

# Window Management in Your Multitasking App

Activate, refresh, and destroy windows

# Programmatic Session Management

On behalf of your users

Activate

Refresh

Destroy

***Demo***

# Programmatic Session Management

On behalf of your users

Activate

Refresh

Destroy

# Programmatic Session Management

On behalf of your users

Activate

Refresh

Destroy

Activate a session only in response to  
direct and local user interaction

# Session Activation

NEW

New method on `UIApplication` to activate

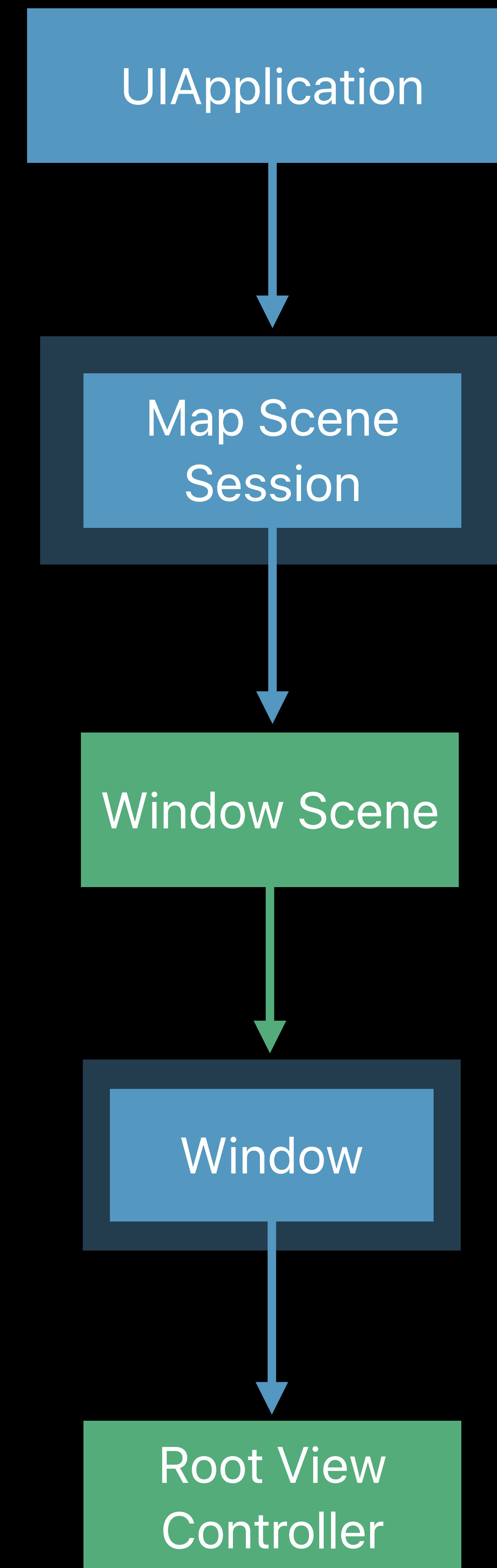
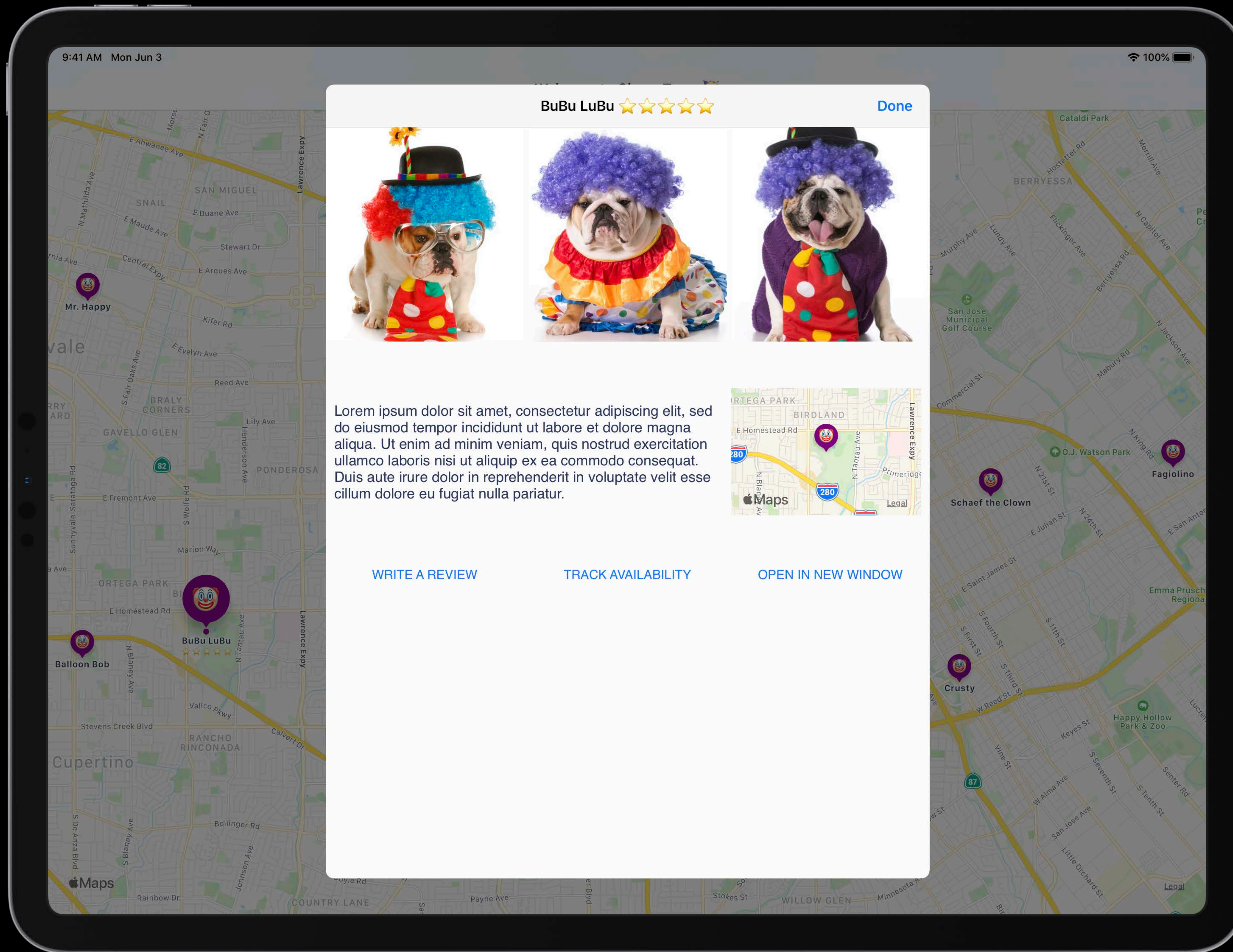
- New scene session
- Existing one

```
open func requestSceneSessionActivation(_ sceneSession: UISceneSession?,
                                       userActivity: NSUserActivity?,
                                       options: UIScene.ActivationRequestOptions?,
                                       errorHandler: ((Error) -> Void)? = nil)
```



# Session Activation

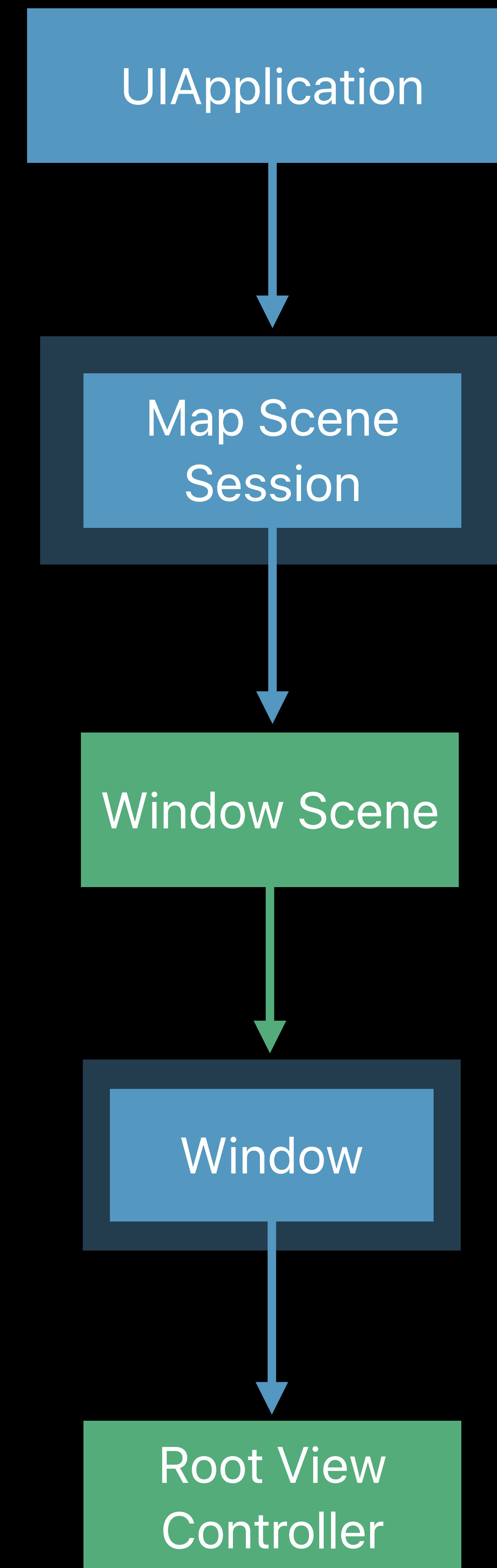
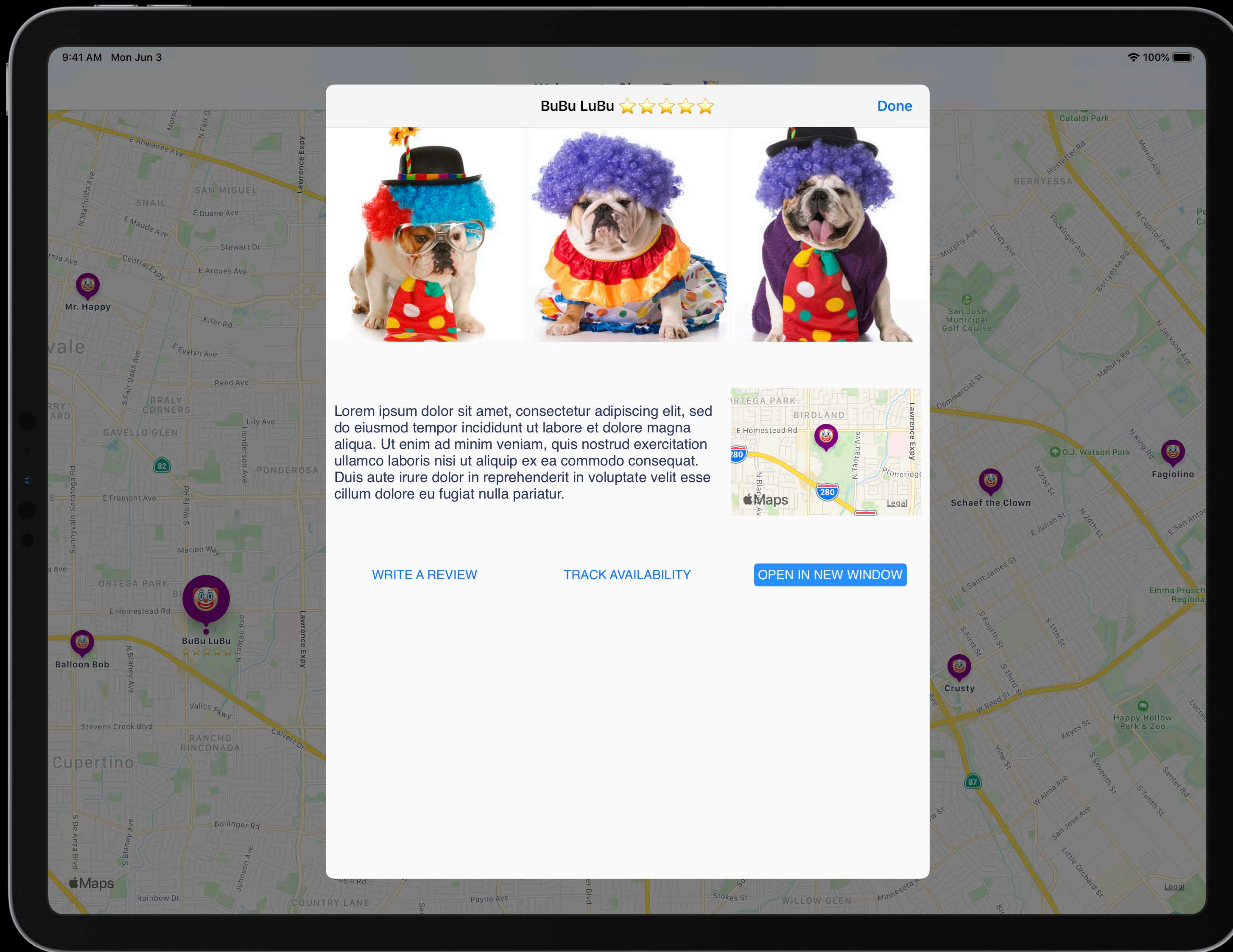
## New scene session





# Session Activation

## New scene session



















```
// ClownTown: Session Activation
```

```
@UIApplicationMain
```

```
class AppDelegate: UIResponder, UIApplicationDelegate {
```

```
    func activateSessionForClown(_ clown: Clown!,  
                                sourceScene: UIWindowScene?,  
                                errorHandler: ((Error) -> Void)? = nil)
```

```
{
```

```
    let existingSession = existingSessionForClown(clown)
```

```
    let userActivity = userActivityForClown(clown)
```

```
    let options = UIWindow.ActivationRequestOptions.init()
```

```
    options.requestingScene = sourceScene
```

```
    UIApplication.shared.requestSceneSessionActivation(existingSession,
```

```
                userActivity: userActivity,
```

```
                options: options,
```

```
                errorHandler: errorHandler)
```

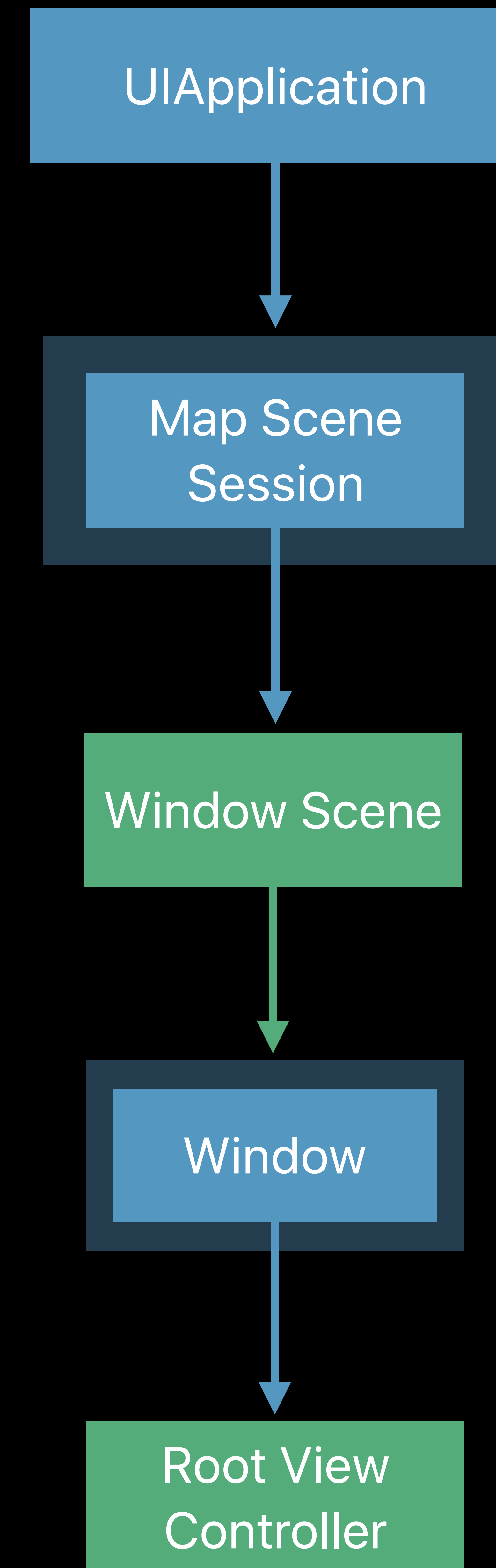
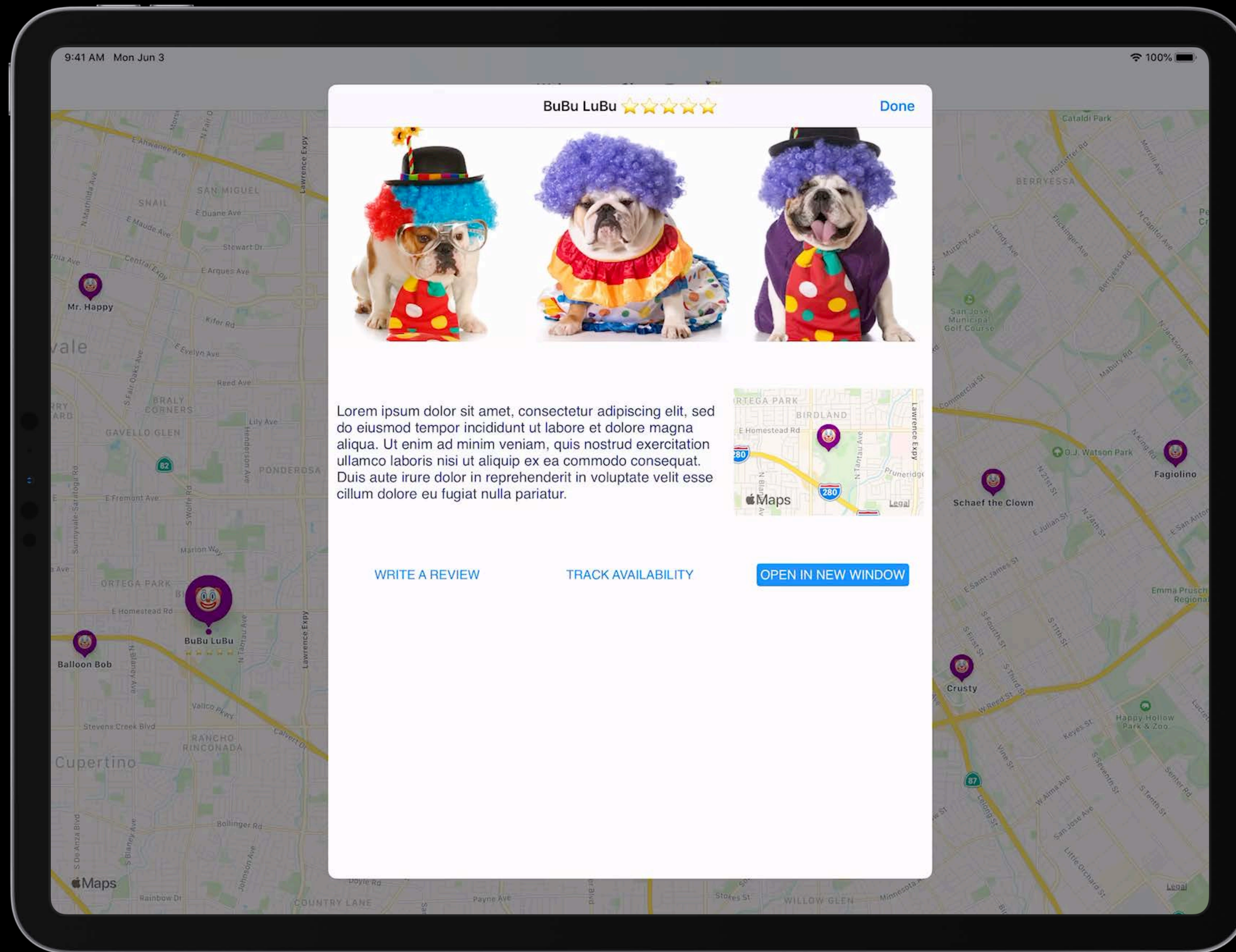
```
    }
```

```
}
```



# Session Activation

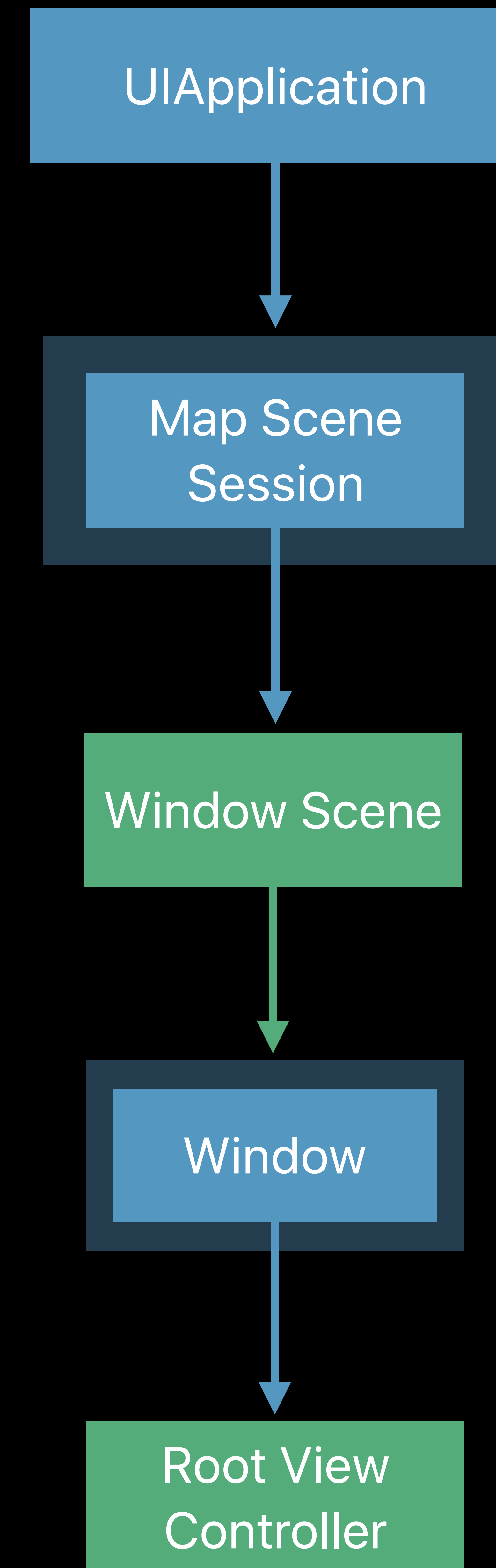
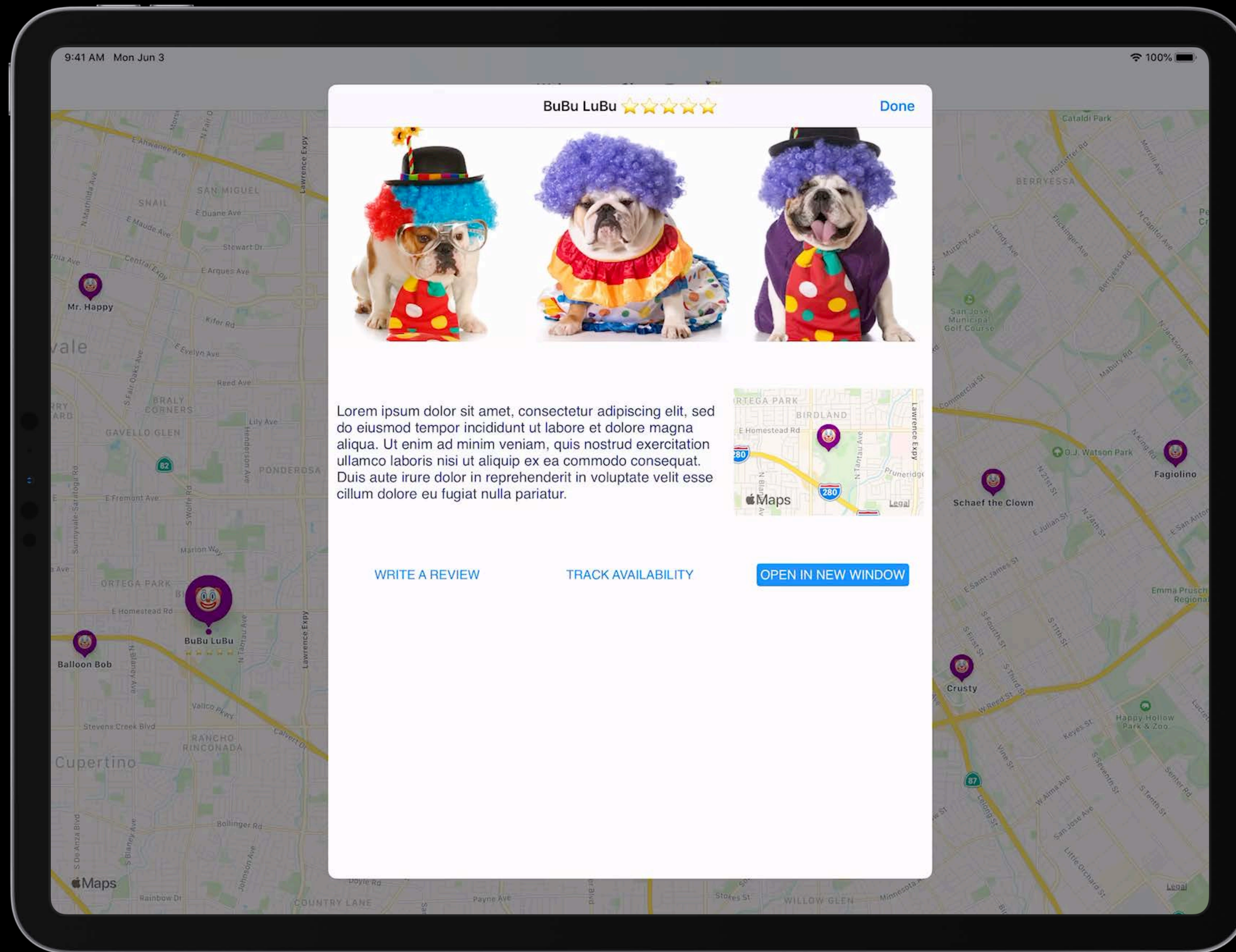
## New scene session





# Session Activation

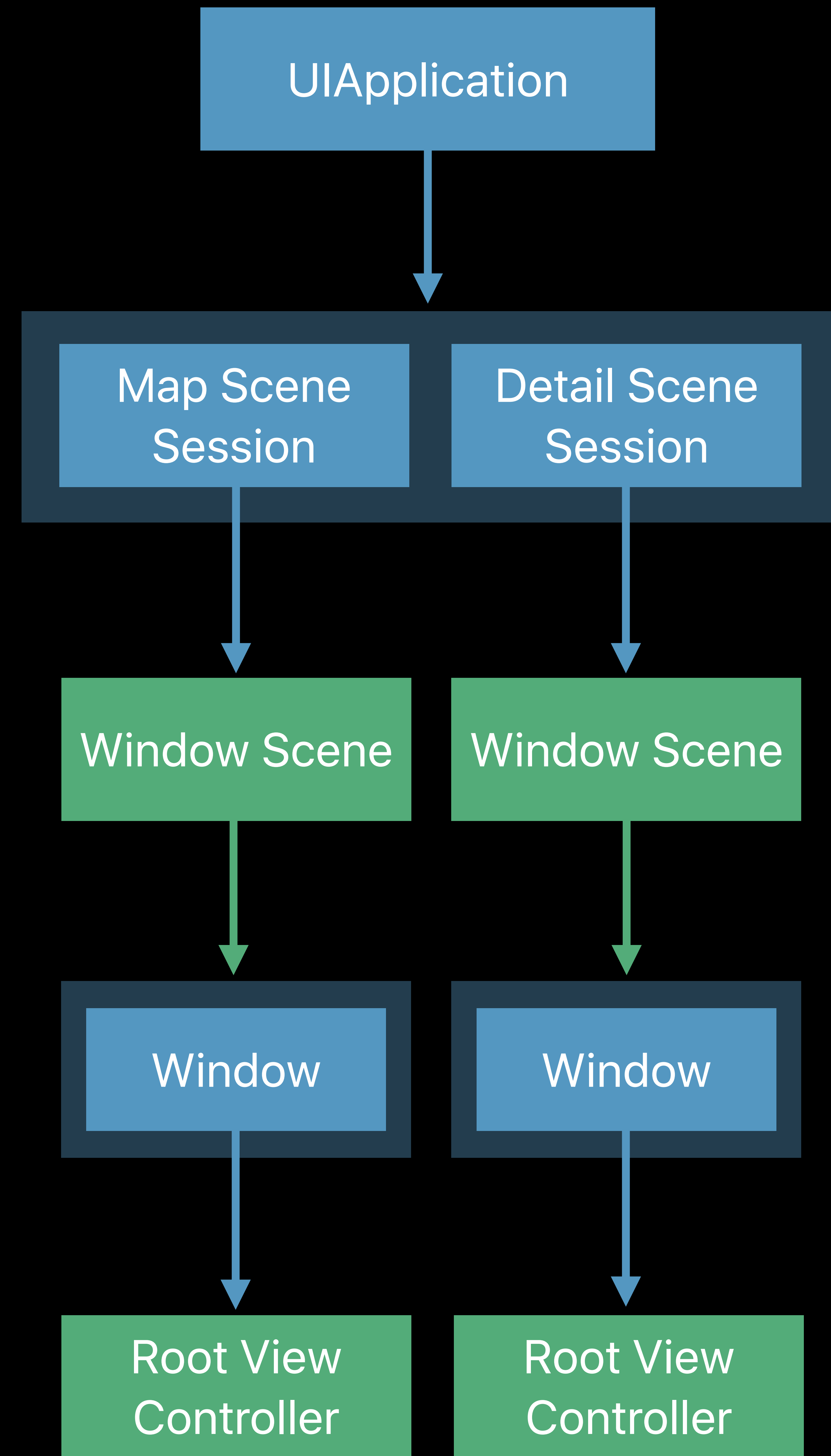
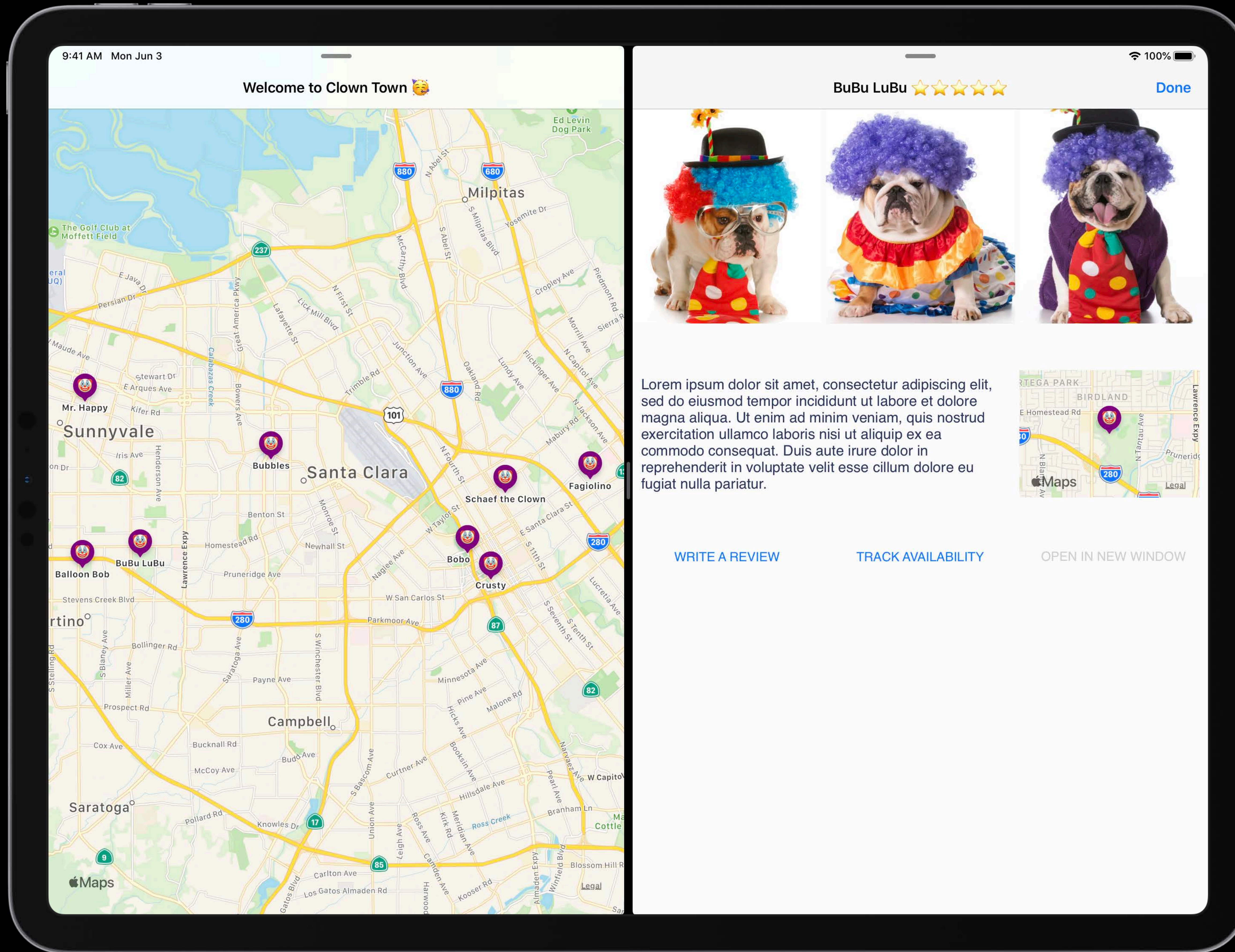
## New scene session





# Session Activation

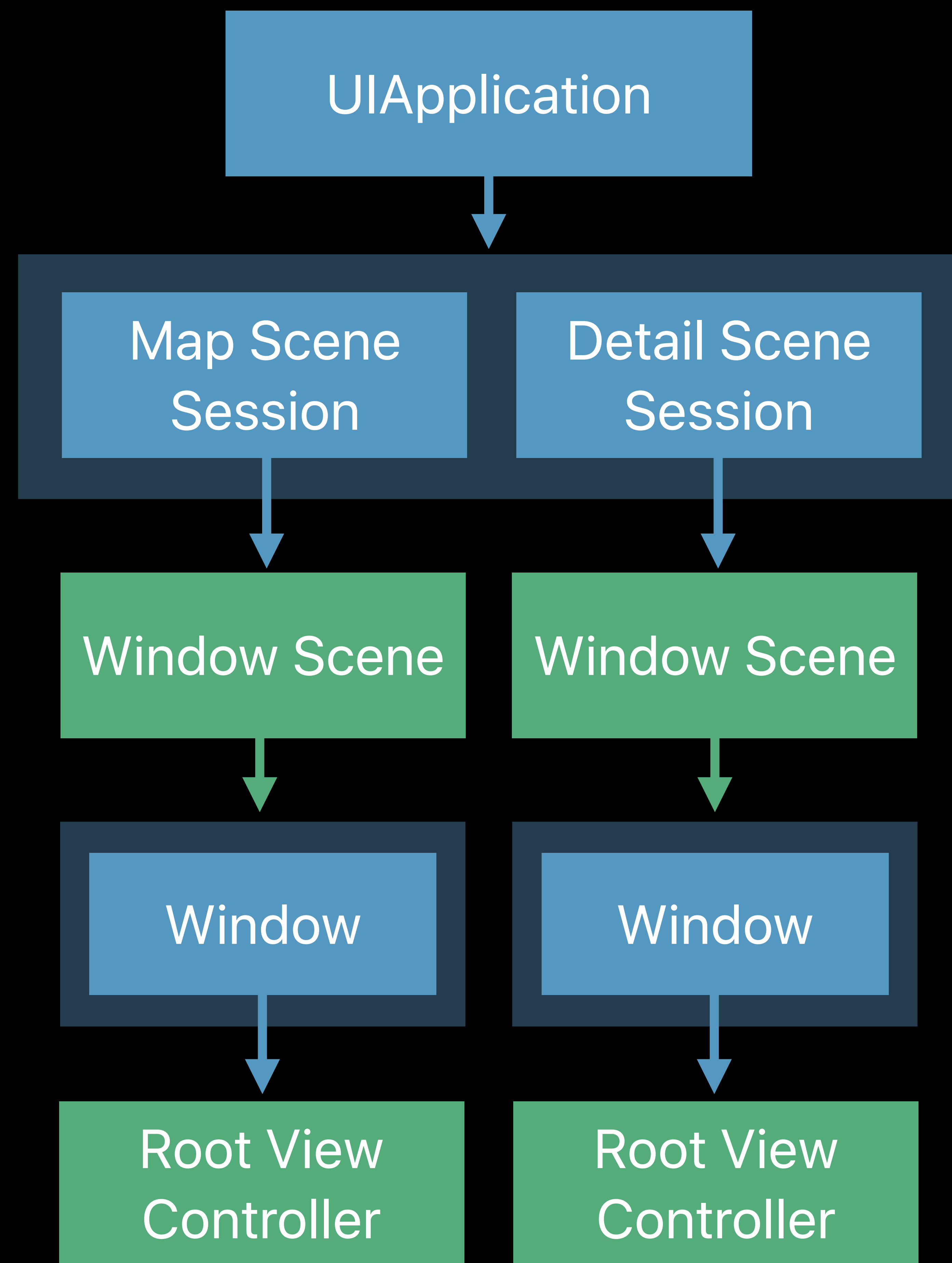
## New scene session





# Session Activation

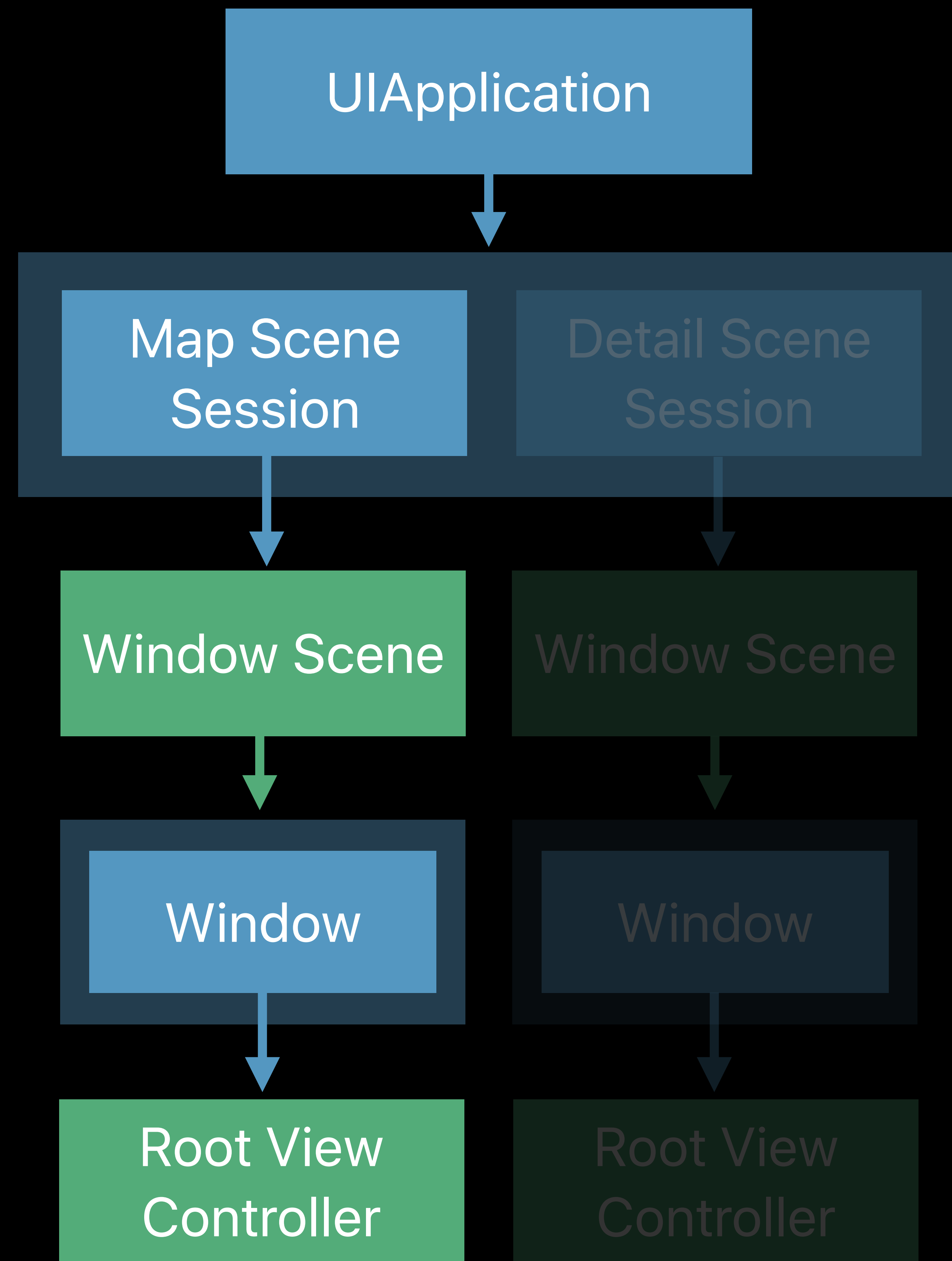
New scene session





# Session Activation

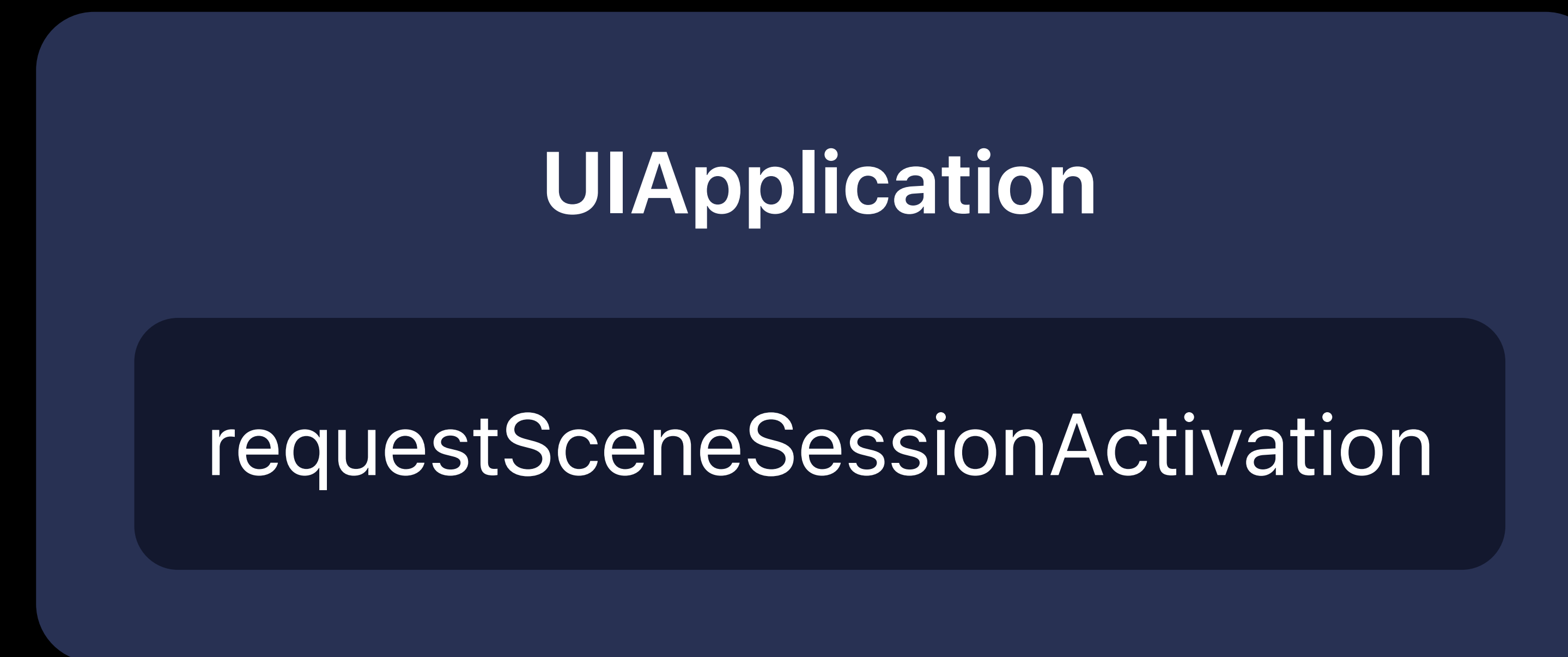
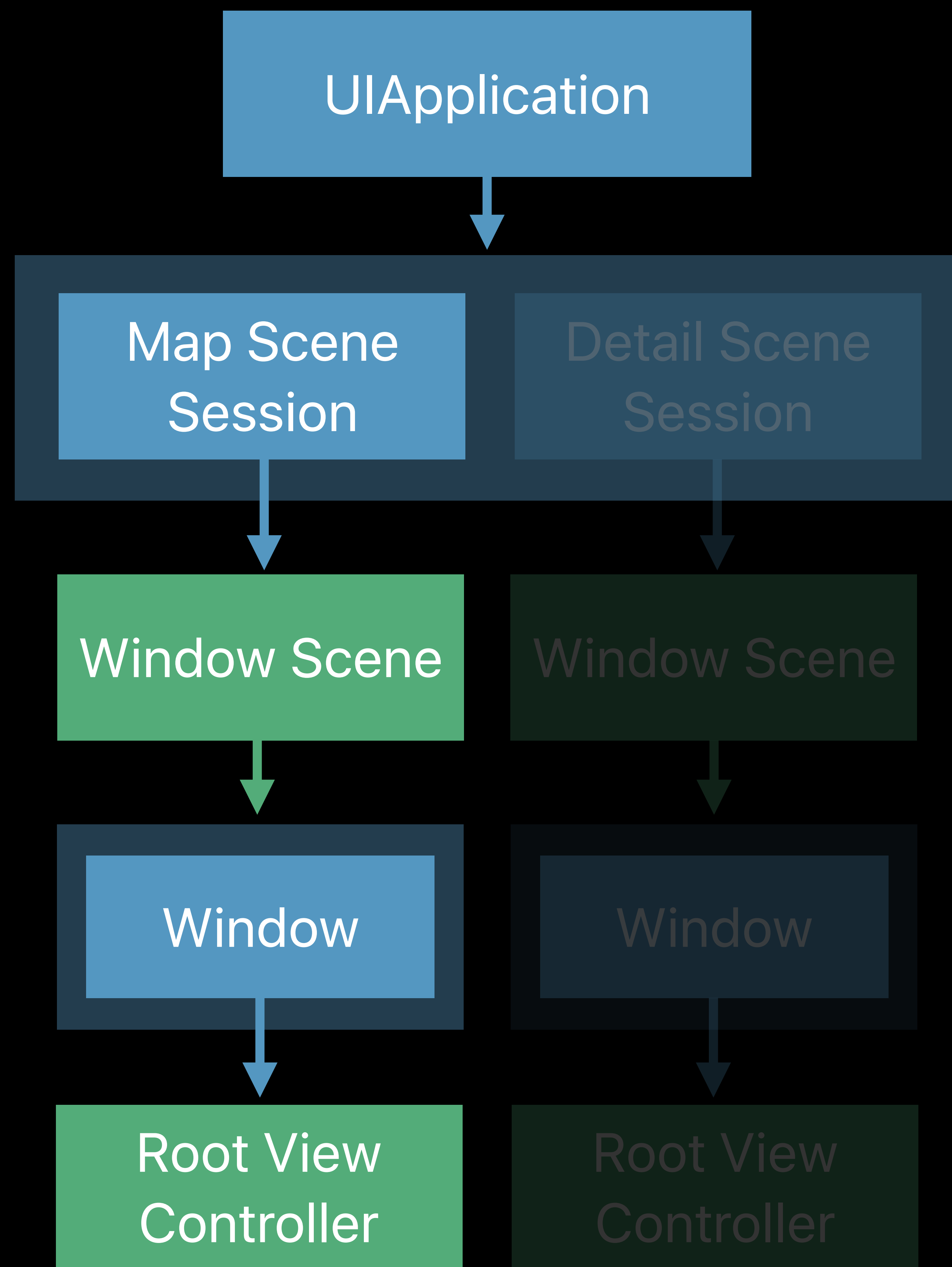
New scene session





# Session Activation

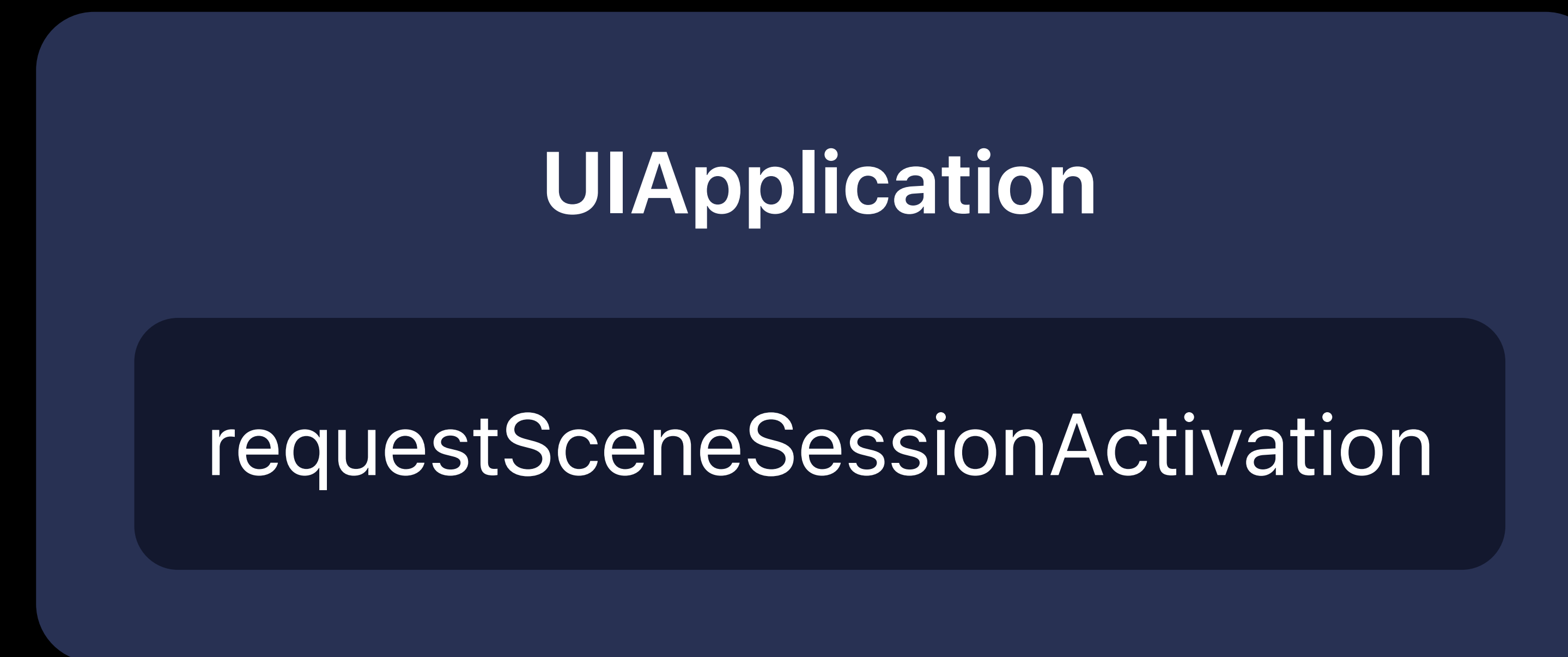
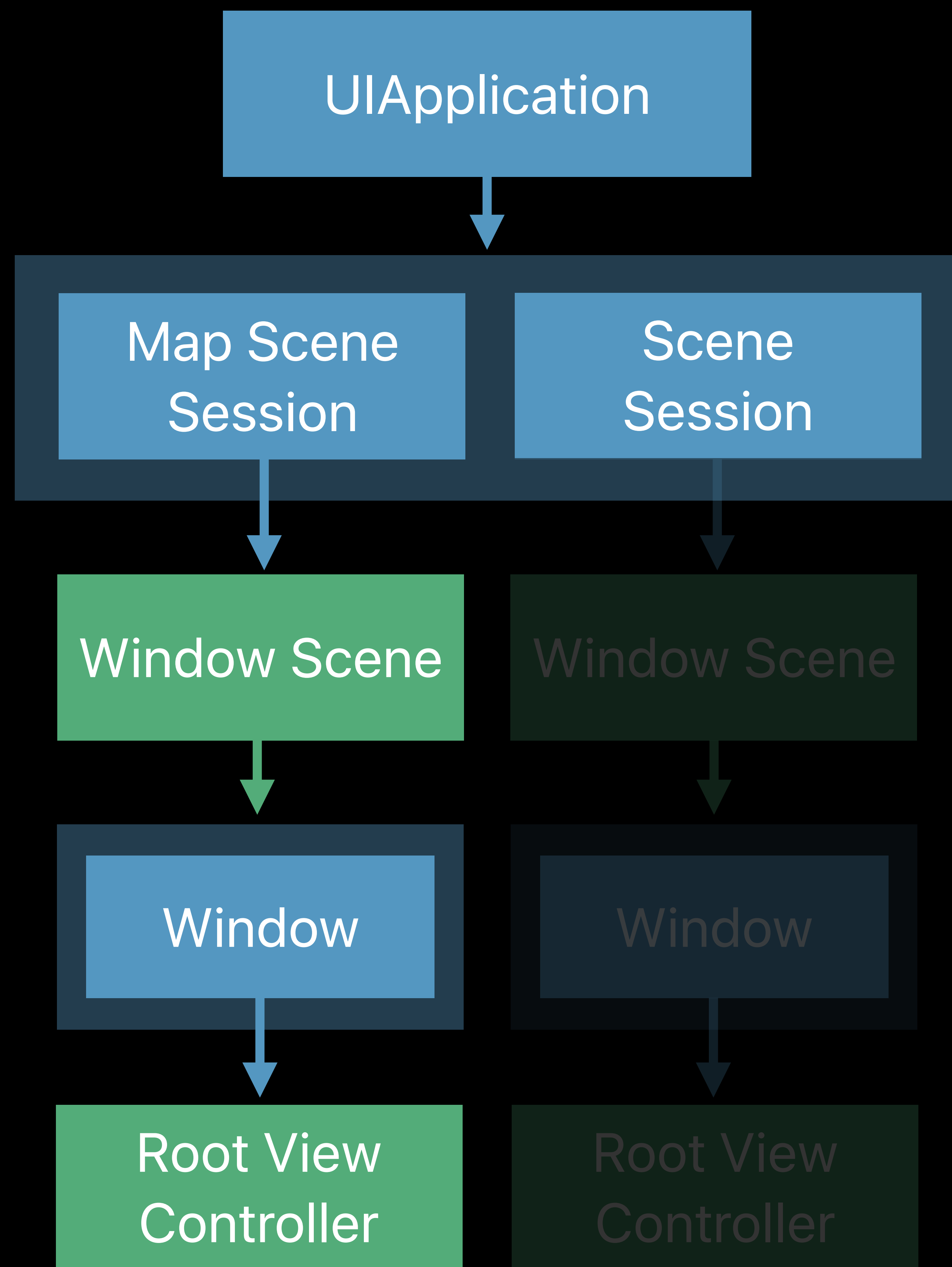
New scene session





# Session Activation

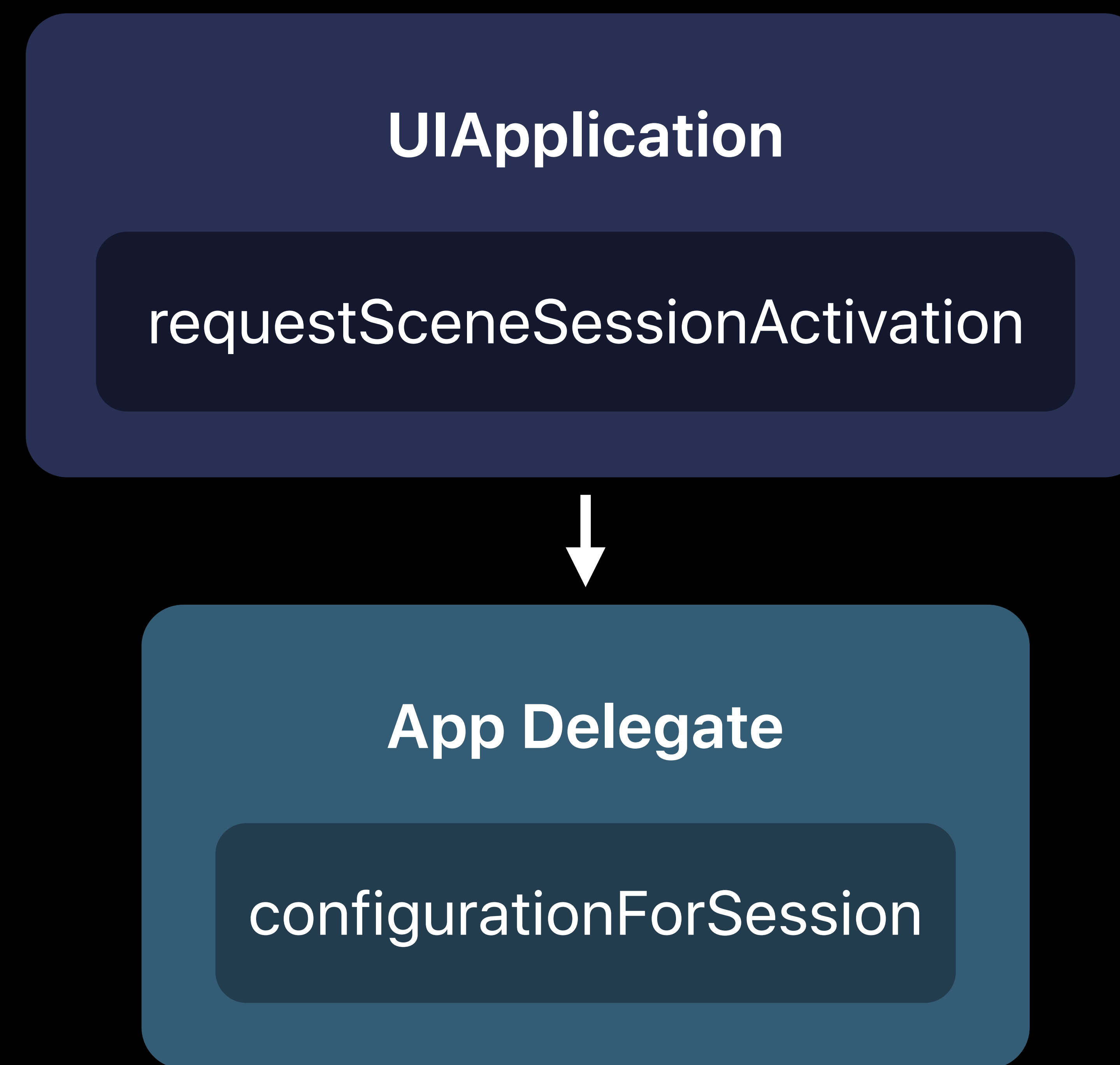
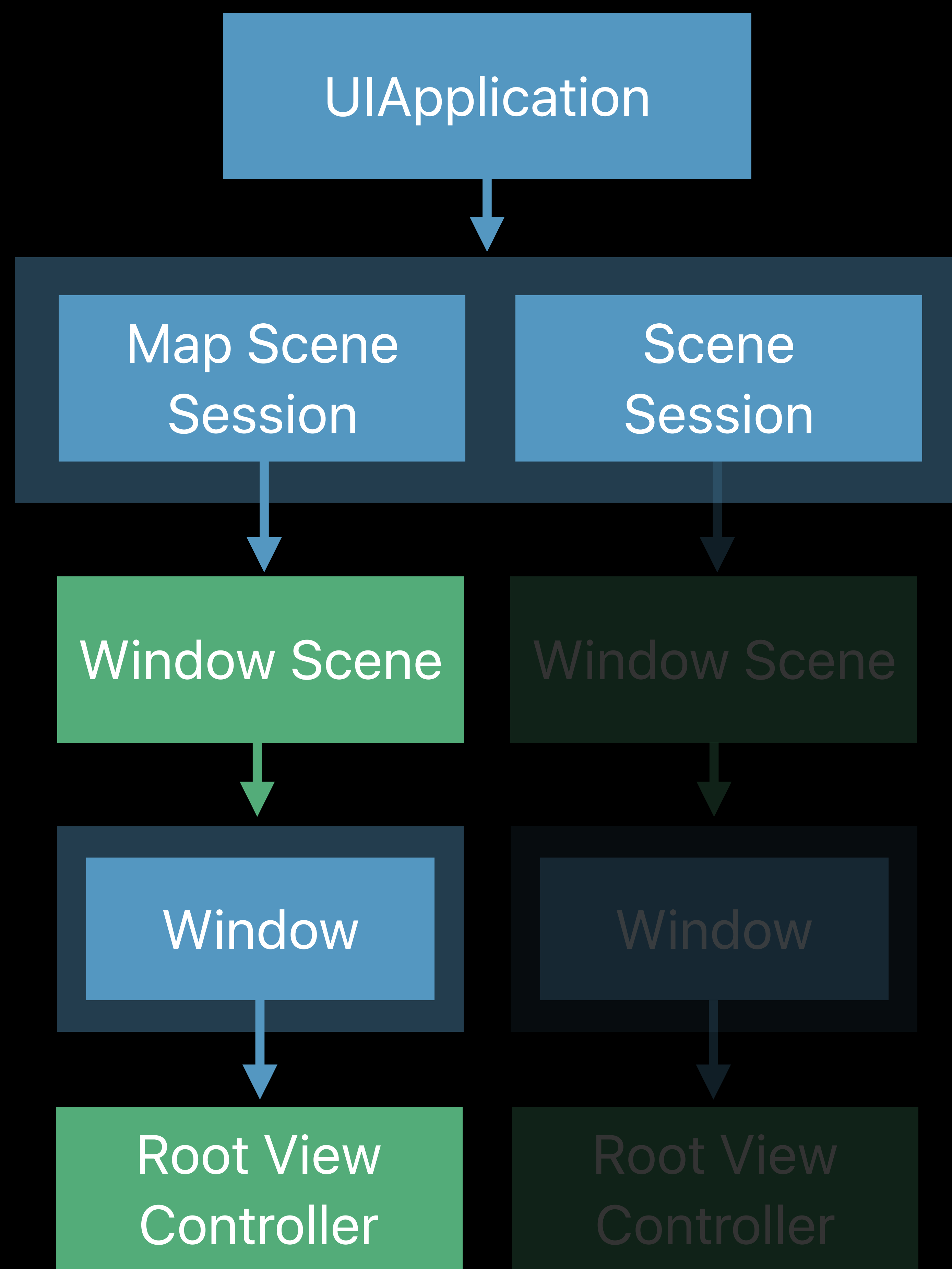
New scene session





# Session Activation

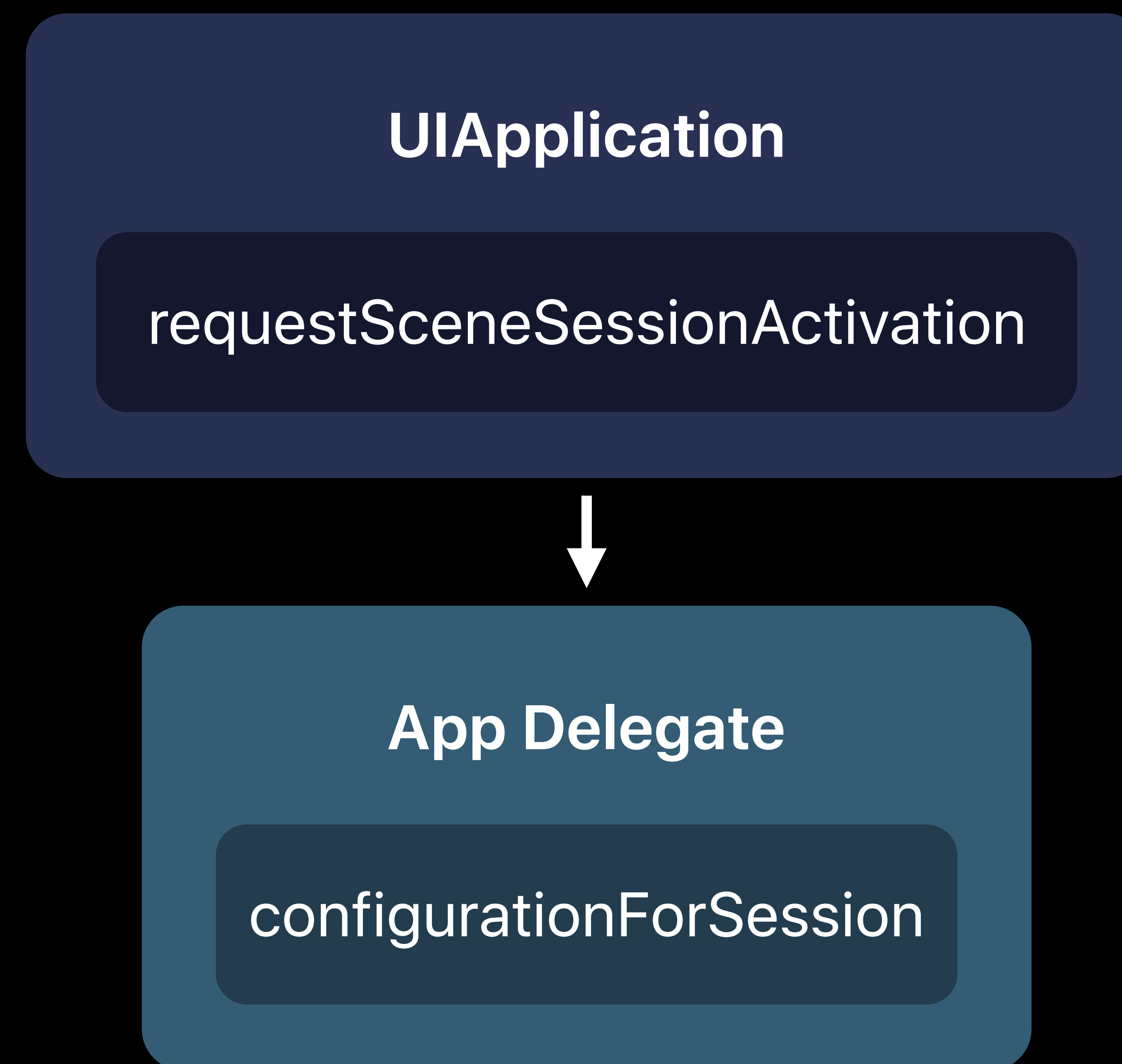
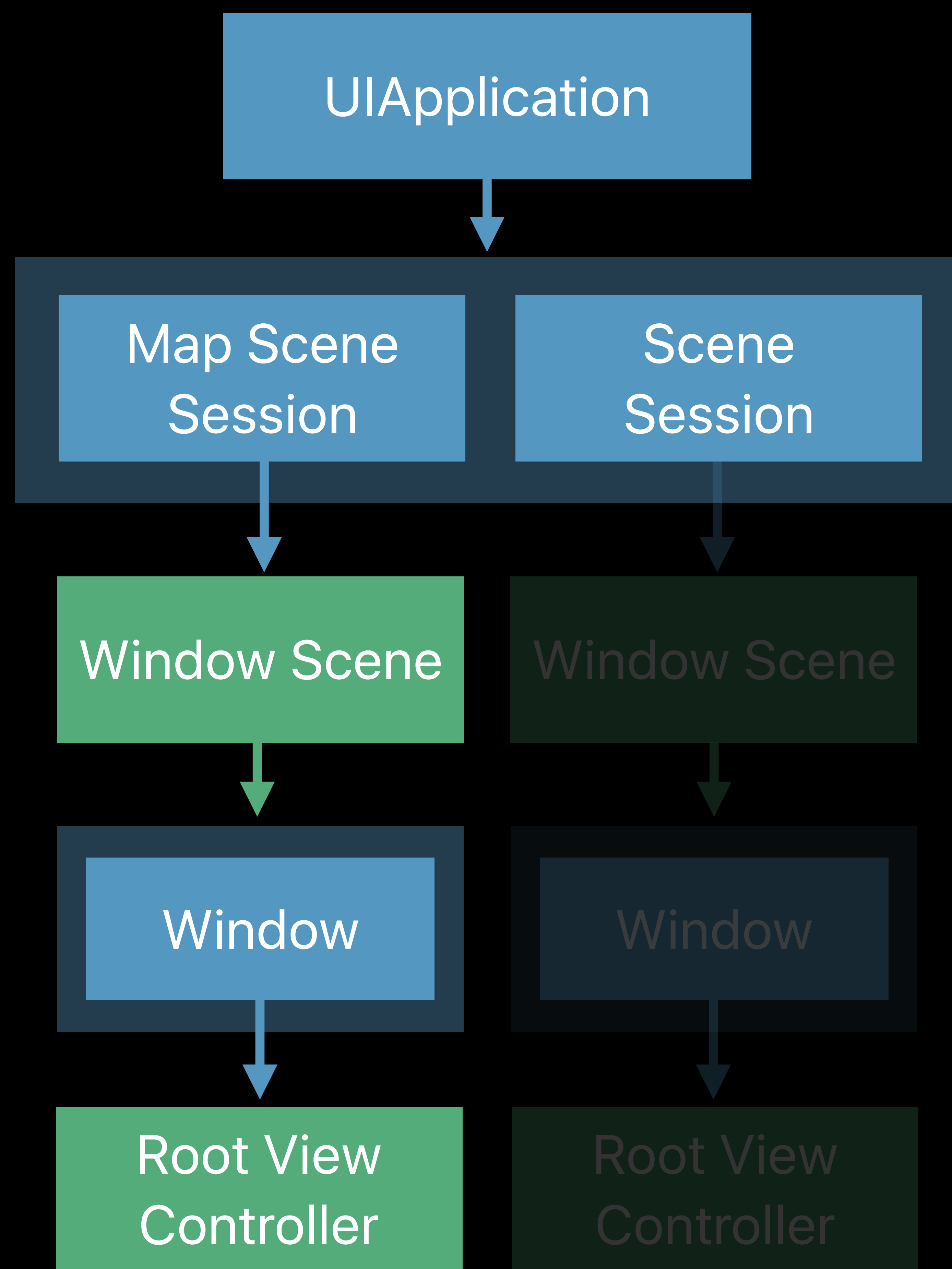
New scene session





# Session Activation

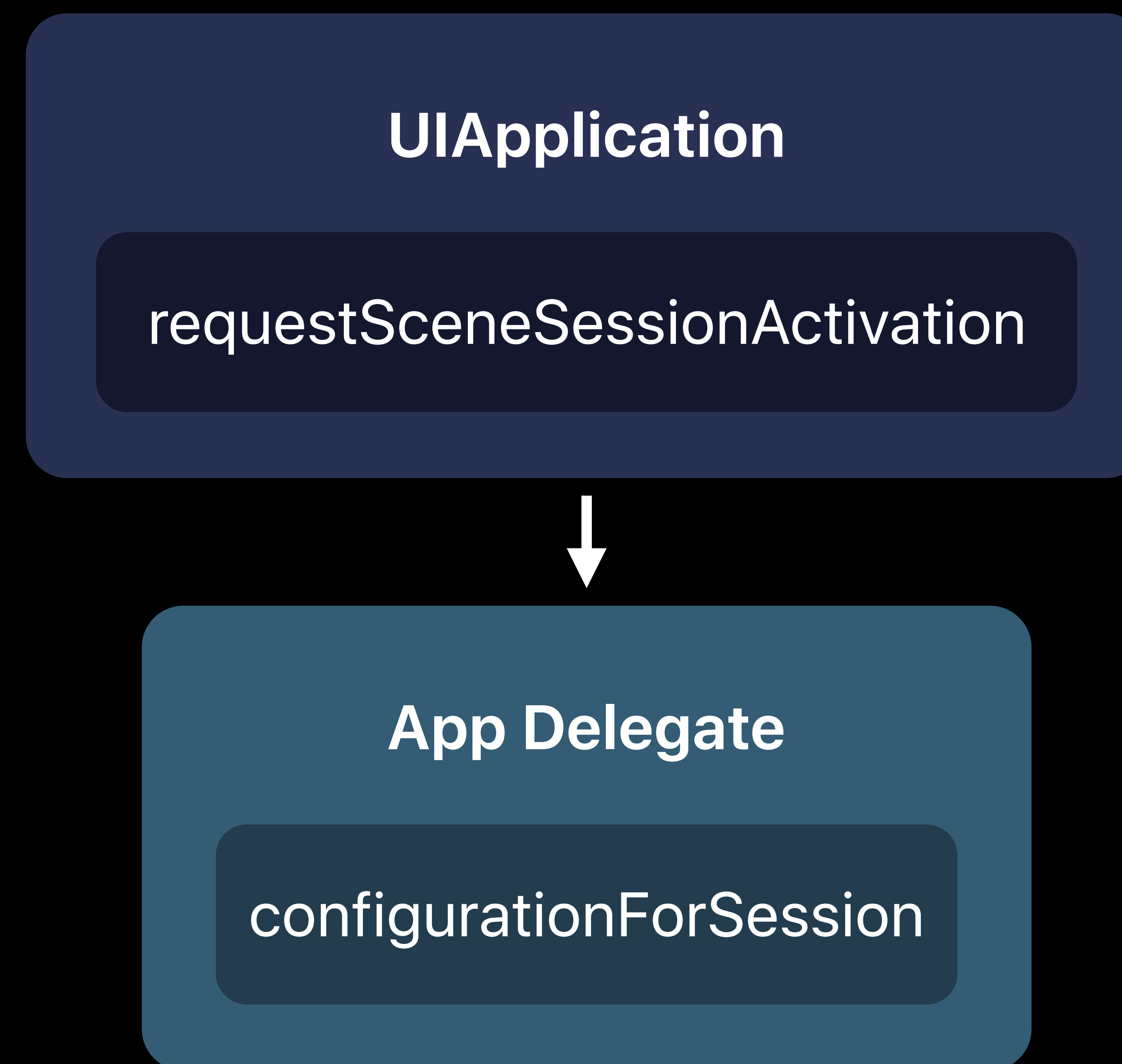
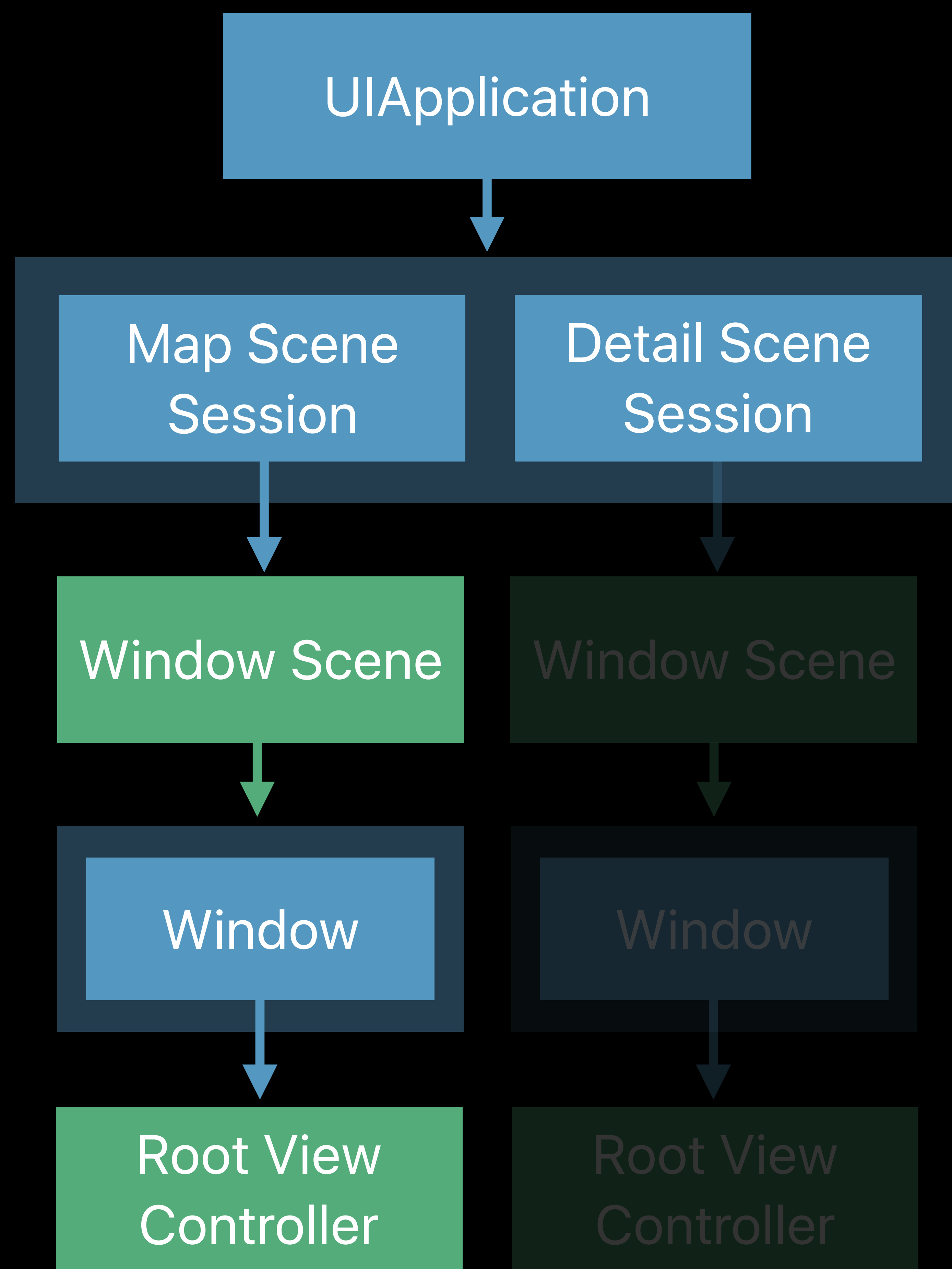
New scene session





# Session Activation

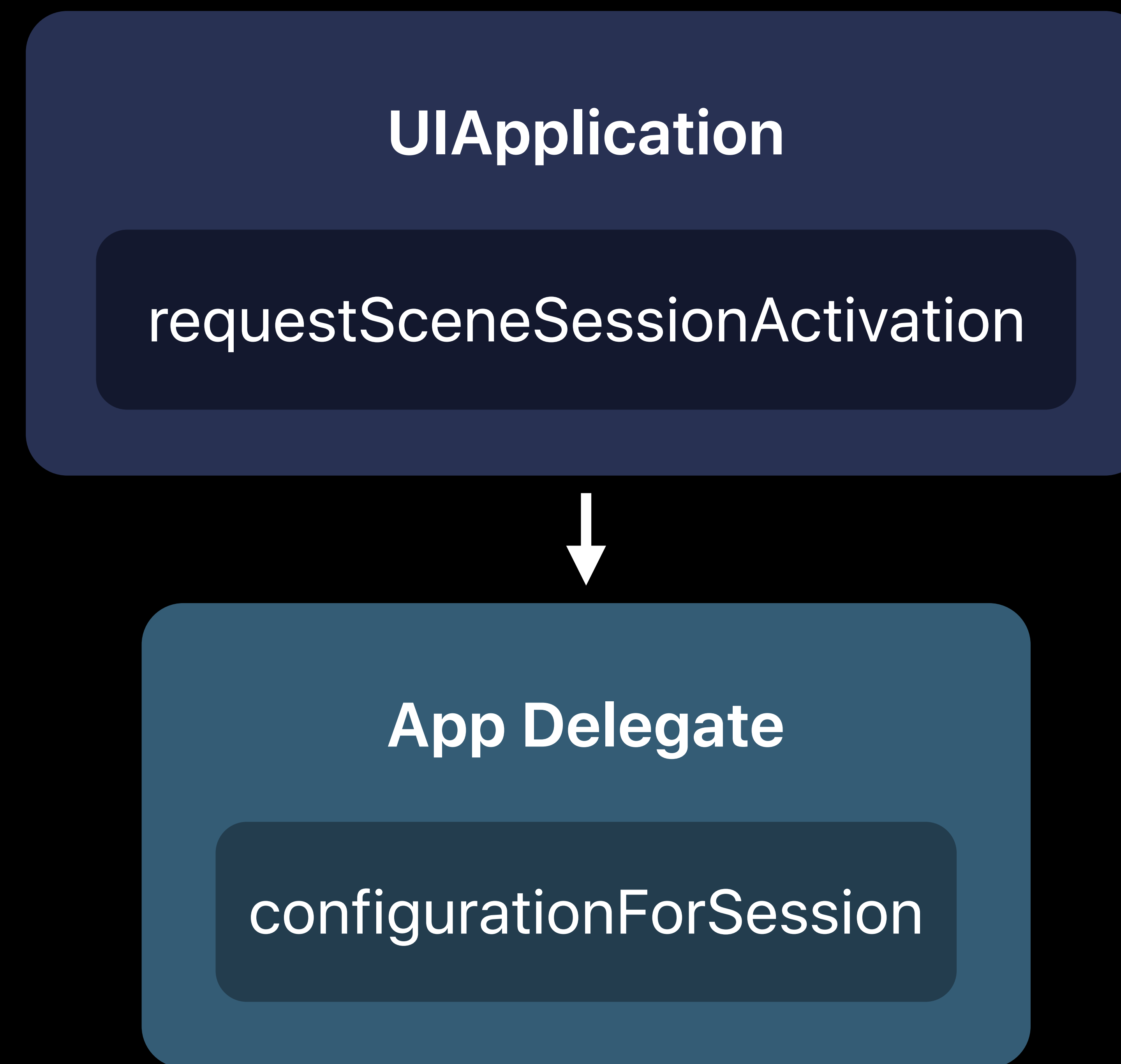
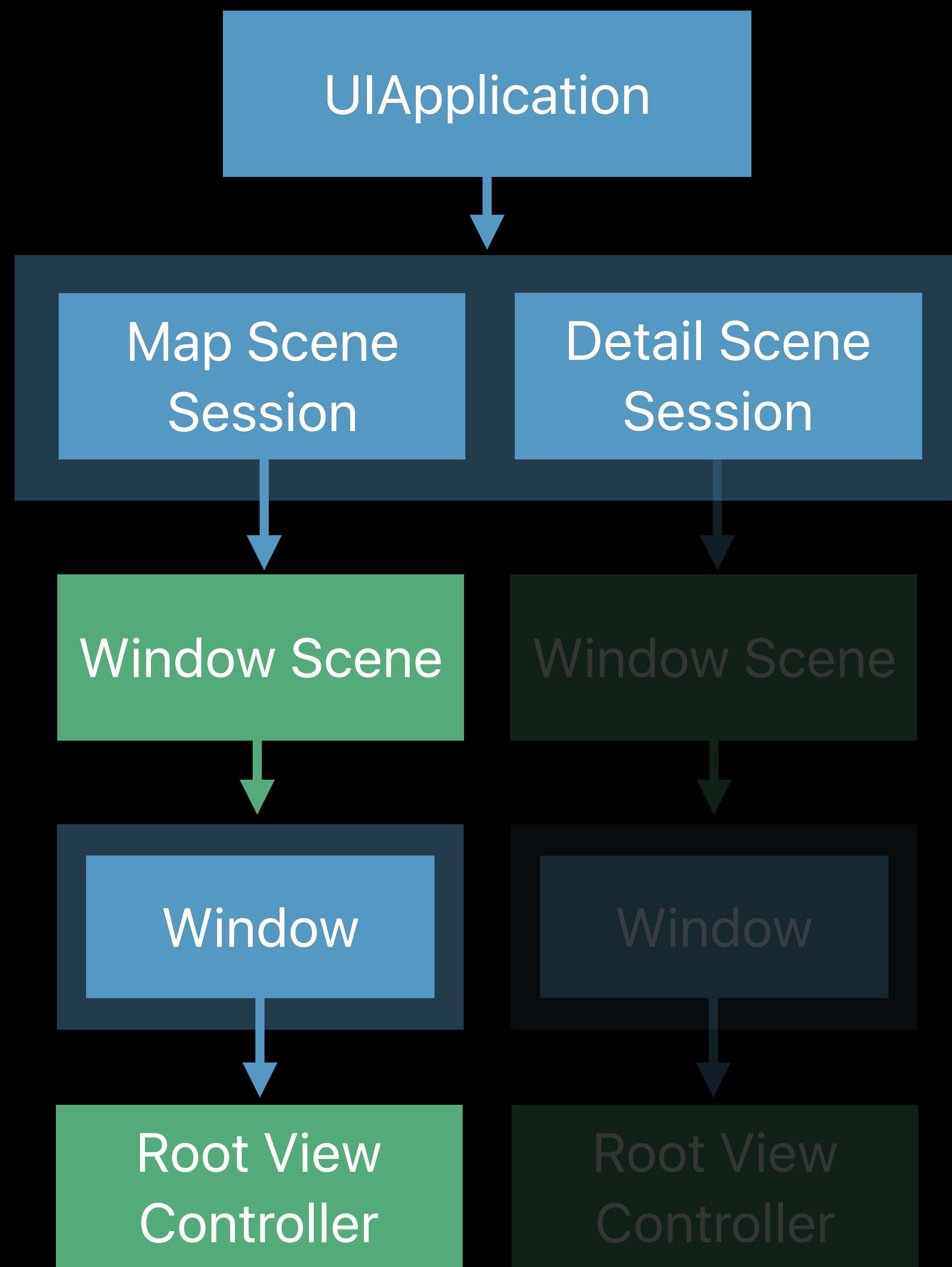
New scene session





# Session Activation

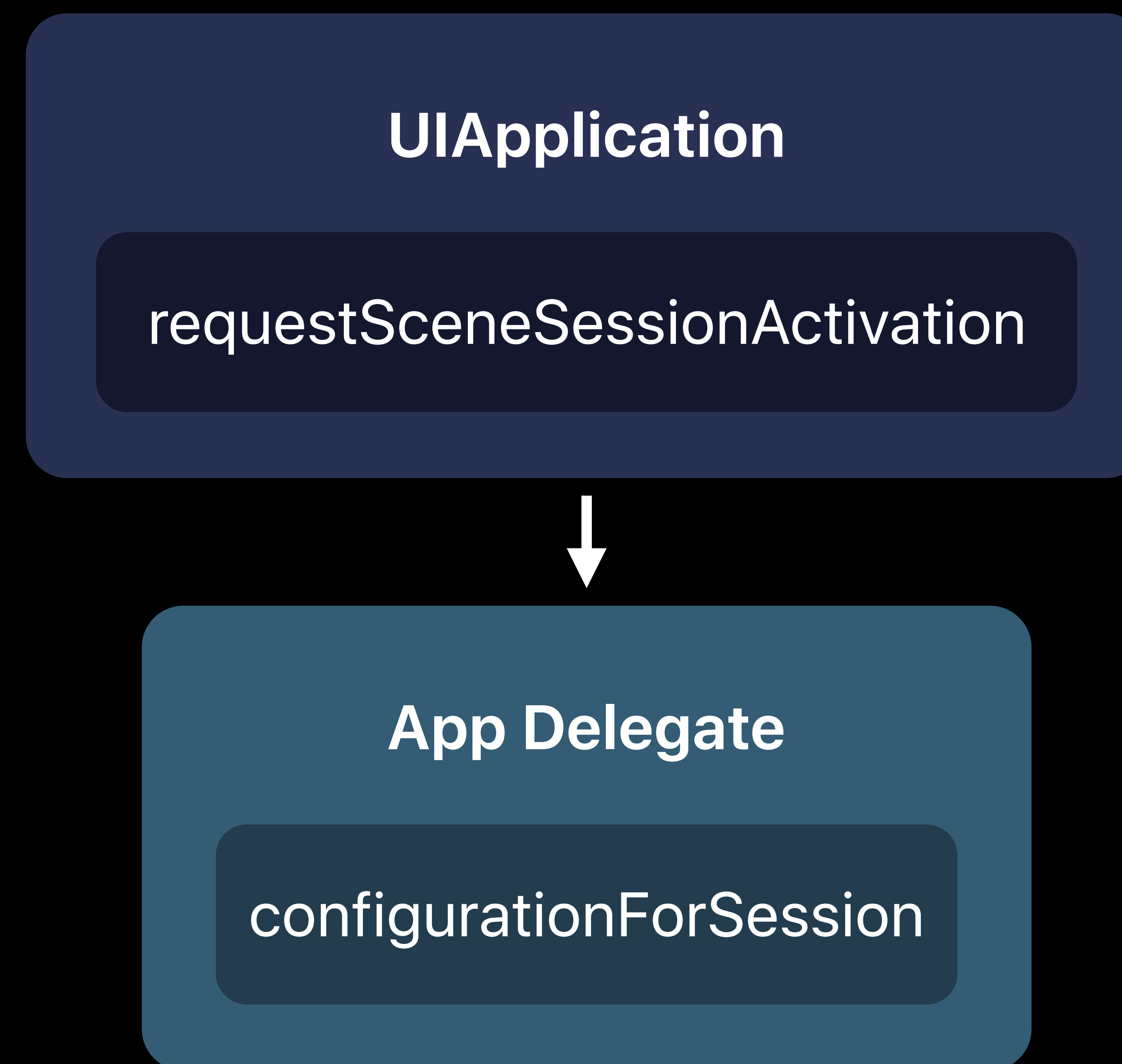
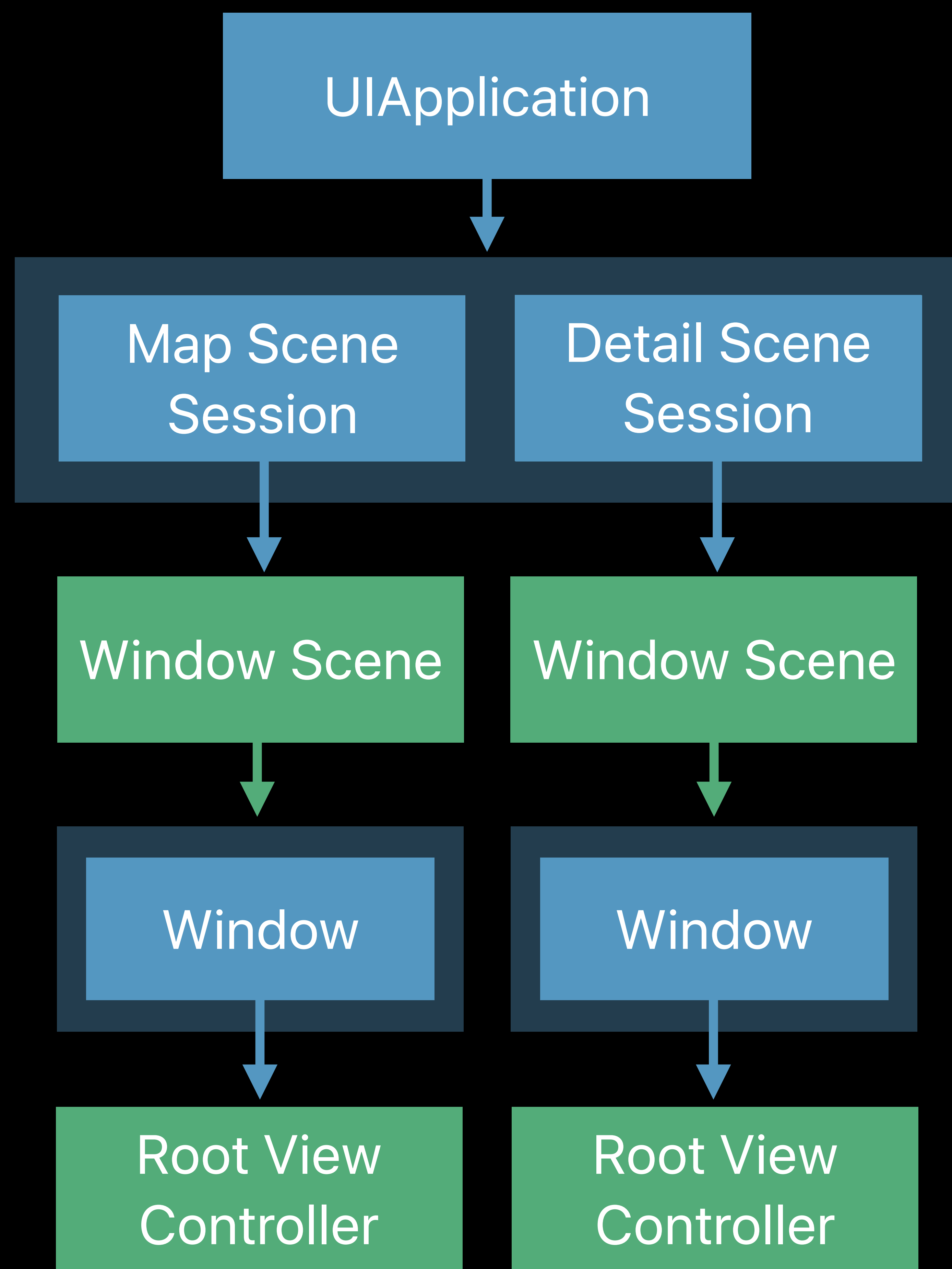
New scene session





# Session Activation

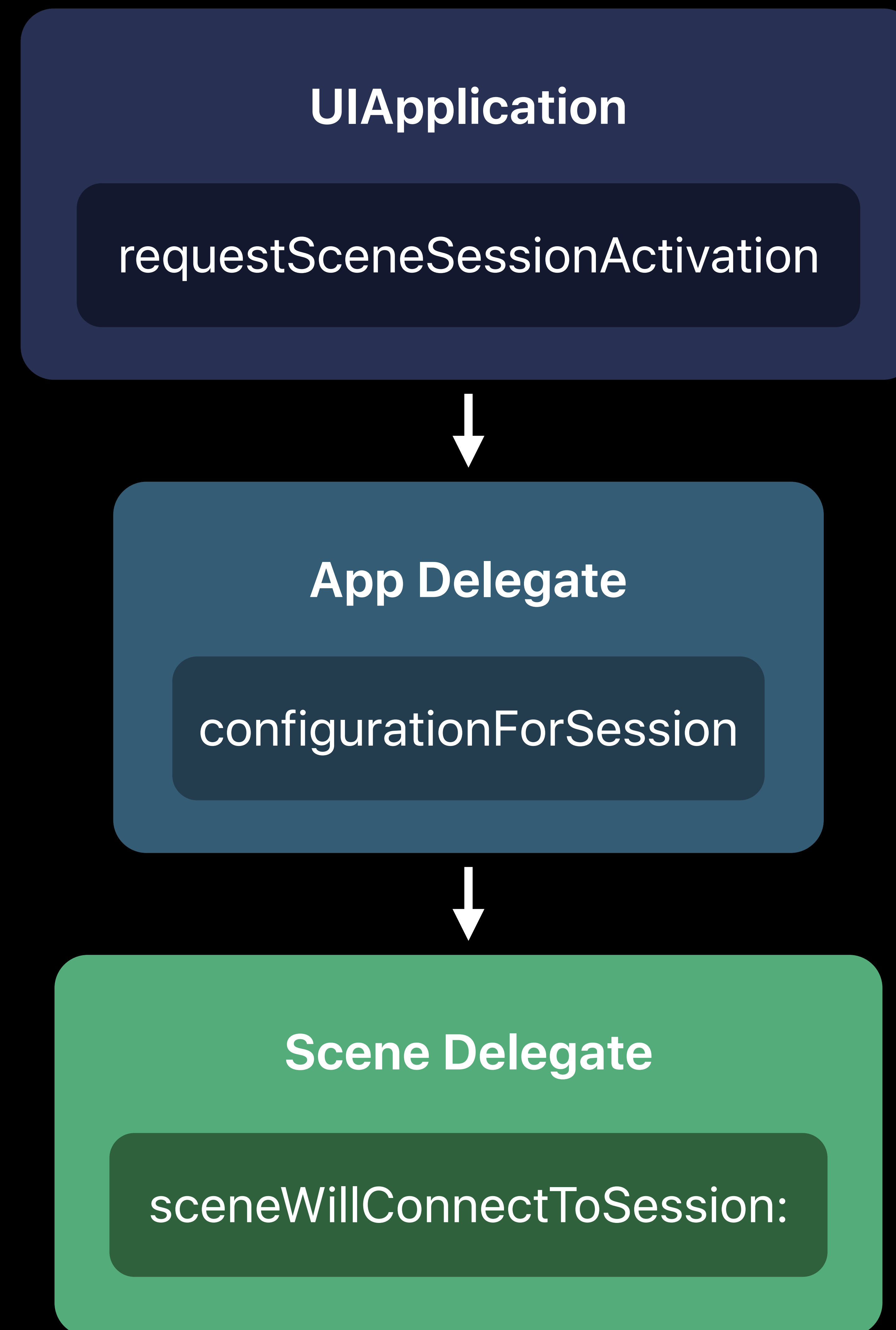
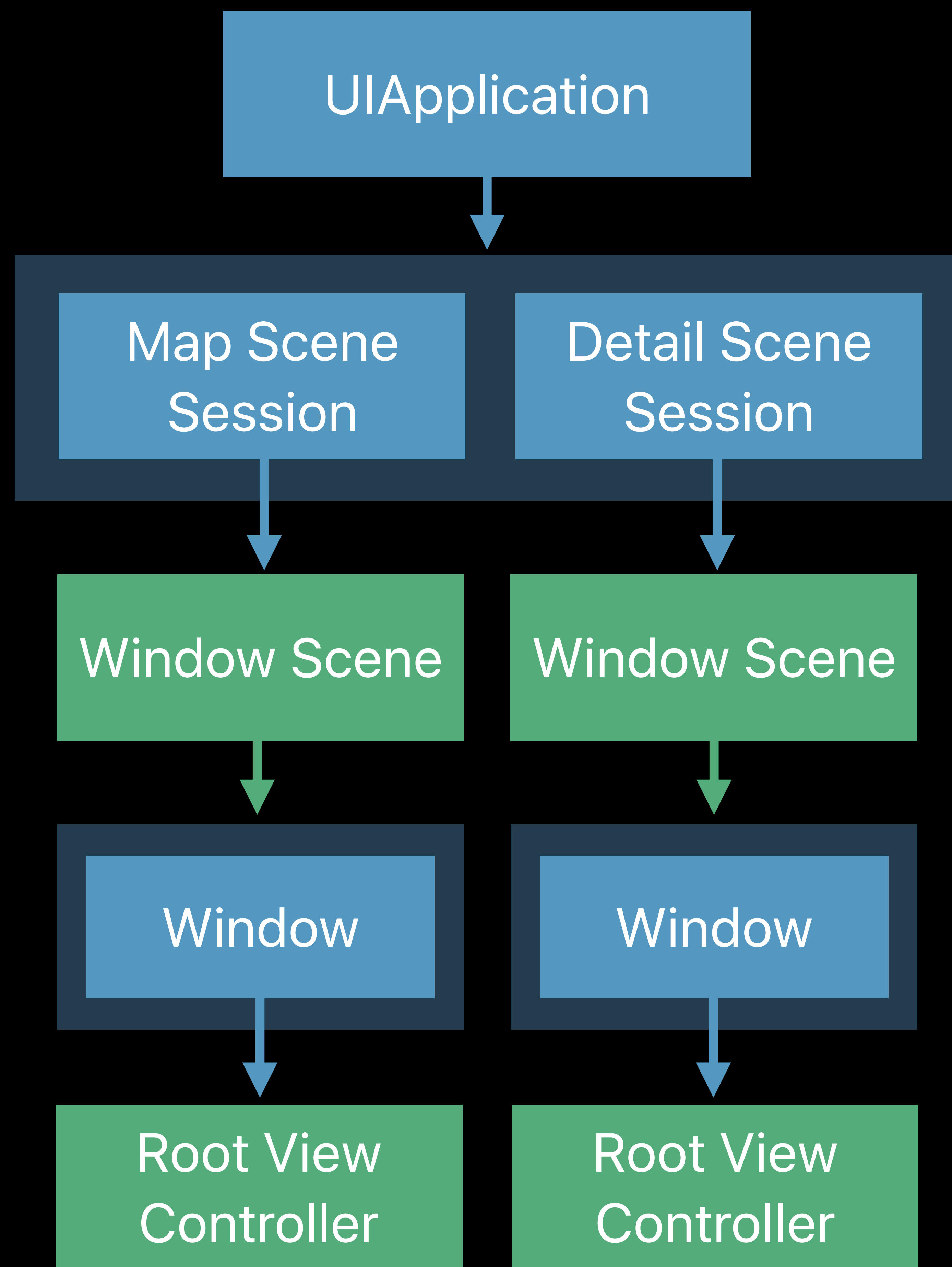
New scene session





# Session Activation

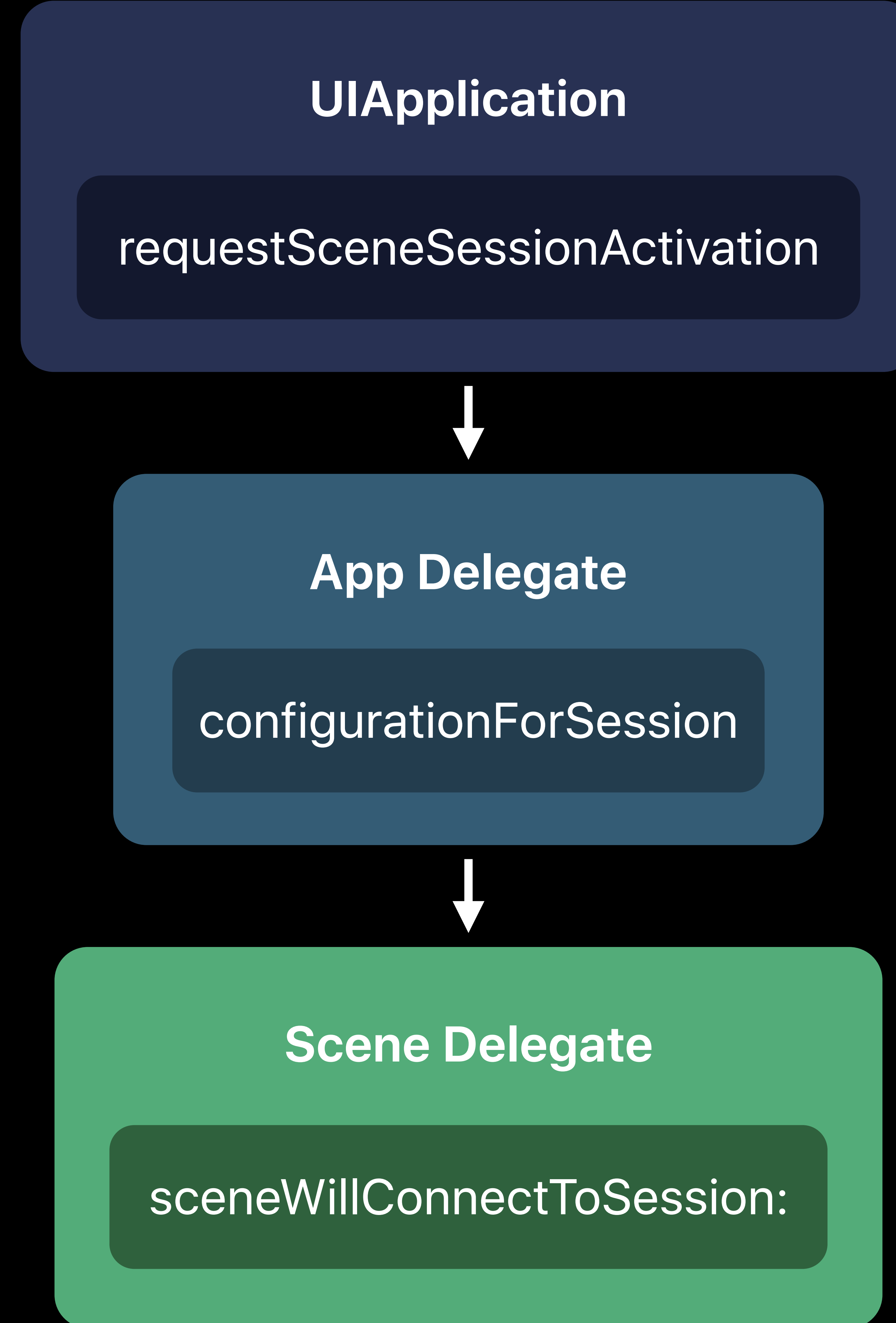
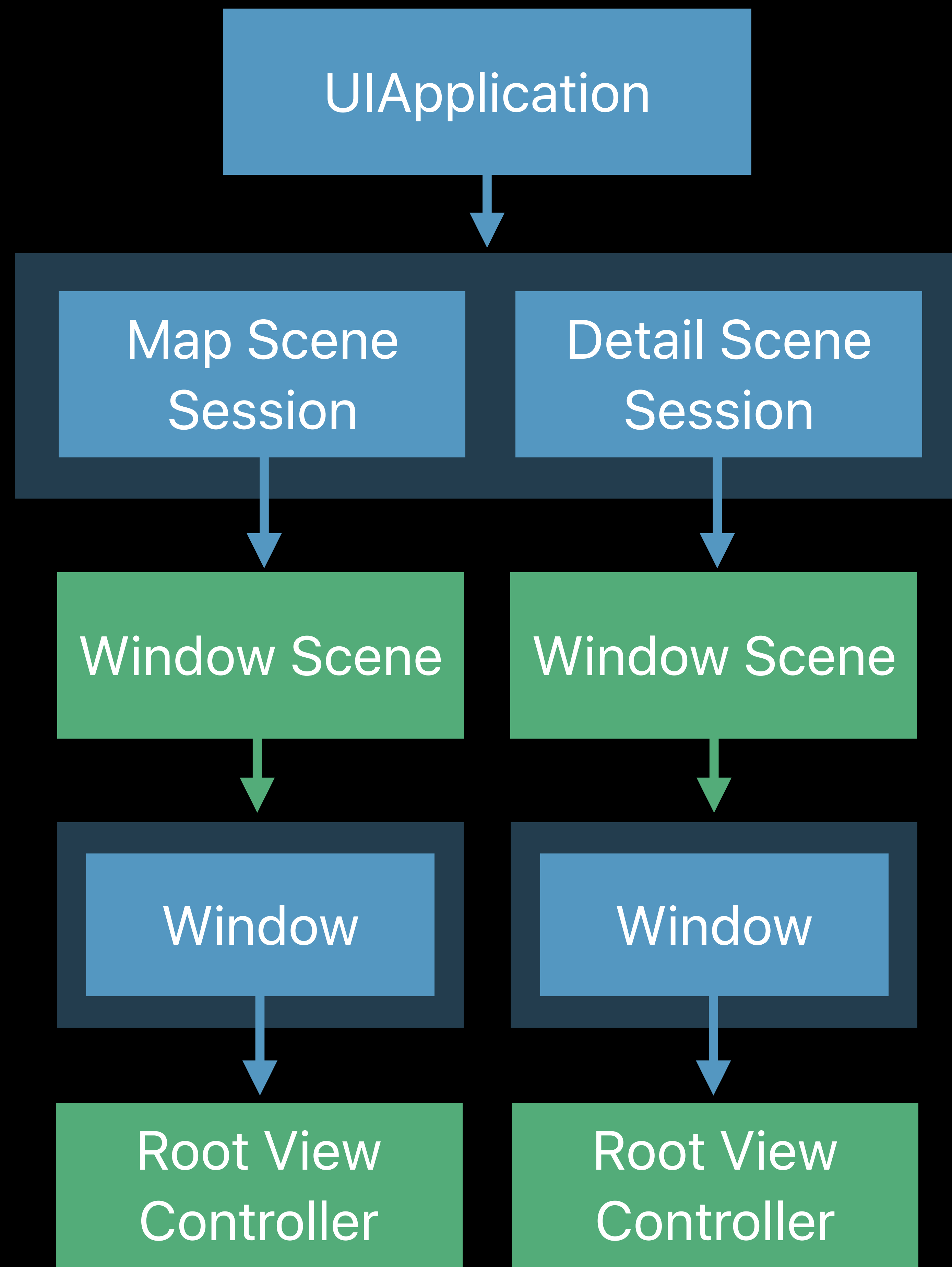
New scene session





# Session Activation

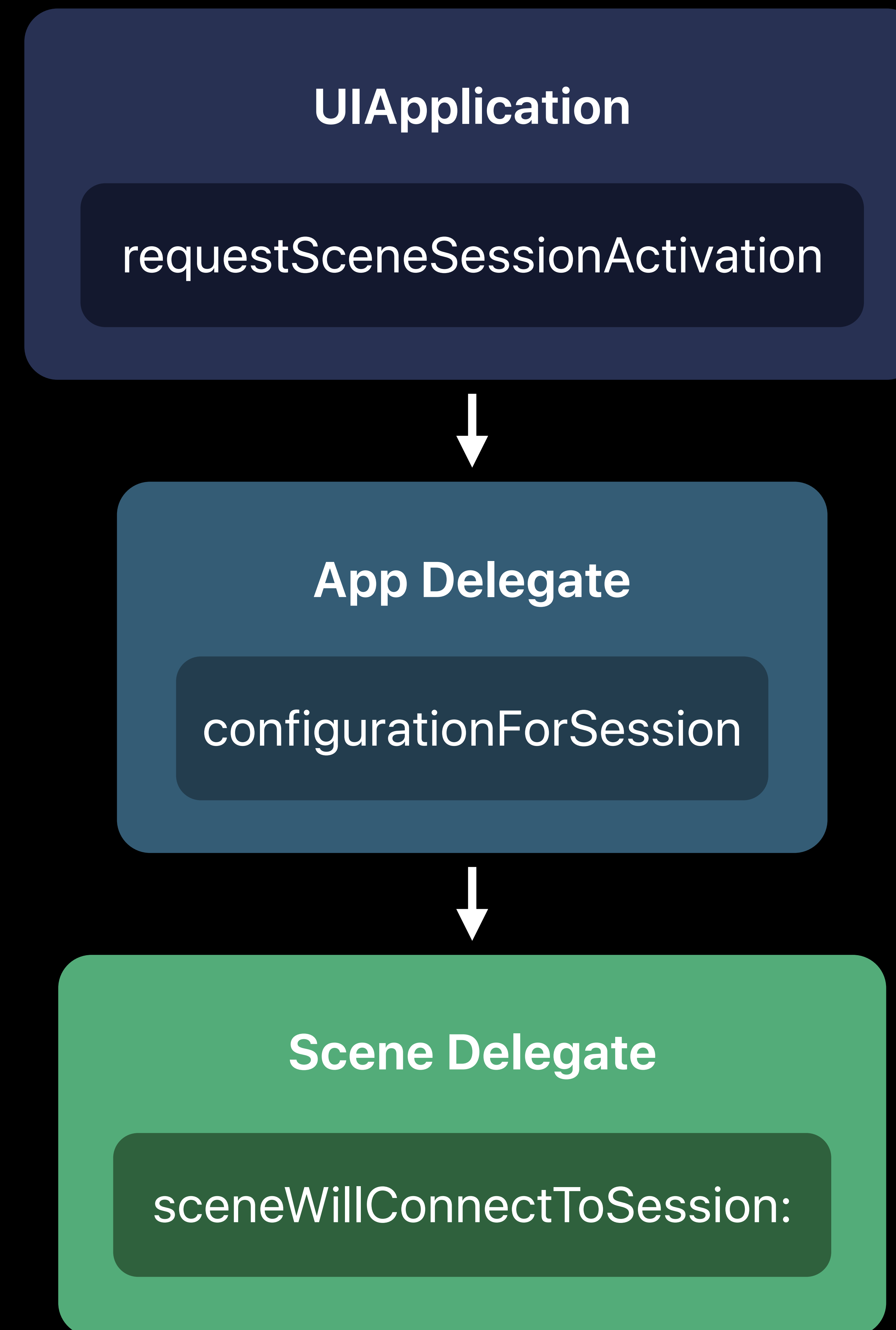
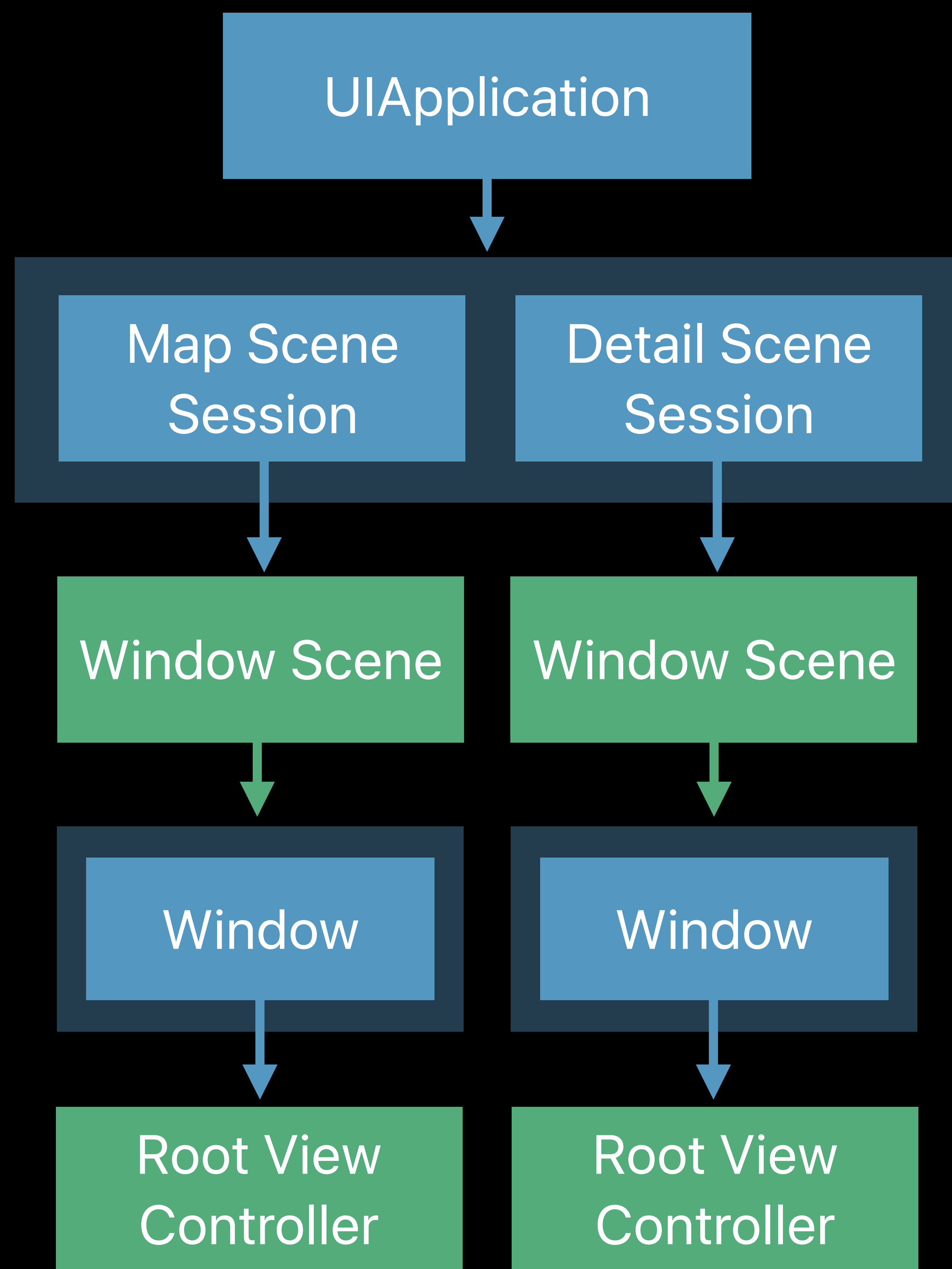
New scene session





# Session Activation

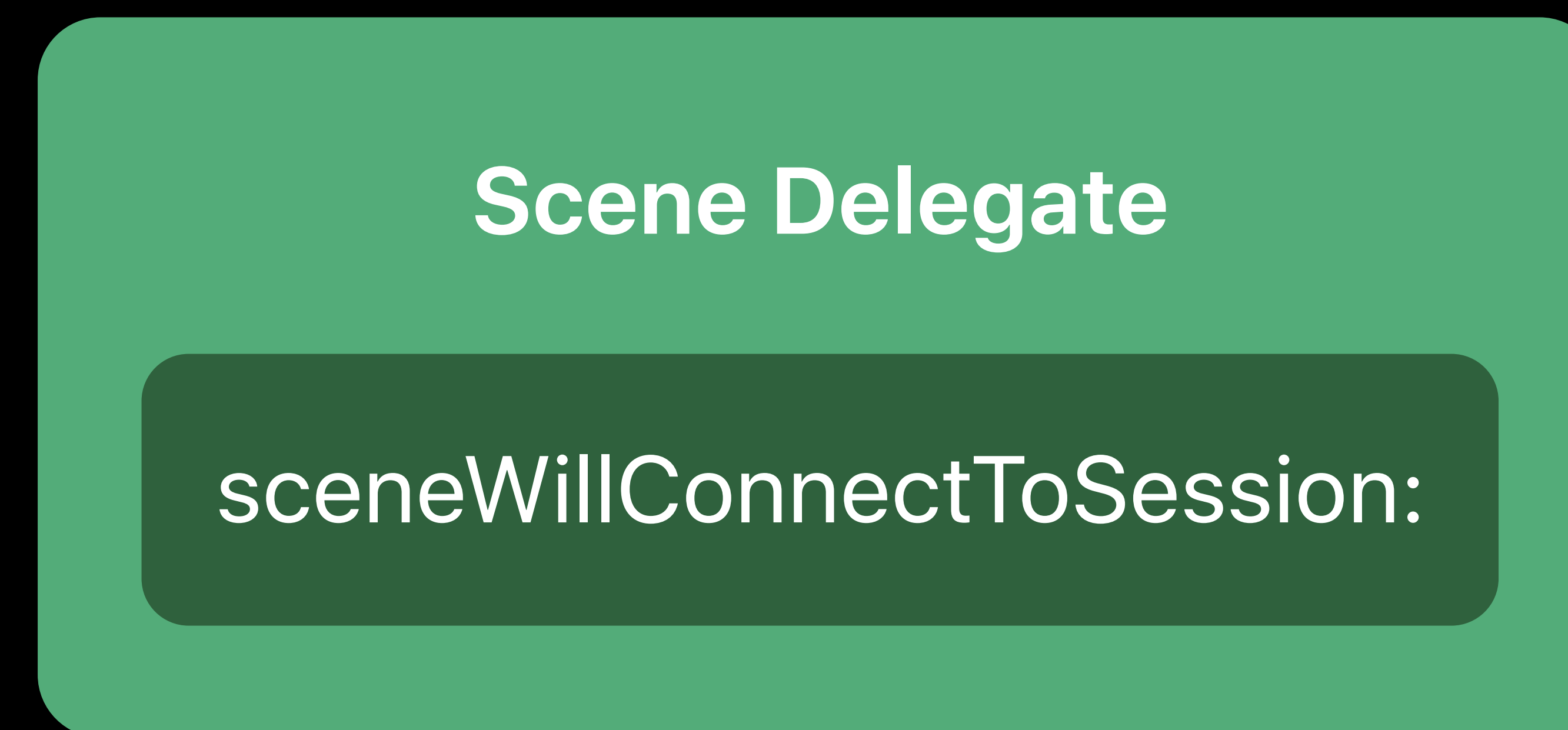
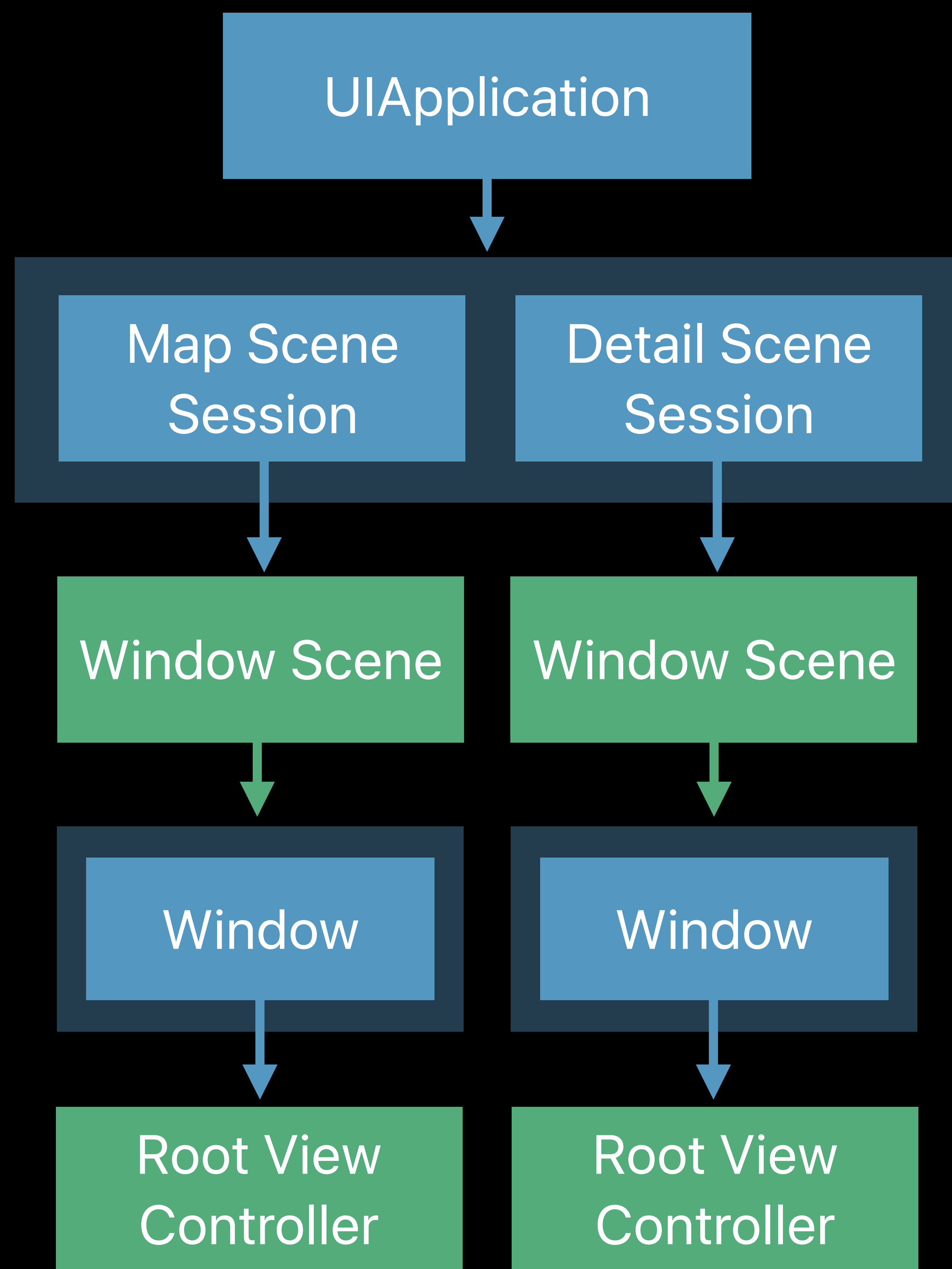
Existing session





# Session Activation

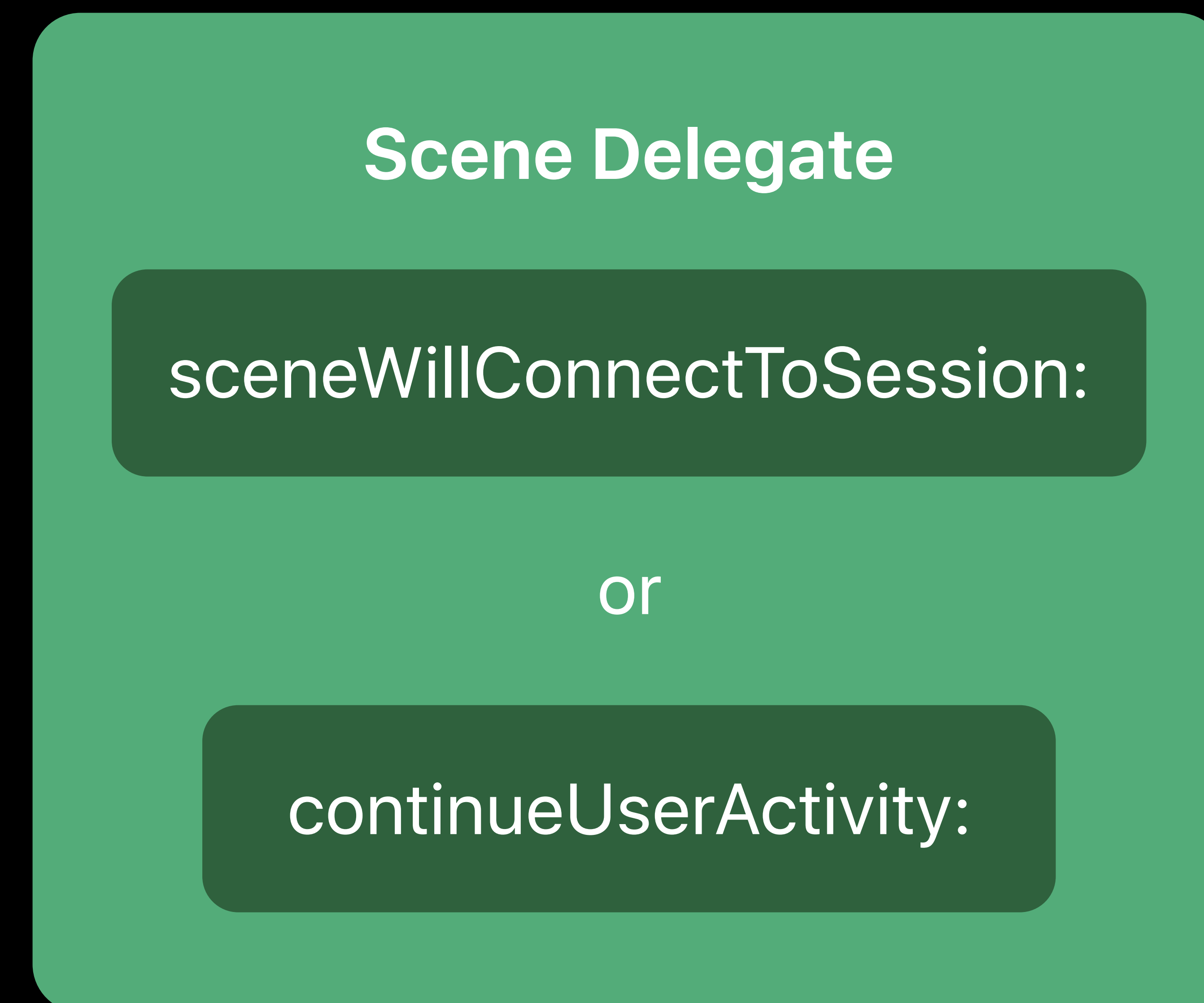
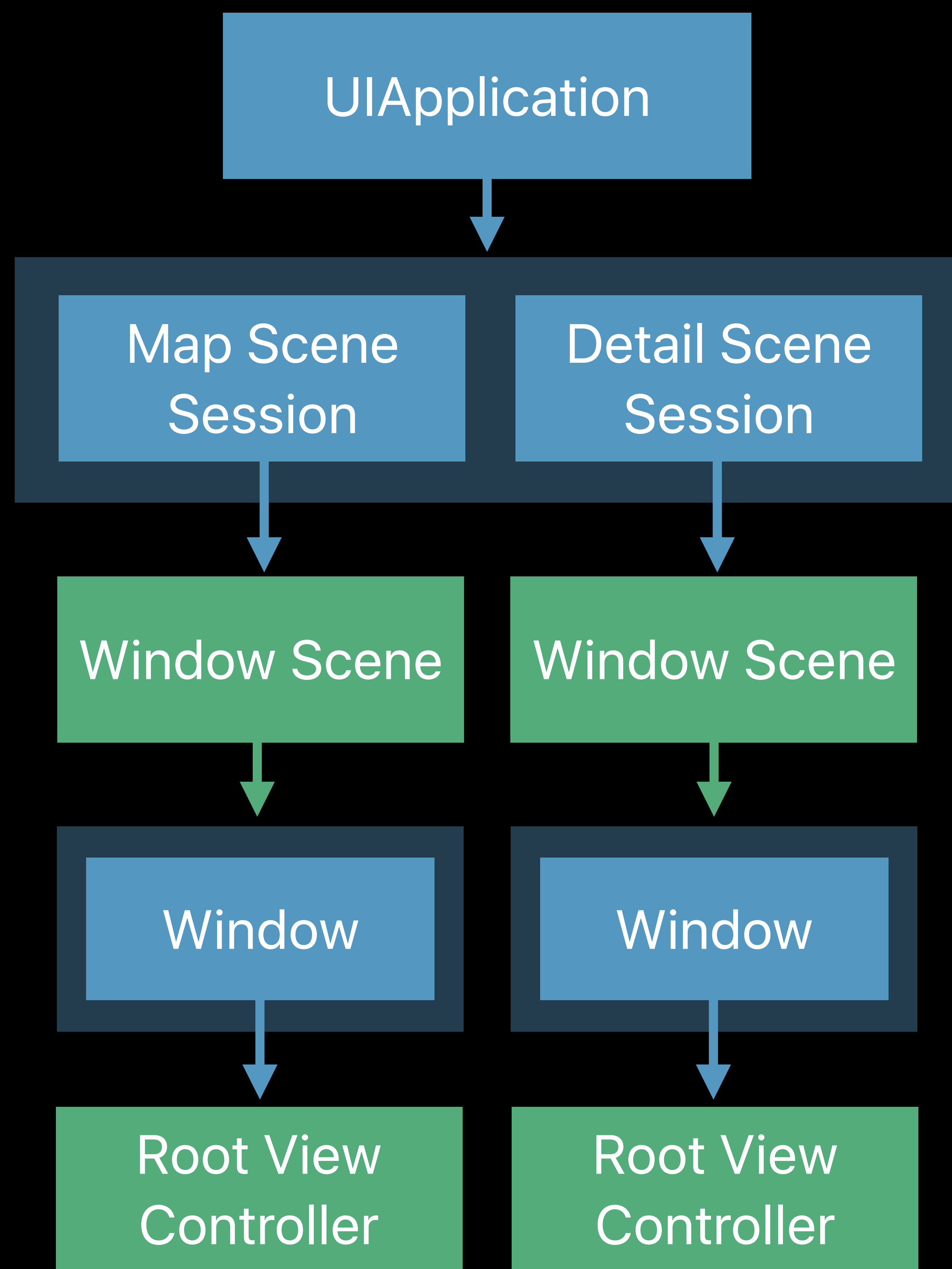
Existing session





# Session Activation

Existing session



# Session Activation Summary

Call in response to direct and local user interaction

- New or existing session

Implement App and Scene delegates

- Select a configuration
- Configure your window



# Programmatic Session Management

On behalf of your users

Activate

Refresh

Destroy

# Programmatic Session Management

On behalf of your users

Activate

Refresh

Destroy



Refresh a session for  
user-relevant updates

# Session Refresh

Updates to shared data

New user-relevant data

Session and scene hygiene



# Session Refresh

NEW

New method on `UIApplication` to refresh your scene session

- Acts on non-foreground scenes only

```
open func requestSceneSessionRefresh(_ sceneSession: UISceneSession)
```

# Session Refresh

State restoration user activity

Activation conditions

Session UI and switcher snapshot



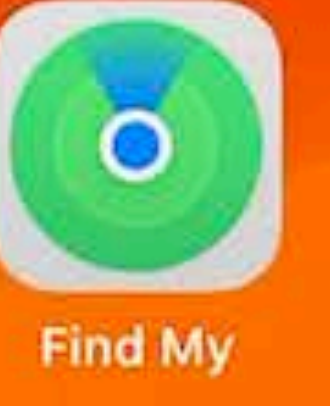
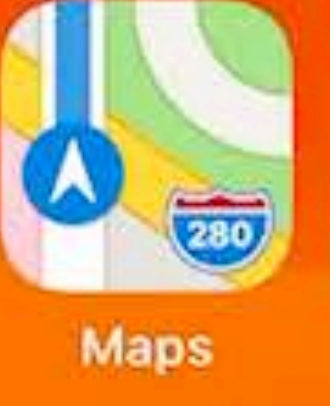
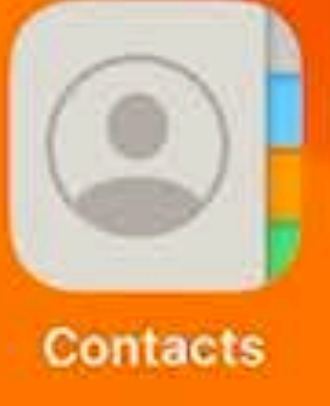
9:41  
Monday, June 3

5 UP NEXT

**Weekly Team Meeting**  
3rd floor conference room  
10:00 – 11:00 AM

SIRI APP SUGGESTIONS

Settings Mail Messages Stocks





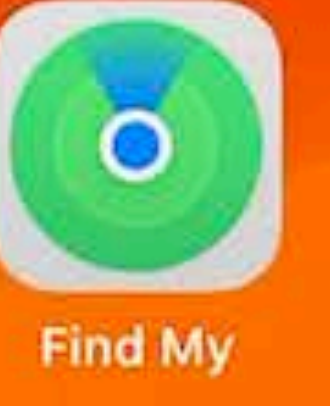
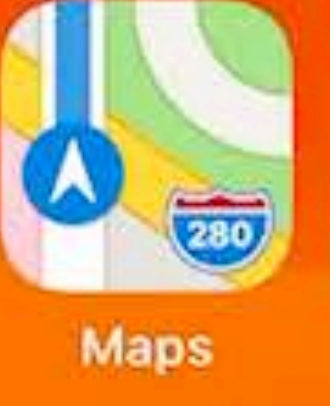
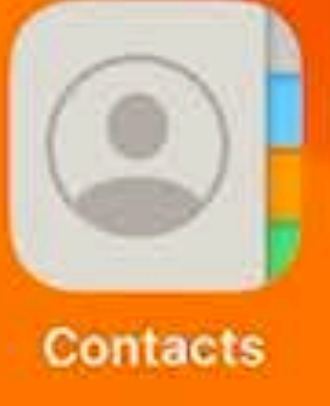
9:41  
Monday, June 3

5 UP NEXT

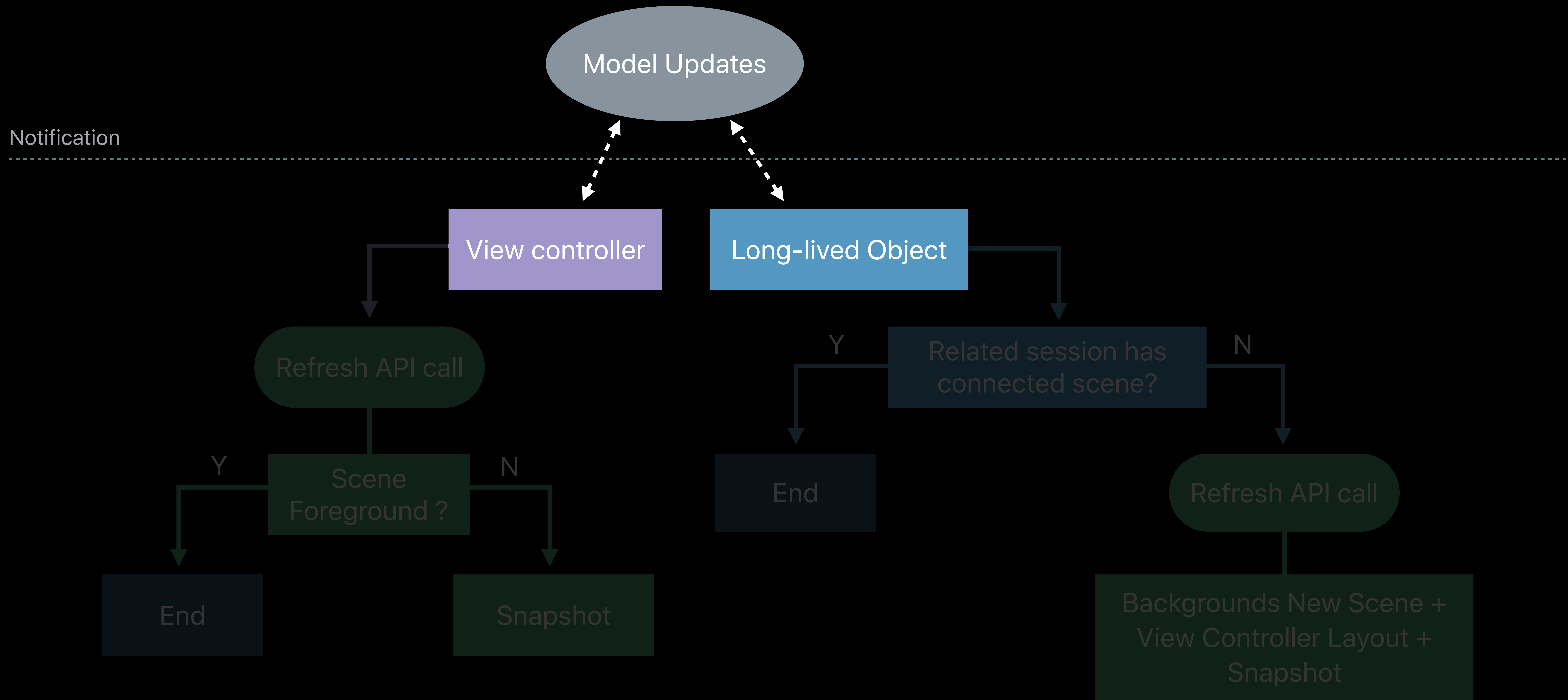
**Weekly Team Meeting**  
3rd floor conference room  
10:00 – 11:00 AM

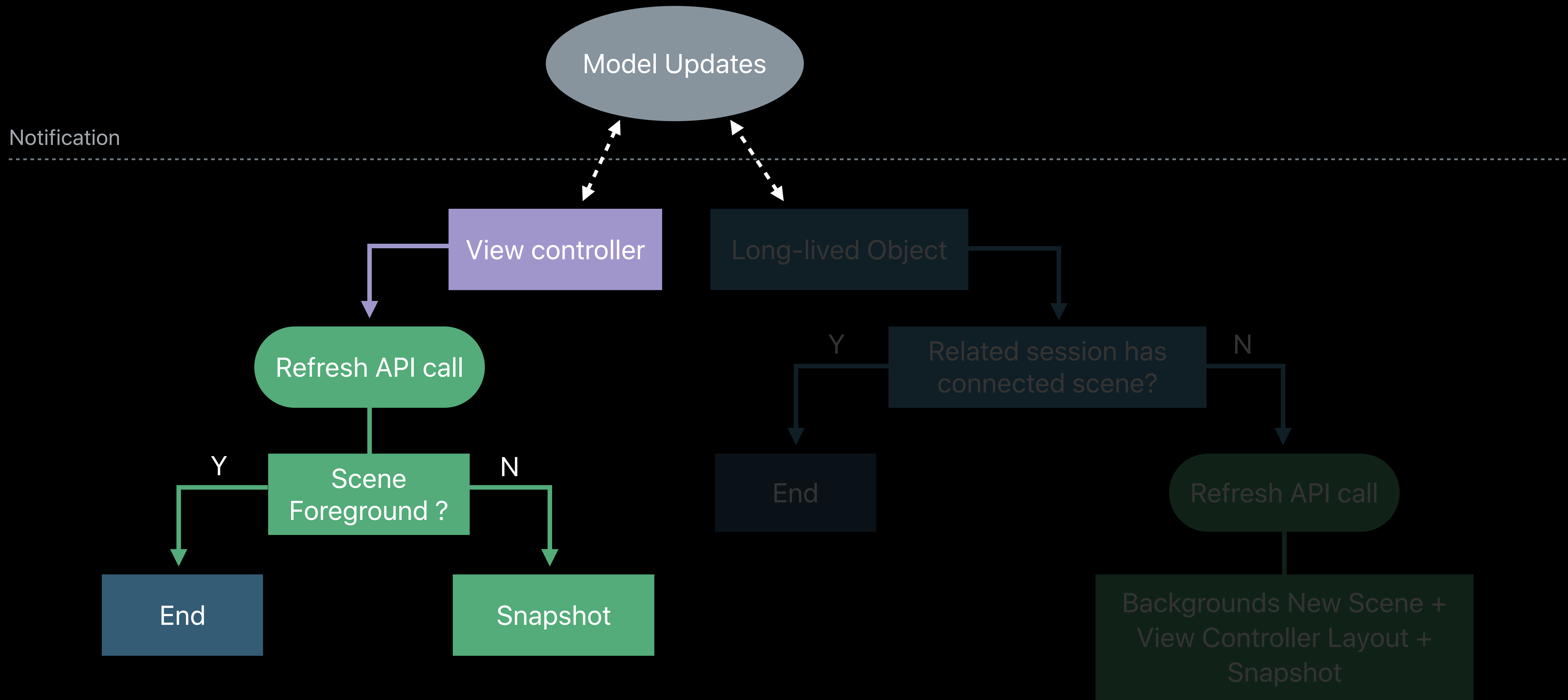
SIRI APP SUGGESTIONS

Settings Mail Messages Stocks

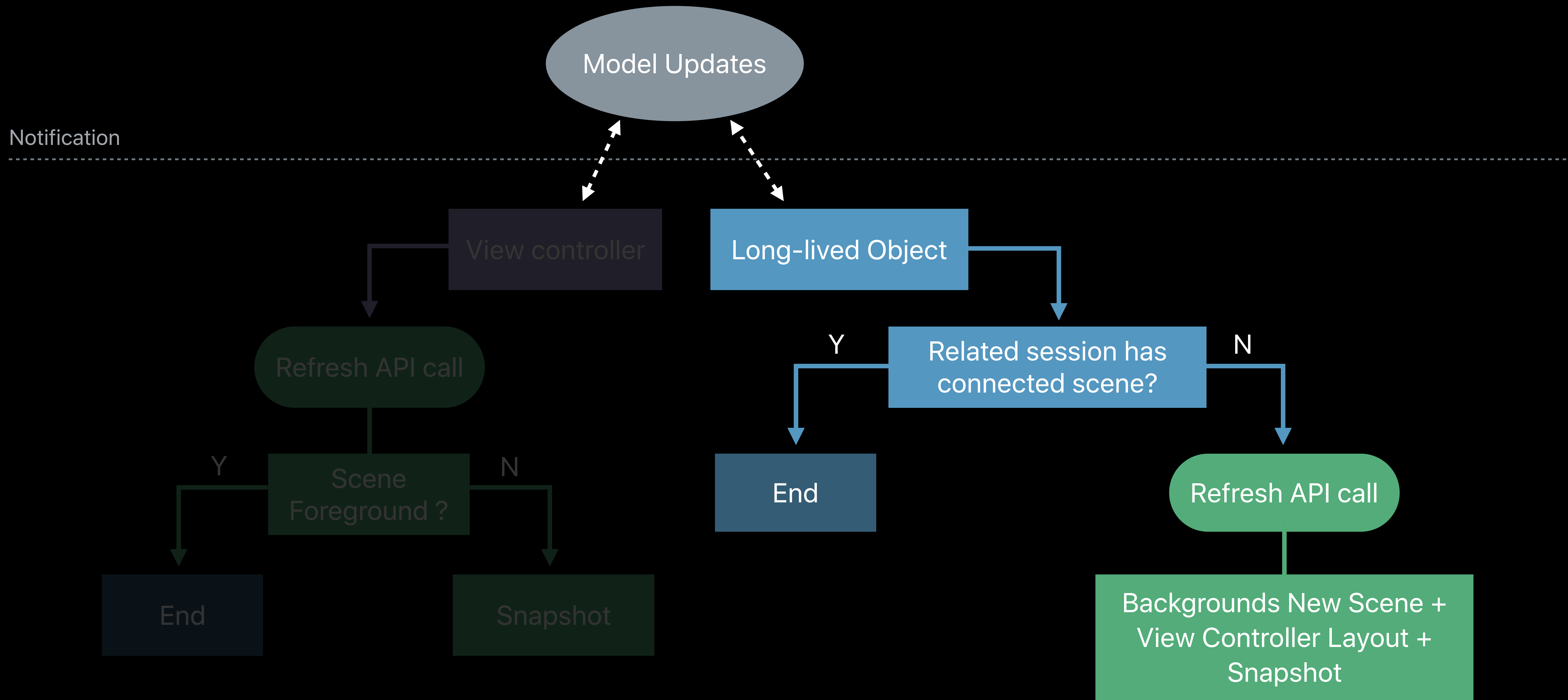












# Session Refresh Summary

For user-relevant state updates only

Optimize your layout time

The system may fulfill the request at a later point in time



# Programmatic Session Management

On behalf of your users

Activate

Refresh

Destroy

# Programmatic Session Management

On behalf of your users

Activate

Refresh

Destroy



Destroy a session only in  
response to direct user interaction

# Session Destruction

NEW

New method on `UIApplication` to destroy a session

```
open func requestSceneSessionDestruction(_ sceneSession: UISceneSession,  
                                         options: UIScene.DestructionRequestOptions?,  
                                         errorHandler: ((Error) -> Void)? = nil)
```



```
// UIWindow DestructionRequestOptions

extension UIWindowScene {

    @available(iOS 13.0, *)
    public enum DismissalAnimation : Int {
        case standard
        case commit
        case decline
    }

    @available(iOS 13.0, *)
    open class DestructionRequestOptions : UIWindowScene.DestructionRequestOptions {
        open var windowDismissalAnimation: UIWindowScene.DismissalAnimation
    }
}
```

```
// UIWindow DestructionRequestOptions

extension UIWindowScene {

    @available(iOS 13.0, *)
    public enum DismissalAnimation : Int {
        case standard
        case commit
        case decline
    }

    @available(iOS 13.0, *)
    open class DestructionRequestOptions : UIWindowScene.DestructionRequestOptions {
        open var windowDismissalAnimation: UIWindowScene.DismissalAnimation
    }
}
```



```
// UIWindow DestructionRequestOptions

extension UIWindowScene {

    @available(iOS 13.0, *)
    public enum DismissalAnimation : Int {
        case standard
        case commit
        case decline
    }

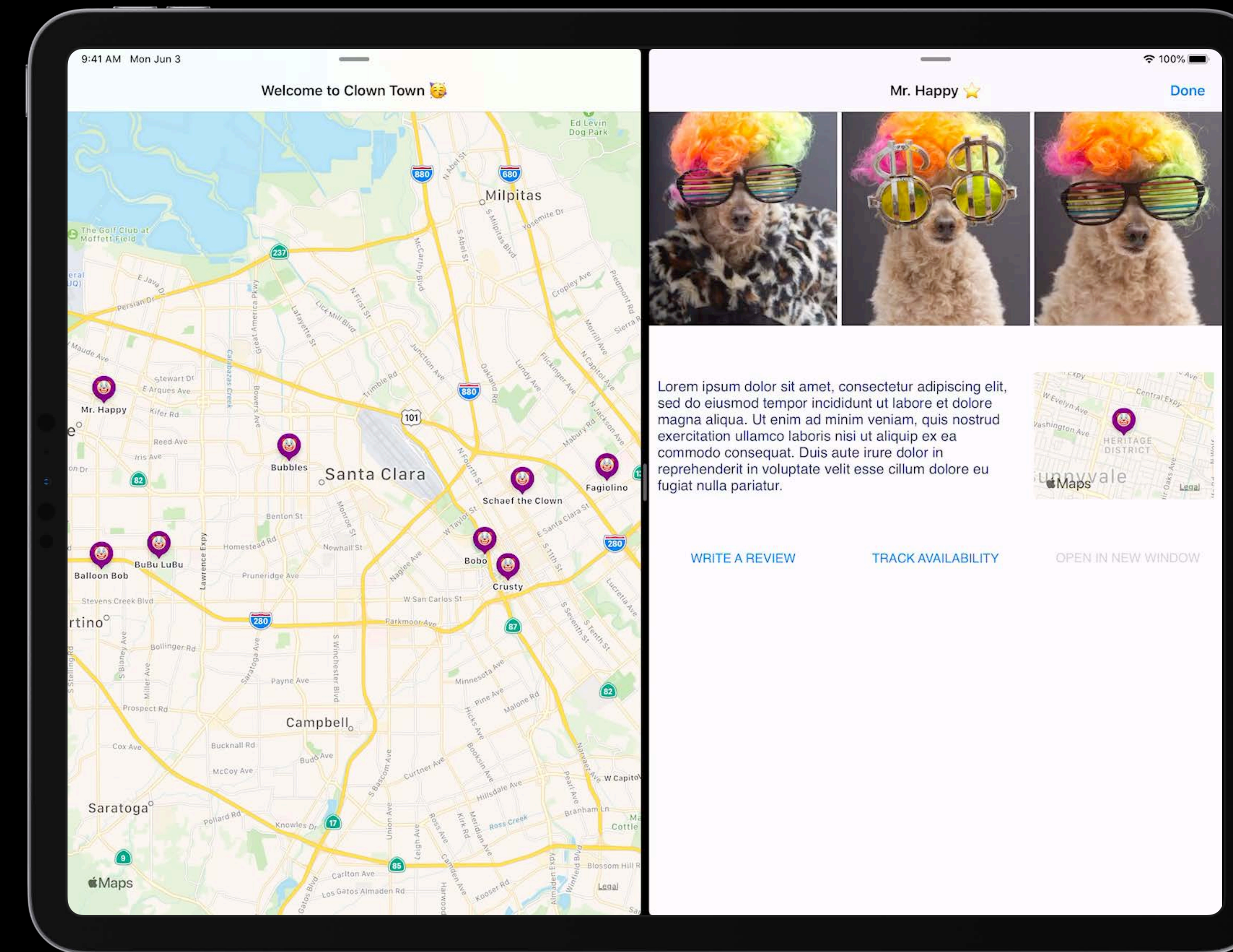
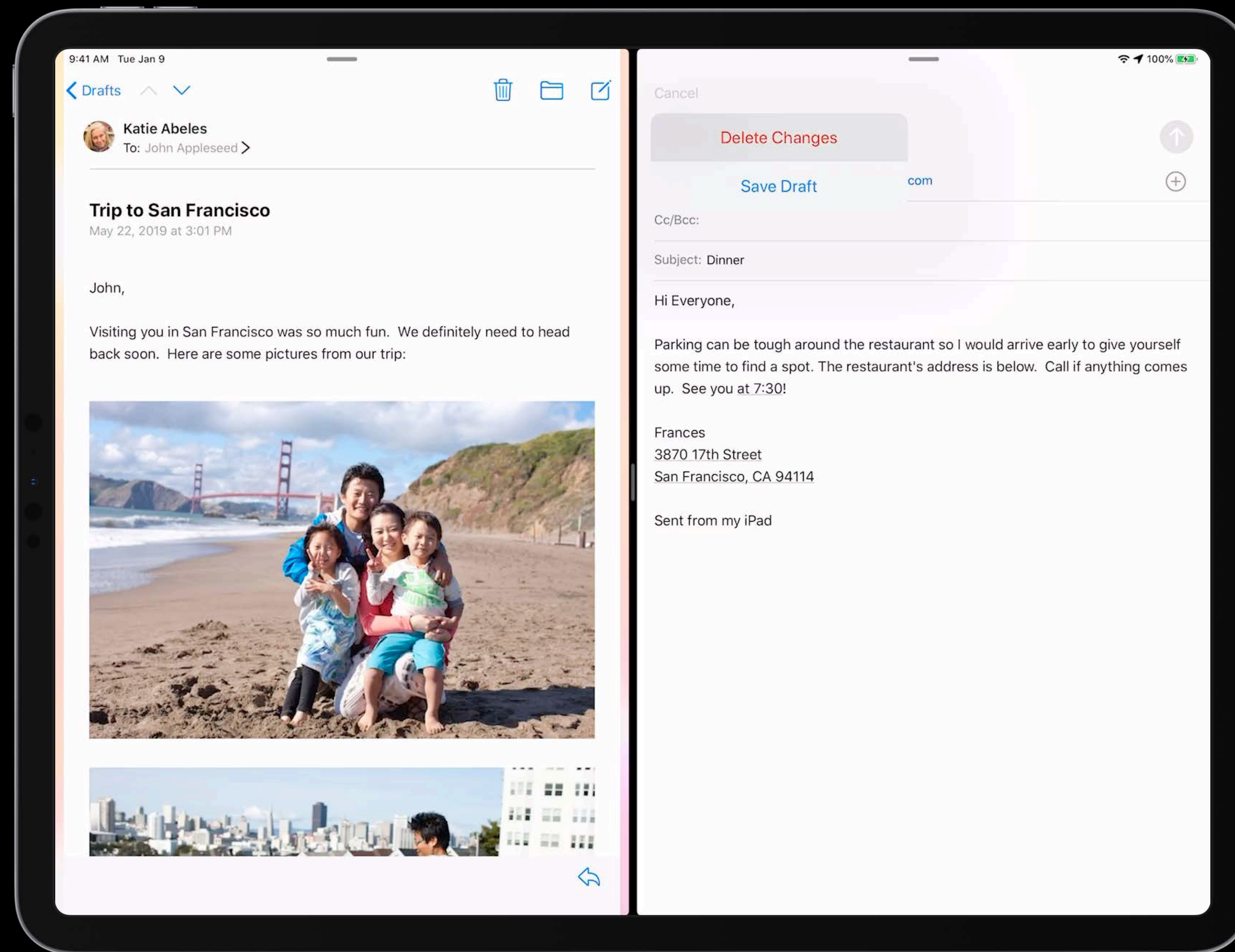
    @available(iOS 13.0, *)
    open class DestructionRequestOptions : UIWindowScene.DestructionRequestOptions {
        open var windowDismissalAnimation: UIWindowScene.DismissalAnimation
    }
}
```

Use the animation to  
acknowledge the user's intent



# Session Destruction Animations

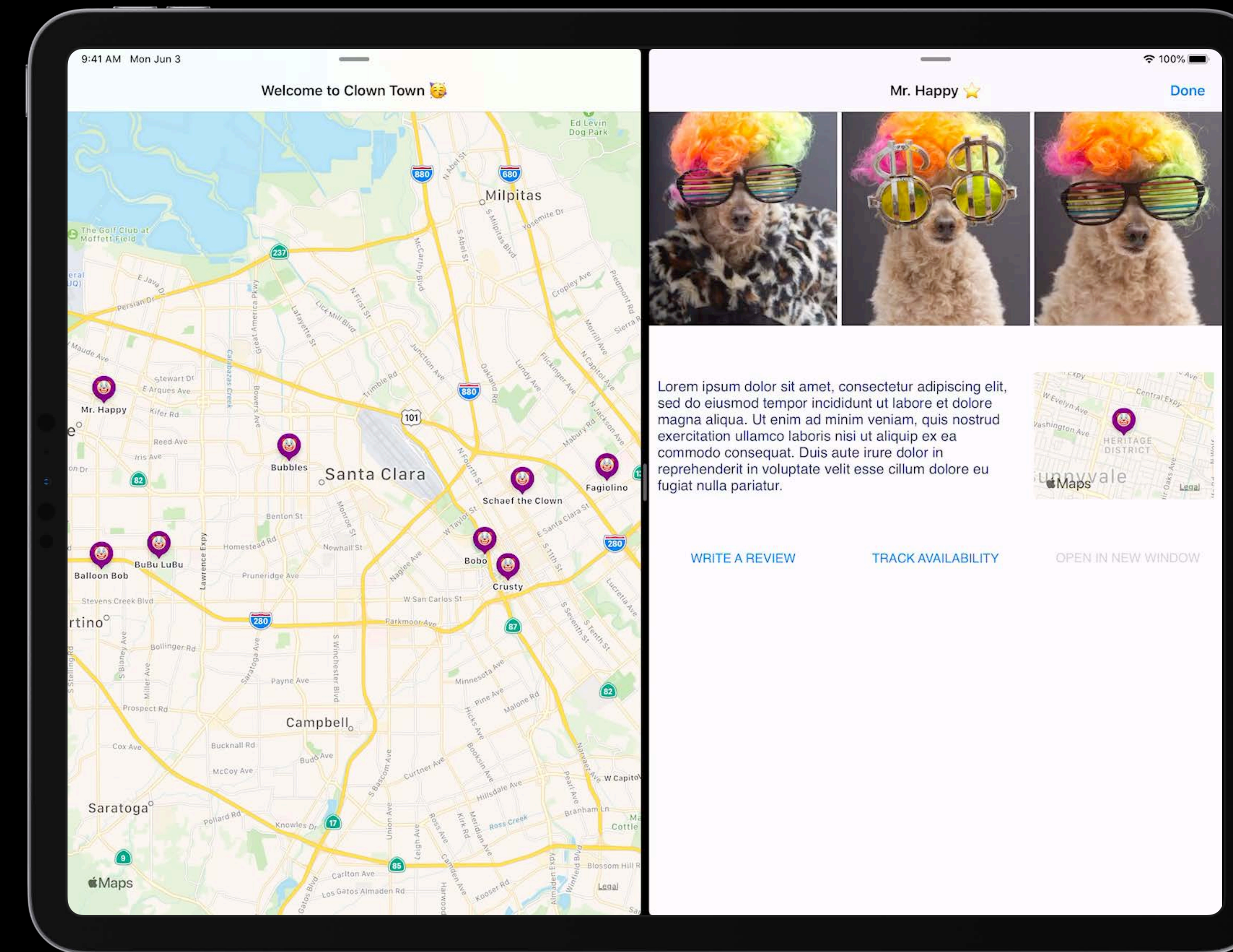
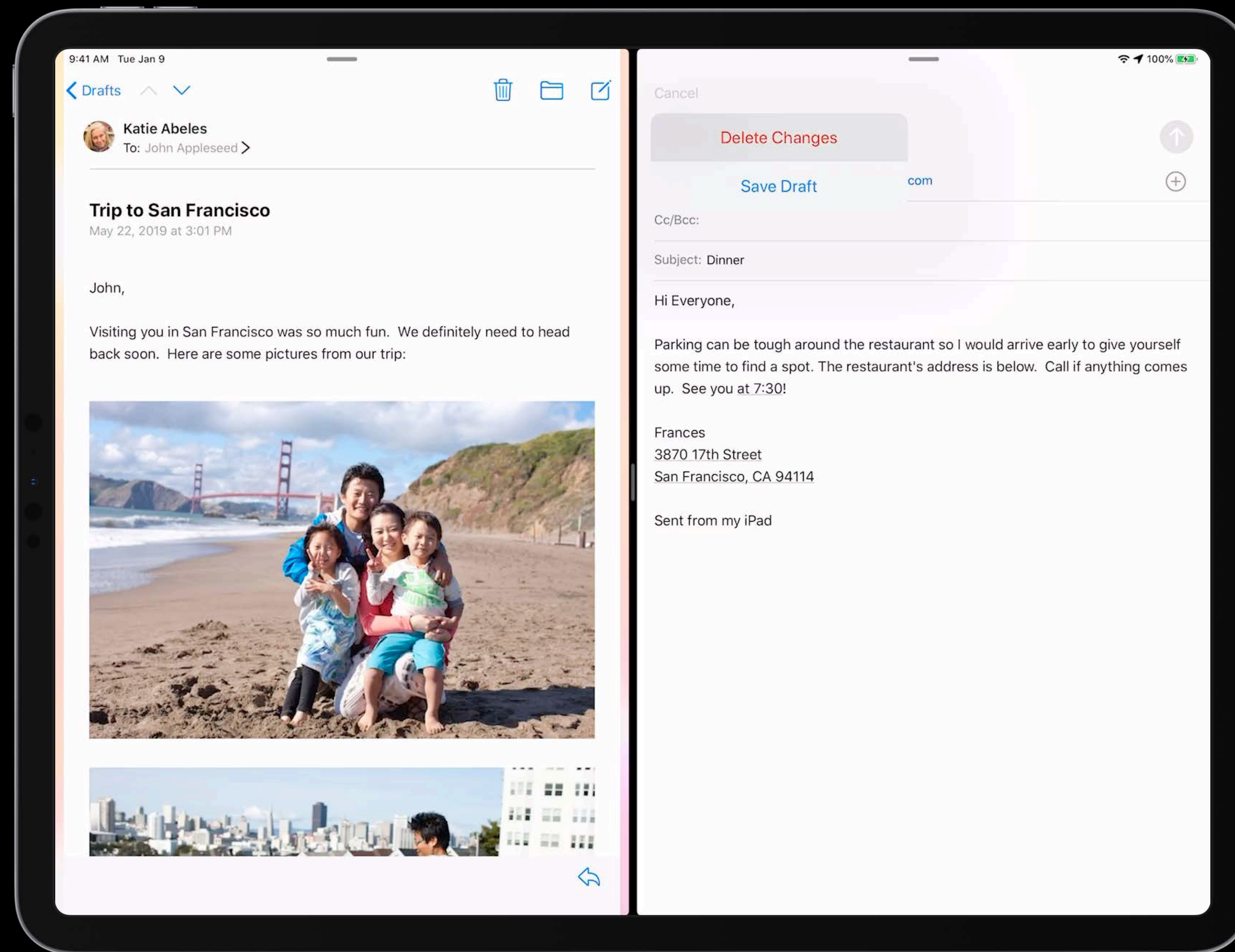
## Standard





# Session Destruction Animations

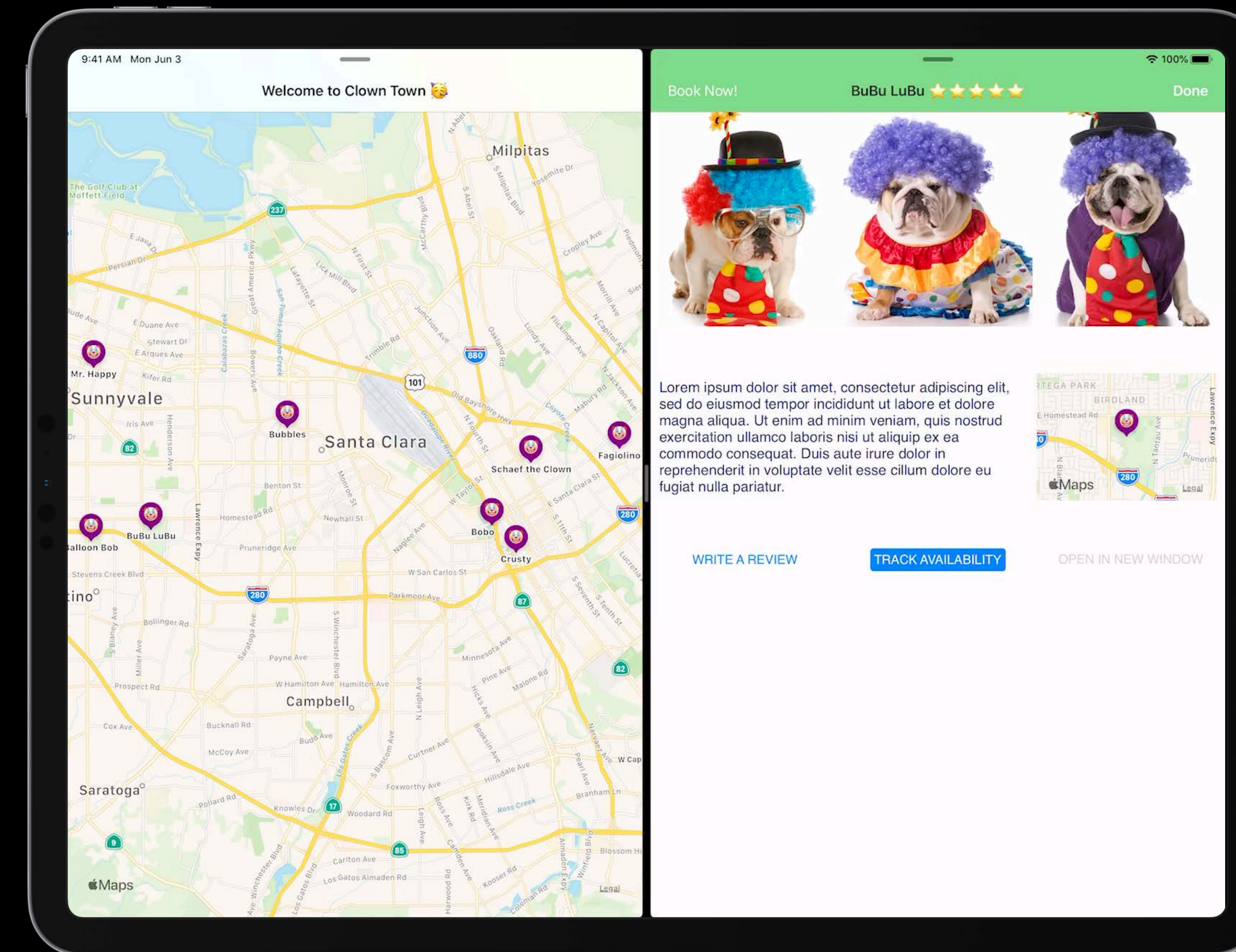
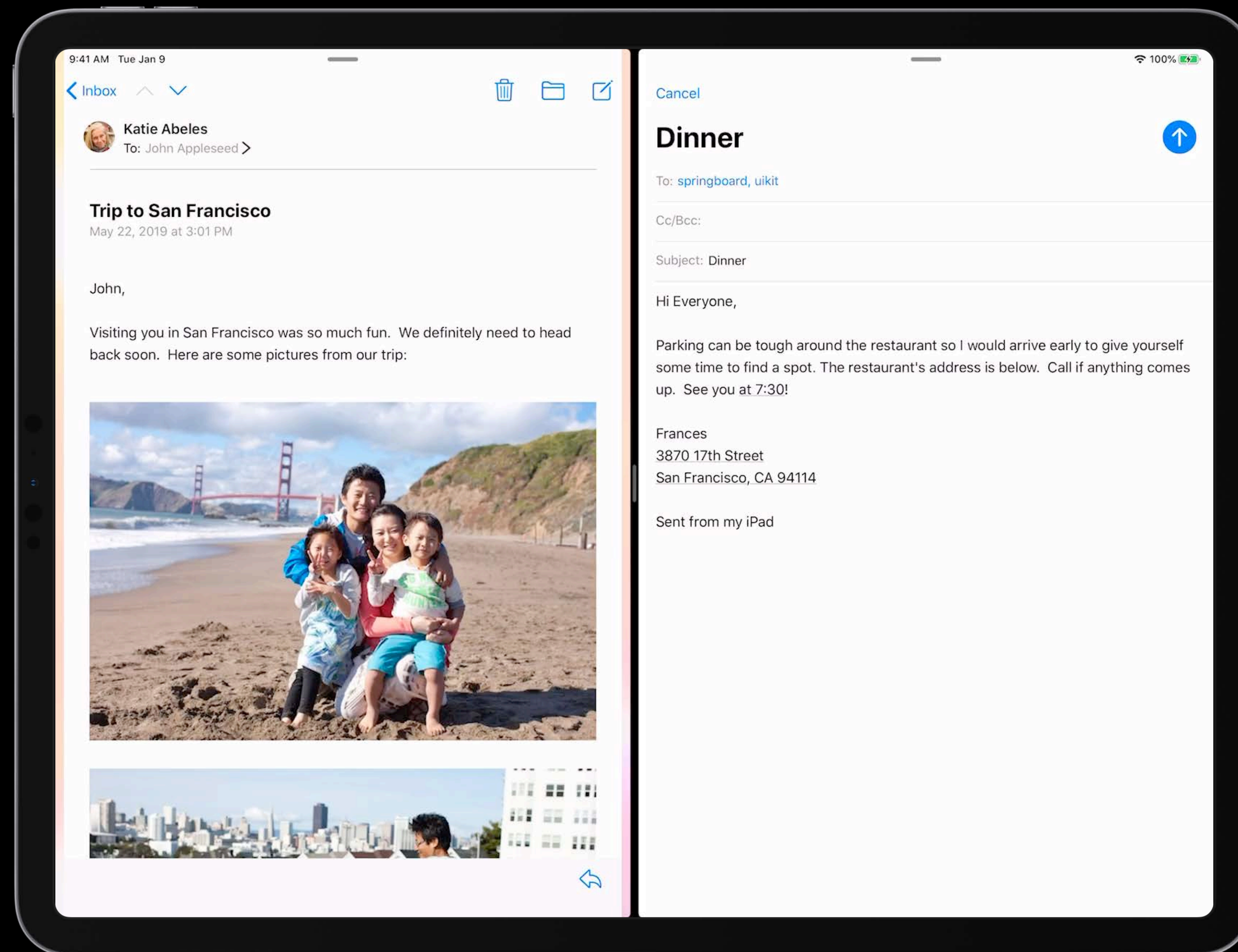
## Standard





# Session Destruction Animations

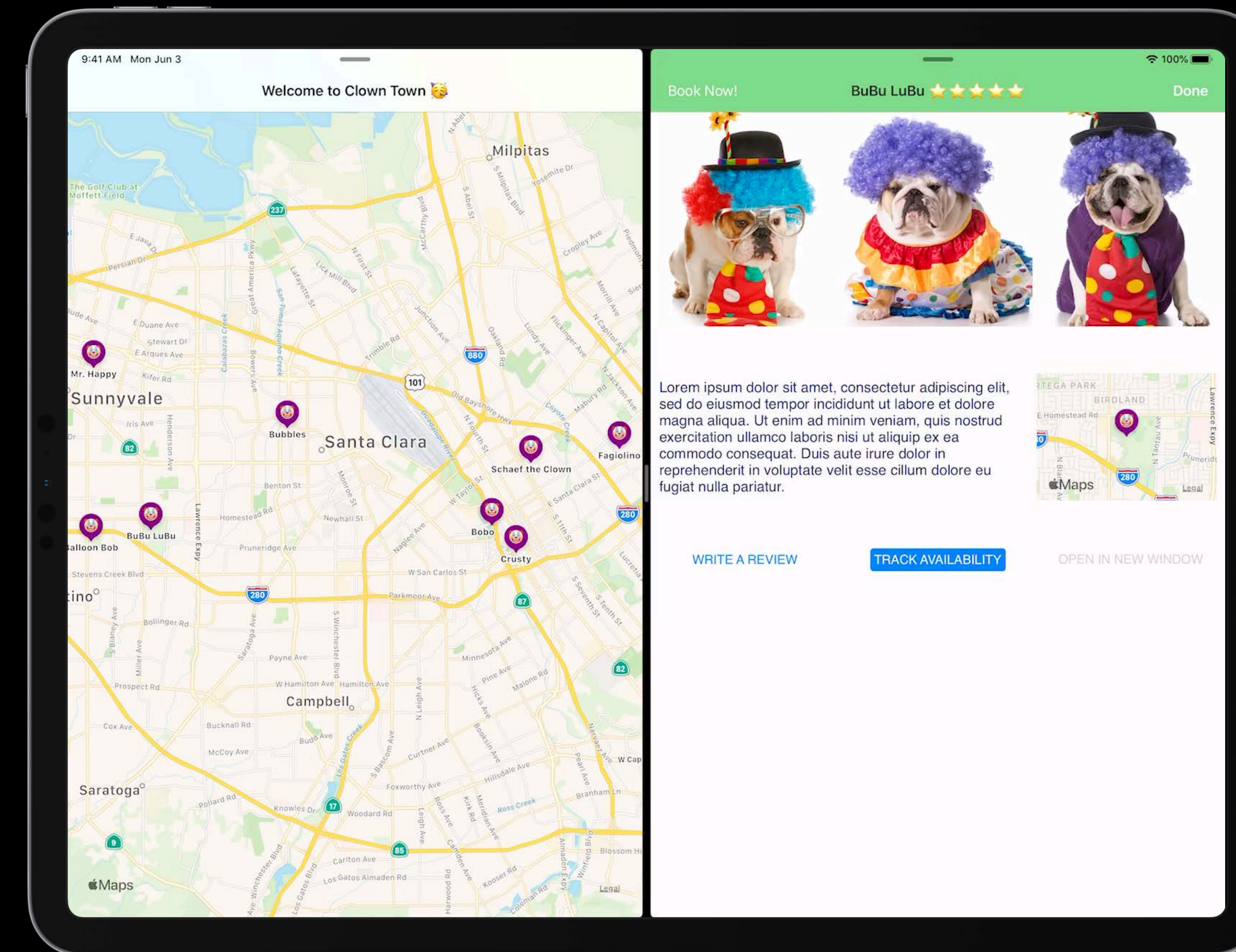
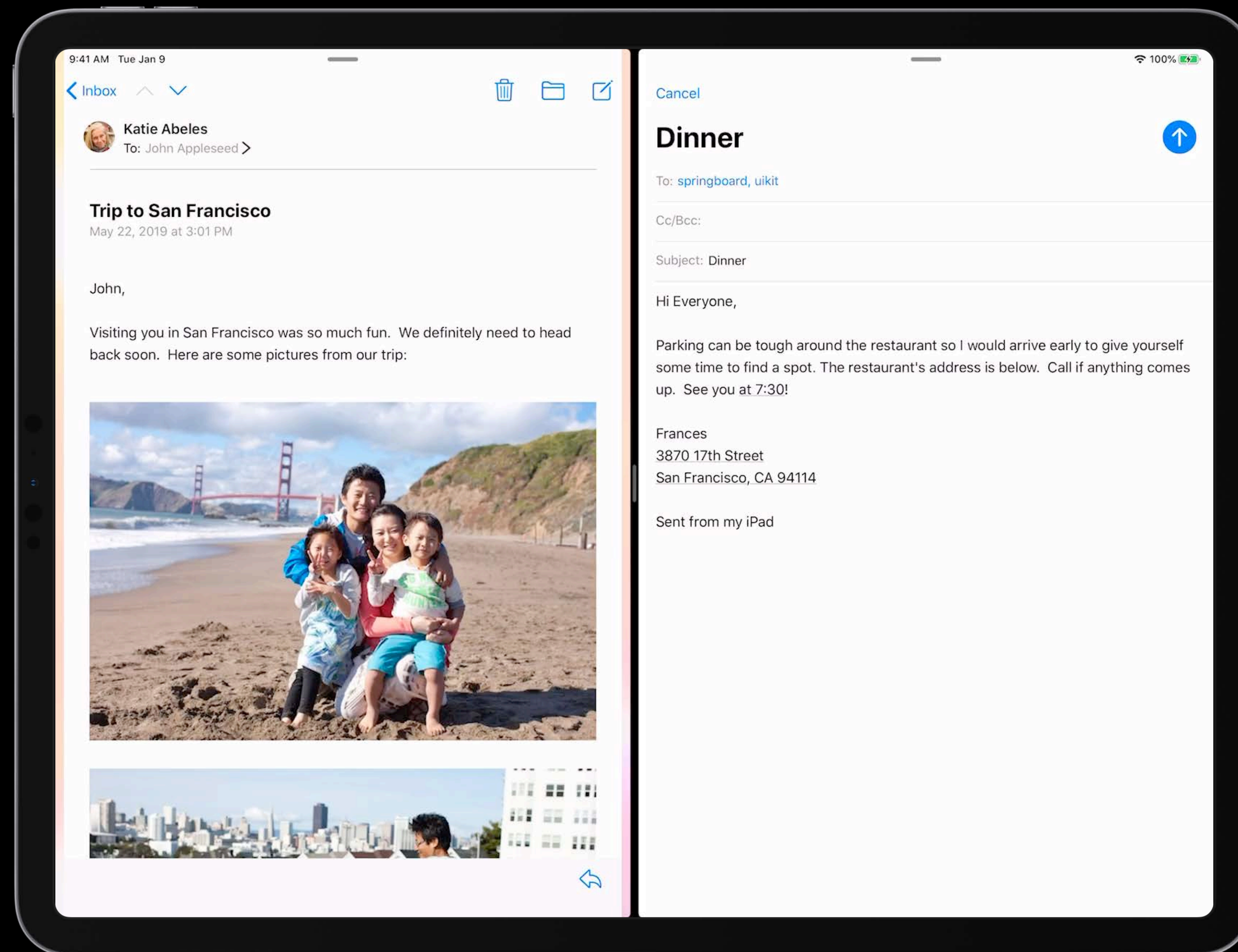
## Commit





# Session Destruction Animations

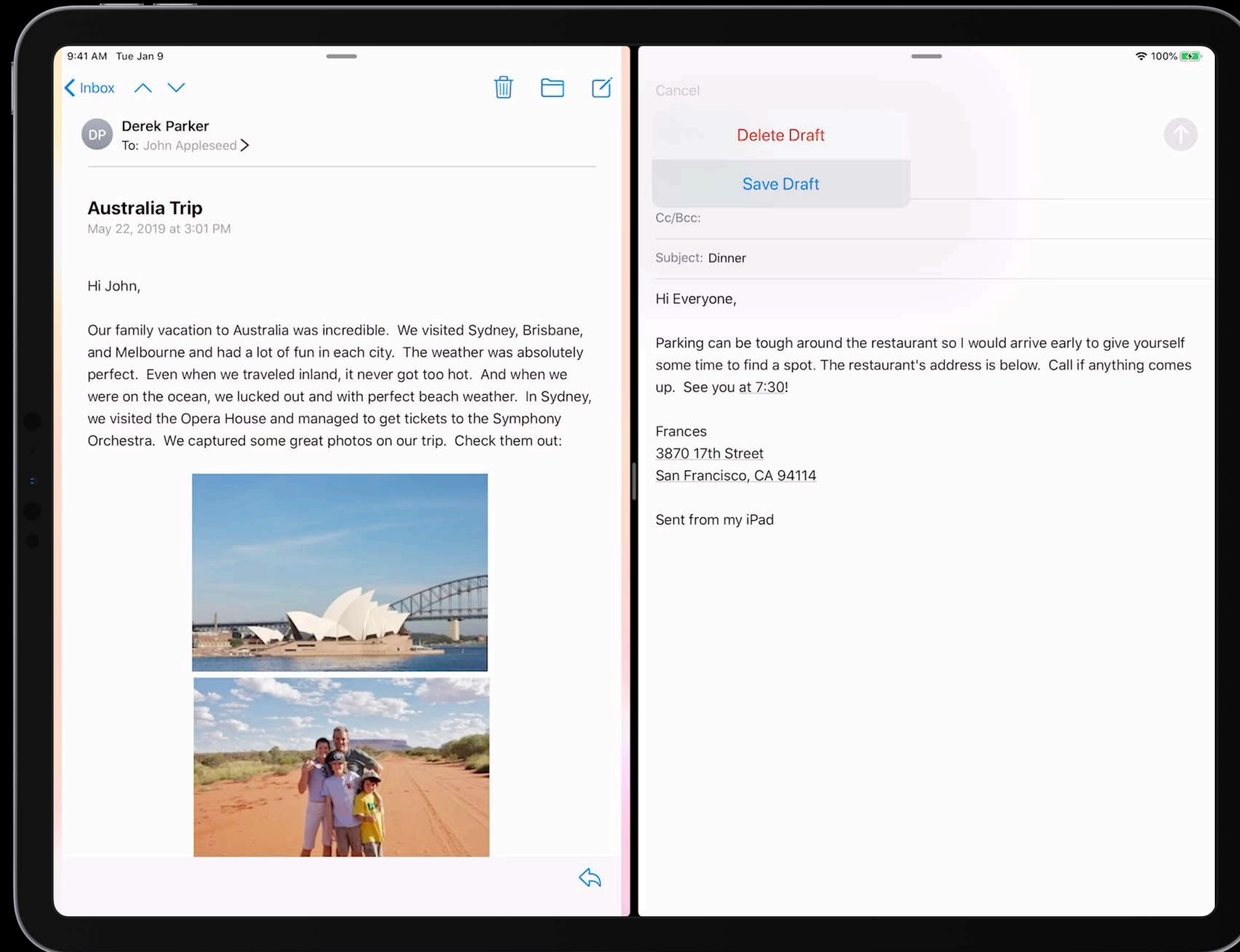
## Commit





# Session Destruction Animations

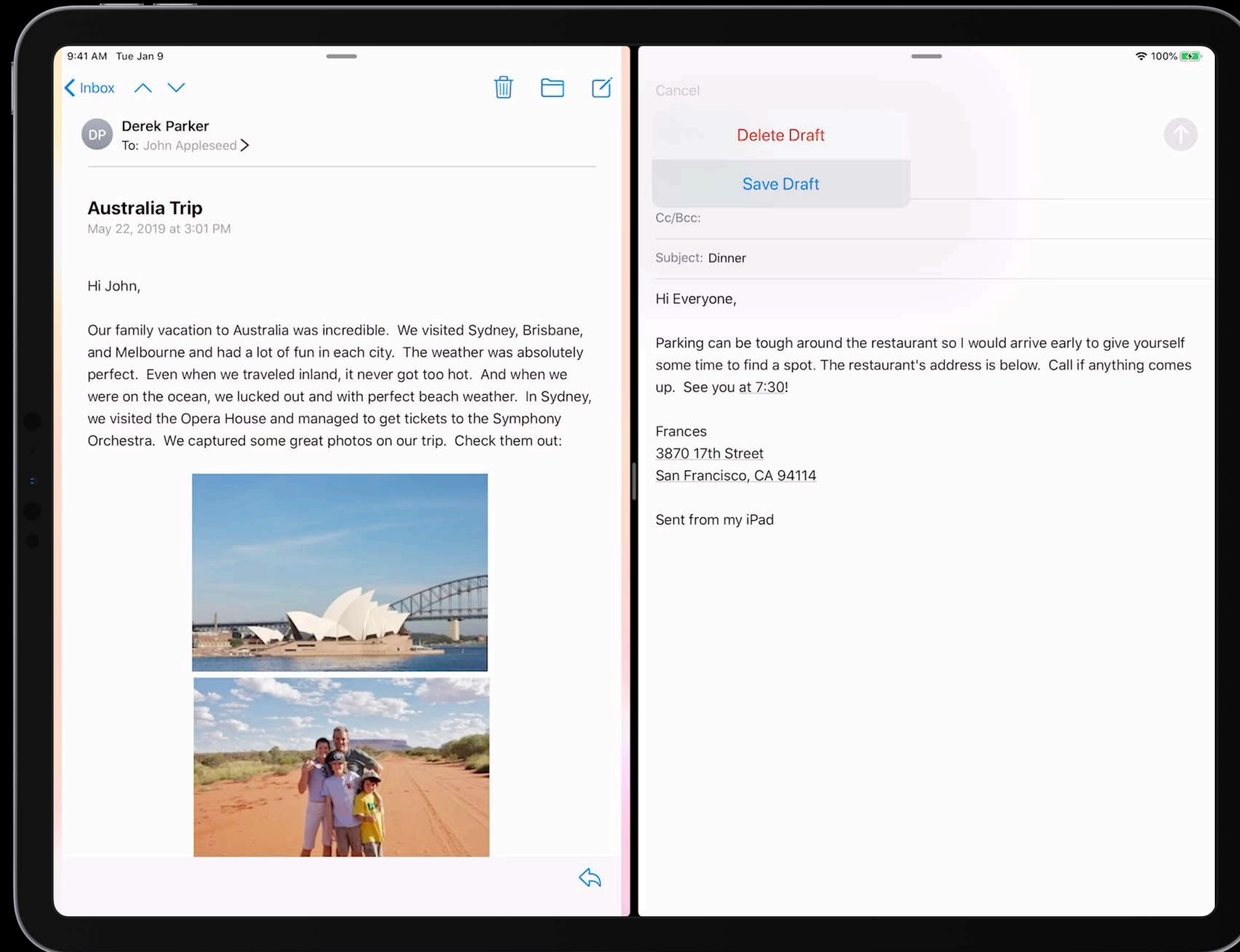
## Decline





# Session Destruction Animations

## Decline





# Session Destruction Summary

Call for direct or remote user interaction

Destroy session dedicated to user-deleted items

Pick the animation type to acknowledge user's intent

# Summary

Empower your user's interactions

Make your new features discoverable, powerful and easy-to-use

Use responsibly and have fun!



# More Information

[developer.apple.com/wwdc19/246](https://developer.apple.com/wwdc19/246)

 WWDC19