Application State Restoration

Saving and Restoring Application State on iOS

Session 208

Gordie Freedman
iOS UIKit Engineer

These are confidential sessions—please refrain from streaming, blogging, or taking pictures

Welcome

CAFFE
NORTH BEACH RESTAURANT
NORTH BEACH RESTAURANT
NORTH BEACH RESTAURANT

Agenda

Agenda

- What is it

Agenda

- What is it
- Conceptual approach

Agenda

- What is it
- Conceptual approach
- How to incorporate it

Agenda

- What is it
- Conceptual approach
- How to incorporate it
- Tips and tricks

New for iOS 6 in UIKit

What Is It?

New

What Is It?

New

- Preserve app state

What Is It?

- Preserve app state
    - Appearance

What Is It?

New

- Preserve app state
  - Appearance
  - Behavior

What Is It?

New

- Preserve app state
  - Appearance
  - Behavior
- Restart is like waking from the background

Feature Details

New

Feature Details

- Similar to OS X app lifecycle

Feature Details

- Similar to OS X app lifecycle
- Based on view controllers

Feature Details

- Similar to OS X app lifecycle
- Based on view controllers
- Opt-in

Feature Details

- Similar to OS X app lifecycle
- Based on view controllers
- Opt-in
- Phase it into your app

Semantic State

Semantic State

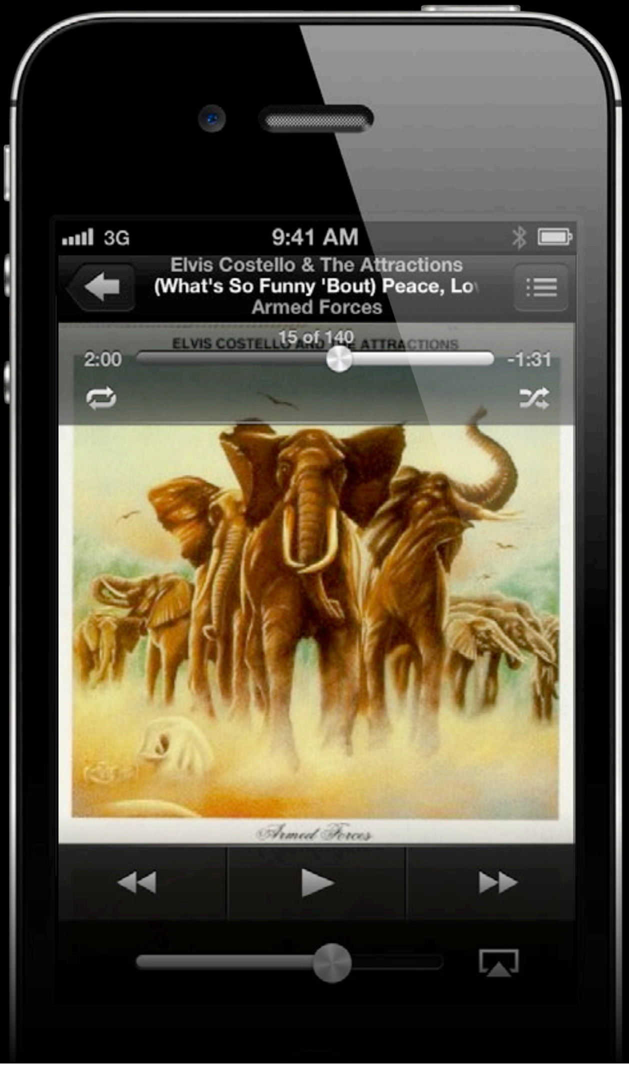
- User experience
  - What the user is doing

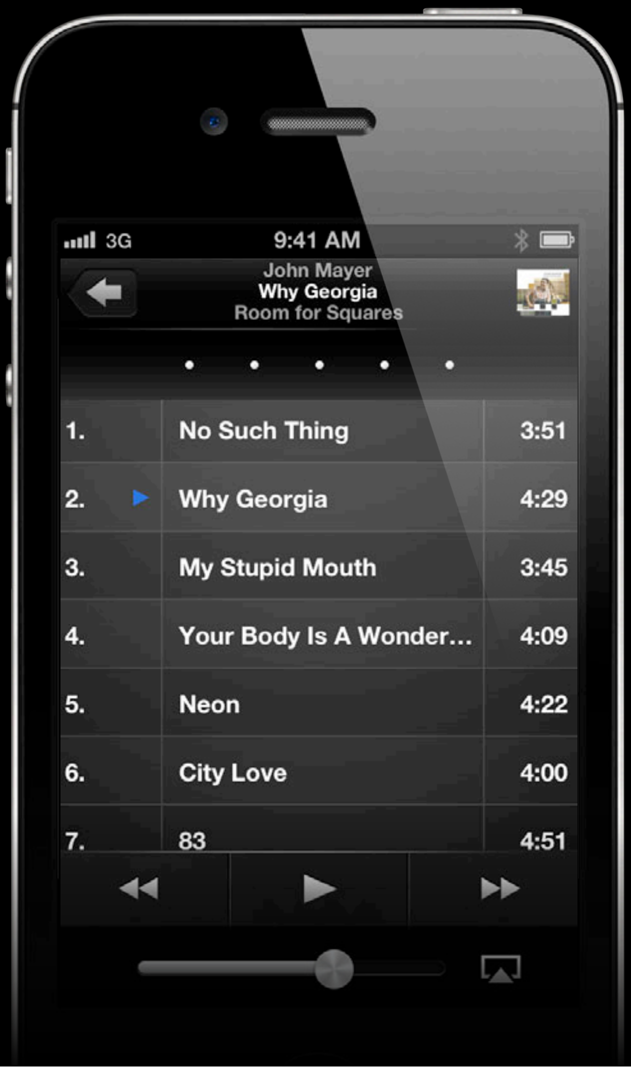
Semantic State

- User experience
  - What the user is doing
- Running state that isn’t visible
  - Computation (i.e., image filter)
  - Specific server connection

Semantic State

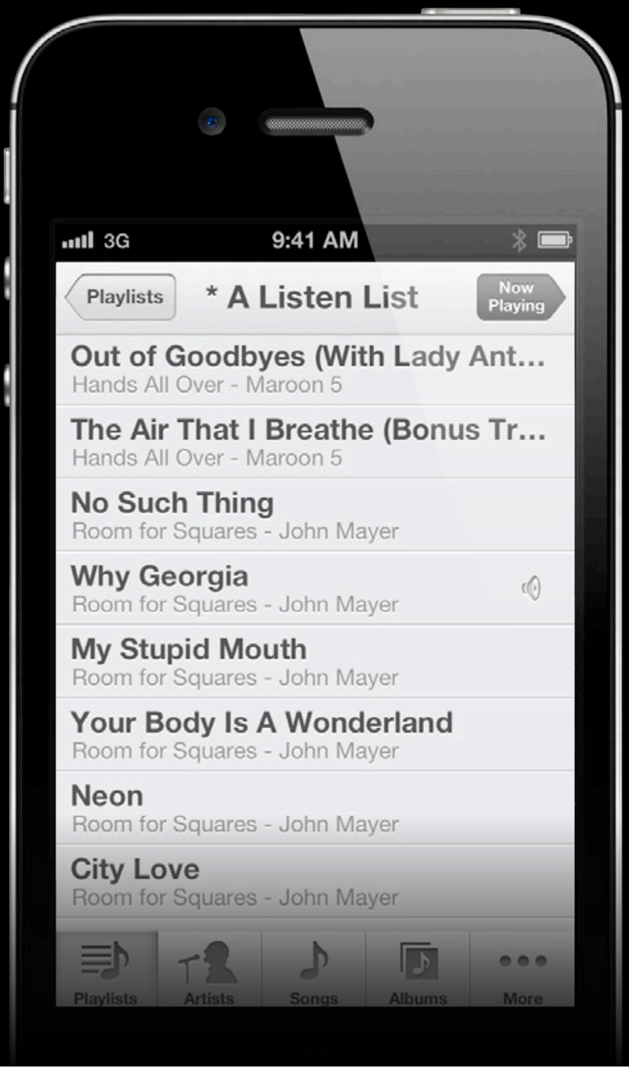
- User experience
  - What the user is doing
- Running state that isn’t visible
  - Computation (i.e., image filter)
  - Specific server connection
- Think like a user


3G
9:41 AM
Elvis Costello & The Attractions
(What's So Funny 'Bout) Peace, Lo
Armed Forces
15 of 140
ELVIS COSTELLO AND THE ATTRACTIONS
2:00
-1:31
Armed Forces



John Mayer
Why Georgia
Room for Squares

1.	No Such Thing	3:51
2. ▶	Why Georgia	4:29
3.	My Stupid Mouth	3:45
4.	Your Body Is A Wonder...	4:09
5.	Neon	4:22
6.	City Love	4:00
7.	83	4:51



* A Listen List

Out of Goodbyes (With Lady Ant…
Hands All Over - Maroon 5

The Air That I Breathe (Bonus Tr…
Hands All Over - Maroon 5

No Such Thing
Room for Squares - John Mayer

Why Georgia
Room for Squares - John Mayer

My Stupid Mouth
Room for Squares - John Mayer

Your Body Is A Wonderland
Room for Squares - John Mayer

Neon
Room for Squares - John Mayer

City Love
Room for Squares - John Mayer

So Why Do This?

So Why Do This?

- Better user experience

So Why Do This?

- Better user experience
  - Avoids uncertainty

So Why Do This?

- Better user experience
  - Avoids uncertainty
  - Saves the user’s time

So Why Do This?

- Better user experience
  - Avoids uncertainty
  - Saves the user’s time
  - More pleasant

So Why Do This?

- Better user experience
    - Avoids uncertainty
    - Saves the user’s time
    - More pleasant
- Appears like the app never quit

So Why Do This?

- Better user experience
  - Avoids uncertainty
  - Saves the user’s time
  - More pleasant
- Appears like the app never quit
- Makes your app more competitive

iOS Makes It Easy

iOS Makes It Easy

- iOS handles the accounting

iOS Makes It Easy

- iOS handles the accounting
- iOS provides default behaviors

iOS Makes It Easy

- iOS handles the accounting
- iOS provides default behaviors
- Easy for you to adopt

iOS Makes It Easy

- iOS handles the accounting
- iOS provides default behaviors
- Easy for you to adopt
- Consistent approach

iOS Makes It Easy

- iOS handles the accounting
- iOS provides default behaviors
- Easy for you to adopt
- Consistent approach
- Guides you to save the right thing

Your Responsibilities

Your Responsibilities

- Handle the semantics

Your Responsibilities

- Handle the semantics
- Identify your app’s state

Your Responsibilities

- Handle the semantics
- Identify your app’s state
- Implement save/restore methods

Your Responsibilities

- Handle the semantics
- Identify your app’s state
- Implement save/restore methods
- Recreate your view controllers and views

Your Responsibilities

- Handle the semantics
- Identify your app's state
- Implement save/restore methods
- Recreate your view controllers and views

Don't save model/view ✗

Demo

WWDC App

Demo Recap

Demo Recap

- Brought the user back to where they were

Demo Recap

- Brought the user back to where they were
- Remembered things that weren’t visible

Demo Recap

- Brought the user back to where they were
- Remembered things that weren’t visible
- How much code was that?

Demo Recap

- Brought the user back to where they were
- Remembered things that weren’t visible
- How much code was that?
  - There is a free lunch

Demo Recap

- Brought the user back to where they were
- Remembered things that weren't visible
- How much code was that?
    - There is a free lunch
    - Or at least a free appetizer

What Do You Have to Do?

What Do You Have to Do?

- App delegate must opt-in

What Do You Have to Do?

- App delegate must opt-in
- Add restoration identifiers to:
  - Views
  - View controllers

Restoration Identifiers

Restoration Identifiers

- Property on views and view controllers

Restoration Identifiers

- Property on views and view controllers
- Can set in Xcode/IB

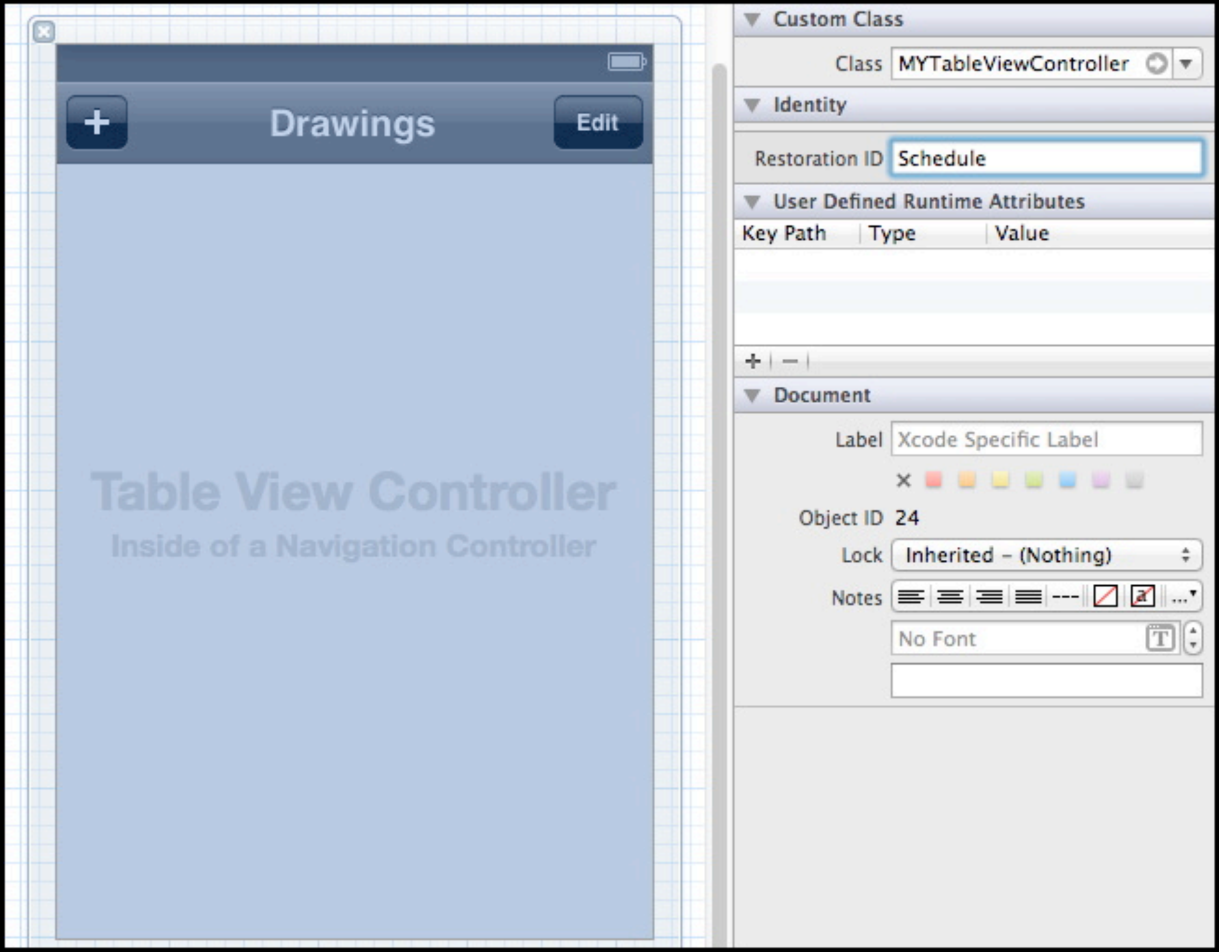
Restoration Identifiers

- Property on views and view controllers
- Can set in Xcode/IB
- Used to indicate what to save

Restoration Identifiers

- Property on views and view controllers
- Can set in Xcode/IB
- Used to indicate what to save
- Used to map and find objects

Restoration Identifiers



Custom Class

Class: MYTableViewController

Identity

Restoration ID: Schedule

User Defined Runtime Attributes

Key Path	Type	Value

Document

Label: Xcode Specific Label

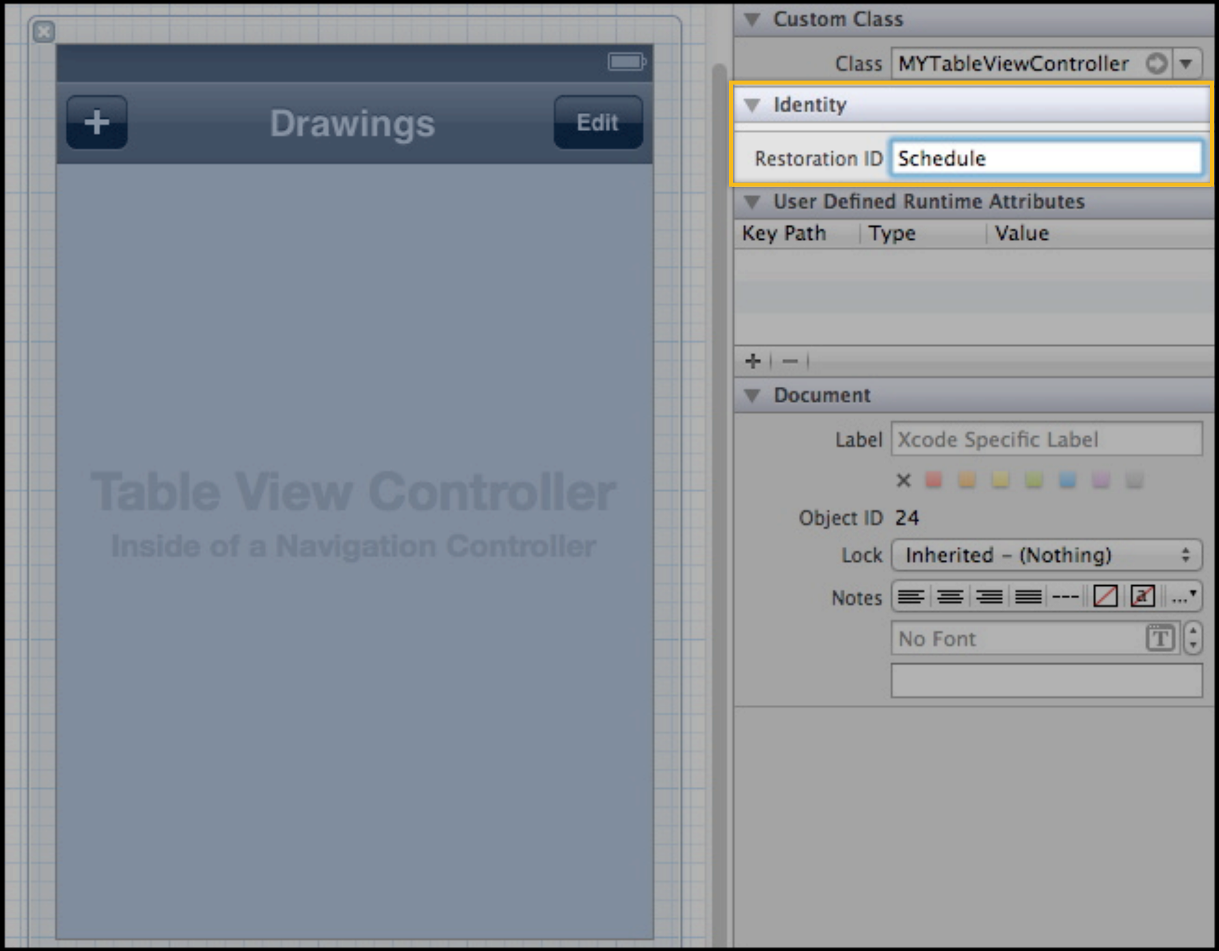
Object ID: 24

Lock: Inherited – (Nothing)

Notes

No Font

Restoration Identifiers



Class MYTableViewController

Restoration ID Schedule

Label Xcode Specific Label

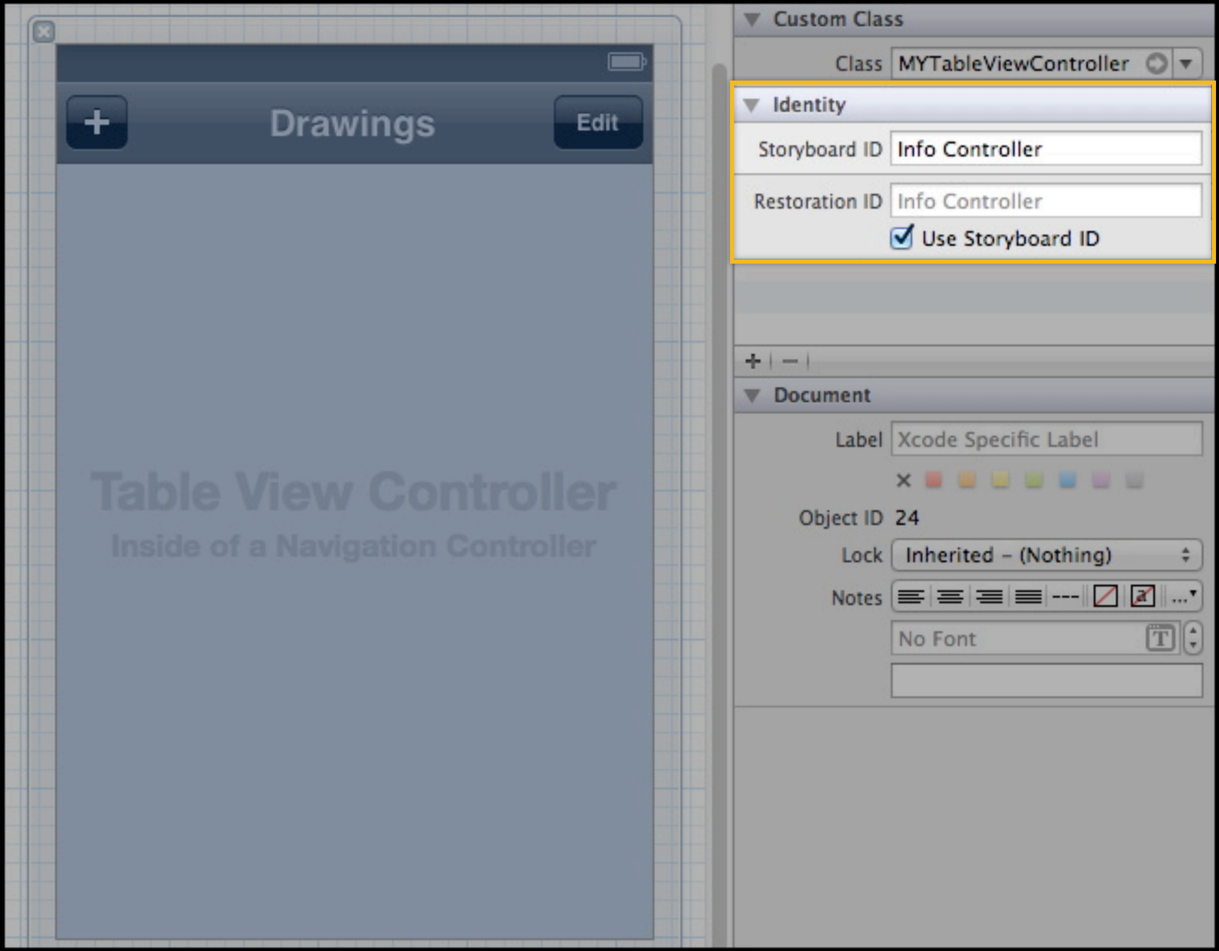
Object ID 24

Lock Inherited - (Nothing)

Notes

No Font

Restoration Identifiers



Custom Class

Class: MYTableViewController

Identity

Storyboard ID: Info Controller

Restoration ID: Info Controller

☑ Use Storyboard ID

Document

Label: Xcode Specific Label

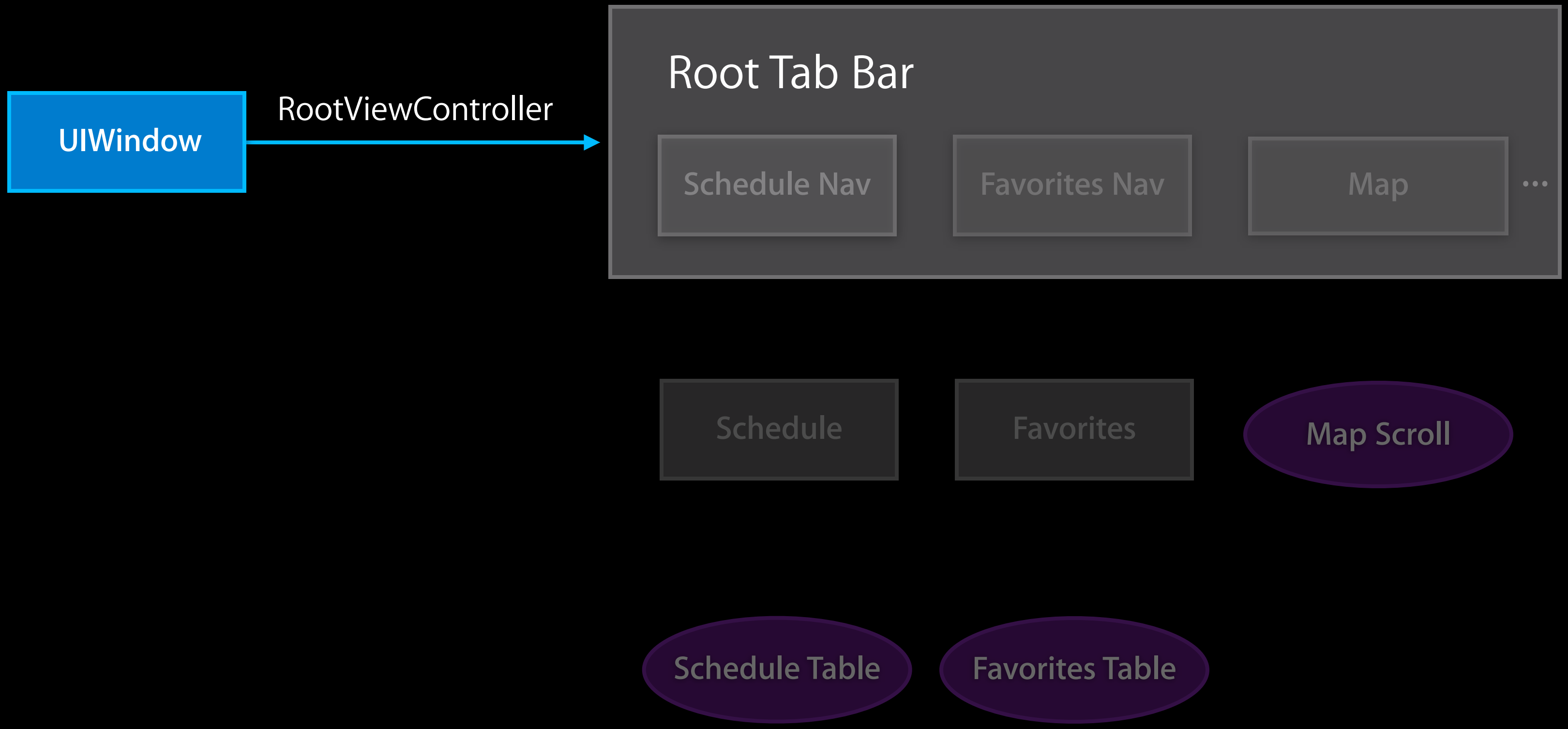
Object ID: 24

Lock: Inherited - (Nothing)

Notes

No Font

Restoration Identifiers



UIWindow

RootViewController

Root Tab Bar

Schedule Nav

Favorites Nav

Map

...

Schedule

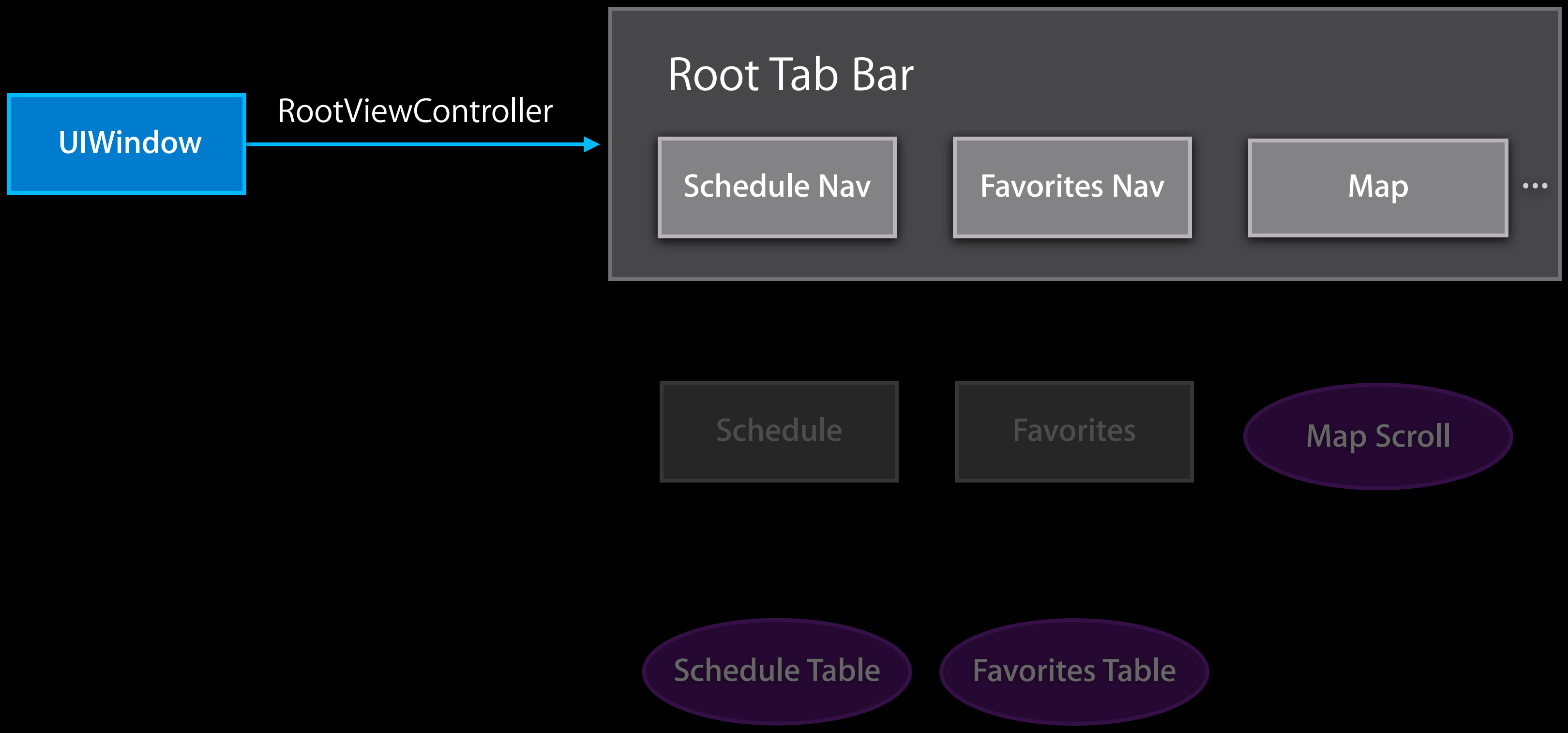
Favorites

Map Scroll

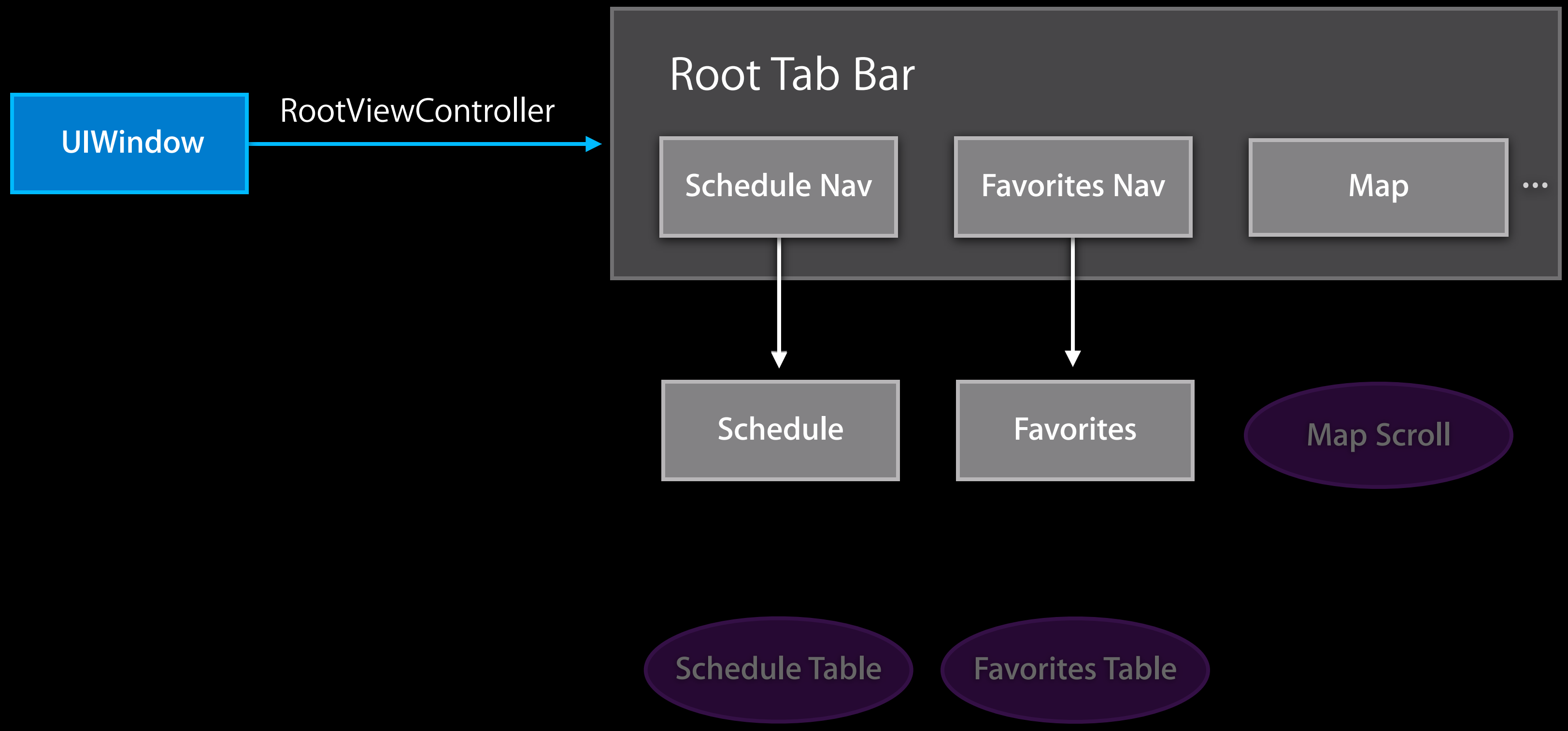
Schedule Table

Favorites Table

Restoration Identifiers



Restoration Identifiers



UIWindow

RootViewController

Root Tab Bar

Schedule Nav

Favorites Nav

Map

...

Schedule

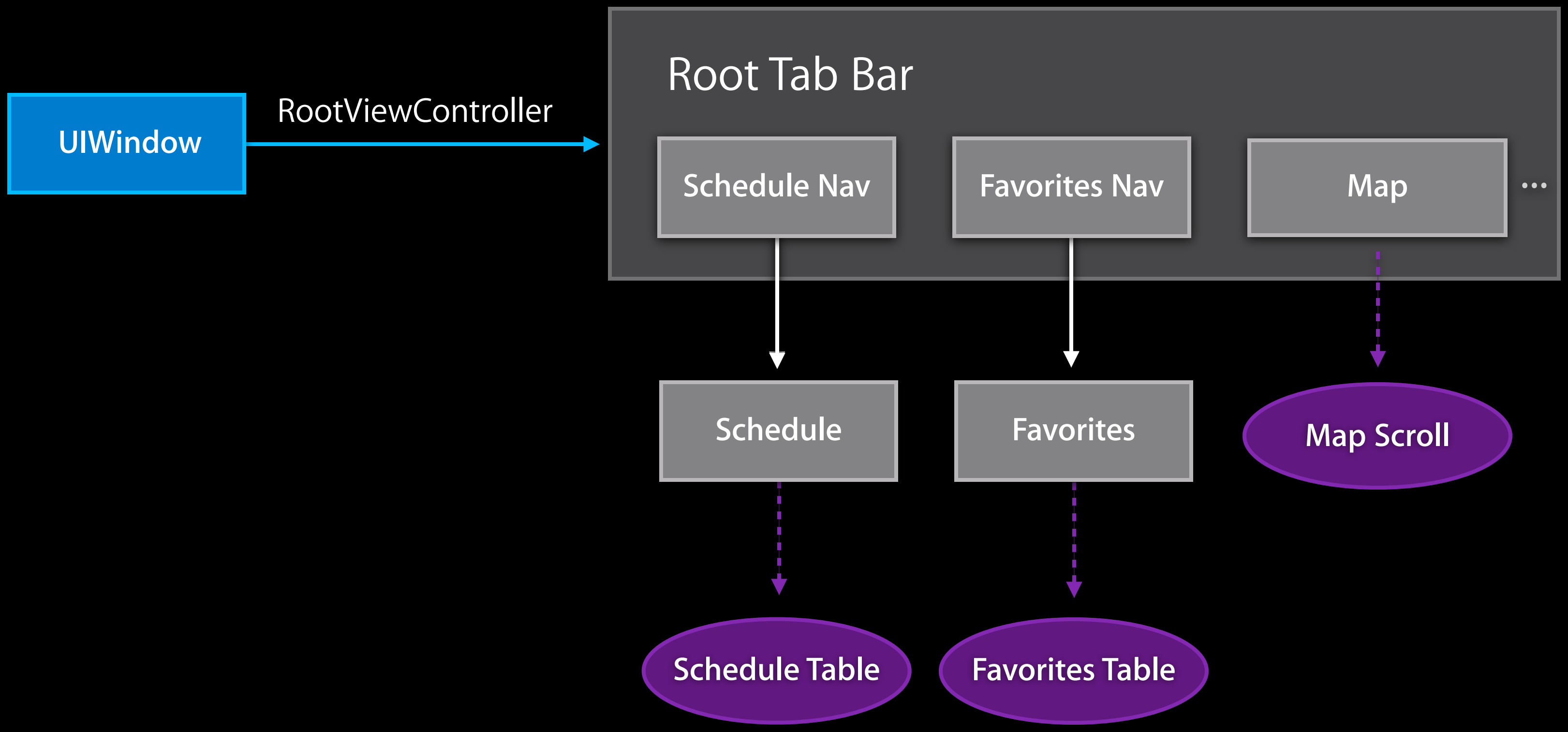
Favorites

Map Scroll

Schedule Table

Favorites Table

Restoration Identifiers



UIWindow

RootViewController

Root Tab Bar

Schedule Nav

Favorites Nav

Map

...

Schedule

Favorites

Map Scroll

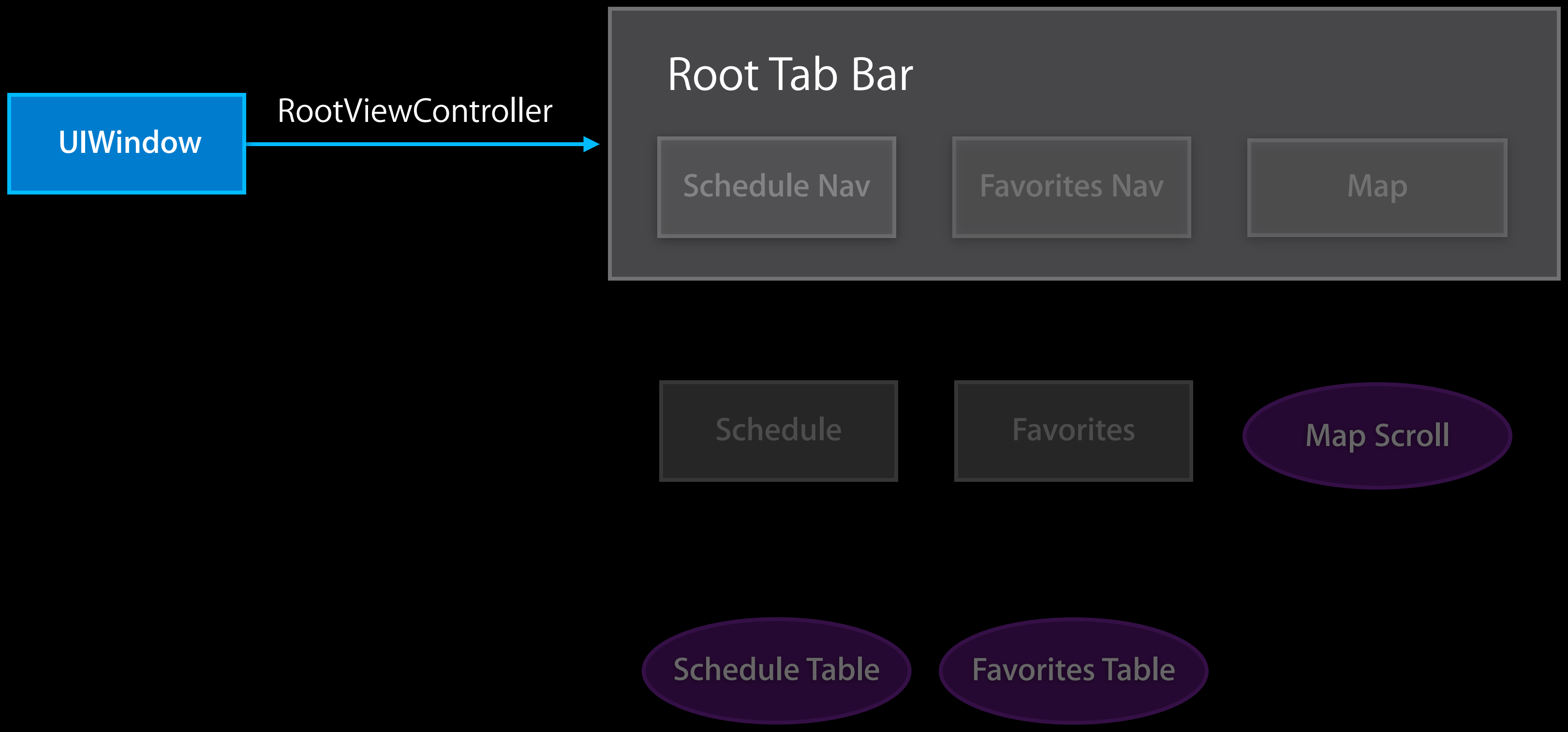
Schedule Table

Favorites Table

Restoration Identifier Path

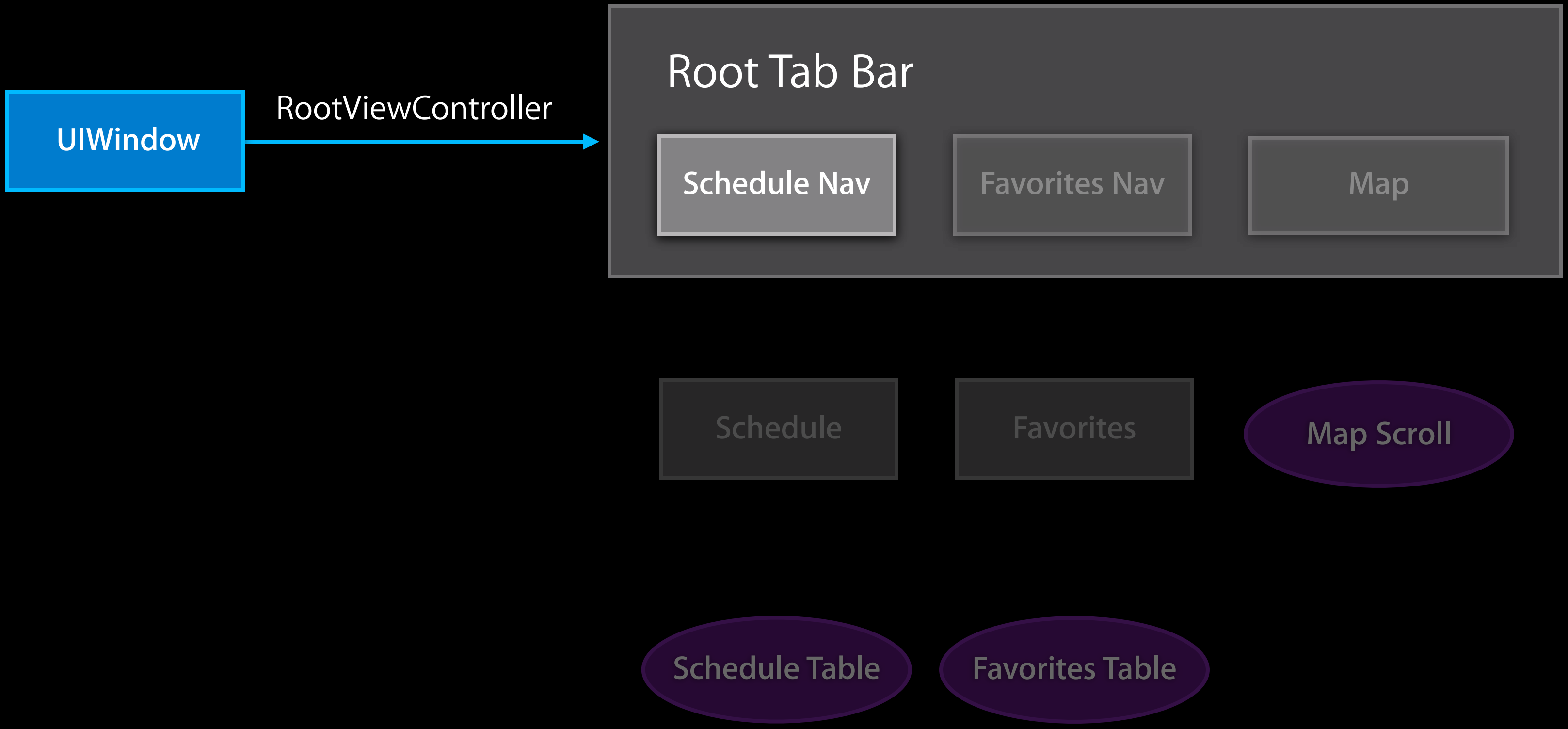
Restoration Identifier Path

Root Tab Bar



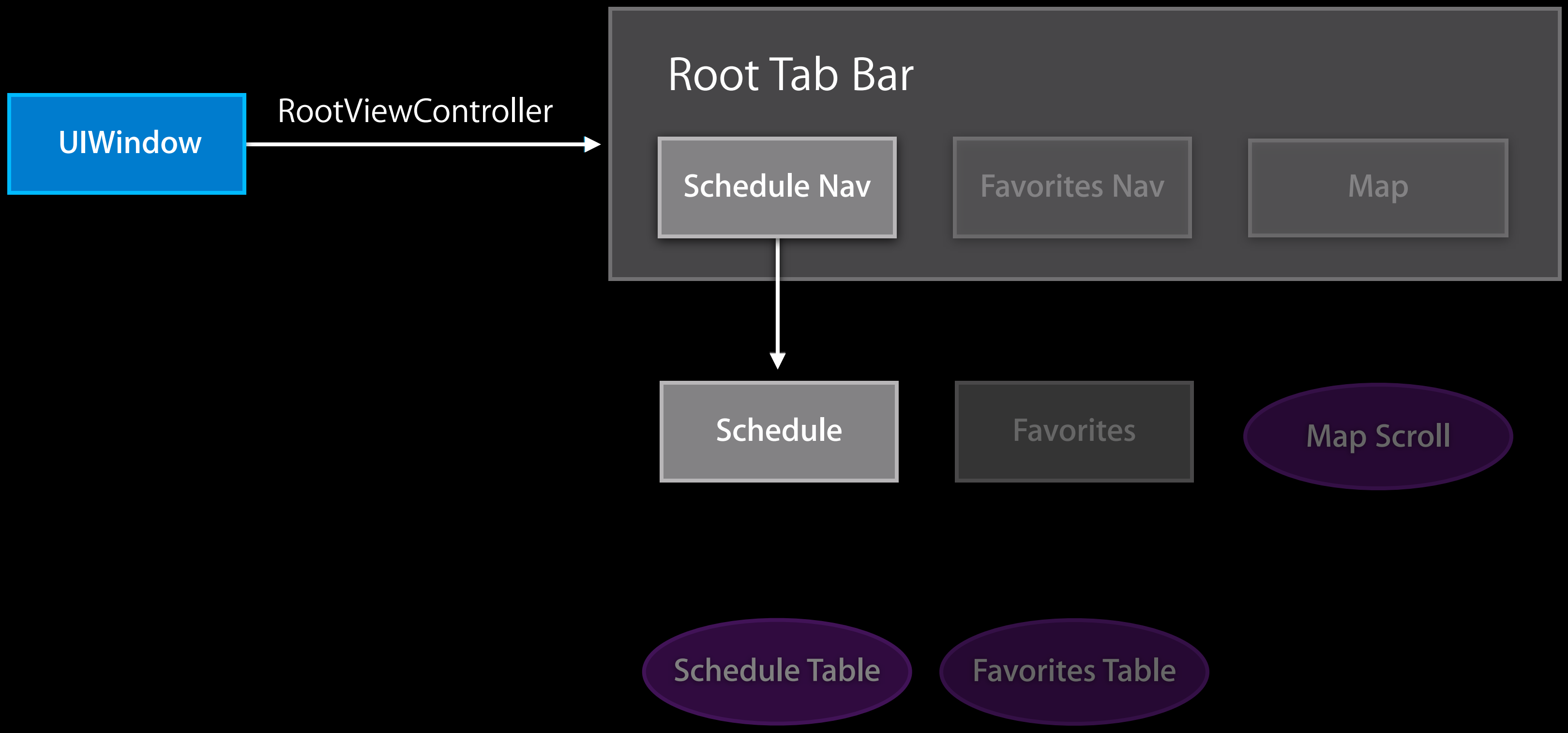
Restoration Identifier Path

Root Tab Bar/Schedule Nav



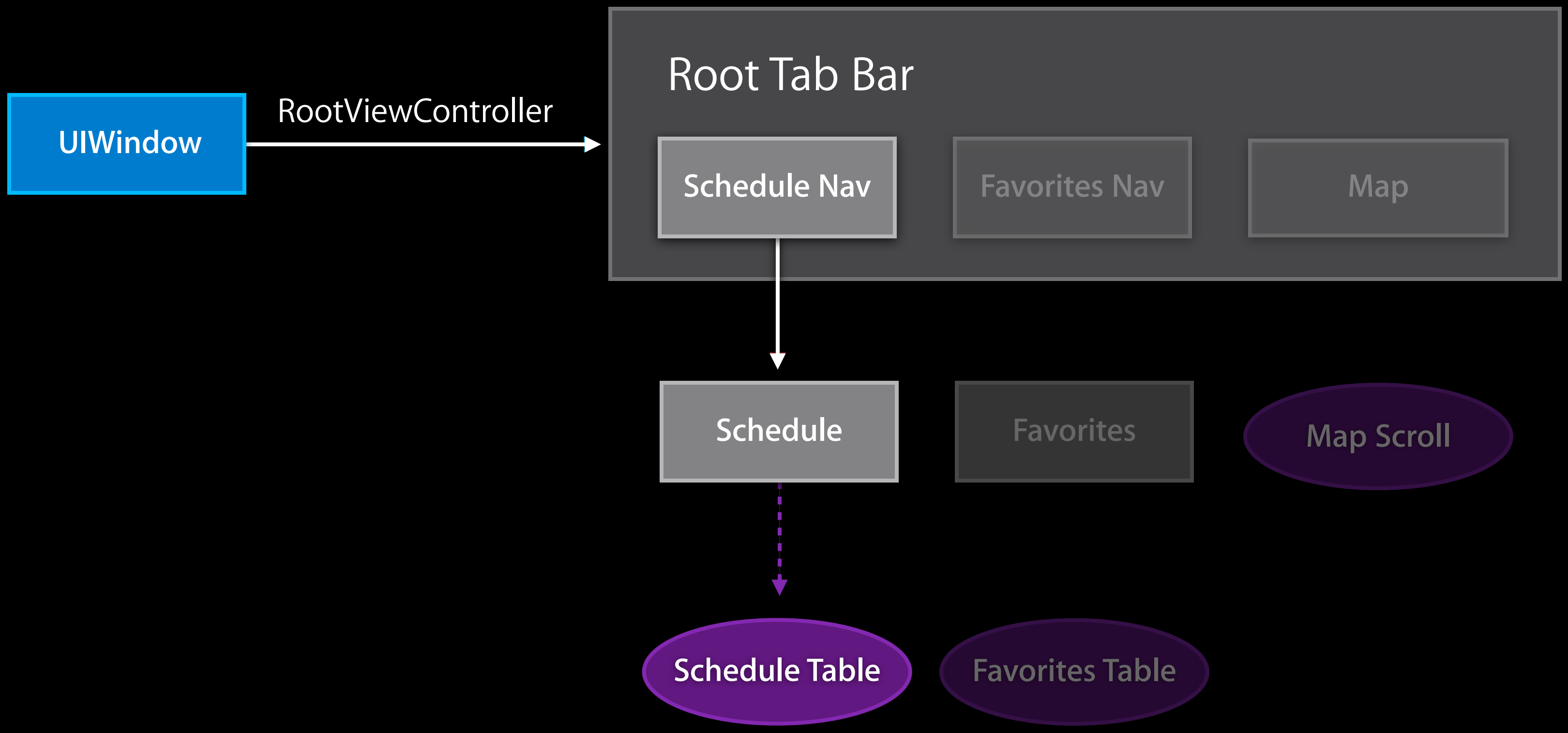
Restoration Identifier Path

Root Tab Bar/Schedule Nav/Schedule

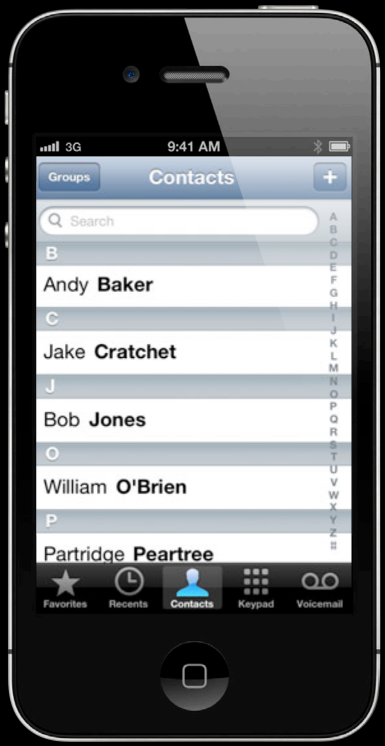


Restoration Identifier Path

Root Tab Bar/Schedule Nav/Schedule/Schedule Table



Using the Same Identifier



B

Andy **Baker**

C

Jake **Cratchet**

J

Bob **Jones**

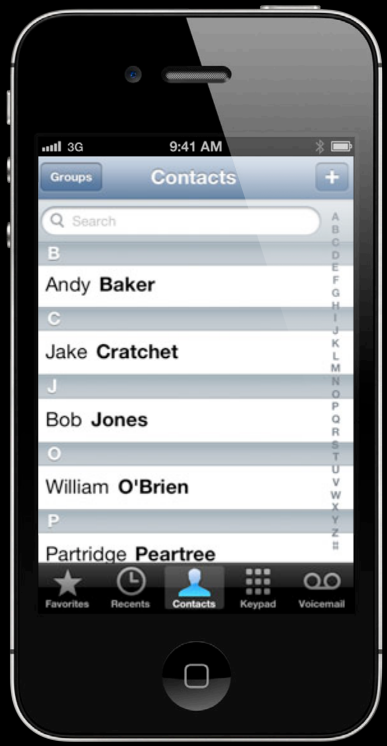
O

William **O'Brien**

P

Partridge **Peartree**

Using the Same Identifier



B

Andy **Baker**

C

Jake **Cratchet**

J

Bob **Jones**

O

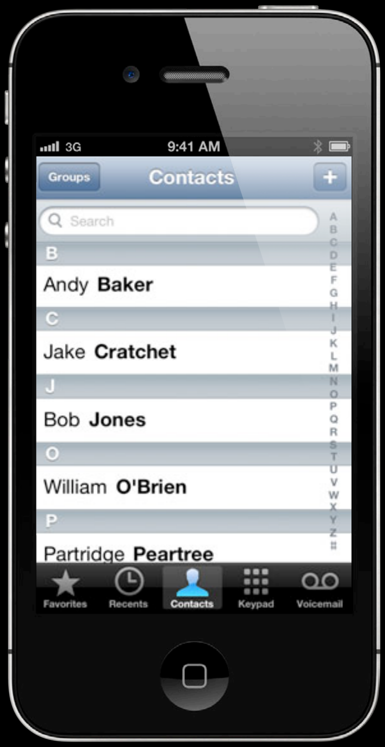
William **O'Brien**

P

Partridge **Peartree**

Contact List

Using the Same Identifier



B

Andy Baker

C

Jake Cratchet

J

Bob Jones

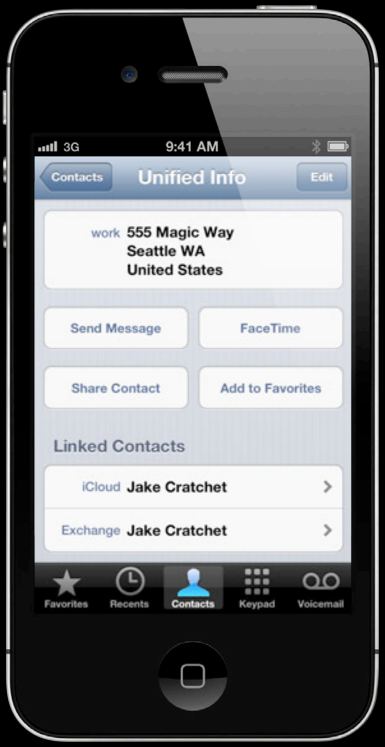
O

William O'Brien

P

Partridge Peartree

Contact List



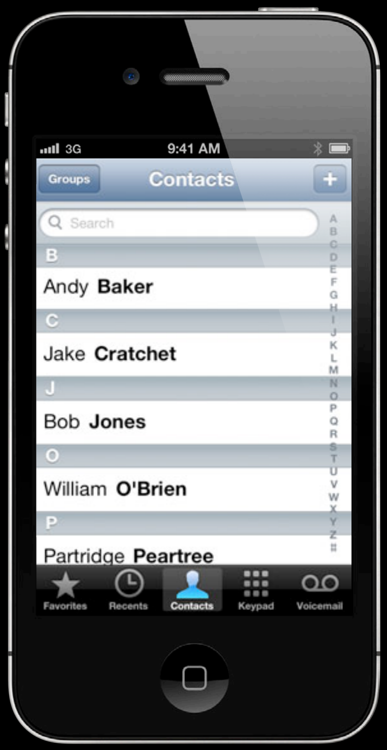
work 555 Magic Way
Seattle WA
United States

Linked Contacts

iCloud Jake Cratchet

Exchange Jake Cratchet

Using the Same Identifier



B

Andy **Baker**

C

Jake **Cratchet**

J

Bob **Jones**

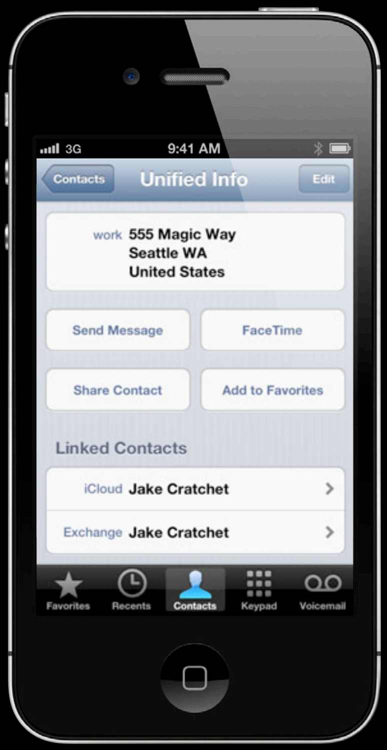
O

William **O'Brien**

P

Partridge **Peartree**

Contact List



work **555 Magic Way**
Seattle WA
United States

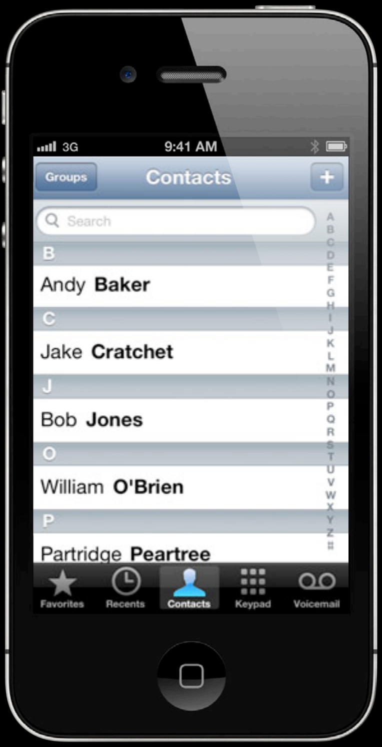
Linked Contacts

iCloud **Jake Cratchet**

Exchange **Jake Cratchet**

Contact Details

Using the Same Identifier



B

Andy **Baker**

C

Jake **Cratchet**

J

Bob **Jones**

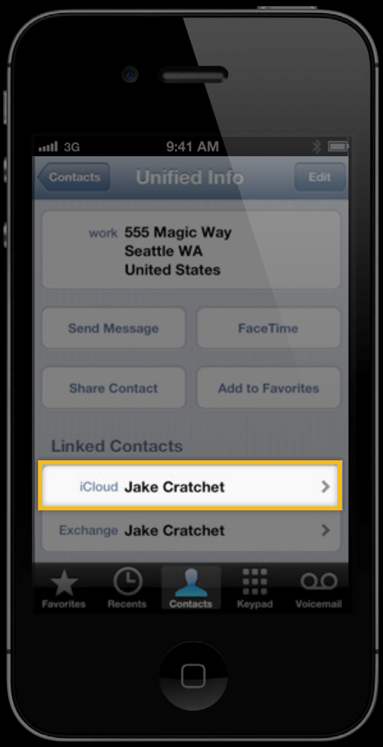
O

William **O'Brien**

P

Partridge **Peartree**

Contact List



work **555 Magic Way**
Seattle WA
United States

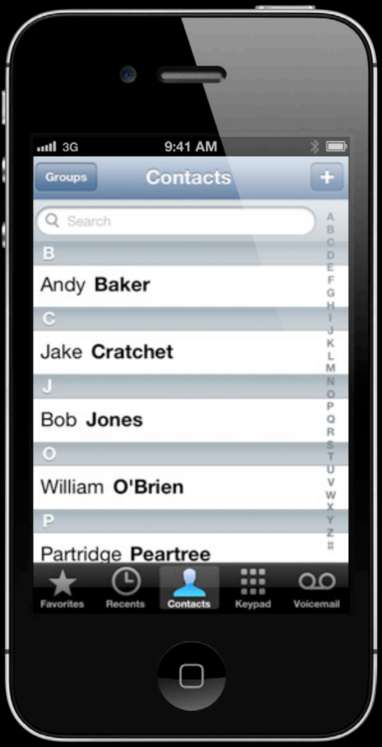
Linked Contacts

iCloud **Jake Cratchet**

Exchange **Jake Cratchet**

Contact Details

Using the Same Identifier



B

Andy **Baker**

C

Jake **Cratchet**

J

Bob **Jones**

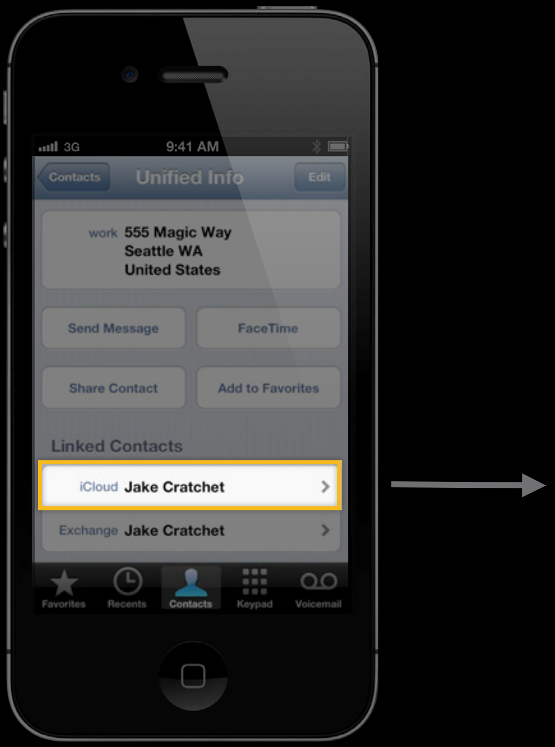
O

William **O'Brien**

P

Partridge **Peartree**

Contact List



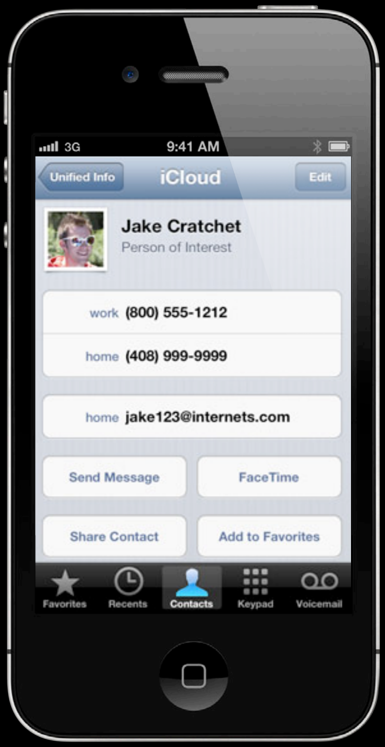
work **555 Magic Way**
Seattle WA
United States

Linked Contacts

iCloud **Jake Cratchet**

Exchange **Jake Cratchet**

Contact Details



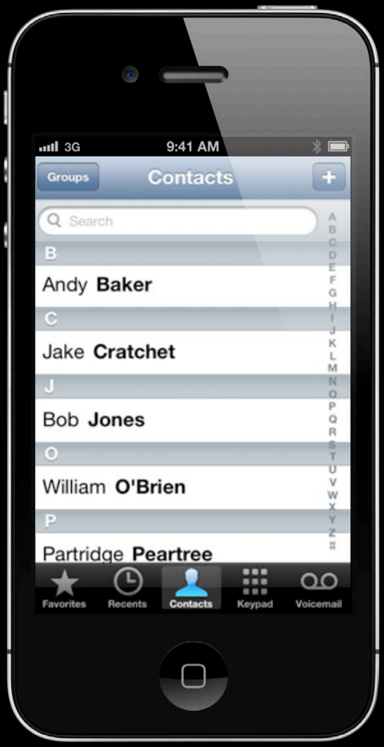
Jake Cratchet
Person of Interest

work **(800) 555-1212**

home **(408) 999-9999**

home **jake123@internets.com**

Using the Same Identifier



B

Andy Baker

C

Jake Cratchet

J

Bob Jones

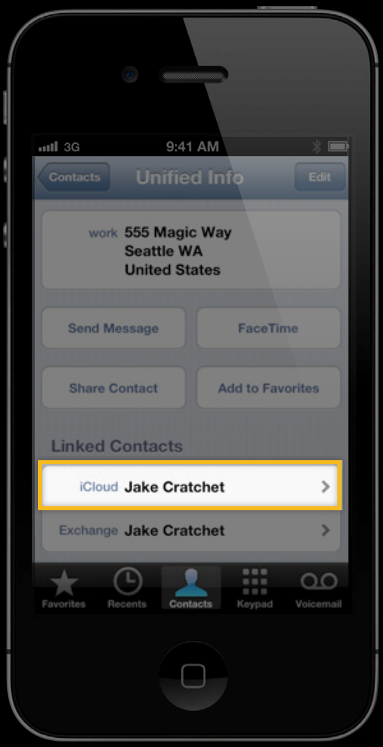
O

William O'Brien

P

Partridge Peartree

Contact List



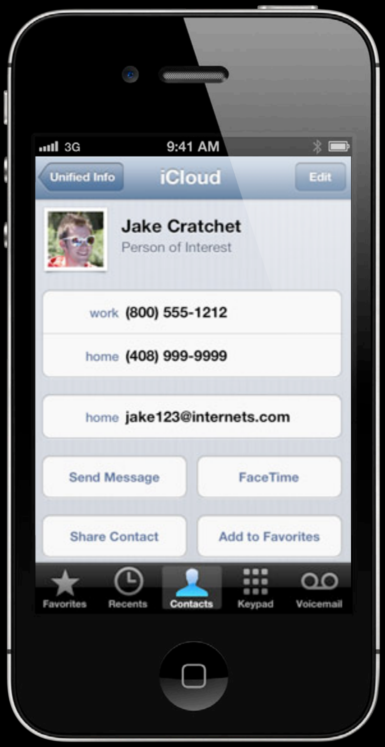
work 555 Magic Way
Seattle WA
United States

Linked Contacts

iCloud Jake Cratchet

Exchange Jake Cratchet

Contact Details



Jake Cratchet
Person of Interest

work (800) 555-1212

home (408) 999-9999

home jake123@internets.com

Contact Details

Demo

Dynamic View Controllers

What Do You Have to Do?

What Do You Have to Do?

- Save state
  - Current day
  - Session
  - Filter shared state

What Do You Have to Do?

- Save state
  - Current day
  - Session
  - Filter shared state
- Find/create view controllers
  - Session details in favorites
  - Presented filter controller

What Do You Have to Do?

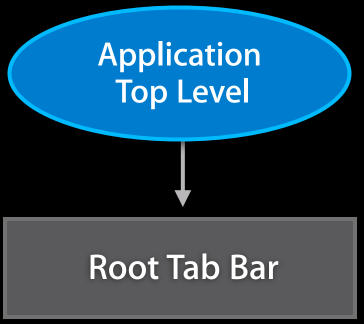
- Save state
    - Current day
    - Session
    - Filter shared state
- Find/create view controllers
    - Session details in favorites
    - Presented filter controller
- Restore state

Restoration Archive Structure

Restoration Archive Structure

Application
Top Level

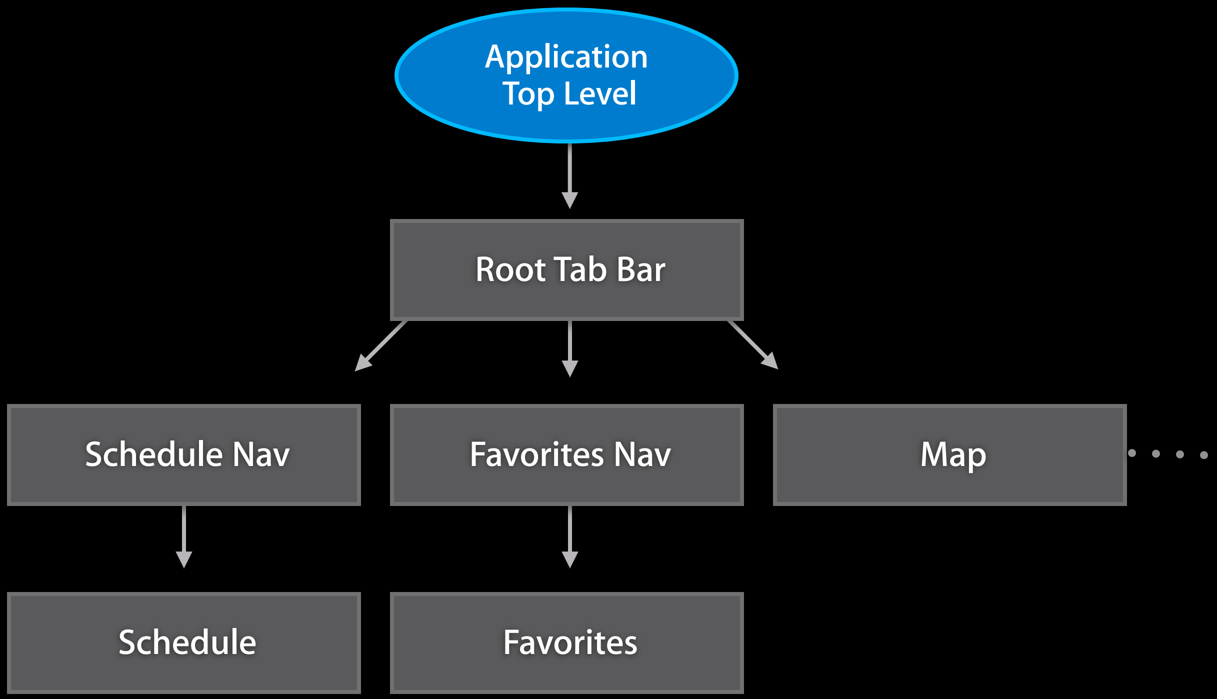
Restoration Archive Structure



Application Top Level

Root Tab Bar

Restoration Archive Structure



Application Top Level

Root Tab Bar

Schedule Nav

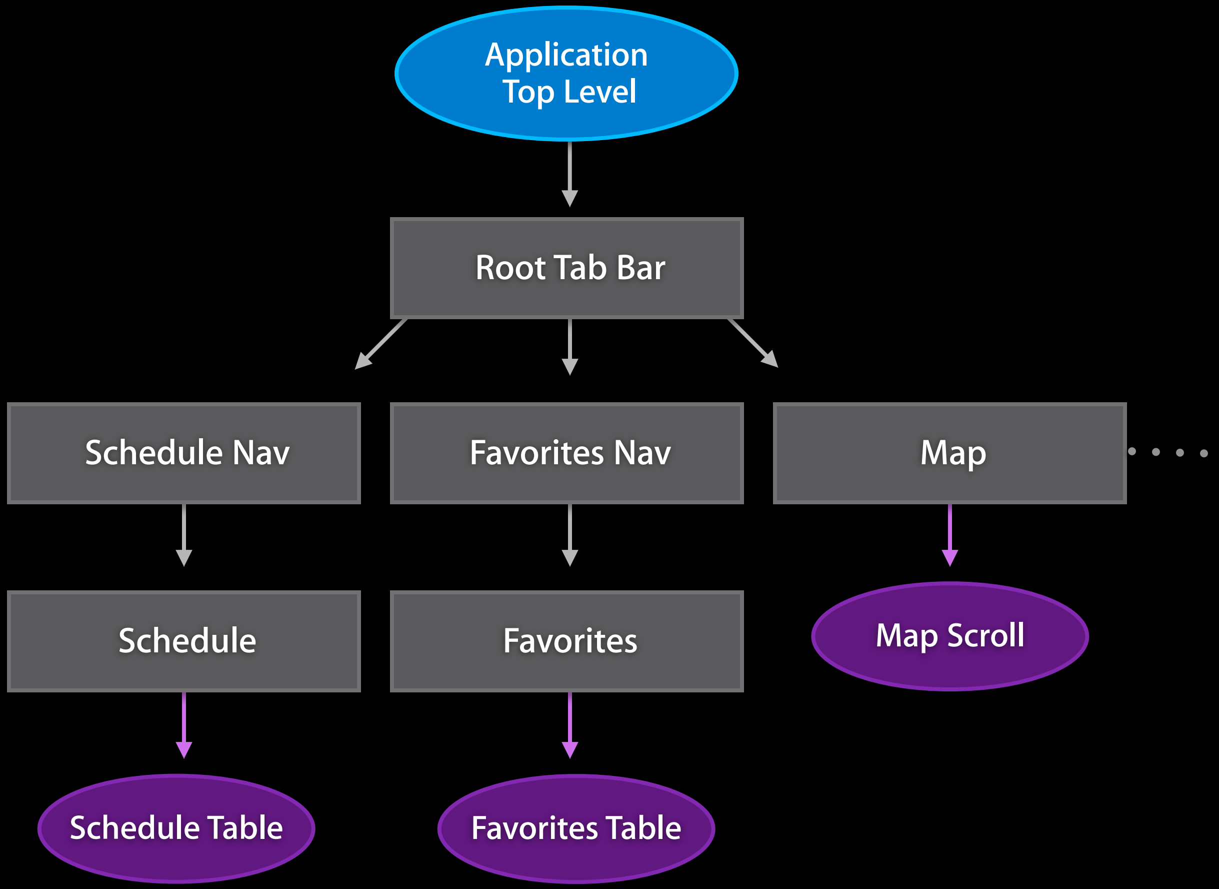
Favorites Nav

Map

Schedule

Favorites

Restoration Archive Structure



Application Top Level

Root Tab Bar

Schedule Nav

Favorites Nav

Map

Schedule

Favorites

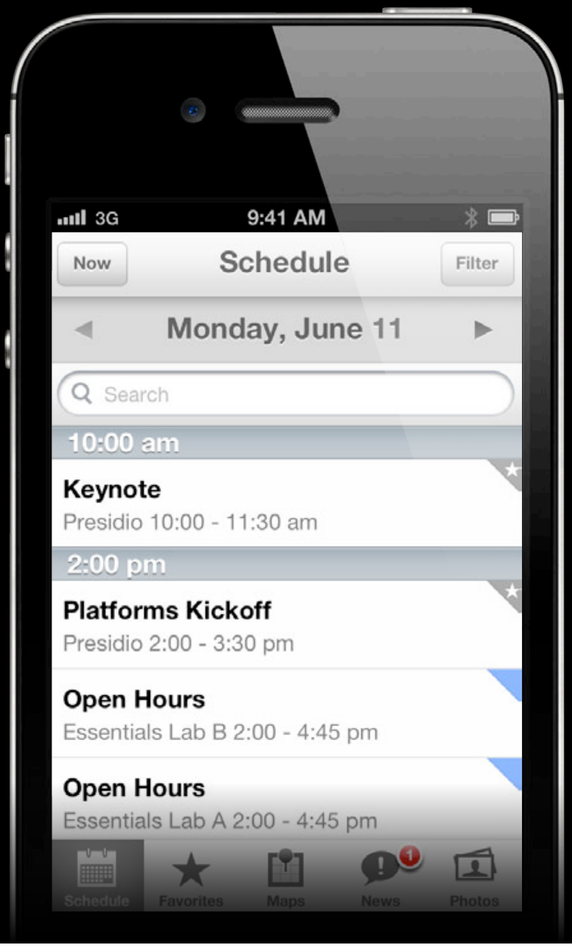
Map Scroll

Schedule Table

Favorites Table

Saving Day, Session, Filter

Schedule Day



Monday, June 11

10:00 am

Keynote
Presidio 10:00 - 11:30 am

2:00 pm

Platforms Kickoff
Presidio 2:00 - 3:30 pm

Open Hours
Essentials Lab B 2:00 - 4:45 pm

Open Hours
Essentials Lab A 2:00 - 4:45 pm

Schedule Day

Schedule View Controller

```
 - (void) encodeRestorableStateWithCoder:(NSCoder *)coder {
```

Schedule Day

Schedule View Controller

```
- (void) encodeRestorableStateWithCoder:(NSCoder *)coder {
        [coder encodeInt:self.tableDataSource.selectedDateIndex
          forKey:kSelectedDateKey];
```

Schedule Day

Schedule View Controller

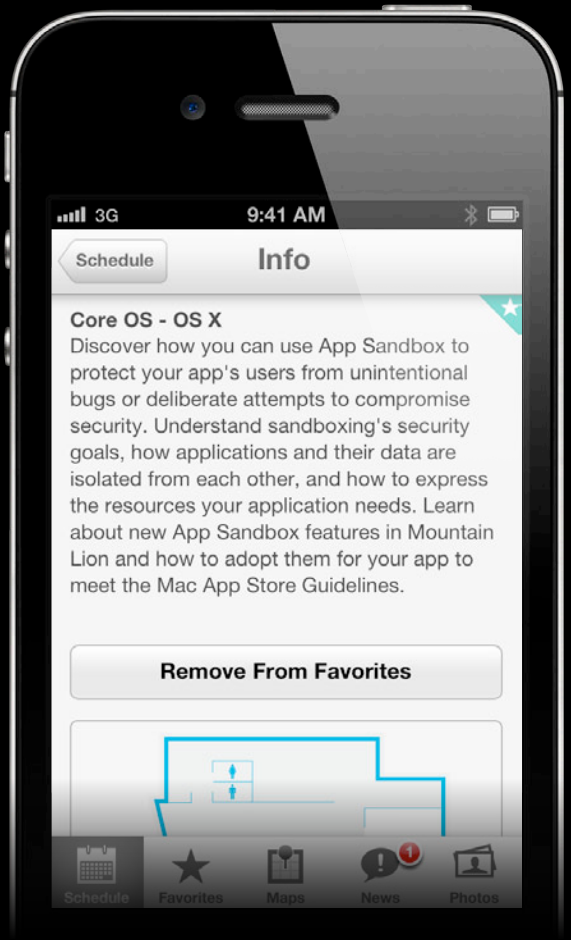
```
– (void) encodeRestorableStateWithCoder:(NSCoder *)coder {
        [coder encodeInt:self.tableDataSource.selectedDateIndex
          forKey:kSelectedDateKey];
        [super encodeRestorableStateWithCoder:coder];
}
```

Schedule Day

Schedule View Controller

```
#define kSelectedDateKey @"SelectedDateKey"
 - (void) encodeRestorableStateWithCoder:(NSCoder *)coder {
         [coder encodeInt:self.tableDataSource.selectedDateIndex
           forKey:kSelectedDateKey];
         [super encodeRestorableStateWithCoder:coder];
 }
```

Session Details



Core OS - OS X

Discover how you can use App Sandbox to protect your app's users from unintentional bugs or deliberate attempts to compromise security. Understand sandboxing's security goals, how applications and their data are isolated from each other, and how to express the resources your application needs. Learn about new App Sandbox features in Mountain Lion and how to adopt them for your app to meet the Mac App Store Guidelines.

Saving Session Details

Session details view controller

```
- (void) encodeRestorableStateWithCoder:(NSCoder *)coder {
```

Saving Session Details

Session details view controller

```
- (void) encodeRestorableStateWithCoder:(NSCoder *)coder {
     [coder encodeObject:self.session.sessionID forKey:kSessionIDKey];
```

Saving Session Details

Session details view controller

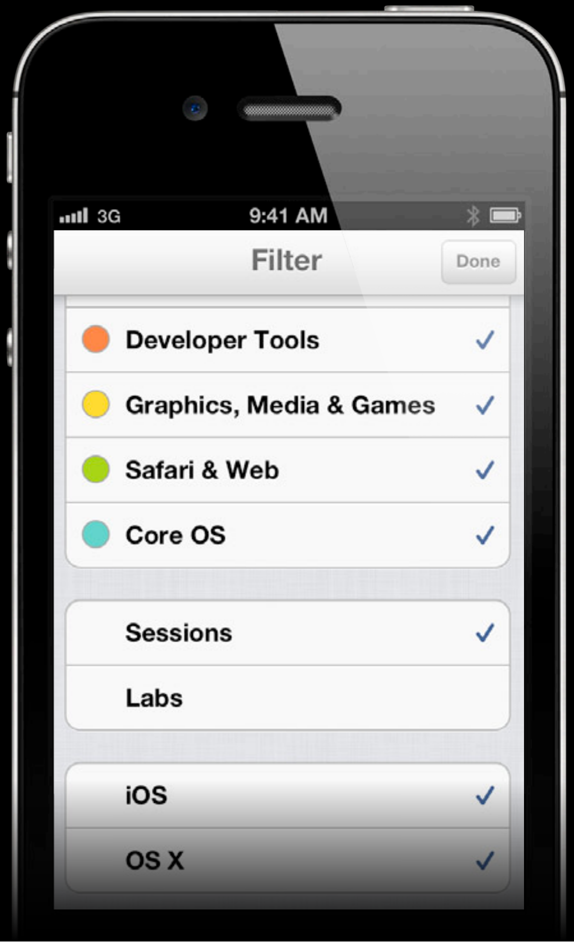
```
- (void) encodeRestorableStateWithCoder:(NSCoder *)coder {
     [coder encodeObject:self.session.sessionID forKey:kSessionIDKey];
     [coder encodeBool:self.launchedFromTheScheduleView
        forKey:kLaunchedFromScheduleViewKey];
```

Saving Session Details

Session details view controller

```
#define kSessionIDKey @"SessionID"
#define kLaunchedFromScheduleViewKey @"LaunchedFromScheduleView"
- (void) encodeRestorableStateWithCoder:(NSCoder *)coder {
     [coder encodeObject:self.session.sessionID forKey:kSessionIDKey];
     [coder encodeBool:self.launchedFromTheScheduleView
        forKey:kLaunchedFromScheduleViewKey];
     [super encodeRestorableStateWithCoder:coder];
}
```

Filter View Controller



- 🟠 Developer Tools ✓
- 🟡 Graphics, Media & Games ✓
- 🟢 Safari & Web ✓
- 🔵 Core OS ✓

- Sessions ✓
- Labs

- iOS ✓
- OS X ✓

Saving Filter View Controller State

Filter View Controller

```
- (void) encodeRestorableStateWithCoder:(NSCoder *)coder {
```

Saving Filter View Controller State

Filter View Controller

```
- (void) encodeRestorableStateWithCoder:(NSCoder *)coder {
     [coder encodeObject:self.scheduleViewController
        forKey:kScheduleViewControllerKey];
```

Saving Filter View Controller State

Filter View Controller

```
#define forKey:kScheduleViewControllerKey @"ScheduleViewController"
- (void) encodeRestorableStateWithCoder:(NSCoder *)coder {
     [coder encodeObject:self.scheduleViewController
        forKey:kScheduleViewControllerKey];
     [super encodeRestorableStateWithCoder:coder];
}
```

What About Views?

What About Views?

- Hey! We didn’t save any view state

What About Views?

- Hey! We didn’t save any view state
- UIKit took care of some of it

What About Views?

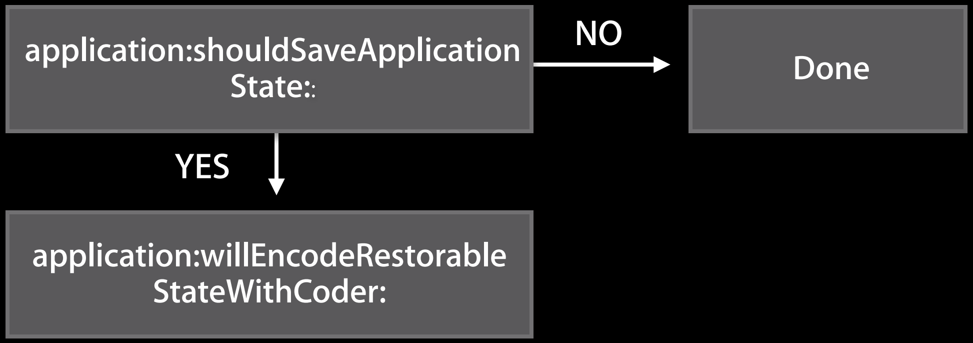
- Hey! We didn’t save any view state
- UIKit took care of some of it
- App took care of the rest naturally

Saving State: Control Flow

Saving State: Control Flow

application:shouldSaveApplicationState:

Saving State: Control Flow



application:shouldSaveApplicationState:

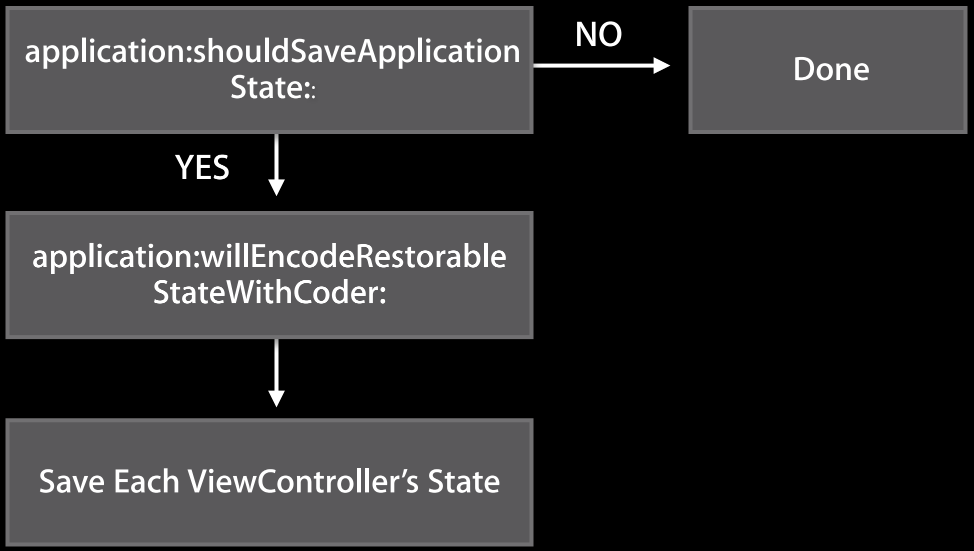
NO

Done

YES

application:willEncodeRestorableStateWithCoder:

Saving State: Control Flow



application:shouldSaveApplicationState::

NO

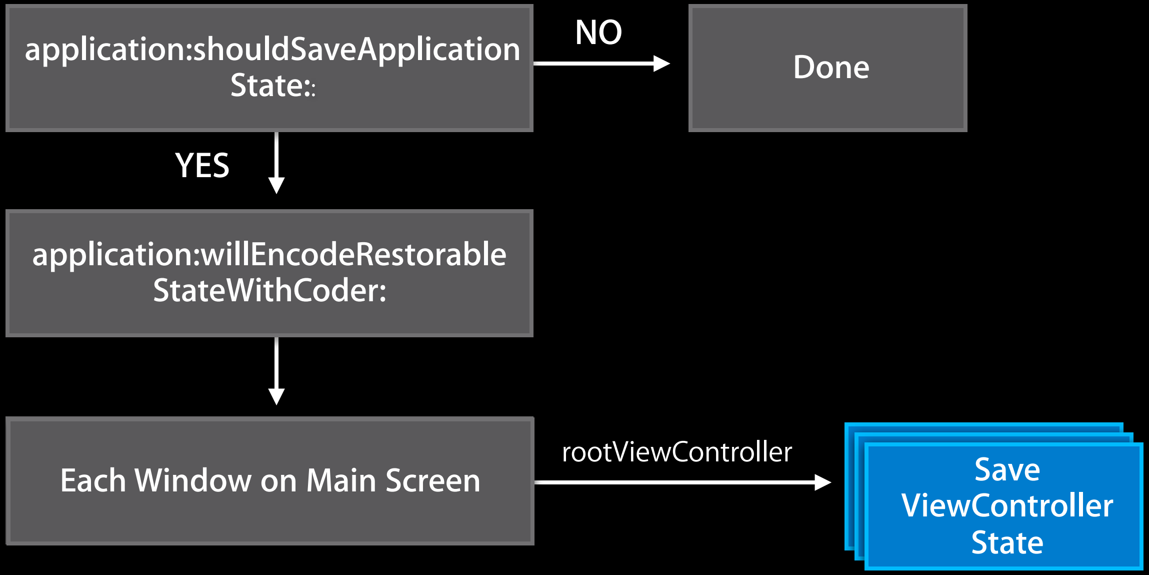
Done

YES

application:willEncodeRestorableStateWithCoder:

Save Each ViewController's State

Saving State: Control Flow



application:shouldSaveApplicationState::

NO

Done

YES

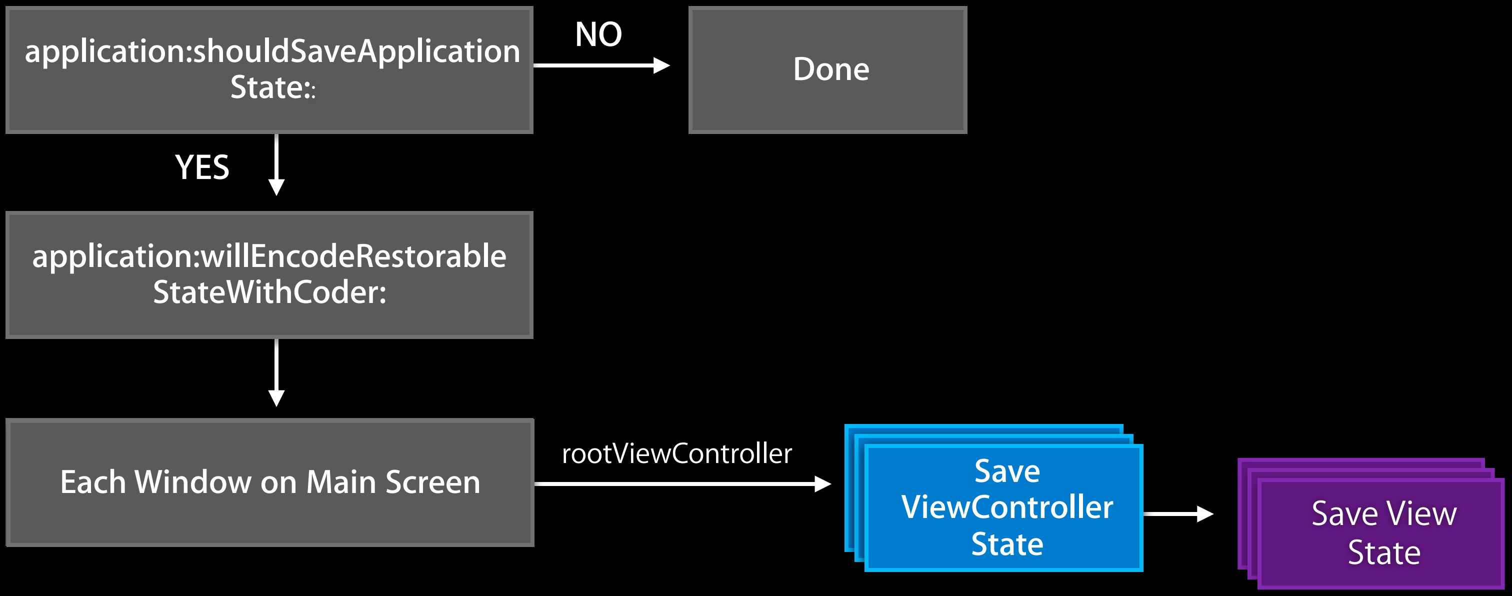
application:willEncodeRestorableStateWithCoder:

Each Window on Main Screen

rootViewController

Save ViewController State

Saving State: Control Flow



application:shouldSaveApplicationState::

NO

Done

YES

application:willEncodeRestorableStateWithCoder:

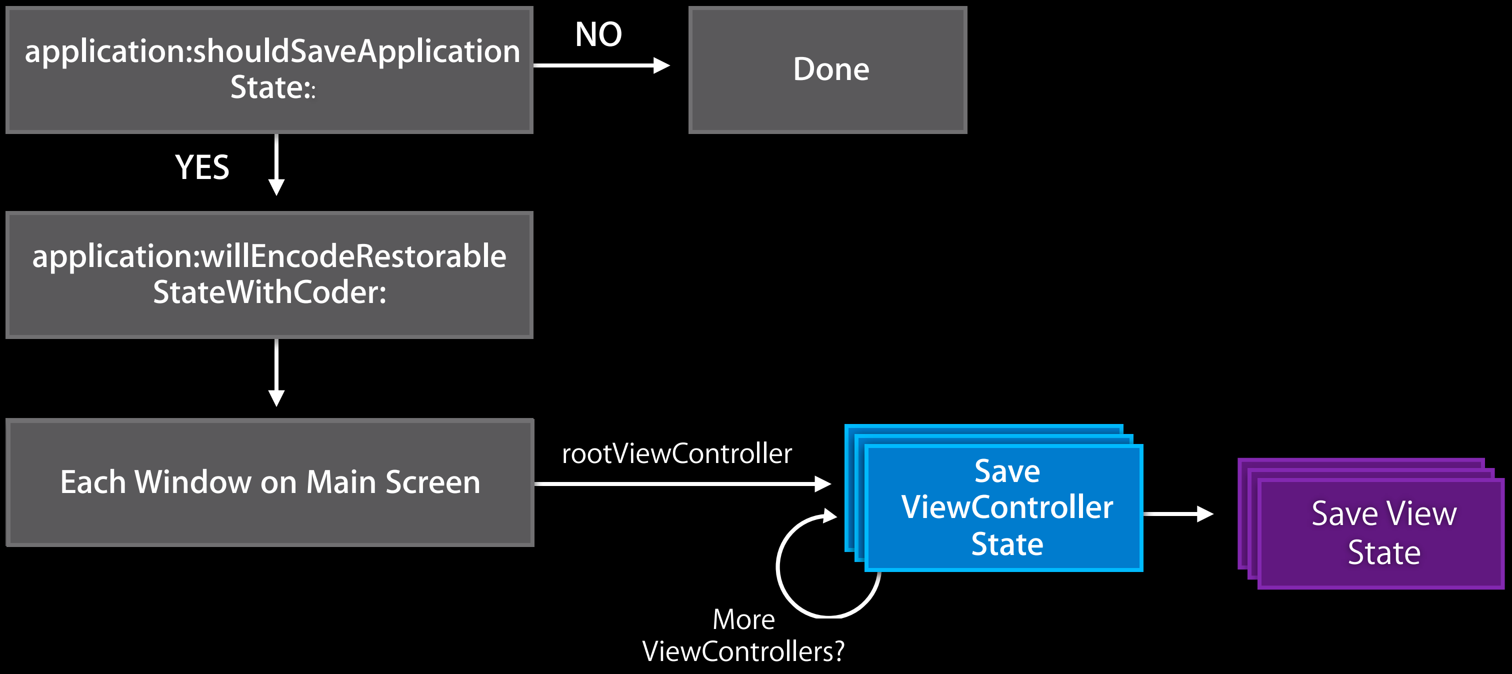
Each Window on Main Screen

rootViewController

Save ViewController State

Save View State

Saving State: Control Flow



application:shouldSaveApplicationState::

NO

Done

YES

application:willEncodeRestorableStateWithCoder:

Each Window on Main Screen

rootViewController

Save ViewController State

More ViewControllers?

Save View State

Digression:
Managing View Controllers

Finding View Controllers

Finding View Controllers

- Implicit

Finding View Controllers

- Implicit
  - Created before needed

Finding View Controllers

- Implicit
  - Created before needed
- Using restoration class

Finding View Controllers

- Implicit
  - Created before needed
- Using restoration class
  - Good encapsulation

Finding View Controllers

- Implicit
  - Created before needed
- Using restoration class
  - Good encapsulation
- Application delegate

Finding View Controllers

- Implicit
  - Created before needed
- Using restoration class
  - Good encapsulation
- Application delegate
  - Useful for root level

Finding View Controllers

- Implicit
  - Created before needed
- Using restoration class
  - Good encapsulation
- Application delegate
  - Useful for root level

Returning nil is fine

Creating View Controllers

Creating View Controllers

- Shared initialization

Creating View Controllers

- Shared initialization
    - Normal operation

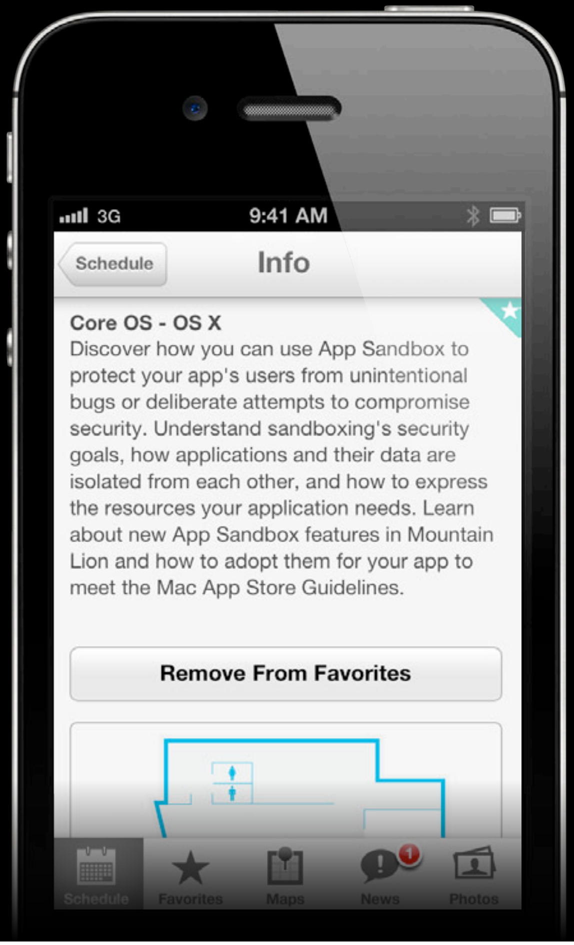
Creating View Controllers

- Shared initialization
    - Normal operation
    - State restoration

Creating View Controllers

- Shared initialization
  - Normal operation
  - State restoration
- Set restoration class

Session Details View Controller



Core OS - OS X
Discover how you can use App Sandbox to protect your app's users from unintentional bugs or deliberate attempts to compromise security. Understand sandboxing's security goals, how applications and their data are isolated from each other, and how to express the resources your application needs. Learn about new App Sandbox features in Mountain Lion and how to adopt them for your app to meet the Mac App Store Guidelines.

Session Details View Controller

Initializing session details

```
- (id) initWithSession:(WWDCSession*)session {
  self = [super init];
  if ( self != nil ) {
   self.session = session;
  }
  if ([self respondsToSelector:@selector(restorationIdentifier)]) {
   self.restorationIdentifier = @"SessionDetails";
   self.restorationClass = [self class];
  }
  return self;
}
```

Session Details View Controller

Initializing session details

```
- (id) initWithSession:(WWDCSession*)session {
  self = [super init];
  if ( self != nil ) {
   self.session = session;
  }
  if ([self respondsToSelector:@selector(restorationIdentifier)]) {
   self.restorationIdentifier = @"SessionDetails";
   self.restorationClass = [self class];
  }
  return self;
}
```

Session Details View Controller

Initializing session details

```
- (id) initWithSession:(WWDCSession*)session {
  self = [super init];
  if ( self != nil ) {
   self.session = session;
  }
  if ([self respondsToSelector:@selector(restorationIdentifier)]) {
   self.restorationIdentifier = @"SessionDetails";
   self.restorationClass = [self class];
  }
  return self;
}
```

Session Details View Controller

Loading View Controller from nib/storyboard

```
- (void) awakeFromNib {
   if ([self respondsToSelector:@selector(restorationClass)]) {
       self.restorationClass = [self class]
   }
   return self;
}
```

```
- (void) setupSession:(WWDCSession*)session {
   self.session = session;
}
```

Session Details View Controller

Loading View Controller from nib/storyboard

```
- (void) awakeFromNib {
   if ([self respondsToSelector:@selector(restorationClass)]) {
       self.restorationClass = [self class]
   }
   return self;
}
```

```
- (void) setupSession:(WWDCSession*)session {
   self.session = session;
}
```

Restoration

Restoring State

Restoring State

- Same starting point

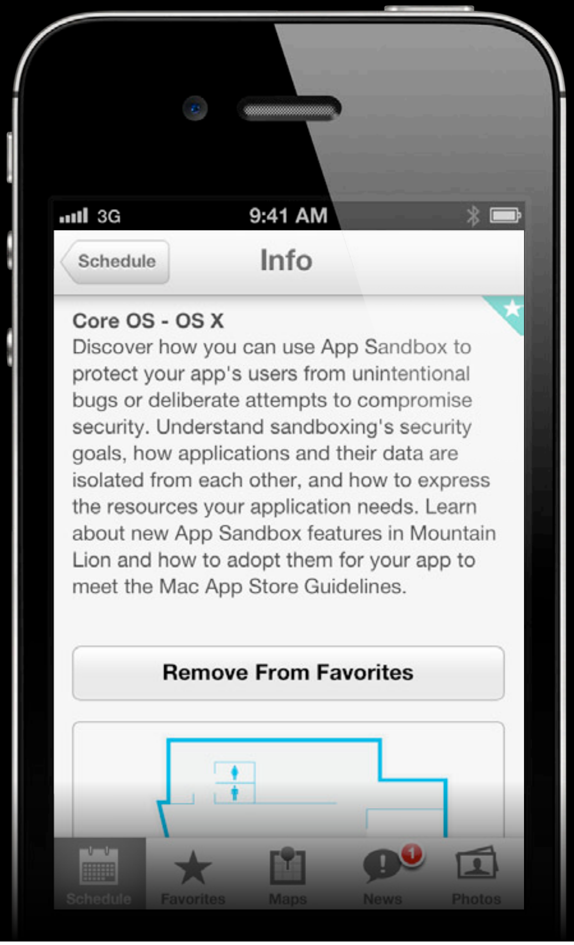
Restoring State

- Same starting point
- Find/create view controllers

Restoring State

- Same starting point
- Find/create view controllers
- Restore view controller state

Session Details View Controller



Core OS - OS X

Discover how you can use App Sandbox to protect your app's users from unintentional bugs or deliberate attempts to compromise security. Understand sandboxing's security goals, how applications and their data are isolated from each other, and how to express the resources your application needs. Learn about new App Sandbox features in Mountain Lion and how to adopt them for your app to meet the Mac App Store Guidelines.

View Controller Restoration Protocol

```
@protocol UIViewControllerRestoration
+ (UIViewController*)viewControllerWithRestorationIdentifierPath:
     (NSArray*)identifierComponents
     coder:(NSCoder*)coder
@end
```

View Controller Restoration Protocol

```
@interface MyController : UIViewController <UIViewControllerRestoration>;
```

Find/Create Session Details

```
+ (UIViewController*)viewControllerWithRestorationIdentifierPath:
     (NSArray*)identifierComponents
     coder:(NSCoder*)coder {
```

Find/Create Session Details

```
+ (UIViewController*)viewControllerWithRestorationIdentifierPath:
     (NSArray*)identifierComponents
     coder:(NSCoder*)coder {
     UIViewController *viewController = nil;
```

Find/Create Session Details

```
+ (UIViewController*)viewControllerWithRestorationIdentifierPath:
     (NSArray*)identifierComponents
     coder:(NSCoder*)coder {
     UIViewController *viewController = nil;
     NSString *sessionID = [coder decodeObjectForKey:kSessionIDKey];
```

Find/Create Session Details

```
+ (UIViewController*)viewControllerWithRestorationIdentifierPath:
     (NSArray*)identifierComponents
     coder:(NSCoder*)coder {
     UIViewController *viewController = nil;
     NSString *sessionID = [coder decodeObjectForKey:kSessionIDKey];
     if (sessionID) {
         WWDCSession *session =
               [WWDCSession sessionForSessionID:sessionID
                    context:[WWDCThreadedContextManager threadedContext]
                    error:NULL];
```

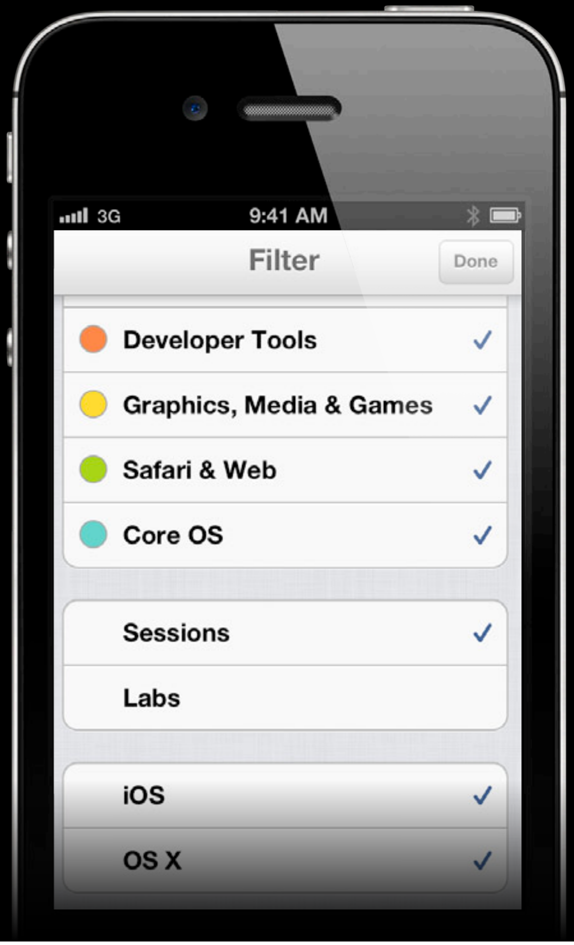
Find/Create Session Details

```
+ (UIViewController*)viewControllerWithRestorationIdentifierPath:
     (NSArray*)identifierComponents
     coder:(NSCoder*)coder {
     UIViewController *viewController = nil;
     NSString *sessionID = [coder decodeObjectForKey:kSessionIDKey];
     if (sessionID) {
         WWDCSession *session =
               [WWDCSession sessionForSessionID:sessionID
                    context:[WWDCThreadedContextManager threadedContext]
                    error:NULL];
         if (session) {
             viewController = [[self alloc] initWithSession:session];
         }
```

Find/Create Session Details

```
+ (UIViewController*)viewControllerWithRestorationIdentifierPath:
     (NSArray*)identifierComponents
     coder:(NSCoder*)coder {
     UIViewController *viewController = nil;
     NSString *sessionID = [coder decodeObjectForKey:kSessionIDKey];
     if (sessionID) {
         WWDCSession *session =
               [WWDCSession sessionForSessionID:sessionID
                    context:[WWDCThreadedContextManager threadedContext]
                    error:NULL];
         if (session) {
             viewController = [[self alloc] initWithSession:session];
         }
     }
     return viewController;
}
```

Filter View Controller



- ● Developer Tools ✓
- ● Graphics, Media & Games ✓
- ● Safari & Web ✓
- ● Core OS ✓

- Sessions ✓
- Labs

- iOS ✓
- OS X ✓

Find/Create Filter View Controller

```
+ (UIViewController*)viewControllerWithRestorationIdentifierPath:
     (NSArray*)identifierComponents
     coder:(NSCoder*)coder {
```

Find/Create Filter View Controller

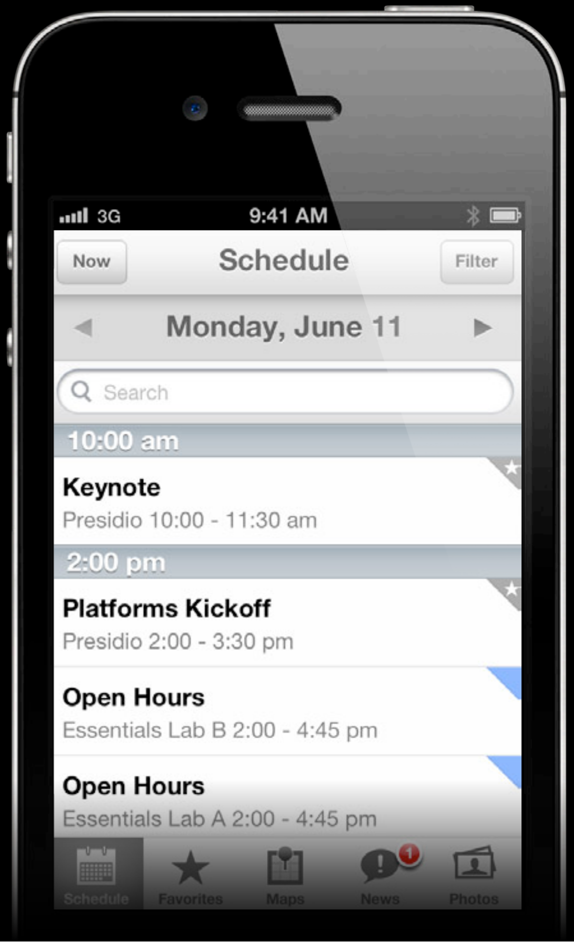
```
+ (UIViewController*)viewControllerWithRestorationIdentifierPath:
     (NSArray*)identifierComponents
     coder:(NSCoder*)coder {
     WWDCiPhoneScheduleViewController *scheduleController =
         [coder decodeObjectForKey:kScheduleViewControllerKey];
```

Find/Create Filter View Controller

```
+ (UIViewController*)viewControllerWithRestorationIdentifierPath:
     (NSArray*)identifierComponents
     coder:(NSCoder*)coder {
     WWDCiPhoneScheduleViewController *scheduleController =
         [coder decodeObjectForKey:kScheduleViewControllerKey];
     WWDCiPhoneSessionFilterViewController *controller =
         [[self alloc] initWithScheduleController:scheduleController];
     return controller;
}
```

Restoring State

Schedule View Controller



10:00 am

Keynote
Presidio 10:00 - 11:30 am

2:00 pm

Platforms Kickoff
Presidio 2:00 - 3:30 pm

Open Hours
Essentials Lab B 2:00 - 4:45 pm

Open Hours
Essentials Lab A 2:00 - 4:45 pm

Restore Schedule’s Day

```
- (void) decodeRestorableStateWithCoder:(NSCoder *)coder {
```

Restore Schedule’s Day

```
- (void) decodeRestorableStateWithCoder:(NSCoder *)coder {
   self.selectedDateIndex = [coder
         decodeIntForKey:kScheduleViewSelectedDateIndexKey];
```

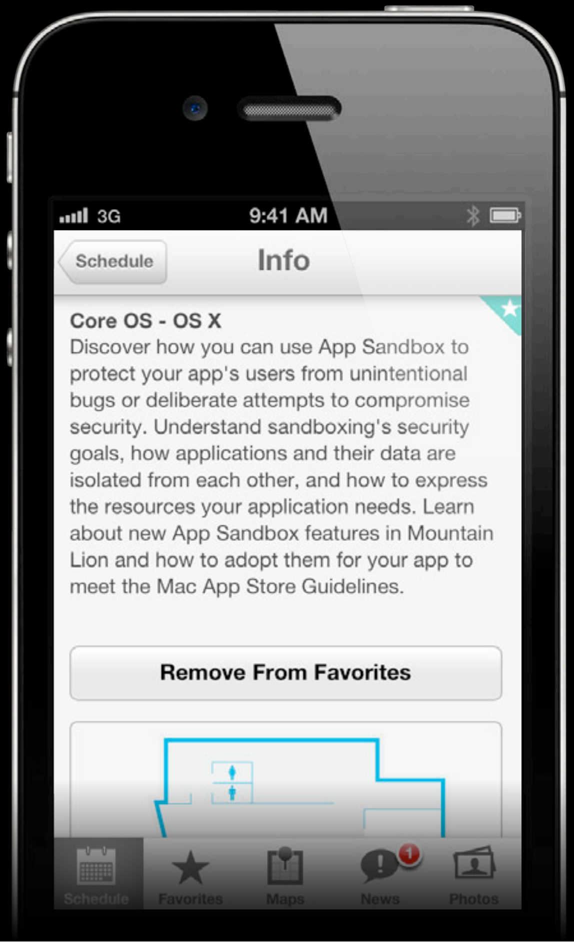
Restore Schedule’s Day

```
- (void) decodeRestorableStateWithCoder:(NSCoder *)coder {
  self.selectedDateIndex = [coder
     decodeIntForKey:kScheduleViewSelectedDateIndexKey];
  [self update];
```

Restore Schedule’s Day

```
- (void) decodeRestorableStateWithCoder:(NSCoder *)coder {
  self.selectedDateIndex = [coder
     decodeIntForKey:kScheduleViewSelectedDateIndexKey];
  [self update];
  [super decodeRestorableStateWithCoder:coder];
}
```

Session Details View Controller



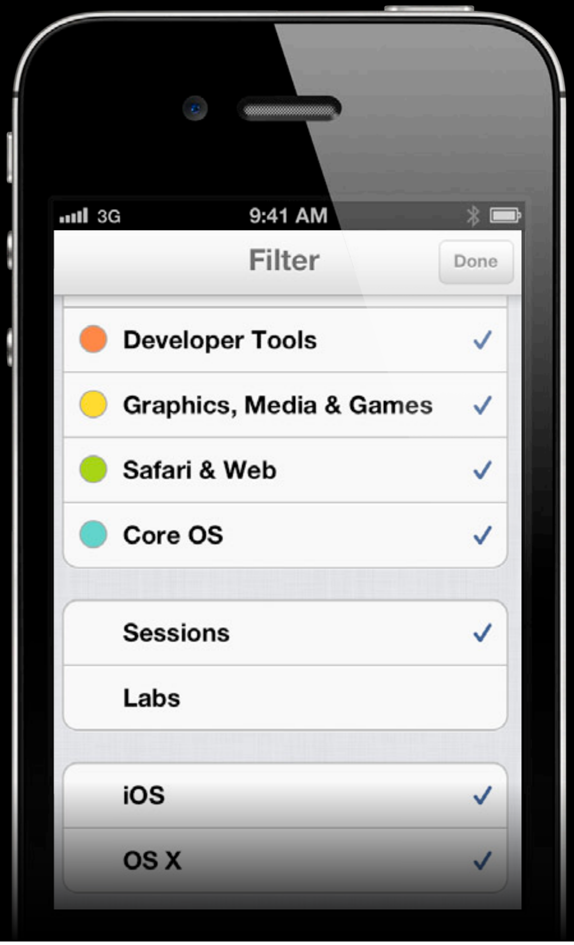
Core OS - OS X

Discover how you can use App Sandbox to protect your app's users from unintentional bugs or deliberate attempts to compromise security. Understand sandboxing's security goals, how applications and their data are isolated from each other, and how to express the resources your application needs. Learn about new App Sandbox features in Mountain Lion and how to adopt them for your app to meet the Mac App Store Guidelines.

Restore Session State

```
- (void) decodeRestorableStateWithCoder:(NSCoder *)coder {
   self.launchedFromTheScheduleView =
      [coder decodeBoolForKey:kLaunchedFromScheduleViewKey];
   [super decodeRestorableStateWithCoder:coder];
}
```

Filter View Controller



Filter

Done

- ● Developer Tools ✓
- ● Graphics, Media & Games ✓
- ● Safari & Web ✓
- ● Core OS ✓

- Sessions ✓
- Labs

- iOS ✓
- OS X ✓

Restoring Filter

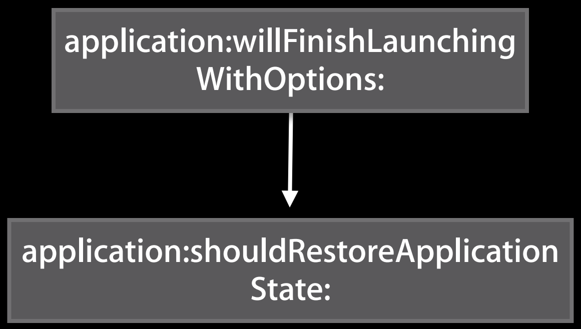
Nothing needs to be done!

```
- (void) decodeRestorableStateWithCoder:(NSCoder *)coder {
     [super decodeRestorableStateWithCoder:coder];
}
```

App Startup

application:willFinishLaunching
WithOptions:

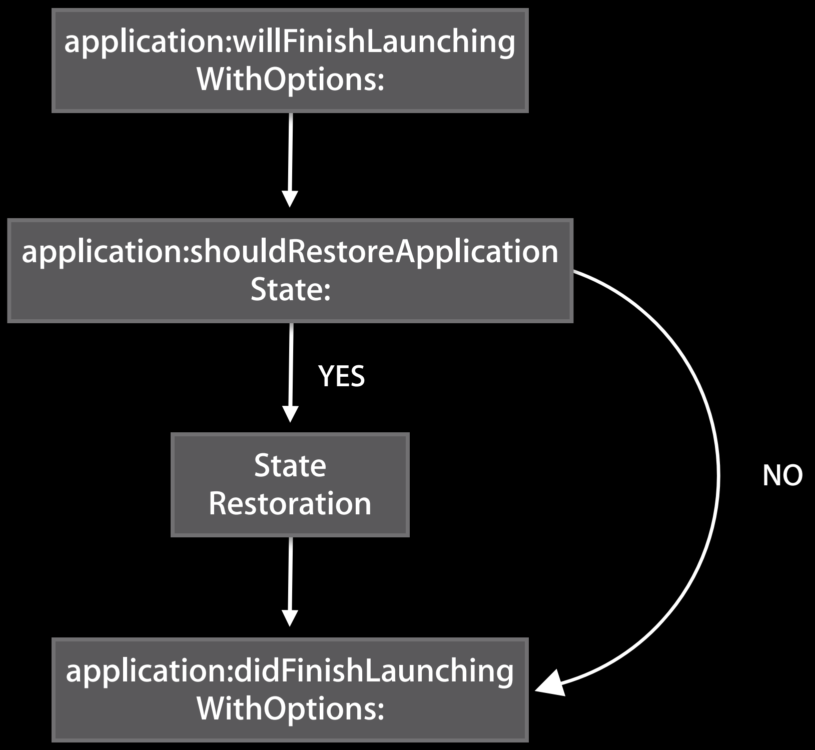
App Startup



application:willFinishLaunchingWithOptions:

application:shouldRestoreApplicationState:

App Startup



application:willFinishLaunching WithOptions:

application:shouldRestoreApplication State:

YES

State Restoration

NO

application:didFinishLaunching WithOptions:

application:willFinishLaunchingWithOptions:

application:willFinishLaunchingWithOptions:

- New method called before state restoration

application:willFinishLaunchingWithOptions:

- New method called before state restoration
- Move initialization code there

application:willFinishLaunchingWithOptions:

- New method called before state restoration
- Move initialization code there

Still use `application:didFinishLaunchingWithOptions:`

application:willFinishLaunchingWithOptions:

- New method called before state restoration
- Move initialization code there

Still use `application:didFinishLaunchingWithOptions:`

  - Example: login, password

application:willFinishLaunchingWithOptions:

- New method called before state restoration
- Move initialization code there

Still use `application:didFinishLaunchingWithOptions:`

  - Example: login, password

- Deploy for iOS 6 and prior versions

Application Launch

```
- (void) commonLaunchInitialization:(NSDictionary *)launchOptions {
    static dispatch_once_t pred;
    dispatch_once(&pred, ^{
        // Perform common setup...
        [self.window makeKeyAndVisible];
    });
}
```

Application Launch

```
- (void) commonLaunchInitialization:(NSDictionary *)launchOptions {
    static dispatch_once_t pred;
    dispatch_once(&pred, ^{
        // Perform common setup...
        [self.window makeKeyAndVisible];
    });
}
```

Application Launch

```
- (BOOL)application:(UIApplication *)application
willFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    [self commonLaunchInitialization:launchOptions];
    return YES;
}
```

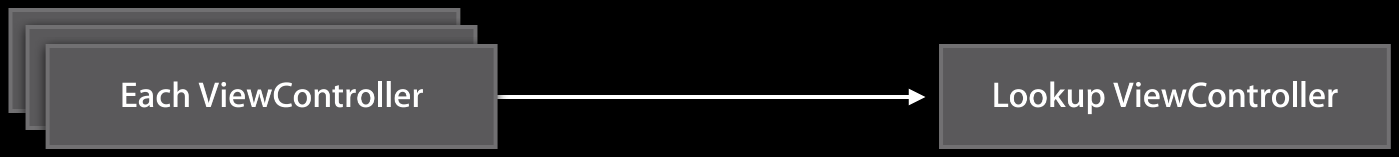
```
- (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    [self commonLaunchInitialization:launchOptions];
    // Other stuff
    return YES;
}
```

Application Launch

```
- (BOOL)application:(UIApplication *)application
willFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    [self commonLaunchInitialization:launchOptions];
    return YES;
}
```

```
- (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    [self commonLaunchInitialization:launchOptions];
    // Other stuff
    return YES;
}
```

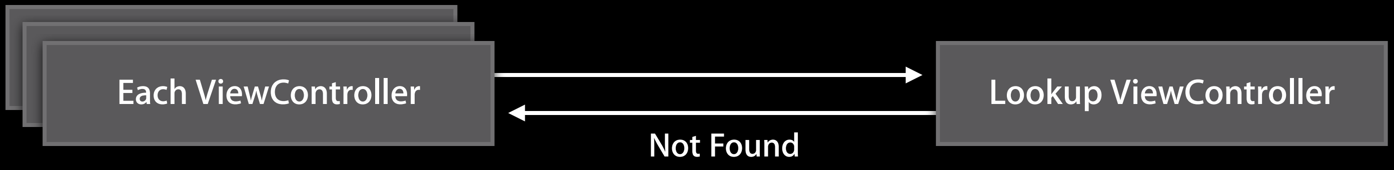
Restoration



Each ViewController

Lookup ViewController

Restoration

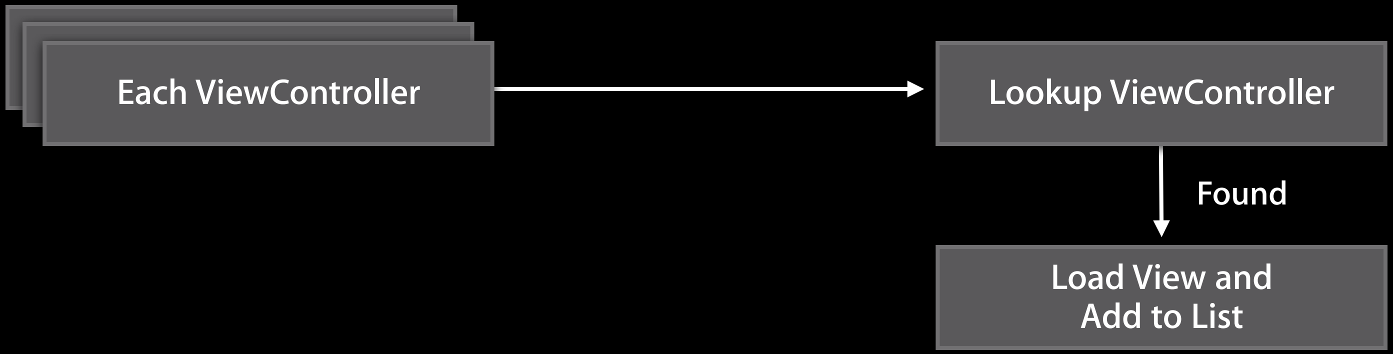


Each ViewController

Lookup ViewController

Not Found

Restoration



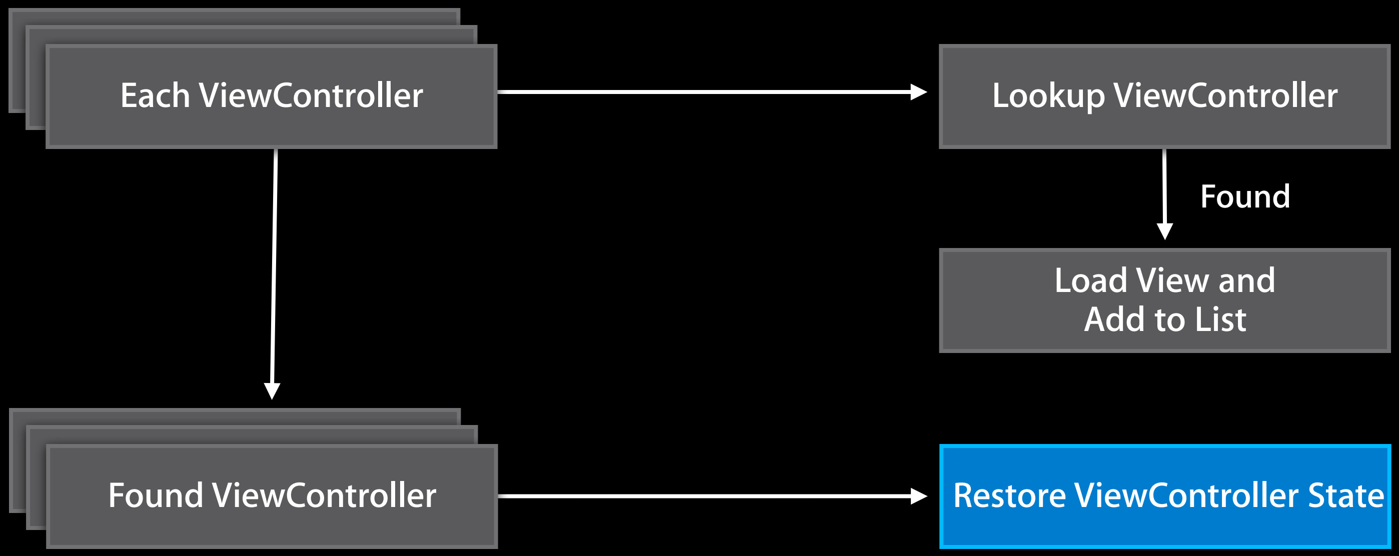
Each ViewController

Lookup ViewController

Found

Load View and Add to List

Restoration



Each ViewController

Lookup ViewController

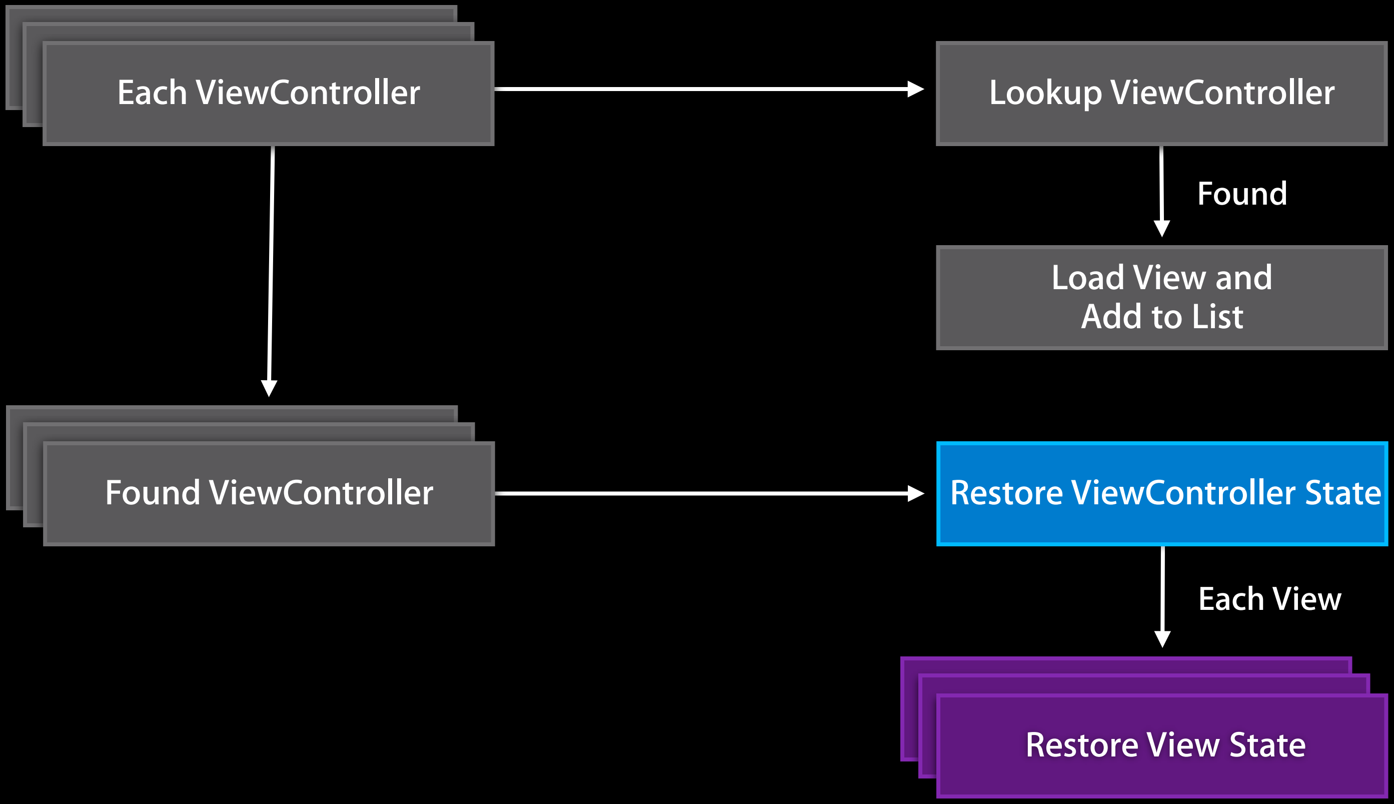
Found

Load View and
Add to List

Found ViewController

Restore ViewController State

Restoration



Each ViewController

Lookup ViewController

Found

Load View and
Add to List

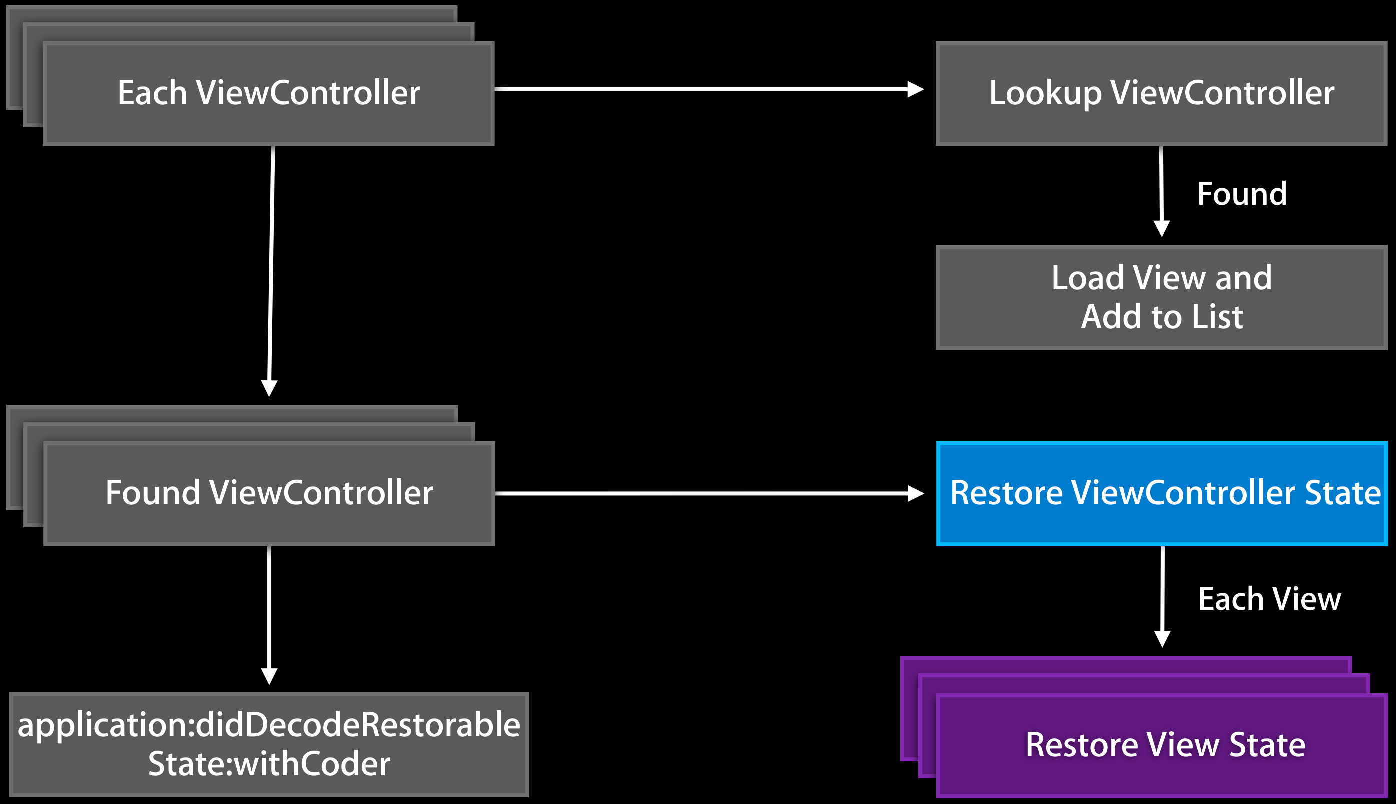
Found ViewController

Restore ViewController State

Each View

Restore View State

Restoration



Each ViewController

Lookup ViewController

Found

Load View and
Add to List

Found ViewController

Restore ViewController State

Each View

application:didDecodeRestorable
State:withCoder

Restore View State

Demo

Text, Search, Mapping Cells

What Do You Have to Do?

What Do You Have to Do?

- Save search and feedback state

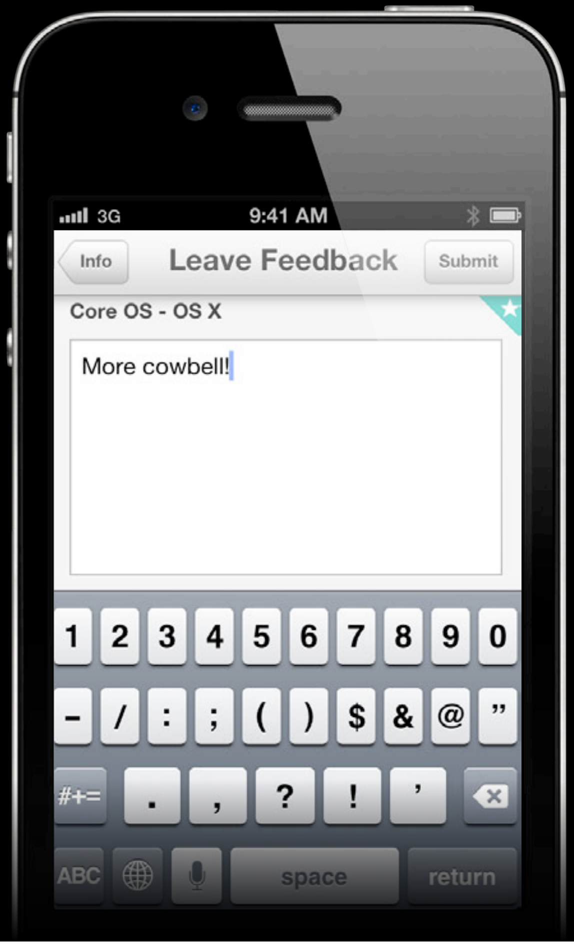
What Do You Have to Do?

- Save search and feedback state
- Find/create feedback controller

What Do You Have to Do?

- Save search and feedback state
- Find/create feedback controller
- Restore search and feedback state

Feedback View Controller



Core OS - OS X

More cowbell!

Saving Feedback State

```
- (void) encodeRestorableStateWithCoder:(NSCoder *)coder {
```

Saving Feedback State

```
- (void) encodeRestorableStateWithCoder:(NSCoder *)coder {
    [coder encodeObject:self.session.sessionID
            forKey:kFeedbackControllerSessionIDKey];
```

Saving Feedback State

```
- (void) encodeRestorableStateWithCoder:(NSCoder *)coder {
    [coder encodeObject:self.session.sessionID
            forKey:kFeedbackControllerSessionIDKey];
    [coder encodeObject:self.delegate
           forKey:kFeedbackControllerDelegateKey];
```

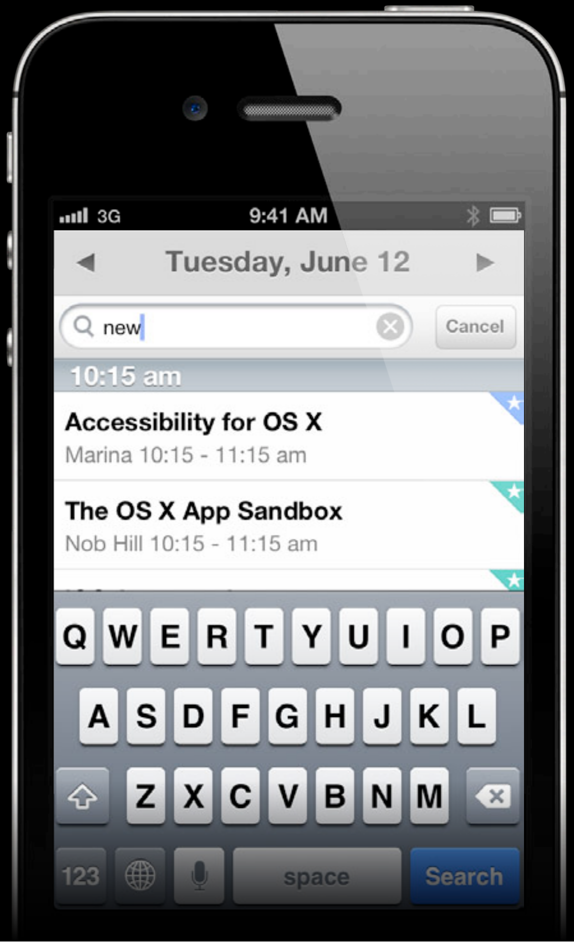
Saving Feedback State

```
- (void) encodeRestorableStateWithCoder:(NSCoder *)coder {
    [coder encodeObject:self.session.sessionID
            forKey:kFeedbackControllerSessionIDKey];
    [coder encodeObject:self.delegate
           forKey:kFeedbackControllerDelegateKey];
    NSString *comment = self.commentTextView.text;
```

Saving Feedback State

```
- (void) encodeRestorableStateWithCoder:(NSCoder *)coder {
    [coder encodeObject:self.session.sessionID
            forKey:kFeedbackControllerSessionIDKey];
    [coder encodeObject:self.delegate
           forKey:kFeedbackControllerDelegateKey];
    NSString *comment = self.commentTextView.text;
    if (comment && ![comment isEqualToString:@""]) {
        [coder encodeObject:comment forKey:kFeedbackControllerTextKey];
    }
    [super encodeRestorableStateWithCoder:coder];
}
```

Schedule—Search



new

10:15 am

Accessibility for OS X
Marina 10:15 - 11:15 am

The OS X App Sandbox
Nob Hill 10:15 - 11:15 am

Saving Search State

```
– (void) encodeRestorableStateWithCoder:(NSCoder *)coder {
```

Saving Search State

```
- (void) encodeRestorableStateWithCoder:(NSCoder *)coder {
    // Save search bar active/editing state
```

Saving Search State

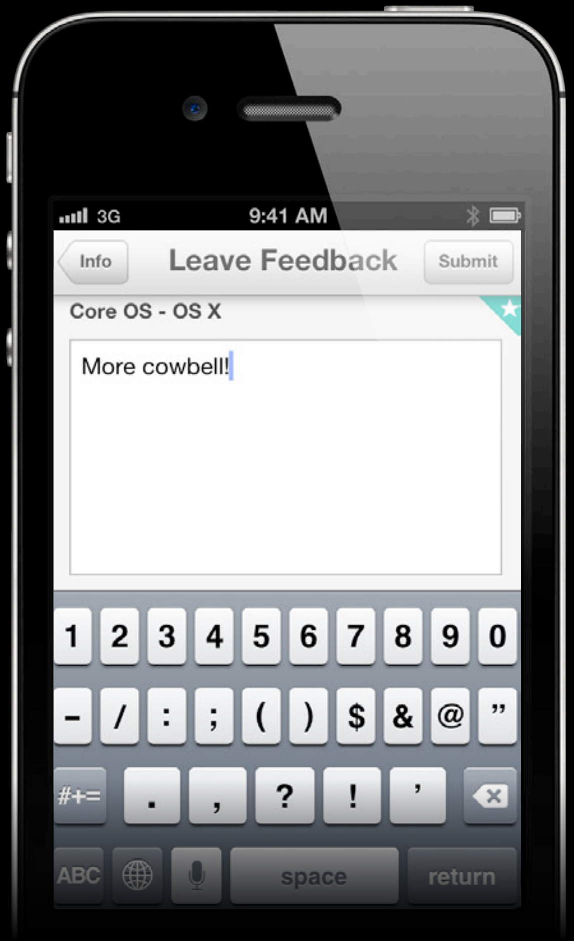
```
- (void) encodeRestorableStateWithCoder:(NSCoder *)coder {
    // Save search bar active/editing state
    // Save search bar text
```

Saving Search State

```
- (void) encodeRestorableStateWithCoder:(NSCoder *)coder {
   // Save search bar active/editing state
   // Save search bar text
   // Save current day, call super
}
```

Restoration

Feedback View Controller



Core OS - OS X

More cowbell!

Finding Feedback Controller

```
+ (UIViewController *) viewControllerWithRestorationIdentifierPath:
  (NSArray *)identifierComponents coder:(NSCoder *)coder {
```

Finding Feedback Controller

```
+ (UIViewController *) viewControllerWithRestorationIdentifierPath:
  (NSArray *)identifierComponents coder:(NSCoder *)coder {

    UIViewController *viewController = nil;
```

Finding Feedback Controller

```
+ (UIViewController *) viewControllerWithRestorationIdentifierPath:
  (NSArray *)identifierComponents coder:(NSCoder *)coder {
    UIViewController *viewController = nil;
    NSString *sessionID =
        [coder decodeObjectForKey:kFeedbackControllerSessionIDKey];
    if (sessionID) {
```

Finding Feedback Controller

```
+ (UIViewController *) viewControllerWithRestorationIdentifierPath:
  (NSArray *)identifierComponents coder:(NSCoder *)coder {
    UIViewController *viewController = nil;
    NSString *sessionID =
        [coder decodeObjectForKey:kFeedbackControllerSessionIDKey];
    if (sessionID) {
        WWDCSession *session = // Find session;
        if (session) {
            viewController = [[self alloc] initWithSession:session];
```

Finding Feedback Controller

```
+ (UIViewController *) viewControllerWithRestorationIdentifierPath:
  (NSArray *)identifierComponents coder:(NSCoder *)coder {
    UIViewController *viewController = nil;
    NSString *sessionID =
        [coder decodeObjectForKey:kFeedbackControllerSessionIDKey];
    if (sessionID) {
        WWDCSession *session = // Find session;
        if (session) {
            viewController = [[self alloc] initWithSession:session];
            viewController.delegate =
                [coder decodeObjectForKey:kFeedbackControllerDelegateKey];
        }
```

Finding Feedback Controller

```
+ (UIViewController *) viewControllerWithRestorationIdentifierPath:
  (NSArray *)identifierComponents coder:(NSCoder *)coder {
    UIViewController *viewController = nil;
    NSString *sessionID =
        [coder decodeObjectForKey:kFeedbackControllerSessionIDKey];
    if (sessionID) {
        WWDCSession *session = // Find session;
        if (session) {
            viewController = [[self alloc] initWithSession:session];
            viewController.delegate =
                [coder decodeObjectForKey:kFeedbackControllerDelegateKey];
        }
    }
    return viewController;
}
```

Restoring Feedback State

```
- (void) decodeRestorableStateWithCoder:(NSCoder *)coder {
```

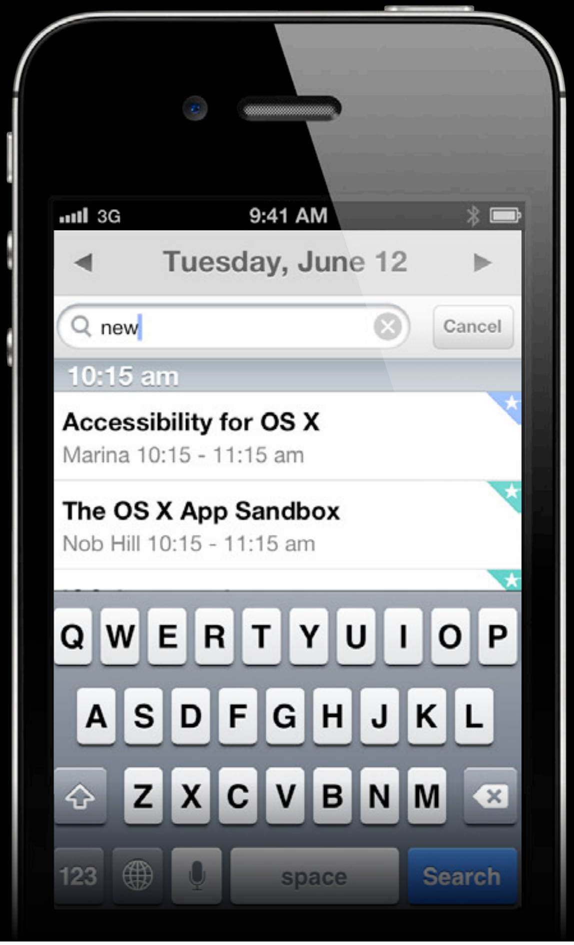
Restoring Feedback State

```
- (void) decodeRestorableStateWithCoder:(NSCoder *)coder {
     NSString *comment =
         [coder decodeObjectForKey:kFeedbackControllerTextKey];
```

Restoring Feedback State

```
- (void) decodeRestorableStateWithCoder:(NSCoder *)coder {
    NSString *comment =
        [coder decodeObjectForKey:kFeedbackControllerTextKey];
    if (comment) {
        self.commentTextView.text = comment;
    }
    [super decodeRestorableStateWithCoder:coder];
}
```

Schedule—Search



10:15 am

Accessibility for OS X
Marina 10:15 - 11:15 am

The OS X App Sandbox
Nob Hill 10:15 - 11:15 am

Restoring Search State

```
- (void) decodeRestorableStateWithCoder:(NSCoder *)coder {
```

Restoring Search State

```
- (void) decodeRestorableStateWithCoder:(NSCoder *)coder {
    // Restore search bar active/editing state
```

Restoring Search State

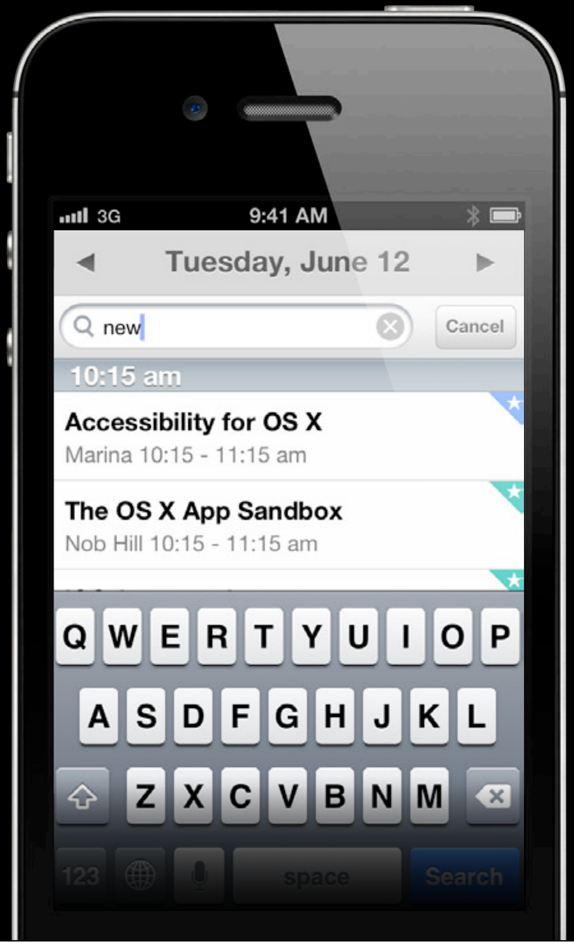
```
- (void) decodeRestorableStateWithCoder:(NSCoder *)coder {
    // Restore search bar active/editing state
    // Restore search bar text
```

Restoring Search State

```
- (void) decodeRestorableStateWithCoder:(NSCoder *)coder {
   // Restore search bar active/editing state
   // Restore search bar text
   // Restore current day, call super
}
```

Applying Search State

Search Behind Details/Feedback



Tuesday, June 12

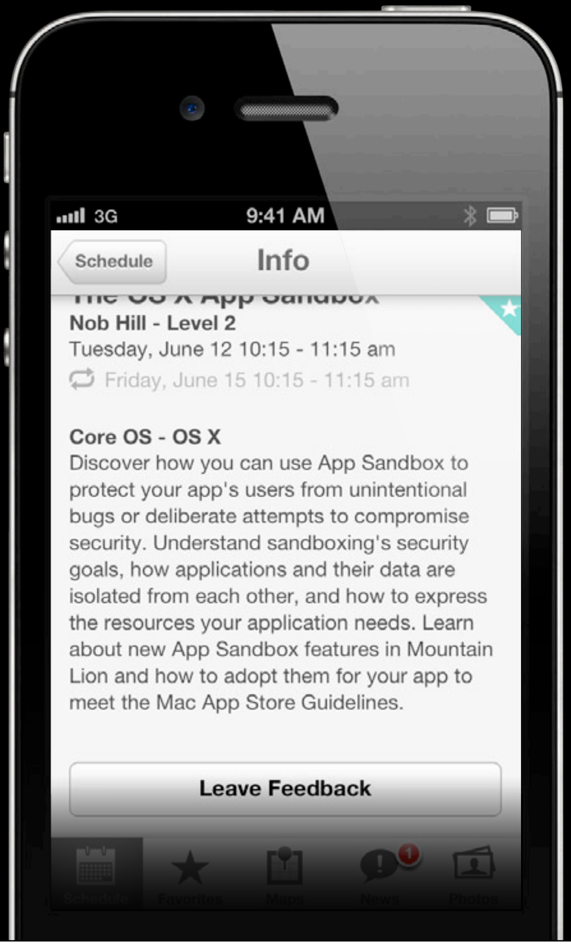
new

10:15 am

Accessibility for OS X
Marina 10:15 - 11:15 am

The OS X App Sandbox
Nob Hill 10:15 - 11:15 am

Search Behind Details/Feedback



The OS X App Sandbox

Nob Hill - Level 2

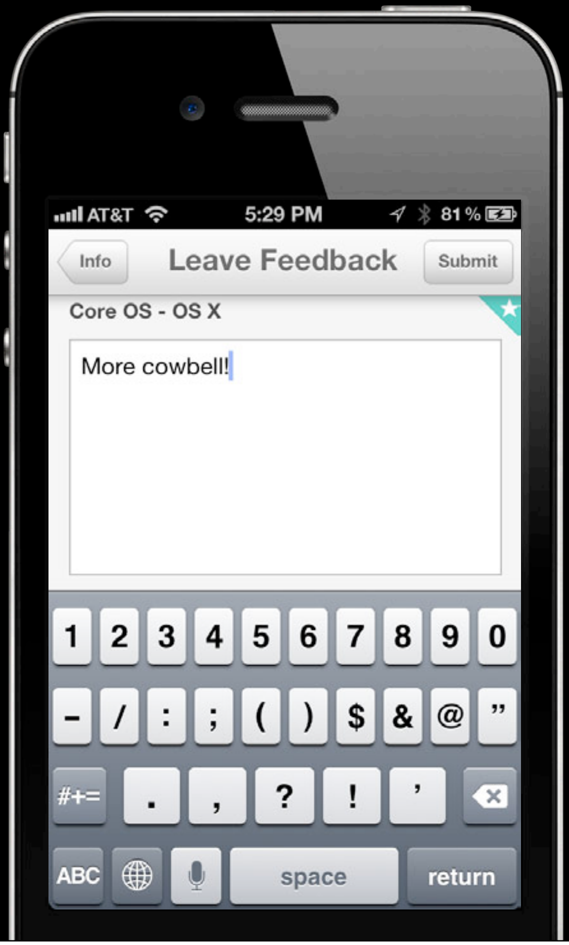
Tuesday, June 12 10:15 - 11:15 am

Friday, June 15 10:15 - 11:15 am

Core OS - OS X

Discover how you can use App Sandbox to protect your app's users from unintentional bugs or deliberate attempts to compromise security. Understand sandboxing's security goals, how applications and their data are isolated from each other, and how to express the resources your application needs. Learn about new App Sandbox features in Mountain Lion and how to adopt them for your app to meet the Mac App Store Guidelines.

Search Behind Details/Feedback



Core OS - OS X

More cowbell!

Restoring Search State

```
- (void) decodeRestorableStateWithCoder:(NSCoder *)coder {
    // Decode search state into instance variables
    // ...
}
```

Applying Search State

```
- (void)viewWillAppear:(BOOL)animated {
```

Applying Search State

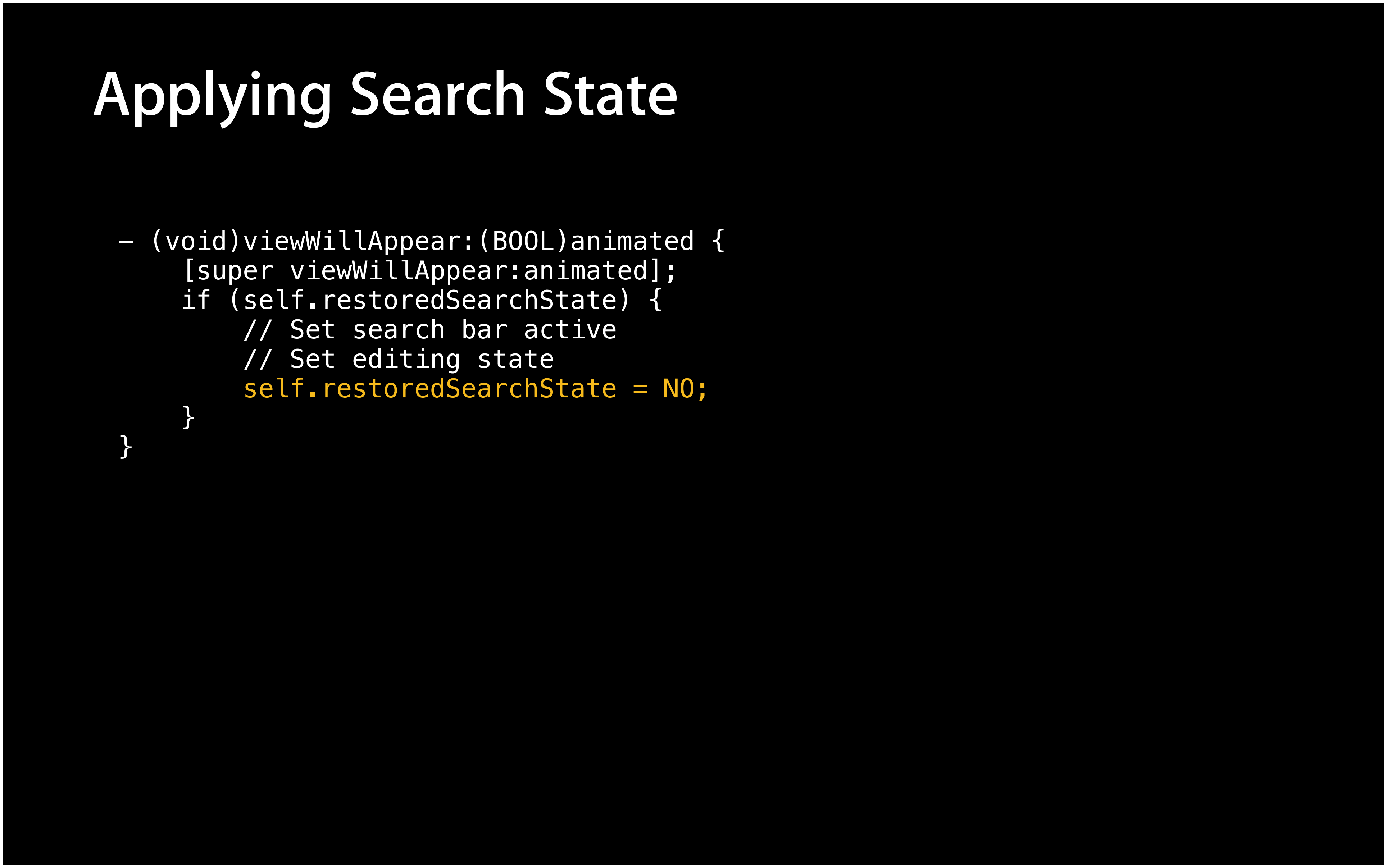
```
- (void)viewWillAppear:(BOOL)animated {
    [super viewWillAppear:animated];
```

Applying Search State

```
- (void)viewWillAppear:(BOOL)animated {
    [super viewWillAppear:animated];
    if (self.restoredSearchState) {
        // Set search bar active
        // Set editing state
        self.restoredSearchState = NO;
    }
}
```

Applying Search State

```
- (void)viewWillAppear:(BOOL)animated {
    [super viewWillAppear:animated];
    if (self.restoredSearchState) {
        // Set search bar active
        // Set editing state
        self.restoredSearchState = NO;
    }
}
```



Applying Search State

```
- (void)viewWillAppear:(BOOL)animated {
    [super viewWillAppear:animated];
    if (self.restoredSearchState) {
        // Set search bar active
        // Set editing state
        self.restoredSearchState = NO;
    }
}
```

One More Small Twist

Saving Search State

```
- (void) encodeRestorableStateWithCoder:(NSCoder *)coder {
```

Saving Search State

```
- (void) encodeRestorableStateWithCoder:(NSCoder *)coder {

    if (self.savedSearchTerm) {
        // We have unapplied restored state
    }
```

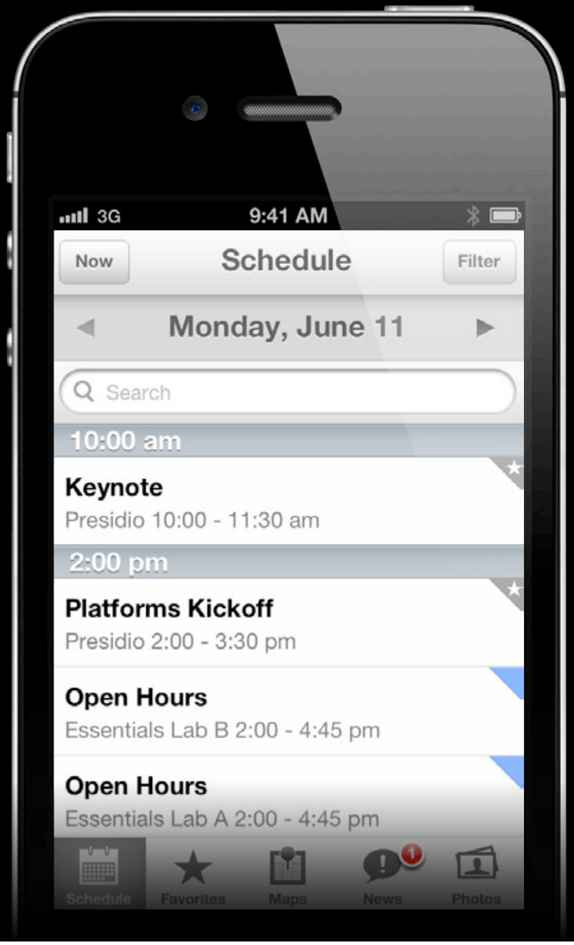
Saving Search State

```
– (void) encodeRestorableStateWithCoder:(NSCoder *)coder {

    if (self.savedSearchTerm) {
        // We have unapplied restored state
    }
    else {
        // Use current state
    }
    [super encodeRestorableStateWithCoder:coder];
}
```

Mapping Cells

Table View Cells



10:00 am

Keynote
Presidio 10:00 - 11:30 am

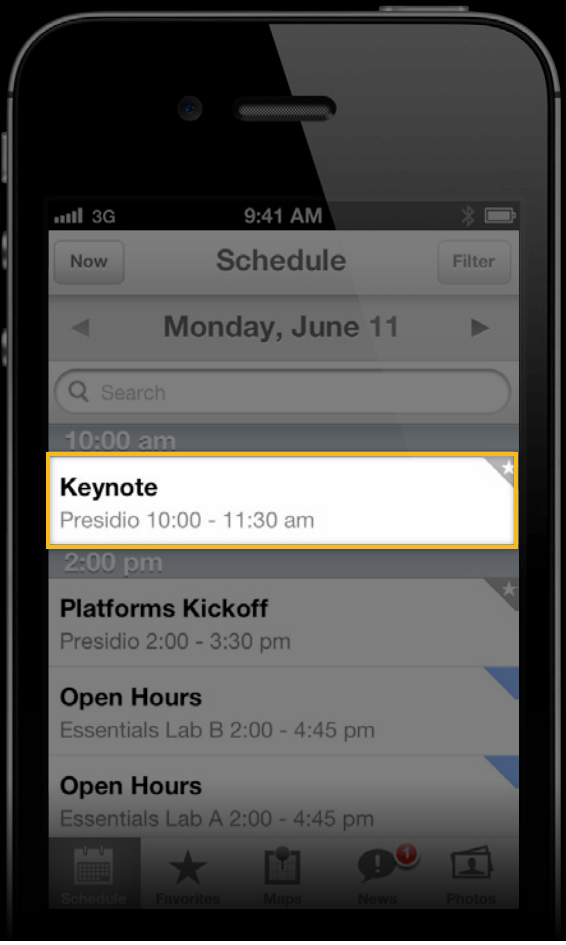
2:00 pm

Platforms Kickoff
Presidio 2:00 - 3:30 pm

Open Hours
Essentials Lab B 2:00 - 4:45 pm

Open Hours
Essentials Lab A 2:00 - 4:45 pm

Table View Cells



10:00 am

Keynote
Presidio 10:00 - 11:30 am

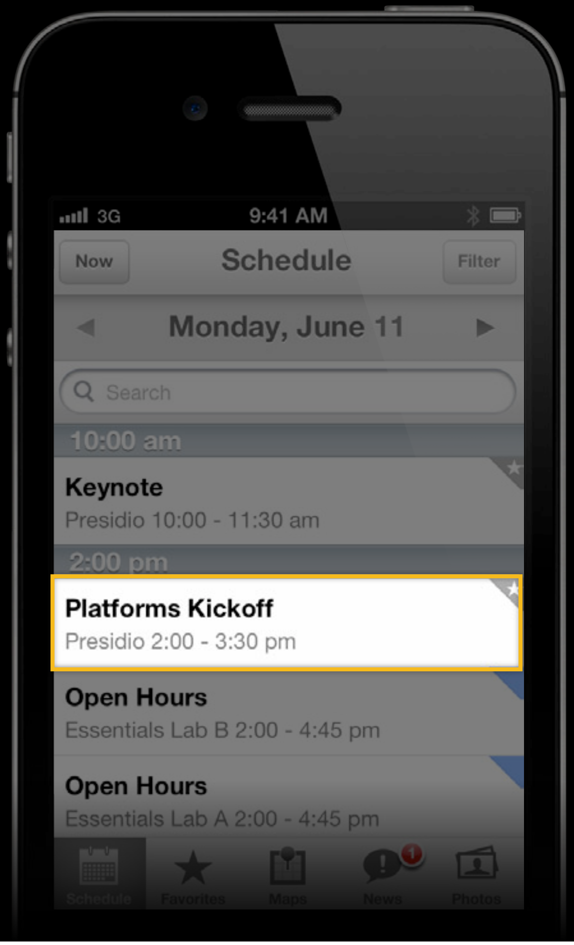
2:00 pm

Platforms Kickoff
Presidio 2:00 - 3:30 pm

Open Hours
Essentials Lab B 2:00 - 4:45 pm

Open Hours
Essentials Lab A 2:00 - 4:45 pm

Table View Cells



10:00 am

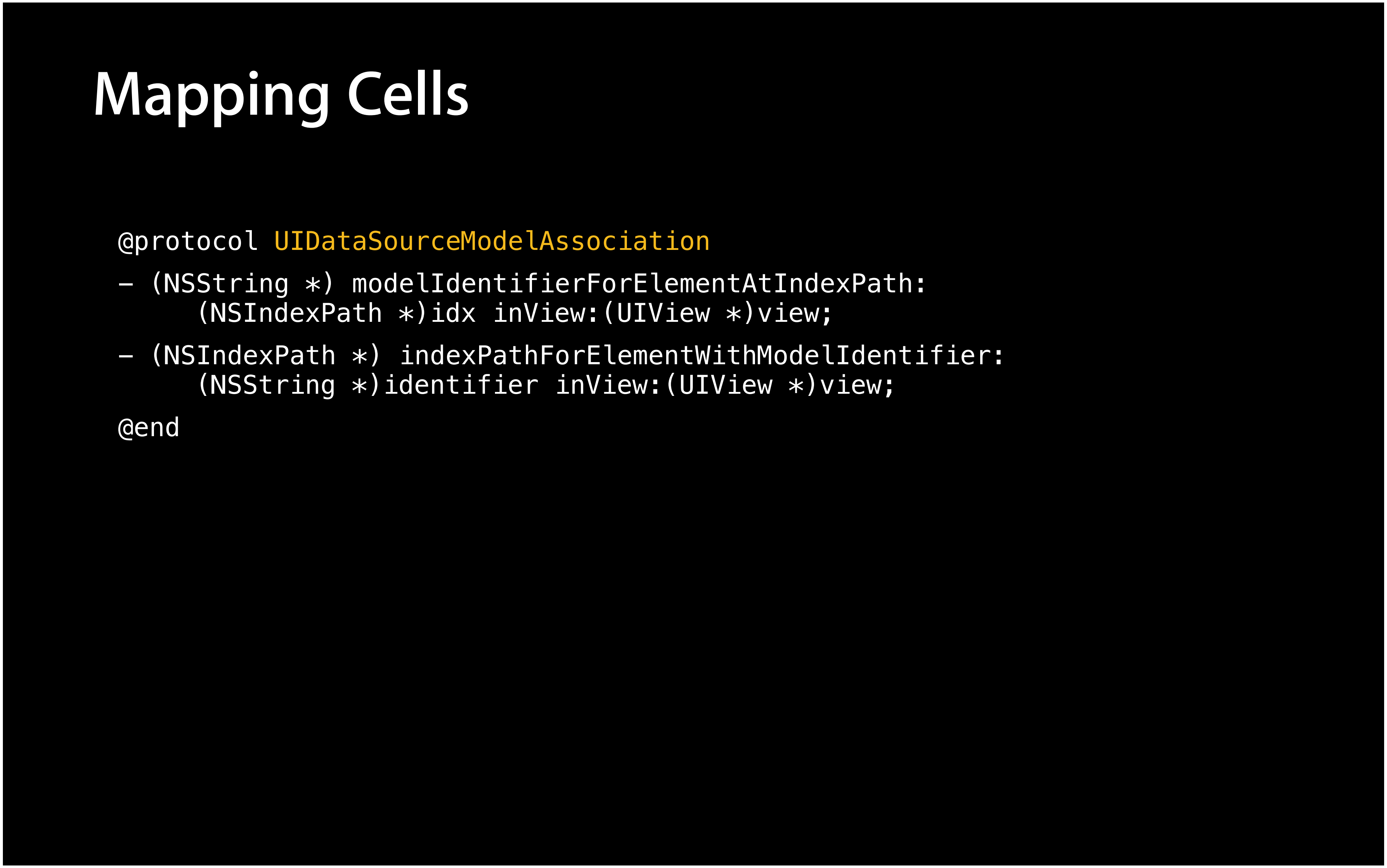
Keynote
Presidio 10:00 - 11:30 am

2:00 pm

Platforms Kickoff
Presidio 2:00 - 3:30 pm

Open Hours
Essentials Lab B 2:00 - 4:45 pm

Open Hours
Essentials Lab A 2:00 - 4:45 pm



Mapping Cells

```
@protocol UIDataSourceModelAssociation
- (NSString *) modelIdentifierForElementAtIndexPath:
     (NSIndexPath *)idx inView:(UIView *)view;
- (NSIndexPath *) indexPathForElementWithModelIdentifier:
     (NSString *)identifier inView:(UIView *)view;
@end
```

Mapping Cells

```
@protocol UIDataSourceModelAssociation
- (NSString *) modelIdentifierForElementAtIndexPath:
     (NSIndexPath *)idx inView:(UIView *)view;
- (NSIndexPath *) indexPathForElementWithModelIdentifier:
     (NSString *)identifier inView:(UIView *)view;
@end
```

Mapping Cells

```
@protocol UIDataSourceModelAssociation
- (NSString *) modelIdentifierForElementAtIndexPath:
     (NSIndexPath *)idx inView:(UIView *)view;
- (NSIndexPath *) indexPathForElementWithModelIdentifier:
     (NSString *)identifier inView:(UIView *)view;
@end
```

Mapping Cells

```
@interface MyDataSource : NSObject
    <UITableViewDataSource,
    NSFetchedResultsControllerDelegate,
    UIDataSourceModelAssociation>;
```

Saving Cell Mapping

Called when saving TableView state

```
– (NSString *) modelIdentifierForElementAtIndexPath:
  (NSIndexPath *)path inView:(UIView *)view {
```

Saving Cell Mapping

```
- (NSString *) modelIdentifierForElementAtIndexPath:
  (NSIndexPath *)path inView:(UIView *)view {
    if (path && view) {
        WWDCSession *session = [self sessionAtIndexPath:path];
```

Saving Cell Mapping

```
- (NSString *) modelIdentifierForElementAtIndexPath:
  (NSIndexPath *)path inView:(UIView *)view {
    if (path && view) {
        WWDCSession *session = [self sessionAtIndexPath:path];
        if (session) {
            return session.sessionID;
        }
```

Saving Cell Mapping

```
- (NSString *) modelIdentifierForElementAtIndexPath:
  (NSIndexPath *)path inView:(UIView *)view {
    if (path && view) {
        WWDCSession *session = [self sessionAtIndexPath:path];
        if (session) {
            return session.sessionID;
        }
    }
    return nil;
}
```

Restoring Cell Mapping

Called when restoring TableView state

```
- (NSIndexPath *) indexPathForElementWithModelIdentifier:
(NSString *)identifier inView:(UIView *)view {
```

Restoring Cell Mapping

```
- (NSIndexPath *) indexPathForElementWithModelIdentifier:
(NSString *)identifier inView:(UIView *)view {
    if (identifier && view) {
        id session =
            [WWDCSession sessionForSessionID:identifier
            context:self.managedObjectContext error:nil];
```

Restoring Cell Mapping

```
- (NSIndexPath *) indexPathForElementWithModelIdentifier:
(NSString *)identifier inView:(UIView *)view {
    if (identifier && view) {
        id session =
            [WWDCSession sessionForSessionID:identifier
            context:self.managedObjectContext error:nil];
        if (session) {
            NSIndexPath *path = [controller indexPathForObject:object];
            return path;
        }
```

Restoring Cell Mapping

```
- (NSIndexPath *) indexPathForElementWithModelIdentifier:
(NSString *)identifier inView:(UIView *)view {
    if (identifier && view) {
        id session =
            [WWDCSession sessionForSessionID:identifier
            context:self.managedObjectContext error:nil];
        if (session) {
            NSIndexPath *path = [controller indexPathForObject:object];
            return path;
        }
    }
    return nil;
}
```

Additional Insights

Default View Controller Behaviors

- Selected tab
- Navigation stack
- Presented view controller

Default View Behaviors

Default View Behaviors

- Scroll position

Default View Behaviors

- Scroll position
- Selected table view cell

Default View Behaviors

- Scroll position
- Selected table view cell
- Image transform

Default View Behaviors

- Scroll position
- Selected table view cell
- Image transform
- Web history

Preventing Default Behaviors

How to override

Preventing Default Behaviors

How to override

- No restoration identifier

Preventing Default Behaviors

How to override

- No restoration identifier
- Don’t call super

Magic Keys

Magic Keys

- Available in all coders

Magic Keys

- Available in all coders

```
UIApplicationStateRestorationBundleVersionKey
```

Magic Keys

- Available in all coders

```
UIApplicationStateRestorationBundleVersionKey
UIApplicationStateRestorationUserInterfaceIdiomKey
```

Magic Keys

- Available in all coders

```
UIApplicationStateRestorationBundleVersionKey
UIApplicationStateRestorationUserInterfaceIdiomKey
```

- Available for view controller coders

Magic Keys

- Available in all coders

```
UIApplicationStateRestorationBundleVersionKey
UIApplicationStateRestorationUserInterfaceIdiomKey
```

- Available for view controller coders

```
UIStateRestorationViewControllerStoryboardKey
```

Testing/Diagnosis

You are not alone

Testing/Diagnosis

You are not alone

- Logging/debugging

Testing/Diagnosis

You are not alone

- Logging/debugging
- Restrospector

Testing/Diagnosis

You are not alone

- Logging/debugging
- Restrospector
- Cleaning up after failure

Testing/Diagnosis

You are not alone

- Logging/debugging
- Restrospector
- Cleaning up after failure
- Methods to extend/complete

Summary

- Provides a better user experience
- Spend your time wisely
- Structure your code for state restoration
- Phase in support

More Information

Jake Behrens

UI Frameworks Evangelist
behrens@apple.com

Documentation

iOS App Programming Guide
http://developer.apple.com/

Apple Developer Forums

http://devforums.apple.com

Labs

Application State Restoration on iOS Lab	Essentials Lab A Wednesday 4:30PM
General UIKit Labs	Ongoing

WWDC2012