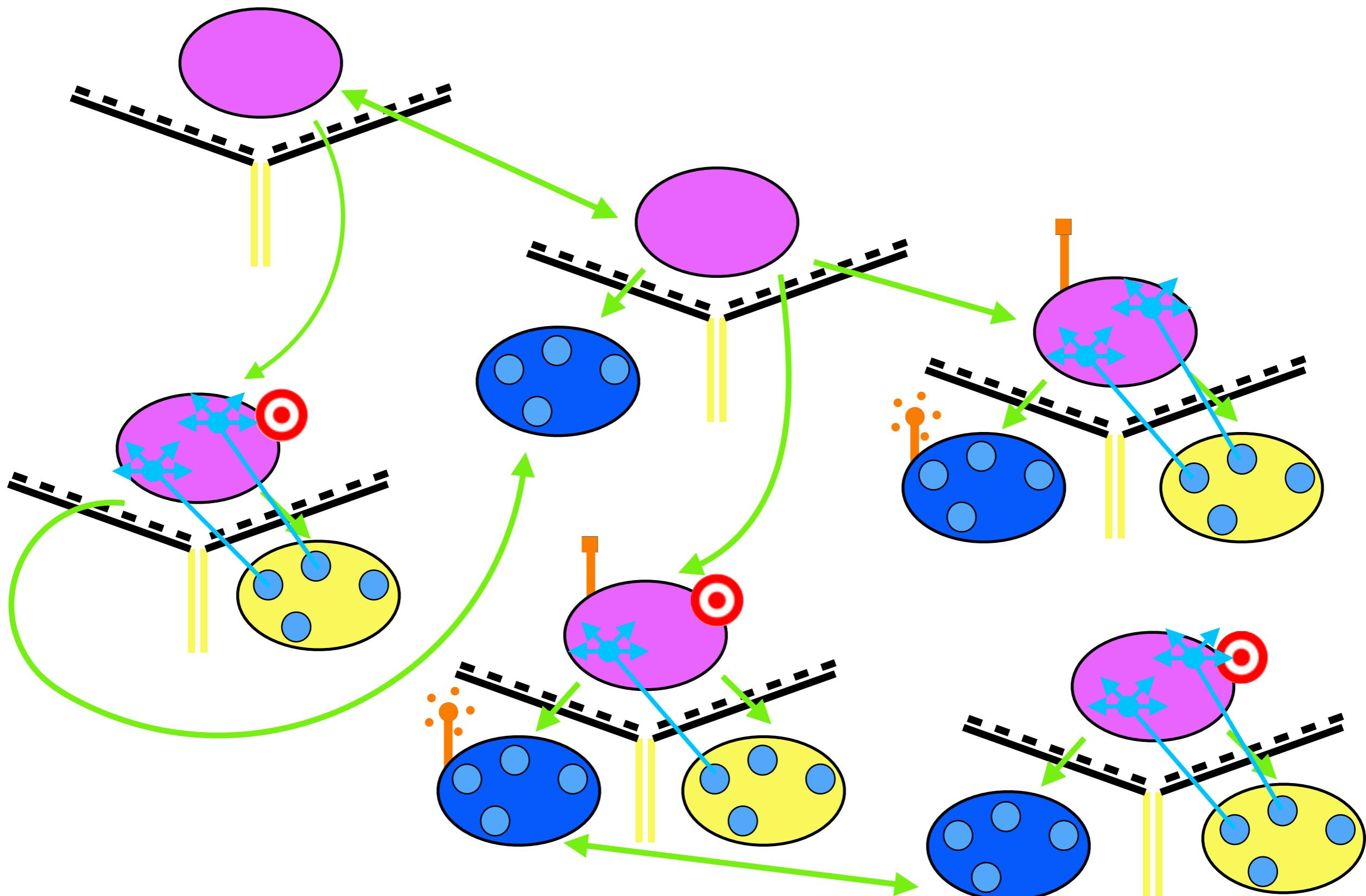


App-Entwicklung für iOS und OS X

SS 2016
Stephan Gimbel



MVCs arbeiten zusammen

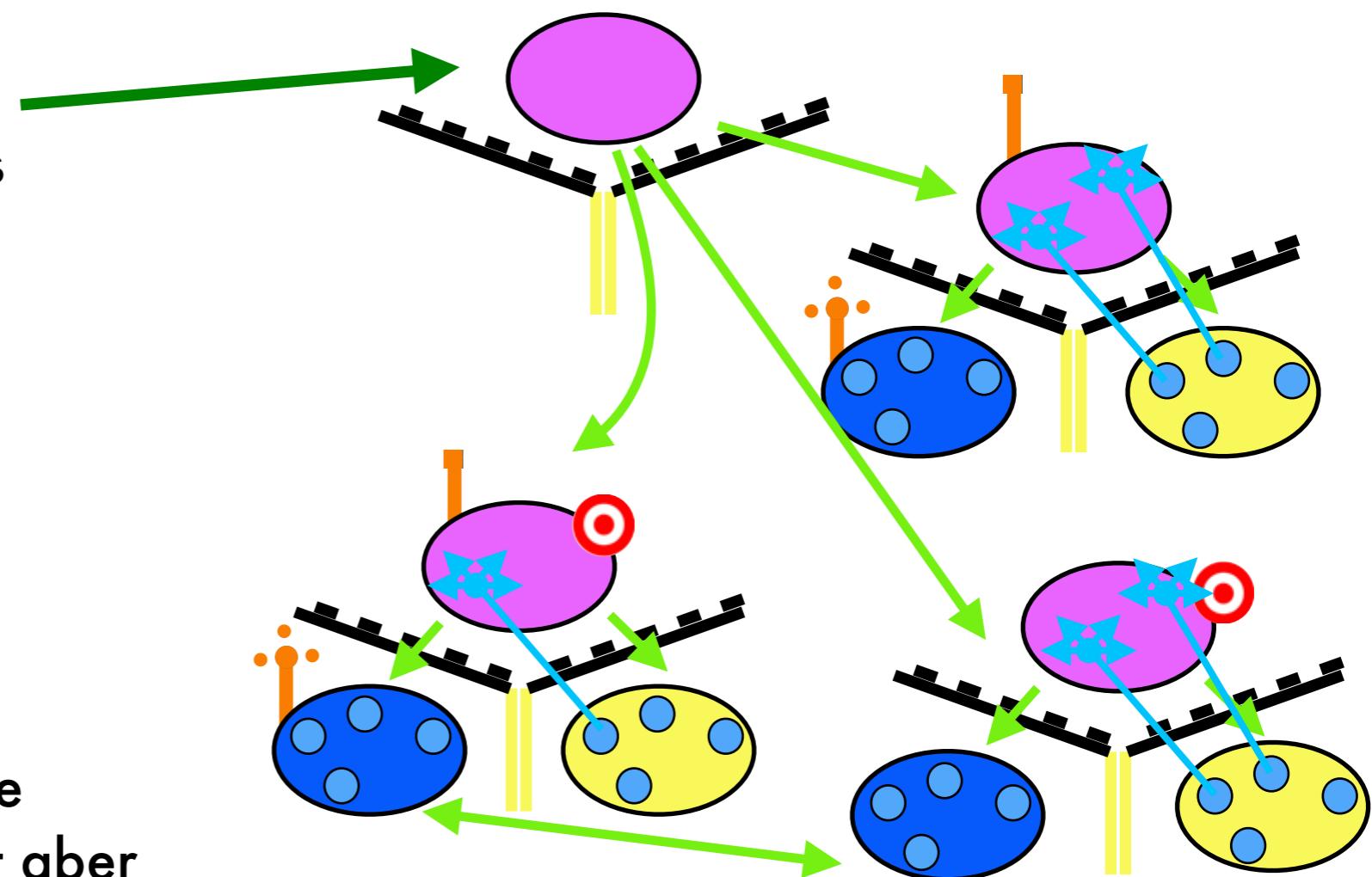


MVCs arbeiten zusammen

Es ist an der Zeit umfangreichere Applikationen zu entwickeln
Hierfür müssen wir MVCs kombinieren...

iOS stellt einige Controller
zur Verfügung deren Views
andere MVCs sind. *

* wir können dafür auch unsere
eigenen Controller erstellen, ist aber
nicht Teil des Kurses.

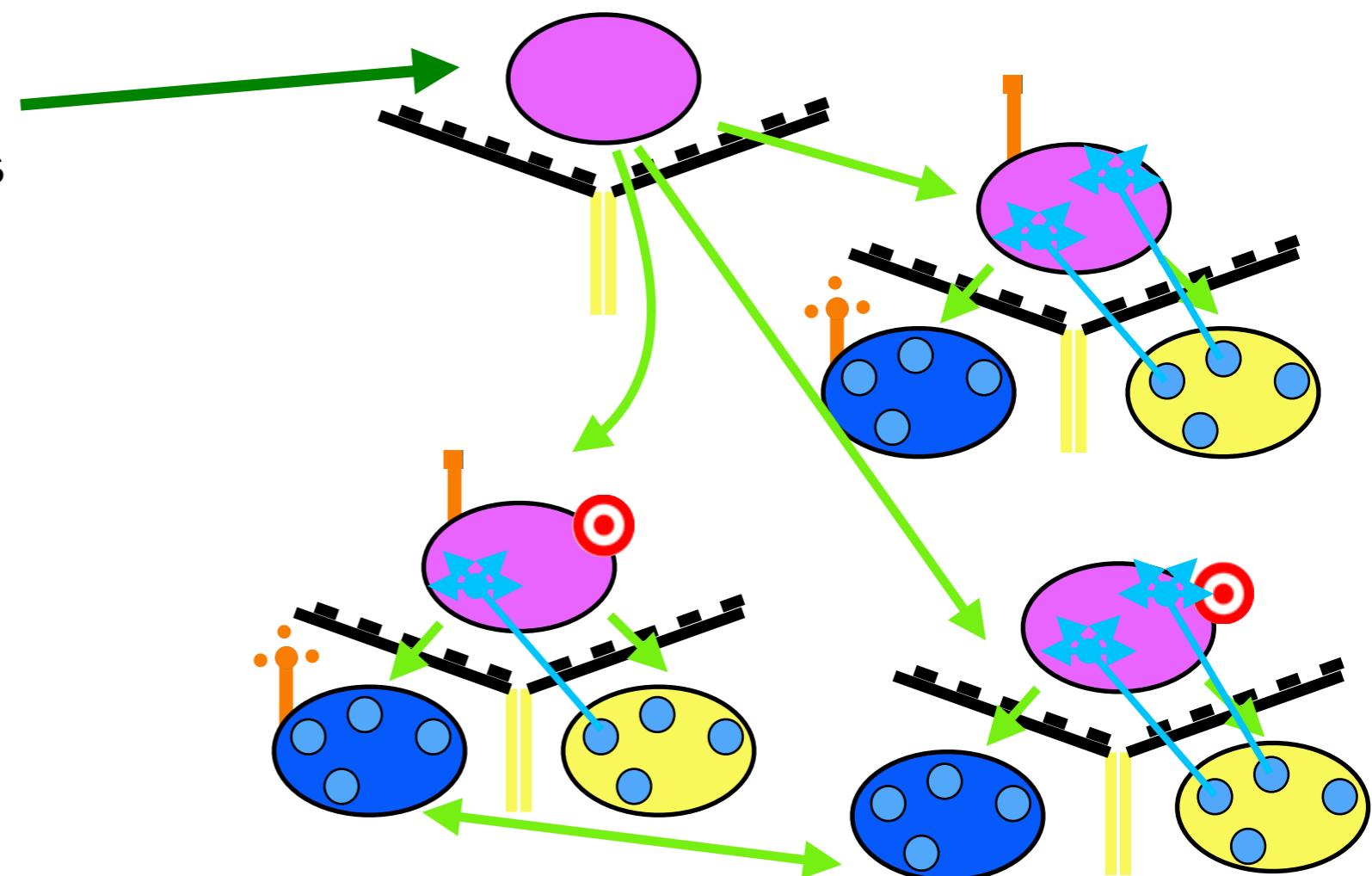


MVCs arbeiten zusammen

Es ist an der Zeit umfangreichere Applikationen zu entwickeln
Hierfür müssen wir MVCs kombinieren...

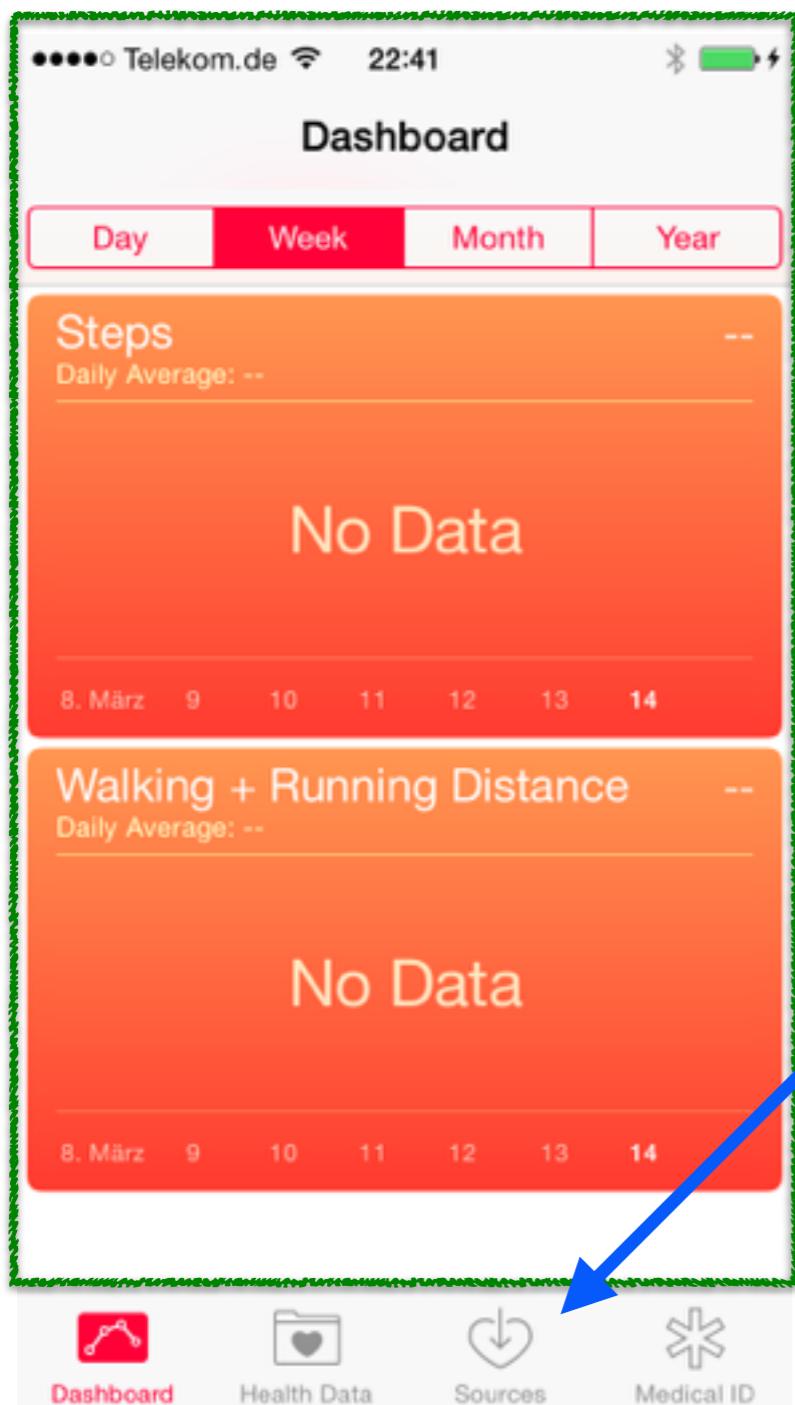
iOS stellt einige Controller
zur Verfügung deren Views
andere MVCs sind.

Beispiele:
UITabBarController
UISplitViewController
UINavigationController



UITabBarController

Lässt den User zwischen verschiedenen MVCs wählen...



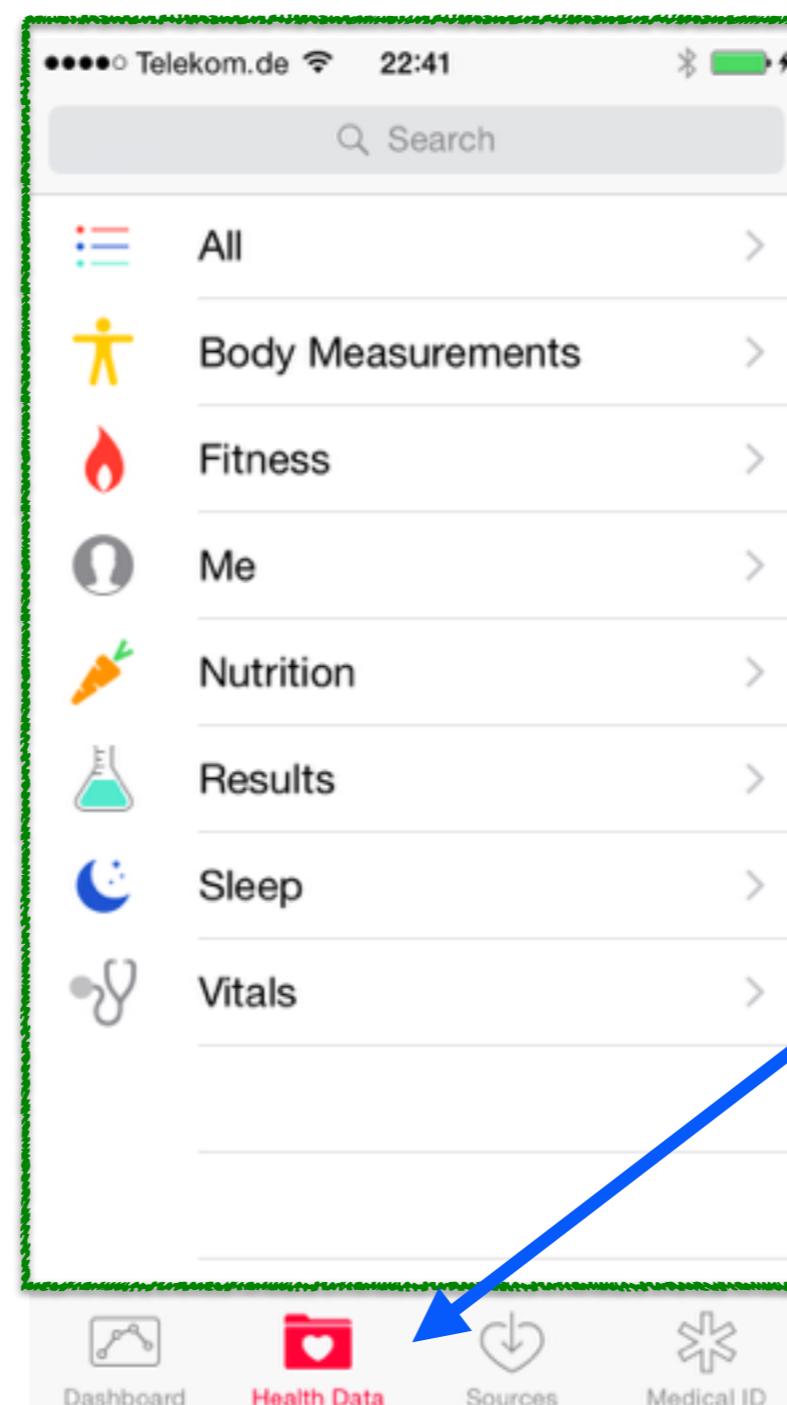
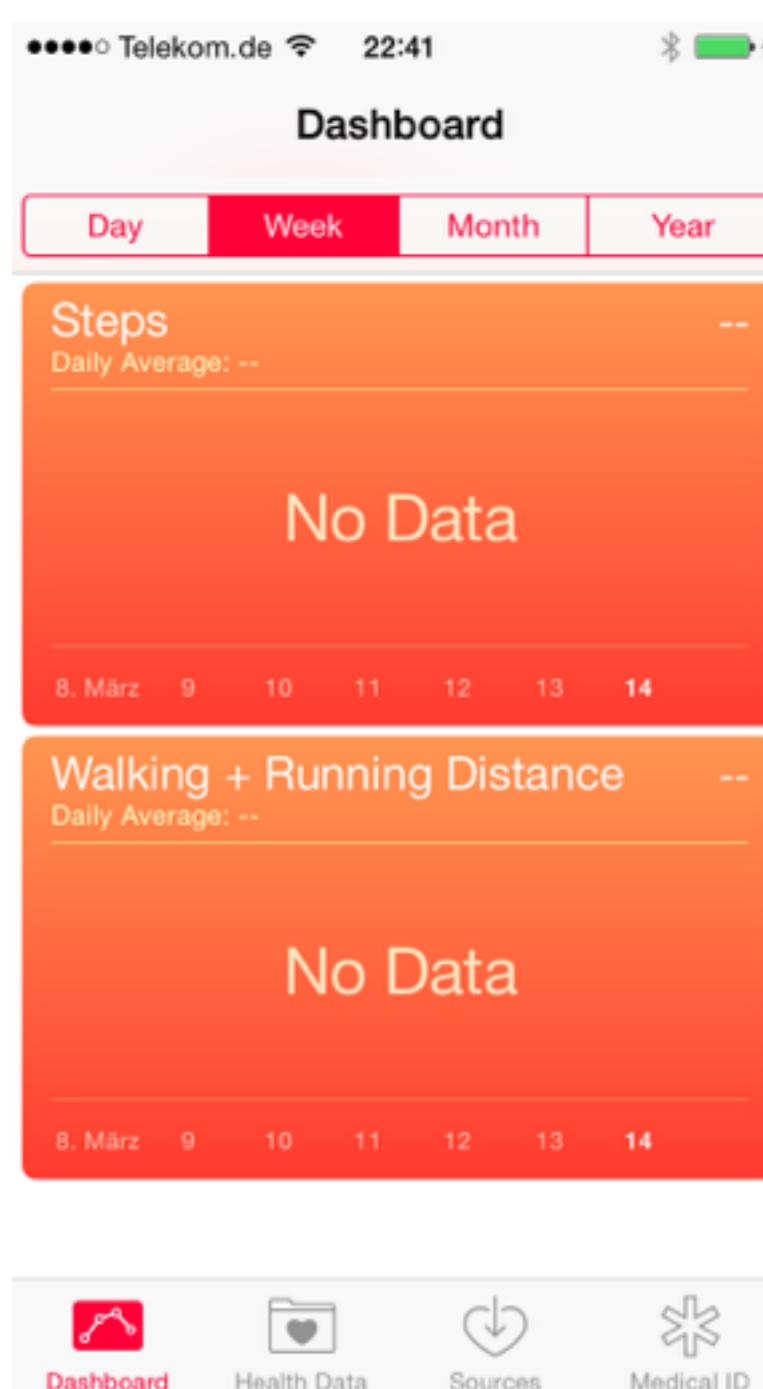
“Dashboard” MVC

Icon, Titel und ein “Badge Value” wird durch die
MVCs selbst bestimmt über deren Property:
`var tabBarItem: UITabBarItem!`

Normalerweise setzen wir dies im Storyboard.

UITabBarController

Lässt den User zwischen verschiedenen MVCs wählen...

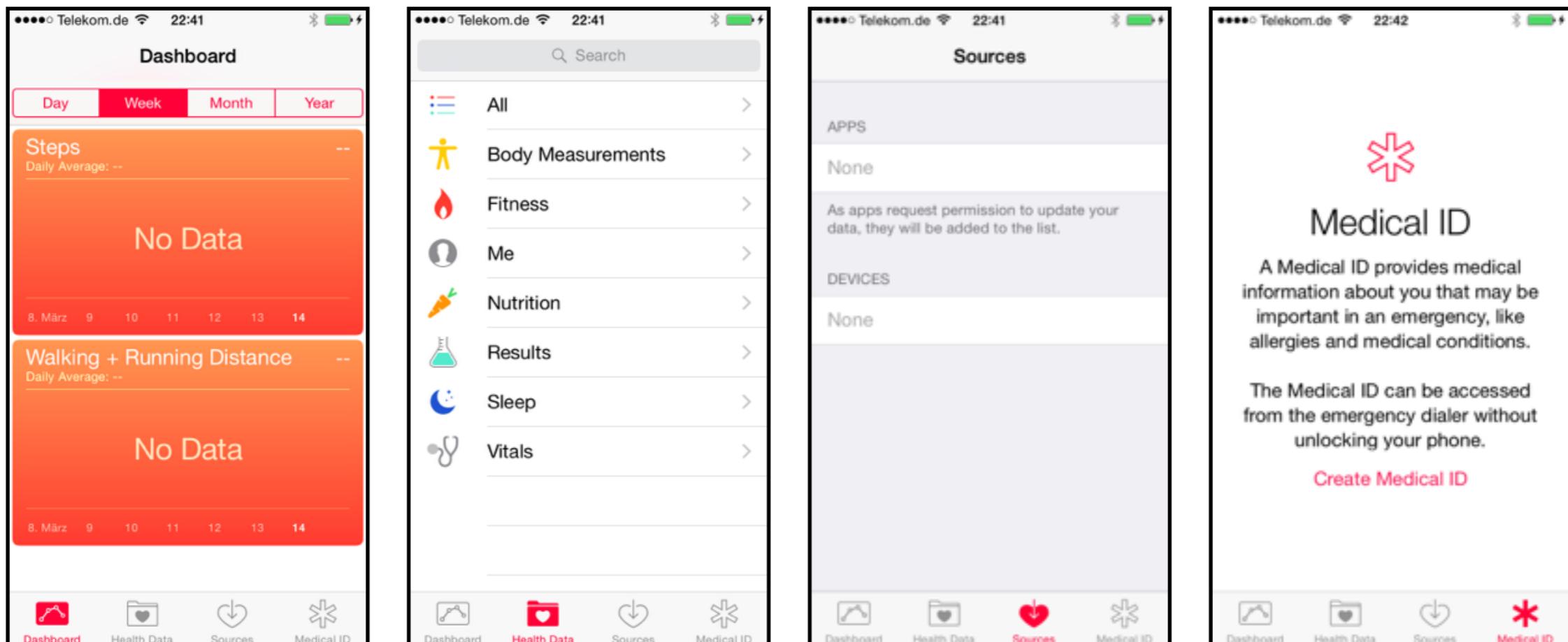


“Health Data” MVC

Bei zu vielen Tabs zum anzeigen, zeigt der UITabBarController automatisch ein UI um den Überschuss zu managen

UITabBarController

Lässt den User zwischen verschiedenen MVCs wählen...

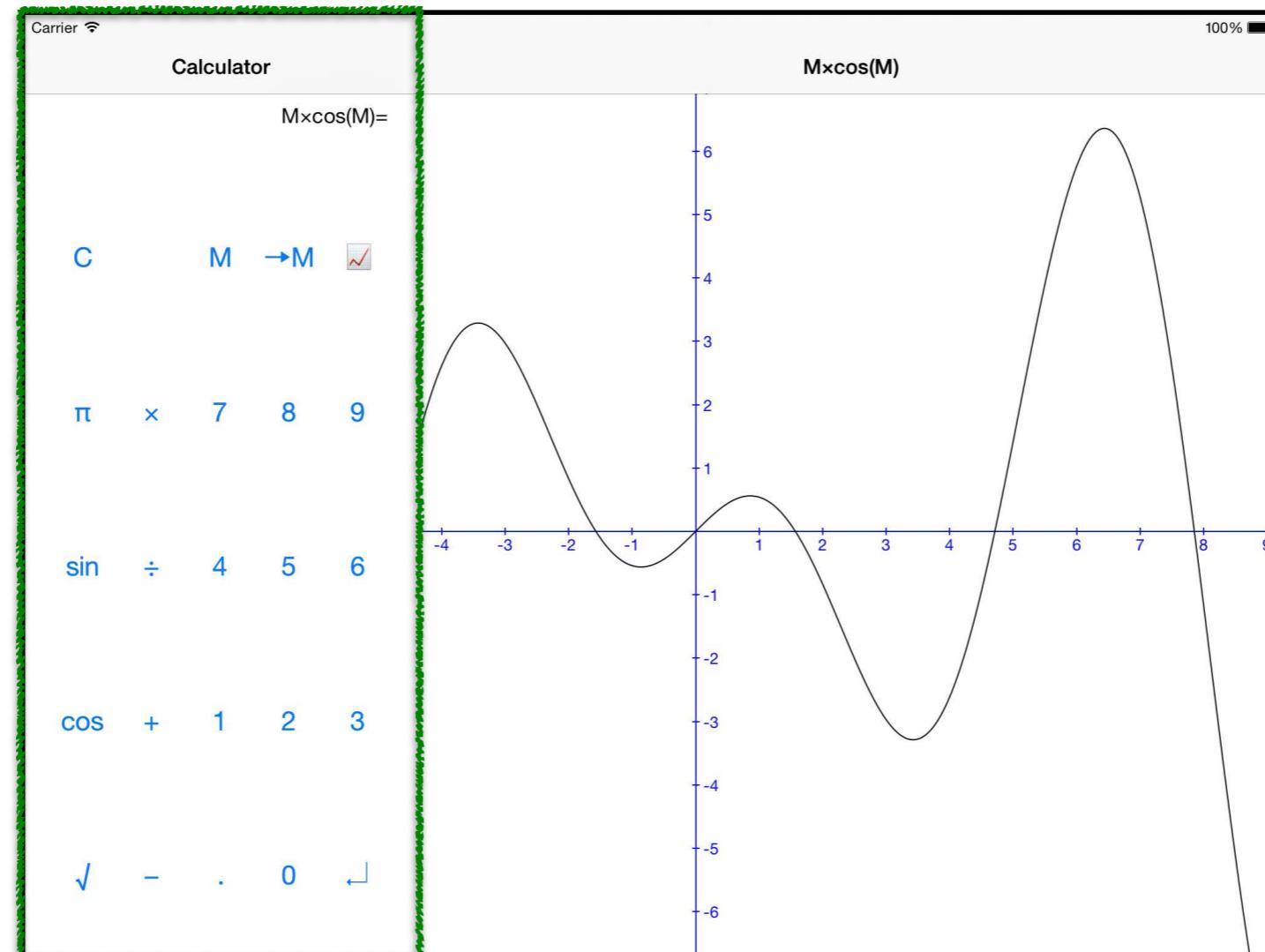


UISplitViewController

Zwei MVCs nebeneinander...

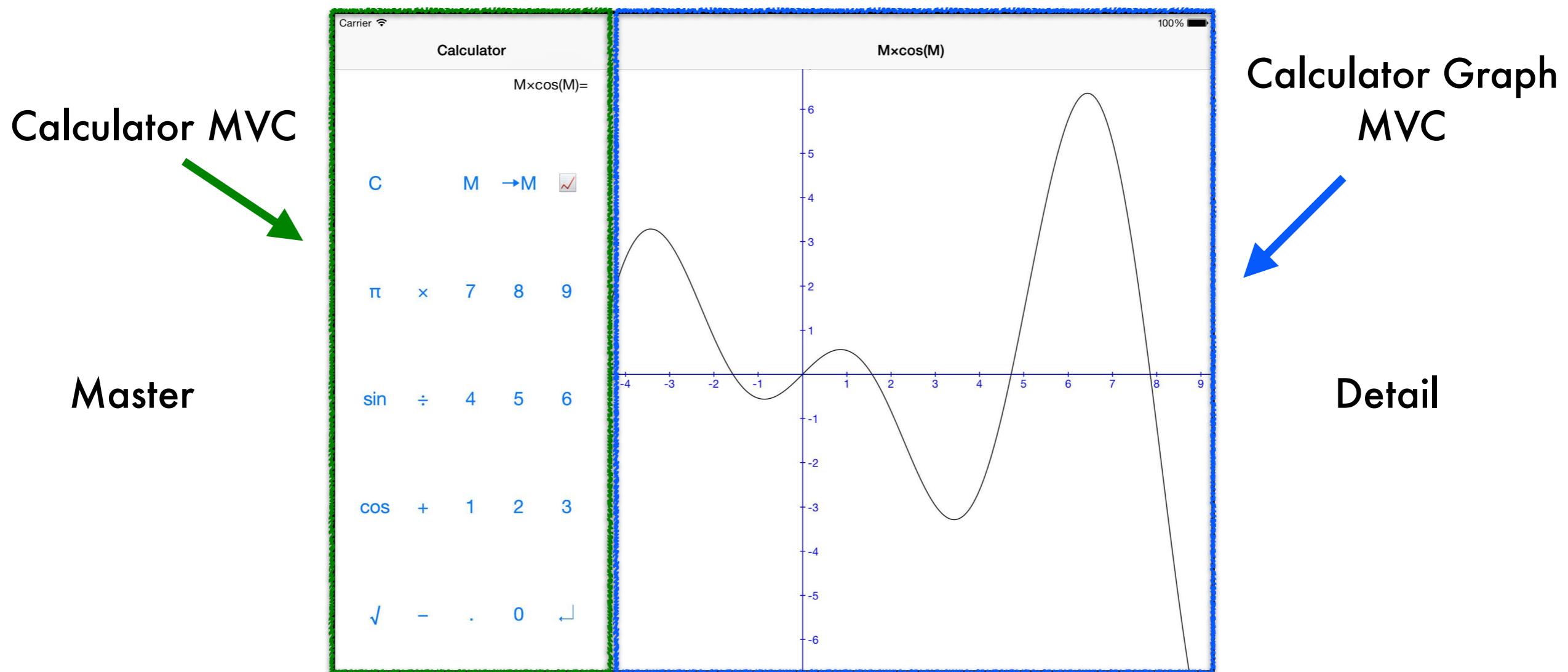
Calculator MVC

Master



UISplitViewController

Zwei MVCs nebeneinander...

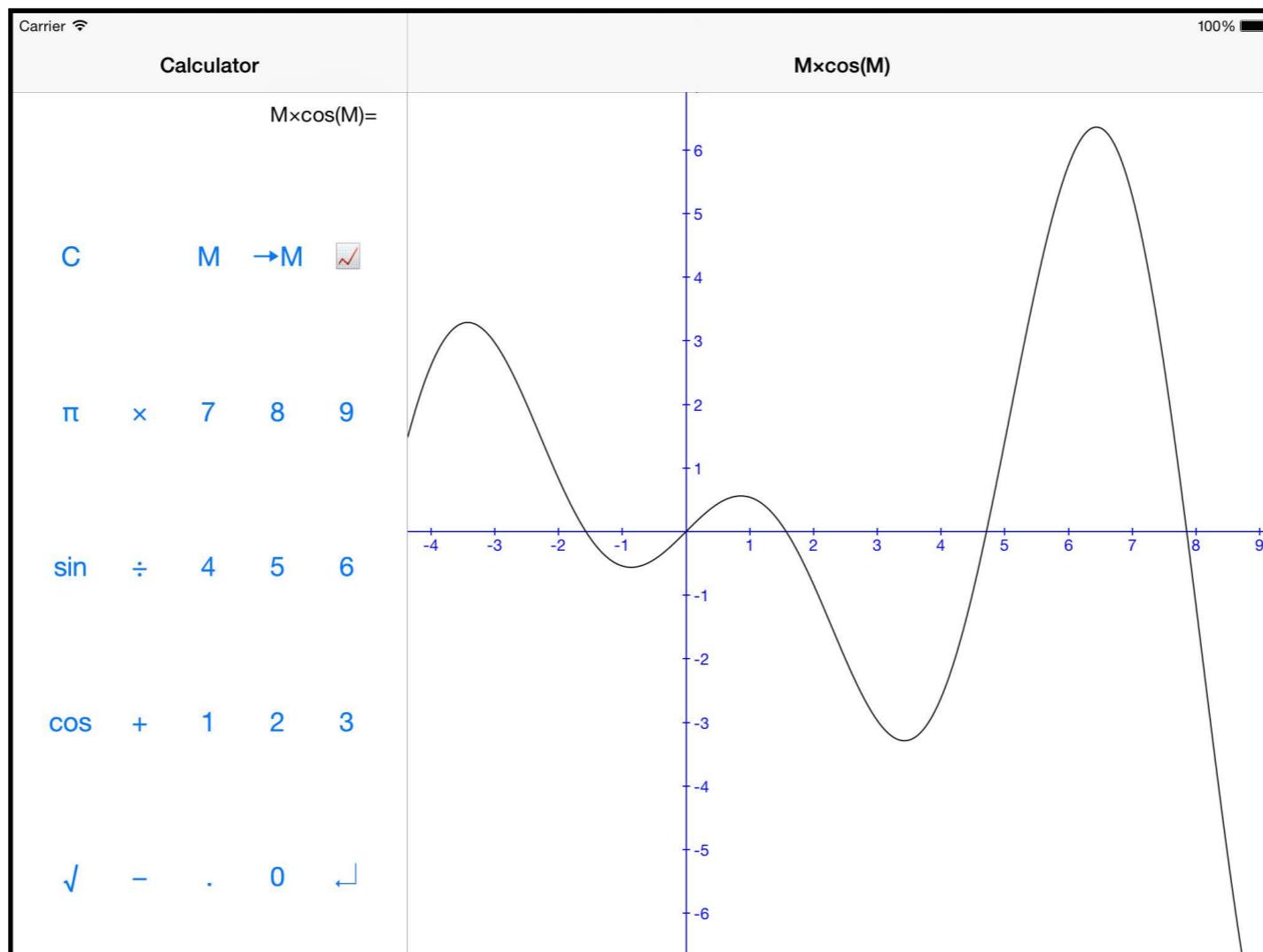


UISplitViewController

Zwei MVCs nebeneinander...

Calculator MVC

Master



Calculator Graph
MVC

Detail

UISplitViewController

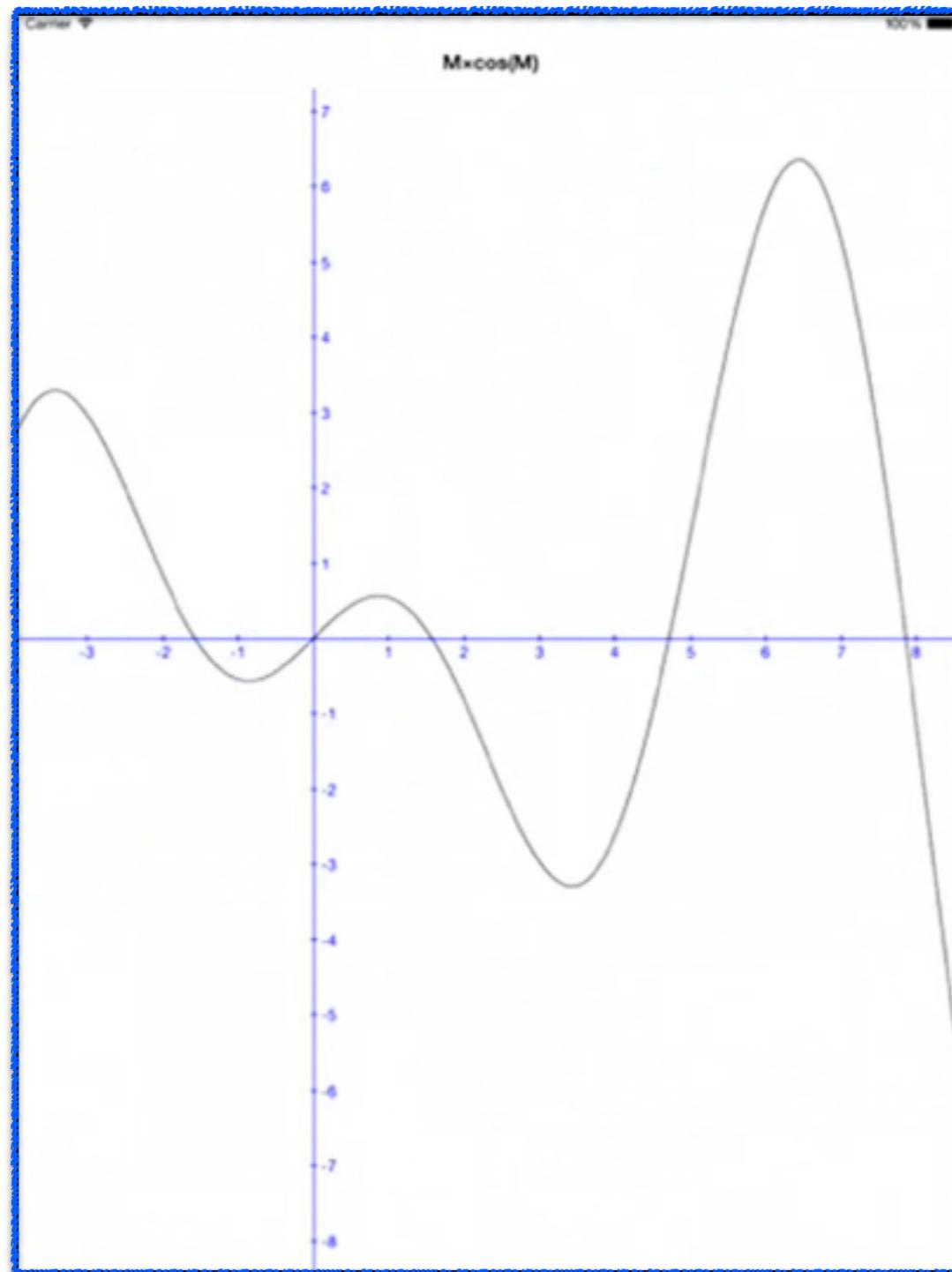
Zwei MVCs nebeneinander...

Calculator MVC

Master

Calculator Graph
MVC

Detail



UISplitViewController

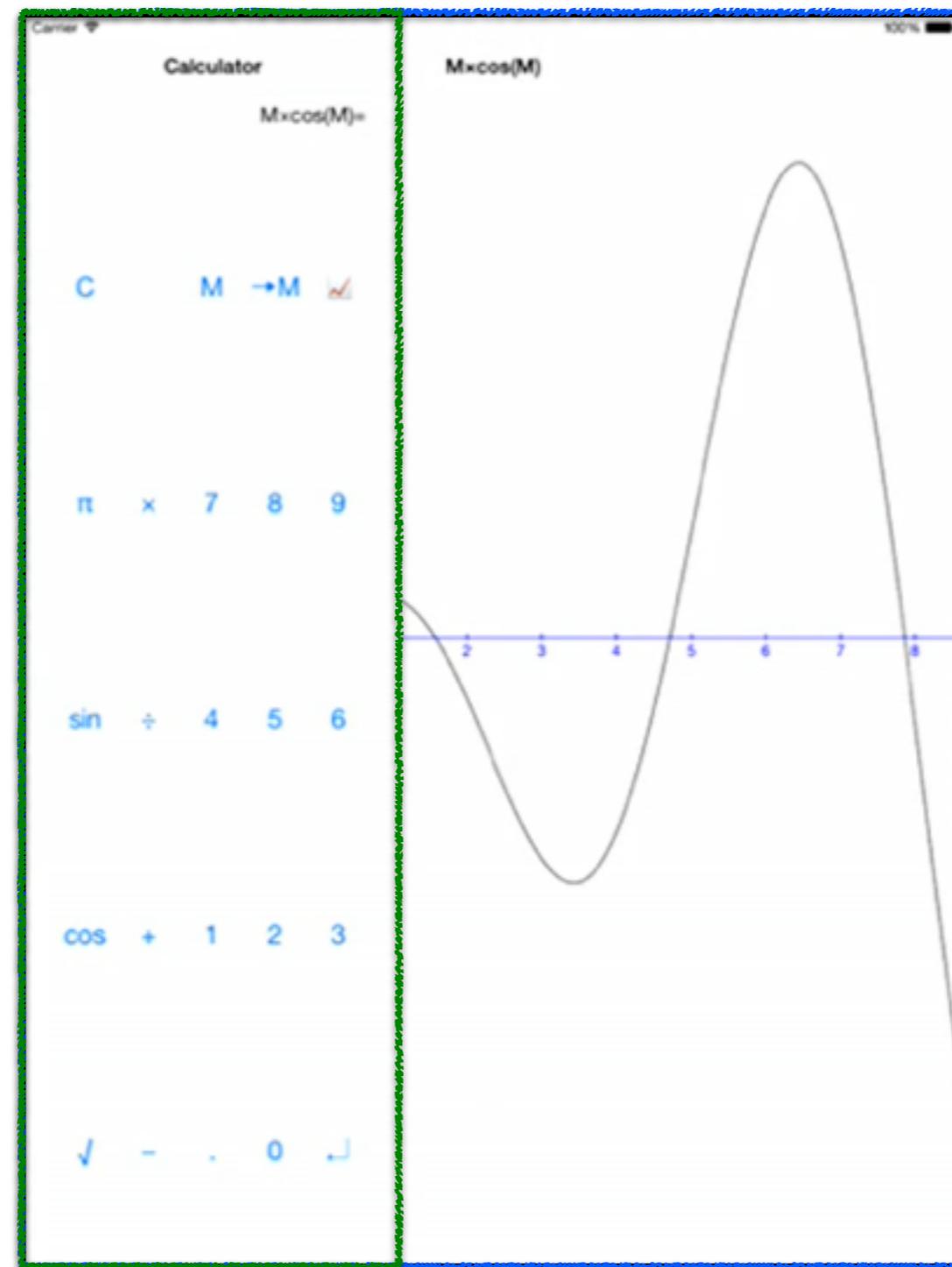
Zwei MVCs nebeneinander...

Calculator MVC

Master

Calculator Graph
MVC

Detail



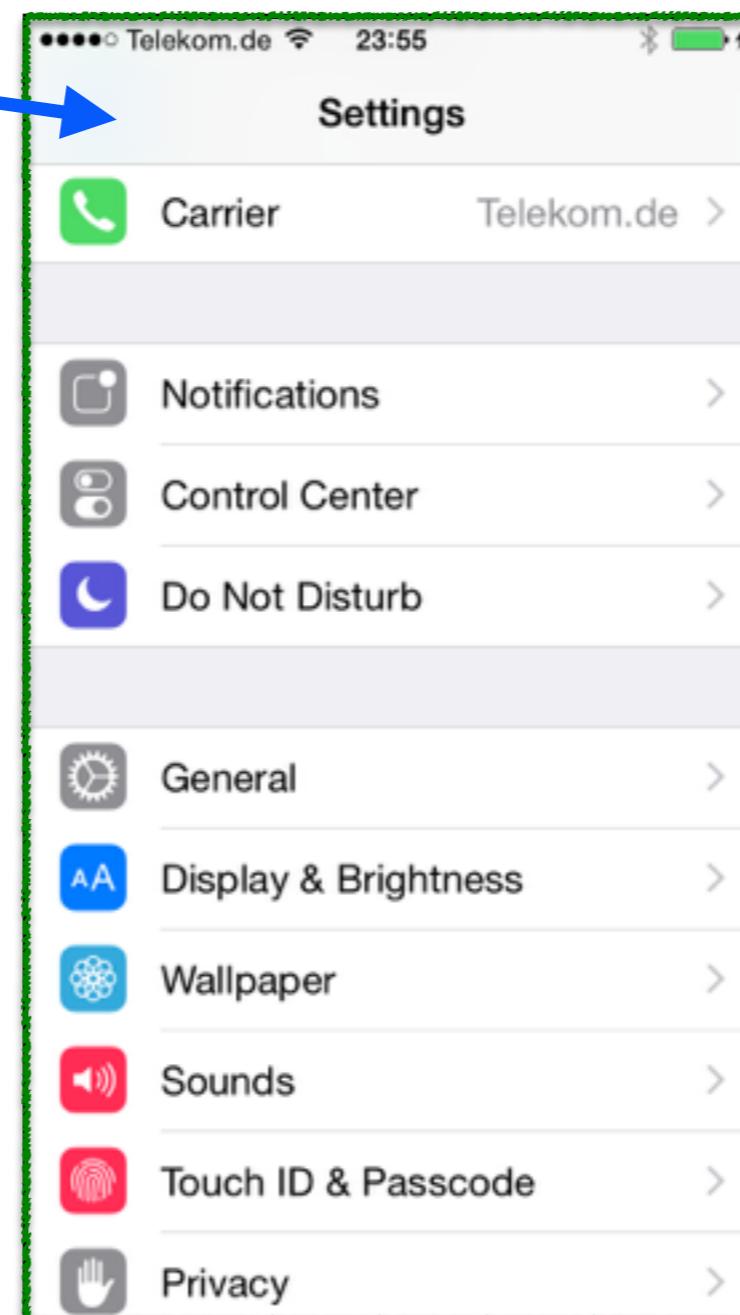
UINavigationController

Push und Pop von MVCs vom Stack (wie ein Stapel Karten)...

Dieser Bereich wird vom UINavigationController gezeichnet.

Aber der Inhalt des oberen Bereiches (wie der Titel und andere Buttons) werden vom aktuell gezeigten MVC vorgegeben (in diesem Fall der "Settings" MVC).

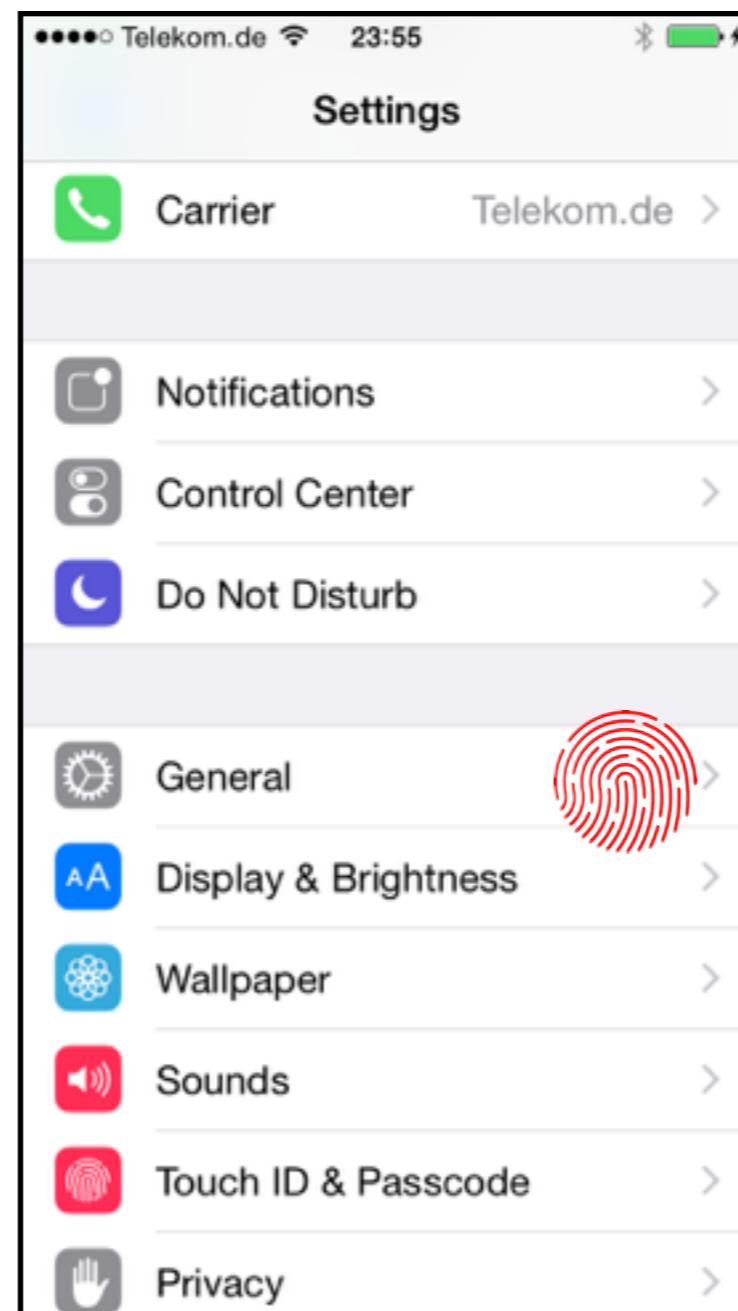
Jeder MVC kommuniziert diesen Inhalt über das UIViewController `navigationItem` Property.



"Settings" MVC

UINavigationController

Push und Pop von MVCs vom Stack (wie ein Stapel Karten)...



UINavigationController

Push und Pop von MVCs vom Stack (wie ein Stapel Karten)...



Es ist möglich hier MVC-spezifische Buttons hinzu zu fügen mittels dem `toolbarItems` Property dies UIViewControllers. →

“General Settings” MVC

UINavigationController

Push und Pop von MVCs vom Stack (wie ein Stapel Karten)...



Beachten dass hier ein “Back”-Button aufgetaucht ist. Dies geschieht automatisch durch den UINavigationController.

“General Settings” MVC

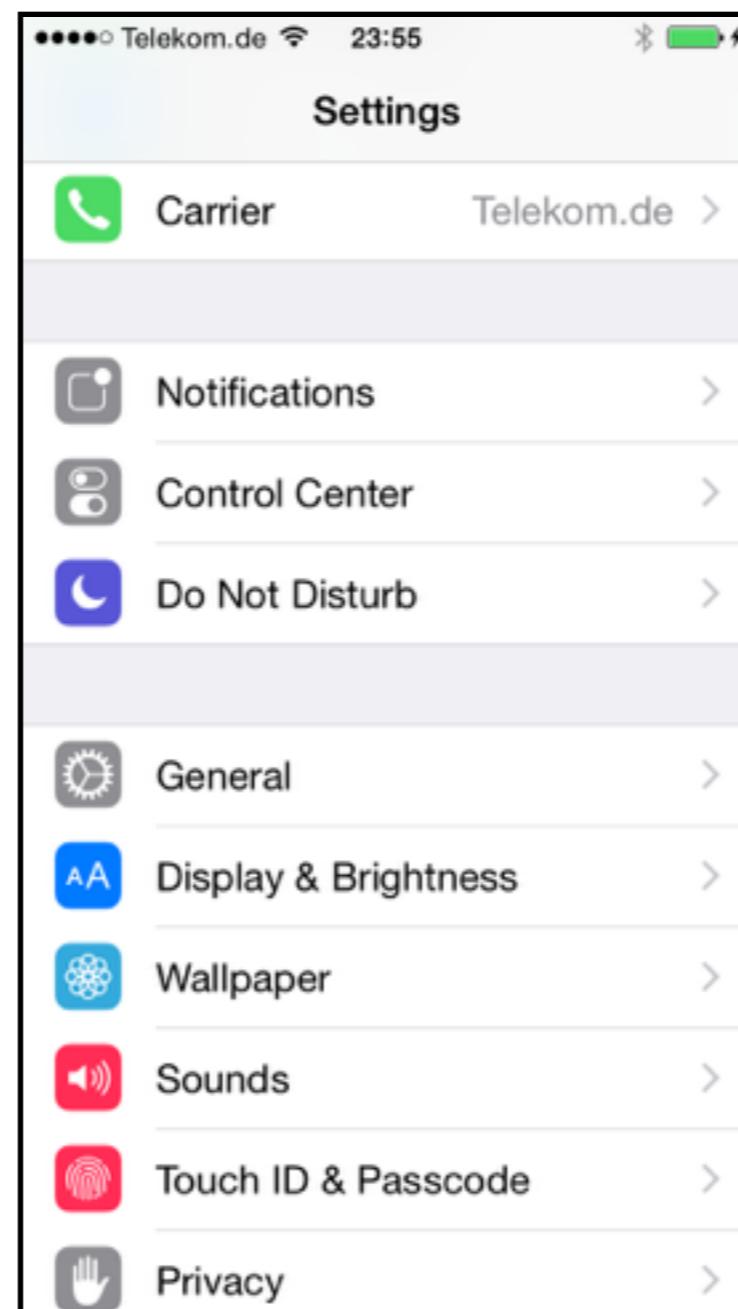
UINavigationController

Push und Pop von MVCs vom Stack (wie ein Stapel Karten)...



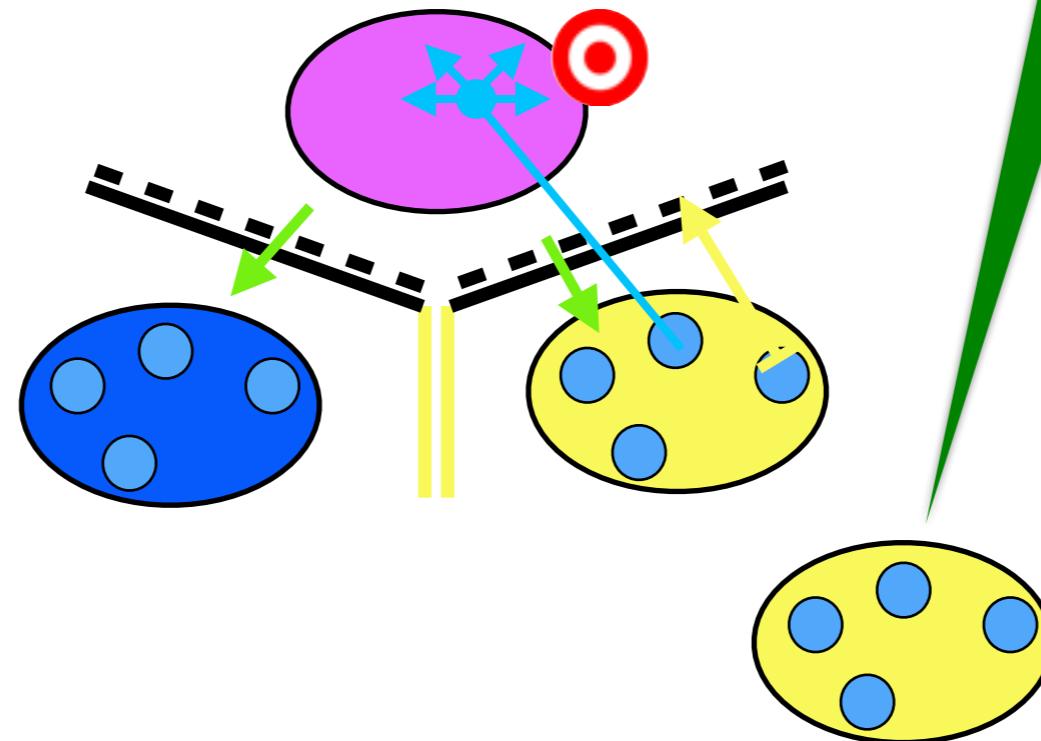
UINavigationController

Push und Pop von MVCs vom Stack (wie ein Stapel Karten)...



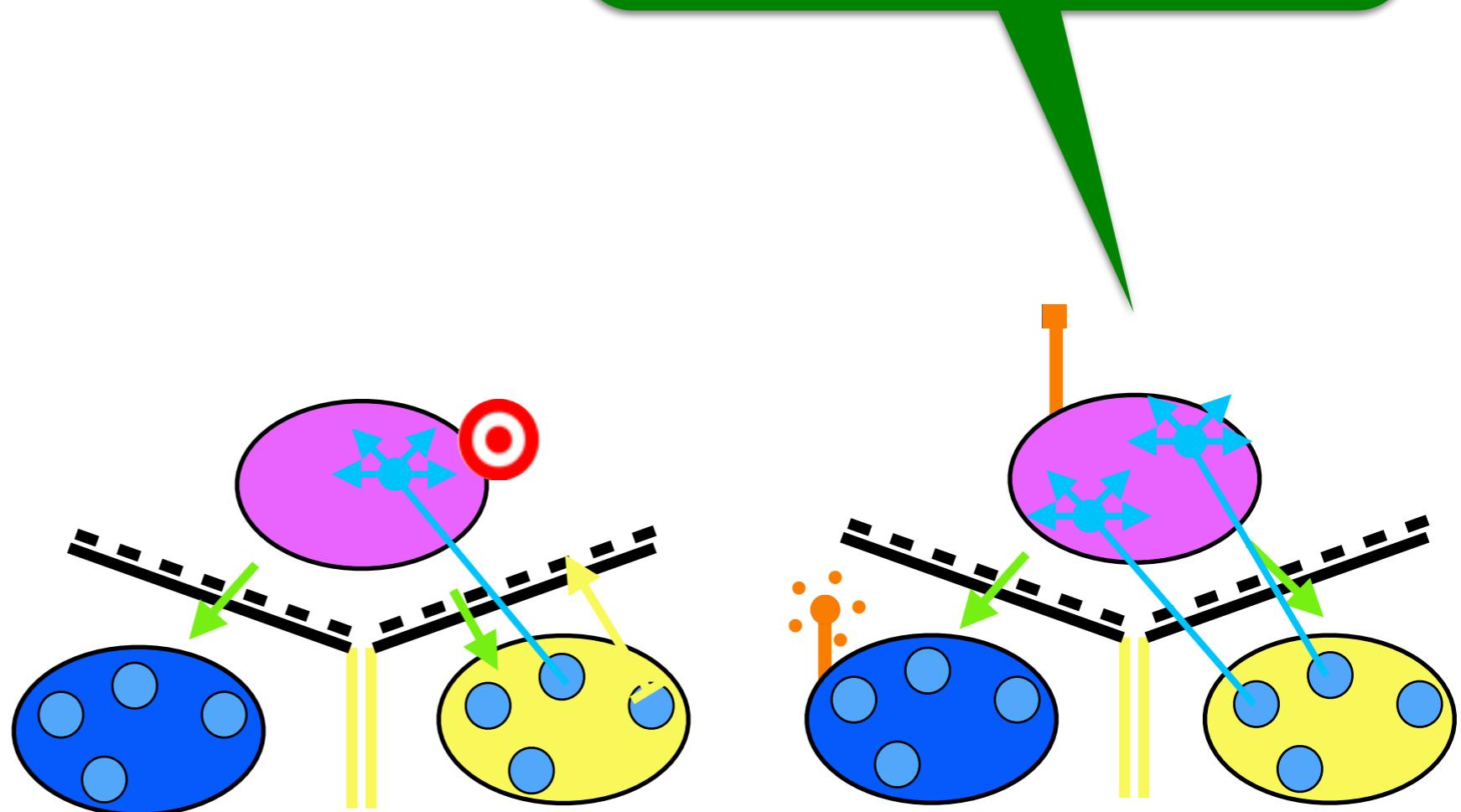
UINavigationController

Wir möchten mehr Features, es macht aber keinen Sinn alles in einen MVC zu packen.



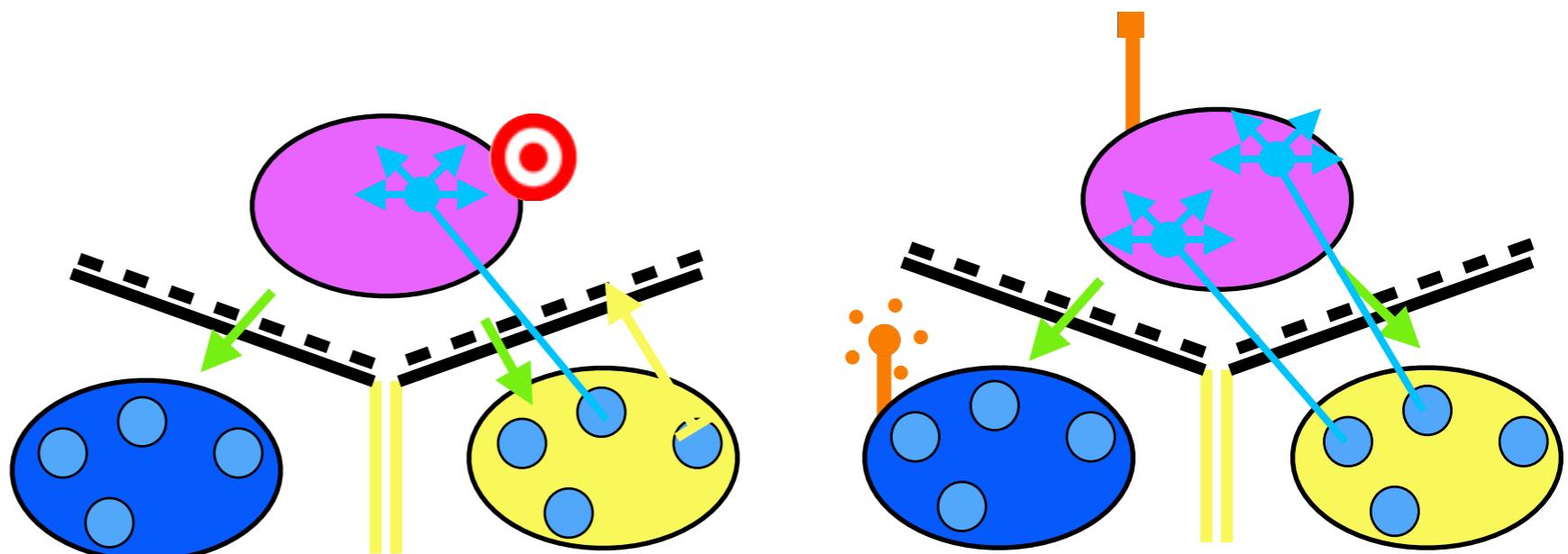
UINavigationController

Also erstellen wir einen neuen MVC um diese Funktionalität zu kapseln.

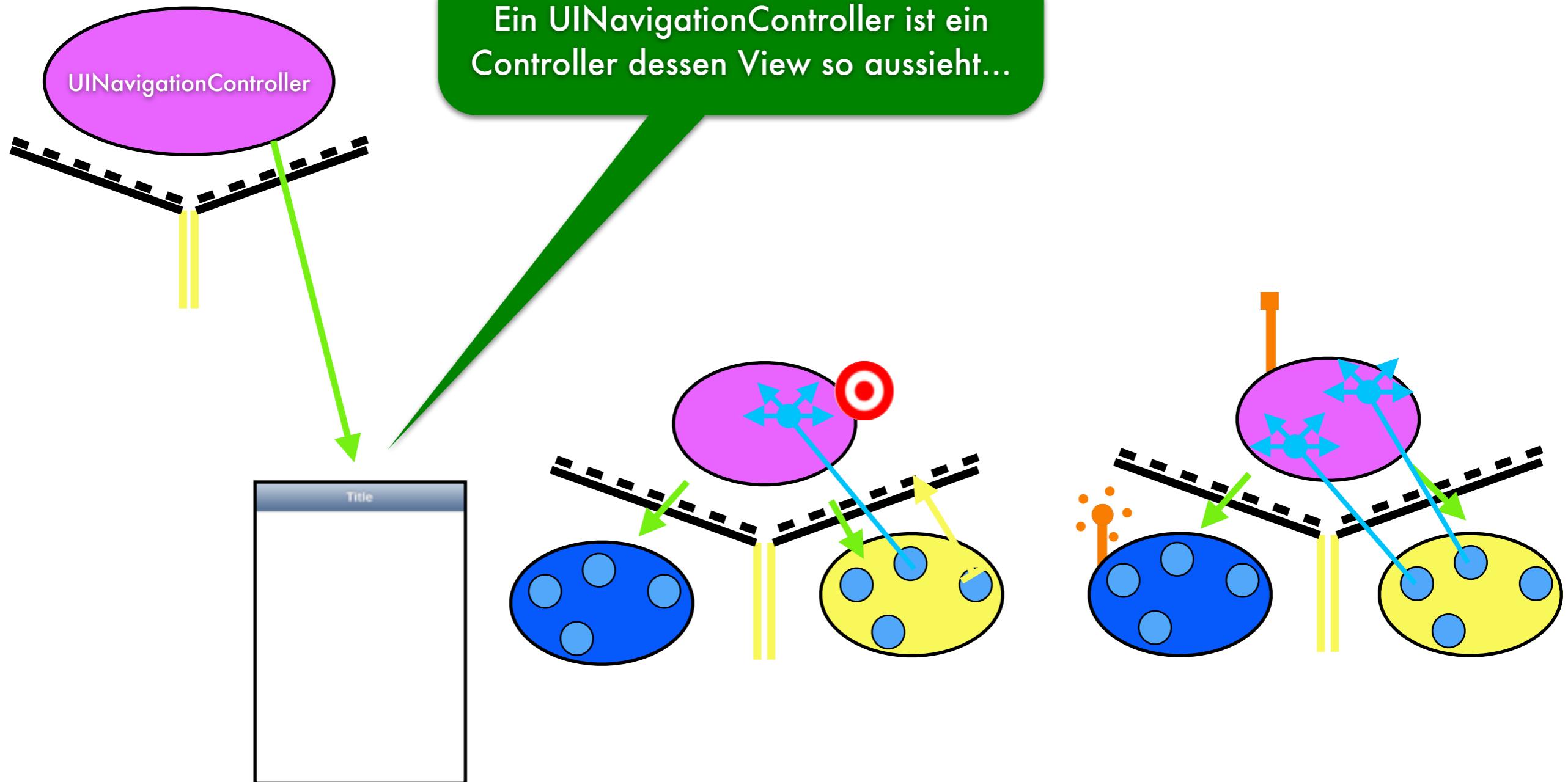


UINavigationController

Wir können einen UINavigationController nutzen um den Screen zu teilen.

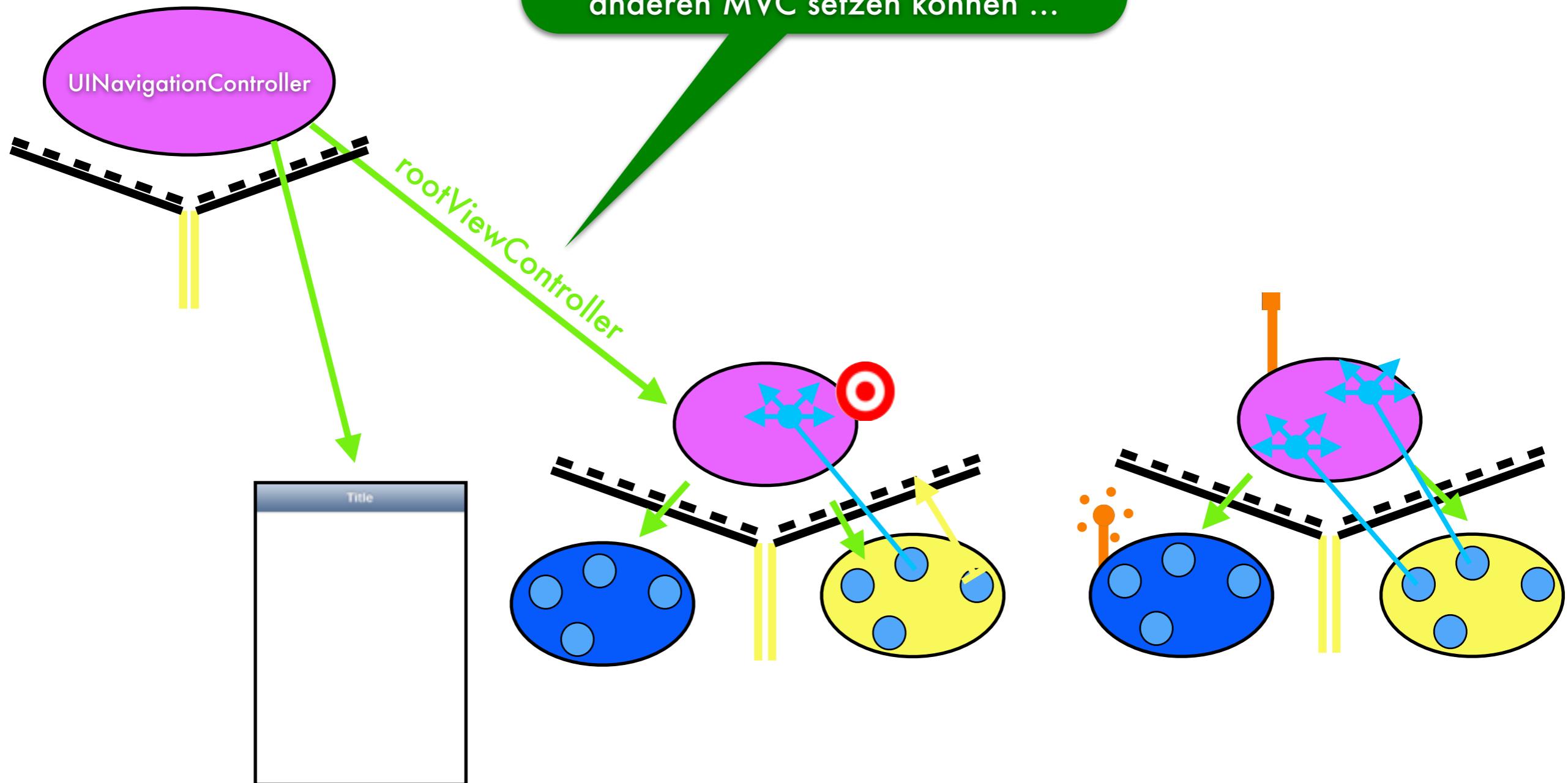


UINavigationController

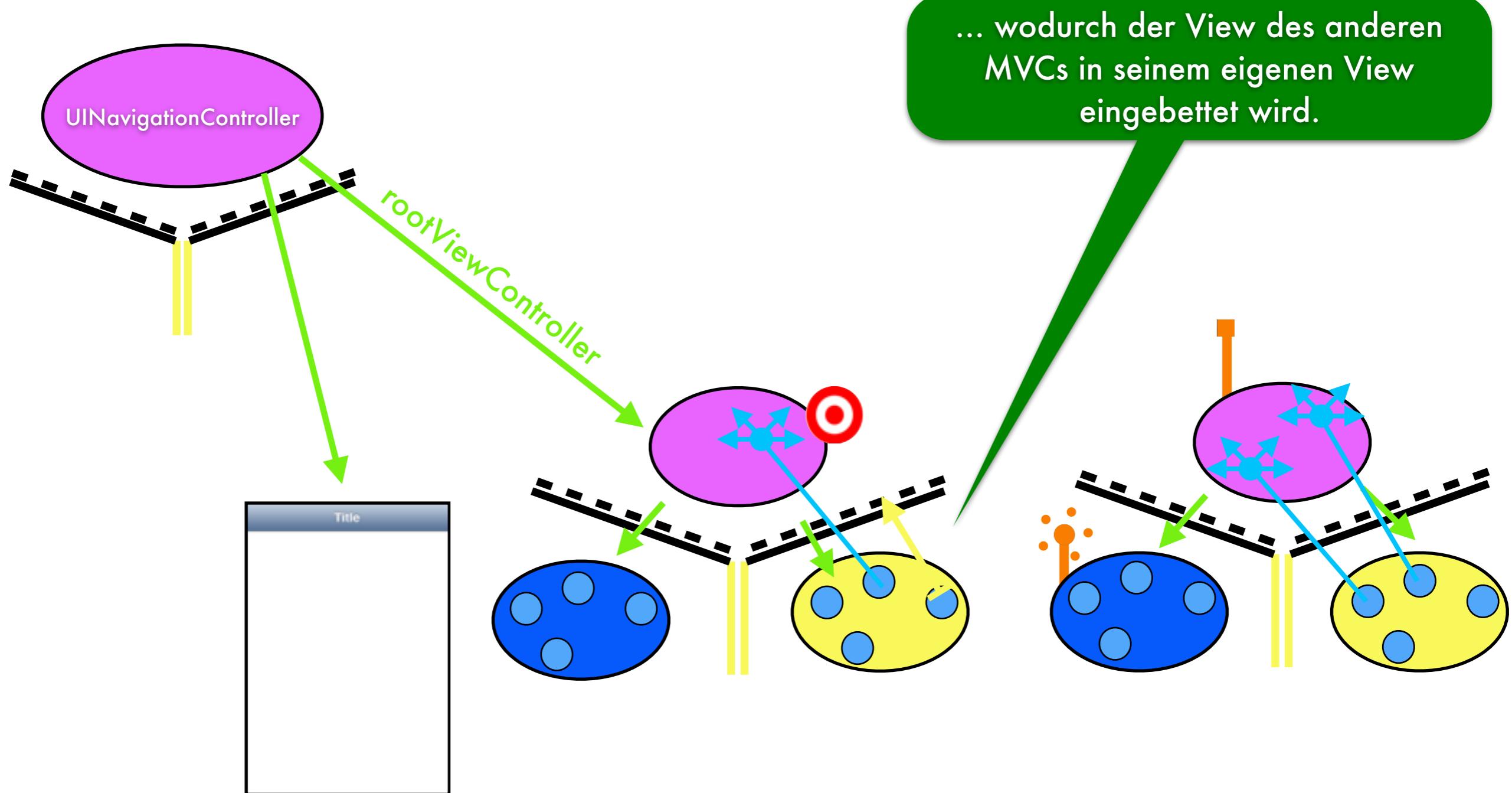


UINavigationController

Er ist aber speziell, da wir sein rootViewController Outlet auf einen anderen MVC setzen können ...

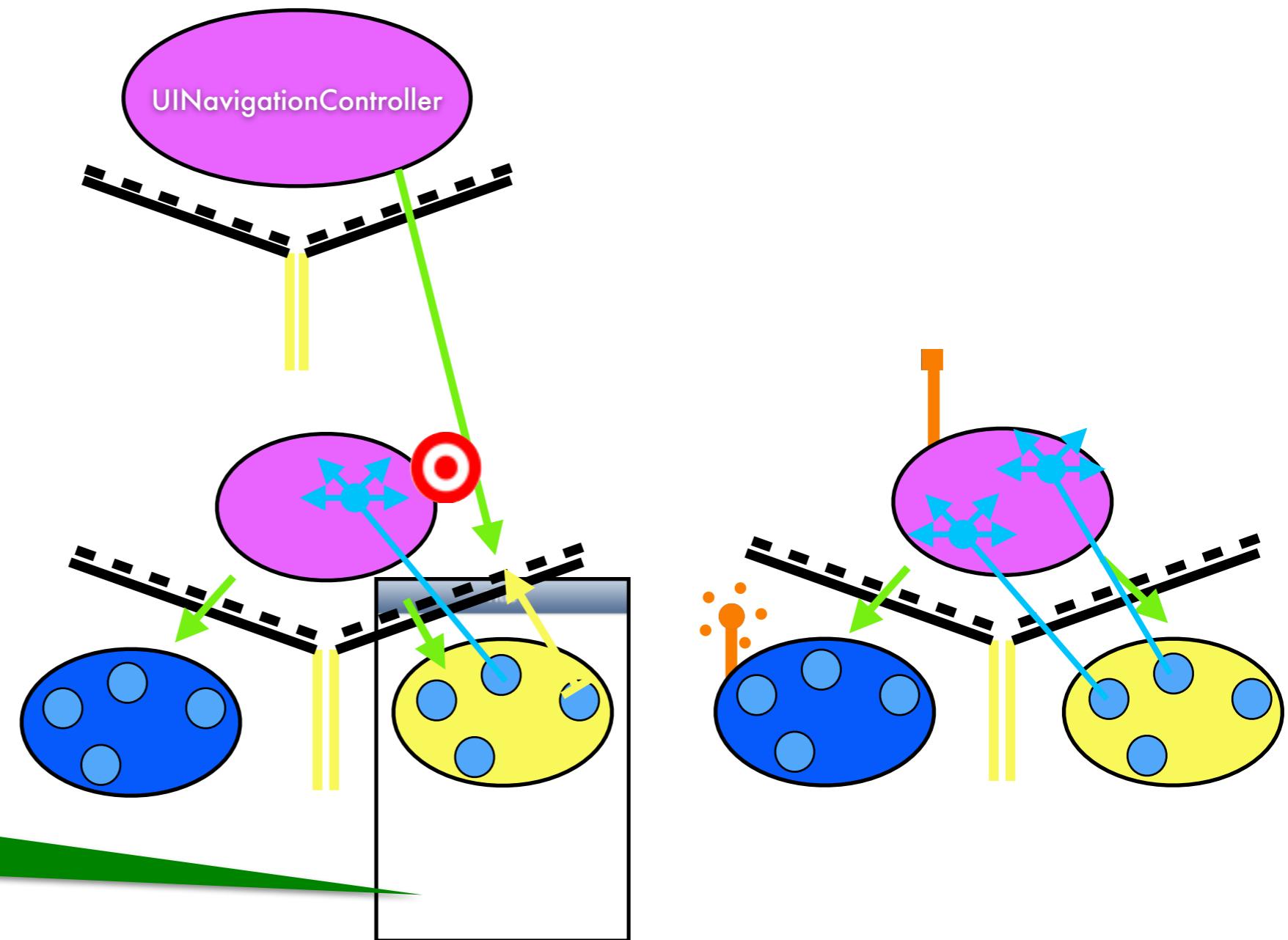


UINavigationController

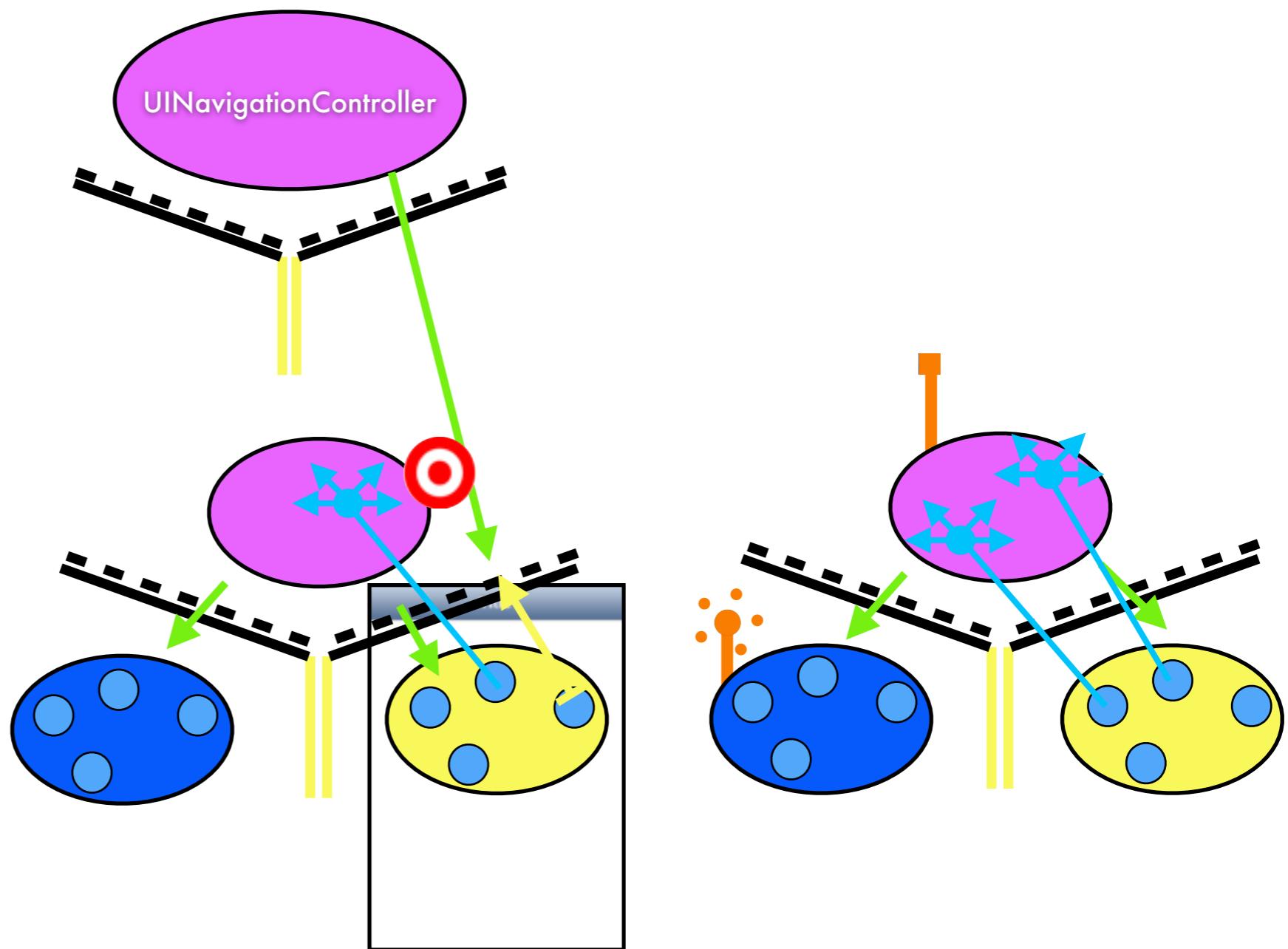


UINavigationController

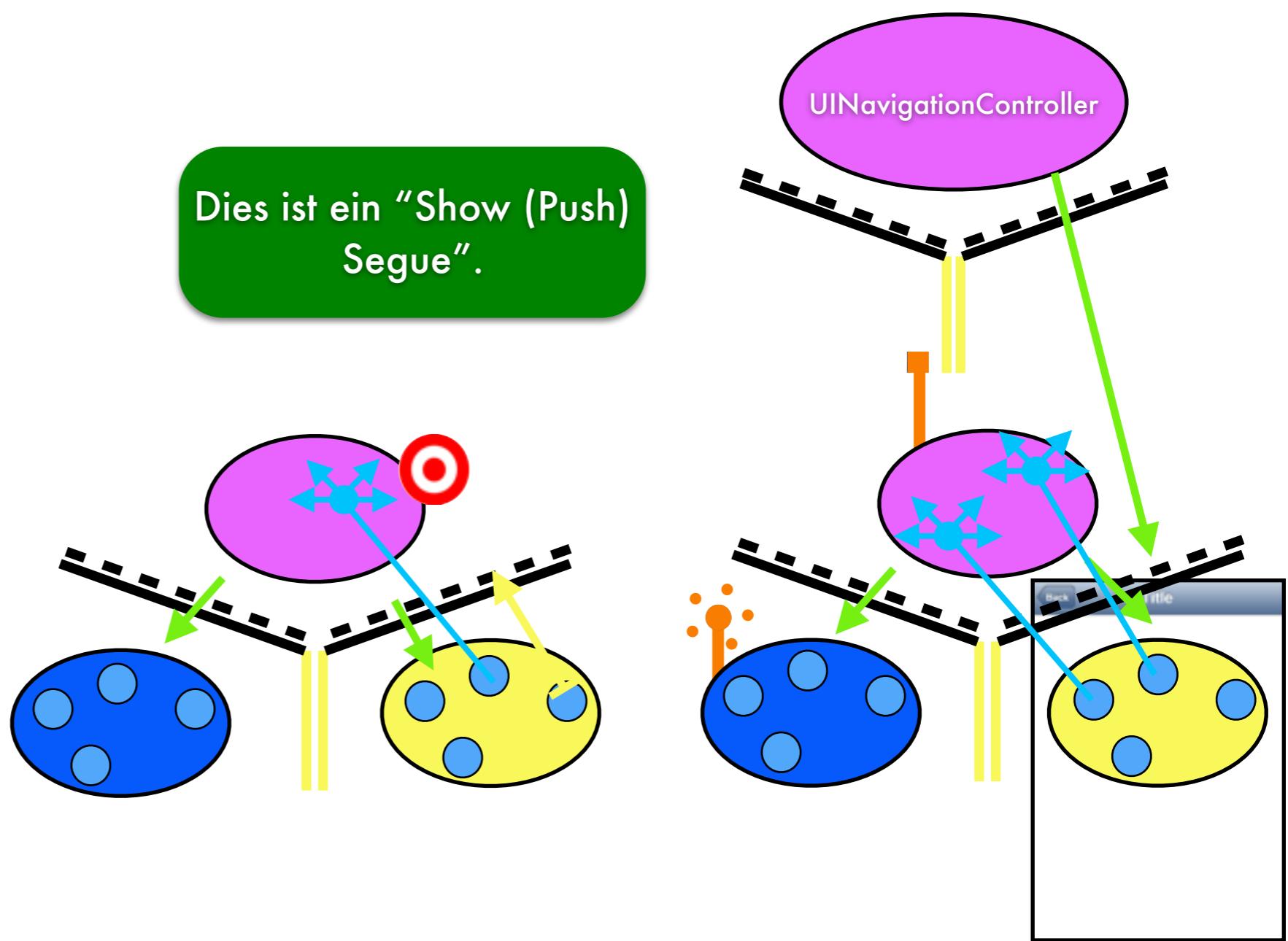
Dann kann ein UI Element (z.B. ein Button) ein Segue zu einem anderen MVC durchführen und dessen View taucht stattdessen im UINavigationController auf.



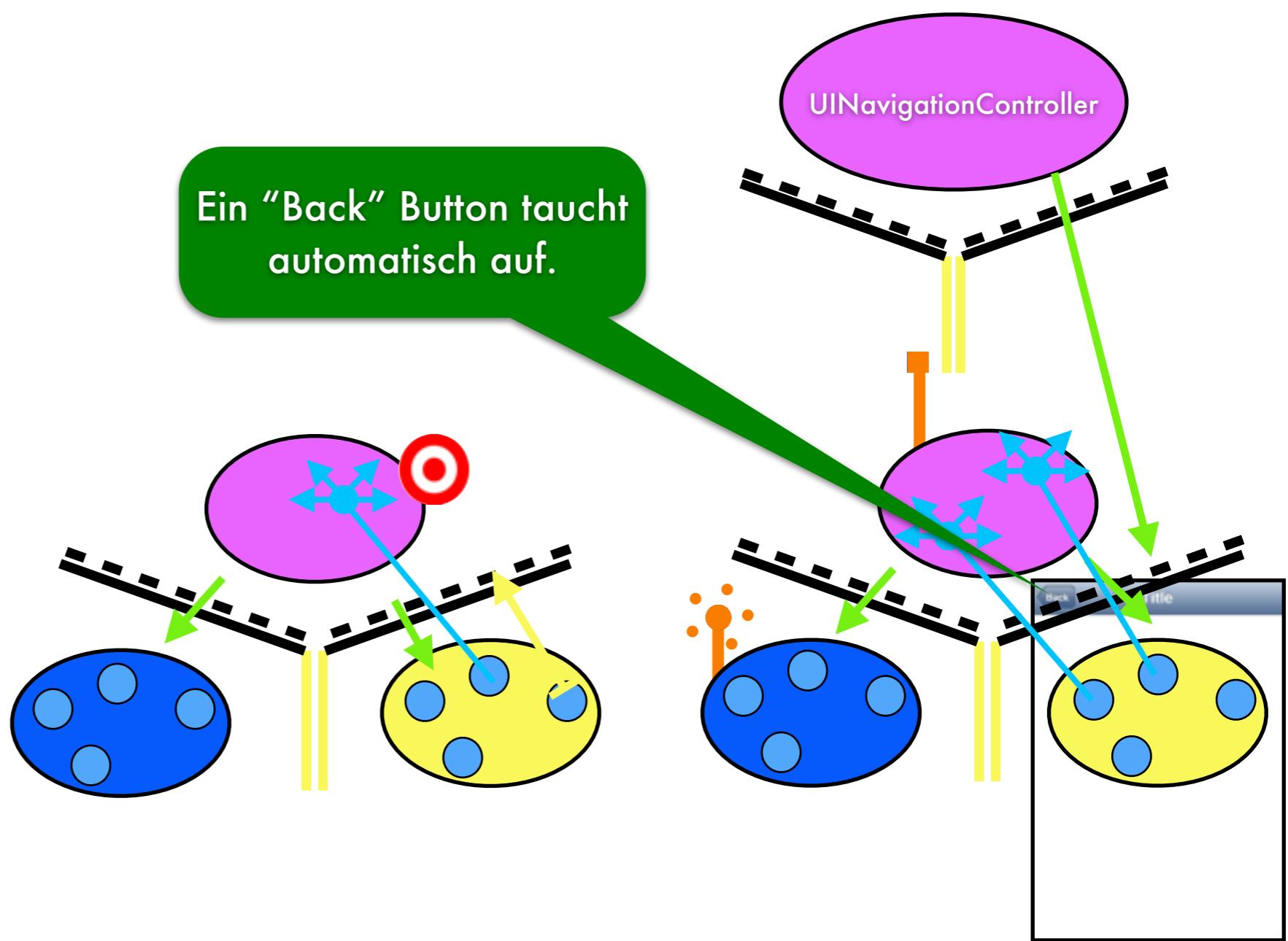
UINavigationController



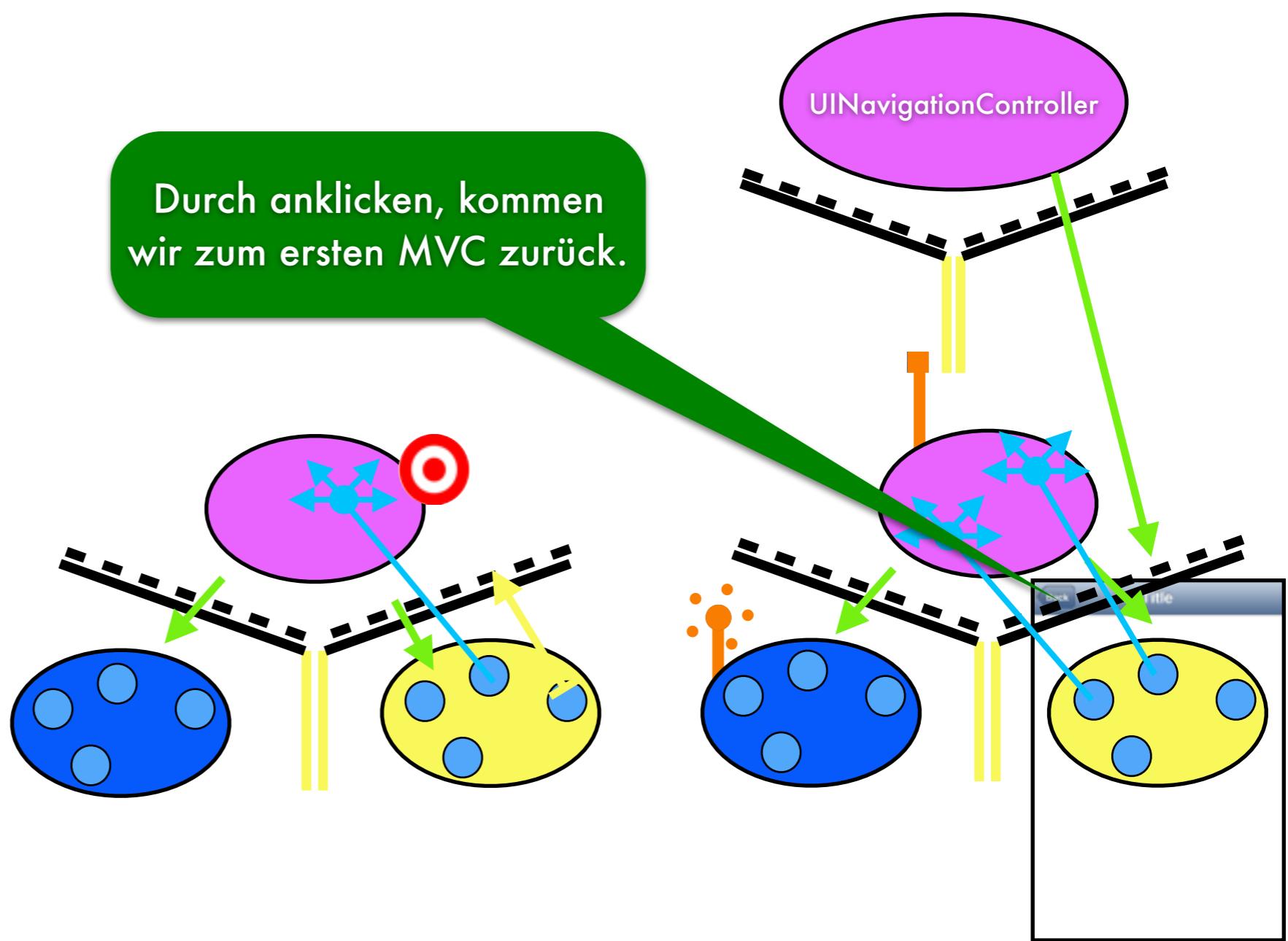
UINavigationController



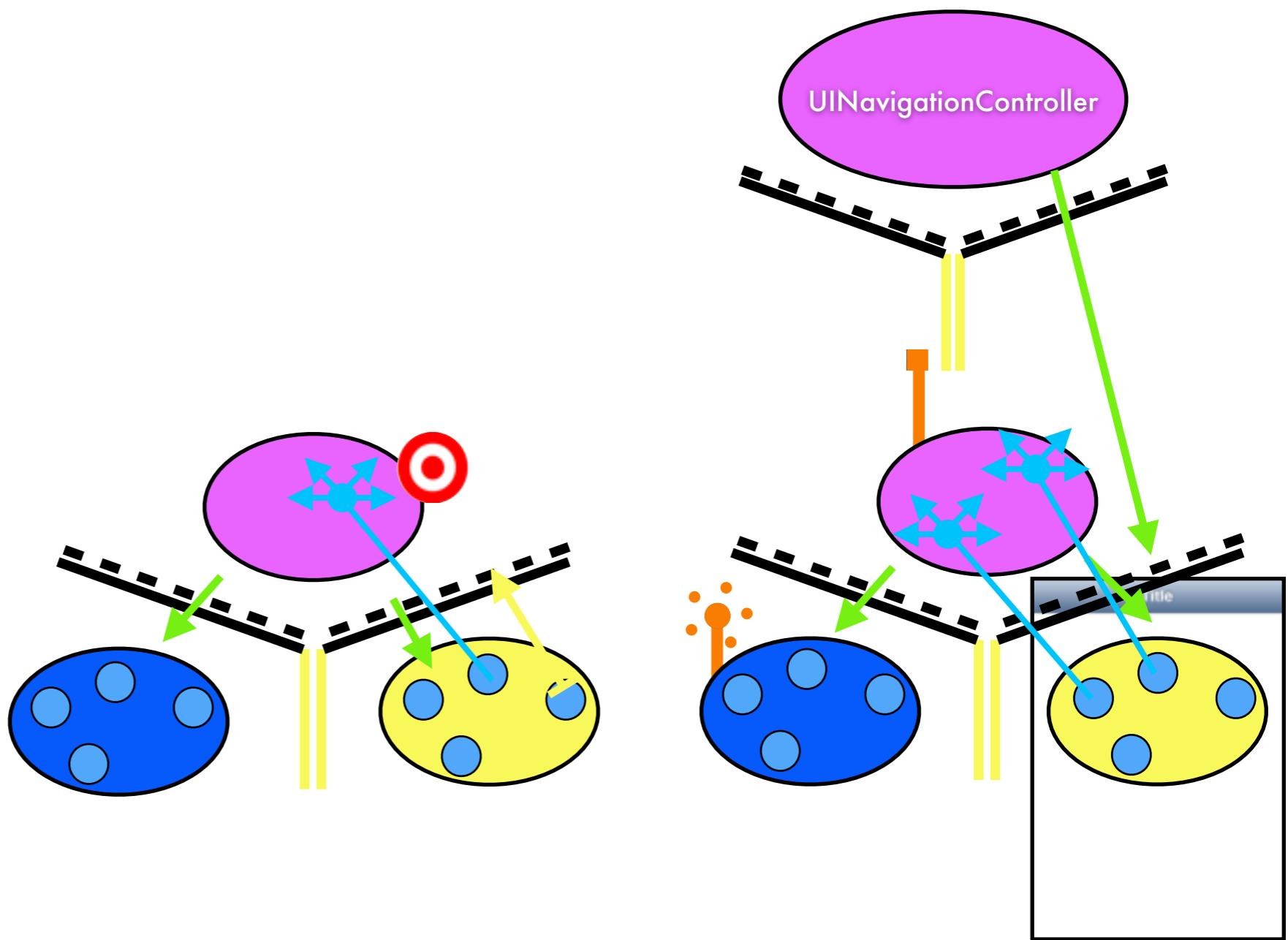
UINavigationController



UINavigationController

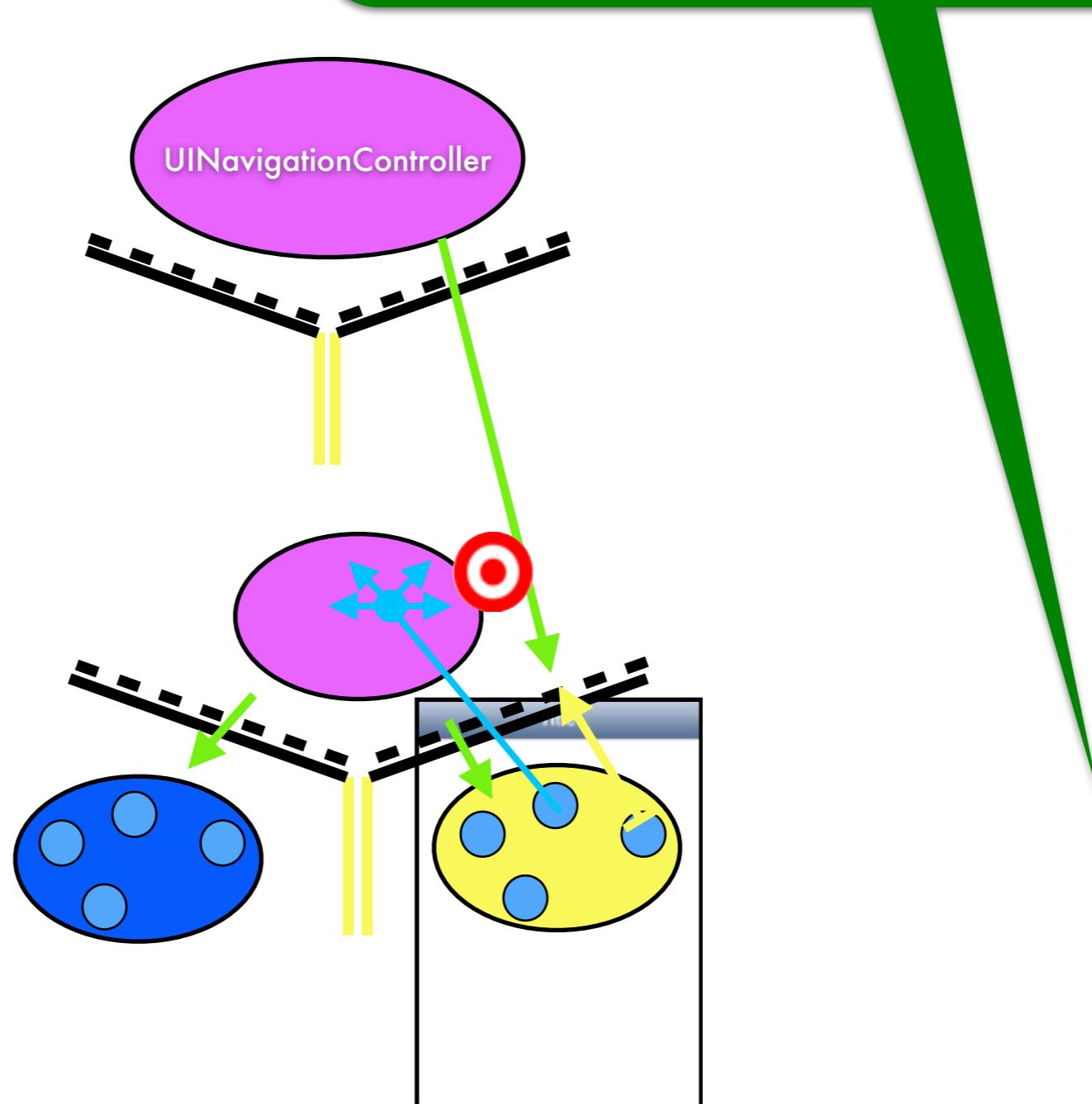


UINavigationController



UINavigationController

Nach dem Zurückkehren von einem MVC verschwindet dieser
(tatsächlich wird dieser vom Heap deallokiert).



Zugriff auf Sub-MVCs

Wir können auf Sub-MVCs via viewControllers Property zugreifen

```
var viewControllers: [UIViewController]? { get set }  
    // kann optional sein (z.B. für Tab Bar)
```

Für Tab Bar, in der Reihenfolge links nach rechts, im Array.

Für Split View ist [0] der Master und [1] der Detail.

Für Navigation Controller ist [0] Root und der Rest in Reihenfolge des Stack.

Obwohl es setzbar ist, wird dies normal im Storyboard, via Segues, etc.

gemacht. Zum Beispiel via push und pop Funktion des Navigation Controller.

Wie können wir den SVC, TBC oder den NC selbst erhalten?

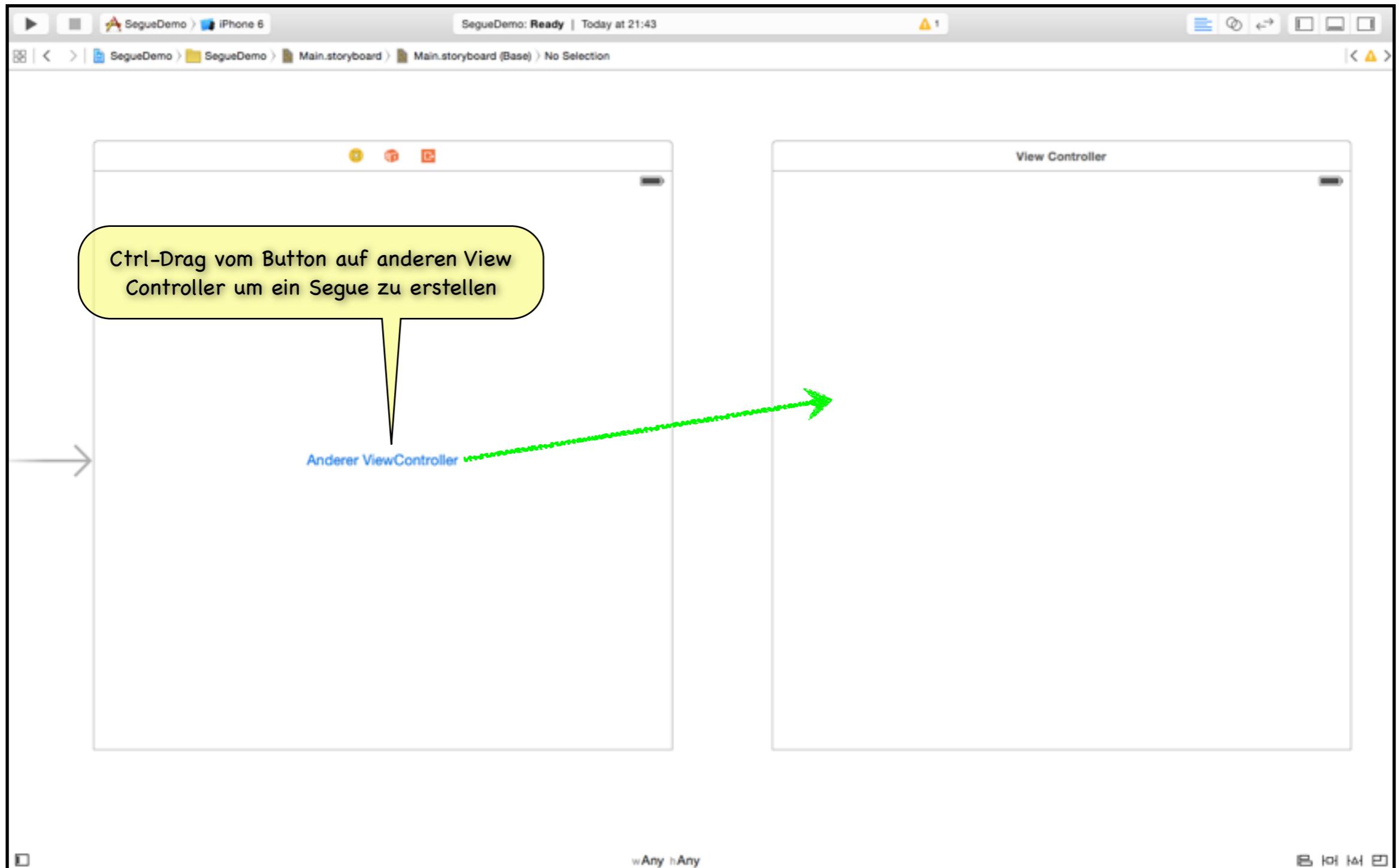
Jeder UIViewController kennt den Split View, Tab Bar oder Navigation Controller in dem er dargestellt wird. Dies sind UIViewController Properties...

```
var tabBarController: UITabBarController? { get }  
var splitViewController: UISplitViewController? { get }  
var navigationController: UINavigationController? { get }
```

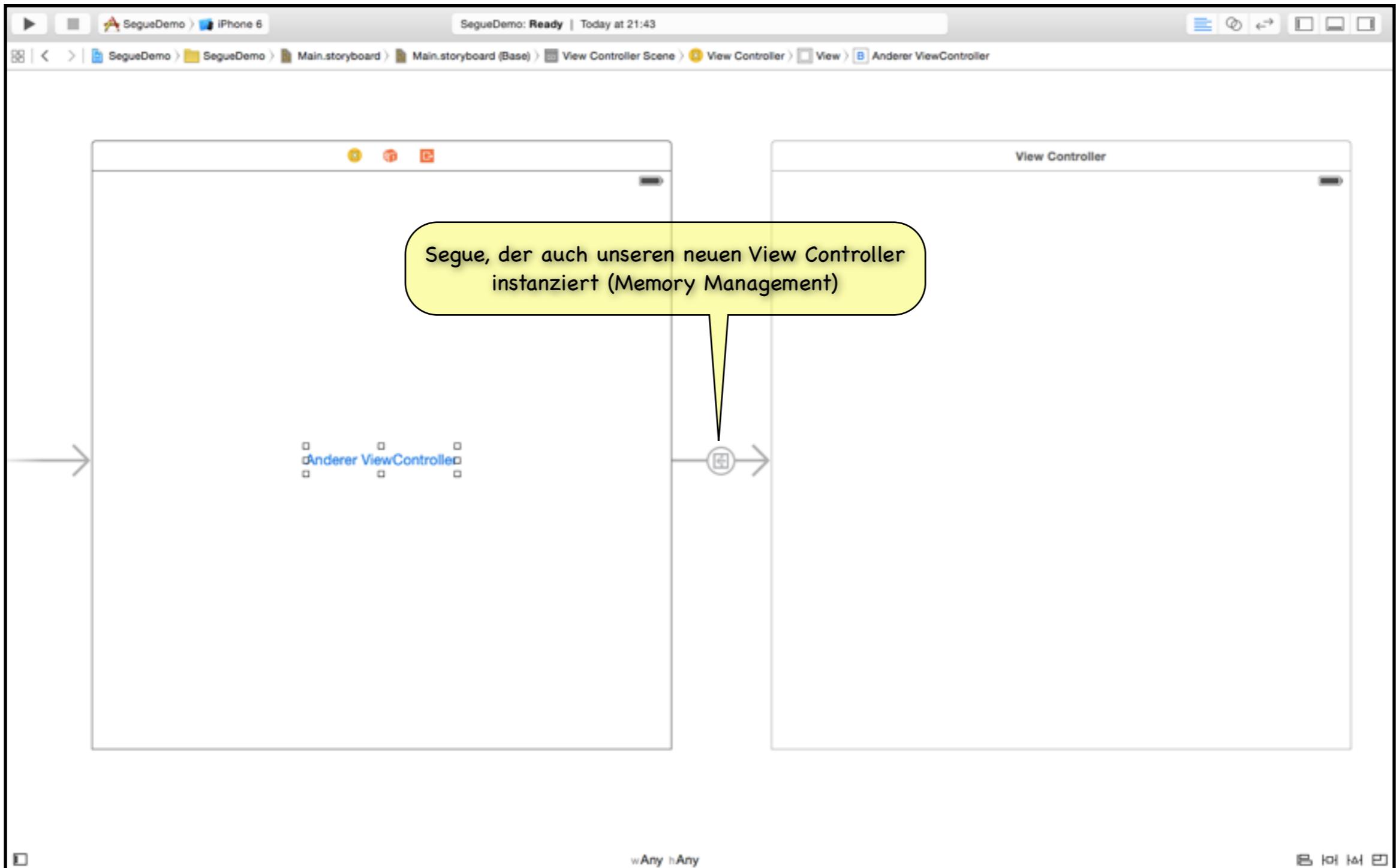
Um den Detail des Split View Controllers zu erhalten, in dem wir uns befinden...

```
if let detailVC: UIViewController = splitViewController?.viewControllers[1] { ... }
```

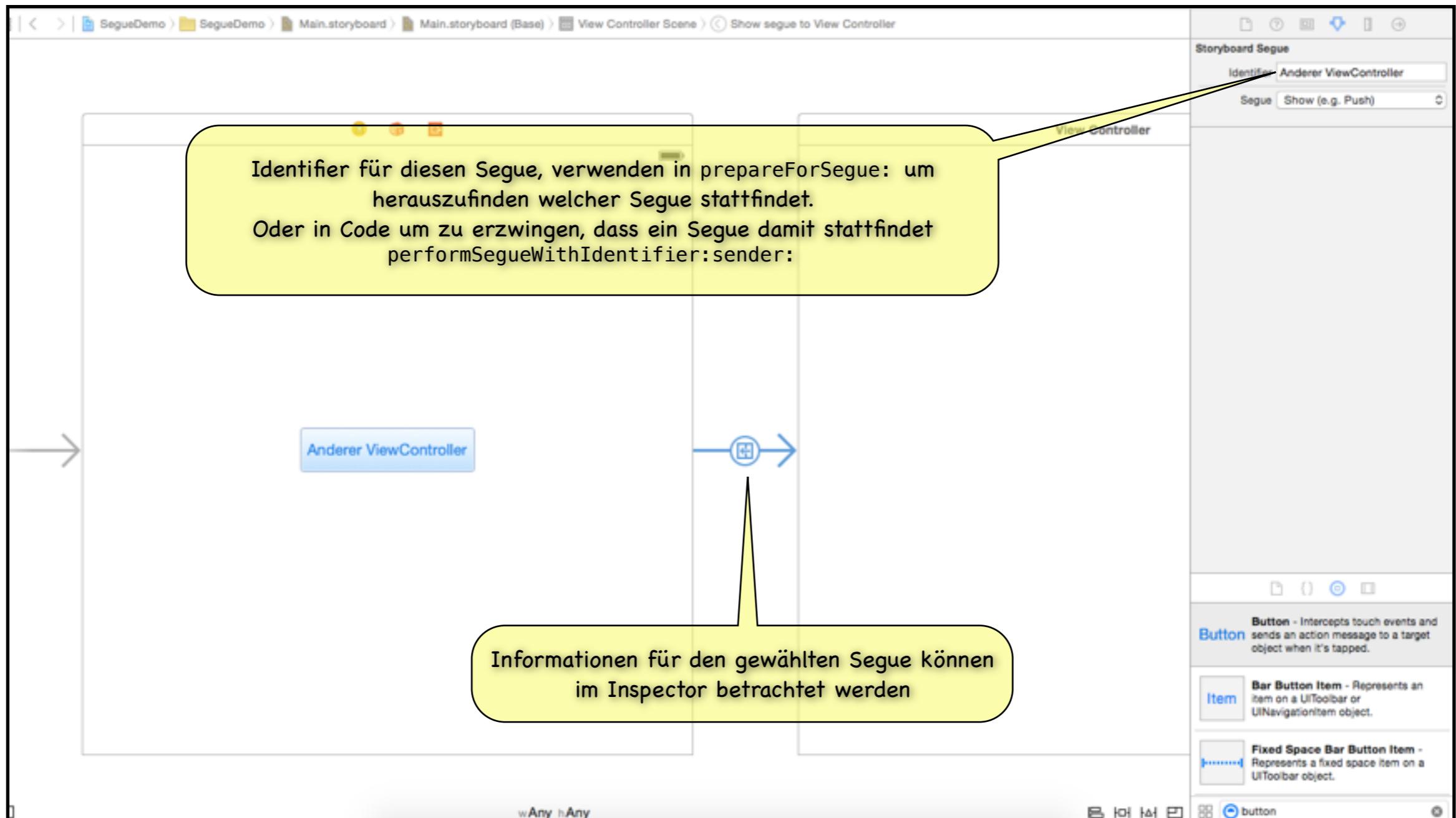
Erstellen von MVCs im Storyboard



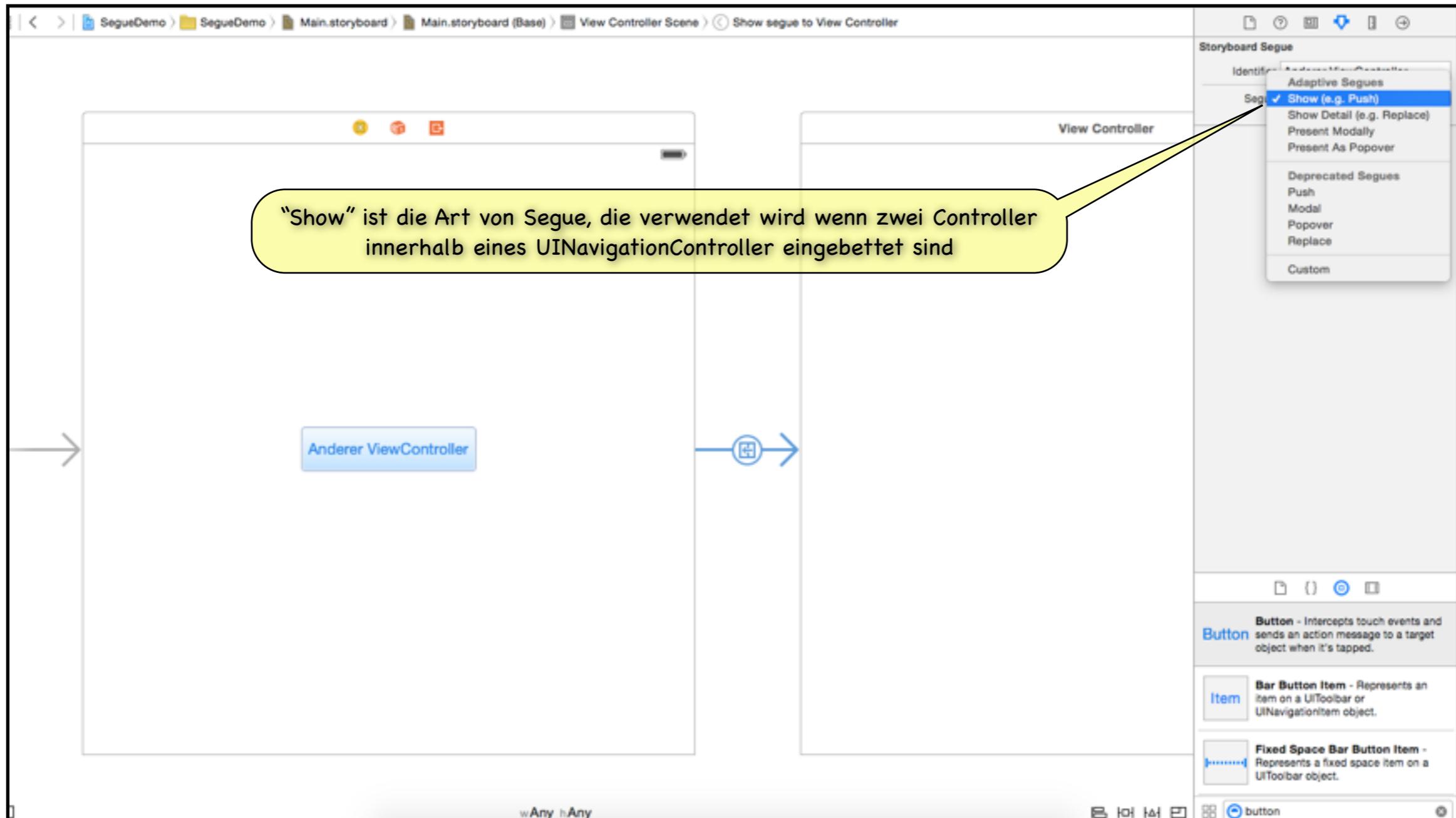
Erstellen von MVCs im Storyboard



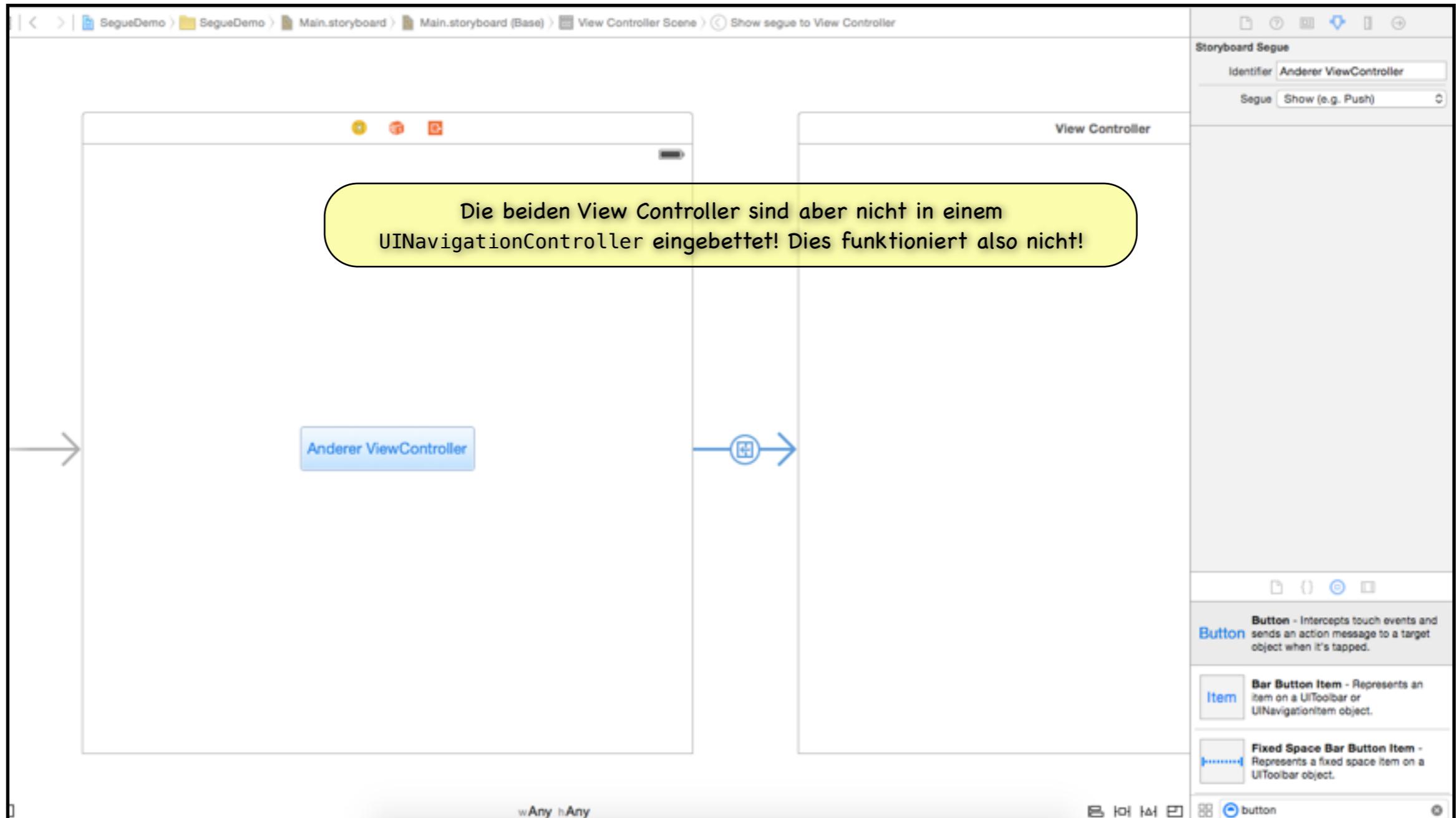
Erstellen von MVCs im Storyboard



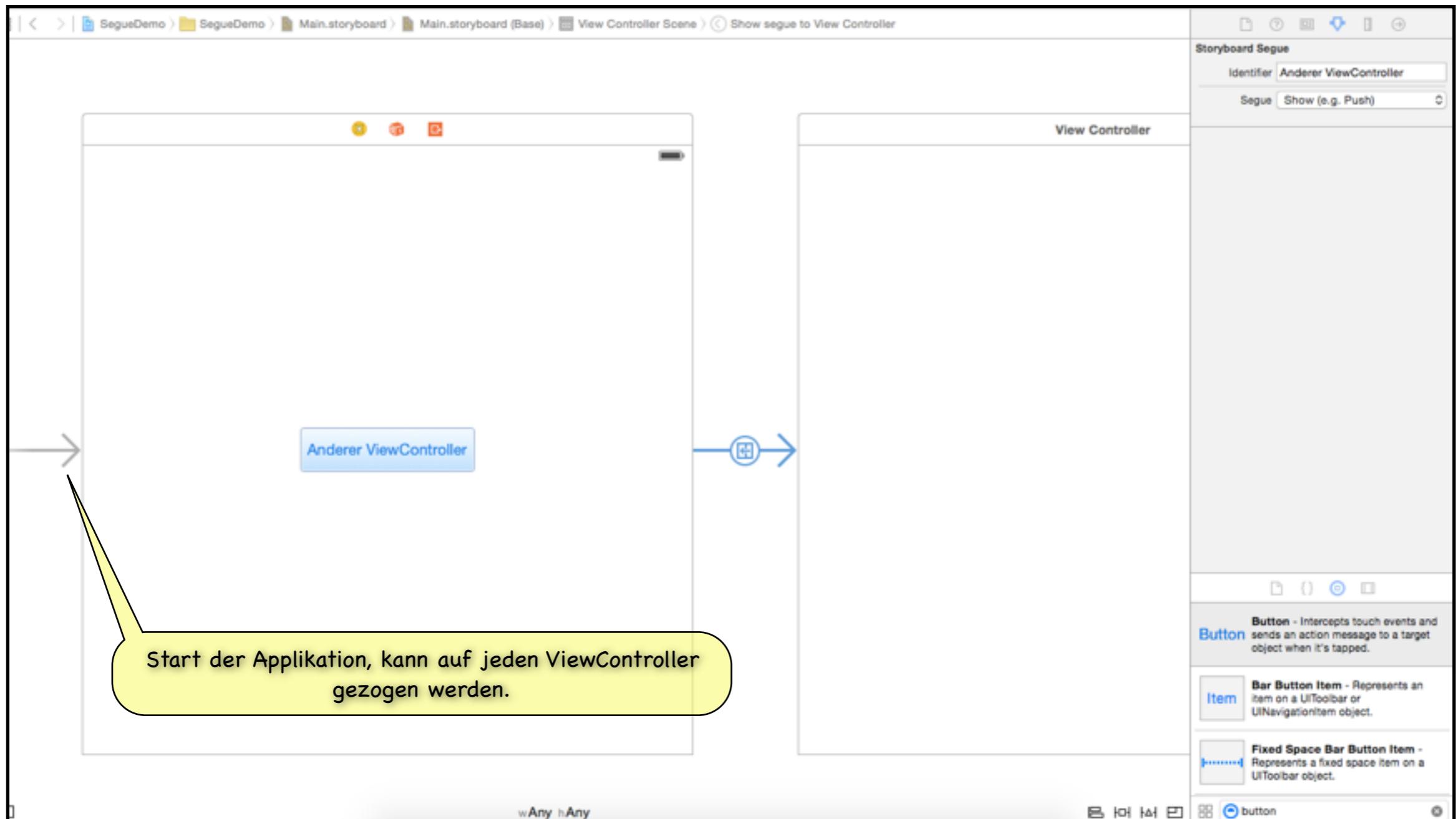
Erstellen von MVCs im Storyboard



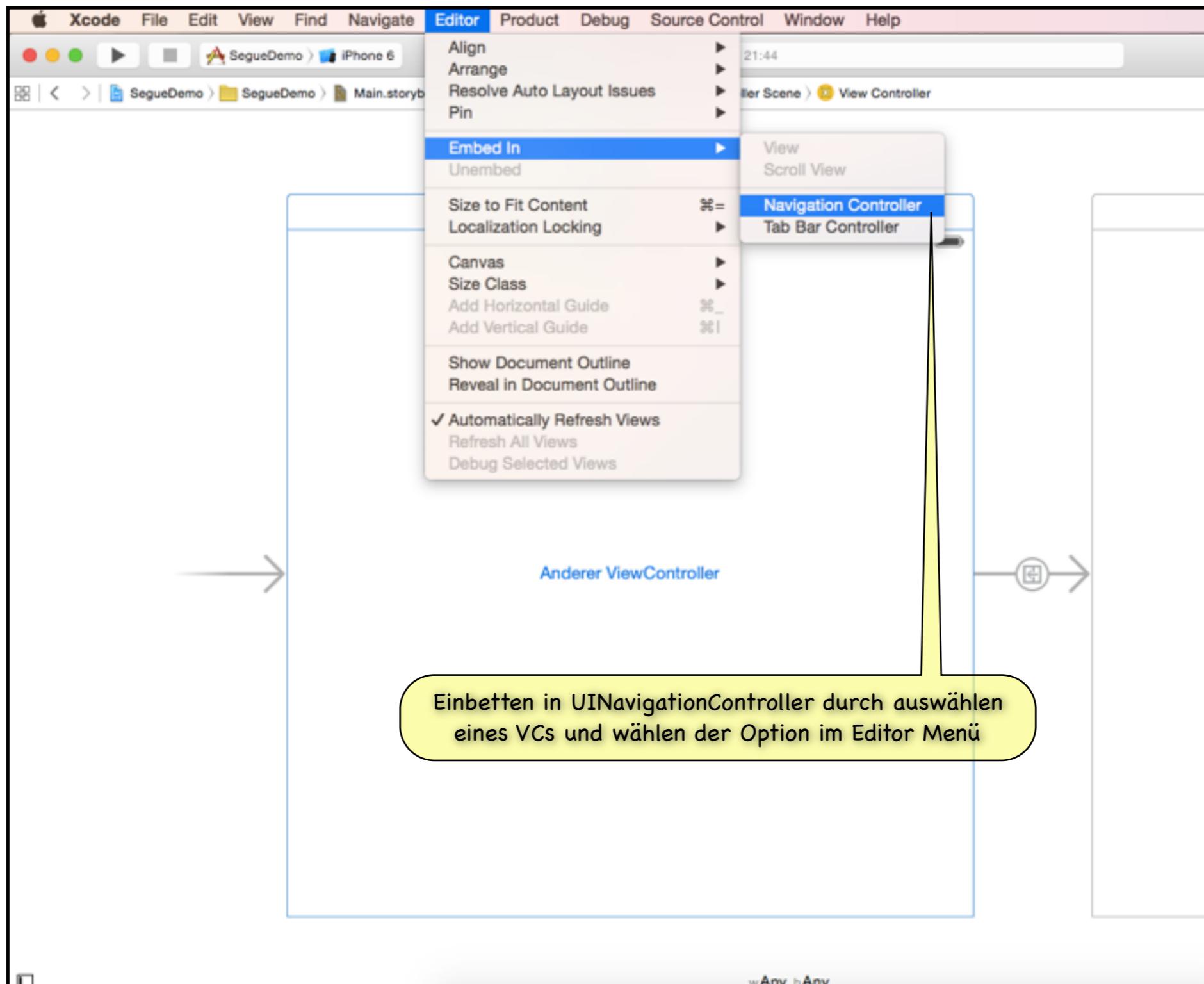
Erstellen von MVCs im Storyboard



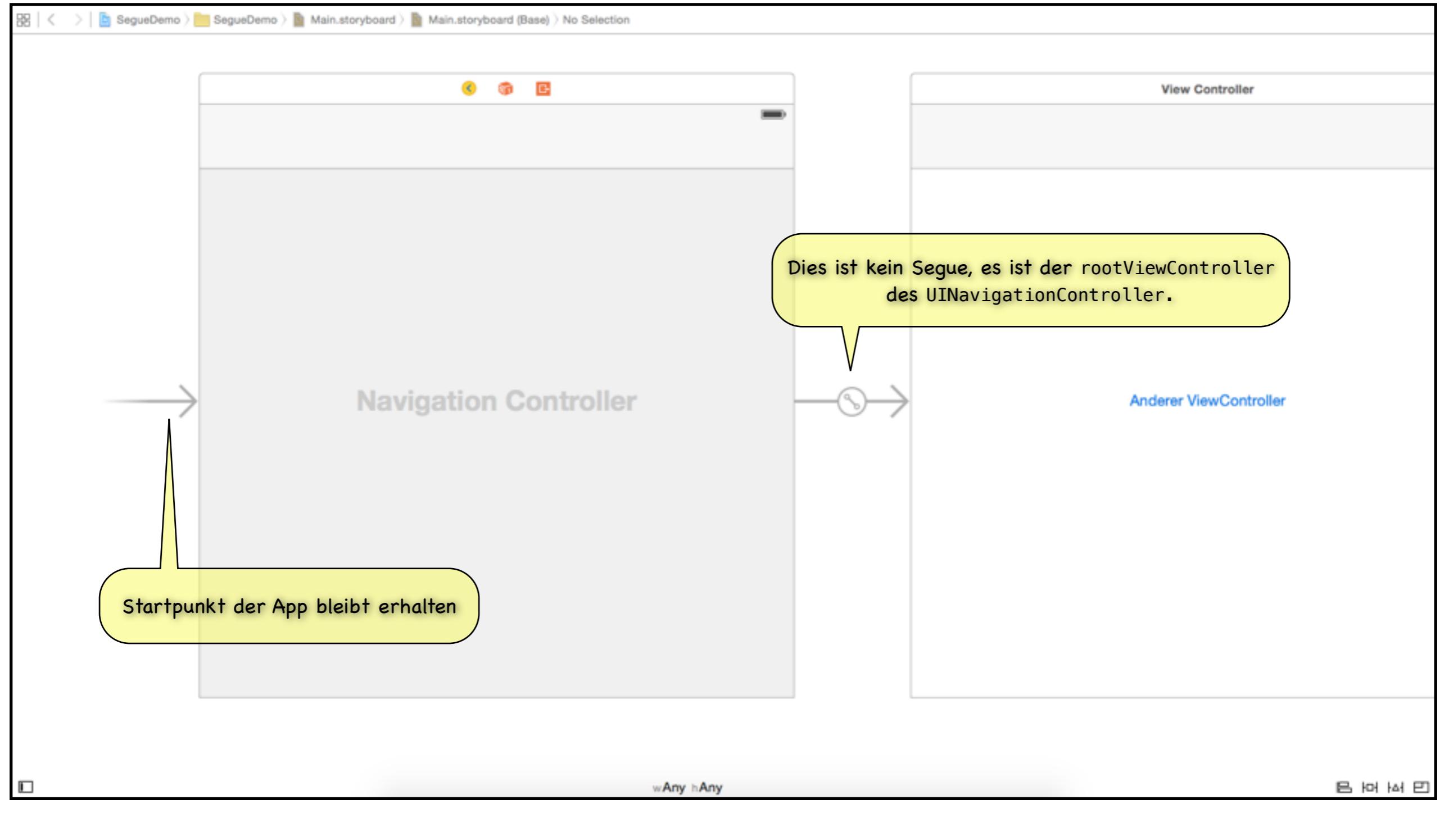
Erstellen von MVCs im Storyboard



Erstellen von MVCs im Storyboard



Erstellen von MVCs im Storyboard



Segue in Code

Bevor der Segue durchgeführt wird, wird `prepareForSegue` ausgeführt

```
override func prepareForSegue(segue: UIStoryboardSegue, sender: AnyObject?) {  
    if segue.identifier == "toNewWorld" {  
        let nextViewController = segue.destinationViewController as! NextWorldViewController  
        nextViewController.world = "This is the new World"  
    }  
}
```

Erstellen von MVCs im Storyboard

Wie kann alles verbunden werden?

Wir haben z.B. ein Calculator MVC und ein Calculator Graph MVC.

Wie kannen beide in einem Split View genutzt werden?

Drag & Drop eines

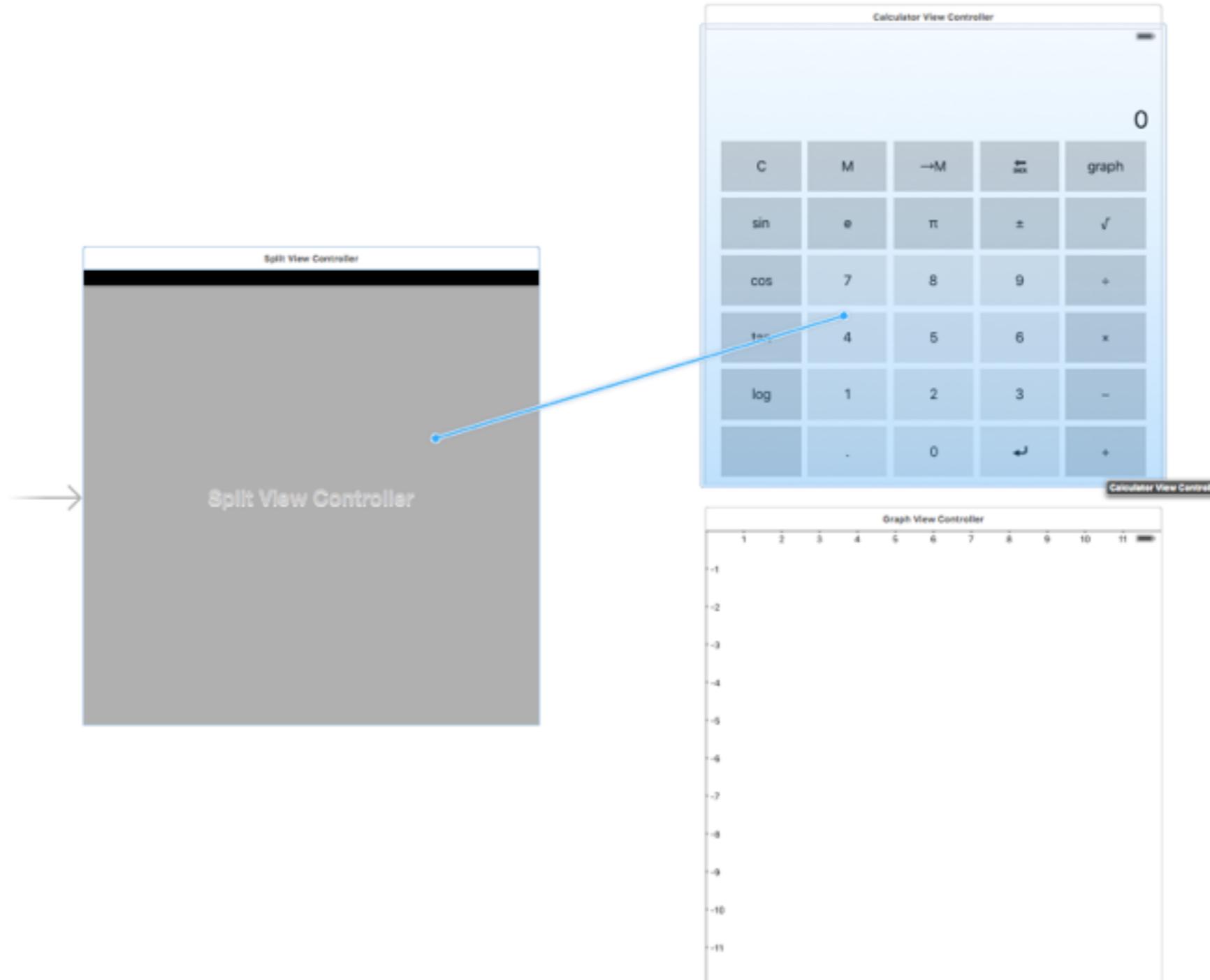


(und löschen aller mitgebrachten VCs)

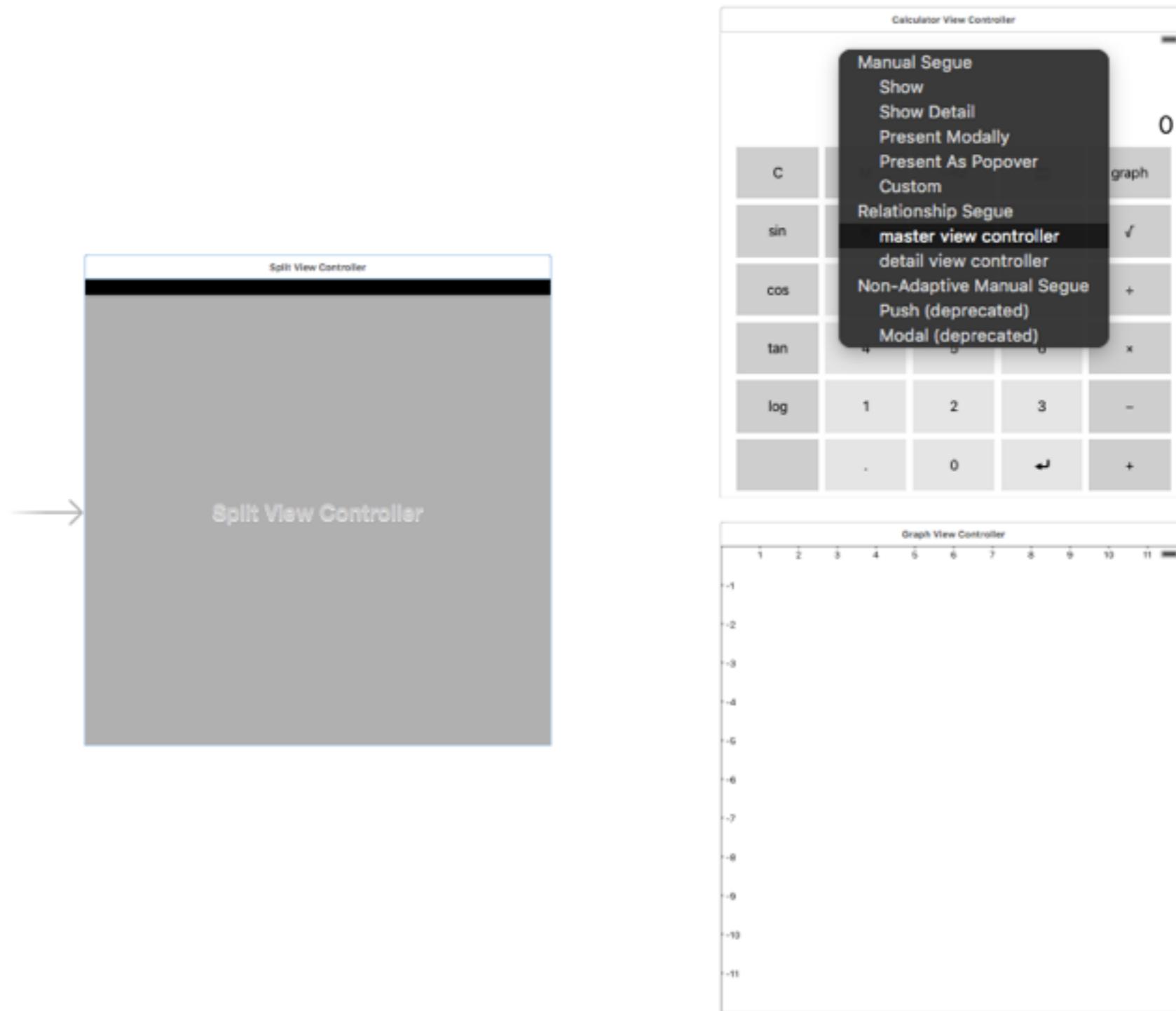
Danach ctrl-drag vom UISplitViewController zum Master und Detail MVC

...

Erstellen von MVCs im Storyboard



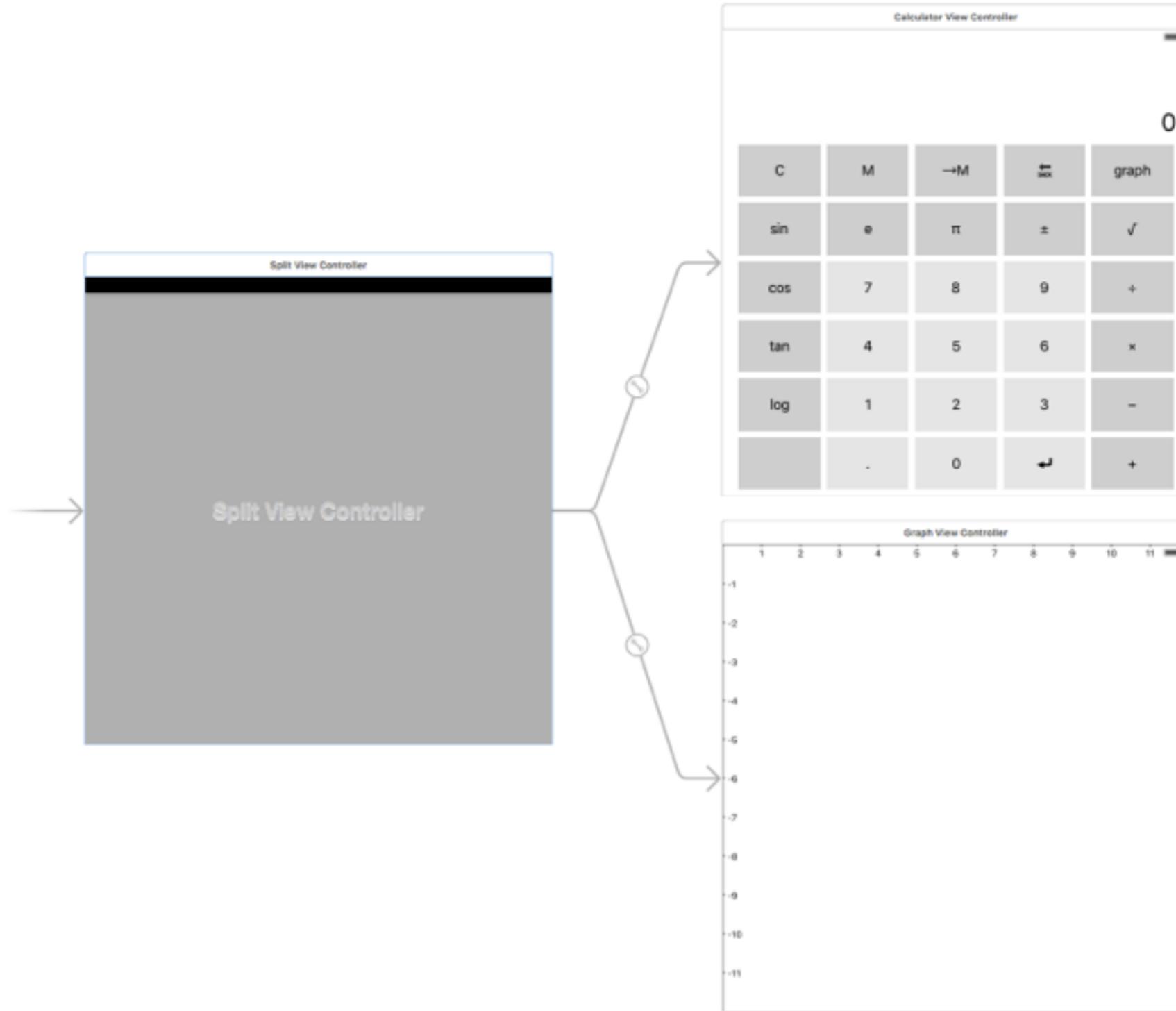
Erstellen von MVCs im Storyboard



Erstellen von MVCs im Storyboard

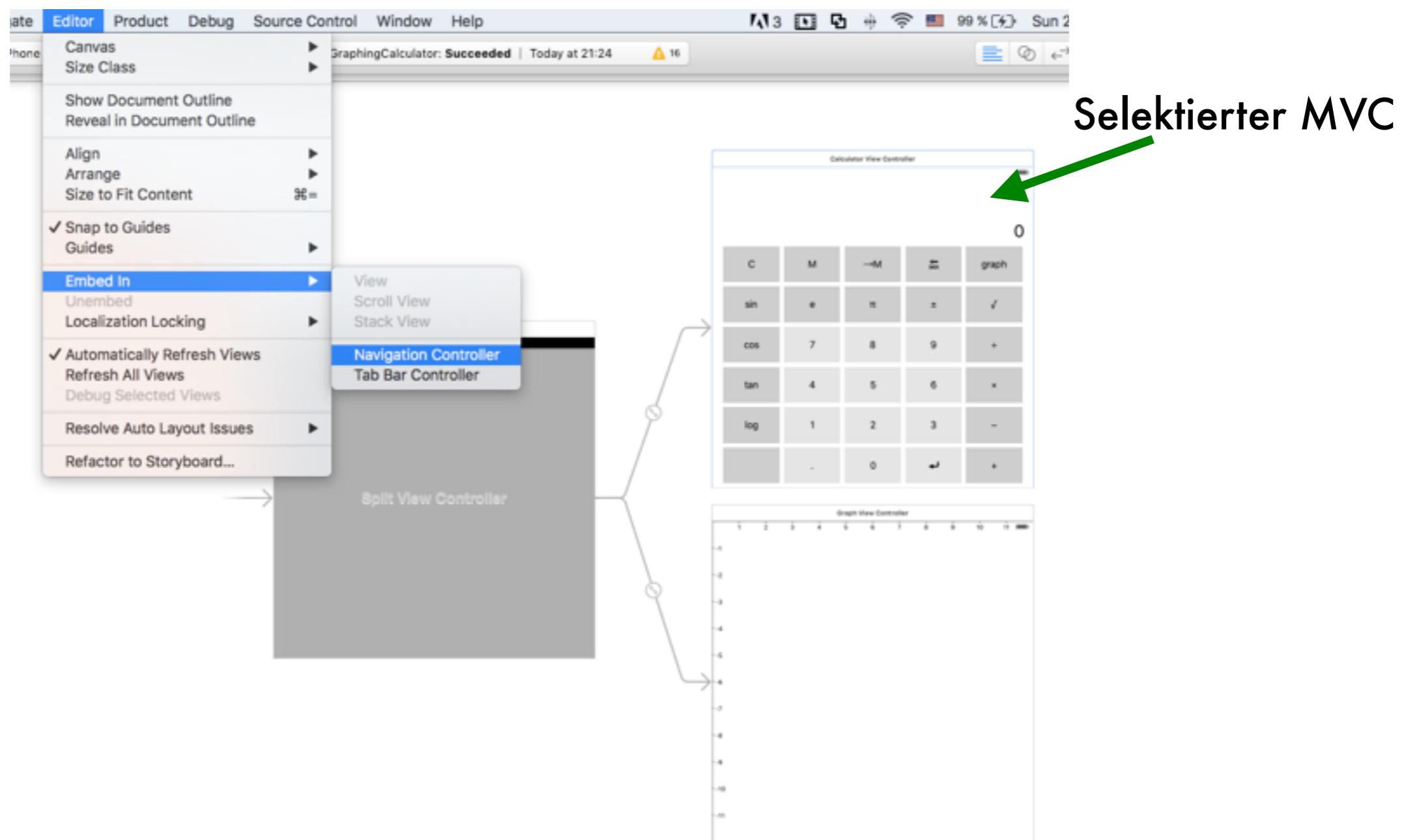


Erstellen von MVCs im Storyboard



Erstellen von MVCs im Storyboard

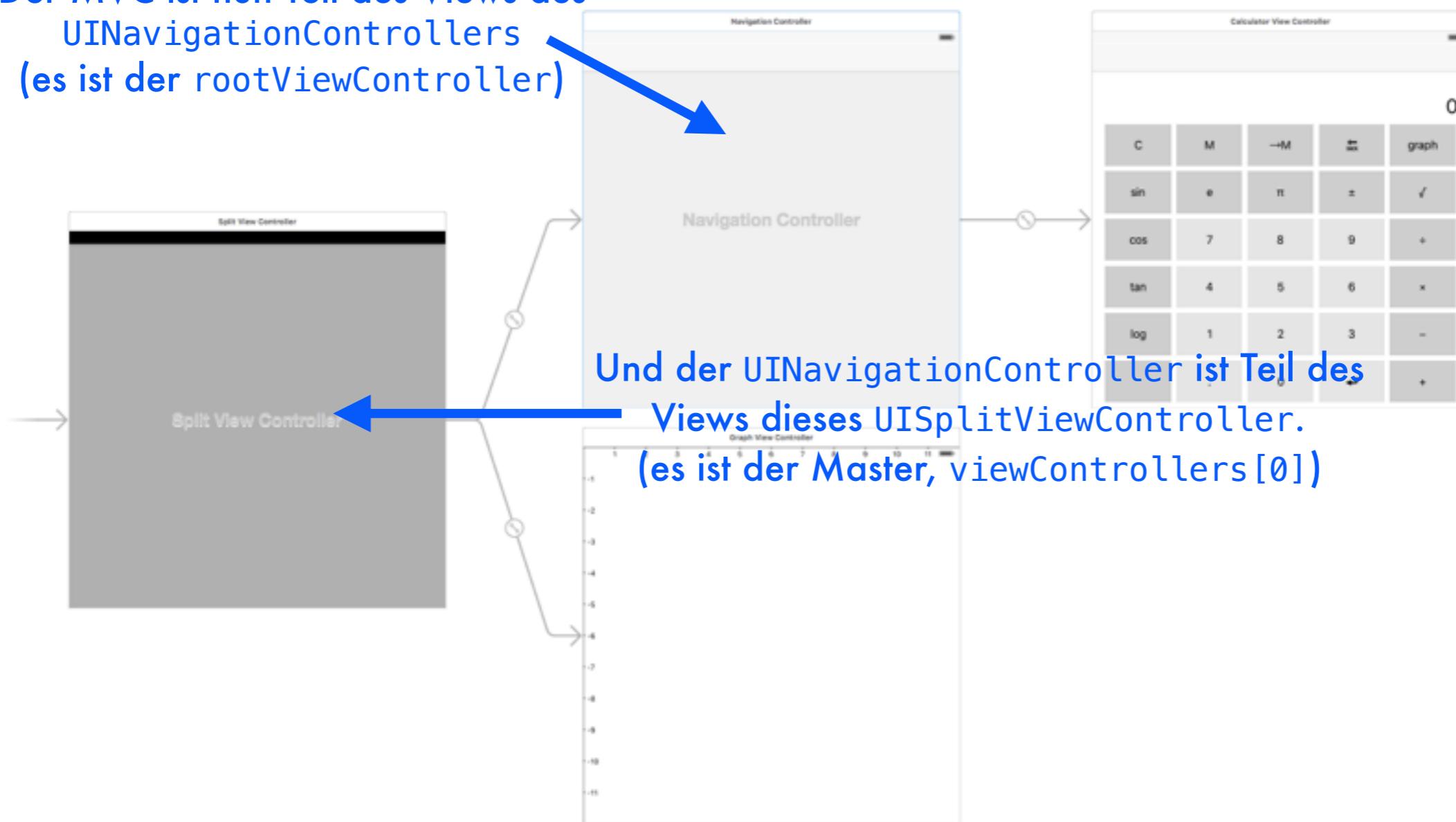
Split Views funktionieren nur korrekt auf dem iPad (und iPhone 6+)
Also brauchen wir Navigation Controller damit dies auf dem iPhone funktioniert.
Der Navigation Controller erlaubt Titel, auch auf dem iPad.



Erstellen von MVCs im Storyboard

Split Views funktionieren nur korrekt auf dem iPad (und iPhone 6+)
Also brauchen wir Navigation Controller damit dies auf dem iPhone funktioniert.
Der Navigation Controller erlaubt Titel, auch auf dem iPad.

Der MVC ist nun Teil des Views des
UINavigationController
(es ist der rootViewController)



Erstellen von MVCs im Storyboard

Split Views funktionieren nur korrekt auf dem iPad (und iPhone 6+)
Also brauchen wir Navigation Controller damit dies auf dem iPhone funktioniert.
Der Navigation Controller erlaubt Titel, auch auf dem iPad.

