iOS Basics with Swift

Introduction

➜ Name

➜ Experience

➜ Expectation

Schedule

	Day 1	Day 2	Day 3
9:00 - 10:30	Introduction + Setup	Storyboards	Networking
10:45 - 12:15	Swift I	Navigation	Dependencies + Wishlist
13:15 - 14:45	Swift II	Lists	Testing + Platform
15:00 - 16:30	Apps + Xcode	The Game	Distribution + Feedback

appropos
academy

GitHub Repository:

github.com/ernesto-elsaesser/ios-training

Let's start!

Please interrupt me if …

- … you have a question
- … you want me to repeat something
- … you need a break

appropos academy

iOS

What's different?

Limitations

- Screen

- Energy

- Mobile Data

- Memory

- Input

Expectations

- Stability

- Performance

- Design

- Responsiveness

- Usability (HIG)

https://developer.apple.com/design/human-interface-guidelines/ios/overview/themes/

Swift

A New Language

After Apple unveiled the Swift programming language, it quickly became one of the **fastest growing** *languages in history. Swift makes it easy to write software that is incredibly* **fast and safe** *by design. Now that Swift is* **open source***, you can help make the best general purpose programming language available everywhere.*

For students, learning Swift has been a great introduction to modern programming concepts and best practices. And because it is now open, their Swift skills will be able to be applied to an even broader range of platforms, from **mobile** *devices to the* **desktop** *to the* **cloud***.*

- **The Swift Team** - https://docs.swift.org
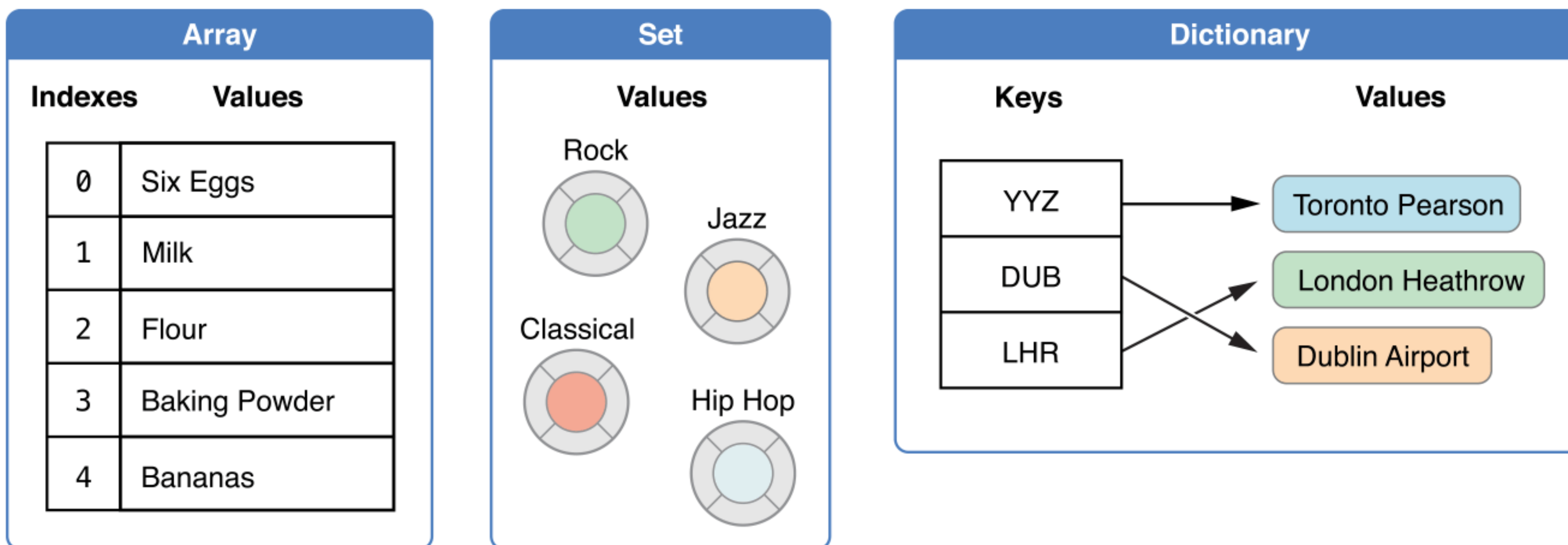
A Safe Language

Swift defines away large classes of common programming errors by adopting modern programming patterns:

• *Variables are always initialized before use*

• *Array indices are checked for out-of-bounds errors**

• *Integers are checked for overflow*

• *Optionals ensure that null values are handled explicitly*

• *Memory is managed automatically*

• *Error handling allows controlled recovery from unexpected failures*

- **The Swift Programming Language (Swift 5)** - https://docs.swift.org/swift-book/
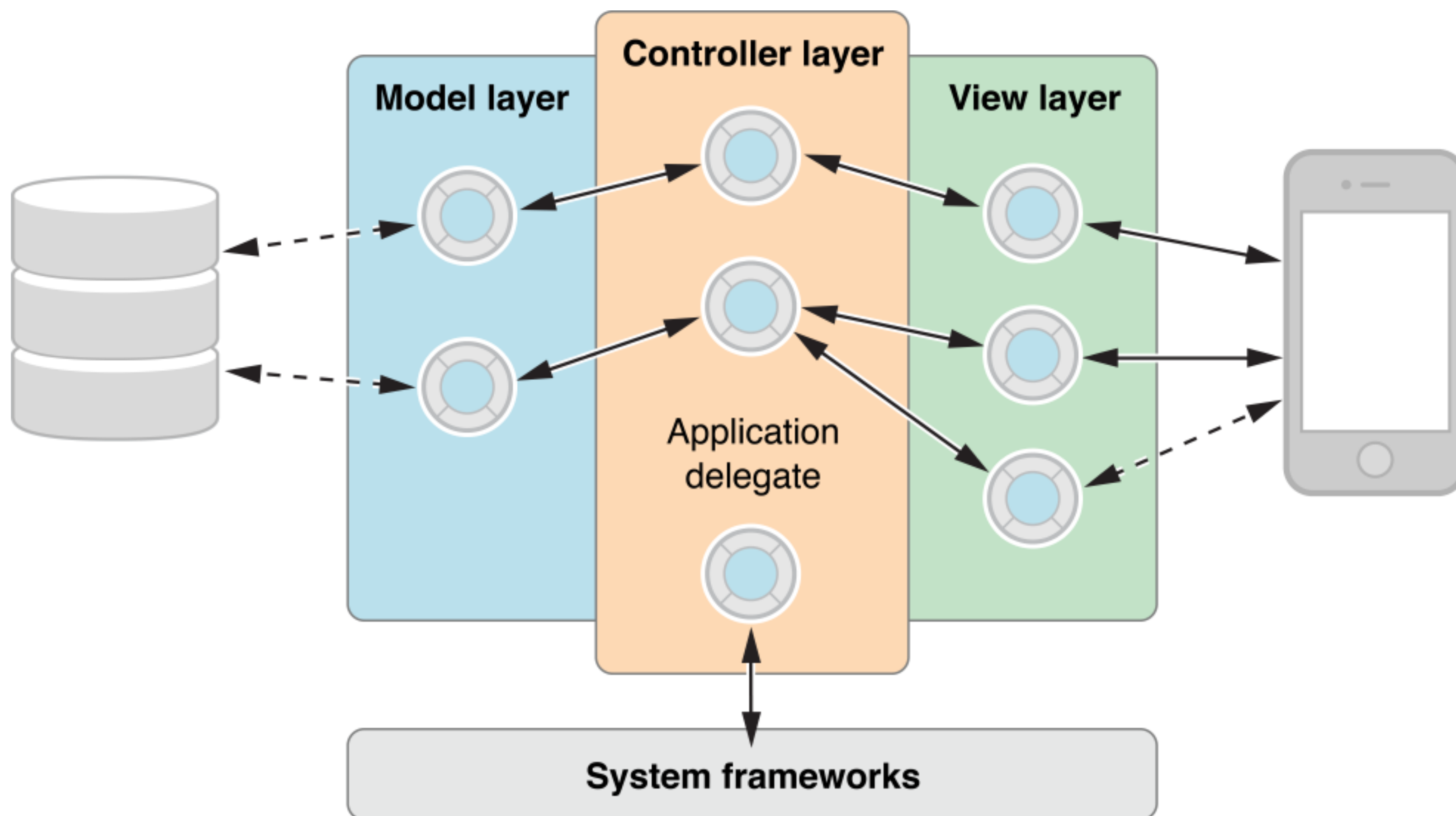
appropos
academy

Collection Types

Array	
Indexes	**Values**
0	Six Eggs
1	Milk
2	Flour
3	Baking Powder
4	Bananas

Set

Values

Rock

Jazz

Classical

Hip Hop

Dictionary

Keys

Values

YYZ → Toronto Pearson

DUB → London Heathrow

LHR → Dublin Airport

appropos **academy**

Apps

Model View Controller

Launch Sequence

iOS App Launch Sequence

Default project template in Xcode 4.2
(without Storyboarding)

```
main()
```

1 calls

```
UIApplicationMain()
```

creates 2

```
UIApplication
```

5 creates and manages

RunLoop

delegate

4 loads

creates 3

```
AppDelegate
```

6 sends

Info.plist

```
application:didFinishLaunchingWithOptions:
```

7 creates and displays

```
Application Window
```

```
Root View Controller
```
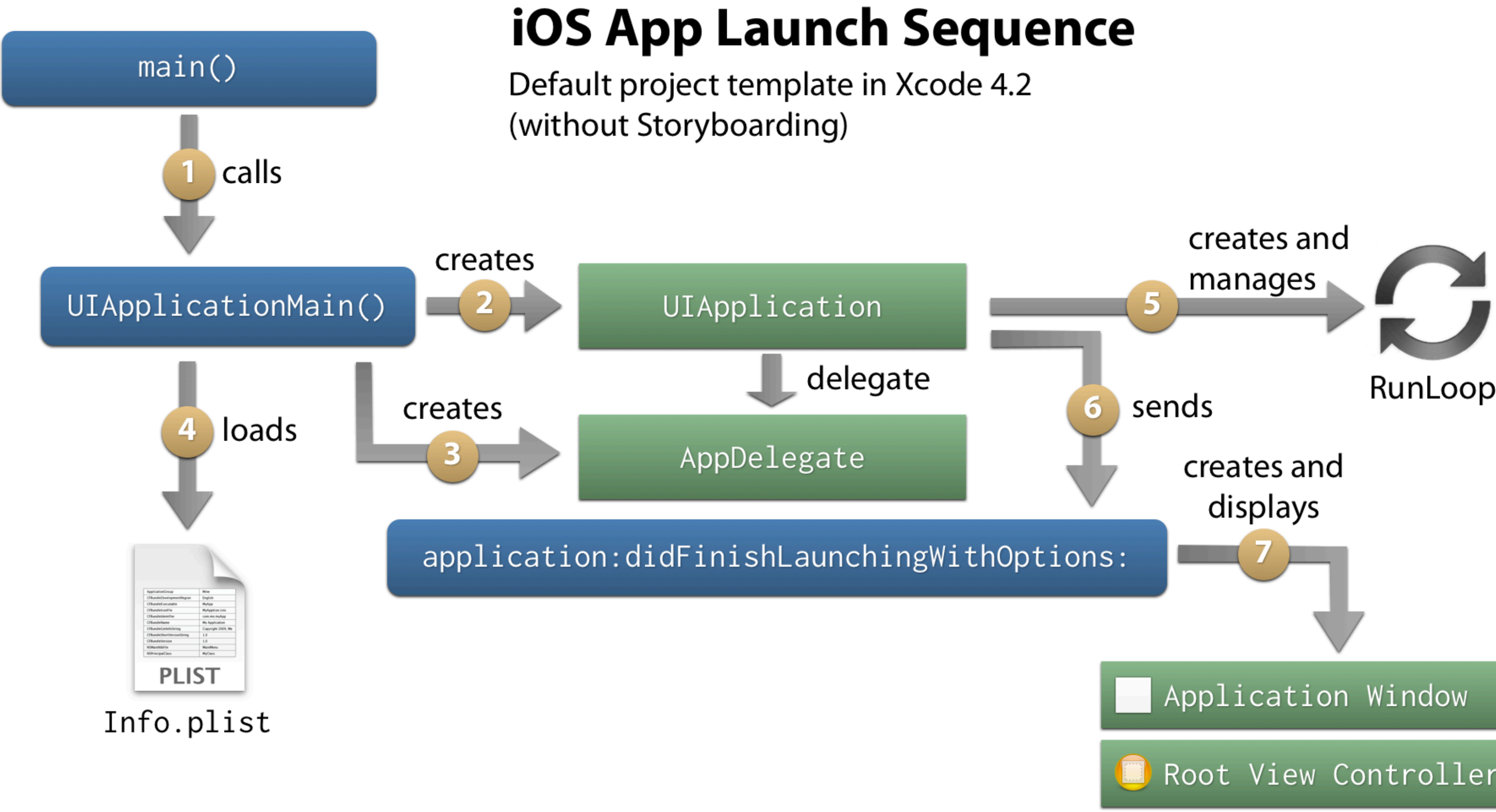
Ole Begemann / oleb.net

appropos
academy

App Lifecycle

Launch Screen UI   App UI

Not Running → Inactive → Active

Suspended ⇄ Background ⇄ Inactive

appropos
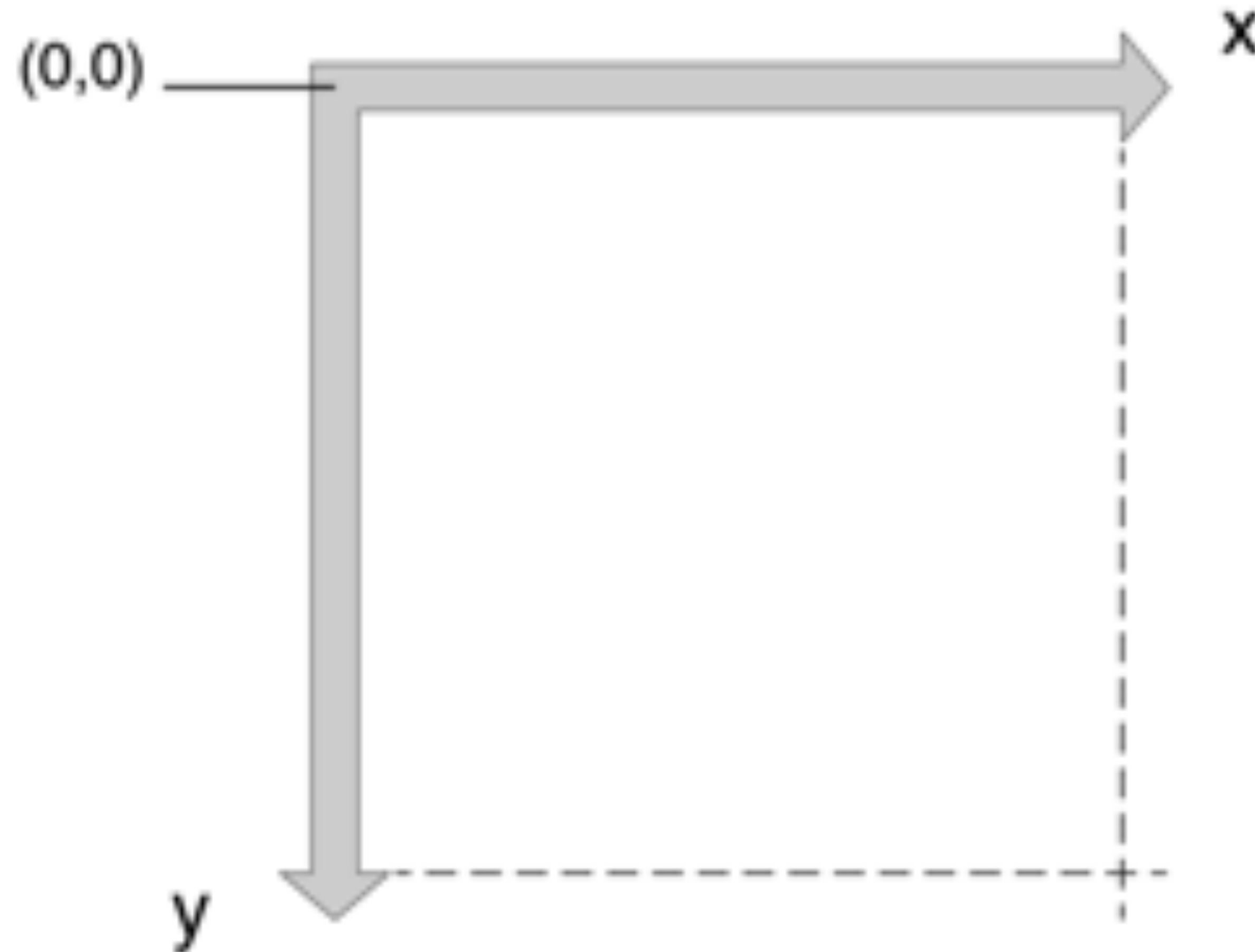academy
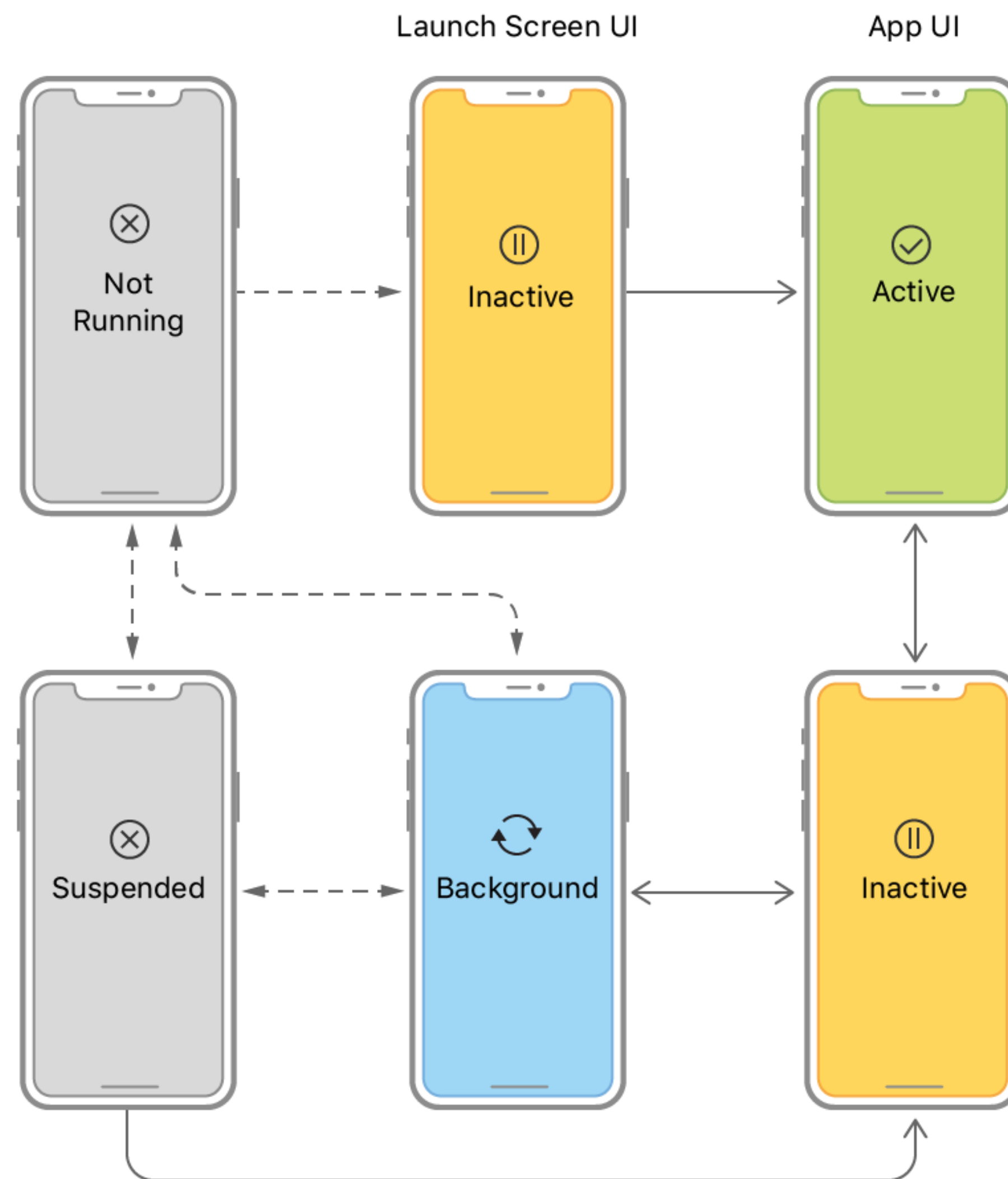
The UI Coordinate System

Let's code!

a very simple game

Exercise 1

- UIViewController with button and label

- Tapping the button appends a "!" to the label text

Exercise 2

- UIViewController with 3 **game buttons** and a start button

- Start button marks a random **game button** (by changing its text)

- Tapping a **game button** unmarks that button

Hint: Int.random(…)

appropos
academy

Exercise 3

- UITableViewController with 5+ **game cells**

- Tapping a **game cells** marks it (by changing its text)

- Tapping it again unmarks it

appropos
academy

Exercise 4

- Use a custom subclass of UITableViewCell for the **game cells**

- Tapping can't mark **game cells**, only unmark them

Exercise 5

- Repeatedly mark random **game cells**

Hint: Timer.scheduledTimer(...)

Exercise 6

- Random marking accelerates with every tap

- When all **game cells** are marked, a UIAlertController pops up

Hint: shorten interval by 5-10% per tap

Bonus Exercise

- **Game cells** show an image which changes when marked

- Adjust Launch Screen and App Icon

- Save Highscores

- Show Elapsed Time

appropos
academy

Dispatch Queues

Queues to which your application can submit executable blocks

- **Main queue**: highest priority, handles UI updates and events

- **Global queues**: universal queues shared by the whole system

  - 4 priority levels: high, default, low, and background

Concurrency Programming Guide https://developer.apple.com/library/archive/documentation/General/Conceptual/ConcurrencyProgrammingGuide/Introduction/Introduction.html

appropos
academy

Dispatch Queues

Blocks can be scheduled in two ways:

- **Synchronously:** blocks the current thread until the block was executed

- **Asynchronously:** continues on the current thread and "forgets" about the block

Synchronously executing a block on the same queue creates a **deadlock**!

Synchronous scheduling from the main queue will **freeze the UI**!

appropos
academy

Package Management

CocoaPods	Carthage	Swift Package Manager
old stable default option deeply integrated complex configuration	newer lightweight more manual steps not always supported repository = package	newest official well integrated **doesn't support iOS yet**

List of selected libraries: https://github.com/matteocrippa/awesome-swift

CocoaPods

- install CocoaPods (of not done yet)

- **pod init** (in project directory)

- list dependencies in the *Podfile* (pod '...')

- **pod install** (in project directory)

- always open the Workspace (.xcworkspace)

The iOS Platform

Hardware

Connectivity	Sensors	Other
• GSM/HSPA/LTE • WLAN • Bluetooth (BLE) • GPS • NFC*	• Face ID / Touch ID • Barometer • Three-axis gyro • Accelerometer • Proximity sensor • Ambient light sensor	• Camera • Speakers • 3D Touch

Software

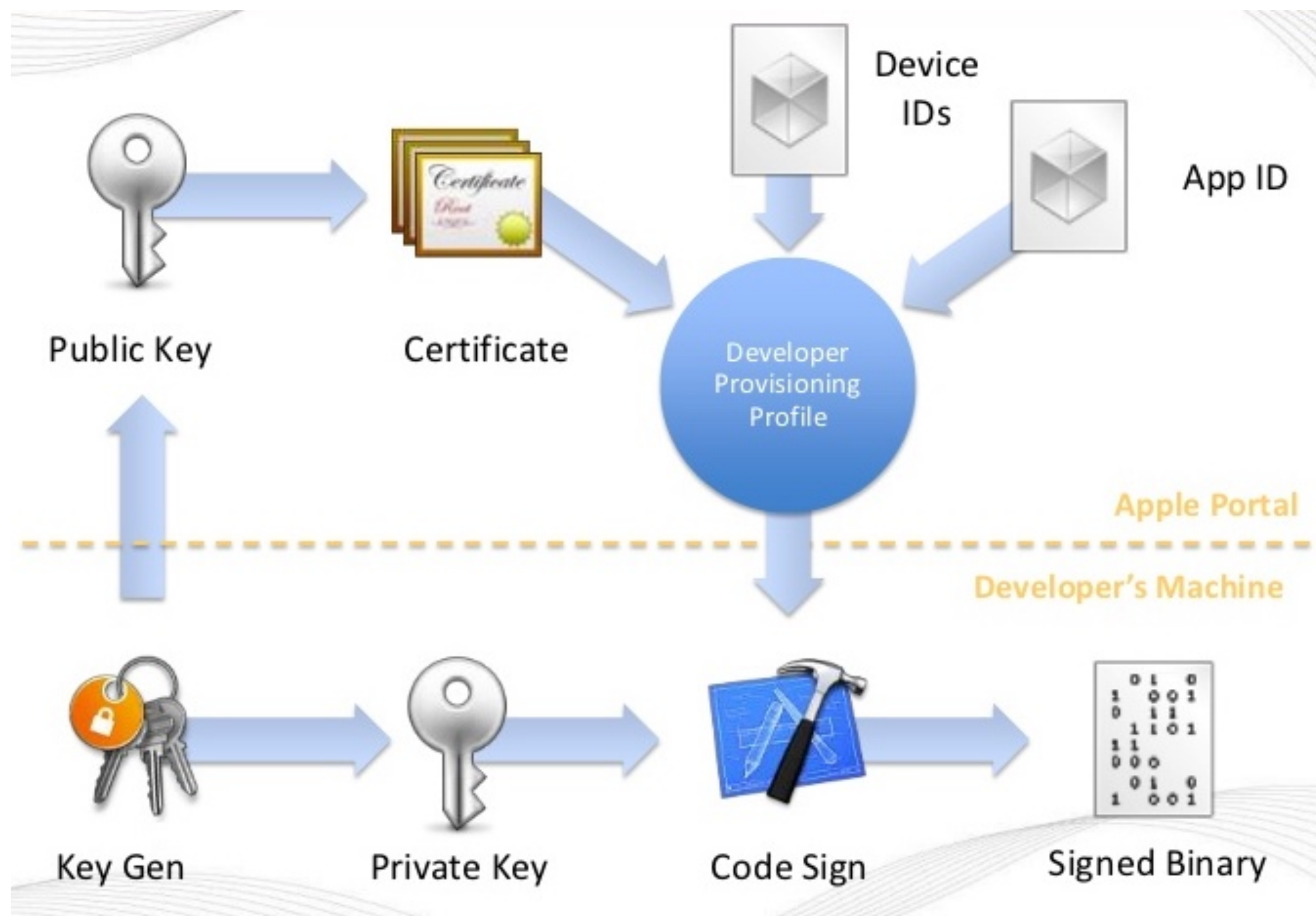| Content	Graphics	Data
• WebKit • MapKit • ARKit • iAd	• Metal • OpenGL • SceneKit • SpriteKit	• CloudKit • EventKit • HealthKit • Core Data
Devices	**Other**	
• CoreBluetooth • iBeacon • WatchKit • HomeKit	• SiriKit • PassKit • Game Center • Core ML	• Core Motion • UserNotifications • AirPlay • StoreKit

appropos
academy

Distribution

Distribution Process

- Enroll into Apple Developer Program

- Create App Store listing (Bundle ID, Texts, Images, ...)

- Submit release build

- Review by Apple (automated + manual, up to 5 days)

- Approval, rejection or request for changes

- Publishing (up to 1 day until visible everywhere)

appropos
academy

Code Signing

Xcode Help: What is app signing? https://help.apple.com/xcode/mac/current/#/dev3a05256b8

Developer Account Help https://help.apple.com/developer-account/

Pricing

Apple Developer Program (prices per year)

Apple ID (0€) / Individual (99€) / Organization (99€) / Enterprise (299€)

https://developer.apple.com/support/compare-memberships/

Business Models (fees per transaction)

Free / Freemium (30%) / Paid (30%) / Subscription (15-30%)

https://developer.apple.com/app-store/business-models/

Feedback

➜ Theory

➜ Practice

➜ Support Material