iOS Development

dewind@atomicobject.com
@dewind on Twitter

iOS Development

dewind@atomicobject.com
@dewind on Twitter

Developer Community

- ~ 50,000 registered iOS Developers [1]
- 500,000 applications [2]
- Objective-C is the 10th most popular language on GitHub
- Thousands of third party libraries on GitHub
- Of those 400+ appear to be well maintained and popular*

[1] http://www.quora.com/iOS-Development/How-many-registered-iPhone-app-developers-are-there-and-is-there-a-directory-somewhere
[2] http://en.wikipedia.org/wiki/App_Store_(iOS)
* An assumption based on any third party library that has 100+ watchers

How it has impacted Apple

- More developers means more competition, innovation, and *bitching*
- iOS has had 5 major releases in 5 years
- These releases were non-trivial for developers.
- It's a copy-cat-world and Apple is no exception

iOS 5

- iCloud
  - Backup
  - Storage
- ARC (Automatic Reference Counting)
- Storyboard
- Core Image
- GLKit
- Twitter

Third Party Libraries

Third Party Libraries

- Tons of them
  - https://github.com/languages/Objective-C
  - http://cocoaobjects.com/
  - http://cocoacontrols.com/
  - http://www.mikeash/pyblog/

Third Party Libraries

- Tons of them
  - https://github.com/languages/Objective-C
  - http://cocoaobjects.com/
  - http://cocoacontrols.com/
  - http://www.mikeash/pyblog/
- Ergo there is a lot of garbage as well

Third Party Libraries

- Tons of them
  - https://github.com/languages/Objective-C
  - http://cocoaobjects.com/
  - http://cocoacontrols.com/
  - http://www.mikeash/pyblog/
- Ergo there is a lot of garbage as well
- Focus on the ones that are maintained

Popular Libraries

- Three20
- ASIHttpRequest
- JSONKit
- ShareKit
- EGOTableViewPullRefresh
- MBProgressHUD
- AQGridView
- Tapku
- Cocos2D
- InAppKitSettings
- UIKit-Artwork Extractor
- Mogenerator
- PSStackedView
- BCTabBarController
- EGOImageView
- IPOfflineQueue

Libraries I've Used

- Kiwi
- LRResty
- ASIHTTPRequest
- Three20
- InAppSettingsKit
- MAZeroingWeakRef
- MKInfoPanel
- JSONKit
- MAObjCRuntime
- CocoaPods

Libraries I’ve Used

- Kiwi
- LRResty
- ASIHTTPRequest
- Three20
- InAppSettingsKit
- MAZeroingWeakRef
- MKInfoPanel
- JSONKit
- MAObjCRuntime
- CocoaPods

Objection

Libraries I’ve Used

- Kiwi
- LRResty
- ASIHTTPRequest
- Three20
- InAppSettingsKit
- MAZeroingWeakRef
- MKInfoPanel
- JSONKit
- MAObjCRuntime
- CocoaPods

Objection

http://www.objection-framework.org

iOS Basics

- Objective-C
    - Superset of C
    - Dynamically typed
    - Dynamic dispatch (Runtime message passing)
    - Powerful Runtime

iOS Basics

- Objective-C
  - Superset of C
  - Dynamically typed
  - Dynamic dispatch (Runtime message passing)
  - Powerful Runtime
  - Odd syntax

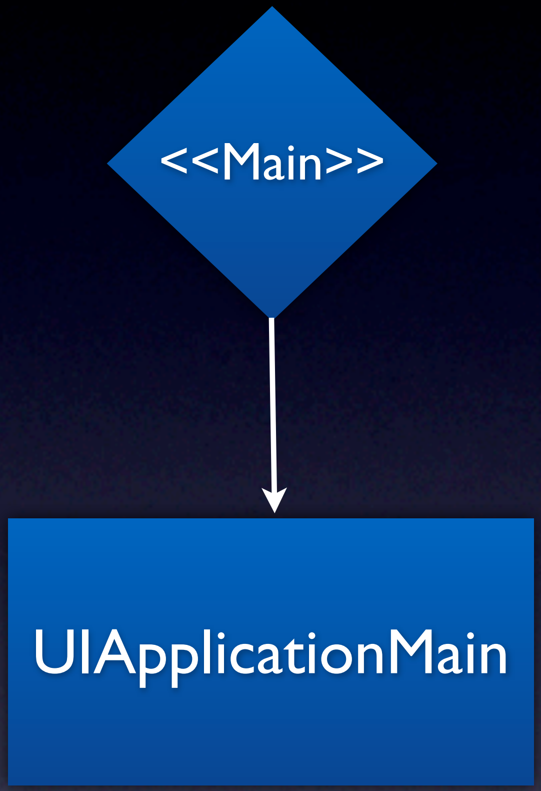
iOS Basics

- Objective-C
  - Superset of C
  - Dynamically typed
  - Dynamic dispatch (Runtime message passing)
  - Powerful Runtime
  - Odd syntax
    - Get over it

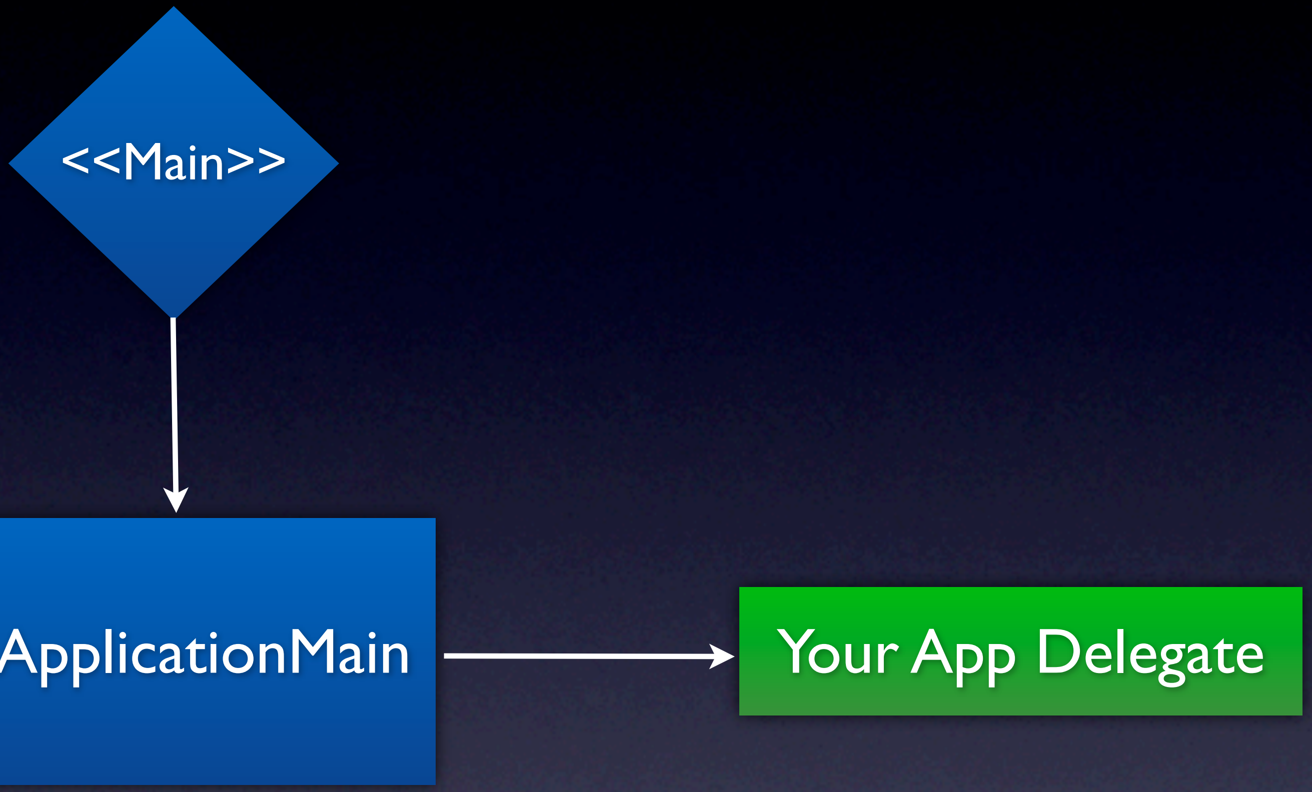
How an app starts

How an app starts

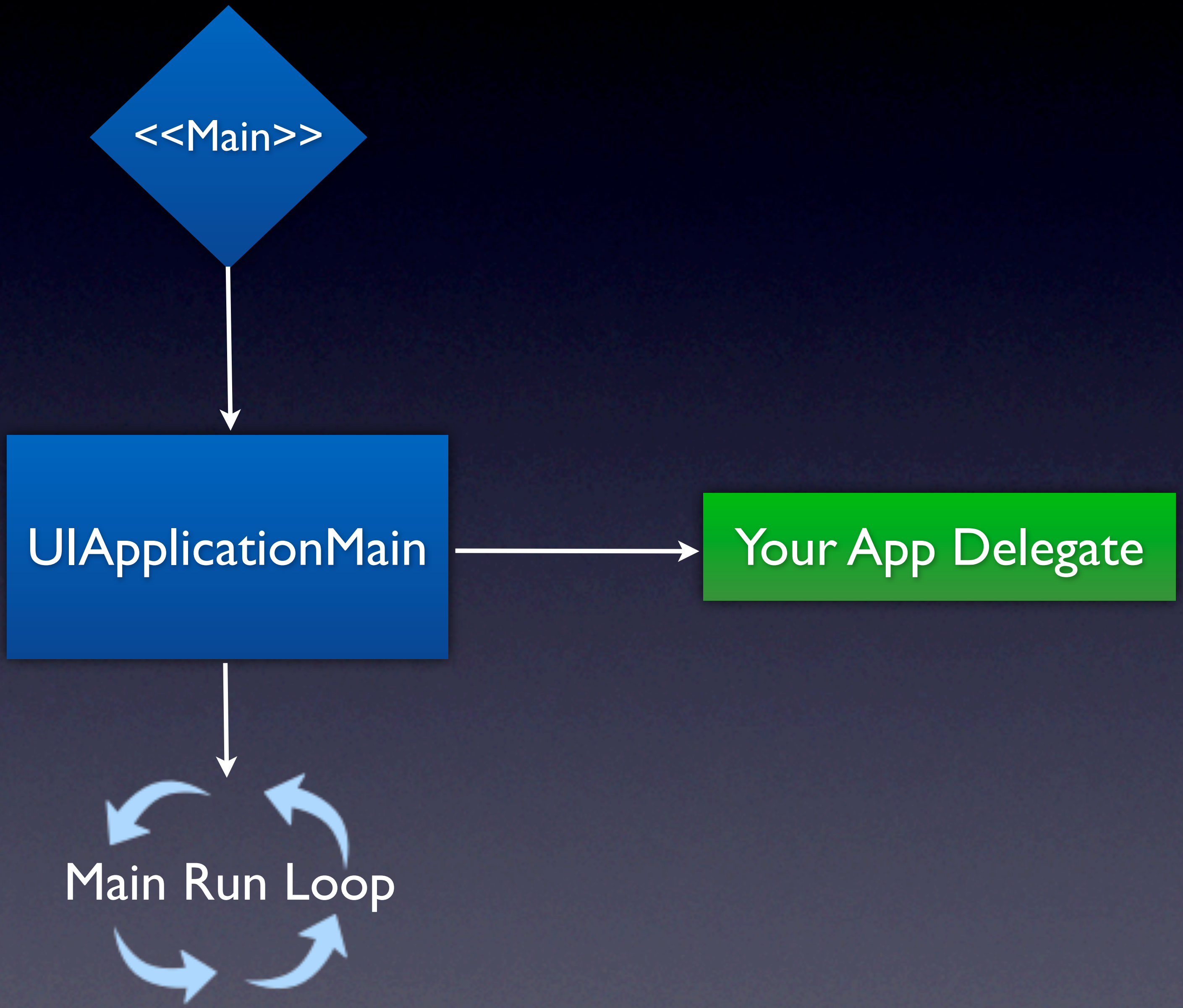
How an app starts



How an app starts



How an app starts



Example

- ARC
- iCloud
- Core Image
- GCD
- Some third party libraries

ARC

ARC

- Automatic Reference Counting

ARC

- Automatic Reference Counting
- Not Garbage Collection

ARC

- Automatic Reference Counting
- Not Garbage Collection
- Static Analysis (Compile Time)

ARC

- Automatic Reference Counting
- Not Garbage Collection
- Static Analysis (Compile Time)
- Think in terms of an object graph

ARC

- Automatic Reference Counting
- Not Garbage Collection
- Static Analysis (Compile Time)
- Think in terms of an object graph
  - @property(nonatomic, **strong**) instead of

ARC

- Automatic Reference Counting
- Not Garbage Collection
- Static Analysis (Compile Time)
- Think in terms of an object graph
    - @property(nonatomic, **strong**) instead of
    - @property(nonatomic, **retain**)

ARC

- Automatic Reference Counting
- Not Garbage Collection
- Static Analysis (Compile Time)
- Think in terms of an object graph
    - @property(nonatomic, **strong**) instead of
    - @property(nonatomic, **retain**)
    - @property(nonatomic, **weak**) instead of

ARC

- Automatic Reference Counting
- Not Garbage Collection
- Static Analysis (Compile Time)
- Think in terms of an object graph
  - @property(nonatomic, **strong**) instead of
  - @property(nonatomic, **retain**)
  - @property(nonatomic, **weak**) instead of
  - @property(nonatomic, **assign**)

ARC Continued

ARC Continued

- Going from C to Objective-C is a little trickier

ARC Continued

- Going from C to Objective-C is a little trickier
- id obj = (id)CFGetSomething(...) - Non-ARC

ARC Continued

- Going from C to Objective-C is a little trickier
- id obj = (id)CFGetSomething(...) - Non-ARC
- id obj = (__bridged id)CFGetSomething(...) - ARC

ARC Continued

- Going from C to Objective-C is a little trickier
- id obj = (id)CFGetSomething(...) - Non-ARC
- id obj = (__bridged id)CFGetSomething(...) - ARC
- But...copying...is different

ARC Continued

- Going from C to Objective-C is a little trickier
- id obj = (id)CFGetSomething(...) - Non-ARC
- id obj = (__bridged id)CFGetSomething(...) - ARC
- But...copying...is different
- id obj = (__bridged_transfer id)CFCopyValue(...)

ARC Continued

- Going from C to Objective-C is a little trickier
- id obj = (id)CFGetSomething(...) - Non-ARC
- id obj = (__bridged id)CFGetSomething(...) - ARC
- But...copying...is different
- id obj = (__bridged_transfer id)CFCopyValue(...)
    - Transfers ownership by balancing copy with release

ARC Continued

- Going from C to Objective-C is a little trickier
- id obj = (id)CFGetSomething(...) - Non-ARC
- id obj = (__bridged id)CFGetSomething(...) - ARC
- But...copying...is different
- id obj = (__bridged_transfer id)CFCopyValue(...)
  - Transfers ownership by balancing copy with release
  - Otherwise CF object would leak

iCloud

iCloud

- Saves documents/preferences to a user’s cloud account

iCloud

- Saves documents/preferences to a user’s cloud account
- Those documents/preferences are shared between different instances of the app under the same account

iCloud

- Saves documents/preferences to a user’s cloud account
- Those documents/preferences are shared between different instances of the app under the same account
- Magic, Right?

Preflight Tasks

Preflight Tasks

- Enable entitlements in project

Preflight Tasks

- Enable entitlements in project
- Define key/value and containers

Preflight Tasks

- Enable entitlements in project
- Define key/value and containers
  - com.companyname.AContainer

Preflight Tasks

- Enable entitlements in project
- Define key/value and containers
  - com.companyname.AContainer
- Create mobile provisioning profile with iCloud enabled

Preflight Tasks

- Enable entitlements in project
- Define key/value and containers
  - com.companyname.AContainer
- Create mobile provisioning profile with iCloud enabled
- Add document types

Preflight Tasks

- Enable entitlements in project
- Define key/value and containers
  - com.companyname.AContainer
- Create mobile provisioning profile with iCloud enabled
- Add document types
- And now you can work on actual software

Document Storage

Document Storage

- Get app sandbox Documents directory and iCloud directory

Document Storage

- Get app sandbox Documents directory and iCloud directory
- UIDocument + NSFileWrapper FTW

Document Storage

- Get app sandbox Documents directory and iCloud directory
- UIDocument + NSFileWrapper FTW
  - Implement serialization

Document Storage

- Get app sandbox Documents directory and iCloud directory
- UIDocument + NSFileWrapper FTW
    - Implement serialization
    - Implement deserialization

Document Storage

- Get app sandbox Documents directory and iCloud directory
- UIDocument + NSFileWrapper FTW
  - Implement serialization
  - Implement deserialization
  - Wrap directories/files using NSFileWrapper

Document Storage

- Get app sandbox Documents directory and iCloud directory
- UIDocument + NSFileWrapper FTW
  - Implement serialization
  - Implement deserialization
  - Wrap directories/files using NSFileWrapper
- Save to app sandbox and then move to iCloud

Core Image

- Provides better mechanisms for image manipulation/filters
- Introduces concept of ‘detector types’ to allow for object detection in an image
  - Such as faces
- Near real-time processing of video

GCD, Briefly

GCD, Briefly

- Grand Central Dispatch

GCD, Briefly

- Grand Central Dispatch
- Built to improve and support concurrent code execution using blocks
  - Intelligently manages thread pools by considering CPU constraints

GCD, Briefly

- Grand Central Dispatch
- Built to improve and support concurrent code execution using blocks
    - Intelligently manages thread pools by considering CPU constraints
- dispatch_async(queue, ^{ ...code...})
    - Global queues (priority based)
    - Main Queue (Main Run Loop)
    - Custom serial and concurrent queues