Frameworks

#WWDC14

Cloud Documents in Your Application

Building a modern document-based application

Session 234
Mike Hess
Software Engineer

Johannes Fortmann
Software Engineer

Changes to iCloud

Changes to iCloud

iCloud daemon rewritten on top of CloudKit

Changes to iCloud

iCloud daemon rewritten on top of CloudKit

Windows support

Changes to iCloud

iCloud daemon rewritten on top of CloudKit

Windows support

iCloud folder in Finder

Changes to iCloud

- iCloud daemon rewritten on top of CloudKit
- Windows support
- iCloud folder in Finder
- Access to iCloud folder on iOS

Agenda

Agenda

Document access

Agenda

Document access

Document discovery

Agenda

Document access

Document discovery

Displaying thumbnails in your UI

Agenda

Document access

Document discovery

Displaying thumbnails in your UI

Accessing documents outside your iCloud container

Agenda

Document access

Document discovery

Displaying thumbnails in your UI

Accessing documents outside your iCloud container

Providing document storage to other apps

Documents in iCloud

What is a document anyway?

Documents in iCloud

What is a document anyway?

A standalone entity, understood as such by the user

Documents in iCloud

What is a document anyway?

A standalone entity, understood as such by the user

User might want to exchange (e.g., move it, modify it, send it via email)

Documents in iCloud
What is a document anyway?

A standalone entity, understood as such by the user

User might want to exchange (e.g., move it, modify it, send it via email)

Documents in iCloud

What is a document anyway?

A standalone entity, understood as such by the user

User might want to exchange (e.g., move it, modify it, send it via email)

Documents in iCloud

What is a document anyway?

A standalone entity, understood as such by the user

User might want to exchange (e.g., move it, modify it, send it via email)

Dealing with Documents

Document Access—Best Practices

Reading and writing documents

Document Access—Best Practices

Reading and writing documents

Important—use file coordination!

Document Access—Best Practices

Reading and writing documents

Important—use file coordination!

Why? Avoid data loss!

Document Access—Best Practices

Reading and writing documents

Important—use file coordination!

Why? Avoid data loss!

There may be multiple readers/writers on the same document

Document Access—Best Practices

Reading and writing documents

Important—use file coordination!

Why? Avoid data loss!

There may be multiple readers/writers on the same document

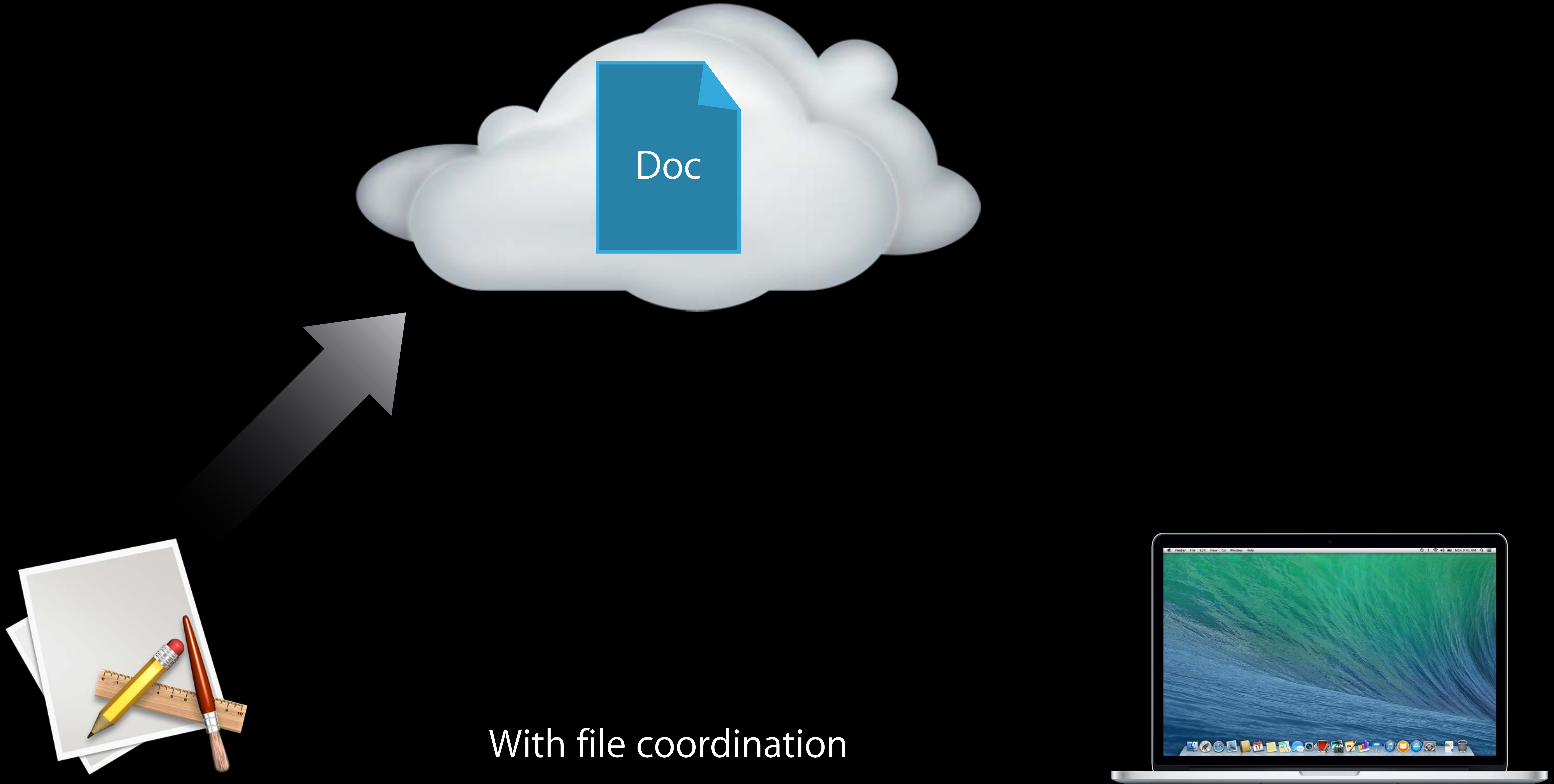
- Examples—iCloud, background saving, etc.

Document Access—Best Practices

Reading and writing documents

Document Access—Best Practices

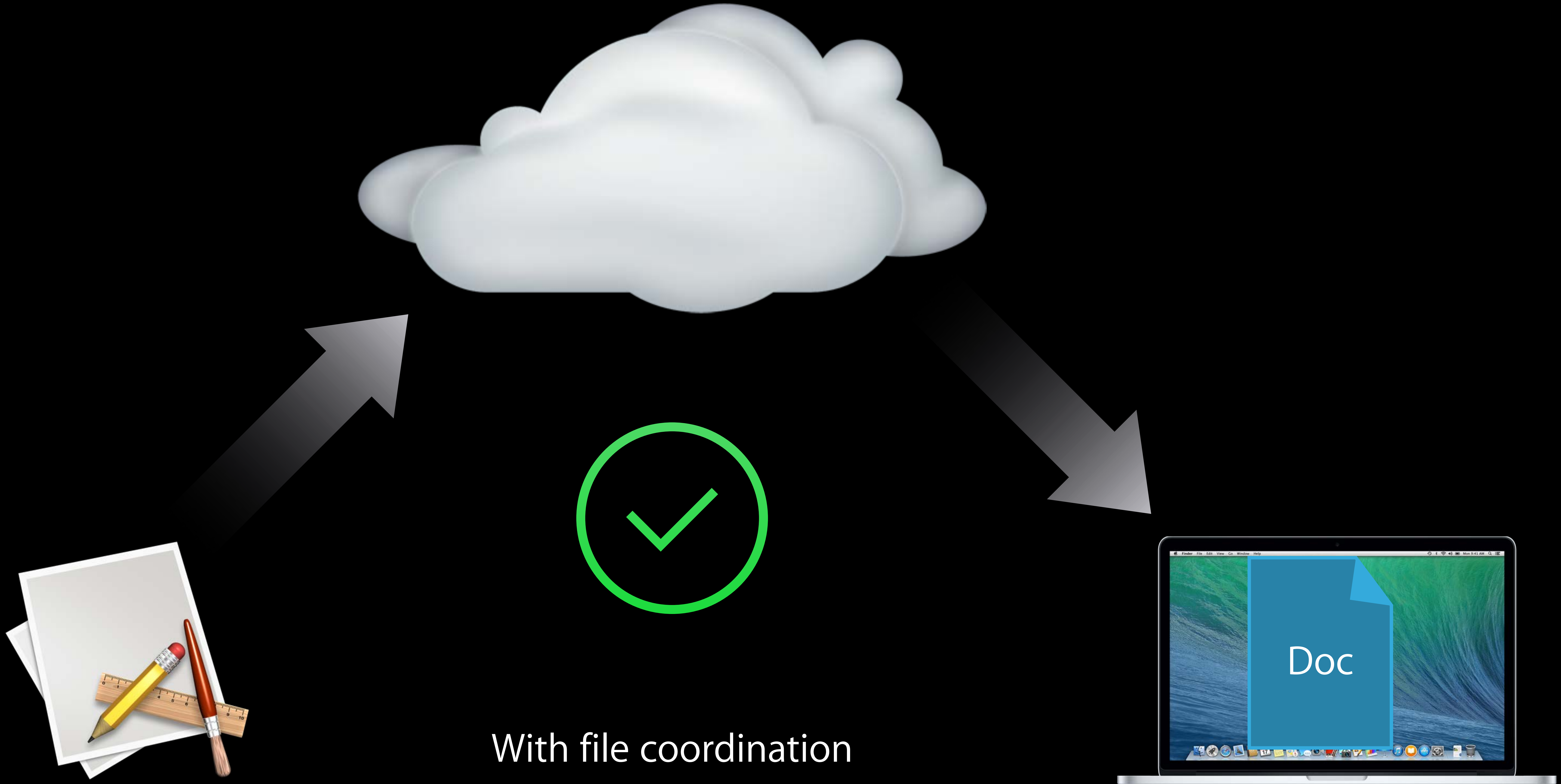
Reading and writing documents



With file coordination

Document Access—Best Practices

Reading and writing documents



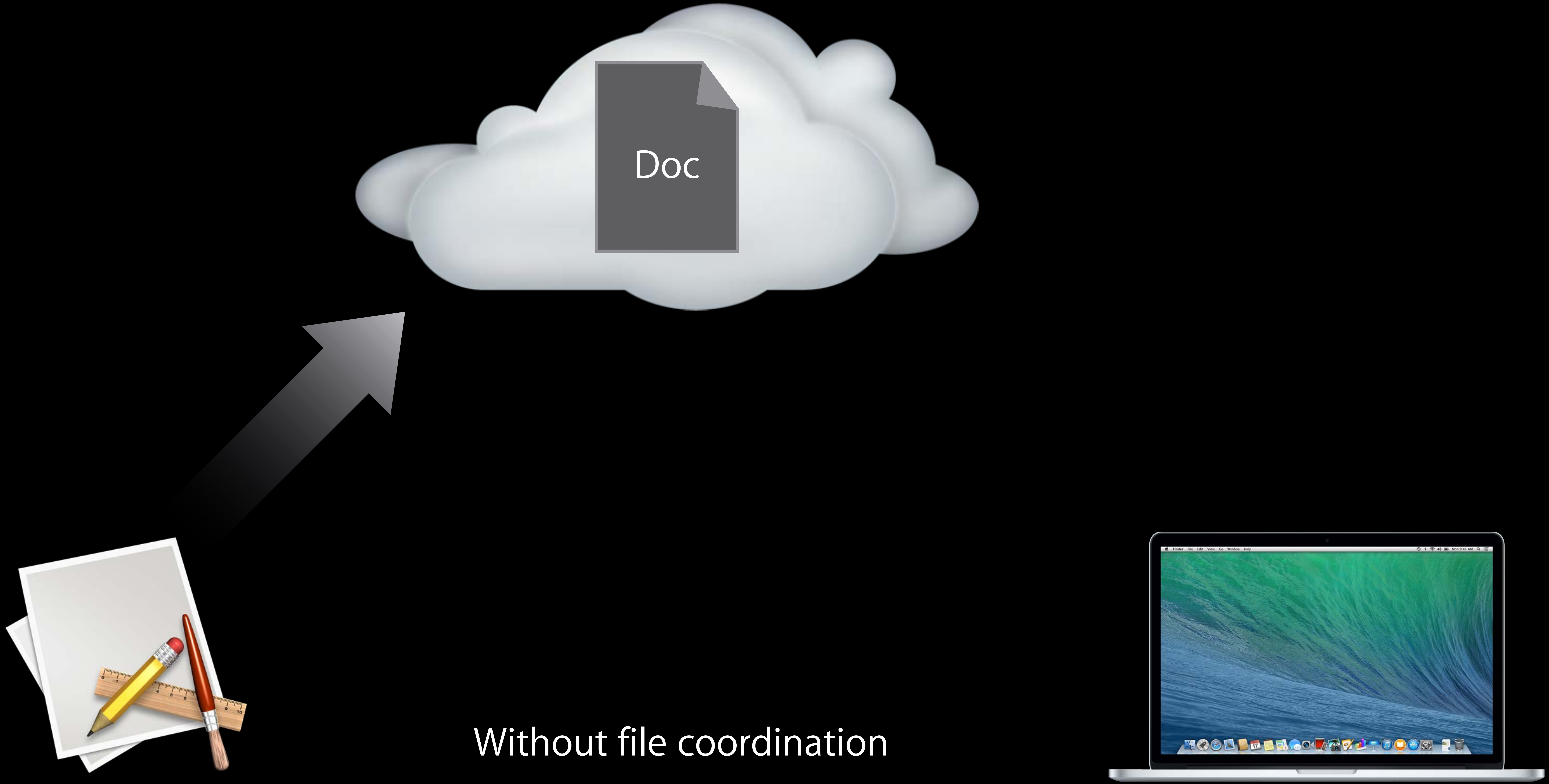
With file coordination

Document Access—Best Practices

Reading and writing documents

Document Access—Best Practices

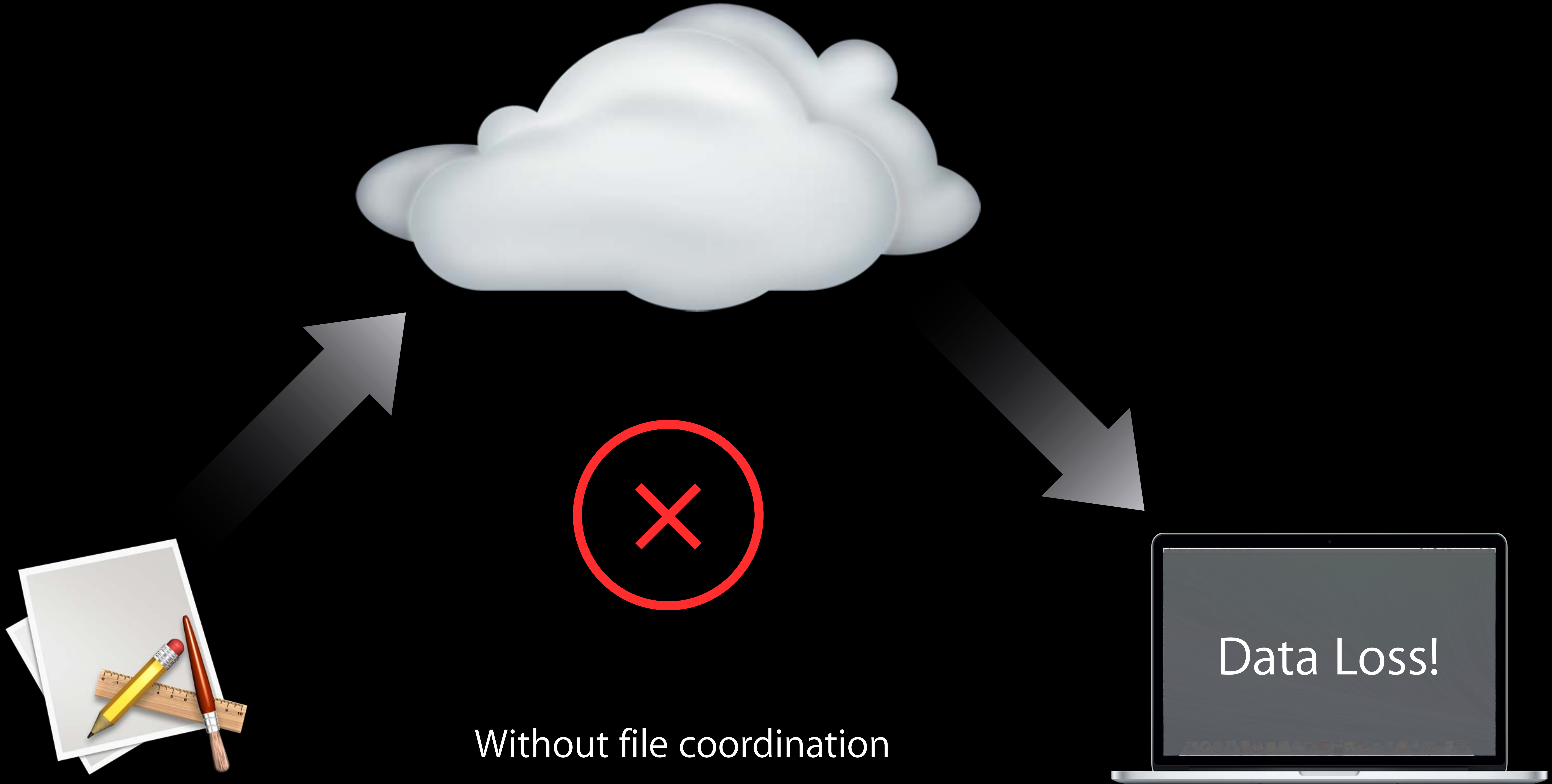
Reading and writing documents



Without file coordination

Document Access—Best Practices

Reading and writing documents



Document Access—Best Practices

UIDocument makes things simple!

Document Access—Best Practices

UIDocument makes things simple!

UIDocument/NSDocument do all heavy lifting for you

Document Access—Best Practices

UIDocument makes things simple!

UIDocument/NSDocument do all heavy lifting for you

- File coordination

Document Access—Best Practices

UIDocument makes things simple!

UIDocument/NSDocument do all heavy lifting for you

- File coordination
- Background queues

Document Access—Best Practices

UIDocument makes things simple!

UIDocument/NSDocument do all heavy lifting for you

- File coordination
- Background queues
- Multiple high-level APIs for every purpose

Document Access—Best Practices

Example—reading a document from disk

Highest level—we read the file, you get an NSData or NSFileWrapper object

```
- loadFromContents:ofType:error:
```

Document Access—Best Practices

Example—reading a document from disk

Highest level—we read the file, you get an NSData or NSFileWrapper object

```
- loadFromContents:ofType:error:
```

Intermediate level—we coordinate, you get an NSURL

```
- readFromURL:error:
```

Document Access—Best Practices

Example—reading a document from disk

Highest level—we read the file, you get an NSData or NSFileWrapper object

`– loadFromContents:ofType:error:`

Intermediate level—we coordinate, you get an NSURL

`– readFromURL:error:`

Lowest level—provided for hooking

`– openWithCompletionHandler:`

Asynchronous File Coordination

New File Coordination APIs

Asynchronous File Coordination

New File Coordination APIs

Existing File Coordination APIs were synchronous

Asynchronous File Coordination

New File Coordination APIs

NEW

Existing File Coordination APIs were synchronous

New File Coordinator API for asynchronous file coordination access

Asynchronous File Coordination

New File Coordination APIs

NEW

Existing File Coordination APIs were synchronous

New File Coordinator API for asynchronous file coordination access

`- coordinateAccessWithIntents:queue:byAccessor:`

Asynchronous File Coordination

New File Coordination APIs

NEW

Existing File Coordination APIs were synchronous

New File Coordinator API for asynchronous file coordination access

`– coordinateAccessWithIntents:queue:byAccessor:`

Use NSFileAccessIntent to specify purpose of coordination

Asynchronous File Coordination

New File Coordination APIs

Existing File Coordination APIs were synchronous

New File Coordinator API for asynchronous file coordination access

```
– coordinateAccessWithIntents:queue:byAccessor:
```

Use NSFileAccessIntent to specify purpose of coordination

```
srcInt = [NSFileAccessIntent readingIntentWithURL:srcURL options:srcOpts];
dstInt = [NSFileAccessIntent writingIntentWithURL:dstURL options:dstOpts];
[fileCoordinator coordinateAccessWithIntents:@[src, dst] queue:queue
    byAccessor:^(NSError *error) {
        // Do operation here with [srcInt URL] and [dstInt URL]
    }];
```

Asynchronous File Coordination

New File Coordination APIs

Existing File Coordination APIs were synchronous

New File Coordinator API for asynchronous file coordination access

```
– coordinateAccessWithIntents:queue:byAccessor:
```

Use NSFileAccessIntent to specify purpose of coordination

```
srcInt = [NSFileAccessIntent readingIntentWithURL:srcURL options:srcOpts];
dstInt = [NSFileAccessIntent writingIntentWithURL:dstURL options:dstOpts];
[fileCoordinator coordinateAccessWithIntents:@[src, dst] queue:queue
    byAccessor:^(NSError *error) {
        // Do operation here with [srcInt URL] and [dstInt URL]
    }];
```

Asynchronous File Coordination

New File Coordination APIs

NEW

Existing File Coordination APIs were synchronous

New File Coordinator API for asynchronous file coordination access

```
- coordinateAccessWithIntents:queue:byAccessor:
```

Use NSFileAccessIntent to specify purpose of coordination

```
srcInt = [NSFileAccessIntent readingIntentWithURL:srcURL options:srcOpts];
dstInt = [NSFileAccessIntent writingIntentWithURL:dstURL options:dstOpts];
[fileCoordinator coordinateAccessWithIntents:@[src, dst] queue:queue
    byAccessor:^(NSError *error) {
        // Do operation here with [srcInt URL] and [dstInt URL]
    }];
```

Asynchronous File Coordination

New File Coordination APIs

Existing File Coordination APIs were synchronous

New File Coordinator API for asynchronous file coordination access

```
- coordinateAccessWithIntents:queue:byAccessor:
```

Use NSFileAccessIntent to specify purpose of coordination

```
srcInt = [NSFileAccessIntent readingIntentWithURL:srcURL options:srcOpts];
dstInt = [NSFileAccessIntent writingIntentWithURL:dstURL options:dstOpts];
[fileCoordinator coordinateAccessWithIntents:@[src, dst] queue:queue
    byAccessor:^(NSError *error) {
        // Do operation here with [srcInt URL] and [dstInt URL]
    }];
```

Discovering and Listing Documents

Document Discovery

Discovering new and existing documents

Document Discovery

Discovering new and existing documents

NSMetadataQuery APIs

Document Discovery

Discovering new and existing documents

NSMetadataQuery APIs

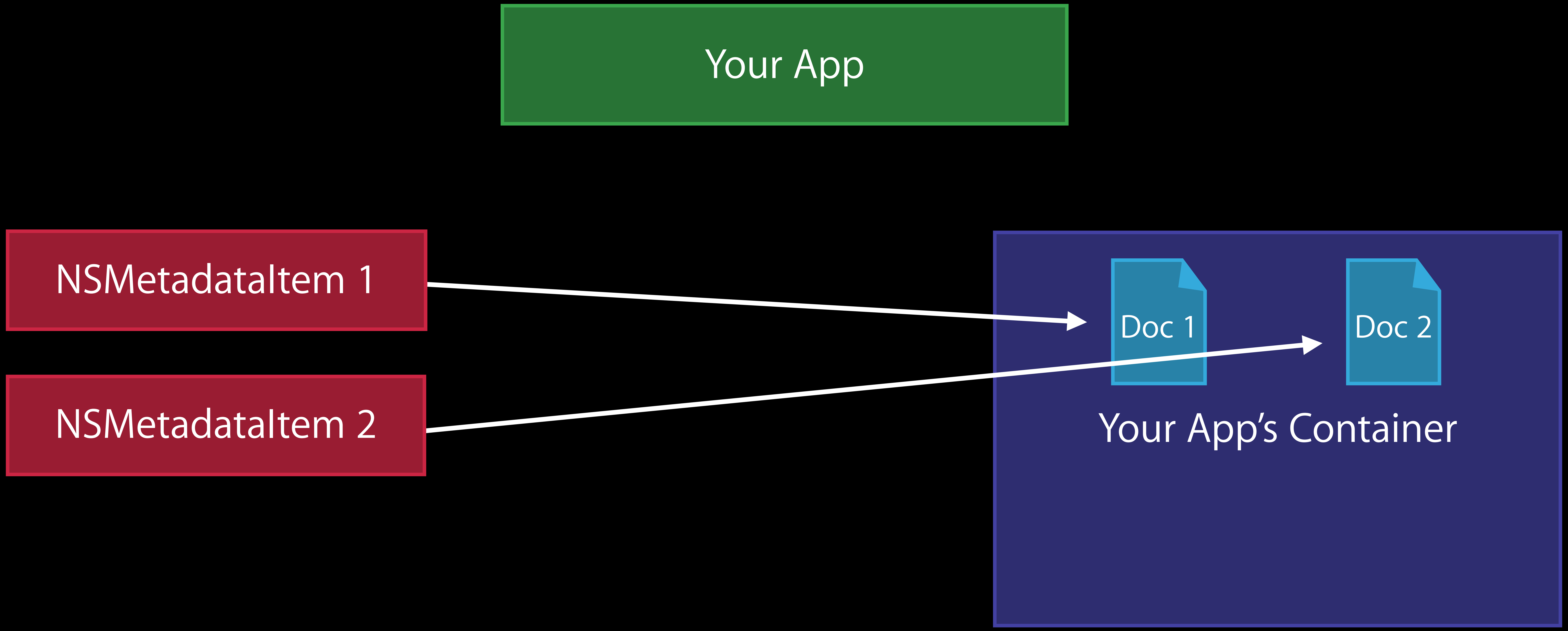
Previously slow to pick up local changes (deletes, creates, renames)

Document Discovery

Discovering new and existing documents

NSMetadataQuery APIs

Previously slow to pick up local changes (deletes, creates, renames)

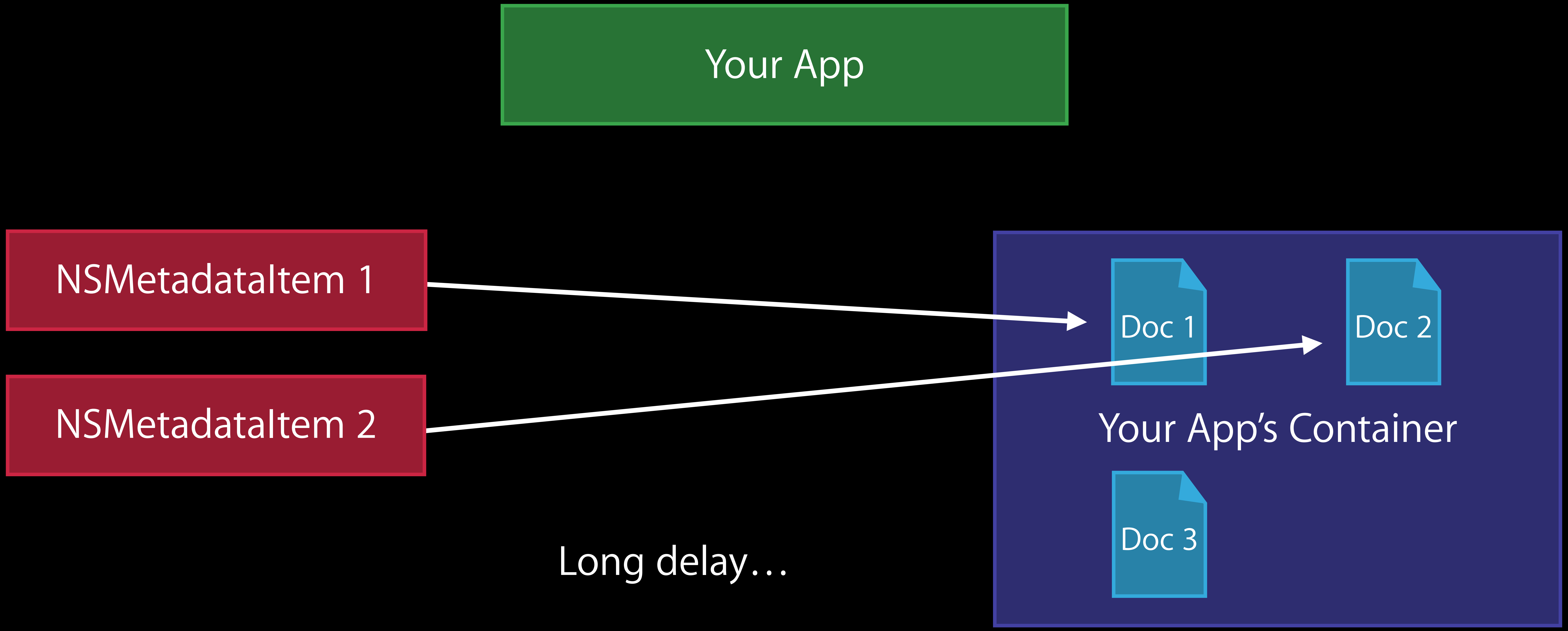


Document Discovery

Discovering new and existing documents

NSMetadataQuery APIs

Previously slow to pick up local changes (deletes, creates, renames)

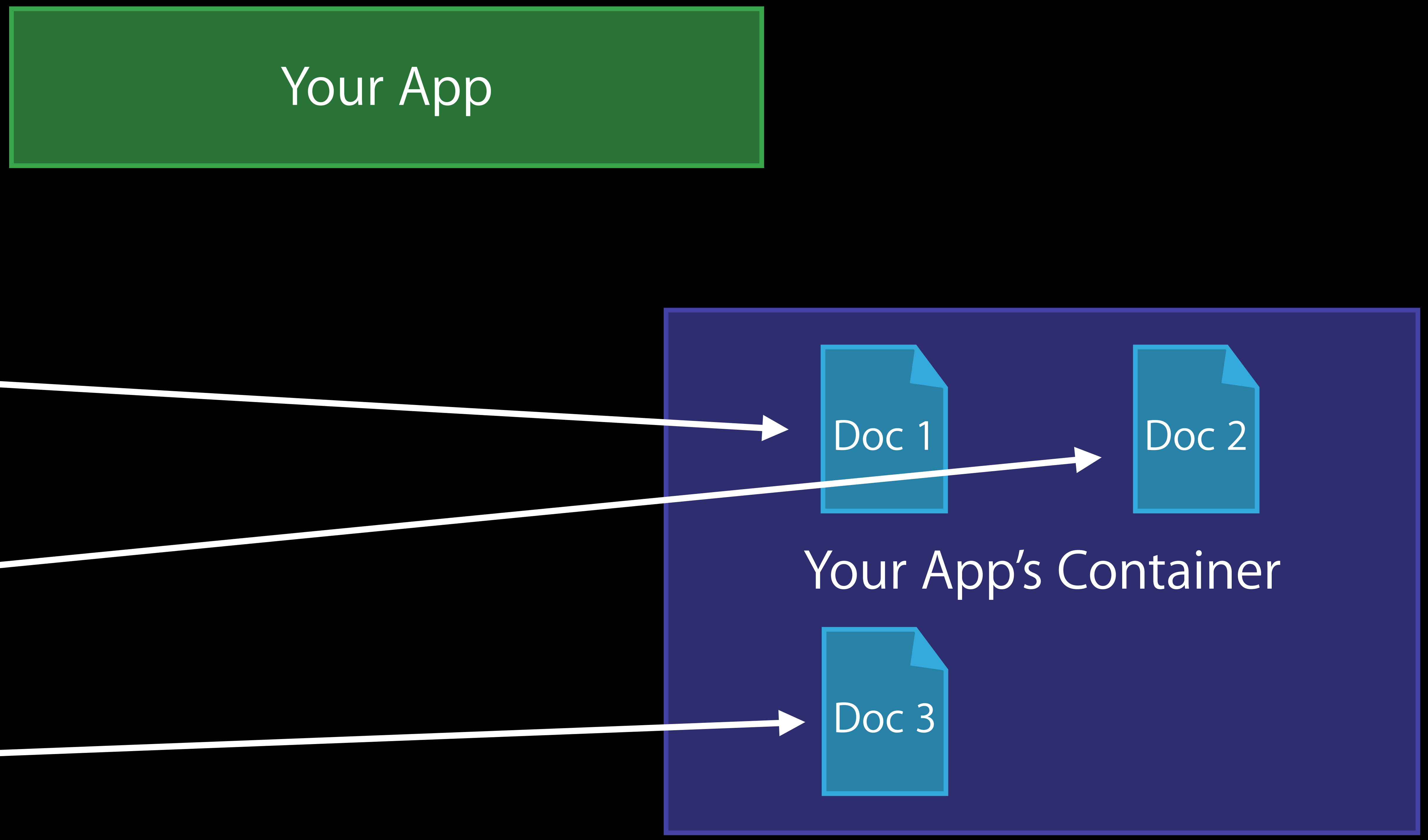


Document Discovery

Discovering new and existing documents

NSMetadataQuery APIs

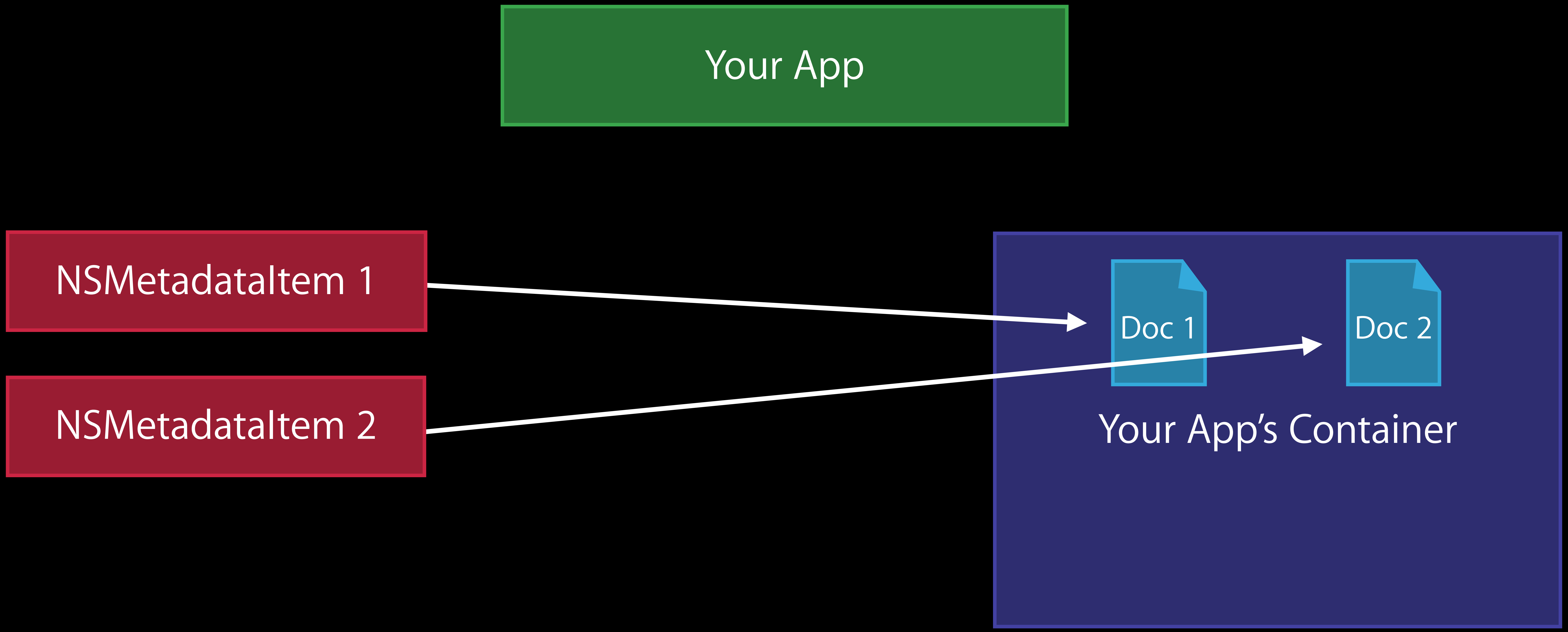
Previously slow to pick up local changes (deletes, creates, renames)



Document Discovery

Discovering new and existing documents

NEW

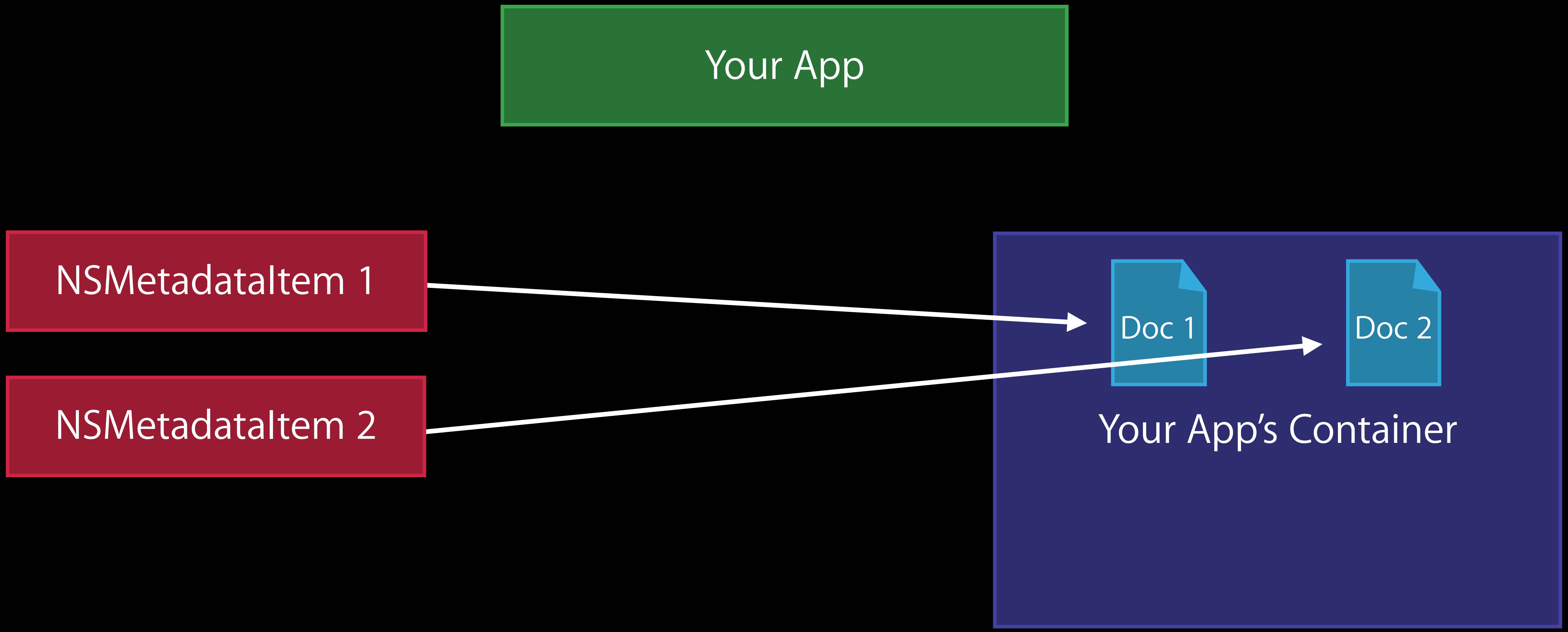


Document Discovery

Discovering new and existing documents

NEW

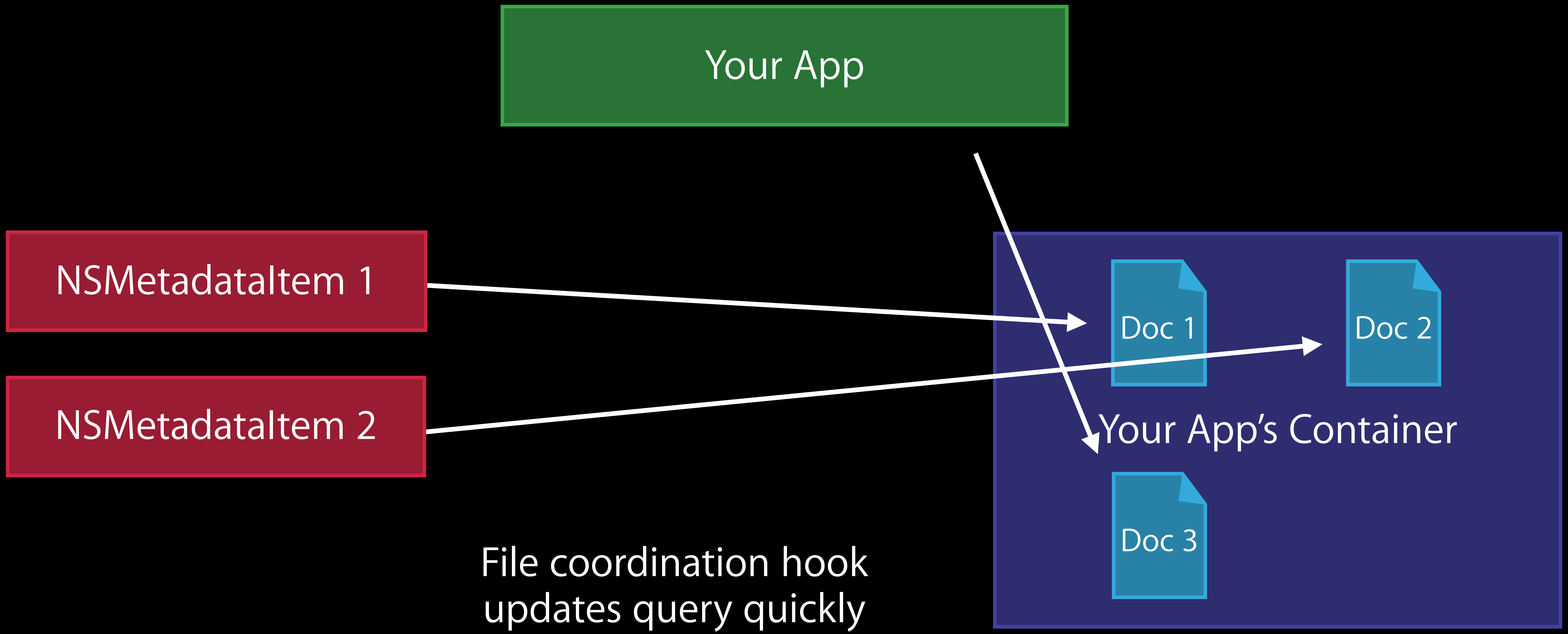
NSMetadataQuery APIs easier to use directly with “stitching”



Document Discovery

Discovering new and existing documents

NSMetadataQuery APIs easier to use directly with “stitching”

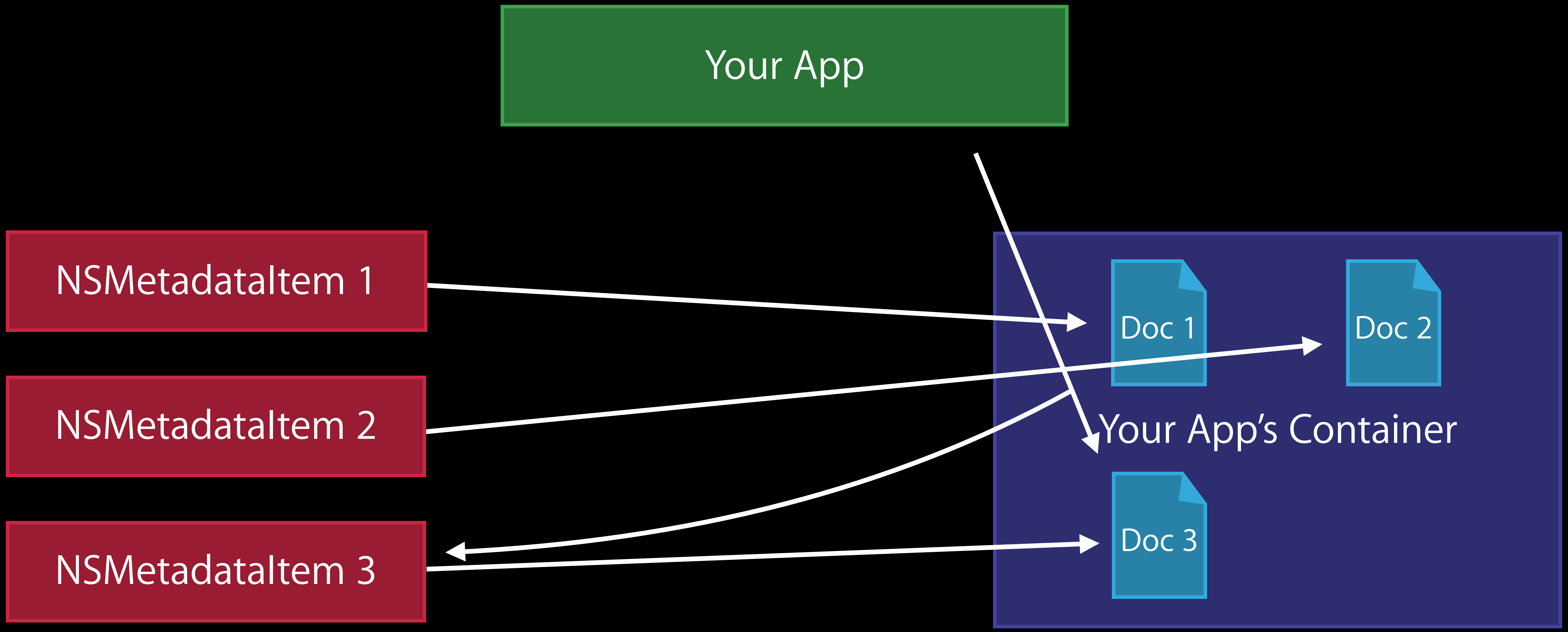


Document Discovery

Discovering new and existing documents

NEW

NSMetadataQuery APIs easier to use directly with “stitching”



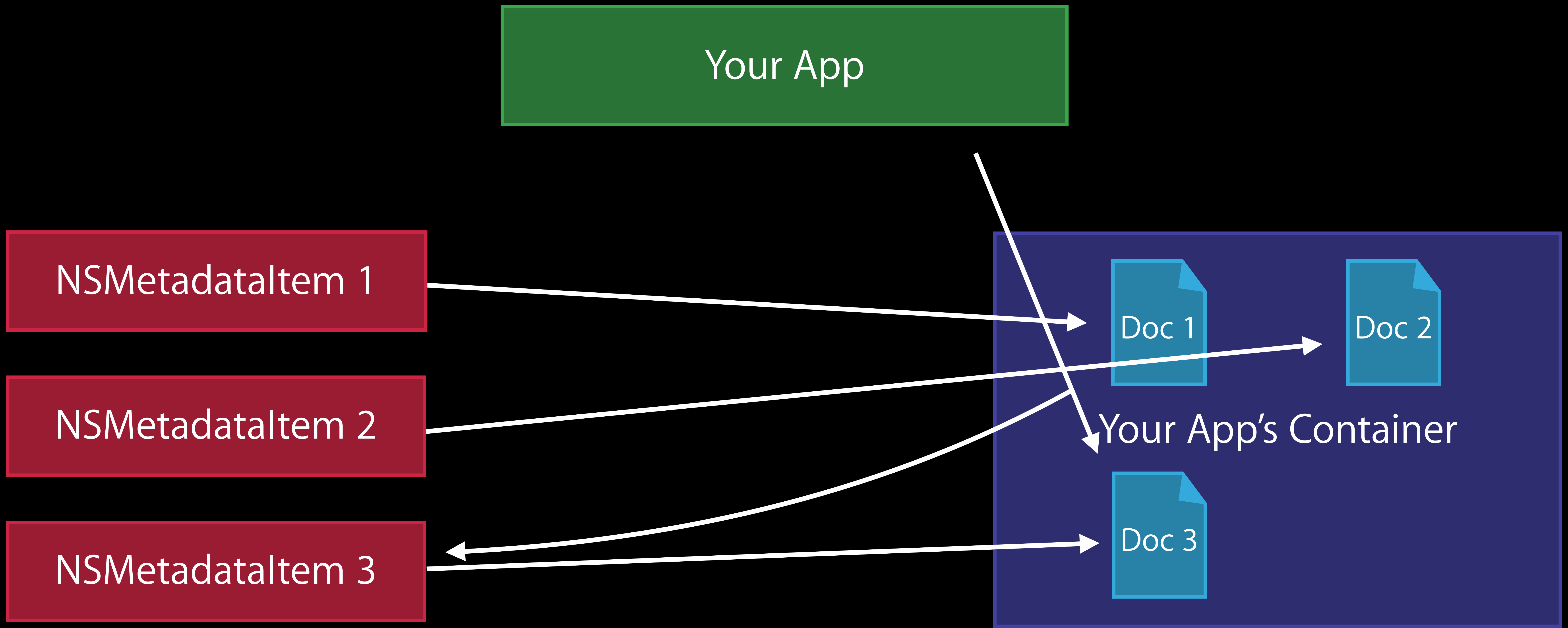
Document Discovery

Discovering new and existing documents

NEW

NSMetadataQuery APIs easier to use directly with “stitching”

No file coordination required when listing documents



Document Discovery

Accessing metadata on documents

NSMetadataQuery can list non-local documents

New APIs to access metadata on those documents

```
– getPromisedItemResourceValue:forKey:error:
```

Document Discovery

Accessing metadata on documents

NEW

NSMetadataQuery can list non-local documents

New APIs to access metadata on those documents

```
– getPromisedItemResourceValue:forKey:error:
– promisedItemResourceValuesForKeys:error:
```

Document Discovery

Accessing metadata on documents

NEW

NSMetadataQuery can list non-local documents

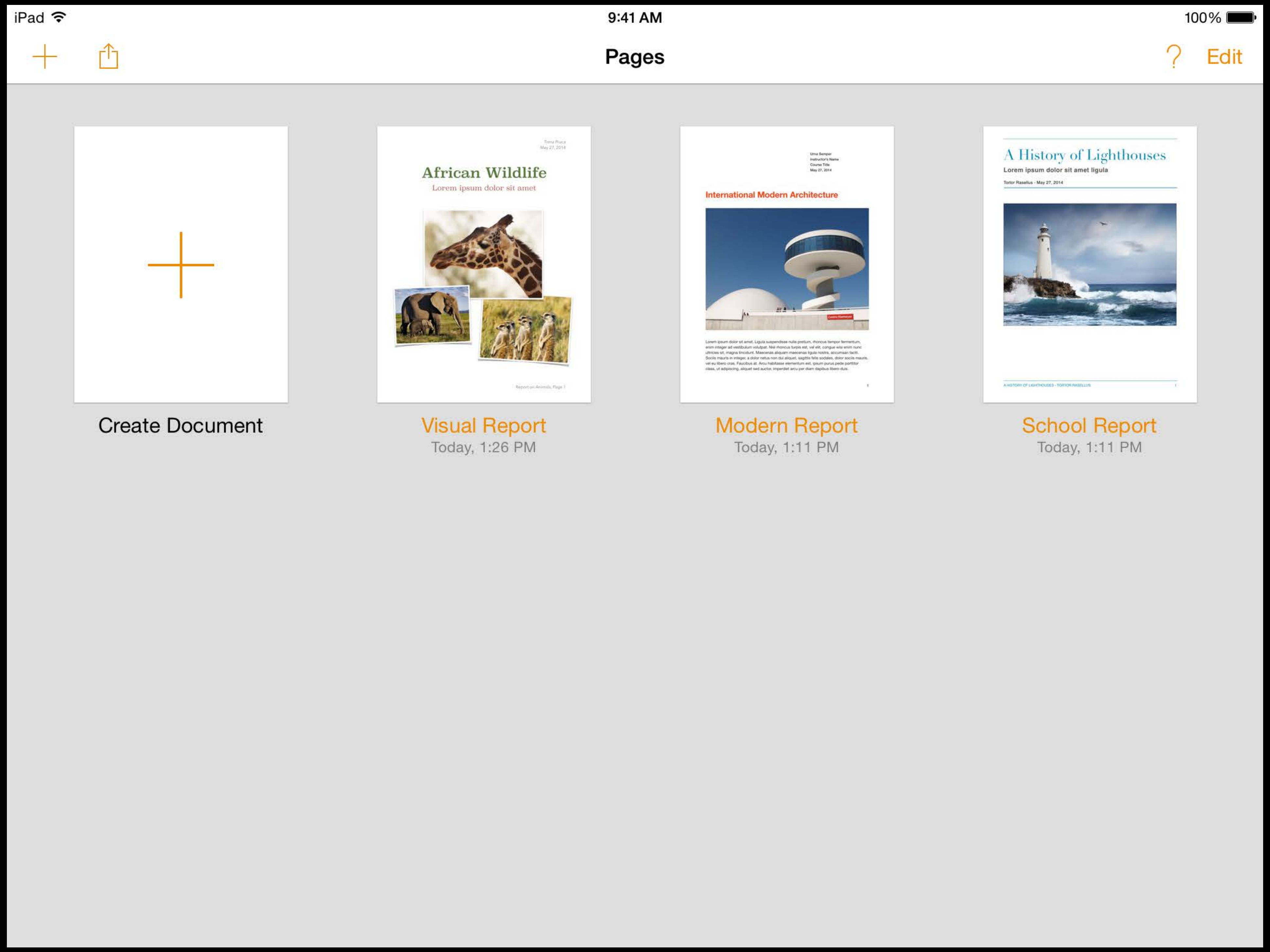
New APIs to access metadata on those documents

```
– getPromisedItemResourceValue:forKey:error:
– promisedItemResourceValuesForKeys:error:
```

New file coordinator flags for dealing with metadata

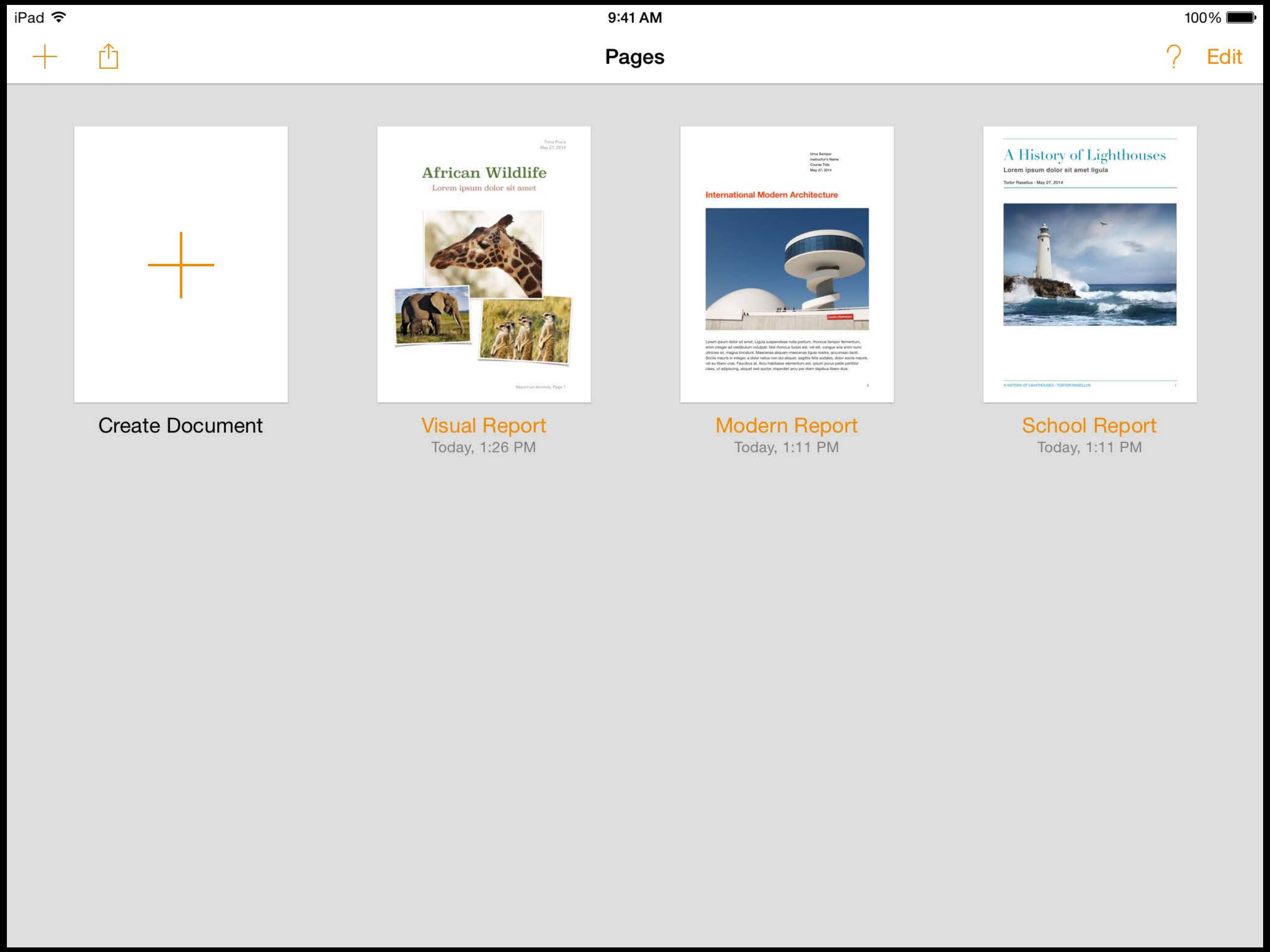
```
    NSFileCoordinatorReadingImmediatelyAvailableMetadataOnly
    NSFileCoordinatorWritingContentIndependentMetadataOnly
```

Document Thumbnails



Document Thumbnails

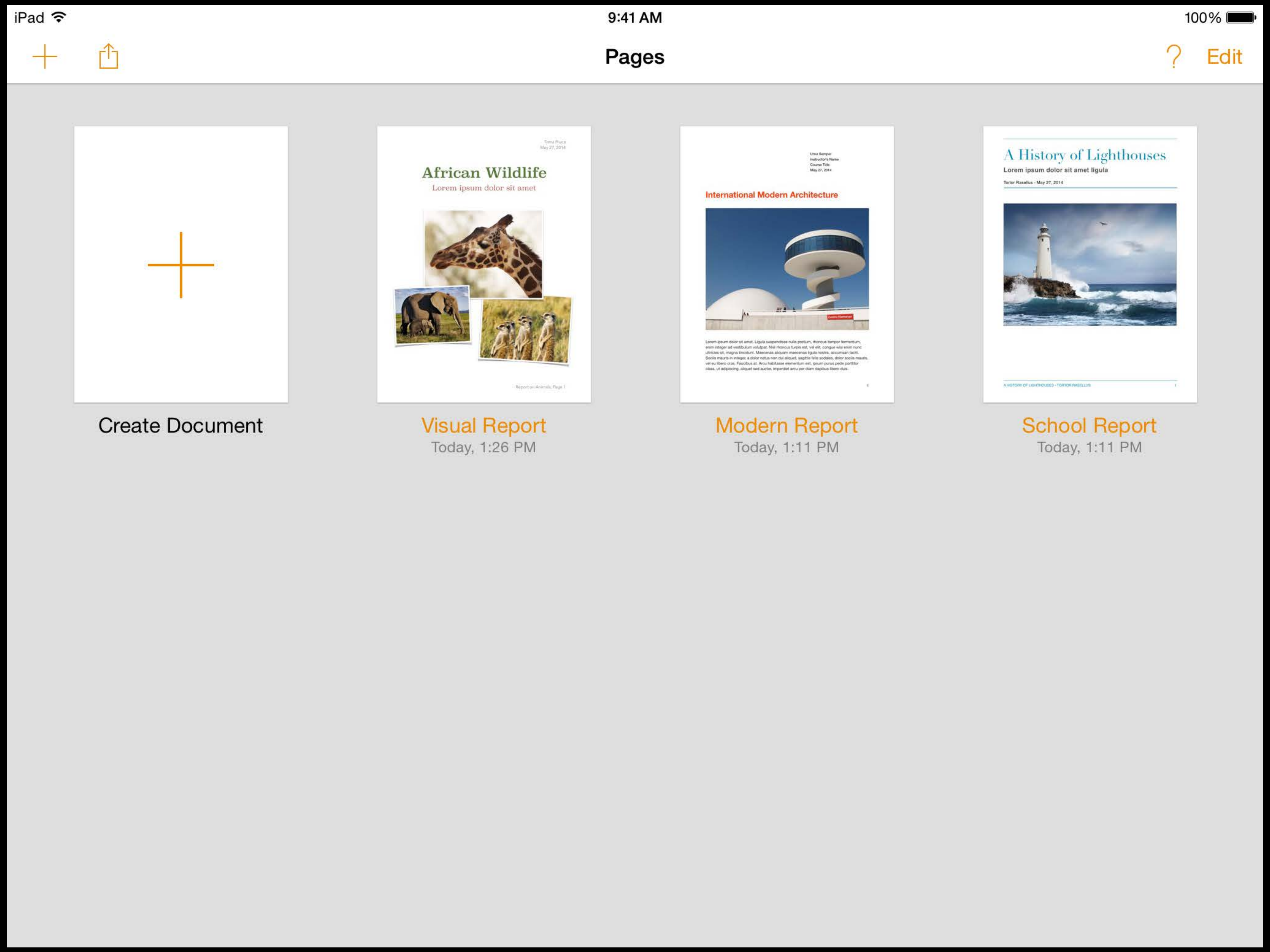
Best user experience when listing documents is to show previews



Document Thumbnails

Best user experience when listing documents is to show previews

Possible previously but required custom implementation



Document Thumbnails

Displaying thumbnails in your UI

NEW

Document Thumbnails

Displaying thumbnails in your UI

NEW

New thumbnail keys available!

- NSURLThumbnailDictionaryKey – NSDictionary
  - NSThumbnail1024x1024SizeKey – UIImage / NSImage
- NSURLThumbnailKey (OSX) – NSImage

Document Thumbnails

Displaying thumbnails in your UI

New thumbnail keys available!

- NSURLThumbnailDictionaryKey – NSDictionary
    - NSThumbnail1024x1024SizeKey – UIImage / NSImage
- NSURLThumbnailKey (OSX) – NSImage

Easiest way to get thumbnails to display in UI

```
– getPromisedItemResourceValue:forKey:error:
– promisedItemResourceValuesForKeys:error:
```

Document Thumbnails

Saving thumbnails

Easiest way to save thumbnails is through UIDocument

```
- fileAttributesToWriteToURL:forSaveOperation:error:
```

Document Thumbnails

NEW

Saving thumbnails

Easiest way to save thumbnails is through UIDocument

`- fileAttributesToWriteToURL:forSaveOperation:error:`

Or you can set thumbnails manually

`- setResourceValue:forKey:error:`

Document Thumbnails

Saving thumbnails

NEW

Easiest way to save thumbnails is through UIDocument

```
– fileAttributesToWriteToURL:forSaveOperation:error:
```

Or you can set thumbnails manually

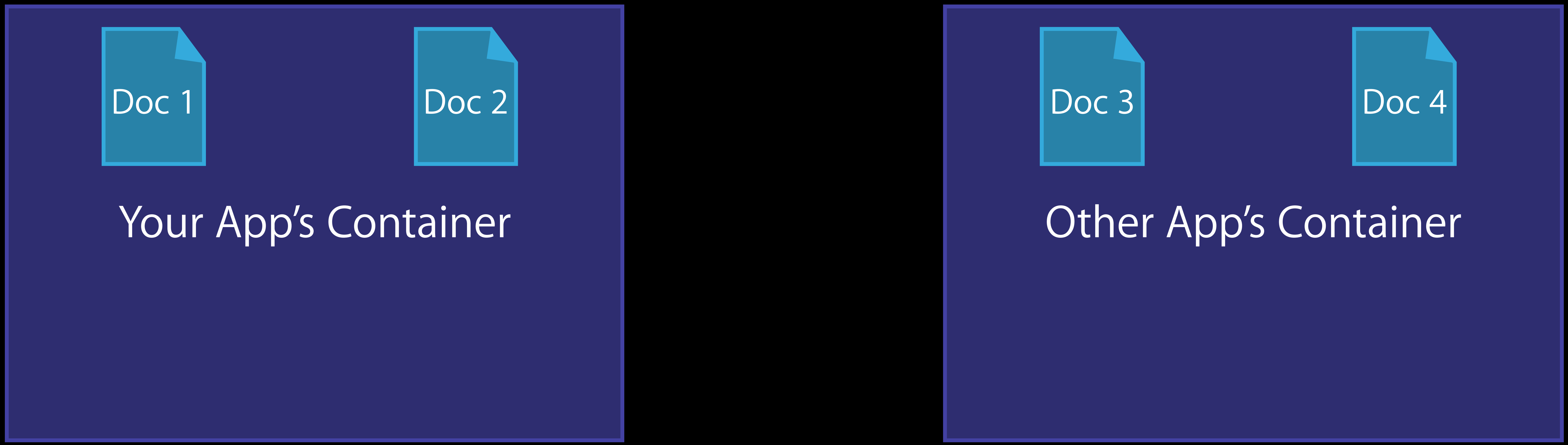
```
– setResourceValue:forKey:error:
```

OS X generates thumbnails automatically using your Quick Look plug-in

Discover Documents Outside Your Container

Document Discovery Not Always Easy

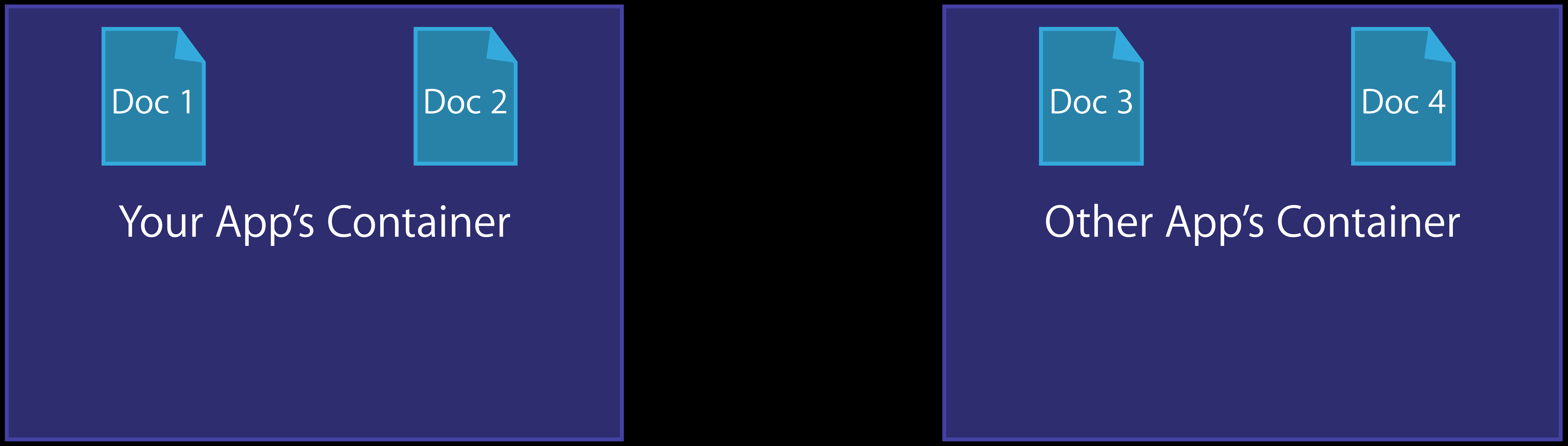
Difficult to access documents from another app



Document Discovery Not Always Easy

Difficult to access documents from another app

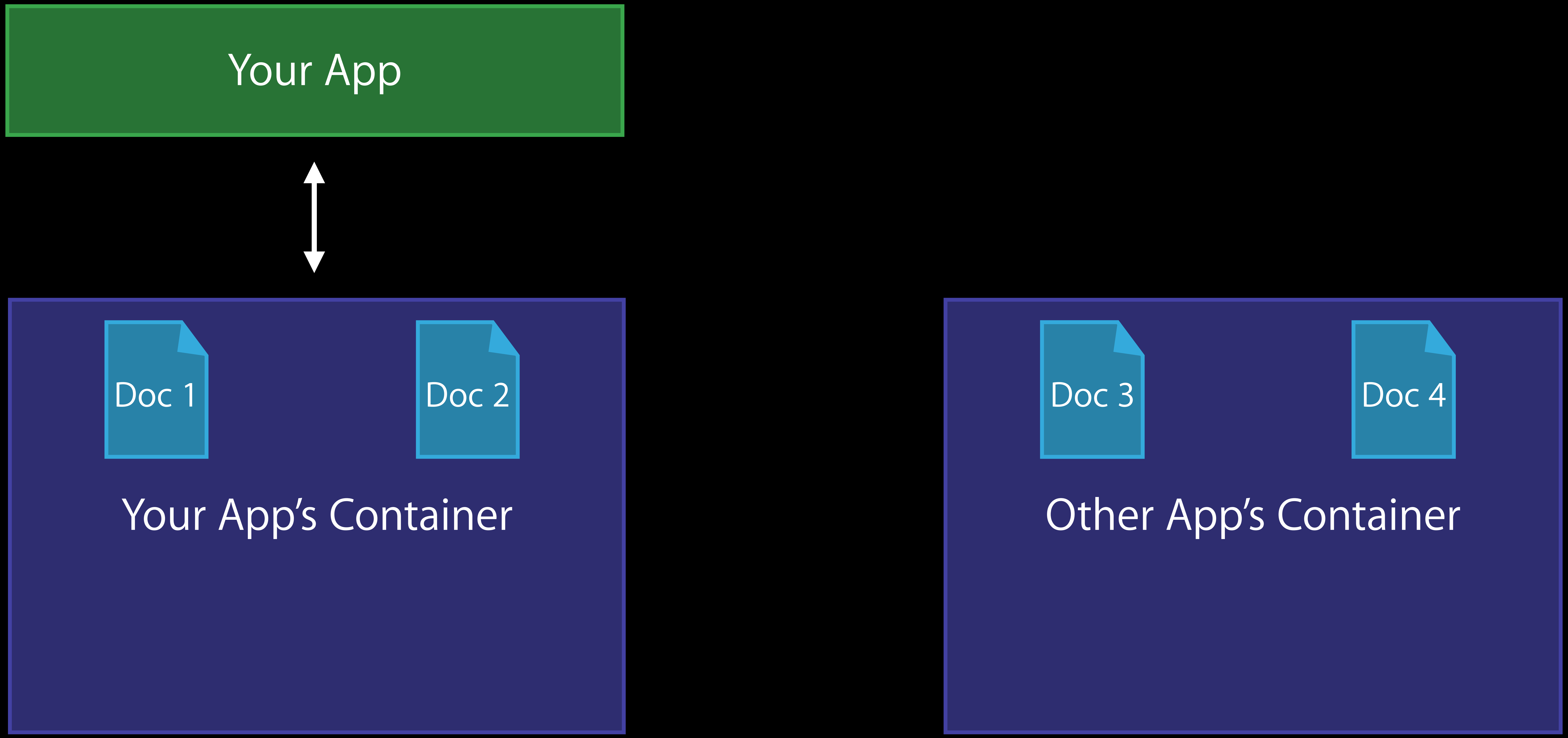
No easy way to discover documents outside your app from within your app



Document Discovery Not Always Easy

Difficult to access documents from another app

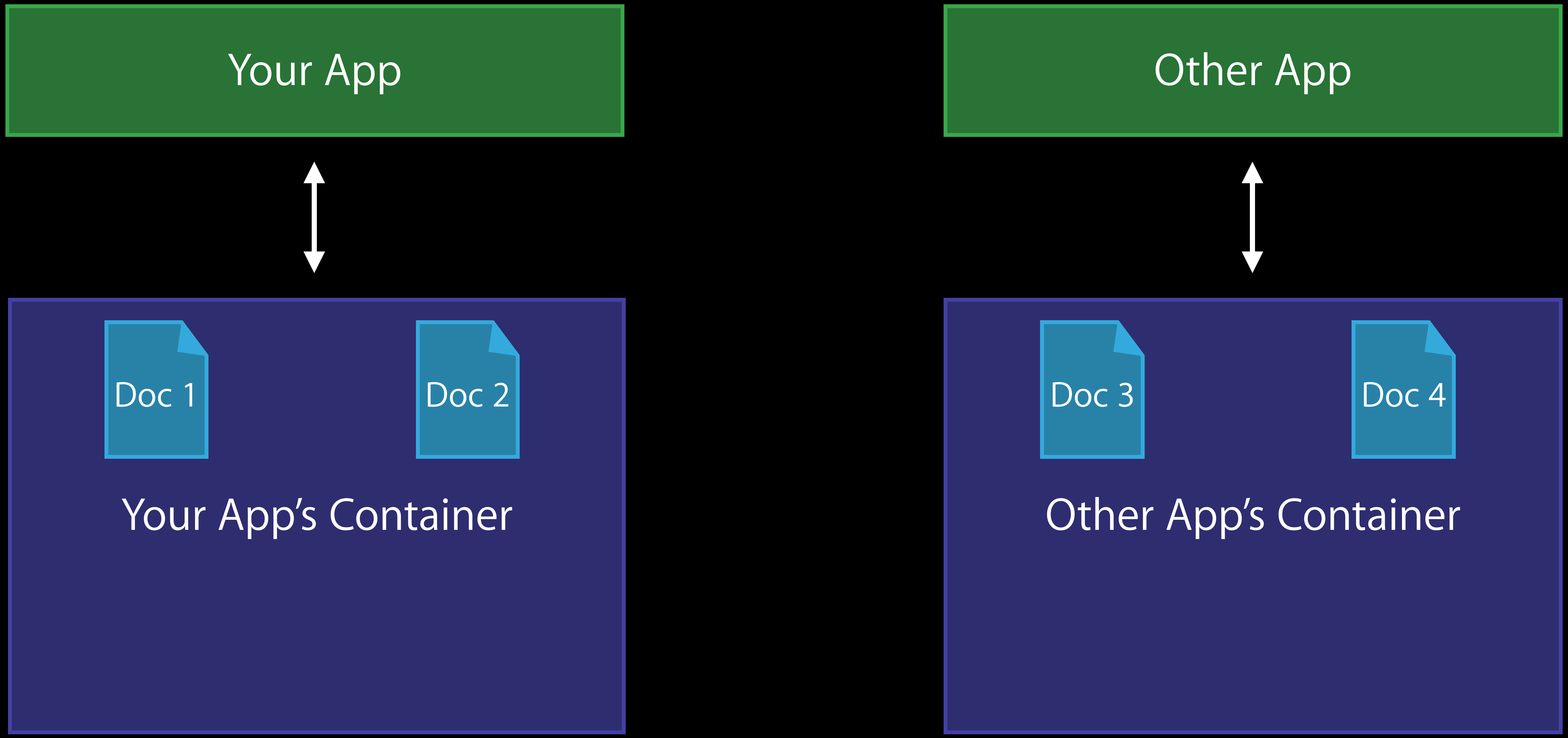
No easy way to discover documents outside your app from within your app



Document Discovery Not Always Easy

Difficult to access documents from another app

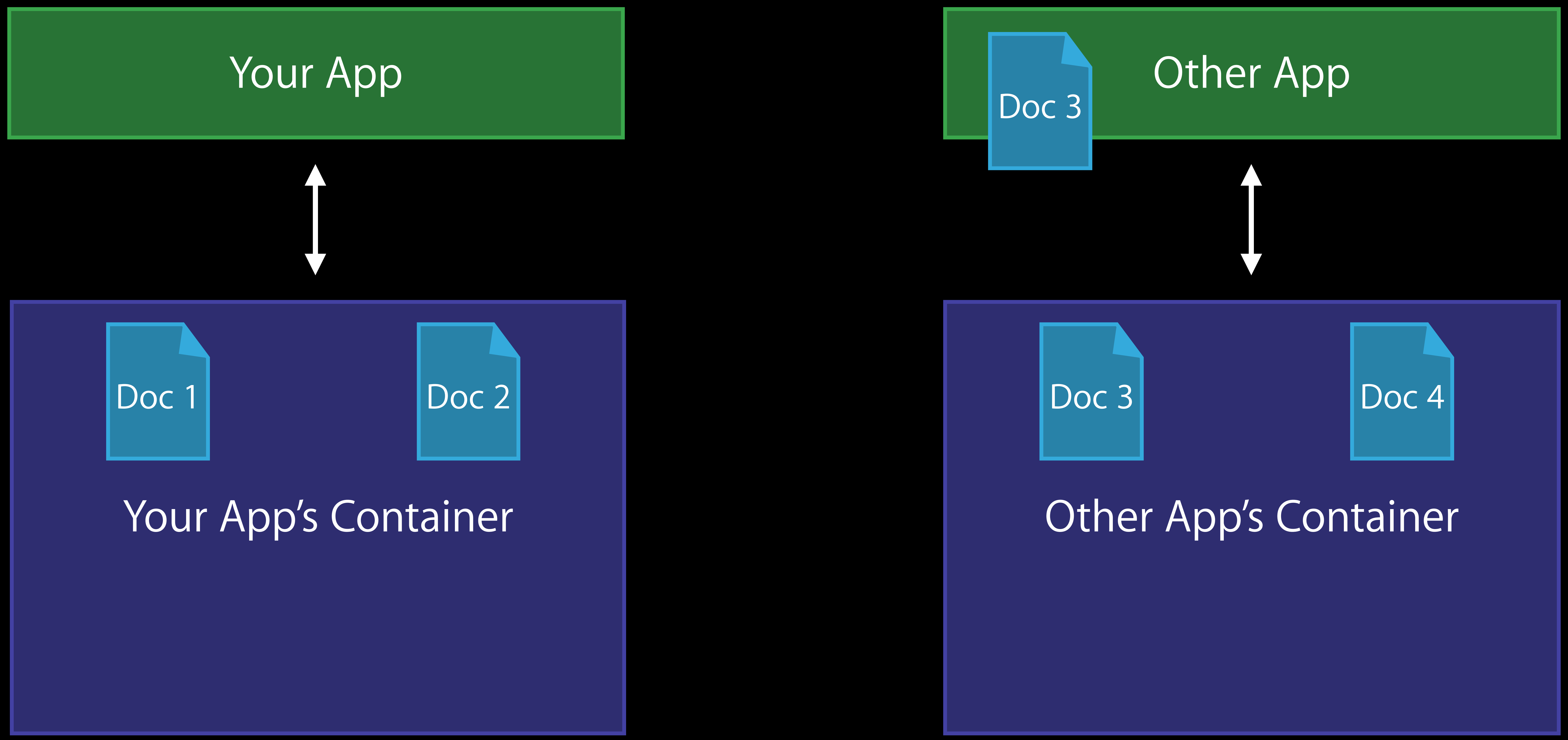
No easy way to discover documents outside your app from within your app



Document Discovery Not Always Easy

Difficult to access documents from another app

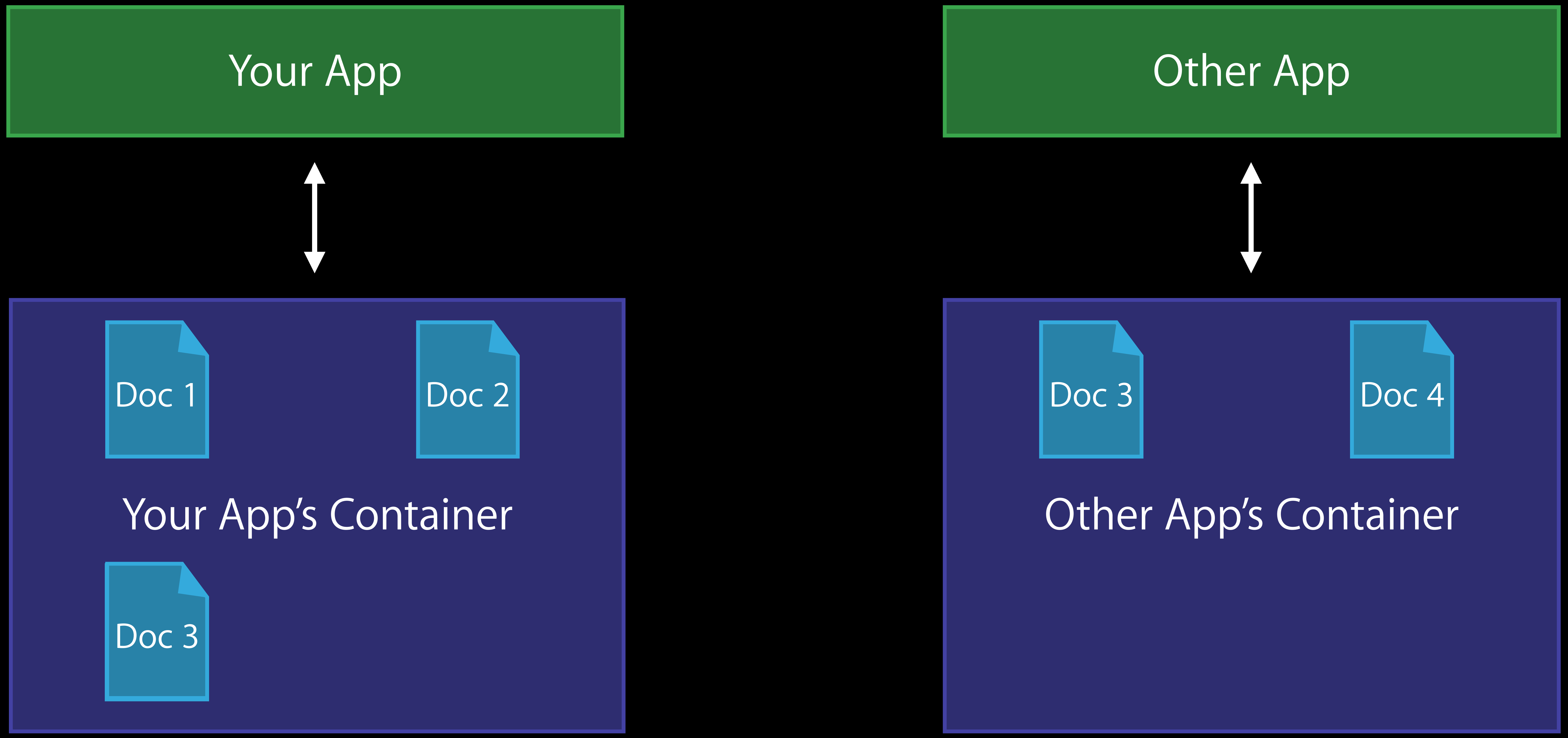
No easy way to discover documents outside your app from within your app



Document Discovery Not Always Easy

Difficult to access documents from another app

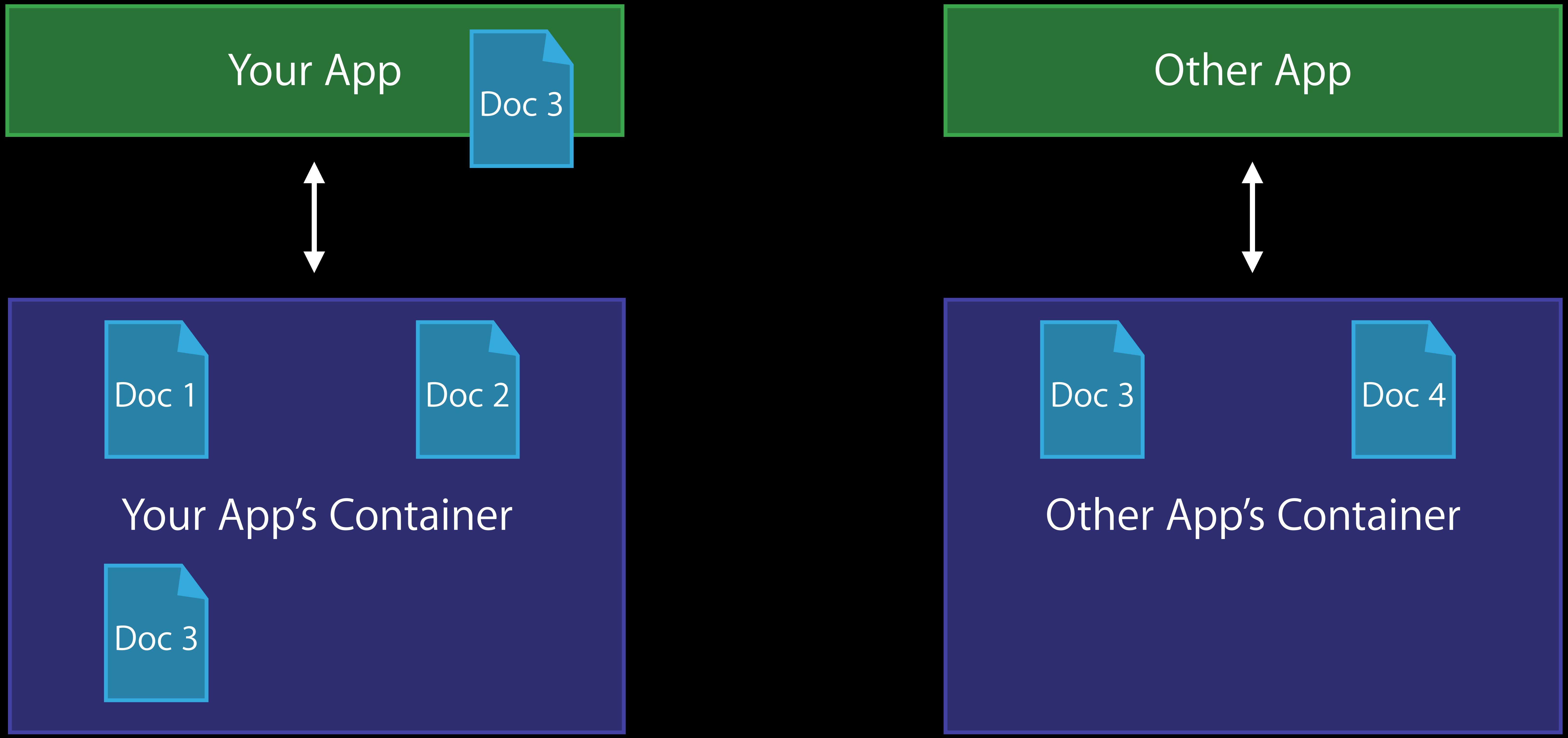
No easy way to discover documents outside your app from within your app



Document Discovery Not Always Easy

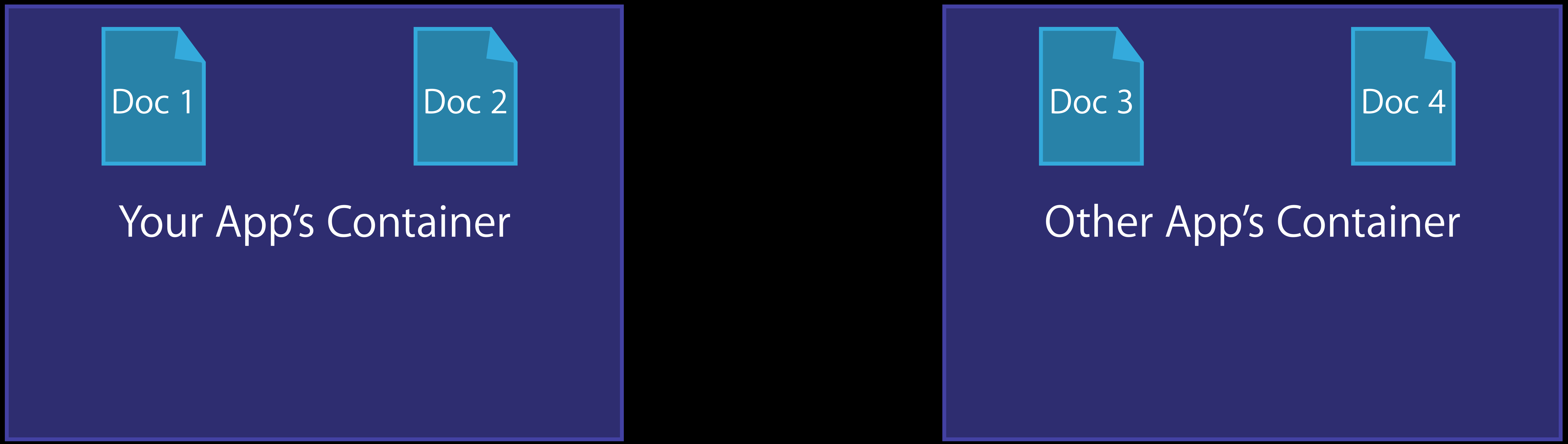
Difficult to access documents from another app

No easy way to discover documents outside your app from within your app



Document Discovery Made Easier

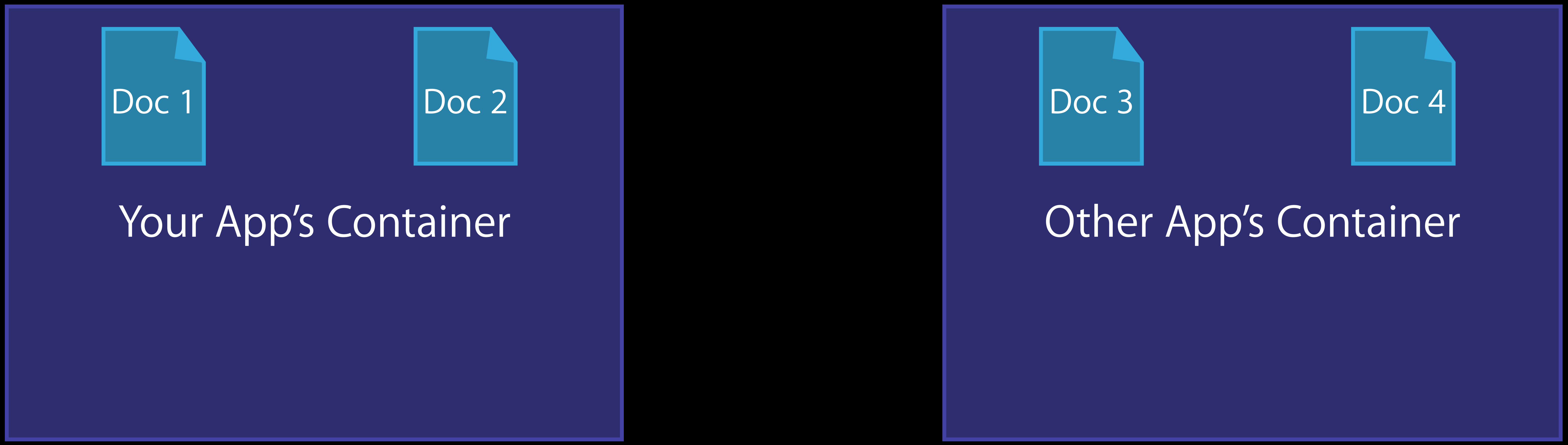
Accessing documents directly



Document Discovery Made Easier

Accessing documents directly

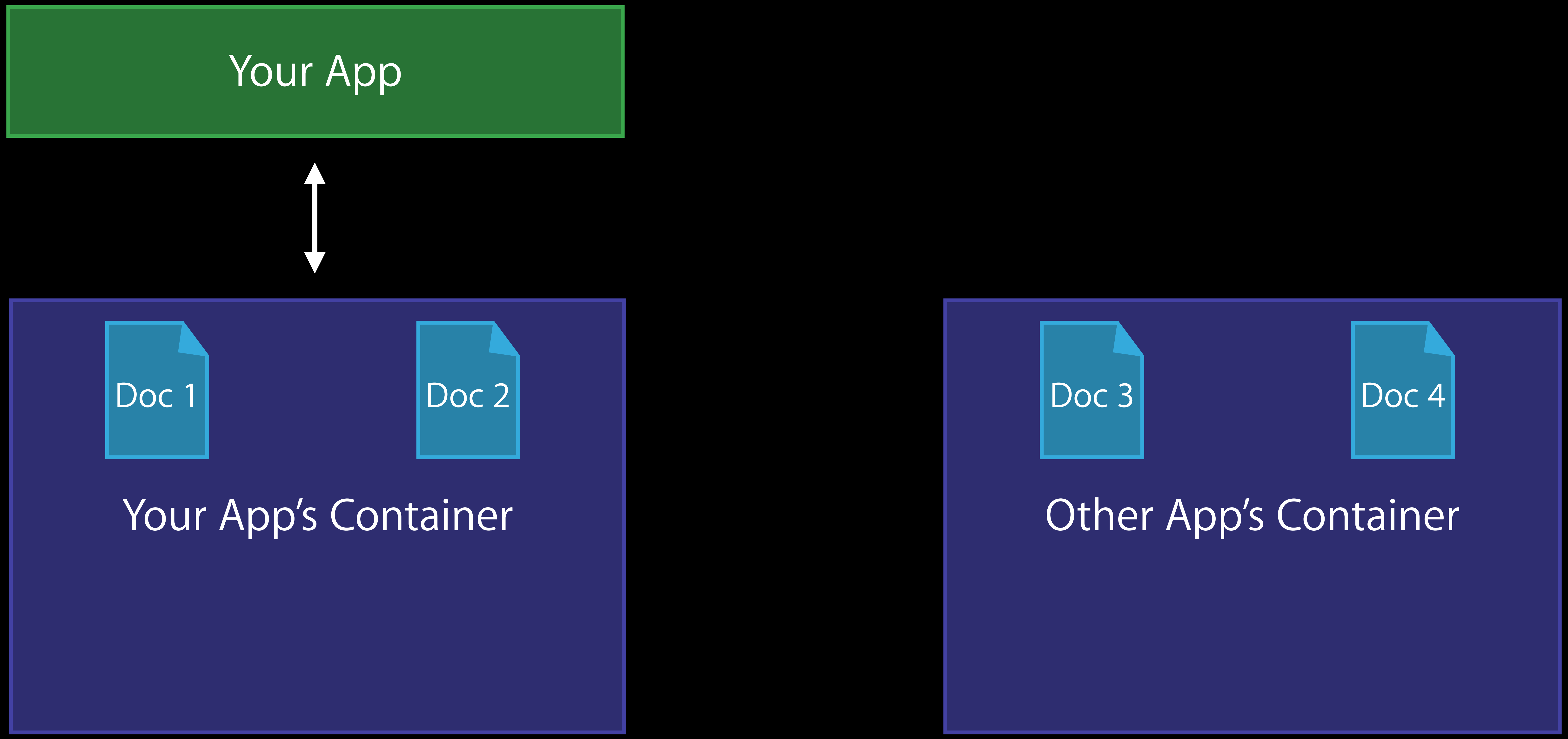
Better user experience to open documents directly



Document Discovery Made Easier

Accessing documents directly

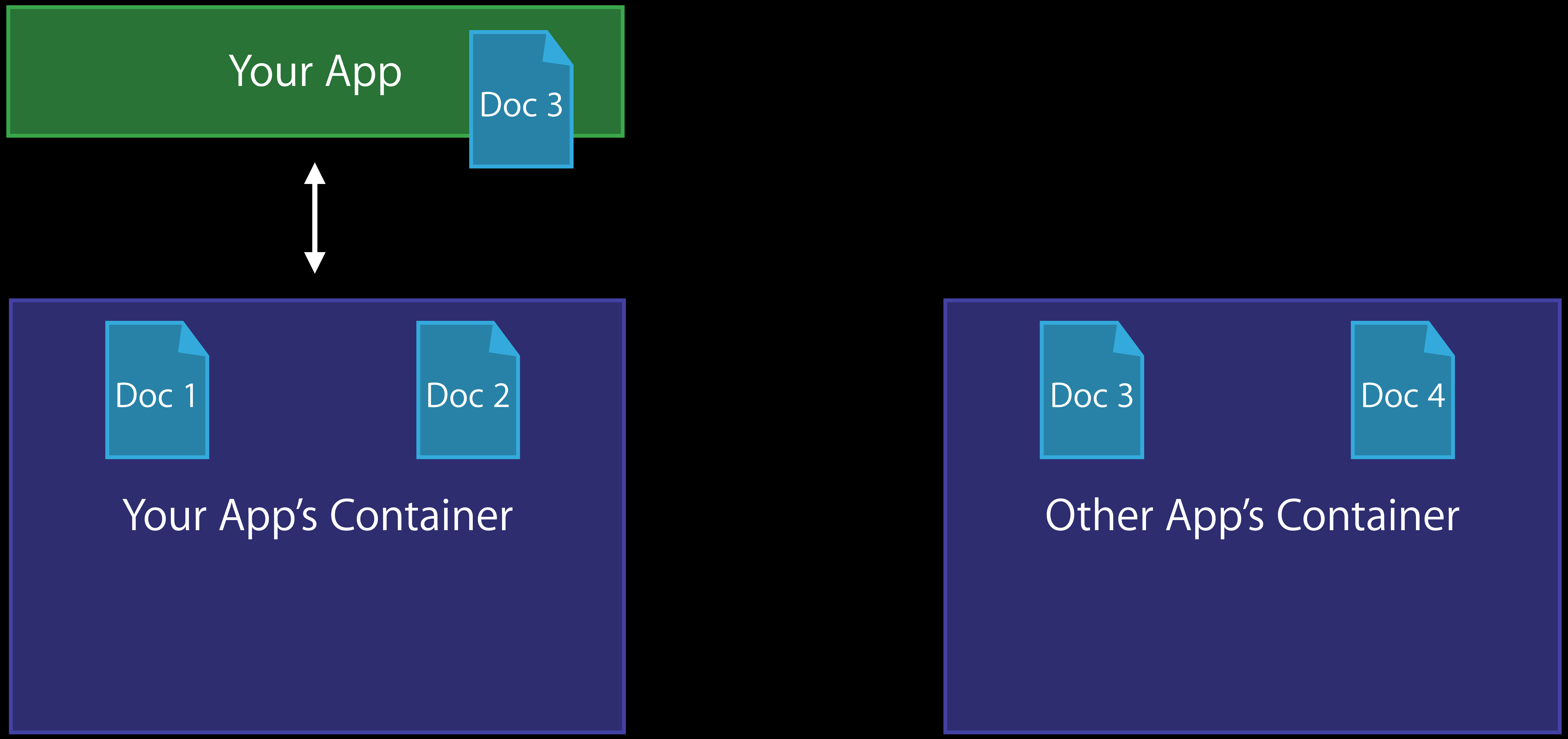
Better user experience to open documents directly



Document Discovery Made Easier

Accessing documents directly

Better user experience to open documents directly



Document Discovery Made Easier

Access documents outside your container

NEW

Document Discovery Made Easier

Access documents outside your container

NEW

UIDocumentPickerViewController allows your users to

- Discover documents outside of your app's sandbox
- Grant your app permissions to access and edit the discovered documents

Document Discovery Made Easier

Access documents outside your container

UIDocumentPickerViewController allows your users to

- Discover documents outside of your app's sandbox
- Grant your app permissions to access and edit the discovered documents

Must opt-in to have documents displayed

```
    NSUbiquitousContainerIsDocumentScopePublic = YES
```

Document Discovery Made Easier

Access documents outside your container

NEW

UIDocumentPickerViewController allows your users to

- Discover documents outside of your app's sandbox
- Grant your app permissions to access and edit the discovered documents

Must opt-in to have documents displayed

```
NSUbiquitousContainerIsDocumentScopePublic = YES
```

Great new UI which you display in your app

Demo

Using the document picker in a simple application

Summary

Summary

Easy to display UIDocumentPickerViewController

Summary

Easy to display UIDocumentPickerViewController

Users can now select documents from other apps' containers

Summary

Easy to display UIDocumentPickerViewController

Users can now select documents from other apps' containers

Easy to make your container show up in UIDocumentPickerViewController

Managing External Documents

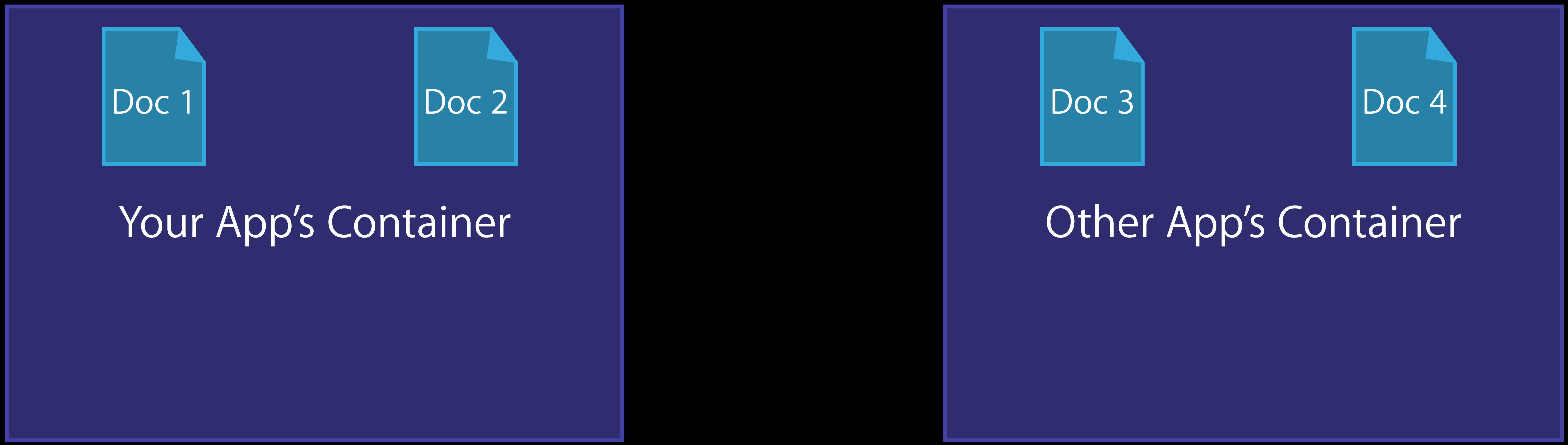
Document references, scoped access, and bookmarks

Johannes Fortmann
Software Engineer

Managing Document References

Document references in your container

NEW

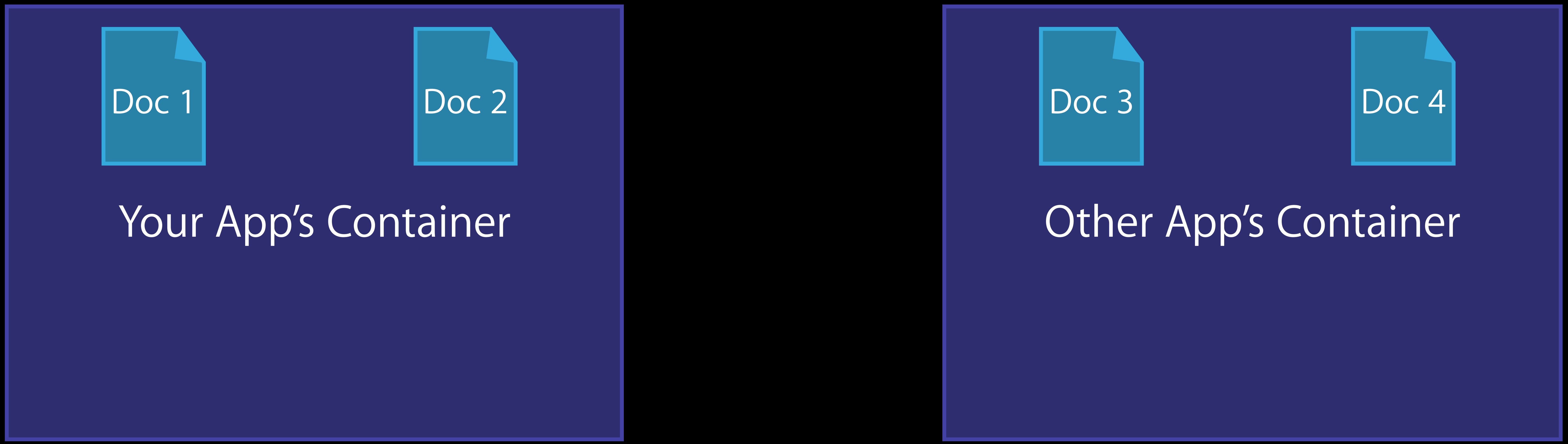


Managing Document References

NEW

Document references in your container

Your app always has access to its own container

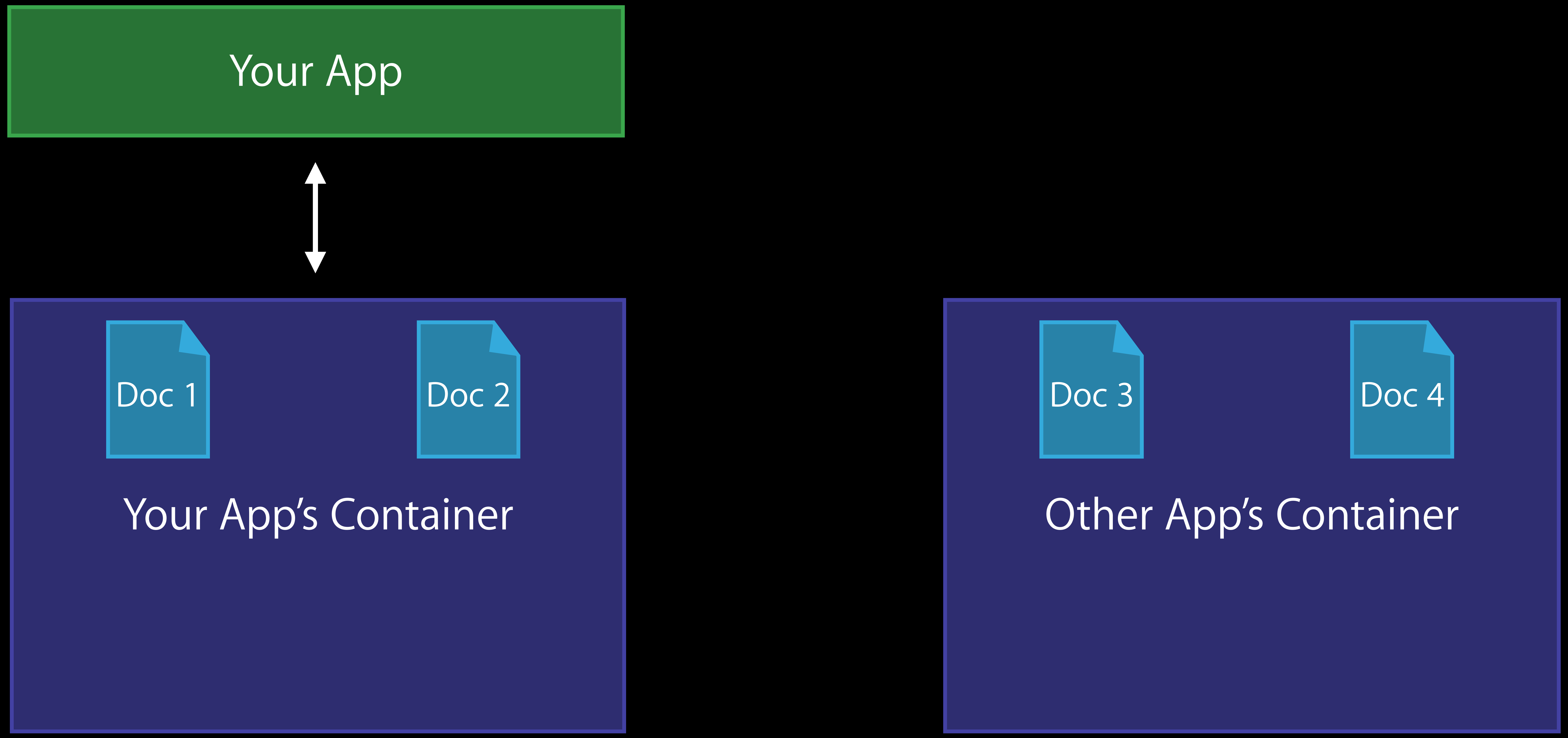


Managing Document References

Document references in your container

NEW

Your app always has access to its own container



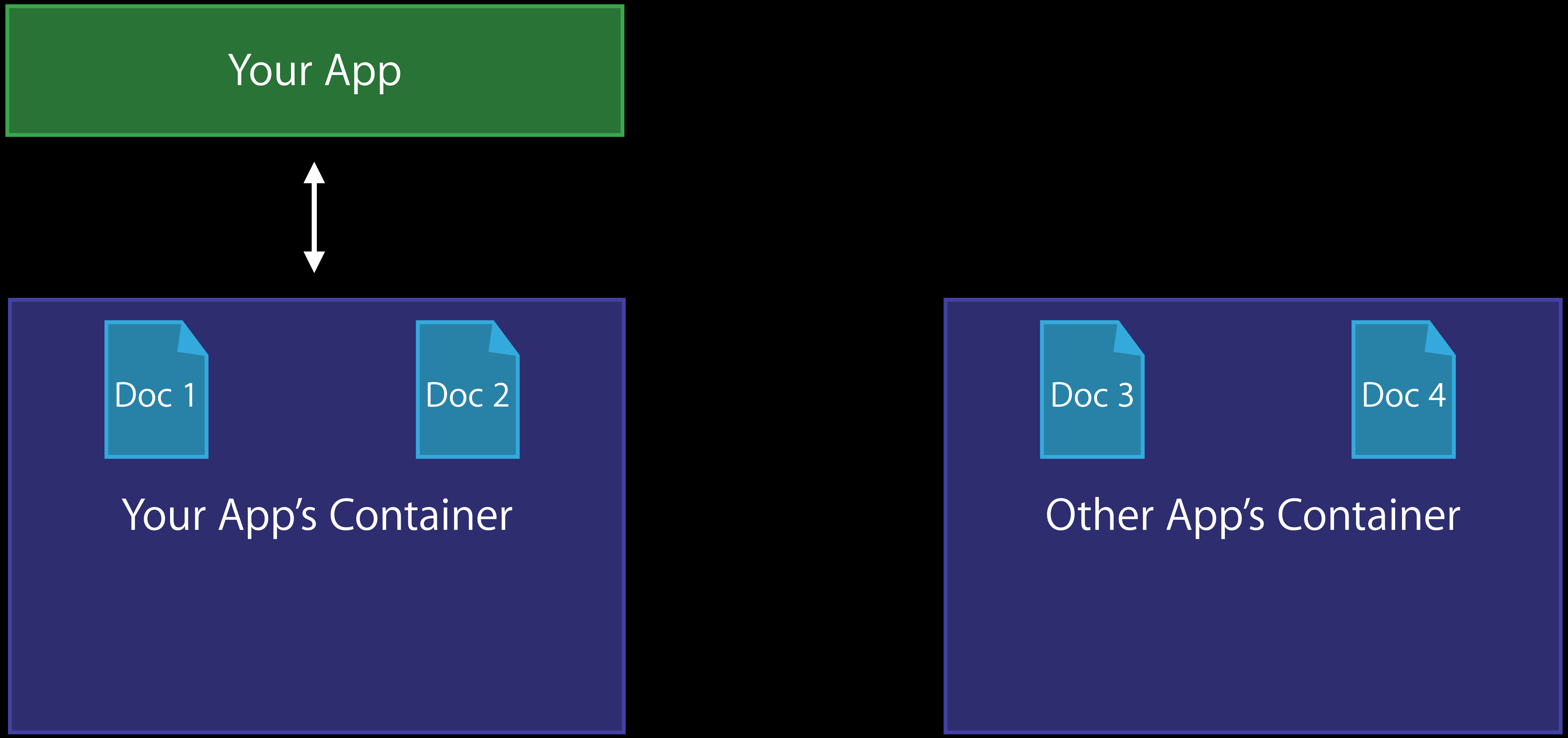
Managing Document References

Document references in your container

NEW

Your app always has access to its own container

For other documents, we save a document reference



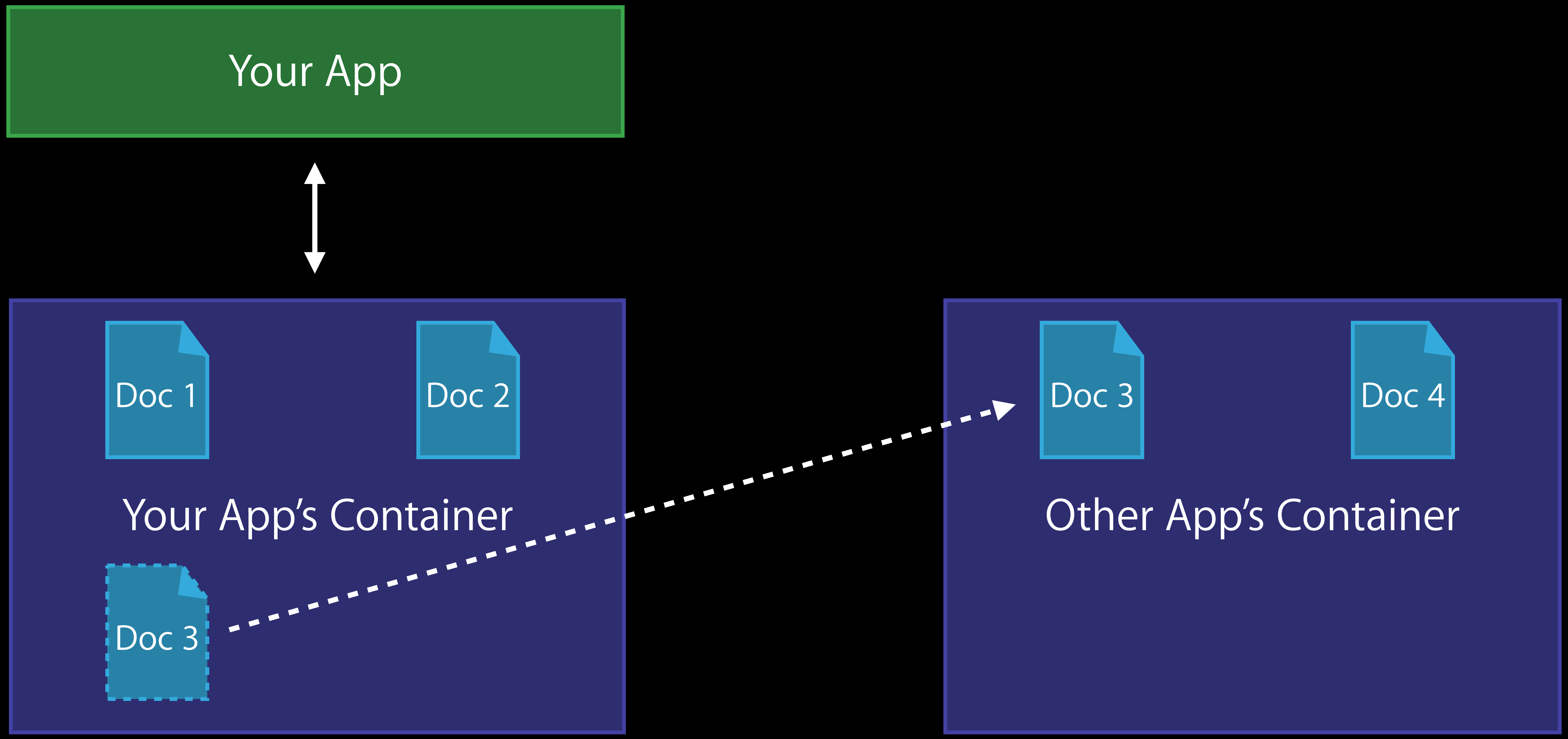
Managing Document References

Document references in your container

NEW

Your app always has access to its own container

For other documents, we save a document reference

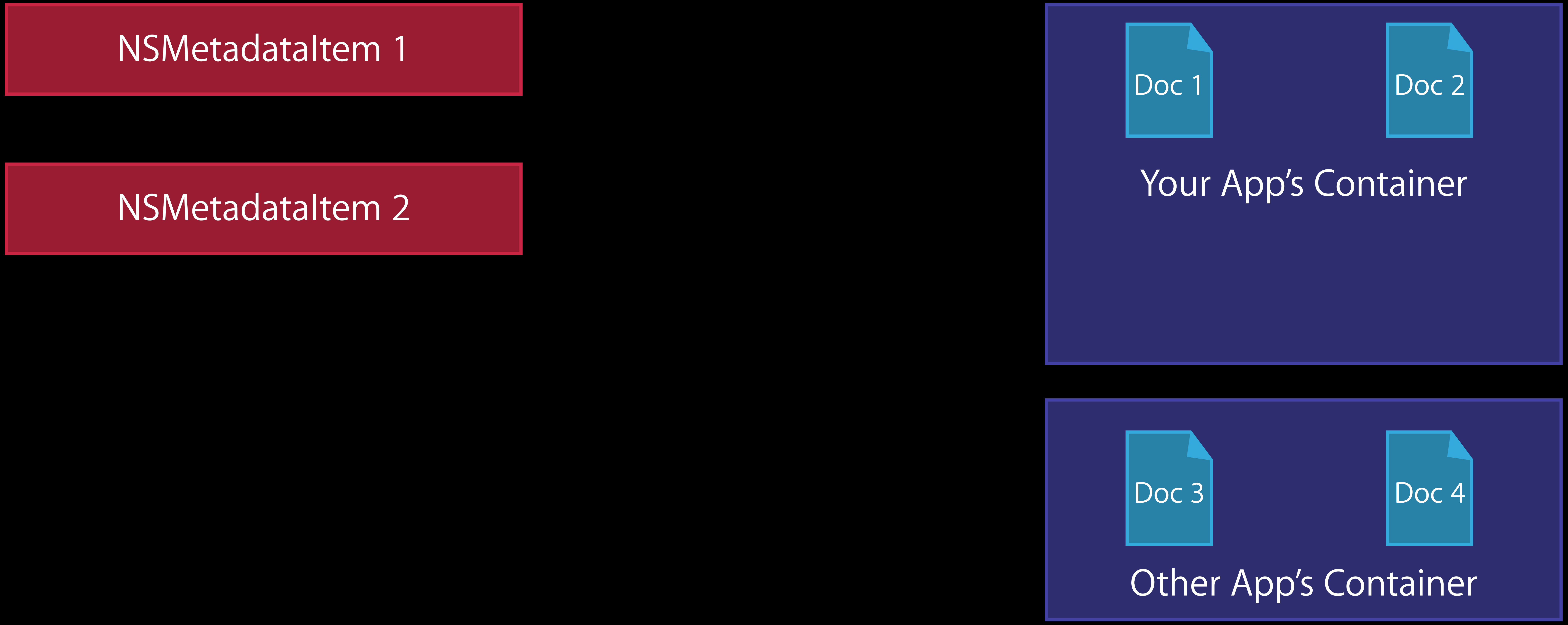


Managing Document References

New attributes

NEW

`NSMetadataQueryUbiquitousDocumentsScope`

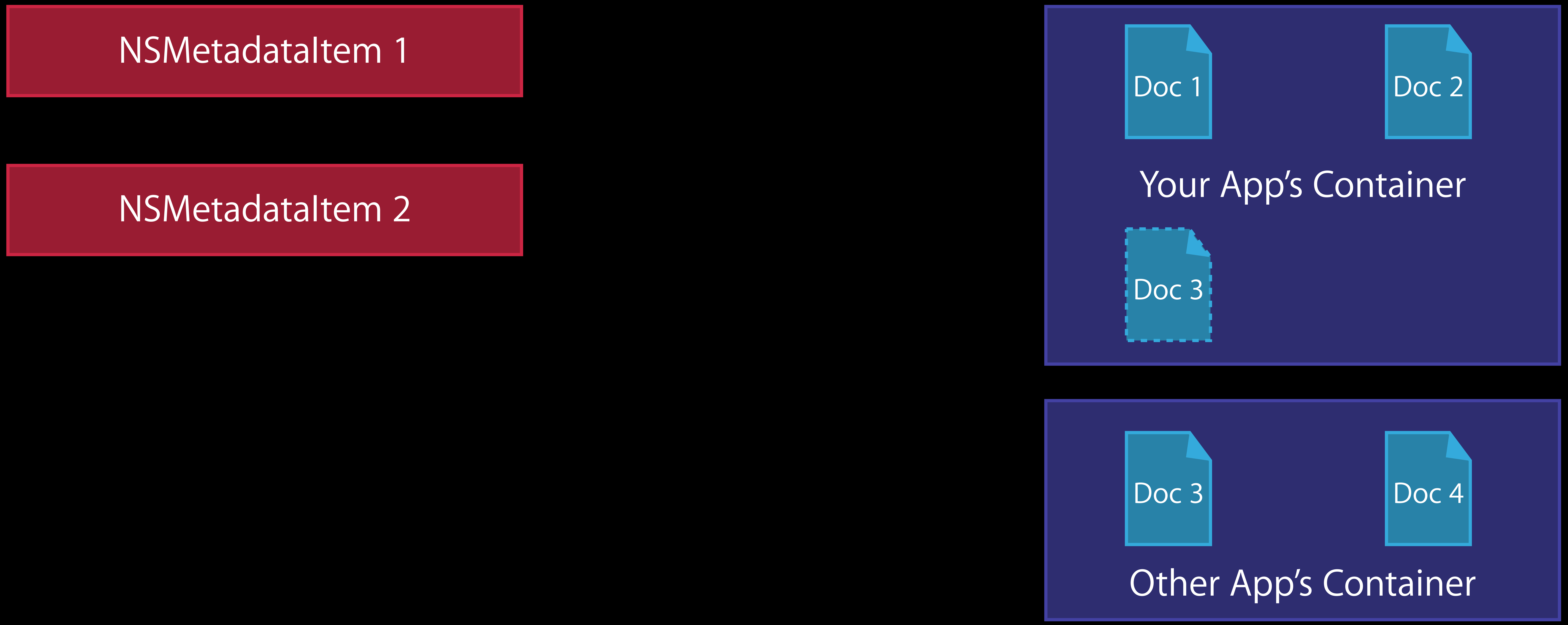


Managing Document References

New attributes

NEW

`NSMetadataQueryUbiquitousDocumentsScope`

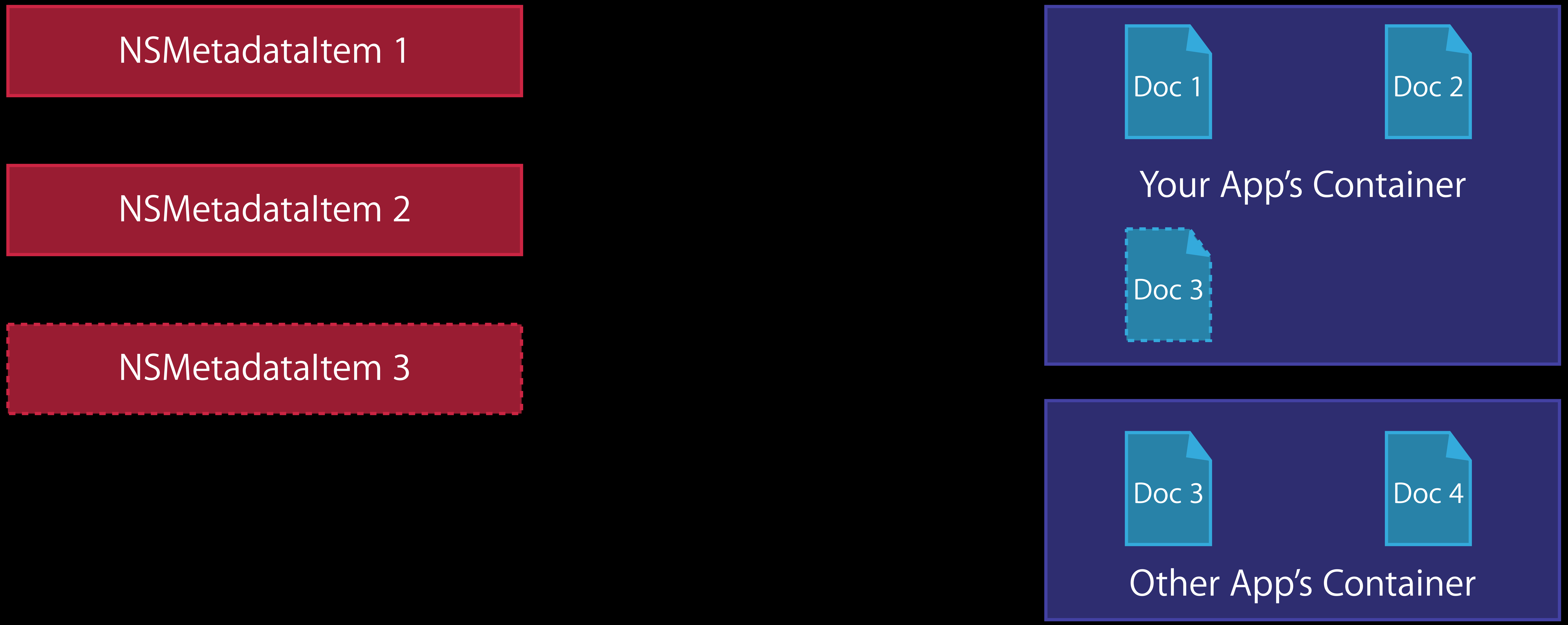


Managing Document References

New attributes

NEW

```
NSMetadataQueryUbiquitousDocumentsScope
NSMetadataQueryAccessibleUbiquitousExternalDocumentsScope
```

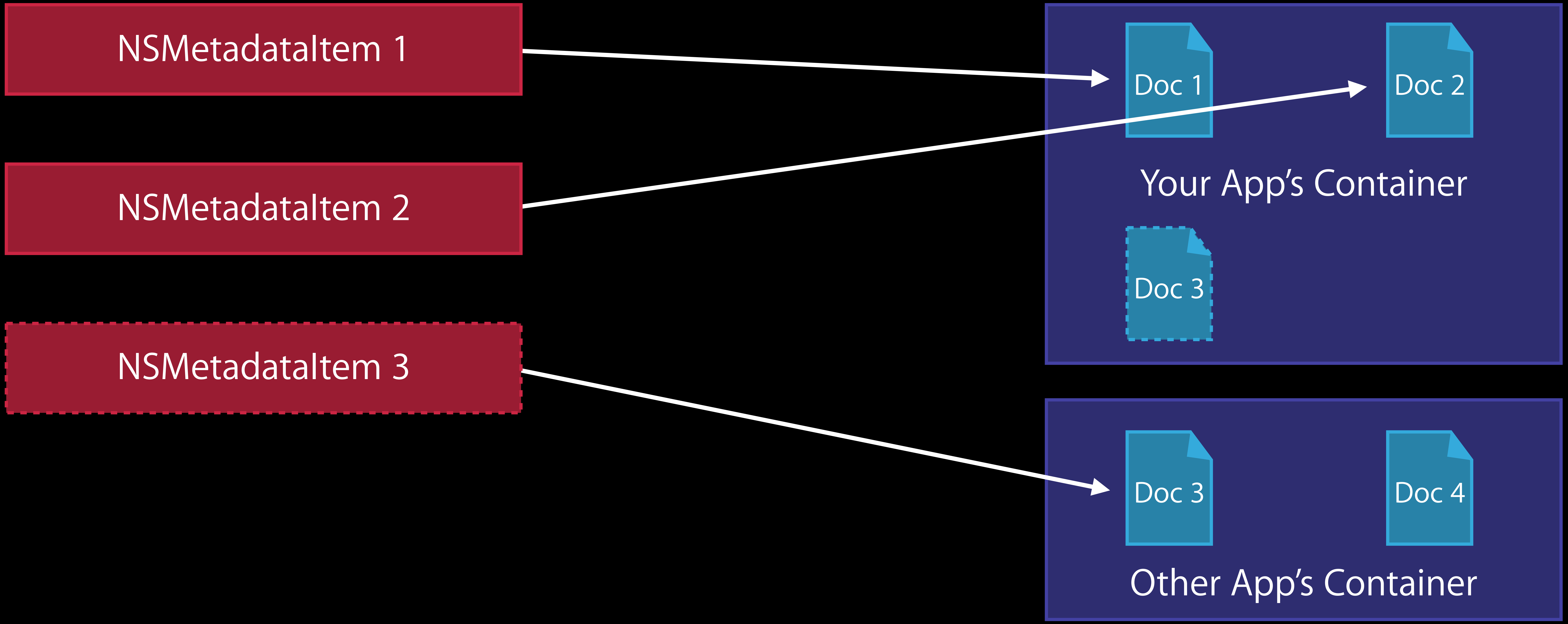


Managing Document References

New attributes

NEW

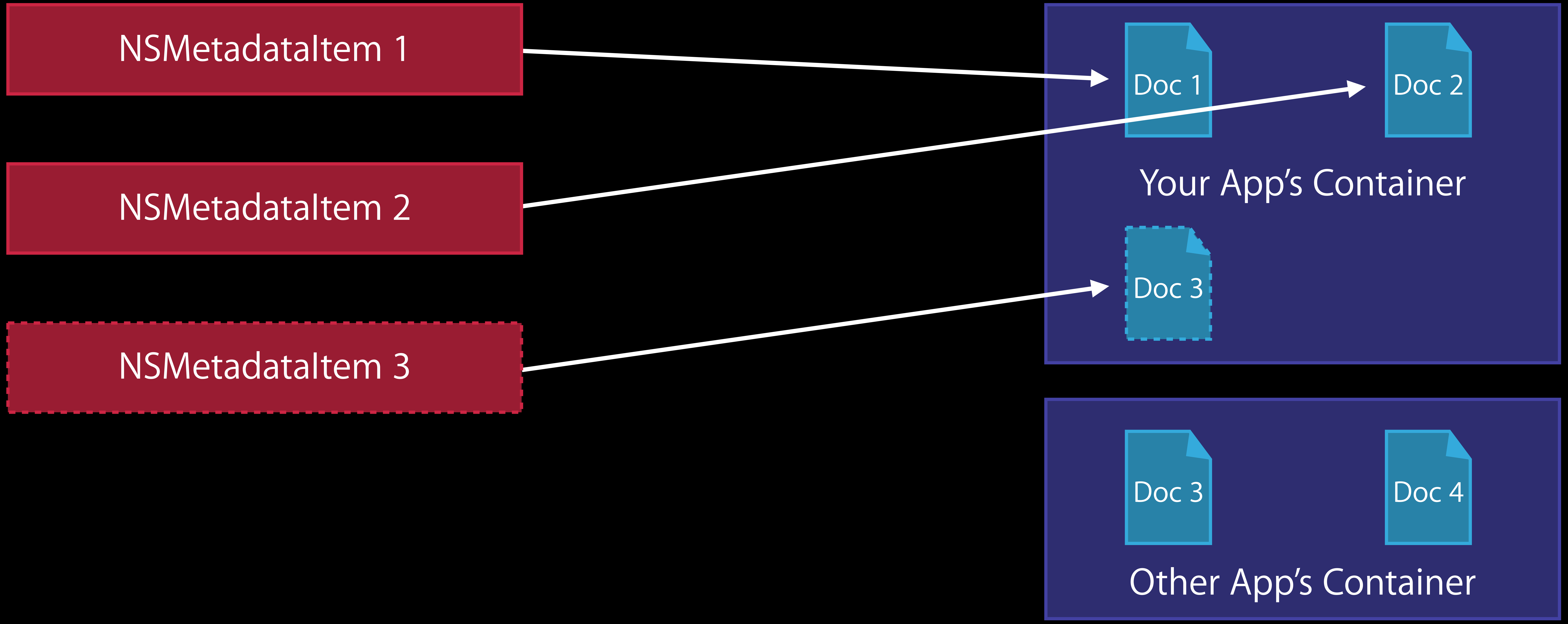
```
NSMetadataQueryUbiquitousDocumentsScope
NSMetadataQueryAccessibleUbiquitousExternalDocumentsScope
NSMetadataItemURLKey
```



Managing Document References

New attributes

```
NSMetadataQueryUbiquitousDocumentsScope
NSMetadataQueryAccessibleUbiquitousExternalDocumentsScope
NSMetadataUbiquitousItemURLInLocalContainerKey
```

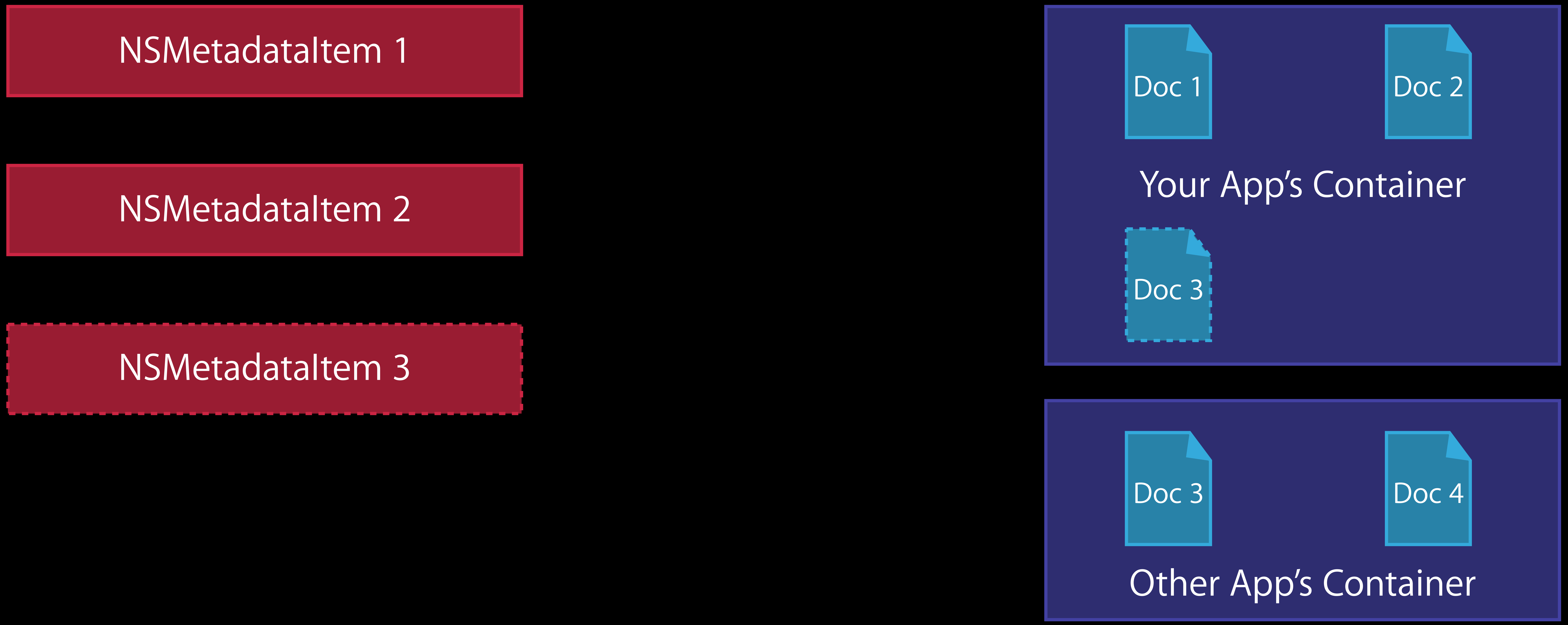


Managing Document References

New attributes

NEW

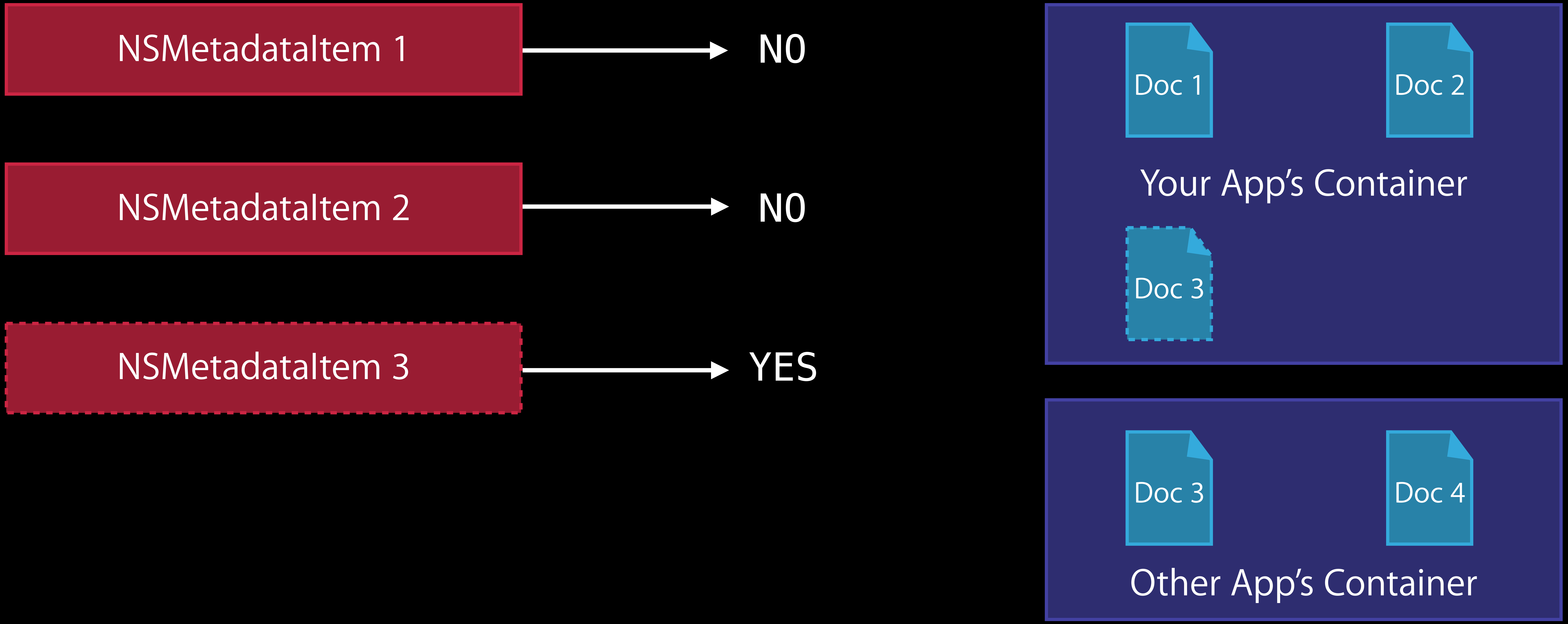
```
NSMetadataQueryUbiquitousDocumentsScope
NSMetadataQueryAccessibleUbiquitousExternalDocumentsScope
NSMetadataUbiquitousItemIsExternalDocumentKey
```



Managing Document References

New attributes

```
NSMetadataQueryUbiquitousDocumentsScope
NSMetadataQueryAccessibleUbiquitousExternalDocumentsScope
NSMetadataUbiquitousItemIsExternalDocumentKey
```

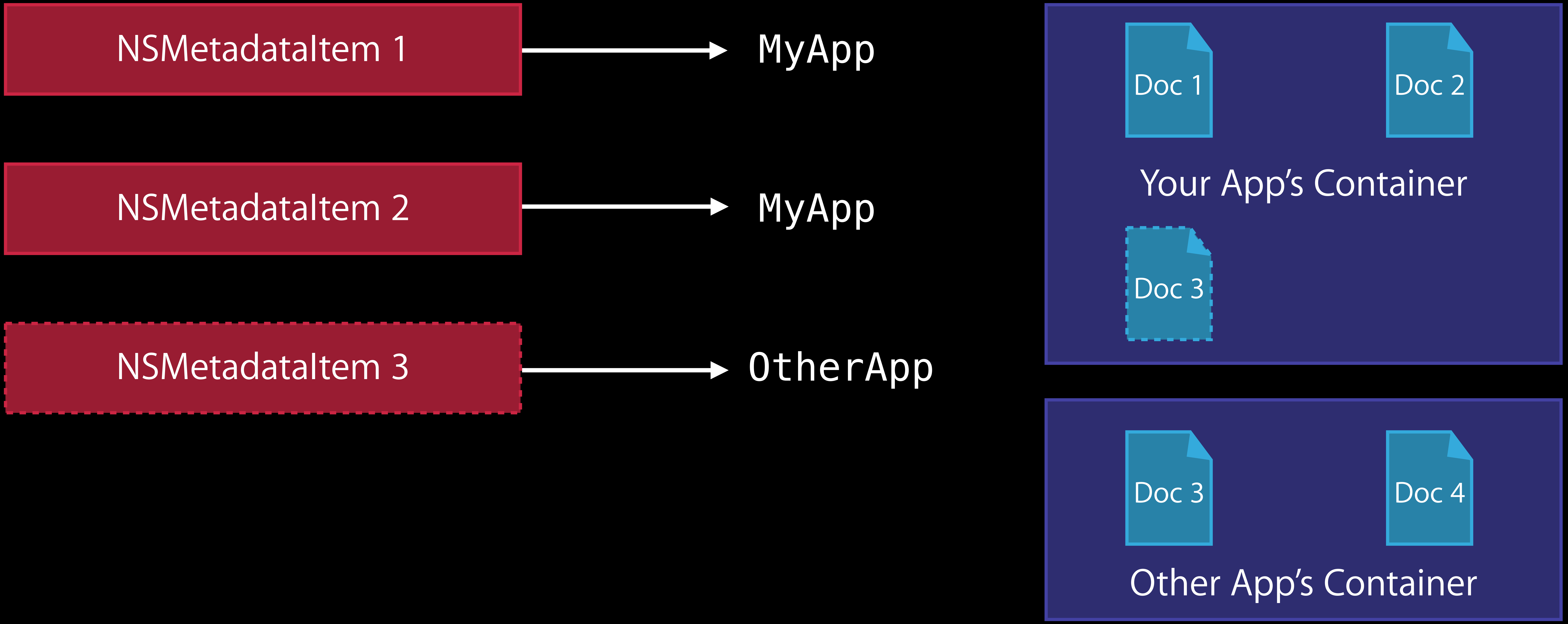


Managing Document References

New attributes

NEW

```
NSMetadataQueryUbiquitousDocumentsScope
NSMetadataQueryAccessibleUbiquitousExternalDocumentsScope
NSMetadataUbiquitousItemContainerDisplayNameKey
```



Managing Document References

Why do you need the document reference?

Managing Document References

Why do you need the document reference?

The document picker is running out-of-process for security

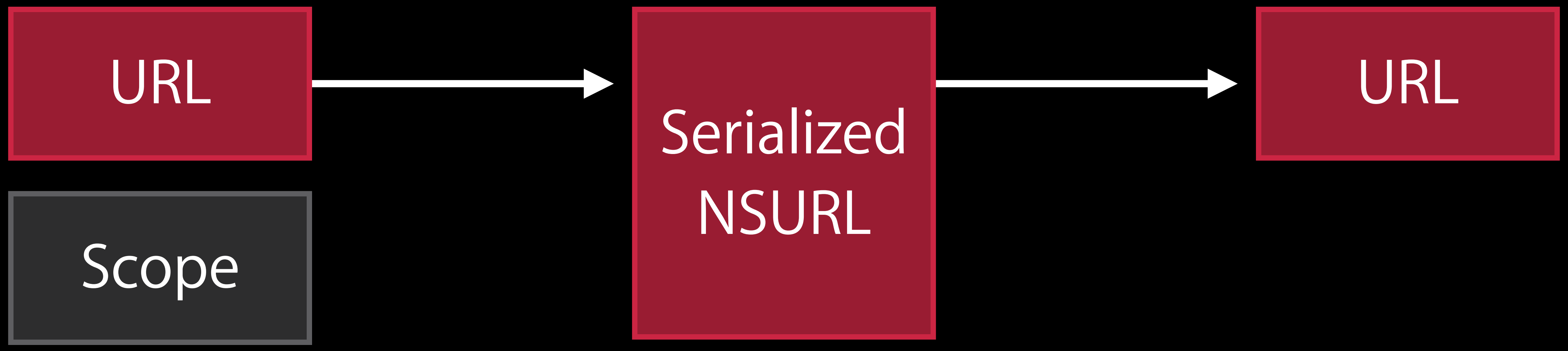
Managing Document References

Why do you need the document reference?

The document picker is running out-of-process for security

The only way to get a document outside your container is by using the picker

URL returned by the picker is security scoped



Managing Document References

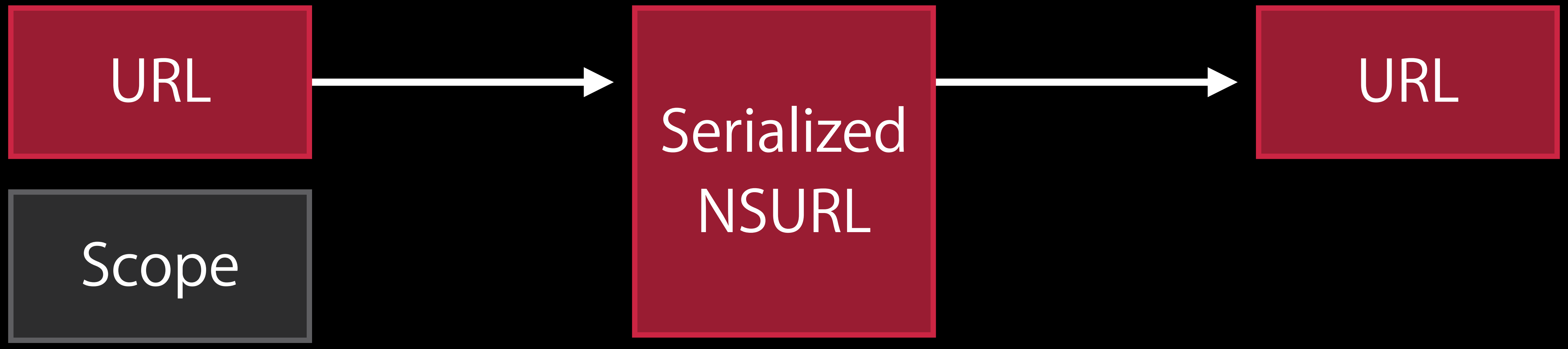
Why do you need the document reference?

The document picker is running out-of-process for security

The only way to get a document outside your container is by using the picker

URL returned by the picker is security scoped

When saving a URL (state restoration, recents list), scope would be lost



Managing Document References

Why do you need the document reference?

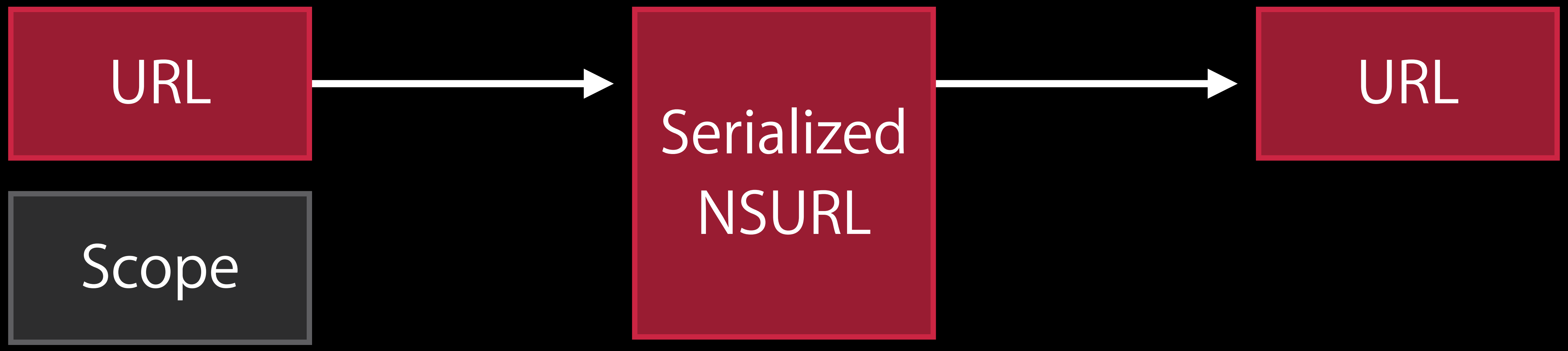
The document picker is running out-of-process for security

The only way to get a document outside your container is by using the picker

URL returned by the picker is security scoped

When saving a URL (state restoration, recents list), scope would be lost

You need a way to get back to the scoped URL



Managing Document References

Why do you need the document reference?

The document picker is running out-of-process for security

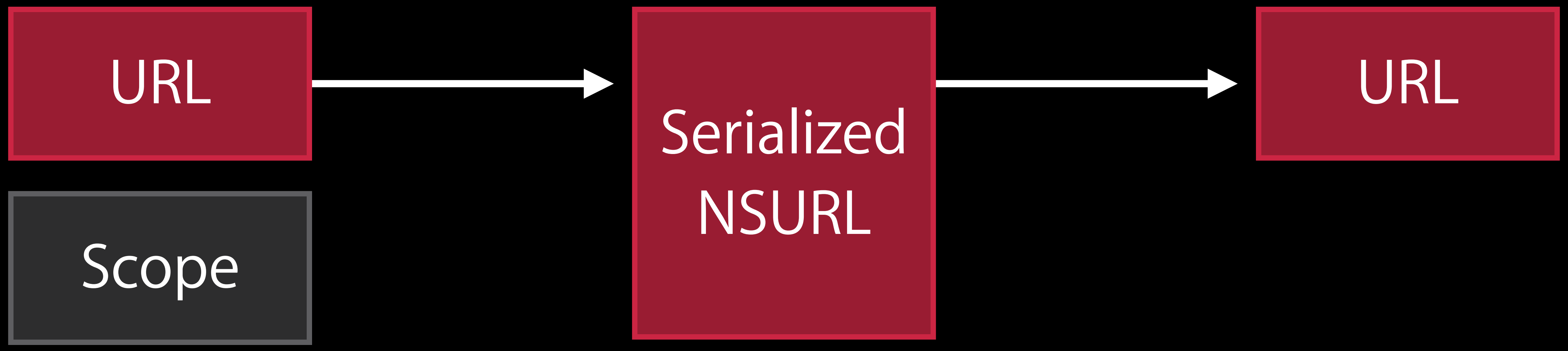
The only way to get a document outside your container is by using the picker

URL returned by the picker is security scoped

When saving a URL (state restoration, recents list), scope would be lost

You need a way to get back to the scoped URL

Document references come with scope pre-attached



Managing Document References

Bookmarks for state restoration

Managing Document References

Bookmarks for state restoration

Sometimes enumerating documents is not the right choice

Managing Document References

Bookmarks for state restoration

Sometimes enumerating documents is not the right choice

- E.g., you want to restore a previously opened document immediately

Managing Document References

Bookmarks for state restoration

Sometimes enumerating documents is not the right choice

- E.g., you want to restore a previously opened document immediately

Creating a bookmark encodes a portable reference to a document

Managing Document References

Bookmarks for state restoration

Sometimes enumerating documents is not the right choice

- E.g., you want to restore a previously opened document immediately

Creating a bookmark encodes a portable reference to a document

Includes the security scope

Managing Document References

Bookmarks for state restoration

Bookmark creation via API

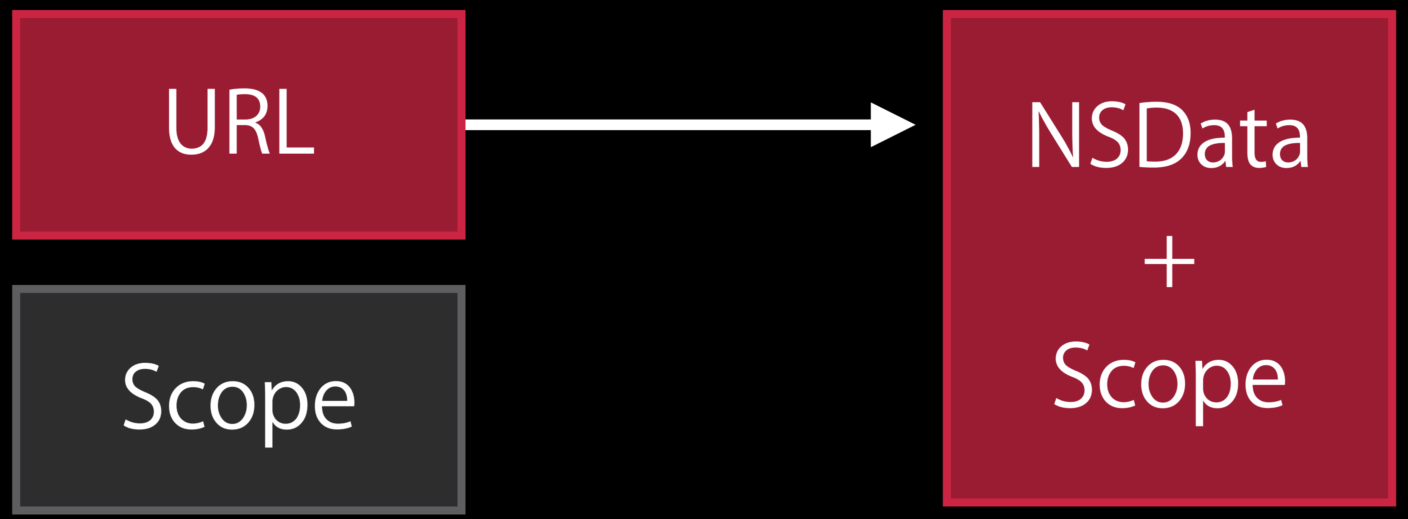
```
-bookmarkDataWithOptions:
```

Managing Document References

Bookmarks for state restoration

Bookmark creation via API

`-bookmarkDataWithOptions:`



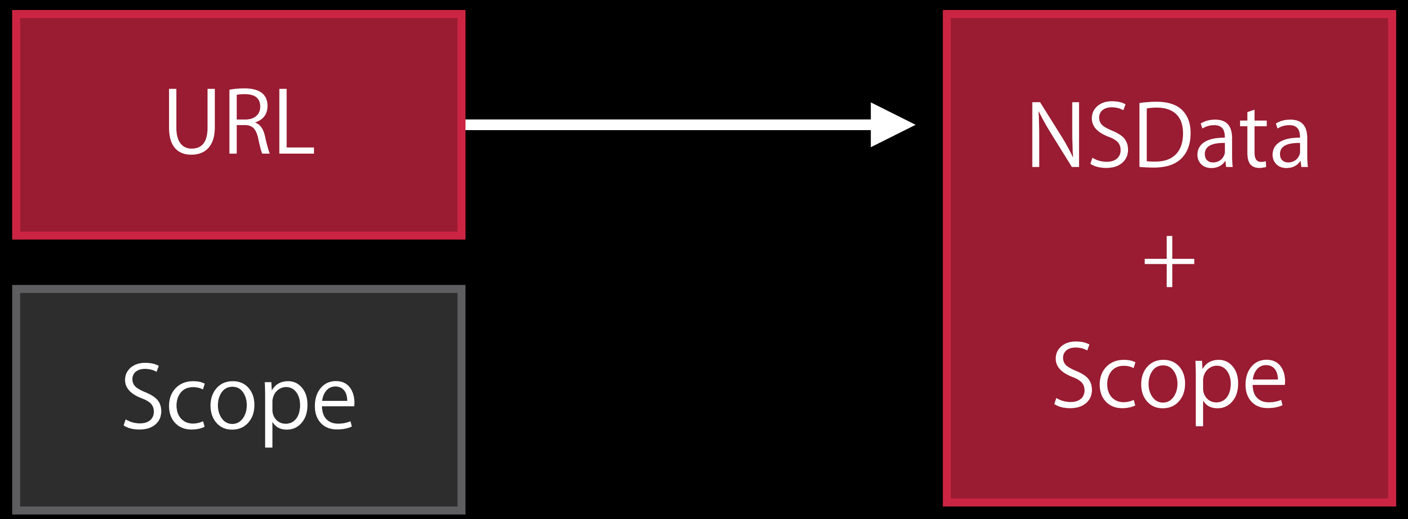
Managing Document References

Bookmarks for state restoration

Bookmark creation via API

```
-bookmarkDataWithOptions:

+URLByResolvingBookmarkData:savedBookmarkData
```



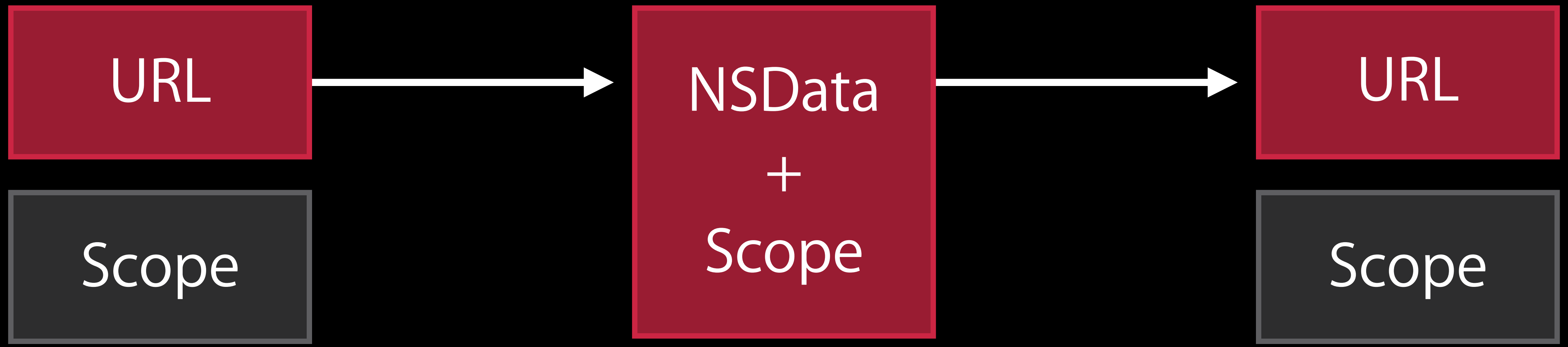
Managing Document References

Bookmarks for state restoration

Bookmark creation via API

```
-bookmarkDataWithOptions:

+URLByResolvingBookmarkData:savedBookmarkData
```



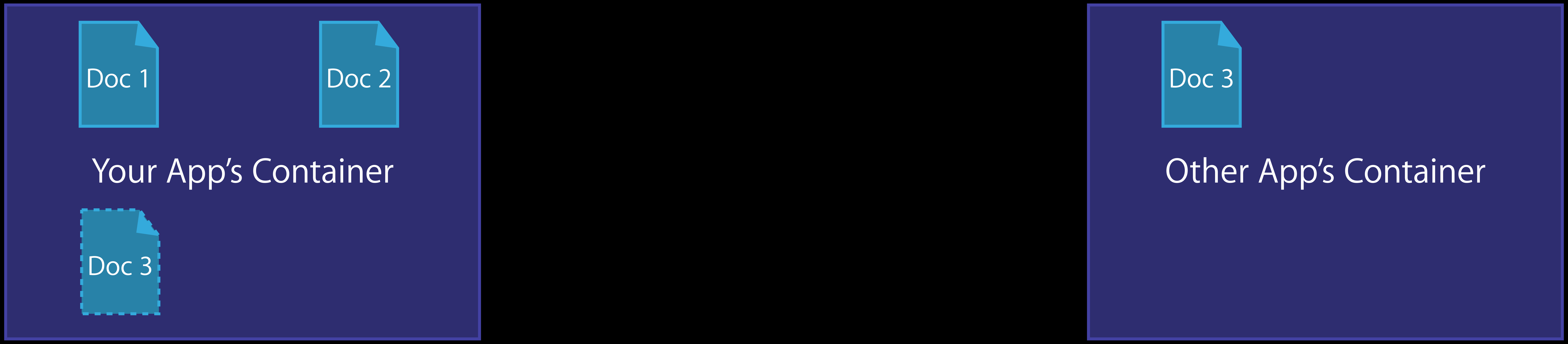
Document References and the Picker

UIDocumentPickerMode—Import vs. Open

Document References and the Picker

UIDocumentPickerMode—Import vs. Open

Open mode will create a bookmark and a security-scoped URL

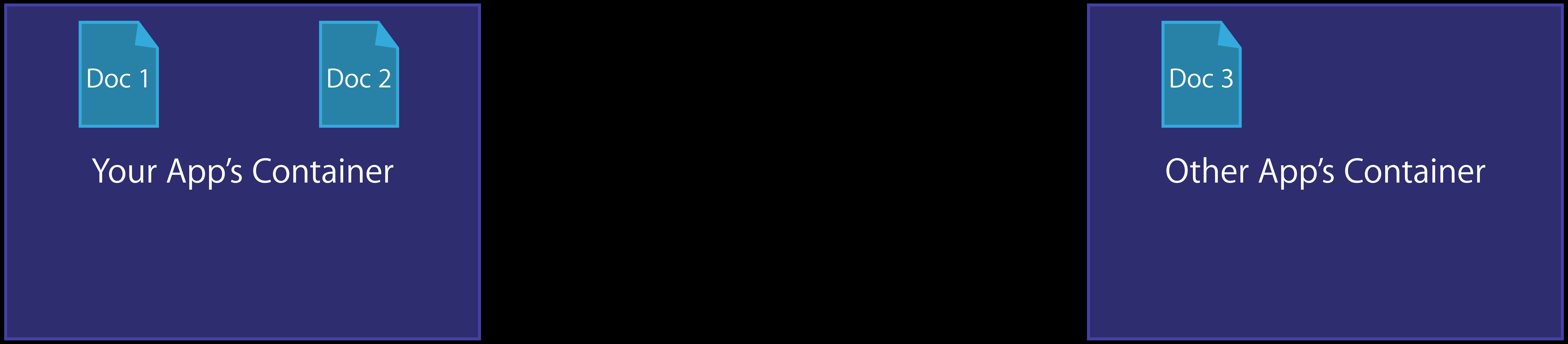


Document References and the Picker

UIDocumentPickerMode—Import vs. Open

Open mode will create a bookmark and a security-scoped URL

Import mode will not create a bookmark



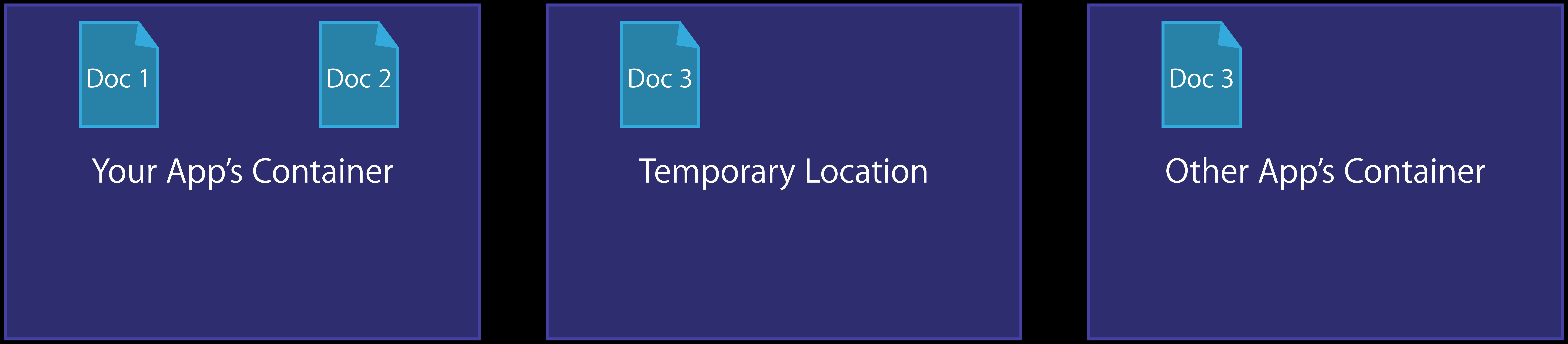
Document References and the Picker

UIDocumentPickerMode—Import vs. Open

Open mode will create a bookmark and a security-scoped URL

Import mode will not create a bookmark

You can read the file, but it is temporary



Document References and the Picker

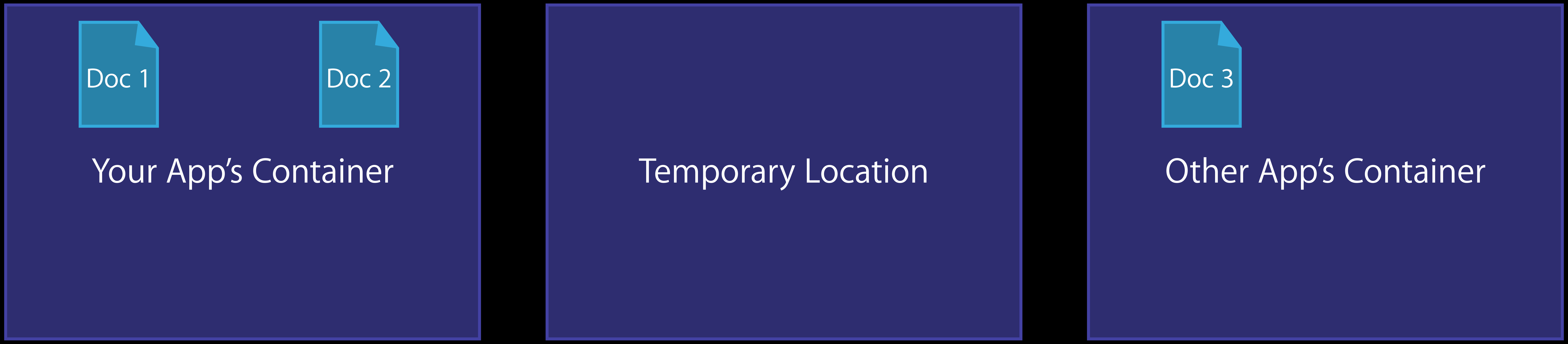
UIDocumentPickerMode—Import vs. Open

Open mode will create a bookmark and a security-scoped URL

Import mode will not create a bookmark

You can read the file, but it is temporary

Make a copy soon if you want to hang on to it



How to Create a New Document

No write access except in your sandbox

How to Create a New Document

No write access except in your sandbox

You will not have general access to other app’s containers

How to Create a New Document

No write access except in your sandbox

You will not have general access to other app’s containers

Write in your own container locally or in a temporary location

How to Create a New Document

No write access except in your sandbox

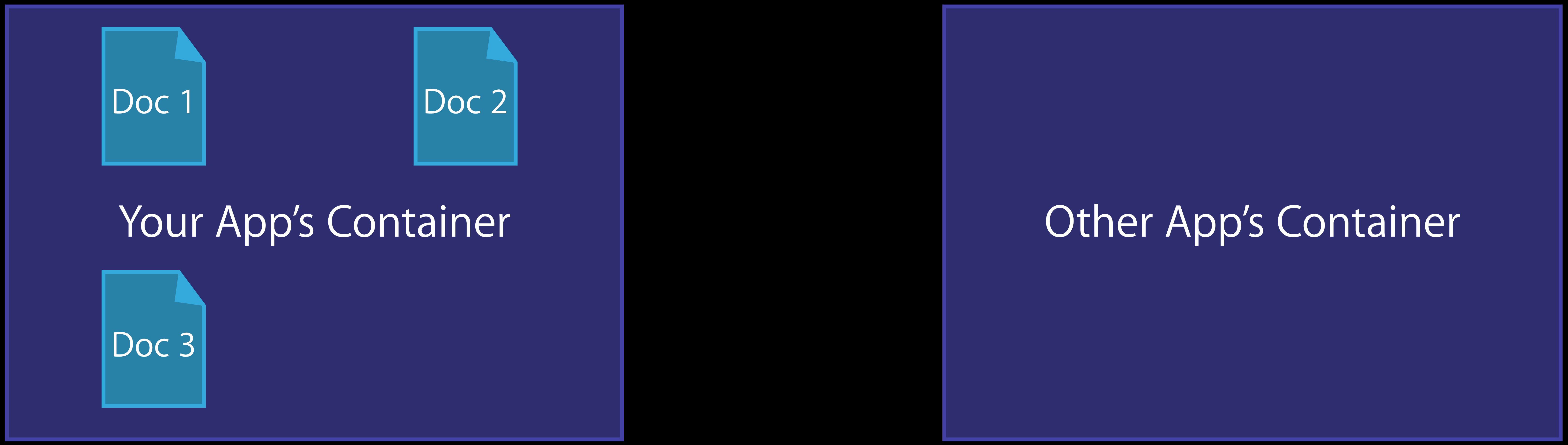
You will not have general access to other app’s containers

Write in your own container locally or in a temporary location

Afterwards, you can move the document

Existing Documents

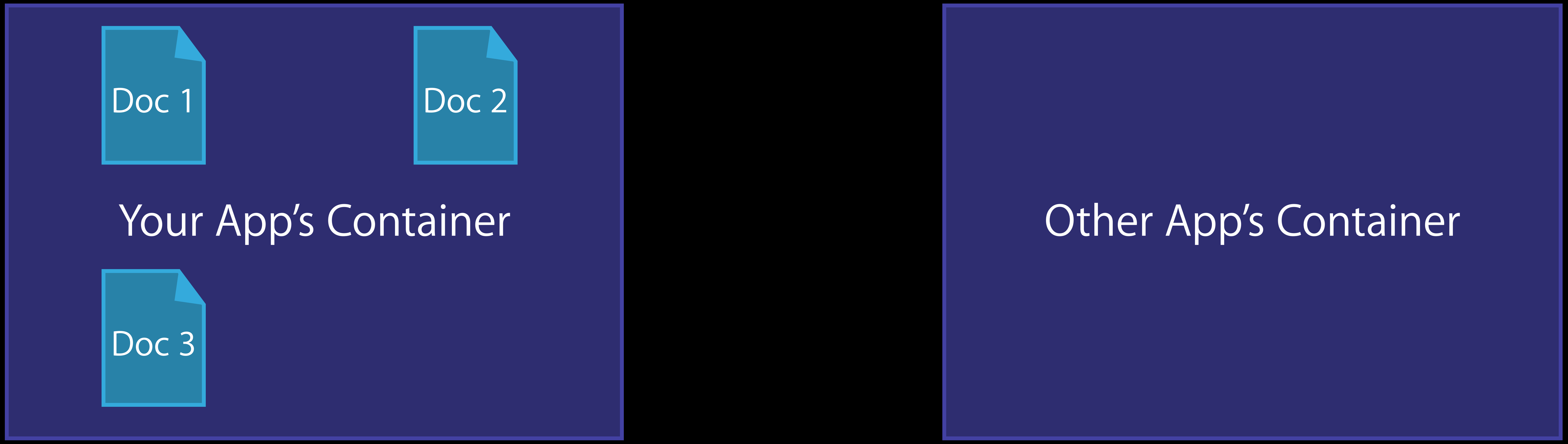
Moving a document



Existing Documents

Moving a document

Use UIDocumentPicker -initWithURL: and your existing document

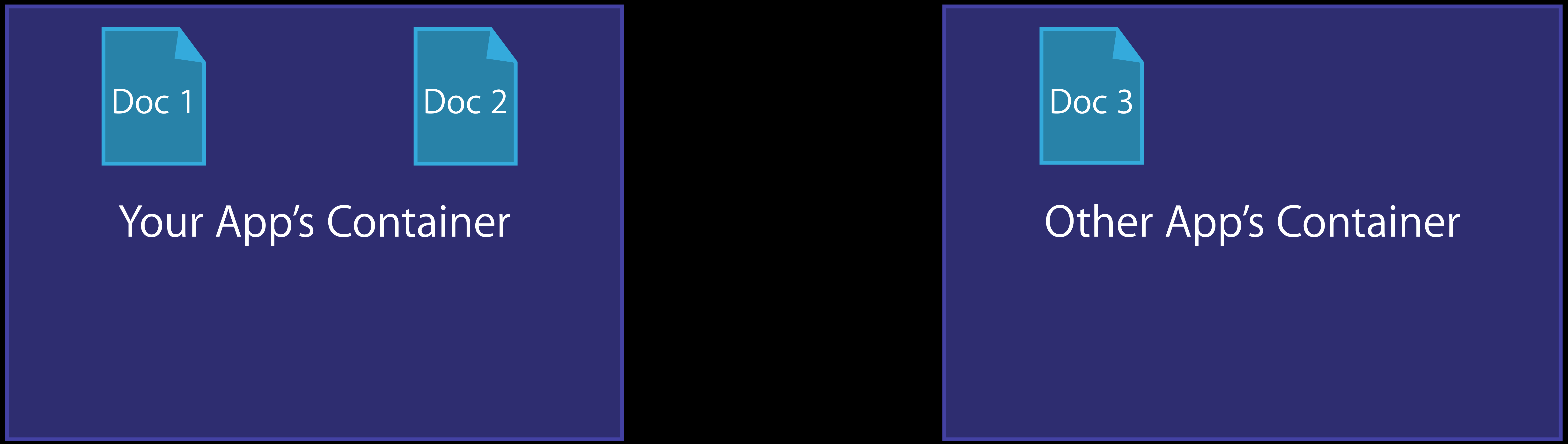


Existing Documents

Moving a document

Use UIDocumentPicker -initWithURL: and your existing document

The file will be moved to the location the user picks



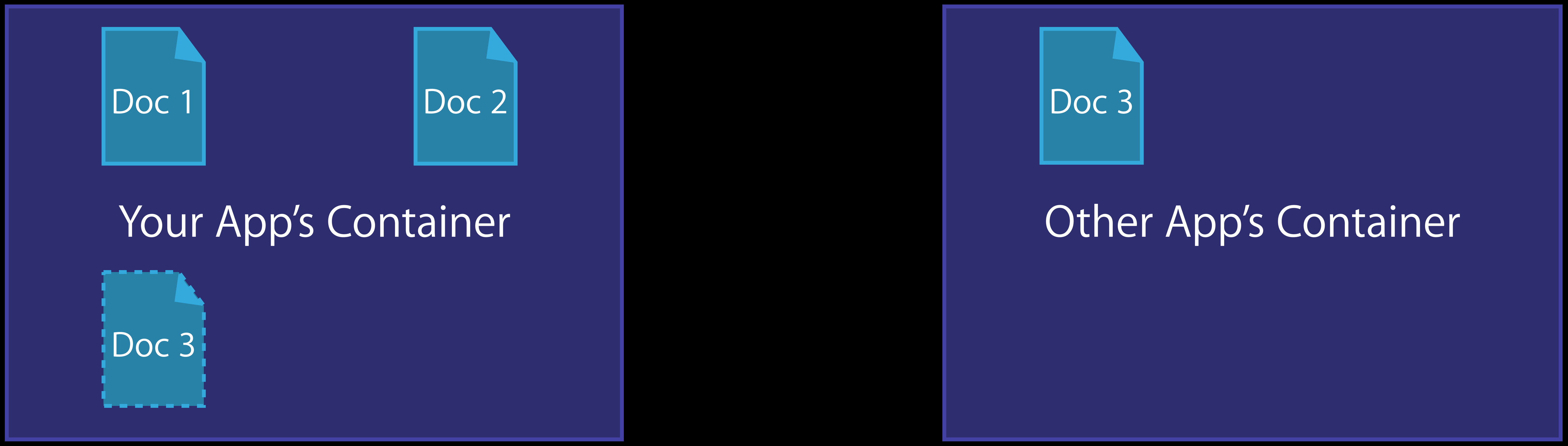
Existing Documents

Moving a document

Use UIDocumentPicker -initWithURL: and your existing document

The file will be moved to the location the user picks

You will get a URL and a reference for the new location



Existing Documents

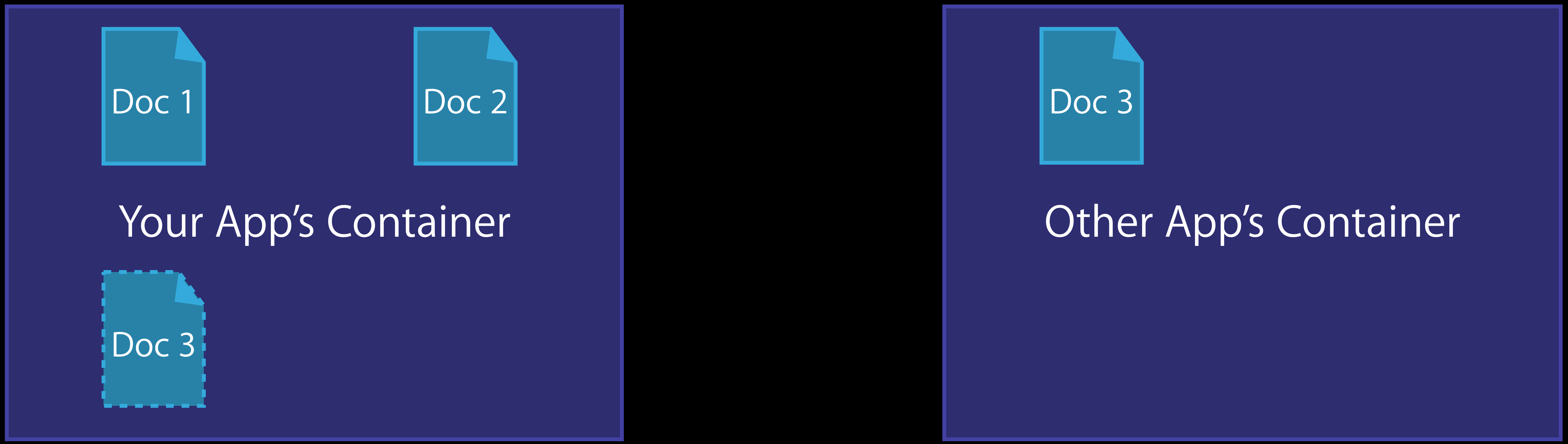
Moving a document

Use UIDocumentPicker -initWithURL: and your existing document

The file will be moved to the location the user picks

You will get a URL and a reference for the new location

The URL behaves like the one returned by Open



Existing Documents

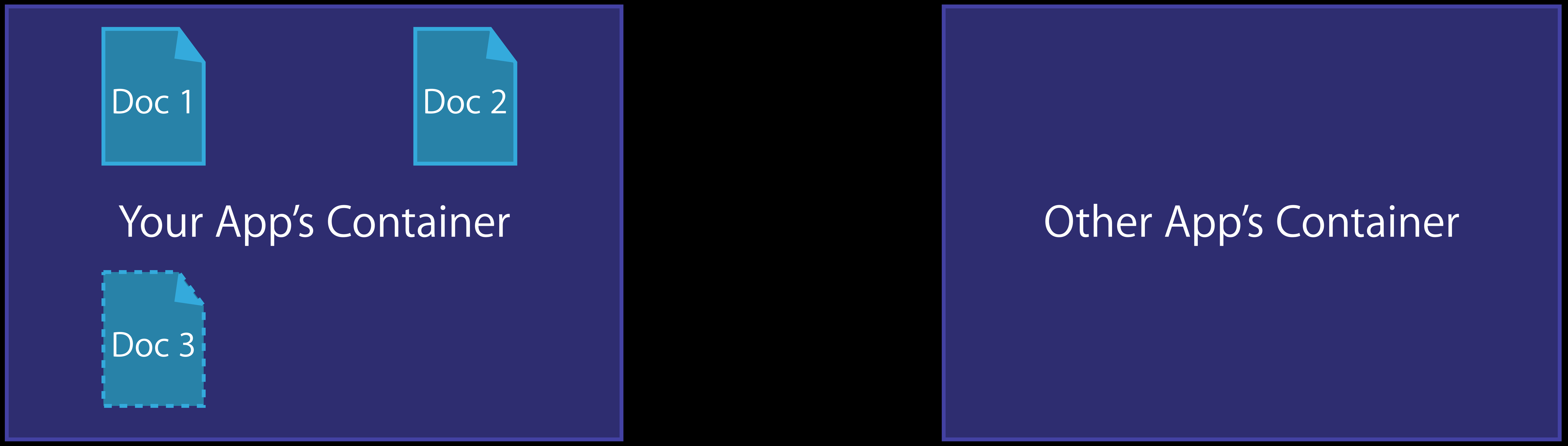
Moving a document

Use UIDocumentPicker -initWithURL: and your existing document

The file will be moved to the location the user picks

You will get a URL and a reference for the new location

The URL behaves like the one returned by Open



Existing Documents

Moving a document

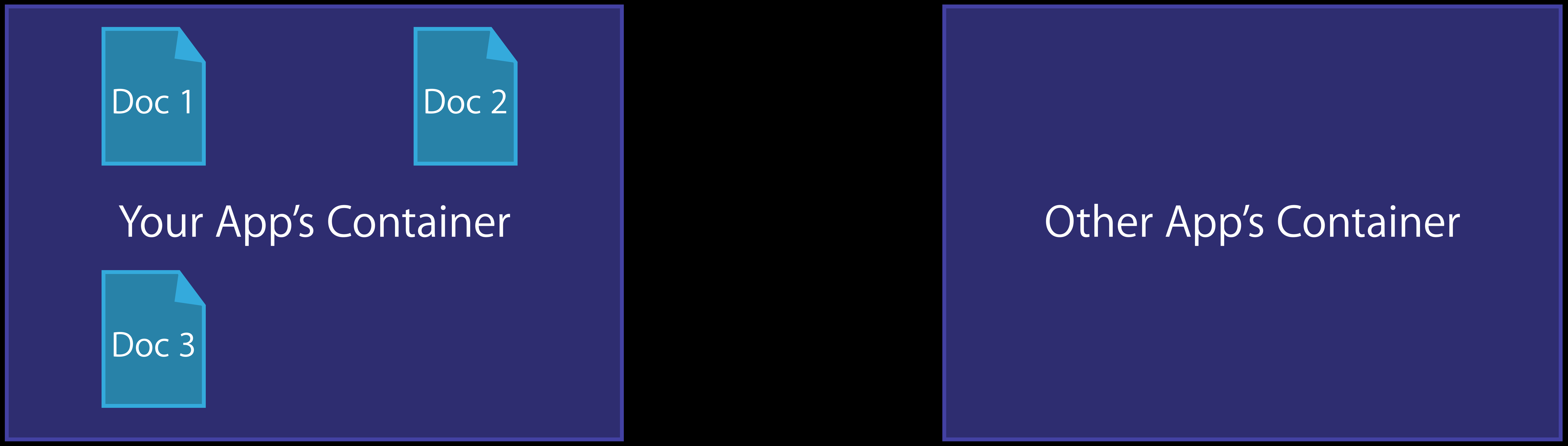
Use UIDocumentPicker -initWithURL: and your existing document

The file will be moved to the location the user picks

You will get a URL and a reference for the new location

The URL behaves like the one returned by Open

Export mode if you don't want to hang on to the reference



Existing Documents

Moving a document

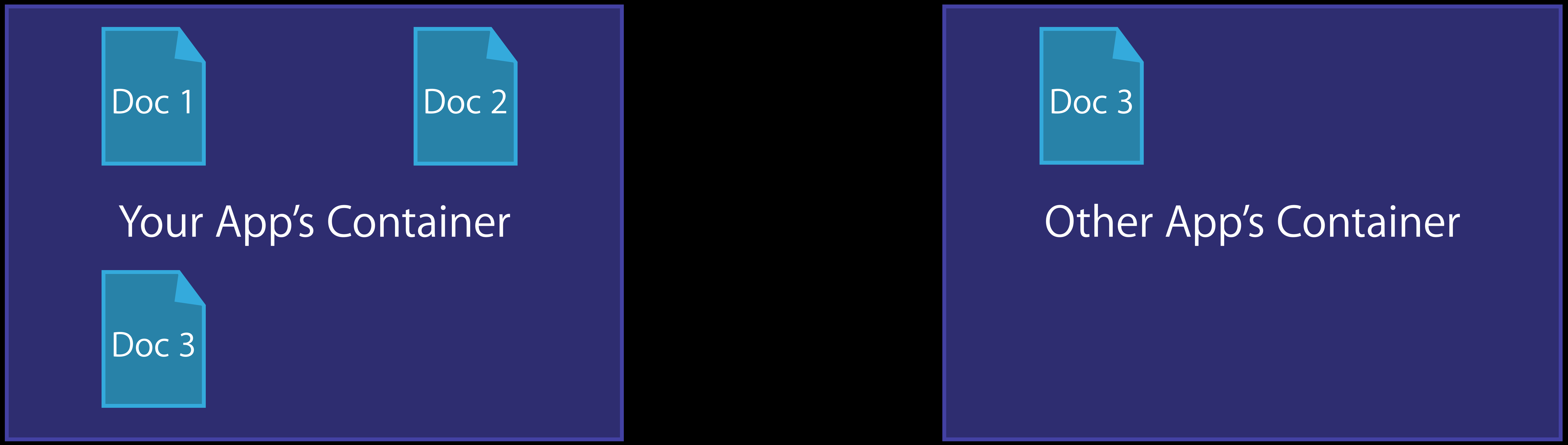
Use UIDocumentPicker -initWithURL: and your existing document

The file will be moved to the location the user picks

You will get a URL and a reference for the new location

The URL behaves like the one returned by Open

Export mode if you don't want to hang on to the reference



Document Management

Summary

Document Management

Summary

Document references for enumerating picked documents

Document Management

Summary

Document references for enumerating picked documents

Bookmarks for state restoration

Document Management

Summary

Document references for enumerating picked documents

Bookmarks for state restoration

Modes for Import, Open, Export, Move

Document Management

Summary

Document references for enumerating picked documents

Bookmarks for state restoration

Modes for Import, Open, Export, Move

Allows the user to access all their iCloud documents

Document Provider Extensions

Providing document storage to other apps

What Is a Document Provider?

Alternative storage locations

What Is a Document Provider?

Alternative storage locations

NEW

Alternative way for a third-party to provide document storage

What Is a Document Provider?

Alternative storage locations

NEW

Alternative way for a third-party to provide document storage

Same perspective from the host app’s point of view

What Is a Document Provider?

Alternative storage locations

NEW

Alternative way for a third-party to provide document storage

Same perspective from the host app's point of view

Selectable from the document picker

What Is a Document Provider?

Alternative storage locations

NEW

Alternative way for a third-party to provide document storage

Same perspective from the host app's point of view

Selectable from the document picker

Can offer the same picker modes as the standard picker

What Is a Document Provider?

Alternative storage locations

NEW

Alternative way for a third-party to provide document storage

Same perspective from the host app's point of view

Selectable from the document picker

Can offer the same picker modes as the standard picker

Implemented using two extensions

What Is a Document Provider?

Alternative storage locations

NEW

- Alternative way for a third-party to provide document storage
- Same perspective from the host app's point of view
- Selectable from the document picker
- Can offer the same picker modes as the standard picker
- Implemented using two extensions

What Is a Document Provider?

Alternative storage locations

NEW

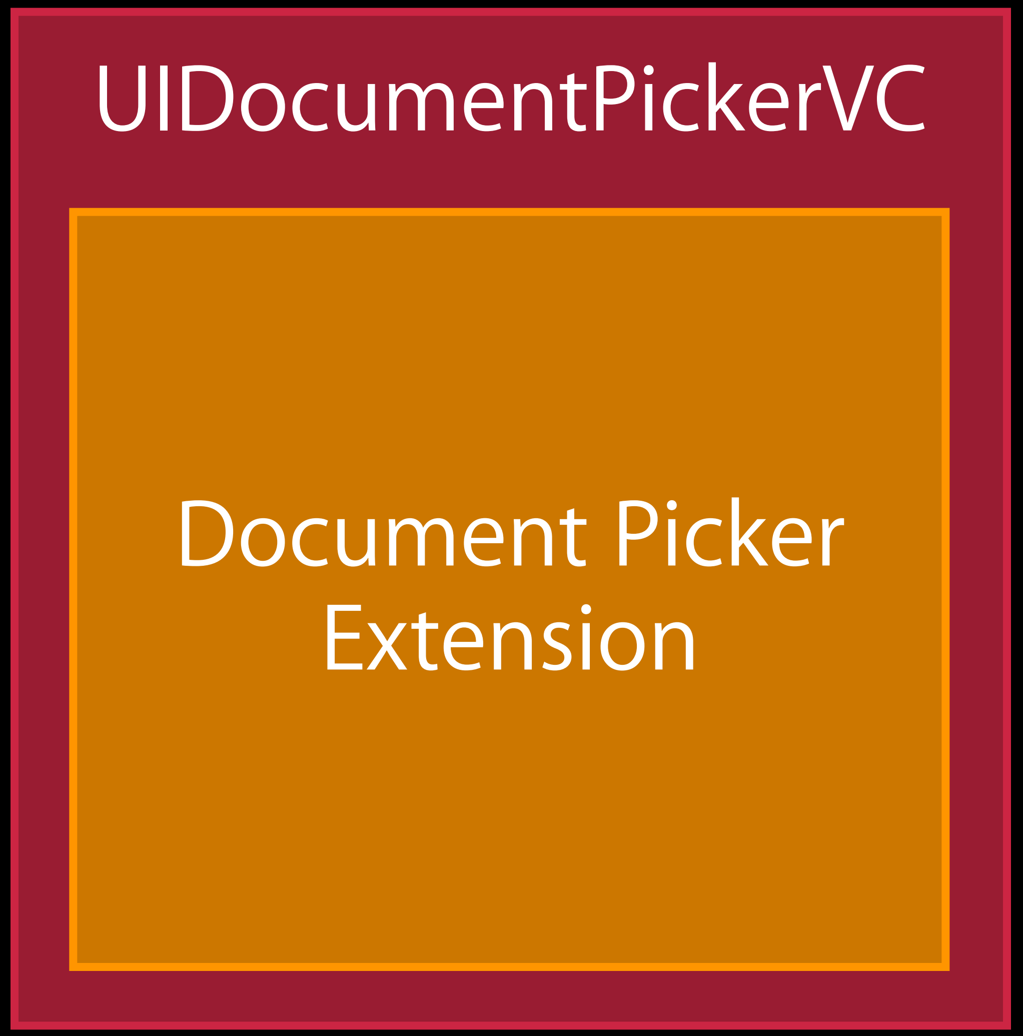
Alternative way for a third-party to provide document storage

Same perspective from the host app's point of view

Selectable from the document picker

Can offer the same picker modes as the standard picker

Implemented using two extensions



What Is a Document Provider?

Alternative storage locations

NEW

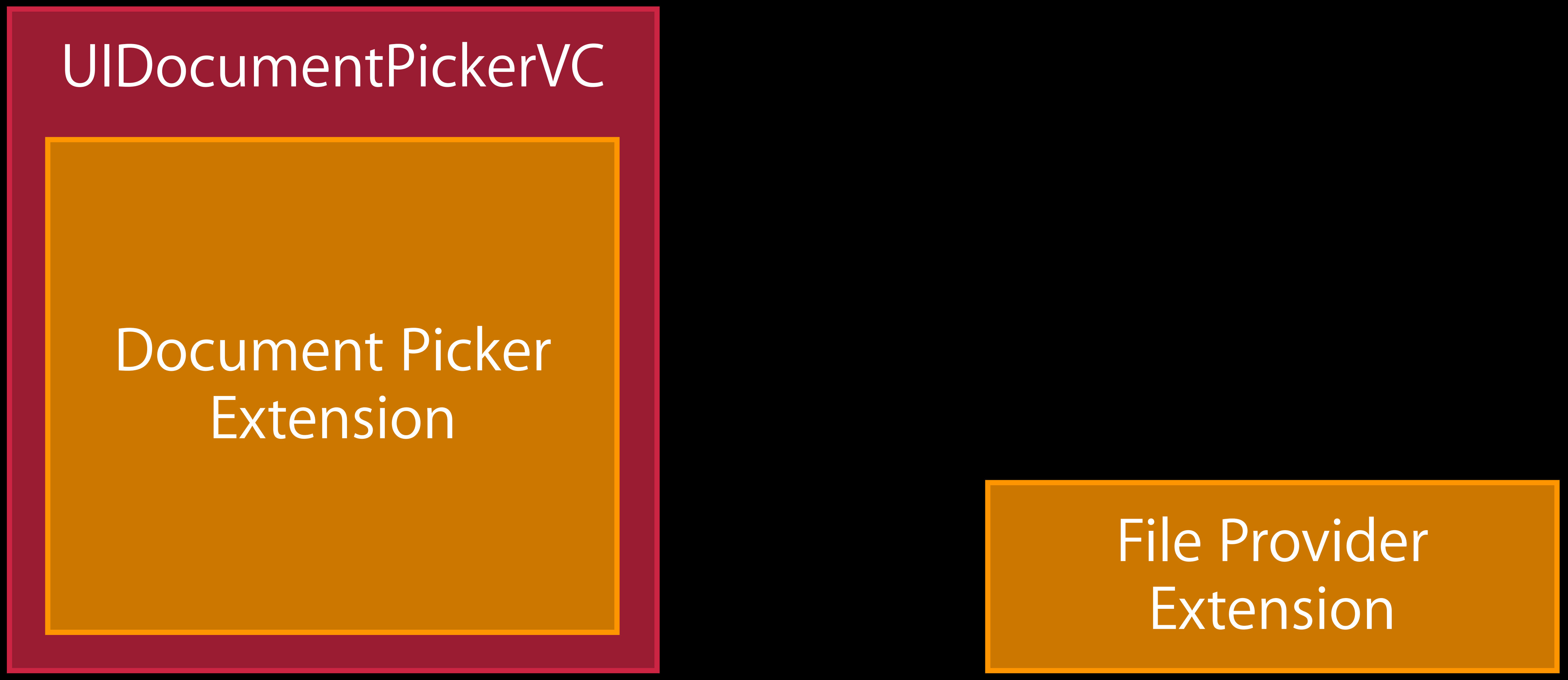
Alternative way for a third-party to provide document storage

Same perspective from the host app's point of view

Selectable from the document picker

Can offer the same picker modes as the standard picker

Implemented using two extensions



What Is a Document Provider?

Alternative storage locations

NEW

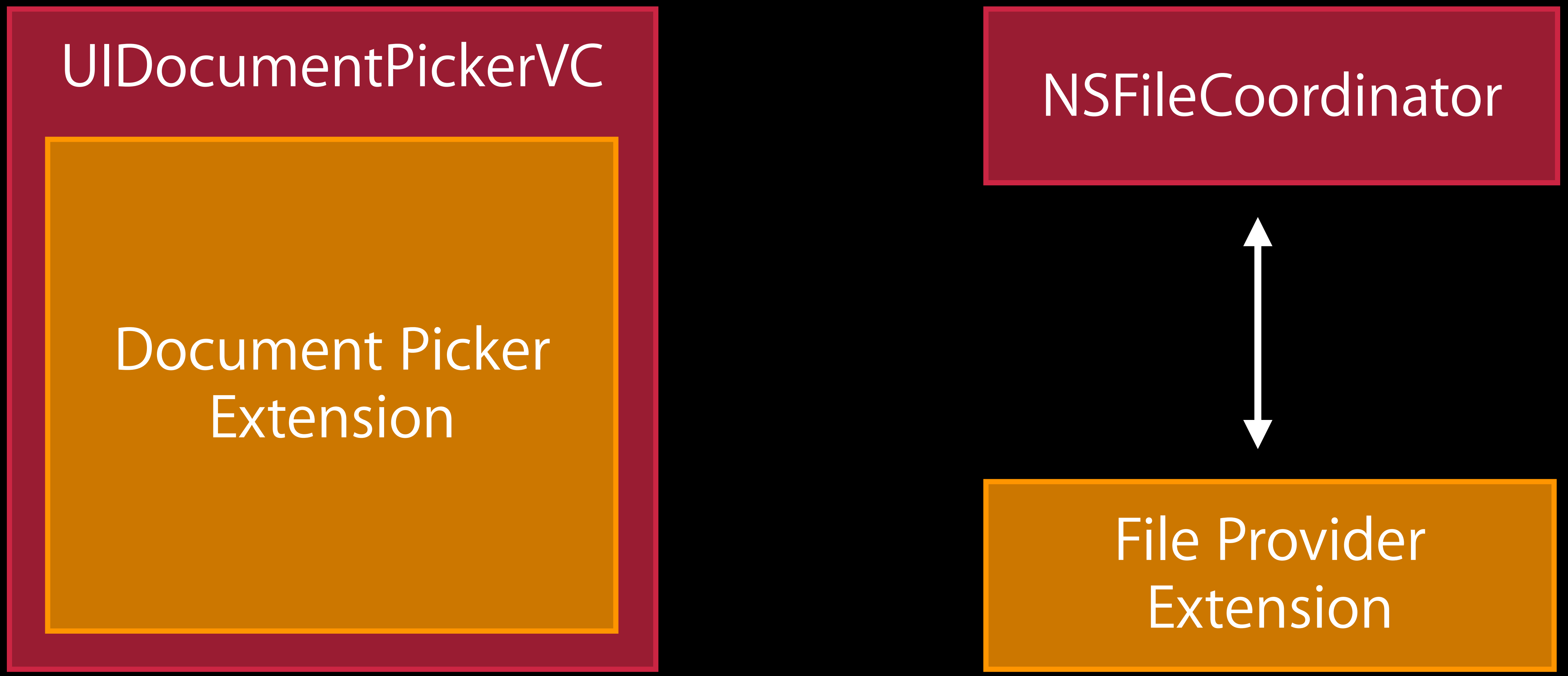
Alternative way for a third-party to provide document storage

Same perspective from the host app's point of view

Selectable from the document picker

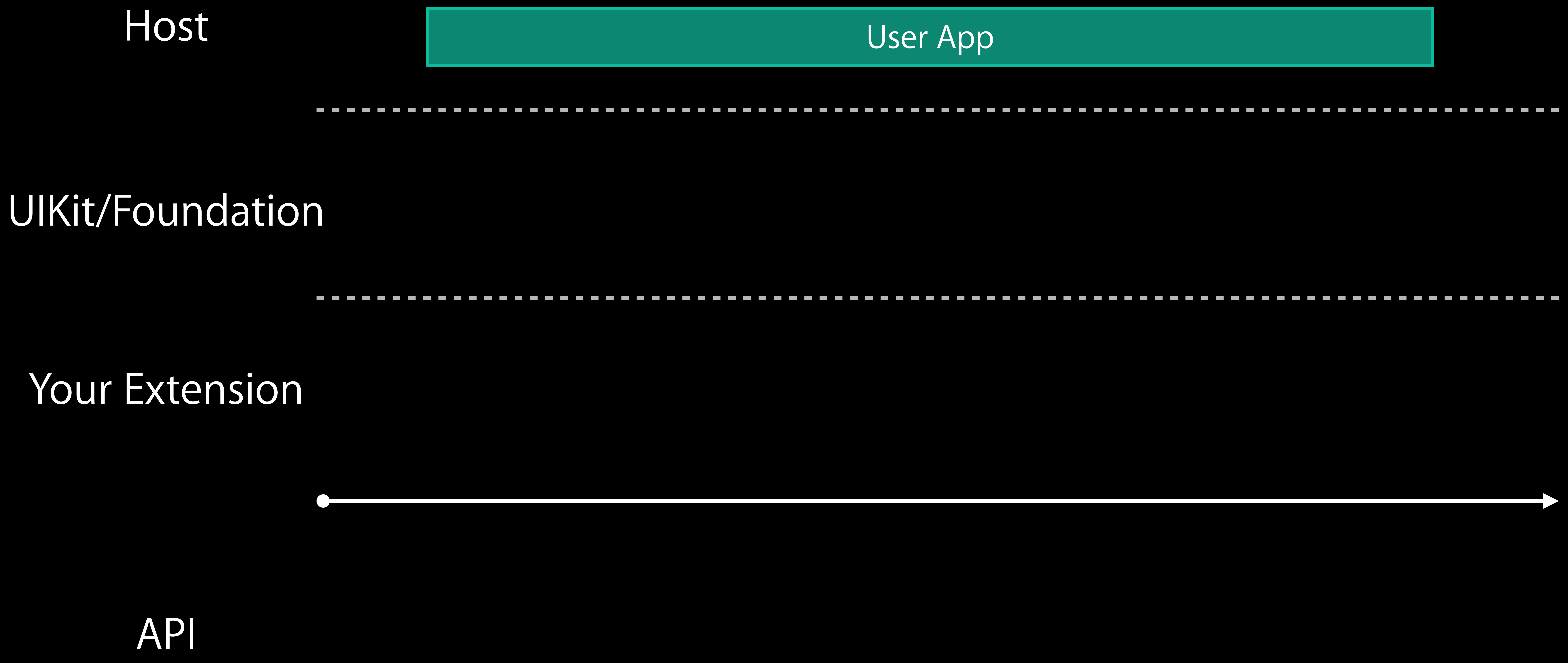
Can offer the same picker modes as the standard picker

Implemented using two extensions



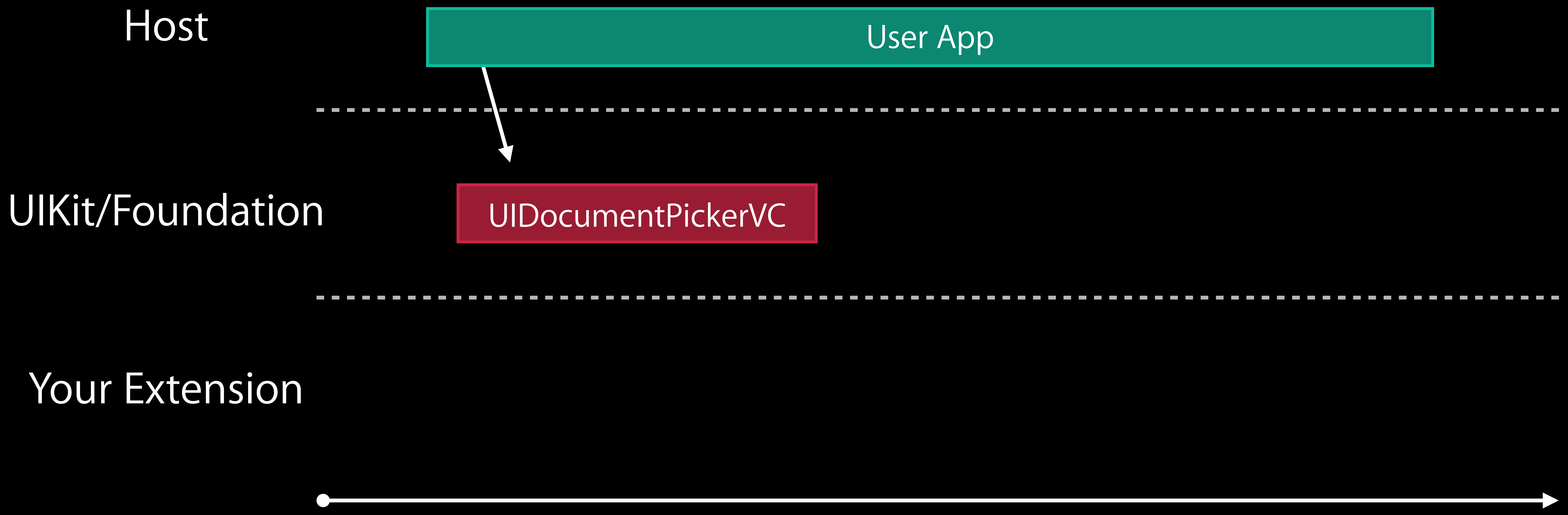
Document Provider Extensions

Data flow for selection and read access



Document Provider Extensions

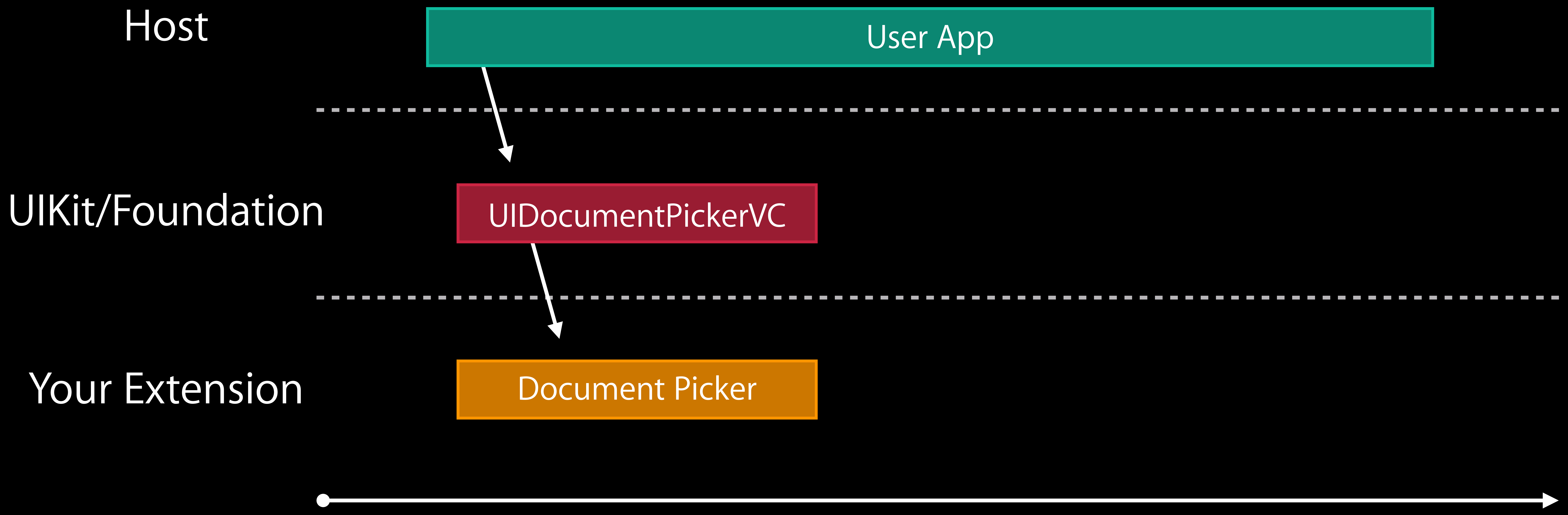
Data flow for selection and read access



API `initWithDocumentTypes:inMode:`

Document Provider Extensions

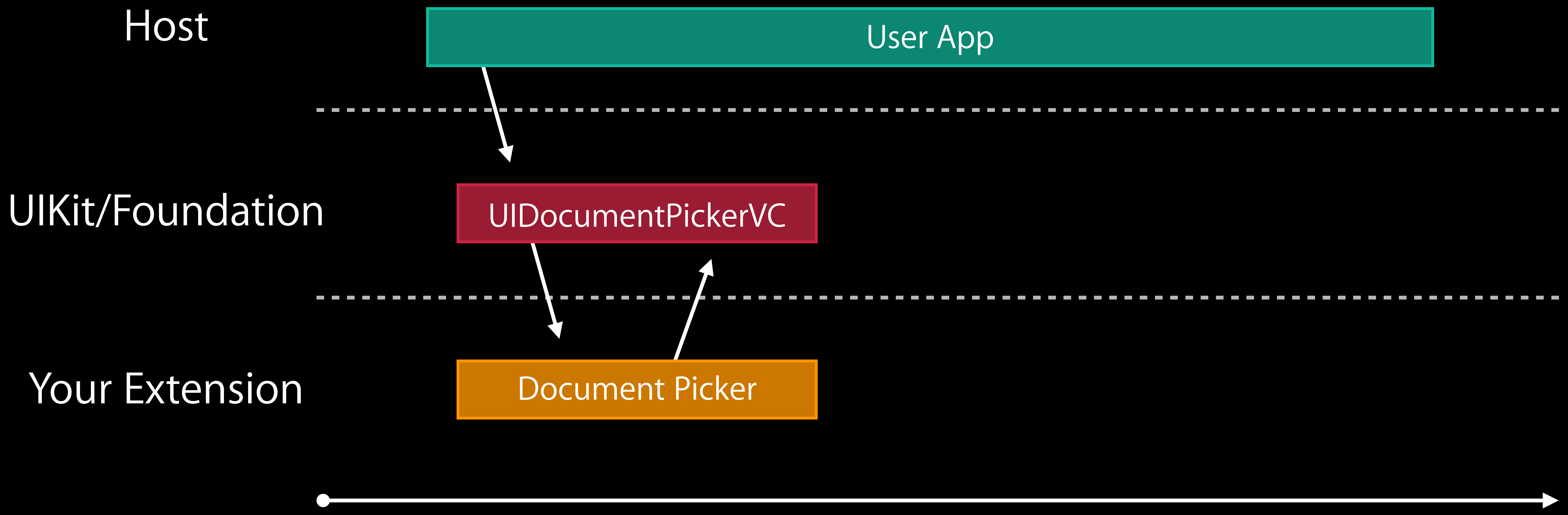
Data flow for selection and read access



API `prepareForPresentationInMode:`

Document Provider Extensions

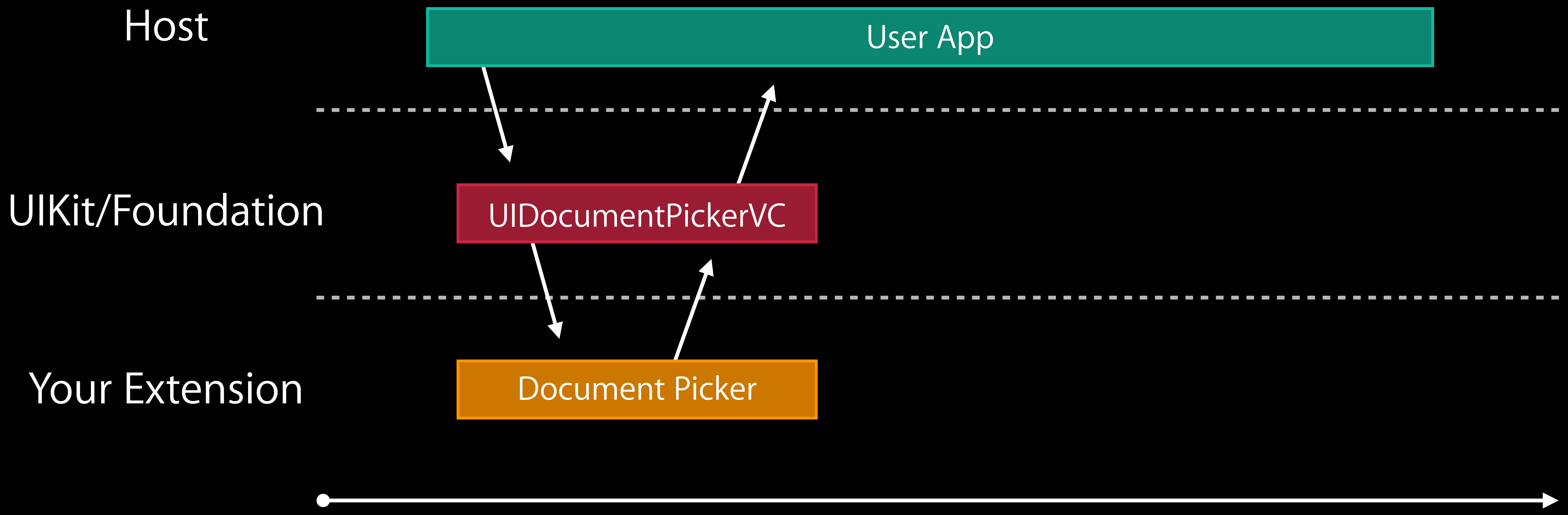
Data flow for selection and read access



API `dismissGrantingAccessToURL:`

Document Provider Extensions

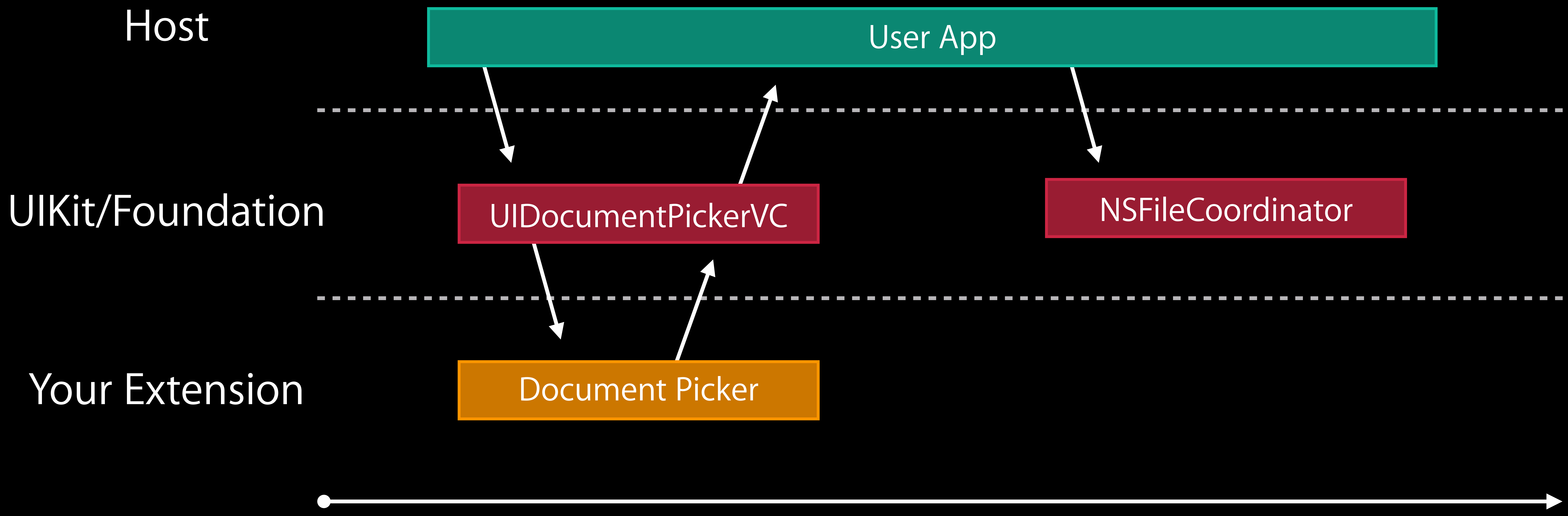
Data flow for selection and read access



API `documentPicker:didPickDocumentAtURL:`

Document Provider Extensions

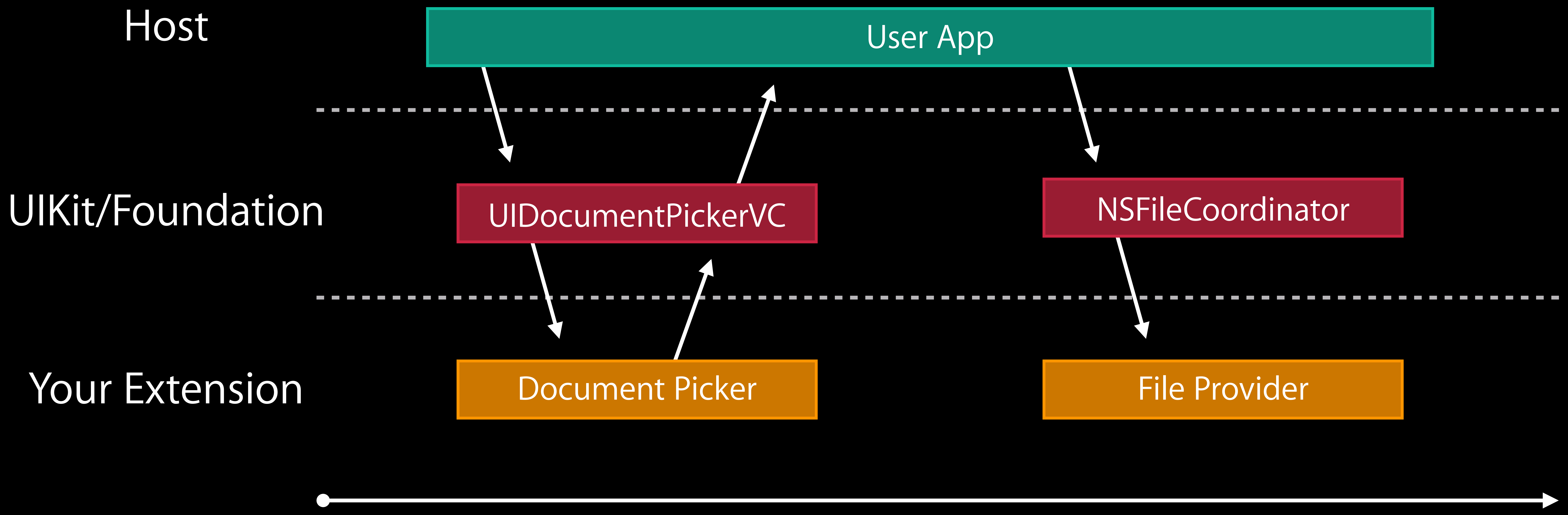
Data flow for selection and read access



API `coordinateReadingItemAtURL:`

Document Provider Extensions

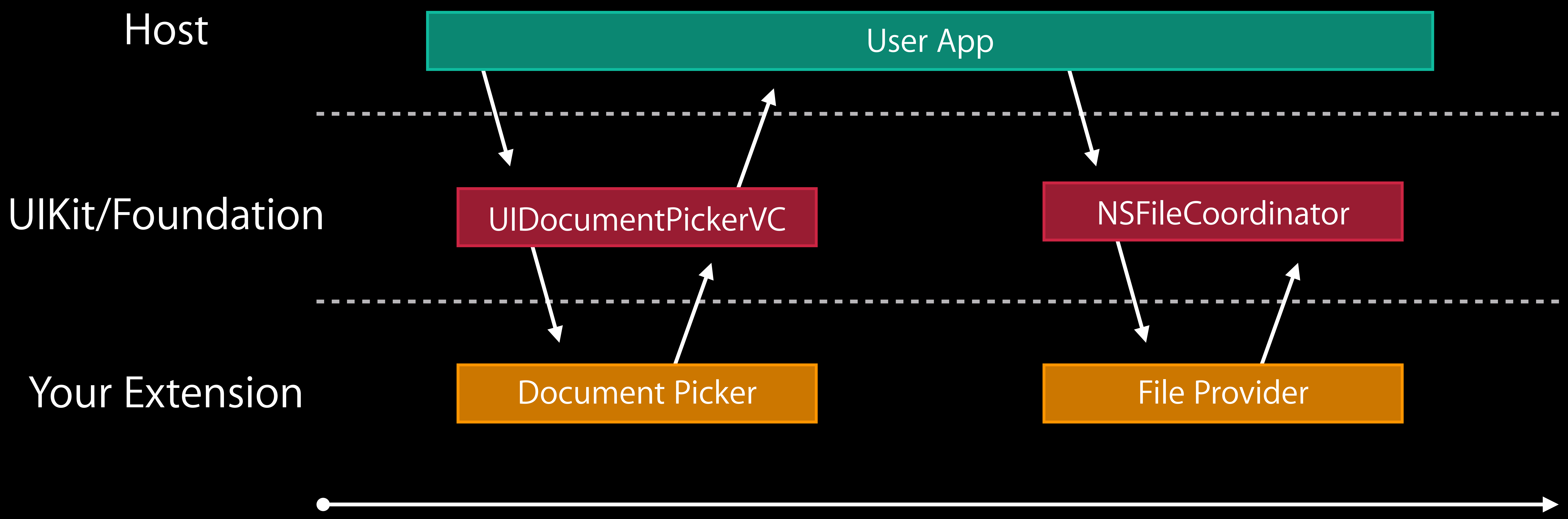
Data flow for selection and read access



API `startProvidingItemAtURL:`

Document Provider Extensions

Data flow for selection and read access

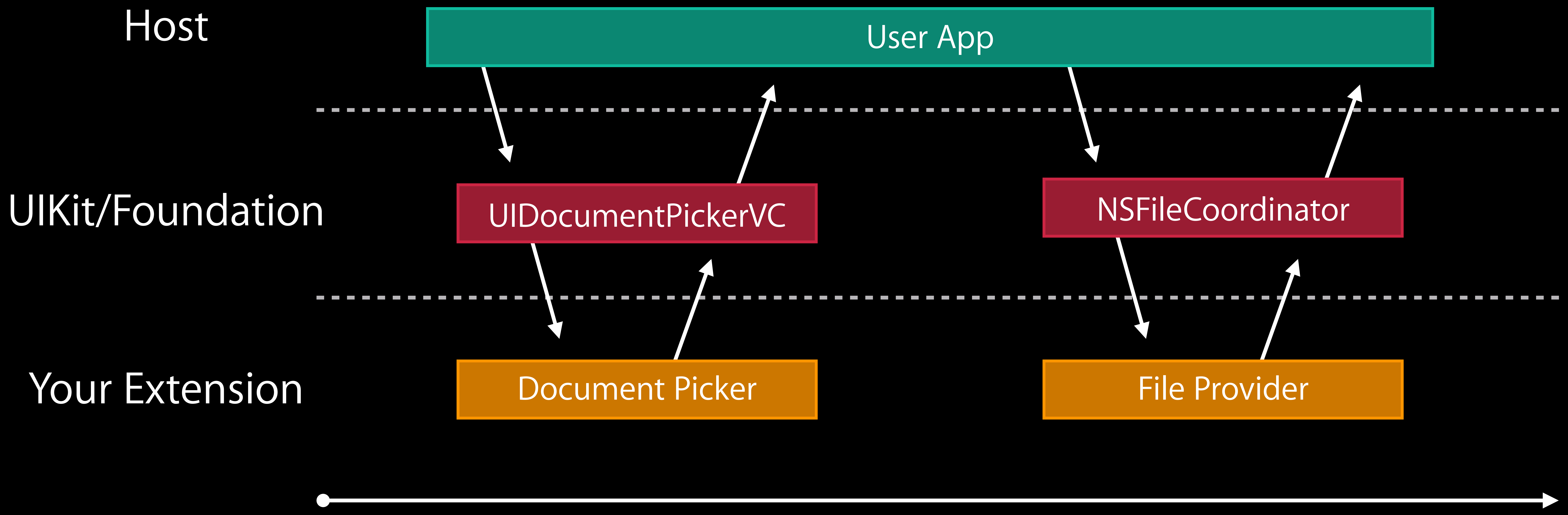


API

```
completion()
```

Document Provider Extensions

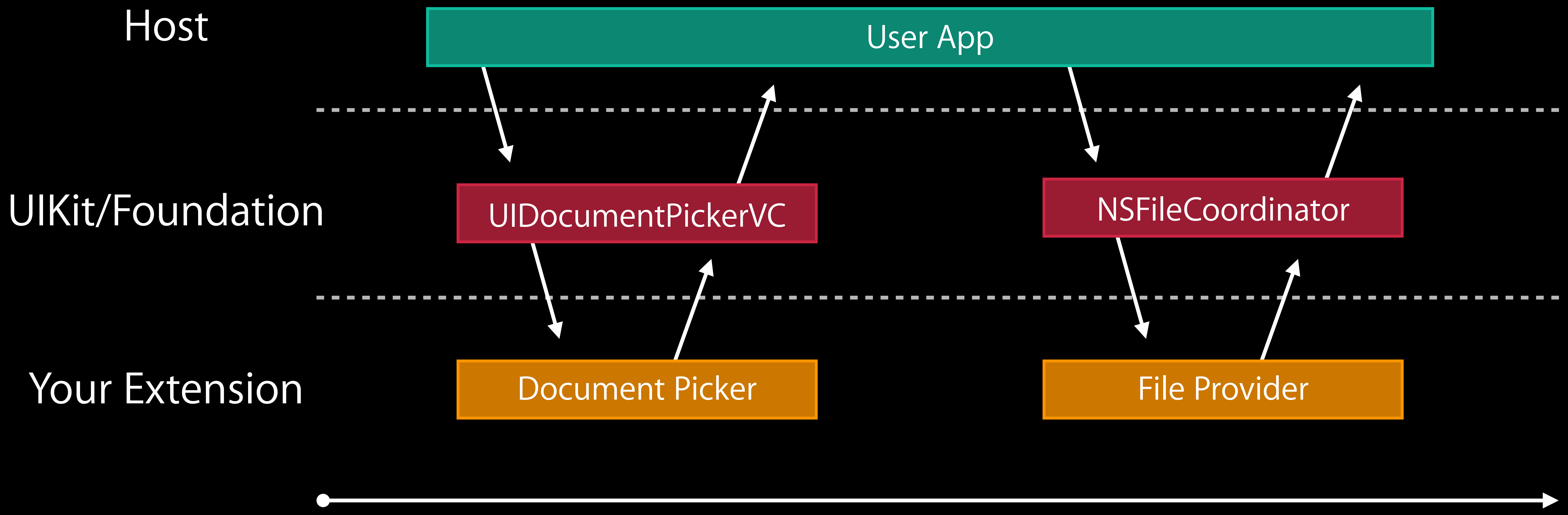
Data flow for selection and read access



API `accessorBlock()`

Document Provider Extensions

Data flow for selection and read access



API `dataWithContentsOfURL:`

Document Provider Extensions

Security and bookmarks

Document Provider Extensions

Security and bookmarks

Document provider extensions don’t take part in document reference mechanism

Document Provider Extensions

Security and bookmarks

Document provider extensions don’t take part in document reference mechanism

But bookmarks still work

Document Provider Extensions

Security and bookmarks

Document provider extensions don’t take part in document reference mechanism

But bookmarks still work

File provider returns/resolves an identifier

Document Provider Extensions

Security and bookmarks

Document provider extensions don’t take part in document reference mechanism

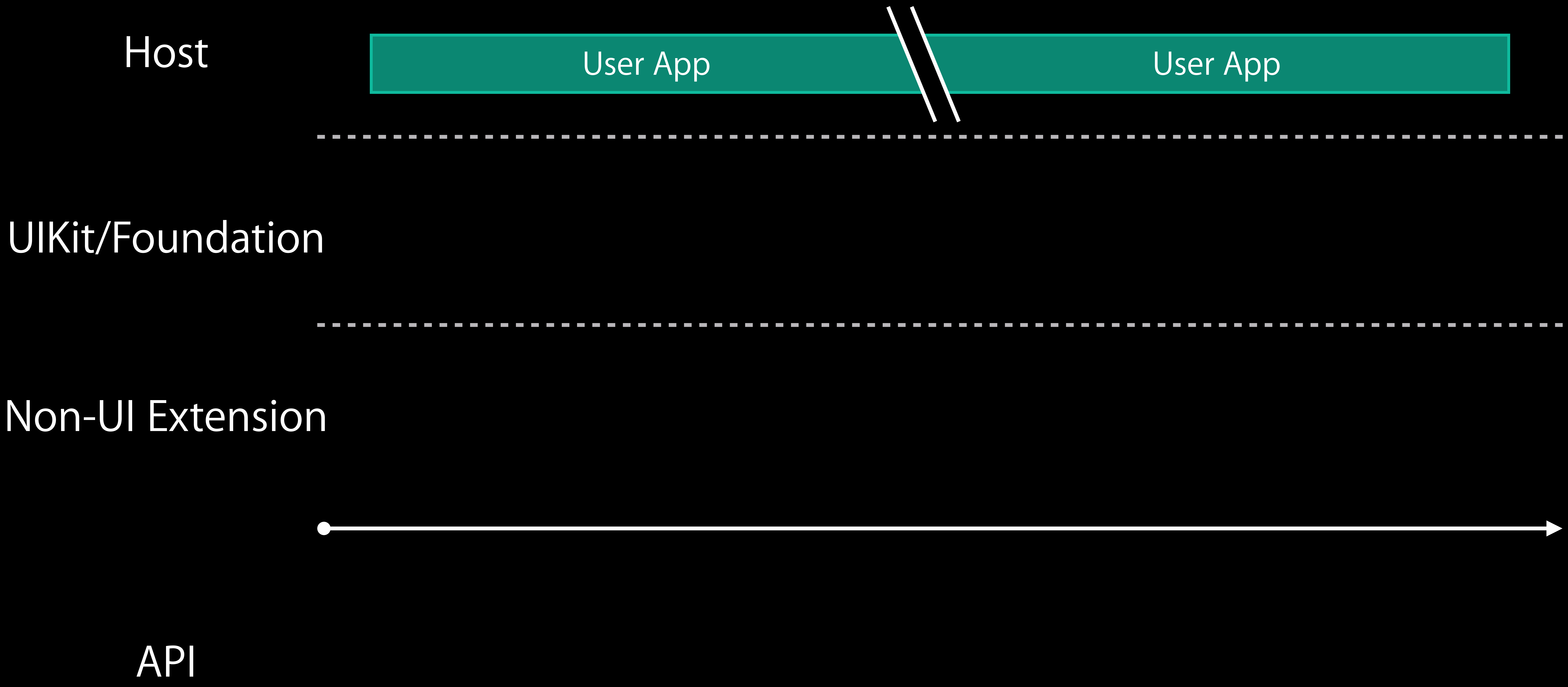
But bookmarks still work

File provider returns/resolves an identifier

Access is controlled by isolation layer

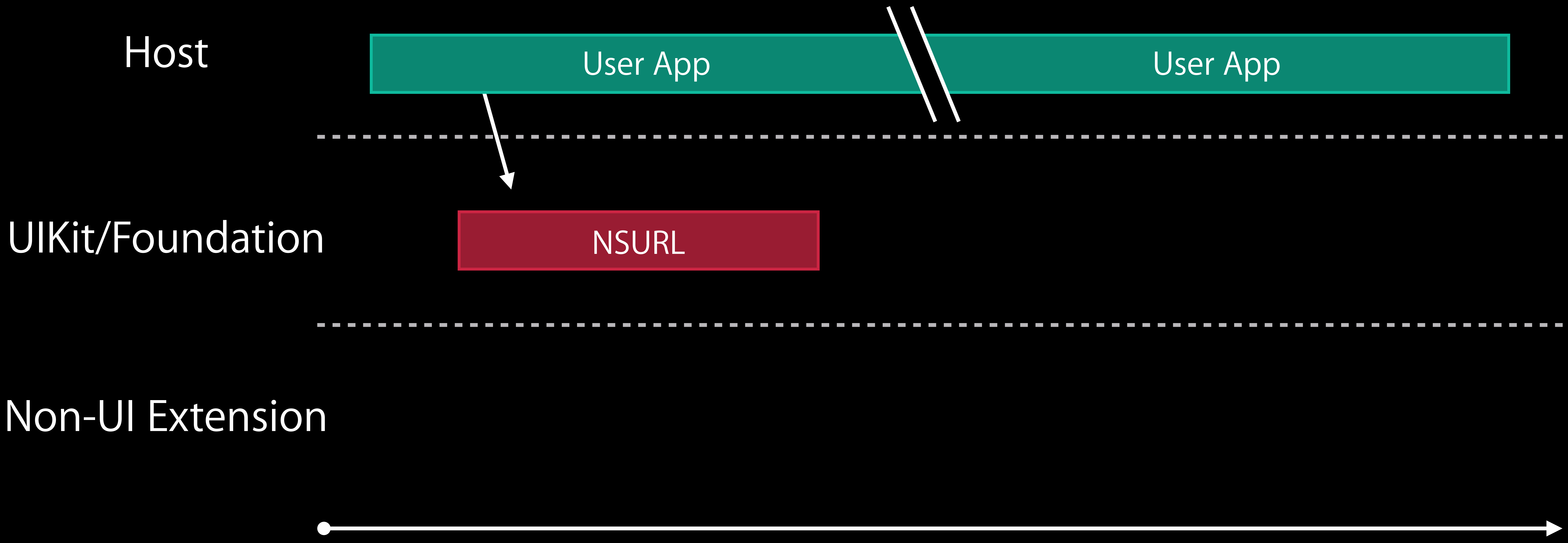
Document Provider Extensions

Bookmark data flow



Document Provider Extensions

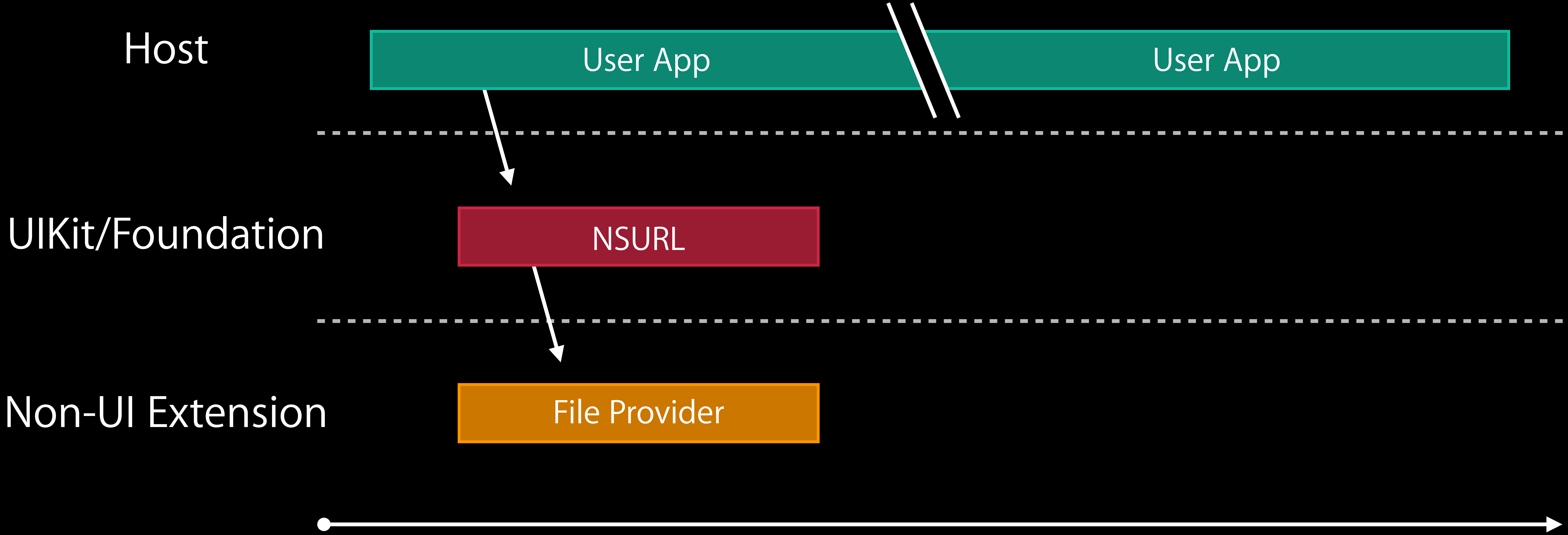
Bookmark data flow



API `bookmarkDataWithOptions:`

Document Provider Extensions

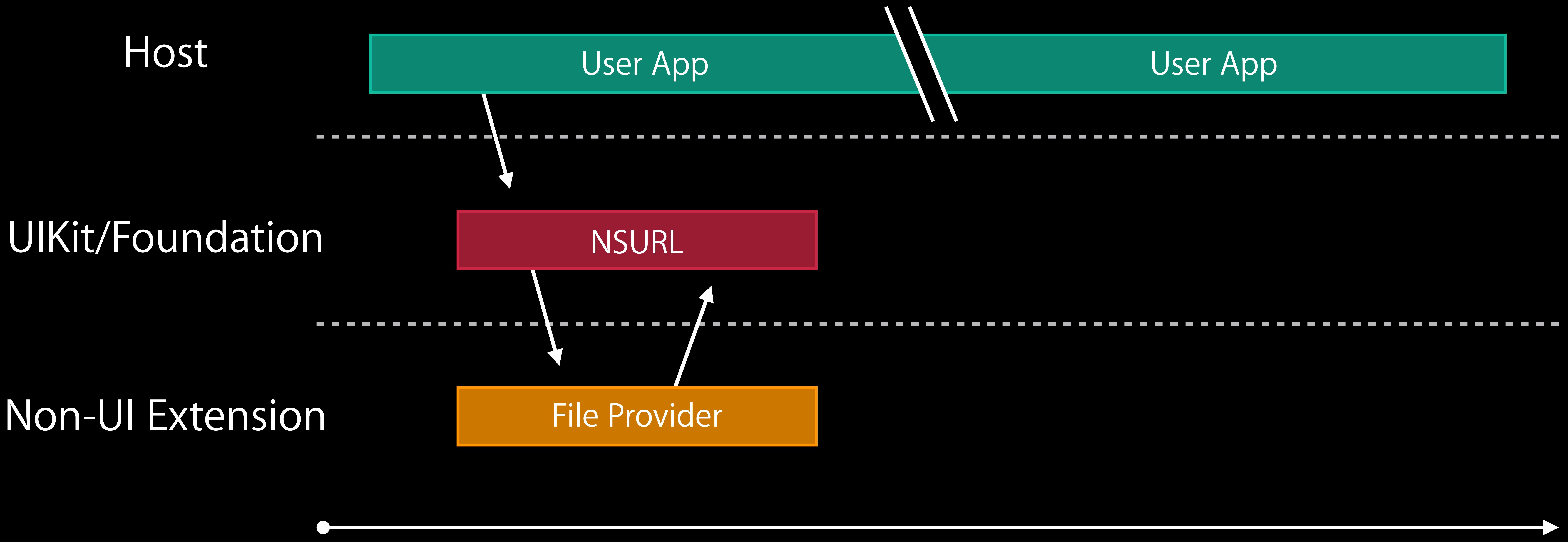
Bookmark data flow



API `persistentIdentifierForItemAtURL:`

Document Provider Extensions

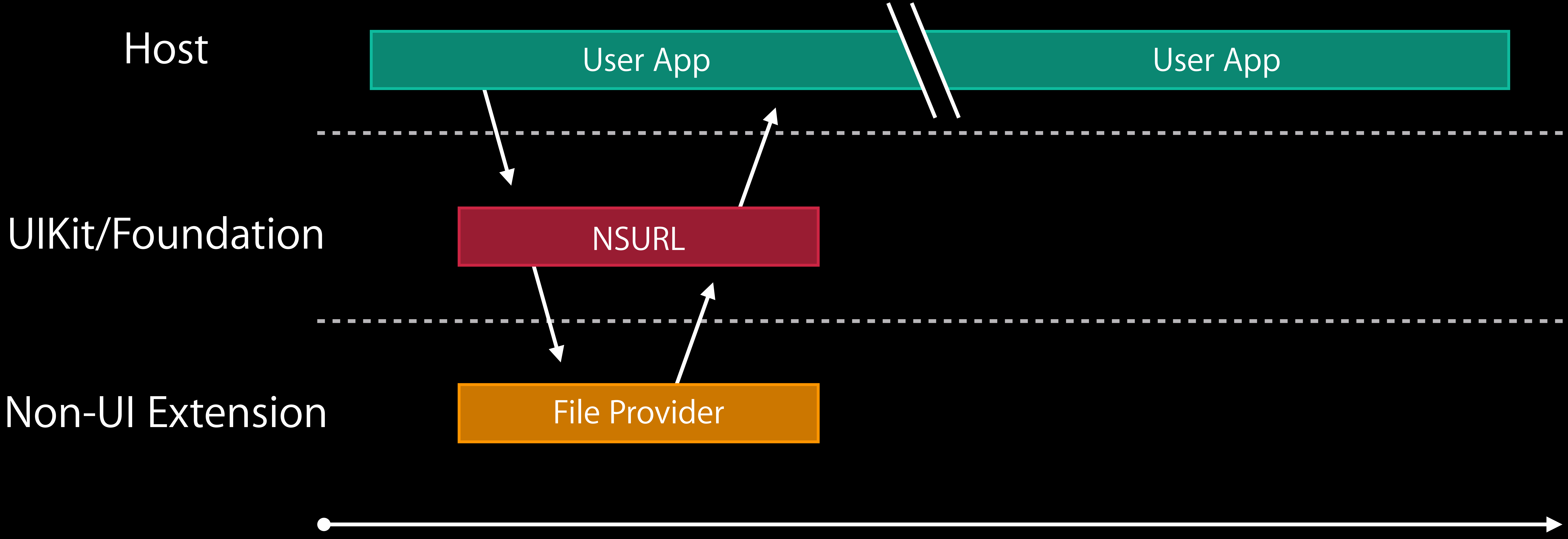
Bookmark data flow



API `NSString *identifier`

Document Provider Extensions

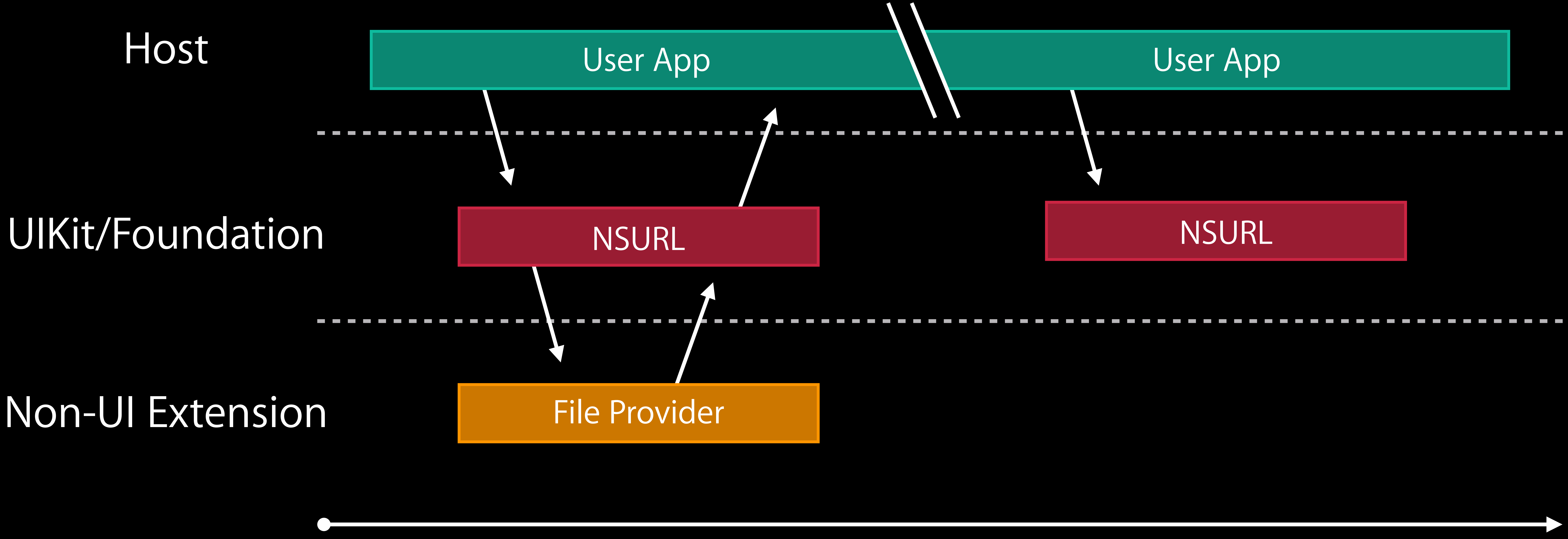
Bookmark data flow



API `NSData *bookmarkData`

Document Provider Extensions

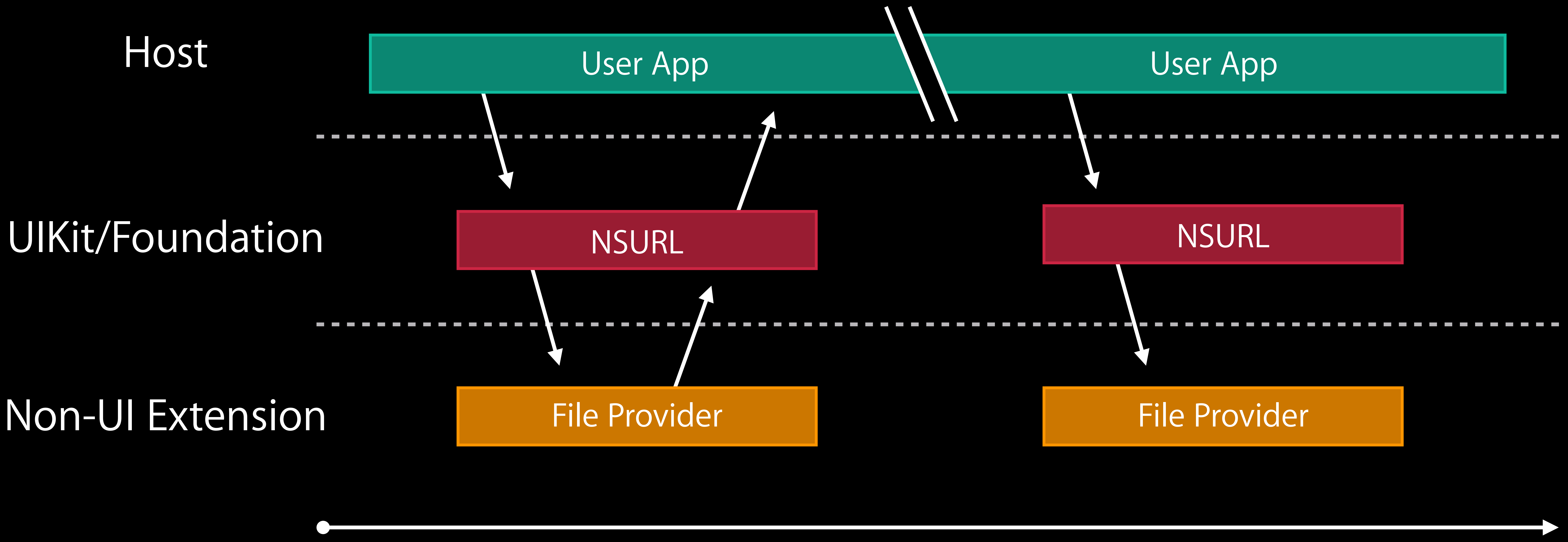
Bookmark data flow



API `URLByResolvingBookmarkData:`

Document Provider Extensions

Bookmark data flow

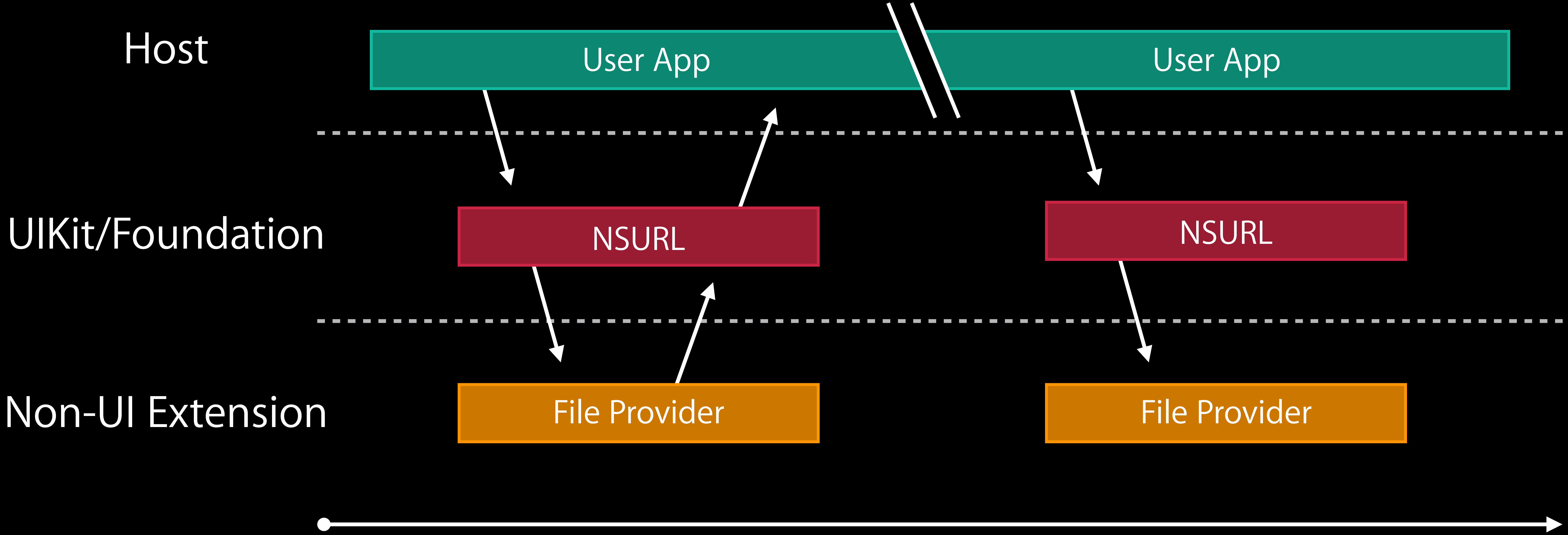


API

```
URLForItemWithPersistentIdentifier:
providePlaceholderAtURL:
```

Document Provider Extensions

Bookmark data flow

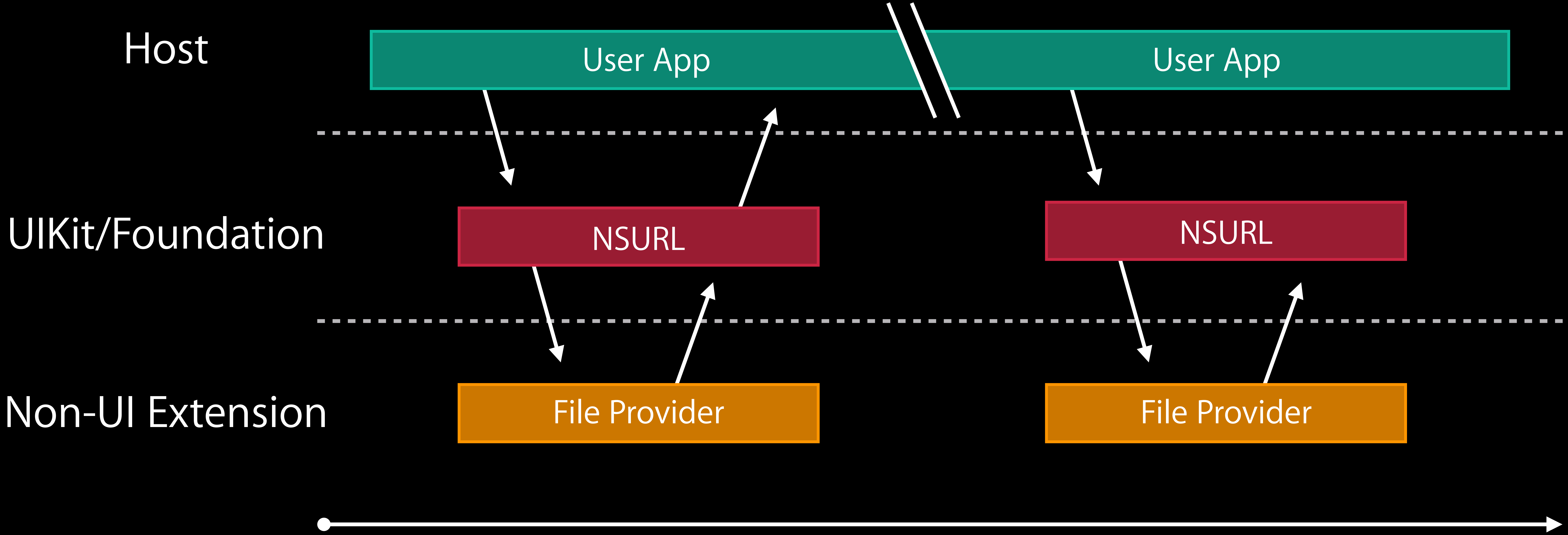


API

```
URLForItemWithPersistentIdentifier:
providePlaceholderAtURL:
```

Document Provider Extensions

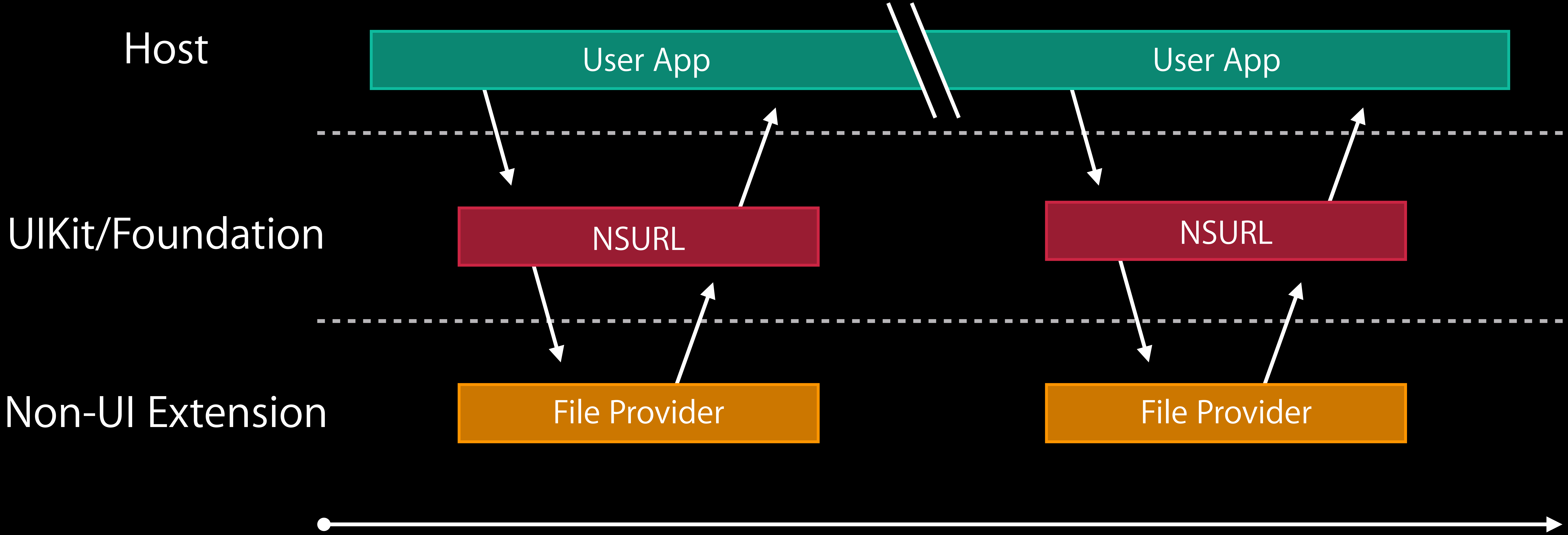
Bookmark data flow



API `NSURL *url`

Document Provider Extensions

Bookmark data flow



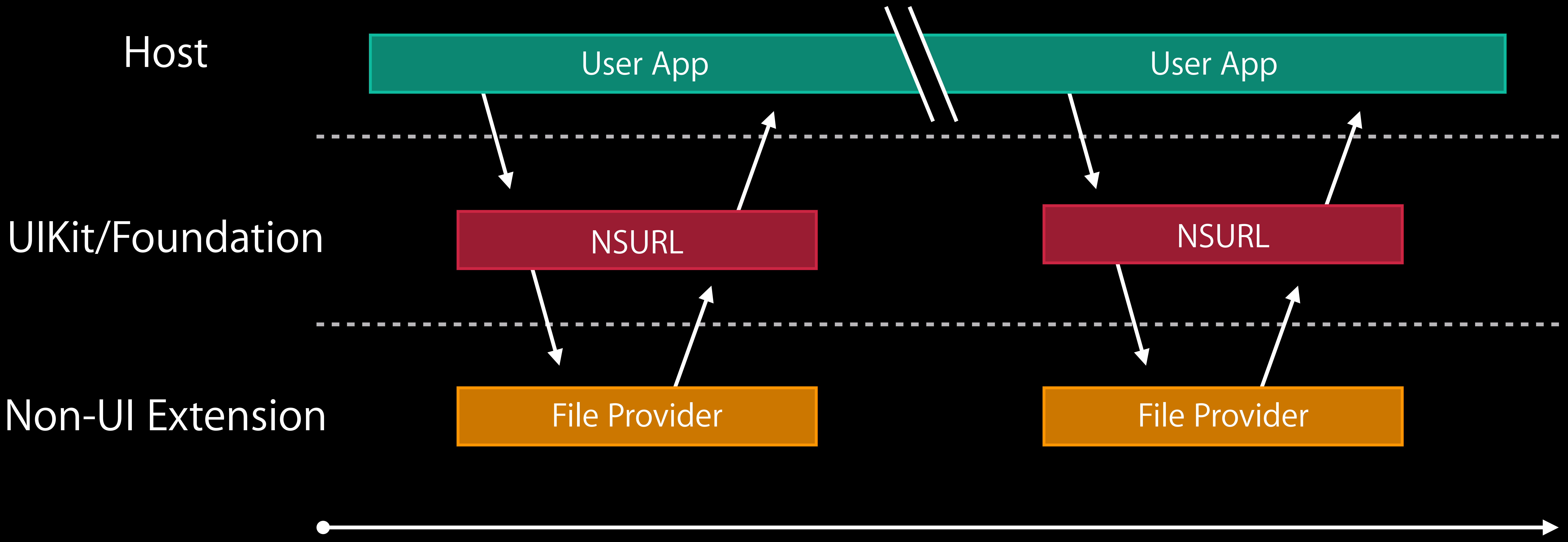
API

`getPromisedItemResourceValue:`

`NSFileCoordinator API`

Document Provider Extensions

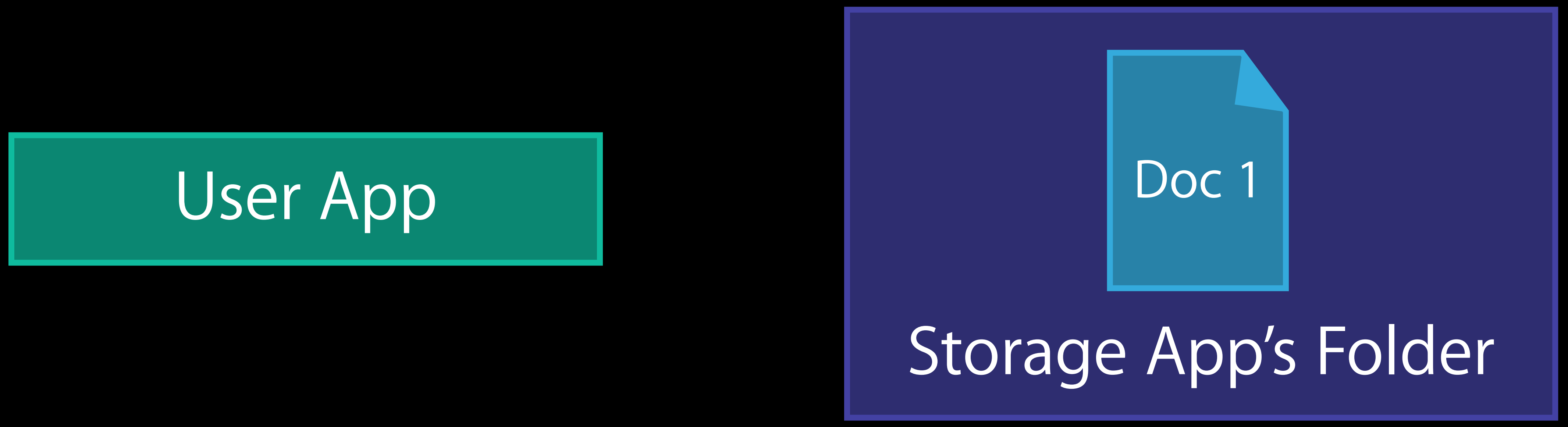
Bookmark data flow



API `getPromisedItemResourceValue:`
`NSFileCoordinator API`

Document Provider Extensions

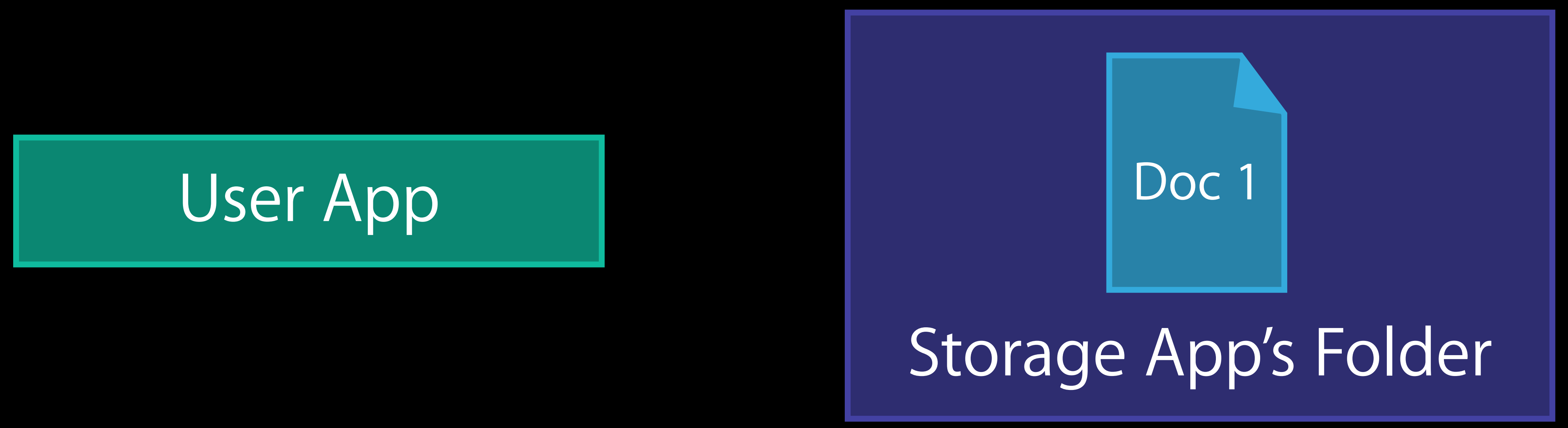
Writing files



Document Provider Extensions

Writing files

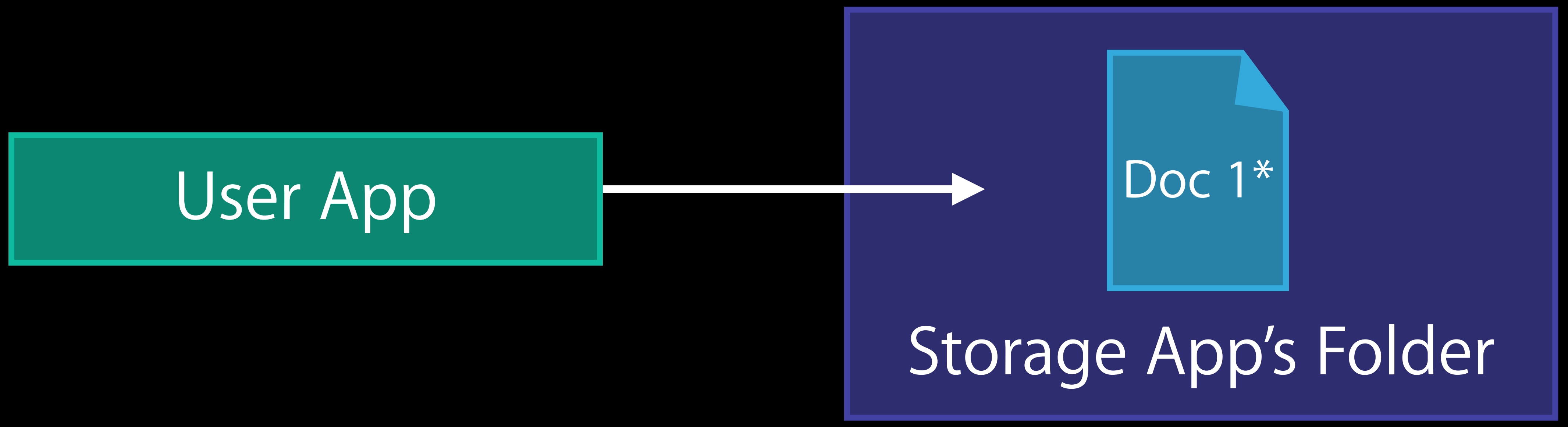
Host app uses file coordination to write file



Document Provider Extensions

Writing files

Host app uses file coordination to write file

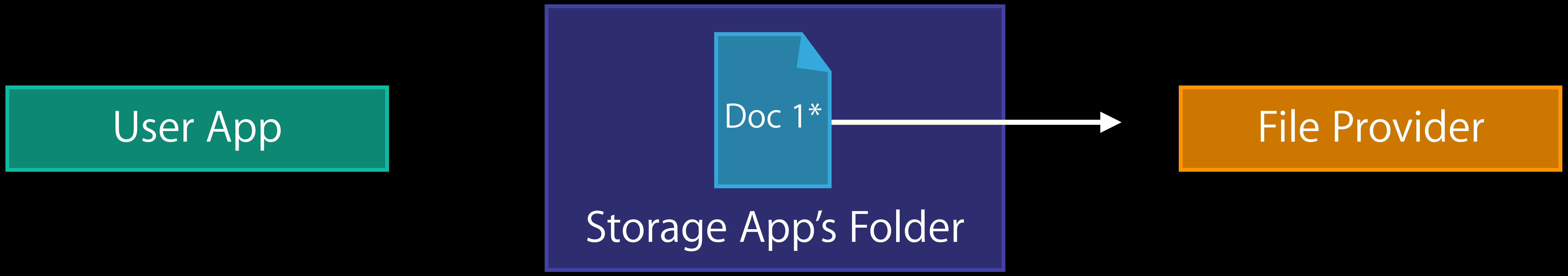


Document Provider Extensions

Writing files

Host app uses file coordination to write file

Soon after, itemChangedAtURL: is called on the file provider



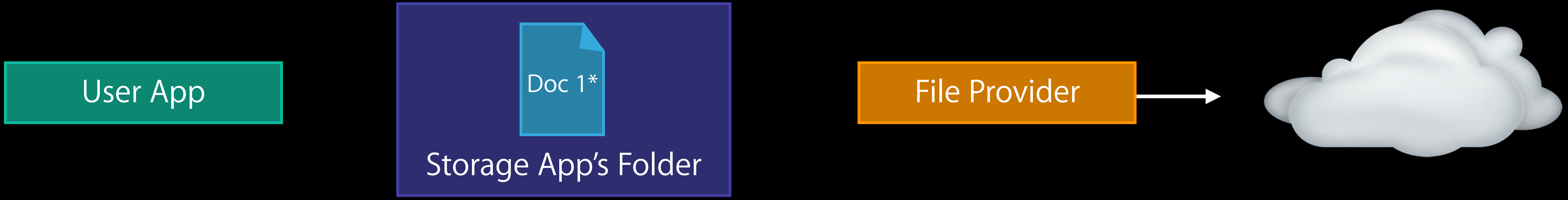
Document Provider Extensions

Writing files

Host app uses file coordination to write file

Soon after, itemChangedAtURL: is called on the file provider

File provider can e.g., initiate an upload



Demo

Building a simple Document Provider

Summary

Summary

Xcode template to get you started with a Document Provider

Summary

Xcode template to get you started with a Document Provider

Document Provider shows up next to iCloud picker

Summary

Xcode template to get you started with a Document Provider

Document Provider shows up next to iCloud picker

Sample code is available to display each of the picker modes

From Documents in the Cloud to iCloud Drive

For users

From Documents in the Cloud to iCloud Drive

For users

On iOS 8, users can choose to continue using iCloud Documents or migrate to iCloud Drive

From Documents in the Cloud to iCloud Drive

For users

On iOS 8, users can choose to continue using iCloud Documents or migrate to iCloud Drive

On OS X Yosemite, users can choose to migrate to iCloud Drive or changes will no longer update across devices

From Documents in the Cloud to iCloud Drive

For users

On iOS 8, users can choose to continue using iCloud Documents or migrate to iCloud Drive

On OS X Yosemite, users can choose to migrate to iCloud Drive or changes will no longer update across devices

After an account is migrated to iCloud Drive, only devices using iCloud Drive will propagate changes to each other

From Documents in the Cloud to iCloud Drive

For developers

From Documents in the Cloud to iCloud Drive

For developers

The OS X Yosemite Developer Preview and the iOS 8 Beta use iCloud Drive

From Documents in the Cloud to iCloud Drive

For developers

The OS X Yosemite Developer Preview and the iOS 8 Beta use iCloud Drive

New features are only available in iCloud Drive

Summary

Summary

File coordination for document access

Summary

File coordination for document access

Discovery and display of files

Summary

File coordination for document access

Discovery and display of files

Using UIDocumentPicker to access files outside your container

Summary

File coordination for document access

Discovery and display of files

Using UIDocumentPicker to access files outside your container

Document Provider extensions to support third-party providers

More Information

Dave DeLong
App Frameworks Evangelist
delong@apple.com

Jake Behrens
App Frameworks Evangelist
behrens@apple.com

More Information

Documentation
iCloud for Developers
http://developer.apple.com/icloud

Document Picker Programming Guide
http://developer.apple.com

Apple Developer Forums
http://devforums.apple.com

Related Sessions

Introducing CloudKit	Mission	Tuesday 3:15PM
Creating Extensions for iOS and OS X, Part 1	Mission	Tuesday 2:00PM
Creating Extensions for iOS and OS X, Part 2	Mission	Wednesday 11:30AM

Labs

● Extensions Lab	Frameworks Lab B Thursday 2:00PM

WWDC14