# Building libraries for iOS
# Going native

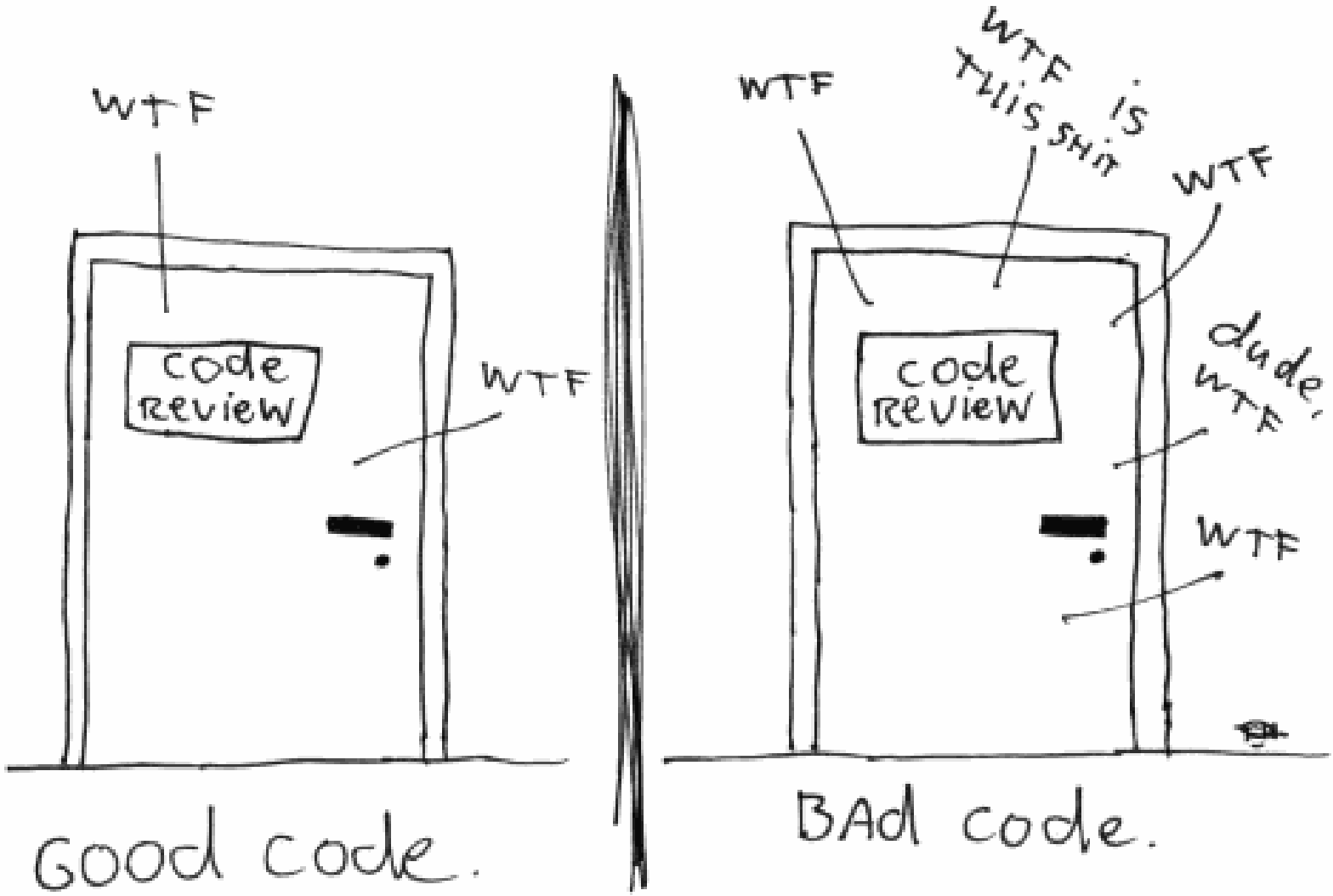
Alexander Dodatko
2014

The standard library saves programmers from having to reinvent the wheel.

Bjarne Stroustrup

http://bit.ly/1iqZ3JO

# The ONLY VALID MEASUREMENT OF Code QUALITY: WTFs/MINUTE




(c) 2008 Focus Shift

# Code Reuse is Important

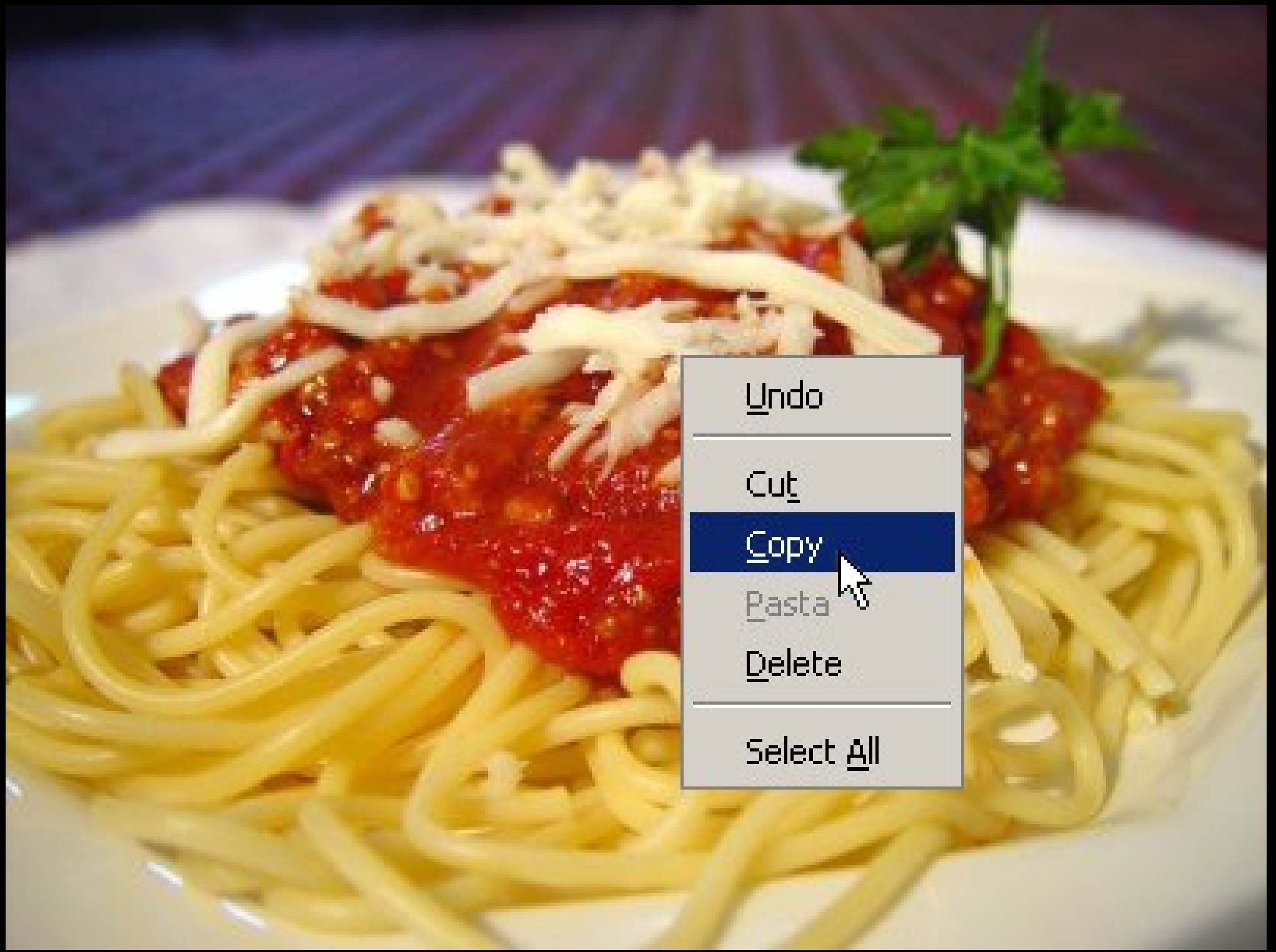
ASCENT
OF BOT

March 2008

http://engt.co/1pOLKH9

The fastest code is the code that reaches the market first

Nobody Cares


Undo
Cut
Copy
Pasta
Delete
Select All

August 2011

**CocoaPods**
@CocoaPods

Follow

@oliverfoggin Thanks! Just over one year old:
github.com/CocoaPods/Coco…

Reply Retweet Favorite More

GitHub

**CocoaPods**

An Objective-C library dependency manager. Contribute to CocoaPods development by creating an account on GitHub.

View on web

4:36 AM - 2 Oct 2012

Flag media

Google +Felix

Trends Worldwide 2004 - present All categories Web Search

Hot Searches
Top Charts
Explore
NCAA March Madness

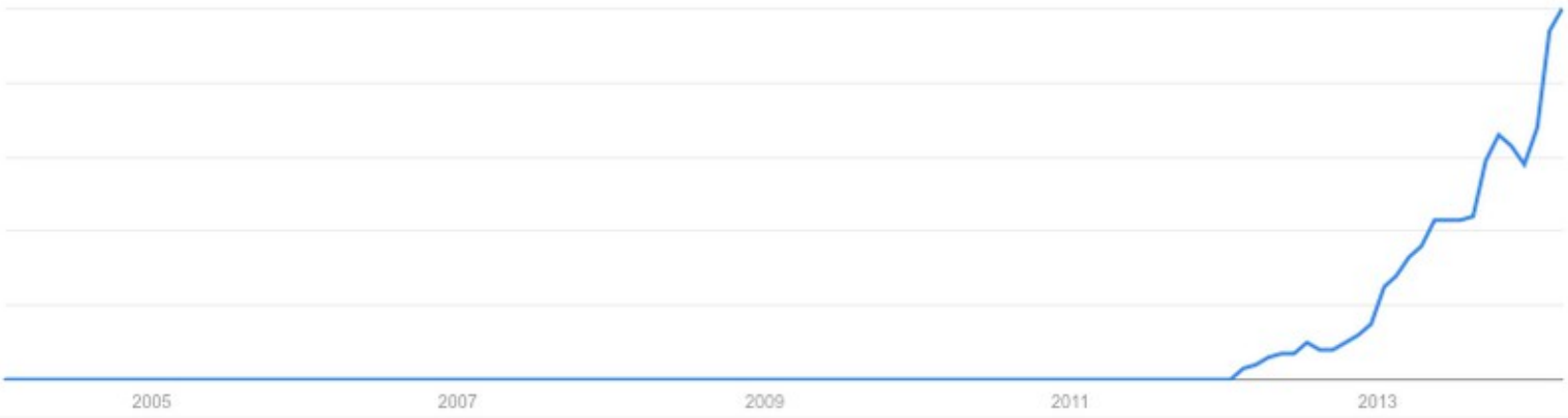
Compare
Search terms
Locations
Time ranges

cocoapods
Search term

+ Add term

Share

Interest over time News headlines Forecast



COCOAPODS ALL THE THINGS

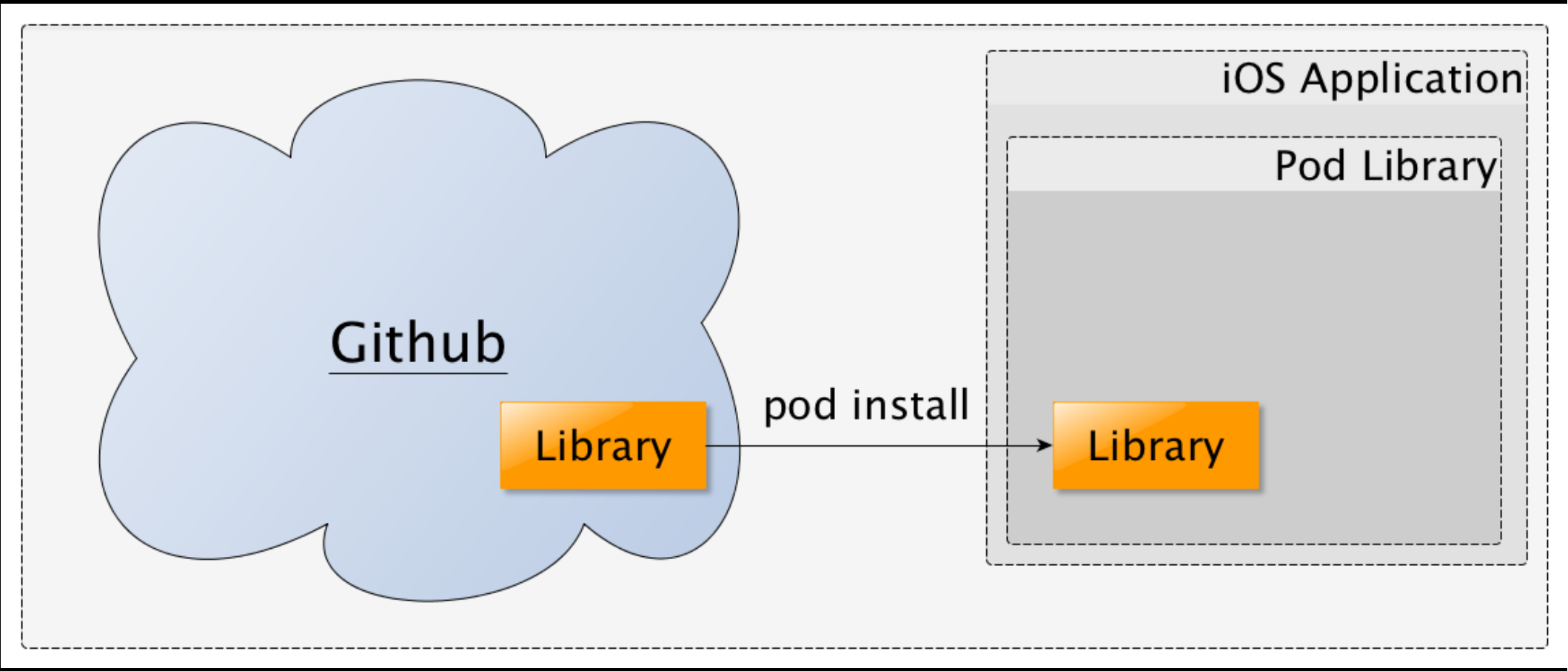
$ SUDO GEM INSTALL COCOAPODS
I HAVE NO IDEA WHAT I'M DOING
7 GEMS INSTALLED
memegenerator.net

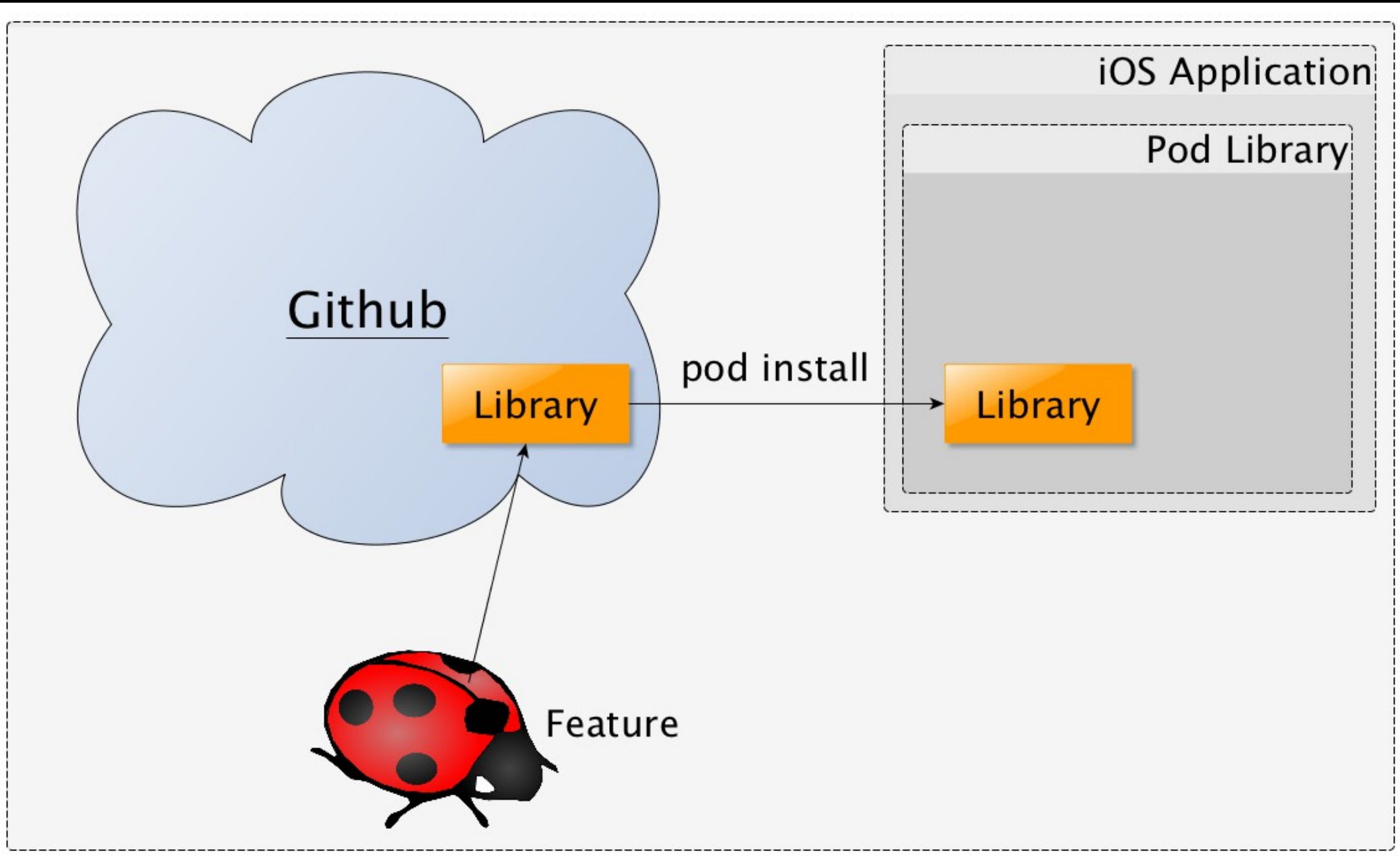
# pod install SomeAwesomeLibXYZ

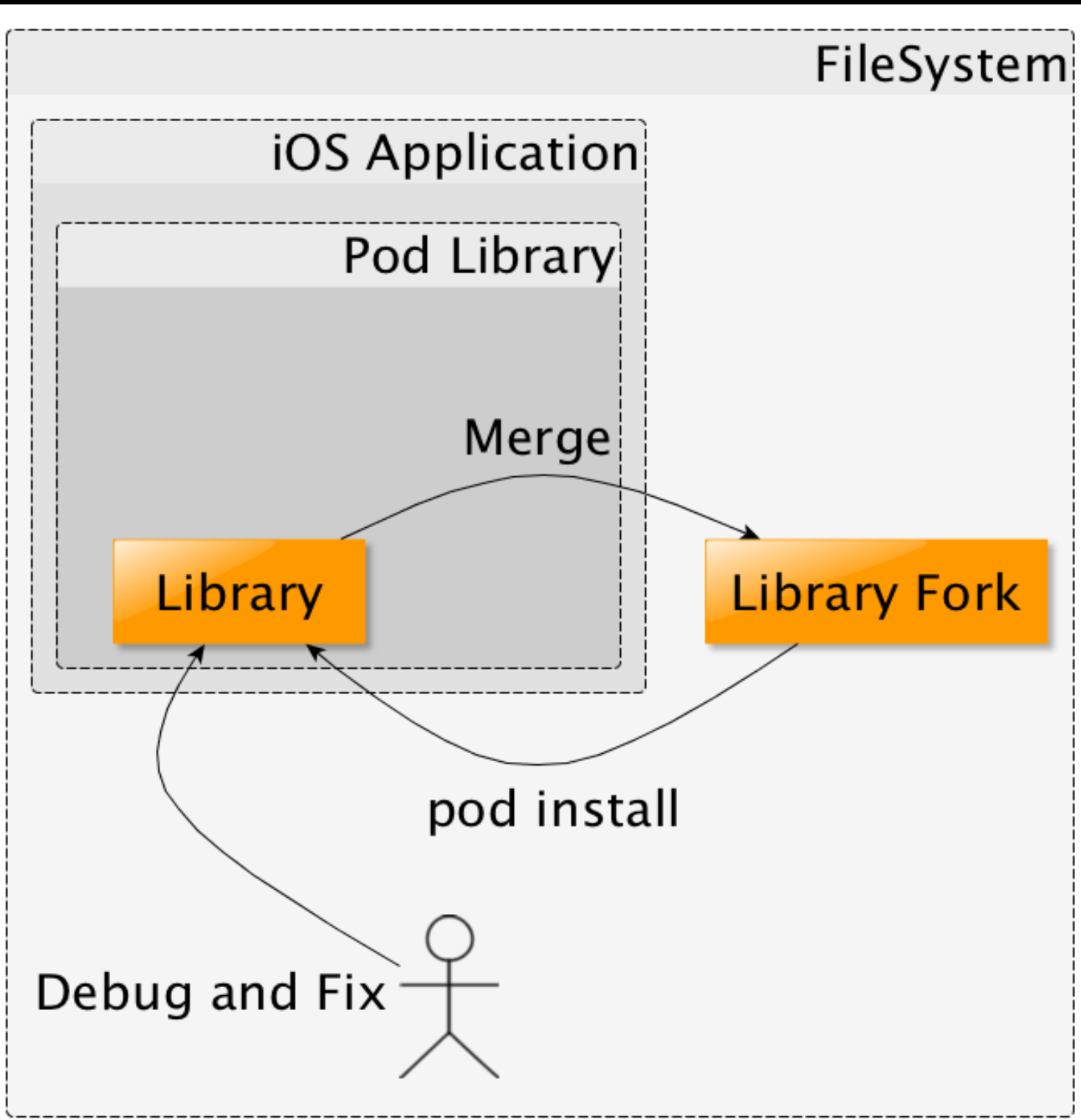
Except...

Library
© dianne j. hook

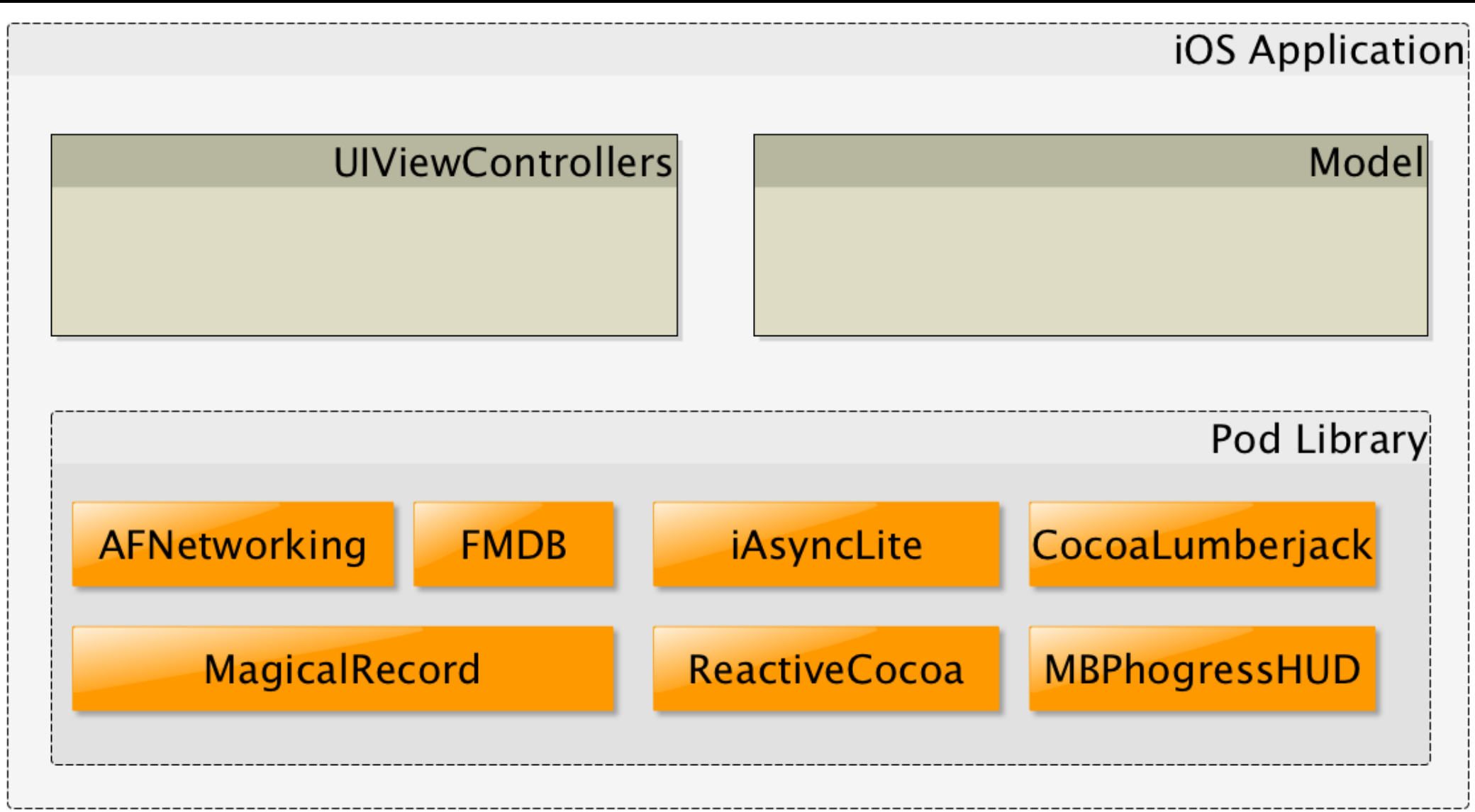
# Typical Workflow


iOS Application
Pod Library
Github
Library
pod install
Library

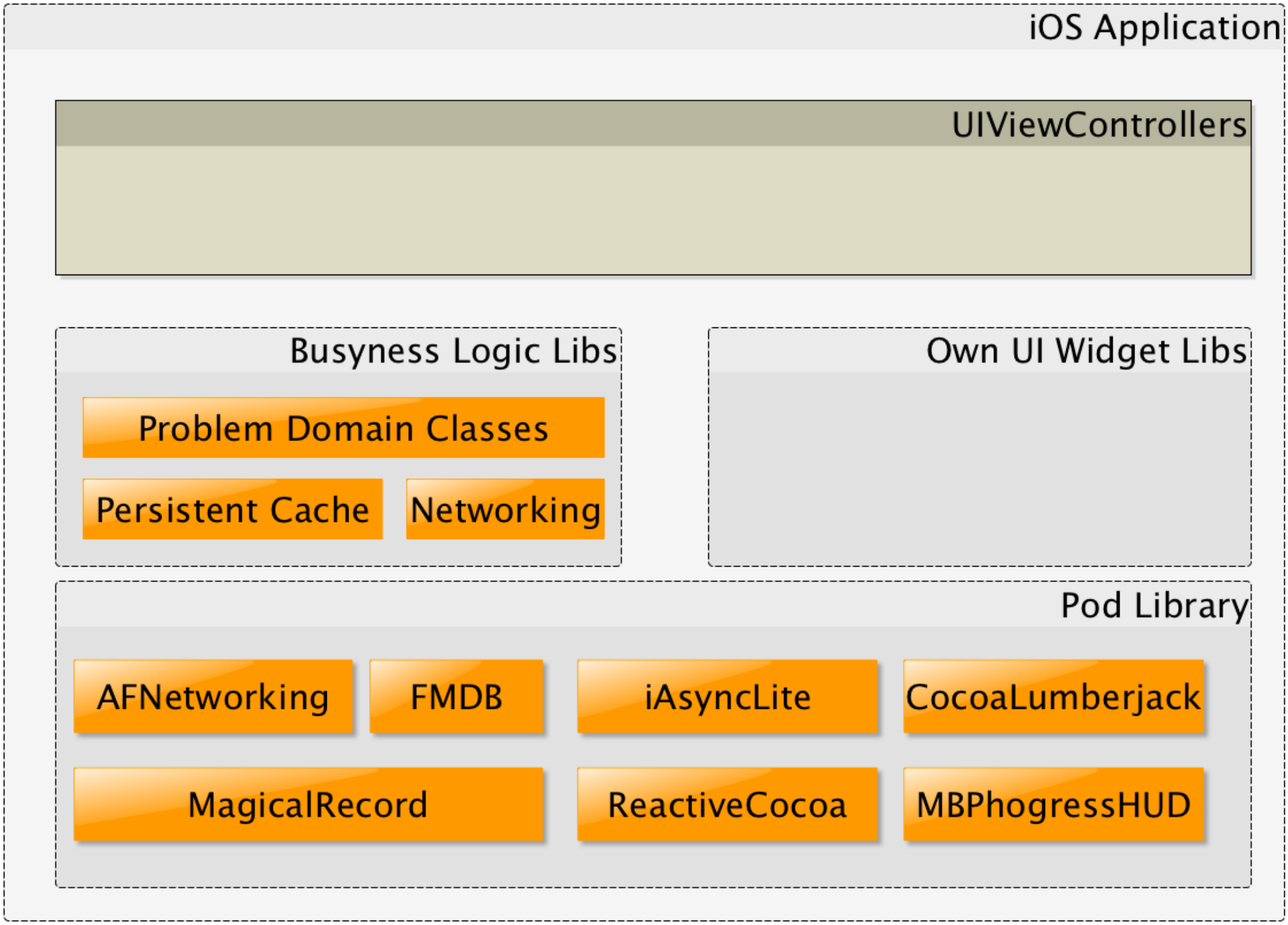

iOS Application
Pod Library
Github
Library
pod install
Library
Feature


FileSystem
iOS Application
Pod Library
Merge
Library
Library Fork
pod install
Debug and Fix

# Our own Busyness Logic


iOS Application
UIViewControllers
Model
Pod Library
AFNetworking
FMDB
iAsyncLite
CocoaLumberjack
MagicalRecord
ReactiveCocoa
MBPhogressHUD

# In-Place Editing

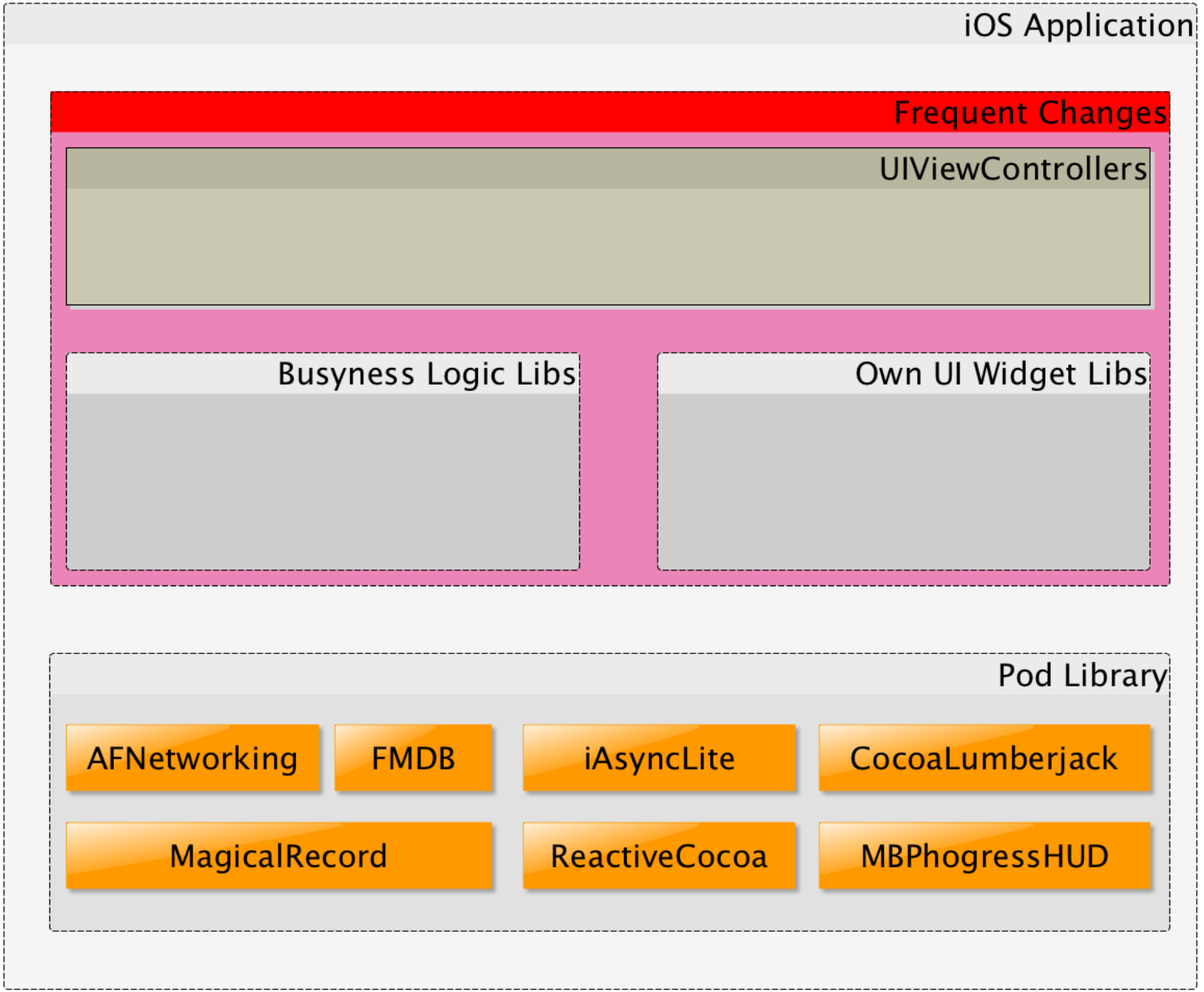

iOS Application
UIViewControllers
Busyness Logic Libs
Problem Domain Classes
Persistent Cache
Networking
Own UI Widget Libs
Pod Library
AFNetworking
FMDB
iAsyncLite
CocoaLumberjack
MagicalRecord
ReactiveCocoa
MBPhogressHUD

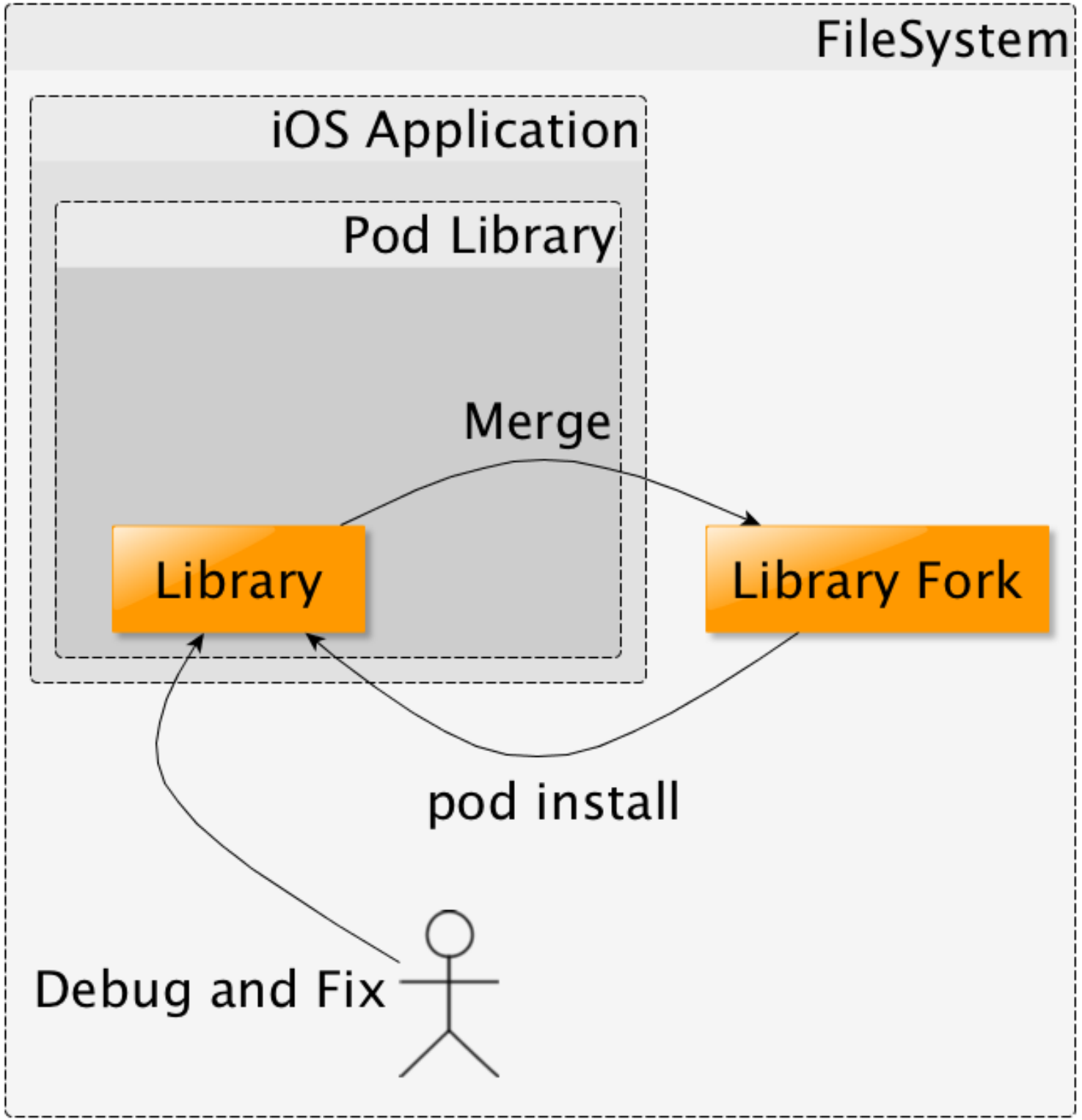
# Modular Architecture Benefits

Code Reuse

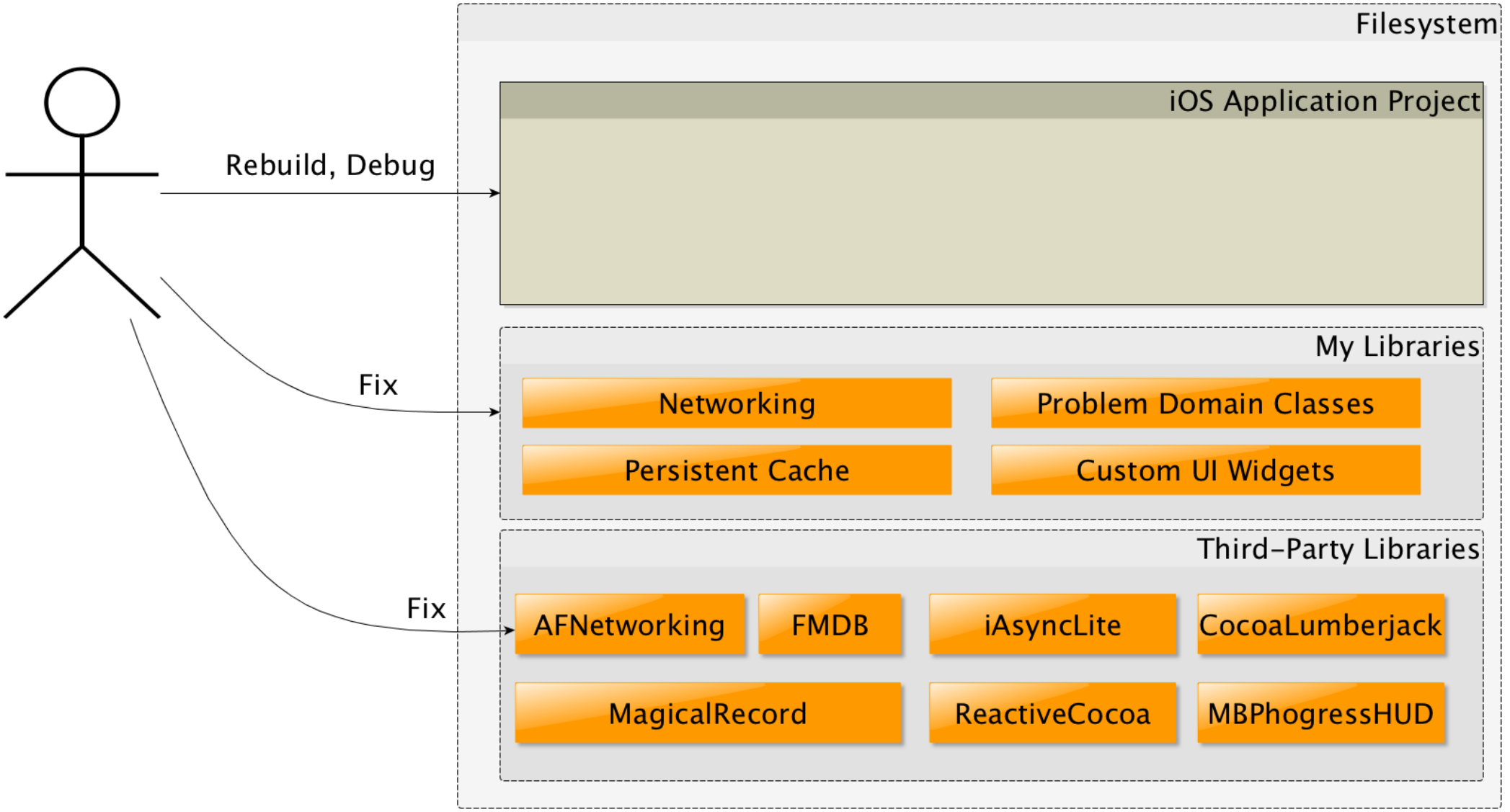
Test Coverage

Easier to Apply Changes

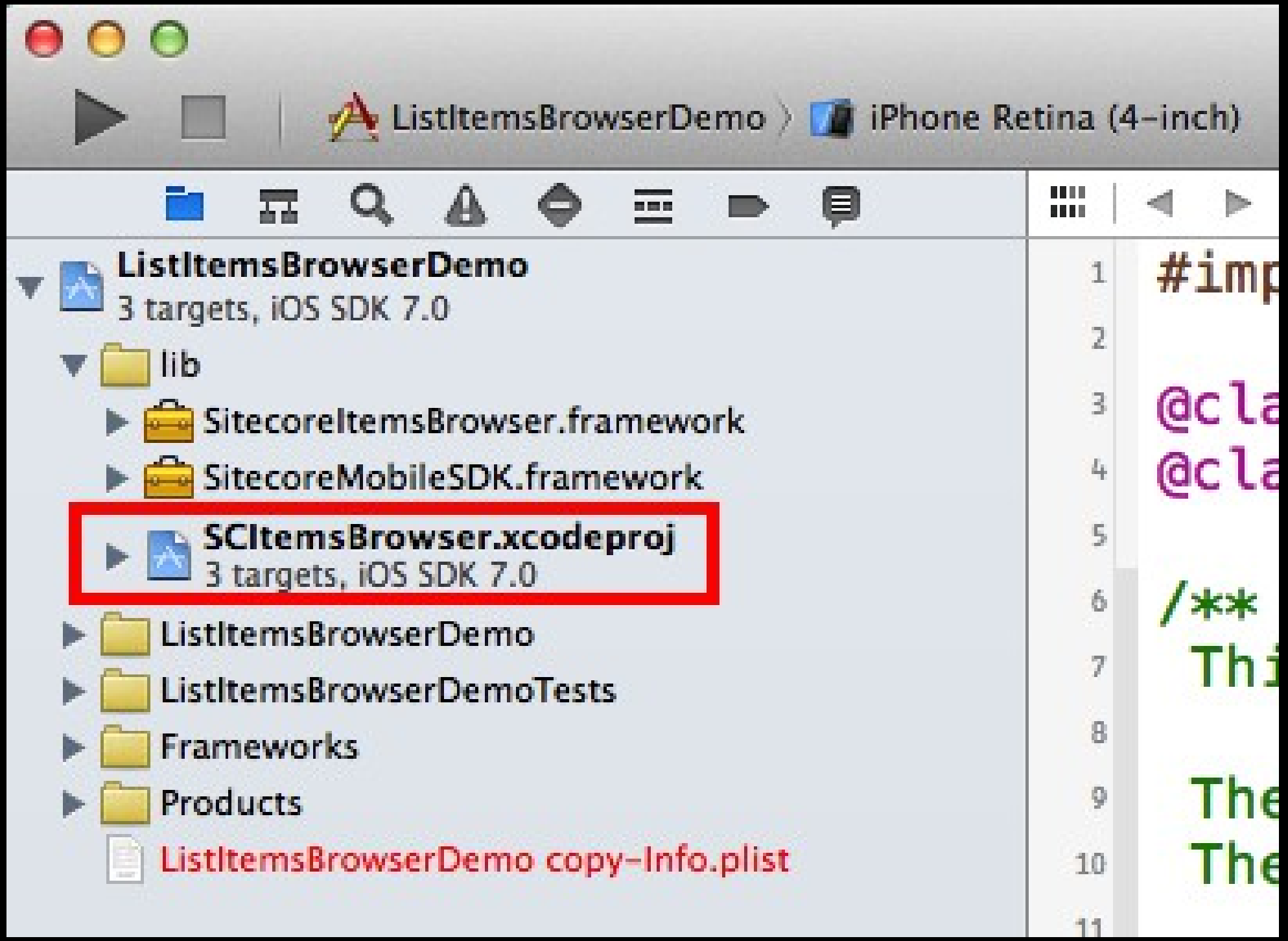

iOS Application
Frequent Changes
UIViewControllers
Busyness Logic Libs
Own UI Widget Libs
Pod Library
AFNetworking
FMDB
iAsyncLite
CocoaLumberjack
MagicalRecord
ReactiveCocoa
MBPhogressHUD

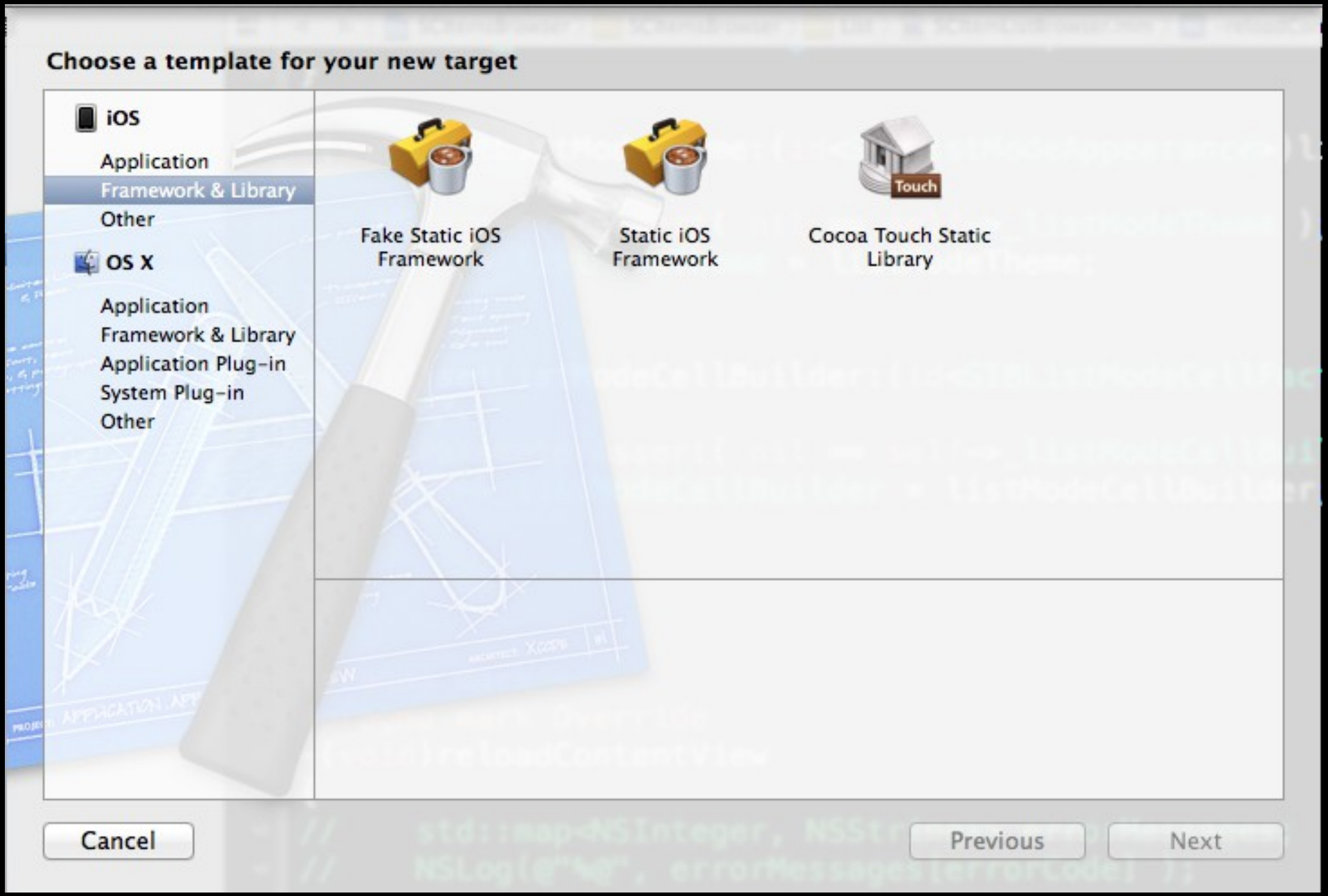

FileSystem
iOS Application
Pod Library
Merge
Library
Library Fork
pod install
Debug and Fix

# Going Native


Filesystem
iOS Application Project
Rebuild, Debug
Fix
My Libraries
Networking
Problem Domain Classes
Persistent Cache
Custom UI Widgets
Fix
Third-Party Libraries
AFNetworking
FMDB
iAsyncLite
CocoaLumberjack
MagicalRecord
ReactiveCocoa
MBPhogressHUD

# --no-integrate


ListItemsBrowserDemo
iPhone Retina (4-inch)
ListItemsBrowserDemo
3 targets, iOS SDK 7.0
lib
SitecoreItemsBrowser.framework
SitecoreMobileSDK.framework
SCItemsBrowser.xcodeproj
3 targets, iOS SDK 7.0
ListItemsBrowserDemo
ListItemsBrowserDemoTests
Frameworks
Products
ListItemsBrowserDemo copy-Info.plist
#imp
@cla
@cla
/**
Thi
The
The


Choose a template for your new target
iOS
Application
Framework & Library
Other
OS X
Application
Framework & Library
Application Plug-in
System Plug-in
Other
Fake Static iOS Framework
Static iOS Framework
Touch
Cocoa Touch Static Library
Cancel
Previous
Next

# Not Just Library Target

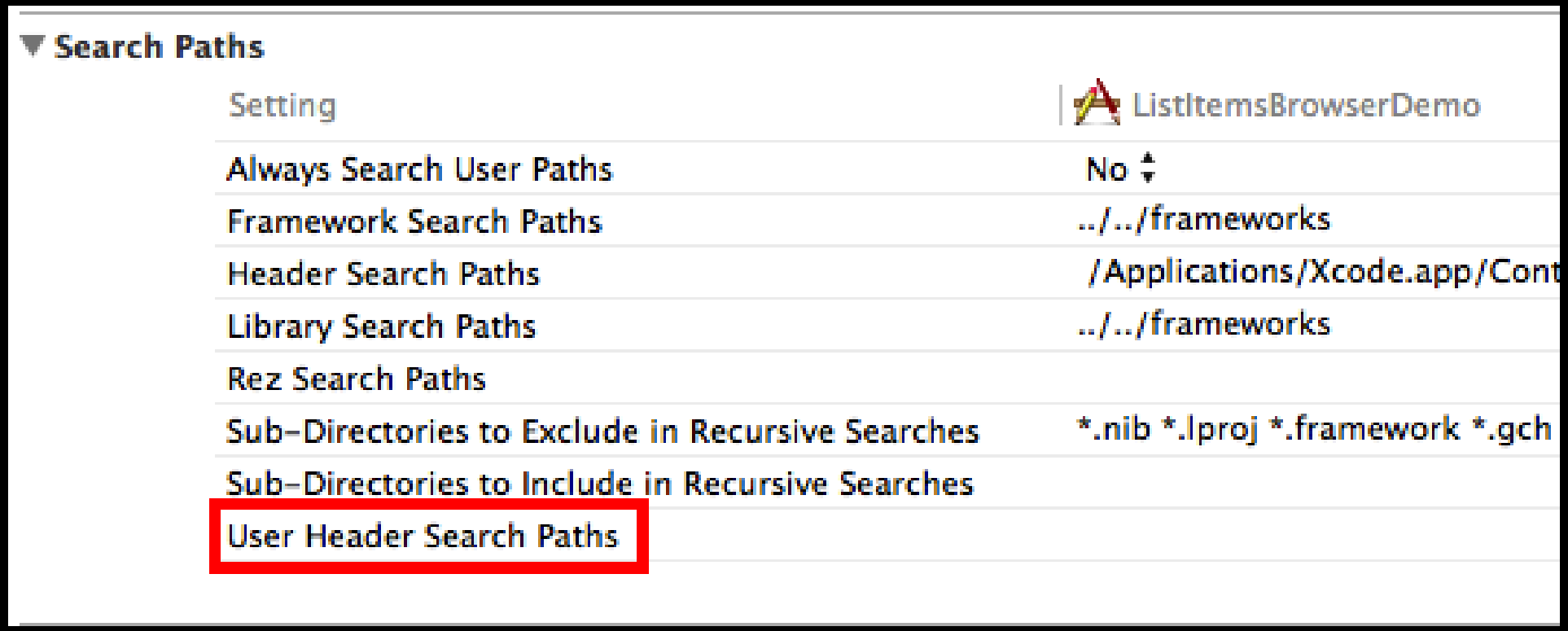
```
#import <AFNetworking/AFNetworking.h>
```

vs

```
#import "AFHTTPRequestOperation.h"
```

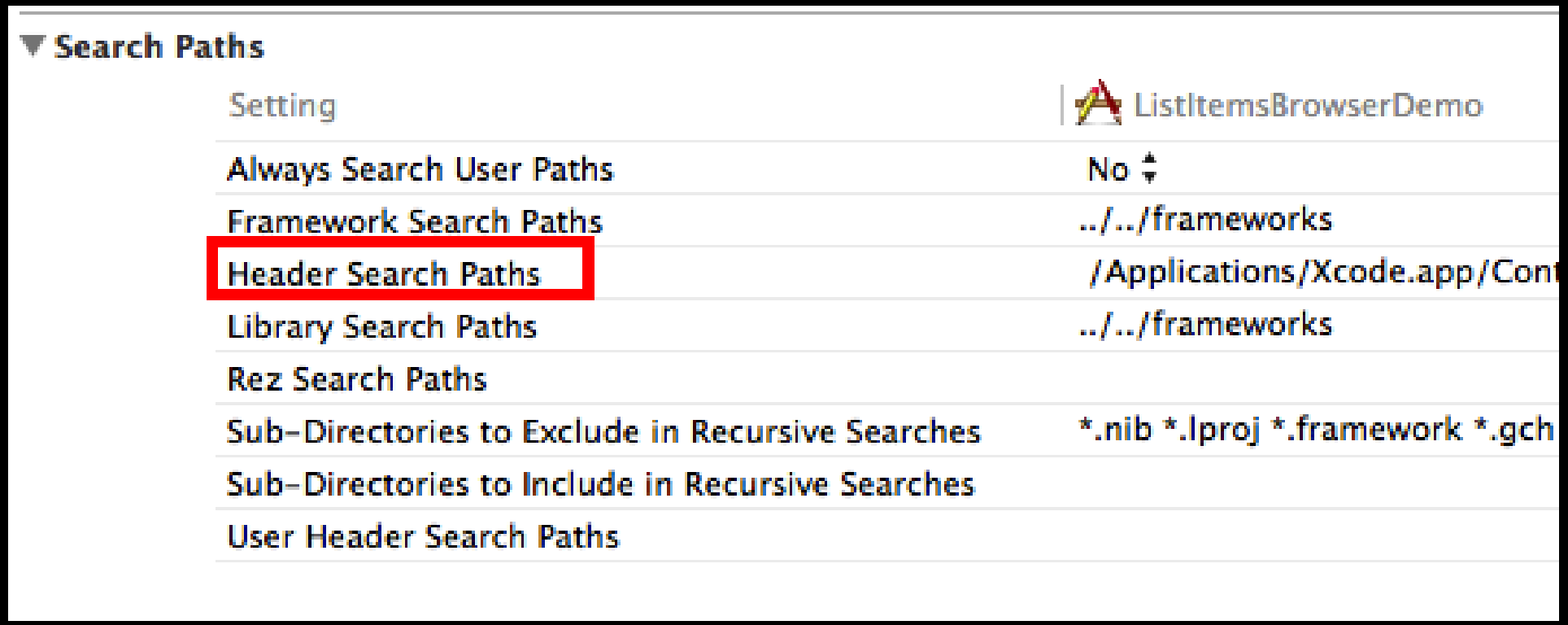
# For Implementation Files

#import "AFHTTPRequestOperation.h"



# For Public Headers

#import <AFNetworking/AFNetworking.h>

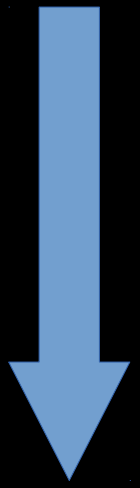


# Incapsulation

| | |
|---|---|
| $(inherited) | non-recursive |
| /Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/... | non-recursive |
| . | non-recursive |
| ./frameworks | non-recursive |
| ./frameworks/SitecoreMobileSDK.framework/Headers | recursive |

+ −

#import <AFNetworking/AFNetworking.h>



#import <Foundation/Foundation.h>

# Static Framework for iOS

- libOCMock.a
- OCMock
  - NSNotificationCenter+OCMAdditions.h
  - OCMArg.h
  - OCMConstraint.h
  - OCMock.h
  - OCMockObject.h
  - OCMockRecorder.h

- OCMock.framework
  - Headers
    - NSNotificationCenter+OCMAdditions.h
    - OCMArg.h
    - OCMConstraint.h
    - OCMock.h
    - OCMockObject.h
    - OCMockRecorder.h
  - OCMock
  - Resources
    - Info.plist

```
<plist version="1.0">
<dict>
    <key>CFBundleDevelopmentRegion</key>
    <string>English</string>
     <key>CFBundleIdentifier</key>
     <string>org.ocmock</string>
    <key>CFBundleInfoDictionaryVersion</key>
    <string>6.0</string>
     <key>CFBundlePackageType</key>
     <string>FMWK</string>
    <key>CFBundleSignature</key>
    <string>????</string>
     <key>CFBundleVersion</key>
     <string>2.2.3</string>
</dict>
</plist>
```

# Drag & Drop

Framework Search Path
is updated by Xcode

# Frameworks Reduce Compilation Time

```swift
//  AppDelegate
//  CrazyOnes
//
//  Copyright (c) 2014 Apple Inc. All rights reserved.

import Cocoa

class AppDelegate : NSObject, NSApplicationDelegate {
    var misfits: Array<NSObject>
    var rebels: Array<NSObject>
    var troubleMakers: Array<NSObject>

    @IBOutlet var window: NSWindow

    init() {
        misfits = Array()
        rebels = Array()
        troubleMakers = Array()
    }

    func applicationDidFinishLaunching(aNotification: NSNotification) {
        heresToTheCrazyOnes()
    }

    func heresToTheCrazyOnes() {
        for misfit in misfits {
            quote(misfit)
        }

        for rebel in rebels {
            glorify(rebel)
        }

        for troubleMaker in troubleMakers {
            vilify(troubleMaker)
        }
    }

    func quote(misfit: NSObject) {

    }

    func glorify(rebel: NSObject) {

    }

    func vilify(troubleMaker: NSObject) {

    }
}
```

# Sharing Swift Code is Painful

# Wrapper Obj-C Classes

swift -emit-module
swift -emit-library

# CocoaPods Makes you Care about Versioning

# CocoaPods set Rules for Library Vendors

# More Binary Pods

In My Opinion

For Development

and Unit Testing

For Distribution

For You

For End Users

# Library Dependencies and Linker

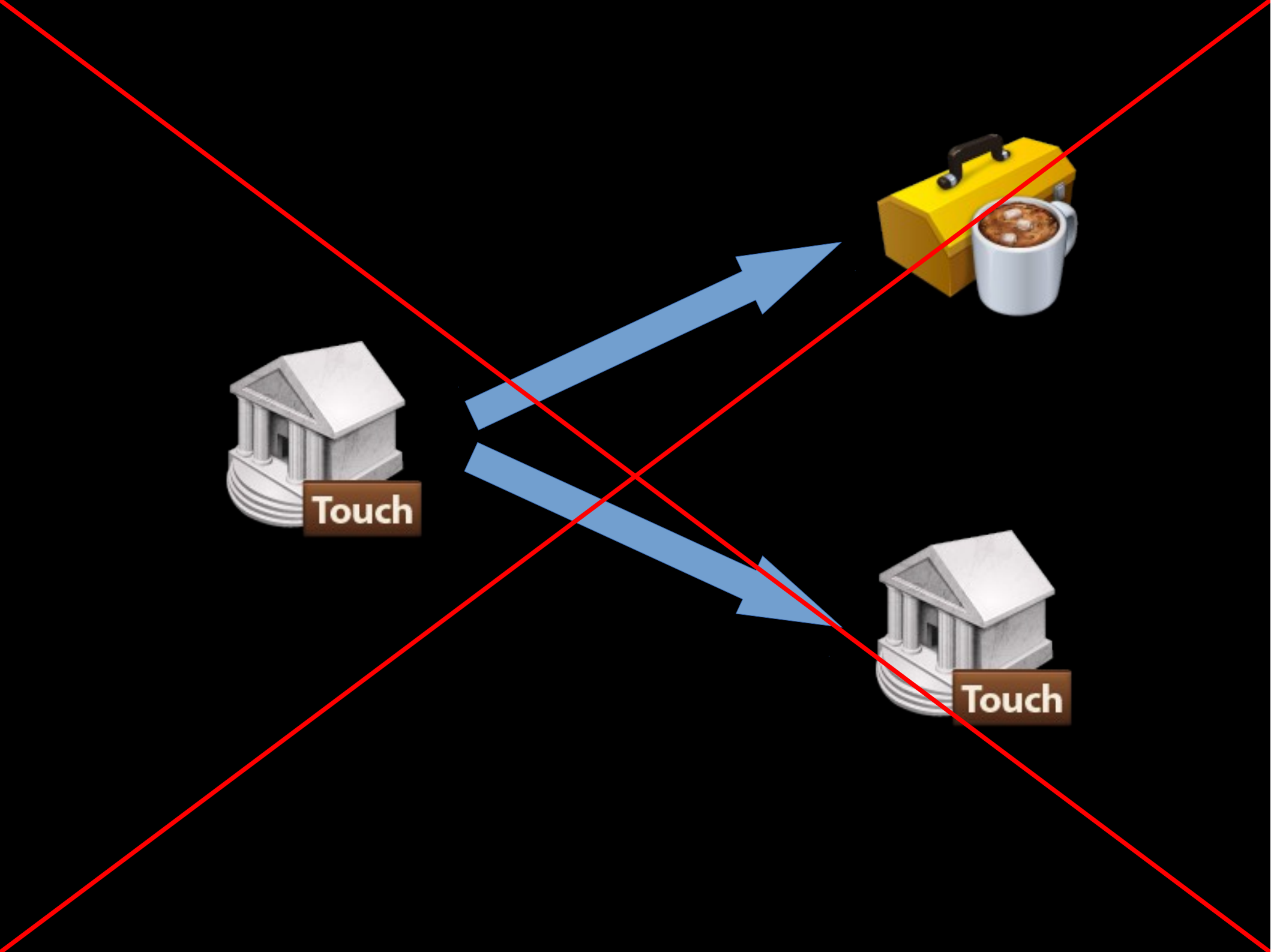
```
ld: duplicate symbol _main in /Users/samuelleeuwenburg/Library/Developer/Xcode/D
  SDLStackOverflowTest1-awqtenipezizyreqnvxupfbheywd/Build/Intermediates/SDLStack
  Debug/SDLStackOverflowTest1.build/Objects-normal/x86_64/SDLMain.o and /Users/sa
  Library/Developer/Xcode/DerivedData/SDLStackOverflowTest1-awqtenipezizyreqnvxup
  Intermediates/SDLStackOverflowTest1.build/Debug/SDLStackOverflowTest1.build/Obj
  main.o for architecture x86_64
clang: error: linker command failed with exit code 1 (use -v to see invocation)
```

Duplicate symbol _main in /Users/samuelleeuwenburg/Library/Developer/Xcode/DerivedData/SDLStackO

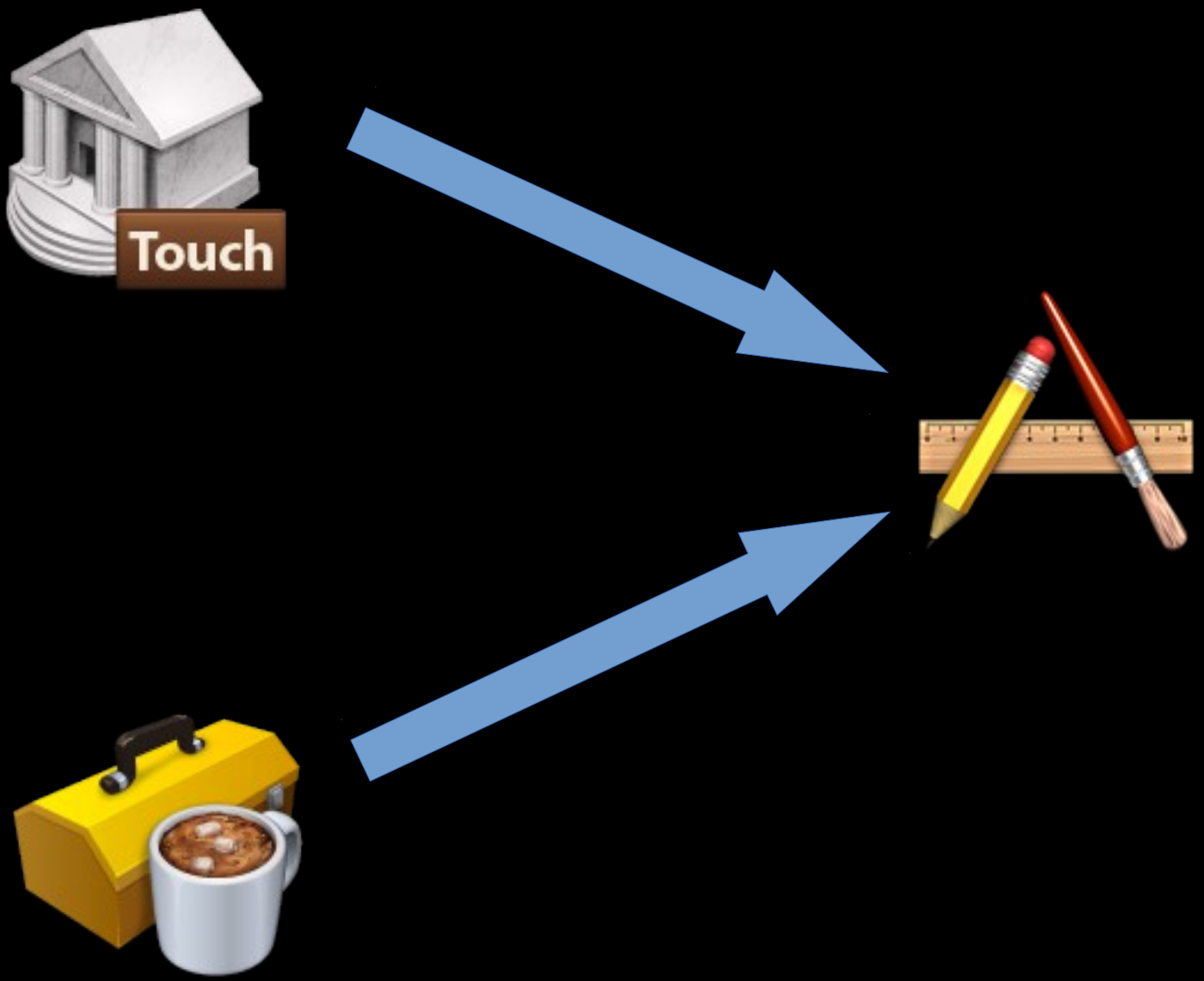
Linker command failed with exit code 1 (use -v to see invocation)

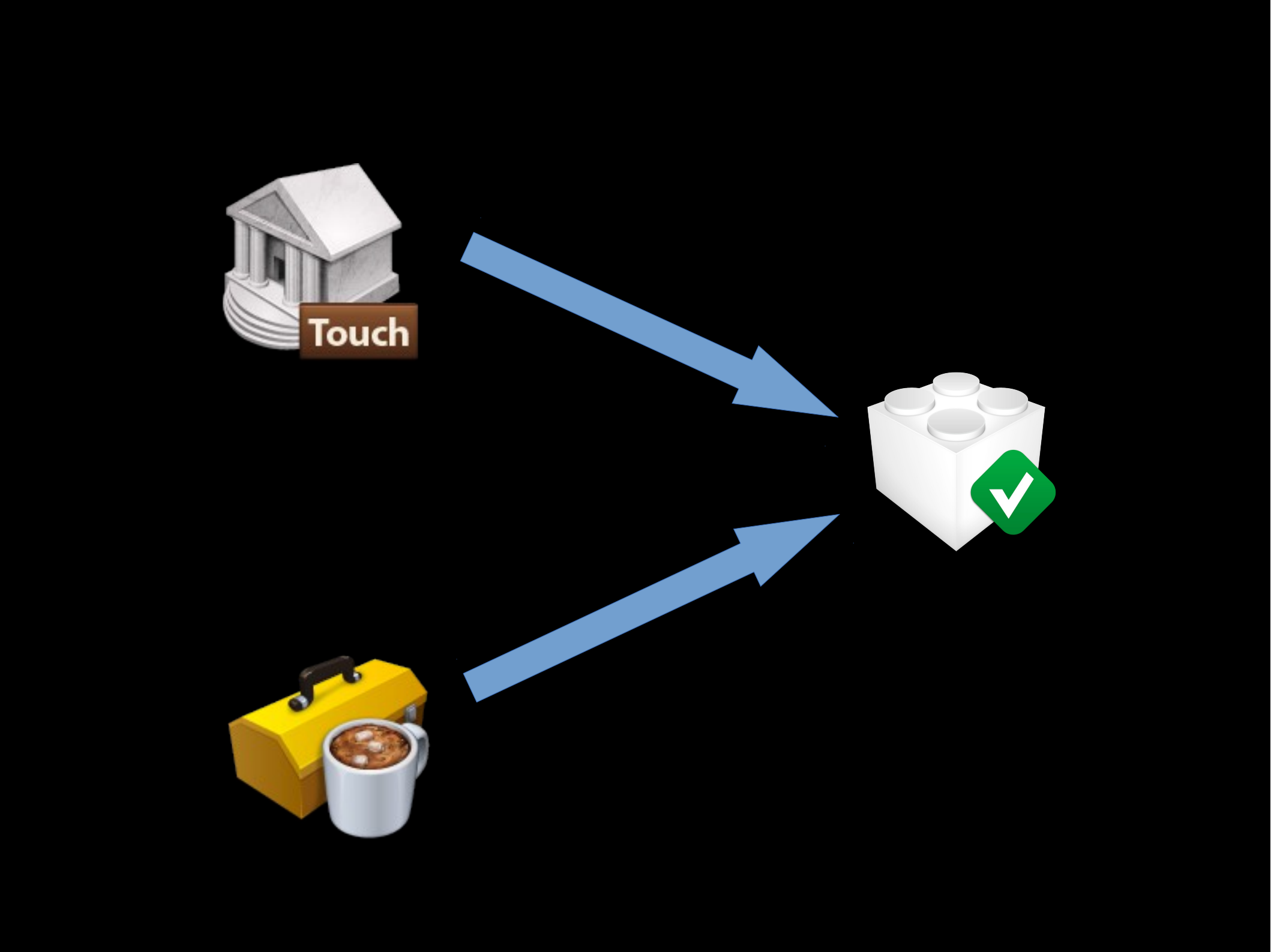
Activity Log Complete 9/26/12 3:41 PM

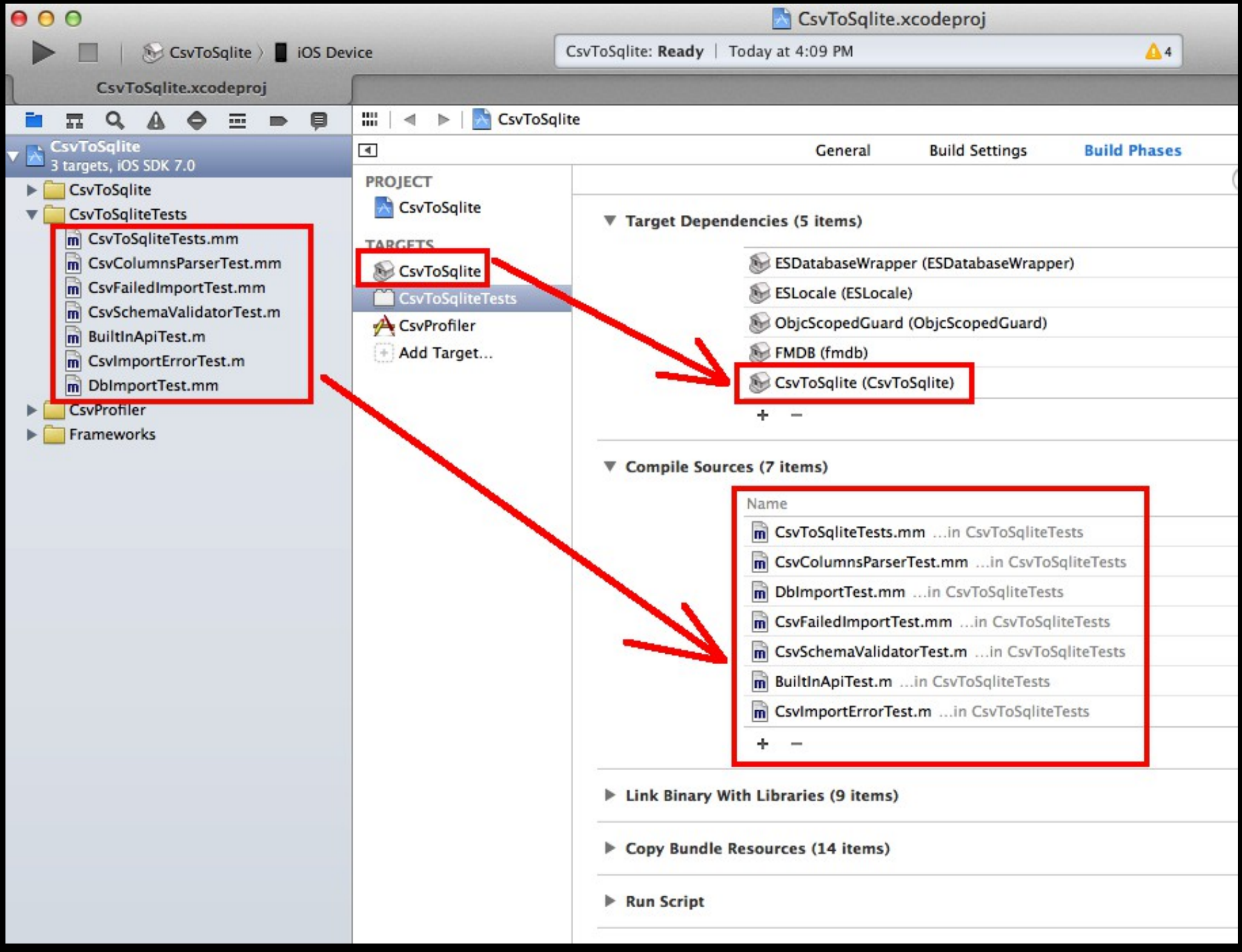
1 error, 1 warning


Touch
Touch

# Never Link Libraries into Other Libraries


Touch


Touch


CsvToSqlite.xcodeproj
CsvToSqlite 〉 iOS Device
CsvToSqlite: Ready | Today at 4:09 PM
4
CsvToSqlite.xcodeproj
CsvToSqlite
CsvToSqlite
3 targets, iOS SDK 7.0
CsvToSqlite
CsvToSqliteTests
CsvToSqliteTests.mm
CsvColumnsParserTest.mm
CsvFailedImportTest.mm
CsvSchemaValidatorTest.m
BuiltInApiTest.m
CsvImportErrorTest.m
DbImportTest.mm
CsvProfiler
Frameworks
General
Build Settings
Build Phases
PROJECT
CsvToSqlite
TARGETS
CsvToSqlite
CsvToSqliteTests
CsvProfiler
Add Target...
Target Dependencies (5 items)
ESDatabaseWrapper (ESDatabaseWrapper)
ESLocale (ESLocale)
ObjcScopedGuard (ObjcScopedGuard)
FMDB (fmdb)
CsvToSqlite (CsvToSqlite)
Compile Sources (7 items)
Name
CsvToSqliteTests.mm ...in CsvToSqliteTests
CsvColumnsParserTest.mm ...in CsvToSqliteTests
DbImportTest.mm ...in CsvToSqliteTests
CsvFailedImportTest.mm ...in CsvToSqliteTests
CsvSchemaValidatorTest.m ...in CsvToSqliteTests
BuiltInApiTest.m ...in CsvToSqliteTests
CsvImportErrorTest.m ...in CsvToSqliteTests
Link Binary With Libraries (9 items)
Copy Bundle Resources (14 items)
Run Script

# All Testable Code Should be in libraries

# Use Precompiled Headers

Reduce Comping Duration

```
#ifdef __OBJC__
    #ifdef __cplusplus
        // Objective-C++ headers go here
        #import "MyObjectiveCppClass.h"
    #endif

    // Objective-C headers go here
    #import <Foundation/Foundation.h>
#endif

#ifdef __cplusplus
    // Pure C++ headers go here
    #include <vector>
#endif

// Pure C headers go here
#include <math.h>
```

# Order by Dependencies

# Library Vendor's Checklist

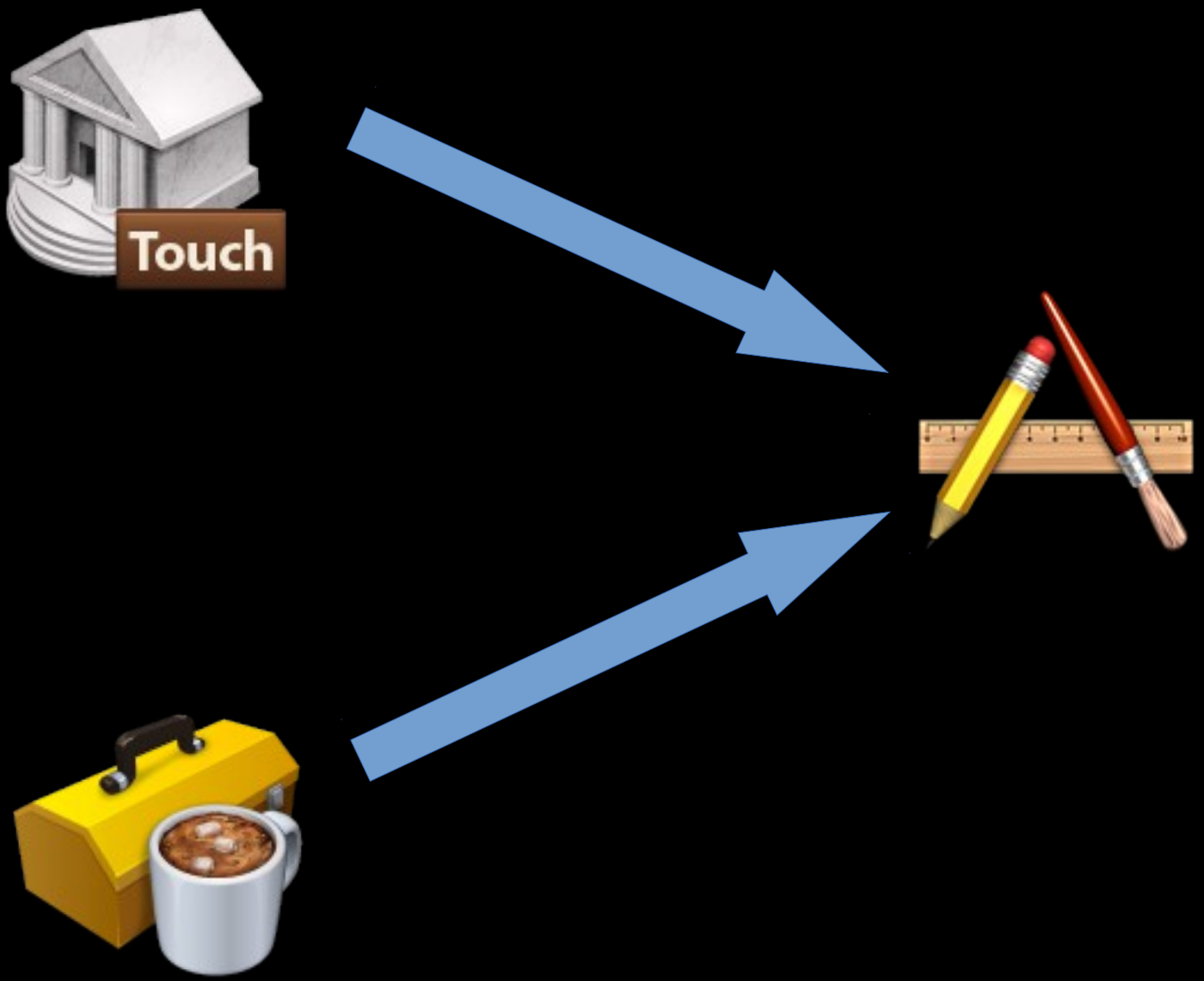
For Development

and Unit Testing

For Distribution

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

For Public Headers


Touch

# Do Write Unit Tests

With best possible coverage
you can afford

# Deliver Binary Pods

# Source Based Pods as a “Plan B”

# Native Rulezzz

Alexander Dodatko

@dodikk88