

File Provider Enhancements

Enumerating and modifying files in the cloud

Session 243

Pierre d'Herbement, Software Engineer
Johannes Fortmann, Software Engineer
Jean-Gabriel Morard, Software Engineer

What is a file provider?

Enumerating files

Modifying files

Customizing your file provider

What Is a File Provider?

At minimum, file providers fulfill file promises

NEW

But we're here to talk about
cloud-based, enumerating file providers



NEW

iPad 9:41 AM 100%

Browse Edit

Locations

- iCloud Drive
- Dropbox
- Box
- On My iPad
- Recently Deleted

Favorites

Tags

- Red
- Orange
- Yellow
- Green
- Blue
- Purple
- Gray
- Home

iCloud Drive Select

Search

Hiking Trip

History of Guitars

Beach

Half Dome

Dad and Jimmy

Pacific Crest Trail

Smithsonian

Science Fair

JPEG image

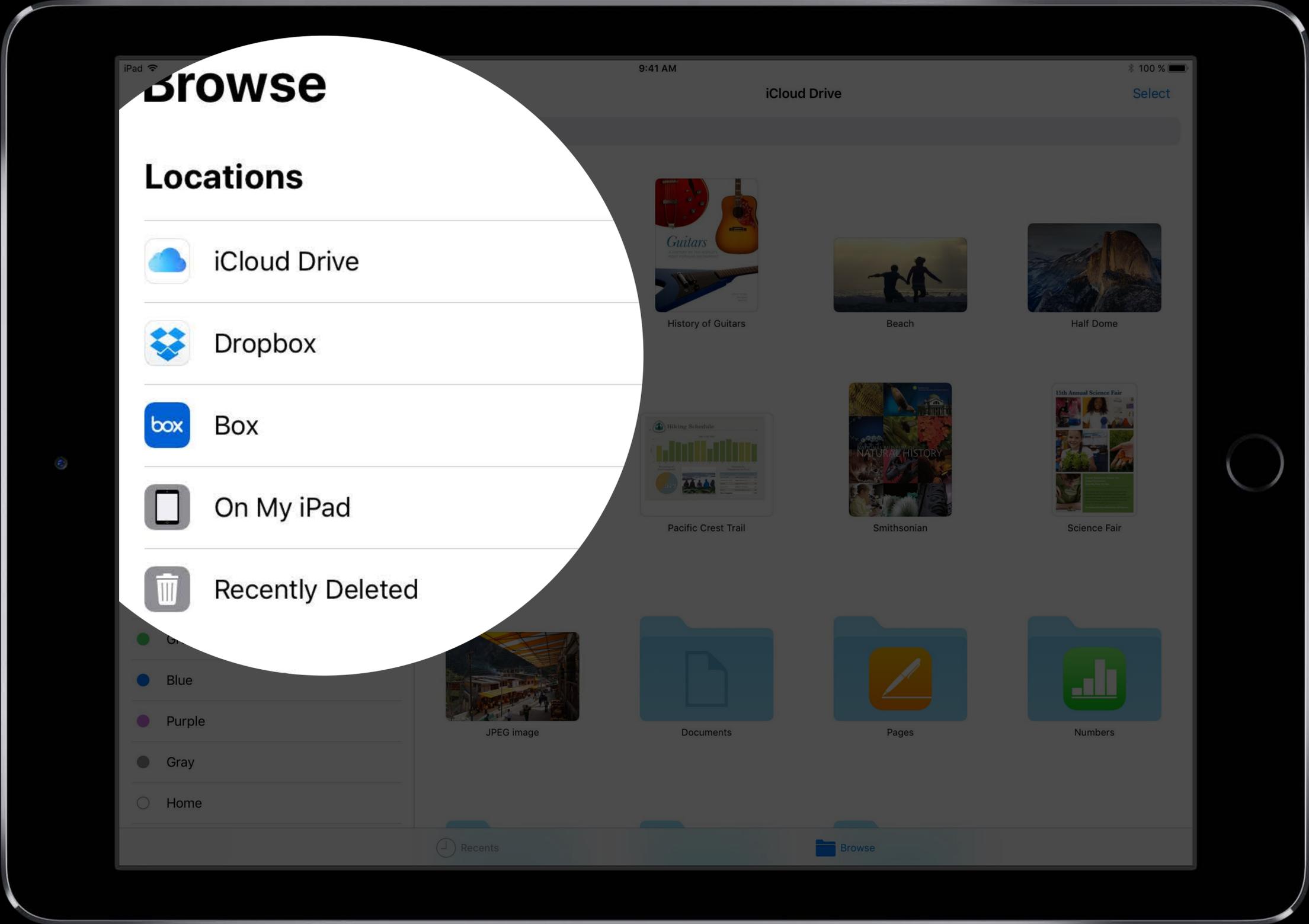
Documents

Pages

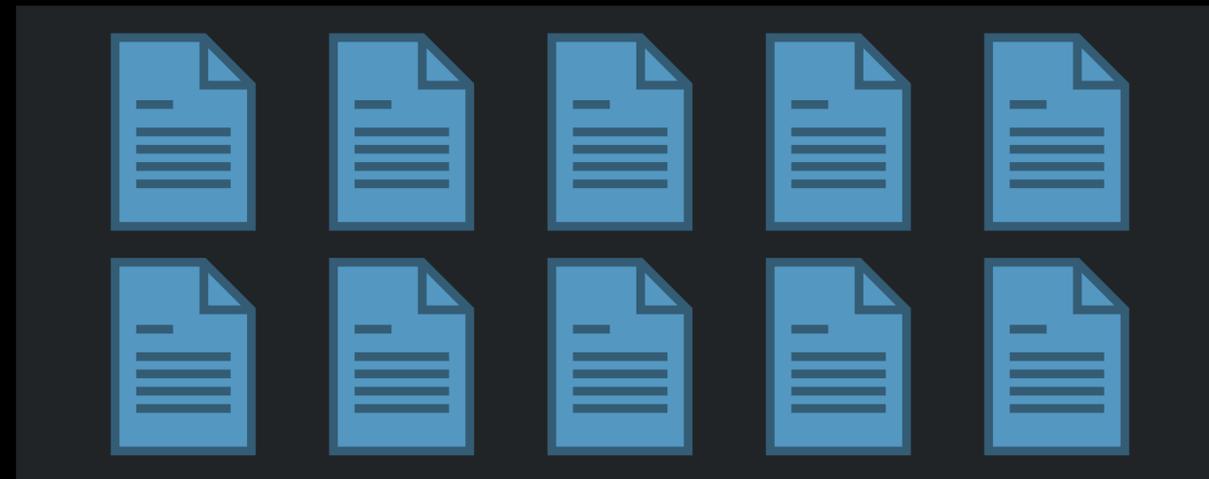
Numbers

Recents Browse

NEW

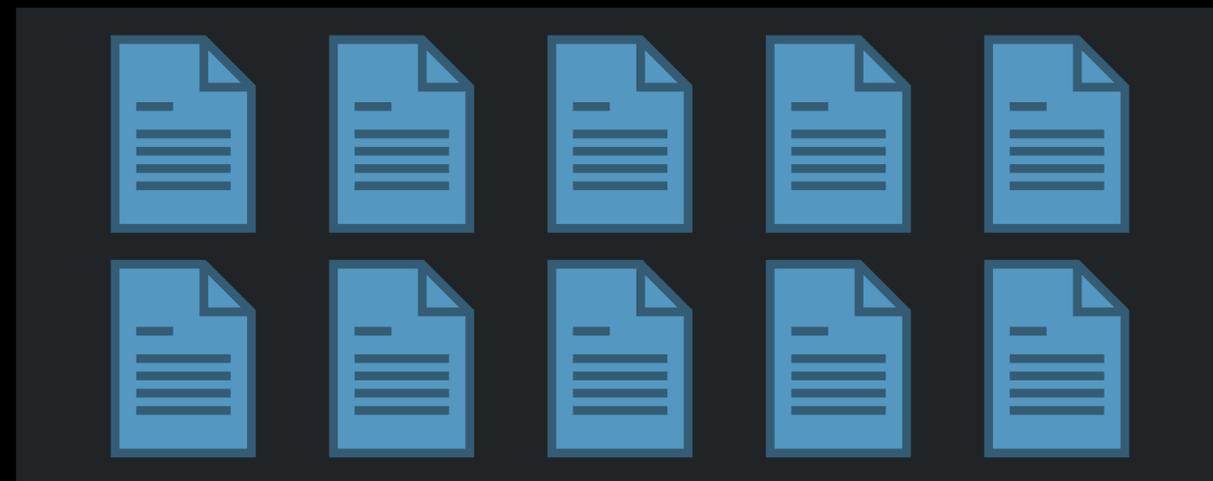


Document Browser



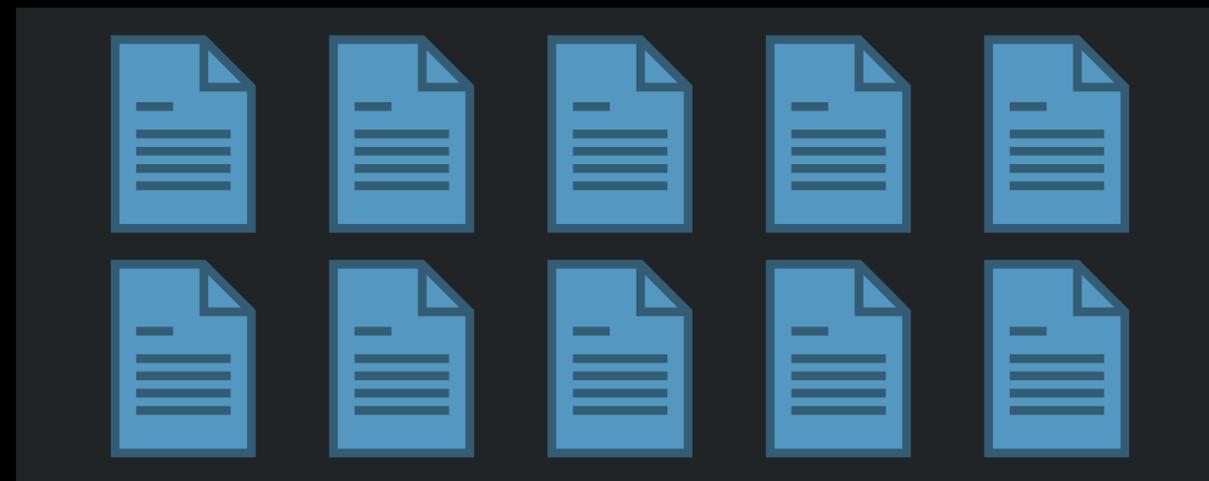
Document Browser

Document-Based
applications



Document Browser

Document-Based applications

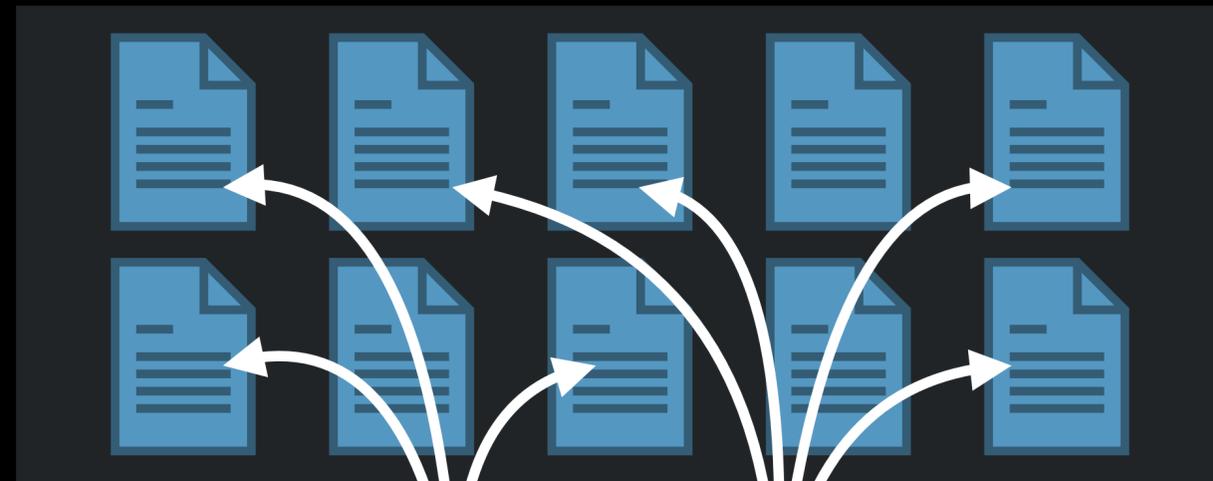


File Providers

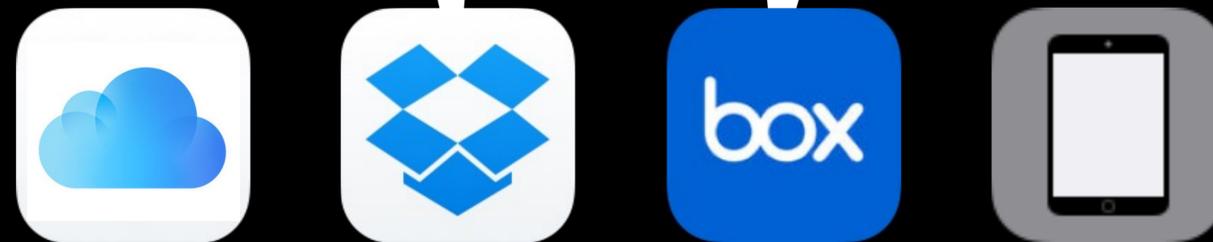


Document Browser

Document-Based applications



File Providers

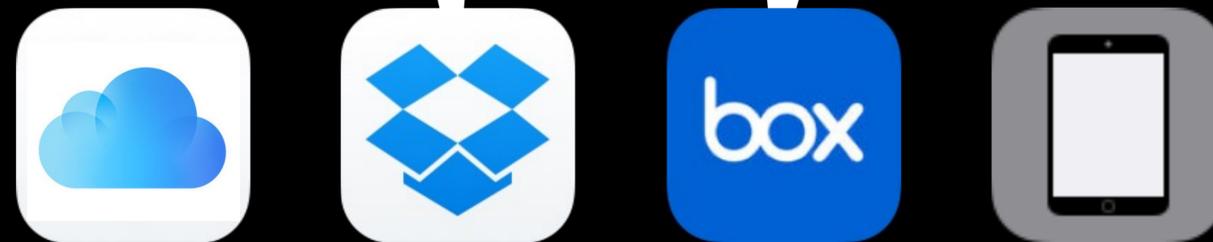


Document Browser

Document-Based applications



File Providers

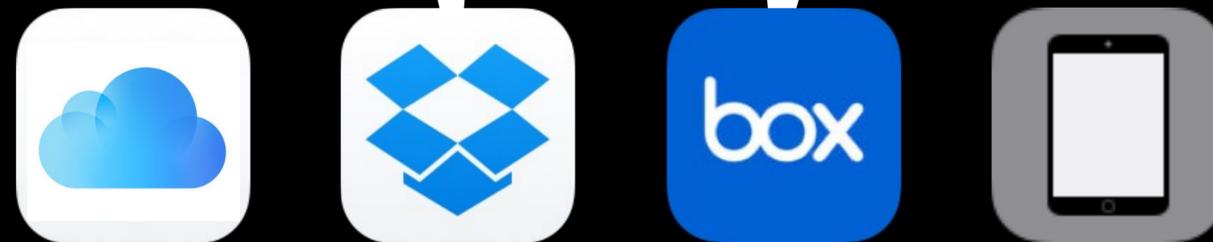


Document Browser

Document-Based applications



File Providers



Document Browser

Document-Based applications

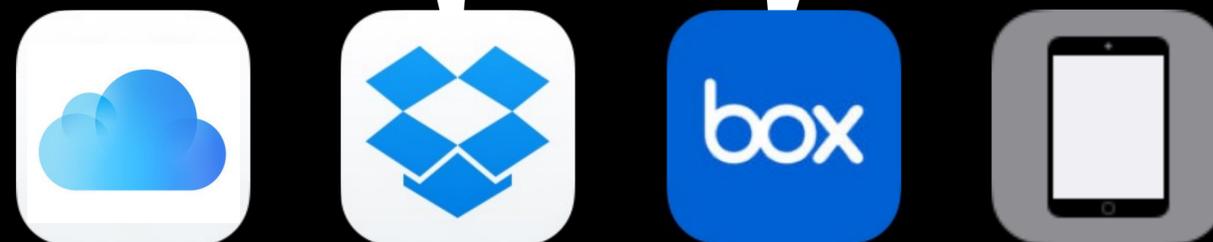


UIDocumentBrowserViewController



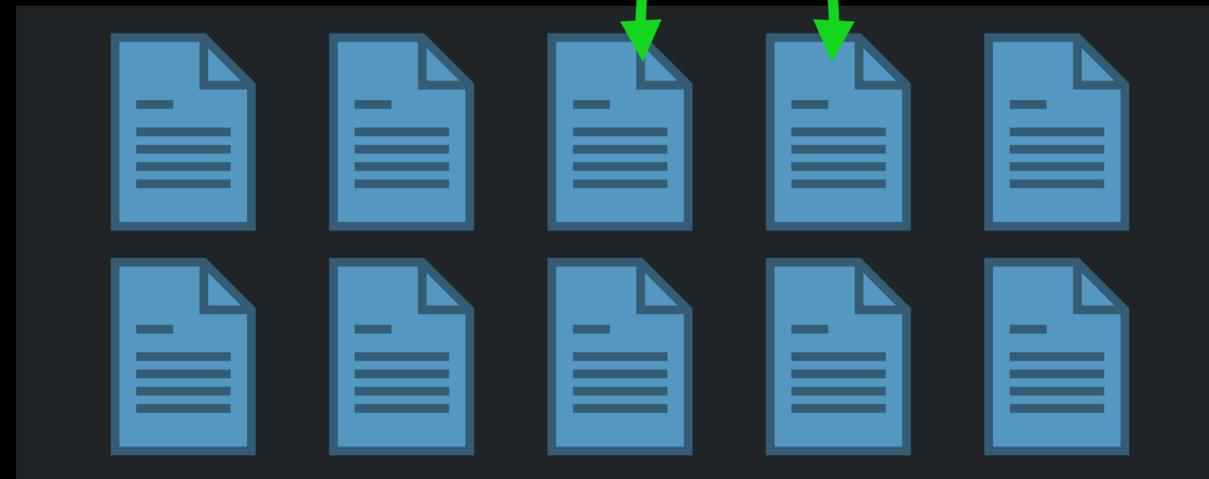
NSFileProviderExtension

File Providers



Document Browser

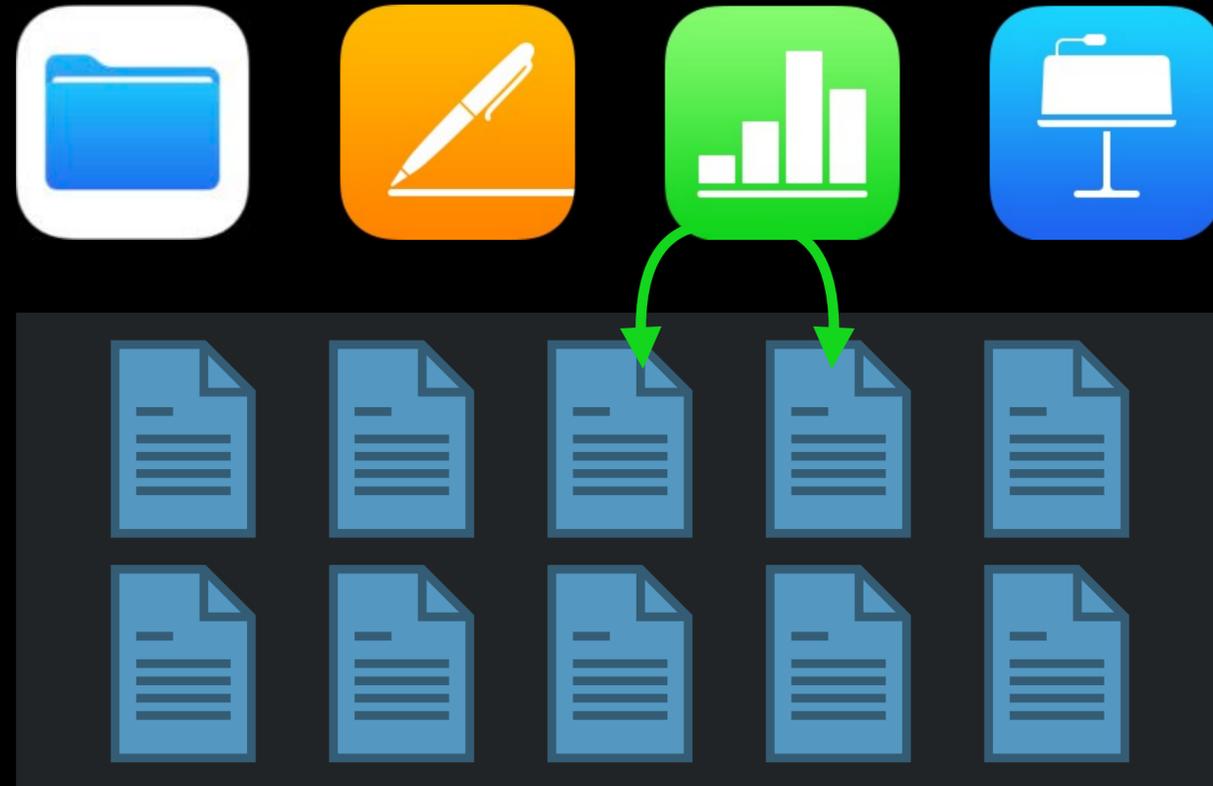
Document-Based
applications



UIDocumentBrowserViewController

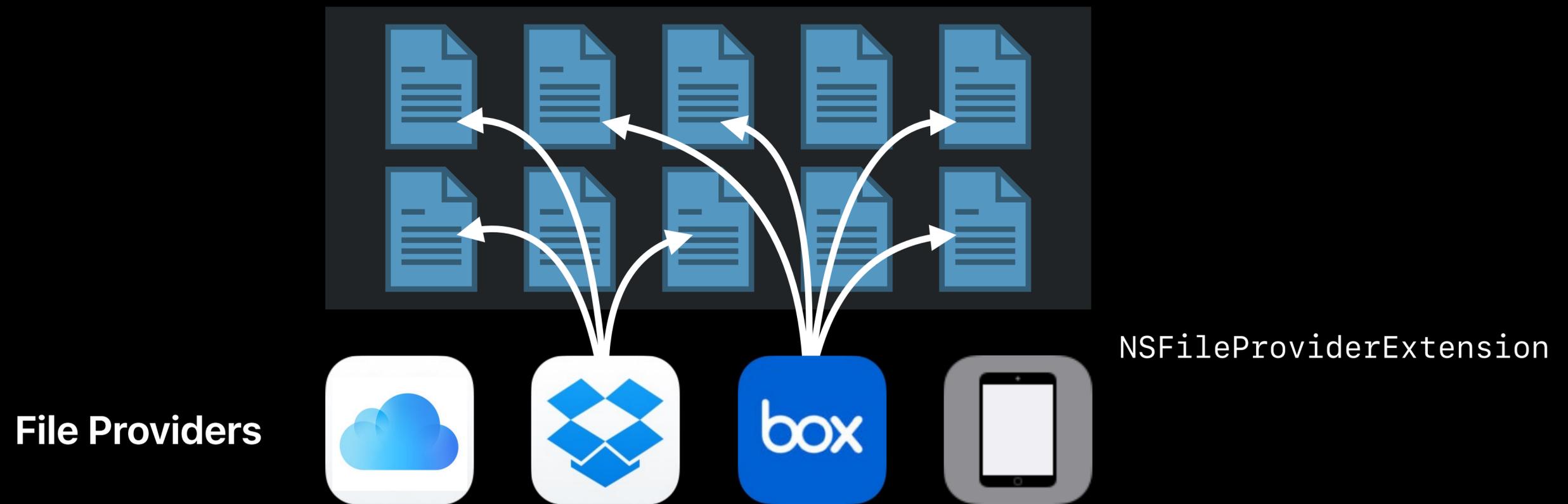
Document Browser

Document-Based applications



UIDocumentBrowserViewController

Document Browser



NEW

iPad 9:41 AM 100%

Browse Edit

Locations

- iCloud Drive
- Dropbox
- Box
- On My iPad
- Recently Deleted

Favorites

Tags

- Red
- Orange
- Yellow
- Green
- Blue
- Purple
- Gray
- Home

iCloud Drive Select

Search

Hiking Trip

History of Guitars

Beach

Half Dome

Dad and Jimmy

Pacific Crest Trail

Smithsonian

Science Fair

JPEG image

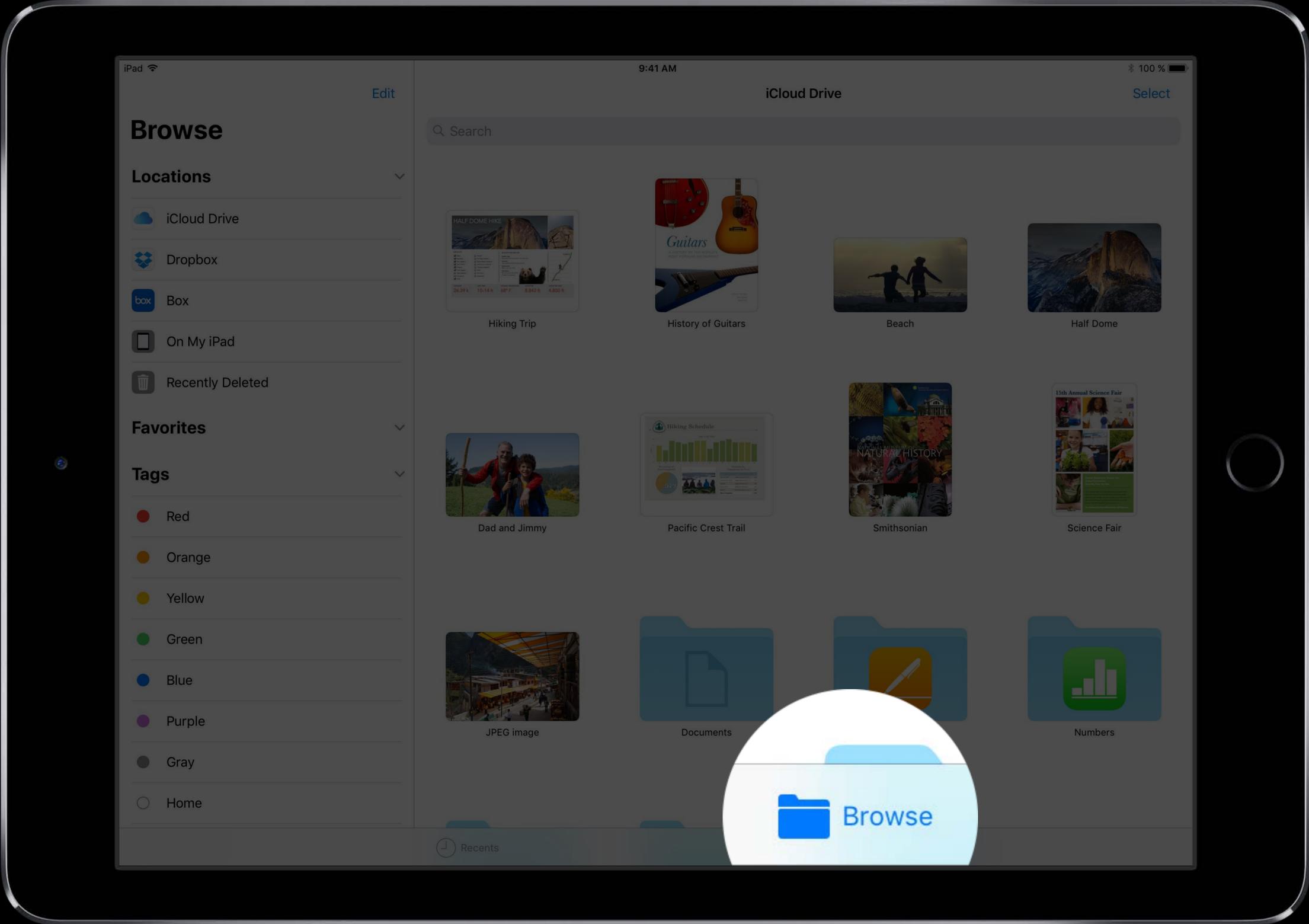
Documents

Pages

Numbers

Recents Browse

NEW



iPad

9:41 AM

100 %

Edit

iCloud Drive

Select

Browse

Locations

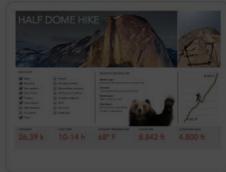
- iCloud Drive
- Dropbox
- Box
- On My iPad
- Recently Deleted

Favorites

Tags

- Red
- Orange
- Yellow
- Green
- Blue
- Purple
- Gray
- Home

Search



Hiking Trip



History of Guitars



Beach



Half Dome



Dad and Jimmy



Pacific Crest Trail



Smithsonian



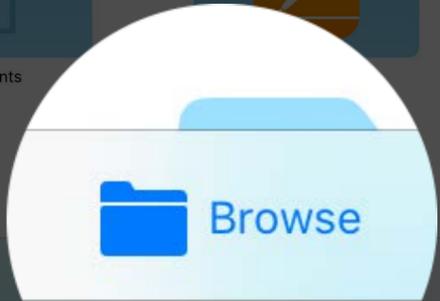
Science Fair



JPEG image

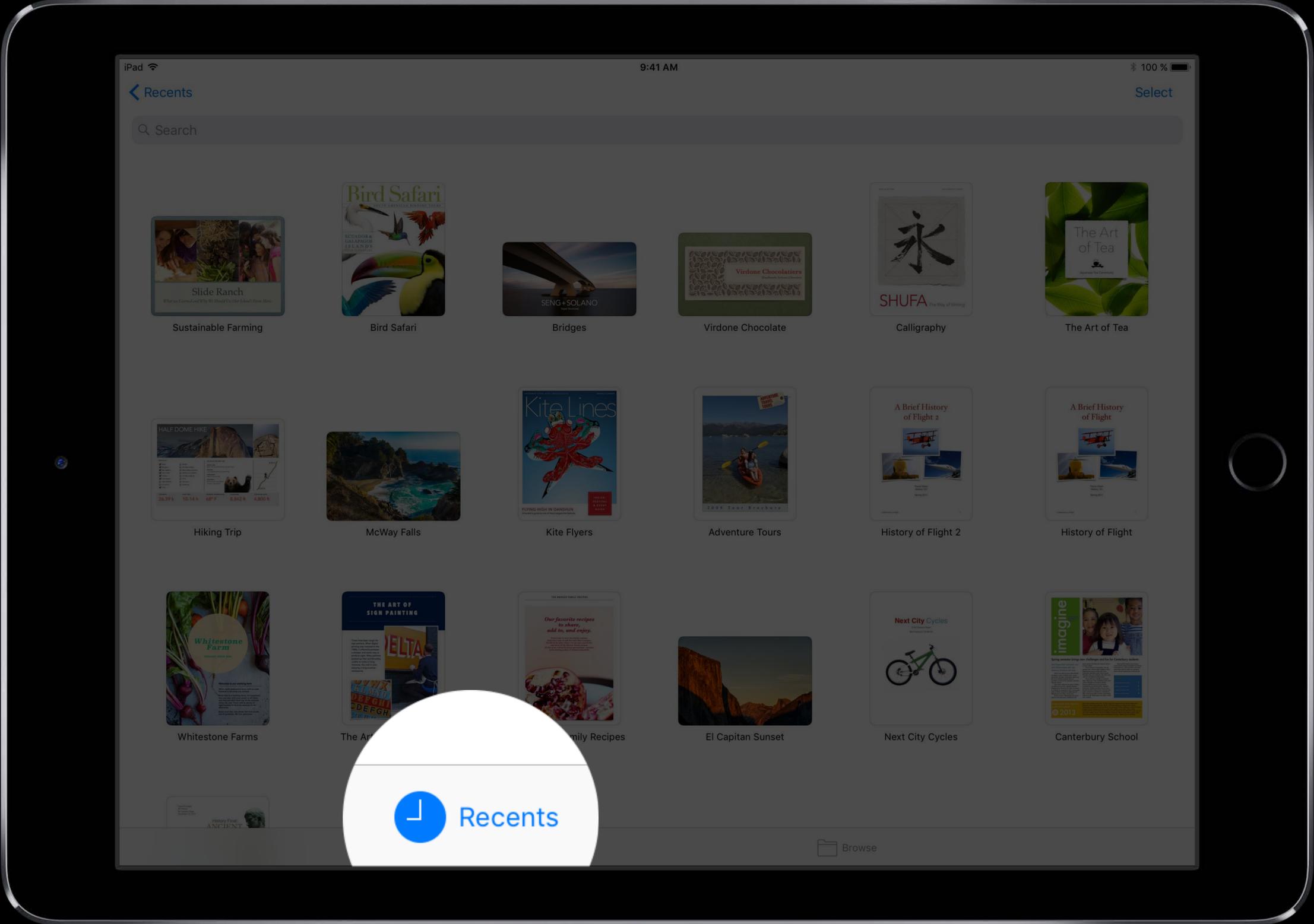


Documents



Recents

NEW



Recents

NEW

iPad 9:41 AM 100%

< Recents Select

Search

Slide Ranch

Sustainable Farming

Bird Safari

Bird Safari

SENG+SOLANO

Bridges

Virdone Chocolatiers

Virdone Chocolate

SHUFA

Calligraphy

The Art of Tea

The Art of Tea

HALF DOME HIKE

Hiking Trip

McWay Falls

Kite Lines

Kite Flyers

Adventure Tours

A Brief History of Flight 2

History of Flight 2

A Brief History of Flight

History of Flight

Whitstone Farms

Whitstone Farms

THE ART OF SIGN PAINTING

The Art of Sign Painting

Our favorite recipes to share, add to, and enjoy.

Lebanese Family Recipes

El Capitan Sunset

Next City Cycles

Next City Cycles

imagine

Canterbury School

Recents Browse

Browse Versus Recents

The "Browse" tab is for pulling new content from each file provider

The "Recents" tab shows the working set

- The working set is not folder-based
- It spans multiple file providers

Working Set

Recents

Favorites

Tags

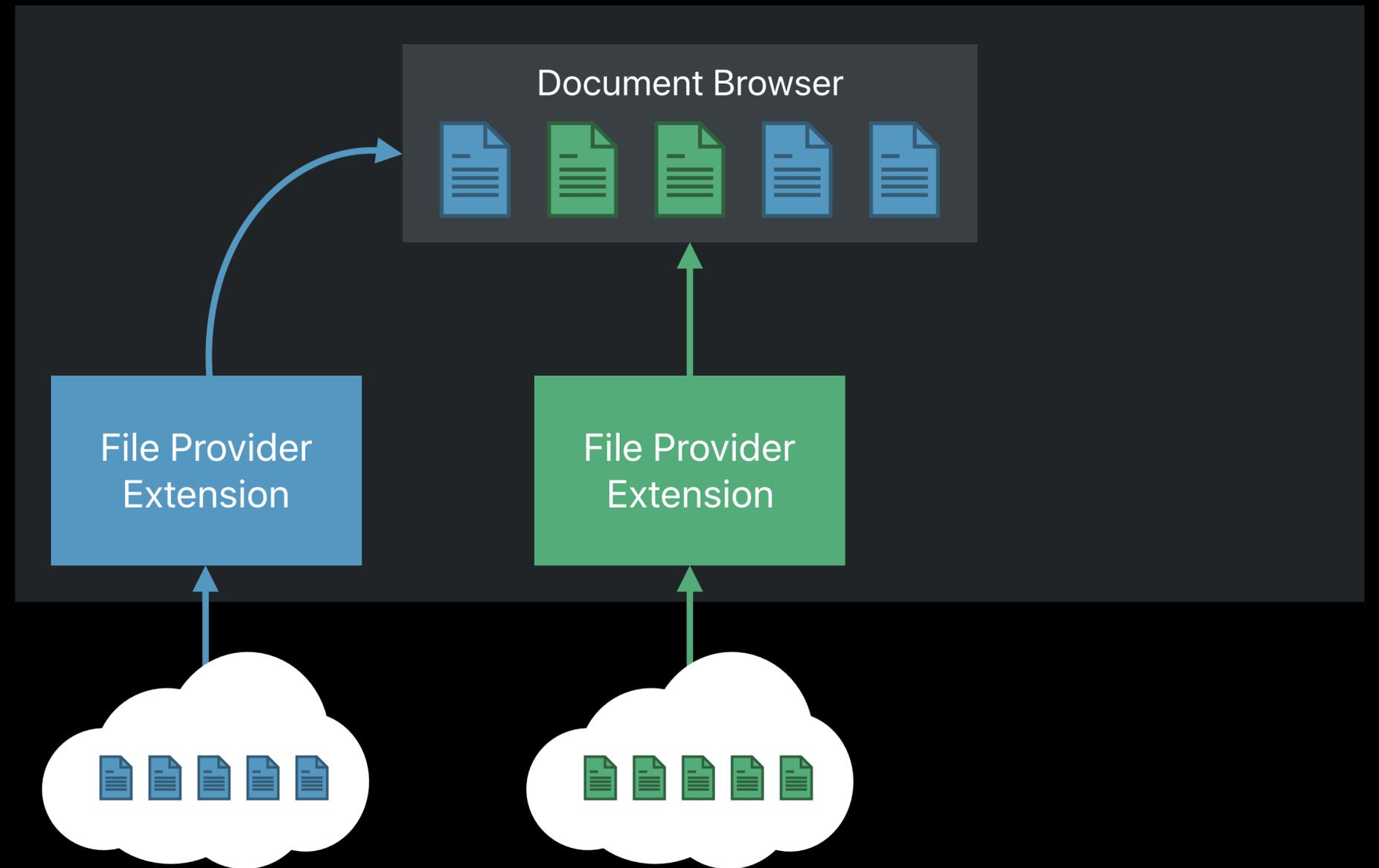
Shares

Local

Recent deletes

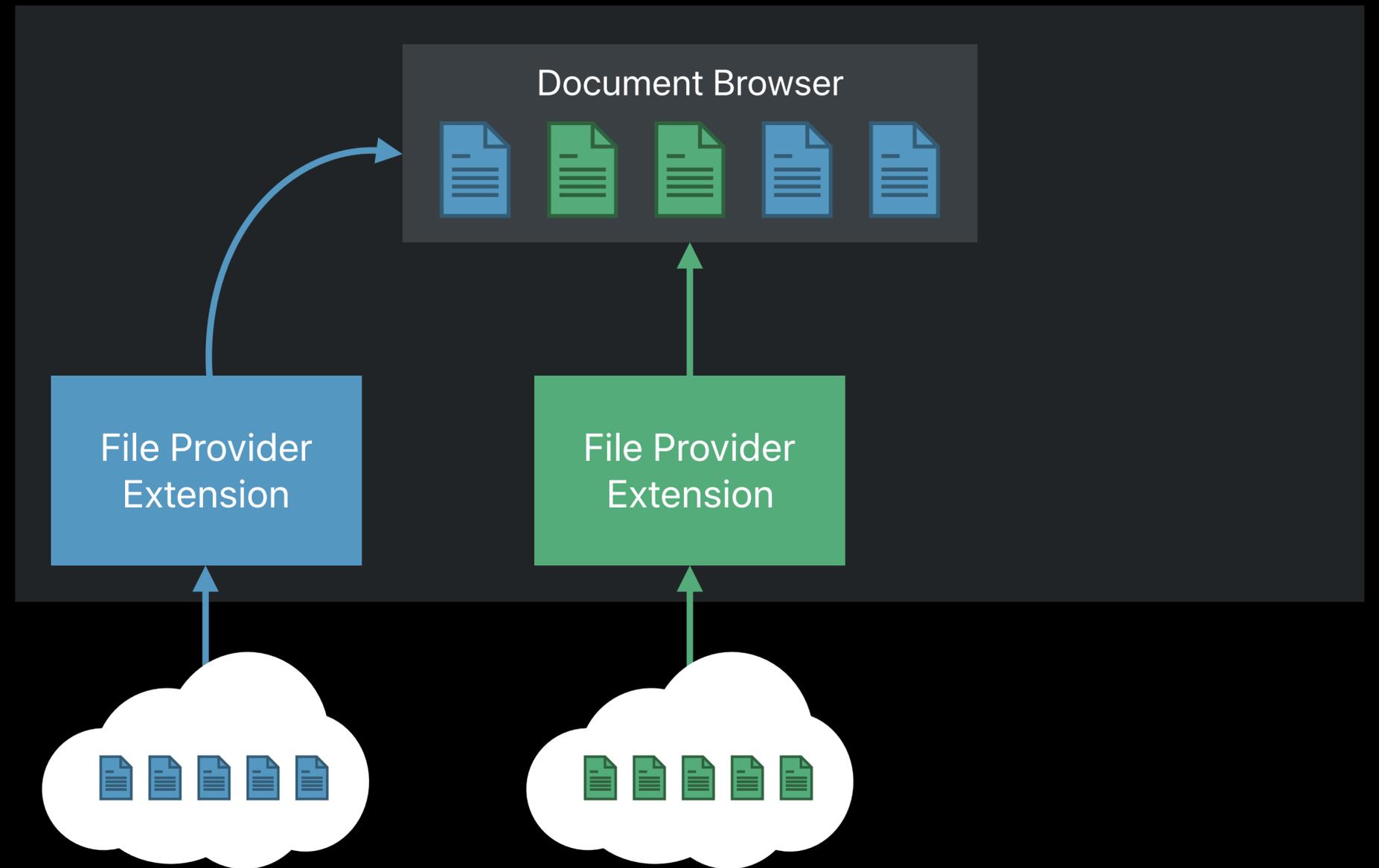
Working Set

Each file provider contributes documents



Working Set

Each file provider
contributes documents
But not at display time

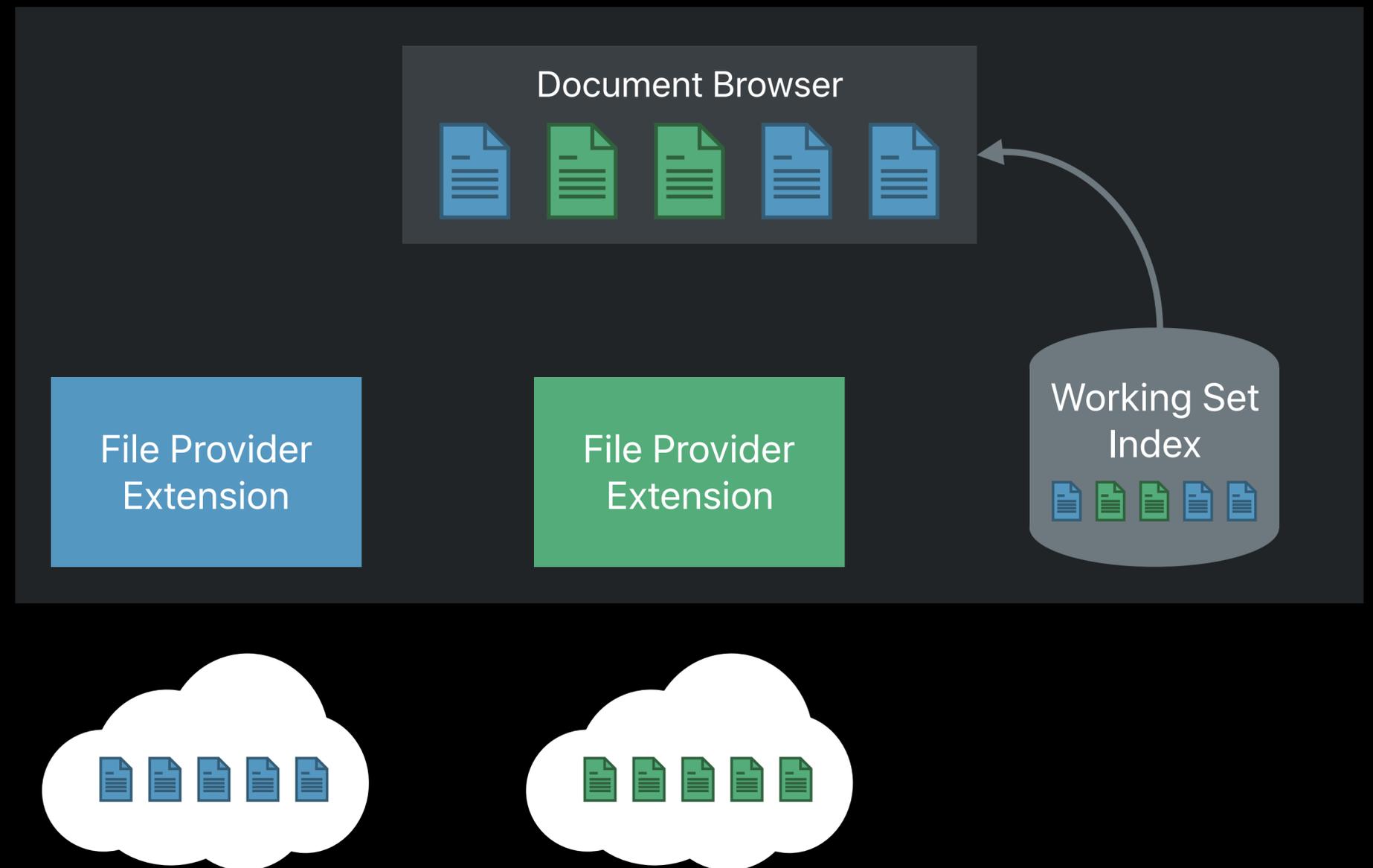


Working Set

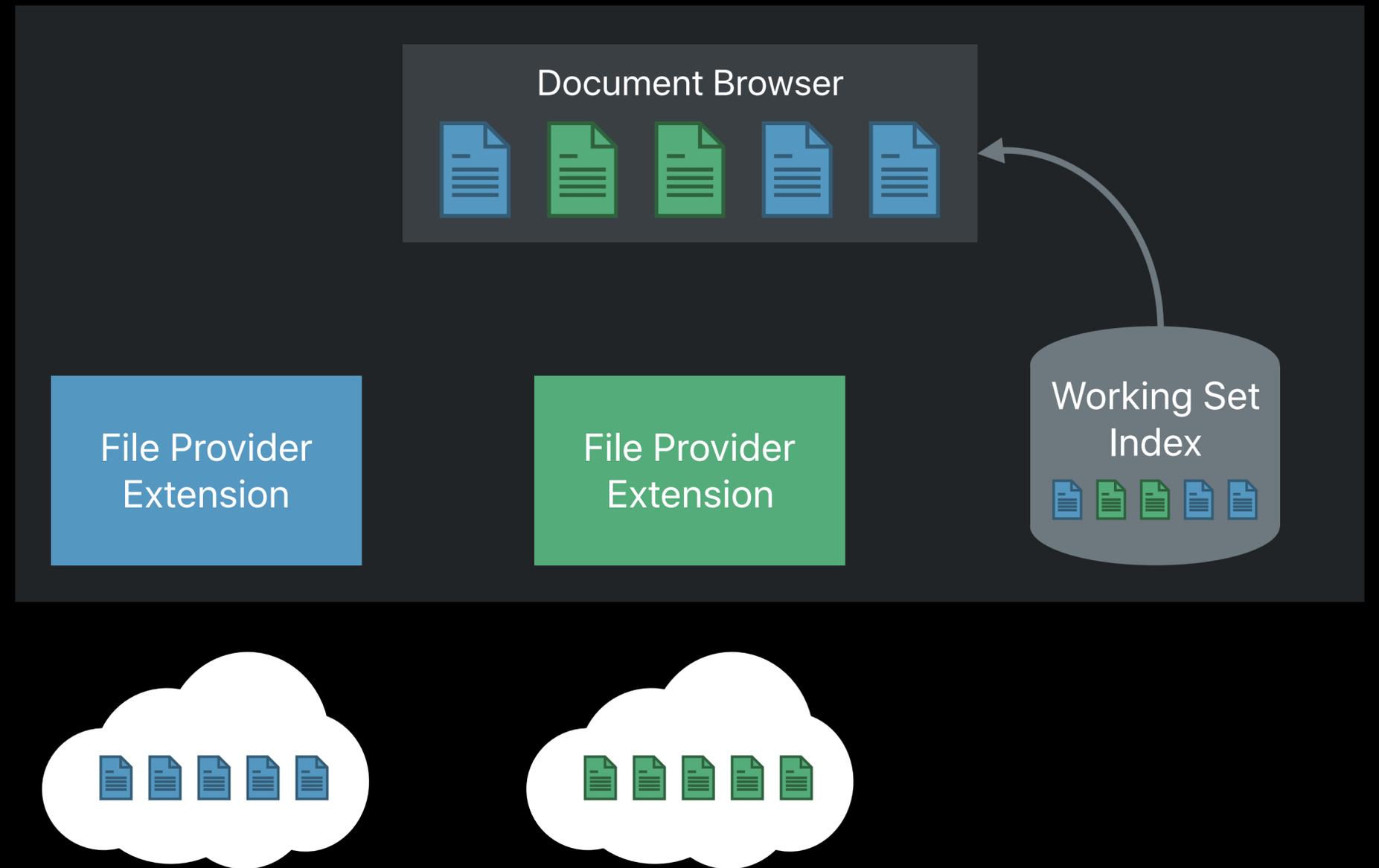
Each file provider
contributes documents

But not at display time

The document browser
enumerates a pre-merged
index of the working sets



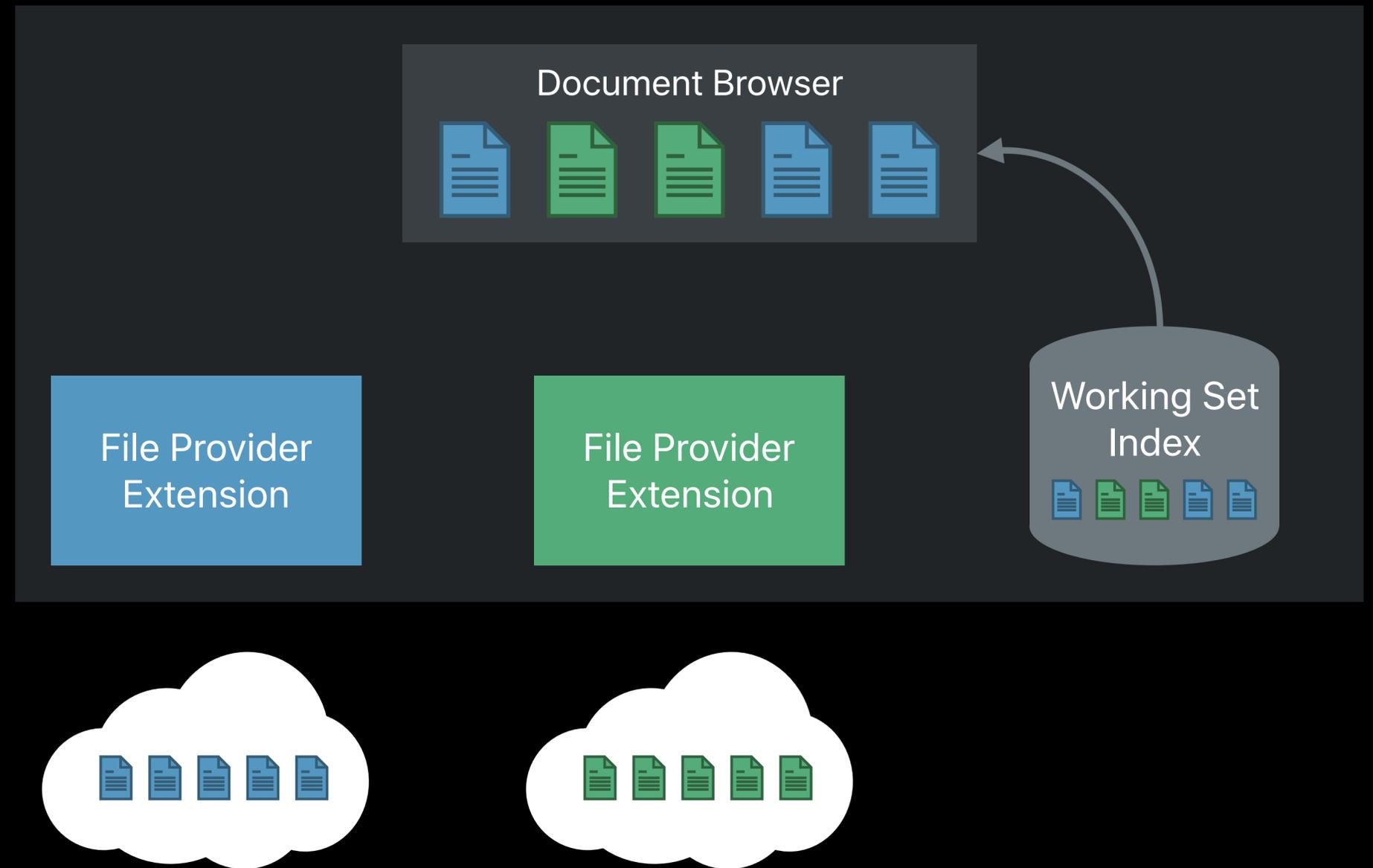
Working Set



Working Set

The index is local

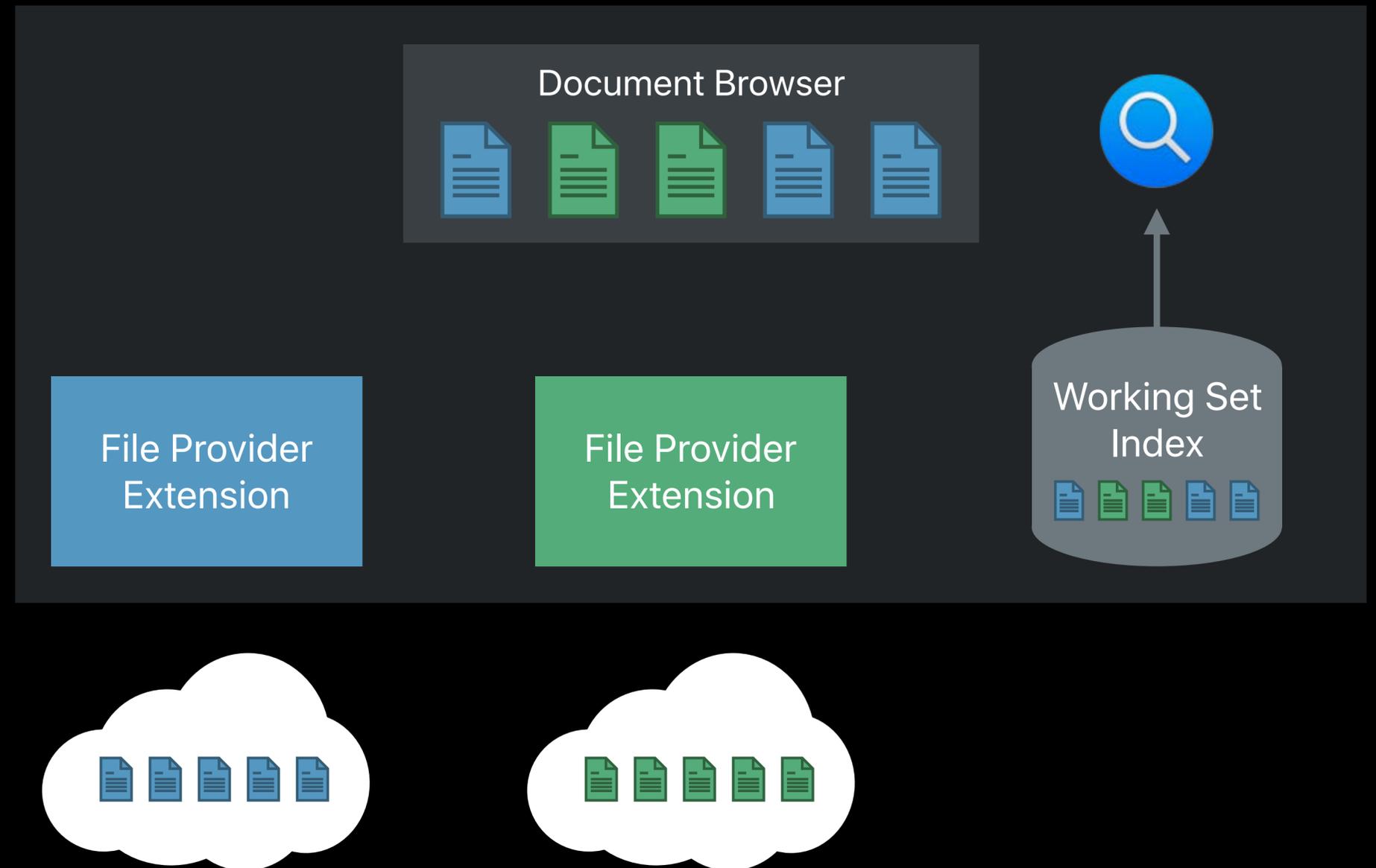
Working set documents are available offline



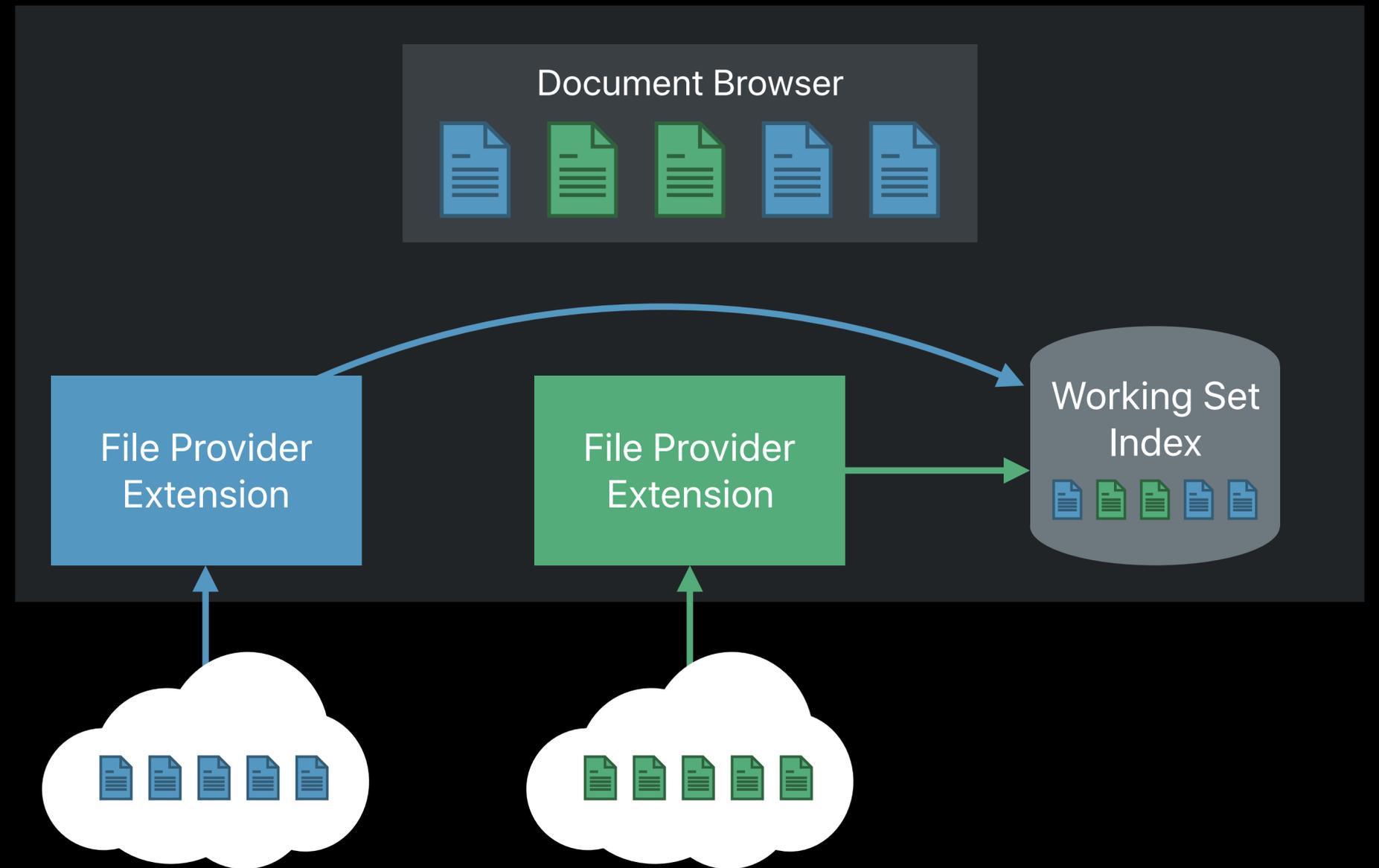
Working Set

The index is local

Working set documents are available offline

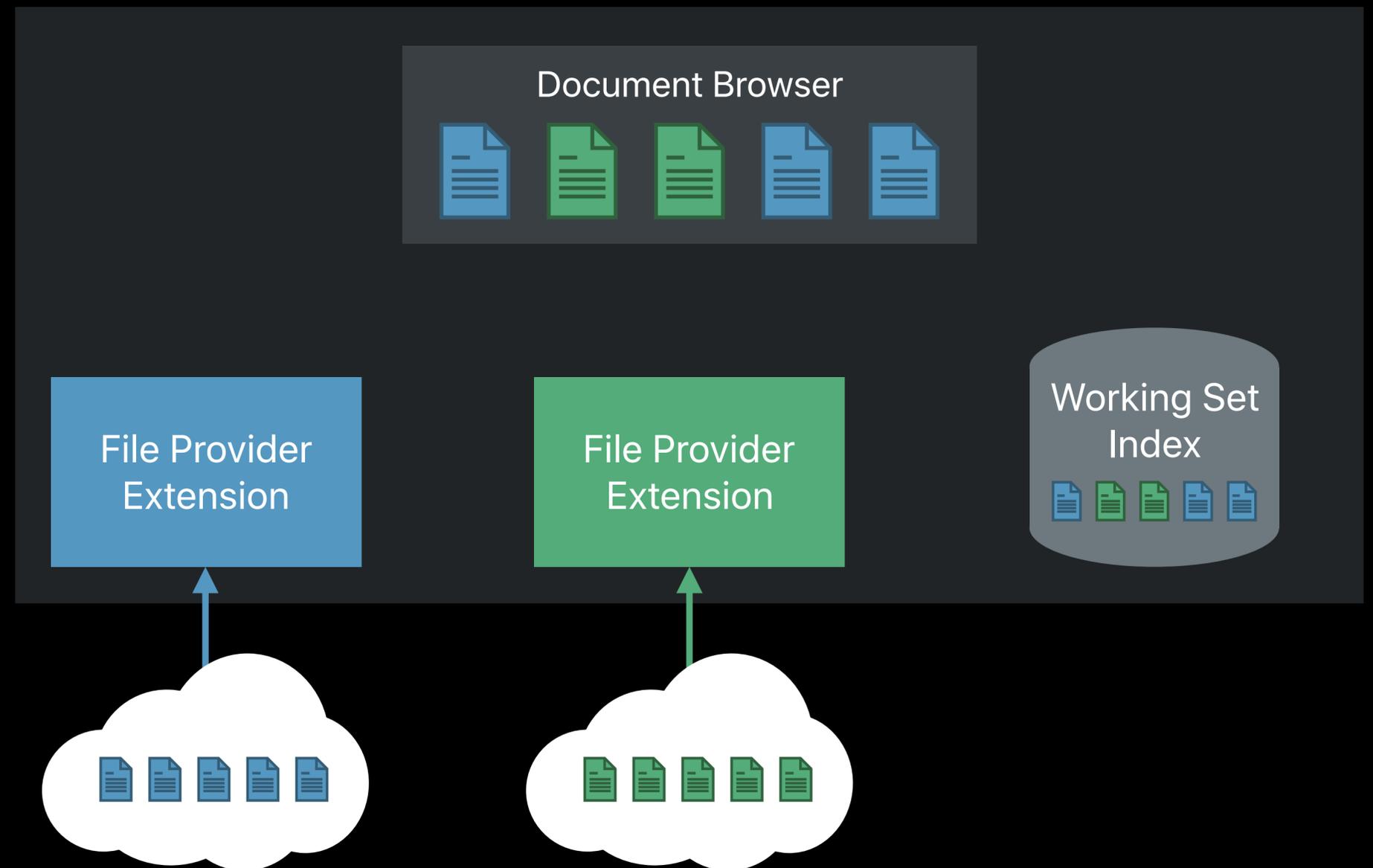


Working Set



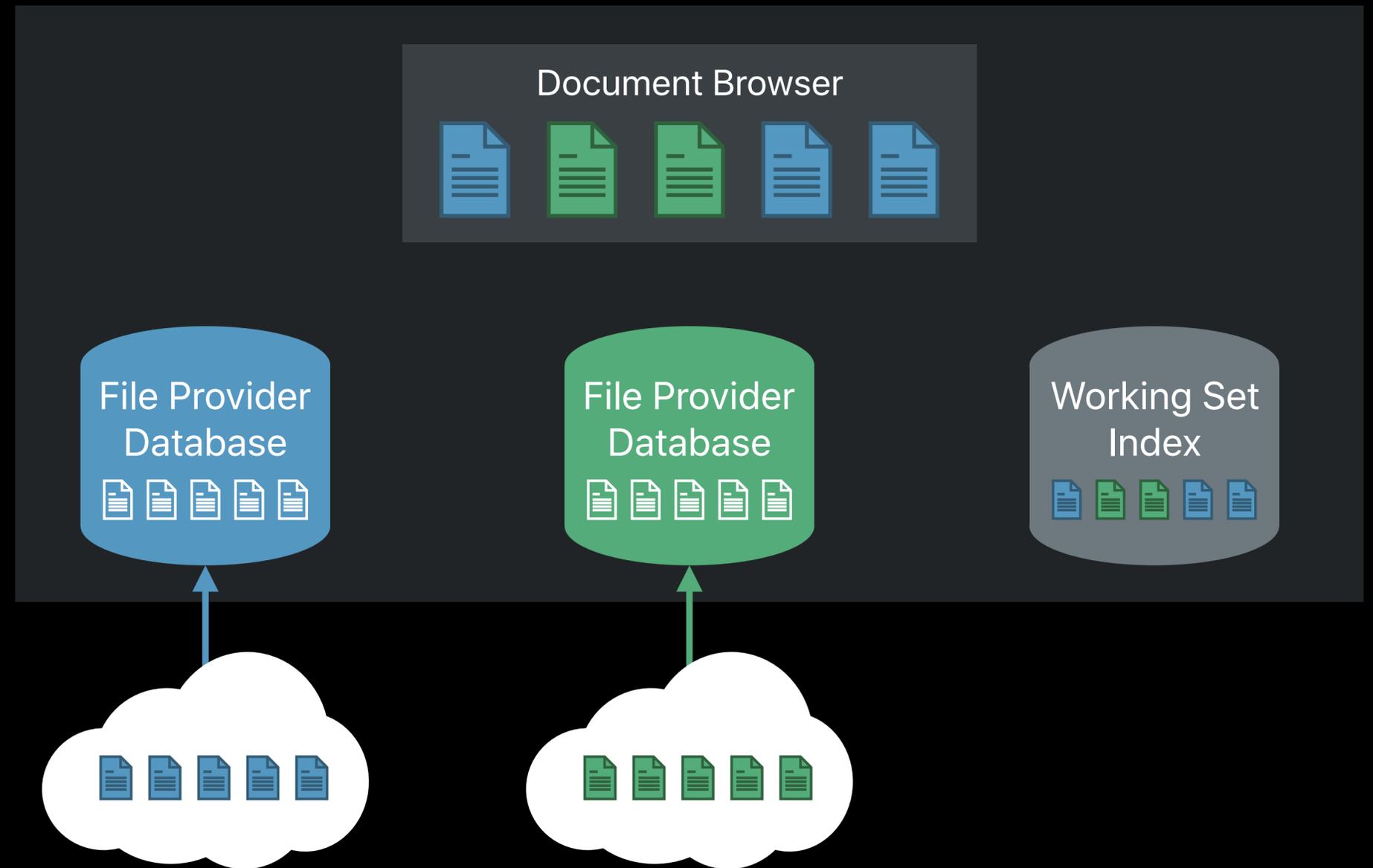
Working Set

Each file provider syncs its working set



Working Set

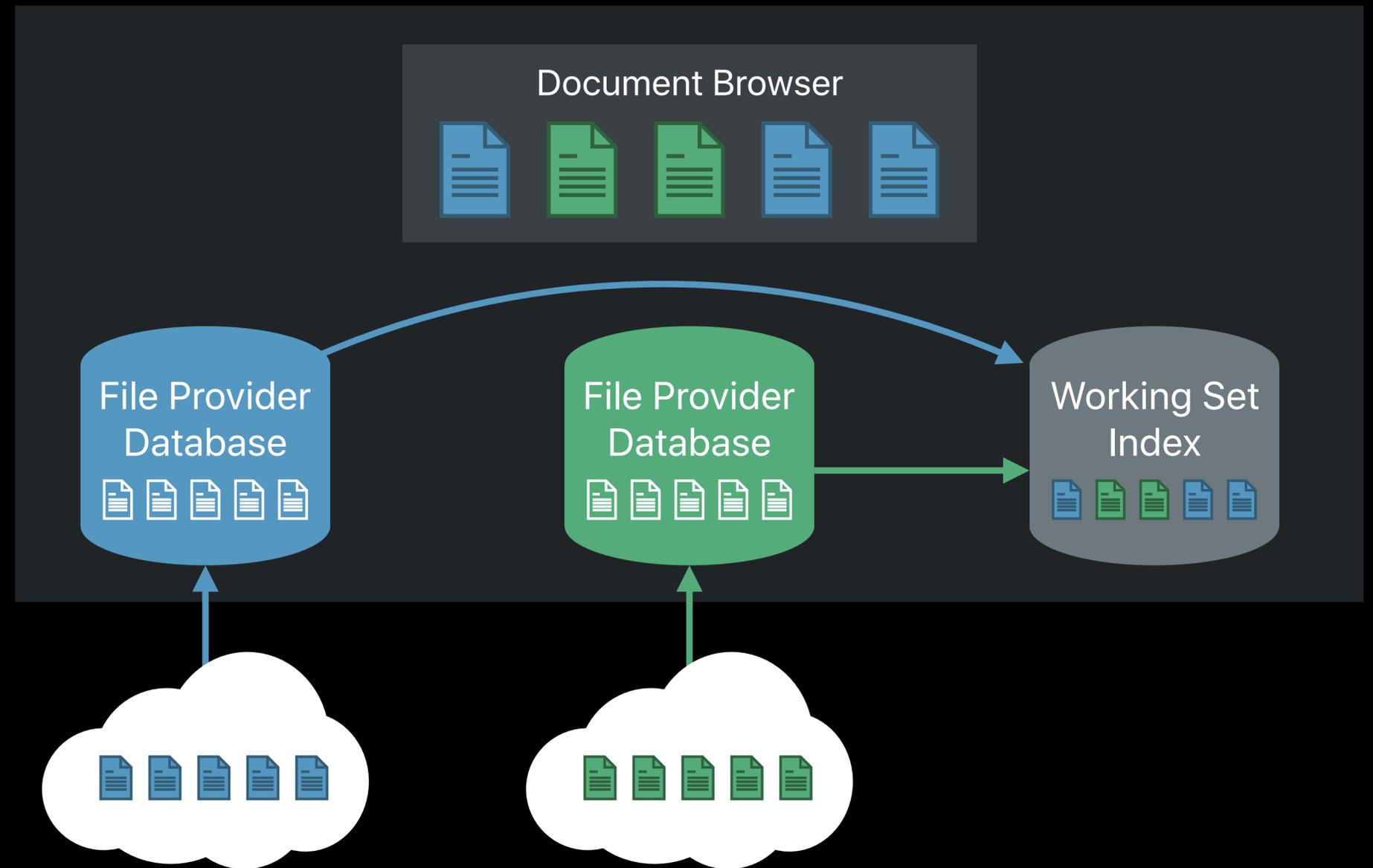
Each file provider syncs its working set



Working Set

Each file provider syncs its working set

The system indexes the working sets

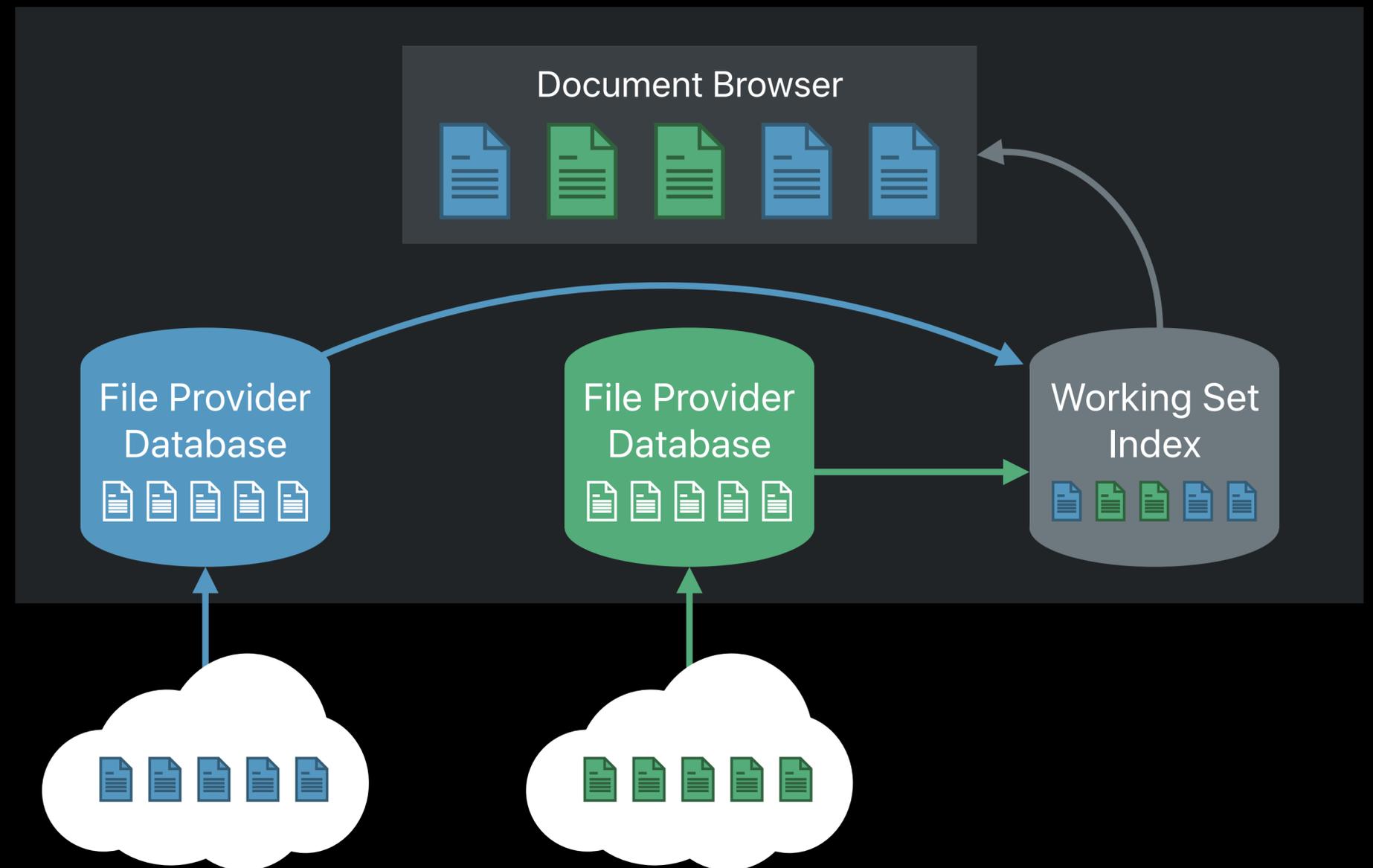


Working Set

Each file provider syncs its working set

The system indexes the working sets

Document browsers enumerate the index



Creating a File Provider Extension

Pierre d'Herbement, Software Engineer

Creating a File Provider Extension

Creating the project

Providing an item

Enumerating items

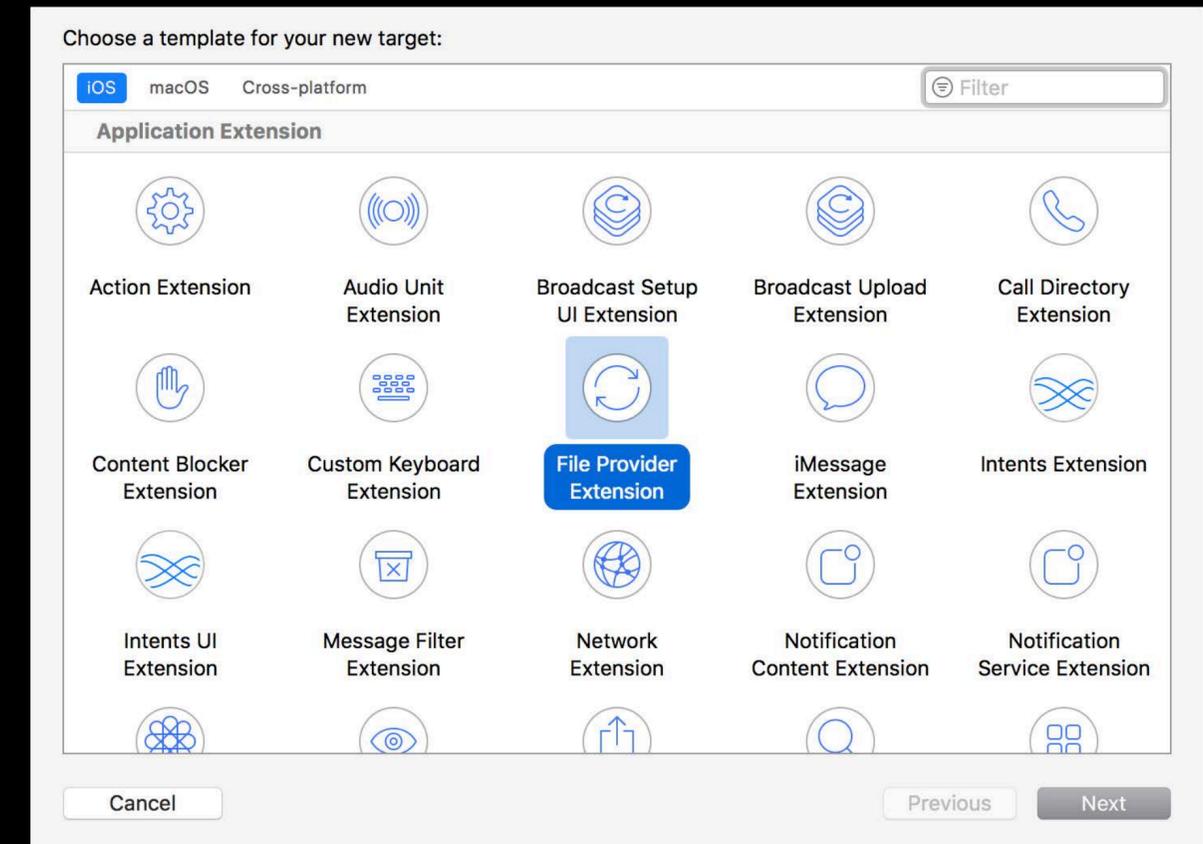
Modifying items

Custom actions

Expose items services to third parties

Creating a File Provider Extension

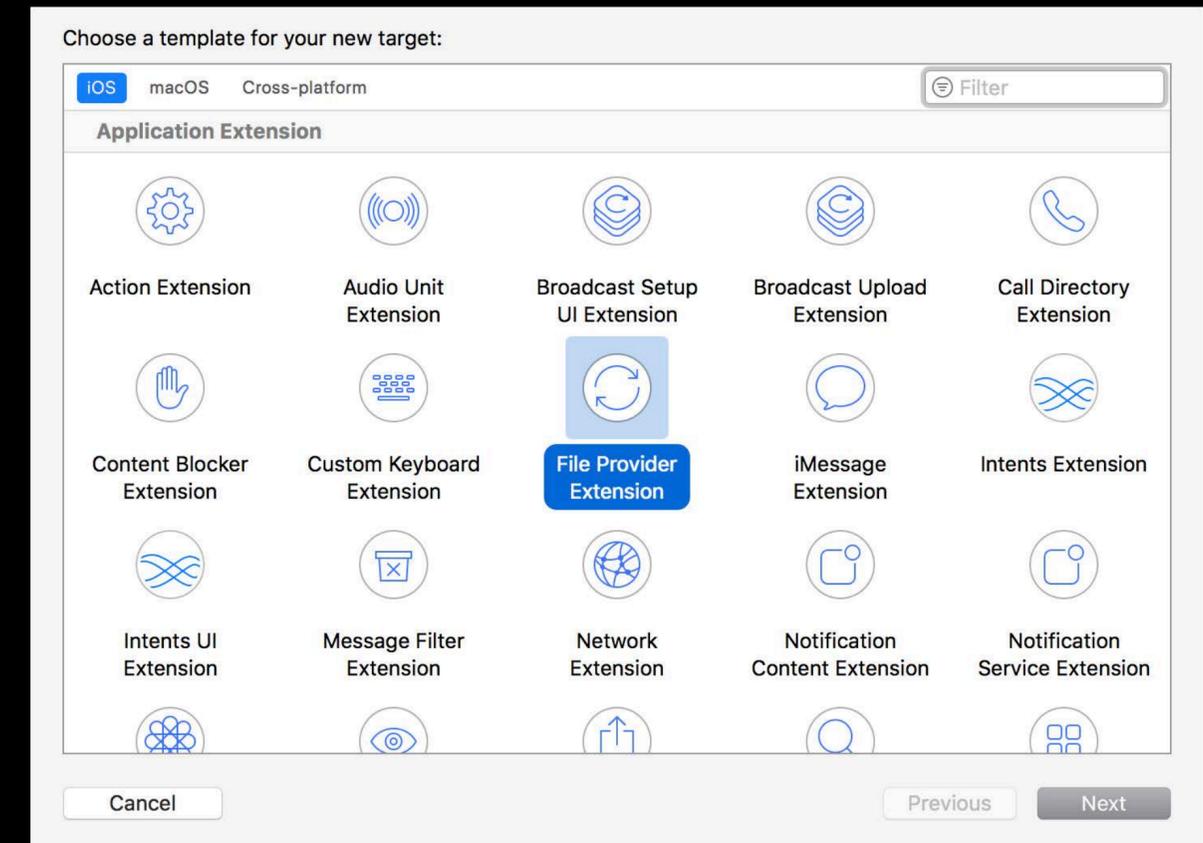
Use the Xcode template



Creating a File Provider Extension

Use the Xcode template

- Subclass `NSFileProviderExtension`



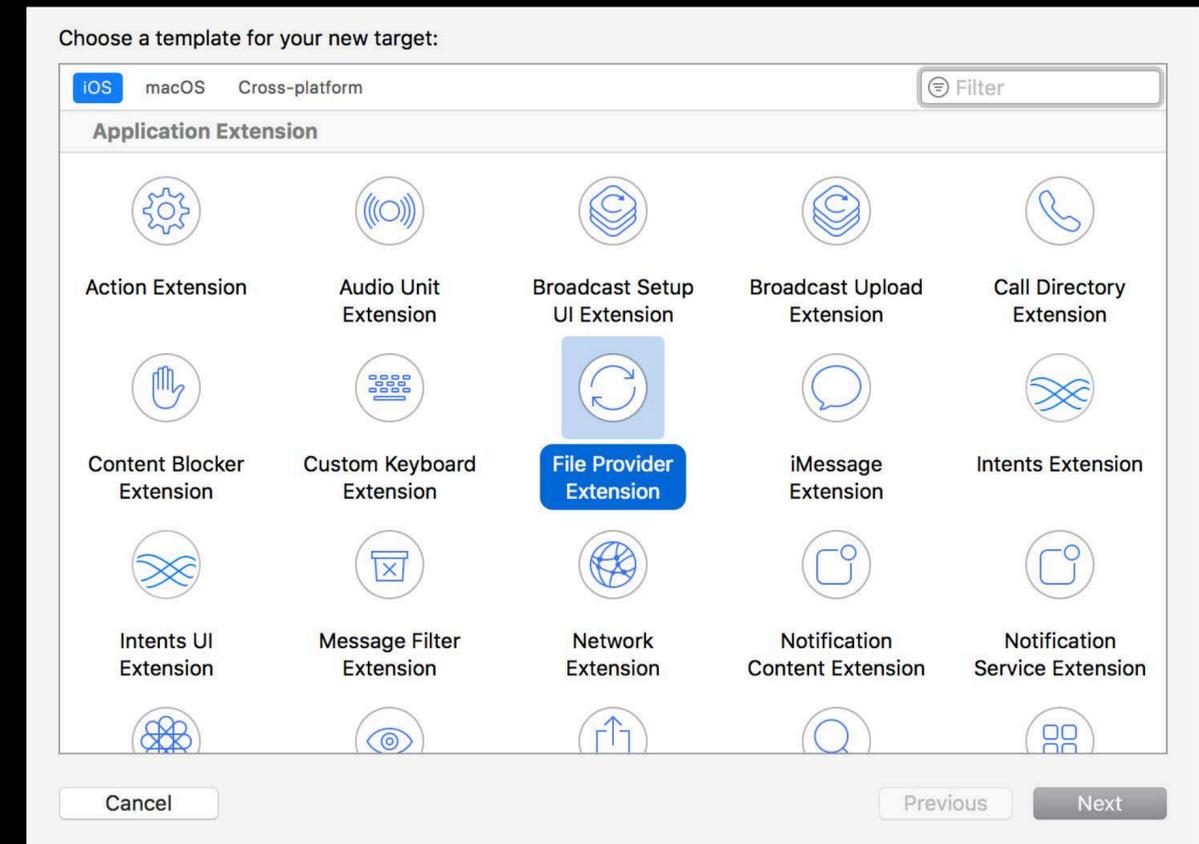
Creating a File Provider Extension

Use the Xcode template

- Subclass `NSFileProviderExtension`

Existing file provider

- Add `NSExtensionFileProviderSupportsEnumeration` to your `Info.plist`



NSFileProviderExtension Subclass

NSFileProviderExtension Subclass

Vend item metadata by implementing `NSFileProviderItem`

NSFileProviderExtension Subclass

Vend item metadata by implementing `NSFileProviderItem`

Identified by `NSFileProviderItemIdentifier` (String)

NSFileProviderExtension Subclass

Vend item metadata by implementing `NSFileProviderItem`

Identified by `NSFileProviderItemIdentifier` (String)

Materialized on disk through file URL

NSFileProviderExtension Subclass

Vend item metadata by implementing `NSFileProviderItem`

Identified by `NSFileProviderItemIdentifier` (String)

Materialized on disk through file URL

Expose folder contents by implementing `NSFileProviderEnumerator`

Anatomy of a File Provider Extension

Anatomy of a File Provider Extension

Item identifier → metadata

Anatomy of a File Provider Extension

Item identifier → metadata

Database	
ID	Metadata
1	...
2	...
3	...
4	...
5	...

Anatomy of a File Provider Extension

Item identifier → metadata

Item identifier ↔ file URL

Database	
ID	Metadata
1	...
2	...
3	...
4	...
5	...

Anatomy of a File Provider Extension

Item identifier → metadata

Item identifier ↔ file URL

Database		
ID	Metadata	URL
1
2
3
4
5

Anatomy of a File Provider Extension

Item identifier → metadata

Item identifier ↔ file URL

Manage the disk storage

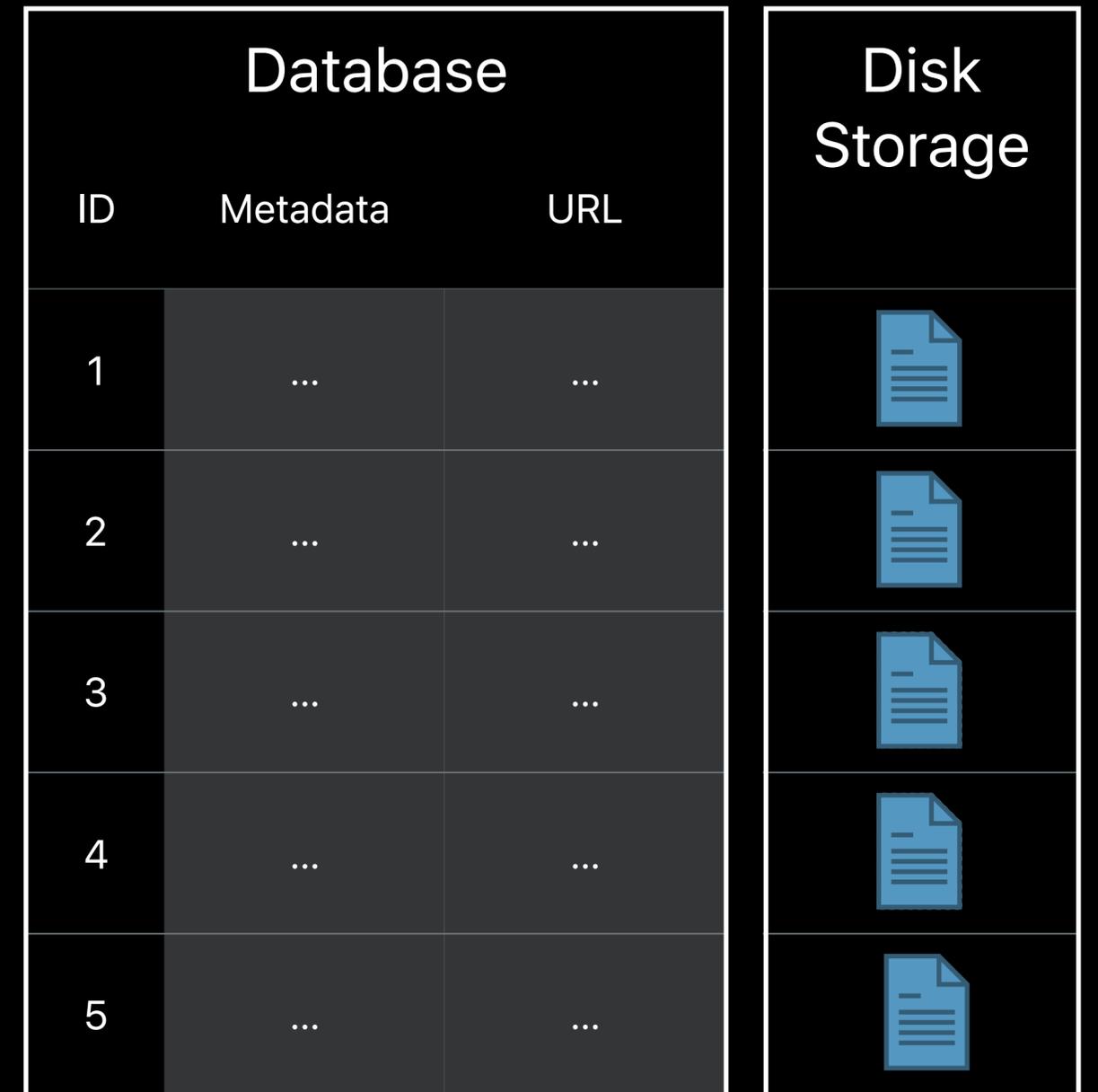
Database		
ID	Metadata	URL
1
2
3
4
5

Anatomy of a File Provider Extension

Item identifier → metadata

Item identifier ↔ file URL

You manage the disk storage

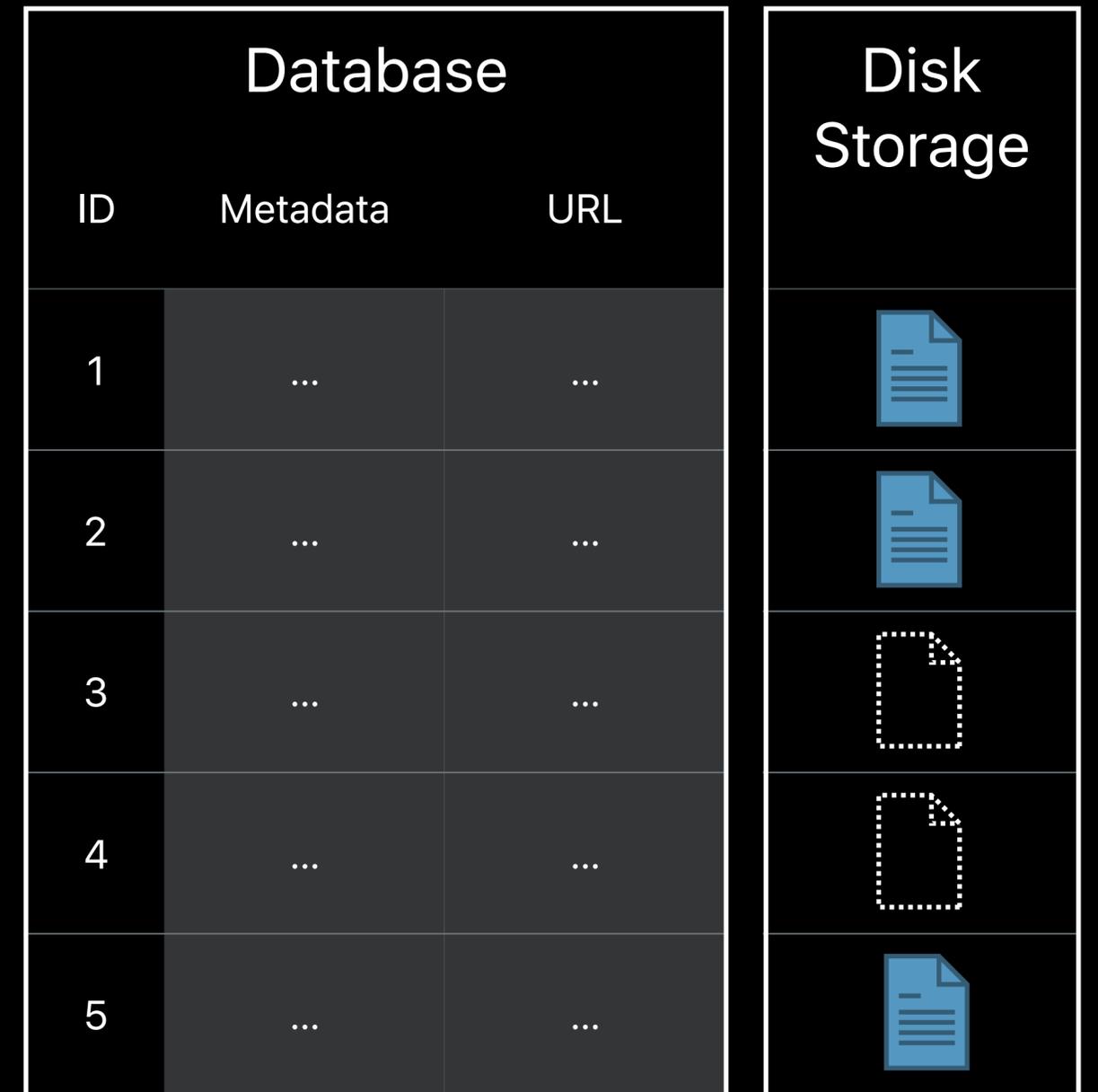


Anatomy of a File Provider Extension

Item identifier \rightarrow metadata

Item identifier \leftrightarrow file URL

You manage the disk storage



Providing an Item

Providing an Item

File URL

Providing an Item

File URL

File contents

Providing an Item

File URL

File contents

Metadata

Providing an Item

File URL

File contents

Metadata

Thumbnail

File URL

File provider manipulates `NSFileProviderItem` and `NSFileProviderItemIdentifier`

File URL

File provider manipulates `NSFileProviderItem` and `NSFileProviderItemIdentifier`

Document-Based apps consume file URLs

File URL

Picking a file

Files app

Document-Based app

File Provider Extension

File URL

Picking a file

Files app

User taps on a file

Document-Based app

File Provider Extension

File URL

Picking a file

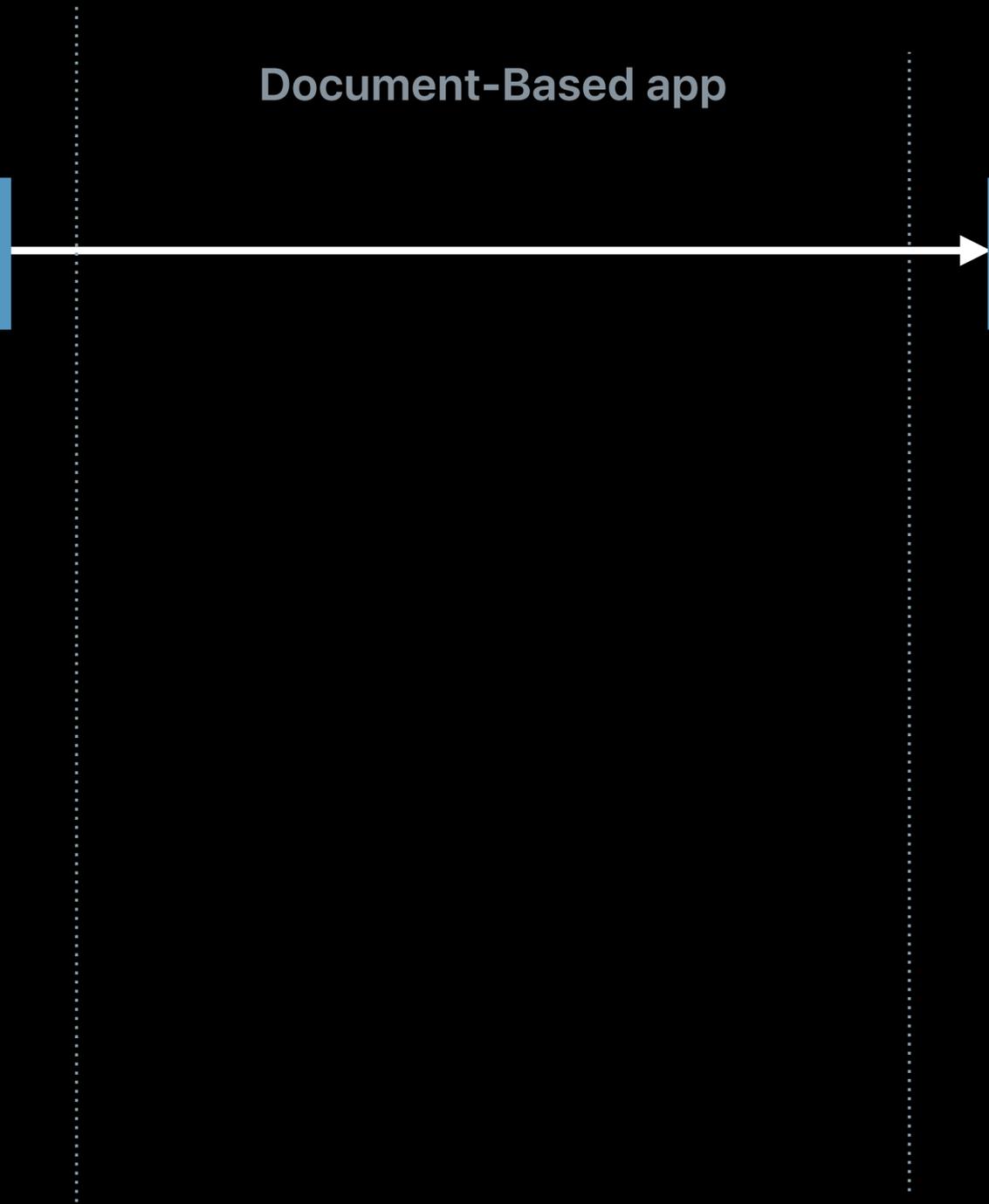
Files app

User taps on a file

Document-Based app

File Provider Extension

```
func urlForItem(with identifier:)
```



File URL

Picking a file

Files app

User taps on a file

Document-Based app

File Provider Extension

```
func urlForItem(with identifier:)
```

ID	Metadata	URL	Storage
1	
2	

File URL

Picking a file

Files app

User taps on a file

Document-Based app

File Provider Extension

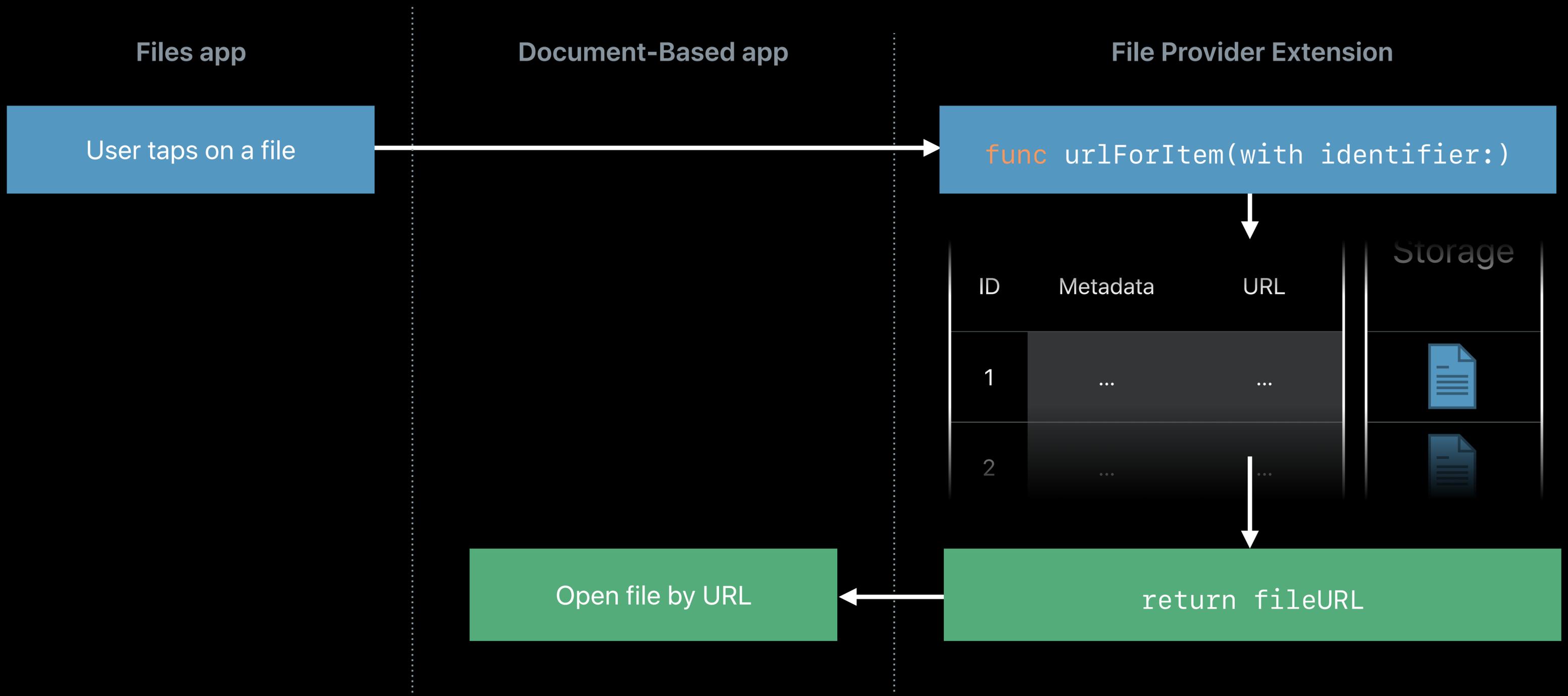
```
func urlForItem(with identifier:)
```

ID	Metadata	URL	Storage
1	
2	

```
return fileURL
```

File URL

Picking a file



File URL

Under the hood

File URL

Under the hood

Store the item identifier in the file URL

```
File Provider Storage/<itemIdentifier>/My Awesome Image.png
```

File URL

Under the hood

Store the item identifier in the file URL

```
File Provider Storage/<itemIdentifier>/My Awesome Image.png
```

- Avoid database lookup

File URL

Under the hood

Store the item identifier in the file URL

```
File Provider Storage/<itemIdentifier>/My Awesome Image.png
```

- Avoid database lookup
- Track item between renames

File URL

Under the hood

Store the item identifier in the file URL

```
File Provider Storage/<itemIdentifier>/My Awesome Image.png
```

- Avoid database lookup
- Track item between renames

Keep a flat list—no need to replicate the folder hierarchy

File Contents

Document-Based app

File Provider Extension

NSURLSession

File Contents

Document-Based app

Open file by URL

File Provider Extension

NSURLSession

File Contents

Document-Based app

Open file by URL



`NSFileCoordinator` read

File Provider Extension

`NSURLSession`

File Contents

Document-Based app

Open file by URL



`NSFileCoordinator` read



File Provider Extension

```
func startProvidingItem(at:completionHandler:)
```

NSURLSession

File Contents

Document-Based app

Open file by URL

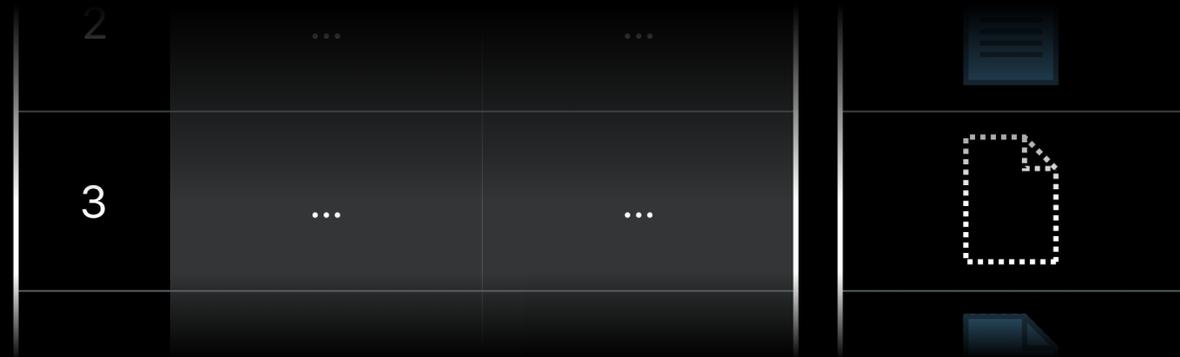


`NSFileCoordinator` read



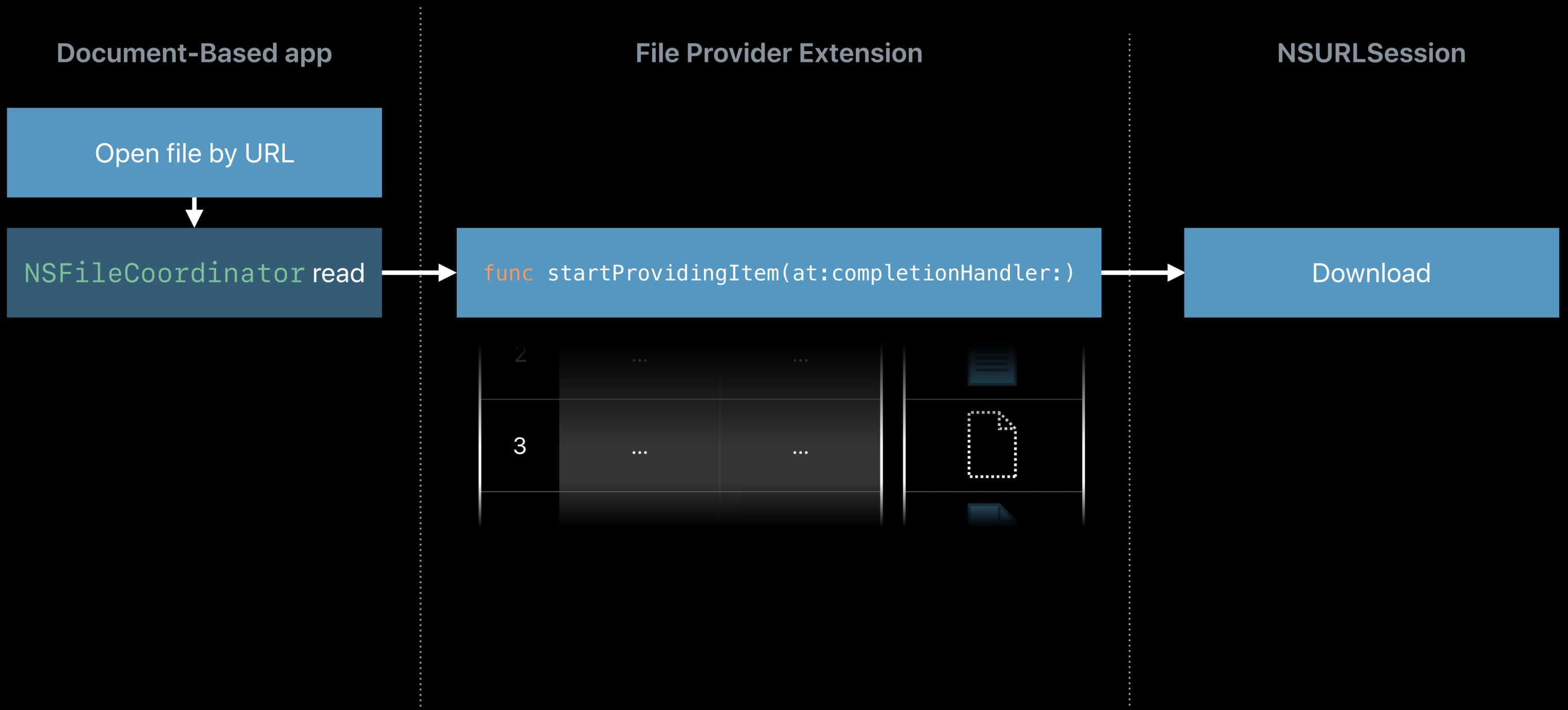
File Provider Extension

`func startProvidingItem(at:completionHandler:)`



NSURLSession

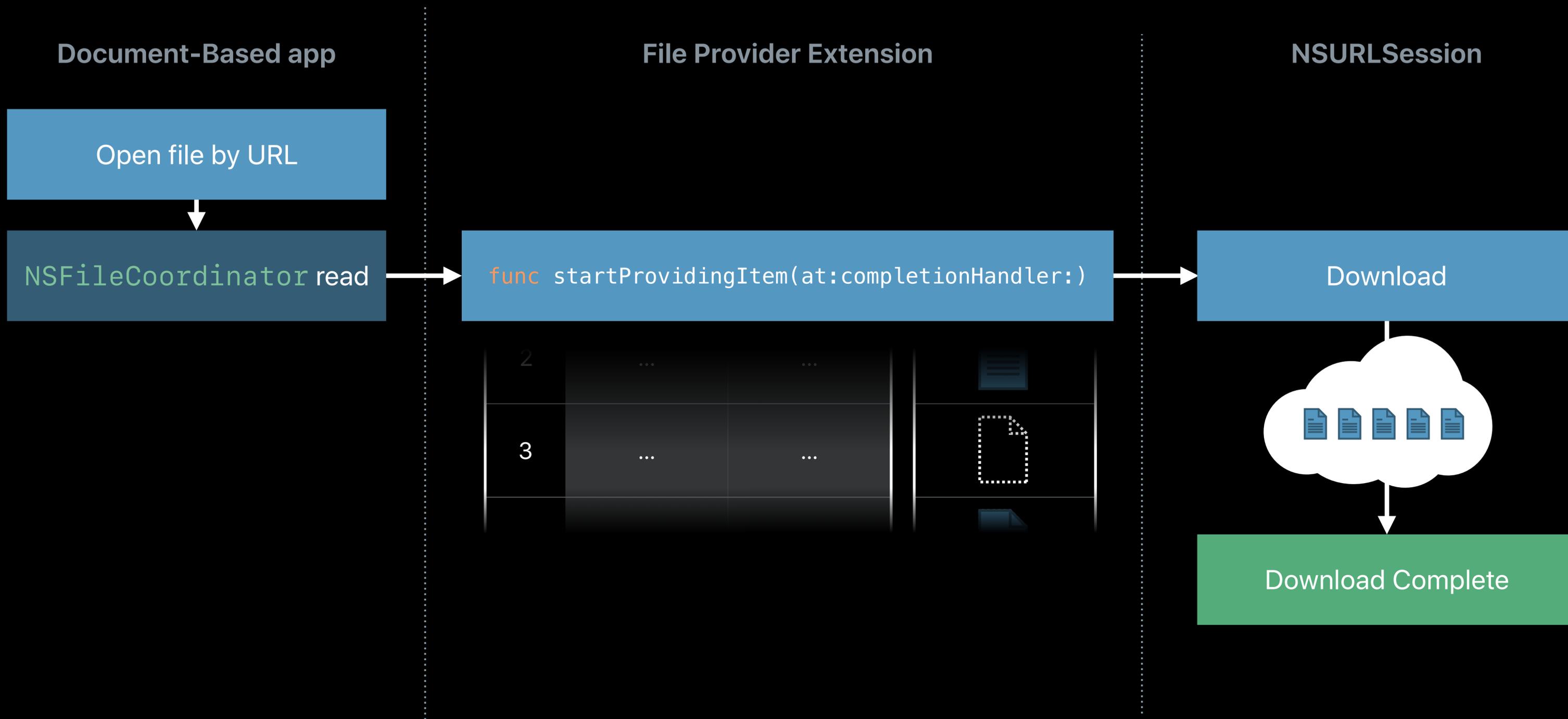
File Contents



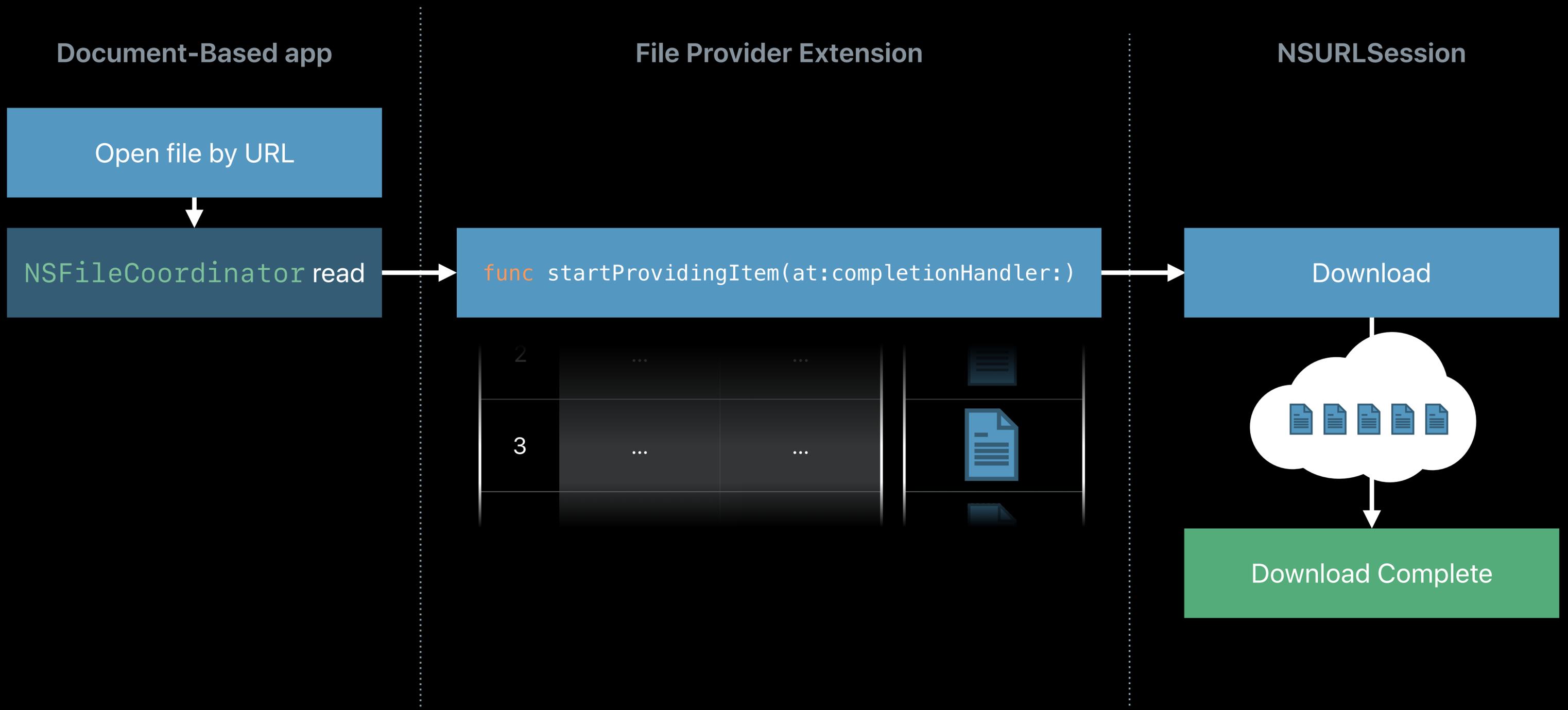
File Contents



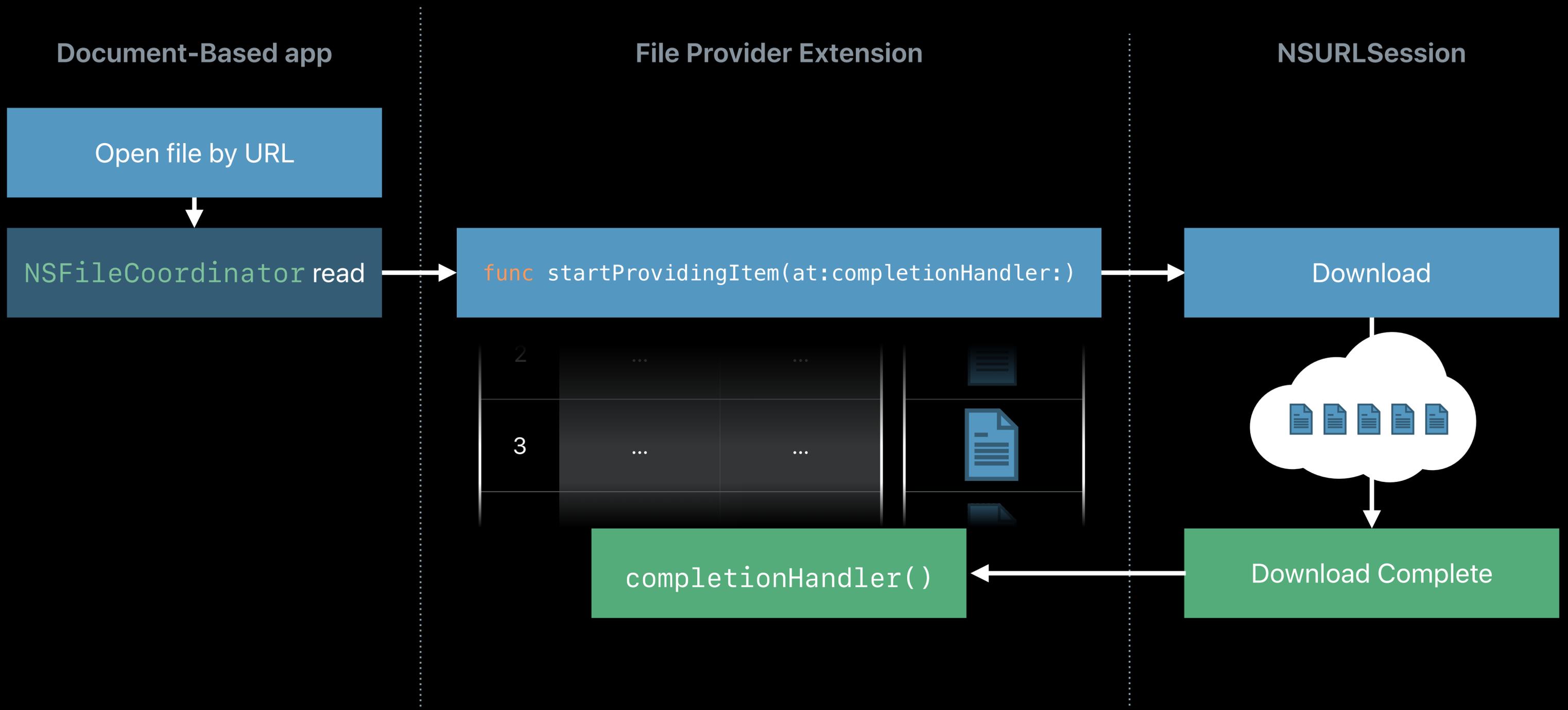
File Contents



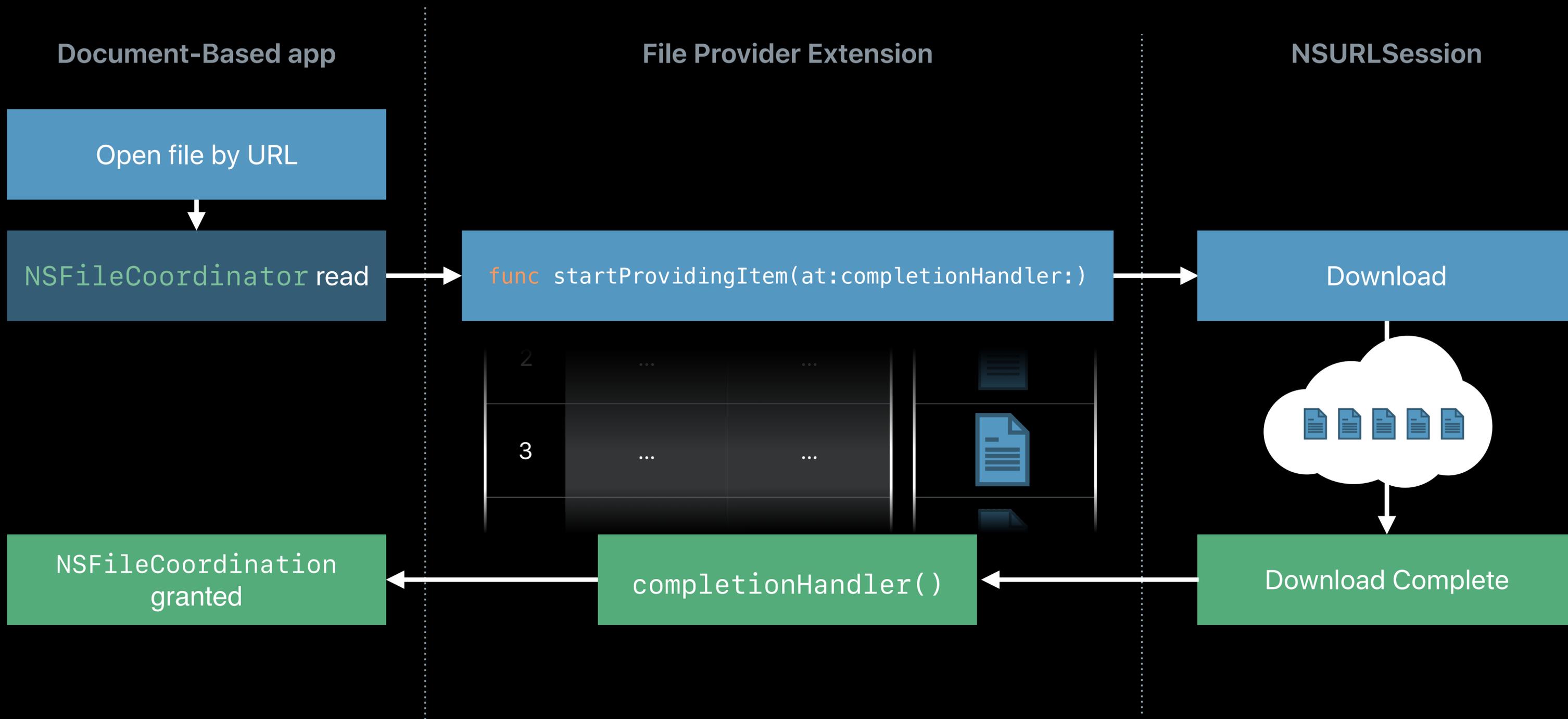
File Contents



File Contents



File Contents



File Contents



File Contents

Apps

File Provider Extension

NSURLSession

File Contents

Apps

Open file by URL

File Provider Extension

NSURLSession

File Contents

Apps

Open file by URL



`NSFileCoordinator` read

File Provider Extension

NSURLSession

File Contents

Apps

Open file by URL

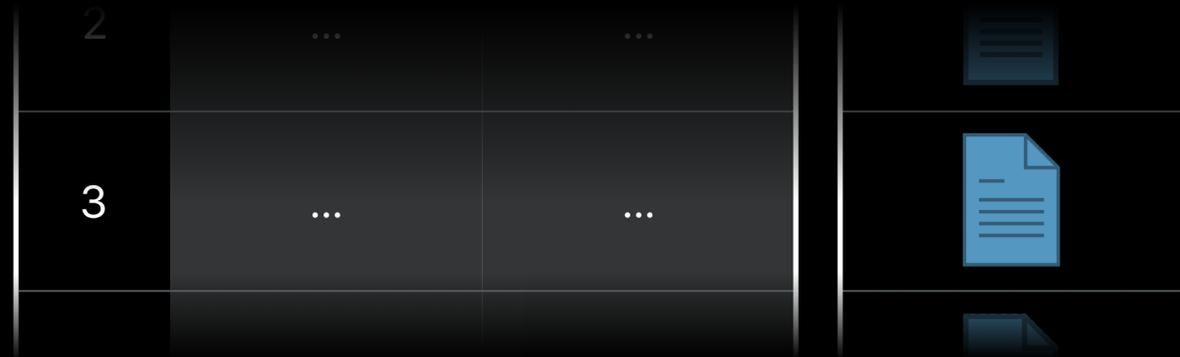


`NSFileCoordinator` read



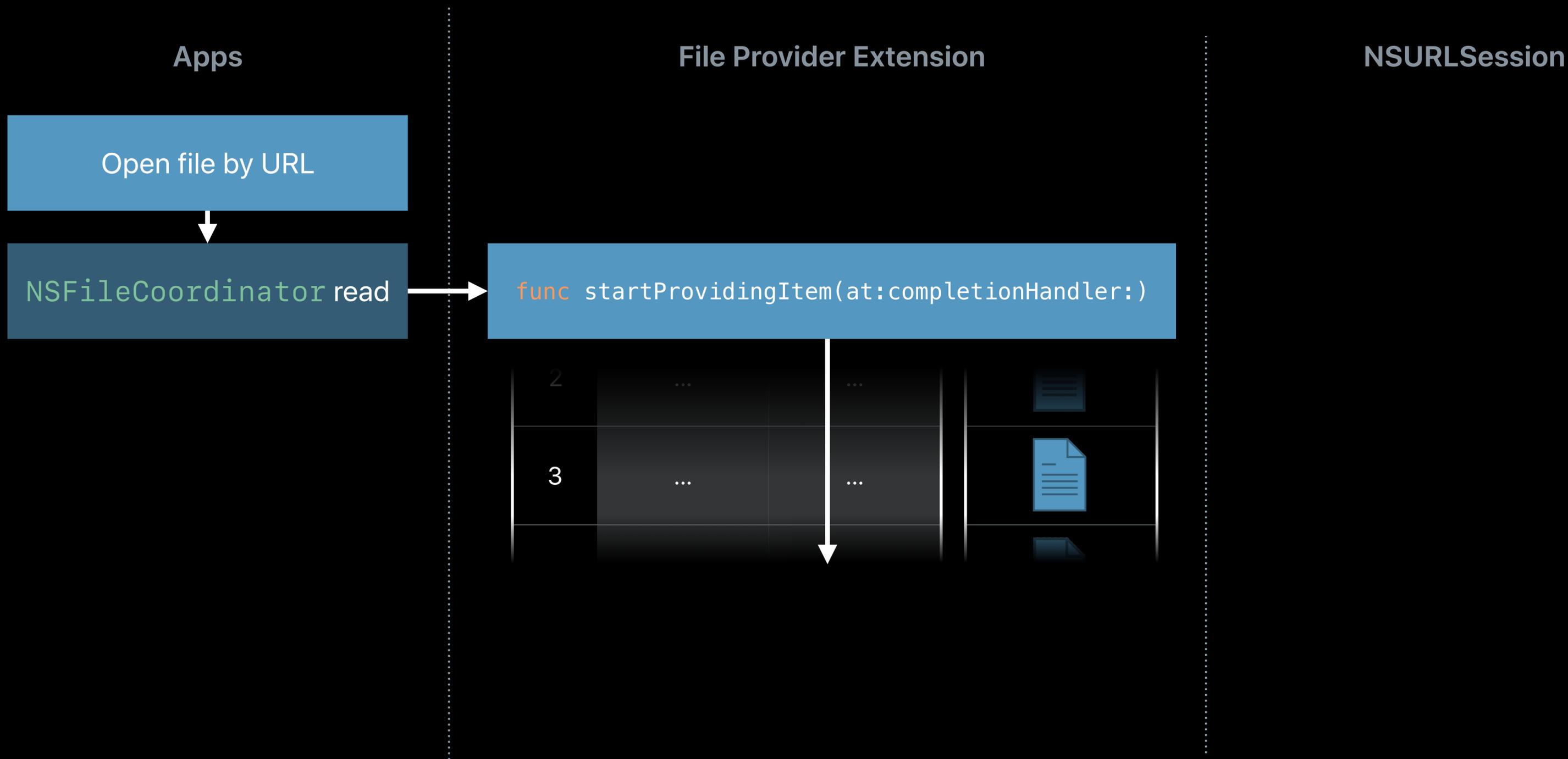
File Provider Extension

```
func startProvidingItem(at:completionHandler:)
```

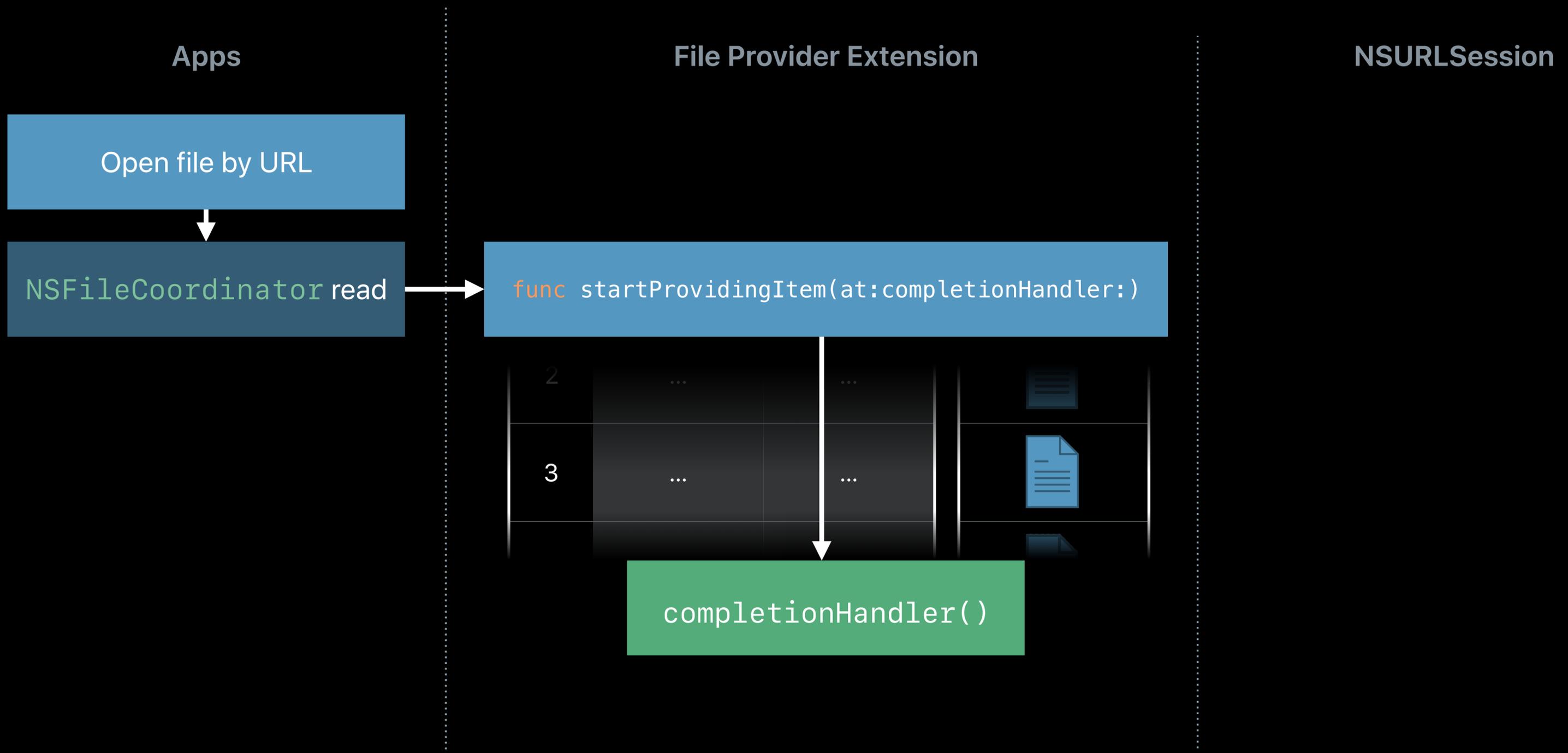


NSURLSession

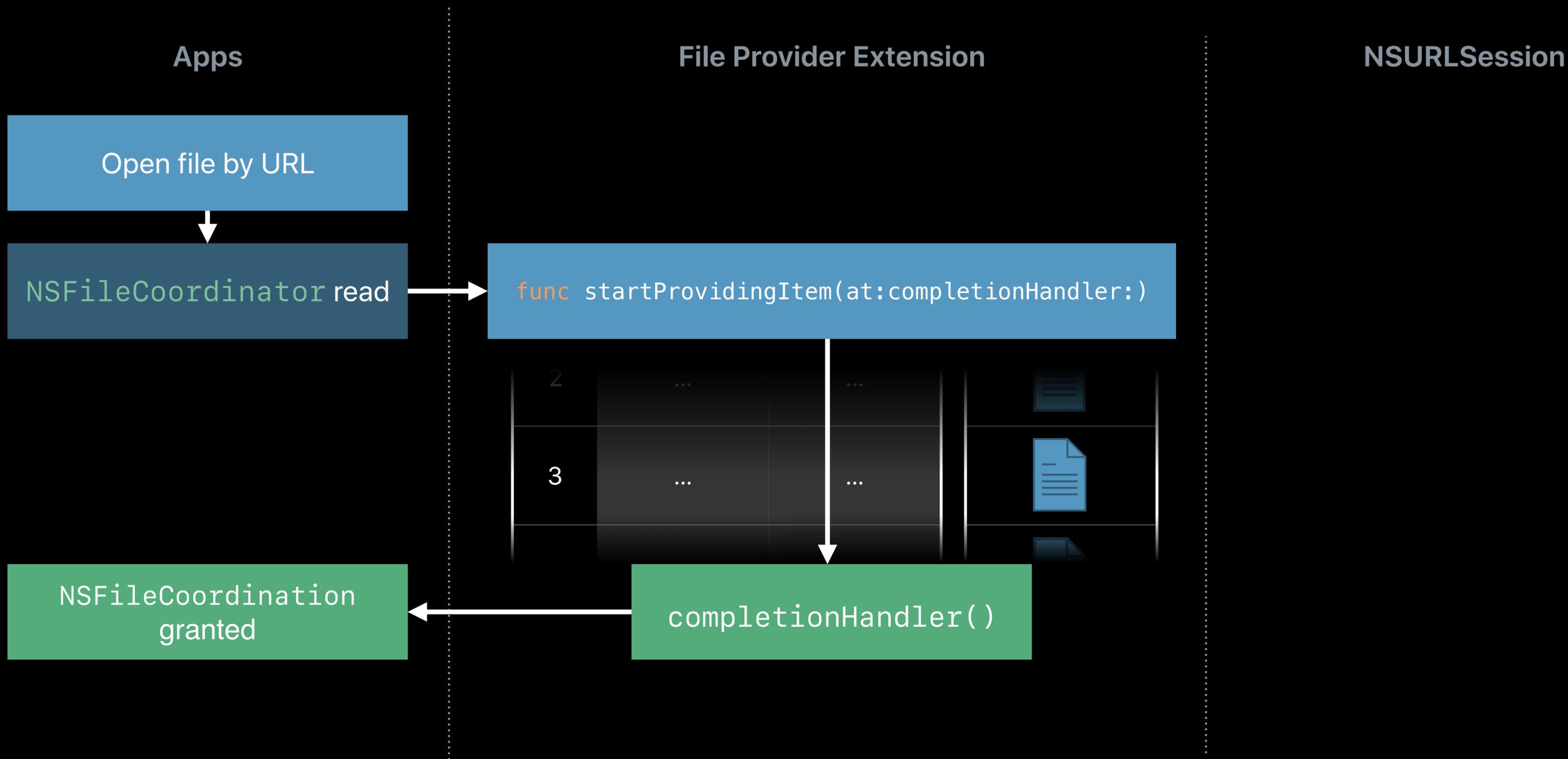
File Contents



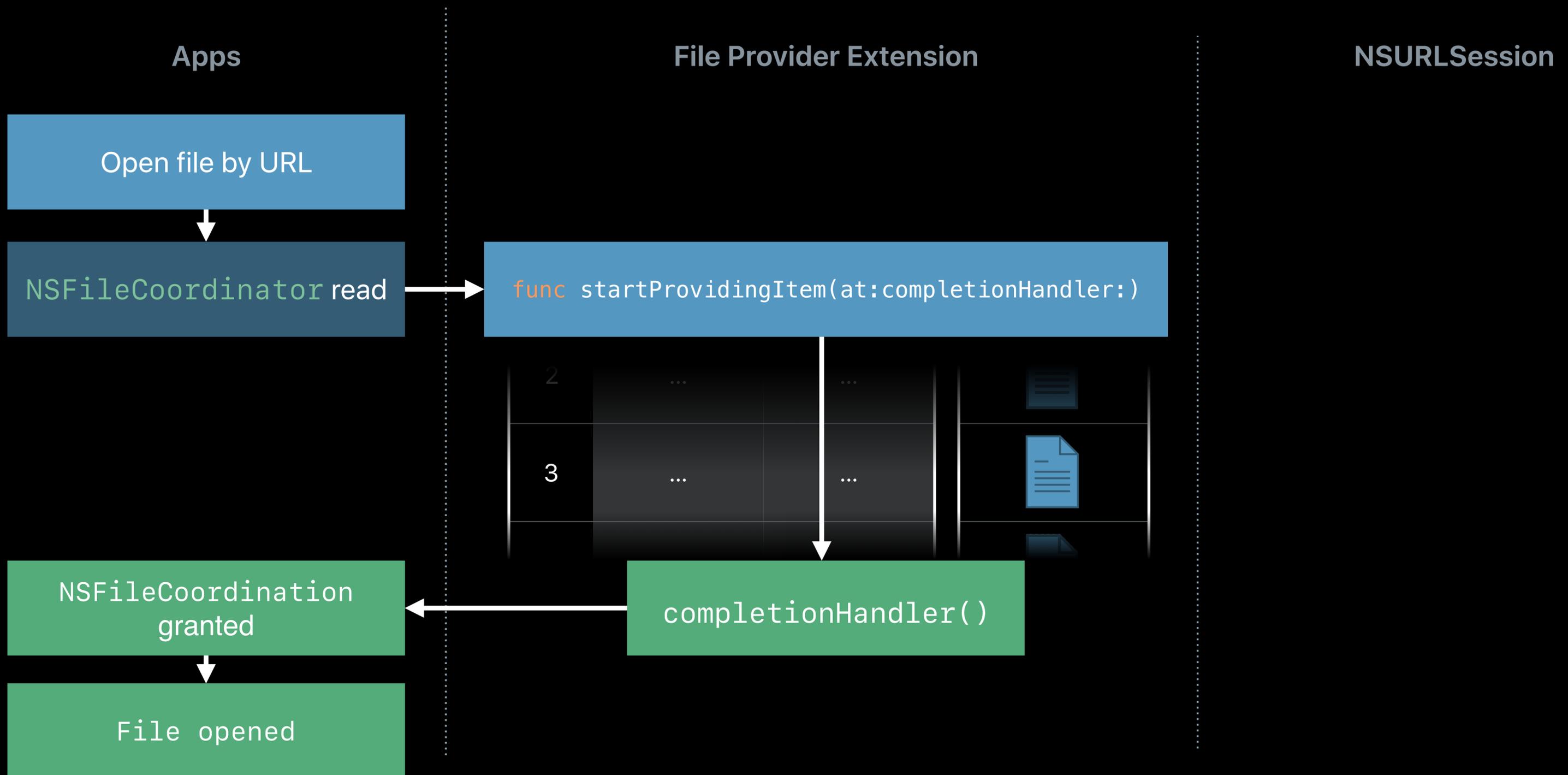
File Contents



File Contents



File Contents



Metadata

Files app



 **Pigments.key**
iCloud Drive



Kind	Keynote Presentation
Size	32.7 MB
Where	iCloud Drive ▸ Keynote
Created	April 14, 2017 at 8:26 AM
Modified	May 31, 2017 at 11:41 AM

Tags [Add Tag](#)

File Provider Extension

Metadata

Files app

Request Item Metadata



 **Pigments.key**
iCloud Drive


Kind Keynote Presentation
Size 32.7 MB
Where iCloud Drive › Keynote
Created April 14, 2017 at 8:26 AM
Modified May 31, 2017 at 11:41 AM

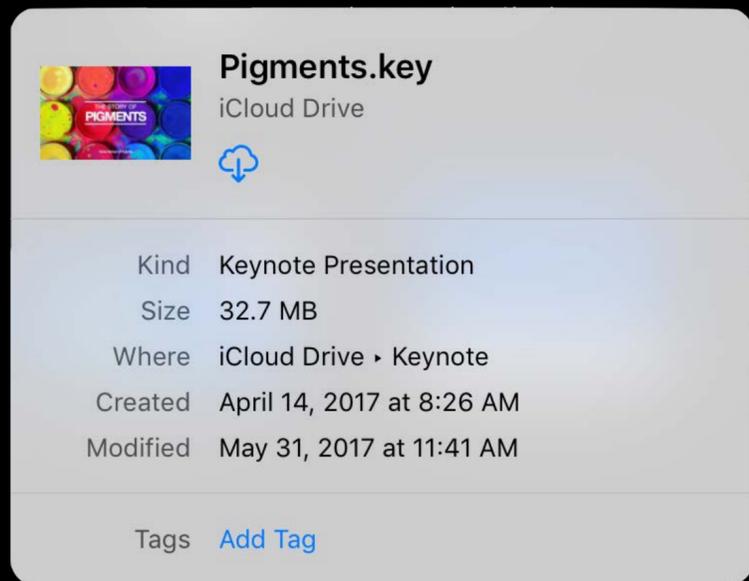
Tags [Add Tag](#)

File Provider Extension

Metadata

Files app

Request Item Metadata



File Provider Extension

```
func item(for itemIdentifier:) throws  
-> NSFileProviderItem
```

Metadata

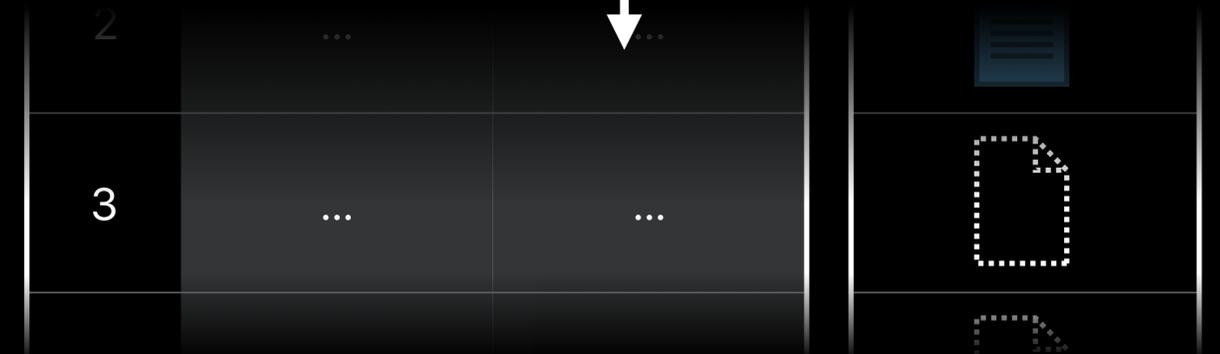
Files app

Request Item Metadata



File Provider Extension

```
func item(for itemIdentifier:) throws  
-> NSFileProviderItem
```



Metadata

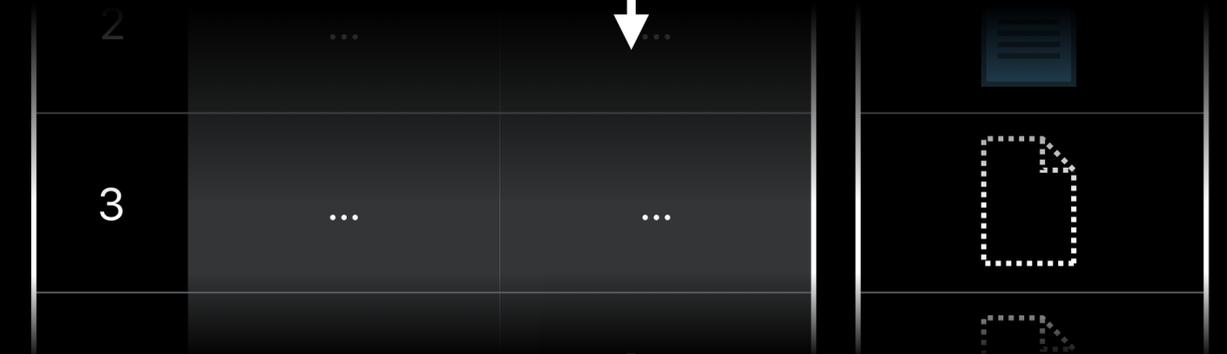
Files app

Request Item Metadata



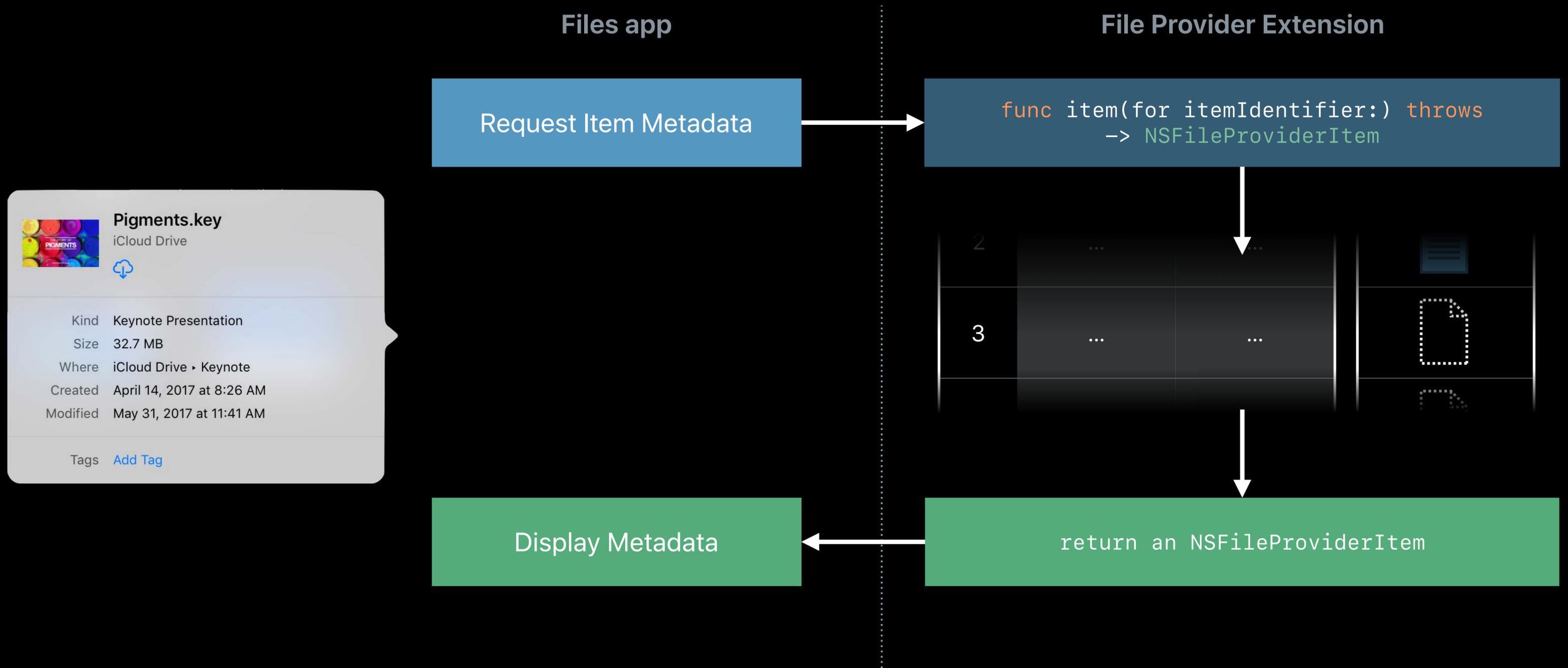
File Provider Extension

```
func item(for itemIdentifier:) throws  
-> NSFileProviderItem
```



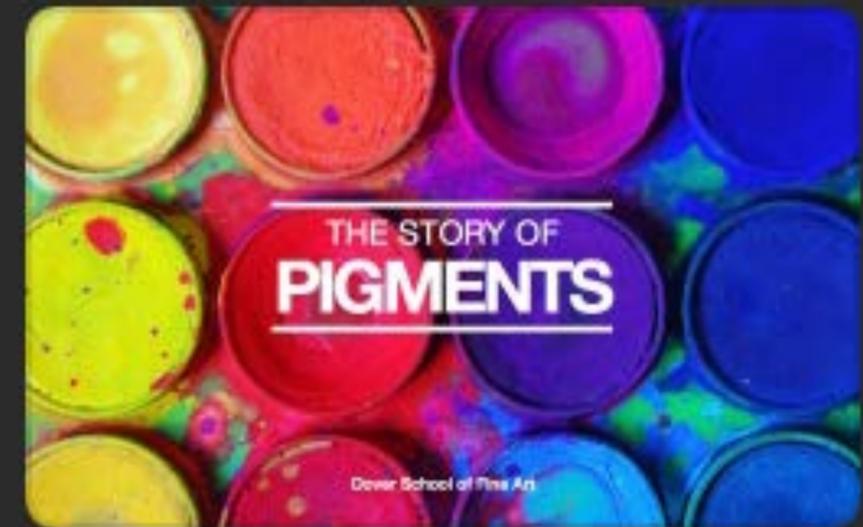
return an NSFileProviderItem

Metadata



Metadata

NSFileProviderItem required properties



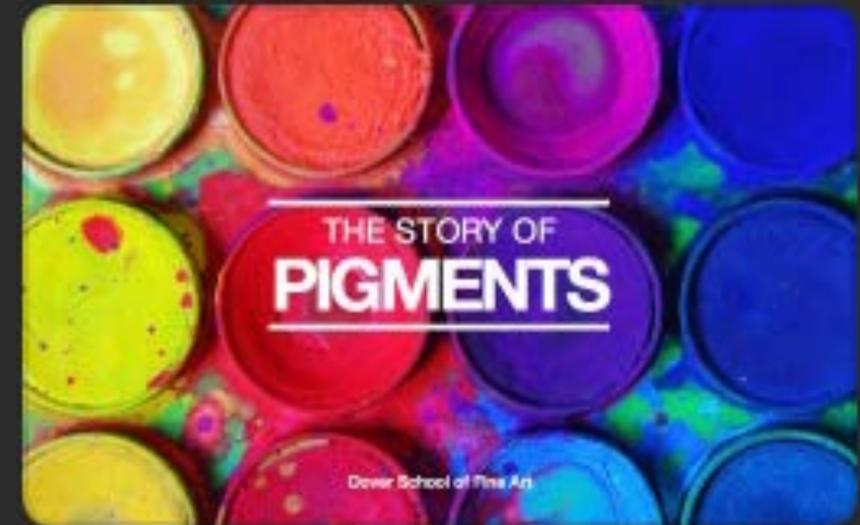
Pigments

Shared by Amaury

Metadata

NSFileProviderItem required properties

```
var itemIdentifier: NSFileProviderItemIdentifier
```



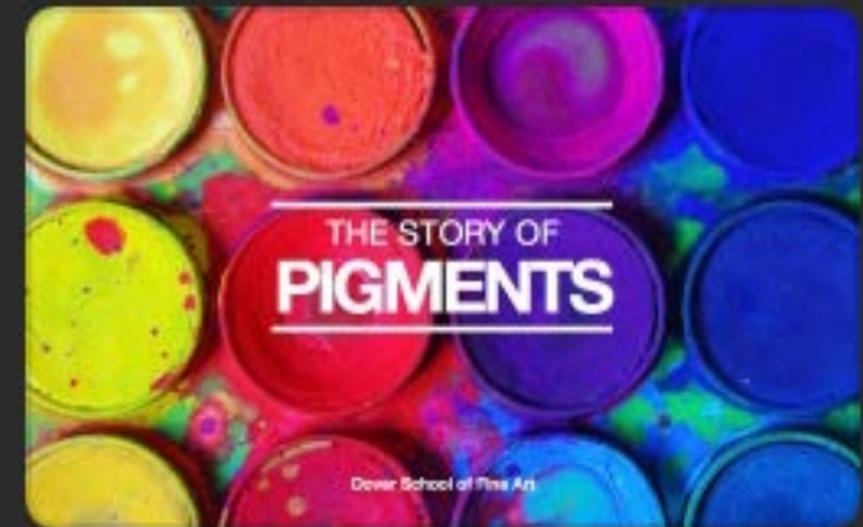
Pigments

Shared by Amaury

Metadata

NSFileProviderItem required properties

```
var itemIdentifier: NSFileProviderItemIdentifier
var typeIdentifier: String
```



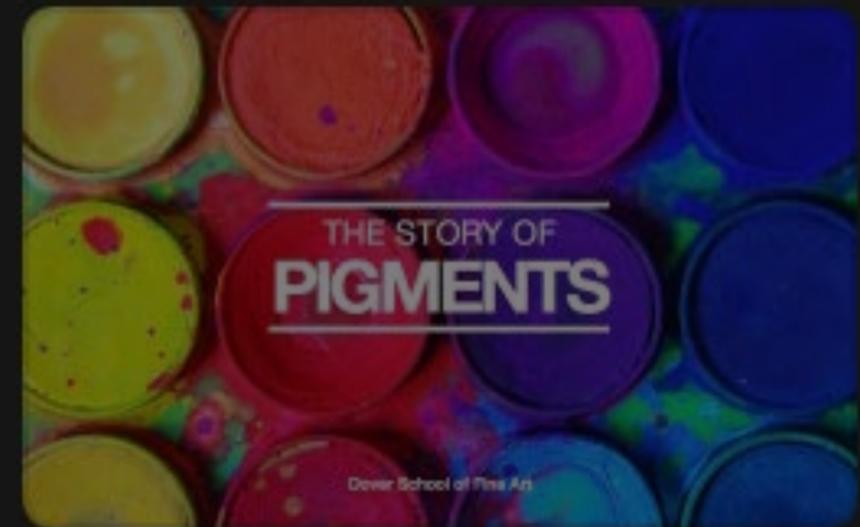
Pigments

Shared by Amaury

Metadata

NSFileProviderItem required properties

```
var itemIdentifier: NSFileProviderItemIdentifier
var typeIdentifier: String
var filename: String
```

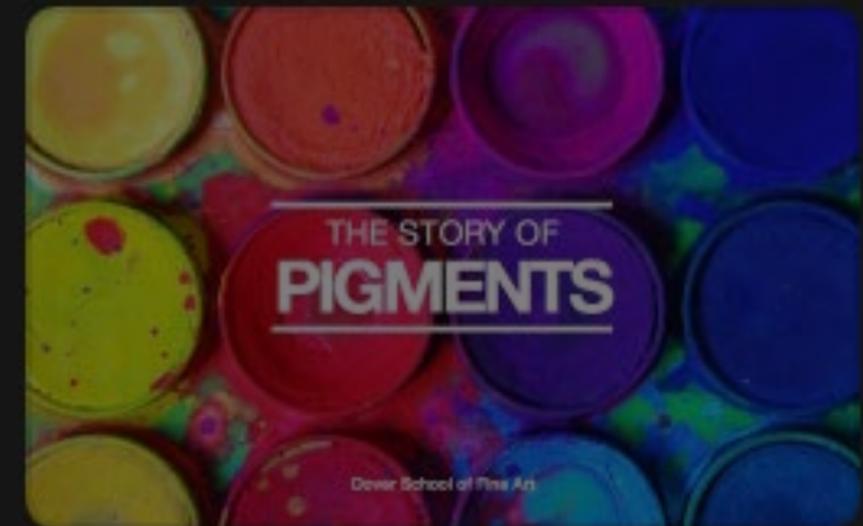


Pigments

Shared by Amaury

Metadata

NSFileProviderItem optional properties



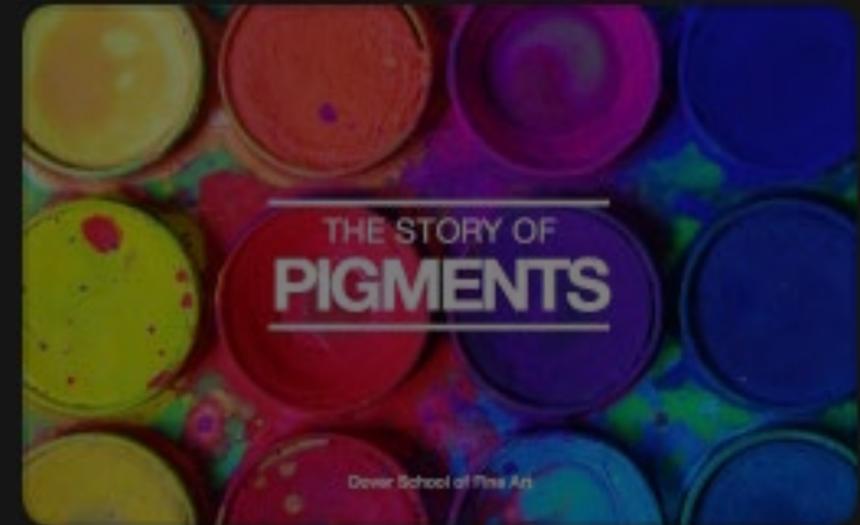
Pigments

Shared by Amaury

Metadata

NSFileProviderItem optional properties

```
var isShared: String
```



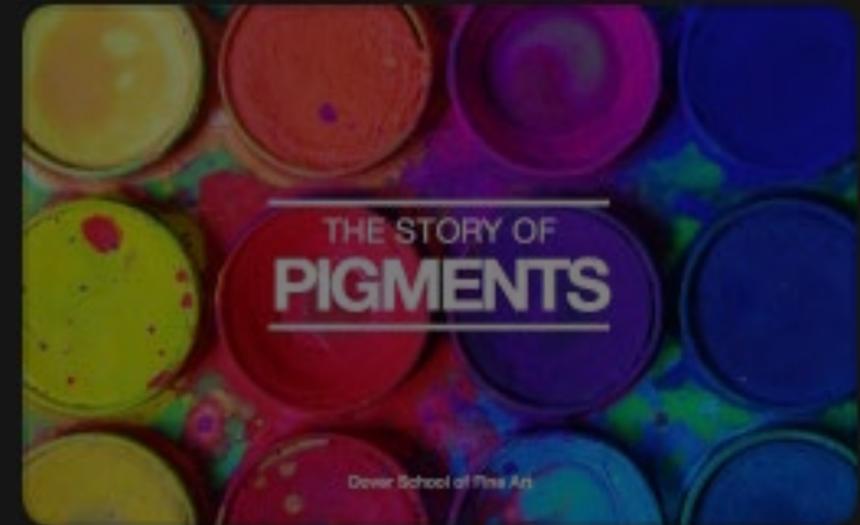
Pigments

Shared by Amaury

Metadata

NSFileProviderItem optional properties

```
var isShared: Bool  
var ownerNameComponents: PersonNameComponents
```



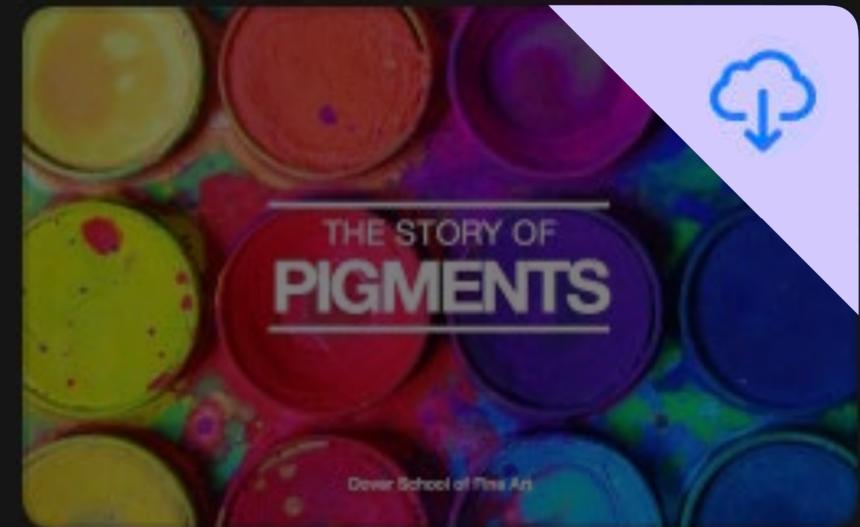
Pigments

Shared by Amaury

Metadata

NSFileProviderItem optional properties

```
var isShared: Bool
var ownerNameComponents: PersonNameComponents
var isDownloaded: Bool
```



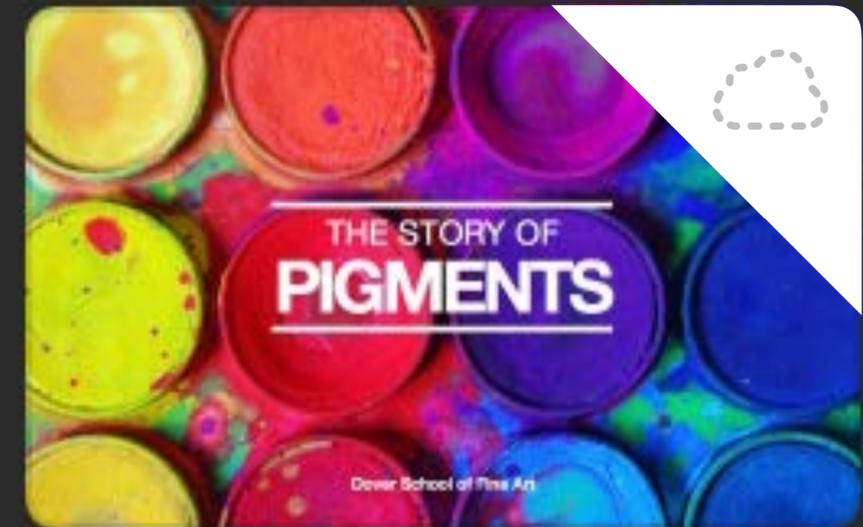
Pigments

Shared by Amaury

Metadata

NSFileProviderItem optional properties

```
var isShared: Bool
var ownerNameComponents: PersonNameComponents
var isDownloaded: Bool
var isUploading: Bool
```

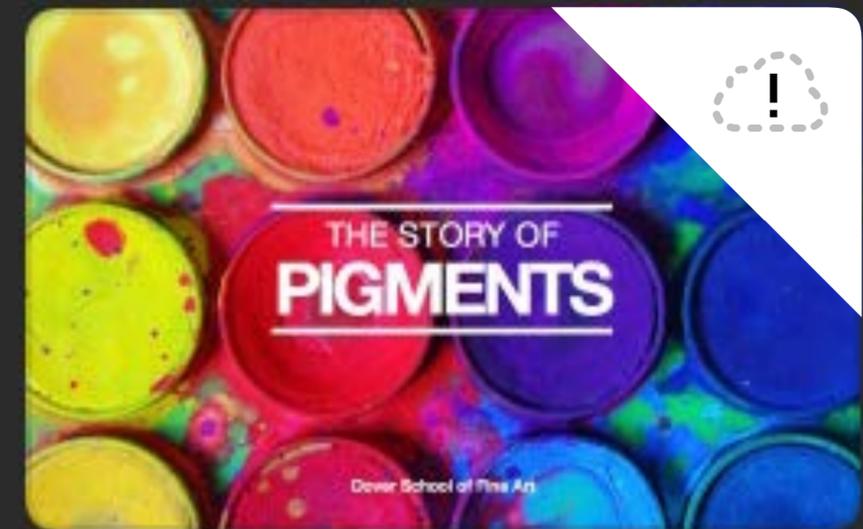


Pigments
Shared by Amaury

Metadata

NSFileProviderItem optional properties

```
var isShared: Bool
var ownerNameComponents: PersonNameComponents
var isDownloaded: Bool
var isUploading: Bool
var uploadError: Error?
```

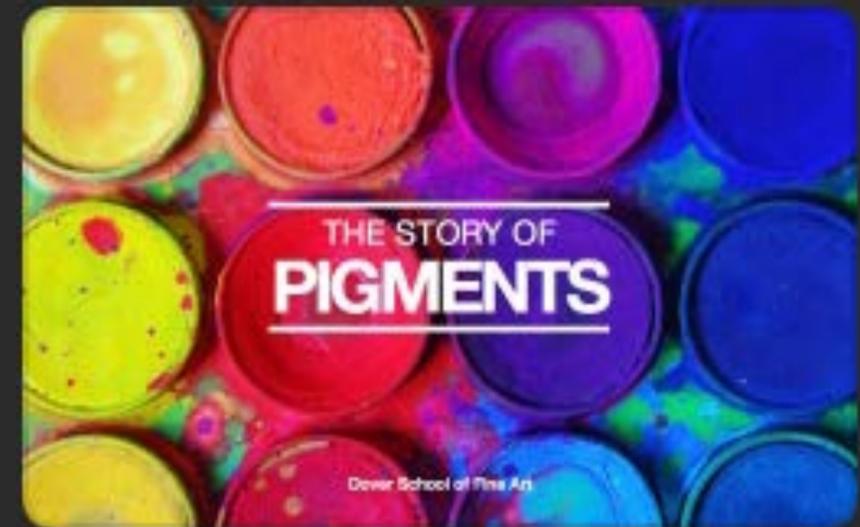


Pigments
Shared by Amaury

Metadata

NSFileProviderItem optional properties

```
var isShared: Bool
var ownerNameComponents: PersonNameComponents
var isDownloaded: Bool
var isUploading: Bool
var uploadError: Error?
// .. and more!
```

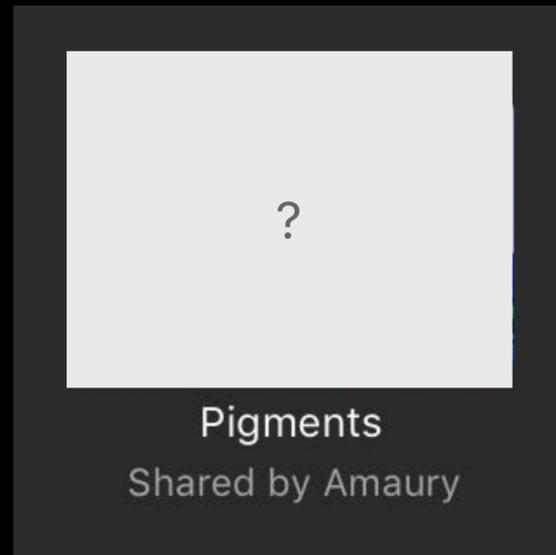


Pigments

Shared by Amaury

Thumbnail

Files app



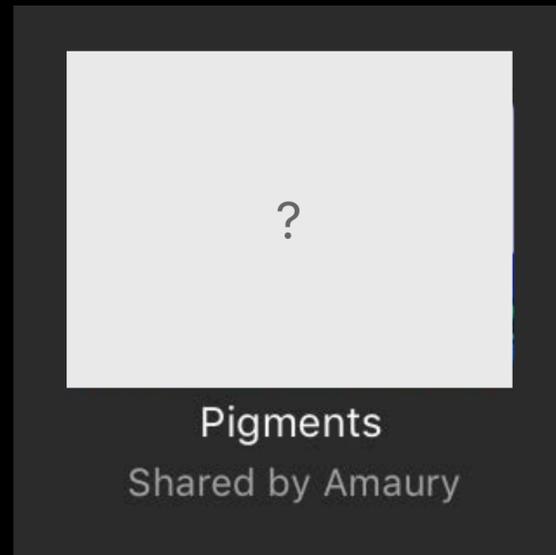
File Provider Extension

NSURLSession

Thumbnail

Files app

Request Thumbnail for an Item



File Provider Extension

NSURLSession

Thumbnail

Files app

File Provider Extension

NSURLSession

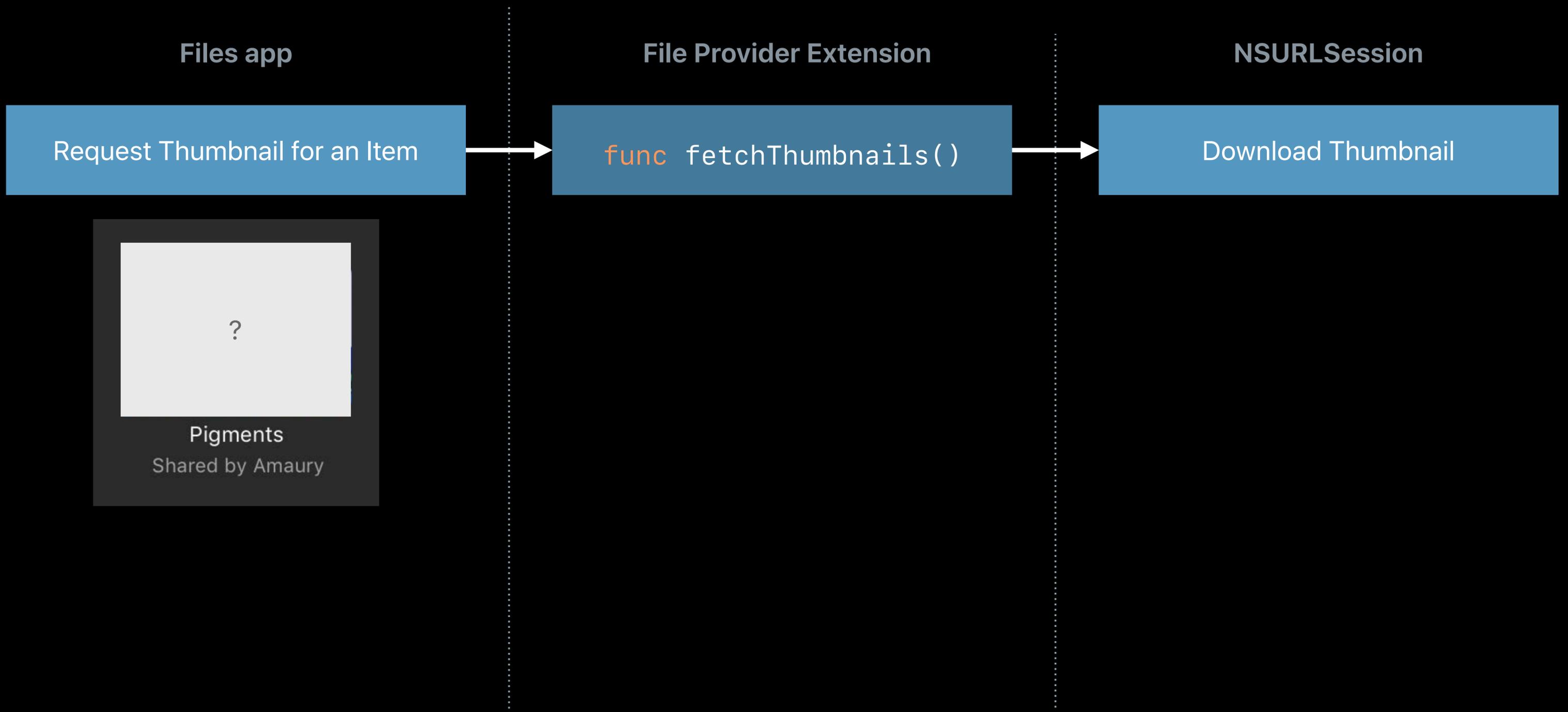
Request Thumbnail for an Item

`func fetchThumbnails()`

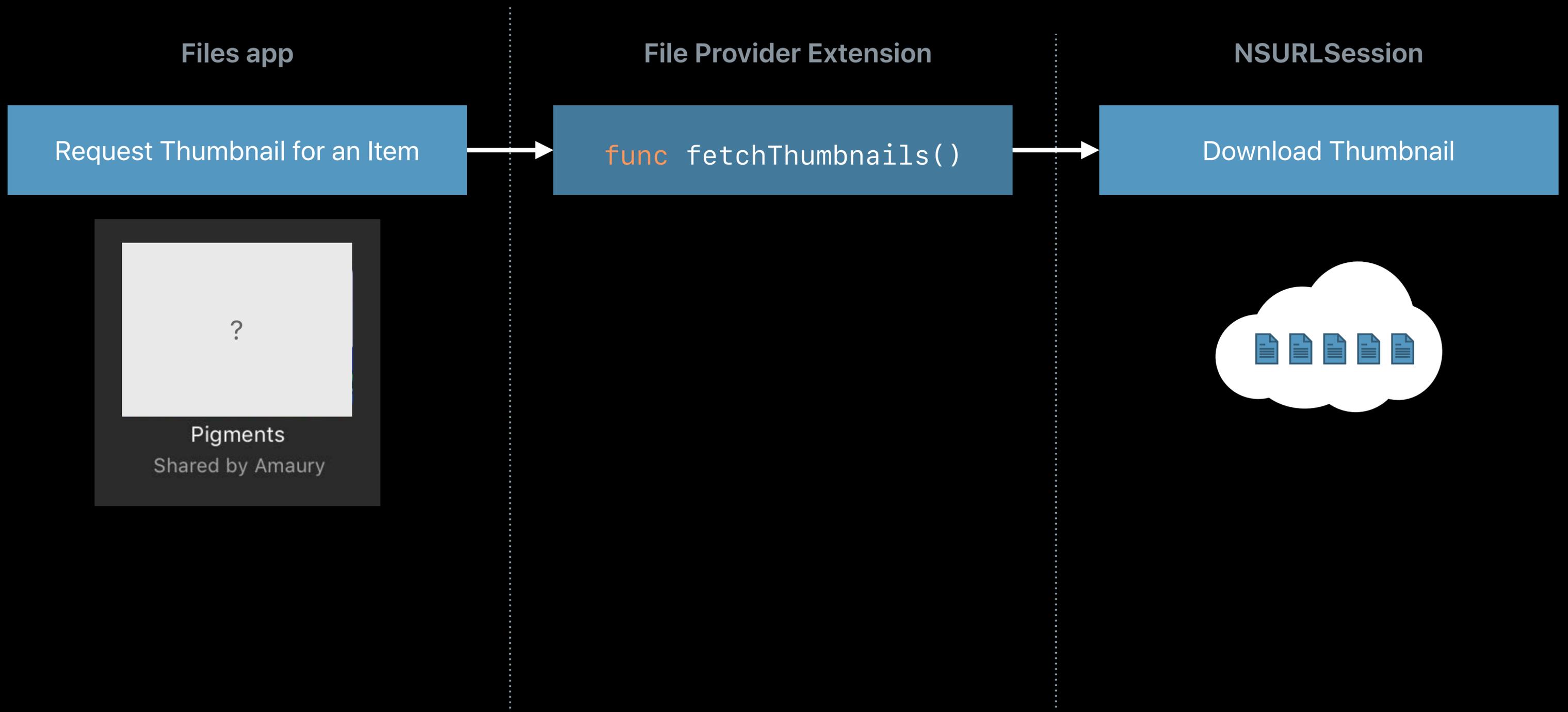
?

Pigments
Shared by Amaury

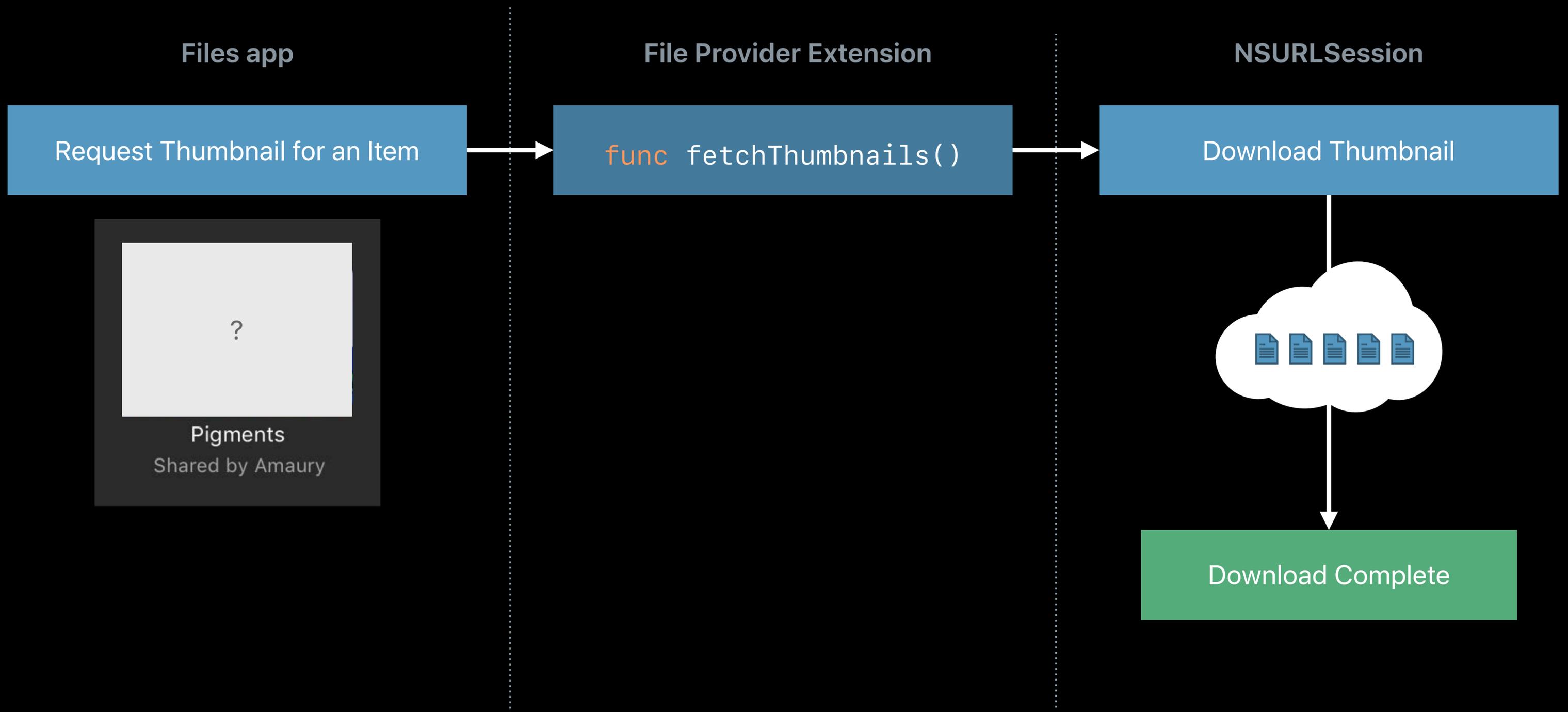
Thumbnail



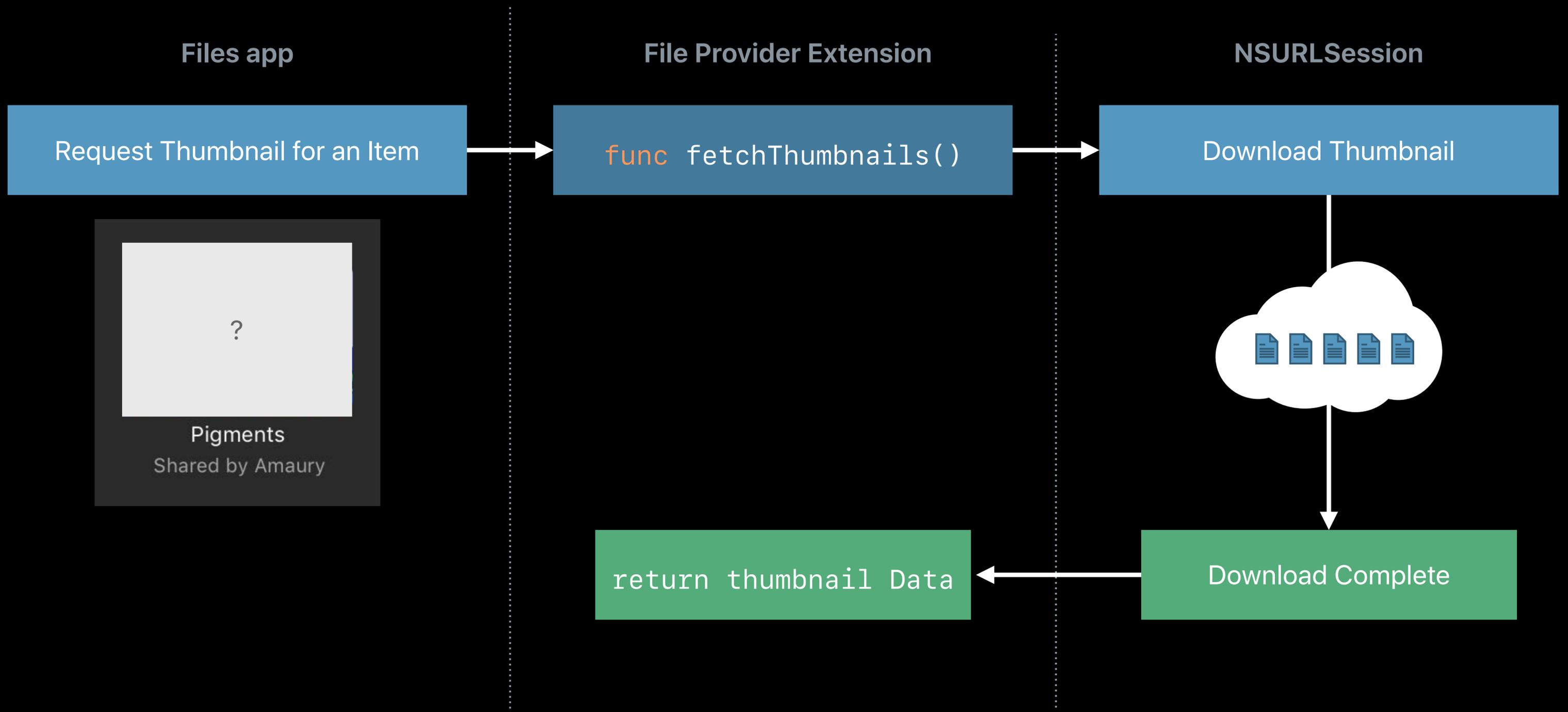
Thumbnail



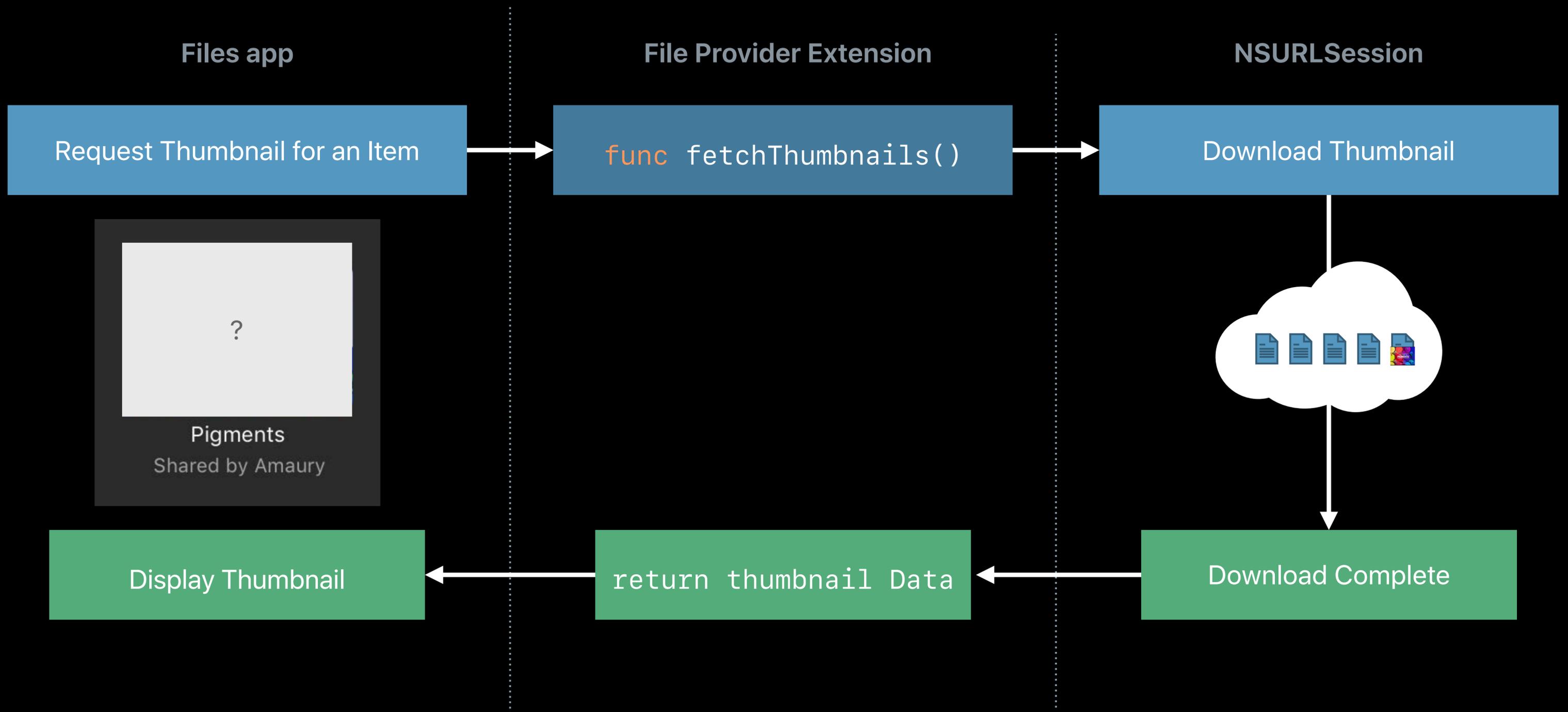
Thumbnail



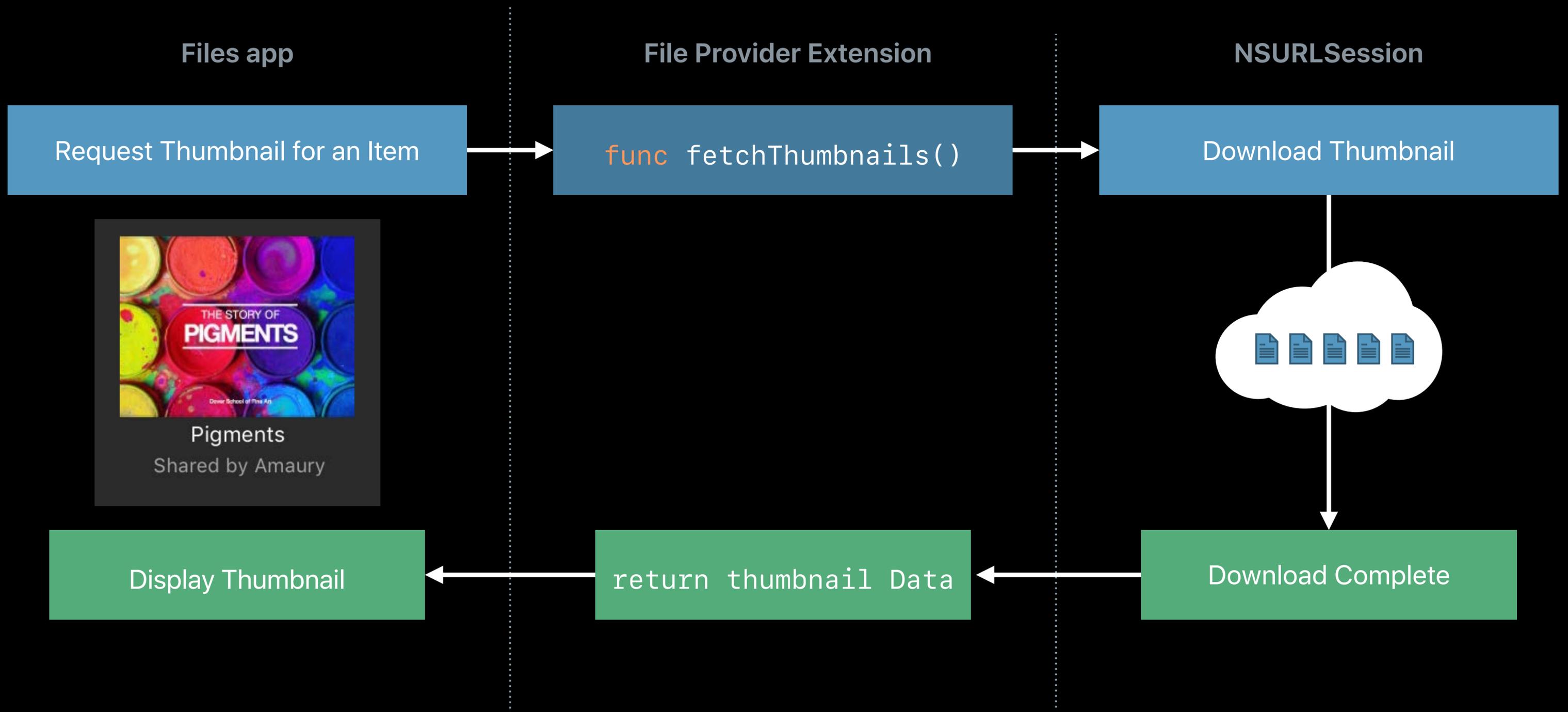
Thumbnail



Thumbnail



Thumbnail




```
class MyFileProviderExtension : NSFileProviderExtension {  
    override  
    func fetchThumbnails(  

```

```
class MyFileProviderExtension : NSFileProviderExtension {  
    override  
    func fetchThumbnails(  
        for itemIdentifiers: [NSFileProviderItemIdentifier],
```

```
class MyFileProviderExtension : NSFileProviderExtension {  
    override  
    func fetchThumbnails(  
        for itemIdentifiers: [NSFileProviderItemIdentifier],  
        requestedSize: CGSize,  
    )
```

```
class MyFileProviderExtension : NSFileProviderExtension {  
    override  
    func fetchThumbnails(  
        for itemIdentifiers: [NSFileProviderItemIdentifier],  
        requestedSize: CGSize,  
        perThumbnailCompletionHandler: @escaping (NSFileProviderItemIdentifier, Data?, Error?) -> Void,  
    )
```

```
class MyFileProviderExtension : NSFileProviderExtension {  
    override  
    func fetchThumbnails(  
        for itemIdentifiers: [NSFileProviderItemIdentifier],  
        requestedSize: CGSize,  
        perThumbnailCompletionHandler: @escaping (NSFileProviderItemIdentifier, Data?, Error?) -> Void,  
        completionHandler: @escaping (Error?) -> Void) -> Progress
```

```
class MyFileProviderExtension : NSFileProviderExtension {  
    override  
    func fetchThumbnails(  
        for itemIdentifiers: [NSFileProviderItemIdentifier],  
        requestedSize: CGSize,  
        perThumbnailCompletionHandler: @escaping (NSFileProviderItemIdentifier, Data?, Error?) -> Void,  
        completionHandler: @escaping (Error?) -> Void) -> Progress  
    {  
    }
```

```
class MyFileProviderExtension : NSFileProviderExtension {  
    override  
    func fetchThumbnails(  
        for itemIdentifiers: [NSFileProviderItemIdentifier],  
        requestedSize: CGSize,  
        perThumbnailCompletionHandler: @escaping (NSFileProviderItemIdentifier, Data?, Error?) -> Void,  
        completionHandler: @escaping (Error?) -> Void) -> Progress  
    {  
        let progress = Progress(totalUnitCount: itemIdentifiers.count)
```

```
class MyFileProviderExtension : NSFileProviderExtension {  
    override  
    func fetchThumbnails(  
        for itemIdentifiers: [NSFileProviderItemIdentifier],  
        requestedSize: CGSize,  
        perThumbnailCompletionHandler: @escaping (NSFileProviderItemIdentifier, Data?, Error?) -> Void,  
        completionHandler: @escaping (Error?) -> Void) -> Progress  
    {  
        let progress = Progress(totalUnitCount: itemIdentifiers.count)  
  
        progress.cancellable = true  
        progress.cancellationHandler = {  
            // Cancel your thumbnail request here...  
        }  
    }  
}
```

```
class MyFileProviderExtension : NSFileProviderExtension {  
  
    override  
    func fetchThumbnails(  
        for itemIdentifiers: [NSFileProviderItemIdentifier],  
        requestedSize: CGSize,  
        perThumbnailCompletionHandler: @escaping (NSFileProviderItemIdentifier, Data?, Error?) -> Void,  
        completionHandler: @escaping (Error?) -> Void) -> Progress  
    {  
        let progress = Progress(totalUnitCount: itemIdentifiers.count)  
  
        progress.cancellable = true  
        progress.cancellationHandler = {  
            // Cancel your thumbnail request here...  
        }  
  
        // download thumbnails from server using NSURLSessionDownloadTask...
```

```
class MyFileProviderExtension : NSFileProviderExtension {

    override
    func fetchThumbnails(
        for itemIdentifiers: [NSFileProviderItemIdentifier],
        requestedSize: CGSize,
        perThumbnailCompletionHandler: @escaping (NSFileProviderItemIdentifier, Data?, Error?) -> Void,
        completionHandler: @escaping (Error?) -> Void) -> Progress
    {
        let progress = Progress(totalUnitCount: itemIdentifiers.count)

        progress.cancellable = true
        progress.cancellationHandler = {
            // Cancel your thumbnail request here...
        }

        // download thumbnails from server using NSURLSessionDownloadTask...
        // ... when download completes:

        for thumbnail in ... {
```

```
class MyFileProviderExtension : NSFileProviderExtension {

    override
    func fetchThumbnails(
        for itemIdentifiers: [NSFileProviderItemIdentifier],
        requestedSize: CGSize,
        perThumbnailCompletionHandler: @escaping (NSFileProviderItemIdentifier, Data?, Error?) -> Void,
        completionHandler: @escaping (Error?) -> Void) -> Progress
    {
        let progress = Progress(totalUnitCount: itemIdentifiers.count)

        progress.cancellable = true
        progress.cancellationHandler = {
            // Cancel your thumbnail request here...
        }

        // download thumbnails from server using NSURLSessionDownloadTask...
        // ... when download completes:

        for thumbnail in ... {
            let data = Data(contentsOf: thumbnail.fileURL,
```

```
class MyFileProviderExtension : NSFileProviderExtension {  
  
    override  
    func fetchThumbnails(  
        for itemIdentifiers: [NSFileProviderItemIdentifier],  
        requestedSize: CGSize,  
        perThumbnailCompletionHandler: @escaping (NSFileProviderItemIdentifier, Data?, Error?) -> Void,  
        completionHandler: @escaping (Error?) -> Void) -> Progress  
    {  
        let progress = Progress(totalUnitCount: itemIdentifiers.count)  
  
        progress.cancellable = true  
        progress.cancellationHandler = {  
            // Cancel your thumbnail request here...  
        }  
  
        // download thumbnails from server using NSURLSessionDownloadTask...  
        // ... when download completes:  
  
        for thumbnail in ... {  
            let data = Data(contentsOf: thumbnail.fileURL,  
                            // Don't allocate memory, map the file in memory  
                            options: .alwaysMapped)  
        }  
    }  
}
```

```
class MyFileProviderExtension : NSFileProviderExtension {

    override
    func fetchThumbnails(
        for itemIdentifiers: [NSFileProviderItemIdentifier],
        requestedSize: CGSize,
        perThumbnailCompletionHandler: @escaping (NSFileProviderItemIdentifier, Data?, Error?) -> Void,
        completionHandler: @escaping (Error?) -> Void) -> Progress
    {
        let progress = Progress(totalUnitCount: itemIdentifiers.count)

        progress.cancellable = true
        progress.cancellationHandler = {
            // Cancel your thumbnail request here...
        }

        // download thumbnails from server using NSURLSessionDownloadTask...
        // ... when download completes:

        for thumbnail in ... {
            let data = Data(contentsOf: thumbnail.fileURL,
                            // Don't allocate memory, map the file in memory
                            options: .alwaysMapped)
            perThumbnailCompletionHandler(thumbnail.identifier, data, nil)
        }
    }
}
```

```
class MyFileProviderExtension : NSFileProviderExtension {

    override
    func fetchThumbnails(
        for itemIdentifiers: [NSFileProviderItemIdentifier],
        requestedSize: CGSize,
        perThumbnailCompletionHandler: @escaping (NSFileProviderItemIdentifier, Data?, Error?) -> Void,
        completionHandler: @escaping (Error?) -> Void) -> Progress
    {
        let progress = Progress(totalUnitCount: itemIdentifiers.count)

        progress.cancellable = true
        progress.cancellationHandler = {
            // Cancel your thumbnail request here...
        }

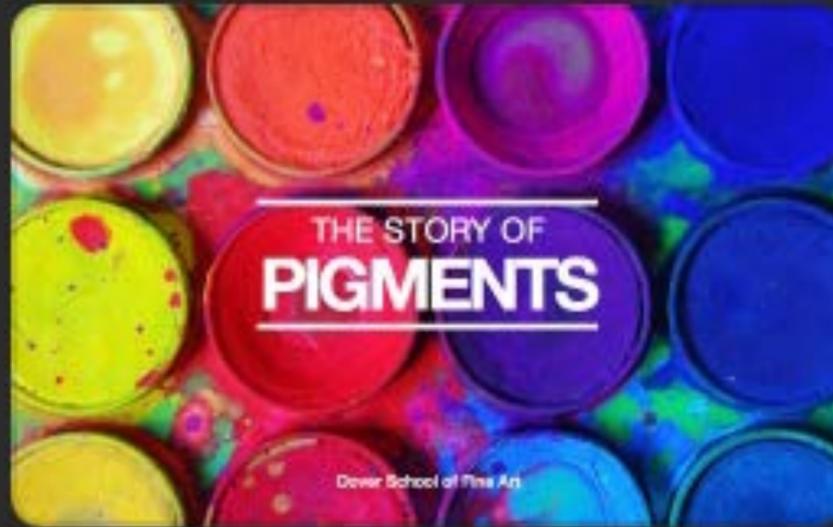
        // download thumbnails from server using NSURLSessionDownloadTask...
        // ... when download completes:

        for thumbnail in ... {
            let data = Data(contentsOf: thumbnail.fileURL,
                            // Don't allocate memory, map the file in memory
                            options: .alwaysMapped)
            perThumbnailCompletionHandler(thumbnail.identifier, data, nil)
        }

        // Always call completionHandler(): in case of success, failure or cancellation
        completionHandler(nil)
    }
}
```

```
class MyFileProviderExtension : NSFileProviderExtension {  
  
    override  
    func fetchThumbnails(  
        for itemIdentifiers: [NSFileProviderItemIdentifier],  
        requestedSize: CGSize,  
        perThumbnailCompletionHandler: @escaping (NSFileProviderItemIdentifier, Data?, Error?) -> Void,  
        completionHandler: @escaping (Error?) -> Void) -> Progress  
    {  
        let progress = Progress(totalUnitCount: itemIdentifiers.count)  
  
        progress.cancellable = true  
        progress.cancellationHandler = {  
            // Cancel your thumbnail request here...  
        }  
  
        // download thumbnails from server using NSURLSessionDownloadTask...  
        // ... when download completes:  
  
        for thumbnail in ... {  
            let data = Data(contentsOf: thumbnail.fileURL,  
                            // Don't allocate memory, map the file in memory  
                            options: .alwaysMapped)  
            perThumbnailCompletionHandler(thumbnail.identifier, data, nil)  
        }  
  
        // Always call completionHandler(): in case of success, failure or cancellation  
        completionHandler(nil)  
        return progress  
    }  
}
```

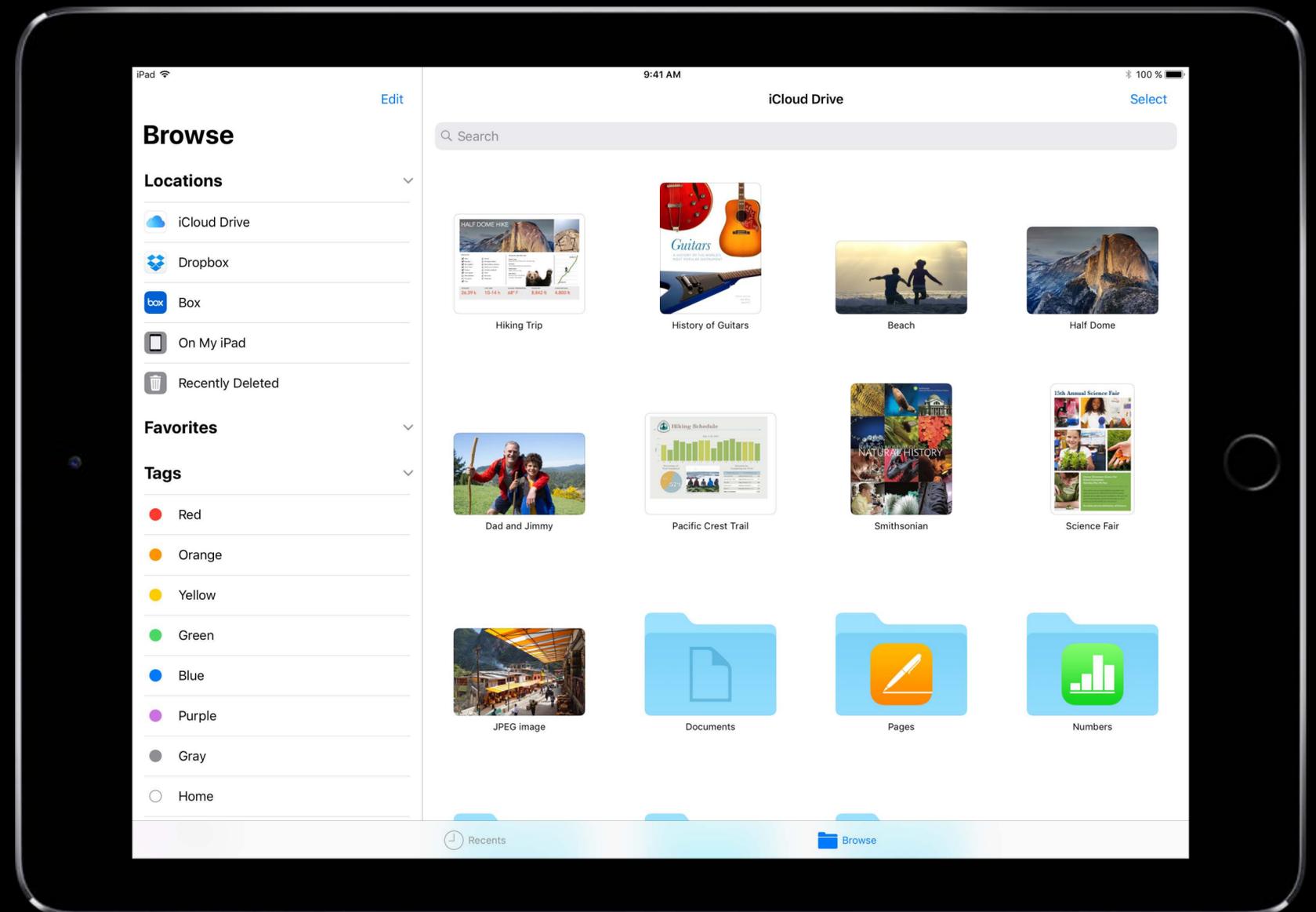
Provided an Item!



Pigments

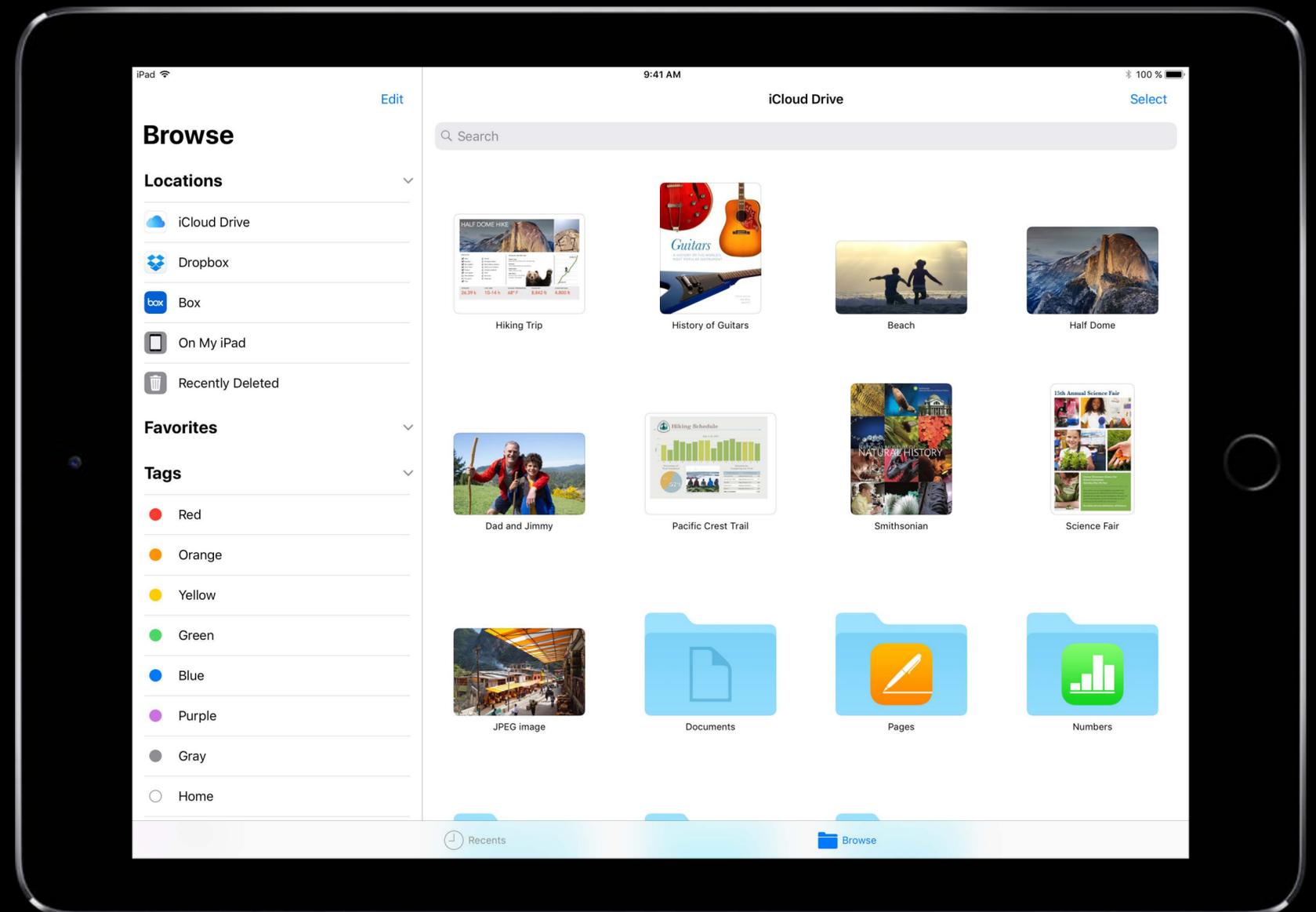
Shared by Amaury

Enumerating Items



Enumerating Items

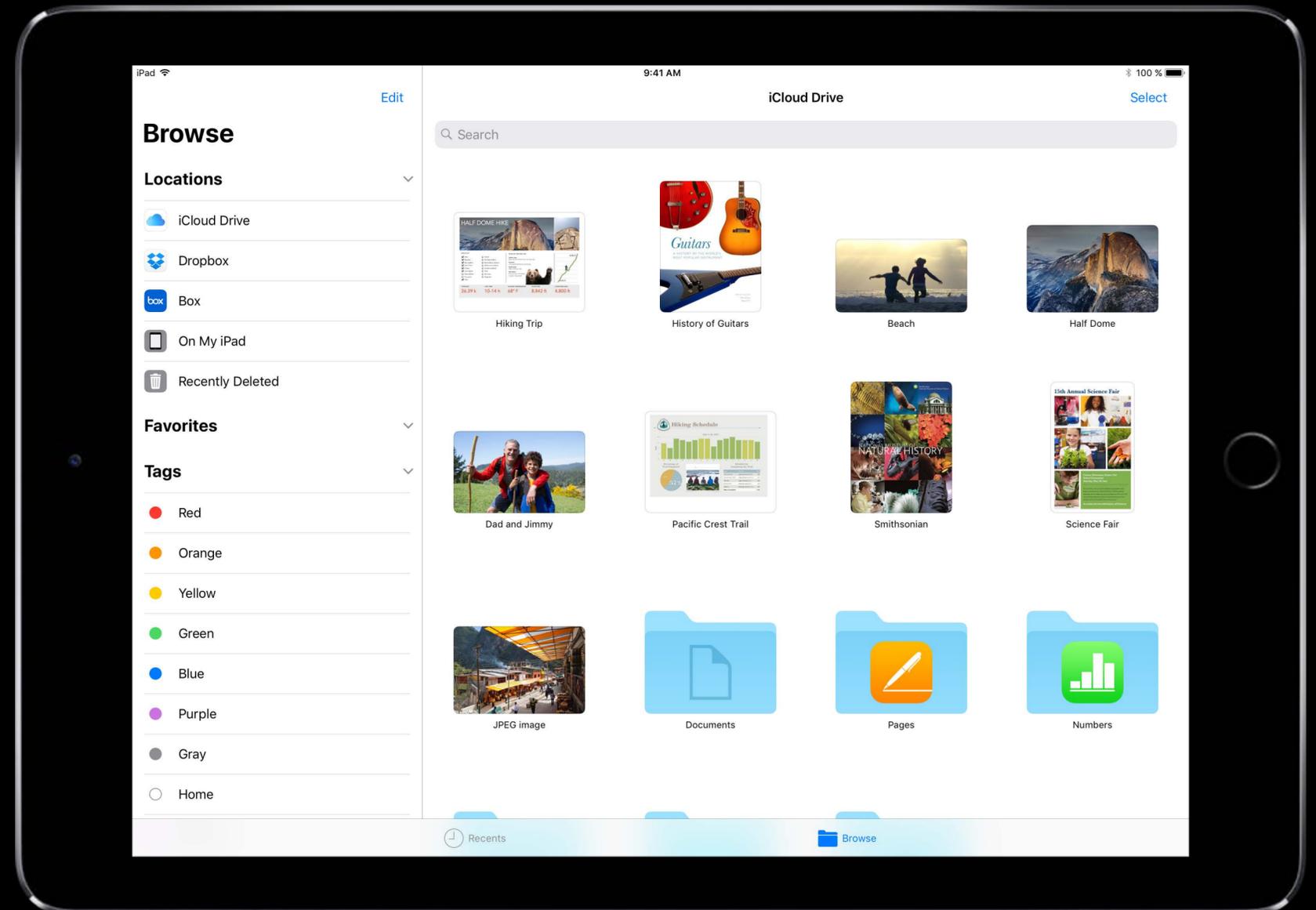
Paginating items



Enumerating Items

Paginating items

Syncing changes

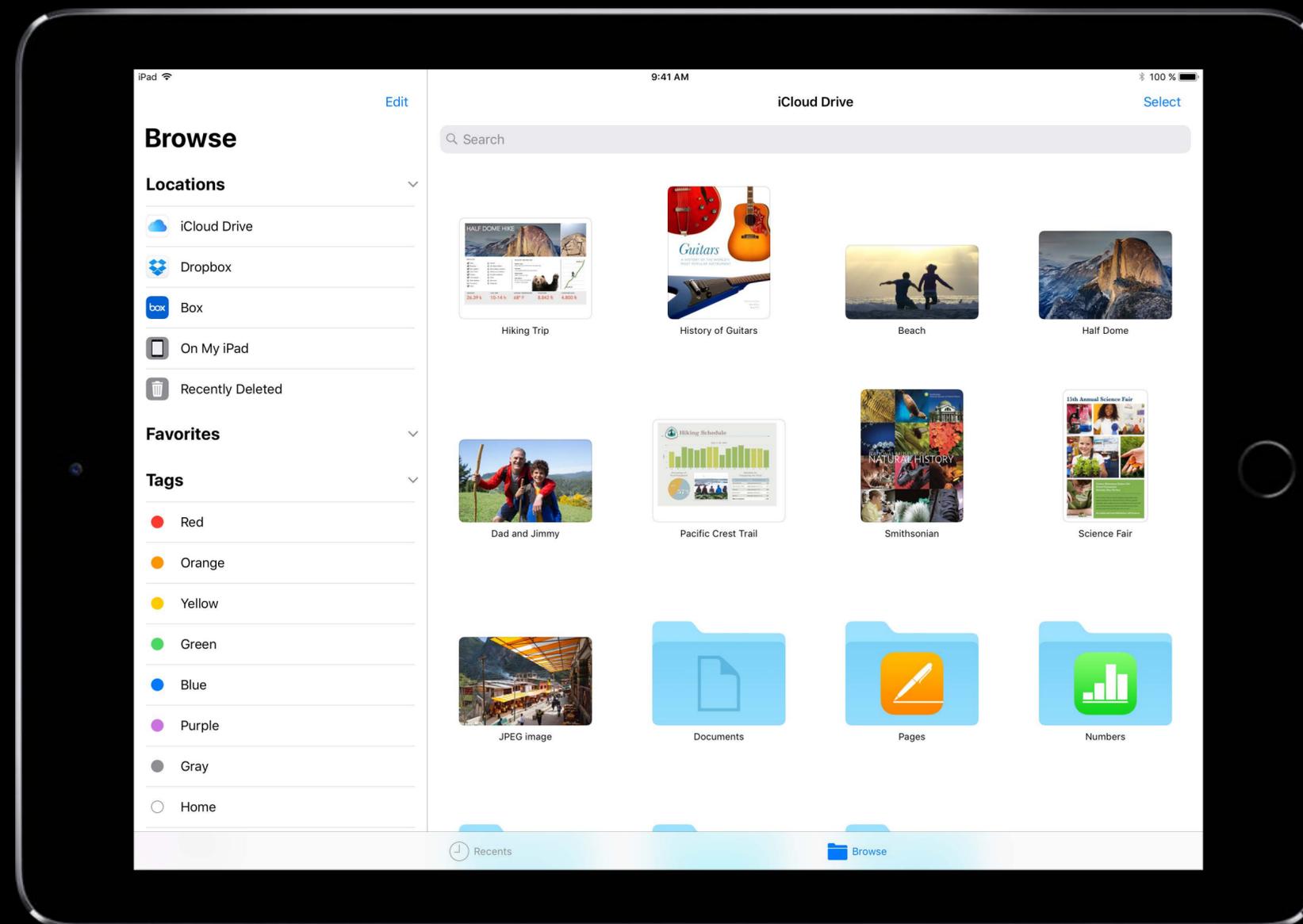


Enumerating Items

Paginating items

Syncing changes

Push notification



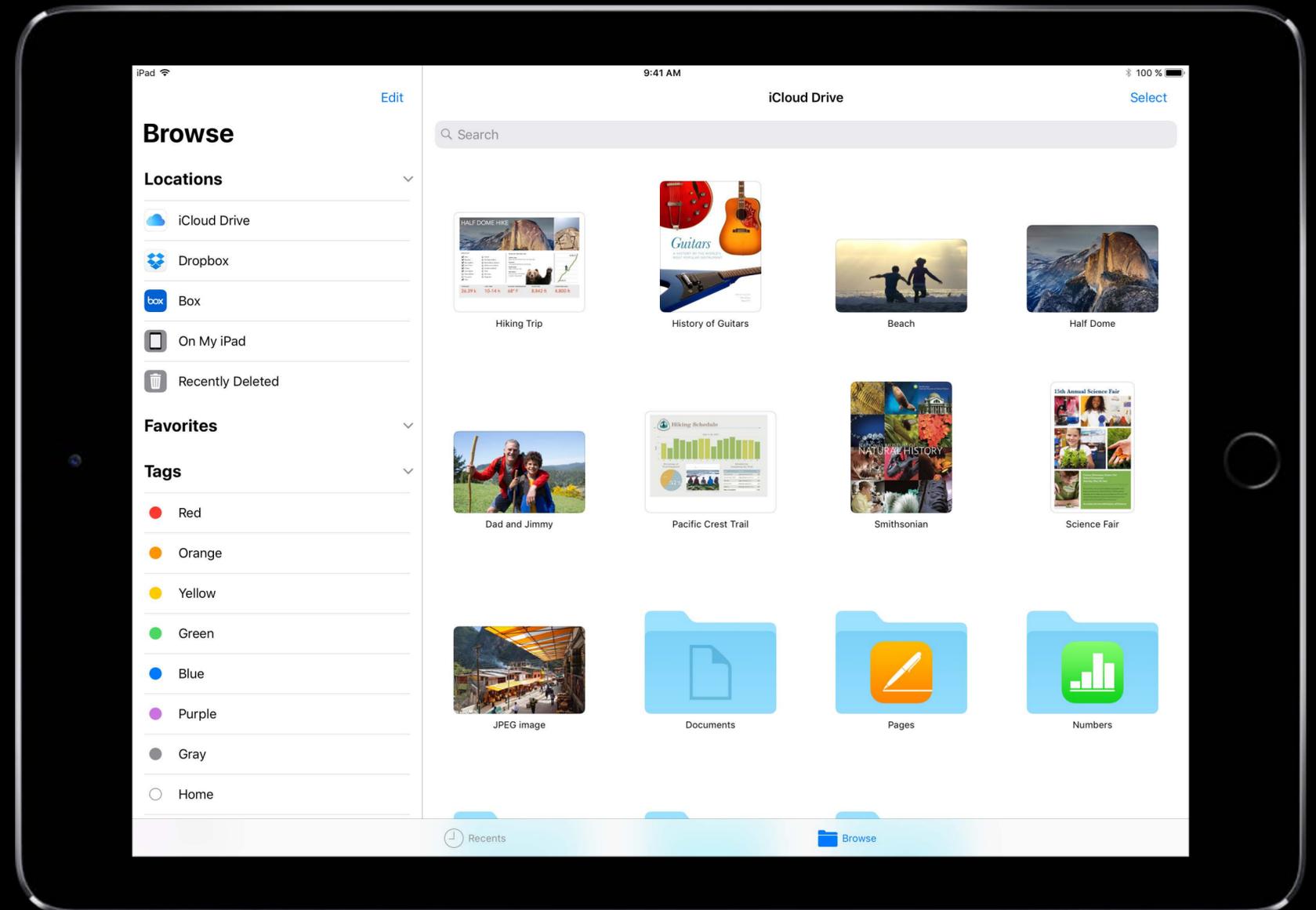
Enumerating Items

Paginating items

Syncing changes

Push notification

Signaling changes

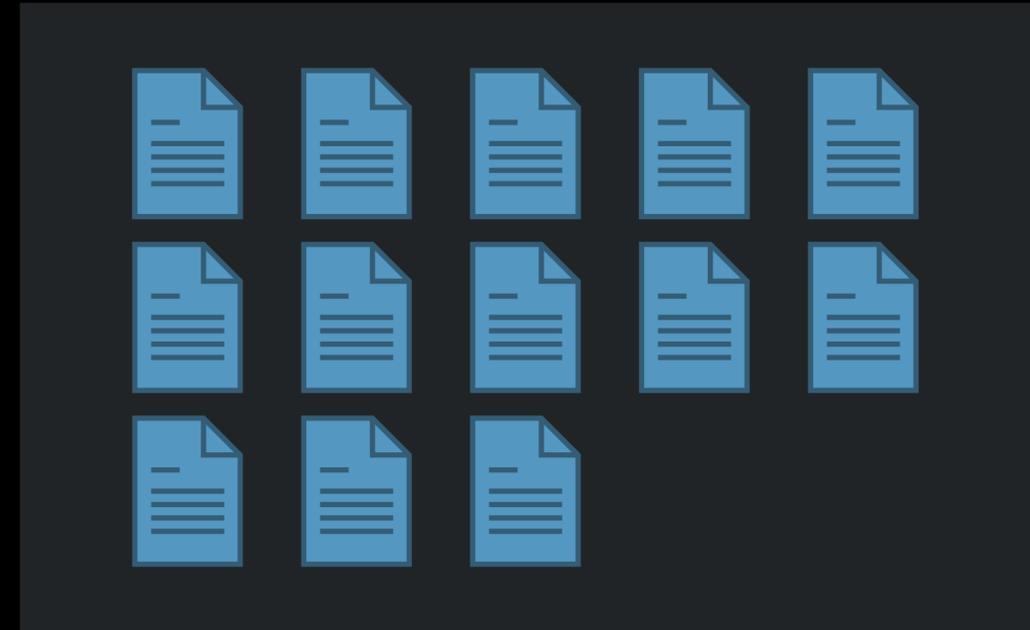


Enumerating Items

Files app



File Provider Extension



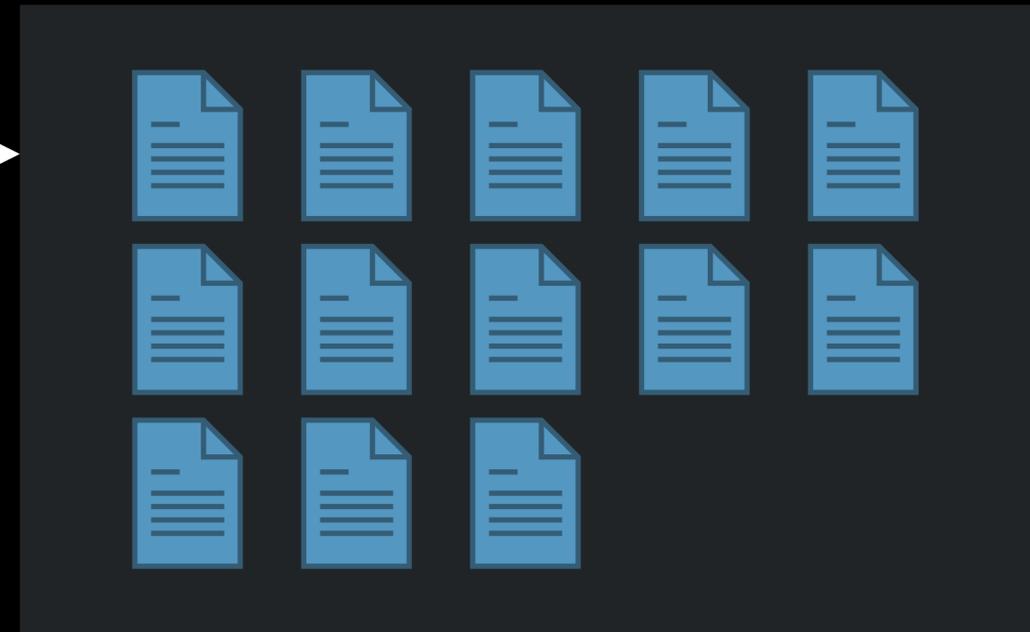
Enumerating Items

Files app

File Provider Extension

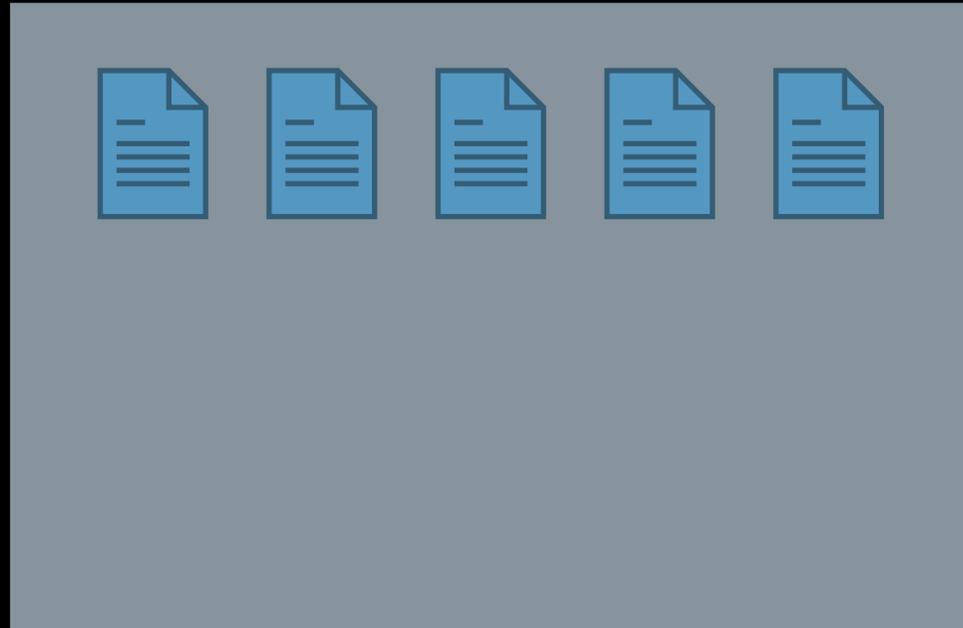


Page?

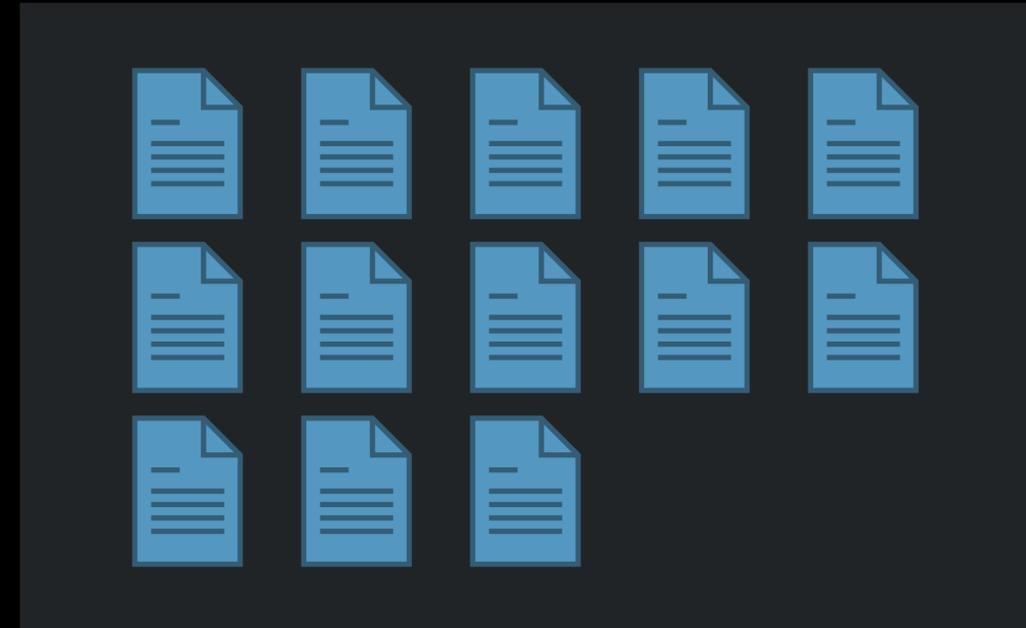


Enumerating Items

Files app

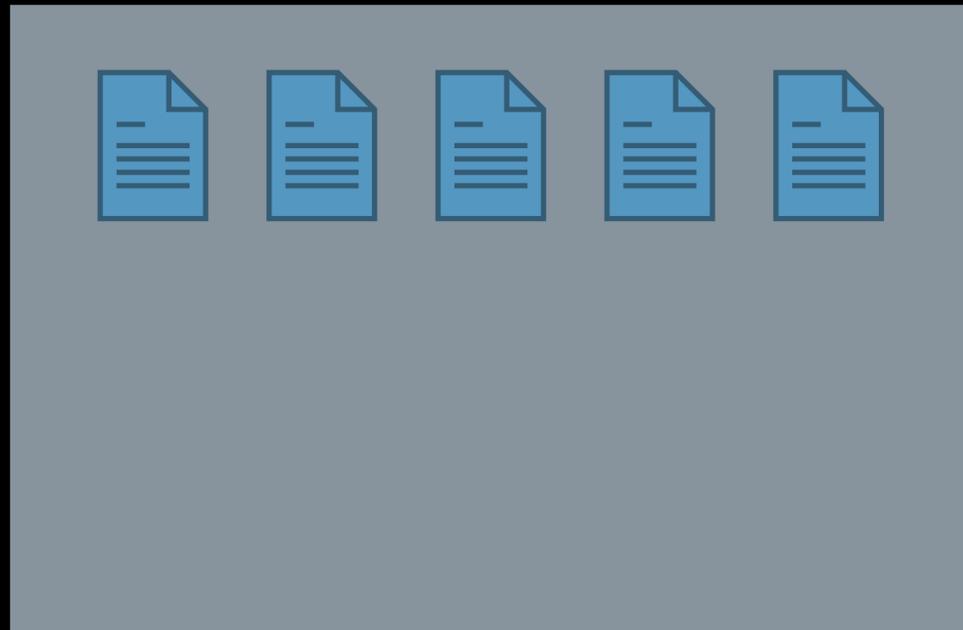


File Provider Extension

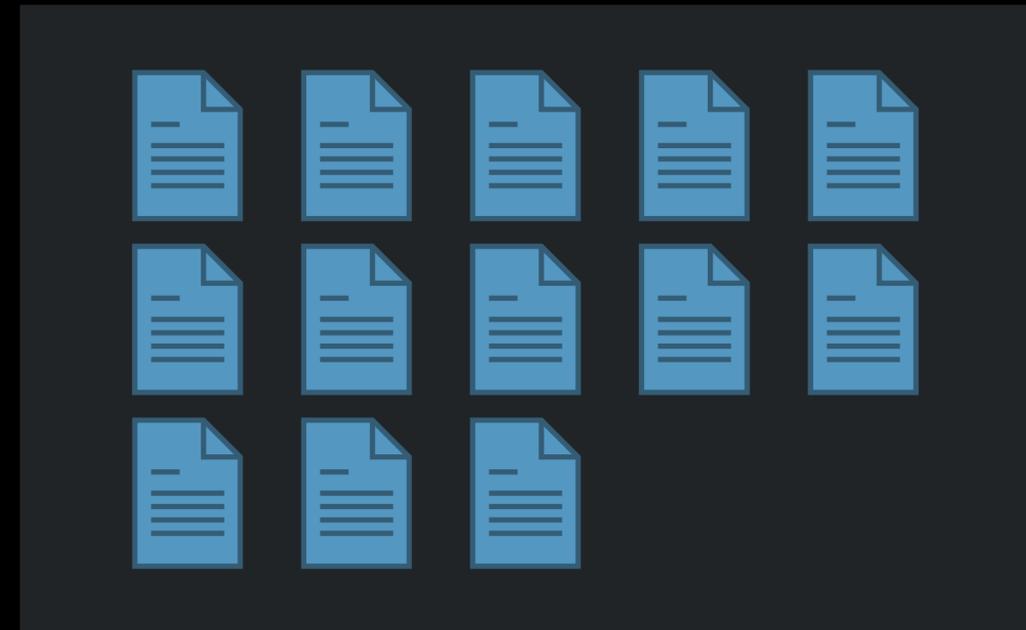


Enumerating Items

Files app

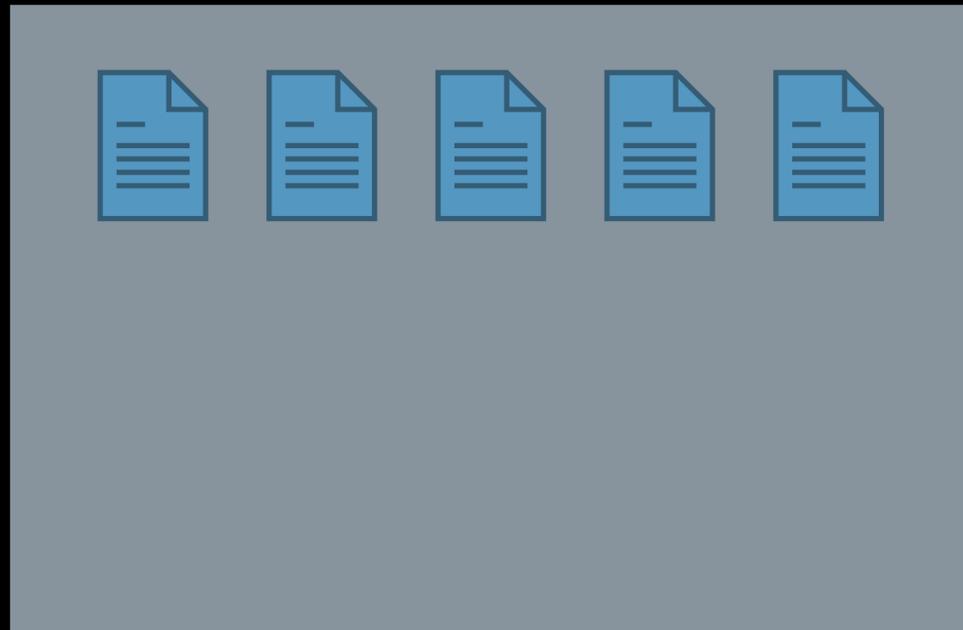


File Provider Extension

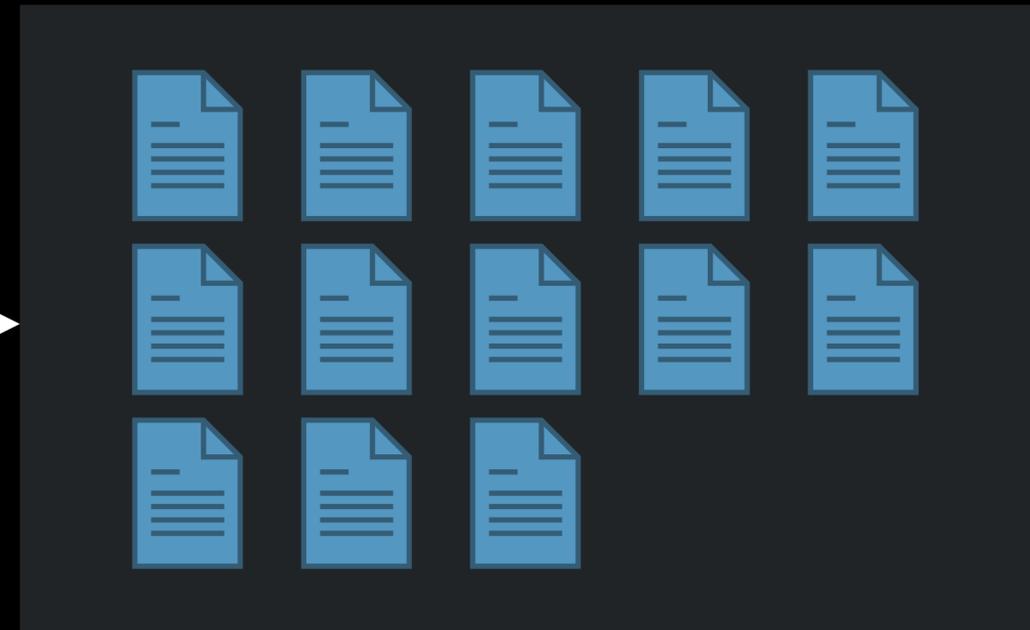


Enumerating Items

Files app



File Provider Extension

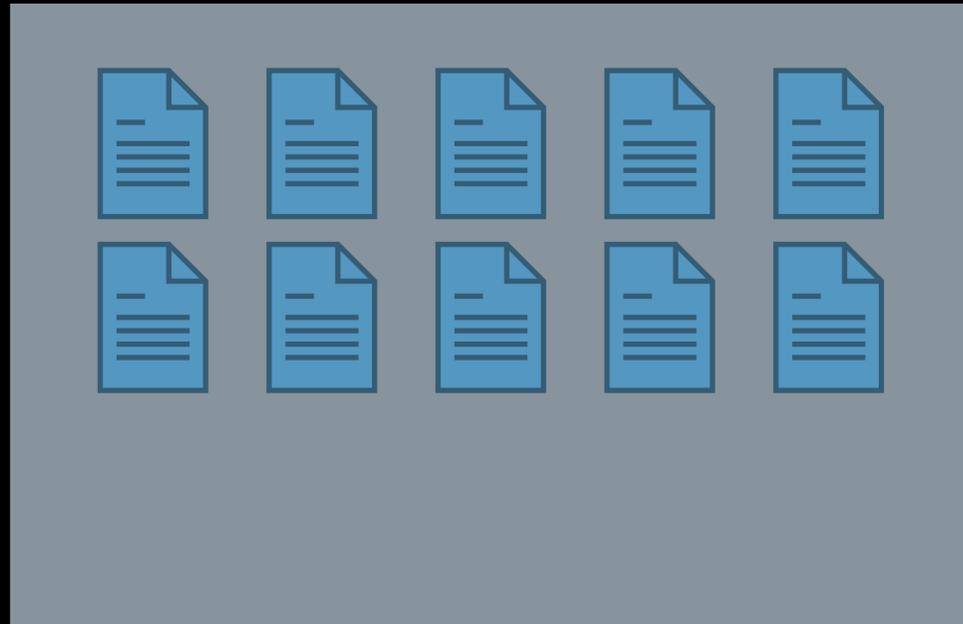


Page?

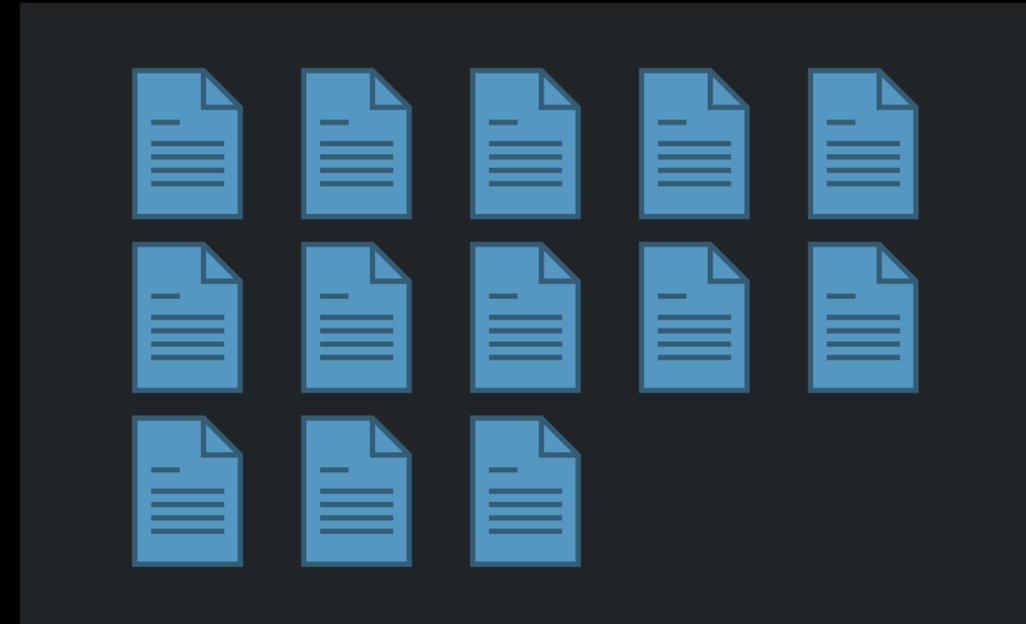


Enumerating Items

Files app

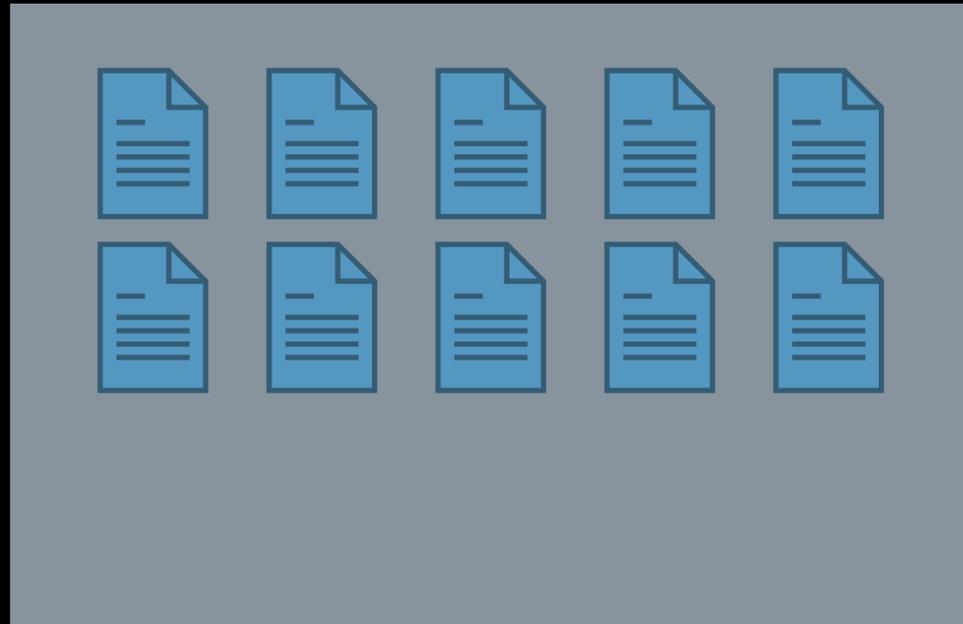


File Provider Extension

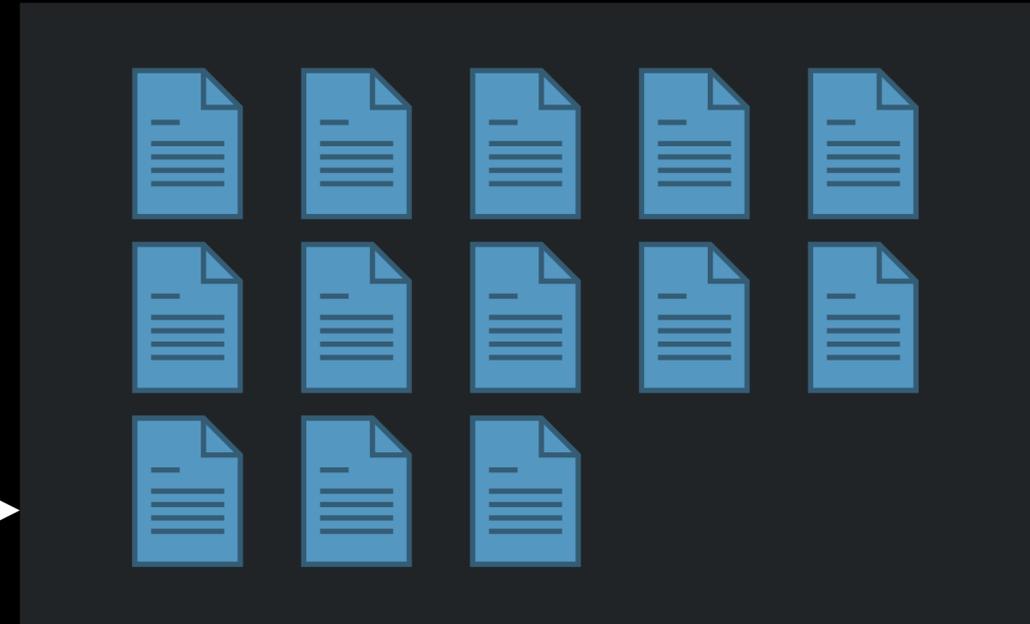


Enumerating Items

Files app



File Provider Extension

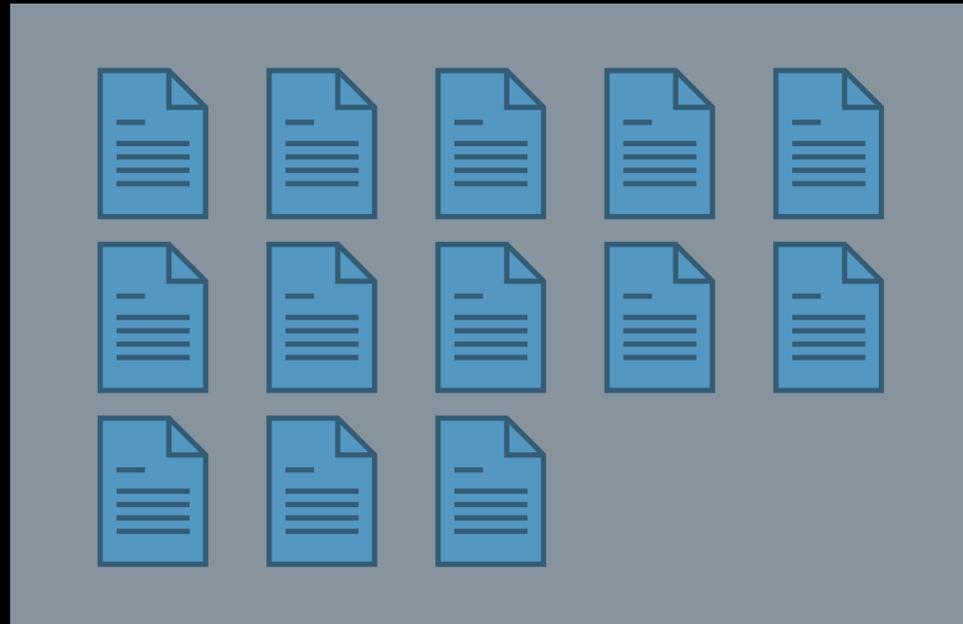


Page?

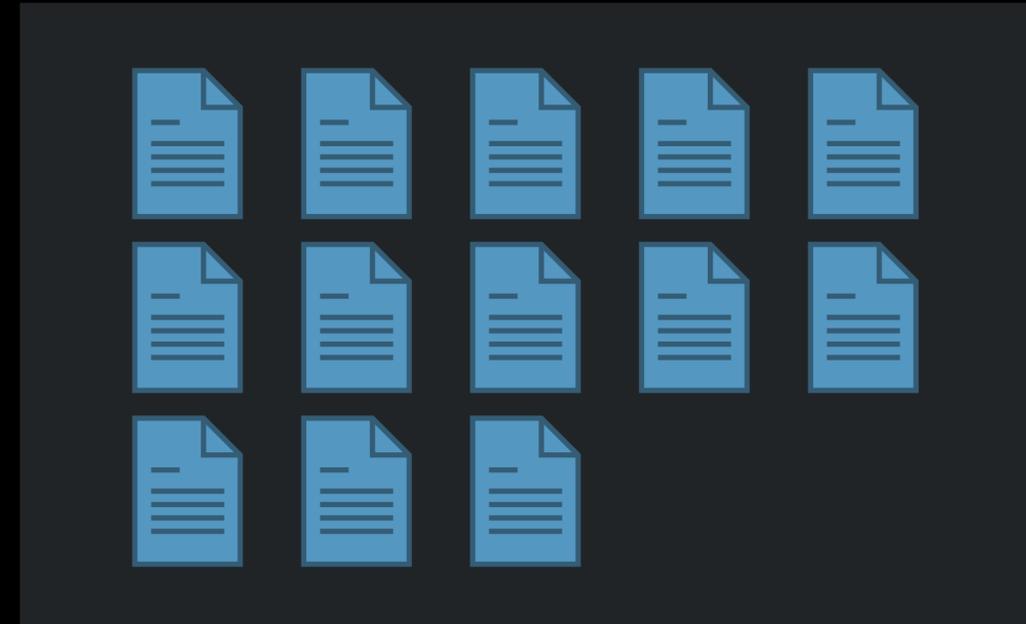


Enumerating Items

Files app



File Provider Extension



Enumerating Items

Why do we paginate?

Enumerating Items

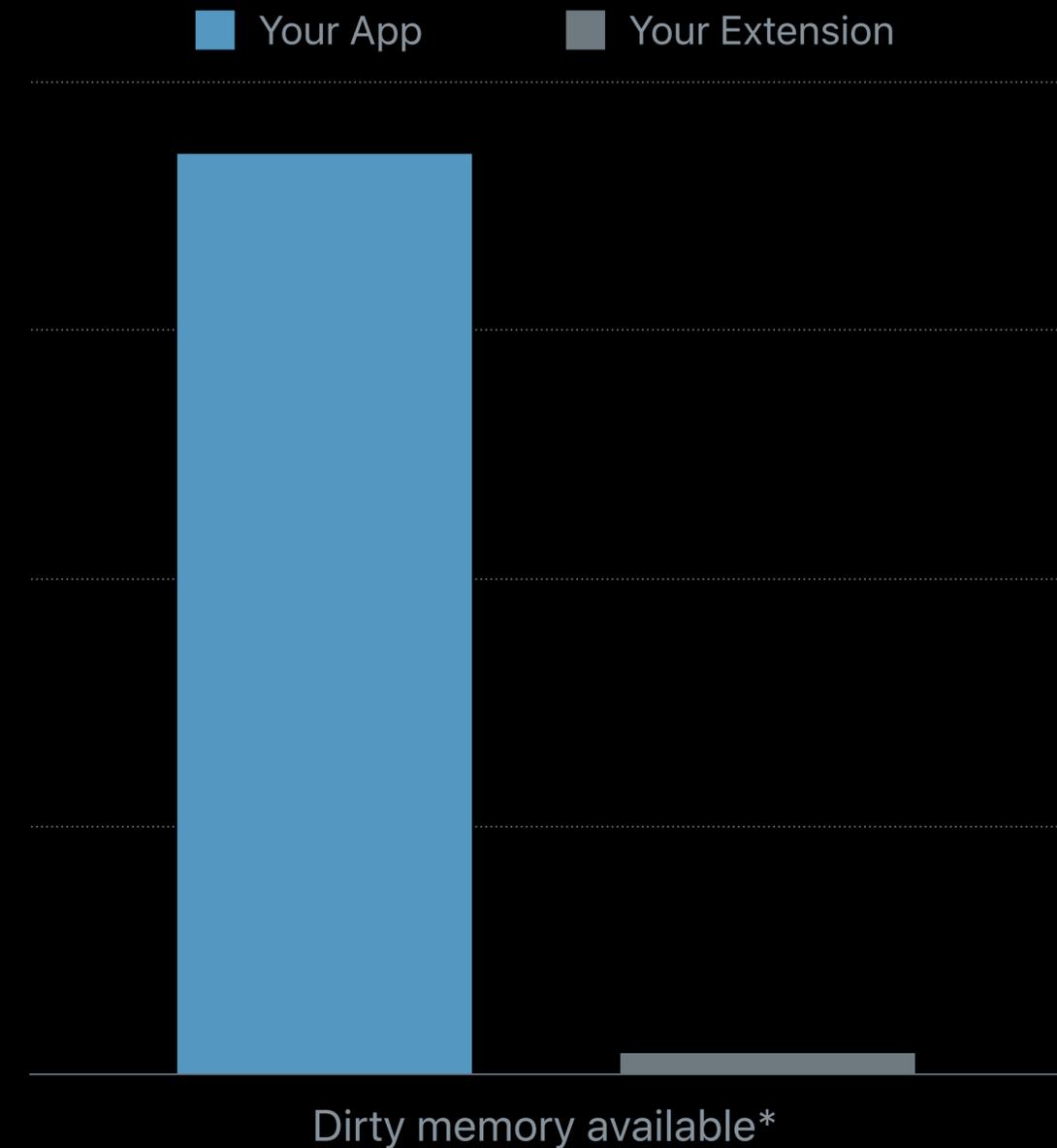
Why do we paginate?

Your extension is granted a tiny fraction of the memory your app is granted

Enumerating Items

Why do we paginate?

Your extension is granted a tiny fraction of the memory your app is granted



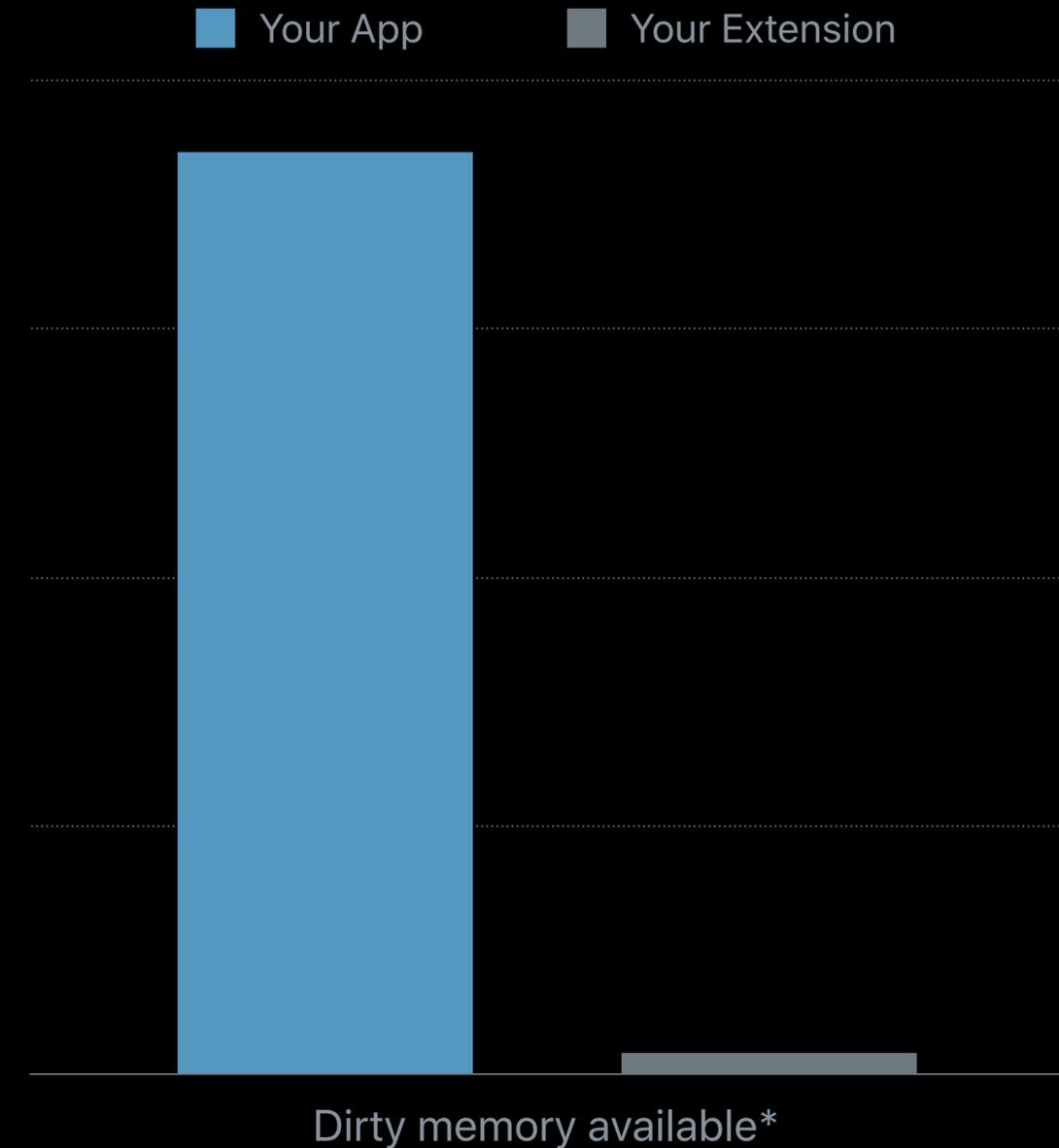
* Exact number varies between devices

Enumerating Items

Why do we paginate?

Your extension is granted a tiny fraction of the memory your app is granted

Above this limit your extension is terminated



* Exact number varies between devices

Enumerating Items

Reducing memory usage

Enumerating Items

Reducing memory usage

Use `NSURLSessionDownloadTask` and `NSURLSessionUploadTask` rather than `NSURLSessionDataTask`

Enumerating Items

Reducing memory usage

Use `NSURLSessionDownloadTask` and `NSURLSessionUploadTask` rather than `NSURLSessionDataTask`

Use `DispatchQueue(autoreleaseFrequency: .always)`

Enumerating Items

Reducing memory usage

Use `NSURLSessionDownloadTask` and `NSURLSessionUploadTask` rather than `NSURLSessionDataTask`

Use `DispatchQueue(autoreleaseFrequency: .always)`

Use `« while () @autoreleasepool {} »`

Enumerating Items

Files app

File Provider Extension

Enumerating Items

Files app

Open a Folder

File Provider Extension

Enumerating Items

Files app

Open a Folder

File Provider Extension

```
func enumerator(forContainer  
itemIdentifier:)
```



Enumerating Items

Files app

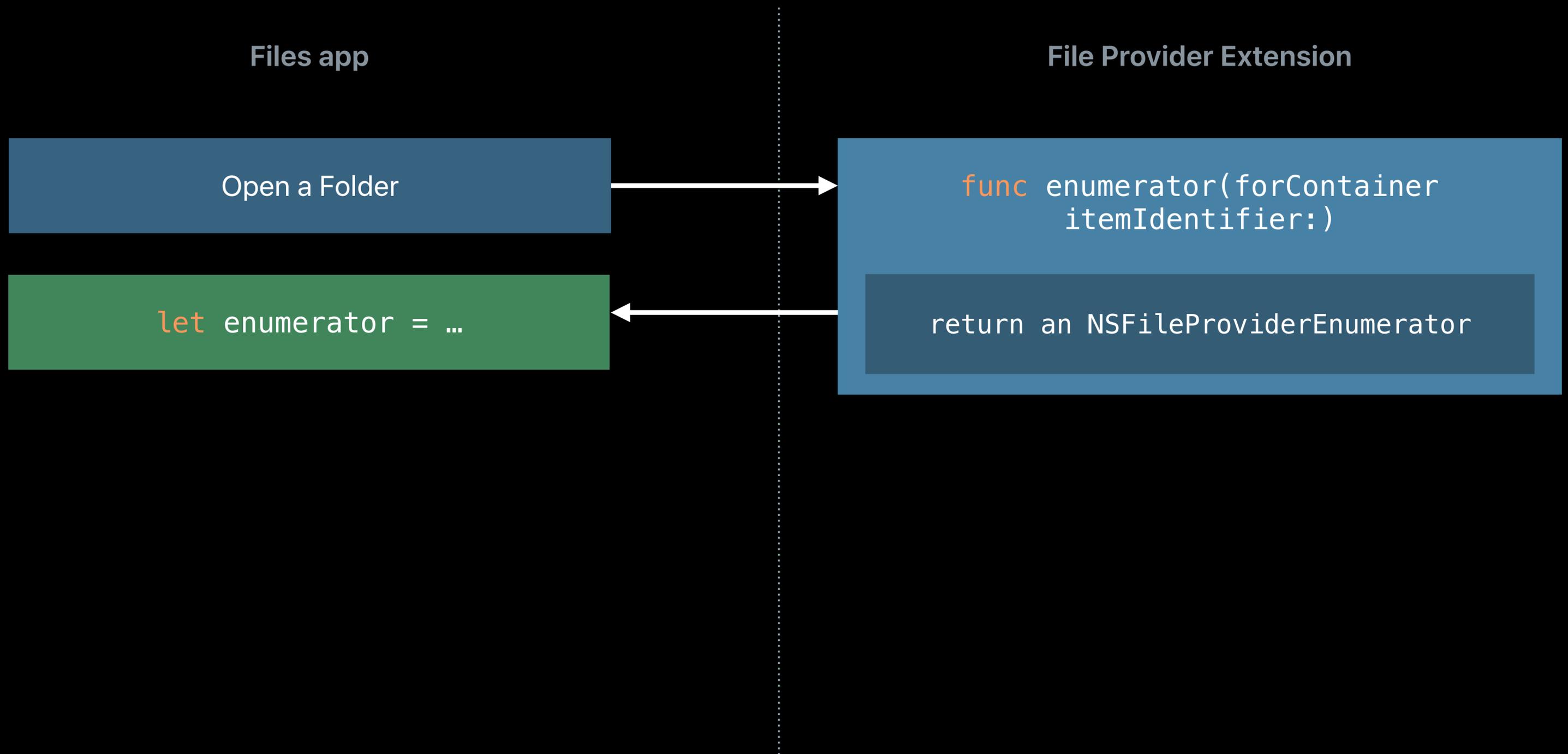
Open a Folder

File Provider Extension

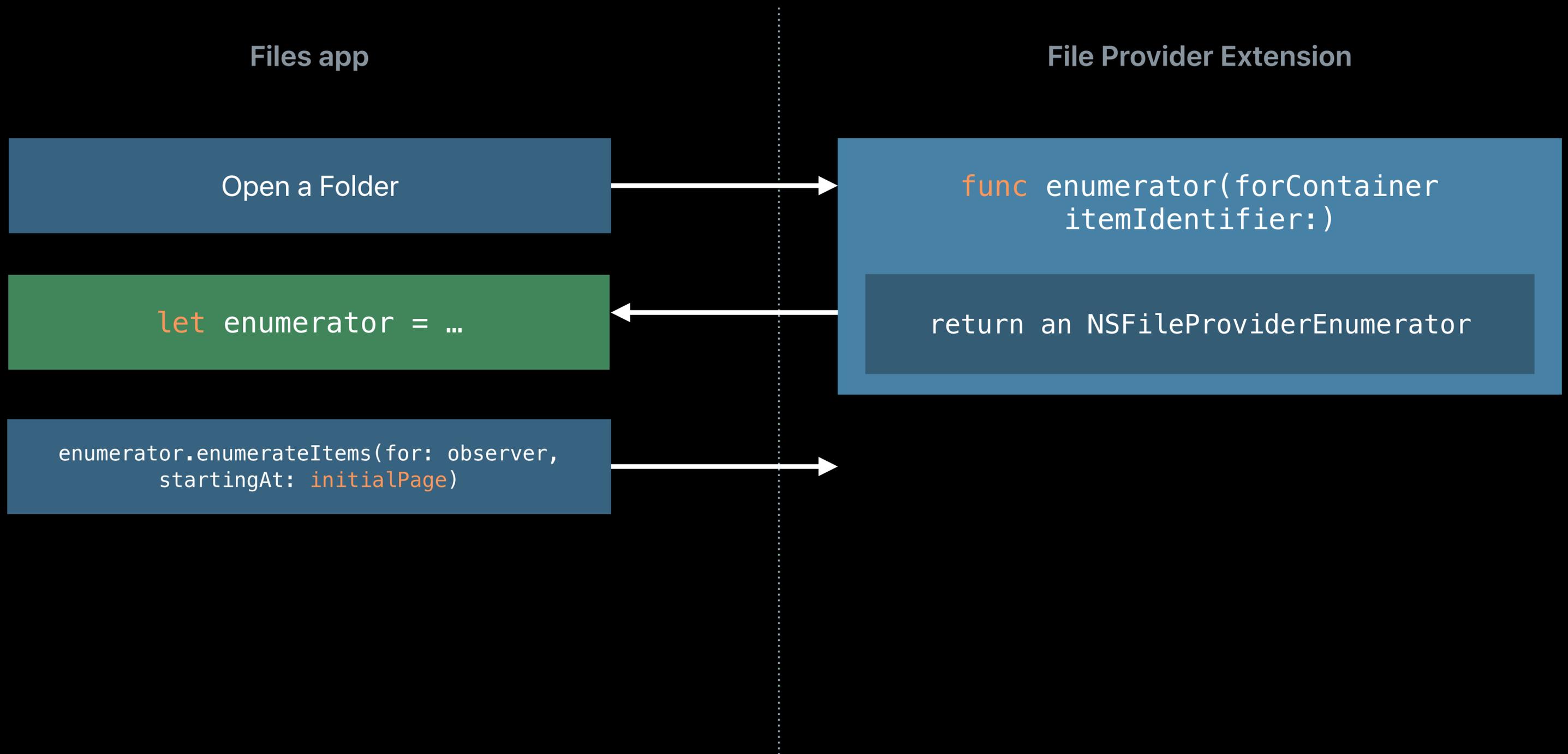
```
func enumerator(forContainer  
itemIdentifier:)
```

```
return an NSFileProviderEnumerator
```

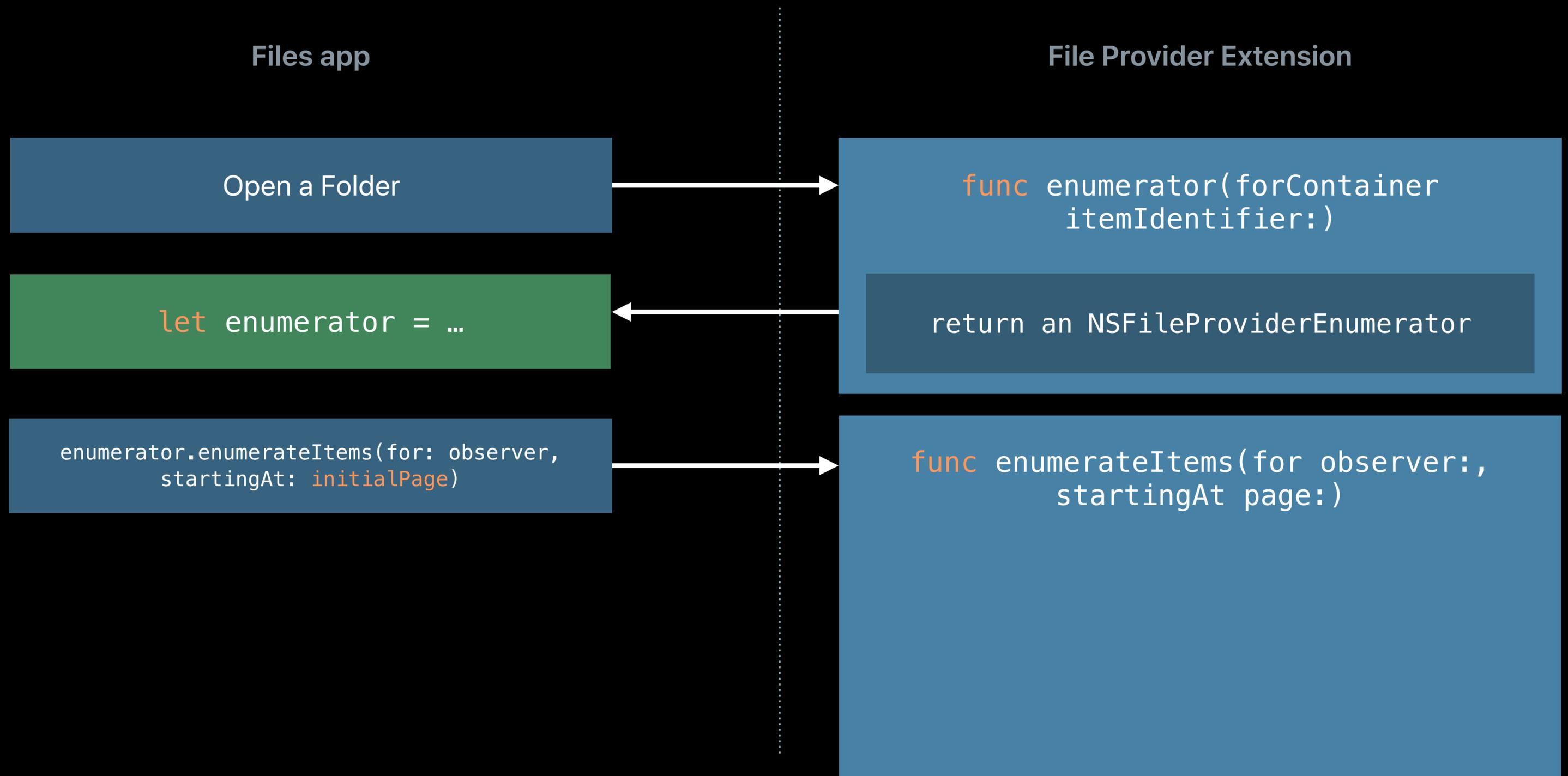
Enumerating Items



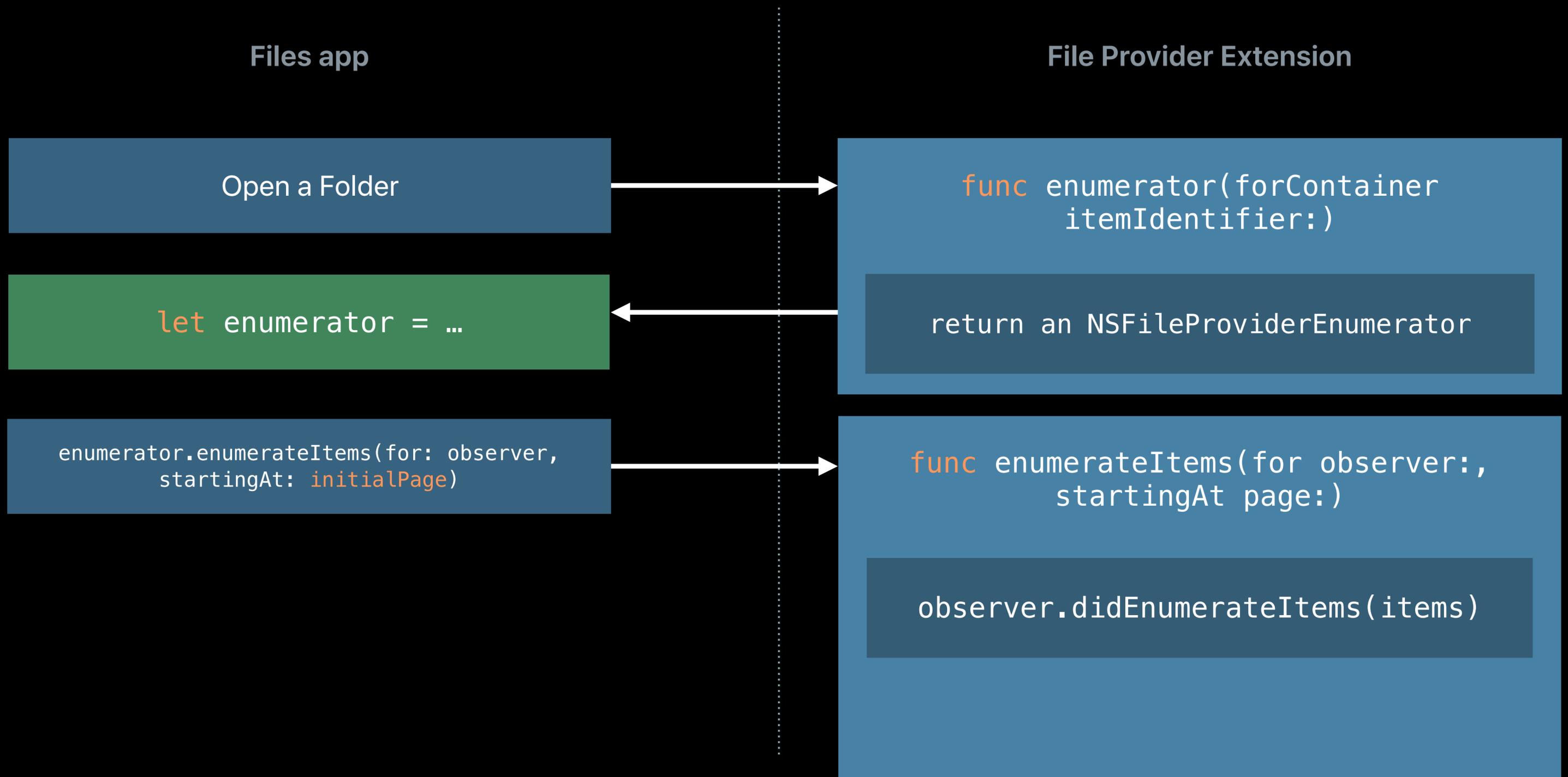
Enumerating Items



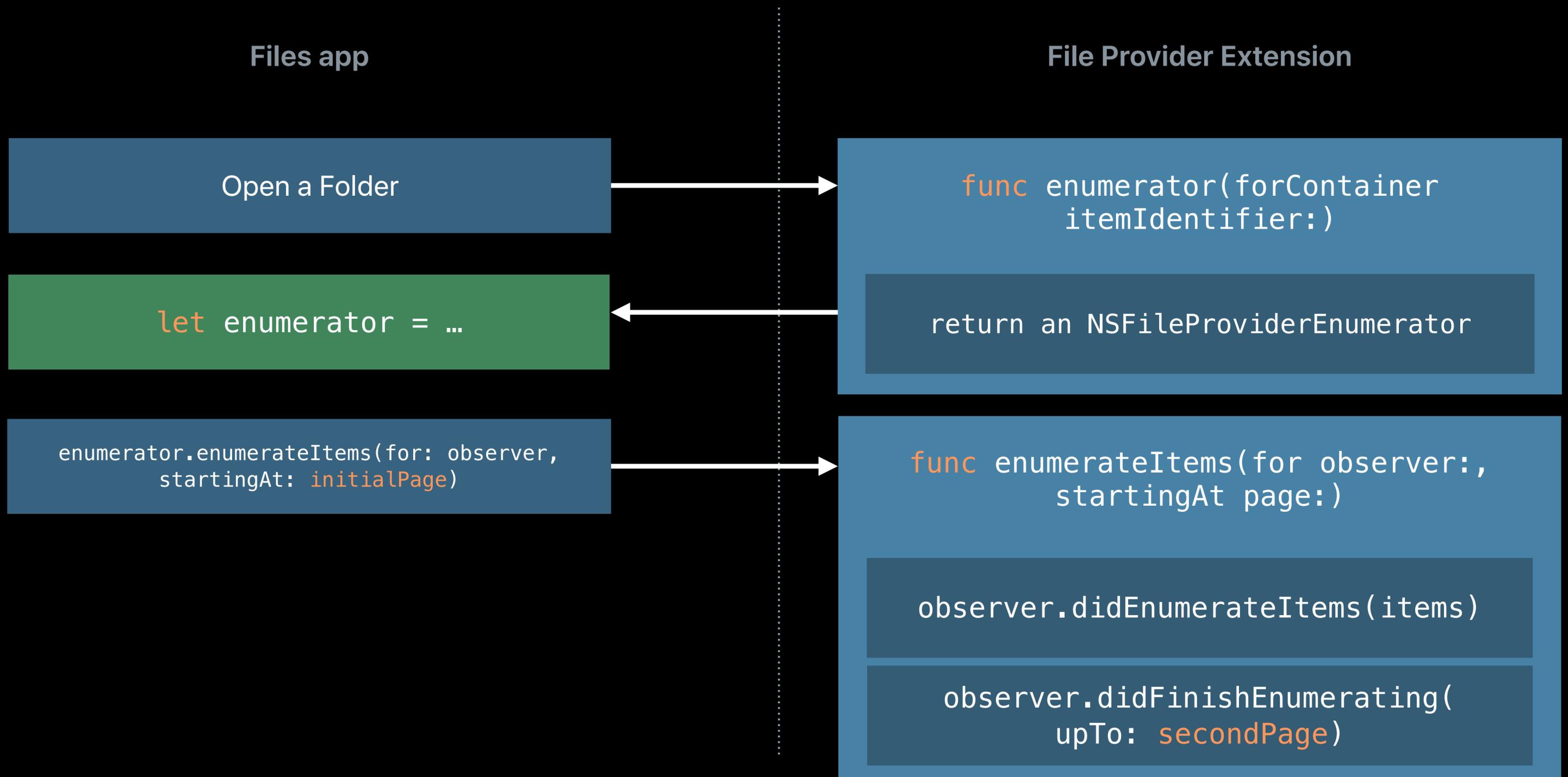
Enumerating Items



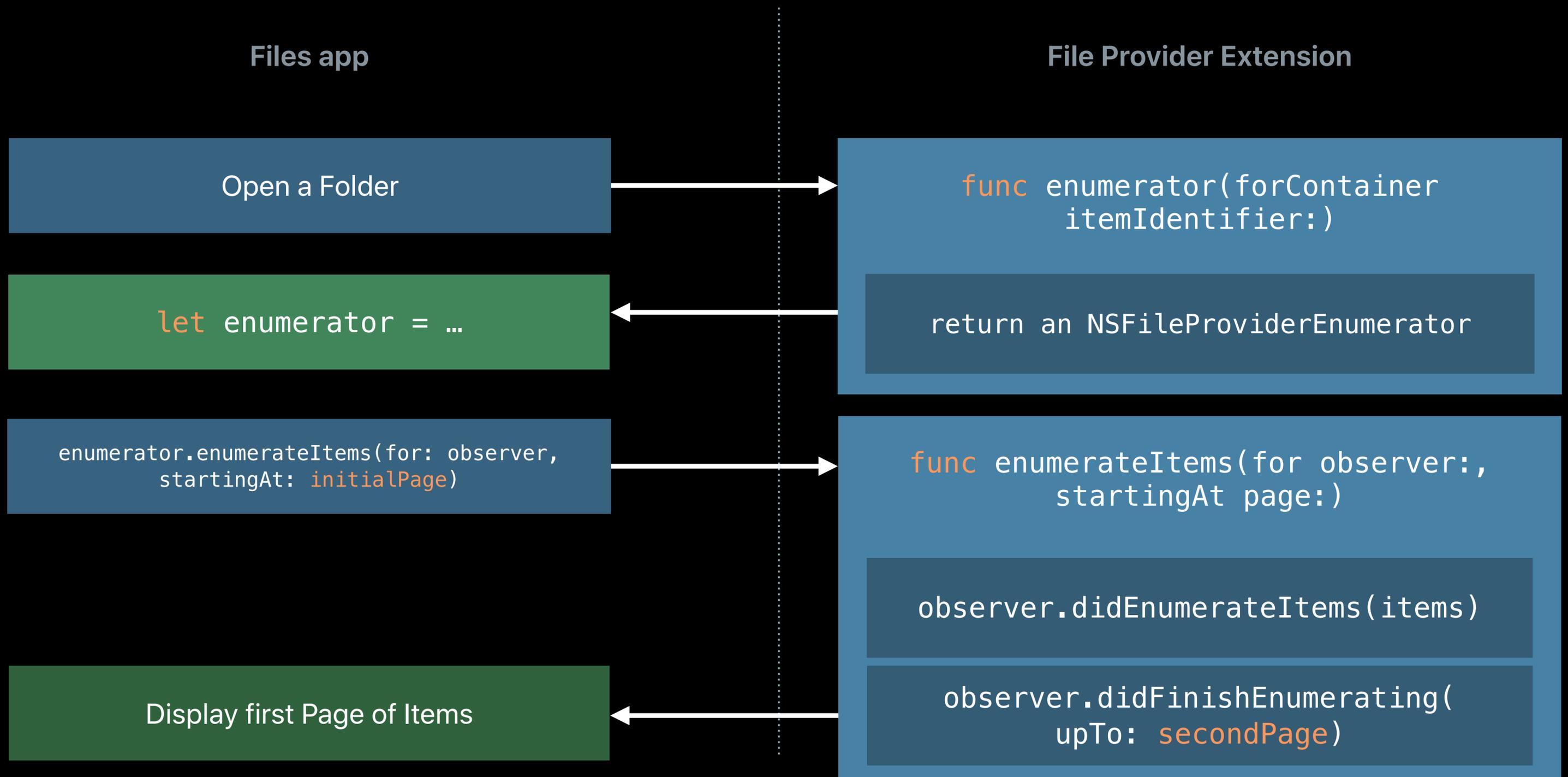
Enumerating Items



Enumerating Items



Enumerating Items



Enumerating Items

Returning an enumerator

Enumerating Items

Returning an enumerator

```
func enumerator(forContainer itemIdentifier: NSFileProviderItemIdentifier)  
    -> NSFileProviderEnumerator
```

Enumerating Items

Returning an enumerator

```
func enumerator(forContainer itemIdentifier: NSFileProviderItemIdentifier)  
    -> NSFileProviderEnumerator
```

`itemIdentifier` can be one of:

```
.workingSet
```

Enumerating Items

Returning an enumerator

```
func enumerator(forContainer itemIdentifier: NSFileProviderItemIdentifier)  
    -> NSFileProviderEnumerator
```

`itemIdentifier` can be one of:

```
.workingSet
```

```
.rootContainer
```

Enumerating Items

Returning an enumerator

```
func enumerator(forContainer itemIdentifier: NSFileProviderItemIdentifier)  
    -> NSFileProviderEnumerator
```

`itemIdentifier` can be one of:

`.workingSet`

`.rootContainer`

- any of your folder item identifiers

Enumerating Items

NSFileProviderEnumerator

Enumerating Items

NSFileProviderEnumerator

```
func enumerateItems(for observer: NSFileProviderObserver,  
                   startingAt page: NSFileProviderPage)
```

observer

Enumerating Items

NSFileProviderEnumerator

```
func enumerateItems(for observer: NSFileProviderObserver,  
                   startingAt page: NSFileProviderPage)
```

observer

System object receiving your items and the next page

Enumerating Items

NSFileProviderEnumerator

```
func enumerateItems(for observer: NSFileProviderObserver,  
                   startingAt page: NSFileProviderPage)
```

observer

System object receiving your items and the next page

page

Enumerating Items

NSFileProviderEnumerator

```
func enumerateItems(for observer: NSFileProviderObserver,  
                   startingAt page: NSFileProviderPage)
```

observer

System object receiving your items and the next page

page

Data blob your extension previously returned (max 500 bytes)

Enumerating Items

NSFileProviderEnumerator

```
func enumerateItems(for observer: NSFileProviderObserver,  
                   startingAt page: NSFileProviderPage)
```

observer

System object receiving your items and the next page

page

Data blob your extension previously returned (max 500 bytes)

or

Enumerating Items

NSFileProviderEnumerator

```
func enumerateItems(for observer: NSFileProviderObserver,  
                   startingAt page: NSFileProviderPage)
```

observer

System object receiving your items and the next page

page

Data blob your extension previously returned (max 500 bytes)

or

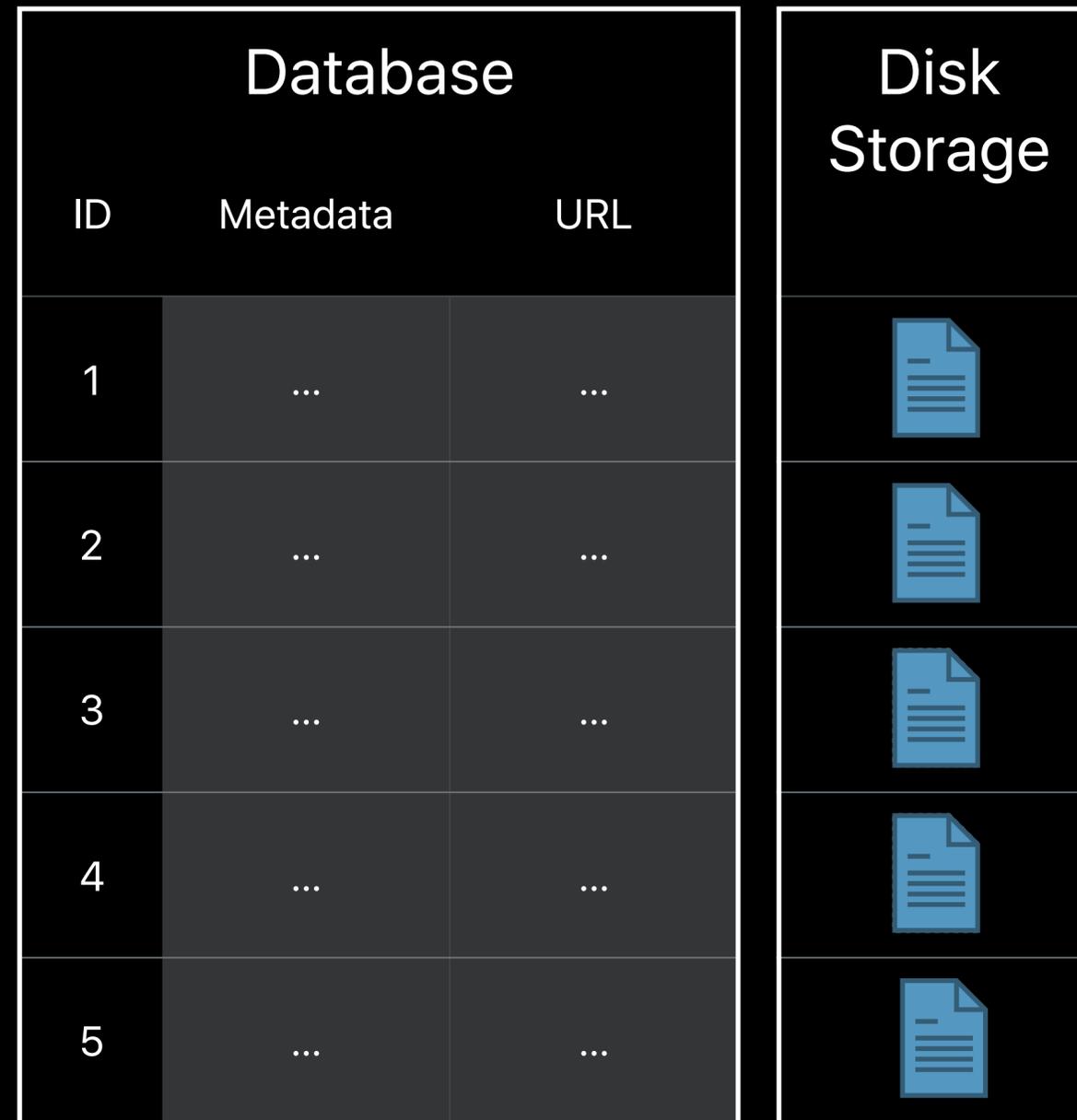
```
NSFileProviderInitialPageSortedByDate, NSFileProviderInitialPageSortedByName
```

Anatomy of a File Provider Extension

Item ID → metadata

Item ID ↔ file URL

Manage the disk storage



Anatomy of a File Provider Extension

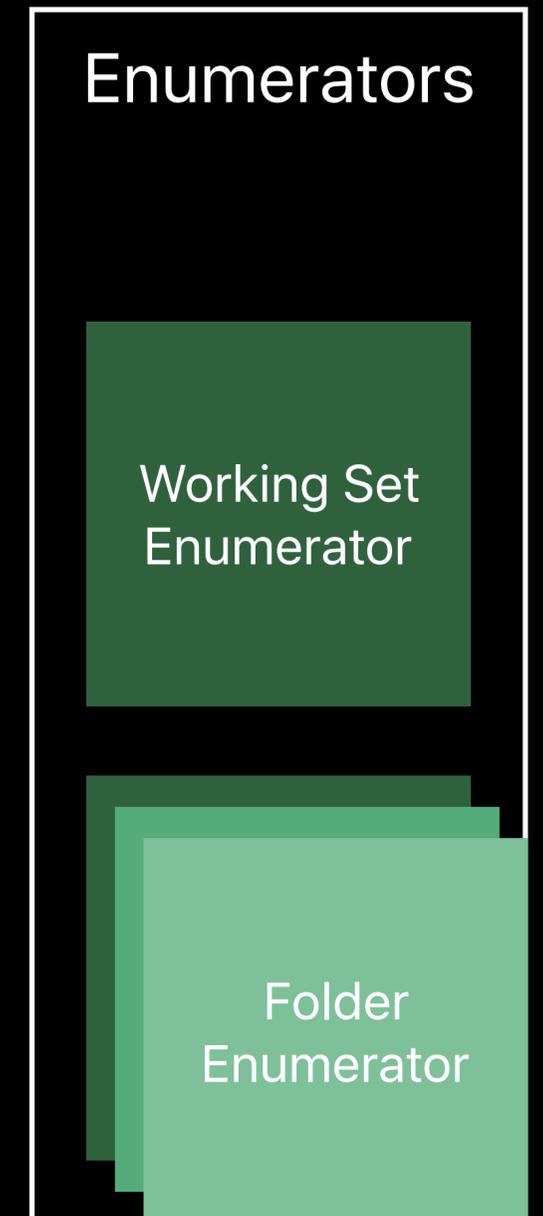
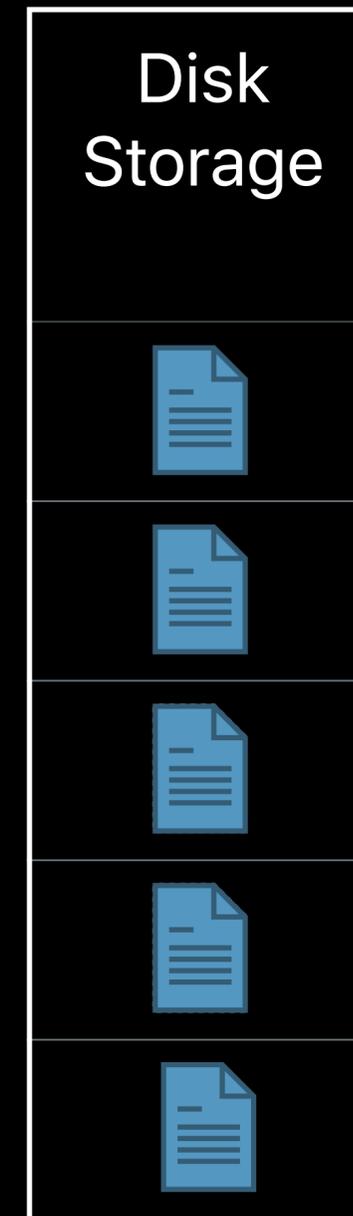
Item ID → metadata

Item ID ↔ file URL

Manage the disk storage

Enumerate folders and working set

Database		
ID	Metadata	URL
1
2
3
4
5



Enumerating Items

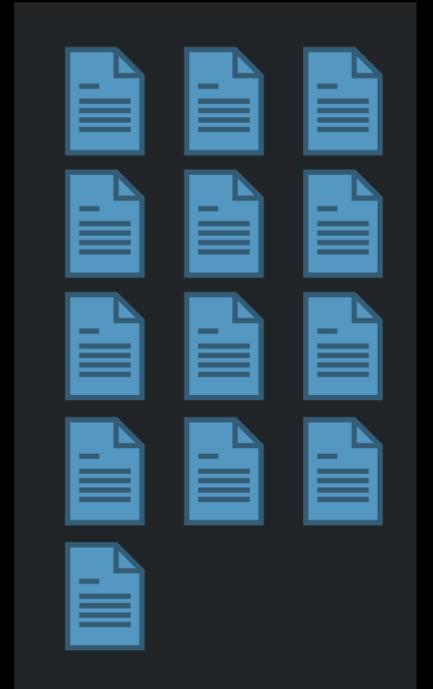
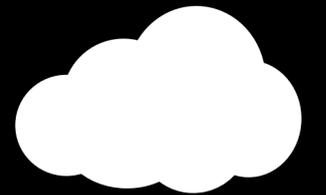
Folder enumerator

Files app



File Provider
Extension

Cloud



Enumerating Items

Folder enumerator

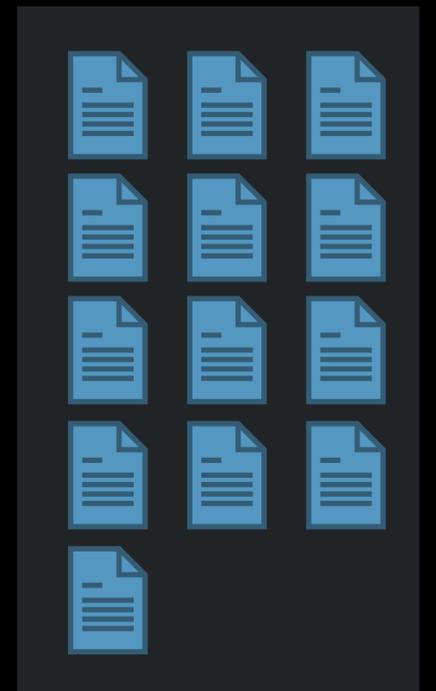
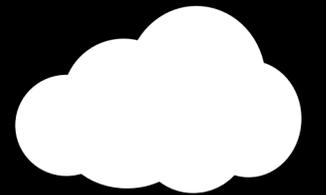
Enumerate the cloud directly

Files app



File Provider
Extension

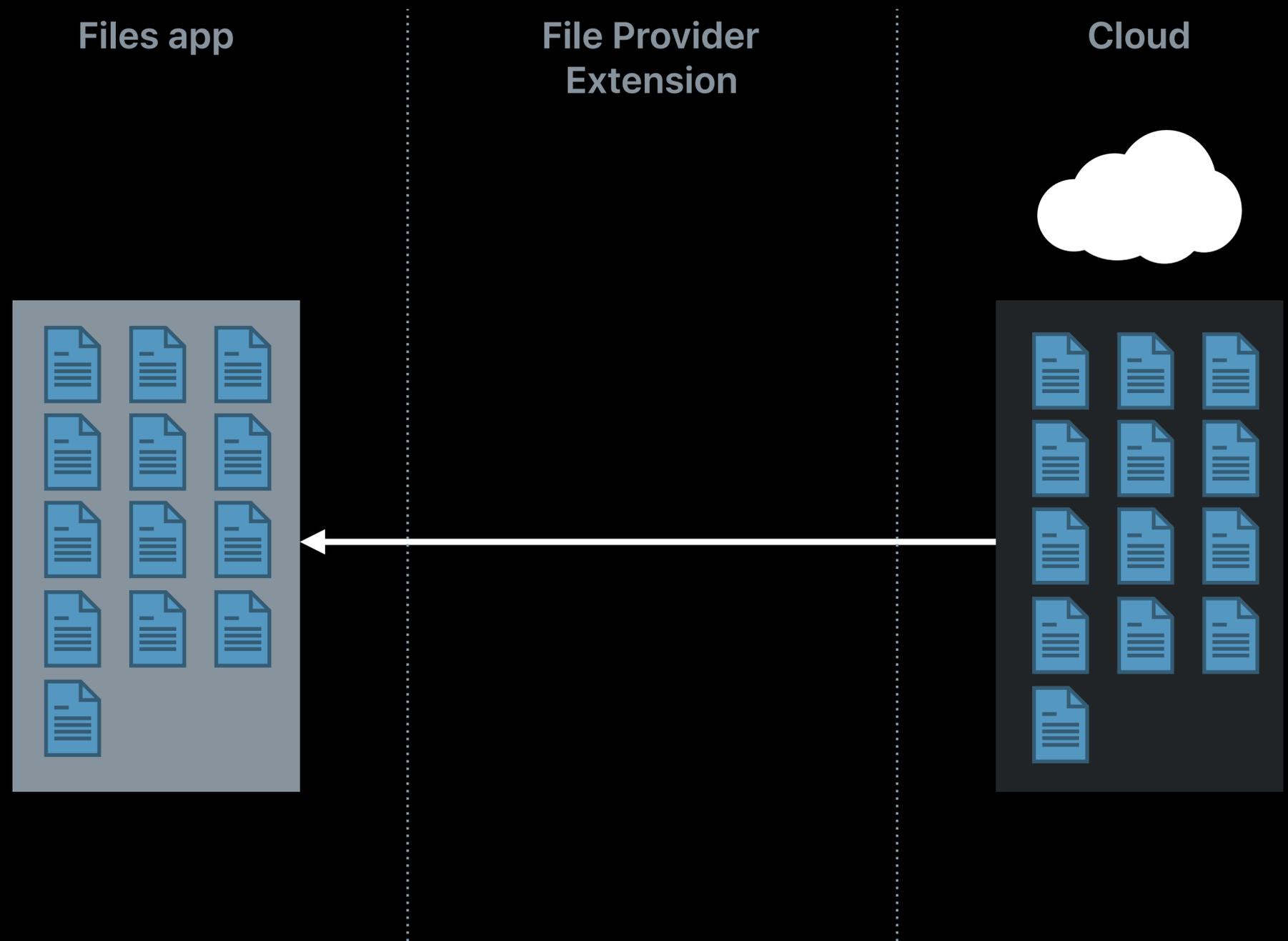
Cloud



Enumerating Items

Folder enumerator

Enumerate the cloud directly

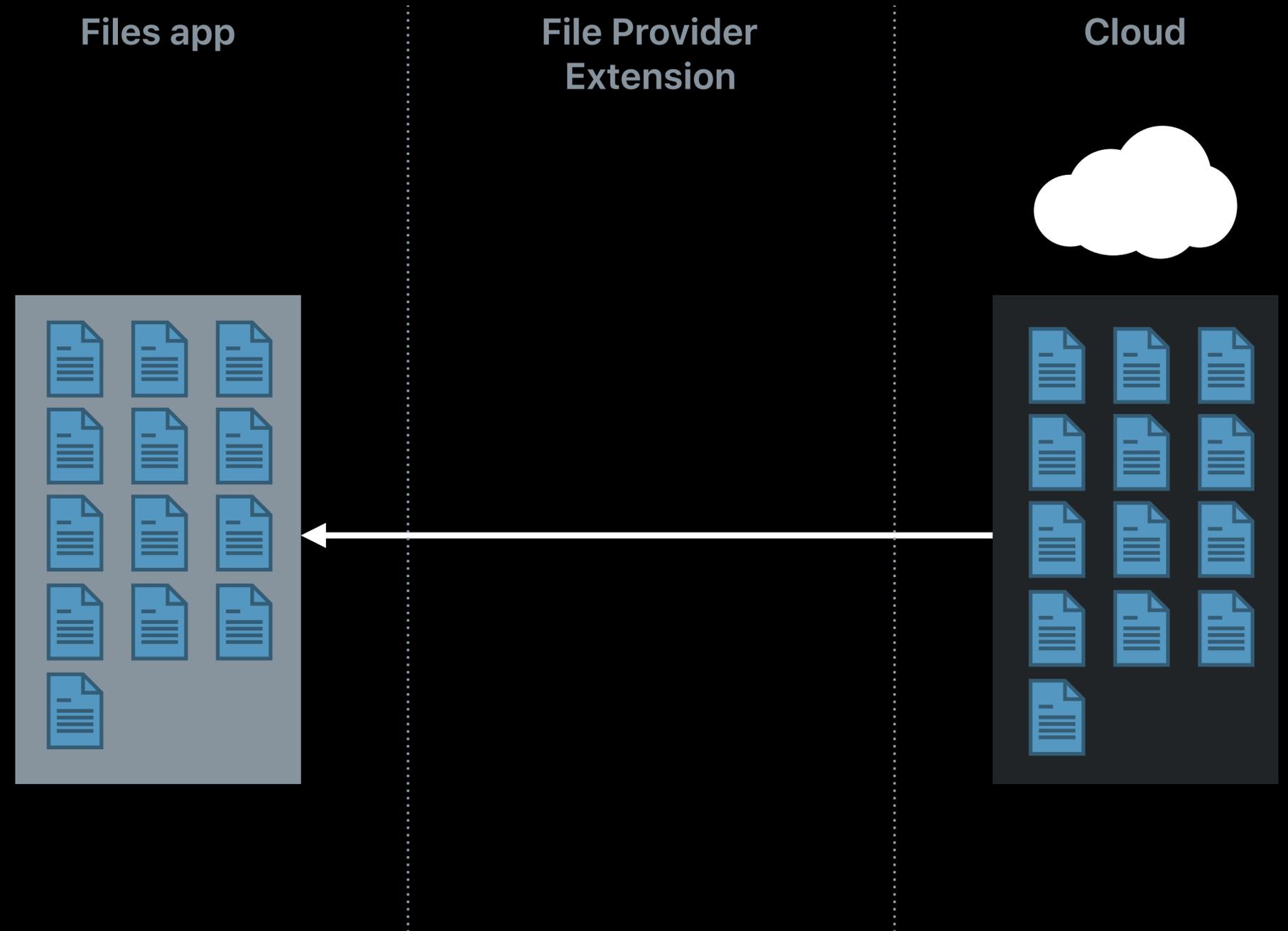


Enumerating Items

Folder enumerator

Enumerate the cloud directly

Caching policy is up to you



Enumerating Items

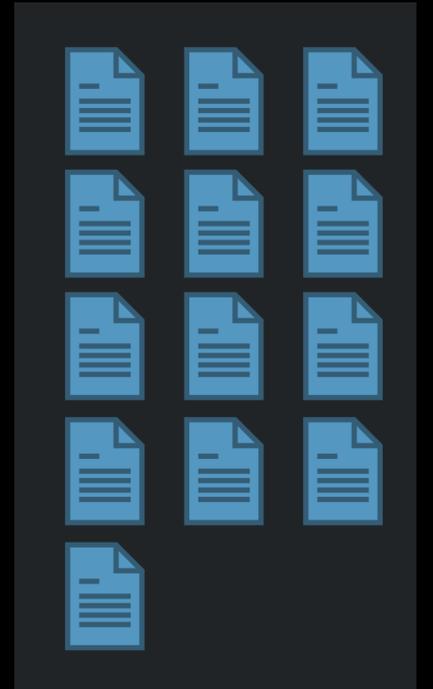
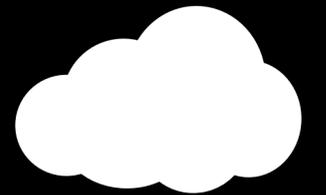
Working set enumerator

Files Provider
Daemon



File Provider
Extension

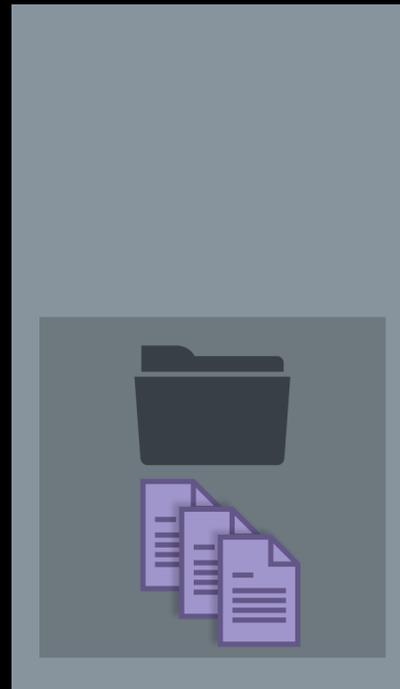
Cloud



Enumerating Items

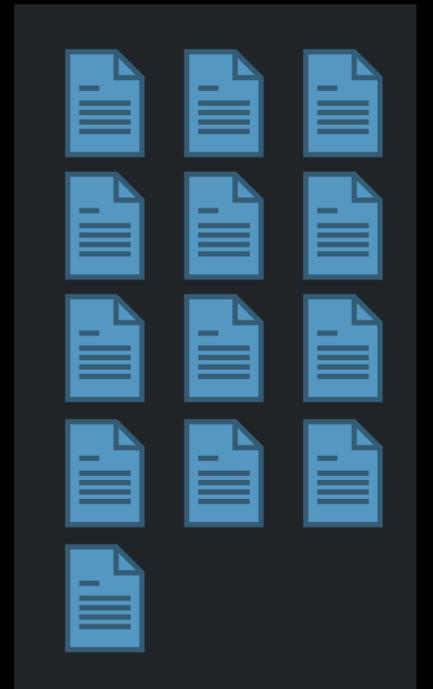
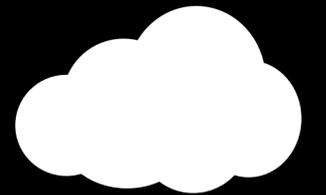
Working set enumerator

Files Provider
Daemon



File Provider
Extension

Cloud

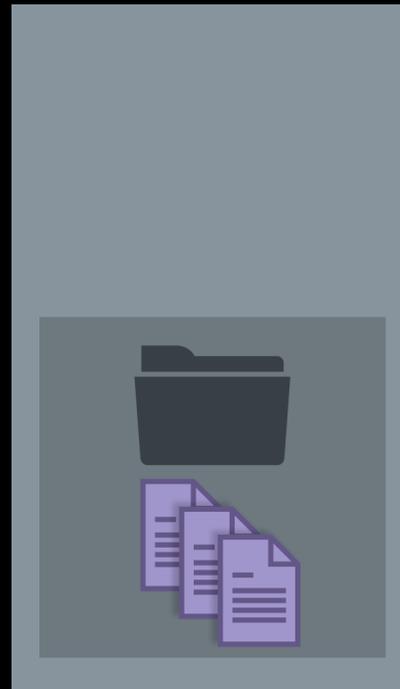


Enumerating Items

Working set enumerator

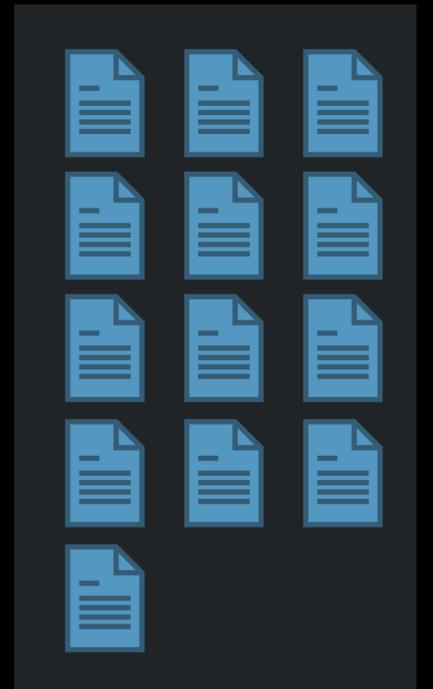
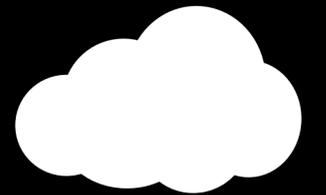
Enumerate your local copy of the working set

Files Provider
Daemon



File Provider
Extension

Cloud

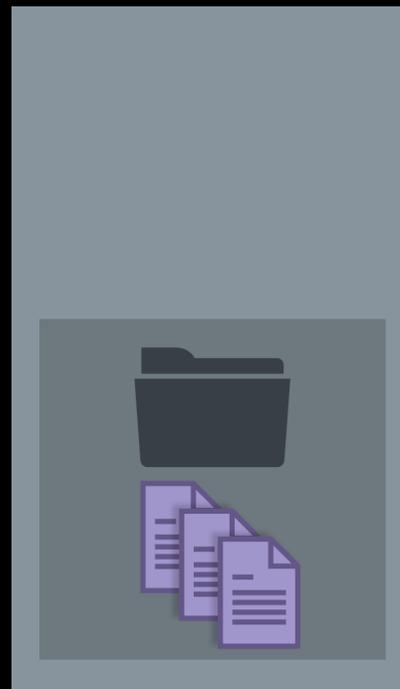


Enumerating Items

Working set enumerator

Enumerate your local copy of the working set

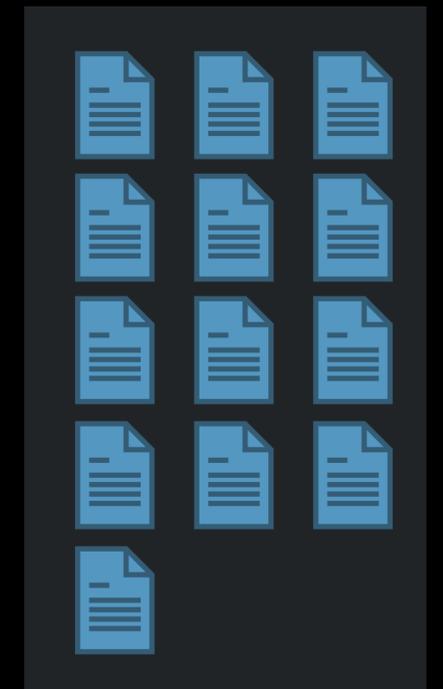
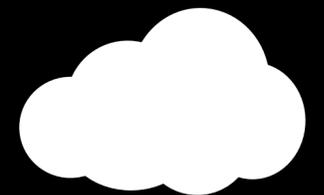
Files Provider
Daemon



File Provider
Extension

Database			Disk Storage
ID	Metadata	URL	
1	
2	
3	
4	
5	
6	
7	
8	
9	

Cloud

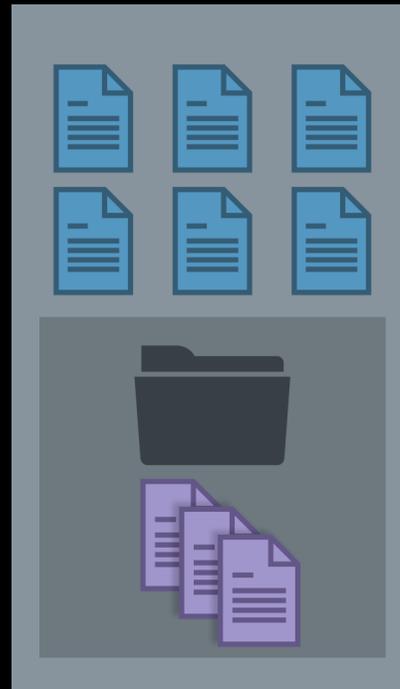


Enumerating Items

Working set enumerator

Enumerate your local copy of the working set

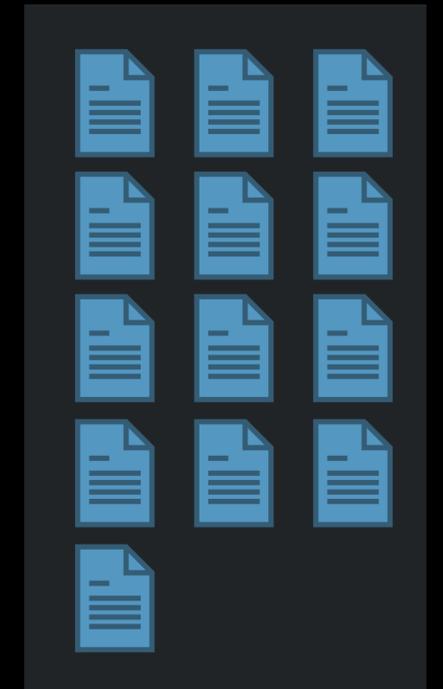
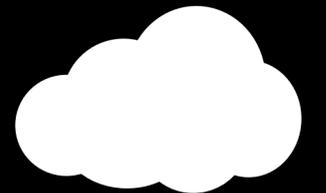
Files Provider Daemon



File Provider Extension

Database			Disk Storage
ID	Metadata	URL	
1	
2	
3	
4	
5	
6	
7	
8	
9	

Cloud



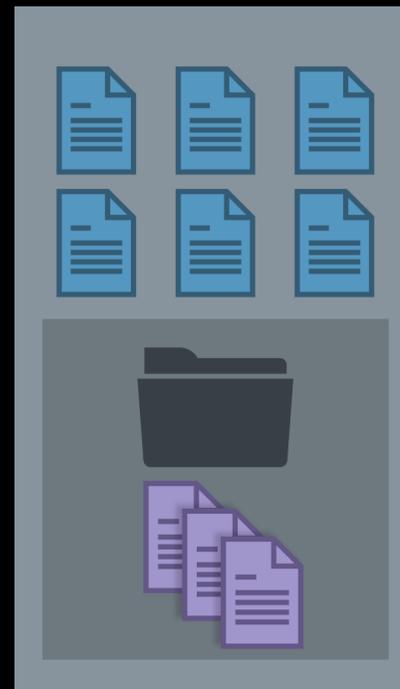
Enumerating Items

Working set enumerator

Enumerate your local copy of
the working set

Not the cloud

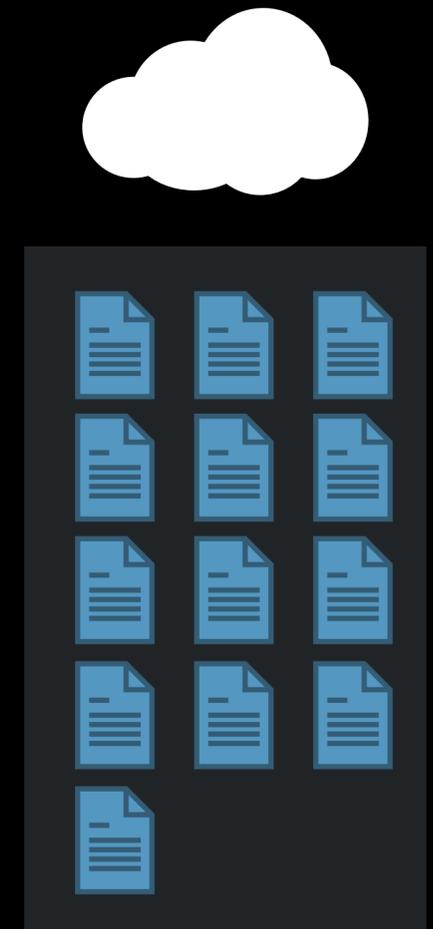
Files Provider
Daemon



File Provider
Extension

Database			Disk Storage
ID	Metadata	URL	
1	
2	
3	
4	
5	
6	
7	
8	
9	

Cloud



Enumerating Items

Working set enumerator

Enumerate your local copy of the working set

Not the cloud

The working set must be available offline

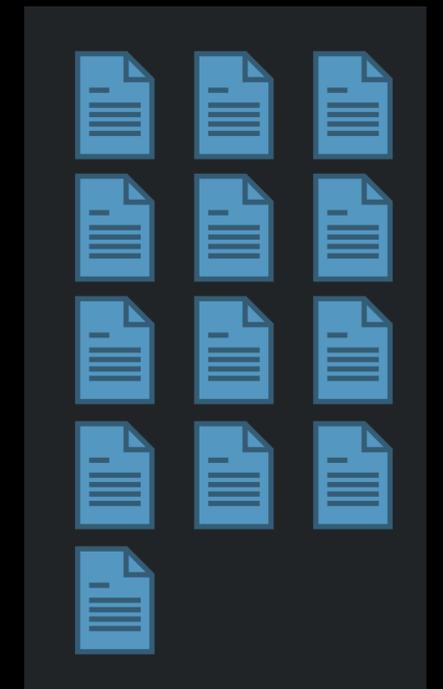
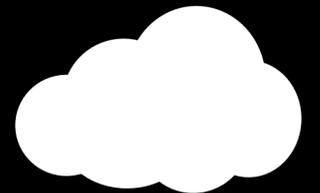
Files Provider Daemon



File Provider Extension

Database			Disk Storage
ID	Metadata	URL	
1	
2	
3	
4	
5	
6	
7	
8	
9	

Cloud



Enumerating Items

Working set versus folder

Enumerating Items

Working set versus folder

Working Set

Folders

Enumerating Items

Working set versus folder

Working Set

Folders

Read the Cloud



Never



OK

Enumerating Items

Working set versus folder

	Working Set	Folders
Read the Cloud	 Never	 OK
Read the Database	 Always	 OK

Syncing Changes

Remembering versions



Syncing Changes

Remembering versions

Assign version numbers



Syncing Changes

Remembering versions

Assign version numbers



Syncing Changes

Remembering versions

Assign version numbers

Maximum known version number is
the sync anchor

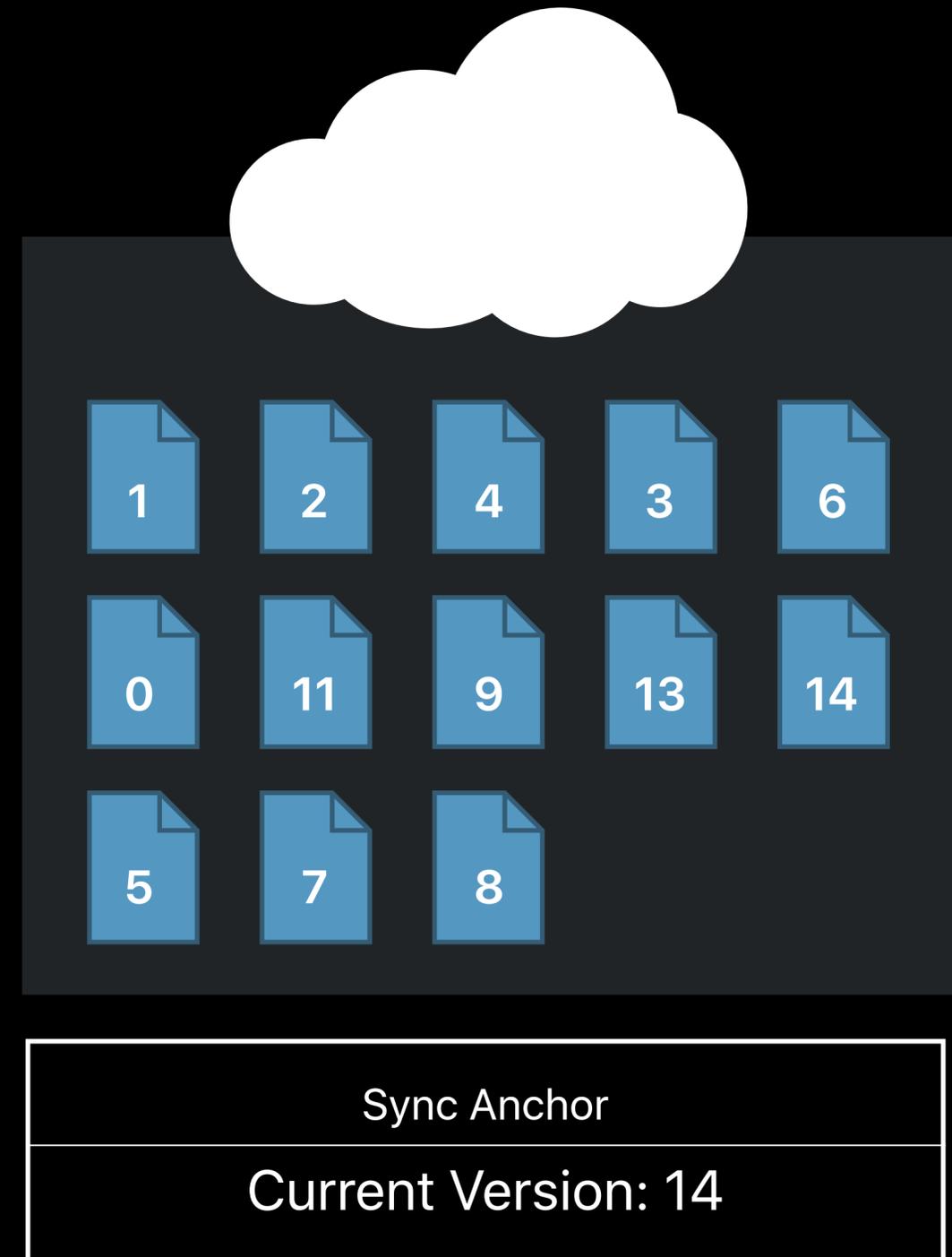


Syncing Changes

Remembering versions

Assign version numbers

Maximum known version number is the sync anchor



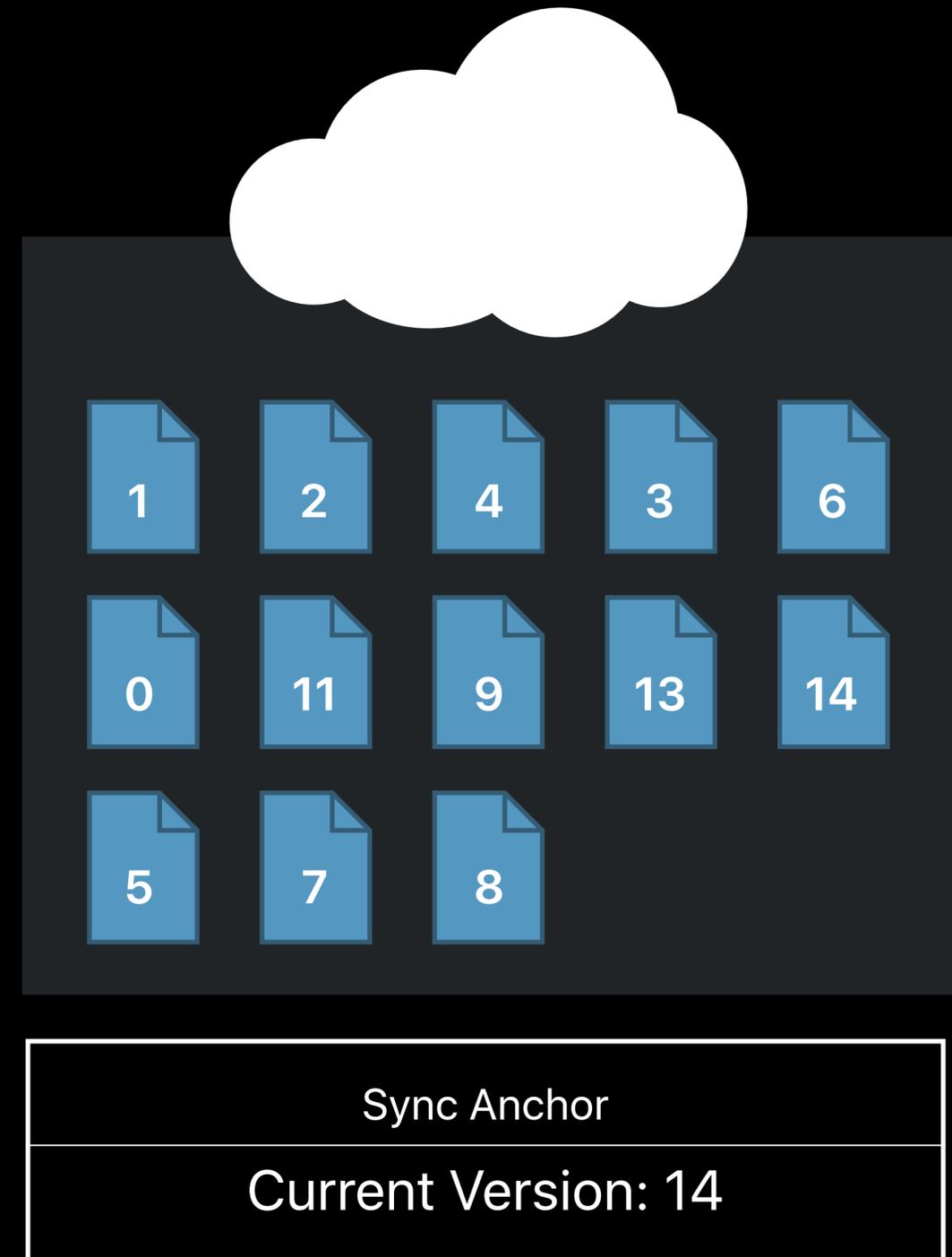
Syncing Changes

Remembering versions

Assign version numbers

Maximum known version number is the sync anchor

Bump version numbers on changes



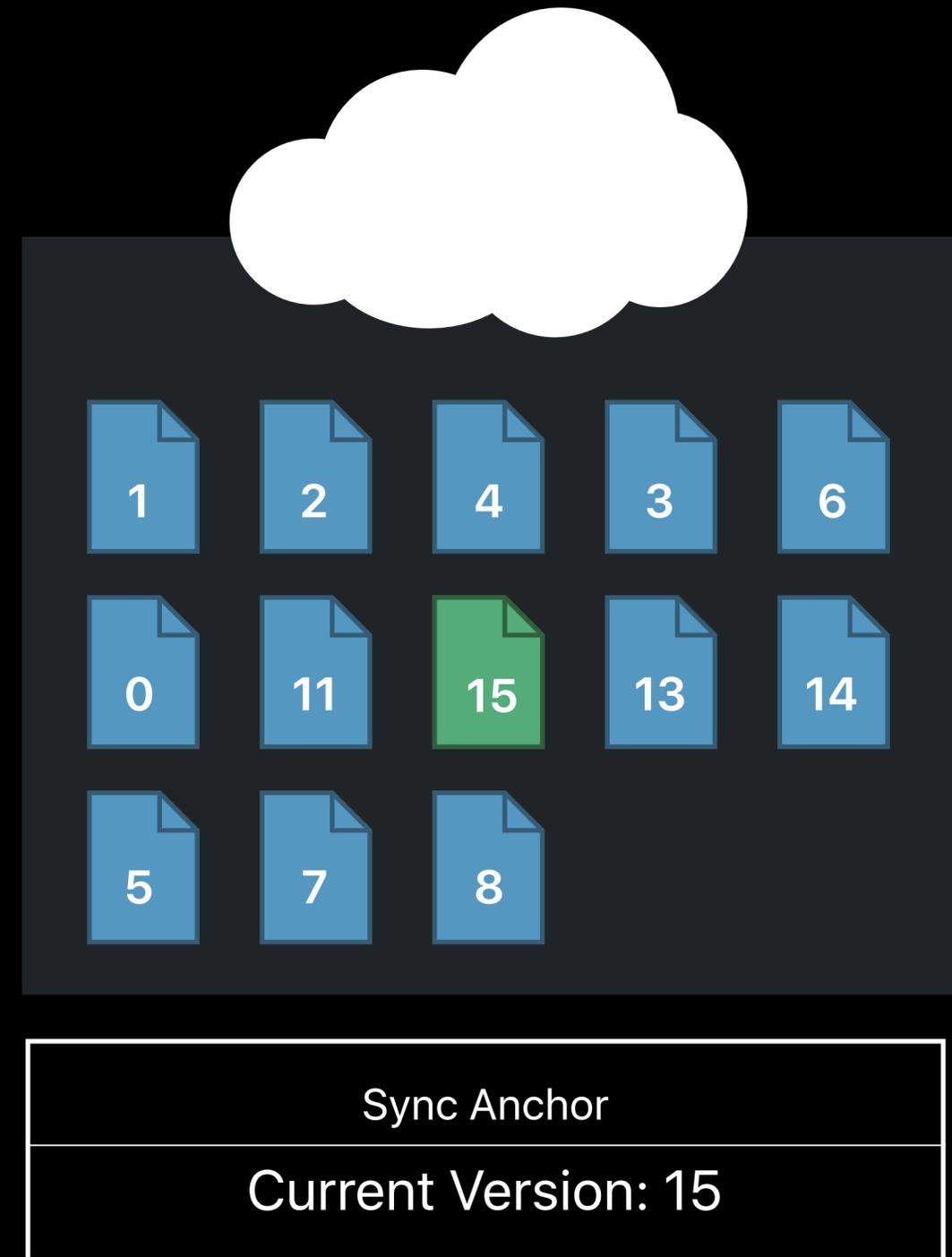
Syncing Changes

Remembering versions

Assign version numbers

Maximum known version number is the sync anchor

Bump version numbers on changes



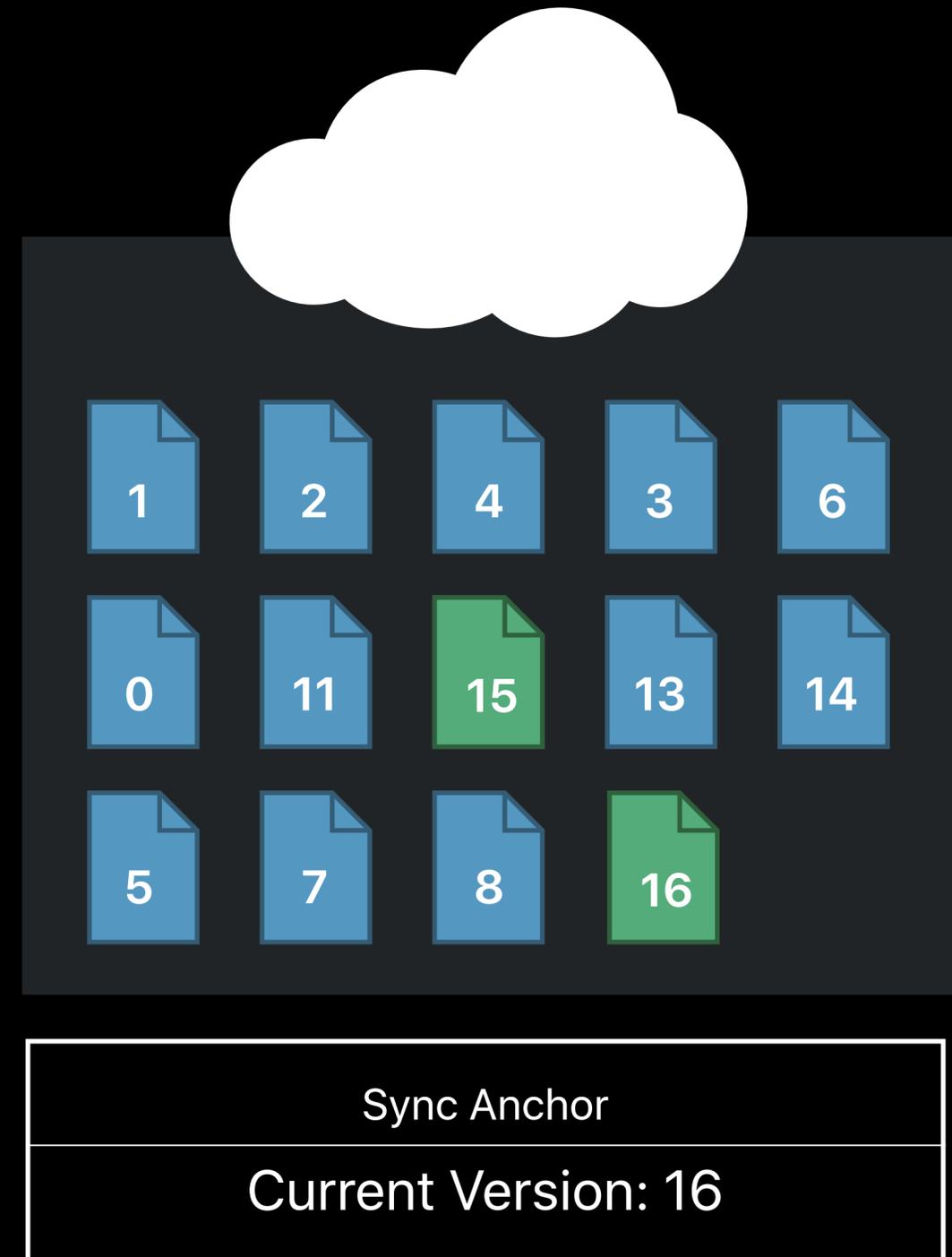
Syncing Changes

Remembering versions

Assign version numbers

Maximum known version number is the sync anchor

Bump version numbers on changes



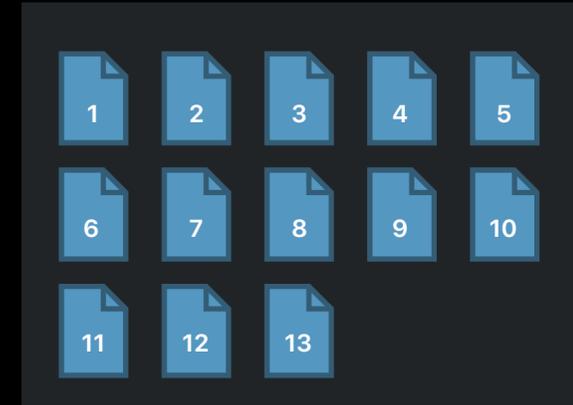
Syncing Changes

Initial sync anchor

Files app



File Provider Extension
Enumerator



NSFileProviderSyncAnchor

Current Version: 13

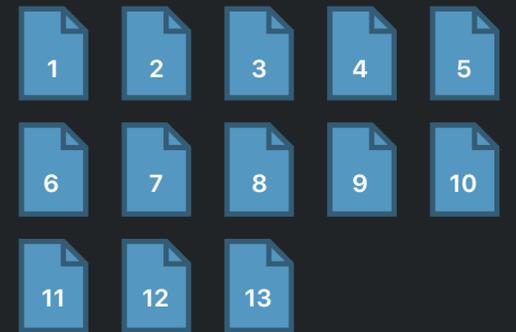
Syncing Changes

Initial sync anchor

Files app

```
enumerator.currentSyncAnchor()
```

File Provider Extension
Enumerator



NSFileProviderSyncAnchor

Current Version: 13

Syncing Changes

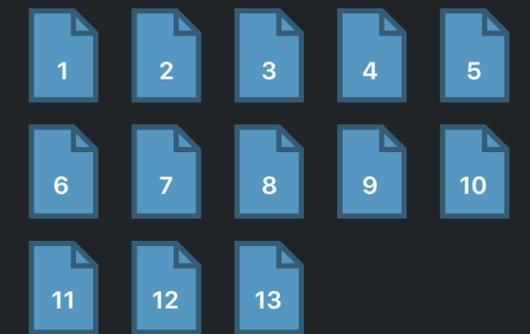
Initial sync anchor

Files app

```
enumerator.currentSyncAnchor()
```

File Provider Extension
Enumerator

```
func currentSyncAnchor(completionHandler:)
```



NSFileProviderSyncAnchor

Current Version: 13

Syncing Changes

Initial sync anchor

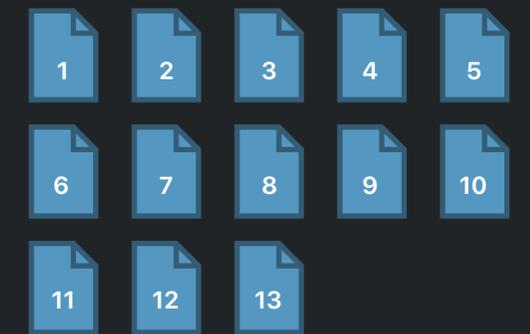
Files app

```
enumerator.currentSyncAnchor()
```

File Provider Extension
Enumerator

```
func currentSyncAnchor(completionHandler:)
```

```
completionHandler(self.syncAnchor)
```

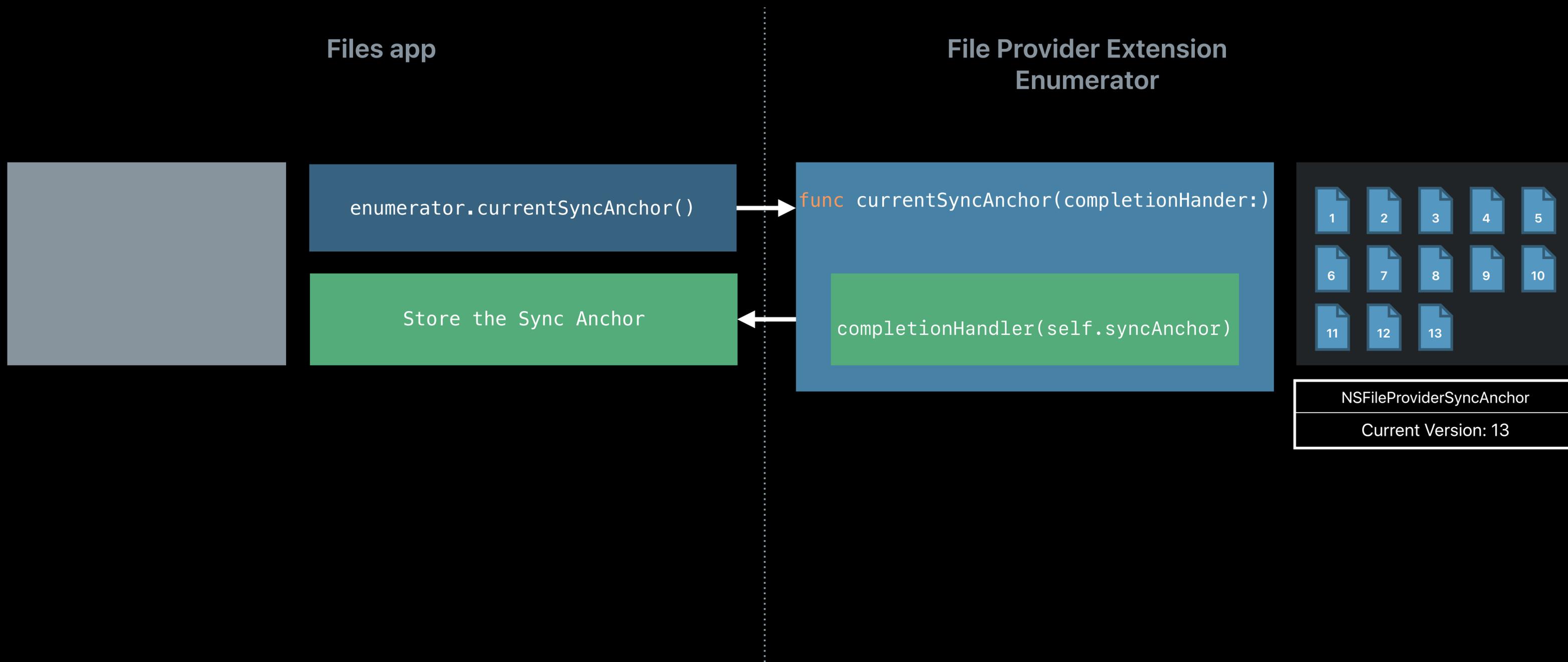


NSFileProviderSyncAnchor

Current Version: 13

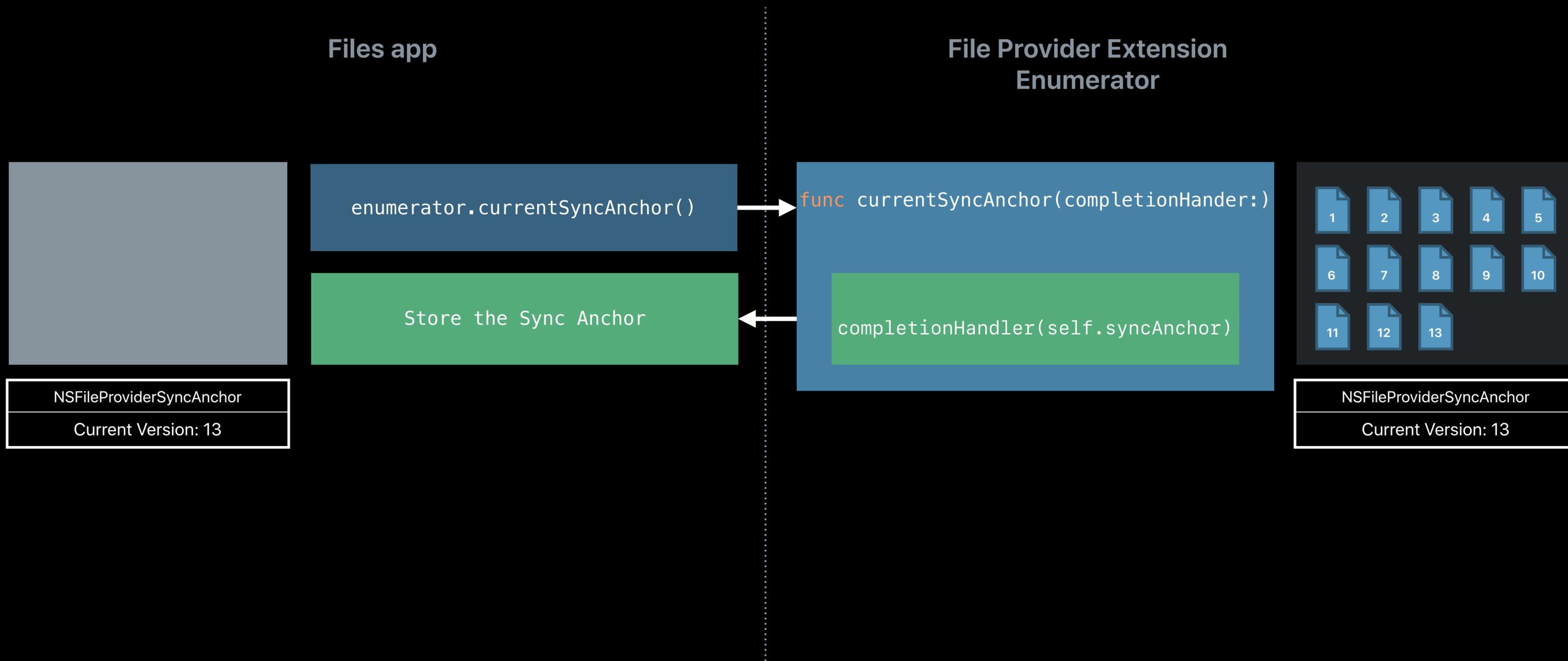
Syncing Changes

Initial sync anchor



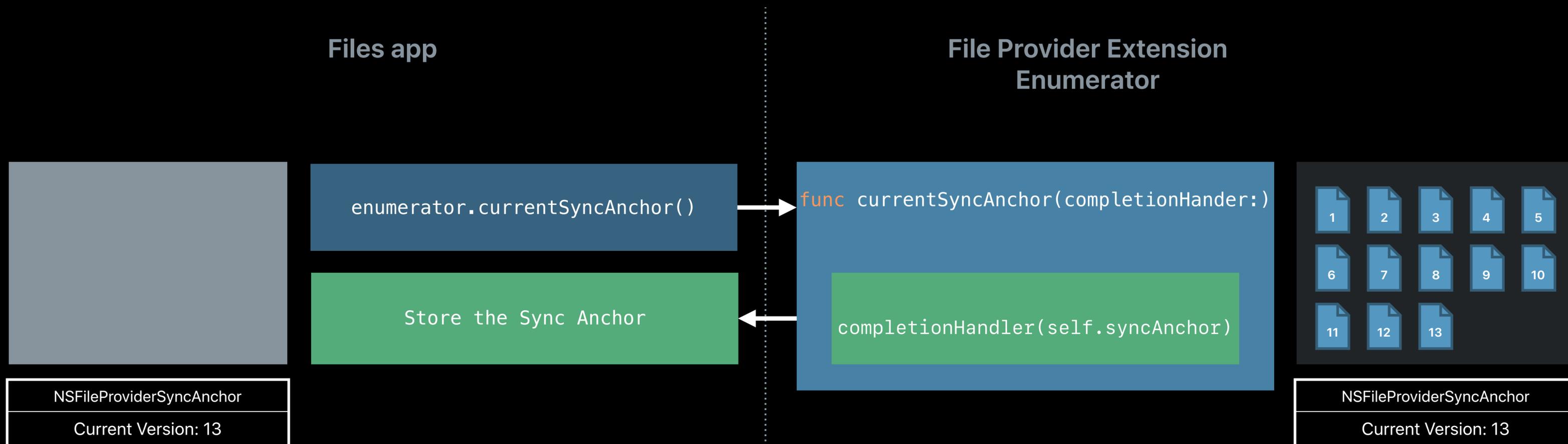
Syncing Changes

Initial sync anchor



Syncing Changes

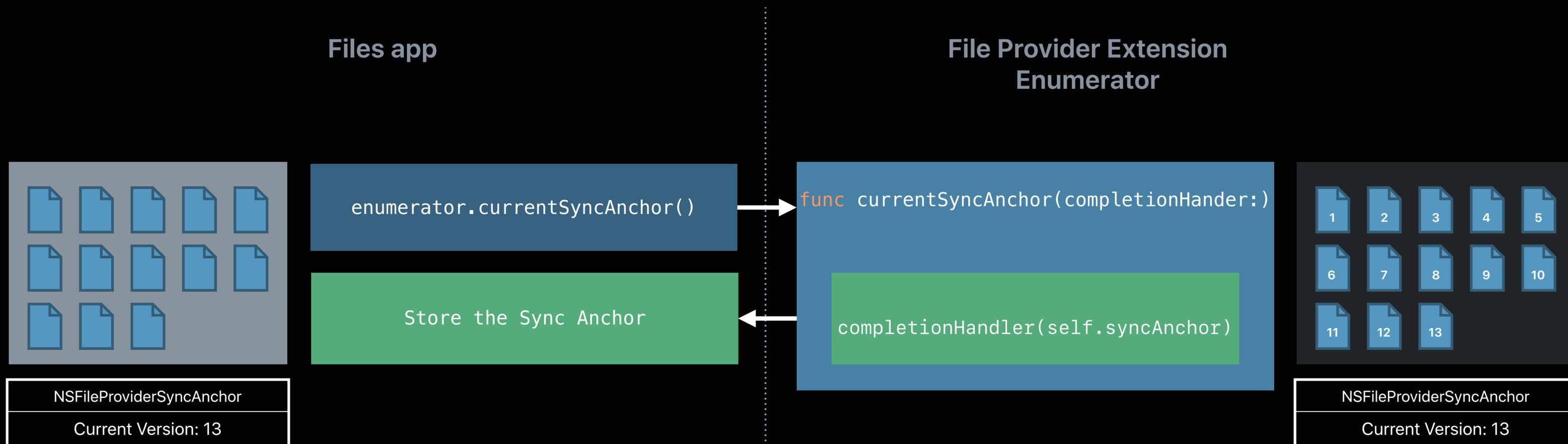
Initial sync anchor



... Requests and receives the pages ...

Syncing Changes

Initial sync anchor

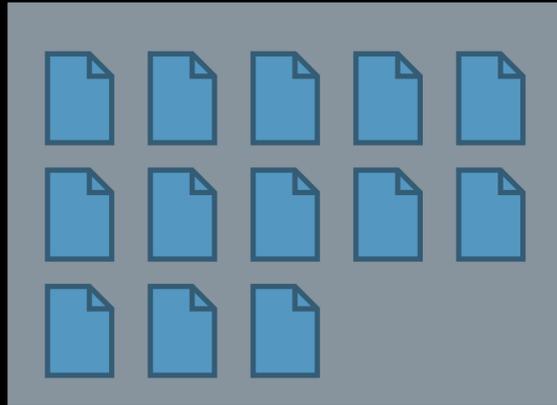


... Requests and receives the pages ...

Syncing Changes

Enumerate changes

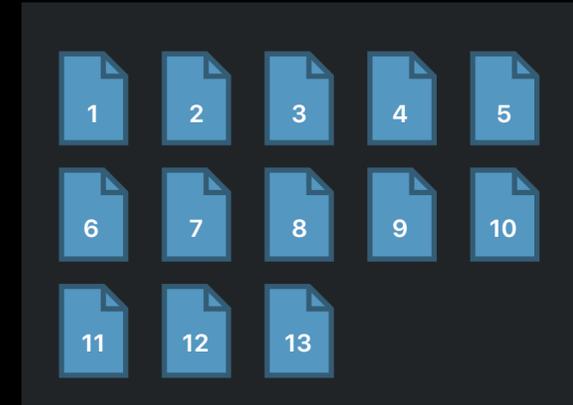
Files app



NSFileProviderSyncAnchor

Current Version: 13

File Provider Extension
Enumerator



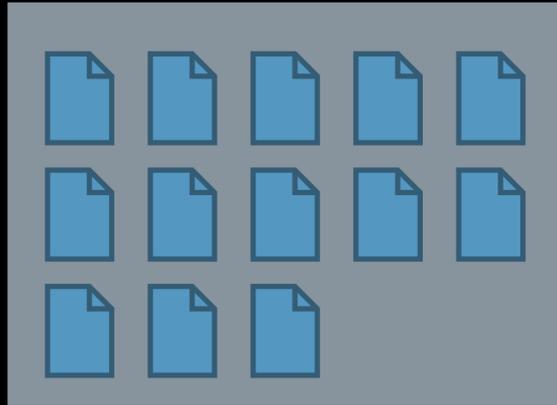
NSFileProviderSyncAnchor

Current Version: 13

Syncing Changes

Enumerate changes

Files app

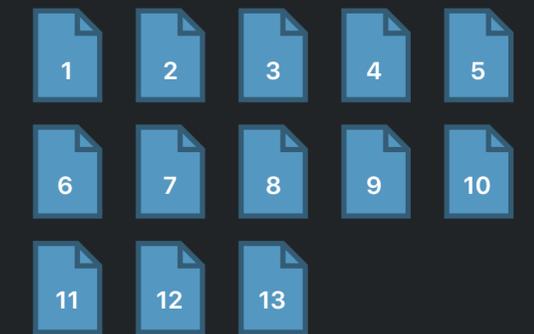


NSFileProviderSyncAnchor

Current Version: 13

File Provider Extension
Enumerator

Add a new File



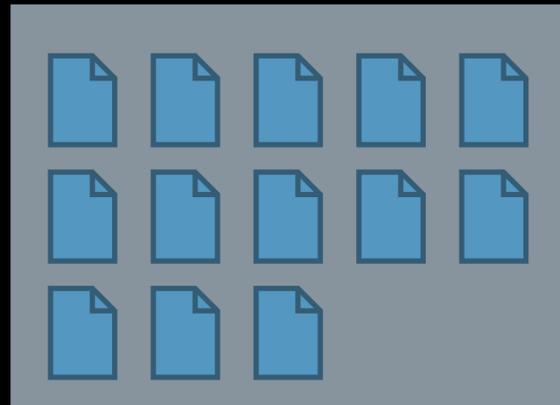
NSFileProviderSyncAnchor

Current Version: 13

Syncing Changes

Enumerate changes

Files app

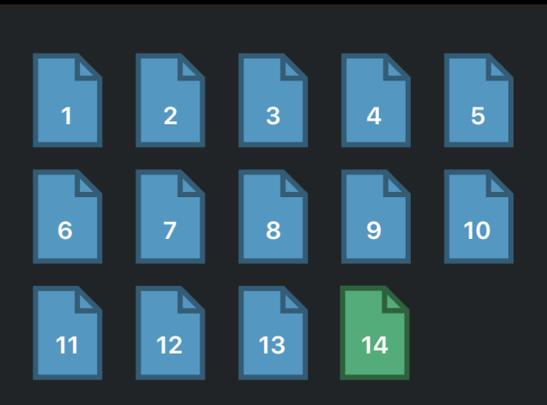


NSFileProviderSyncAnchor

Current Version: 13

File Provider Extension
Enumerator

Add a new File



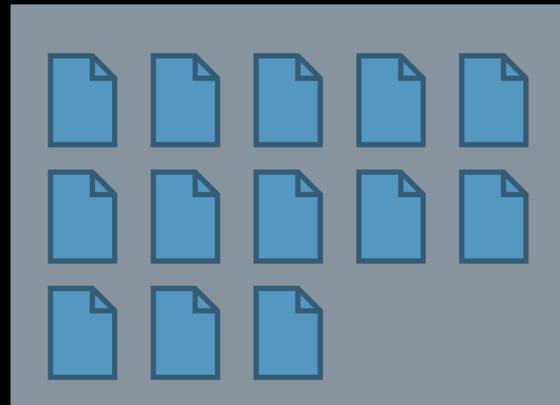
NSFileProviderSyncAnchor

Current Version: 13

Syncing Changes

Enumerate changes

Files app

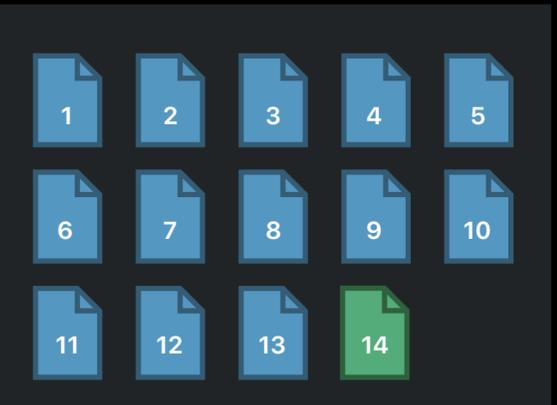


NSFileProviderSyncAnchor

Current Version: 13

File Provider Extension
Enumerator

Add a new File

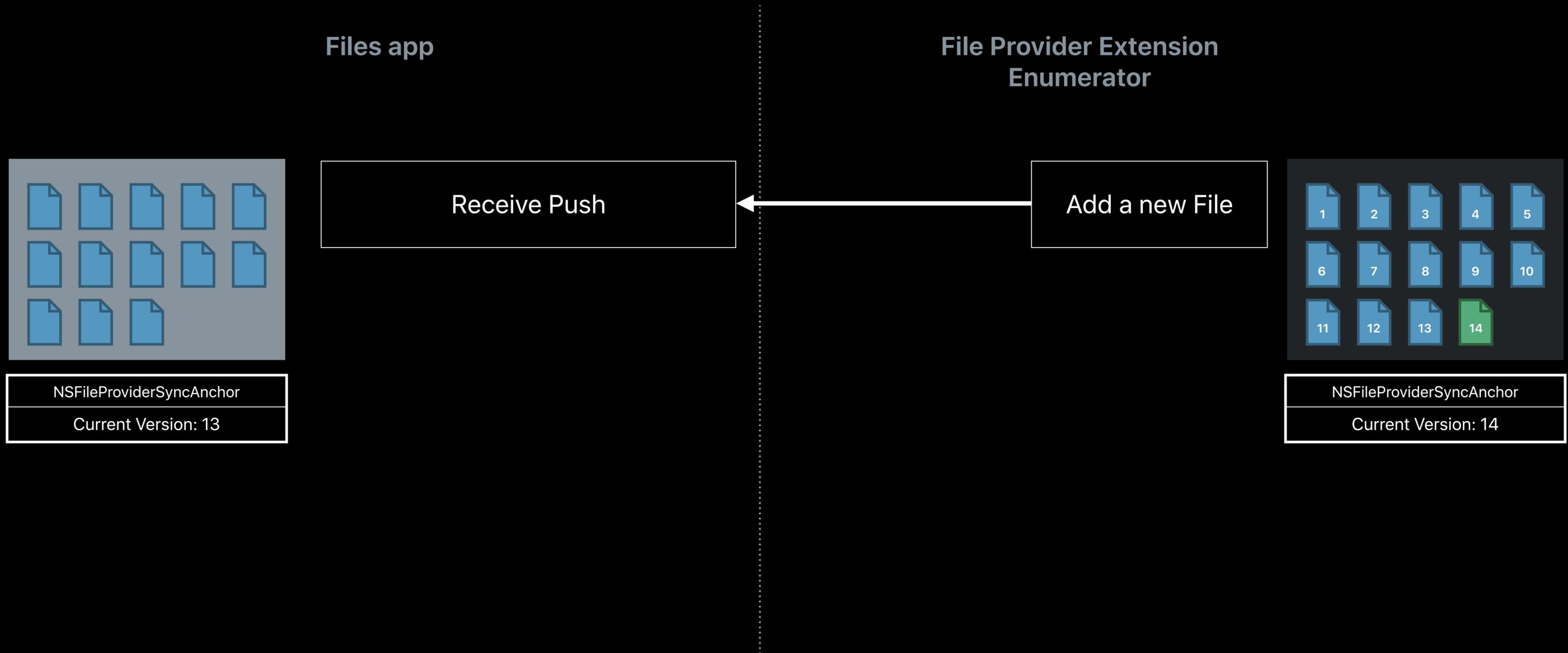


NSFileProviderSyncAnchor

Current Version: 14

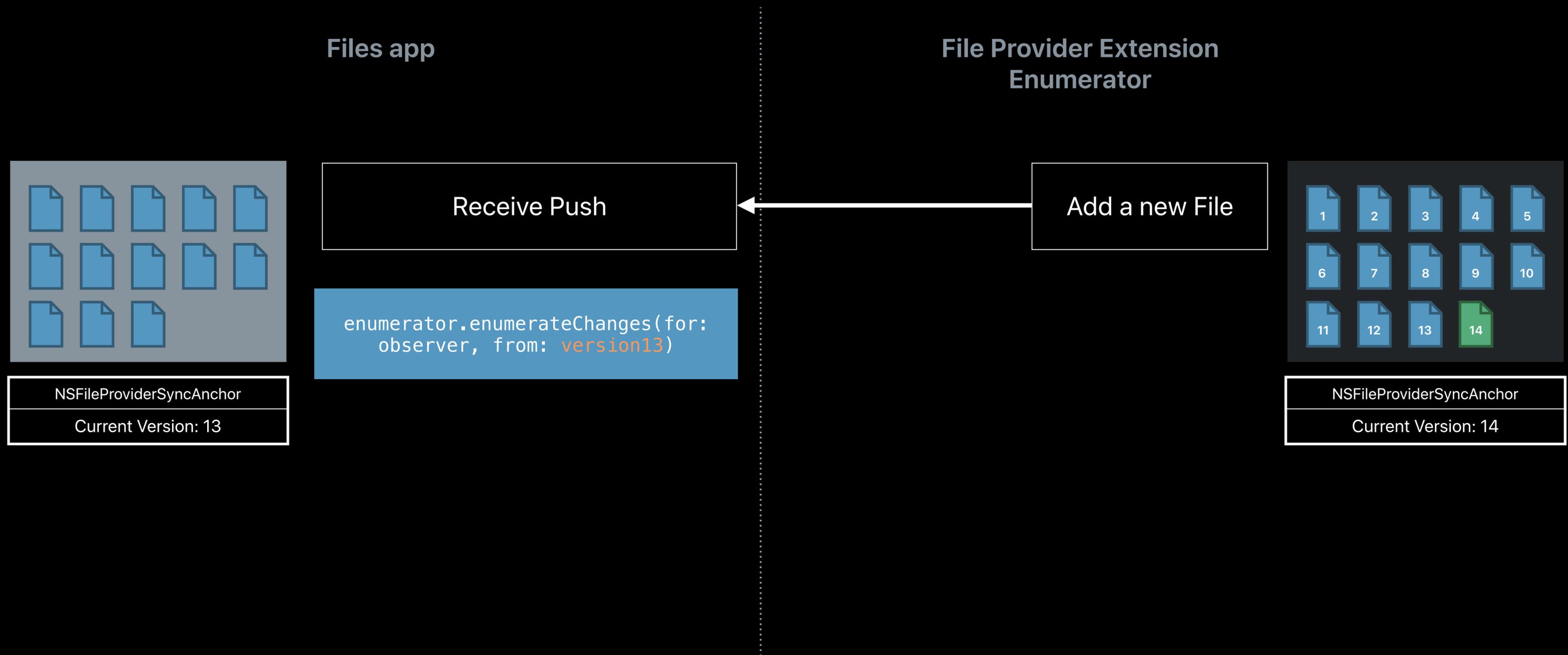
Syncing Changes

Enumerate changes



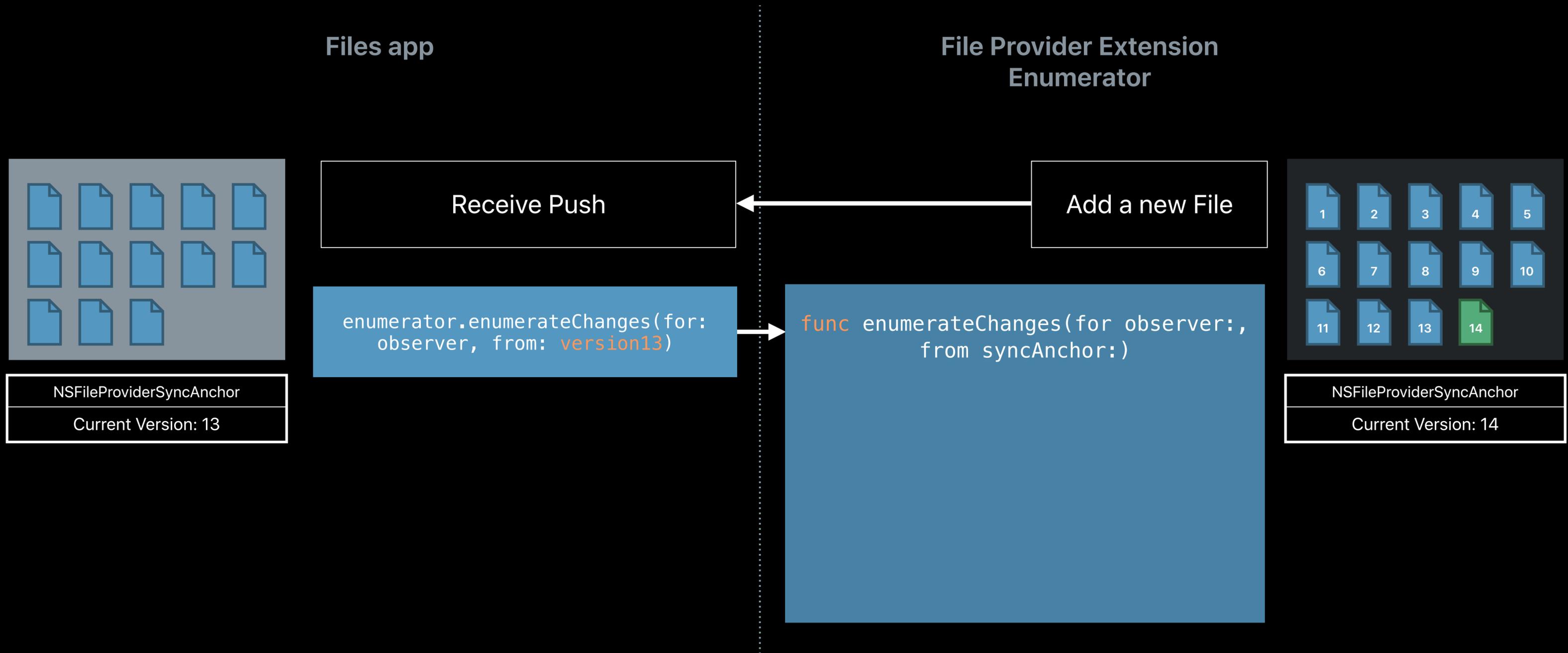
Syncing Changes

Enumerate changes



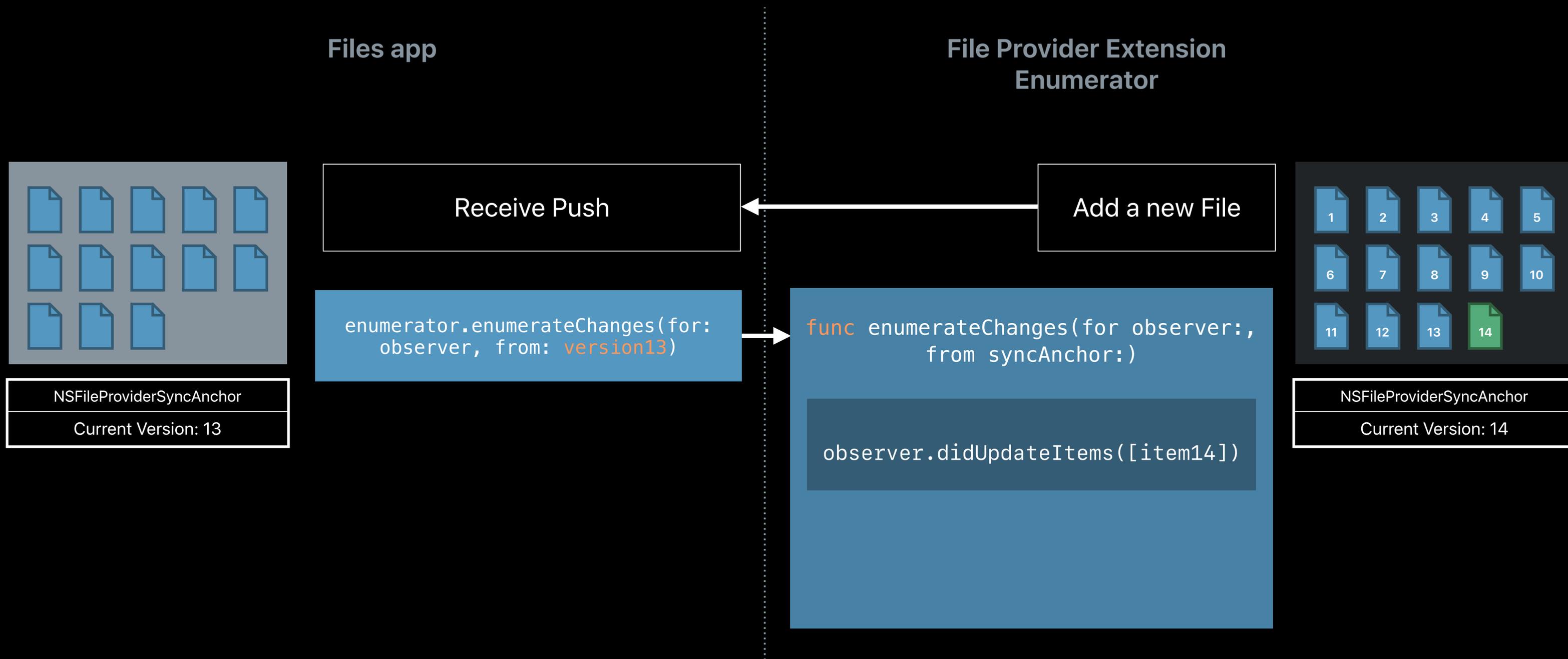
Syncing Changes

Enumerate changes



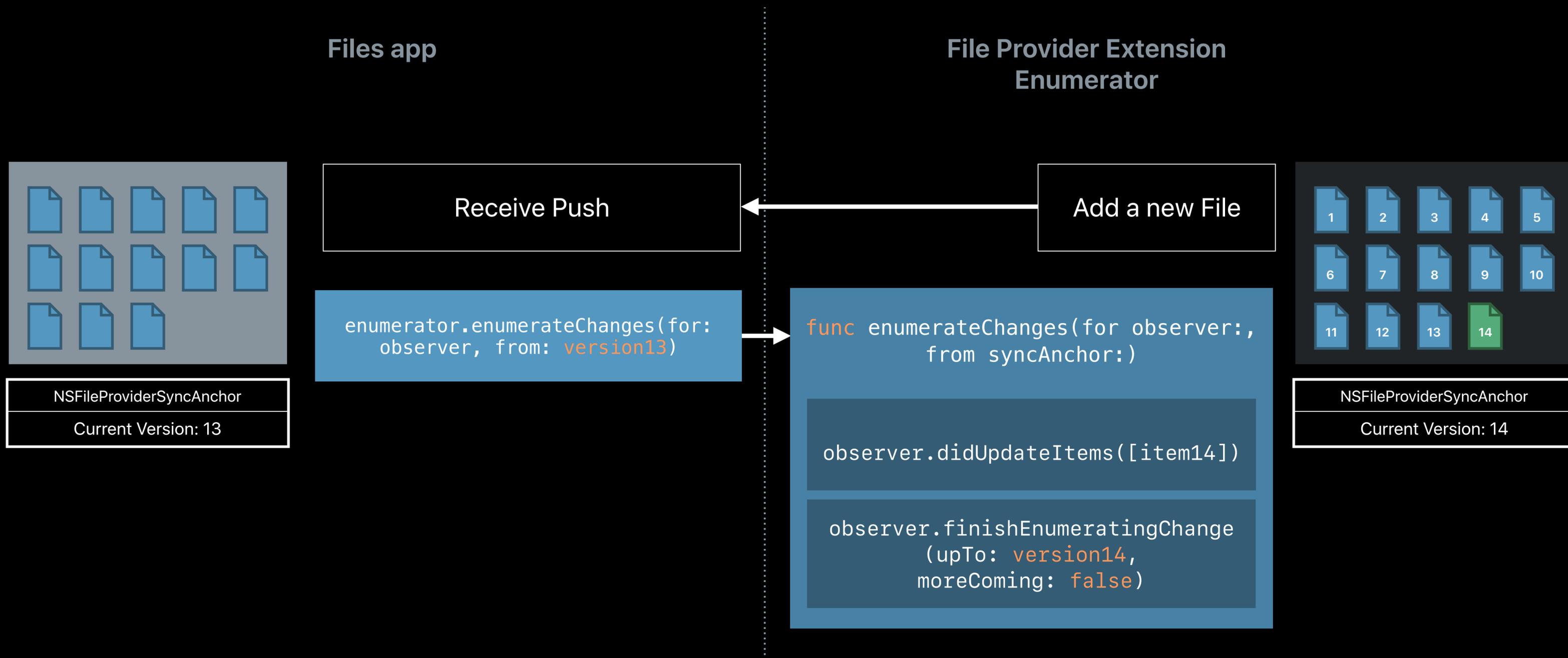
Syncing Changes

Enumerate changes



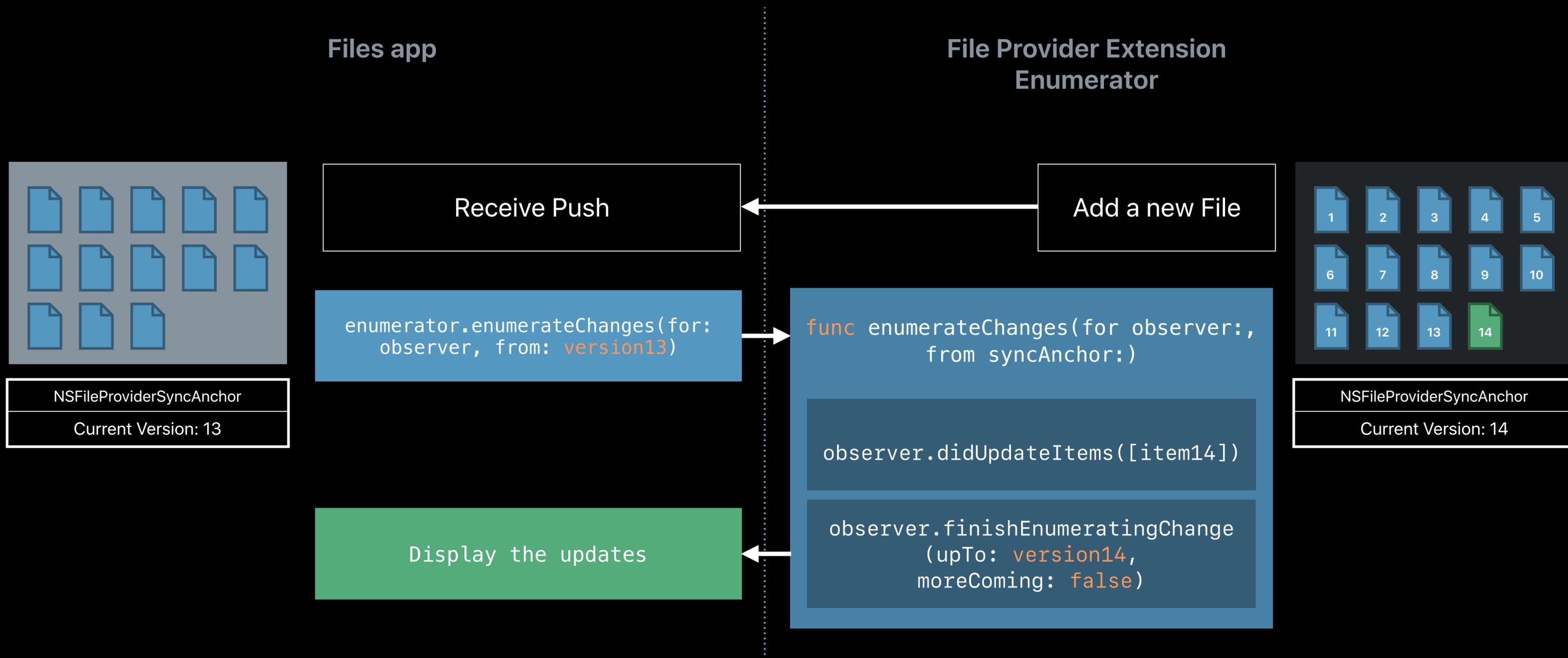
Syncing Changes

Enumerate changes



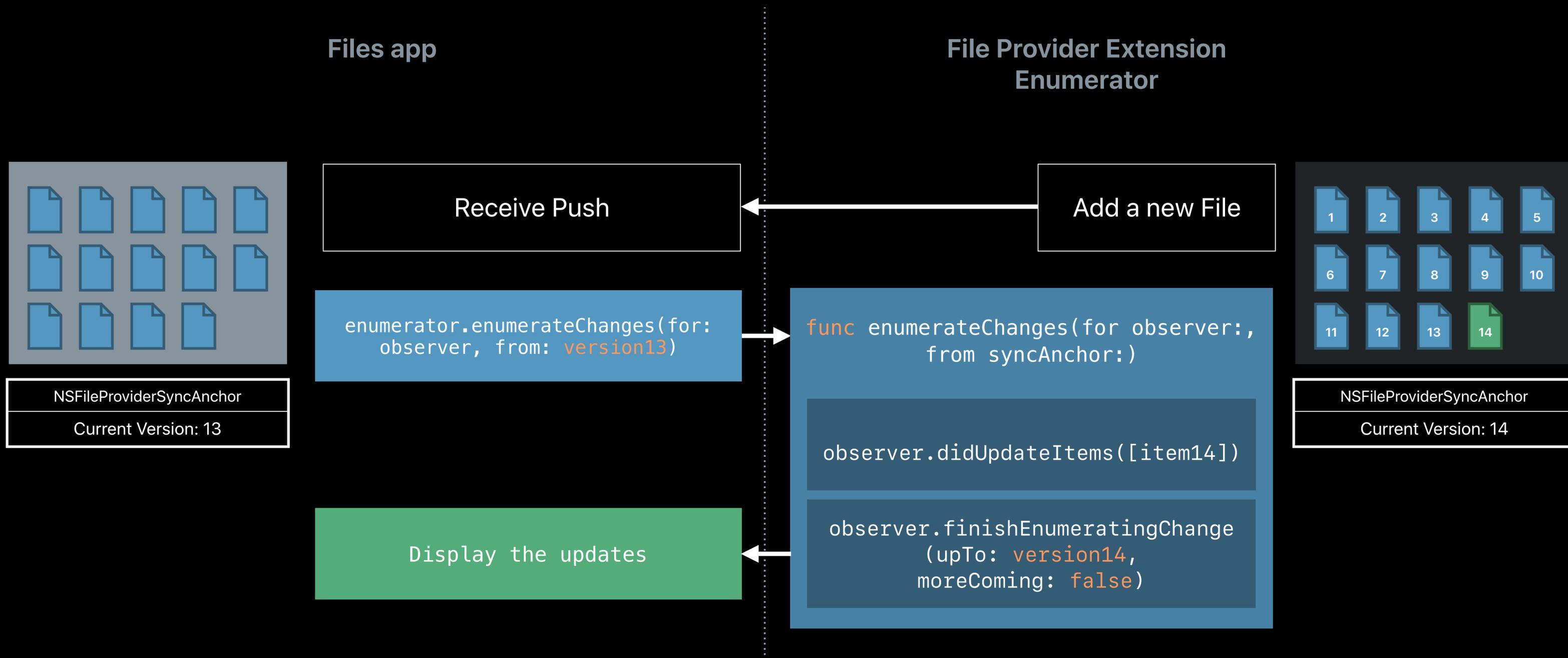
Syncing Changes

Enumerate changes



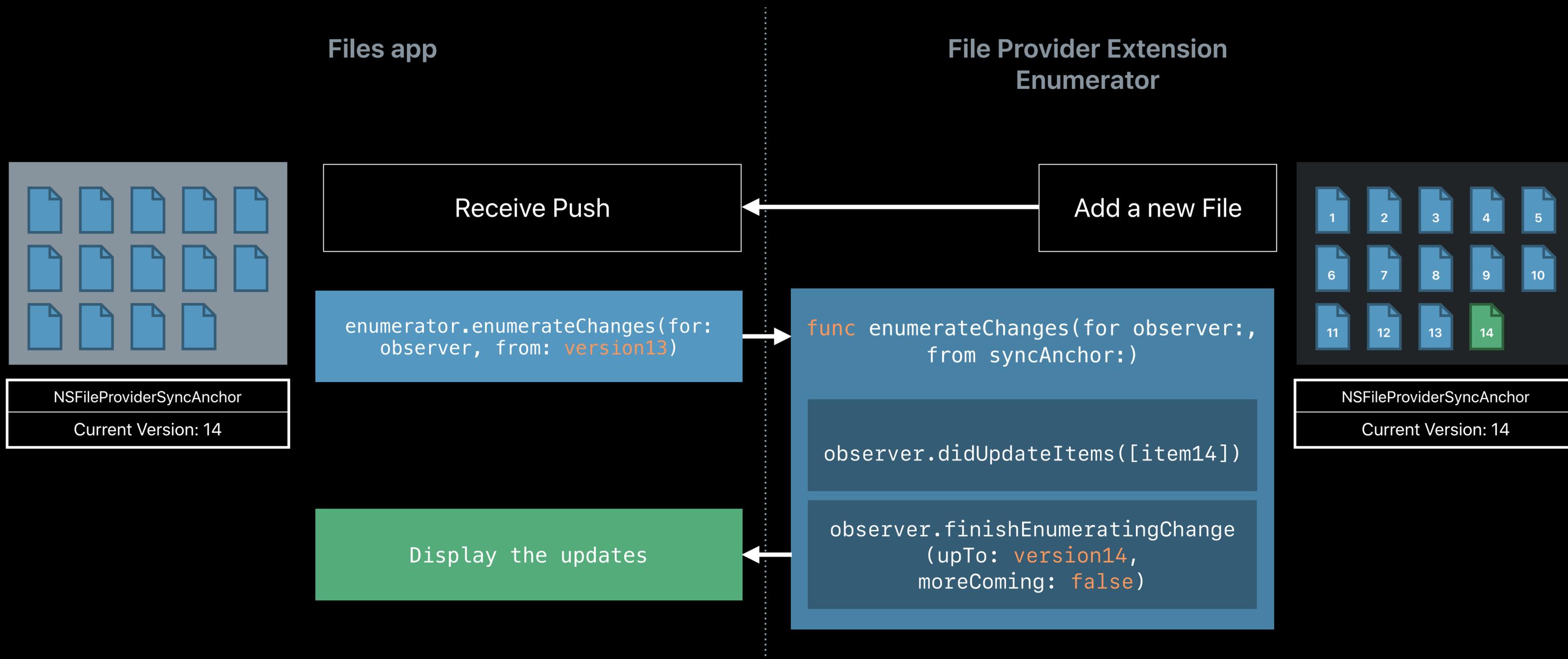
Syncing Changes

Enumerate changes



Syncing Changes

Enumerate changes

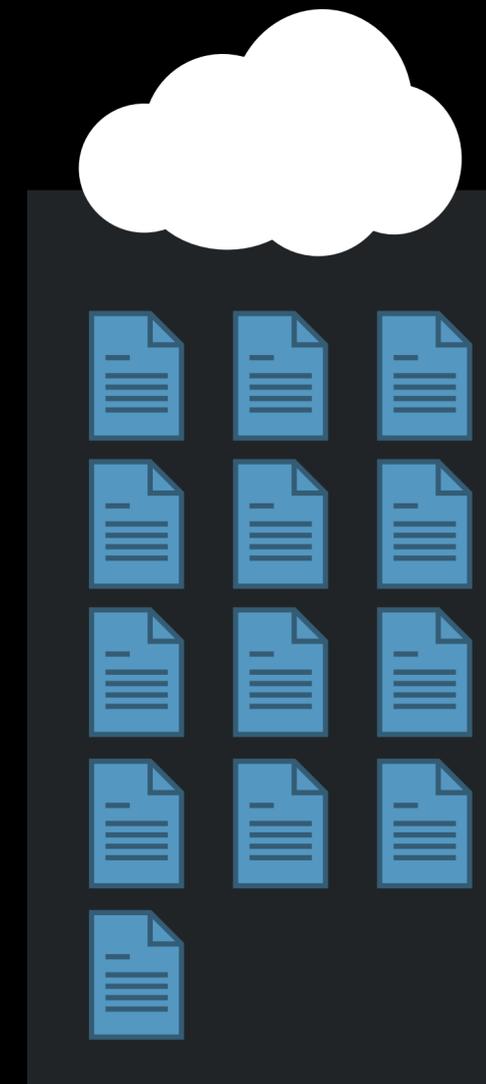


Push Notification

Device

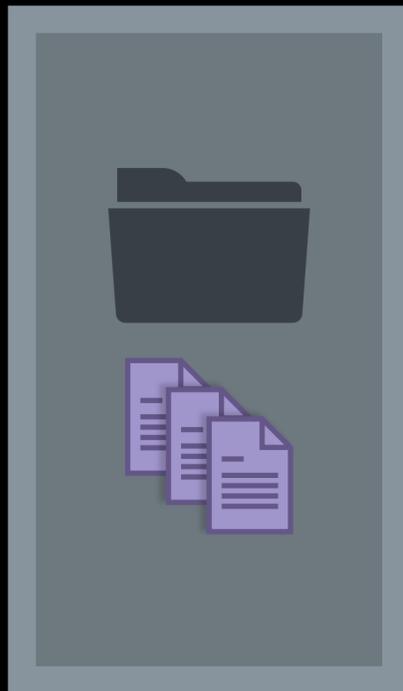
Database			Disk Storage
ID	Metadata	URL	
1	
2	
3	
4	
5	
6	
7	
8	
9	

Cloud



Push Notification

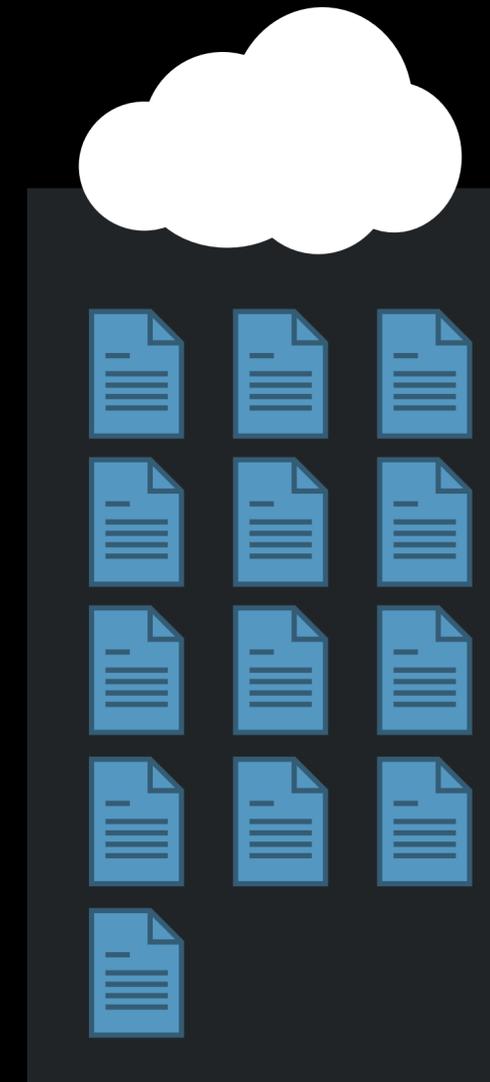
File Provider Daemon



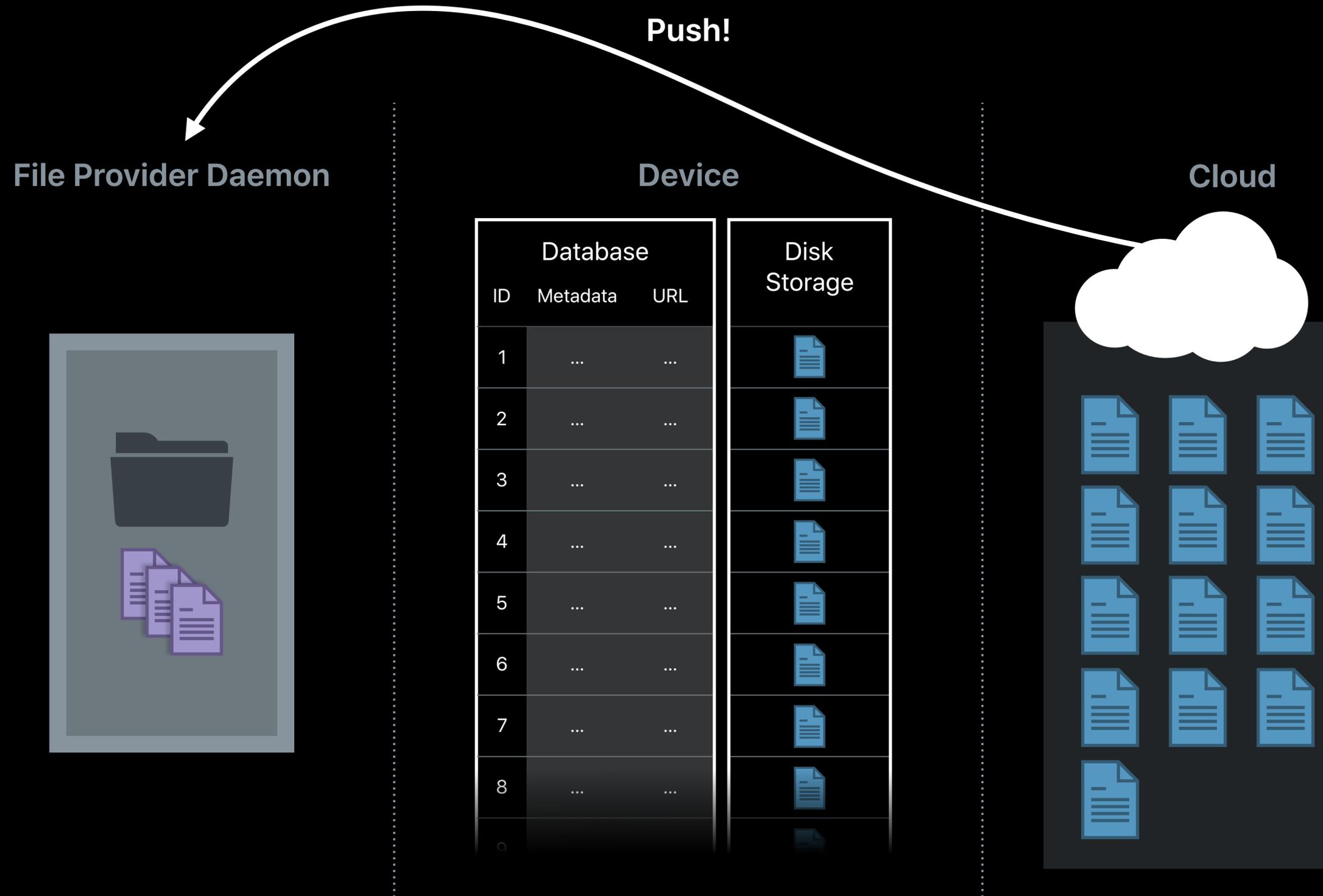
Device

Database			Disk Storage
ID	Metadata	URL	
1	
2	
3	
4	
5	
6	
7	
8	
9	

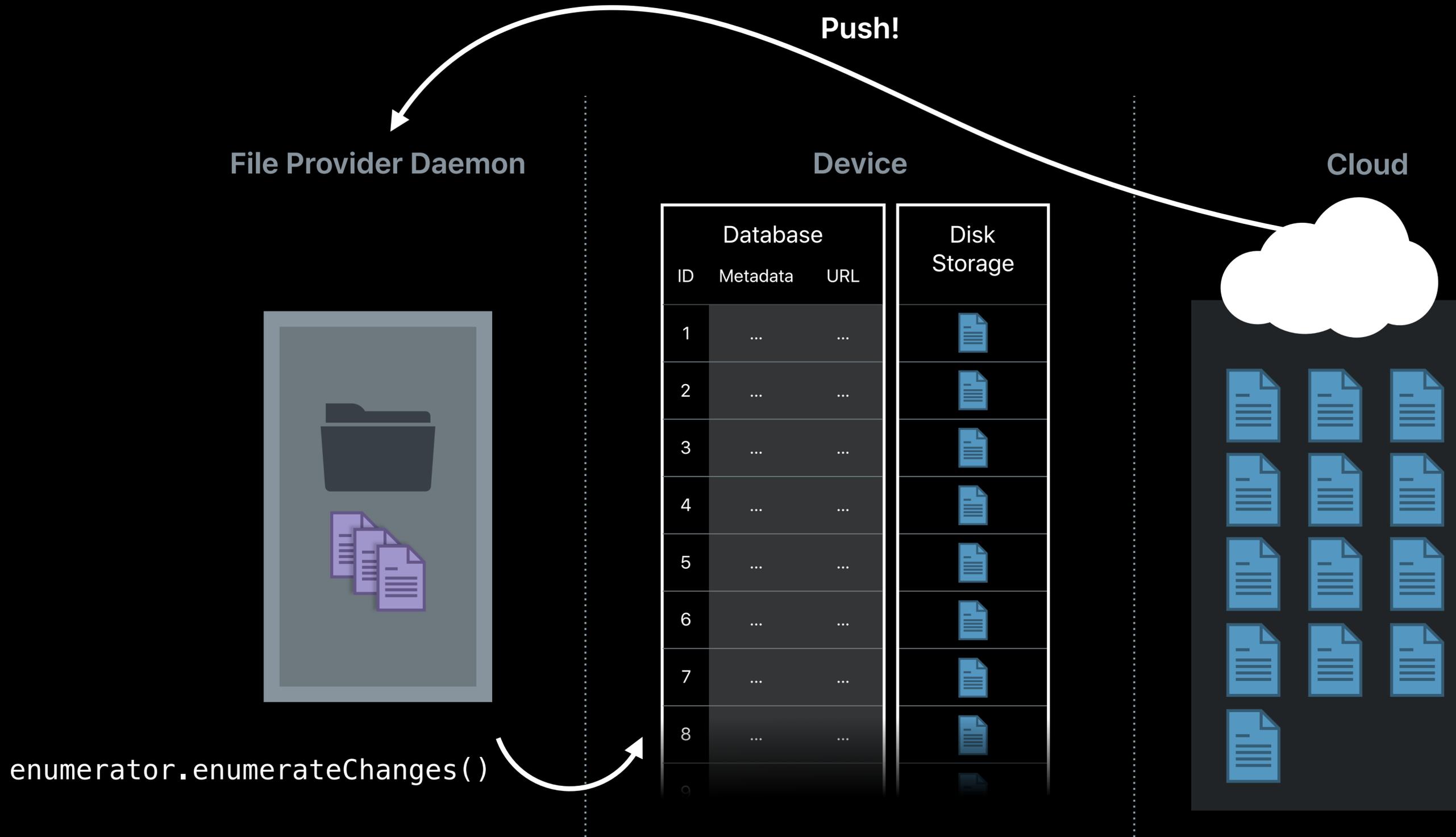
Cloud



Push Notification



Push Notification



Push Notification

Under the hood

Push Notification

Under the hood

PKPushType is

```
PKPushTypeFileProvider
```

Push Notification

Under the hood

PKPushType is

```
PKPushTypeFileProvider
```

Topic name is

```
<bundle-identifier>.pushkit.fileprovider
```

Push Notification

Under the hood

PKPushType is

```
PKPushTypeFileProvider
```

Topic name is

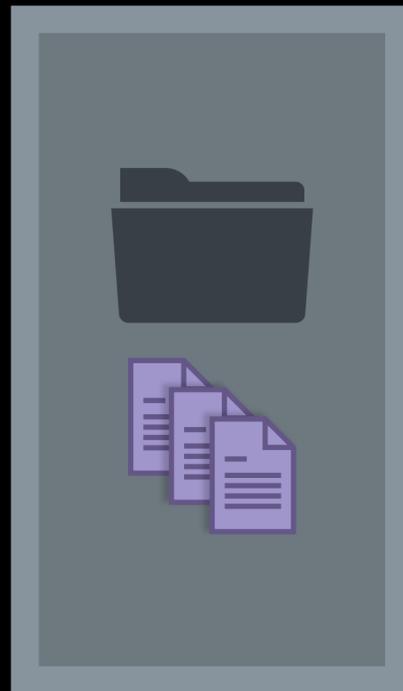
```
<bundle-identifier>.pushkit.fileprovider
```

Push payload must include

```
"container-identifier": NSFileProviderItemIdentifier
```

Signaling

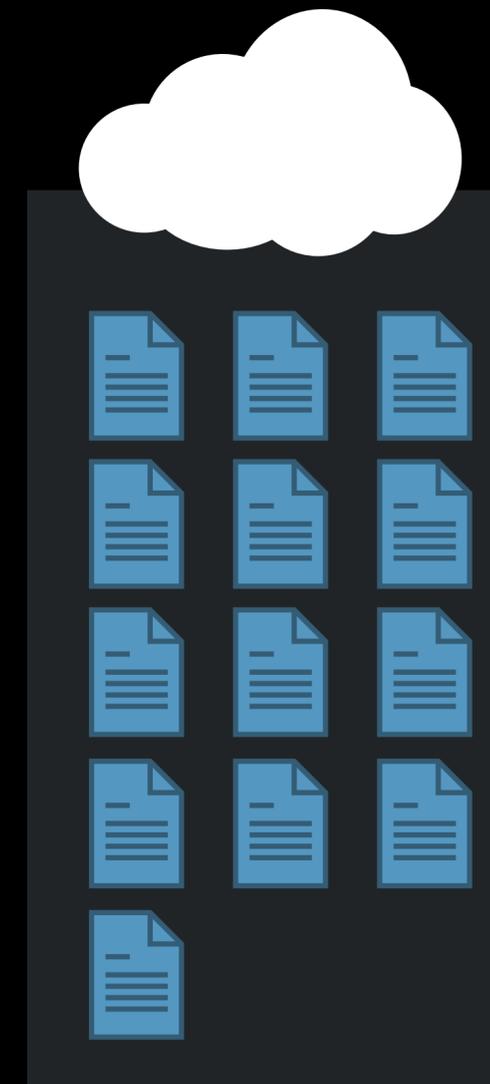
File Provider Daemon



Device

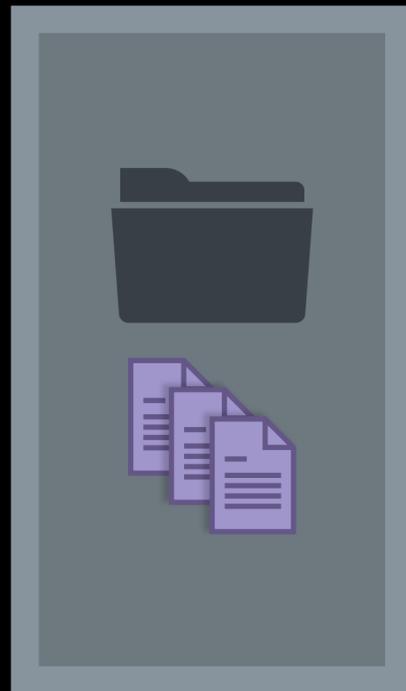
Database			Disk Storage
ID	Metadata	URL	
1	
2	
3	
4	
5	
6	
7	
8	
9	

Cloud



Signaling

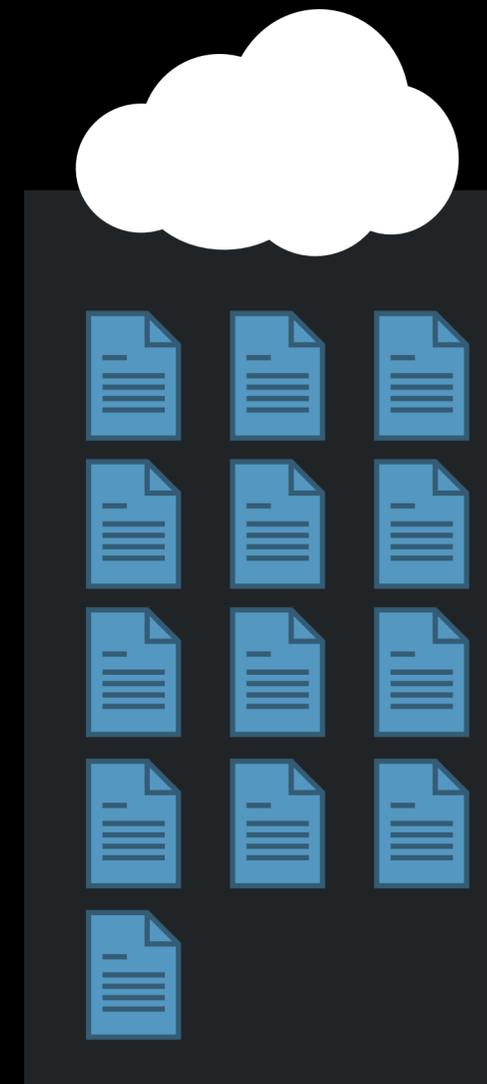
File Provider Daemon



Device

Database			Disk Storage
ID	Metadata	URL	
1	
2	
3	
4	
5	
6	
7	
8	
⋮			

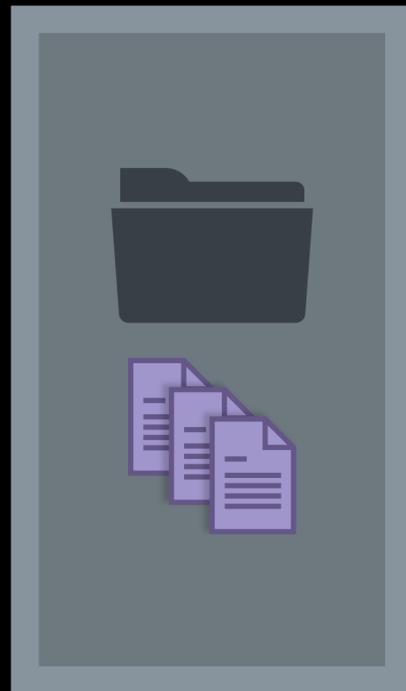
Cloud



Signaling

```
NSFileManagerManager.signalChanges(forContainer: NSFileManagerItemIdentifier)
```

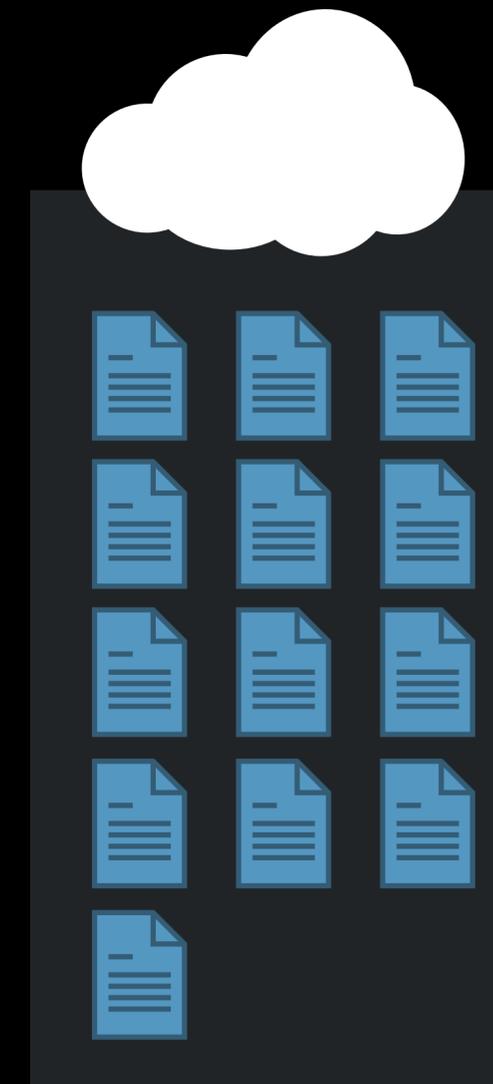
File Provider Daemon



Device

Database			Disk Storage
ID	Metadata	URL	
1	
2	
3	
4	
5	
6	
7	
8	
9	

Cloud



Signaling

```
NSFileManagerManager.signalChanges(forContainer: NSFileManagerItemIdentifier)
```



Signaling Changes

```
NSFileManager.signalChanges(forContainer: NSFileProviderItemIdentifier)
```

Signaling Changes

```
NSFileManager.signalChanges(forContainer: NSFileProviderItemIdentifier)
```

When?

Signaling Changes

```
NSFileManager.signalChanges(forContainer: NSFileProviderItemIdentifier)
```

When?

- Tell the system about the initial set of files

Signaling Changes

```
NSFileManager.signalChanges(forContainer: NSFileProviderItemIdentifier)
```

When?

- Tell the system about the initial set of files
- Tell the system to reload after an account change

Signaling Changes

```
NSFileManager.signalChanges(forContainer: NSFileProviderItemIdentifier)
```

When?

- Tell the system about the initial set of files
- Tell the system to reload after an account change
- For any change without a corresponding push, including the one made locally

Signaling Changes

Notes

Signaling Changes

Notes

After calling `startProvidingItem(at:)` the item may be downloading

Signaling Changes

Notes

After calling `startProvidingItem(at:)` the item may be downloading

If so, you should

Signaling Changes

Notes

After calling `startProvidingItem(at:)` the item may be downloading

If so, you should

`signalChanges(forContainer: .workingSet)` if the item is in the `.workingSet`

Signaling Changes

Notes

After calling `startProvidingItem(at:)` the item may be downloading

If so, you should

`signalChanges(forContainer: .workingSet)` if the item is in the `.workingSet`

`signalChanges(forContainer: item.parentItemIdentifier)` if an enumerator is running

Signaling Changes

Under the hood

Signaling Changes

Under the hood

`signalChanges()` can be called from both your app and your file provider extension

Signaling Changes

Under the hood

`signalChanges()` can be called from both your app and your file provider extension

You must call `signalChanges()` even for system-initiated changes

Signaling Changes

Under the hood

`signalChanges()` can be called from both your app and your file provider extension

You must call `signalChanges()` even for system-initiated changes

`signalChanges()` is cheap!

Creating a File Provider Extension

Creating the project

Providing an item

Enumerating items

Modifying items

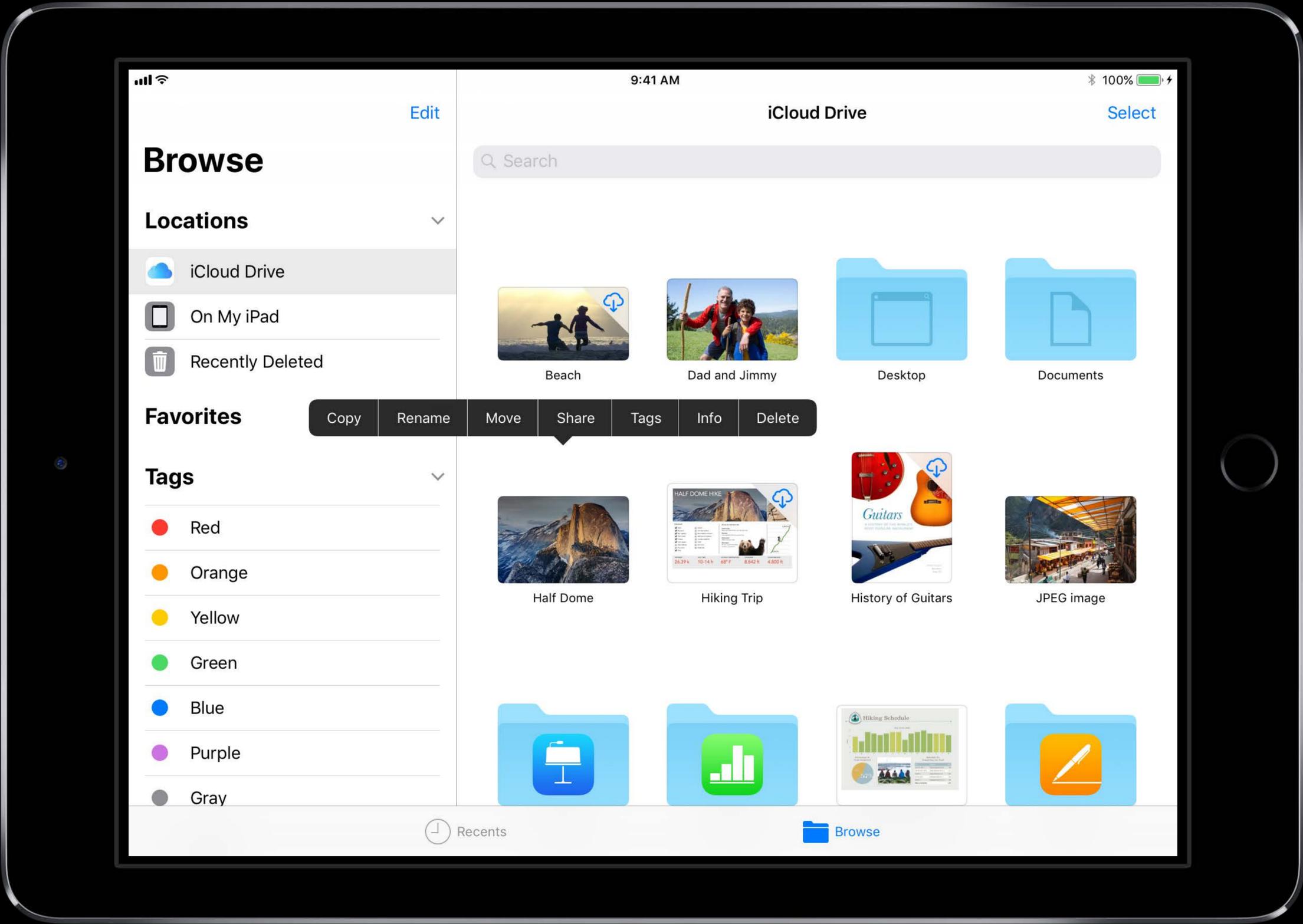
Custom actions

Expose items services to third parties

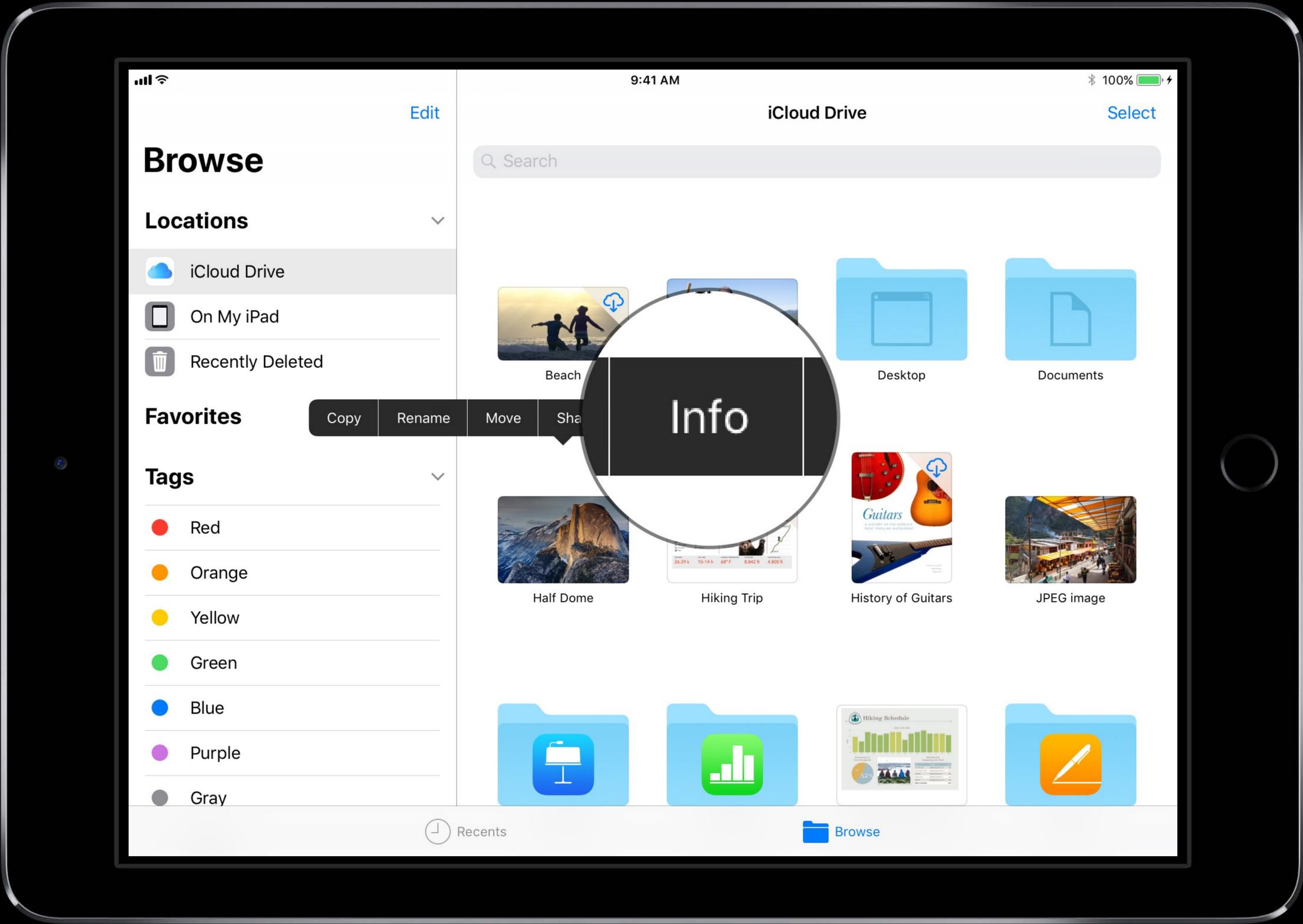
Modifying Items

Johannes Fortmann, Software Engineer

NEW



NEW



NEW



NEW



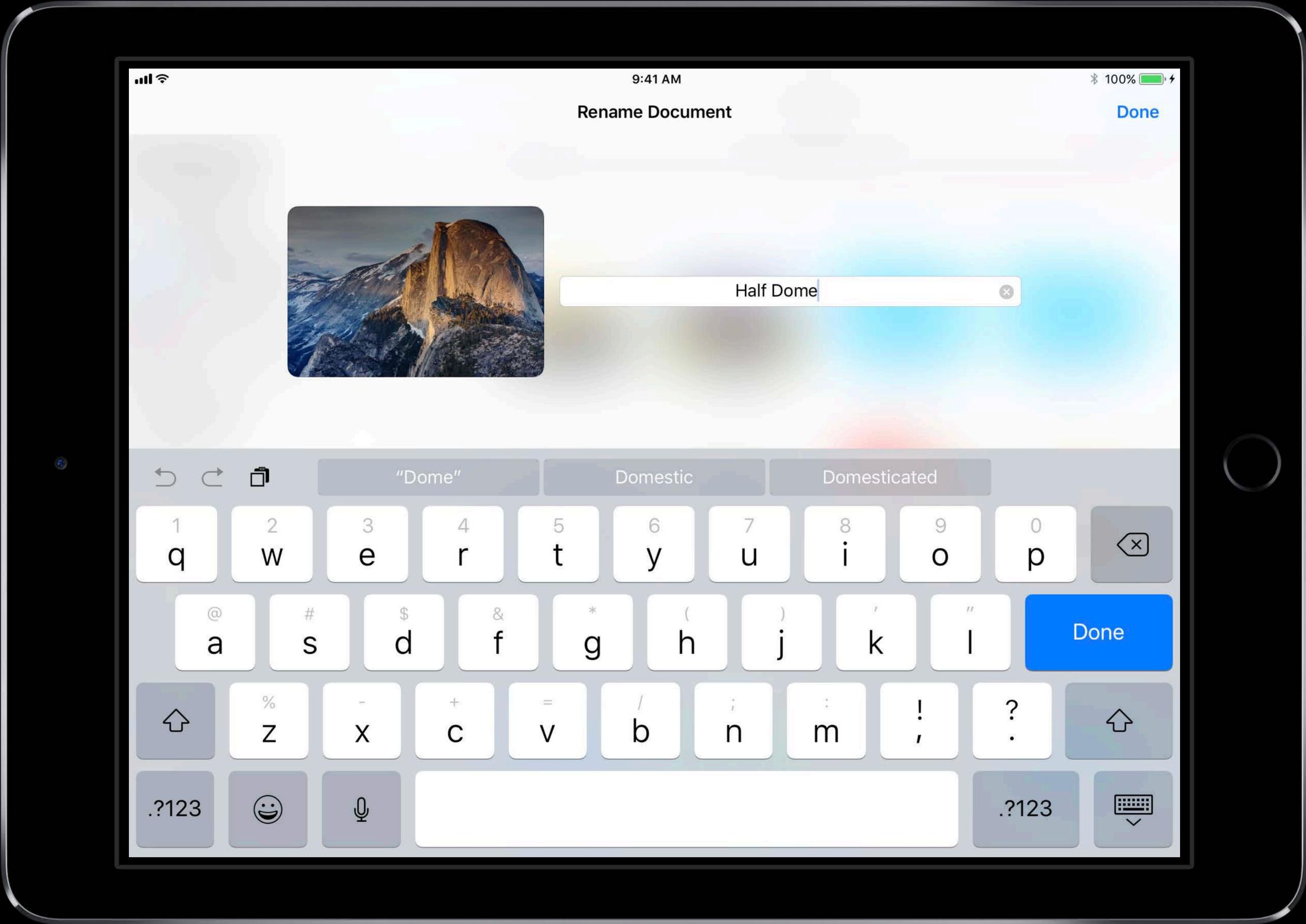
NEW



NEW



NEW



Modifying Items

Modifications on the structure

Modifying Items

Modifications on the structure

Wide variety of operations

Modifying Items

Modifications on the structure

Wide variety of operations

- Move

Modifying Items

Modifications on the structure

Wide variety of operations

- Move
- Create

Modifying Items

Modifications on the structure

Wide variety of operations

- Move
- Create
- Modify attributes

Modifying Items

Modifications on the structure

Wide variety of operations

- Move
- Create
- Modify attributes

Back user operations

Modifying Items

Walk through an action

```
func importDocument(at FileURL:, toParentItemIdentifier parent:, completionHandler:)
```

Modifying Items

Walk through an action

The system gives you a file (URL)

```
func importDocument(at FileURL:, toParentItemIdentifier parent:, completionHandler:)
```

Modifying Items

Walk through an action

The system gives you a file (URL)

You import it (= move it into your container)

```
func importDocument(at FileURL:, toParentItemIdentifier parent:, completionHandler:)
```

Modifying Items

Walk through an action

The system gives you a file (URL)

You import it (= move it into your container)

Schedule a background upload

```
func importDocument(at FileURL:, toParentItemIdentifier parent:, completionHandler:)
```

Modifying Items

Walk through an action

The system gives you a file (URL)

You import it (= move it into your container)

Schedule a background upload

Call the completion handler

```
func importDocument(at FileURL:, toParentItemIdentifier parent:, completionHandler:)
```

Modifying Items

Walk through an action

The system gives you a file (URL)

You import it (= move it into your container)

Schedule a background upload

Call the completion handler

Upload the file in the background

```
func importDocument(at FileURL:, toParentItemIdentifier parent:, completionHandler:)
```

Modifying Items

Walk through an action

UIDocumentBrowser or



Files

Create/Paste/Drop a file

File Provider Extension

NSURLSession

Modifying Items

Walk through an action

UIDocumentBrowser or



Files

Create/Paste/Drop a file



File Provider Extension

```
importDocument(at fileURL:, toParent  
itemIdentifier:, completionHandler:)
```

NSURLSession

Modifying Items

Walk through an action

UIDocumentBrowser or



Files

Create/Paste/Drop a file

File Provider Extension

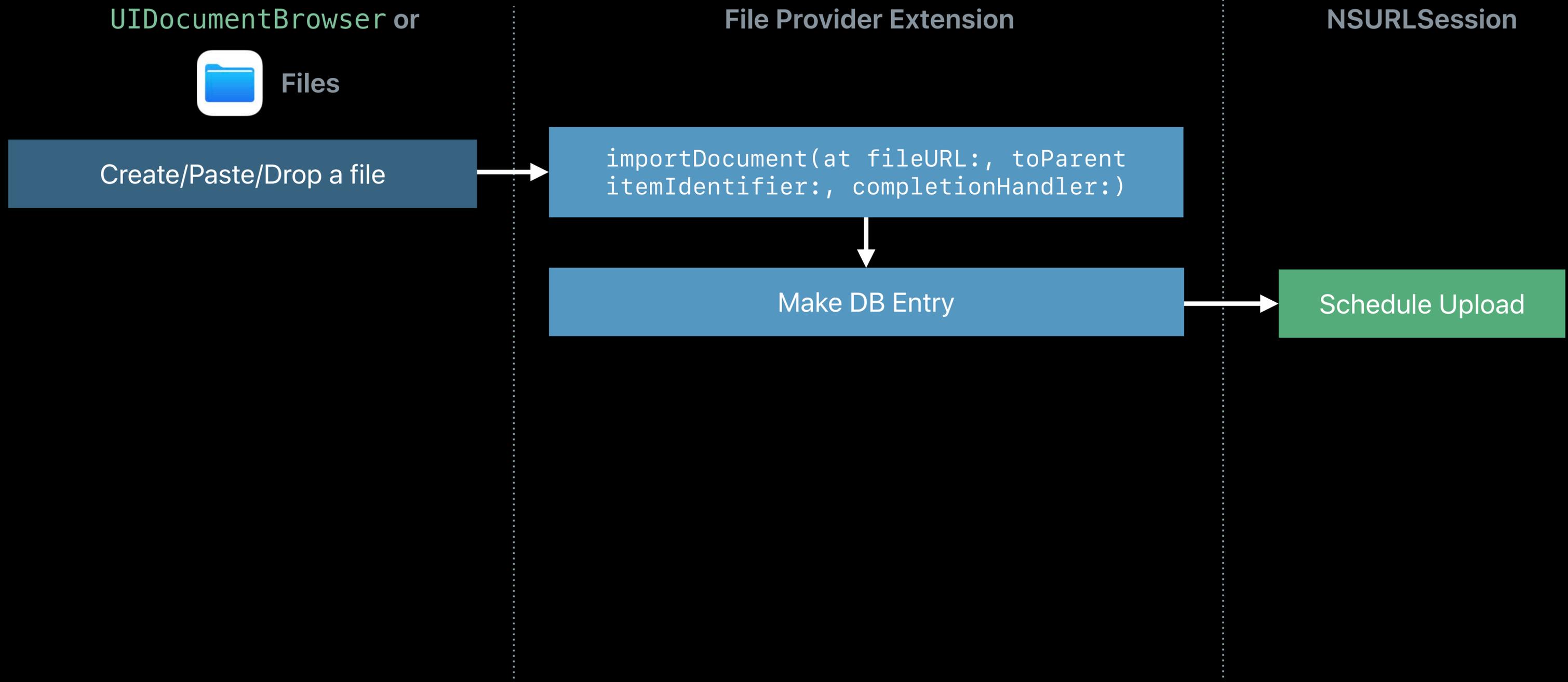
```
importDocument(at fileURL:, toParent  
itemIdentifier:, completionHandler:)
```

Make DB Entry

NSURLSession

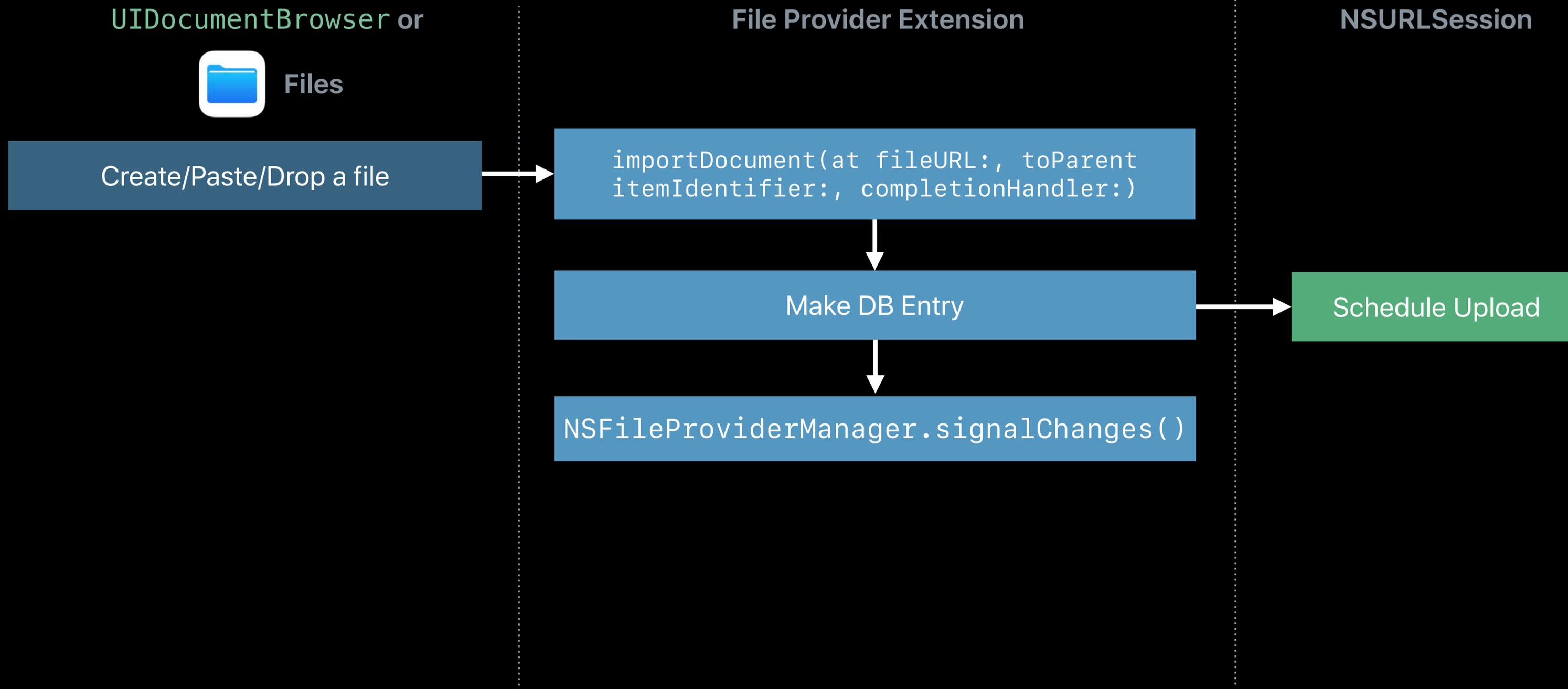
Modifying Items

Walk through an action



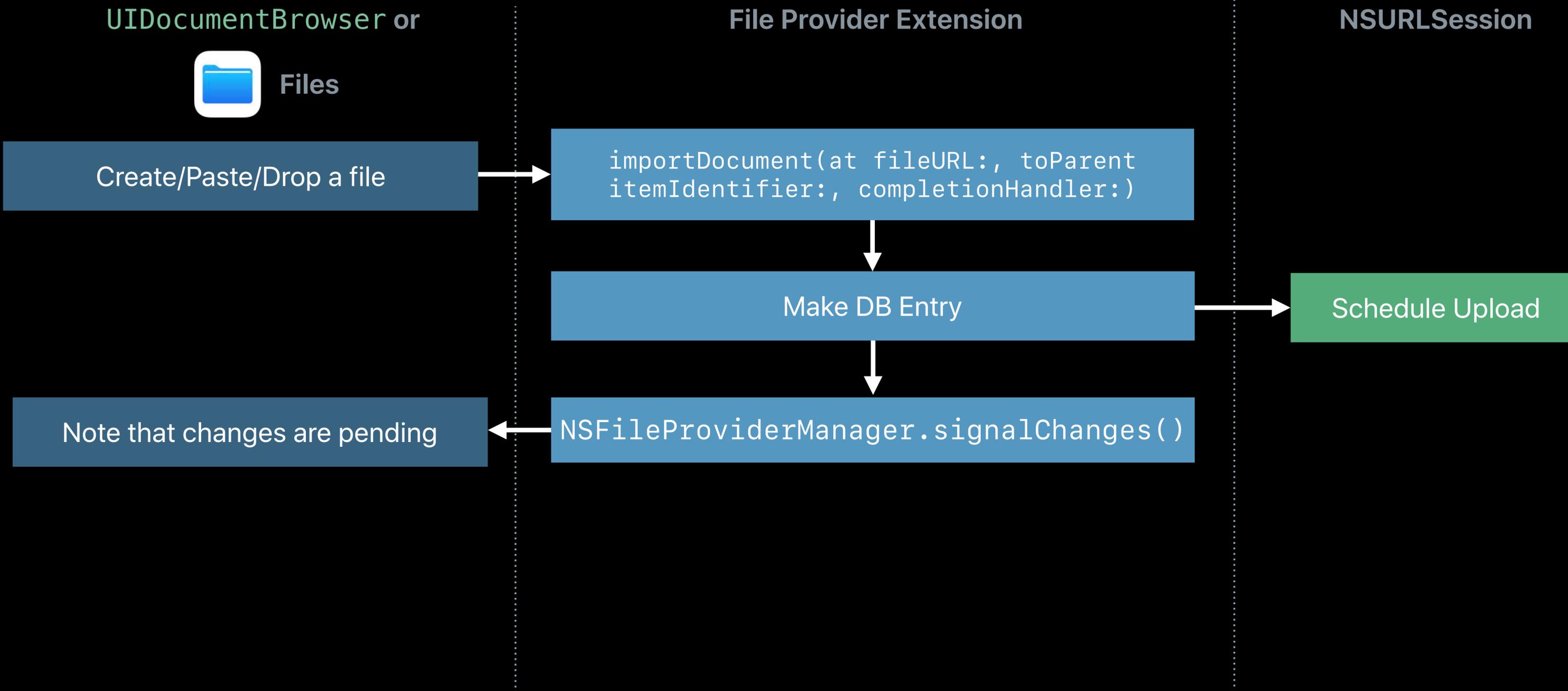
Modifying Items

Walk through an action



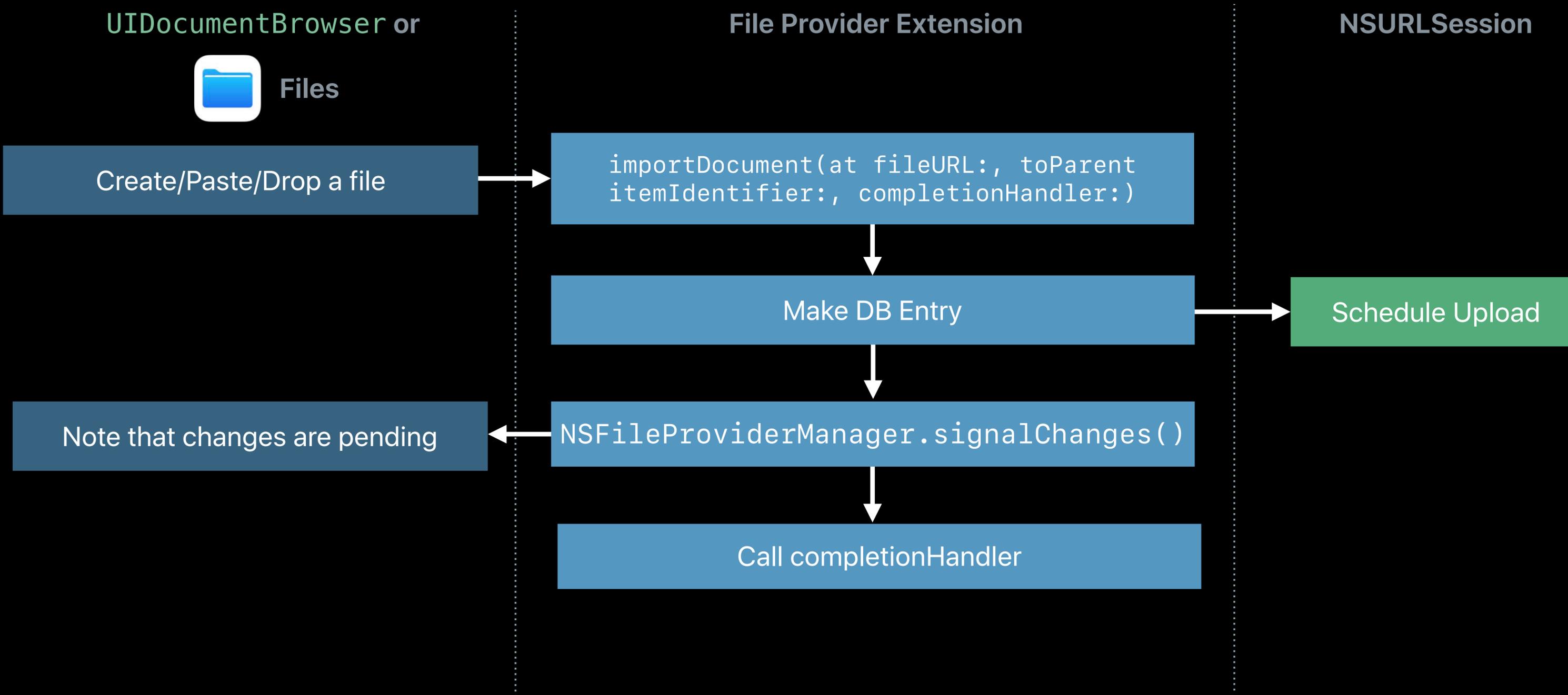
Modifying Items

Walk through an action



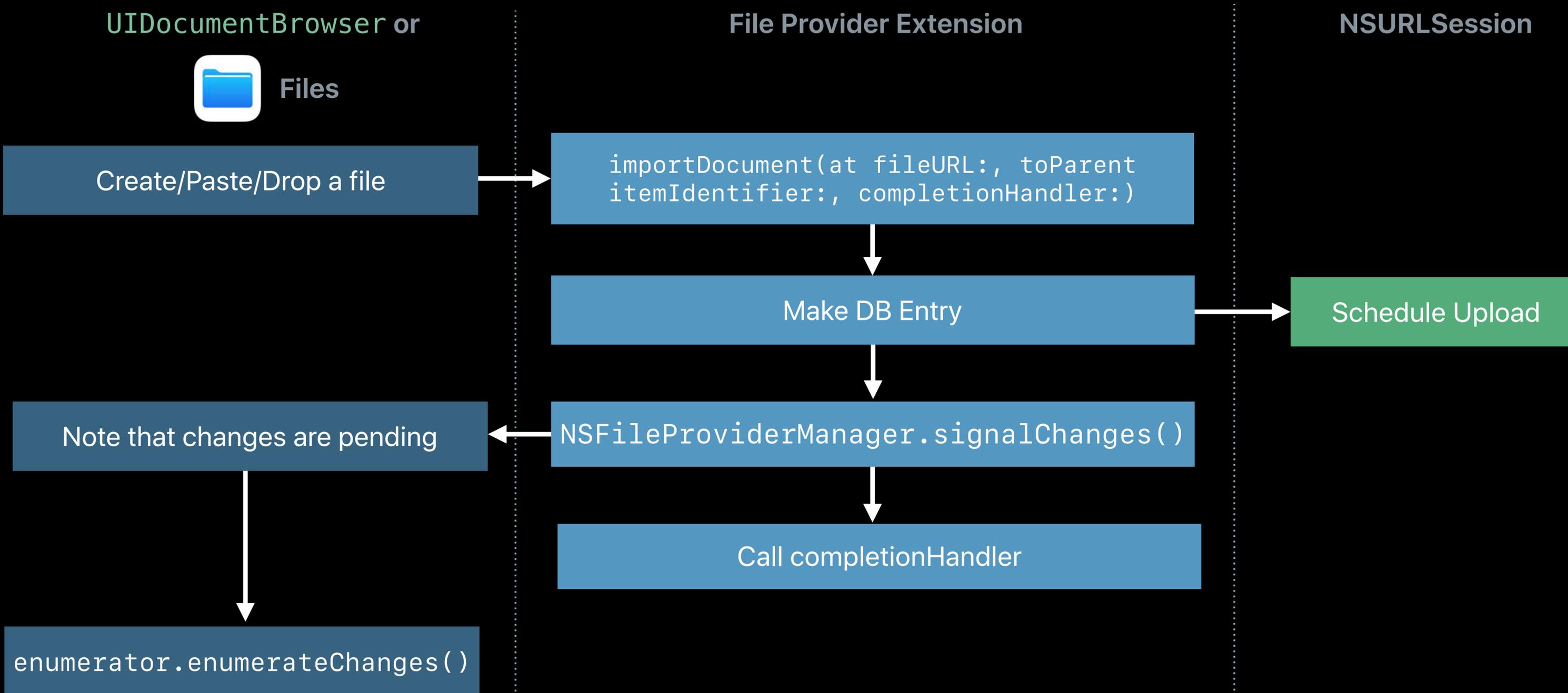
Modifying Items

Walk through an action



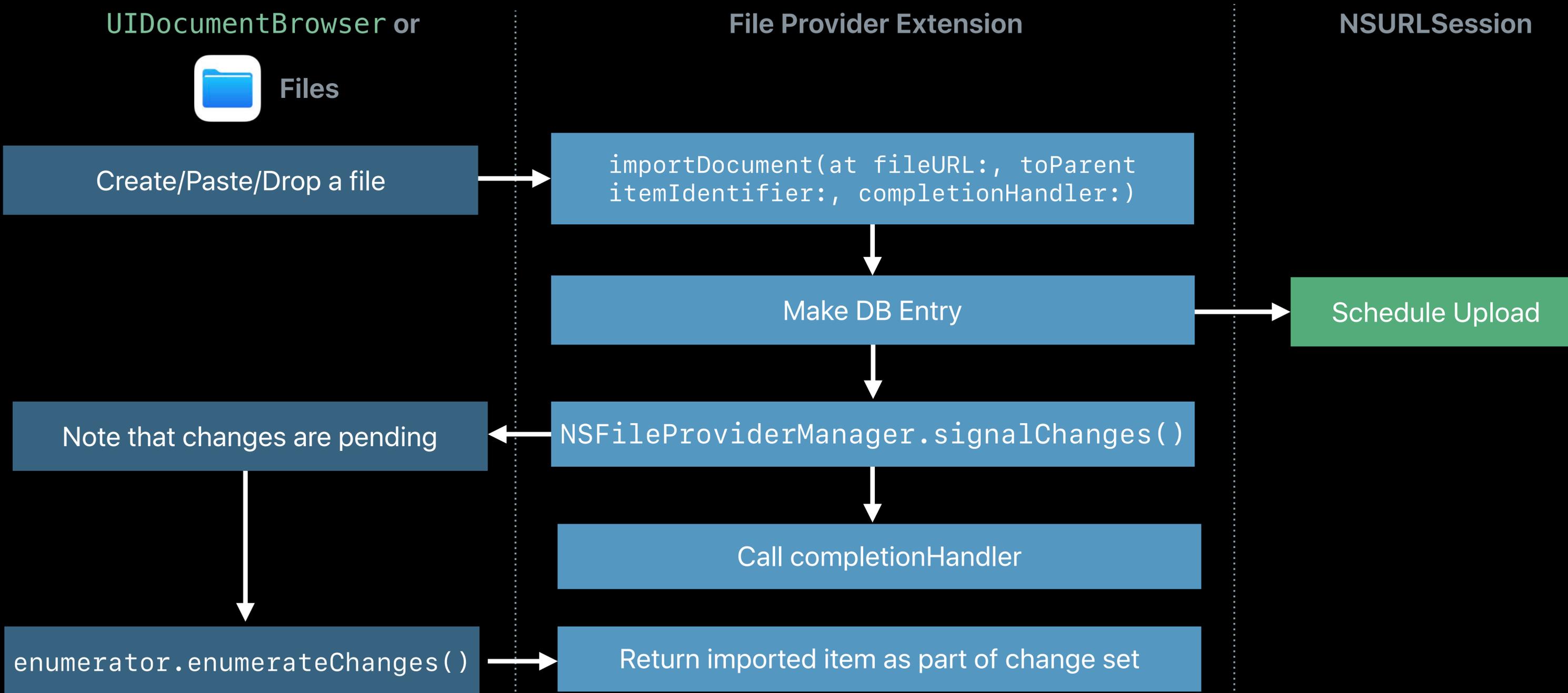
Modifying Items

Walk through an action



Modifying Items

Walk through an action



Actions

State management

Operations are expected to finish immediately

Defer long-running tasks

- Enqueue background uploads

Modifying Items

Successful action resolution

UIDocumentBrowser or



Files

File Provider
Extension

Parent Application

NSURLSession

Upload Succeeded

Modifying Items

Successful action resolution

UIDocumentBrowser or



Files

File Provider
Extension

Parent Application

NSURLSession

Update DB Entry

Upload Succeeded



Modifying Items

Successful action resolution

UIDocumentBrowser or

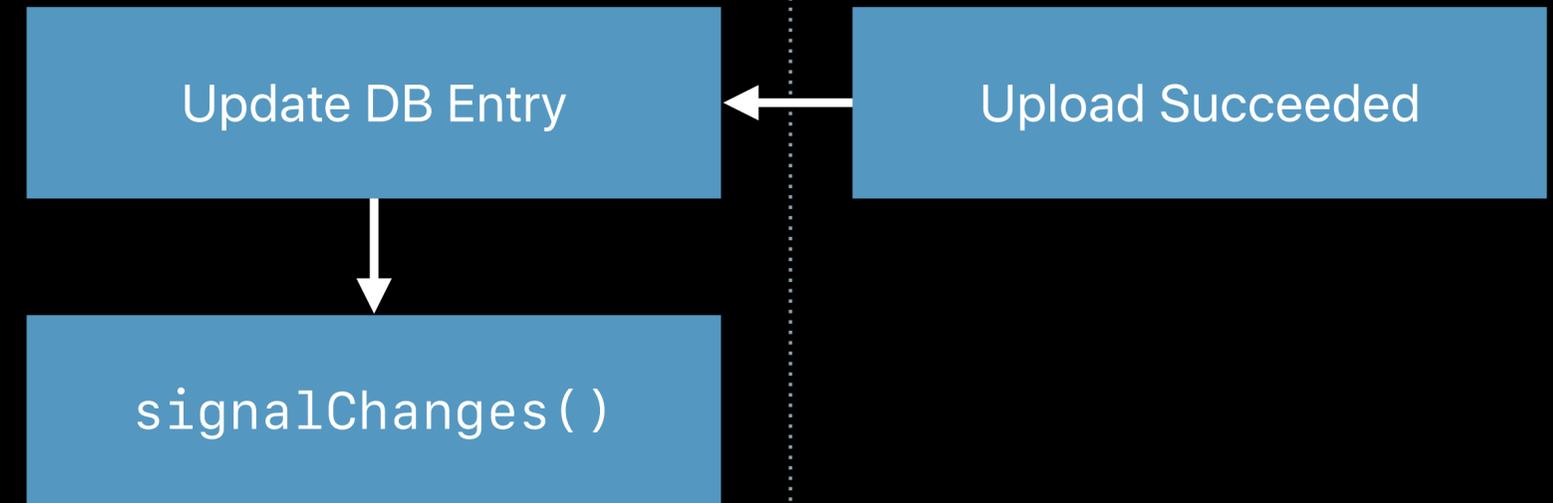


Files

File Provider
Extension

Parent Application

NSURLSession



Modifying Items

Successful action resolution

UIDocumentBrowser or



Files

File Provider
Extension

Parent Application

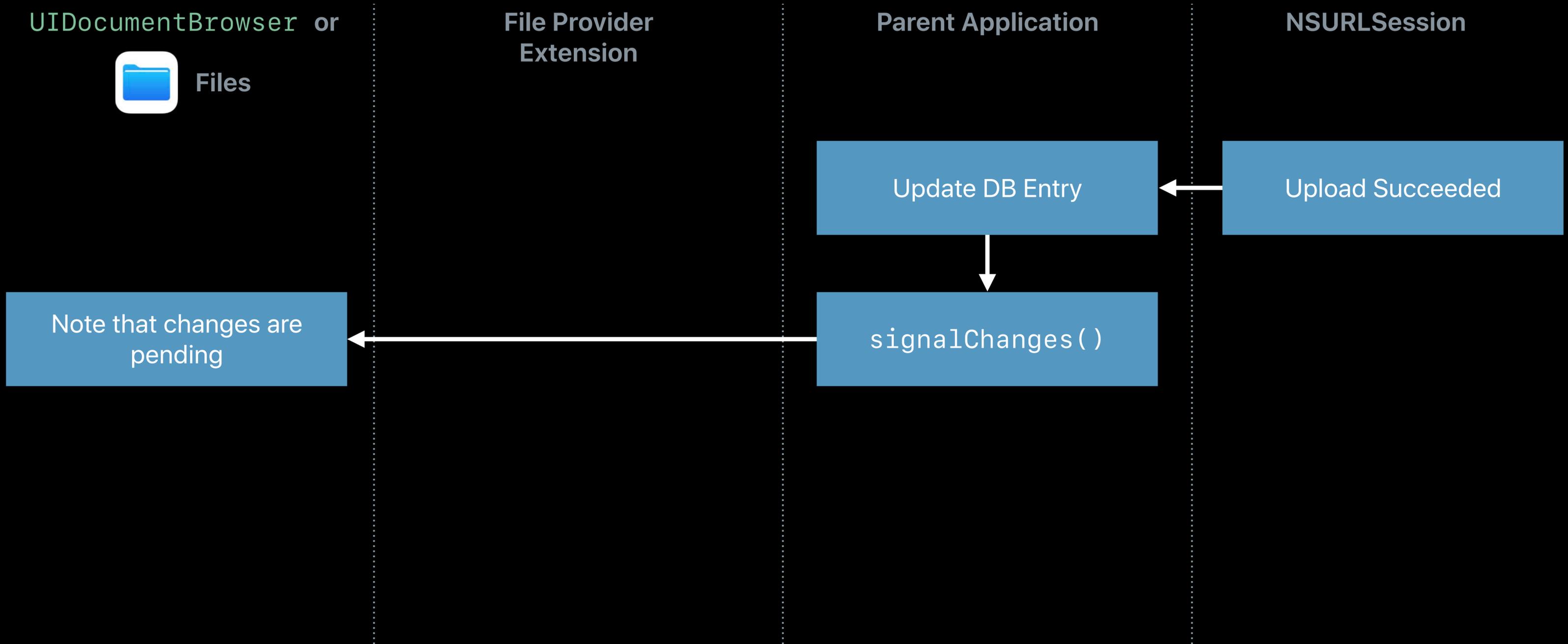
NSURLSession

Update DB Entry

Upload Succeeded

Note that changes are
pending

signalChanges()



Modifying Items

Successful action resolution

UIDocumentBrowser or



Files

File Provider
Extension

Parent Application

NSURLSession

Note that changes are
pending



Re-enumerate

Update DB Entry



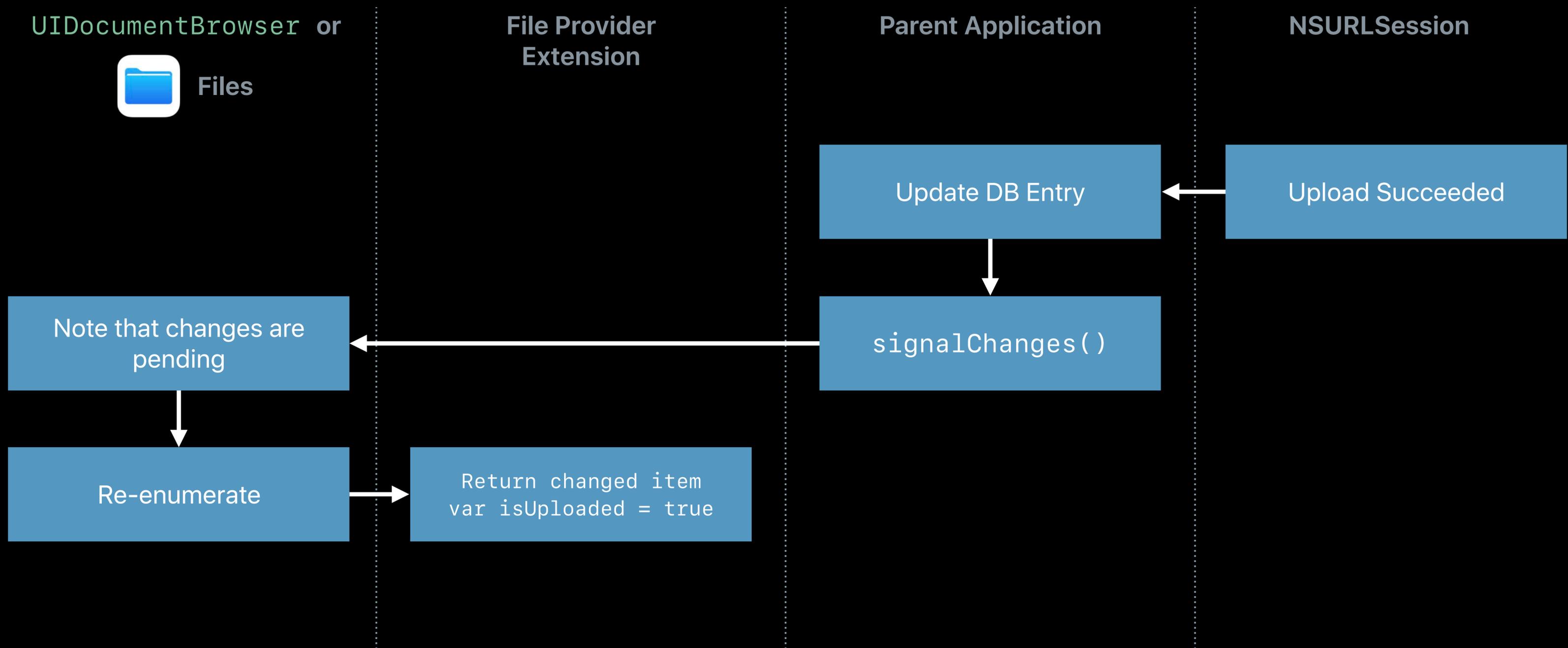
signalChanges()

Upload Succeeded



Modifying Items

Successful action resolution



Modifying Items

Error case

UIDocumentBrowser or



Files

File Provider
Extension

Parent Application

NSURLSession

Upload Failed

Modifying Items

Error case

UIDocumentBrowser or



Files

File Provider
Extension

Parent Application

NSURLSession

Update DB Entry

Upload Failed



Modifying Items

Error case

UIDocumentBrowser or

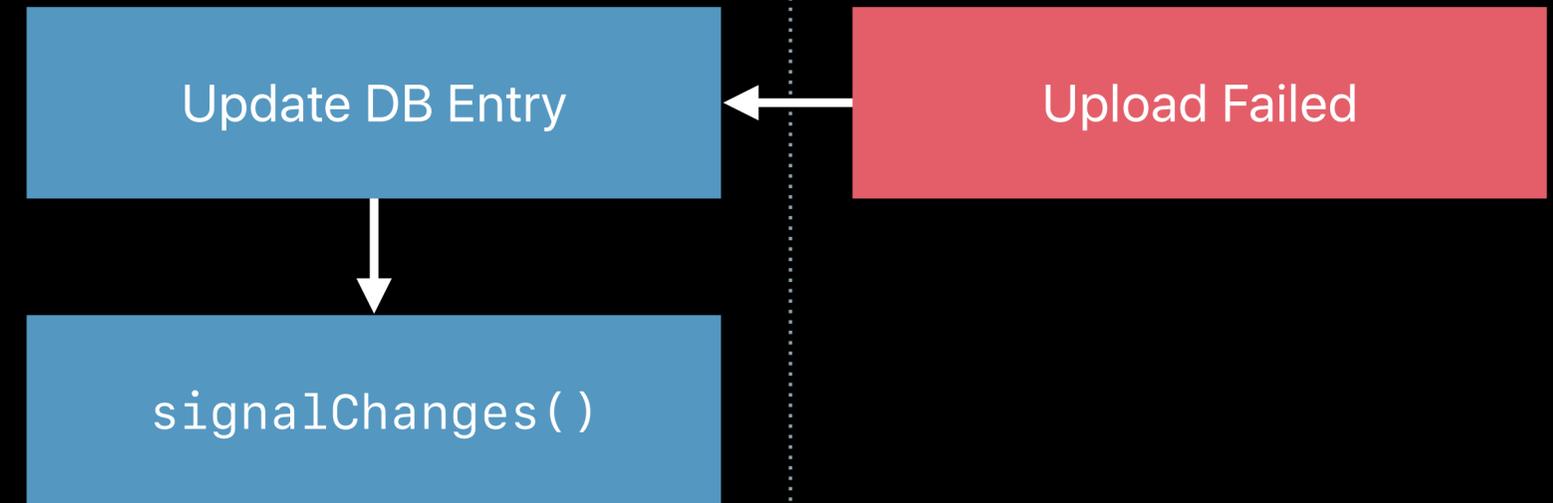


Files

File Provider
Extension

Parent Application

NSURLSession



Modifying Items

Error case

UIDocumentBrowser or



Files

File Provider
Extension

Parent Application

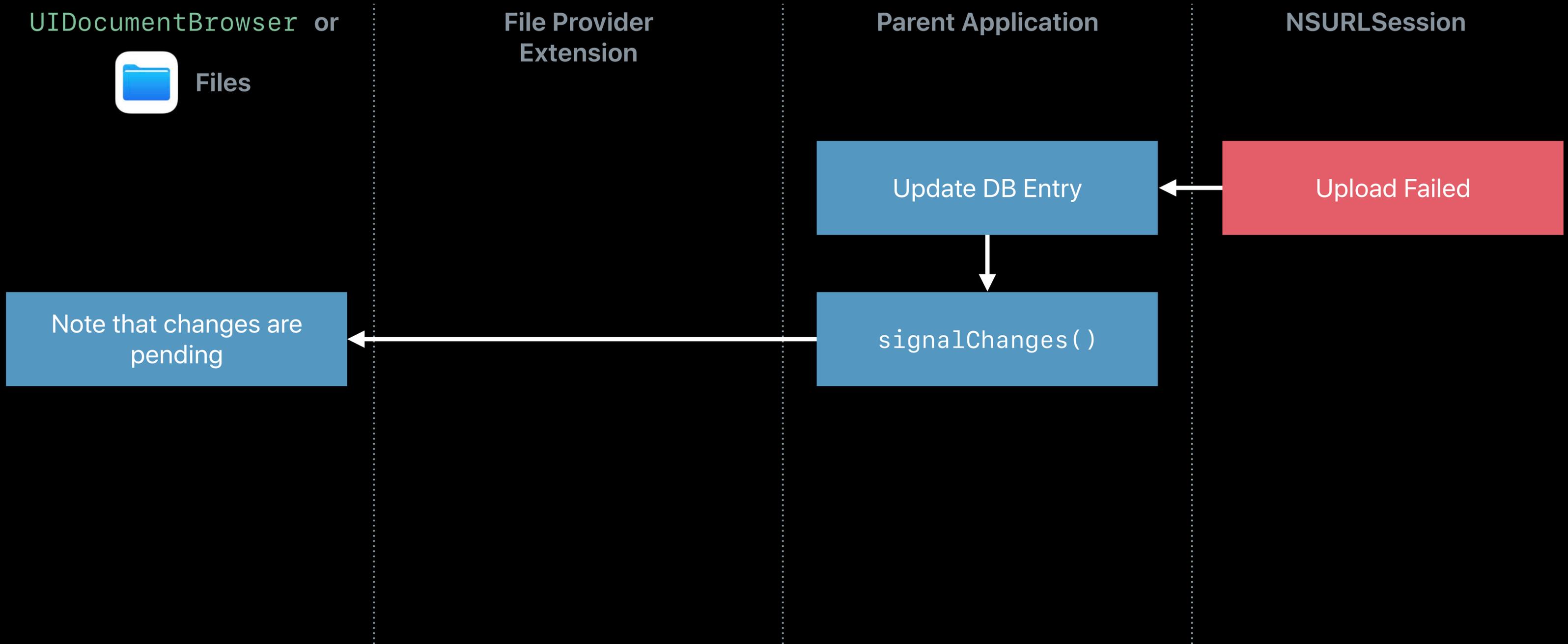
NSURLSession

Update DB Entry

Upload Failed

Note that changes are
pending

signalChanges()



Modifying Items

Error case

UIDocumentBrowser or



Files

File Provider
Extension

Parent Application

NSURLSession

Note that changes are
pending



Re-enumerate

Update DB Entry



signalChanges()

Upload Failed



Modifying Items

Error case

UIDocumentBrowser or

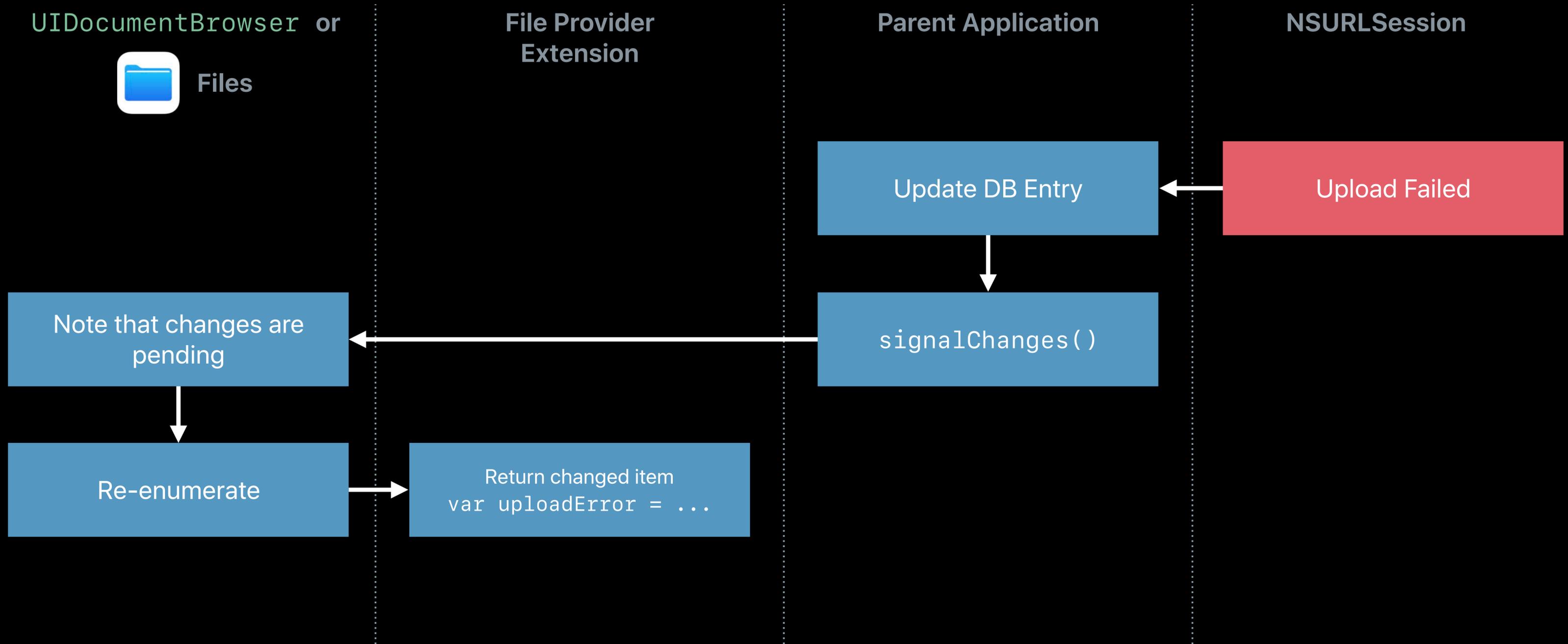


Files

File Provider
Extension

Parent Application

NSURLSession



Actions

Error handling

Actions

Error handling

Mark item status via error properties

```
var uploadError: Error?
```

Actions

Error handling

Mark item status via error properties

```
var uploadError: Error?
```

Signal re-enumeration

Actions

Error handling

Mark item status via error properties

```
var uploadError: Error?
```

Signal re-enumeration

Recovery depends on cause

Actions

Error handling

Mark item status via error properties

```
var uploadError: Error?
```

Signal re-enumeration

Recovery depends on cause

- Intermittent error—retry

Actions

Error handling

Mark item status via error properties

```
var uploadError: Error?
```

Signal re-enumeration

Recovery depends on cause

- Intermittent error—retry
- Persistent error—suggest solution in error description

Actions

Error handling

Mark item status via error properties

```
var uploadError: Error?
```

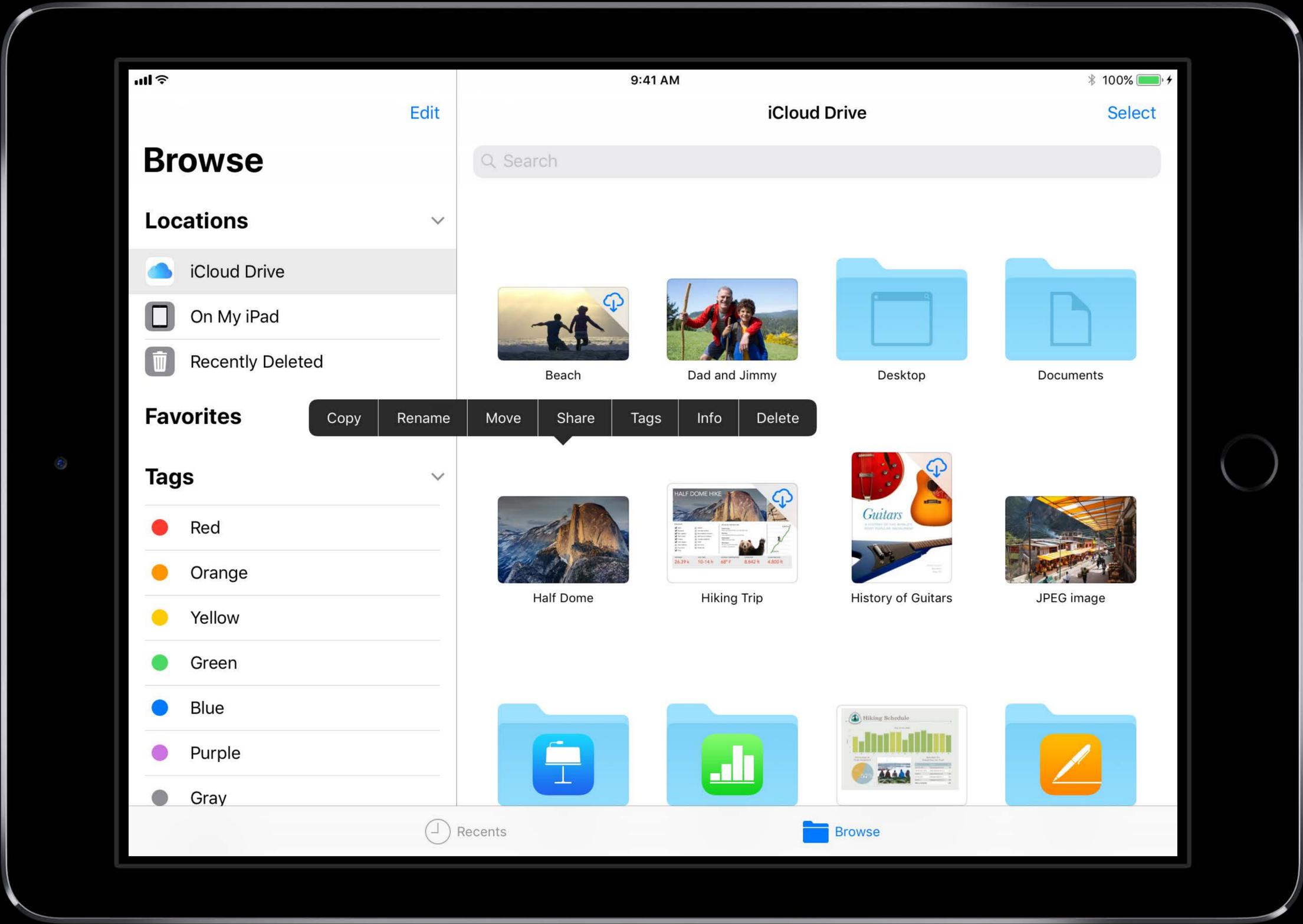
Signal re-enumeration

Recovery depends on cause

- Intermittent error—retry
- Persistent error—suggest solution in error description
- Authentication error—custom action will be called by the system

Custom Actions

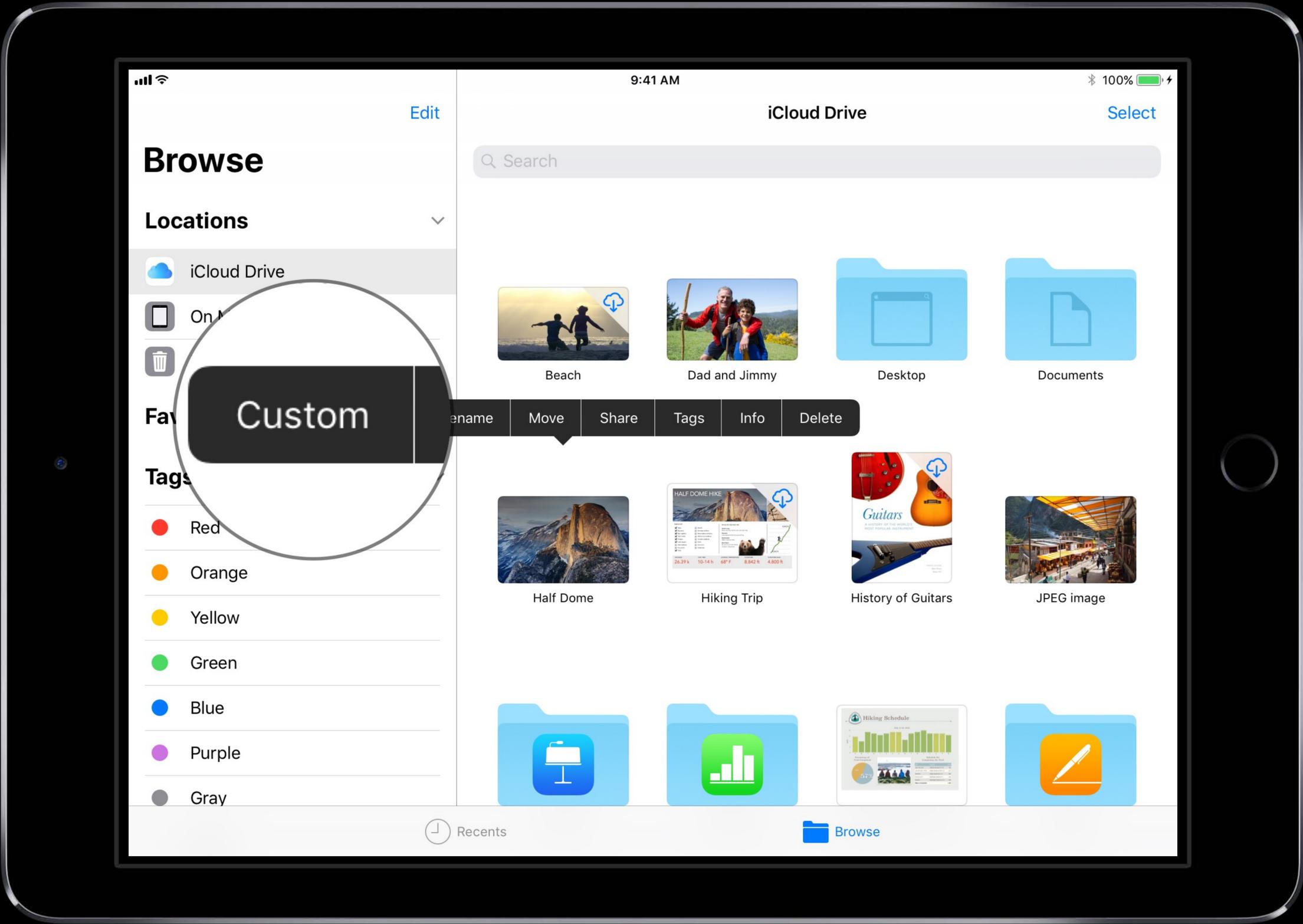
NEW



NEW

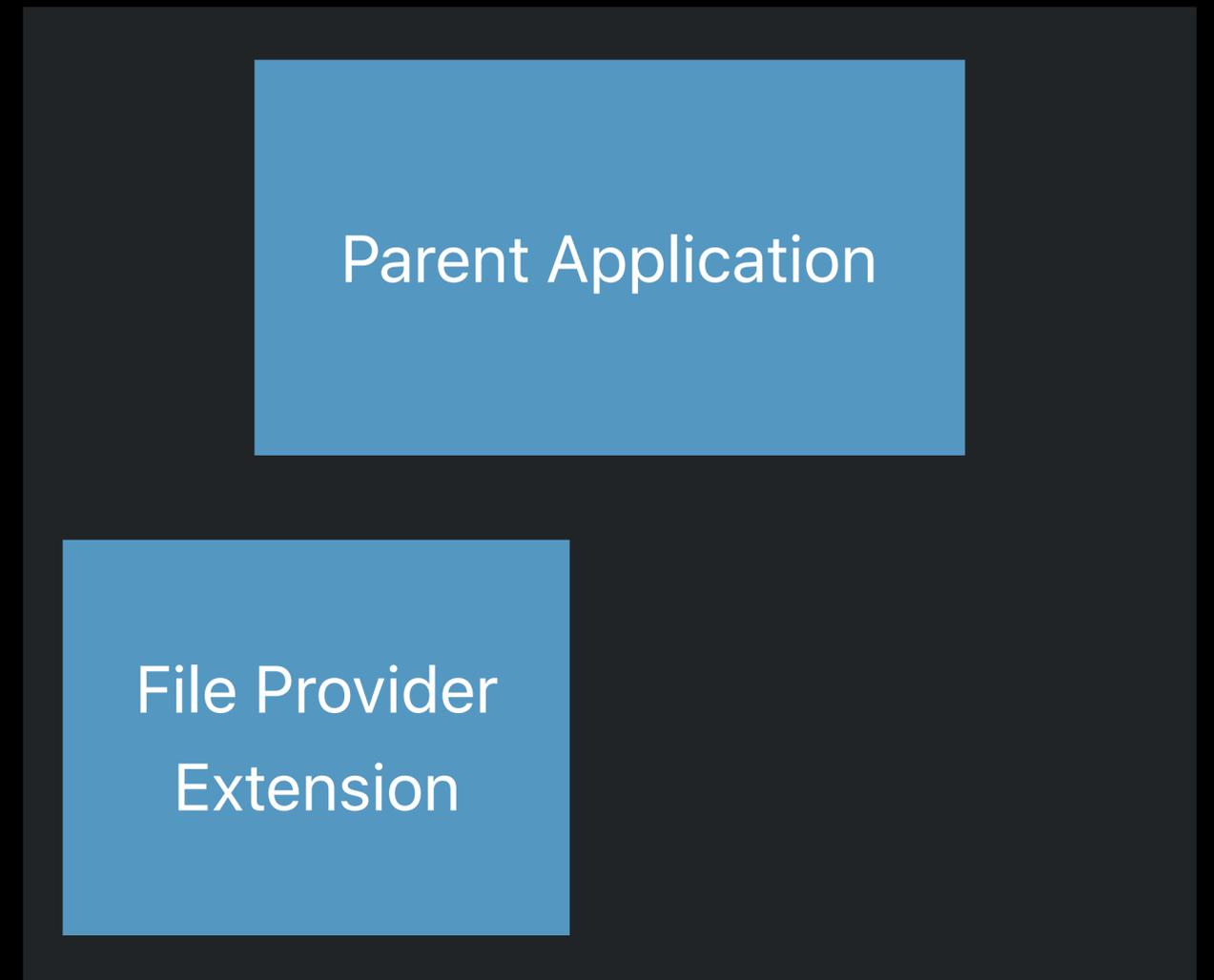


NEW



Custom actions

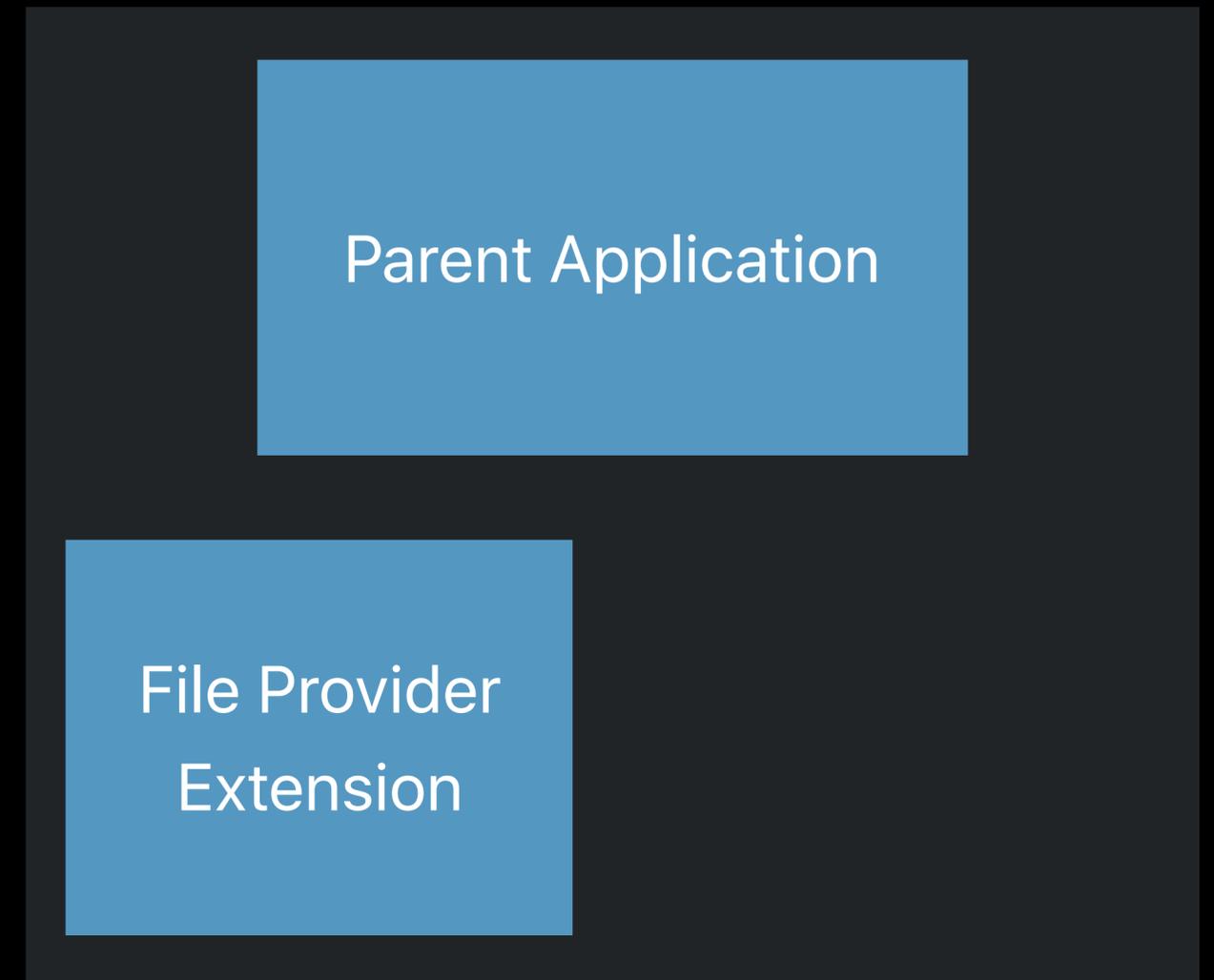
UI implementation



Custom actions

UI implementation

Operations are scoped to files from
your extension

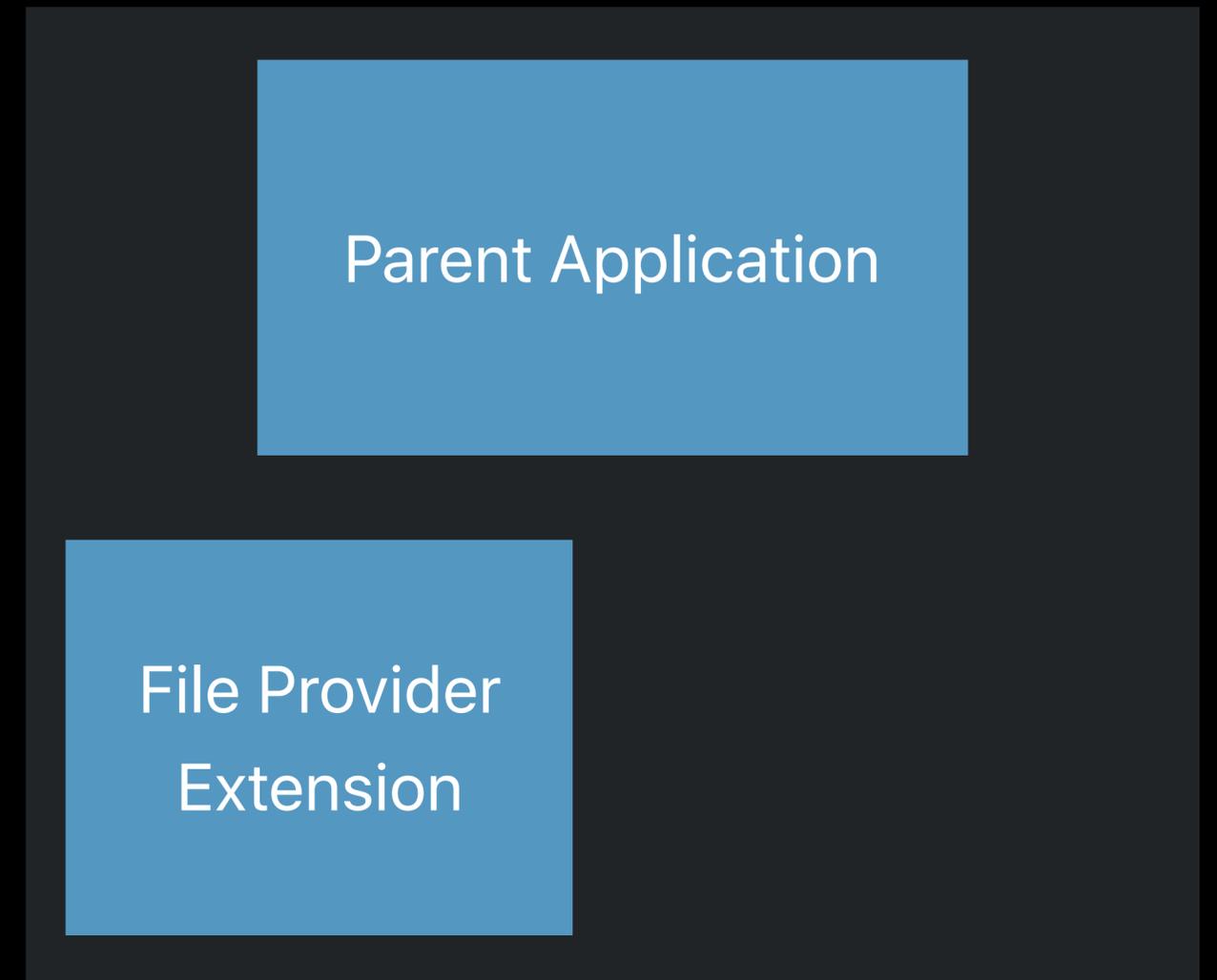


Custom actions

UI implementation

Operations are scoped to files from
your extension

Interface is provided by a UI extension



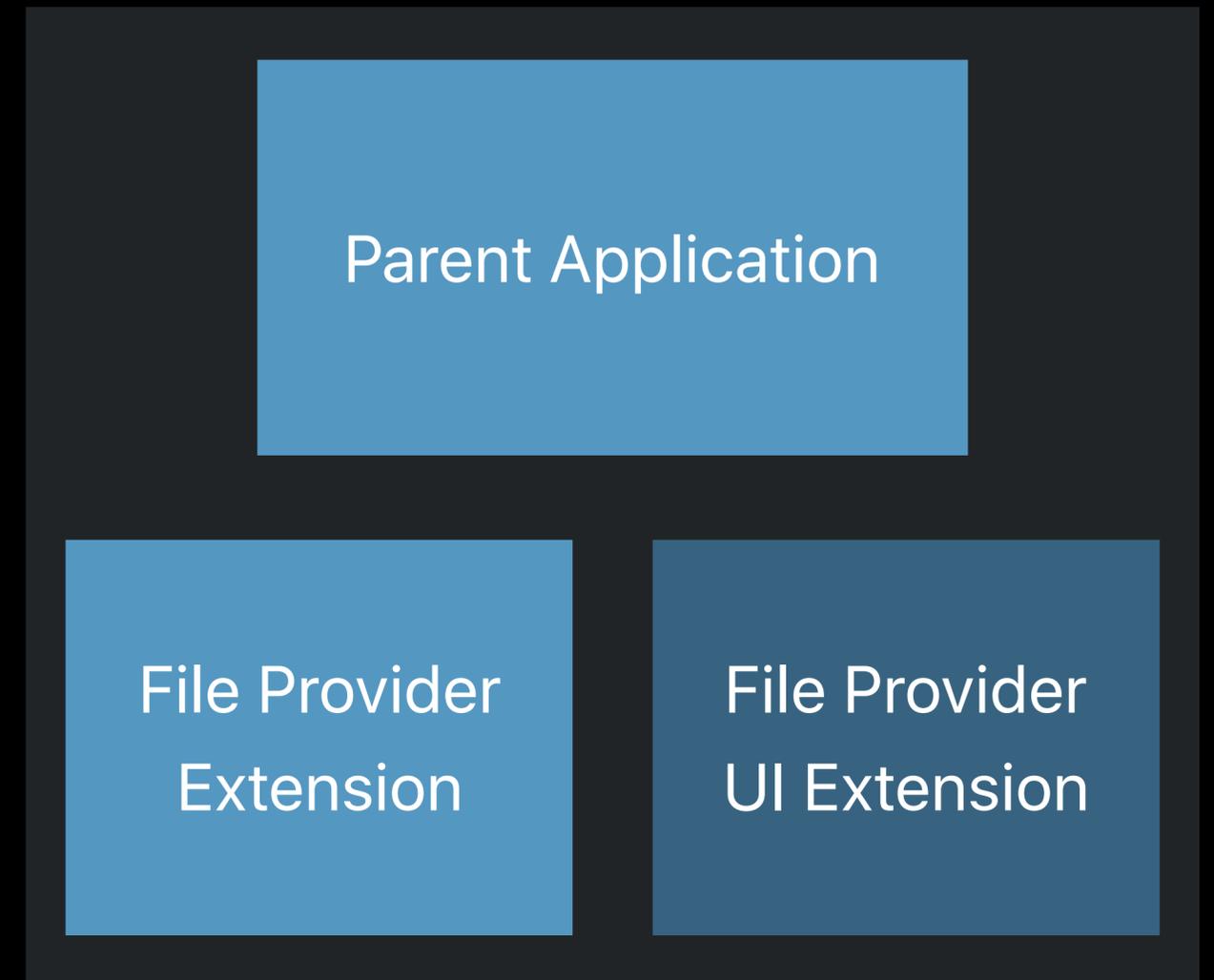
Custom actions

UI implementation

Operations are scoped to files from
your extension

Interface is provided by a UI extension

Subclass of
FPUActionExtensionViewController



Custom actions

Implementation details

Custom actions

Implementation details

Extension lists possible actions in Info.plist

▼ NSExtension	↕	Dictionary	(3 items)
NSExtensionMainStoryboard		String	MainInterface
NSExtensionPointIdentifier		String	com.apple.fileprovider-actionsui
▼ NSExtensionFileProviderActions		Array	(1 item)
▼ Item 0		Dictionary	(3 items)
NSExtensionFileProviderActionName	+ -	String	Comment
NSExtensionFileProviderActionActivationRule		String	TRUEPREDICATE
NSExtensionFileProviderActionIdentifier		String	com.example.commentAction

Custom actions

Implementation details

Extension lists possible actions in Info.plist

▼ NSExtension	↕	Dictionary	(3 items)
NSExtensionMainStoryboard		String	MainInterface
NSExtensionPointIdentifier		String	com.apple.fileprovider-actionsui
▼ NSExtensionFileProviderActions		Array	(1 item)
▼ Item 0		Dictionary	(3 items)
NSExtensionFileProviderActionName	+ -	String	Comment
NSExtensionFileProviderActionActivationRule		String	TRUEPREDICATE
NSExtensionFileProviderActionIdentifier		String	com.example.commentAction

Matching using predicate for activation rule

Modifying Items

Walk through an action

UIDocumentBrowser or



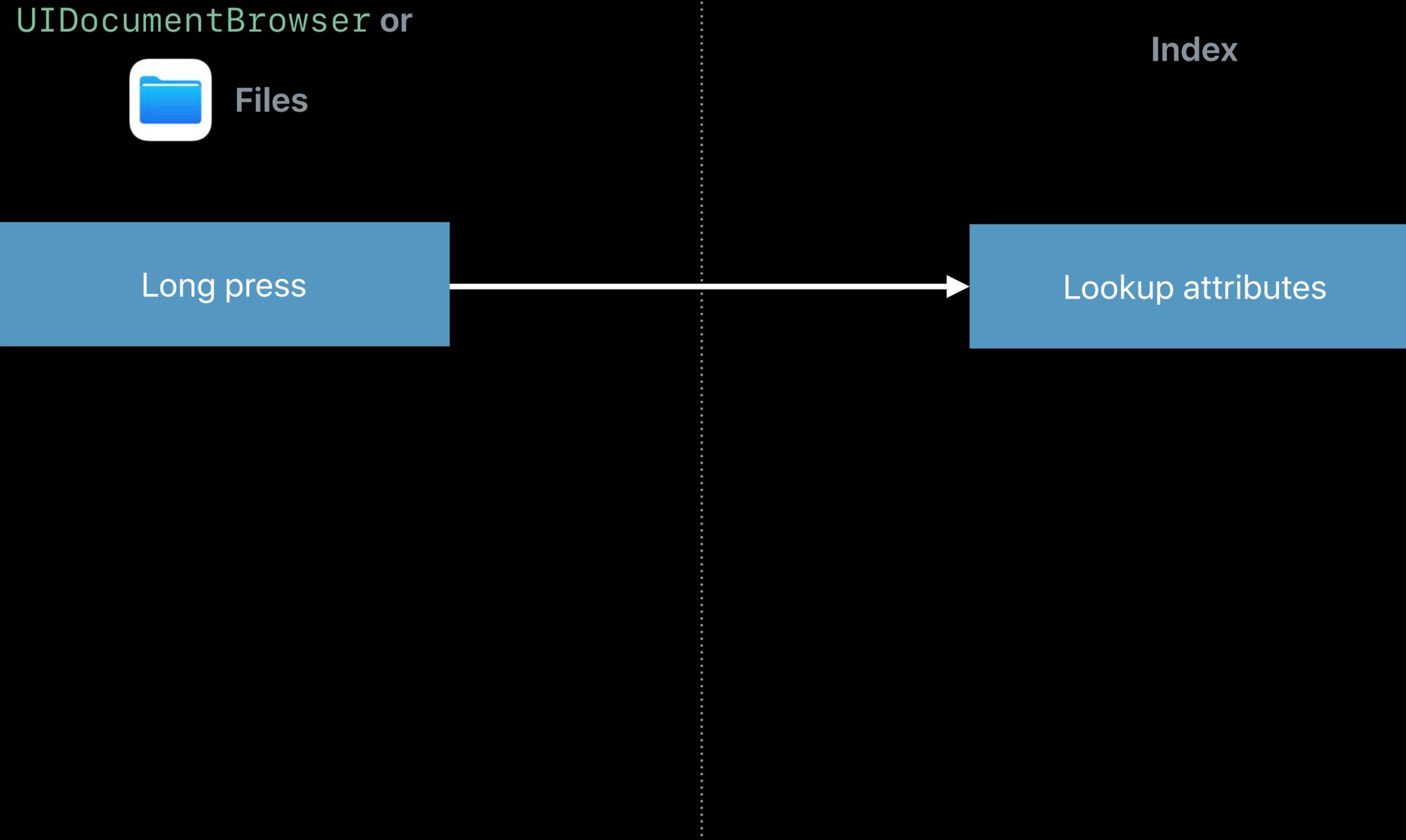
Files

Long press

Index

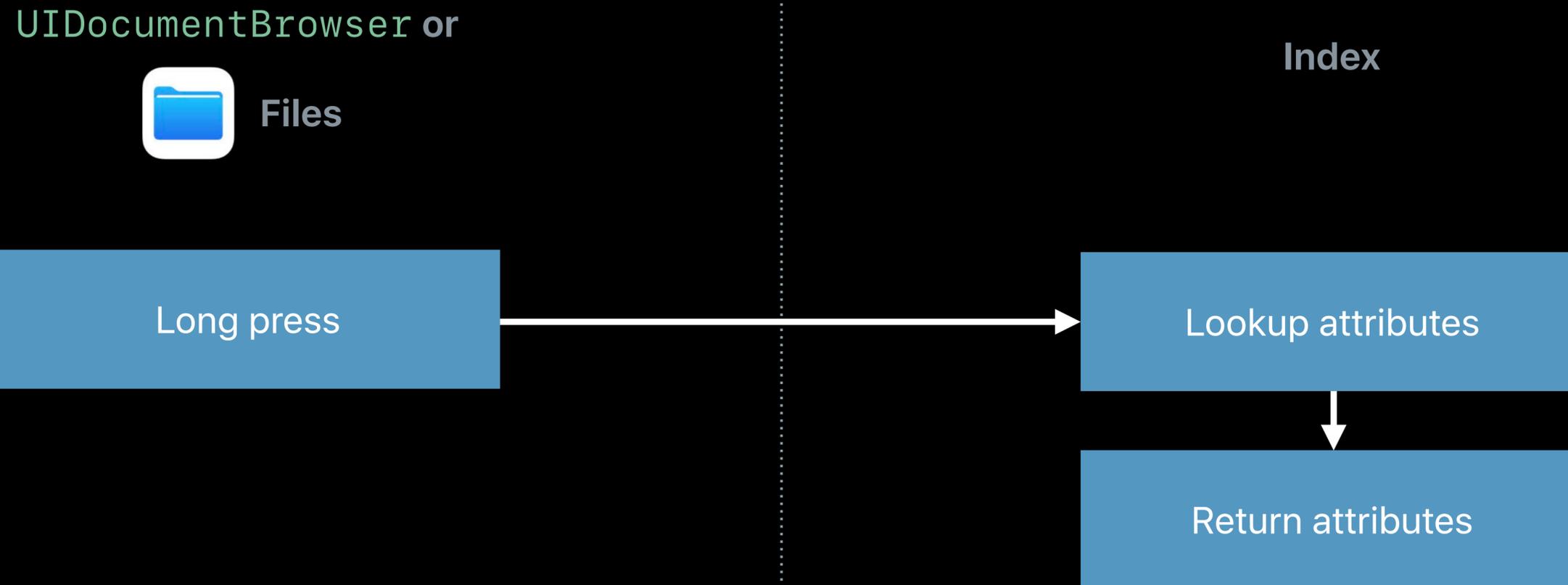
Modifying Items

Walk through an action



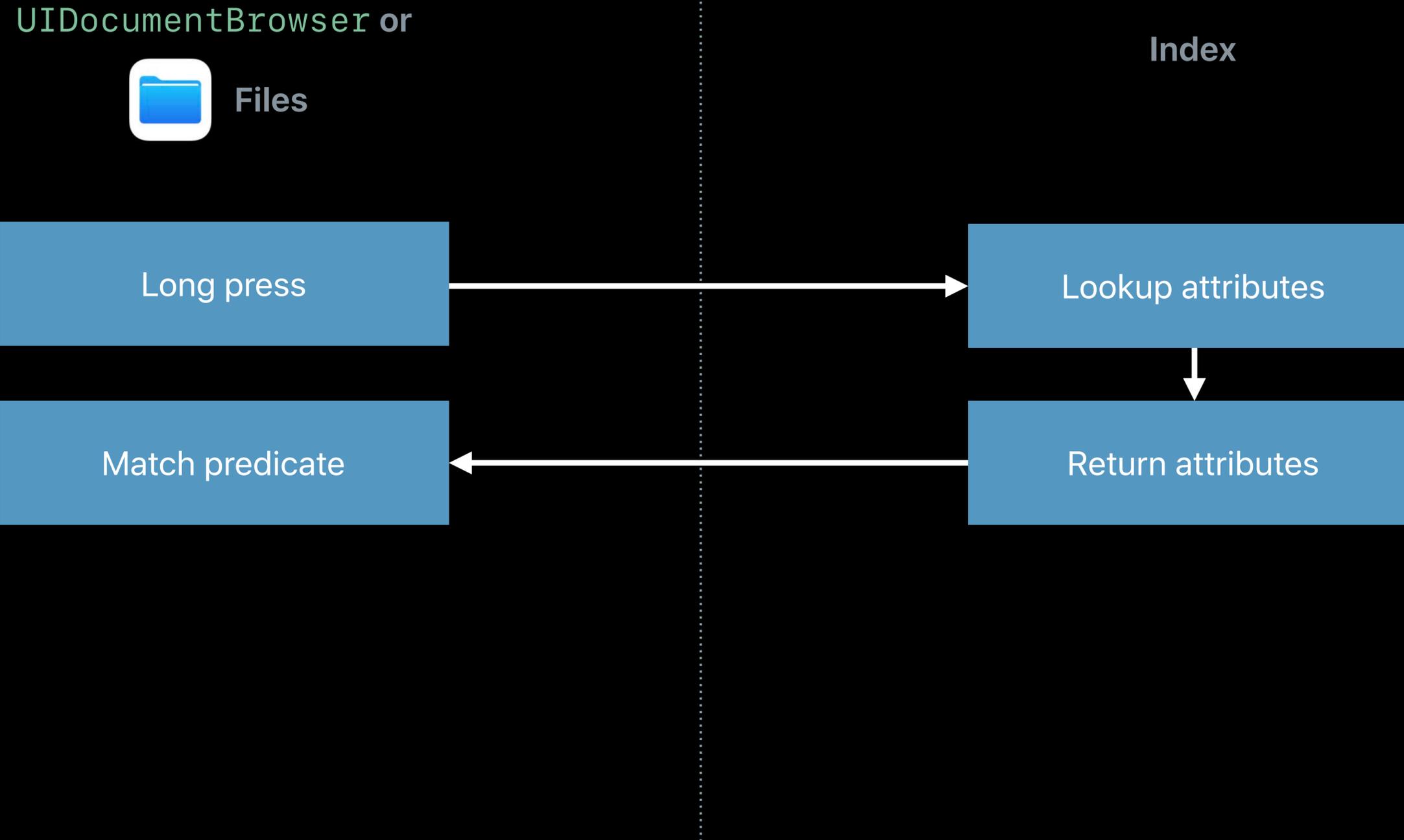
Modifying Items

Walk through an action



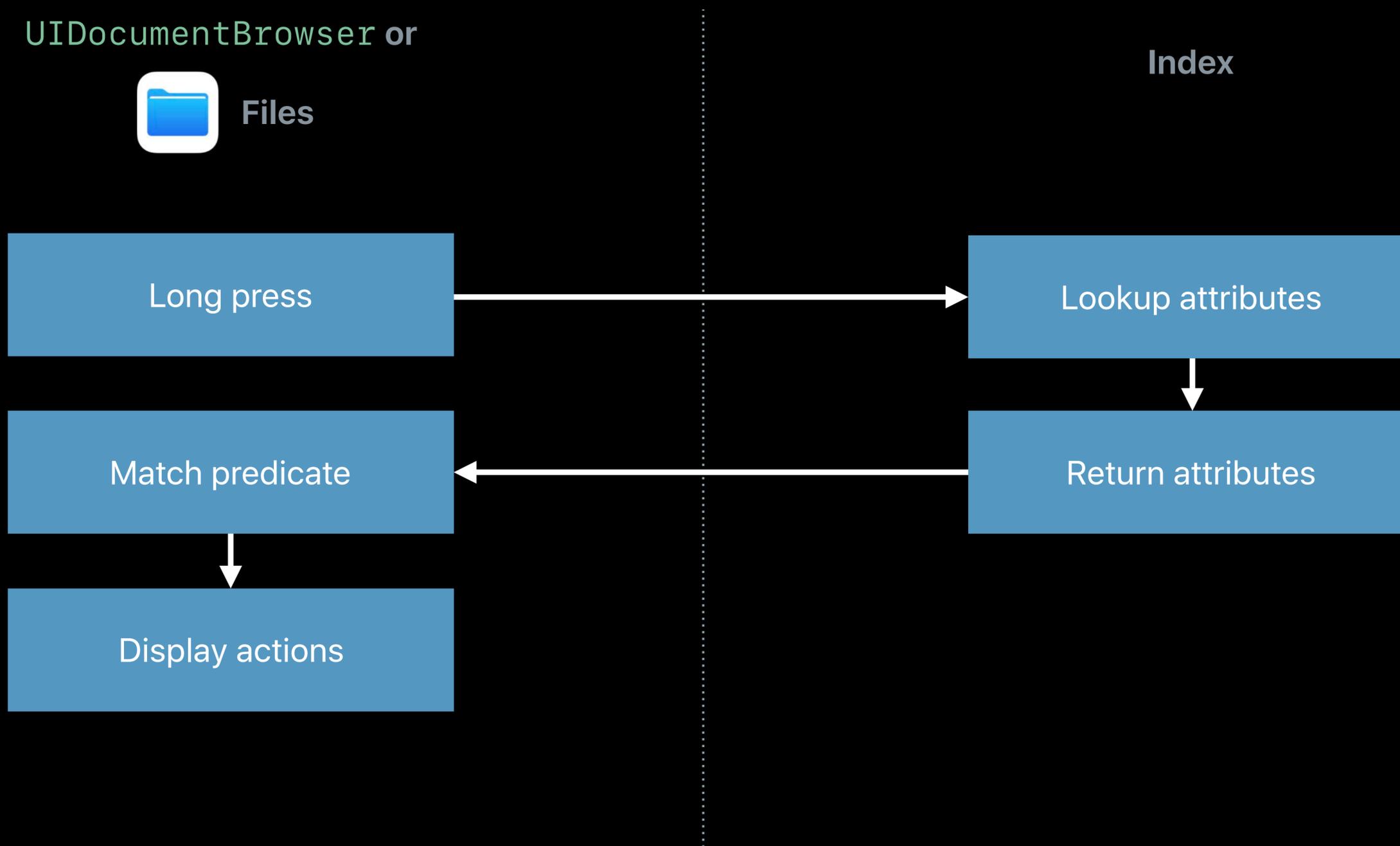
Modifying Items

Walk through an action



Modifying Items

Walk through an action



Modifying Items

Walk through an action

UIDocumentBrowser or

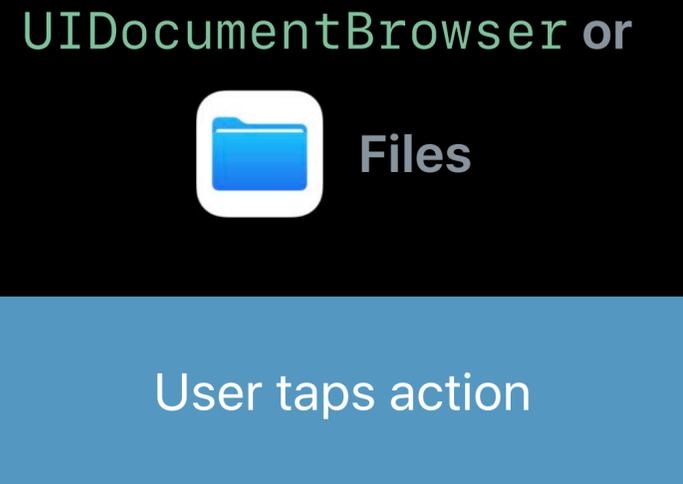


Files

File Provider UI Extension

Modifying Items

Walk through an action



File Provider UI Extension

Modifying Items

Walk through an action

UIDocumentBrowser or



Files

User taps action

File Provider UI Extension

```
override func prepareForAction(withIdentifier
actionIdentifier: String, itemIdentifiers:
[NSFileProviderItemIdentifier]) {
    // prepare UI for this action
}
```

Modifying Items

Walk through an action

UIDocumentBrowser or



Files

User taps action

Present UI

File Provider UI Extension

```
override func prepareForAction(withIdentifier  
actionIdentifier: String, itemIdentifiers:  
[NSFileProviderItemIdentifier]) {  
    // prepare UI for this action  
}
```

Modifying Items

Walk through an action

UIDocumentBrowser or



Files

User taps action

Present UI

File Provider UI Extension

```
override func prepareForAction(withIdentifier
actionIdentifier: String, itemIdentifiers:
[NSFileProviderItemIdentifier]) {
    // prepare UI for this action
}
```

```
func doneButtonPressed(_ sender: AnyObject?) {
    self.extensionContext.completeRequest()
}
```

Modifying Items

Walk through an action

UIDocumentBrowser or



Files

User taps action

Present UI

Dismiss UI

File Provider UI Extension

```
override func prepareForAction(withIdentifier  
actionIdentifier: String, itemIdentifiers:  
[NSFileProviderItemIdentifier]) {  
    // prepare UI for this action  
}
```

```
func doneButtonPressed(_ sender: AnyObject?) {  
    self.extensionContext.completeRequest()  
}
```

Custom Actions

Authentication action

Custom Actions

Authentication action

Called by the system in response to an authentication error

Custom Actions

Authentication action

Called by the system in response to an authentication error

Method on `FPUIActionExtensionViewController`

```
override func prepareForAuthentication() {  
    ...  
}
```

Custom Actions

Authentication action

Called by the system in response to an authentication error

Method on `FUIActionExtensionViewController`

```
override func prepareForAuthentication() {  
    ...  
}
```

Same as other actions otherwise

Services

Direct communication based on an item

Services

Services

Custom actions show up in the document browser interface

Services

Custom actions show up in the document browser interface

Sometimes, you need programmatic access

Services

Defining a service

Services

Defining a service

Services are defined by a name and a protocol

Services

Defining a service

Services are defined by a name and a protocol

This protocol must be known to you and the other developer using it

Services

Defining a service

Services are defined by a name and a protocol

This protocol must be known to you and the other developer using it

XPC rules—only secure codable parameters, return via completion block

Services

Defining a service

Services are defined by a name and a protocol

This protocol must be known to you and the other developer using it

XPC rules—only secure codable parameters, return via completion block

Services

Data flow

Third-Party Application

File Provider Extension

Services

Data flow

Third-Party Application

```
getFileProviderServicesForItem(at  
url:completionHandler:)
```

File Provider Extension

Services

Data flow

Third-Party Application

```
getFileProviderServicesForItem(at  
url:completionHandler:)
```



File Provider Extension

```
supportedServiceSources(for itemIdentifier:)
```

Services

Data flow

Third-Party Application

```
getFileProviderServicesForItem(at  
url:completionHandler:)
```



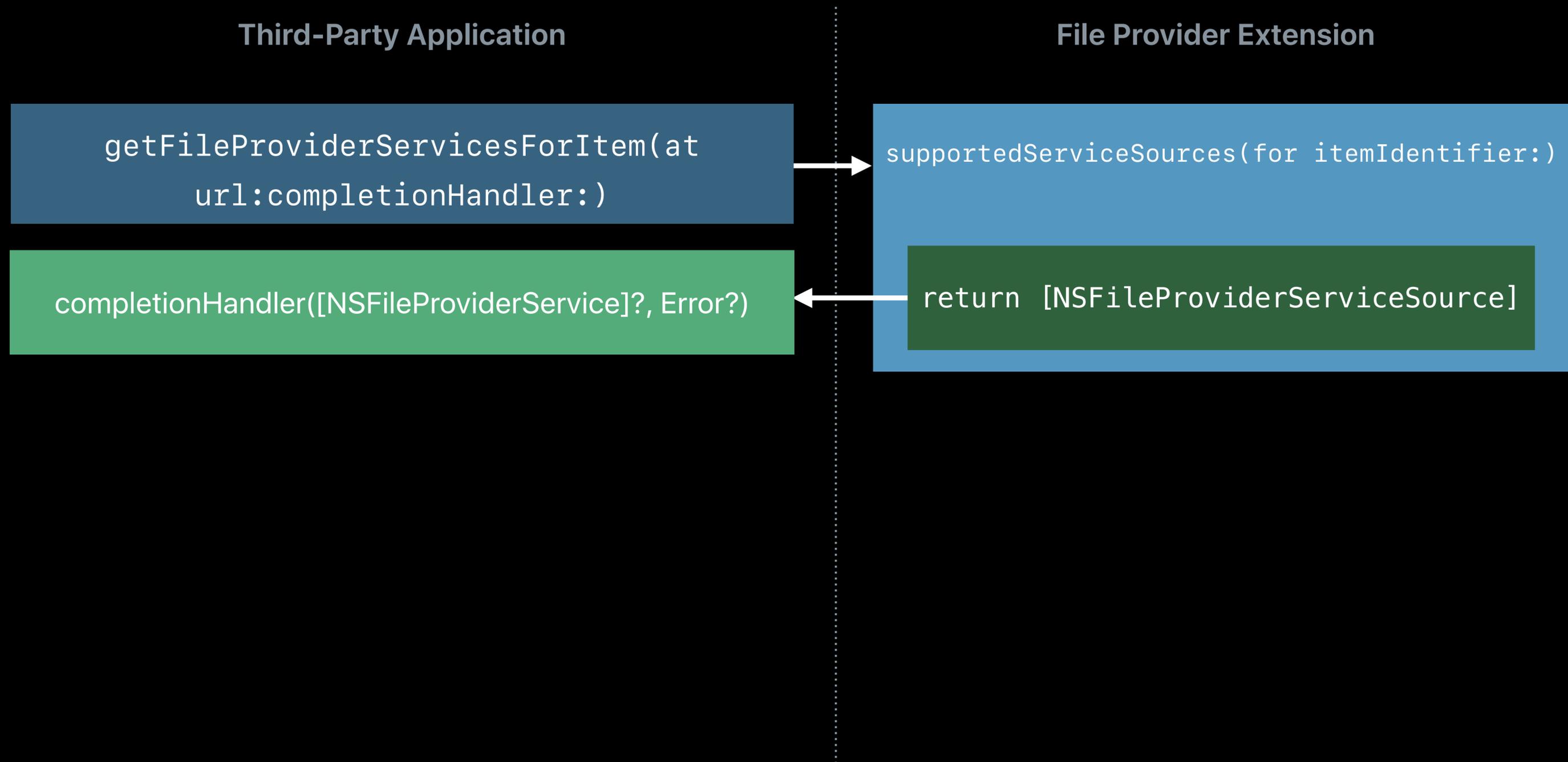
File Provider Extension

```
supportedServiceSources(for itemIdentifier:)
```

```
return [NSFileProviderServiceSource]
```

Services

Data flow



Services

Data flow

Third-Party Application

```
NSFileProviderMessenger(interface:,  
                          protocol:)
```

File Provider Extension

Services

Data flow

Third-Party Application

```
NSFileProviderMessenger(interface:,  
protocol:)
```



File Provider Extension

```
makeListenerEndpoint
```

Services

Data flow

Third-Party Application

```
NSFileProviderMessenger(interface:,  
protocol:)
```



File Provider Extension

```
makeListenerEndpoint
```

```
return NSXPCListenerEndpoint
```

Services

Data flow

Third-Party Application

```
NSFileProviderMessenger(interface:,  
protocol:)
```



```
let proxy = messenger.remoteObjectProxy
```



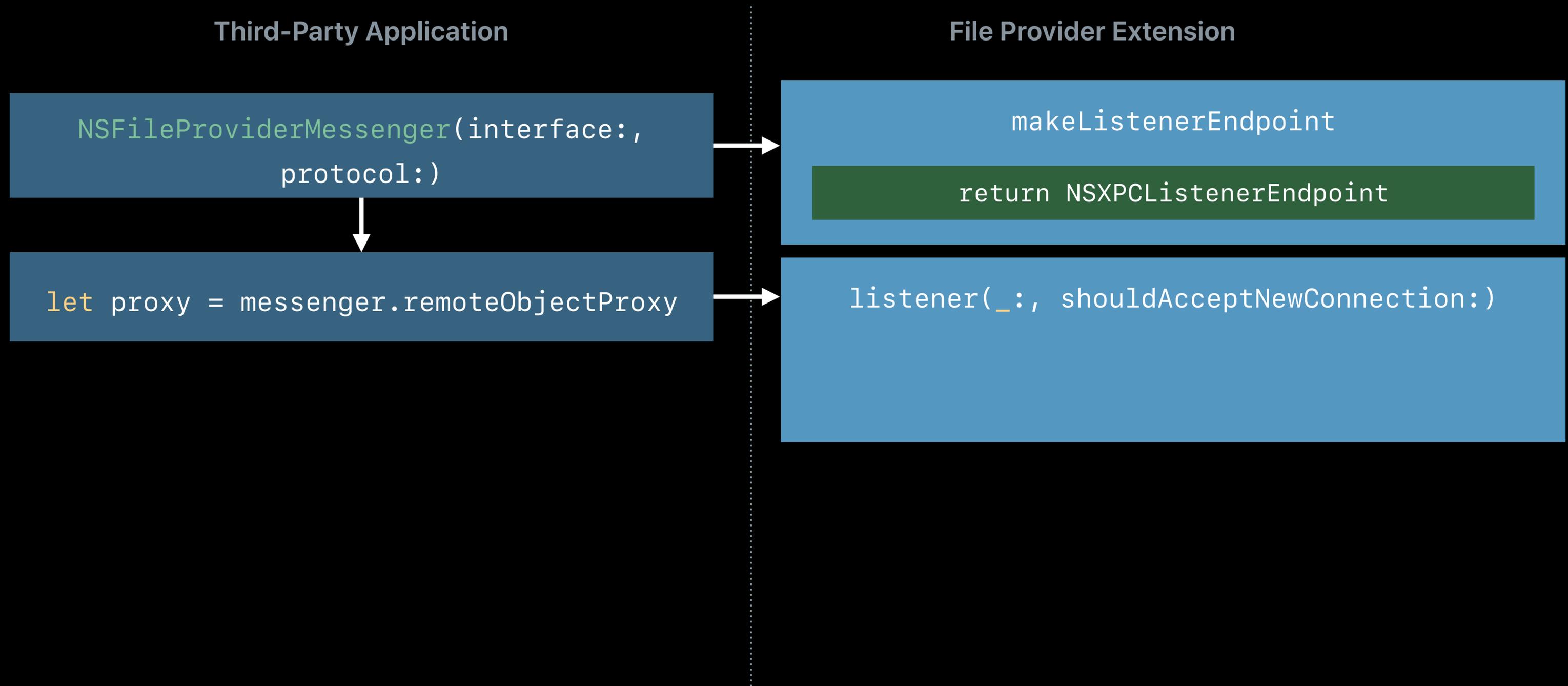
File Provider Extension

```
makeListenerEndpoint
```

```
return NSXPCListenerEndpoint
```

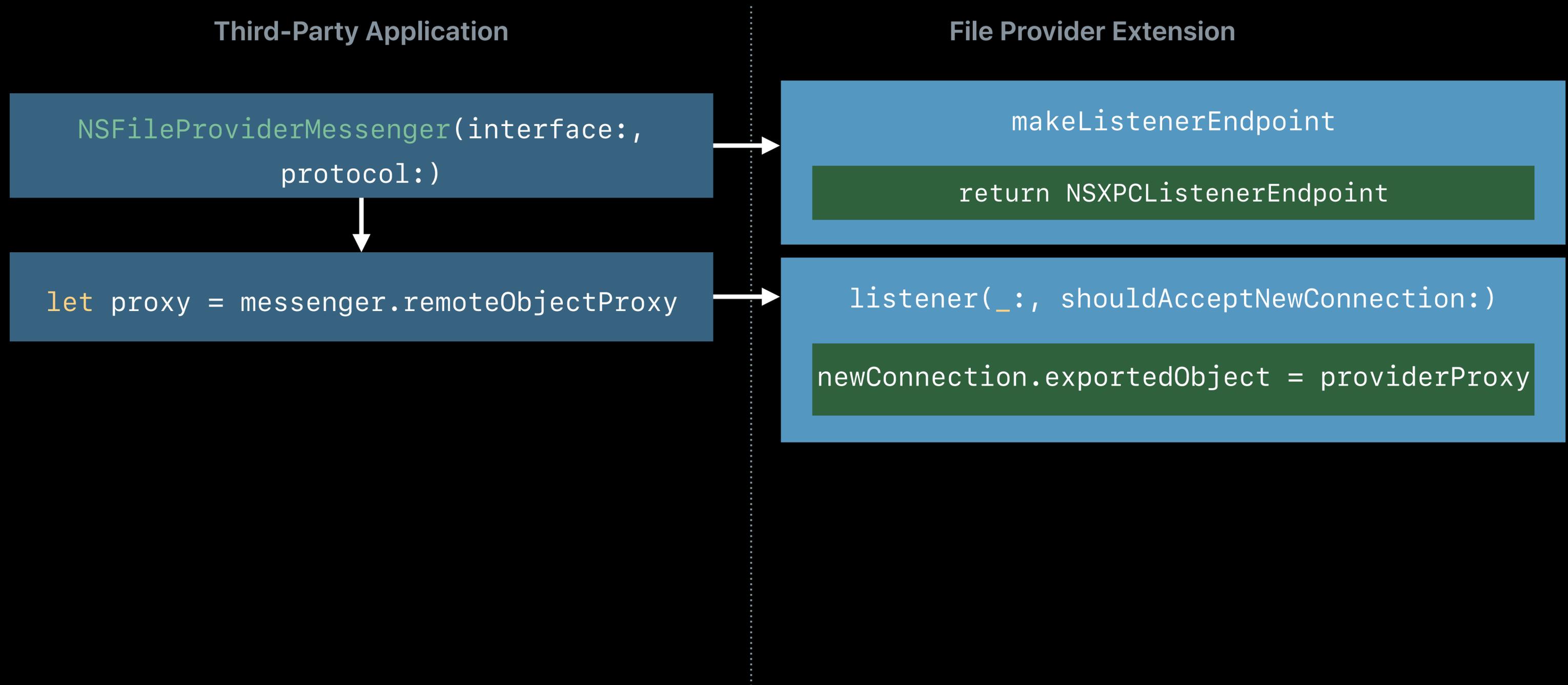
Services

Data flow



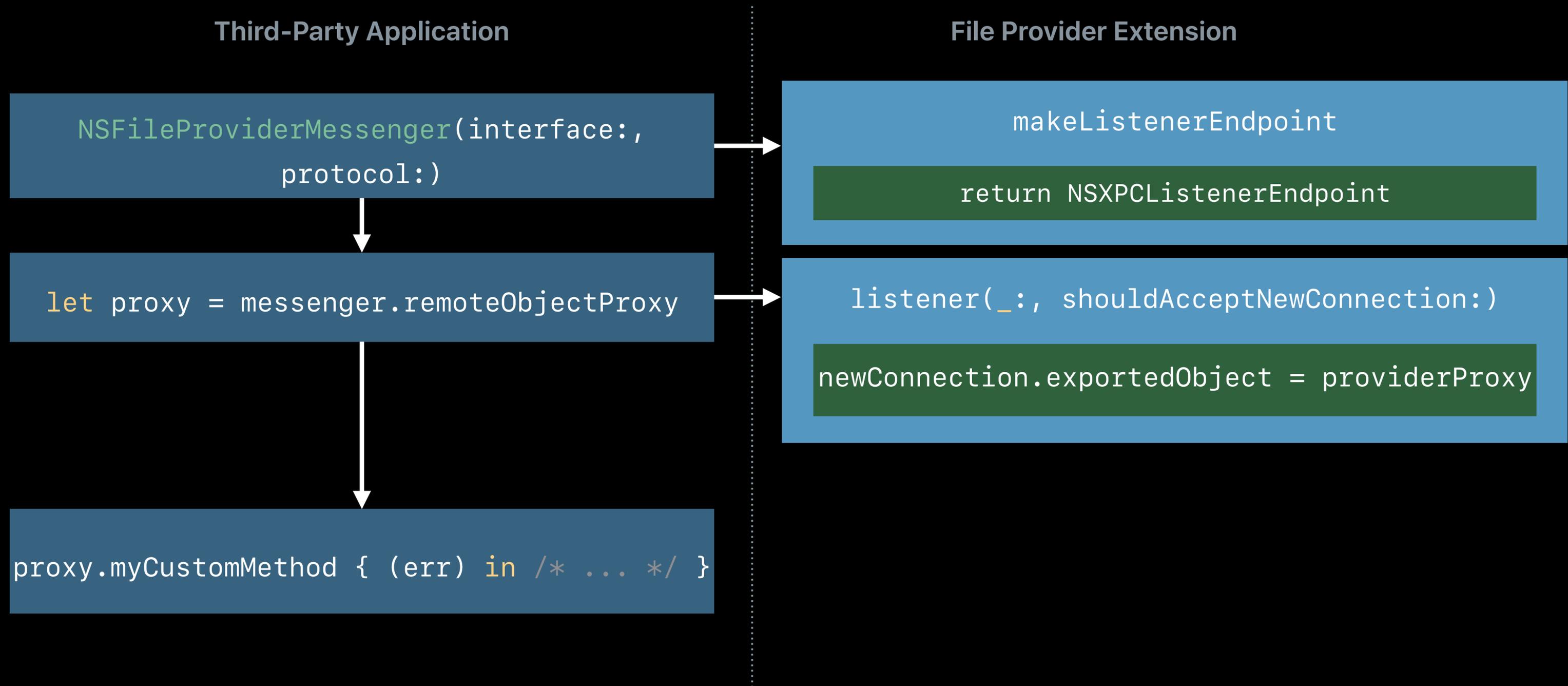
Services

Data flow



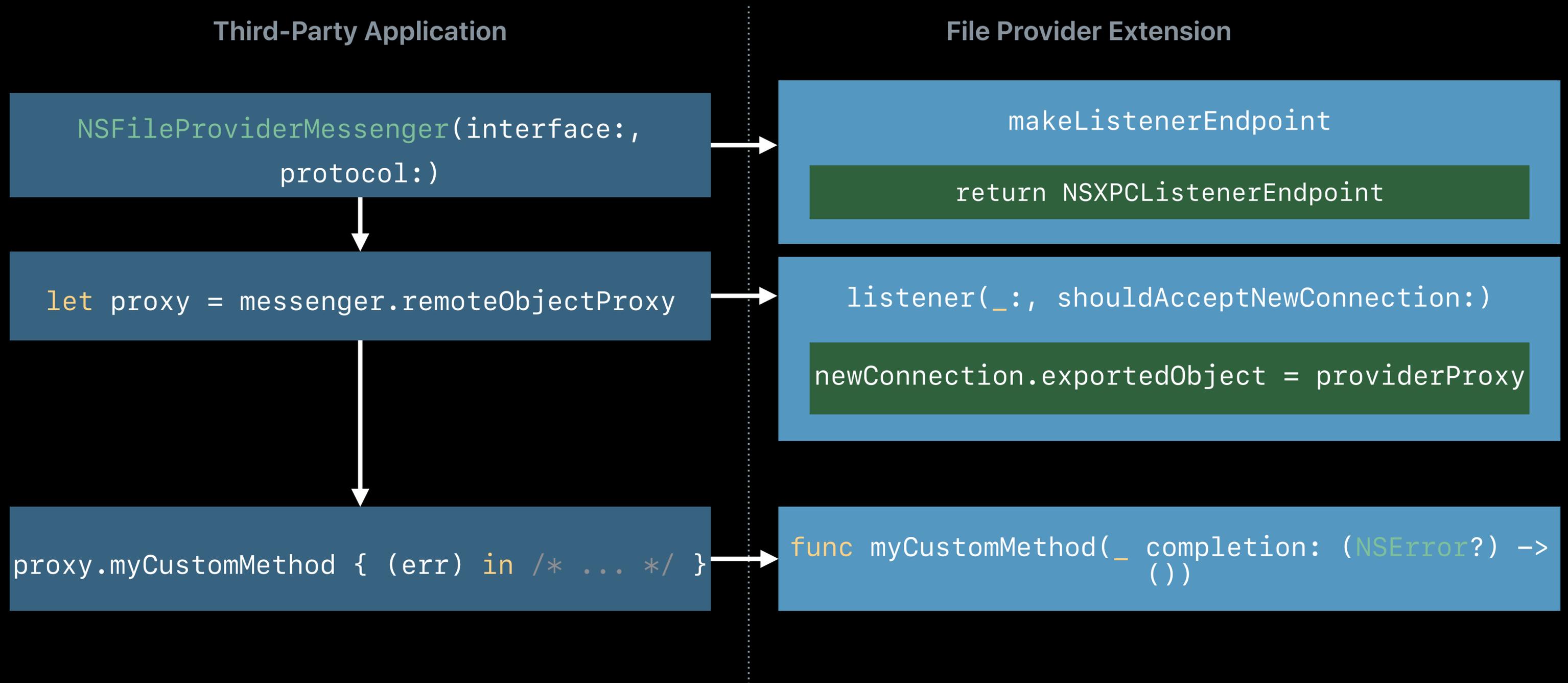
Services

Data flow



Services

Data flow



Summary

Summary

File provider extensions give you a way to integrate into the system

Summary

File provider extensions give you a way to integrate into the system

Expose items through enumerations to back browsing and the working set

Summary

File provider extensions give you a way to integrate into the system

Expose items through enumerations to back browsing and the working set

Implement actions to allow modifications on the hierarchy

More Information

<https://developer.apple.com/wwdc17/243>

Related Sessions

Performance Optimization on iPhone OS

WWDC 2010

Cocoa Interprocess Communication with XPC

WWDC 2012

Building Great Document-Based Apps in iOS 11

WWDC 2017

Labs

File Providers and Document Browser Lab

Technology Lab H

Fri 1:00PM-4:00PM

Cocoa Lab

Technology Lab B

Fri 1:50PM-3:20PM

