Frameworks

#WWDC14

A Look Inside Presentation Controllers

With a case study on alerts and search

Session 228

Peter Hajas
UIKit Engineer

Jim Turner
UIKit Engineer

© 2014 Apple Inc. All rights reserved. Redistribution or public display not permitted without written permission from Apple.

Intro

Intro

Animator objects

Intro

Animator objects

- Custom presentation difficult

Intro

Animator objects

- Custom presentation difficult

UIPresentationController

Agenda

Agenda

Presentation basics

Agenda

Presentation basics

UIKit presentations

Agenda

Presentation basics

UIKit presentations

Demo

iPad

9:41 AM

Settings

- Notifications
- Control Center
- Do Not Disturb
- General
- Wallpaper & Brightness
- Sounds
- Passcode
- Privacy
- iCloud
- iTunes & App Store
- Mail, Contacts, Calendars
- Notes
- Reminders

iTunes & App Store

Apple ID example@icloud.com

Password required

Sign In

Forgot Apple ID or Password?

Create New Apple ID

An Apple ID is the login you use for just about everything you do with Apple.

AUTOMATIC DOWNLOADS

Updates

Use Cellular Data

Use cellular network for automatic downloads, iTunes Match, and iTunes Radio.

iPad

9:41 AM

Settings

- Notifications
- Control Center
- Do Not Disturb
- General
- Wallpaper & Brightness
- Sounds
- Passcode
- Privacy
- iCloud
- iTunes & App Store
- Mail, Contacts, Calendars
- Notes
- Reminders

iTunes & App Store

Apple ID example@icloud.com

Password required

Sign In

Forgot Apple ID or Password?

Create New Apple ID

An Apple ID is the login you use for just about everything you do with Apple.

AUTOMATIC DOWNLOADS

Updates

Use Cellular Data

Use cellular network for automatic downloads, iTunes Match, and iTunes Radio.

iPad

9:41 AM

Settings

iTunes & App Store

Notifications

Control Center

Do Not Disturb

General

Wallpaper & Brightness

Sounds

Passcode

Privacy

iCloud

iTunes & App Store

Mail, Contacts, Calenda

Notes

Reminders

Apple.

es Radio.

Cancel

New Account

CONFIRM YOUR COUNTRY OR REGION

Choose a country or region for the Store that will match the billing address for your payment method, then tap Next.

United States

Albania

Algeria

Angola

Anguilla

Antigua and Barbuda

Argentina

Armenia

Australia

Azerbaijan

Next

Presented

iPad 9:41 AM

Settings

iTunes & App Store

Notifications

Control Center

Do Not Disturb

General

Wallpaper & Brightness

Sounds

Passcode

Privacy

iCloud

iTunes & App Store

Mail, Contacts, Calenda

Notes

Reminders

Cancel

New Account

CONFIRM YOUR COUNTRY OR REGION

Choose a country or region for the Store that will match the billing address for your payment method, then tap Next.

United States ✓

Albania

Algeria

Angola

Anguilla

Antigua and Barbuda

Argentina

Armenia

Australia

Azerbaijan

Next

Apple.

es Radio.

Presented

Presenting

iPad 9:41 AM

Settings

Notifications
Control Center
Do Not Disturb
General
Wallpaper & Brightness
Sounds
Passcode
Privacy
iCloud
iTunes & App Store
Mail, Contacts, Calenda
Notes
Reminders

iTunes & App Store

Apple.

es Radio.

Cancel

New Account

CONFIRM YOUR COUNTRY OR REGION

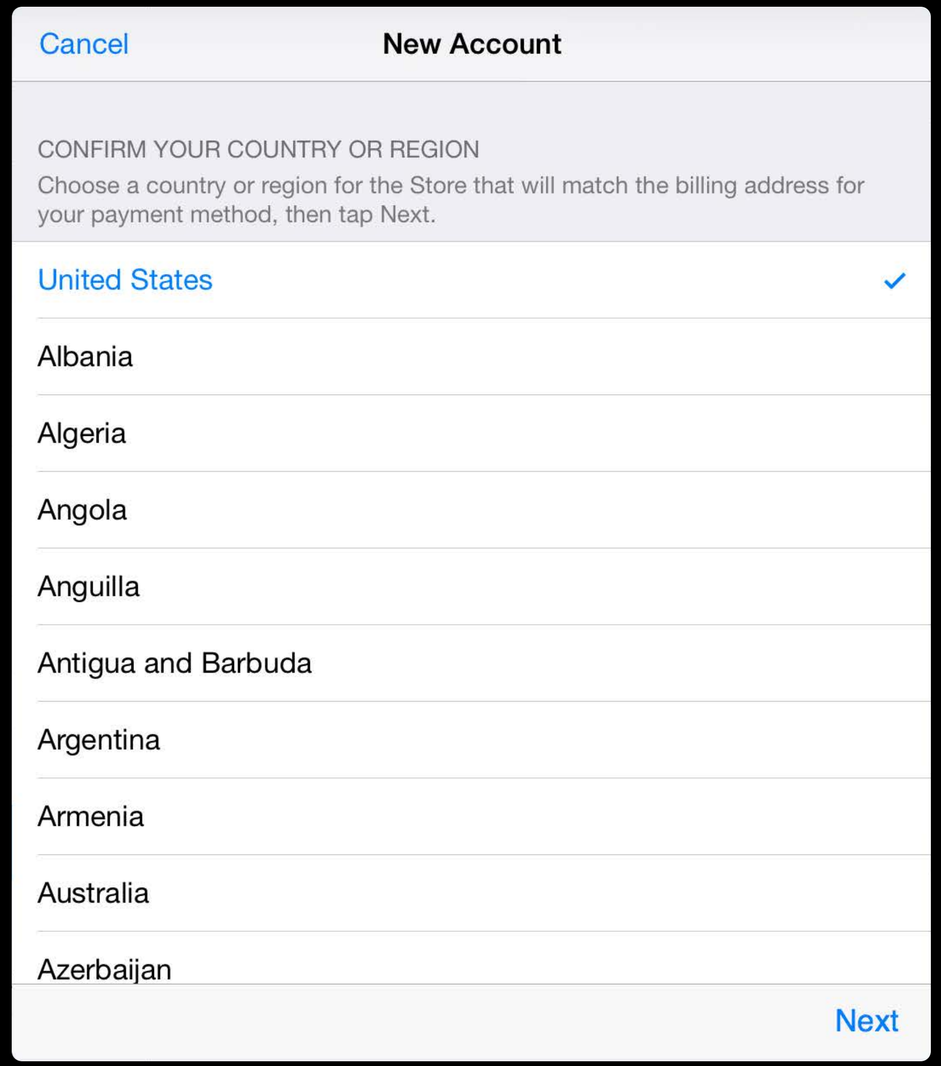
Choose a country or region for the Store that will match the billing address for your payment method, then tap Next.

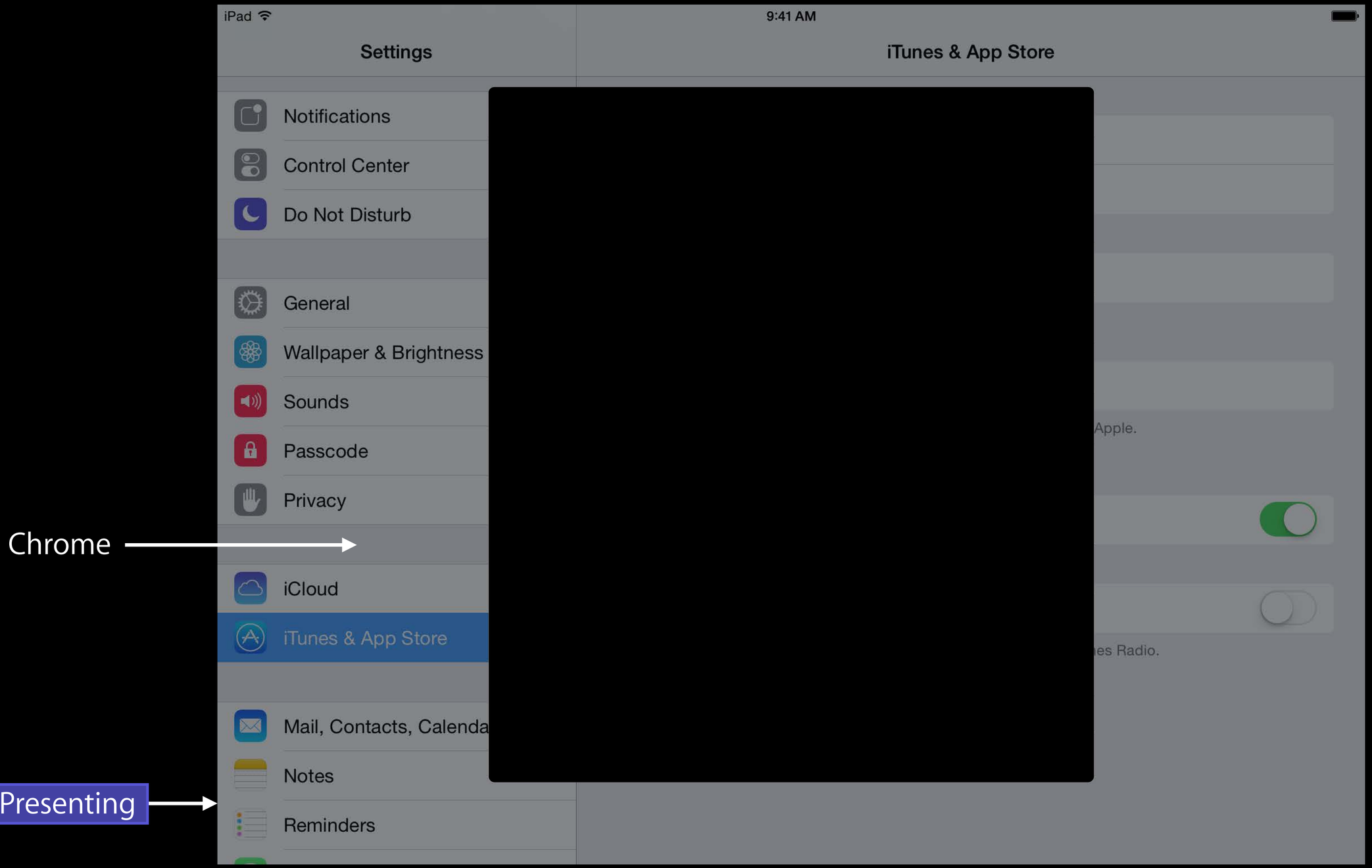
United States ✓
Albania
Algeria
Angola
Anguilla
Antigua and Barbuda
Argentina
Armenia
Australia
Azerbaijan

Next

Presented

Content




iPad
9:41 AM
Settings
iTunes & App Store
Notifications
Control Center
Do Not Disturb
General
Wallpaper & Brightness
Sounds
Passcode
Privacy
iCloud
iTunes & App Store
Mail, Contacts, Calenda
Notes
Reminders
Apple.
es Radio.
Chrome
Presenting

Hello, UIPresentationController

Hello, UIPresentationController

Presentation management

Hello, UIPresentationController

Presentation management

Animates

Hello, UIPresentationController

Presentation management

Animates

Trait and size changes

Hello, UIPresentationController

Presentation management

Animates

Trait and size changes

Reusable

UIPresentationController

Division of responsibilities

UIPresentationController

Division of responsibilities

Animator object

UIPresentationController

Division of responsibilities

Animator object

Animating controller content

UIPresentationController

Division of responsibilities

Animator object

Animating controller content

Positioning

UIPresentationController

Division of responsibilities

Animator object

Animating controller content

Positioning

Chrome

UIPresentationController

Division of responsibilities

Animator object	Presentation controller
Animating controller content	
	Positioning
	Chrome

UIPresentationController

Division of responsibilities

Animator object	Presentation controller
Animating controller content	Positioning
	Chrome

UIPresentationController

Division of responsibilities

Animator object	Presentation controller
Animating controller content	Positioning
	Chrome
	Adaptation

UIKit Presentations

Popovers, alerts, action sheets, and search

UIPopoverPresentationController

UIPopoverPresentationController

Replacing UIPopoverController

UIPopoverPresentationController

Replacing UIPopoverController

Functionally equivalent

UIPopoverPresentationController

Replacing UIPopoverController

Functionally equivalent

Built-in adaptivity

UIPopoverController

Old API

```
UIViewController *contentController = …;

UIPopoverController *poc = [[UIPopoverController alloc]
initWithContentViewController:contentController];
[poc presentPopoverFromBarButtonItem:item
    permittedArrowDirections:UIPopoverArrowDirectionAny animated:YES];
```

UIPopoverController

Old API

```
UIViewController *contentController = …;

if([[UIDevice currentDevice] userInterfaceIdiom] == UIUserInterfaceIdiomPad)
{
    UIPopoverController *poc = [[UIPopoverController alloc]
initWithContentViewController:contentController];

    [poc presentPopoverFromBarButtonItem:item
permittedArrowDirections:UIPopoverArrowDirectionAny animated:YES];
}

else
{
    [self presentViewController:contentController …];
}
```

UIPopoverPresentationController

New API


```
contentController.modalPresentationStyle = UIModalPresentationPopover;

UIPopoverPresentationController *popPC =
contentController.popoverPresentationController;
popPC.barButtonItem = item;
popPC.permittedArrowDirections = UIPopoverArrowDirectionAny;

[self presentViewController:contentController animated:YES completion:nil];
```

UIPopoverPresentationController

New API


```
contentController.modalPresentationStyle = UIModalPresentationPopover;

UIPopoverPresentationController *popPC =
contentController.popoverPresentationController;
popPC.barButtonItem = item;
popPC.permittedArrowDirections = UIPopoverArrowDirectionAny;

[self presentViewController:contentController animated:YES completion:nil];
```

UIPopoverPresentationController

New API


```
contentController.modalPresentationStyle = UIModalPresentationPopover;

UIPopoverPresentationController *popPC =
contentController.popoverPresentationController;
popPC.barButtonItem = item;
popPC.permittedArrowDirections = UIPopoverArrowDirectionAny;

[self presentViewController:contentController animated:YES completion:nil];
```

UIPopoverPresentationController

New API


```
contentController.modalPresentationStyle = UIModalPresentationPopover;

UIPopoverPresentationController *popPC =
contentController.popoverPresentationController;
popPC.barButtonItem = item;
popPC.permittedArrowDirections = UIPopoverArrowDirectionAny;

[self presentViewController:contentController animated:YES completion:nil];
```

UIPopoverPresentationController

New API


```
contentController.modalPresentationStyle = UIModalPresentationPopover;

UIPopoverPresentationController *popPC =
contentController.popoverPresentationController;
popPC.barButtonItem = item;
popPC.permittedArrowDirections = UIPopoverArrowDirectionAny;

[self presentViewController:contentController animated:YES completion:nil];
```

Demo

Adaptive popovers

UIPopoverPresentationController

P

Fullscreen adaptation

```
contentController.modalPresentationStyle = UIModalPresentationPopover;

UIPopoverPresentationController *popPC =
contentController.popoverPresentationController;
popPC.barButtonItem = item;
popPC.permittedArrowDirections = UIPopoverArrowDirectionAny;

[self presentViewController:contentController animated:YES completion:nil];
```

UIPopoverPresentationController

P

Fullscreen adaptation

```
contentController.modalPresentationStyle = UIModalPresentationPopover;

UIPopoverPresentationController *popPC =
contentController.popoverPresentationController;
popPC.barButtonItem = item;
popPC.permittedArrowDirections = UIPopoverArrowDirectionAny;
popPC.delegate = self;

[self presentViewController:contentController animated:YES completion:nil];
```

UIPopoverPresentationController

P

Fullscreen adaptation

```
- (UIModalPresentationStyle)adaptivePresentationStyleForPresentationController:
(UIPresentationController *)controller
{
    return UIModalPresentationFullScreen;
}
```

UIPopoverPresentationController

P

Fullscreen adaptation

```
- (UIModalPresentationStyle)adaptivePresentationStyleForPresentationController:
(UIPresentationController *)controller
{
    return UIModalPresentationFullScreen;
}
```

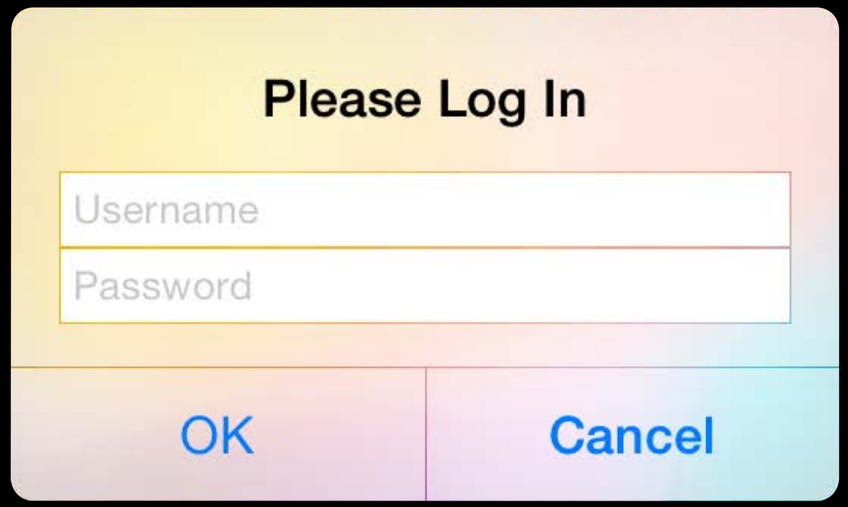
```
- (UIViewController *)presentationController:(UIPresentationController
*)controller viewControllerForAdaptivePresentationStyle:
(UIModalPresentationStyle)style
{
    UINavigationController *navController = [[UINavigationController alloc]
initWithRootViewController:controller.presentedViewController];
    return navController;
}
```

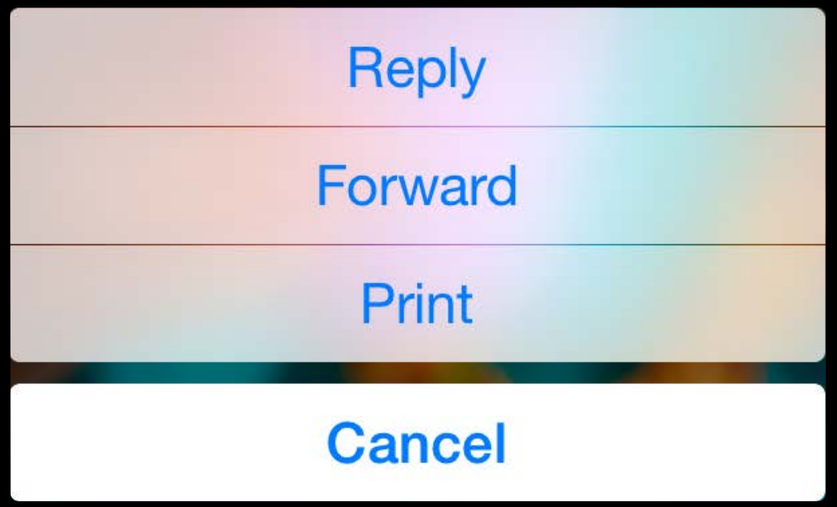
Alerts and Action Sheets

UIAlertView and UIActionSheet

Alerts and Action Sheets

UIAlertView and UIActionSheet

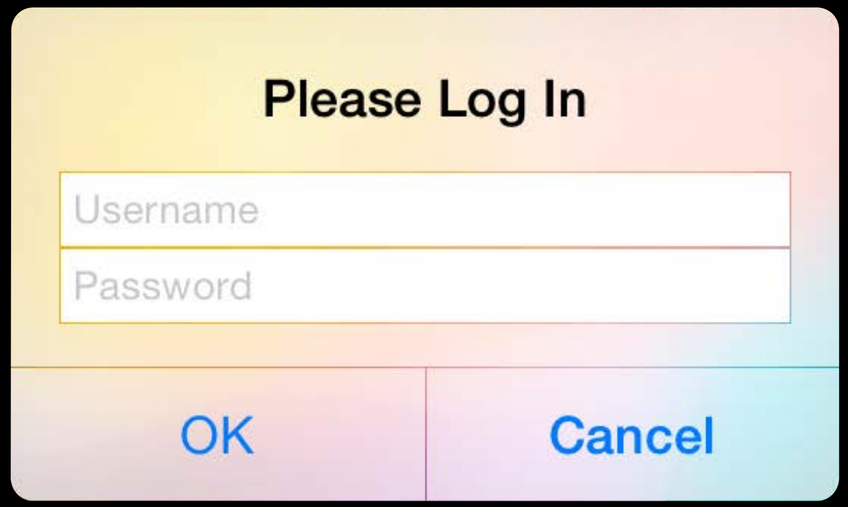


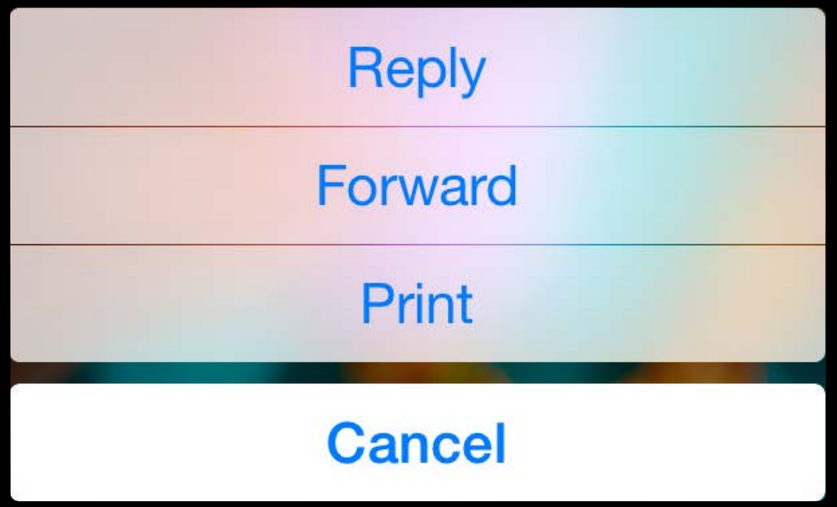


Alerts and Action Sheets

UIAlertView and UIActionSheet

Used to be plain views



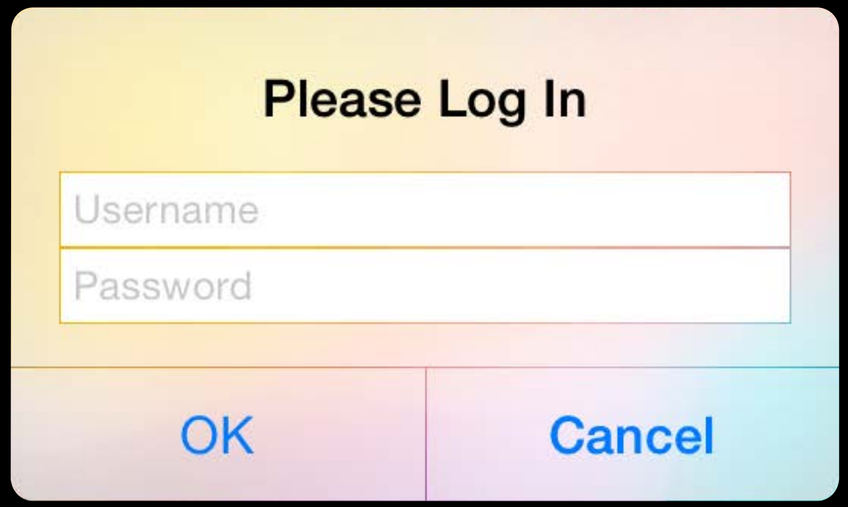


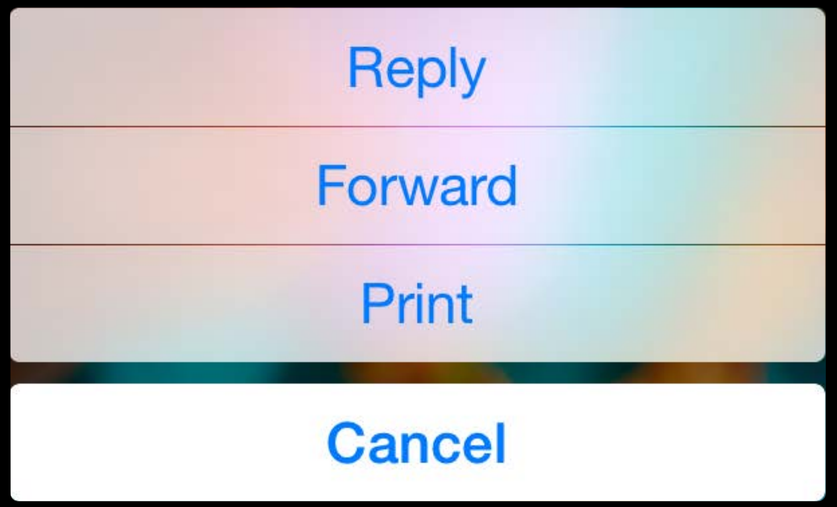
Alerts and Action Sheets

UIAlertView and UIActionSheet

Used to be plain views

New window for presentation





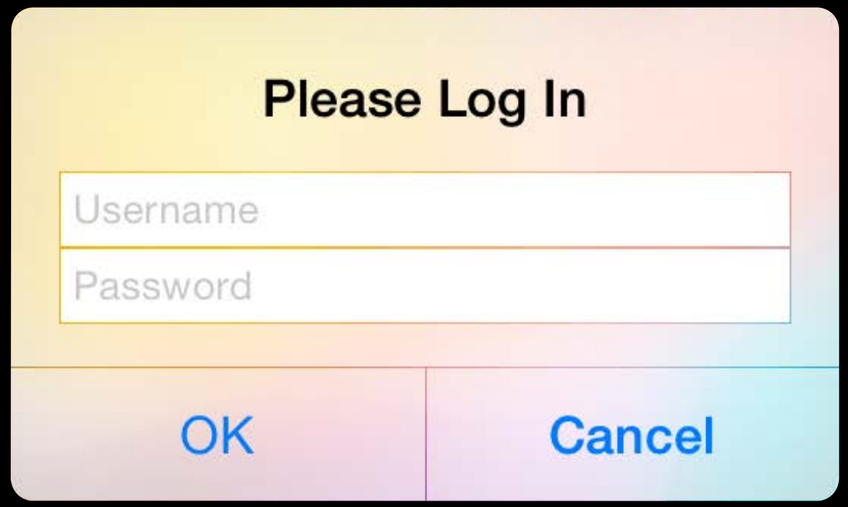
Alerts and Action Sheets

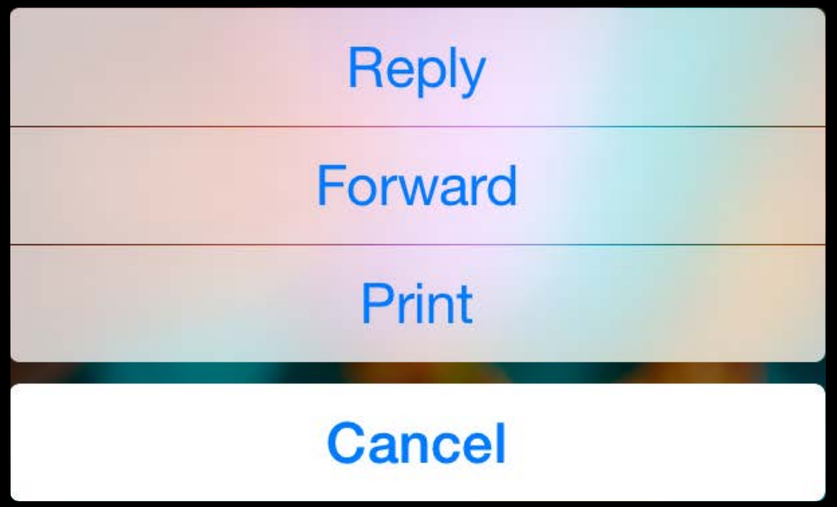
UIAlertView and UIActionSheet

Used to be plain views

New window for presentation

Predate modern language features



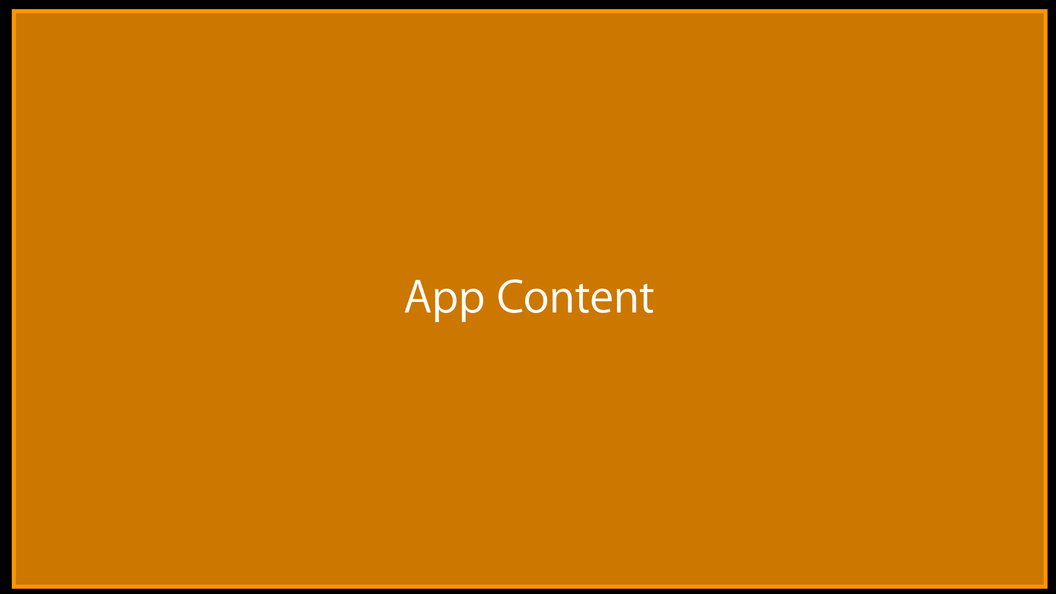


Alerts and Action Sheets

Old alerts presentation

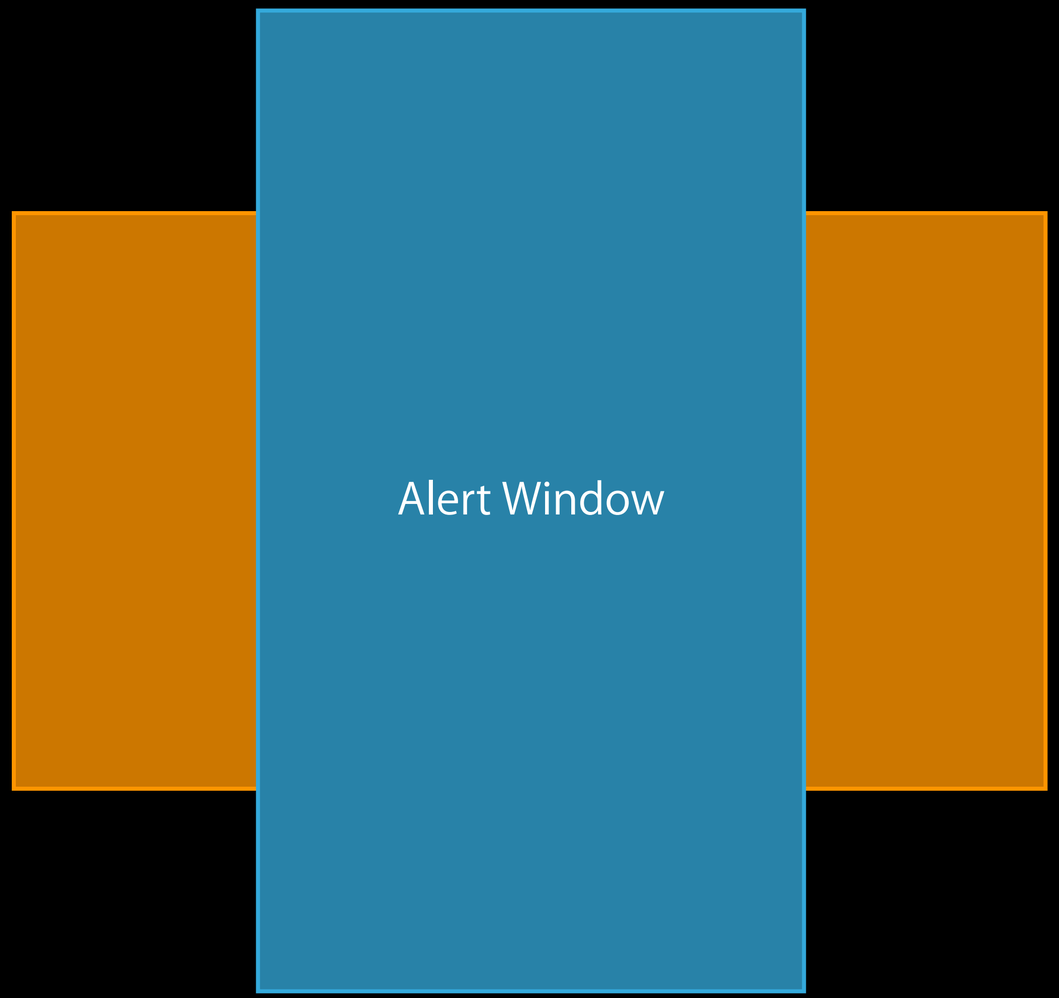
Alerts and Action Sheets

Old alerts presentation



Alerts and Action Sheets

Old alerts presentation



Alerts and Action Sheets

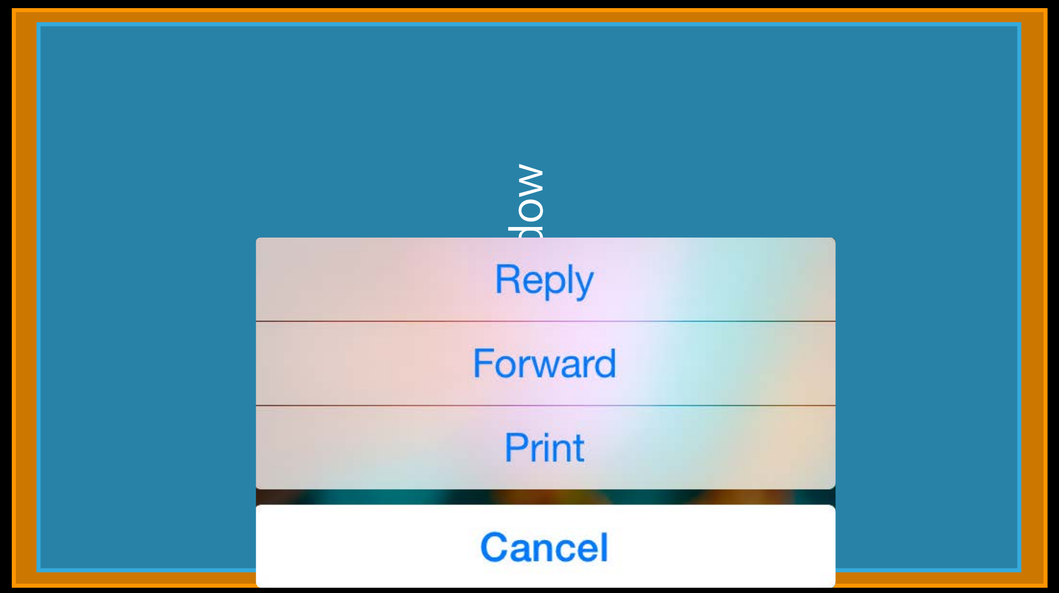
Old alerts presentation

Alerts and Action Sheets

Old alerts presentation

Alerts and Action Sheets

Old alerts presentation



UIAlertController

UIAlertController

Used for alerts and action sheets

UIAlertController

Used for alerts and action sheets

Block based

UIAlertController

Used for alerts and action sheets

Block based

Presents in your window

UIAlertController

Used for alerts and action sheets

Block based

Presents in your window

Action sheets adapt to popovers

UIAlertController

New API


```
UIAlertController* sheet = [UIAlertController alertControllerWithTitle:...];
[sheet addAction:[UIAlertAction actionWithTitle:@“Reply”
style:UIAlertActionStyleDefault handler:^void (UIAlertAction *action)
{
    // do stuff
}];

[self presentViewController:sheet...];
```

UIAlertController

New API


```
UIAlertController* sheet = [UIAlertController alertControllerWithTitle:...];
[sheet addAction:[UIAlertAction actionWithTitle:@“Reply”
style:UIAlertActionStyleDefault handler:^void (UIAlertAction *action)
{
    // do stuff
}];

[self presentViewController:sheet...];
```

UIAlertController

New API


```
UIAlertController* sheet = [UIAlertController alertControllerWithTitle:...];
[sheet addAction:[UIAlertAction actionWithTitle:@“Reply”
style:UIAlertActionStyleDefault handler:^void (UIAlertAction *action)
{
    // do stuff
}];

[self presentViewController:sheet...];
```

UIAlertController

New API

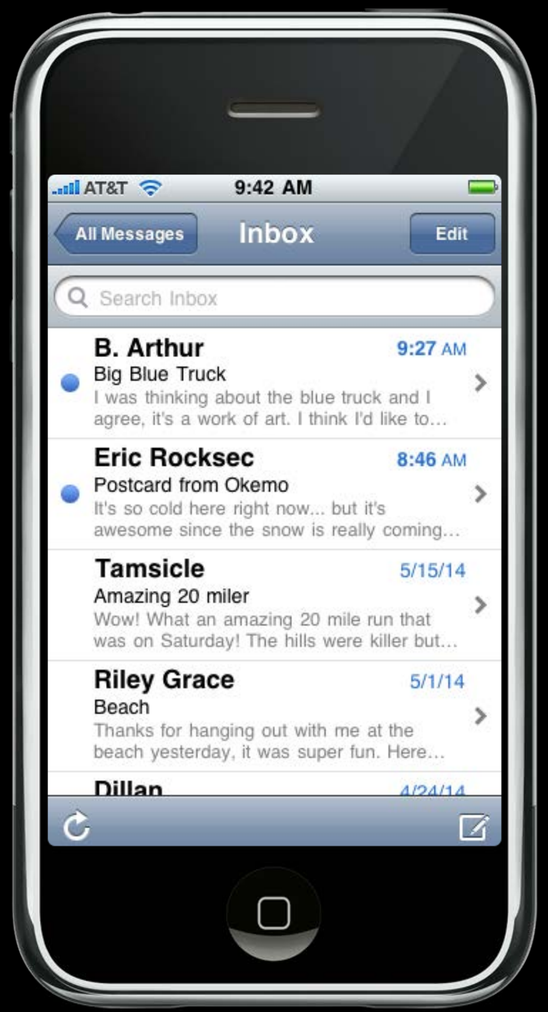

```
UIAlertController* sheet = [UIAlertController alertControllerWithTitle:...];
[sheet addAction:[UIAlertAction actionWithTitle:@“Reply”
style:UIAlertActionStyleDefault handler:^void (UIAlertAction *action)
{
    // do stuff
}];

[self presentViewController:sheet...];
```

Search

UISearchDisplayController

Search

UISearchDisplayController



Search

UISearchDisplayController

Search

UISearchDisplayController

Limited configurability

Search

UISearchDisplayController

Limited configurability

Ambiguous intent

Search

UISearchDisplayController

Limited configurability

Ambiguous intent

“Presented” via addSubview:

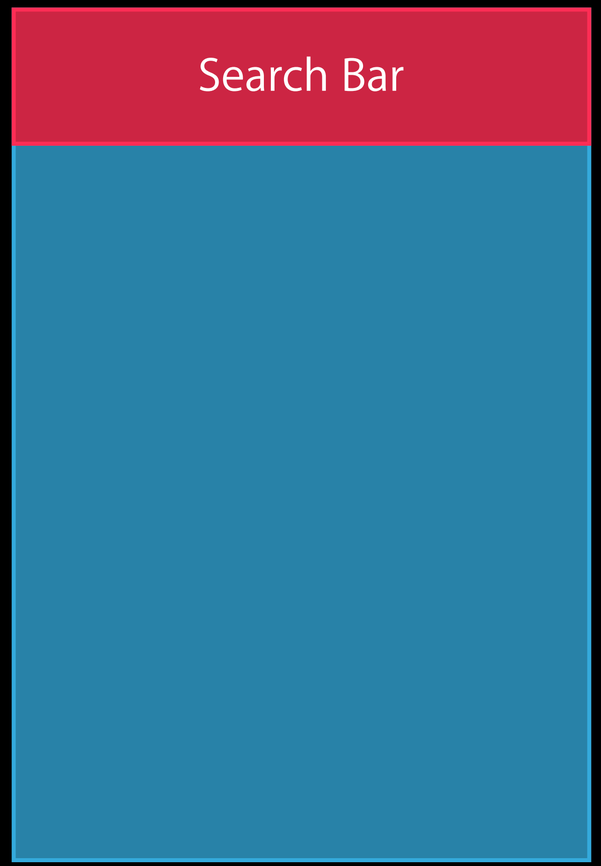
Search

UISearchDisplayController

Limited configurability

Ambiguous intent

“Presented” via addSubview:



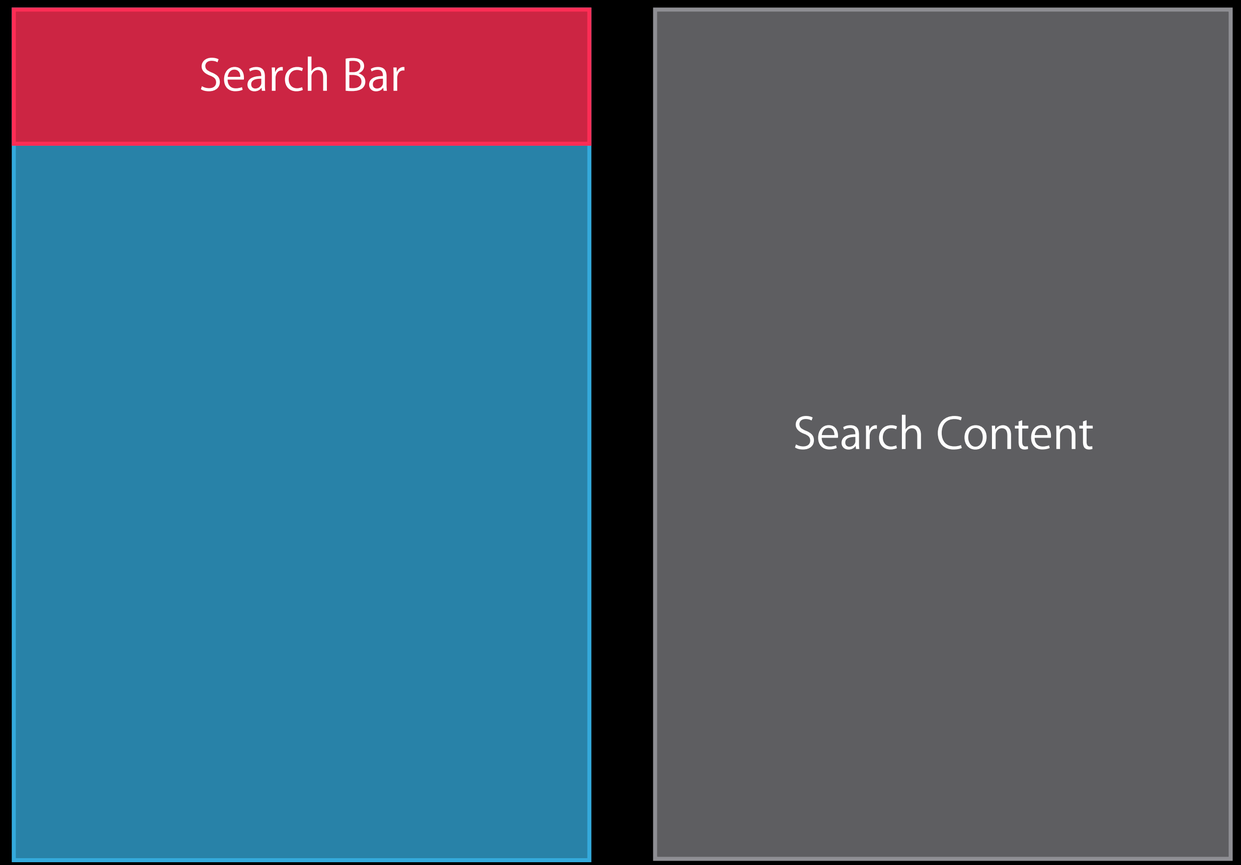
Search

UISearchDisplayController

Limited configurability

Ambiguous intent

“Presented” via addSubview:



Search

UISearchDisplayController

Limited configurability

Ambiguous intent

“Presented” via addSubview:

UISearchController

UISearchController

Replaces UISearchDisplayController

UISearchController

Replaces UISearchDisplayController

Any presenting controller

UISearchController

Replaces UISearchDisplayController

Any presenting controller

Decoupled results

UISearchController

Replaces UISearchDisplayController

Any presenting controller

Decoupled results

Search bar animation

UISearchController

Replaces UISearchDisplayController

Any presenting controller

Decoupled results

Search bar animation

Adaptivity

UISearchController

Replaces UISearchDisplayController

Any presenting controller

Decoupled results

Search bar animation

Adaptivity

More control, fewer assumptions

UISearchDisplayController

Old API

```
UISearchBar *searchBar = [[UISearchBar alloc] initWithFrame:CGRectZero];
UISearchDisplayController *searchController = [[UISearchDisplayController
alloc] initWithSearchBar:searchBar contentsController:self];
searchController.searchResultsDataSource = self;
searchController.searchResultsDelegate = self;
self.tableView.tableHeaderView = searchController.searchBar;
```

UISearchController

New API


```
MyResultsController *resultsController = [[MyResultsController alloc] init];

UISearchController *searchController = [[UISearchController alloc]
initWithSearchResultsController:resultsController];
searchController.searchResultsUpdater = resultsController;
self.tableView.tableHeaderView = searchController.searchBar;
self.definesPresentationContext = YES;
```

UISearchController

New API


```
MyResultsController *resultsController = [[MyResultsController alloc] init];

UISearchController *searchController = [[UISearchController alloc]
initWithSearchResultsController:resultsController];
searchController.searchResultsUpdater = resultsController;
self.tableView.tableHeaderView = searchController.searchBar;
self.definesPresentationContext = YES;
```

UISearchController

New API


```
MyResultsController *resultsController = [[MyResultsController alloc] init];

UISearchController *searchController = [[UISearchController alloc]
initWithSearchResultsController:resultsController];
searchController.searchResultsUpdater = resultsController;
self.tableView.tableHeaderView = searchController.searchBar;
self.definesPresentationContext = YES;
```

UISearchController

New API


```
MyResultsController *resultsController = [[MyResultsController alloc] init];

UISearchController *searchController = [[UISearchController alloc]
initWithSearchResultsController:resultsController];
searchController.searchResultsUpdater = resultsController;
self.tableView.tableHeaderView = searchController.searchBar;
self.definesPresentationContext = YES;
```

UISearchController

New API


```
MyResultsController *resultsController = [[MyResultsController alloc] init];

UISearchController *searchController = [[UISearchController alloc]
initWithSearchResultsController:resultsController];
searchController.searchResultsUpdater = resultsController;
self.tableView.tableHeaderView = searchController.searchBar;
self.definesPresentationContext = YES;
```

UISearchController

New API


```
MyResultsController *resultsController = [[MyResultsController alloc] init];

UISearchController *searchController = [[UISearchController alloc]
initWithSearchResultsController:resultsController];
searchController.searchResultsUpdater = resultsController;
self.tableView.tableHeaderView = searchController.searchBar;
self.definesPresentationContext = YES;
```

UISearchController

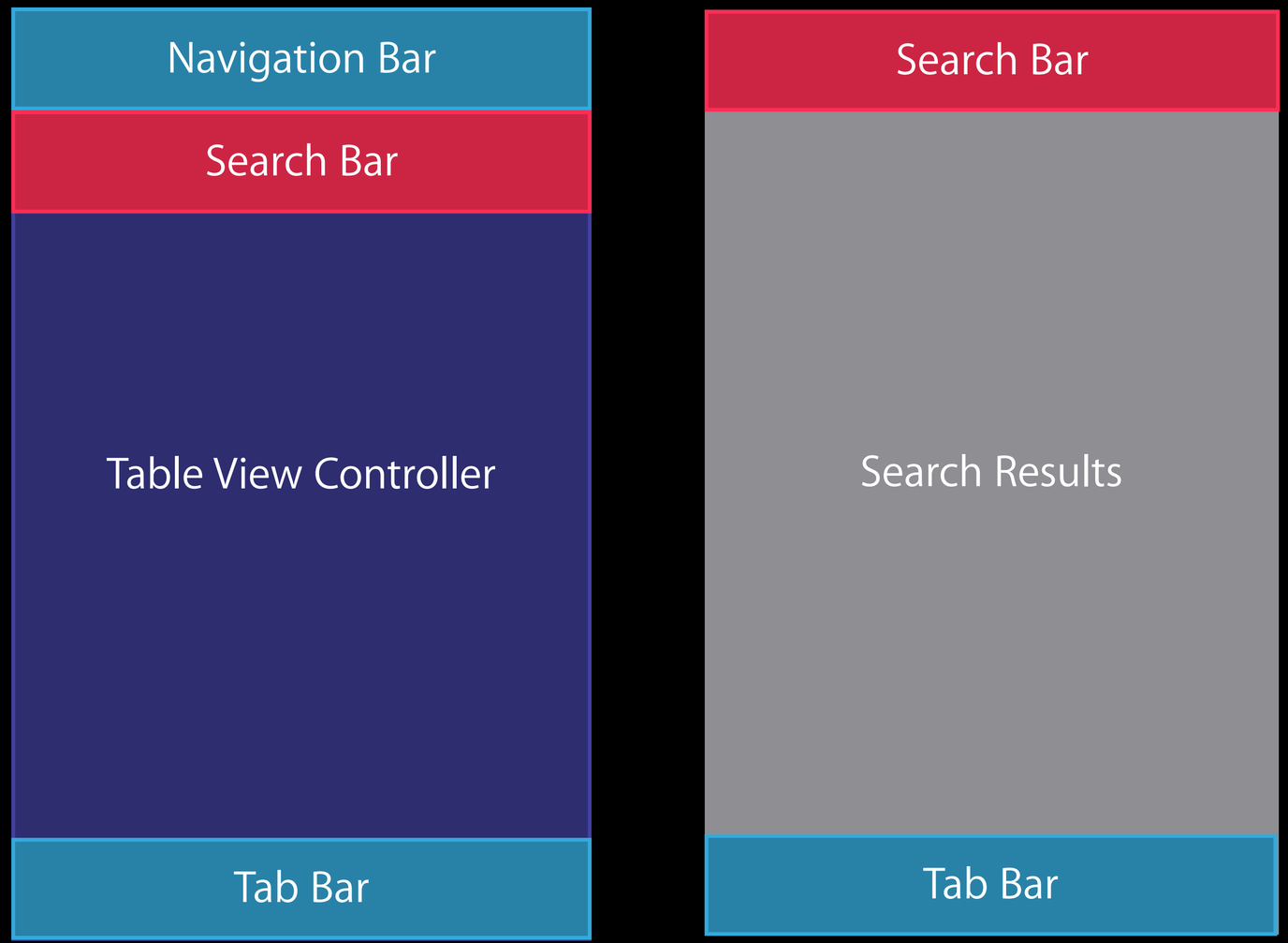
New API

NEW

UISearchController

New API

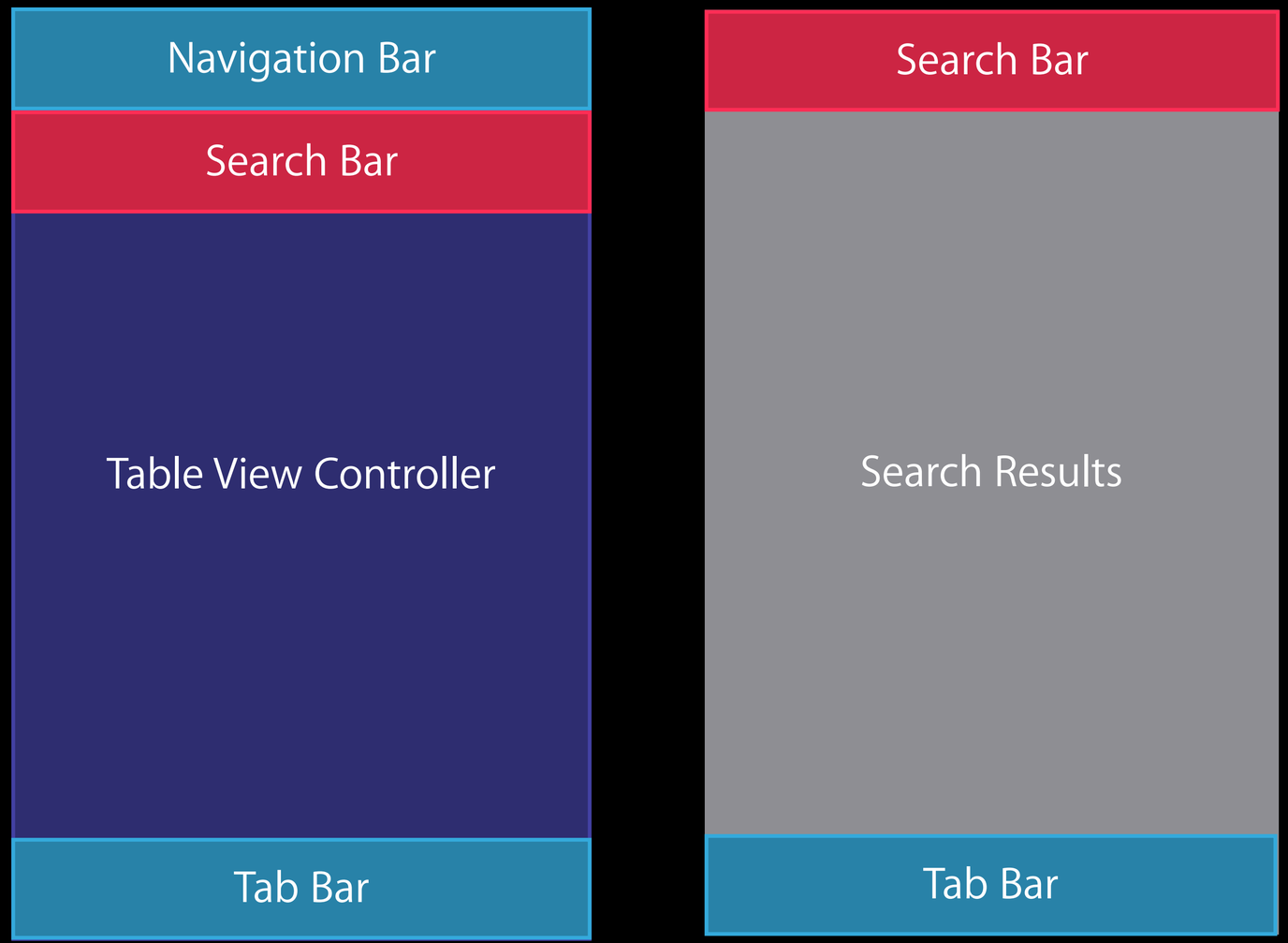
NEW



UISearchController

New API

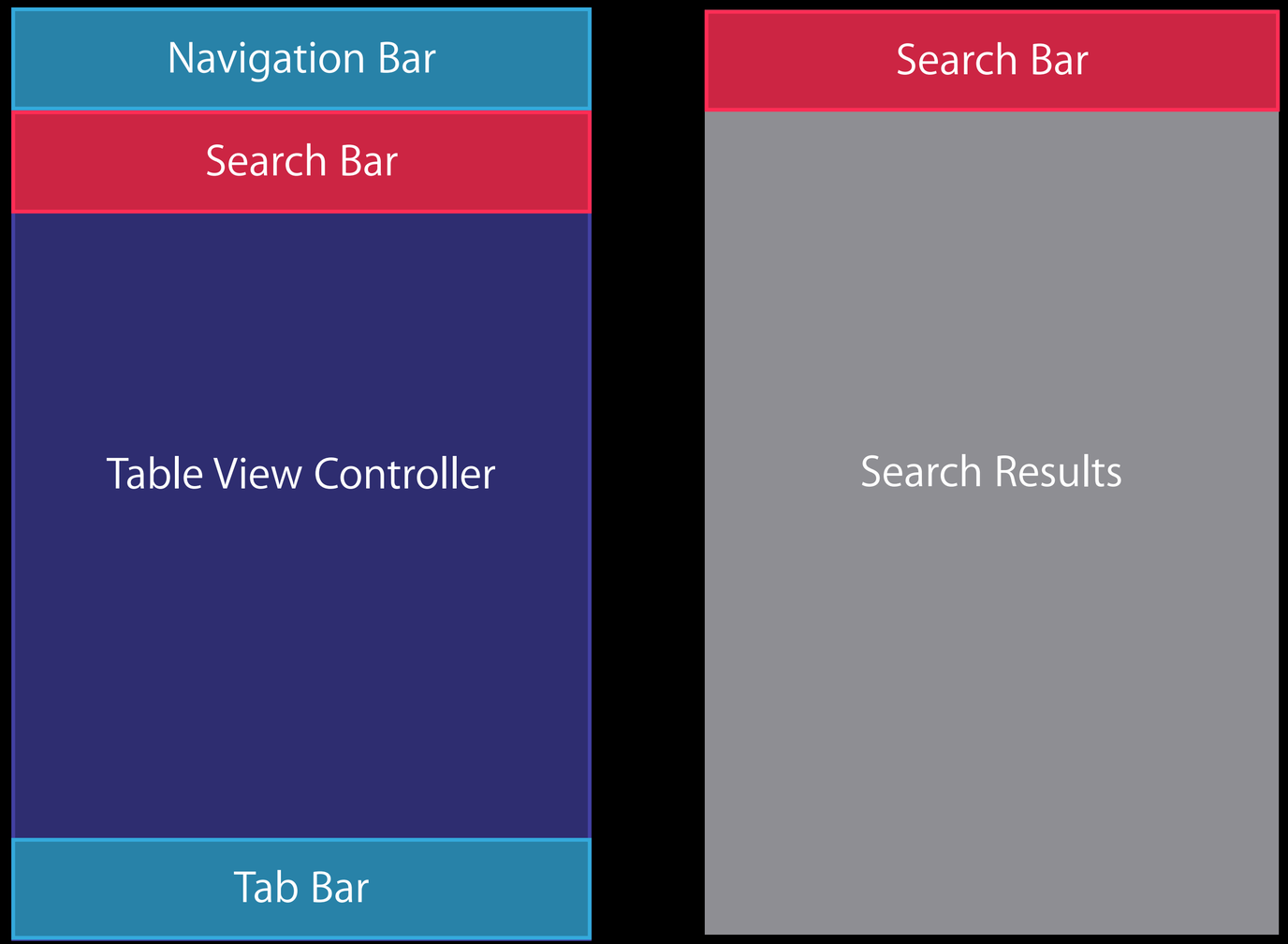
NEW



UISearchController

New API

NEW



UIPresentationController

Benefits

UIPresentationController

Benefits

Controllers all the way down

UIPresentationController

Benefits

Controllers all the way down

Content and chrome abstraction

UIPresentationController

Benefits

Controllers all the way down

Content and chrome abstraction

Adaptivity

Demo

Let’s build something new

Presenting the Overlay View Controller

P

Main view controller

```
OverlayViewController* overlay = [[OverlayViewController alloc] init];
[overlay setPhoto:photo];

[self presentViewController:overlay animated:YES completion:NULL];
```

Presenting the Overlay View Controller

P

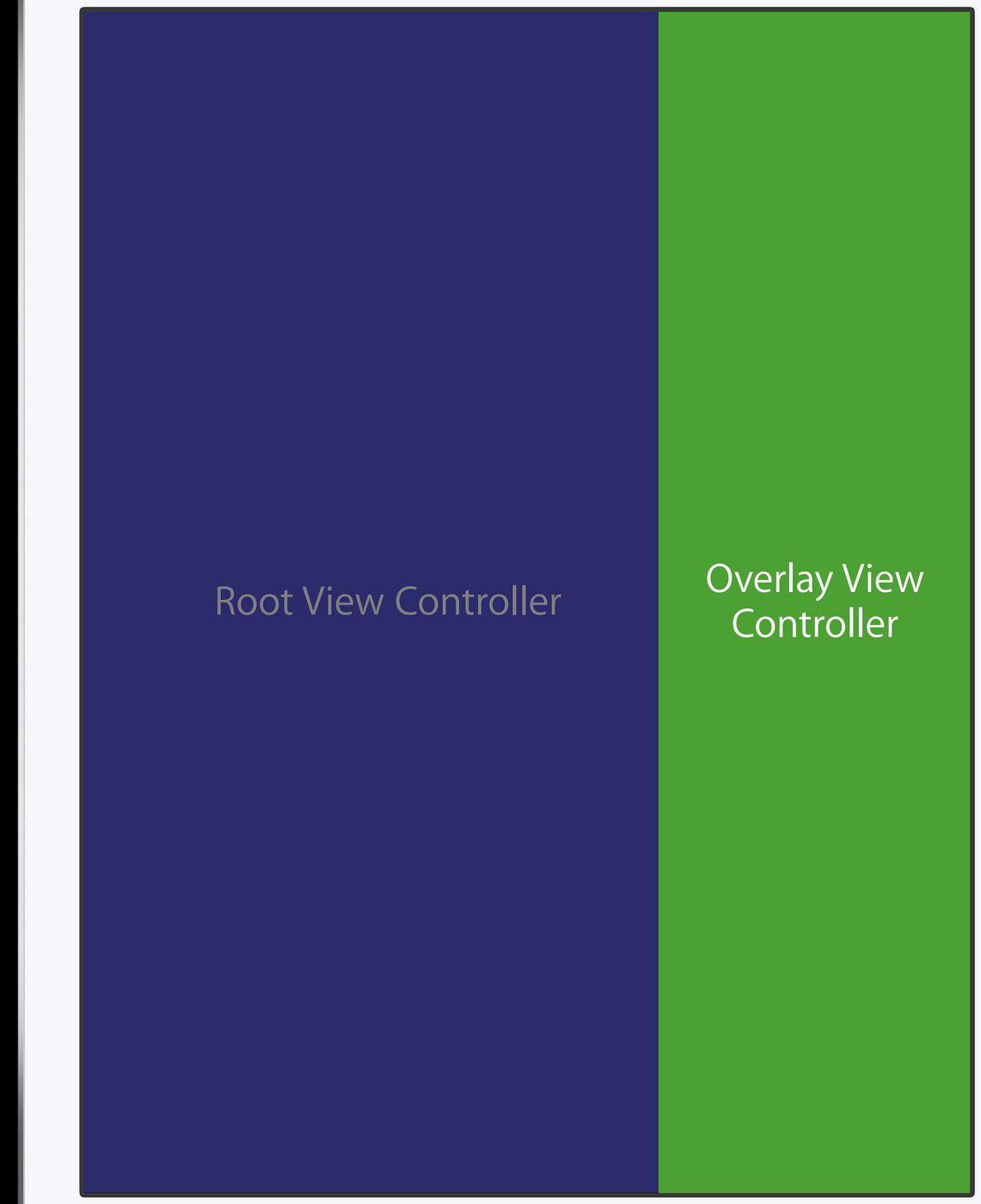
Main view controller

```
OverlayViewController* overlay = [[OverlayViewController alloc] init];
[overlay setPhoto:photo];

[self presentViewController:overlay animated:YES completion:NULL];
```

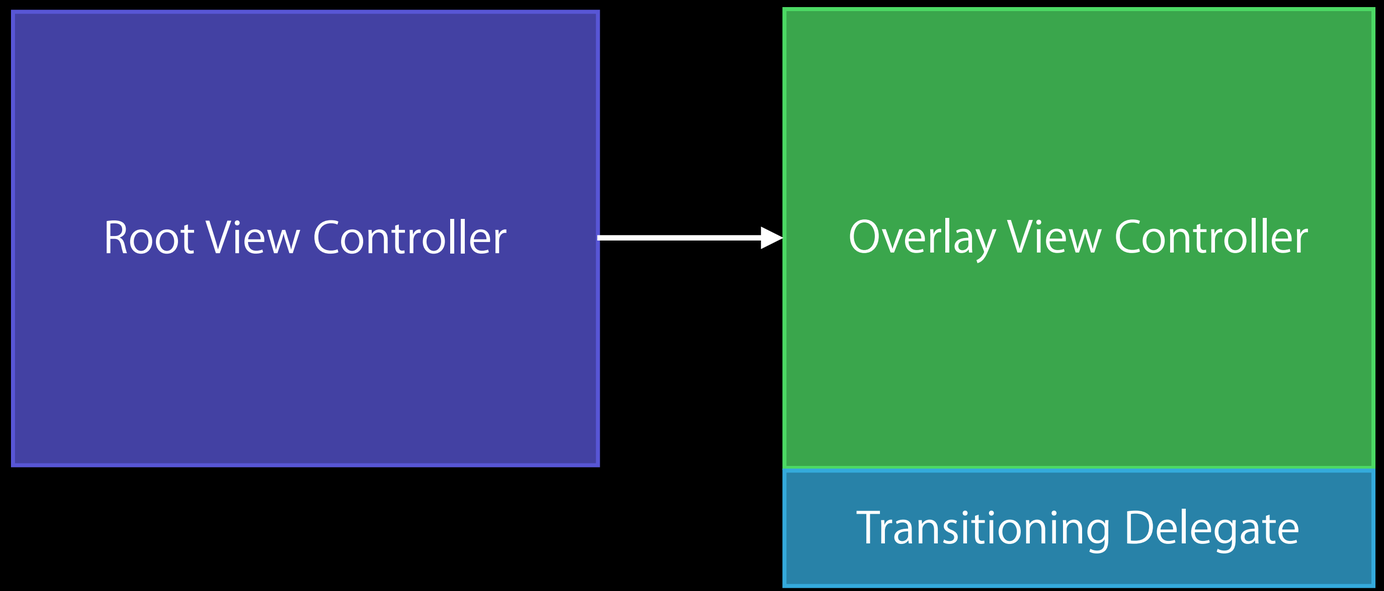
Root View Controller

Overlay View Controller

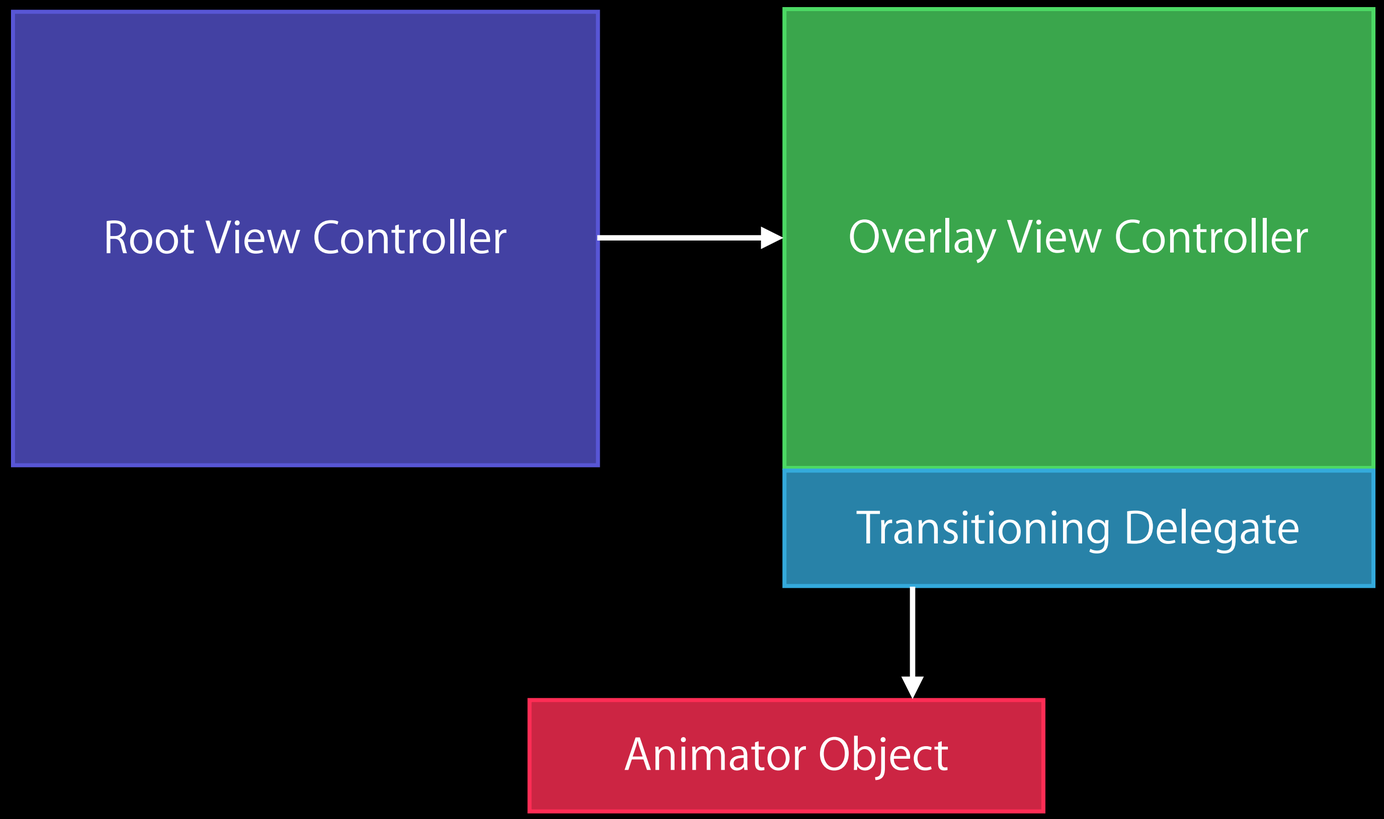
Root View Controller


Root View Controller
Overlay View Controller

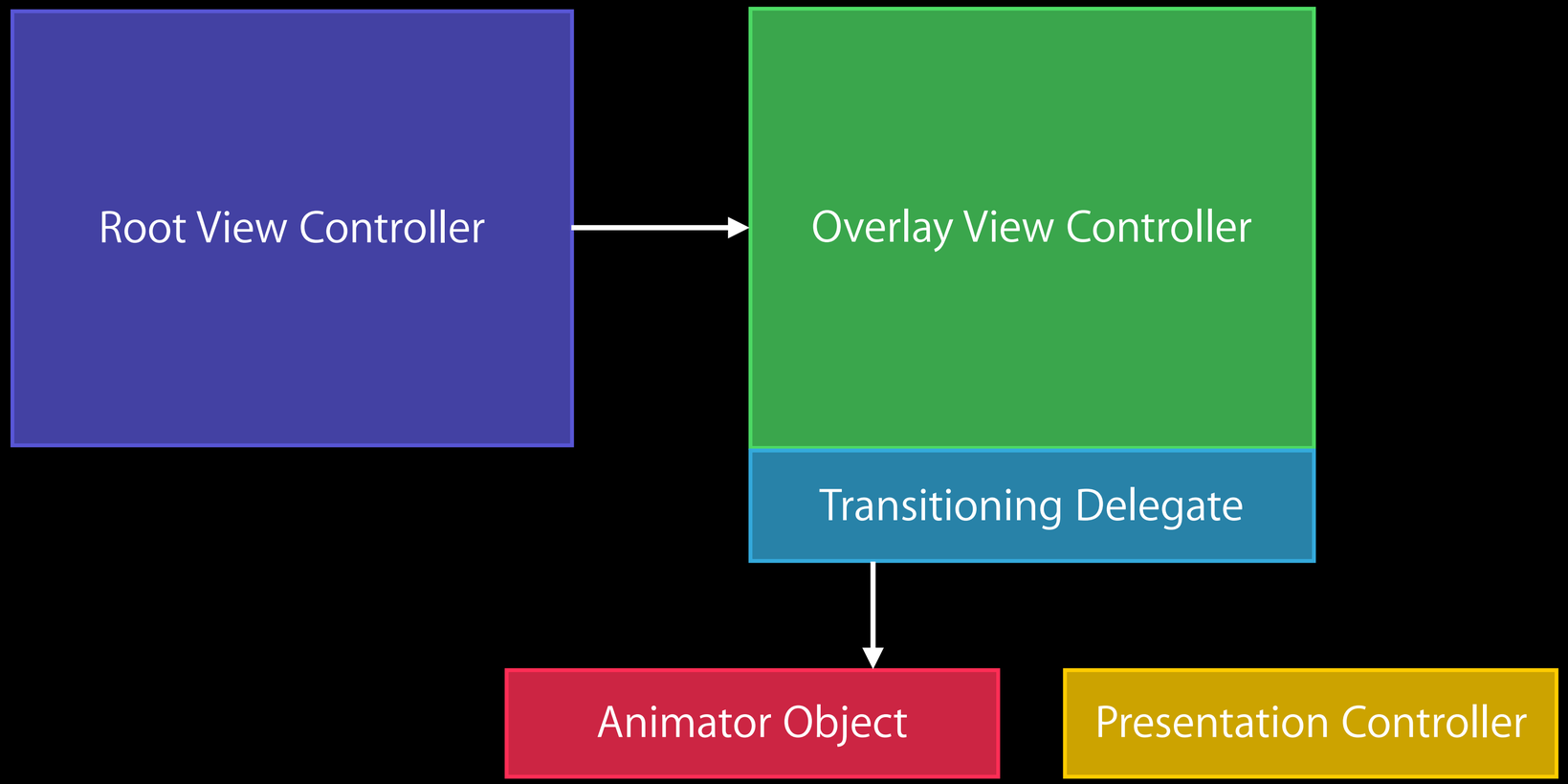
Customizing the Overlay Presentation



Customizing the Overlay Presentation



Customizing the Overlay Presentation



Customizing the Overlay Presentation

P

Root view controller

```
transitionDelegate = [[OverlayTransitioningDelegate alloc] init];
[overlayViewController setTransitioningDelegate:transitionDelegate];
```

Customizing the Overlay Presentation

P

Root view controller

```
transitionDelegate = [[OverlayTransitioningDelegate alloc] init];
[overlayViewController setTransitioningDelegate:transitionDelegate];
```

Customizing the Overlay Presentation

Overlay view controller

```
[self setModalPresentationStyle:UIModalPresentationCustom];
```

Customizing the Overlay Presentation

TD

Providing a custom UIPresentationController

```
- (UIPresentationController
*)presentationControllerForPresentedViewController:(UIViewController
*)presented presentingViewController:(UIViewController *)presenting
sourceViewController:(UIViewController *)source
{
    return [[OverlayPresentationController alloc]
          initWithPresentingViewController:presenting
                presentedViewController:presented];
}
```

Customizing the Overlay Presentation

PC

Presentation controller animations

```
- (void)presentationTransitionWillBegin
{
    UIView* containerView = [self containerView];
    UIViewController* presentedViewController = [self
presentedViewController];
    [dimmingView setFrame:[containerView bounds]];
    [dimmingView setAlpha:0.0];

    [containerView insertSubview:dimmingView atIndex:0];

    [[presentedViewController transitionCoordinator]
animateAlongsideTransition:^(id<UIViewControllerTransitionCoordinatorContext>
context) {
        [dimmingView setAlpha:1.0];
    } completion:nil];
}
```

Customizing the Overlay Presentation

PC

Presentation controller animations

```
- (void)presentationTransitionWillBegin
{
    UIView* containerView = [self containerView];
    UIViewController* presentedViewController = [self
presentedViewController];
    [dimmingView setFrame:[containerView bounds]];
    [dimmingView setAlpha:0.0];

    [containerView insertSubview:dimmingView atIndex:0];

    [[presentedViewController transitionCoordinator]
animateAlongsideTransition:^(id<UIViewControllerTransitionCoordinatorContext>
context) {
        [dimmingView setAlpha:1.0];
    } completion:nil];
}
```

Customizing the Overlay Presentation

PC

Presentation controller animations

```
- (void)presentationTransitionWillBegin
{
    UIView* containerView = [self containerView];
    UIViewController* presentedViewController = [self
presentedViewController];
    [dimmingView setFrame:[containerView bounds]];
    [dimmingView setAlpha:0.0];

    [containerView insertSubview:dimmingView atIndex:0];

    [[presentedViewController transitionCoordinator]
animateAlongsideTransition:^(id<UIViewControllerTransitionCoordinatorContext>
context) {
        [dimmingView setAlpha:1.0];
    } completion:nil];
}
```

Customizing the Overlay Presentation

PC

Presentation controller animations

```
- (void)dismissalTransitionWillBegin
{
    [[[self presentedViewController] transitionCoordinator]
animateAlongsideTransition:^(id<UIViewControllerTransitionCoordinatorContext>
context) {
        [dimmingView setAlpha:0.0];
    } completion:nil];
}
```

Root View Controller

Root View Controller

Customizing the Overlay Presentation

TD

Transitioning delegate animations

```
- (id <UIViewControllerAnimatedTransitioning>)
animationControllerForPresentedController:(UIViewController *)presented
presentingController:(UIViewController *)presenting
sourceController:(UIViewController *)source
{
    OverlayAnimatedTransitioning *animationController =
[[OverlayAnimatedTransitioning alloc] init];
    return animationController;
}
```

Customizing the Overlay Presentation

TD

Transitioning delegate animations

```
- (id <UIViewControllerAnimatedTransitioning>)
animationControllerForDismissedController:(UIViewController *)dismissed
{
    OverlayAnimatedTransitioning *animationController =
[[OverlayAnimatedTransitioning alloc] init];
    return animationController;
}
```

Customizing the Overlay Presentation

PC

Presented view controller sizing

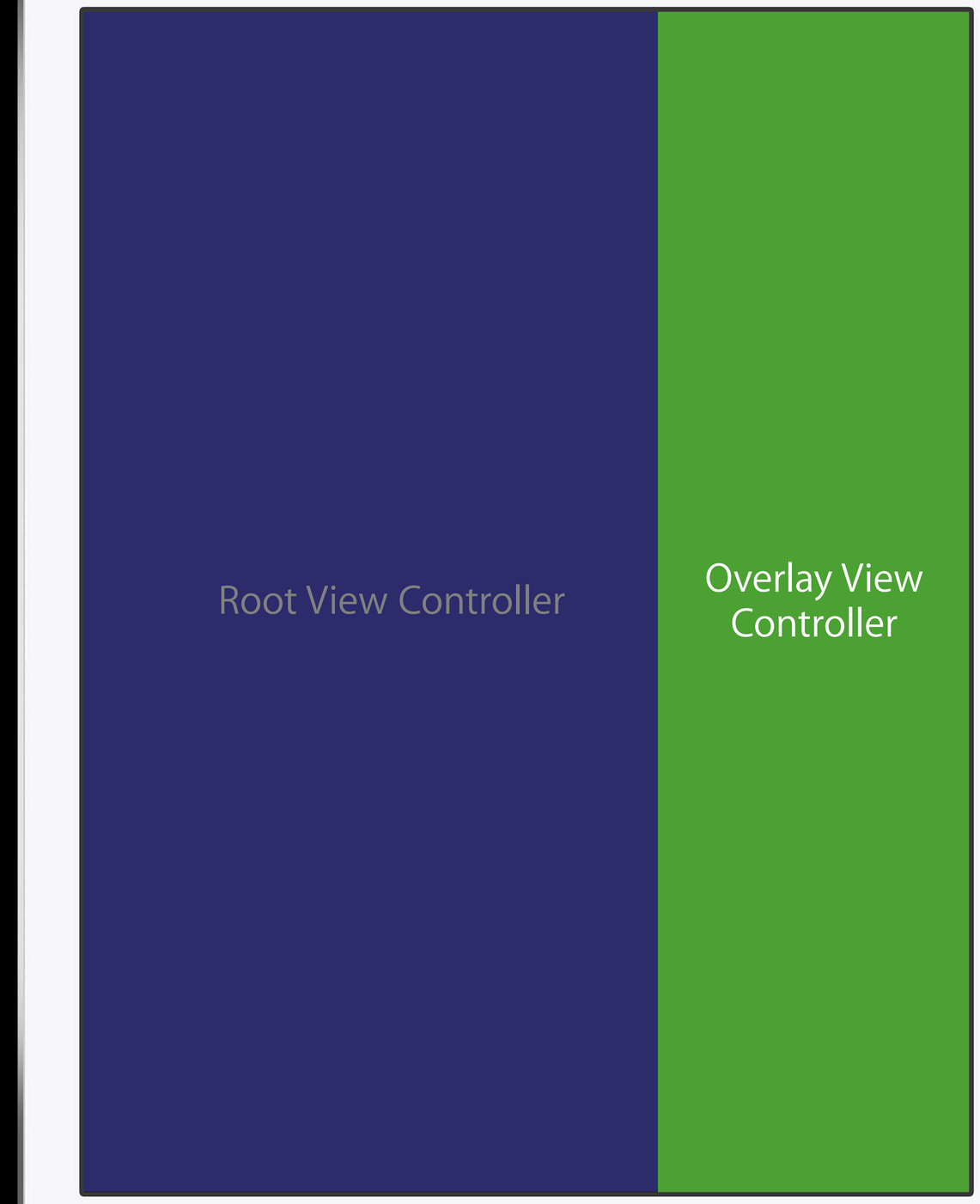
```
- (CGSize)sizeForChildContentContainer:(id <UIContentContainer>)container
withParentContainerSize:(CGSize)parentSize
{
    return CGSizeMake(floorf(parentSize.width / 3.0),
                      parentSize.height);
}
```

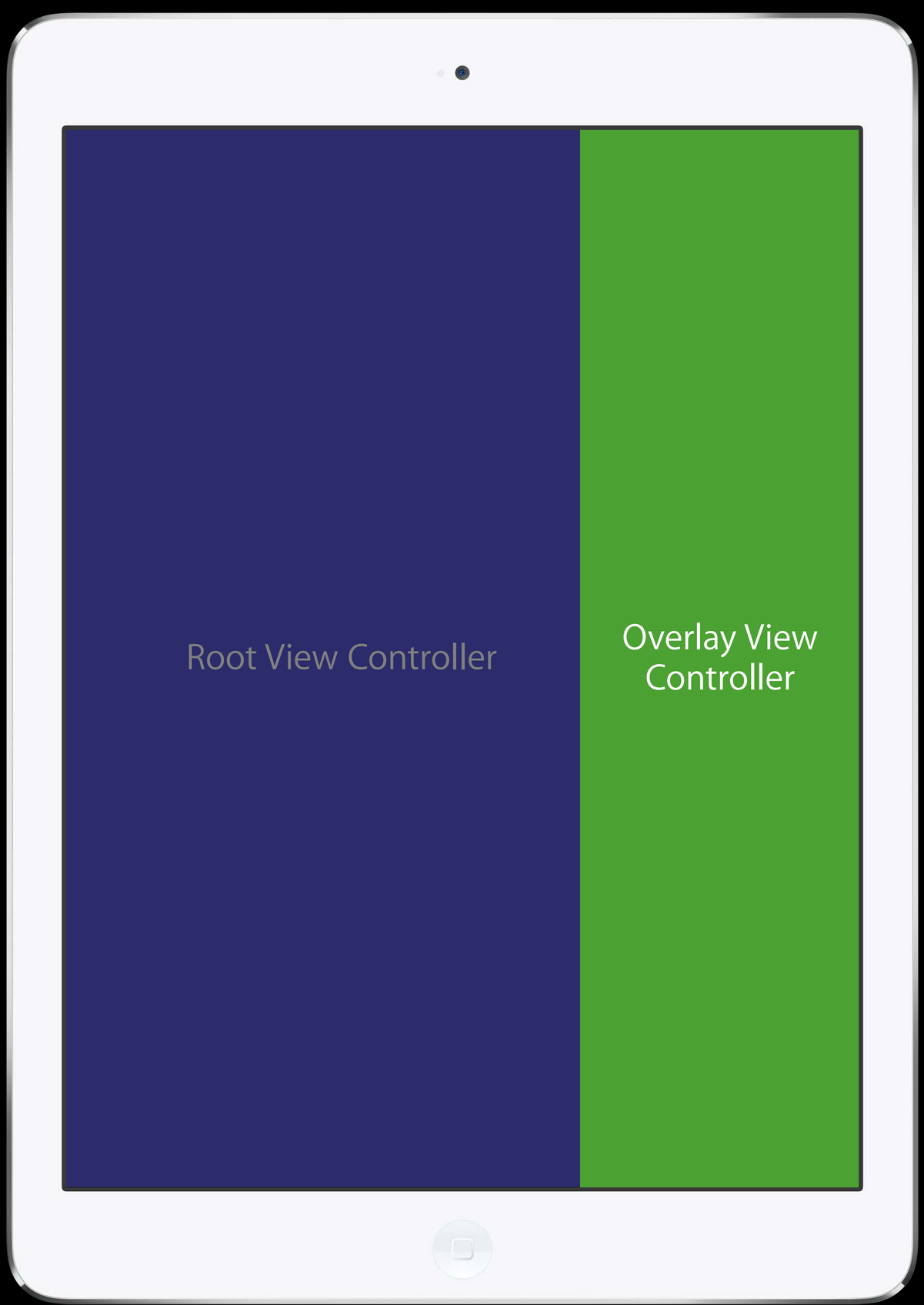
Customizing the Overlay Presentation

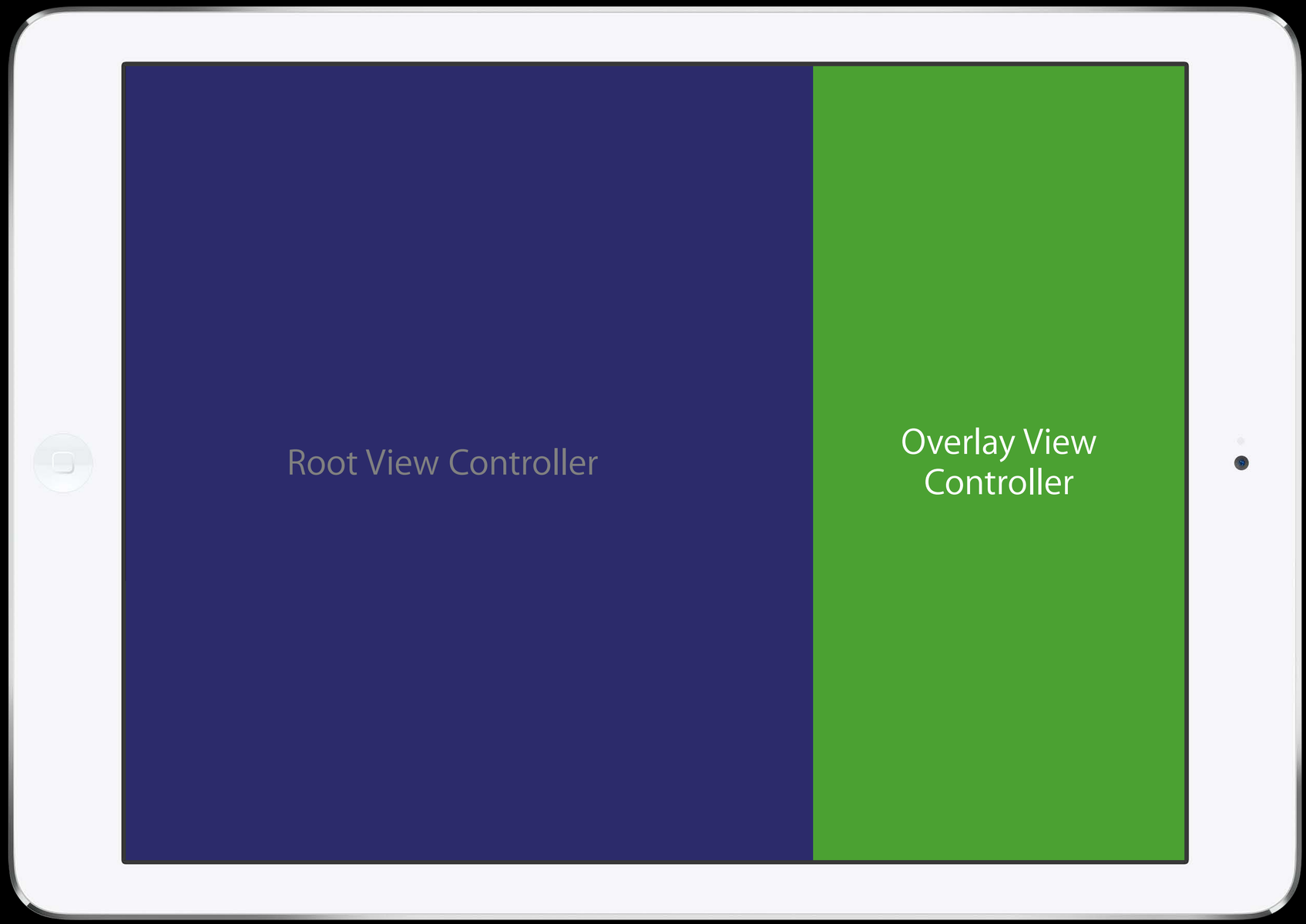
PC

Presented view controller sizing

```
- (CGRect)frameOfPresentedViewInContainerView
{
    CGRect presentedViewFrame = CGRectZero;
    CGRect containerBounds = [[self containerView] bounds];

    presentedViewFrame.size = [self sizeForChildContentContainer:
(UIView<UIContentContainer> *)[self presentedView]
    withParentContainerSize:containerBounds.size];

    presentedViewFrame.origin.x = containerBounds.size.width -
presentedViewFrame.size.width;

    return presentedViewFrame;
}
```


Root View Controller
Overlay View Controller


Root View Controller
Overlay View Controller


Root View Controller
Overlay View Controller

Customizing the Overlay Presentation

PC

Adding rotation support

```
- (void)containerViewWillLayoutSubviews
{
    [dimmingView setFrame:[[self containerView] bounds]];
    [[self presentedView] setFrame:[self frameOfPresentedViewInContainerView]];
}
```

Customizing the Overlay Presentation

PC

Adding rotation support

```
- (void)containerViewWillLayoutSubviews
{
    [dimmingView setFrame:[[self containerView] bounds]];
    [[self presentedView] setFrame:[self frameOfPresentedViewInContainerView]];
}
```

UIContentContainer

UIContentContainer

UIViewController

UIContentContainer

UIViewController

UIPresentationController

UIContentContainer

UIViewController

UIPresentationController

Environment changes

UIContentContainer

Transitioning to size

UIContentContainer

Transitioning to size

```
- (void)viewWillTransitionToSize:(CGSize)size withTransitionCoordinator:
(id<UIViewControllerTransitionCoordinator>)coordinator
```

UIContentContainer

Transitioning to size

```
- (void)viewWillTransitionToSize:(CGSize)size withTransitionCoordinator:
(id<UIViewControllerTransitionCoordinator>)coordinator

- (CGSize)sizeForChildContentContainer:(id <UIContentContainer>)container
withParentContainerSize:(CGSize)parentSize
```

UIContentContainer

Transitioning to size

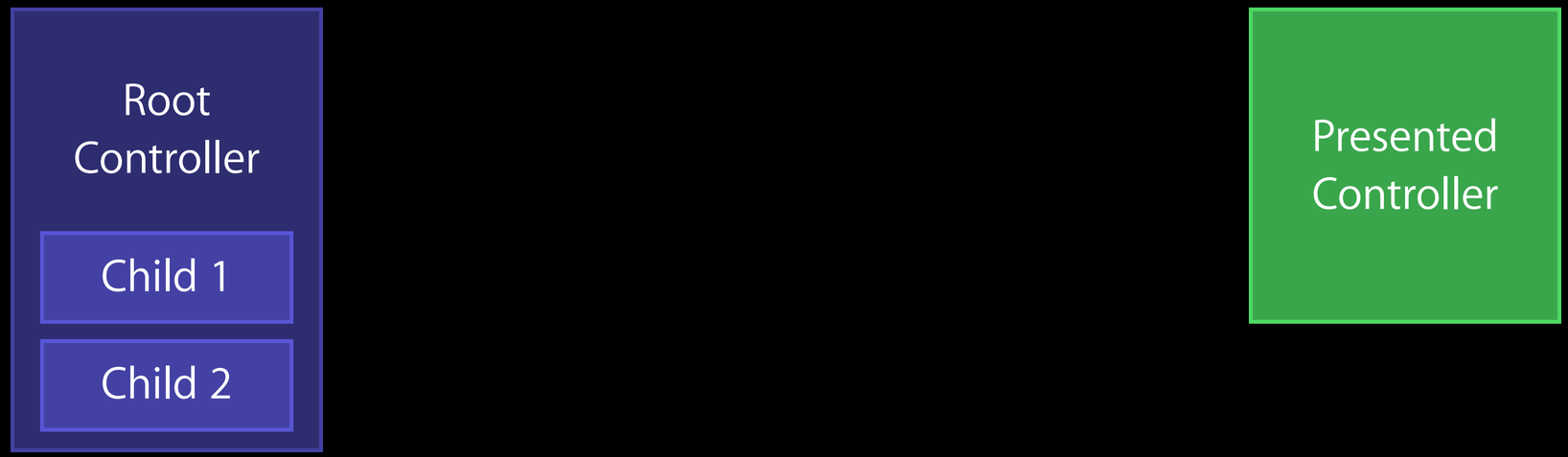
UIContentContainer

Transitioning to size

UIContentContainer

Transitioning to size

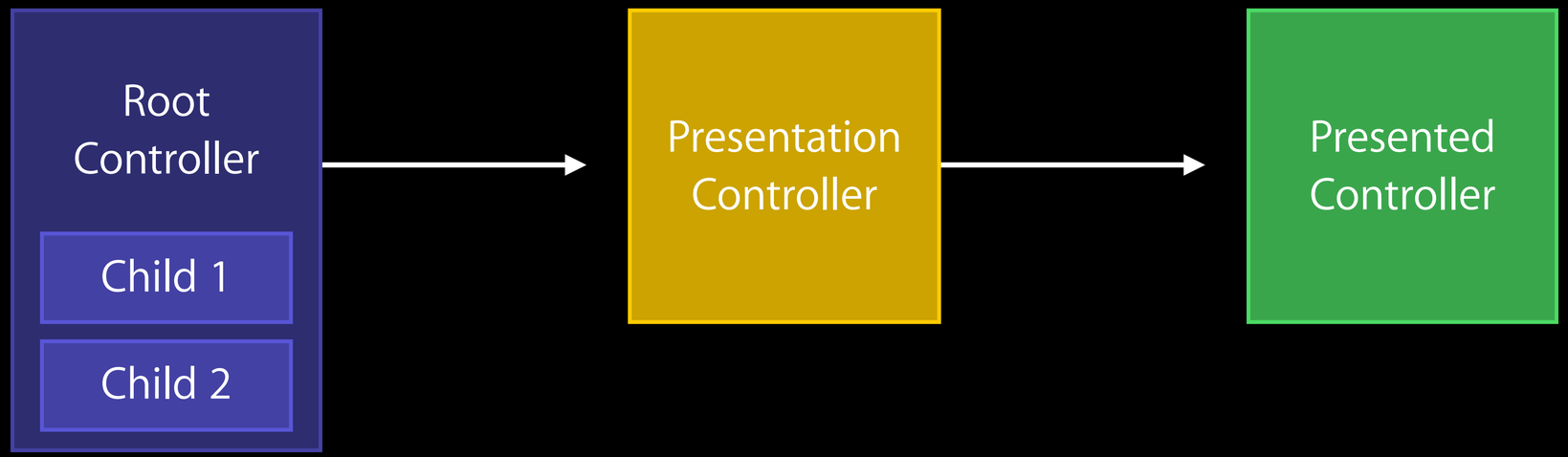
```
[root presentViewController:presented …];
```



UIContentContainer

Transitioning to size

```
[root presentViewController:presented …];
```



UIContentContainer

Transitioning to size

UIContentContainer

Transitioning to size

```
viewWillTransitionToSize:withTransitionCoordinator:
```

UIContentContainer

Transitioning to size

```
viewWillTransitionToSize:withTransitionCoordinator:
```

UIContentContainer

Transitioning to size

```
viewWillTransitionToSize:withTransitionCoordinator:
```

UIContentContainer

Transitioning to size

```
viewWillTransitionToSize:withTransitionCoordinator:
```


```
[Root sizeForChildContentContainer:Child1 withParentContainerSize:size]
```

UIContentContainer

Transitioning to size

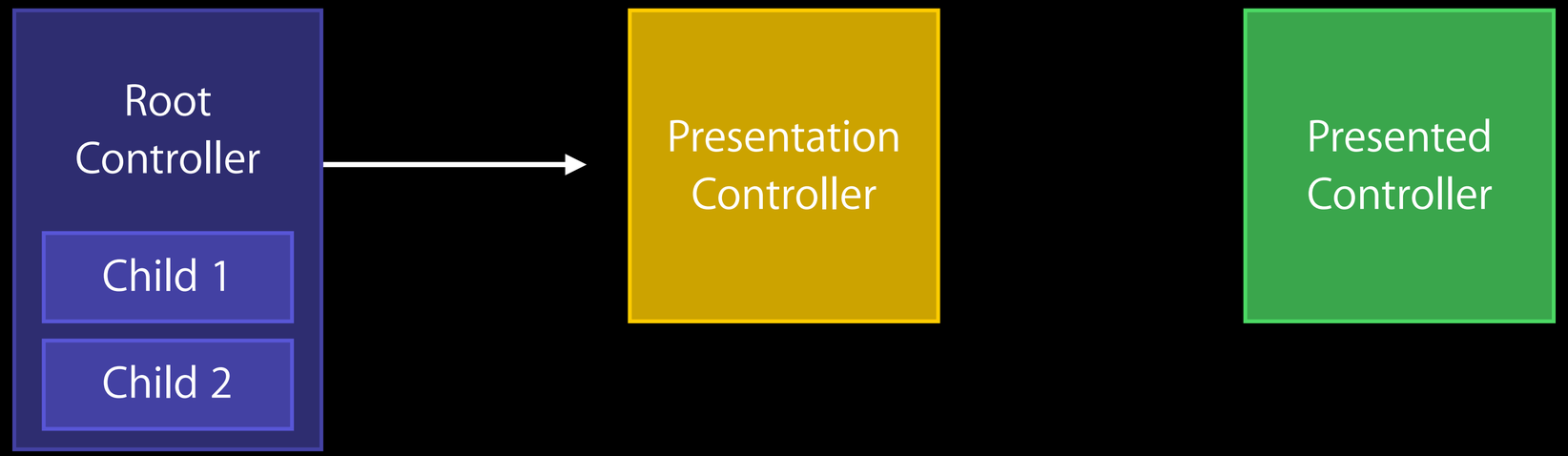
```
viewWillTransitionToSize:withTransitionCoordinator:
```


```
[Root sizeForChildContentContainer:Child2 withParentContainerSize:size]
```

UIContentContainer

Transitioning to size

```
viewWillTransitionToSize:withTransitionCoordinator:
```



UIContentContainer

Transitioning to size

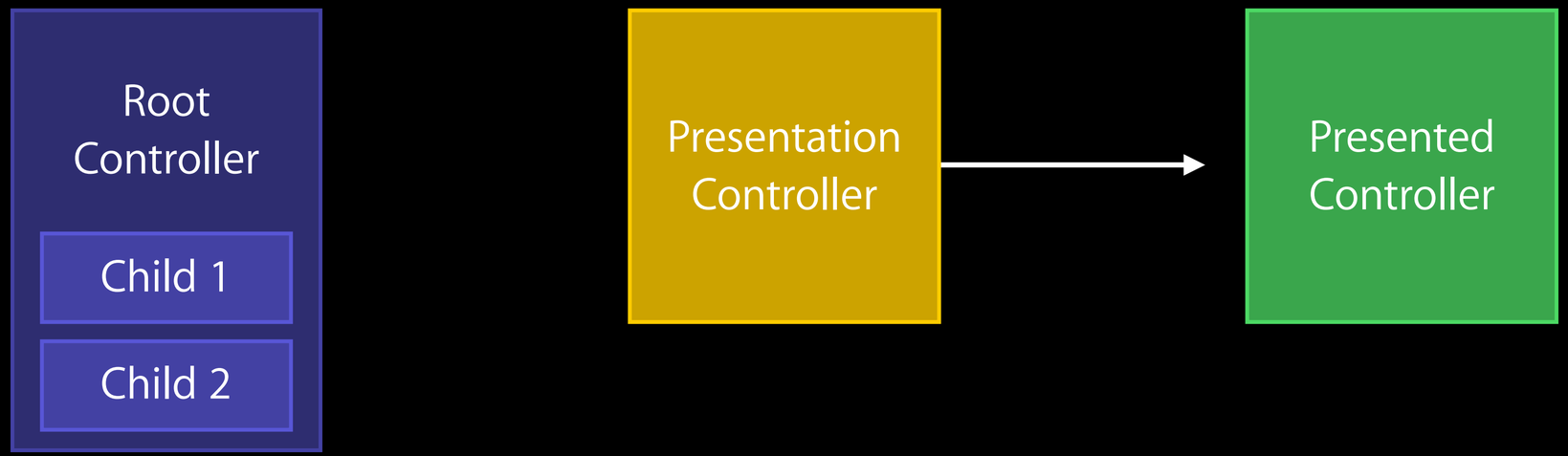
```
viewWillTransitionToSize:withTransitionCoordinator:
```


```
[Presentation sizeForChildContentContainer:Presented withParentContainerSize:size]
```

UIContentContainer

Transitioning to size

```
viewWillTransitionToSize:withTransitionCoordinator:
```



UIContentContainer

Preferred content size

UIContentContainer

Preferred content size

```
@property (nonatomic, readonly) CGSize preferredContentSize
```

UIContentContainer

Preferred content size

```
@property (nonatomic, readonly) CGSize preferredContentSize

- (void)preferredContentSizeDidChangeForChildContentContainer:(id
<UIContentContainer>)container
```

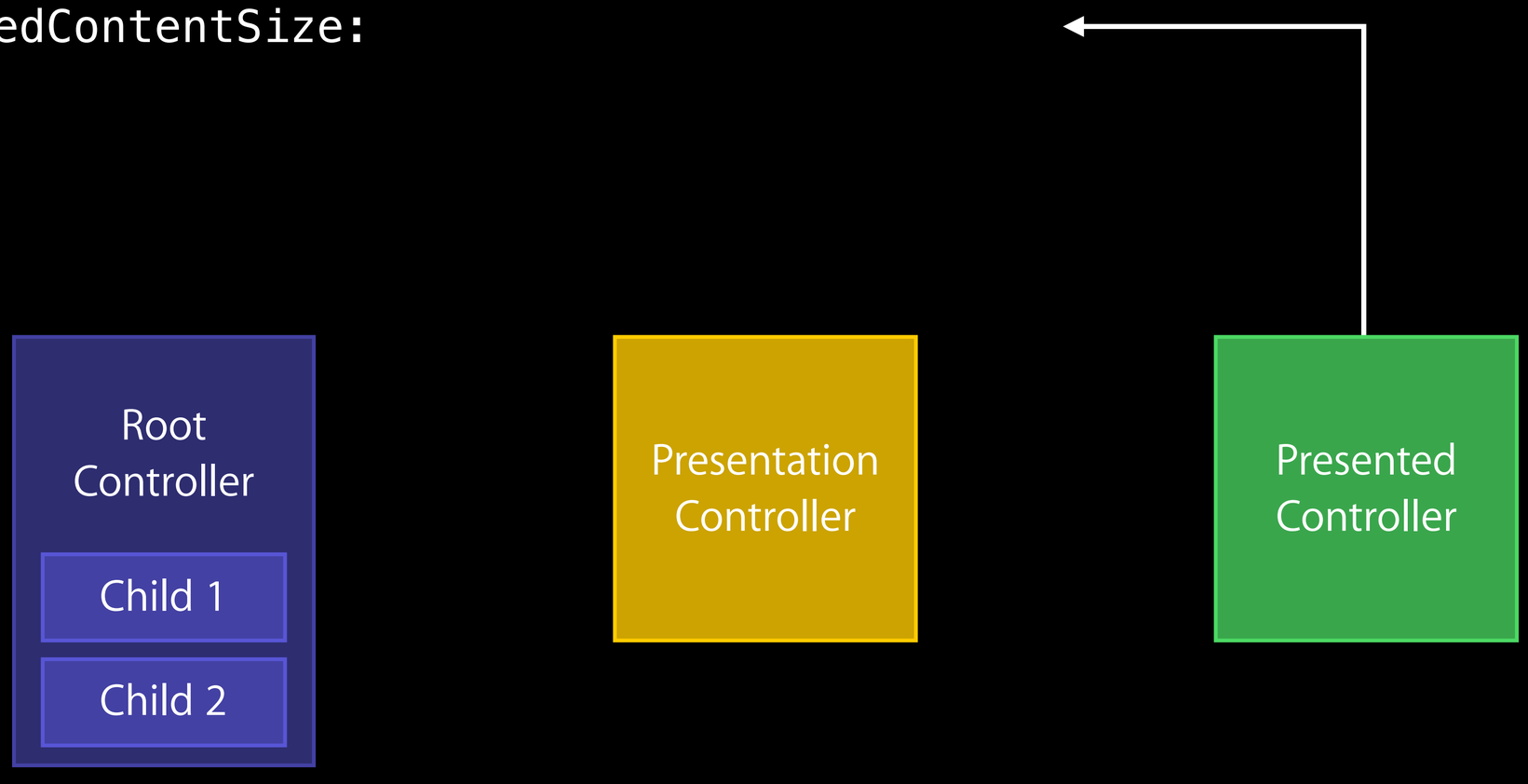
UIContentContainer

Preferred content size

UIContentContainer

Preferred content size

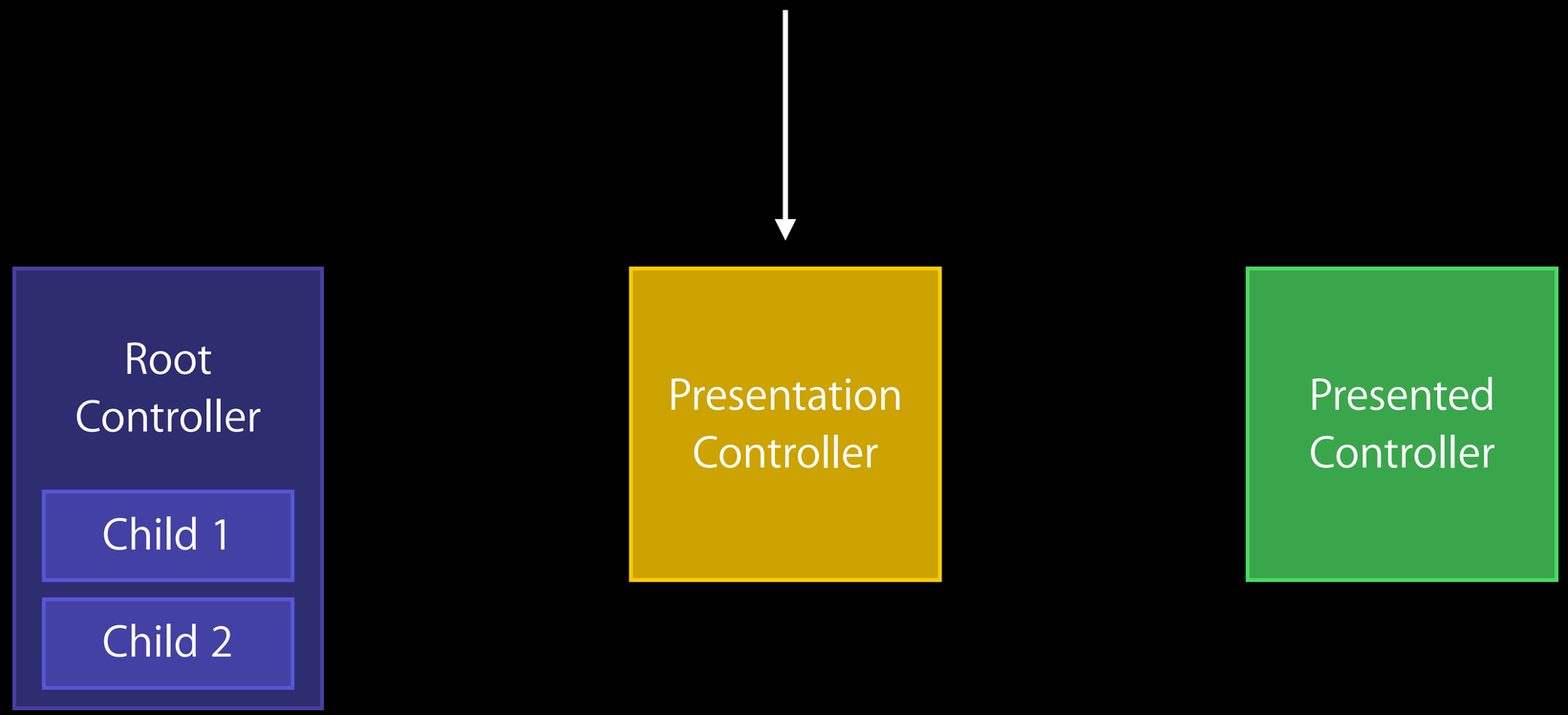
`setPreferredContentSize:`



UIContentContainer

Preferred content size

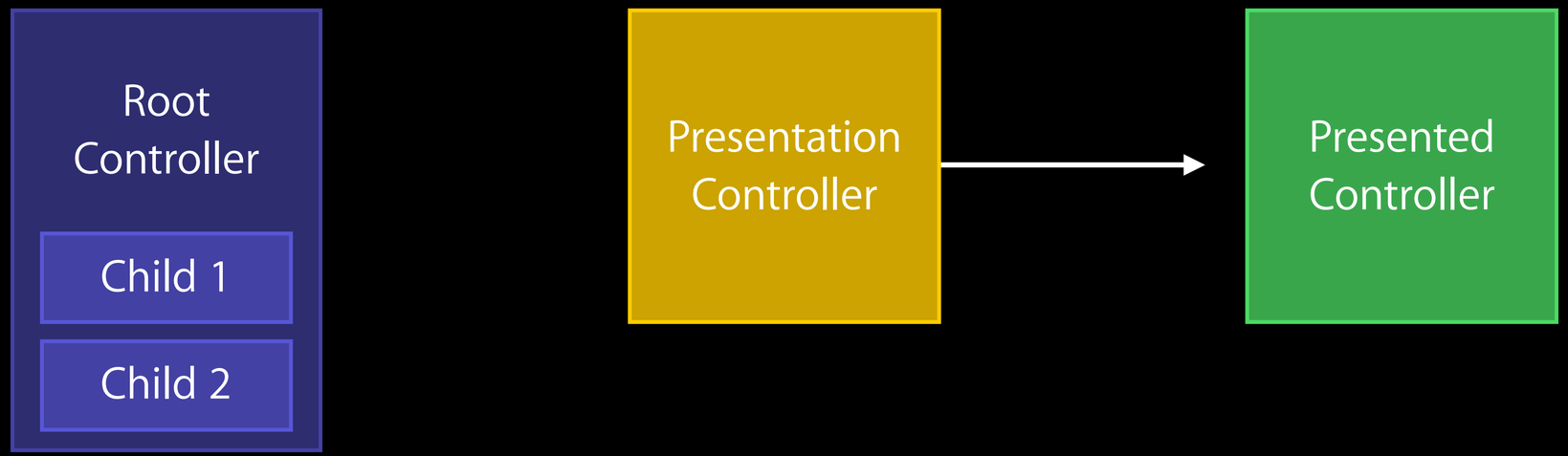
```
preferredContentSizeDidChangeForChildContentContainer:
```



UIContentContainer

Preferred content size

```
viewWillTransitionToSize:withTransitionCoordinator:
```



UIContentContainer

Preferred content size

UIContentContainer

Trait collections

UIContentContainer

Trait collections

```
- (void)willTransitionToTraitCollection:(UITraitCollection *)newCollection
withTransitionCoordinator:
(id<UIViewControllerTransitionCoordinator>)coordinator
```

Trait Collections

```
@property (nonatomic, readonly) UIUserInterfaceSizeClass horizontalSizeClass;
@property (nonatomic, readonly) UIUserInterfaceSizeClass verticalSizeClass;
@property (nonatomic, readonly) UIUserInterfaceIdiom userInterfaceIdiom;
@property (nonatomic, readonly) CGFloat displayScale;
```

horizontalSizeClass

verticalSizeClass

userInterfaceIdiom

displayScale

Trait Collections

```
@property (nonatomic, readonly) UIUserInterfaceSizeClass horizontalSizeClass;
@property (nonatomic, readonly) UIUserInterfaceSizeClass verticalSizeClass;
@property (nonatomic, readonly) UIUserInterfaceIdiom userInterfaceIdiom;
@property (nonatomic, readonly) CGFloat displayScale;
```

horizontalSizeClass	Compact
verticalSizeClass	Regular
userInterfaceIdiom	Phone
displayScale	2.0

Responding to Changes

Responding to Changes

```
- (void)willTransitionToTraitCollection:(UITraitCollection *)newCollection
withTransitionCoordinator:
(id<UIViewControllerTransitionCoordinator>)coordinator
```

Responding to Changes

```
- (void)willTransitionToTraitCollection:(UITraitCollection *)newCollection
withTransitionCoordinator:
(id<UIViewControllerTransitionCoordinator>)coordinator

- (void)viewWillTransitionToSize:(CGSize)size withTransitionCoordinator:
(id<UIViewControllerTransitionCoordinator>)coordinator
```

Responding to Changes

```
- (void)willTransitionToTraitCollection:(UITraitCollection *)newCollection
withTransitionCoordinator:
(id<UIViewControllerTransitionCoordinator>)coordinator

- (void)viewWillTransitionToSize:(CGSize)size withTransitionCoordinator:
(id<UIViewControllerTransitionCoordinator>)coordinator

- (void)containerViewWillLayoutSubviews
- (void)viewWillLayoutSubviews
```

Responding to Changes

Responding to Changes

willTransitionToTraitCollection

Responding to Changes

willTransitionToTraitCollection

- Coarse-grain control

Responding to Changes

willTransitionToTraitCollection

- Coarse-grain control
- Adaptivity

Responding to Changes

willTransitionToTraitCollection

- Coarse-grain control
- Adaptivity
- Screens

Responding to Changes

willTransitionToTraitCollection

- Coarse-grain control
- Adaptivity
- Screens
- Auto-hiding status bar

Responding to Changes

Responding to Changes

viewWillTransitionToSize

Responding to Changes

viewWillTransitionToSize

- Medium-grain control

Responding to Changes

viewWillTransitionToSize

- Medium-grain control
- Child controller size requests

Responding to Changes

viewWillTransitionToSize

- Medium-grain control
- Child controller size requests
- Rotation

Responding to Changes

viewWillTransitionToSize

- Medium-grain control
- Child controller size requests
- Rotation
- Resizable simulator

Responding to Changes

Responding to Changes

containerViewWillLayoutSubviews and viewWillLayoutSubviews

Responding to Changes

containerViewWillLayoutSubviews and viewWillLayoutSubviews

- Fine-grain control

Responding to Changes

containerViewWillLayoutSubviews and viewWillLayoutSubviews

- Fine-grain control
- Resize subviews

Responding to Changes

containerViewWillLayoutSubviews and viewWillLayoutSubviews

- Fine-grain control
- Resize subviews
- Child container views

Demo

Adaptivity and multiple presentation controllers

Adding a New Presentation Style

P

Root view controller

```
transitionDelegate = [[OverlayTransitioningDelegate alloc] init];
[overlayViewController setTransitioningDelegate:transitionDelegate];
```

Adding a New Presentation Style

P

Root view controller

```
if([self presentationShouldBeAwesome])
{
    transitioningDelegate = [[CoolTransitioningDelegate alloc] init];
}
else
{
    transitioningDelegate = [[OverlayTransitioningDelegate alloc] init];
}

[overlay setTransitioningDelegate:transitioningDelegate];
```

Adding a New Presentation Style

Root view controller

P

```
if([self presentationShouldBeAwesome])
{
    transitioningDelegate = [[CoolTransitioningDelegate alloc] init];
}
else
{
    transitioningDelegate = [[OverlayTransitioningDelegate alloc] init];
}

[overlay setTransitioningDelegate:transitioningDelegate];
```

Adding a New Presentation Style

P

Root view controller

```
if([self presentationShouldBeAwesome])
{
    transitioningDelegate = [[CoolTransitioningDelegate alloc] init];
}
else
{
    transitioningDelegate = [[OverlayTransitioningDelegate alloc] init];
}

[overlay setTransitioningDelegate:transitioningDelegate];
```

Summary

Summary

Improved API through UIAlertController and UISearchController

Summary

Improved API through UIAlertController and UISearchController

Presentations are now in your hands

Summary

Improved API through UIAlertController and UISearchController

Presentations are now in your hands

- Go make your own Cool Mode!

More Information

Jake Behrens
App Frameworks Evangelist
behrens@apple.com

Documentation
View Controllers and Presentation Controllers
http://developer.apple.com

Apple Developer Forums
http://devforums.apple.com

Related Sessions

View Controller Advancements in iOS 8	Mission	Wednesday 9:00AM
Building Adaptive Apps with UIKit	Mission	Wednesday 10:15AM
What's New in Interface Builder	Mission	Wednesday 3:15PM
Core iOS Application Architectural Patterns	Mission	Thursday 9:00AM

Labs

• Cocoa Touch Lab	Frameworks Lab A	Thursday 2:00PM

WWDC14