

Application

Faisons plus ample connaissance...

Application

- ✦ Structure
- ✦ Cycle de vie
- ✦ Spécificités

Application

Structure

Architecture d'un projet Xcode

- ✦ Fichiers .swift
 - ✦ Code
 - ✦ AppDelegate permet de réagir aux changements d'états de l'application
- ✦ Fichiers .storyboard
 - ✦ Interface graphique dans Main
 - ✦ Écran de lancement dans LaunchScreen

Architecture d'un projet Xcode

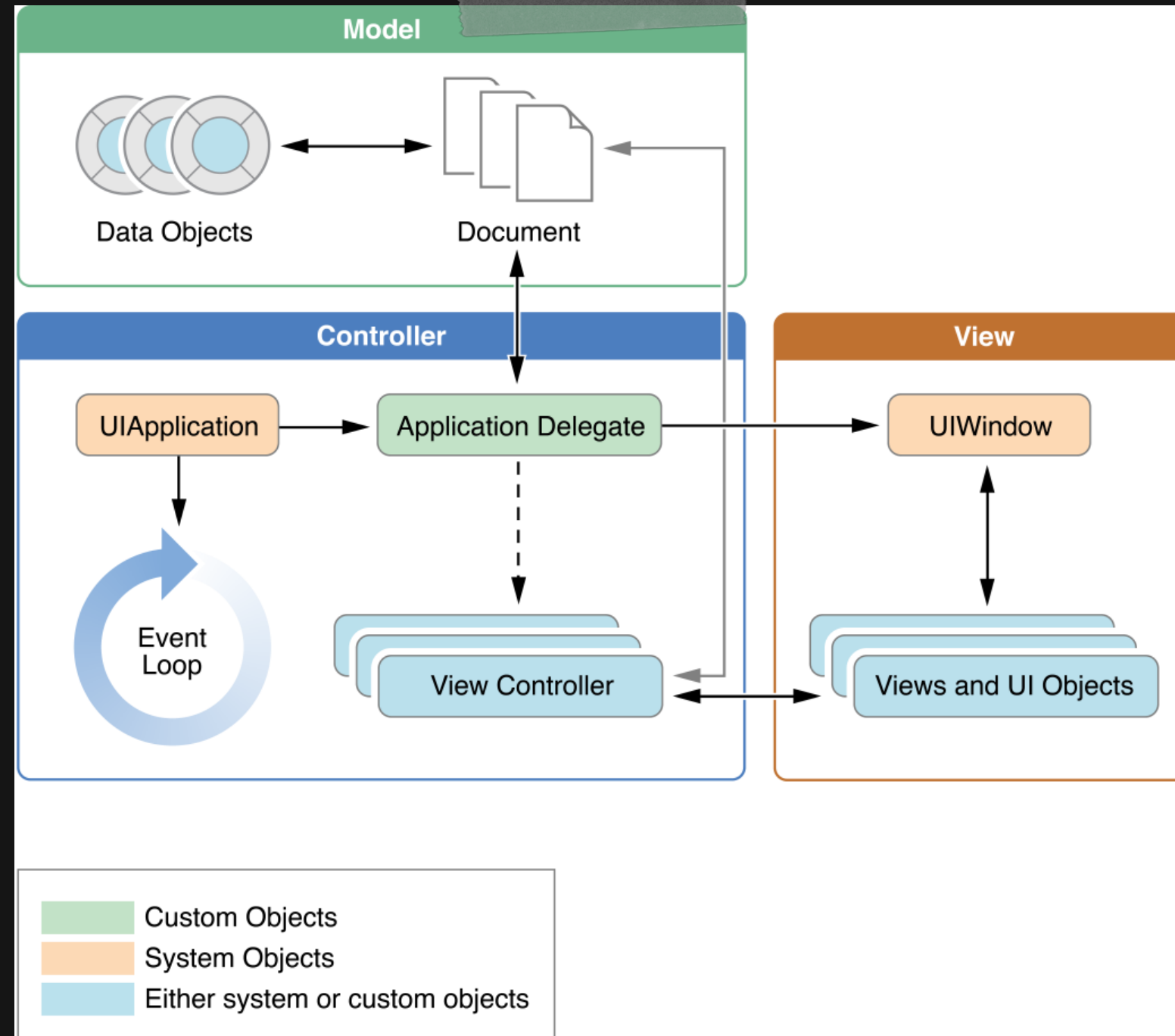
- ✦ Fichier .xcodeproj
 - ✦ Contient la structure du projet et ses réglages
- ✦ Fichiers .xcassets
 - ✦ Bibliothèque de ressources graphiques
- ✦ Fichier Info.plist
 - ✦ Stocke des infos essentielles au fonctionnement du projet
 - ✦ On y déclare les données privées auxquelles on veut accéder

Cycle de vie

Fonction main()

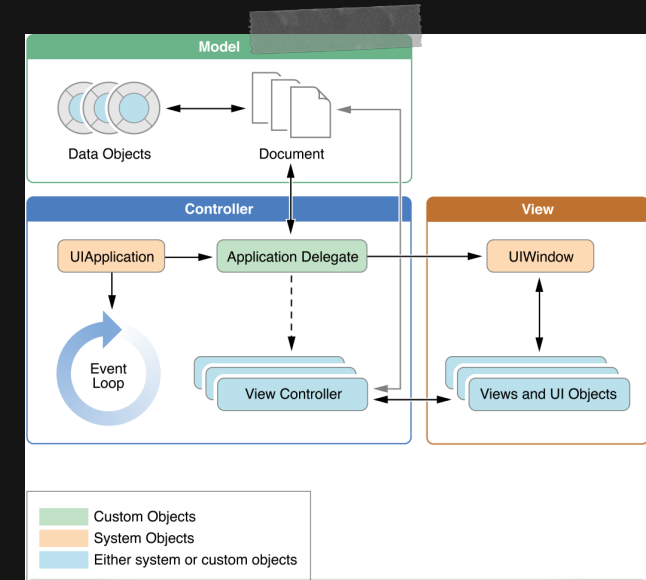
- ✦ Le point d'entrée d'une app basée sur du C est la fonction main()
 - ✦ On n'a pas à l'écrire nous même, elle est générée par Xcode
 - ✦ Son job est de lancer l'app et de transmettre le contrôle à UIKit
- ✦ En Swift, cette fonction n'existe pas à proprement parler
 - ✦ UIKit est lancé grâce à un attribut @UIApplicationMain placé dans l'App Delegate

Structure de l'app

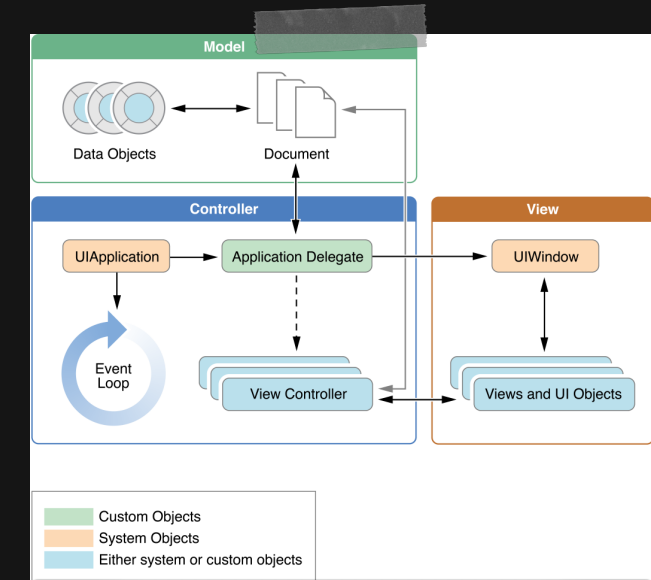


UIApplication

- ✧ Gère la boucle d'événements et les comportements "haut-niveau"
- ✧ Rapporte les événements importants de l'app à son delegate
 - ✧ Changements d'état
 - ✧ Arrivée de notifications
 - ✧ etc...
- ✧ Ne se sous-classe pas !

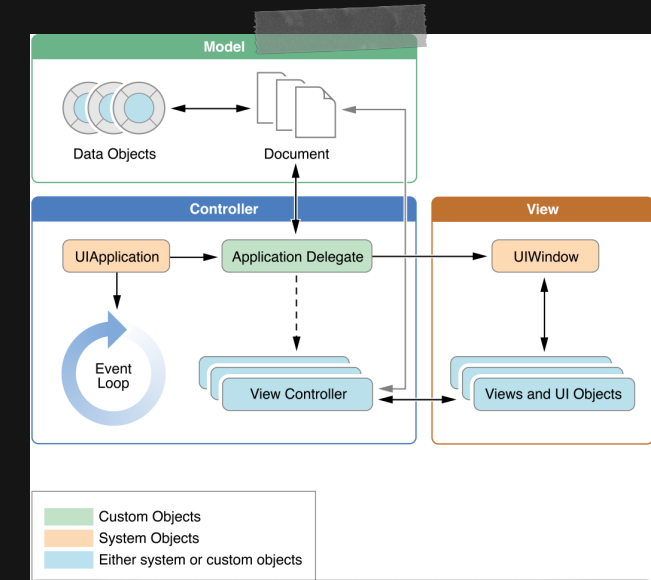


App Delegate



- ✦ Le coeur de la personnalisation de notre app
- ✦ Fonctionne en tandem avec l'objet UIApplication pour réagir aux changements de notre app
- ✦ On est certain de retrouver cet objet dans toutes les apps
 - ✦ Souvent utilisé pour mettre en place la base d'un app

UIWindow

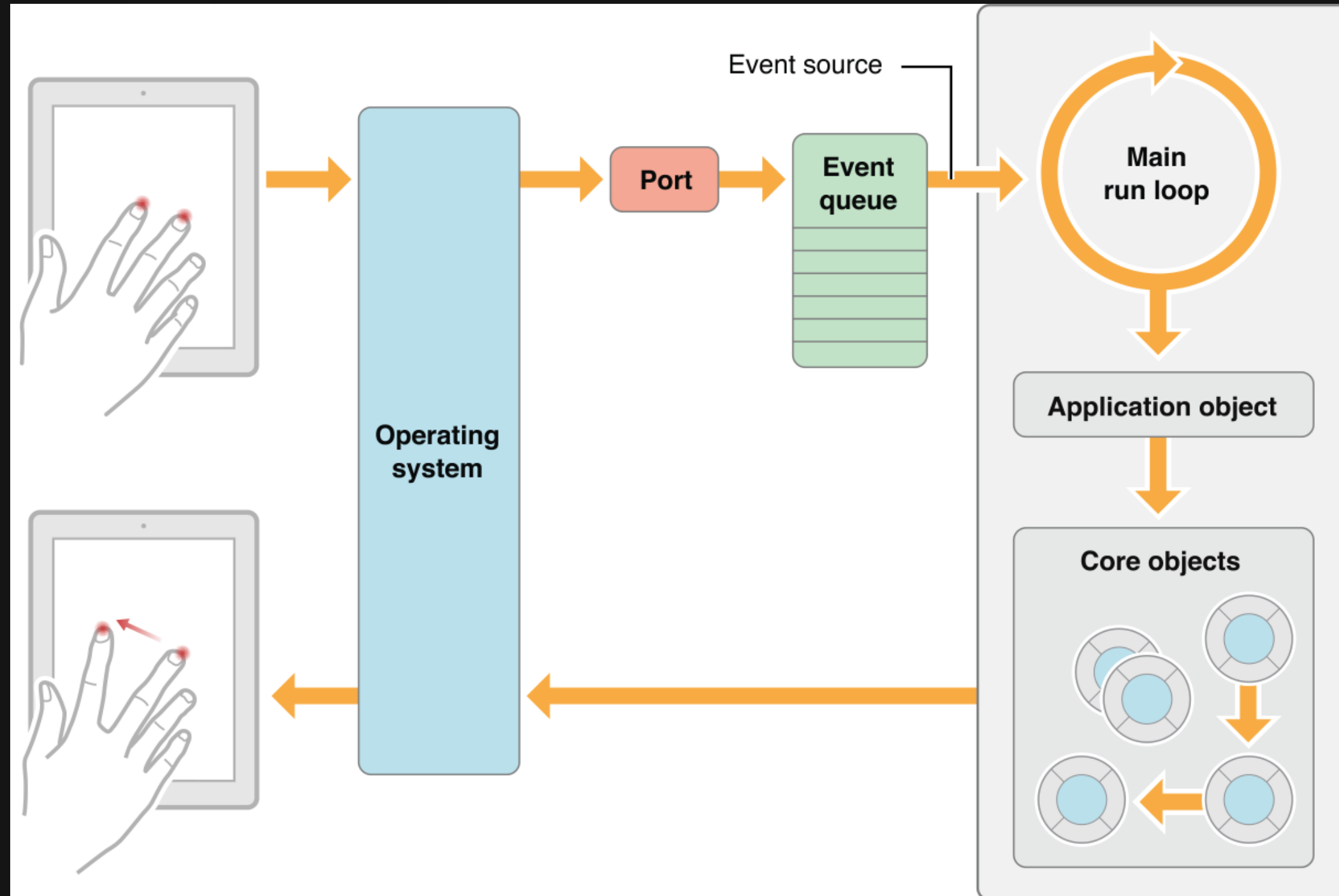


- ✦ Coordonnes la présentation des vues à l'écran
- ✦ Unique en général, sauf cas particuliers (AirPlay, écran externe, etc.)
- ✦ Les View Controllers de notre app seront affichés dans la fenêtre
- ✦ Travaille avec UIApplication pour délivrer des événements à nos vues et contrôleurs de vues.

Run Loop principale

- ✦ Mise en place par l'objet UIApplication au lancement
- ✦ Gère les événements
- ✦ Fonctionne sur le thread principal
- ✦ S'assure que les événements utilisateurs sont traités en série, dans l'ordre d'arrivée

Run Loop principale



États de l'app

- ✦ 5 états pour une application
 - ✦ Not Running
 - ✦ Inactive
 - ✦ Active
 - ✦ Background
 - ✦ Suspended

État de l'app : Not Running

- ✦ App jamais lancée ou stoppée par le système
 - ✦ N'utilise pas de ressources CPU ou RAM.

État de l'app : Inactive

- ✧ App au premier plan, mais ne reçoit pas d'événements
 - ✧ État de transition
 - ✧ Utilise des ressources CPU et RAM.

État de l'app : Active

- ✦ App au premier plan, reçoit des événements
 - ✦ État habituel de fonctionnement des app en premier plan
 - ✦ Utilise des ressources CPU et RAM.

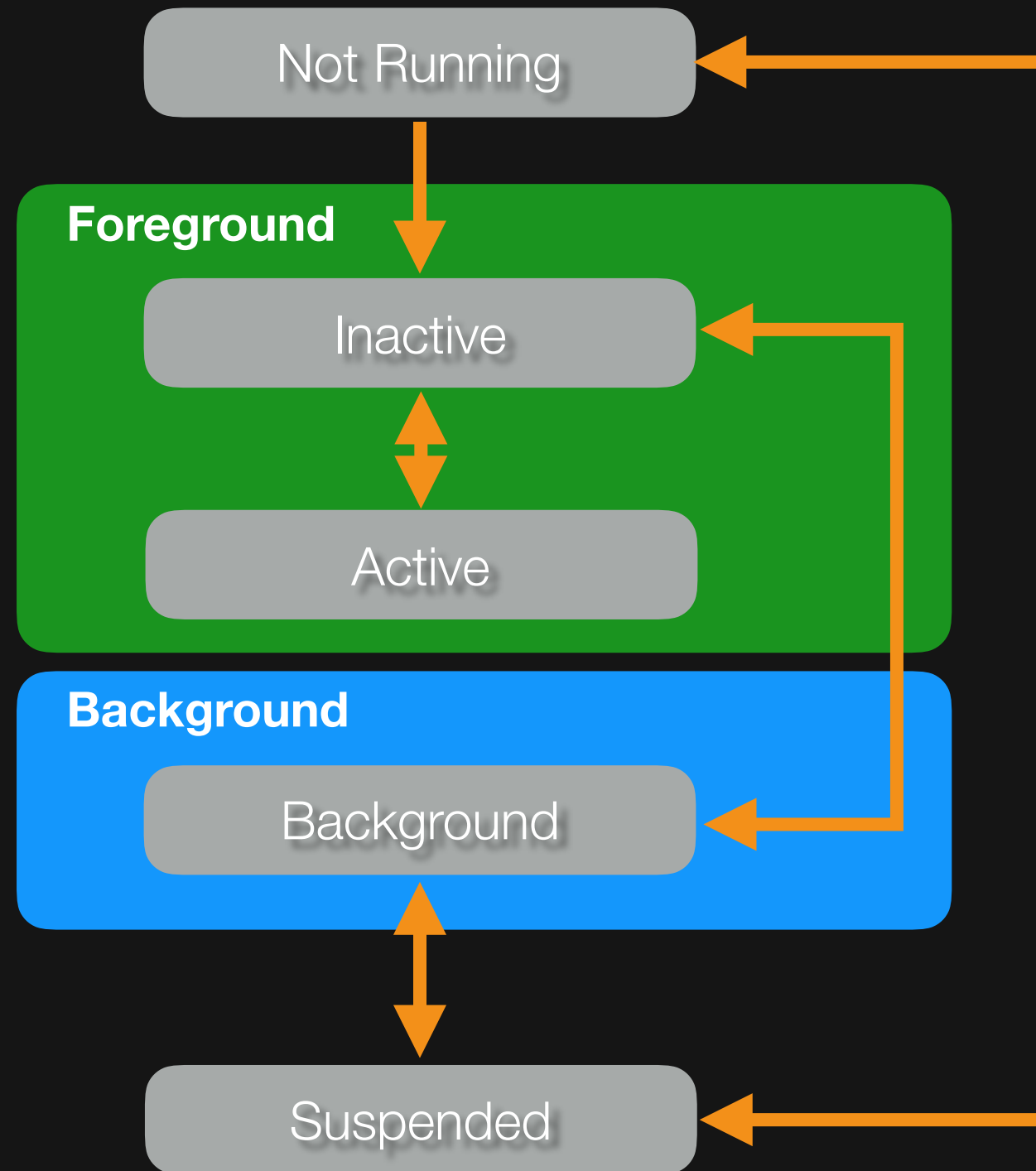
État de l'app : Background

- ✦ App en arrière plan, et exécutant du code
 - ✦ La plupart des apps entrent dans ce mode quelques secondes avant d'être suspendues
 - ✦ Certaines peuvent demander plus de temps en arrière plan
 - ✦ Une app se lançant en arrière plan rentre directement dans cet état

État de l'app : Suspended

- ✦ App en arrière plan, et n'exécutant pas de code
 - ✦ L'app reste lancée en mémoire, mais ne peut plus exécuter de code
 - ✦ En cas de situation de manque de mémoire, l'app peut être arrêtée par le système sans préavis.

États de l'app et transitions



États de l'app : Y répondre

- ✦ Des messages sont envoyés à l'AppDelegate
 - ✦ `func application(_:, willFinishLaunchingWithOptions:) -> Bool`
 - ✦ `func application(_:, didFinishLaunchingWithOptions:) -> Bool`
 - ✦ `func applicationDidBecomeActive(_:)`
 - ✦ `func applicationWillResignActive(_:)`
 - ✦ `func applicationDidEnterBackground(_:)`
 - ✦ `func applicationWillEnterForeground(_:)`
 - ✦ `func applicationWillTerminate(_:)`
- ✦ Des notifications peuvent aussi être écoutées auprès du NotificationCenter

Exécution en arrière-plan

- ✦ État “Background”
 - ✦ N’importe quelle app peut en profiter “quelques secondes” (~5)
 - ✦ Pour en profiter plus longtemps :
 - ✦ Réclamer un peu de temps supplémentaire pour finir une action
 - ✦ Utiliser un des services qui autorise le fonctionnement en arrière plan

Exécution en arrière-plan

- ✦ Demander du temps supplémentaire
 - ✦ Utilisé pour finir une tâche limitée dans le temps
 - ✦ Permet d'augmenter le temps avant la suspension
 - ✦ Se demande avec une méthode d'instance de UIApplication
 - ✦ Peut être arrêté à tout moment

Exécution en arrière-plan

```
func applicationDidEnterBackground(_ application: UIApplication) {  
    backgroundTaskID = application.beginBackgroundTask(withName: "Task") {  
        // Clean up any unfinished task business by marking where you  
        // stopped or ending the task outright.  
        application.endBackgroundTask(self.backgroundTaskID)  
        self.backgroundTaskID = UIBackgroundTaskInvalid  
    }  
  
    // Start the long-running task and return immediately.  
    DispatchQueue.global(qos: .background).async {  
        // Do the work associated with the task, preferably in chunks.  
  
        //When finished, end background task and invalidate the ID  
        application.endBackgroundTask(self.backgroundTaskID)  
        self.backgroundTaskID = UIBackgroundTaskInvalid  
    }  
}
```


Utiliser des services d'arrière plan

- ✦ Permet un fonctionnement sans limite de temps (mais avec des restrictions)
- ✦ Utilisable pour des applications qui :
 - ✦ Jouent de la musique
 - ✦ Suivent la position de l'utilisateur en temps réel
 - ✦ Utilisent de la VoIP
 - ✦ Sont des apps Newsstand
 - ✦ Utilisent des accessoires externes / bluetooth
 - ✦ Récupère des données en arrière-plan / Utilisent des notifications silencieuses

Utiliser des services d'arrière plan

Xcode background mode	UIBackgroundModes value	Description
Audio and AirPlay	audio	The app plays audible content to the user or records audio while in the background. (This content includes streaming audio or video content using AirPlay.) The user must grant permission for apps to use the microphone prior to the first use; for more information, see Supporting User Privacy .
Location updates	location	The app keeps users informed of their location, even while it is running in the background.
Voice over IP	voip	The app provides the ability for the user to make phone calls using an Internet connection.
Newsstand downloads	newsstand-content	The app is a Newsstand app that downloads and processes magazine or newspaper content in the background.
External accessory communication	external-accessory	The app works with a hardware accessory that needs to deliver updates on a regular schedule through the External Accessory framework.
Uses Bluetooth LE accessories	bluetooth-central	The app works with a Bluetooth accessory that needs to deliver updates on a regular schedule through the Core Bluetooth framework.
Acts as a Bluetooth LE accessory	bluetooth-peripheral	The app supports Bluetooth communication in peripheral mode through the Core Bluetooth framework. Using this mode requires user authorization; for more information, see Supporting User Privacy .
Background fetch	fetch	The app regularly downloads and processes small amounts of content from the network.
Remote notifications	remote-notification	The app wants to start downloading content when a push notification arrives. Use this notification to minimize the delay in showing content related to the push notification.

Utiliser des services d'arrière plan

- ✦ Pour utiliser un service :
 - ✦ Ajoutez la clé `UIBackgroundModes` au fichier `Info.plist` de l'app
 - ✦ Et y associer un tableau qui déclare l'une des clés (`audio`, `location`, `voip`, etc.)

Spécificités

Sandboxing

- ✦ Une app iOS tourne dans un bac à sable
- ✦ Accès uniquement à ses propres documents
- ✦ La majorité du système de fichiers est inaccessible à l'app
- ✦ Il faut en tenir compte lors de l'accès aux données et pour le partage des données entre apps.
- ✦ Possibilité de créer une sandbox commune à plusieurs app d'un même développeur

API publique

- ✦ Une app iOS est ne peut utiliser que les APIs publiques et officielles d'Apple
- ✦ L'utilisation de méthodes privées, non documentées entraine le rejet lors de la validation App Store
- ✦ En cas de publication en interne, pas de validation App Store, donc pas de contrôle... Mais ça reste mal !

Pour aller plus loin...



- ✦ App Programming Guide for iOS