# iPhone Application Programming
# Lecture 5: View Controllers

*Chat Wacharamanotham*
*Media Computing Group*
*RWTH Aachen University*

RWTHAACHEN
UNIVERSITY

# View Controllers

iPhone Application Programming • Prof. Jan Borchers

# Windows, Views, ViewControllers

- Window
    - Every application has at least one window
    - Blank canvas to host views

- View
    - Draw and animate content
    - Layout subviews
    - Receive and forward events
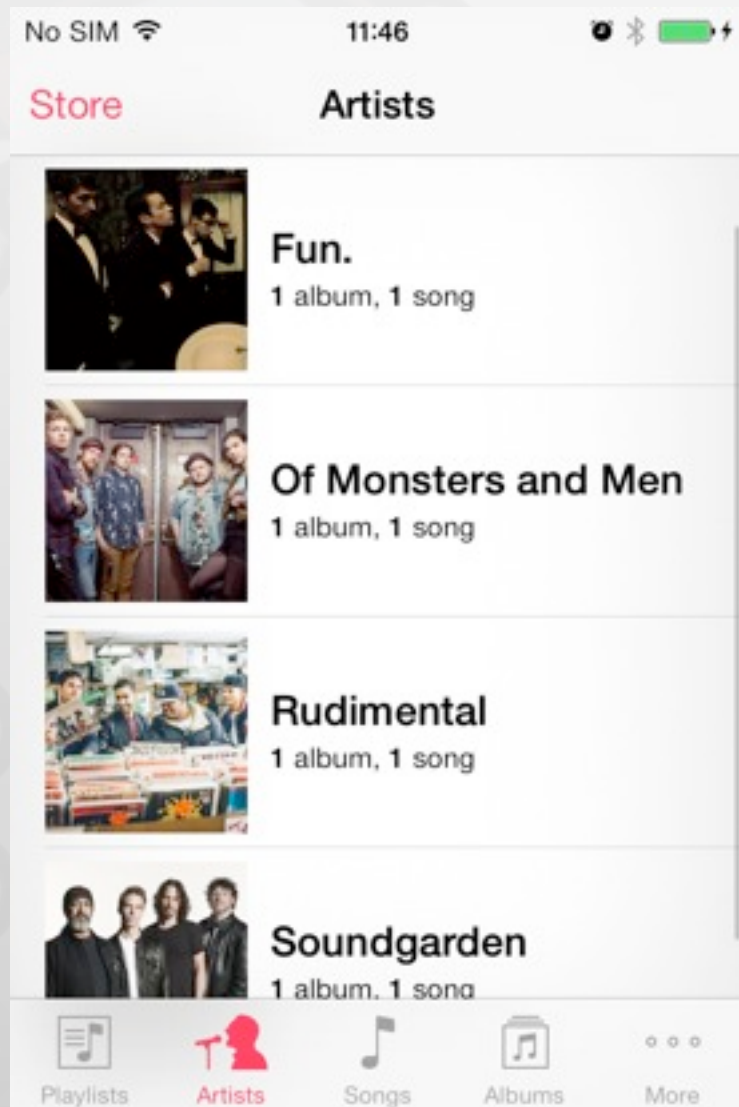
- ViewController
    - Manages a self-contained view hierarchy

# Screens



List controller



Detail controller
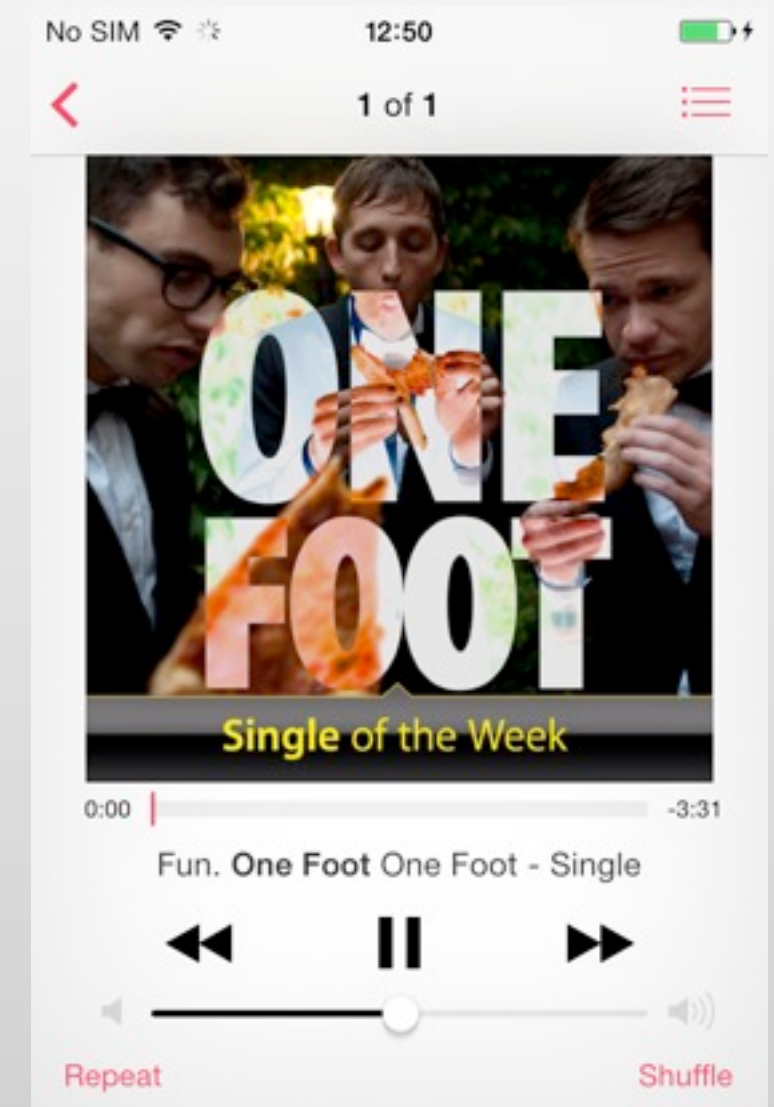


Player controller

RWTH AACHEN UNIVERSITY
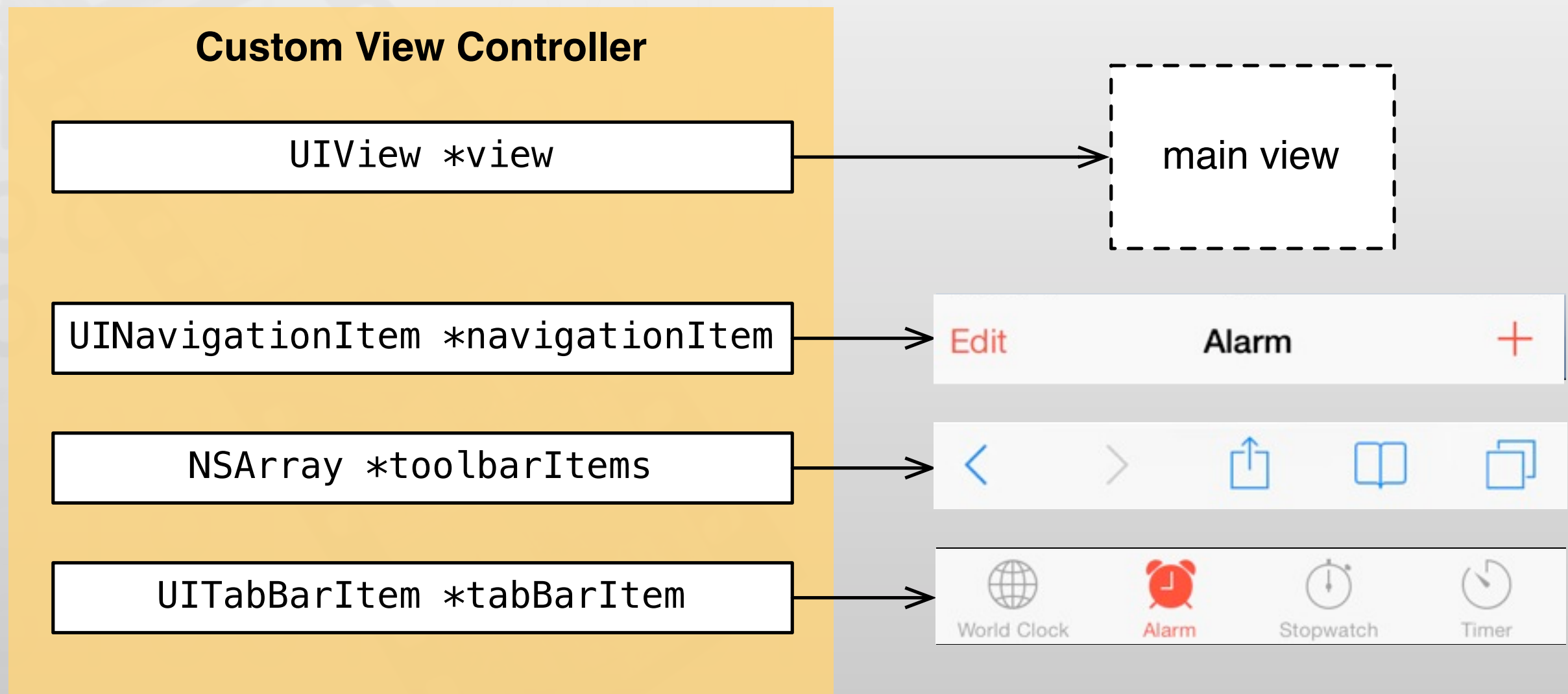
# View Controllers

- Every screen should have its own controller

- UIViewController implements much of the iOS standard interface behavior

  - Loading nib file

  - Working with navigation, tab, and tool bars

  - Composing multiple view controllers

  - Handling events and memory warnings

  - Managing interface orientation change

# View Controller Anatomy

**Custom View Controller**

| UIView *view |
| --- |

→ main view

| UINavigationItem *navigationItem |
| --- |

→ Edit　　　　　Alarm　　　　　+

| NSArray *toolbarItems |
| --- |

→ (toolbar icons)

| UITabBarItem *tabBarItem |
| --- |

→ World Clock　Alarm　Stopwatch　Timer

# View Management Cycle

**Custom View Controller**

`UIView *view`

`– (void)loadView`

`– (void)viewDidLoad`

main view

Nib file

Empty UIView

Custom view creation code
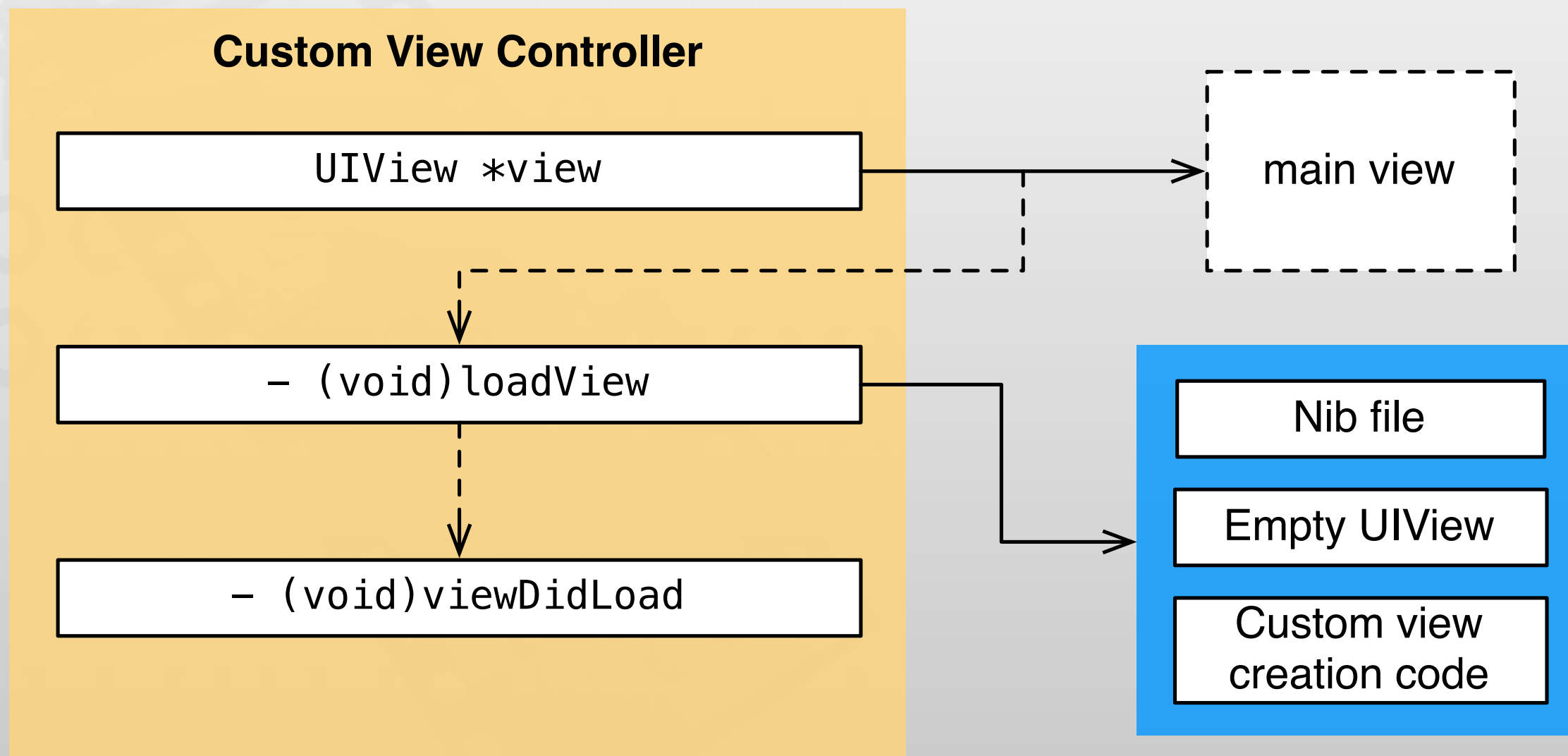
# View Controller Life Cycle

**UIKit**

## UIViewController

```
– (void)init
```

```
– (void)loadView
```

```
– (void)viewDidLoad
```

```
– (void)viewWillAppear:(BOOL)
           animated
```

```
– (void)viewDidAppear:(BOOL)
           animated
```

```
– (void)viewWillDisappear:(BOOL)
           animated
```

```
– (void)viewDidDisappear:(BOOL)
           animated
```

```
– (void)dealloc
```

RWTH AACHEN UNIVERSITY

# Interface Orientation

**UIViewController**

? → `- (BOOL)shouldAutorotateTo InterfaceOrientation:`

→ `- (void)willRotateTo InterfaceOrientation:Duration:`

**UIKit**

→ `willAnimateFirstHalfOfRotation ToInterfaceOrientation:duration:`

→ `didAnimateFirstHalfOfRotation ToInterfaceOrientation:`

→ `willAnimateSecondHalfOfRotation ToInterfaceOrientation:duration:`

→ `- (void)didRotateFrom InterfaceOrientation:`

RWTH AACHEN UNIVERSITY
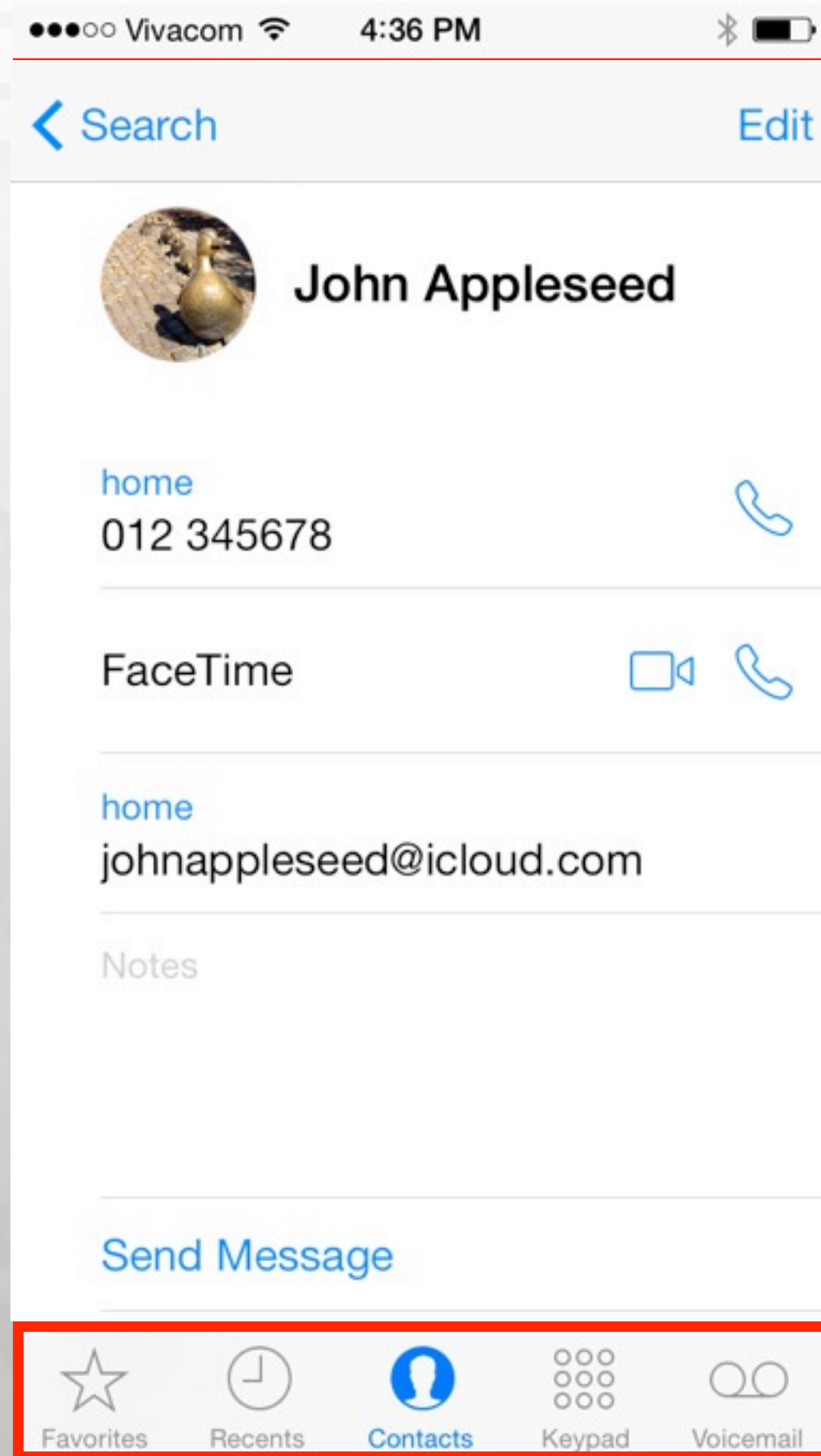
# Types of View Controllers

- Custom view controller

  - Directly express content on the screen

- Container view controller

  - Manage other view controllers

  - Usually does not express content directly

- Modal view controller

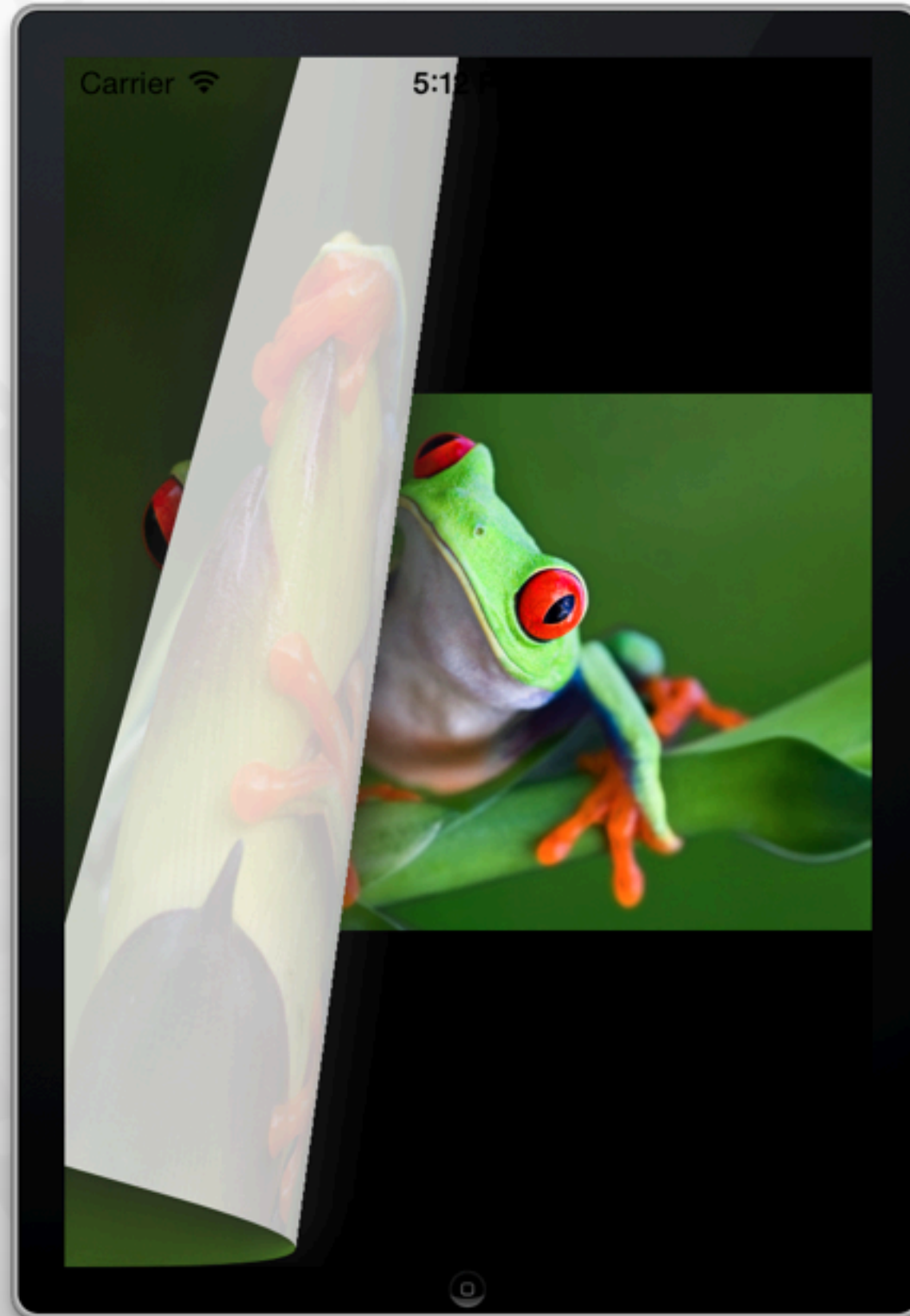  - Any view controller can be presented modally

Navigation Bar

Modal View

Table View

Tab Bar

iPhone Application Programming  •  Prof. Jan Borchers

# Page View Controller
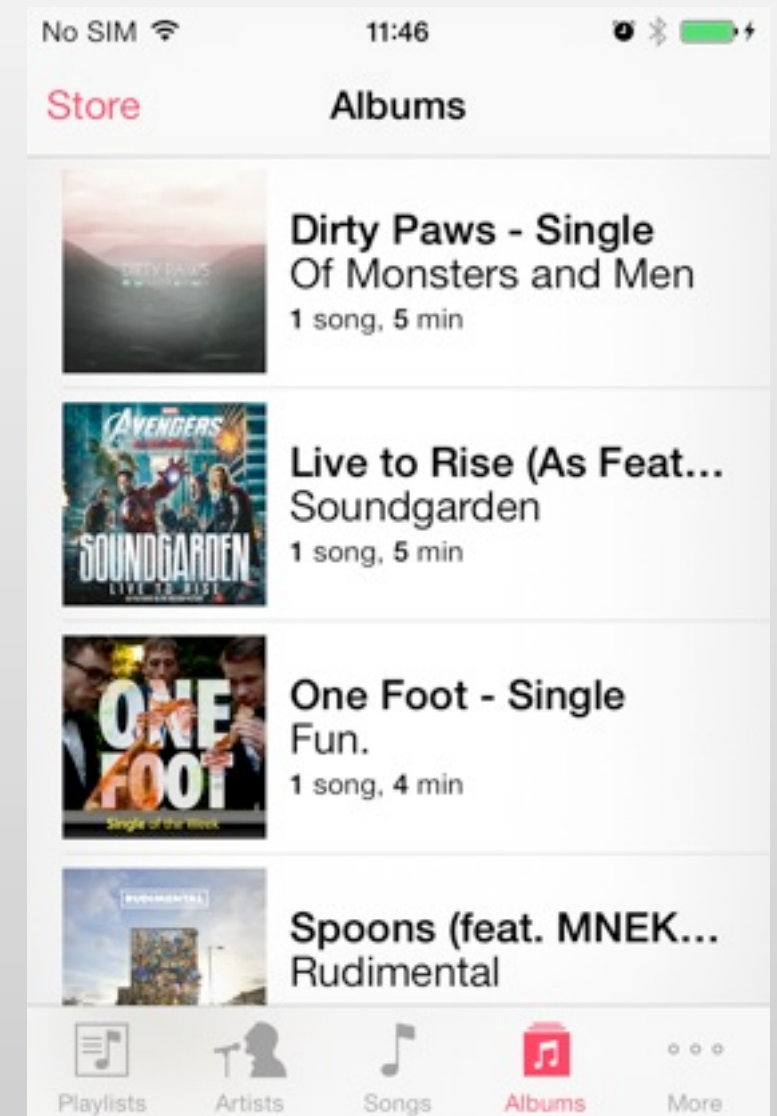


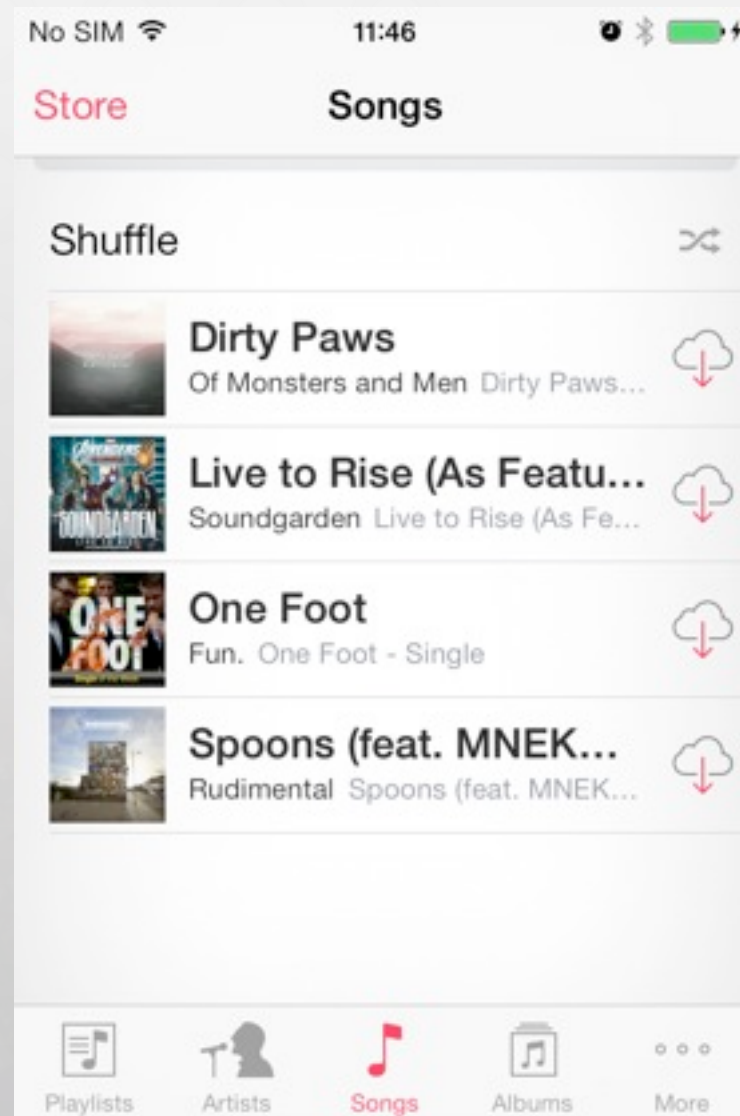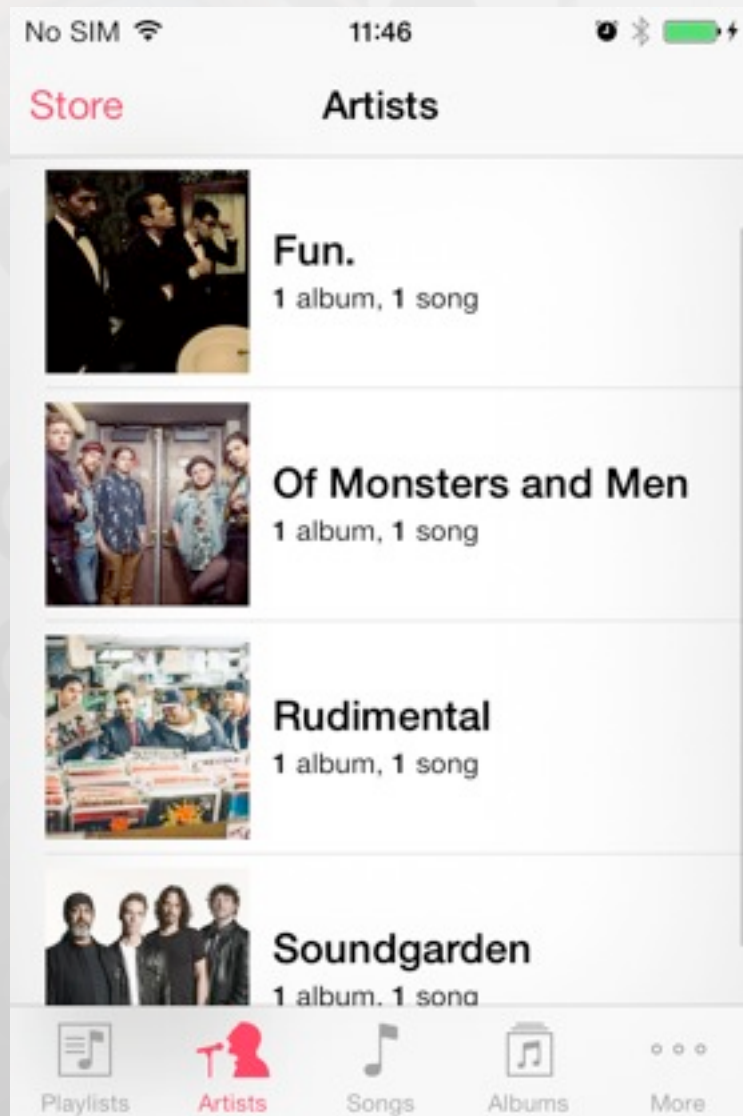iPhone Application Programming • Prof. Jan Borchers

# View Controller Showcase

- Tab Bar Controller

- Navigation Controller
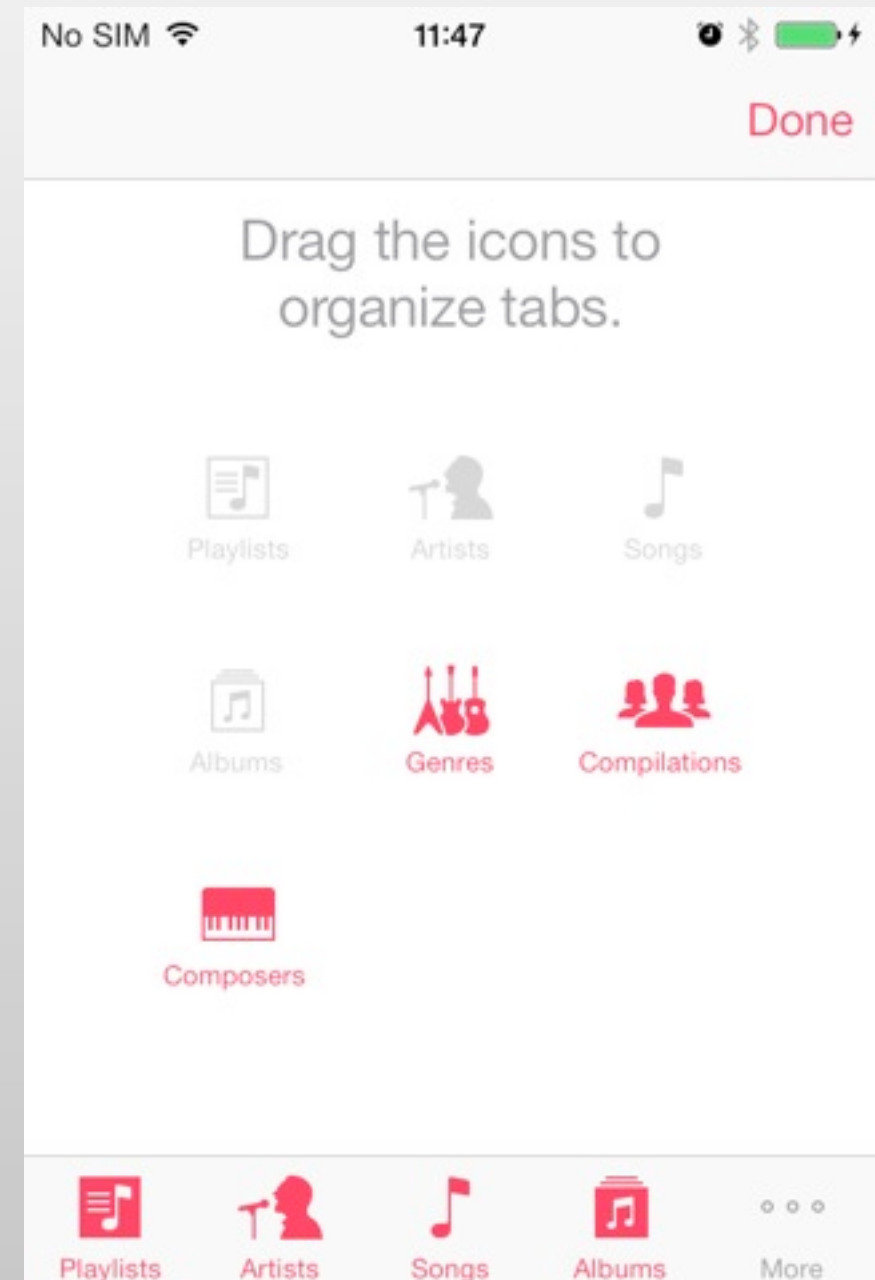
- Table View Controller

# Tab Bar Controller

# Tab Bar Controller

# More Tab Items

- More than 5 view controllers cannot be shown

- "More" tab is shown automatically
  - User can navigate remaining view controllers and customize the order

# Creating a Tab Bar Controller
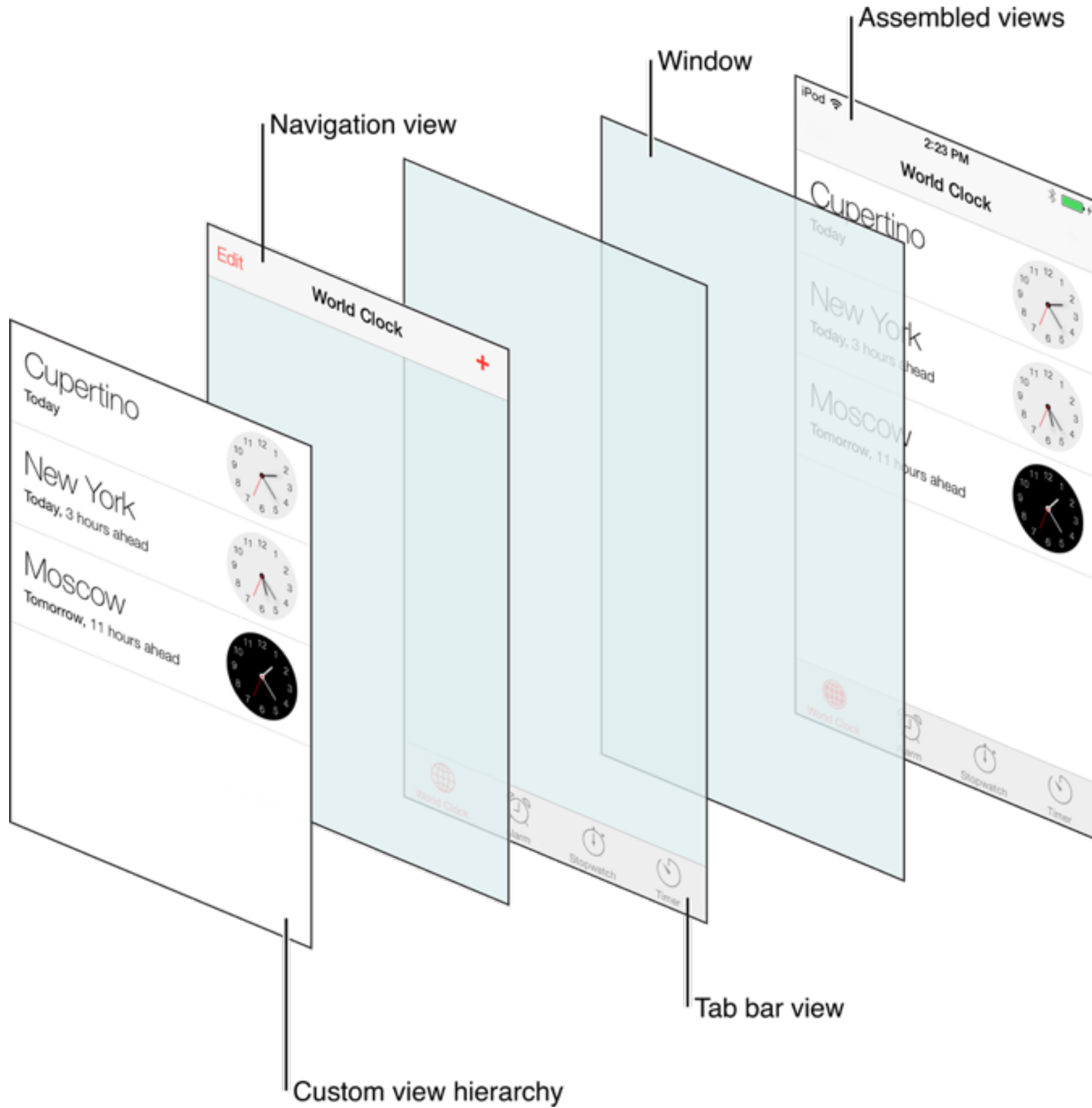
```objc
   // Create a tab bar controller
tabBarController = [[UITabBarController alloc] init];

// Set the array of view controllers
NSArray *myViewControllers = ...
tabBarController.viewControllers = myViewControllers;

// Add the tab bar controller's view to the window
[window addSubview:tabBarController.view];
```
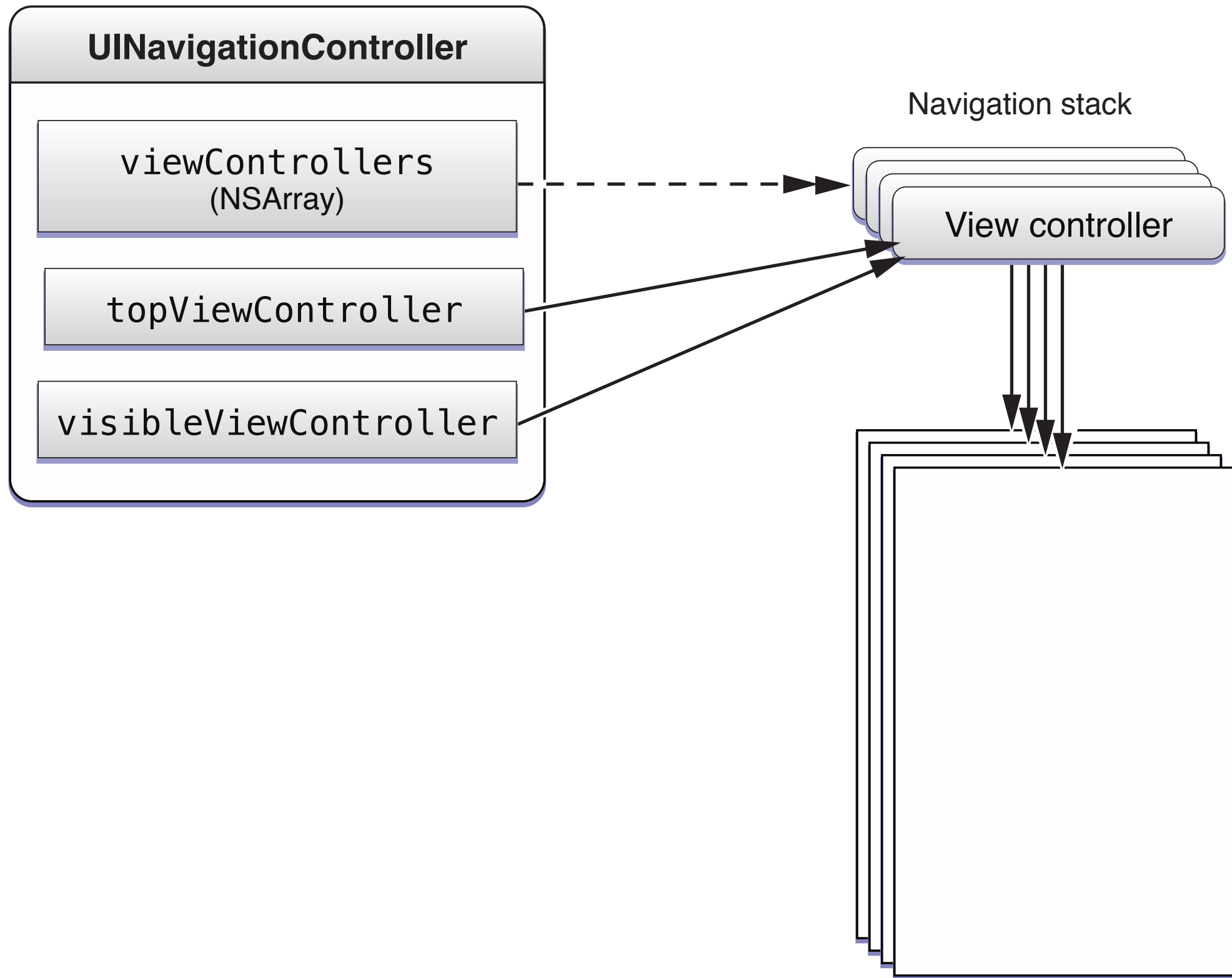
RWTHAACHEN
UNIVERSITY

# Navigation View Controller

RWTHAACHEN
UNIVERSITY

Assembled views

Window

Navigation view

Cupertino
Today

New York
Today, 3 hours ahead

Moscow
Tomorrow, 11 hours ahead

Edit
World Clock

iPod
2:23 PM
World Clock

Cupertino
Today

New York
Today, 3 hours ahead

Moscow
Tomorrow, 11 hours ahead

Tab bar view

Custom view hierarchy

[View Controller Programming Guide for iOS]

## UINavigationController

viewControllers
(NSArray)

topViewController

visibleViewController

Navigation stack

View controller

[View Controller Programming Guide for iOS]

# Creating the Navigation Controller

```objc
- (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {

// set up the root view controller
rootViewController = [[MyViewController alloc] init];

// set up the navigation controller
navigationController = [[UINavigationController alloc]
  initWithRootViewController:rootViewController];

// Add the view controller's view to the window and display.
[self.window addSubview:navigationController.view];
[self.window makeKeyAndVisible];

return YES;
}
```

RWTHAACHEN
UNIVERSITY

# Working with the Navigation Stack

```objc
    // load the view controller
UIViewController *myViewController = [UIViewController
alloc];[myViewController initWithNibName:@"MyNib"
bundle:nil];

// push it to the navigation stack
[self.navigationController
pushViewController:myViewController
                                animated:YES];

// we can release the view controller now
[myViewController release];

...

// pop the top view controller from the stack
[self.navigationController popViewControllerAnimated:YES];
```
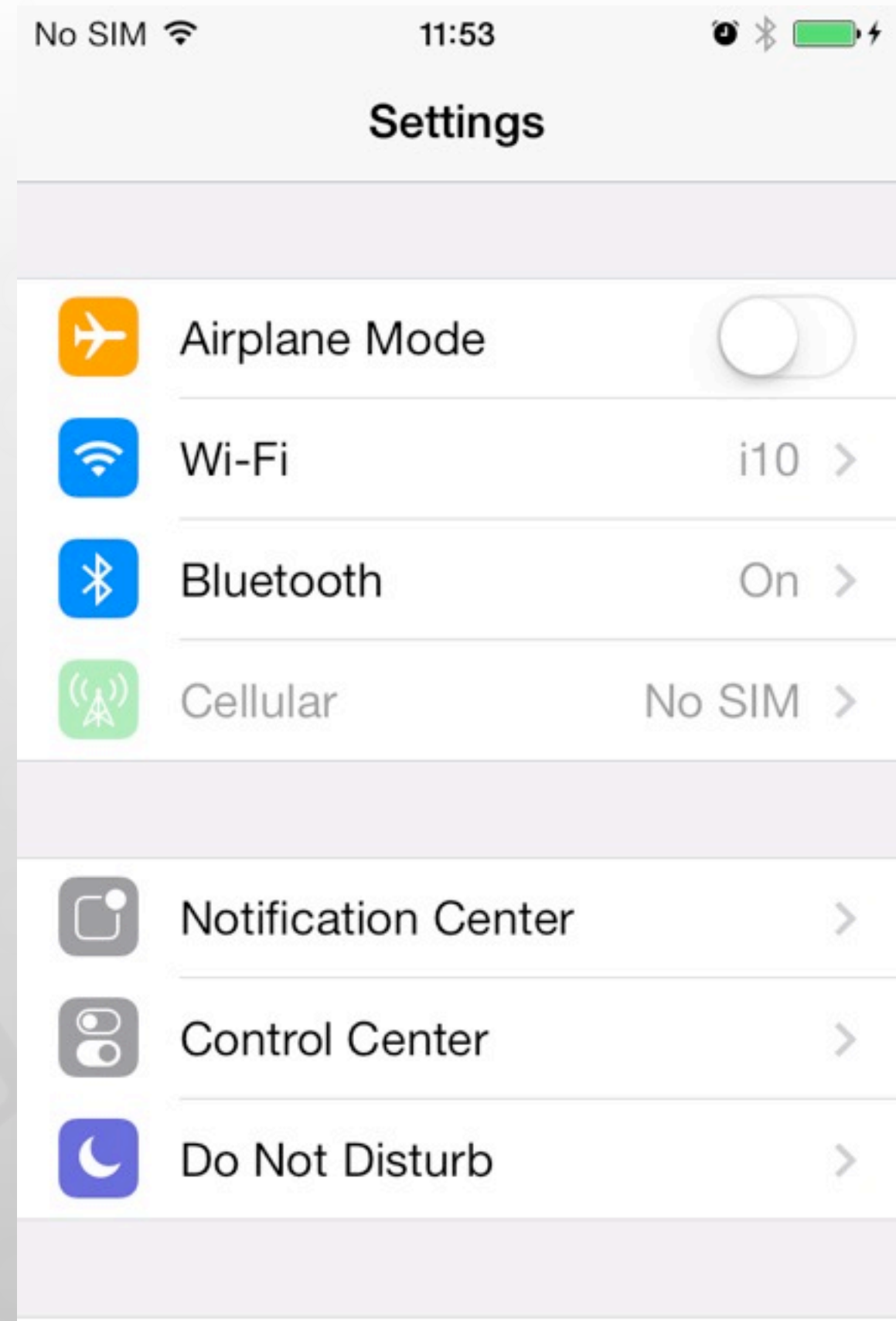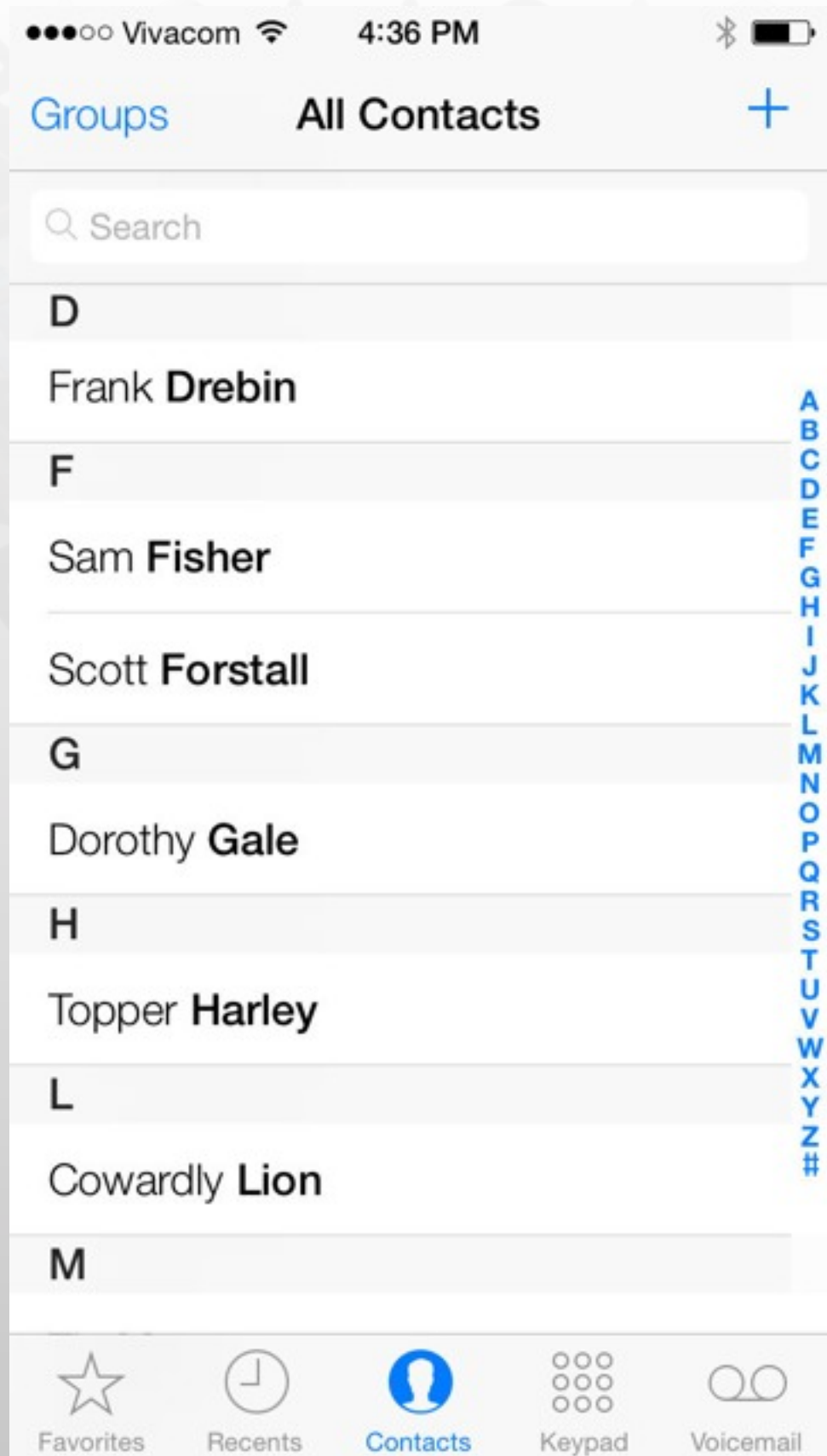
# Table View Controller

iPhone  Application Programming   •   Prof. Jan Borchers

iPhone Application Programming   •   Prof. Jan Borchers

# Table Views

UITableViewDelegate ↔ UITableViewController

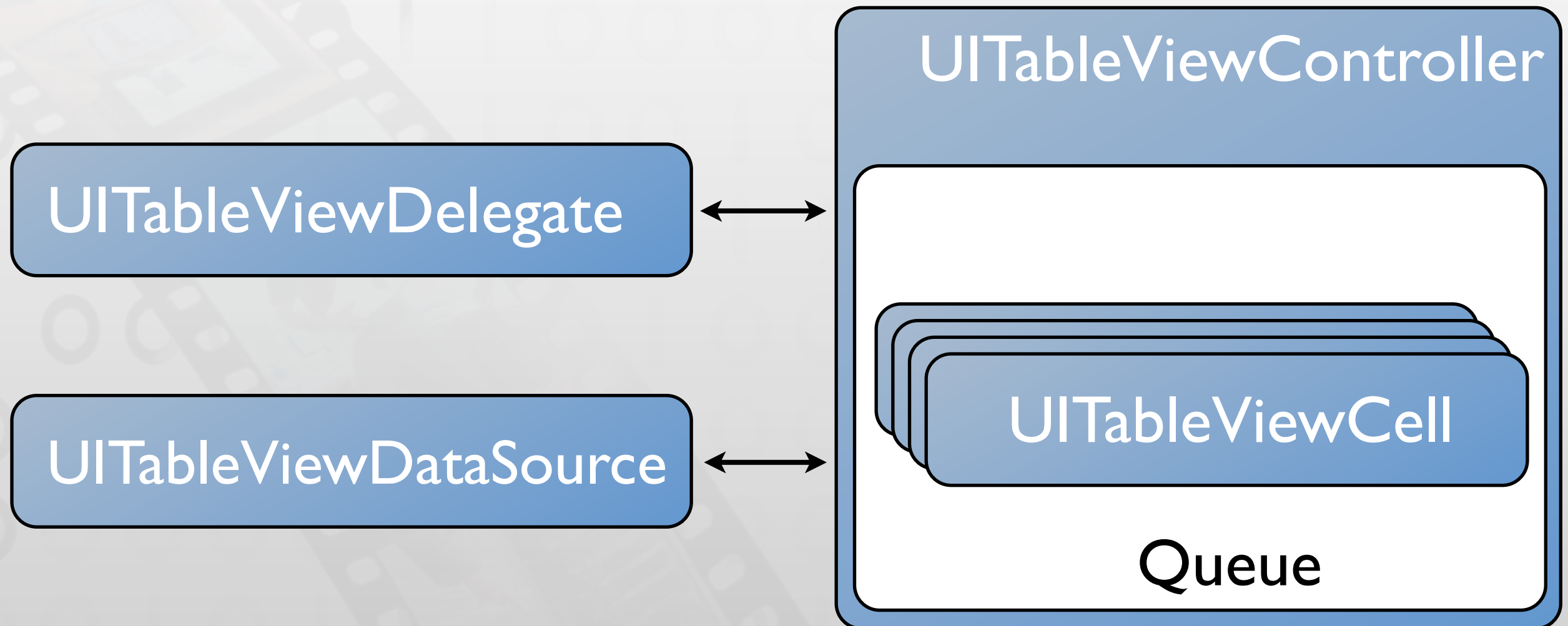UITableViewDataSource ↔ UITableViewCell

Queue

# Table View Delegate

- Conform to UITableViewDelegate protocol

- Respond to selection
  - `tableView:didSelectRowAtIndexPath:`
  - `tableView:accessoryButtonTappedForRowWithIndexPath:`

- Manage header and footer views

# Table View Data Source

- Conform to `UITableViewDataSource` protocol

- Provide size of data
  - `numberOfSectionsInTableView:`
  - `tableView:numberOfRowsInSection:`

- Provide data
  - `tableView:cellForRowAtIndexPath:`

- Manage editing
  - `tableView:canEditRowAtIndexPath:`
  - `tableView:commitEditingStyle:forRowAtIndexPath:`

# UITableViewCell Styles

- `UITableViewCellStyleDefault`

- `UITableViewCellStyleValue1`

- `UITableViewCellStyleValue2`
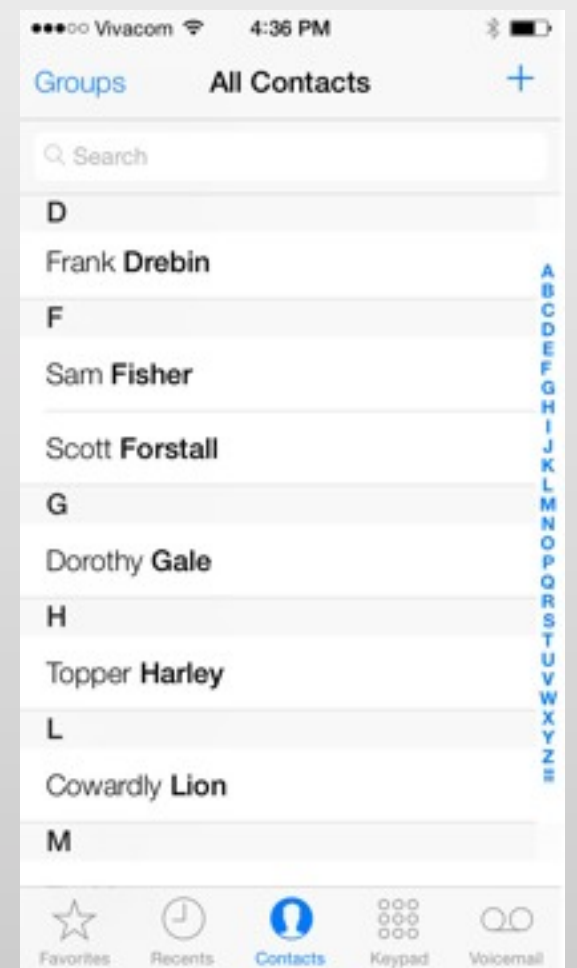
- `UITableViewCellStyleSubtitle`

| Title |
|---|

| Title | Detail |
|---|---|

| | Title Detail |
|---|---|

| Title |
|---|
| Detail |

RWTHAACHEN
UNIVERSITY

# Memory Management

UITableViewCell



- Views are reused
  - `dequeueReusableCellWithIdentifier:`
  - Use different identifiers for different cell styles

# UITableViewController

- `UIViewController` for table views

  - Delegate and data source

- Does not need a Nib file

  - `initWithStyle:`

- Takes care of some standard operations

  - Calls `reloadData` on appearing

  - Deselect rows in combination with navigation controller

  - Flashes scroll indicators

  - Powerful combination with Core Data (later)
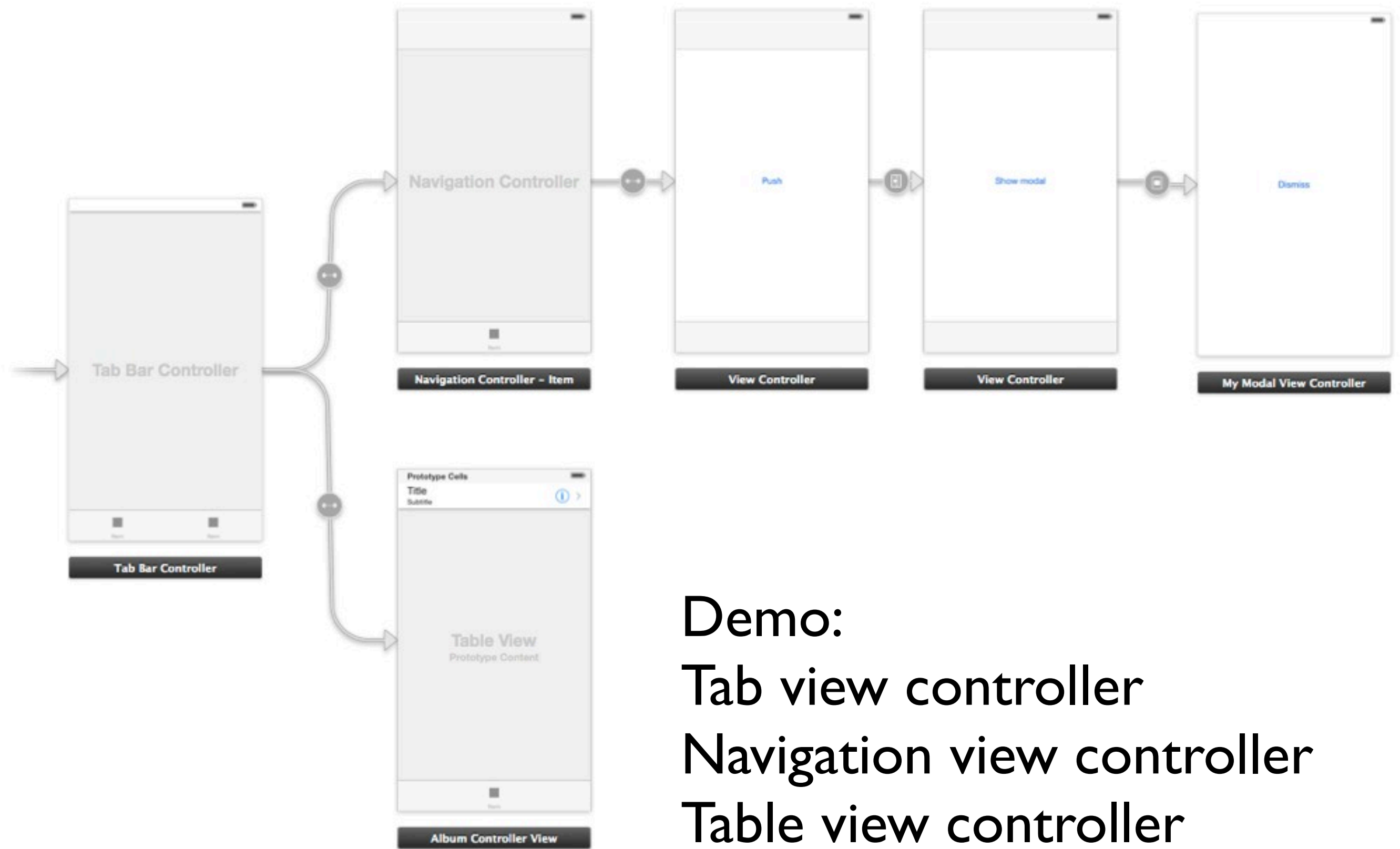
Demo:
Tab view controller
Navigation view controller
Table view controller

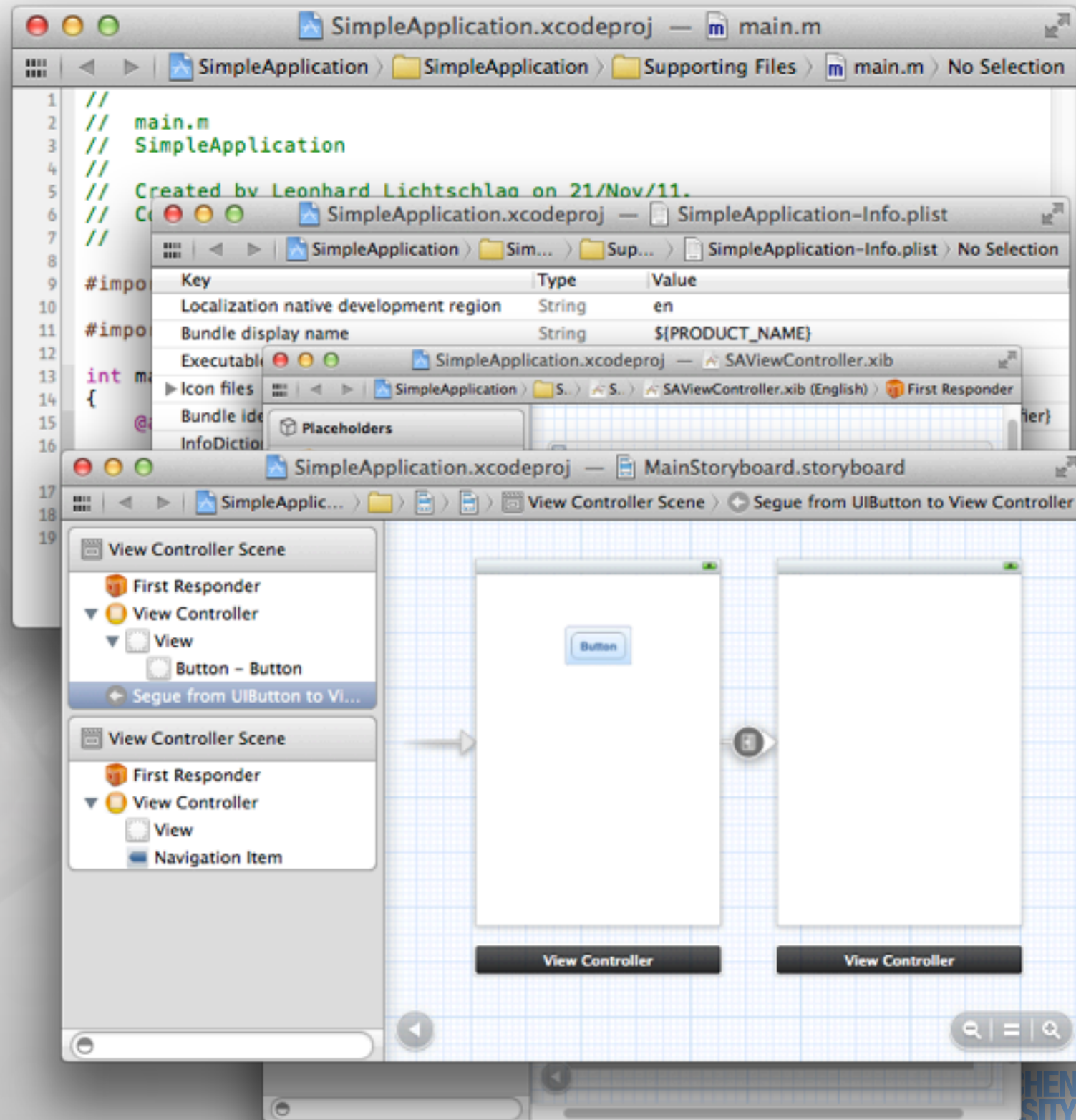# Application Life Cycle

# Application Components
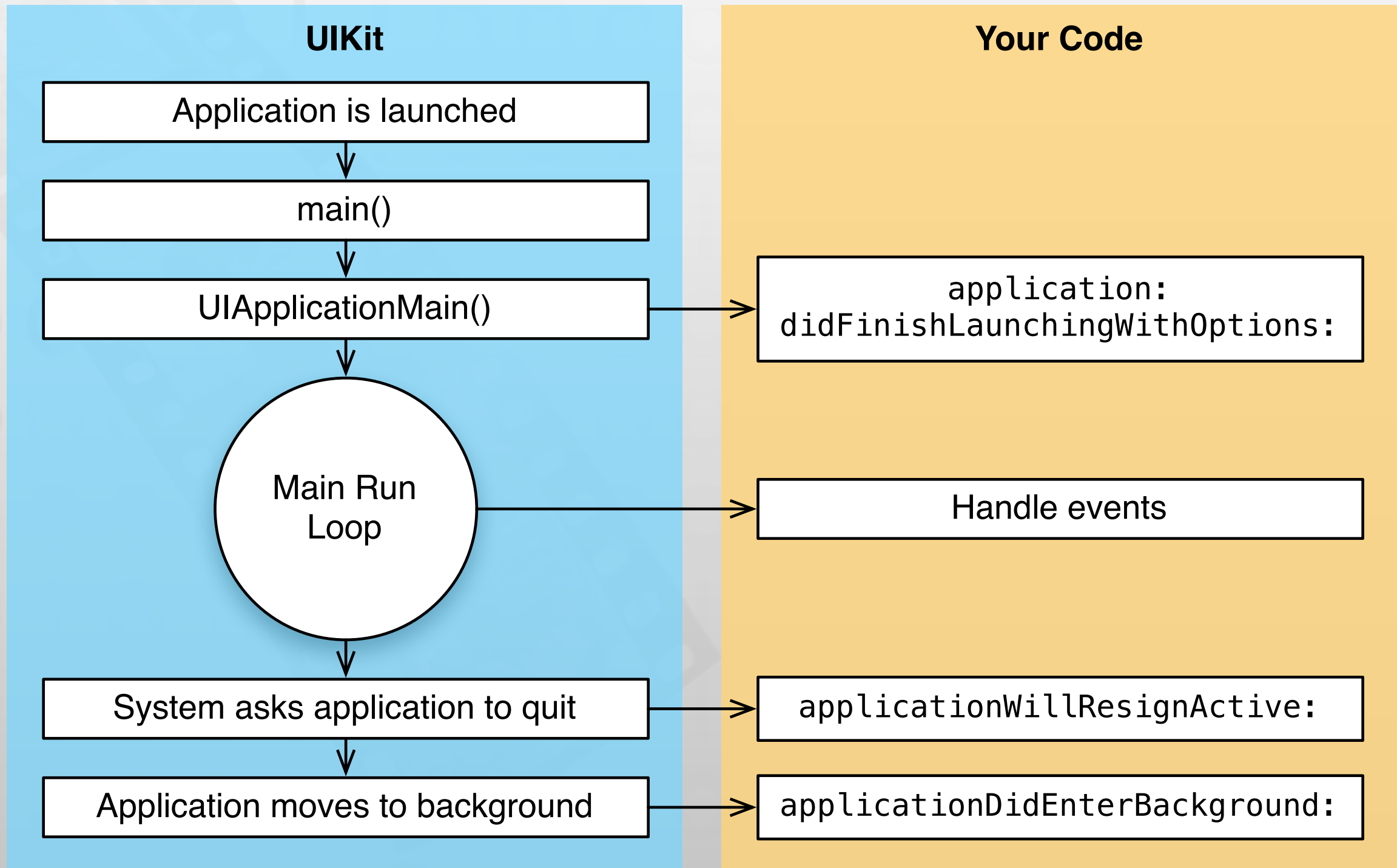
main.m

- UIApplication
- CFRunLoop

Info.plist

MainWindow.xib

- UIApplicationDelegate
- UIWindow

Main.storyboard

# Application Life Cycle

| UIKit | Your Code |
|---|---|
| Application is launched | |
| main() | |
| UIApplicationMain() | `application:`<br>`didFinishLaunchingWithOptions:` |
| Main Run Loop | Handle events |
| System asks application to quit | `applicationWillResignActive:` |
| Application moves to background | `applicationDidEnterBackground:` |

RWTHAACHEN
UNIVERSITY
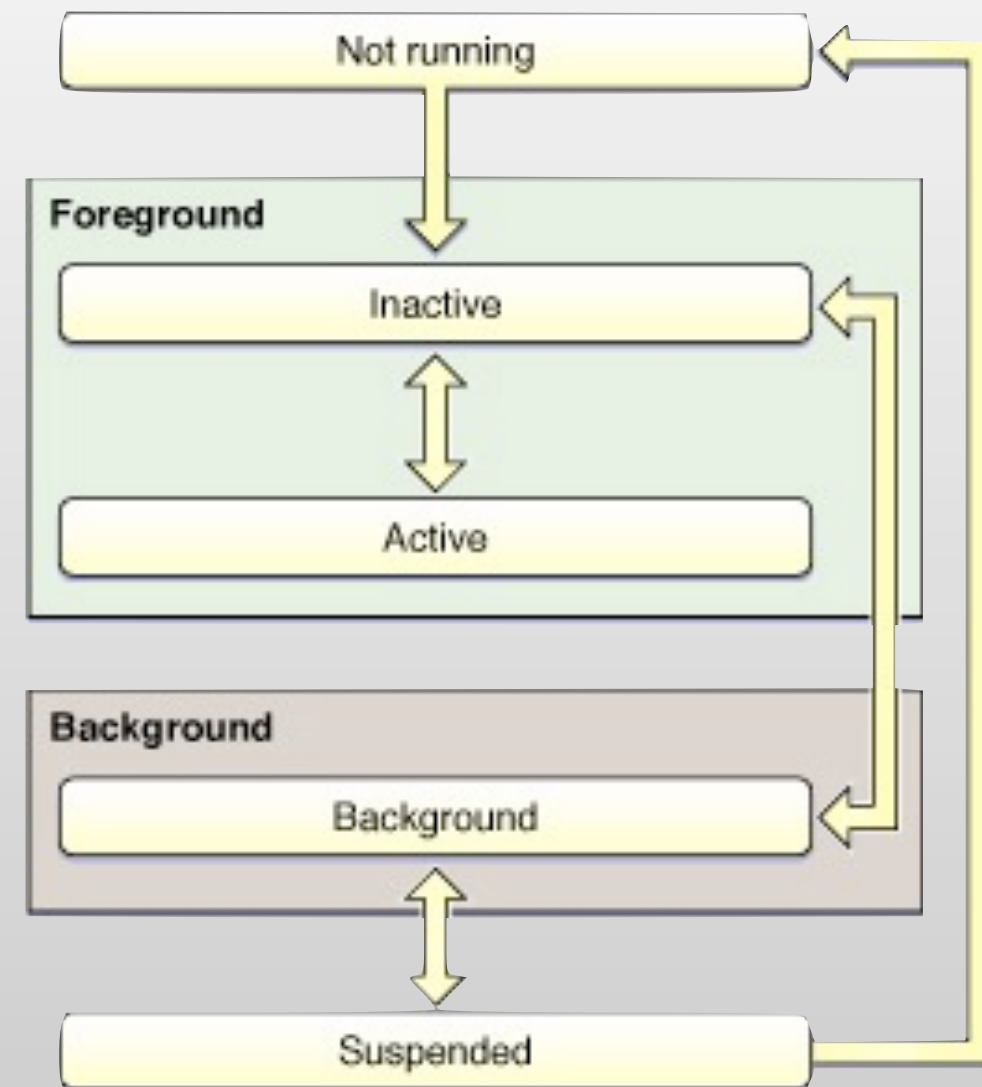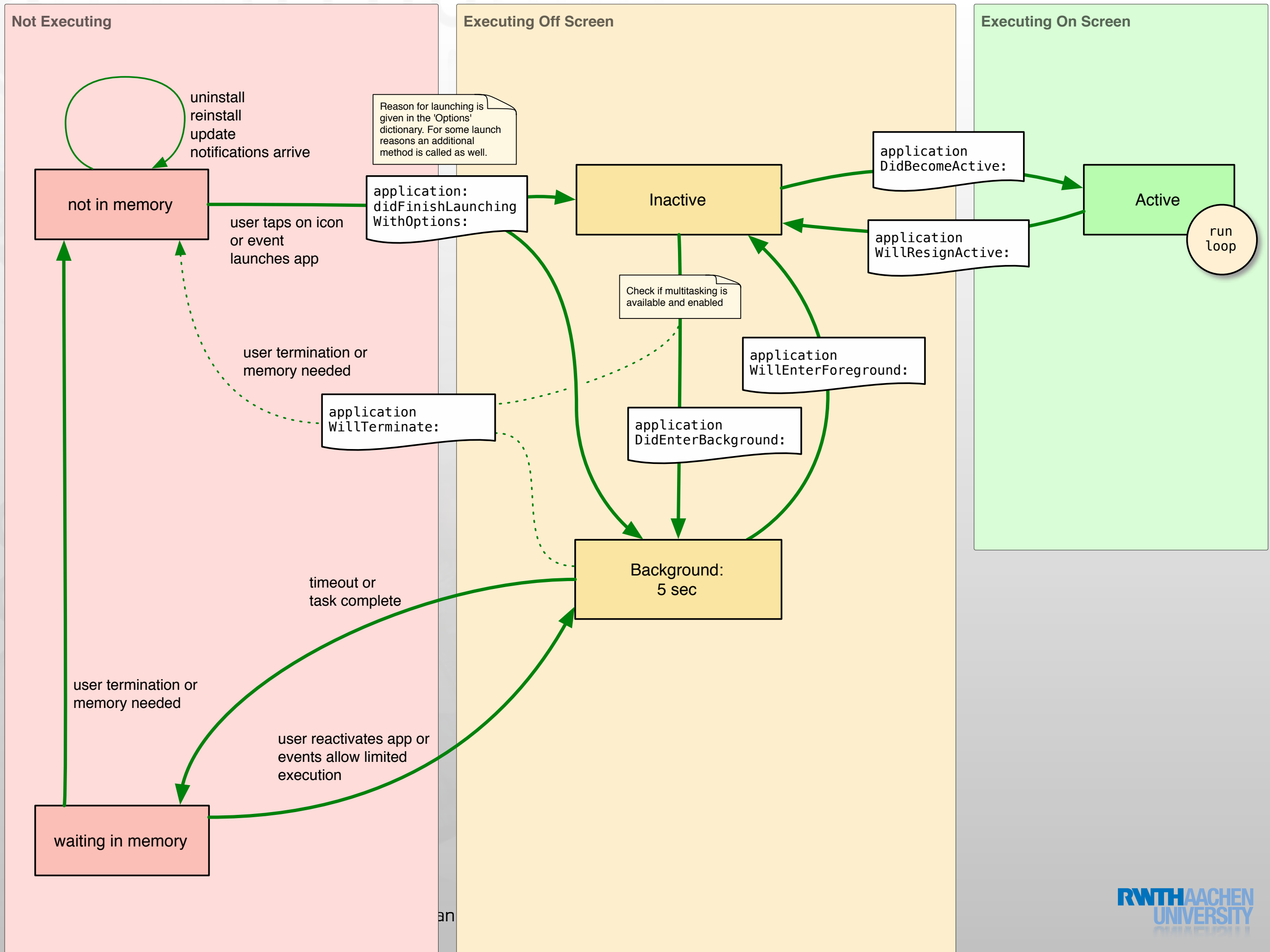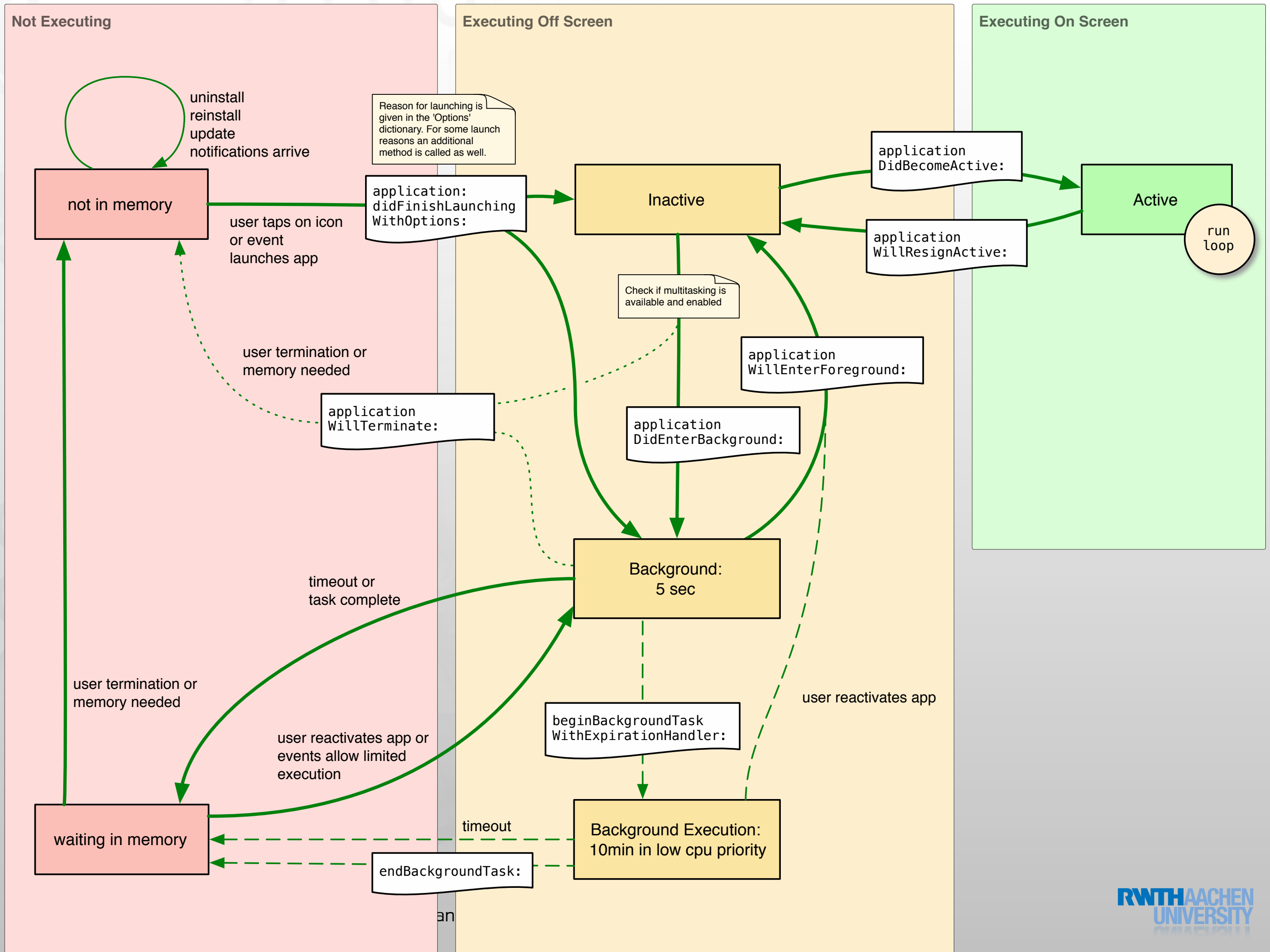
# Background Apps

- Complete long-running tasks

- Schedule local notifications

- Receive location updates

- Play background audio

- Implement VoIP application

- Newsstand apps



```
Check [[UIApplication sharedApplication] applicationState]
and [[UIDevice currentDevice] isMultitaskingSupported]
```

# Not Executing

# Executing Off Screen

# Executing On Screen

uninstall
reinstall
update
notifications arrive

**not in memory**

Reason for launching is
given in the 'Options'
dictionary. For some launch
reasons an additional
method is called as well.

```
application:
didFinishLaunching
WithOptions:
```

**Inactive**

```
application
DidBecomeActive:
```

**Active**

```
application
WillResignActive:
```

run
loop

user taps on icon
or event
launches app

user termination or
memory needed

Check if multitasking is
available and enabled

```
application
WillEnterForeground:
```

```
application
WillTerminate:
```

```
application
DidEnterBackground:
```

**Background:**
**5 sec**

timeout or
task complete

user termination or
memory needed

user reactivates app or
events allow limited
execution

**waiting in memory**

# Demo

# Summary

- View Controllers

  - Tab Bar Controllers

  - Navigation Controller

  - Table View Controller

- Storyboard demos

- Application Life Cycle

- Reading assignment:

  View Controller Programming Guide

  iPhone Application Programming Guide