

#WWDC19

Using Core Data with CloudKit

Nick Gillett, Core Data

Using Core Data with CloudKit

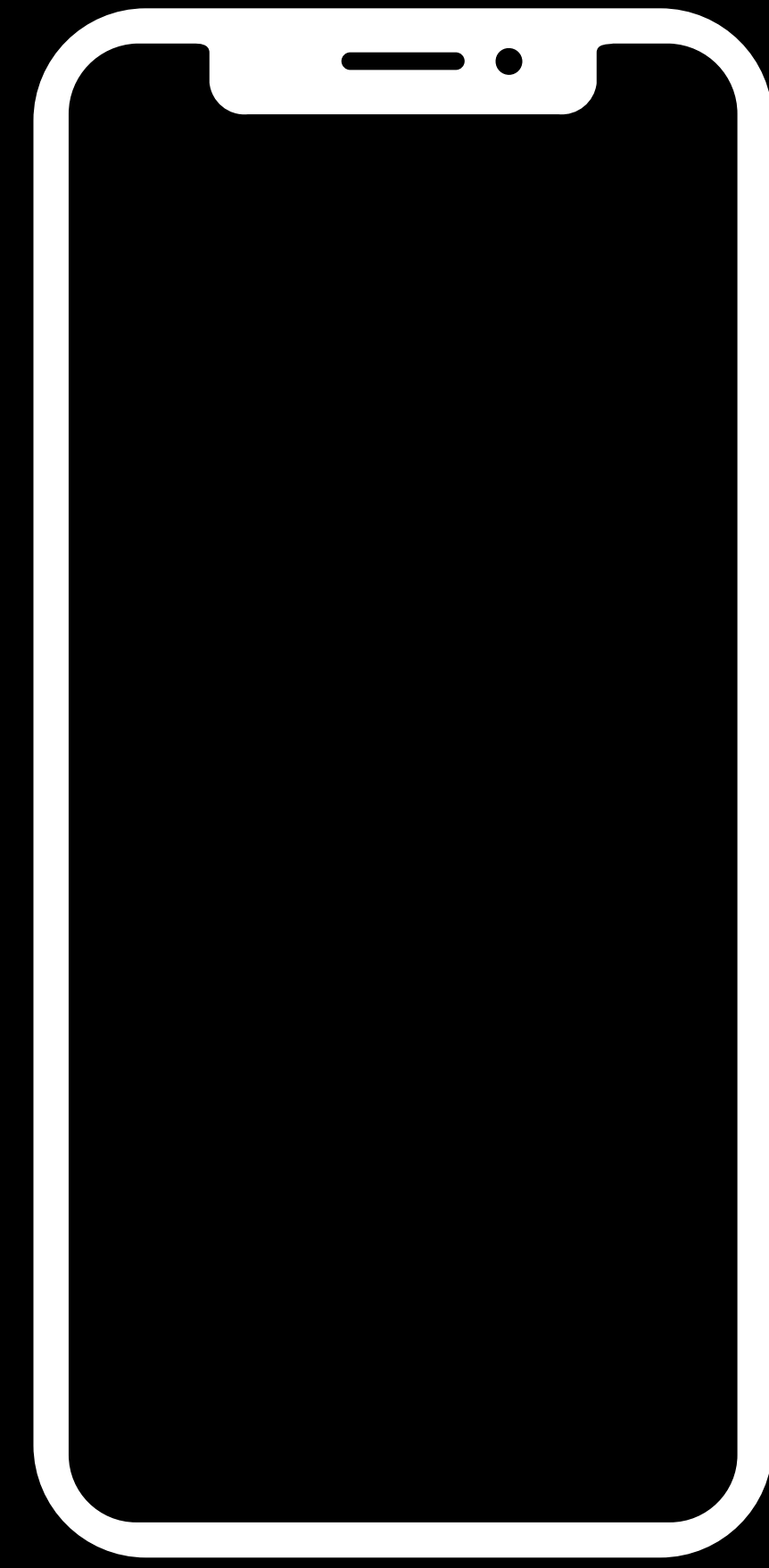
Using Core Data with CloudKit

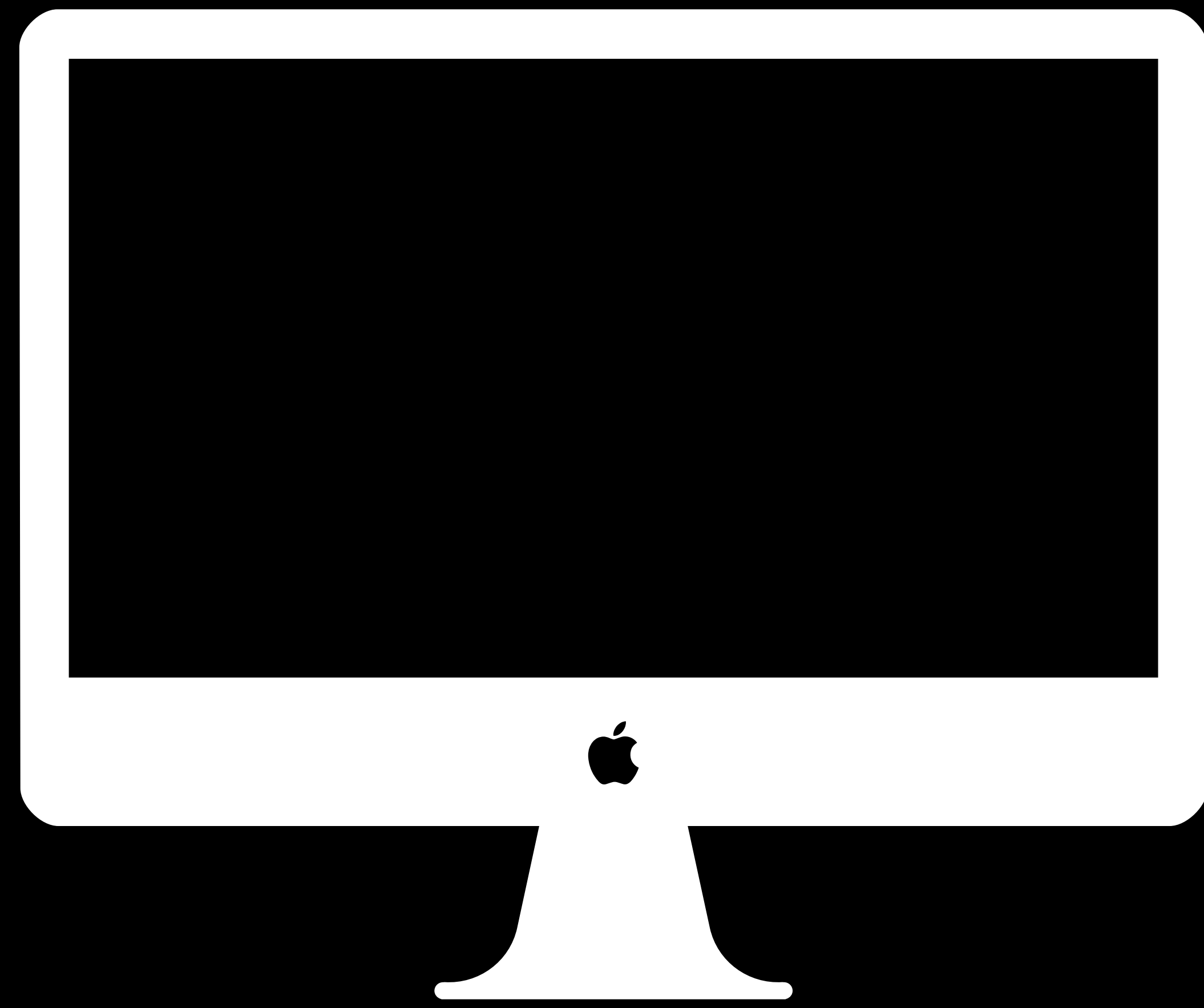
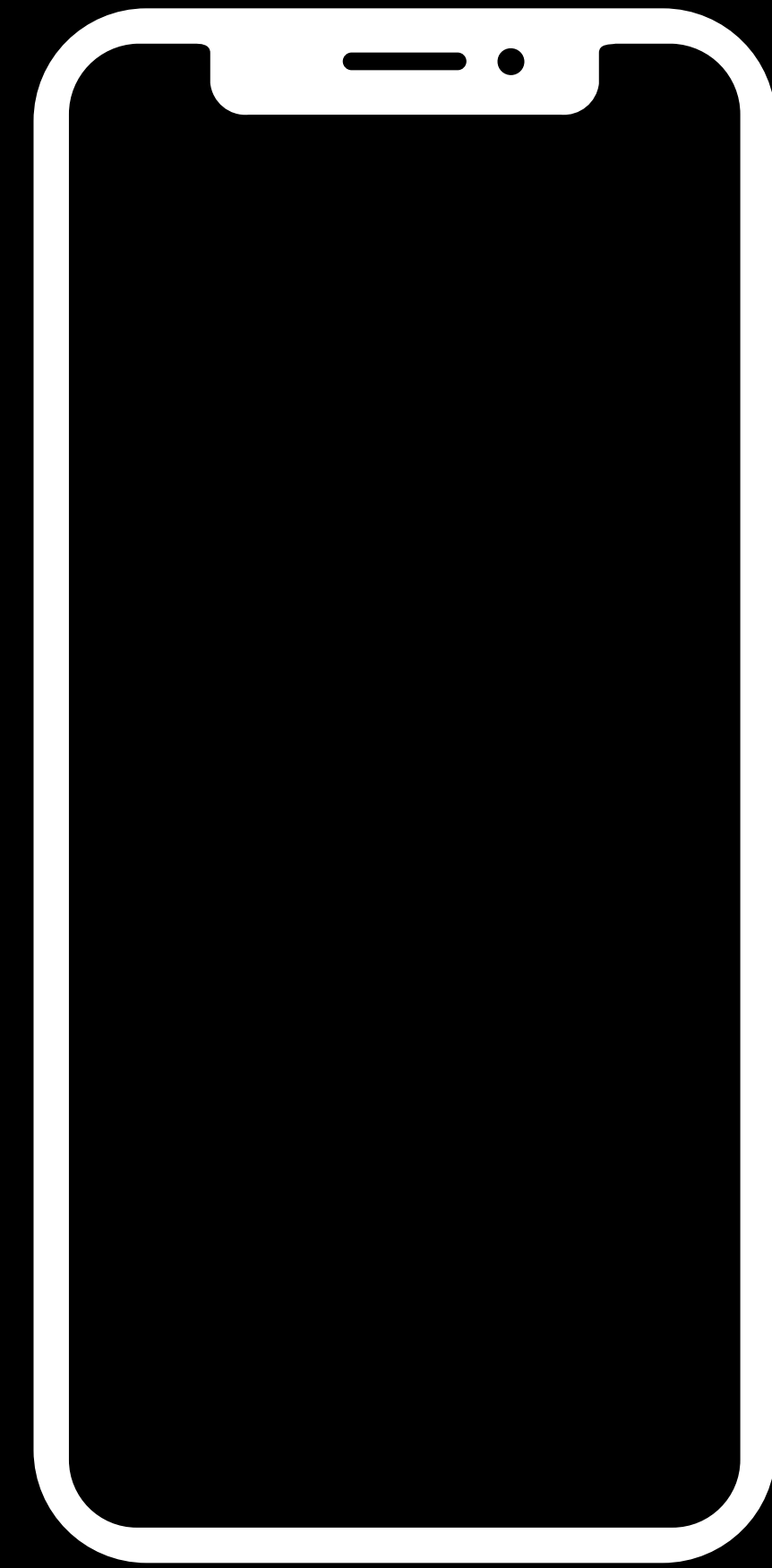
Everything should sync

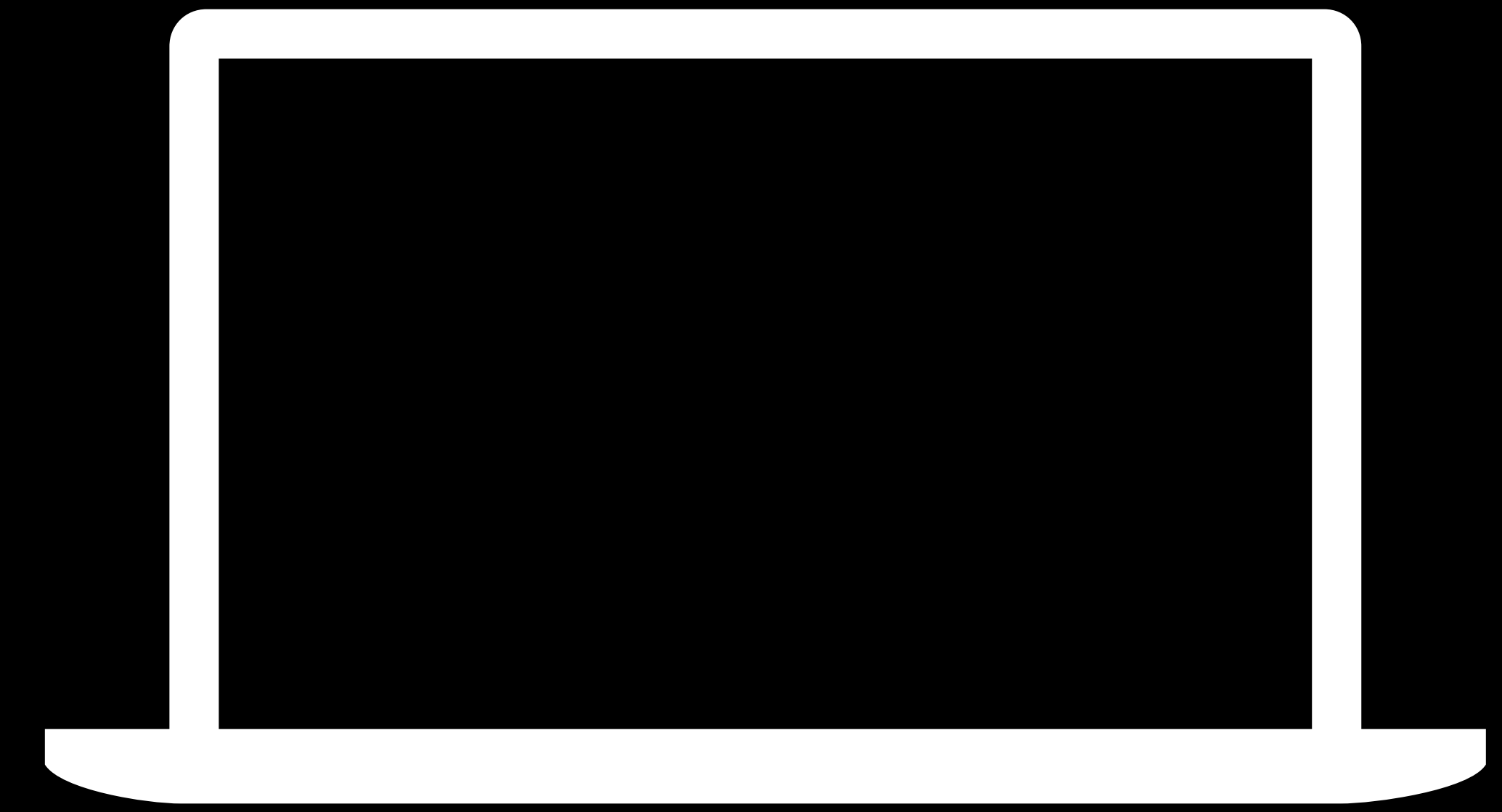
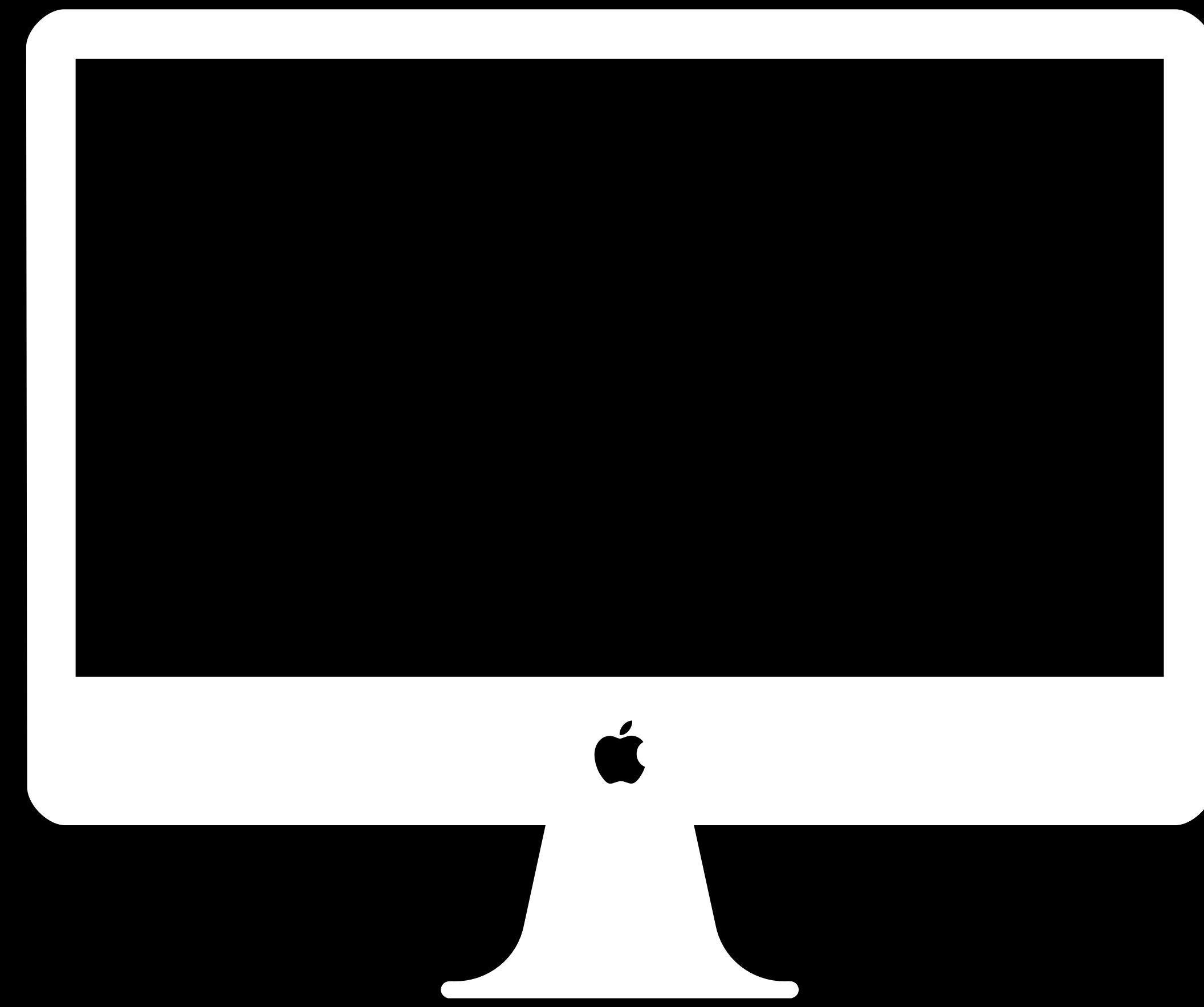
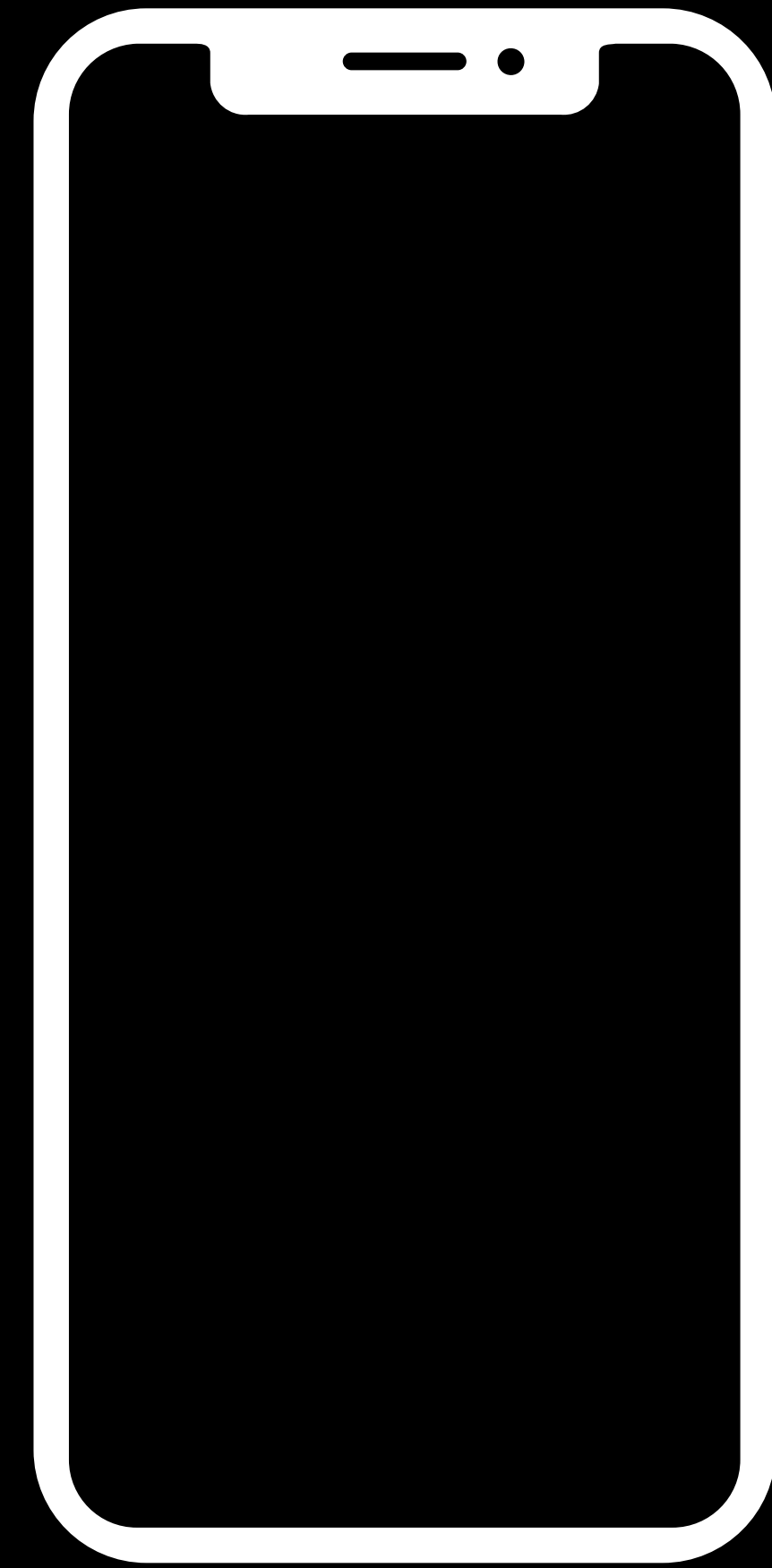
Using Core Data with CloudKit

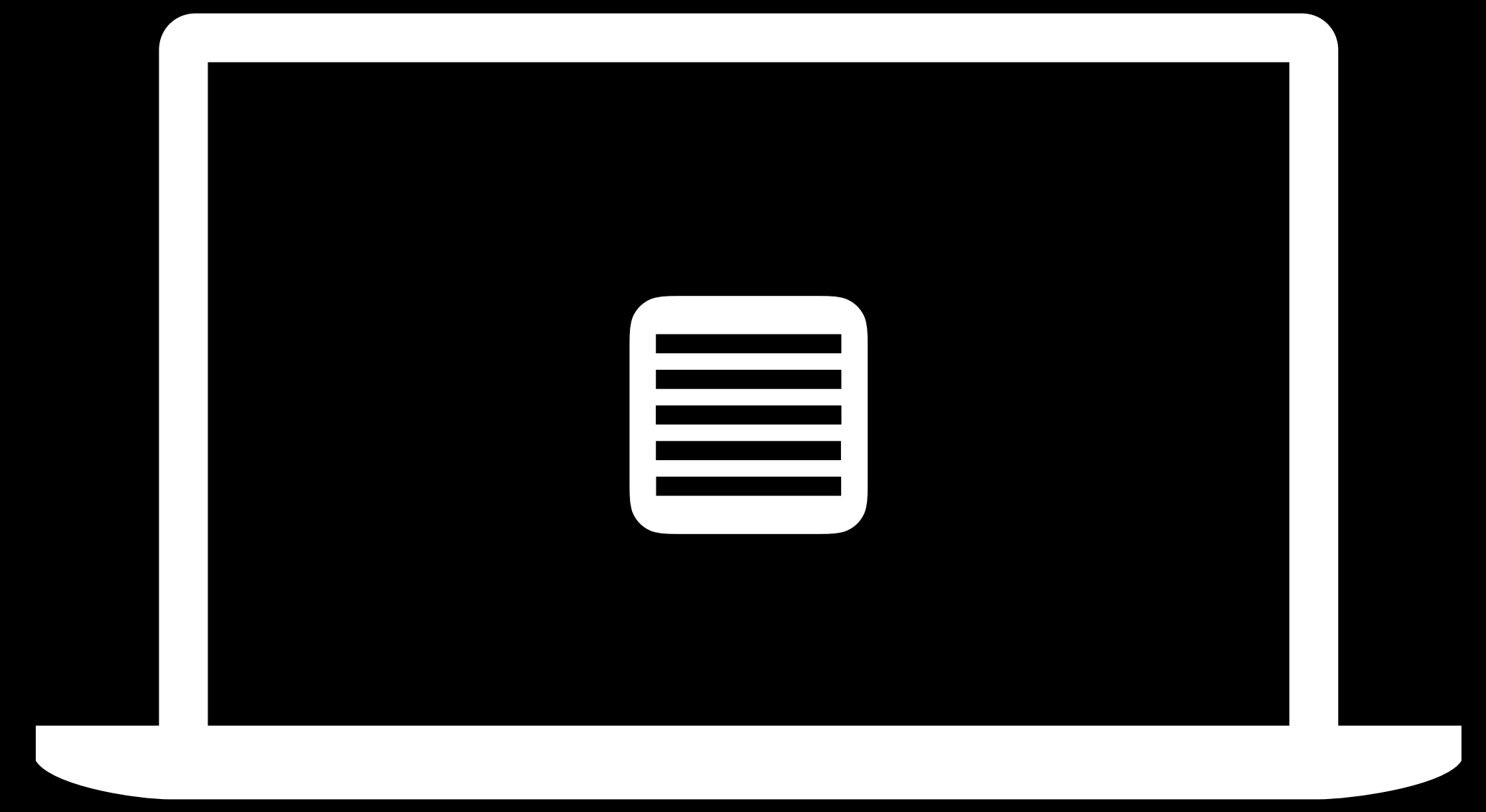
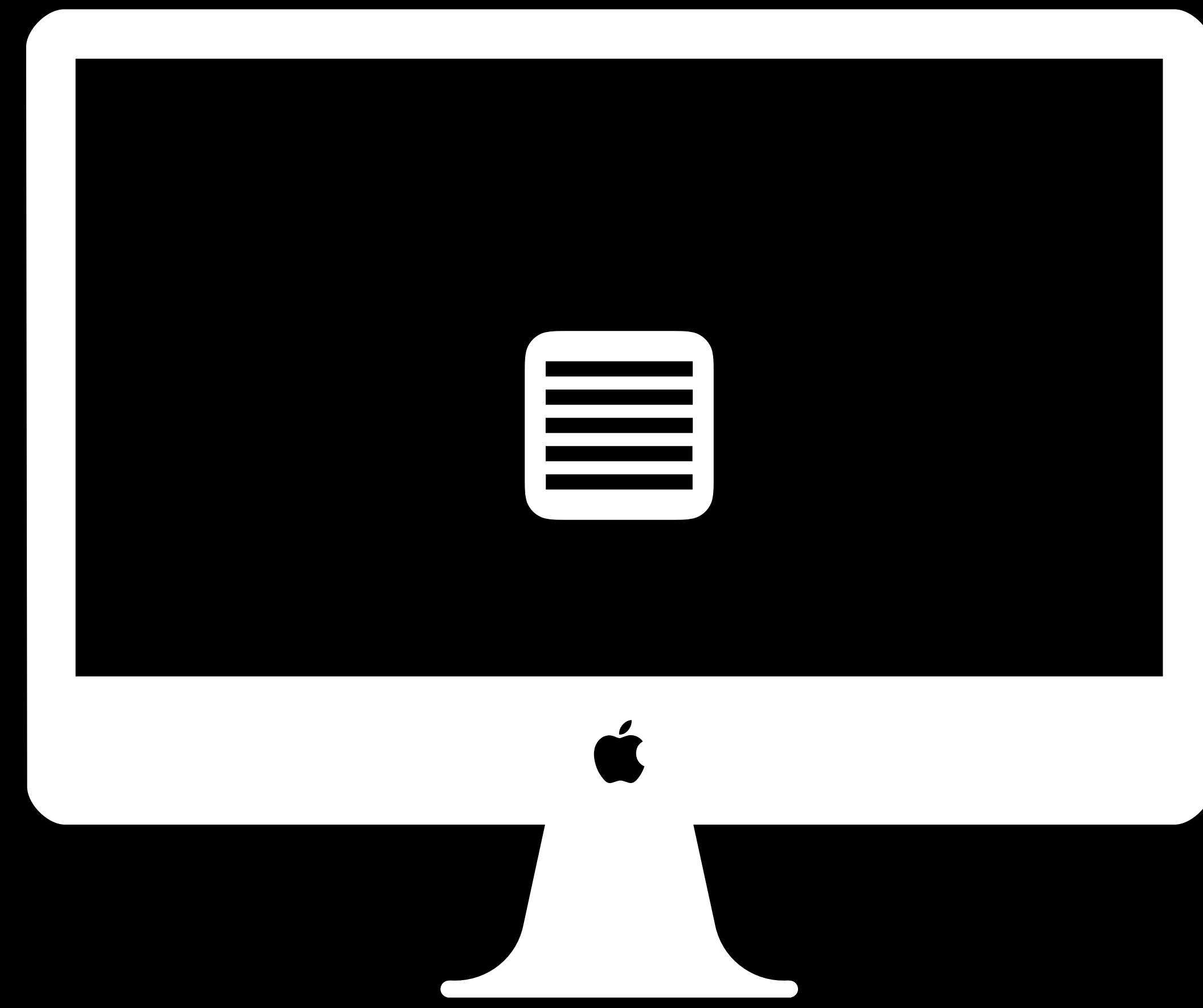
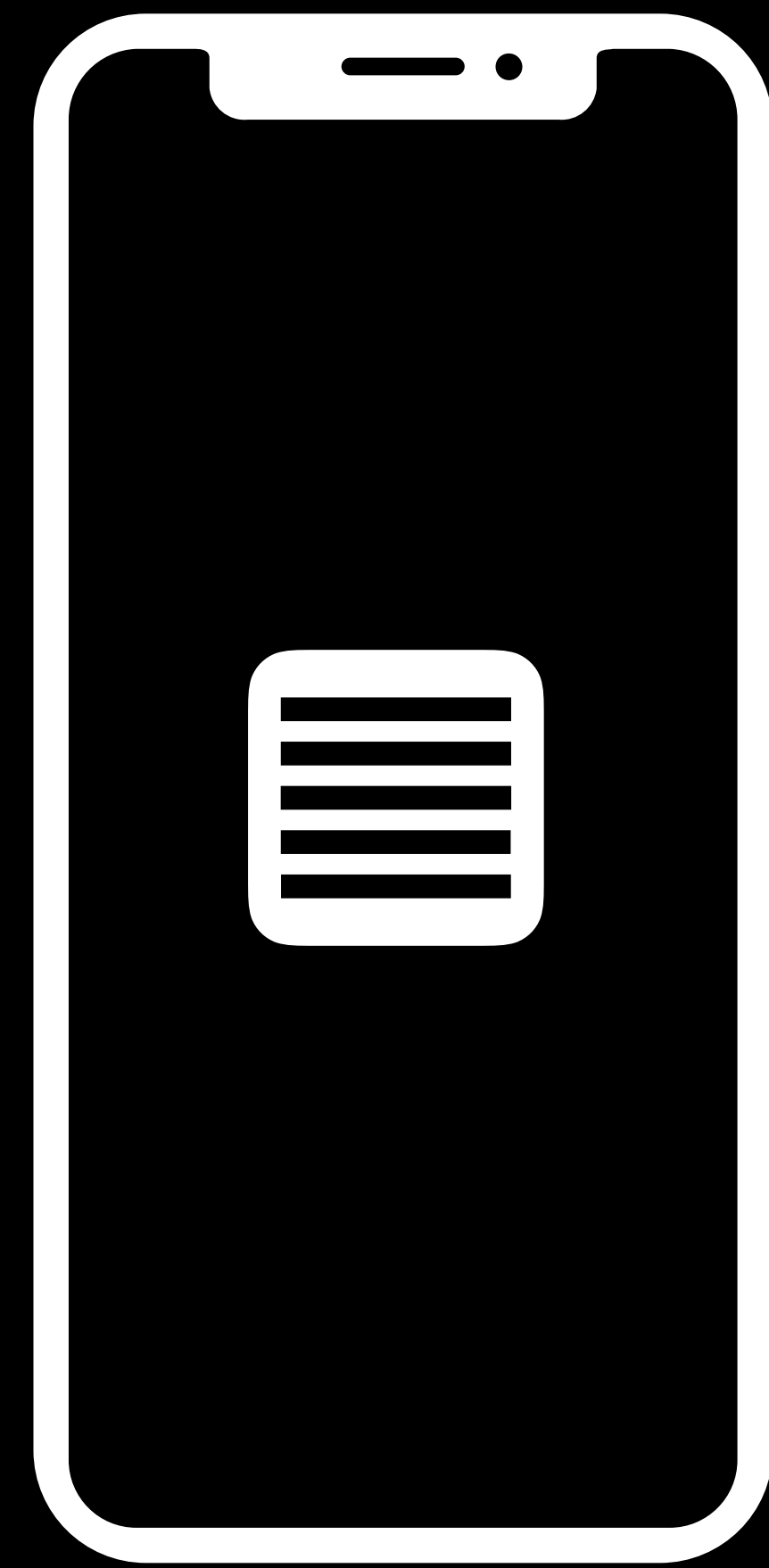
Everything should sync

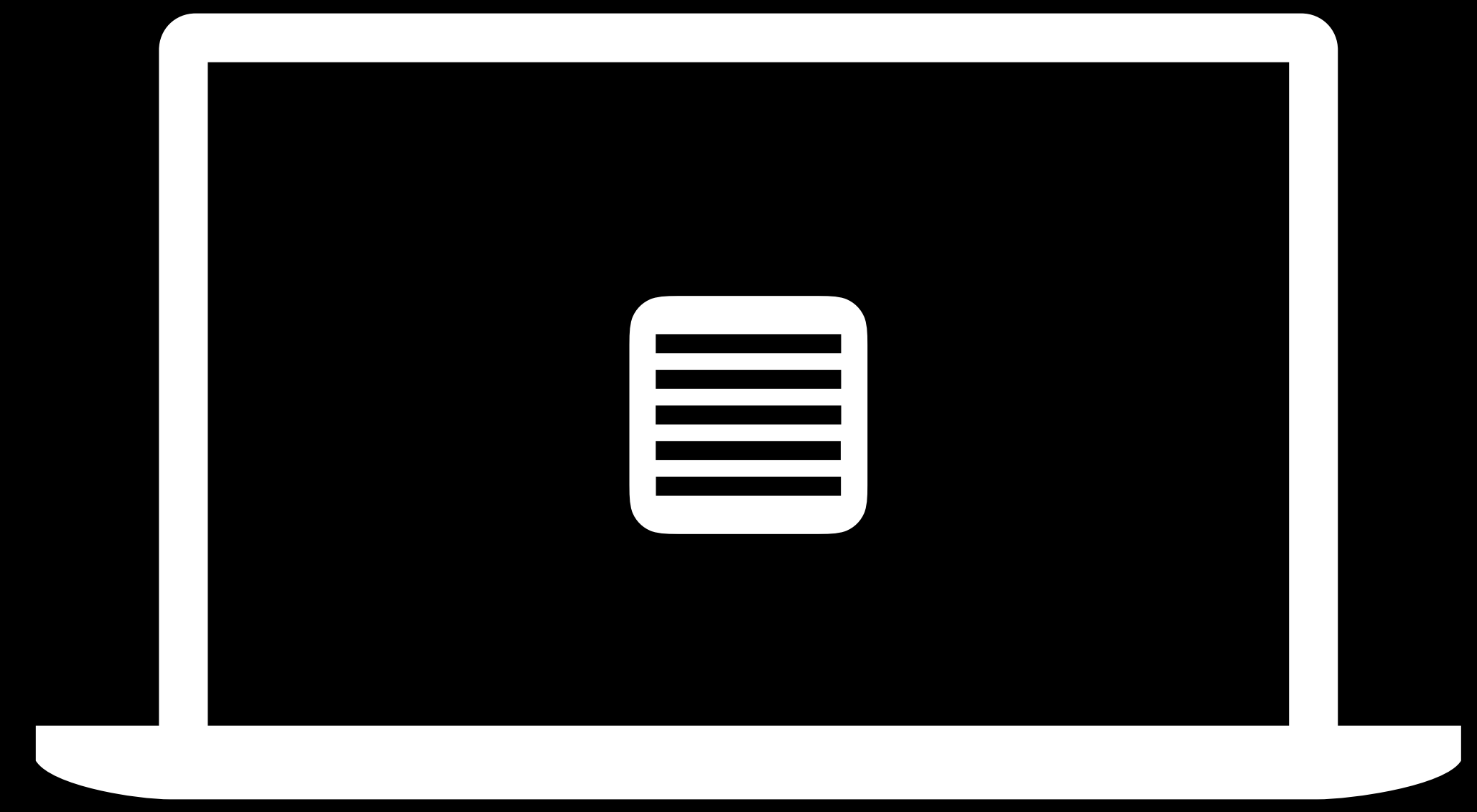
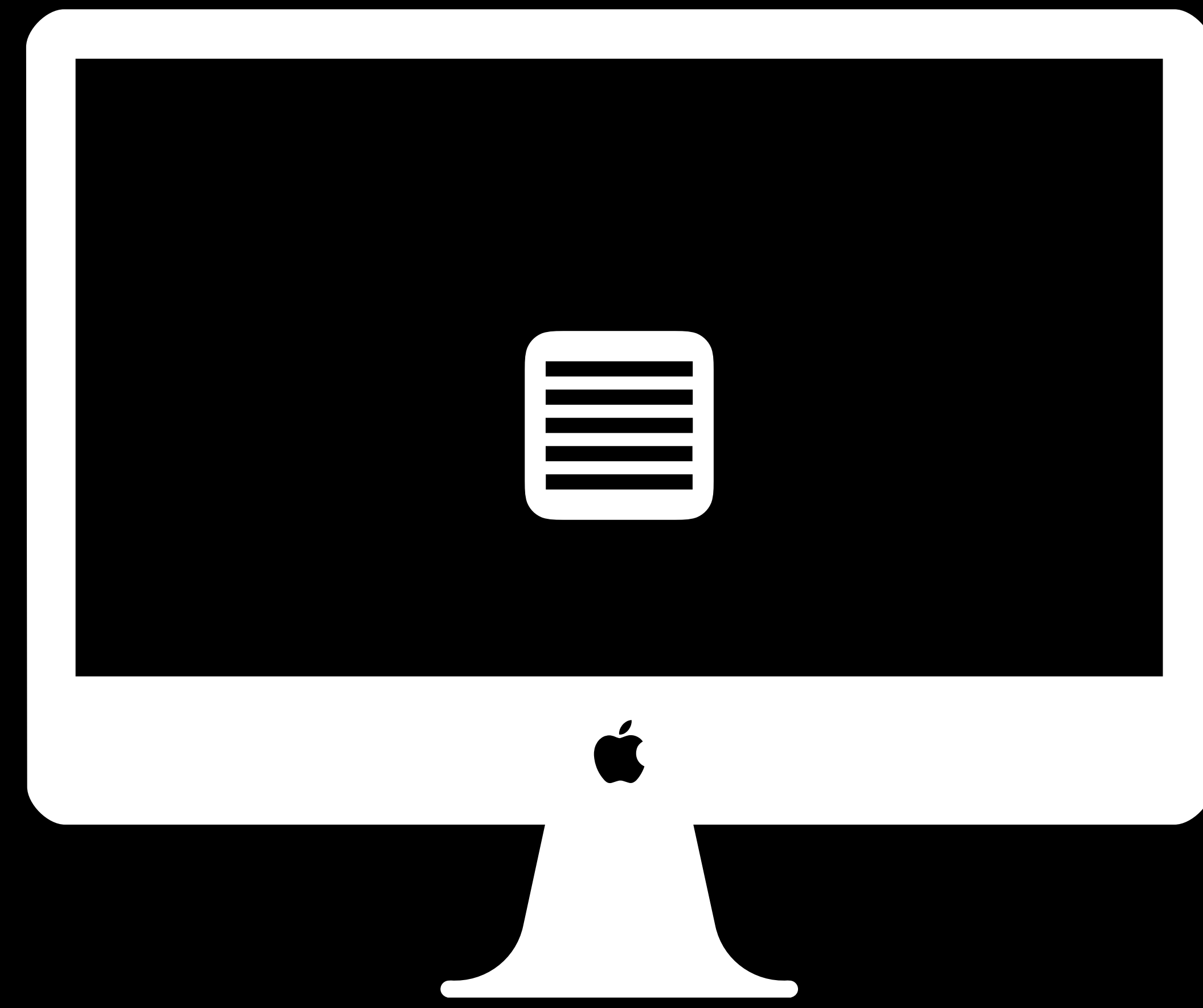
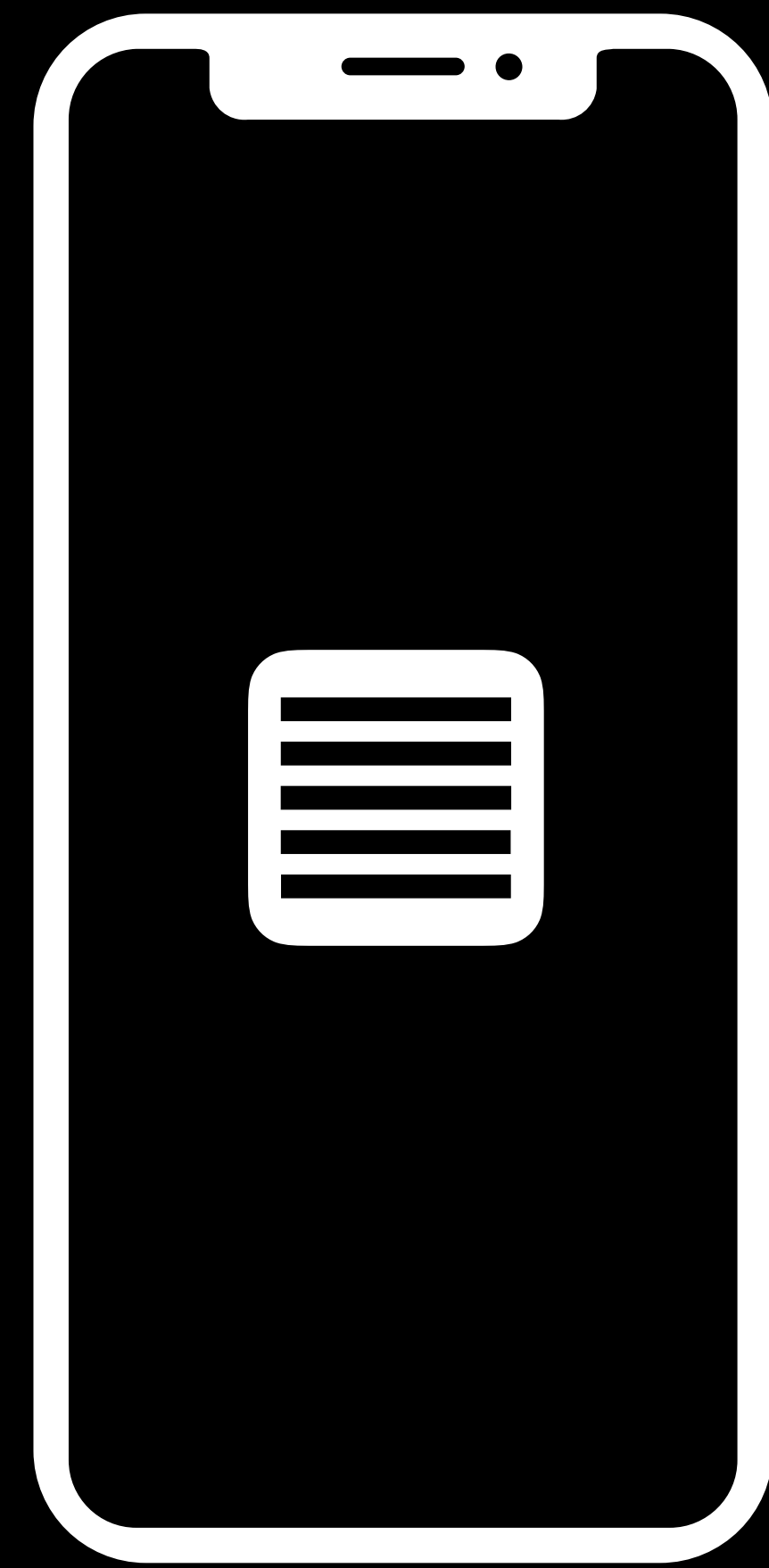
Sync should be easy

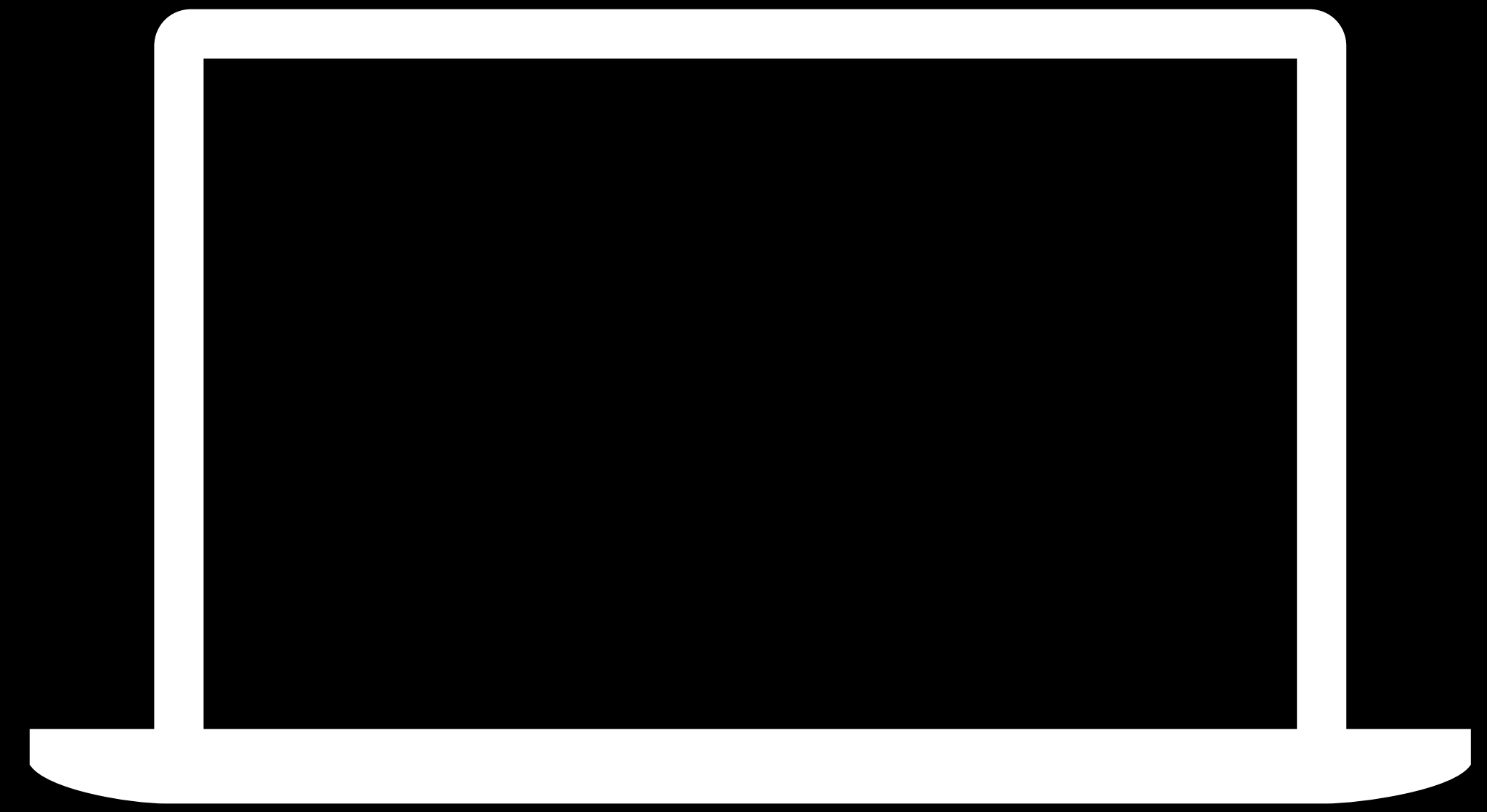
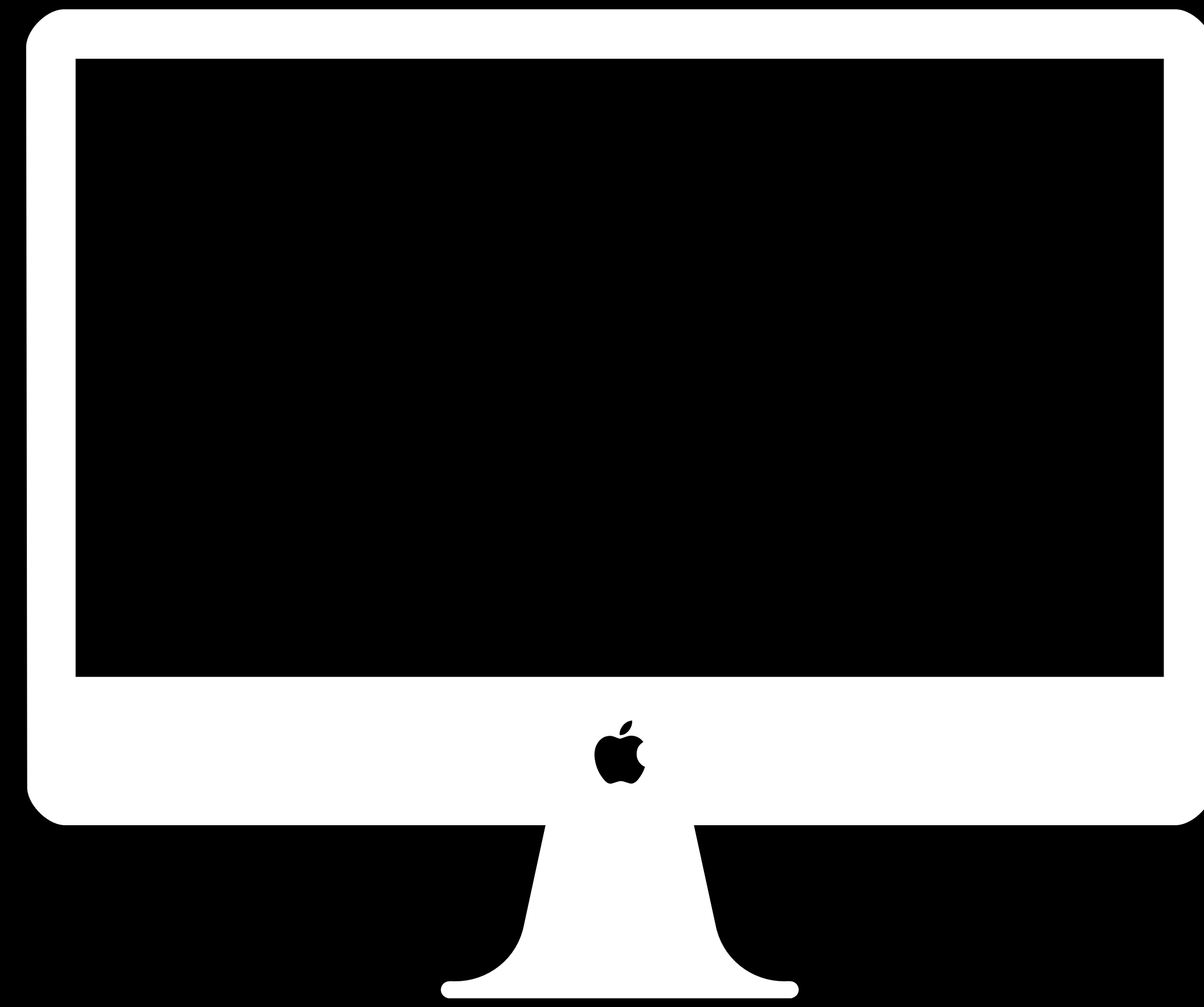
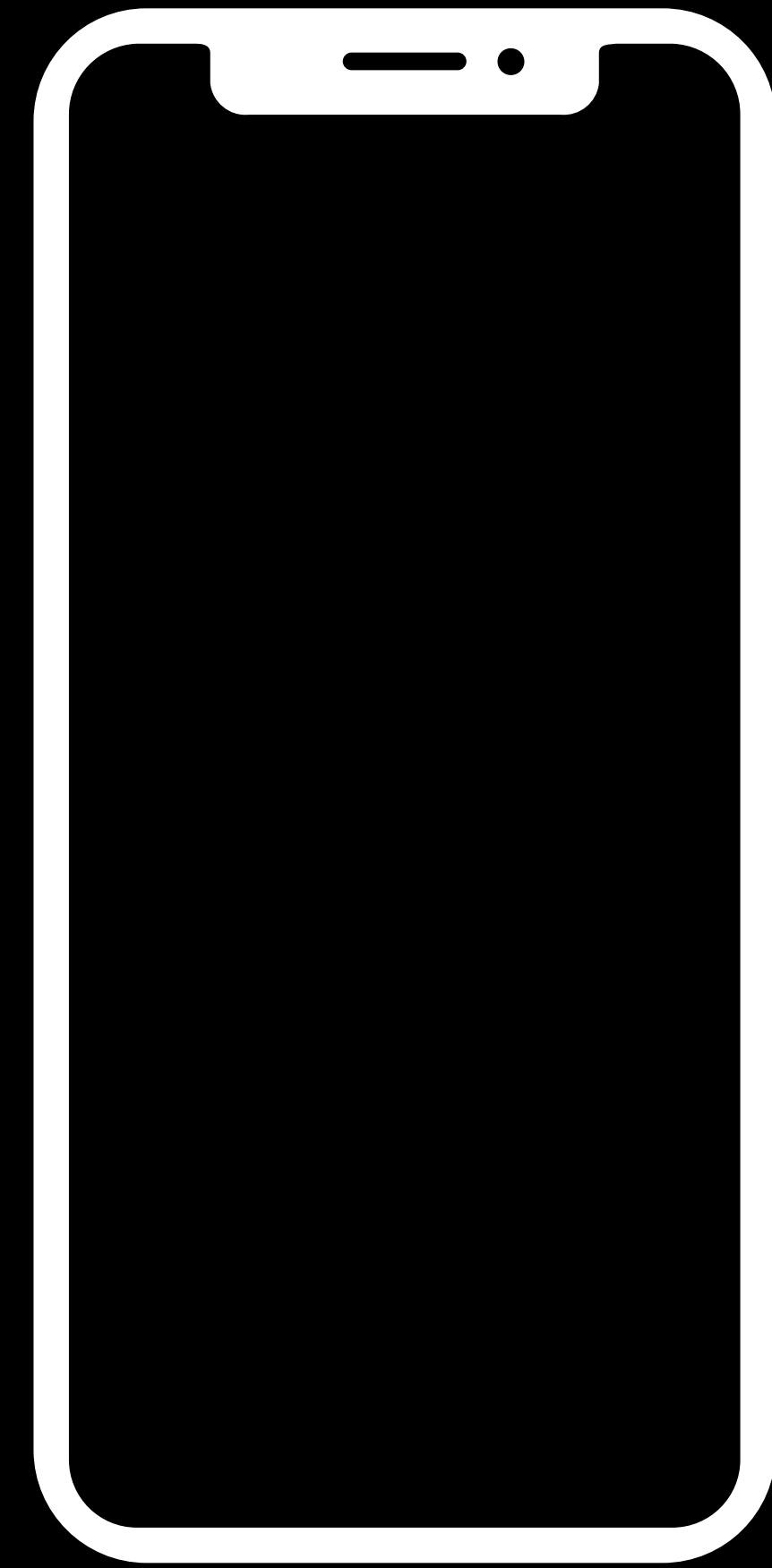


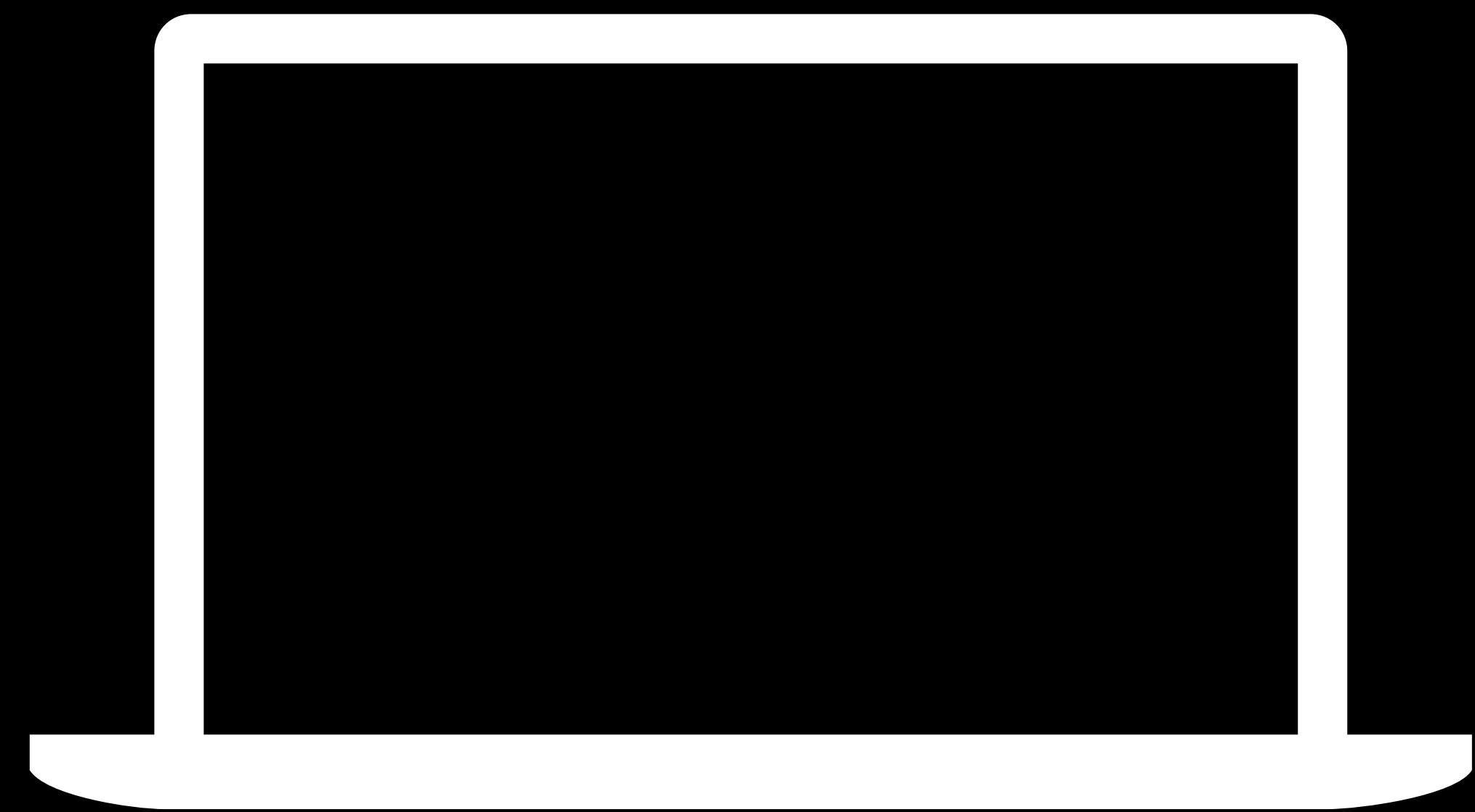
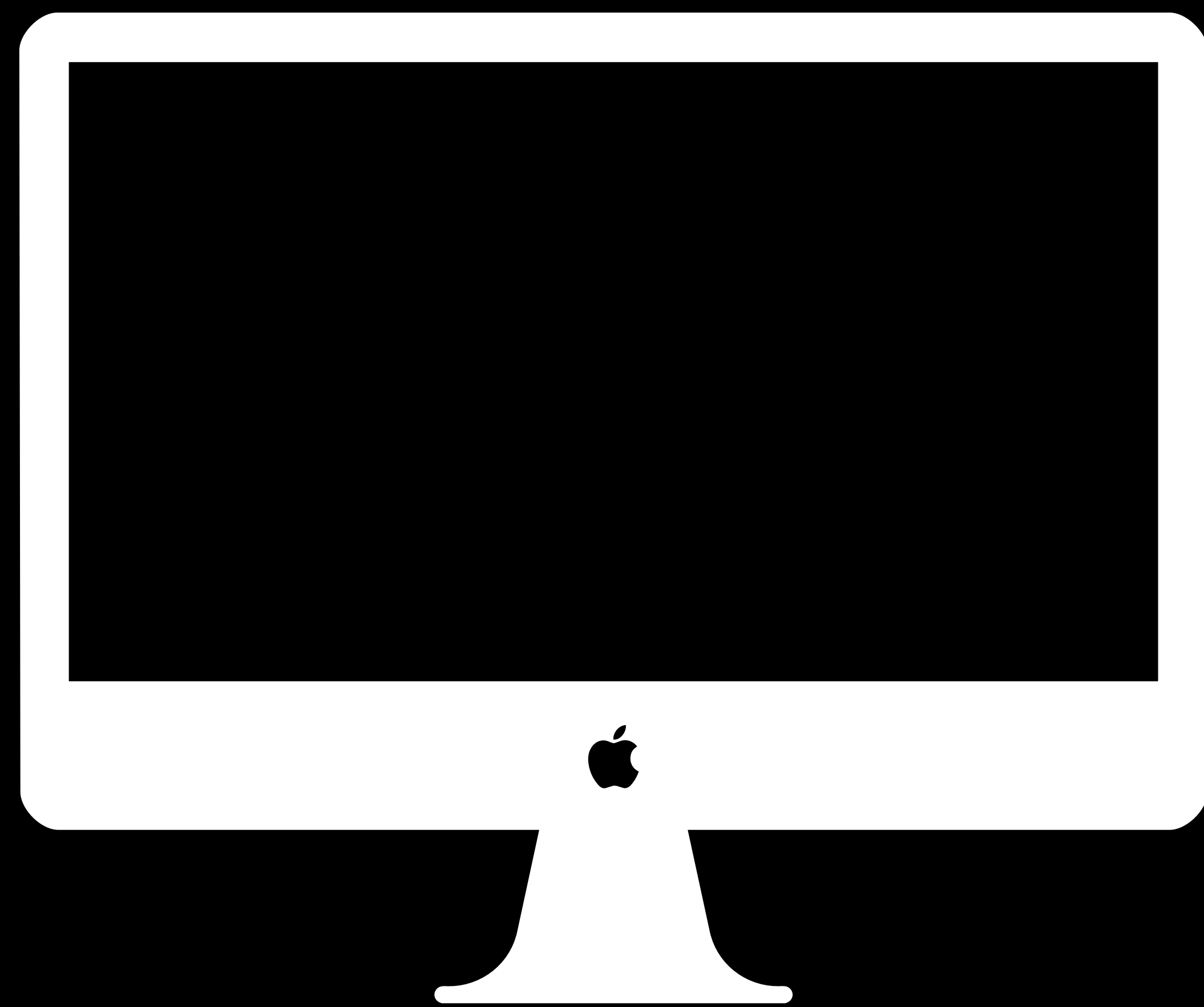
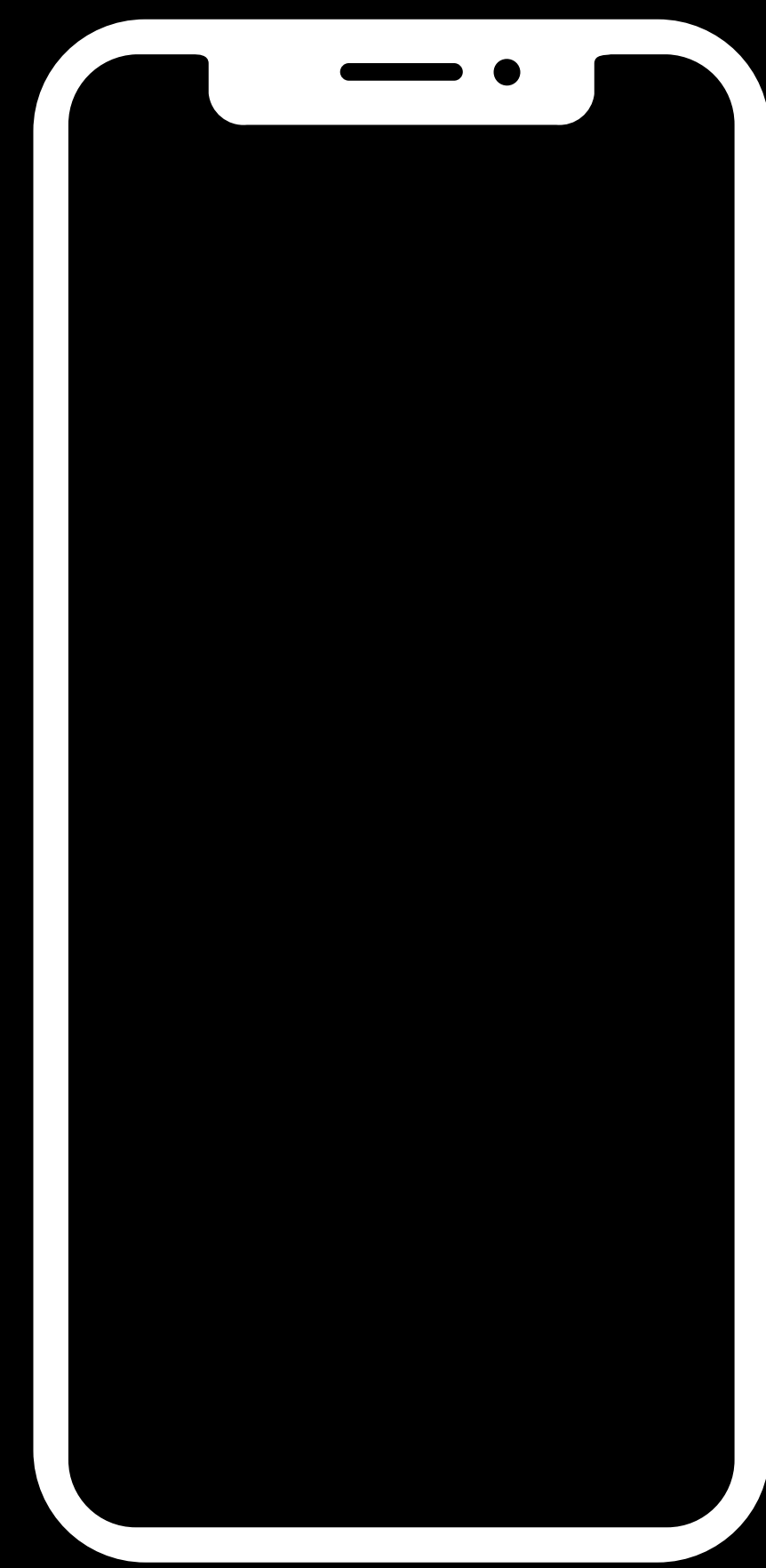


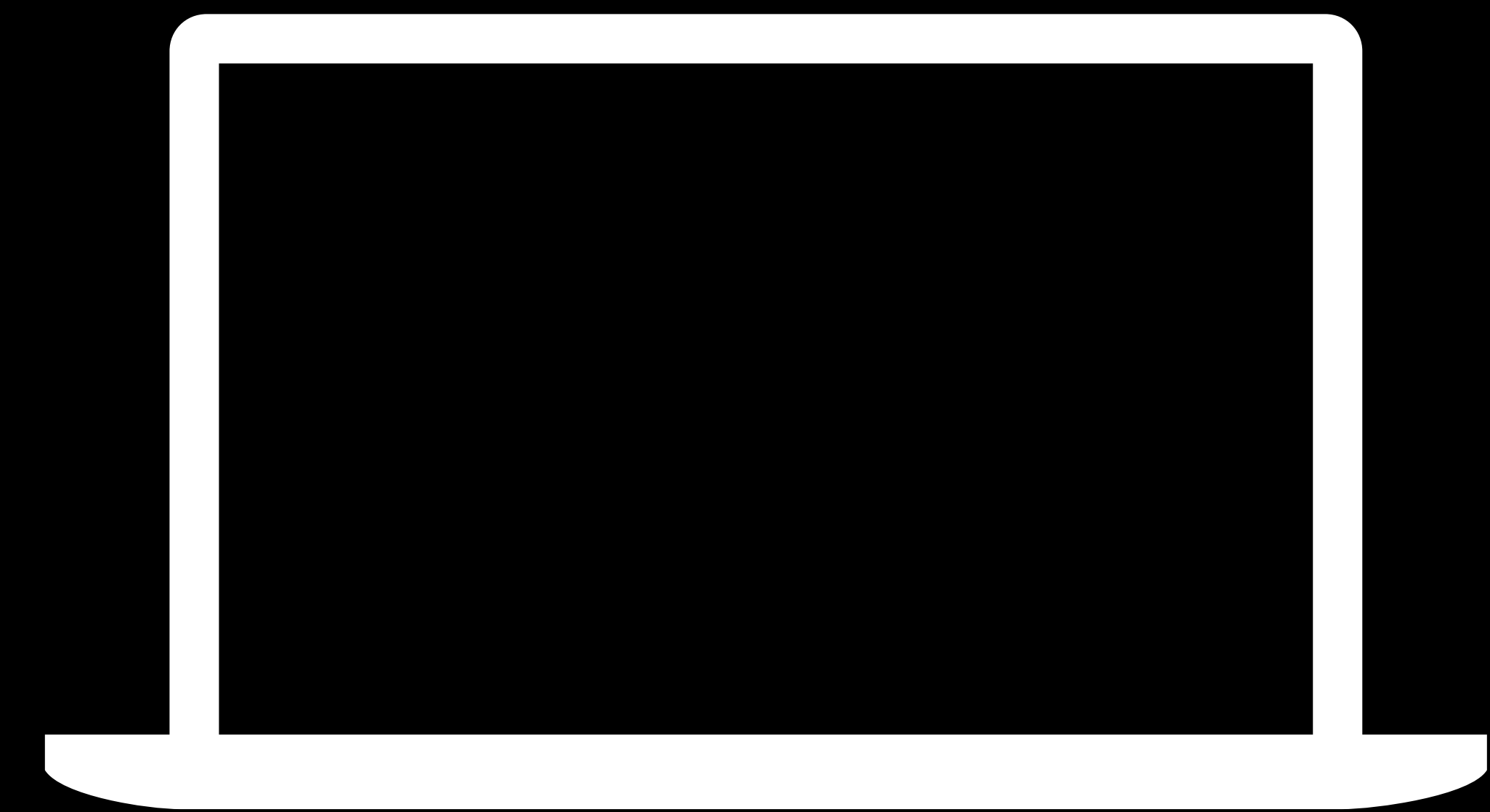
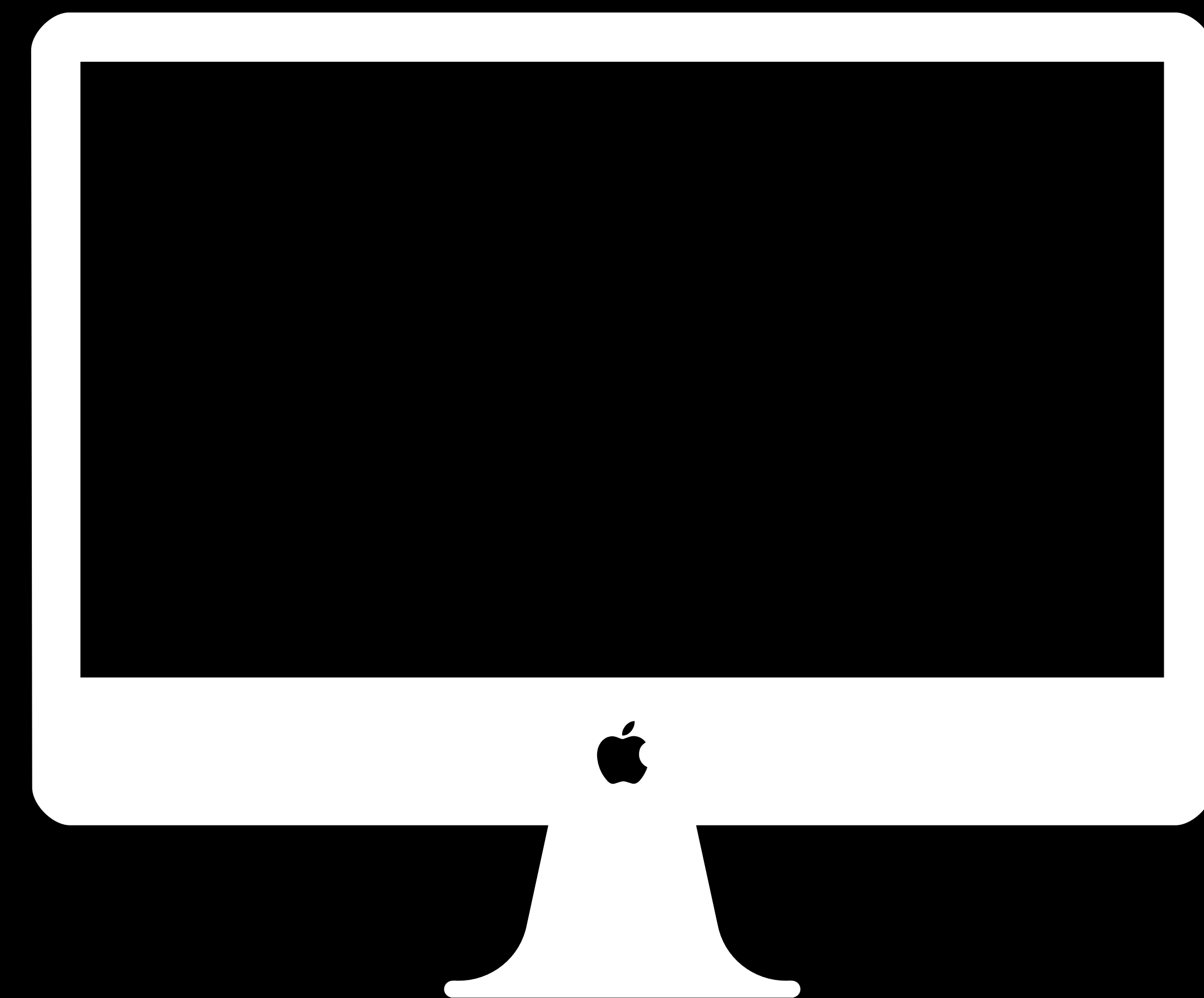
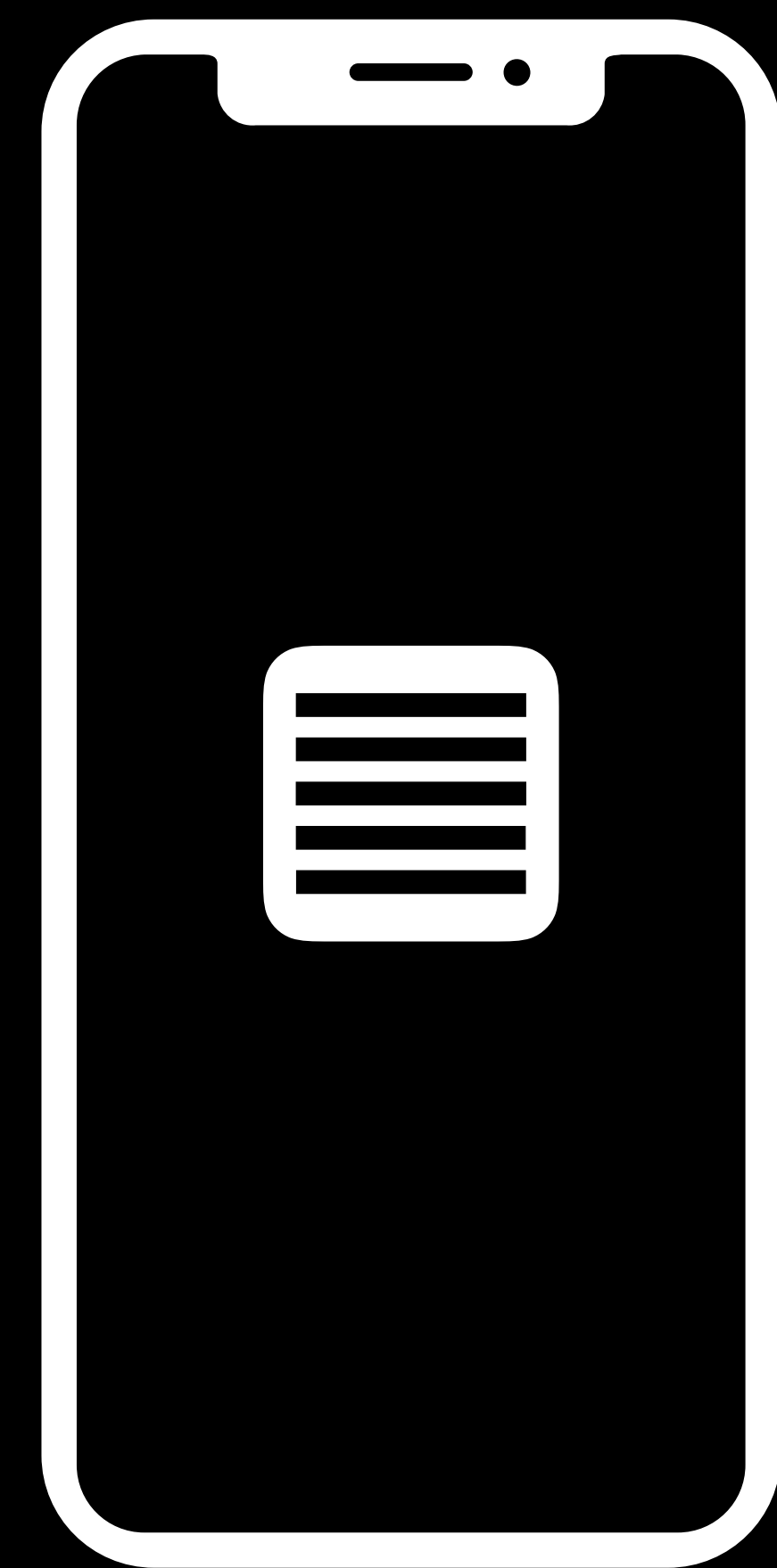
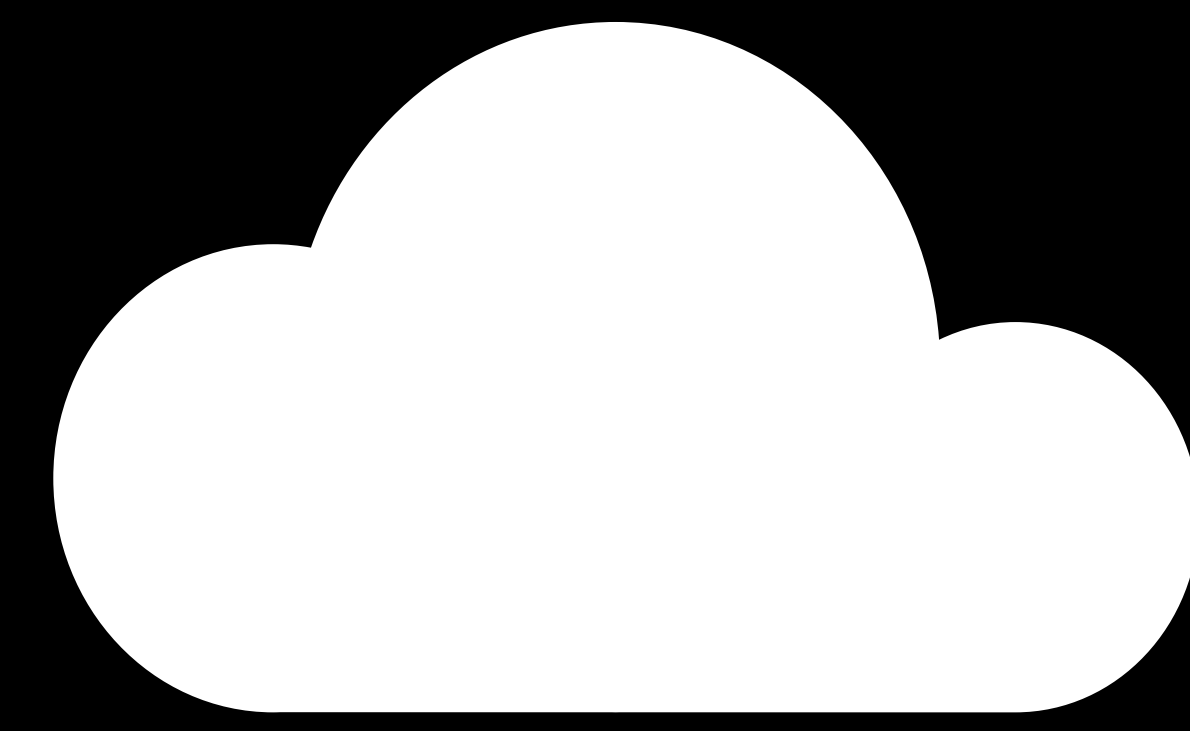


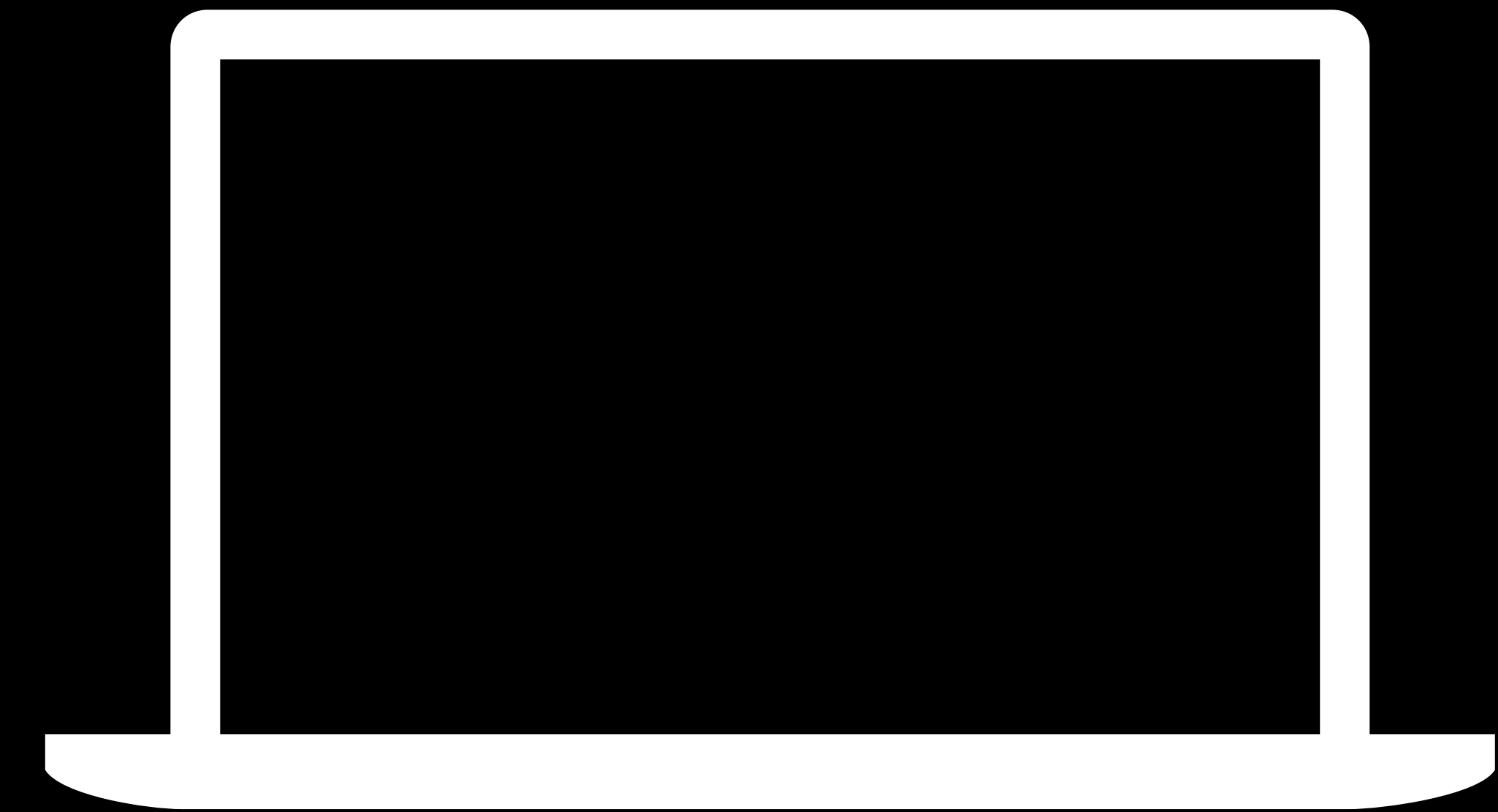
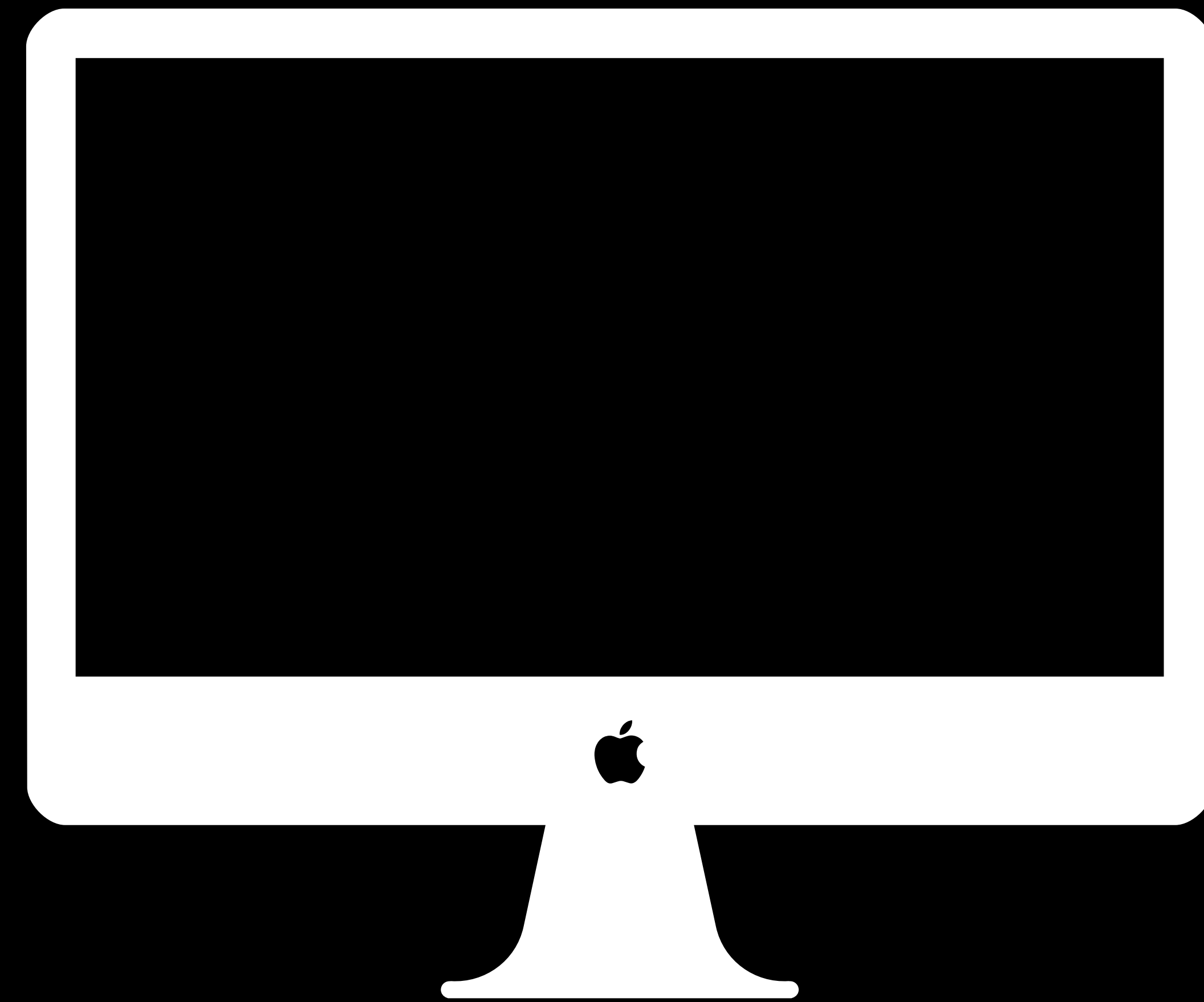
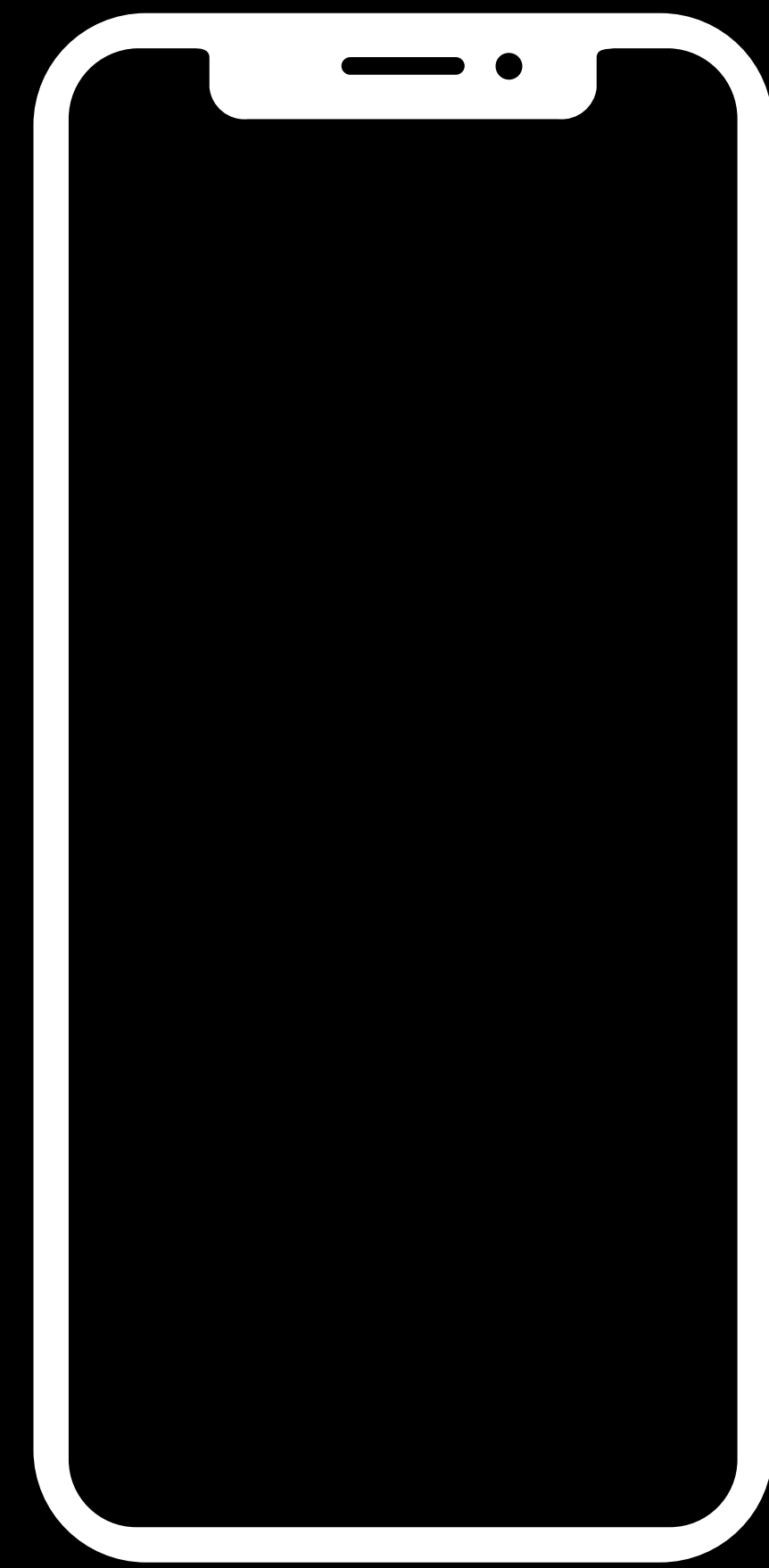
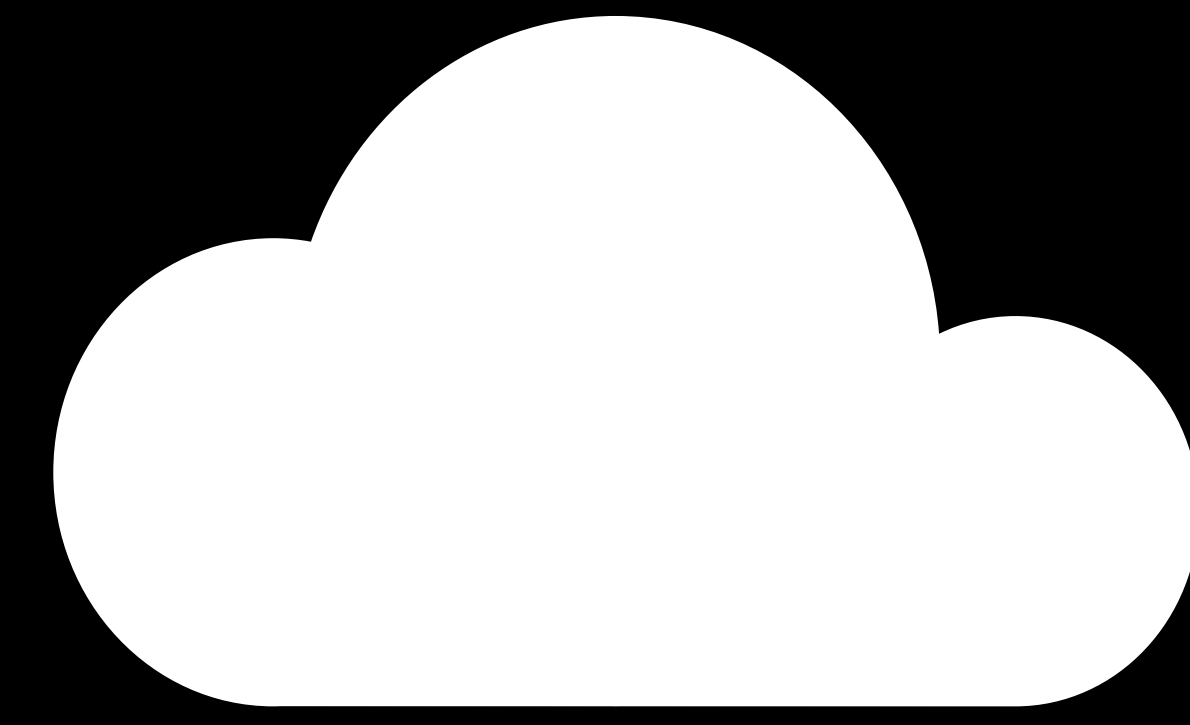


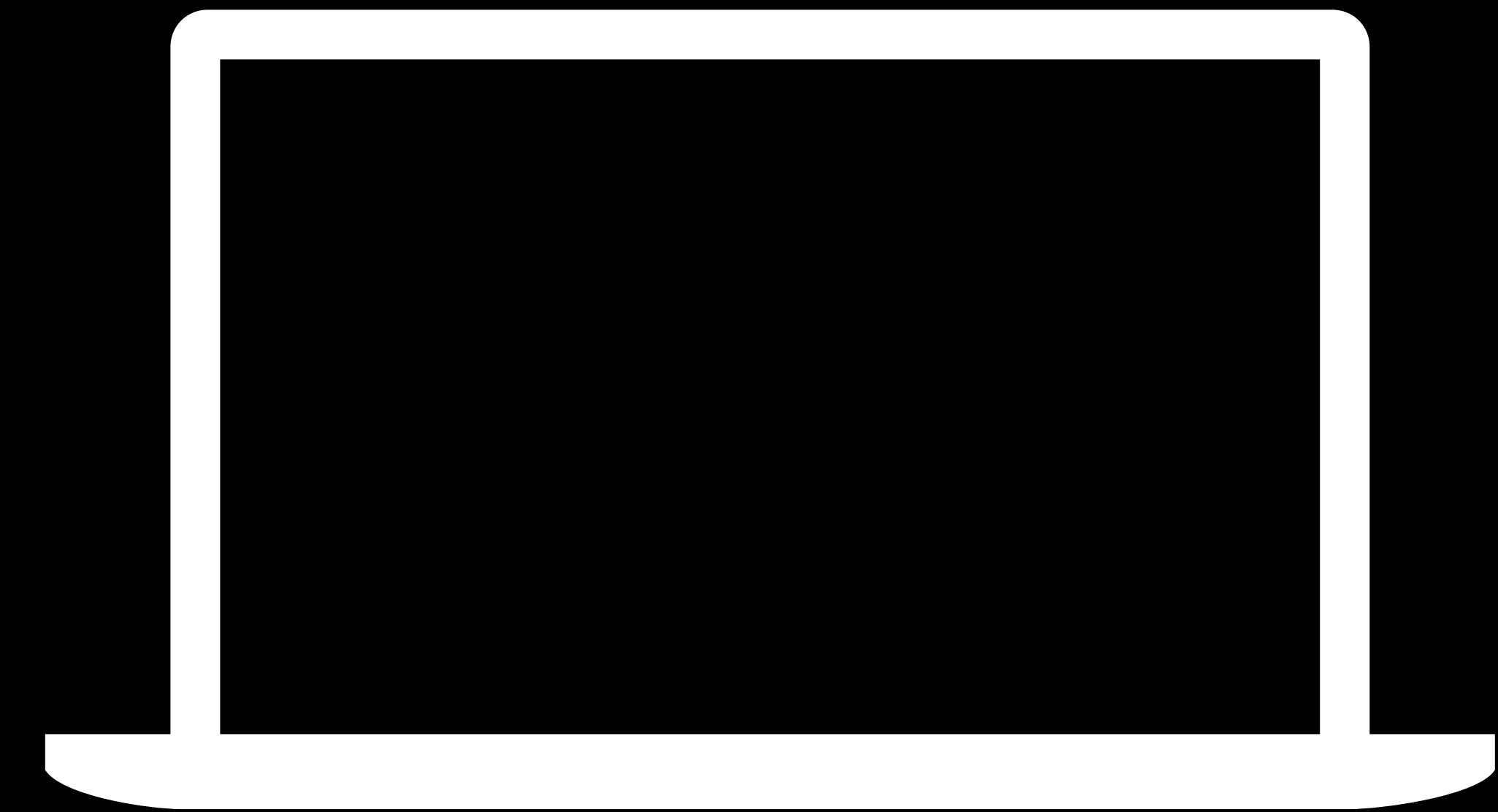
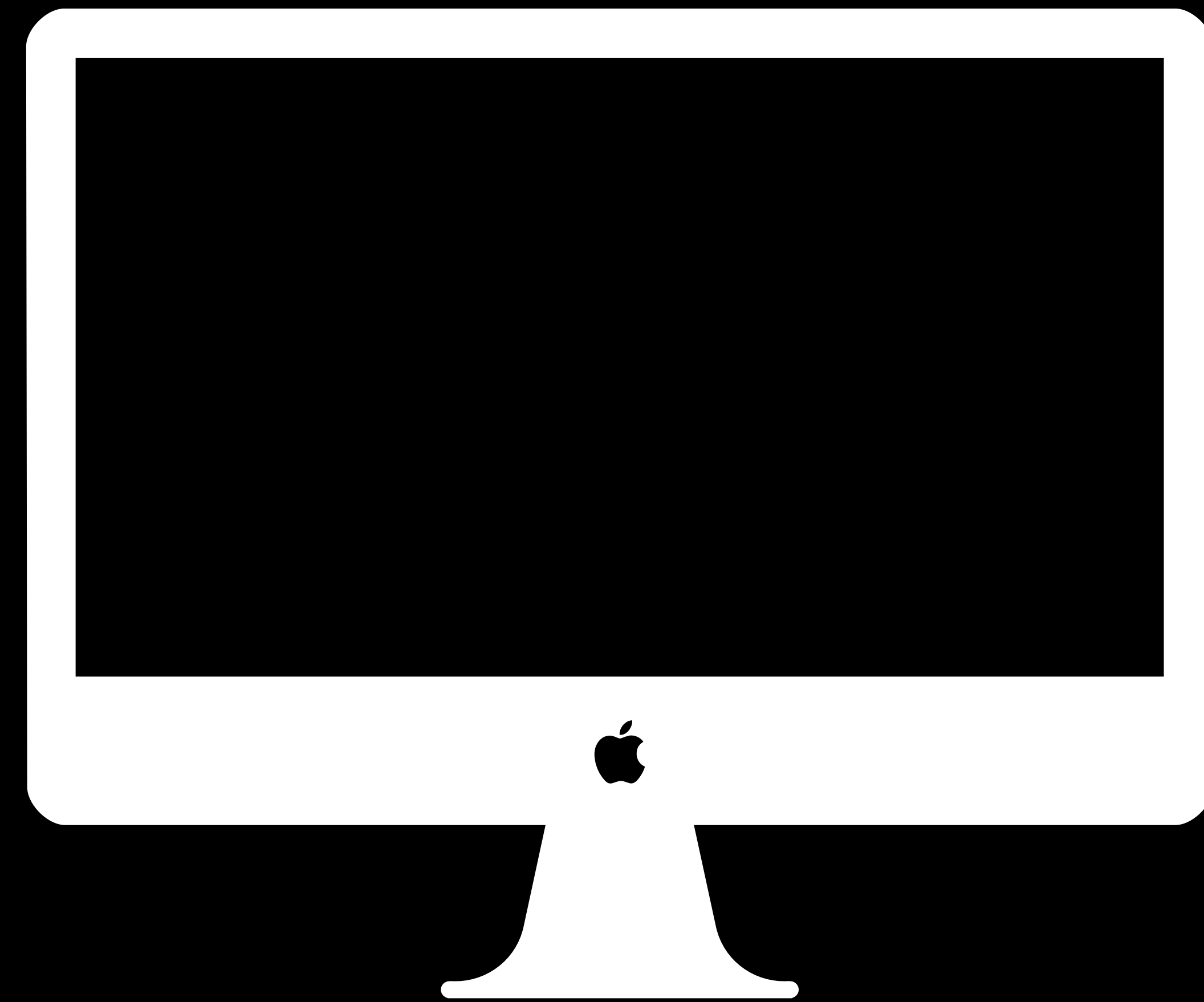
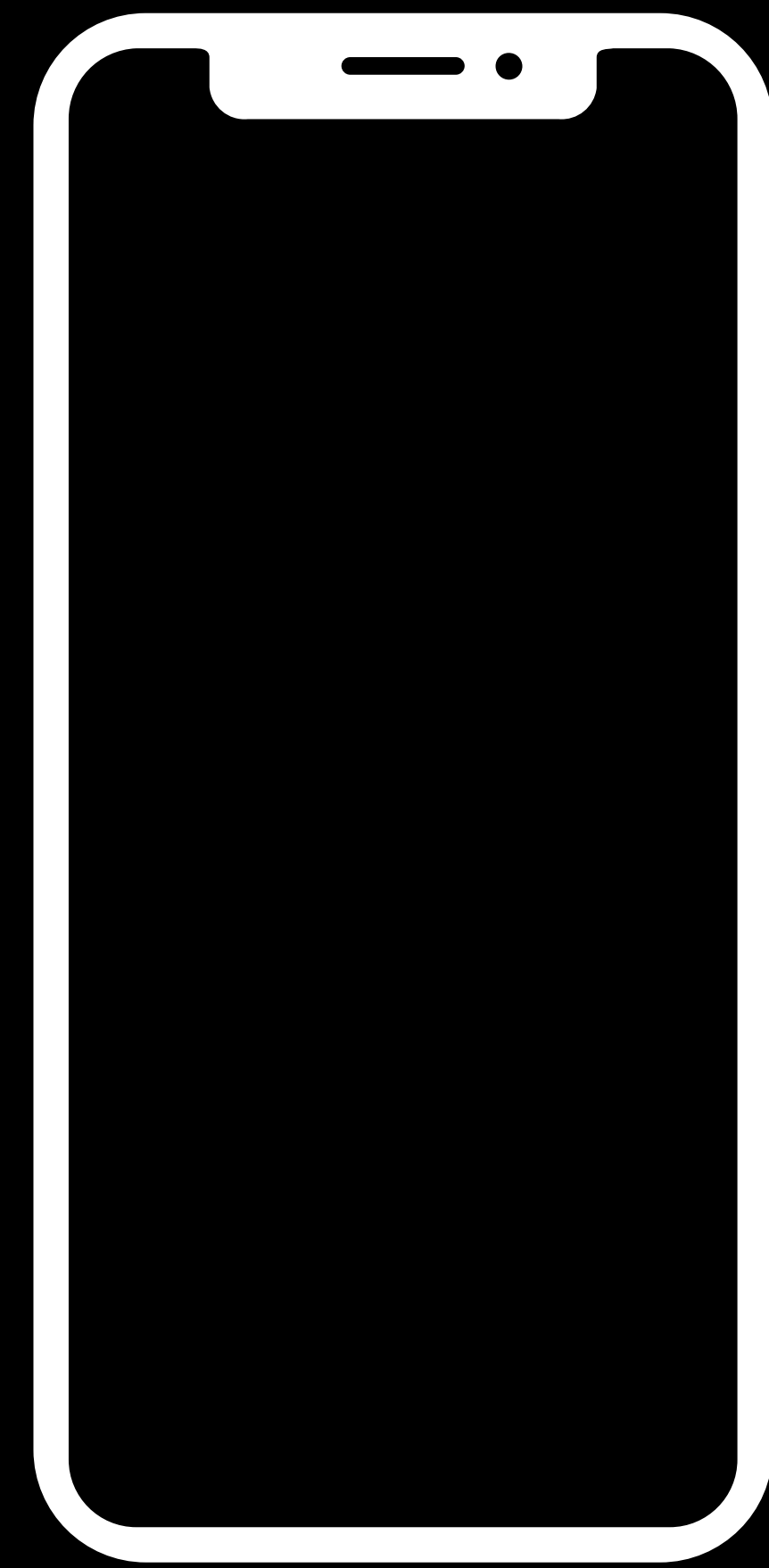
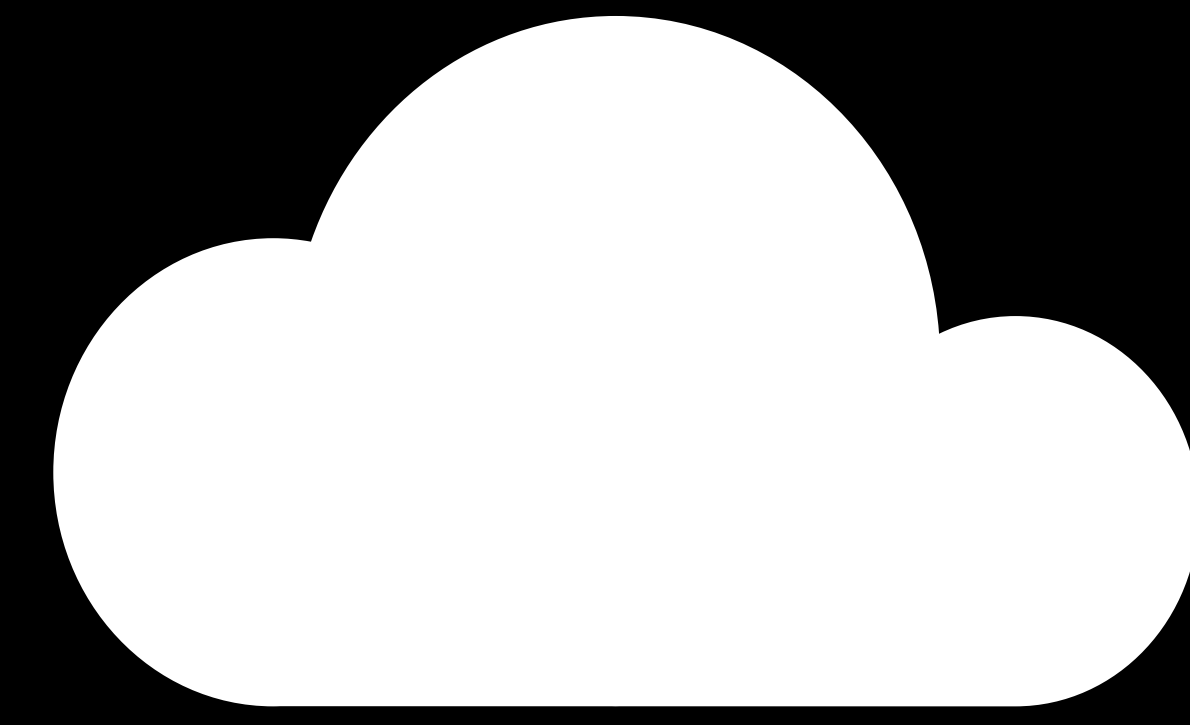


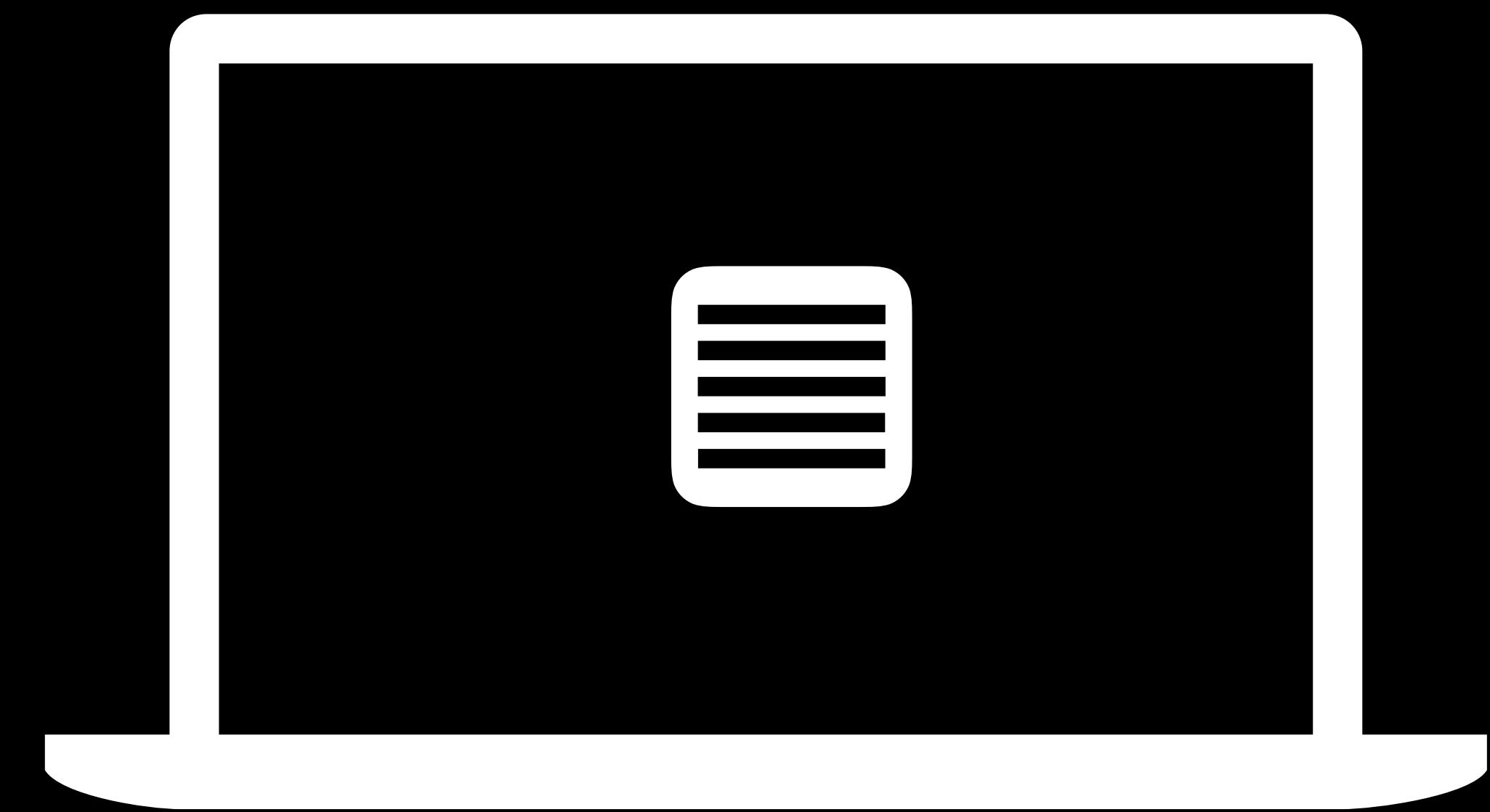
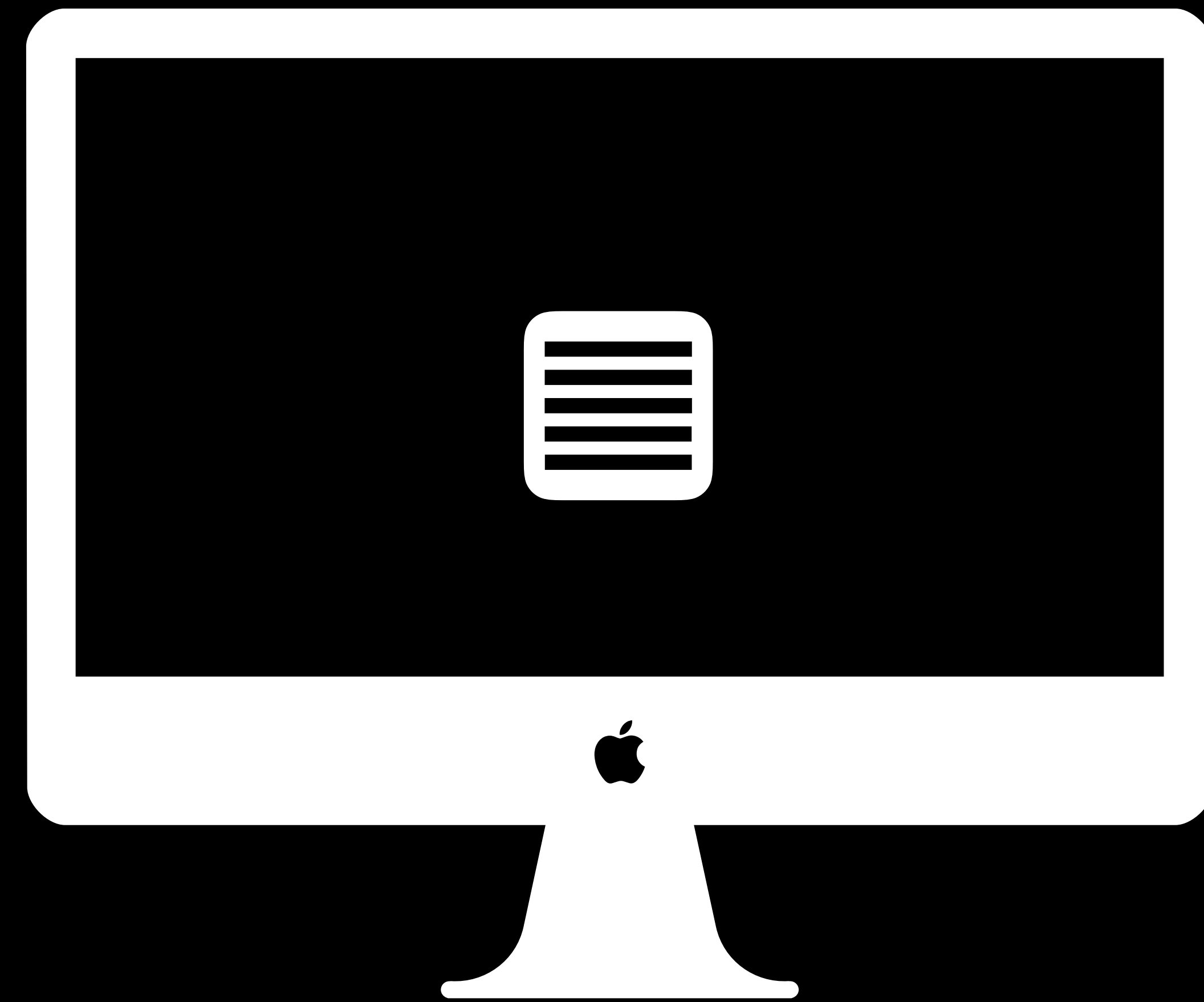
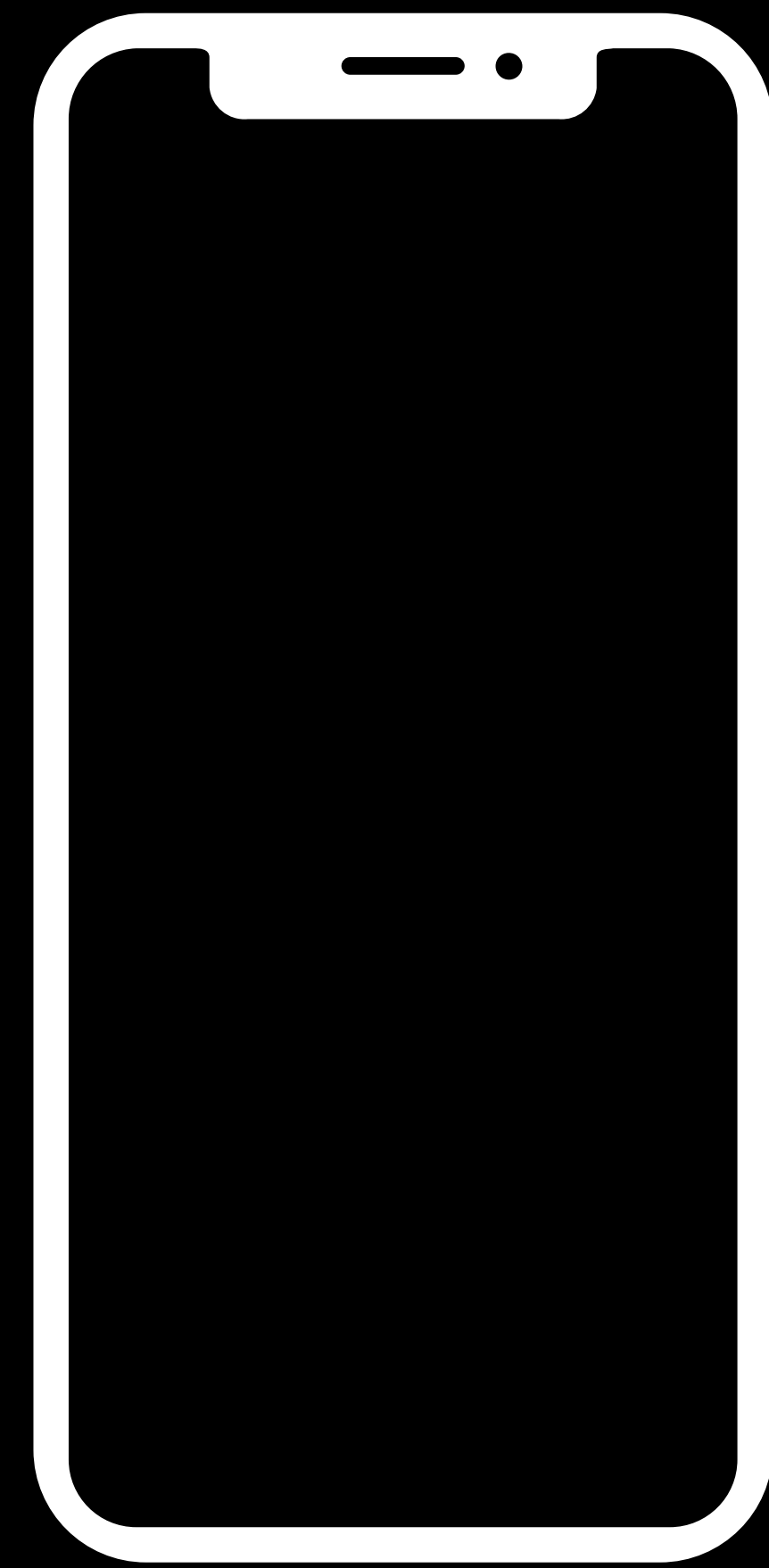
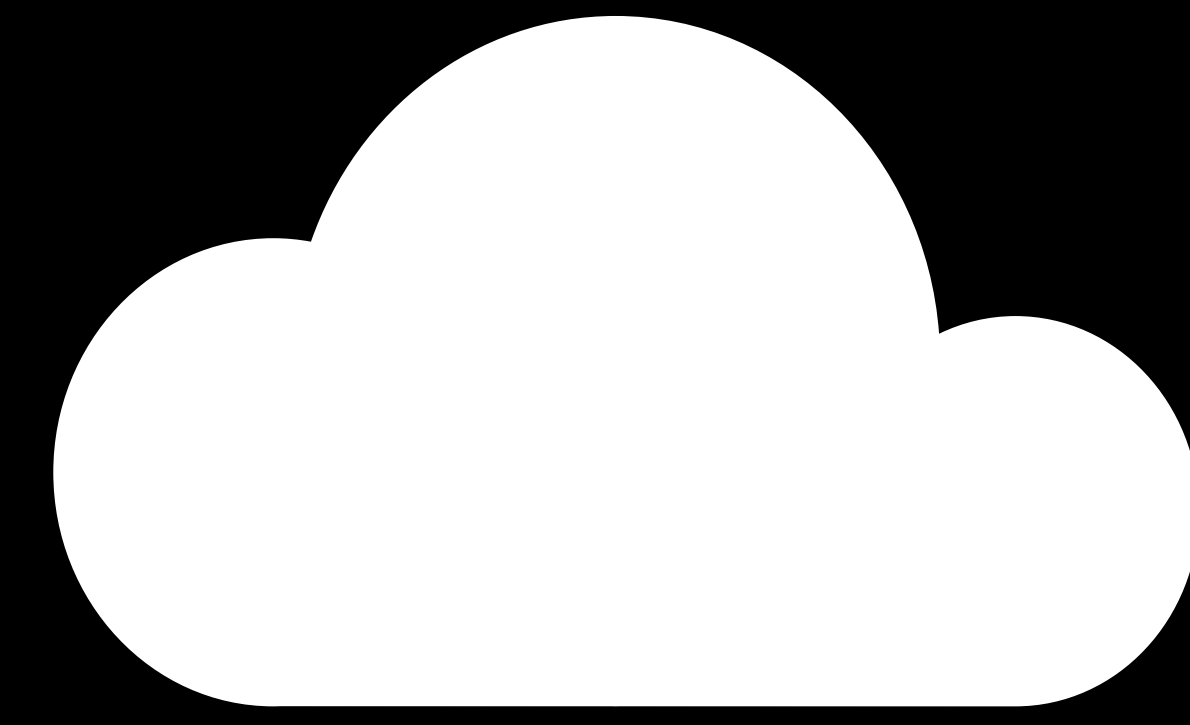


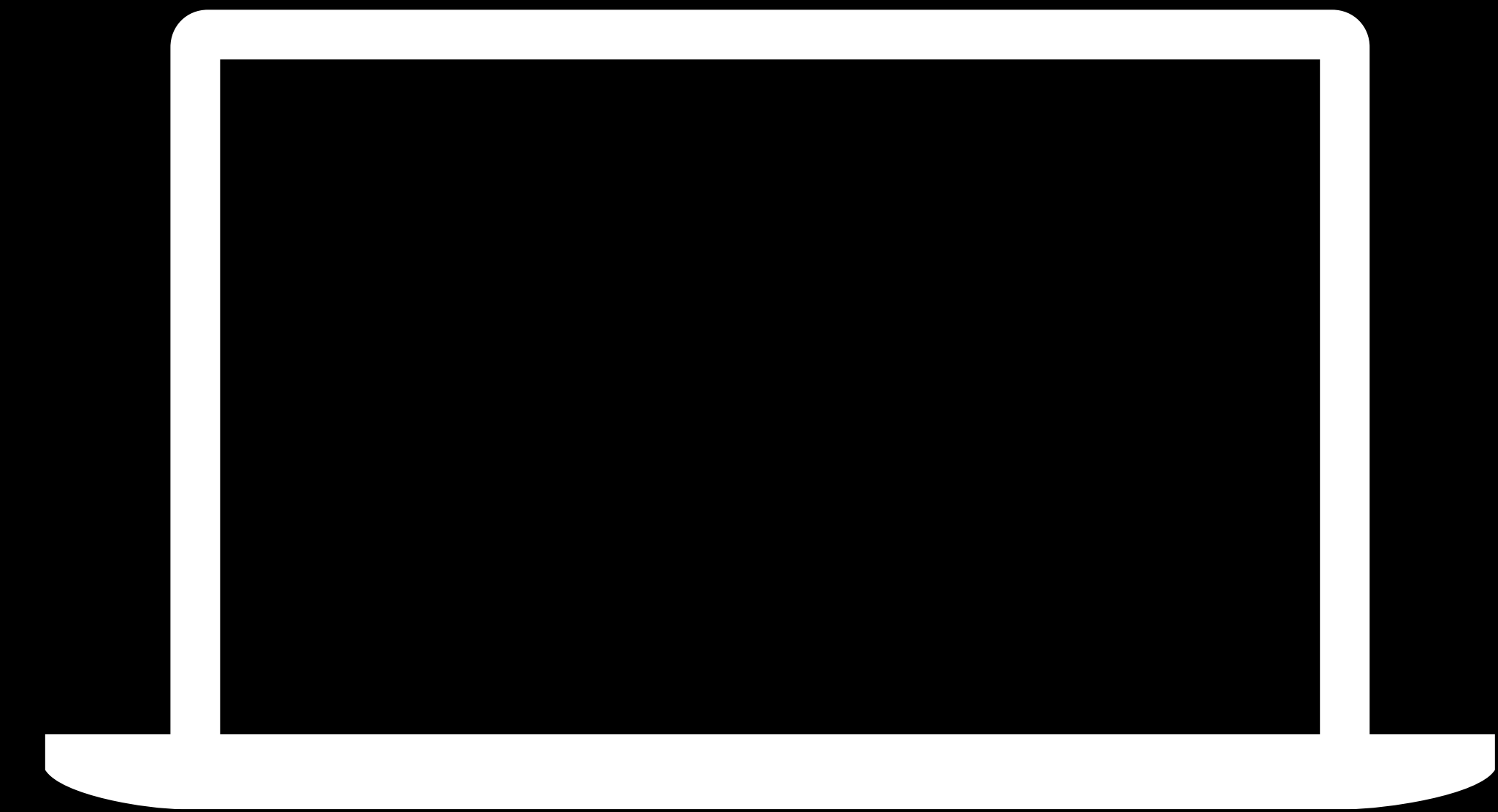
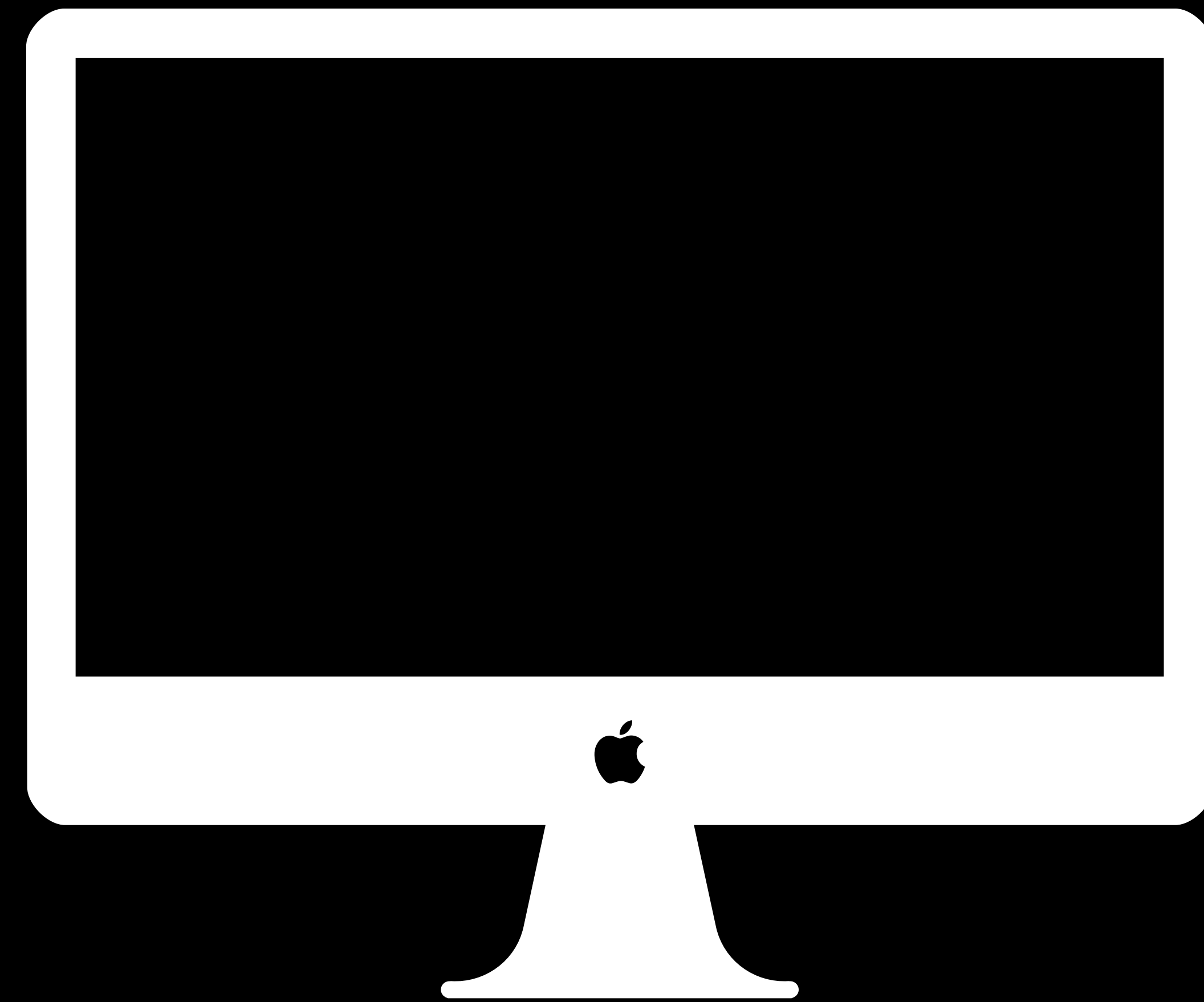
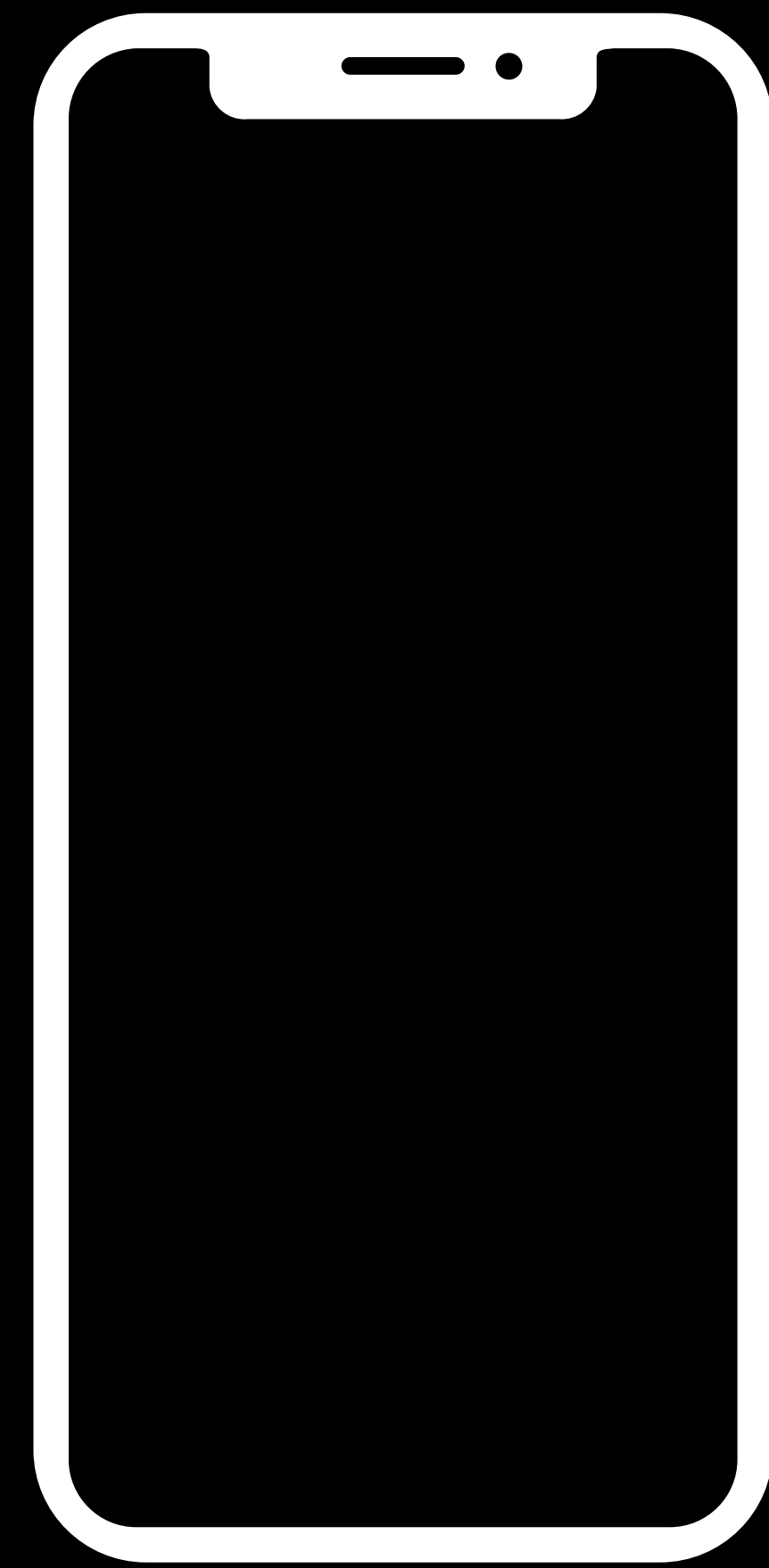
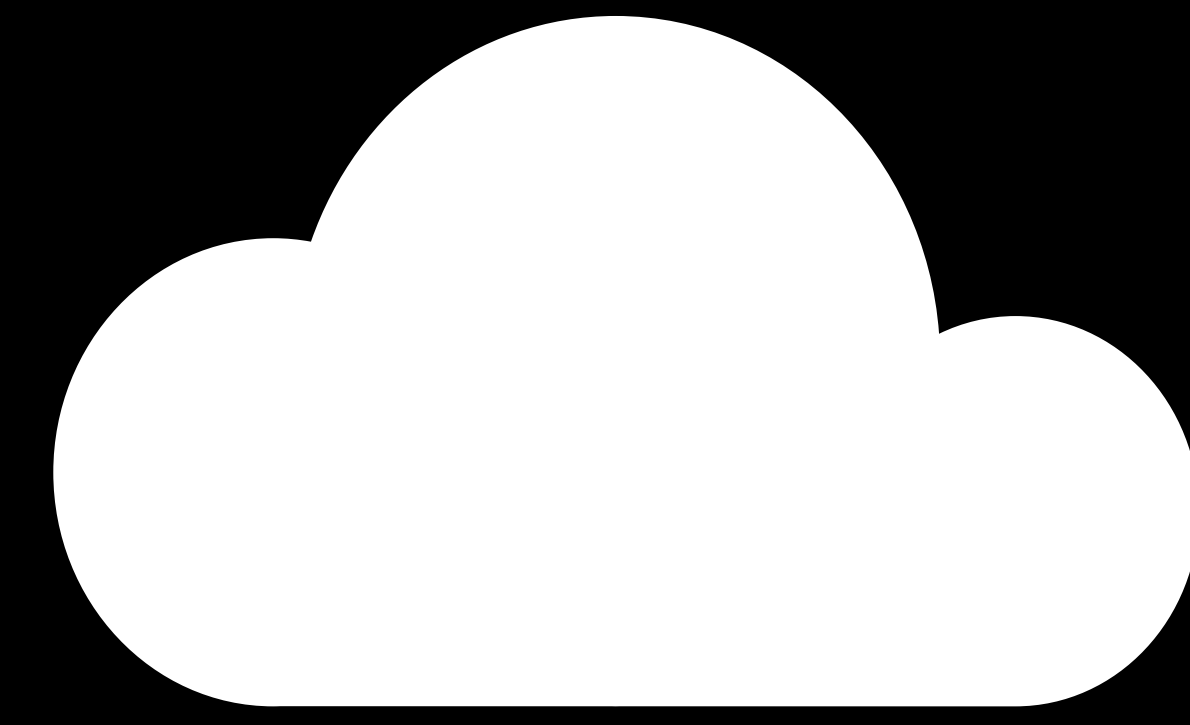




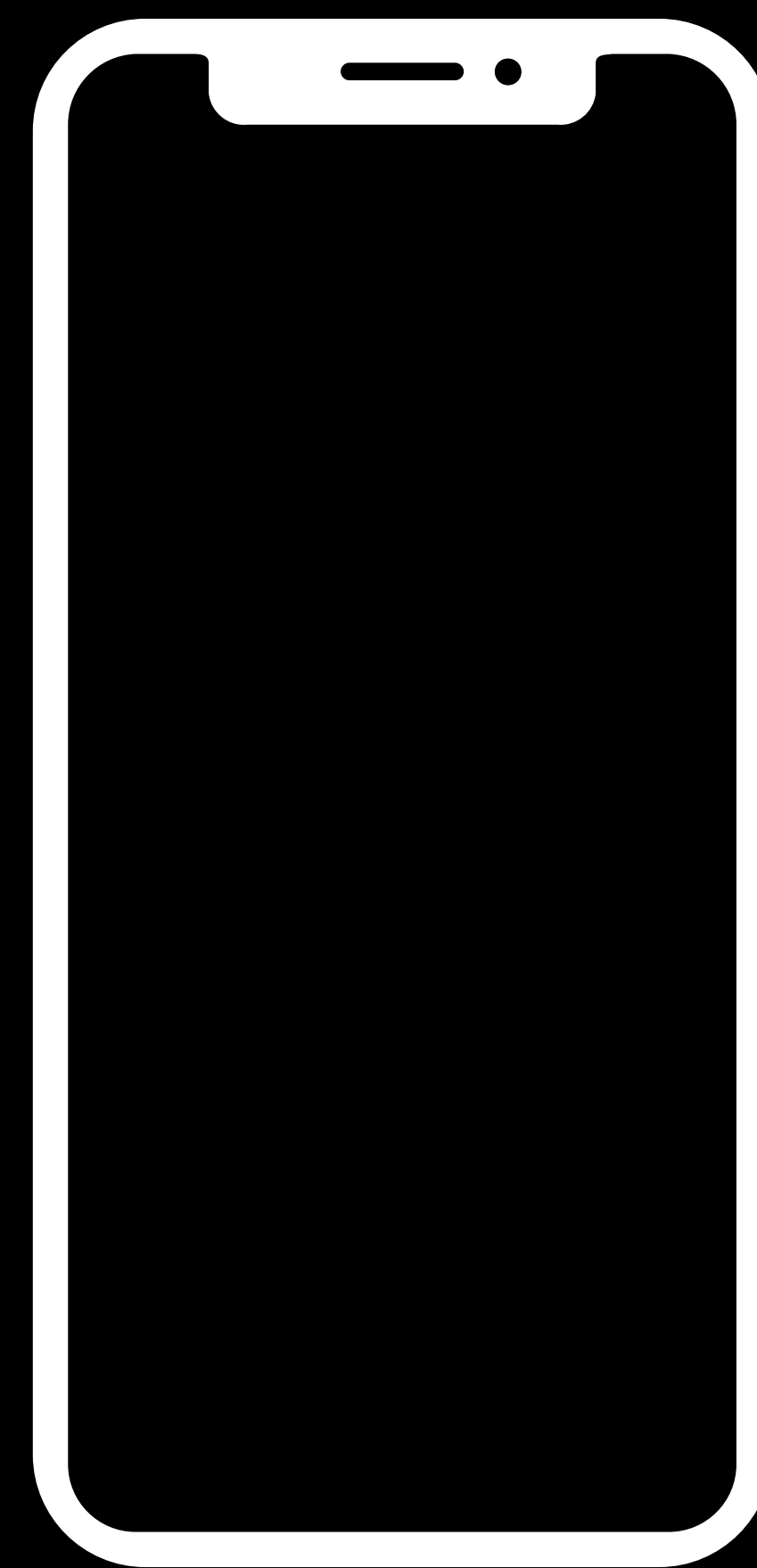
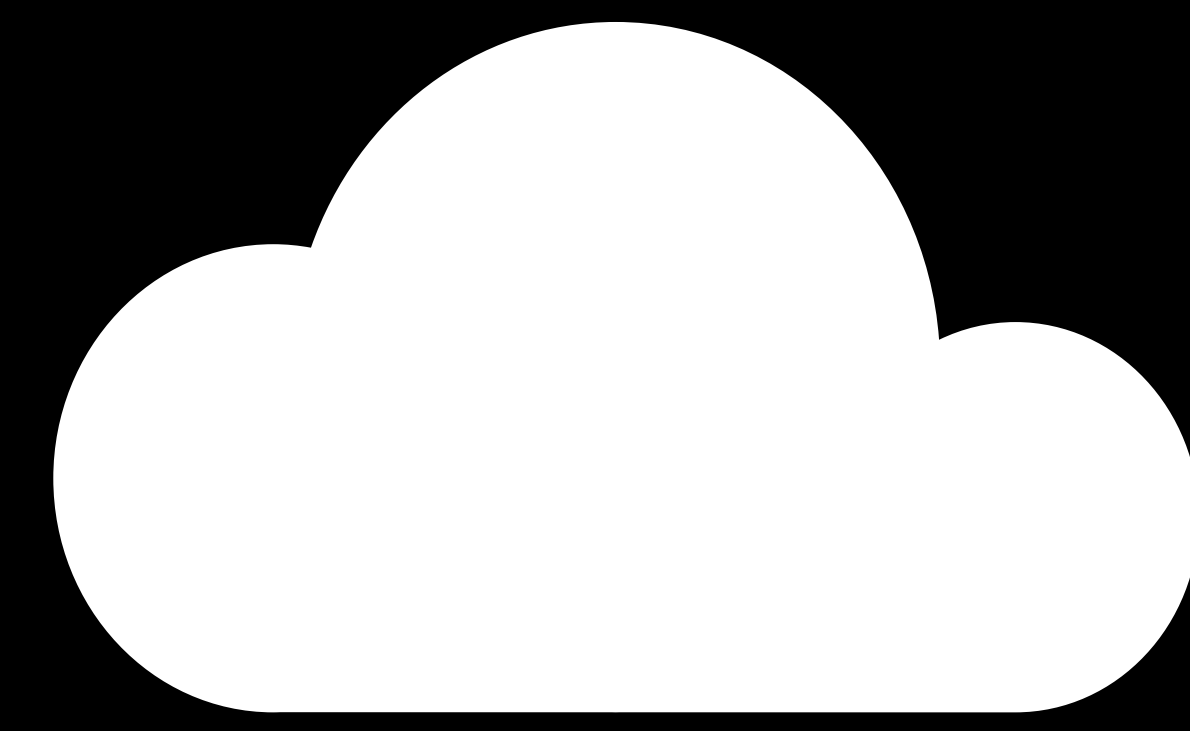


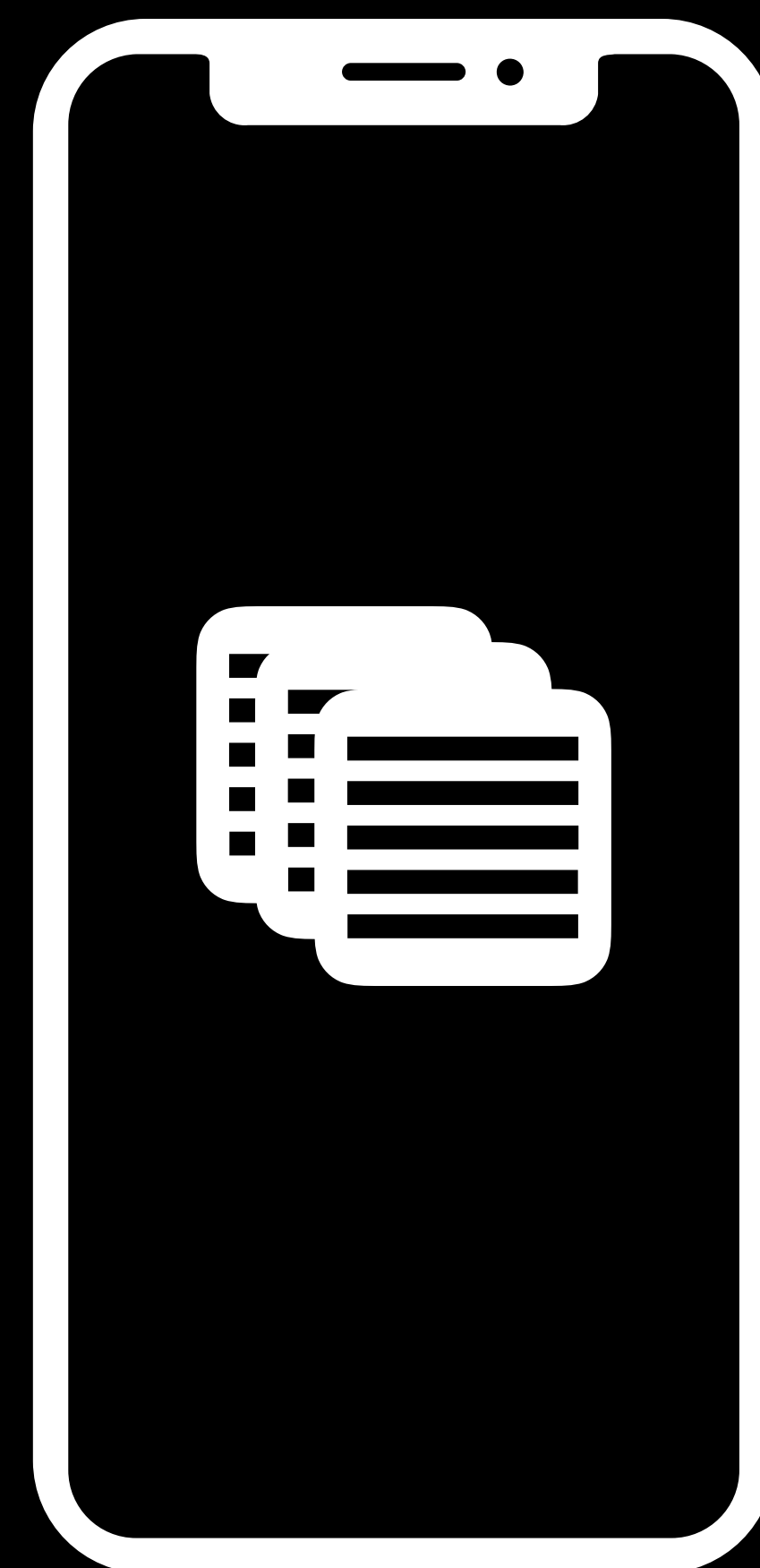
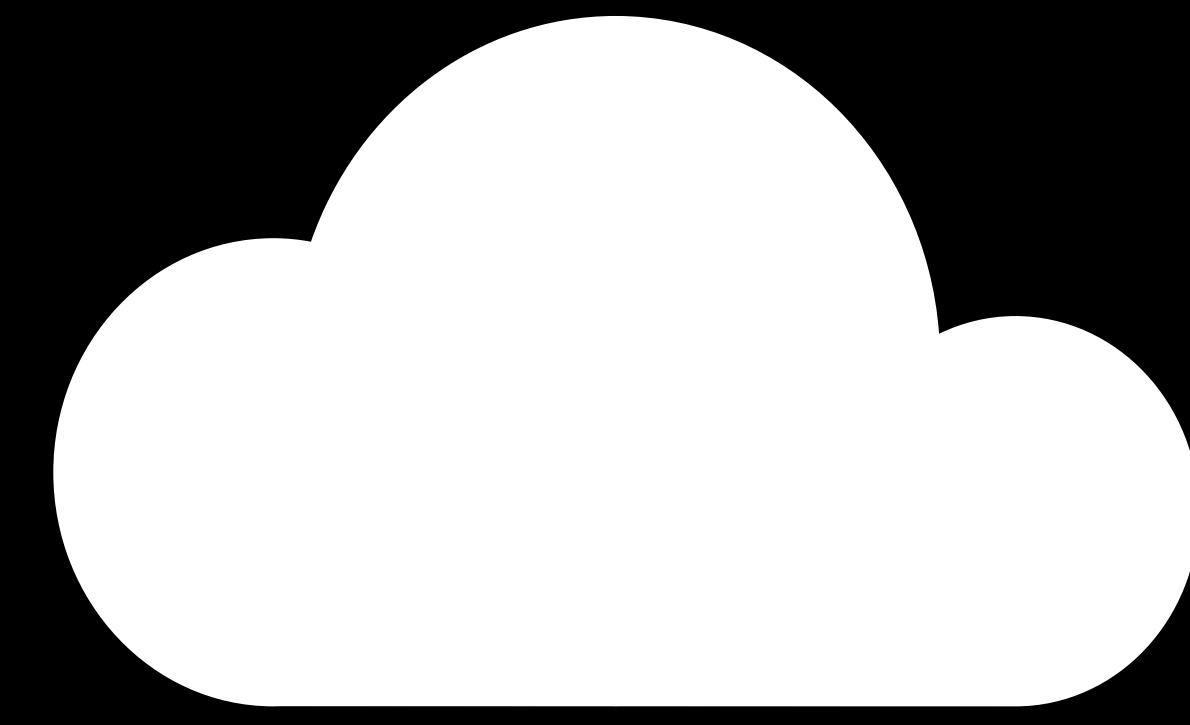


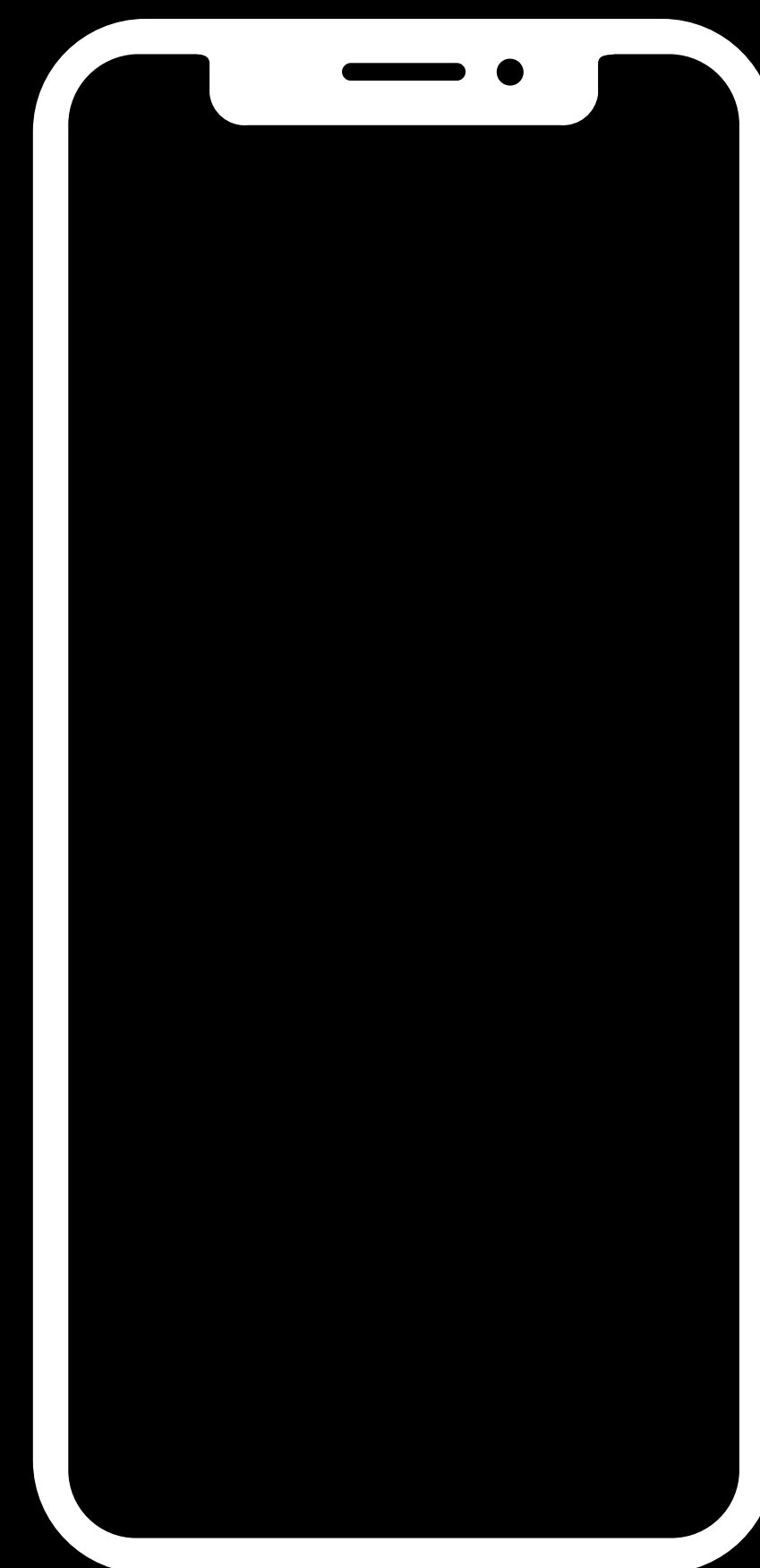
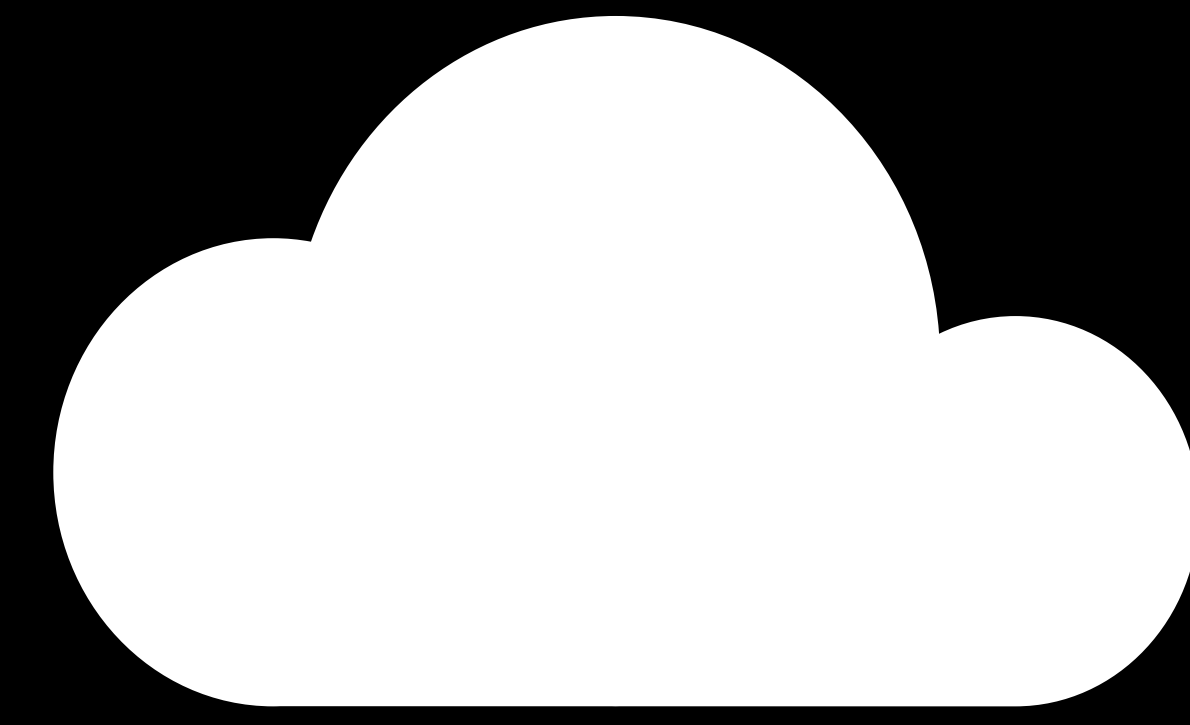


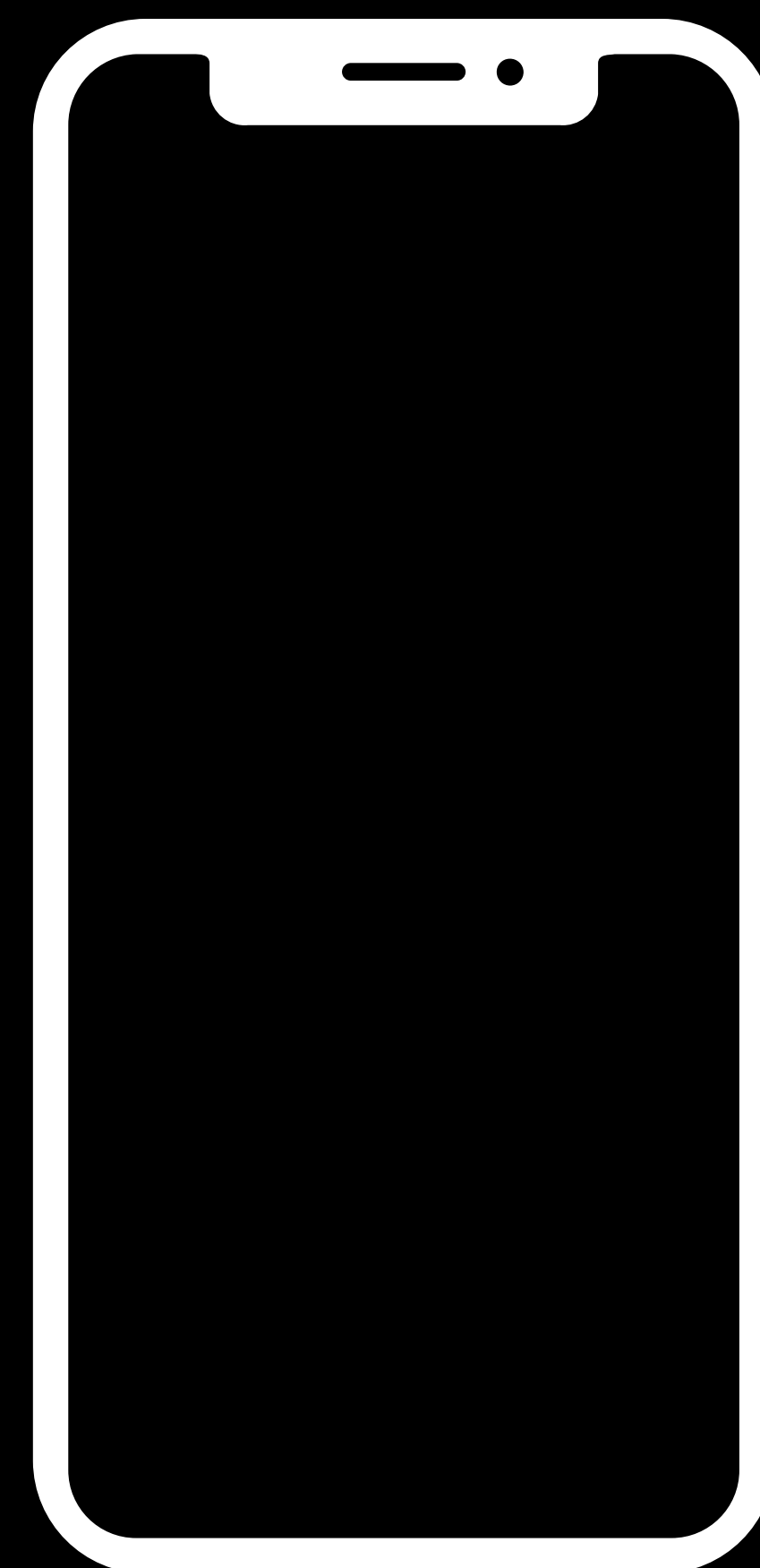
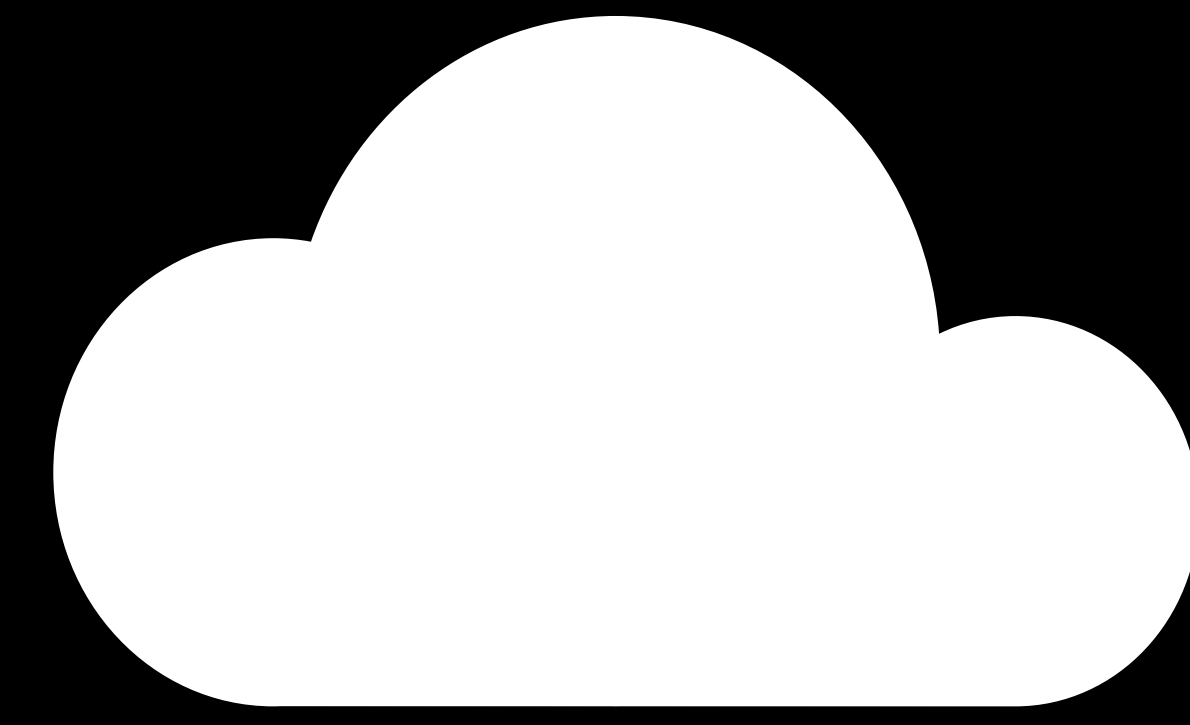


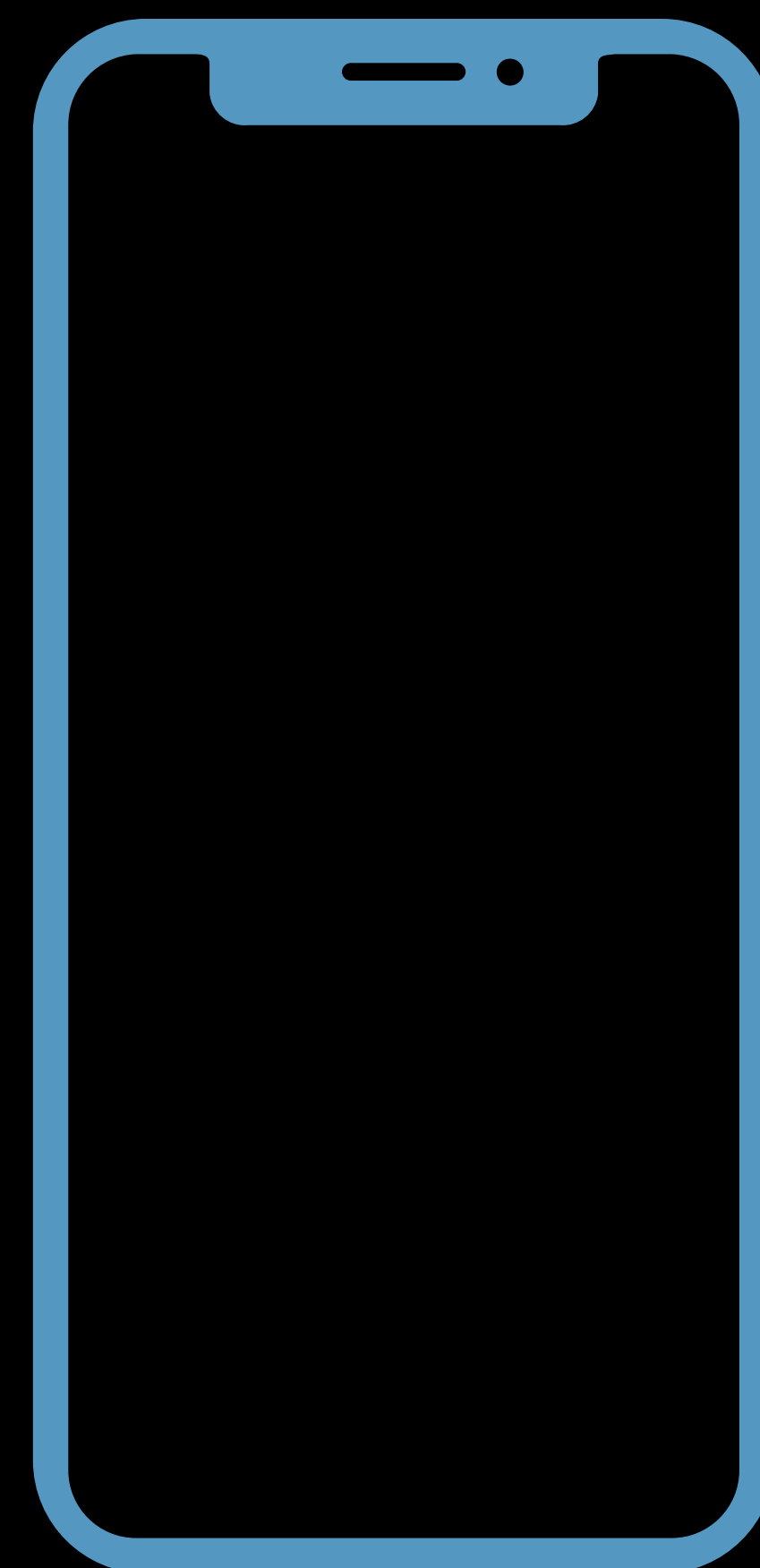
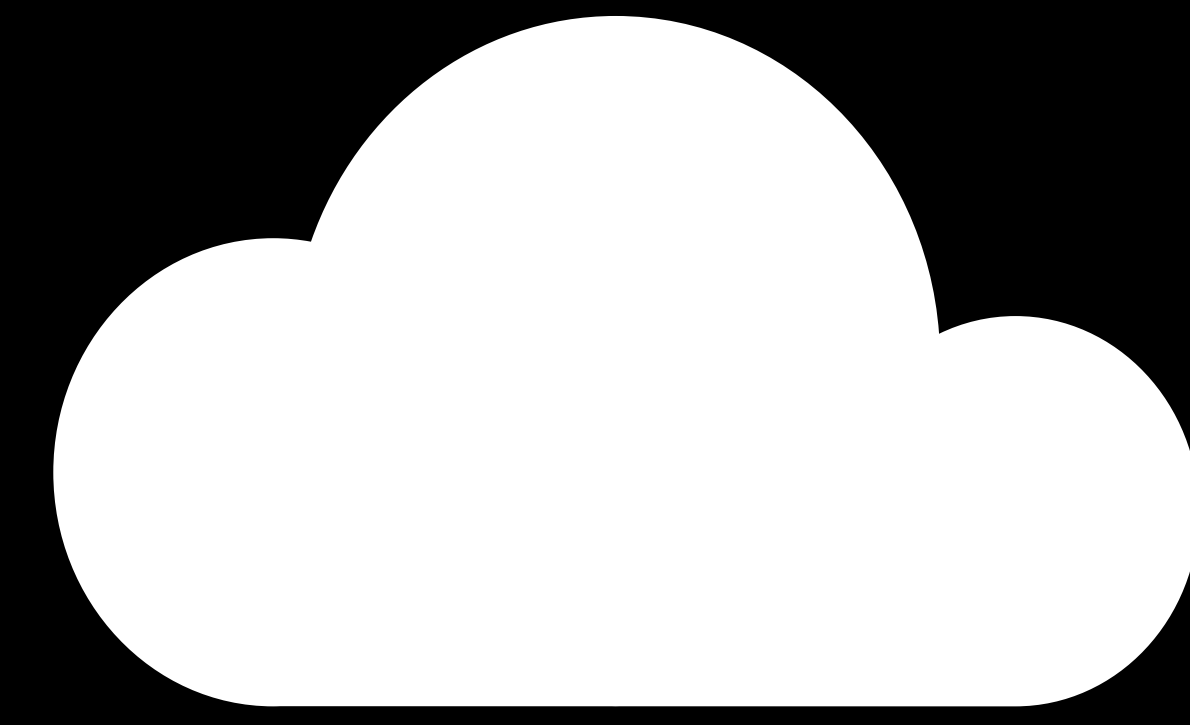


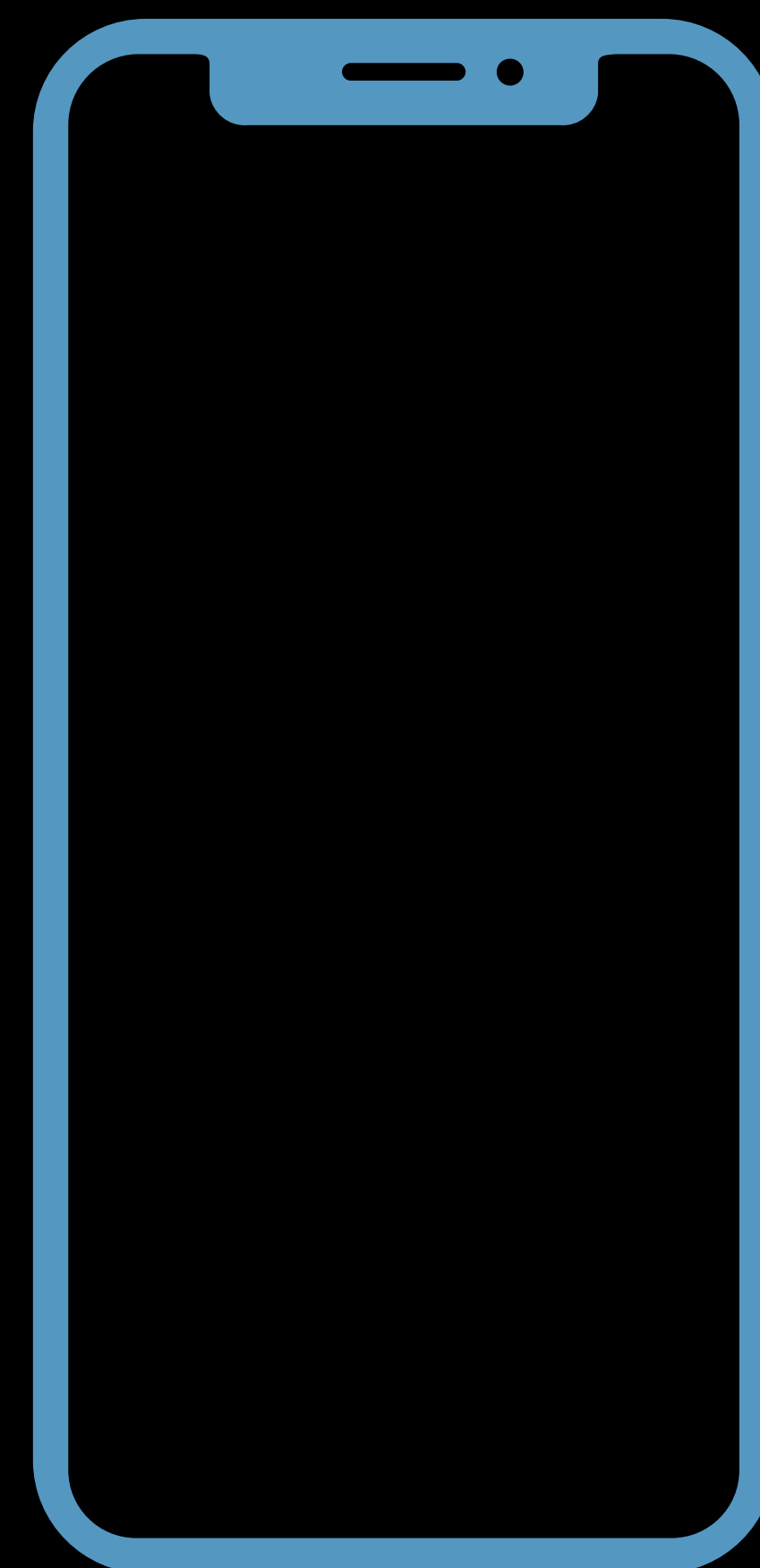
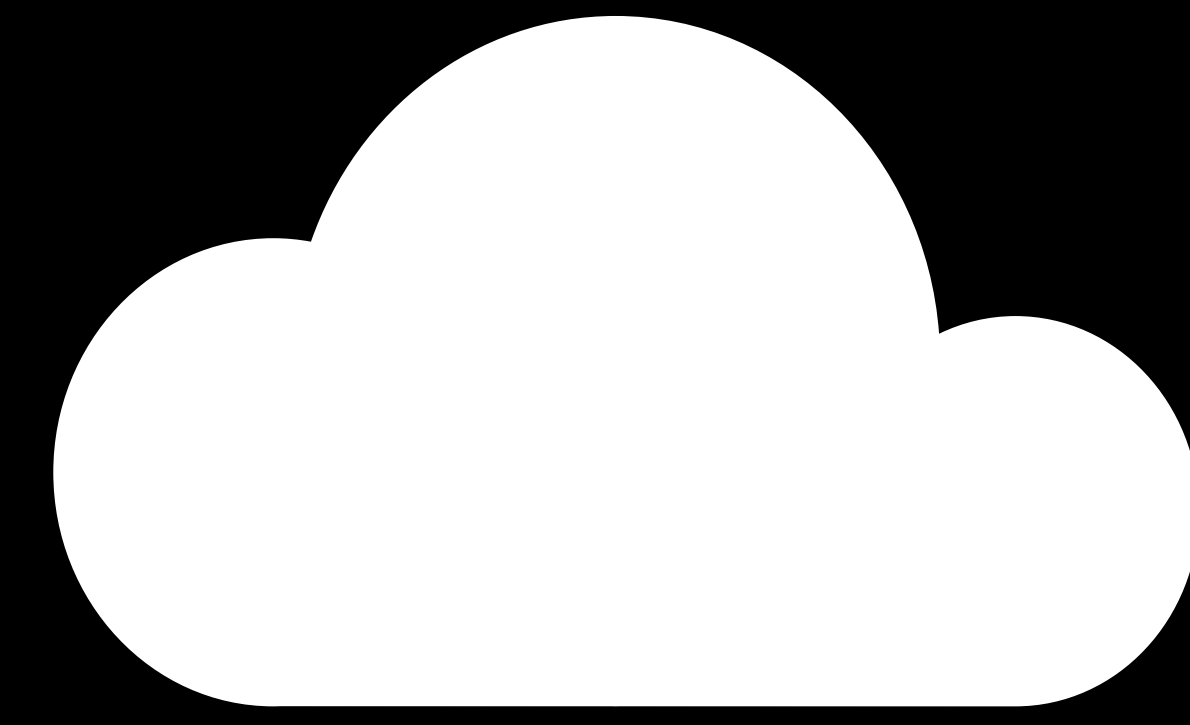


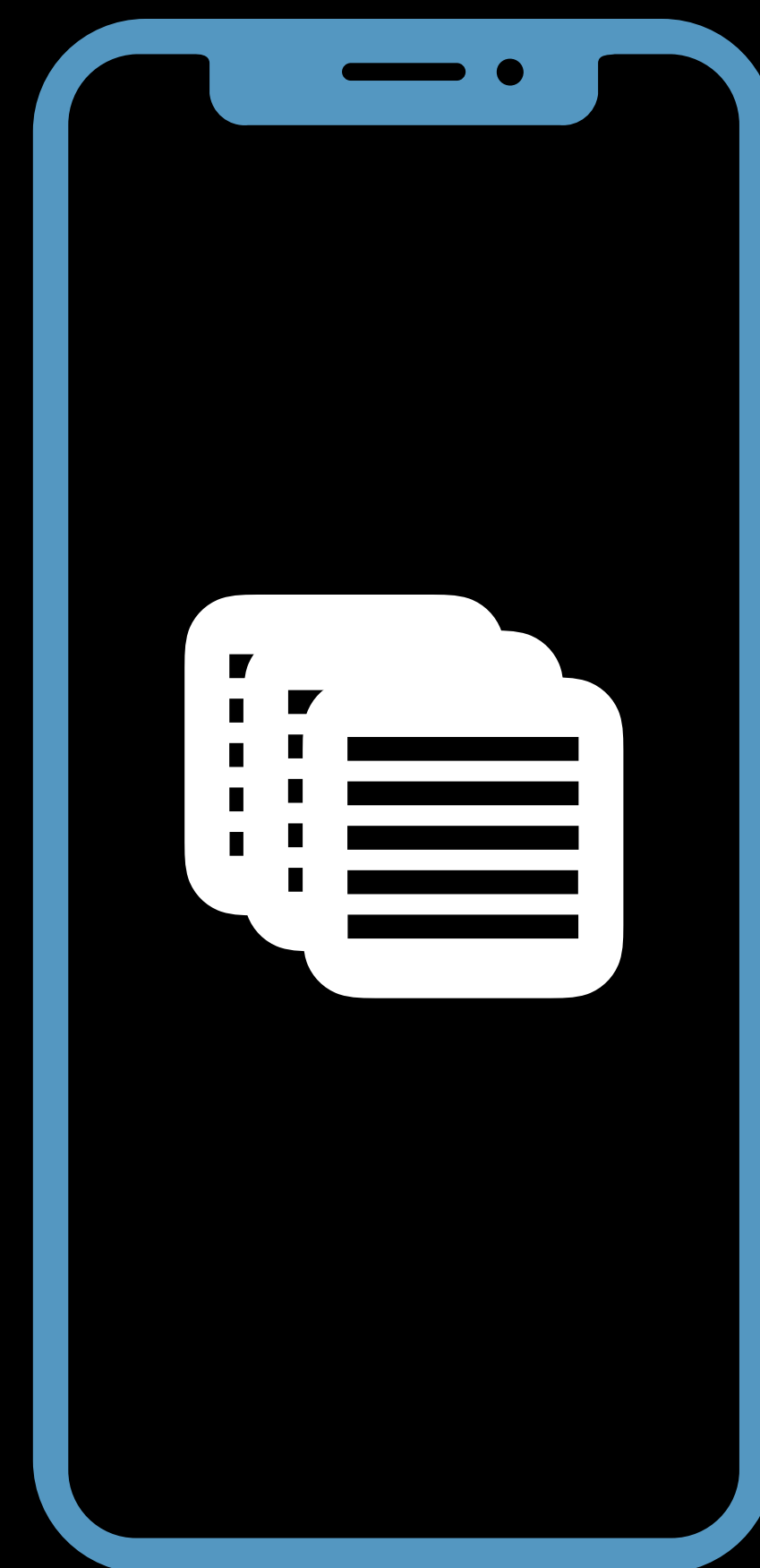
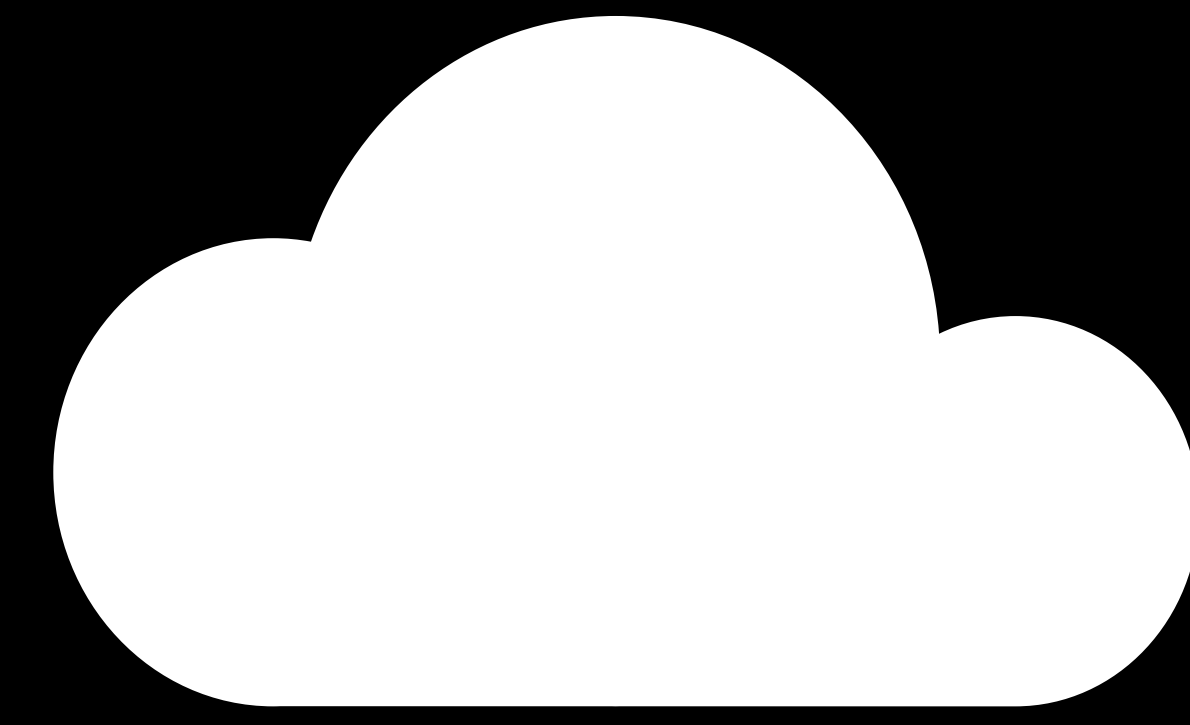












Existing Technologies

Existing Technologies

Core Data provides local persistence

Existing Technologies

Core Data provides local persistence

CloudKit provides distributed persistence

Existing Technologies

Core Data provides local persistence

CloudKit provides distributed persistence

Both exist on all platforms

Existing Technologies

Core Data provides local persistence

CloudKit provides distributed persistence

Both exist on all platforms

Both support a wide variety of applications

Existing Technologies

Existing Technologies

Objects

Models

Stores

Existing Technologies

	Core Data	CloudKit
Objects		
Models		
Stores		

Existing Technologies

	Core Data	CloudKit
Objects	NSManagedObject	
Models		
Stores		

Existing Technologies

	Core Data	CloudKit
Objects	NSManagedObject	CKRecord
Models		
Stores		

Existing Technologies

	Core Data	CloudKit
Objects	NSManagedObject	CKRecord
Models	NSManagedObjectModel	
Stores		

Existing Technologies

	Core Data	CloudKit
Objects	NSManagedObject	CKRecord
Models	NSManagedObjectModel	Schema
Stores		

Existing Technologies

	Core Data	CloudKit
Objects	NSManagedObject	CKRecord
Models	NSManagedObjectModel	Schema
Stores	NSPersistentStore	

Existing Technologies

	Core Data	CloudKit
Objects	NSManagedObject	CKRecord
Models	NSManagedObjectModel	Schema
Stores	NSPersistentStore	CKRecordZone / CKDatabase



Welcome to Xcode

Version 11.0 beta (11L335v)

-  **Get started with a playground**
Explore new ideas quickly and easily.
-  **Create a new Xcode project**
Create an app for iPhone, iPad, Mac, Apple Watch, or Apple TV.
-  **Clone an existing project**
Start working on something from a Git repository.

No Recent Projects

[Open another project...](#)



Welcome to Xcode

Version 11.0 beta (11L335v)

-  **Get started with a playground**
Explore new ideas quickly and easily.
-  **Create a new Xcode project**
Create an app for iPhone, iPad, Mac, Apple Watch, or Apple TV.
-  **Clone an existing project**
Start working on something from a Git repository.





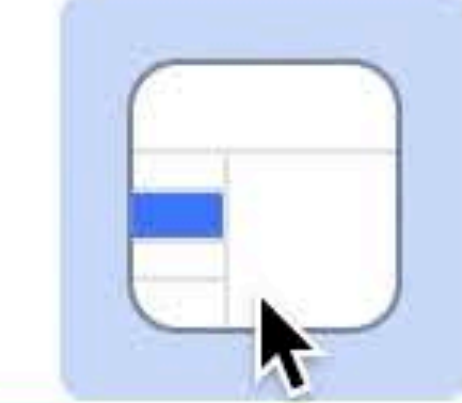



No Recent Projects

[Open another project...](#)




Choose a template for your new project:

ios watchOS tvOS macOS Linux Cross-platform Filter

Application

 Single View App	 Game	 Augmented Reality App	 Document Based App	 Master-Detail App
 Tabbed App	 Sticker Pack App	 iMessage App		

Framework & Library

 iOS Framework	 iOS Static Library	 Metal Library
---	---	--

Cancel Previous Next

No Selection

Choose a template for your new project:

Platform: **iOS** watchOS tvOS macOS Linux Cross-platform Filter

Application

- Single View App
- Game
- Augmented Reality App
- Document Based App
- Master-Detail App**
- Tabbed App
- Sticker Pack App
- iMessage App

Framework & Library

- iOS Framework
- iOS Static Library
- Metal Library

Cancel

Previous

Next

No Selection

Choose options for your new project:

Product Name: WWDCDemo

Team: Apple Inc. - Cocoa

Organization Name: Demo

Organization Identifier: com.apple.wwdc

Bundle Identifier: com.apple.wwdc.WWDCDemo

Language: Swift

- Use Core Data
- Use CloudKit
- Include Unit Tests
- Include UI Tests

Cancel

Previous

Next

No Selection

Choose options for your new project:

Product Name: WWDCDemo

Team: Apple Inc. - Cocoa

Organization Name: Demo

Organization Identifier: com.apple.wwdc

Bundle Identifier: com.apple.wwdc.WWDCDemo

Language: Swift

- Use Core Data
- Use CloudKit
- Include Unit Tests
- Include UI Tests

Cancel

Previous

Next

No Selection

WWDCDemo

- WWDCDemo
 - AppDelegate.swift
 - MasterViewController.swift
 - DetailViewController.swift
 - Main.storyboard
 - Assets.xcassets
 - LaunchScreen.storyboard
 - Info.plist
 - WWDCDemo.xcdatamodeld
- WWDCDemoTests
 - WWDCDemoTests.swift
 - Info.plist
- WWDCDemoUITests
 - WWDCDemoUITests.swift
 - Info.plist
- Products

WWDCDemo

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

PROJECT

- WWDCDemo

TARGETS

- WWDCDemo
- WWDCDemoTests
- WWDCDemoUITests

Identity

Display Name: WWDCDemo

Bundle Identifier: com.apple.wwdc.WWDCDemo

Version: 1.0

Build: 1

Deployment Info

Target	Device
iOS 13.0	<input checked="" type="checkbox"/> iPhone
	<input checked="" type="checkbox"/> iPad
	<input type="checkbox"/> Mac (on macOS 10.15)

Main Interface: Main

Device Orientation: Portrait, Upside Down, Landscape Left, Landscape Right

Status Bar Style: Default

Hide status bar

Requires full screen

Supports multiple windows

App Icons and Launch Images

App Icons Source: AppIcon

Launch Images Source: Use Asset Catalog...

Launch Screen File: LaunchScreen

Frameworks, Libraries, and Embedded Content

Name	Embed
------	-------

Identity and Type

Name: WWDCDemo

Location: Absolute

WWDCDemo.xcodeproj

Full Path: /Users/demo/Desktop/WWDCDemo/WWDCDemo.xcodeproj

Project Document

Project Format: Xcode 9.3-compatible

Organization: Demo

Class Prefix:

Text Settings

Indent Using: Spaces

Widths: Tab: 4, Indent: 4

Wrap lines

WWDCDemo

- WWDCDemo
 - AppDelegate.swift
 - MasterViewController.swift
 - DetailViewController.swift
 - Main.storyboard
 - Assets.xcassets
 - LaunchScreen.storyboard
 - Info.plist
 - WWDCDemo.xcdatamodeld
- WWDCDemoTests
 - WWDCDemoTests.swift
 - Info.plist
- WWDCDemoUITests
 - WWDCDemoUITests.swift
 - Info.plist
- Products

WWDCDemo

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

PROJECT

- WWDCDemo

TARGETS

- WWDCDemo
- WWDCDemoTests
- WWDCDemoUITests

Identity

Display Name: WWDCDemo

Bundle Identifier: com.apple.wwdc.WWDCDemo

Version: 1.0

Build: 1

Deployment Info

Target	Device
iOS 13.0	<input checked="" type="checkbox"/> iPhone
	<input checked="" type="checkbox"/> iPad
	<input type="checkbox"/> Mac (on macOS 10.15)

Main Interface: Main

Device Orientation: Portrait, Upside Down, Landscape Left, Landscape Right

Status Bar Style: Default

Hide status bar

Requires full screen

Supports multiple windows

App Icons and Launch Images

App Icons Source: AppIcon

Launch Images Source: Use Asset Catalog...

Launch Screen File: LaunchScreen

Frameworks, Libraries, and Embedded Content

Name	Embed
------	-------

Identity and Type

Name: WWDCDemo

Location: Absolute

WWDCDemo.xcodeproj

Full Path: /Users/demo/Desktop/WWDCDemo/WWDCDemo.xcodeproj

Project Document

Project Format: Xcode 9.3-compatible

Organization: Demo

Class Prefix:

Text Settings

Indent Using: Spaces

Widths: Tab: 4, Indent: 4

Wrap lines

WWDCDemo

- WWDCDemo
 - AppDelegate.swift
 - MasterViewController.swift
 - DetailViewController.swift
 - Main.storyboard
 - Assets.xcassets
 - LaunchScreen.storyboard
 - Info.plist
 - WWDCDemo.xcdatamodeld
- WWDCDemoTests
 - WWDCDemoTests.swift
 - Info.plist
- WWDCDemoUITests
 - WWDCDemoUITests.swift
 - Info.plist
- Products

+ - Filter

WWDCDemo

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

+ Capability All Debug Release

PROJECT

- WWDCDemo

TARGETS

- WWDCDemo
- WWDCDemoTests
- WWDCDemoUITests

Signing

- Automatically manage signing
Xcode will create and update profiles, app IDs, and certificates.
- Team: Apple Inc. - Cocoa
- Bundle Identifier: com.apple.wwdc.WWDCDemo
- Provisioning Profile: Xcode Managed Profile ⓘ
- Signing Certificate: Apple Development: Nick Gillett (3U5Q9B57E4)

Add capabilities by clicking the "+" button above.

+ - Filter

Identity and Type

- Name: WWDCDemo
- Location: Absolute
- WWDCDemo.xcodeproj
- Full Path: /Users/demo/Desktop/WWDCDemo/WWDCDemo.xcodeproj

Project Document

- Project Format: Xcode 9.3-compatible
- Organization: Demo
- Class Prefix:

Text Settings

- Indent Using: Spaces
- Widths: Tab: 4, Indent: 4
- Wrap lines

WWDCDemo

- WWDCDemo
 - AppDelegate.swift
 - MasterViewController.swift
 - DetailViewController.swift
 - Main.storyboard
 - Assets.xcassets
 - LaunchScreen.storyboard
 - Info.plist
 - WWDCDemo.xcdatamodeld
- WWDCDemoTests
 - WWDCDemoTests.swift
 - Info.plist
- WWDCDemoUITests
 - WWDCDemoUITests.swift
 - Info.plist
- Products

+ - Filter

WWDCDemo

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

+ Capability **All** Debug Release

PROJECT

- WWDCDemo

TARGETS

- WWDCDemo**
- WWDCDemoTests
- WWDCDemoUITests

Signing

- Automatically manage signing
Xcode will create and update profiles, app IDs, and certificates.
- Team: Apple Inc. - Cocoa
- Bundle Identifier: com.apple.wwdc.WWDCDemo
- Provisioning Profile: Xcode Managed Profile ⓘ
- Signing Certificate: Apple Development: Nick Gillett (3U5Q9B57E4)

Add capabilities by clicking the "+" button above.

+ - Filter

Identity and Type

- Name: WWDCDemo
- Location: Absolute
- WWDCDemo.xcodeproj
- Full Path: /Users/demo/Desktop/WWDCDemo/WWDCDemo.xcodeproj

Project Document

- Project Format: Xcode 9.3-compatible
- Organization: Demo
- Class Prefix:

Text Settings

- Indent Using: Spaces
- Widths: Tab: 4, Indent: 4
- Wrap lines

WWDCDemo

- WWDCDemo
 - WWDCDemo.entitlements
 - AppDelegate.swift
 - MasterViewController.swift
 - DetailViewController.swift
 - Main.storyboard
 - Assets.xcassets
 - LaunchScreen.storyboard
 - Info.plist
 - WWDCDemo.xcdatamodeld
- WWDCDemoTests
 - WWDCDemoTests.swift
 - Info.plist
- WWDCDemoUITests
 - WWDCDemoUITests.swift
 - Info.plist
- Products
- Frameworks

WWDCDemo

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

PROJECT: WWDCDemo

TARGETS: WWDCDemo (selected), WWDCDemoTests, WWDCDemoUITests

Capabilities: + Capability All Debug Release

Signing

- Automatically manage signing
Xcode will create and update profiles, app IDs, and certificates.
- Team: Apple Inc. - Cocoa
- Bundle Identifier: com.apple.wwdc.WWDCDemo
- Provisioning Profile: Xcode Managed Profile ⓘ
- Signing Certificate: Apple Development: Nick Gillett (3U5Q9B57E4)

iCloud

Services

- Key-value storage
- iCloud Documents
- CloudKit

Containers

- Use default container
- Specify custom containers
 - iCloud.com.apple.wwdc.WWDCDemo iCloud.
 - com.apple.coredata.cdutil
 - com.apple.sidenotes

+ ⓘ

CloudKit Dashboard

Push Notifications

Add capabilities by clicking the "+" button above.

Identity and Type

Name: WWDCDemo

Location: Absolute

WWDCDemo.xcodeproj

Full Path: /Users/demo/Desktop/WWDCDemo/WWDCDemo.xcodeproj ⓘ

Project Document

Project Format: Xcode 9.3-compatible ⓘ

Organization: Demo

Class Prefix:

Text Settings

Indent Using: Spaces ⓘ

Widths: Tab: 4, Indent: 4

Wrap lines

WWDCDemo

- WWDCDemo
 - WWDCDemo.entitlements
 - AppDelegate.swift
 - MasterViewController.swift
 - DetailViewController.swift
 - Main.storyboard
 - Assets.xcassets
 - LaunchScreen.storyboard
 - Info.plist
 - WWDCDemo.xcdatamodeld
- WWDCDemoTests
 - WWDCDemoTests.swift
 - Info.plist
- WWDCDemoUITests
 - WWDCDemoUITests.swift
 - Info.plist
- Products
- Frameworks

WWDCDemo

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

PROJECT: WWDCDemo

TARGETS: WWDCDemo (selected), WWDCDemoTests, WWDCDemoUITests

Capabilities: + Capability All Debug Release

Signing

- Automatically manage signing
Xcode will create and update profiles, app IDs, and certificates.
- Team: Apple Inc. - Cocoa
- Bundle Identifier: com.apple.wwdc.WWDCDemo
- Provisioning Profile: Xcode Managed Profile ⓘ
- Signing Certificate: Apple Development: Nick Gillett (3U5Q9B57E4)

iCloud

Services

- Key-value storage
- iCloud Documents
- CloudKit

Containers

- Use default container
- Specify custom containers
 - iCloud.com.apple.wwdc.WWDCDemo iCloud.
 - com.apple.coredata.cdutil
 - com.apple.sidenotes

+ iCloud Dashboard

Push Notifications

Add capabilities by clicking the "+" button above.

Identity and Type

Name: WWDCDemo

Location: Absolute

WWDCDemo.xcodeproj

Full Path: /Users/demo/Desktop/WWDCDemo/WWDCDemo.xcodeproj

Project Document

Project Format: Xcode 9.3-compatible

Organization: Demo

Class Prefix:

Text Settings

Indent Using: Spaces

Widths: Tab: 4, Indent: 4

Wrap lines



FaceTime



Calendar



Photos



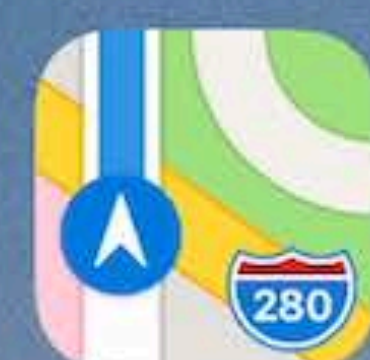
Camera



Contacts



Clock



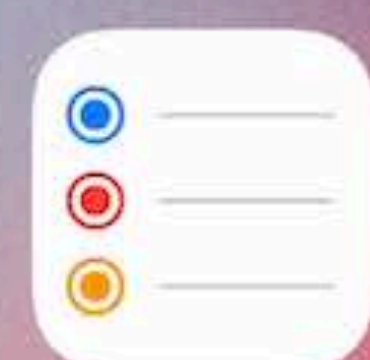
Maps



Home



Notes



Reminders



Stocks



News



TV



iTunes Store



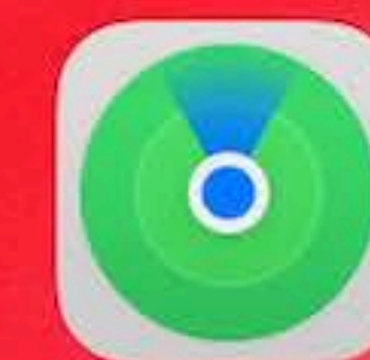
App Store



Books



Settings



Find My



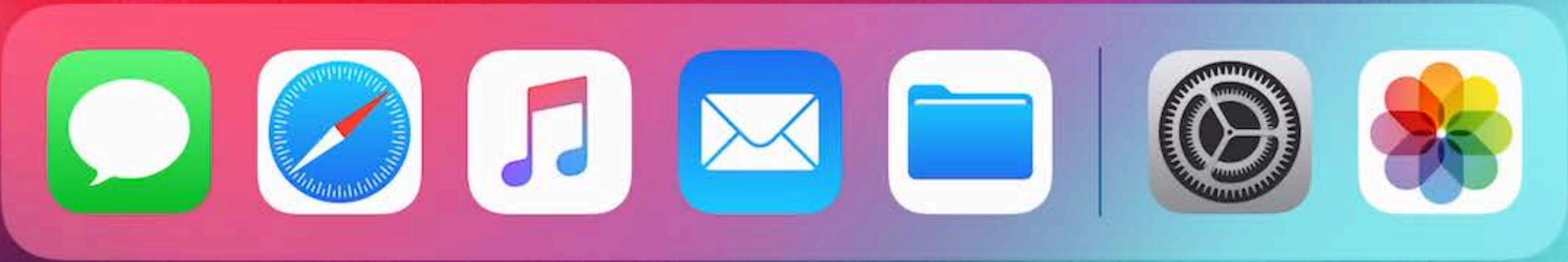
Podcasts



Photo Booth



Feedback





FaceTime



Calendar



Photos



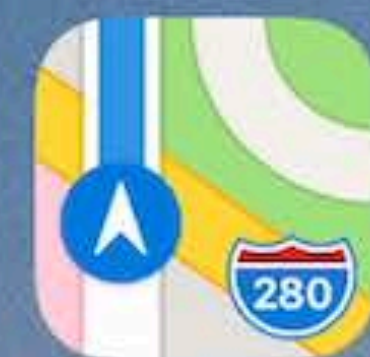
Camera



Contacts



Clock



Maps



Home



Notes



Reminders



Stocks



News



TV



iTunes Store



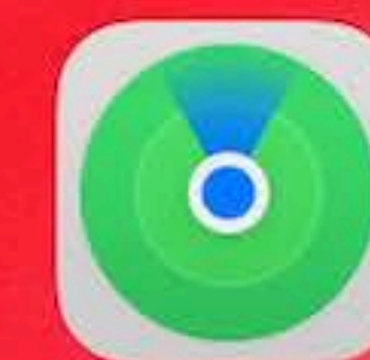
App Store



Books



Settings



Find My



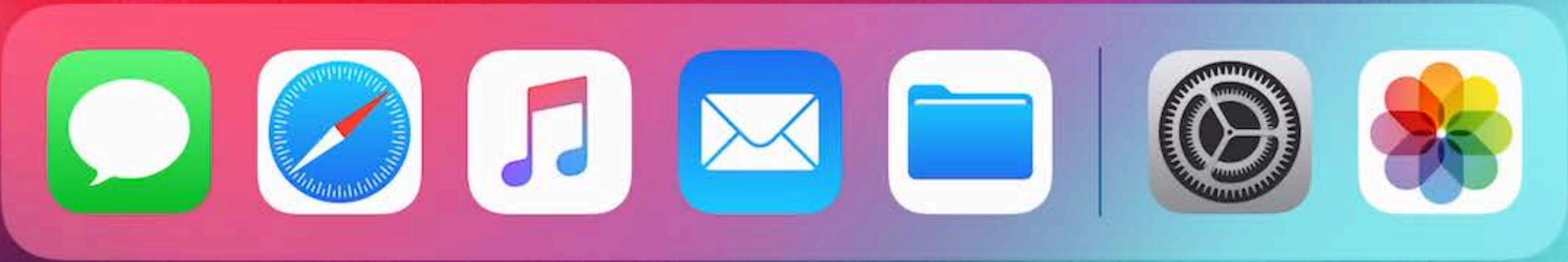
Podcasts



Photo Booth



Feedback





9:41



FaceTime



Calendar



Photos



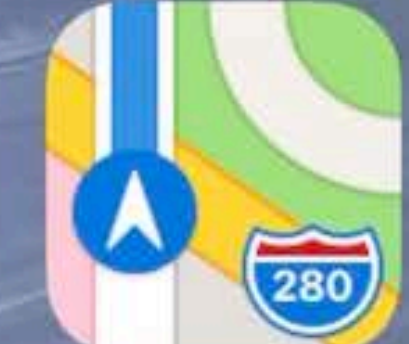
Camera



Mail



Clock



Maps



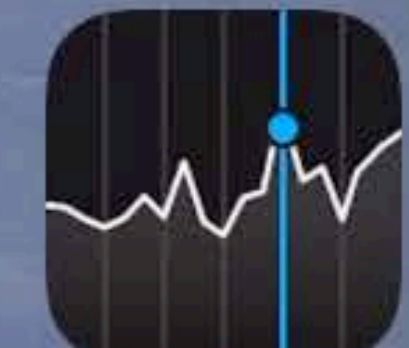
Weather



Reminders



Notes



Stocks



News



Books



App Store



Podcasts



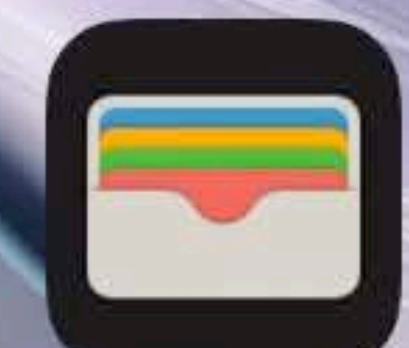
TV



Health



Home



Wallet



Settings





9:41



FaceTime



Calendar



Photos



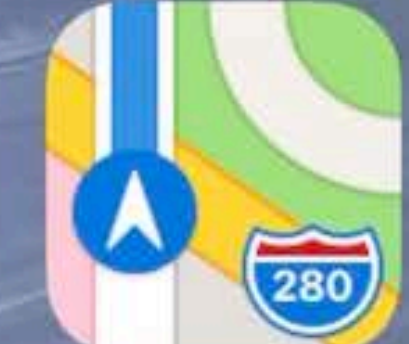
Camera



Mail



Clock



Maps



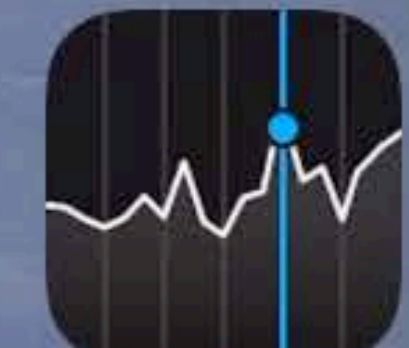
Weather



Reminders



Notes



Stocks



News



Books



App Store



Podcasts



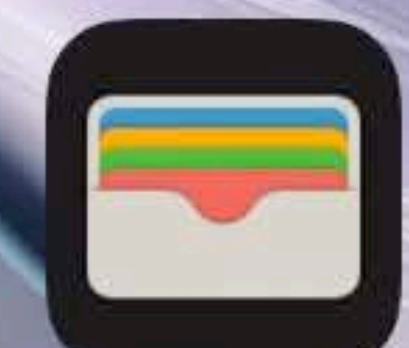
TV



Health



Home

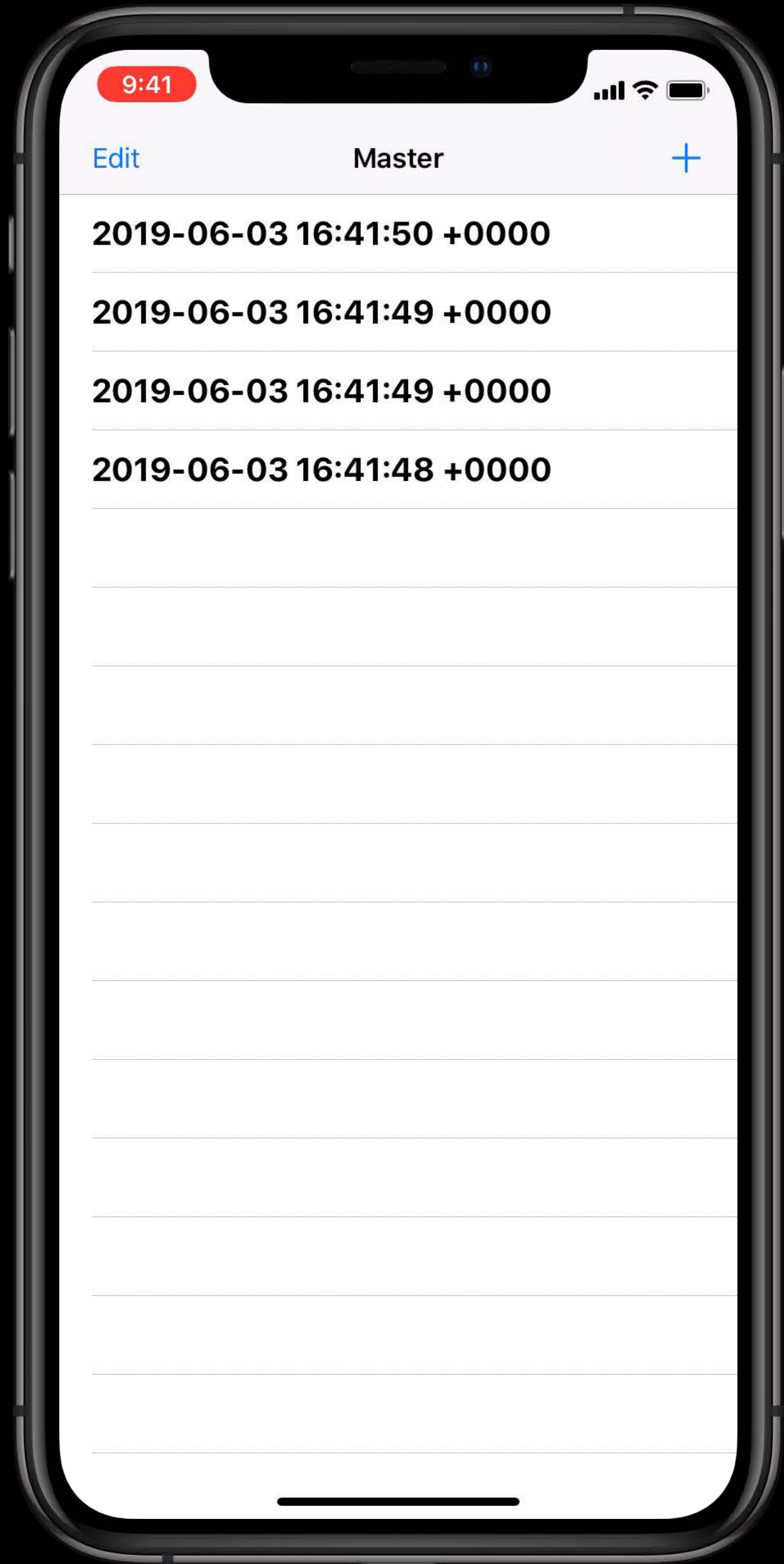


Wallet



Settings





9:41



Edit

Master

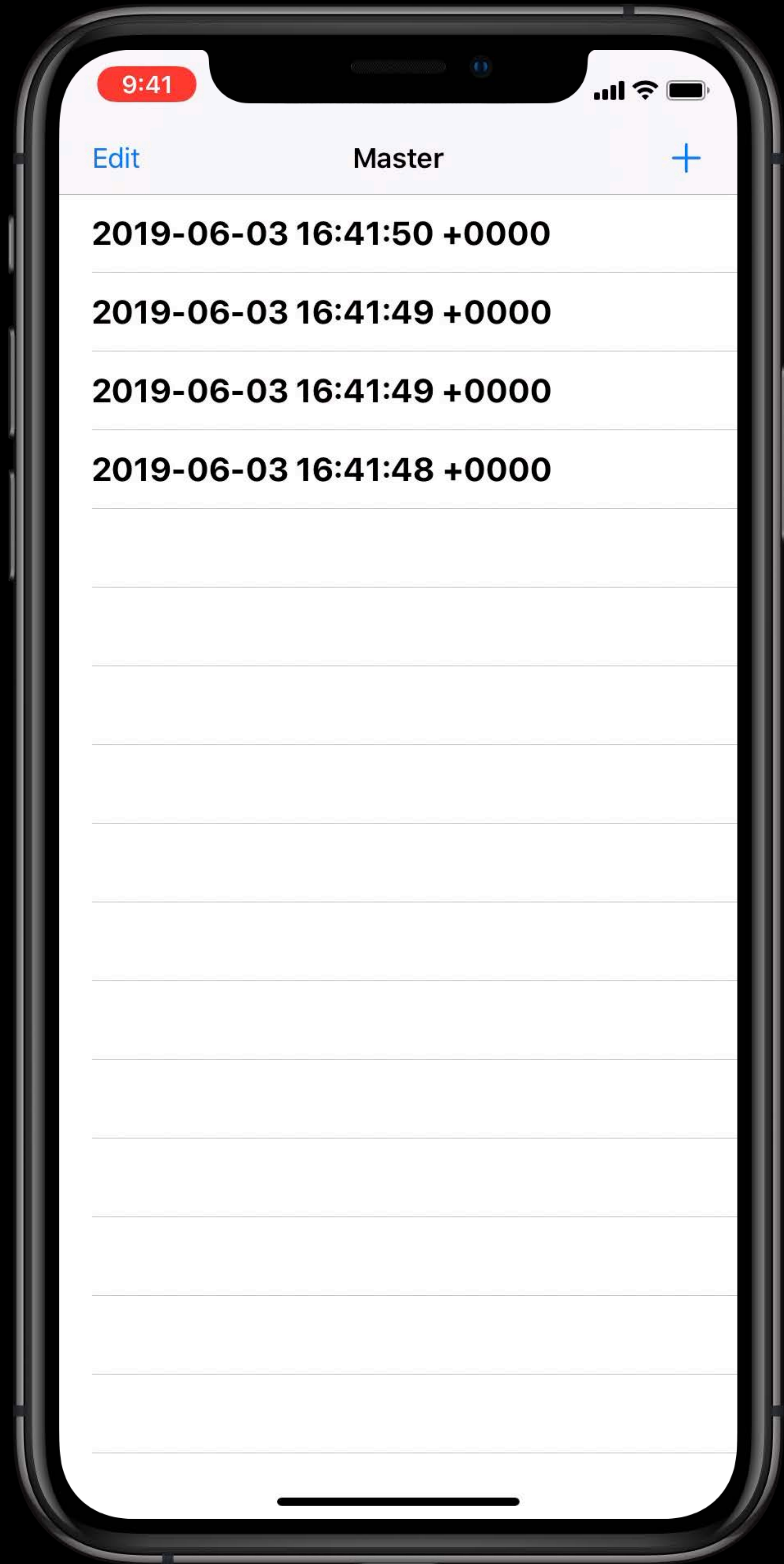


2019-06-03 16:41:50 +0000

2019-06-03 16:41:49 +0000

2019-06-03 16:41:49 +0000

2019-06-03 16:41:48 +0000



9:41



Edit

Master



2019-06-03 16:41:50 +0000

2019-06-03 16:41:49 +0000

2019-06-03 16:41:49 +0000

2019-06-03 16:41:48 +0000

Edit

Master



Detail

2019-06-03 16:41:50 +0000

2019-06-03 16:41:49 +0000

2019-06-03 16:41:49 +0000

2019-06-03 16:41:48 +0000

Detail view content goes here

Edit

Master



Detail

2019-06-03 16:41:50 +0000

2019-06-03 16:41:49 +0000

2019-06-03 16:41:49 +0000

2019-06-03 16:41:48 +0000

Detail view content goes here

Edit

Master



Detail

2019-06-03 16:42:34 +0000

2019-06-03 16:42:34 +0000

2019-06-03 16:42:33 +0000

2019-06-03 16:42:33 +0000

2019-06-03 16:41:50 +0000

2019-06-03 16:41:49 +0000

2019-06-03 16:41:49 +0000

2019-06-03 16:41:48 +0000

Detail view content goes here

Edit

Master



Detail

2019-06-03 16:42:34 +0000

2019-06-03 16:42:34 +0000

2019-06-03 16:42:33 +0000

2019-06-03 16:42:33 +0000

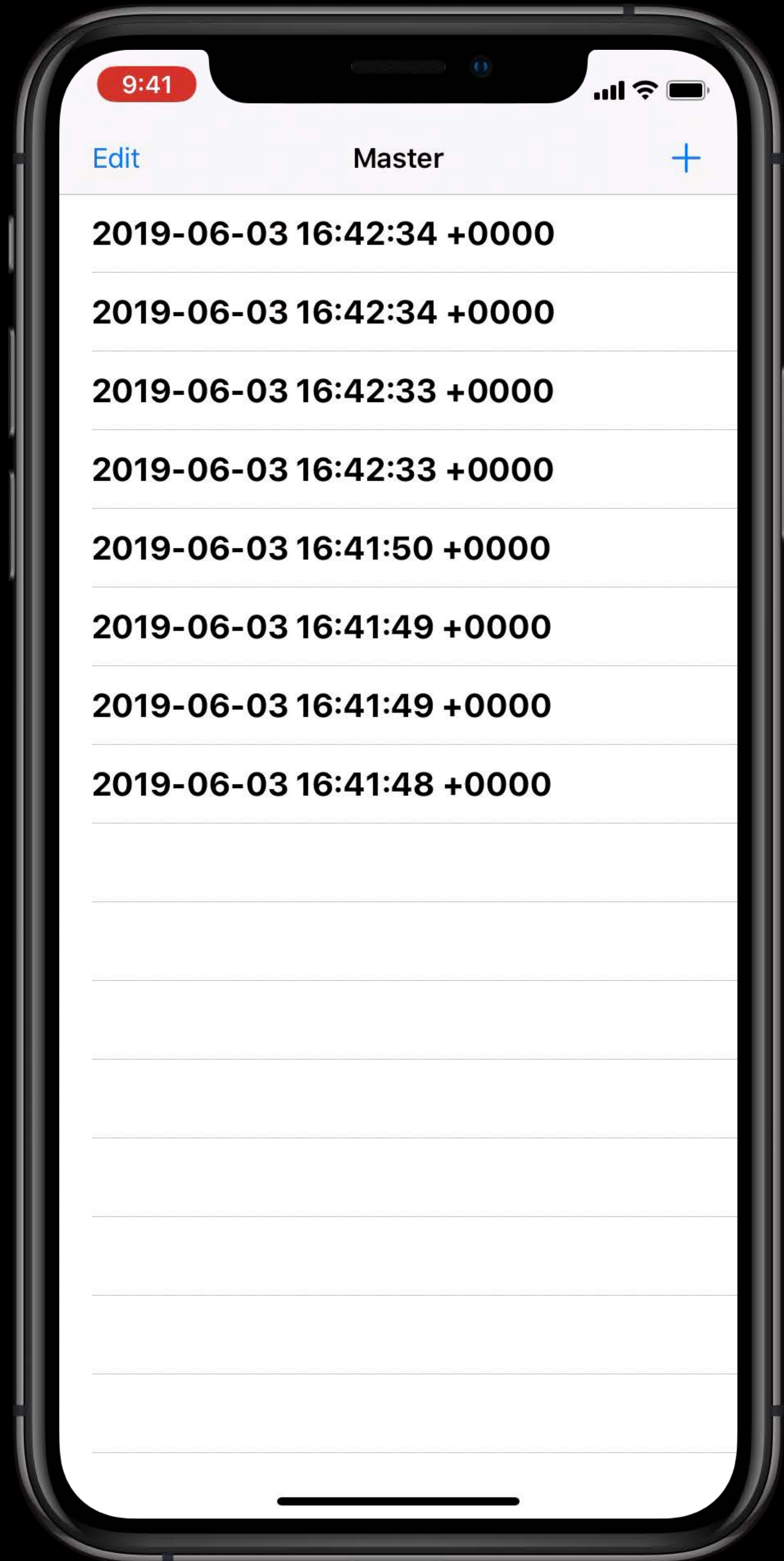
2019-06-03 16:41:50 +0000

2019-06-03 16:41:49 +0000

2019-06-03 16:41:49 +0000

2019-06-03 16:41:48 +0000

Detail view content goes here



9:41



Edit

Master



2019-06-03 16:42:34 +0000

2019-06-03 16:42:34 +0000

2019-06-03 16:42:33 +0000

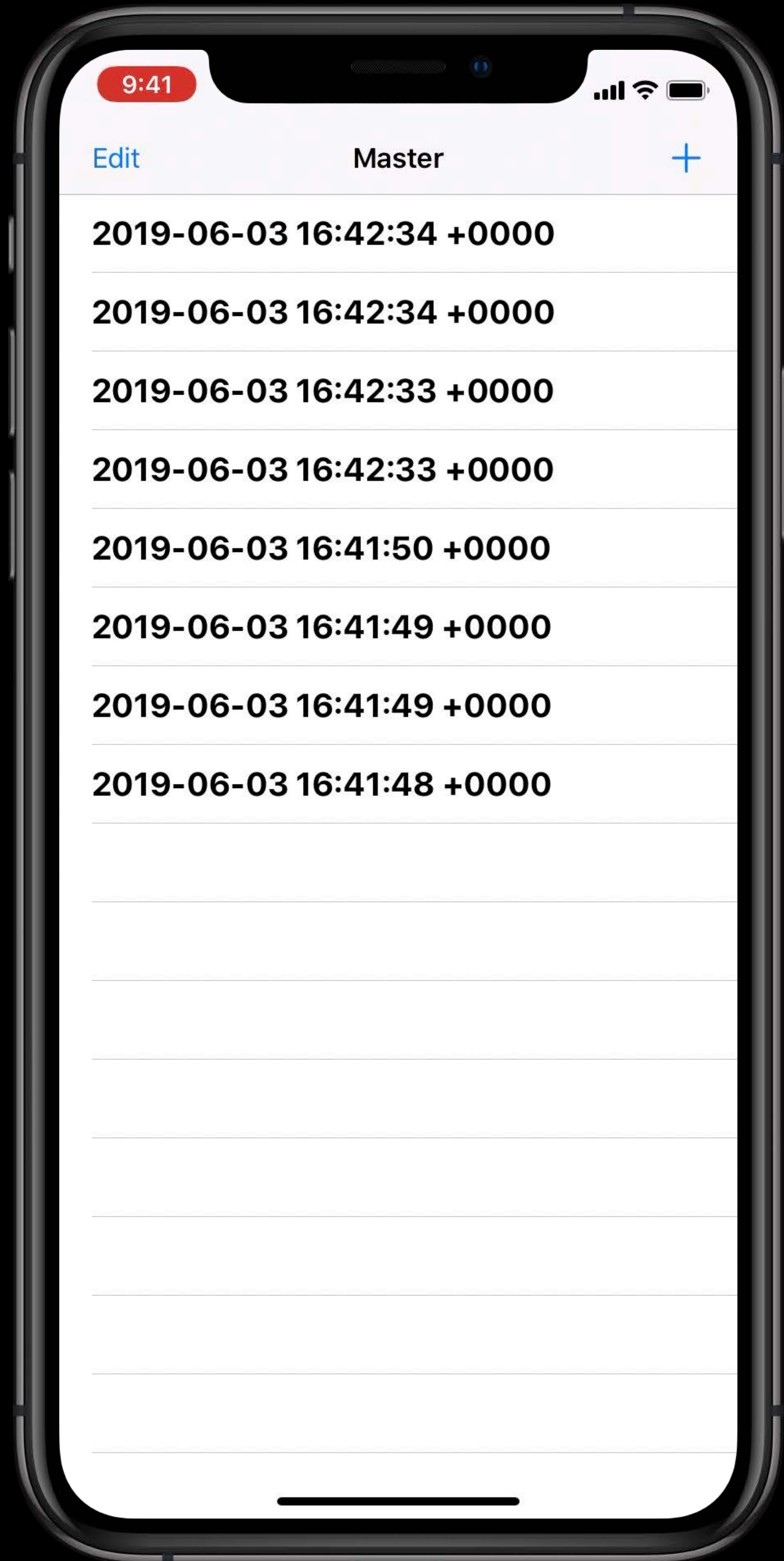
2019-06-03 16:42:33 +0000

2019-06-03 16:41:50 +0000

2019-06-03 16:41:49 +0000

2019-06-03 16:41:49 +0000

2019-06-03 16:41:48 +0000



9:41



Edit

Master



2019-06-03 16:42:34 +0000

2019-06-03 16:42:34 +0000

2019-06-03 16:42:33 +0000

2019-06-03 16:42:33 +0000

2019-06-03 16:41:50 +0000

2019-06-03 16:41:49 +0000

2019-06-03 16:41:49 +0000

2019-06-03 16:41:48 +0000

- WWDCDemo
 - WWDCDemo
 - WWDCDemo.entitlements
 - AppDelegate.swift
 - MasterViewController.swift
 - DetailViewController.swift
 - Main.storyboard
 - Assets.xcassets
 - LaunchScreen.storyboard
 - Info.plist
 - WWDCDemo.xcdatamodeld
 - WWDCDemoTests
 - WWDCDemoTests.swift
 - Info.plist
 - WWDCDemoUITests
 - WWDCDemoUITests.swift
 - Info.plist
 - Products
 - Frameworks

```

16
17
18 func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions:
    [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
19     // Override point for customization after application launch.
20     let splitViewController = self.window!.rootViewController as! UISplitViewController
21     let navigationController =
        splitViewController.viewControllers[splitViewController.viewControllers.count-1] as!
        UINavigationController
22     navigationController.topViewController!.navigationItem.leftBarButtonItem =
        splitViewController.displayModeButtonItem
23     splitViewController.delegate = self
24
25     let masterNavigationController = splitViewController.viewControllers[0] as!
        UINavigationController
26     let controller = masterNavigationController.topViewController as! MasterViewController
27     controller.managedObjectContext = self.persistentContainer.viewContext
28     self.persistentContainer.viewContext.automaticallyMergesChangesFromParent = true
29     return true
30 }
31
32 func applicationWillResignActive(_ application: UIApplication) {
33     // Sent when the application is about to move from active to inactive state. This can occur
        for certain types of temporary interruptions (such as an incoming phone call or SMS
        message) or when the user quits the application and it begins the transition to the
        background state.
34     // Use this method to pause ongoing tasks, disable timers, and invalidate graphics
        rendering callbacks. Games should use this method to pause the game.
35 }
36
37 func applicationDidEnterBackground(_ application: UIApplication) {
38     // Use this method to release shared resources, save user data, invalidate timers, and
        store enough application state information to restore your application to its current
    
```

Split view

Core Data stack

Core Data Saving support

- WWDCDemo
 - WWDCDemo
 - WWDCDemo.entitlements
 - AppDelegate.swift
 - MasterViewController.swift
 - DetailViewController.swift
 - Main.storyboard
 - Assets.xcassets
 - LaunchScreen.storyboard
 - Info.plist
 - WWDCDemo.xcdatamodeld
 - WWDCDemoTests
 - WWDCDemoTests.swift
 - Info.plist
 - WWDCDemoUITests
 - WWDCDemoUITests.swift
 - Info.plist
 - Products
 - Frameworks

```

16
17
18 func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions:
    [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
19     // Override point for customization after application launch.
20     let splitViewController = self.window!.rootViewController as! UISplitViewController
21     let navigationController =
        splitViewController.viewControllers[splitViewController.viewControllers.count-1] as!
        UINavigationController
22     navigationController.topViewController!.navigationItem.leftBarButtonItem =
        splitViewController.displayModeButtonItem
23     splitViewController.delegate = self
24
25     let masterNavigationController = splitViewController.viewControllers[0] as!
        UINavigationController
26     let controller = masterNavigationController.topViewController as! MasterViewController
27     controller.managedObjectContext = self.persistentContainer.viewContext
28     self.persistentContainer.viewContext.automaticallyMergesChangesFromParent = true
29     return true
30 }
31
32 func applicationWillResignActive(_ application: UIApplication) {
33     // Sent when the application is about to move from active to inactive state. This can occur
        for certain types of temporary interruptions (such as an incoming phone call or SMS
        message) or when the user quits the application and it begins the transition to the
        background state.
34     // Use this method to pause ongoing tasks, disable timers, and invalidate graphics
        rendering callbacks. Games should use this method to pause the game.
35 }
36
37 func applicationDidEnterBackground(_ application: UIApplication) {
38     // Use this method to release shared resources, save user data, invalidate timers, and
        store enough application state information to restore your application to its current
    
```

Split view

Core Data stack

Core Data Saving support

- WWDCDemo
 - WWDCDemo
 - WWDCDemo.entitlements
 - AppDelegate.swift
 - MasterViewController.swift
 - DetailViewController.swift
 - Main.storyboard
 - Assets.xcassets
 - LaunchScreen.storyboard
 - Info.plist
 - WWDCDemo.xcdatamodeld
 - WWDCDemoTests
 - WWDCDemoTests.swift
 - Info.plist
 - WWDCDemoUITests
 - WWDCDemoUITests.swift
 - Info.plist
 - Products
 - Frameworks

```

64     }
65     return false
66 }
67 // MARK: - Core Data stack
68
69 lazy var persistentContainer: NSPersistentContainer = {
70     /*
71     The persistent container for the application. This implementation
72     creates and returns a container, having loaded the store for the
73     application to it. This property is optional since there are legitimate
74     error conditions that could cause the creation of the store to fail.
75     */
76     let container = NSPersistentContainer(name: "WWDCDemo")
77     container.loadPersistentStores(completionHandler: { (storeDescription, error) in
78         if let error = error as NSError? {
79             // Replace this implementation with code to handle the error appropriately.
80             // fatalError() causes the application to generate a crash log and terminate. You
81             // should not use this function in a shipping application, although it may be
82             // useful during development.
83
84             /*
85             Typical reasons for an error here include:
86             * The parent directory does not exist, cannot be created, or disallows writing.
87             * The persistent store is not accessible, due to permissions or data protection
88               when the device is locked.
89             * The device is out of space.
90             * The store could not be migrated to the current model version.
91             Check the error message to determine what the actual problem was.
92             */
93             fatalError("Unresolved error \(error), \(error.userInfo)")
94         }
95     })
96 }

```

Split view

Core Data stack

Core Data Saving support

- WWDCDemo
 - WWDCDemo
 - WWDCDemo.entitlements
 - AppDelegate.swift
 - MasterViewController.swift
 - DetailViewController.swift
 - Main.storyboard
 - Assets.xcassets
 - LaunchScreen.storyboard
 - Info.plist
 - WWDCDemo.xcdatamodeld
 - WWDCDemoTests
 - WWDCDemoTests.swift
 - Info.plist
 - WWDCDemoUITests
 - WWDCDemoUITests.swift
 - Info.plist
 - Products
 - Frameworks

```

64     }
65     return false
66 }
67 // MARK: - Core Data stack
68
69 lazy var persistentContainer: NSPersistentContainer = {
70     /*
71     The persistent container for the application. This implementation
72     creates and returns a container, having loaded the store for the
73     application to it. This property is optional since there are legitimate
74     error conditions that could cause the creation of the store to fail.
75     */
76     let container = NSPersistentContainer(name: "WWDCDemo")
77     container.loadPersistentStores(completionHandler: { (storeDescription, error) in
78         if let error = error as NSError? {
79             // Replace this implementation with code to handle the error appropriately.
80             // fatalError() causes the application to generate a crash log and terminate. You
81             // should not use this function in a shipping application, although it may be
82             // useful during development.
83
84             /*
85             Typical reasons for an error here include:
86             * The parent directory does not exist, cannot be created, or disallows writing.
87             * The persistent store is not accessible, due to permissions or data protection
88               when the device is locked.
89             * The device is out of space.
90             * The store could not be migrated to the current model version.
91             Check the error message to determine what the actual problem was.
92             */
93             fatalError("Unresolved error \(error), \(error.userInfo)")
94         }
95     })
96 }

```

Split view

Core Data stack

Core Data Saving support

What Is NSPersistentCloudKitContainer?

What Is NSPersistentCloudKitContainer?

Encapsulation of common patterns

What Is NSPersistentCloudKitContainer?

Encapsulation of common patterns

Save thousands of lines of code

What Is NSPersistentCloudKitContainer?

Encapsulation of common patterns

Save thousands of lines of code

Foundation we can build on

What Is NSPersistentCloudKitContainer?

Encapsulation of common patterns

Save thousands of lines of code

Foundation we can build on

Help us help you! Submit feedback!

NSPersistentCloudKitContainer

NSPersistentCloudKitContainer

A local replica of all CloudKit data

NSPersistentCloudKitContainer

A local replica of all CloudKit data

Robust scheduling and error recovery

NSPersistentCloudKitContainer

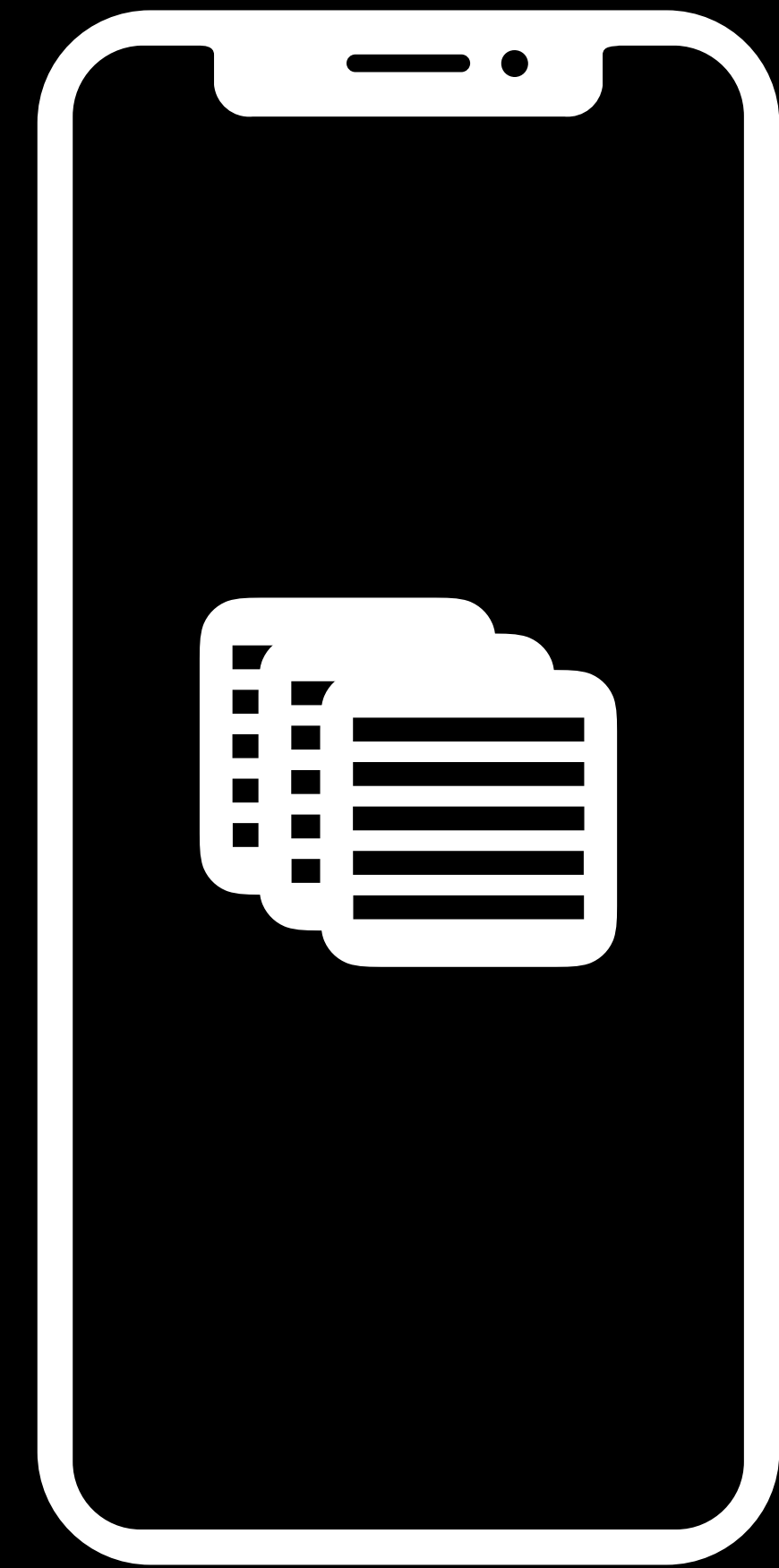
A local replica of all CloudKit data

Robust scheduling and error recovery

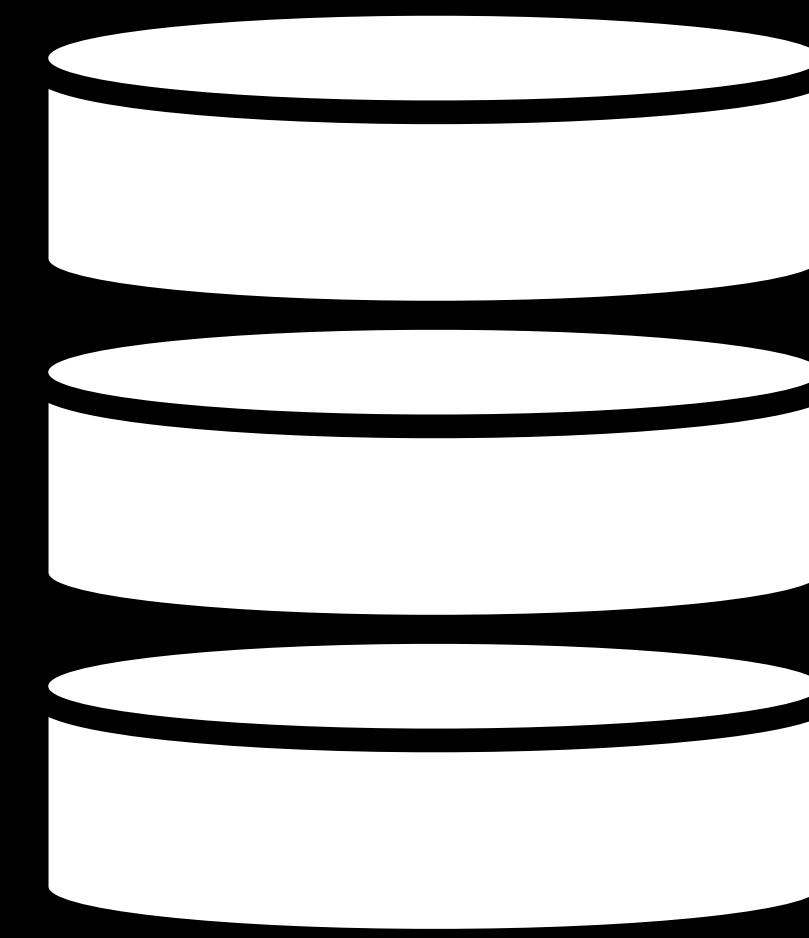
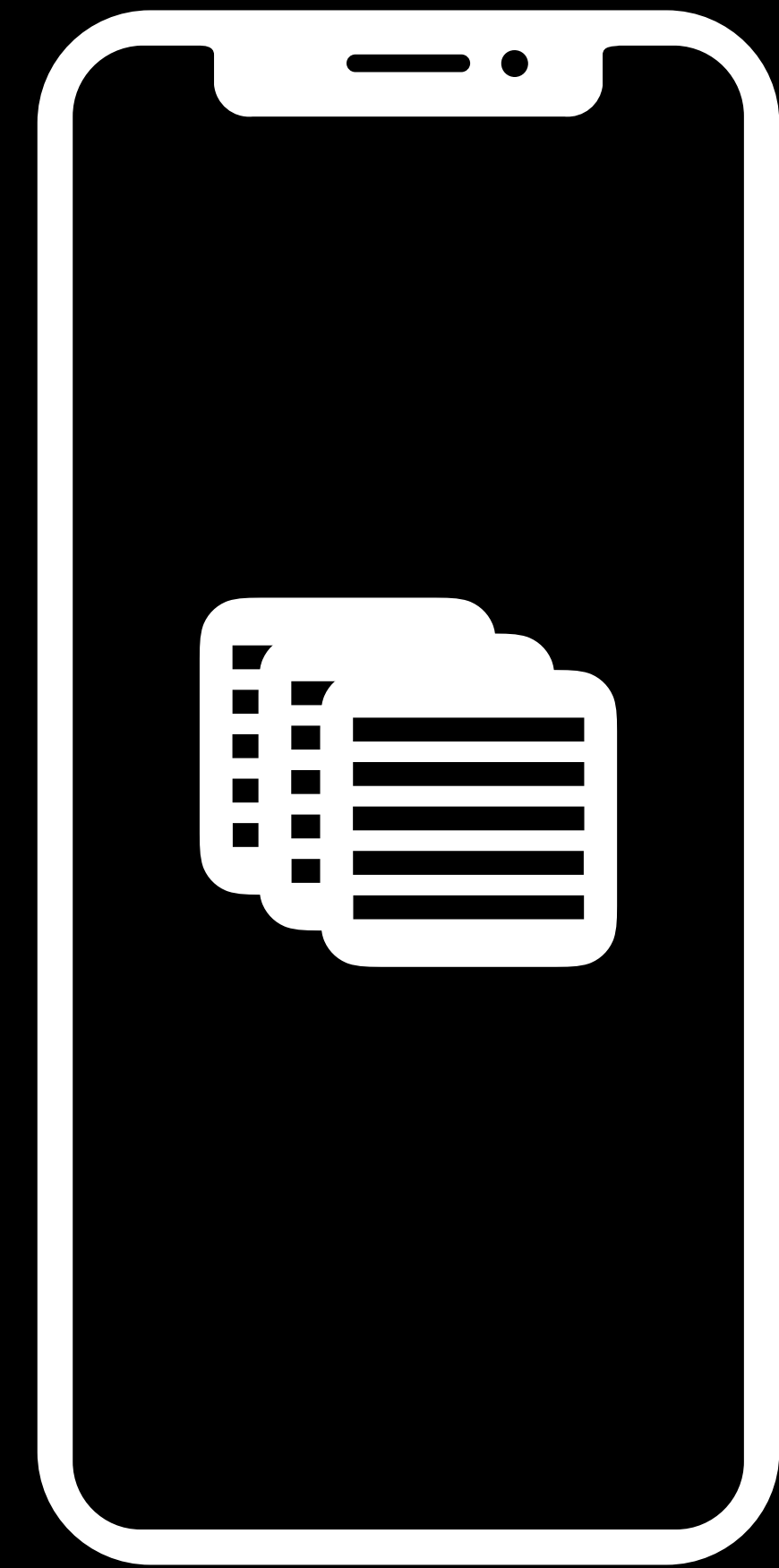
Transformation of NSManagedObject to CKRecord

Local Replica

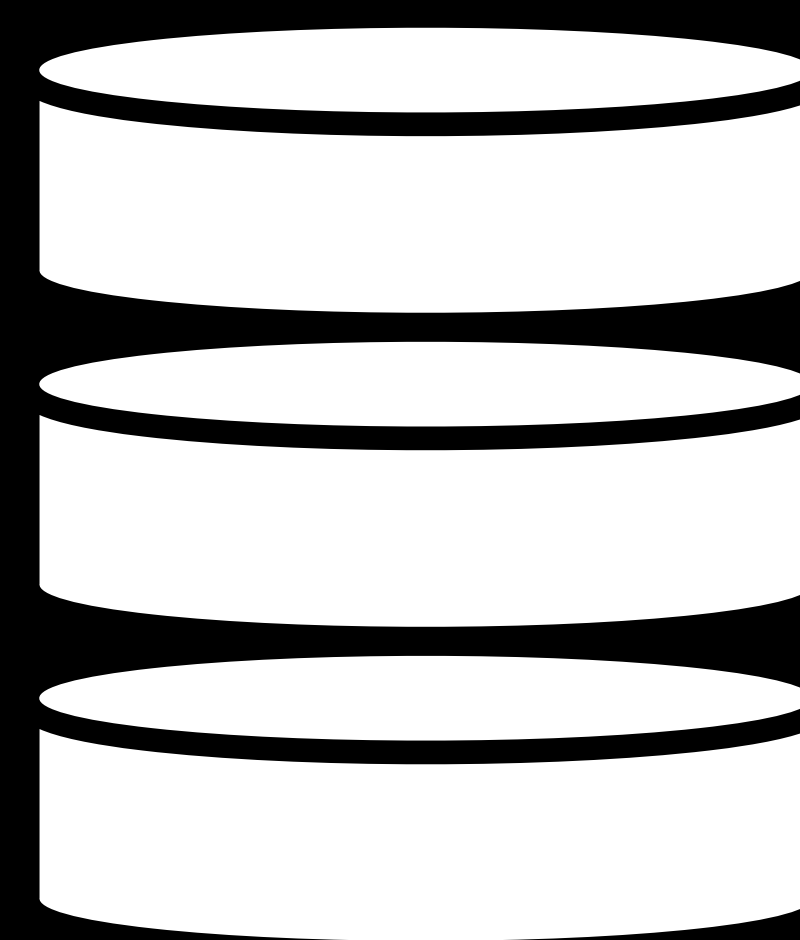
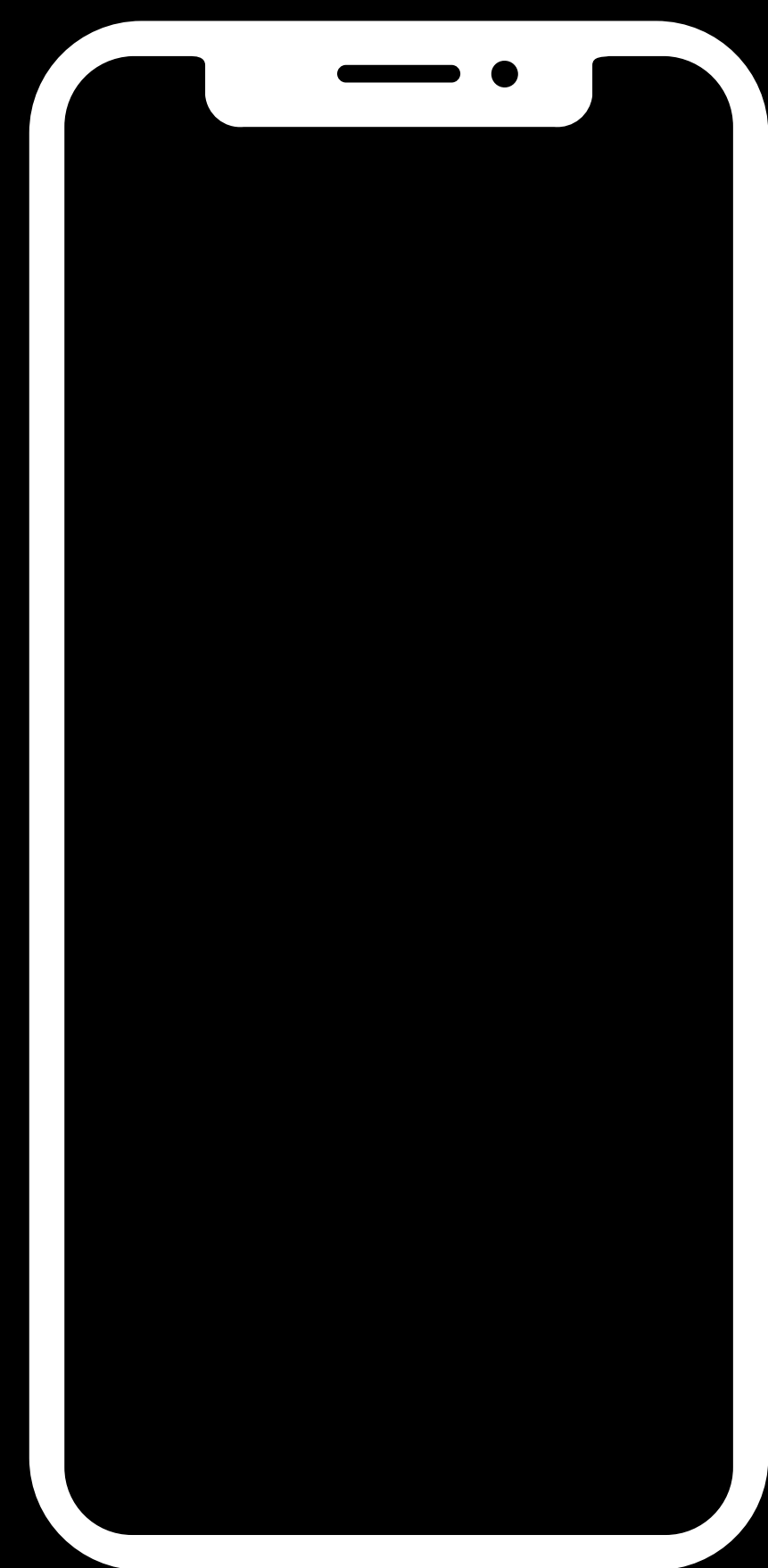
Local Replica



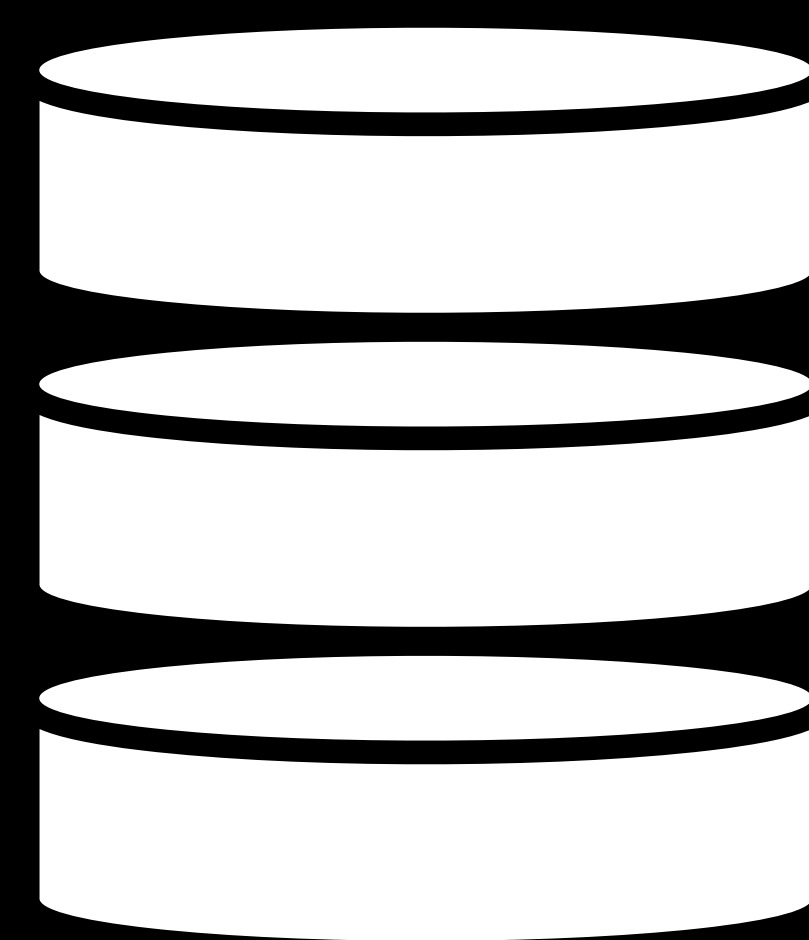
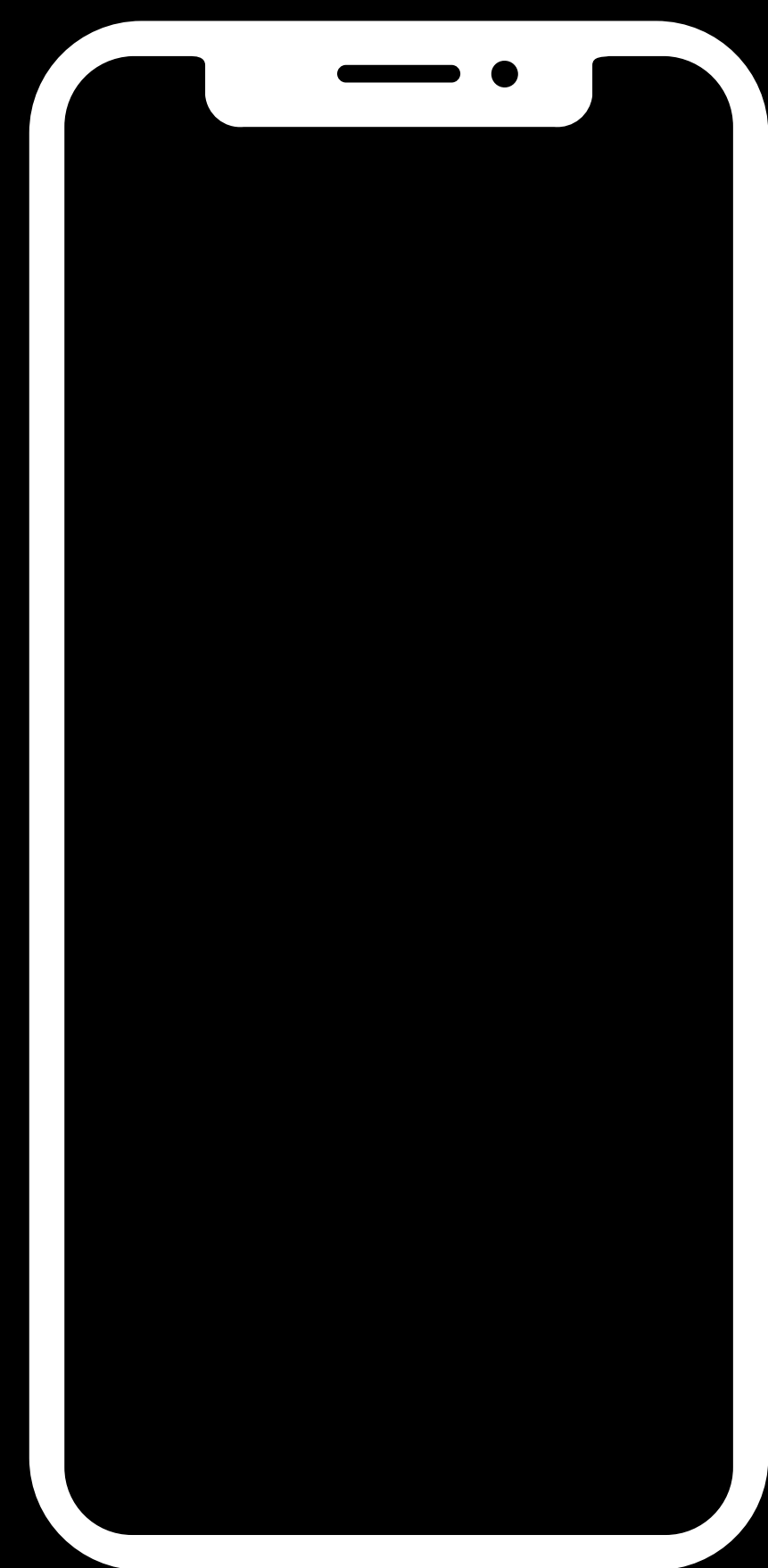
Local Replica



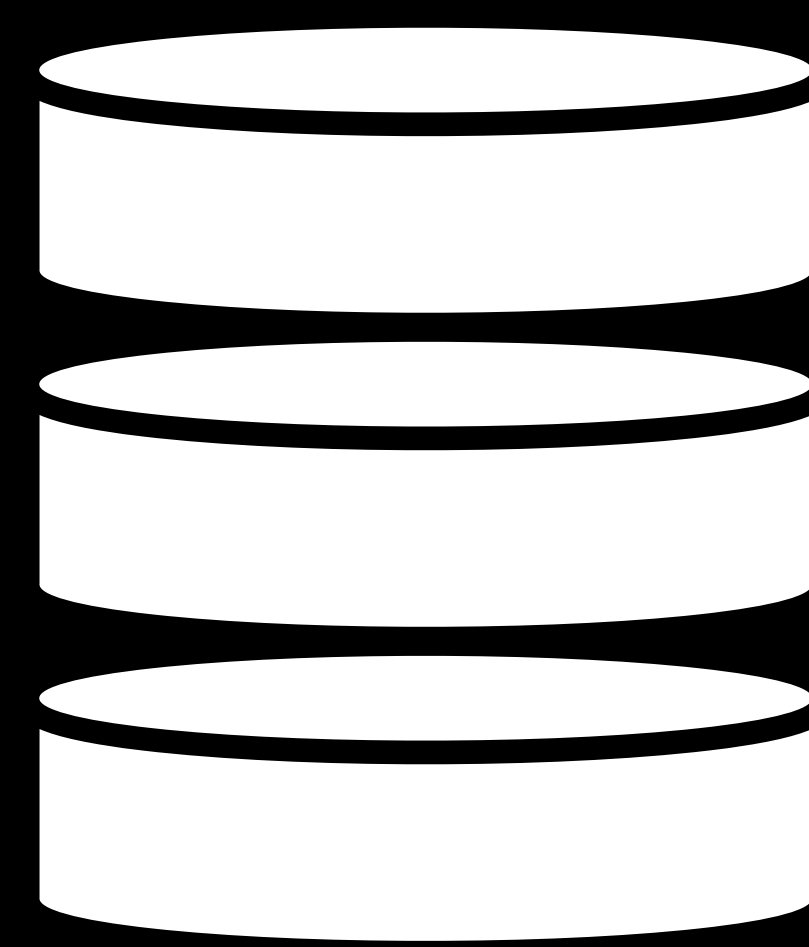
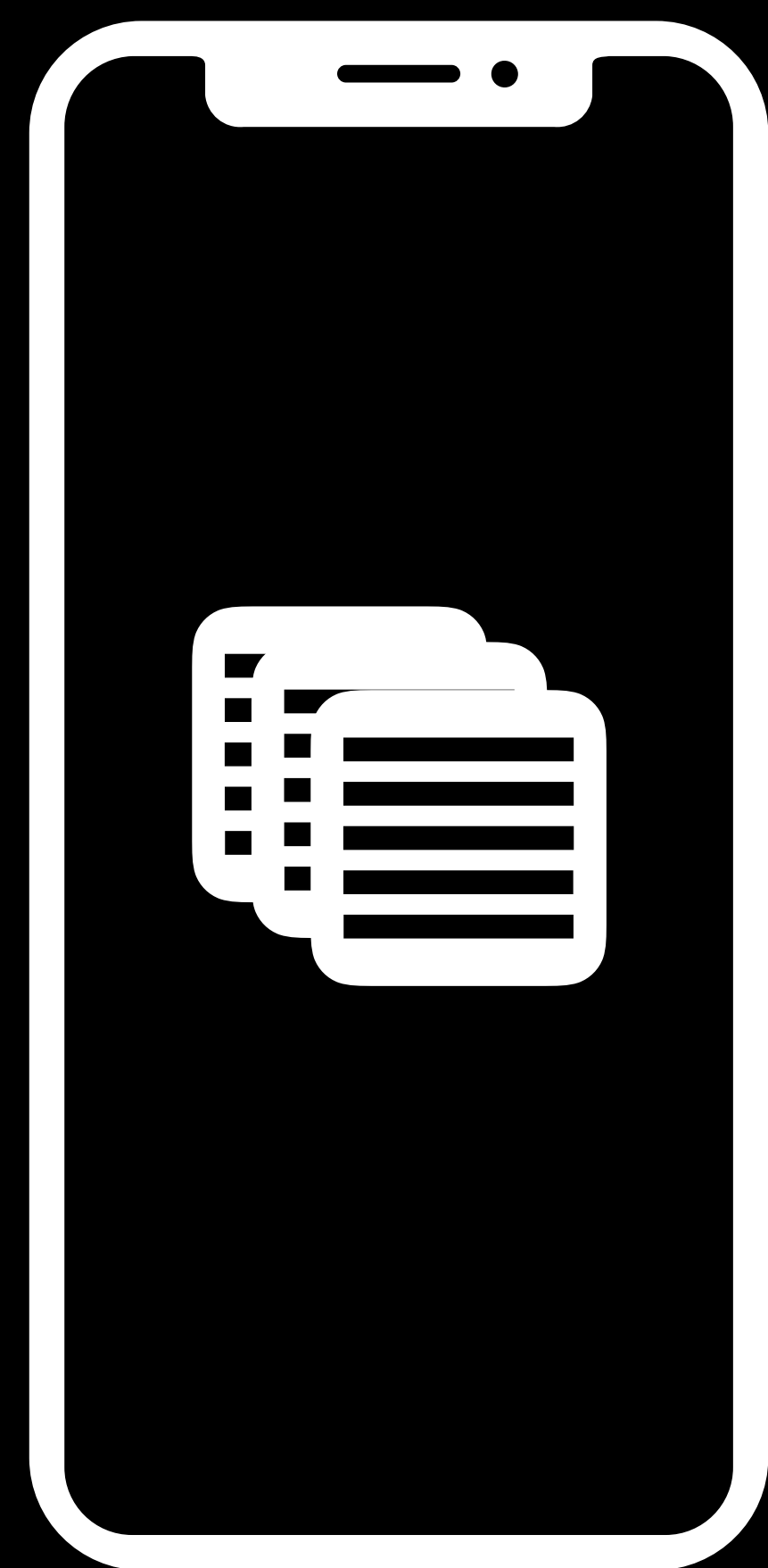
Local Replica



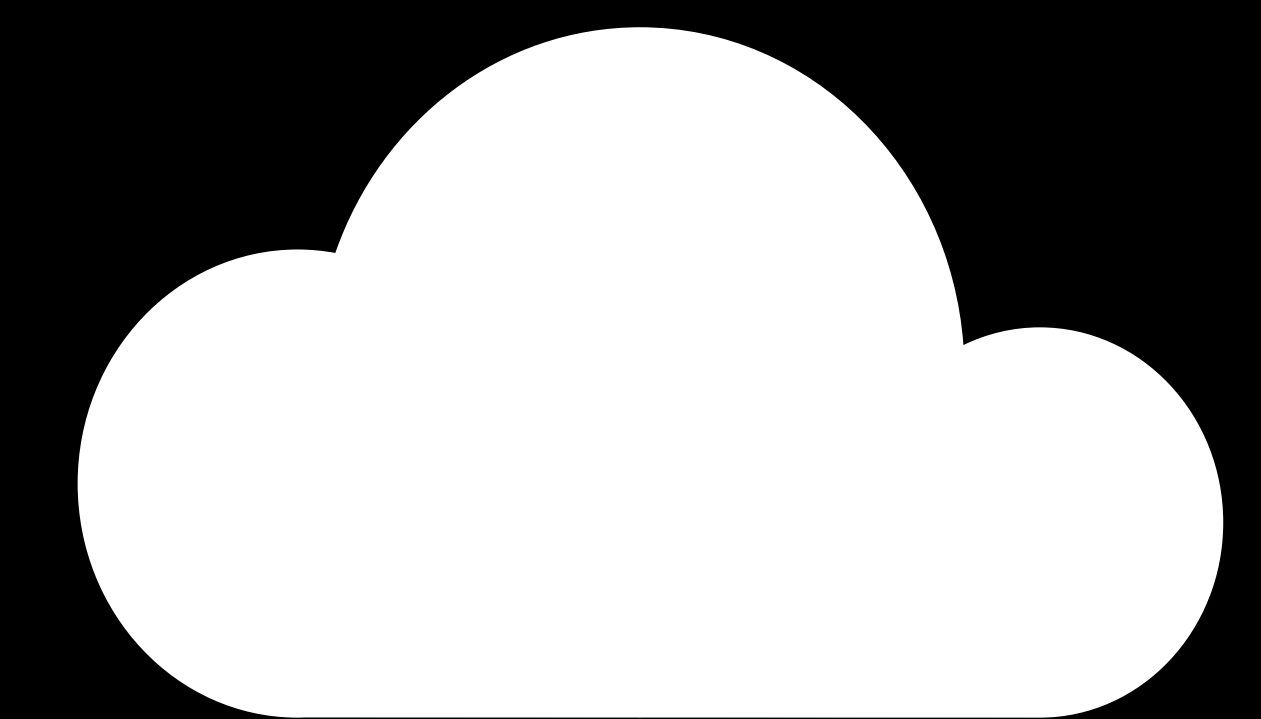
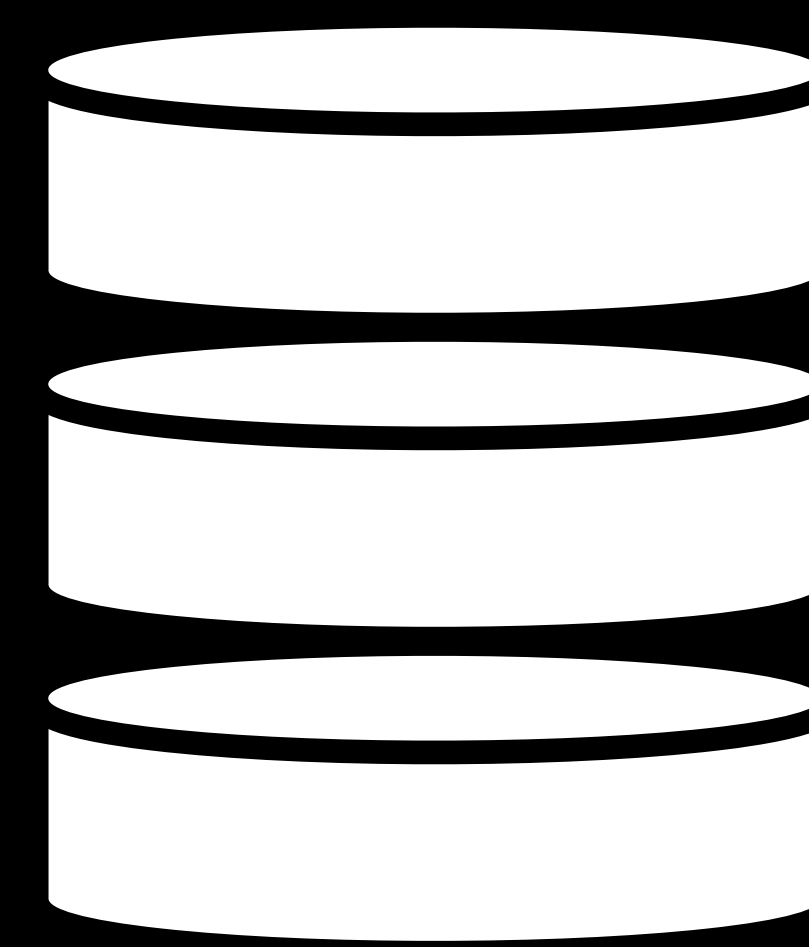
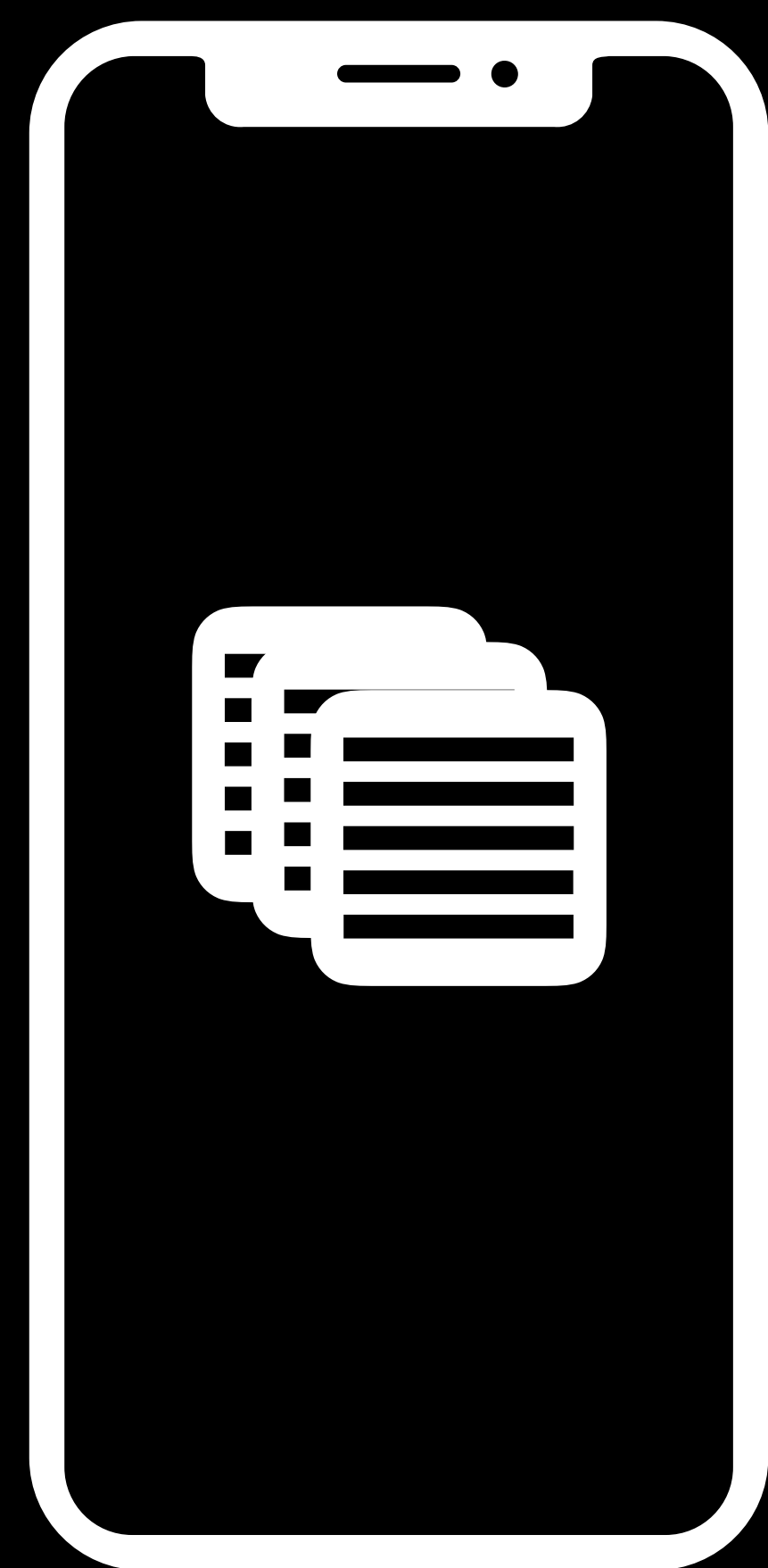
Local Replica



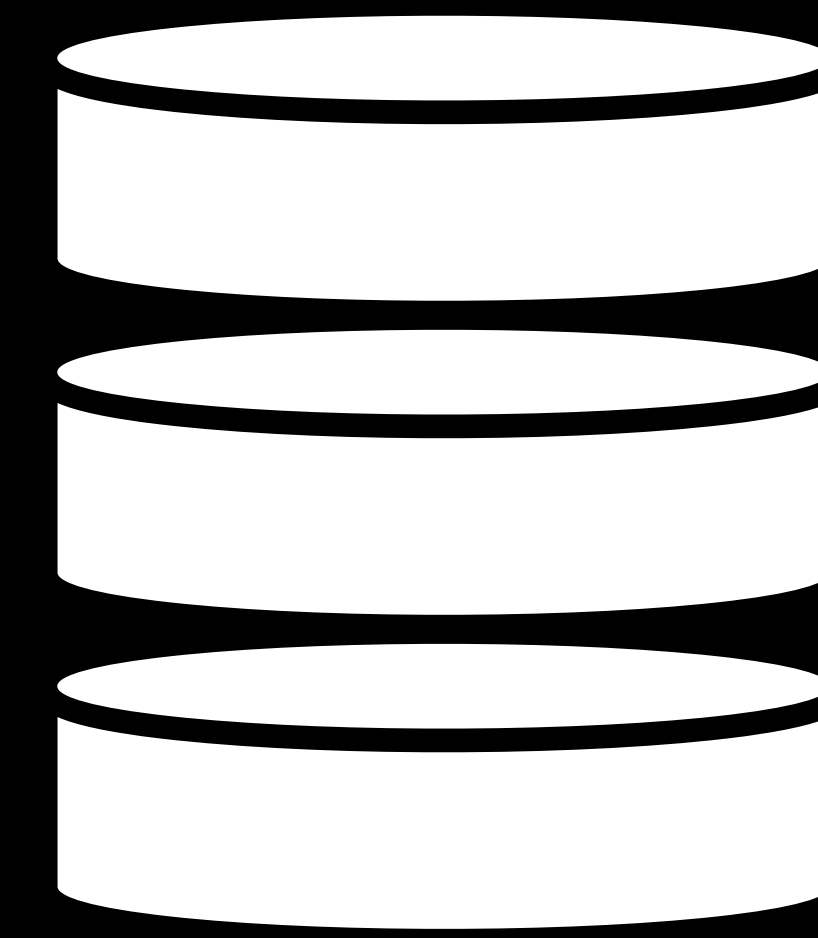
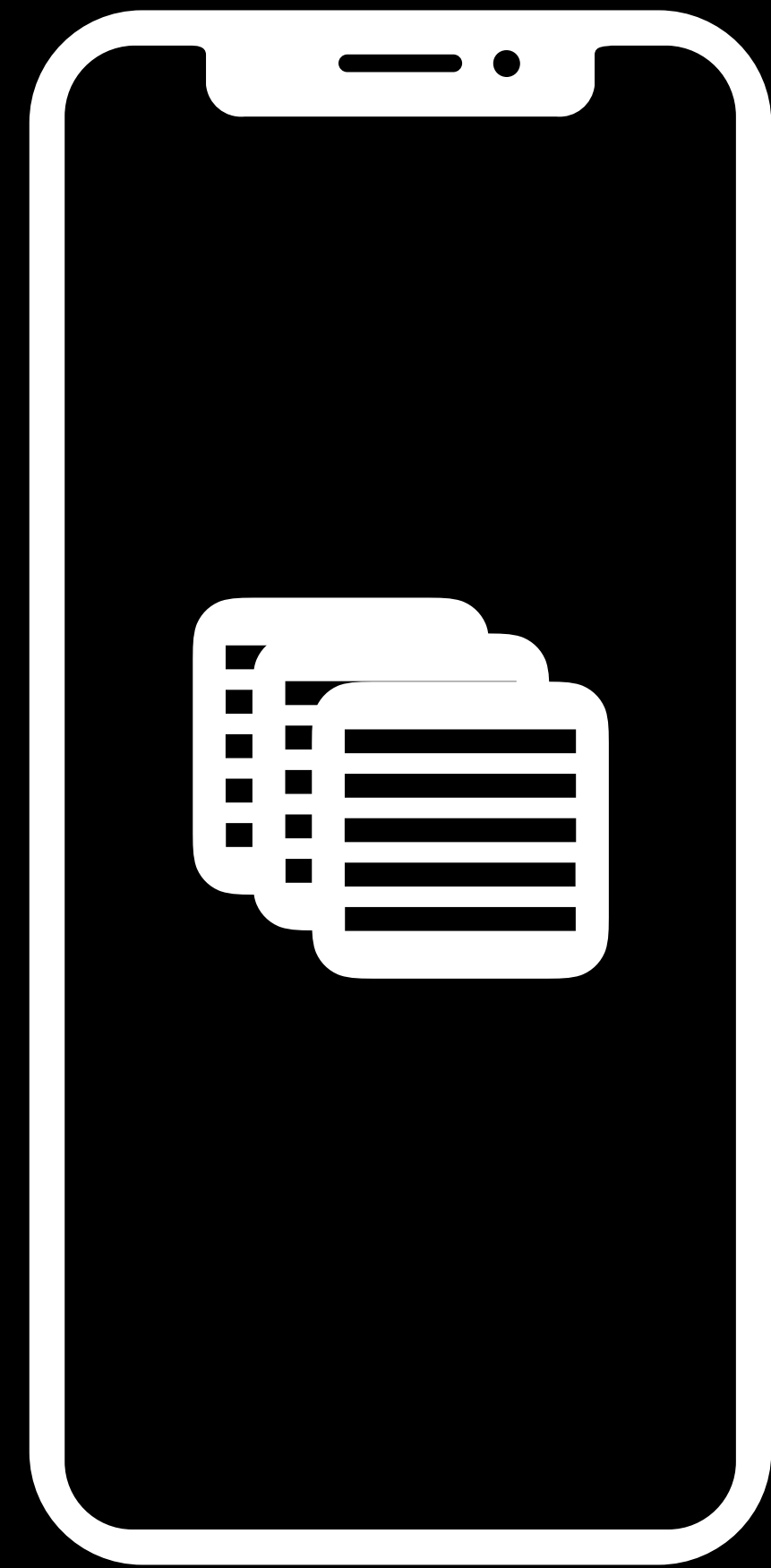
Local Replica



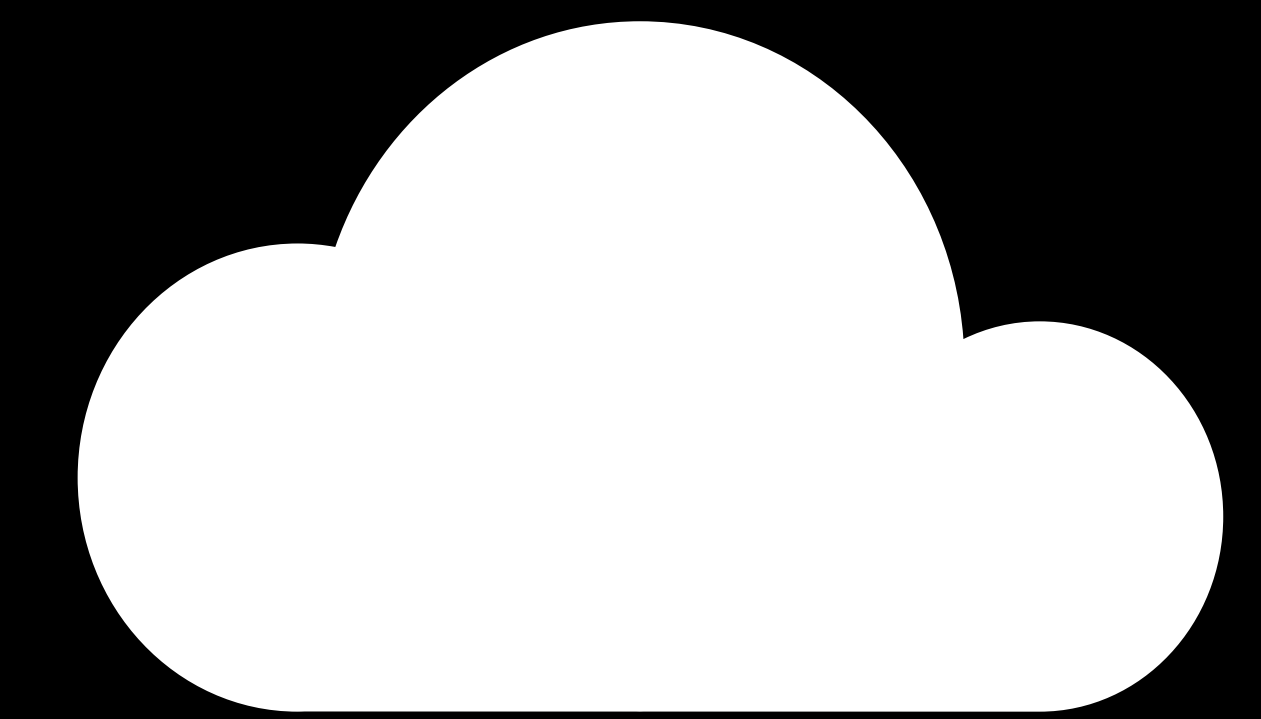
Local Replica



Local Replica

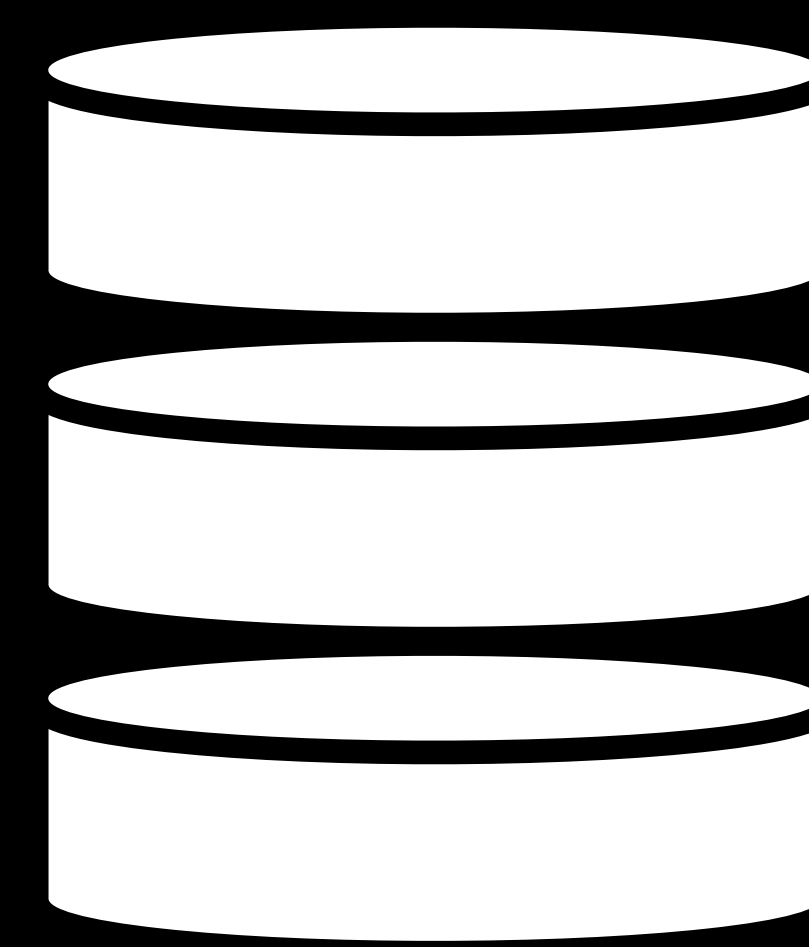
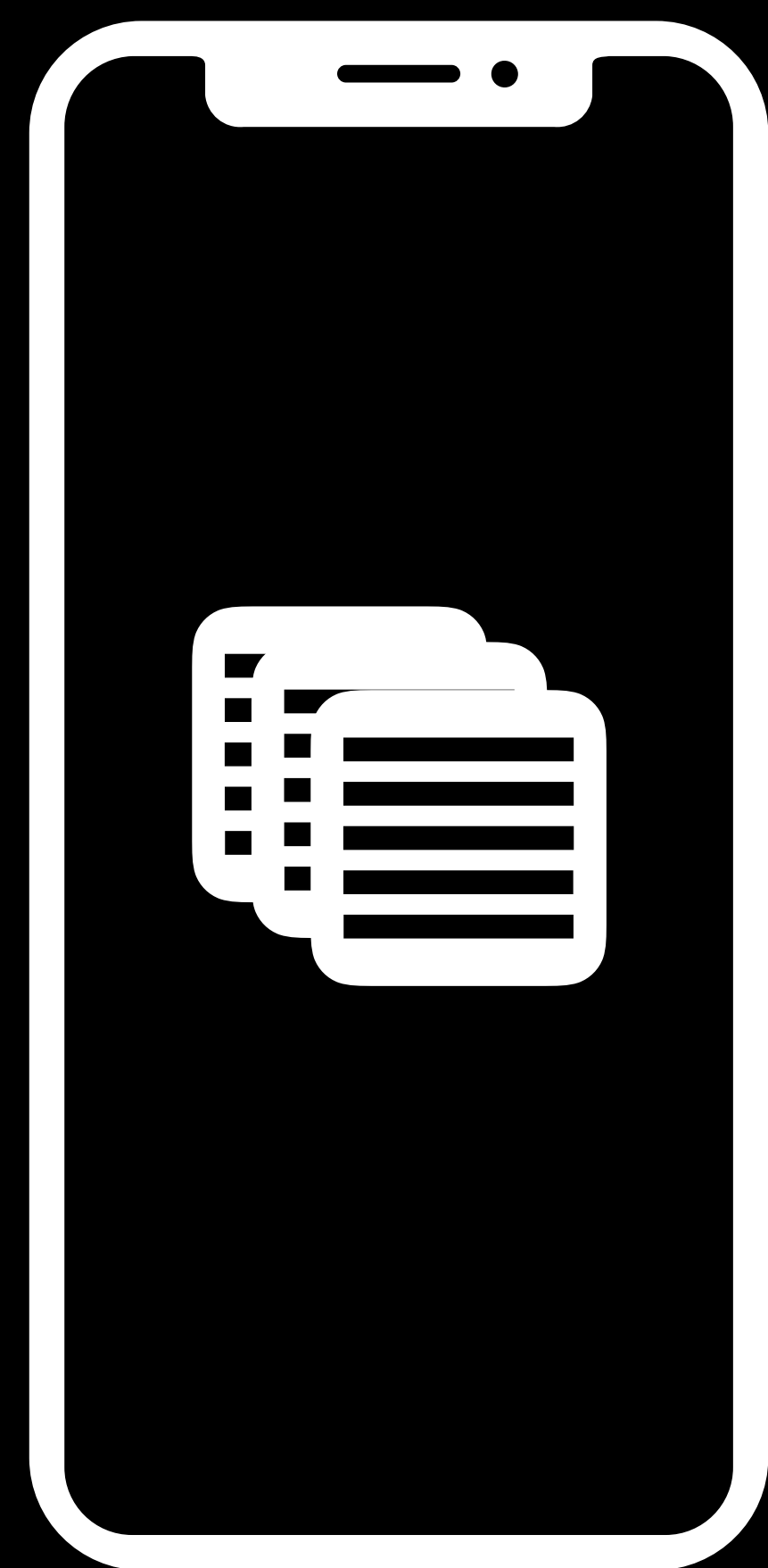


milliseconds

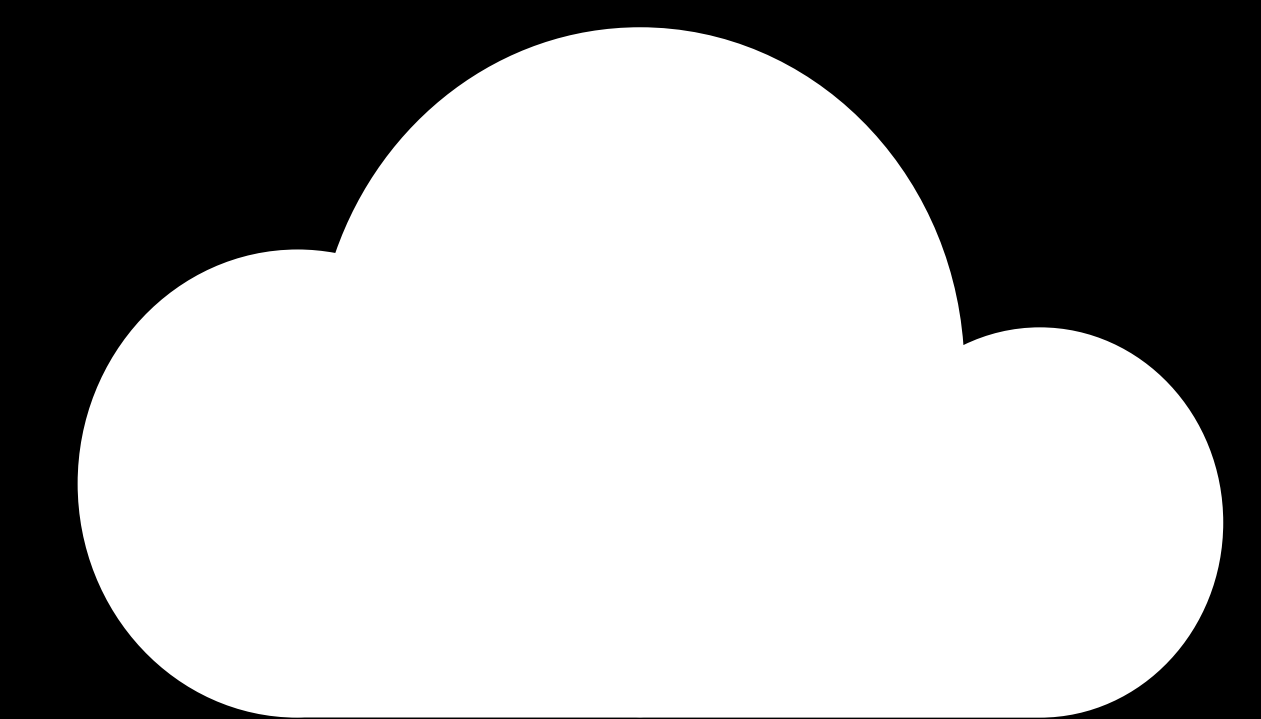


seconds – minutes

Local Replica

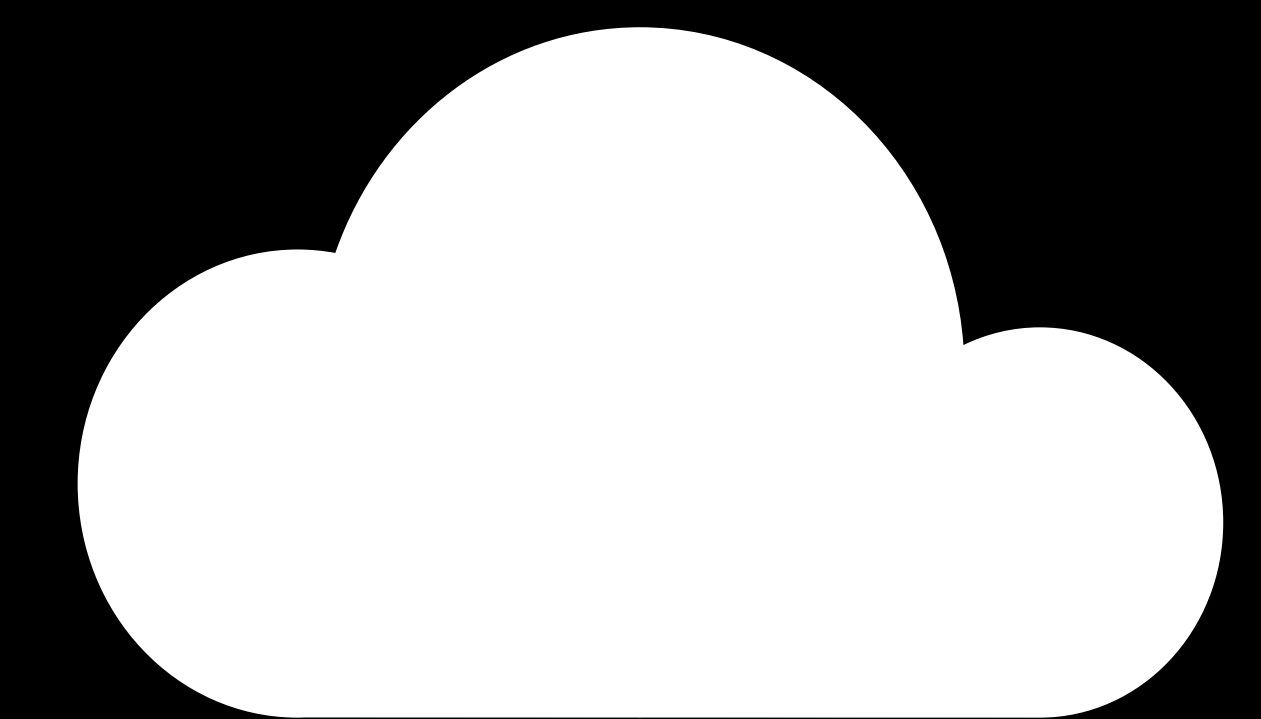
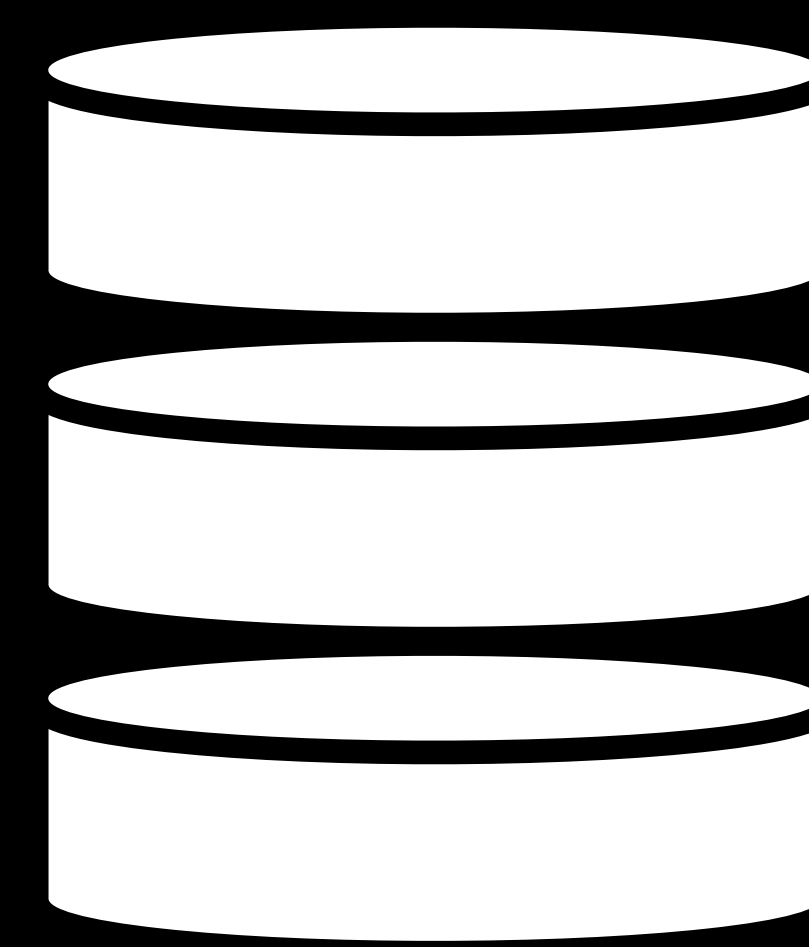
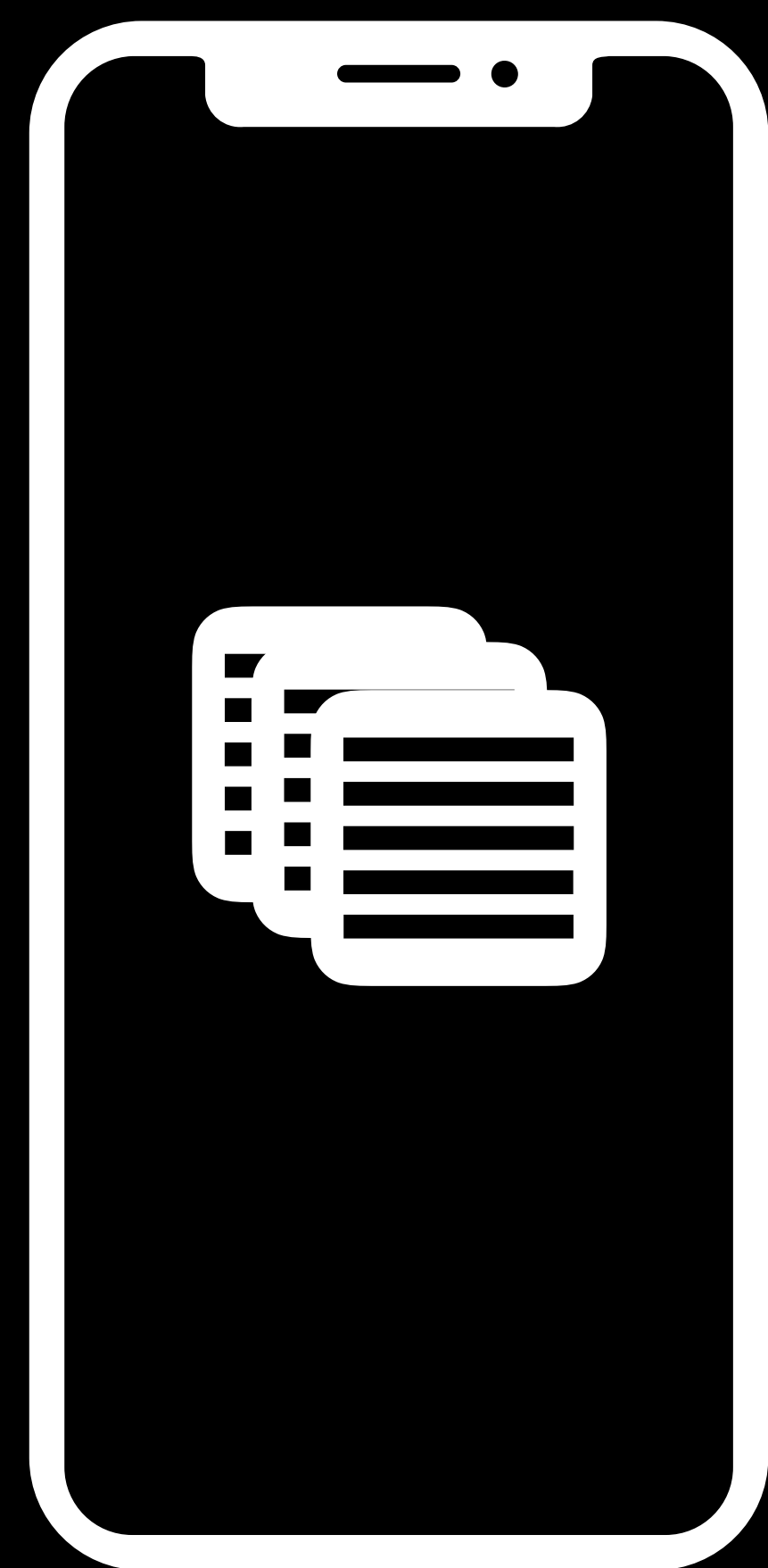


GB/s

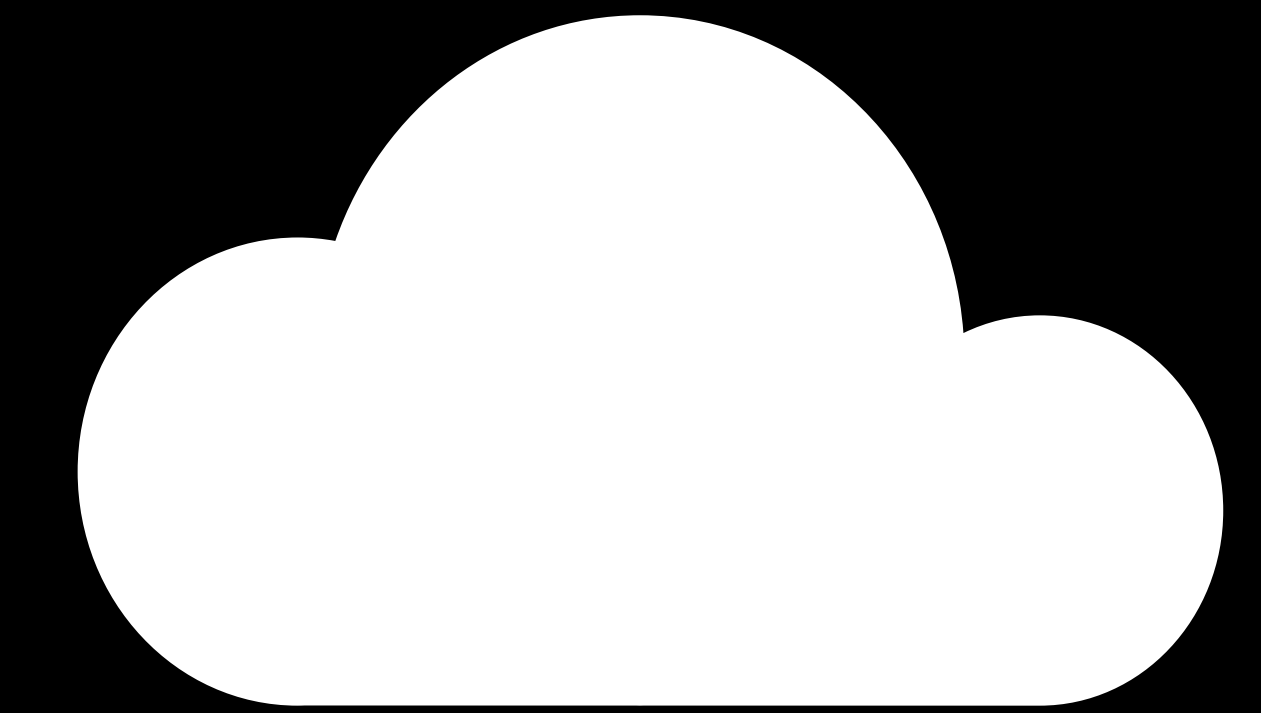
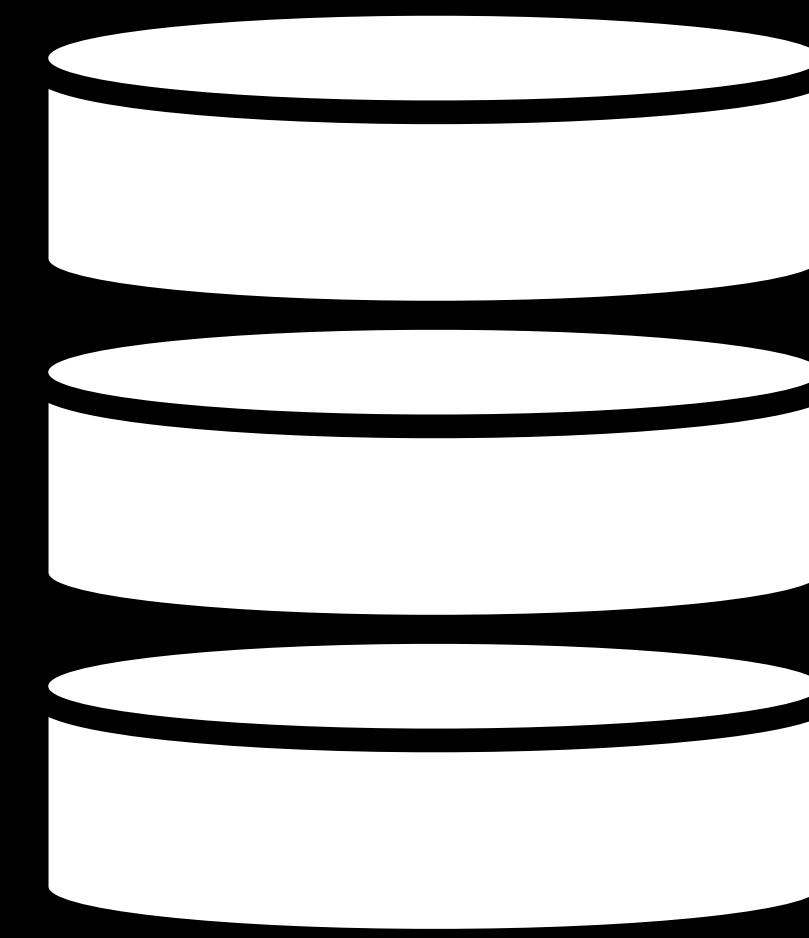
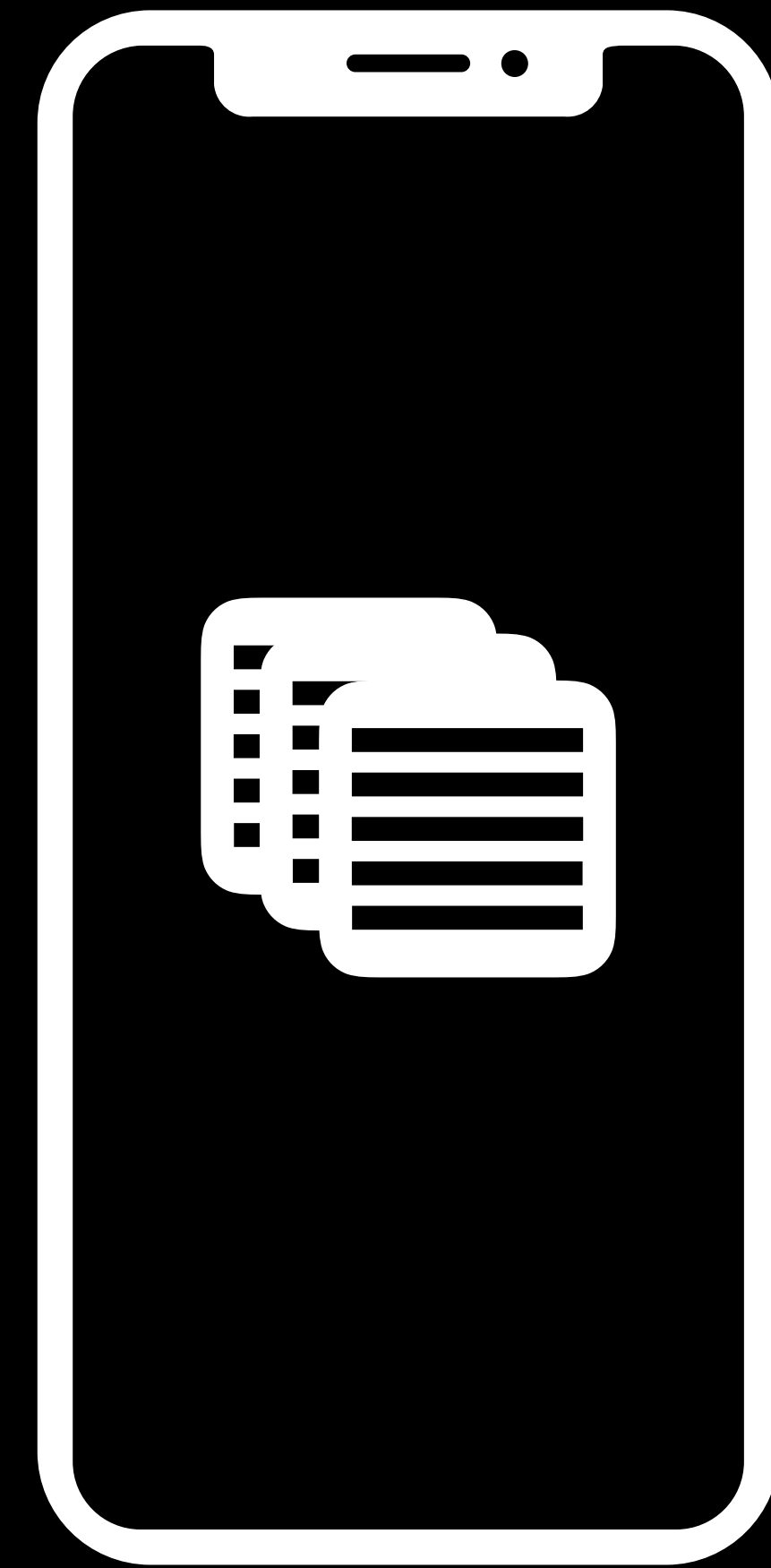


KB/s – MB/s

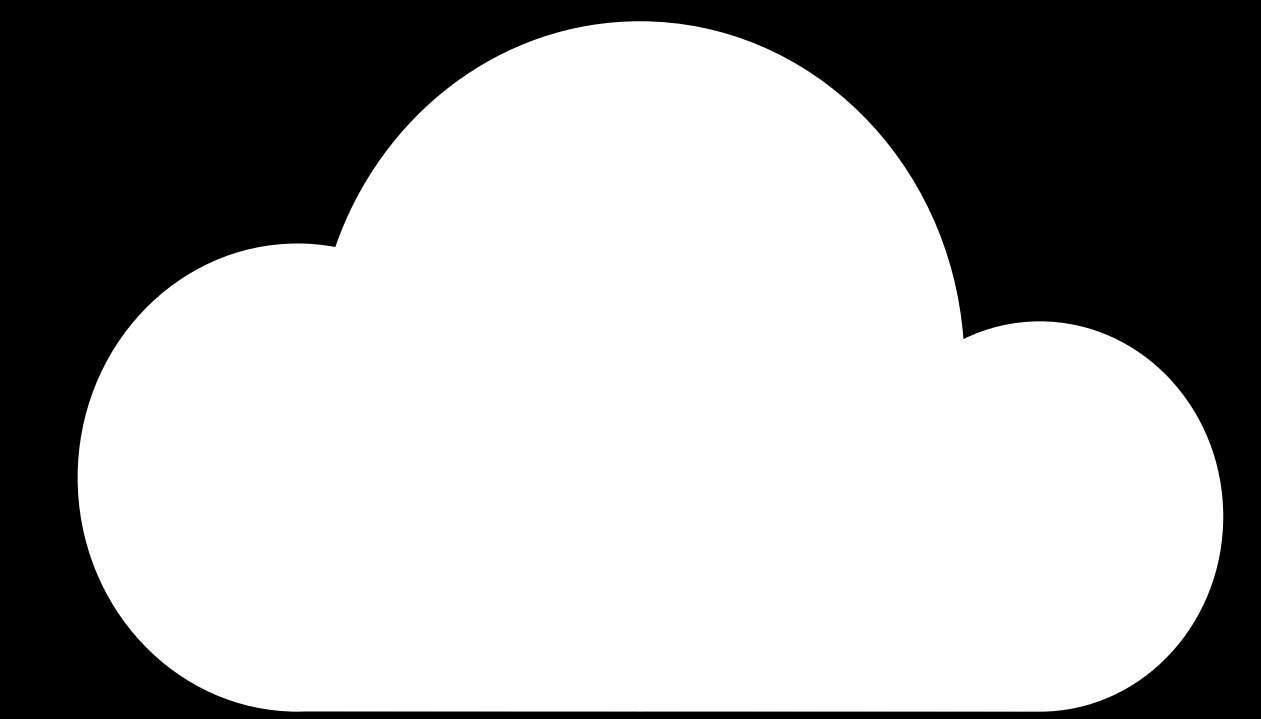
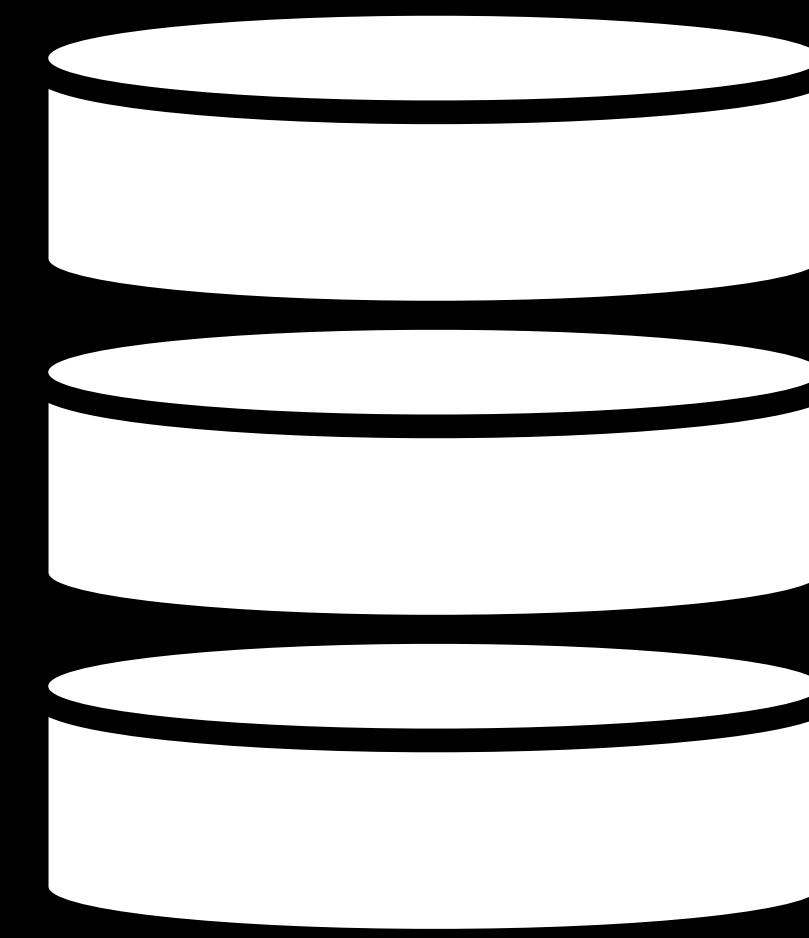
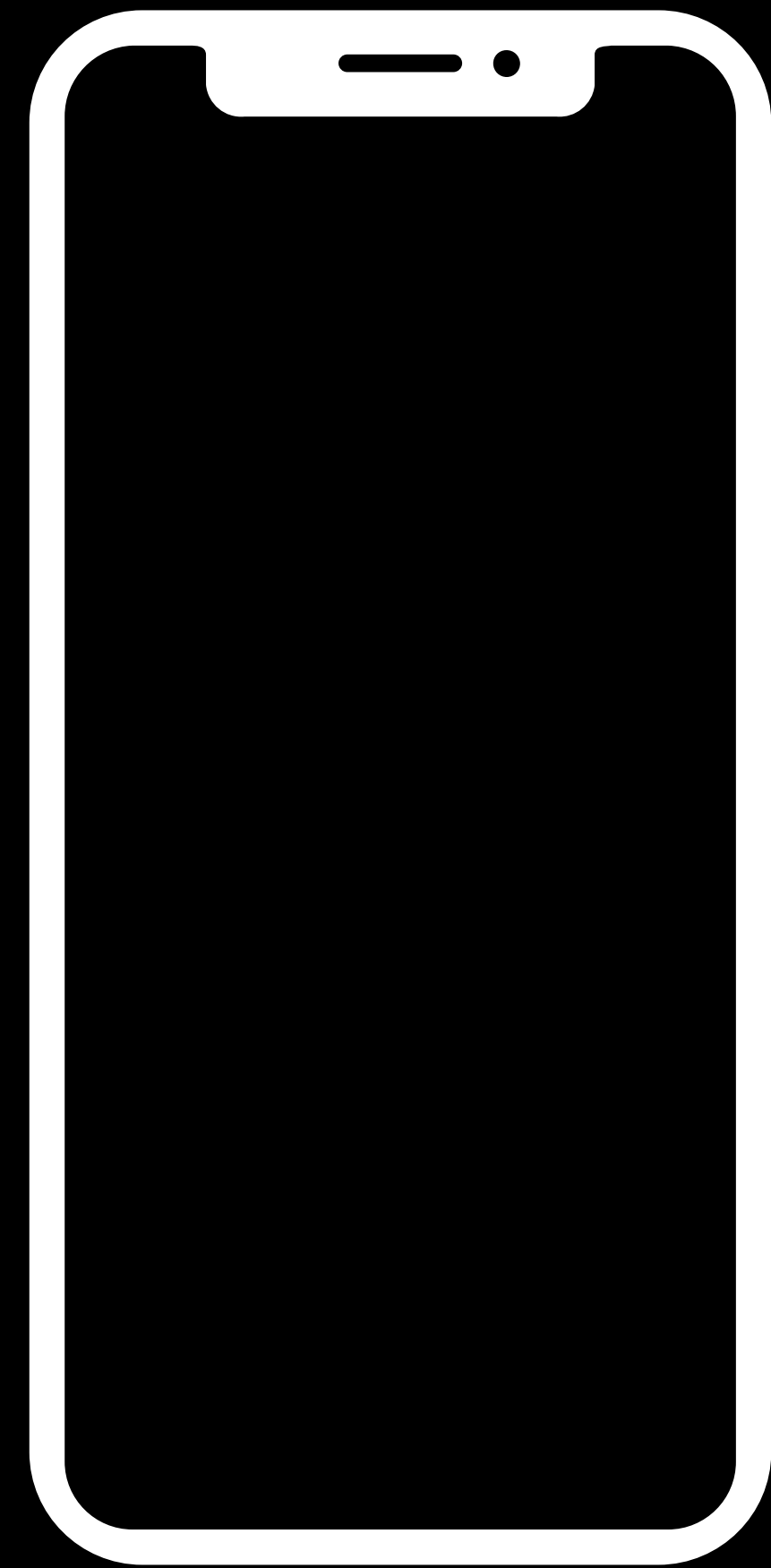
Local Replica



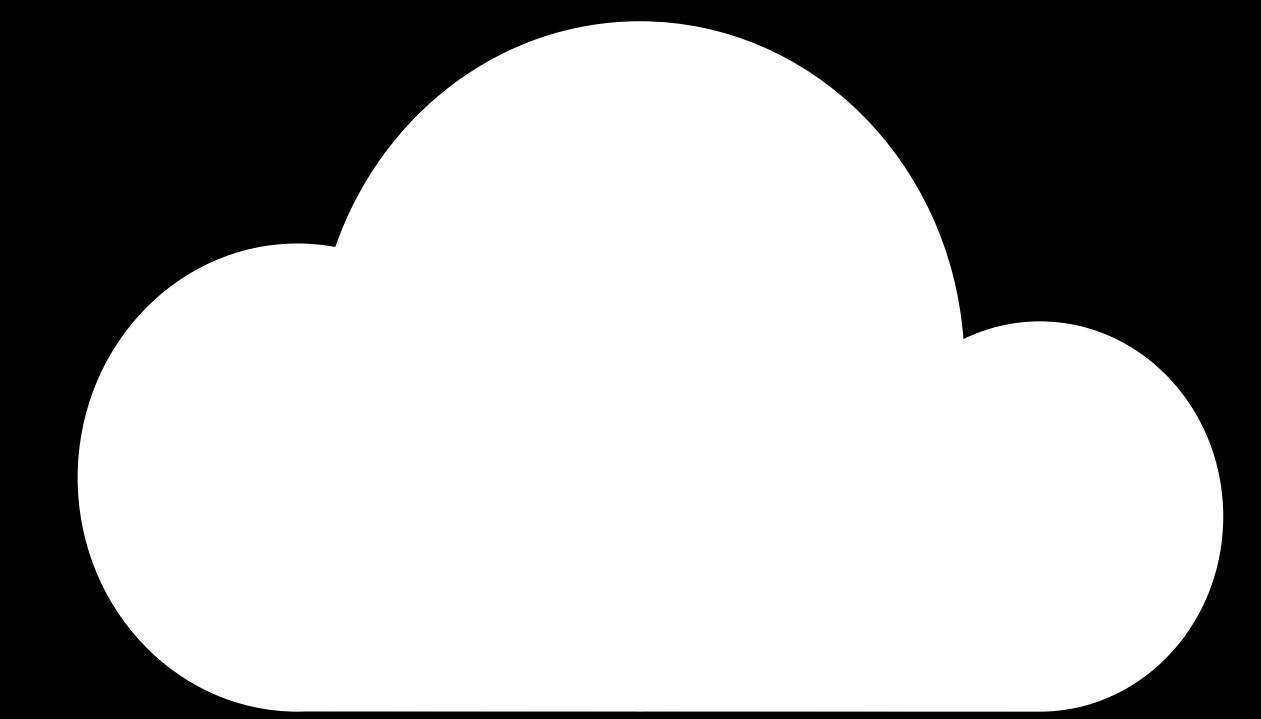
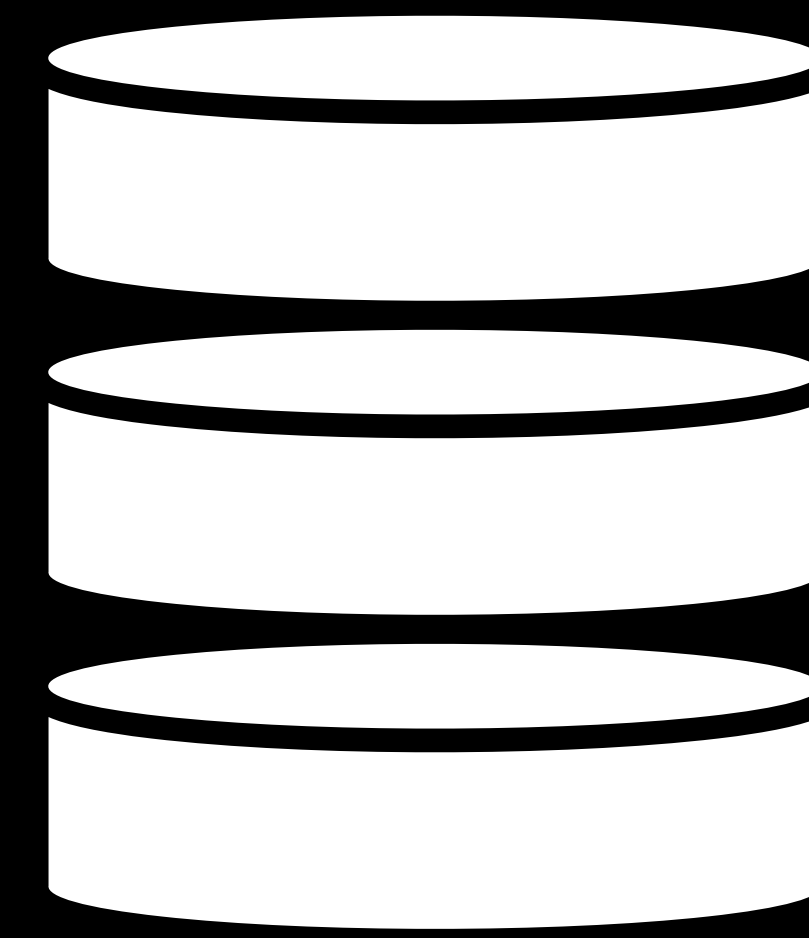
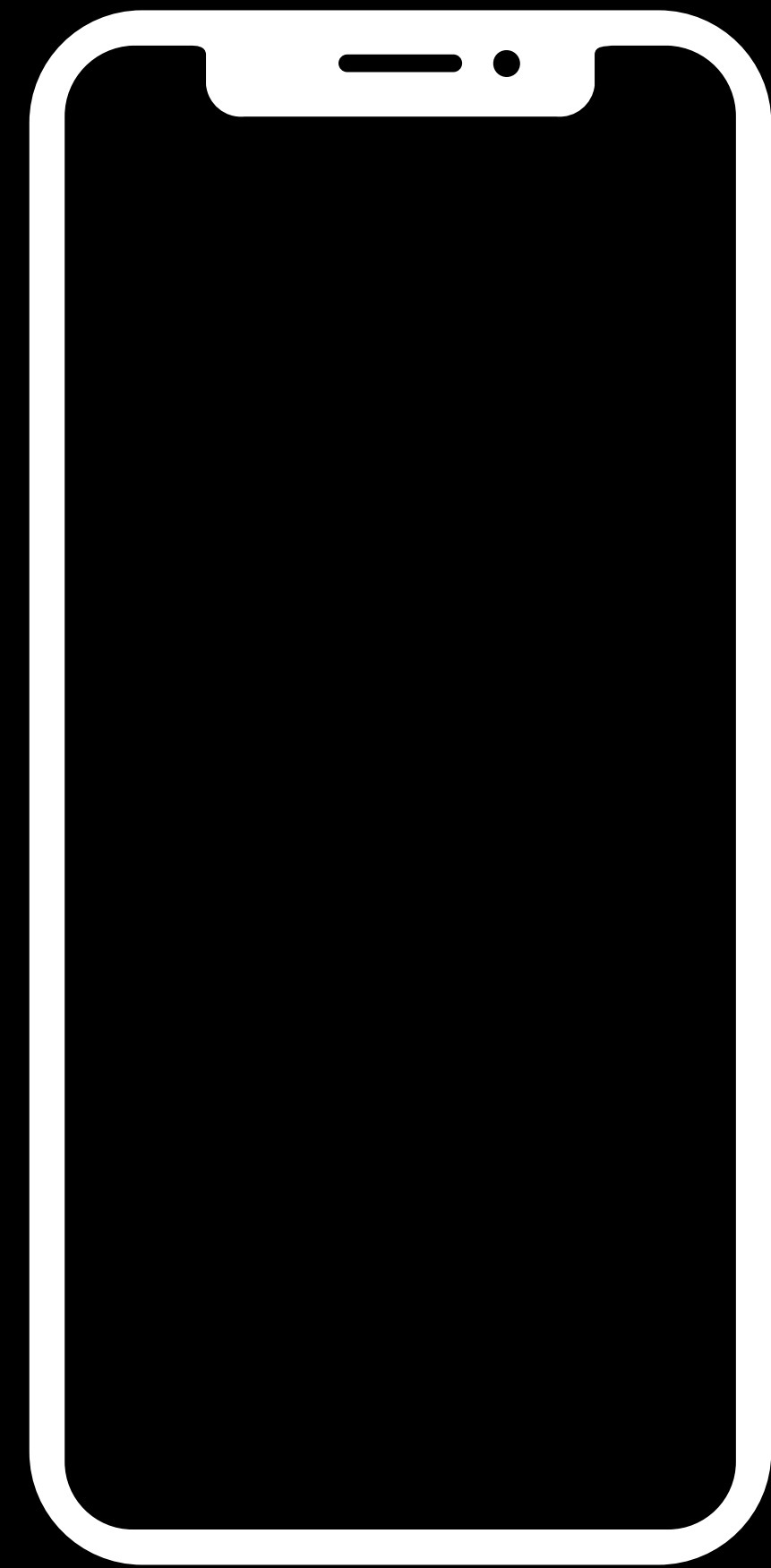
Scheduling



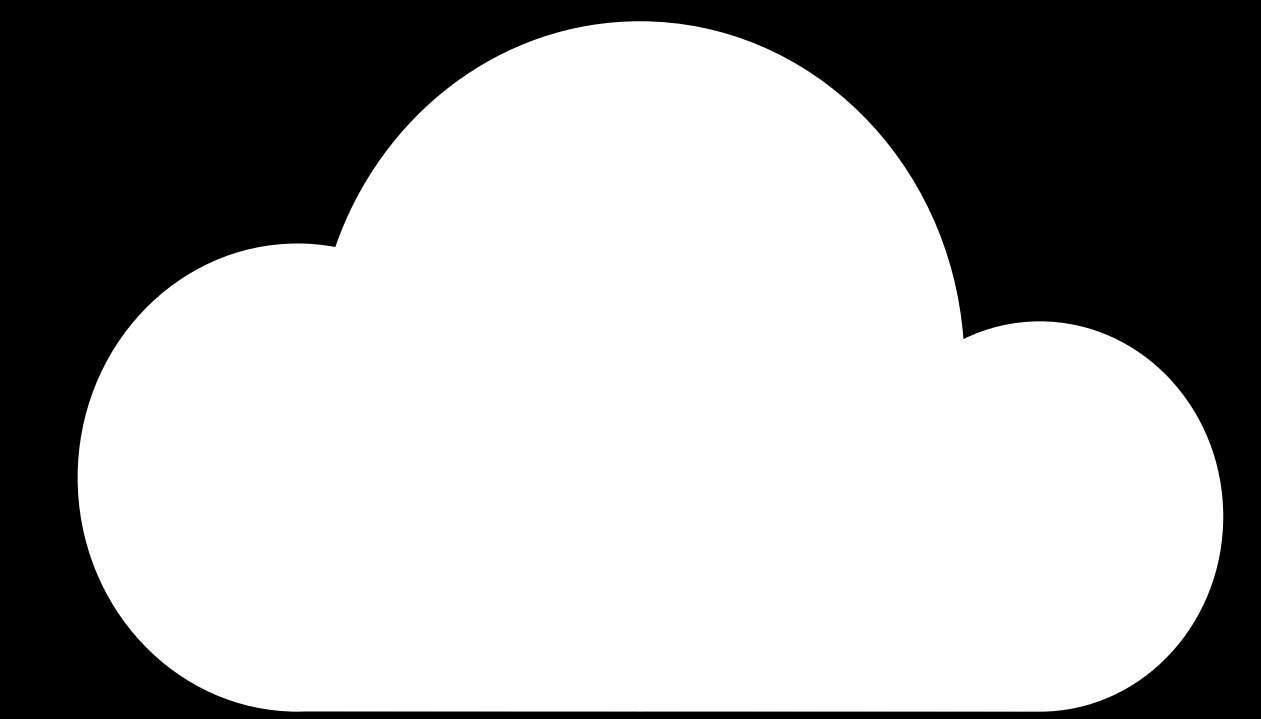
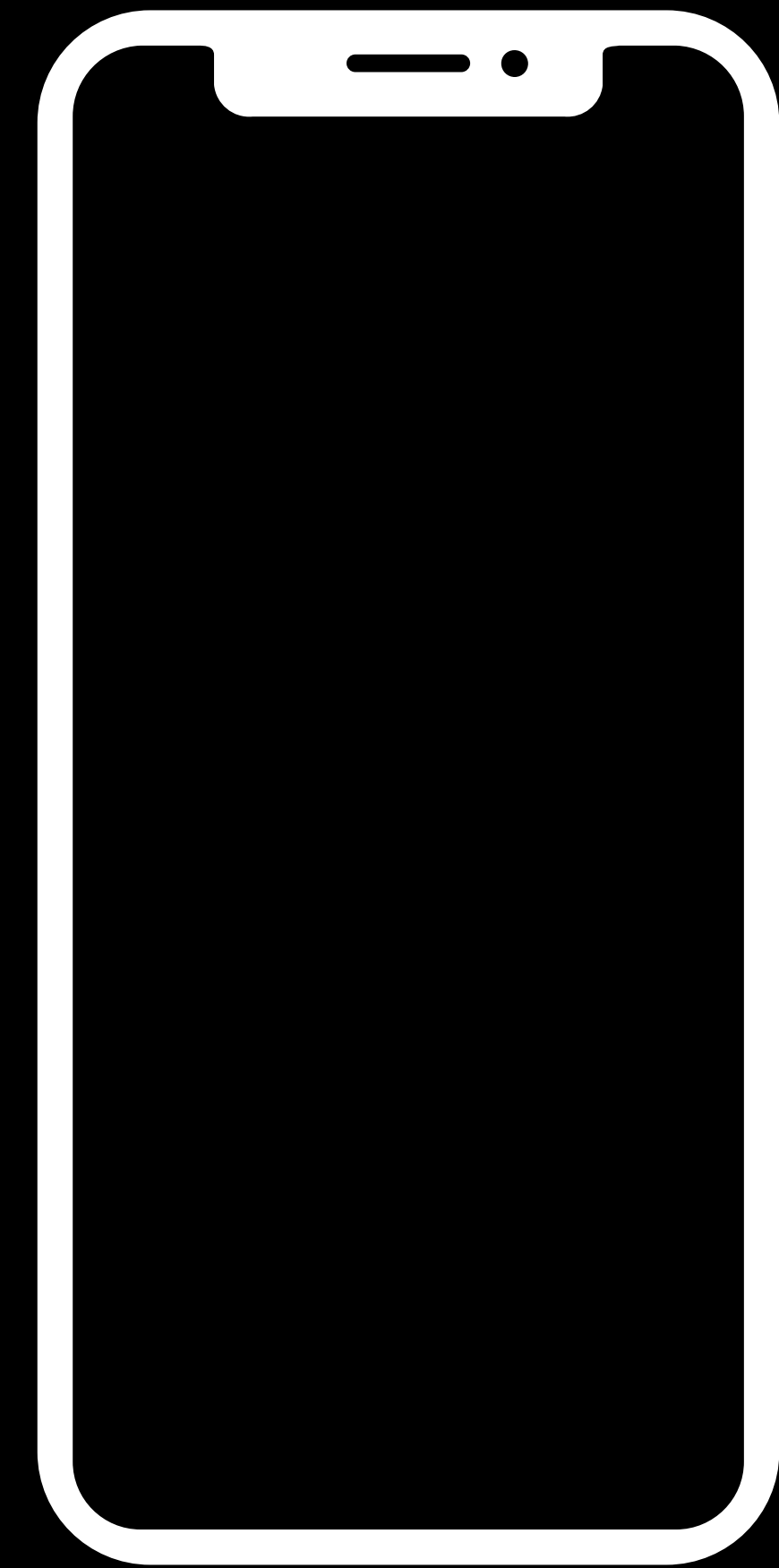
Scheduling



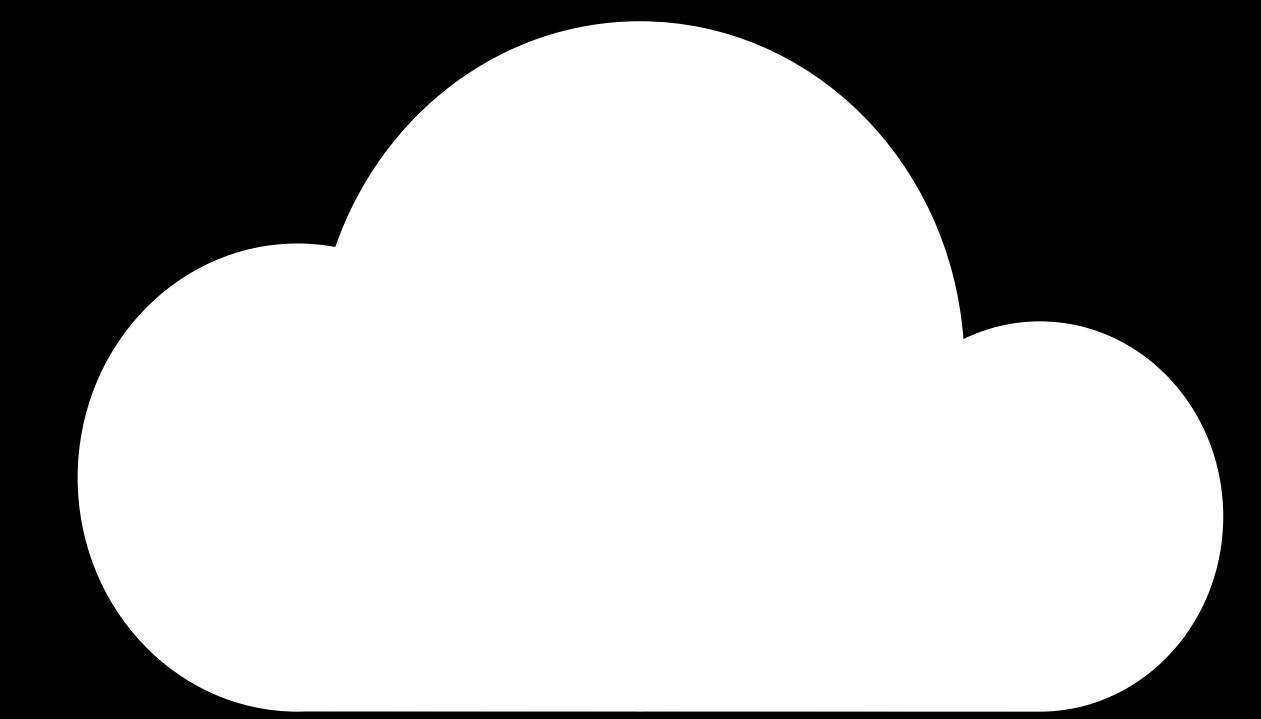
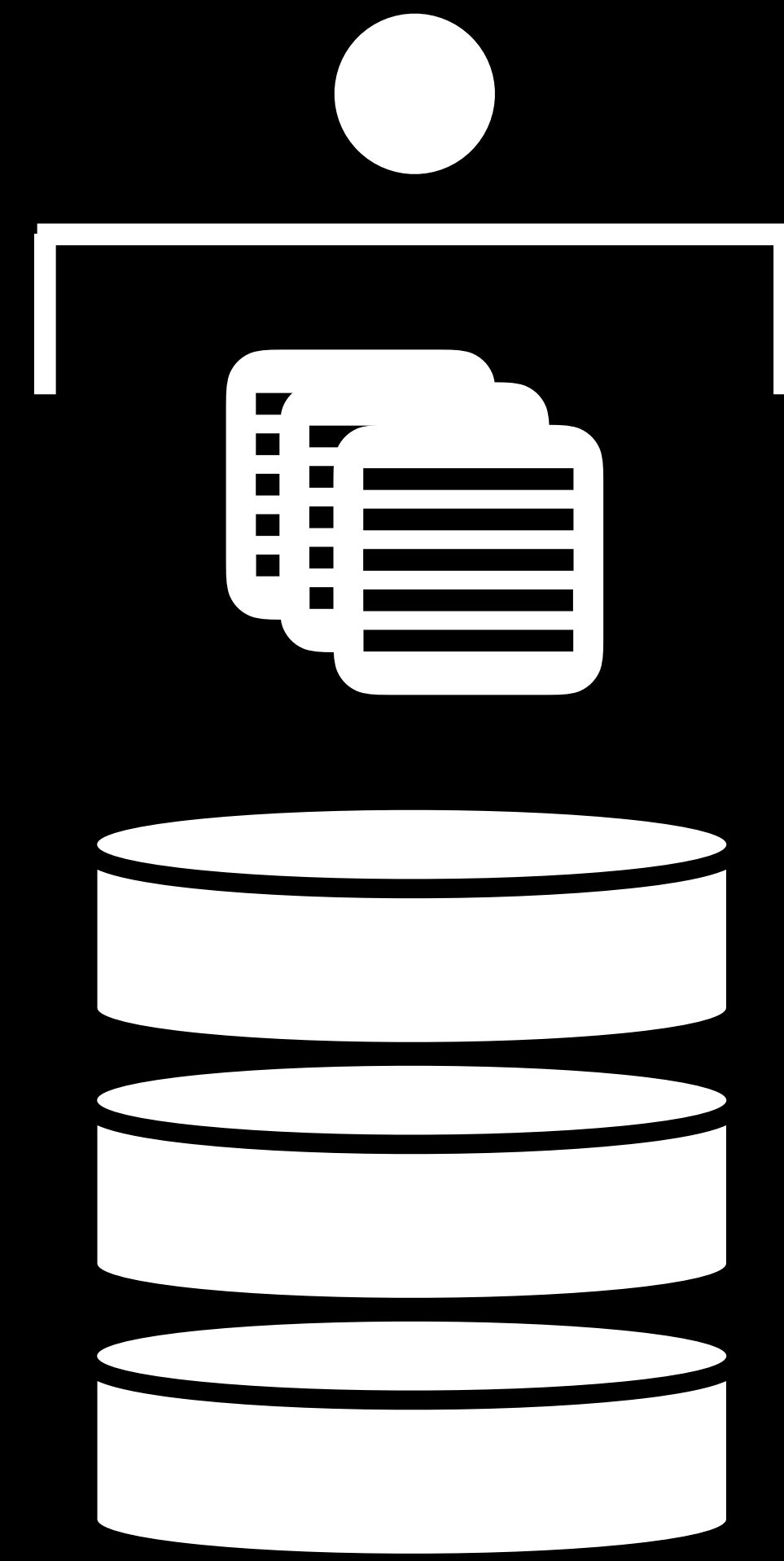
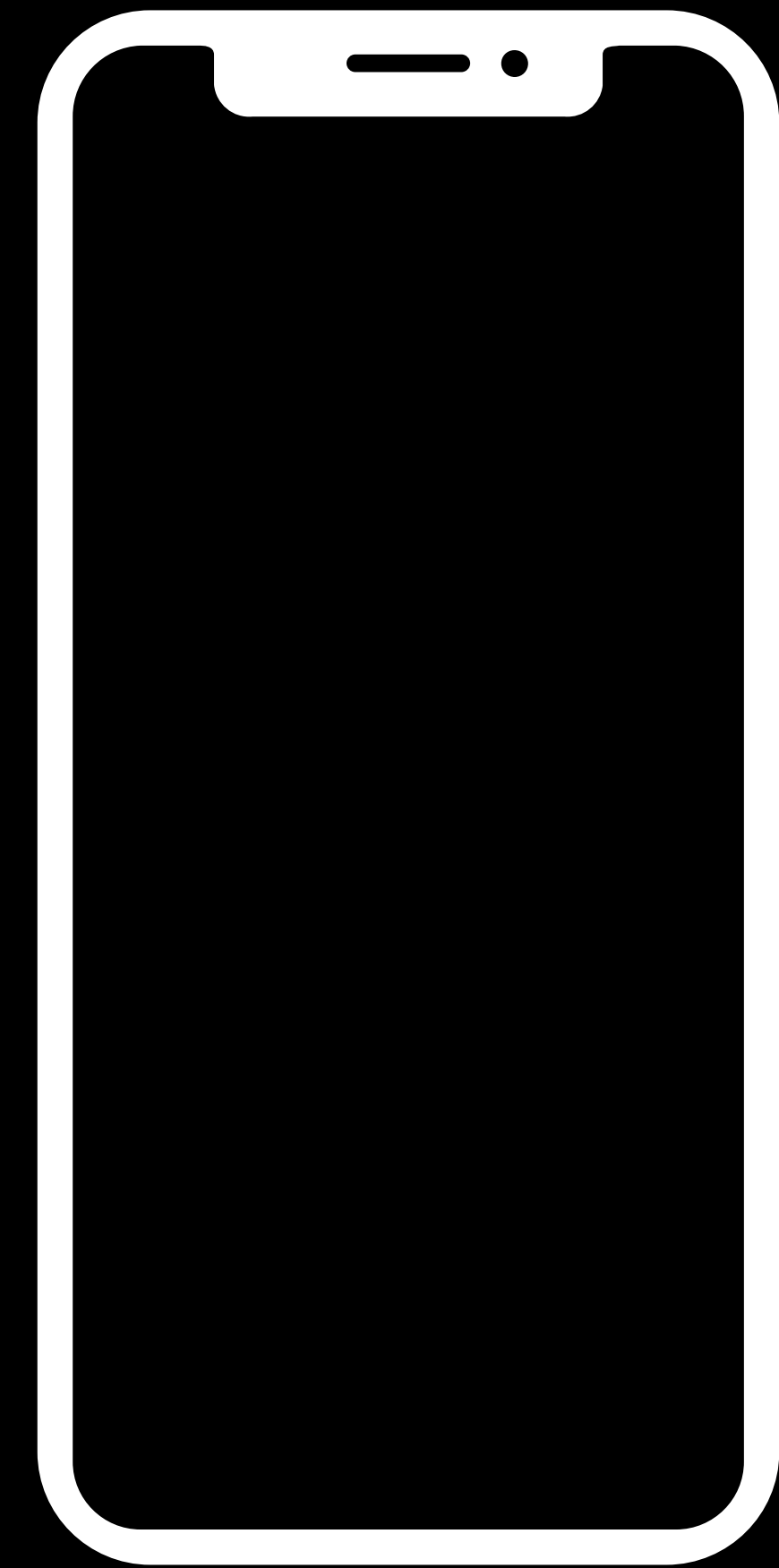
Scheduling



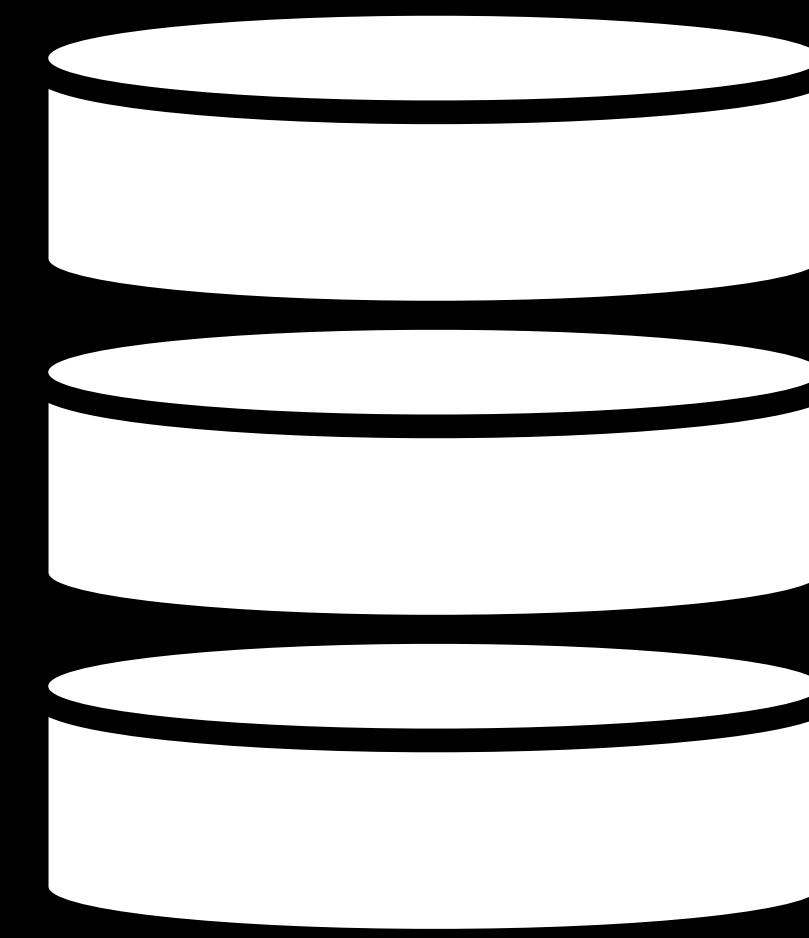
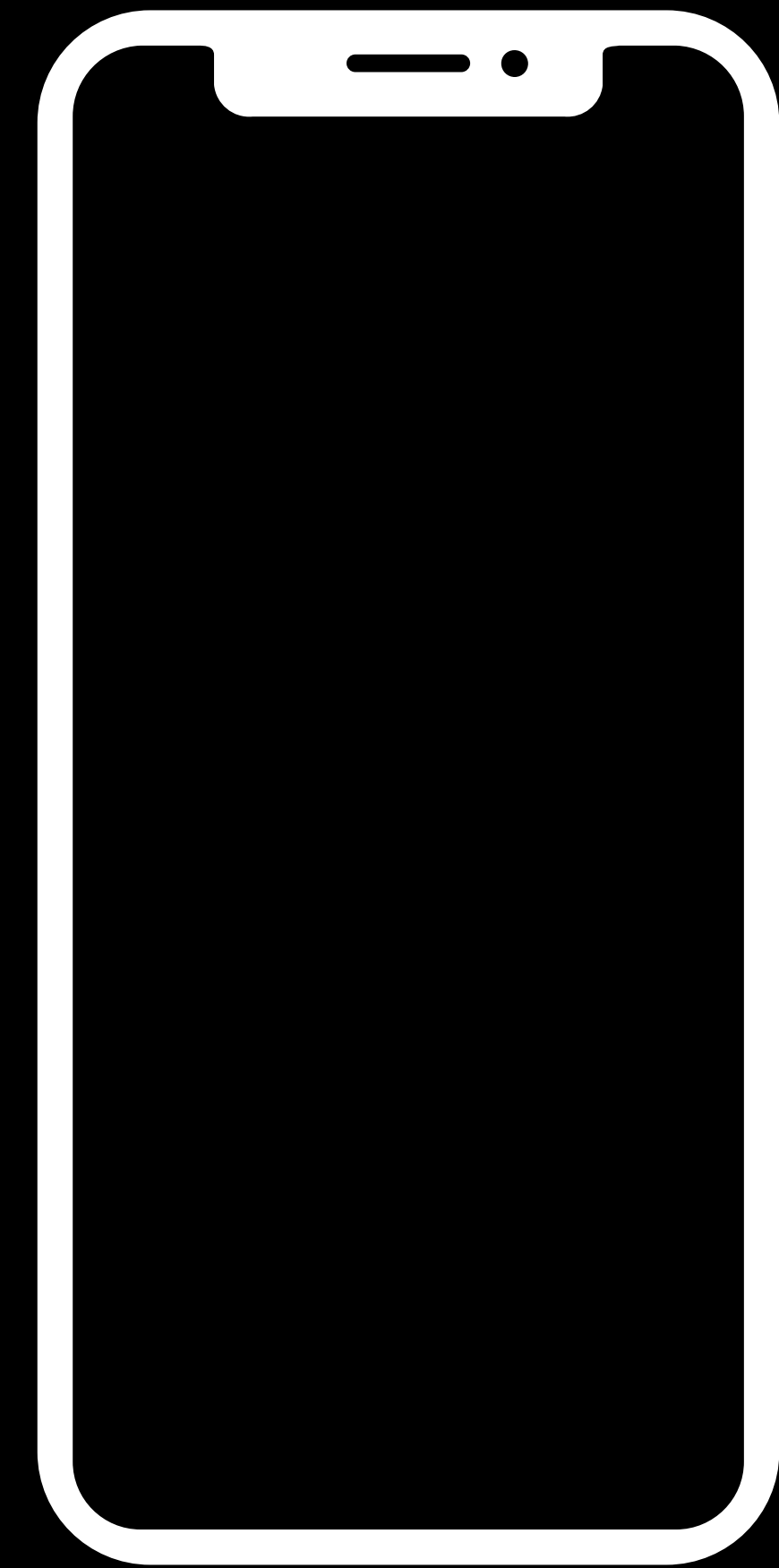
Scheduling



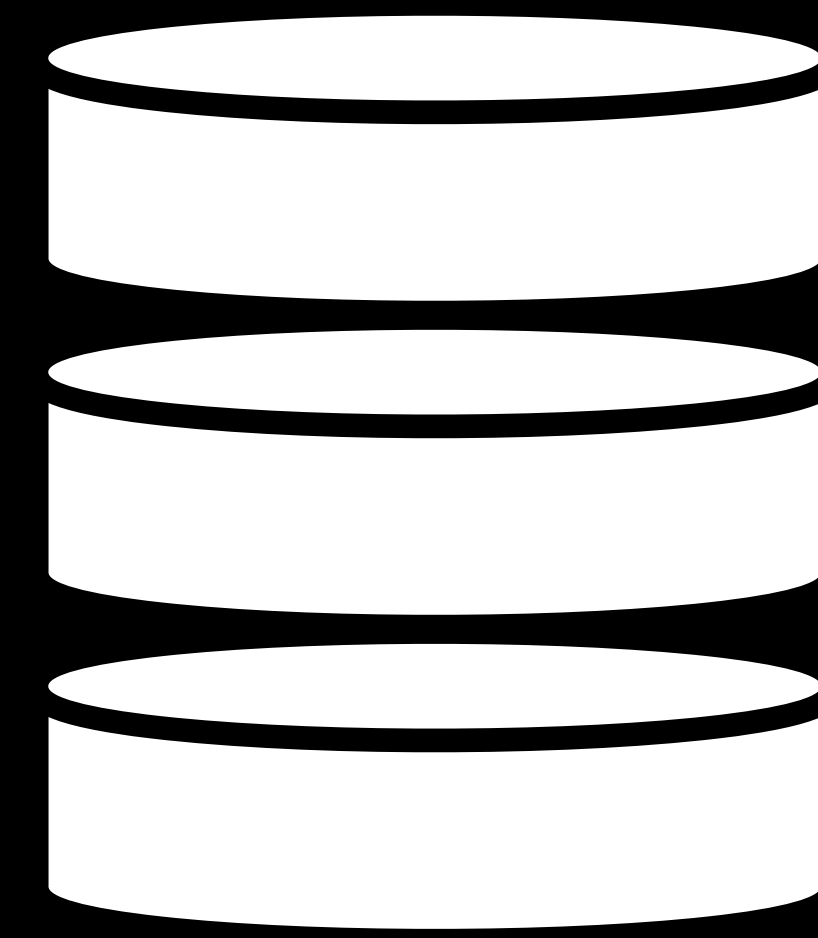
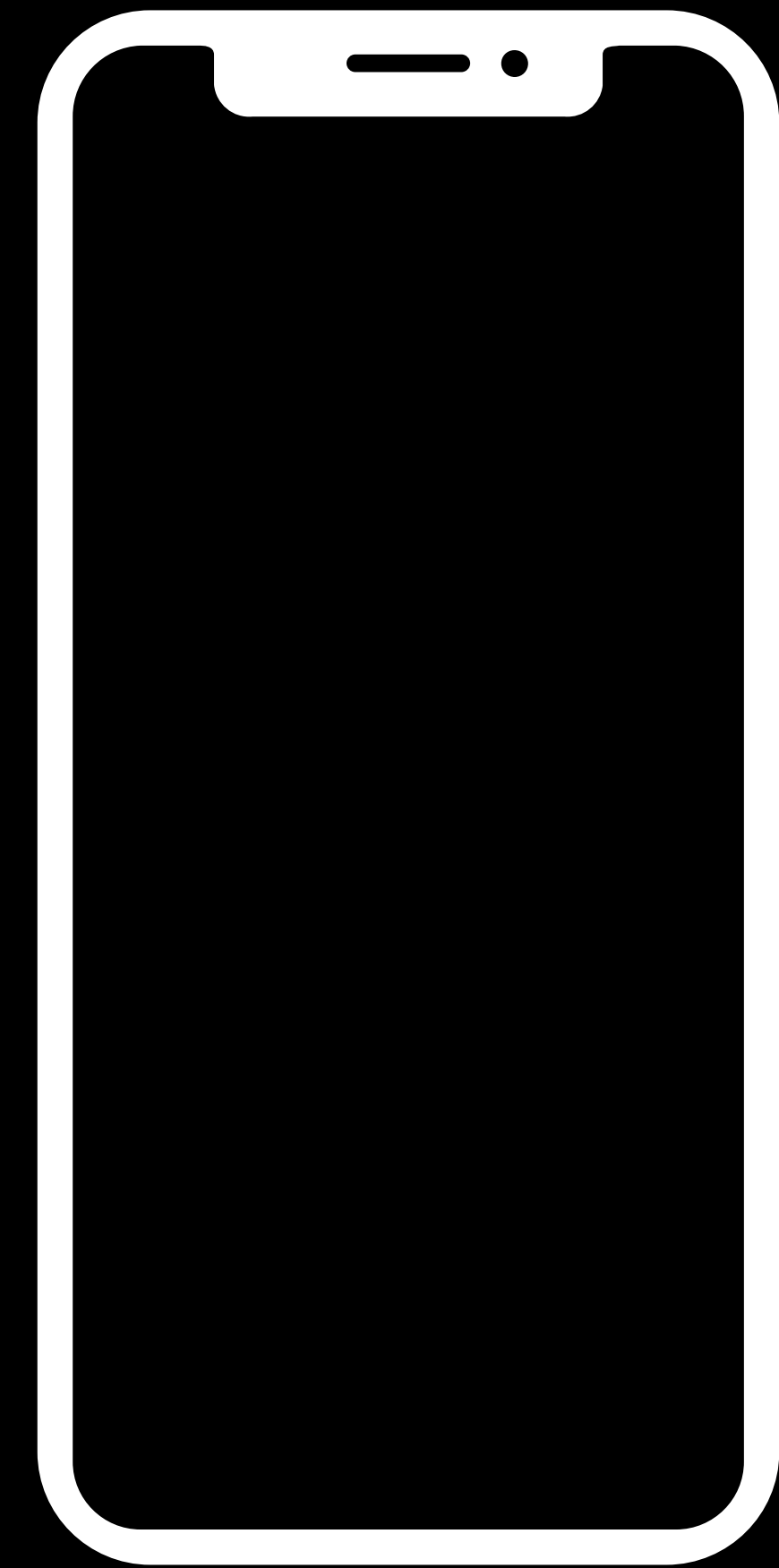
Scheduling



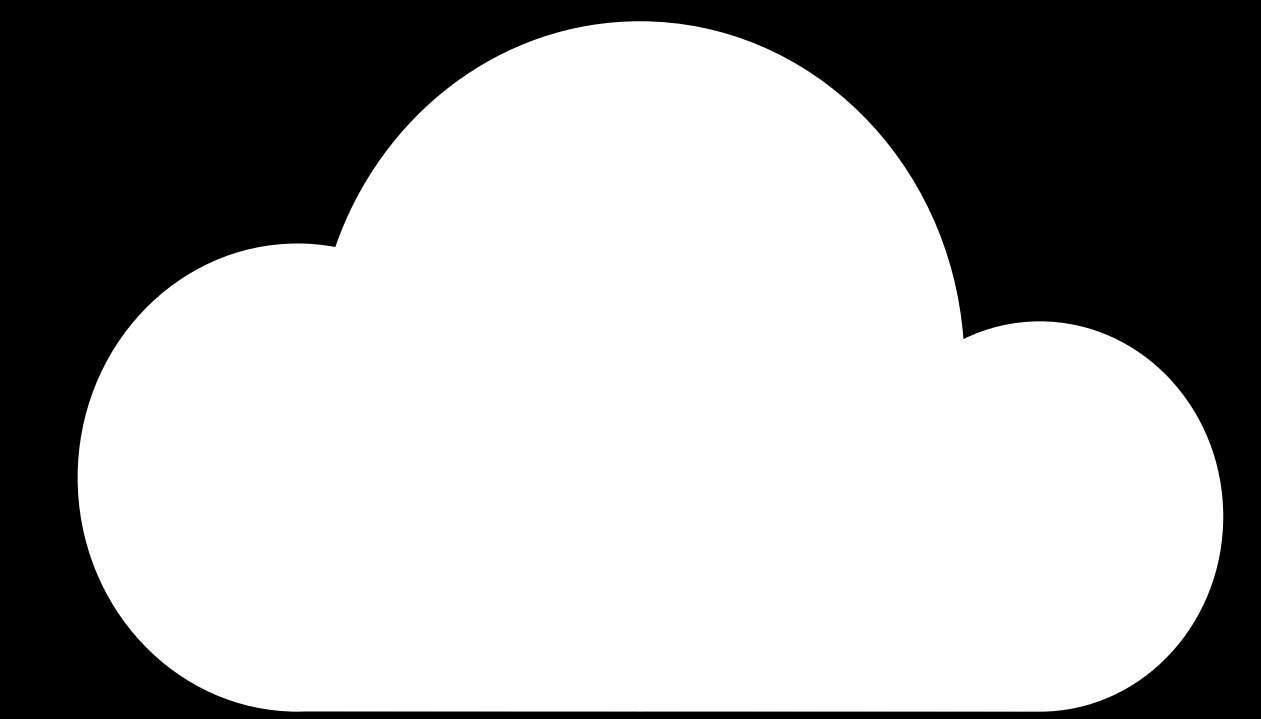
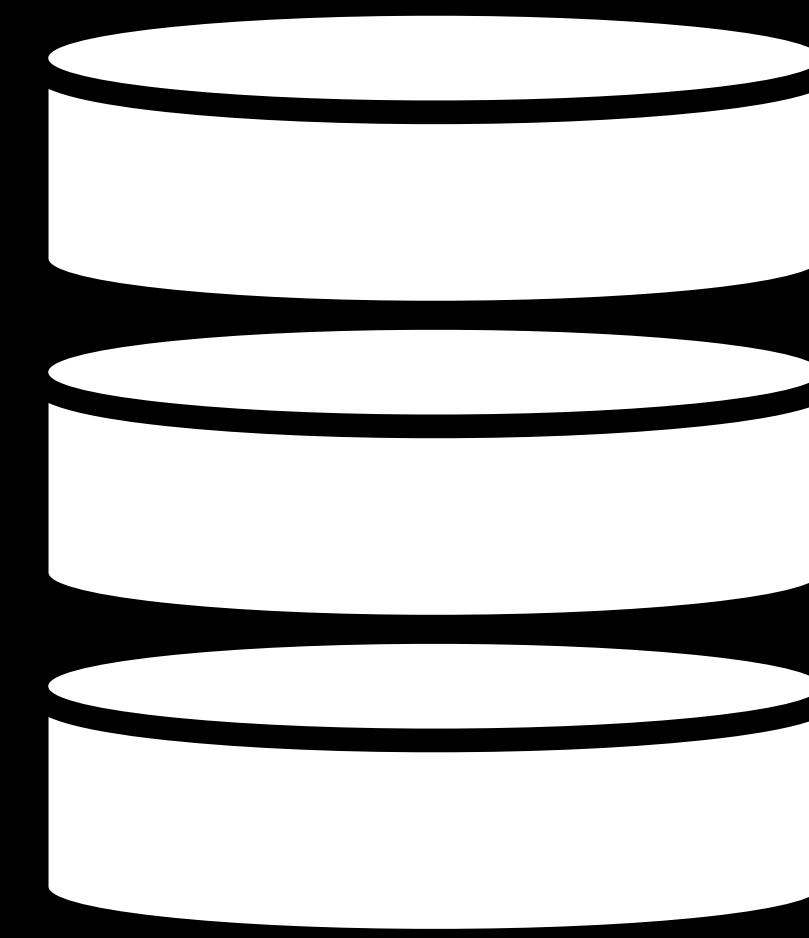
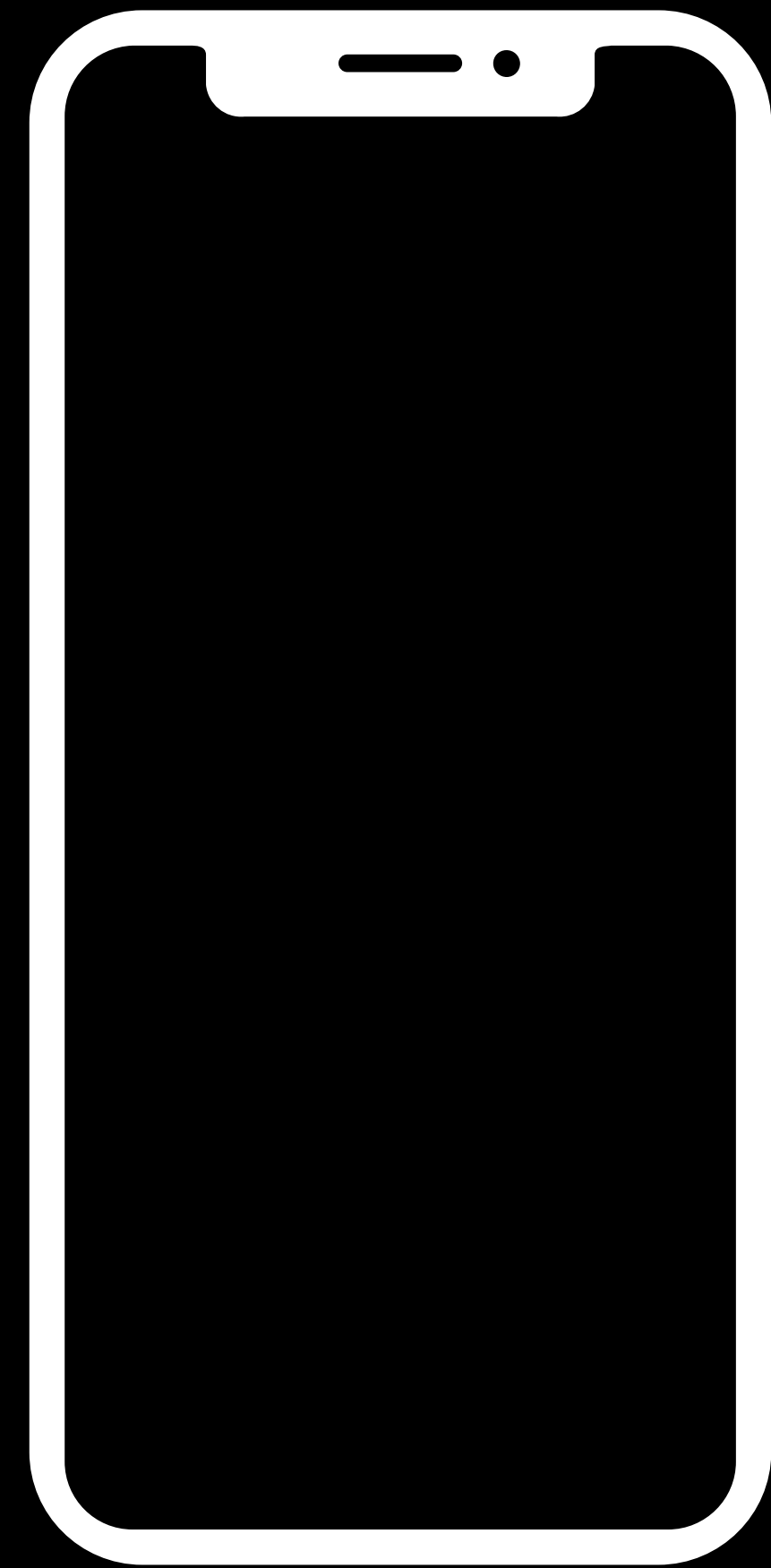
Scheduling



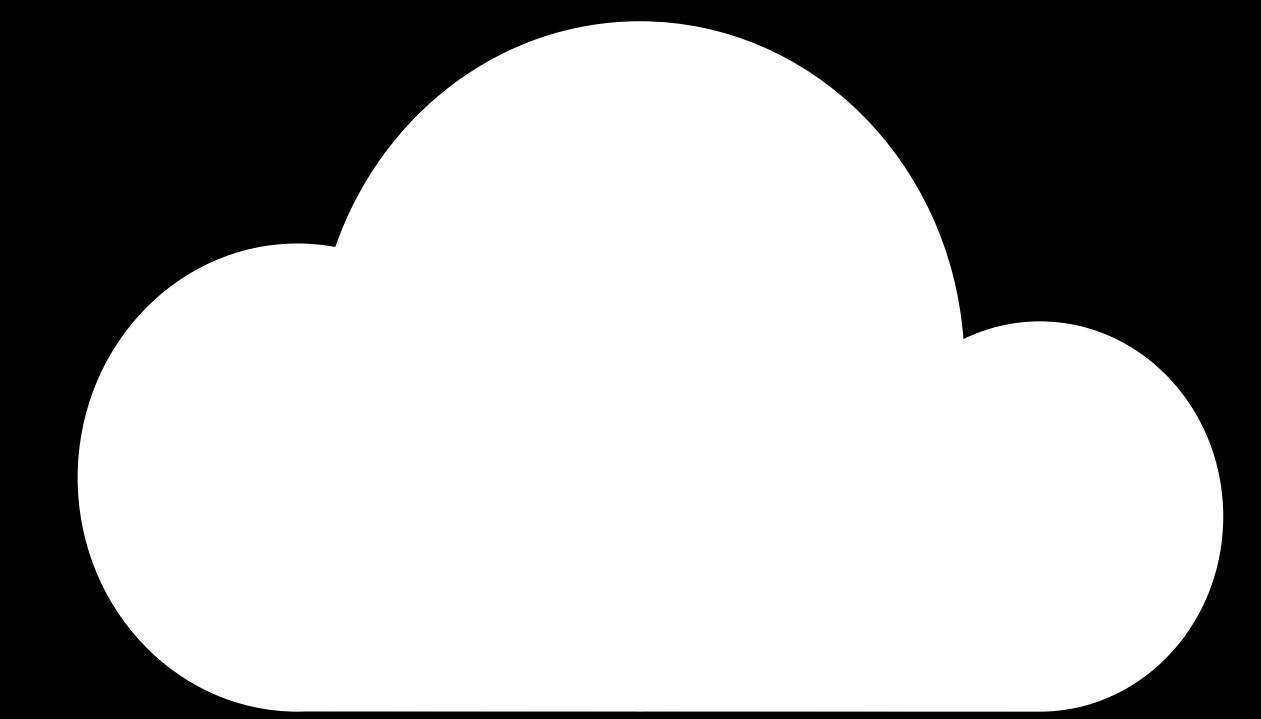
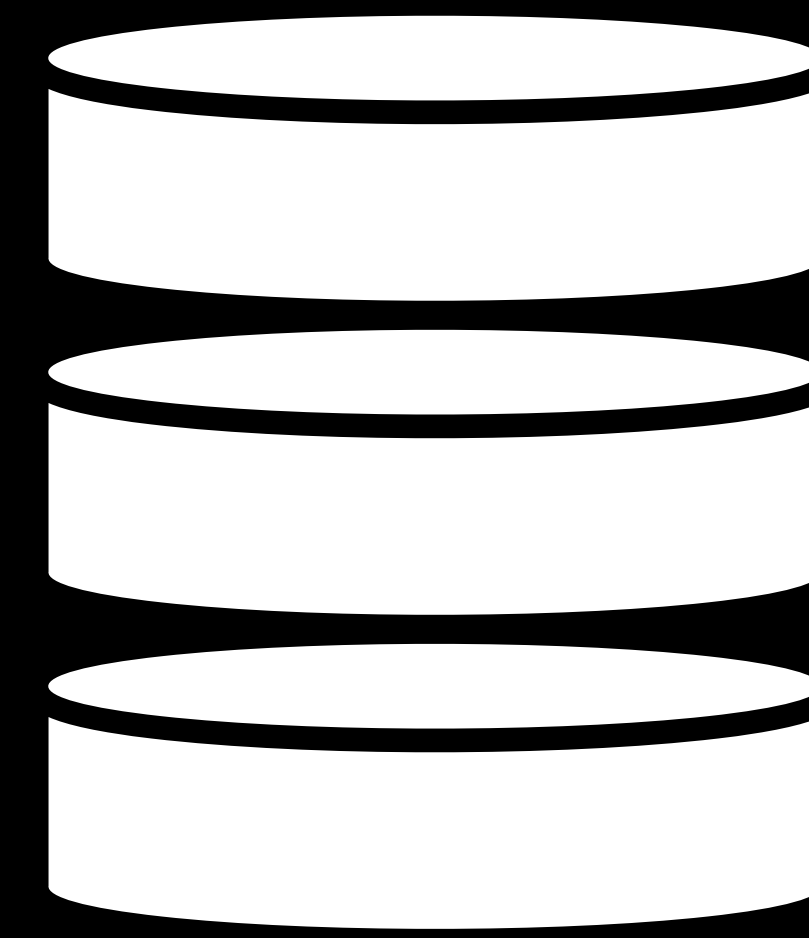
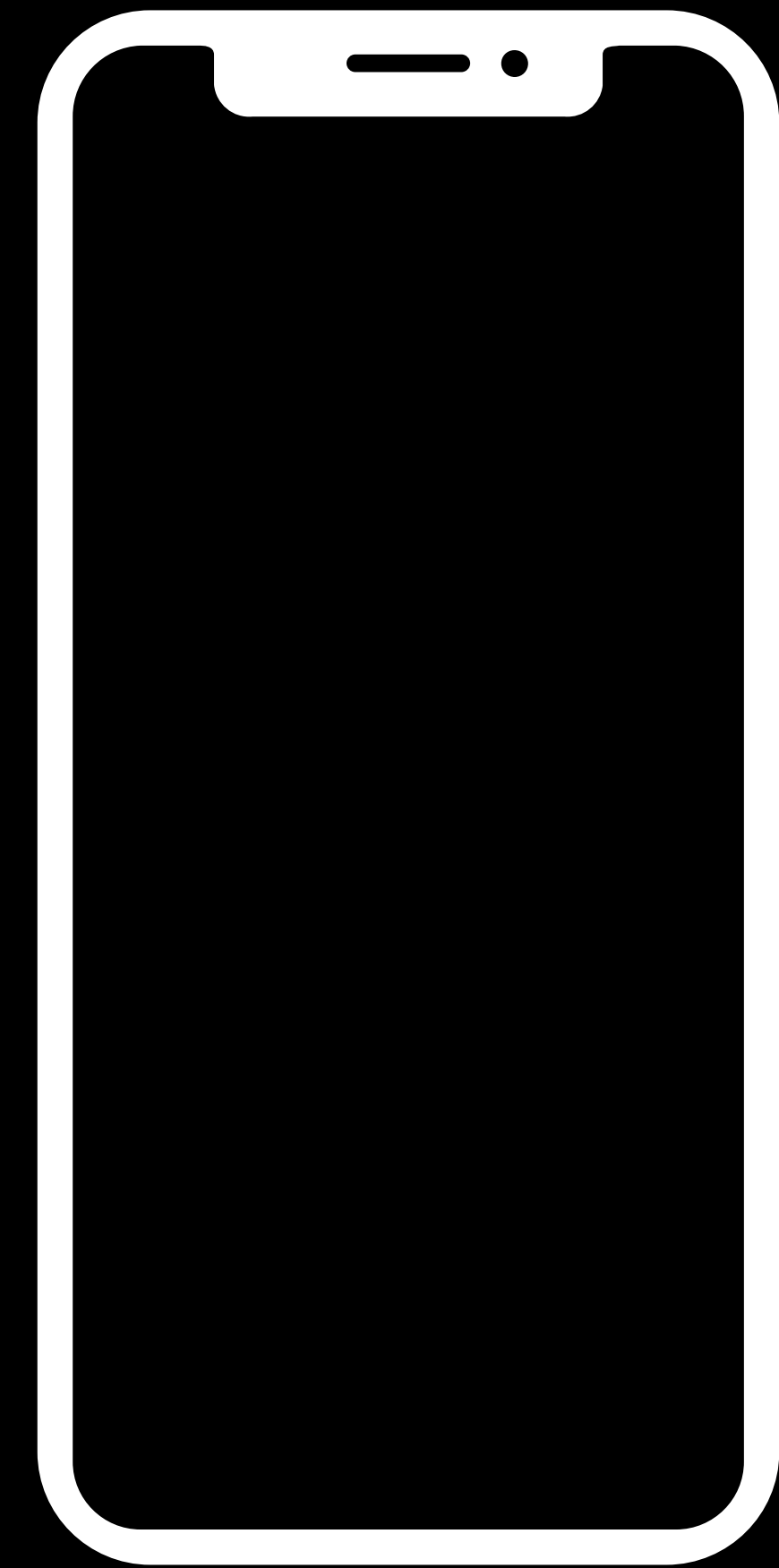
Scheduling



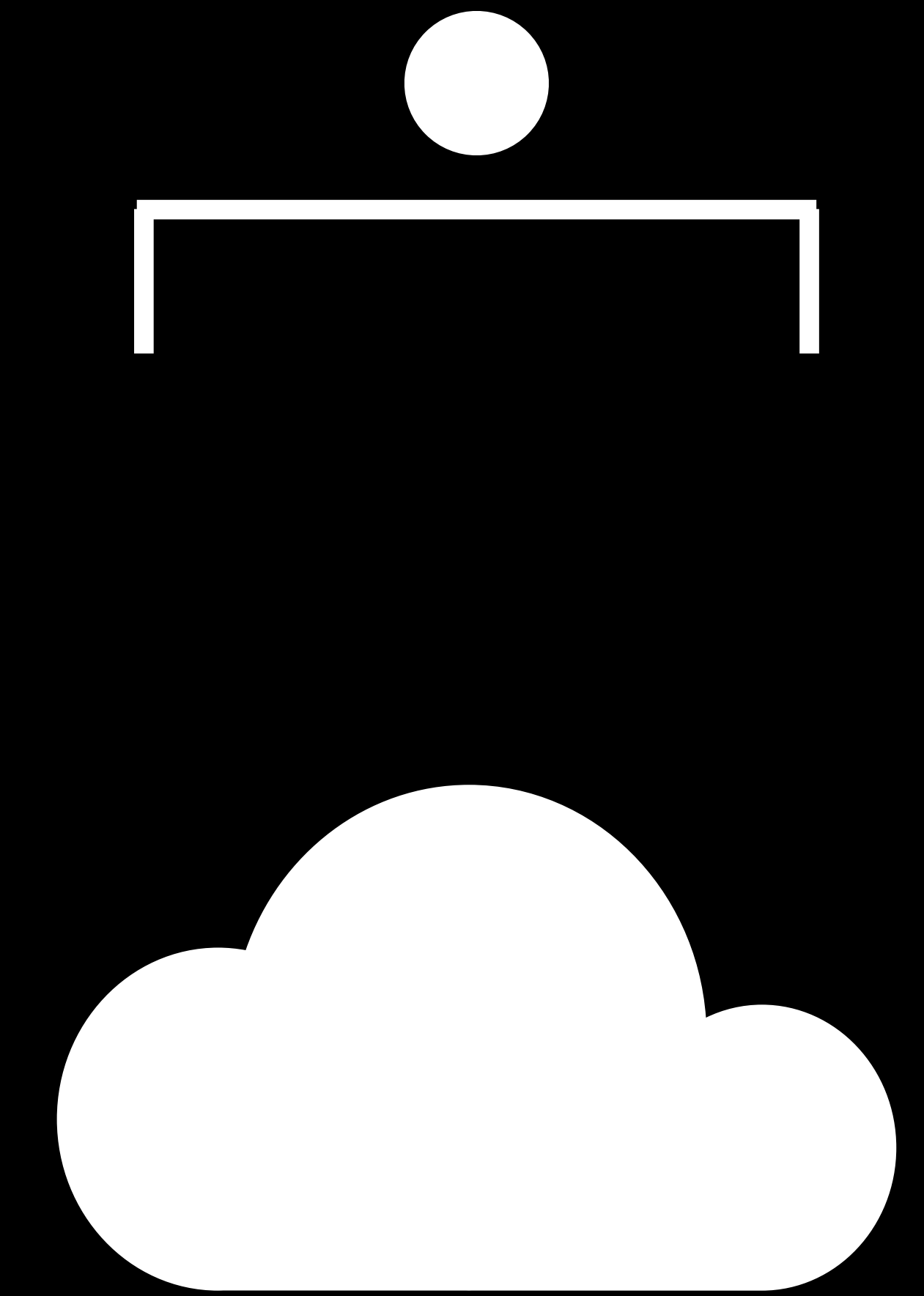
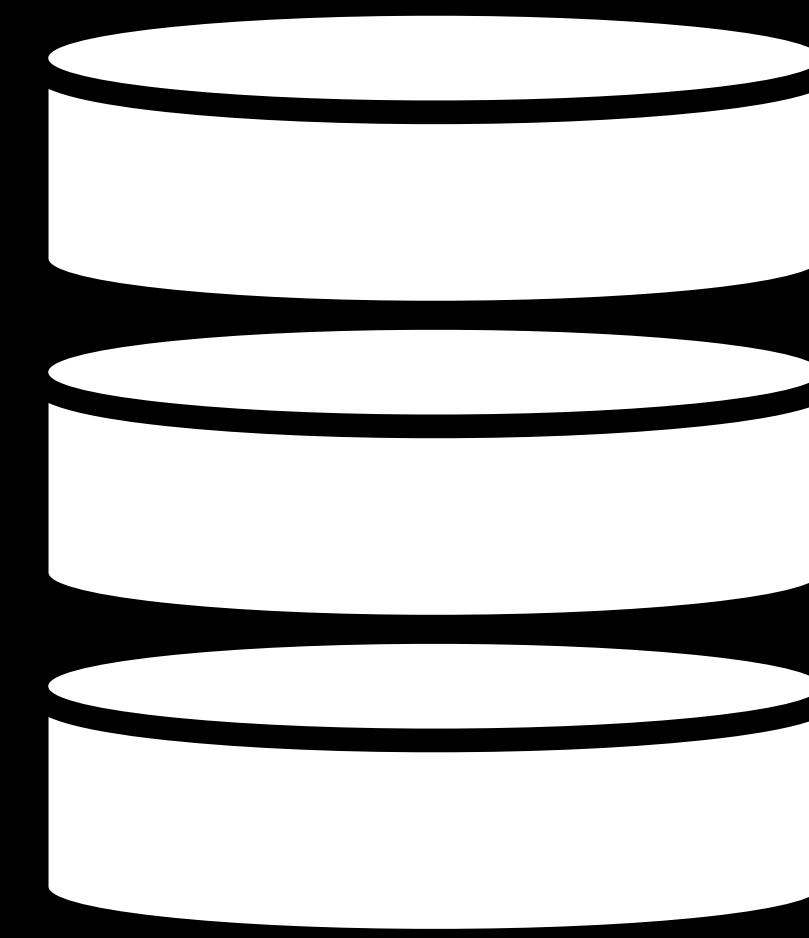
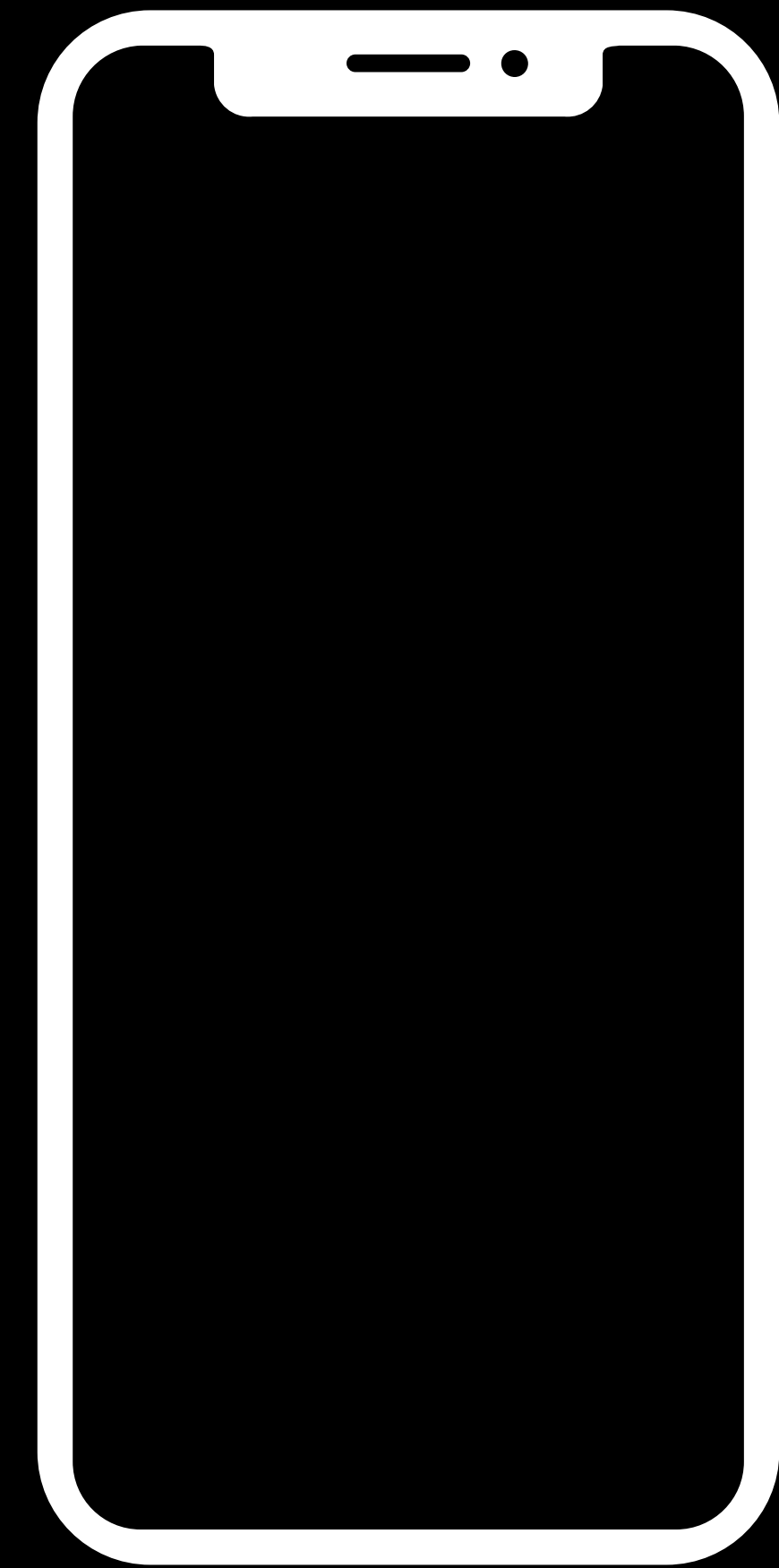
Scheduling



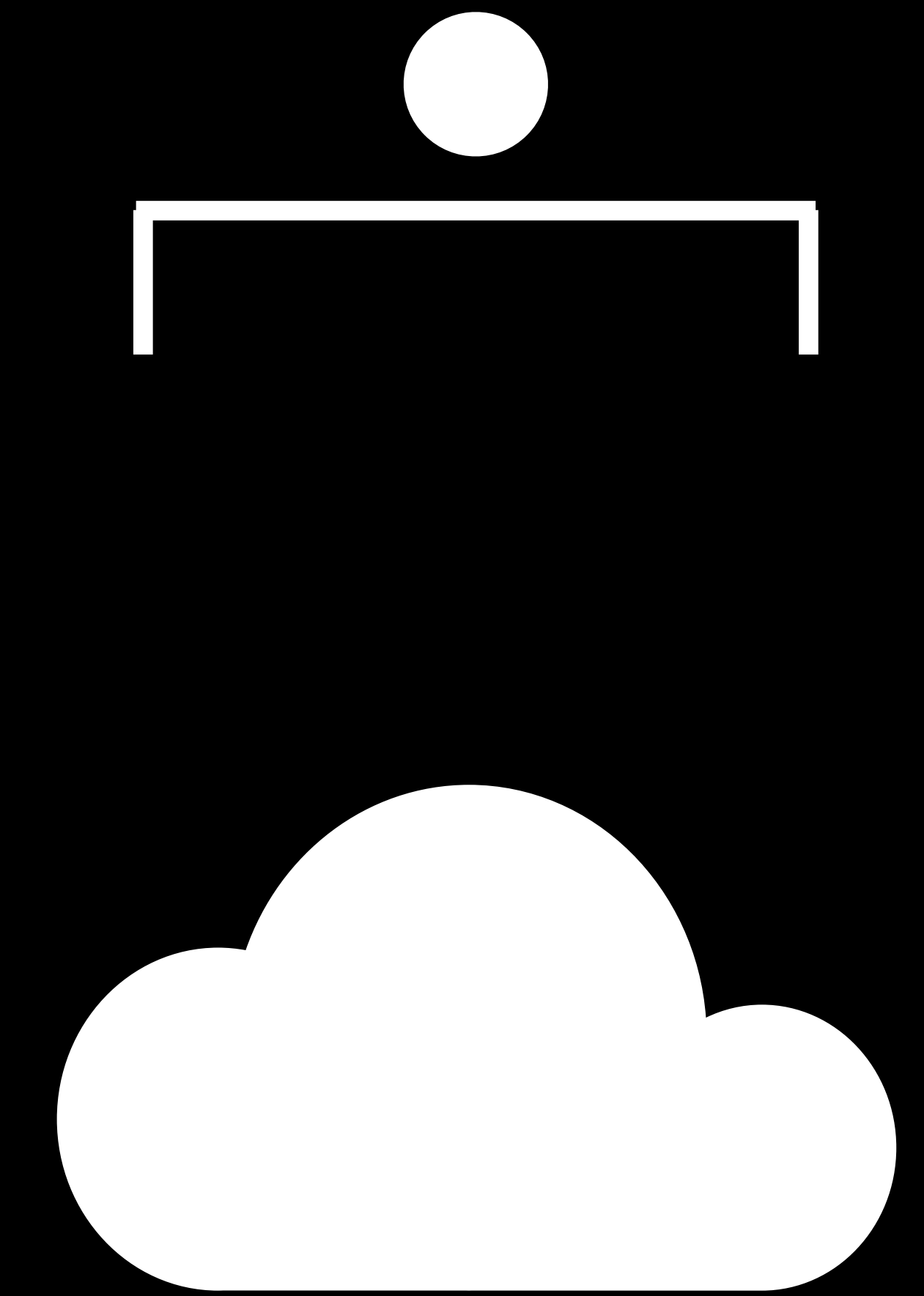
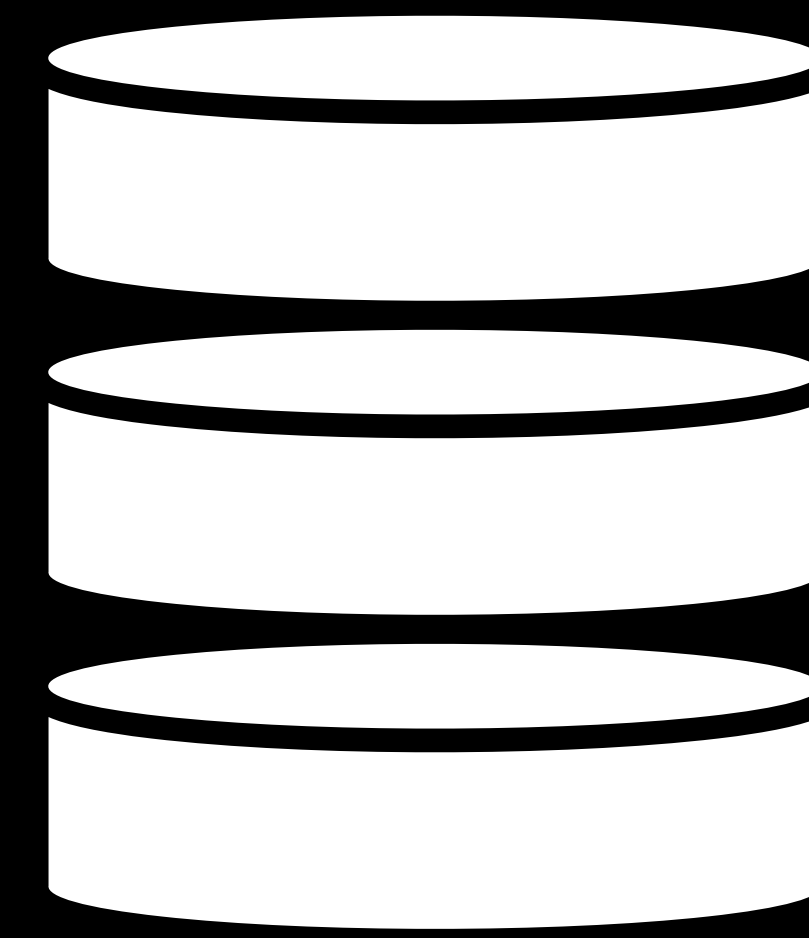
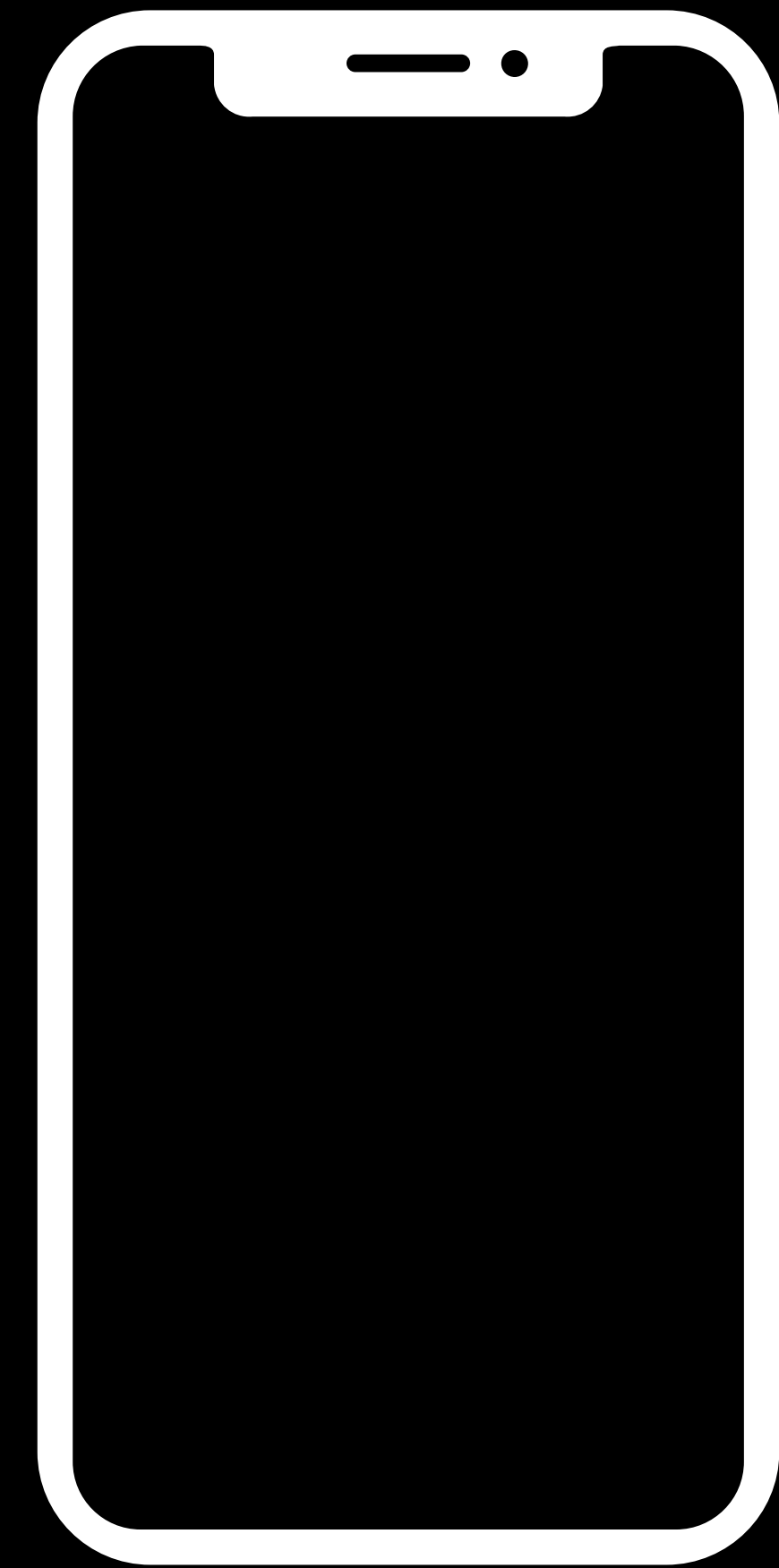
Scheduling



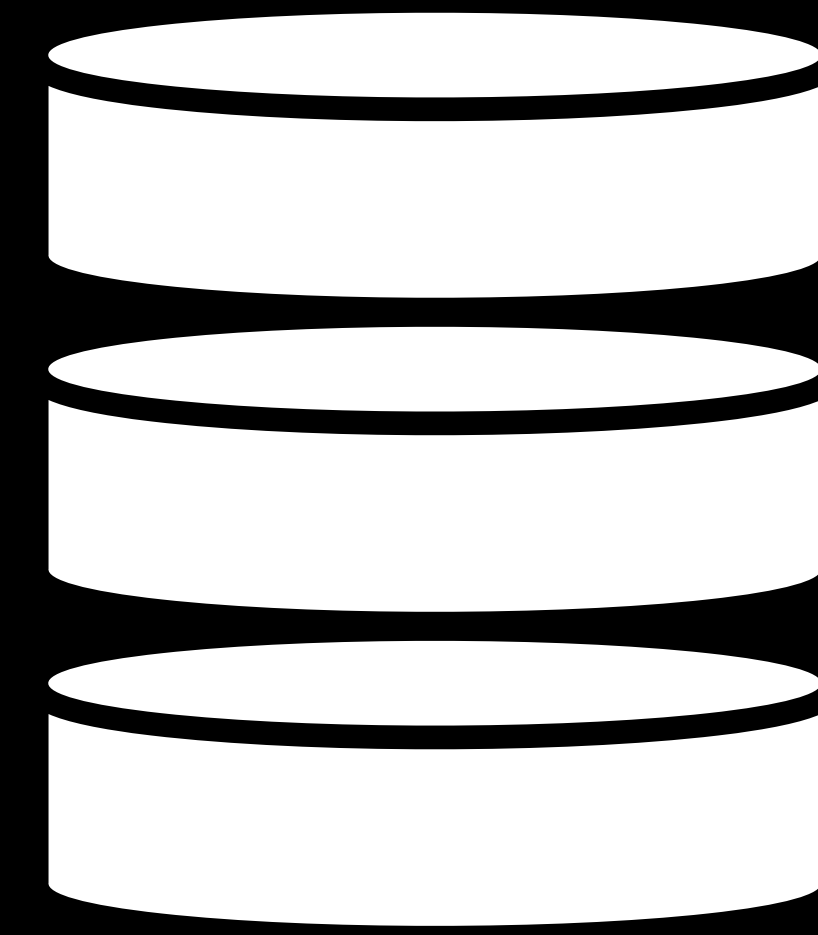
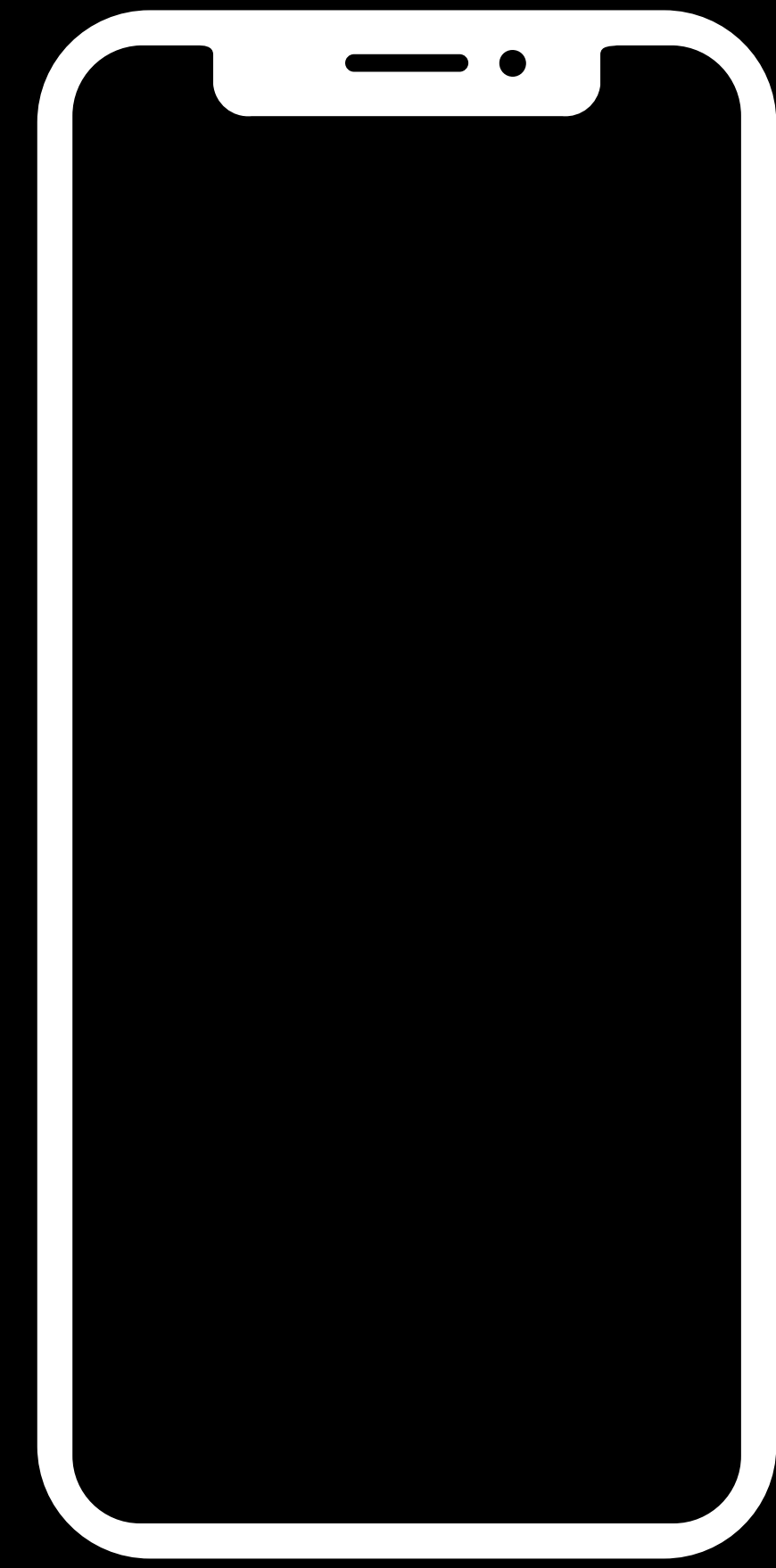
Scheduling



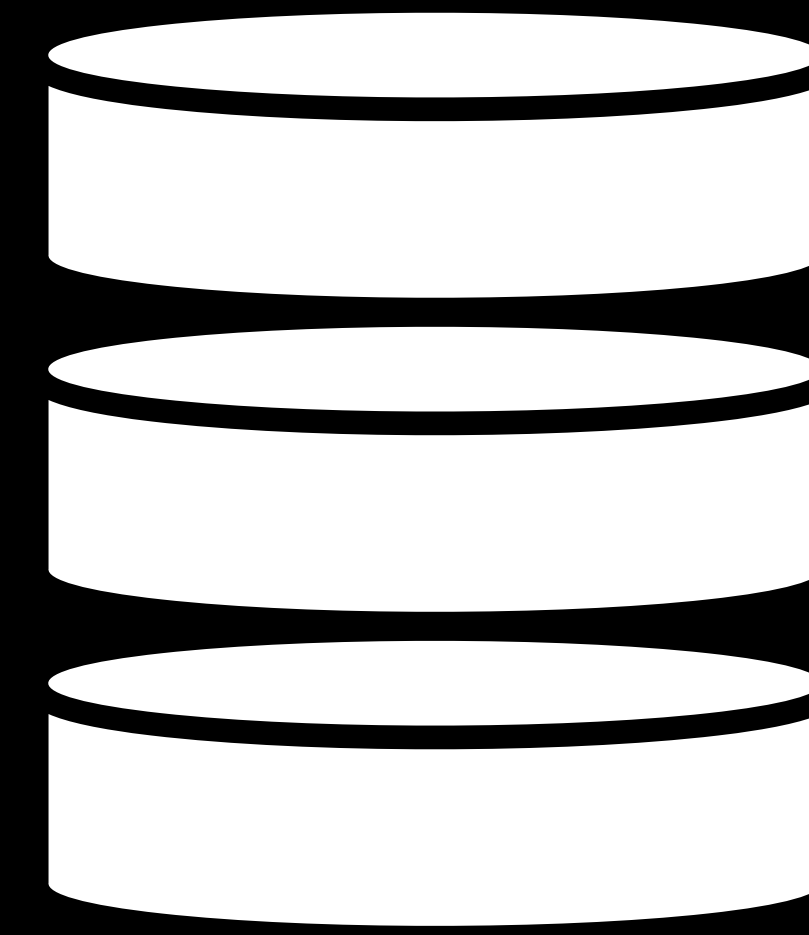
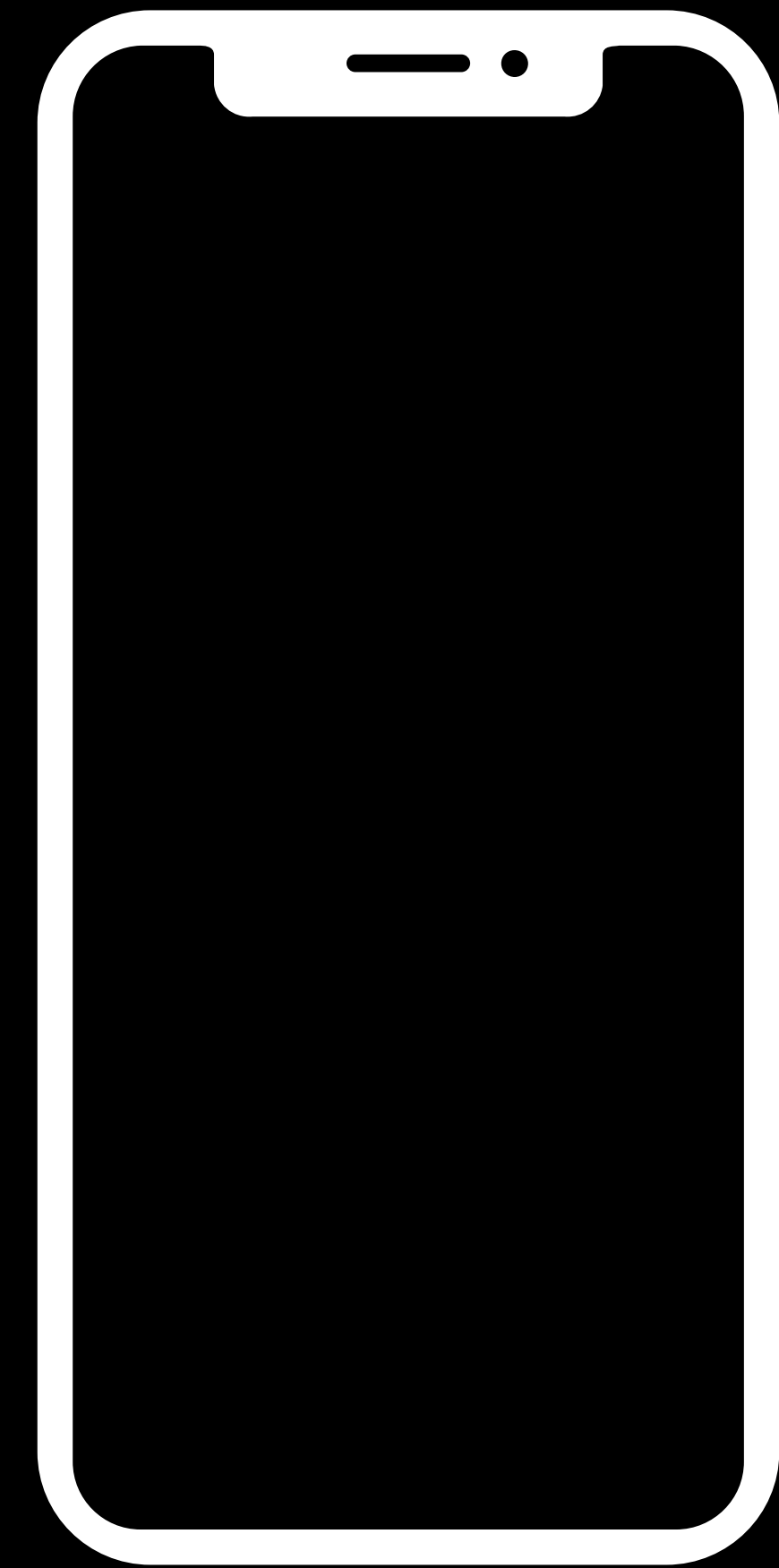
Scheduling



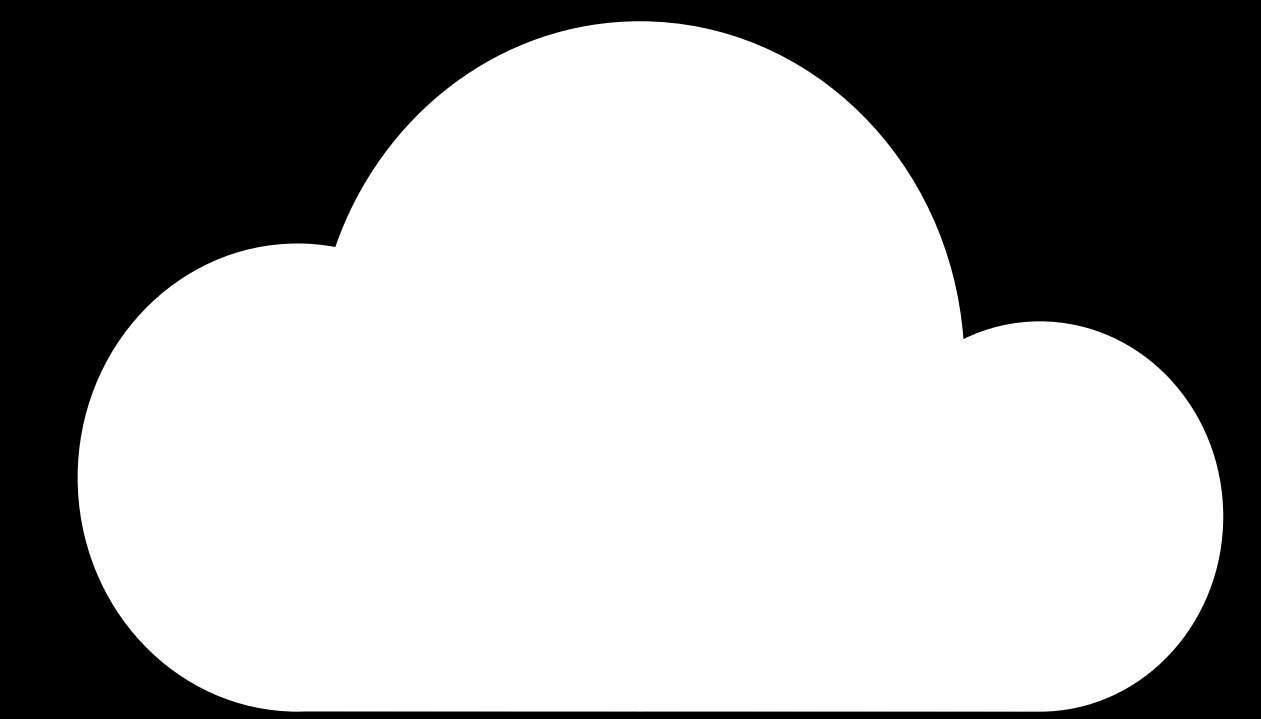
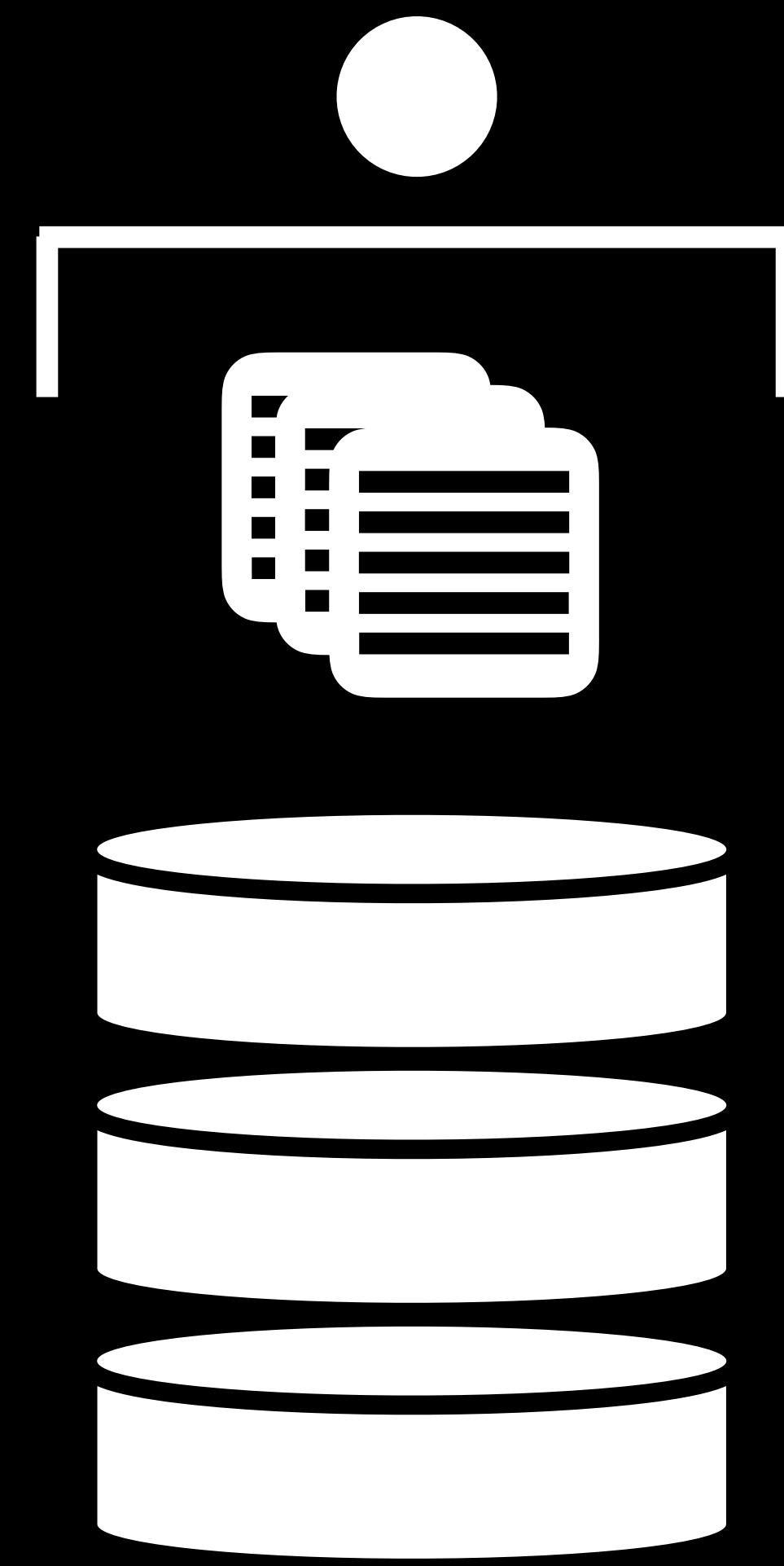
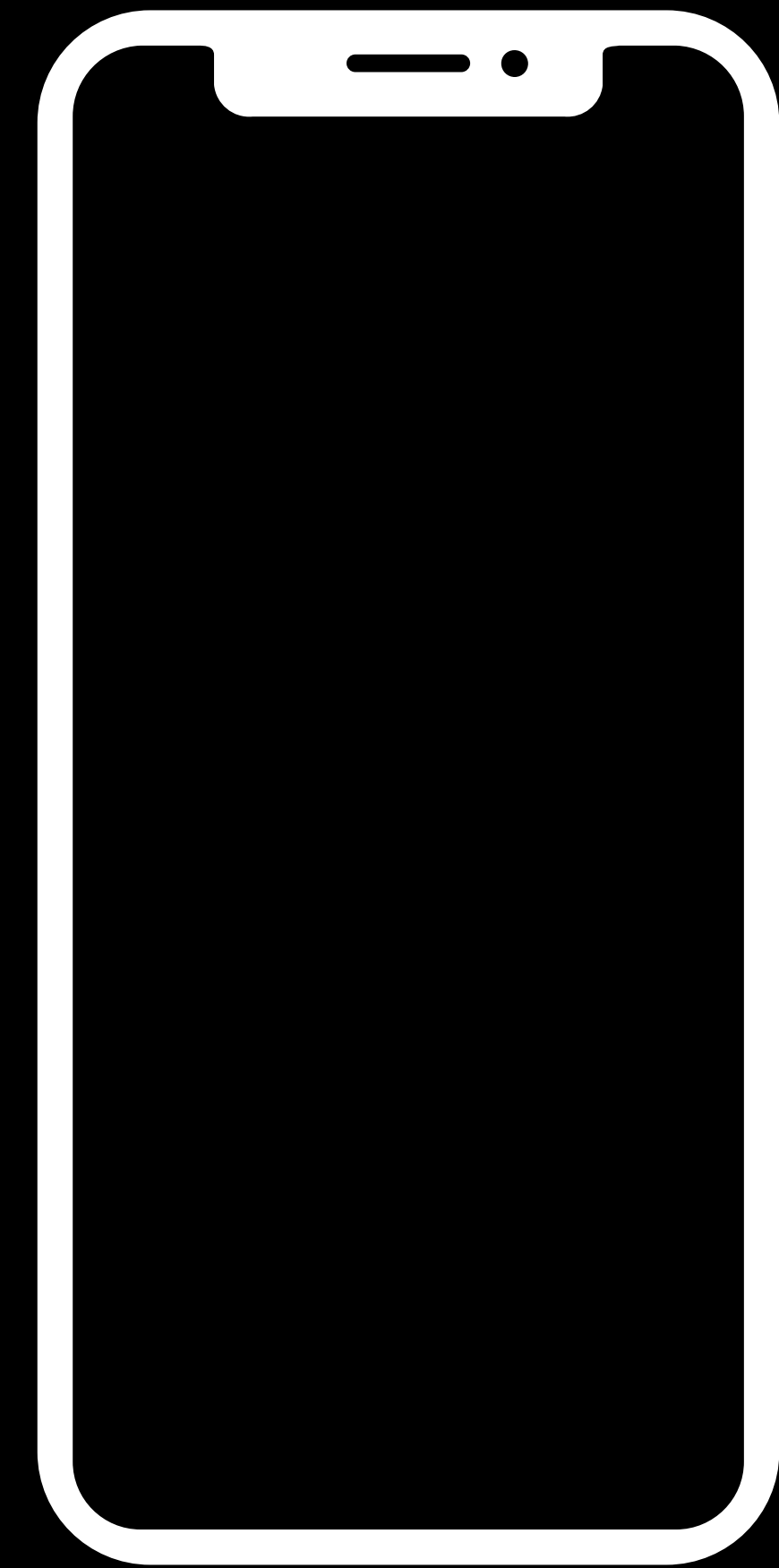
Scheduling



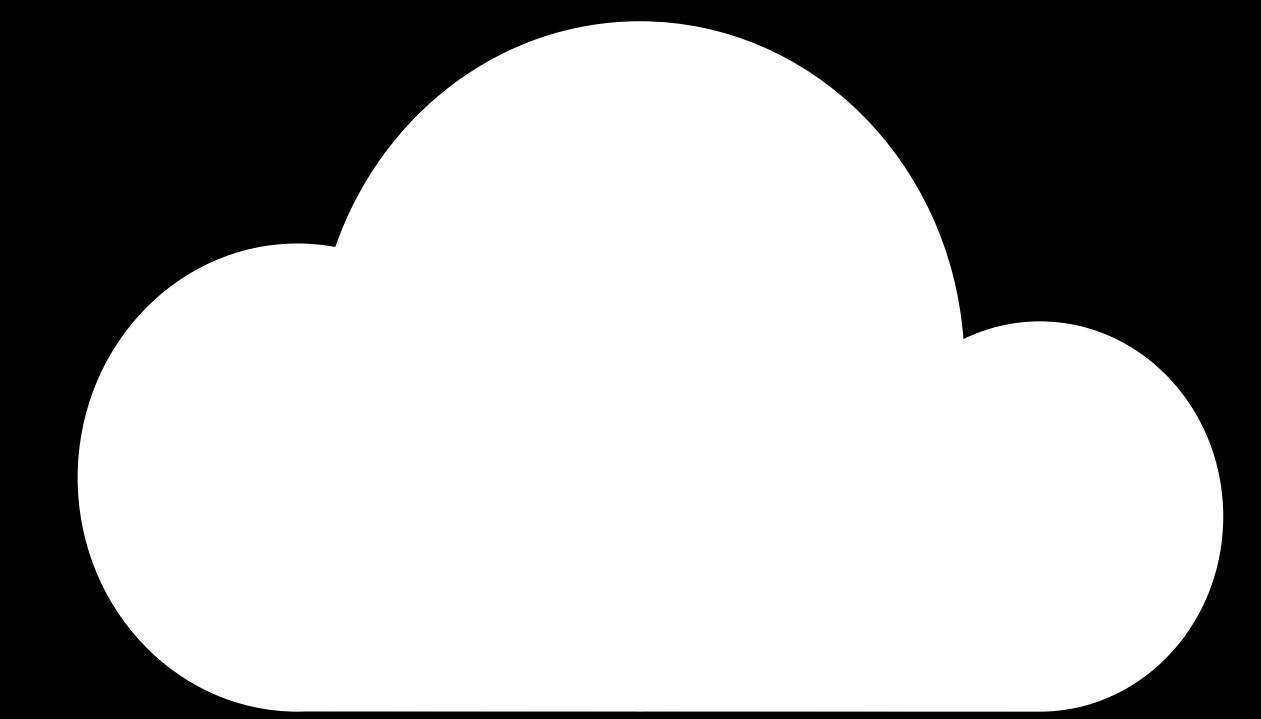
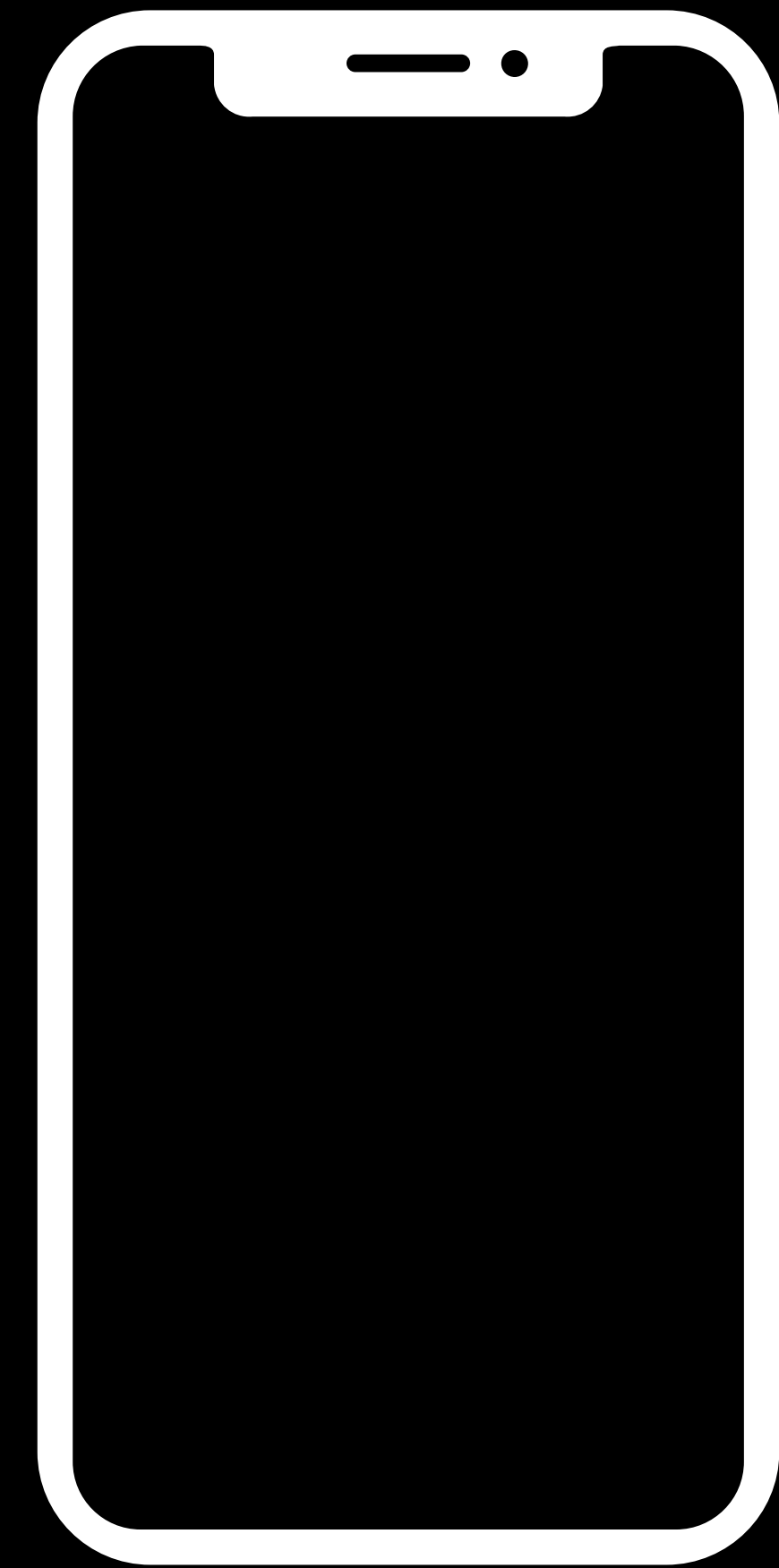
Scheduling



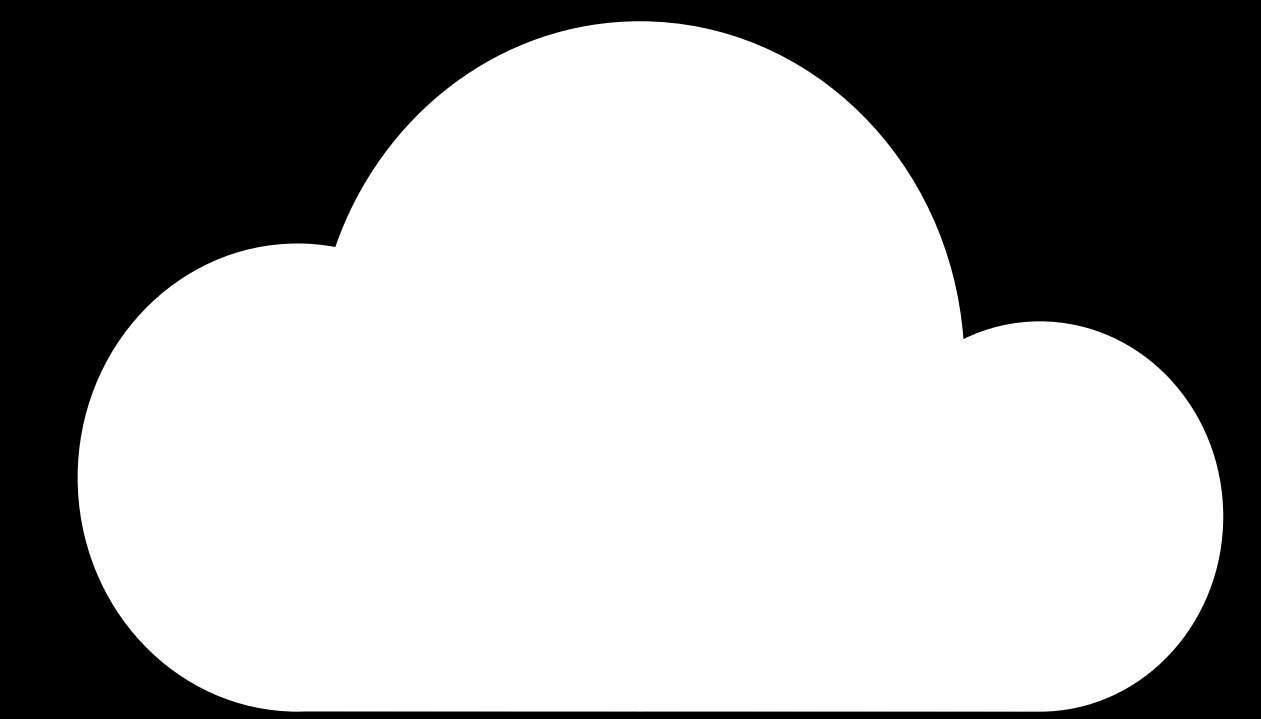
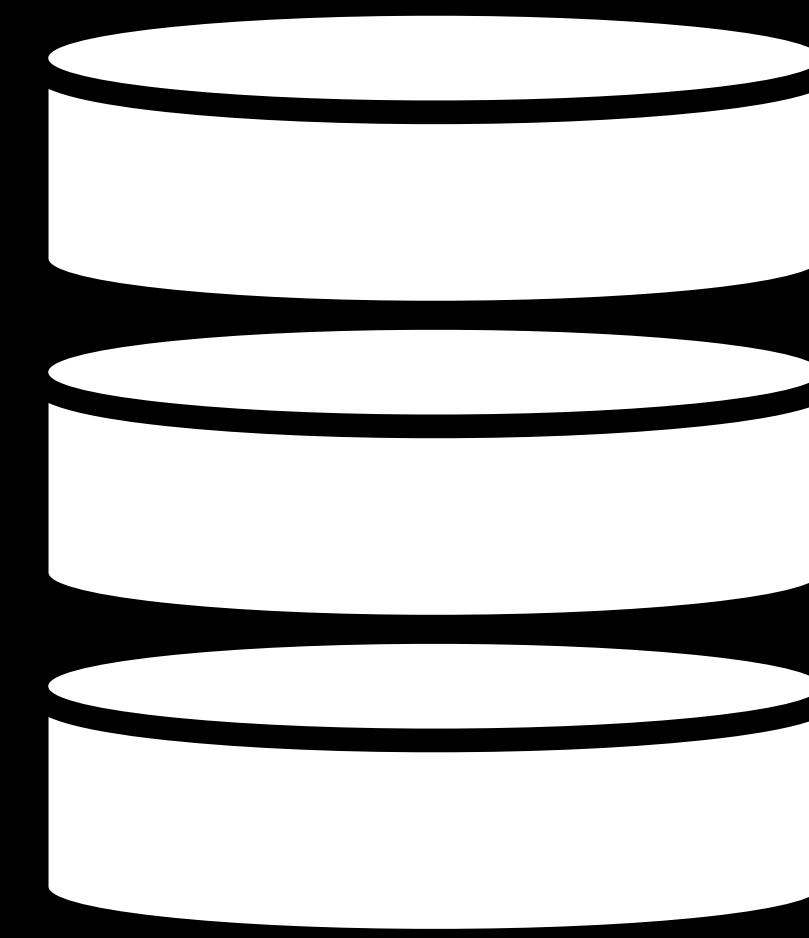
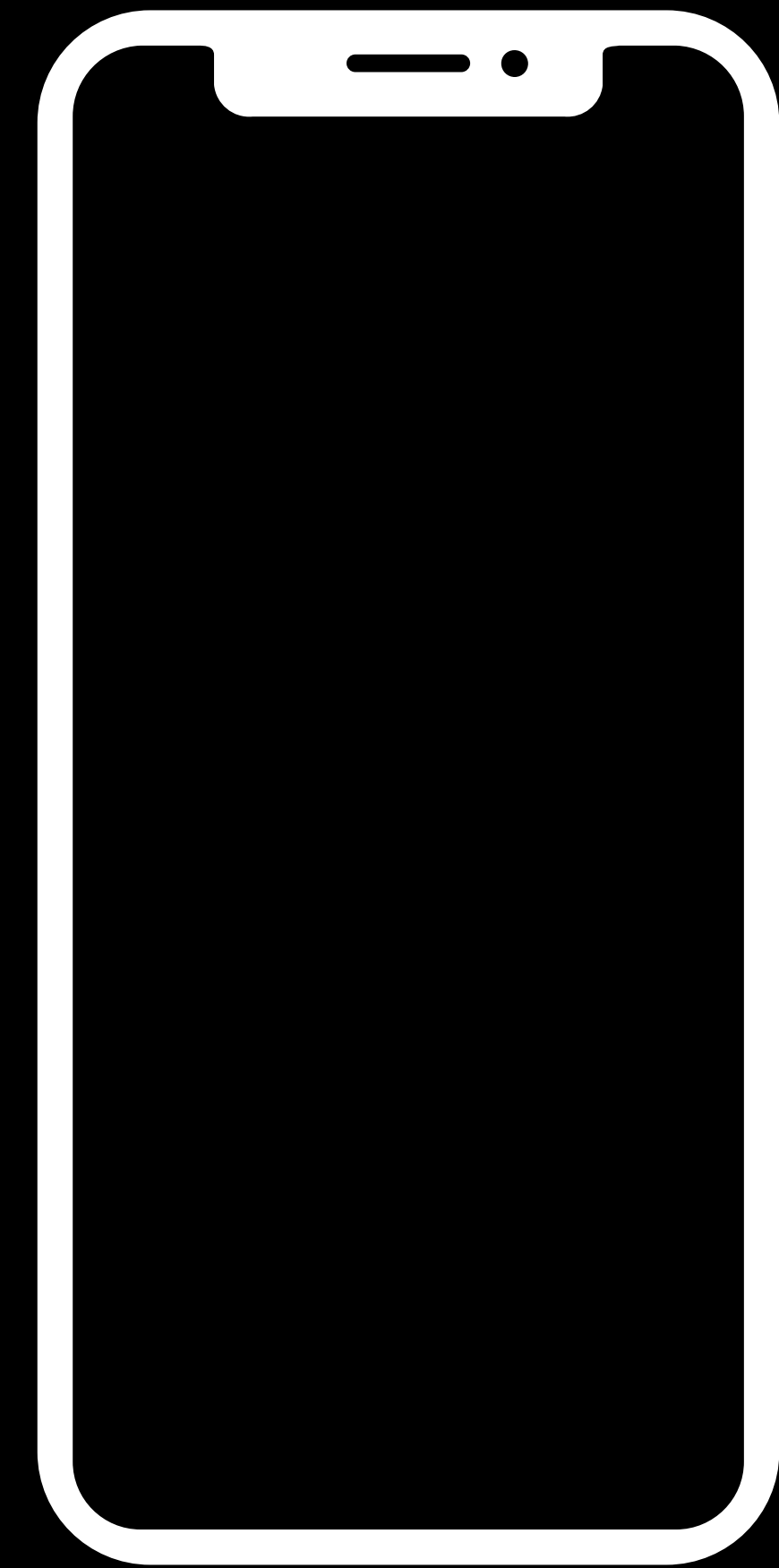
Scheduling



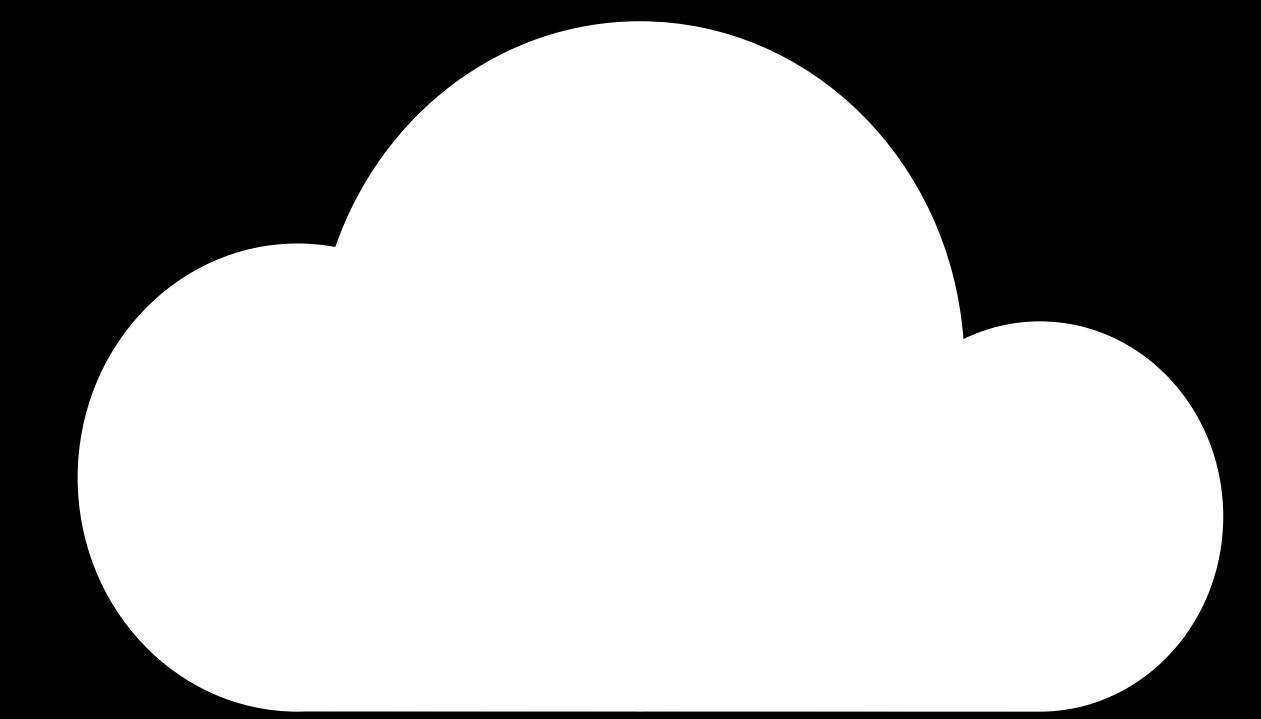
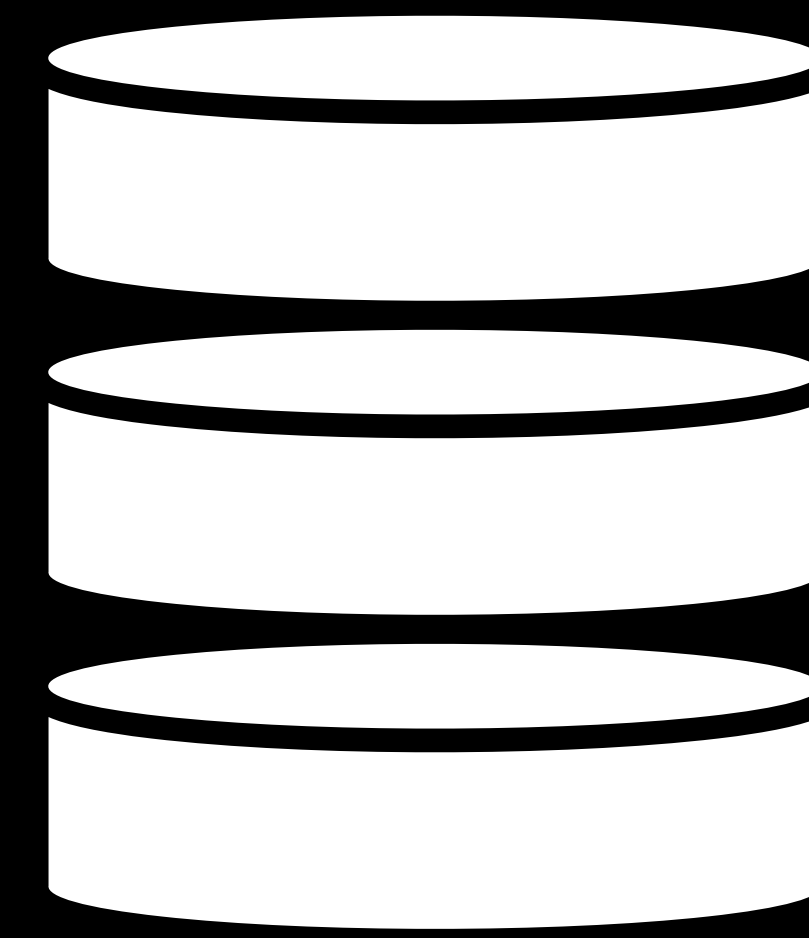
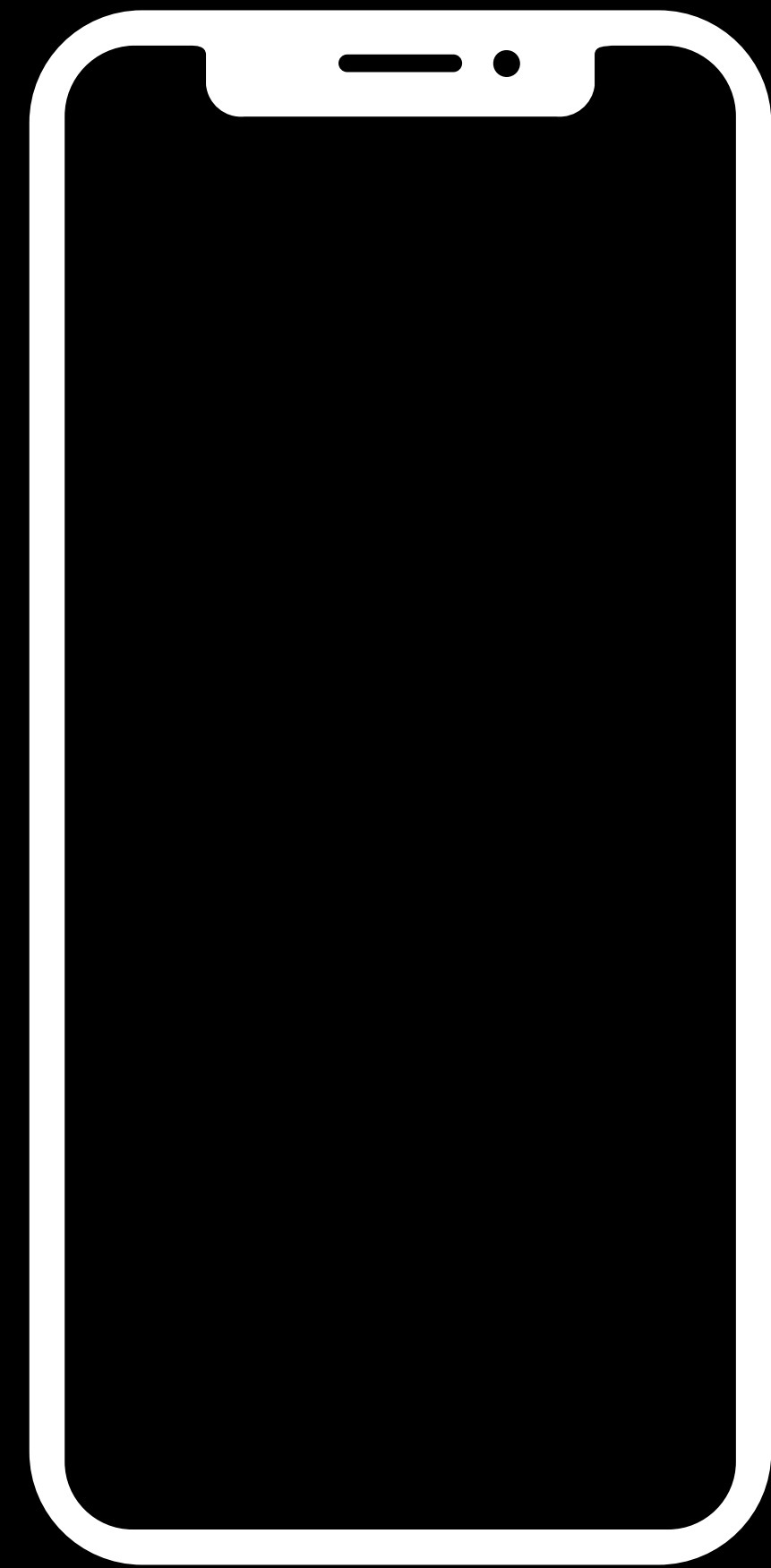
Scheduling



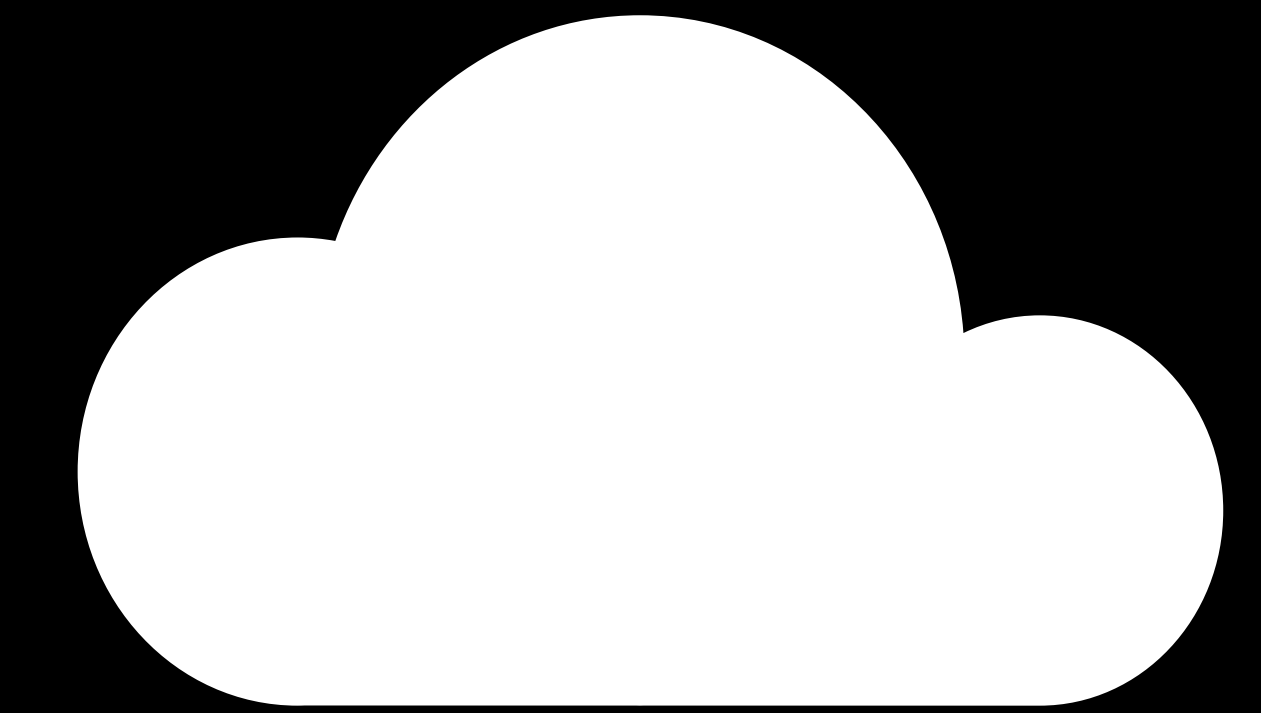
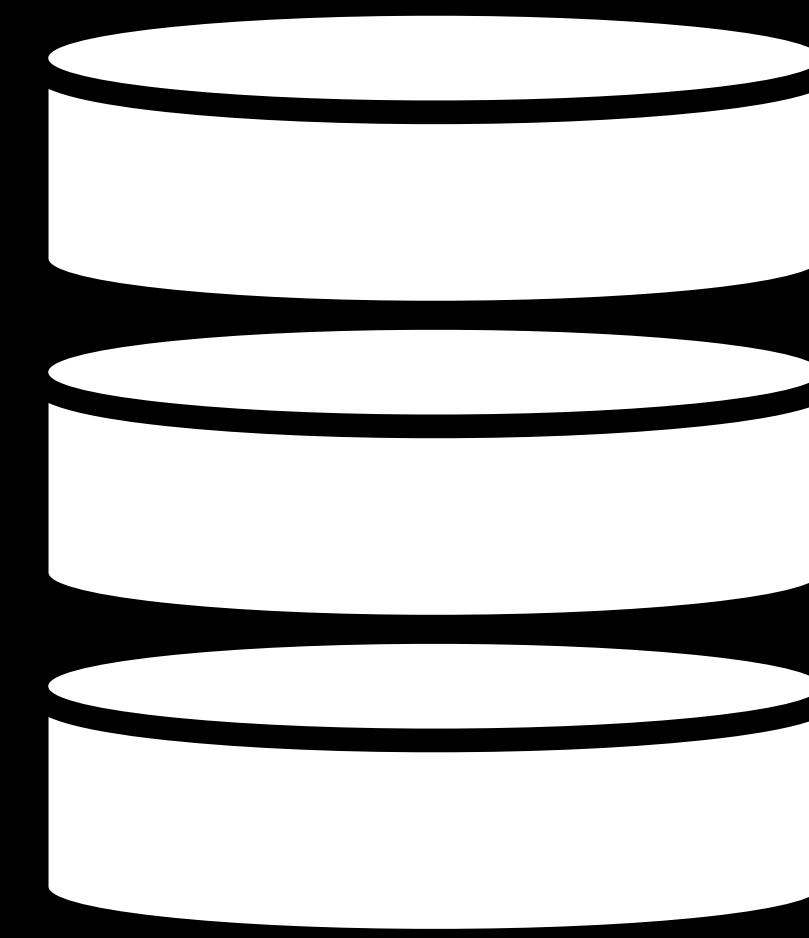
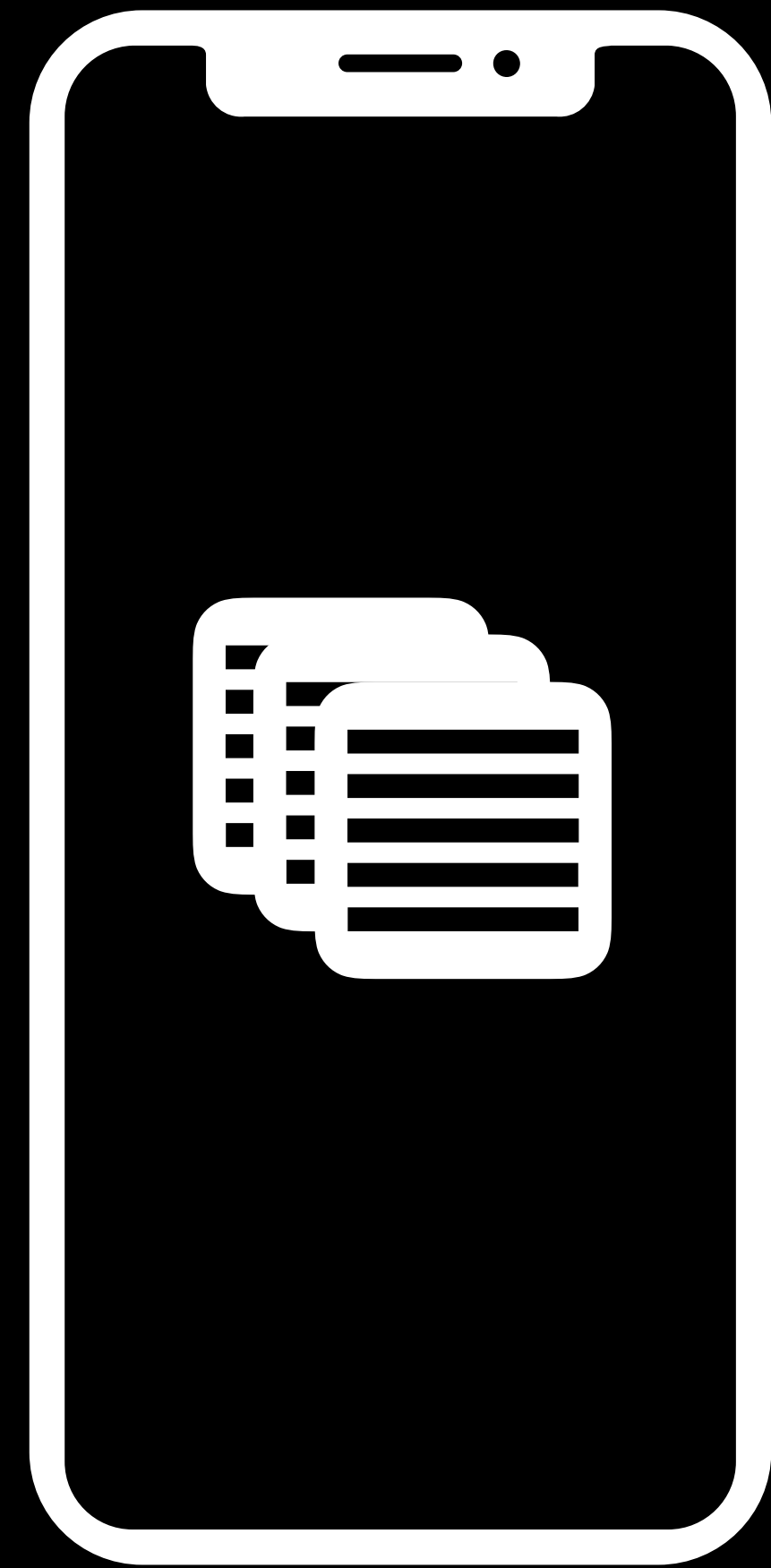
Scheduling



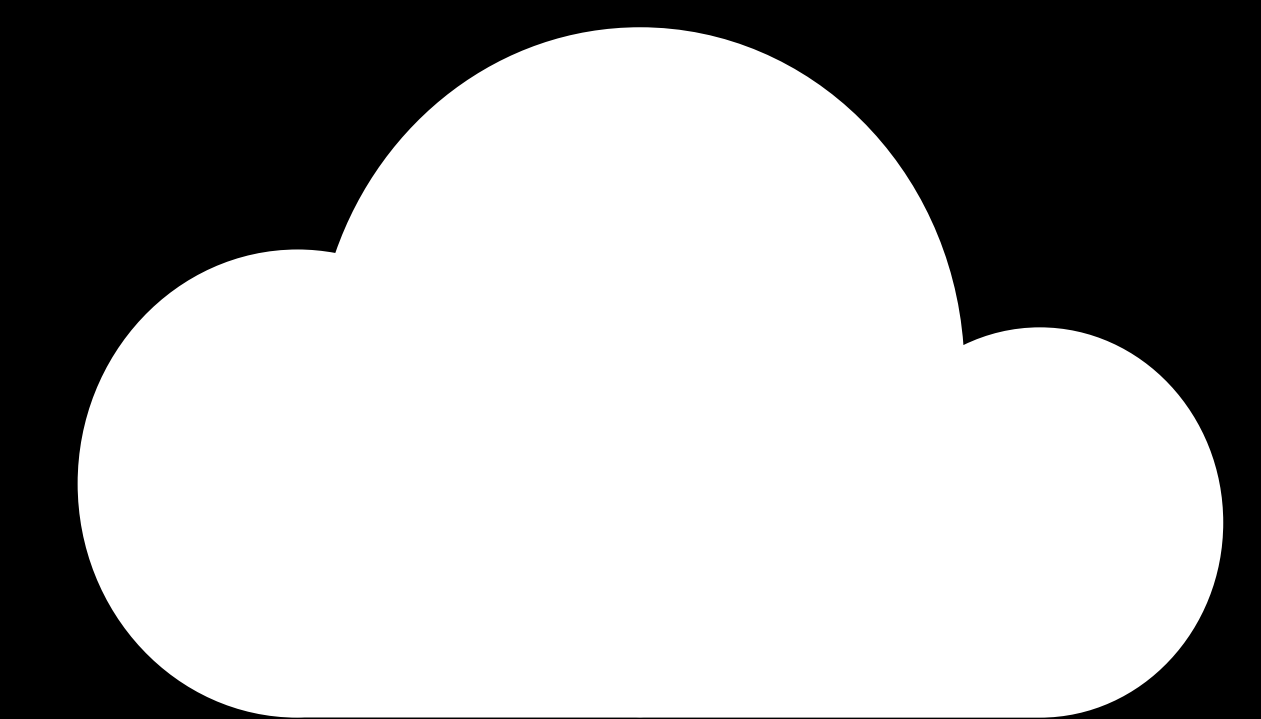
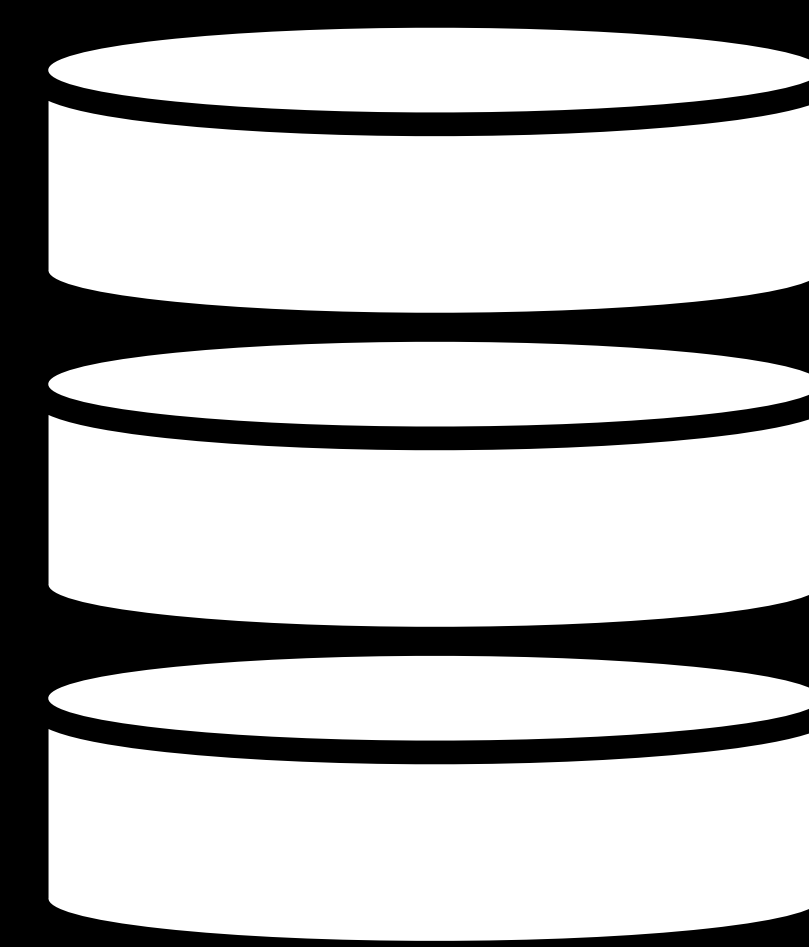
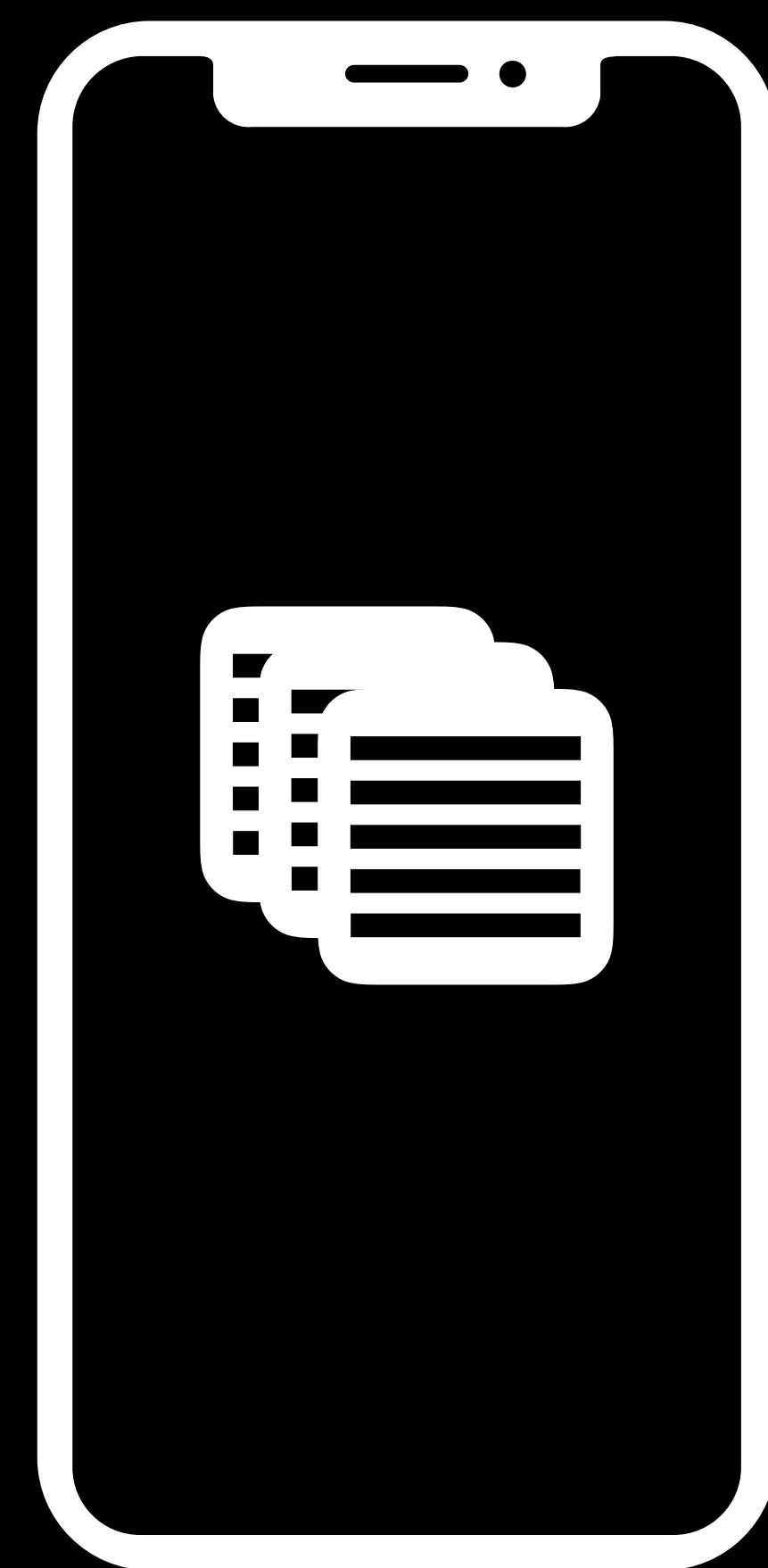
Scheduling



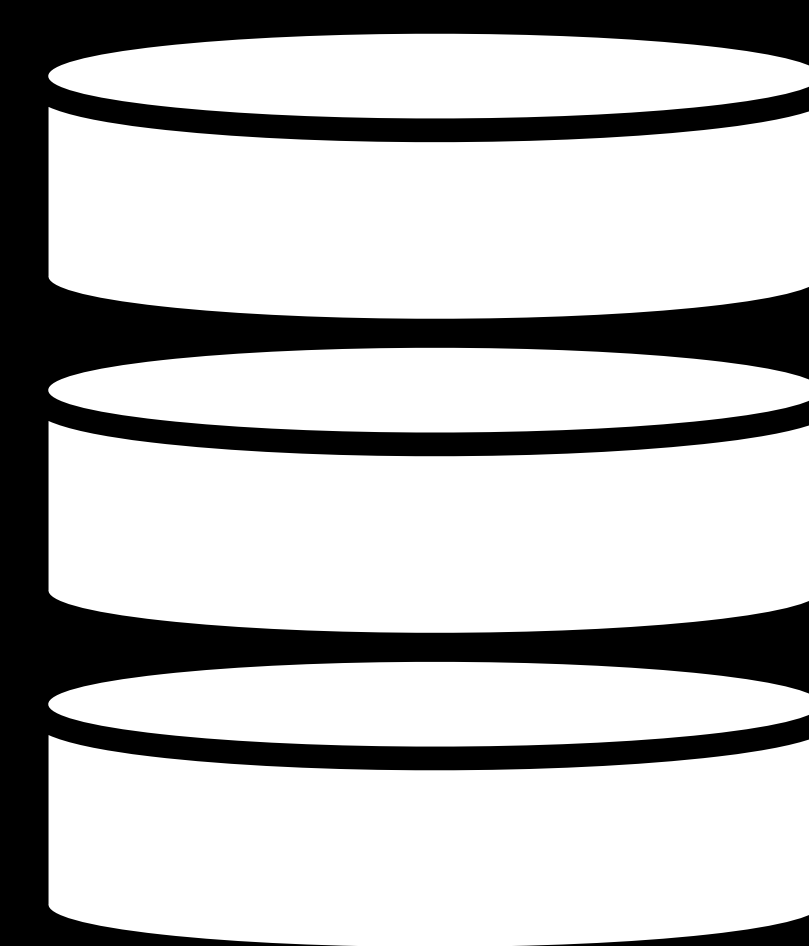
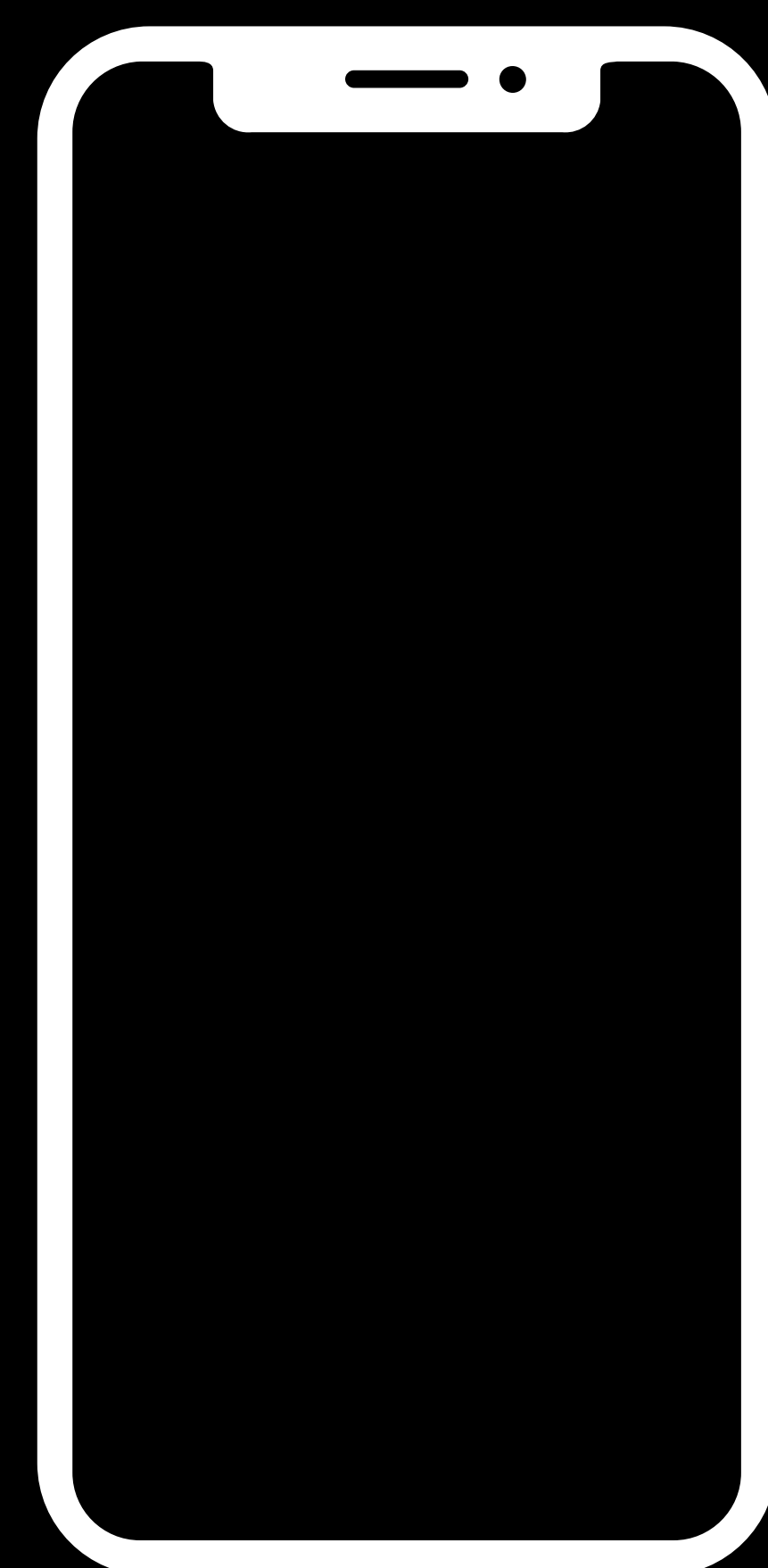
Scheduling



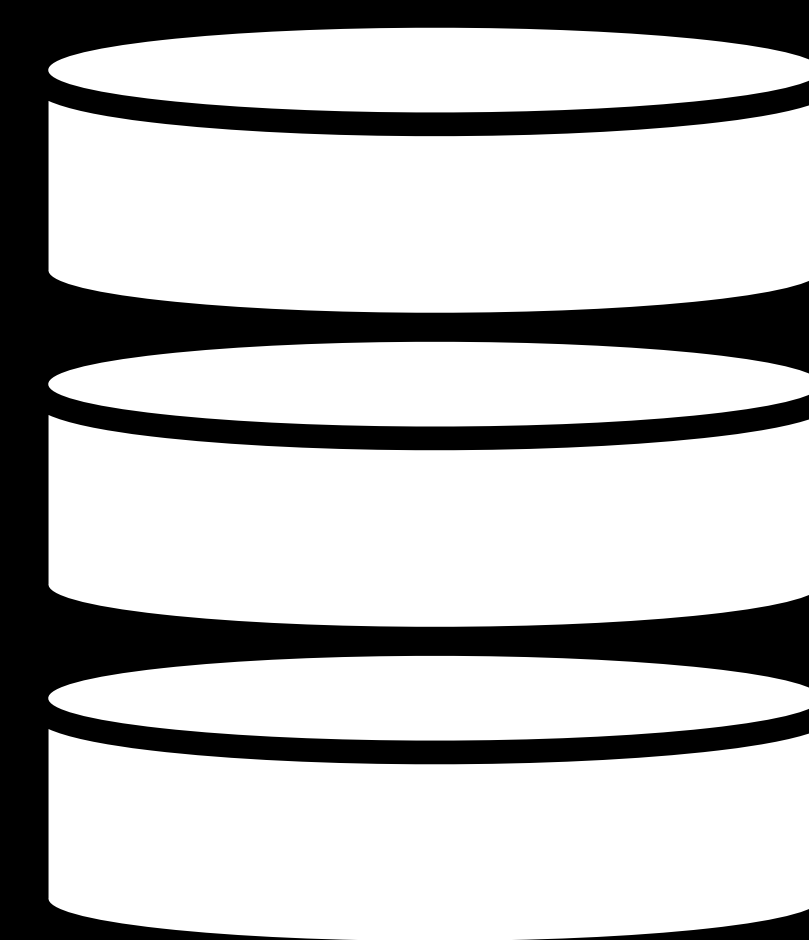
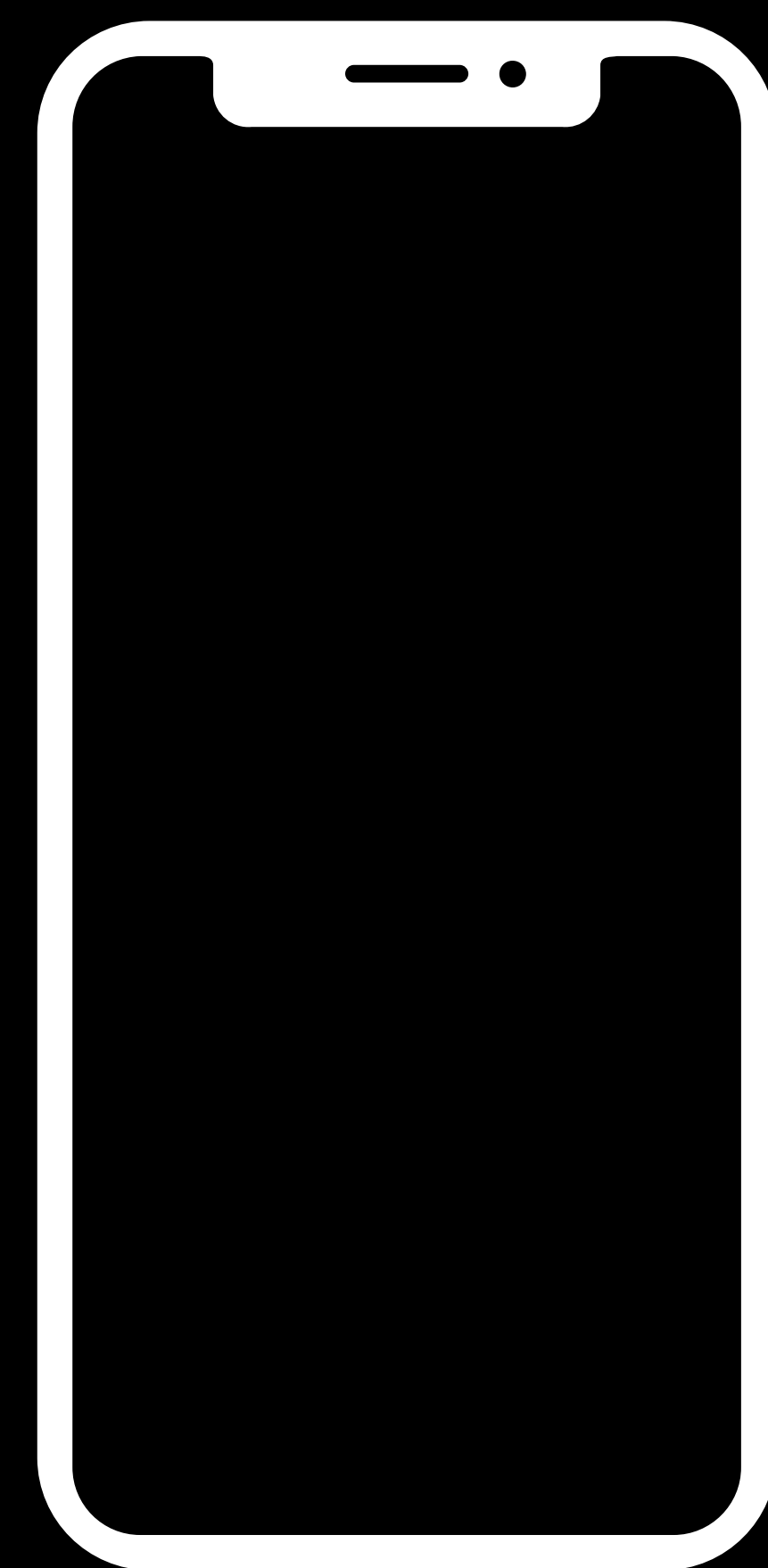
NSManagedObject to CKRecord



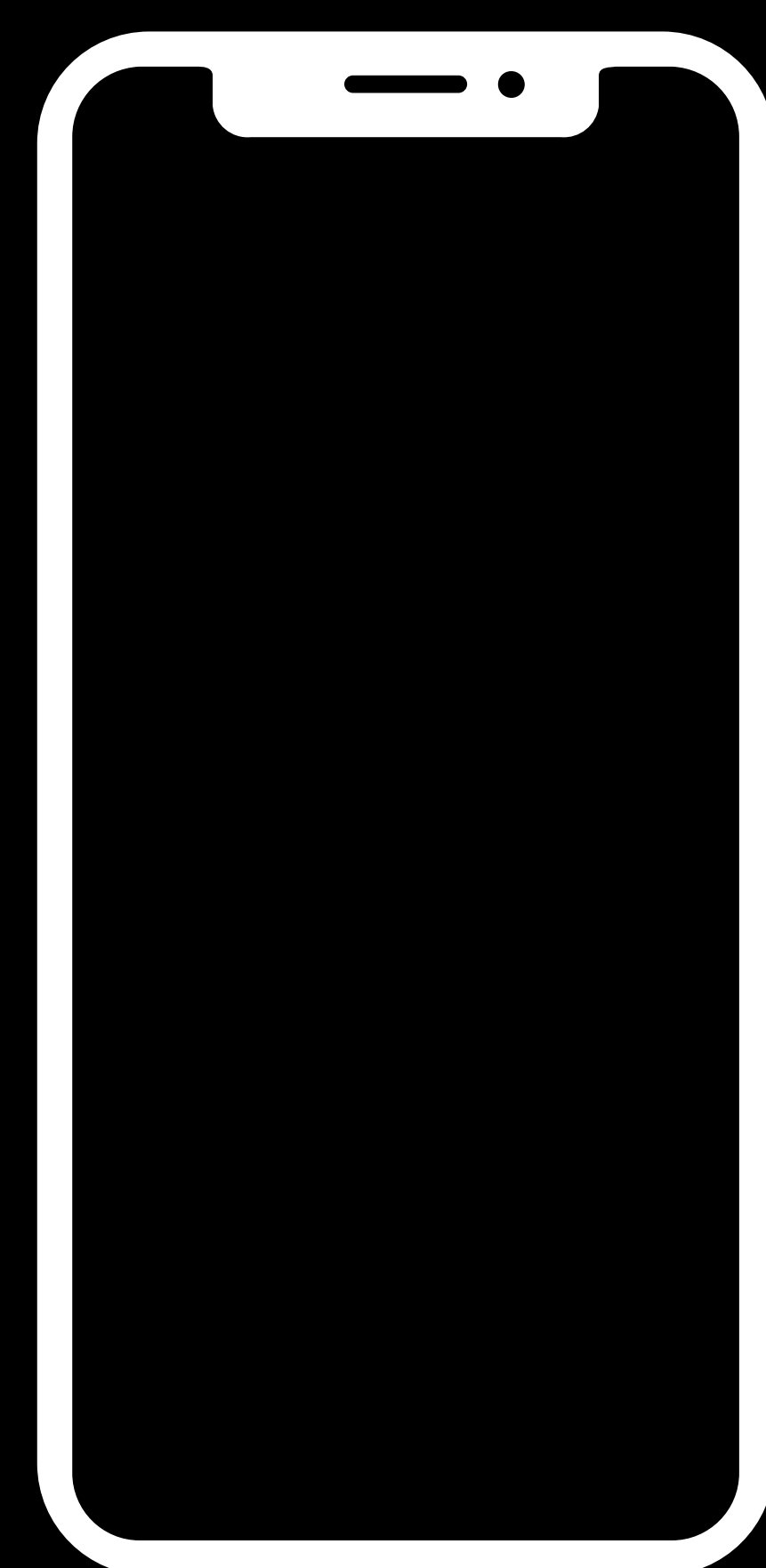
NSManagedObject to CKRecord



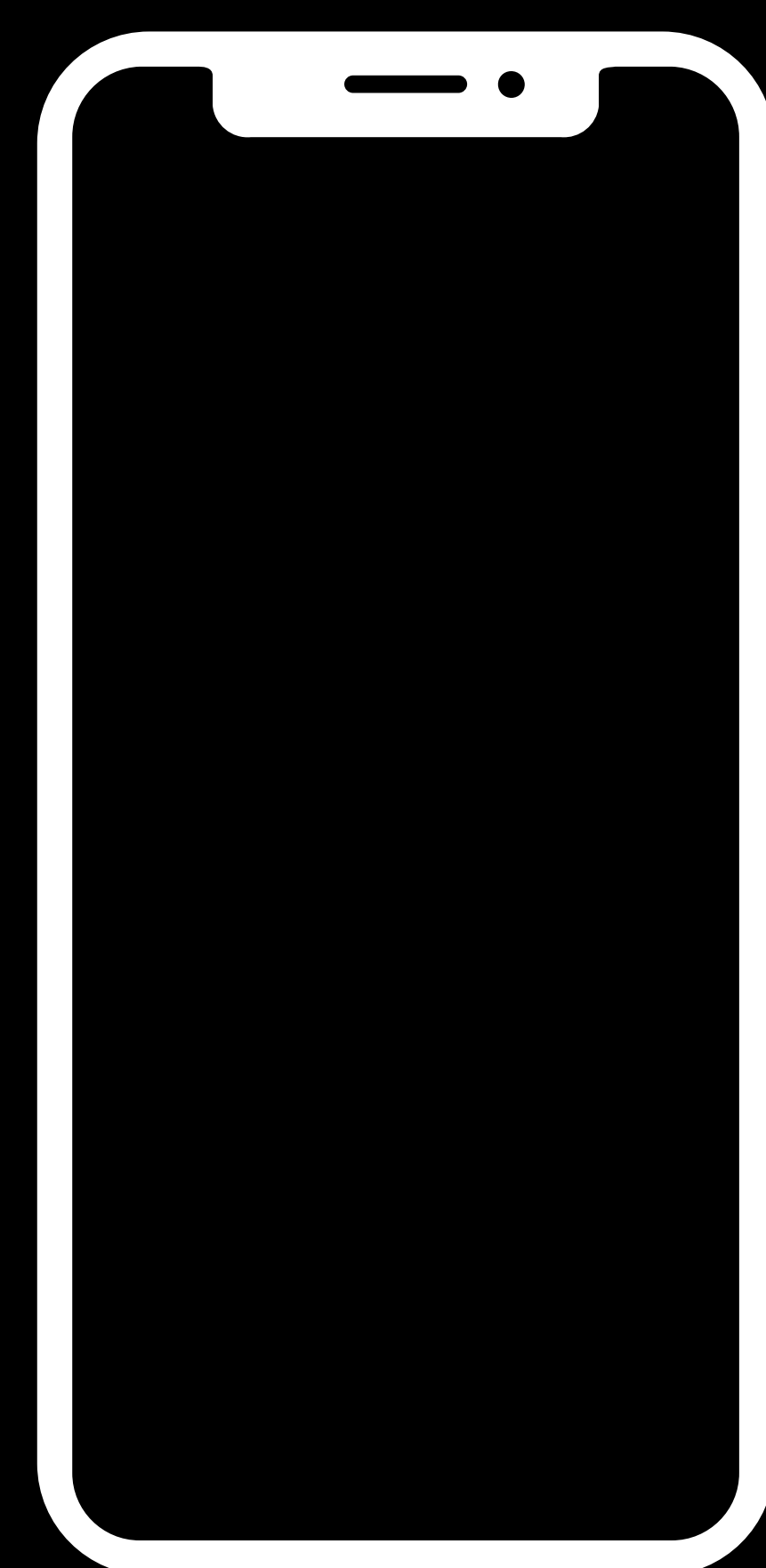
NSManagedObject to CKRecord



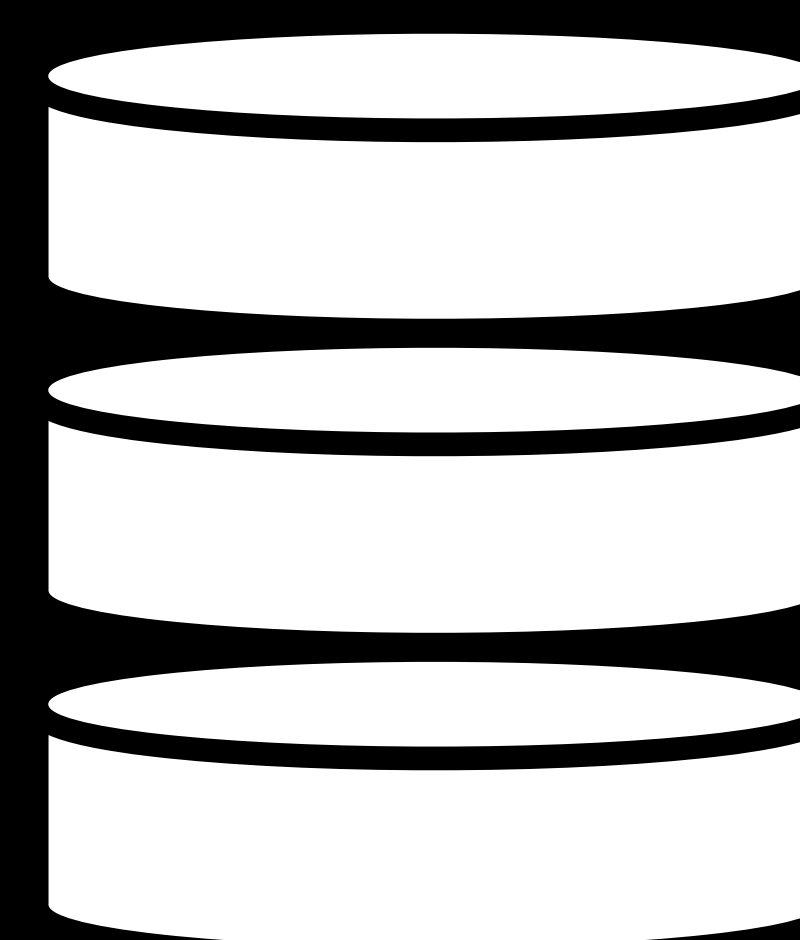
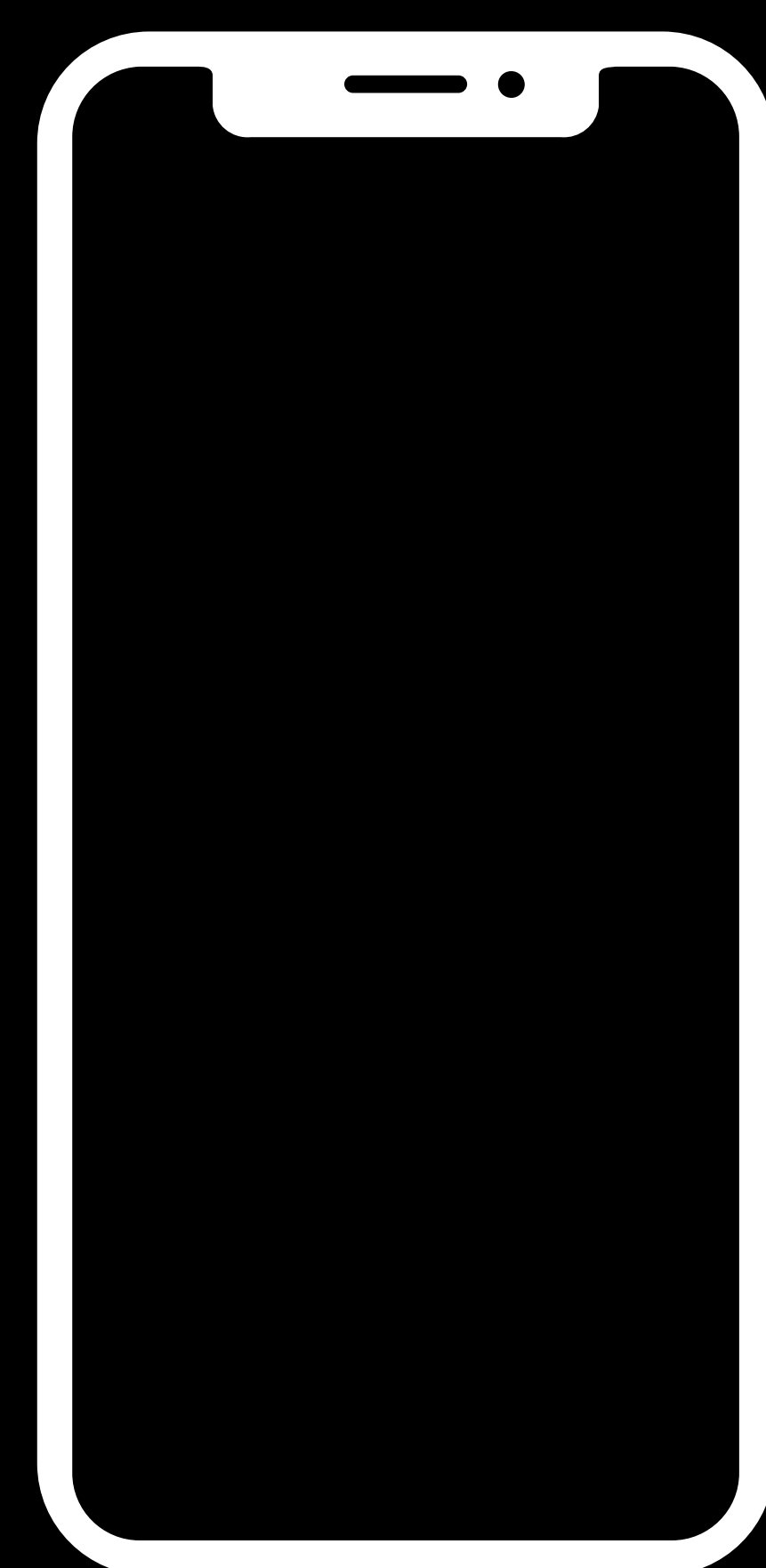
NSManagedObject to CKRecord



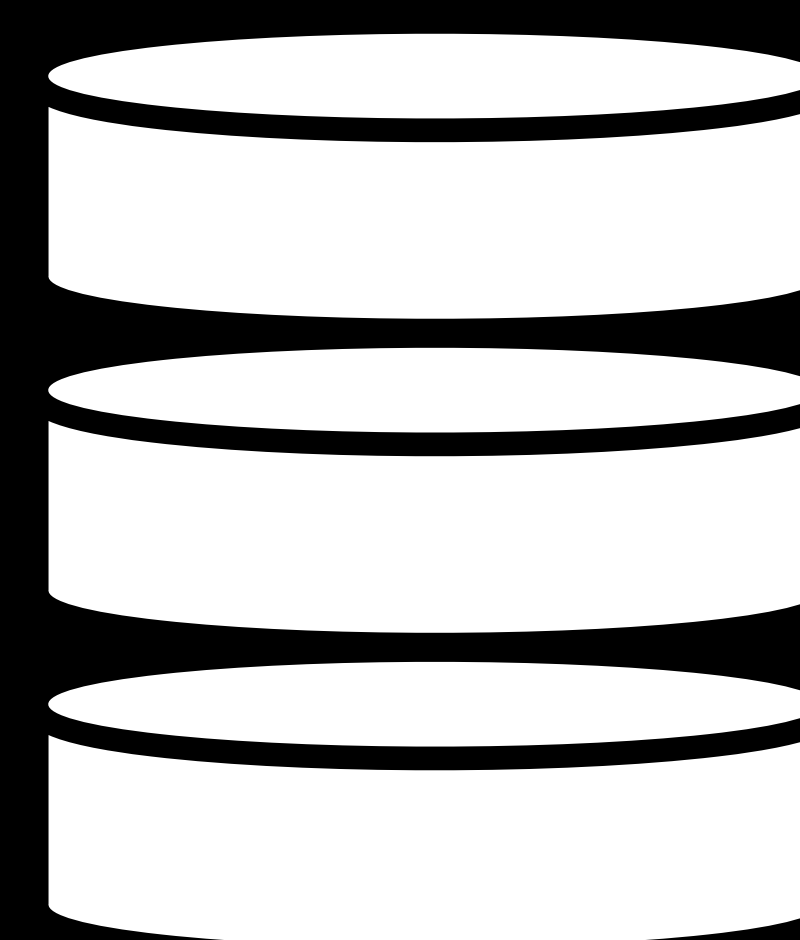
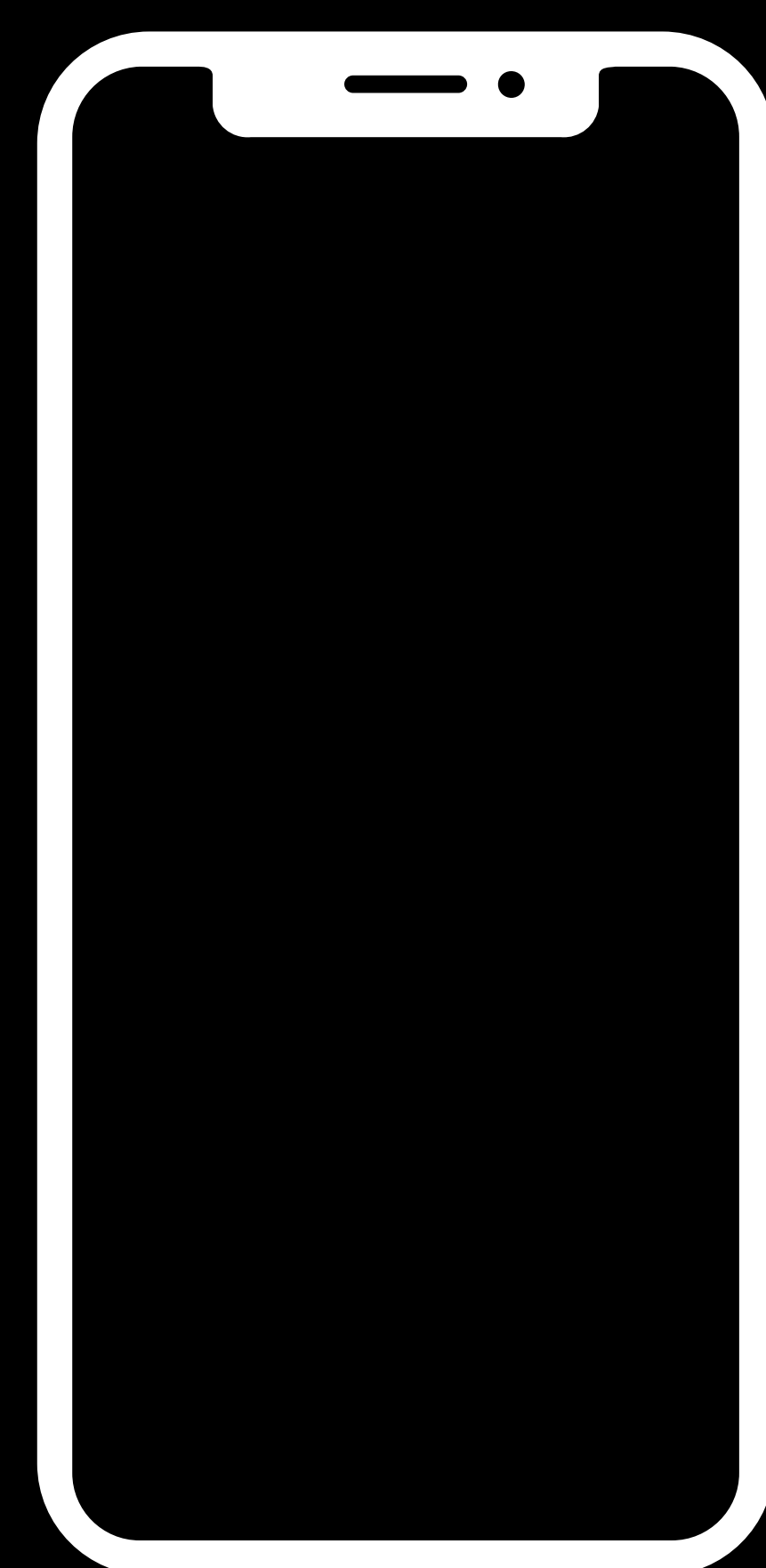
NSManagedObject to CKRecord



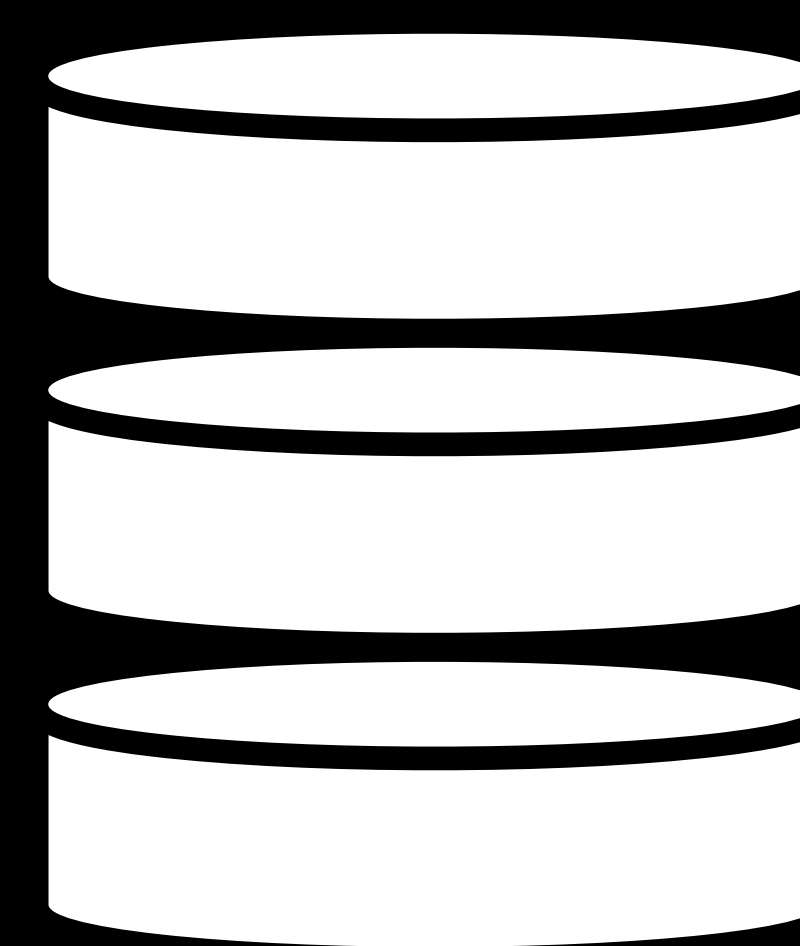
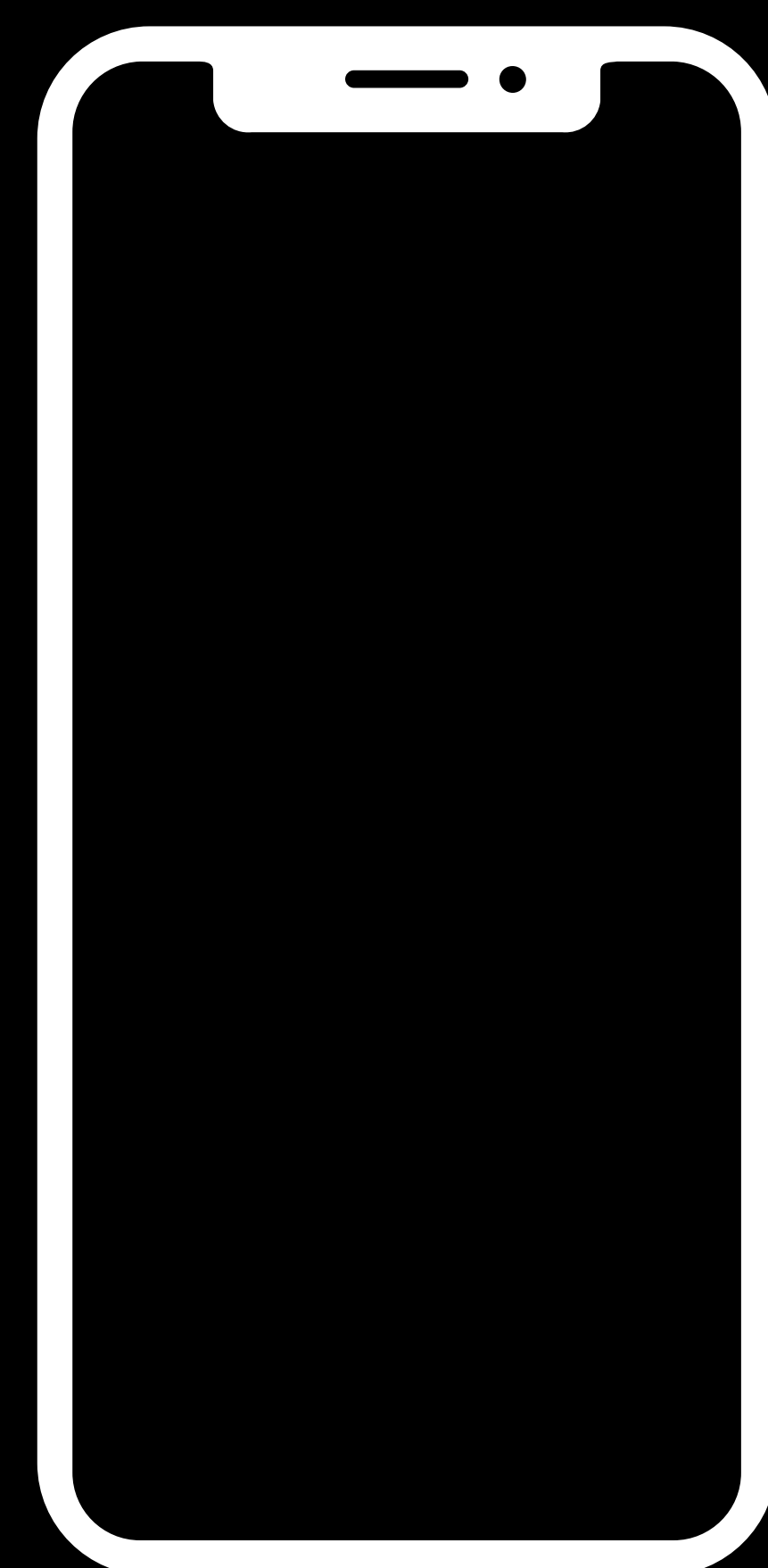
NSManagedObject to CKRecord



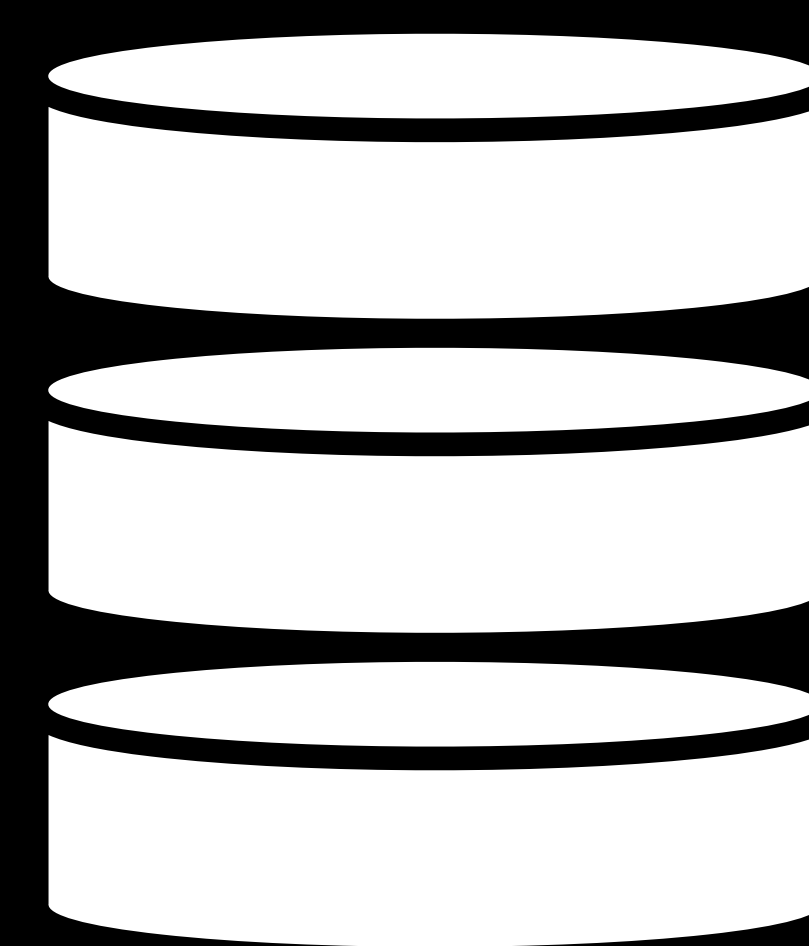
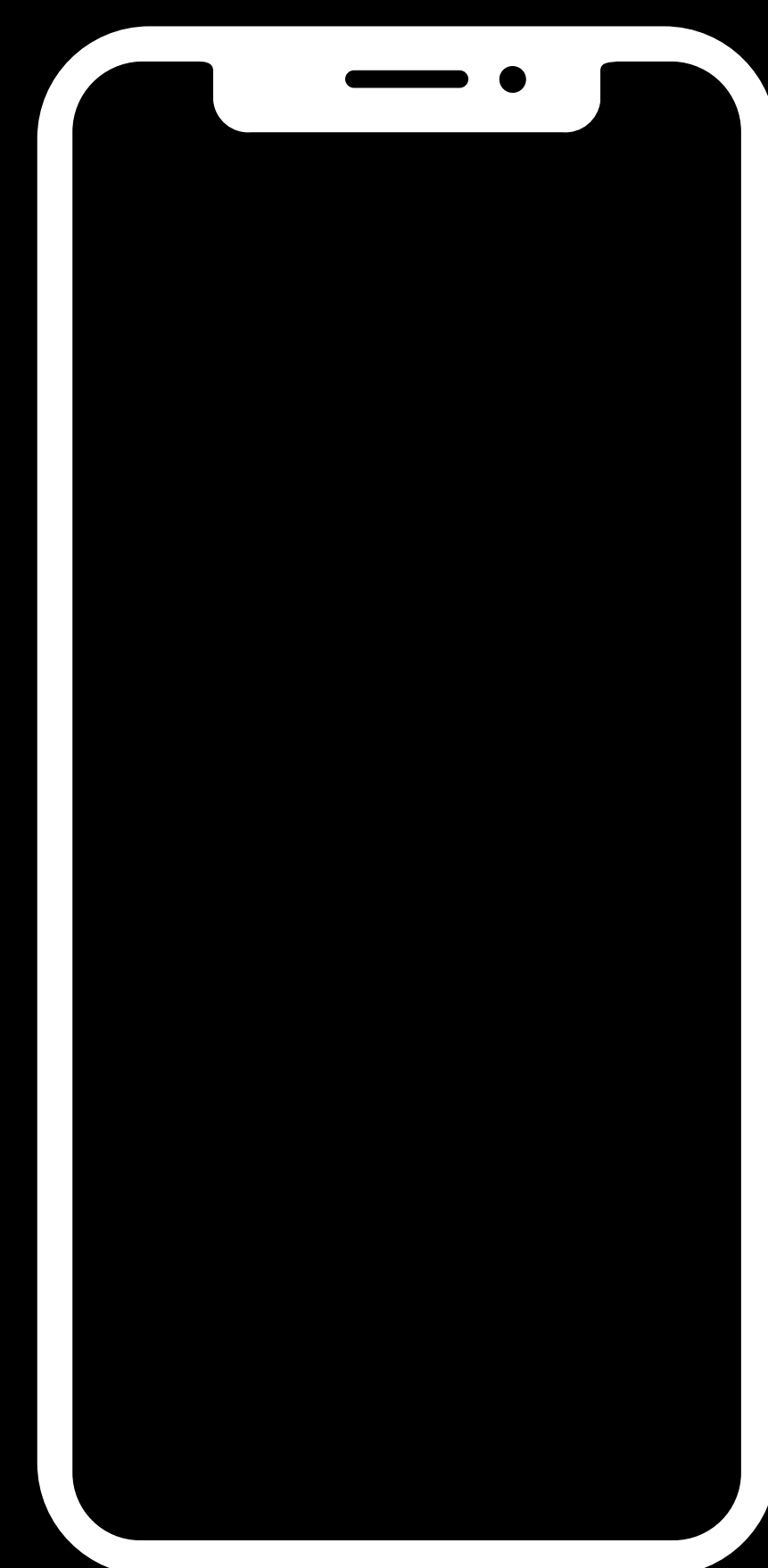
NSManagedObject to CKRecord



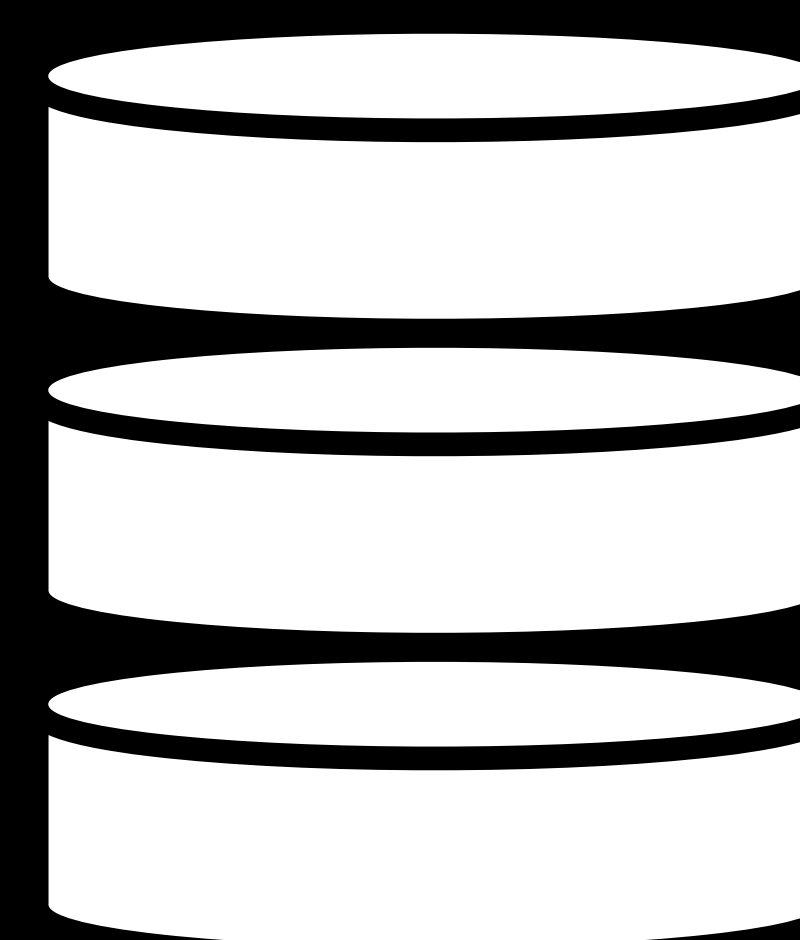
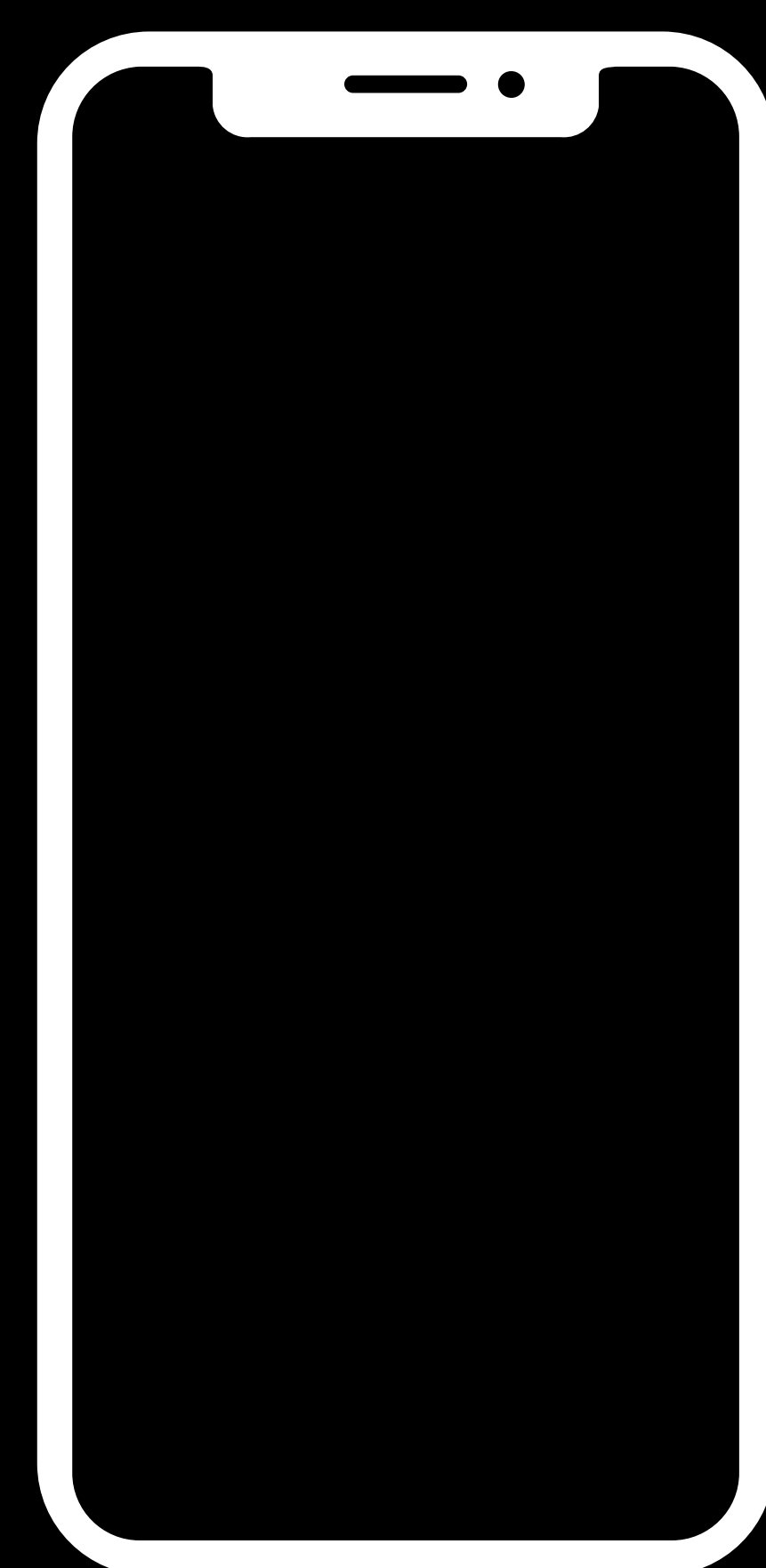
NSManagedObject to CKRecord



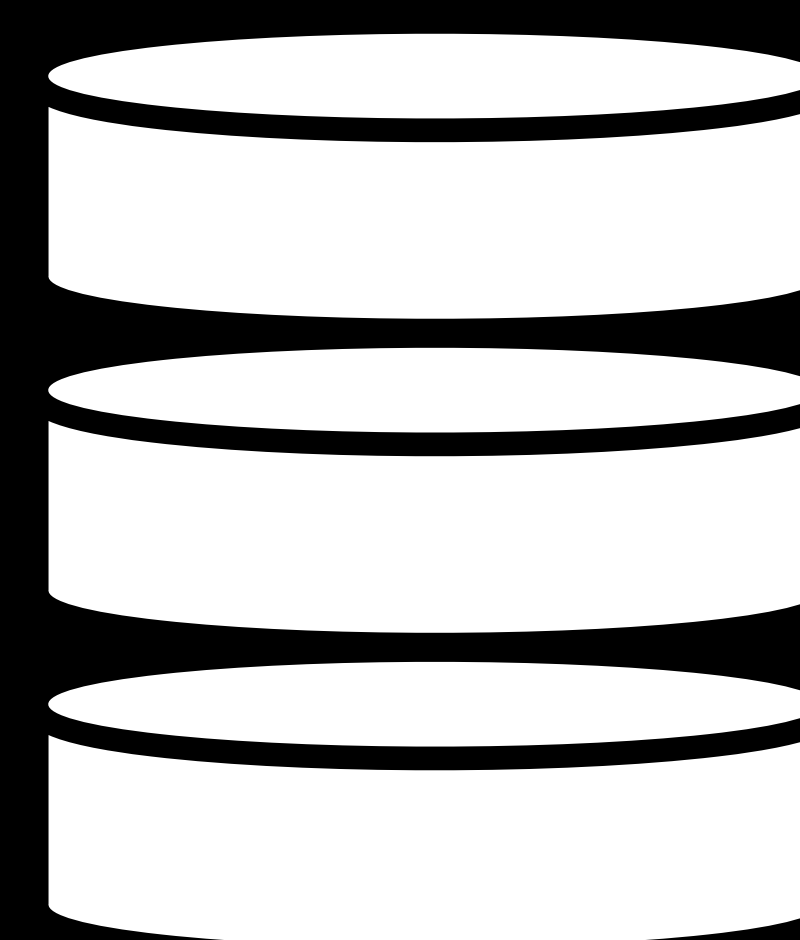
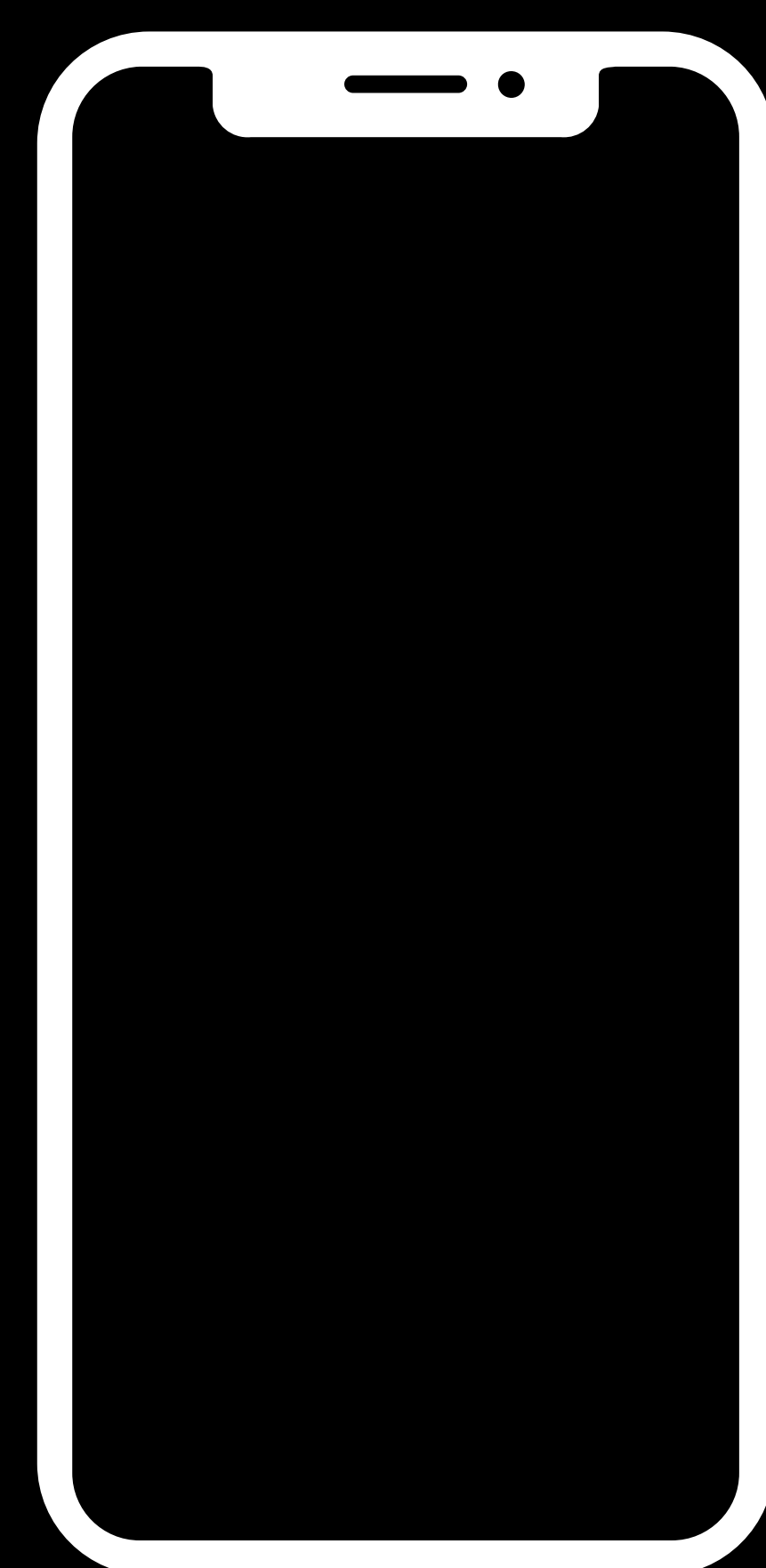
NSManagedObject to CKRecord



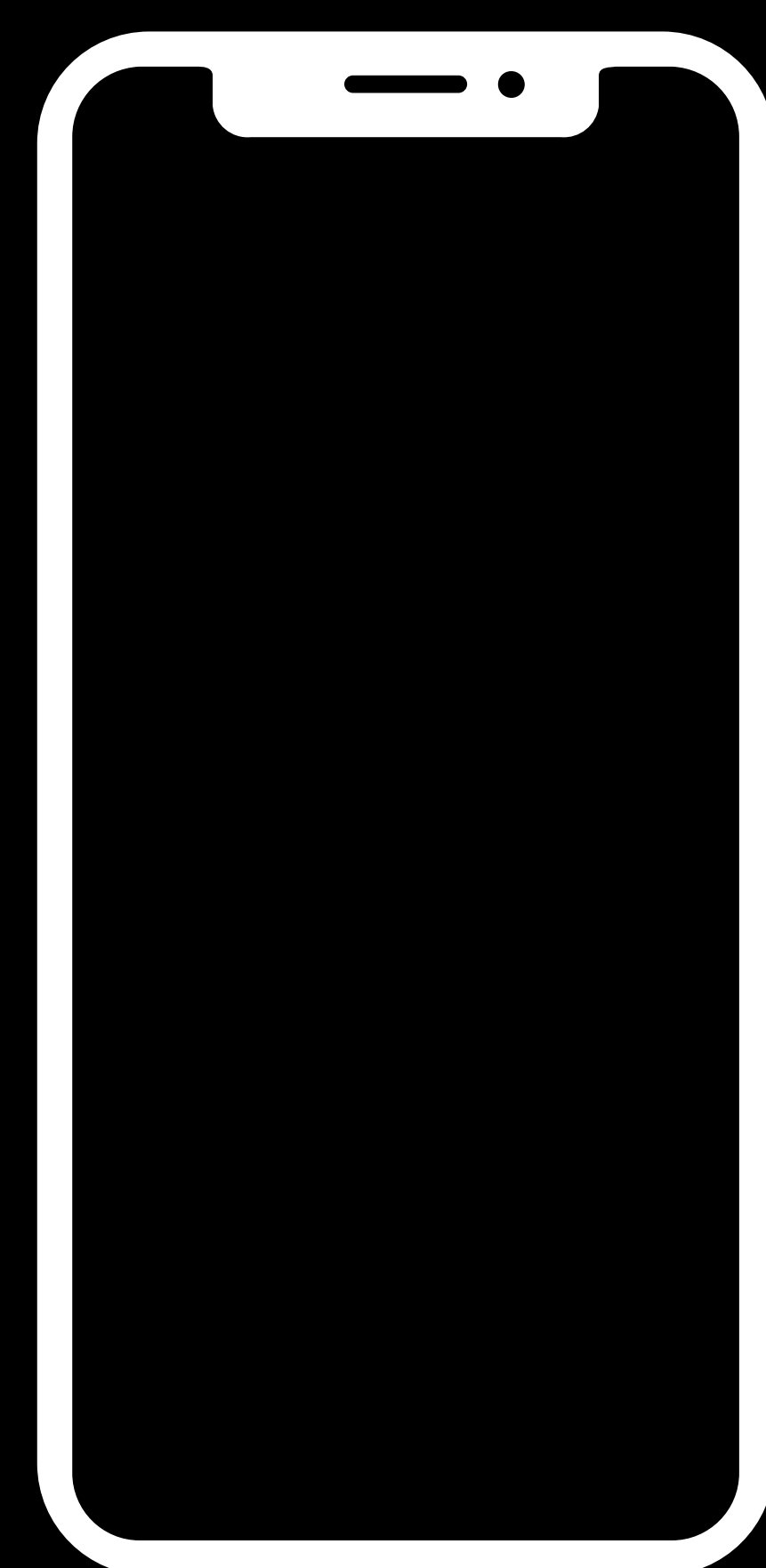
NSManagedObject to CKRecord



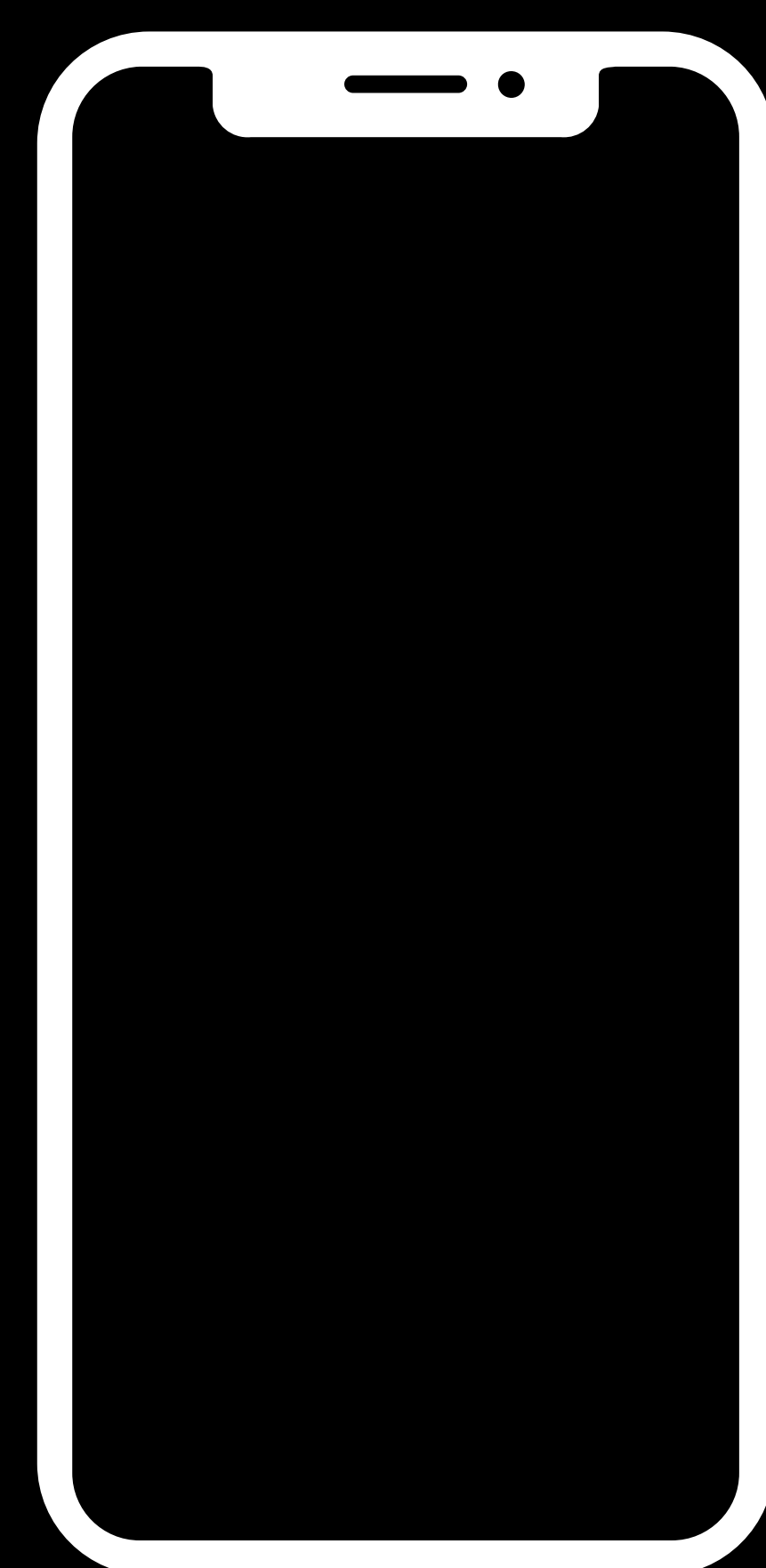
NSManagedObject to CKRecord



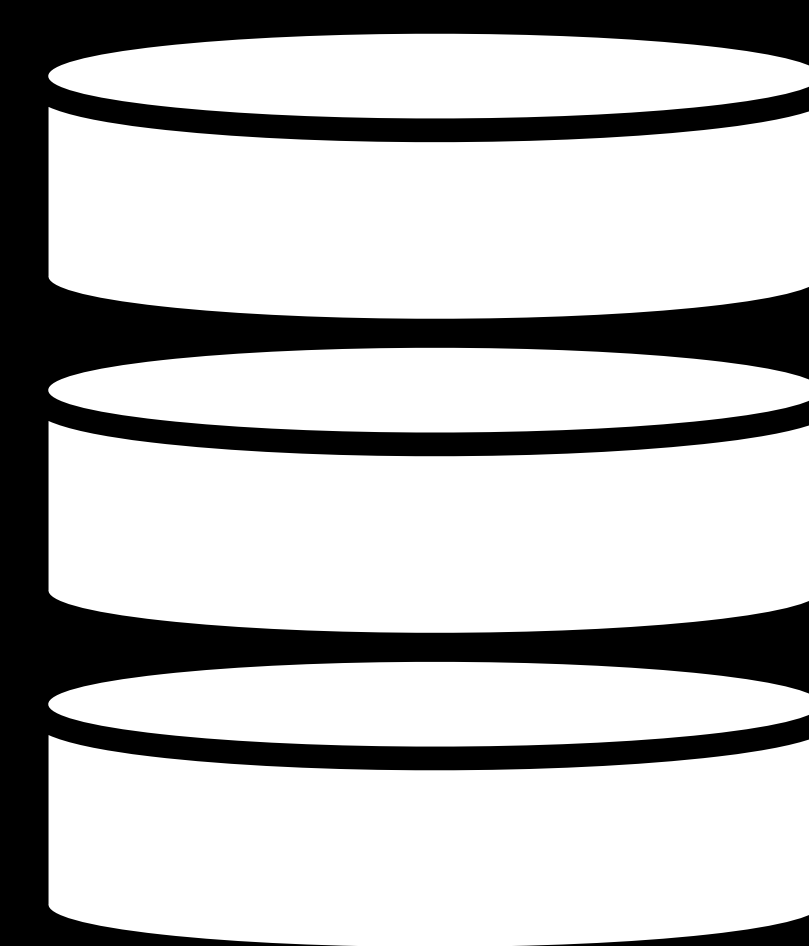
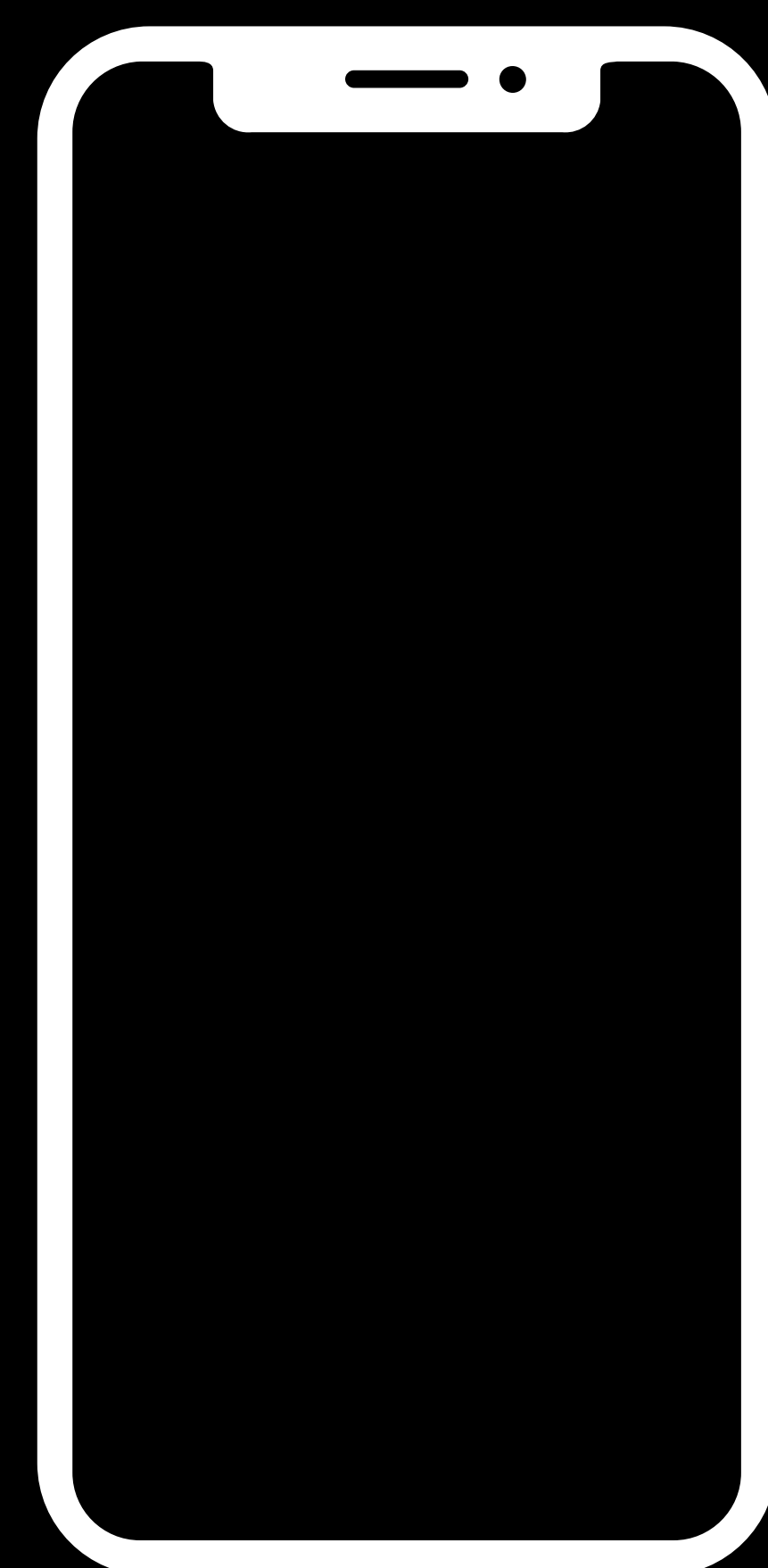
NSManagedObject to CKRecord



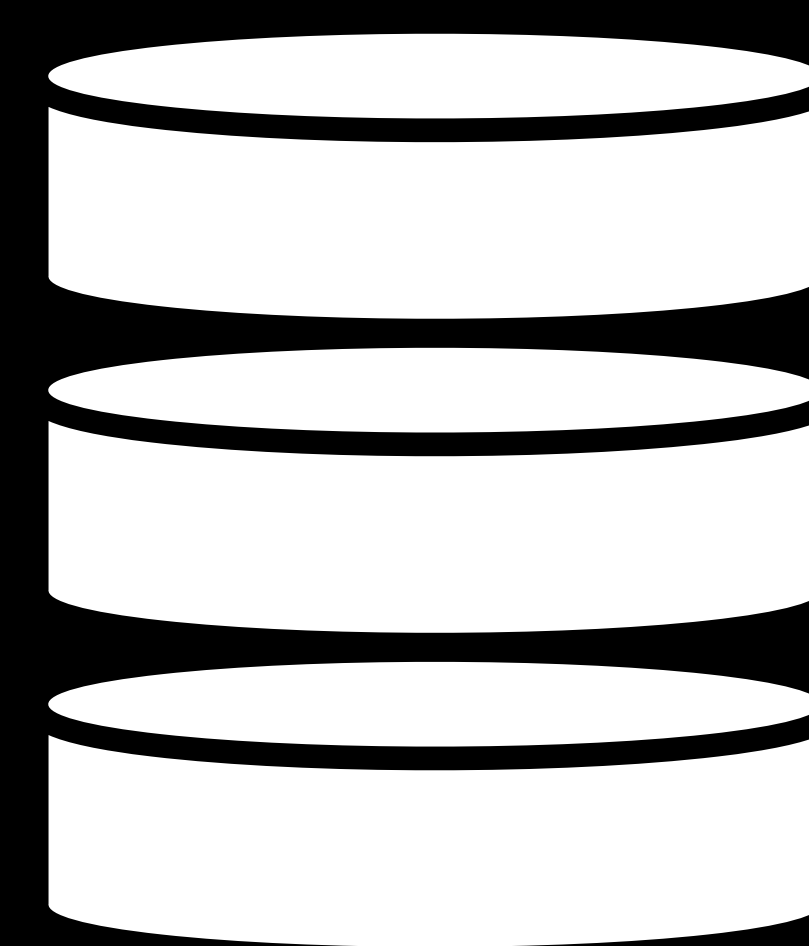
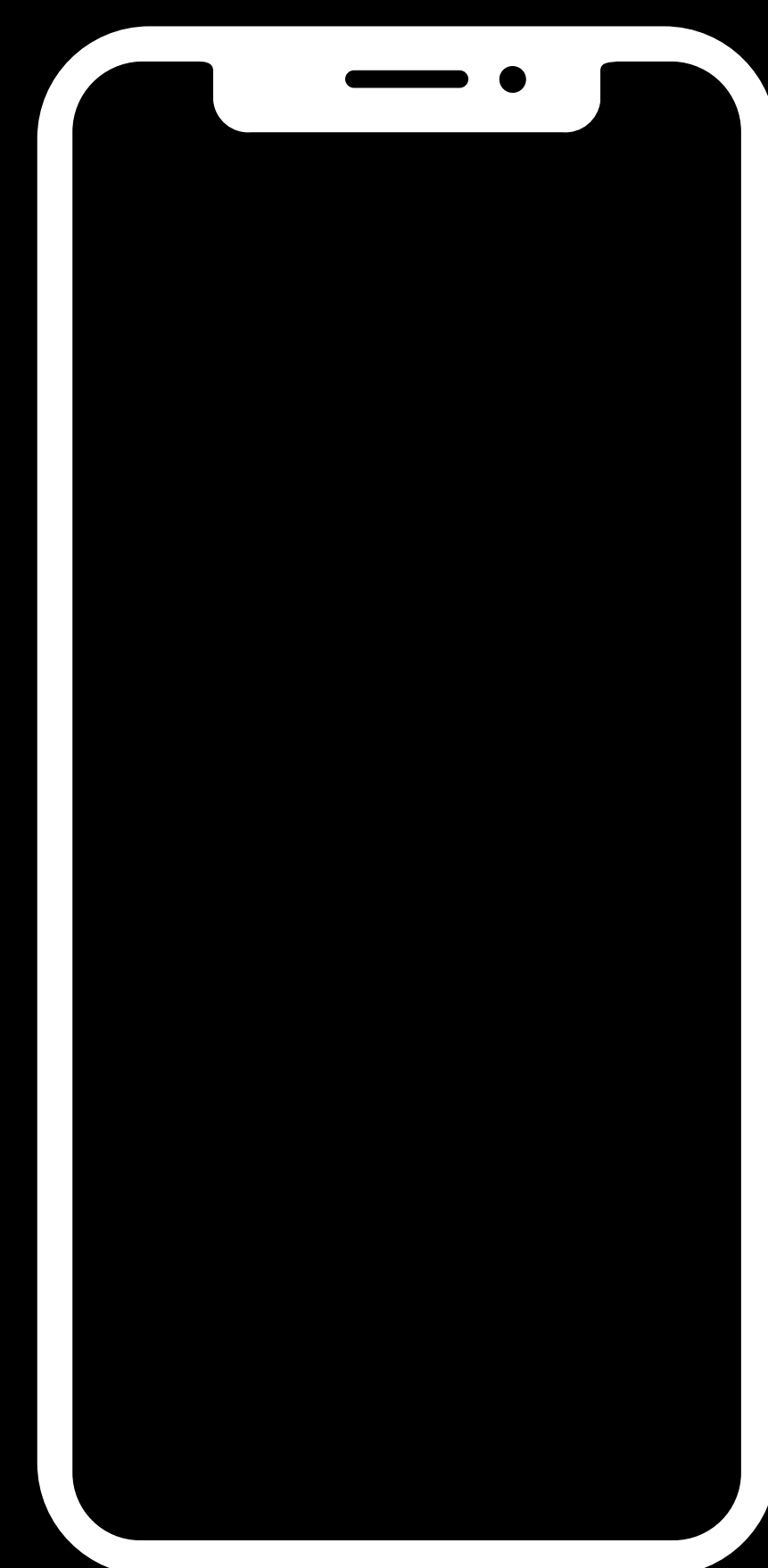
NSManagedObject to CKRecord



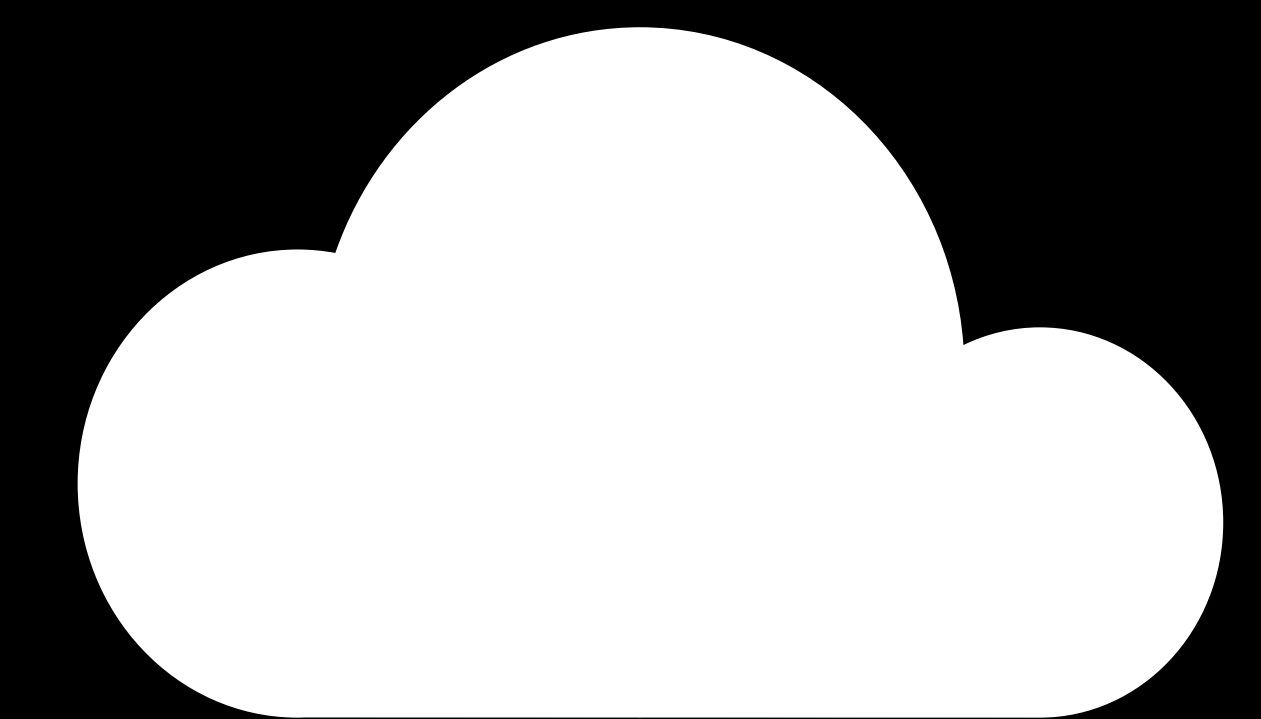
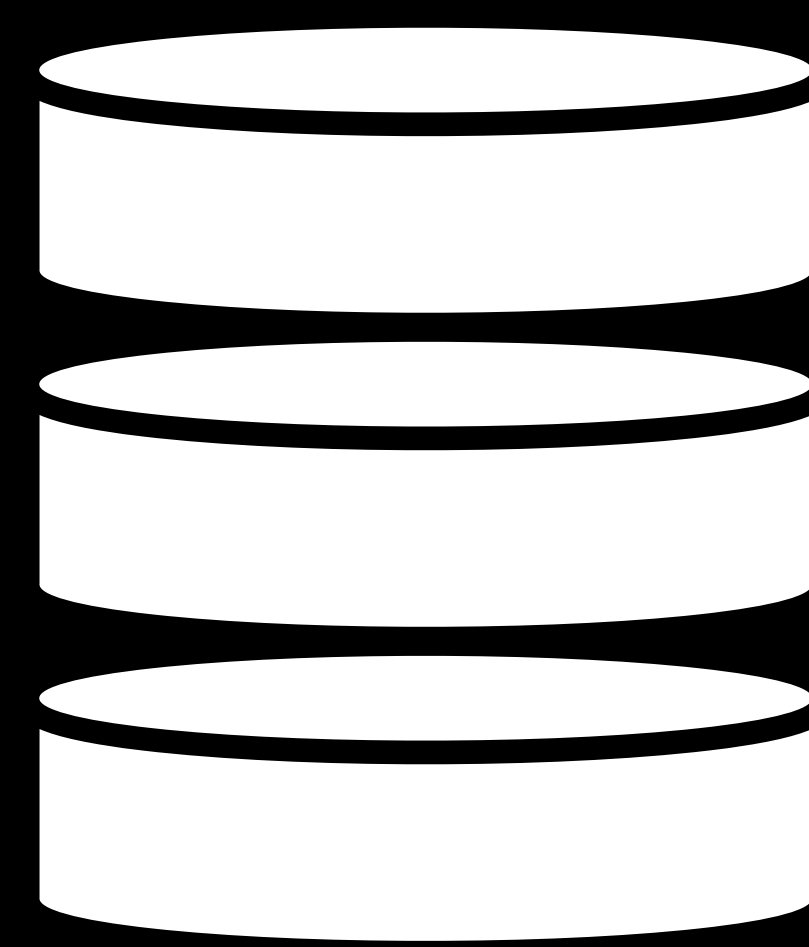
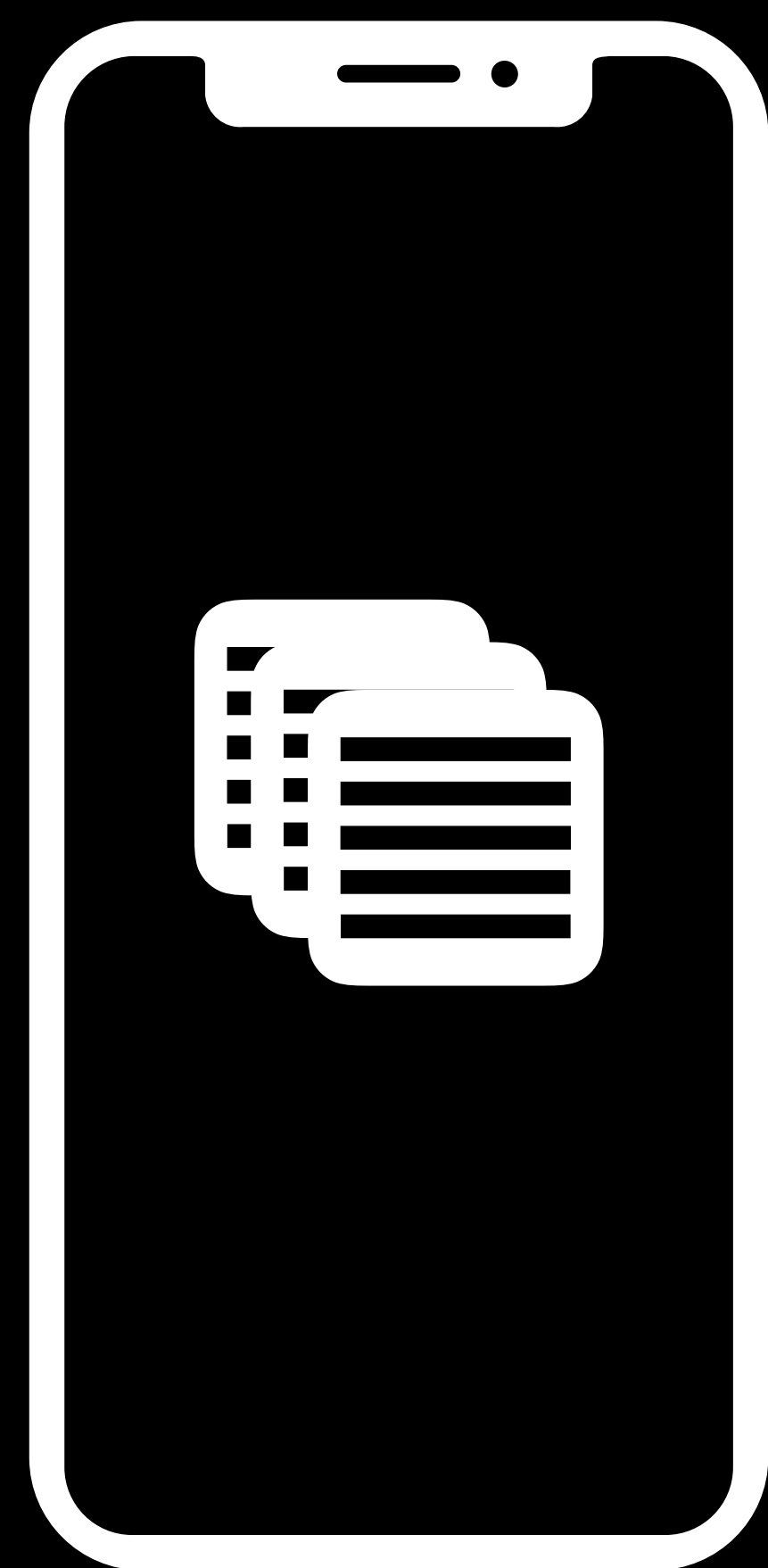
NSManagedObject to CKRecord



NSManagedObject to CKRecord



NSManagedObject to CKRecord



NSPersistentCloudKitContainer

NSPersistentCloudKitContainer

Complete local replica

NSPersistentCloudKitContainer

Complete local replica

Automatic scheduling

NSPersistentCloudKitContainer

Complete local replica

Automatic scheduling

Serialization from NSManagedObject to CKRecord

Life After Adopting NSPersistentCloudKitContainer

Life After Adopting NSPersistentCloudKitContainer

Build great applications with Core Data

Life After Adopting NSPersistentCloudKitContainer

Build great applications with Core Data

Add on to our foundation

Learning About NSPersistentCloudKitContainer

Learning About NSPersistentCloudKitContainer

Mirroring a Core Data Store with CloudKit

Setting Up Core Data with CloudKit

Creating a Core Data Model for CloudKit

Syncing a Core Data Store with CloudKit

Reading CloudKit Records for Core Data

Build Great Apps with Core Data

Build Great Apps with Core Data

Responsive user interfaces with NSFetchedResultsController

Build Great Apps with Core Data

Responsive user interfaces with NSFetchedResultsController

Stable views with query generations

Build Great Apps with Core Data

Responsive user interfaces with NSFetchedResultsController

Stable views with query generations

Change processing with history tracking

Build Great Apps with Core Data

Responsive user interfaces with NSFetchedResultsController

Stable views with query generations

Change processing with history tracking



FaceTime



Calendar



Photos



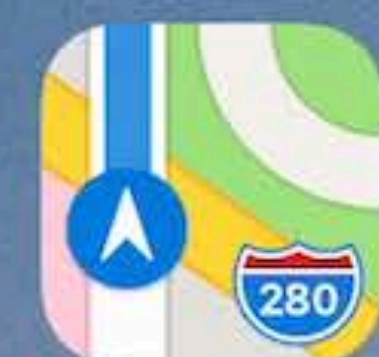
Camera



Contacts



Clock



Maps



Home



Notes



Reminders



Stocks



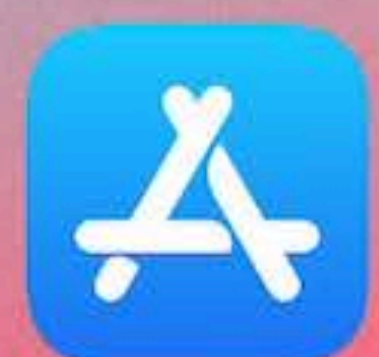
News



TV



iTunes Store



App Store



Books



Settings



Find My



Podcasts



Photo Booth



Feedback





FaceTime



Calendar



Photos



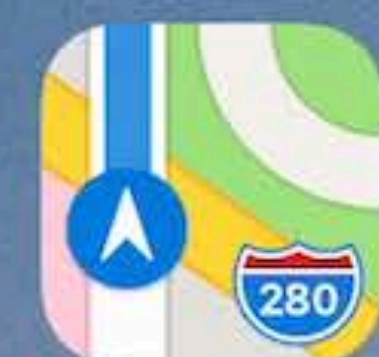
Camera



Contacts



Clock



Maps



Home



Notes



Reminders



Stocks



News



TV



iTunes Store



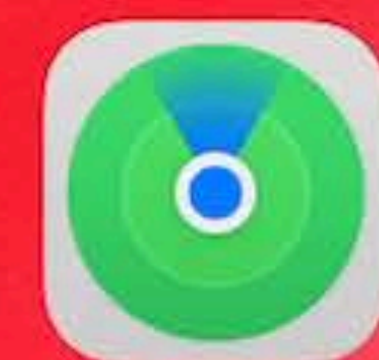
App Store



Books



Settings



Find My



Podcasts



Photo Booth



Feedback



Edit Posts +



Detail

Edit

Another great demo!

demo



TITLE

Hi WWDC!

demo



CONTENT

TAGS



ATTACHMENTS



Edit Posts +



Detail Edit

Another great demo!

demo



TITLE

Hi WWDC!

demo



CONTENT

TAGS



ATTACHMENTS



Life After Adopting NSPersistentCloudKitContainer

Build great applications with Core Data

Add on to our foundation

Extending NSPersistentCloudKitContainer

Extending NSPersistentCloudKitContainer

Working with multiple stores

Extending NSPersistentCloudKitContainer

Working with multiple stores

Working with the CloudKit Schema

Extending NSPersistentCloudKitContainer

Working with multiple stores

Working with the CloudKit Schema

Data Modeling for collaboration

Multiple Store Use Cases

Multiple Store Use Cases

Data Segregation

Multiple Store Use Cases

Data Segregation

Enforcement of different types of constraints

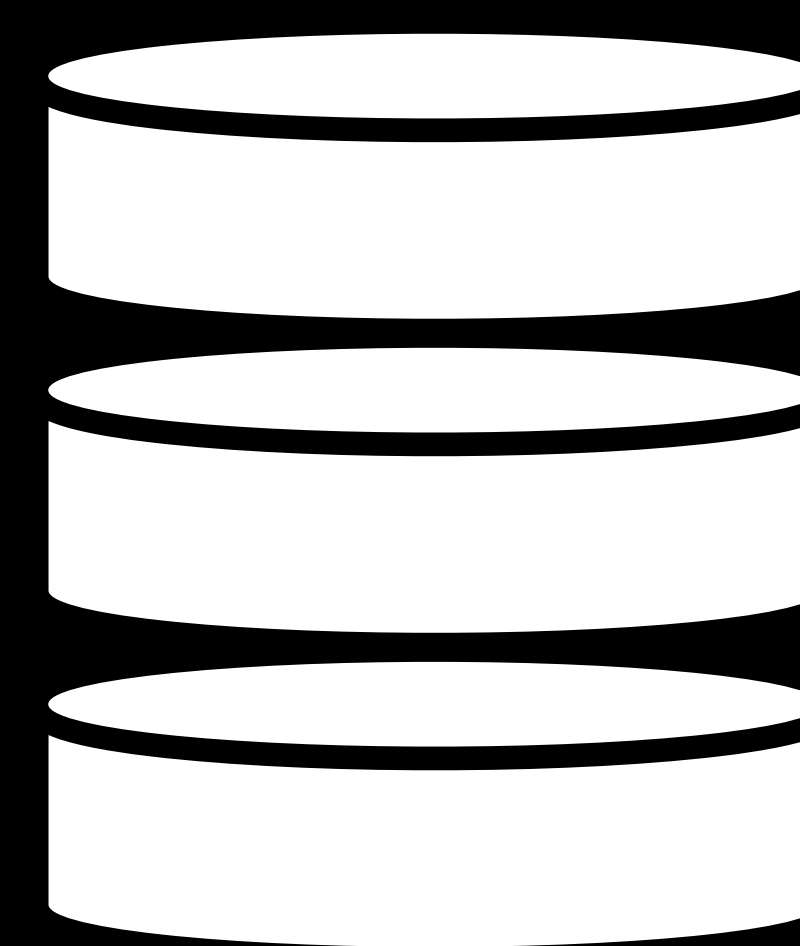
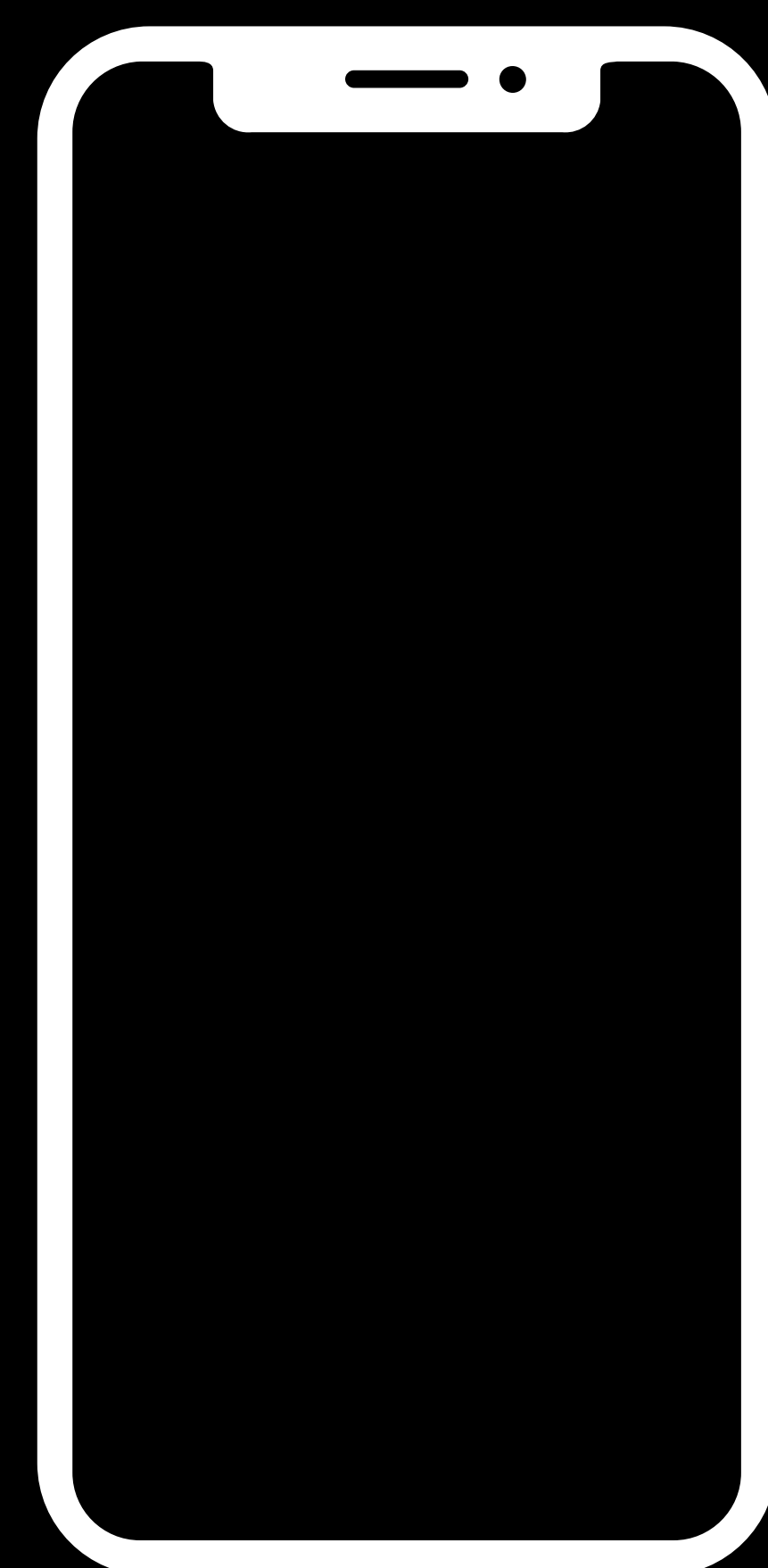
Multiple Store Use Cases

Data Segregation

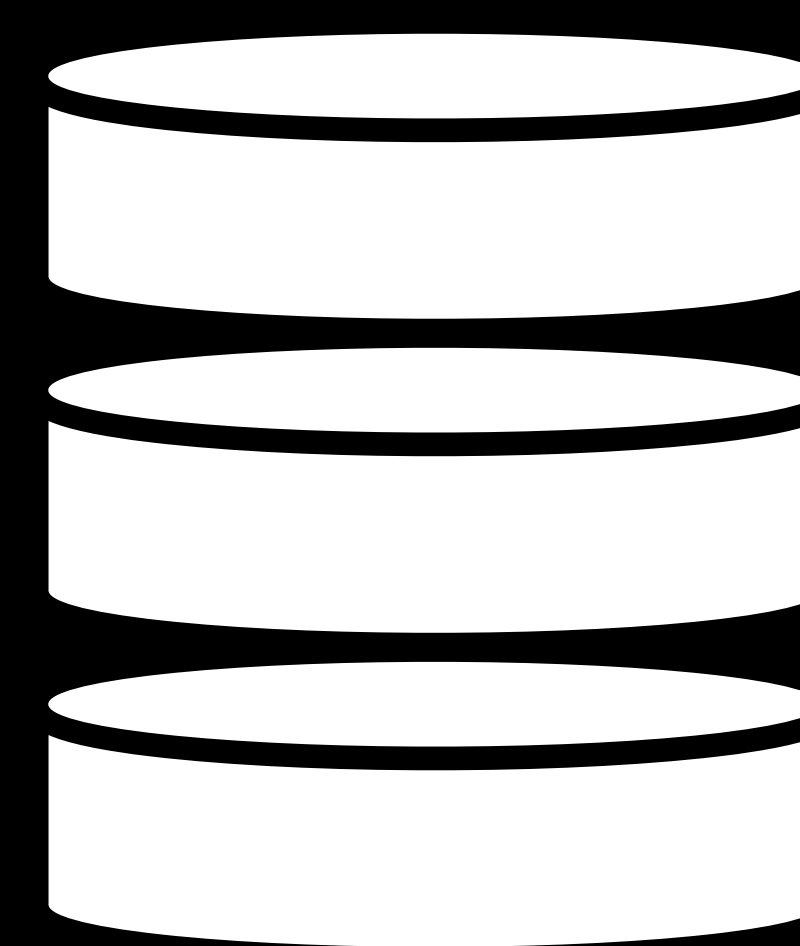
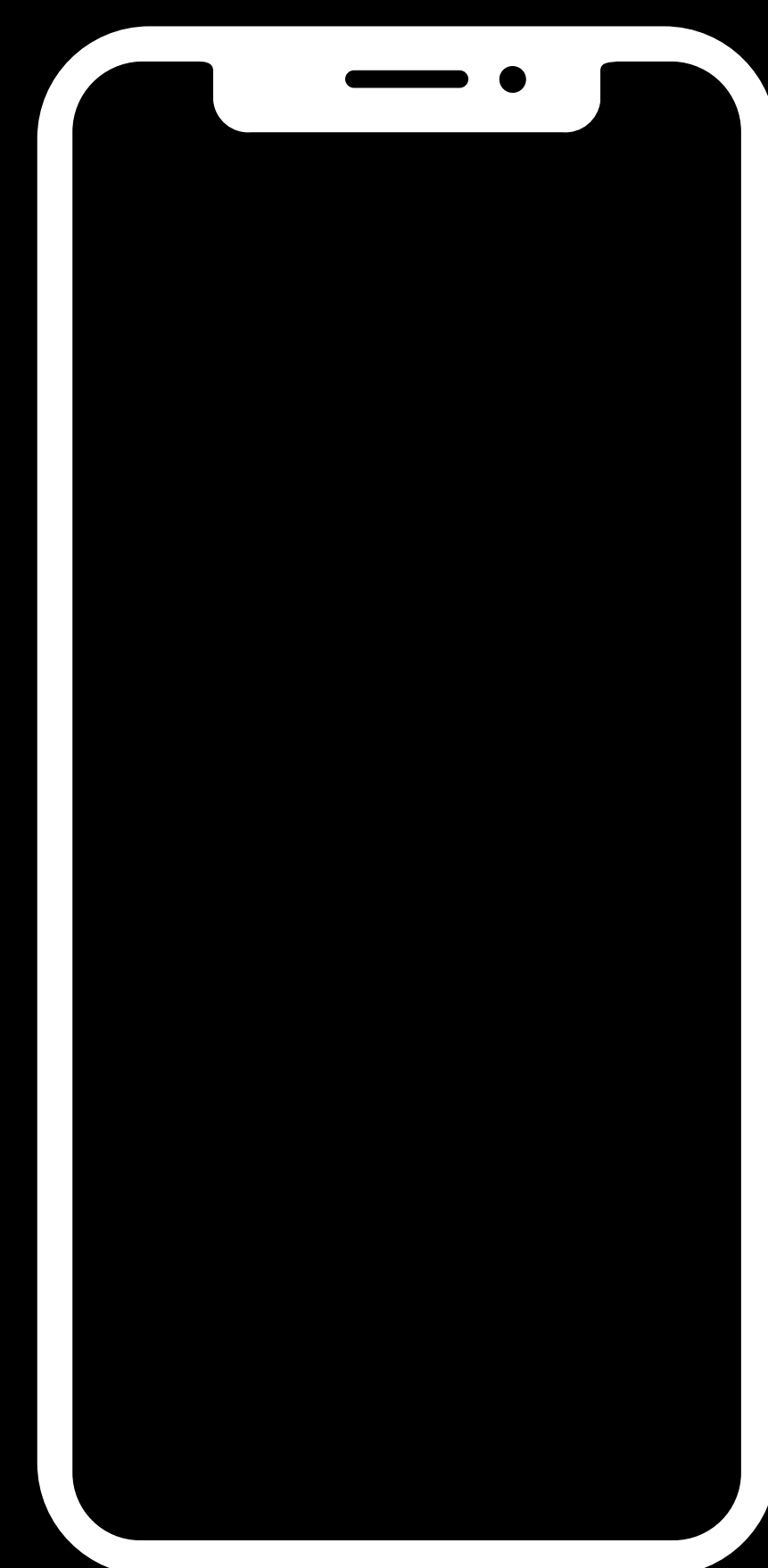
Enforcement of different types of constraints

Throttling/Coalescing

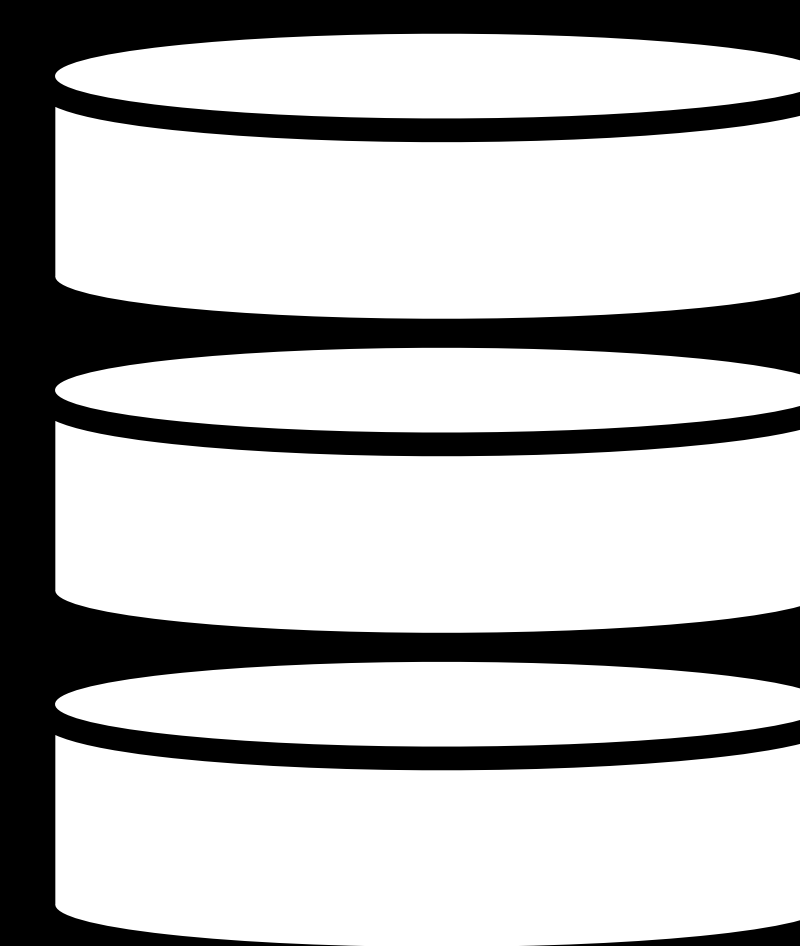
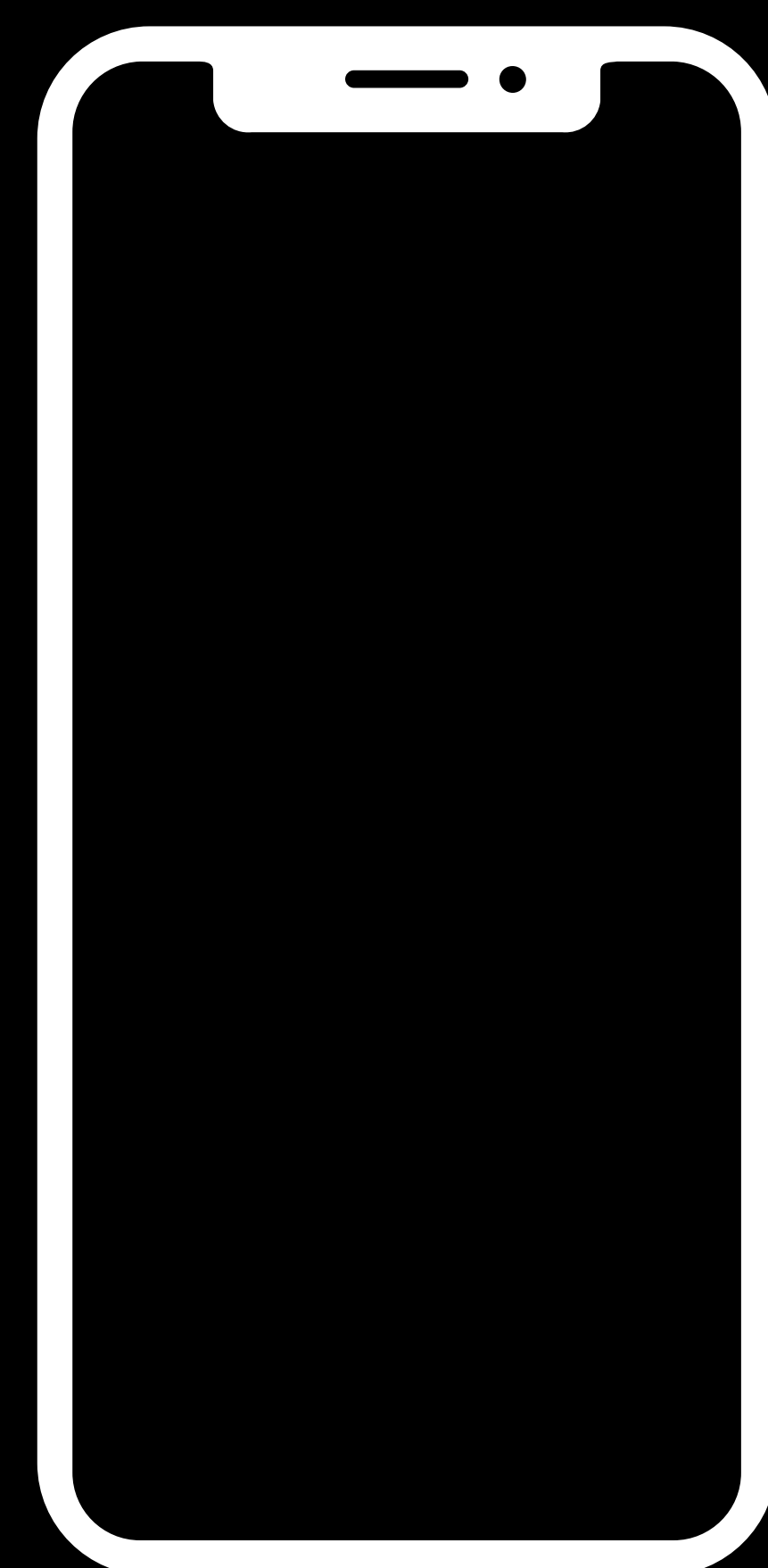
NSPersistentCloudKitContainer — Multiple Stores



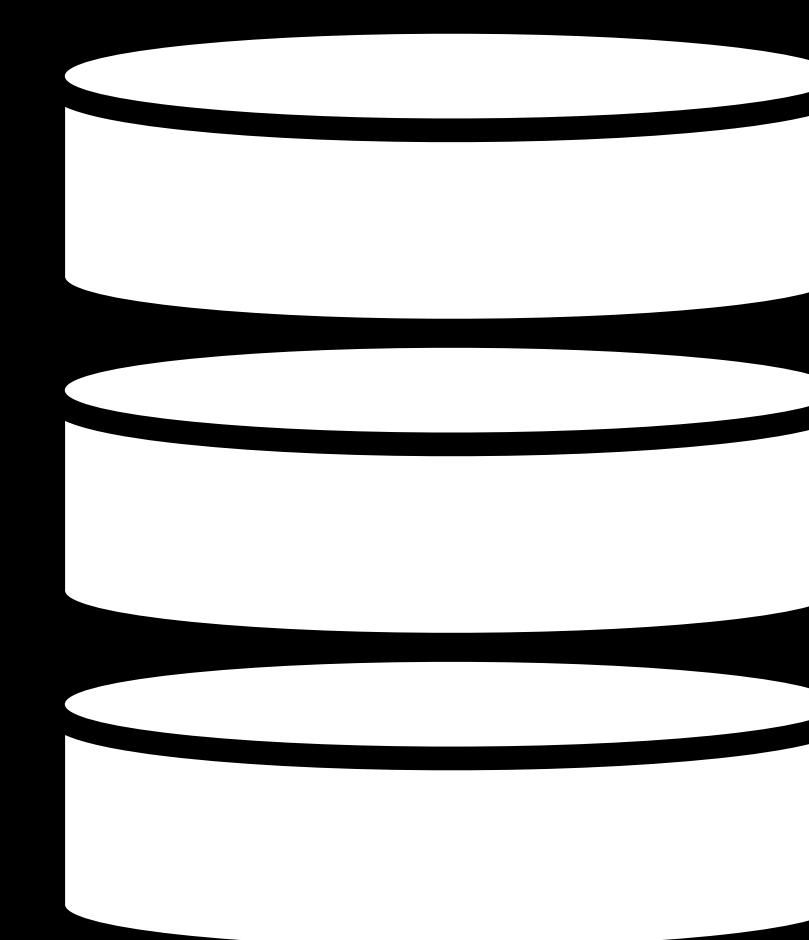
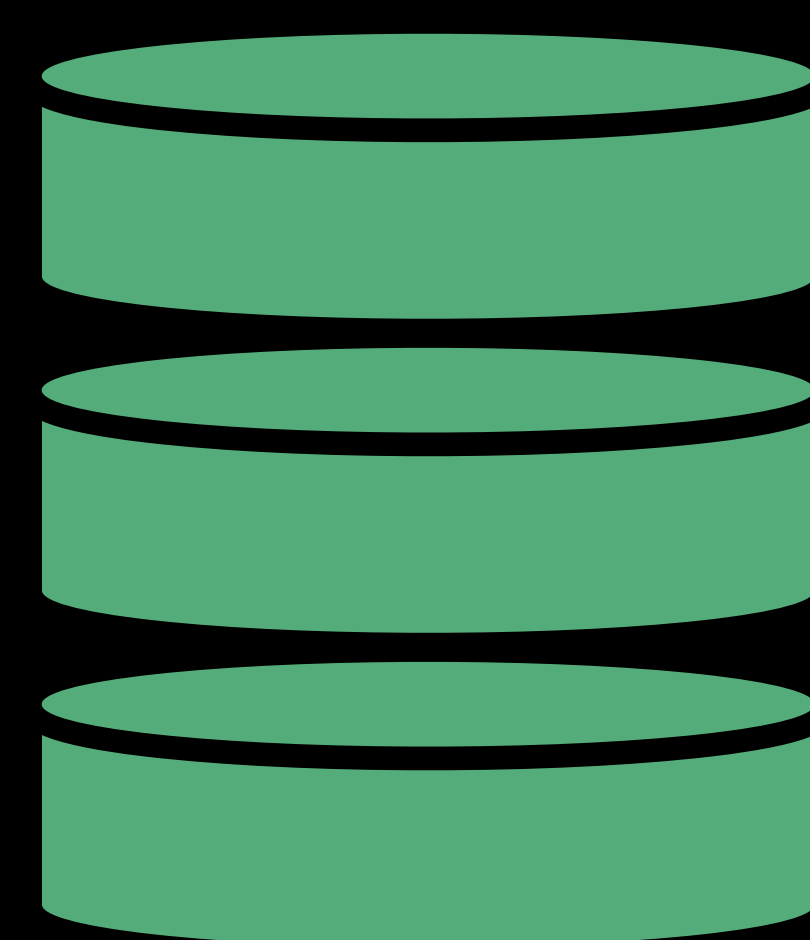
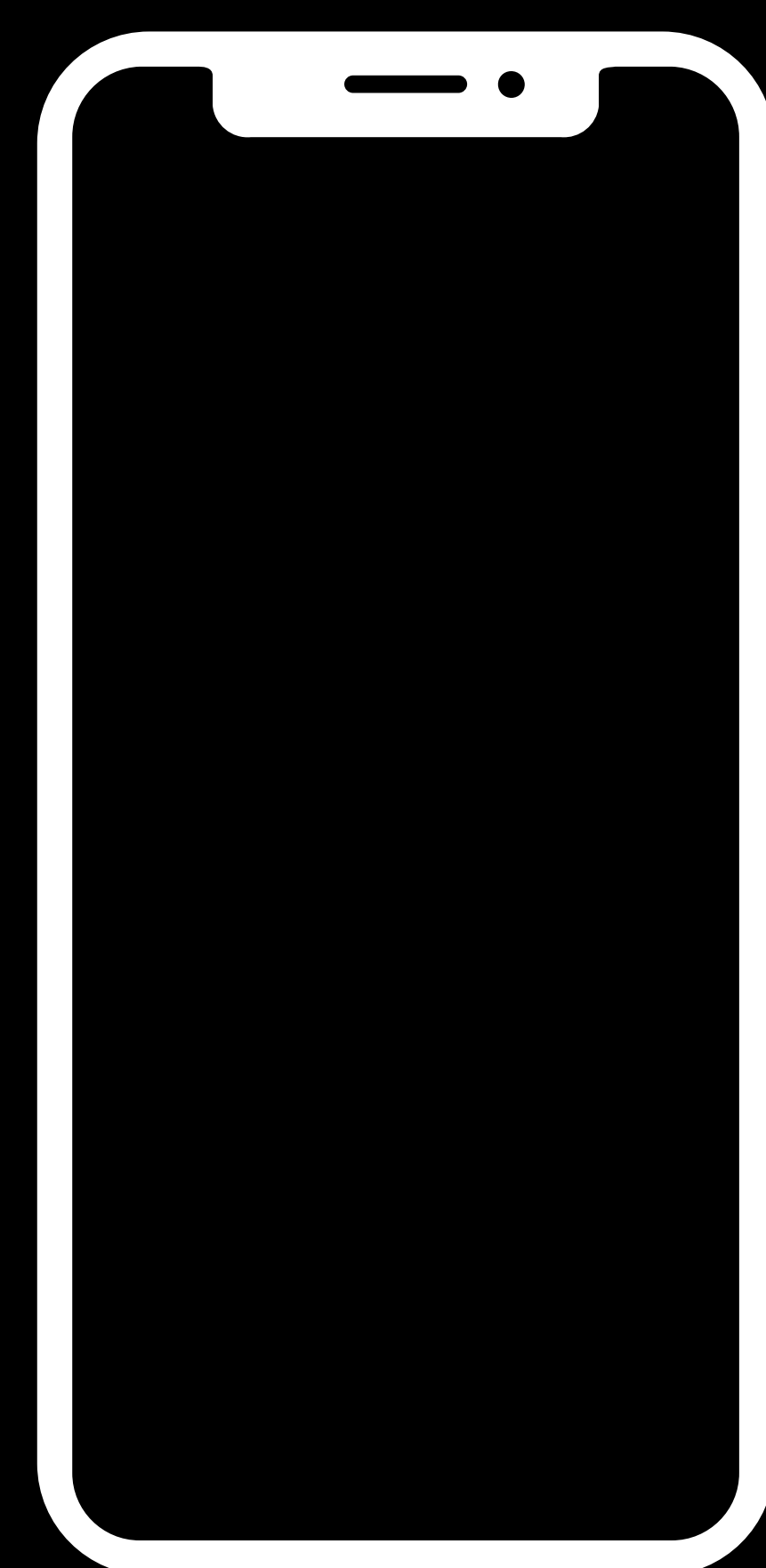
NSPersistentCloudKitContainer — Multiple Stores



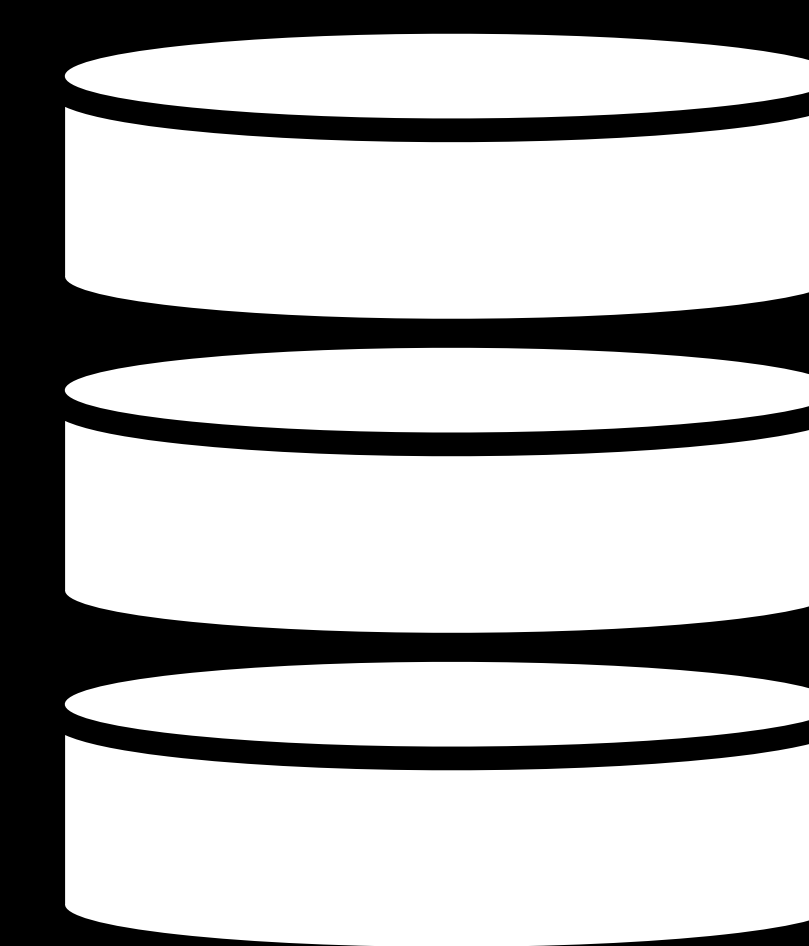
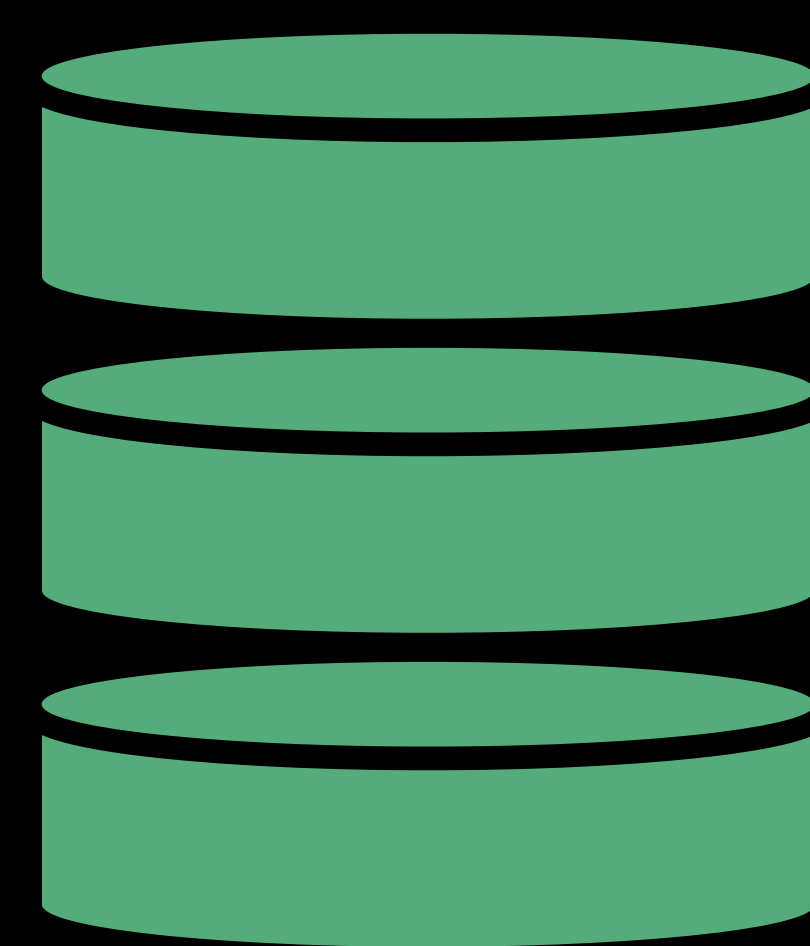
NSPersistentCloudKitContainer — Multiple Stores



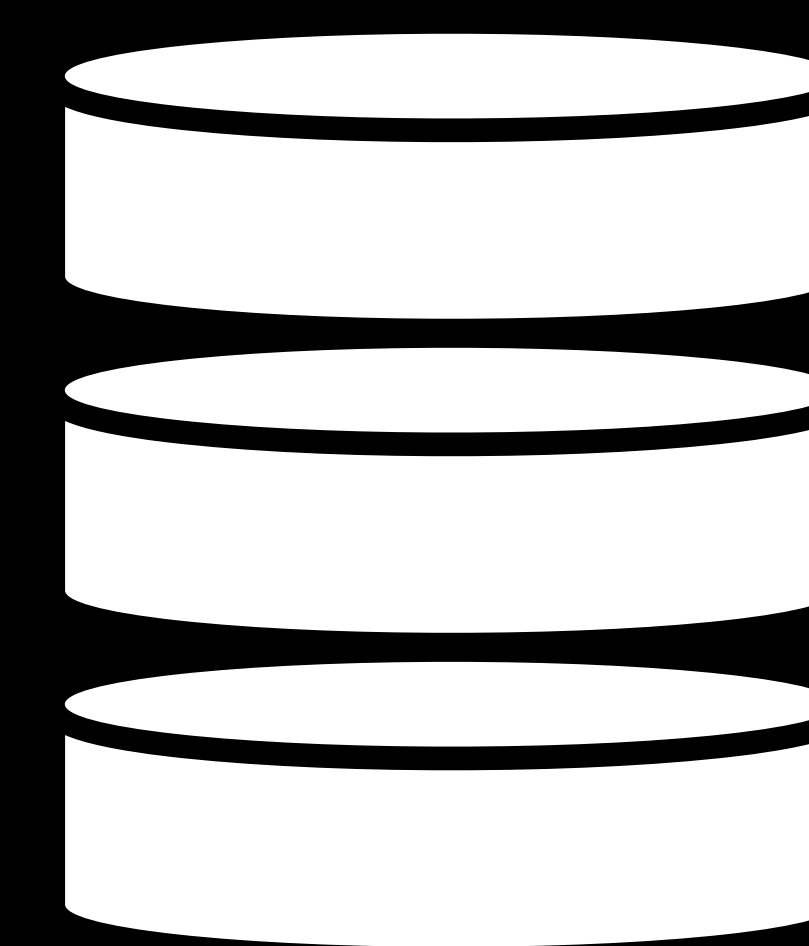
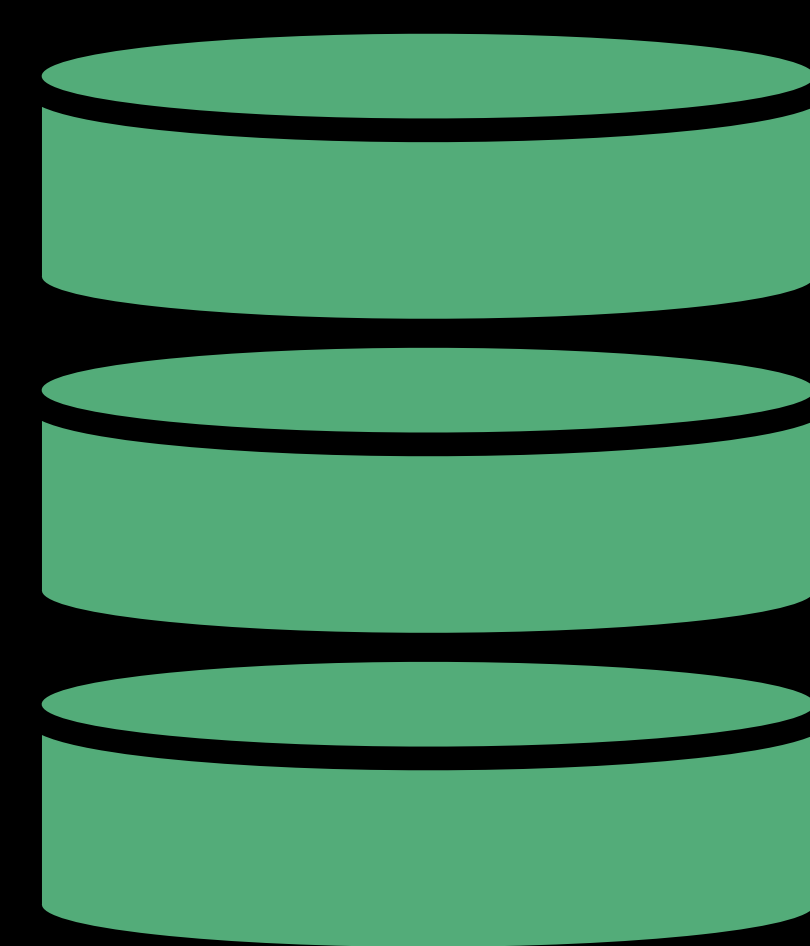
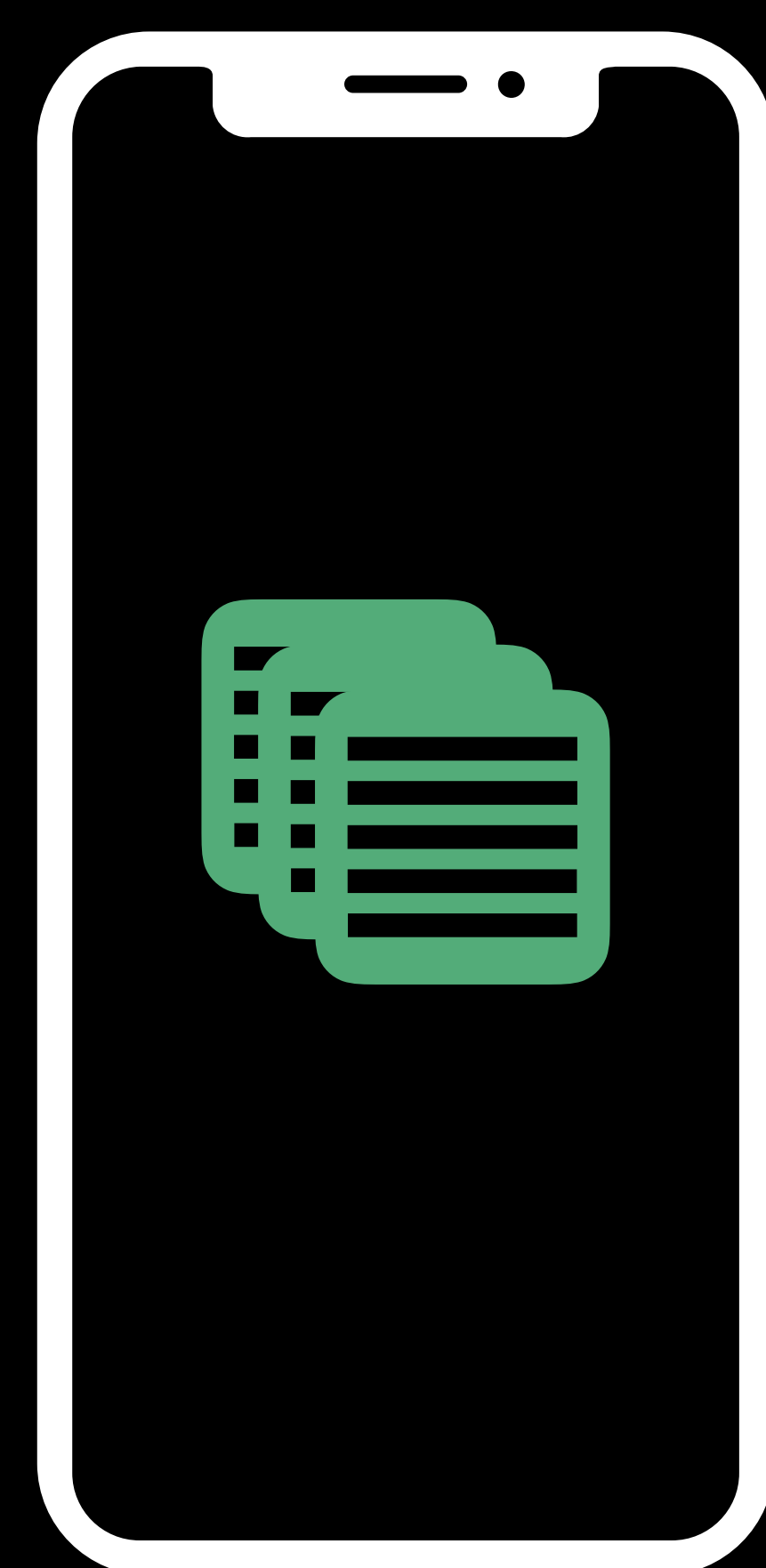
NSPersistentCloudKitContainer — Multiple Stores



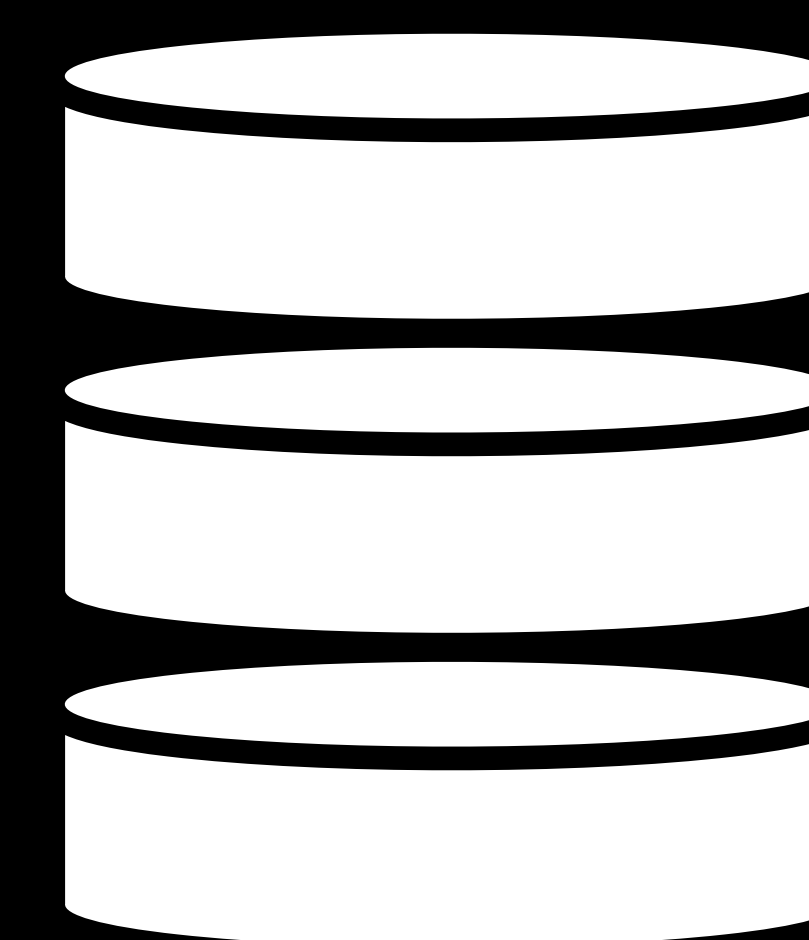
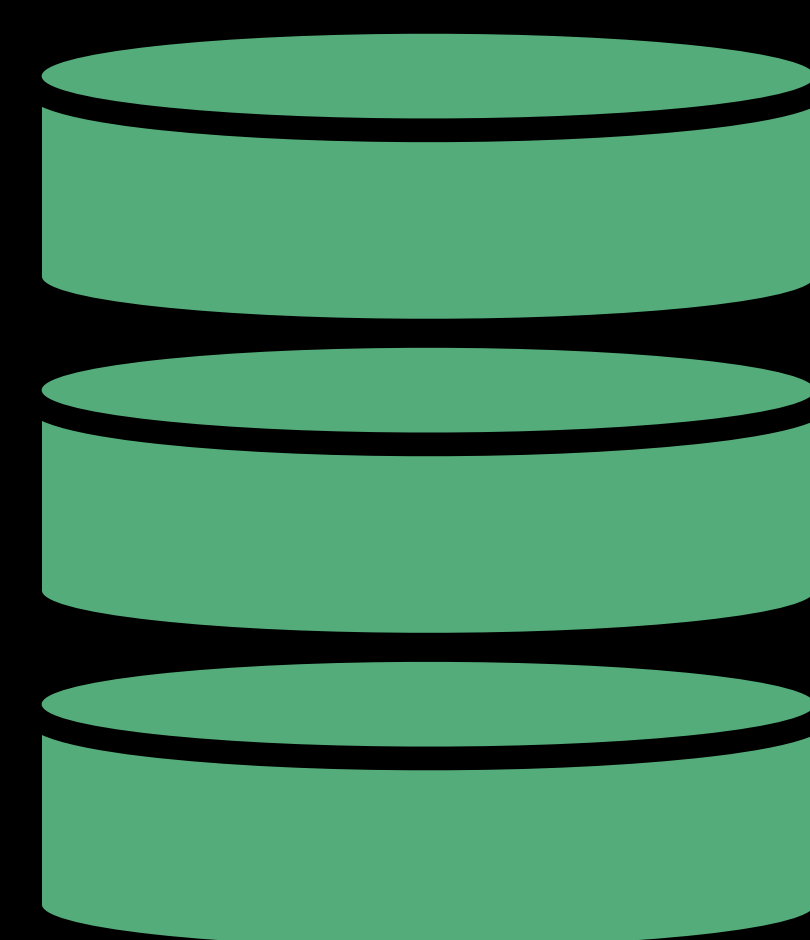
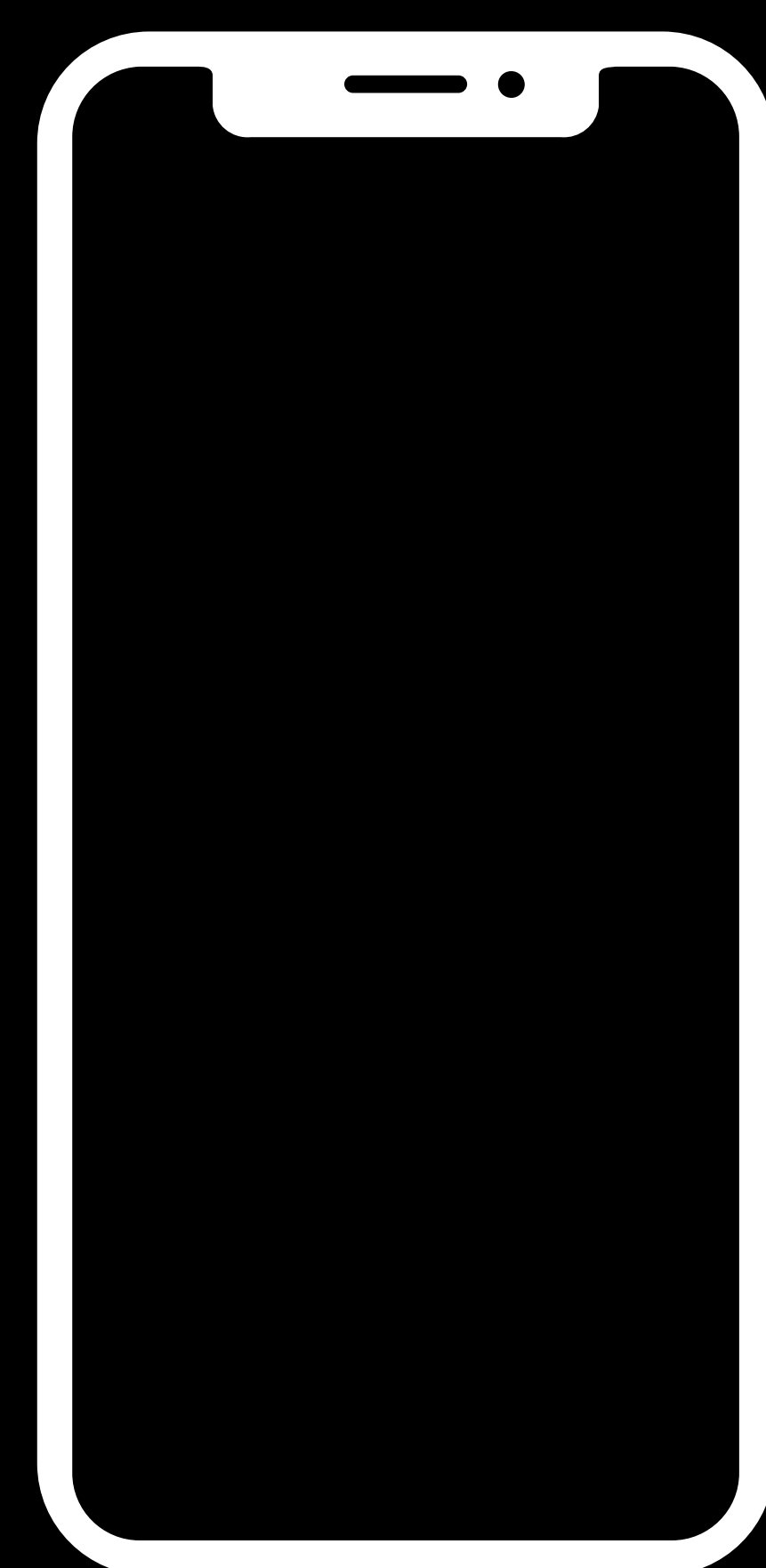
NSPersistentCloudKitContainer — Multiple Stores



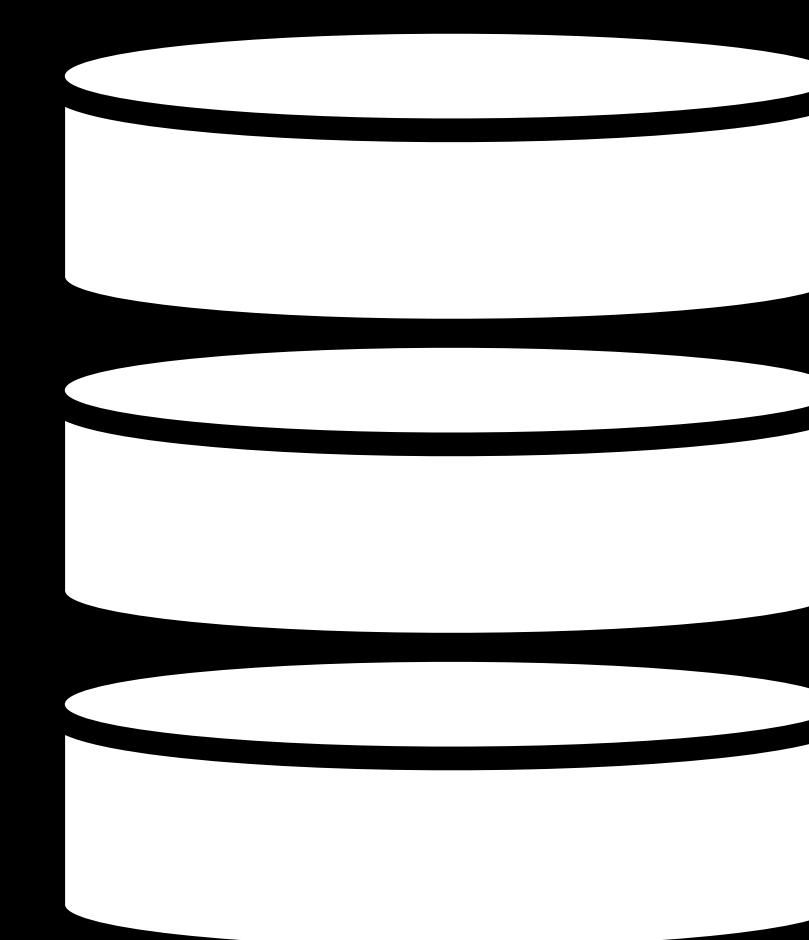
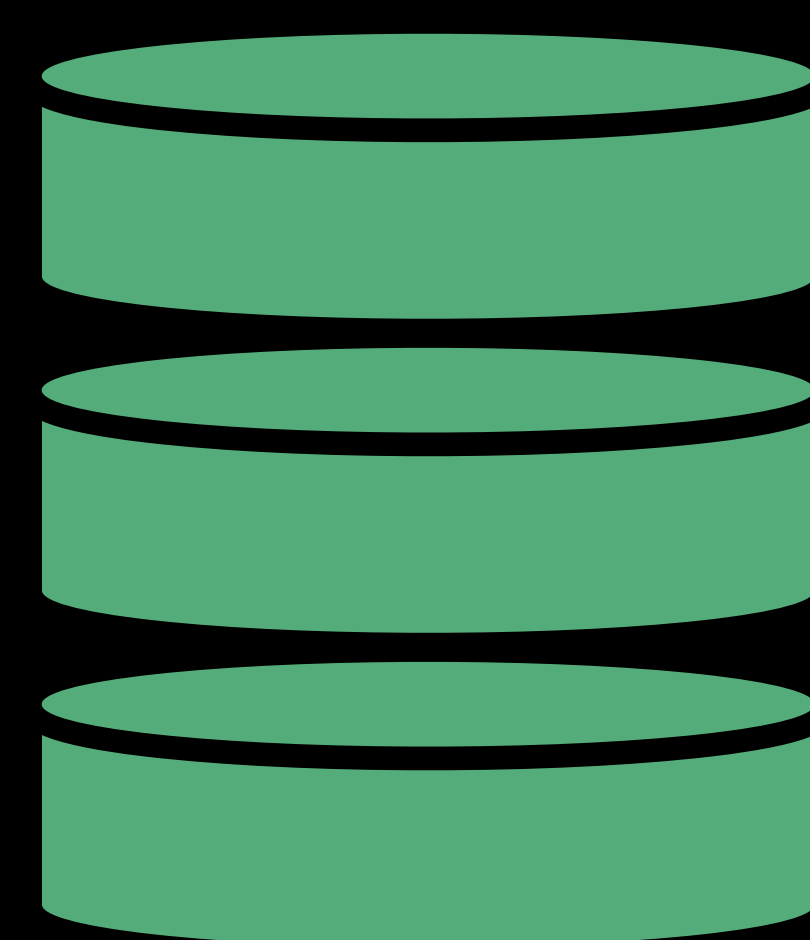
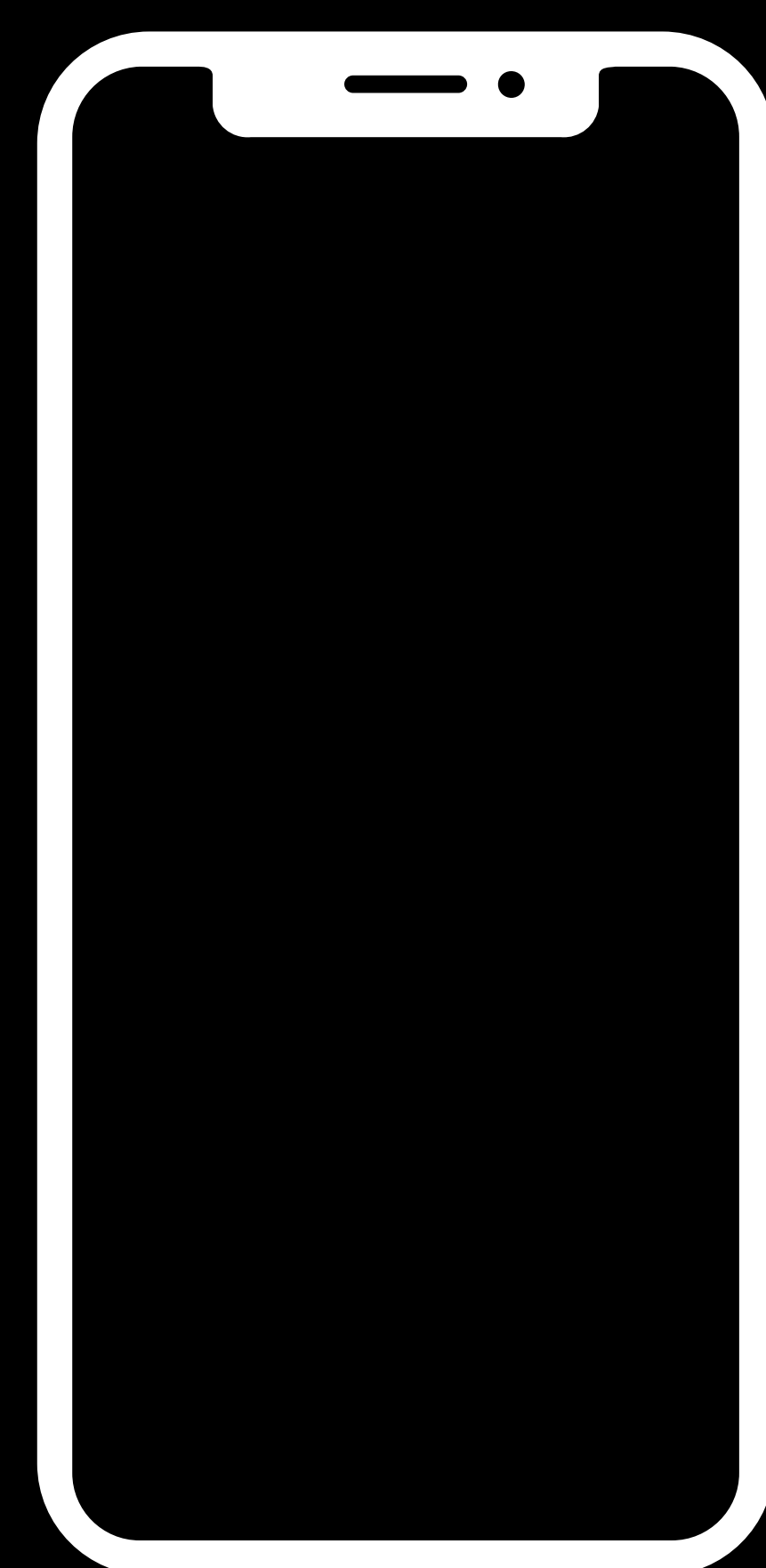
NSPersistentCloudKitContainer — Multiple Stores



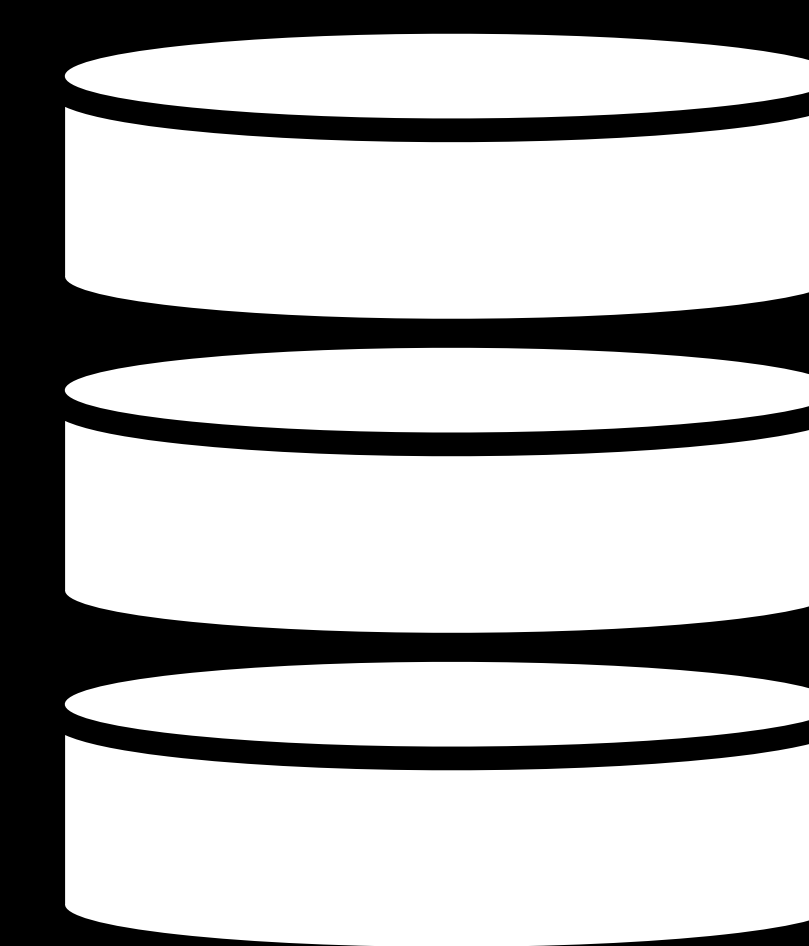
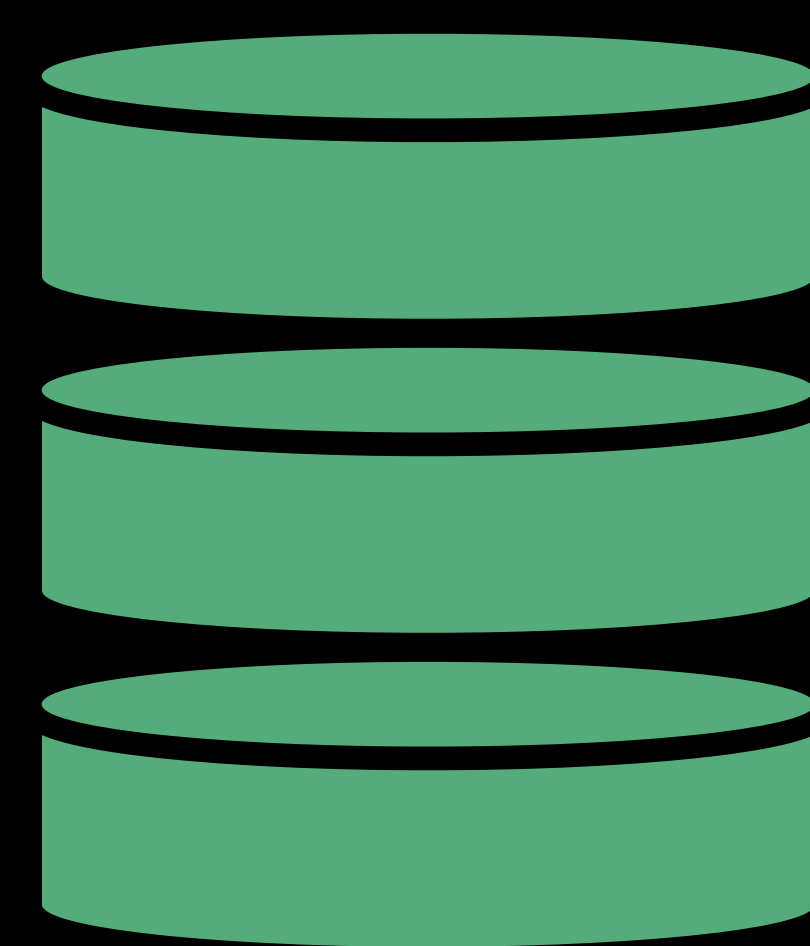
NSPersistentCloudKitContainer — Multiple Stores



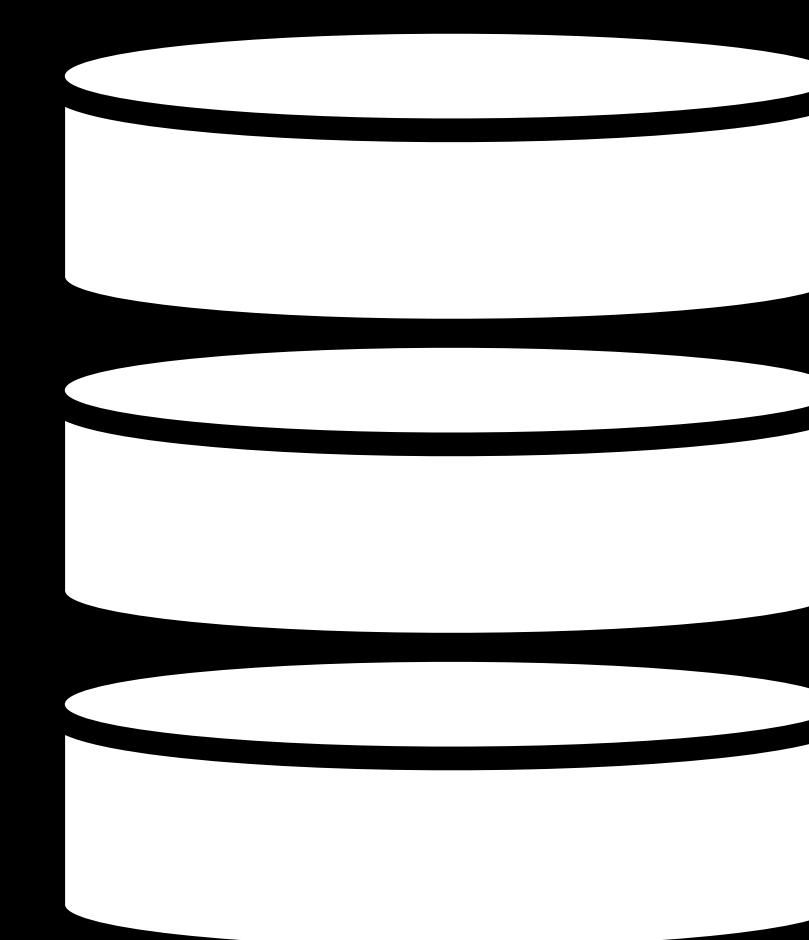
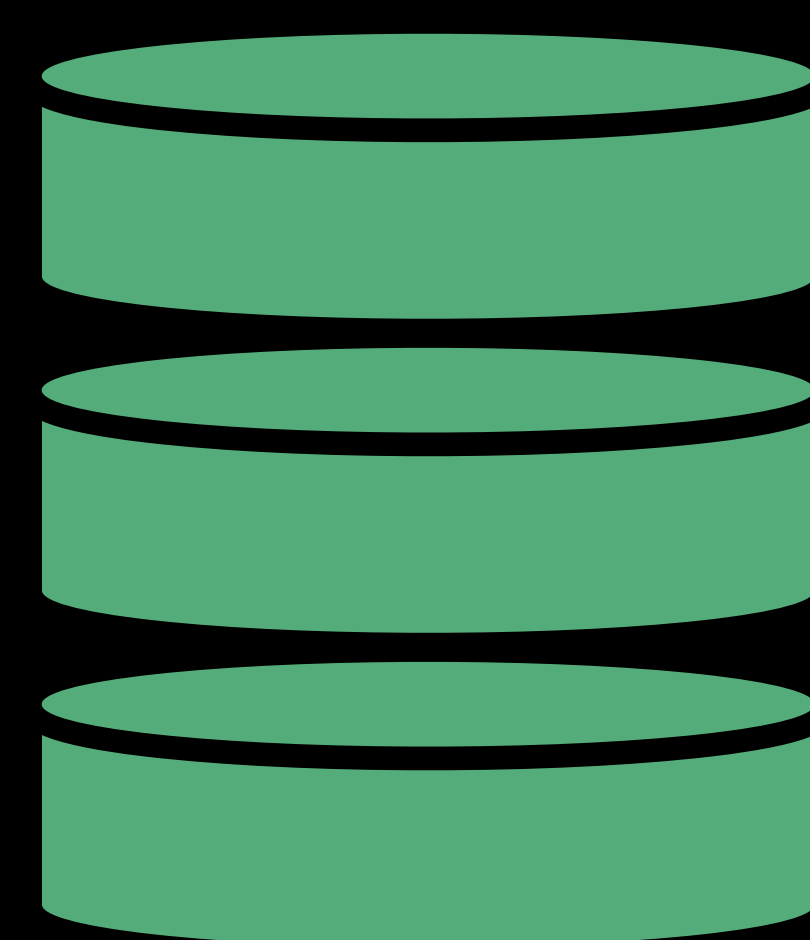
NSPersistentCloudKitContainer — Multiple Stores



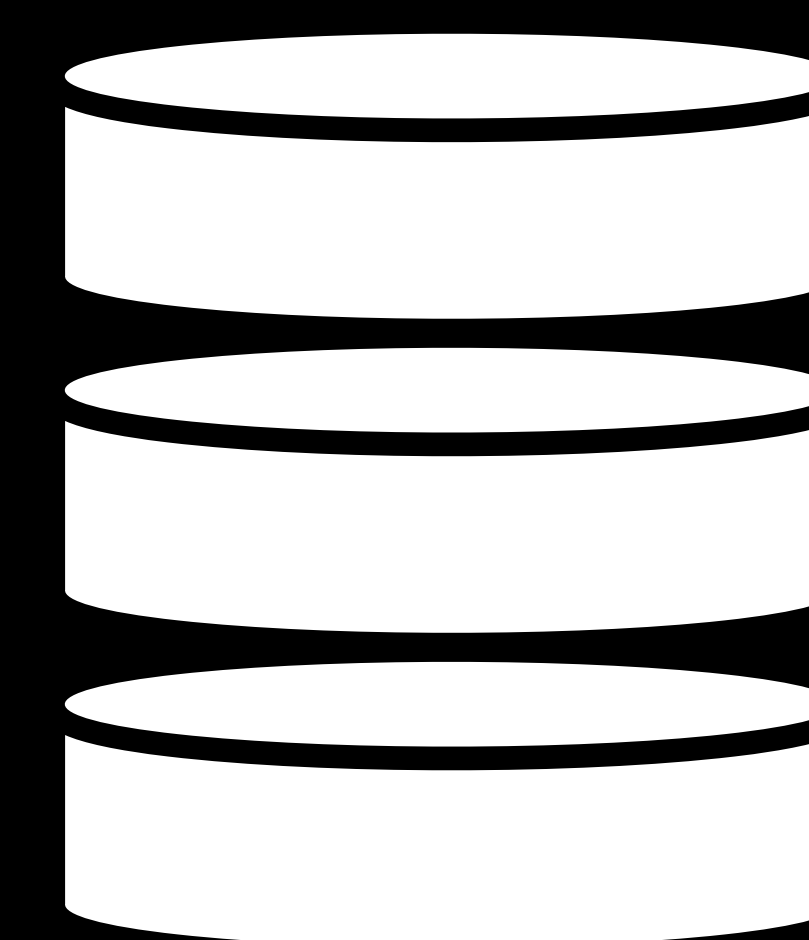
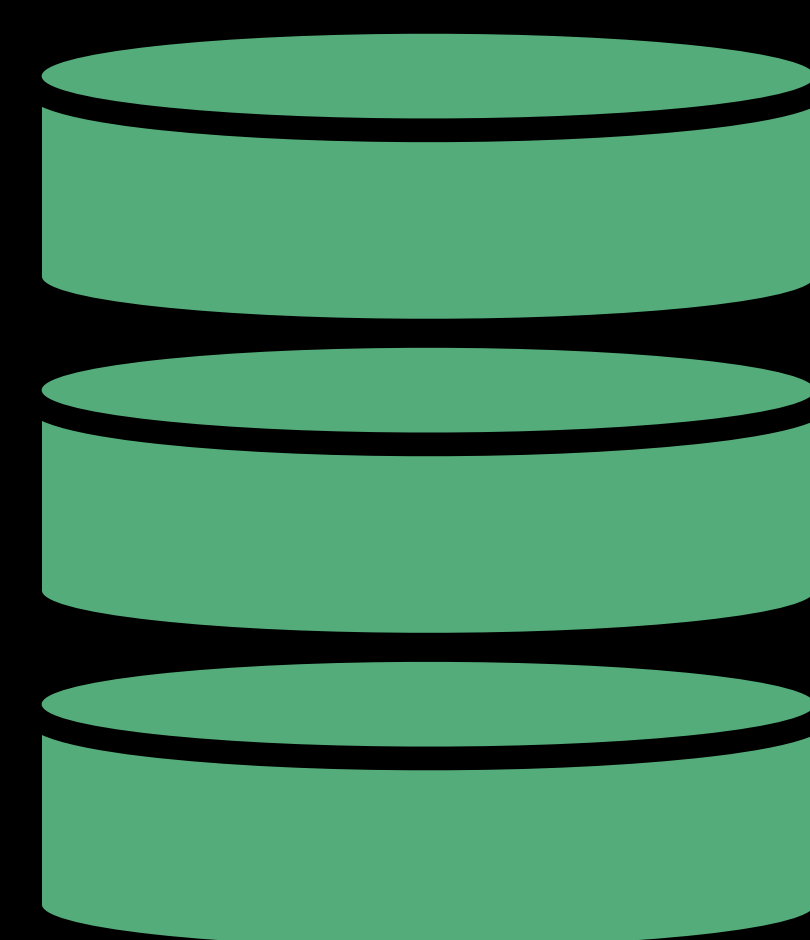
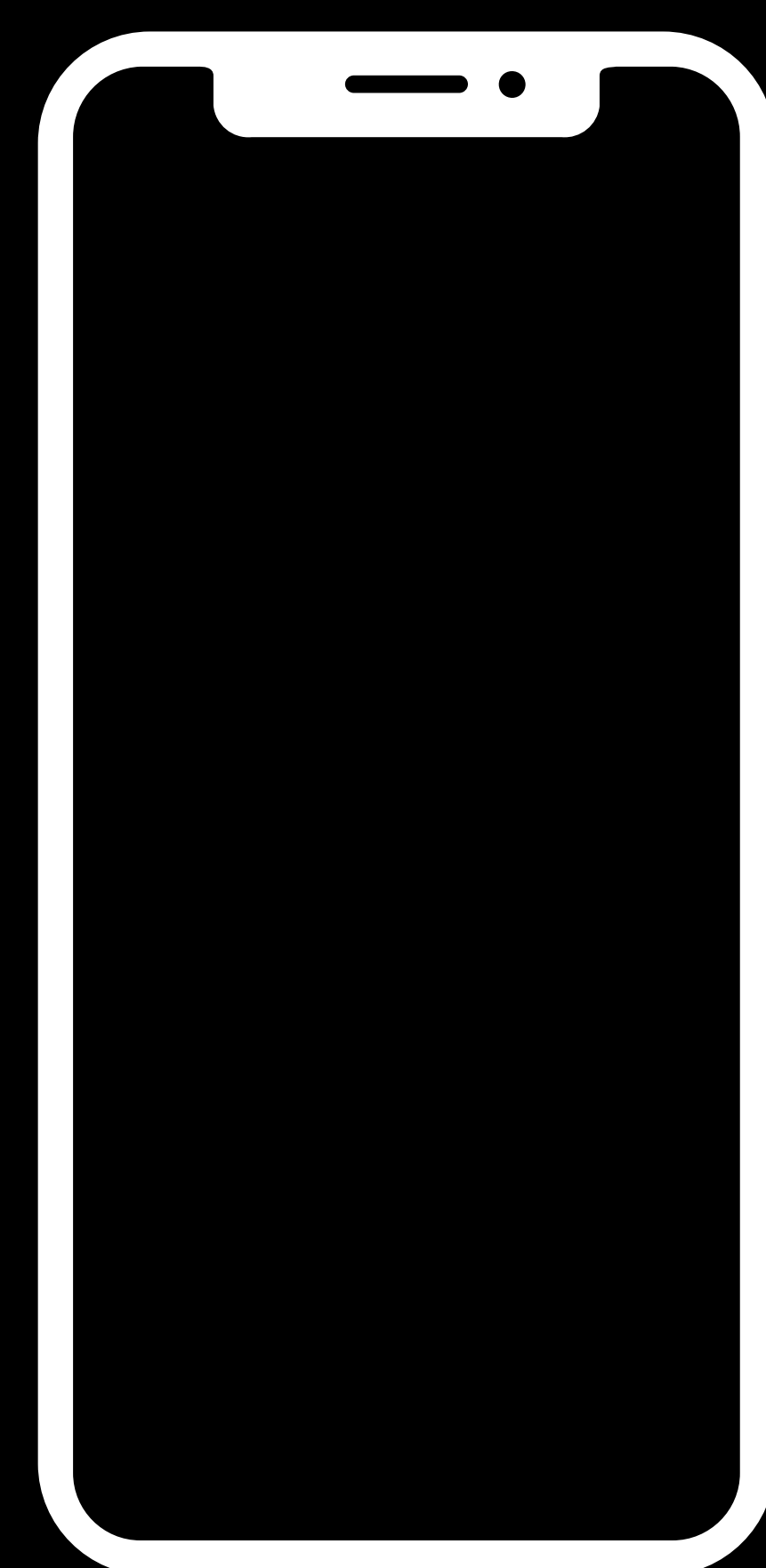
NSPersistentCloudKitContainer — Multiple Stores



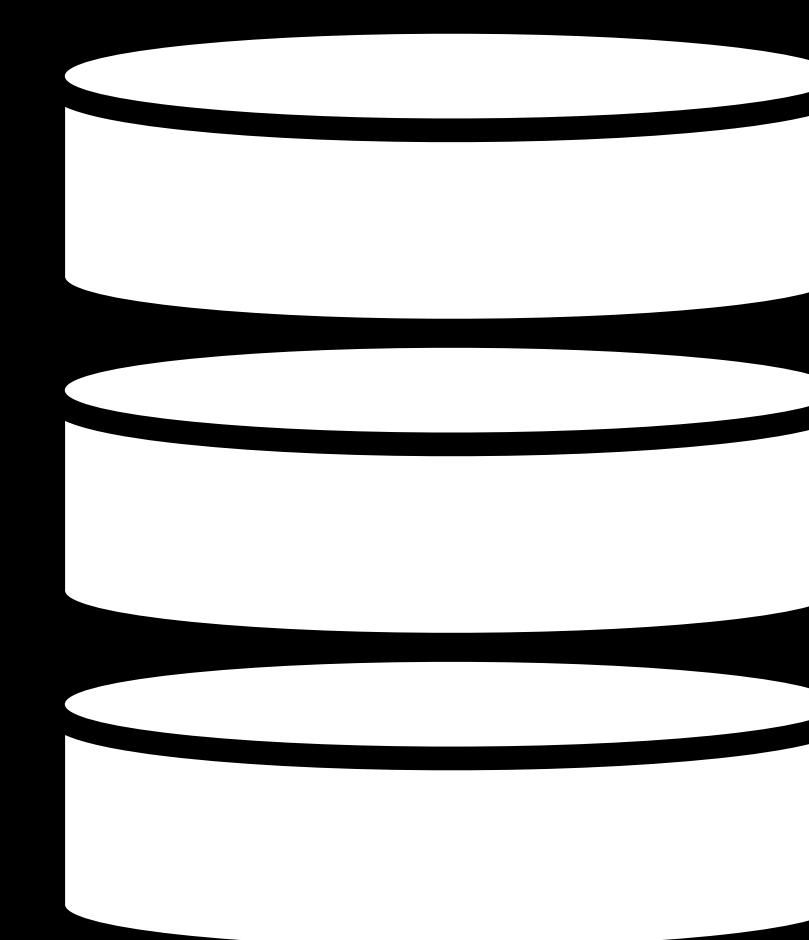
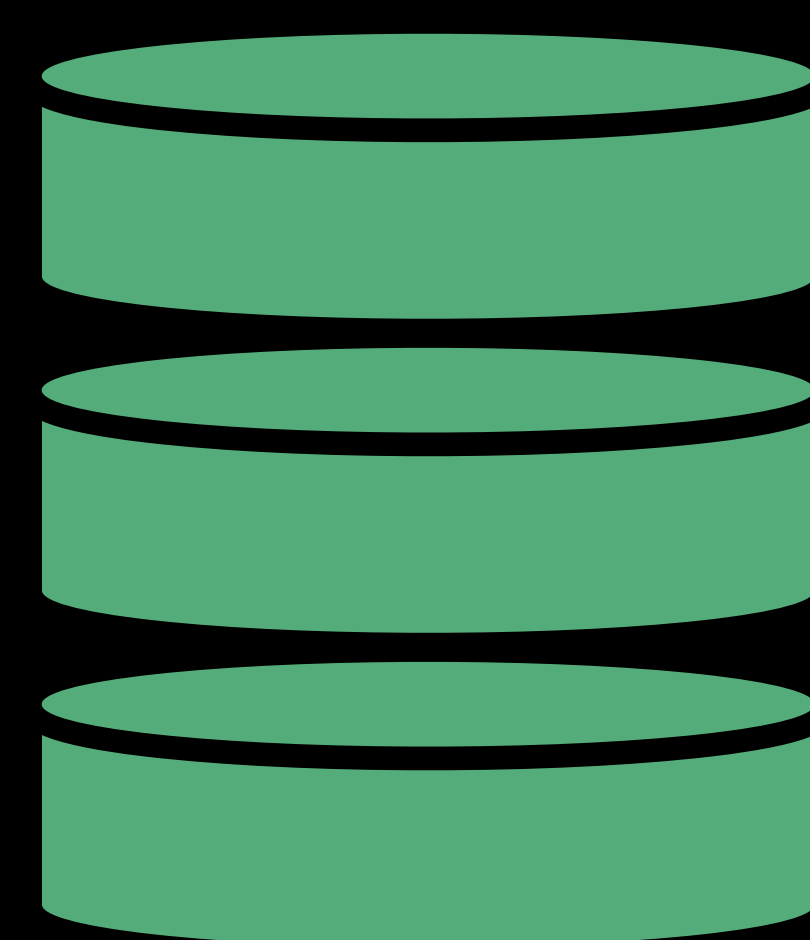
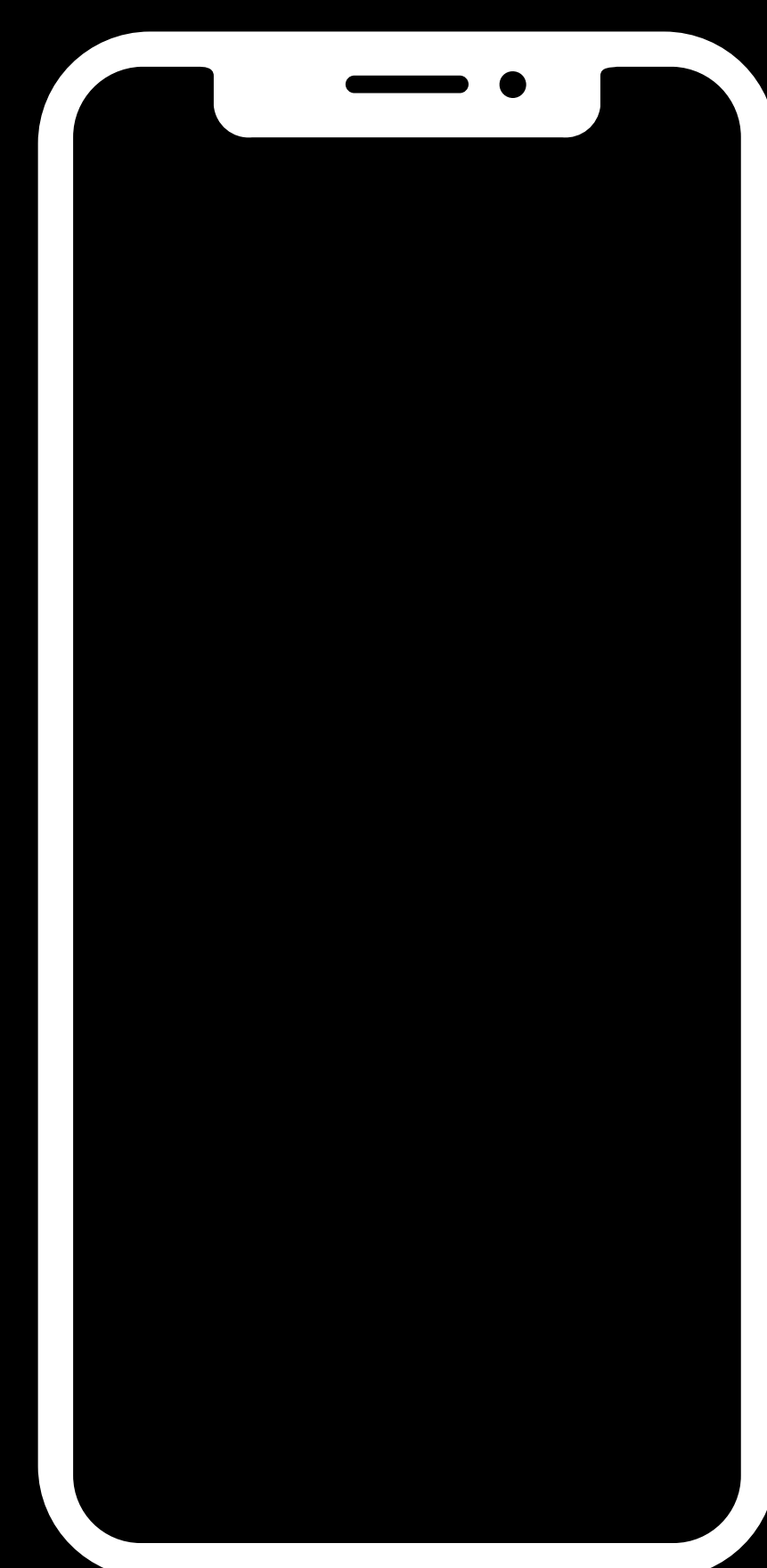
NSPersistentCloudKitContainer — Multiple Stores



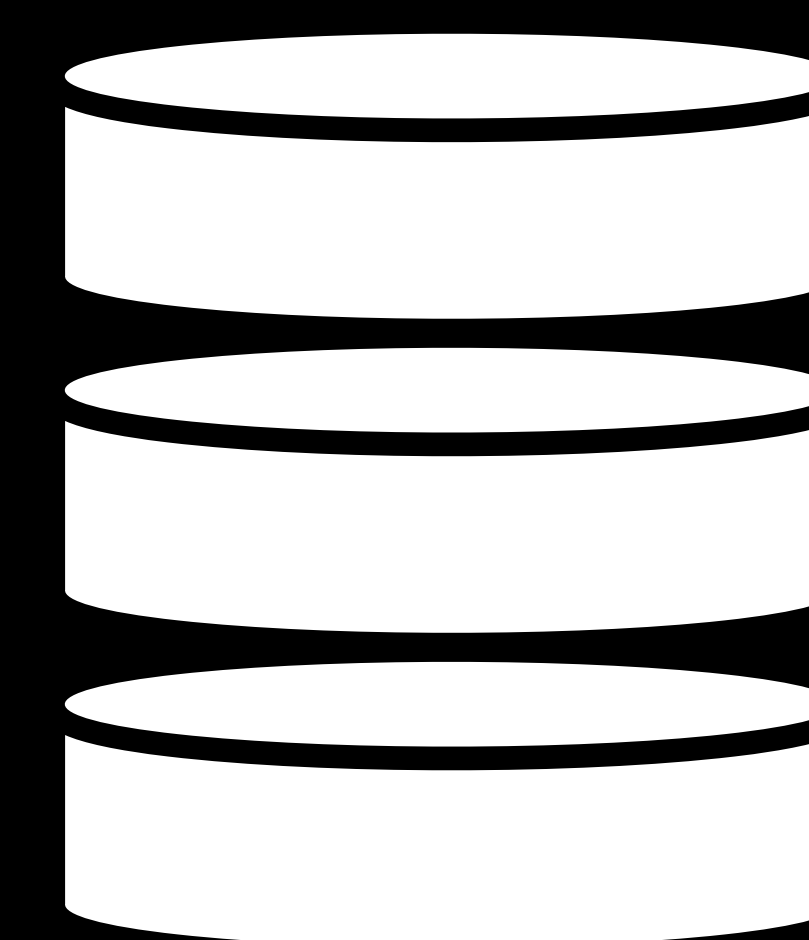
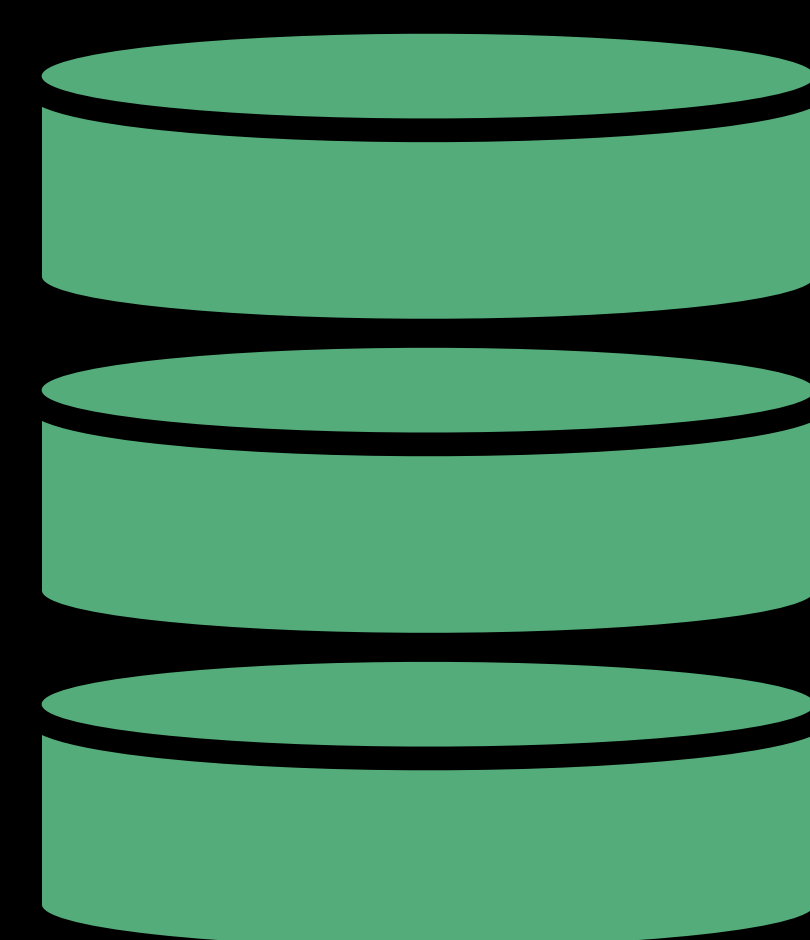
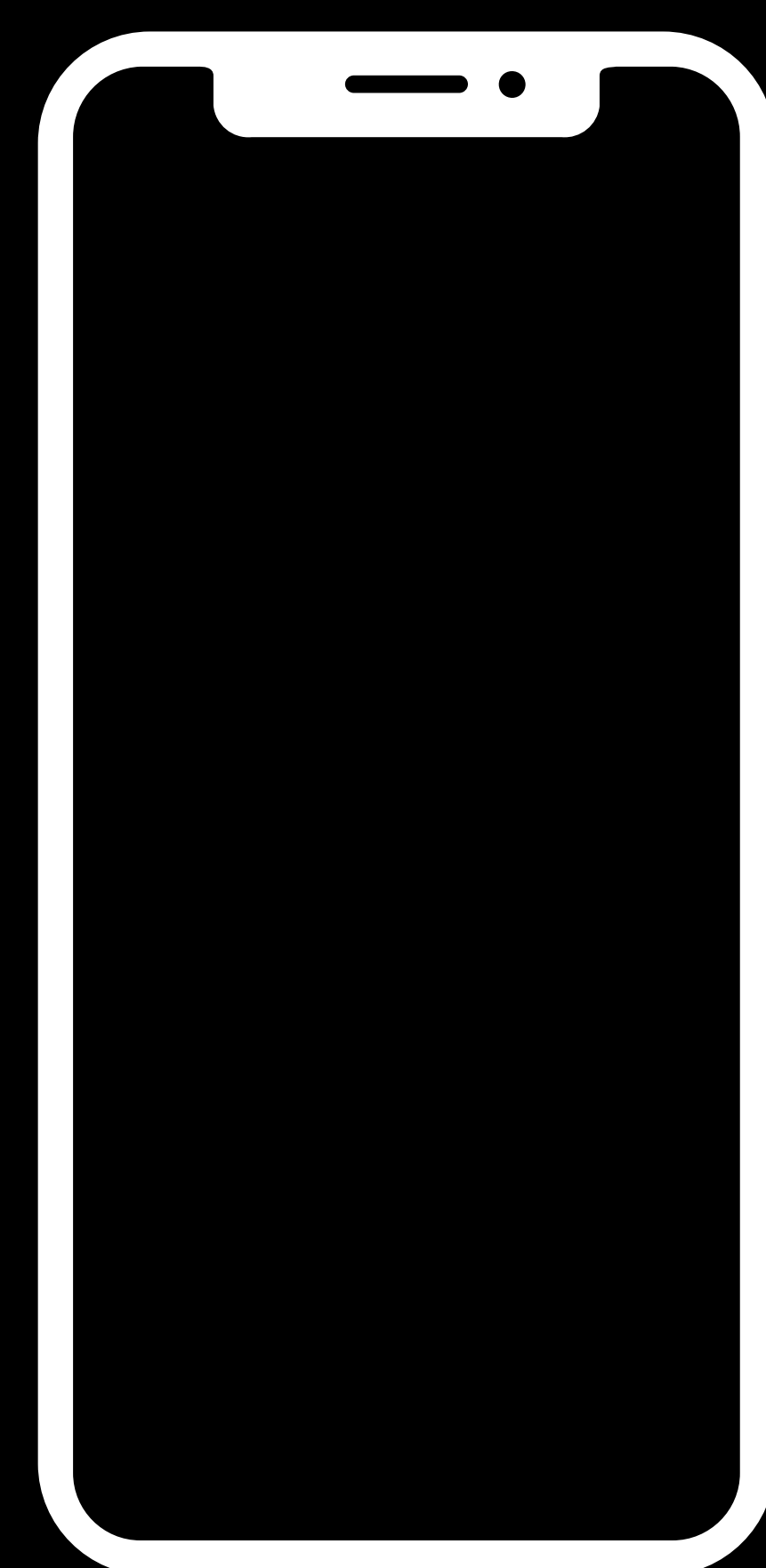
NSPersistentCloudKitContainer — Multiple Stores



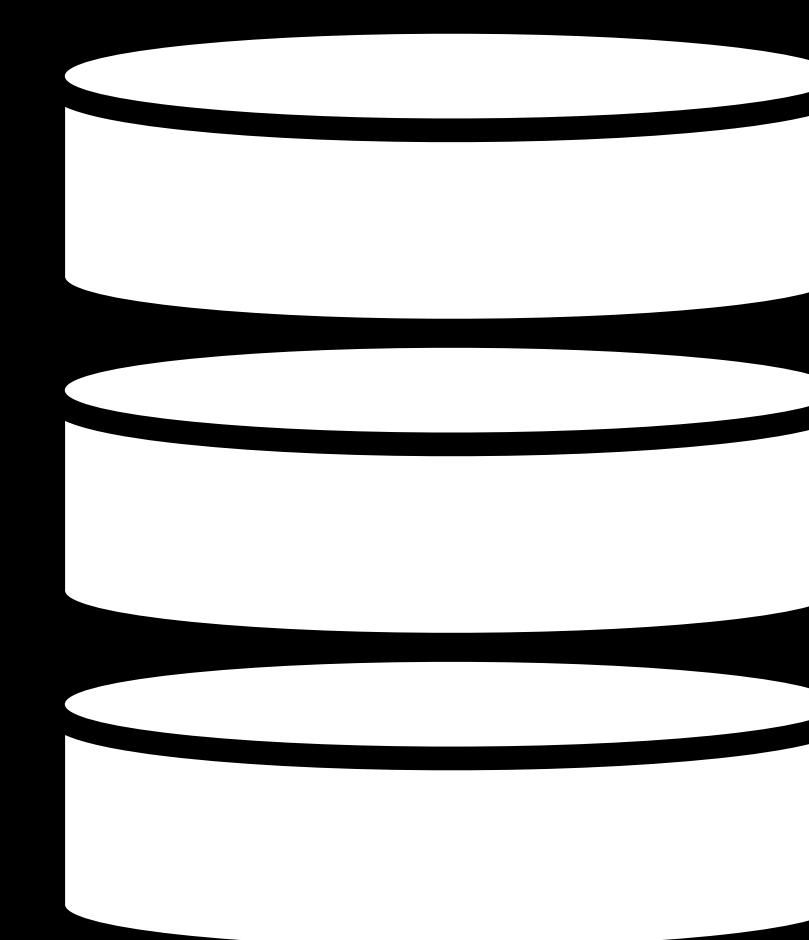
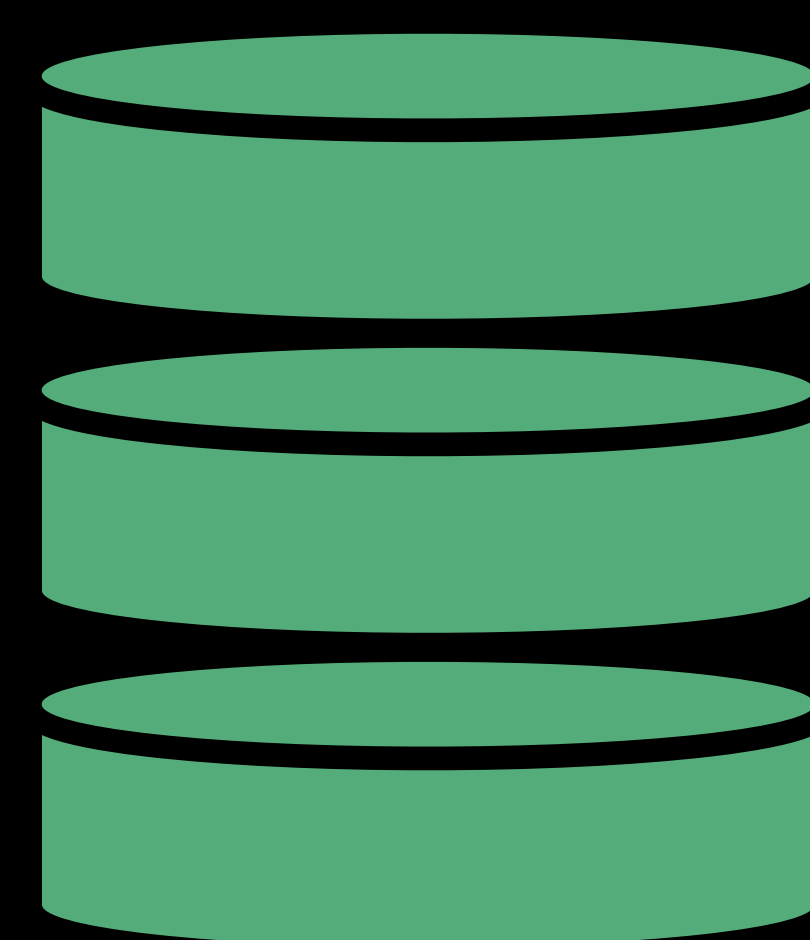
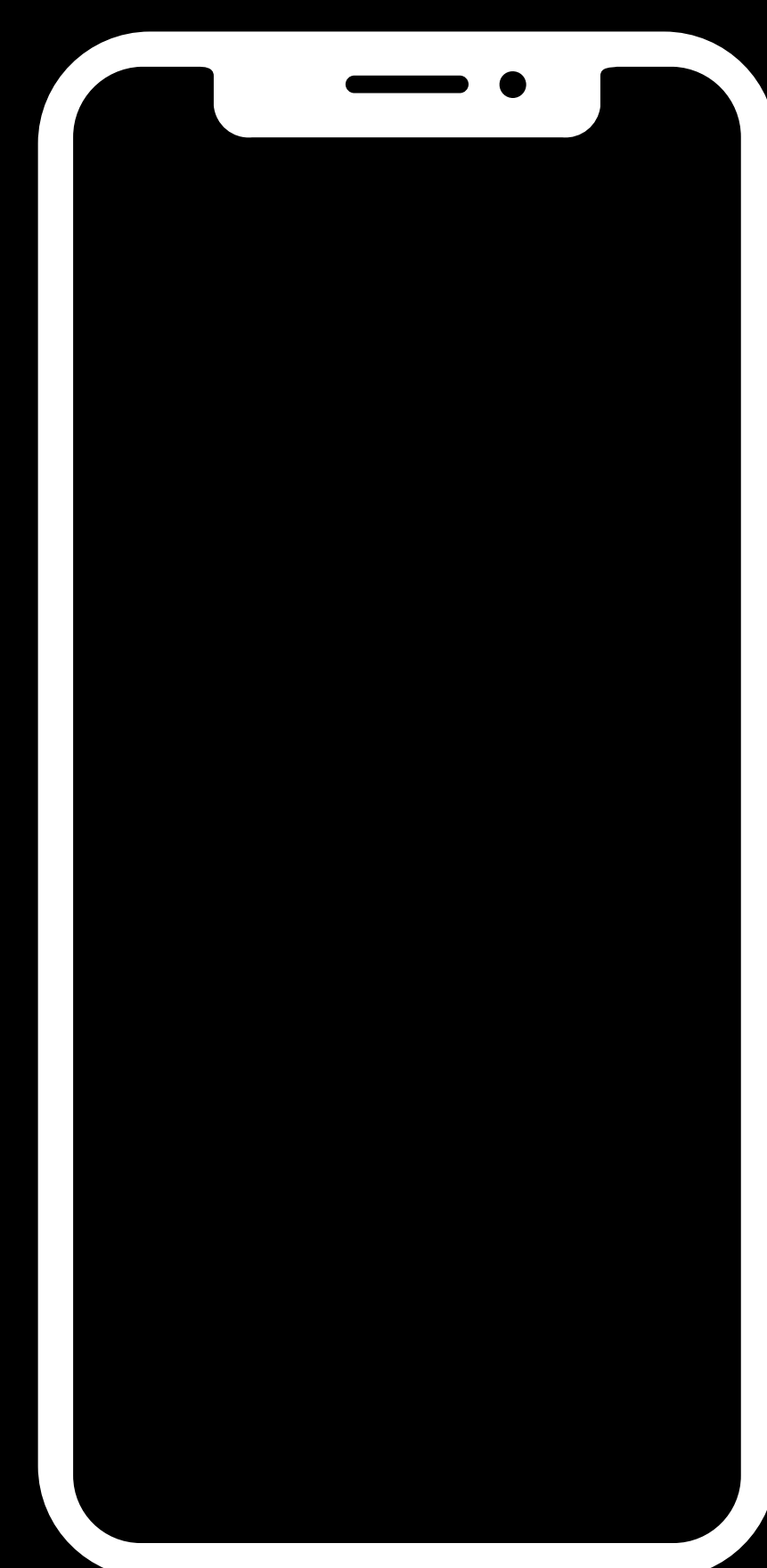
NSPersistentCloudKitContainer — Multiple Stores



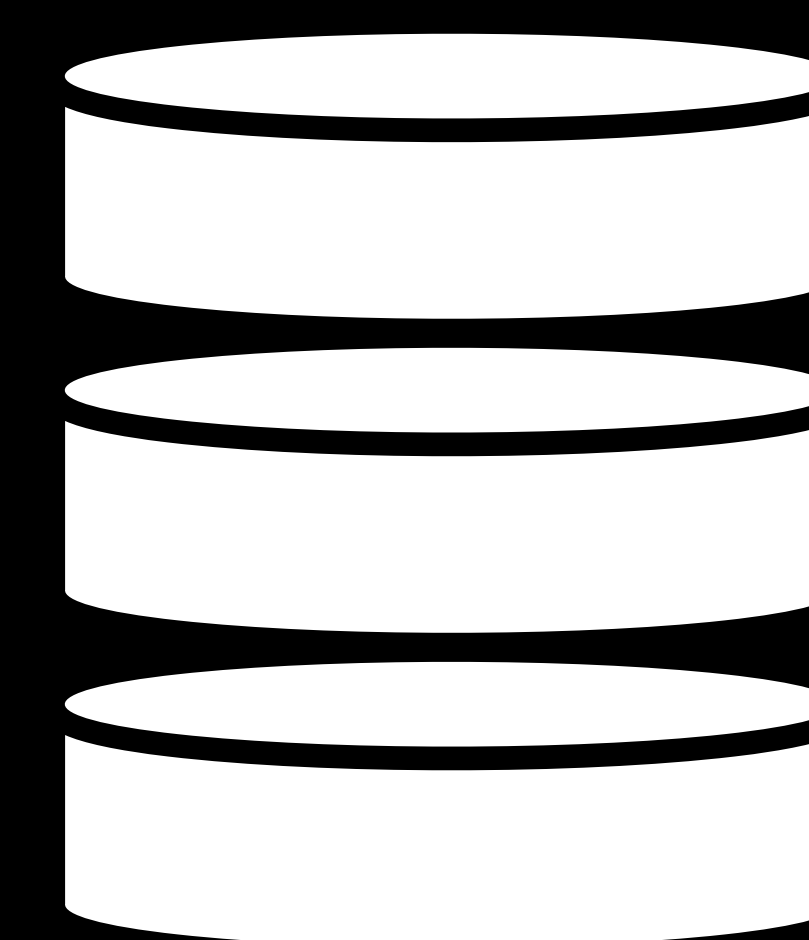
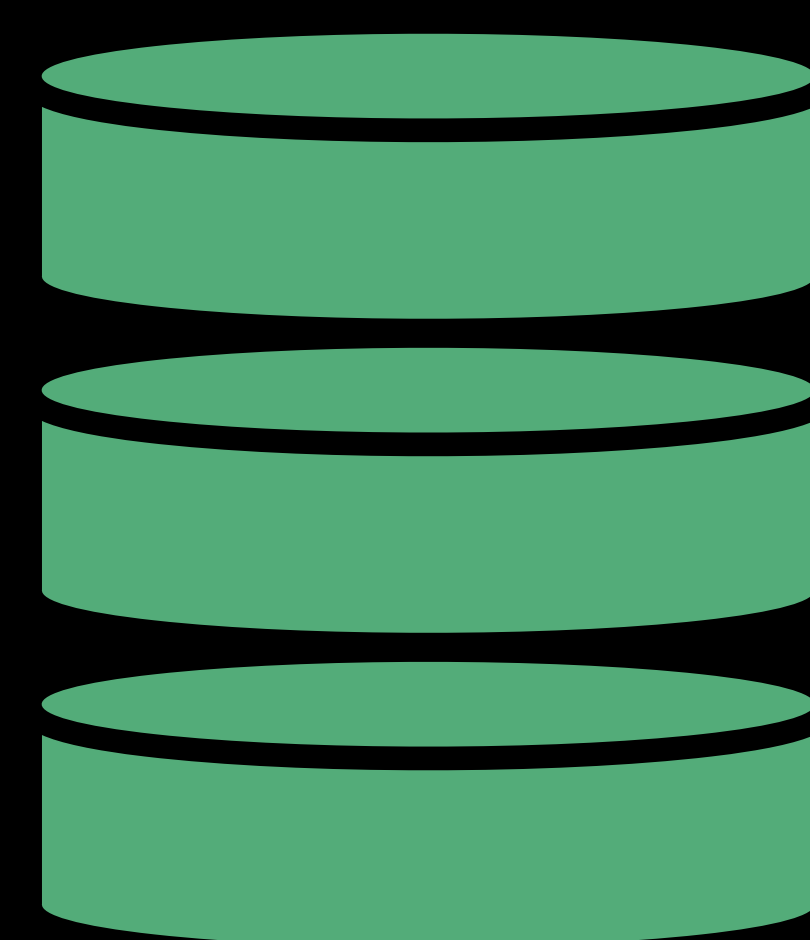
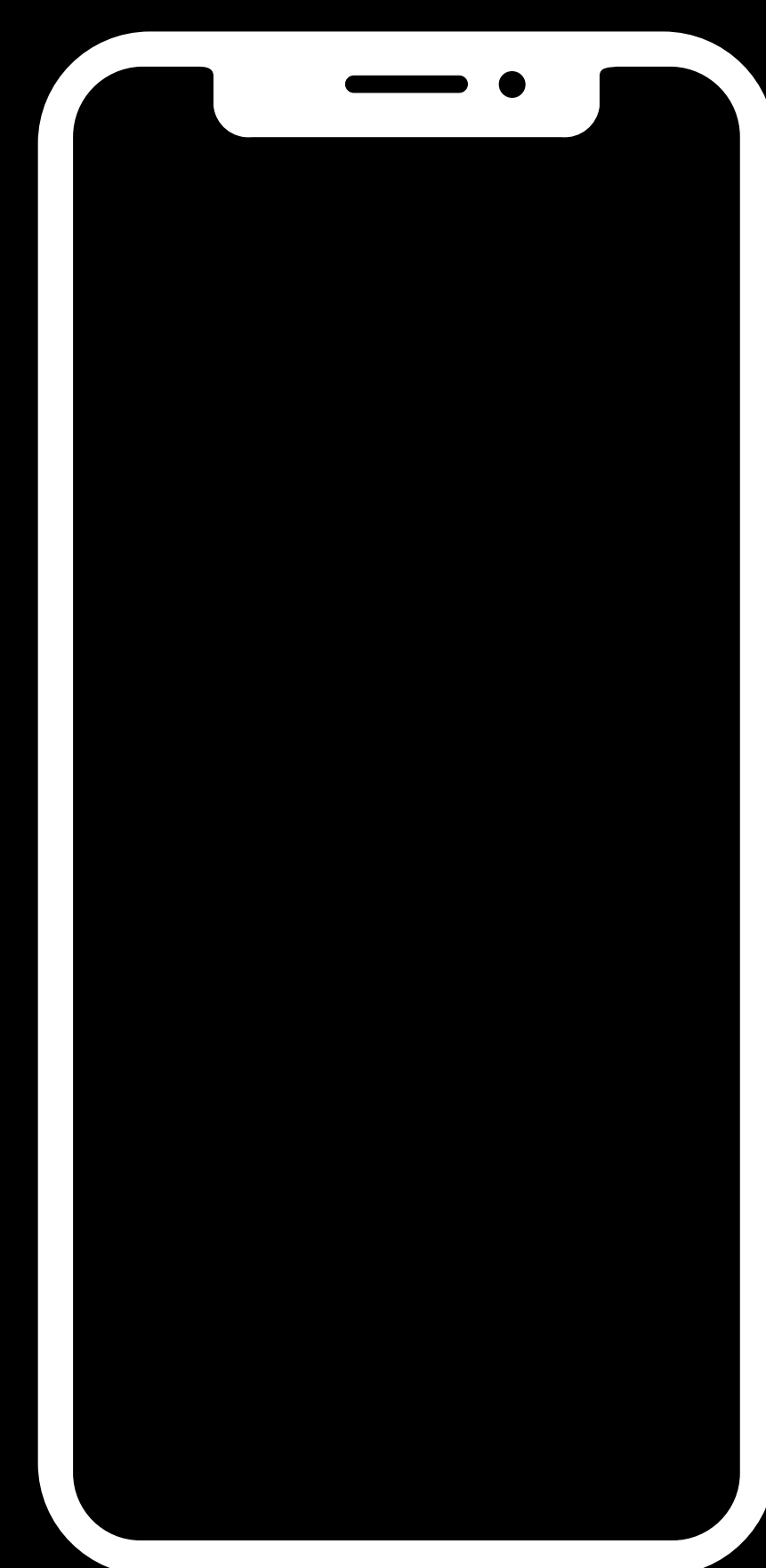
NSPersistentCloudKitContainer — Multiple Stores



NSPersistentCloudKitContainer — Multiple Stores



NSPersistentCloudKitContainer — Multiple Stores



ENTITIES

- Attachment
- ImageData
- Post
- Tag

FETCH REQUESTS

CONFIGURATIONS

- Default

Attributes

Attribute	Type

+ -

Relationships

Relationship	Destination	Inverse

+ -

Fetches Properties

Fetches Property	Predicate

+ -

ENTITIES

- Attachment
- ImageData
- Post
- Tag

FETCH REQUESTS

CONFIGURATIONS

- Default

Attributes

Attribute	Type

+ -

Relationships

Relationship	Destination	Inverse

+ -

Fetches Properties

Fetches Property	Predicate

+ -

ENTITIES

- Attachment
- ImageData
- Location**
- Post
- Tag

FETCH REQUESTS

CONFIGURATIONS

- Default

Attributes

Attribute	Type
longitude	Double
latitude	Double

Relationships

Relationship	Destination	Inverse
--------------	-------------	---------

Fetches Properties

Fetches Property	Predicate
------------------	-----------

ENTITIES

- Attachment
- ImageData
- Location**
- Post
- Tag

FETCH REQUESTS

CONFIGURATIONS

- Default

Attributes

Attribute	Type
longitude	Double
latitude	Double

Relationships

Relationship	Destination	Inverse
--------------	-------------	---------

Fetches Properties

Fetches Property	Predicate
------------------	-----------

ENTITIES

- Attachment
- ImageData
- Location
- Post
- Tag

FETCH REQUESTS

CONFIGURATIONS

- Default
- Cloud**

Entities

Entity	Abstract	Class
Attachment	<input type="checkbox"/>	Attachment
ImageData	<input type="checkbox"/>	ImageData
Post	<input type="checkbox"/>	Post
Tag	<input type="checkbox"/>	Tag

ENTITIES

- Attachment
- ImageData
- Location
- Post
- Tag

FETCH REQUESTS

CONFIGURATIONS

- Default
- Cloud**

Entities

Entity	Abstract	Class
Attachment	<input type="checkbox"/>	Attachment
ImageData	<input type="checkbox"/>	ImageData
Post	<input type="checkbox"/>	Post
Tag	<input type="checkbox"/>	Tag


```
let container = NSPersistentCloudKitContainer(name: "CloudKitContainer")

let local = NSPersistentStoreDescription(url: URL(fileURLWithPath: "/files/local.sqlite"))
local.configuration = "Local"

let cloud = NSPersistentStoreDescription(url: URL(fileURLWithPath: "/files/cloud.sqlite"))
cloud.configuration = "Cloud"
cloud.cloudKitContainerOptions = NSPersistentCloudKitContainerOptions(containerIdentifier:
    "iCloud.com.wwdc.demo")

container.persistentStoreDescriptions = [ local, cloud ]
```

```
let container = NSPersistentCloudKitContainer(name: "CloudKitContainer")
```

```
let local = NSPersistentStoreDescription(url: URL(fileURLWithPath: "/files/local.sqlite"))  
local.configuration = "Local"
```

```
let cloud = NSPersistentStoreDescription(url: URL(fileURLWithPath: "/files/cloud.sqlite"))  
cloud.configuration = "Cloud"  
cloud.cloudKitContainerOptions = NSPersistentCloudKitContainerOptions(containerIdentifier:  
"iCloud.com.wwdc.demo")
```

```
container.persistentStoreDescriptions = [ local, cloud ]
```

```
let container = NSPersistentCloudKitContainer(name: "CloudKitContainer")
```

```
let local = NSPersistentStoreDescription(url: URL(fileURLWithPath: "/files/local.sqlite"))  
local.configuration = "Local"
```

```
let cloud = NSPersistentStoreDescription(url: URL(fileURLWithPath: "/files/cloud.sqlite"))  
cloud.configuration = "Cloud"  
cloud.cloudKitContainerOptions = NSPersistentCloudKitContainerOptions(containerIdentifier:  
"iCloud.com.wwdc.demo")
```

```
container.persistentStoreDescriptions = [ local, cloud ]
```



```
let container = NSPersistentCloudKitContainer(name: "CloudKitContainer")
```

```
let local = NSPersistentStoreDescription(url: URL(fileURLWithPath: "/files/local.sqlite"))
```

```
local.configuration = "Local"
```

```
let cloud = NSPersistentStoreDescription(url: URL(fileURLWithPath: "/files/cloud.sqlite"))
```

```
cloud.configuration = "Cloud"
```

```
cloud.cloudKitContainerOptions = NSPersistentCloudKitContainerOptions(containerIdentifier:  
"iCloud.com.wwdc.demo")
```

```
container.persistentStoreDescriptions = [ local, cloud ]
```

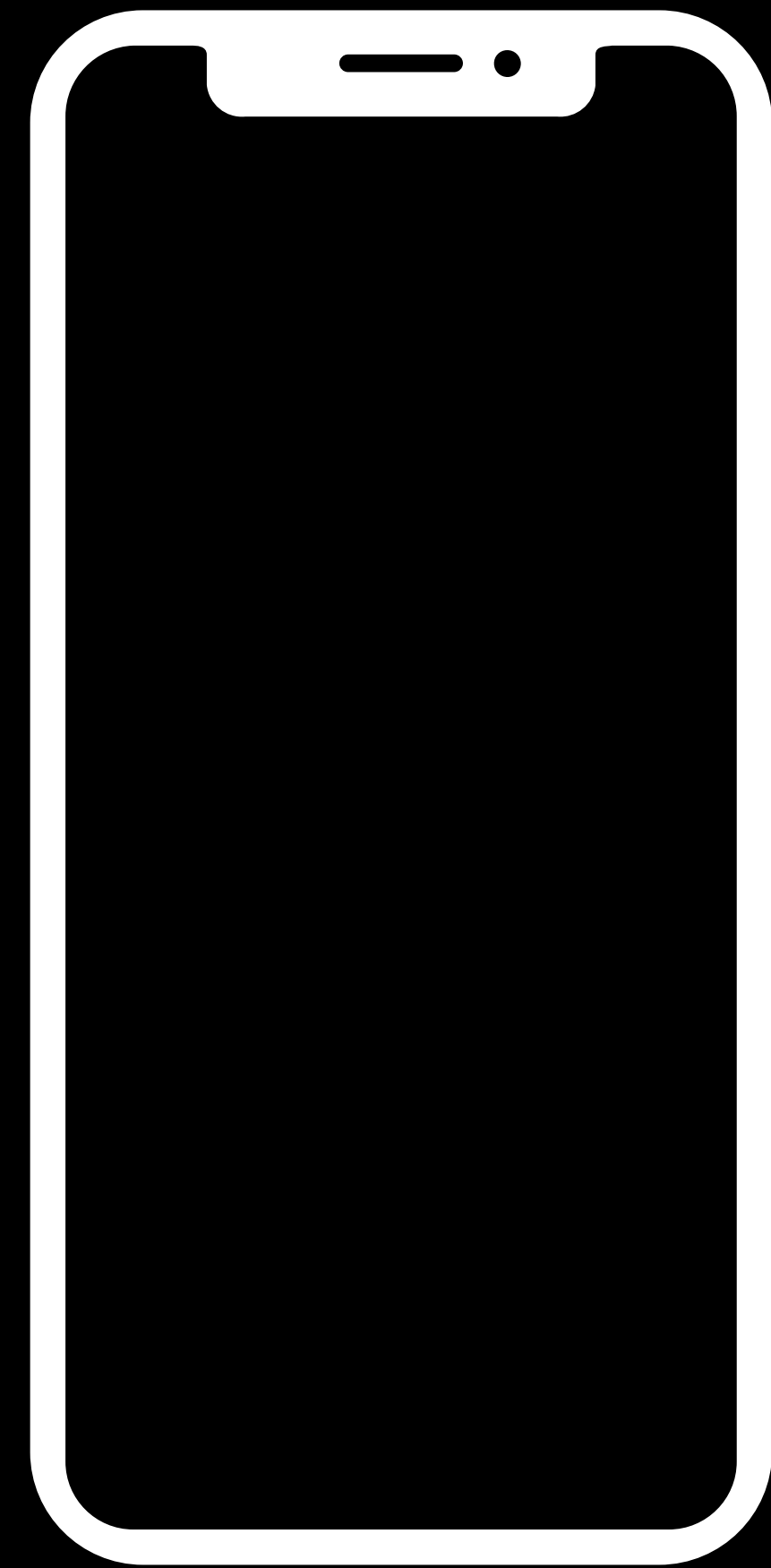
```
let container = NSPersistentCloudKitContainer(name: "CloudKitContainer")

let local = NSPersistentStoreDescription(url: URL(fileURLWithPath: "/files/local.sqlite"))
local.configuration = "Local"

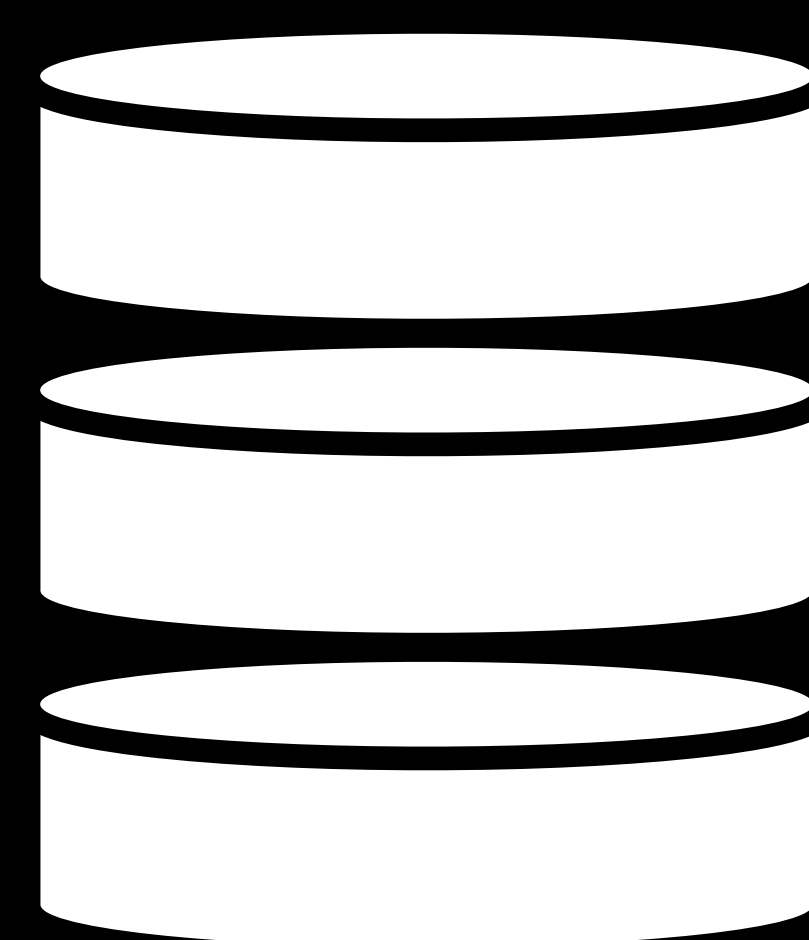
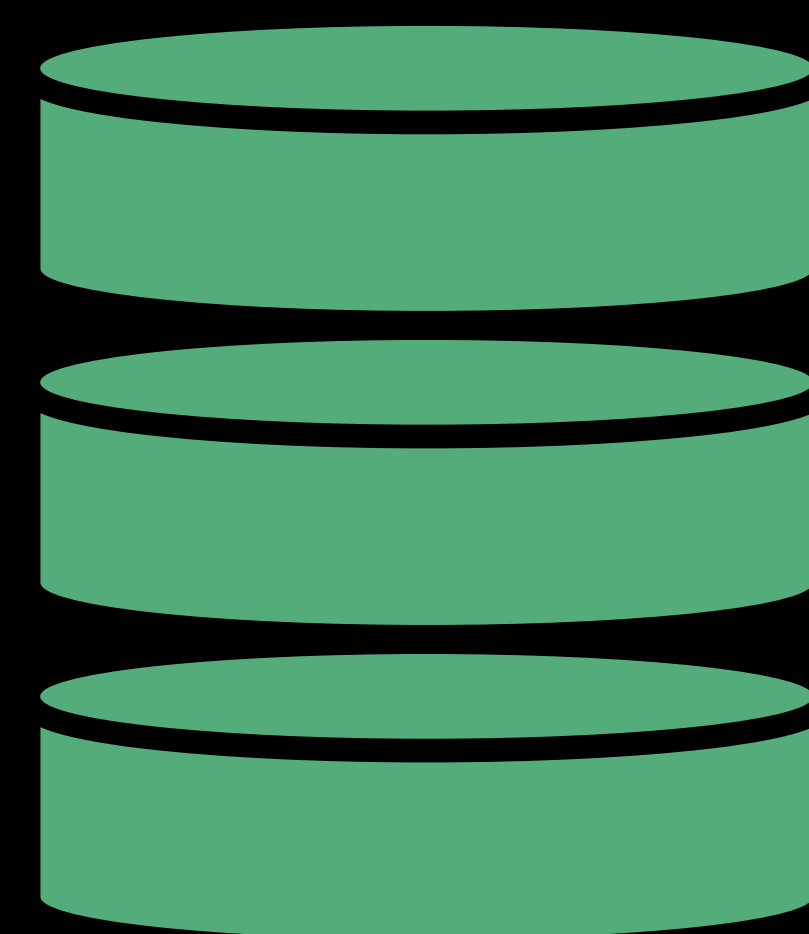
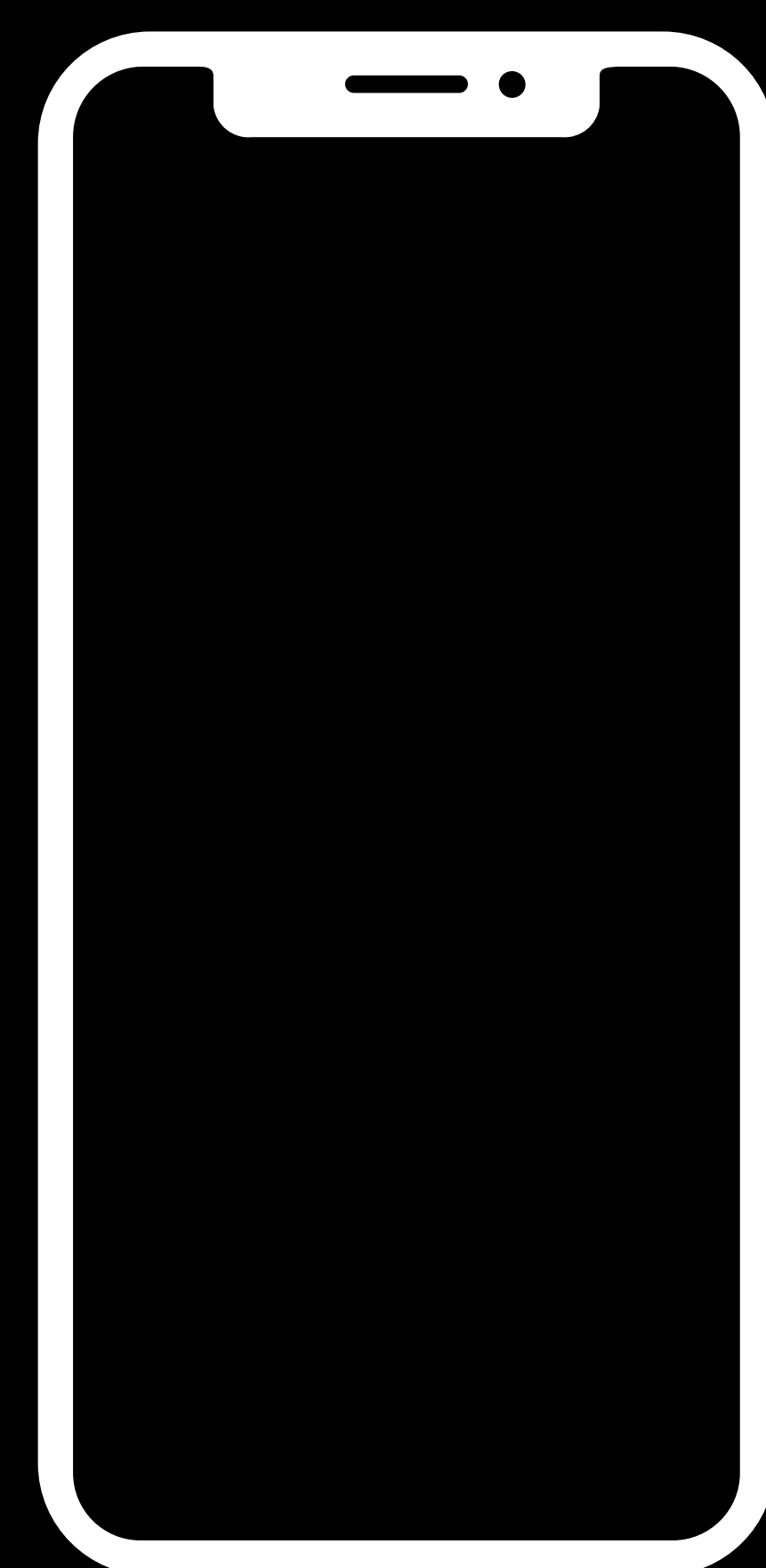
let cloud = NSPersistentStoreDescription(url: URL(fileURLWithPath: "/files/cloud.sqlite"))
cloud.configuration = "Cloud"
cloud.cloudKitContainerOptions = NSPersistentCloudKitContainerOptions(containerIdentifier:
    "iCloud.com.wwdc.demo")

container.persistentStoreDescriptions = [ local, cloud ]
```

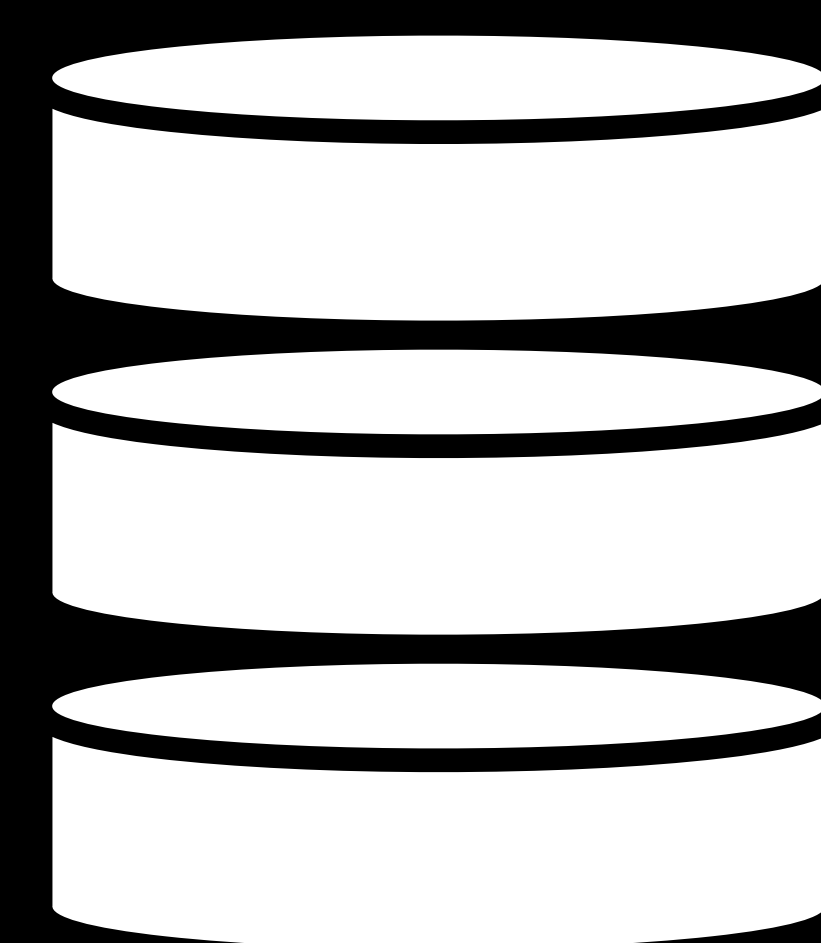
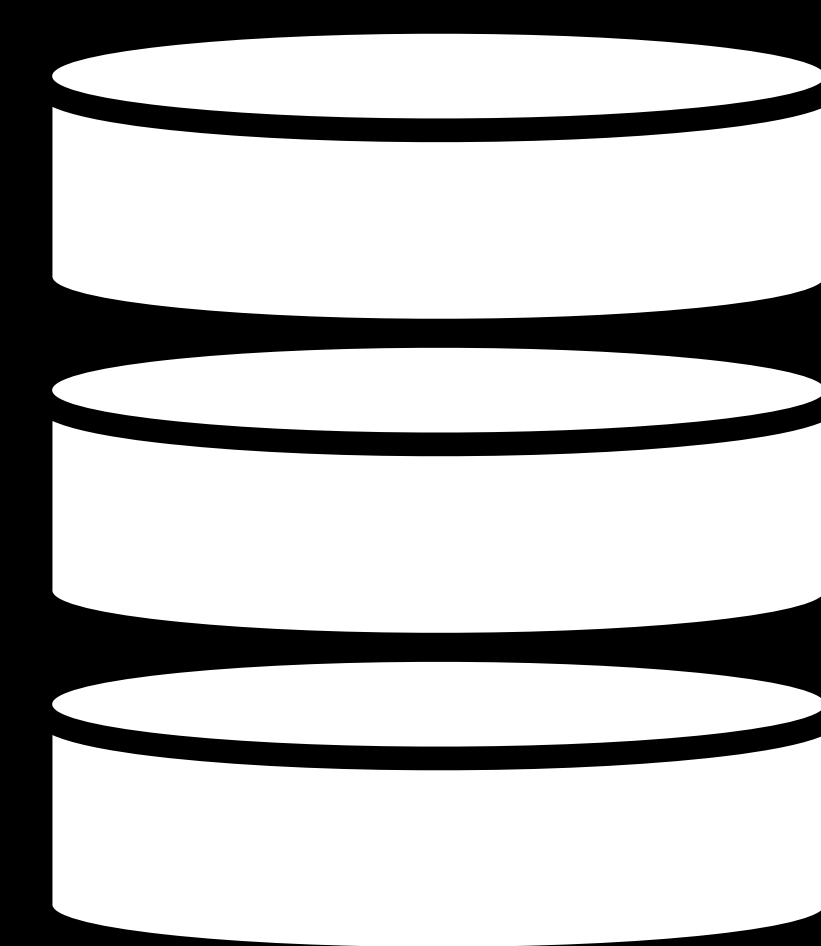
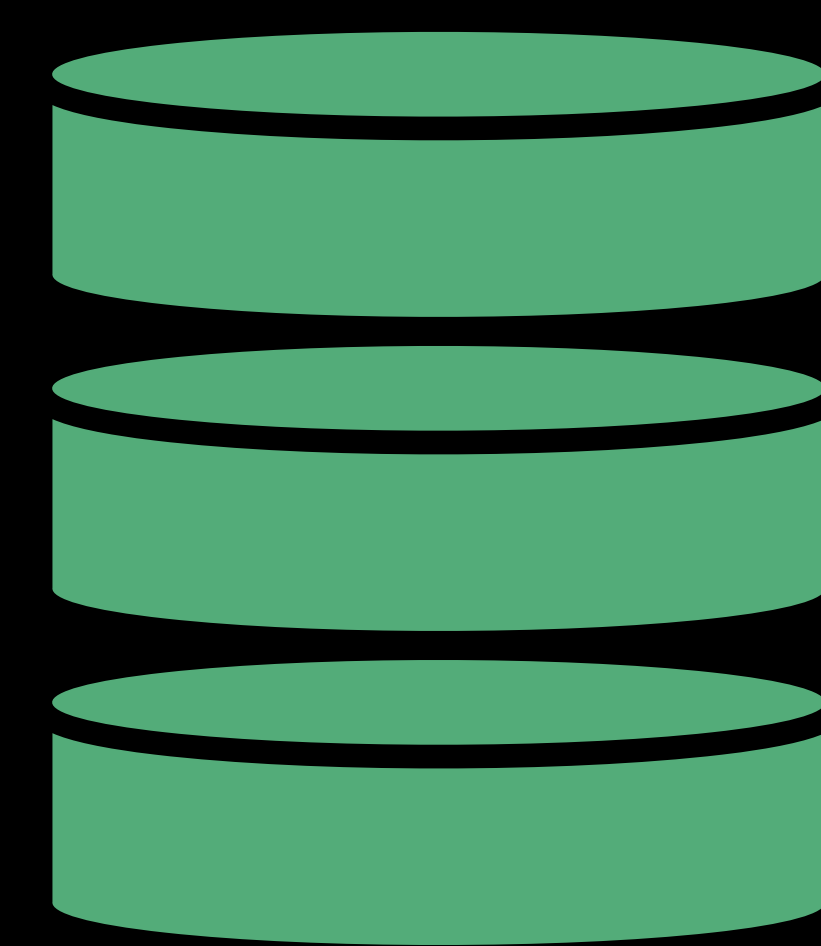
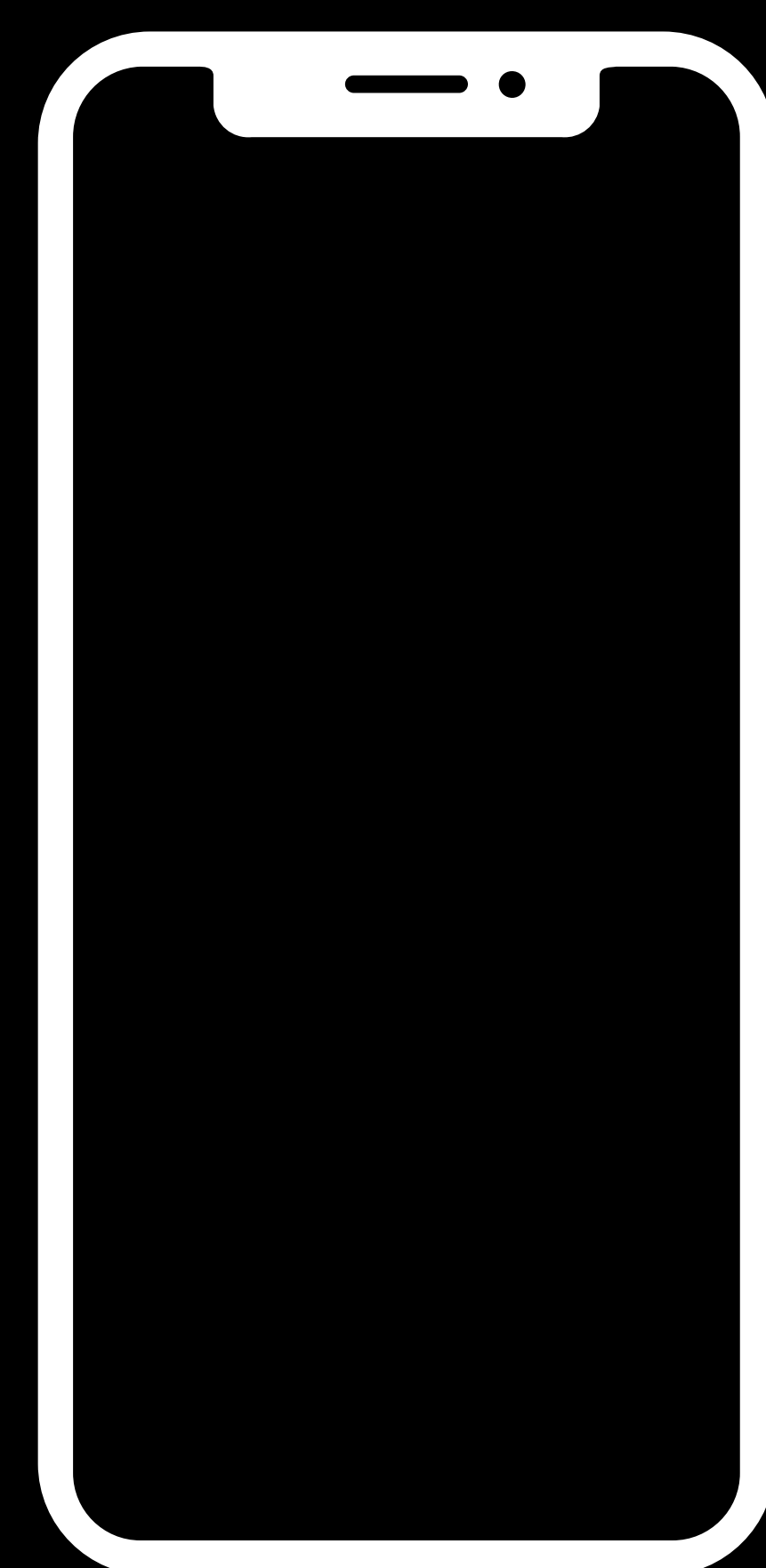

NSPersistentCloudKitContainer — Multiple Stores



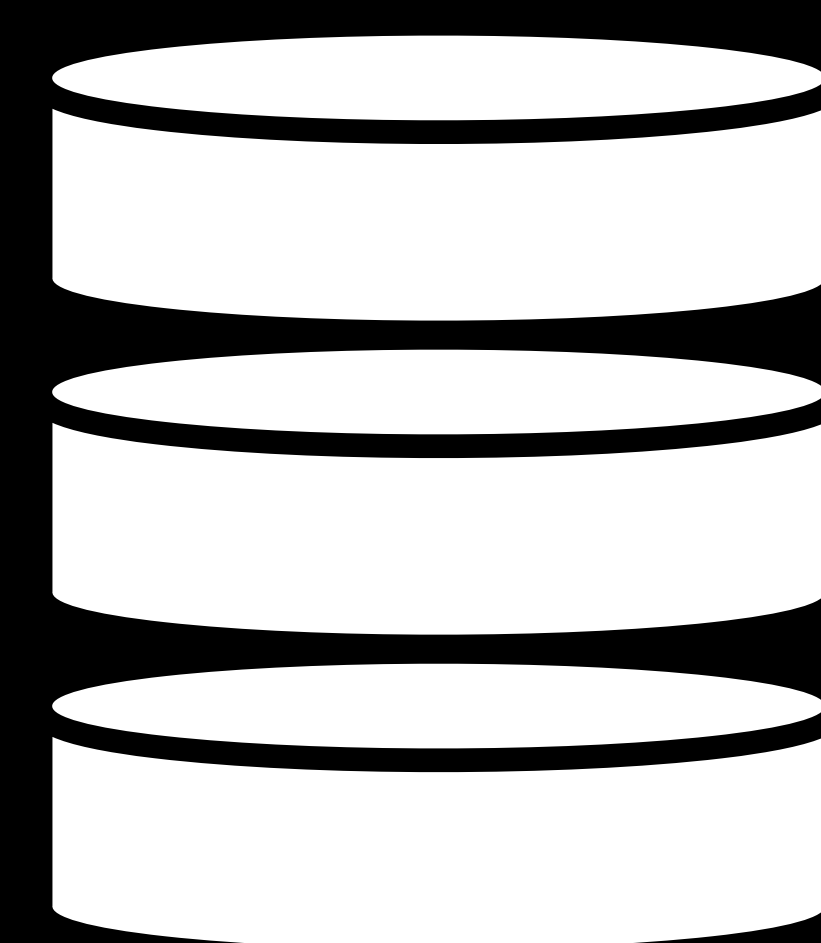
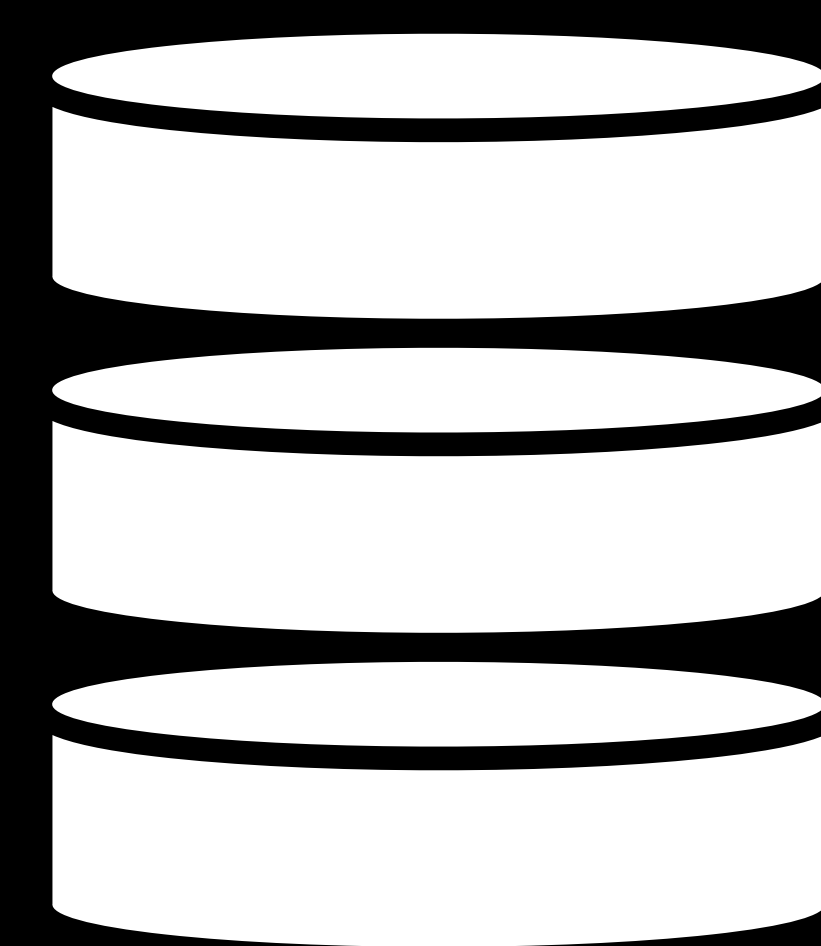
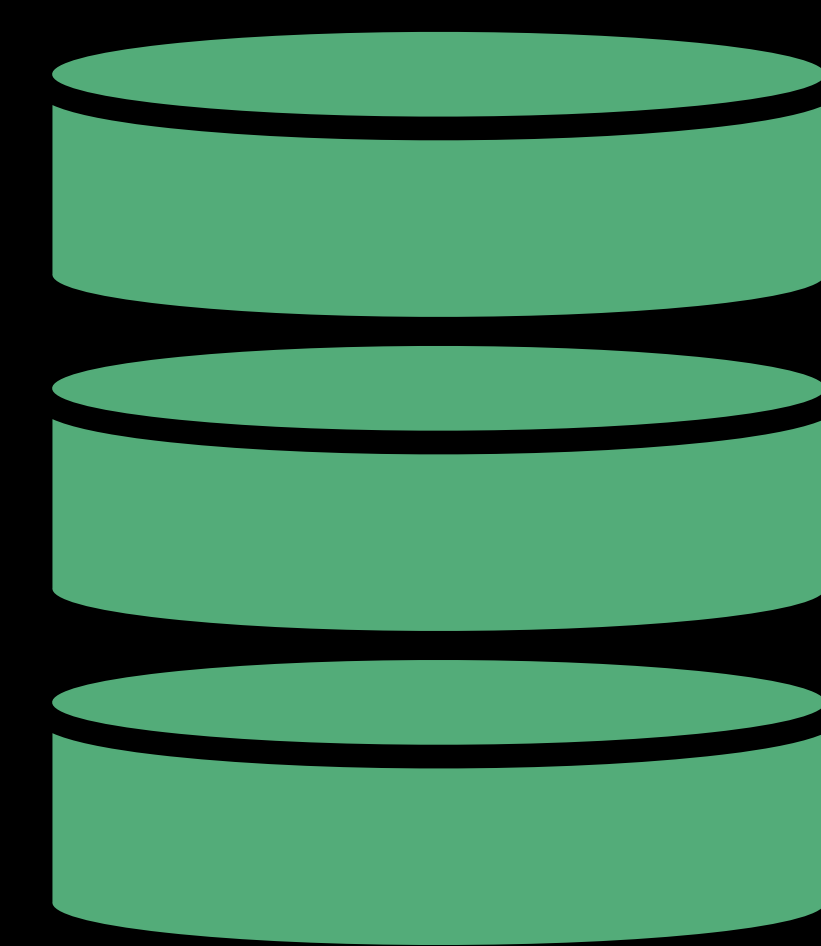
NSPersistentCloudKitContainer — Multiple Stores



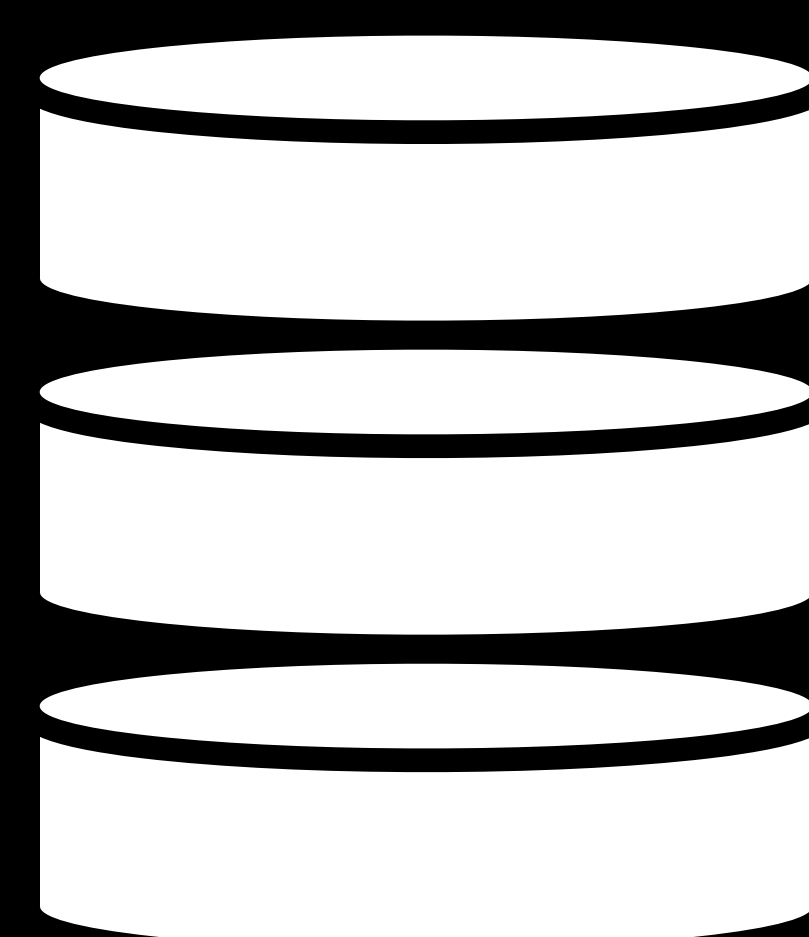
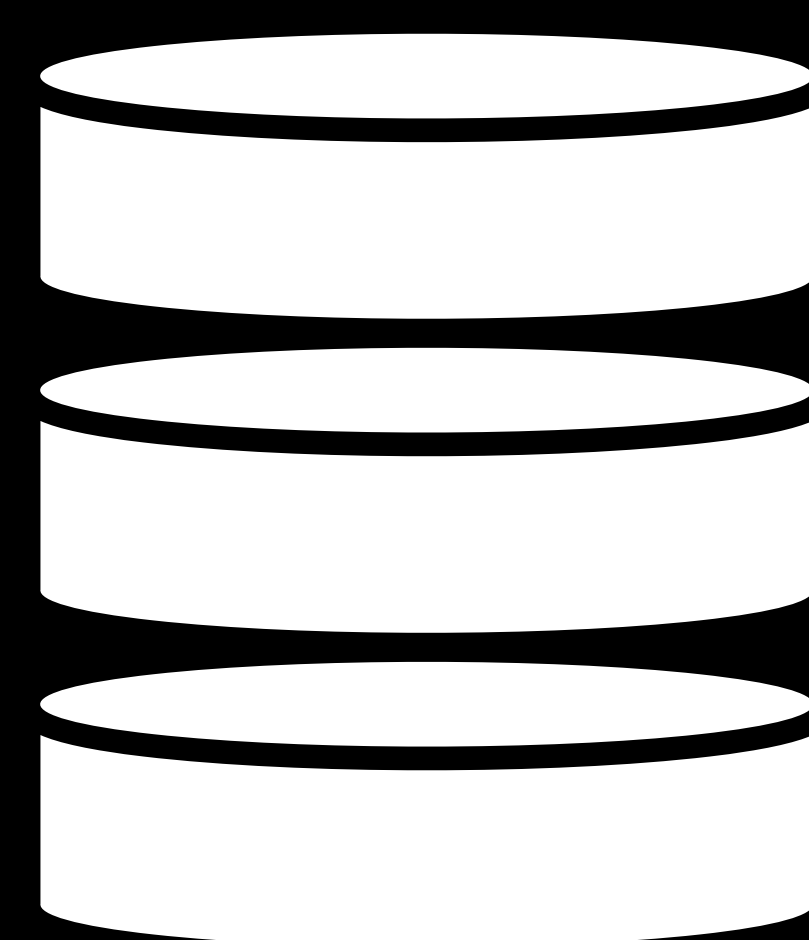
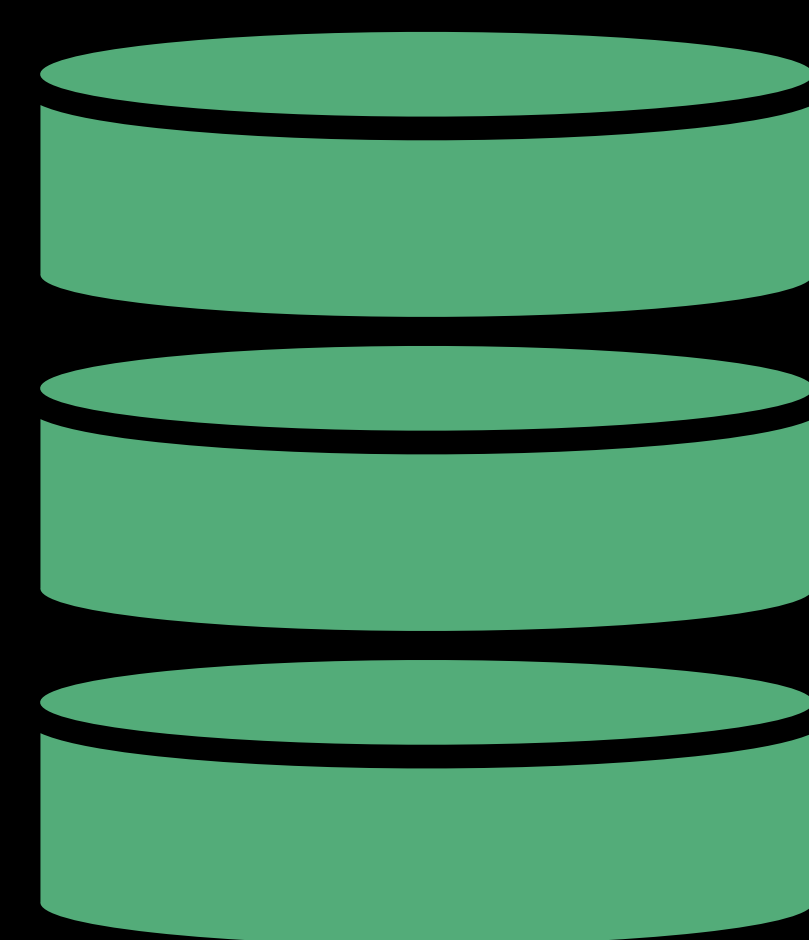
NSPersistentCloudKitContainer — Multiple Stores



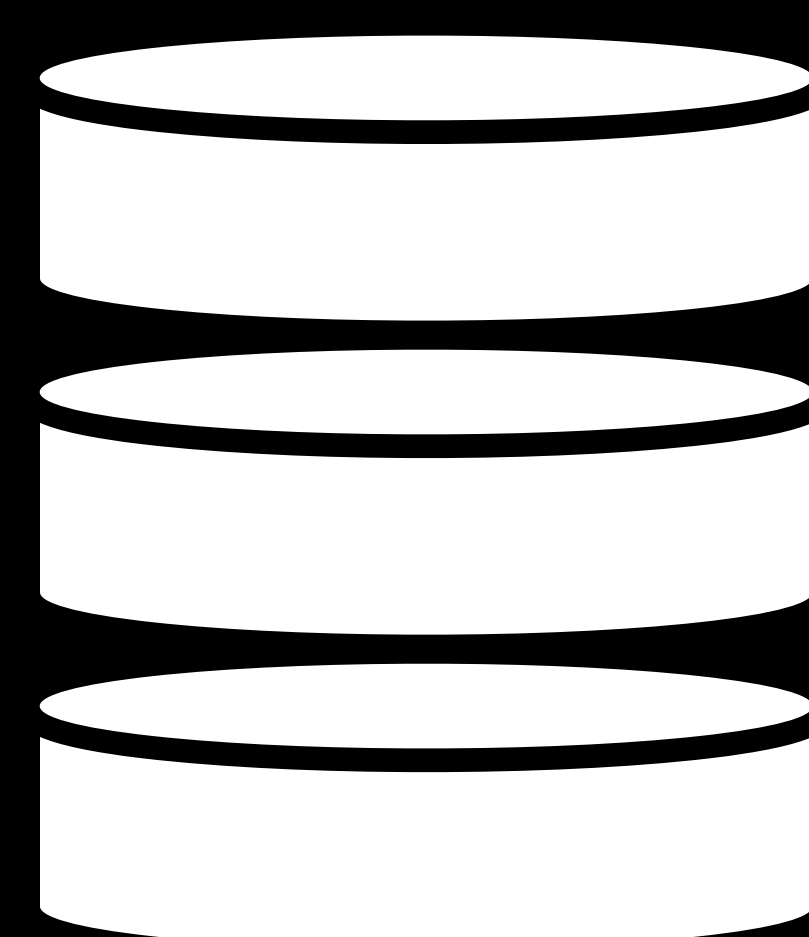
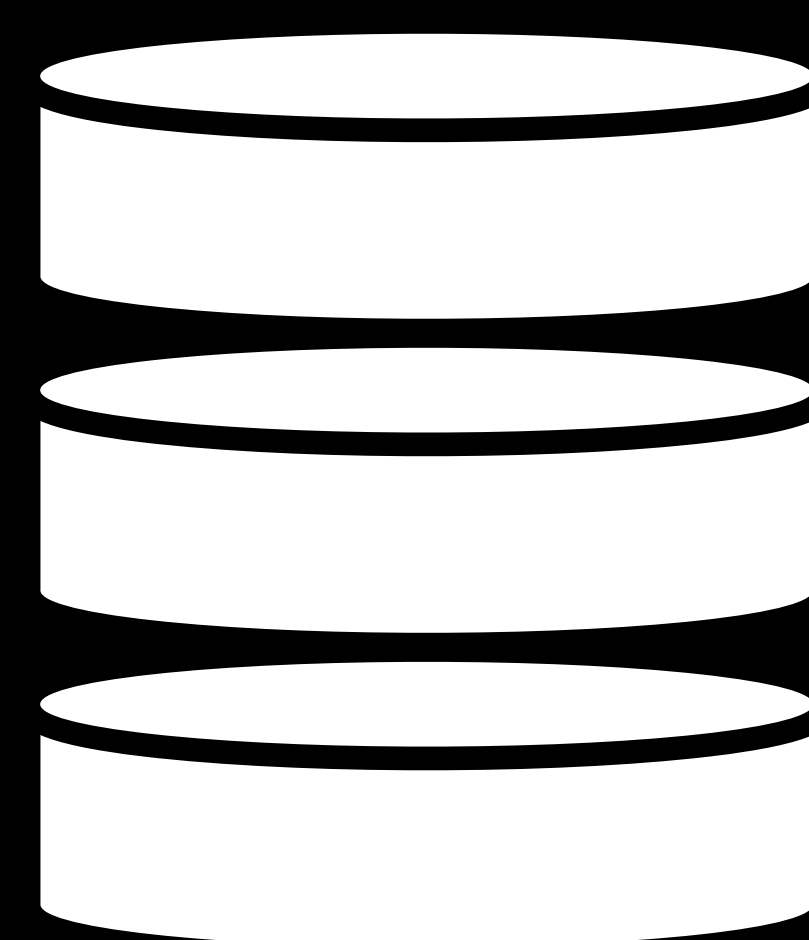
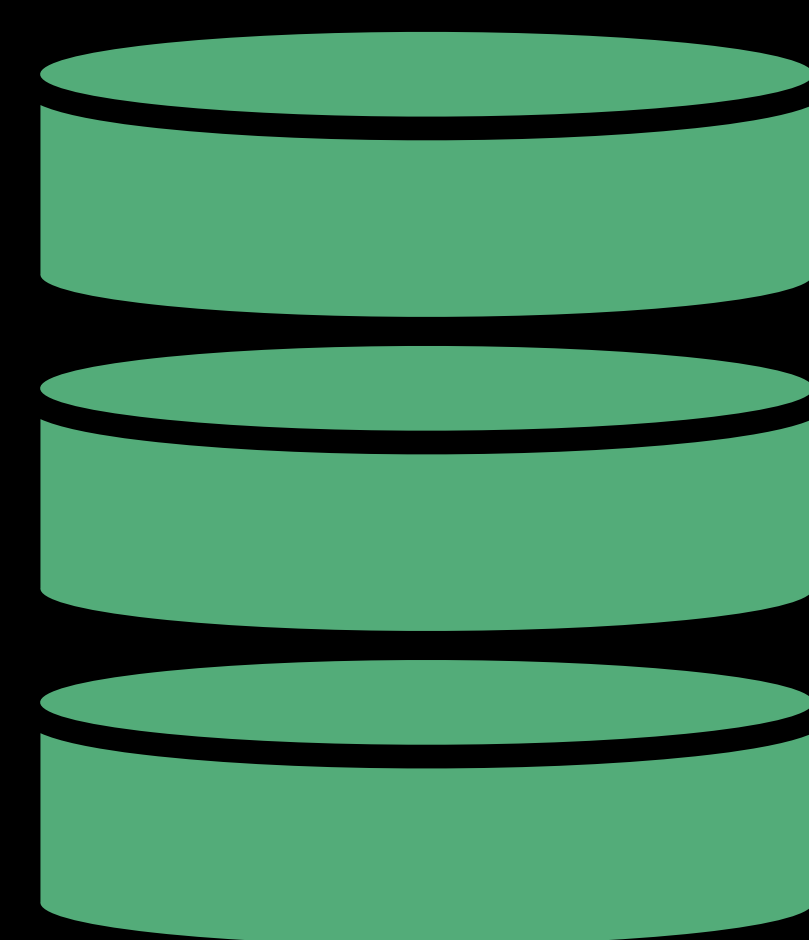
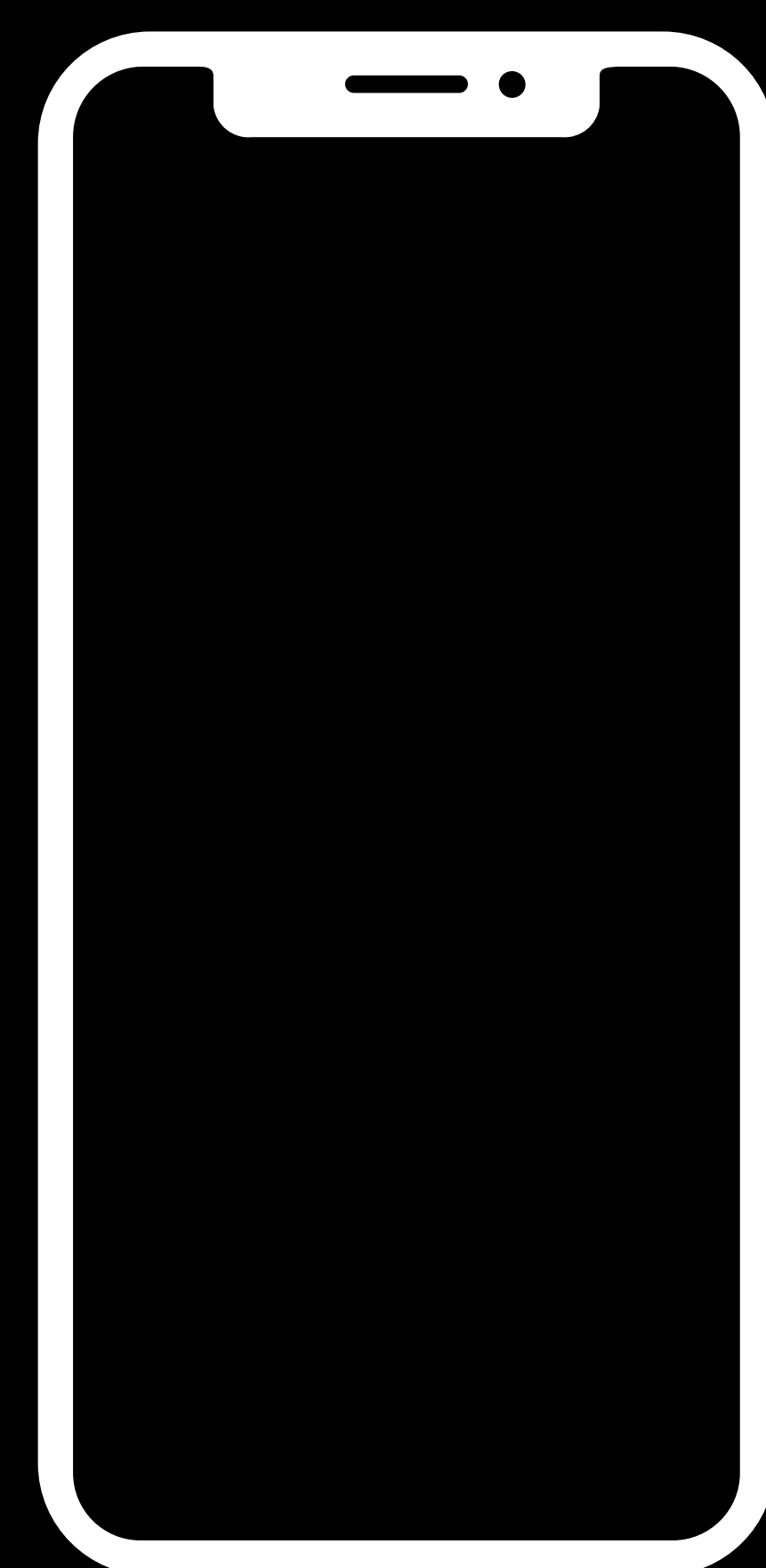
NSPersistentCloudKitContainer — Multiple Stores



NSPersistentCloudKitContainer — Multiple Stores



NSPersistentCloudKitContainer — Multiple Stores




```
let container = NSPersistentCloudKitContainer(name: "CloudKitContainer")

let local = NSPersistentStoreDescription(url: URL(fileURLWithPath: "/files/local.sqlite"))
local.configuration = "Local"

let cloud = NSPersistentStoreDescription(url: URL(fileURLWithPath: "/files/cloud.sqlite"))
cloud.configuration = "Cloud"
cloud.cloudKitContainerOptions = NSPersistentCloudKitContainerOptions(containerIdentifier:
    "iCloud.com.wwdc.demo")

let shared = NSPersistentStoreDescription(url: URL(fileURLWithPath: "/files/shared.sqlite"))
shared.configuration = "Shared"
shared.cloudKitContainerOptions = NSPersistentCloudKitContainerOptions(containerIdentifier:
    "iCloud.com.wwdc.shared")

container.persistentStoreDescriptions = [ local, cloud, shared ]
```

```
let container = NSPersistentCloudKitContainer(name: "CloudKitContainer")

let local = NSPersistentStoreDescription(url: URL(fileURLWithPath: "/files/local.sqlite"))
local.configuration = "Local"

let cloud = NSPersistentStoreDescription(url: URL(fileURLWithPath: "/files/cloud.sqlite"))
cloud.configuration = "Cloud"
cloud.cloudKitContainerOptions = NSPersistentCloudKitContainerOptions(containerIdentifier:
    "iCloud.com.wwdc.demo")

let shared = NSPersistentStoreDescription(url: URL(fileURLWithPath: "/files/shared.sqlite"))
shared.configuration = "Shared"
shared.cloudKitContainerOptions = NSPersistentCloudKitContainerOptions(containerIdentifier:
    "iCloud.com.wwdc.shared")

container.persistentStoreDescriptions = [ local, cloud, shared ]
```



```
let container = NSPersistentCloudKitContainer(name: "CloudKitContainer")

let local = NSPersistentStoreDescription(url: URL(fileURLWithPath: "/files/local.sqlite"))
local.configuration = "Local"

let cloud = NSPersistentStoreDescription(url: URL(fileURLWithPath: "/files/cloud.sqlite"))
cloud.configuration = "Cloud"
cloud.cloudKitContainerOptions = NSPersistentCloudKitContainerOptions(containerIdentifier:
    "iCloud.com.wwdc.demo")

let shared = NSPersistentStoreDescription(url: URL(fileURLWithPath: "/files/shared.sqlite"))
shared.configuration = "Shared"
shared.cloudKitContainerOptions = NSPersistentCloudKitContainerOptions(containerIdentifier:
    "iCloud.com.wwdc.shared")

container.persistentStoreDescriptions = [ local, cloud, shared ]
```

NSPersistentCloudKitContainer's Schema

NSPersistentCloudKitContainer's Schema

Record types and entity names

NSPersistentCloudKitContainer's Schema

Record types and entity names

Asset externalization

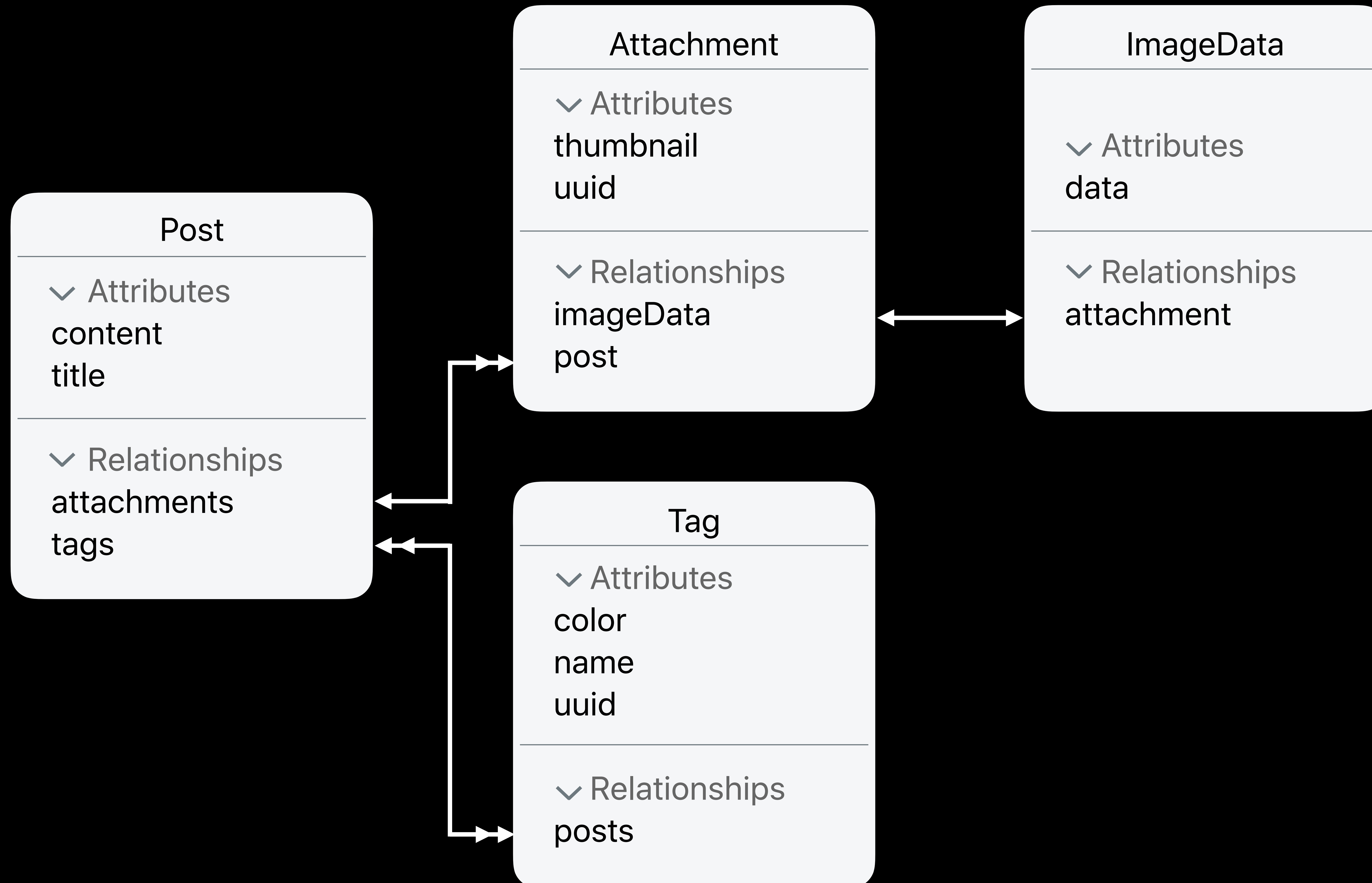
NSPersistentCloudKitContainer's Schema

Record types and entity names

Asset externalization

Relationships

Sample Application Data Model



```
@objc(Post)
public class Post: NSManagedObject {
    @NSManaged public var title: String?
    @NSManaged public var content: String?
    @NSManaged public var attachments: NSSet?
    @NSManaged public var tags: NSSet?
}
```

```
@objc(Post)
public class Post: NSManagedObject {
    @NSManaged public var title: String?
    @NSManaged public var content: String?
    @NSManaged public var attachments: NSSet?
    @NSManaged public var tags: NSSet?
}
```



```
@objc(Post)
```

```
public class Post: NSManagedObject {  
    @NSManaged public var title: String?  
    @NSManaged public var content: String?  
    @NSManaged public var attachments: NSSet?  
    @NSManaged public var tags: NSSet?  
}
```

```
<CKRecord: 0x7fd0b2539220; recordID=Post_UUID, values={  
    "CD_content" = "An example core data string";  
    "CD_content_ckAsset" = "<CKAsset: 0x7fd0b2515c20;...>";  
    "CD_title" = "An example core data string";  
    "CD_title_ckAsset" = "<CKAsset: 0x7fd0b240bd40;...>";  
    "CD_entityName" = Post;  
}, recordType=CD_Post>
```

```
@objc(Post)
```

```
public class Post: NSManagedObject {  
    @NSManaged public var title: String?  
    @NSManaged public var content: String?  
    @NSManaged public var attachments: NSSet?  
    @NSManaged public var tags: NSSet?  
}
```

```
<CKRecord: 0x7fd0b2539220; recordID=Post_UUID, values={  
    "CD_content" = "An example core data string";  
    "CD_content_ckAsset" = "<CKAsset: 0x7fd0b2515c20;...>";  
    "CD_title" = "An example core data string";  
    "CD_title_ckAsset" = "<CKAsset: 0x7fd0b240bd40;...>";  
    "CD_entityName" = Post;  
}, recordType=CD_Post>
```

```
@objc(Post)
```

```
public class Post: NSManagedObject {  
    @NSManaged public var title: String?  
    @NSManaged public var content: String?  
    @NSManaged public var attachments: NSSet?  
    @NSManaged public var tags: NSSet?  
}
```

```
<CKRecord: 0x7fd0b2539220; recordID=Post_UUID, values={  
    "CD_content" = "An example core data string";  
    "CD_content_ckAsset" = "<CKAsset: 0x7fd0b2515c20;...>";  
    "CD_title" = "An example core data string";  
    "CD_title_ckAsset" = "<CKAsset: 0x7fd0b240bd40;...>";  
    "CD_entityName" = Post;  
}, recordType=CD_Post>
```



```
@objc(Post)
```

```
public class Post: NSManagedObject {  
    @NSManaged public var title: String?  
    @NSManaged public var content: String?  
    @NSManaged public var attachments: NSSet?  
    @NSManaged public var tags: NSSet?  
}
```

```
<CKRecord: 0x7fd0b2539220; recordID=Post_UUID, values={  
    "CD_content" = "An example core data string";  
    "CD_content_ckAsset" = "<CKAsset: 0x7fd0b2515c20;...>";  
    "CD_title" = "An example core data string";  
    "CD_title_ckAsset" = "<CKAsset: 0x7fd0b240bd40;...>";  
    "CD_entityName" = ImagePost;  
}, recordType=CD_Post>
```

```
@objc(Post)
```

```
public class Post: NSManagedObject {  
    @NSManaged public var title: String?  
    @NSManaged public var content: String?  
    @NSManaged public var attachments: NSSet?  
    @NSManaged public var tags: NSSet?  
}
```

```
<CKRecord: 0x7fd0b2539220; recordID=Post_UUID, values={  
    "CD_content" = "An example core data string";  
    "CD_content_ckAsset" = "<CKAsset: 0x7fd0b2515c20;...>";  
    "CD_title" = "An example core data string";  
    "CD_title_ckAsset" = "<CKAsset: 0x7fd0b240bd40;...>";  
    "CD_entityName" = ImagePost;  
}, recordType=CD_Post>
```

```
@objc(Post)
```

```
public class Post: NSManagedObject {  
    @NSManaged public var title: String?  
    @NSManaged public var content: String?  
    @NSManaged public var attachments: NSSet?  
    @NSManaged public var tags: NSSet?  
}
```

```
<CKRecord: 0x7fd0b2539220; recordID=Post_UUID, values={  
    "CD_content" = "An example core data string";  
    "CD_content_ckAsset" = "<CKAsset: 0x7fd0b2515c20;...>";  
    "CD_title" = "An example core data string";  
    "CD_title_ckAsset" = "<CKAsset: 0x7fd0b240bd40;...>";  
    "CD_entityName" = VideoPost;  
}, recordType=CD_Post>
```



```
@objc(Post)
```

```
public class Post: NSManagedObject {  
    @NSManaged public var title: String?  
    @NSManaged public var content: String?  
    @NSManaged public var attachments: NSSet?  
    @NSManaged public var tags: NSSet?  
}
```

```
<CKRecord: 0x7fd0b2539220; recordID=Post_UUID, values={  
    "CD_content" = "An example core data string";  
    "CD_content_ckAsset" = "<CKAsset: 0x7fd0b2515c20;...>";  
    "CD_title" = "An example core data string";  
    "CD_title_ckAsset" = "<CKAsset: 0x7fd0b240bd40;...>";  
    "CD_entityName" = VideoPost;  
}, recordType=CD_Post>
```

```
@objc(Post)
```

```
public class Post: NSManagedObject {  
    @NSManaged public var title: String?  
    @NSManaged public var content: String?  
    @NSManaged public var attachments: NSSet?  
    @NSManaged public var tags: NSSet?  
}
```

```
<CKRecord: 0x7fd0b2539220; recordID=Post_UUID, values={  
    "CD_content" = "An example core data string";  
    "CD_content_ckAsset" = "<CKAsset: 0x7fd0b2515c20;...>";  
    "CD_title" = "An example core data string";  
    "CD_title_ckAsset" = "<CKAsset: 0x7fd0b240bd40;...>";  
    "CD_entityName" = Post;  
}, recordType=CD_Post>
```

```
@objc(Post)
```

```
public class Post: NSManagedObject {  
    @NSManaged public var title: String?  
    @NSManaged public var content: String?  
    @NSManaged public var attachments: NSSet?  
    @NSManaged public var tags: NSSet?  
}
```

```
<CKRecord: 0x7fd0b2539220; recordID=Post_UUID, values={  
    "CD_content" = "An example core data string";  
    "CD_content_ckAsset" = "<CKAsset: 0x7fd0b2515c20;...>";  
    "CD_title" = "An example core data string";  
    "CD_title_ckAsset" = "<CKAsset: 0x7fd0b240bd40;...>";  
    "CD_entityName" = Post;  
}, recordType=CD_Post>
```



```
@objc(Post)
```

```
public class Post: NSManagedObject {  
    @NSManaged public var title: String?  
    @NSManaged public var content: String?  
    @NSManaged public var attachments: NSSet?  
    @NSManaged public var tags: NSSet?  
}
```

```
<CKRecord: 0x7fd0b2539220; recordID=Post_UUID, values={  
    "CD_content" = "An example core data string";  
    "CD_content_ckAsset" = "<CKAsset: 0x7fd0b2515c20;...>";  
    "CD_title" = "An example core data string";  
    "CD_title_ckAsset" = "<CKAsset: 0x7fd0b240bd40;...>";  
    "CD_entityName" = Post;  
}, recordType=CD_Post>
```

```
@objc(Post)
```

```
public class Post: NSManagedObject {  
    @NSManaged public var title: String?  
    @NSManaged public var content: String?  
    @NSManaged public var attachments: NSSet?  
    @NSManaged public var tags: NSSet?  
}
```

```
<CKRecord: 0x7fd0b2539220; recordID=Post_UUID, values={  
    "CD_content" = "Amazing... but brief!";  
    "CD_title" = "An amazing post!";  
    "CD_entityName" = Post;  
}, recordType=CD_Post>
```

```
@objc(Post)
```

```
public class Post: NSManagedObject {  
    @NSManaged public var title: String?  
    @NSManaged public var content: String?  
    @NSManaged public var attachments: NSSet?  
    @NSManaged public var tags: NSSet?  
}
```

```
<CKRecord: 0x7fd0b2539220; recordID=Post_UUID, values={  
    "CD_content_ckAsset" = "<CKAsset: 0x7fd0b2515c20;...>";  
    "CD_title" = "An amazing post!";  
    "CD_entityName" = Post;  
}, recordType=CD_Post>
```



```
@objc(Post)
```

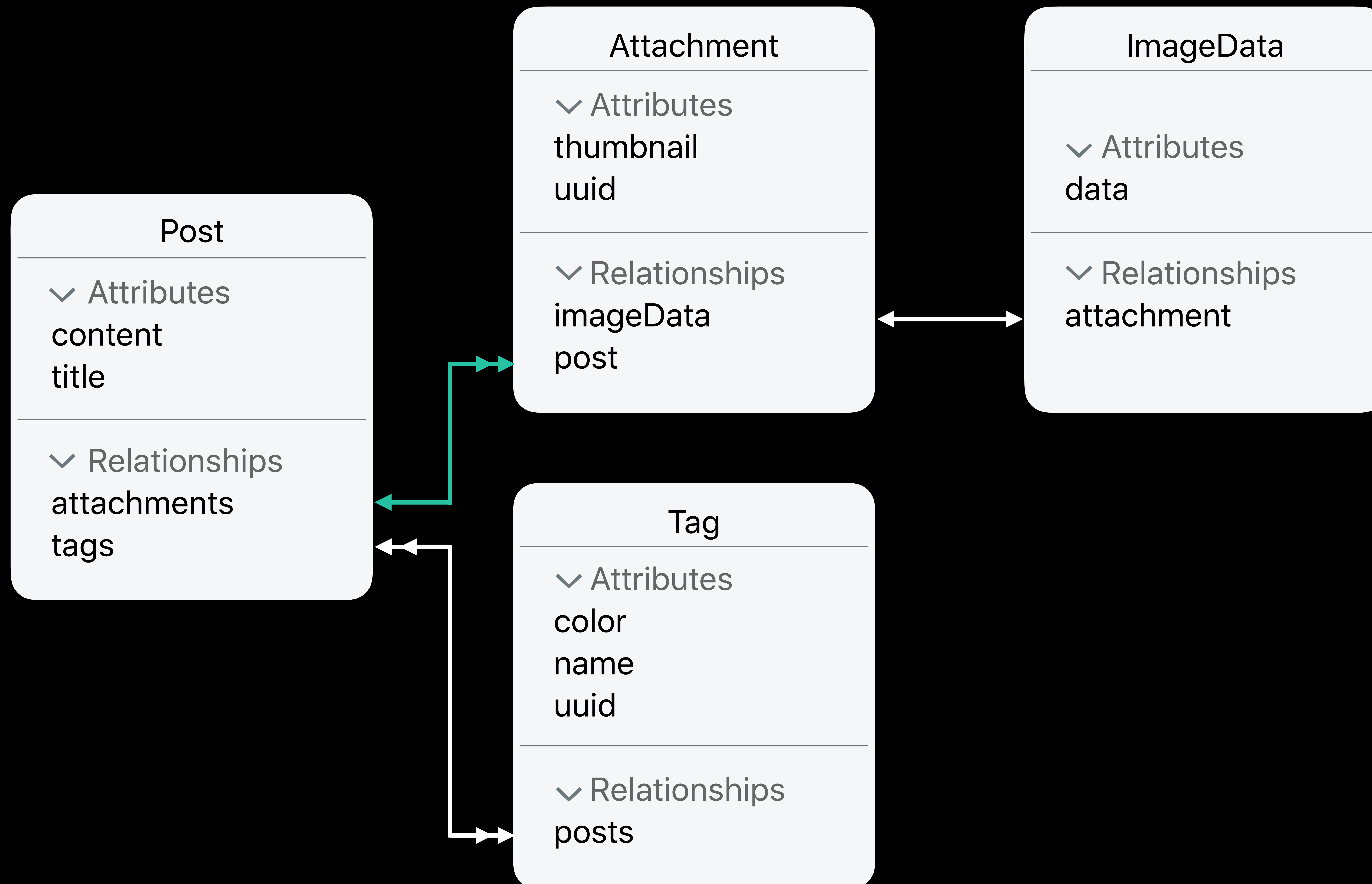
```
public class Post: NSManagedObject {  
    @NSManaged public var title: String?  
    @NSManaged public var content: String?  
    @NSManaged public var attachments: NSSet?  
    @NSManaged public var tags: NSSet?  
}
```

```
<CKRecord: 0x7fd0b2539220; recordID=Post_UUID, values={  
    "CD_content_ckAsset" = "<CKAsset: 0x7fd0b2515c20;...>";  
    "CD_title" = "An amazing post!";  
    "CD_entityName" = Post;  
}, recordType=CD_Post>
```

```
@objc(Post)
public class Post: NSManagedObject {
    @NSManaged public var title: String?
    @NSManaged public var content: String?
    @NSManaged public var attachments: NSSet?
    @NSManaged public var tags: NSSet?
}
```

```
@objc(Post)
public class Post: NSManagedObject {
    @NSManaged public var title: String?
    @NSManaged public var content: String?
    @NSManaged public var attachments: NSSet?
    @NSManaged public var tags: NSSet?
}
```


Sample Application Data Model



```
@objc(Post)
public class Post: NSManagedObject {
    @NSManaged public var attachments: NSSet?
}
```

```
@objc(Attachment)
public class Attachment: NSManagedObject {
    @NSManaged public var post: Post?
}
```

```
@objc(Post)
public class Post: NSManagedObject {
    @NSManaged public var attachments: NSSet?
}
```

```
@objc(Attachment)
public class Attachment: NSManagedObject {
    @NSManaged public var post: Post?
}
```



```
@objc(Post)
public class Post: NSManagedObject {
    @NSManaged public var attachments: NSSet?
}
```

```
@objc(Attachment)
public class Attachment: NSManagedObject {
    @NSManaged public var post: Post?
}
```

```
@objc(Post)
```

```
public class Post: NSManagedObject {  
    @NSManaged public var attachments: NSSet?  
}
```

```
@objc(Attachment)
```

```
public class Attachment: NSManagedObject {  
    @NSManaged public var post: Post?  
}
```

```
<CKRecord: 0x7fd0b4011bf0; recordID=Attachment_UUID, values={  
    "CD_entityName" = Attachment;  
    "CD_imageData" = "ImageData_UUID";  
    "CD_post" = "Post_UUID";  
    "CD_uuid" = "B7841A69-868D-49A9-851D-3BB7030DA73A";  
}, recordType=CD_Attachment>
```

```
@objc(Post)
public class Post: NSManagedObject {
    @NSManaged public var attachments: NSSet?
}
```

```
@objc(Attachment)
public class Attachment: NSManagedObject {
    @NSManaged public var post: Post?
}
```

```
<CKRecord: 0x7fd0b4011bf0; recordID=Attachment_UUID, values={
    "CD_entityName" = Attachment;
    "CD_imageData" = "ImageData_UUID";
    "CD_post" = "Post_UUID";
    "CD_uuid" = "B7841A69-868D-49A9-851D-3BB7030DA73A";
}, recordType=CD_Attachment>
```



```
@objc(Post)
```

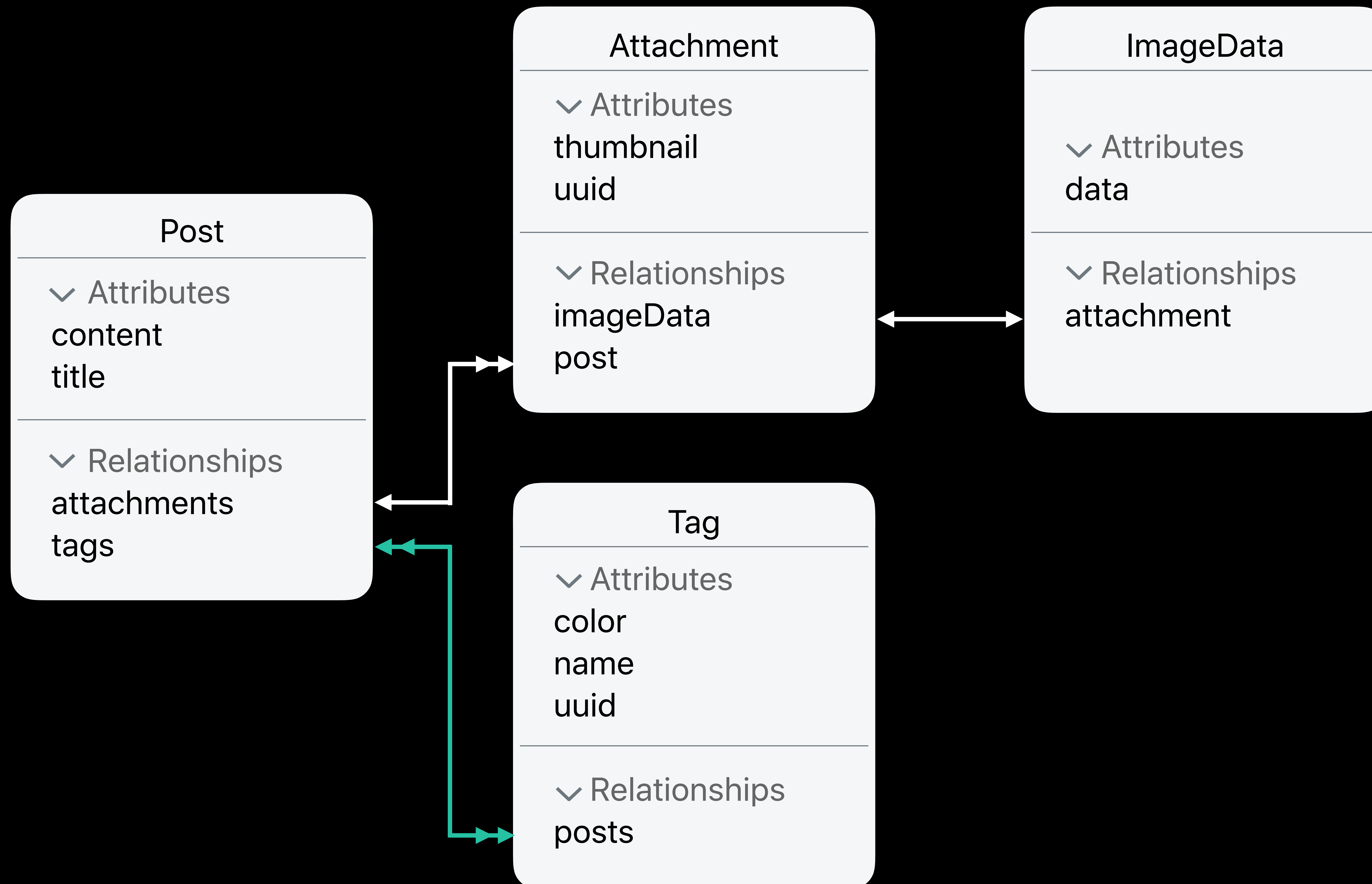
```
public class Post: NSManagedObject {  
    @NSManaged public var attachments: NSSet?  
}
```

```
@objc(Attachment)
```

```
public class Attachment: NSManagedObject {  
    @NSManaged public var post: Post?  
}
```

```
<CKRecord: 0x7fd0b4011bf0; recordID=Attachment_UUID, values={  
    "CD_entityName" = Attachment;  
    "CD_imageData" = "ImageData_UUID";  
    "CD_post" = "Post_UUID";  
    "CD_uuid" = "B7841A69-868D-49A9-851D-3BB7030DA73A";  
}, recordType=CD_Attachment>
```

Sample Application Data Model



```
@objc(Post)
public class Post: NSManagedObject {
    @NSManaged public var tags: NSSet?
}
```

```
@objc(Tag)
public class Tag: NSManagedObject {
    @NSManaged public var posts: NSSet?
}
```



```
@objc(Post)
```

```
public class Post: NSManagedObject {
```

```
    @NSManaged public var tags: NSSet?
```

```
}
```

```
@objc(Tag)
```

```
public class Tag: NSManagedObject {
```

```
    @NSManaged public var posts: NSSet?
```

```
}
```

```
<CKRecord: 0x7fd0b2539220; recordID=Post_UUID, values={
  "CD_content" = "An example core data string";
  "CD_content_ckAsset" = "<CKAsset: 0x7fd0b2515c20;...>";
  "CD_title" = "An example core data string";
  "CD_title_ckAsset" = "<CKAsset: 0x7fd0b240bd40;...>";
  "CD_entityName" = Post;
}, recordType=CD_Post>
```

```
<CKRecord: 0x7fd0b2419da0; recordID=Tag_UUID, values={
  "CD_color" = {length = 17, bytes = 0x536f6d65206578616d706c652064617461};
  "CD_color_ckAsset" = "<CKAsset: 0x7fd0b2532de0;...>";
  "CD_name" = "An example core data string";
  "CD_name_ckAsset" = "<CKAsset: 0x7fd0b24215d0;...>";
  "CD_uuid" = "DEB9D10A-55C1-42EC-BA15-77A41028AF9B";
  "CD_entityName" = Tag;
}, recordType=CD_Tag>
```

```
<CKRecord: 0x7fd0b24202f0; recordID=CDMR_UUID, values={
    "CD_entityNames" = "Post:Tag";
    "CD_recordNames" = "Post_UUID:Tag_UUID";
    "CD_relationships" = "tags:posts";
}, recordType=CDMR>
```



```
<CKRecord: 0x7fd0b24202f0; recordID=CDMR_UUID, values={
  "CD_entityNames" = "Post:Tag";
  "CD_recordNames" = "Post_UUID:Tag_UUID";
  "CD_relationships" = "tags:posts";
}, recordType=CDMR>
```

```
<CKRecord: 0x7fd0b24202f0; recordID=CDMR_UUID, values={  
  "CD_entityNames" = "Post:Tag";  
  "CD_recordNames" = "Post_UUID:Tag_UUID";  
  "CD_relationships" = "tags:posts";  
}, recordType=CDMR>
```

```
<CKRecord: 0x7fd0b24202f0; recordID=CDMR_UUID, values={
  "CD_entityNames" = "Post:Tag";
  "CD_recordNames" = "Post_UUID:Tag_UUID";
  "CD_relationships" = "tags:posts";
}, recordType=CDMR>
```



```
<CKRecord: 0x7fd0b24202f0; recordID=CDMR_UUID, values={
  "CD_entityNames" = "Post:Tag";
  "CD_recordNames" = "Post_UUID:Tag_UUID";
  "CD_relationships" = "tags:posts";
}, recordType=CDMR>
```

Data Modeling for Collaboration

Data Modeling for Collaboration

Collaboration is not conflict resolution

Data Modeling for Collaboration

Collaboration is not conflict resolution

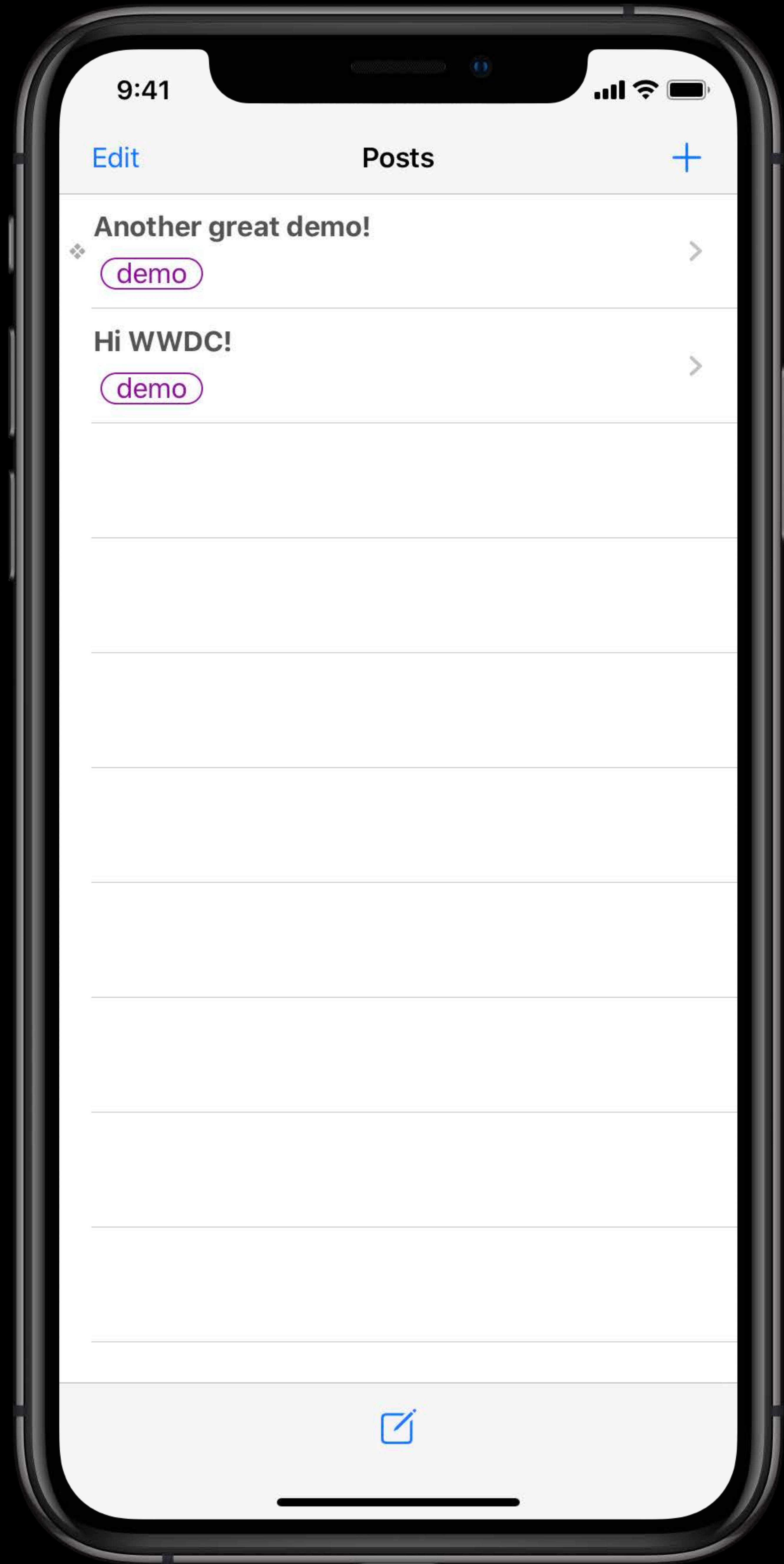
NSPersistentCloudKitContainer resolves conflicts automatically

Data Modeling for Collaboration

Collaboration is not conflict resolution

NSPersistentCloudKitContainer resolves conflicts automatically

Get better merge behavior with relationships



9:41



Edit

Posts



Another great demo!



demo



Hi WWDC!

demo



9:41



Edit

Posts



Another great demo!

demo



Hi WWDC!

demo



Let's talk about collaboration



9:41



Edit

Posts



Another great demo!

demo



Hi WWDC!

demo



Let's talk about collaboration



9:41



Edit

Posts



Another great demo!

demo



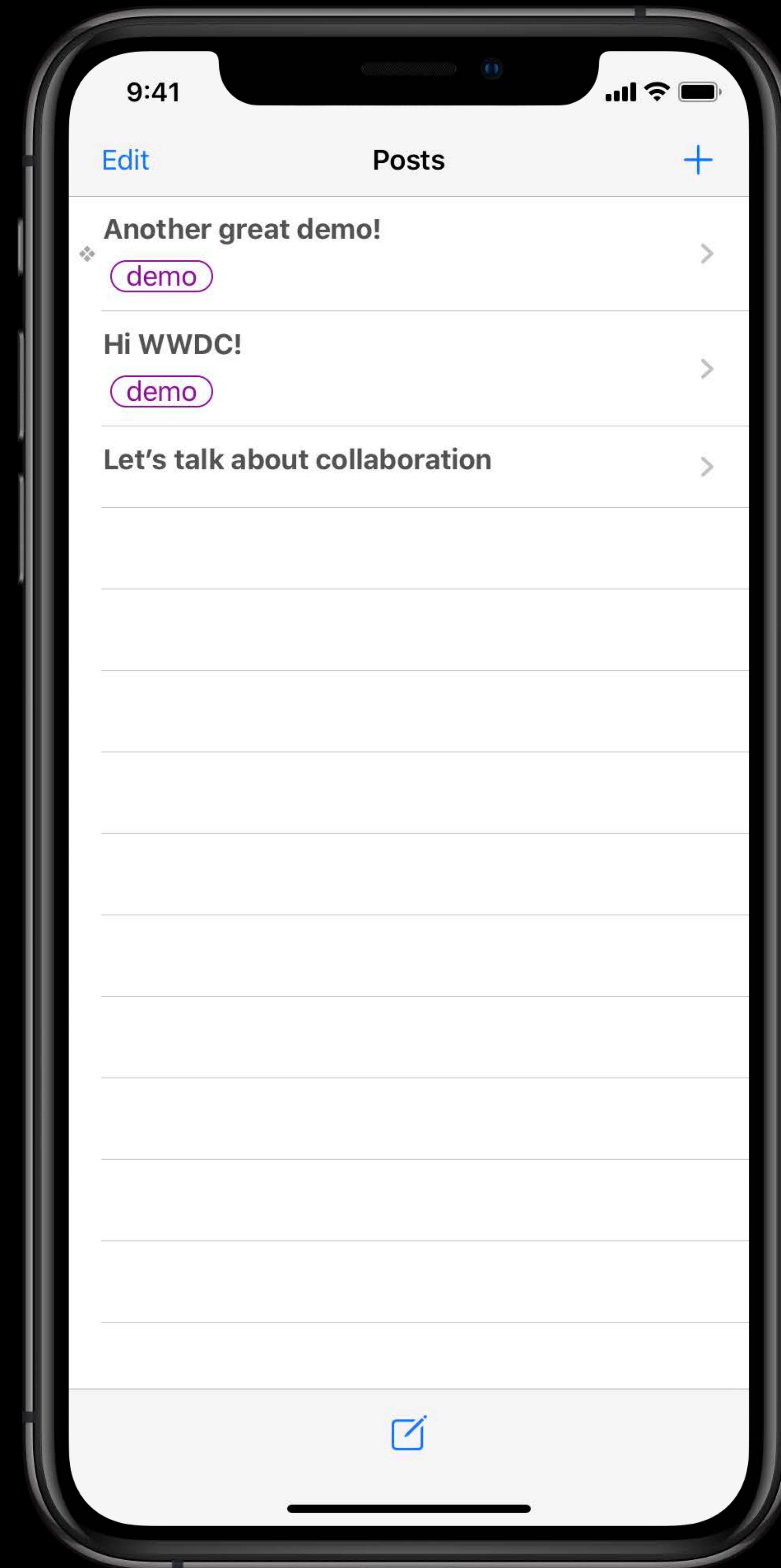
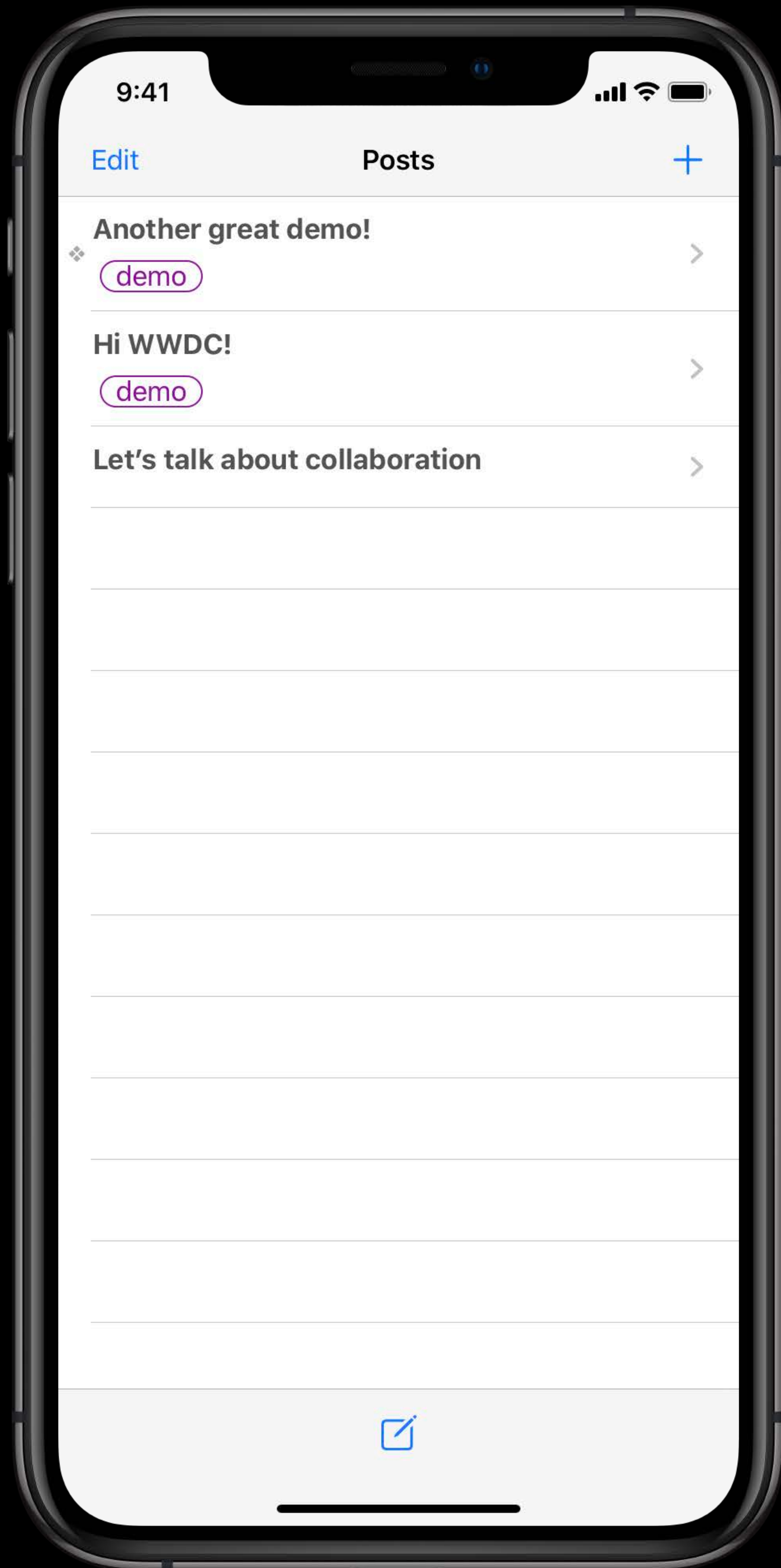
Hi WWDC!

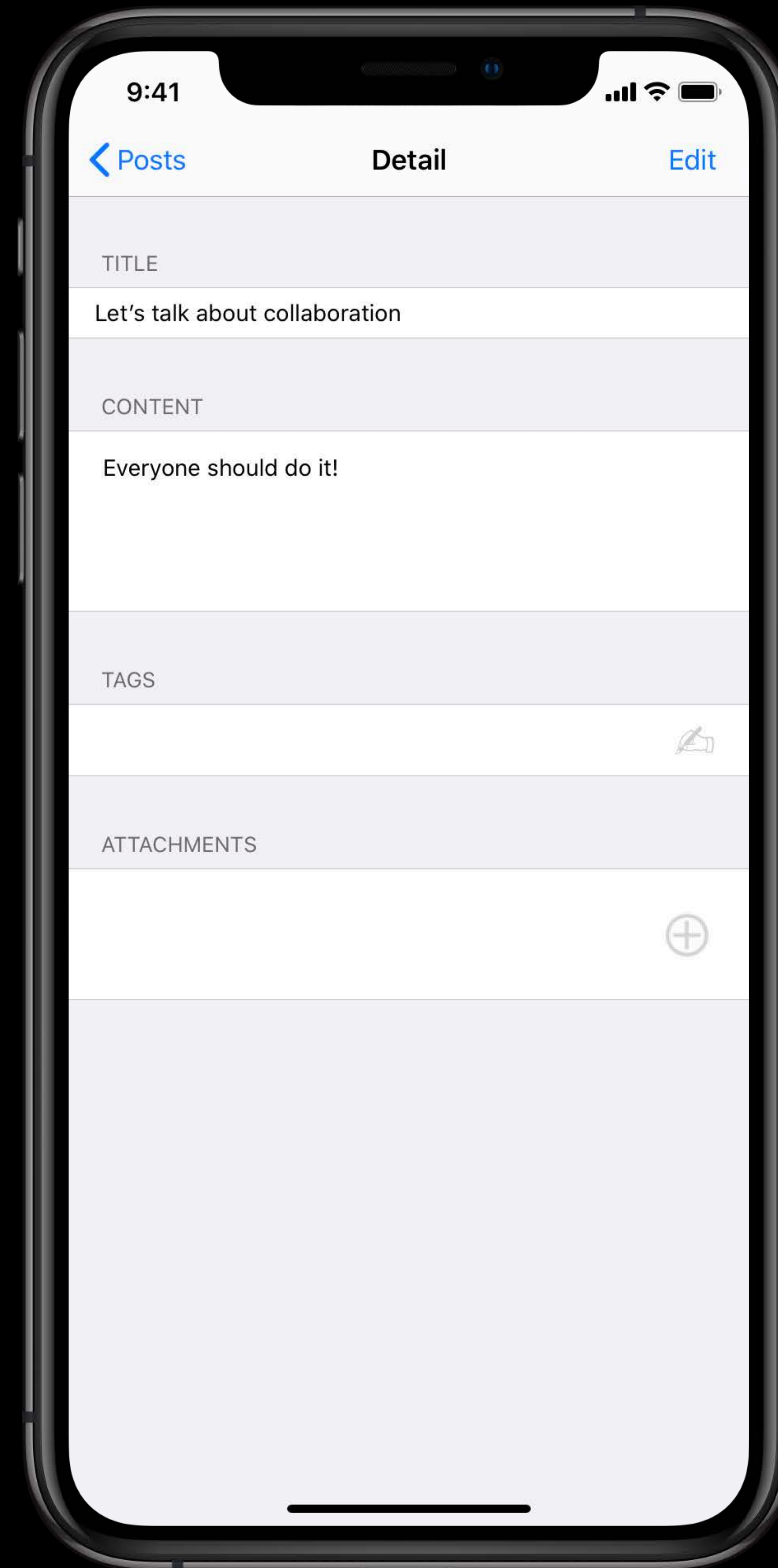
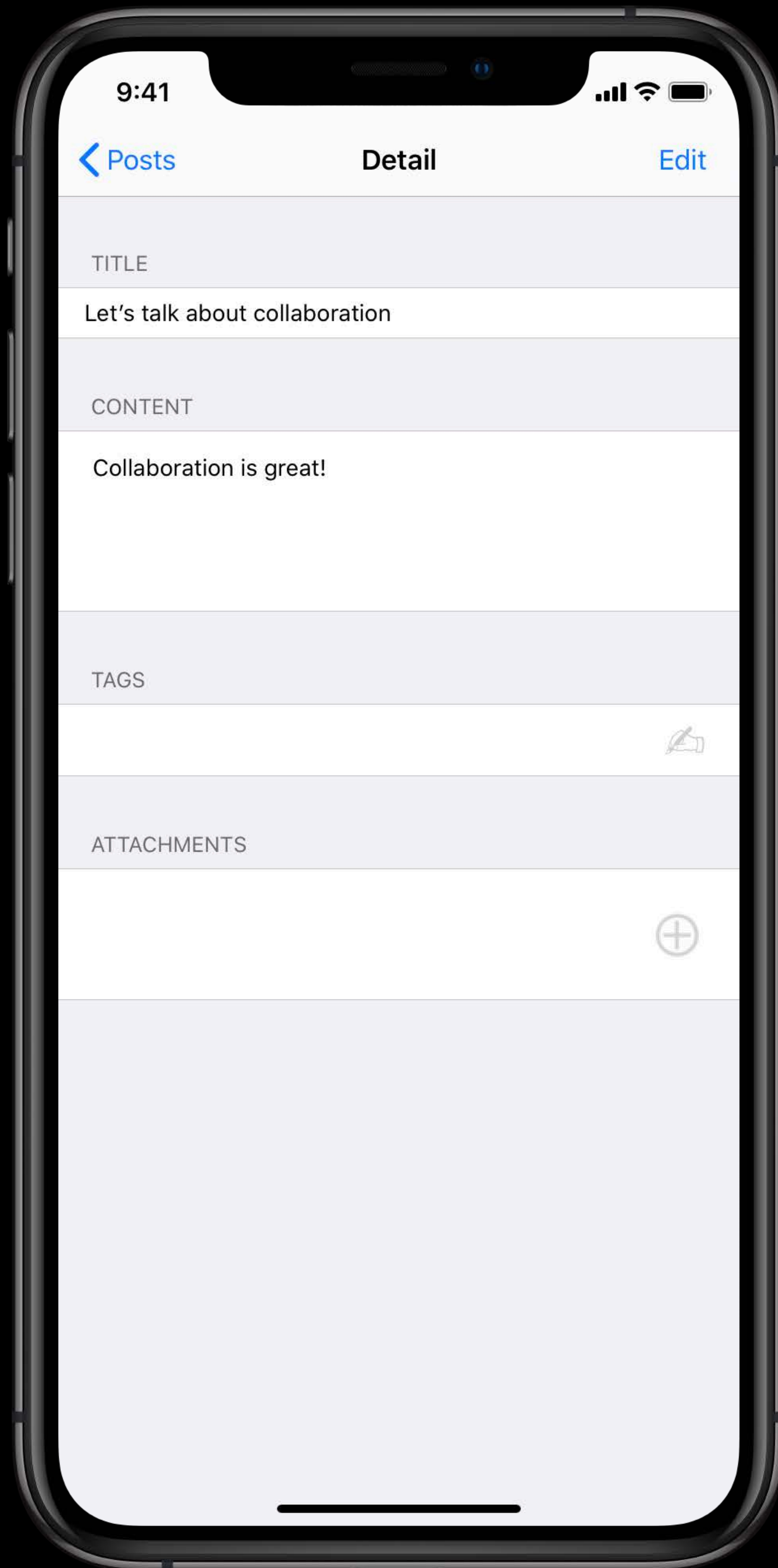
demo



Let's talk about collaboration







9:41



< Posts

Detail

Edit

TITLE

Let's talk about collaboration

CONTENT

Collaboration is great!

TAGS



ATTACHMENTS



9:41



< Posts

Detail

Edit

TITLE

Let's talk about collaboration

CONTENT

Everyone should do it!

TAGS



ATTACHMENTS



9:41



[← Posts](#)

Detail

[Edit](#)

TITLE

Let's talk about collaboration

CONTENT

Collaboration is great!

TAGS



ATTACHMENTS



9:41



[← Posts](#)

Detail

[Edit](#)

TITLE

Let's talk about collaboration

CONTENT

Collaboration is great!

TAGS



ATTACHMENTS



9:41



[← Posts](#)

Detail

[Edit](#)

TITLE

Let's talk about collaboration

CONTENT

Everyone should do it!

TAGS



ATTACHMENTS



9:41



[← Posts](#)

Detail

[Edit](#)

TITLE

Let's talk about collaboration

CONTENT

Everyone should do it!

TAGS



ATTACHMENTS



9:41



[← Posts](#)

Detail

[Edit](#)

TITLE

Let's talk about collaboration

CONTENT

Collaboration is great! Everyone should do it!

TAGS



ATTACHMENTS



9:41



[← Posts](#)

Detail

[Edit](#)

TITLE

Let's talk about collaboration

CONTENT

Collaboration is great! Everyone should do it!

TAGS



ATTACHMENTS



9:41



[← Posts](#)

Detail

[Edit](#)

TITLE

Let's talk about collaboration

CONTENT

Everyone Collaboration is should great! do it!

TAGS



ATTACHMENTS



9:41



[← Posts](#)

Detail

[Edit](#)

TITLE

Let's talk about collaboration

CONTENT

Everyone Collaboration is should great! do it!

TAGS

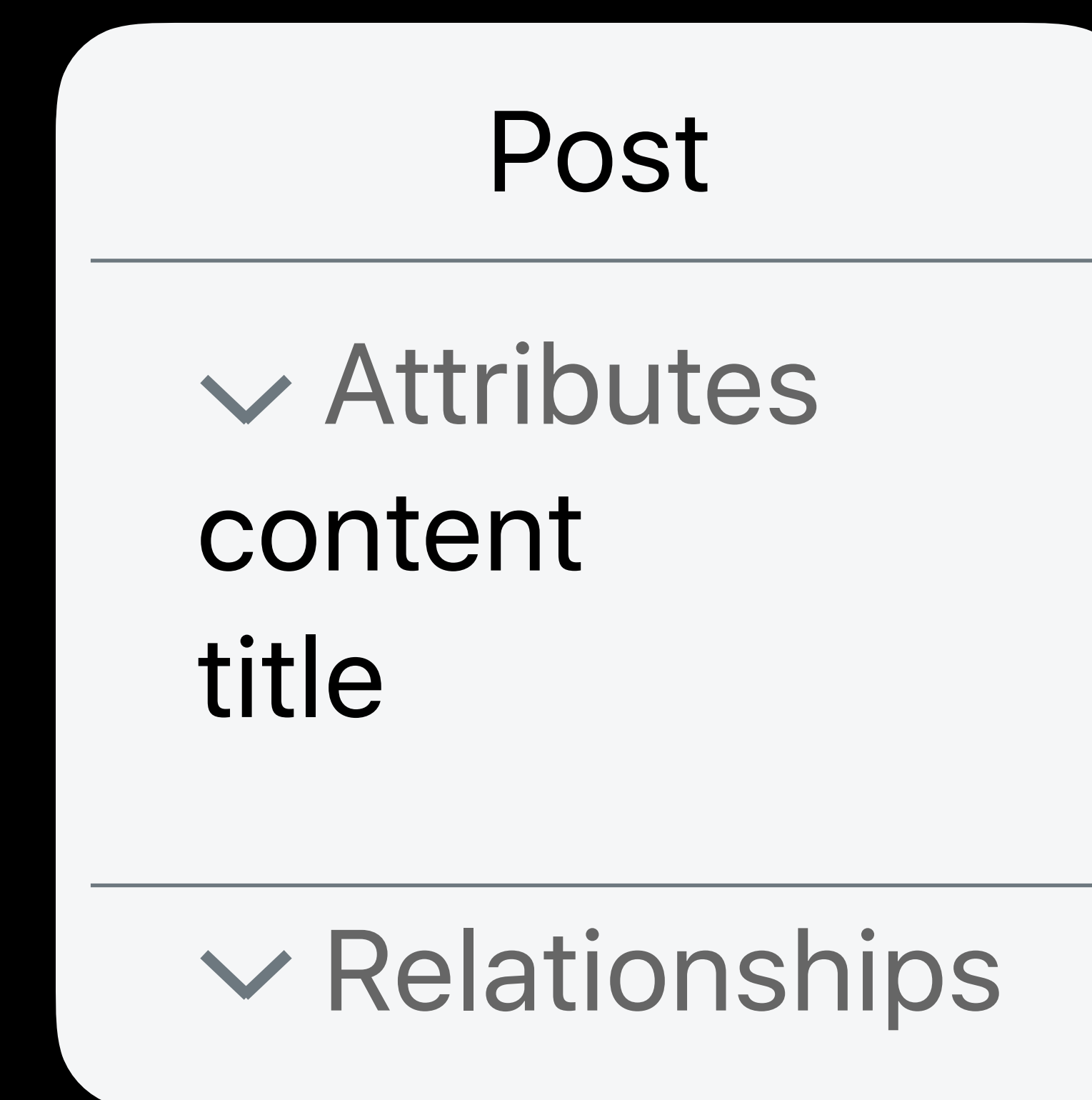


ATTACHMENTS



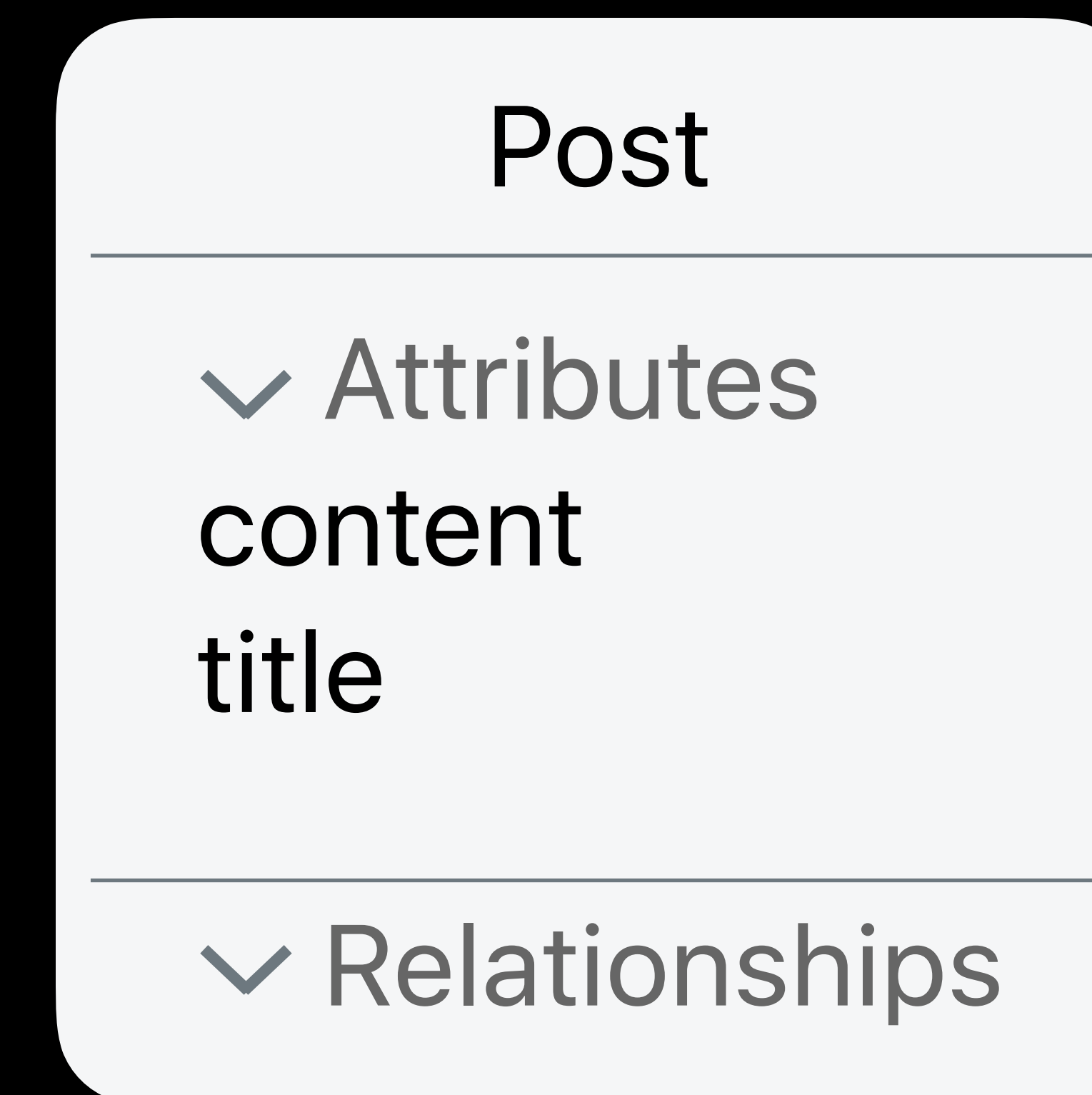
Data Modeling for Collaboration

Data Modeling for Collaboration



Data Modeling for Collaboration

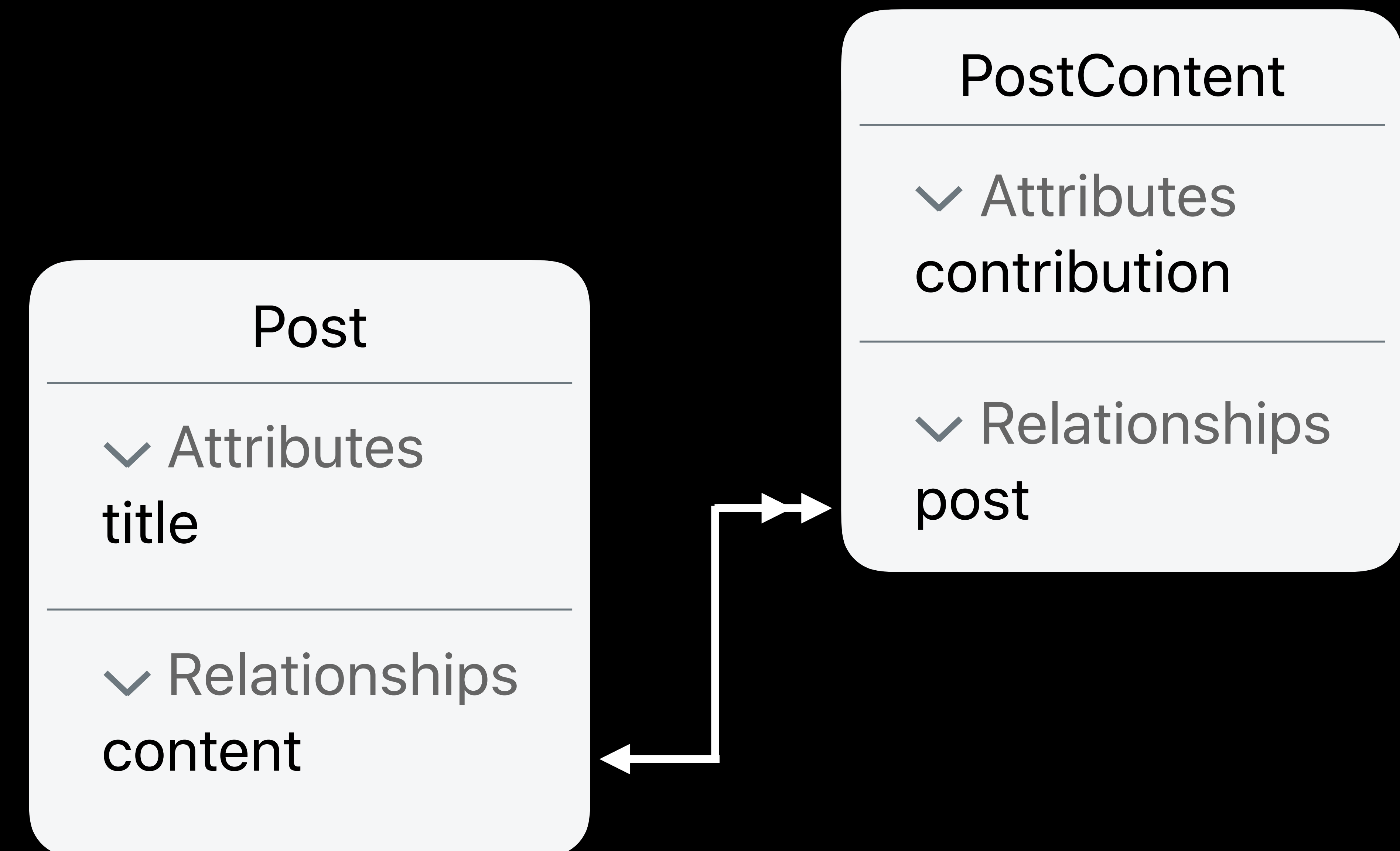
Avoid collisions on "flat" values



Data Modeling for Collaboration

Avoid collisions on "flat" values

Instead model values as contributions

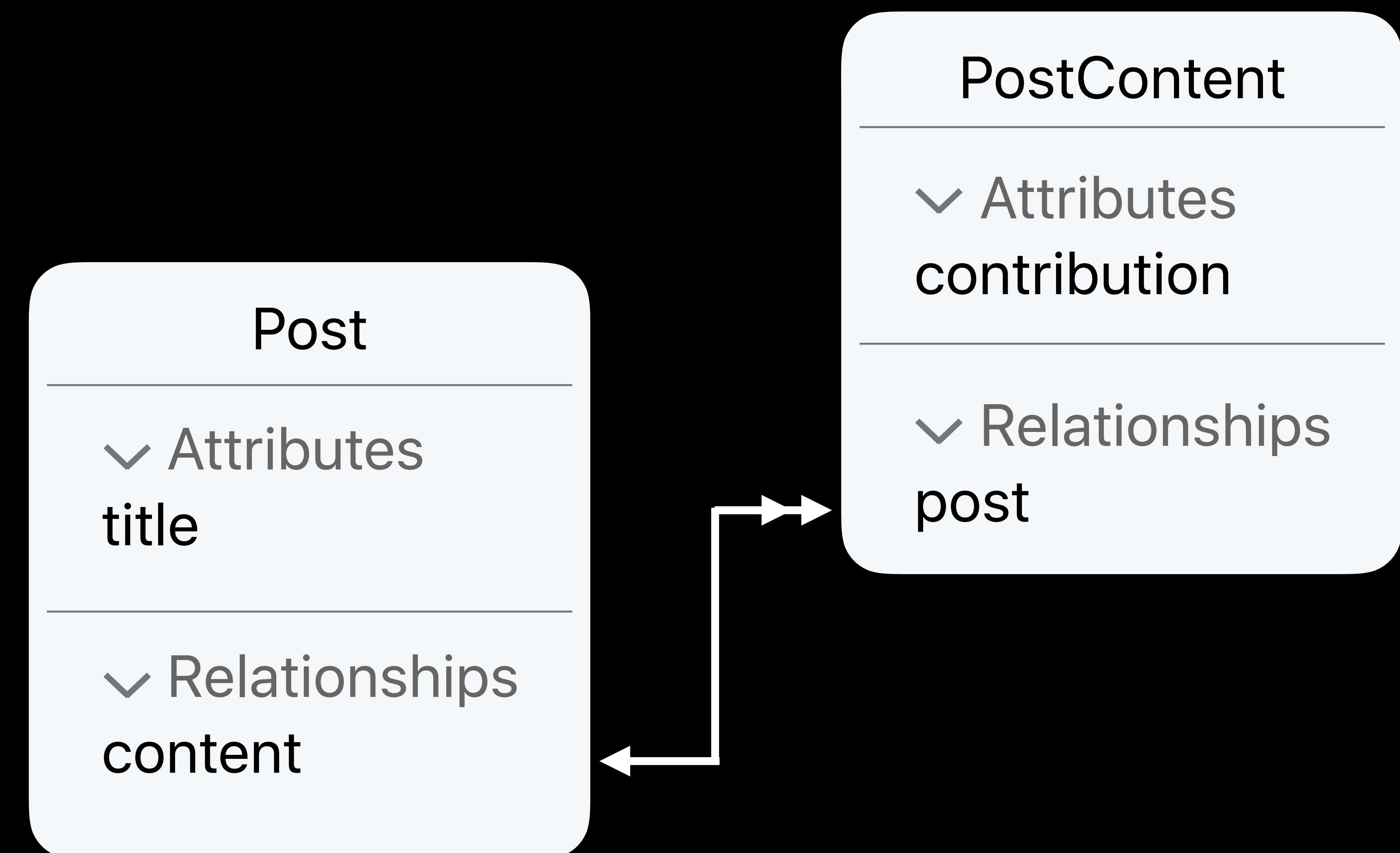


Data Modeling for Collaboration

Avoid collisions on "flat" values

Instead model values as contributions

Leverage relationships for eventual consistency



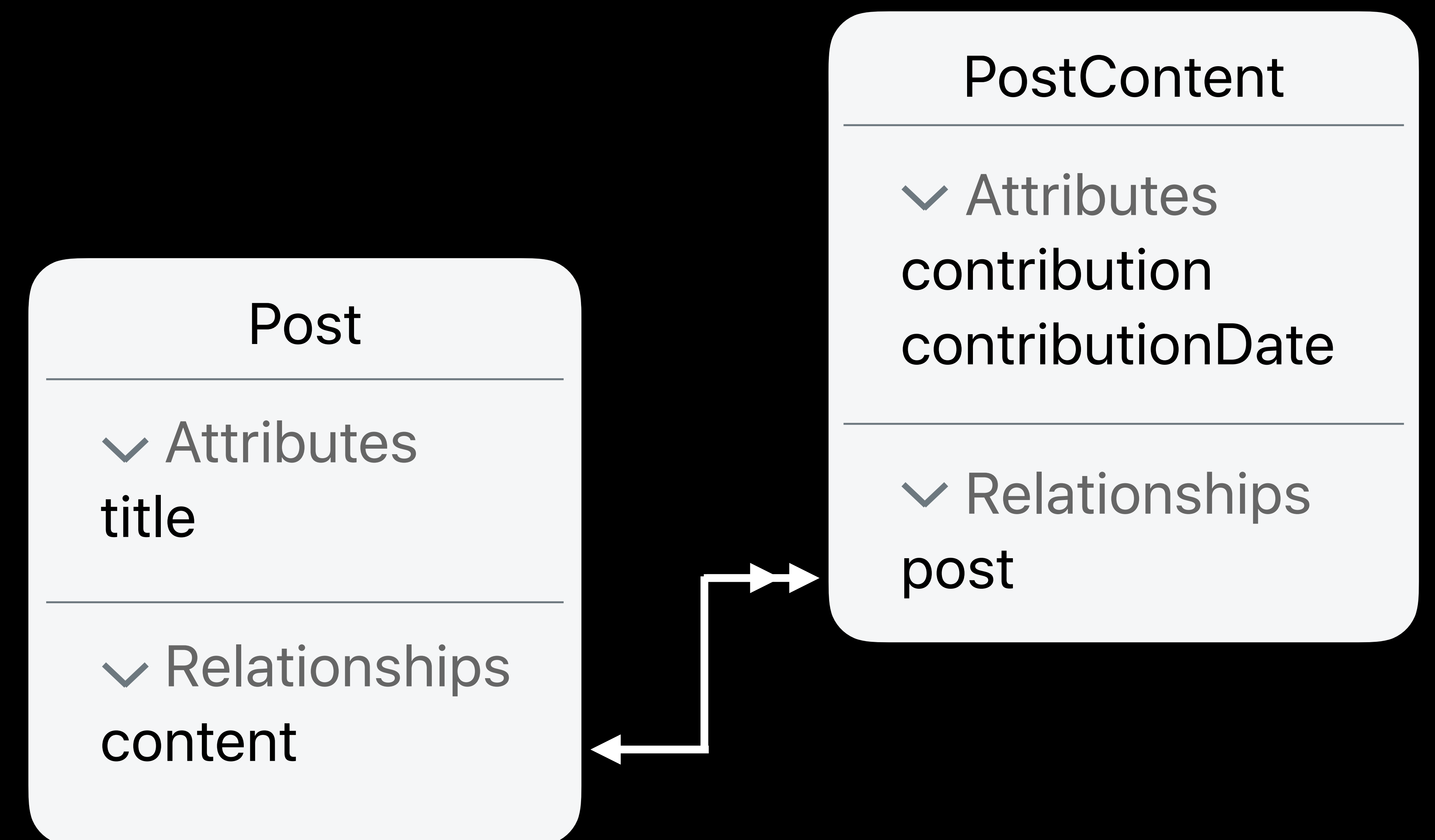
Data Modeling for Collaboration

Avoid collisions on "flat" values

Instead model values as contributions

Leverage relationships for eventual consistency

Order contributions



Data Modeling for Collaboration

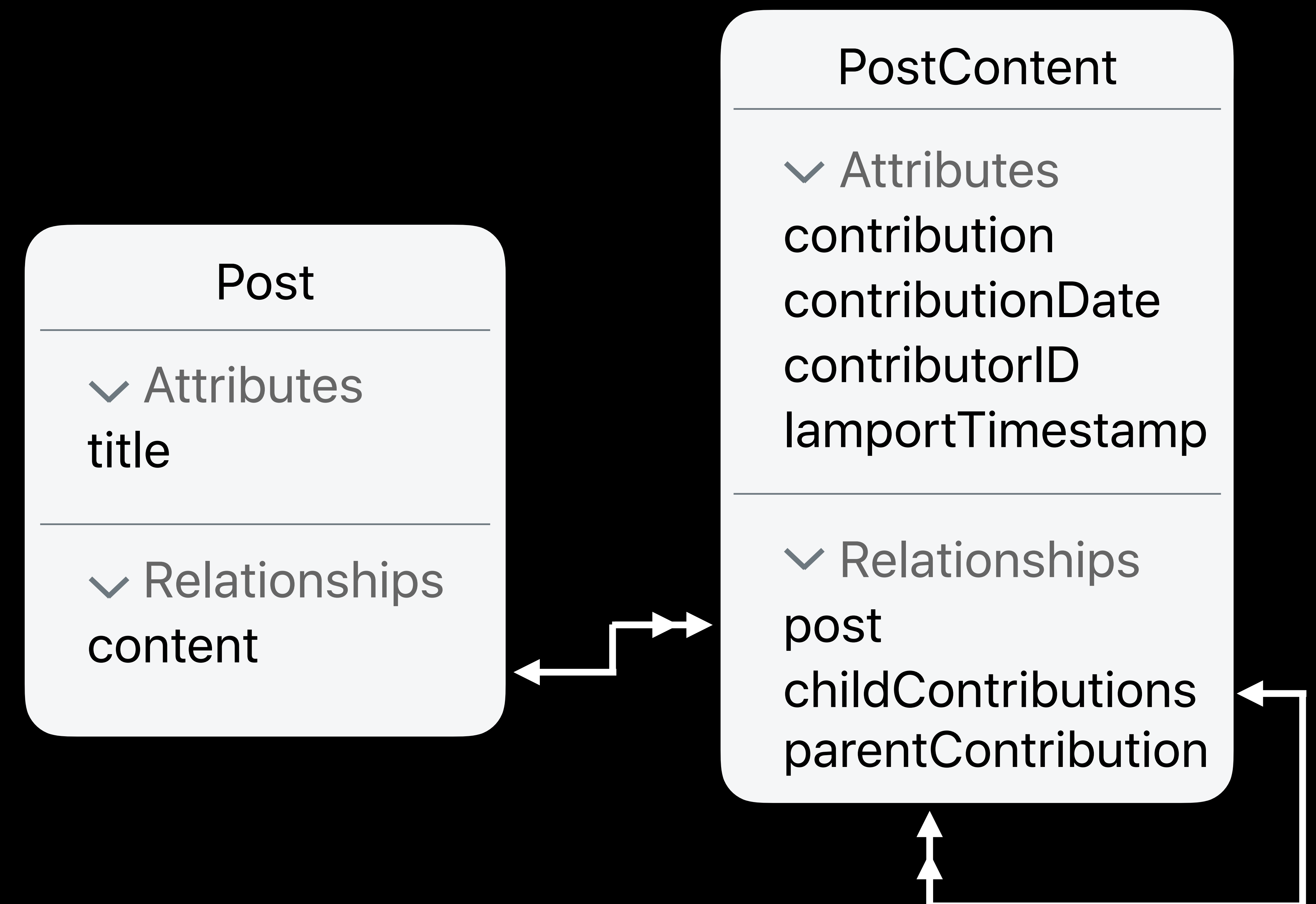
Avoid collisions on "flat" values

Instead model values as contributions

Leverage relationships for eventual consistency

Order contributions

Iterate as necessary



Summary

Summary

Easy sync with `NSPersistentCloudKitContainer`

Summary

Easy sync with `NSPersistentCloudKitContainer`

New sample code and documentation!

Summary

Easy sync with `NSPersistentCloudKitContainer`

New sample code and documentation!

Build on `NSPersistentCloudKitContainer`

More Information

developer.apple.com/wwdc19/202

Core Data and CloudKit Lab

Tuesday, 3:00

CloudKit Lab

Wednesday, 9:00

Core Data Lab

Wednesday, 2:00

