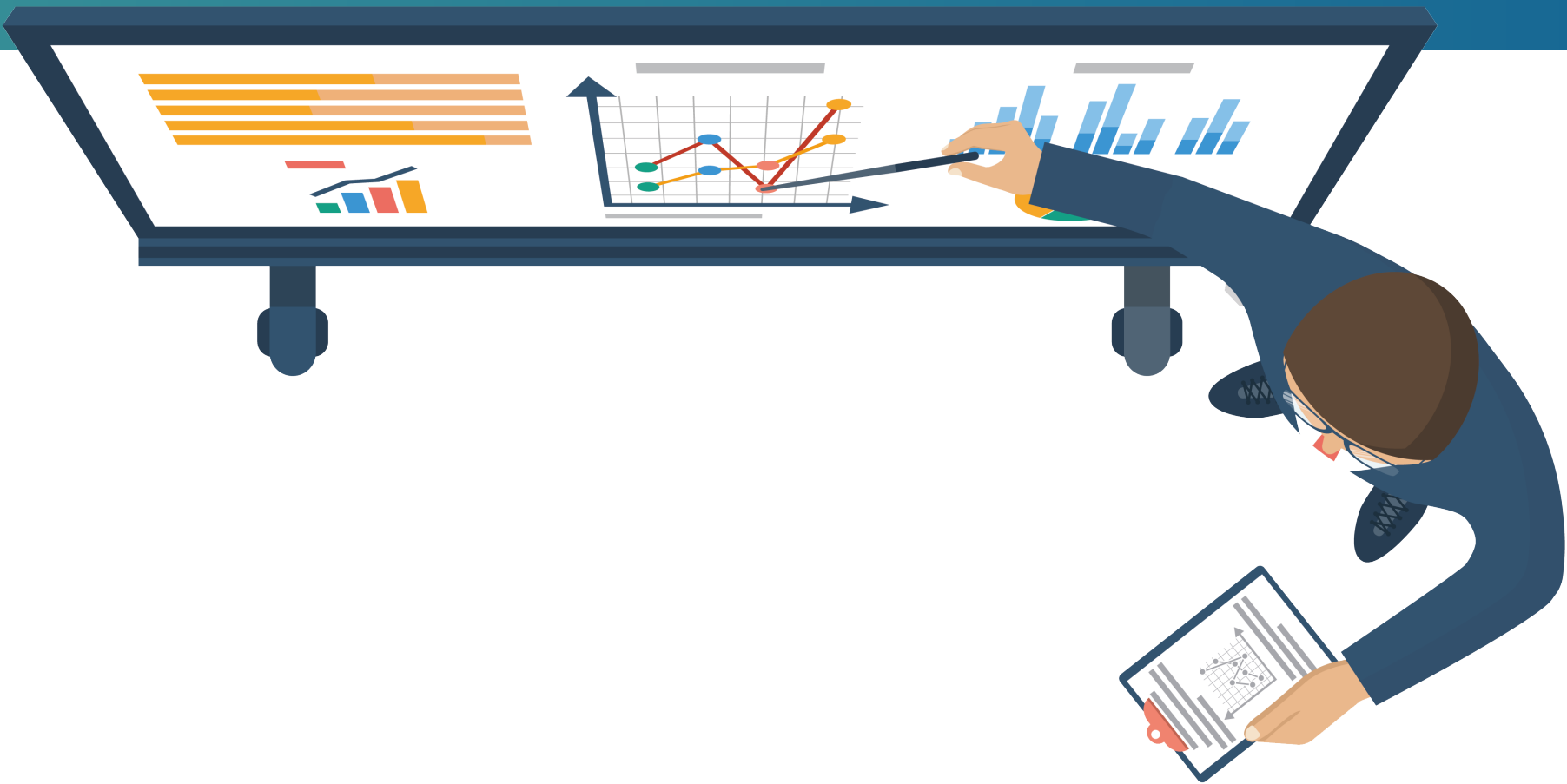


appropos
academy





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 + Lists	Dependencies + Wishlist
13:15 - 14:45	Swift II	Assets + Scheduling	Testing + Platform
15:00 - 16:30	Apps + Xcode	Finishing the Game	Distribution + Feedback



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



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

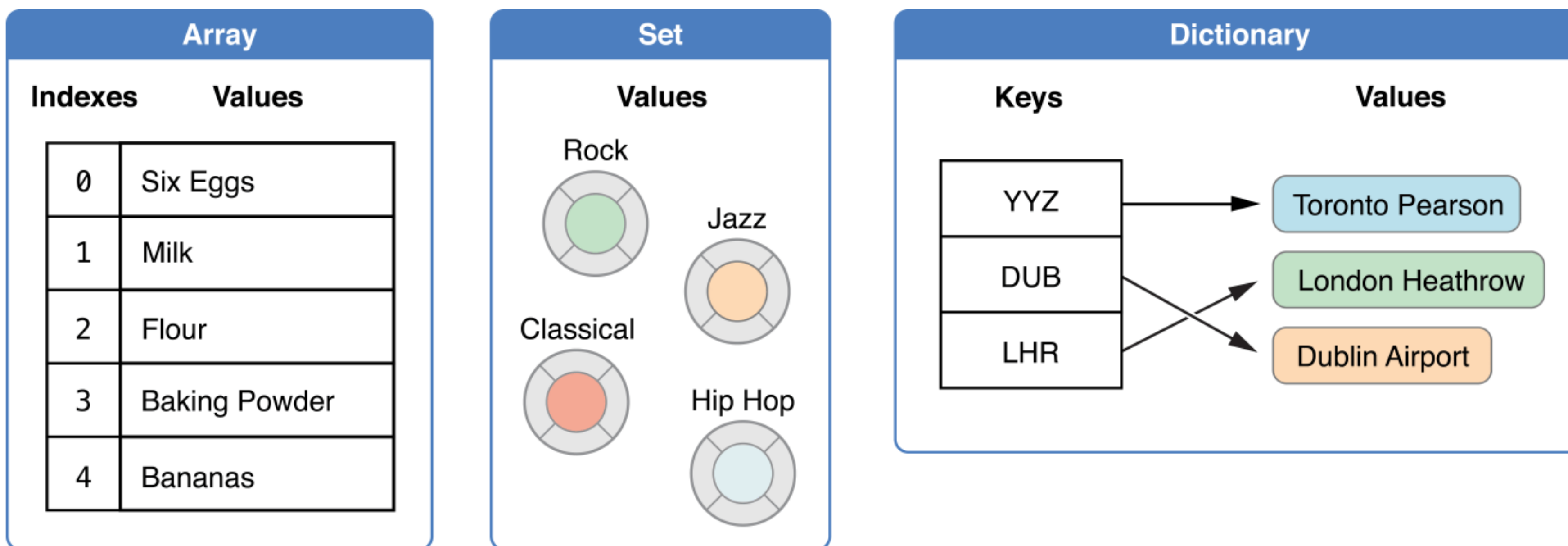
Swift defines away large classes of common programming errors by adopting modern programming patterns:

- *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/>



Collection Types

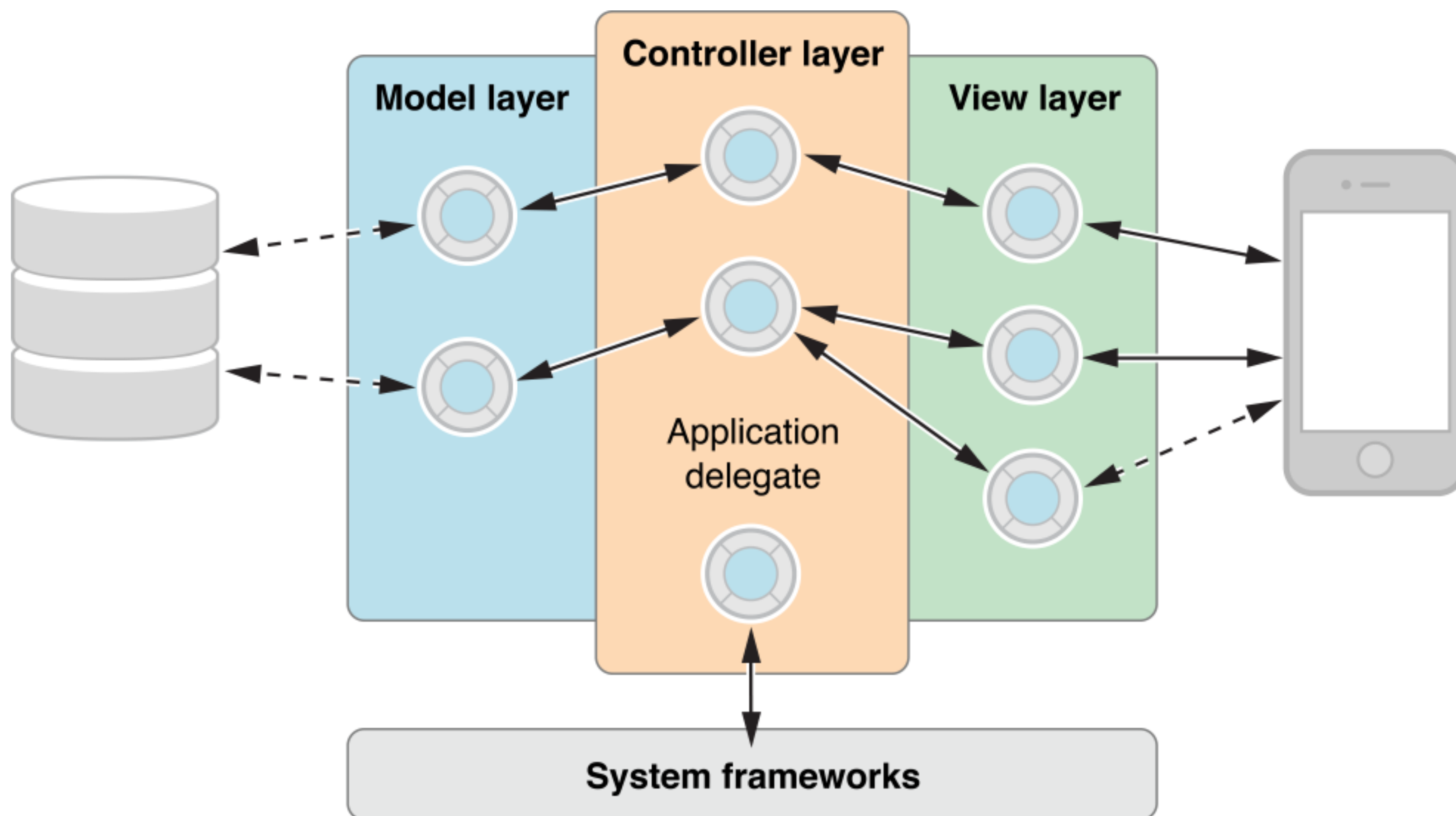




Apps

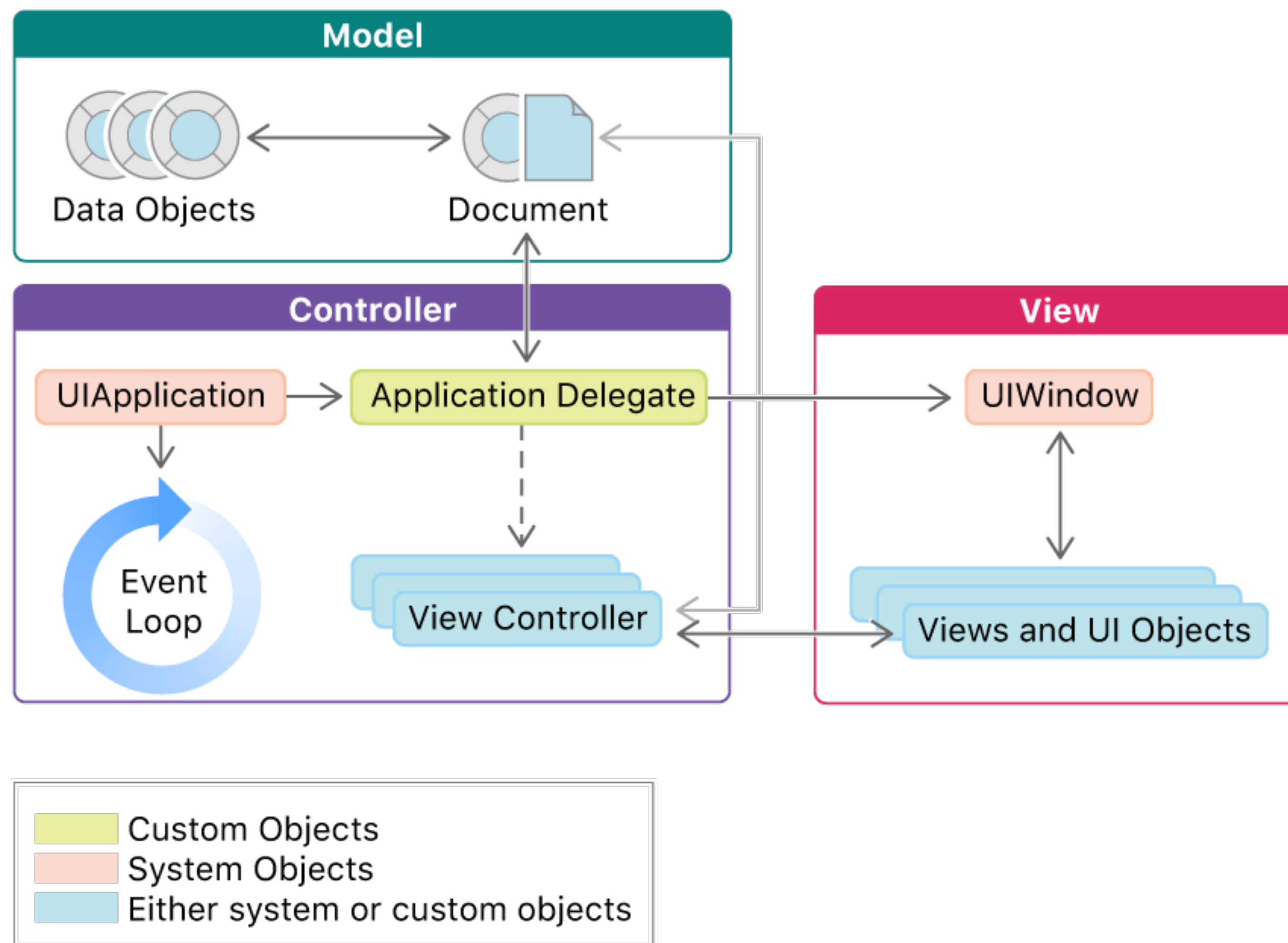


Model View Controller



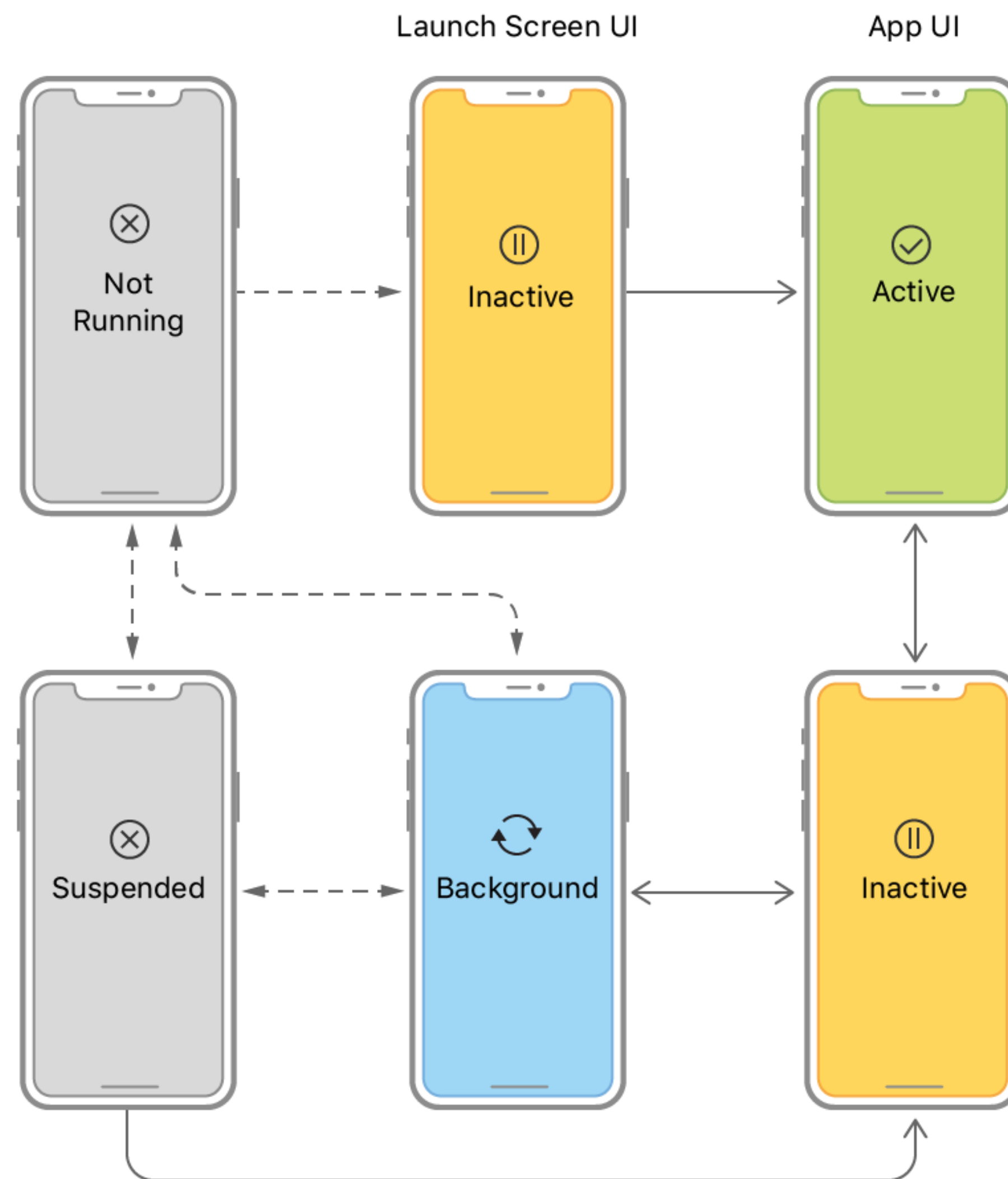


App Architektur



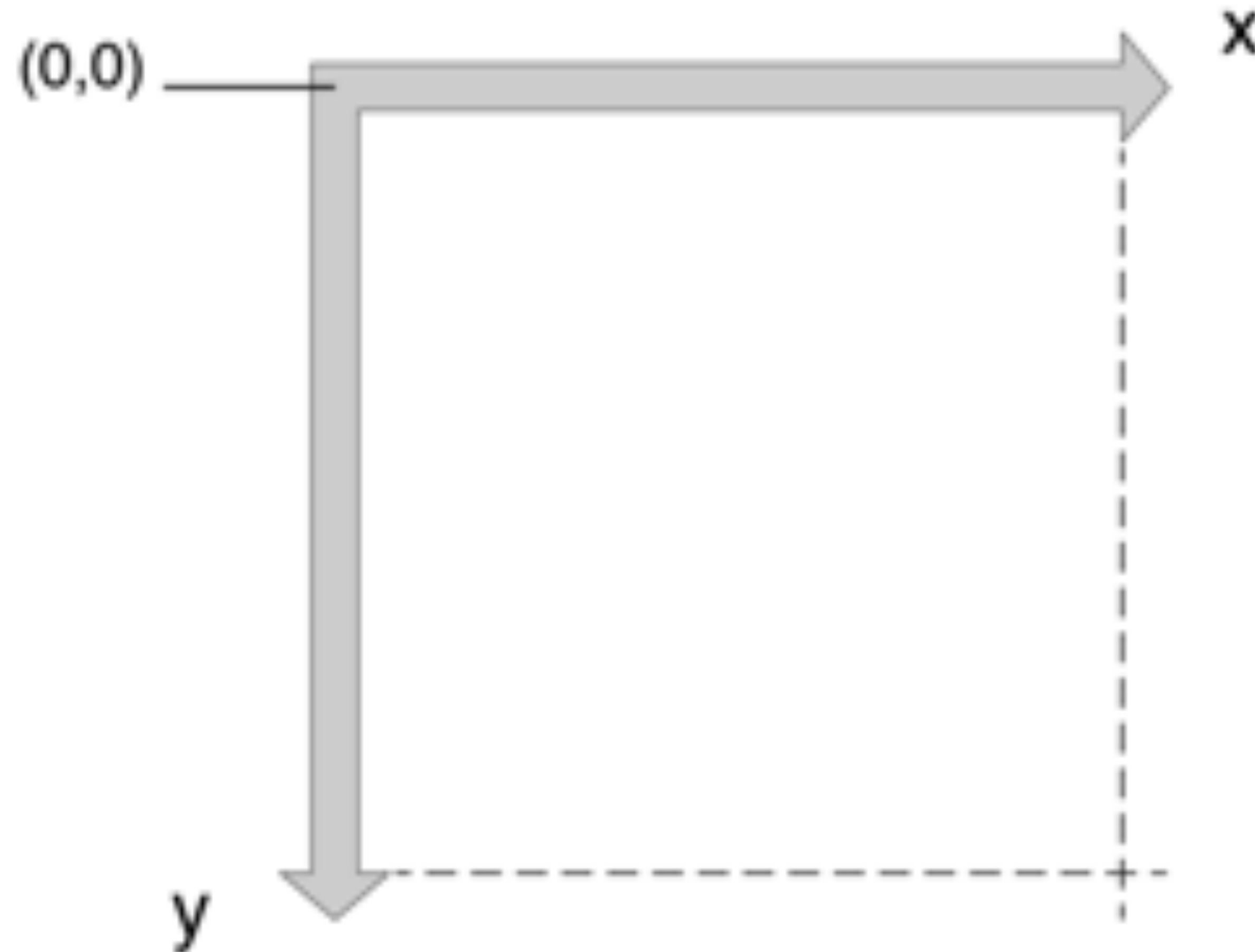


App Lifecycle





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(...)`



Exercise 3

- UITableViewController with 5+ **game cells**
- Tapping a **game cells** marks it (by changing its text)
- Tapping it again unmarks it



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



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!**



Exercise 4

- Tapping can't make **game cells**, only unmark them
- **Game cells** are gettings marked periodically and randomly

Hint: `asyncAfter(...)`



Exercise 5

- Random marking accelerates with every tap
- When all **game cells** are marked, a UIAlertController pops up
- Bonus: Counter, Highscores, App Icon, Launch Screen, colors ...

Hint: cut interval repeatedly by 5-10%



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>



The iOS Platform



Hardware

Connectivity	Sensors	Other
<ul style="list-style-type: none">• GSM/HSPA/LTE• WLAN• Bluetooth (BLE)• GPS• NFC*	<ul style="list-style-type: none">• Face ID / Touch ID• Barometer• Three-axis gyro• Accelerometer• Proximity sensor• Ambient light sensor	<ul style="list-style-type: none">• Camera• Speakers• 3D Touch



Software

Content	Graphics	Data
<ul style="list-style-type: none">• WebKit• MapKit• ARKit• iAd	<ul style="list-style-type: none">• Metal• OpenGL• SceneKit• SpriteKit	<ul style="list-style-type: none">• CloudKit• EventKit• HealthKit• Core Data
Devices	Other	
<ul style="list-style-type: none">• CoreBluetooth• iBeacon• WatchKit• HomeKit	<ul style="list-style-type: none">• SiriKit• PassKit• Game Center• Core ML	<ul style="list-style-type: none">• Core Motion• UserNotifications• AirPlay• StoreKit



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)



Pricing

Apple Developer Program (prices per year)

Apple ID (0\$/y) / Individual (99\$/y) / Organization (99\$/y) / Enterprise (299\$/y)

<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