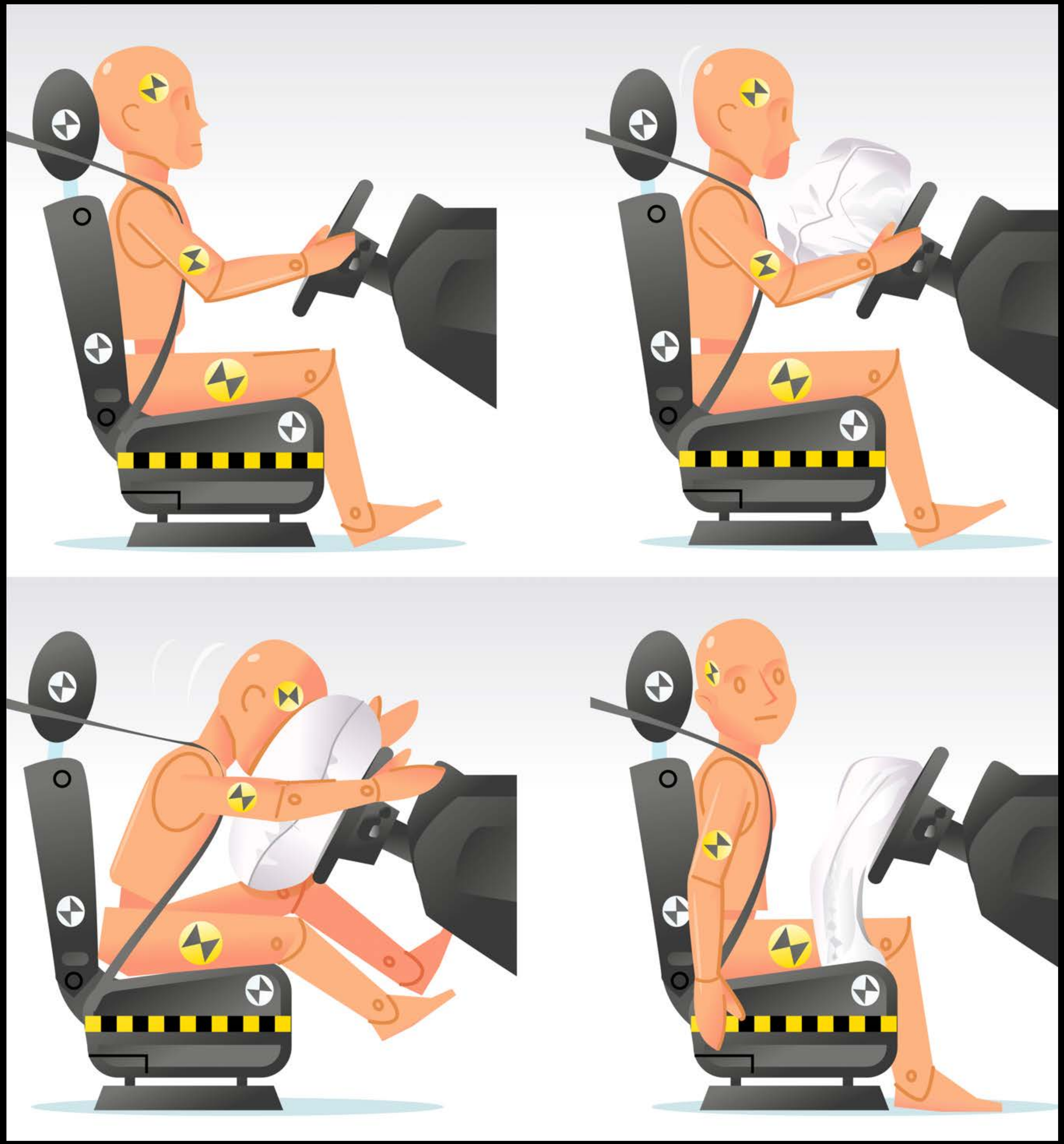
A Practical Guide to the App Sandbox

Session 710

Ivan Krstić
Core OS Honey Badger

Modern Car Safety

- Mandatory standardized crash testing performed by the government
- Traction control, blind spot warnings, lane-drift alerts
- But, damage containment
- When all else fails, there are seat belts and airbags



Traditional Desktop Security

- Defender must protect everything at all times, attacker must breach one protection at any time
- Emphasis on damage prevention (ASLR, NX, antivirus), not containment
- One thing goes wrong, game over
- No seatbelt and airbag for the computer

GAME OVER

The Unfortunate Assumption

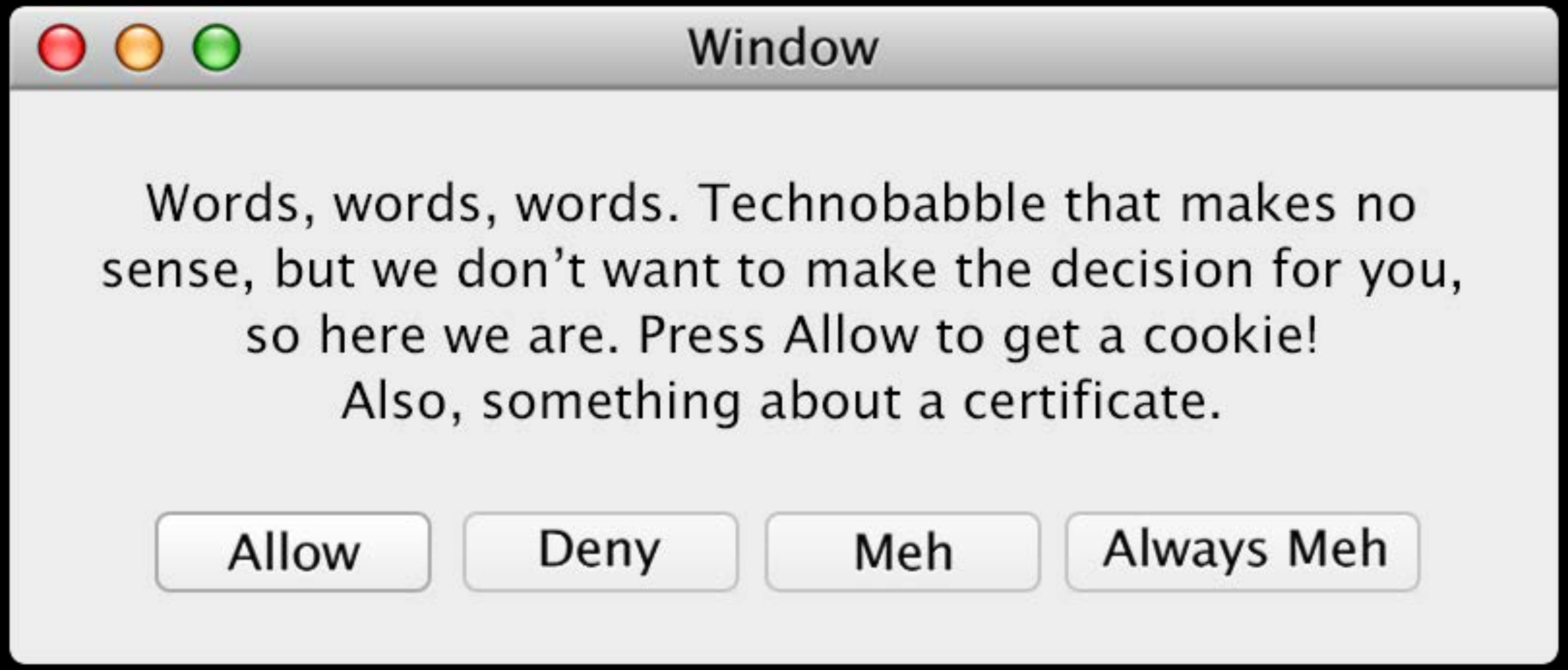
- All programs should execute with the full privileges of the executing user
    - Or, security is a barrier between different users, not different programs
- But most modern computer devices are single-user systems
- Not every app should have access to the most sensitive data
    - Apps should only have access to the resources they need

An Unfortunate Example

- The unfortunate assumption does not work
- Compromising any app must not grant access to all user data

Security UI Does Not Work

- Security dialogs are mysterious and opaque; riddles wrapped inside enigmas
- Clicking "Permit" or "Allow" maximizes the likelihood of getting work done
- "If you're explaining, you're losing"
- Pavlovian conditioning to ignore security



Window

Words, words, words. Technobabble that makes no sense, but we don't want to make the decision for you, so here we are. Press Allow to get a cookie! Also, something about a certificate.

Landscape Changes

- Many apps, many developers
- Computers are always on a network
- Easier than ever to find and run new software
- Security challenge: Isolate data between programs

Software Reality

- Complex systems will always have vulnerabilities
  - Complexity is never decreasing
- Single buffer overflow can ruin your user's day
- Frameworks and libraries you don't control
  - Every `WebView` instance: Millions of lines of code and a full-featured JavaScript engine
- No limit on exploit damage

App Sandbox

App Sandbox

- Introduced in OS X Lion
- More secure applications
- Drive security policy by user intent
- Contain exploit damage
- Reduce ability for a compromised or misbehaving application to steal, corrupt, or destroy user data

Key Concepts

- Developer expresses what an app is supposed to be able to do
- Each app runs in its own container
- User controls access to documents
    - Special cases (e.g., recent items, drag and drop) work automatically

Key Components

Key Components

- Entitlements
- Containers
- PowerBox
- XPC Services

Entitlements

- What apps can do is determined by the developer-specified entitlements in the code signature
- Just a property list, editable in Xcode
- Simple, easy to understand

Entitlements

- User-selected files, Downloads folder
- Personal information
    - Address book, calendars, location
- Assets: Music, movies, pictures
- Network client, server
- Devices
    - Camera, microphone, printing, USB, FireWire, Bluetooth, serial
- Application groups and scripting/automation targets

Key Components

Entitlements

Containers

Powerbox

XPC Services

Key Components

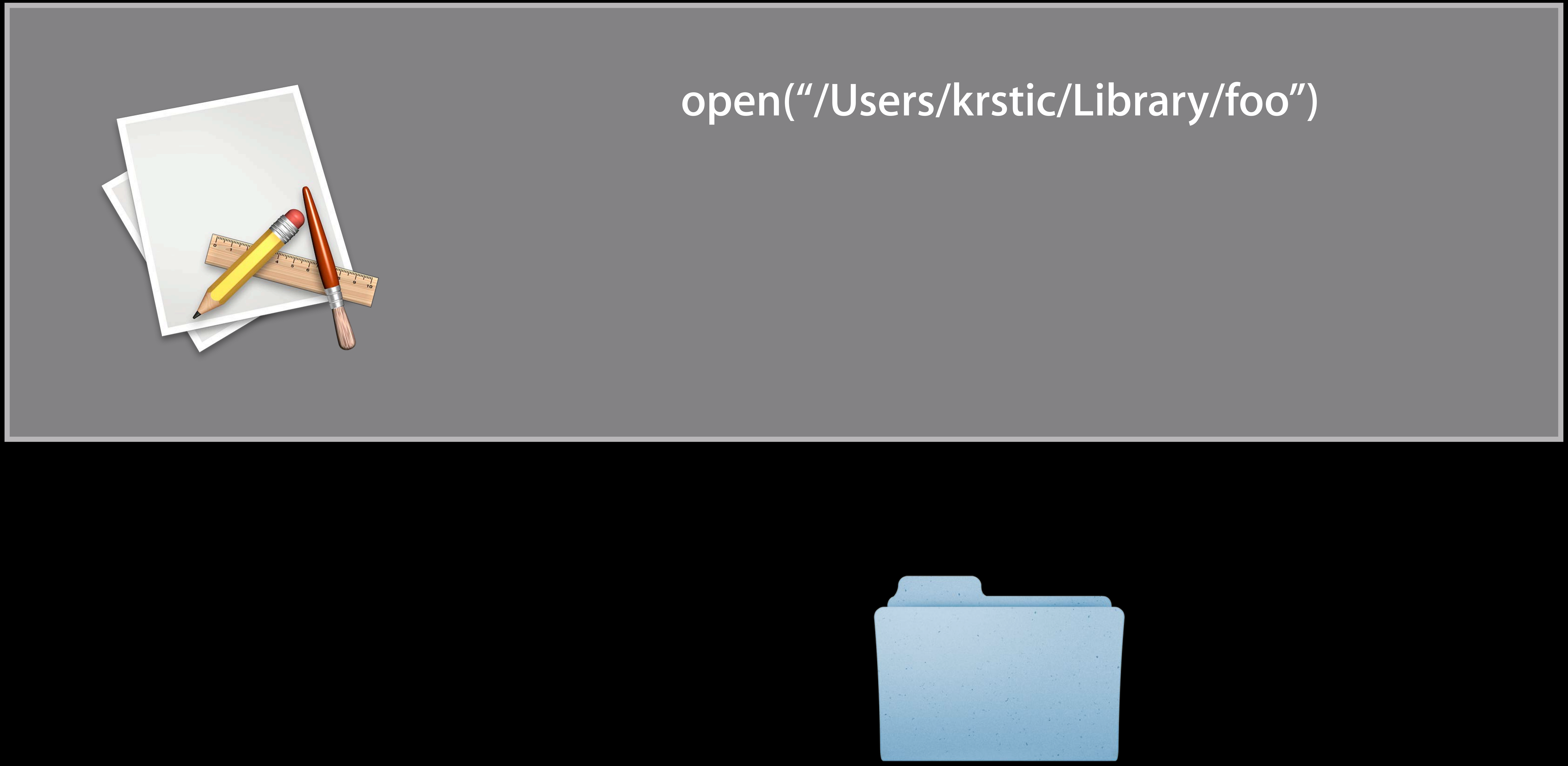
- Entitlements
- **Containers**
- Powerbox
- XPC Services

```
HOME=~/Library/Containers/App/
CFFIXED_USER_HOME=~/Library/Containers/App/
```

```
HOME=~/Library/Containers/App/
CFFIXED_USER_HOME=~/Library/Containers/App/
```

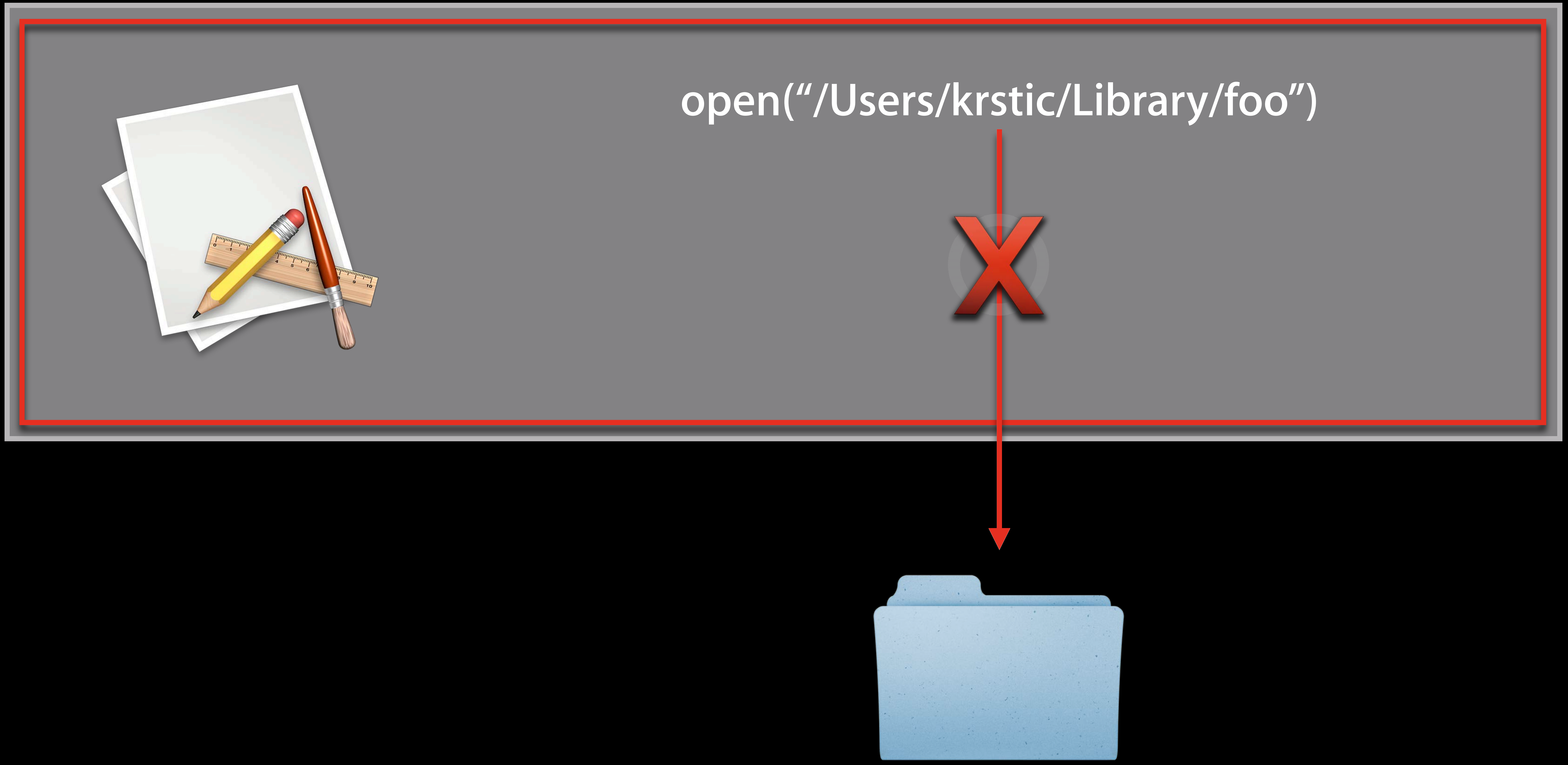
```
HOME=~/Library/Containers/App/
CFFIXED_USER_HOME=~/Library/Containers/App/
```

```
HOME=~/Library/Containers/App/
CFFIXED_USER_HOME=~/Library/Containers/App/
```

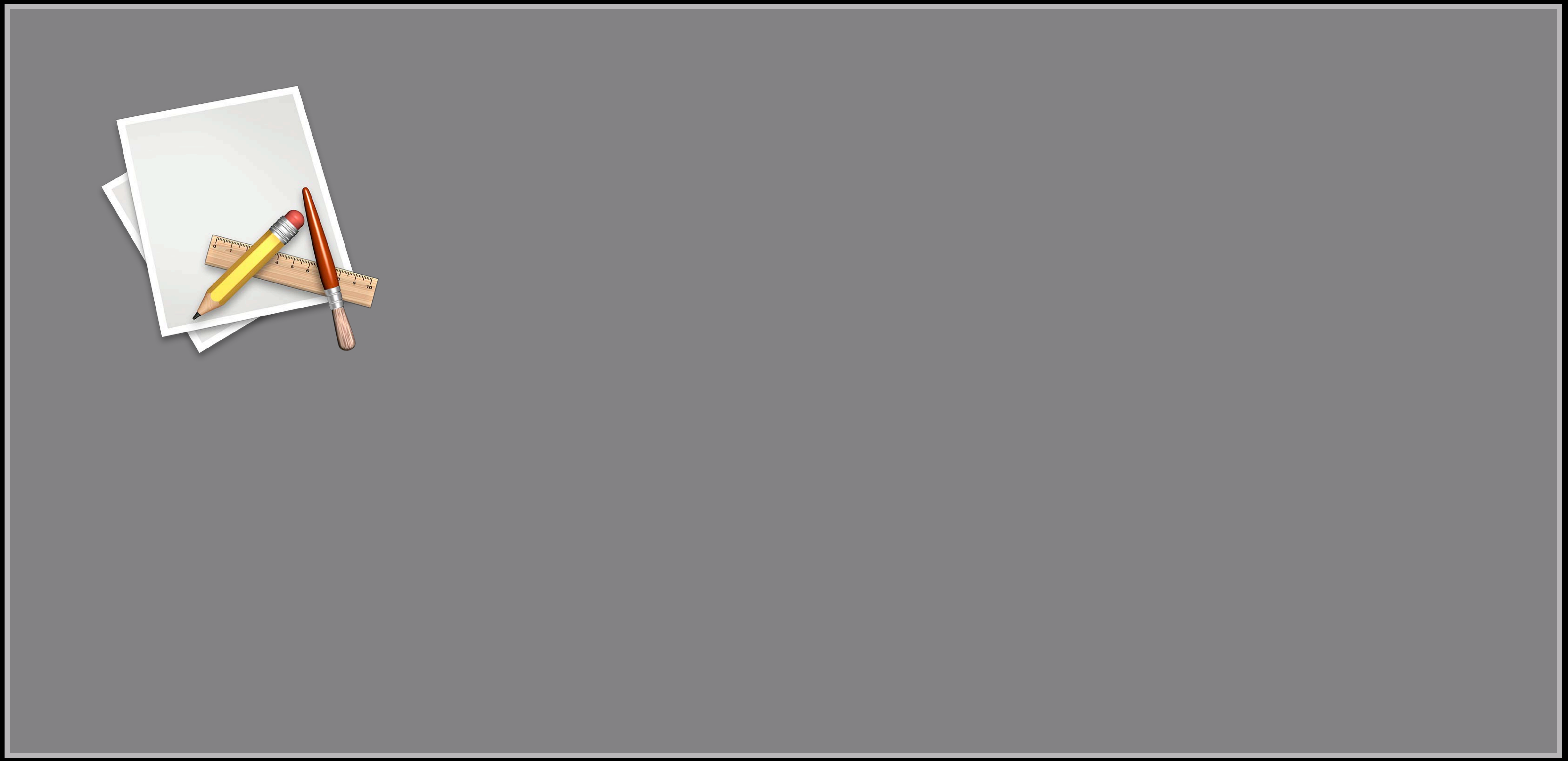


```
open("/Users/krstic/Library/foo")
```

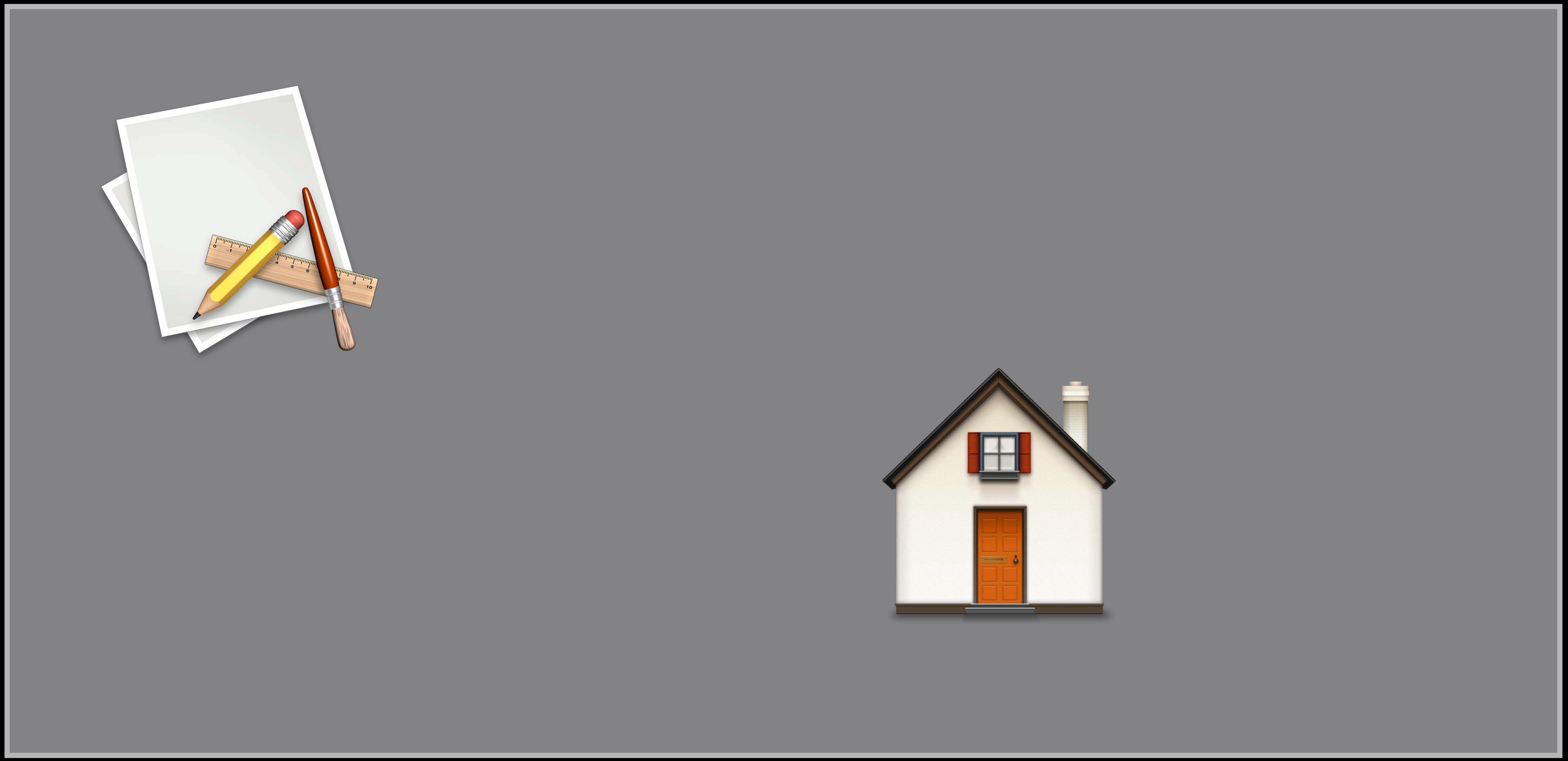
```
HOME=~/Library/Containers/App/
CFFIXED_USER_HOME=~/Library/Containers/App/
```



```
HOME=~/Library/Containers/App/
CFFIXED_USER_HOME=~/Library/Containers/App/
```



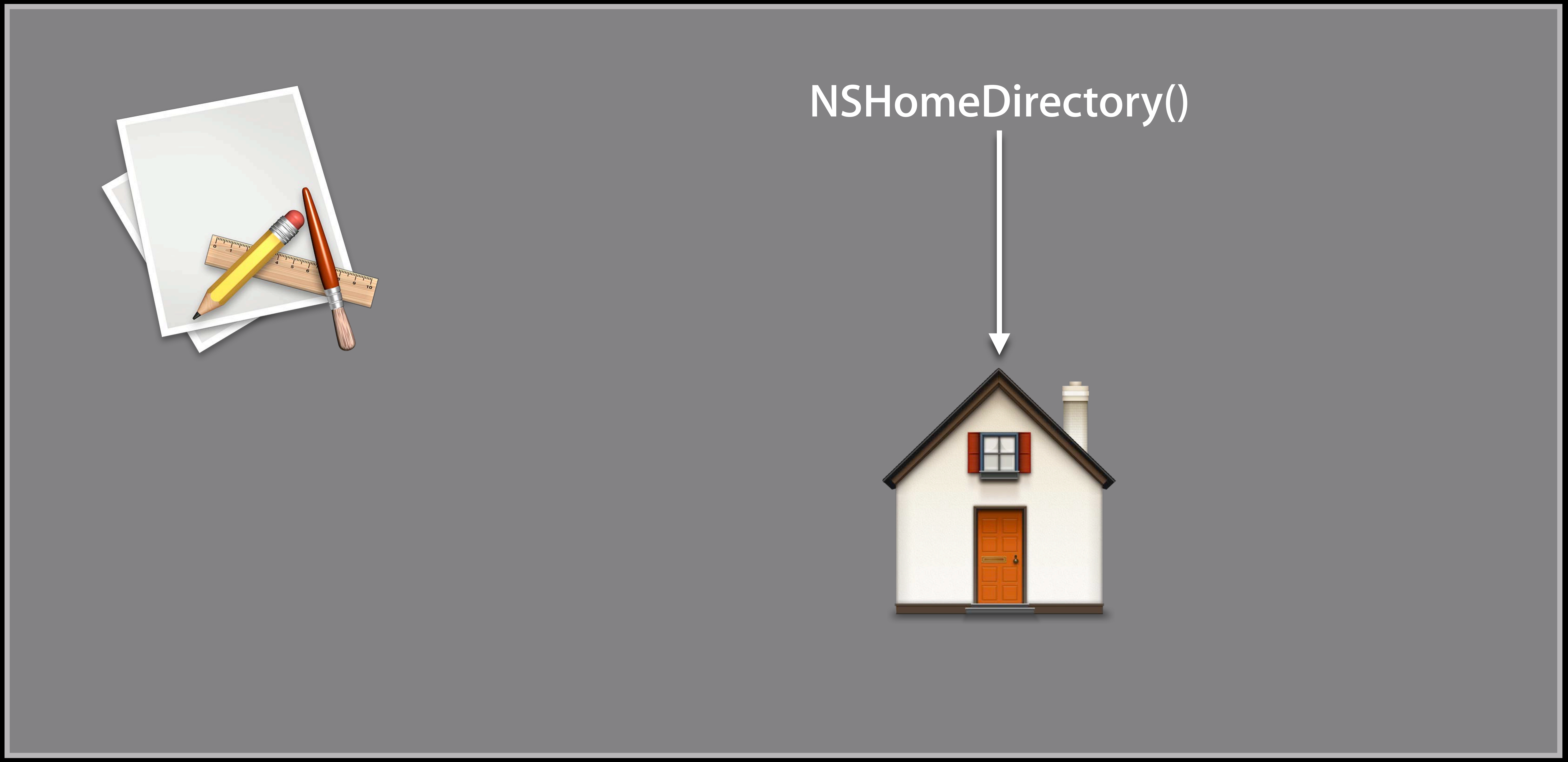
```
HOME=~/Library/Containers/App/
CFFIXED_USER_HOME=~/Library/Containers/App/
```



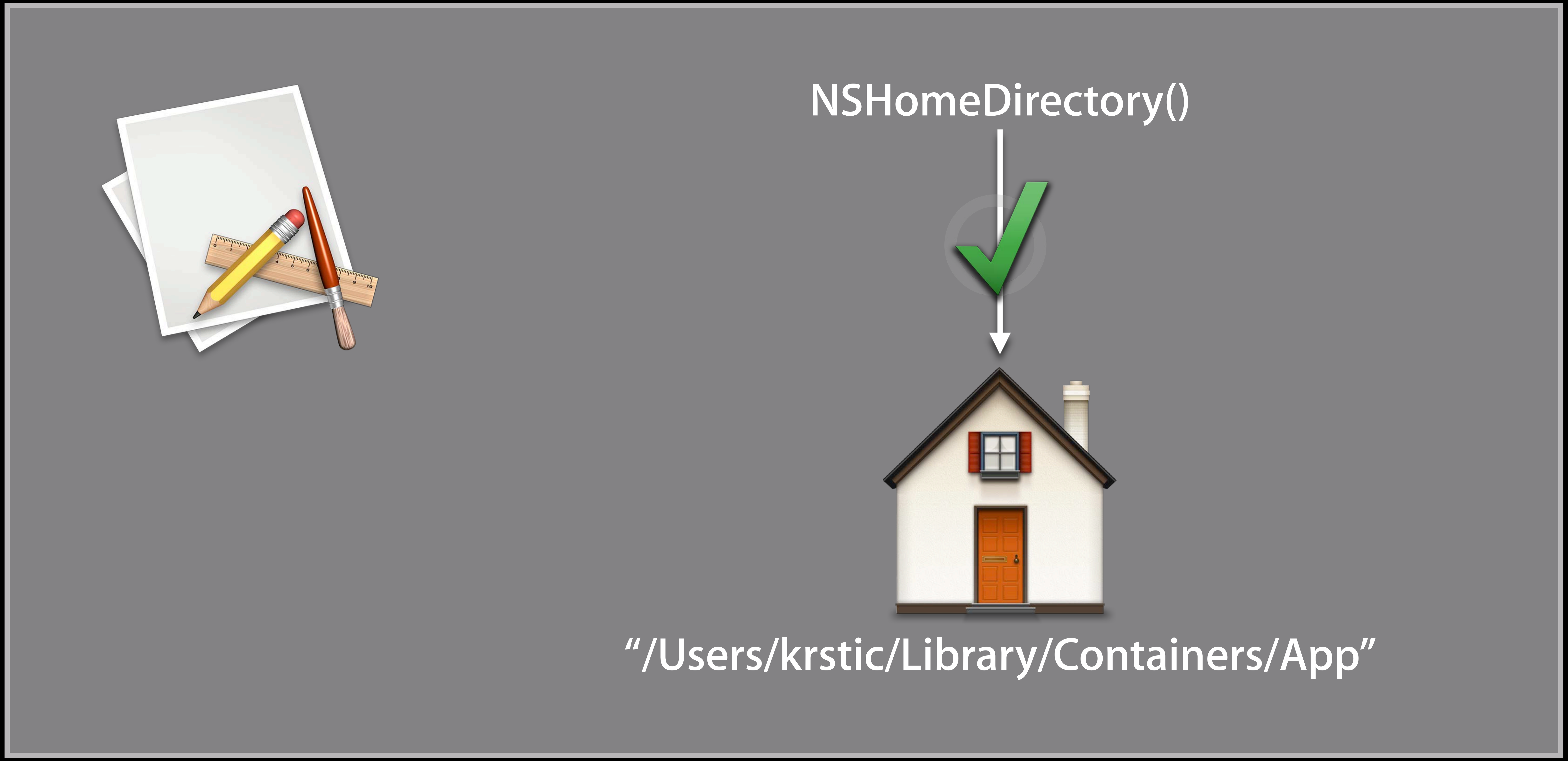
```
HOME=~/Library/Containers/App/
CFFIXED_USER_HOME=~/Library/Containers/App/
```

NSHomeDirectory()

```
HOME=~/Library/Containers/App/
CFFIXED_USER_HOME=~/Library/Containers/App/
```



```
HOME=~/Library/Containers/App/
CFFIXED_USER_HOME=~/Library/Containers/App/
```



Key Components

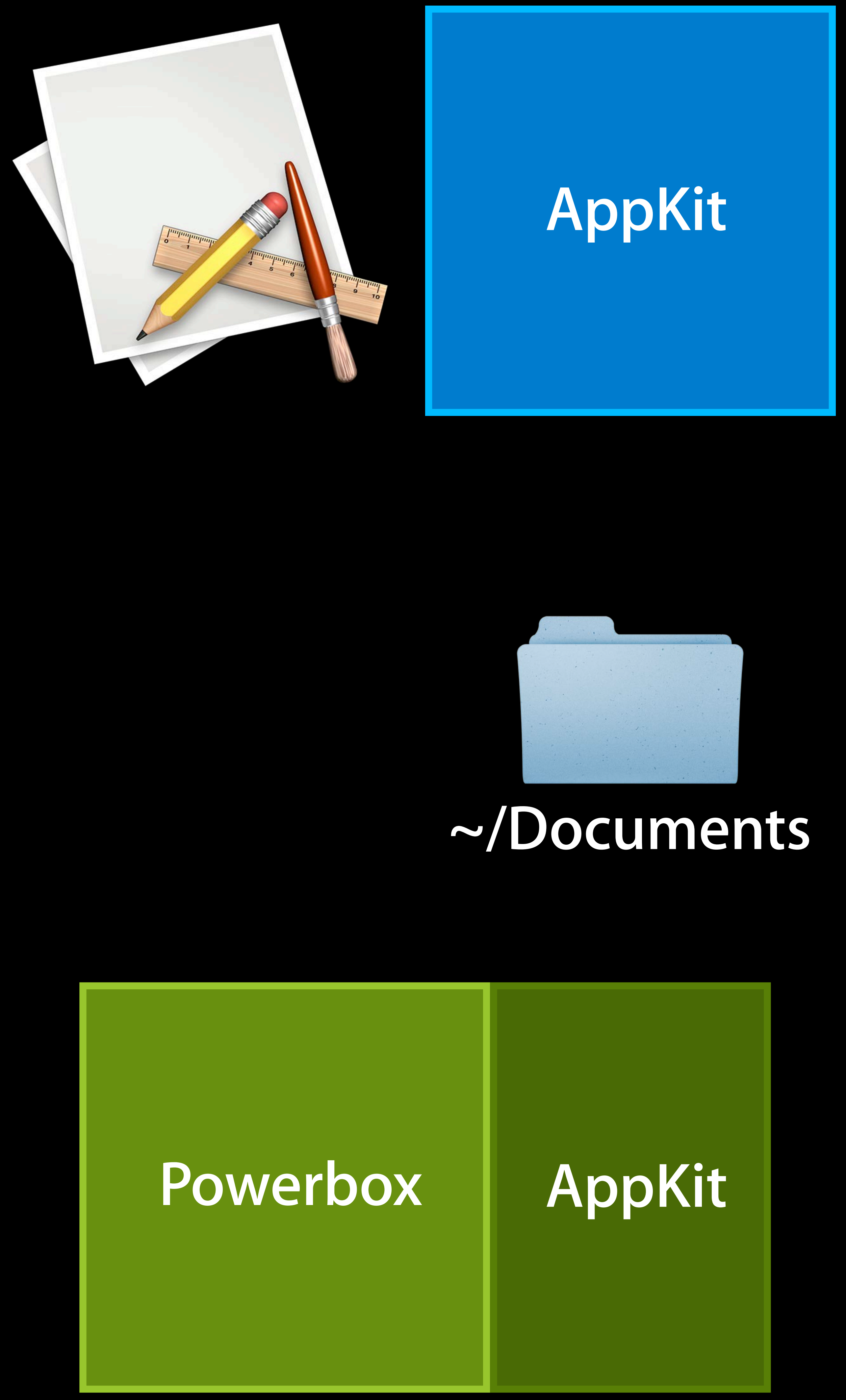
- Entitlements
- Containers
- Powerbox
- XPC Services

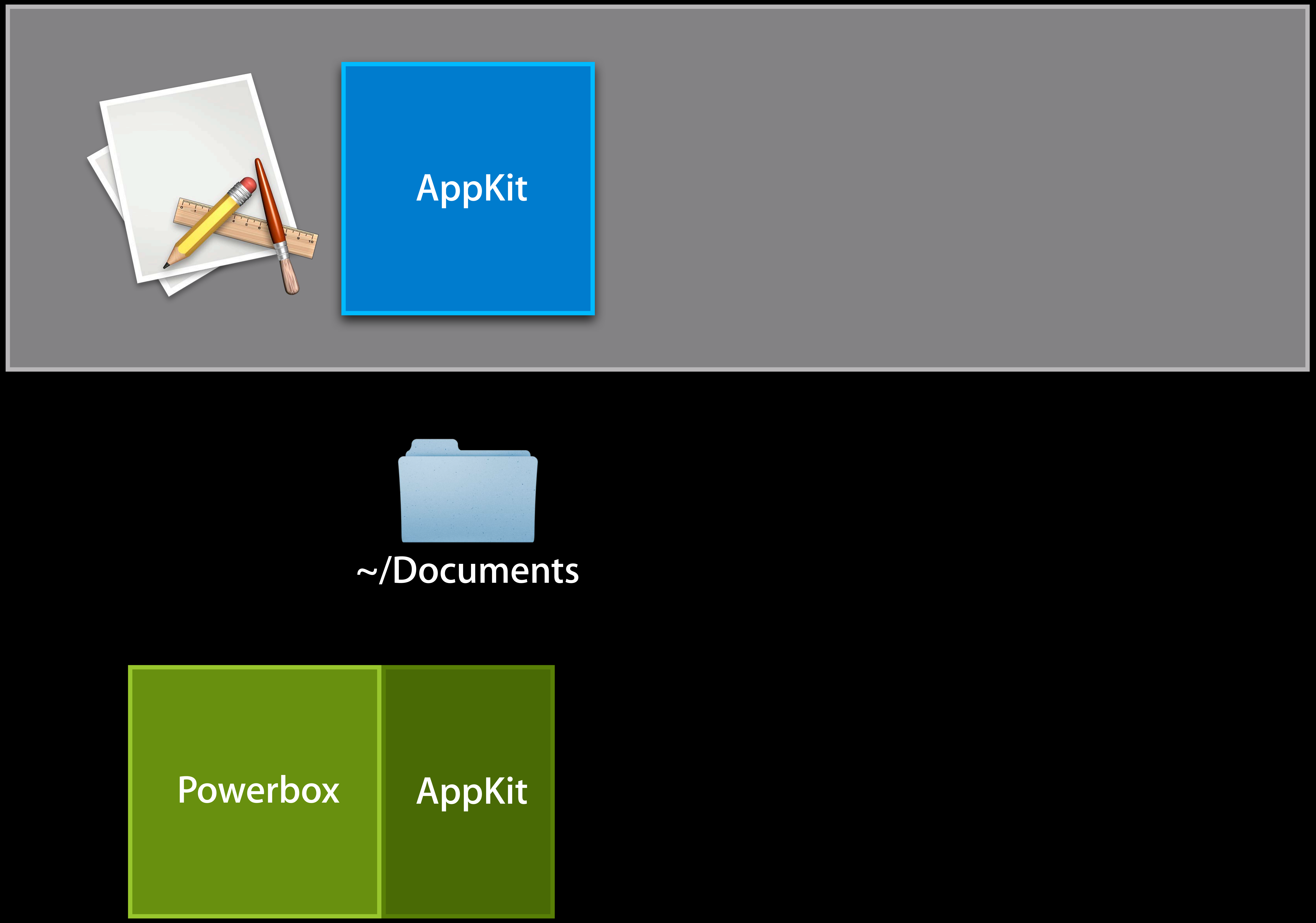
Key Components

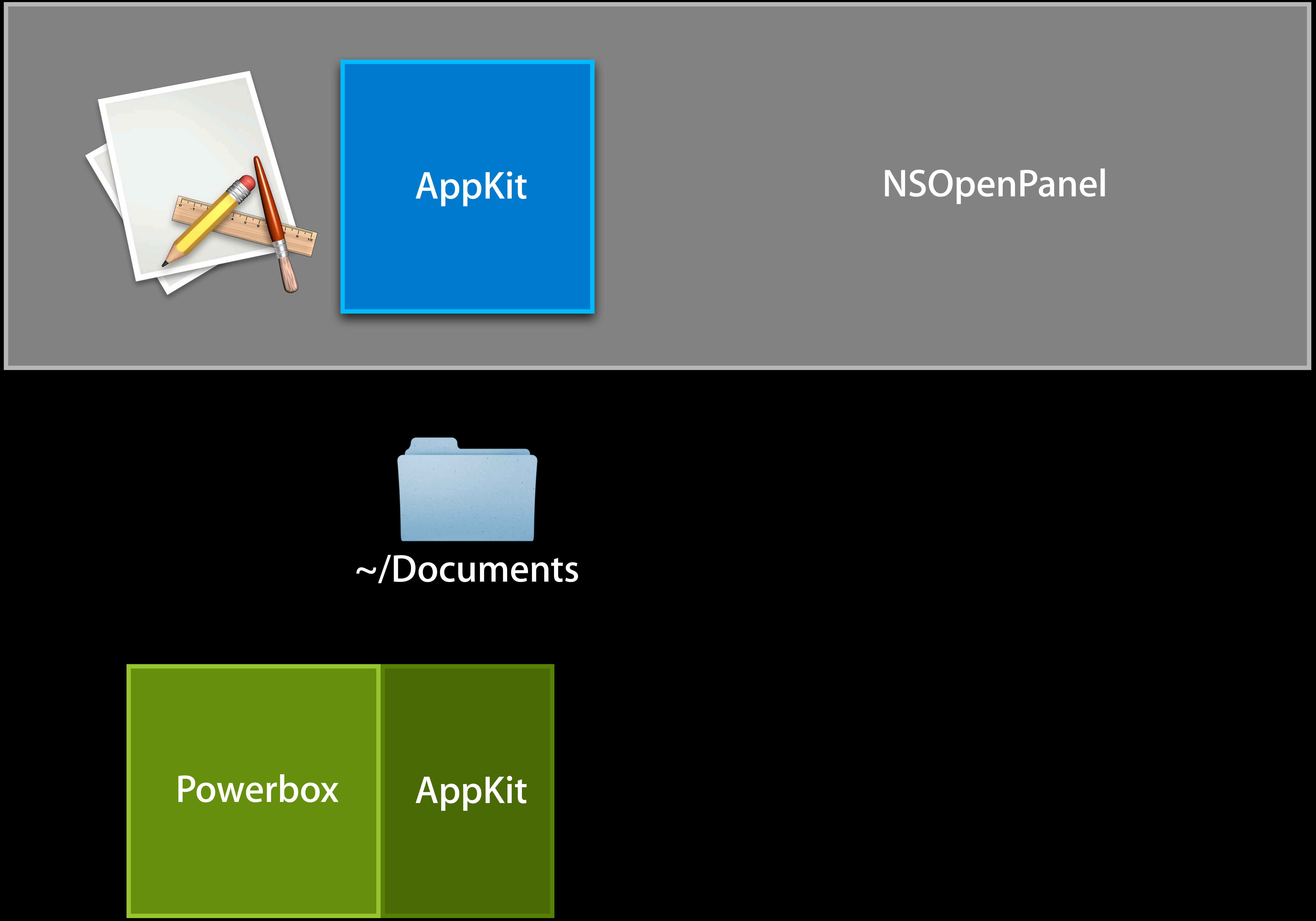
- Entitlements
- Containers
- **Powerbox**
- XPC Services

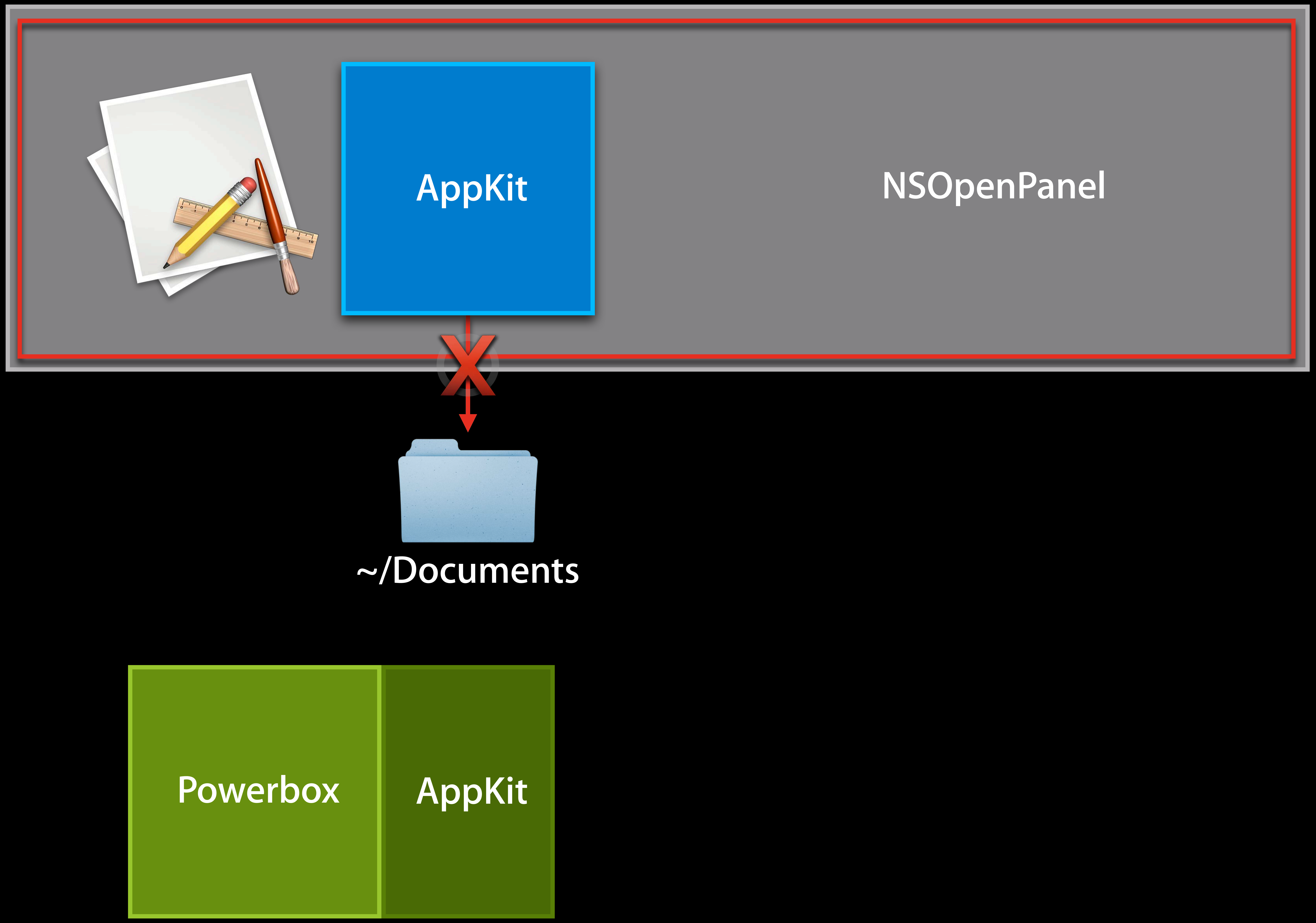
Powerbox

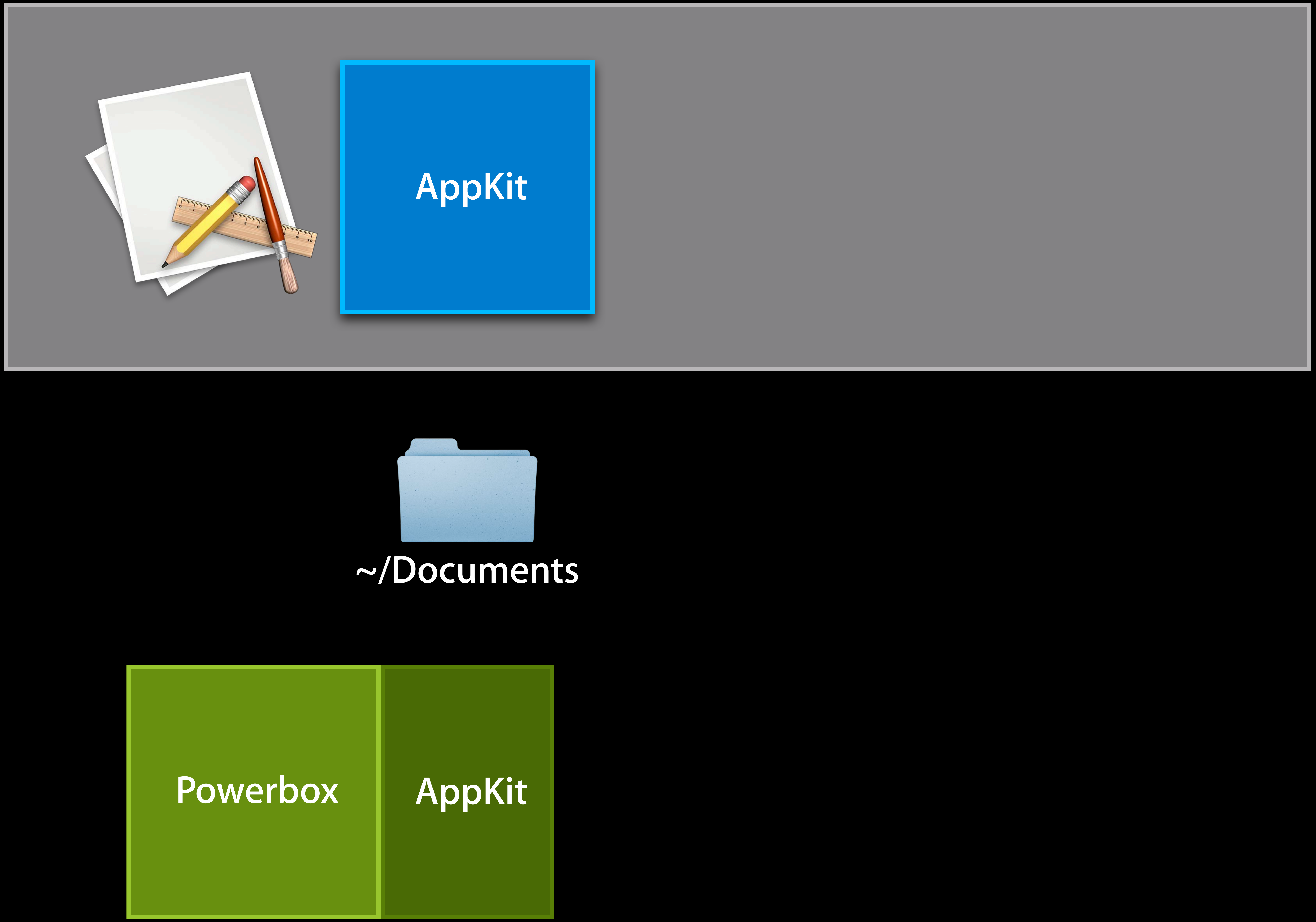
- Cocoa NSOpenPanel/NSSavePanel
- Trusted mediator process
- Clear declaration of user intent
  - Drives security policy
  - Sandboxed apps cannot synthesize user input events

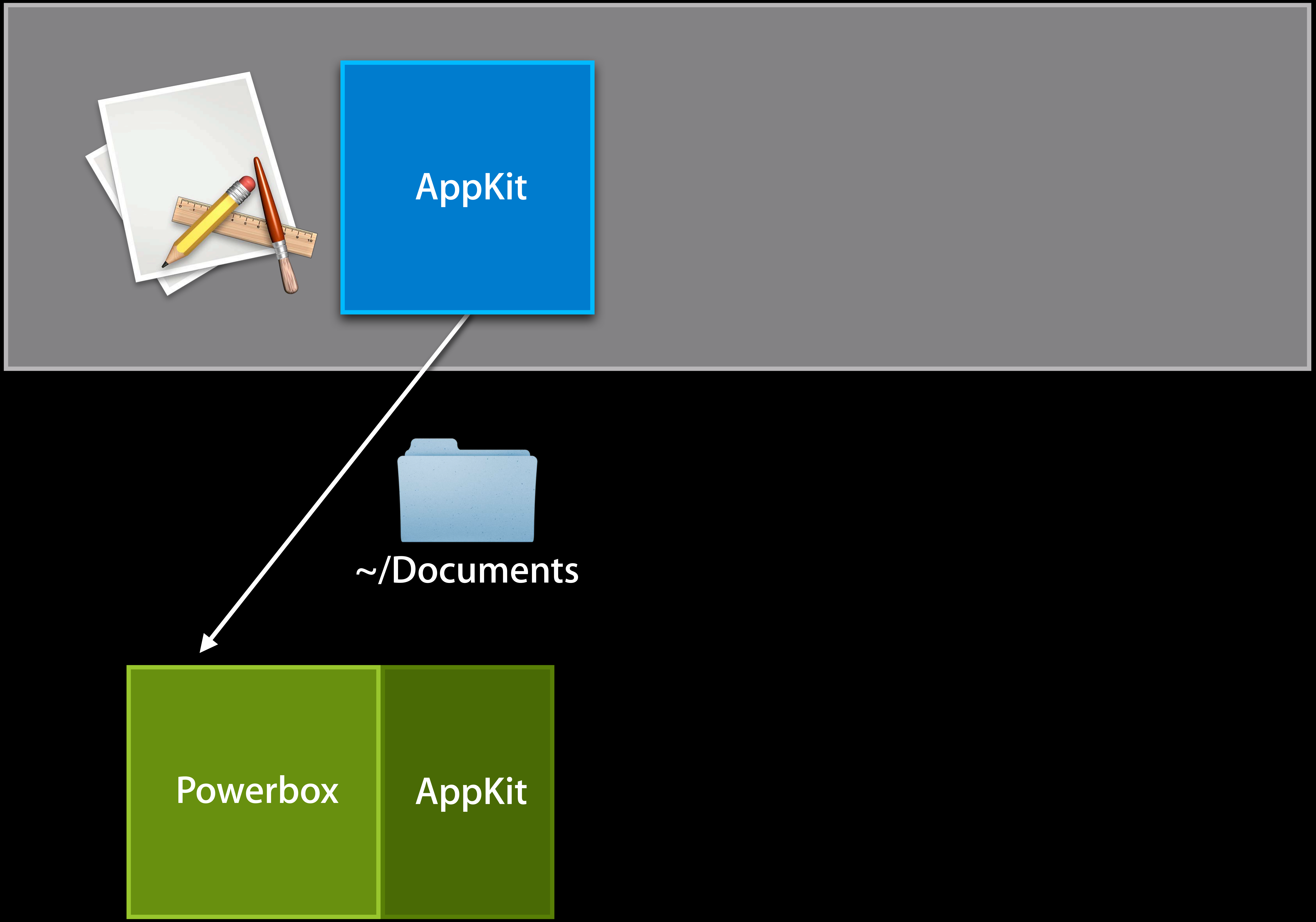

AppKit
~/Documents
Powerbox
AppKit

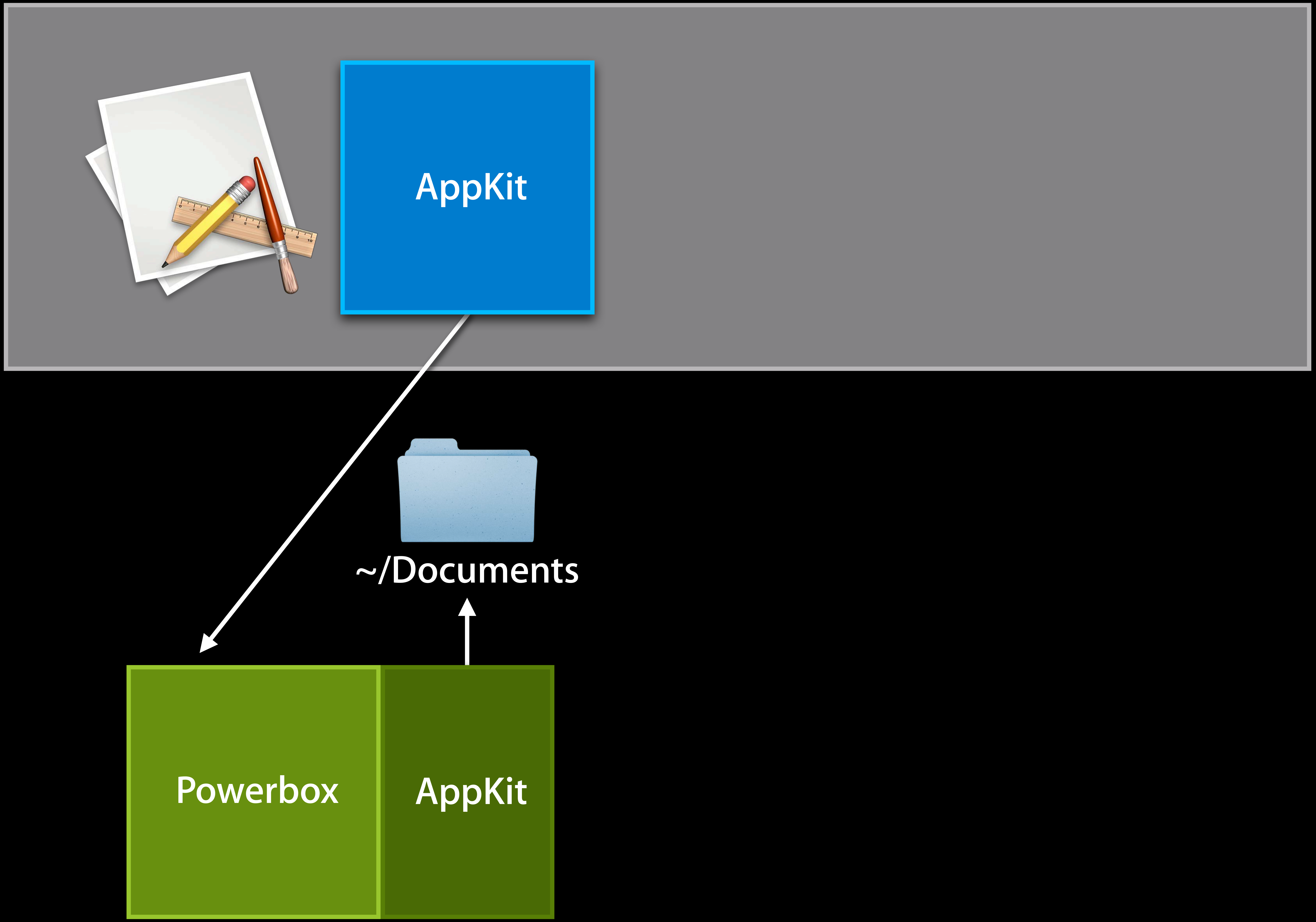

AppKit
~/Documents
Powerbox
AppKit

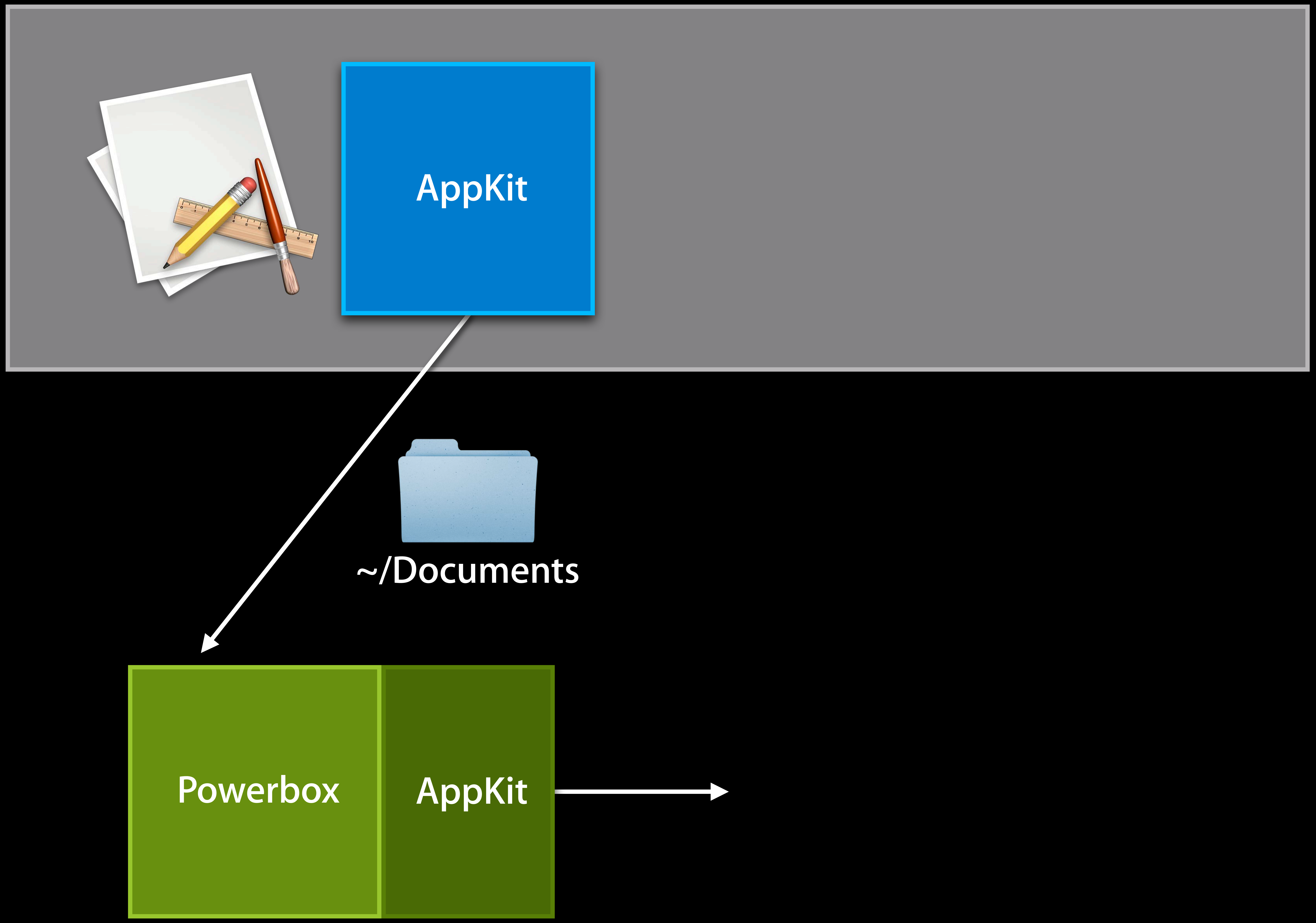

AppKit
NSOpenPanel
~/Documents
Powerbox
AppKit

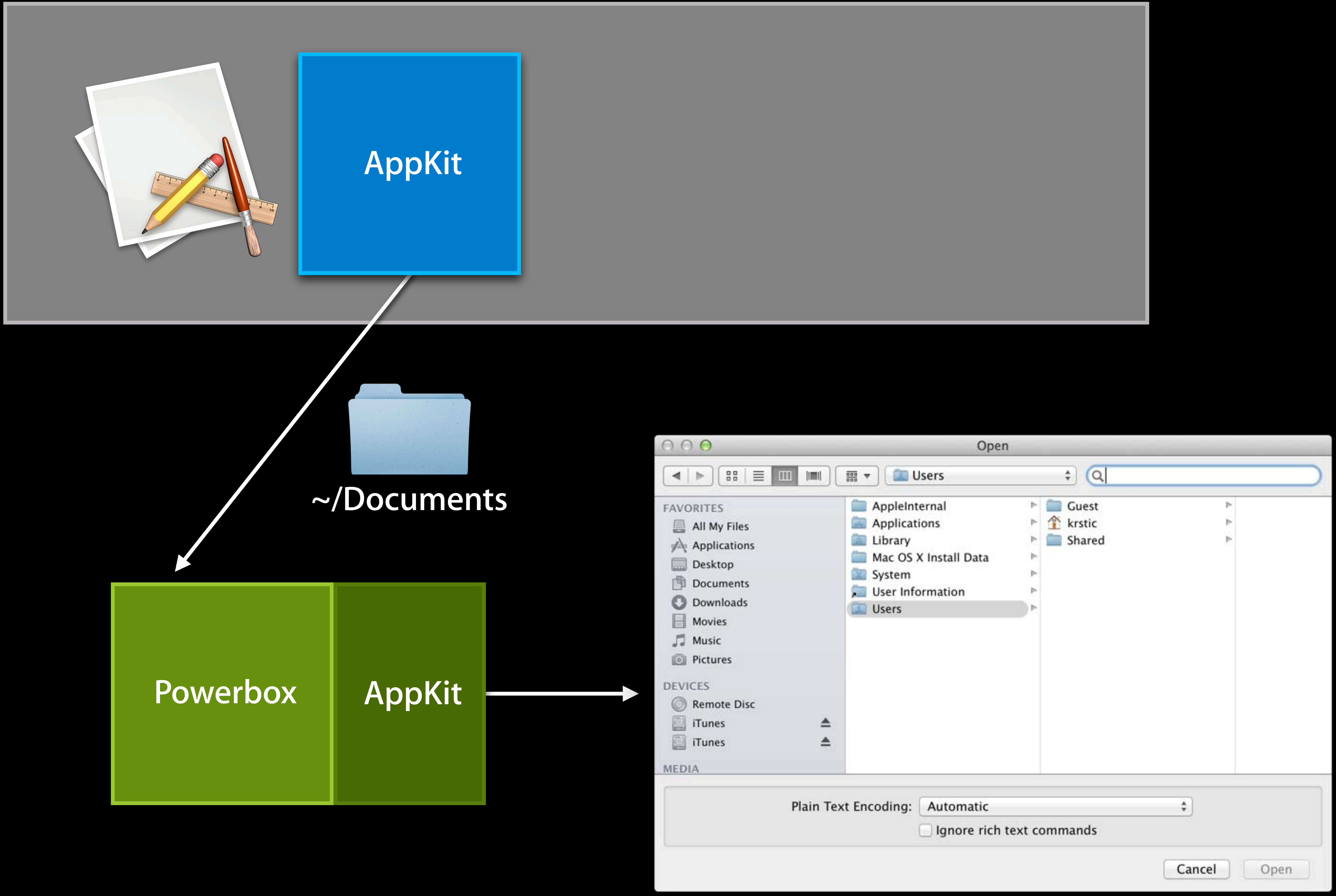

AppKit
NSOpenPanel
~/Documents
Powerbox
AppKit


AppKit
~/Documents
Powerbox
AppKit


AppKit
~/Documents
Powerbox
AppKit

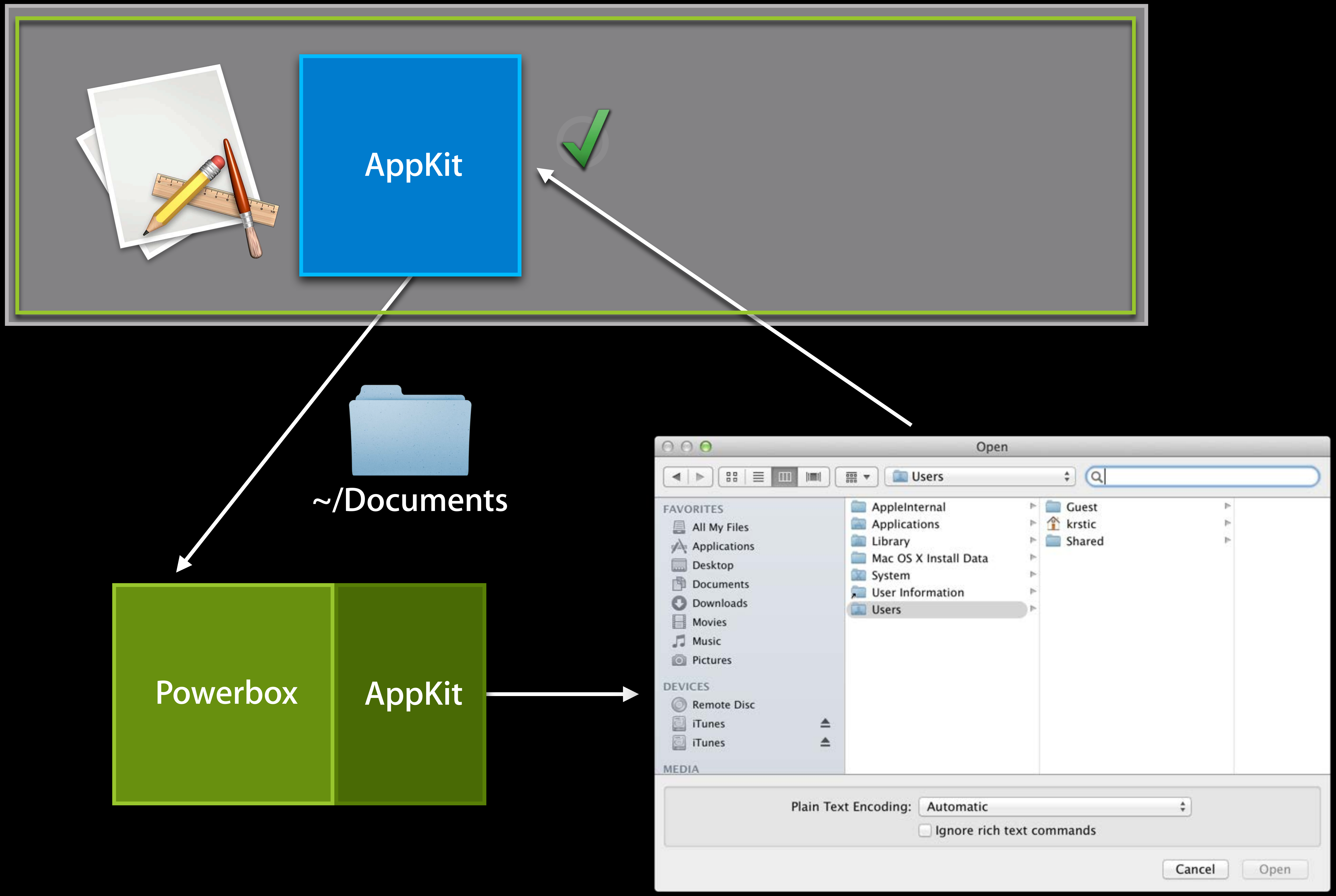

AppKit
~/Documents
Powerbox
AppKit


AppKit
~/Documents
Powerbox
AppKit



Plain Text Encoding: Automatic

Ignore rich text commands



Plain Text Encoding: Automatic

☐ Ignore rich text commands

Key Components

- Entitlements
- Containers
- Powerbox
- XPC Services

Key Components

Entitlements

Containers

Powerbox

XPC Services

XPC Services

- Very easy app and framework privilege separation
- Services have their own entitlements
- No fork/exec, process lifecycle managed by XPC
- Only available to the containing app

Putting It All Together

TextEdit

TextEdit

Process

- Prepare entitlements
- Code sign program
- Run and verify App Sandbox status
- Look for violations

Demo

TextEdit

Exploitation

- The attacker only has access to documents that the user opened during this TextEdit run
- No ability to access or modify other apps or documents
- Need multiple vulnerabilities for a successful exploit

Advanced App Sandbox

Security-Scoped Bookmarks

Security-Scoped Bookmarks

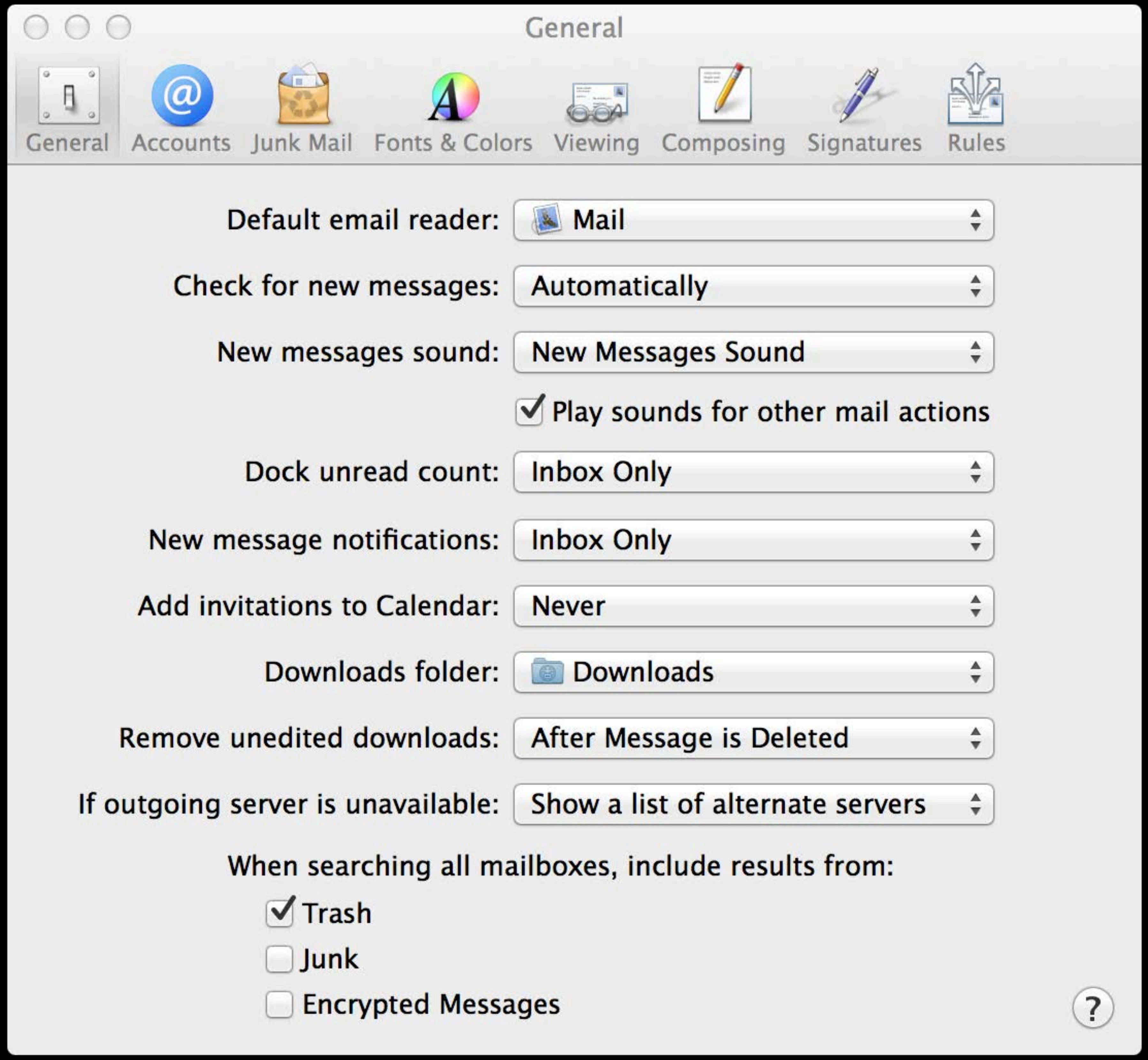
- Preserve access to user-chosen files and folders across system reboot
- Per-user app configuration—Input and output folders, commonly accessed files
- Document formats that contain references to files

Security-Scoped Bookmarks

- App scope

  `com.apple.security.files.user-selected.read-{write,only}`

  - Locked to the app and user that created them



Default email reader:	Mail
Check for new messages:	Automatically
New messages sound:	New Messages Sound
	✓ Play sounds for other mail actions
Dock unread count:	Inbox Only
New message notifications:	Inbox Only
Add invitations to Calendar:	Never
Downloads folder:	Downloads
Remove unedited downloads:	After Message is Deleted
If outgoing server is unavailable:	Show a list of alternate servers

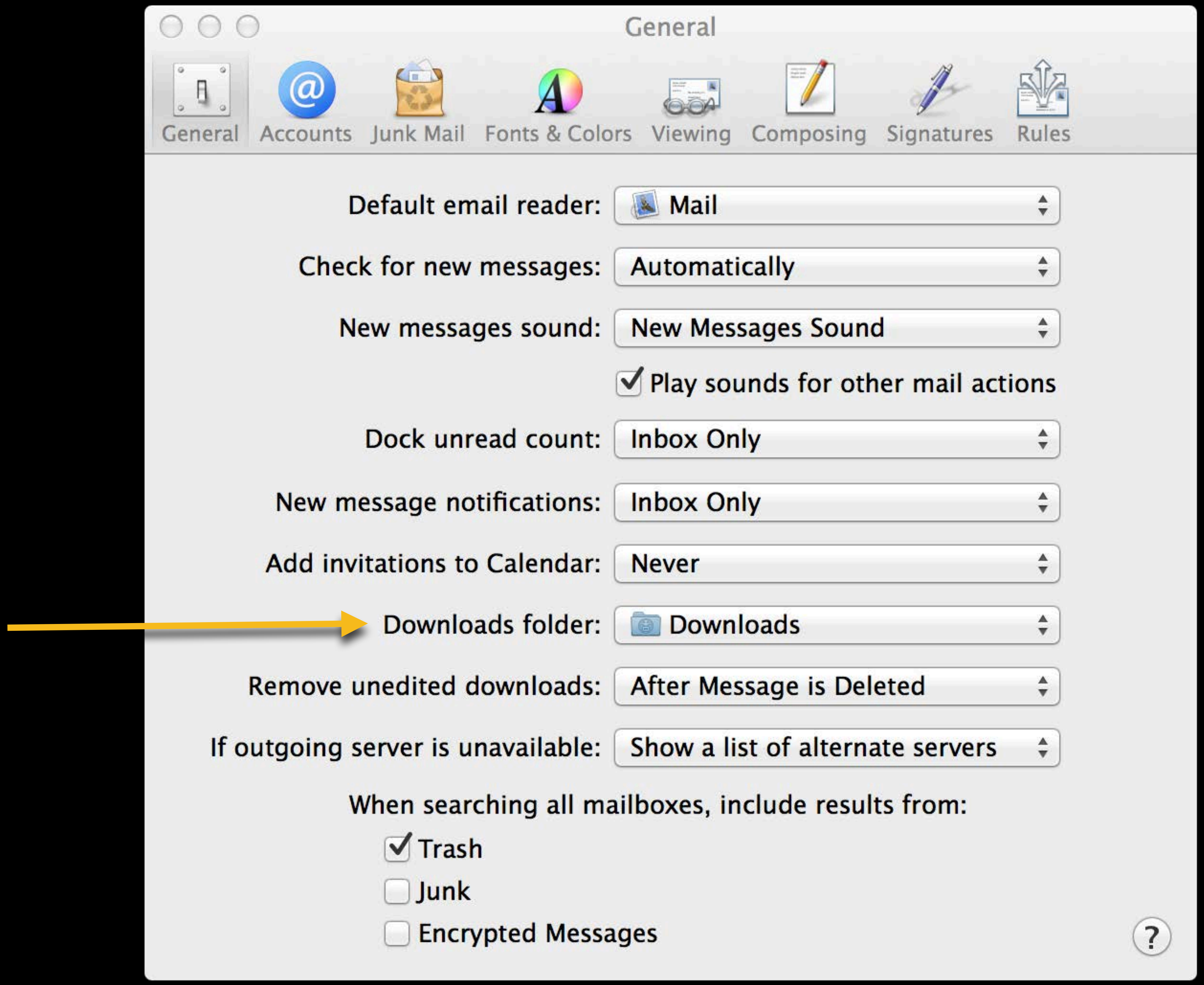
When searching all mailboxes, include results from:
- ✓ Trash
- ☐ Junk
- ☐ Encrypted Messages

Security-Scoped Bookmarks

- App scope

  `com.apple.security.files.user-selected.read-{write,only}`

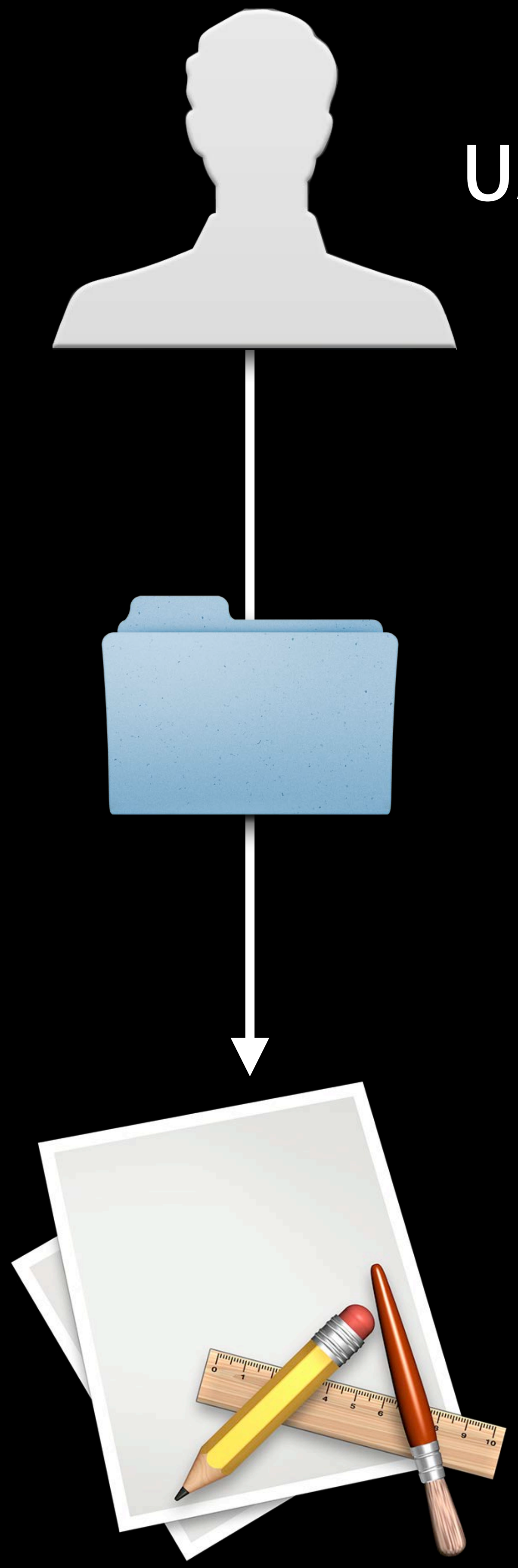
  - Locked to the app and user that created them

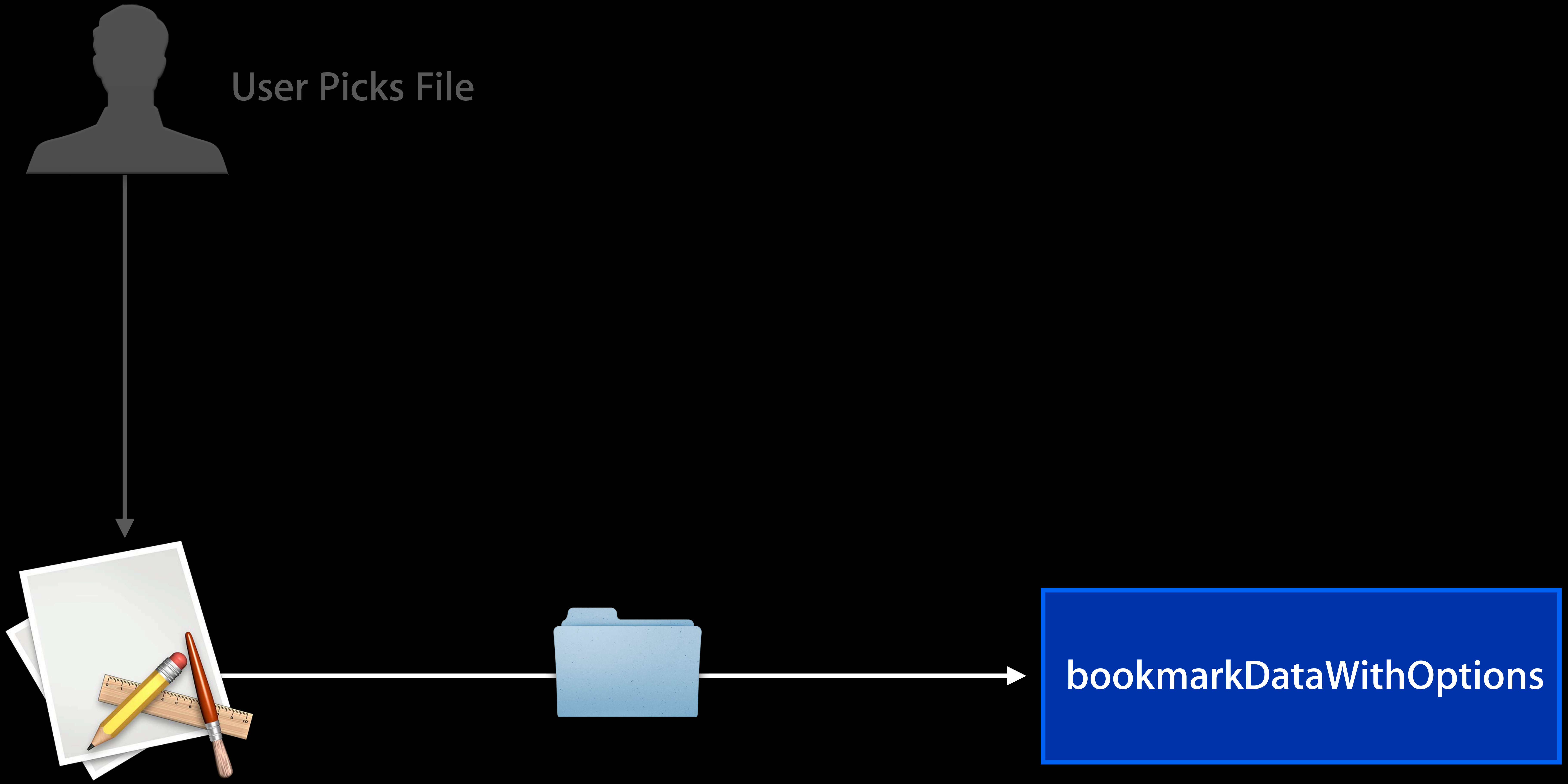


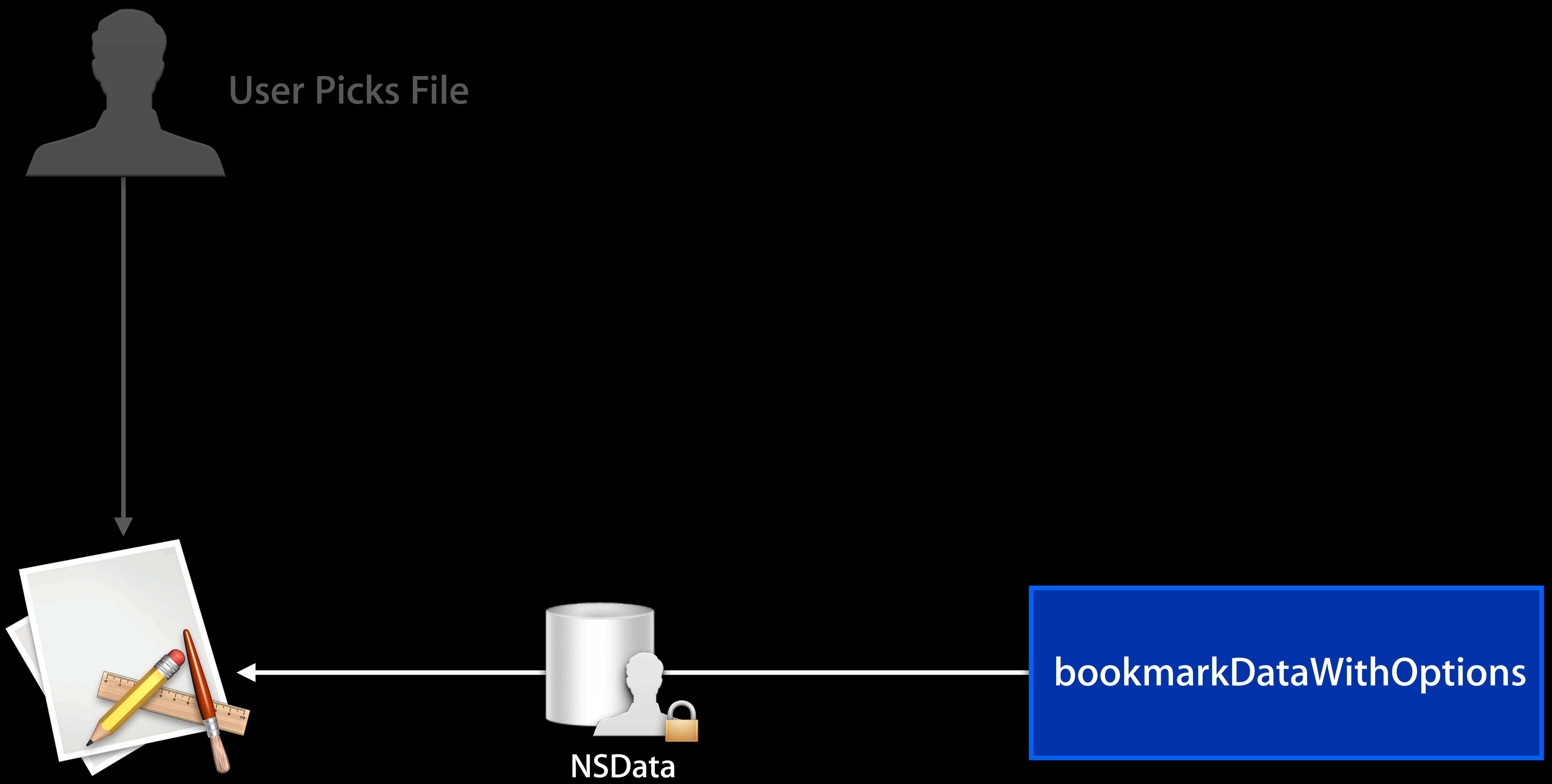
Default email reader:	Mail
Check for new messages:	Automatically
New messages sound:	New Messages Sound
	✓ Play sounds for other mail actions
Dock unread count:	Inbox Only
New message notifications:	Inbox Only
Add invitations to Calendar:	Never
Downloads folder:	Downloads
Remove unedited downloads:	After Message is Deleted
If outgoing server is unavailable:	Show a list of alternate servers

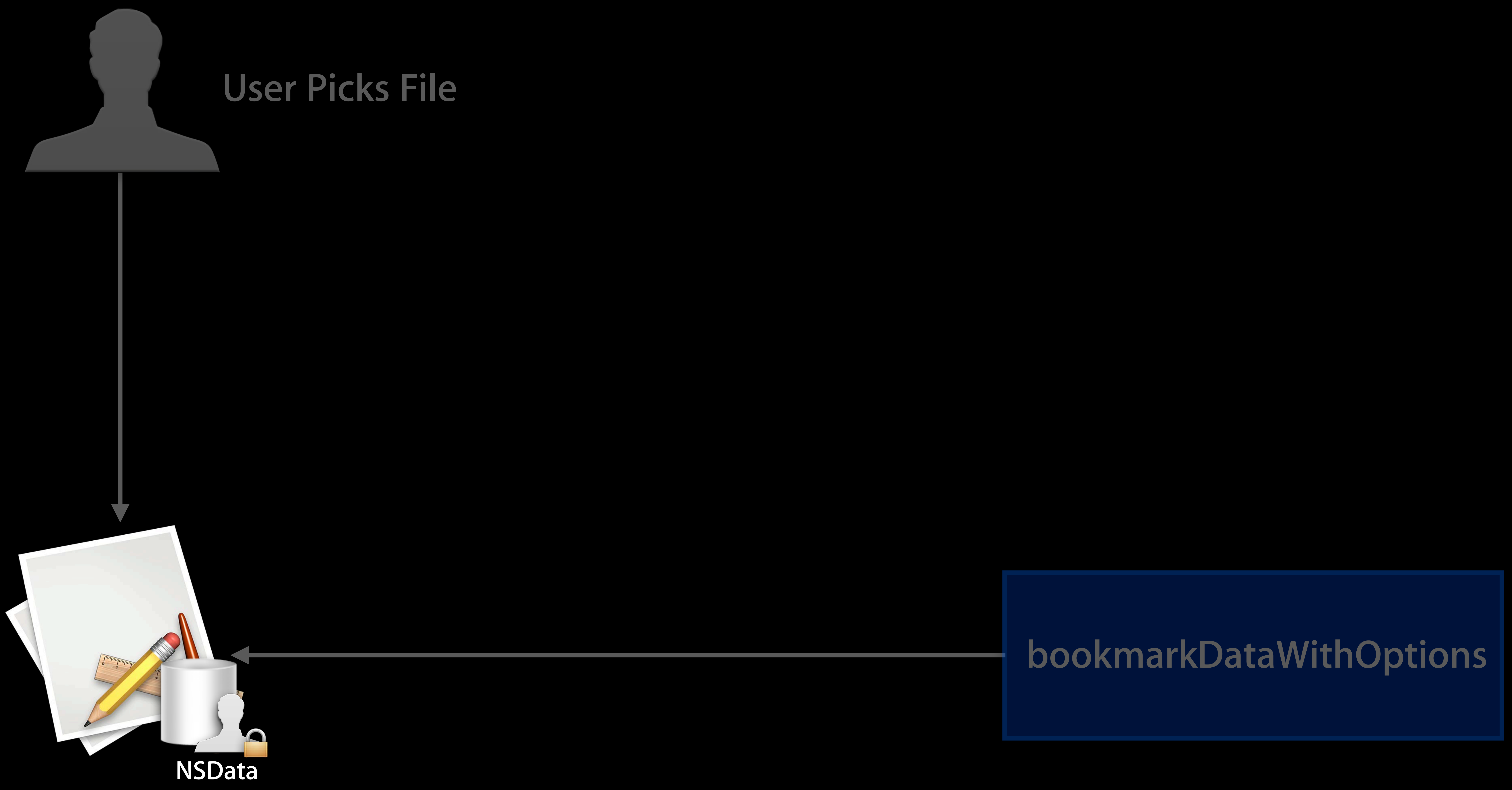
When searching all mailboxes, include results from:

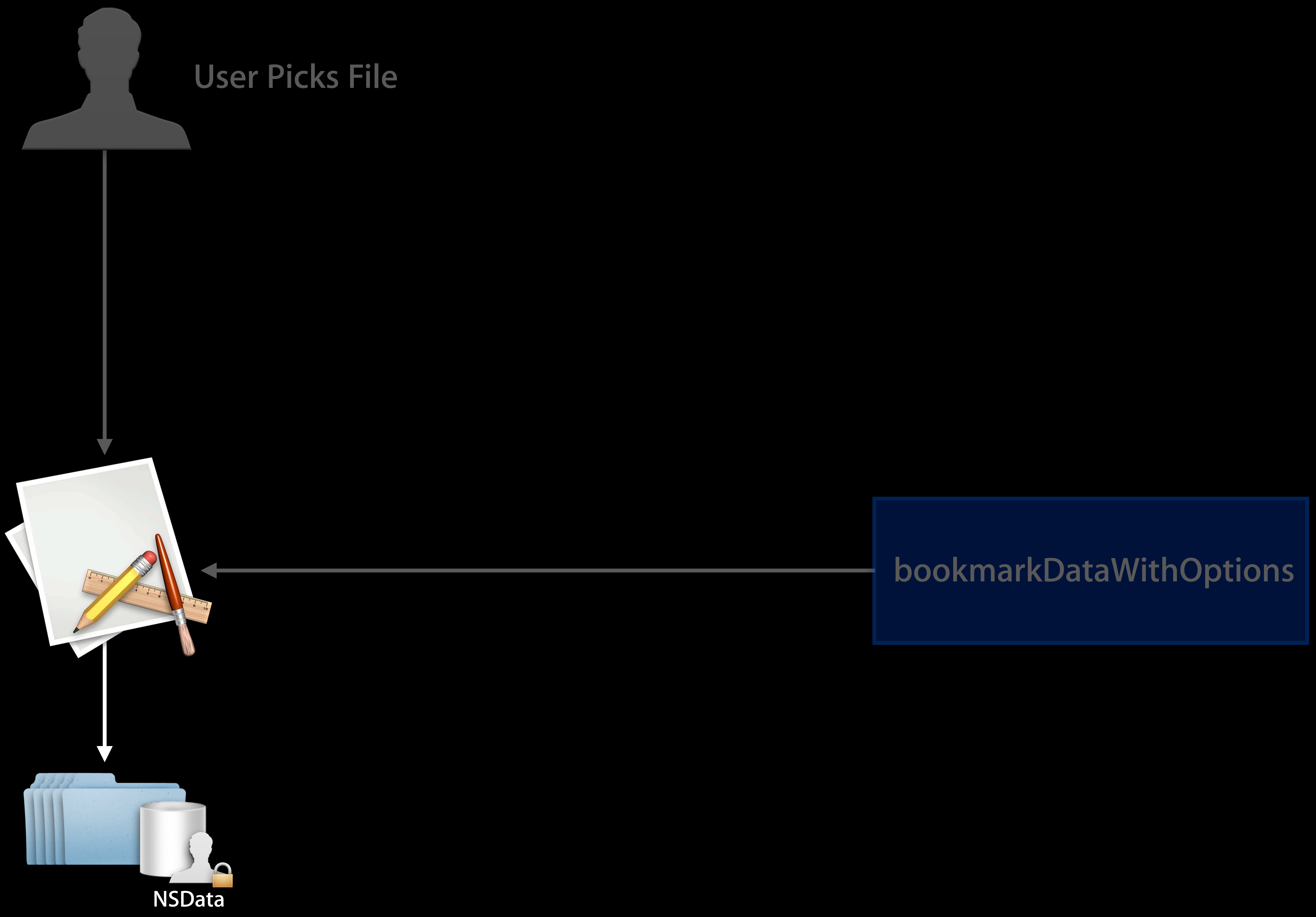
- ✓ Trash
- ☐ Junk
- ☐ Encrypted Messages

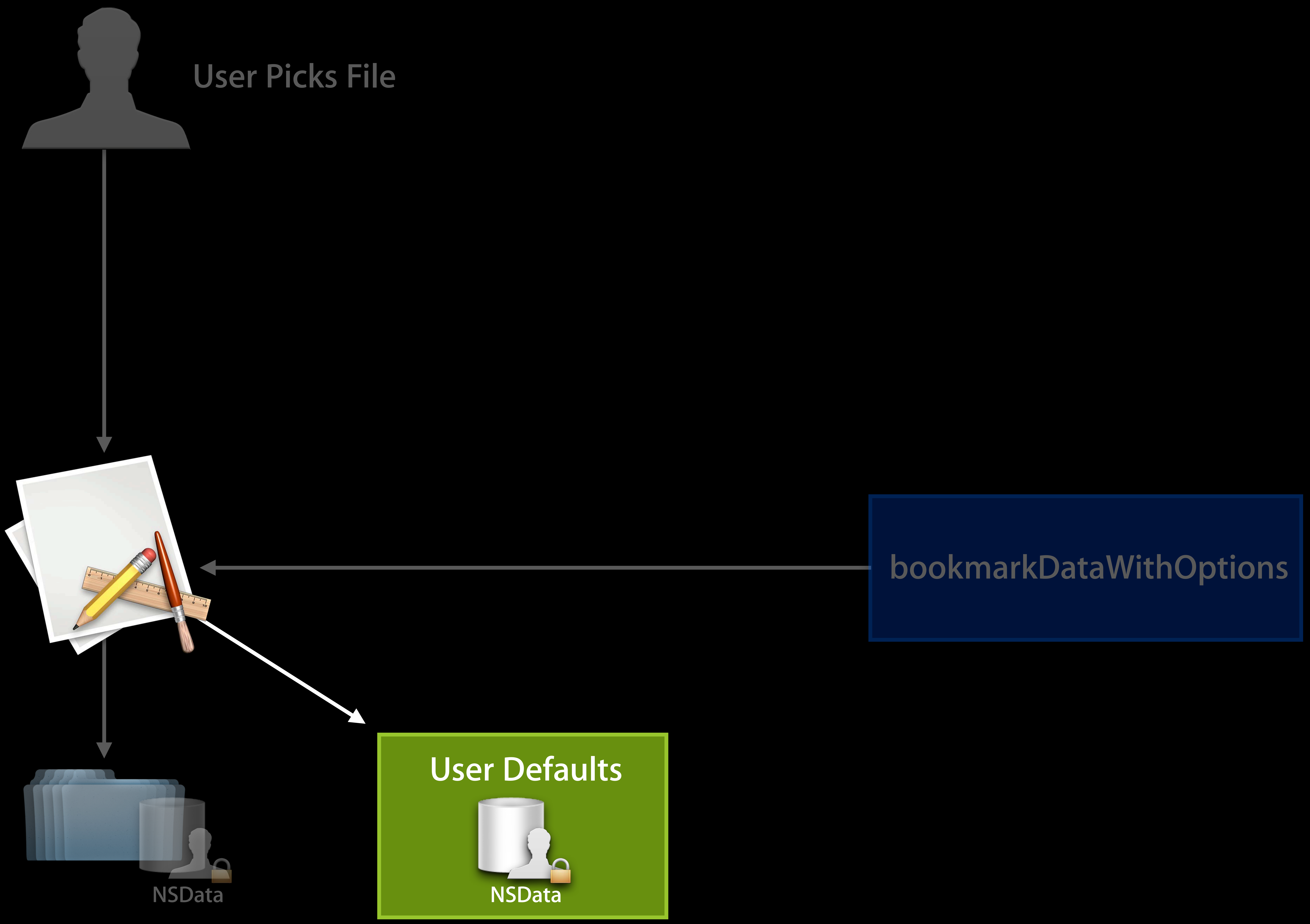

User Picks File

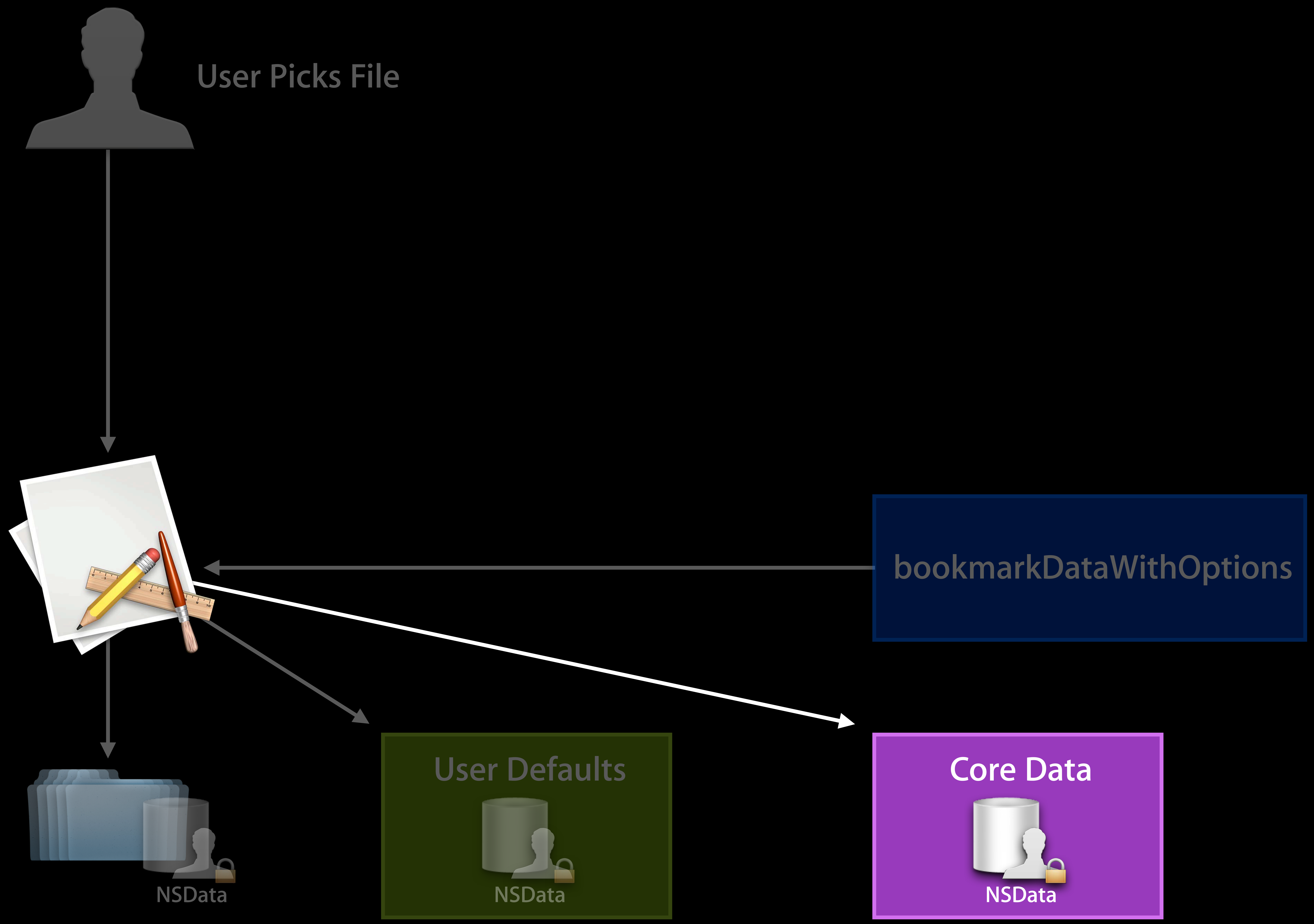

User Picks File
bookmarkDataWithOptions


User Picks File
NSData
bookmarkDataWithOptions

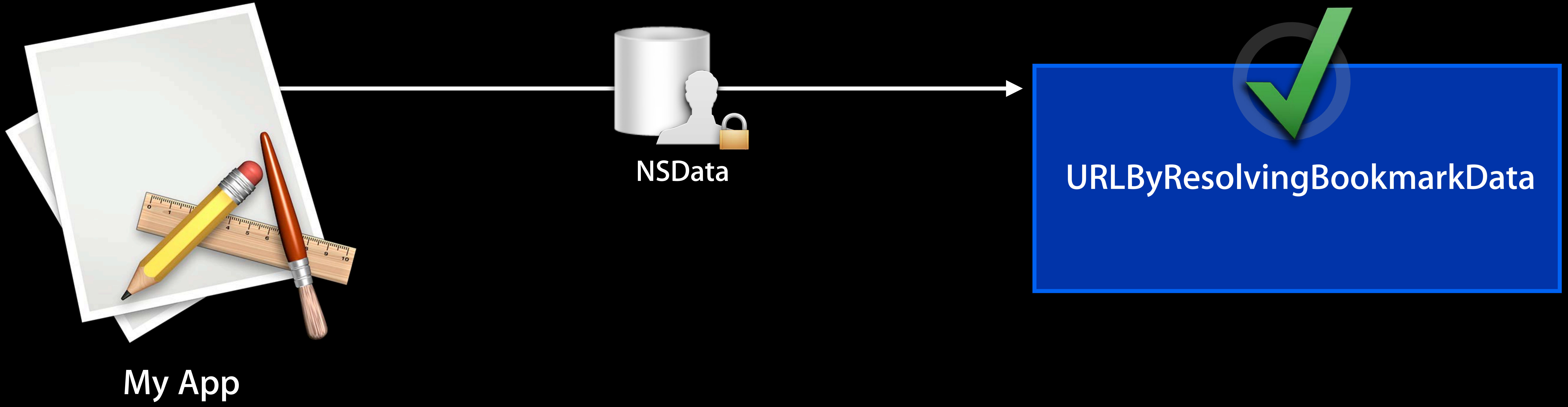

User Picks File
NSData
bookmarkDataWithOptions

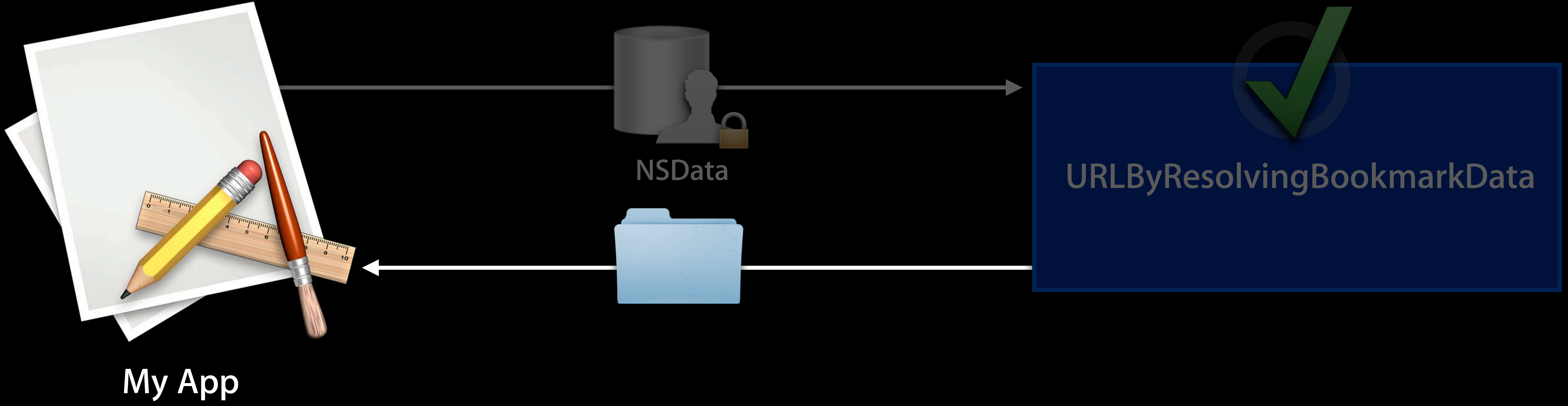

User Picks File
bookmarkDataWithOptions
NSData


User Picks File
bookmarkDataWithOptions
NSData
User Defaults
NSData

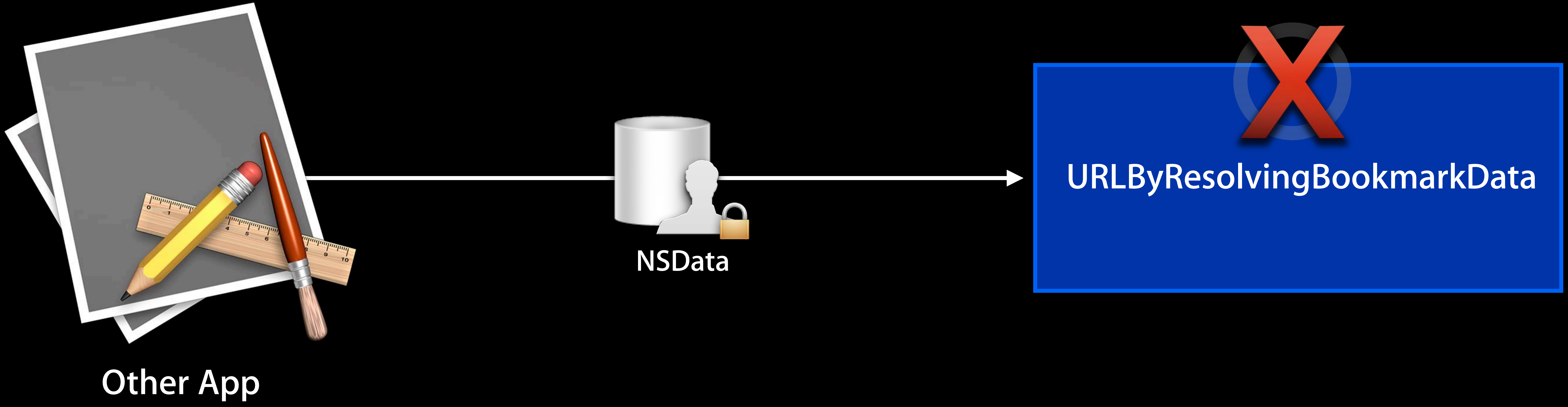

User Picks File
bookmarkDataWithOptions
NSData
User Defaults
NSData
Core Data
NSData

My App


My App
NSData
URLByResolvingBookmarkData


NSData
URLByResolvingBookmarkData
My App

Other App

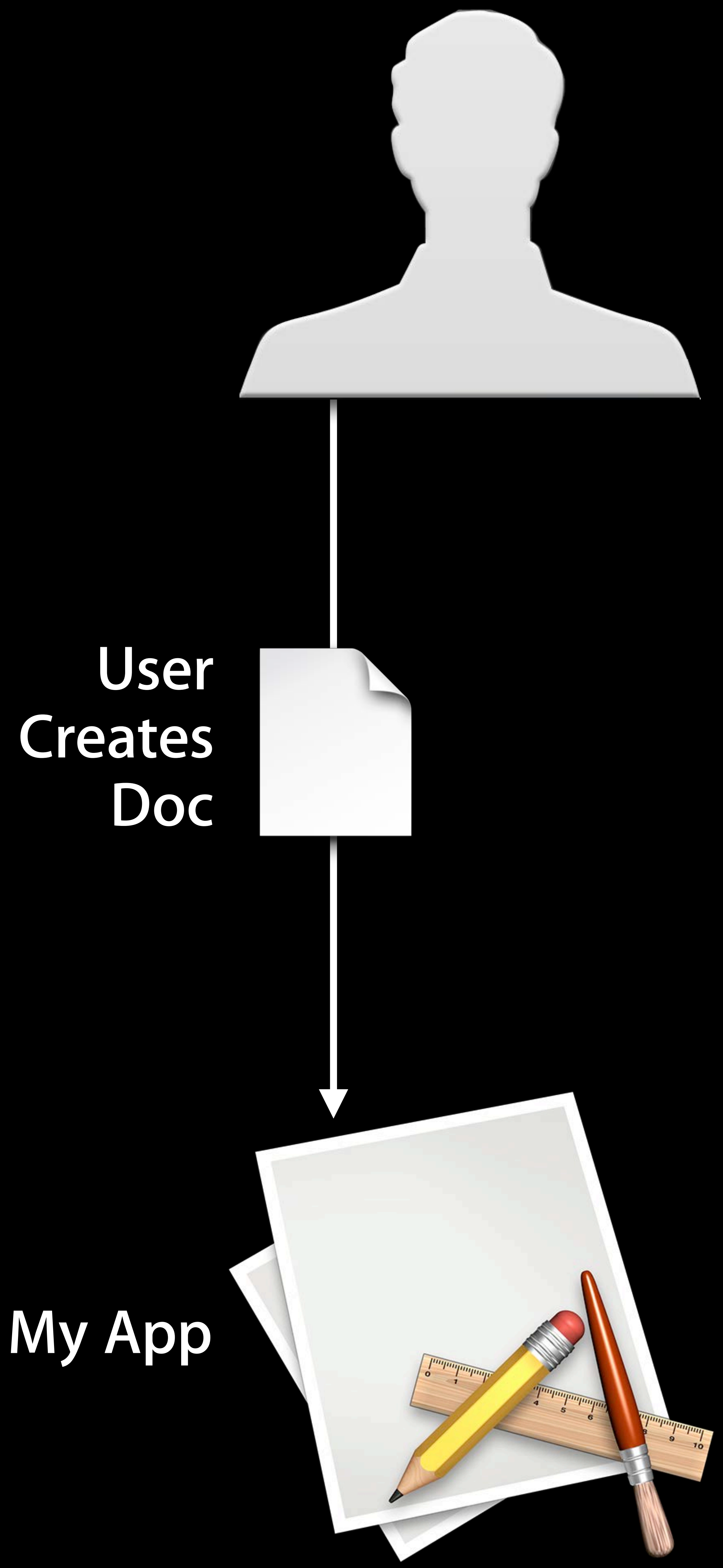

Other App
NSData
URLByResolvingBookmarkData

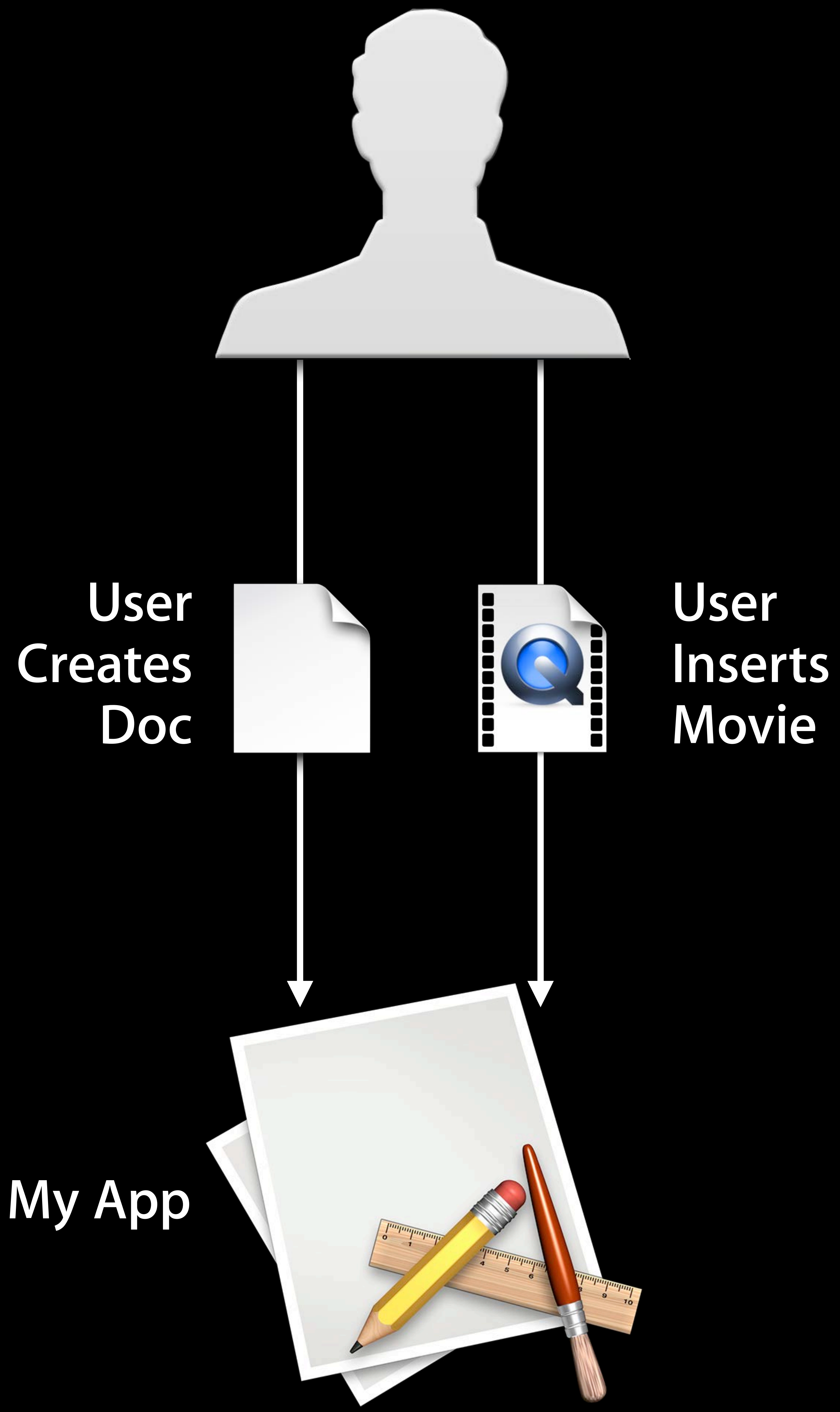
Security-Scoped Bookmarks

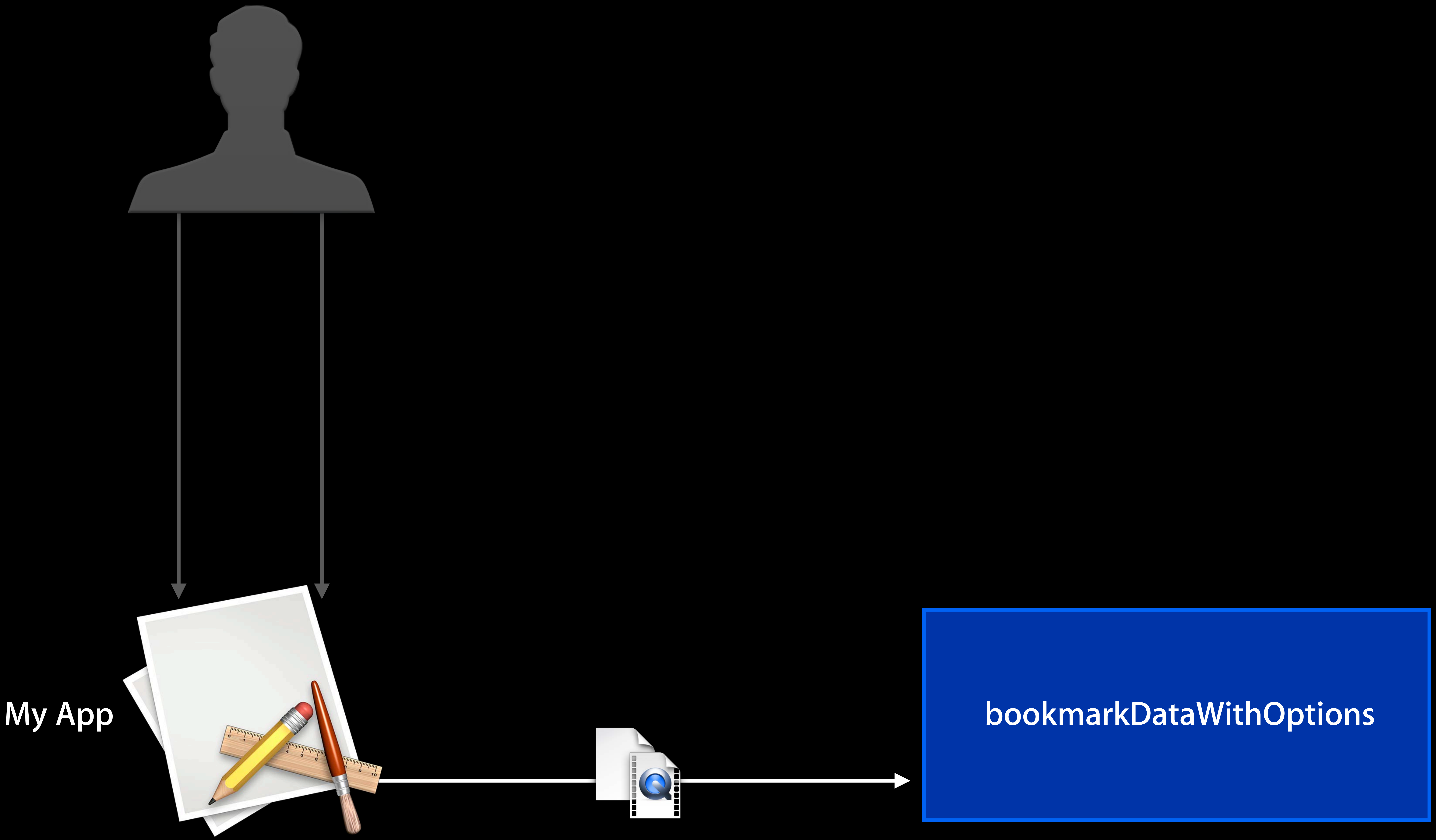
- Document scope

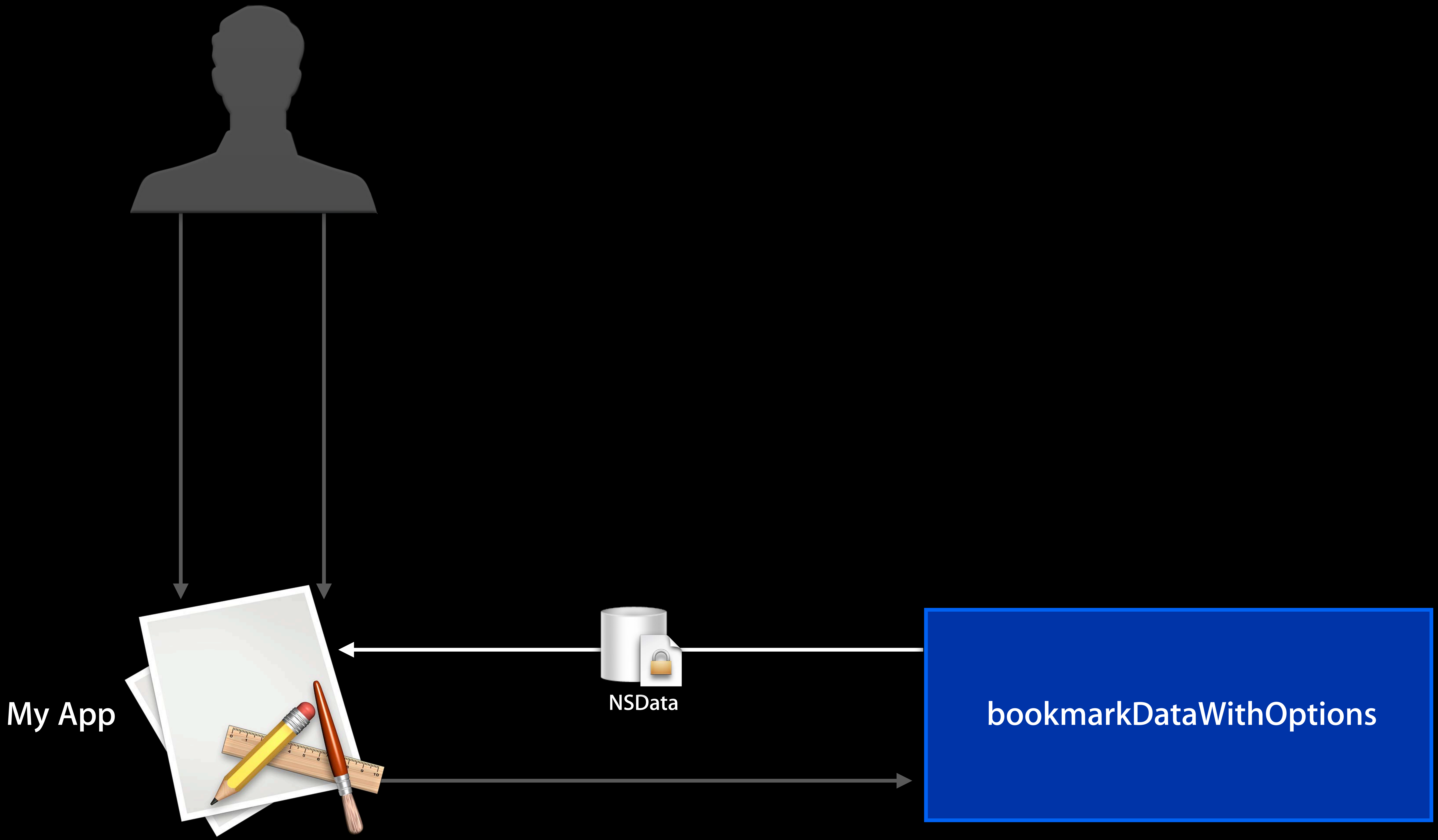
  `com.apple.security.files.bookmarks.document-scope`

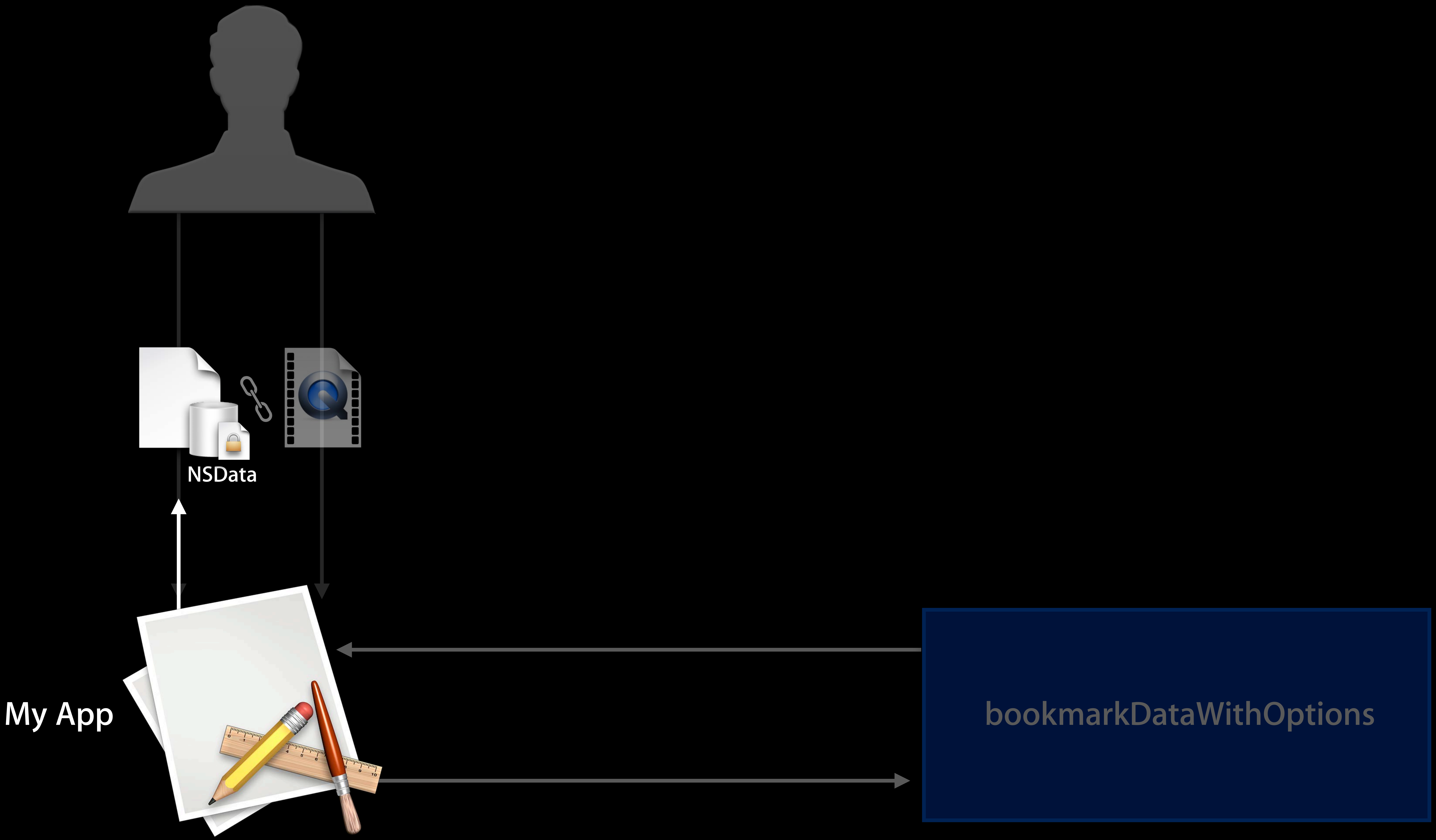
  - Allows a document format to contain references to files (but not folders) that travel with it
  - Bookmark must be stored in the document file/bundle itself
  - Cannot point to system or hidden locations (~/Library)

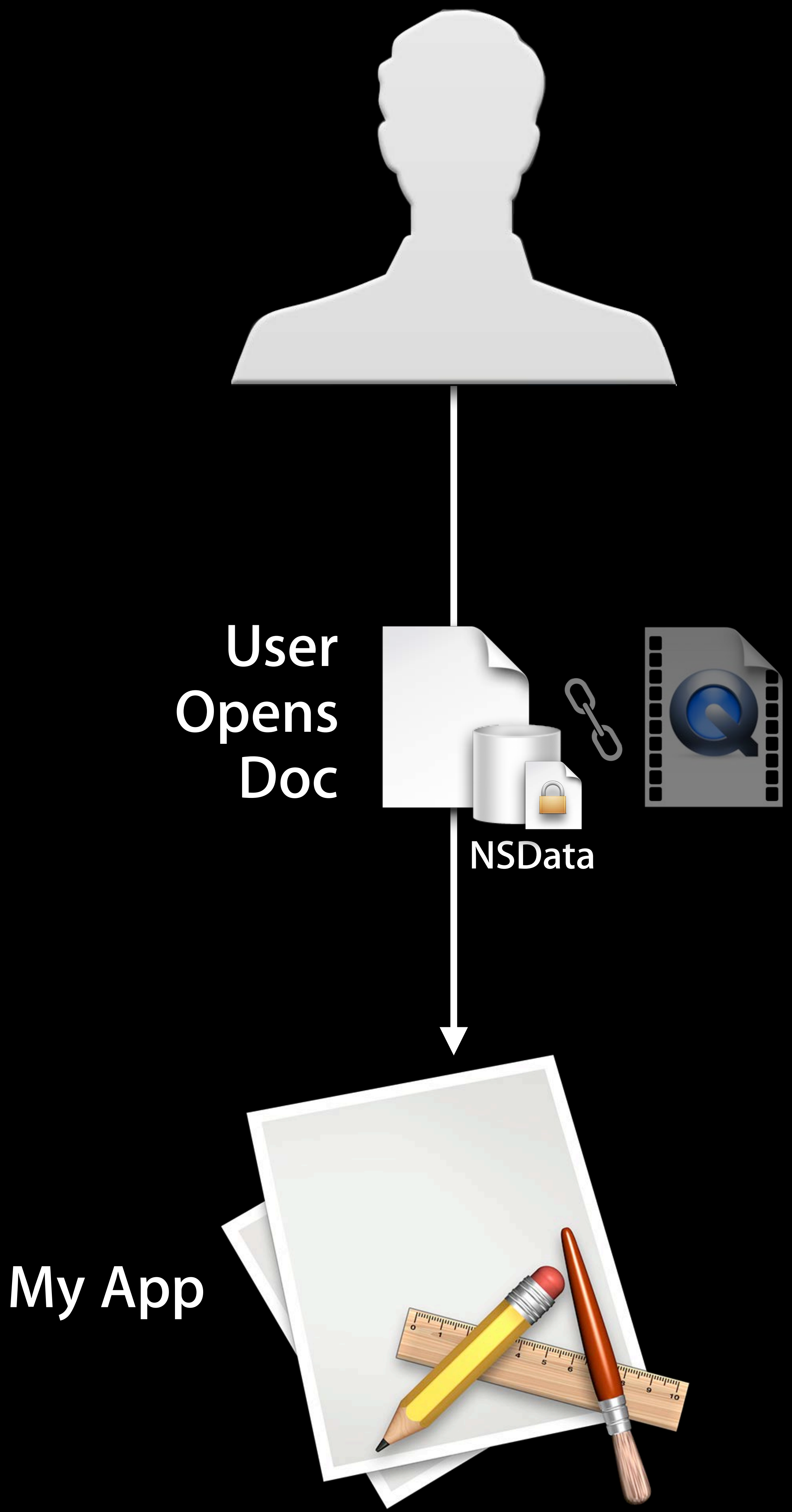

User Creates Doc
My App

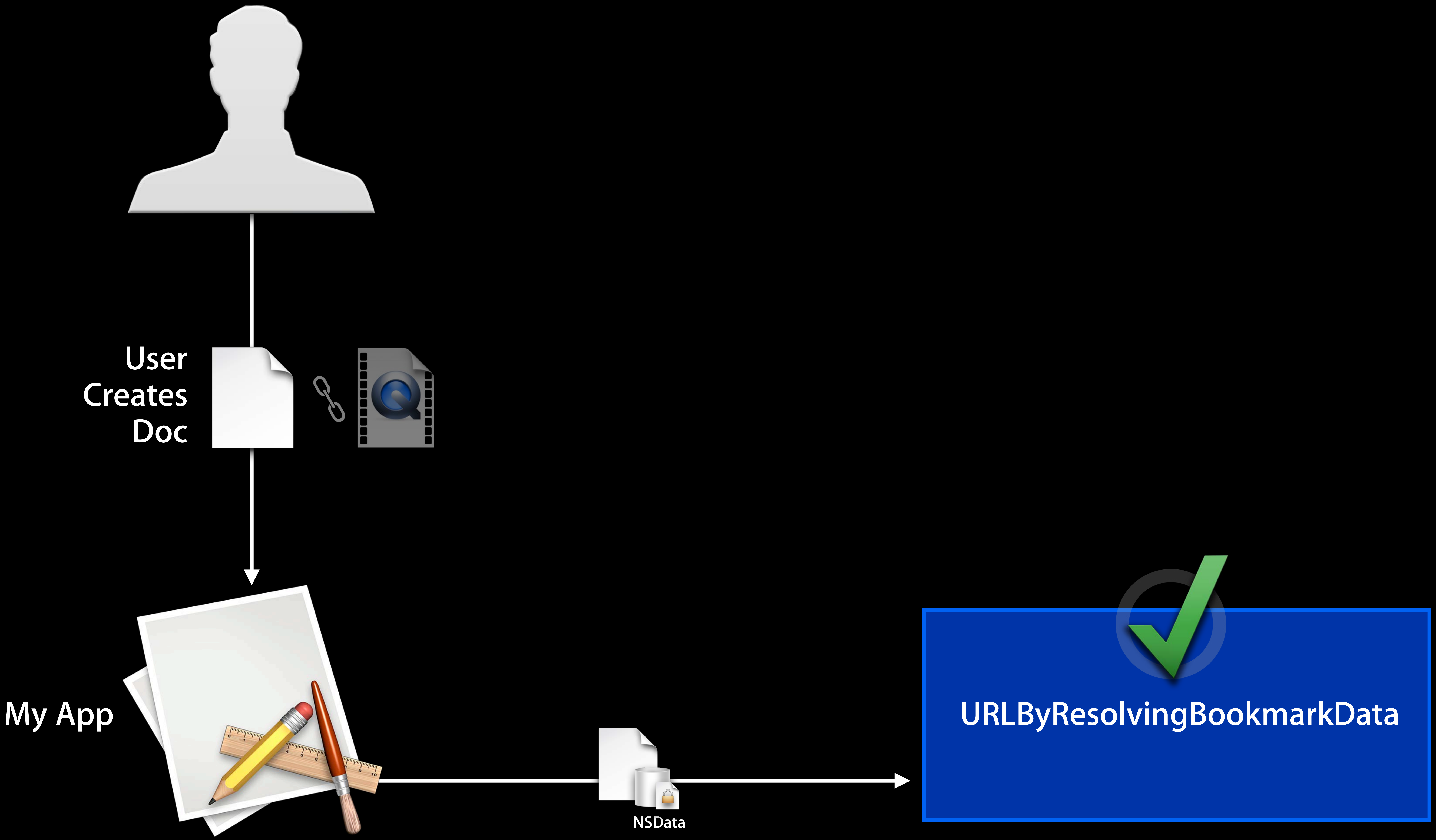

User
Creates
Doc
User
Inserts
Movie
My App

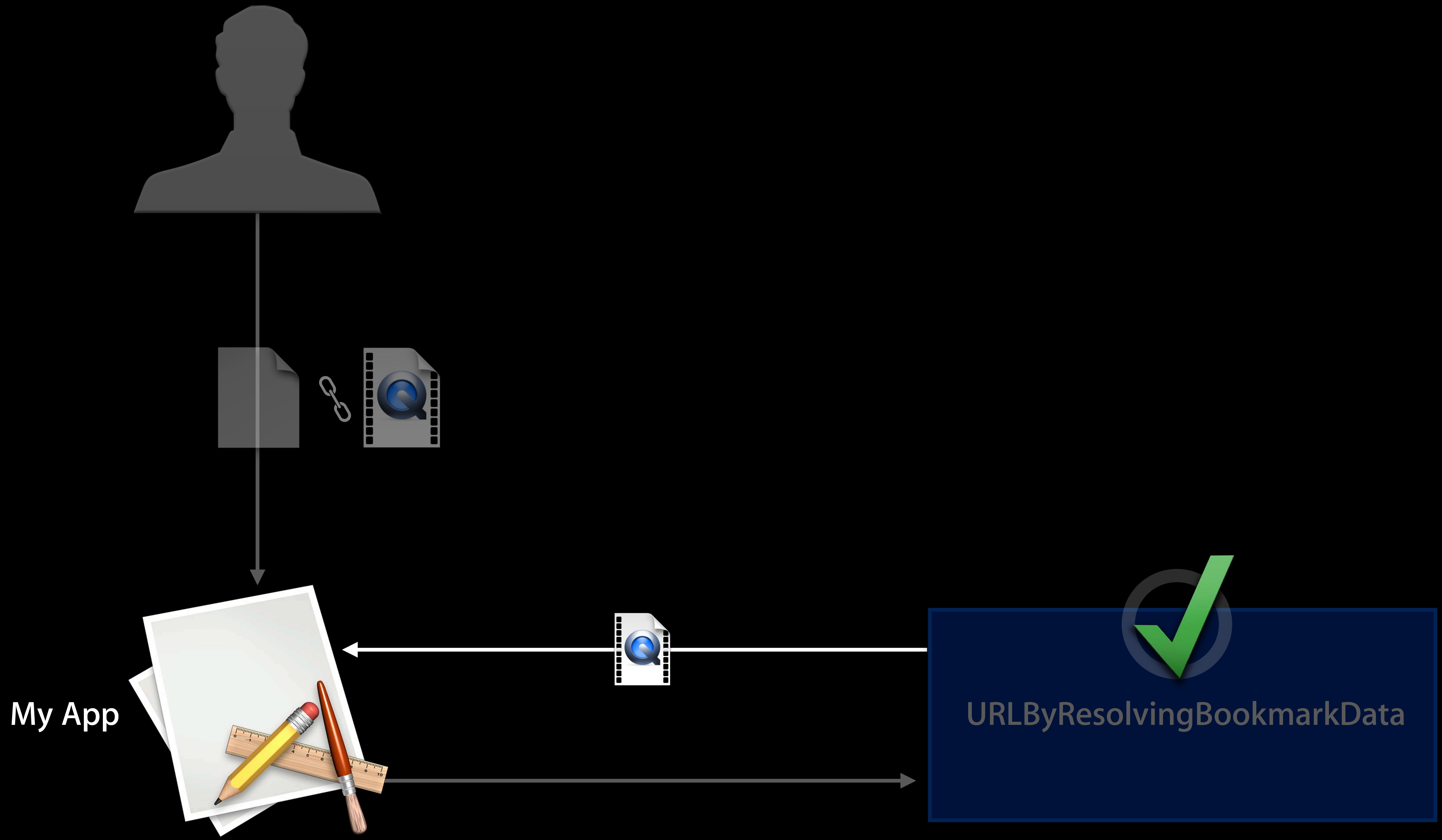

My App
bookmarkDataWithOptions

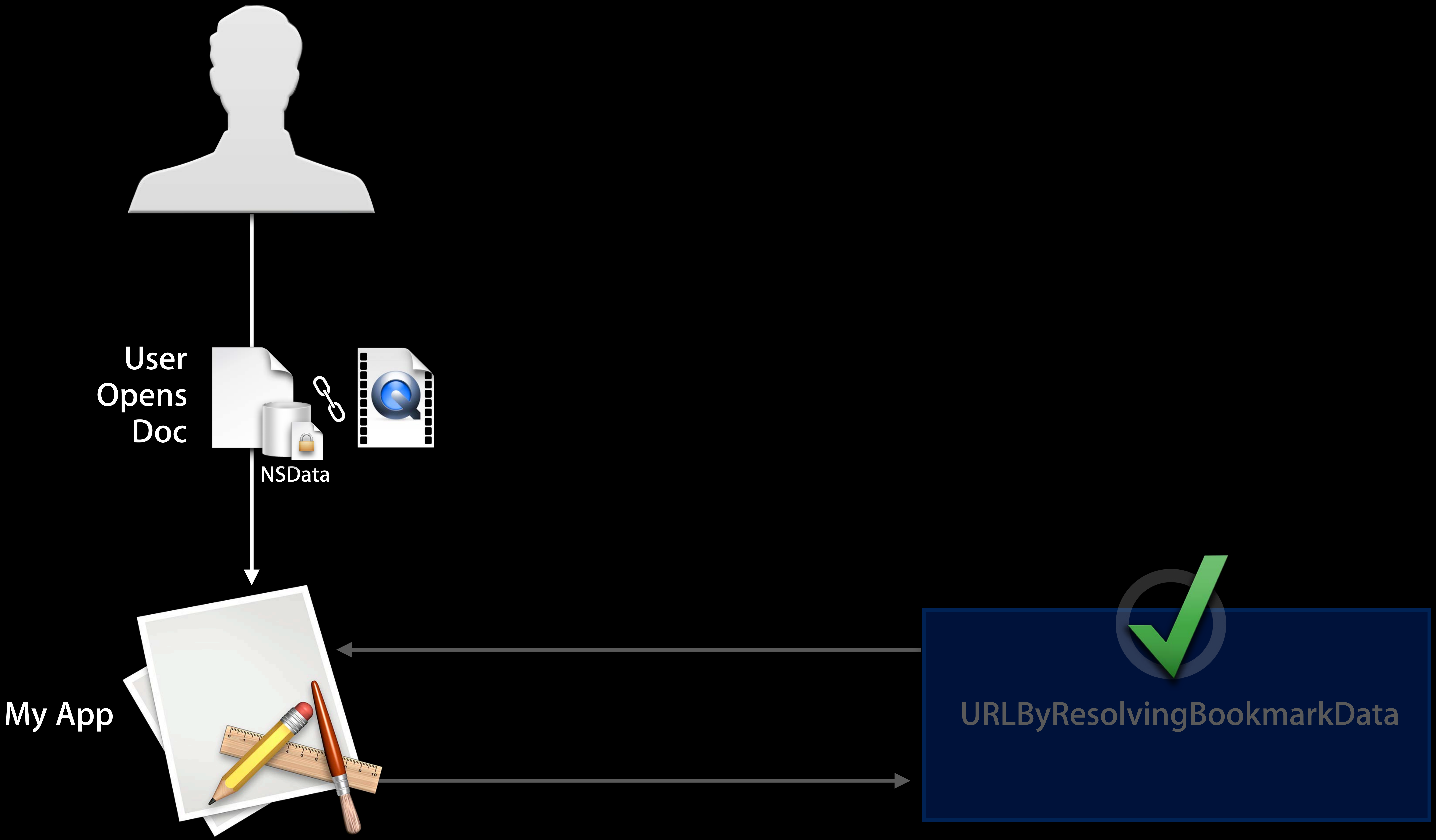

My App
NSData
bookmarkDataWithOptions

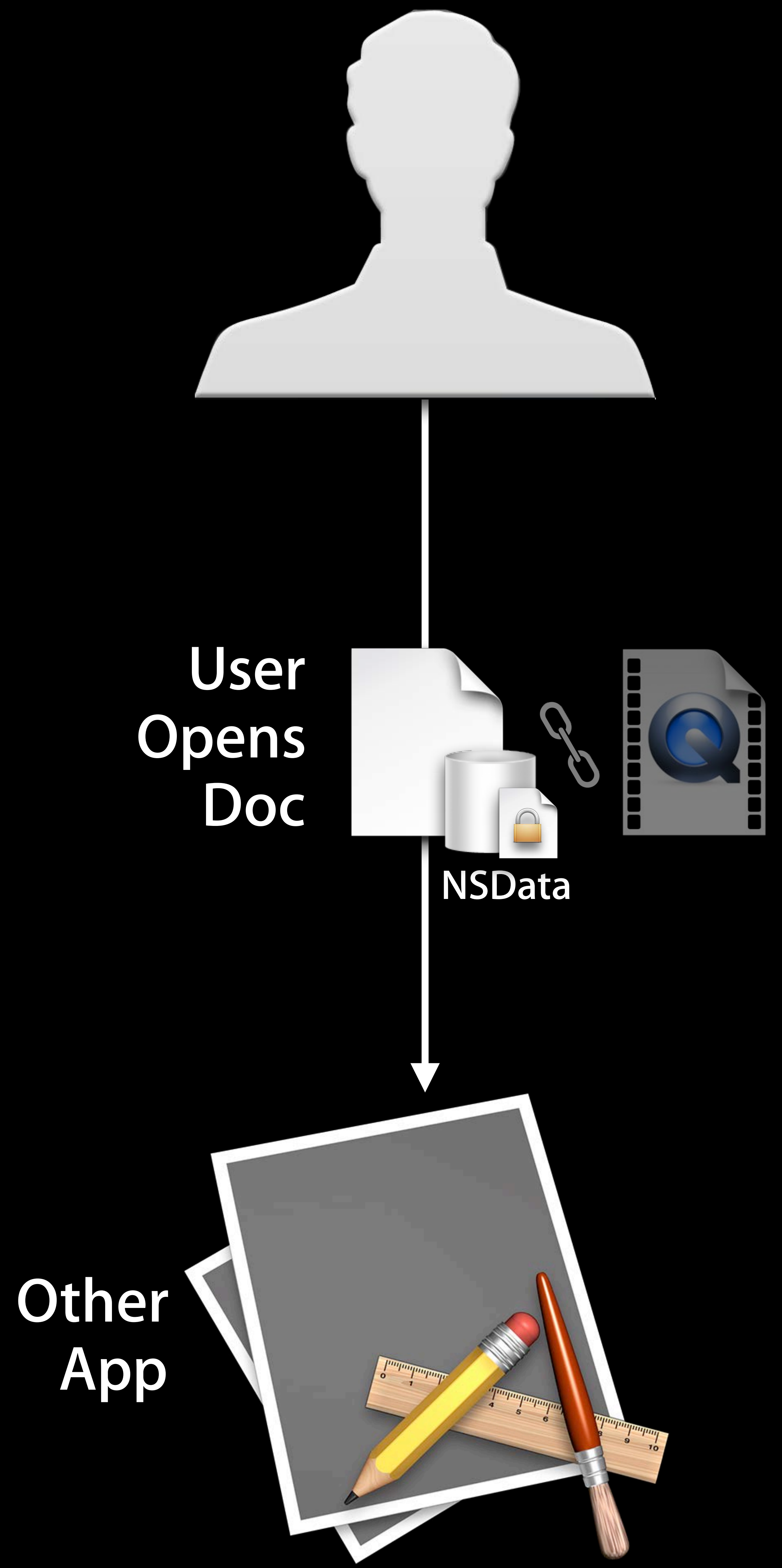

NSData
My App
bookmarkDataWithOptions

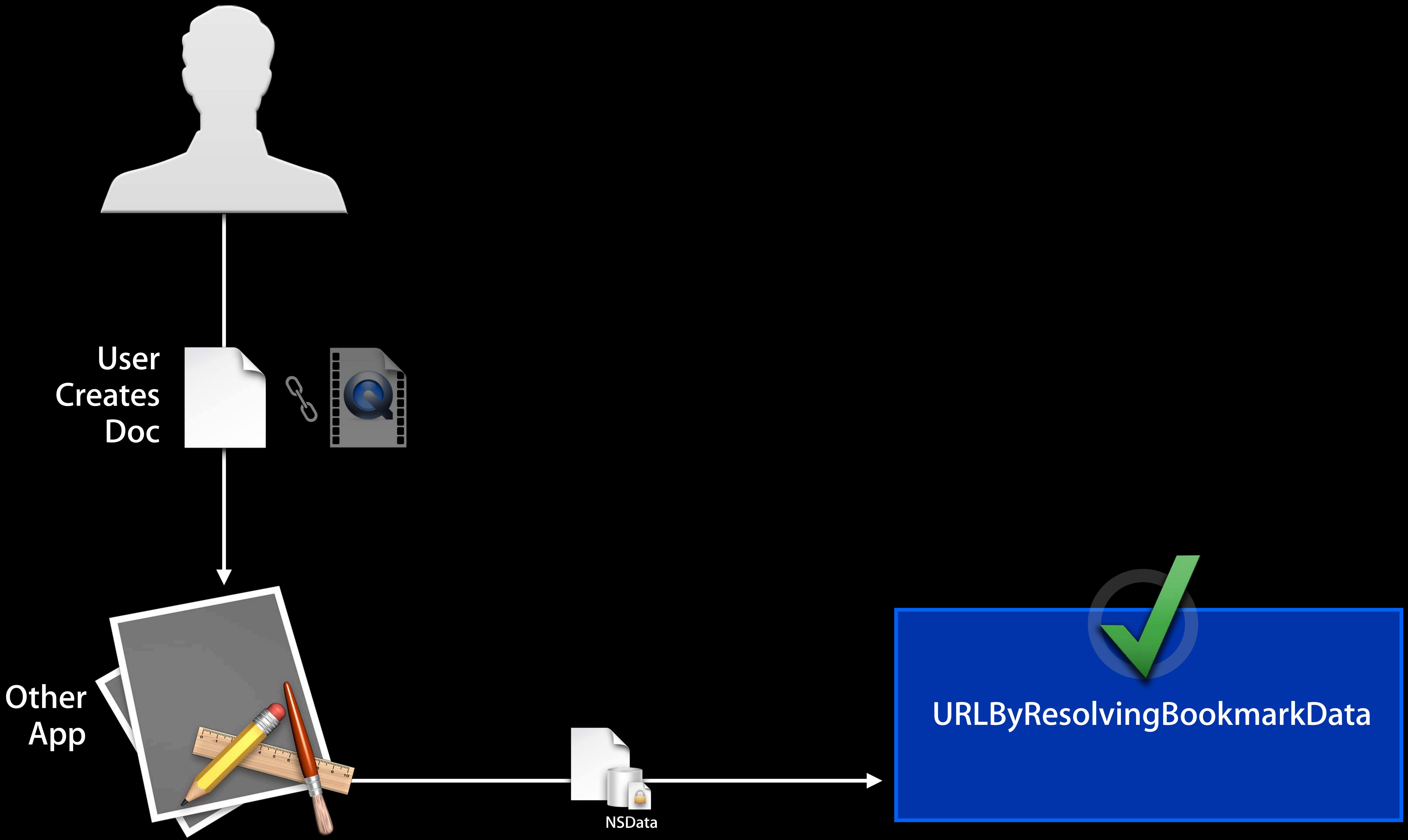

User
Opens
Doc
NSData
My App

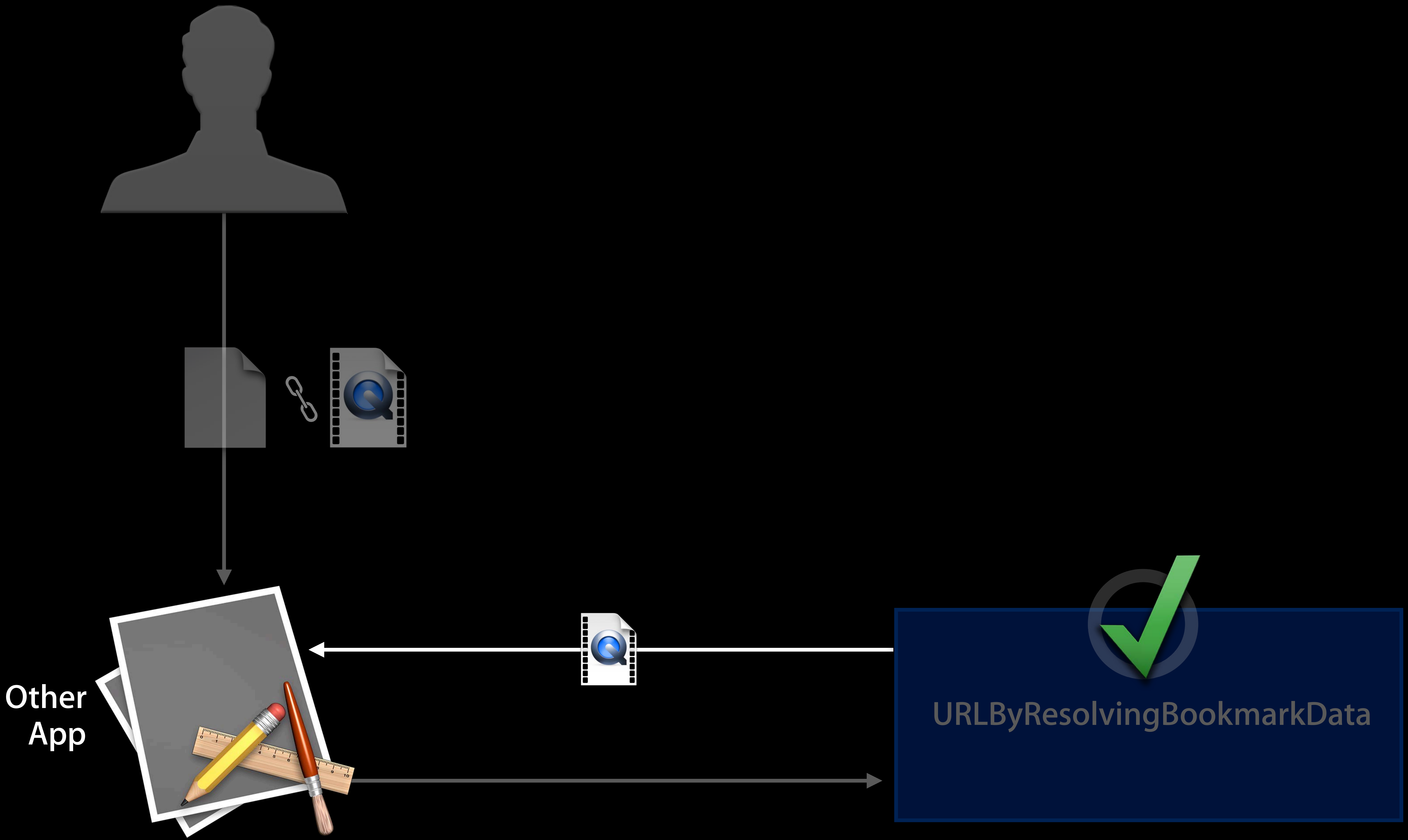

User Creates Doc
My App
NSData
URLByResolvingBookmarkData

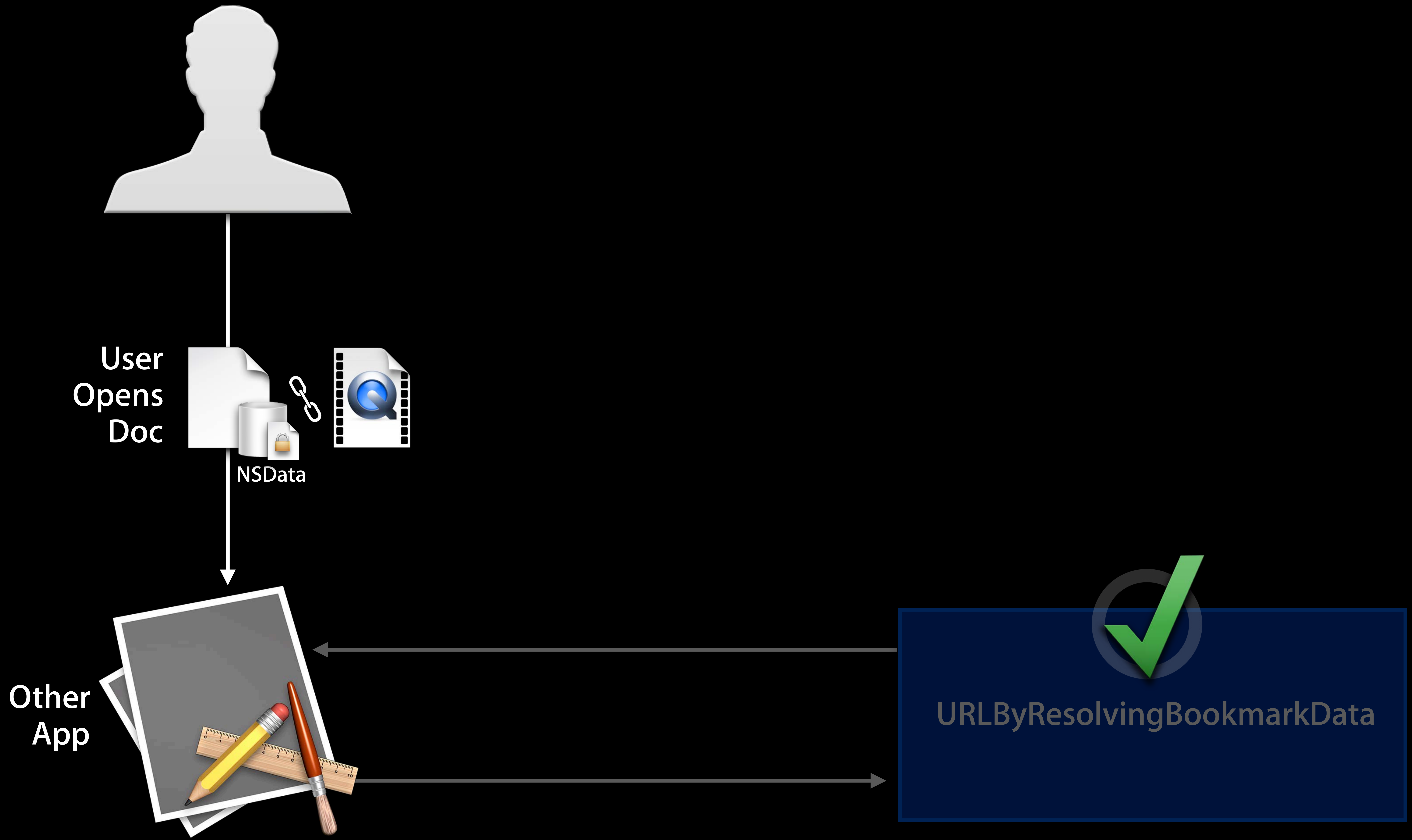

My App
URLByResolvingBookmarkData


User Opens Doc
NSData
My App
URLByResolvingBookmarkData


User
Opens
Doc
NSData
Other
App


User
Creates
Doc
Other
App
NSData
URLByResolvingBookmarkData


Other App
URLByResolvingBookmarkData


User
Opens
Doc
NSData
Other
App
URLByResolvingBookmarkData

Security-Scoped Bookmarks

- No new API, just a flag on existing NSURL methods

```
+ URLByResolvingBookmarkData:options:relativeToURL:bookmarkDataIsStale:error:
– bookmarkDataWithOptions:includingResourceValuesForKeys:relativeToURL:error:
```

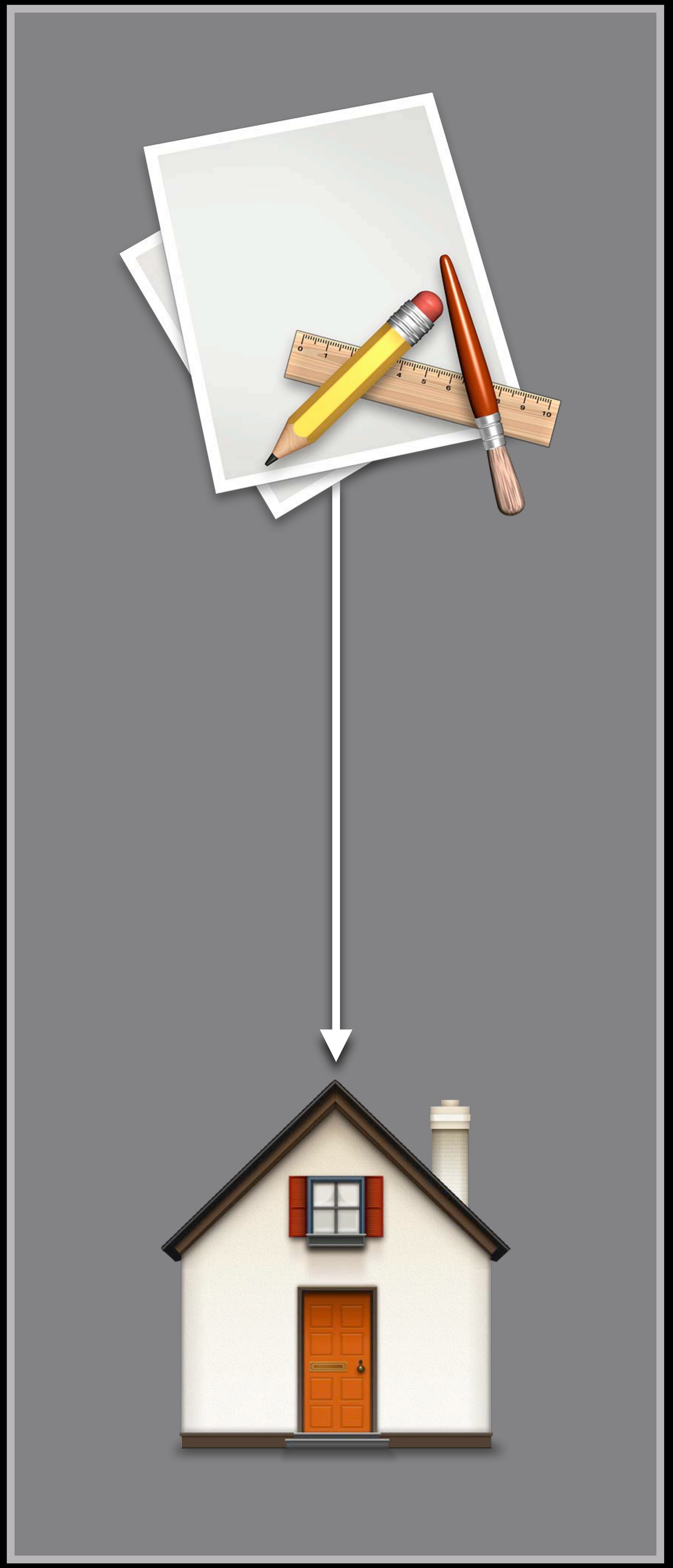
- Big difference—Resolution returns a security-scoped *NSURL*
    - Must call `{start,stop}AccessingSecurityScopedResource` to gain and discontinue access to resource

Application Groups

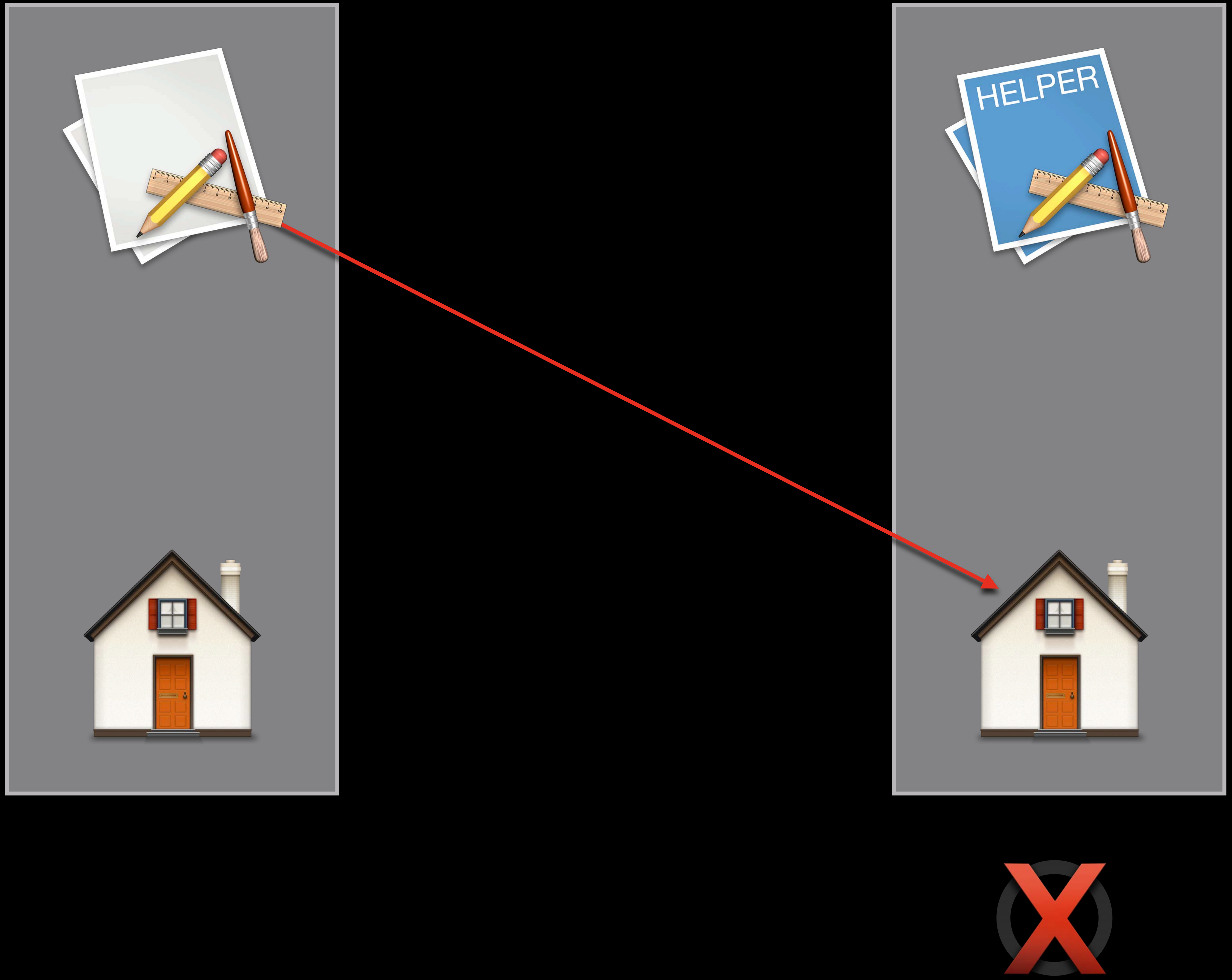
Application Groups

`com.apple.security.application-groups`

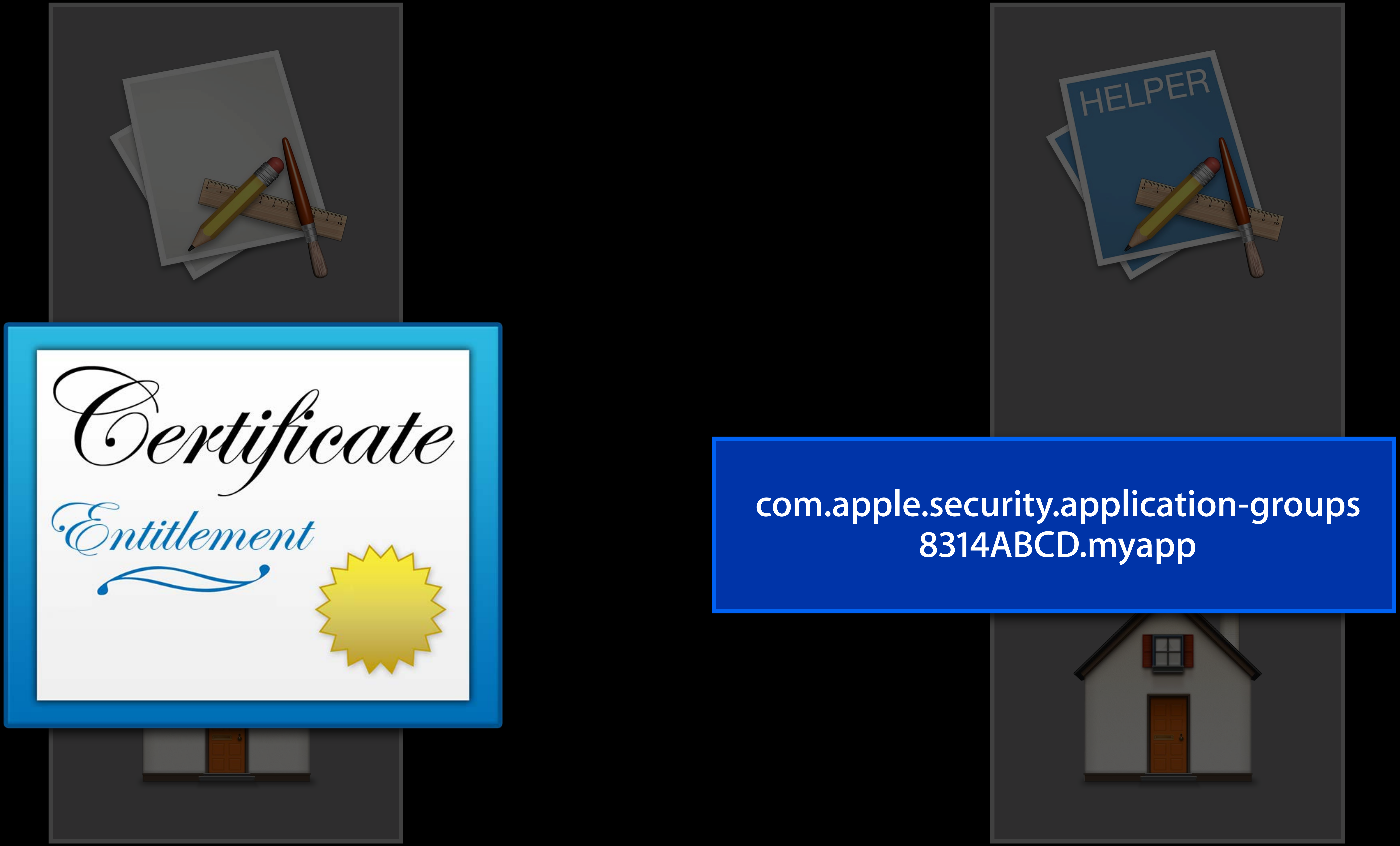
- Each group name must begin with Apple-assigned Team ID
- Useful for suites of different apps, or a single app and its helper(s)
- Direct IPC permitted: XPC, POSIX
- Each group is assigned a shared file system location



✓

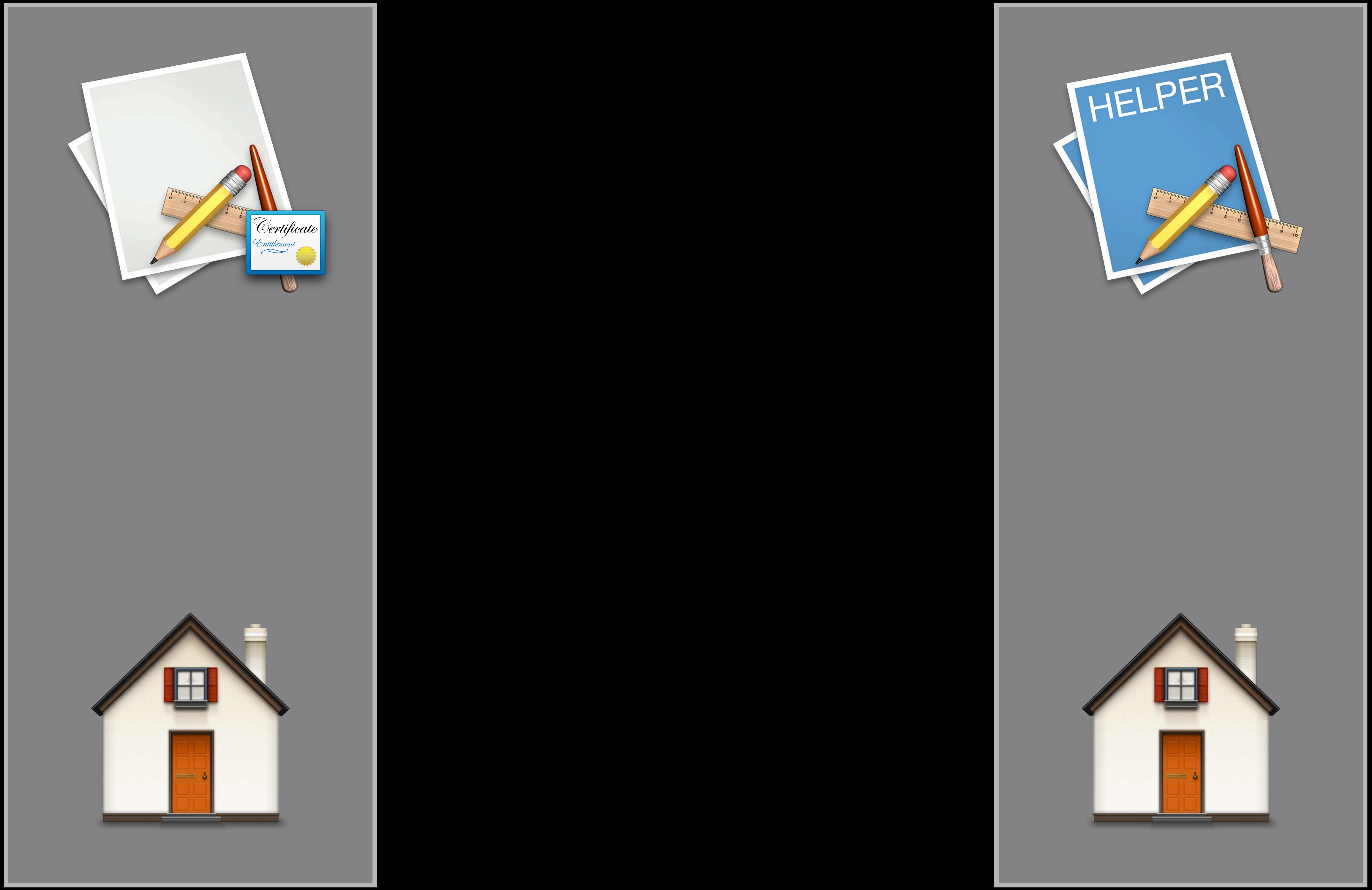

HELPER

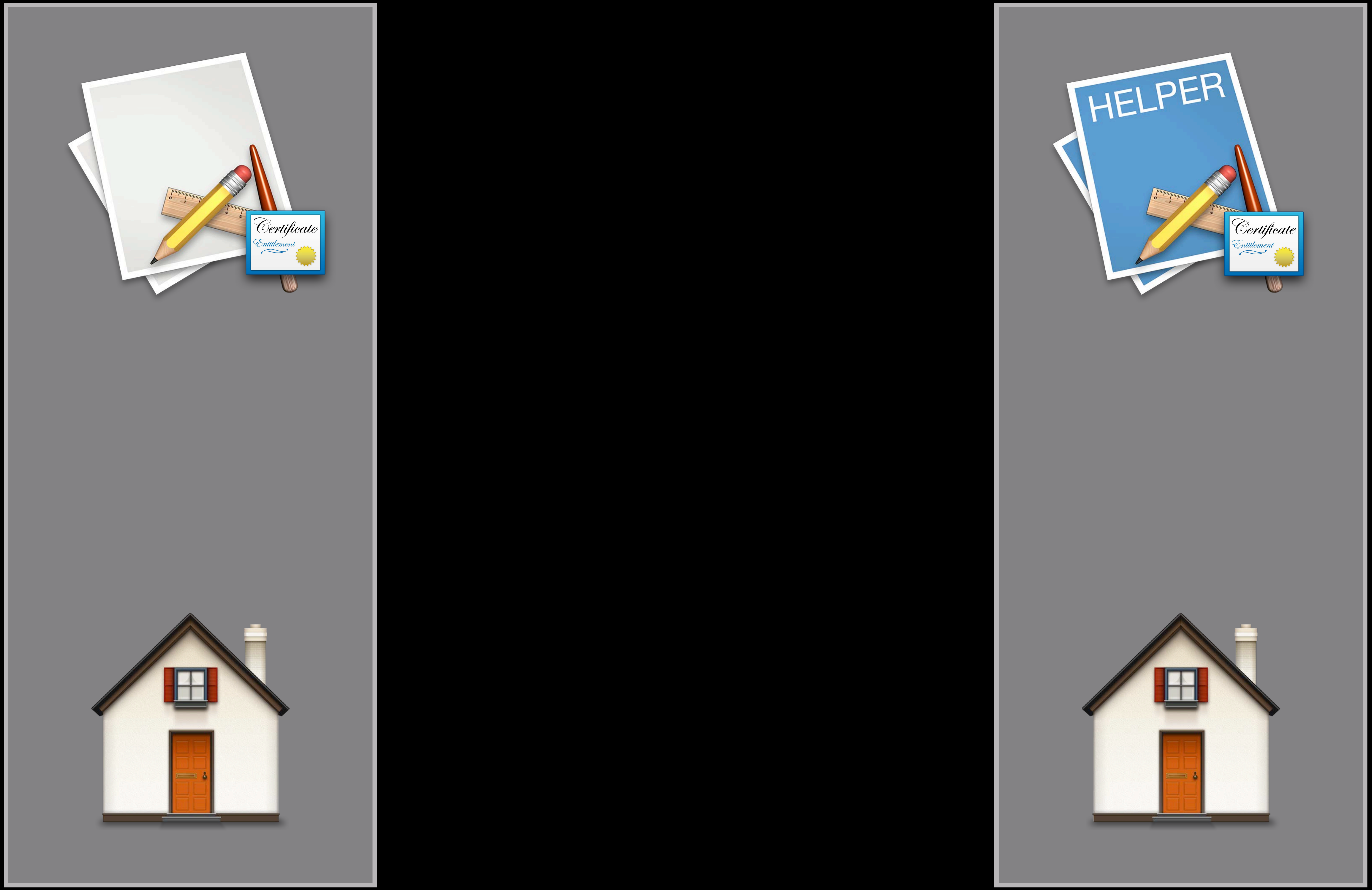
HELPER

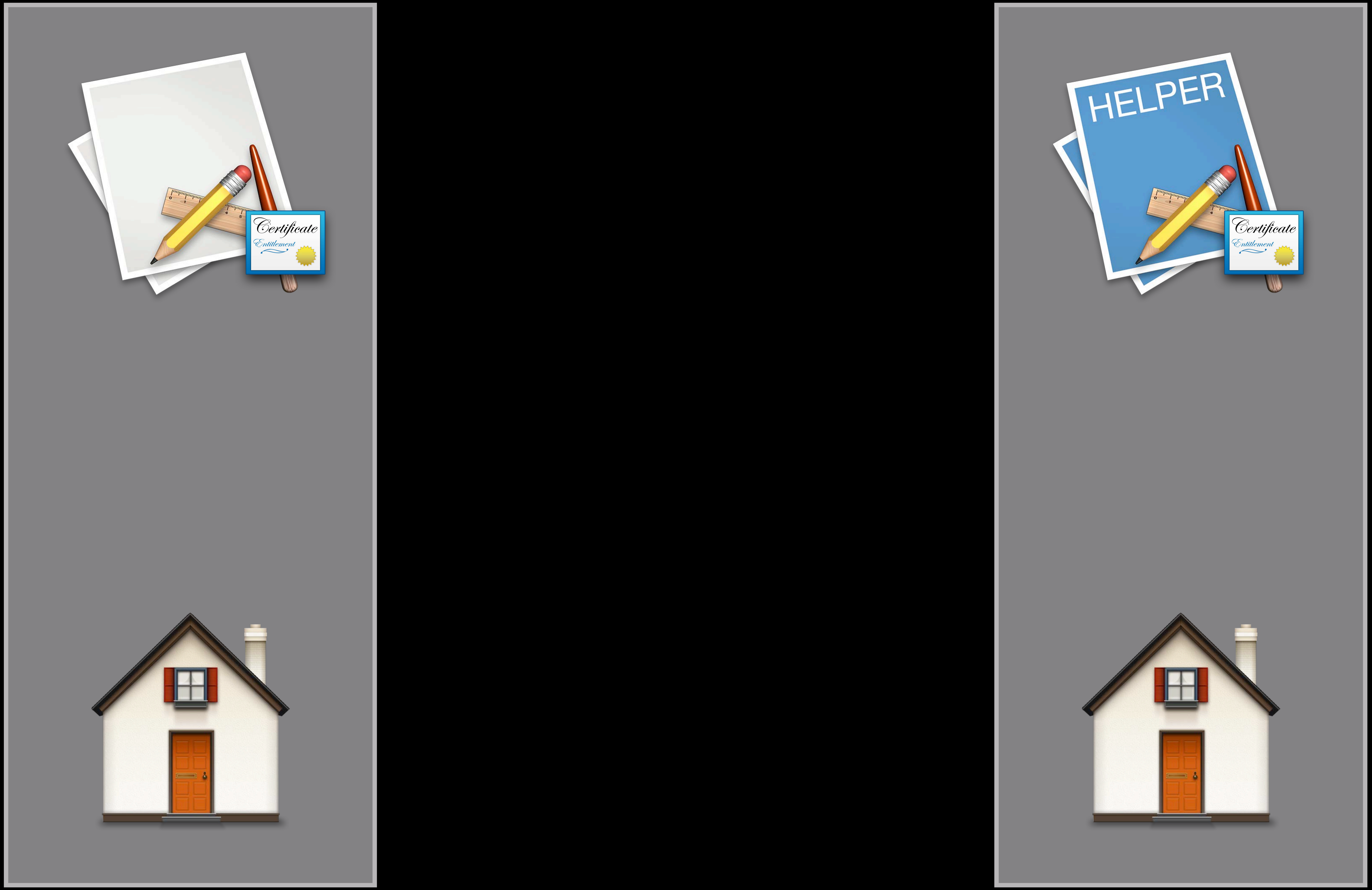


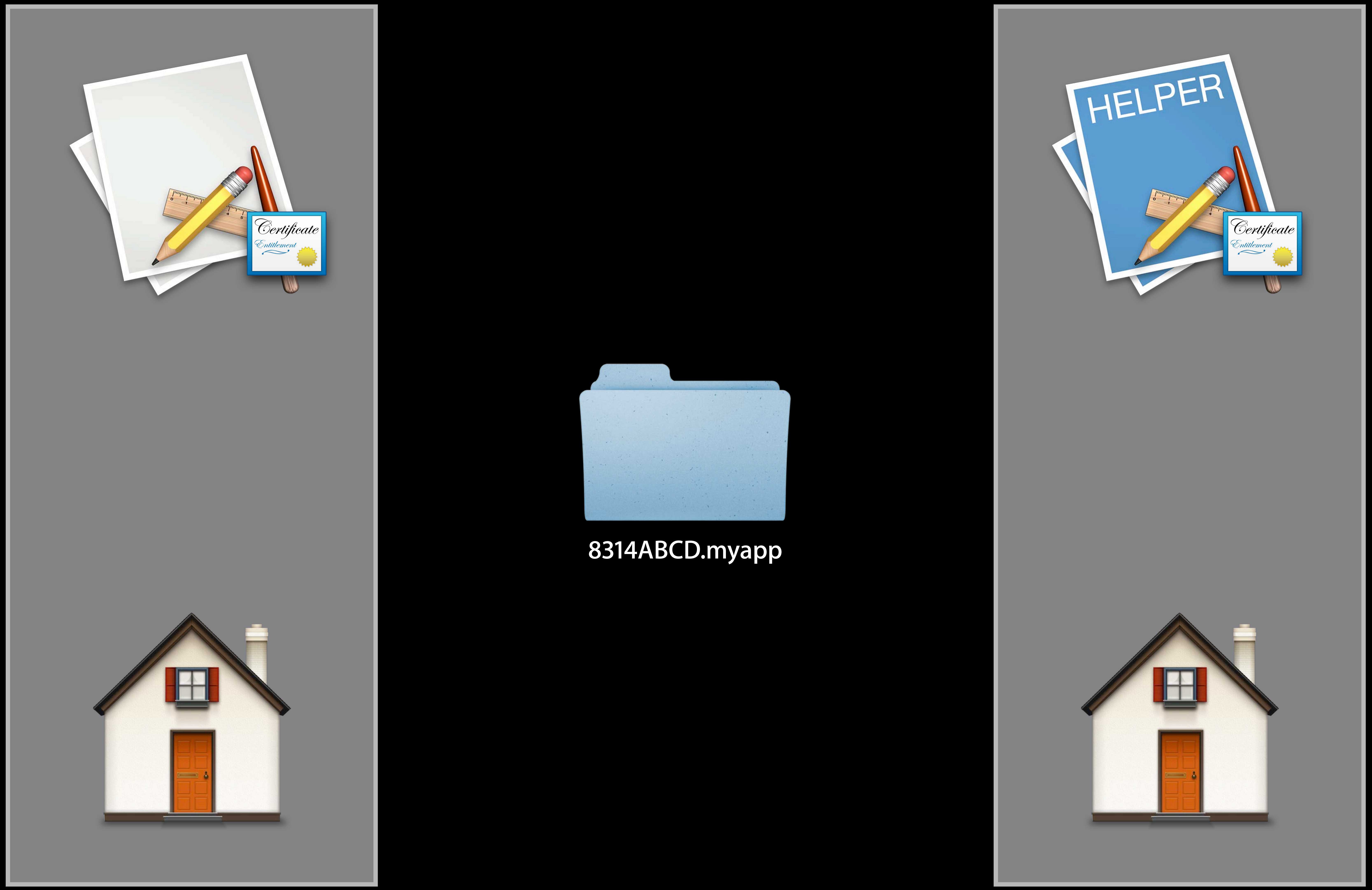
com.apple.security.application-groups
8314ABCD.myapp

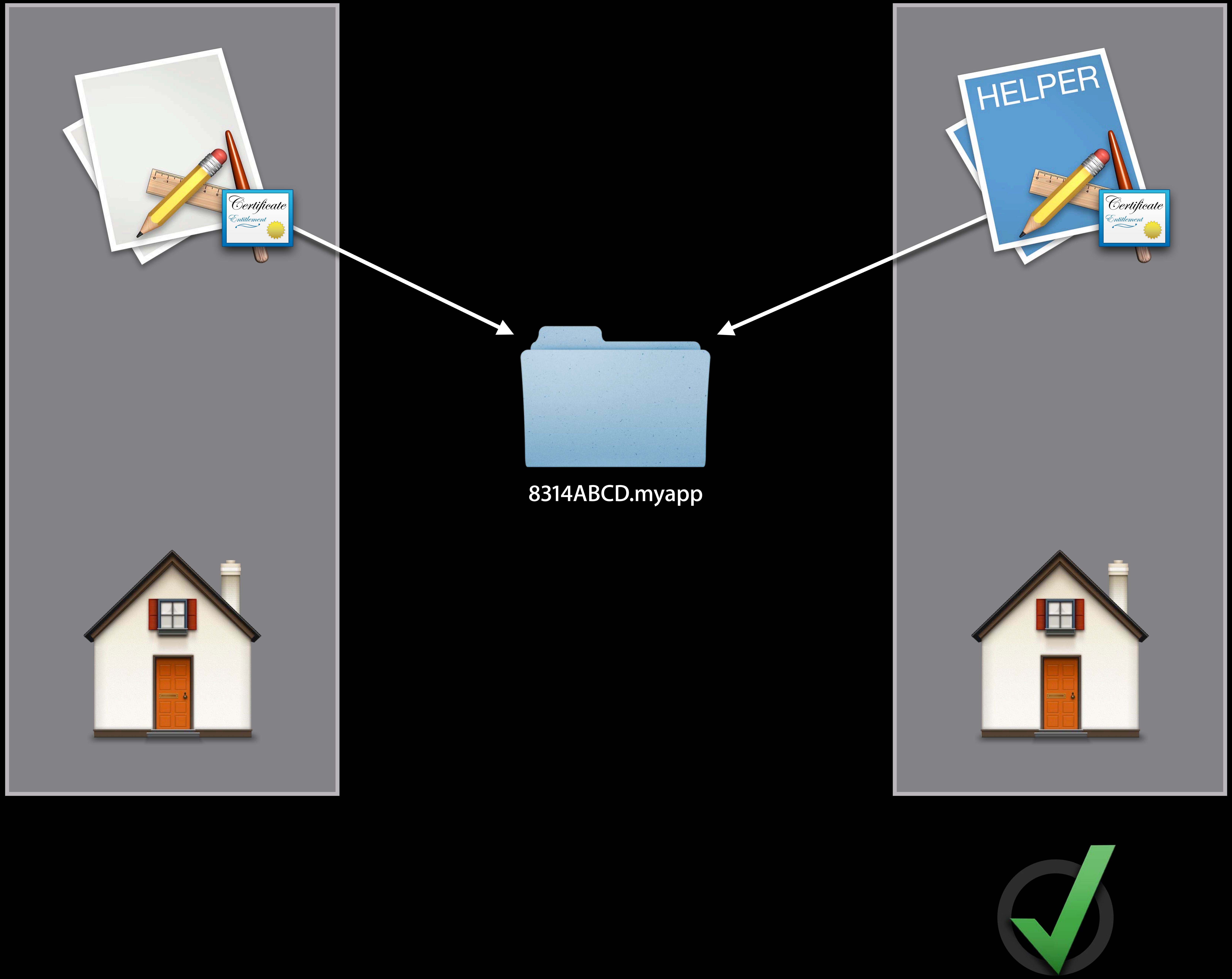
HELPER
Certificate
Entitlement

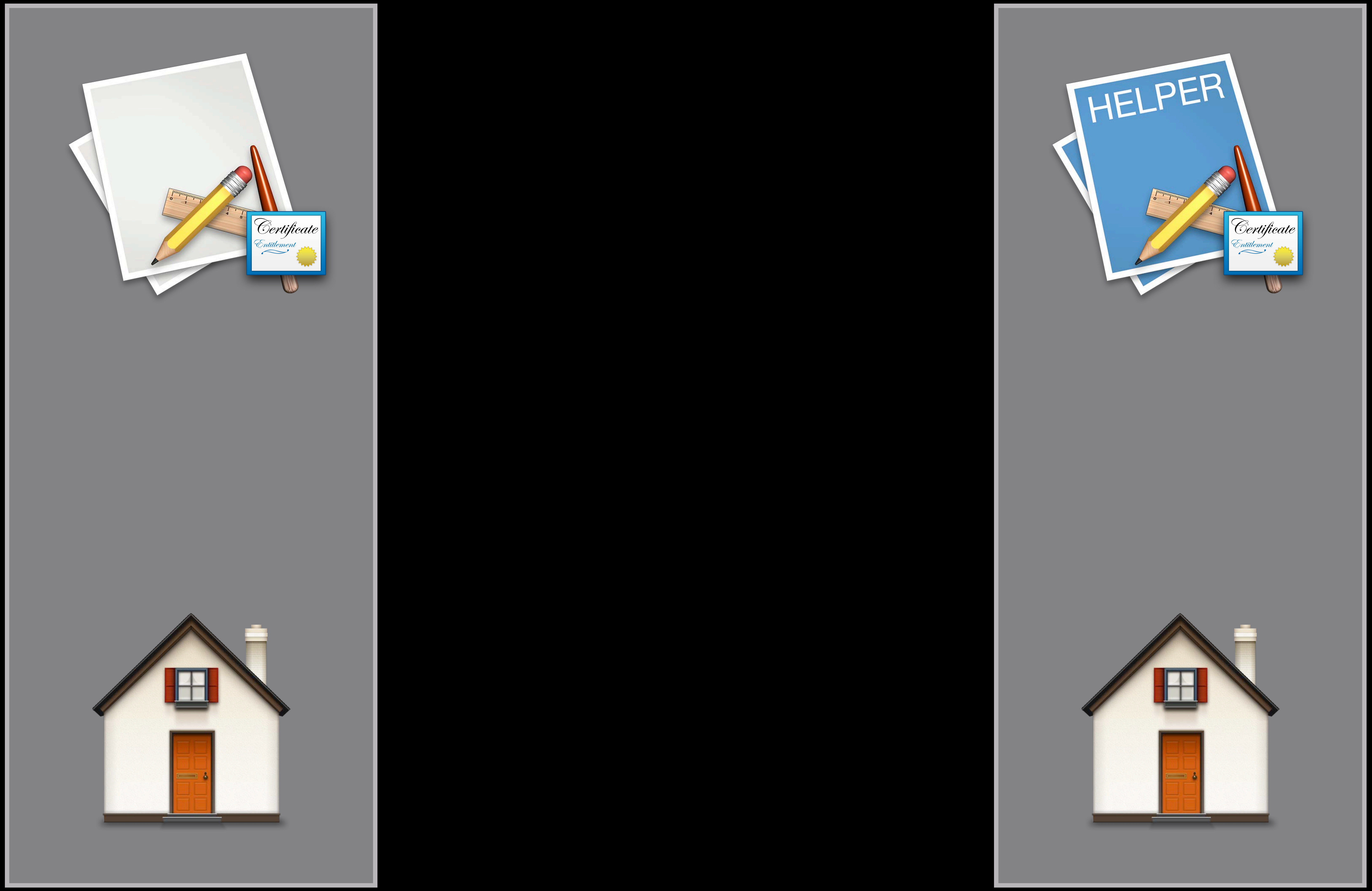

Certificate
Entitlement
HELPER

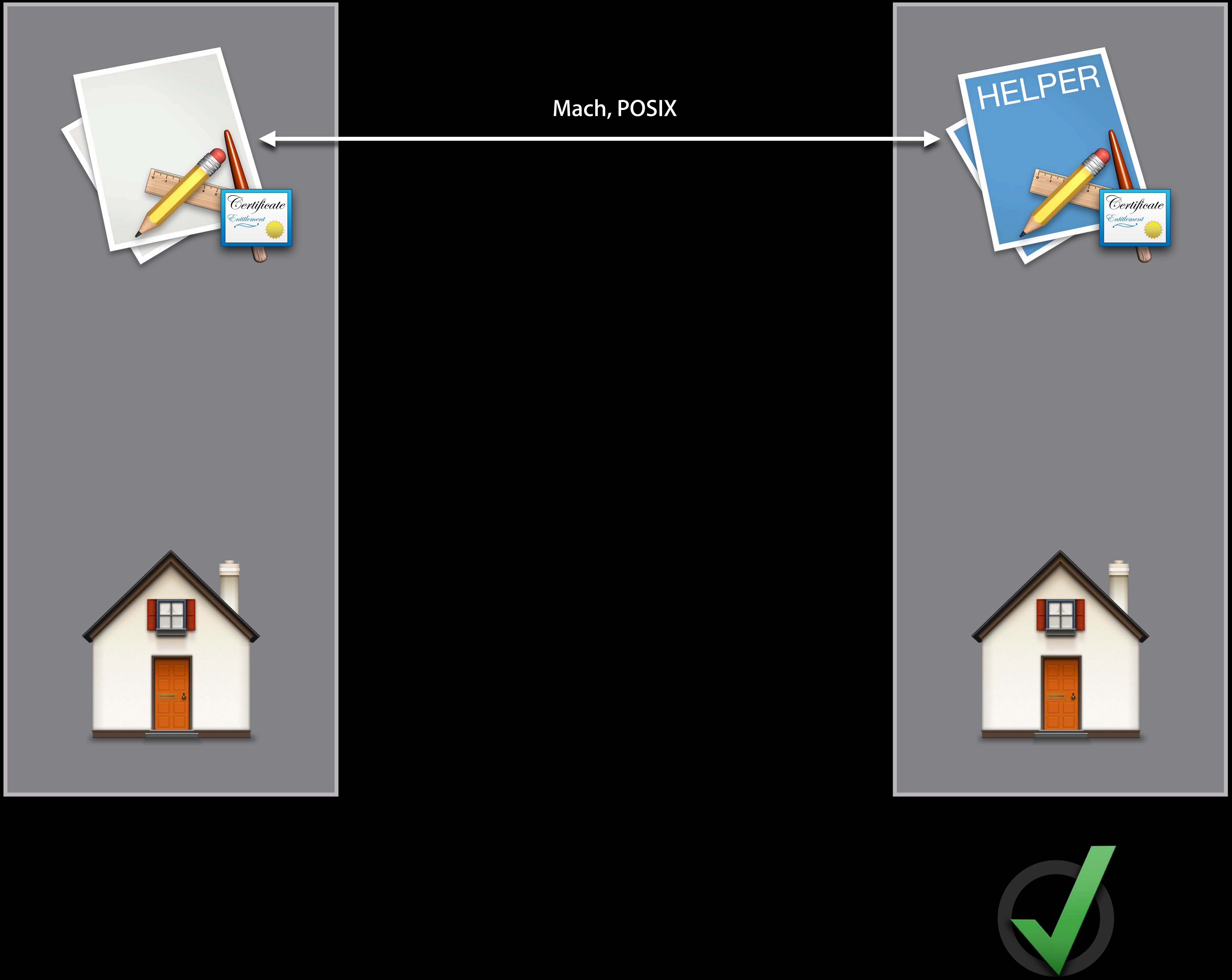

Certificate
Entitlement
HELPER
Certificate
Entitlement


Certificate
Entitlement
HELPER
Certificate
Entitlement

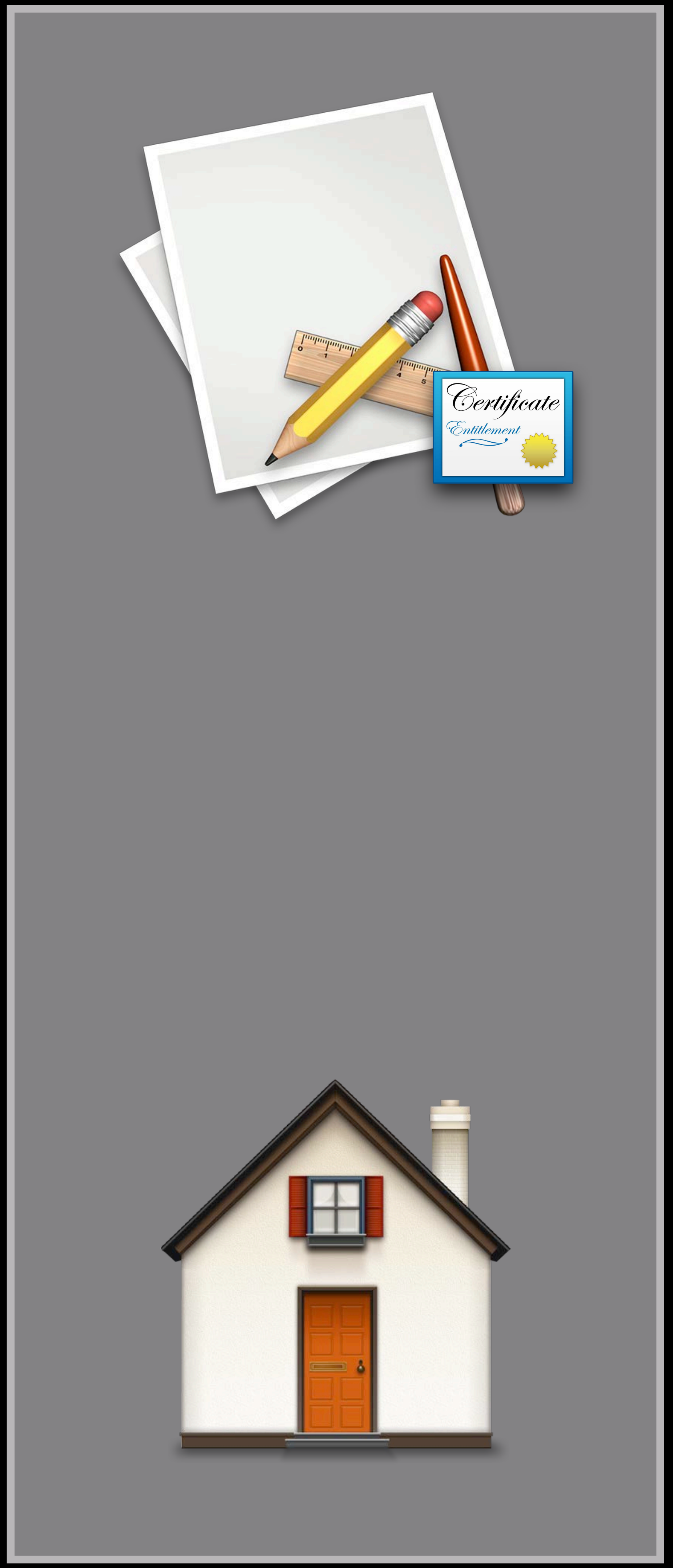

Certificate
Entitlement
8314ABCD.myapp
HELPER
Certificate
Entitlement

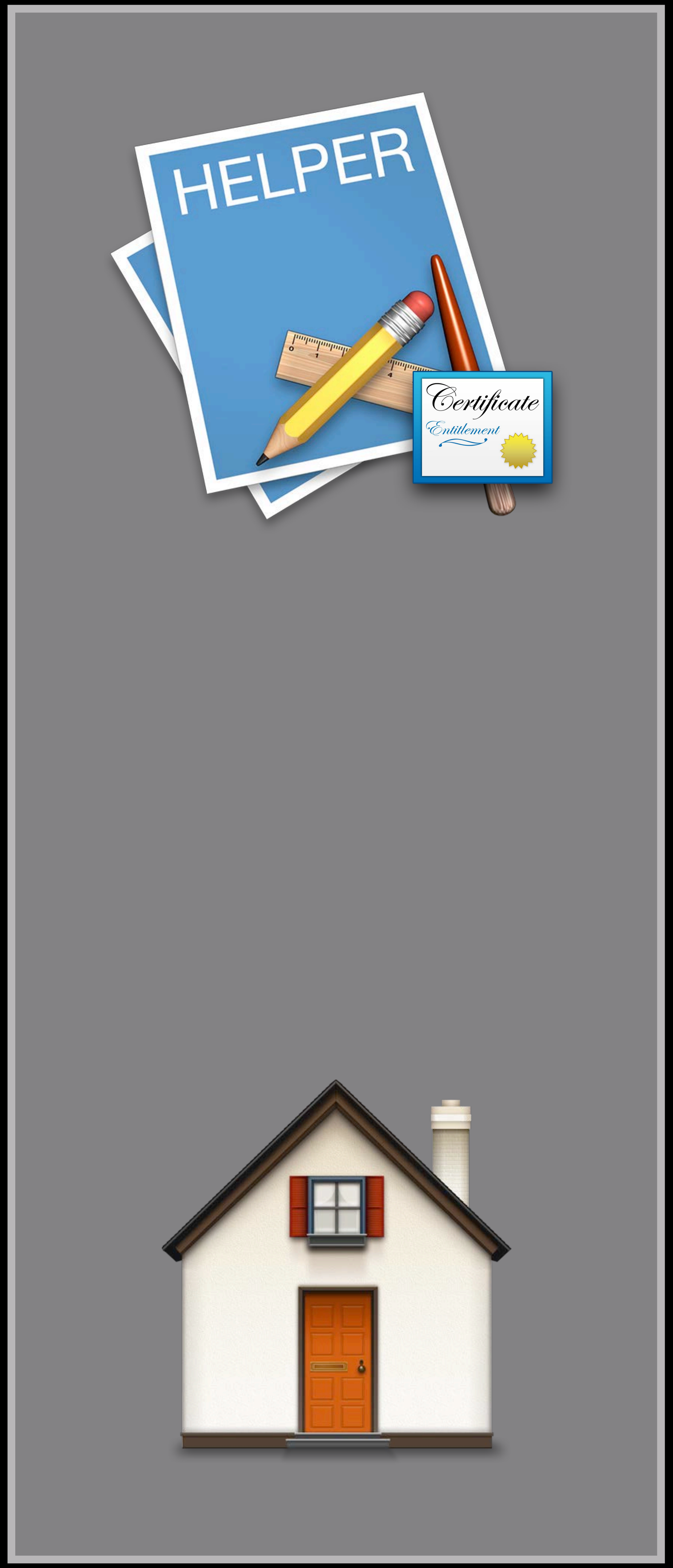

HELPER
Certificate
Entitlement
Certificate
Entitlement
8314ABCD.myapp

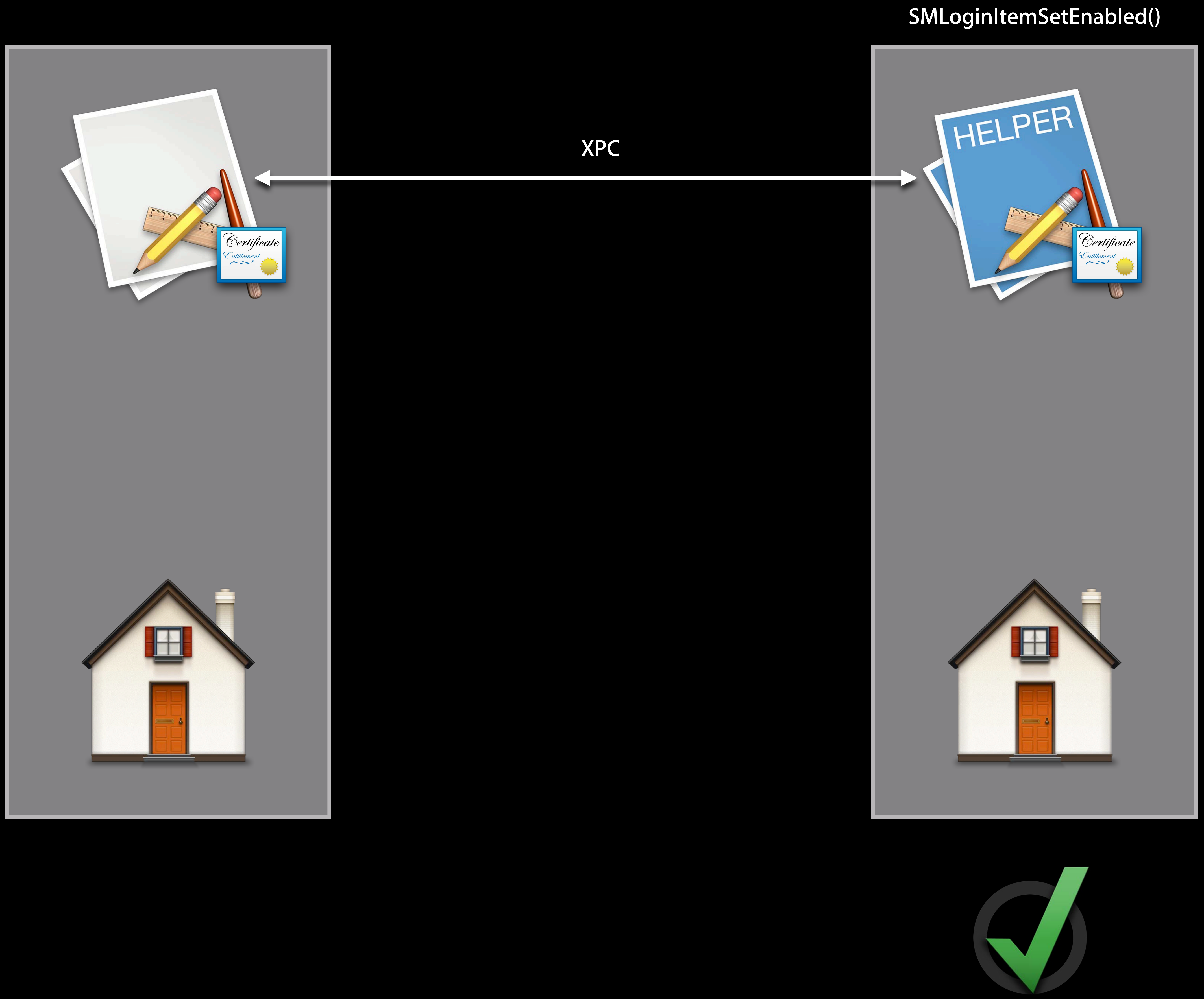

Certificate
Entitlement
HELPER
Certificate
Entitlement


Mach, POSIX
HELPER
Certificate
Entitlement
Certificate
Entitlement

SMLoginItemSetEnabled()






SMLoginItemSetEnabled()
XPC
HELPER
Certificate
Entitlement
Certificate
Entitlement

Related Items

Related Items

- Access to files/folders with same name, but different file extension
  - Movie player opening a subtitle file for a movie
  - TextEdit upgrading a .rtf document to a .rtfd for attachments
- NSFilePresenter's `primaryPresentedItemURL` for the former, `itemAtURL:willMoveToURL:` for the latter
- Requires a declaration of allowed patterns in the app's Info.plist

Automation

Automation

- Rich history of automation on OS X
- App Sandbox does not impose restrictions on how your apps can be scripted
- But your apps were very limited in how they can script other apps
    - Scripting Terminal, Finder or Safari can be complete sandbox escapes

Apple Event Access Groups

- Access groups define groups of scriptable operations
  - Commands, classes, properties
  - Part of the application's scripting interface (sdef)
  - man 5 sdef
- Already in OS X applications
  - Mail: com.apple.mail.compose
  - iTunes: com.apple.iTunes.playback, com.apple.iTunes.library.read, com.apple.iTunes.library.read-write

Using an Access Group

- com.apple.security.scripting-targets
- Value is a dictionary
    - Keys are application code signing identifiers
    - Values are access group identifiers

Using an Access Group

Compose Mail message

```
<key>com.apple.security.scripting-targets<key>
<dict>
    <key>com.apple.mail</key>
    <array>
        <string>com.apple.mail.compose<string>
    </array>
</dict>
```

Using an Access Group

Compose Mail message

```
<key>com.apple.security.scripting-targets<key>
<dict>
    <key>com.apple.mail</key>
    <array>
        <string>com.apple.mail.compose<string>
    </array>
</dict>
```

Using an Access Group

Compose Mail message

```
<key>com.apple.security.scripting-targets<key>
<dict>
    <key>com.apple.mail</key>
    <array>
        <string>com.apple.mail.compose<string>
    </array>
</dict>
```

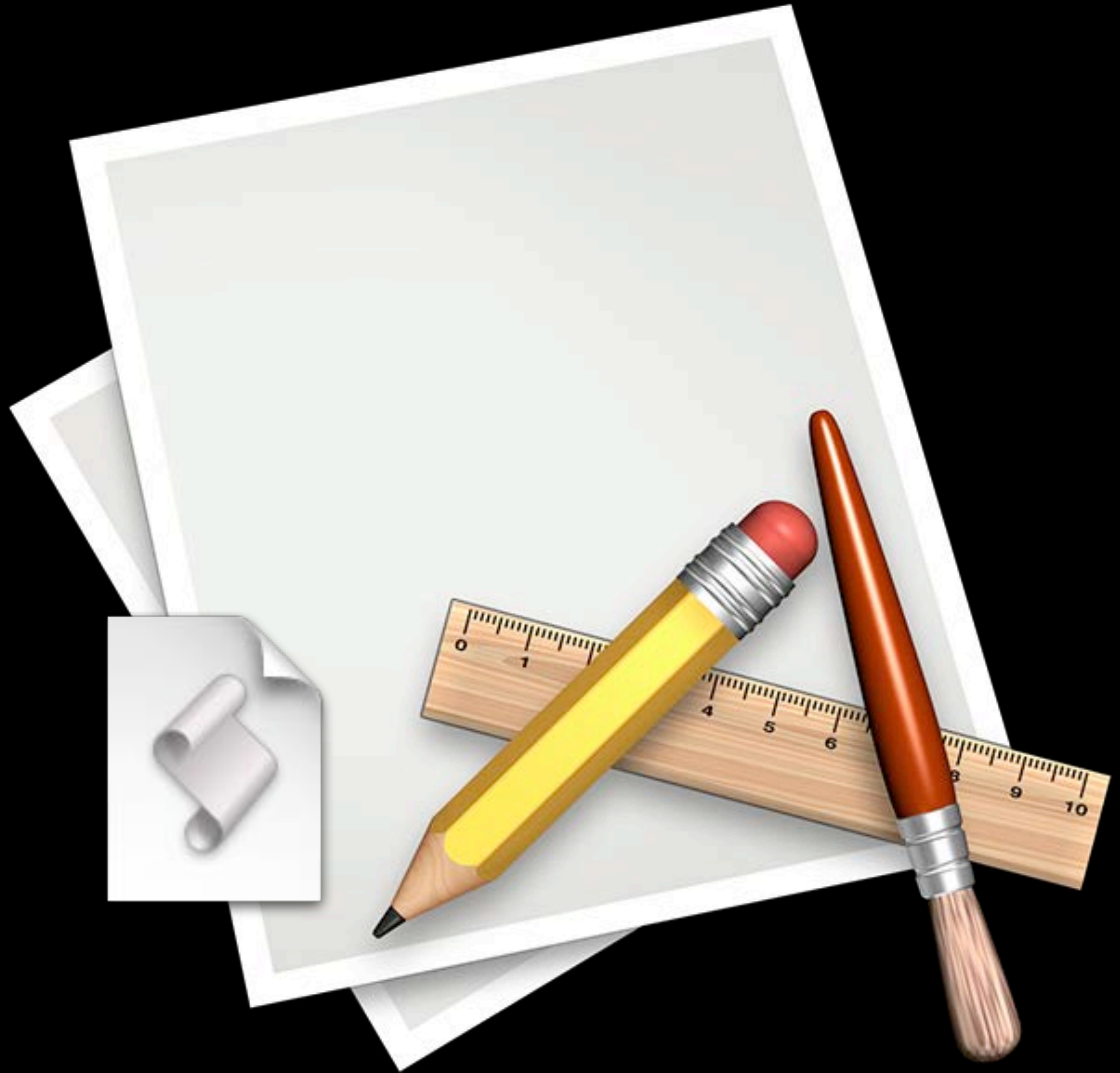
Using an Access Group

Compose Mail message

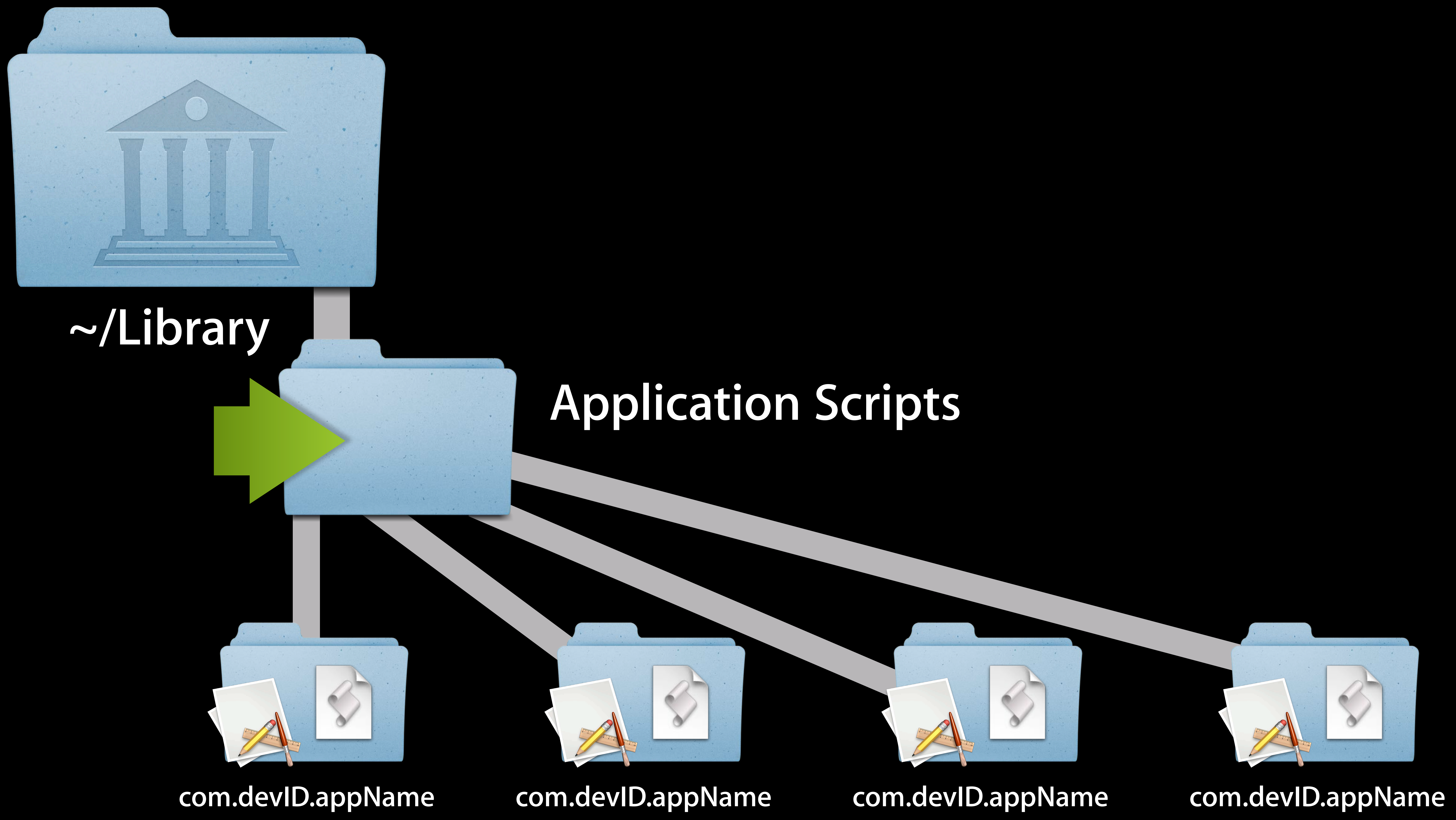
```
<key>com.apple.security.scripting-targets<key>
<dict>
    <key>com.apple.mail</key>
    <array>
        <string>com.apple.mail.compose<string>
    </array>
</dict>
```

Application-Run User Scripts

- Application Script Menu
- Event Handlers
    - Mail Rule
    - Aperture Import Action
    - Messages Events
- Scripts executed by the application
- Inherit application’s permissions



NSUserScriptTask



NSUserScriptTask

Running attached user scripts

- Part of Foundation.framework
- NSUserScriptTask for generic scripts
    - Supports AppleScript, Automator, and UNIX scripts
- Subclasses for specific control
    - NSUserAppleScriptTask, NSUserAutomatorTask, NSUserUnixTask
- Script runs outside the sandbox
- No entitlement required

iTunes Library Access

iTunes Library Framework

New in iTunes 11

- Access to iTunes Library media and artwork regardless of disk location
- Objective-C API instead of the XML database
- Requires `com.apple.security.music.read-{write,only}` entitlement
- Returns security-scoped NSURLs
  - `{start,stop}AccessingSecurityScopedResource`

App Sandbox and the Mac App Store

Mac App Store

- Technical Q&A QA1773
- All binaries must be sandboxed, including XPC services and other helper tools
- Entitlements must match app functionality
  - If you don’t need it, don’t request it
  - Don’t request entitlements that silence sandbox violations which have no functional impact

Mac App Store

- Understand the entitlements you're requesting
    - USB access not required for the user to choose files on USB media
    - Incoming connections (Server) not needed for most network applications
- Temporary exception requests must not effectively disable the sandbox
    - Scripting Finder or Terminal
    - Filesystem access to /

Summary

App Sandbox

- Strong barrier against exploitation and coding errors
- Drives policy by user intent
- Complementary to Gatekeeper
- See the App Sandbox Design Guide
- Sample code available

Summary

- iOS—50 billion sandboxed apps downloaded with confidence
- Delight users with carefree apps on OS X

Related Sessions

Efficient Design with XPC	Russian Hill Tuesday 2:00PM

Related Labs

OS X Sandbox Lab	Core OS Lab Wednesday 3:15PM
Security Lab	Core OS Lab Thursday 2:00PM
App Store Lab	Third Floor Daily 9:00AM

WWDC2013