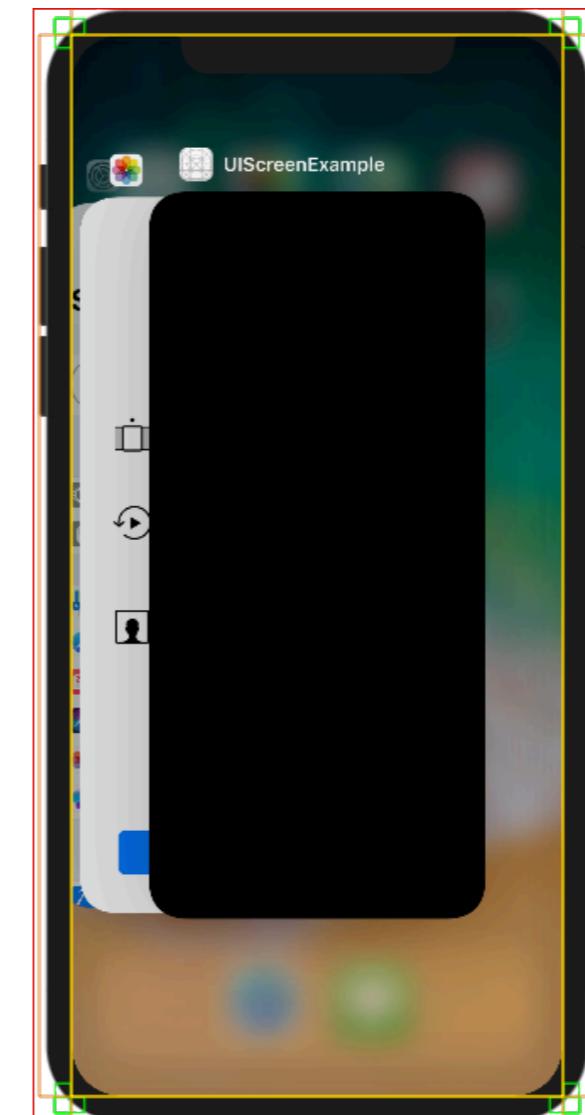
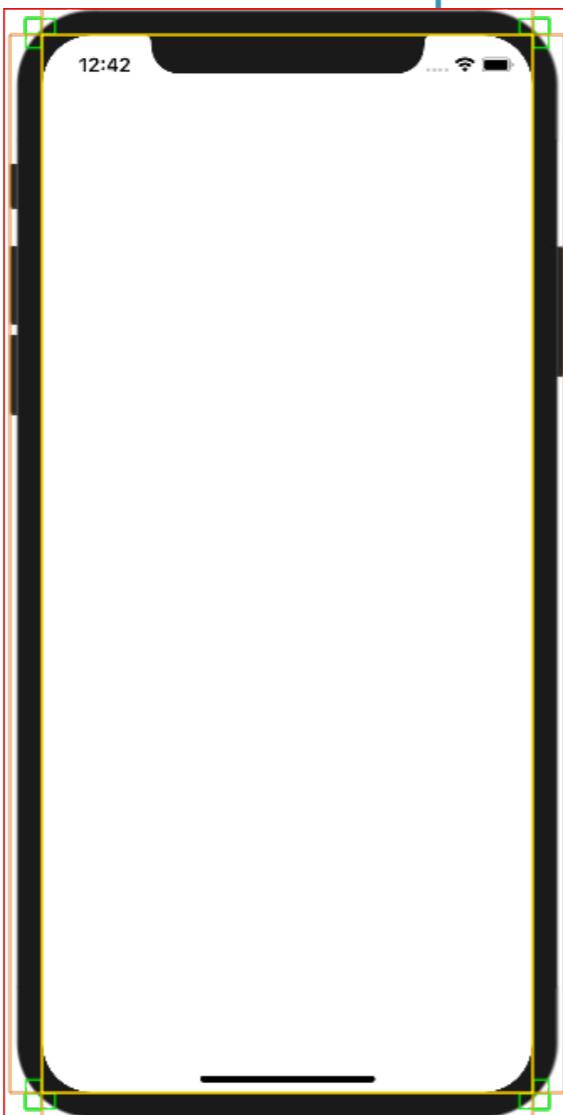


Hide Sensitive Data

- **Cover Screen**
- **UIApplicationExitsOnSuspend**
- **ignoreSnapshotOnNextApplicationLaunch**

Cover screen

```
func applicationWillResignActive(_ application: UIApplication) {  
    coverView.backgroundColor = .black  
    window?.addSubview(coverView)  
}  
func applicationDidBecomeActive(_ application: UIApplication) {  
    if window!.contains(coverView) {  
        coverView.removeFromSuperview()  
    }  
}
```



UIApplicationExitsOnSuspend

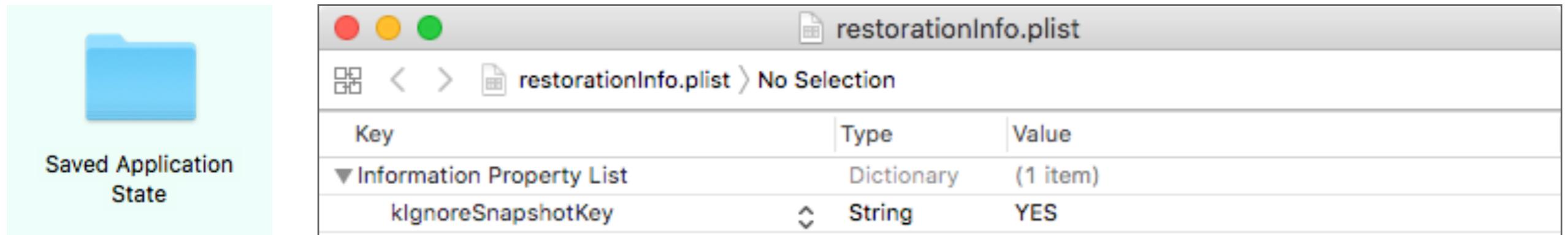
YES 설정 시, Background 로 진입할 때 Suspend 상태 없이 앱을 바로 종료 (terminate)

Key	Type	Value
▼ Information Property List	Dictionary	(15 items)
Application does not run in background	Boolean	YES

ignoreSnapshotOnNextApplicationLaunch

(앱 상태 복원을 사용할 때 적용) 다음 앱 실행 시, 스냅샷 이미지를 사용하지 않도록 함.

```
func application(_ application: UIApplication, shouldSaveApplicationState coder: NSCoder) -> Bool {  
    return true  
}  
func application(_ application: UIApplication, willEncodeRestorableStateWith coder: NSCoder) {  
    UIApplication.shared.ignoreSnapshotOnNextApplicationLaunch()  
}  
func application(_ application: UIApplication, shouldRestoreApplicationState coder: NSCoder) -> Bool {  
    return true  
}  
func application(_ application: UIApplication, didDecodeRestorableStateWith coder: NSCoder) {  
}
```



Library/Preferences

- ✓ Read, Write, Backup
- ✗ Delete

[Directory]

- ✓ Read, Write, Delete, Backup
- ✗

[Files]

`$LibraryPath + "Preferences"`

- 앱의 설정 정보 저장
- 별도의 파일을 저장할 수는 있지만 직접 조작하지 않는 것을 권장
- 대신 `User Defaults`, `Settings.bundle`, `CFPreferences API` 등을 이용할 것
- `iTunes`, `iCloud` 에 백업

tmp(Temporary)

- ✓ Read, Write, Delete
- ✗ Backup

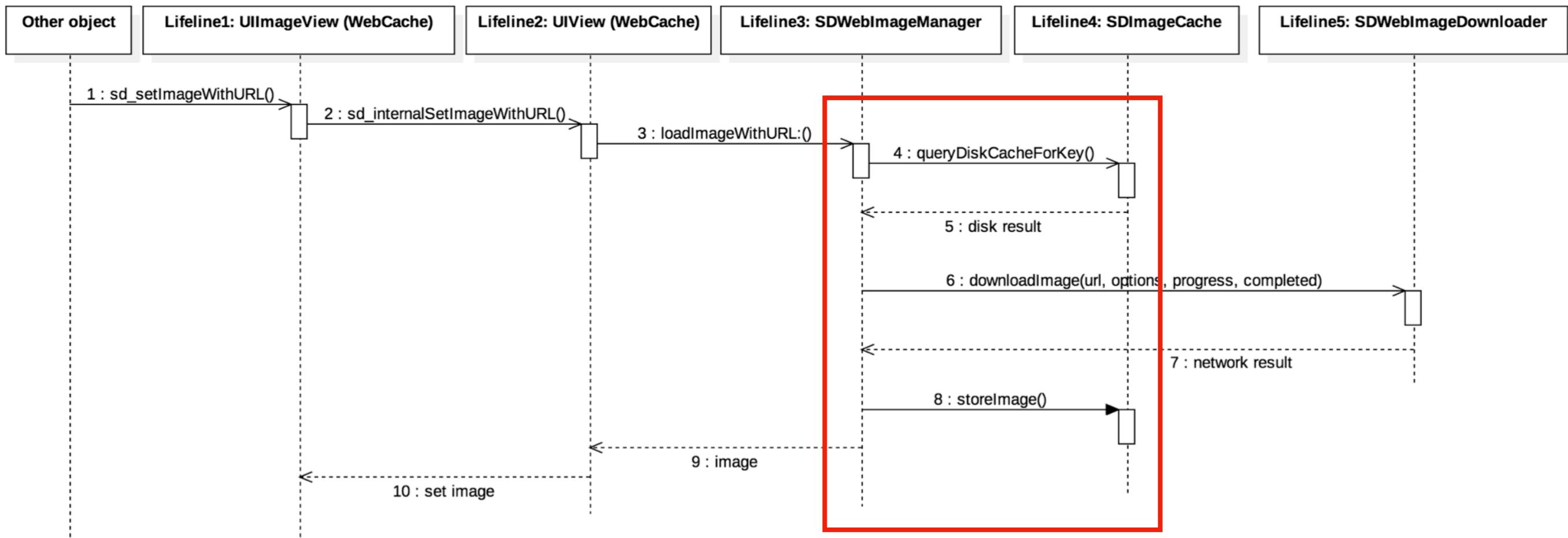
[Directory]

- ✓ Read, Write, Delete
- ✗ Backup

[Files]

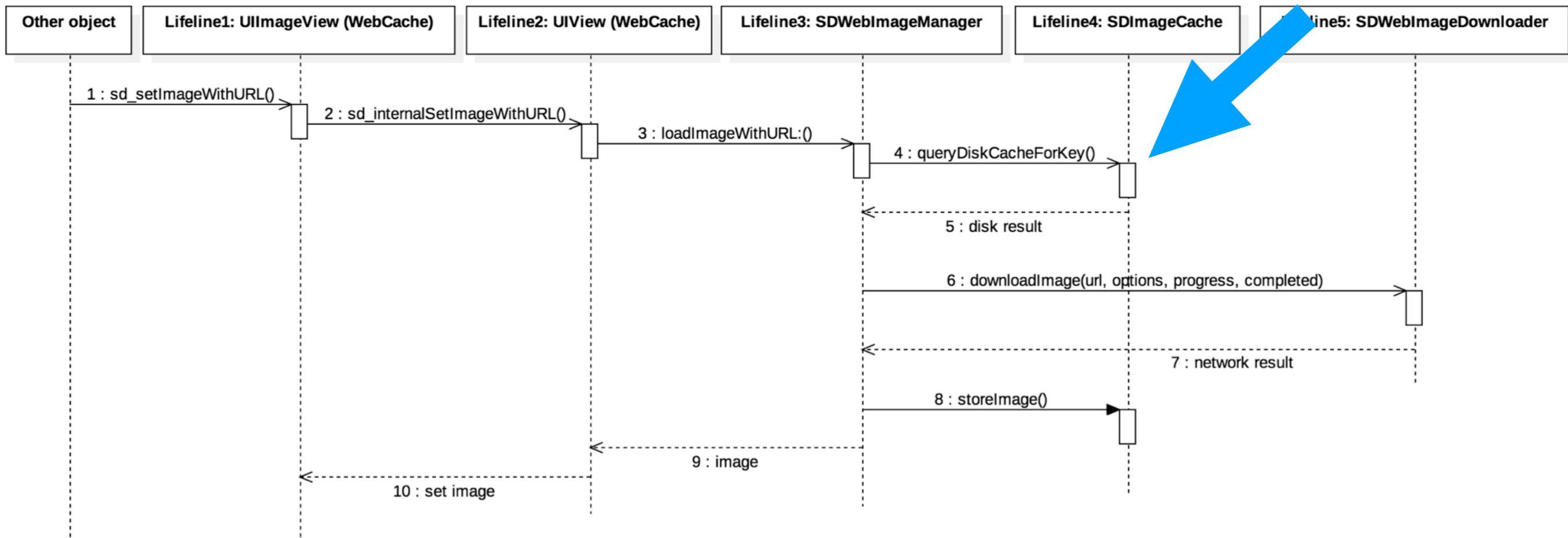
- [NSTemporaryDirectory\(\)](#)
- [FileManager.default.temporaryDirectory](#)
- 현재 앱 실행 중에 사용하지만 다음 앱 런칭시까지 유지할 필요 없는 임시 파일 저장
- 시스템이 주기적으로 파일을 삭제하더라도, 사용 후 필요없어진 파일은 직접 삭제해주는 것을 권장
- 앱이 실행되고 있는 동안에는 삭제되지 않음
- 백업되지 않음

SDWebImage Flow



QueryDiskCache: Library/Caches

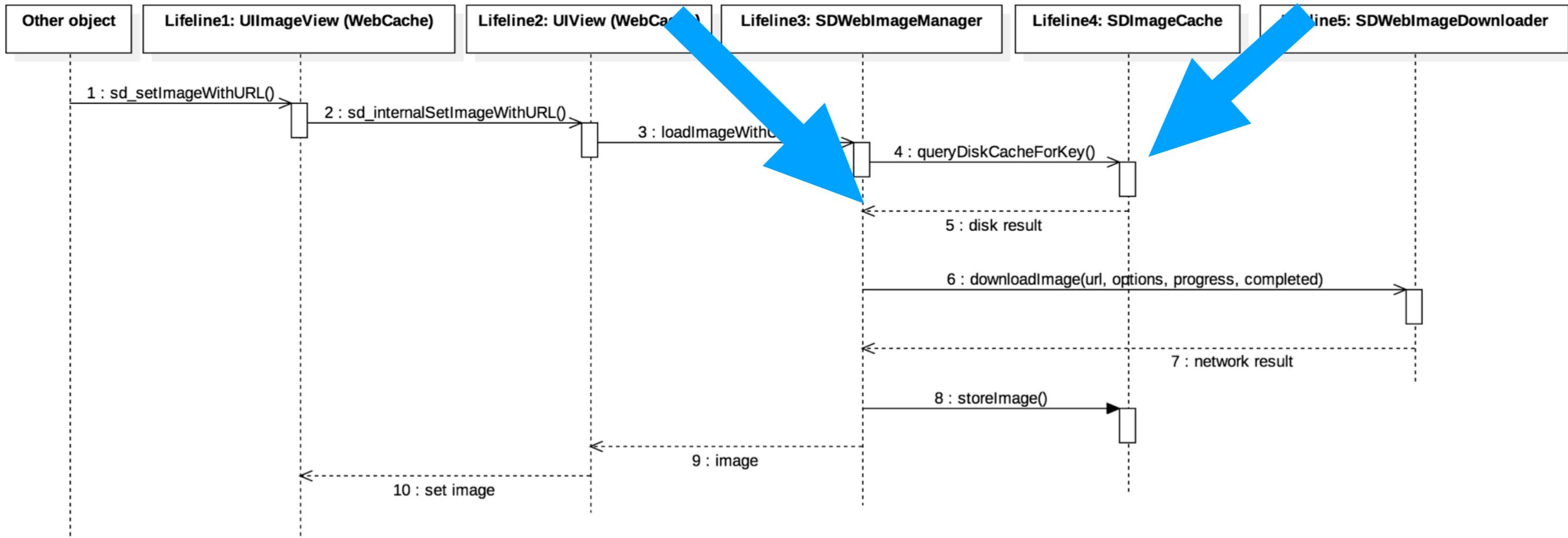
Library/Caches



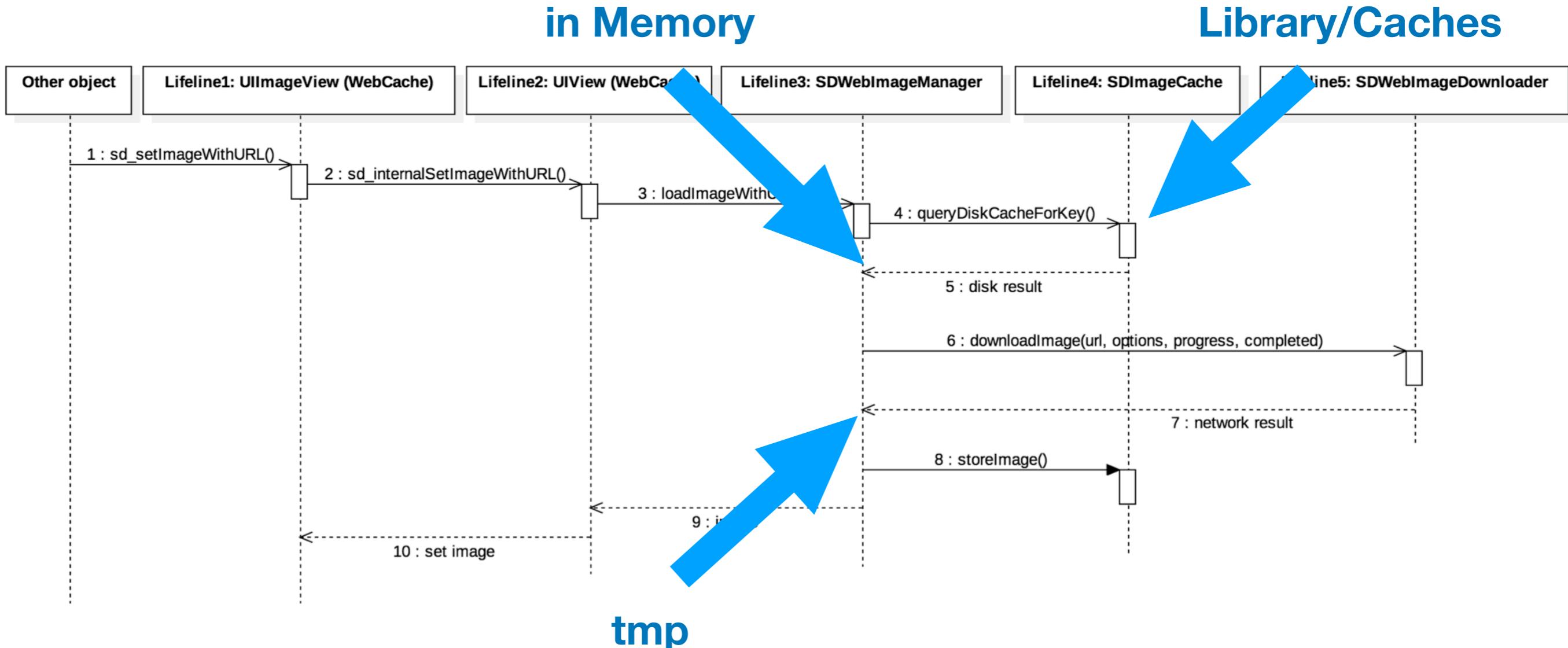
Disk Result: in Memory

in Memory

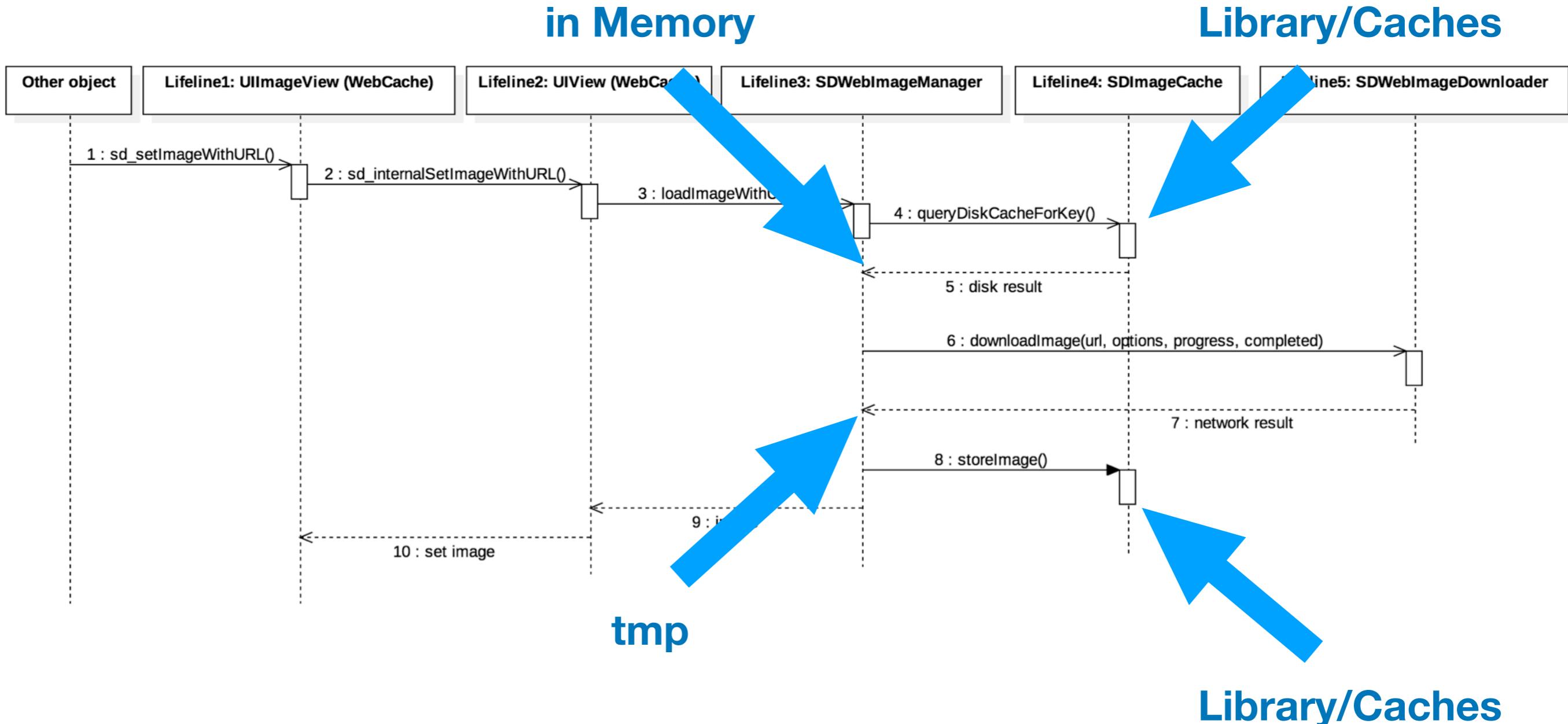
Library/Caches



Network Result: tmp



Store Image: Library/Caches





Tips

Do not backup attribute

URLResourceKey.isExcludedFromBackupKey (NSURLIsExcludedFromBackupKey)

- 백업 대상에서 제외 + 용량 부족 시에도 삭제 방지
- 디렉토리와 무관하게 적용 가능
- 주기적으로 모니터링하여 필요 없는 파일은 수동 제거 필요
- 백업되는 디렉토리 (Documents, Library) 에 허용되지 않는 파일을 넣었을 경우 리젝되는 것을 방지

```
@discardableResult
func excludeFileFromBackup(at filePath: String) -> Bool {
    var url = URL(fileURLWithPath: filePath)
    do {
        var values = try url.resourceValues(forKeys: [.isExcludedFromBackupKey])
        let isExcluded = values.isExcludedFromBackup ?? false
        guard !isExcluded else { return true }
        values.isExcludedFromBackup = true
        try url.setResourceValues(values)
        return true
    } catch {
        return false
    }
}
```

A User's iCloud Storage Is Limited

- **DO** store the following in iCloud:
 - User documents
 - App-specific files containing user-created data
 - Preferences and app state (using key-value storage, which does not count against a user's iCloud storage allotment)
 - Change log files for a SQLite database (a SQLite database's store file must never be stored in iCloud)
- **DO NOT** store the following in iCloud:
 - Cache files
 - Temporary files
 - App support files that your app creates and can recreate
 - Large downloaded data files

App Store Review Guidelines

2.23: Apps must follow the iOS Data Storage Guidelines or They Will be rejected

appstore approval - Rejection: "2.23: Apps must follow the iOS ..."

<https://stackoverflow.com/.../rejection-2-23-apps-must-follow-the...> ▾ 이 페이지 번역하기

2014. 7. 9. - Just because the Documents directory is empty and you are setting the do-not-backup flag on your directory in Application Support does not mean there is nothing else that could get backed up. The App Programming Guide for iOS in the Performance Tips chapter has a section which states: App Backup ...

iphone - iOS App rejected: 2.23 - Apps must follow the iOS Data ... 답변 2 2016년 5월 3일

2.23 - Apps must follow the iOS Data Storage Guidelines or they ... 답변 2 2015년 3월 5일

iphone - Rejection: "2.23: Apps must follow the iOS Data Storage ... 답변 1 2014년 10월 22일

objective c - 2.23: Apps must follow the iOS Data Storage ... 답변 2 2014년 3월 2일

[stackoverflow.com 검색결과 더보기](#)

2.23 - Apps must follow the iOS Data Storage Guidelines or they will ...

<https://github.com/realm/realm-cocoa/issues/2016> ▾ 이 페이지 번역하기

2.23 - Apps must follow the iOS Data Storage Guidelines or they will be rejected #2016. Closed. vlonjatg opened this Issue on May 27, 2015 · 11 comments ...

iOs Data Storage guidelines reject (2.23 - App Store Review)

www.francescoficietola.it/.../ios-ios-data-storage-guidelines-reject-... ▾ 이 페이지 번역하기

2012. 7. 11. - Brutta parola, "Apple reject". La strada per farsi approvare un'app da Apple per il rilascio su Apple Store è assai tortuosa. Occorre seguire per filo e per segno ciascun punto delle App Store Review Guidelines. C'è un punto (il 2.23) che riguarda le iOS Data Storage Guidelines che onestamente non riesco a ...

2.23: Apps must follow the iOS Data Storage Guidelines or they ...

forums.gamesalad.com/.../iPhone-and-Android-Publishing ▾ 이 페이지 번역하기

2015. 1. 7. - 댓글 11 - 작성자 5

Hello,I'm pro user,I make sure every well. But today apple rejected it,why? Reasons. 2.23: Apps must follow the iOS Data Storage Guidelines or they will be rejected ----- 2.23 -----. We found that your app does not follow the iOS Data Storage Guidelines, which is required per the App Store Review ...

App Store Review Guidelines

2.23: Apps must follow the iOS Data Storage Guidelines or They Will be rejected



updated, but ...

2.2 Beta Testing

Demos, betas, and trial versions of your app don't belong on the App Store – use TestFlight instead. Any app submitted for beta distribution via TestFlight should be intended for public distribution and should comply with the App Review Guidelines. Note, however, that apps using TestFlight cannot be distributed to testers in exchange for compensation of any kind, including as a reward for crowd-sourced funding. Significant updates to your beta build should be submitted to TestFlight App Review before being distributed to your testers. To learn more, visit the [TestFlight Beta Testing](#).

2.3 Accurate Metadata

Customers should know what they're getting when they download or buy your app, so make sure your app description, screenshots, and previews accurately reflect the app's core experience and remember to keep them up-to-date with new versions.

2.3.1 Don't include any hidden or undocumented features in your app; your app's functionality should be clear to end-users and App Review. Similarly, you should not market your app on the App Store or offline as including content or services that it does not actually offer (e.g. iOS-based virus and malware scanners). Egregious or repeated behavior is grounds for removal from the Developer Program. We work hard to make the App Store a trustworthy ecosystem and expect our app developers to follow suit; if you're dishonest, we don't want to do business with you.

iOS Data Storage Guidelines

Development Guidelines

- [!\[\]\(f80140652f308cecd214820f98498cdd_img.jpg\) App Programming Guide](#)
- [!\[\]\(478a5a804f5b8ef82db39072f2942b4c_img.jpg\) App Extension Programming Guide](#)
- [!\[\]\(e9dfcca1a2ca7d0fda66af7c00858200_img.jpg\) iOS Data Storage Guidelines](#)
- [!\[\]\(f14cf09a29dc22f02201057b8859c85d_img.jpg\) macOS File System Documentation](#)
- [!\[\]\(a5f1339c7e49af1160b888c85dfd726e_img.jpg\) Safari Extensions Development Guide](#)
- [!\[\]\(34a0928158f9afd9d57ad2d8dbeb06f0_img.jpg\) iTunes Connect Developer Help](#)



iOS Data Storage Guidelines

Overview

iCloud includes Backup, which automatically backs up a user's iOS device daily over Wi-Fi. Everything in your app's home directory is backed up, with the exception of the application bundle itself, the caches directory, and temp directory. Purchased music, apps, books, the Camera Roll, device settings, home screen and app organization, messages, and ringtones are backed up as well. Because backups are done wirelessly and stored in iCloud for each user, it's best to minimize the amount of data that's stored for your app. Large files will lengthen the time it takes to perform a backup and consume more of a user's available iCloud storage.

Recommendation

- 디스크에 파일을 읽고 쓰는 것은 컴퓨터의 가장 느린 작업 중 하나. 파일 작업 수를 최소화

동일 파일에 대한 다중 작업 - 가능한 그룹화하여 일괄 처리

실제 데이터가 필요할 때까지 Dist I/O 연산 최대한 지연

Recommendation

- 디스크에 파일을 읽고 쓰는 것은 컴퓨터의 가장 느린 작업 중 하나. 파일 작업 수를 최소화
동일 파일에 대한 다중 작업 - 가능한 그룹화하여 일괄 처리
실제 데이터가 필요할 때까지 Dist I/O 연산 최대한 지연
- 메모리에 파일 캐싱 -> 항상 빠른 속도 보장 X
메모리 사용량 증가로 또 다른 성능이 악화될 수 있고 시스템 자체 캐시와 중복 가능성

Recommendation

- 디스크에 파일을 읽고 쓰는 것은 컴퓨터의 가장 느린 작업 중 하나. 파일 작업 수를 최소화
동일 파일에 대한 다중 작업 - 가능한 그룹화하여 일괄 처리
실제 데이터가 필요할 때까지 Dist I/O 연산 최대한 지연
- 메모리에 파일 캐싱 -> 항상 빠른 속도 보장 X
메모리 사용량 증가로 또 다른 성능이 악화될 수 있고 시스템 자체 캐시와 중복 가능성
- URL 객체 - 생성 비용이 크므로 `resourceValuesForKeys(_:)` 를 이용해 캐싱 후 재사용

Recommendation

- 디스크에 파일을 읽고 쓰는 것은 컴퓨터의 가장 느린 작업 중 하나. 파일 작업 수를 최소화
동일 파일에 대한 다중 작업 - 가능한 그룹화하여 일괄 처리
실제 데이터가 필요할 때까지 Dist I/O 연산 최대한 지연
- 메모리에 파일 캐싱 -> 항상 빠른 속도 보장 X
메모리 사용량 증가로 또 다른 성능이 악화될 수 있고 시스템 자체 캐시와 중복 가능성
- URL 객체 - 생성 비용이 크므로 `resourceValuesForKeys(_:)` 를 이용해 캐싱 후 재사용
- 캐싱 필요없는 데이터 - `Data.ReadingOptions uncached` 적용

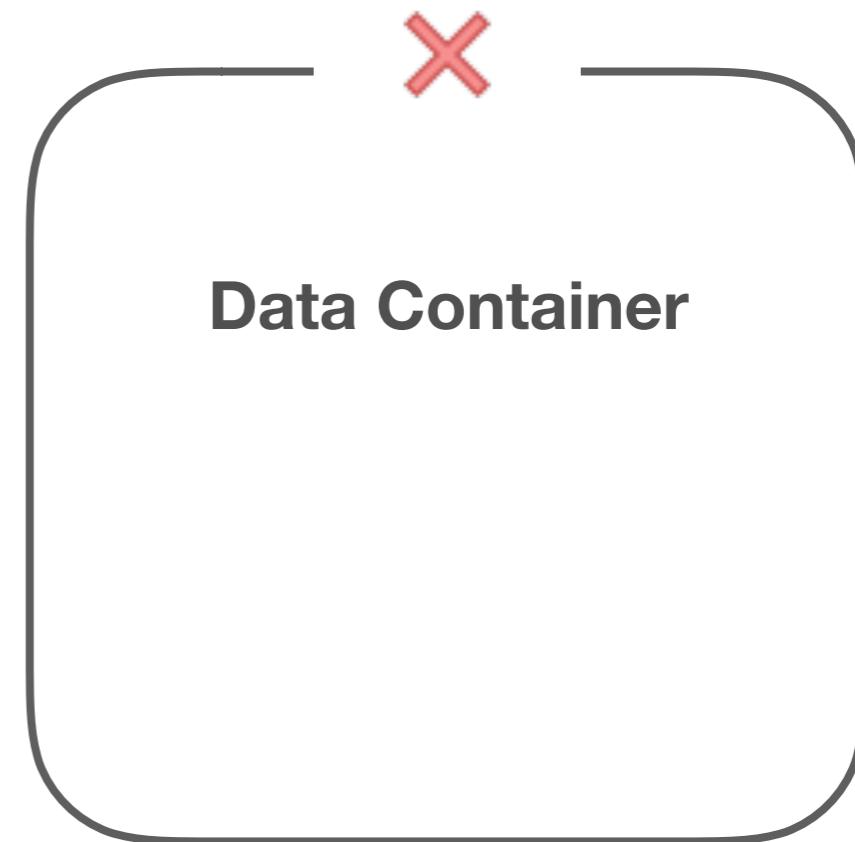
Recommendation

- 디스크에 파일을 읽고 쓰는 것은 컴퓨터의 가장 느린 작업 중 하나. 파일 작업 수를 최소화
 - 동일 파일에 대한 다중 작업 - 가능한 그룹화하여 일괄 처리
 - 실제 데이터가 필요할 때까지 Dist I/O 연산 최대한 지연
 - 메모리에 파일 캐싱 -> 항상 빠른 속도 보장 X
 - 메모리 사용량 증가로 또 다른 성능이 악화될 수 있고 시스템 자체 캐시와 중복 가능성
 - URL 객체 - 생성 비용이 크므로 `resourceValuesForKeys(_:)` 를 이용해 캐싱 후 재사용
 - 캐싱 필요없는 데이터 - `Data.ReadingOptions uncached` 적용
 - 파일 접근 시, String 대신 URL 이용 권장
- e.g.) `func write(to url: URL, atomically useAuxiliaryFile: Bool, encoding enc: UInt) throws`
- `func write(toFile path: String, atomically useAuxiliaryFile: Bool, encoding enc: UInt) throws`

Summary

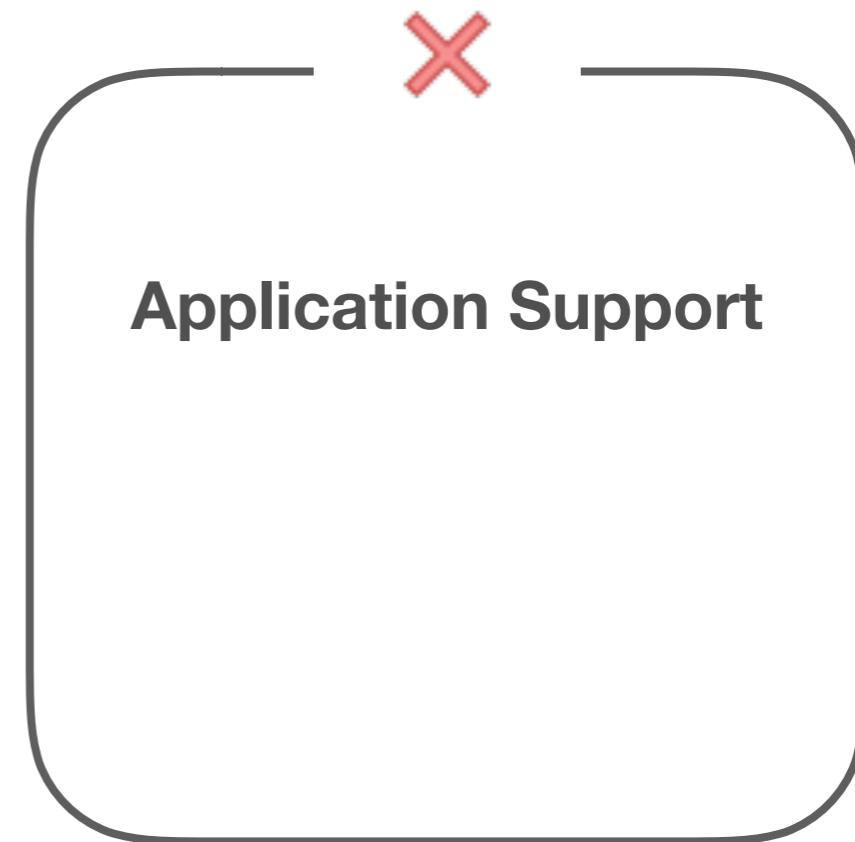
What to use

- 컴파일 타임 파일 생성 여부



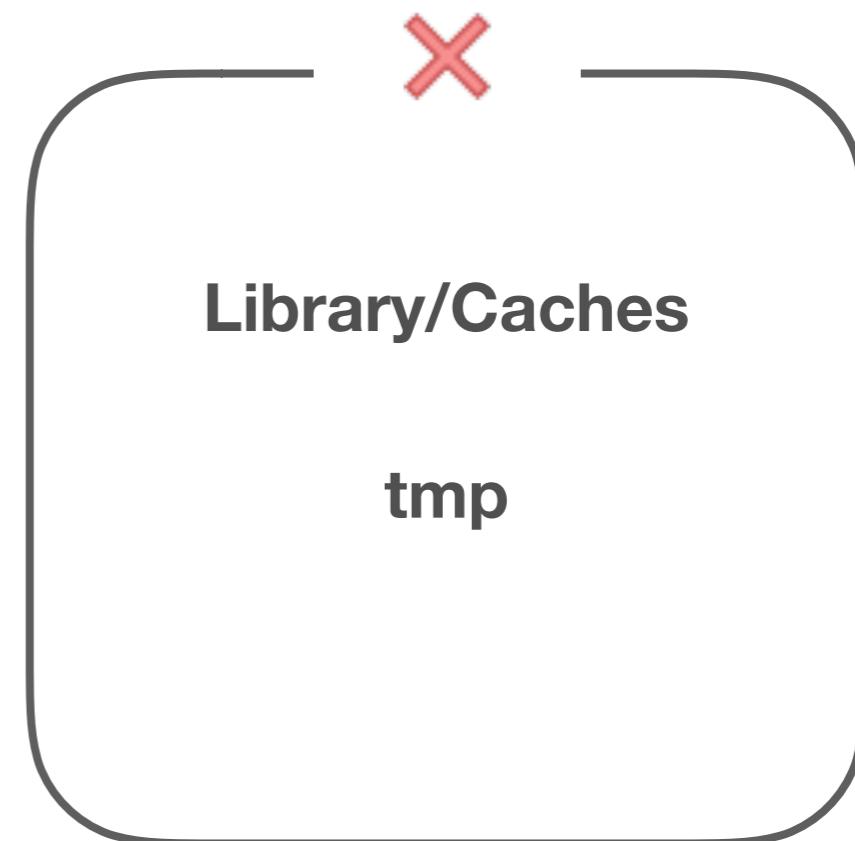
What to use

- 컴파일 타임 파일 생성 여부
- 유저 접근 가능 여부



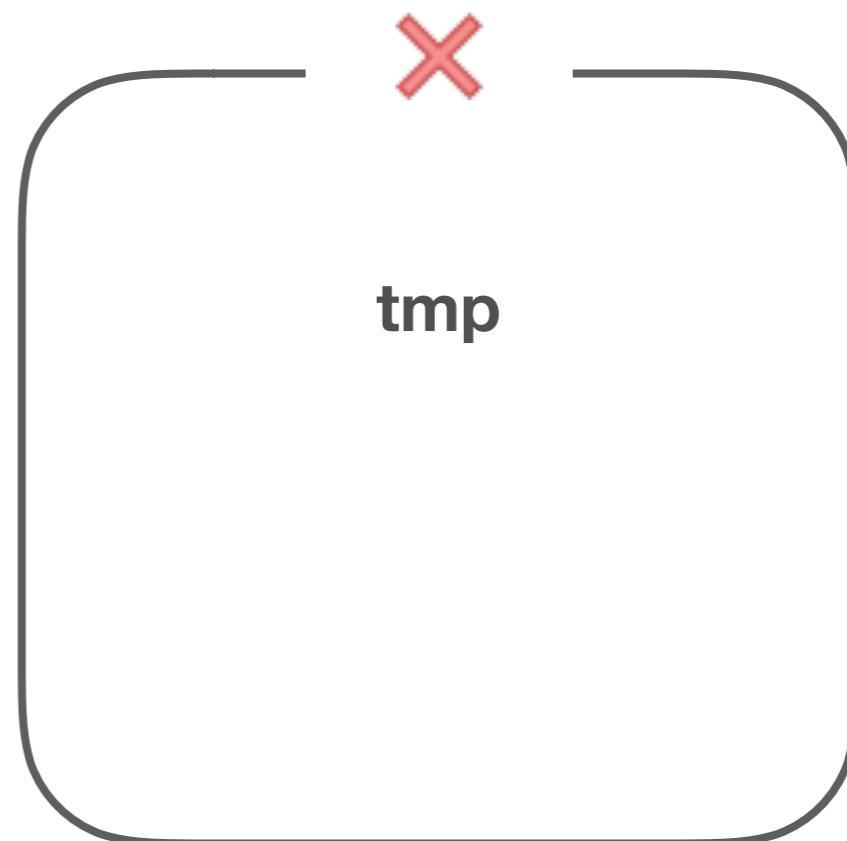
What to use

- 컴파일 타임 파일 생성 여부
- 유저 접근 가능 여부
- 지속성 및 백업 여부



What to use

- 컴파일 타임 파일 생성 여부
- 유저 접근 가능 여부
- 지속성 및 백업 여부
- 앱 런칭 간 데이터 유지 필요 여부



End