

Introduction of

iOS Application Development &

Swift Programming Language

Presented by

Chii Chang

chang@ut.ee



UNIVERSITY OF TARTU
Institute of Computer Science

Outlines

- Basic understanding about iOS App Development
- Development environment: Xcode IDE
- Foundations and Tools
- Introduction of Swift programming language

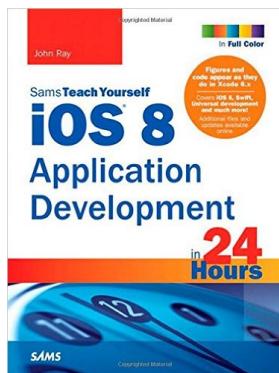
References



Free on iBooks store



Free online (partial content)
http://www.techotopia.com/index.php/IOS_8_App_Development_Essentials



Basic app

- <http://swift-tutorials.com/tutorials/>

What is iOS App Development? Why you should care?



Why iOS apps have higher revenue?

1. iOS-device users are willing to pay for apps
2. Billing issue in developing countries for Google Play store
3. Less pirate apps in iOS (close platform)

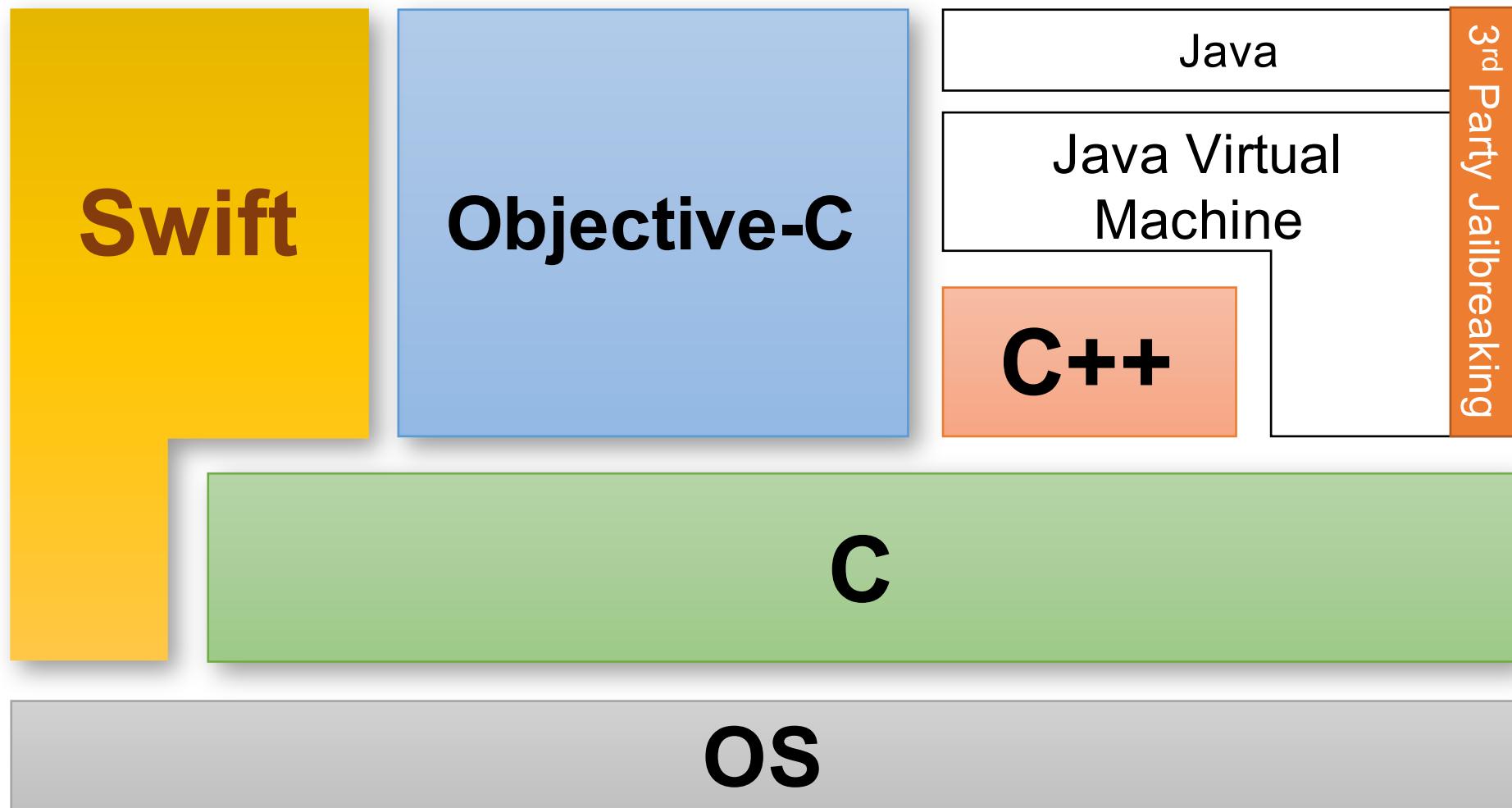
<http://thenextweb.com/apps/2015/07/15/app-annie-report-google-plays-downloads-top-the-app-store-as-apple-retains-revenue-lead/>

by JACKIE DOVE, 15 Jul, 03:00pm in APPS

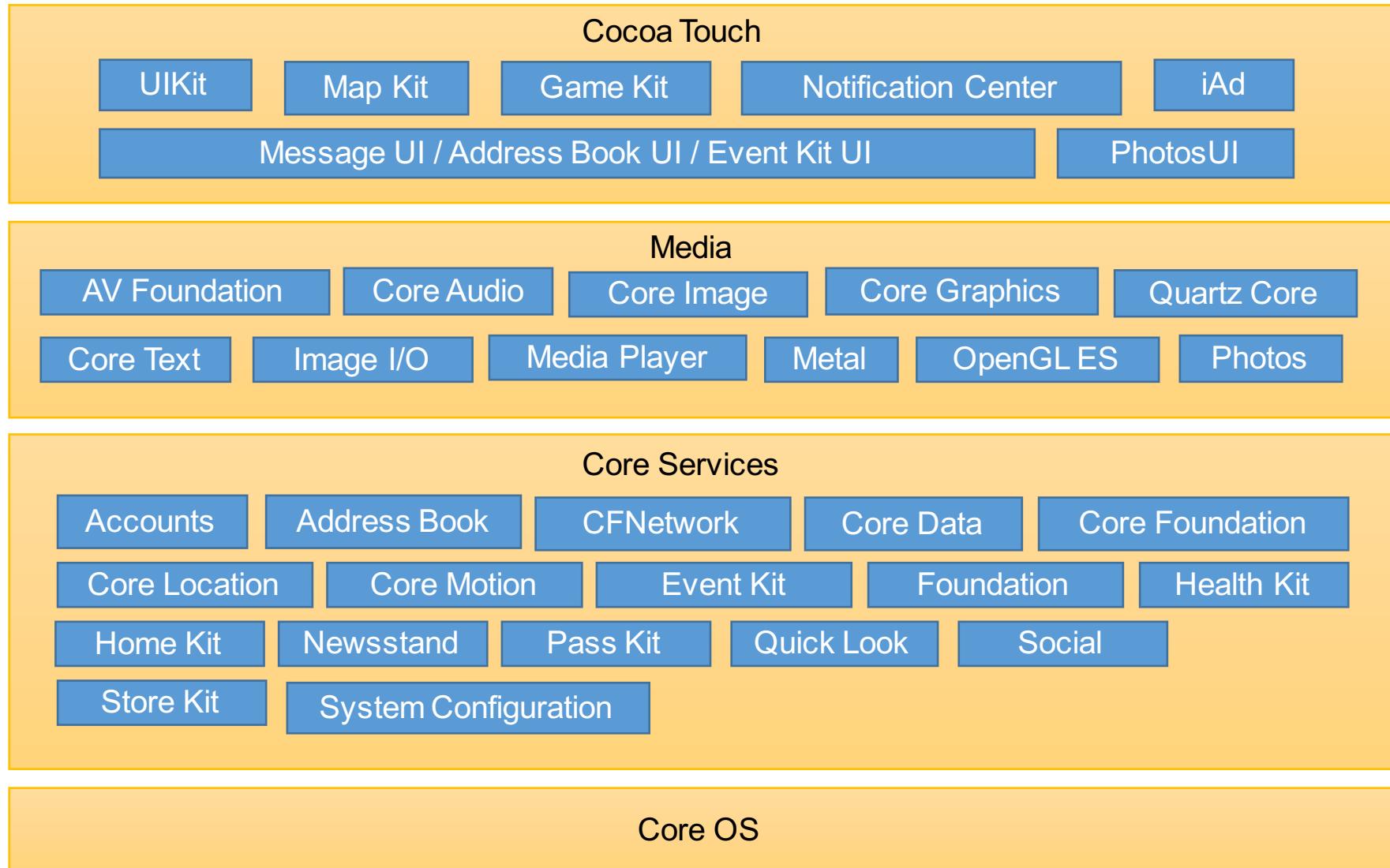
What is iOS?

- Previously—iPhone OS
- Unix-based operating system.
Subset of **Mac OS X** (based on NeXTSTEP Unix OS, 1989~1997).
- First smartphone OS with **multi-touch** graphical user interface
- Latest version: iOS 9
- iOS Devices: iPod, iPod Touch, iPhone, iPad etc.
- Highly integrated (hardware + software)
- Security reason; Applications run individually, cannot interact with each other easily (iOS 7-)

iOS Application Compiler Architecture



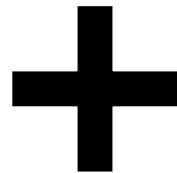
iOS Software Stack



iOS Versions and Compatibility

iOS ver.	Devices							
	Media Player			Smart Phone			Tablet	
	iPod Touch 3	iPod Touch 4	iPod Touch 5~	iPhone 3GS	iPhone 4	iPhone 4S~	iPad	iPad 2~
~ 5.1.1	✓	✓		✓	✓	✓	✓	✓
~ 6.1.6		✓	✓	✓	✓	✓		✓
~ 7.1.2			✓		✓	✓		✓
~ 8.4.1			✓			✓		✓
9.0 ~			✓			✓		✓

Tools Required for iOS App Development



Xcode IDE

Mac OS compiler

Developer Program

Apple Developer Program

Overview What's Included How it Works **Enroll**

How the Program Works



Getting Started

If you're new to development on Apple Platforms, you can get started with our [tools and resources for free](#). If you're ready to build more advanced capabilities and distribute your apps on the App Store, enroll in the Apple Developer Program. The cost is **99 USD** per membership year.

If you want to distribute your app to [App Store](#)

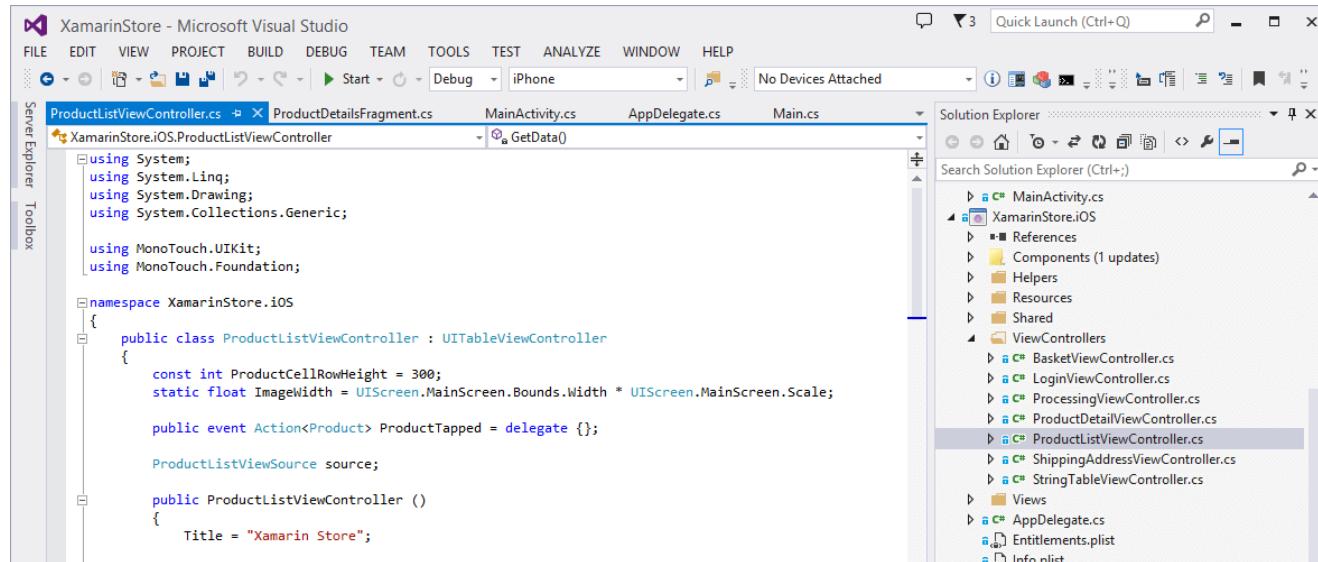
[Get started with enrollment >](#)

Development Types

	Apple ID	Individual	Organisation	Enterprise Program
Xcode Developer Tools	◎	◎	◎	◎
Xcode Beta	◎	◎	◎	◎
Test on Device	◎	◎	◎	◎
App Store Distribution		◎	◎	
In-house App Distribution				◎
Team Management			◎	◎
Cost	Free	99 USD	99 USD	299 USD
Requirement	13+	18+	DUNS Number	DUNS Number

Alternative Development Environment (1)

- <http://xamarin.com/platform>
- Previous Mono Touch
- C#
- Write once, deploy on Android, iOS, Windows Phone
- **Still requires a Mac OS computer/compiler**



The screenshot shows the Microsoft Visual Studio interface with the title bar "XamarinStore - Microsoft Visual Studio". The menu bar includes FILE, EDIT, VIEW, PROJECT, BUILD, DEBUG, TEAM, TOOLS, TEST, ANALYZE, WINDOW, and HELP. The toolbar has icons for file operations like Open, Save, and Build. The status bar at the bottom says "Quick Launch (Ctrl+Q)" and "No Devices Attached". The main window displays several tabs: ProductListViewController.cs, ProductDetailsFragment.cs, MainActivity.cs, AppDelegate.cs, and Main.cs. The ProductListViewController.cs tab is active, showing C# code for a Xamarin iOS application. The code includes namespaces System, System.Linq, System.Drawing, System.Collections.Generic, and MonoTouch.UIKit, along with a class definition for ProductListViewController that extends UITableViewController. The Solution Explorer on the right lists the project structure: MainActivity.cs, XamarinStore.iOS (which contains References, Components (1 update), Helpers, Resources, Shared, and ViewControllers folders containing BasketViewController.cs, LoginViewController.cs, ProcessingViewController.cs, ProductDetailViewController.cs, ProductListViewController.cs, ShippingAddressViewController.cs, StringTableViewController.cs, and Views folder), AppDelegate.cs, Entitlements.plist, and Info.plist.

```
using System;
using System.Linq;
using System.Drawing;
using System.Collections.Generic;

using MonoTouch.UIKit;
using MonoTouch.Foundation;

namespace XamarinStore.iOS
{
    public class ProductListViewController : UITableViewController
    {
        const int ProductCellRowHeight = 300;
        static float ImageWidth = UIScreen.MainScreen.Bounds.Width * UIScreen.MainScreen.Scale;

        public event Action<Product> ProductTapped = delegate {};
        ProductListViewSource source;

        public ProductListViewController ()
        {
            Title = "Xamarin Store";
        }

        protected override void AwakeFromNib()
        {
            base.AwakeFromNib();
            source = new ProductListViewSource();
            source.Source = GetProducts();
            source.Delegate = this;
            UITableView:UITableViewSource = source;
        }

        public override void DidReceiveMemoryWarning()
        {
            source.Dispose();
            source = null;
        }

        public override void ViewDidLoad()
        {
            base.ViewDidLoad();
            source.ReloadData();
        }

        public override void ViewWillAppear(bool animated)
        {
            base.ViewWillAppear(animated);
            source.ReloadData();
        }

        public override void ViewDidAppear(bool animated)
        {
            base.ViewDidAppear(animated);
            source.ReloadData();
        }

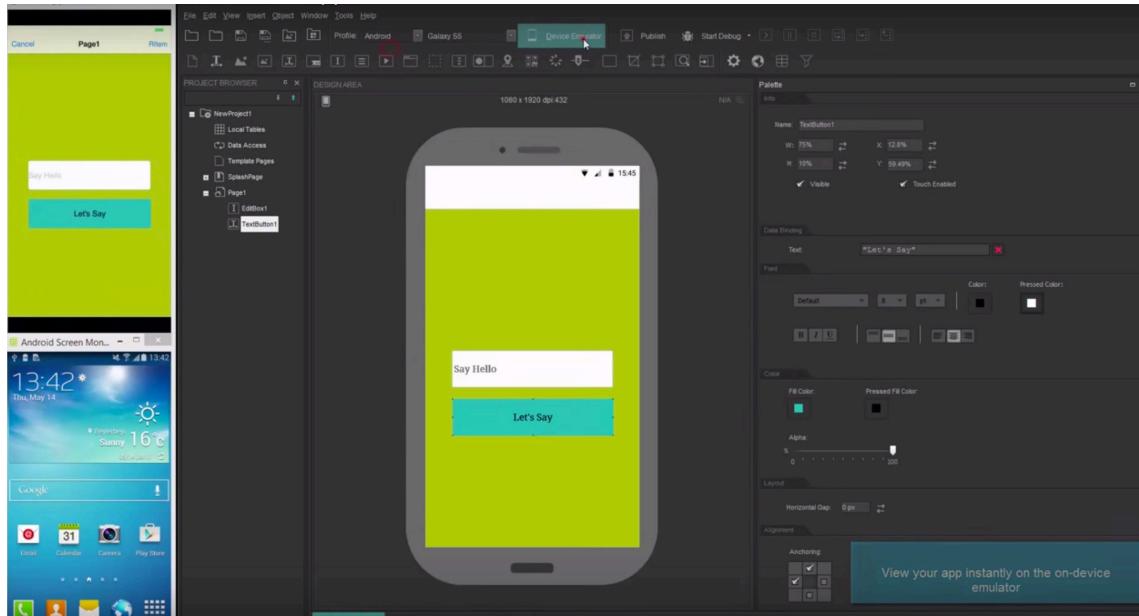
        public override void ViewWillDisappear(bool animated)
        {
            base.ViewWillDisappear(animated);
            source.ReloadData();
        }

        public override void ViewDidDisappear(bool animated)
        {
            base.ViewDidDisappear(animated);
            source.ReloadData();
        }

        public override void DidSelectRowAtIndexPath(NSIndexPath indexPath)
        {
            var product = source[indexPath];
            ProductTapped?.Invoke(product);
        }
    }
}
```

Alternative Development Environment (2)

- <http://www.smartface.io/>
- JavaScript
- Write once, deploy on Android, iOS,
- **Still requires a Mac OS computer for App Distribution**



Source: <http://www.smartface.io/developer/guides/get-started/hello-world/>

Alternative Development Environment (3)

- Cloud Service
 - Example: <https://virtualmacosx.com/>

*Own a Shared Mac Server

why rent when you can own



- Equity
- Capital
- Asset Control
- Power
- Compliance

\$14.75
starting from

- Apple Branded Mac Servers
- Tier 3 Datacenters
- Xeon Processors / ECC DDR3 RAM
- Shared Resources 4GB/2CPU
- Install your own Software (BYOL)
- Complete the *Pay as you Go* term and pay only hosting fees!
- Cancel Anytime

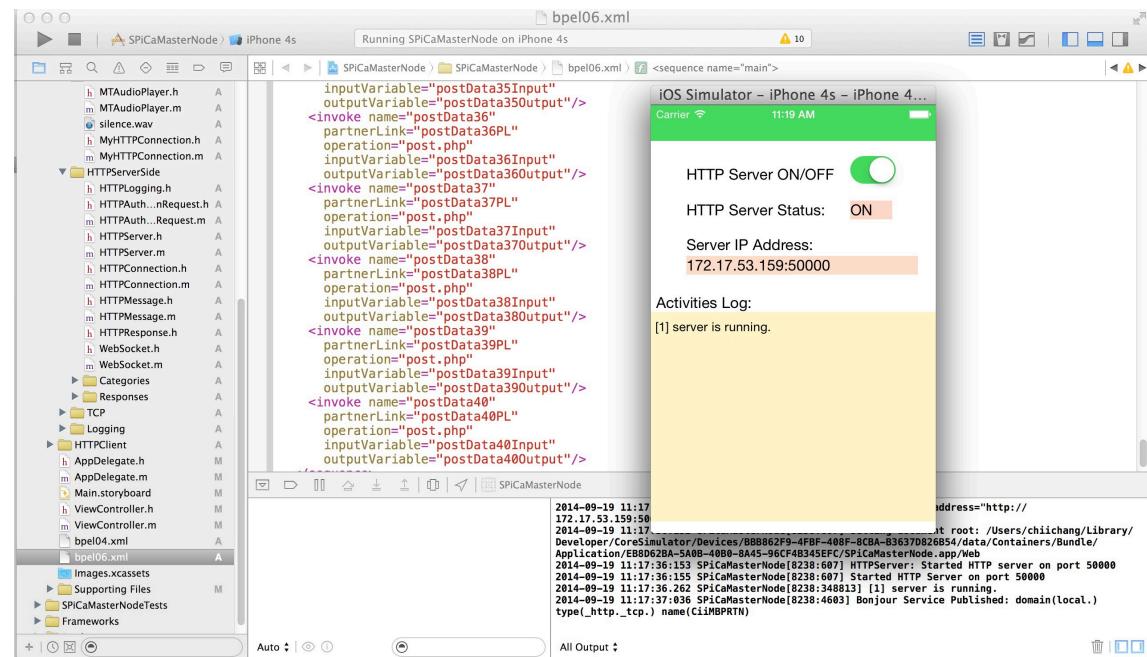
*Fractional ownership starting @ just \$4.75 per month + \$10.00 hosting fees

iOS Device Simulator

- Simulator ≠ Emulator
- Simulator:
 - Share hardware resources
 - Subset of current OS
 - Fast
- Emulator:
 - Virtual machine
 - Different OS
 - Slow

Demo

Hello World



iPhone 4S Simulator

Development Environment: Xcode

Files

The screenshot shows the Xcode interface with the following components:

- Project Navigator:** Shows the project structure for "HelloWorldSwift".
- Code Editor:** Displays the "ViewController.swift" file content:

```
// ViewController.swift
// HelloWorldSwift
//
// Created by chii chang on 22/08/2015.
// Copyright (c) 2015 chii chang. All rights reserved.

import UIKit

class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from a nib.
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }
}
```

- UI Setting:** A sidebar titled "UI Setting" lists various UI components with their descriptions.
- Console Output:** A panel at the bottom right shows "All Output" with the message "HelloWorldSwift: Ready | Today at 16:11".

Coding

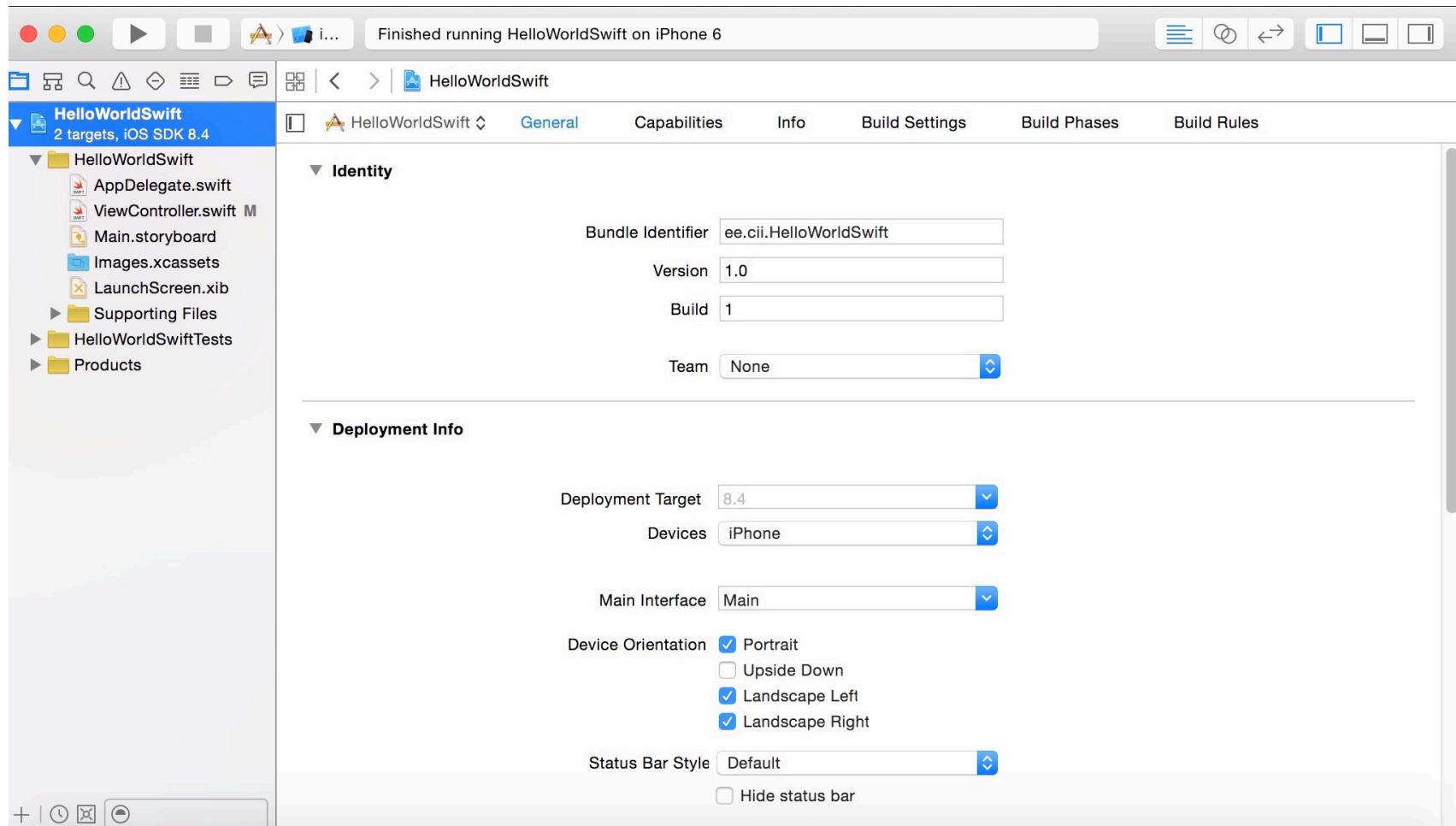
Thread tracking
(e.g. for debug)

Console output

UI Components

UI Setting

Application Project

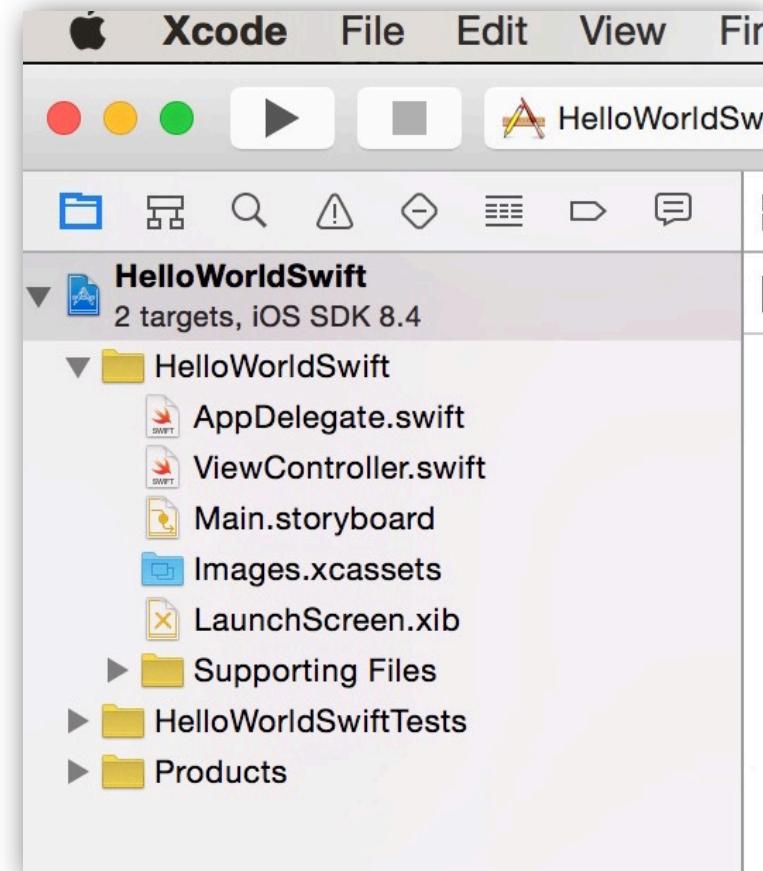
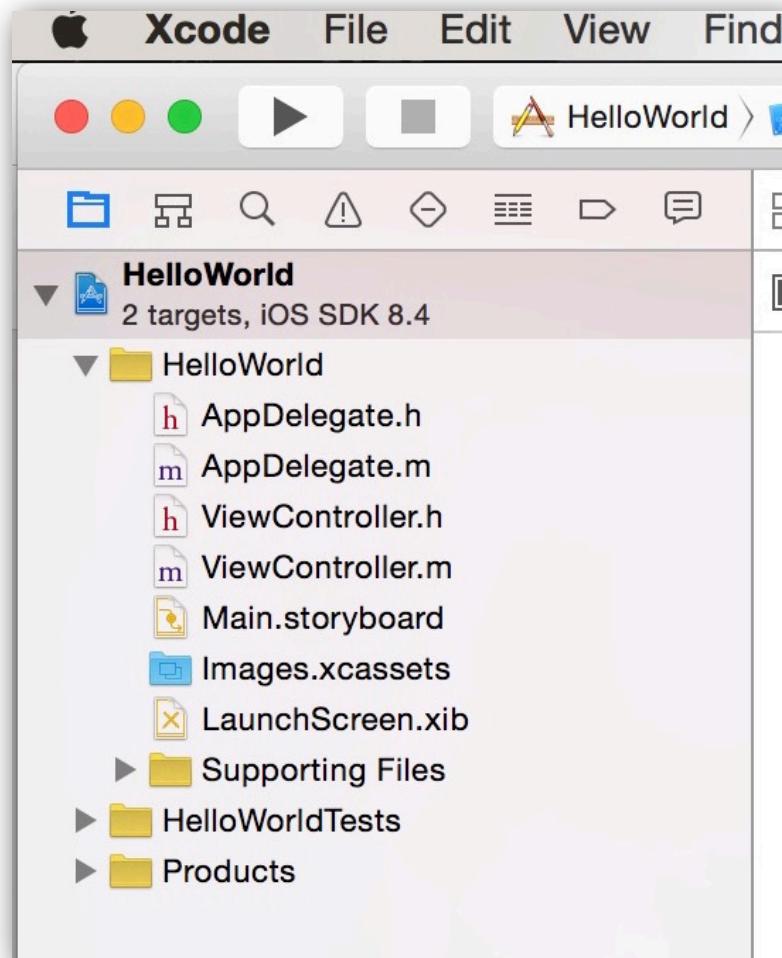


Programming Language

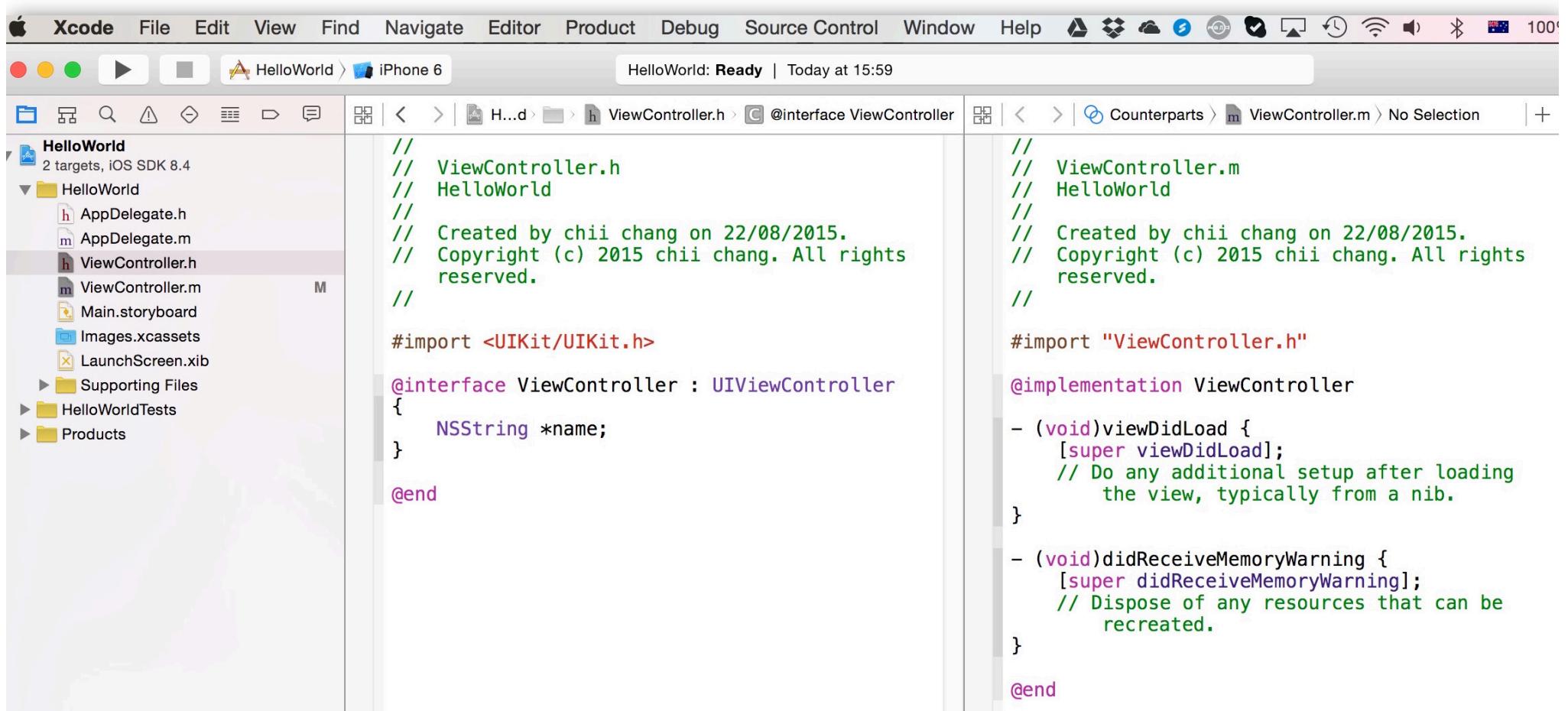
- Objective-C
- Swift
- C (limited usage)
- C++ (uncommon)

File Structure of iOS App (1/2)

- Objective-C vs. Swift



Objective-C Class



The screenshot shows the Xcode interface with two files open: ViewController.h and ViewController.m. The left panel shows the project structure for 'HelloWorld' with files like AppDelegate.h, AppDelegate.m, ViewController.h, ViewController.m, Main.storyboard, Images.xcassets, LaunchScreen.xib, Supporting Files, HelloWorldTests, and Products. The top bar shows 'HelloWorld: Ready | Today at 15:59'. The ViewController.h file contains the class definition and imports. The ViewController.m file contains the implementation of the methods.

```
// ViewController.h
// HelloWorld
//
// Created by chii chang on 22/08/2015.
// Copyright (c) 2015 chii chang. All rights reserved.

#import <UIKit/UIKit.h>

@interface ViewController : UIViewController
{
    NSString *name;
}

@end
```

```
// ViewController.m
// HelloWorld
//
// Created by chii chang on 22/08/2015.
// Copyright (c) 2015 chii chang. All rights reserved.

#import "ViewController.h"

@implementation ViewController

- (void)viewDidLoad {
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.
}

- (void)didReceiveMemoryWarning {
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.
}

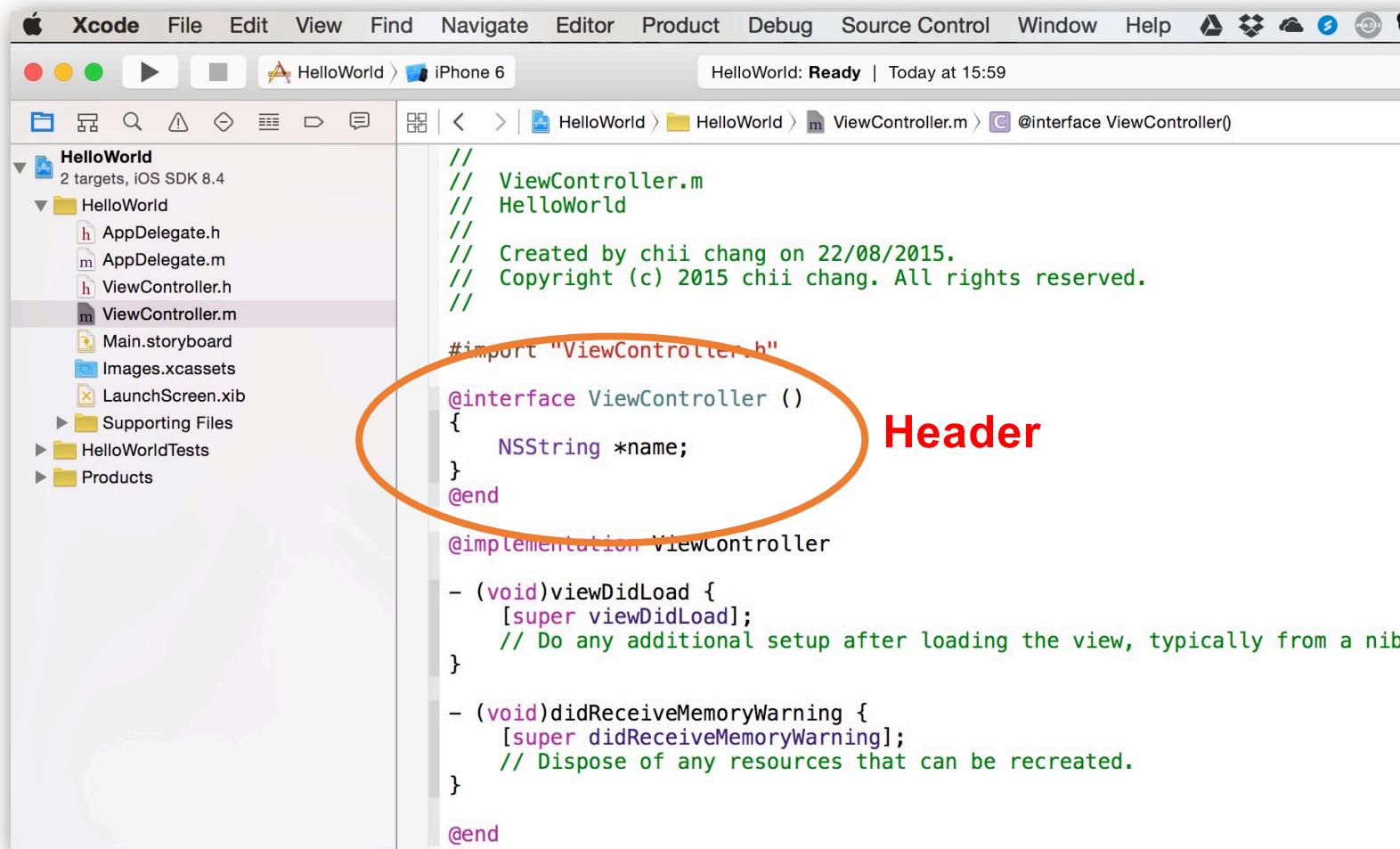
@end
```

Header

Method

File Structure of iOS App (2/2)

- Header description can be included in the method file (in Objective-C)



The screenshot shows the Xcode interface with the "ViewController.m" file open. The left sidebar displays the project structure under the "HelloWorld" target. The main editor area shows the following code:

```
// ViewController.m
// HelloWorld
//
// Created by chii chang on 22/08/2015.
// Copyright (c) 2015 chii chang. All rights reserved.

#import "ViewController.h"

@interface ViewController : UIViewController
{
    NSString *name;
}
@end

@implementation ViewController

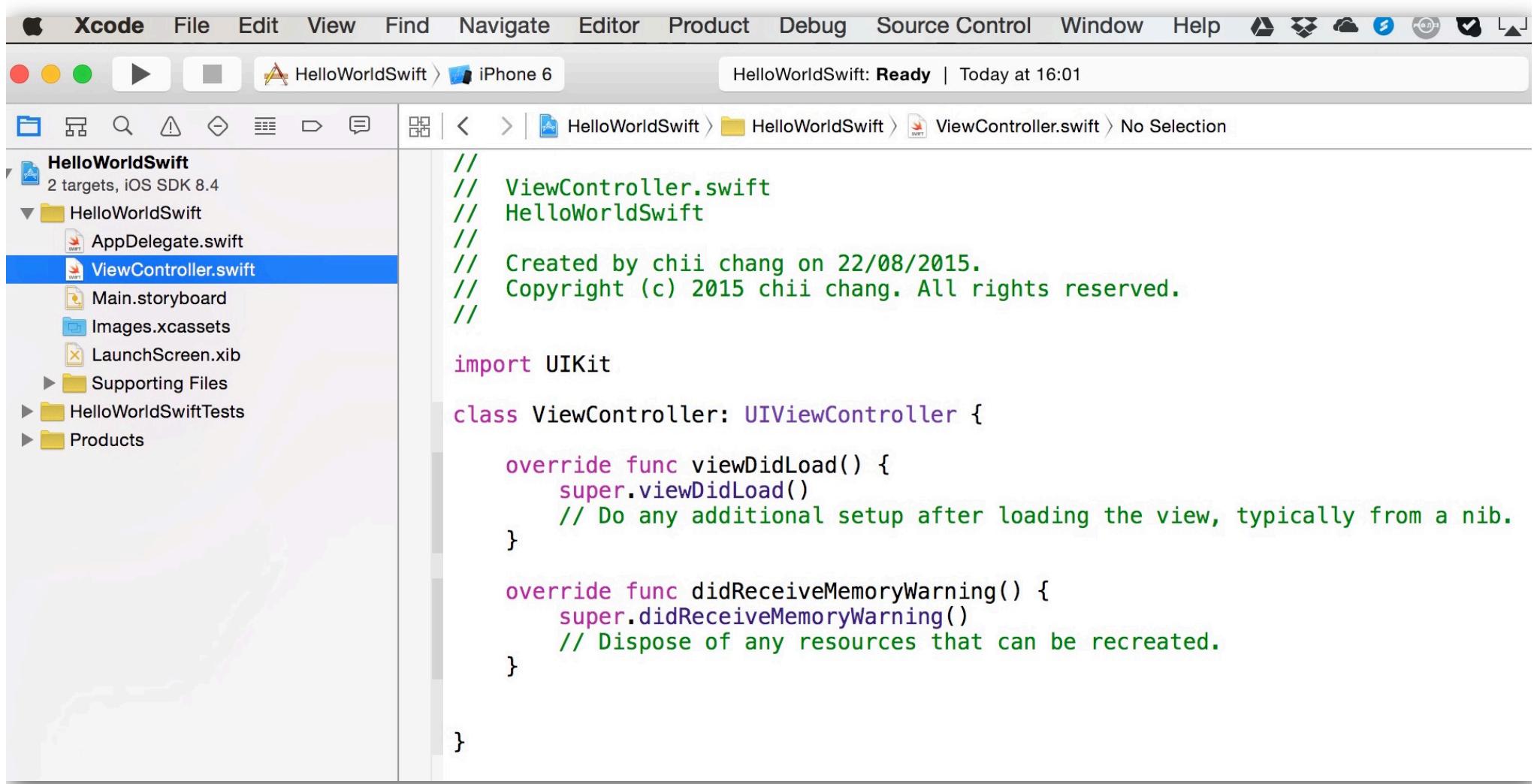
- (void)viewDidLoad {
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.
}

- (void)didReceiveMemoryWarning {
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.
}

@end
```

A red oval highlights the first few lines of the code, which contain the header description and imports. To the right of the oval, the word "Header" is written in red.

Swift Class



The screenshot shows the Xcode interface with the following details:

- Menu Bar:** Xcode, File, Edit, View, Find, Navigate, Editor, Product, Debug, Source Control, Window, Help.
- Toolbar:** Standard Xcode toolbar icons.
- Project Navigator:** Shows the project structure for "HelloWorldSwift".
 - Root folder: HelloWorldSwift (2 targets, iOS SDK 8.4)
 - Source files:
 - AppDelegate.swift
 - ViewController.swift (selected)
 - Main.storyboard
 - Images.xcassets
 - LaunchScreen.xib
 - Supporting Files
 - Test Targets: HelloWorldSwiftTests
 - Products
- File Navigator:** Shows the current file path: HelloWorldSwift > HelloWorldSwift > ViewController.swift.
- Editor:** Displays the code for ViewController.swift.

```
// ViewController.swift
// HelloWorldSwift
//
// Created by chii chang on 22/08/2015.
// Copyright (c) 2015 chii chang. All rights reserved.

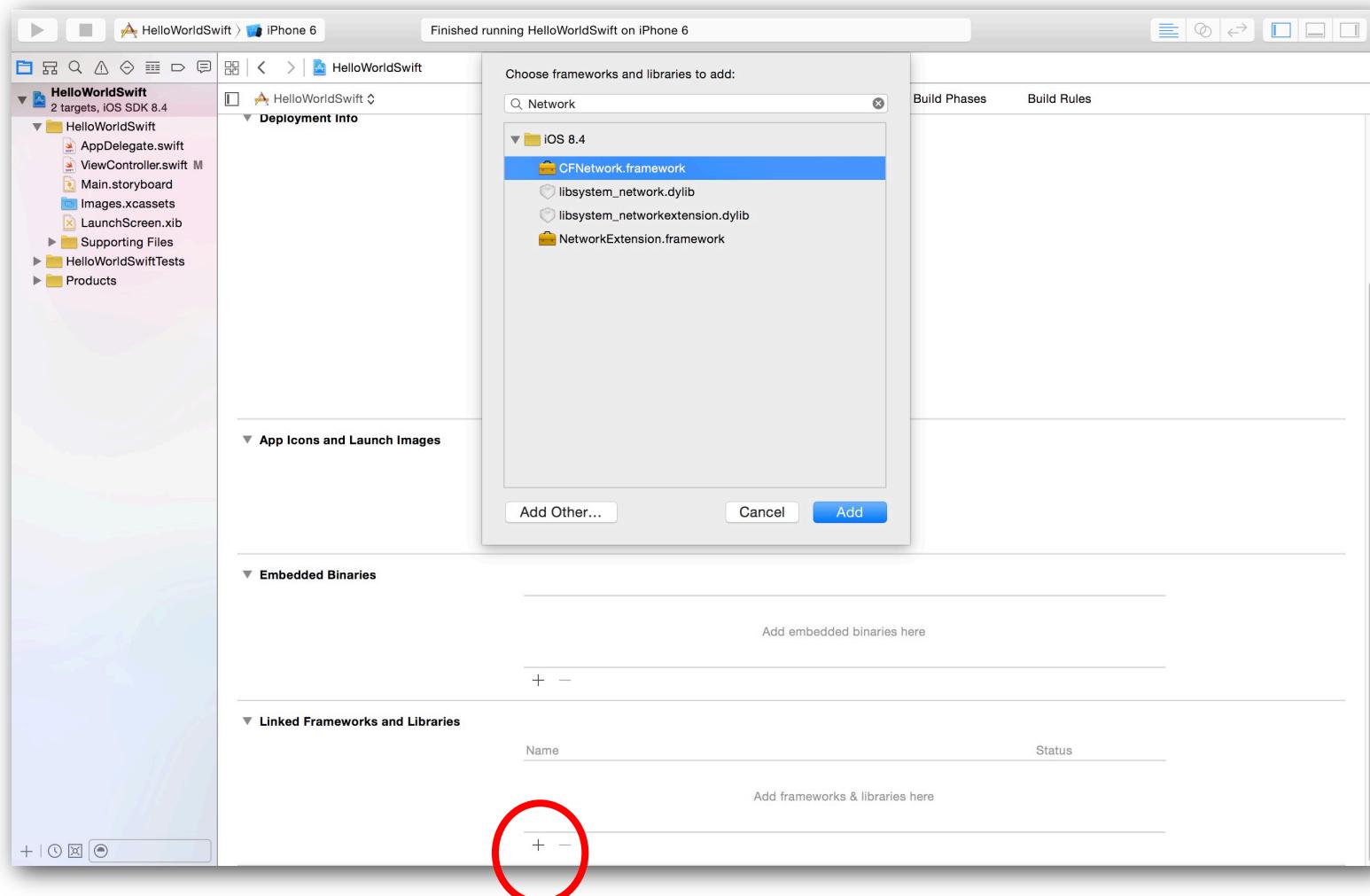
import UIKit

class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from a nib.
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }
}
```
- Status Bar:** HelloWorldSwift: Ready | Today at 16:01

Application Settings - Linked Framework and Libraries

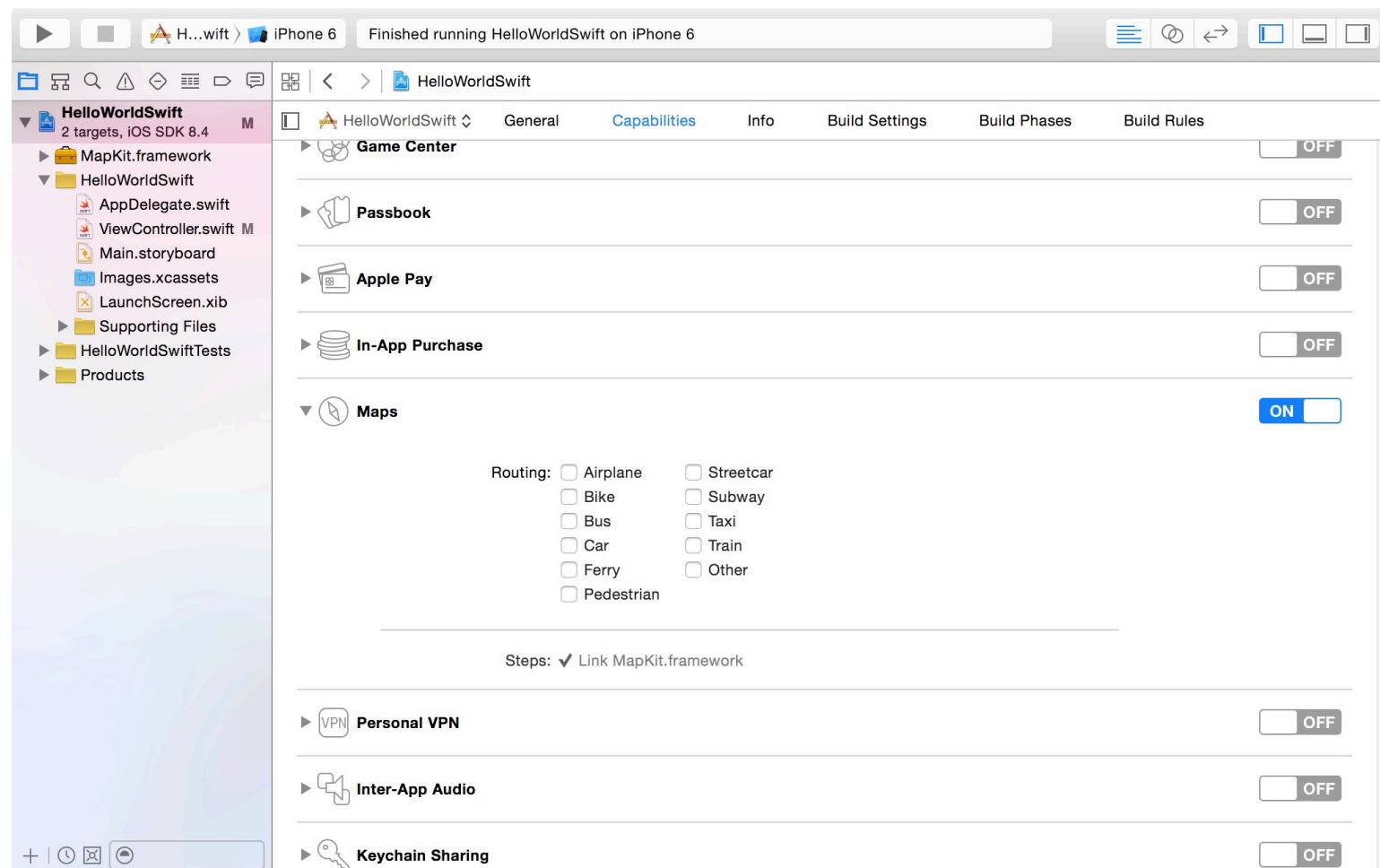


Importing 3rd Party Resources?

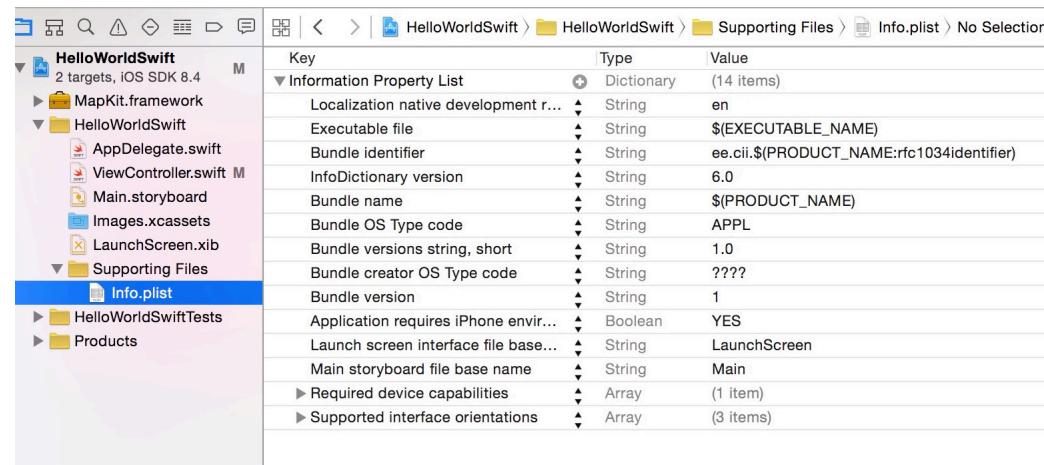
Demo

Bridging Objective-C class with Swift class

Application Settings - Media Kit



Application Settings - Property List (plist) and Permission



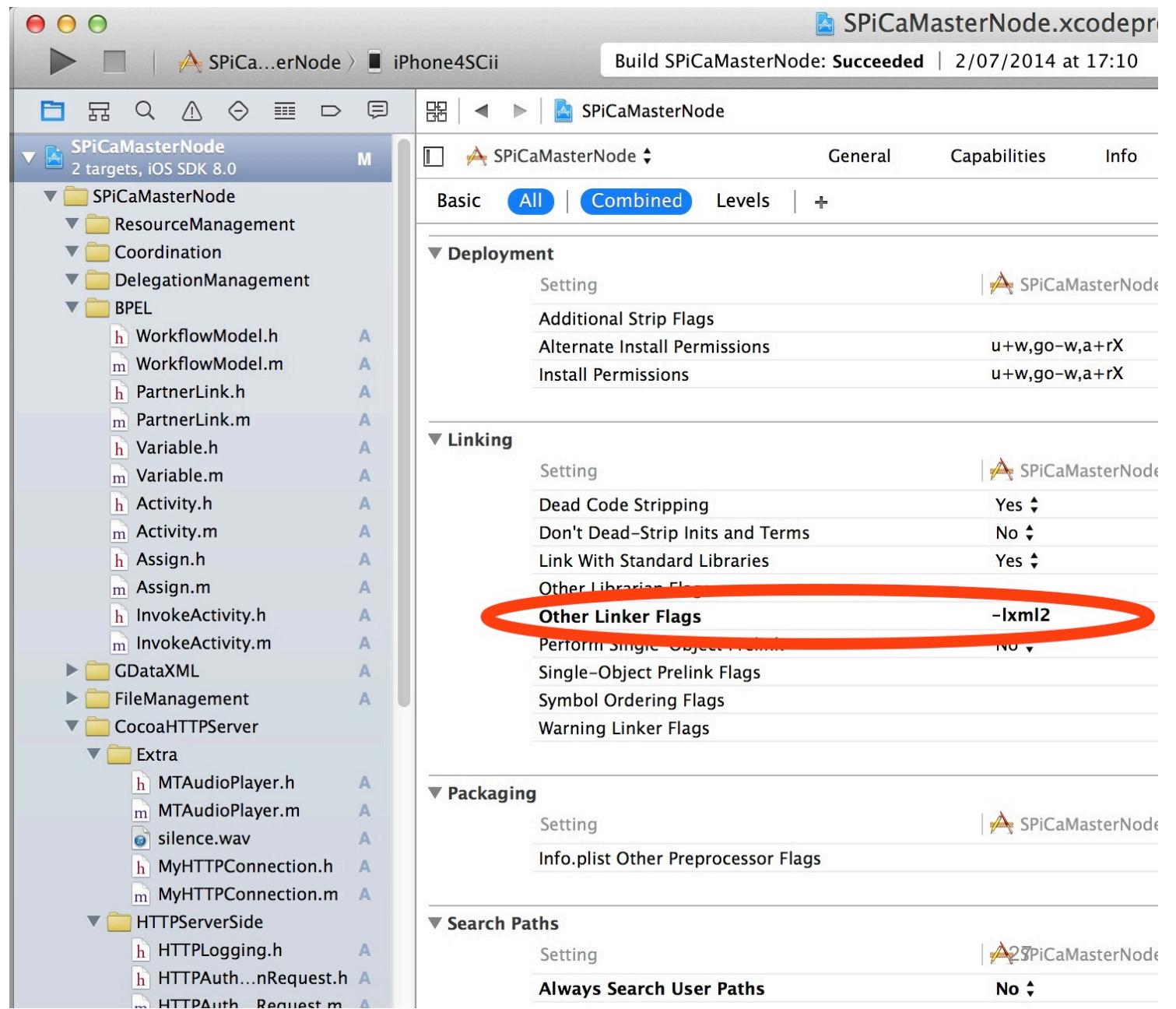
Example: Location Tracking Permission

Application requires location en...	Type	Value
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)

Source: <http://willd.me/posts/getting-started-with-ibeacon-a-swift-tutorial>

Application Settings - Flag

- Example, using GDataXML (of Gdata API)



Application Settings – Search Path

The screenshot shows the Xcode interface with the project 'SPiCaMasterNode' selected. The 'Build Settings' tab is active. In the 'Search Paths' section, the 'Header Search Paths' setting is expanded, showing several paths listed. The path '/usr/include/libxml2' is highlighted with a red circle.

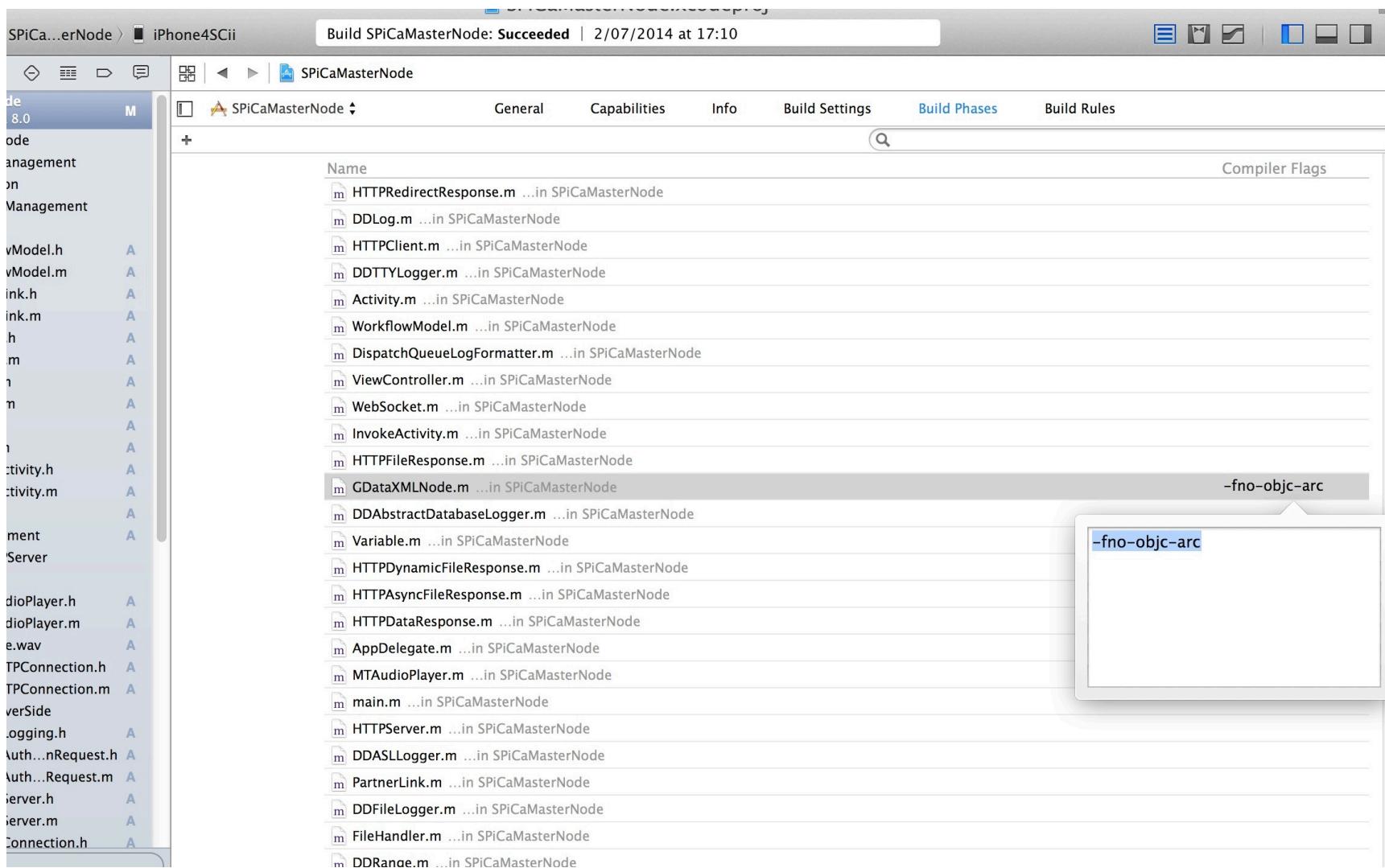
Setting	Value	Options
Always Search User Paths	No	▼
Header Search Paths	/usr/include/libxml2 /Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/include	non-recursive ▾
Library Search Paths		non-recursive ▾
Rez Search Paths		non-recursive ▾
Sub-Directories to Exclude in Recursive Searches		
Sub-Directories to Include in Recursive Searches		
User Header Search Paths		

Header Search Paths:

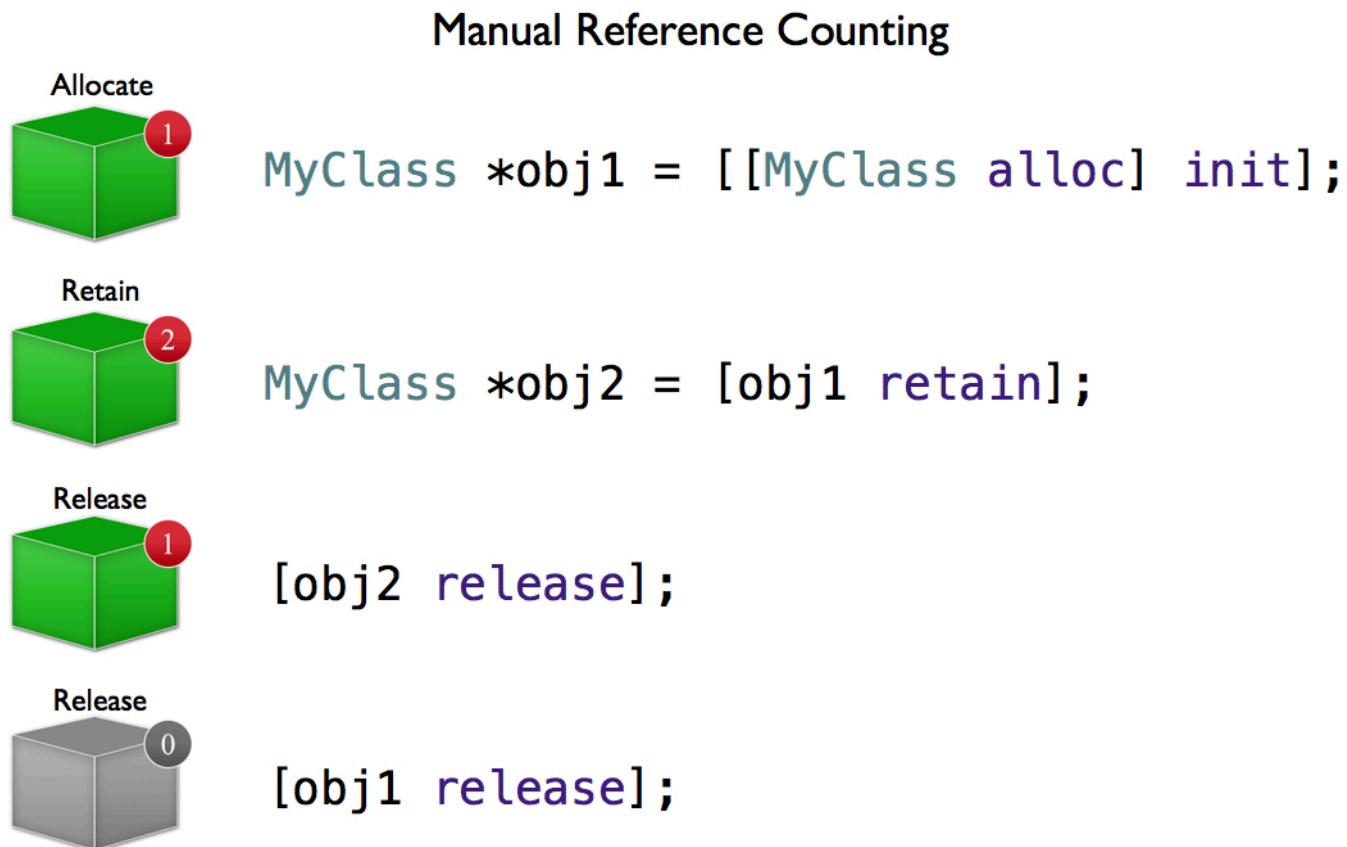
- /usr/include/libxml2
- /Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/include

Application Settings – Individual Class Setting (disable ARC)

Disable “Automatic Reference Counting (ARC)”



Automatic Reference Counting (ARC)



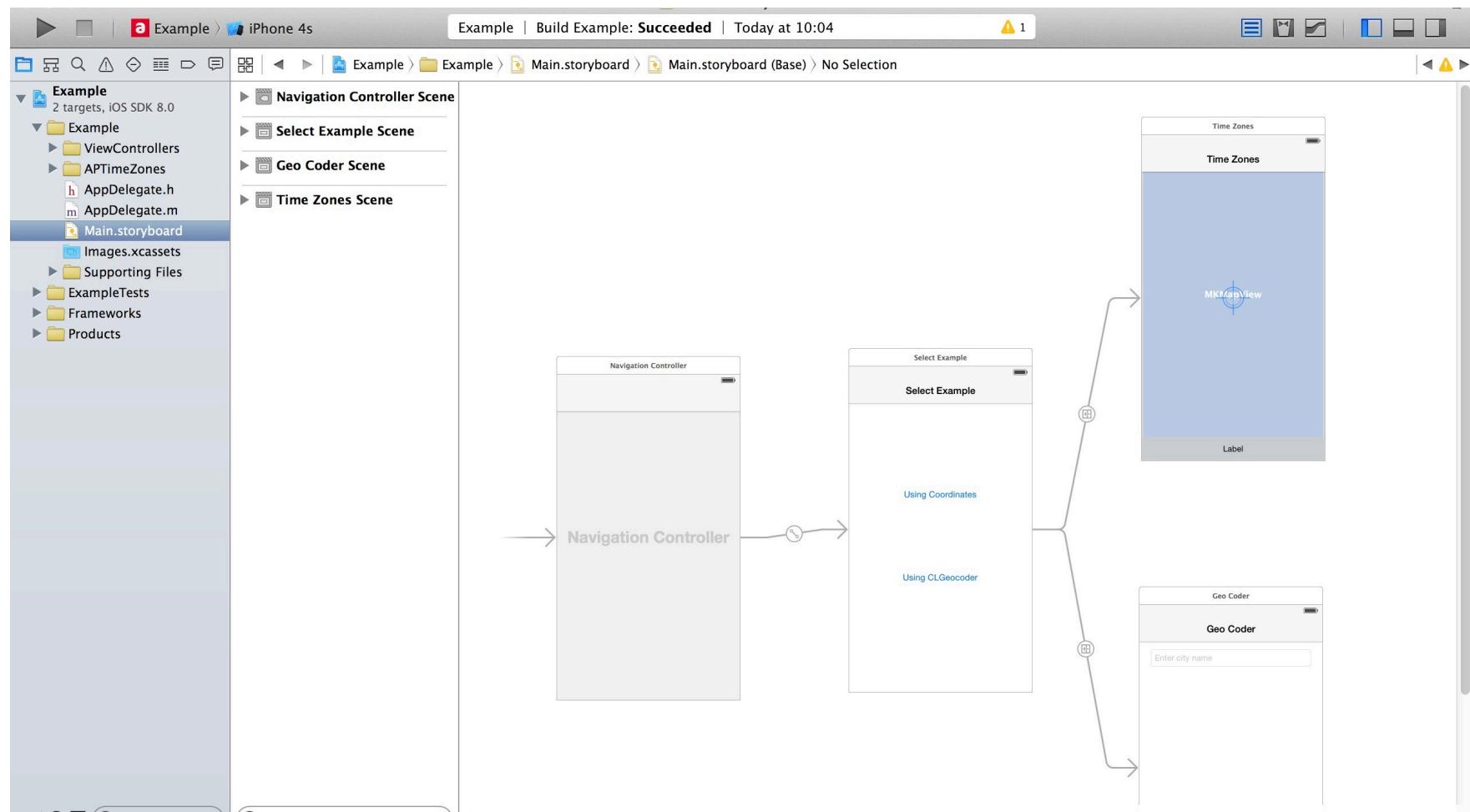
<http://blog.teamtreehouse.com/ios-5-automatic-reference-counting-arc>

Debug

The screenshot shows the Xcode interface with the following details:

- Project Navigator:** Shows "SpriteKit" and "iPhone 4s".
- Editor:** Displays the code for `SKFoePlane.m`. A green highlight covers the `createMediumPlane` method. A blue arrow points to the first line of this method, indicating it is the current target. A yellow arrow points to the second line, indicating it is the next instruction to be executed.
- Call Stack:** Shows the call stack for Thread 1, starting with `0 +[SKFoePlane createMediumPlane]`.
- Breakpoint:** A yellow arrow points to the line `foePlane.hp = 5;` in the `createMediumPlane` method, which is highlighted in green. A tooltip indicates "Thread 1: breakpoint 1.1".
- Variables:** The variable `self` is set to `(Class) SKFoePlane`, and `foePlane` is set to `(SKFoePlane *) 0x7eb7e9b0`.
- lldb:** The command `0 +[SKFoePlane createMediumPlane]` is entered in the lldb console.
- Right Sidebar:** Shows three icons: View Controller, Navigation Controller, and Table View Controller, each with a brief description.

Storyboard

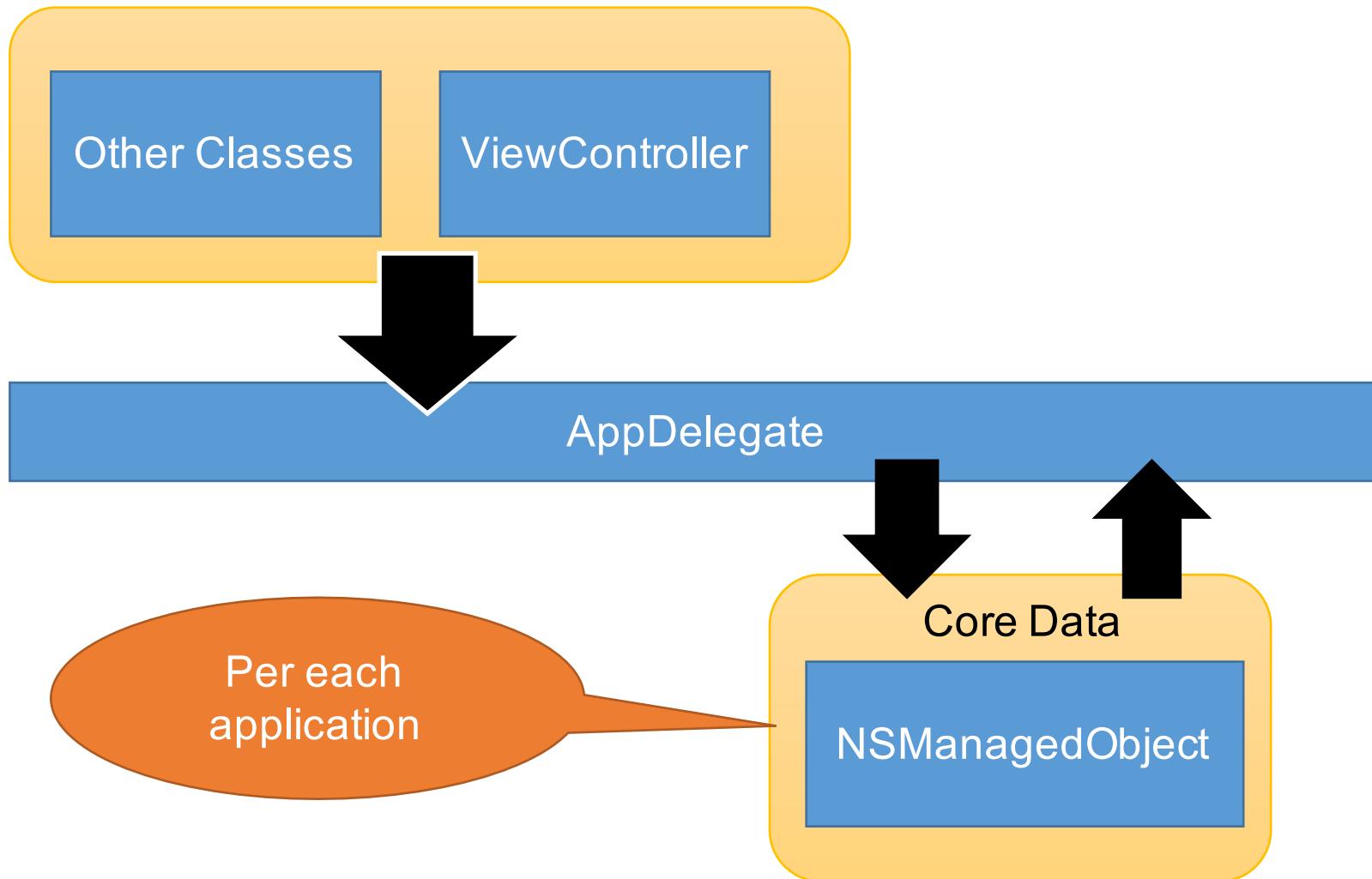


Basic Graphical User Interface

- Layout
- Linking GUI objects to classes (drag and drop) in Objective-C

DEMO

Inbuilt Database: Core Data



Detail in later slides

Accessing File System?

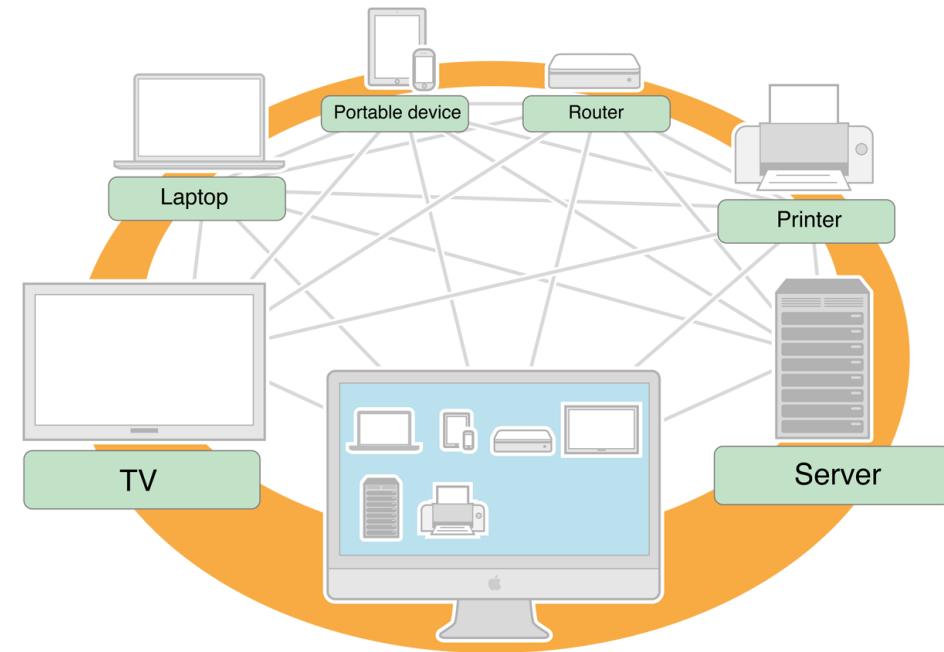
- By default, an application can only access its corresponding directories
- Example in Objective-C:

```
+ (void)saveFile: (NSData *)aData withFileName: (NSString *)fileName andFileType: (NSString *)fileType
{
    NSString *docDir =
[NSSearchPathForDirectoriesInDomains (NSDocumentDirectory,
NSUserDomainMask, YES) objectAtIndex:0];
    NSString *ffPath = [NSString
stringWithFormat:@"%@/%@.%@", docDir, fileName, fileType];
    [aData writeToFile:ffPath atomically:YES];
}
```

Zero-configuration: Bonjour

- Multicast DNS (mDNS)
- Included in foundation library: **NSNetService; CFNetServices**

“Bonjour is a suite of protocols for zero-configuration networking over IP that Apple has submitted to the IETF as part of the ongoing standards-creation process.”
—(Mac Developer Library)

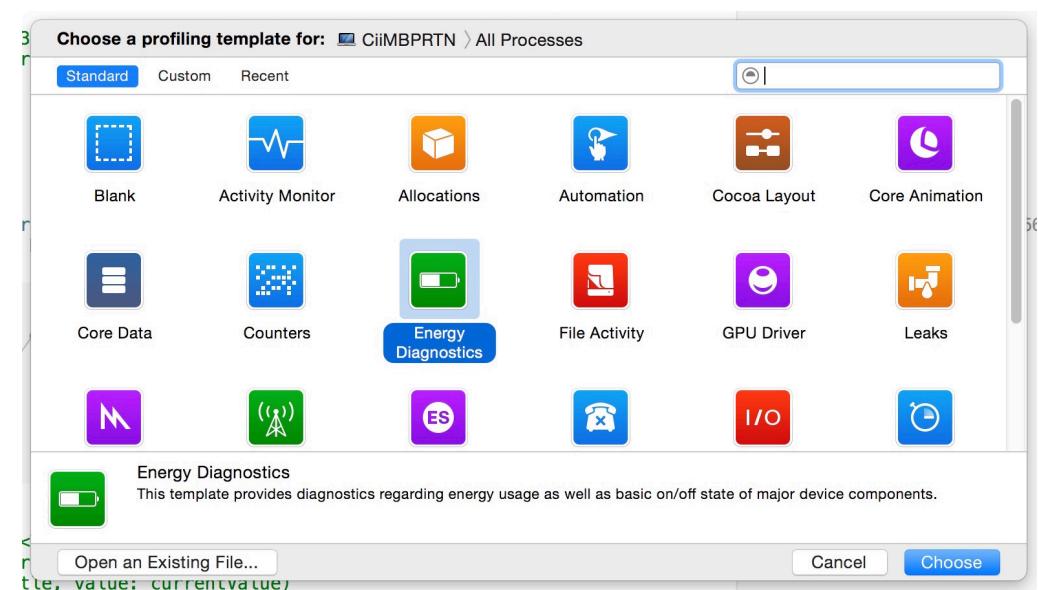
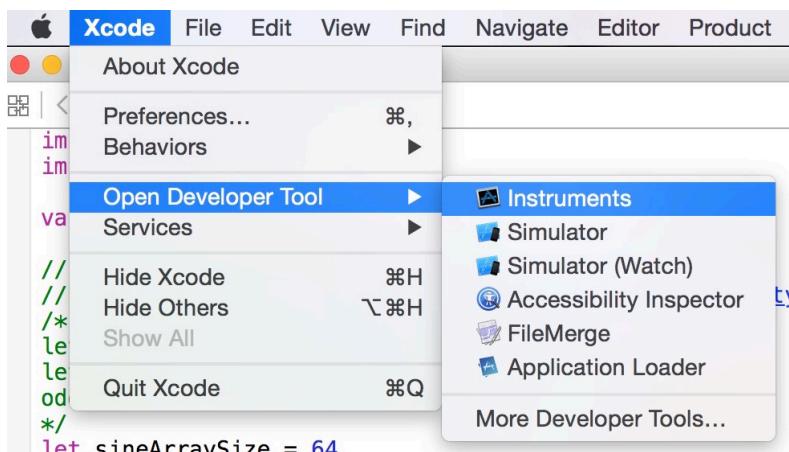


Source:

<https://developer.apple.com/library/mac/documentation/Cocoa/Conceptual/NetServices/Introduction.html>

Instruments

- Demo

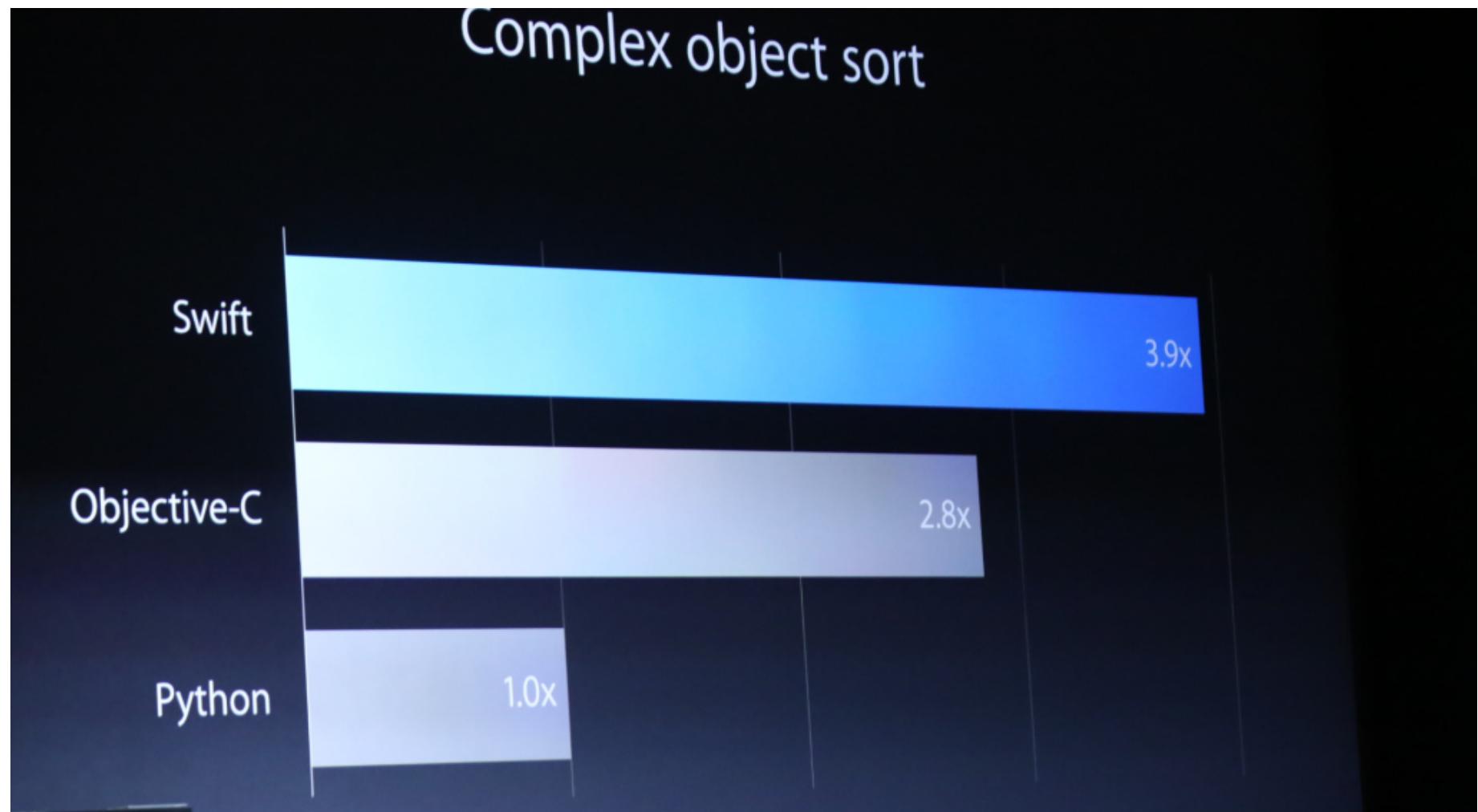


Programming Languages for iOS App Development

- Objective-C (ObjC)
 - Programming language for NeXTSTEP OS (1980s)
 - from NeXT inc. founded by Steve Jobs in 1985 – 1996 (Acquired by Apple Computer)
- **Swift**
 - Ver. 1 introduced in 2014;
 - Now: **Swift 2, Open Source** in fall 2015, introduced in WWDC 2015¹
- C++
 - See <http://www.raywenderlich.com/62989/introduction-c-ios-developers-part-1>

¹ <https://developer.apple.com/swift/blog/?id=29>

Swift has better performance



Demo: Playground

Swift: Basic Syntax

- **NS***; e.g. **NSURL**, **NSURLRequest**, **NSURLSession** or **NSURLConnection** etc.
- ‘NS’ stands for ‘NeXTSTEP’;
- No Semicolon required
- Type safe

Example of ‘Type Safe’:

```
let url = NSURL(string:  
"http://www.stackoverflow.com" )
```

```
var v1 = 123
```

Java, Objective-C, Swift (without importing extra libraries)

- Java

```
java.lang.String gender = new String("Male");
```

or

```
java.lang.String gender = "Male";
```

- Objective-C

```
NSString *gender = [[NSString alloc] initWithString:@"Male"];
```

or

```
NSString *gender = @"Male";
```

- Swift

```
let gender = "Male" // Constant; Cannot be changed
```

or

```
var gender = "Male" // Can be changed
```

or

```
var gender: String? = "Male"
```

Demo

Swift: Basic Syntax

- ‘Let’ (constant value) and ‘Var’ (variable)

```
let person = "Peter"  
var man = "Peter"  
man = "Ken" // this is ok  
person = "Ken" // this is an error
```

Let π = 3.14159265359

```
let 🐱 = "cat"  
  
print(🐱) //display "cat"
```

Swift: Basic Syntax

- Array

```
var a2 = ["cat", "mouse", "dog"]
print(a2[1]) //display "mouse"
```

- Dictionary (Hashmap)

```
var animalList = ["🐱": "cat", "🐶": "dog", "🐭": "mouse"]

print(animalList["🐶"]!) //display "dog"
```

Swift: Basic Syntax

- Dictionary (Hashmap)

```
var animalList: [Character: String] = [:]
```

```
animalList["🐱"] = "cat"
```

```
animalList["🐶"] = "dog"
```

```
animalList["🐭"] = "mouse"
```

```
print(animalList["🐶"]!) //display "dog"
```

Swift: Basic Syntax

- Loop

```
for i in 0...3
{
    print(i)
}
```

//display

0
1
2
3

Same as

```
for var i = 0; i<=3; ++i {
    print(i)
}
```

```
for j in 0..<3
{
    print(j)
}
```

// display

0
1
2

Swift: Basic Syntax

```
If let actualNumber = Int(possibleNumber) {  
    print("\\" \ (possibleNumber) \\' has an  
integer value of \\ (actualNumber)")  
} else {  
    print("\\" \ (possibleNumber) \\' could not be  
converted to an integer")  
}
```

Source: “The Swift Programming Language (Swift 2)”

Swift: Tuples

```
let http404Error = (404, "Not Found")
```

```
// http404Error is of type (Int, String), and  
// equals (404, "Not Found")
```

From: Apple Inc. “The Swift Programming Language.” iBooks. <https://itun.es/au/jEUH0.l>

Swift: Functions

```
func greet(name: String, day: String) -> String
{
    return "Hello \(name)! Today is \(day)."
}

greet( "Peter" , day: "Monday" )
//display "Hello Peter! Today is Monday."
```

Swift: Protocol

- ‘Protocol’ (in Swift) is similar to ‘Interface’ in Java

```
protocol ExampleProtocol {  
    var simpleDescription: String { get }  
    mutating func adjust()  
}  
  
class SimpleClass: ExampleProtocol {  
    var simpleDescription: String =  
        "A very simple class."  
    var anotherProperty: Int = 69105  
    func adjust() {  
        simpleDescription += " Now 100% adjusted."  
    }  
}
```

More GUI

Demo

Navigation Control, ImageView, TableView,
WebView, MapView

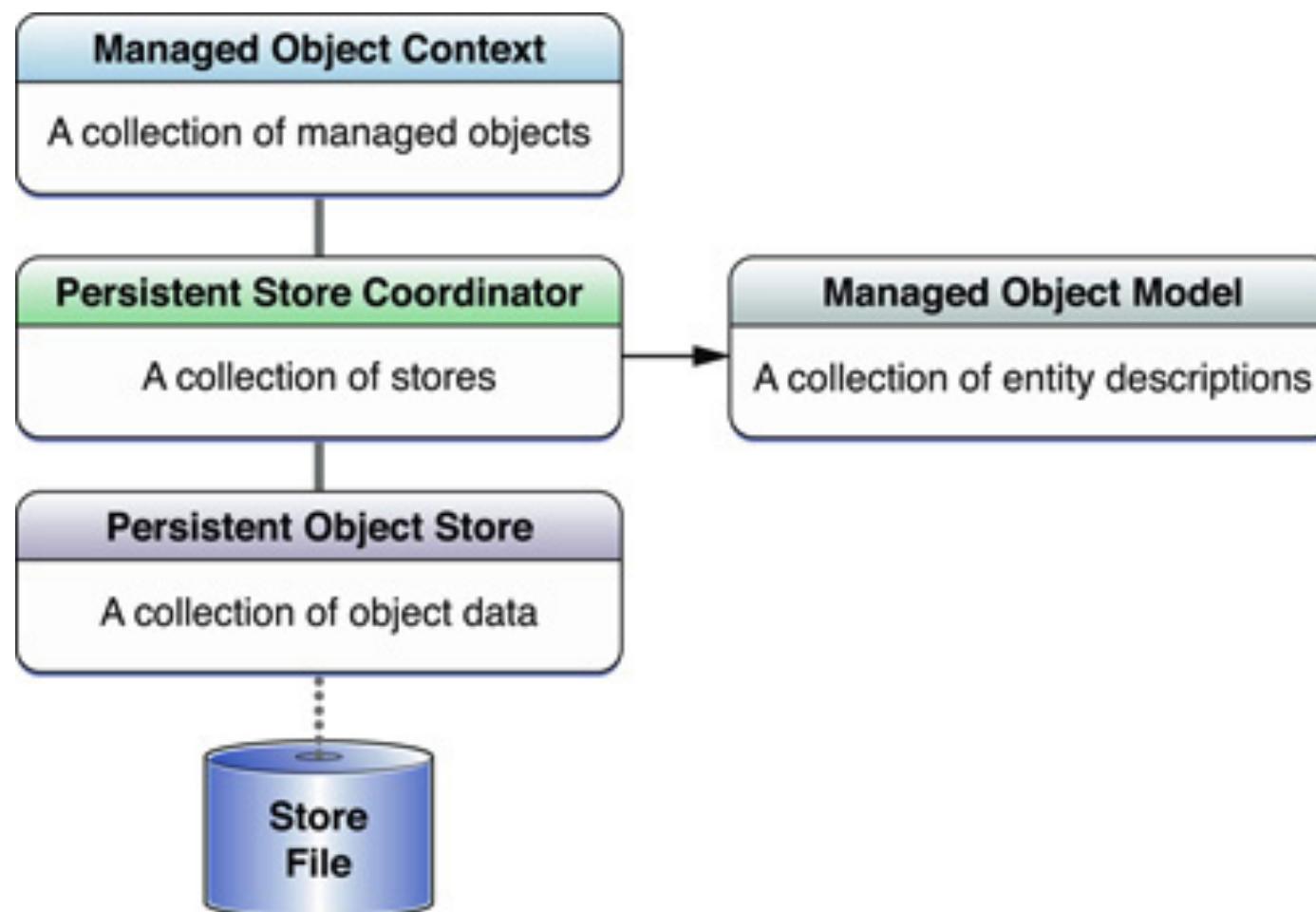
in Swift

[Partial Source code]

Core Data Example: <https://github.com/BNakum/Tutorials>
from <http://agstya.com/core-data-tutorial-in-swift-2-0/>

TableView Example: <http://kodu.ut.ee/~chang/table.zip>

Swift: Core Data



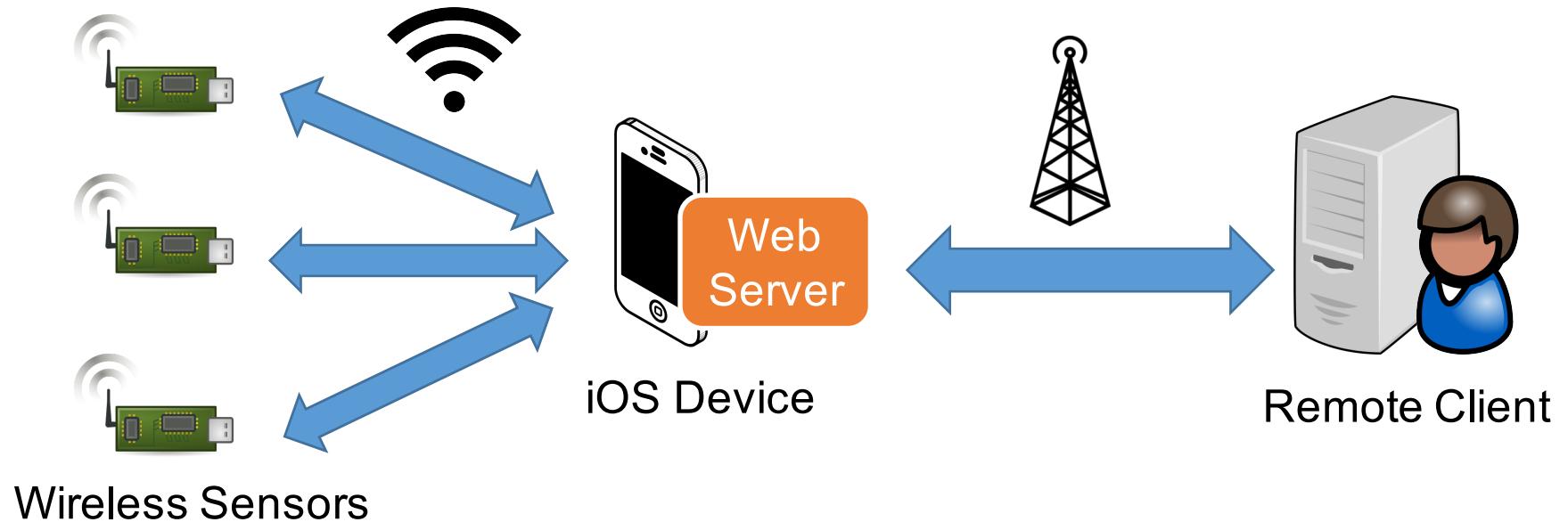
[source] <https://developer.apple.com/library/ios/documentation/DataManagement/Devpedia-CoreData/coreDataStack.html>

Swift: Core Data

```
let appDelegate =  
UIApplication.sharedApplication().delegate as!  
AppDelegate  
  
let managedContext =  
appDelegate.managedObjectContext  
  
//fetch all data  
let fetchRequest = NSFetchedRequest(entityName:  
"TheEntityName" )
```

Demo

Mobile Web Service Provisioning



Demo

File Upload Service hosted on iPhone

Interesting Tools

- <https://developer.apple.com/videos/wwdc/2015/>
- Mobile Web Server: CocoaHTTPServer
<https://github.com/robbiehanson/CocoaHTTPServer>
- GData Objective-C Client <https://code.google.com/p/gdata-objectivec-client/>
- RESTKit <https://github.com/RestKit/RestKit>
- AFNetworking <https://github.com/AFNetworking/AFNetworking>
- RESTful BPEL workflow execution engine
by Mobile & Cloud Lab

Thank you.
Questions?