App-Lifecycle, View Controllers Storyboards

Das iOS besteht aus folgenden **iOS-Schichten**:

- **Cocoa Touch**: Datenstrukturen, Dateimanagement, Netzoperationen, Visuelle Infrastruktur ObjC & Swift
- **Media**: 2/3D zeichnen, Audio, Video, ...C & ObjC
- **Core Services**: Elementare Dienste wie, Dateizugriff, elem. Datentypen, Sockets
- **Core OS**: Elementare Dienste wie, Dateizugriff, elem. Datentypen, Sockets

App-Zustände

Die App-Zustände sind:

- **not running**: Nicht gestartet oder terminiert
- **inactive**: Läuft im Vordergrund (ohne Events), meist kurz während Übergang
- **active**: Läuft im Vordergrund
- **background**: Läuft im Hintergrund (temp)
- **suspended**: Background, kein Code ausgeführt

In Swift gibt es statt eine main.m ein neues Attribut:
`@UIApplicationMain class AppDelegate: UIResponder,` → Implementation nicht ändern!

Wichtige Funktionen von Klasse **AppDelegate**: Monitoring Application State Changes,

- `application:didFinishLaunchingWithOptions:`
- `applicationDidBecomeActive:`
- `applicationWillResignActive:`
- `applicationDidEnterBackground:`
- `applicationWillEnterForeground:`
- `applicationWillTerminate:`
- `applicationDidFinishLaunching:`

iOS kann "Suspended" App jederzeit terminieren (z.B. bei Speicherknappheit) → Benutzer soll nicht merken ob suspended oder terminiert

Background Tasks

In iOS folgende Tasks im Hintergrund erlaubt:

- Audio Player / Recorder
- Location
- Voice over Internet Protocol
- Newsstand Apps
- App that receive regular updates from external accessories (Bluetooth)
- Fetch

Dazu muss App Rechte beantragen.

Info.plist (Information Property List File): Structured text file that contains essential configuration information for a bundled executable (App-Name, Exec-Datei-Name, Icon-Dateinamen, Bundle ID, Bundle-Name, Bundle-Version)

View Controller

View controllers are the **foundation of your app's internal structure** (most apps have several) → manages a portion of app's user interface and interaction between that interface + underlying data

MVC Design-Muster:

- Model: "Daten"
- View: "Ansicht"
- Controller: "Vermittler"

UIViewController: Controller einer Bildschirmseite in iOS (Basisklasse für alle iOS-ViewController)

Properties von UIViewController:

- **nib-Name**: für Interface-Builder Dateien
- **view**: View, welcher dieser Controller steuert
- **title**: wichtig bei Navigation (Default für Titel NavigationBar & Back-Button
- Für **ModalViews**: modalViewController, modalTransitionStyle
- Für **Navigation & Tabs**: navigationController/-Item, etc.

Methoden von UIViewController:

- `initWithNibName:bundle` : Initialisierung aus Nib-Datei (meist automatisch)
- `loadView` : Wird aufgerufen, wenn eine "view" angewählt wird, danach wird `viewDidLoad` aufgerufen
- `viewWillAppear` :/ `viewDidLoad` :
- `viewWillDisappear` :/ `viewDidDisappear` :
- `didReceiveMemoryWarning` : Speicherwarnung

Für das Laden einer "view" in das Memory hat die `loadView` -Funktion 3 Optionen:

1. Wenn es eine nib-Datei hat, dann wird aus dieser eine View erstellt.
2. Man kann auch die View manuell implementieren (`UIView`) im Source Code
3. Wenn es beide Datei nicht hat, dann wird eine leere UIView erstellt

→ Für Apple sind nib-Dateien und .storyboard mittlerweile praktisch äquivalent

Zwei Arten von View Controller,

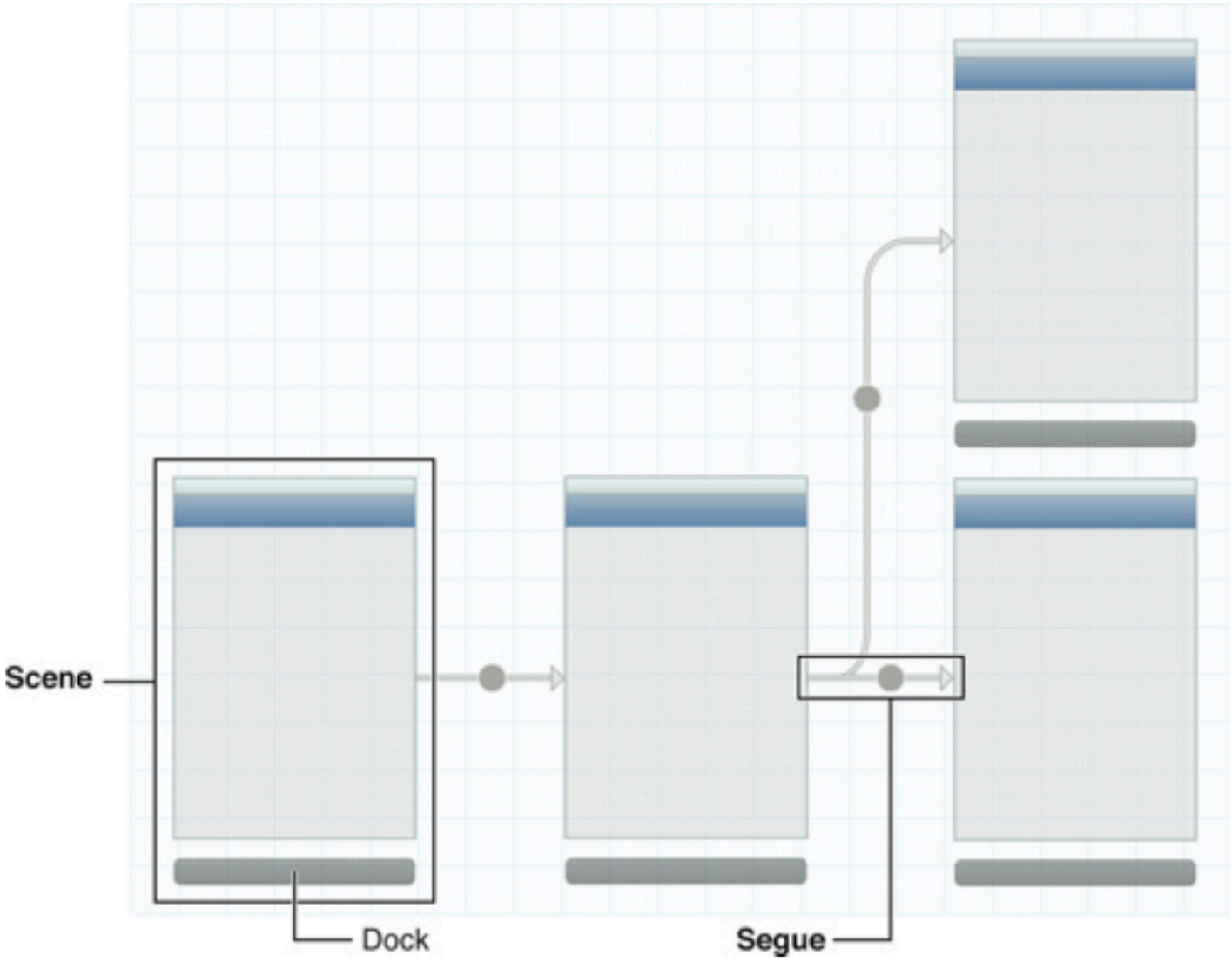
- **Content View Controller**: Stellen Inhalt dar
- **Container View Controller**: Arrangieren Inhalt von anderen ViewControllers

ModalViewController stapeln die ViewControllers übereinander. Wenn etwas hierarchisch mit vielen ViewControllers aufgebaut ist, sollte man deshalb den UINavigationController verwenden.

Neu sollte man **show** anstelle von **present** verwenden.

Storyboard

Storyboard = Visual representation of user interface of iOS app, showing screens of content and the connections between those screens.



Hinter jeder Szene steht ein ViewController.

Seque: Übergang von einem ViewController zum nächsten.