Mobile Application Development
Aileen Pierce

APP LIFECYCLE

Single View App

- The single-view app template creates the following files:
  - ViewController.swift
    - view controller implementation file
  - Main.storyboard
    - Storyboard file for your views
  - Info.plist
    - Application configuration property list

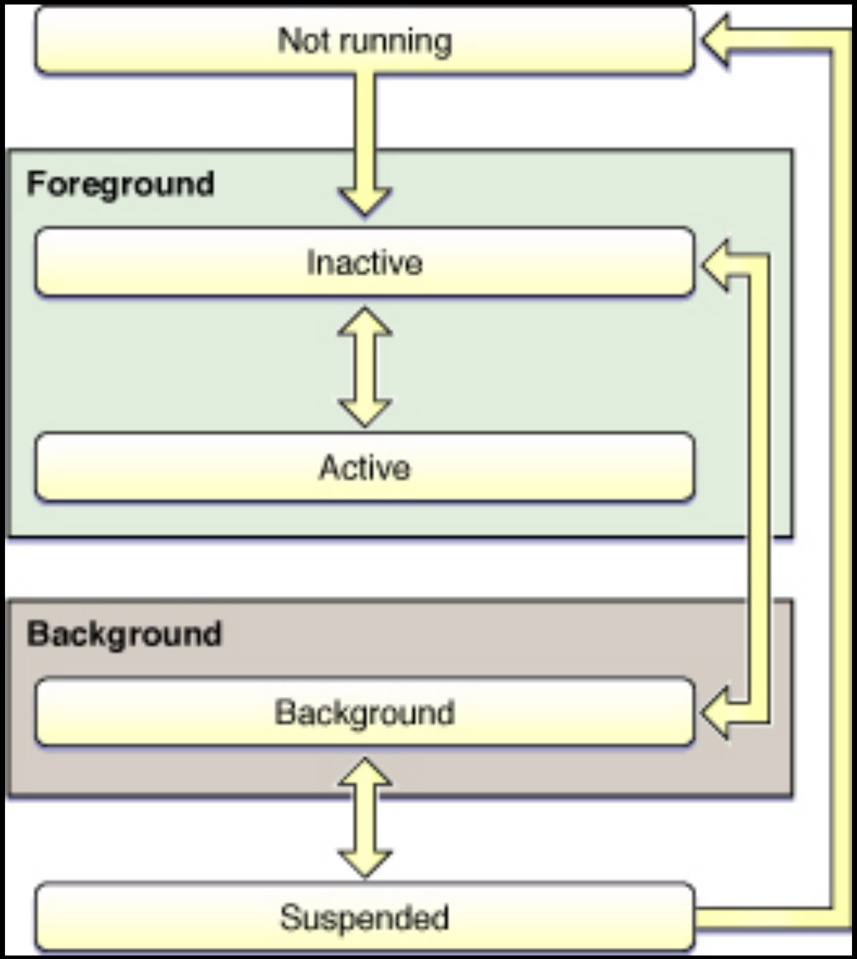
Application Delegate

– AppDelegate.swift
  - Implements the **`UIApplicationDelegate`** protocol
    – ensures your app interacts properly with the system and other apps
  - Creates the variable for **`UIWindow`**
  - Stub methods for applicationDidThis and applicationWillDoThat
  - **`application(_, didFinishLaunchingWithOptions`**)
    – Tells the delegate that the launch process is almost done and the app is almost ready to run

App Lifecycle

- Launch
- Initialize
- Process
- Respond
- Terminate

App State Changes



App States

- Not running
  - Not yet launched or was terminated
- Active
  - Running and receiving input and events
- Background
  - In the background and executing code
- Suspended
  - In the background and not executing code
- Inactive
  - Running in the foreground but not receiving events.

App State Transitions

- Most state transitions are accompanied by a corresponding call to an app delegate method
  - **`application(_, willFinishLaunchingWithOptions)`**
    - The app's first chance to execute code at launch time
  - **`application(_, didFinishLaunchingWithOptions)`**
    - The app's chance to perform any final initialization
  - **`applicationDidBecomeActive()`**
    - The app's chance to prepare to run as the foreground app
  - **`applicationWillResignActive()`**
    - The app is transitioning away from being in the foreground

App State Transitions

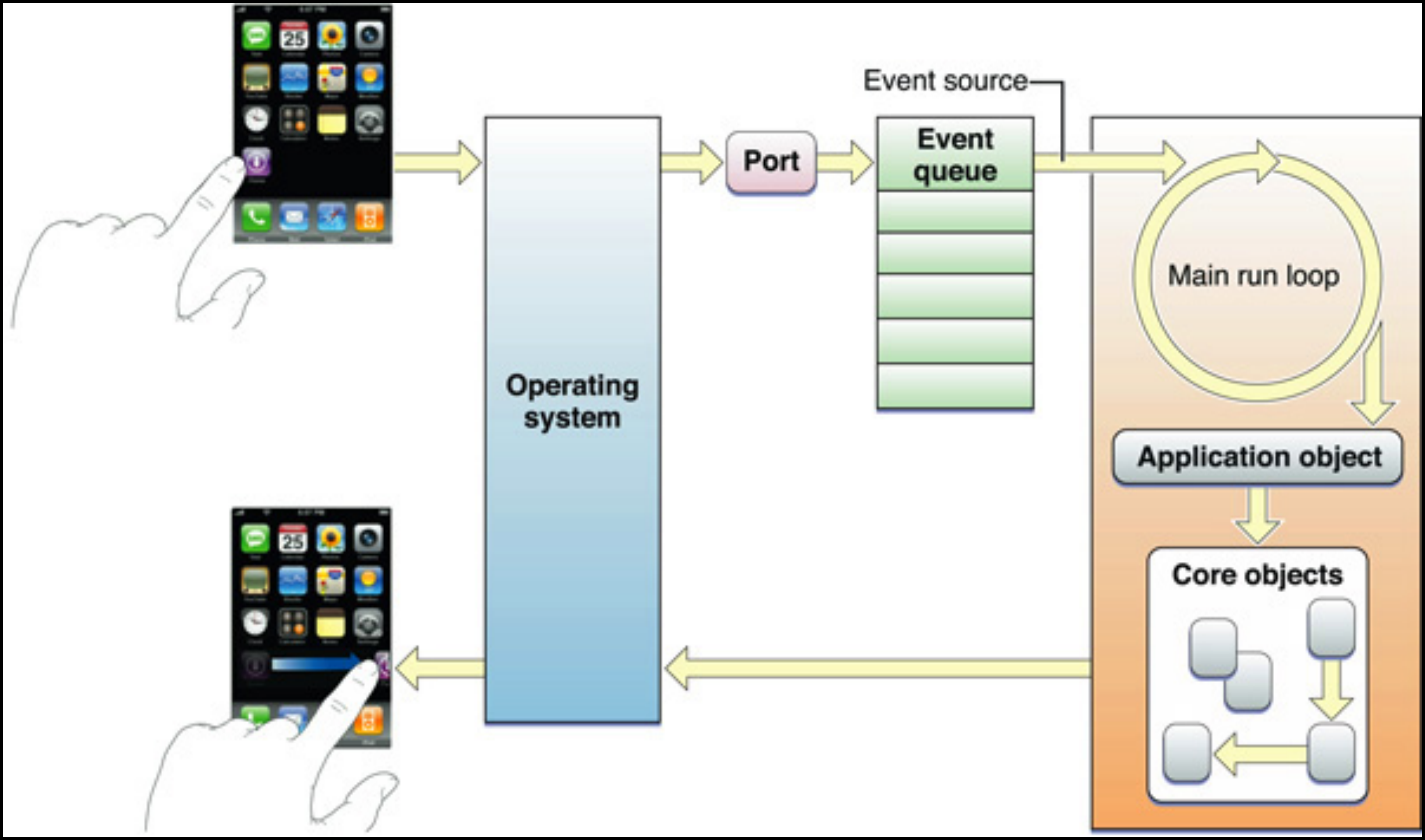
- **`applicationDidEnterBackground()`**
  - The app is now running in the background and may be suspended at any time.
- **`applicationWillEnterForeground()`**
  - The app is moving out of the background and back into the foreground, but it is not yet active.
- **`applicationWillTerminate()`**
  - The app is being terminated.
  - This method is not called if your app is suspended.

- There are also matching notifications for each state change

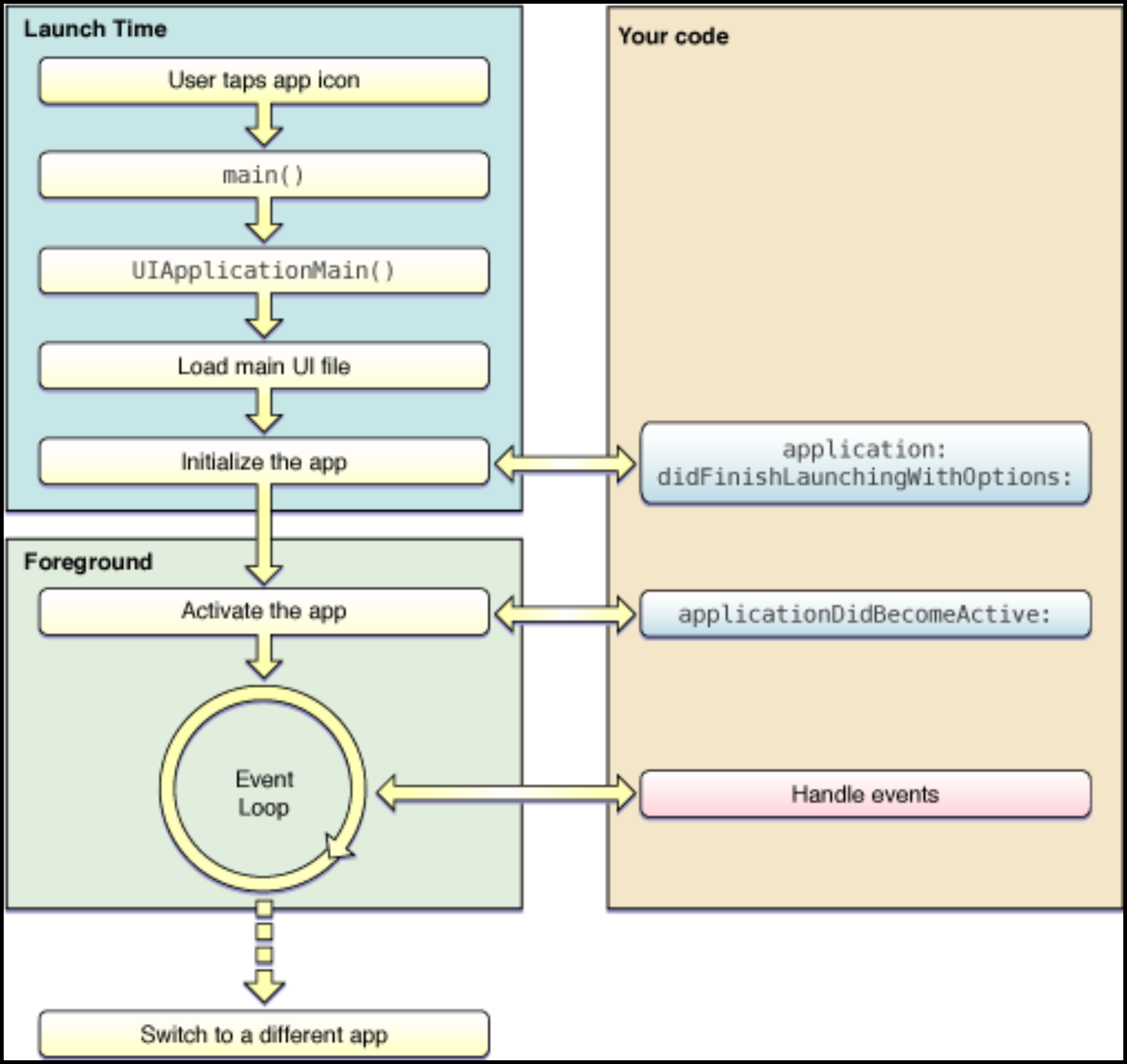
App Launch Cycle

- When an app is launched it moves from the not running state to the active state.
- Calls its **`main()`** function
- **`main()`** hands control over to the UIKit framework
- Calls **`UIApplicationMain()`**
  - – Creates your UIApplication object, window, view controller, and interface file.

Processing Events



App Launching into Foreground



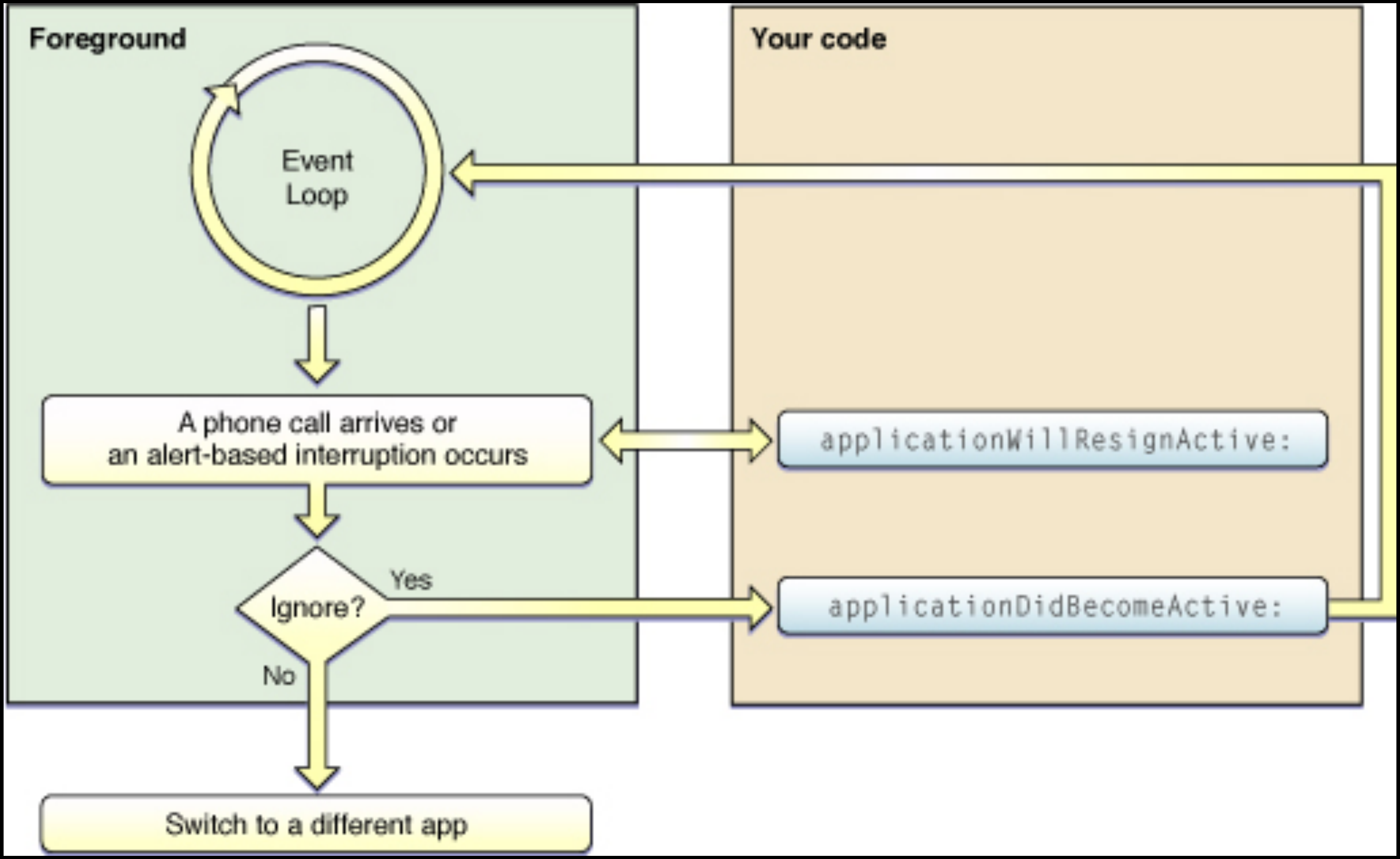
App Lifecycle

- **`application:didFinishLaunchingWithOptions:`** the application enters a run loop that does the following:
  - Creates a memory pool
  - Waits for events
  - Events are dispatched through UIKit objects
  - Screen is updated when needed
  - Memory pool is drained
  - Repeat

App Launch

- **`viewDidLoad`**()
  - Great place to initialize anything you couldn't in IB
- **`viewWillAppear()`**
  - View just about to be put on screen
  - Bounds of the view get set
  - Called each time the view appears

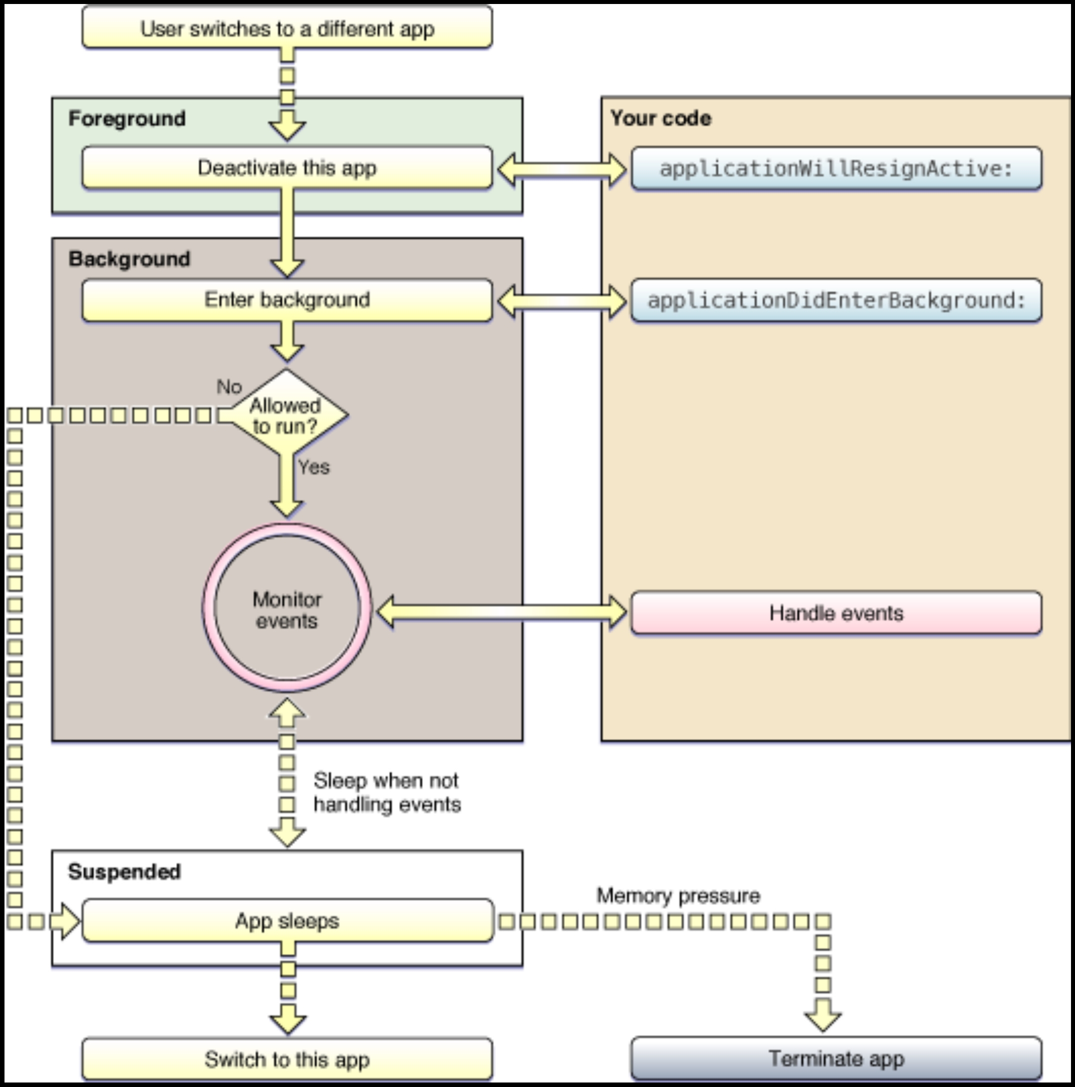
Interruptions


Foreground
Your code
Event Loop
A phone call arrives or an alert-based interruption occurs
applicationWillResignActive:
Ignore?
Yes
No
applicationDidBecomeActive:
Switch to a different app

Interruptions

- Interruptions cause an app to go into an inactive state.
- When a user quits an app its process is not terminated, just moved to the background
- The app and its objects are still in memory and are not recreated when the app is relaunched
- **`applicationWillResignActive()`** is called when an app is about to move from the active to inactive state
  - All ongoing tasks should be paused

Background

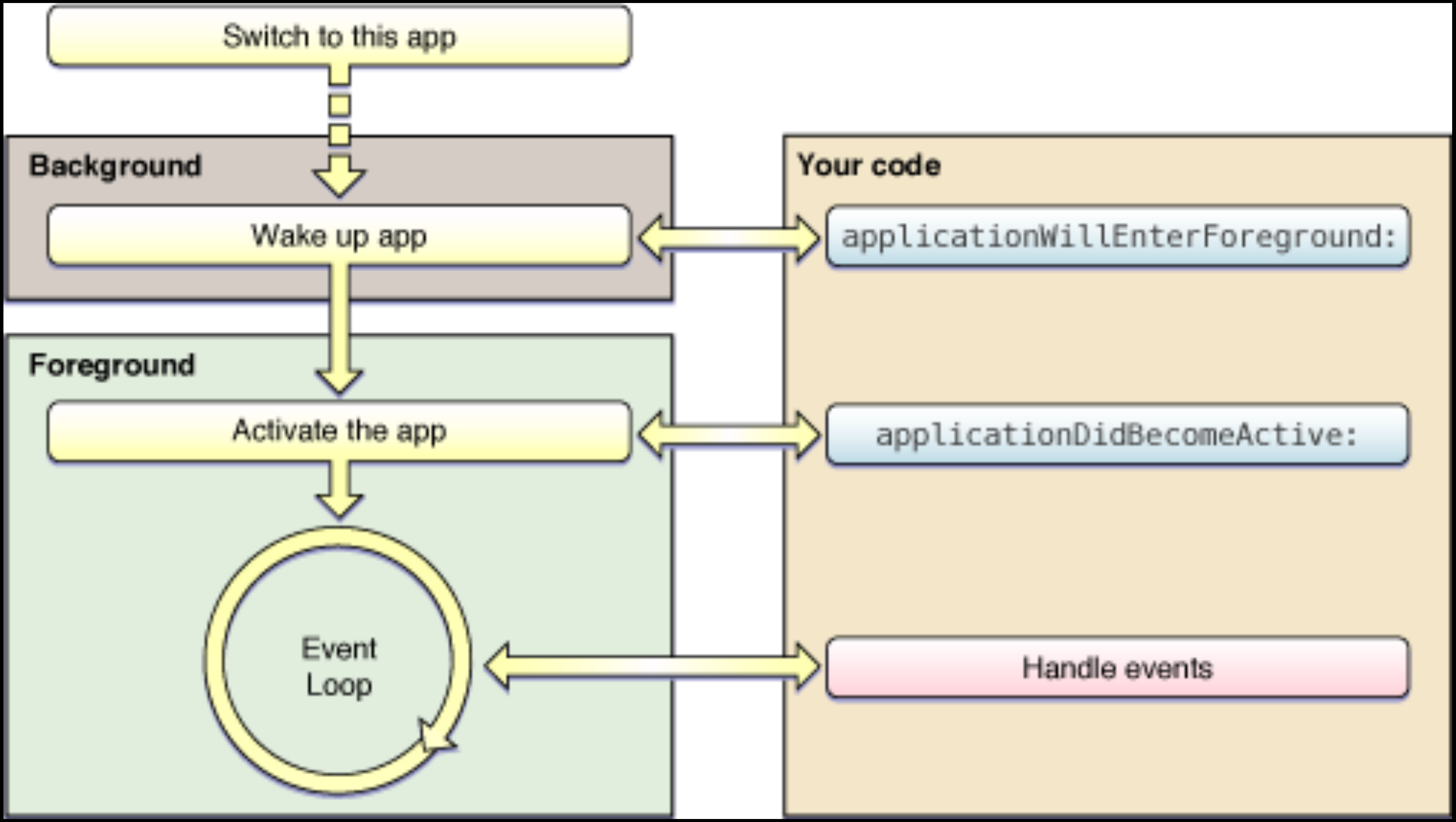


User switches to a different app
Foreground
Deactivate this app
Your code
applicationWillResignActive:
Background
Enter background
applicationDidEnterBackground:
No
Allowed to run?
Yes
Monitor events
Handle events
Sleep when not handling events
Suspended
App sleeps
Memory pressure
Switch to this app
Terminate app

Background

- When the system launches another app (such as when the user presses the Home button) your app moves into the background state.
  - **`applicationWillResignActive()`**
  - **`applicationDidEnterBackground()`**
  - Save any unsaved data or state
  - Your app has 5 seconds in this state
- When the user goes back to the app it moves to the foreground state.
  - **`applicationWillEnterForeground()`**
- Then the app becomes active
  - **`applicationDidBecomeActive()`**
  - Restart anything stopped and get ready to handle events again.

Returning to the Foreground



Returning to the Foreground

- When your app returns to the foreground it restarts the tasks that it stopped when it went to the background.
  - **`applicationWillEnterForeground()`**
  - **`applicationDidBecomeActive()`**
  - perform the same activation tasks that it would at launch time.

Termination

- When your app is terminated
  - **`applicationWillTerminate()`**
  - Perform any needed cleanup
  - Save user data or state information
  - Has 5 seconds to clean up
  - Not called if the app is currently suspended
- Can terminate due to memory constraints
- Must be prepared for your app to be killed without any notification.

Memory

- **`applicationDidReceiveMemoryWarning()`** is sent low-memory notifications
- Implement the **`didReceiveMemoryWarning()`** method in your view controller class to release views or other controllers
- Not freeing up enough memory will result in **`applicationWillTerminate()`** being called and your app terminating