Full Stack iOS Entwicklung mit Swift

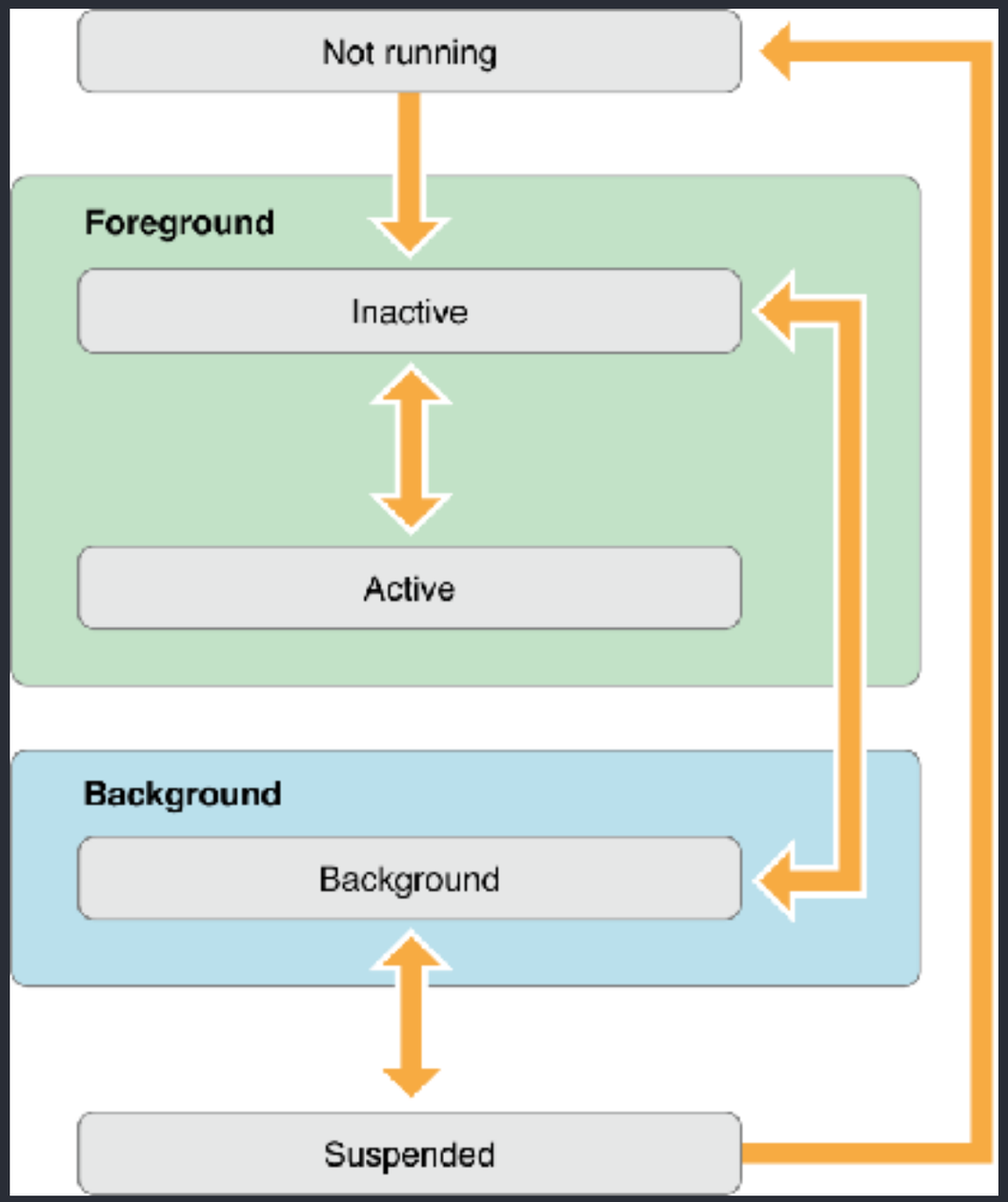
WPF im MIM - SS 17

Alexander Dobrynin M.Sc.

Heute

Application & ViewController Lifecycle
Segues (Detail, Modal, Popover, Unwind, Embedded)

Application Lifecycle



- application:willFinishLaunchingWithOptions: This method is your app's first chance to execute code at launch time.
- application:didFinishLaunchingWithOptions: This method allows you to perform any final initialization before your app is displayed to the user.
- applicationDidBecomeActive: Lets your app know that it is about to become the foreground app. Use this method for any last minute preparation.
- applicationWillResignActive: Lets you know that your app is transitioning away from being the foreground app. Use this method to put your app into a quiescent state.
- applicationDidEnterBackground: Lets you know that your app is now running in the background and may be suspended at any time.
- applicationWillEnterForeground: Lets you know that your app is moving out of the background and back into the foreground, but that it is not yet active.
- applicationWillTerminate: Lets you know that your app is being terminated. This method is not called if your app is suspended.

https://developer.apple.com/library/content/documentation/iPhone/Conceptual/iPhoneOSProgrammingGuide/TheAppLifeCycle/TheAppLifeCycle.html

ViewController Lifecycle

- viewDidLoad: Erst hier sind Preparation und Outlets durchlaufen. Ein guter Platz für Setup-Code
- viewWill/DidAppear: Der VC ist für den Benutzer sichtbar. Hier werden oftmals Animationen oder sonstiger UI bezogener Code gestartet
- viewWill/DidDisappear: Der VC verschwindet. Hier werden häufig die entgegengesetzten Aktionen zu viewWill/DidAppear vorgenommen. Des Weiteren werden laufende Routinen o.ä. aufgeräumt
- Ist der VC einmal geladen, kann er öfters zwischen appear und disappear wechseln. Deshalb sollte man sichergehen, dass hier kein Code ausgeführt wird, der zu viewDidLoad gehört. Geeigneter ist es die Inhalte neu zu laden, Animationen zu starten oder die View mit dem Model zu synchronisieren
- viewWill/DidLayoutSubviews: Die Geometrie des VC hat sich geändert
- viewWillTransition: Der Benutzer rotiert das Gerät
- didReceiveMemoryWarning: Der Speicher wird knapp… hier bekommt man Zeit zum Reagieren
- deinit: Der ViewController wurde erfolgreich dealloziert und liegt nicht mehr im Speicher. Hier kann man mit print's sicherstellen, dass der VC keinen Memory-Leak hat

Segues

- Segues sind Transitionen zwischen zwei ViewController src und dest, wobei src den kompletten MVC-Stack von dest präsentieren möchte
- Der Stil der Präsentation kann über var modalPresentationStyle: UIModalPresentationStyle u.a. auf .fullScreen, .overFullScreen, .popover oder .formSheet gesetzt werden
- Die Animation der Transition kann über var modalTransitionStyle: UIModalTransitionStyle auf .coverVertical, .flipHorizontal, .crossDissolve oder .partialCurl gesetzt werden
- Segues können entweder im Storyboard mit einem ctrl-drag zwischen src- und dest-VC modelliert oder im Code ausgeführt werden

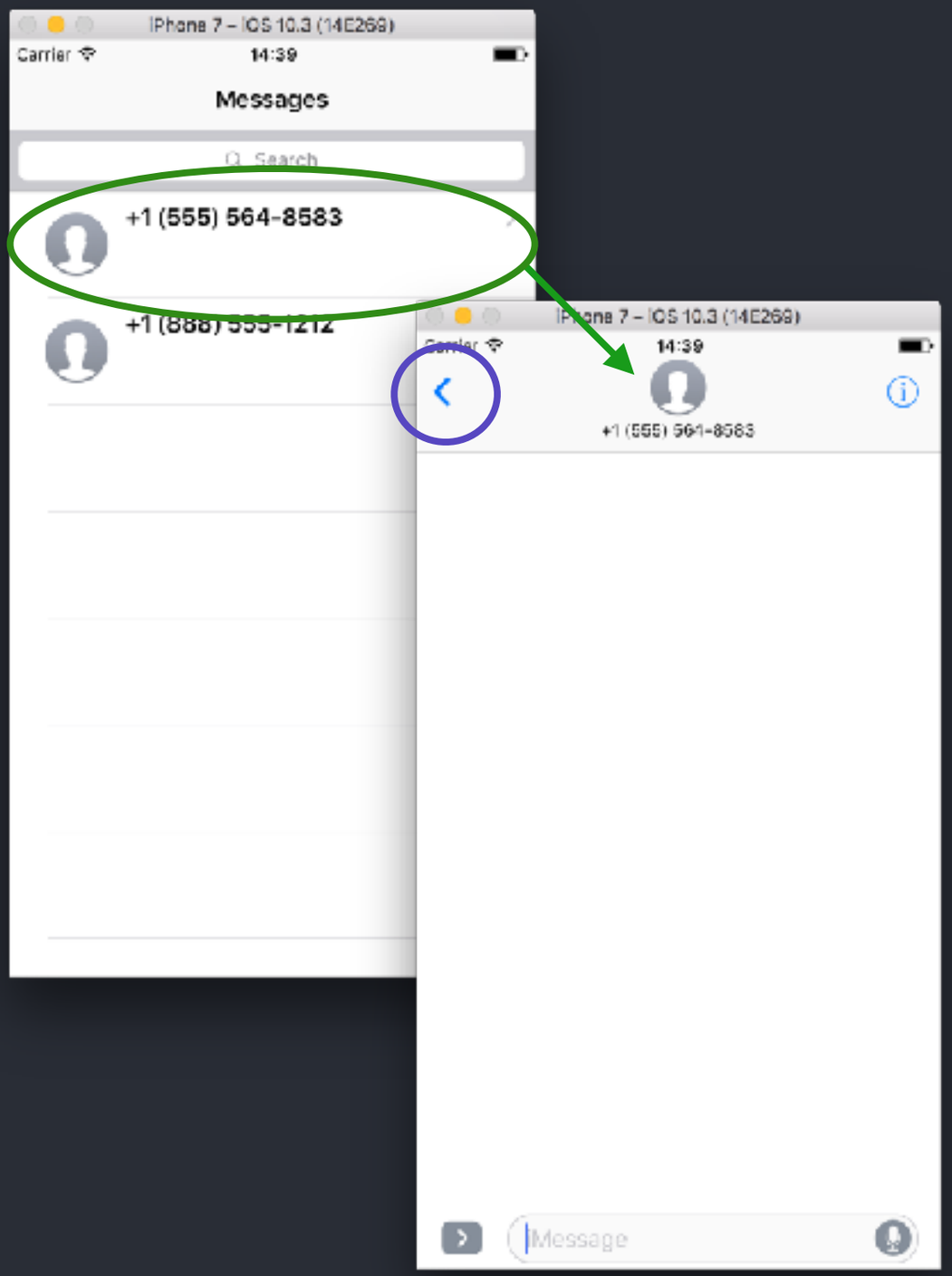
```
self.performSegue(withIdentifier: "Open Settings", sender: self)
```

- Der src-VC kann die Segue vorbereiten, um bswp. das Model des dest-VC zu setzten oder sonstigen Payload zu übertragen

```
override func prepare(for segue: UIStoryboardSegue, sender: Any?)
```

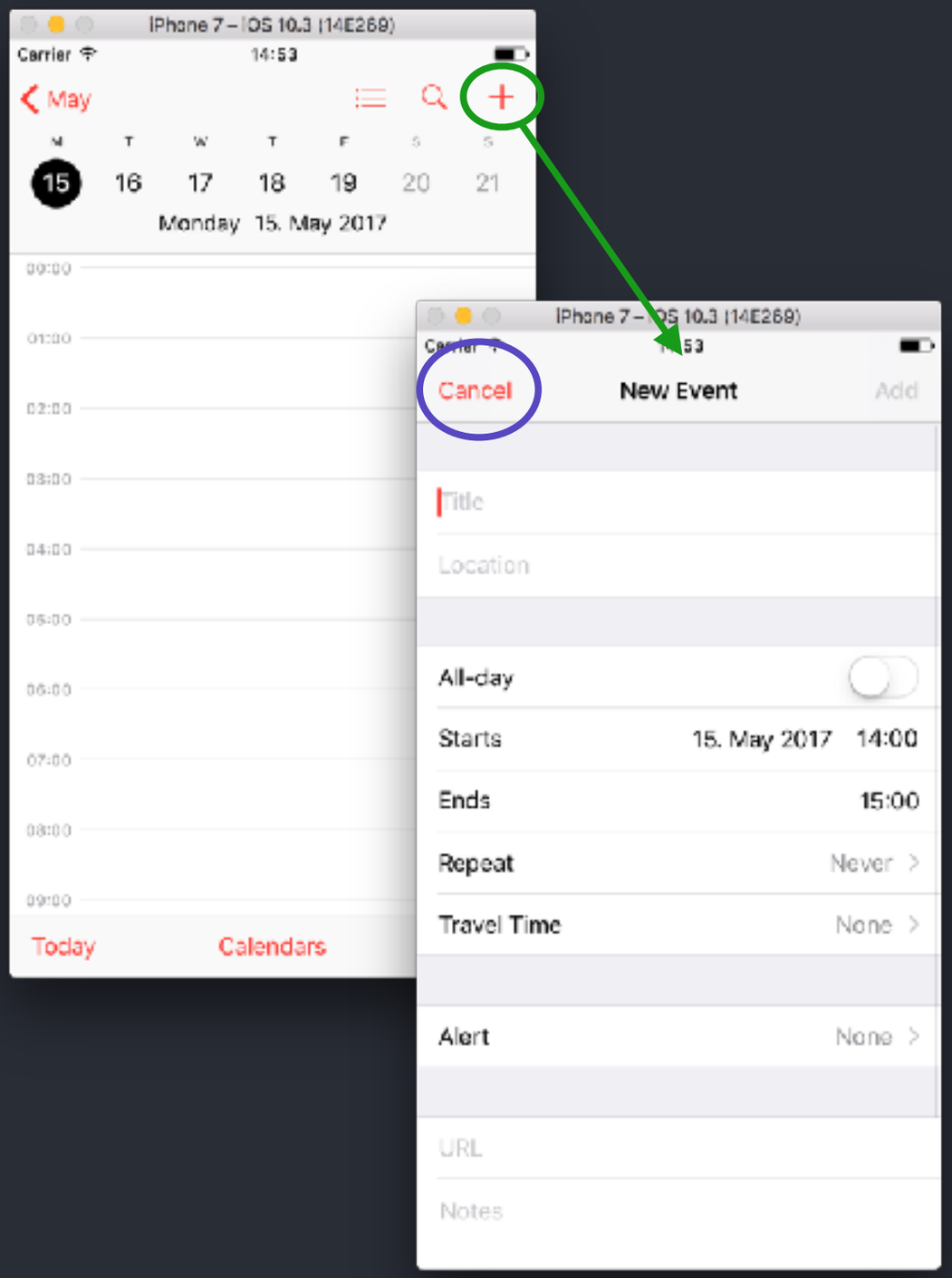
- Innerhalb von prepare(for segue) ist auf den VC-Lifecycle zu achten. Hierbei sind beispielsweise die Outlets des dest-VC noch nicht gesetzt

Segues



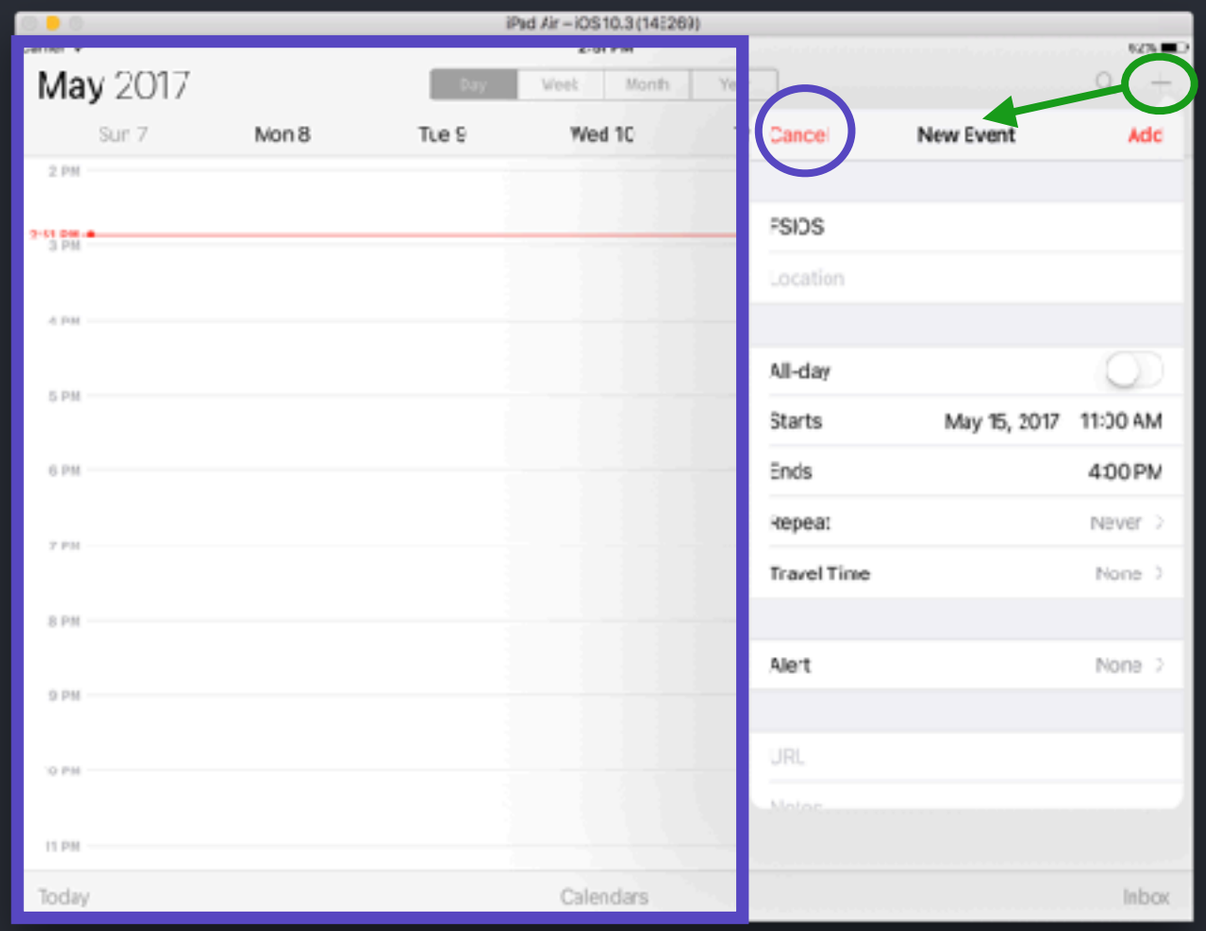
- Detail: Erstellt und präsentiert einen MVC, indem der VC auf den aktuellen Navigations-Stack gepushed wird
- Modal: Erstellt und präsentiert ein MVC über den gesamten Screen
- Popover: Fast das gleiche wie Modal, nur dass es beim iPad “über” den Screen gelegt wird
- Unwind: Erstellt kein neues MVC, sondern kehrt zu einem bestehenden MVC zurück
- Embed: Einen gesamten MVC-Stack innerhalb eines VC “einbetten”

Segues



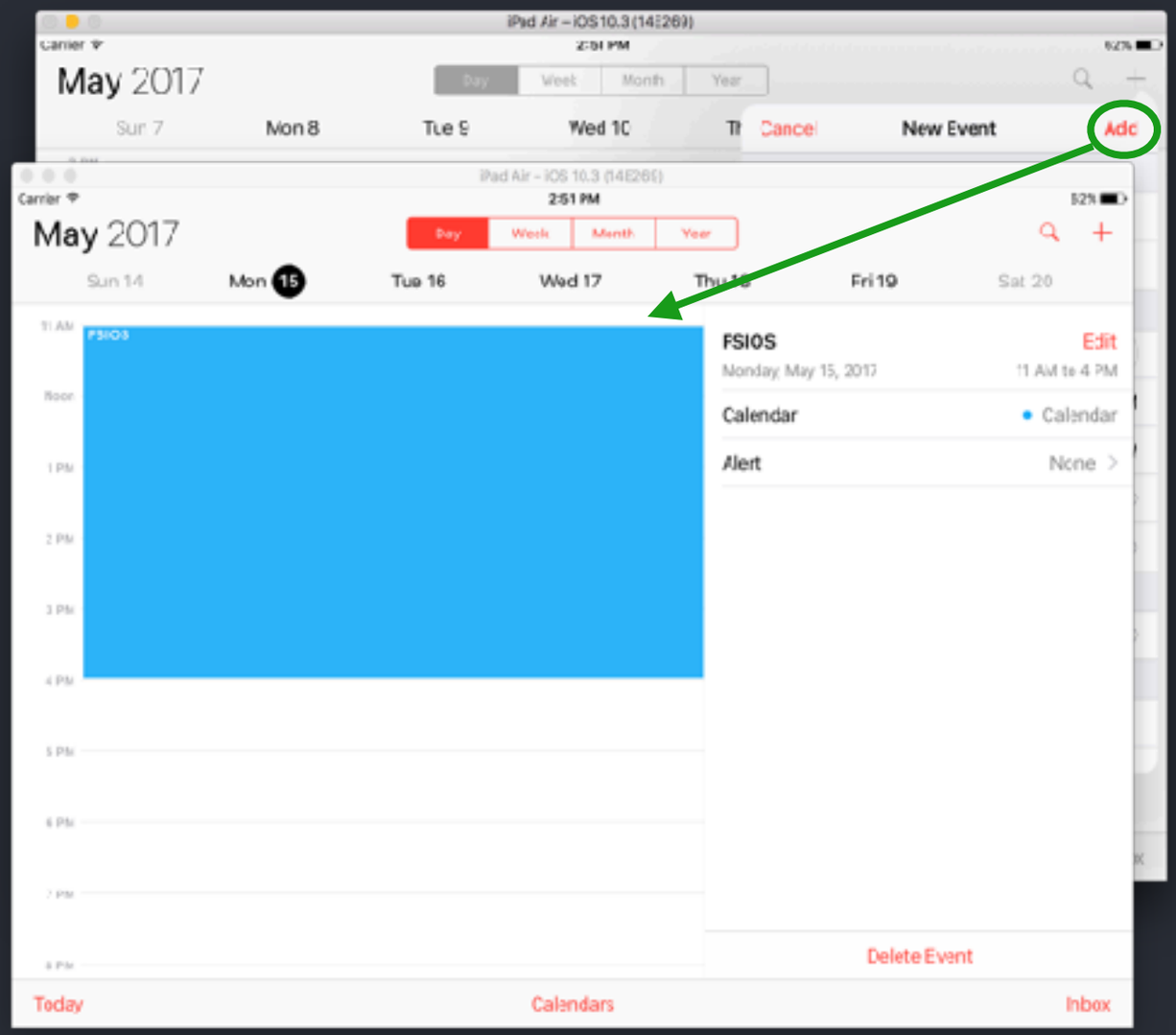
- Detail: Erstellt und präsentiert einen MVC, indem der VC auf den aktuellen Navigations-Stack gepushed wird
- Modal: Erstellt und präsentiert ein MVC über den gesamten Screen
- Popover: Fast das gleiche wie Modal, nur dass es beim iPad “über” den Screen gelegt wird
- Unwind: Erstellt kein neues MVC, sondern kehrt zu einem bestehenden MVC zurück
- Embed: Einen gesamten MVC-Stack innerhalb eines VC “einbetten”

Segues



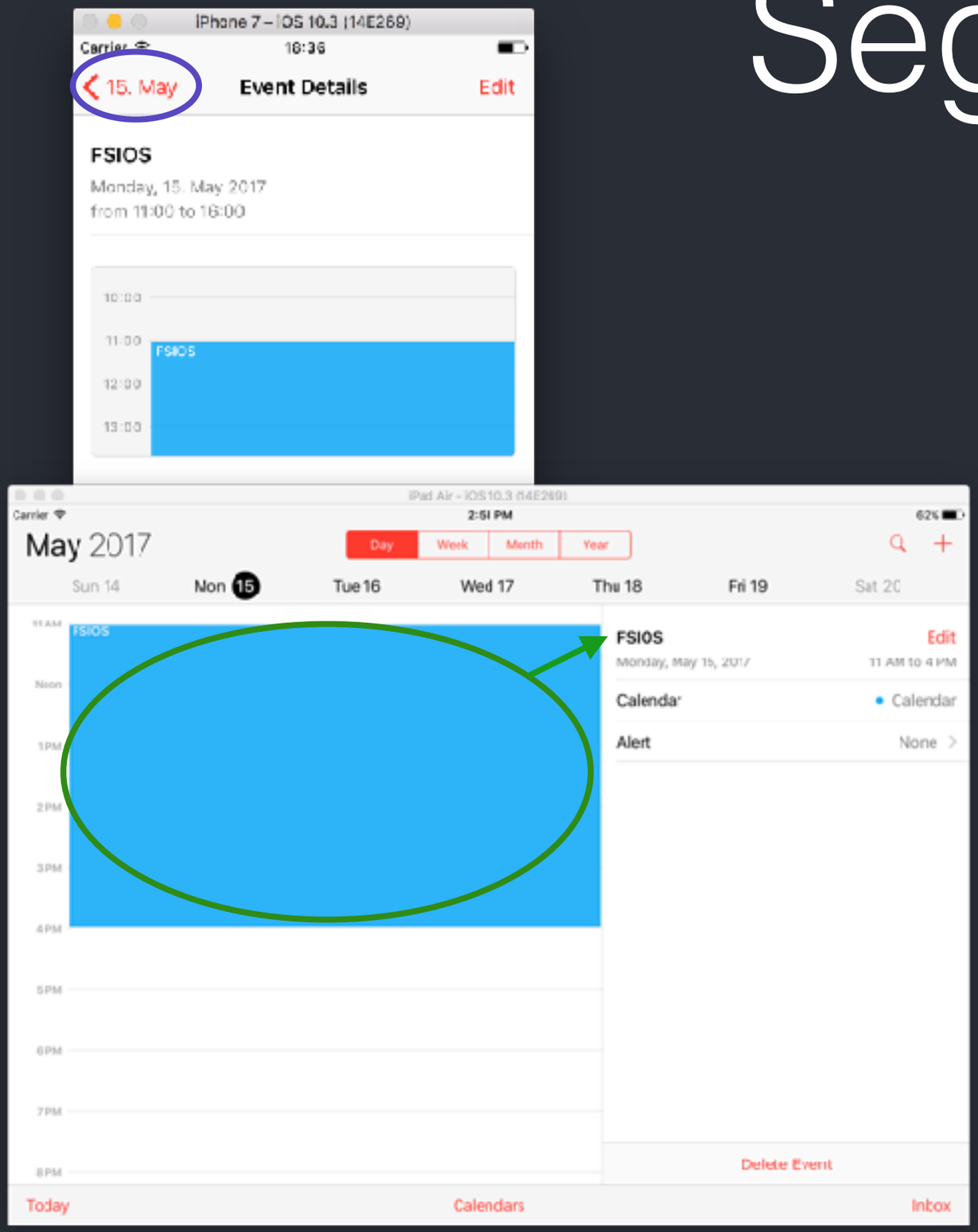
- Detail: Erstellt und präsentiert einen MVC, indem der VC auf den aktuellen Navigations-Stack gepushed wird
- Modal: Erstellt und präsentiert ein MVC über den gesamten Screen
- Popover: Fast das gleiche wie Modal, nur dass es beim iPad “über” den Screen gelegt wird
- Unwind: Erstellt kein neues MVC, sondern kehrt zu einem bestehenden MVC zurück
- Embed: Einen gesamten MVC-Stack innerhalb eines VC “einbetten”

Segues



- Detail: Erstellt und präsentiert einen MVC, indem der VC auf den aktuellen Navigations-Stack gepushed wird
- Modal: Erstellt und präsentiert ein MVC über den gesamten Screen
- Popover: Fast das gleiche wie Modal, nur dass es beim iPad “über” den Screen gelegt wird
- Unwind: Erstellt kein neues MVC, sondern kehrt zu einem bestehenden MVC zurück
- Embed: Einen gesamten MVC-Stack innerhalb eines VC “einbetten”

Segues



- Detail: Erstellt und präsentiert einen MVC, indem der VC auf den aktuellen Navigations-Stack gepushed wird
- Modal: Erstellt und präsentiert ein MVC über den gesamten Screen
- Popover: Fast das gleiche wie Modal, nur dass es beim iPad “über” den Screen gelegt wird
- Unwind: Erstellt kein neues MVC, sondern kehrt zu einem bestehenden MVC zurück
- Embed: Einen gesamten MVC-Stack innerhalb eines VC “einbetten”

Morgen

TableView, Multithreading, Networking und Storage

Demo - CardGames

PageViewController und ContainerView, Segues (Embedded, Unwind, Modal), Multiple MVC, UISegmentedControl, UISlider, Application-Lifecycle, verbesserte Custom-Views