#WWDC19

Integrating SwiftUI

Tanu Singhal, UIKit
Raleigh Ledet, AppKit

Goals

Hosting SwiftUI views in your app

Embedding existing views in SwiftUI

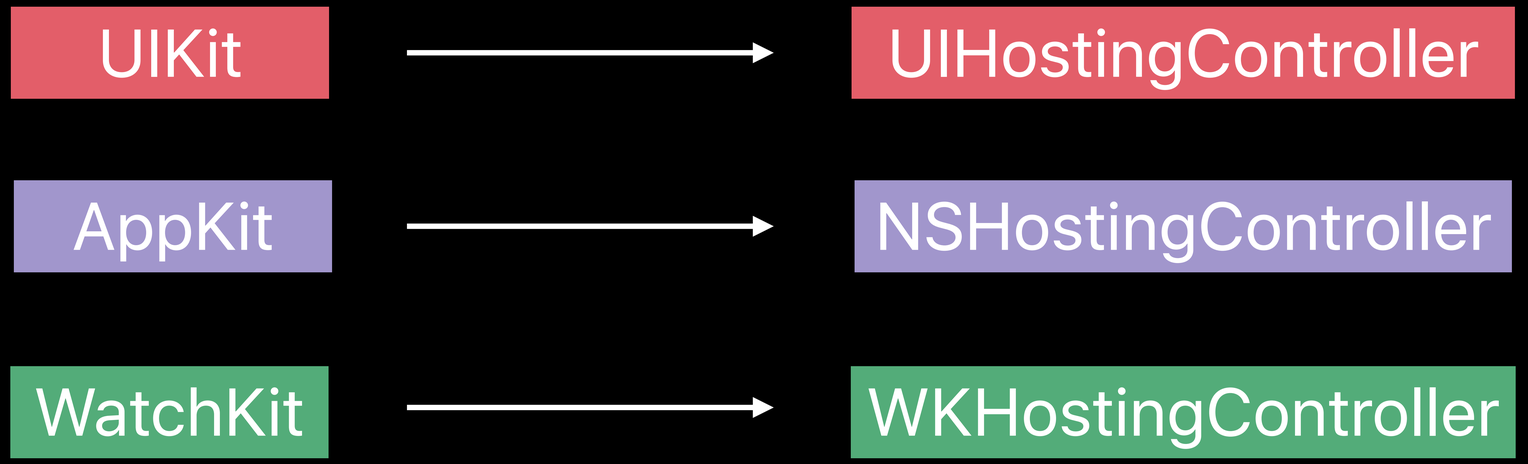
Integrating your data model

Integrating with the system

Hosting SwiftUI Views in Your App

Hosting Controller

Wraps SwiftUI hierarchy into a ViewController/InterfaceController



UIHostingController

Subclass of UIViewController

rootView property represents a SwiftUI view

```
let hostingController = UIHostingController(rootView: MyView())
```

NSHostingController

Subclass of NSViewController

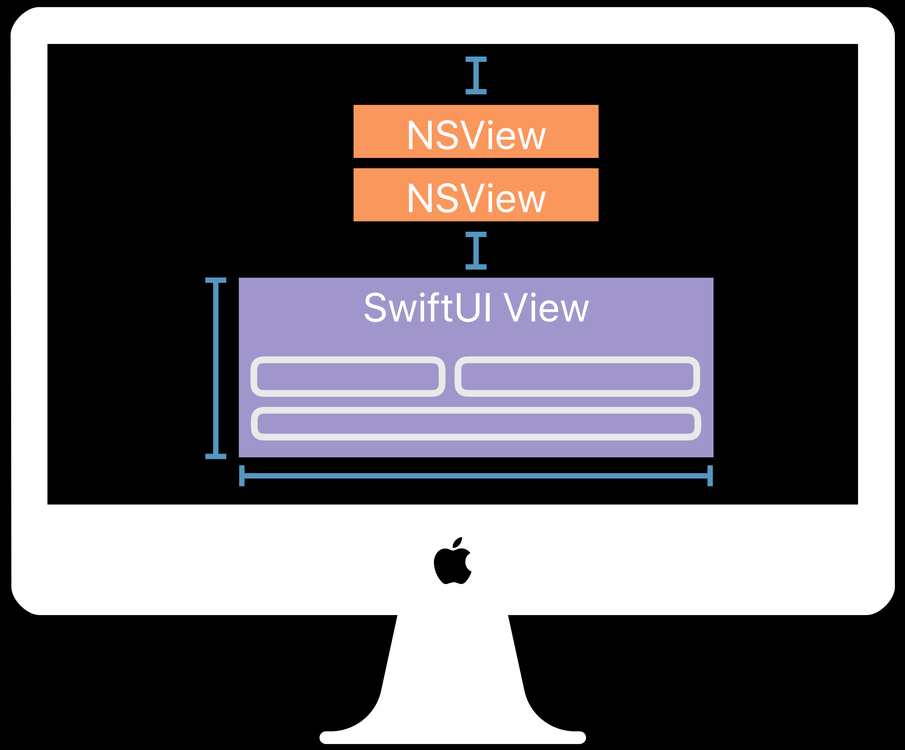
rootView property represents a SwiftUI view

```
let hostingController = NSHostingController(rootView: MyView())
```

NSHostingView

Embeds a SwiftUI view in AppKit view hierarchy

```
let hostView = NSHostingView(rootView: MyView())
```



WKHostingController

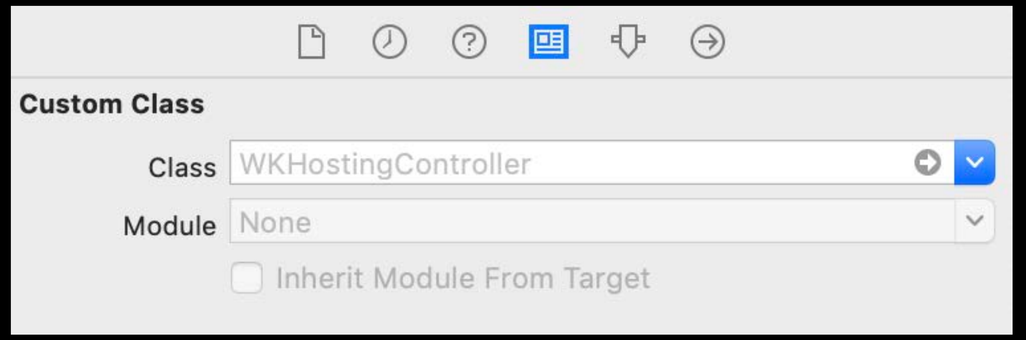
```
class MyWKHostingController: WKHostingController<MyView> {
    override var body: MyView {
        return MyView()
    }
}
```

WKHostingController

```
class MyWKHostingController: WKHostingController<MyView> {
    override var body: MyView {
        return MyView()
    }
}
```

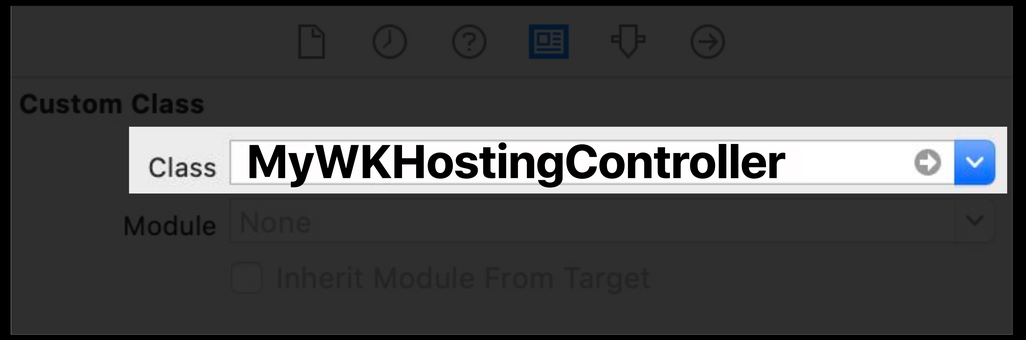
WKHostingController

```
class MyWKHostingController: WKHostingController<MyView> {
    override var body: MyView {
        return MyView()
    }
}
```



WKHostingController

```
class MyWKHostingController: WKHostingController<MyView> {
    override var body: MyView {
        return MyView()
    }
}
```



WKHostingController

```
class MyWKHostingController: WKHostingController<MyView> {
    override var body: MyView {
        return MyView()
    }
}

func setNeedsBodyUpdate()
func updateBodyIfNeeded()
```

WKUserNotificationHostingController

```
class MyNotificationController: WKUserNotificationHostingController<MyNotificationView> {
    var notification: UNNotification?

    override var body: MyNotificationView {
        return MyNotificationView(notification: notification!)
    }

    override func didReceive(_ notification: UNNotification) {
        self.notification = notification
    }
}
```

WKUserNotificationHostingController

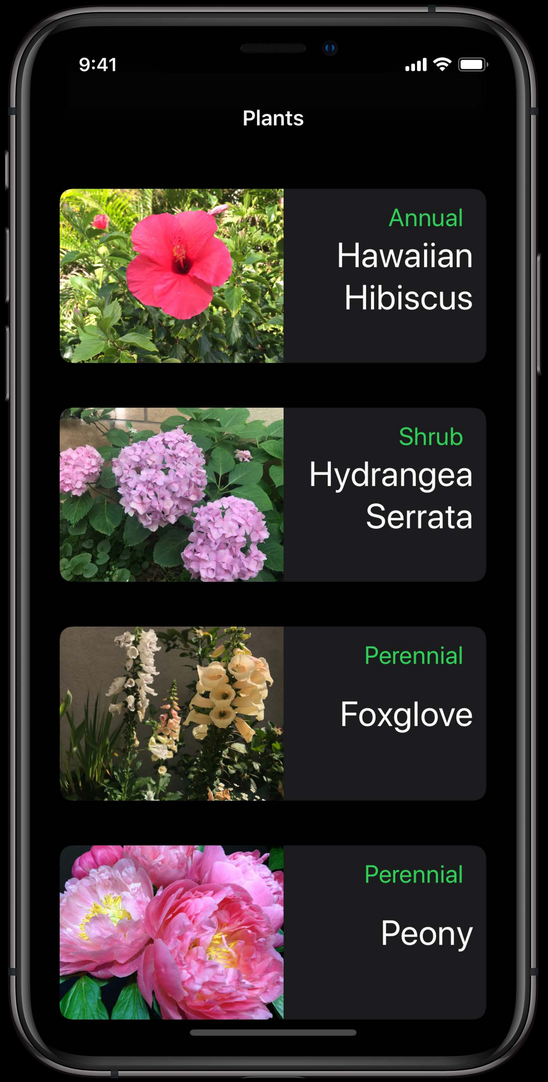
```
class MyNotificationController: WKUserNotificationHostingController<MyNotificationView> {
    var notification: UNNotification?

    override var body: MyNotificationView {
        return MyNotificationView(notification: notification!)
    }

    override func didReceive(_ notification: UNNotification) {
        self.notification = notification
    }
}
```

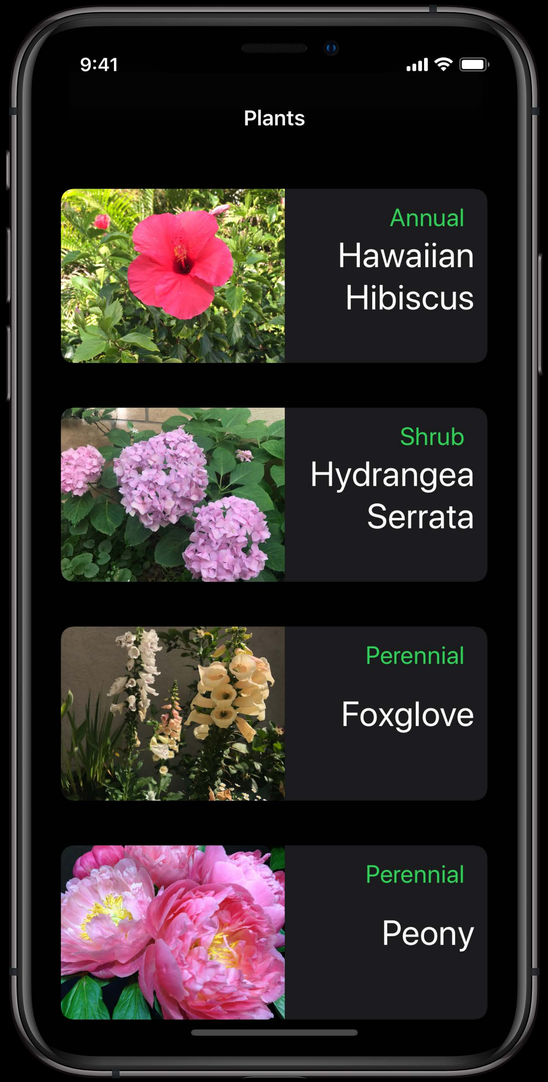
WKUserNotificationHostingController

```
class MyNotificationController: WKUserNotificationHostingController<MyNotificationView> {
    var notification: UNNotification?

    override var body: MyNotificationView {
        return MyNotificationView(notification: notification!)
    }

    override func didReceive(_ notification: UNNotification) {
        self.notification = notification
    }
}
```

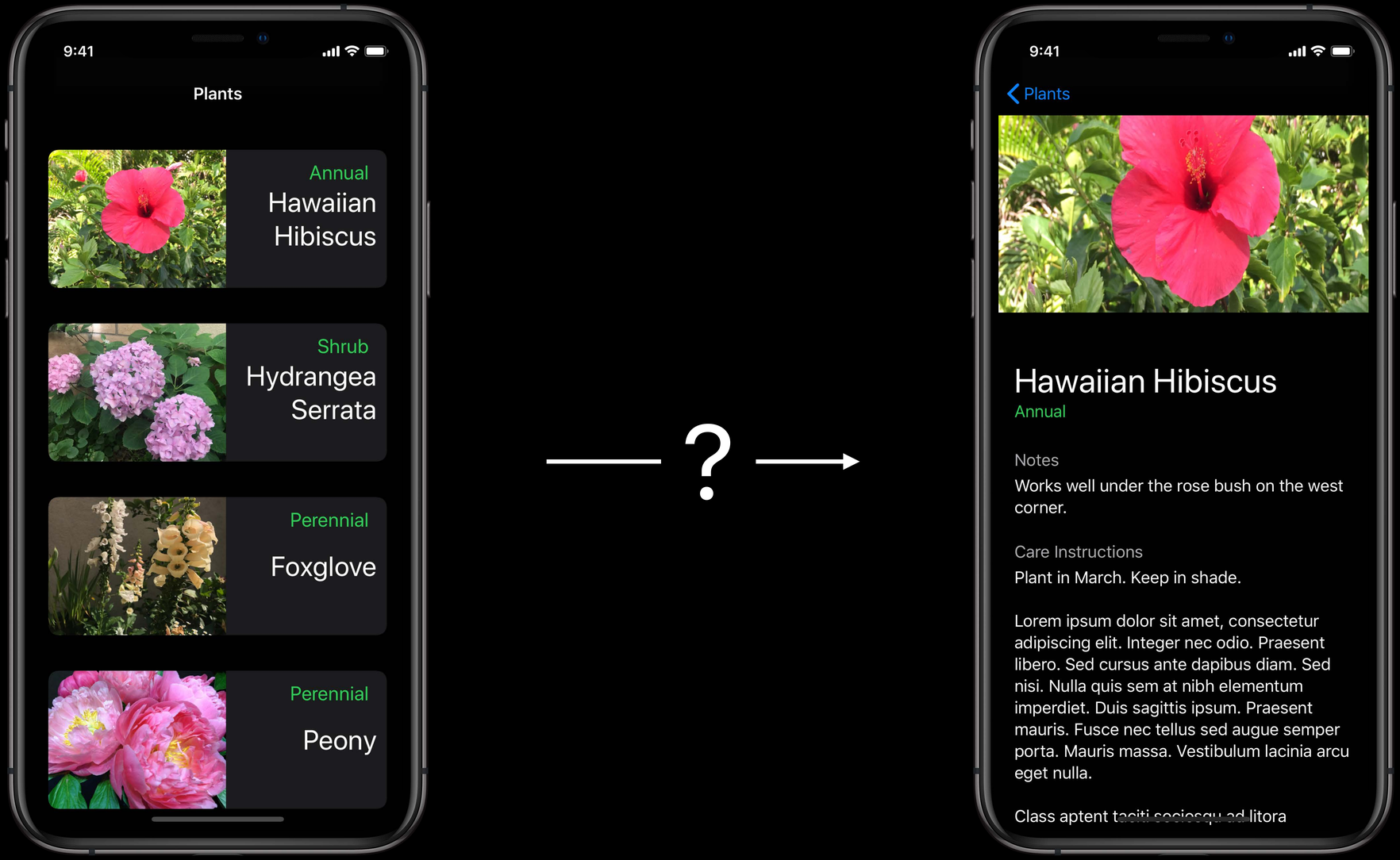
WKUserNotificationHostingController

```
class MyNotificationController: WKUserNotificationHostingController<MyNotificationView> {
    var notification: UNNotification?

    override var body: MyNotificationView {
        return MyNotificationView(notification: notification!)
    }

    override func didReceive(_ notification: UNNotification) {
        self.notification = notification
    }
}
```


9:41
Plants
Annual
Hawaiian Hibiscus
Shrub
Hydrangea Serrata
Perennial
Foxglove
Perennial
Peony


9:41
Plants
Annual
Hawaiian Hibiscus
Shrub
Hydrangea Serrata
Perennial
Foxglove
Perennial
Peony


9:41
Plants
Hawaiian Hibiscus
Annual
Notes
Works well under the rose bush on the west corner.
Care Instructions
Plant in March. Keep in shade.
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer nec odio. Praesent libero. Sed cursus ante dapibus diam. Sed nisi. Nulla quis sem at nibh elementum imperdiet. Duis sagittis ipsum. Praesent mauris. Fusce nec tellus sed augue semper porta. Mauris massa. Vestibulum lacinia arcu eget nulla.
Class aptent taciti sociosqu ad litora


9:41
Plants
Annual
Hawaiian Hibiscus
Shrub
Hydrangea Serrata
Perennial
Foxglove
Perennial
Peony
?
9:41
Plants
Hawaiian Hibiscus
Annual
Notes
Works well under the rose bush on the west corner.
Care Instructions
Plant in March. Keep in shade.
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer nec odio. Praesent libero. Sed cursus ante dapibus diam. Sed nisi. Nulla quis sem at nibh elementum imperdiet. Duis sagittis ipsum. Praesent mauris. Fusce nec tellus sed augue semper porta. Mauris massa. Vestibulum lacinia arcu eget nulla.
Class aptent taciti sociosqu ad litora

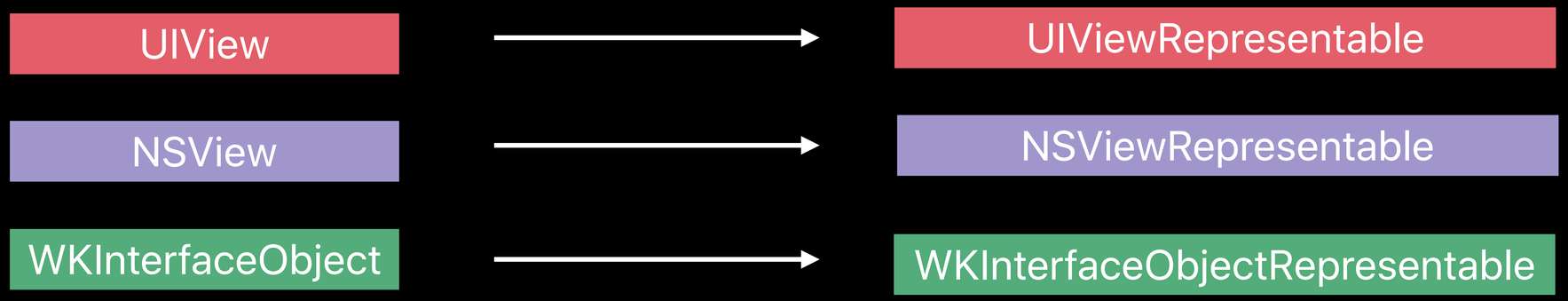
Demo

Hosting SwiftUI views in your app

Embedding Existing Views in SwiftUI

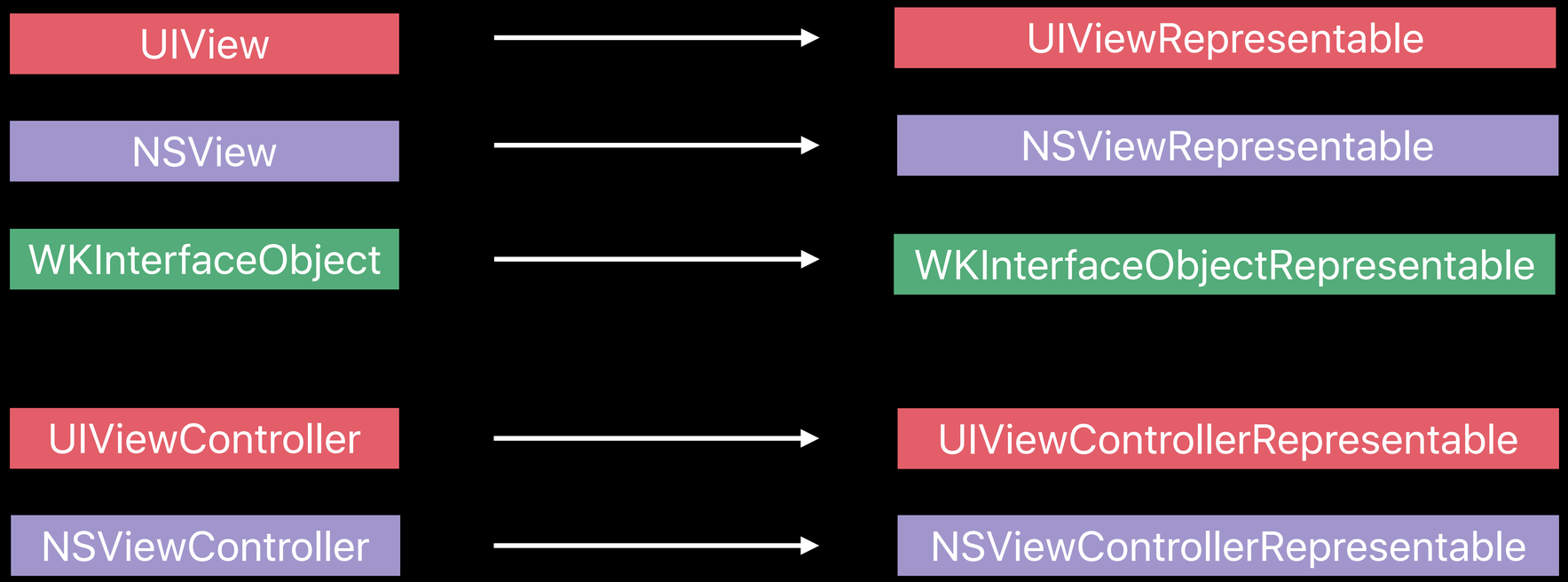
Representable Protocol

Wraps UIKit/AppKit/WatchKit content into SwiftUI



Representable Protocol

Wraps UIKit/AppKit/WatchKit content into SwiftUI



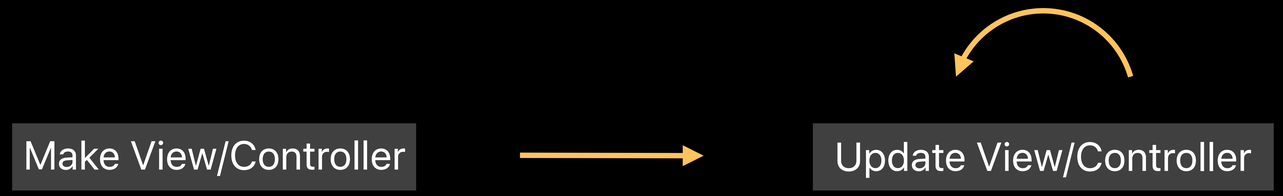
Representable Protocol

Make View/Controller

Update View/Controller

Representable Protocol

Representable Protocol



Representable Protocol


UIViewRepresentable
makeUIView(context:)
updateUIView(_:context:)
dismantleUIView(_:coordinator:)
UIViewControllerRepresentable
makeUIViewController(context:)
updateUIViewController(_:context:)
dismantleUIViewController(_:coordinator:)
NSViewRepresentable
makeNSView(context:)
updateNSView(_:context:)
dismantleNSView(_:coordinator:)
NSViewControllerRepresentable
makeNSViewController(context:)
updateNSViewController(_:context:)
dismantleNSViewController(_:coordinator:)
WKInterfaceObjectRepresentable
makeWKInterfaceObject(context:)
updateWKInterfaceObject(_:context:)
dismantleWKInterfaceObject(_:coordinator:)

Advanced Integration of Views

Target/action or delegation

React to the environment

Animate with SwiftUI

Representable Context

Coordinator

Environment

Transaction

UIViewRepresentableContext

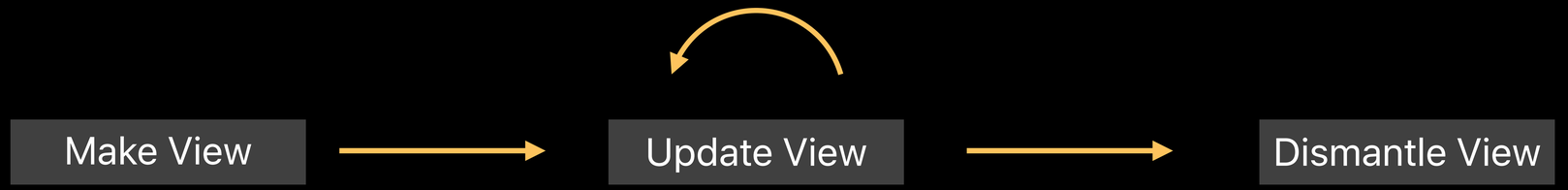
NSViewRepresentableContext

WKInterfaceObjectRepresentableContext

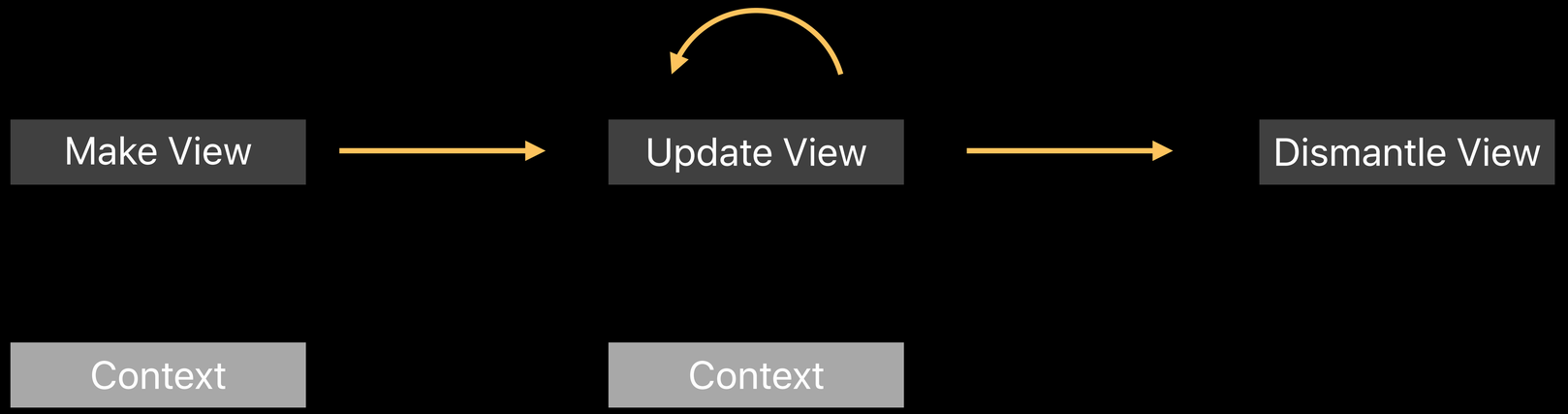
UIViewControllerRepresentableContext

NSViewControllerRepresentableContext

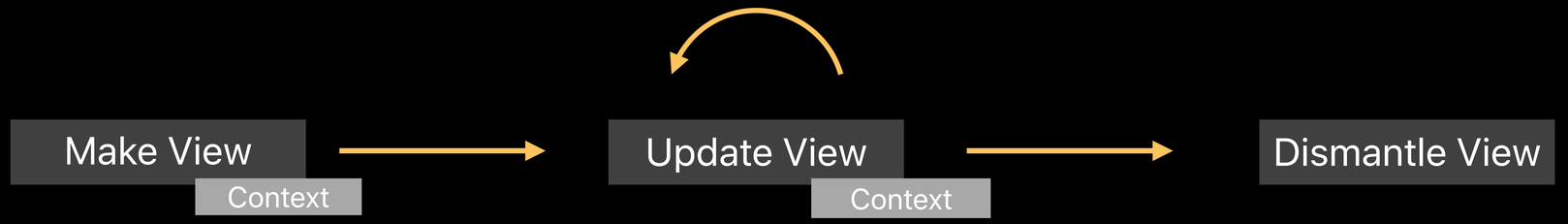
Representable Protocol



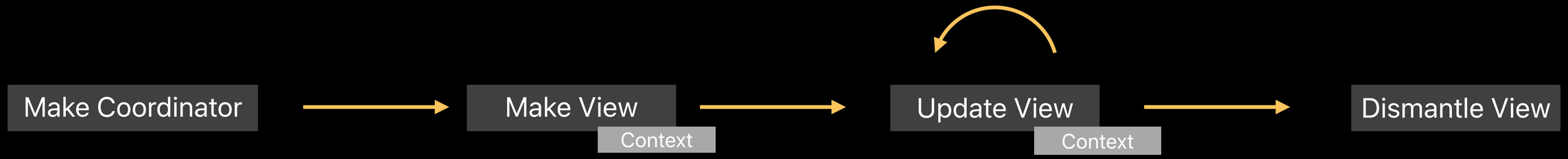
Representable Protocol



Representable Protocol



Representable Protocol



9:41

Plants

Hawaiian Hibiscus

Annual

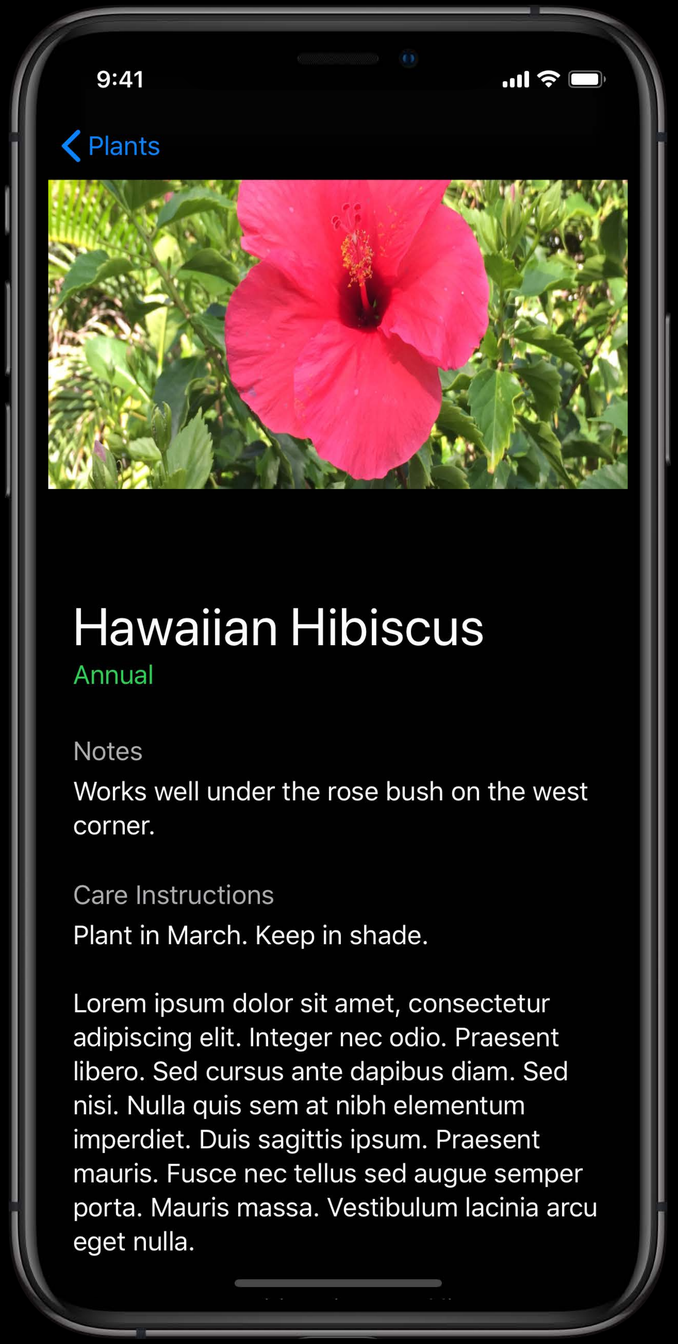
Notes

Works well under the rose bush on the west corner.

Care Instructions

Plant in March. Keep in shade.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer nec odio. Praesent libero. Sed cursus ante dapibus diam. Sed nisi. Nulla quis sem at nibh elementum imperdiet. Duis sagittis ipsum. Praesent mauris. Fusce nec tellus sed augue semper porta. Mauris massa. Vestibulum lacinia arcu eget nulla.


9:41
Plants
Hawaiian Hibiscus
Annual
Notes
Works well under the rose bush on the west corner.
Care Instructions
Plant in March. Keep in shade.
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer nec odio. Praesent libero. Sed cursus ante dapibus diam. Sed nisi. Nulla quis sem at nibh elementum imperdiet. Duis sagittis ipsum. Praesent mauris. Fusce nec tellus sed augue semper porta. Mauris massa. Vestibulum lacinia arcu eget nulla.

9:41

Plants

Hawaiian Hibiscus

Annual

Notes

Works well under the rose bush on the west corner.

Care Instructions

Plant in March. Keep in shade.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer nec odio. Praesent libero. Sed cursus ante dapibus diam. Sed nisi. Nulla quis sem at nibh elementum imperdiet. Duis sagittis ipsum. Praesent mauris. Fusce nec tellus sed augue semper porta. Mauris massa. Vestibulum lacinia arcu eget nulla.

9:41

Plants

4/5

Hawaiian Hibiscus

Annual

Notes

Works well under the rose bush on the west corner.

Care Instructions

Plant in March. Keep in shade.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer nec odio. Praesent libero. Sed cursus ante dapibus diam. Sed nisi. Nulla quis sem at nibh elementum imperdiet. Duis sagittis ipsum. Praesent mauris. Fusce nec tellus sed augue semper porta. Mauris massa. Vestibulum lacinia arcu eget nulla.

Demo

Embedding a UIKit view in SwiftUI

Try It!

Embed SwiftUI views using Hosting Controllers

Connect Hosting Controllers with IBSegueActions

Add existing views to SwiftUI

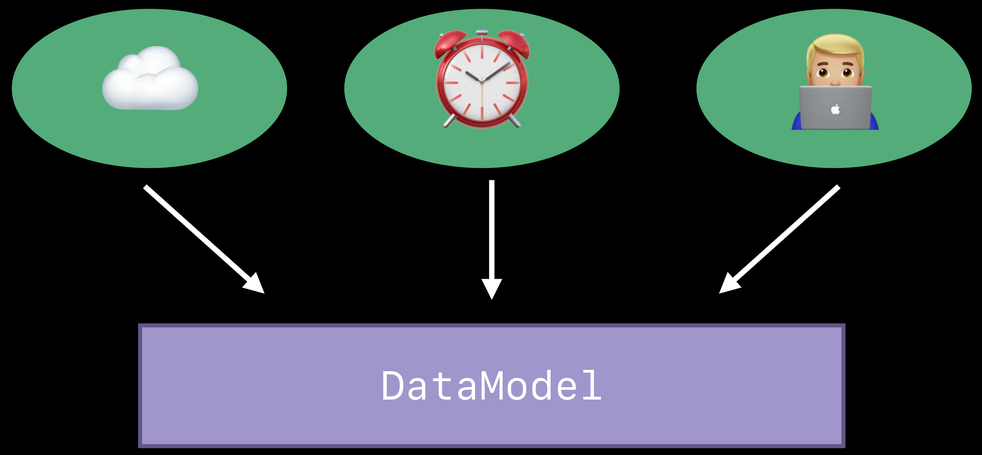
Leverage representable context

Integrating Your Data Model

Raleigh Ledet, SwiftUI and Senior AppKit Engineer

Static Data

Dynamic Data

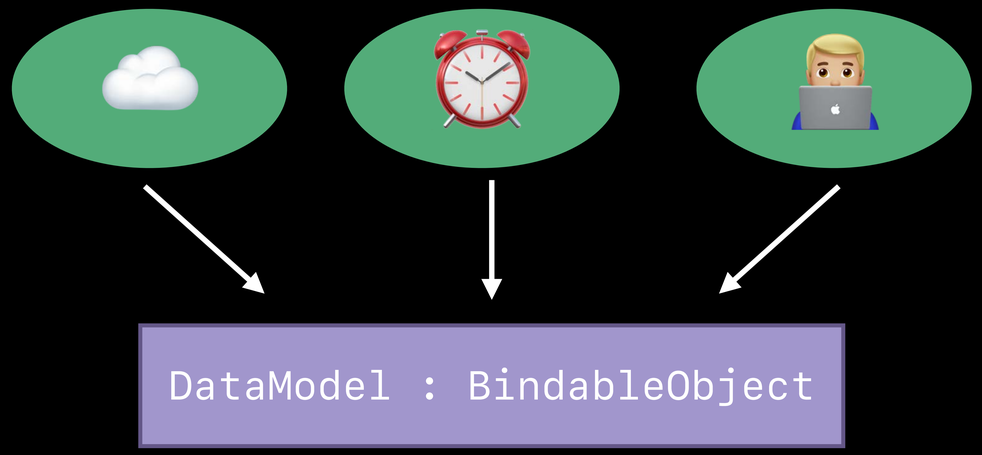


DataModel

SwiftUI

BindableObject

Dynamic data

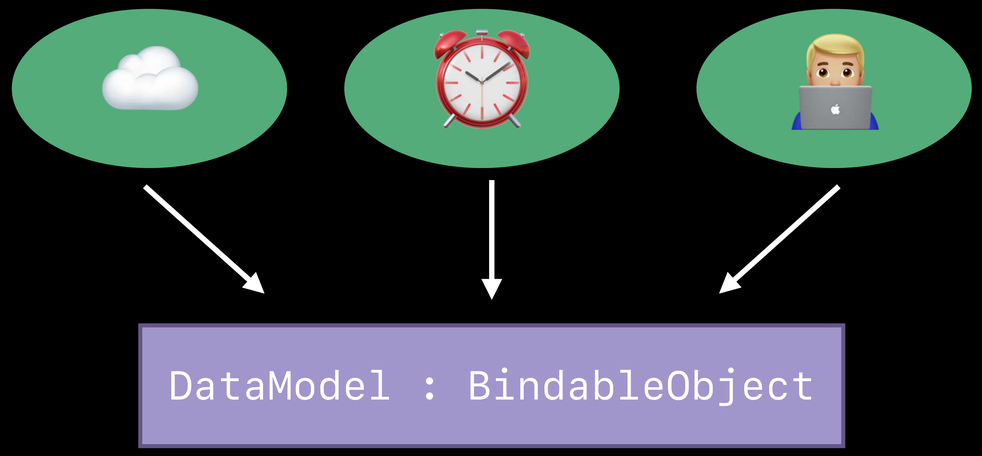


`DataModel : BindableObject`

SwiftUI

BindableObject

Dynamic data



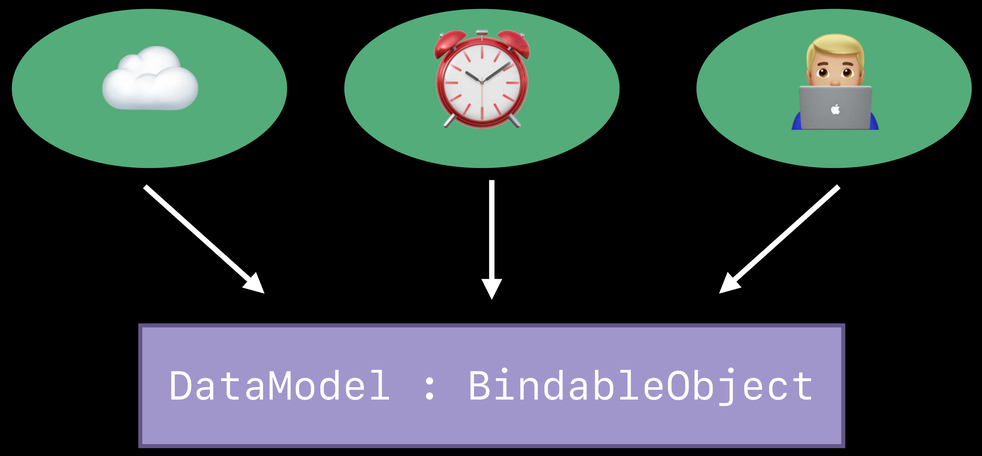
```
DataModel : BindableObject
```

SwiftUI

```
var didChange: PublisherType
```

BindableObject

Dynamic data



`DataModel : BindableObject`

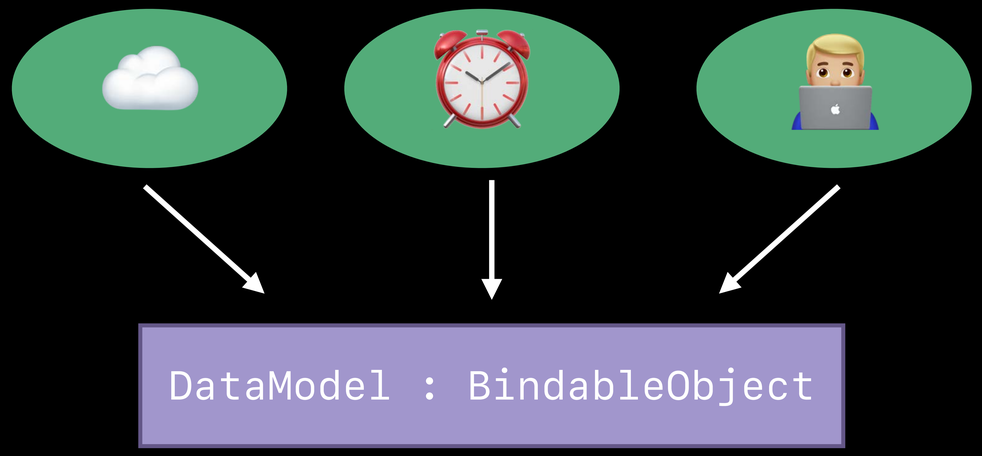
`var didChange: PublisherType`

SwiftUI

`var data: DataModel`

BindableObject

Dynamic data



DataModel : BindableObject

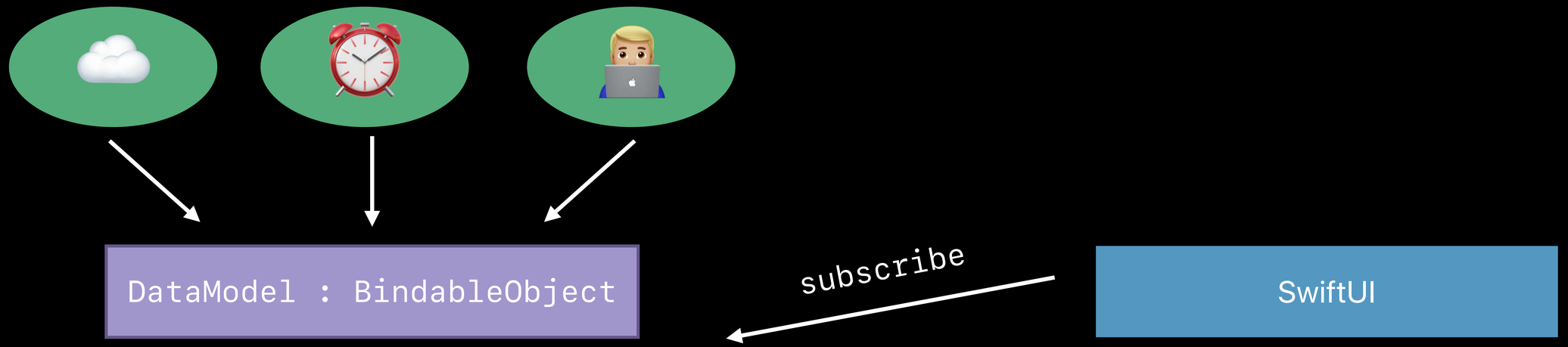
SwiftUI

```
var didChange: PublisherType
```

```
@ObjectBinding var data: DataModel
```

BindableObject

Dynamic data

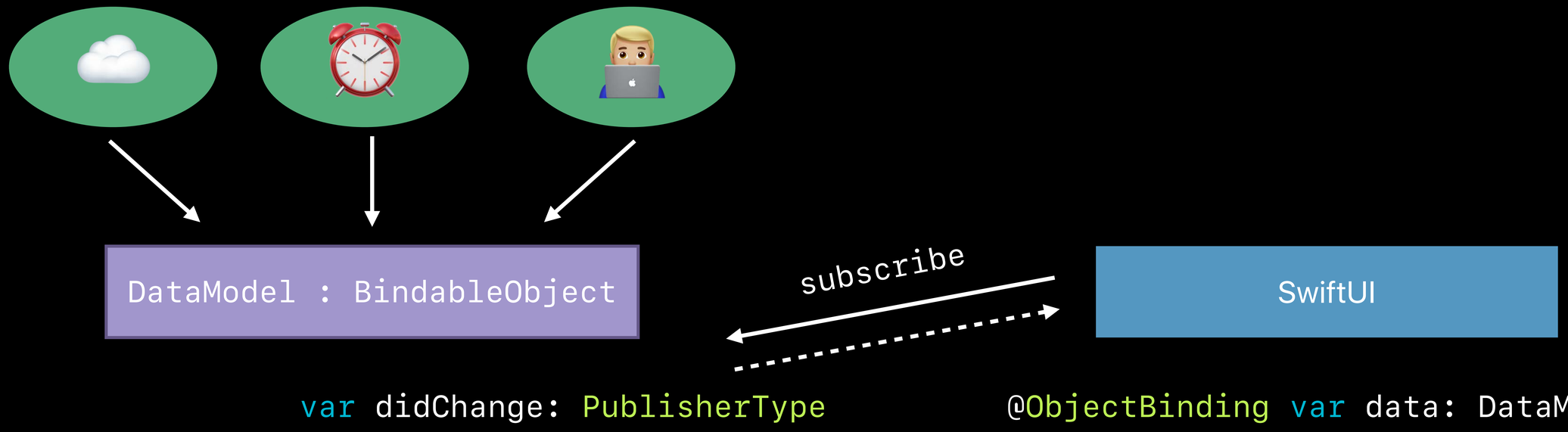



```
var didChange: PublisherType
```

```
@ObjectBinding var data: DataModel
```

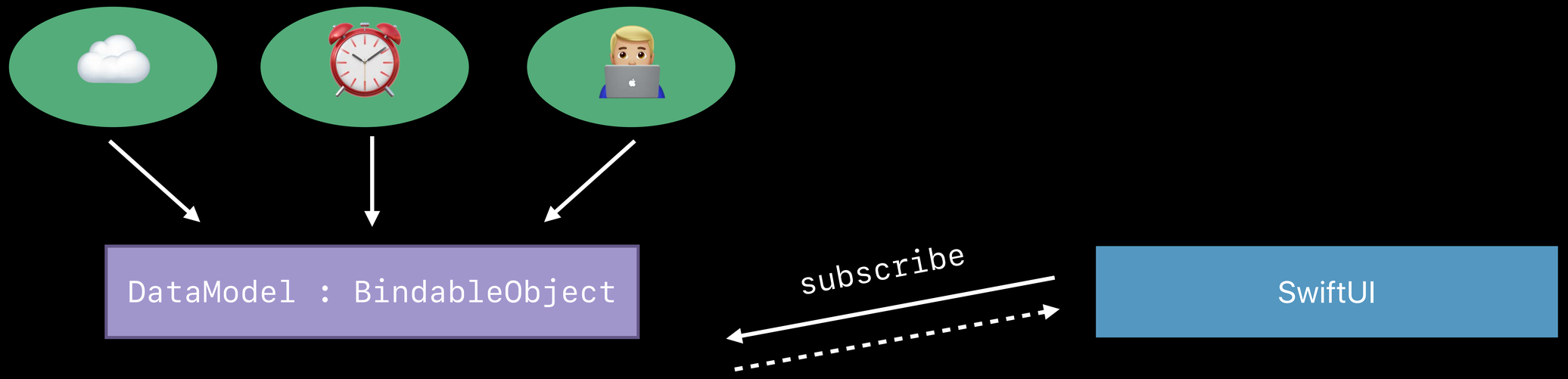
BindableObject

Dynamic data



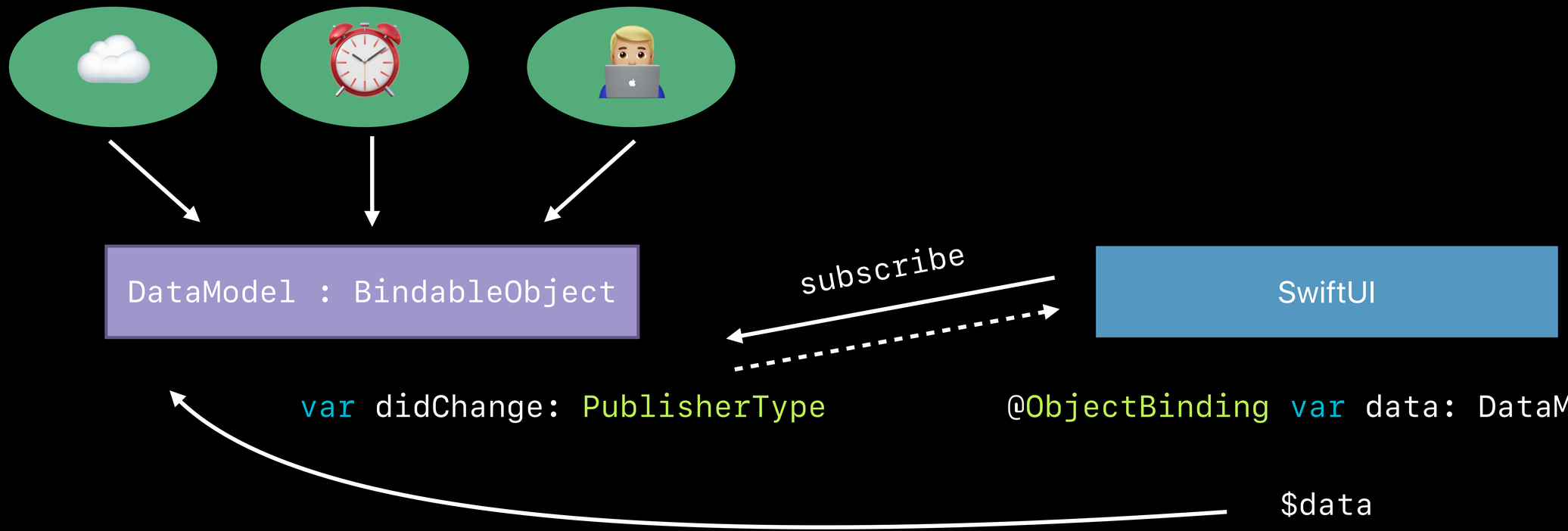
BindableObject

Dynamic data



BindableObject

Dynamic data



BindableObject

Dynamic data

Simple protocol

Flexible

Data model is source of truth

Demo

Integrating with our data

BindableObject

Notifications

```
class MyDataModel : BindableObject {
   public var didChange: NotificationCenter.default.publisher(for: .MyNotification)
        .receive(on: RunLoop.main)
}
```

BindableObject

Key Value Observing

```
class MyDataModel : NSObject, BindableObject {
   public var didChange: MyObject.publisher(for: \.myKeypath)
        .receive(on: RunLoop.main)
}
```

BindableObject

Key Value Observing

```
class MyDefaults : BindableObject {
   public var didChange: AnyPublisher<Bool, Never>

   init() {
      let defaults = UserDefaults.standard
      didChange = Publishers.Merge(
         defaults.publisher(for: \.userOption1),
         defaults.publisher(for: \.userOption2)
      )
      .receive(on: RunLoop.main)
      .eraseToAnyPublisher()
   }
}
```

BindableObject

Key Value Observing

```
class MyDefaults : BindableObject {
   public var didChange: AnyPublisher<Bool, Never>

   init() {
      let defaults = UserDefaults.standard
      didChange = Publishers.Merge(
         defaults.publisher(for: \.userOption1),
         defaults.publisher(for: \.userOption2)
      )
      .receive(on: RunLoop.main)
      .eraseToAnyPublisher()
   }
}
```

Combine

Unified declarative API for processing values over time

KVO

Notifications

URLSession

Operators

And more...

BindableObject

Delegate

```
class MyDataModel : BindableObject {
    var didChange = PassthroughSubject<Void, Never>()
}

extension MyDataModel : NSFetchedResultsControllerDelegate {
    func controllerDidChangeContent(
        _ controller: NSFetchedResultsController<NSFetchRequestResult>)
    {
        didChange.send()
    }
}
```

BindableObject

Delegate

```
class MyDataModel : BindableObject {
    var didChange = PassthroughSubject<Void, Never>()
}

extension MyDataModel : NSFetchedResultsControllerDelegate {
    func controllerDidChangeContent(
        _ controller: NSFetchedResultsController<NSFetchRequestResult>)
    {
        didChange.send()
    }
}
```

BindableObject

Delegate

```
class MyDataModel : BindableObject {
    var didChange = PassthroughSubject<Void, Never>()
}

extension MyDataModel : NSFetchedResultsControllerDelegate {
    func controllerDidChangeContent(
        _ controller: NSFetchedResultsController<NSFetchRequestResult>)
    {
        didChange.send()
    }
}
```

Tools for Data Flow

@Environment

Property

@Binding

BindableObject

Subscription

@State

@EnvironmentObject

@ObjectBinding

Integrating with the System

NSItemProviders

Cross or inter-process data passing

Universal Type Identifier

Asynchronous

Drag and Drop

```
public func onDrag(_ data: @escaping () -> NSItemProvider) -> some View
```

Drag and Drop

```
public func onDrag(_ data: @escaping () -> NSItemProvider) -> some View

public func onDrop(of supportedTypes: [String],
   isTargeted: Binding<Bool>?,
   perform action: @escaping ([NSItemProvider], CGPoint) -> Bool) -> some View
```

Drag and Drop

```
public func onDrag(_ data: @escaping () -> NSItemProvider) -> some View

public func onDrop(of supportedTypes: [String],
   isTargeted: Binding<Bool>?,
   perform action: @escaping ([NSItemProvider], CGPoint) -> Bool) -> some View

public func onDrop(of supportedTypes: [String], delegate: DropDelegate) -> some View
```

Pasteboard

```
public func onPaste(of supportedTypes: [String],
    perform action: @escaping ([NSItemProvider]) -> Void) -> some View

public func onPaste<Payload>(of supportedTypes: [String],
    validator: @escaping ([NSItemProvider]) -> Payload?,
    perform action: @escaping (Payload) -> Void) -> some View
```

Focus

Used to navigate to UI elements

Important for all platforms

Focus

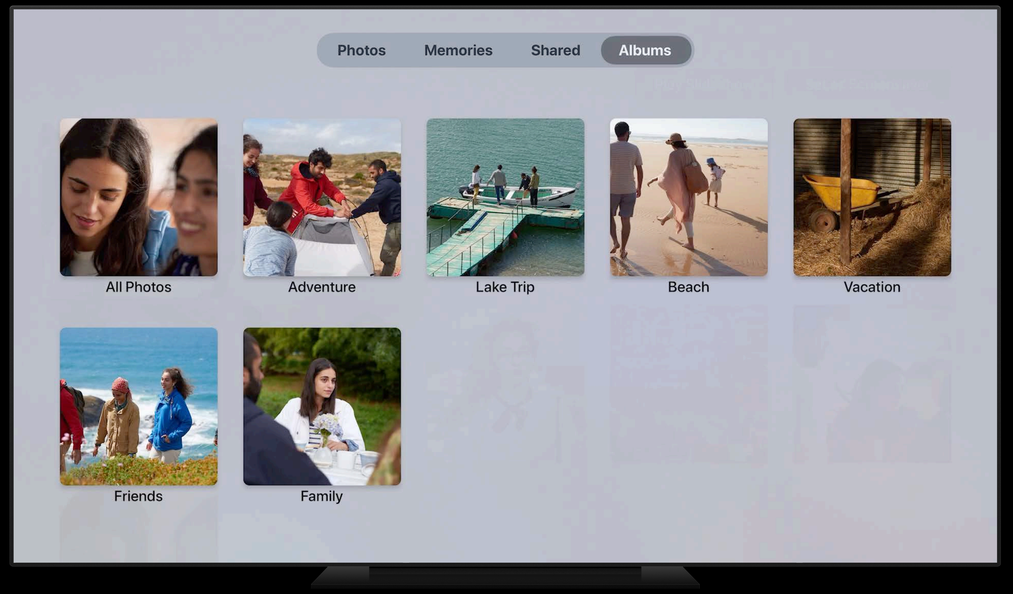
Used to navigate to UI elements

Important for all platforms

Focus

Used to navigate to UI elements

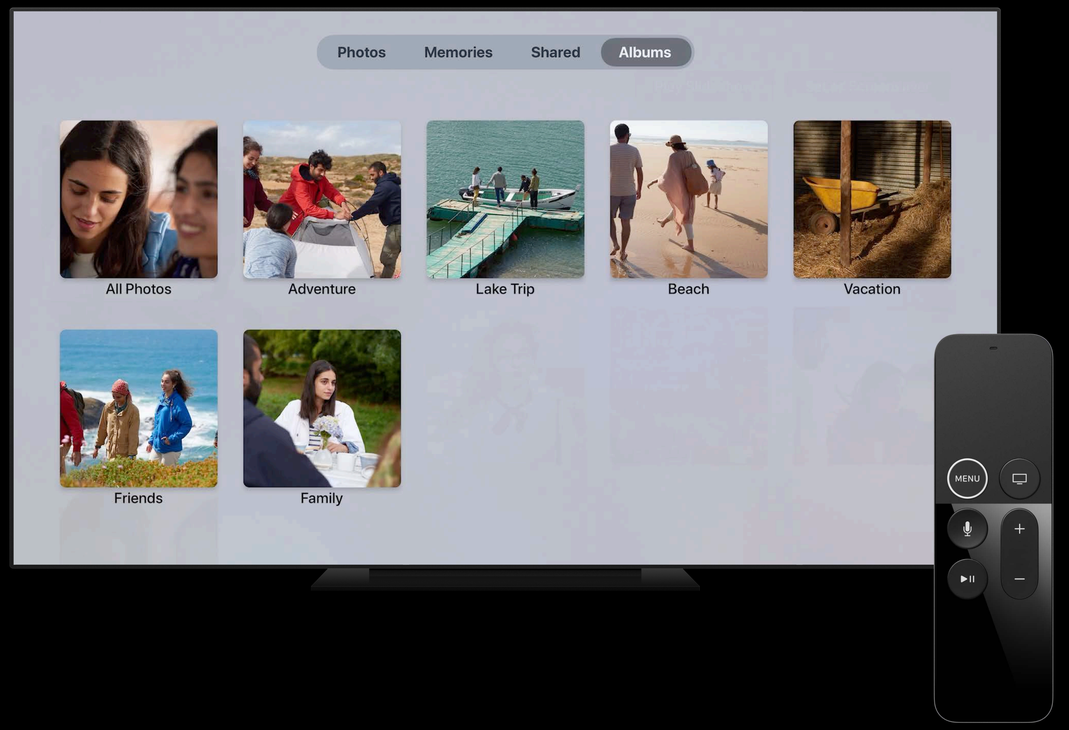
Important for all platforms



Focus

Used to navigate to UI elements

Important for all platforms



Focus

Used to navigate to UI elements

Important for all platforms

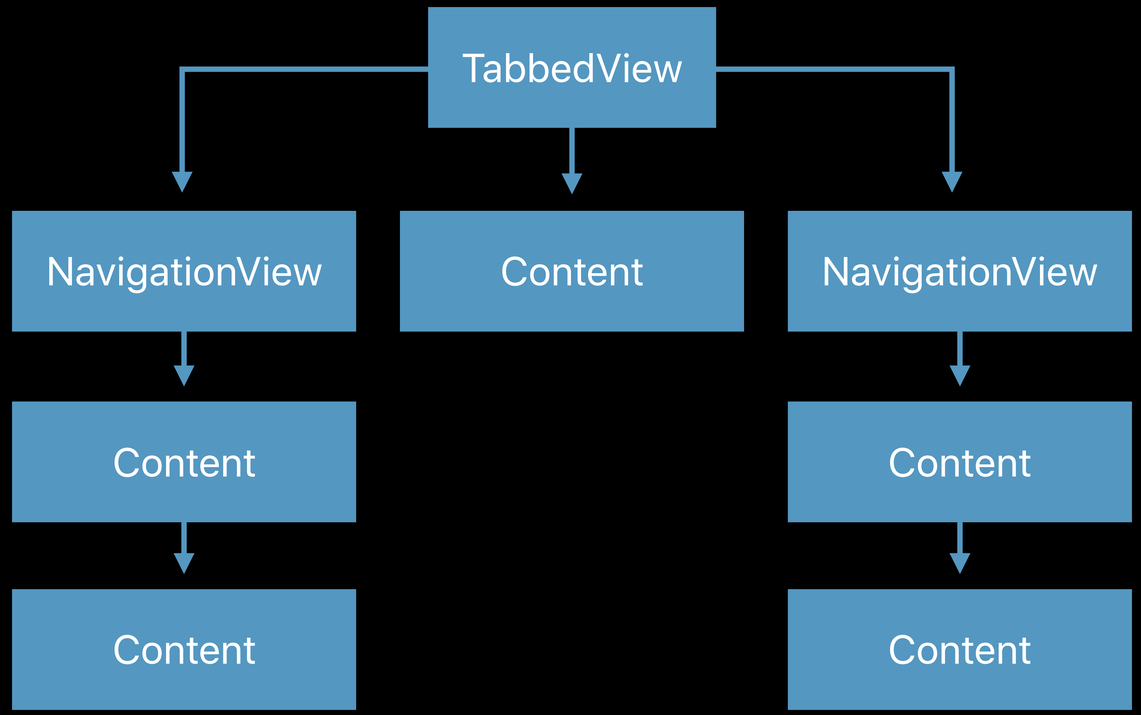
Focus

Used to navigate to UI elements

Important for all platforms

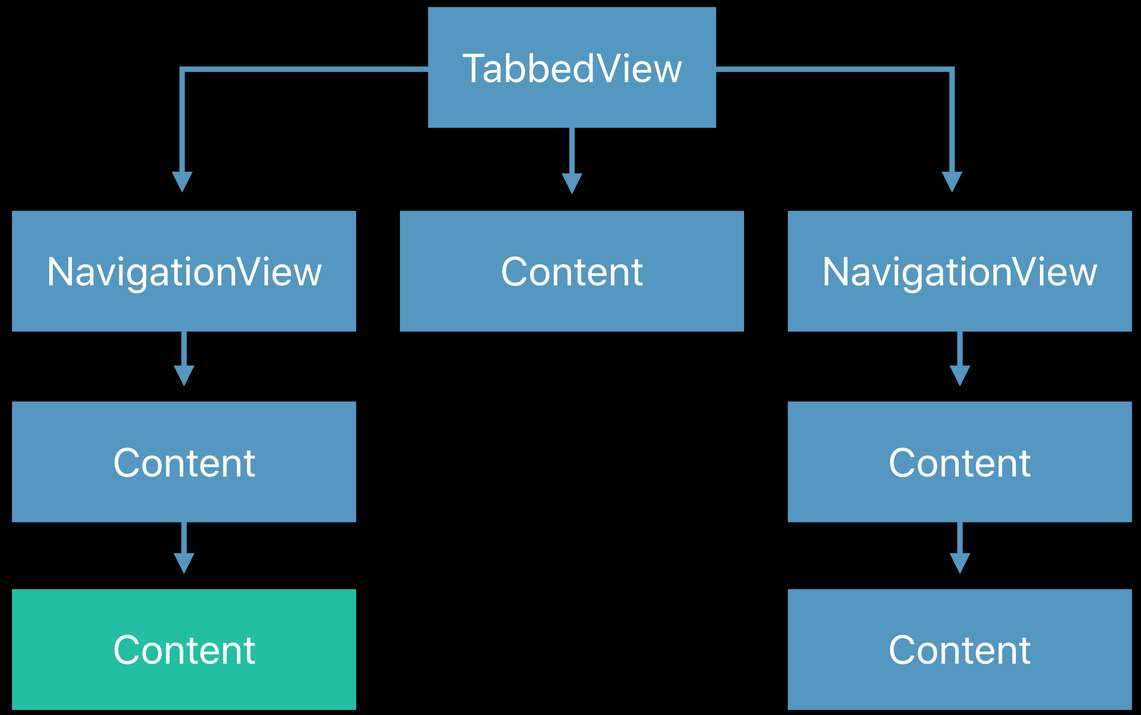
Focus


MINUTES
00
01
00
01
29
30
31
Start
Cancel

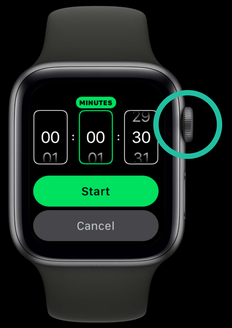

TabbedView
NavigationView
Content
NavigationView
Content
Content
Content
Content

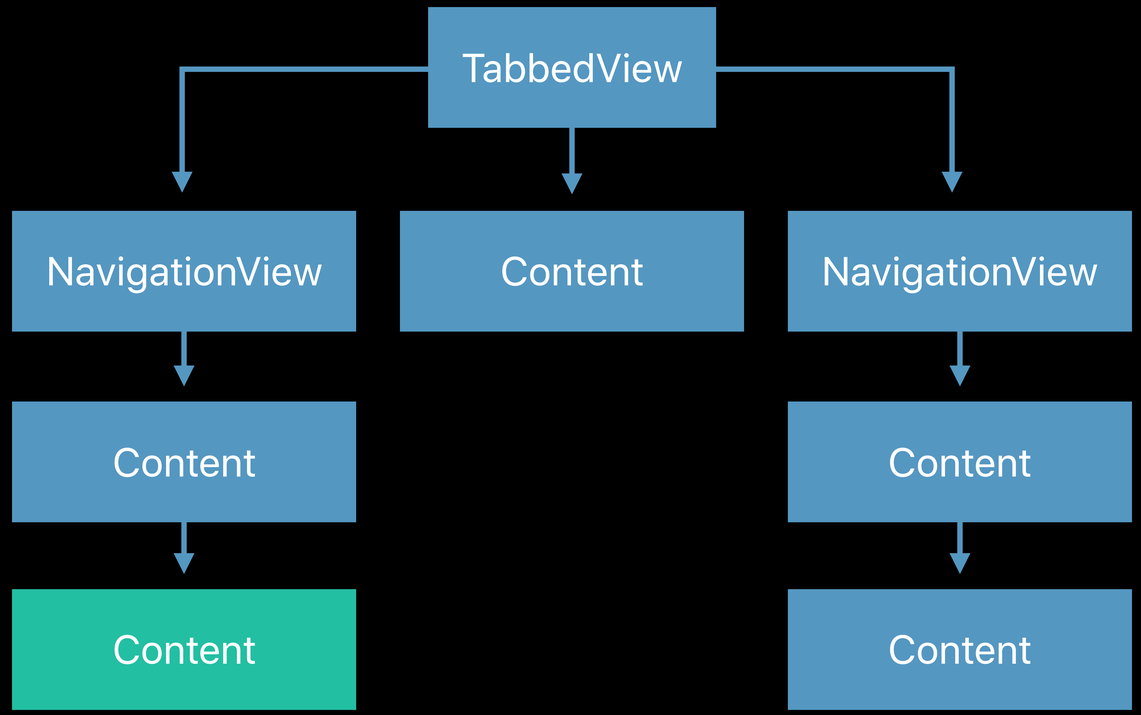
Focus


MINUTES
00
01
00
01
29
30
31
Start
Cancel

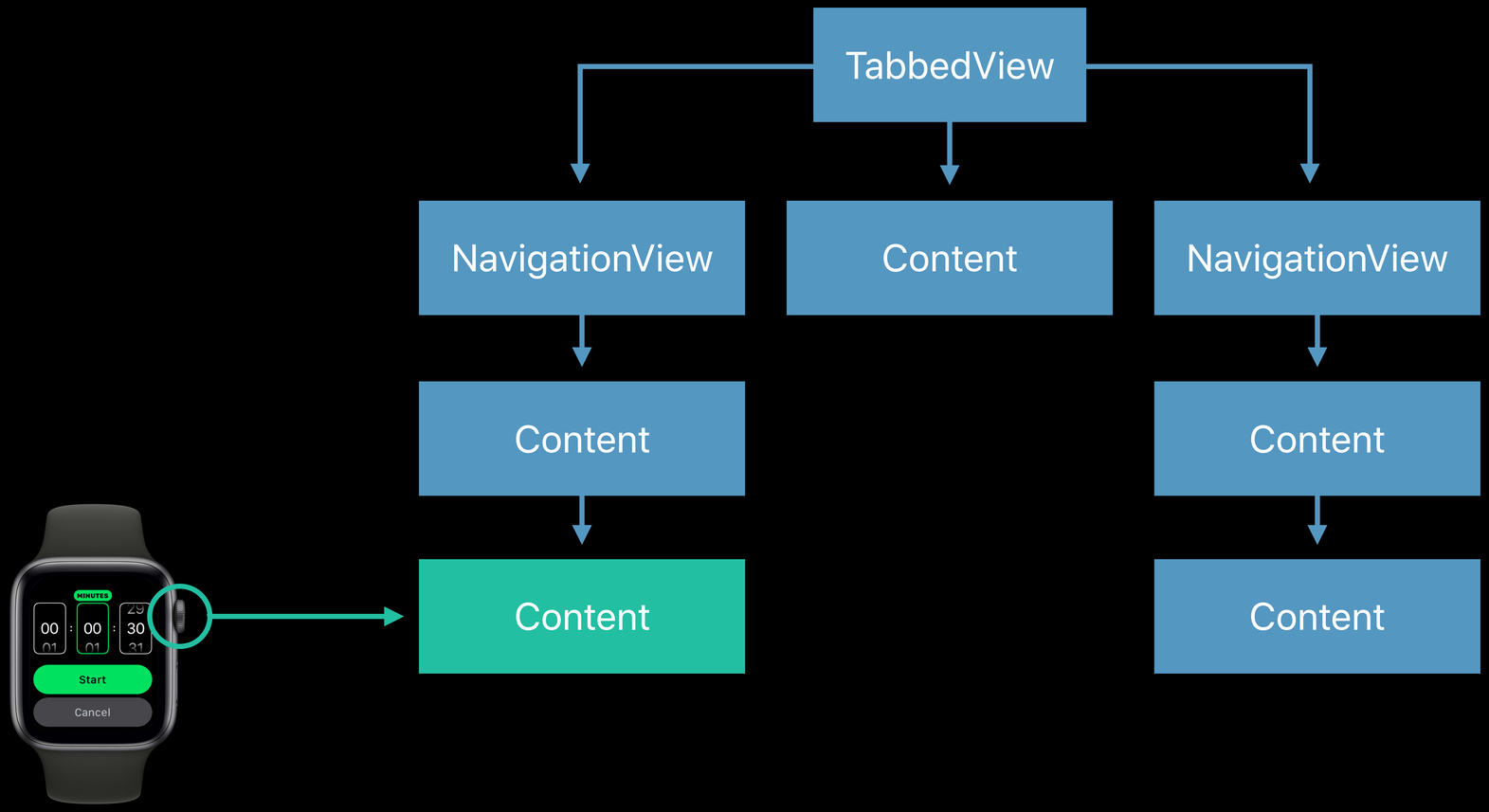

TabbedView
NavigationView
Content
NavigationView
Content
Content
Content
Content

Focus

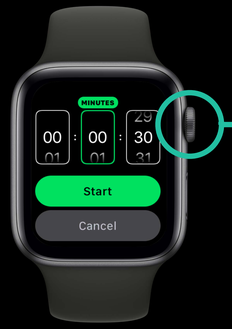

MINUTES
00
00
30
Start
Cancel

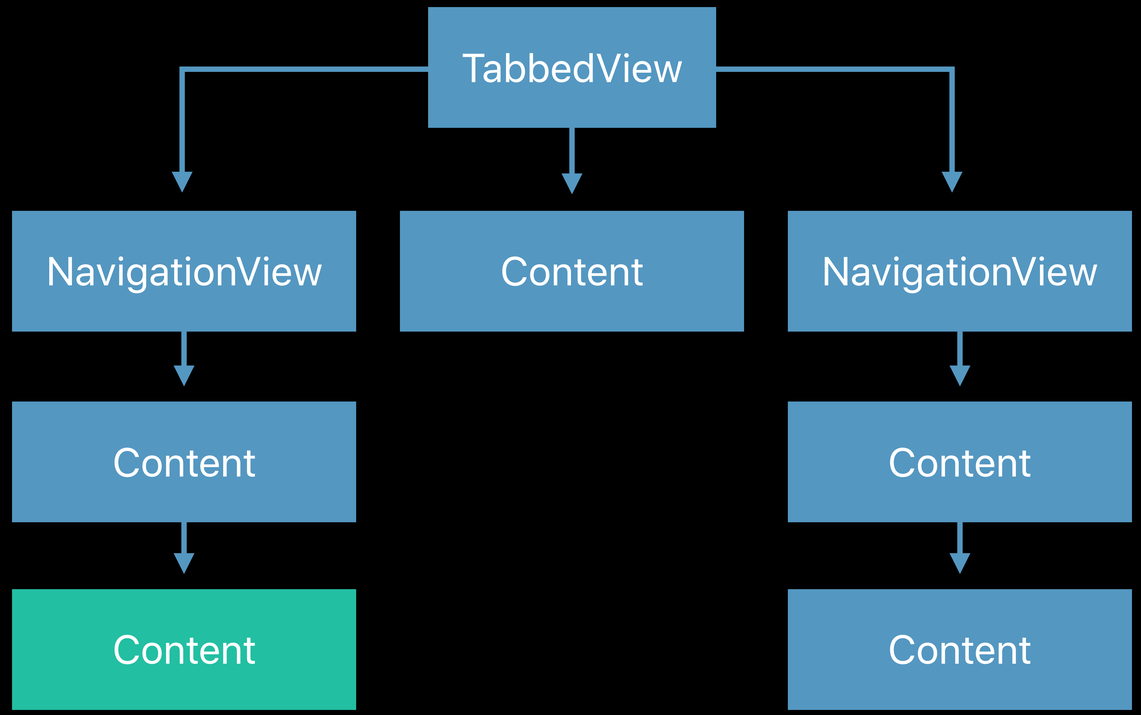

TabbedView
NavigationView
Content
NavigationView
Content
Content
Content
Content

Focus

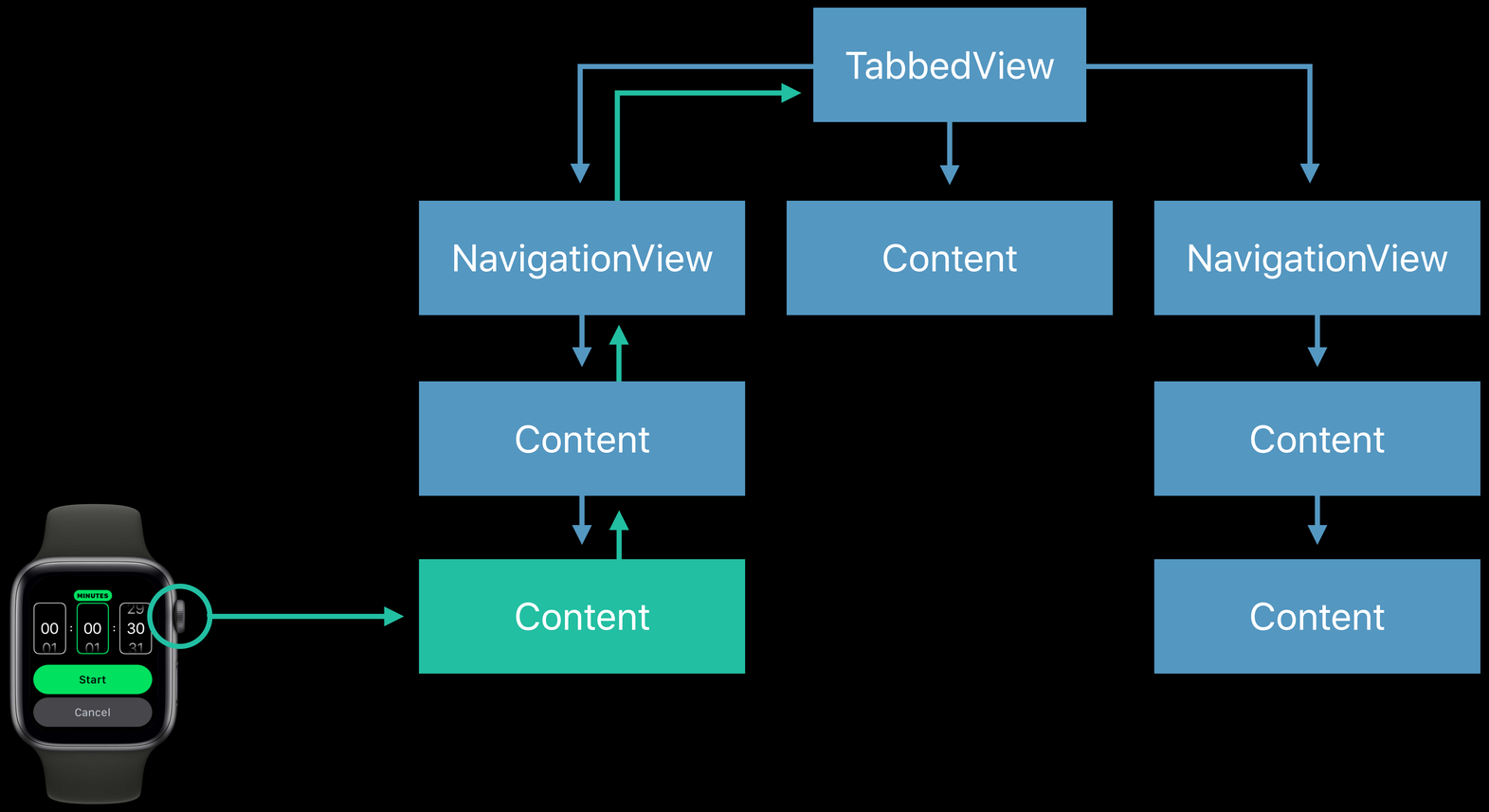

TabbedView
NavigationView
Content
NavigationView
Content
Content
Content
Content
MINUTES
00
00
30
Start
Cancel

Focus


MINUTES
00
00
30
Start
Cancel


TabbedView
NavigationView
Content
NavigationView
Content
Content
Content
Content

Focus


TabbedView
NavigationView
Content
NavigationView
Content
Content
Content
Content
MINUTES
00
00
30
Start
Cancel

Focus

```
public func focusable(_ isFocusable: Bool,
                      onFocusChange: @escaping (Bool) -> Void) -> some View
```

Commands

System

```
// 'Menu' button on tvOS, Esc on macOS
public func onExit(perform action: (() -> Void)?) -> some View

// 'Play/Pause' button on tvOS
public func onPlayPause(perform action: (() -> Void)?) -> some View

And more…
```

Commands

Menus and actions

```
public func onCommand(_ command: Command, perform action: (() -> Void)?) -> some View

Command(#selector(Object.mySelector(_ :)))
```

Undo and Redo

UndoManager

Environment

Undo and Redo

UndoManager

Environment

```
@Environment(\.undoManager) var undoManager
```

Objective-C Integration

Objective-C Integration

Standard Objective-C/Swift rules

Objective-C Integration

Wrap HostingControllers in Swift

```
@objc class WrappedHostingController : UIViewController {
   let hostingController = UIHostingController(rootView: RootView())
   ...
}


[self presentViewController: wrappedHostingController animated: YES completion: ^{...}];
```

Objective-C Integration

Wrap HostingControllers in Swift

```
@objc class WrappedHostingController : UIViewController {
    let hostingController = UIHostingController(rootView: RootView())
    ...
}
```

```
[self presentViewController: wrappedHostingController animated: YES completion: ^{...}];
```

Objective-C Integration

Wrap HostingControllers in Swift

```
@objc class WrappedHostingController : UIViewController {
    let hostingController = UIHostingController(rootView: RootView())
    ...
}



[self presentViewController: wrappedHostingController animated: YES completion: ^{...}];
```

Objective-C Integration

Wrap HostingControllers in Swift

```
@objc class WrappedHostingController : UIViewController {
    let hostingController = UIHostingController(rootView: RootView())
    ...
}
```

```
[self presentViewController: wrappedHostingController animated: YES completion: ^{...}];
```

Objective-C Integration

Wrap data model in Swift

```
class WrappedDataModel : NSObject, BindableObject {
   var didChange = NotificationCenter.default.publisher(for: .DataDidChange)
            .receive(on: RunLoop.main)
   var dataModel: ObjCDataModel
   ...
}
```

Objective-C Integration

Wrap data model in Swift

```
class WrappedDataModel : NSObject, BindableObject {
    var didChange = NotificationCenter.default.publisher(for: .DataDidChange)
                .receive(on: RunLoop.main)
    var dataModel: ObjCDataModel
    ...
}
```

Objective-C Integration

Wrap data model in Swift

```
class WrappedDataModel : NSObject, BindableObject {
    var didChange = NotificationCenter.default.publisher(for: .DataDidChange)
                .receive(on: RunLoop.main)
    var dataModel: ObjCDataModel
    ...
}
```

Objective-C Integration

Wrap data model in Swift

```
class WrappedDataModel : NSObject, BindableObject {
    var didChange = NotificationCenter.default.publisher(for: .DataDidChange)
                .receive(on: RunLoop.main)
    var dataModel: ObjCDataModel
    ...
}
```

Objective-C Integration

Wrap HostingControllers in Swift

Wrap data model in Swift


GardenPlants File Edit View Go Window Help
Mon 9:41 AM
Untitled
Hawaiian Hibiscus Annual
Colorful punch.
Hydrangea Serrata Shrub
Combact hydrangea. Good to create a shrub border.
Foxglove Perennial
Towers of blooms.
Peony Perennial
Bright flowers.
Pineapple Perennial
Tropical plant with delicious fruit.
Edit
4/5
Hawaiian Hibiscus
Annual
Notes
Works well under the rose bush on the west corner.
Care Instructions
Plant in March. Keep in shade.
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer nec odio. Praesent libero. Sed cursus ante dapibus diam. Sed nisi. Nulla quis sem at nibh elementum imperdiet. Duis sagittis ipsum. Praesent mauris. Fusce nec tellus sed augue semper porta. Mauris massa. Vestibulum lacinia arcu eget nulla.
Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Curabitur sodales ligula in libero. Sed dignissim lacinia nunc. Curabitur tortor. Pellentesque nibh. Aenean quam. In scelerisque sem at dolor. Maecenas mattis. Sed convallis tristique sem. Proin ut
MacBook Pro
9:41
Plants
Edit
4/5
Hawaiian Hibiscus
Annual
Notes
Works well under the rose bush on the west corner.
Care Instructions
Plant in March. Keep in shade.
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer nec odio. Praesent libero. Sed cursus ante dapibus diam. Sed nisi. Nulla quis sem at nibh elementum imperdiet. Duis sagittis ipsum. Praesent mauris. Fusce nec tellus sed augue semper porta. Mauris massa. Vestibulum lacinia arcu eget nulla.

More Information

developer.apple.com/wwdc19/231

SwiftUI Lab	Friday, 11:00


WWDC19