

appropos
academy





iOS Grundlagen mit Swift





Vorstellungsrunde

→ Name

→ Erfahrung

→ Erwartungen



Zeitplan

	Tag 1	Tag 2	Tag 3
9:00 - 10:30	Vorstellung + Setup	Storyboards	Networking
10:45 - 12:15	Swift I	Navigation + Listen	Abhängigkeiten + Wunschliste
13:15 - 14:45	Swift II	Assets + Scheduling	Testen + Plattform
15:00 - 16:30	Apps + Xcode	Spiel fertigstellen	Veröffentlichung + Feedback



GitHub Repository:

github.com/ernesto-elsaesser/ios-training



Los geht's!

Gerne jederzeit ...

- ... unterbrechen
- ... Fragen stellen
- ... an Pausen erinnern



iOS

Was ist anders?



Einschränkungen

- Bildschirm
- Akku
- Mobile Daten
- Speicher
- Eingabe



Erwartungen

- Stabilität
- Schnelligkeit
- Design
- Responsiveness
- Bedienbarkeit (HIG)

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



Swift



Eine neue Sprache

*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>



Eine sichere Sprache

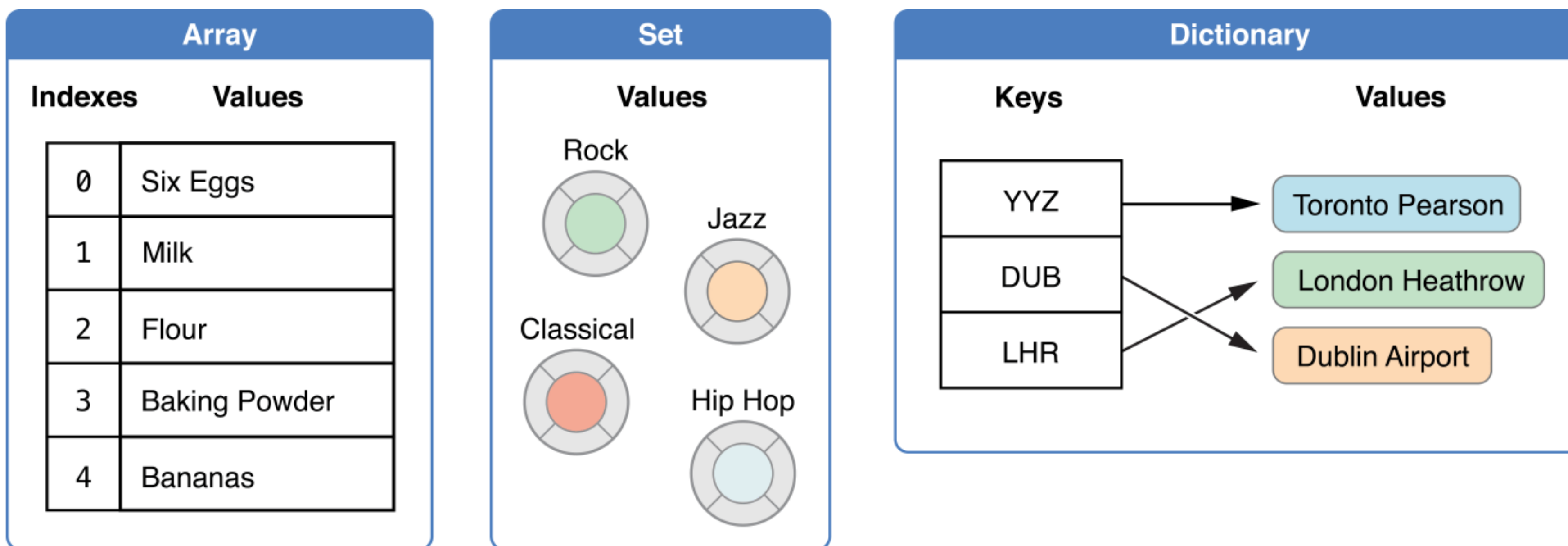
Swift defines away large classes of common programming errors by adopting modern programming patterns:

- *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-Typen

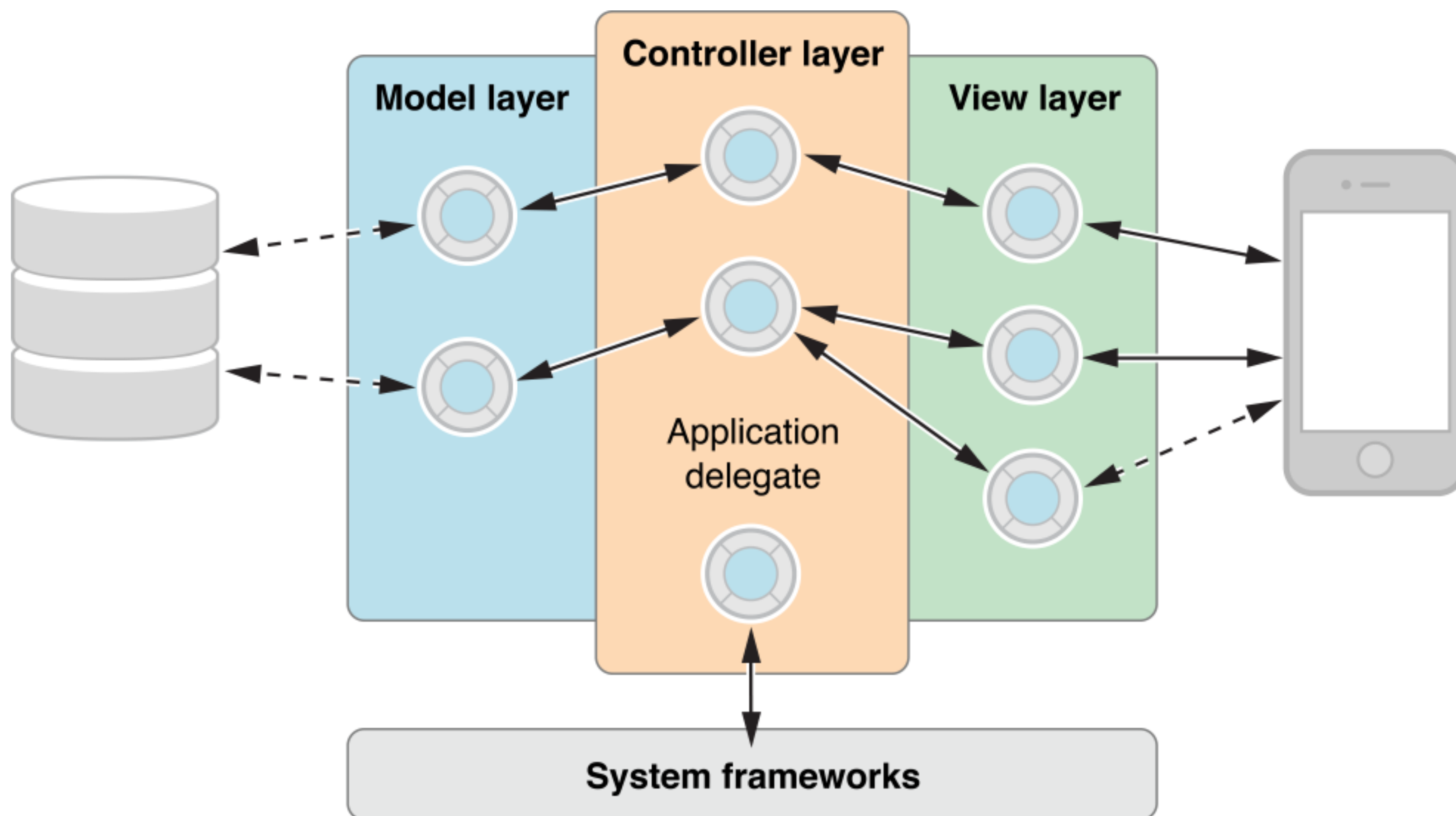




Apps

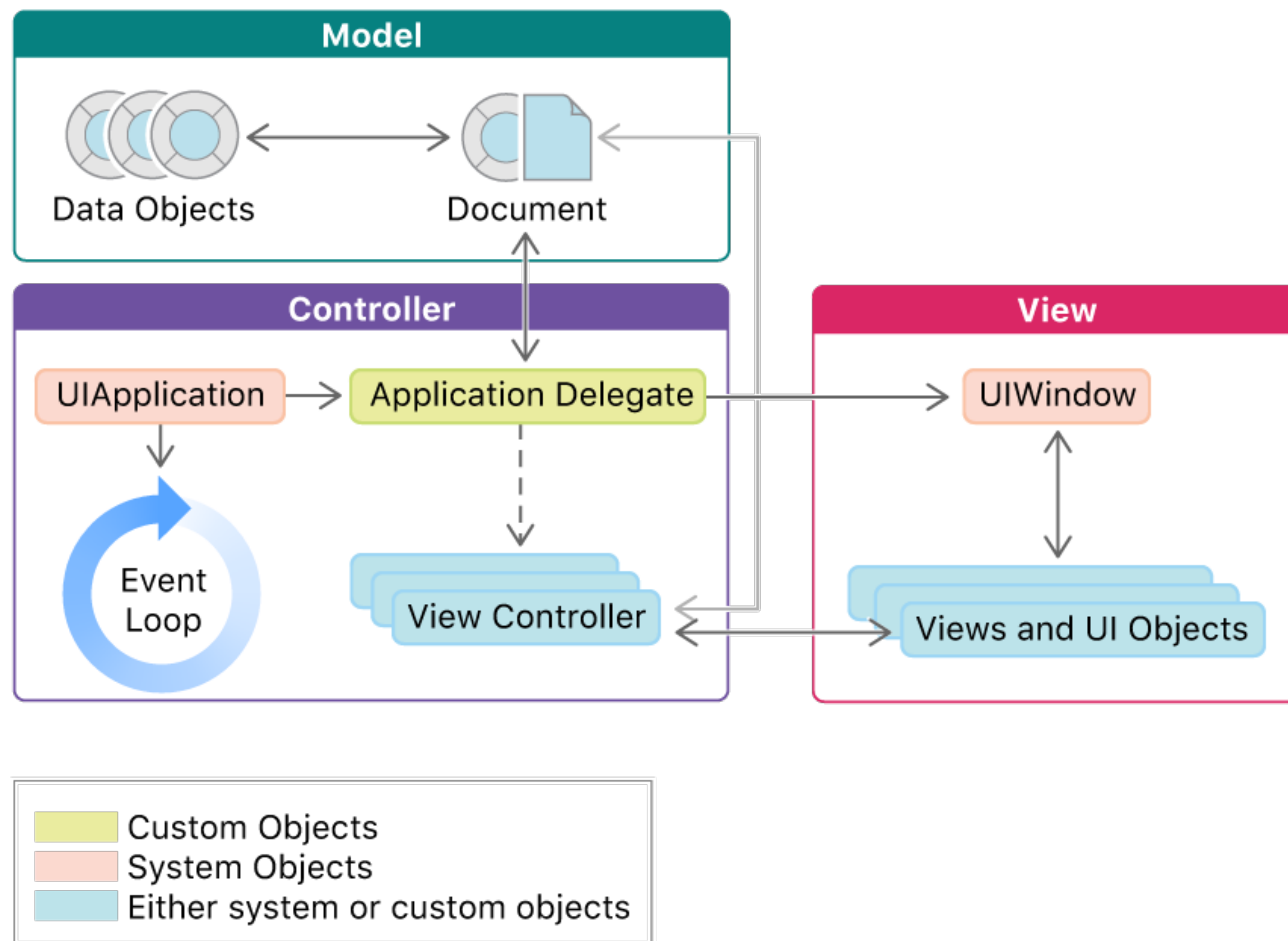


Model View Controller



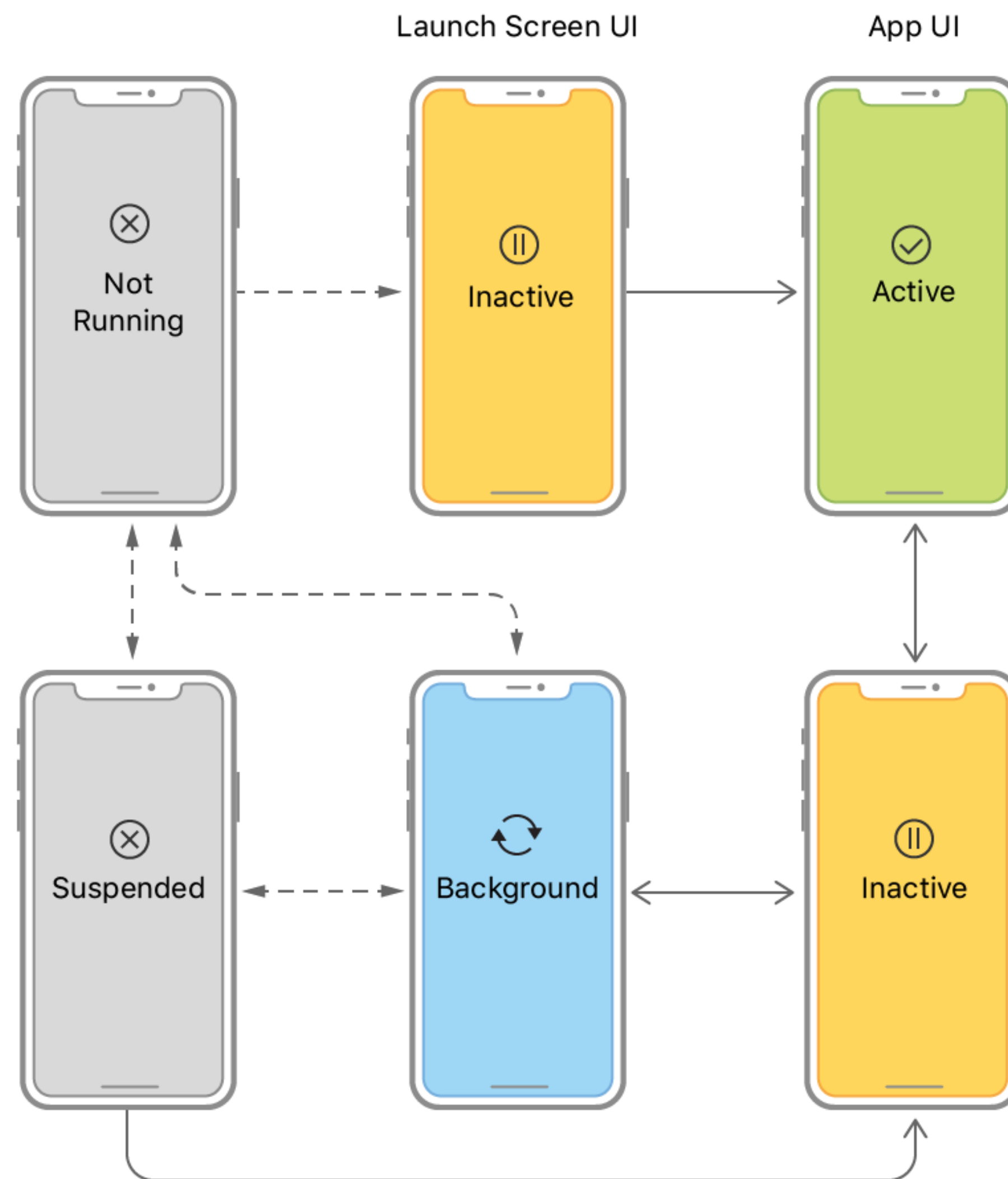


App Architektur



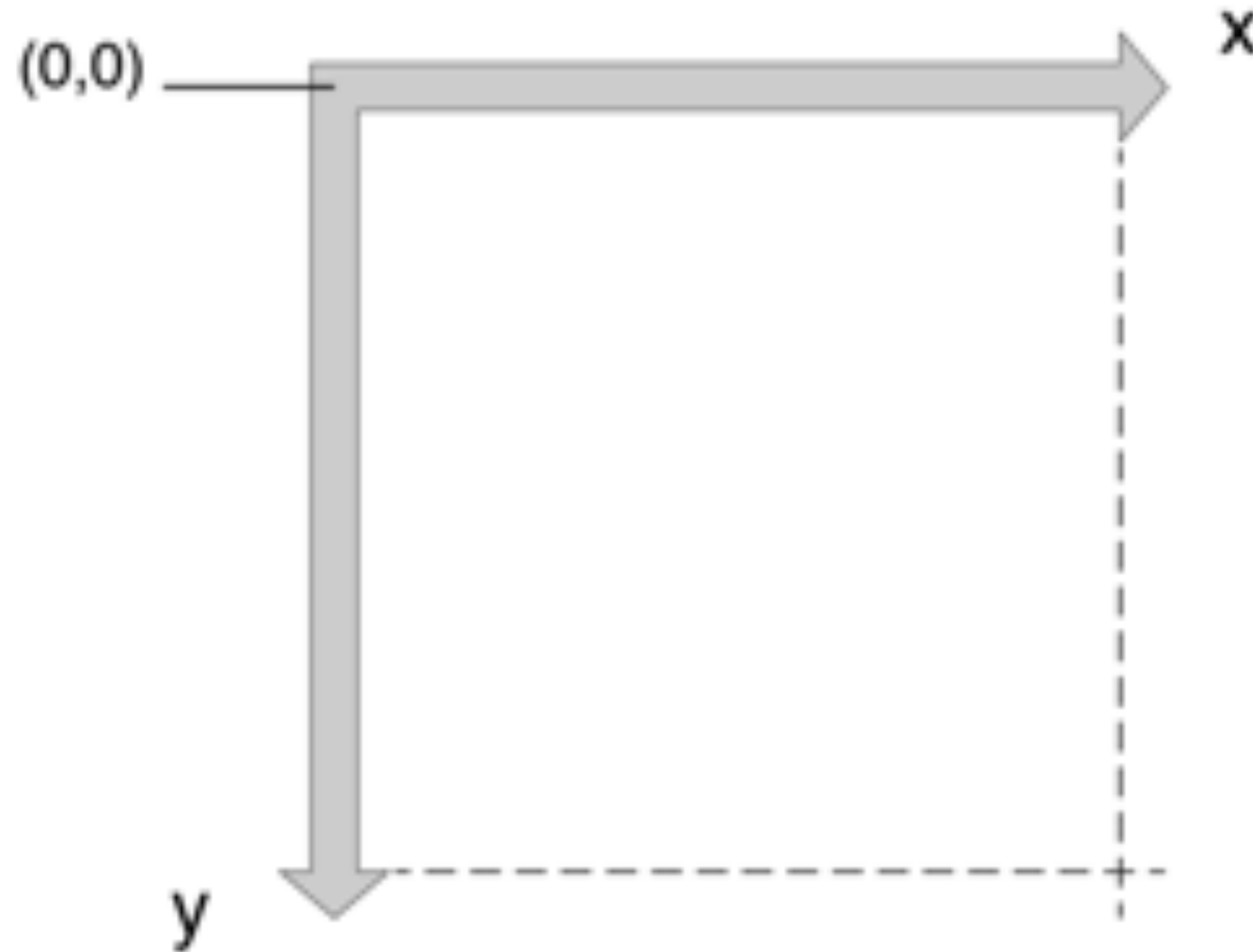


App Lifecycle





Das UI Koordinatensystem





Jetzt dürft ihr!

ein kleines Spiel programmieren



Aufgabe 1

- UIViewController mit Knopf und Label
- Drücken des Knopfes hängt ein “!” an den Labeltext an



Aufgabe 2

- UIViewController mit 3 **Spielknöpfen** und einem Startknopf
- Startknopf markiert einen zufälligen **Spielknopf** (Änderung des Textes)
- Druck auf einen **Spielknopf** entfernt dessen Markierung

Hinweis: `Int.random(...)`



Aufgabe 3

- UITableViewController mit 5+ **Spielzellen**
- Durch anklicken wird eine **Spielzelle** markiert (Änderung des Textes)
- Durch weiteren Klick wird die Markierung wieder entfernt



Dispatch Queues

Warteschlangen, in welche Codeblöcke eingereiht werden können

- **Main Queue:** höchste Priorität, genutzt für Benutzeroberfläche
- **Global Queues:** universelle systemweite Warteschlangen
 - 4 Prioritätsstufen: hoch, normal, niedrig und Hintergrund



Dispatch Queues

Blöcke können auf zwei Arten eingereiht werden:

- **Synchron:** blockiert den aktuellen Thread bis der Block ausgeführt wurde
- **Asynchron:** setzt den aktuellen Thread fort und "vergisst" den eingereihten Block

Wird ein Block synchron in die selbe Queue eingereiht entsteht ein **Deadlock!**

Synchrone Aufrufe von der Main Queue **frieren die Benutzeroberfläche ein!**



Aufgabe 4

- Klick auf **Spielzelle** kann nicht markieren, nur Markierung entfernen
- **Spielzellen** werden periodisch und zufällig markiert

Hinweis: `Timer.scheduledTimer(...)`



Aufgabe 5

- Zufällige Markierung wird nach jedem Klick schneller
- Wenn alle **Spielzellen** markiert sind poppt ein UIAlertController auf
- Bonus: Counter, Highscores, App Icon, Launch Screen, Farben ...

Hint: Zeitintervall kontinuierlich um 5-10% verkürzen



Package Management

CocoaPods	Carthage	Swift Package Manager
alt stabil Standard tiefe Integration komplexe Konfiguration	neuer leichtgewichtig manuelle Schritte nötig nicht immer unterstützt Repository = Paket	am neusten offiziell gut integriert noch keine iOS Unterstützung

Liste ausgewählter Bibliotheken: <https://github.com/matteocrippa/awesome-swift>



CocoaPods

- ggf. CocoaPods installieren
- **pod init** (im Projektordner)
- Abhängigkeiten im *Podfile* auflisten (pod '...')
- **pod install** (im Projektordner)
- Immer im Workspace arbeiten (.xcworkspace)



Die iOS Plattform



Hardware

Connectivity	Sensoren	Andere
<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• 3-Achsen Gyrosensor• Beschleunigungssensor• Näherungssensor• Umgebungslichtsensor	<ul style="list-style-type: none">• Kamera• Lautsprecher• 3D Touch



Software

Inhalte	Grafik	Daten
<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
Geräte	Andere	
<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



Veröffentlichung

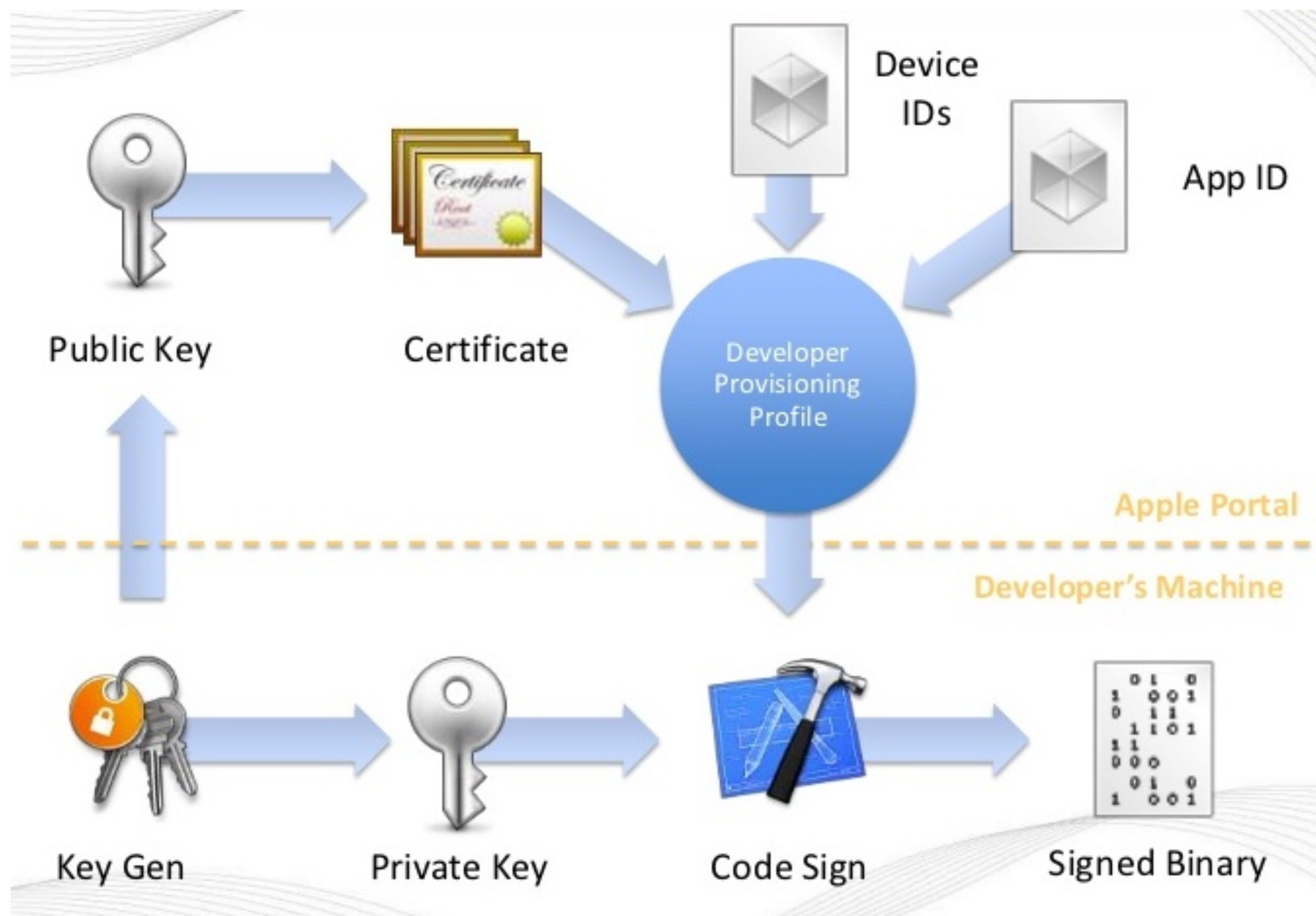


Veröffentlichungsprozess

- Registrieren im Apple Developer Program
- Erstellen des App Store Eintrages (Bundle ID, Texte, Bilder, ...)
- Release Build hochladen
- Review durch Apple (automatisiert + manuell, bis zu 5 Tage)
- Freigabe, Ablehnung oder Änderungsanfrage
- Veröffentlichung (weltweit sichtbar nach ~1 Tag)



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/>



Kosten

Apple Developer Program (Preise pro Jahr)

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

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

Geschäftsmodelle (Gebühren per Transaktion)

Kostenlos / Freemium (30%) / Beahlt (30%) / Abonnement (15-30%)

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



Feedback

→ Theorie

→ Praxis

→ Begleitmaterial