

#WWDC19

SwiftUI on watchOS

Matthew Koonce, watchOS Frameworks Engineer

Josh Weinberg, watchOS Frameworks Engineer

watchOS

watchOS

SiriKit

Taptic Engine

PassKit

Native apps

Interactive notifications

HealthKit

Background App Refresh

Complications

Audio recording

SpriteKit

Background modes

Digital Crown

SceneKit

ClockKit

MapKit

Navigation session

NSURLSession

Single process runtime

watchOS

SiriKit

Taptic Engine

PassKit

Native apps

Interactive notifications

HealthKit

Background App Refresh

Complications

Audio recording

SpriteKit

Background modes

Digital Crown

SceneKit

ClockKit

MapKit

Navigation session

NSURLSession

Single process runtime

Asset and architecture thinning

Health authorization

Smart alarms

SiriKit

PassKit

Taptic Engine

Native apps

Apple Neural Engine

Interactive notifications

Independent apps

HealthKit

Background App Refresh

Improved debugging

Complications

Streaming audio

Audio recording

Self-care

watchOS 6

SpriteKit

watchOS App Store

Background modes

Digital Crown

SceneKit

Watch push target

Mindfulness

Physical therapy

Heart monitoring

Extended runtime

ClockKit

MapKit

Single process runtime

Navigation session

NSURLSession

Continuity keyboard

Tinted complications

Sign in with Apple

Asset and architecture thinning

Health authorization

Smart alarms

SiriKit

PassKit

Taptic Engine

Native apps

Apple Neural Engine

Interactive notifications

HealthKit

Independent apps

Background App Refresh

Improved debugging

Complications

Streaming audio

Audio recording

Self-care

watchOS App Store

watchOS 6

SpriteKit

Background modes

Digital Crown

SceneKit

Watch push target

Mindfulness

Physical therapy

Heart monitoring

Extended runtime

ClockKit

MapKit

Single process runtime

Navigation session

NSURLSession

Continuity keyboard

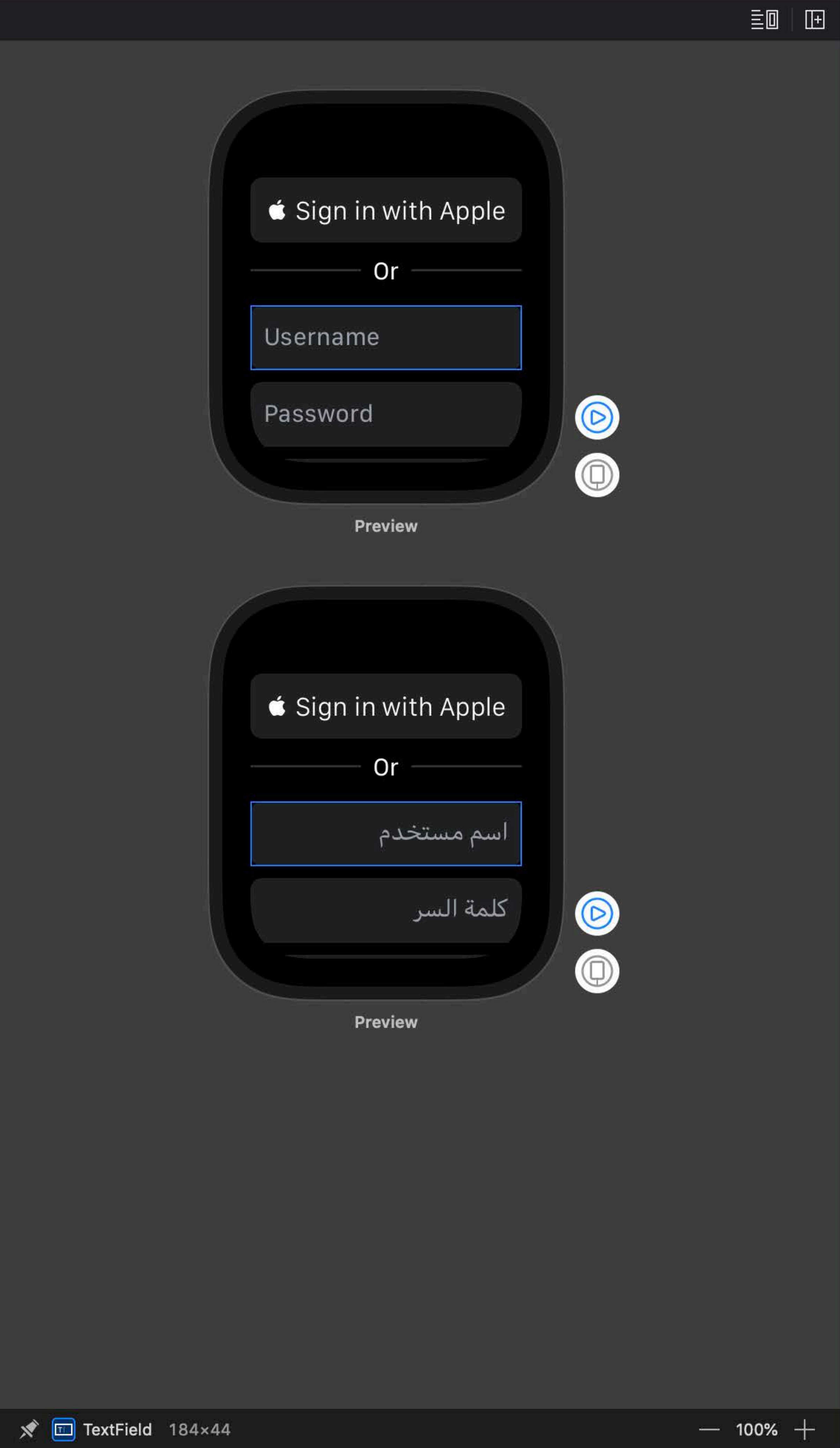
Tinted complications

Sign in with Apple



- Travel
- Travel Watch
 - Interface.storyboard
 - Assets.xcassets
 - Info.plist
- Travel Watch Extension
 - HostingController.swift
 - ExtensionDelegate.swift
 - SignInView.swift
 - SignInOniPhoneView.swift
 - Assets.xcassets
 - Info.plist
 - Localizable.strings
 - Preview Content
 - Products
 - Frameworks

```
1 //
2 // SignInView.swift
3 // Travel Watch Extension
4 //
5
6 import SwiftUI
7
8 struct SignInView : View {
9
10     @State var username = ""
11     @State var password = ""
12
13     var body: some View {
14         ScrollView {
15             VStack {
16                 SignInWithAppleButton() {
17                     // execute the Apple sign in
18                     // flow
19                 }
20
21                 HorizontalDivider(
22                     label: Text("Or")
23                 )
24
25                 TextField($username, placeholder:
26                     Text("Username"))
27                     .textContentType(.username)
28                 SecureField($password,
29                     placeholder: Text("Password"))
30                     .textContentType(.password)
31
32                 Button(action: {
33                     // execute the sign in flow
34                 }) {
35                     Text("Sign In")
36                 }
37             }
38         }
39     }
40 }
```



TextField

Placeholder: \$username

Placeholder: Text

Modifiers

Padding: Inherited

Frame: Width: inherited, Height: inherited

Text Content Type: Username









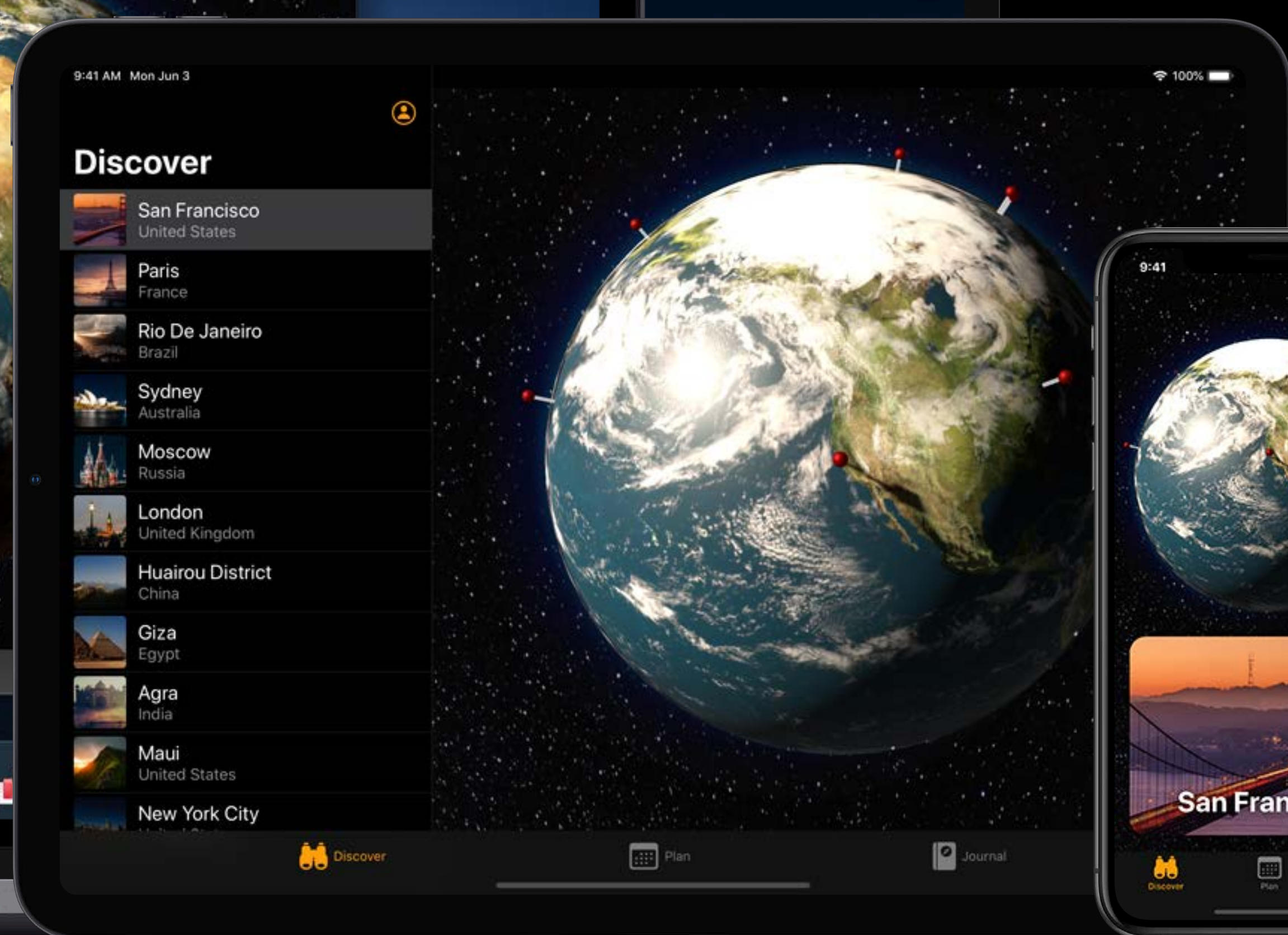
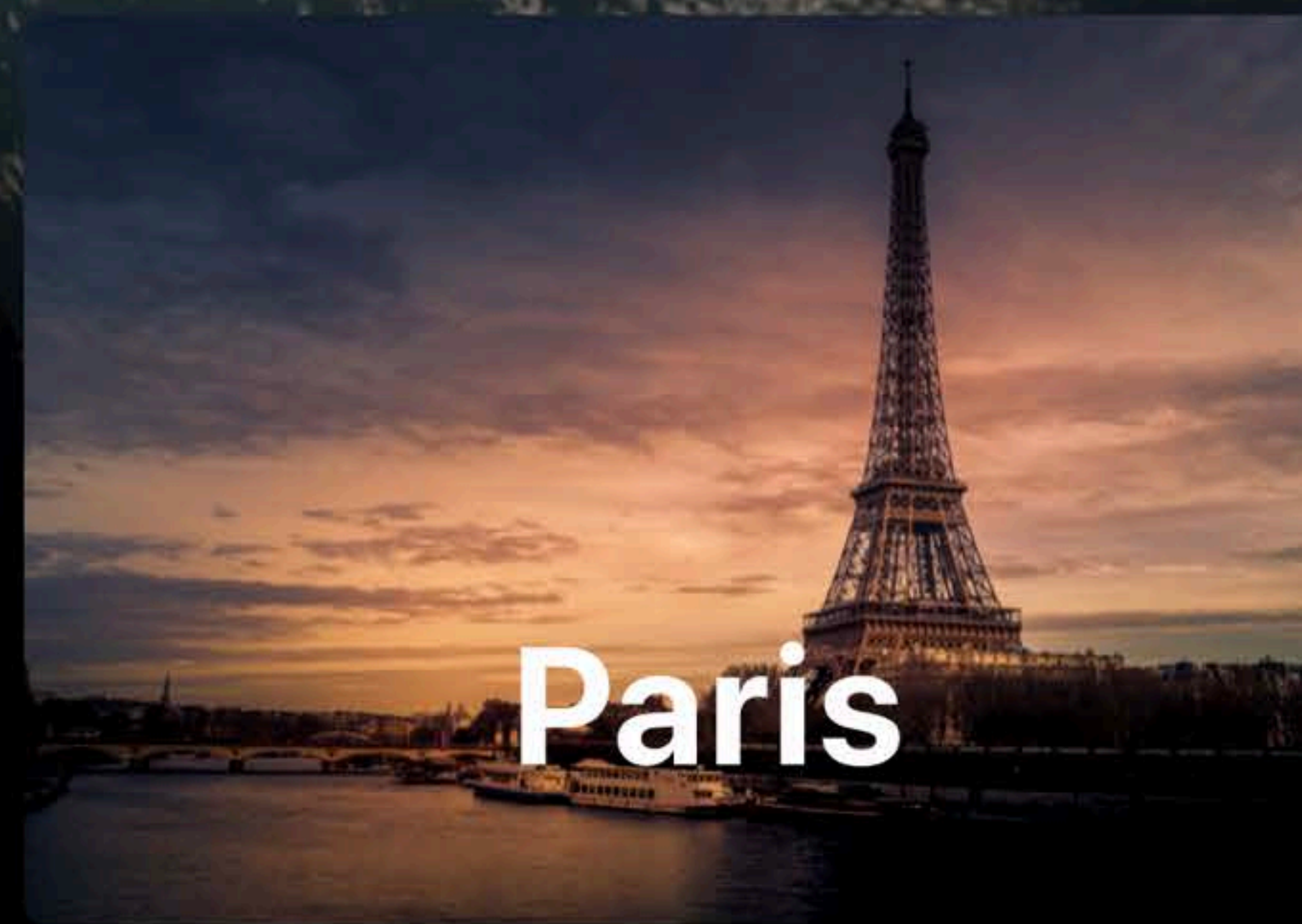
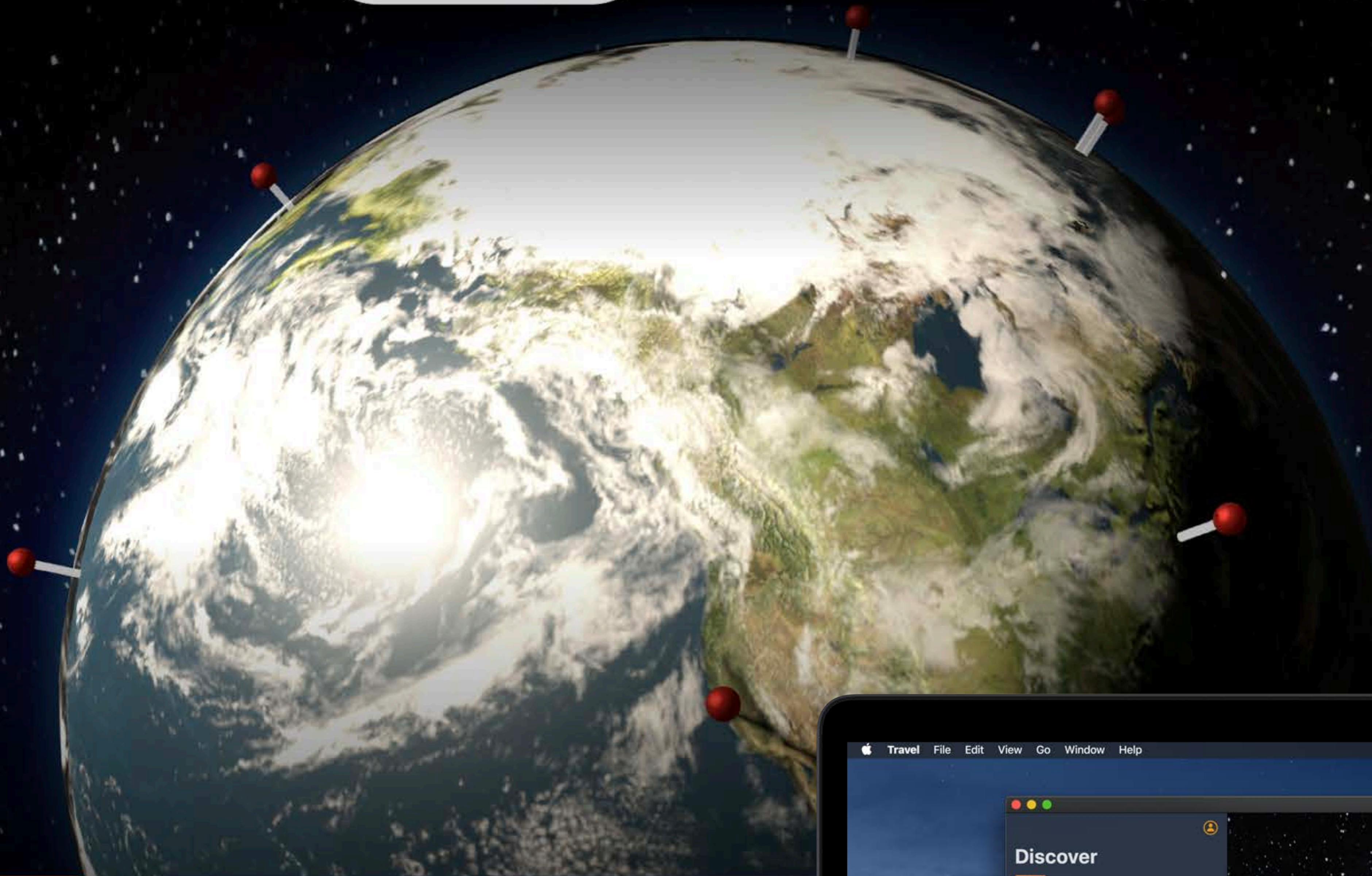




Discover

Plan

Journal













Contextual and Timely



Legible



Legible



Quick Interactions



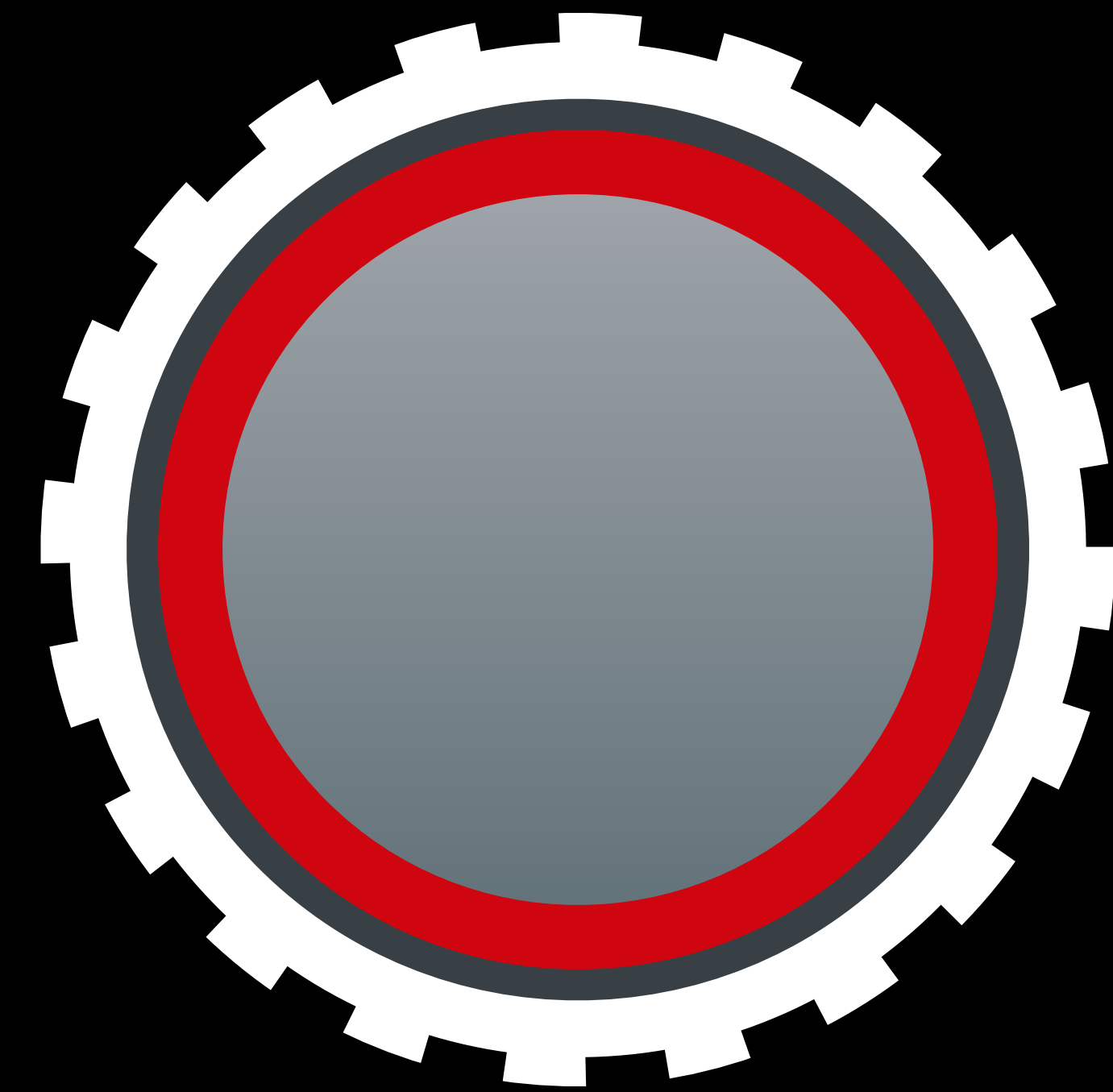
Quick Interactions

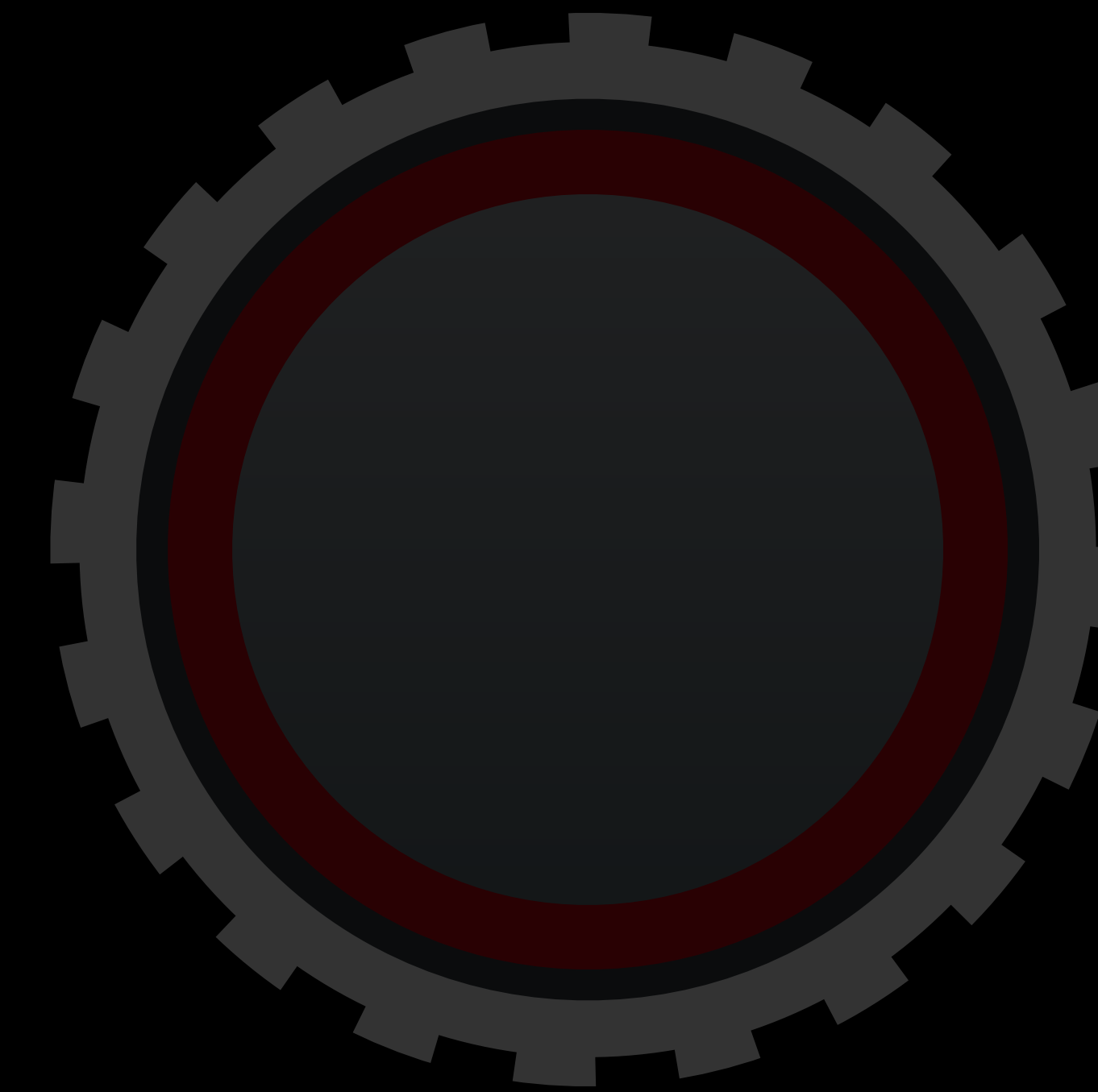










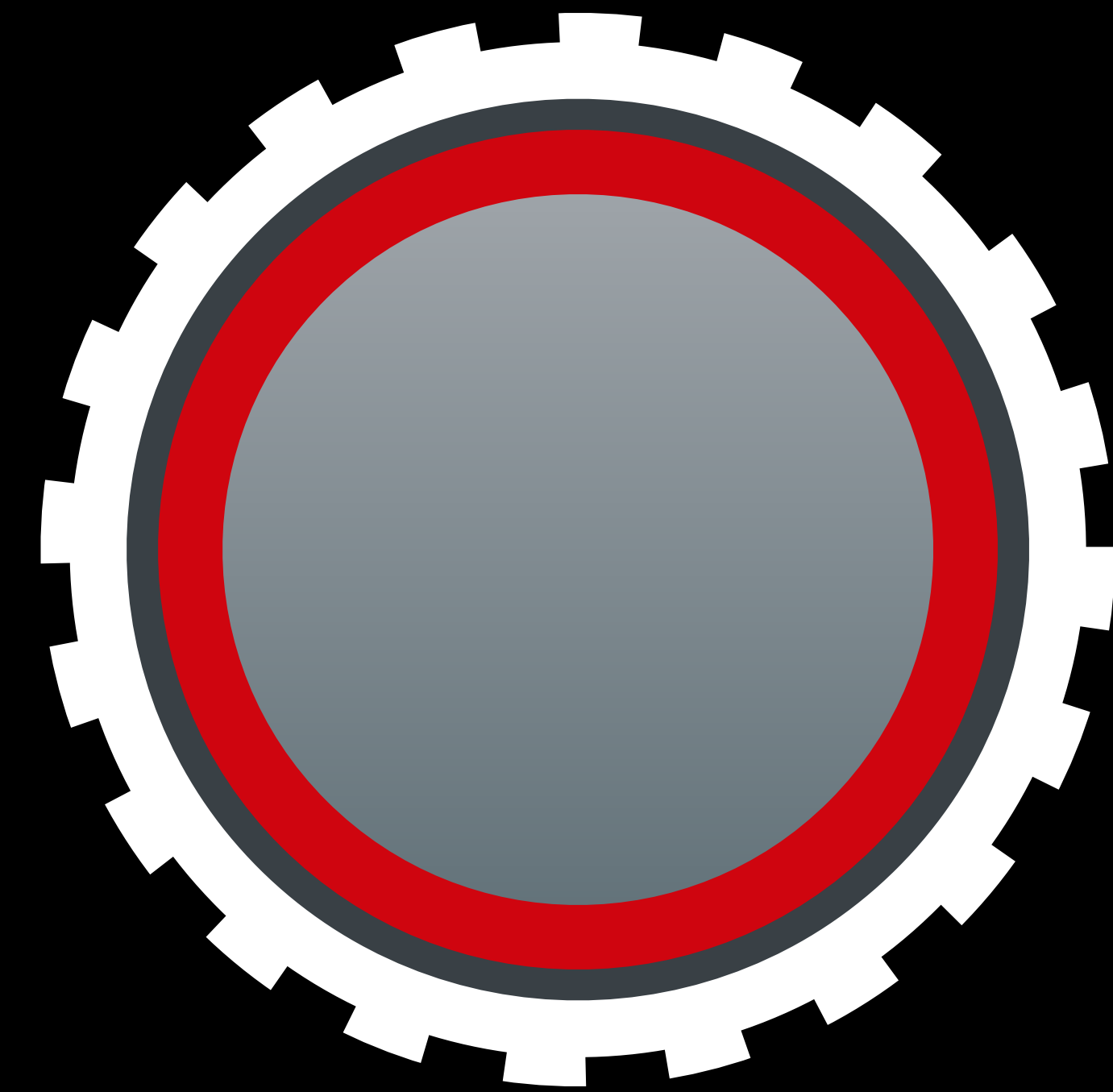


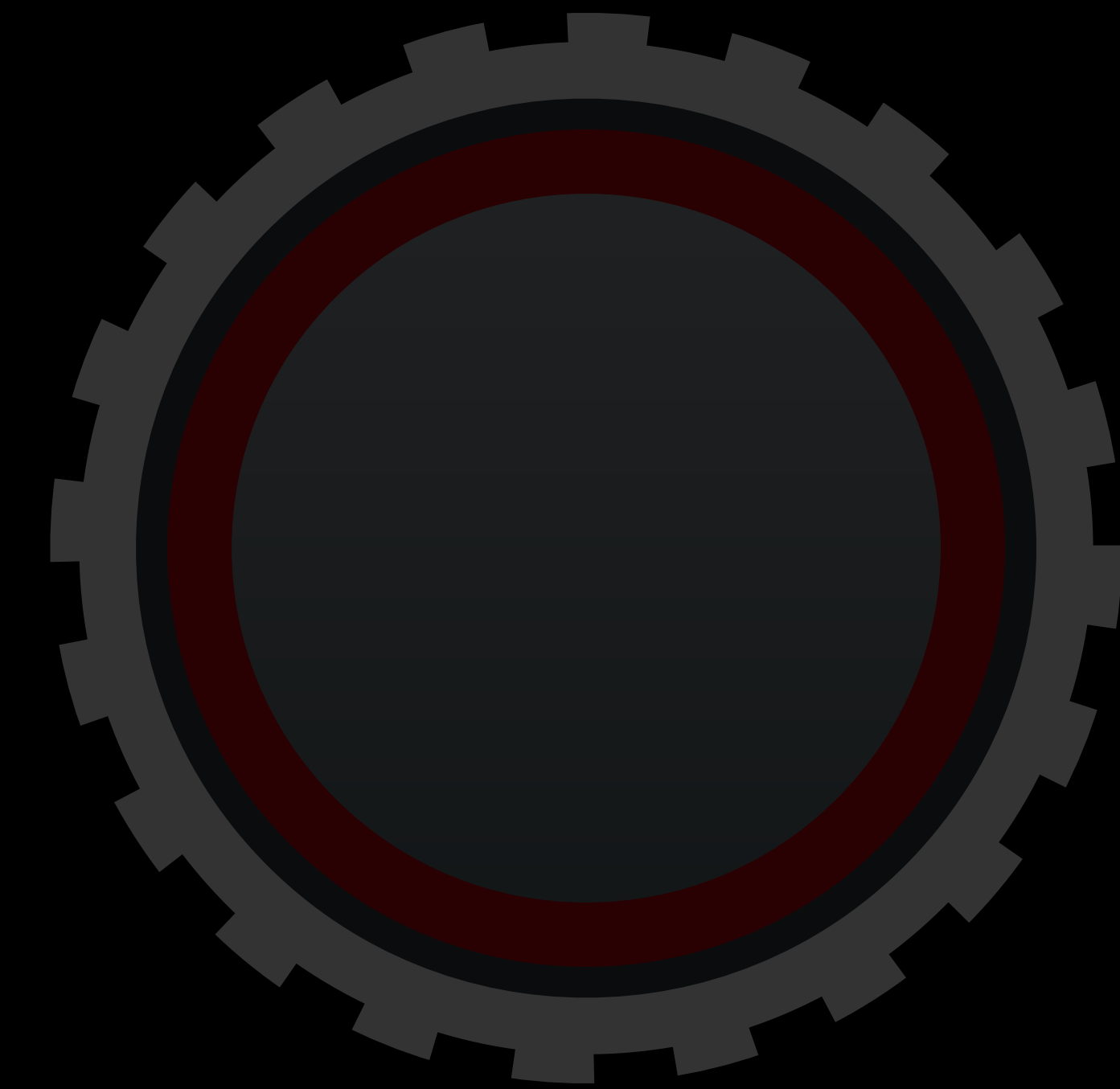
Demo

#WWDC19

Next-Level Customizations

Josh Weinberg, watchOS Frameworks Engineer









Your Reservation

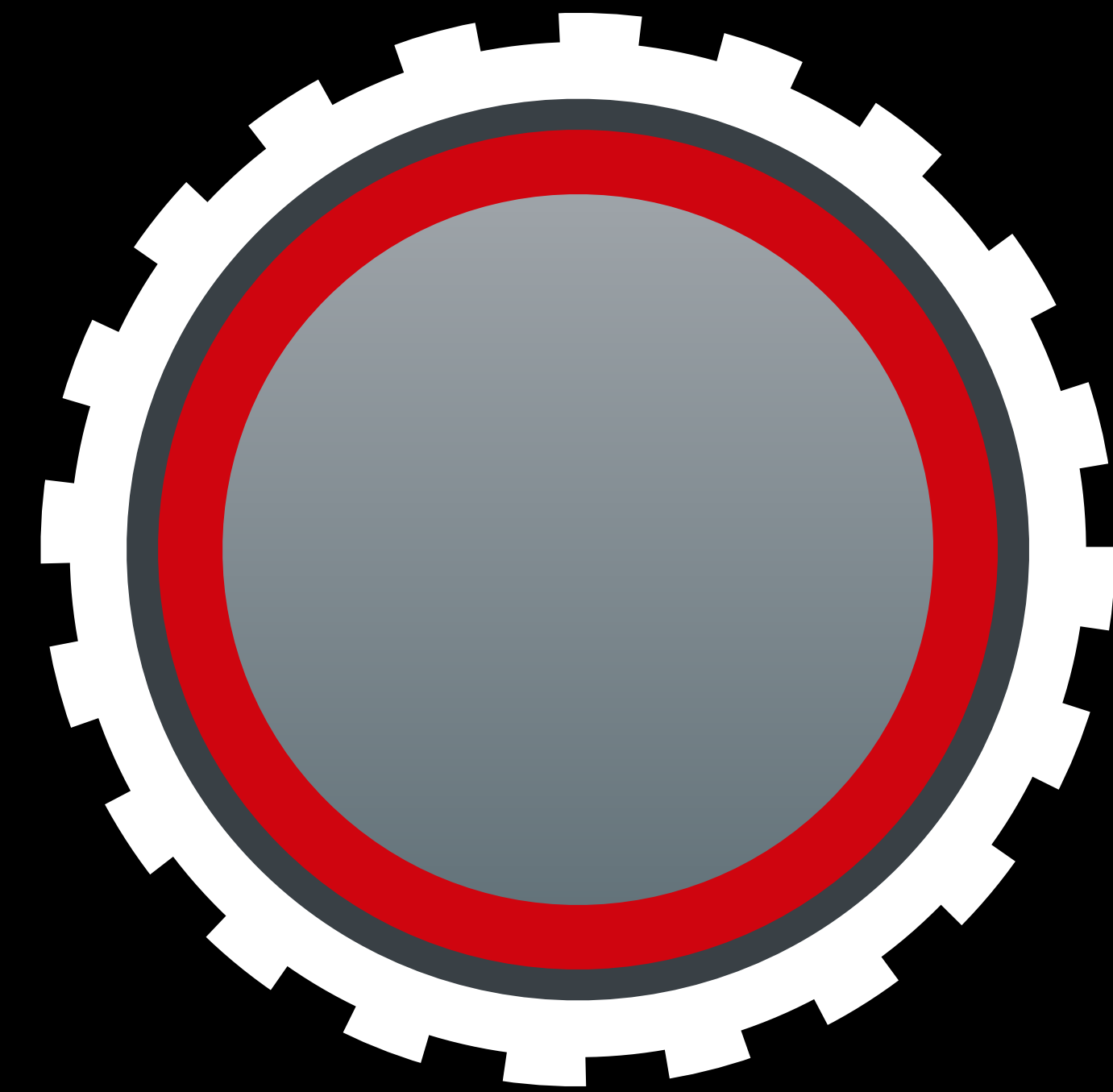
YELP

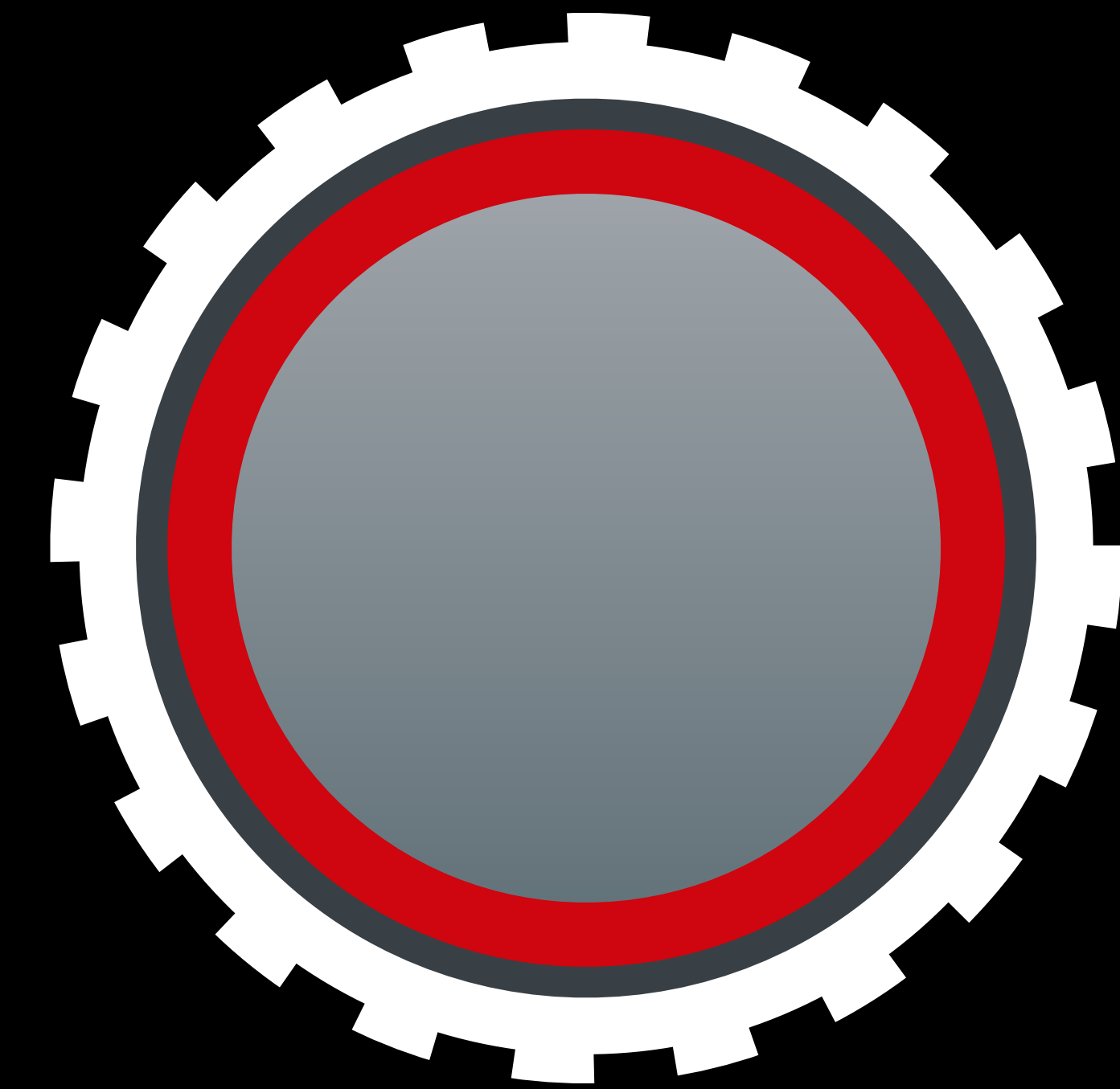
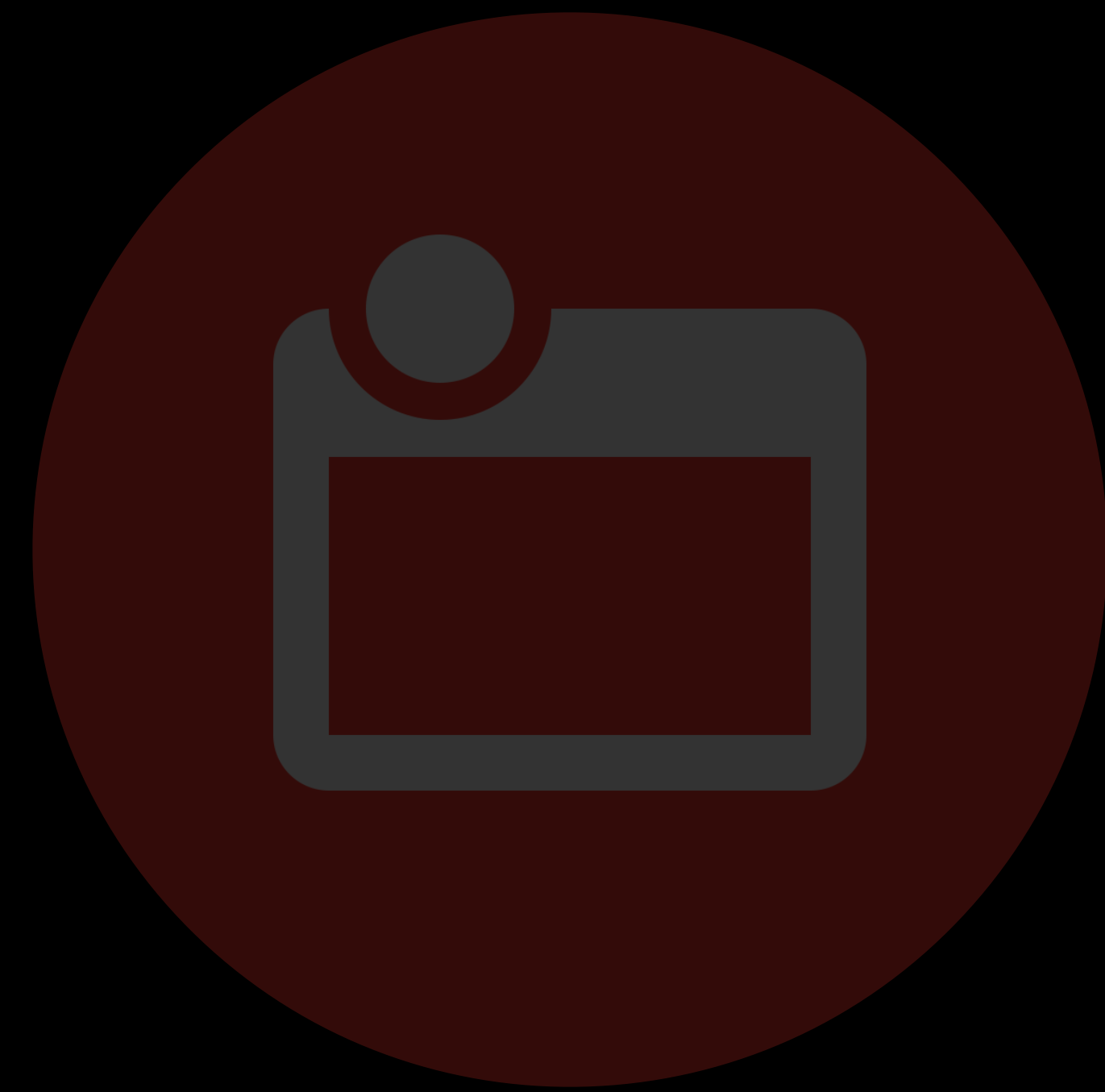


Your Reservation

YELP

Demo





















← Today

10:09

Coffee with Tiffany
The Coffee Shop



← Today

10:09

Coffee with Tiffany
The Coffee Shop



← Today

10:09

Coffee with Tiffany
The Coffee Shop

Digital Crown



```
.digitalCrownRotation(  
    Binding<BinaryFloatingPoint>  
    from: BinaryFloatingPoint  
    through: BinaryFloatingPoint  
)
```

Digital Crown



```
.digitalCrownRotation(  
    Binding<BinaryFloatingPoint>  
    from: BinaryFloatingPoint  
    through: BinaryFloatingPoint  
)
```

Digital Crown



```
.digitalCrownRotation(  
  $offset  
  from: BinaryFloatingPoint  
  through: BinaryFloatingPoint  
)
```

Digital Crown



```
.digitalCrownRotation(  
  $offset  
  from: BinaryFloatingPoint  
  through: BinaryFloatingPoint  
)
```

Digital Crown



```
.digitalCrownRotation(  
  $offset  
  from: 0.0  
  through: BinaryFloatingPoint  
)
```

Digital Crown



```
.digitalCrownRotation(  
  $offset  
  from: 0.0  
  through: BinaryFloatingPoint  
)
```


Digital Crown



```
.digitalCrownRotation(  
  $offset  
  from: 0.0  
  through: size.height  
)
```

Digital Crown



```
.digitalCrownRotation(  
  $offset  
  from: 0.0  
  through: size.height  
)
```

Digital Crown



```
.digitalCrownRotation(  
  $offset  
  from: 0.0  
  through: size.height  
)
```

Digital Crown



```
.digitalCrownRotation(  
  $offset  
  from: 0.0  
  through: size.height  
)
```

Digital Crown



```
.digitalCrownRotation(  
  $offset  
  from: 0.0  
  through: size.height  
)
```



×

1,138

18% Tip

204.84

People

1

1,342.84

Total



x

1,138

18% Tip

204.84

People

1

1,342.84

Total

Digital Crown



```
.digitalCrownRotation(  
    Binding<BinaryFloatingPoint>  
    from: BinaryFloatingPoint  
    through: BinaryFloatingPoint  
    by: BinaryFloatingPoint  
)
```


Digital Crown



```
.digitalCrownRotation(  
    Binding<BinaryFloatingPoint>  
    from: BinaryFloatingPoint  
    through: BinaryFloatingPoint  
    by: BinaryFloatingPoint  
)
```

Digital Crown



```
.digitalCrownRotation(  
  $people  
  from: 1.0  
  through: 15.0  
  by: BinaryFloatingPoint  
)
```

Digital Crown



```
.digitalCrownRotation(  
  $people  
  from: 1.0  
  through: 15.0  
  by: BinaryFloatingPoint  
)
```

Digital Crown



```
.digitalCrownRotation(  
  $people  
  from: 1.0  
  through: 15.0  
  by: 1.0  
)
```

Digital Crown



```
.digitalCrownRotation(  
    $people  
    from: 1.0  
    through: 15.0  
    by: 1.0  
)
```

Digital Crown



```
.digitalCrownRotation(  
  $people  
  from: 1.0  
  through: 15.0  
  by: 1.0  
)
```

Digital Crown



```
.digitalCrownRotation(  
  $people  
  from: 1.0  
  through: 15.0  
  by: 1.0  
)
```

Digital Crown



```
.digitalCrownRotation(  
  $people  
  from: 1.0  
  through: 15.0  
  by: 1.0  
)
```




Cancel

Set



Cancel

Set

Digital Crown



```
.digitalCrownRotation(  
    Binding<BinaryFloatingPoint>  
    from: BinaryFloatingPoint  
    through: BinaryFloatingPoint  
    by: BinaryFloatingPoint  
    sensitivity: RotationSensitivity  
    isContinuous: Bool  
)
```

Digital Crown



```
.digitalCrownRotation(  
    Binding<BinaryFloatingPoint>  
    from: BinaryFloatingPoint  
    through: BinaryFloatingPoint  
    by: BinaryFloatingPoint  
    sensitivity: RotationSensitivity  
    isContinuous: Bool  
)
```

Digital Crown



```
.digitalCrownRotation(  
  $hours  
  from: 1.0  
  through: 12.0  
  by: 1.0  
  sensitivity: RotationSensitivity  
  isContinuous: Bool  
)
```

Digital Crown



```
.digitalCrownRotation(  
  $hours  
  from: 1.0  
  through: 12.0  
  by: 1.0  
  sensitivity: RotationSensitivity  
  isContinuous: Bool  
)
```

Digital Crown



```
.digitalCrownRotation(  
  $hours  
  from: 1.0  
  through: 12.0  
  by: 1.0  
  sensitivity: .low  
  isContinuous: Bool  
)
```

Digital Crown



```
.digitalCrownRotation(  
  $hours  
  from: 1.0  
  through: 12.0  
  by: 1.0  
  sensitivity: .low  
  isContinuous: Bool  
)
```


Digital Crown



```
.digitalCrownRotation(  
  $hours  
  from: 1.0  
  through: 12.0  
  by: 1.0  
  sensitivity: .low  
  isContinuous: true  
)
```

Digital Crown



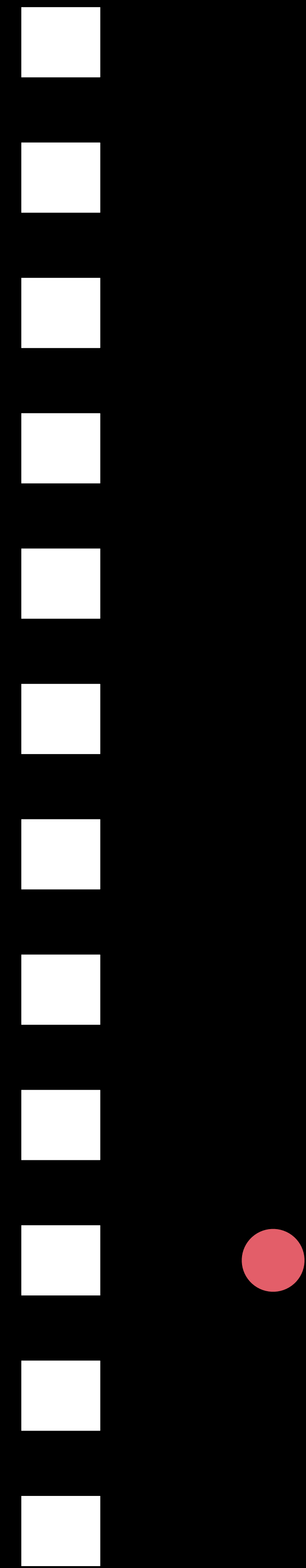
```
.digitalCrownRotation(  
  $hours  
  from: 1.0  
  through: 12.0  
  by: 1.0  
  sensitivity: .low  
  isContinuous: true  
)
```

Digital Crown



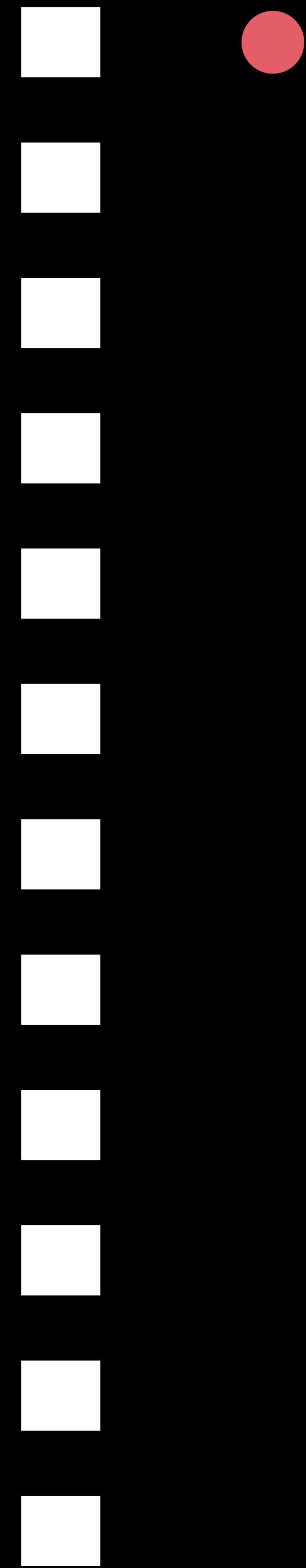
```
.digitalCrownRotation(  
  $hours  
  from: 1.0  
  through: 12.0  
  by: 1.0  
  sensitivity: .low  
  isContinuous: true  
)
```

Digital Crown



```
.digitalCrownRotation(  
  $hours  
  from: 1.0  
  through: 12.0  
  by: 1.0  
  sensitivity: .low  
  isContinuous: true  
)
```

Digital Crown



```
.digitalCrownRotation(  
  $hours  
  from: 1.0  
  through: 12.0  
  by: 1.0  
  sensitivity: .low  
  isContinuous: true  
)
```

Focus



```
VStack {  
    Tip(selection: $percent)  
  
    People(selection: $people)  
  
}
```

Focus



```
VStack {  
    Tip(selection: $percent)  
  
    People(selection: $people)  
  
}
```

Focus



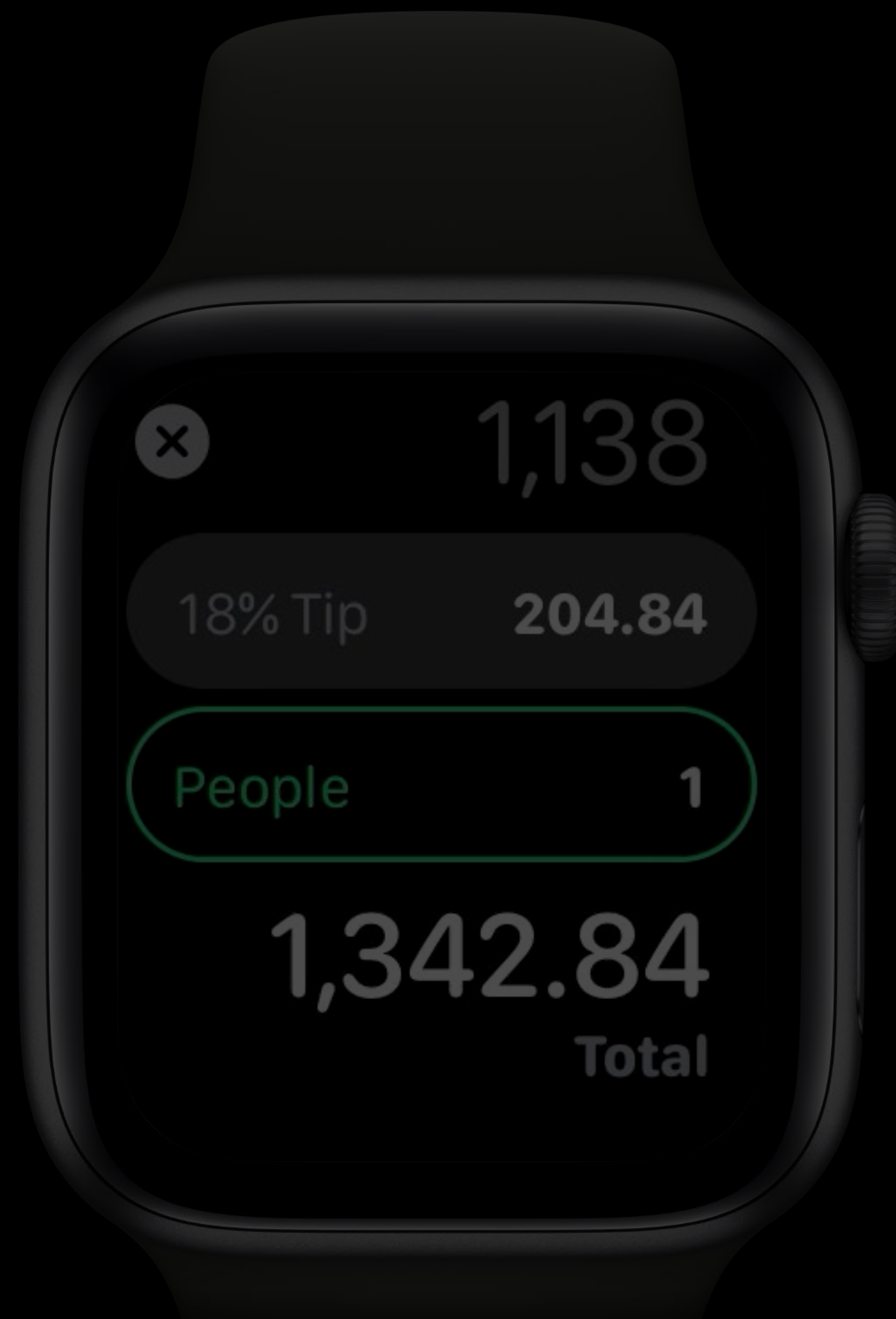
```
VStack {  
    Tip(selection: $percent)  
        .focusable(true)  
    People(selection: $people)  
        .focusable(true)  
}
```


Focus



```
VStack {  
  Tip(selection: $percent)  
    .focusable(true)  
  People(selection: $people)  
    .focusable(true)  
}
```

Focus



```
VStack {  
    Tip(selection: $percent)  
        .focusable(true)  
    People(selection: $people)  
        .focusable(true)  
}
```

Demo

SwiftUI and watchOS

Engaging and interactive experiences

Easily create custom UI

Concepts apply everywhere

More Information

developer.apple.com/wwdc19/219

Data Flow Through SwiftUI

Thursday, 9:00

Building Custom Views with SwiftUI

Friday, 9:00

Creating Independent Watch Apps

WWDC 2019

