BEUTH HOCHSCHULE FÜR TECHNIK BERLIN
University of Applied Sciences

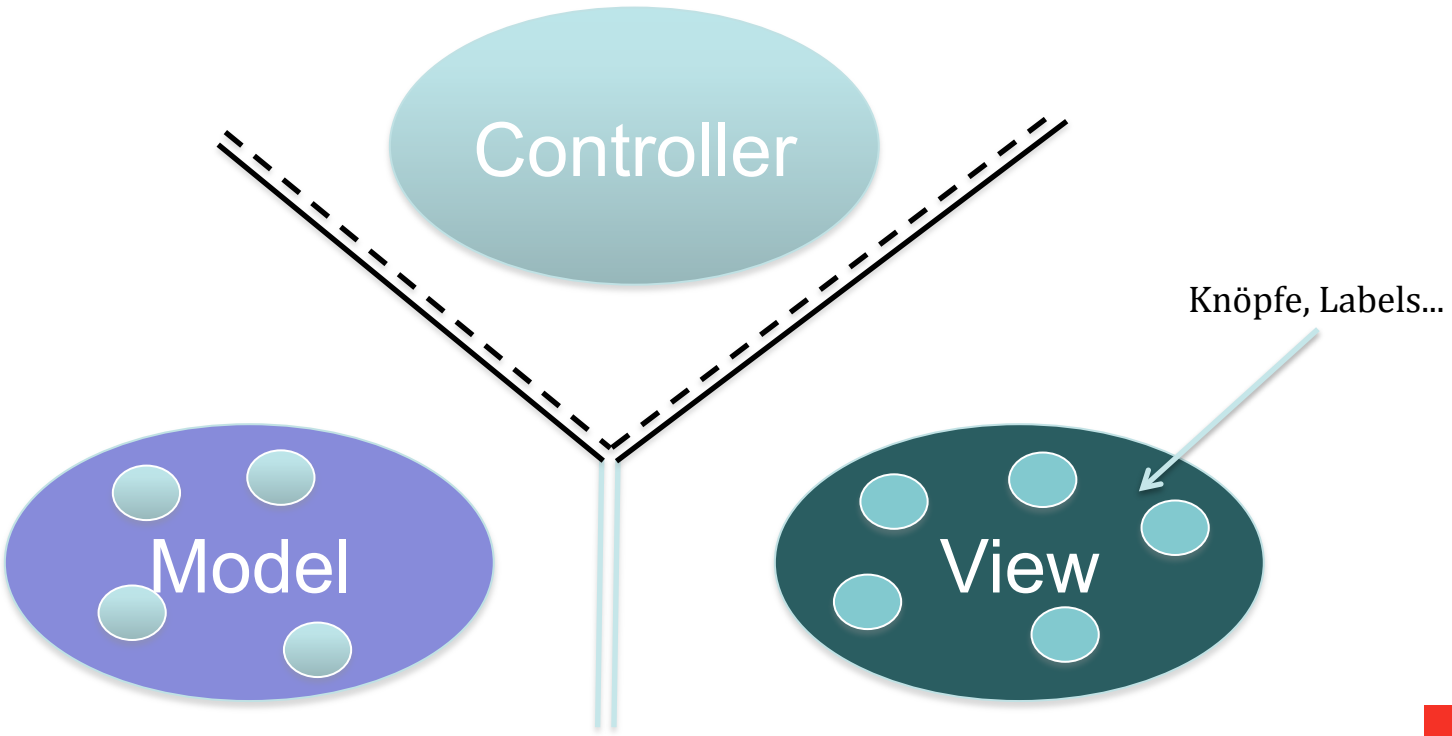
Entwicklung mobiler Applikationen für iPhone und iPad

- MAD 8 -

Prof. Dr. Dragan Macos

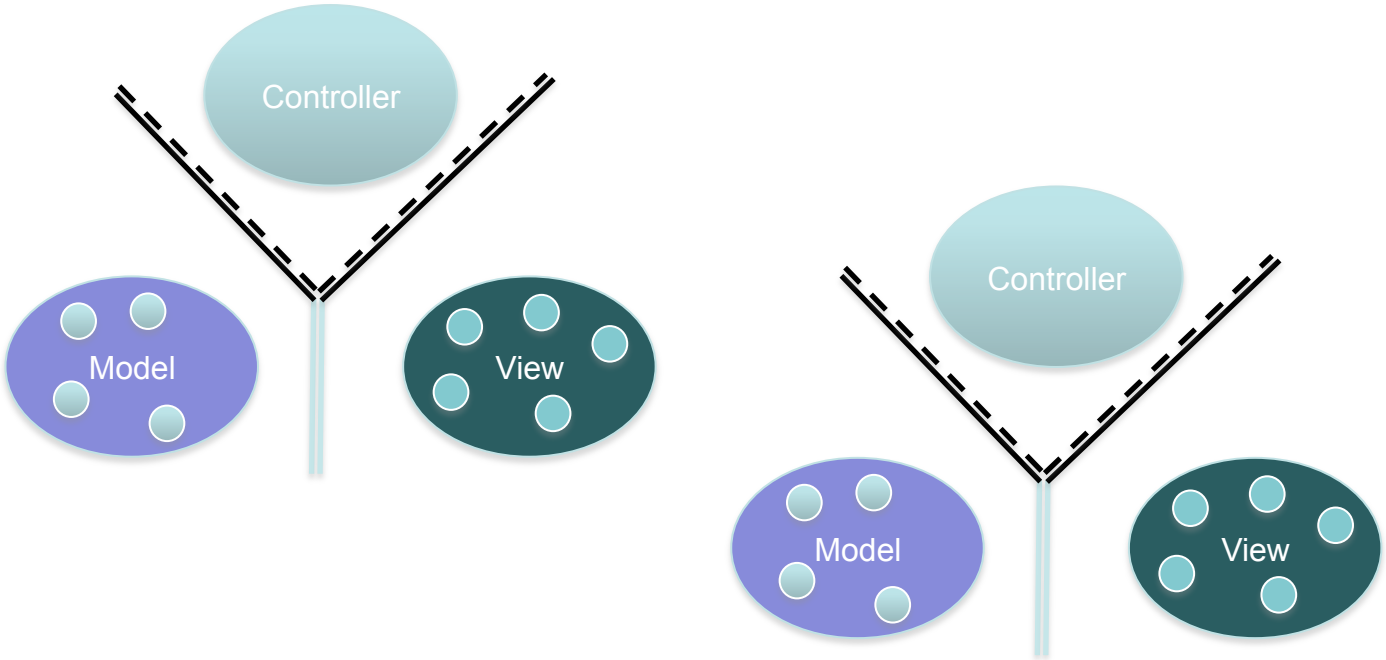
View Controller

- Komplexität unserer Applikationen bis jetzt:
  - Eine model-view-controller-Kombination



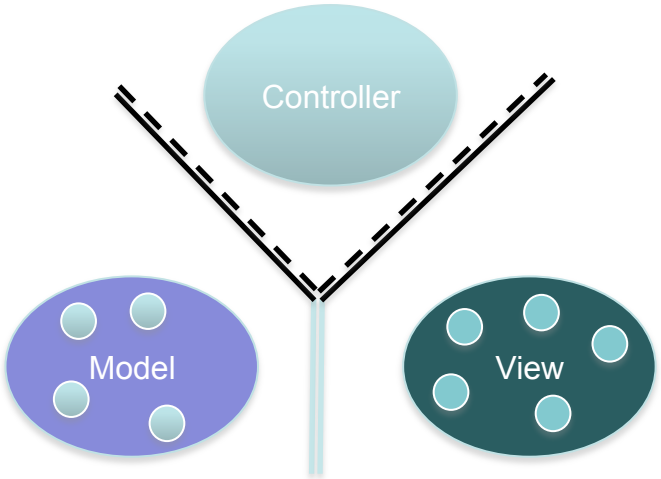
Mehrere MVC

- Wenn wir mehrere unterschiedliche Funktionalitäten haben, müssen wir mehrere unterschiedliche Controller implementieren
  - Mehrere model-view-controller-Kombinationen

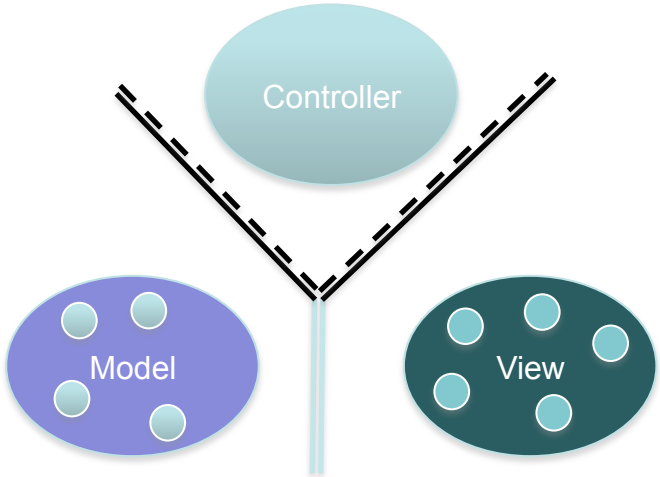


Mehrere MVC

- Wenn wir mehrere unterschiedliche Funktionalitäten - Arten haben, müssen wir mehrere unterschiedliche Controller implementieren.
  - Ein Screen -ein Controller
  - Mehrere model-view-controller-Kombinationen



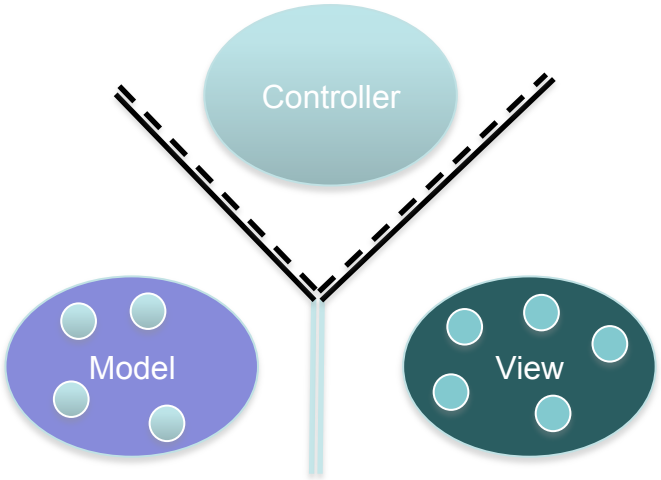
z. B. Daten-Erfassungsscreen



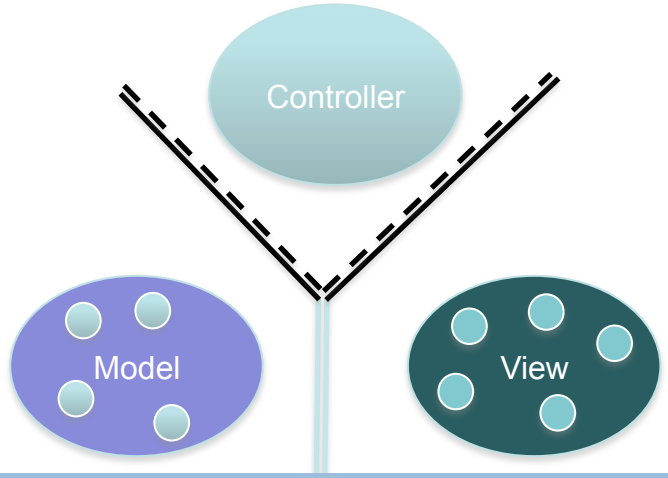
z. B. Navigationsscreen

Mehrere MVC

- Wenn wir mehrere unterschiedliche Funktionalitäten – Arten haben, müssen wir mehrere unterschiedliche Controller implementieren.
  - Ein Screen -ein Controller
  - Mehrere model-view-controller-Kombinationen



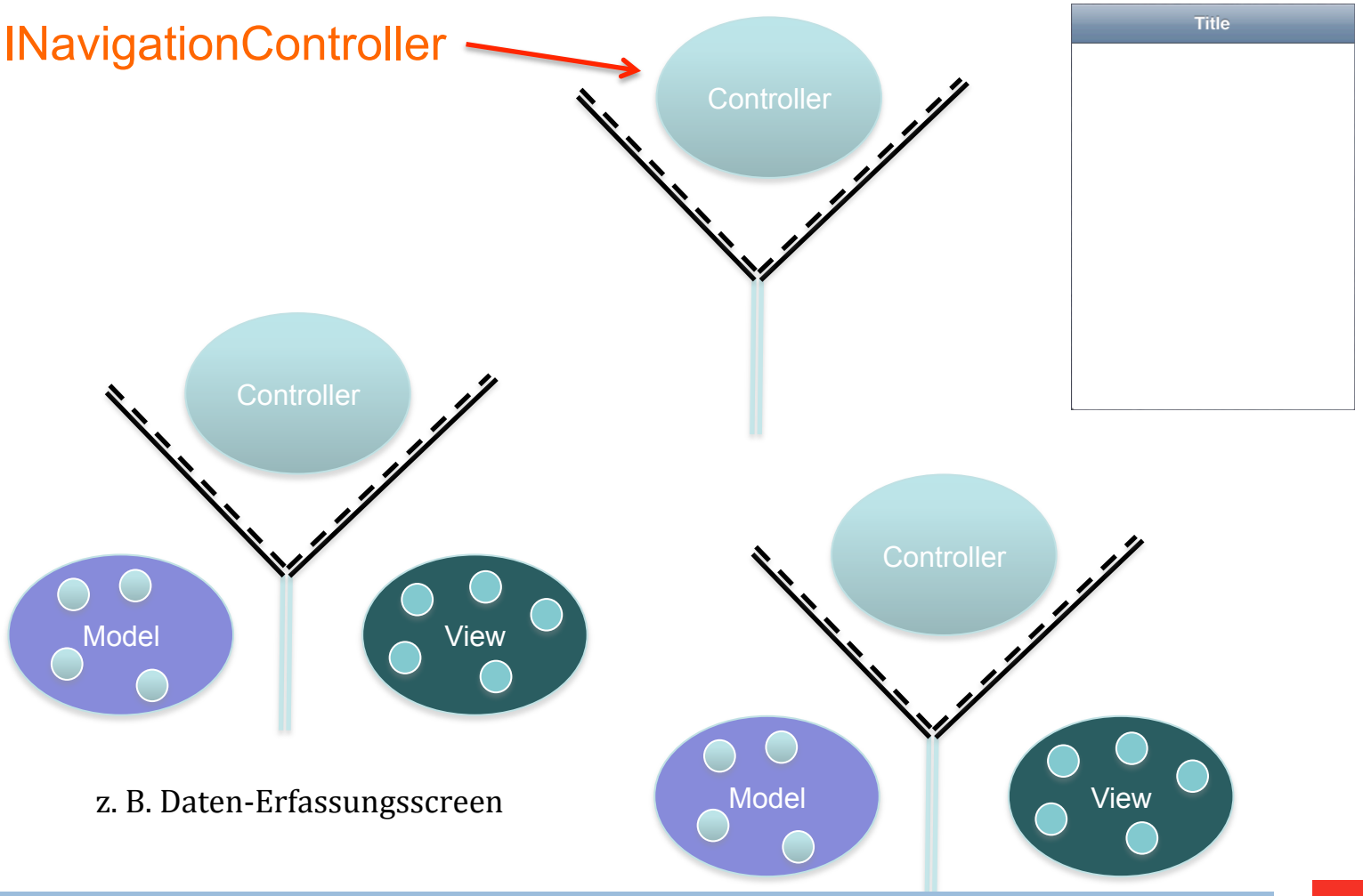
z. B. Daten-Erfassungsscreen



Wir brauchen einen Controller für die Controller

Mehrere MVC

- UINavigationController

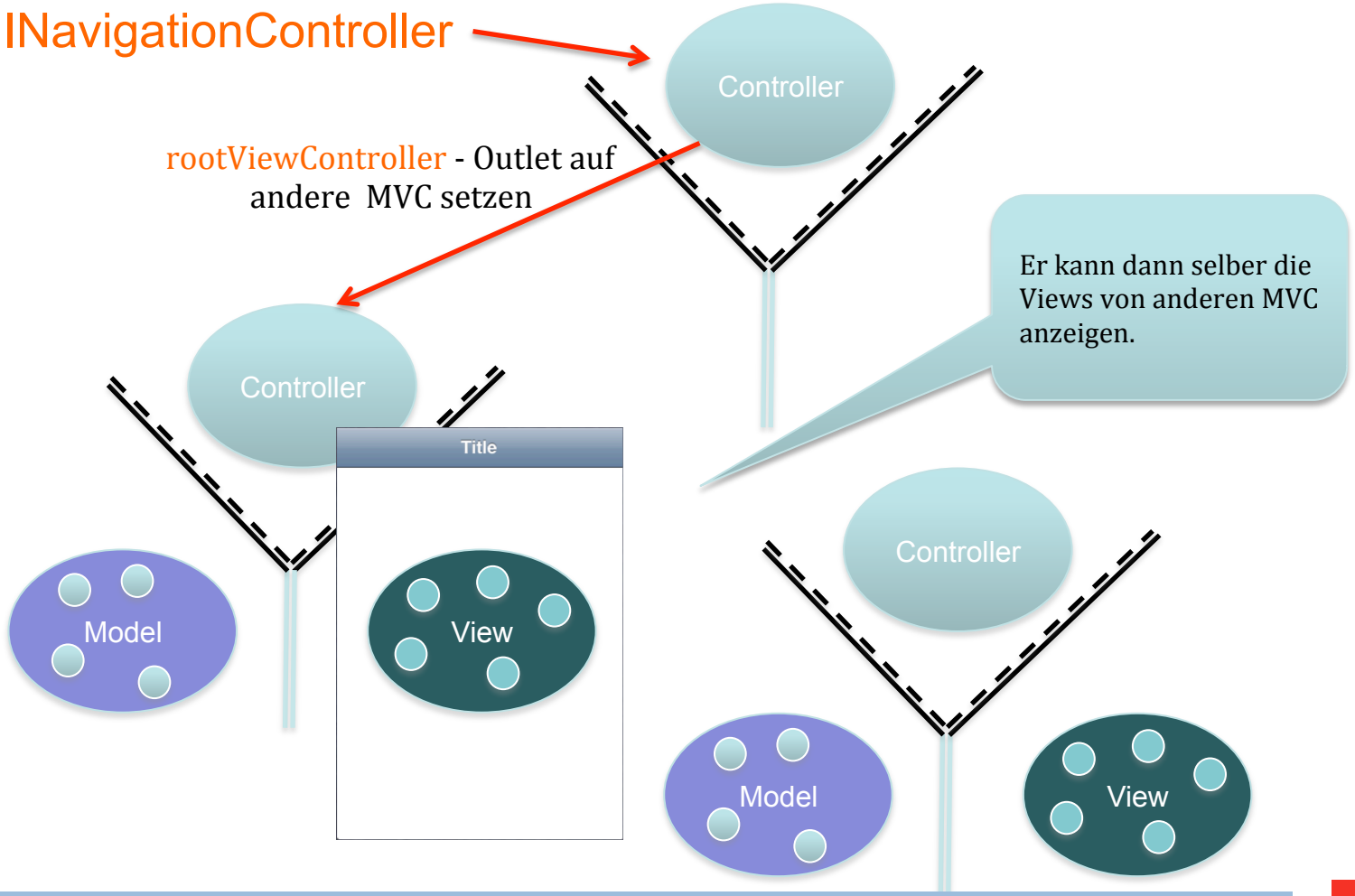


z. B. Daten-Erfassungsscreen

Wir brauchen einen Controller für die Controller

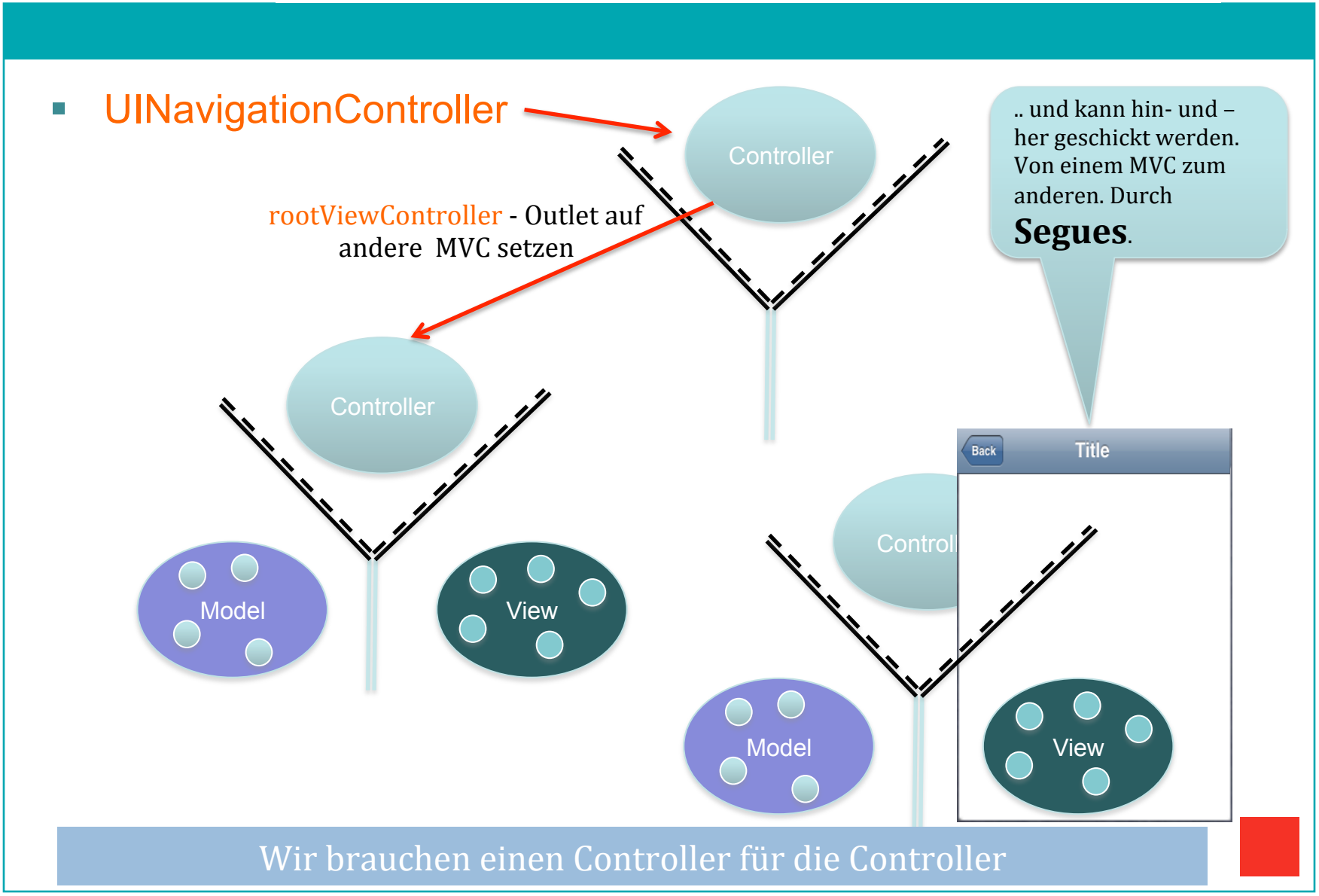
Mehrere MVC

- UINavigationController

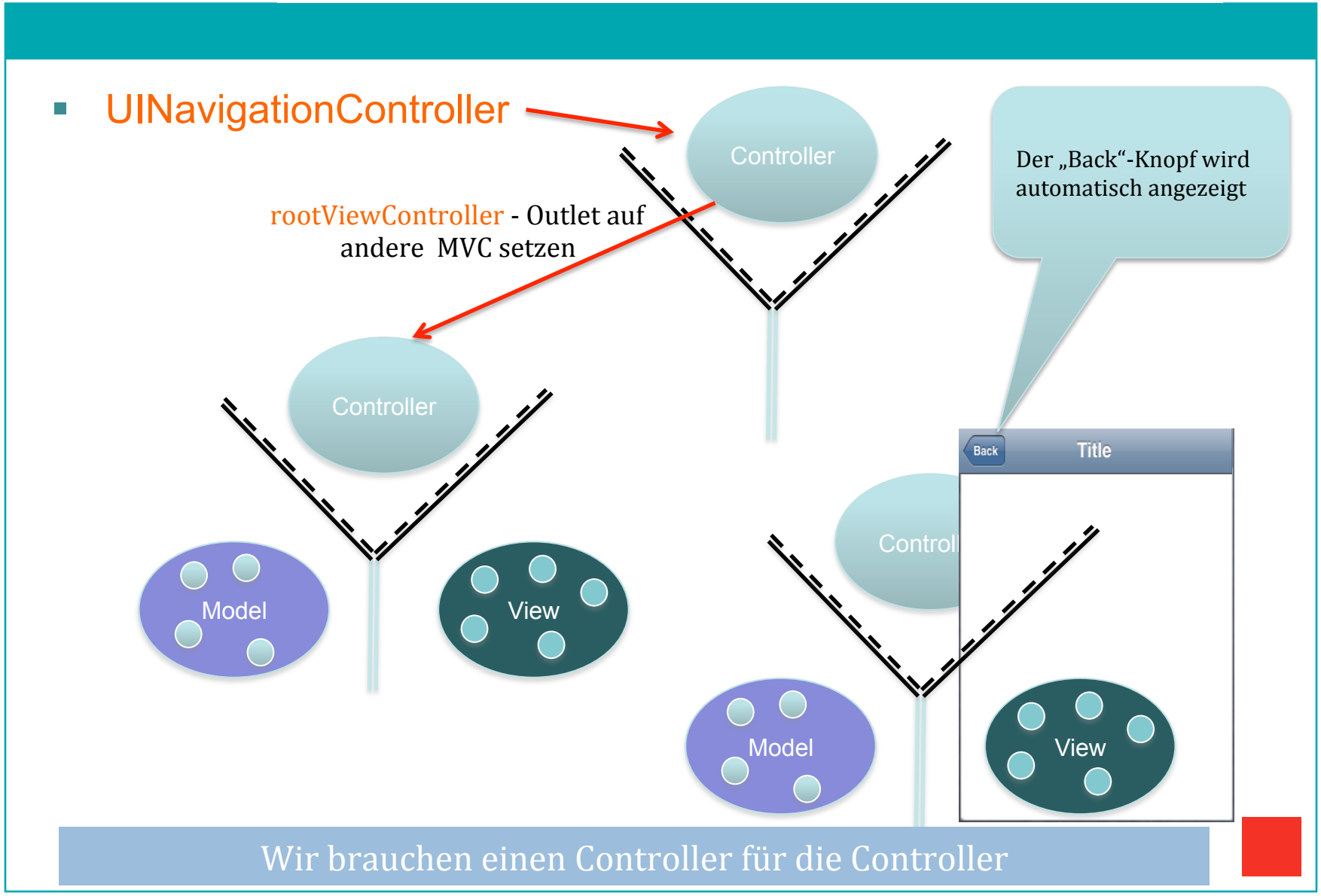


Wir brauchen einen Controller für die Controller

Mehrere MVC


UINavigationController
Controller
rootViewController - Outlet auf andere MVC setzen
.. und kann hin- und –her geschickt werden. Von einem MVC zum anderen. Durch **Segues**.
Controller
Model
View
Back
Title
Control
Model
View
Wir brauchen einen Controller für die Controller

Mehrere MVC

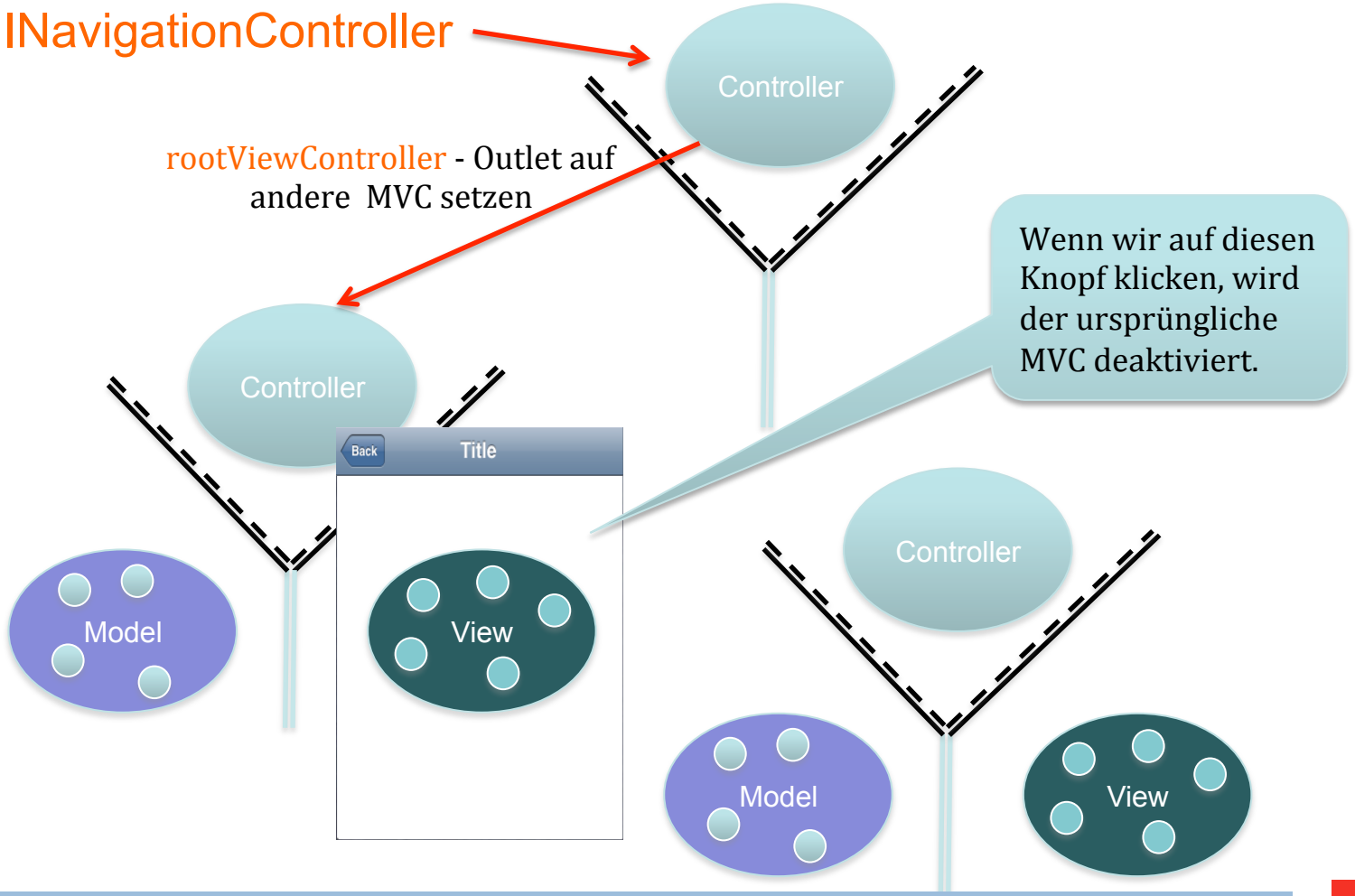


- UINavigationController

Wir brauchen einen Controller für die Controller

Mehrere MVC

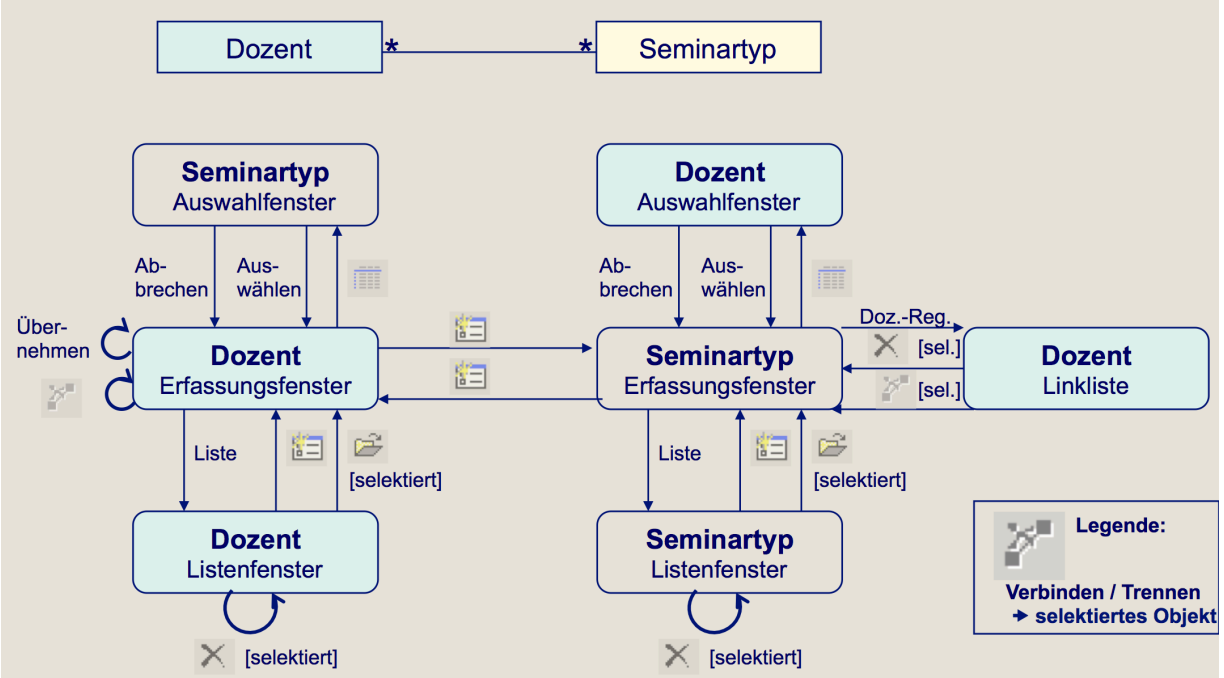
- UINavigationController



Wir brauchen einen Controller für die Controller

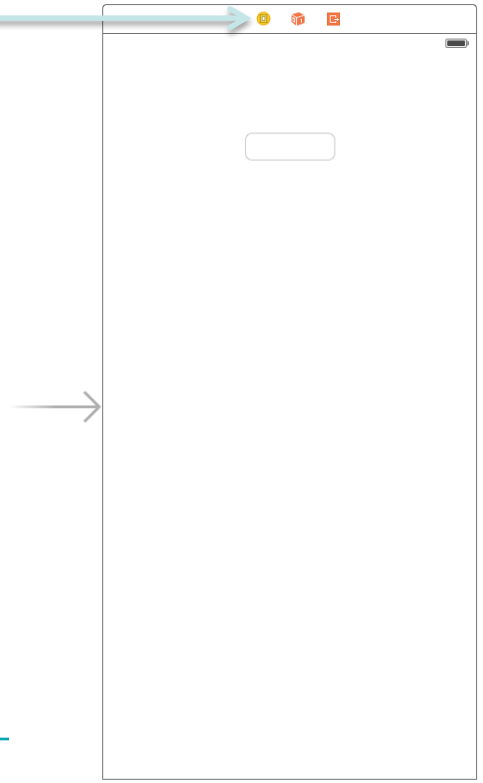
Software-Engineering-Grundlage

- Software-Ergonomie: Wissenschaftliche Disziplin, die sich mit der Entwicklung gebrauchstauglicher Systeme beschäftigt.
- Dynamik eines Dialogsystems wird durch State-Chart-Diagramme definiert.

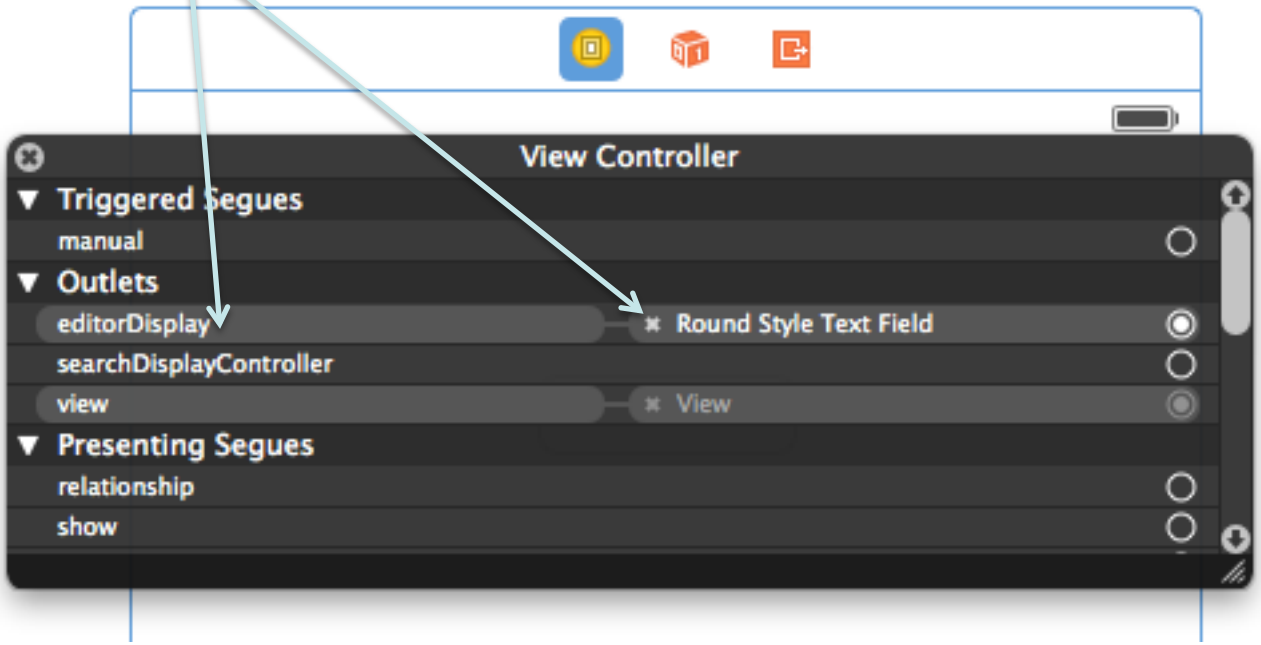


Segues

- Die Segues definieren die Übergänge zwischen MVCs.
- Zustände: MVCs
- .. Aber zunächst einige Erklärungen im Storyboard.
- View Controller

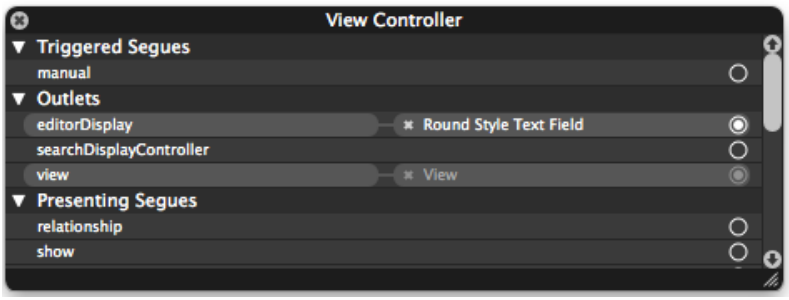


- Da sind alle Infos über den Views und dem Controller
- Z. B. Outlets



Outlet-Infos, Beispiel

```
9  import UIKit
10
11 class ViewController: UIViewController {
12
13
14
15     @IBOutlet weak var editorDisplay: UITextField!
16
17
```



Demo

- First Responder
  - Was ist First Responder?
  - Das „aktive“ Subview.
  - Das Subview, das die Aktionen des Benutzers entgegennimmt.

Demo, First Responder

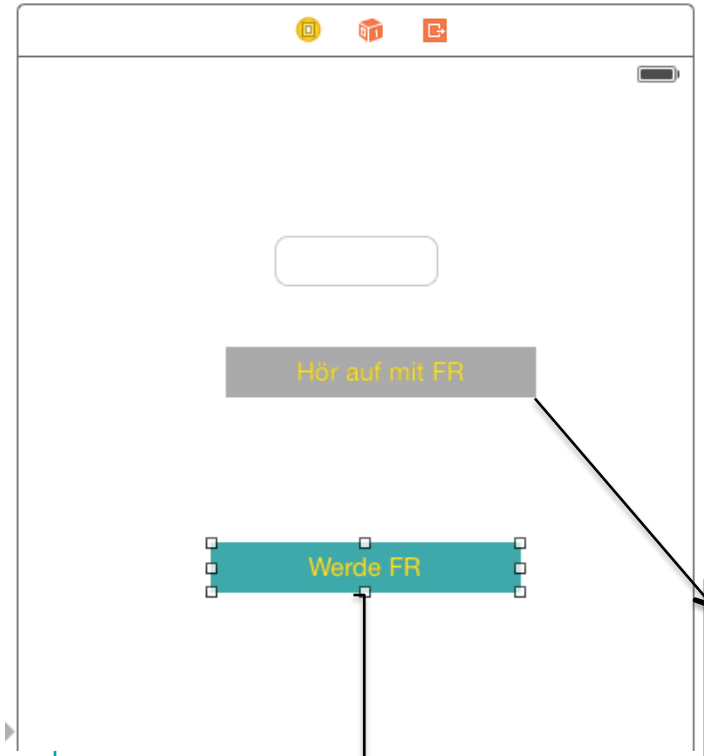
```
14
15  @IBOutlet weak var editorDisplay: UITextField!
16
17
```

First Responder

.. weil er die Kontrolle über die Benutzerreaktionen übernommen hat...

1.) iOS Simulator -> Hardware -> Keyboard
2.) "Connect Hardware Keyboard“ ausschalten

Demo, First Responder




```
14
15  @IBOutlet weak var editorDisplay: UITextField!
```

```
28  @IBAction func hoerAufFR(sender: AnyObject) {
29      editorDisplay.resignFirstResponder()
30  }
31
32  @IBAction func werdeFR(sender: AnyObject) {
33
34      editorDisplay.becomeFirstResponder()
35  }
```

First Responder-Rolle ein- und ausschalten.

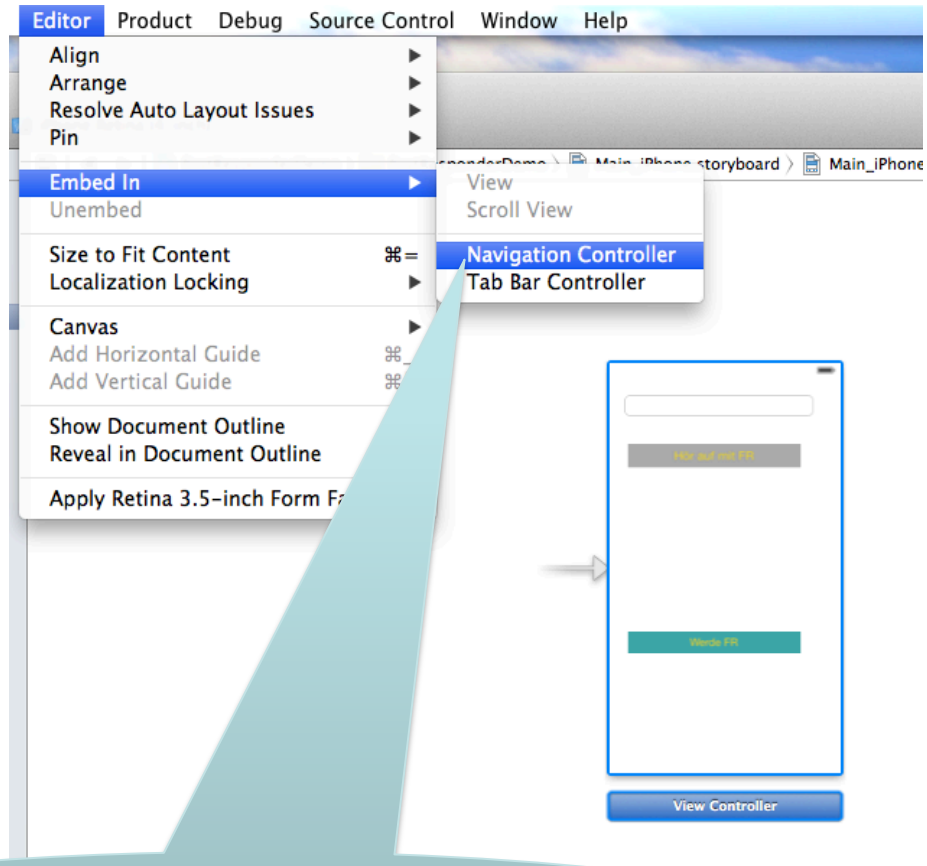
Demo Segues

View-Controller-Objekte im Storyboard einfach erzeugen

Wenn die Applikation gestartet wird, das ist unser erster Screen.

Da sind sie. Sie sind noch nicht angemeldet für das „Navigieren“.

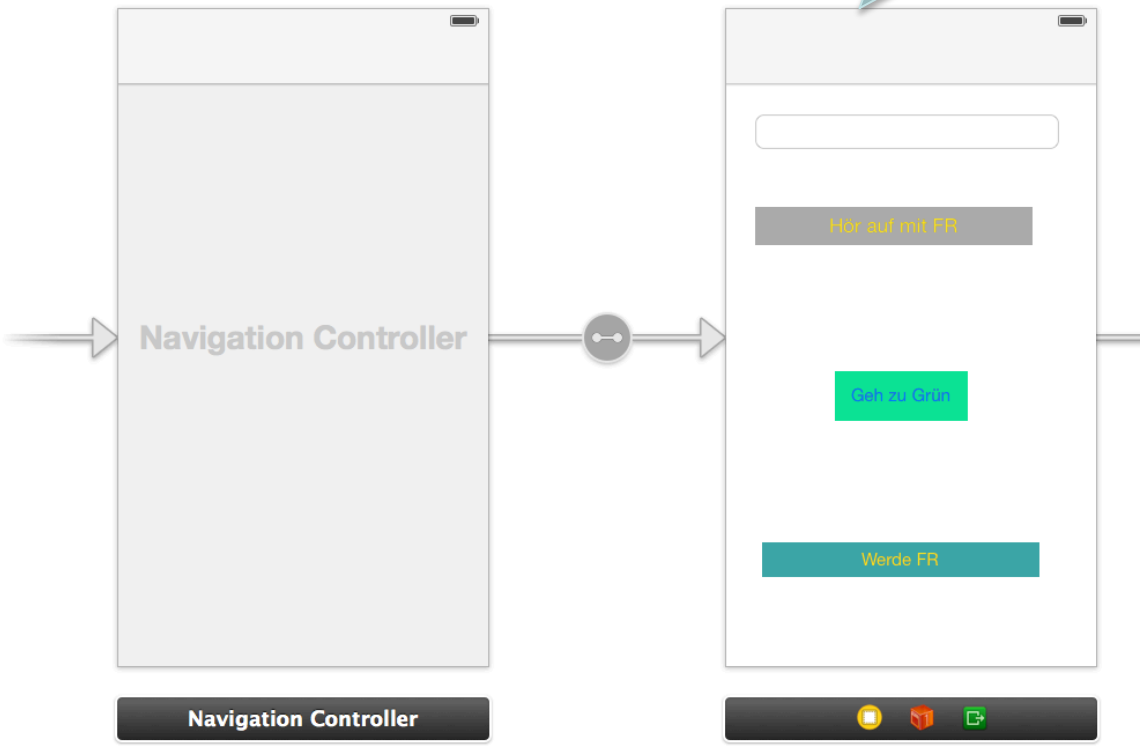
Demo Segues



Anmeldung bei dem Navigation Controller. Controller von Controllern. Er zeigt die Controller-Screens an, nach der Anwendung von Segues (*Seguwei* ☺)

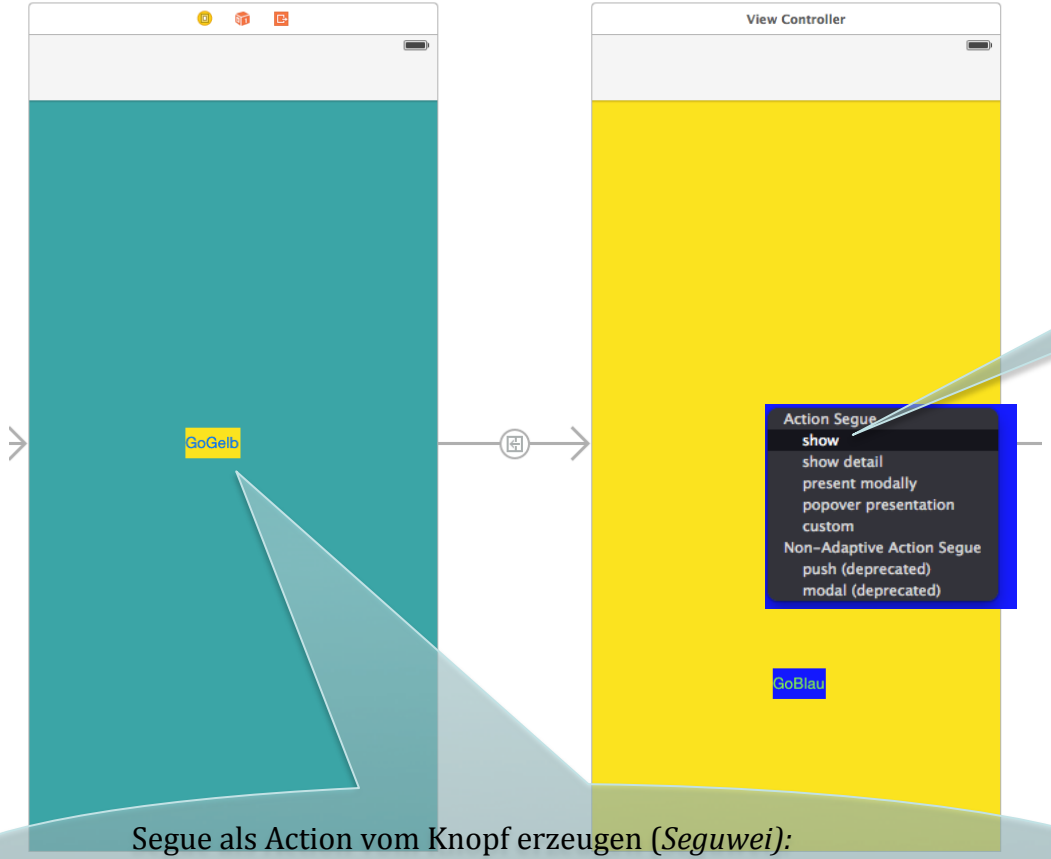
.

Navigation Controller „baut“ standardmäßig die Navigationsleiste ein. Mit dem „Back“-Knopf. Die Leiste kann ausgeschaltet/geändert werden.



Segues

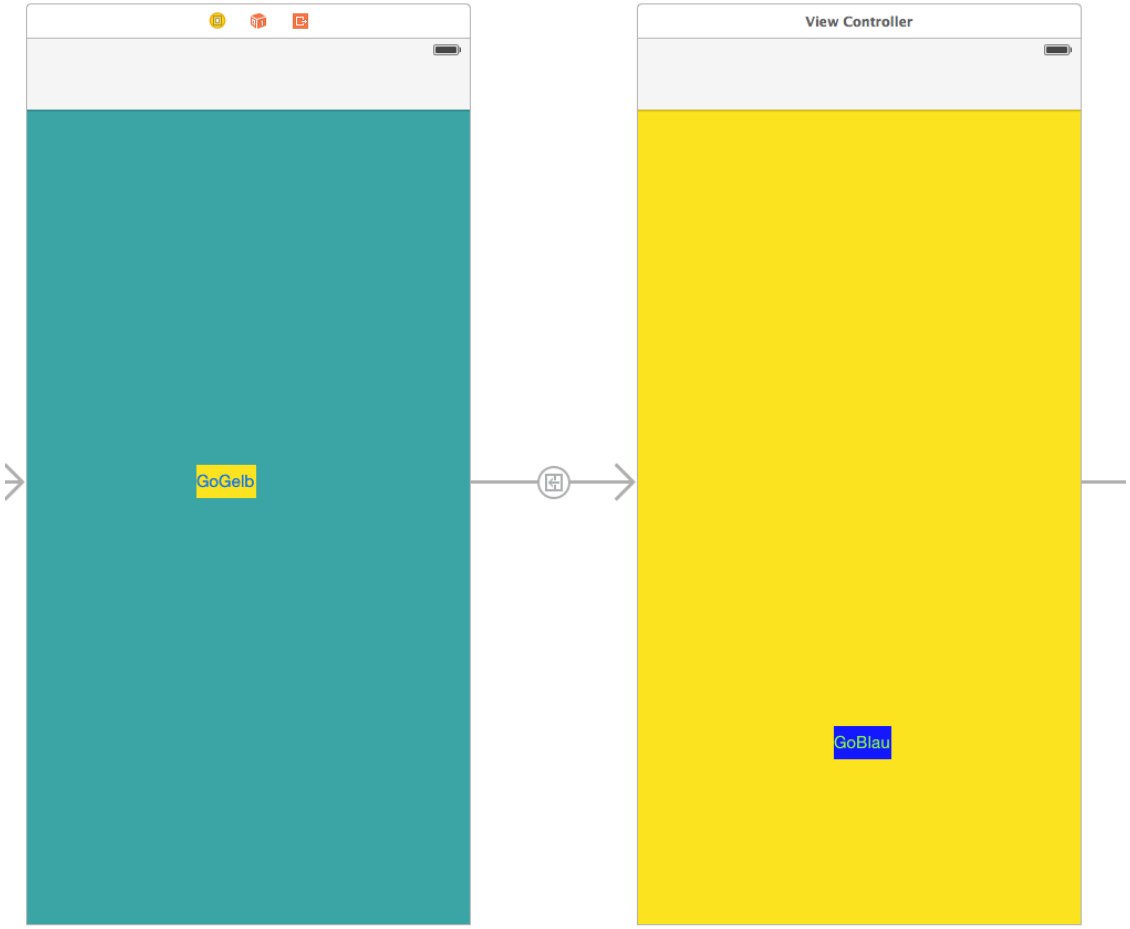
Segues: Übergänge von einem zum anderen View Controller



Wir machen jetzt Show-Segues. Die haben einen Push-Charakter.

Segue als Action vom Knopf erzeugen (*Seguwei):*
-rechte Maustaste (oder ctrl-linke Maustaste), den Knopf mit dem Controller verbinden.

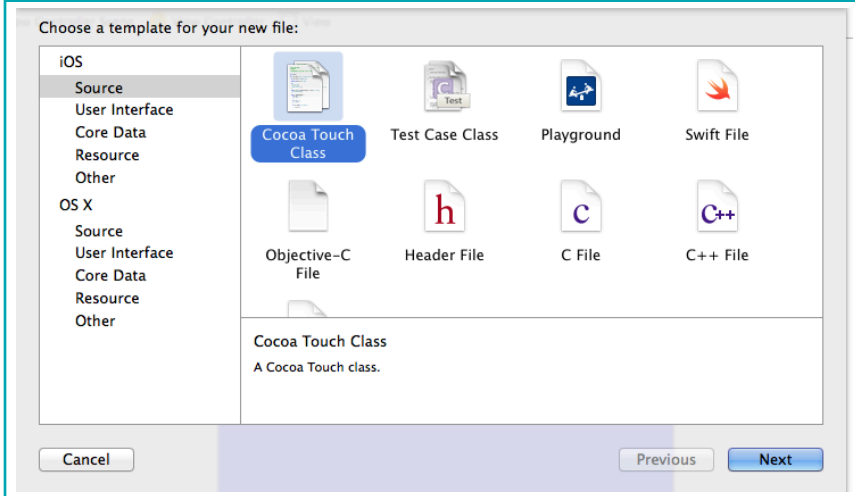
Demo Segues

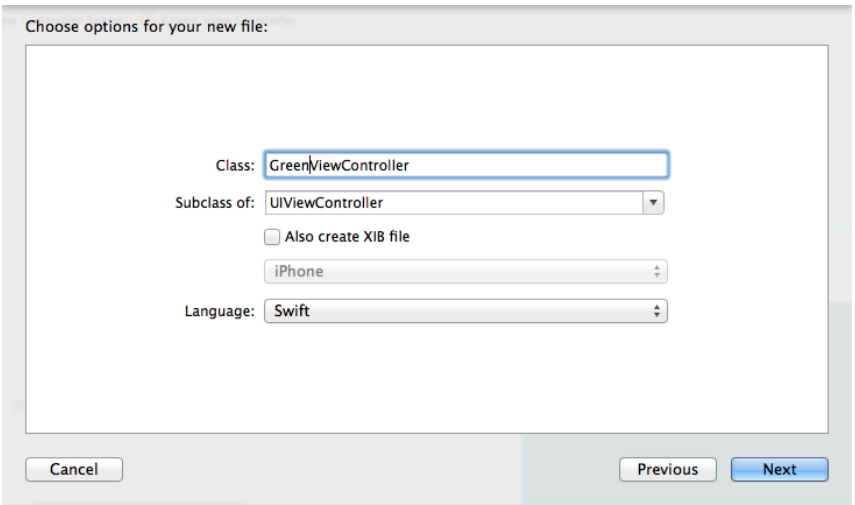


Demo Segues

- Die Kette von Segues wird im Navigation Controller als Stack („Stapel“) realisiert
- Wenn ein Segue ausgeführt wird, wird der Controller auf den Stapel gelegt: **Push**.
- Beim Knopfdruck auf den „Back“-Knopf, wird der erste Controller vom Stapel genommen: **Pop**.
- Es wird immer der erste Controller auf dem Stapel angezeigt.

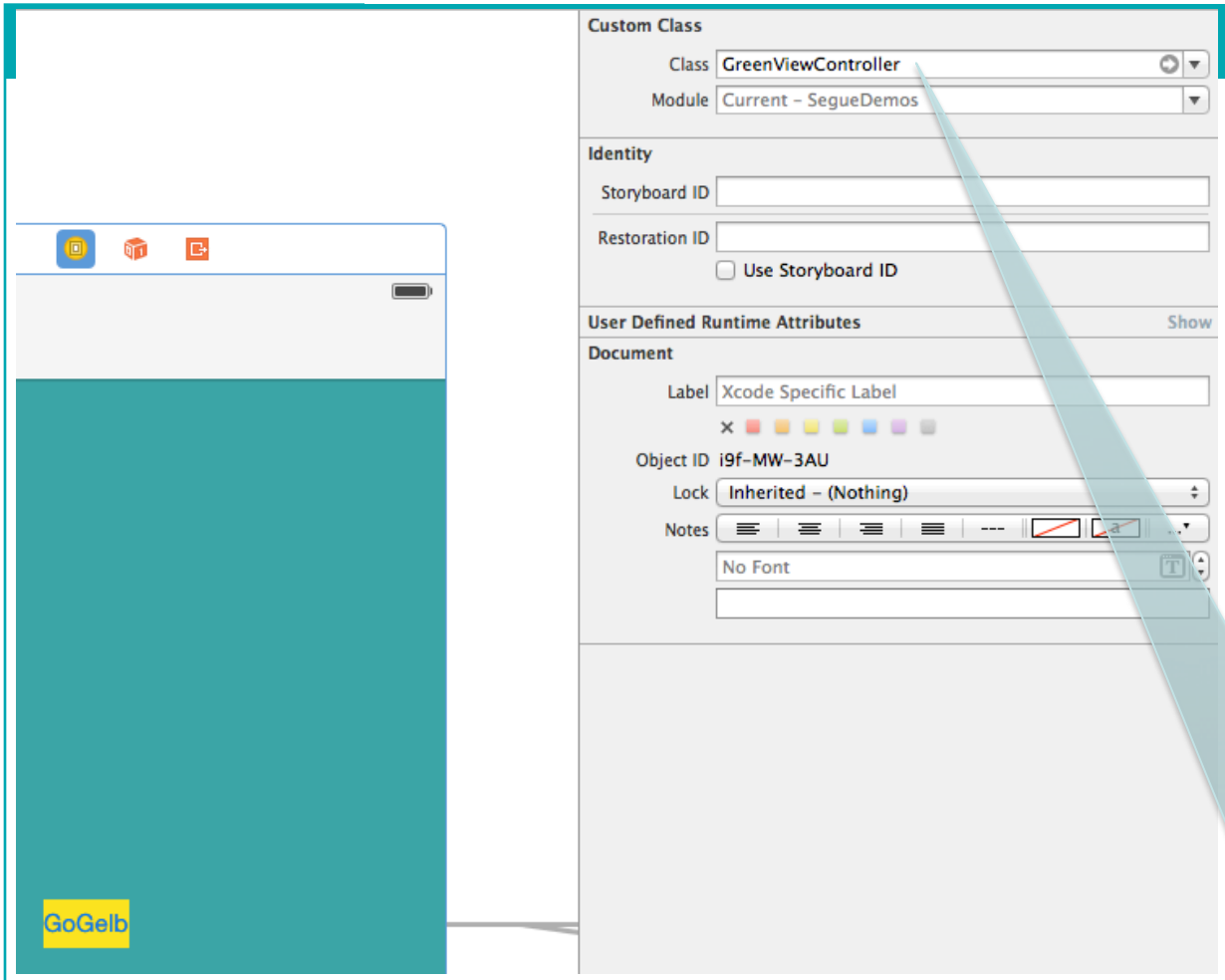
Demo Segues





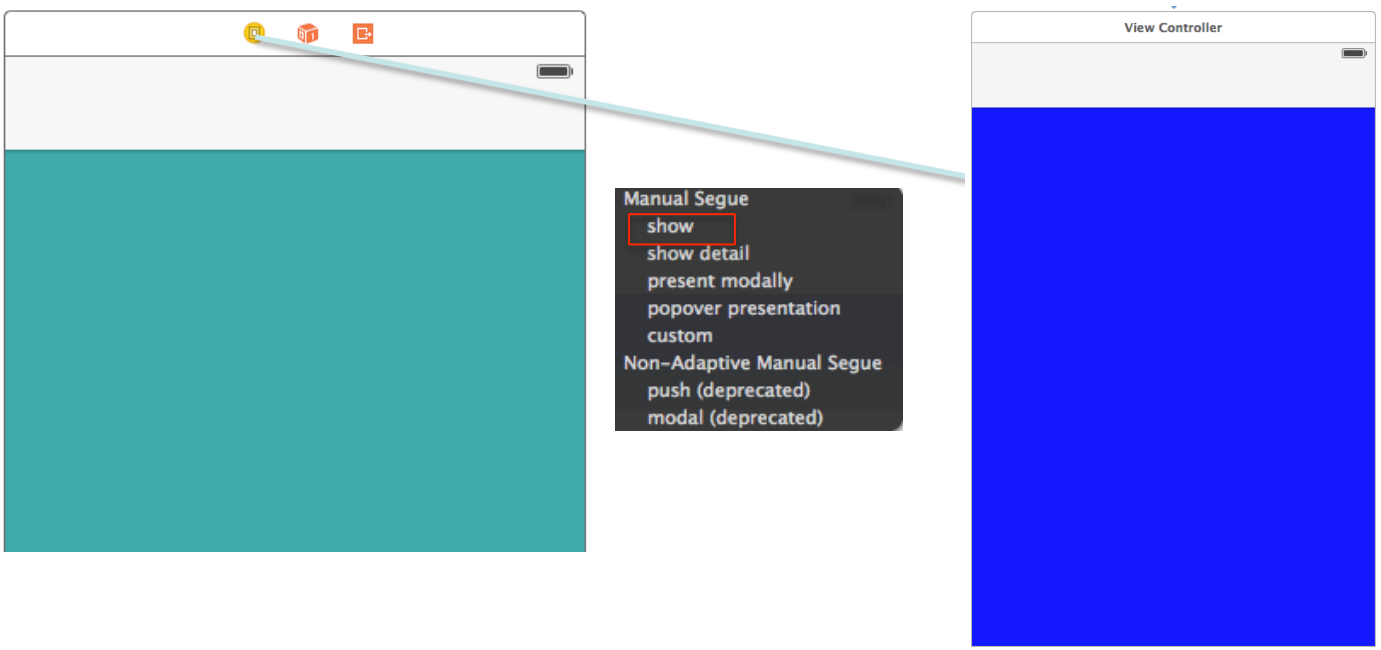
- Jeder neue Controller soll seine eigene Swift-Klasse haben
- Cocoa-Touch-Klasse, die Unterklasse der Klasse UIViewController ist.

Demo Segues



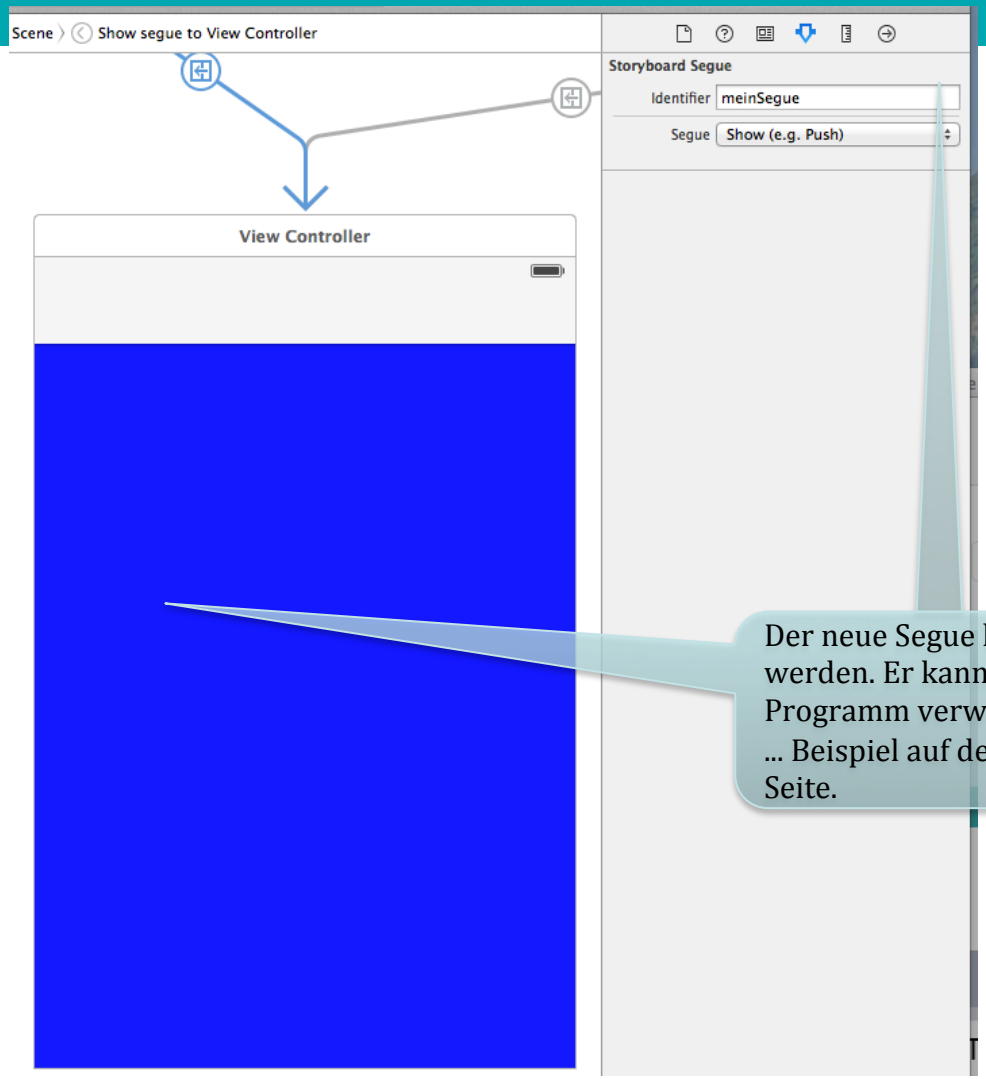
.. und die neue Klasse soll dem entsprechenden Controller im Storyboard zugeordnet werden.

Segues definieren

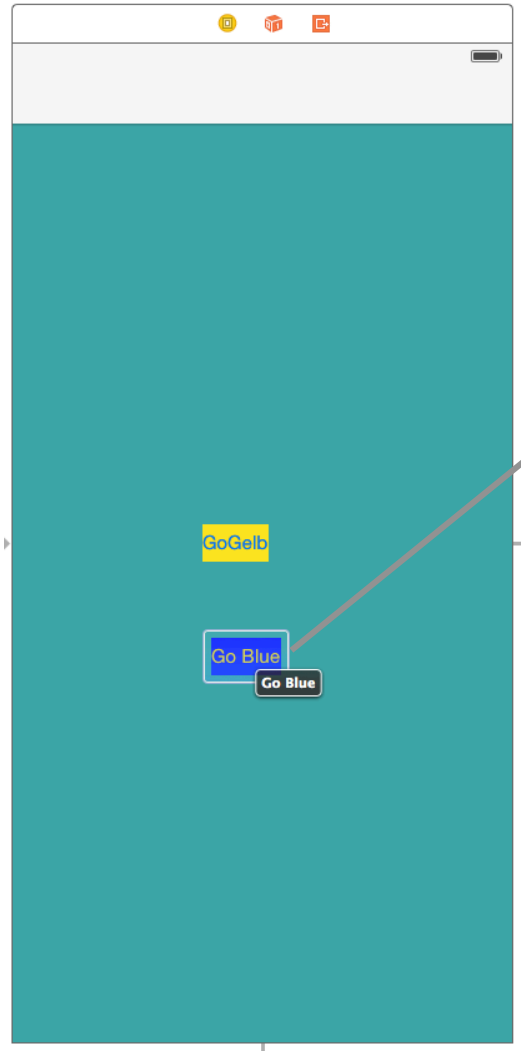


- Man kann ein Segue einfach als Übergang von einem Viewcontroller zum nächsten

Segues definieren



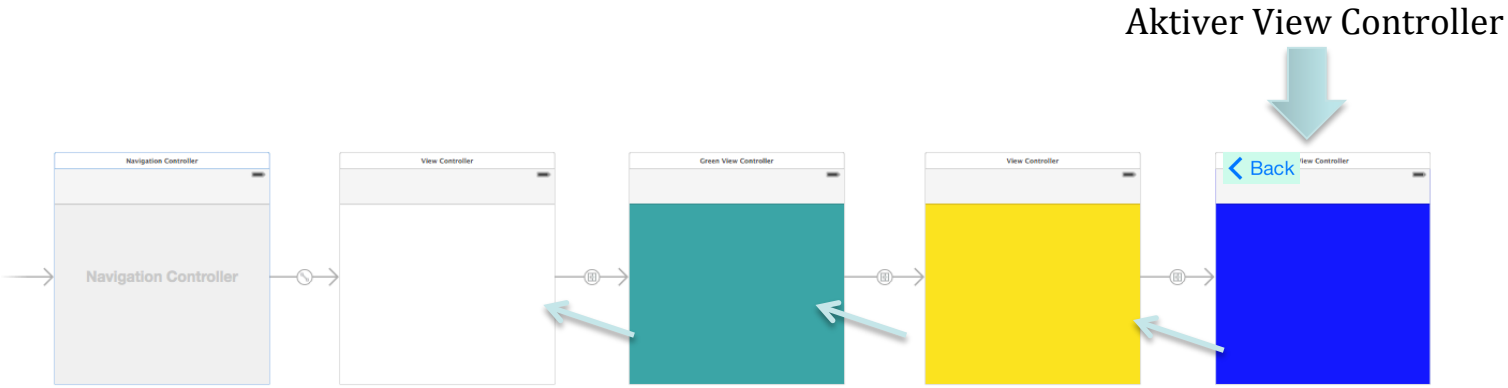
Der neue Segue kann benannt werden. Er kann jetzt im Programm verwendet werden. ... Beispiel auf der nächsten Seite.




```
//
//  GreenViewController.swift
//  SegueDemos
//
//  Created by Dragan Macos on 13.11.14.
//  Copyright (c) 2014 de.beuth-hochschule. All rights reserved.
//

import UIKit

class GreenViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()

        // Do any additional setup after loading the view.
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }


    @IBAction func trockenGang(sender: AnyObject) {
        self.performSegueWithIdentifier("meinSegue", sender: self)
    }

}
```

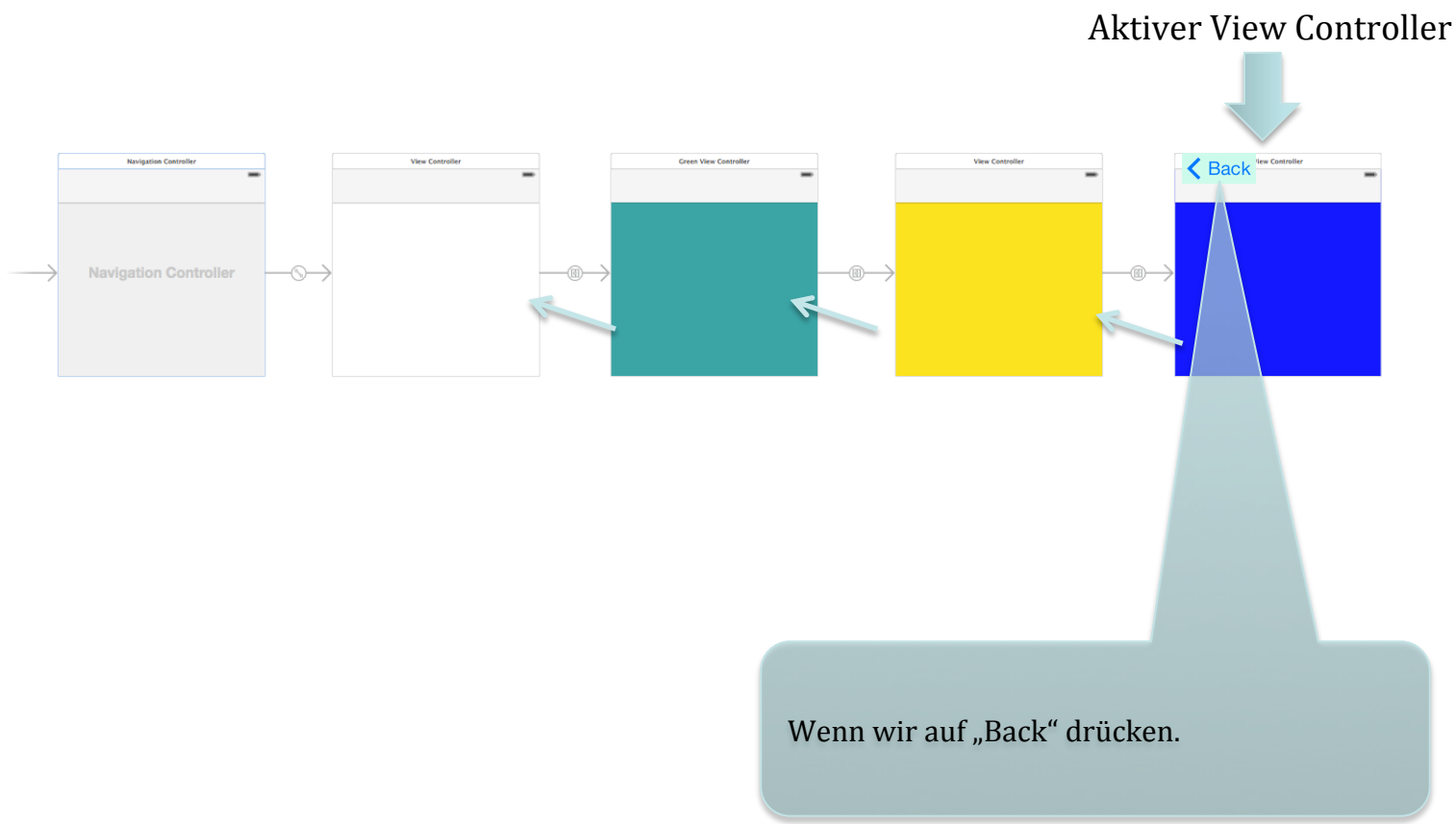
Das neue Segue.

Unwind Segues

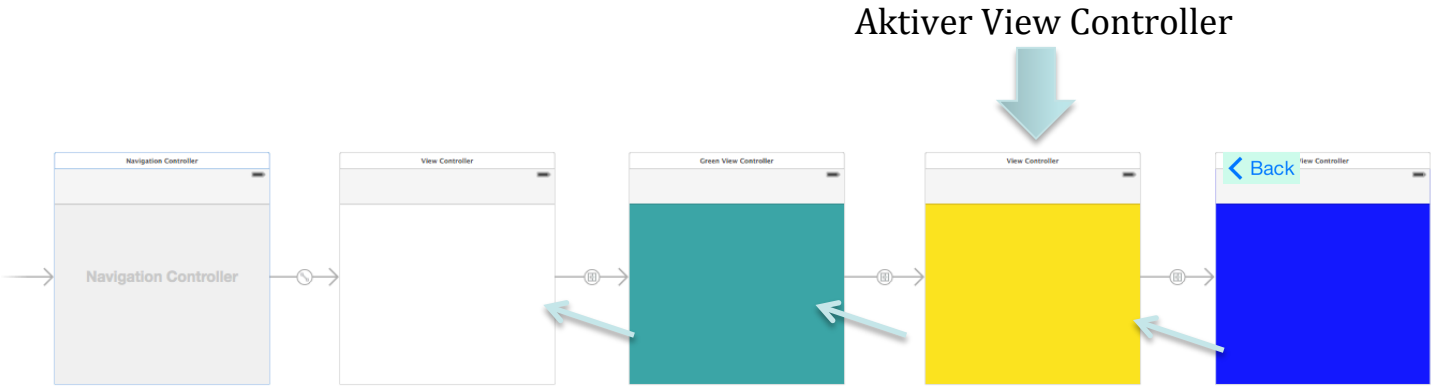


Vom weißen Controller zum gelben Controller können wir über den grünen und den blauen kommen. Durch entsprechende Knopfdrücke (wie gesehen, die Knöpfe sind mit Segues verbunden):
3xPush: Weiß-Grün, Grün-Gelb, Gelb-Blau.

Unwind Segues

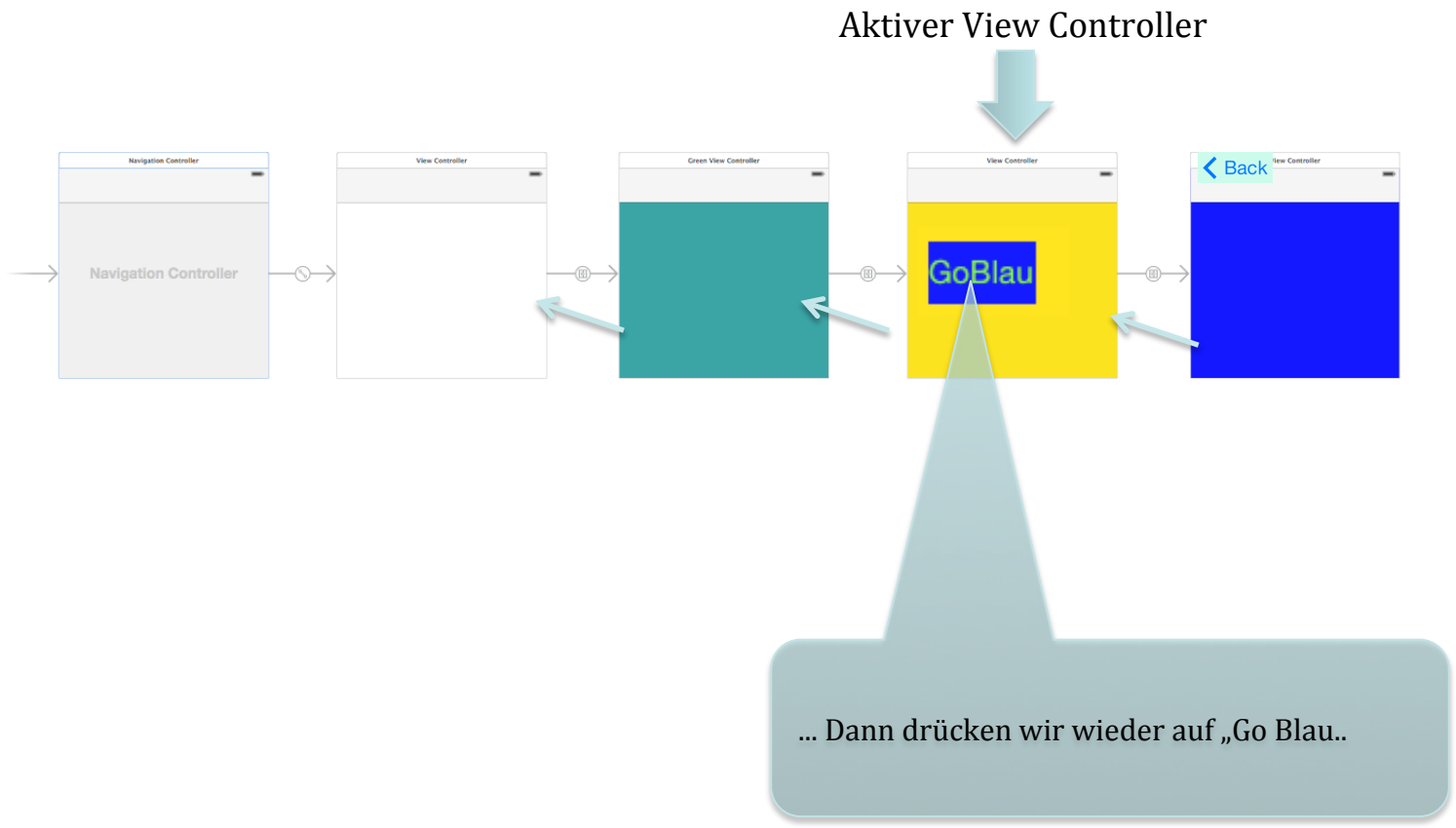


Unwind Segues

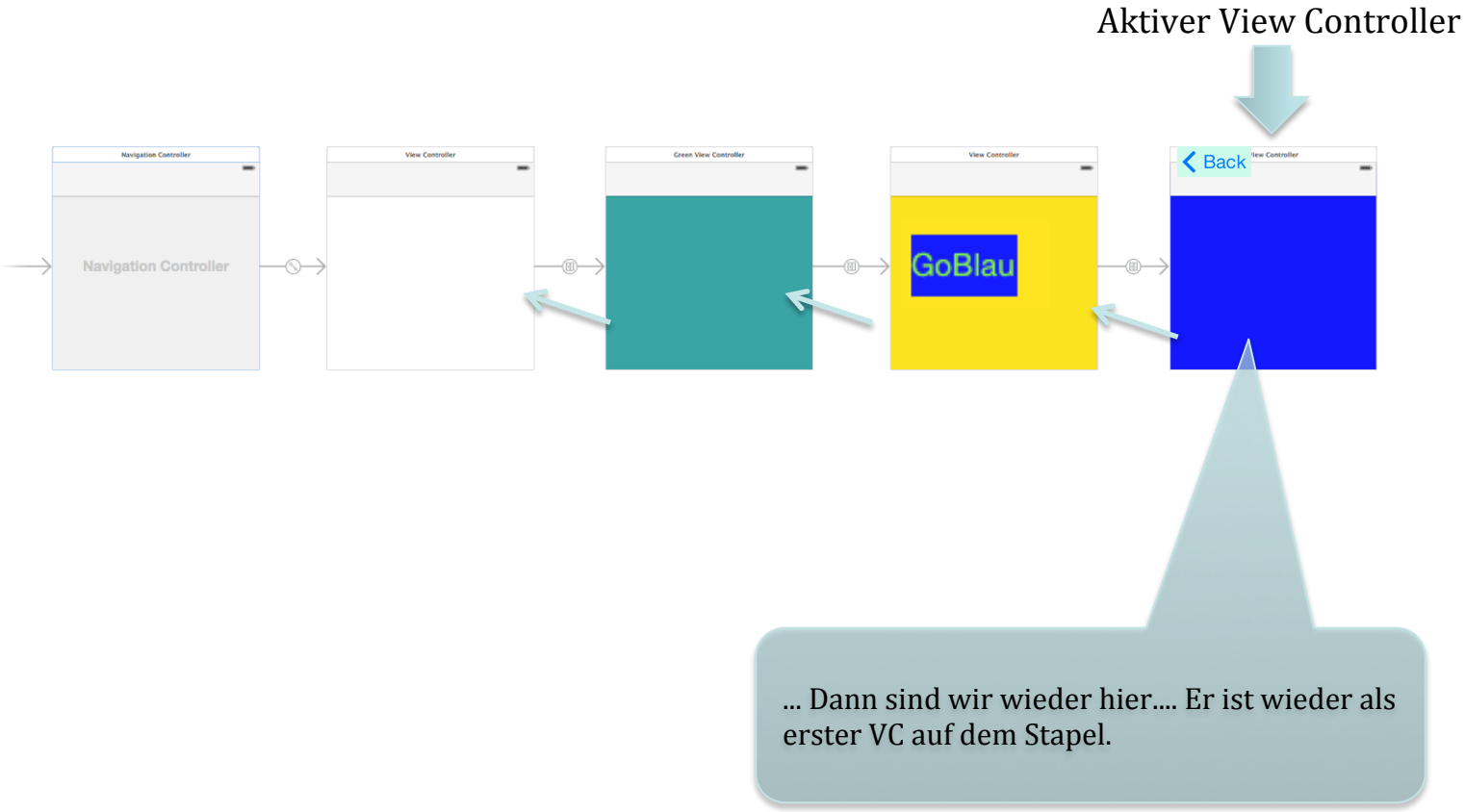


... Sind wir zurück bei dem gelbenViewController. Vom Stapel des NavigationController wurde der gelbe ViewController heruntergenommen.

Unwind Segues



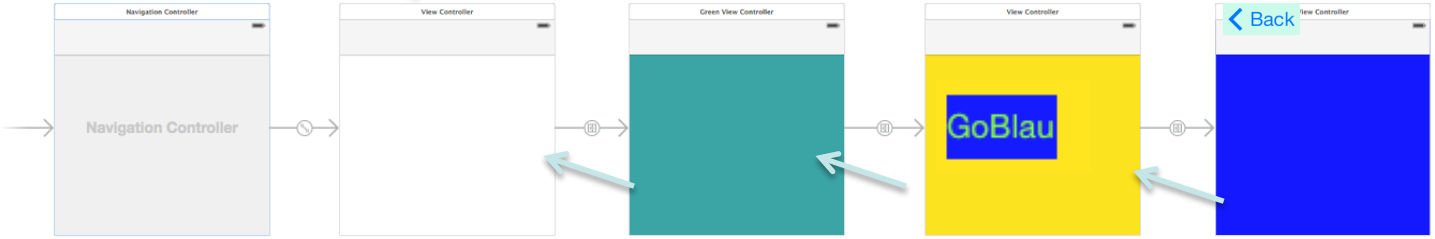
Unwind Segues



Unwind Segues

.. Aber jetzt wollen wir wieder zum weißen ViewController.
Mit einem Druckknopf.
Ohne dass wir 3x auf „Back“ drücken müssen.

Aktiver View Controller



Ein Segue vom blauenzum weißen ViewController wollen wir nicht, weil er noch mal den „Push“ aufrufen wurde. D. h., in diesem Fall würden wir mit „Back“ aus dem weißen ViewController zum blauen kommen. Wir wollen aber aus dem blauen VC zurück zum weißen, als hätten wir 3x auf „Back“ gedrückt. Wir wollen einfach schnell zurück und alles dazwischen vom Stack löschen.

Unwind Segues

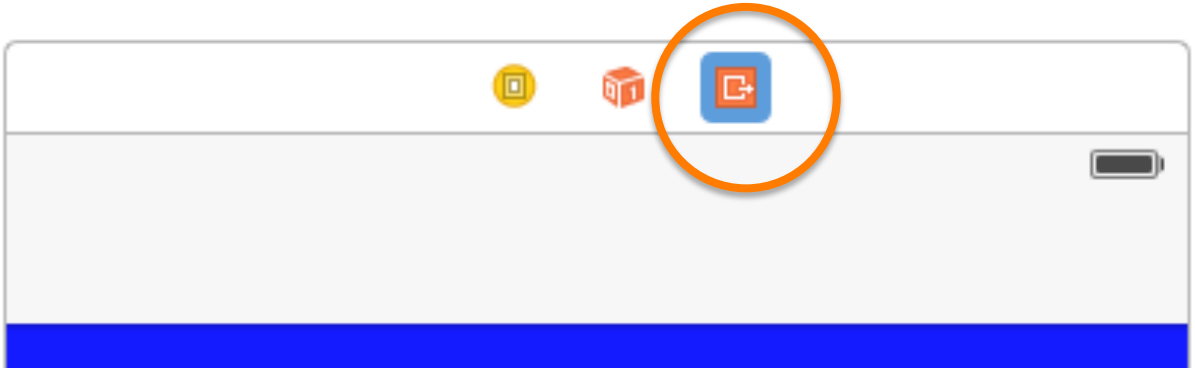
.. Aber jetzt wollen wir wieder
ViewController.
Mit einem Druckkn
Ohne dass wir

Aktiver View Controller

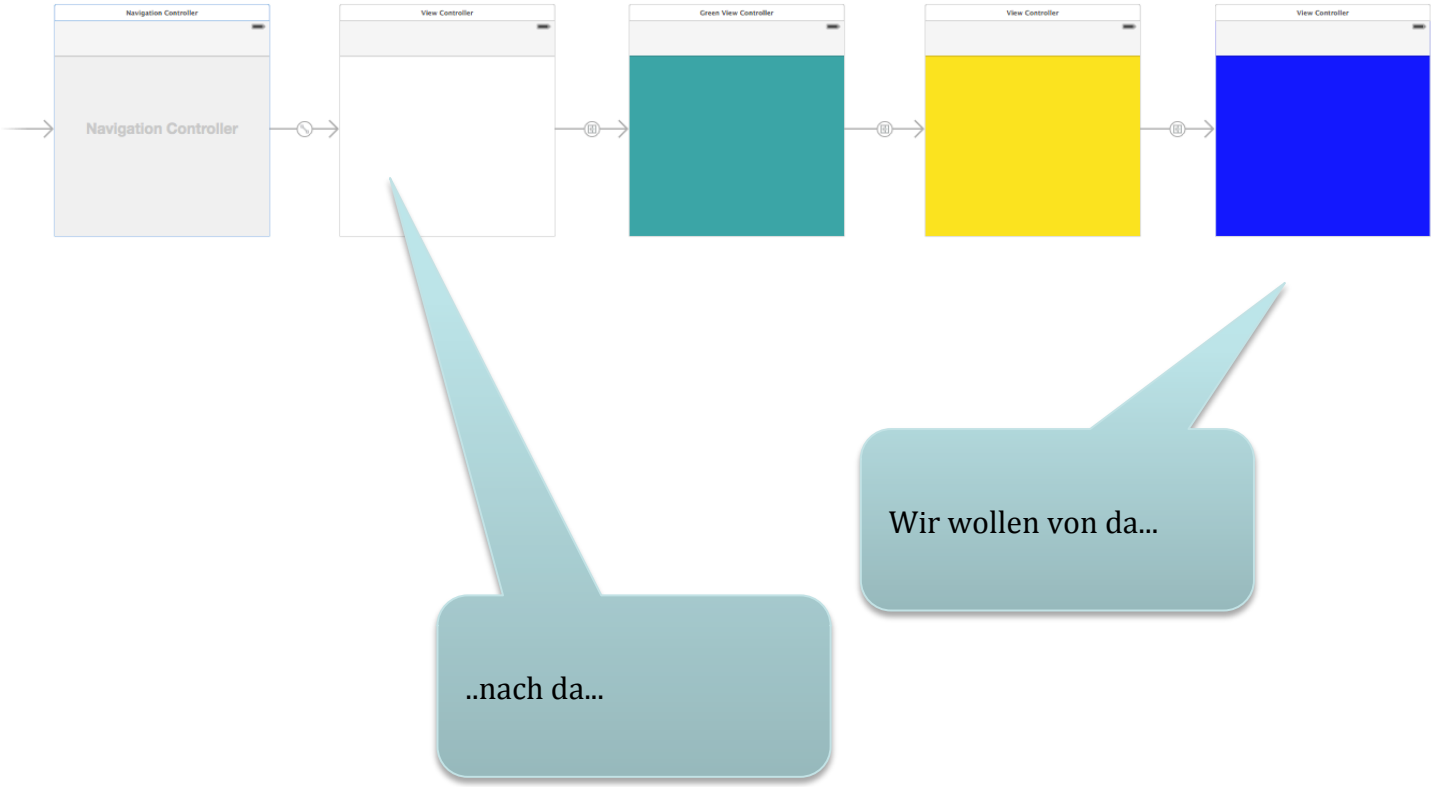
Lösung: Unwind Segues
Segues, die den „Weg zurück“ machen.
Im Storyboard: „Exit“

er noch mal den
dem weißen
VC zurück zum
nell zurück und alles

Exit



Exit, Unwind Segue



Exit, Unwind Segue

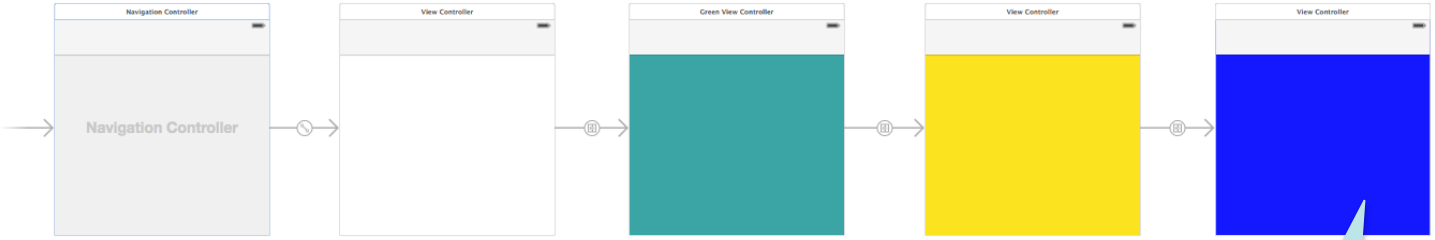
...dann müssen wir hier eine Methode realisieren, die ein Unwind-Segue empfängt.

```
37  @IBAction func ausBlau(segue: UIStoryboardSegue)
38  {
39
40
41  }
```

Das ist die Methode im weißen VC, die den Unwind-Segue empfängt.

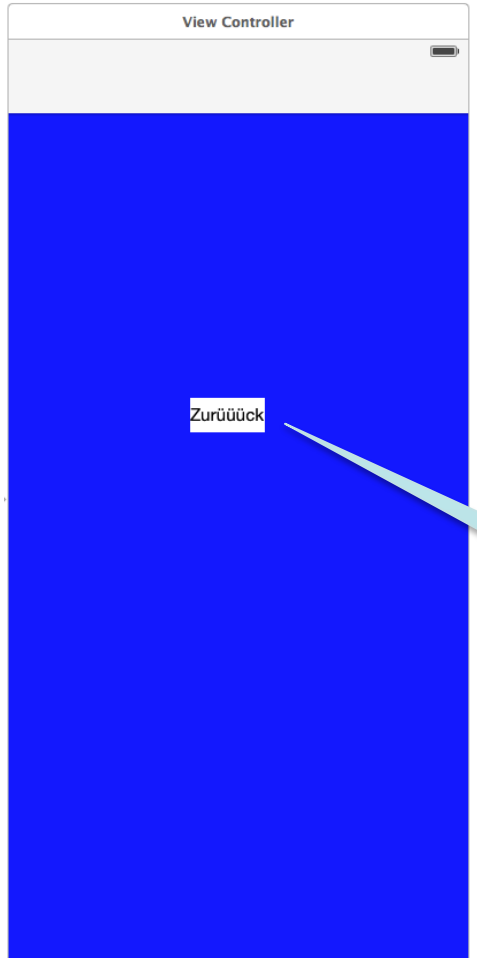
Diese Methode hat einen Parameter- ein Segue.....

Exit, Unwind Segue



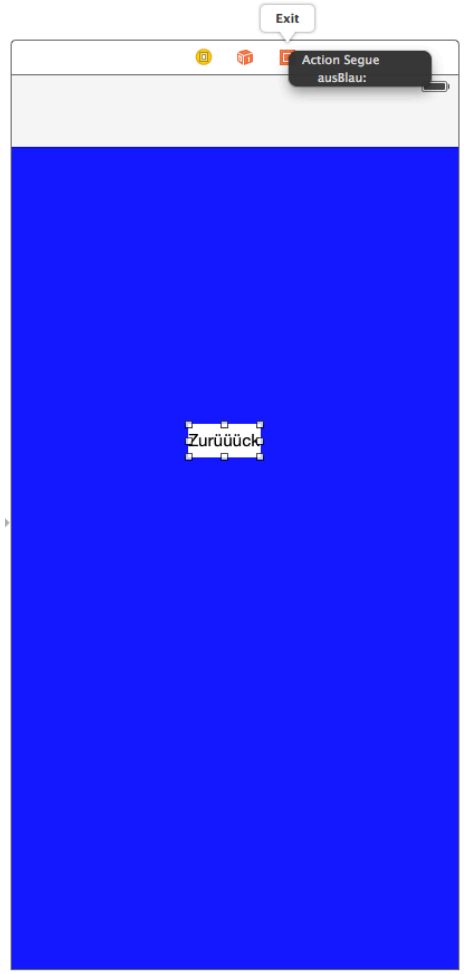
Von hier aus, soll ein Unwind-Segue aktiviert werden. Per Druckknopf.

Unwind Segue, Exit



Von hier aus, soll ein Unwind-Segue aktiviert werden. Per Druckknopf „Zurüüück“.

Unwind Segue, Exit



Wir verbinden den Knopf mit „Exit“ und wählen die Aktion „ausBlau“ aus. Diese ist im weißen VC. Nach dem Knopfdruck wird der weiße VC aktiviert. Der ganze Weg zurück, vom gelben VC zum weißen wird vom Stack gelöscht, als hätten wir 3x den „Back“-Button betätigt. Anschließend wird die Aktion

ausBlau

ausgeführt. Diese ist bei uns leer.

Der Lebenszyklus des Views

Methoden, die automatisch aufgerufen werden, abhängig vom Ereignis des Views (geladen, erschienen, kurz vor dem Erscheinen, kurz vor dem Verschwinden...) Bis jetzt hatten wir die Methode viewDidAppear: verwendet.

```
viewDidLoad
viewWillAppear
viewWillDisappear
viewDidAppear
viewDidDisappear
```

- ...

Fertig

- MAD8 fertig