House Course 59-20

Web and Mobile Applications
Week 6: Data Persistence, App Store Submission

Attendance: http://goo.gl/forms/QQfQMakPLc
Attendance will appear later if you miss it

Davis Gossage
Jesse Hu

Demo so far

Attendance

Attendance: http://goo.gl/forms/QQfQMakPLc

What does it mean for an app to be “turned off”?

- Two different active states:
- Active. This is the state of the app that is actively controlling the screen.
- Background. This is the state of apps that are actively running, but are not on screen. Consider the Maps app, giving directions while you check your email. Or a music app, playing audio while you look up something on the map.

What does it mean for an app to be “turned off”?

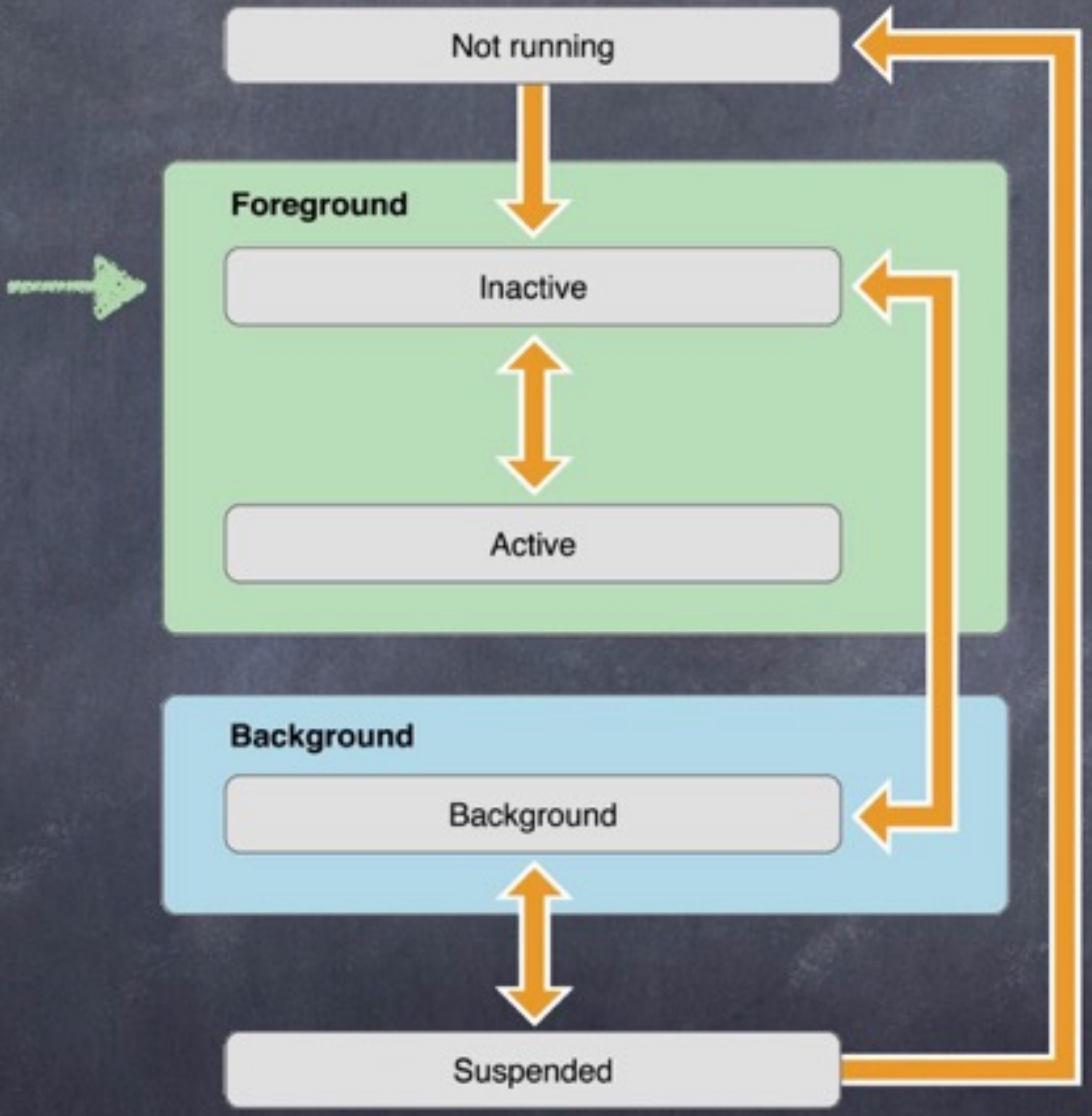
- Two states of rest:
  - Not Running. The state of apps that have never been started, for instance.
  - Inactive. This is the state of an app that was running, until the user taps the home button, or receives a phone call.
- Fifth state that is harder to see:
  - Suspended. This is the state of apps that could be running in the background, but have no work to do.

What does it mean for an app to be “turned off”?

- Of these five states, it is the not running state that most interests us.
  - This is the state that we might call “turned off”. It is the only state in which an app's memory is not preserved.
  - When apps transition in and out of the not running state we need to make sure that any state that we want to be persistent has made its way into a persistent storage mechanism.

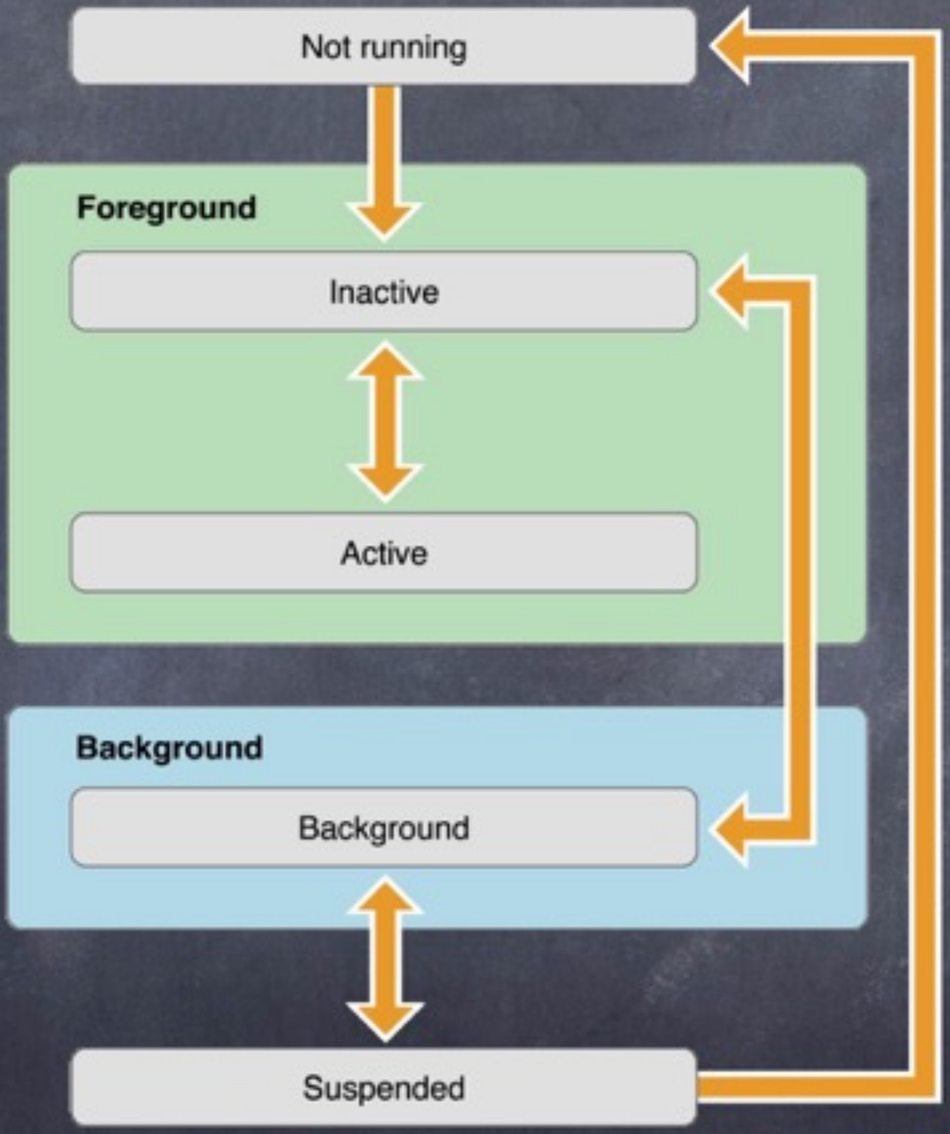
Application Lifecycle

Running your code, but no UI events.

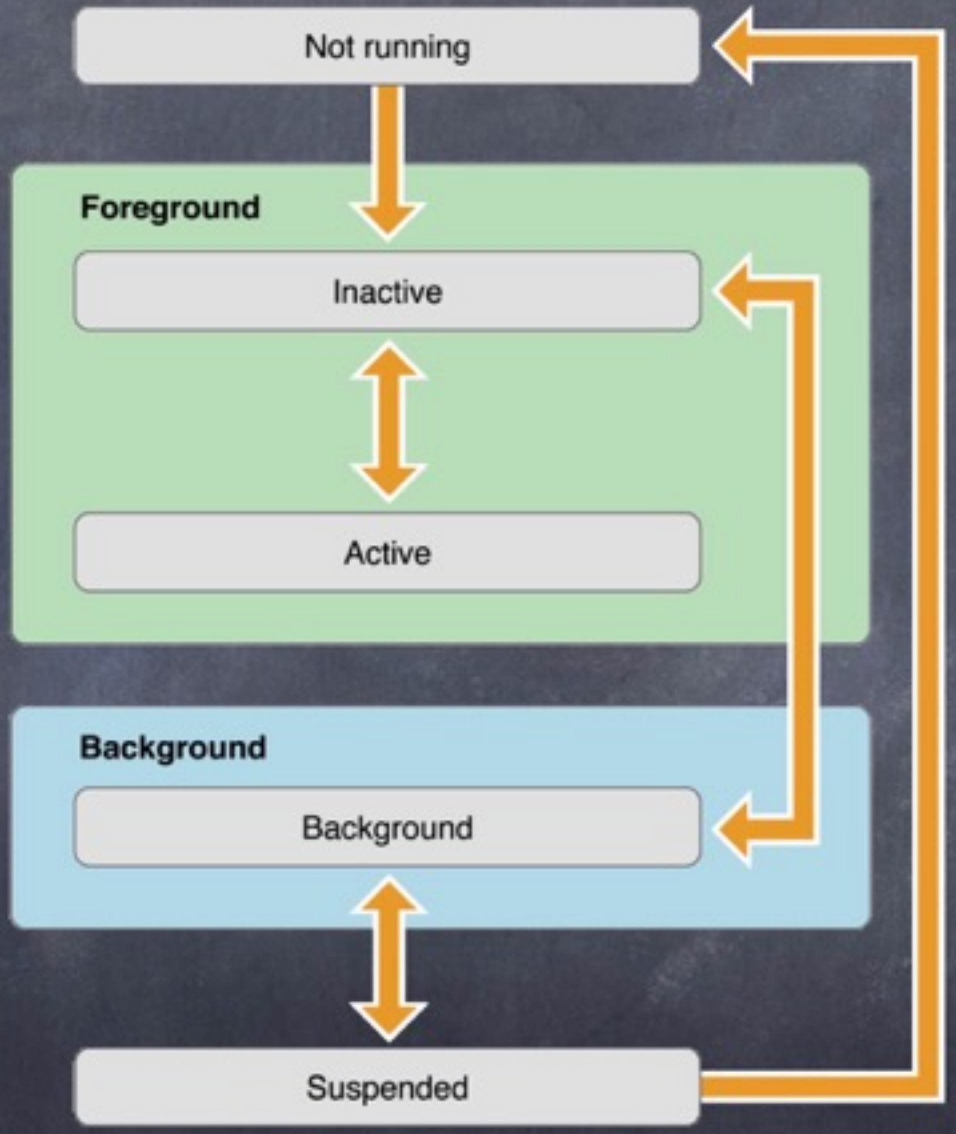


Application Lifecycle

Running your code, receiving and processing UI events.

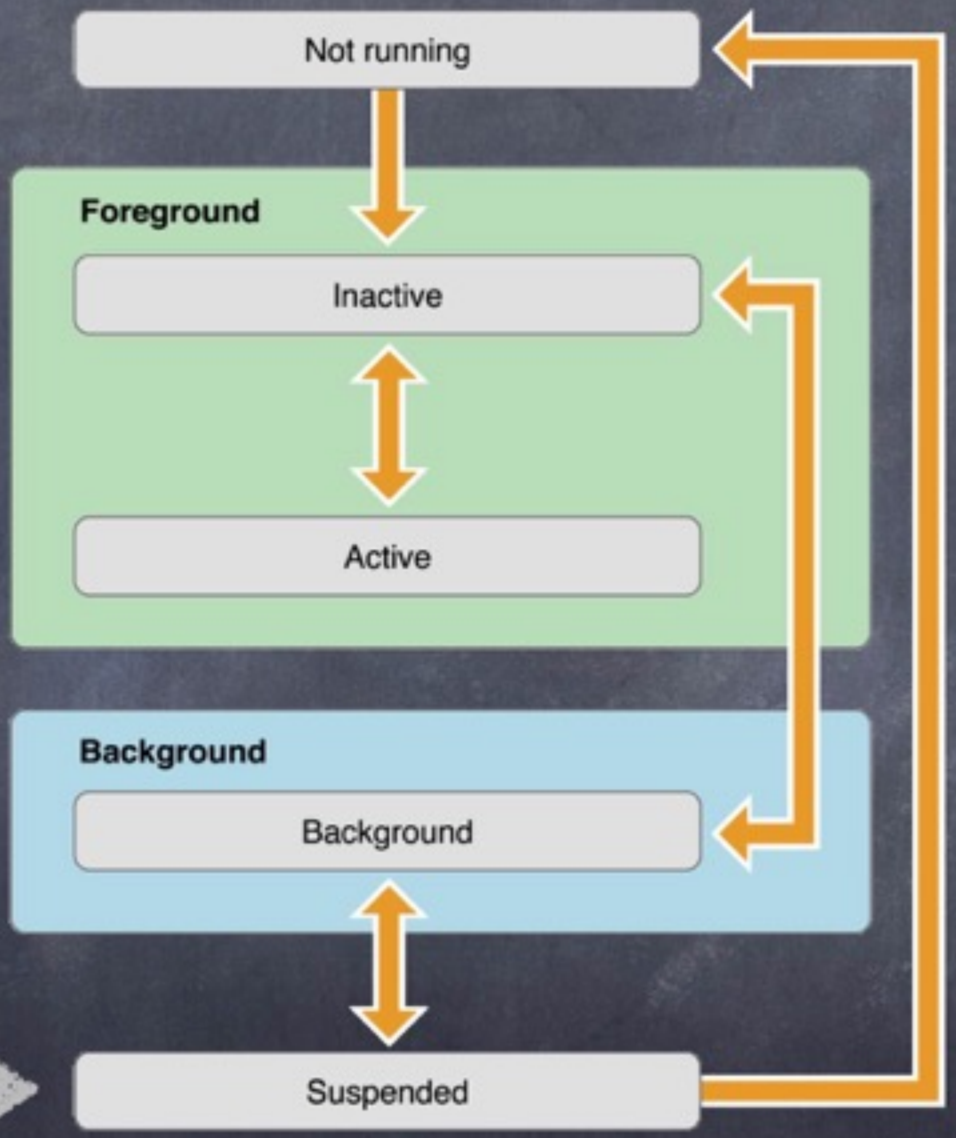


Application Lifecycle



Running your code for a limited time, no UI events.

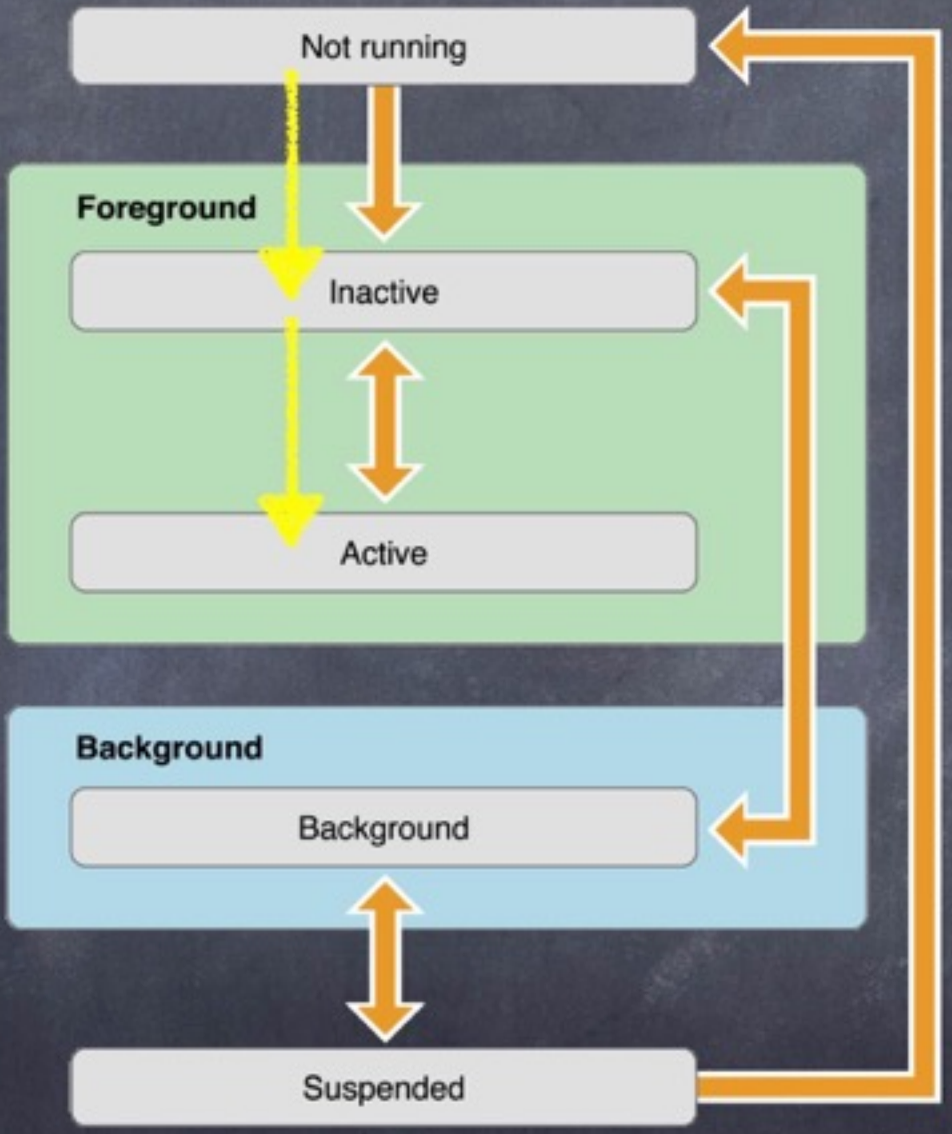
Application Lifecycle



Your code not running.
You could be killed.

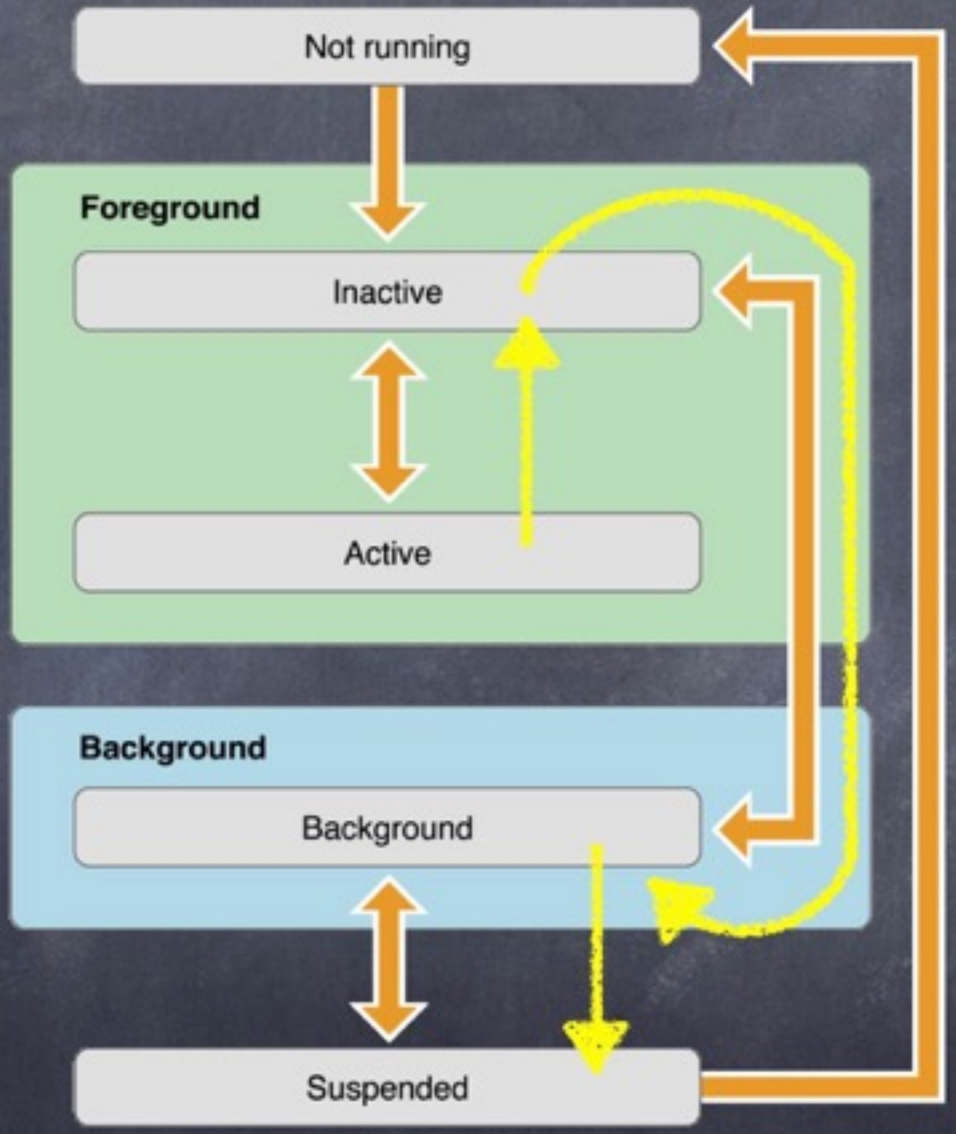
Application Lifecycle

Launch



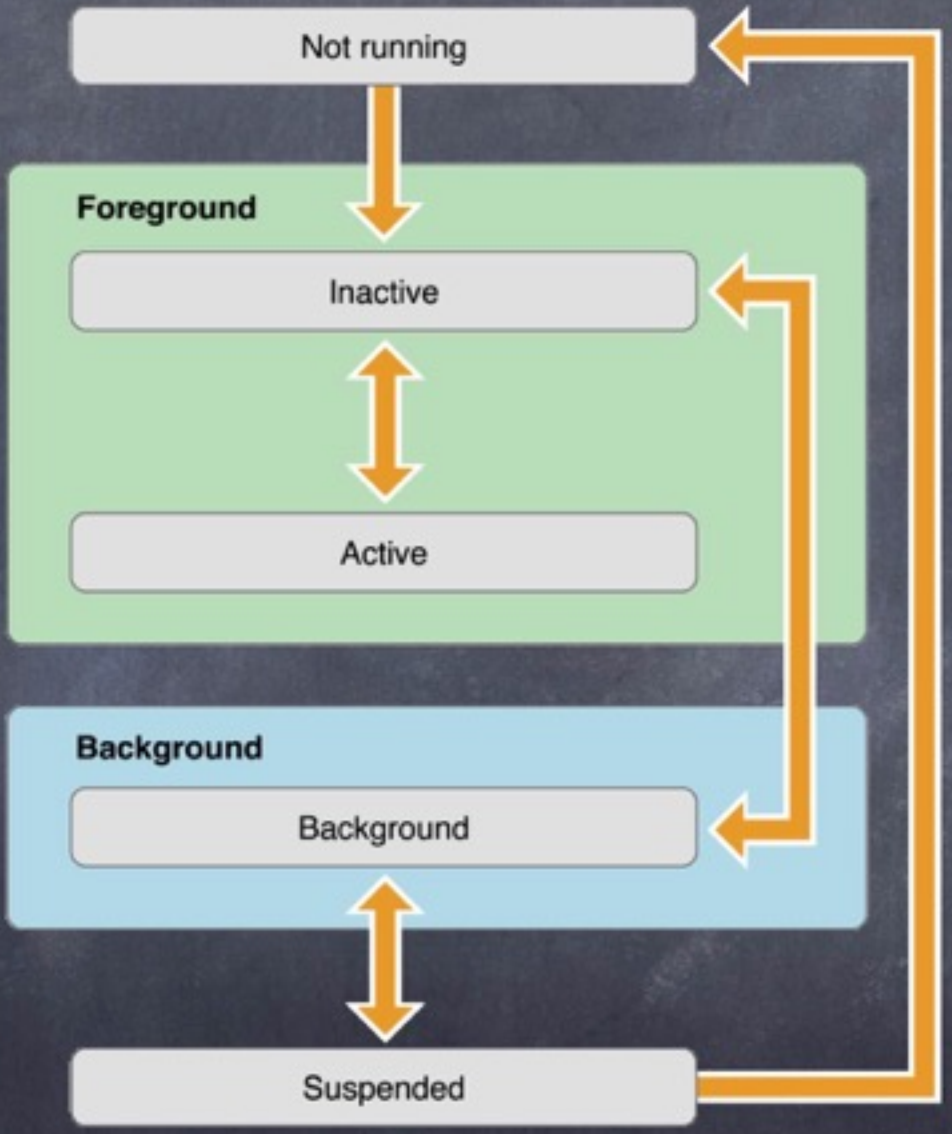
Application Lifecycle

Switch to another application



CS193p
Winter 2015

Application Lifecycle



Killed
(notice no code runs between suspended and killed)

UIApplicationDelegate

- Protocol methods provide notification of app lifecycle events
    - `application(_:willFinishLaunchingWithOptions:)`
    - `application(_:didFinishLaunchingWithOptions:)`
    - `applicationDidBecomeActive(_:)`
    - `applicationWillResignActive(_:)`
    - `applicationDidEnterBackground(_:)`
    - `applicationWillEnterForeground(_:)`
    - `applicationWillTerminate(_:)`

Persistence

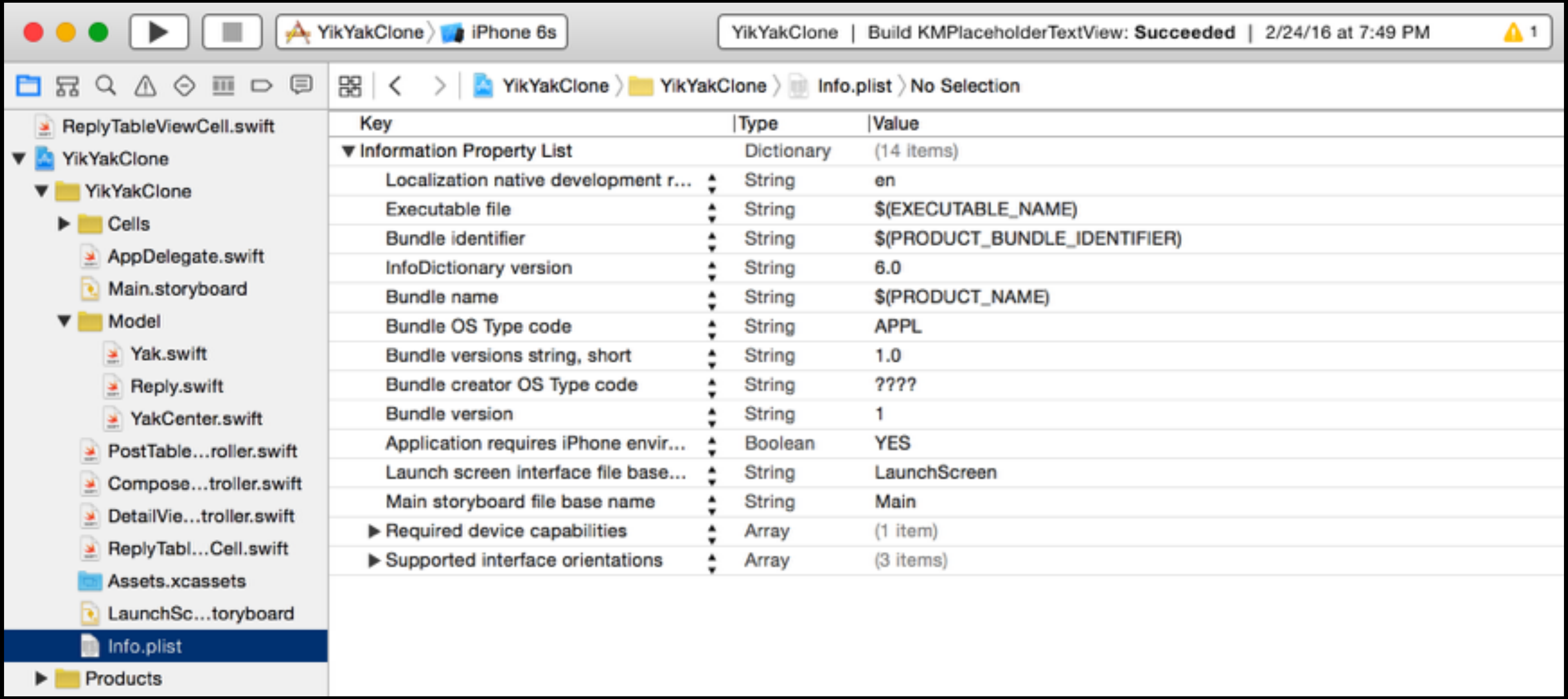
- Property Lists
- NSUserDefaults
- Archiving
- SQLite
- File System
- Core Data
- mBaaS (Firebase, Parse)

Property Lists

- Only viable for very small data sets.
- Convenient but they do not scale well.
  - Many of your application’s settings are in **Info.plist**
  - You can edit this file (in Xcode’s property list editor) by clicking on Info.plist

Property Lists

- Many of your application's settings are in **Info.plist**
- Key-value storage



YikYakClone 〉 iPhone 6s

YikYakClone | Build KMPlaceholderTextView: **Succeeded** | 2/24/16 at 7:49 PM

YikYakClone 〉 YikYakClone 〉 Info.plist 〉 No Selection

```
ReplyTableViewCell.swift
▼ YikYakClone
  ▼ YikYakClone
    ▶ Cells
      AppDelegate.swift
      Main.storyboard
    ▼ Model
        Yak.swift
        Reply.swift
        YakCenter.swift
      PostTable...roller.swift
      Compose...troller.swift
      DetailVie...troller.swift
      ReplyTabl...Cell.swift
      Assets.xcassets
      LaunchSc...toryboard
      Info.plist
  ▶ Products
```

Key	Type	Value
▼ Information Property List	Dictionary	(14 items)
Localization native development r...	String	en
Executable file	String	$(EXECUTABLE_NAME)
Bundle identifier	String	$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version	String	6.0
Bundle name	String	$(PRODUCT_NAME)
Bundle OS Type code	String	APPL
Bundle versions string, short	String	1.0
Bundle creator OS Type code	String	????
Bundle version	String	1
Application requires iPhone envir...	Boolean	YES
Launch screen interface file base...	String	LaunchScreen
Main storyboard file base name	String	Main
▶ Required device capabilities	Array	(1 item)
▶ Supported interface orientations	Array	(3 items)

NSUserDefaults

- Apple's easy-to-use infrastructure for saving app information associated with user preferences.
- Open-ended. We can store primitive types and object types. It behaves like a dictionary—data is stored as keys and values.
- iOS keeps a database of these user preference values for each app on the device. The values are persisted from one run of an app to another.
- "Singleton Design Pattern" (shared instance):

```
let defaults =
NSUserDefaults.standardUserDefaults()

defaults.setObject("Coding Explorer", forKey:
"userNameKey"
```

File System

- iOS has a Unix filesystem underneath it
- You can read and write files into it with some restrictions
- Can store files in the "documents" directory, using NSFileManager

```
let dirPath =
NSSearchPathForDirectoriesInDomains(.DocumentDirectory,
.UserDomainMask, true)[0] as! String
let pathArray = [dirPath, filename]
let fileURL =
NSURL.fileURLWithPathComponents(pathArray)!
```

Archiving

- Very rarely used for persistence, but it is how storyboards are made persistent
- objects in the graph to implement NSCoding protocol
  - `func encodeWithCoder(encoder: NSCoder` and `init(coder: NSCoder)`

SQLite

- SQLite is good if you prefer a more database-like approach and don't mind writing code to translate between SQL records and model objects.
- Rarely used unless you have a legacy SQL database you need to access

Core Data

- An object-oriented database
- Primary way to store data in a sophisticated application
- Core Data is good if you like the idea of being able to read and write your model objects directly.
- It offers the same kind of searching and filtering you might expect with SQL except that you deal with your own objects all the way through.
- Integration with iCloud

Firebase Persistence

- Firebase apps automatically handle temporary network interruptions for you.
    - Cached data will still be available while offline and writes will be resent when network connectivity is recovered.
- One line:

```
Firebase.defaultConfig().persistenceEnabled = true
```

Firebase Transactions

- When an app needs realtime bidding, voting, etc. the traditional setValue concept breaks
- Use transactions

App Distribution Overview

- Your identity (certificate) is of type “iOS Development”
  - This allows you to load apps onto devices connected to your Xcode environment.
  - Provisioning profiles are updated automatically for you by Xcode, or you can register device in Member Center
  - You can managed your identities and provisioning profiles in Xcode -> Preferences -> Accounts
- To actually distribute an app, an “iOS Distribution” identity is required

Credit: Ric Telford ECE 590

Archiving and Distribution

- There are two kinds of distribution of your app:
  1. Ad Hoc Distribution – you provide a copy of your app to a limited set of known users so that they can try it on their devices to report bugs. See *TestFlight* for more info
  2. App Store Distribution – you provide the app to the App Store so that anyone can download and run it.
- To create a copy of your app for distribution, you need first to build an *archive* of the app
  - A "preserved" build that you can use for distribution, reproduction, and symbolication (viewing crash logs)
  - Set device to "iOS Device" and then Product -> Archive
- To actually distribute the archive, you would then need an "iOS Distribution" identity

Credit: Ric Telford ECE 590

Submission to the App Store

- The Apple portal page for the app store is http://itunesconnect.apple.com .
  - Provides App management, App analytics, Sales and Trends, Payments and iAd
  - See *iTunes Connect Developer's Guide* for more information
- Key information you will need to supply at some point:
  - App name – the name that will appear in the App Store. 25 characters or fewer, unique to App Store
  - Description – fewer than 4,000 characters, pure text.
  - Keywords – comma-separated
  - SKU number / string – you generate, unique to your apps
  - Support site URL, Copyright, Price, Avail Date
- Export App to .ipa using Window -> Organizer
- Upload App to iTunes using Xcode -> Open Developer Tool -> Application Loader

Credit: Ric Telford ECE 590