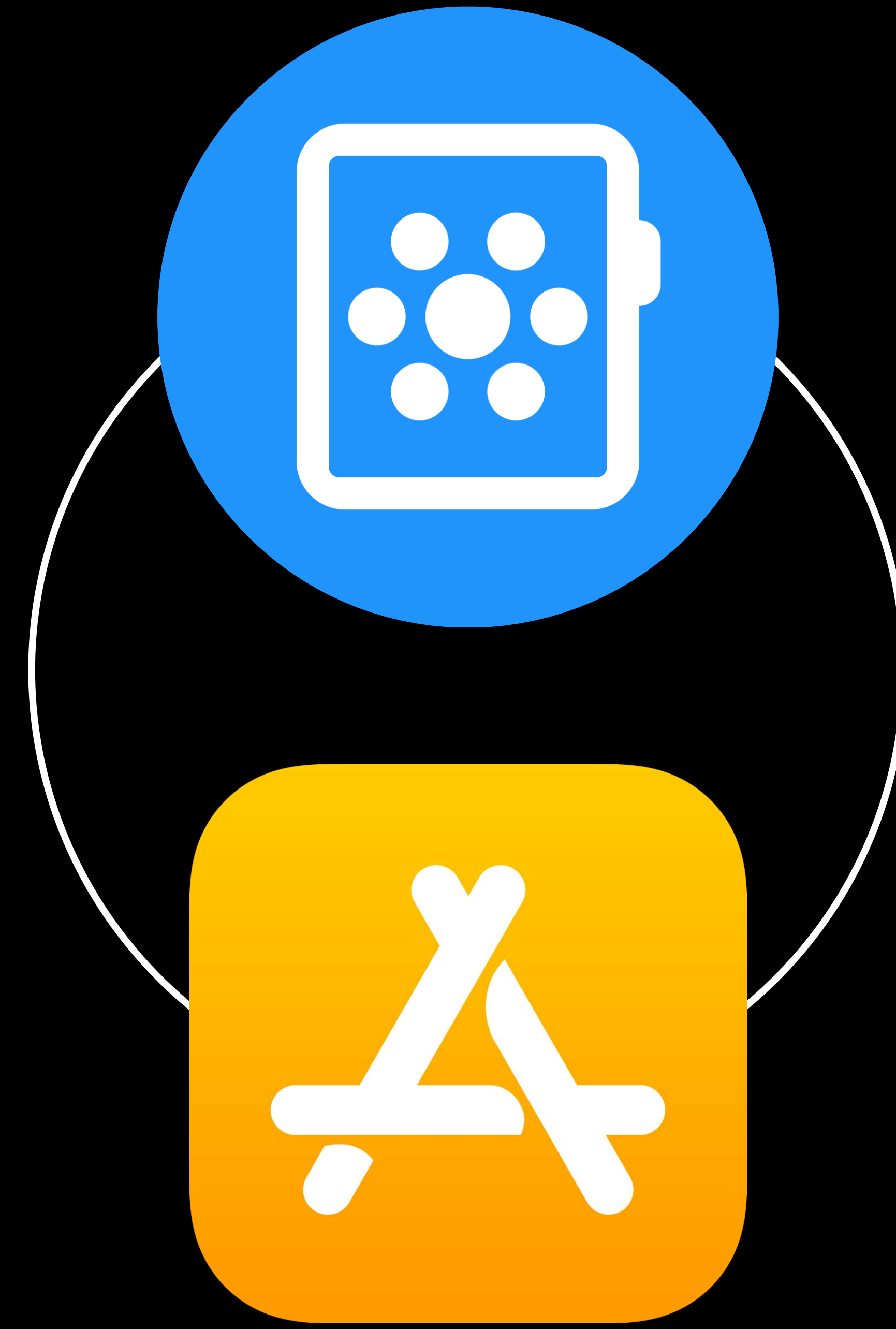


#WWDC18

What's New in watchOS

Session 206

Lori Hylan-Cho, watchOS Frameworks Engineering







Notifications



10:09



CARROT⁵

TODAY'S FORECAST



69↑ 51↓

Mostly cloudy until
afternoon. Winds
SW at 3 to 9 mph.
Chance of precip
16%.



10:09



CARROT⁵

Today: 69 ↑ 51 ↓
Mostly cloudy until
afternoon. Winds
SW at 3 to 9 mph.
Chance of precip
16%.

NEW





WALKIE-TALKIE

Neil Desai wants to talk to you now with Walkie-Talkie.

12:26 PM

+2 more

```
{
  "aps": {
    "alert": {
      "body": "Test message",
      "title": "Optional title"
    },
    "category": "myCategory",
    "thread-id": "5280"
  },
  ...
}
```

NEW





Notification Category

- Has Dynamic Interface
- Has Interactive Interface
- Handles Grouping

Name

+ Sash Color Custom

+ Wants Sash Blur

+ Title Color Default

+ Description


```
// Deprecated in watchOS 5; continue to define if targeting <= watchOS 4
```

```
func didReceive(_ notification: UNNotification,  
withCompletion completionHandler: @escaping (WKUserNotificationInterfaceType) -> Void)
```



```
func didReceive(_ notification: UNNotification) WK_AVAILABLE_WATCHOS_ONLY(5.0);
```

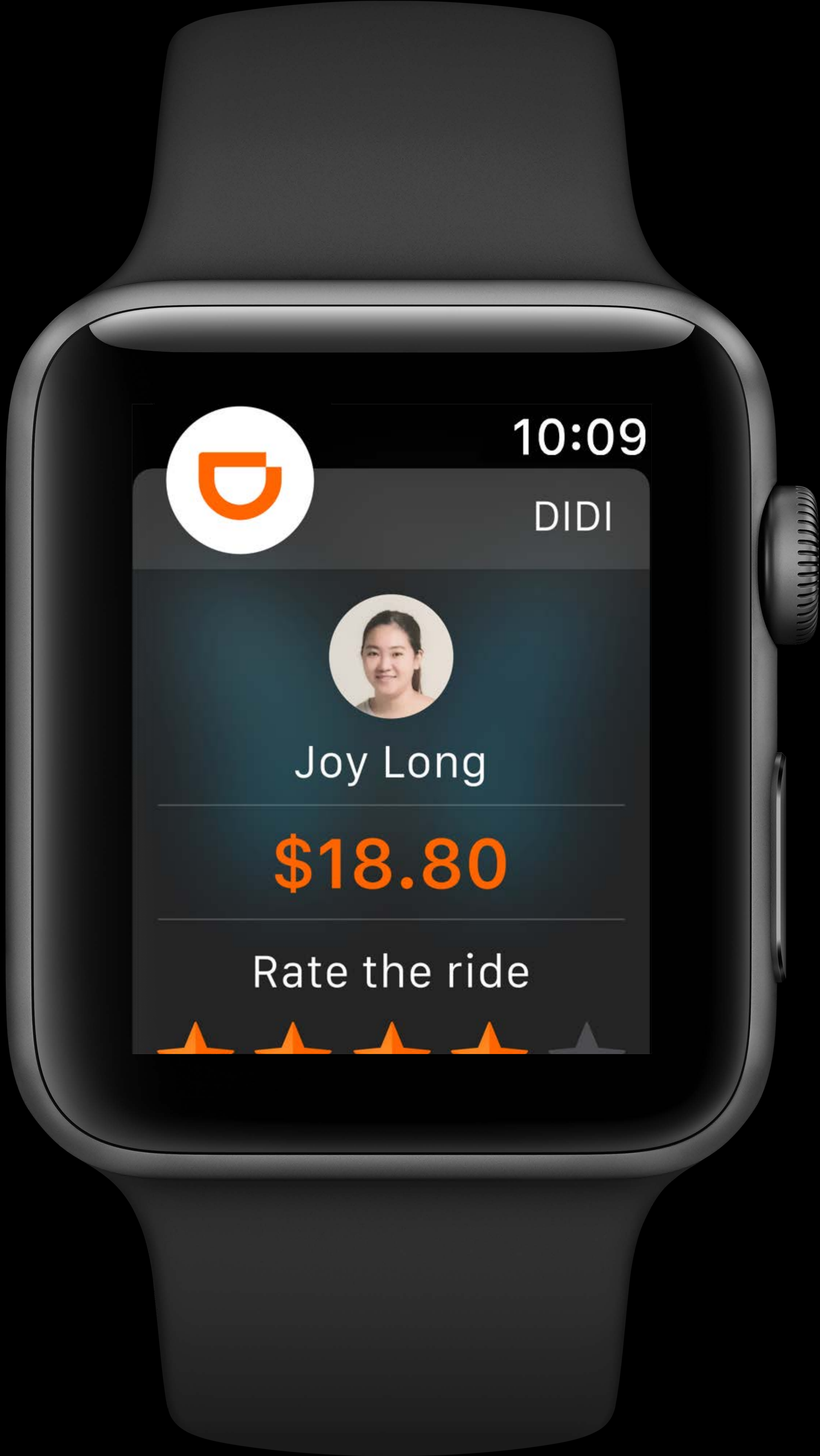


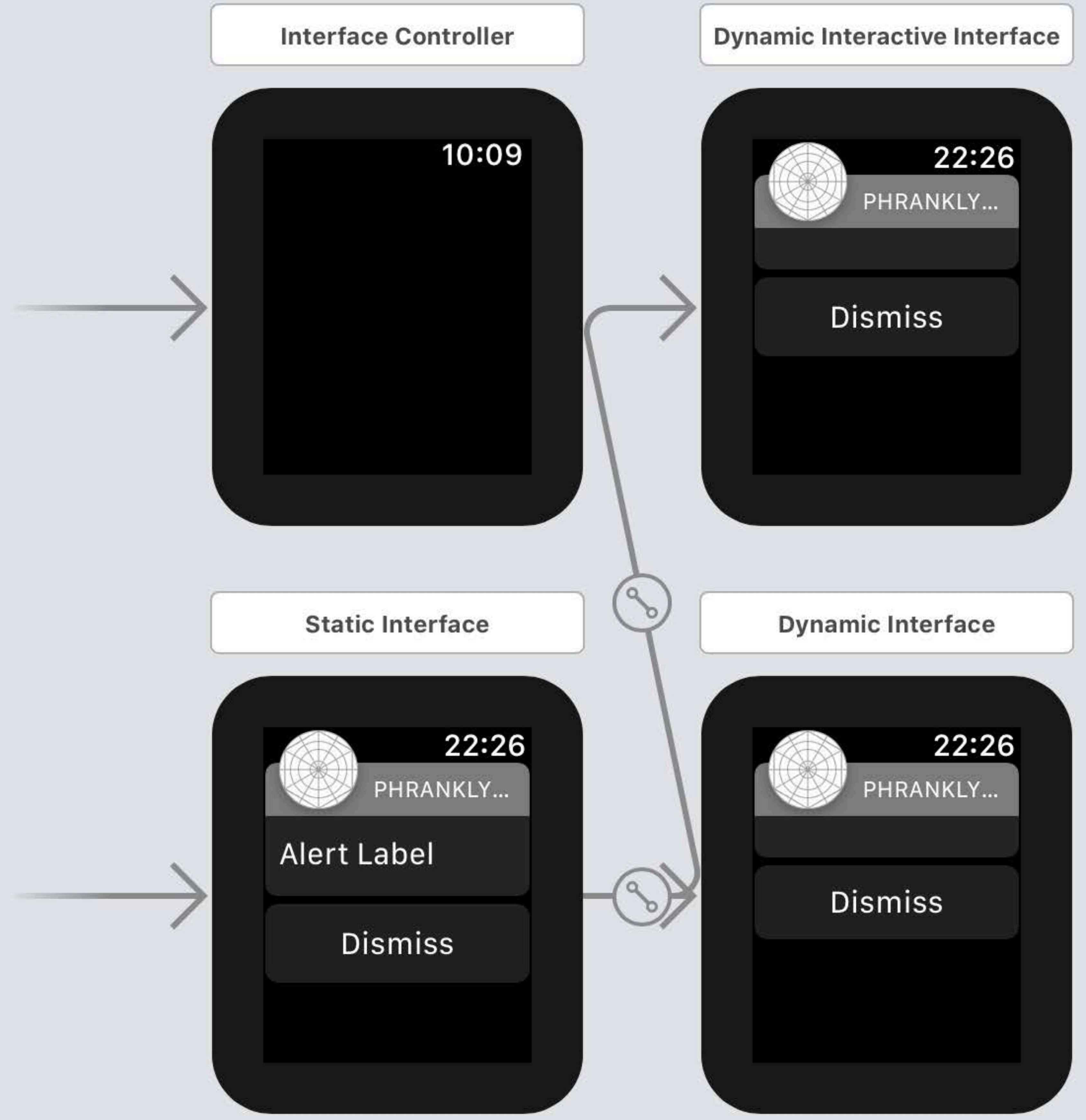
NEW

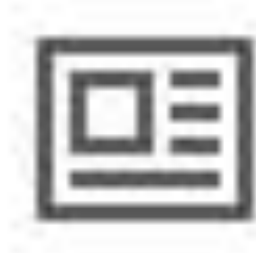
NEW



NEW







Notification Category

Has Dynamic Interface

Has Interactive Interface

Handles Grouping

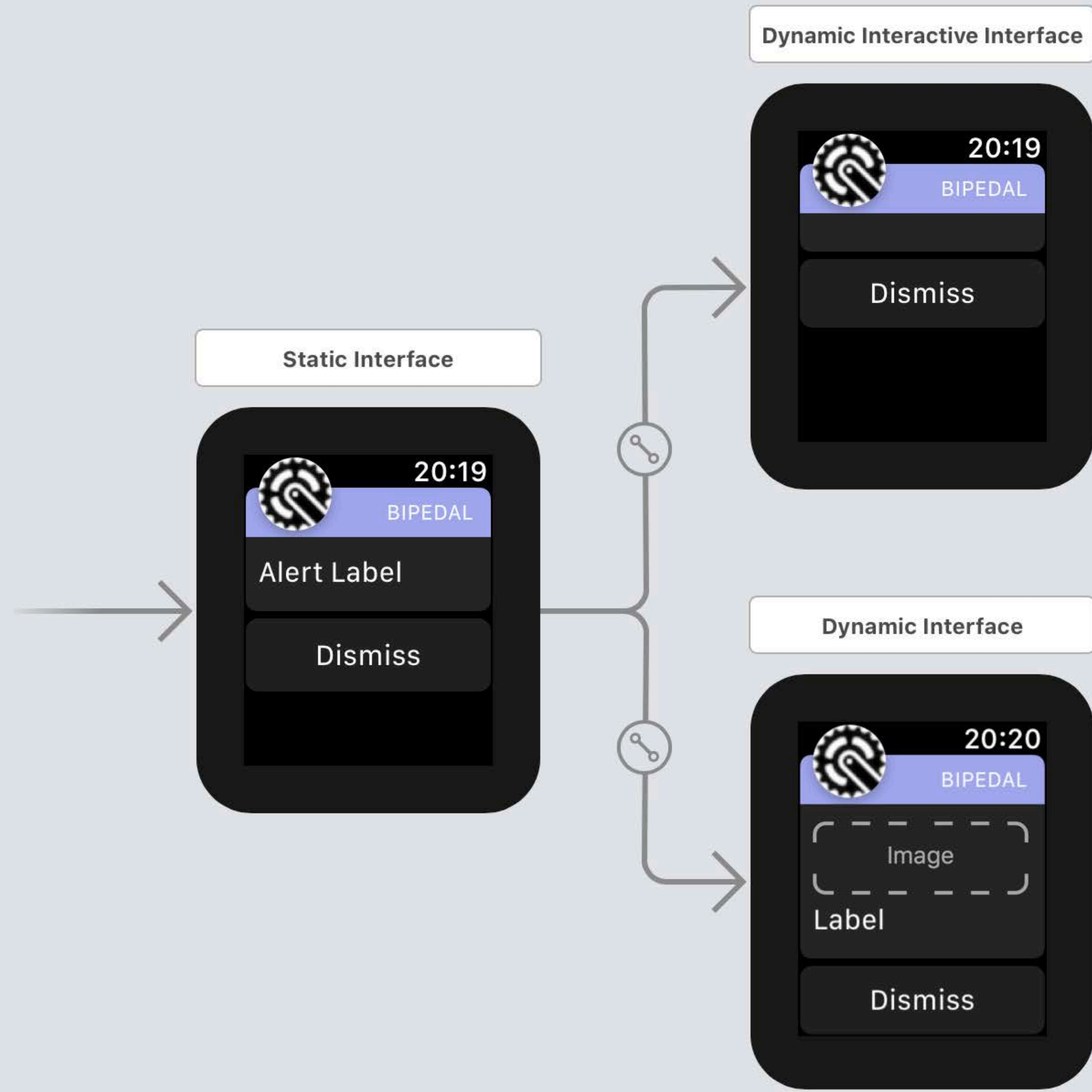
Name

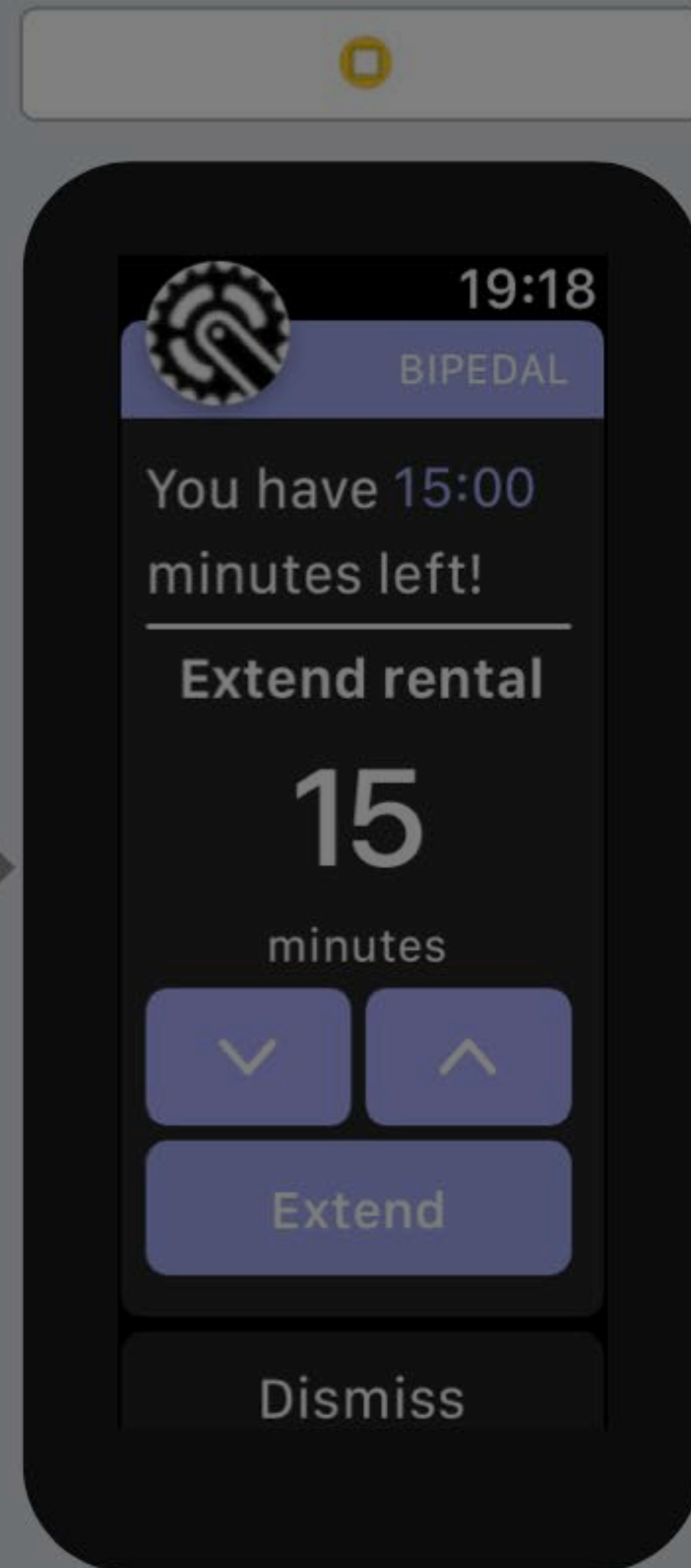
+ Sash Color Custom 

+ Wants Sash Blur

+ Title Color Default 

+ Description





Objects



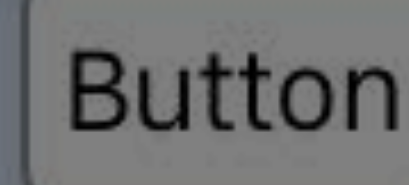
Table - Displays one or more rows of data.



Image - Displays a static or animated image.



Separator - A line for separating content in your interface.



Button - A tappable area with a title and/or image.



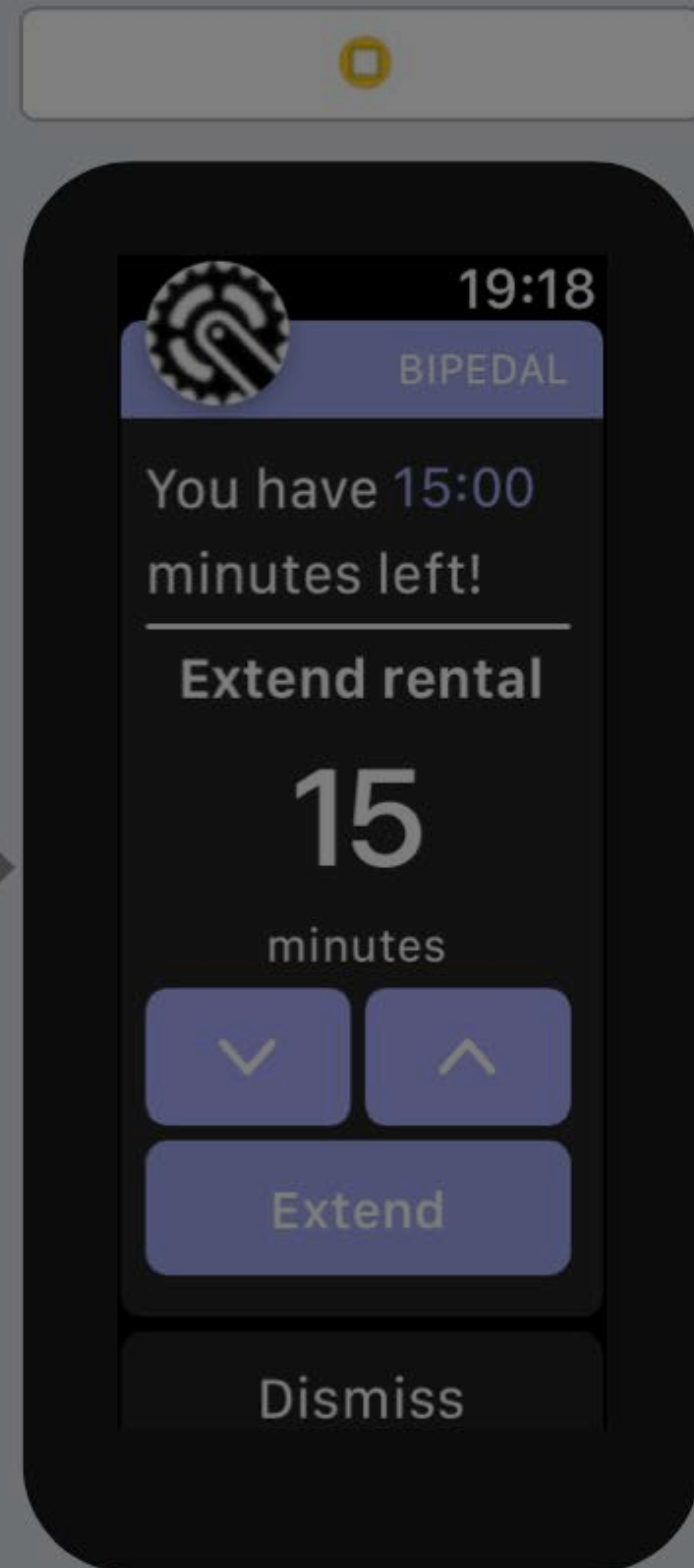
Payment Button - Standard button for initiating Apple Pay transactions.



Switch - A control for indicating a binary value.



Slider - A control for selecting a floating-point value from a range of continuous or discrete values.



Objects



Table - Displays one or more rows of data.



Image - Displays a static or animated image.



Separator - A line for separating content in your interface.



Button - A tappable area with a title and/or image.



Payment Button - Standard button for initiating Apple Pay transactions.



Switch - A control for indicating a binary value.



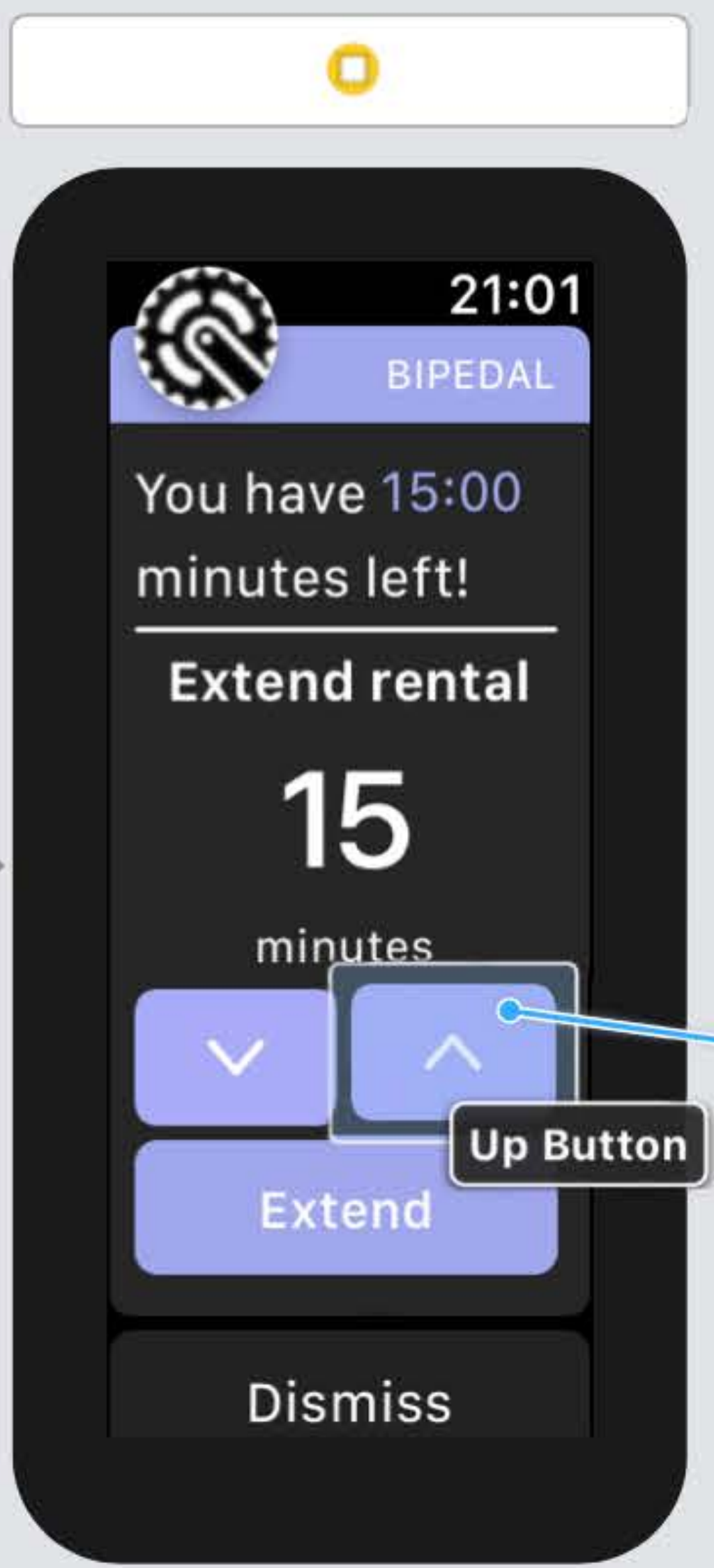
Slider - A control for selecting a floating-point value from a range of continuous or discrete values.

Interface.storyboard

Interface.storyboard (Base) > No Selection

Automatic > NotificationController.swift > upButtonTapped()

< 2 > + x



The storyboard shows a mobile notification interface. At the top, there is a yellow dot in a white bar. Below it, a notification card is displayed. The card has a header with a logo and the text "BIPEDAL". The main content of the notification says "You have 15:00 minutes left!". Below this, there is a large "15" with "minutes" underneath it. There are two buttons: a blue "Extend rental" button and a grey "Dismiss" button. A blue callout box labeled "Up Button" points to the up arrow button in the "Extend rental" button.

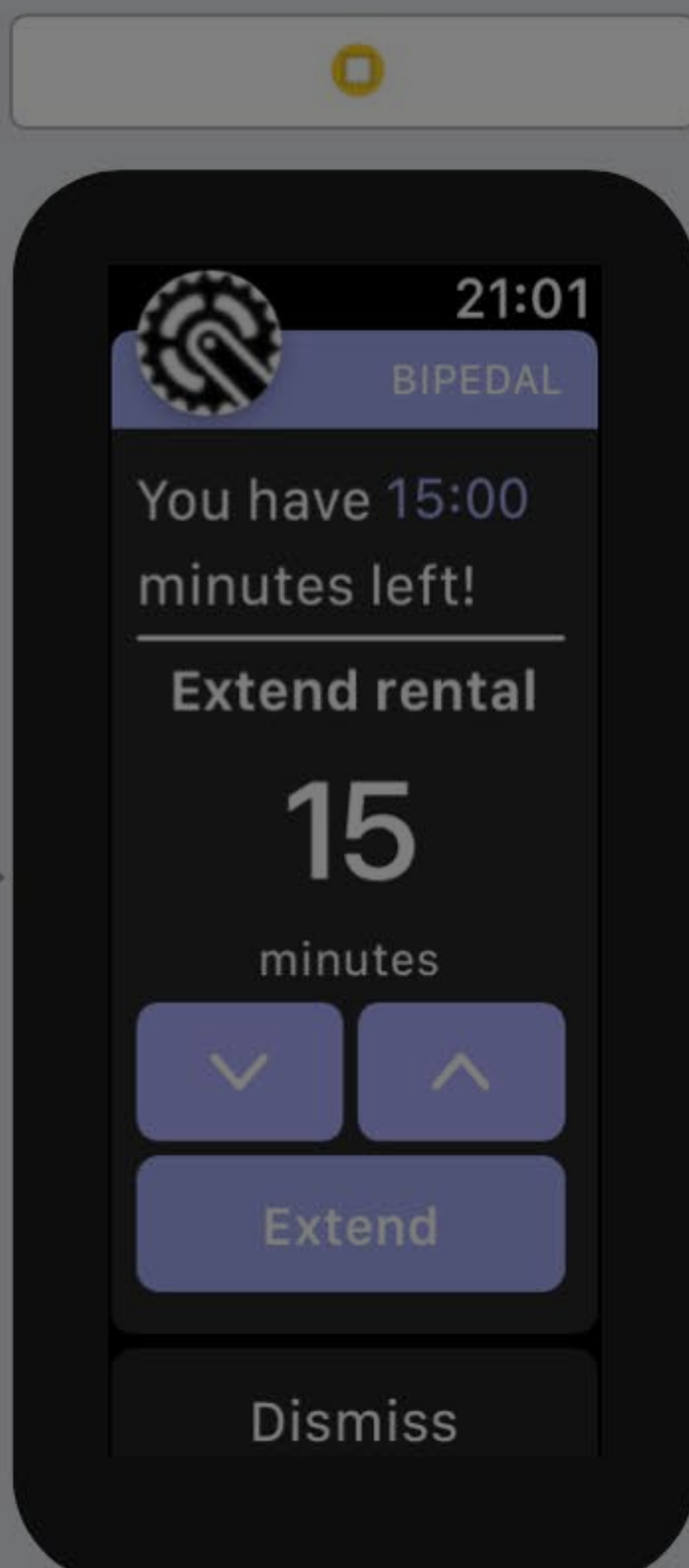
```
26
27 class DynamicInteractiveNotificationController: WKUserNotificationInterfaceController {
28
29     var endDate: Date!
30     var timeExtension: Int = 15
31
32     var countdownStopper: Timer?
33
34     @IBOutlet weak var countdownTimer: WKInterfaceTimer!
35     @IBOutlet weak var downButton: WKInterfaceButton!
36     @IBOutlet weak var upButton: WKInterfaceButton!
37     @IBOutlet weak var timeLabel: WKInterfaceLabel!
38     @IBOutlet weak var extendButton: WKInterfaceButton!
39
40     @IBAction func upButtonTapped() {
41         timeExtension += 15
42         updateTimeExtensionUI()
43     }
44
45     @IBAction func downButtonTapped() {
46         timeExtension -= 15
47         updateTimeExtensionUI()
48     }
49
50     @IBAction func extendButtonTapped() {
51         // code to process rental extension goes here
52         performDismissAction()
53     }
54
55     func launchApp() {
56         // TODO: wire this up to a button
57         performNotificationDefaultAction()
58     }
59
```


Interface.storyboard

Interface.storyboard > Interface.storyboard (Base) > No Selection

Automatic > NotificationController.swift > launchApp()

< 2 > + x



The storyboard preview shows a notification interface on a mobile device. At the top, there is a status bar with a yellow dot. The notification itself has a header with a logo and the time 21:01. Below the header, it says "BIPEDAL" and "You have 15:00 minutes left!". There is a section titled "Extend rental" with a large "15" in the center, followed by "minutes". Below this are two buttons: a blue button with a downward arrow and a blue button with an upward arrow. At the bottom of the notification are two buttons: "Extend" and "Dismiss".

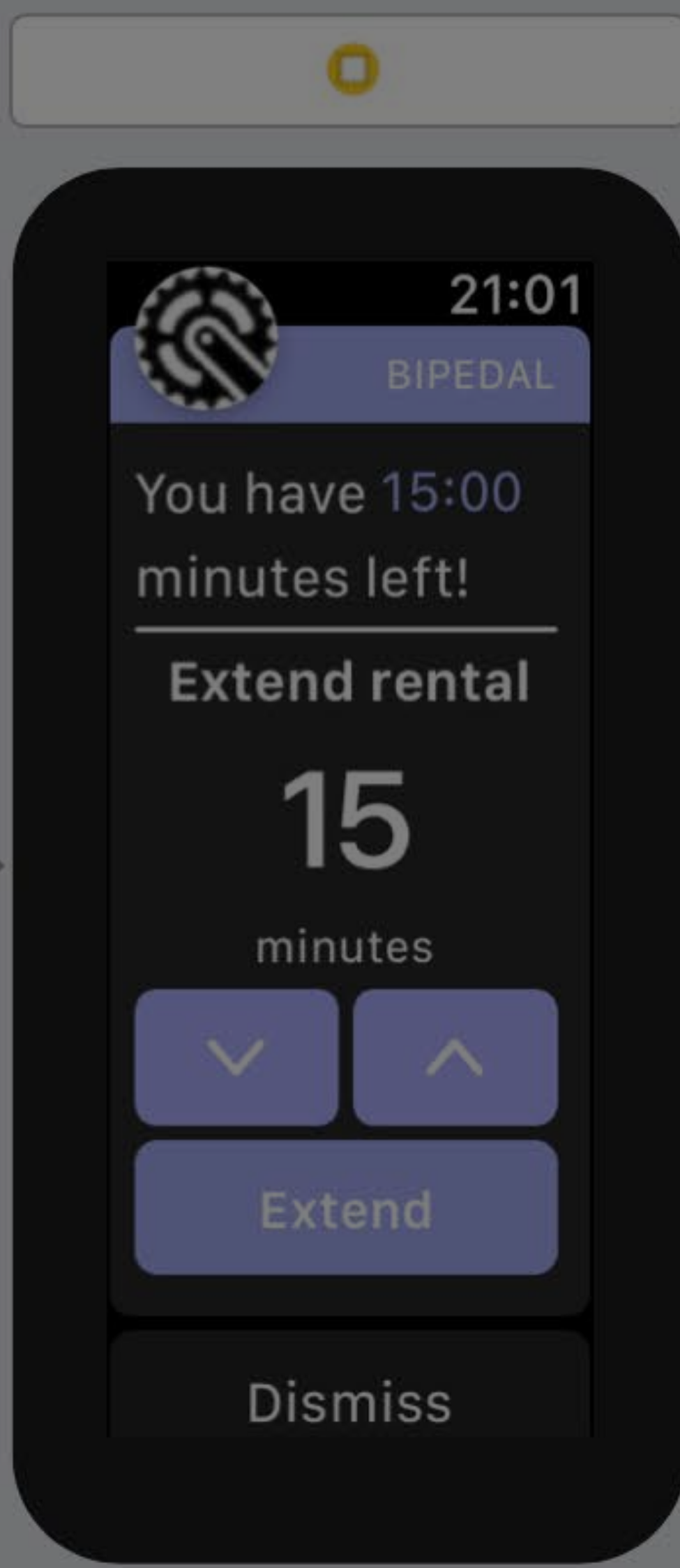
```
26
27 class DynamicInteractiveNotificationController: WKUserNotificationInterfaceController {
28
29     var endDate: Date!
30     var timeExtension: Int = 15
31
32     var countdownStopper: Timer?
33
34     @IBOutlet weak var countdownTimer: WKInterfaceTimer!
35     @IBOutlet weak var downButton: WKInterfaceButton!
36     @IBOutlet weak var upButton: WKInterfaceButton!
37     @IBOutlet weak var timeLabel: WKInterfaceLabel!
38     @IBOutlet weak var extendButton: WKInterfaceButton!
39
40     @IBAction func upButtonTapped() {
41         timeExtension += 15
42         updateTimeExtensionUI()
43     }
44
45     @IBAction func downButtonTapped() {
46         timeExtension -= 15
47         updateTimeExtensionUI()
48     }
49
50     @IBAction func extendButtonTapped() {
51         // code to process rental extension goes here
52         performDismissAction()
53     }
54
55     func launchApp() {
56         // TODO: wire this up to a button
57         performNotificationDefaultAction()
58     }
59
```


Interface.storyboard

Interface.storyboard (Base) > No Selection

Automatic > NotificationController.swift > extendButtonTapped()

< 2 > + x



The storyboard preview shows a notification card for 'BIPEDAL'. The card displays the time '21:01', the text 'You have 15:00 minutes left!', and a large '15 minutes' timer. Below the timer are 'Extend rental' and 'Dismiss' buttons. A hand icon is shown tapping the 'Extend rental' button.

```
26
27 class DynamicInteractiveNotificationController: WKUserNotificationInterfaceController {
28
29     var endDate: Date!
30     var timeExtension: Int = 15
31
32     var countdownStopper: Timer?
33
34     @IBOutlet weak var countdownTimer: WKInterfaceTimer!
35     @IBOutlet weak var downButton: WKInterfaceButton!
36     @IBOutlet weak var upButton: WKInterfaceButton!
37     @IBOutlet weak var timeLabel: WKInterfaceLabel!
38     @IBOutlet weak var extendButton: WKInterfaceButton!
39
40     @IBAction func upButtonTapped() {
41         timeExtension += 15
42         updateTimeExtensionUI()
43     }
44
45     @IBAction func downButtonTapped() {
46         timeExtension -= 15
47         updateTimeExtensionUI()
48     }
49
50     @IBAction func extendButtonTapped() {
51         // code to process rental extension goes here
52         performDismissAction()
53     }
54
55     func launchApp() {
56         // TODO: wire this up to a button
57         performNotificationDefaultAction()
58     }
59
```




ISS flyover
starts now.

Take Me To It

More Info

Dismiss

```
override fun didReceive(_ notification: UNNotification) {  
    let newActions =  
        [UNNotificationAction(identifier: "endRental", title: "End Rental", options: []),  
         UNNotificationAction(identifier: "help", title: "Get Help", options: .foreground)]  
  
    self.notificationActions = newActions  
  
    ...  
}
```



NEW


```
func updateUI() {  
    if endDate > Date() {  
        countdownTimer.setDate(endDate)  
        countdownTimer.start()  
    } else {  
        countdownTimer.setDate(Date())  
        countdownTimer.stop()  
        let newActions =  
            [UNNotificationAction(identifier: "payFine", title: "Pay Overtime", options: .foreground),  
             UNNotificationAction(identifier: "help", title: "Get Help", options: .foreground)]  
        self.notificationActions = newActions  
    }  
    ...  
}
```



10:09

YELP

Buckeye
Roadhouse



Your Reservation

2 people

7:00 pm, June 4



10:09

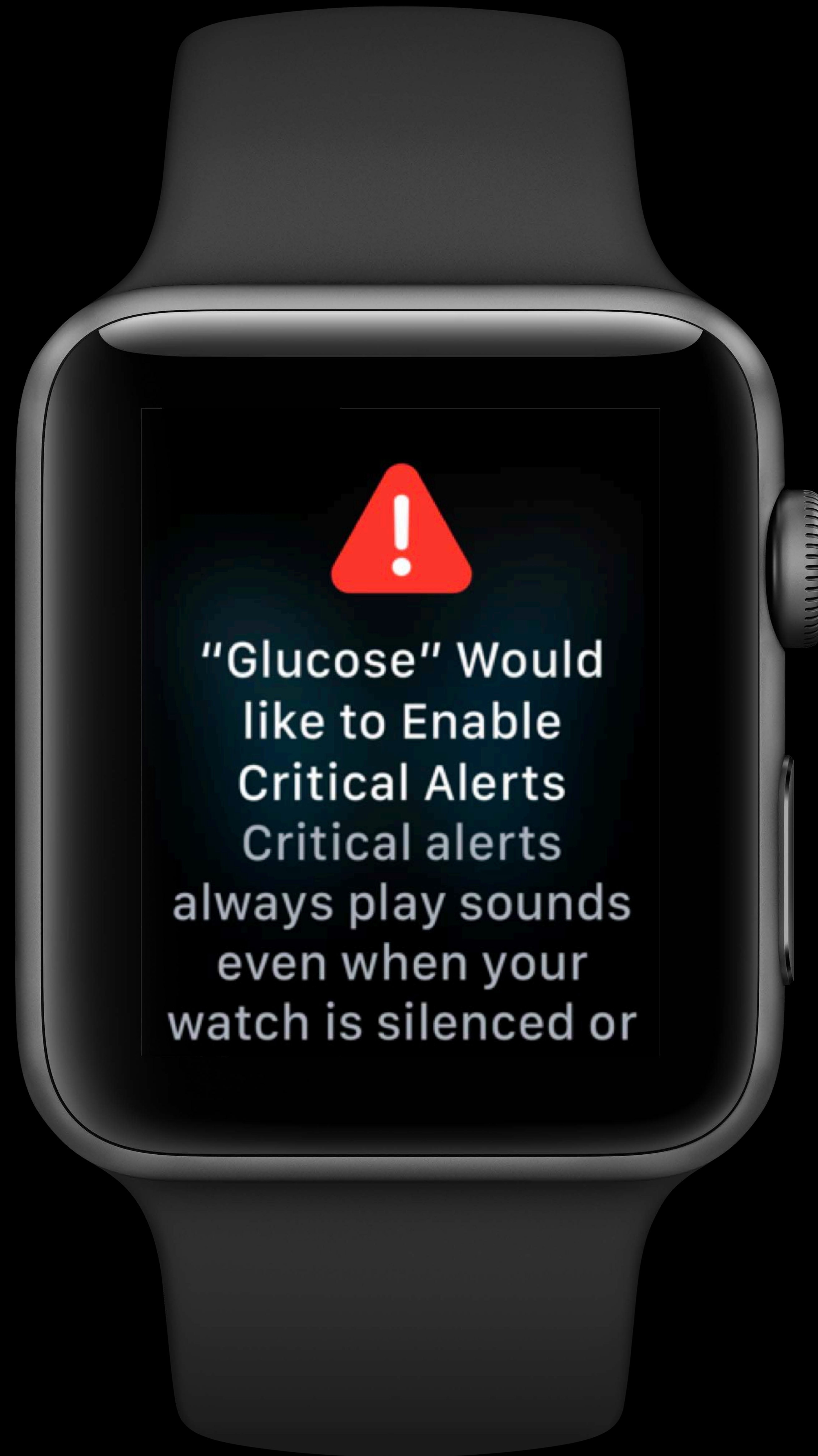


GLUCOSE

**Significant Drop
in Glucose.**

Blood sugar has
dropped 20 mg/dL.

View Graph




"Glucose" Would
like to Enable
Critical Alerts
Critical alerts
always play sounds
even when your
watch is silenced or



MON
4

10:09

 **Crash Boom Ban...**
Unloved - Crash Bo...
Music


11:23



WWDC
2018


```
// Request permission to deliver notifications directly to Notification Center
```

```
override fun awake(withContext context: Any?) {
```

```
    super.awake(withContext: context)
```

```
    UNUserNotificationCenter.current().requestAuthorization(options: [.provisional]) { (granted, error) in
```

```
        self.isGrantedAccess = granted
```

```
    }
```

```
}
```



BREATHE

A minute of deep breathing can improve concentration.

3:50 PM

Notifications

More dynamic

Optionally grouped

Interactive and actionable

Delivered with varying levels of urgency

What's New in User Notifications

Hall 3

Wednesday 2:00PM

Designing Notifications

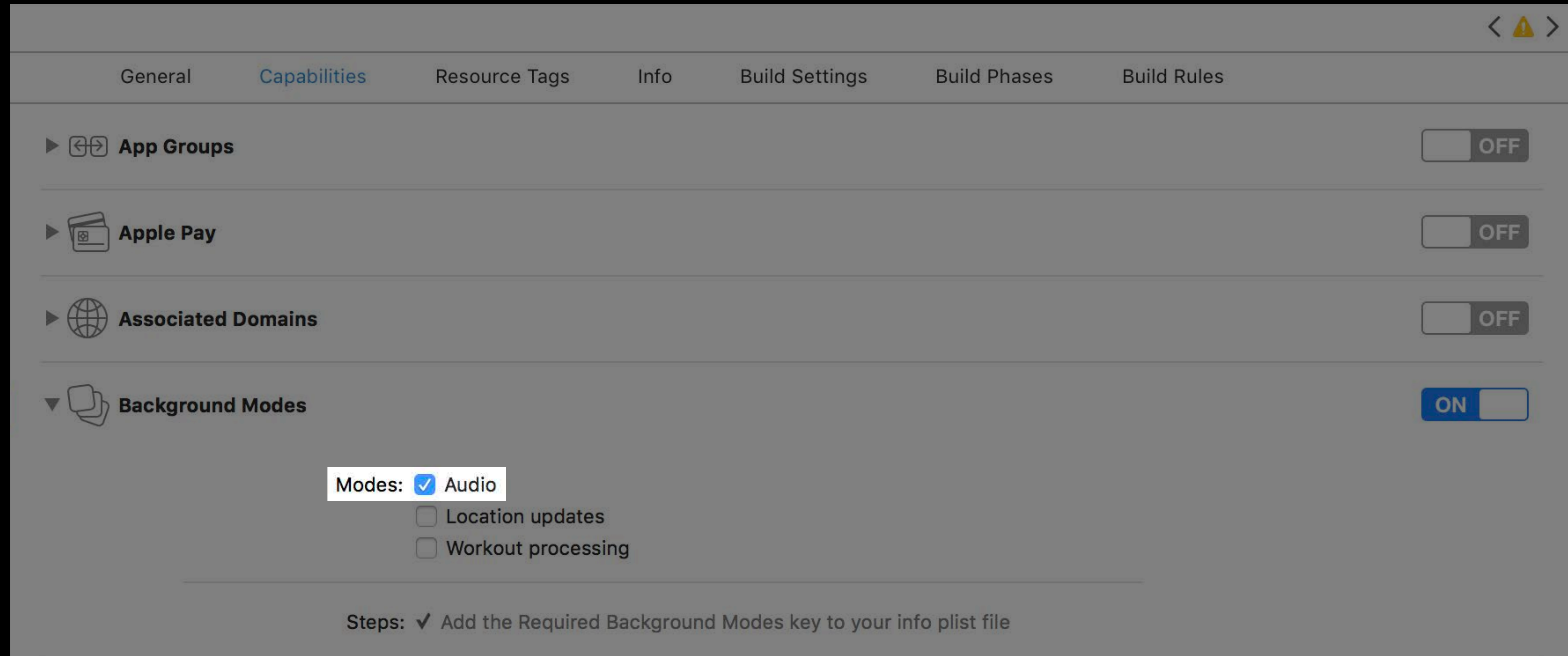
Hall 2

Friday 10:00AM

Watch Apps



Background Mode—Audio

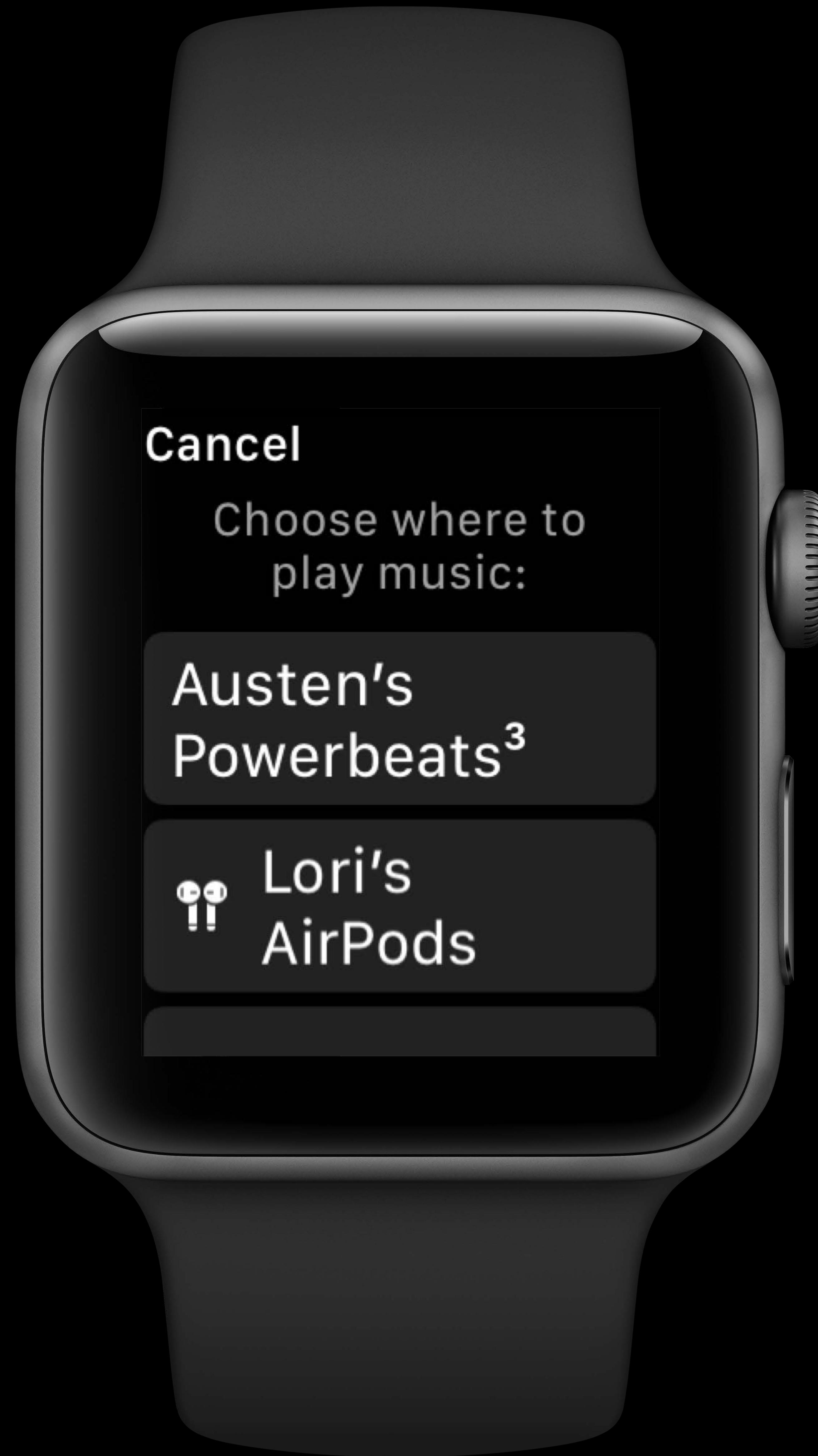


AVFoundation APIs

Direct access to `AVAudioPlayer` and `AVAudioEngine`

Share code between iOS and watchOS






Cancel

Choose where to
play music:

Austen's
Powerbeats³

 Lori's
AirPods

MediaPlayer.framework

```
var nowPlayingInfo: [String: Any] = [  
    MPMediaItemPropertyTitle: asset.title,  
    MPMediaItemPropertyArtist: asset.artist  
]  
  
MPNowPlayingInfoCenter.default().nowPlayingInfo = nowPlayingInfo
```



MediaPlayer.framework

```
let rcc: MPRemoteCommandCenter = MPRemoteCommandCenter.shared()

rcc.togglePlayPauseCommand.addTarget(self, action:
#selector(playPauseTarget(event:)))
rcc.nextTrackCommand.addTarget(self, action:
#selector(changeTrackTarget(event:)))
rcc.previousTrackCommand.addTarget(self, action:
#selector(changeTrackTarget(event:)))
rcc.likeCommand.addTarget(self, action: #selector(likeDislikeTarget(event:)))
rcc.dislikeCommand.addTarget(self, action:
#selector(likeDislikeTarget(event:)))
updateCommandCenter()
```



Audio Controls



Creating Audio Apps for watchOS

Executive Ballroom

Thursday 9:00AM

watchOS Runtime and Connectivity Lab

Technology Lab 9

Thursday 12:00PM



10:09



10:10.00

13 ACTIVE
CAL

27 TOTAL
CAL

--BPM 





Interface Controller

Identifier

+ Title

+ Is Initial Controller

+ Activity Indicator On Load

+ Always Bounce

+ Full Screen

+ Background

+ Mode

Animate

+ Color

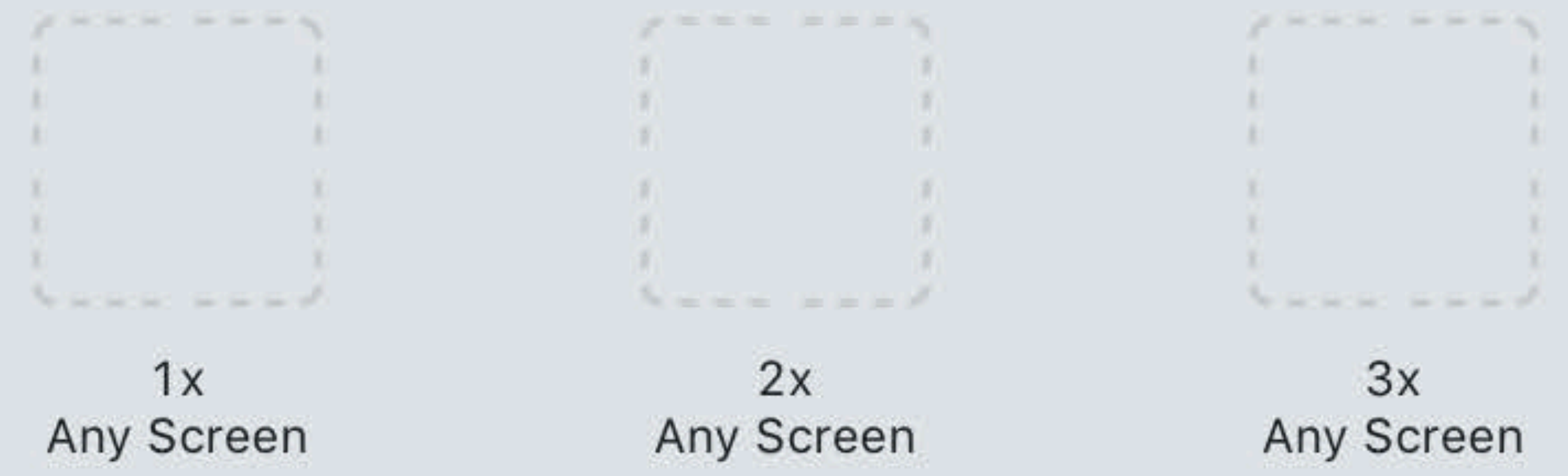
Insets

+ Spacing Custom

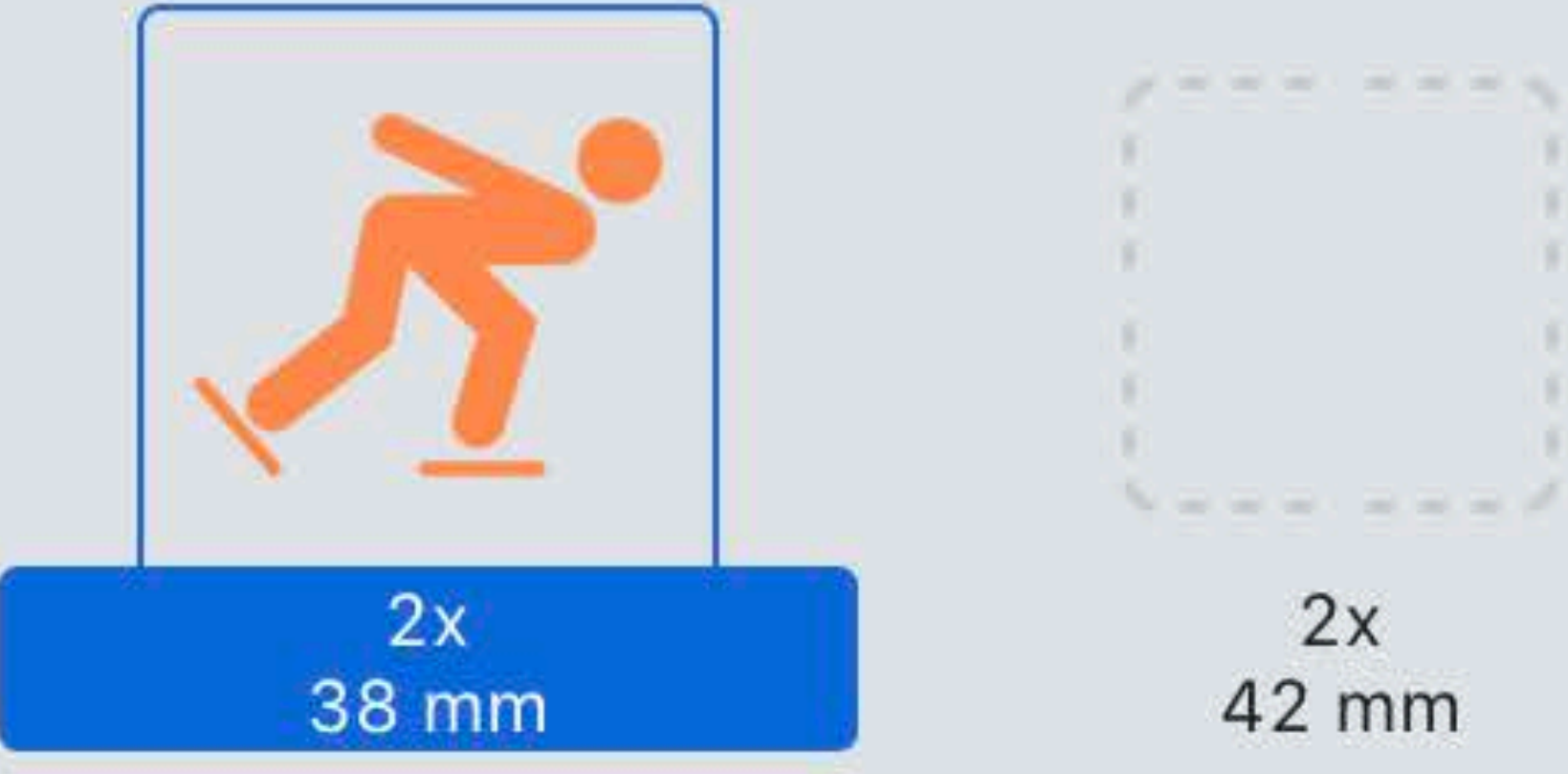
Page Direction

speedskater

Image



Universal



Apple Watch

Image Set

Name speedskater

Render As Default

Compression Inherited (Automatic)

Resizing Preserve Vector Data

- Devices
- Universal
 - iPhone
 - iPad
 - Car
 - Apple Watch
 - Apple TV
 - Mac

Appearances None

High Contrast

Scales Individual Scales

Gamut Any

Direction Fixed

Width Class Any

Height Class Any

- Memory
- 1 GB
 - 2 GB
 - 3 GB
 - 4 GB

- Graphics
- Metal 1v2
 - Metal 2v2
 - Metal 3v1
 - Metal 3v2
 - Metal 4v1

Screen Width 38mm and 42mm

Auto Scaling None

speedskater



1x
Any Screen

2x
Any Screen

3x
Any Screen

Universal



2x
38 mm



2x
42 mm

Apple Watch

Image

Image Set

Name

Render As

Compression

Resizing Preserve Vector Data

Devices Universal

iPhone

iPad

Car

Apple Watch

Apple TV

Mac

Appearances

High Contrast

Scales

Gamut

Direction

Width Class

Height Class

Memory 1 GB

2 GB

3 GB

4 GB

Graphics Metal 1v2

Metal 2v2

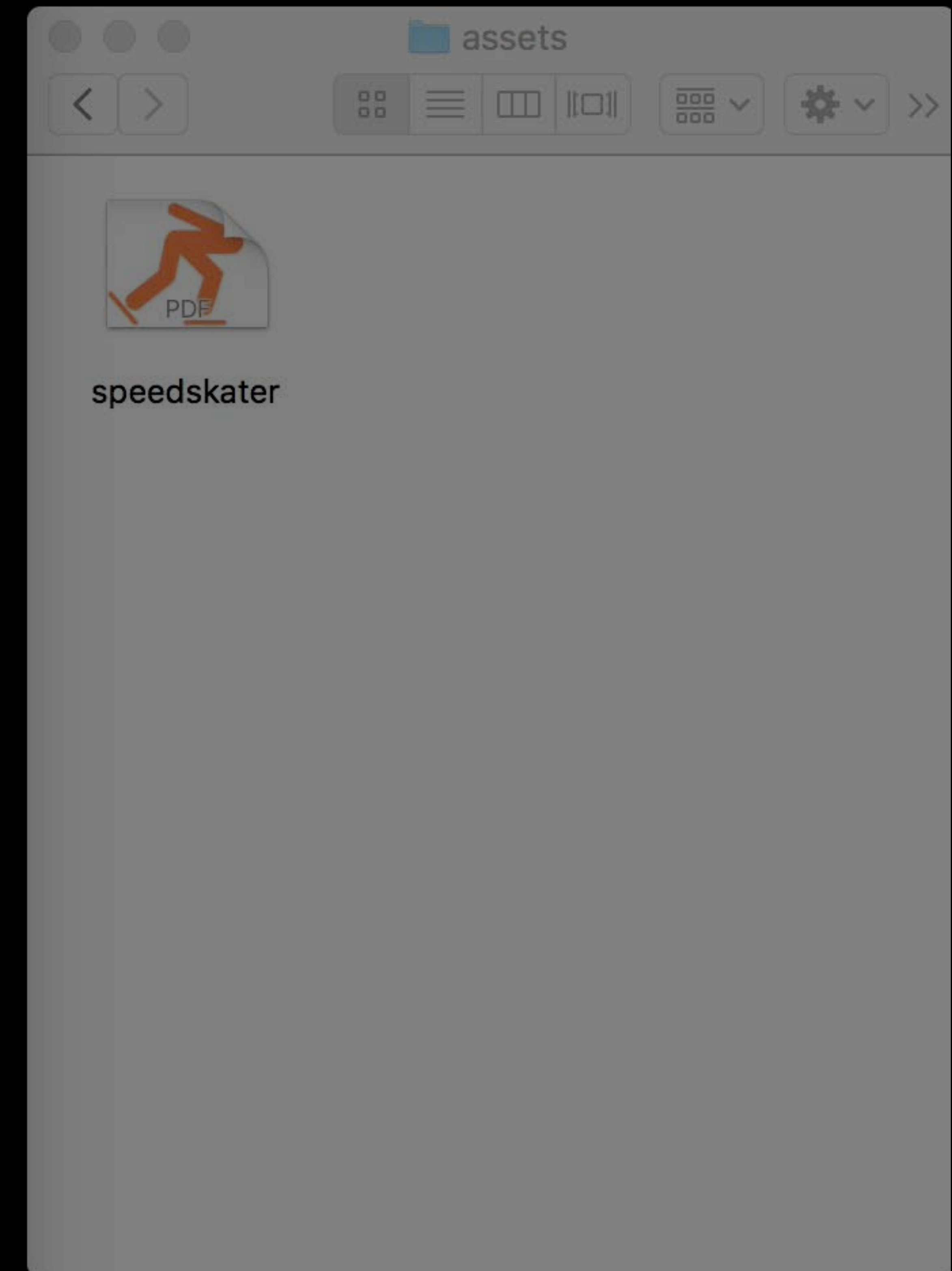
Metal 3v1

Metal 3v2

Metal 4v1

Screen Width

Auto Scaling





Label

- + Text: Take the ice!
- + Text Color: Default
- + Font: Title 3
- + Min Scale: [empty]
- + Baseline: Align Bas
- + Alignment: [empty]
- + Lines: [empty]

View

- + Alpha: [empty]
- + Hidden
- + Installed
- + Semantic: Unspecified

Alignment

- + Horizontal: Center
- + Vertical: Center

Size

- Width: Size To Fit Content
- Height: Size To Fit Content

Text Styles

- Body
- Caption 1
- Caption 2
- Footnote
- Headline
- Subhead
- Large Title
- Title 1
- Title 2
- ✓ Title 3

Family: Not Applicable

Style: Not Applicable

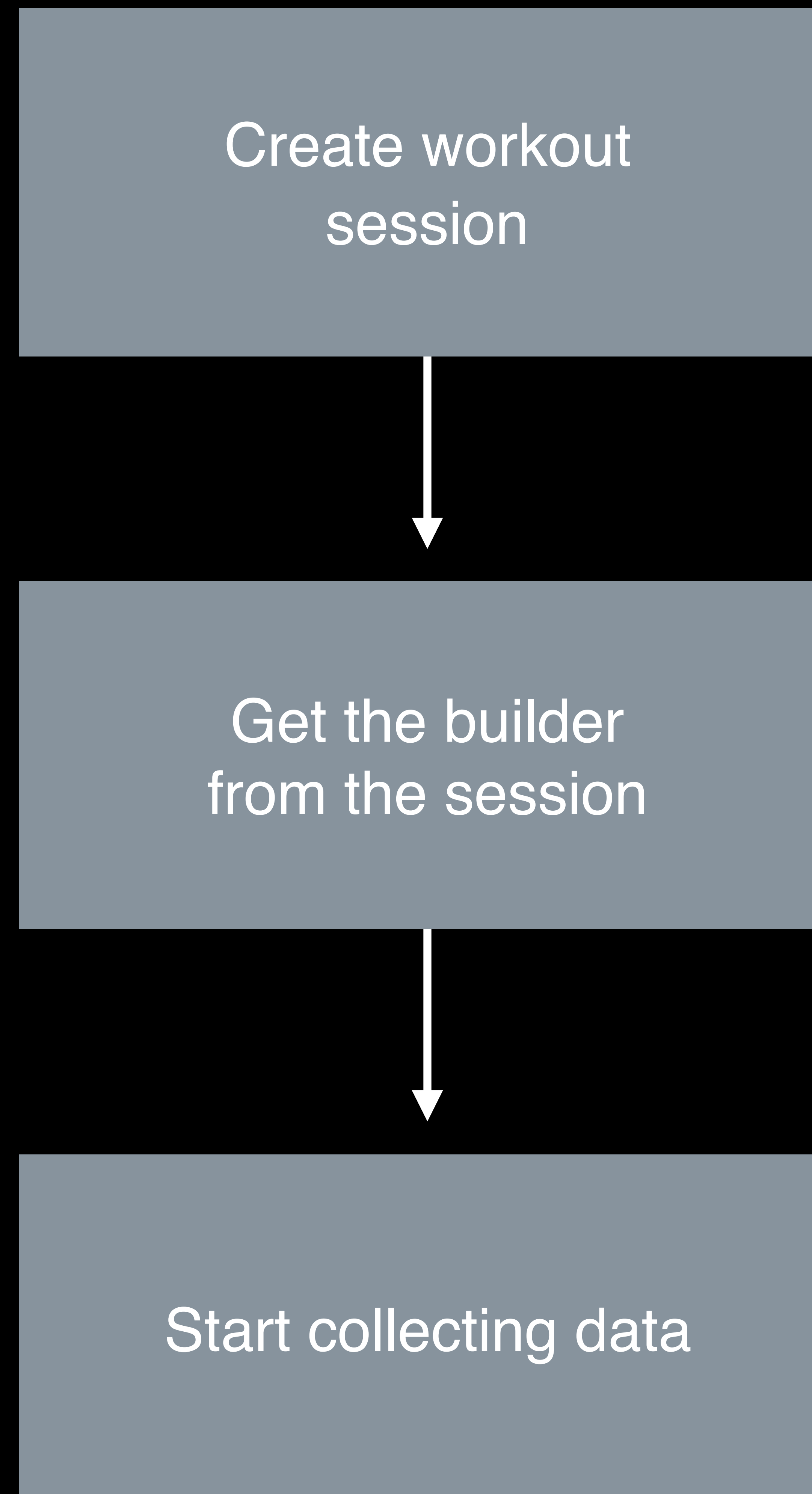
Size: Not Applicable

Done


```
// Use Large Title style for section titles
func setTitle(_ title: String) {
    setTitle(title, withFont: UIFont.preferredFont(forTextStyle: .largeTitle))
}
```




Start Me Up!



Start Me Up!

NEW

```
// Create session
let session = try HKWorkoutSession(healthStore: healthStore,
                                   configuration: workoutConfiguration)

// Retrieve builder
let builder = workoutSession.associatedWorkoutBuilder()

// Start
builder.beginCollection(withStart: nil, completion: { (success, error) in

// Handle error
})
```

(Re-)start Me Up!

NEW

Automatic relaunch after crash

```
// HKHealthStore.h
@available(watchOS 5.0, *)
open func recoverActiveWorkoutSession(completion: @escaping (HKWorkoutSession?, Error?) -> Void)
```

Session and builder restored in previous state

New Ways to Work with Workouts

Hall 3

Wednesday 9:00AM

Health and Fitness Technologies Lab

Technology Lab 1

Wednesday 1:00PM

Siri Shortcuts



MON 4

10:09



Audible



Resume

The Martian



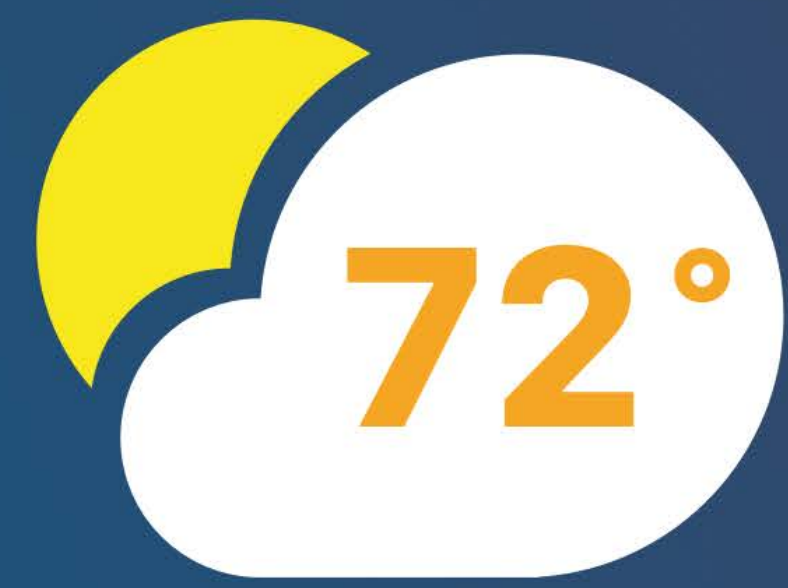
8:16PM

Sunset

6HR 17MIN



CARROT



Wow, much
clouds.



Citymapper



Get me to
Work



10% Happier


Start your evening
meditation



Lose It!



Had an apple?
Add it now

 **Glow Baby**
Did you finish
feeding Beanie?



< Feed

10:09



Save



10:09



Philz

**Is it Mint
Mojito Time?**

Order your usual
large, medium
cream, medium
sugar at Philz
Burlingame



INMediaPlaybackIntent

Up Next

Most Relevant

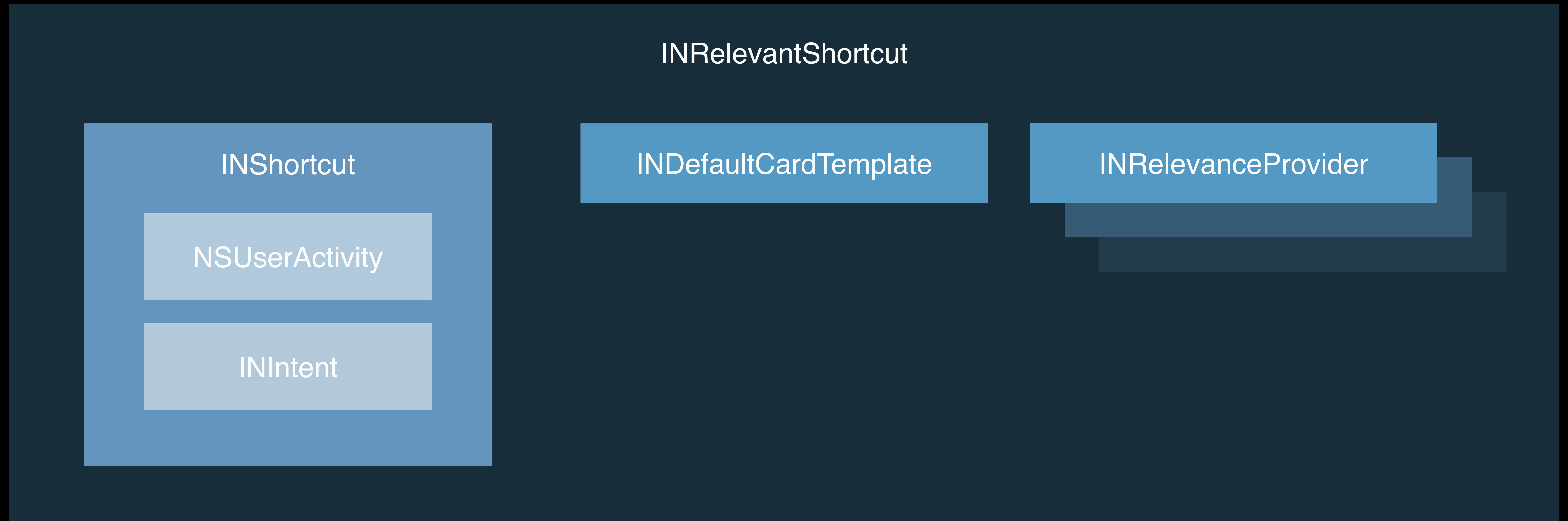
Least Relevant

Up Next

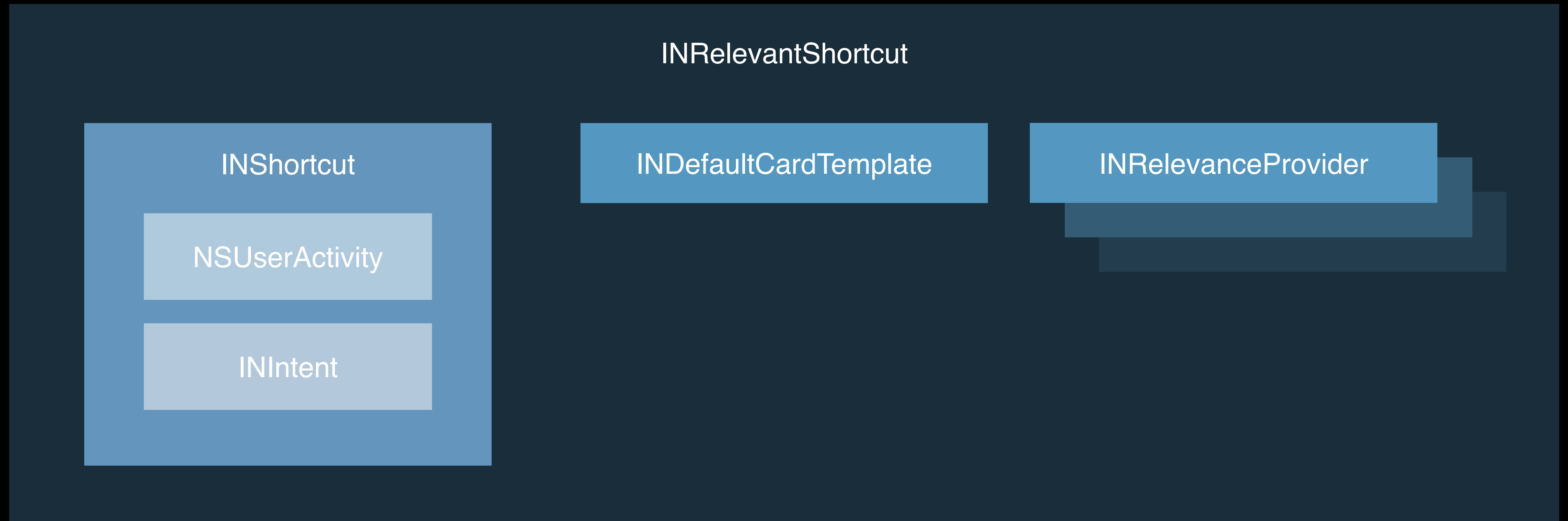
Most Relevant

Least Relevant

What's in a Relevant Shortcut?



What's in a Relevant Shortcut?



The Watch Template

 **Recipes**



Chickpea
Tagine

 **Ice Time**

Evening
9:15 PM

 **Art Around**

Five Skaters
Autumn Way

Relevance Providers

 **Recipes**



Chickpea
Tagine

 **Ice Time**

Evening
9:15 PM

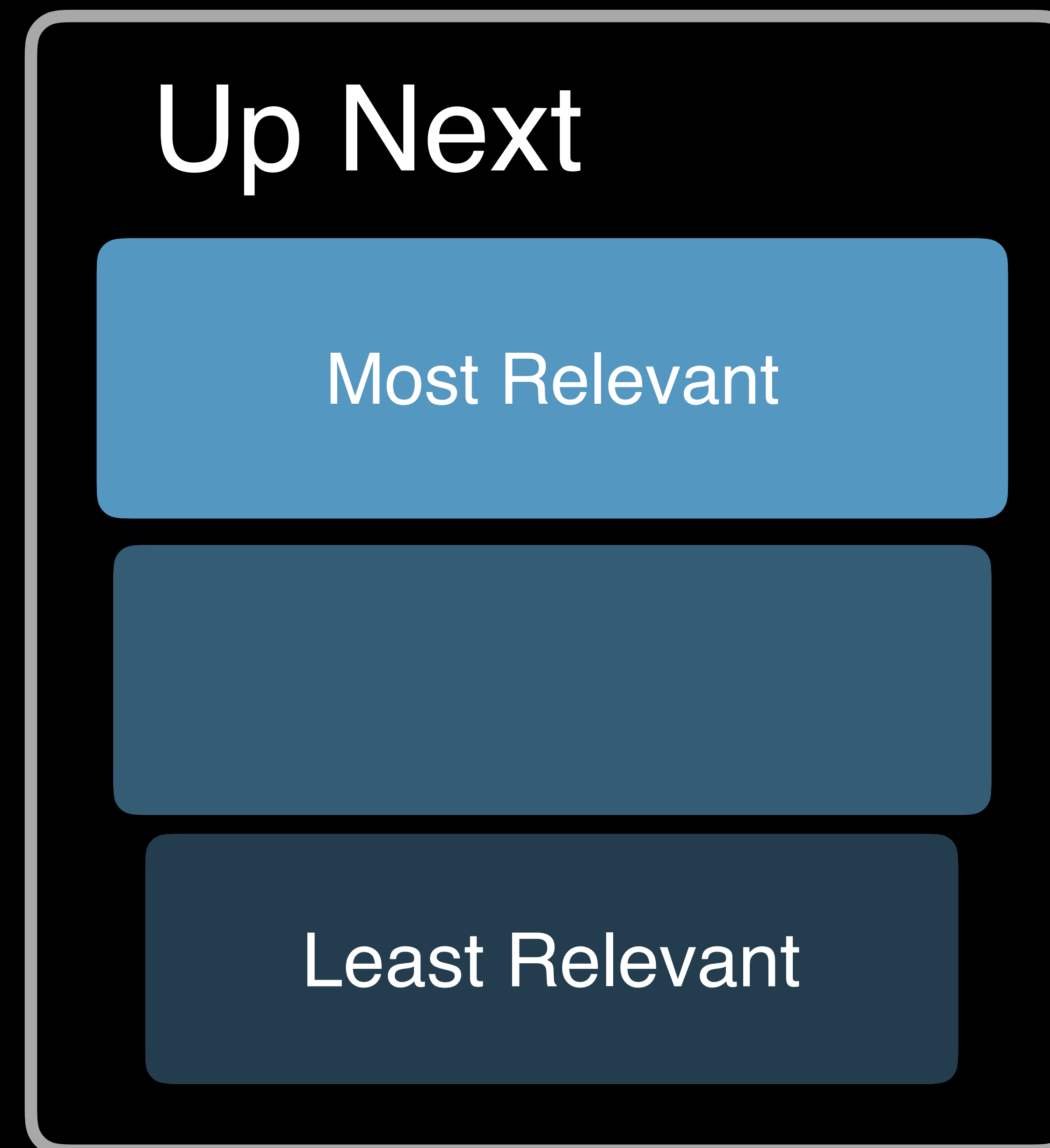
Time

 **Art Around**

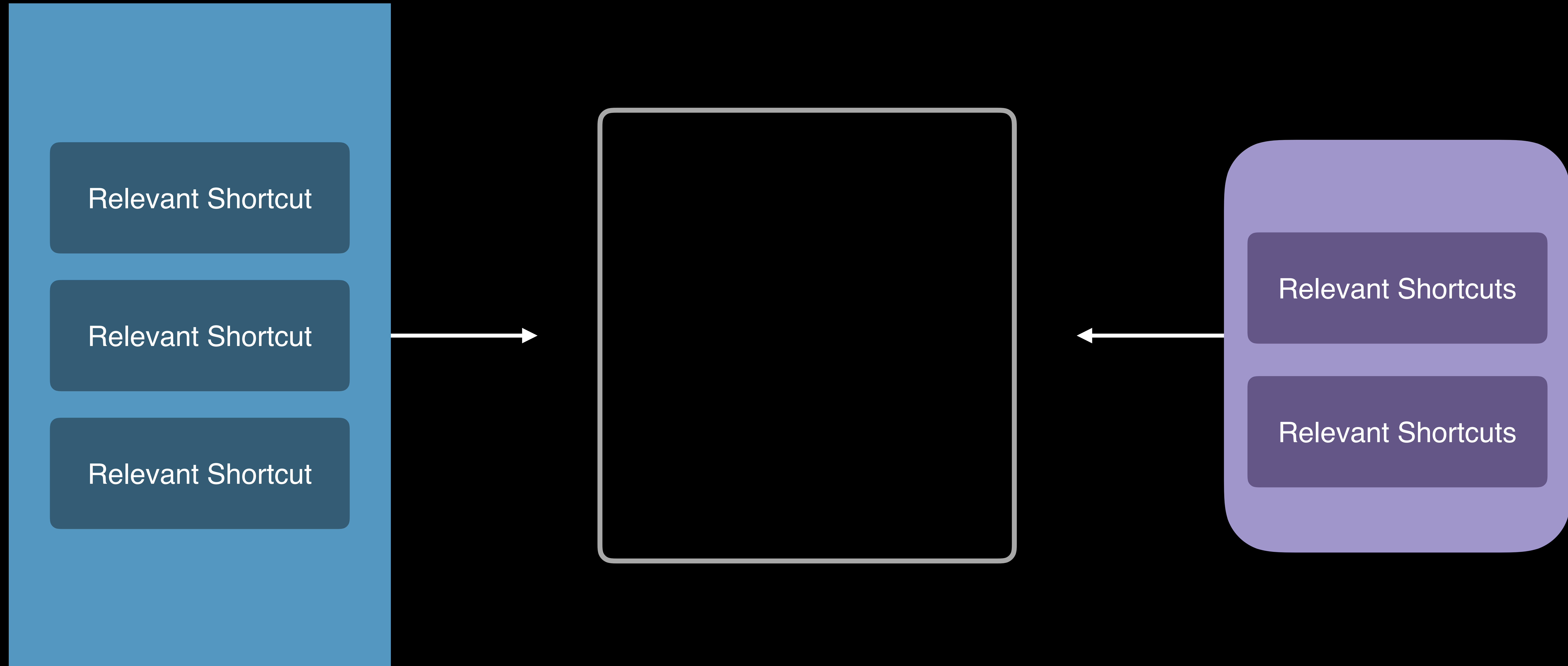
Five Skaters
Autumn Way

Location

Interaction Is Key



The Relevant Shortcut Store



iOS Relevant Shortcuts on Siri Face

Surfacing and executing iOS shortcuts requires

- Intent based
- Background execution
- No protected data needed

Updating Your Relevant Shortcuts

When the user launches your app

`WKRelevantShortcutRefreshBackgroundTask`

- Refresh data and relevant shortcuts
- Scheduled by user engagement

Updating Snapshots and Complications

`WKIntentDidRunRefreshBackgroundTask` after intent execution

Update appropriate UI



What can I help
you with?



10:09

 Philz

**Is it Mint
Mojito Time?**

Order your usual
large, medium
cream, medium
sugar at Philz
Burlingame

Siri Shortcuts

Provide engaging relevant shortcuts for the user

Supply relevant shortcuts from both iPhone and Apple Watch

Richest watchOS shortcut experience available with a watchOS app

Introduction to Siri Shortcuts

Hall 2

Tuesday 5:00PM

Building for Voice with Siri Shortcuts

Hall 2

Wednesday 10:00AM

Siri Shortcuts on the Siri Watch Face

Hall 2

Wednesday 11:00AM

Summary

Make notifications engaging and actionable

Bring audio into the background

Use the new `WorkoutBuilder` API

Provide value on the Siri watch face

More Information

<https://developer.apple.com/wwdc2018/206>

WatchKit, ClockKit, and Shortcuts Lab

Technology Lab 7

Wednesday 3:00PM

WatchKit and watchOS Notifications

Technology Lab 7

Friday 9:00AM

 **WWDC18**