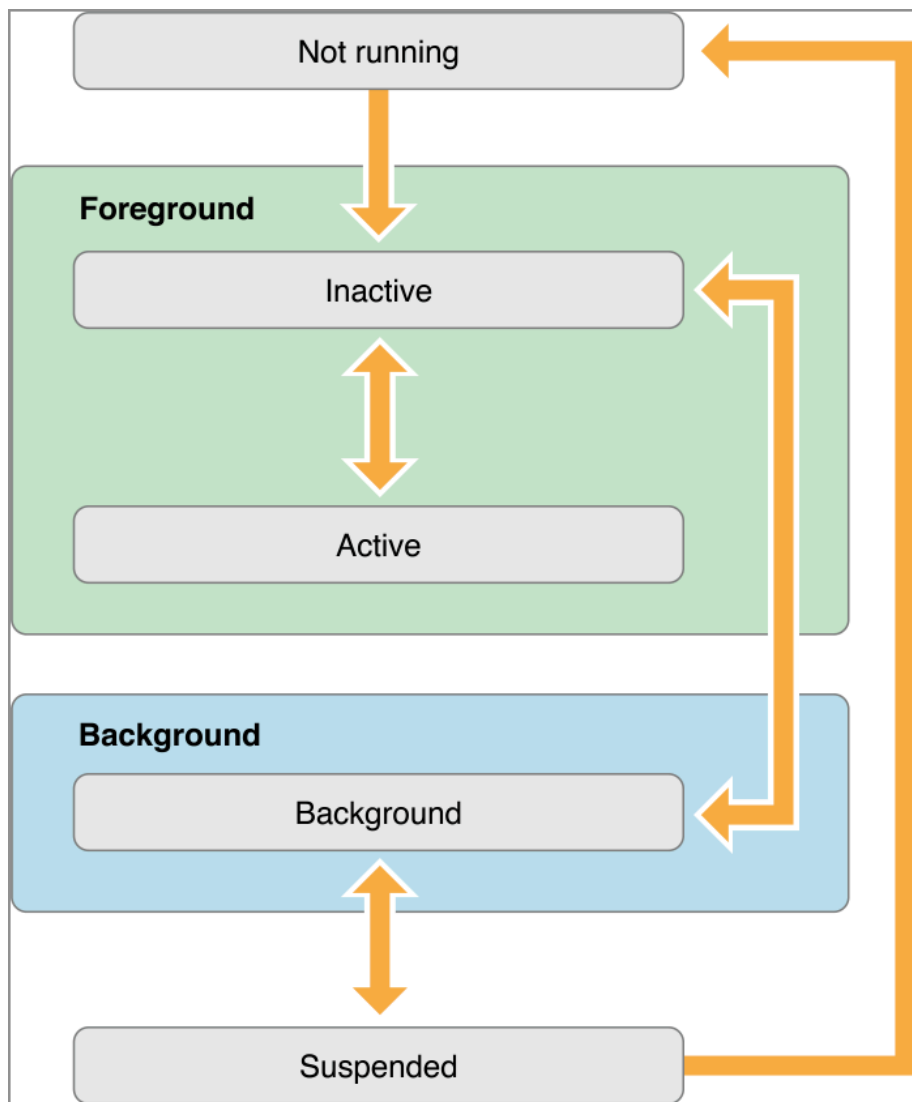


App States and Multitasking

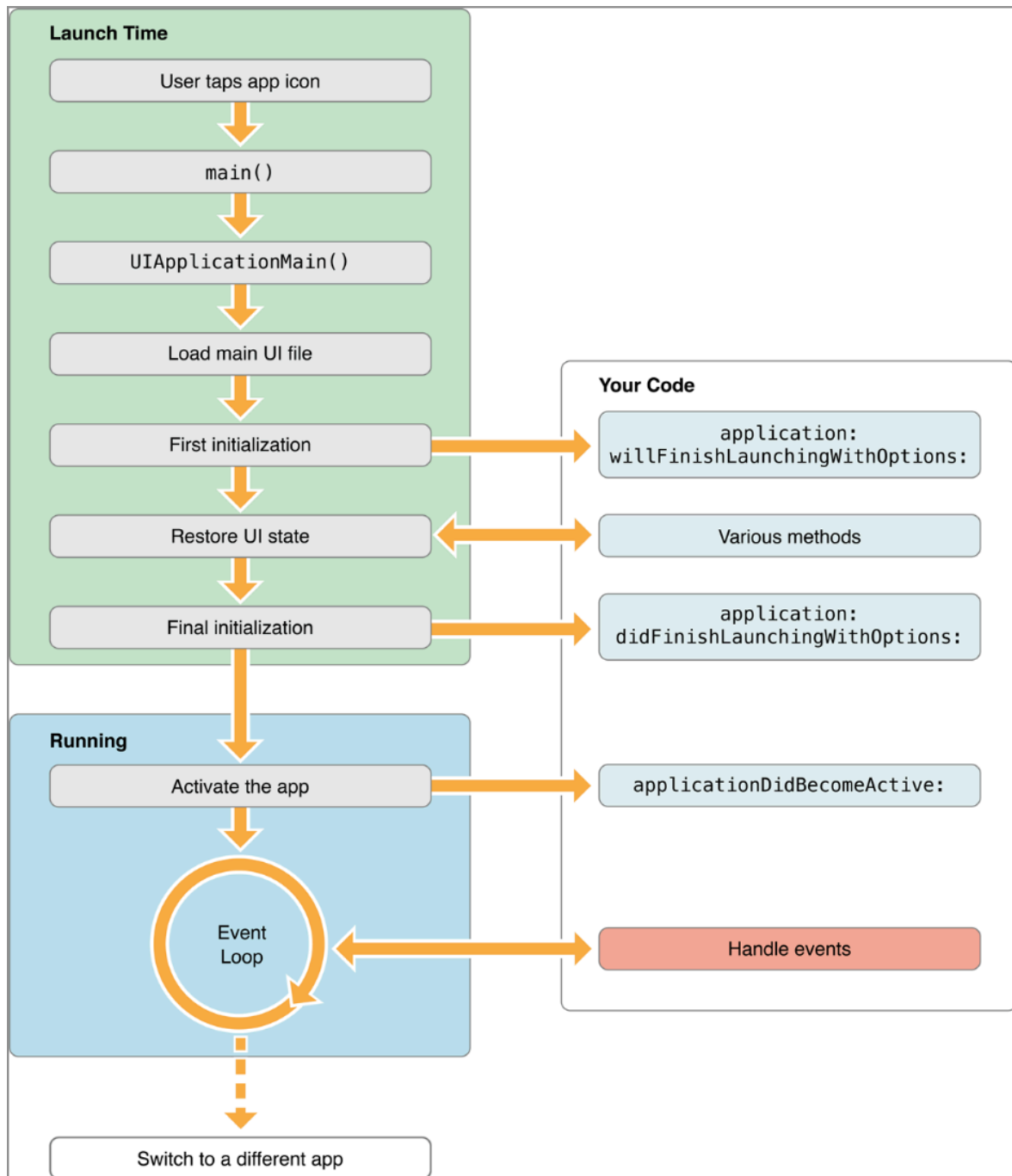
Managing App State Changes

State	Description
Not running	The app has not been launched or was running but was terminated by the system.
Inactive	The app is running in the foreground but is currently not receiving events. (It may be executing other code though.) An app usually stays in this state only briefly as it transitions to a different state.
Active	The app is running in the foreground and is receiving events. This is the normal mode for foreground apps.
Background	The app is in the background and executing code. Most apps enter this state briefly on their way to being suspended. However, an app that requests extra execution time may remain in this state for a period of time. In addition, an app being launched directly into the background enters this state instead of the inactive state. For information about how to execute code while in the background, see “Background Execution and Multitasking.”
Suspended	The app is in the background but is not executing code. The system moves apps to this state automatically and does not notify them before doing so. While suspended, an app remains in memory but does not execute any code. When a low-memory condition occurs, the system may purge suspended apps without notice to make more space for the foreground app.

State changes in an iOS app



> The App Launch Cycle



> About the main Function

```
#import <UIKit/UIKit.h>
```

```
int main(int argc, char *argv[])
```

```
{
```

```
    @autoreleasepool {
```

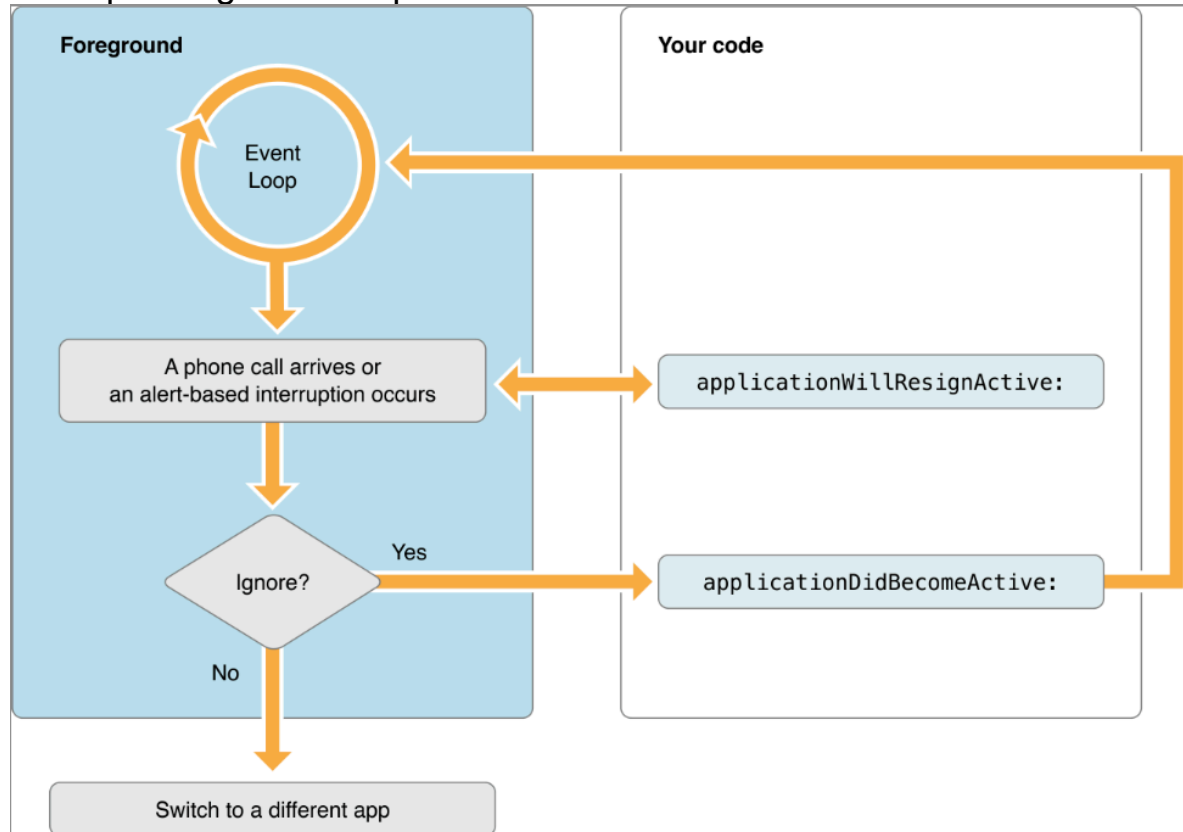
```
        return UIApplicationMain(argc, argv, nil,
    NSStringFromClass([MyAppDelegate class]));
```

```
    }
```

```
}
```

> What to Do at Launch Time

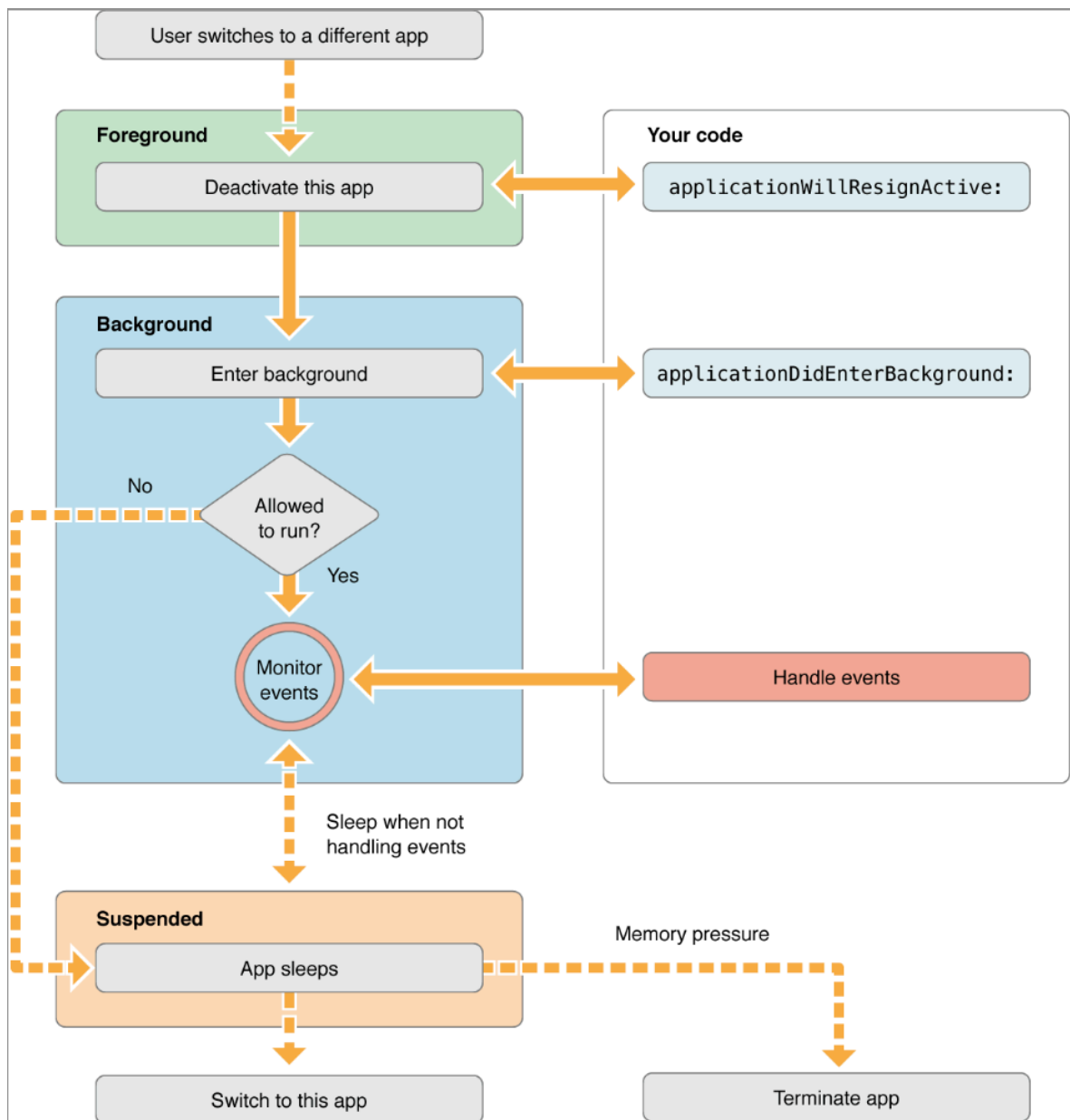
> Responding to Interruptions



> What to Do When an Interruption Occurs

> Adjusting Your User Interface During a Phone Call

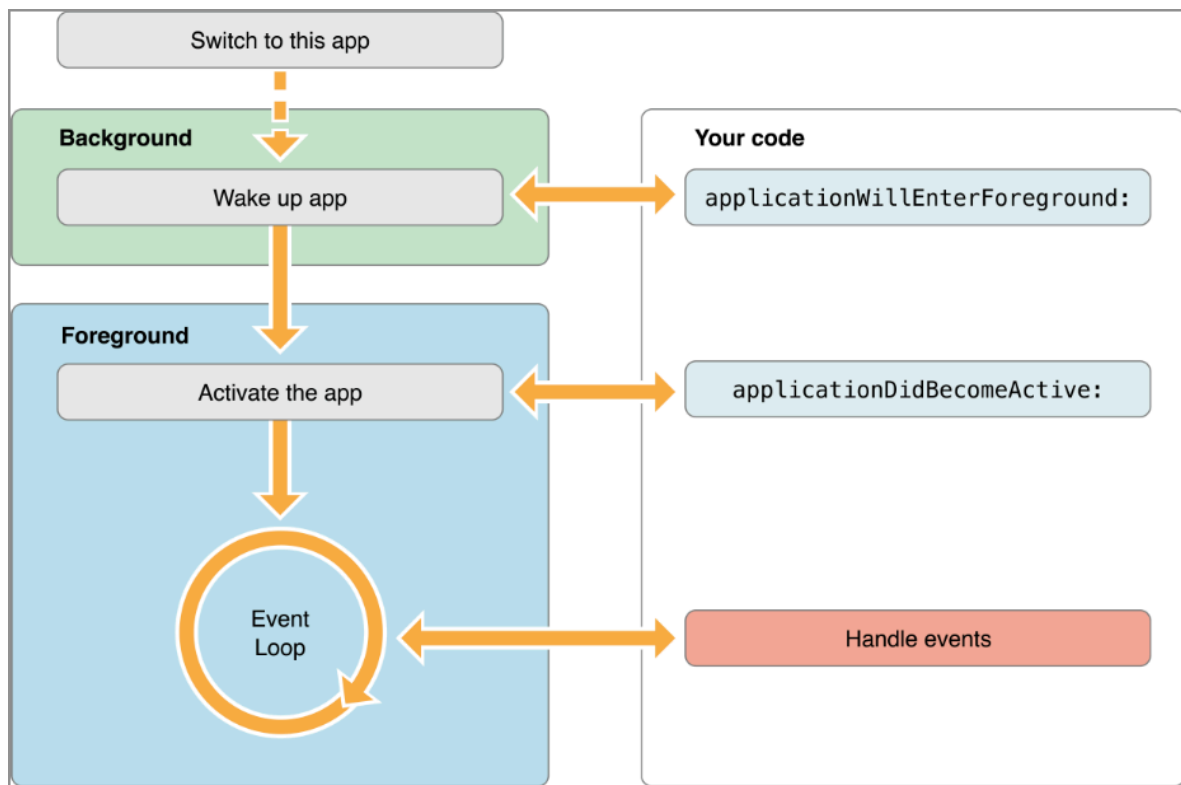
> Moving to the Background



> What to Do When Moving to the Background

> Memory Usage for Background Apps

> Returning to the Foreground



> Processing Queued Notifications at Wakeup Time

> Handling iCloud Changes

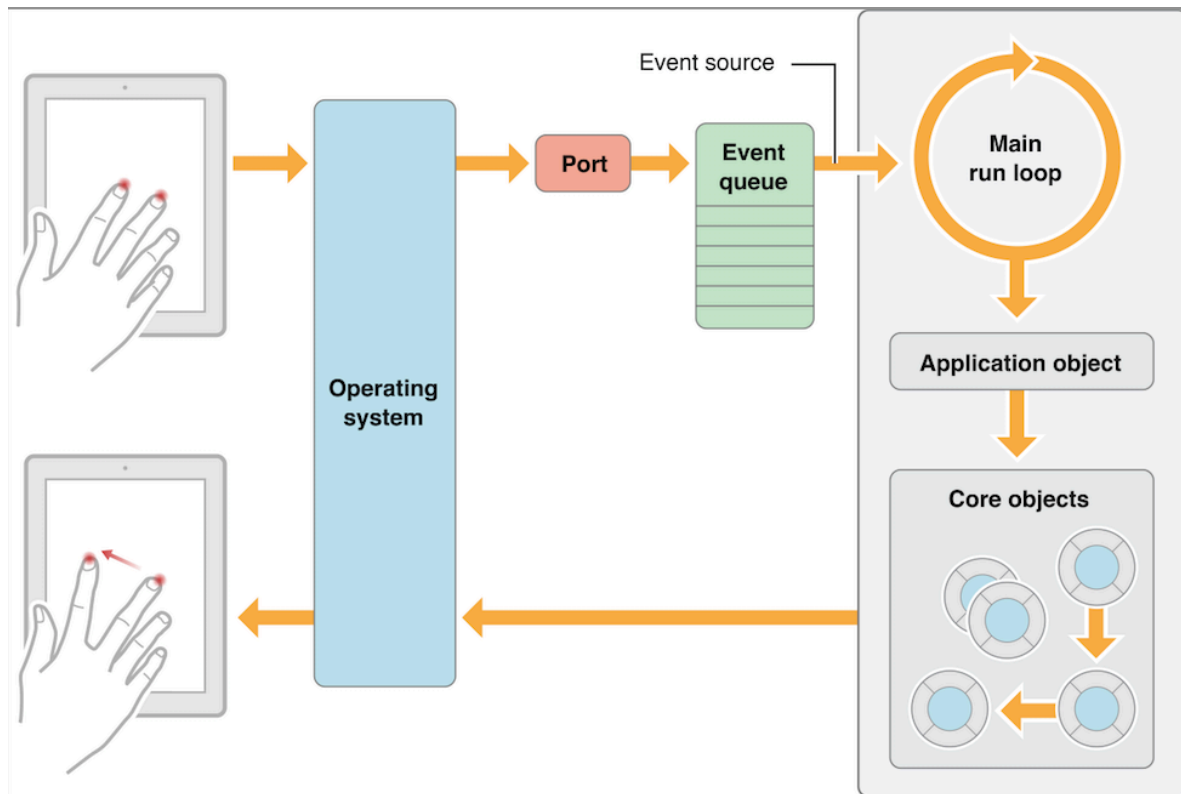
> Handling Locale Changes Gracefully

> Responding to Changes in Your App's Settings

> App Termination

The Main Run Loop

Processing events in the main run loop



> Background Execution and Multitasking

guidelines for background execution

1. You need to implement at least one of several specific user services.
2. You need to perform a single finite-length task.
3. You need to use notifications to alert the user to some relevant piece of information when your app is not running.

> Determining Whether Multitasking Is Available

Checking for background support in earlier versions of iOS

```

UIDevice* device = [UIDevice currentDevice];
BOOL backgroundSupported = NO;
if ([device respondsToSelector:@selector(isMultitaskingSupported)])
    backgroundSupported = device.multitaskingSupported;

```

> Executing a Finite-Length Task in the Background

> Scheduling the Delivery of Local Notifications

> Implementing Long-Running Background Tasks

> Declaring Your App's Supported Background Tasks

> Tracking the User's Location

- > Playing Background Audio
- > Implementing a VoIP App
- > Downloading Newsstand Content in the Background
- > Communicating with an External Accessory
- > Communicating with a Bluetooth Accessory
- > Being a Responsible Background App
- > Opting out of Background Execution

Concurrency and Secondary Threads