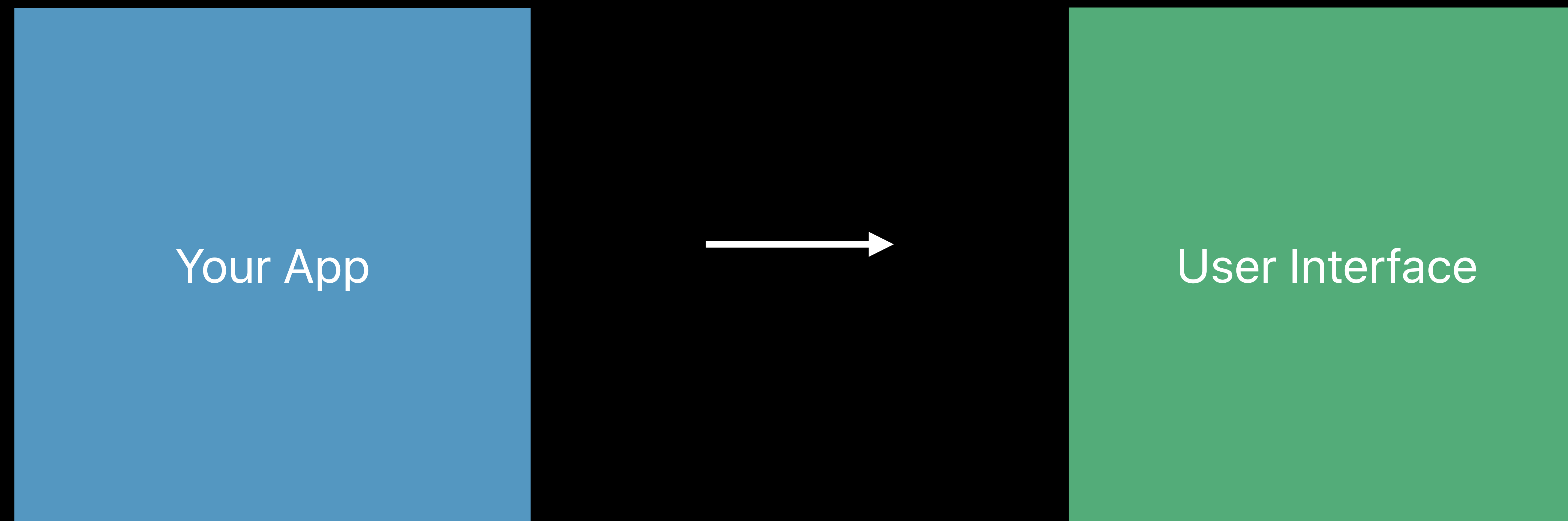


#WWDC19

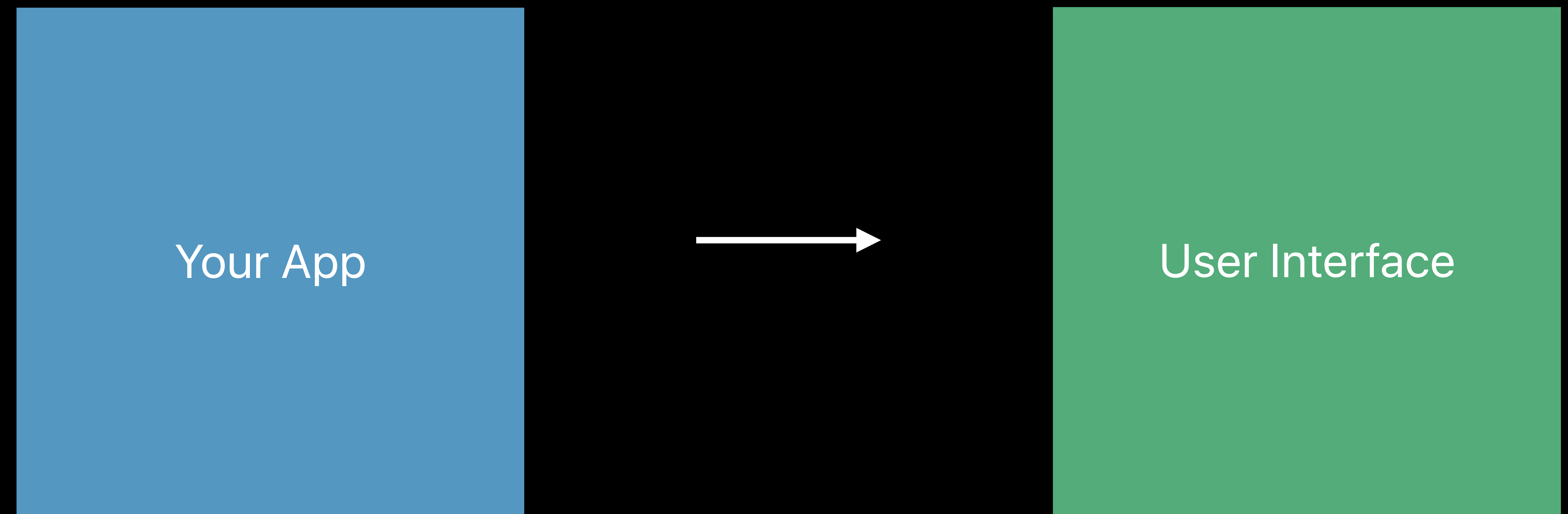
Targeting Content with Multiple Windows

Alex Schaefer, iOS System UI

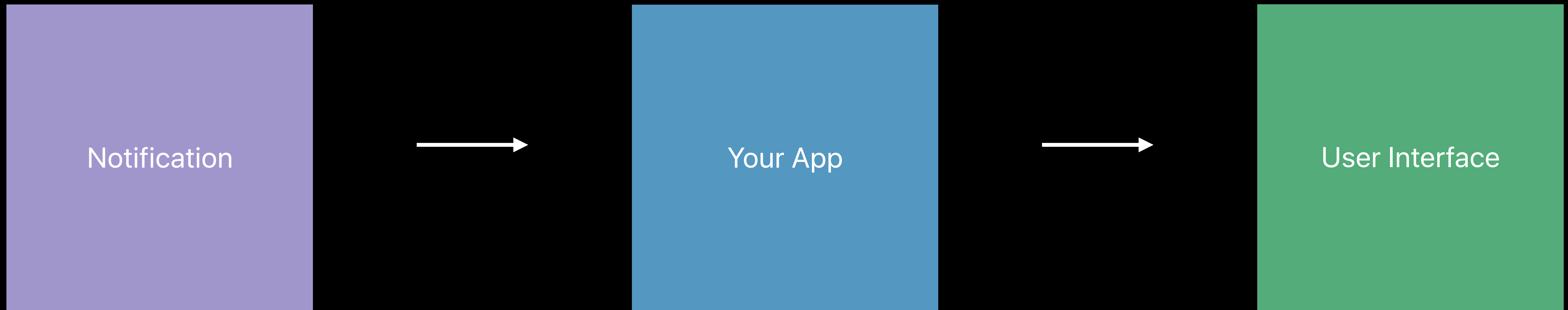
iOS 12



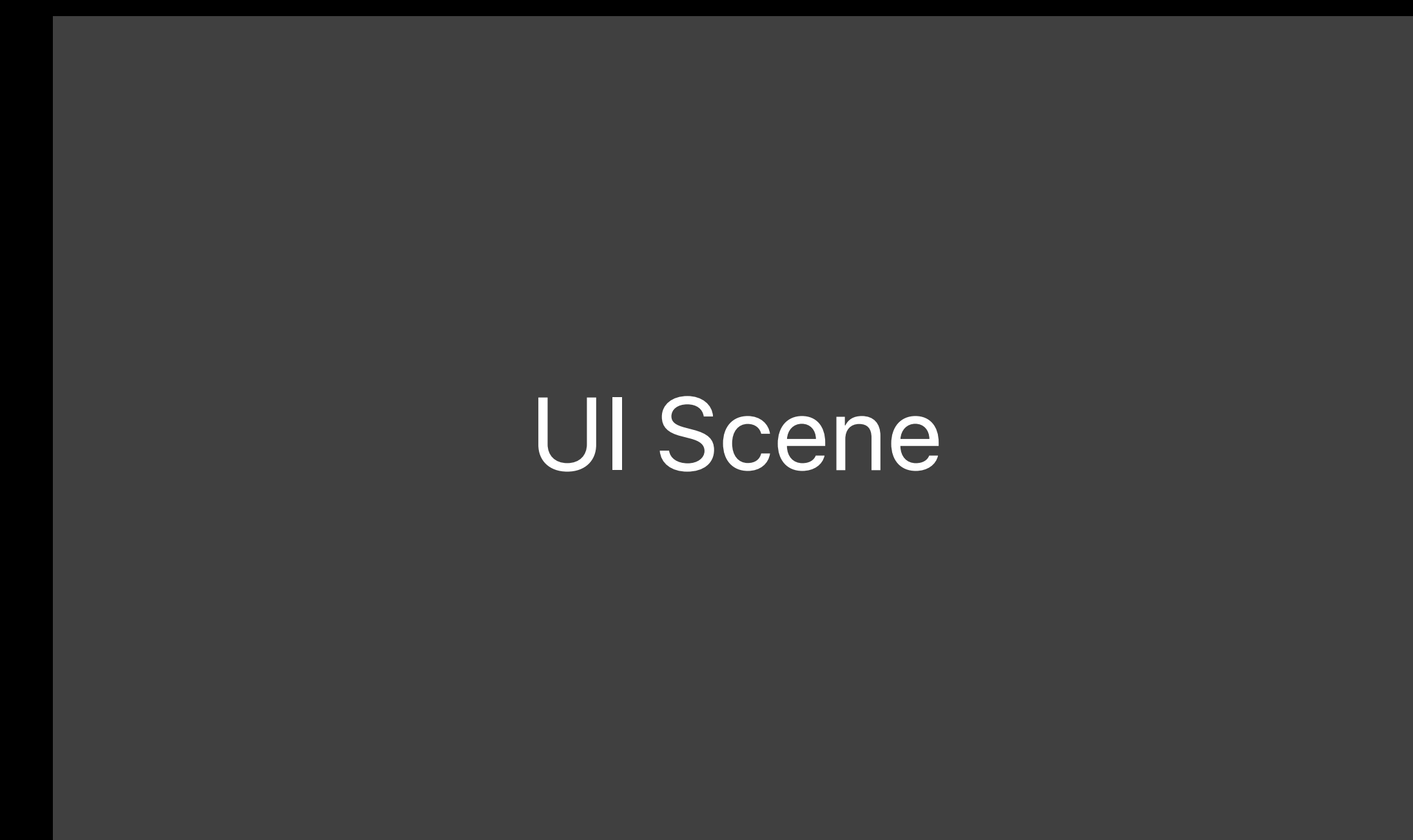
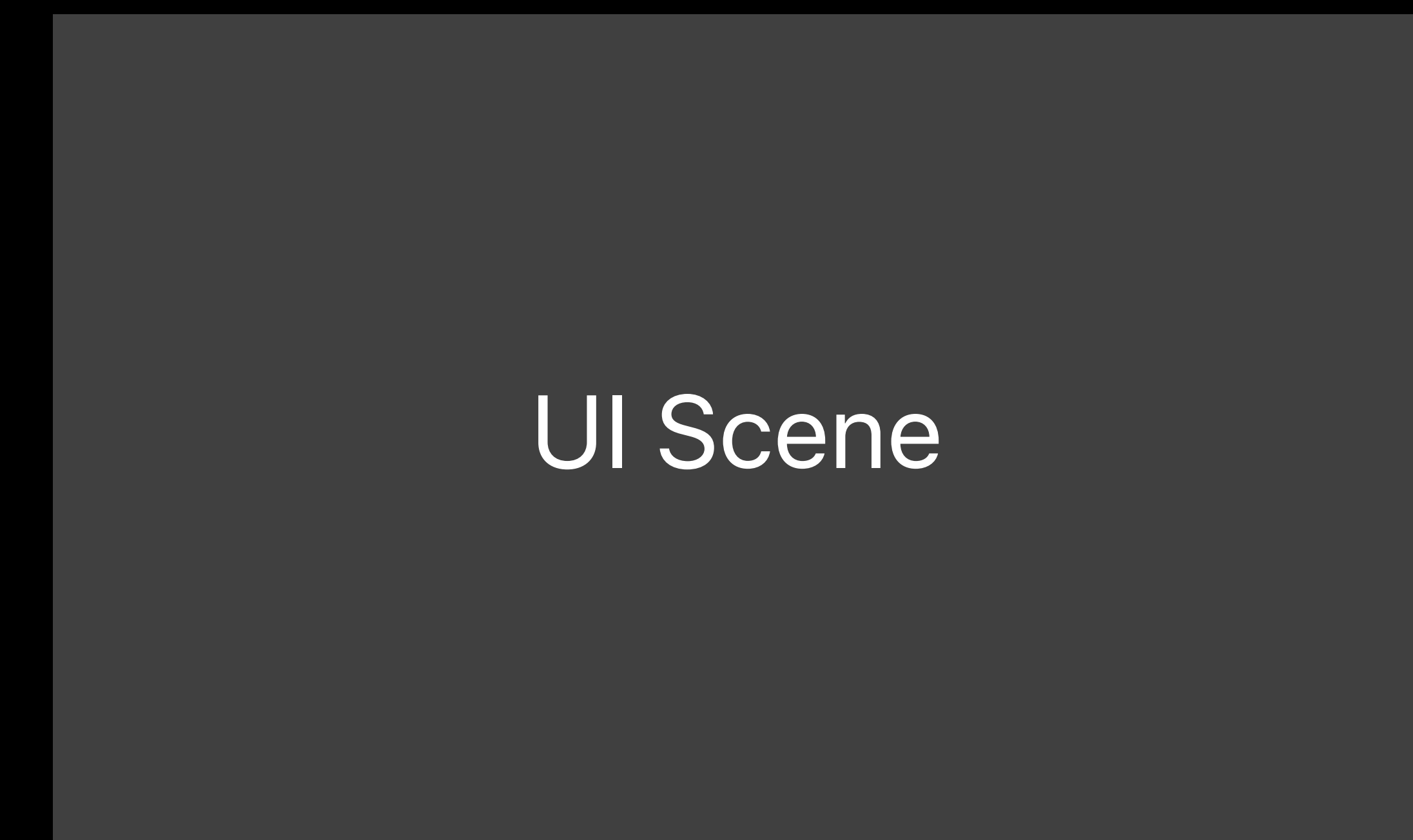
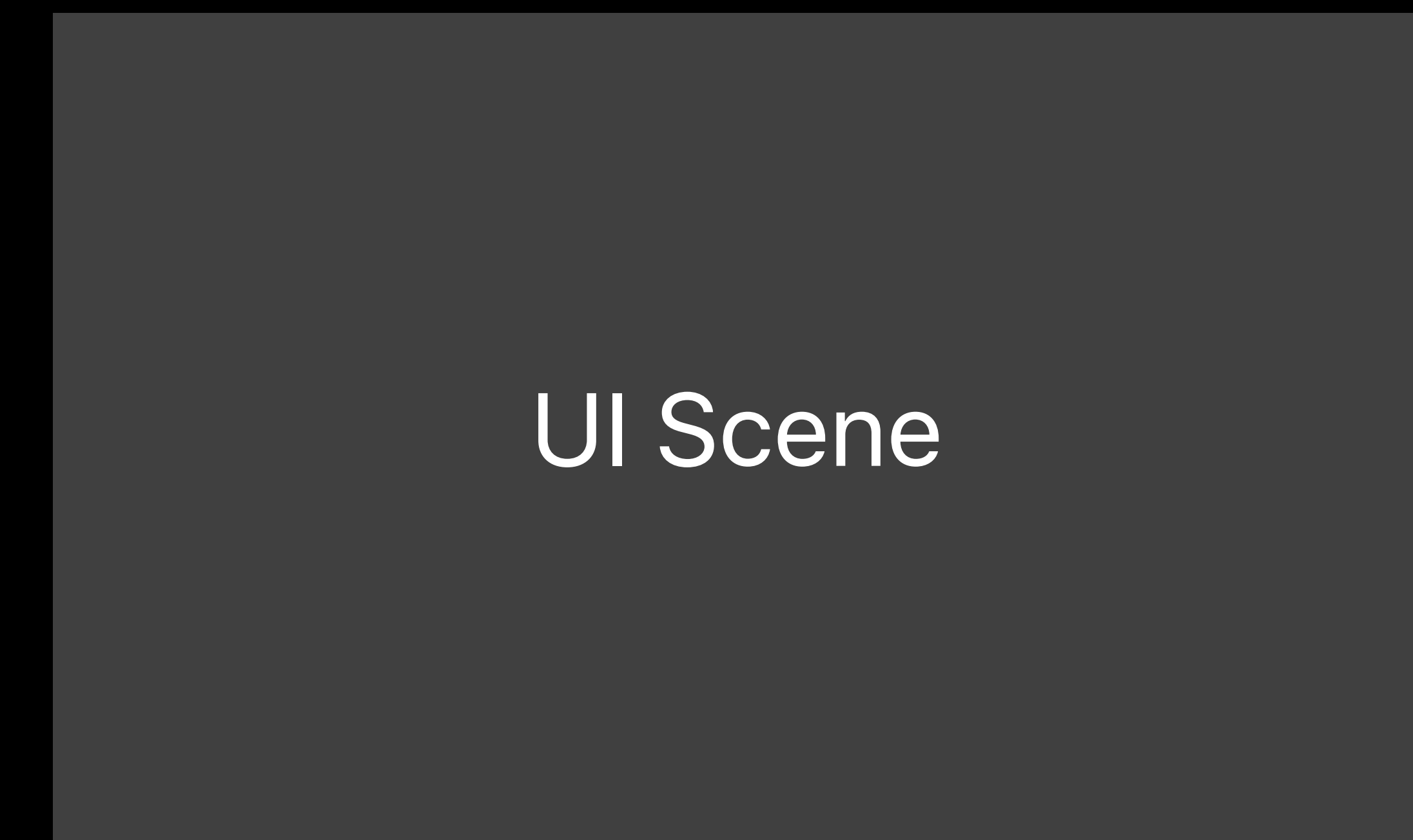
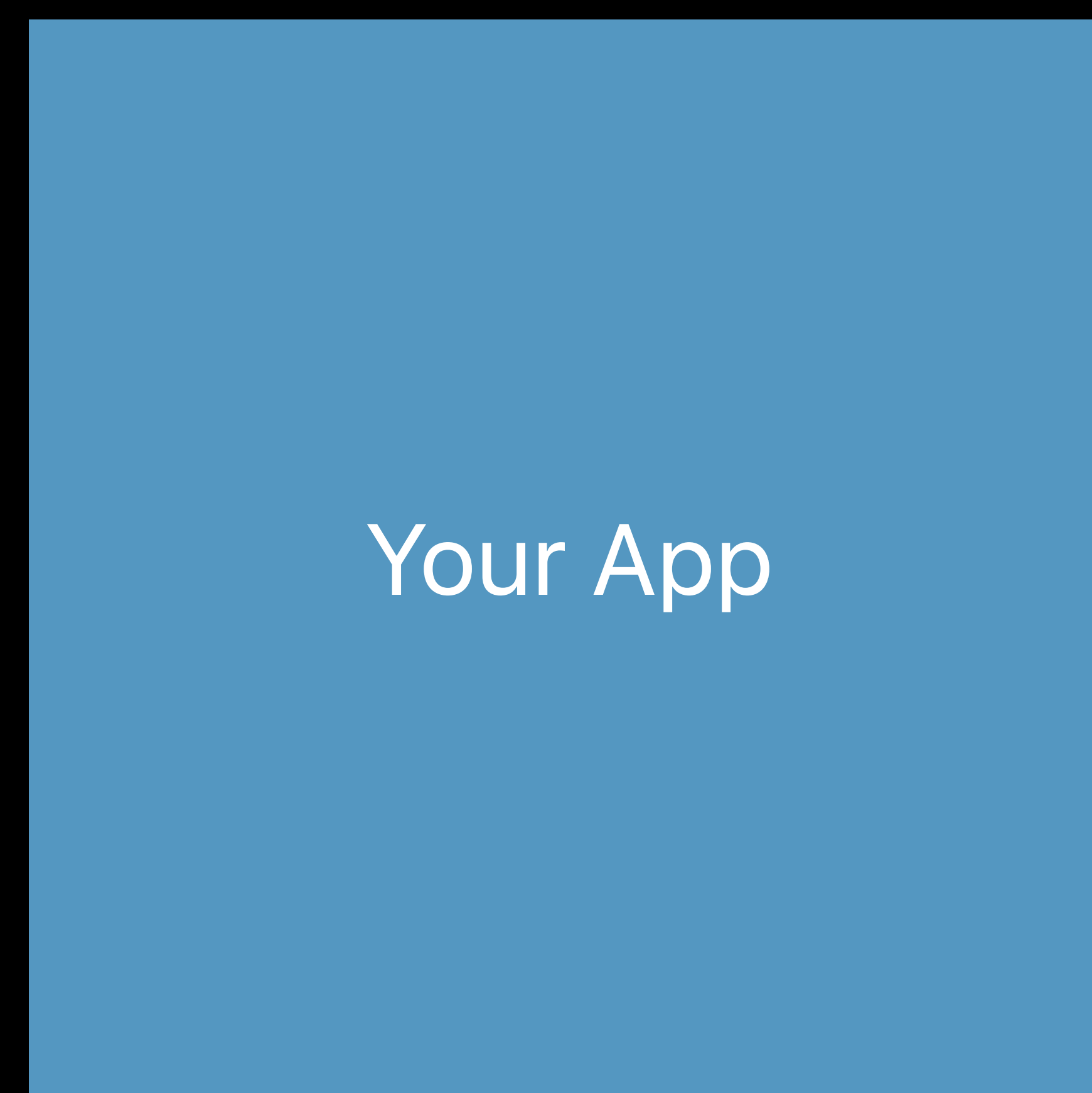
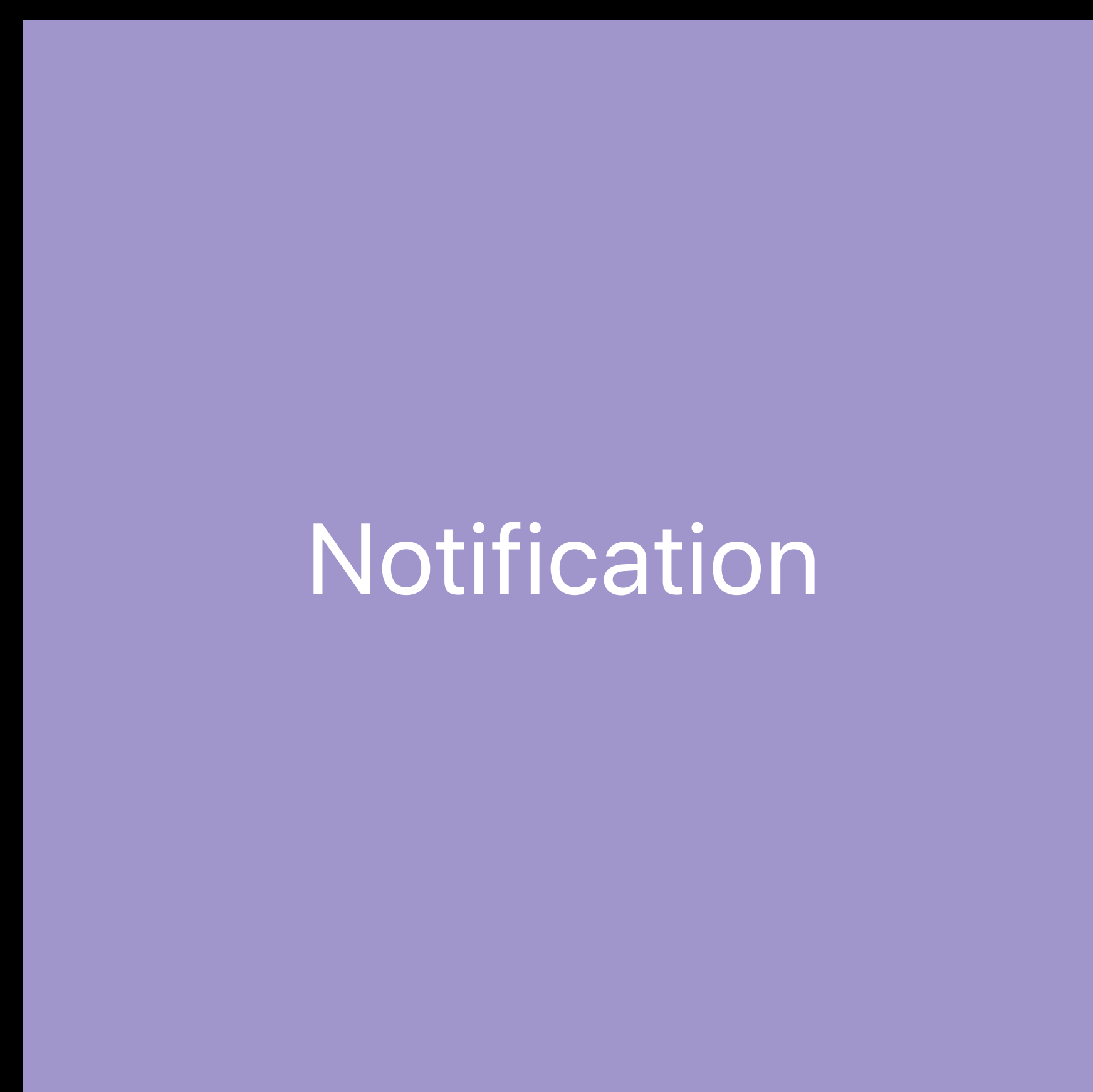
iOS 12



iOS 12



iOS 13



Which Scene should activate?

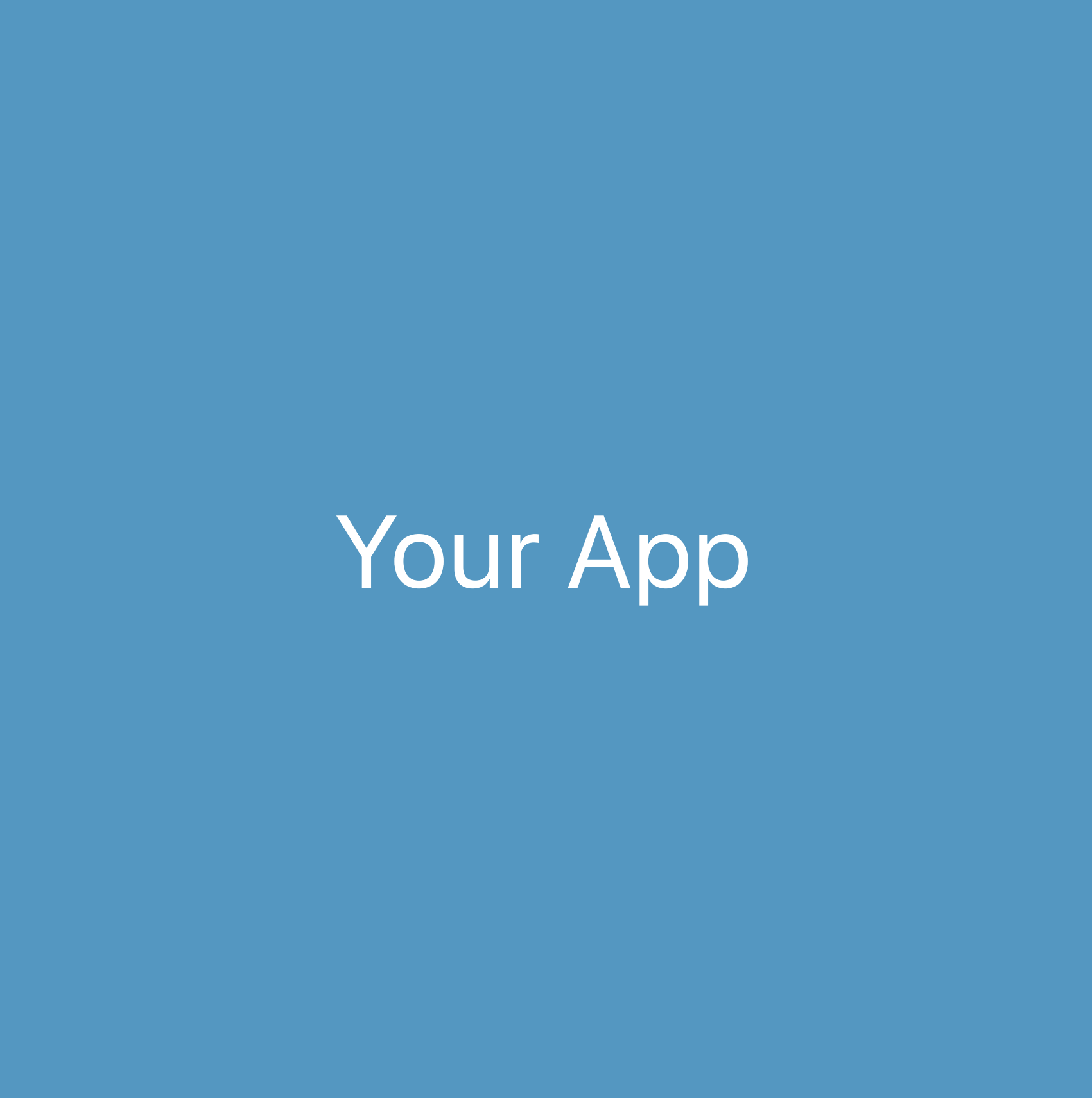
Just ask my app!

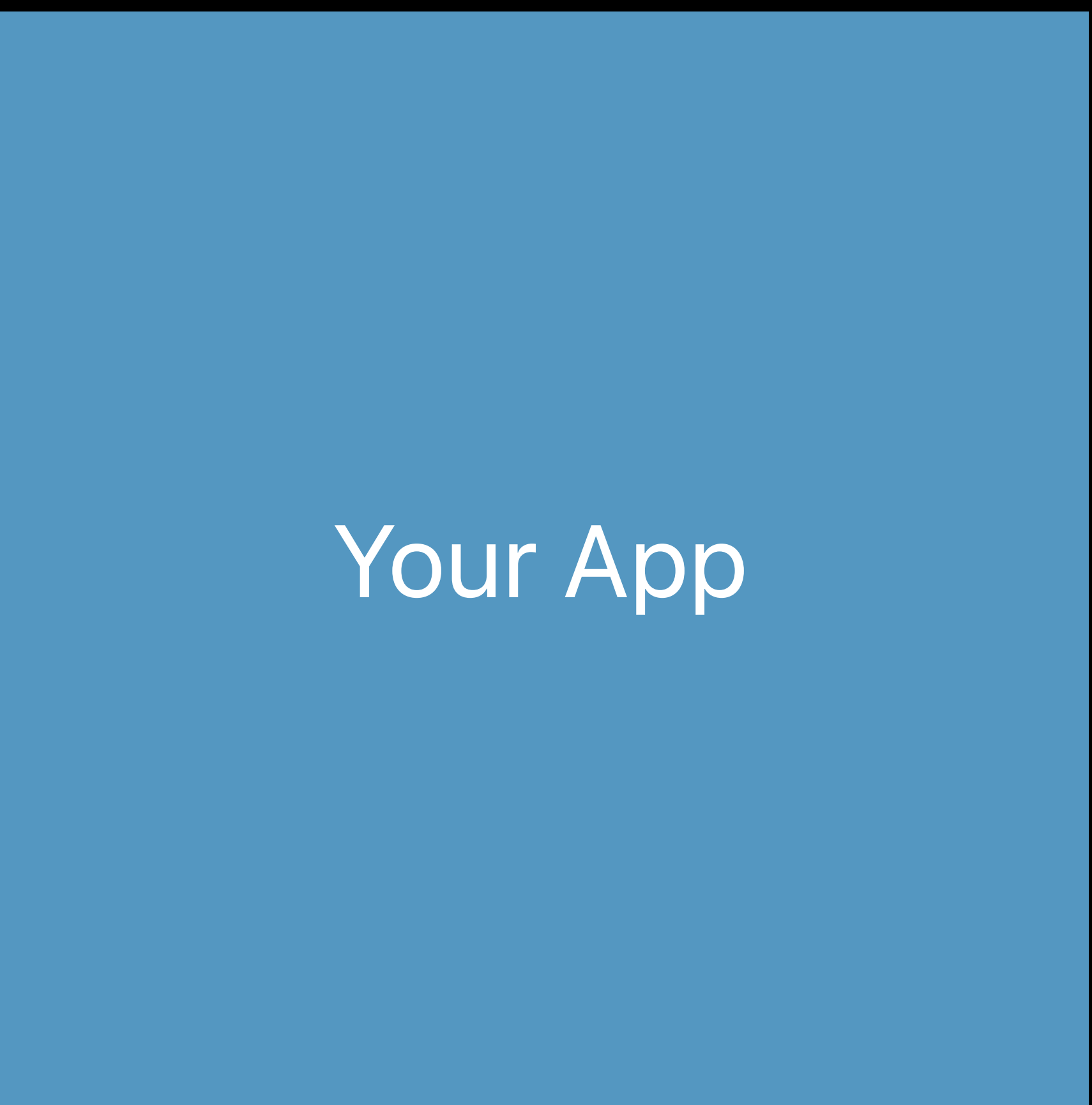
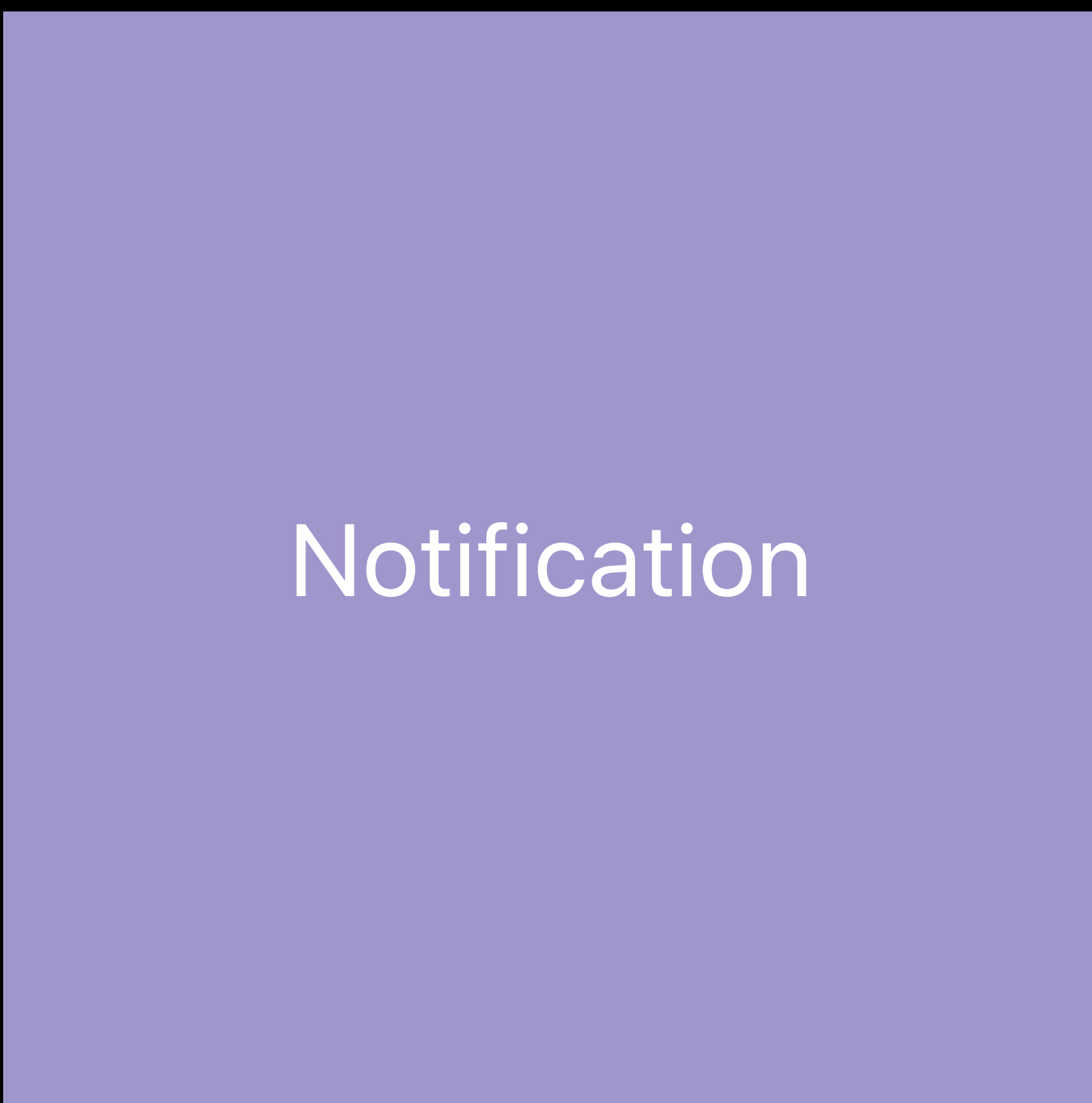
Who cares?
Just launch my app!

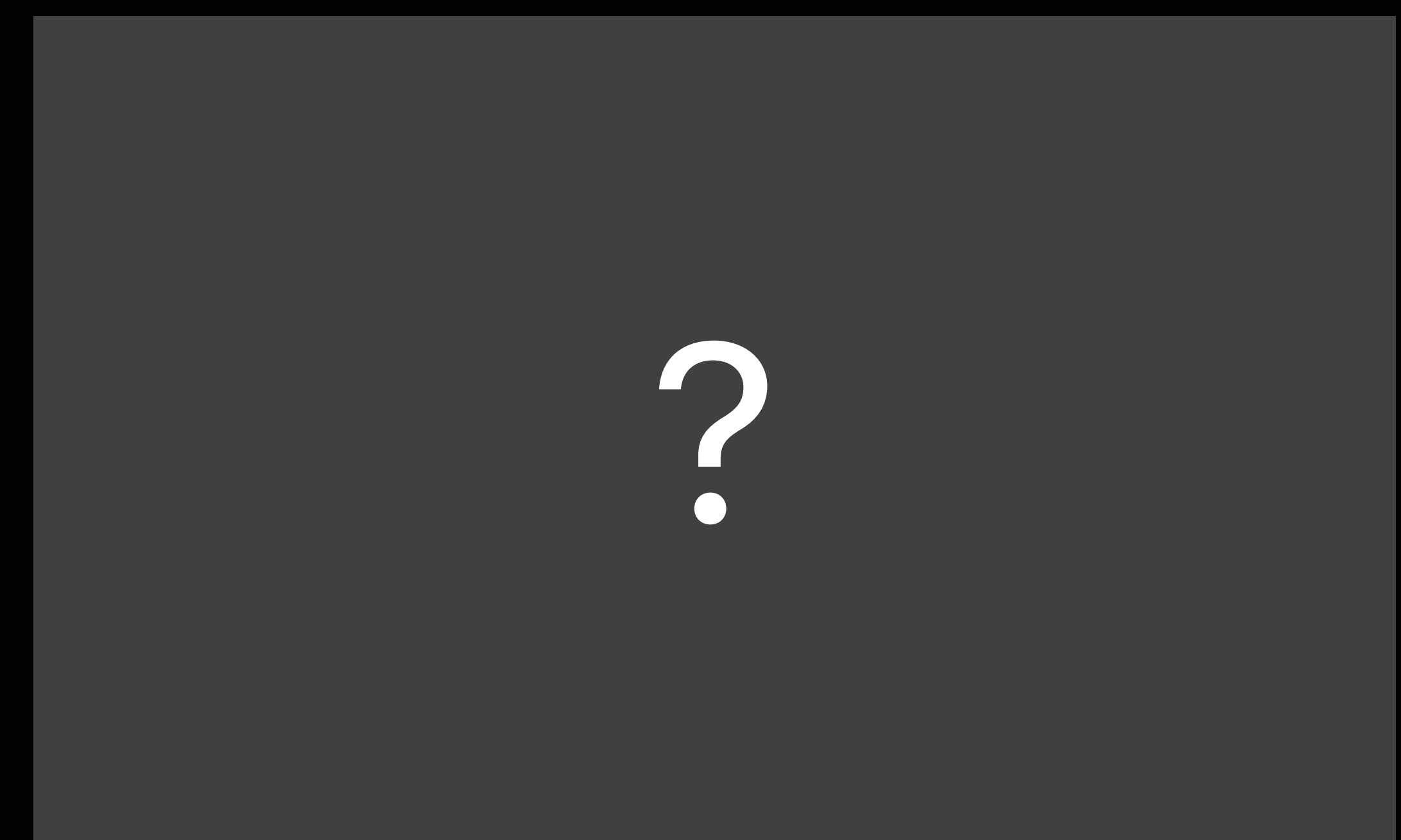
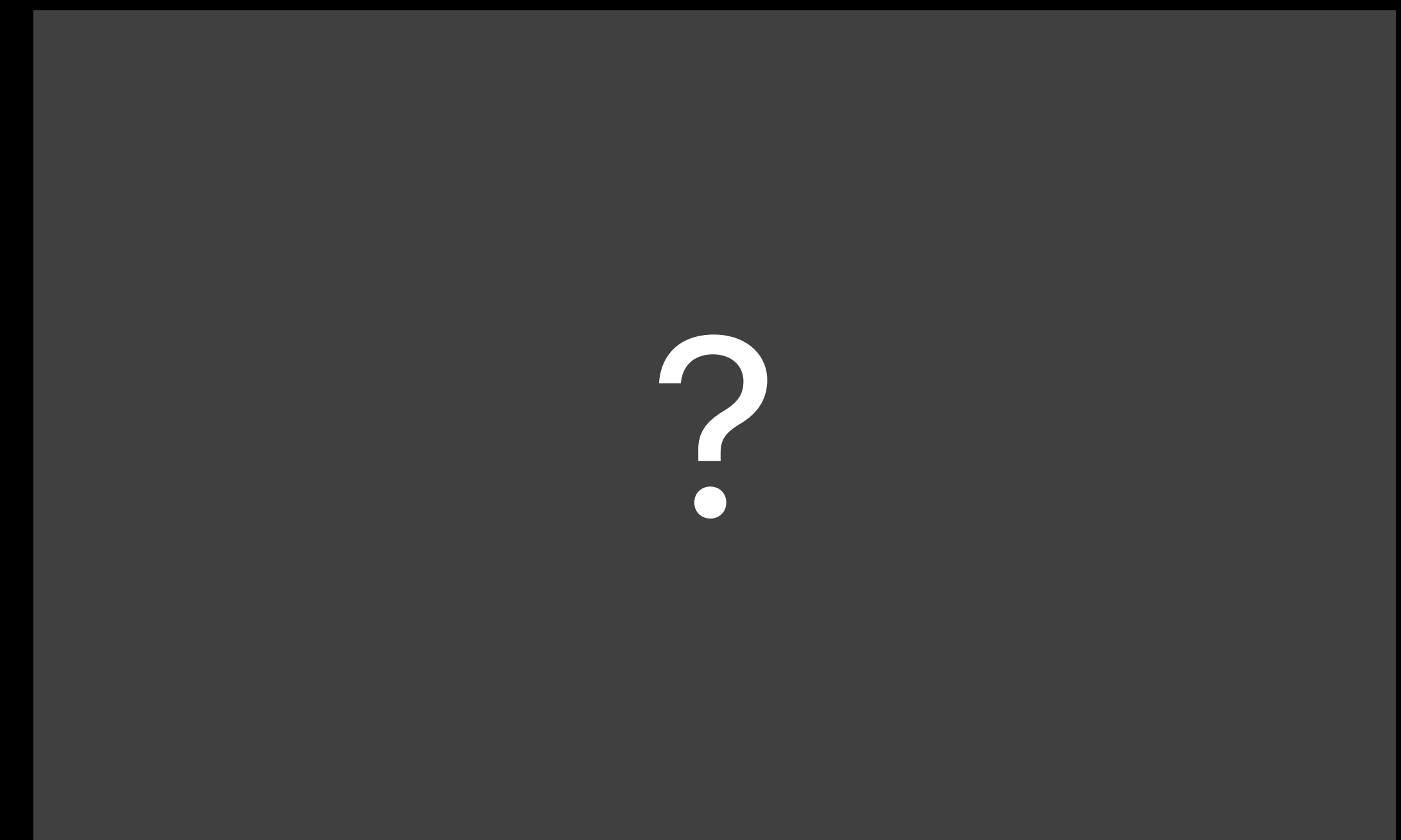
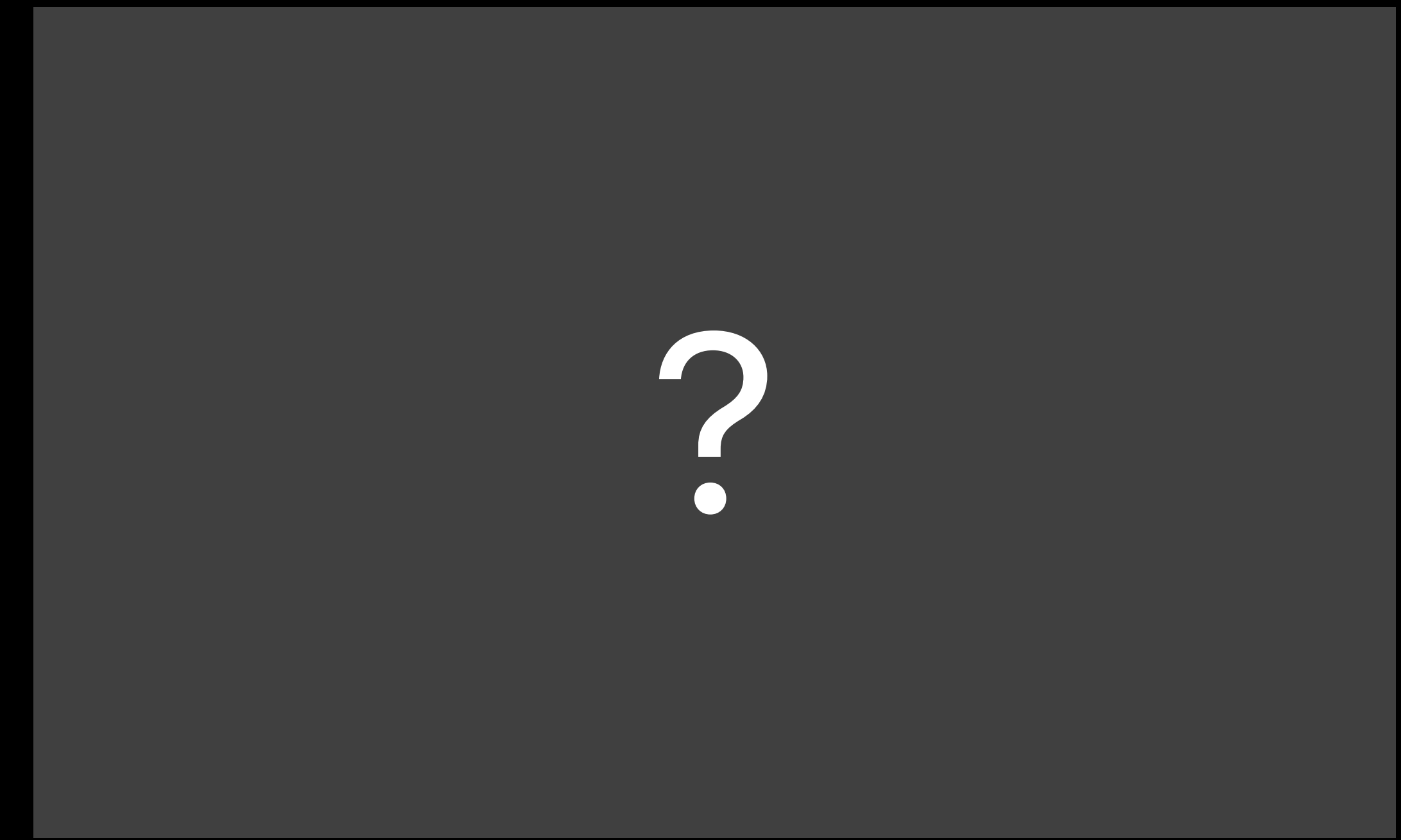
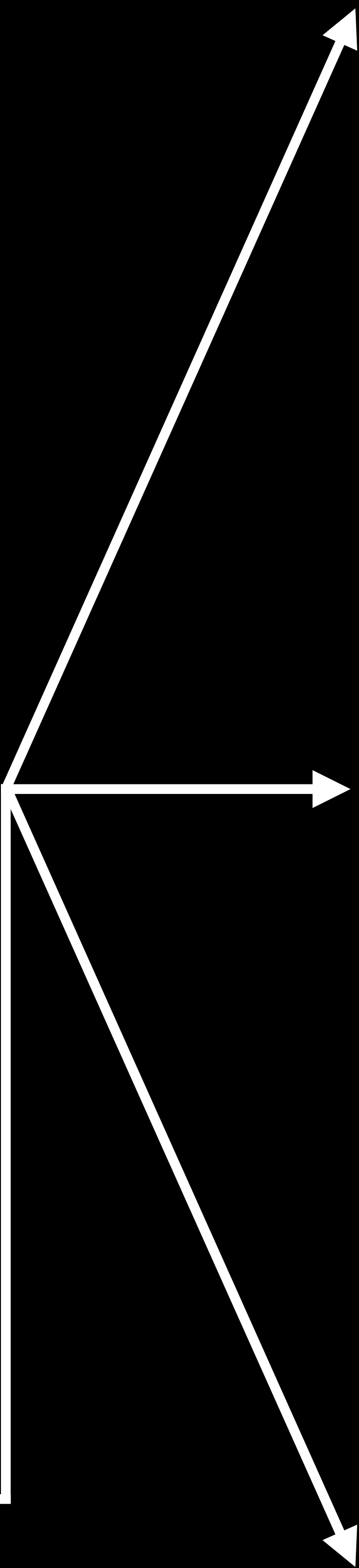
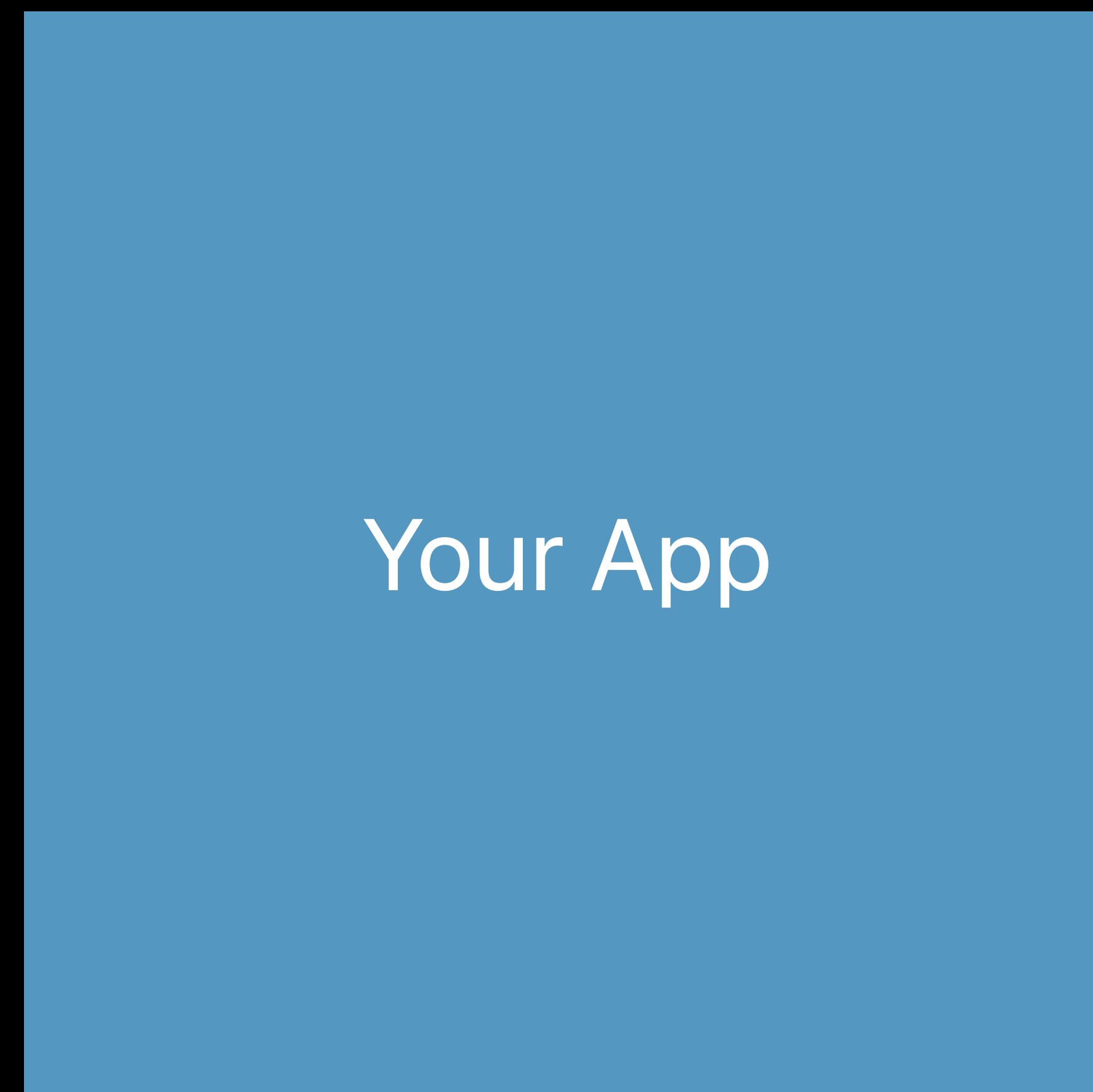
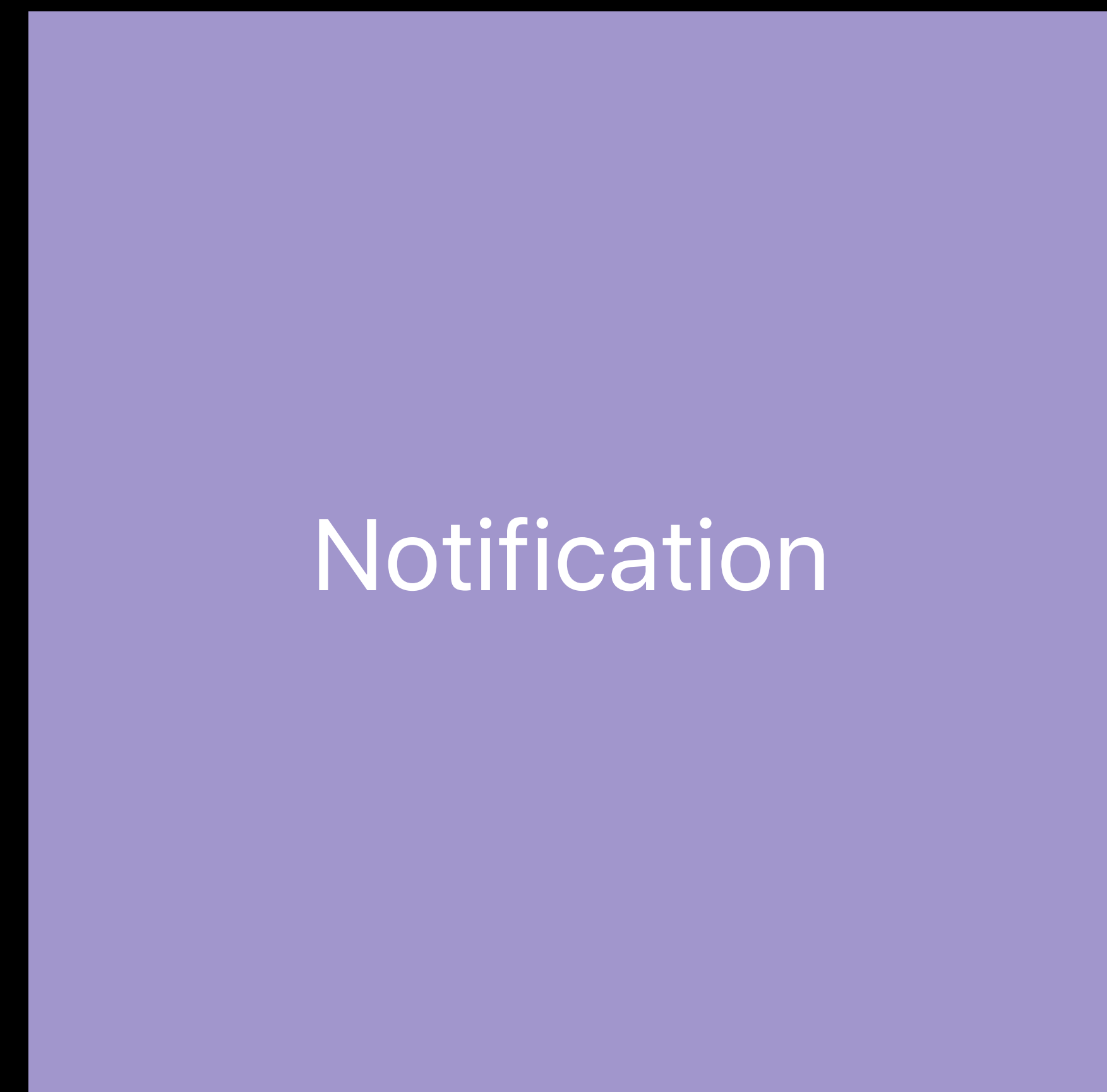
Hold your horses.

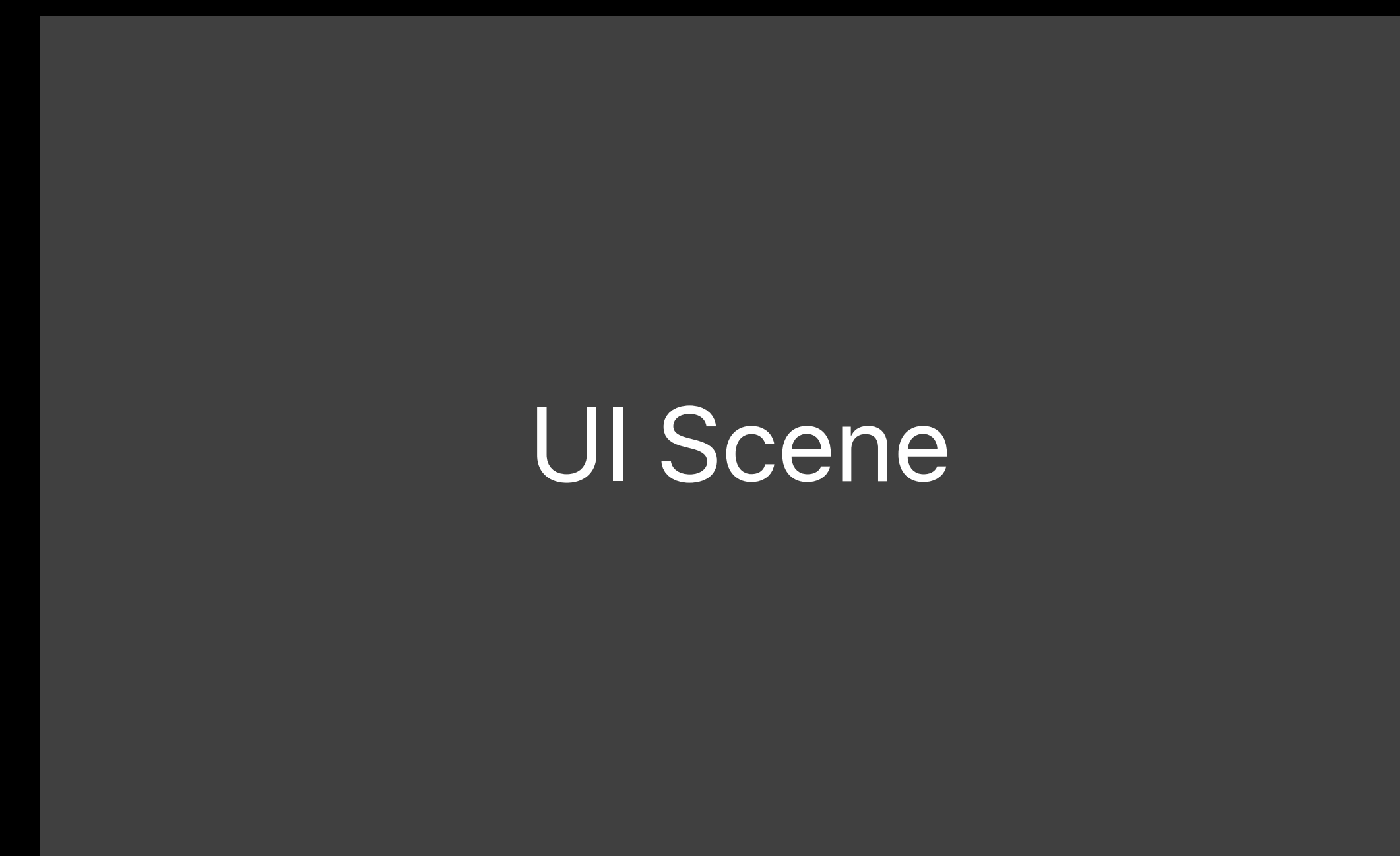
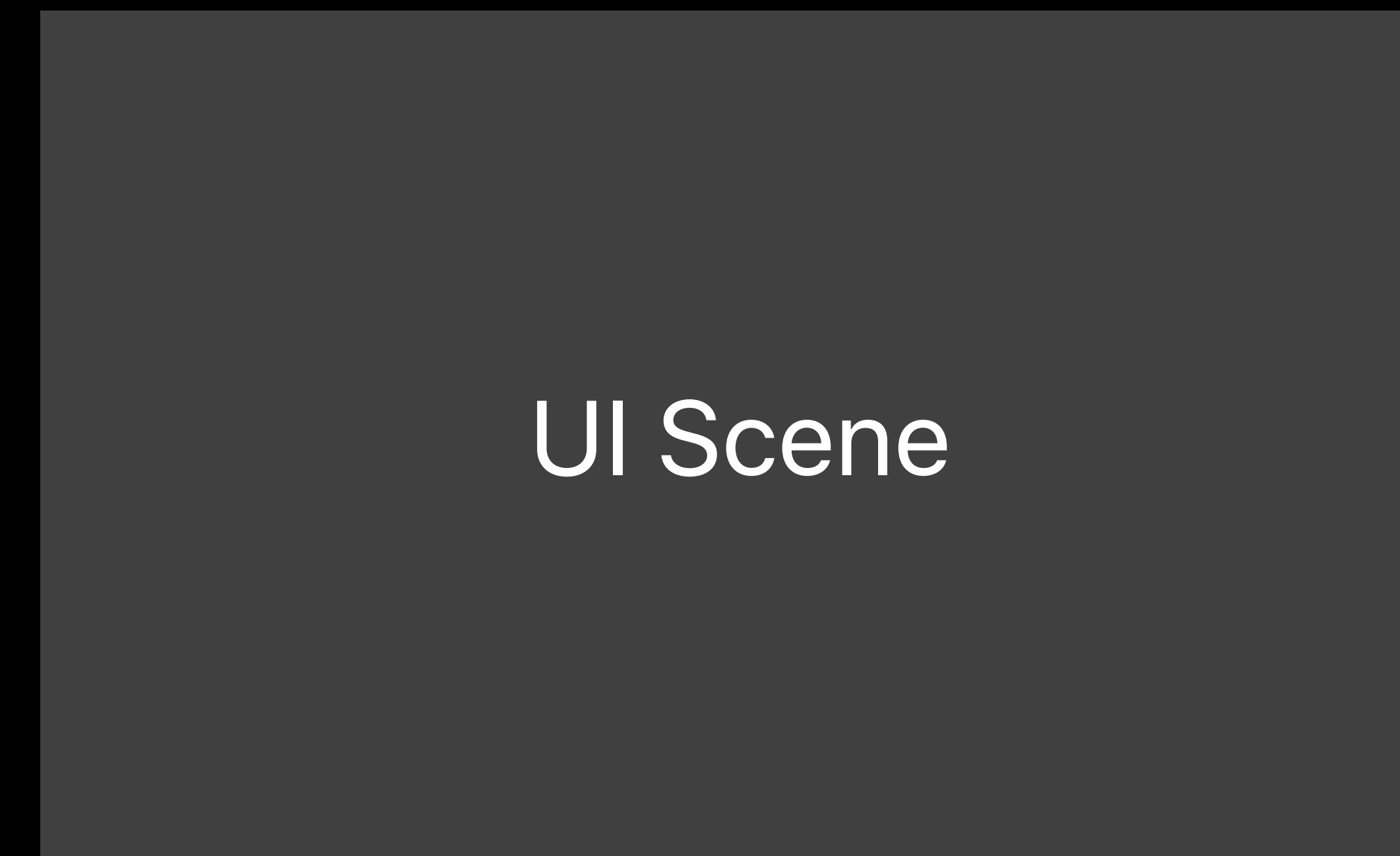
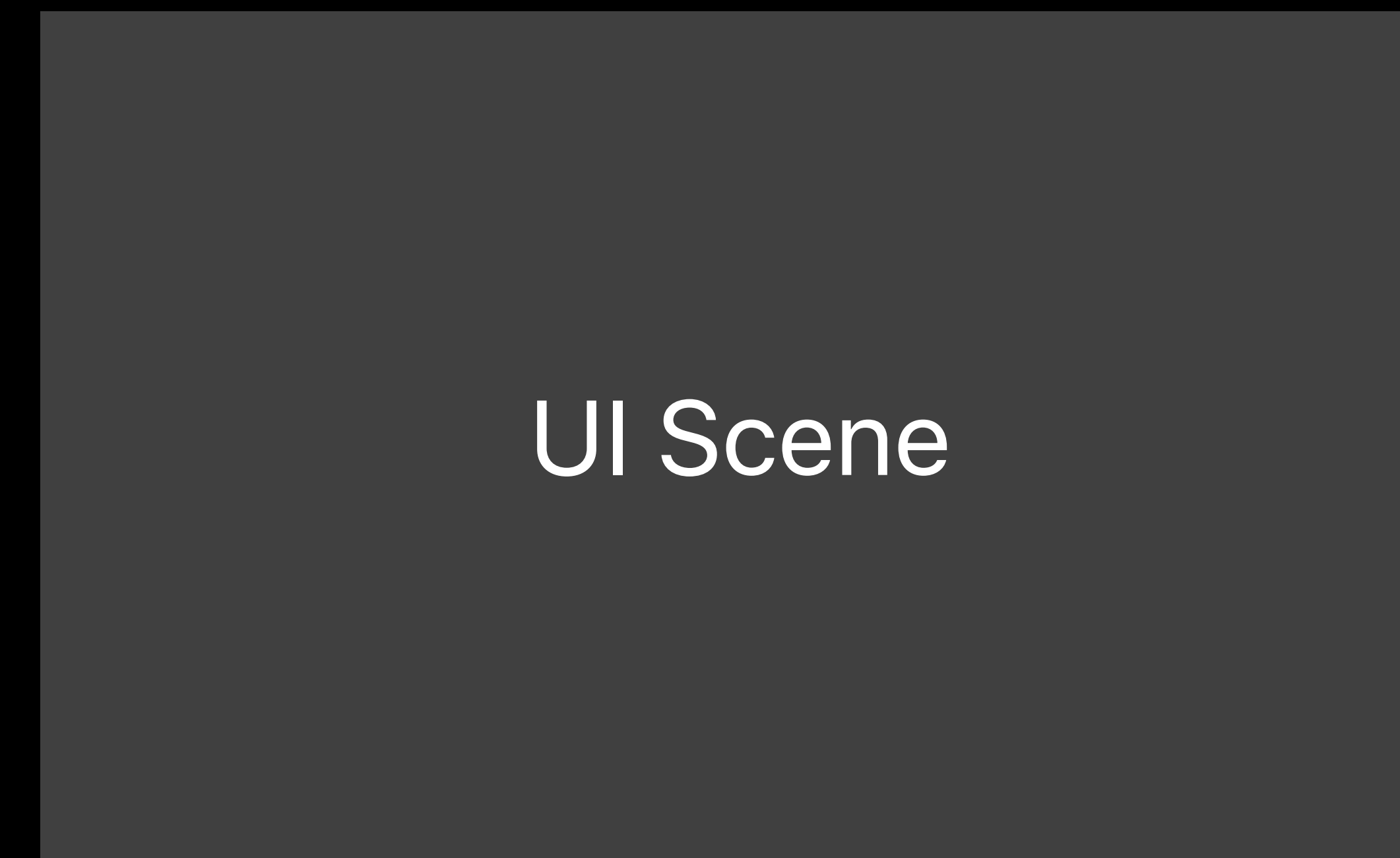
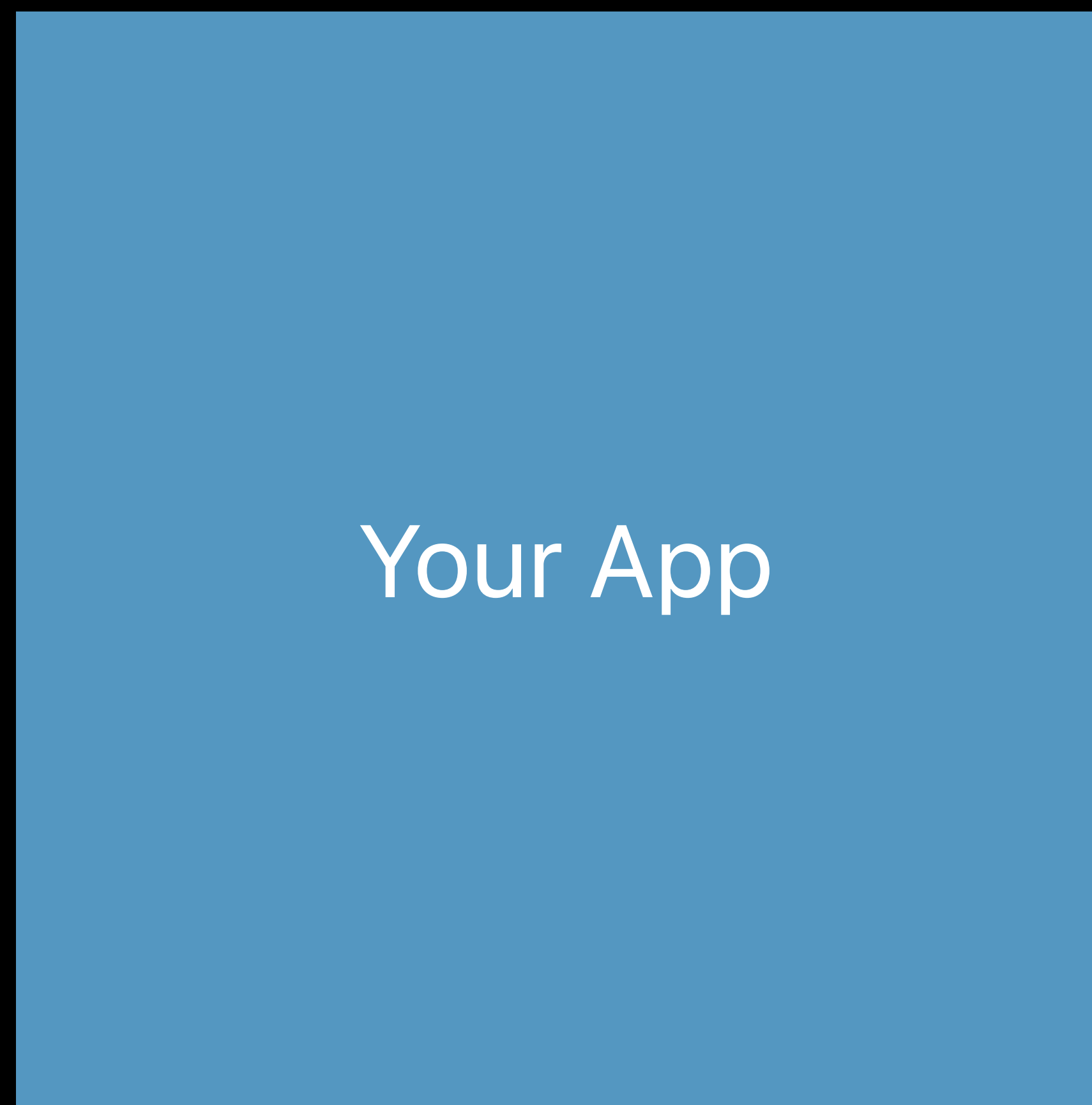
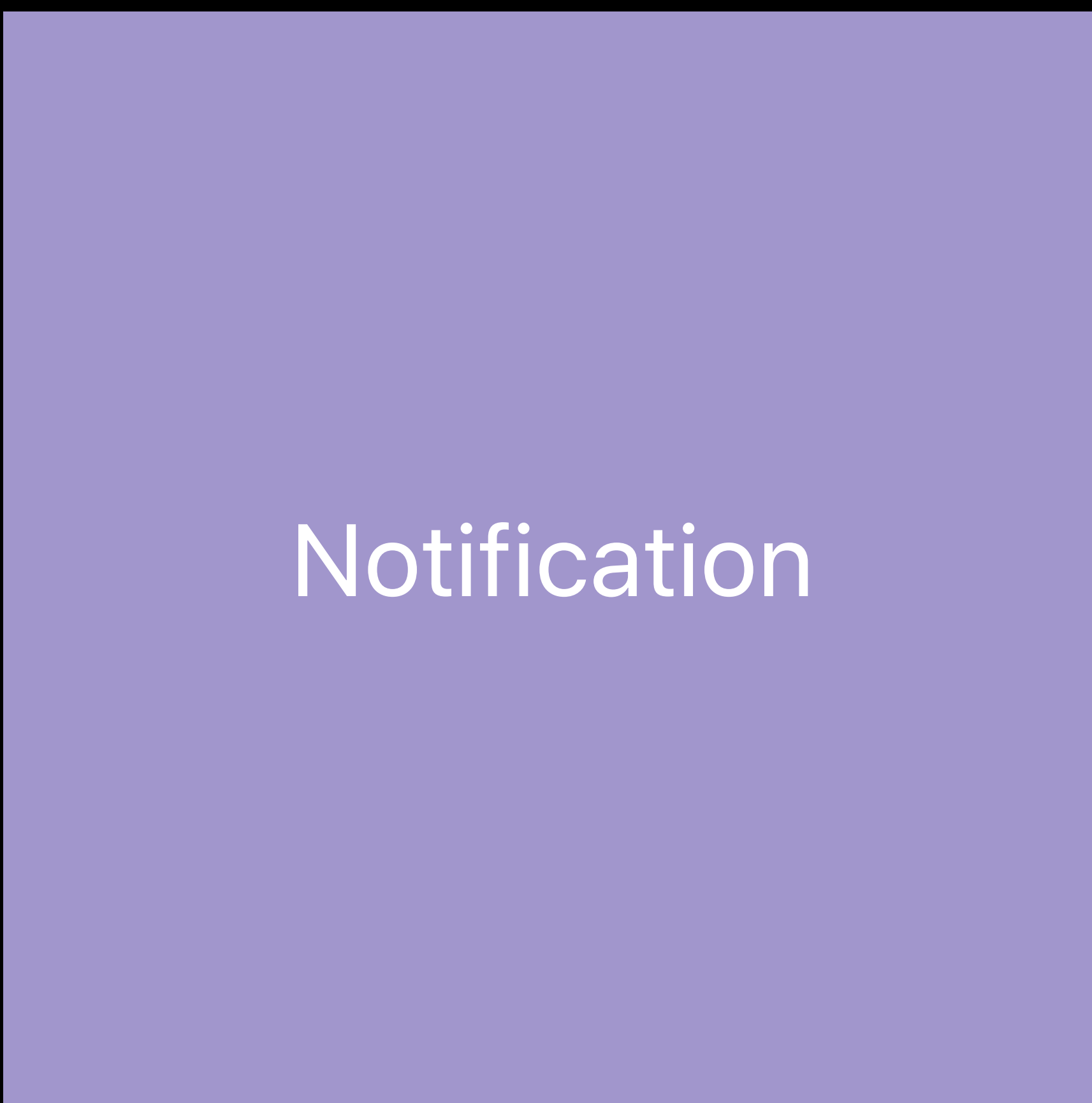
What do we have to work with?

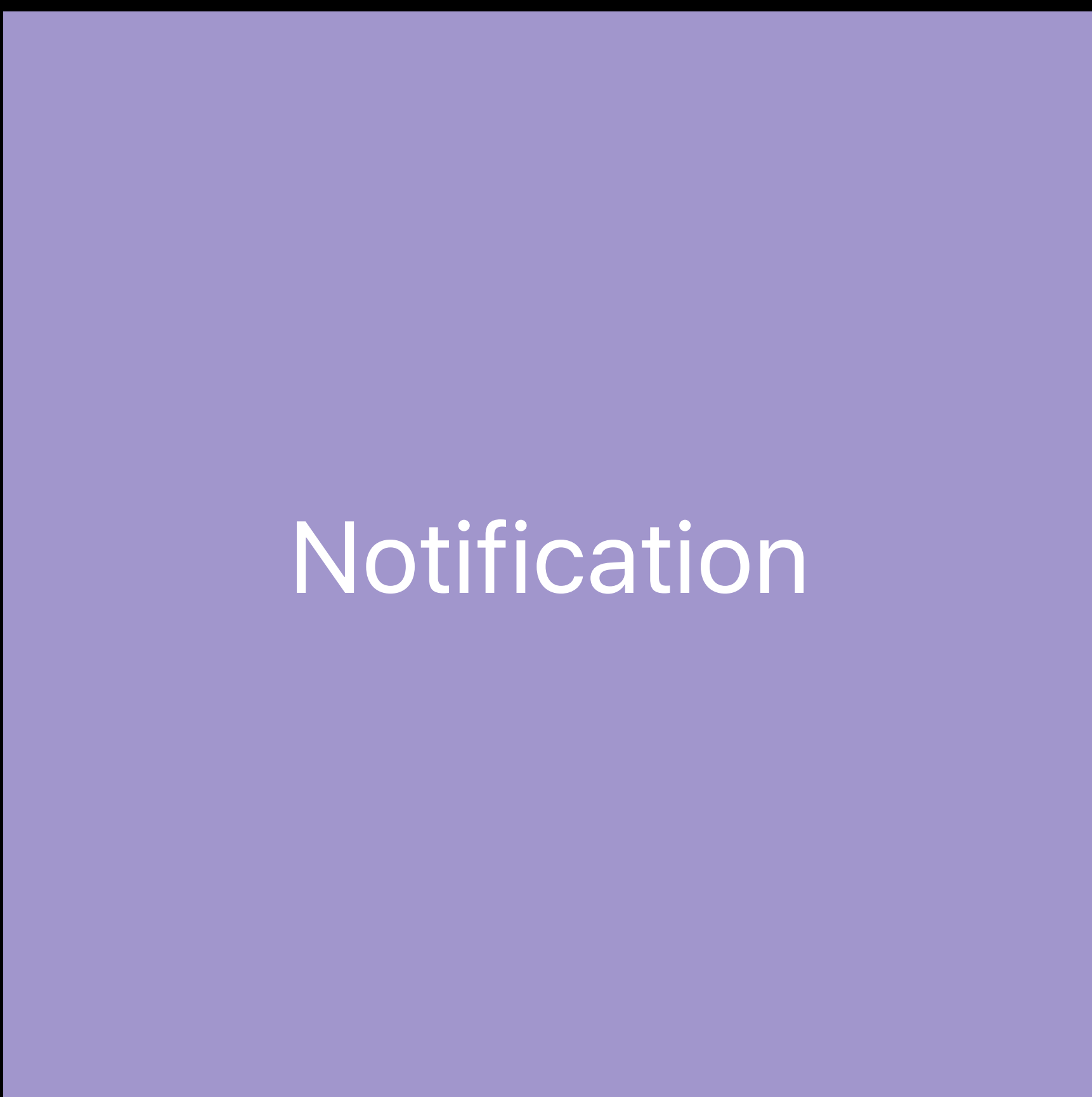
Your App





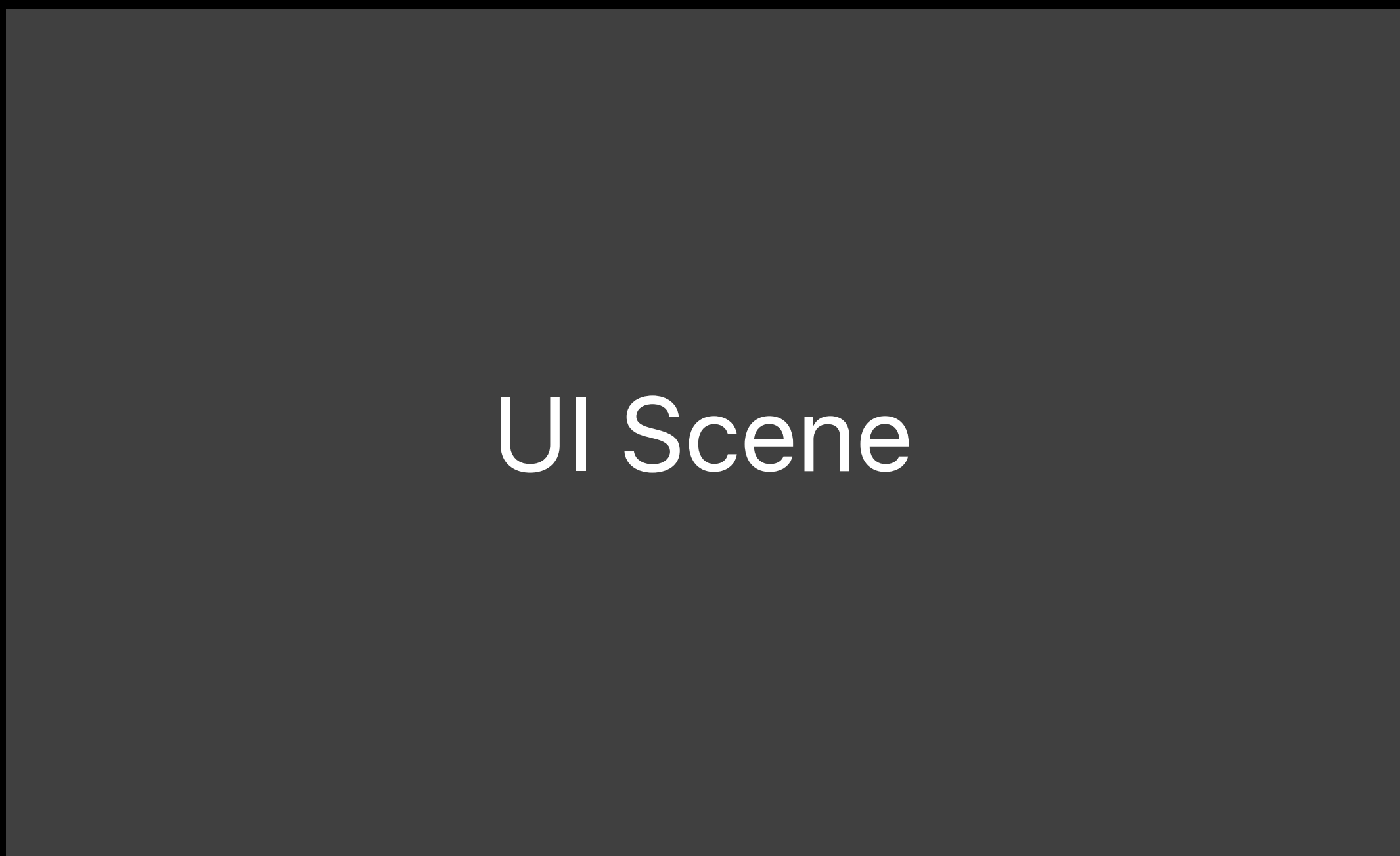
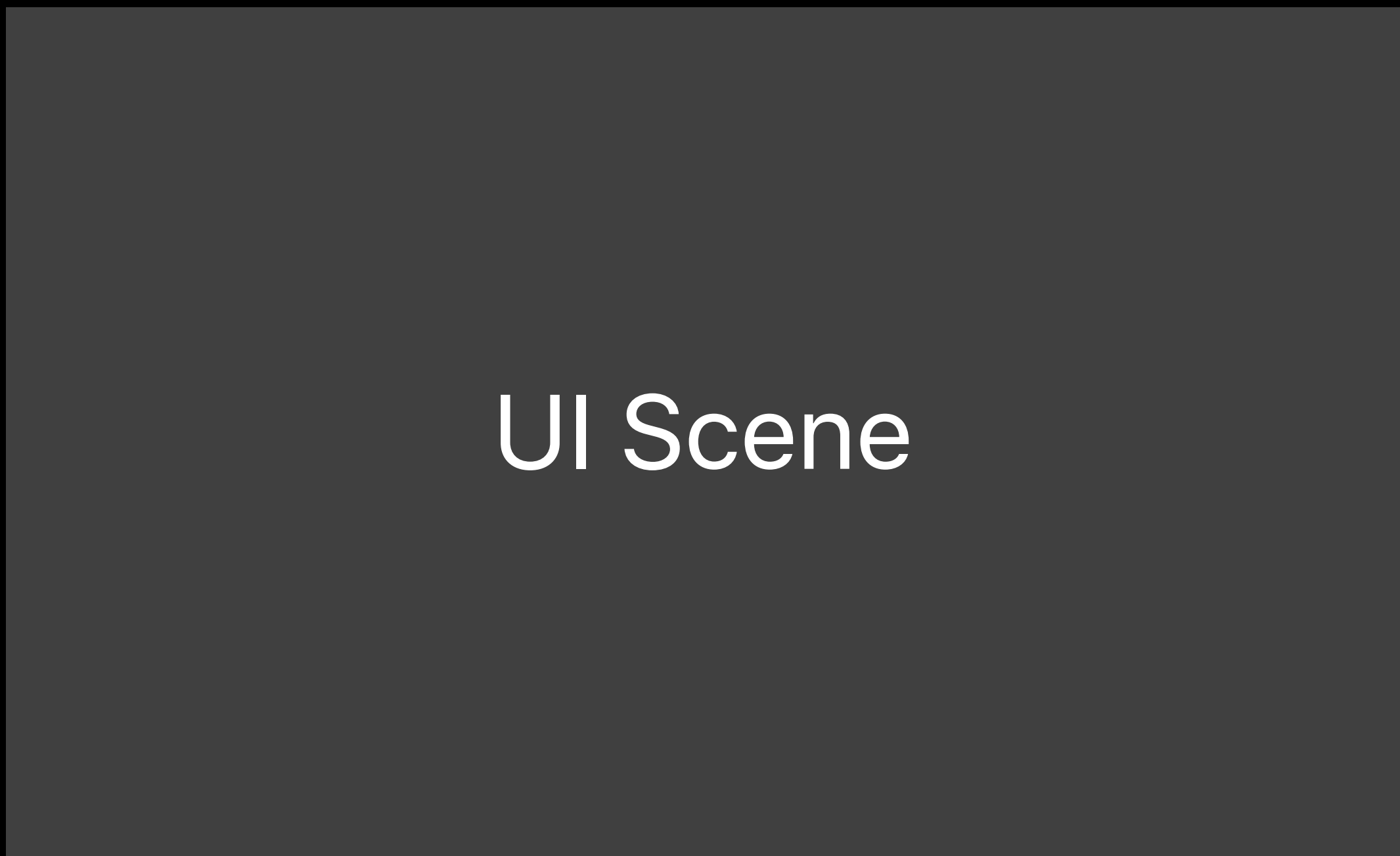
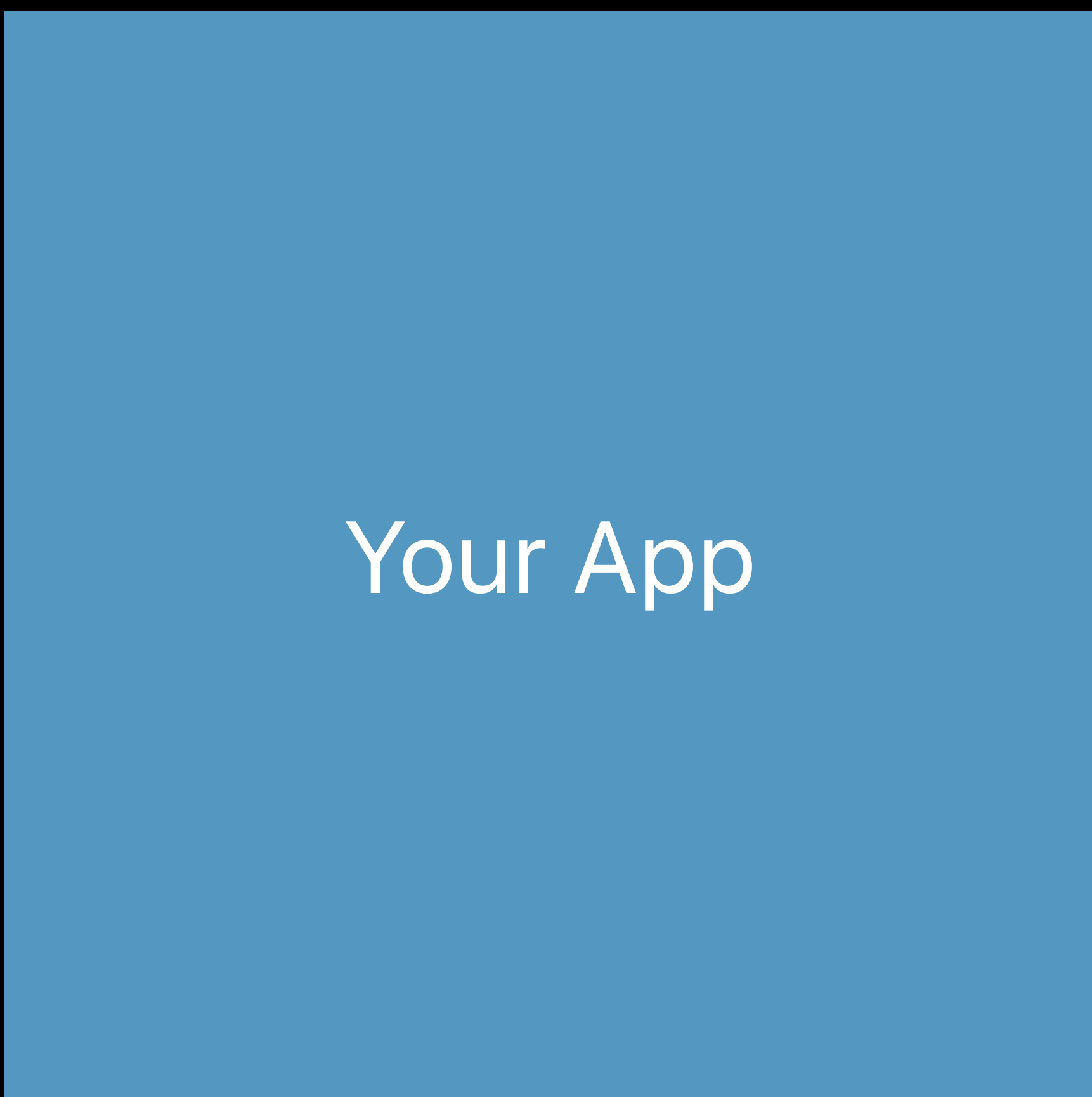


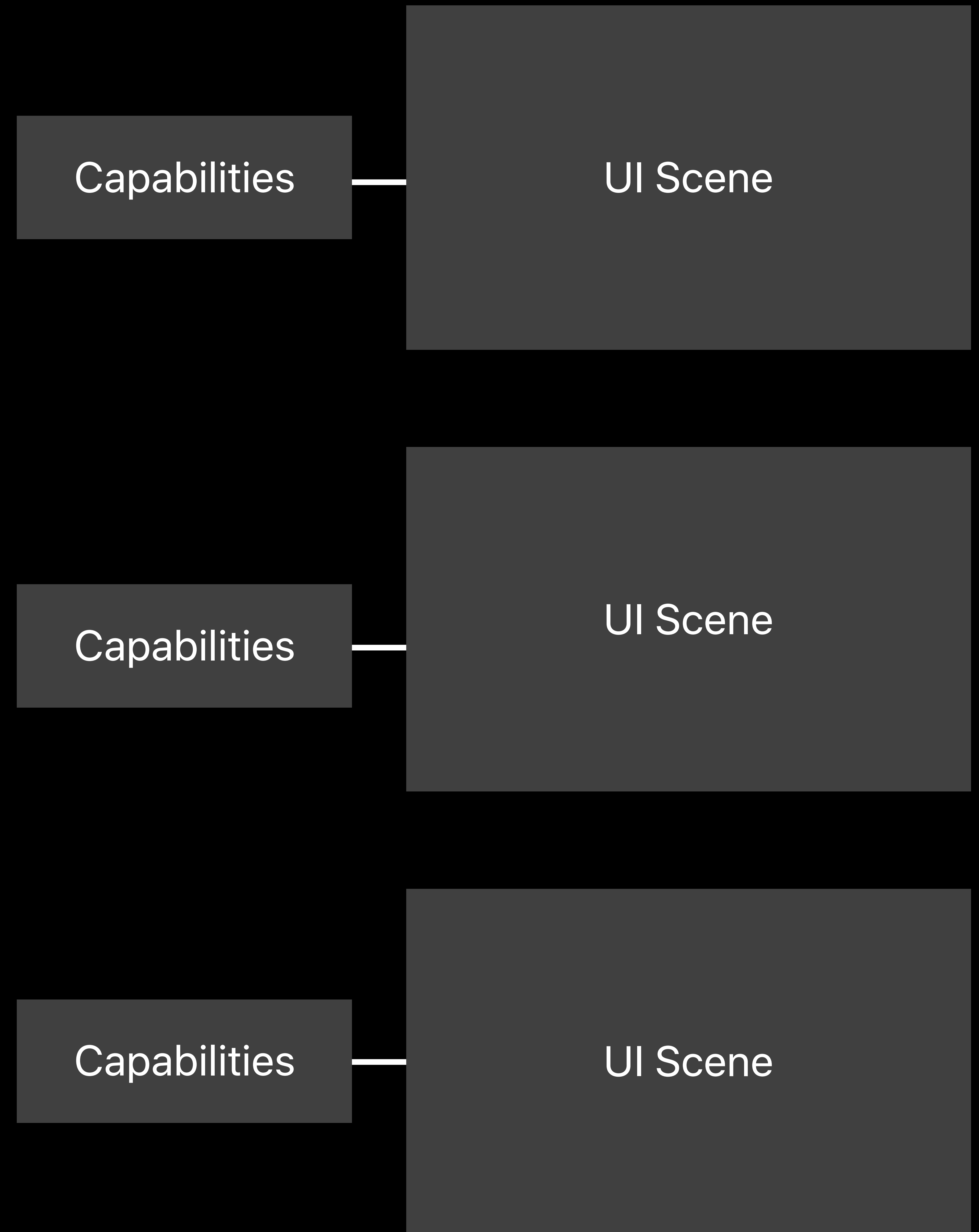
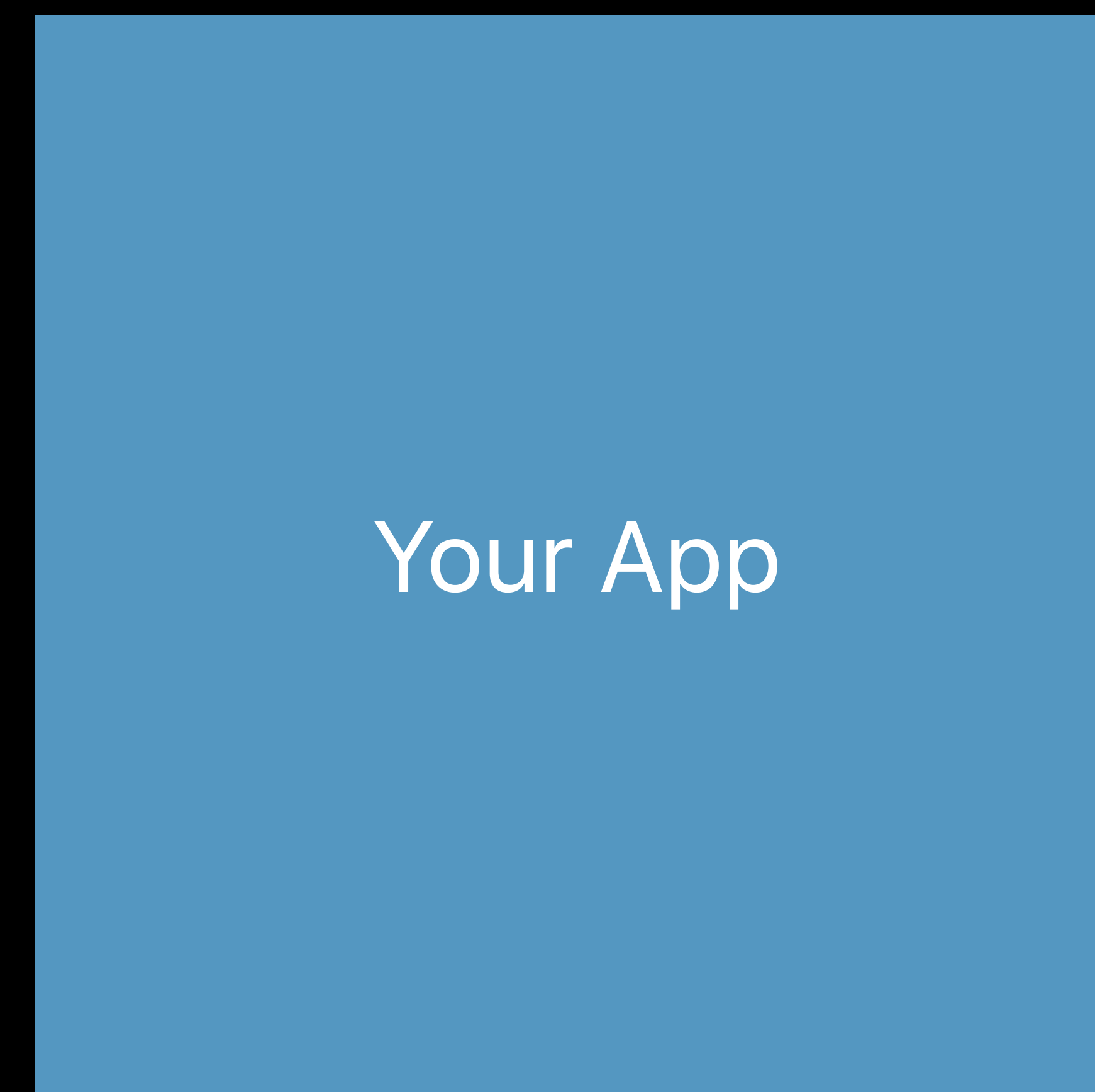
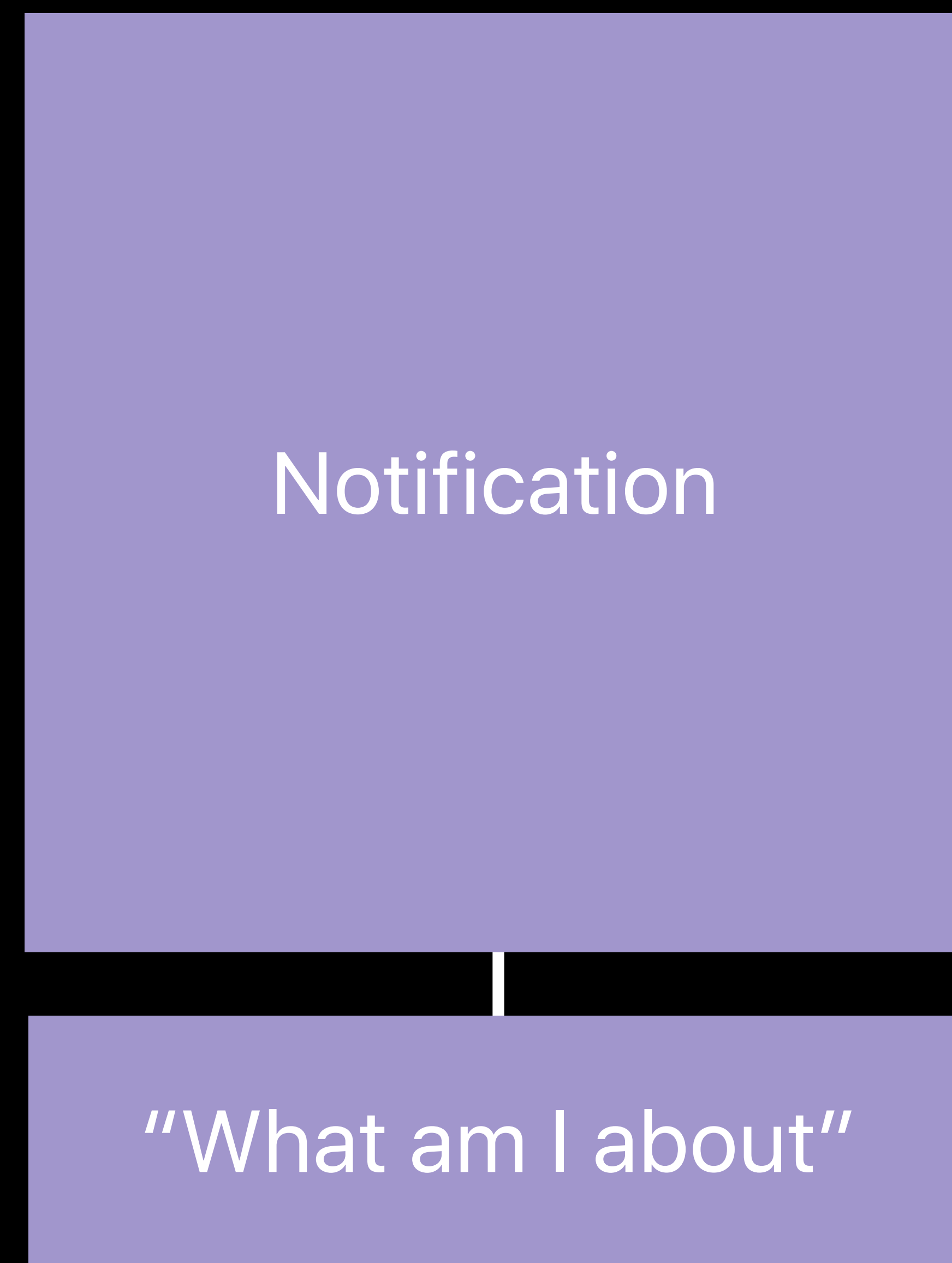


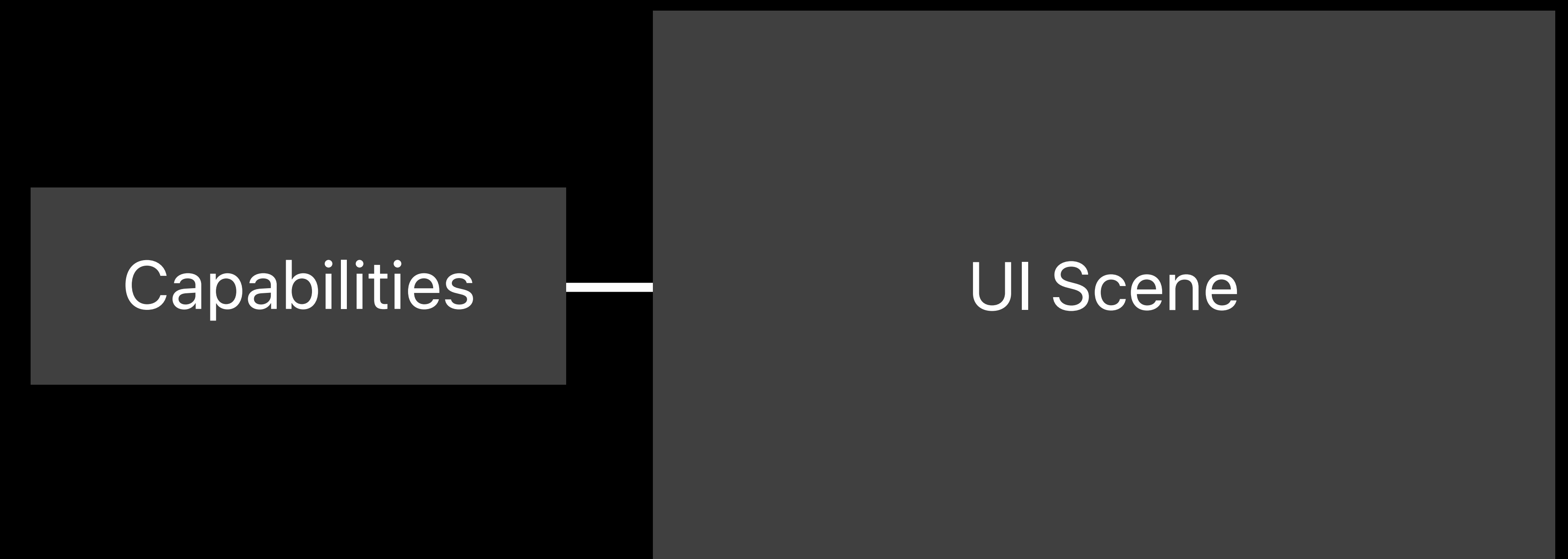
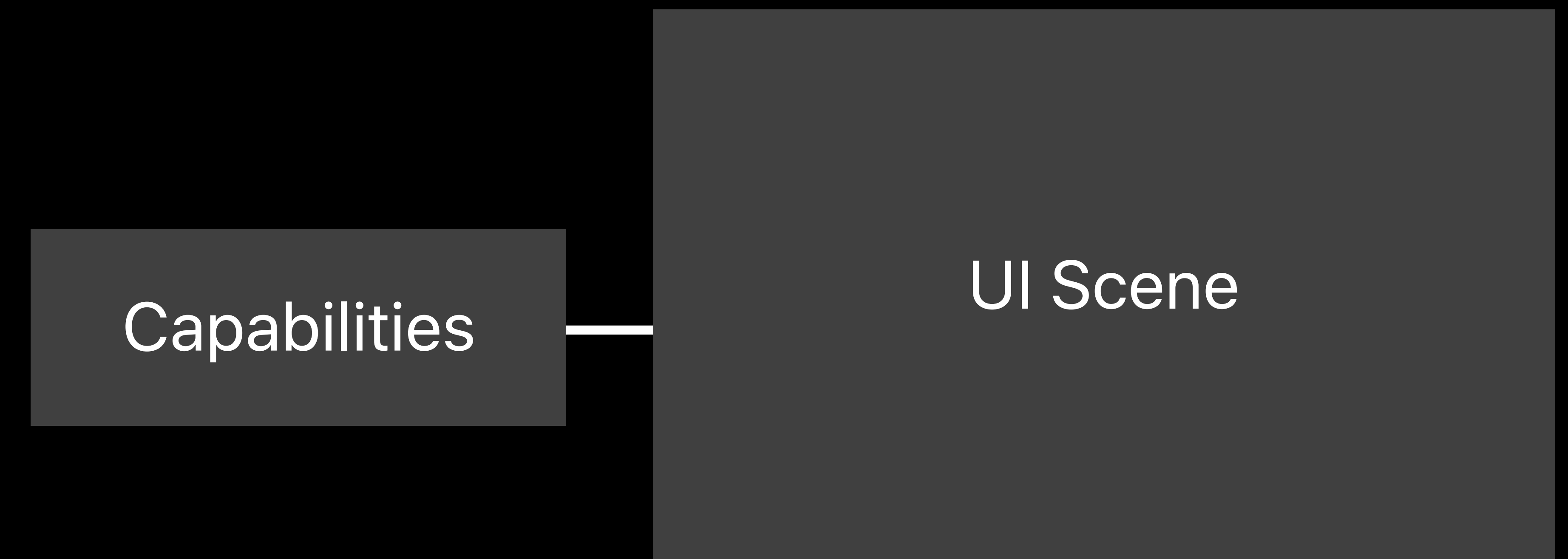
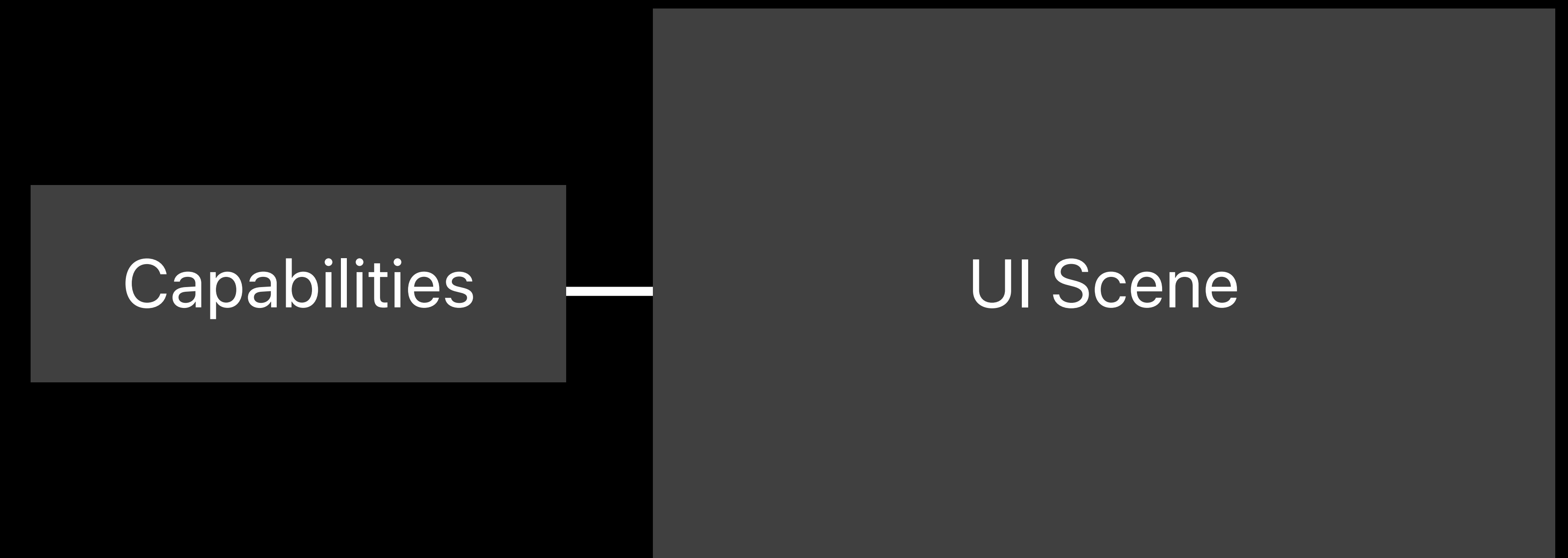
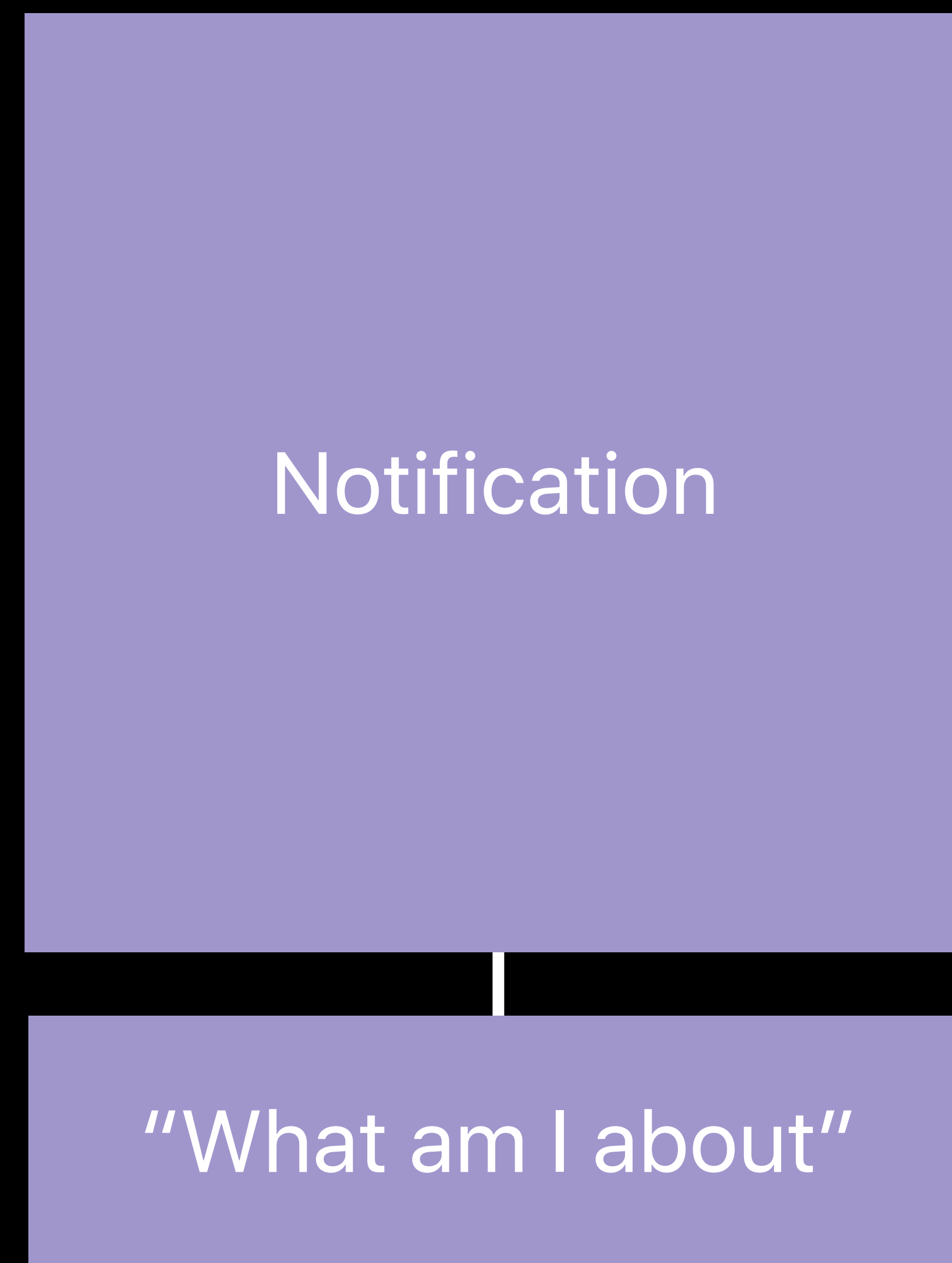


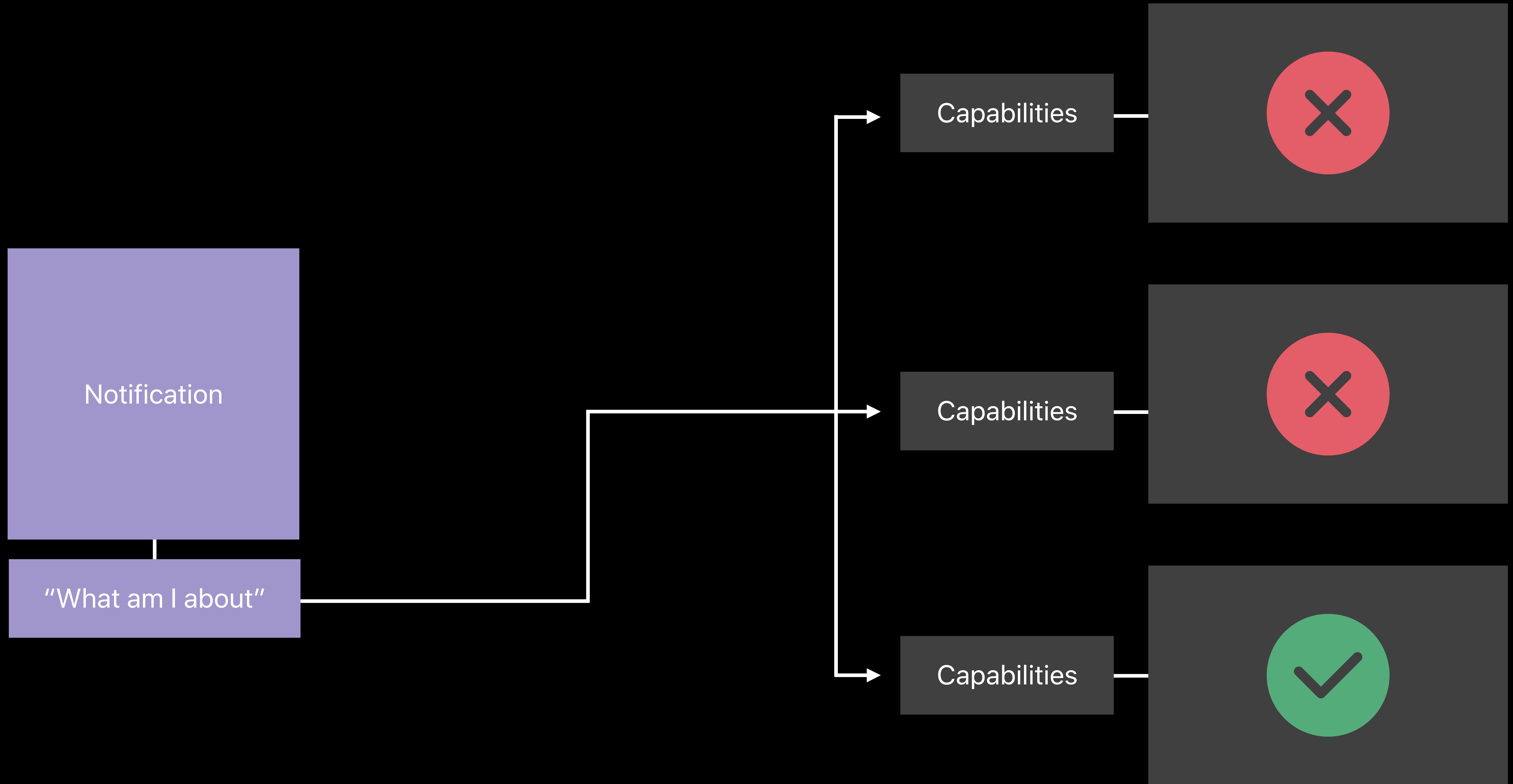
"What am I about"

A smaller light purple rectangular box with the text "What am I about" in white, positioned below the notification box and connected to it by a thin white vertical line.









UIScene

UIScene

UISceneActivationConditions

```
class UISceneActivationConditions : NSObject {  
    var canActivateForTargetContentIdentifierPredicate : NSPredicate  
    var prefersToActivateForTargetContentIdentifierPredicate : NSPredicate  
}
```

```
class UISceneActivationConditions : NSObject {  
    var canActivateForTargetContentIdentifierPredicate : NSPredicate  
    var prefersToActivateForTargetContentIdentifierPredicate : NSPredicate  
}
```

```
class UISceneActivationConditions : NSObject {  
    var canActivateForTargetContentIdentifierPredicate : NSPredicate  
    var prefersToActivateForTargetContentIdentifierPredicate : NSPredicate  
}
```



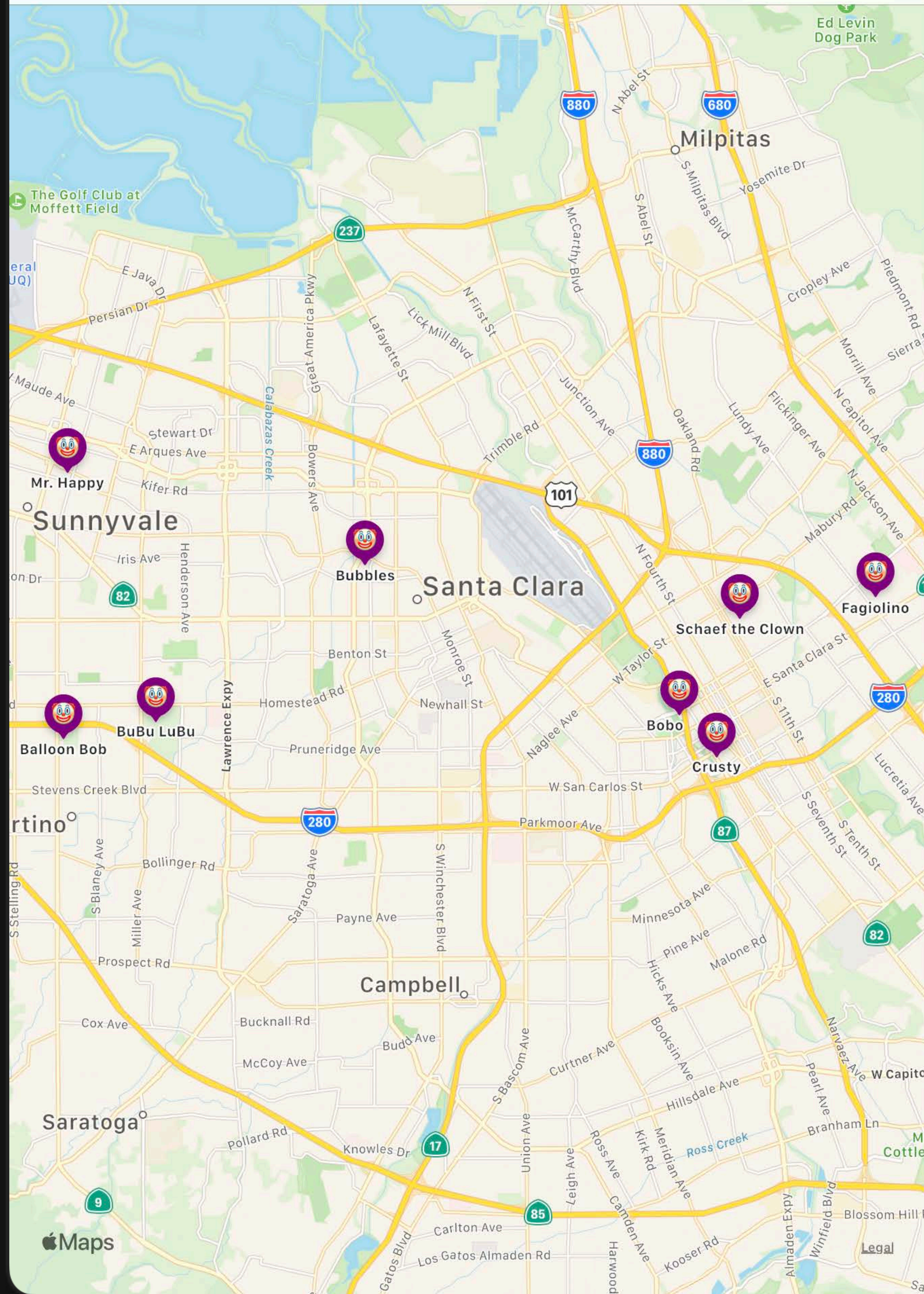
```
class UISceneActivationConditions : NSObject {  
    var canActivateForTargetContentIdentifierPredicate : NSPredicate  
    var prefersToActivateForTargetContentIdentifierPredicate : NSPredicate  
}
```

```
class UISceneActivationConditions : NSObject {  
    var canActivateForTargetContentIdentifierPredicate : NSPredicate  
    var prefersToActivateForTargetContentIdentifierPredicate : NSPredicate  
}
```

```
class UISceneActivationConditions : NSObject {  
    var canActivateForTargetContentIdentifierPredicate : NSPredicate  
    var prefersToActivateForTargetContentIdentifierPredicate : NSPredicate  
}
```

A Target Content Identifier is a string

Welcome to Clown Town 🤡

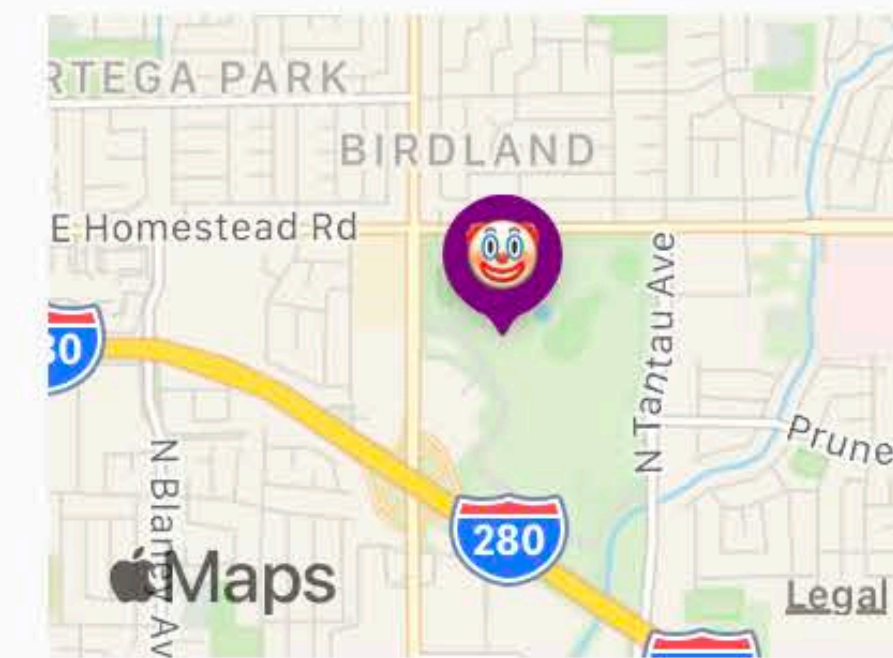


BuBu LuBu ★★★★★

Done



Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.



[WRITE A REVIEW](#)

[TRACK AVAILABILITY](#)

[OPEN IN NEW WINDOW](#)

<http://clowntown.example.com/clown/Crusty>

Target Content Identifiers

Target Content Identifiers

UNNotificationContent

- APNS `"target-content-id"` key

Target Content Identifiers

UNNotificationContent

- APNS `"target-content-id"` key

UIApplicationShortcutItem

Target Content Identifiers

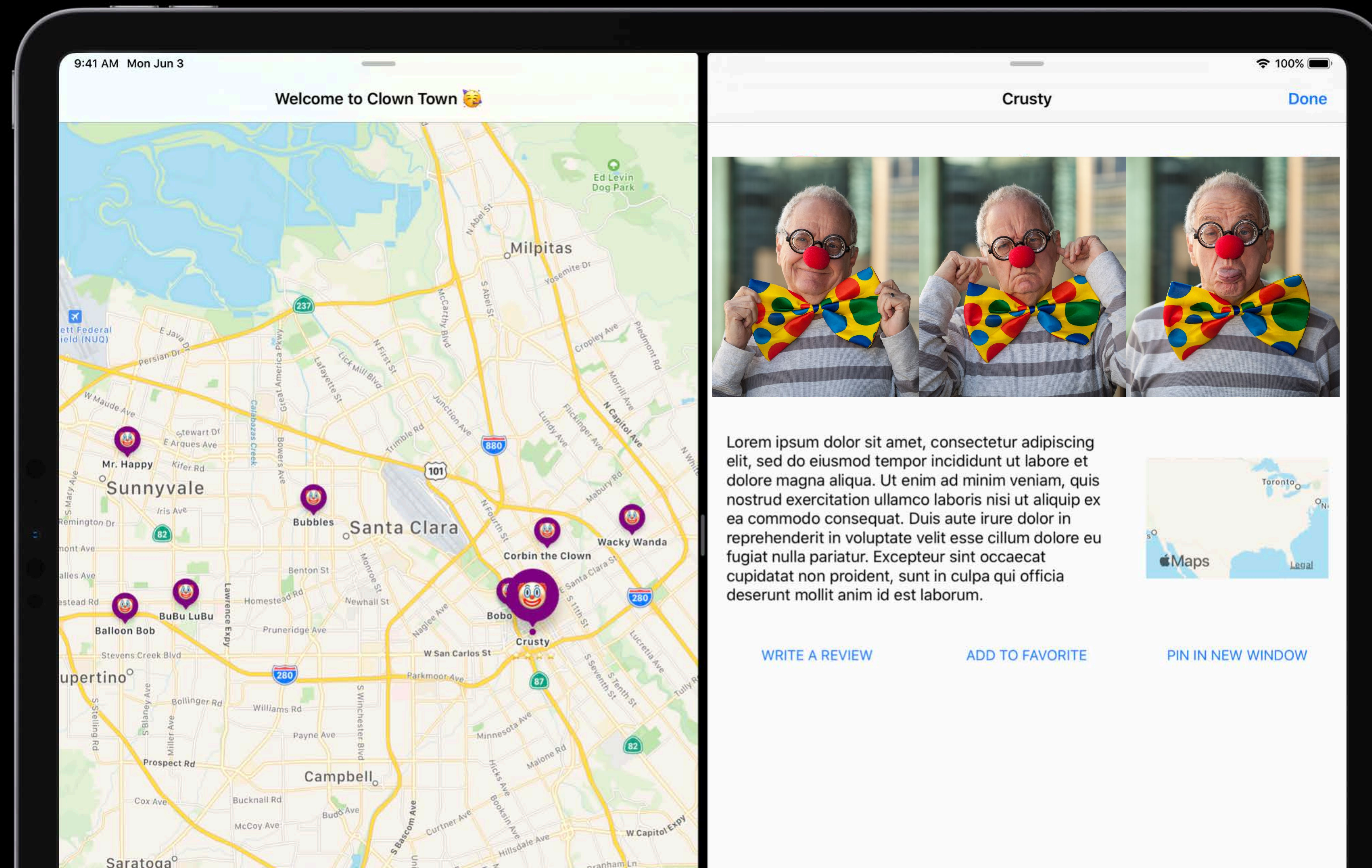
UNNotificationContent

- APNS `"target-content-id"` key

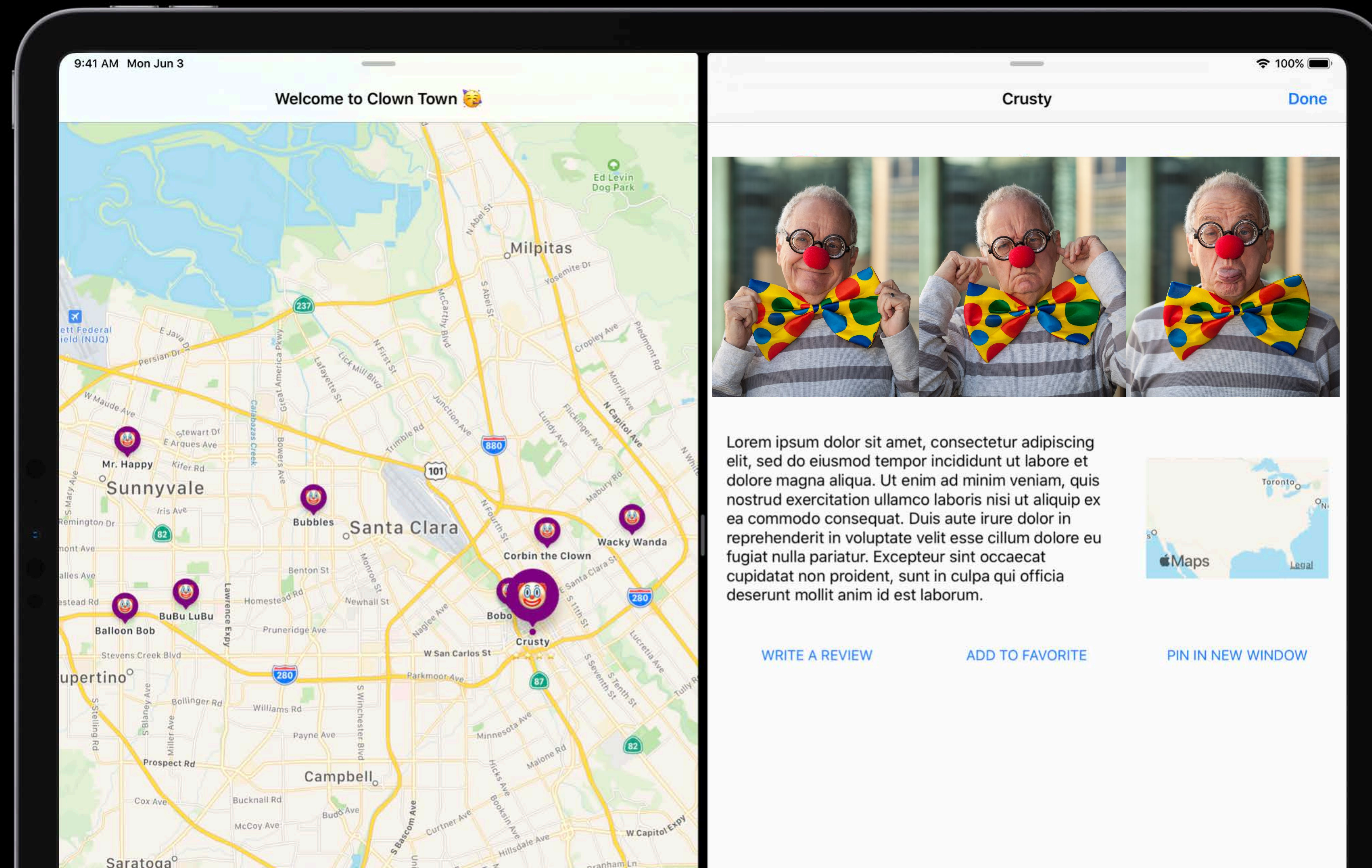
UIApplicationShortcutItem

NSUserActivity

http://clowntown.example.com/clown/Crusty

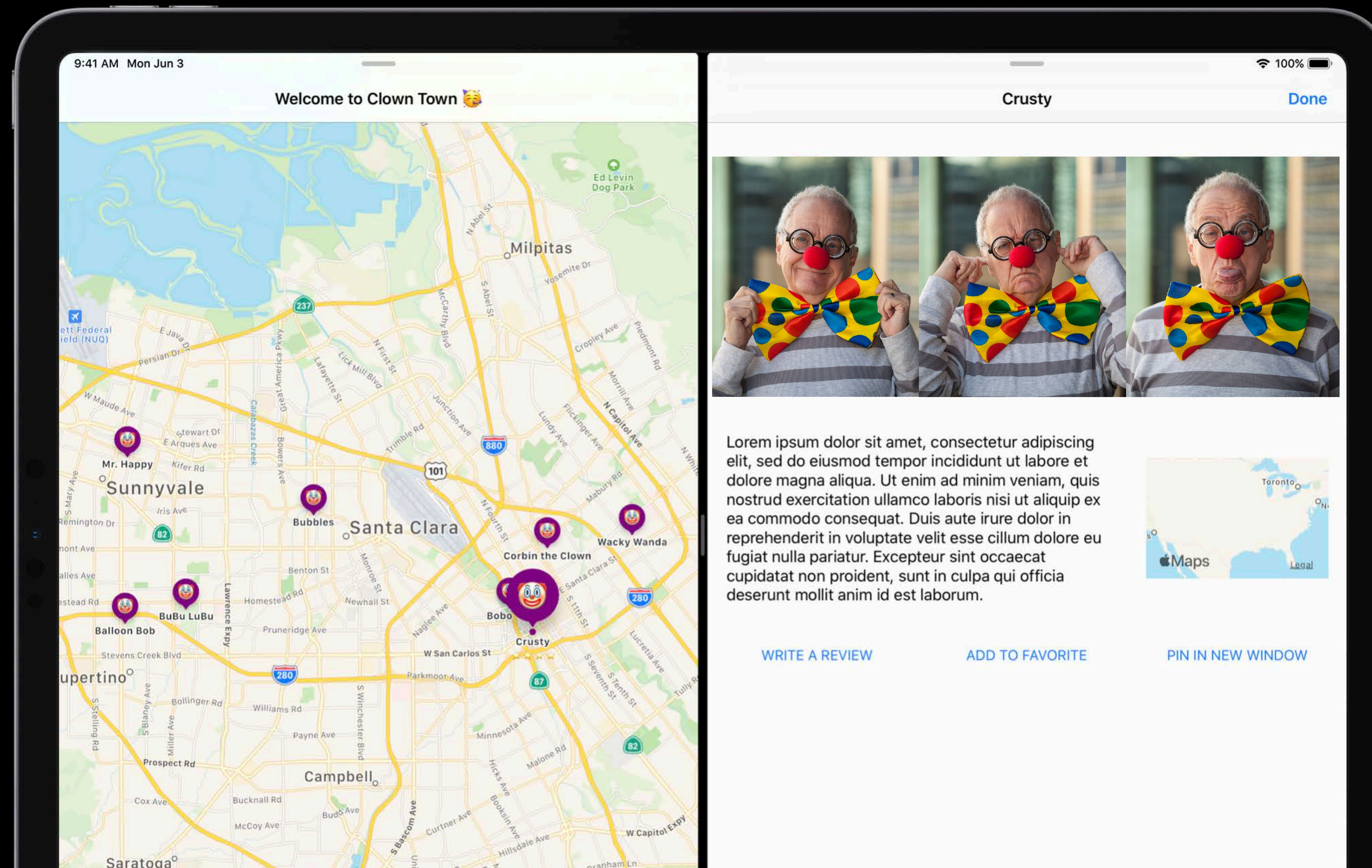


http://clowntown.example.com/clown/Crusty



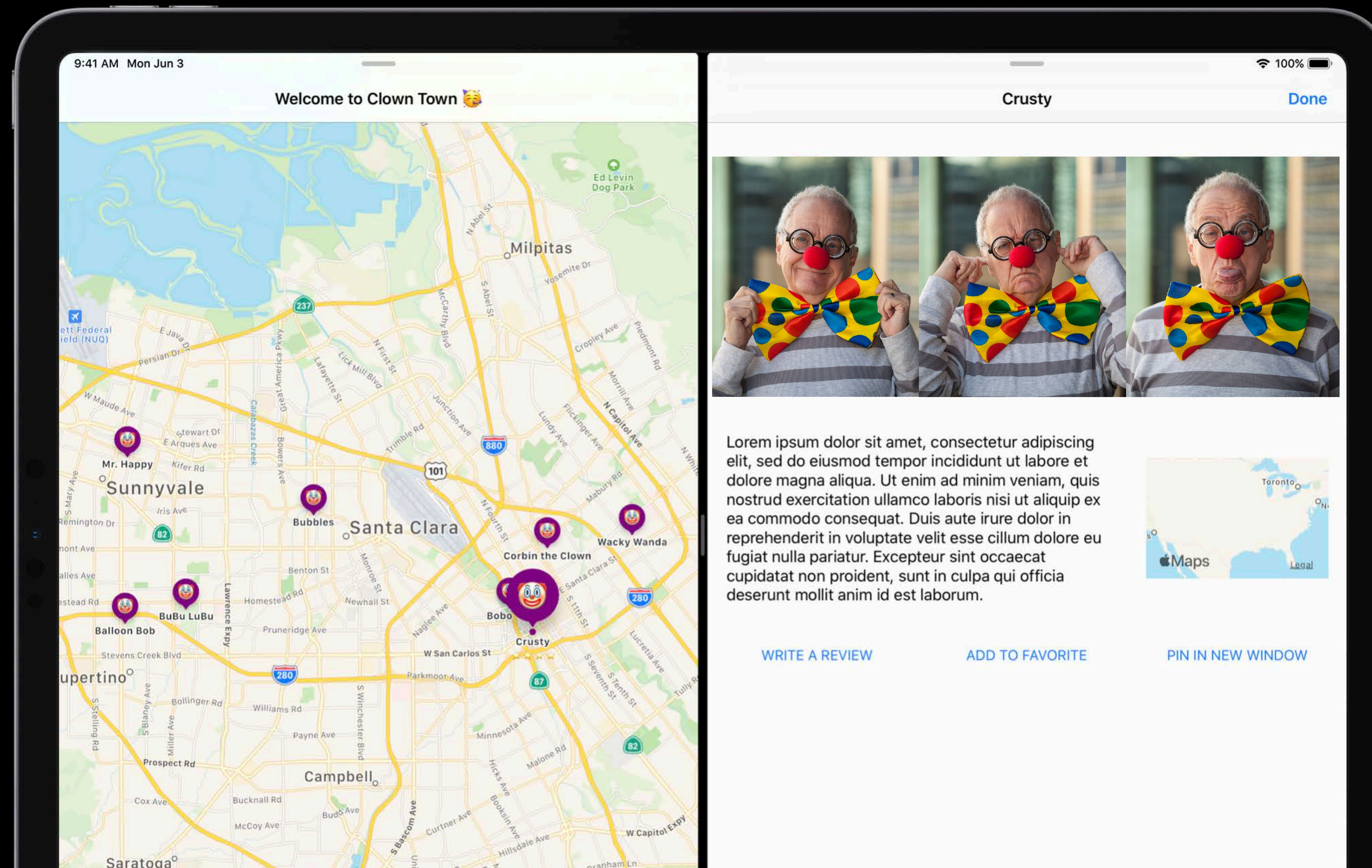
http://clowntown.example.com/clown/Crusty

Can be activated



http://clowntown.example.com/clown/Crusty

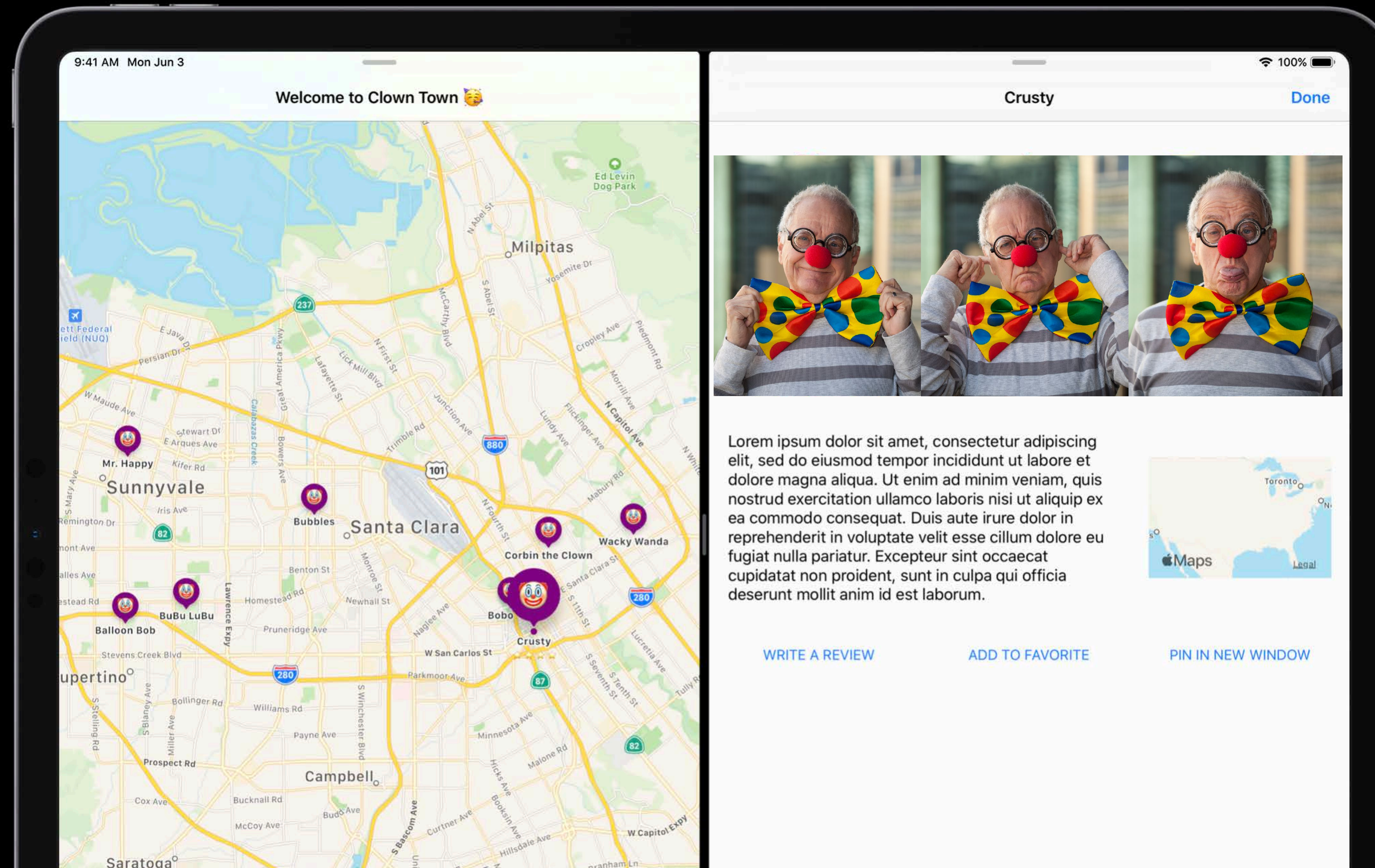
Can be activated



http://clowntown.example.com/clown/Crusty

Can be activated

Can be activated

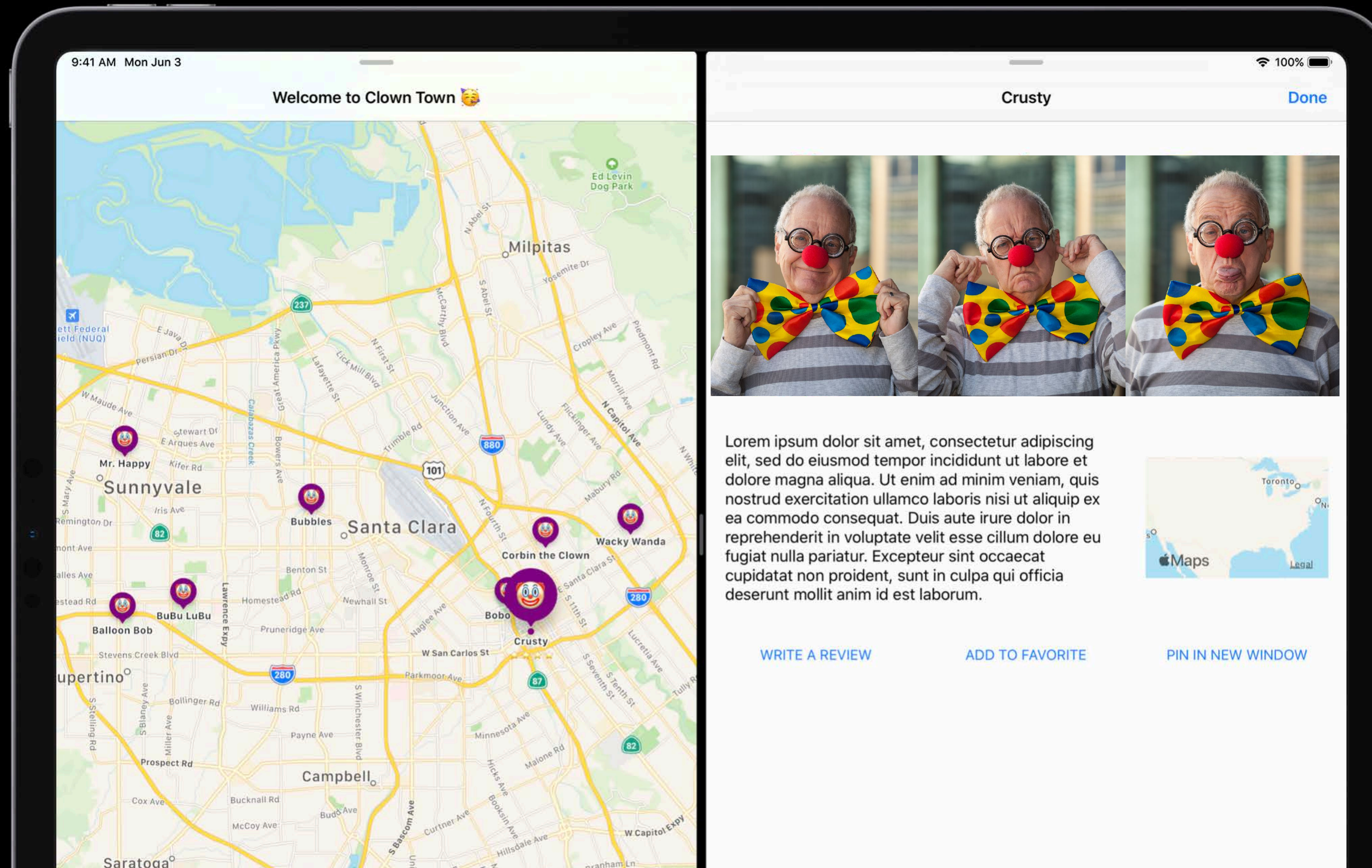


http://clowntown.example.com/clown/Crusty

Can be activated

Can be activated

Prefers activation




```
let crustyPredicate = NSPredicate(format: "self == 'http://clowntown.example.com/clown/  
Crusty'")
```

```
let conditions = scene.activationConditions
```

```
conditions.prefersToActivateForTargetContentIdentifierPredicate = crustyPredicate
```

```
conditions.canActivateForTargetContentIdentifierPredicate = crustyPredicate
```

```
let crustyPredicate = NSPredicate(format: "self == 'http://clowntown.example.com/clown/  
Crusty'")
```

```
let conditions = scene.activationConditions
```

```
conditions.prefersToActivateForTargetContentIdentifierPredicate = crustyPredicate
```

```
conditions.canActivateForTargetContentIdentifierPredicate = crustyPredicate
```

```
let crustyPredicate = NSPredicate(format: "self == 'http://clowntown.example.com/clown/  
Crusty'")
```

```
let conditions = scene.activationConditions
```

```
conditions.prefersToActivateForTargetContentIdentifierPredicate = crustyPredicate  
conditions.canActivateForTargetContentIdentifierPredicate = crustyPredicate
```

Restrictions: NSPredicate

Restrictions: NSPredicate

✘ Block Based Predicates

Restrictions: NSPredicate

- ✘ Block Based Predicates
- ✘ Regular Expression Predicates

Restrictions: NSPredicate

- ✘ Block Based Predicates
- ✘ Regular Expression Predicates
- ✘ The Only Keypath You Can Reference is "SELF"



iPad Pro



iPad Air
New



iPad



iPad mini
New



Compare



Apple Pencil



Smart Keyboard



Accessories



iOS 12

Welcome to the new generation of iPad.

iPad Pro

12.9" and 11"
Liquid Retina display.
A12X Bionic chip.
Face ID.

Starting at \$799




```
let canPredicate = NSPredicate(value: true)

let subpredicates = []
for tab in openTabs {
    subpredicates.append(NSPredicate(format: "self == %@", tab.targetContentIdentifier))
}
let prefersPredicate = NSCompoundPredicate(orPredicateWithSubpredicates: subpredicates)
```

```
let canPredicate = NSPredicate(value: true)

let subpredicates = []
for tab in openTabs {
    subpredicates.append(NSPredicate(format: "self == %@", tab.targetContentIdentifier))
}
let prefersPredicate = NSCompoundPredicate(orPredicateWithSubpredicates: subpredicates)
```

```
let canPredicate = NSPredicate(value: true)
```

```
let subpredicates = []
```

```
for tab in openTabs {
```

```
    subpredicates.append(NSPredicate(format: "self == %@", tab.targetContentIdentifier))
```

```
}
```

```
let prefersPredicate = NSCompoundPredicate(orPredicateWithSubpredicates: subpredicates)
```

```
let canPredicate = NSPredicate(value: true)

let subpredicates = []
for tab in openTabs {
    subpredicates.append(NSPredicate(format: "self == %@", tab.targetContentIdentifier))
}
let prefersPredicate = NSCompoundPredicate(orPredicateWithSubpredicates: subpredicates)
```

```
let canPredicate = NSPredicate(value: true)
```

```
let subpredicates = []
```

```
for tab in openTabs {
```

```
    subpredicates.append(NSPredicate(format: "self == %@", tab.targetContentIdentifier))
```

```
}
```

```
let prefersPredicate = NSCompoundPredicate(orPredicateWithSubpredicates: subpredicates)
```

More Information

developer.apple.com/wwdc19/259

