

#WWDC19

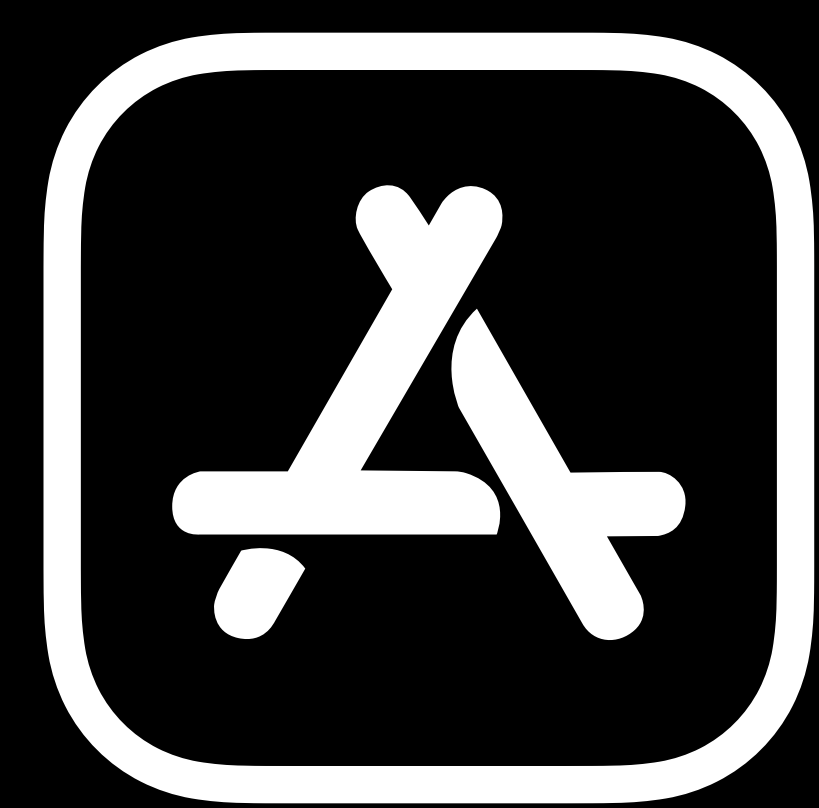
What's New in Core Location

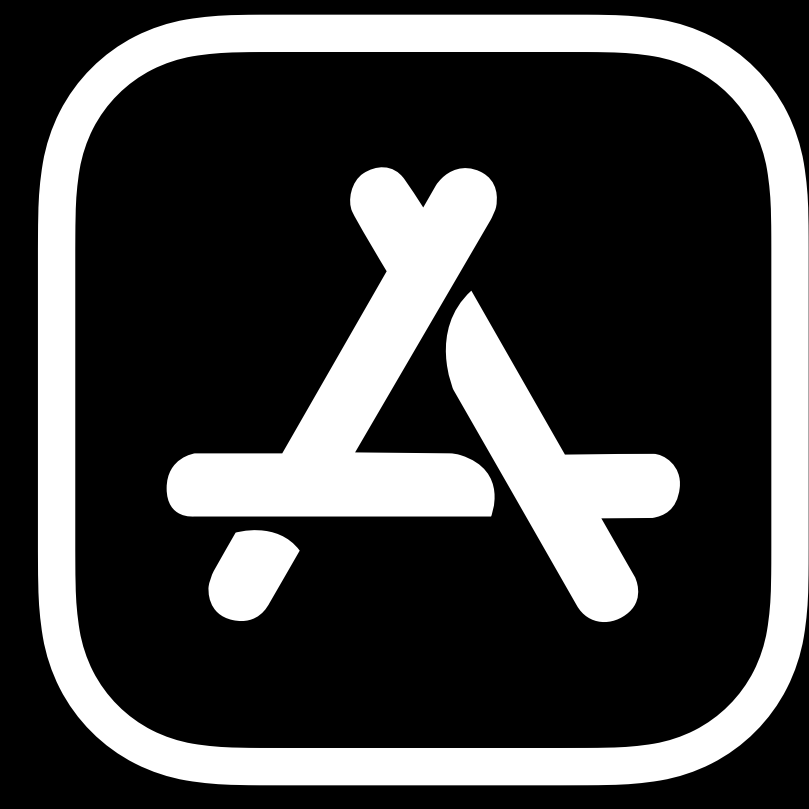
Privacy, authorization, and a dash of new API

Adam Driscoll, Core Location Engineer

Andrea Guzzo, Core Location Engineer









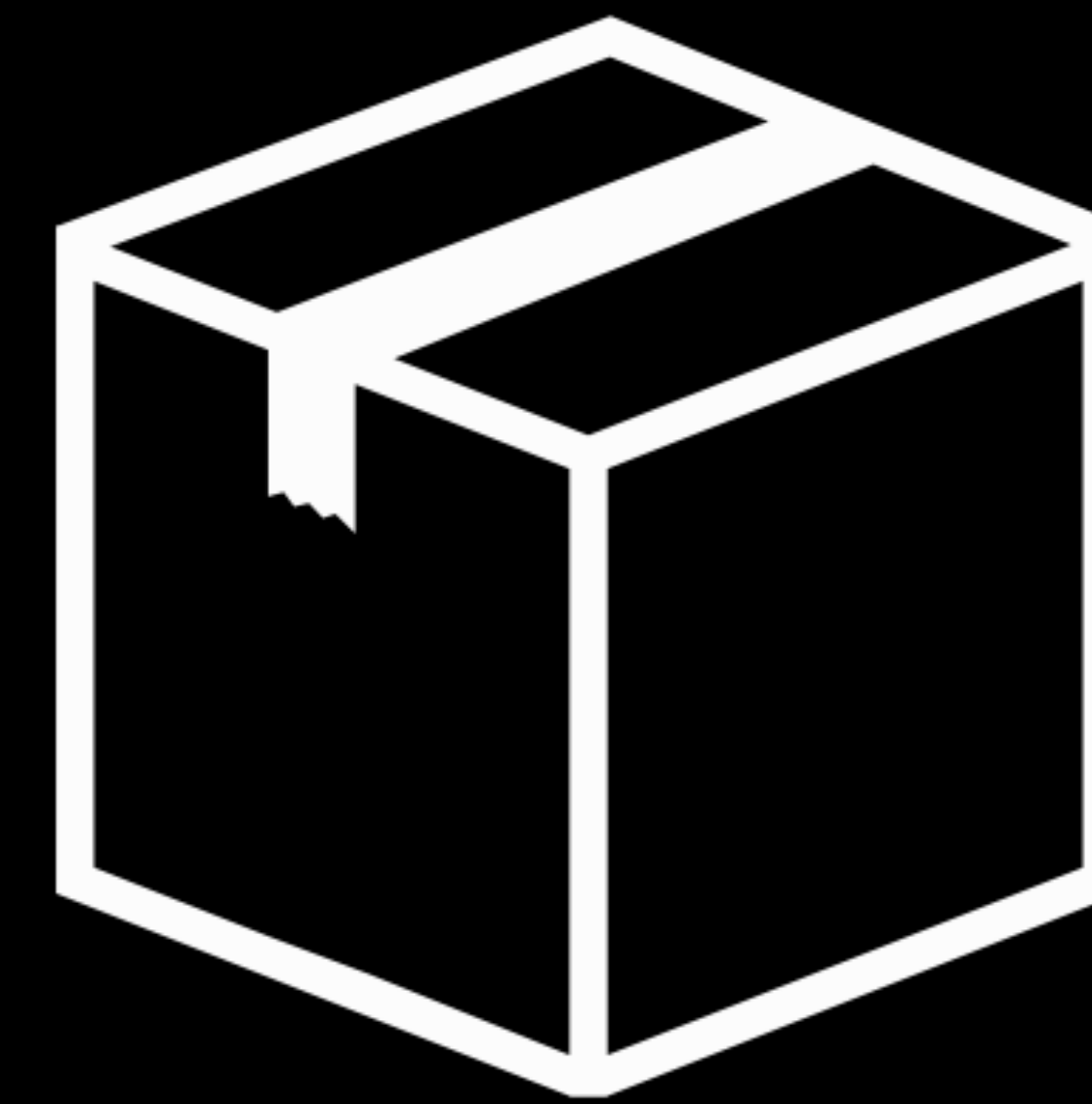
Allow "Travel Eats" to access your location while you are using the app?
Your location is used to display restaurants around you.

Don't Allow Allow

Only While Using the App

Always Allow

Don't Allow





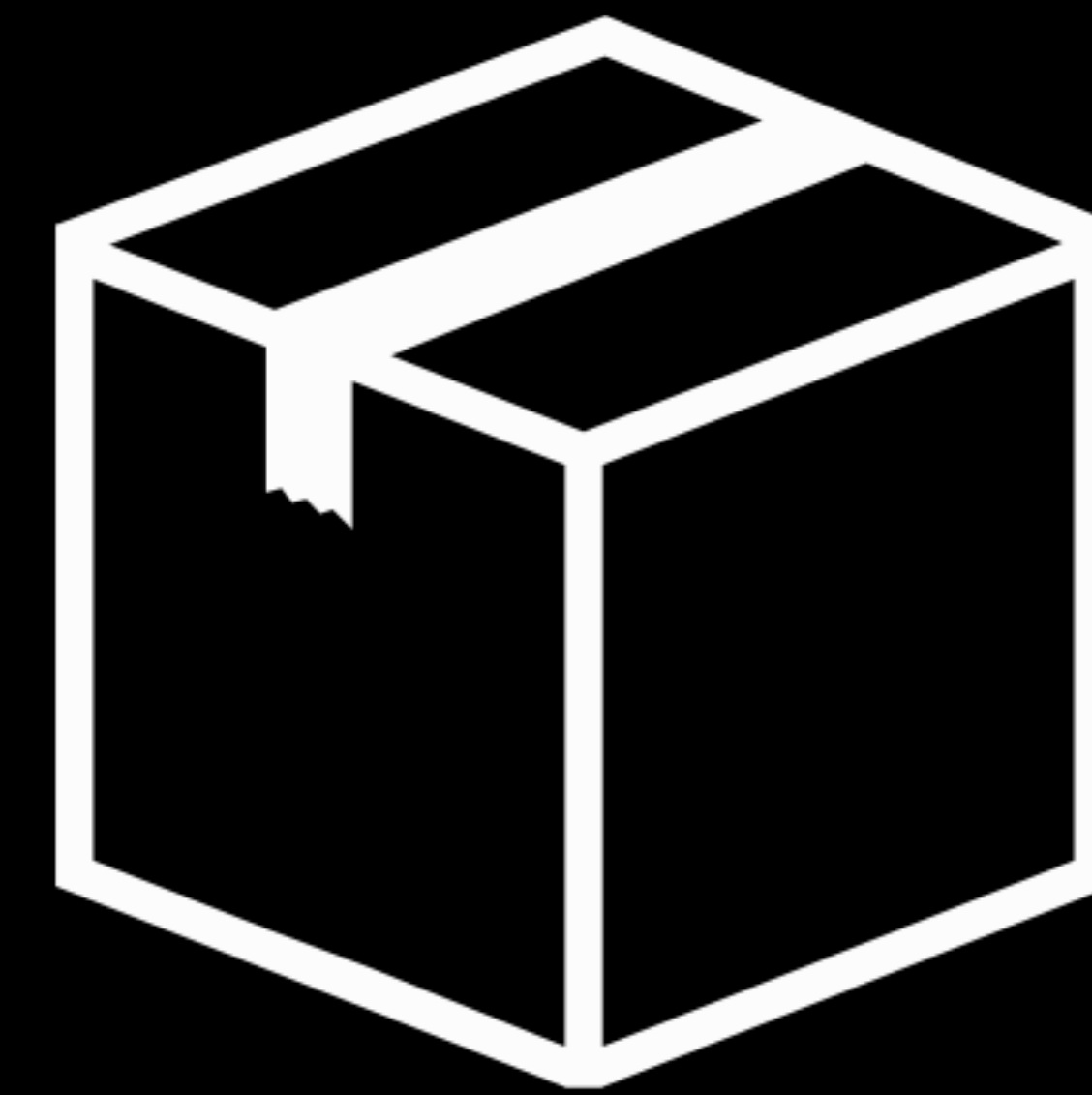
Allow "Travel Eats" to access your location while you are using the app?
Your location is used to display restaurants around you.

Don't Allow Allow

Only While Using the App

Always Allow

Don't Allow



Always Authorization: Deferred

WhenInUse Authorization: Empowered

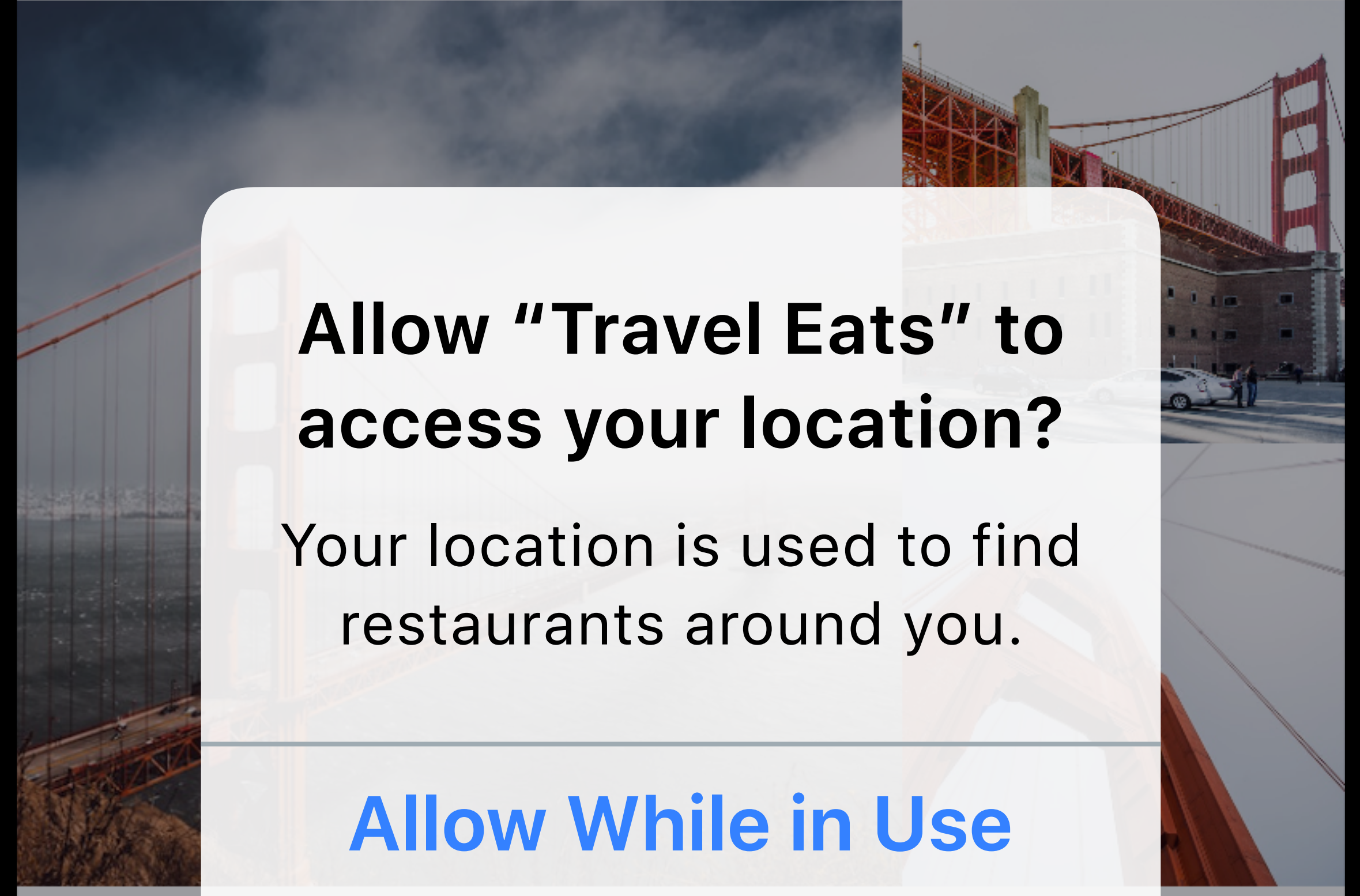
Temporary Authorization: New

Ranging: In Context

9:41



Golden Gate Bridge



**Allow "Travel Eats" to
access your location?**

Your location is used to find
restaurants around you.

[Allow While in Use](#)

[Allow Once](#)

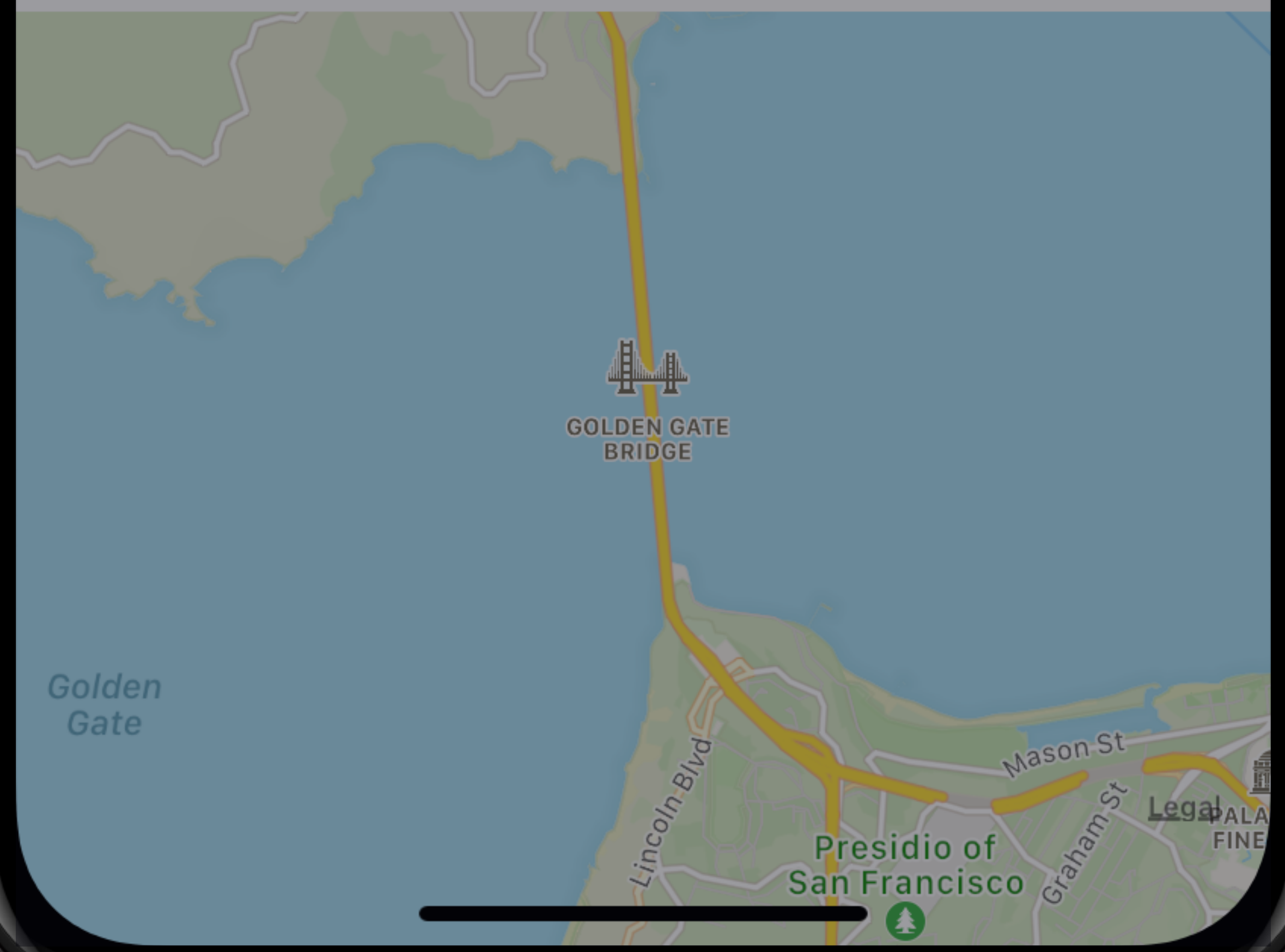
[Don't Allow](#)

The Golden Gate is a suspension bridge
that links San Francisco to Marin County.

Hours

Every Day

Open 24 hours




```
// Requesting an Authorization prompt
```

```
// CLLocationManagers need to have a long lifecycle, so an App Delegate var is ideal
```

```
var locationManager = CLLocationManager()
```

```
// If you do not need Always authorization, request a prompt for WhenInUse authorization
```

```
locationManager.requestWhenInUseAuthorization()
```

```
// -- OR --
```

```
// If you do need Always authorization, request a prompt for that
```

```
locationManager.requestAlwaysAuthorization()
```

```
// Requesting an Authorization prompt
```

```
// CLLocationManagers need to have a long lifecycle, so an App Delegate var is ideal
```

```
var locationManager = CLLocationManager()
```

```
// If you do not need Always authorization, request a prompt for WhenInUse authorization
```

```
locationManager.requestWhenInUseAuthorization()
```

```
// -- OR --
```

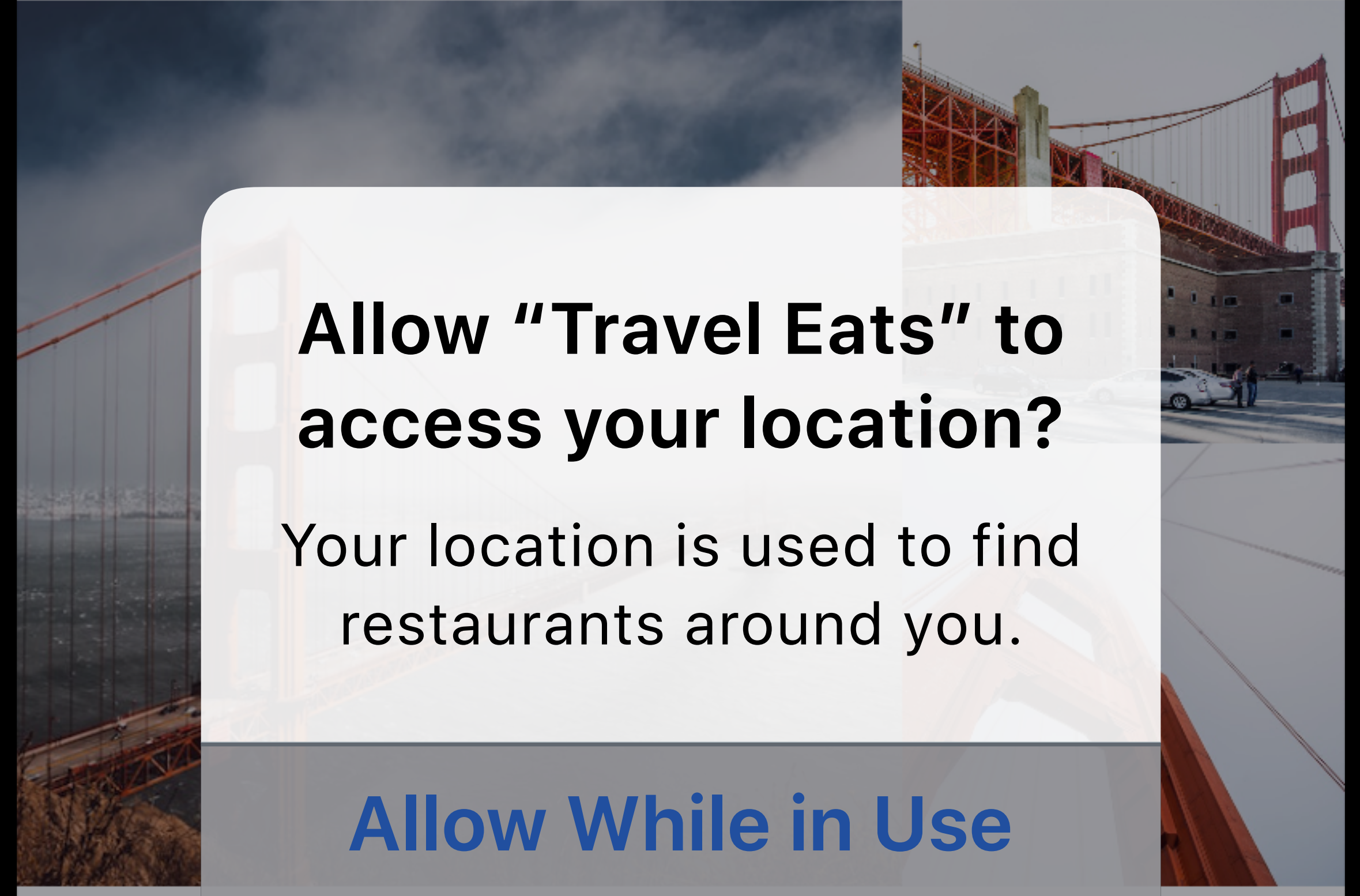
```
// If you do need Always authorization, request a prompt for that
```

```
locationManager.requestAlwaysAuthorization()
```

9:41



Golden Gate Bridge



**Allow "Travel Eats" to
access your location?**

Your location is used to find
restaurants around you.

[Allow While in Use](#)

[Allow Once](#)

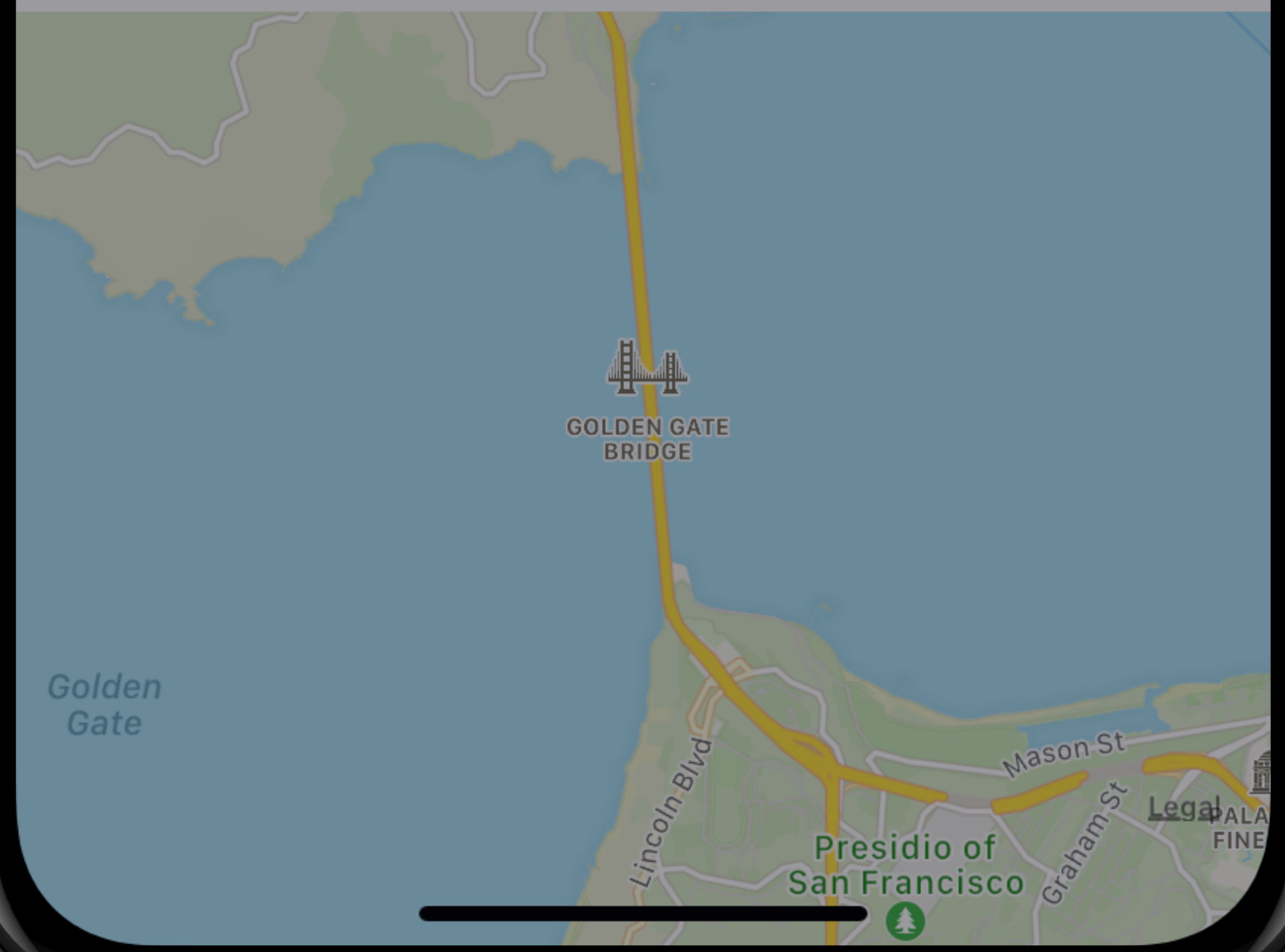
[Don't Allow](#)

The Golden Gate is a suspension bridge
that links San Francisco to Marin County.

Hours

Every Day

Open 24 hours



Provisional Always Authorization

The User Perspective

User picked
While In Use

Settings shows
While In Use

App requested Always

Delegate receives
Always

Start using
Always powers

What Your App Sees

Provisional Always Authorization

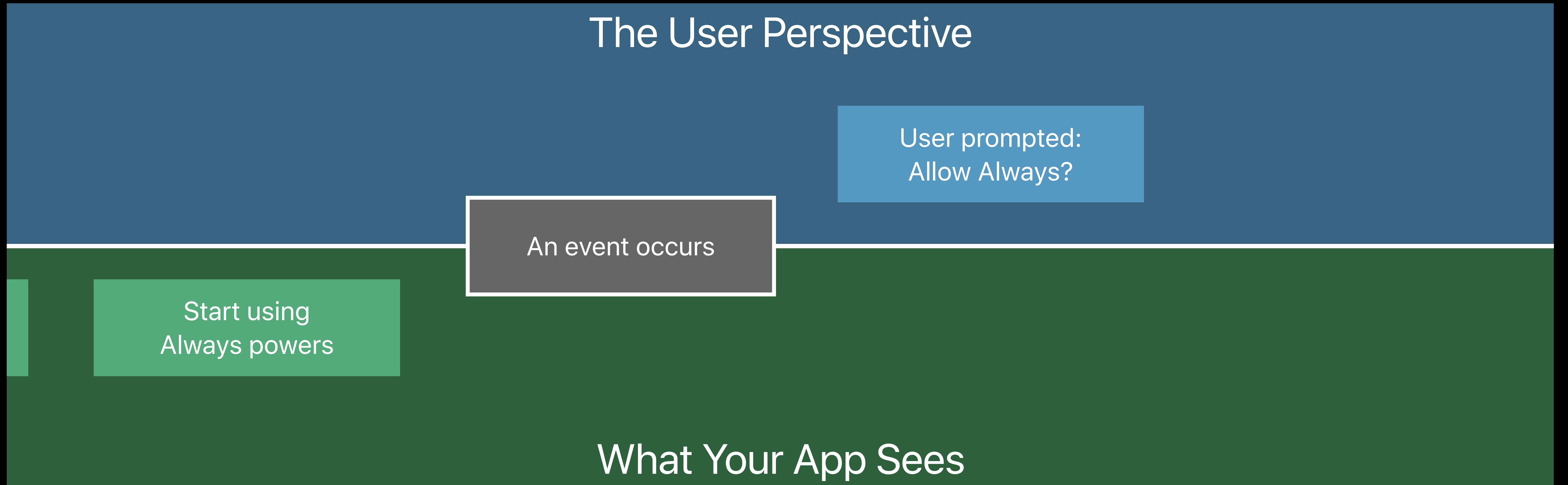
The User Perspective

User prompted:
Allow Always?

An event occurs

Start using
Always powers

What Your App Sees





Allow "Travel Eats" to access your location even when you are not using the app?

Your location is used to find restaurants around you. If you always allow access, you'll never miss nearby Top Eats.

[Keep Only While Using](#)

[Change to Always Allow](#)

Provisional Always Authorization

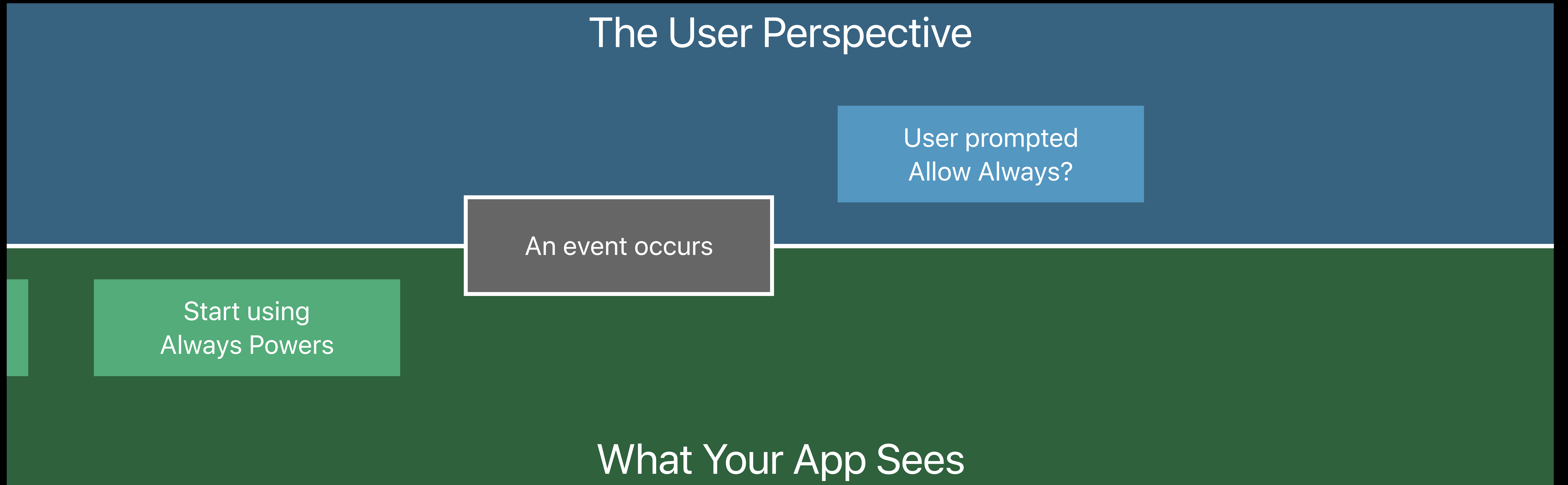
The User Perspective

User prompted
Allow Always?

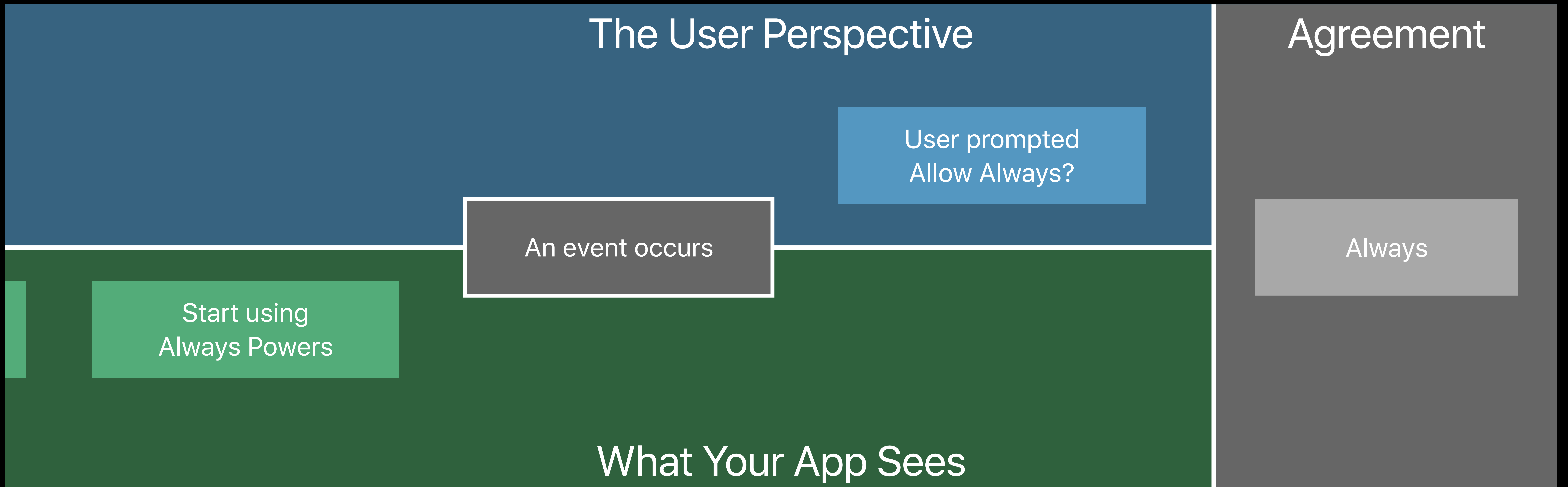
An event occurs

Start using
Always Powers

What Your App Sees



Provisional Always Authorization



Deferred Always Authorization Details

Nothing until the user says "Always"

Prompt will display later

One shot only

Request can be made

- Directly
- As an upgrade

Event Delivery Details

Delivered if app is granted Always

Not delivered if

- User granted WhenInUse instead
- User has not yet been prompted
- Event replaced by a newer event
- Just stale

Other Platforms

tvOS does not support *Always*

watchOS does not need *Always**

macOS does not have *Always* or *WhenInUse*

iPad Apps for Mac may use either *Always* or *WhenInUse*

Always Authorization: Deferred

WhenInUse Authorization: Empowered

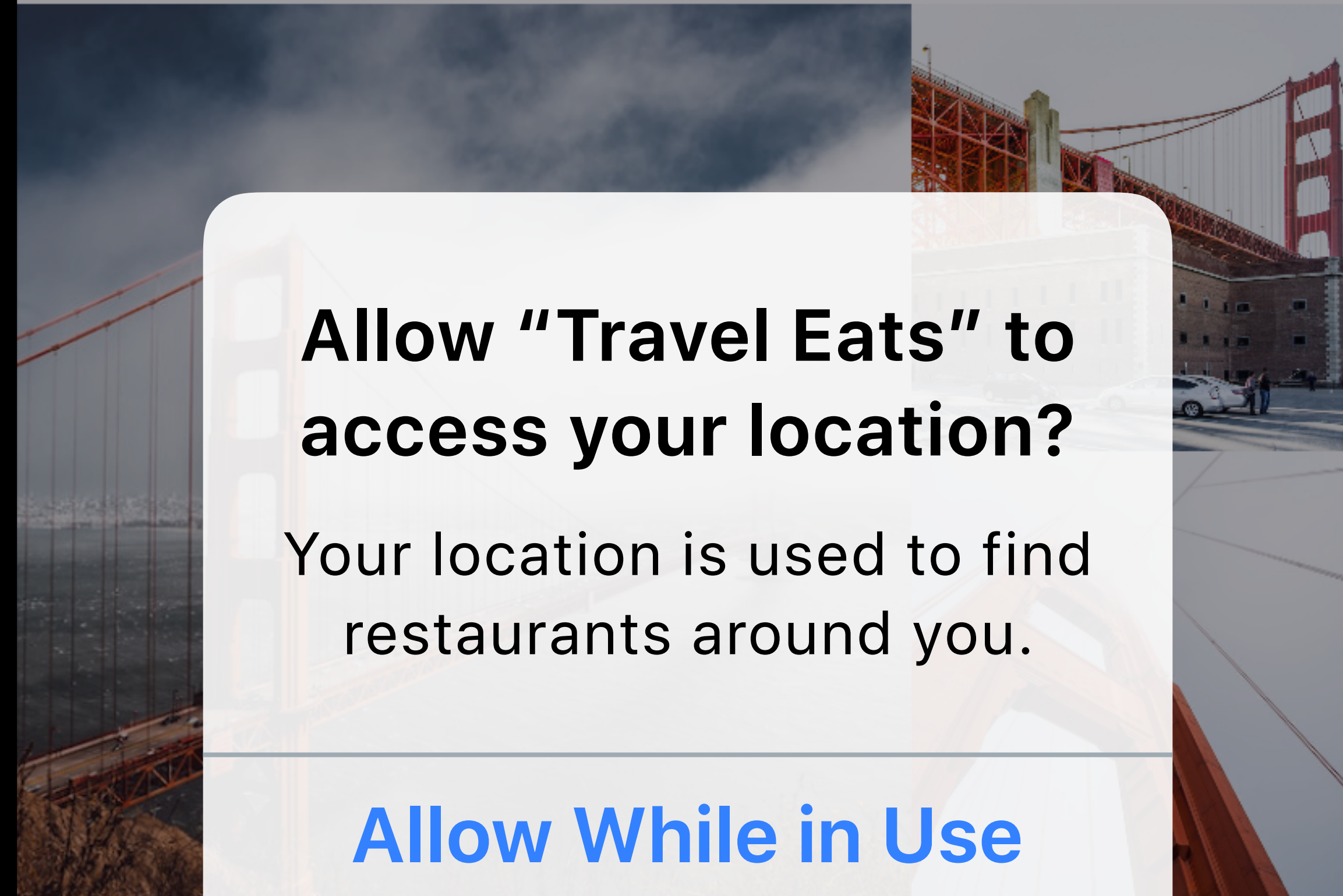
Temporary Authorization: New

Ranging: In Context

9:41



Golden Gate Bridge



**Allow "Travel Eats" to
access your location?**

Your location is used to find
restaurants around you.

[Allow While in Use](#)

[Allow Once](#)

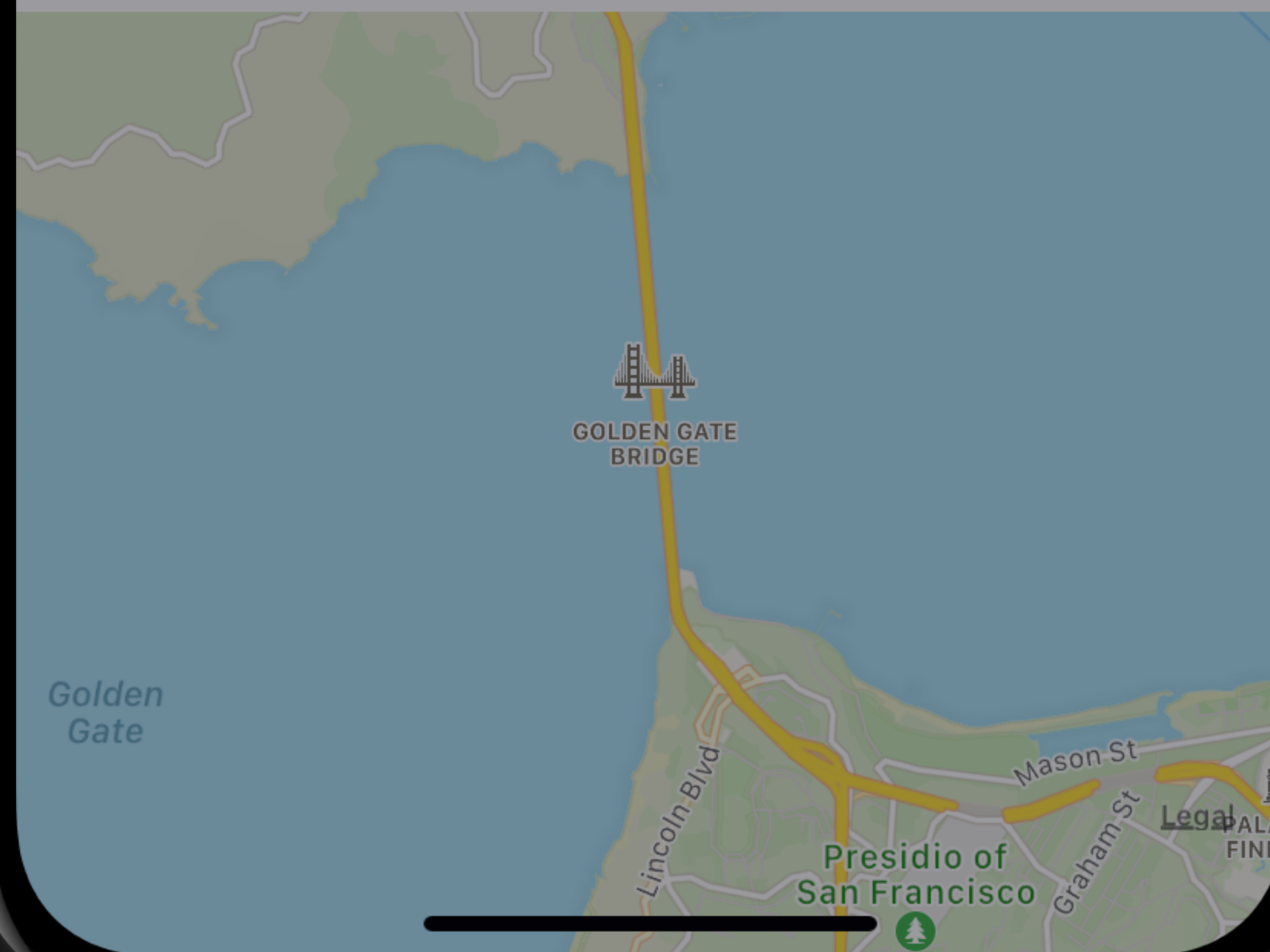
[Don't Allow](#)

The Golden Gate is a suspension bridge
that links San Francisco to Marin County.











Hours

Every Day

Open 24 hours



Authorization, iOS 12

	Always	WhenInUse
Receive Location		
Range Beacons		
Continue Location in the Background		
Start Location in the Background		
Use Significant Location Change Monitoring		
Use Region Monitoring		
Use Visit Monitoring		




Authorization, iOS 13

NEW

Will Core Location API
yield events when my app is

Always

WhenInUse

In use?		
Not in use?		

When is an App In Use?

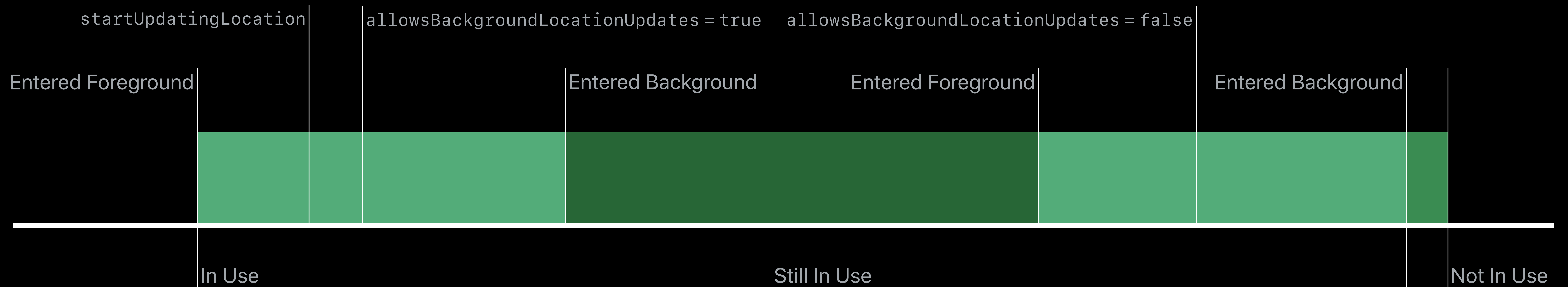
Foreground



When is an App In Use?

Continuous Background Updates

Background Mode: Location updates



When is an App In Use?

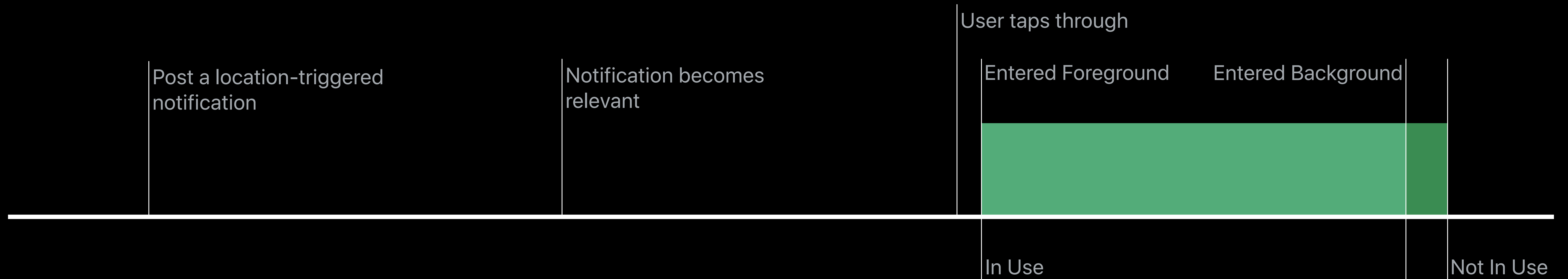
Complications on the Watch



Always In Use

When is an App In Use?

Local notification bootstrap






WhenInUse Authorization, iOS 13

NEW

Will Core Location API
yield events when my app is

Always

WhenInUse

In use?		
Not in use?		

Always Authorization: Deferred

WhenInUse Authorization: Empowered

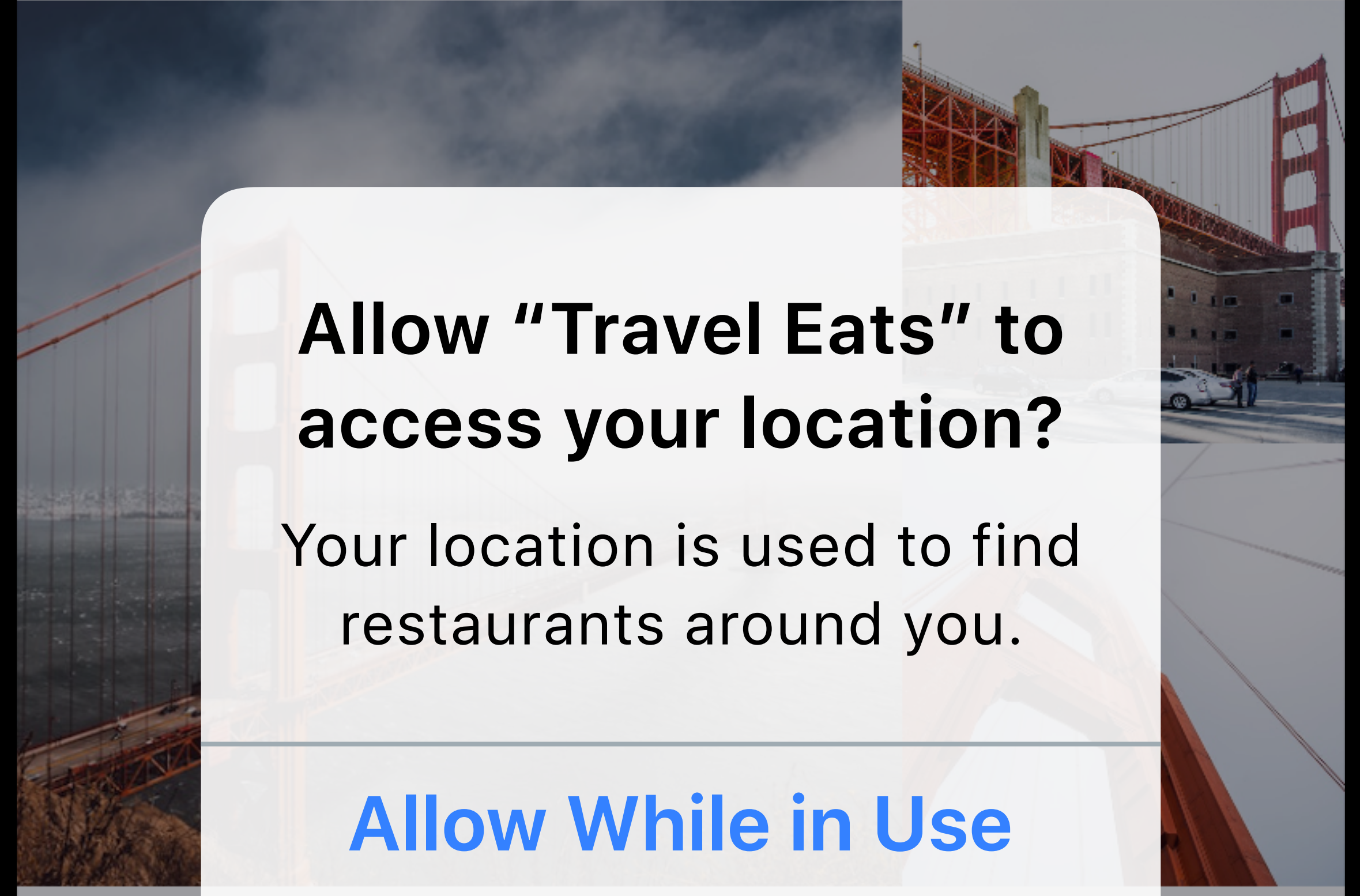
Temporary Authorization: New

Ranging: In Context

9:41



Golden Gate Bridge



Allow "Travel Eats" to access your location?

Your location is used to find restaurants around you.

[Allow While in Use](#)

[Allow Once](#)

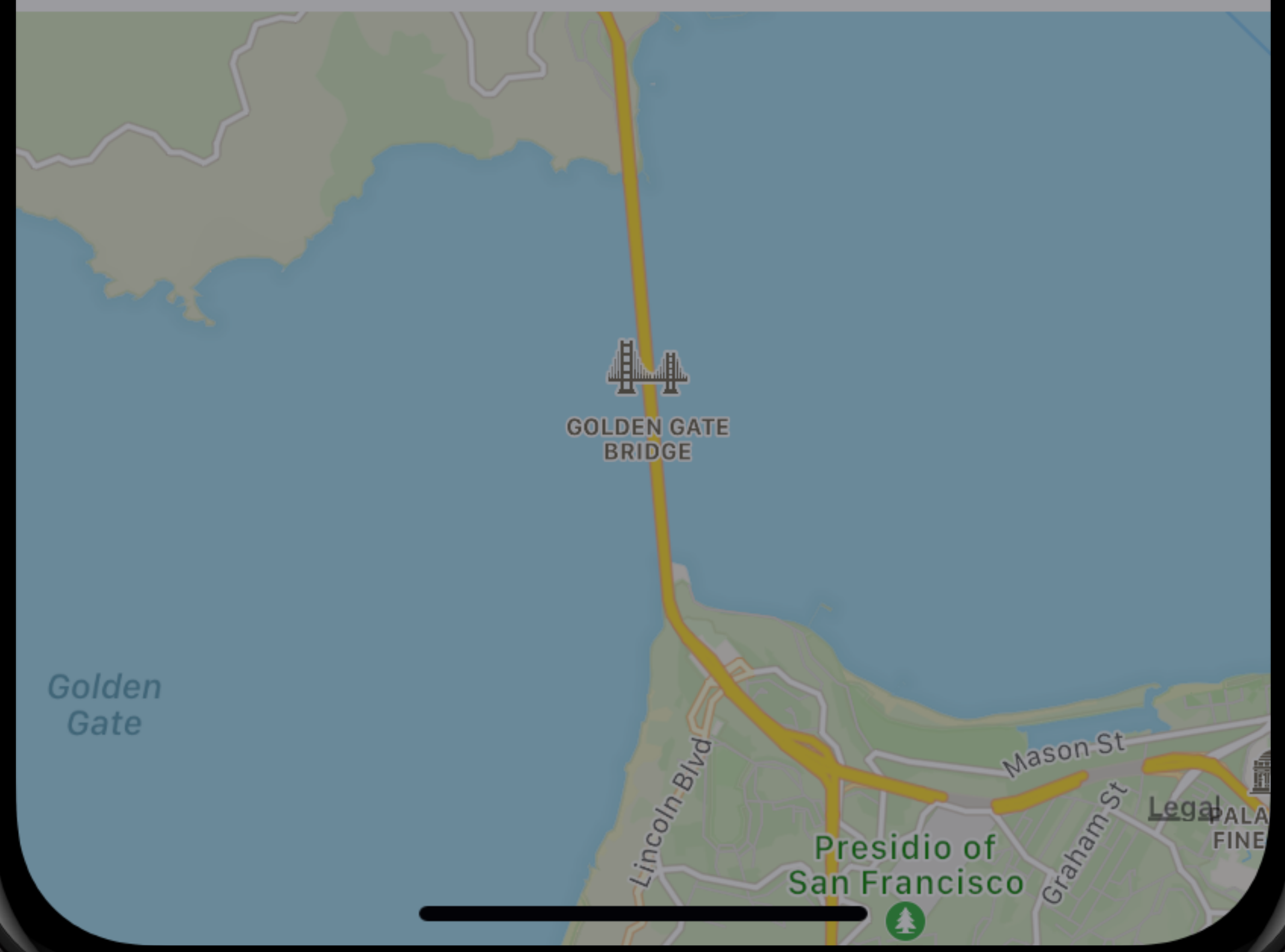
[Don't Allow](#)

The Golden Gate Bridge is a suspension bridge that links San Francisco to Marin County.

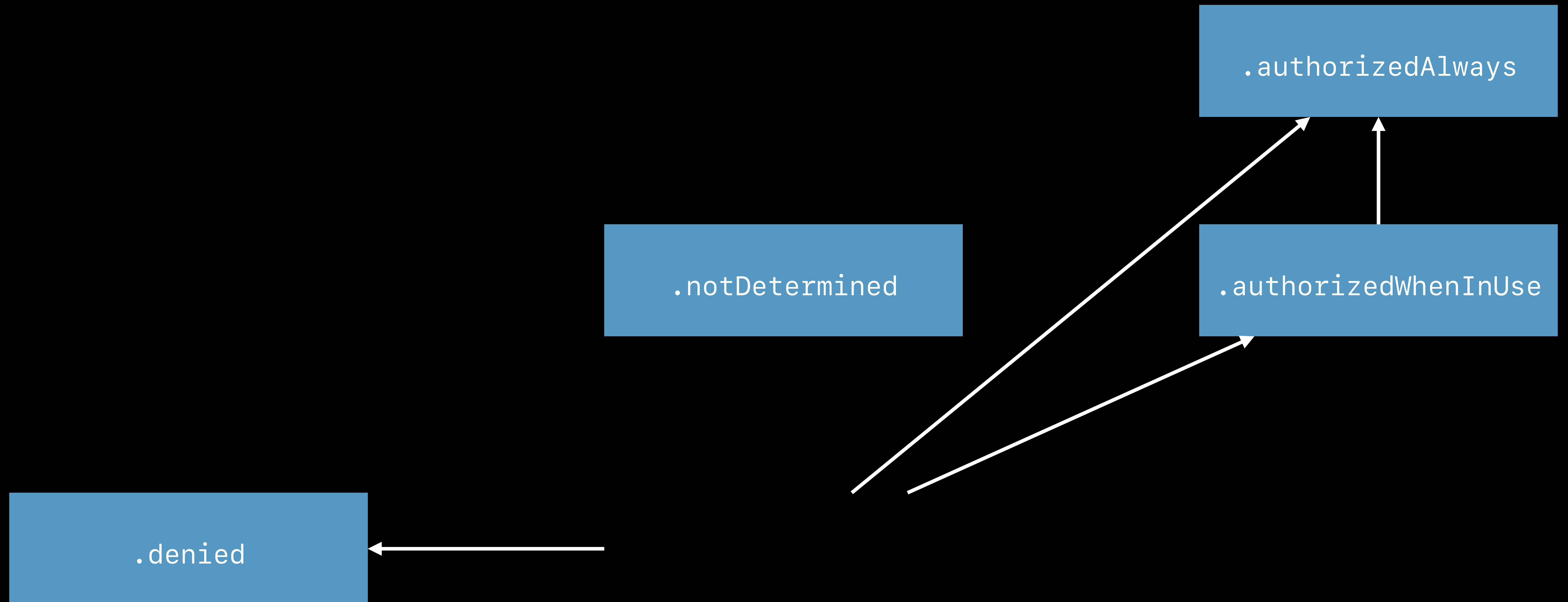
Hours

Every Day

Open 24 hours

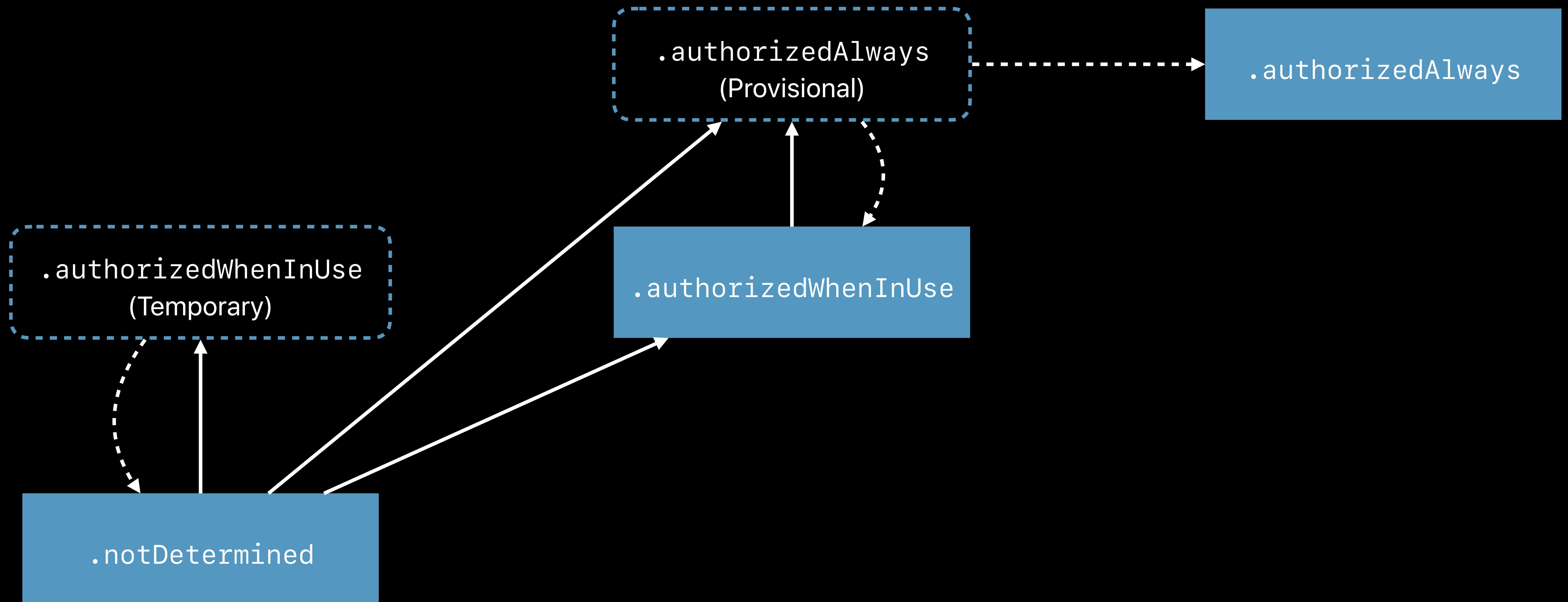


Authorization States, iOS 12



Authorization States, iOS 13

NEW



Allow Once

Temporary authorization

NEW

Grants authorization temporarily

Yields `.authorizedWhenInUse`

Request authorization again next time

Very sensitive to end-of-use

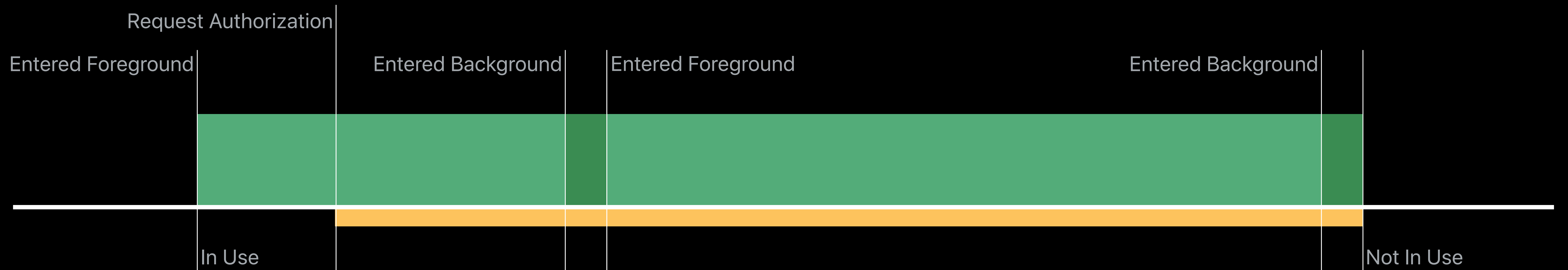
Temporary Authorization Case Study

Foreground



Temporary Authorization Case Study

Foreground



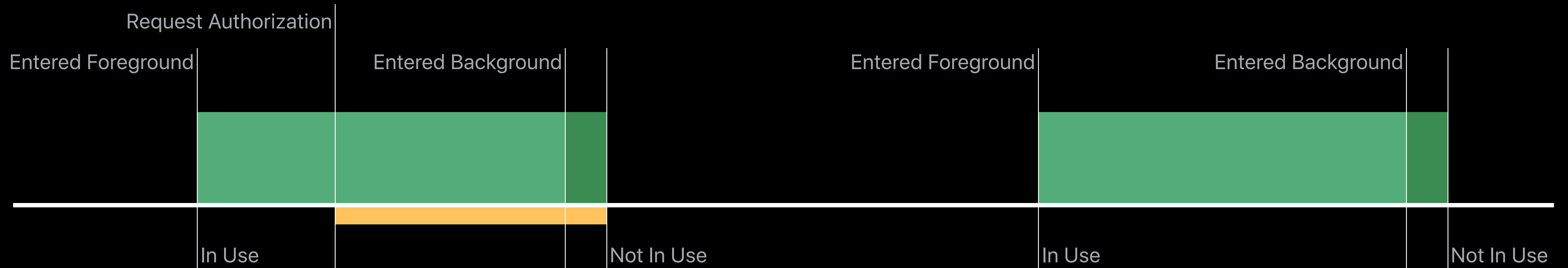
Temporary Authorization Case Study

Foreground



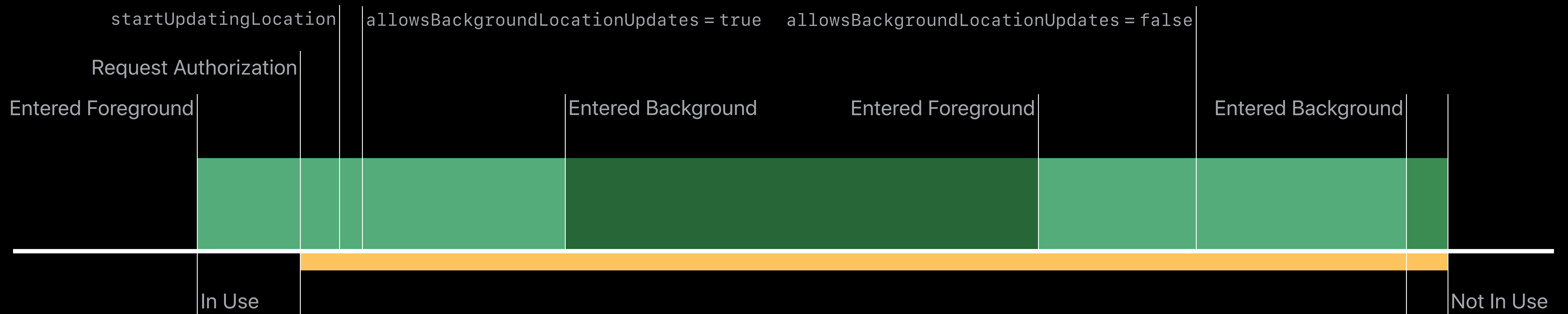
Temporary Authorization Case Study

Foreground



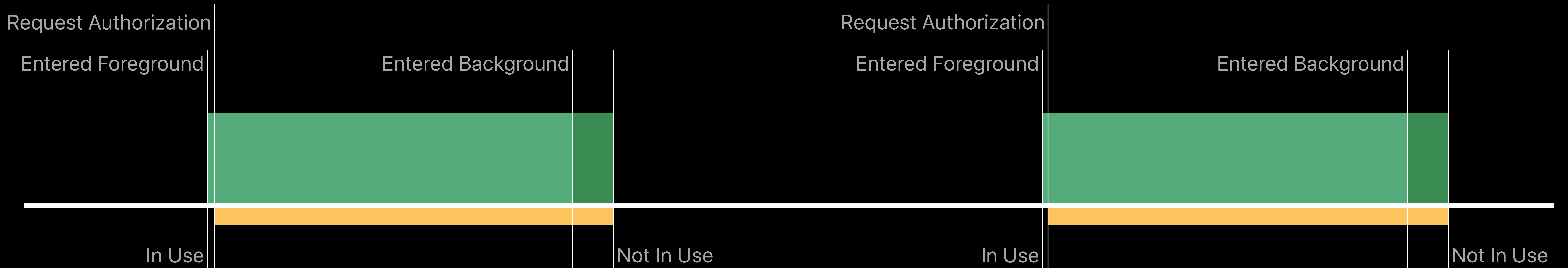
Temporary Authorization Case Study

Continuous Background Updates



Temporary Authorization Case Study

Continuous foreground updates



Always Authorization: Deferred

WhenInUse Authorization: Empowered

Temporary Authorization: New

Ranging: In Context

Beacon Ranging Overview

API Changes

Example

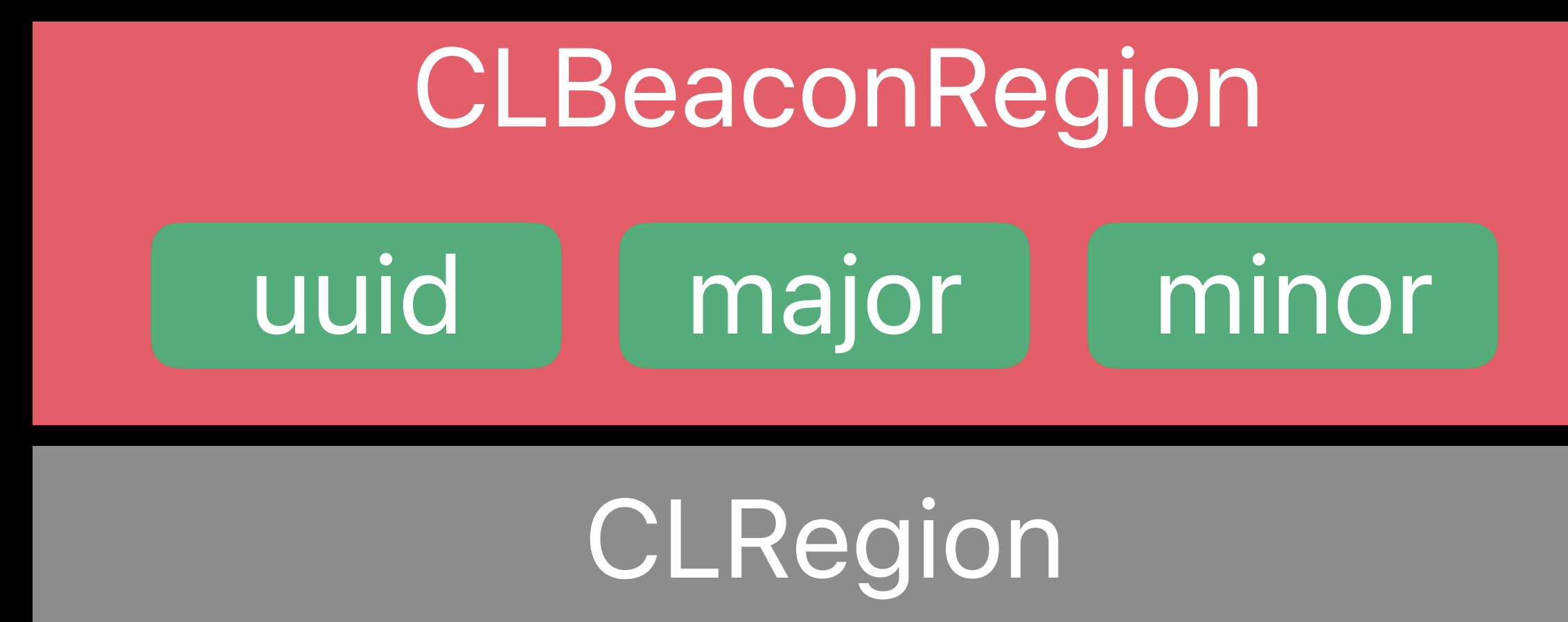
Beacon Ranging

Introduced in iOS 7 as extension of the Region Monitoring API

Region Monitoring to initiate ranging

Region Monitoring no longer requires Always Authorization

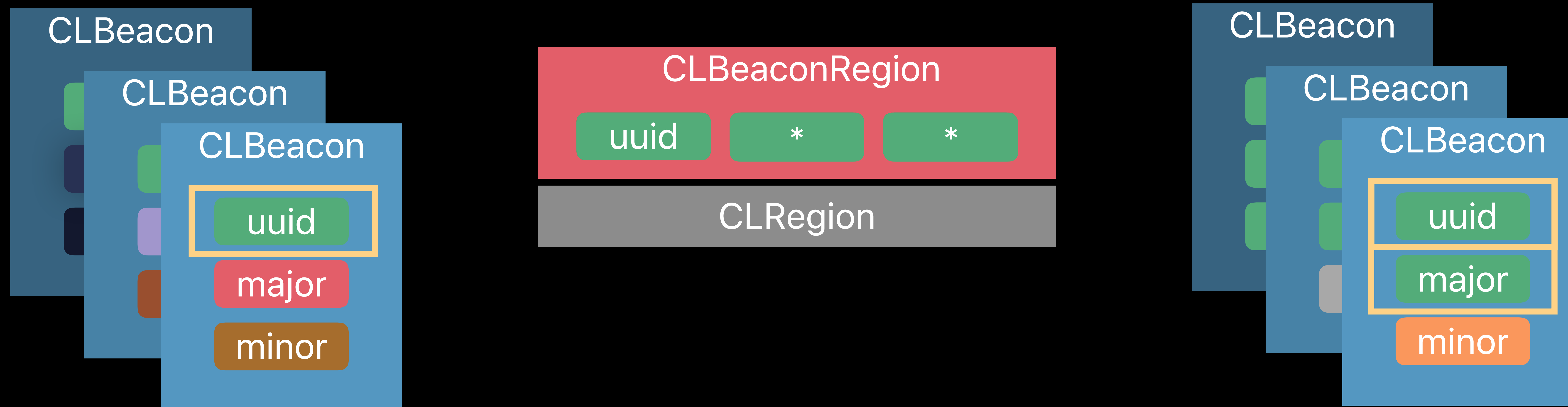
Beacon Ranging API, iOS 12



```
open func startRangingBeacons(in region: CLBeaconRegion)
```

```
open func stopRangingBeacons(in region: CLBeaconRegion)
```

Beacon Ranging API Updates



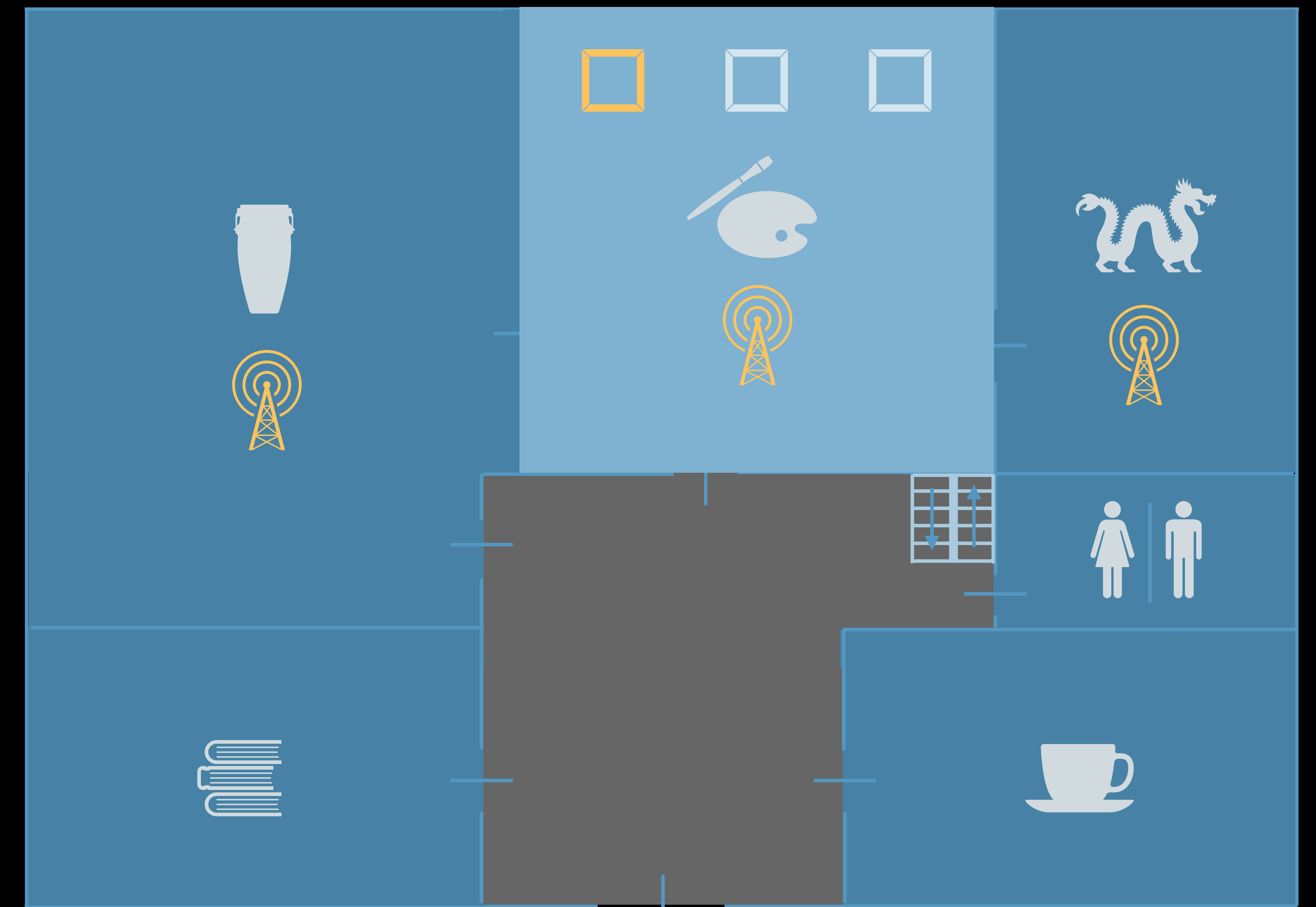
Example: Beacon Ranging in a museum

Install beacons in the exhibition rooms

Determine when a visitor is in an exhibition room

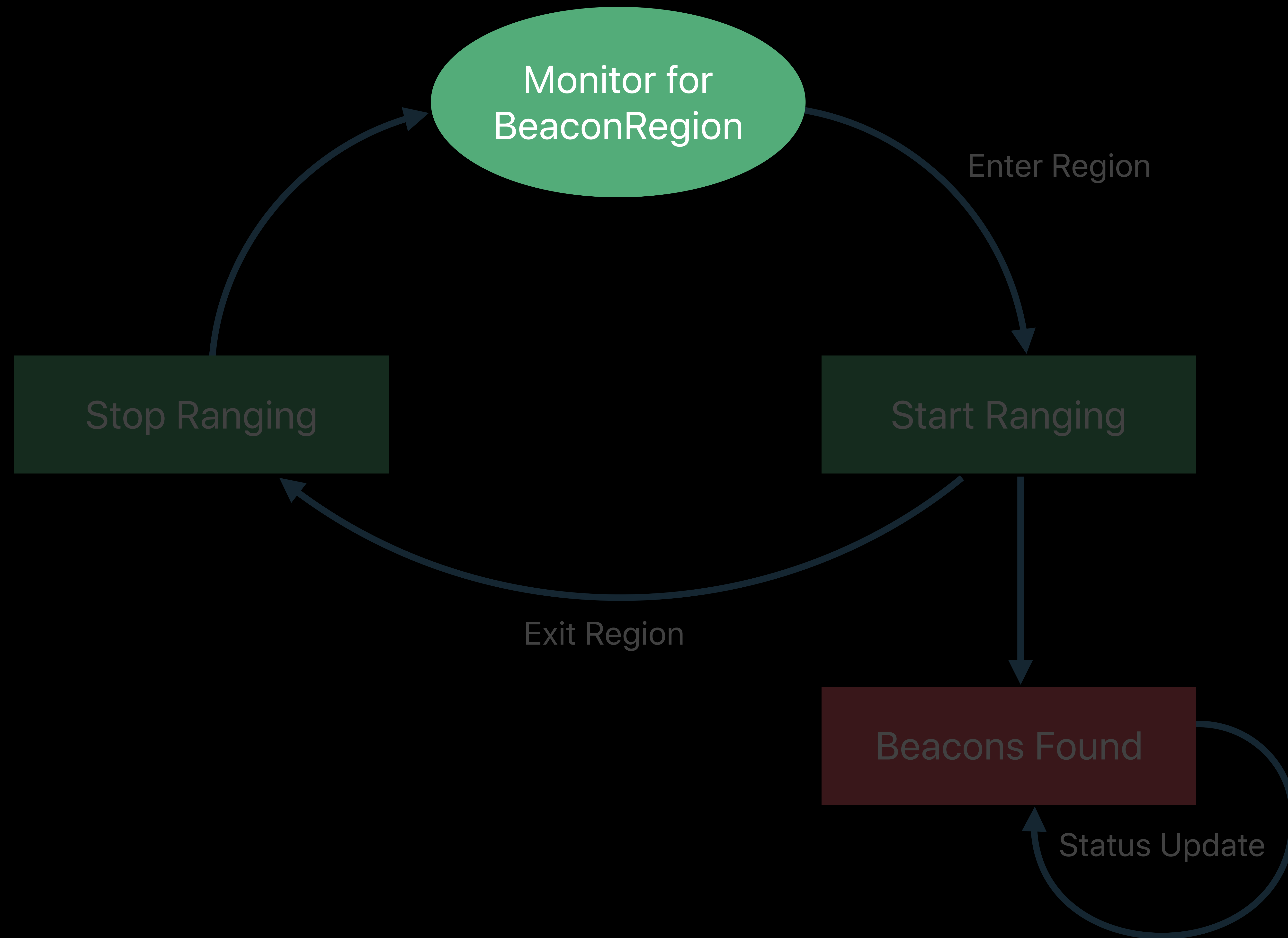
Find which objects are nearby

Location-triggered notification at arrival

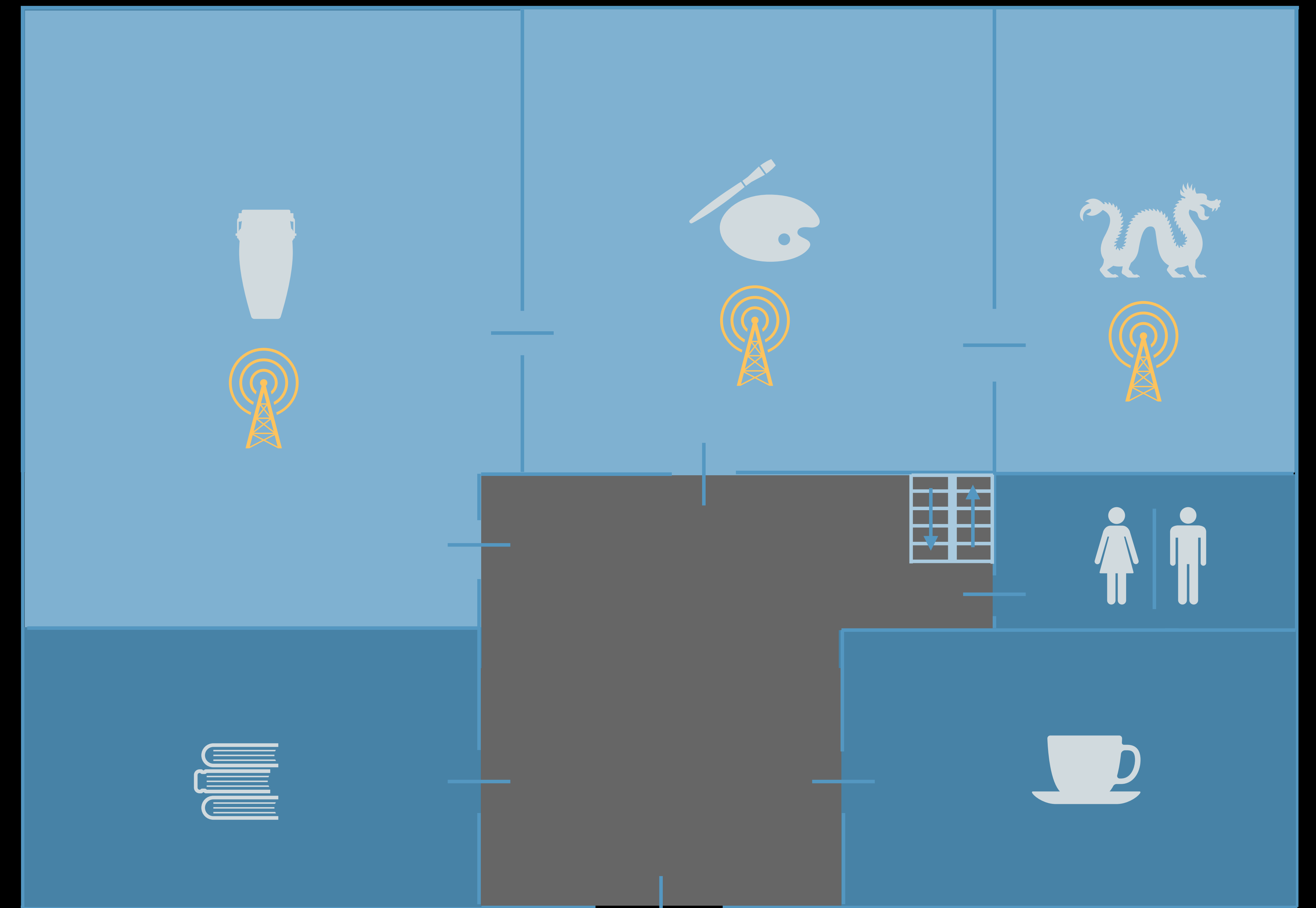
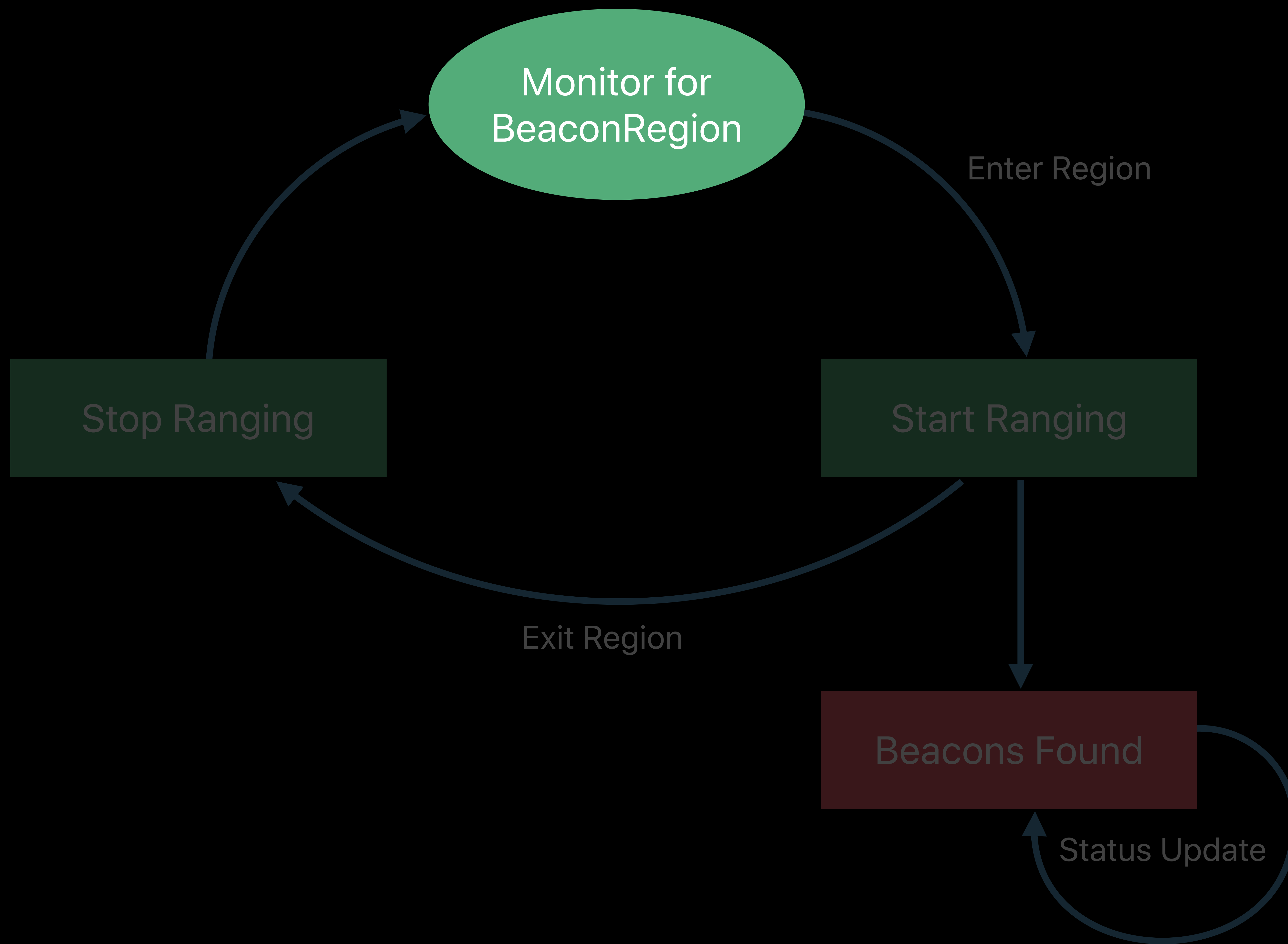


UUID: MyMuseum
Major: PaintersRoom
Minor: SomePainting

When in Use



When in Use



UUID: MyMuseum
Major: PaintersRoom
Minor: SomePainting


```
// Monitor for the beacon-defined region
```

```
self.locationManager.requestWhenInUseAuthorization()
```

```
// Create a new constraint and add it to the dictionary.
```

```
let constraint = CLBeaconIdentityConstraint(uuid: uuid)
```

```
/*
```

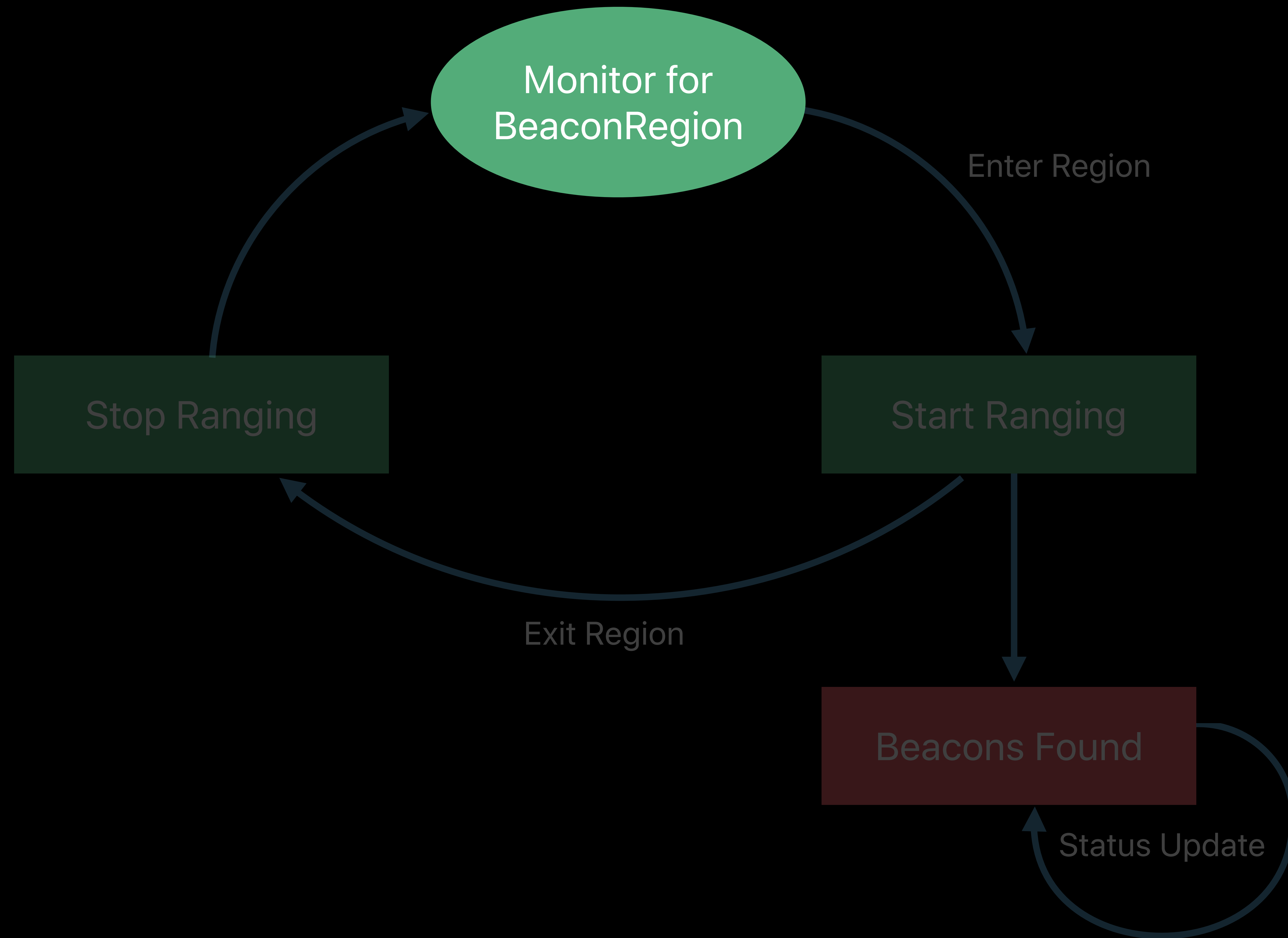
```
By monitoring for the beacon before ranging, the app is more  
energy efficient if the beacon is not immediately observable.
```

```
*/
```

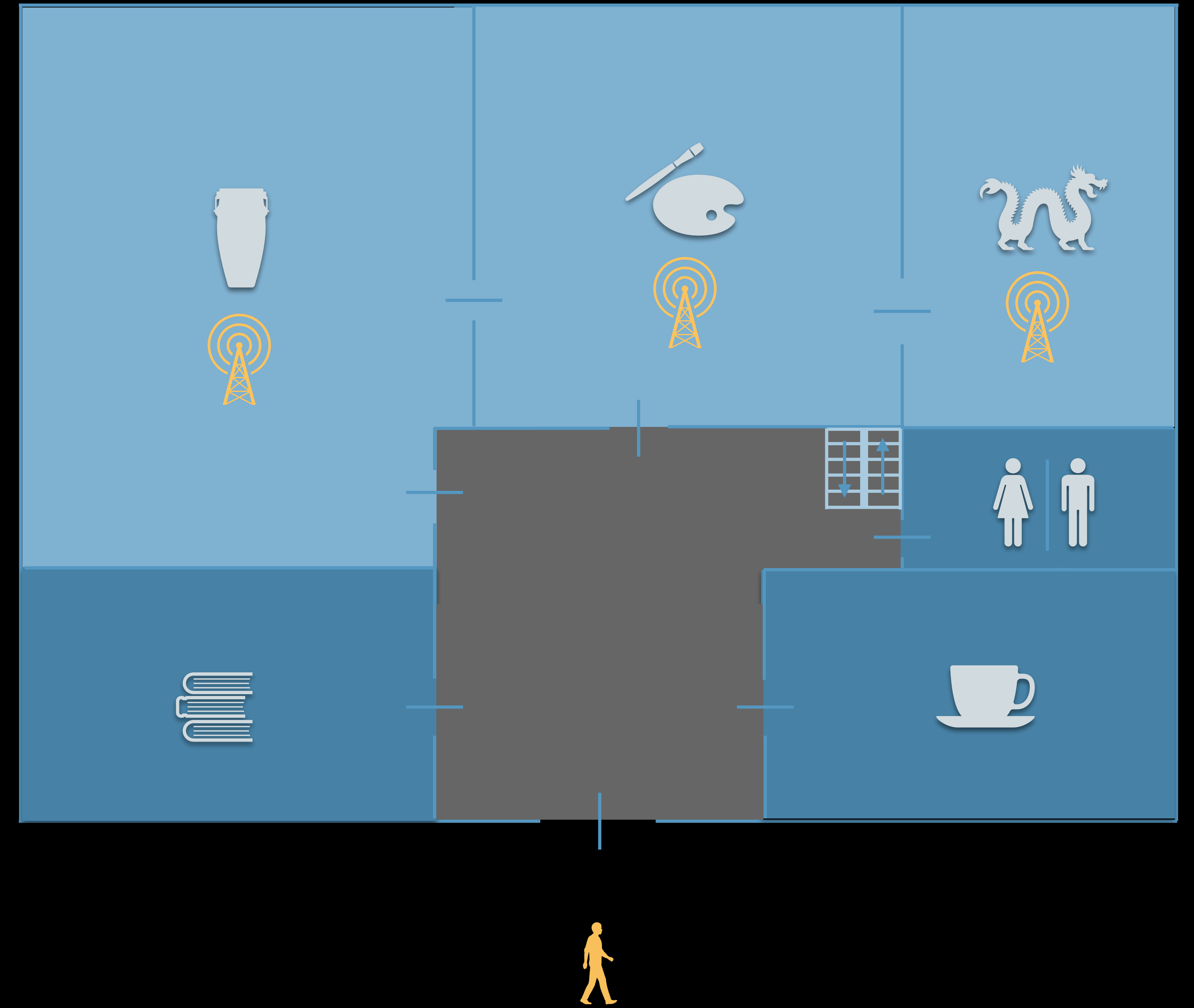
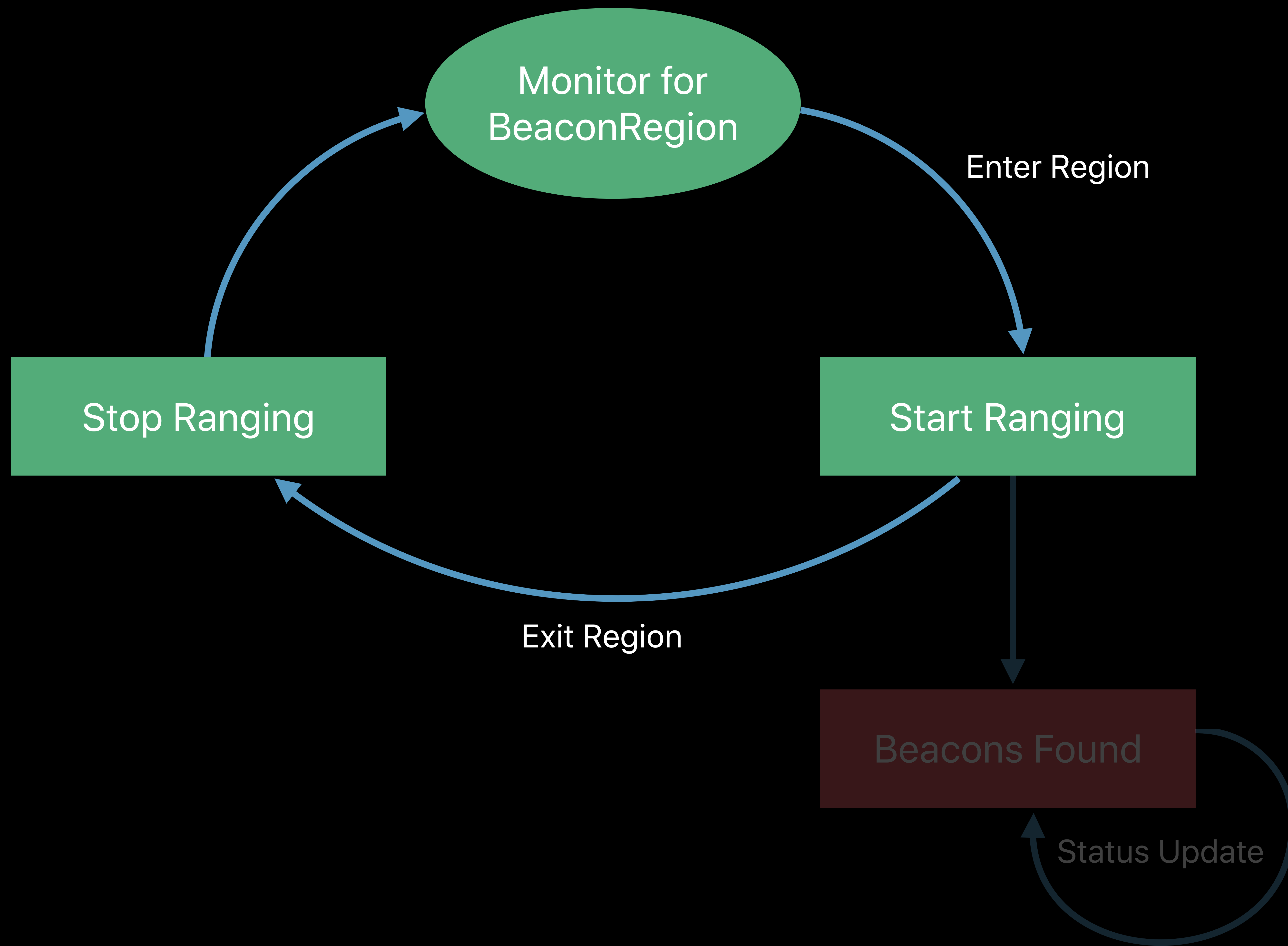
```
let beaconRegion = CLBeaconRegion(beaconIdentityConstraint: constraint,  
                                   identifier: uuid.uuidString)
```

```
self.locationManager.startMonitoring(for: beaconRegion)
```

When in Use



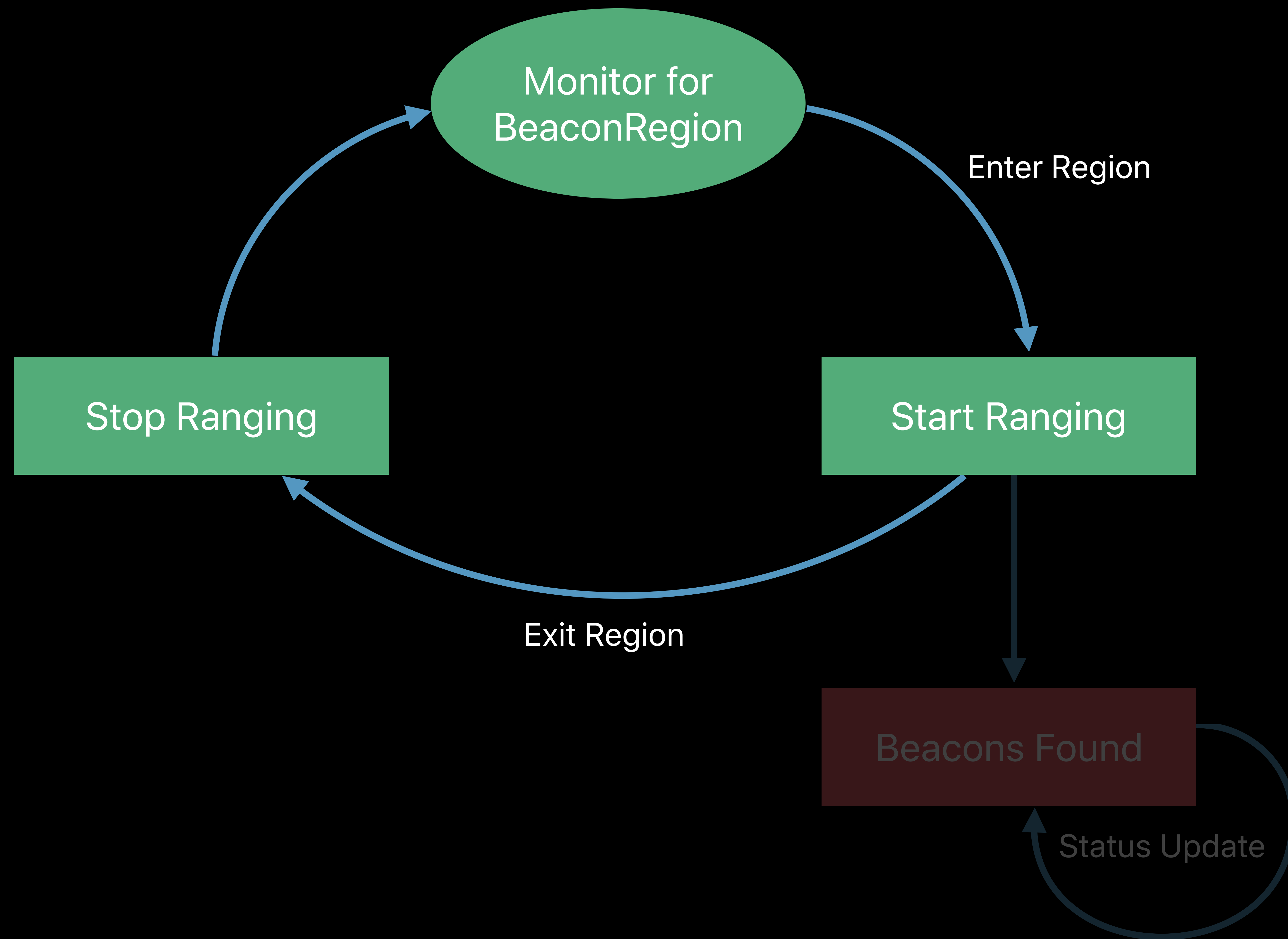
When in Use



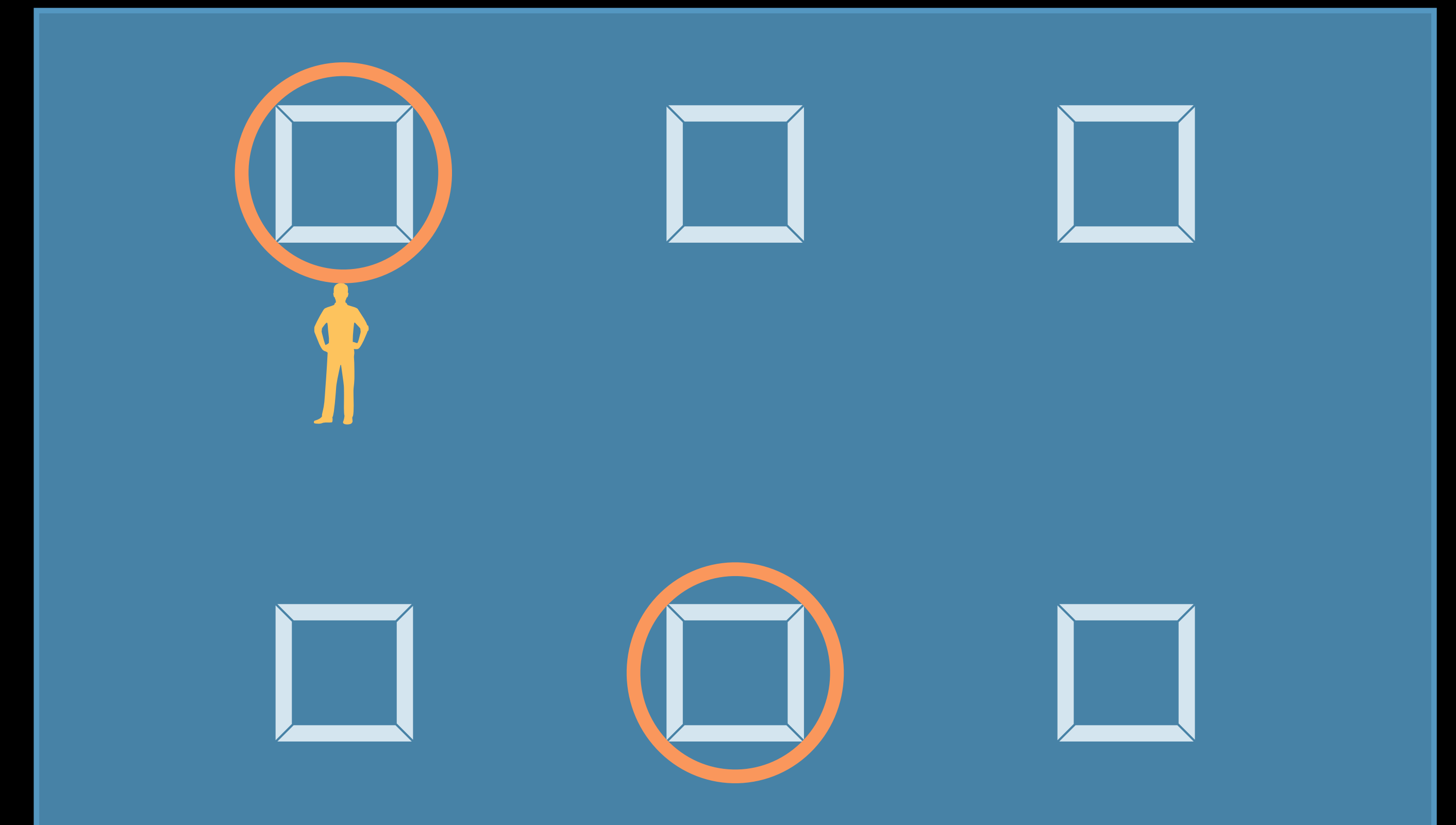
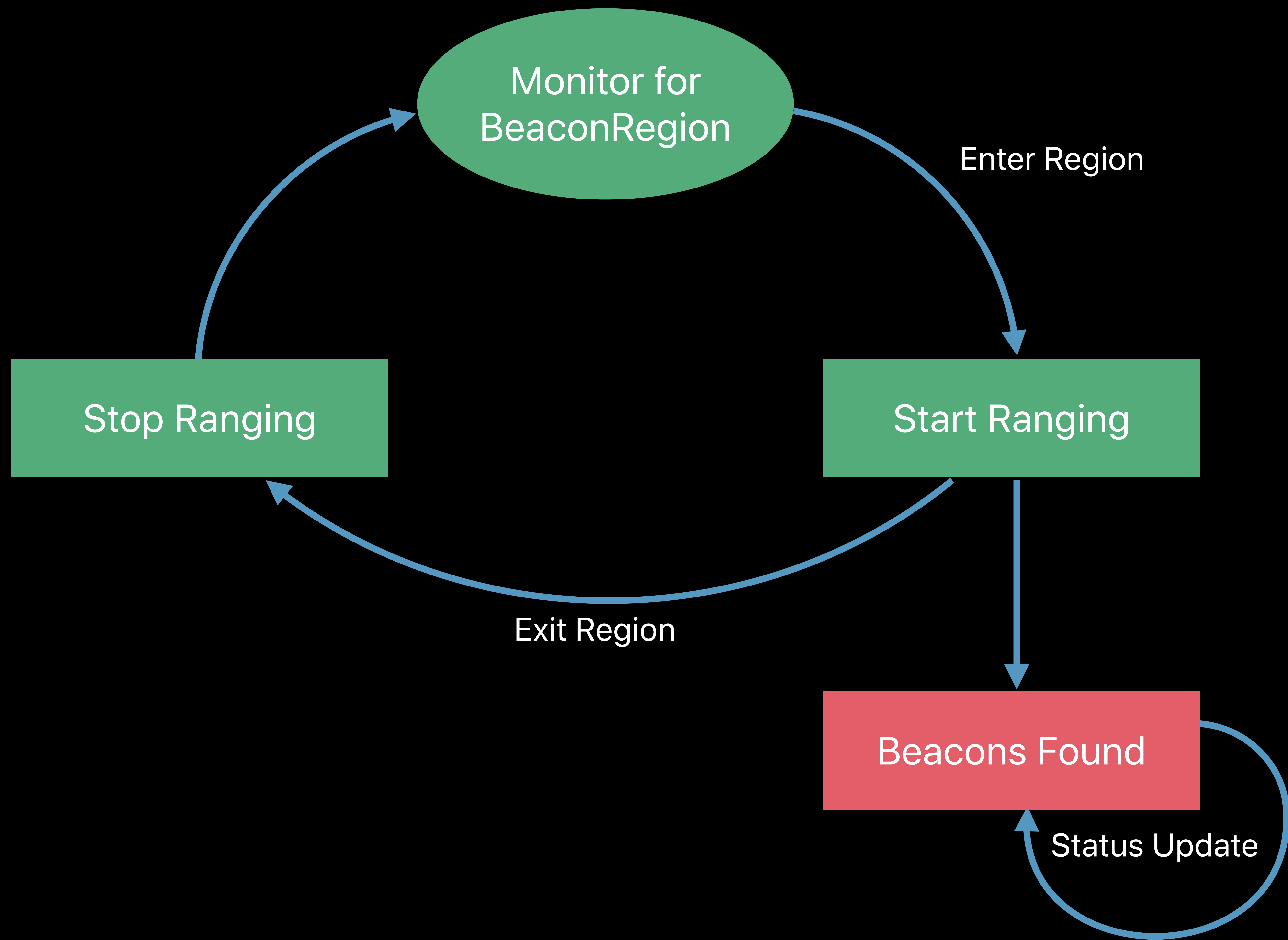
```
// React to entering/exiting the beacon-defined region

func locationManager(_ manager: CLLocationManager,
                    didDetermineState state: CLRegionState,
                    for region: CLRegion)
{
    let beaconRegion = region as? CLBeaconRegion
    if state == .inside {
        // Start ranging when inside a region.
        manager.startRangingBeacons(satisfying: beaconRegion!.beaconIdentityConstraint)
    } else {
        // Stop ranging when not inside a region.
        manager.stopRangingBeacons(satisfying: beaconRegion!.beaconIdentityConstraint)
    }
}
```

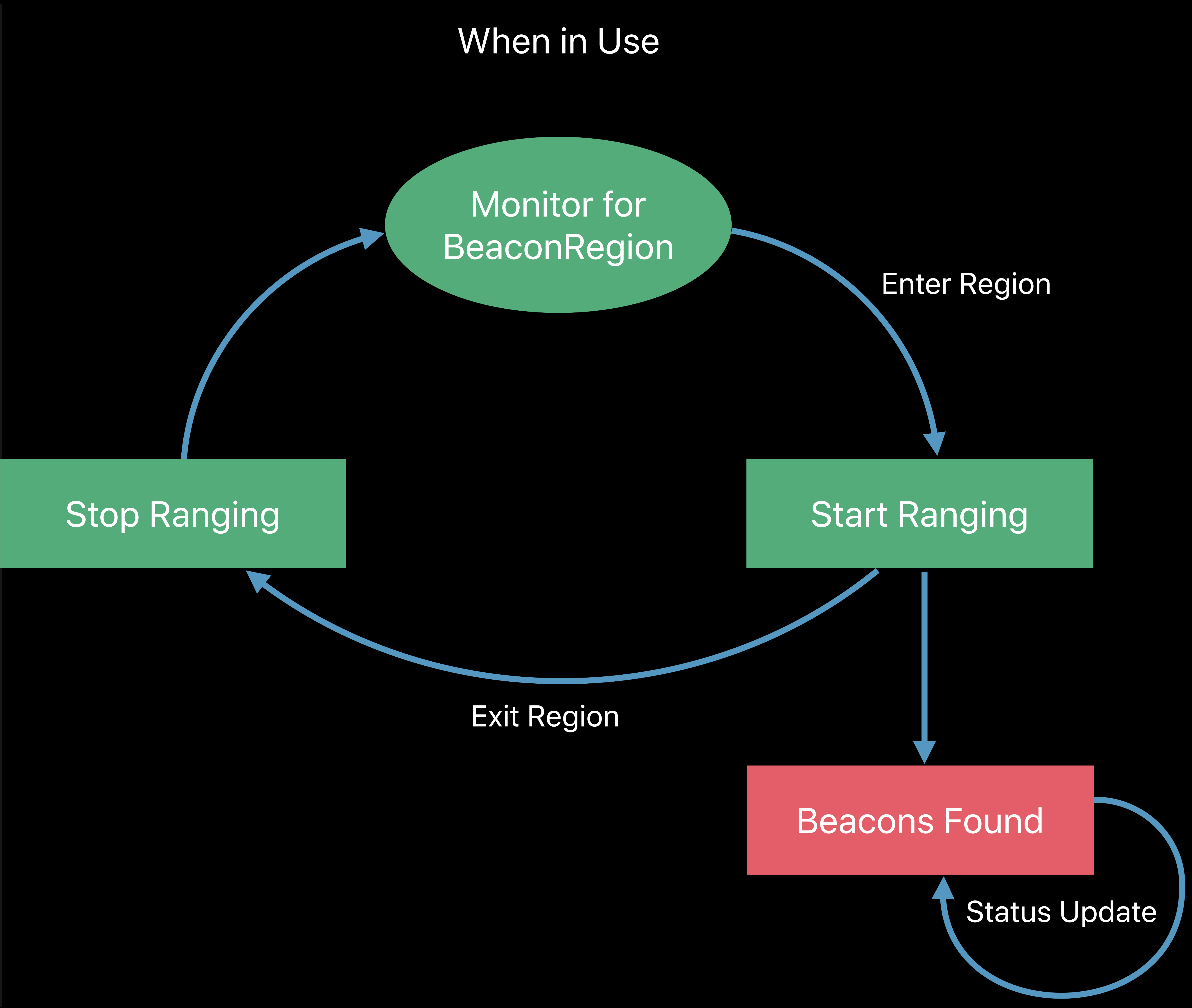
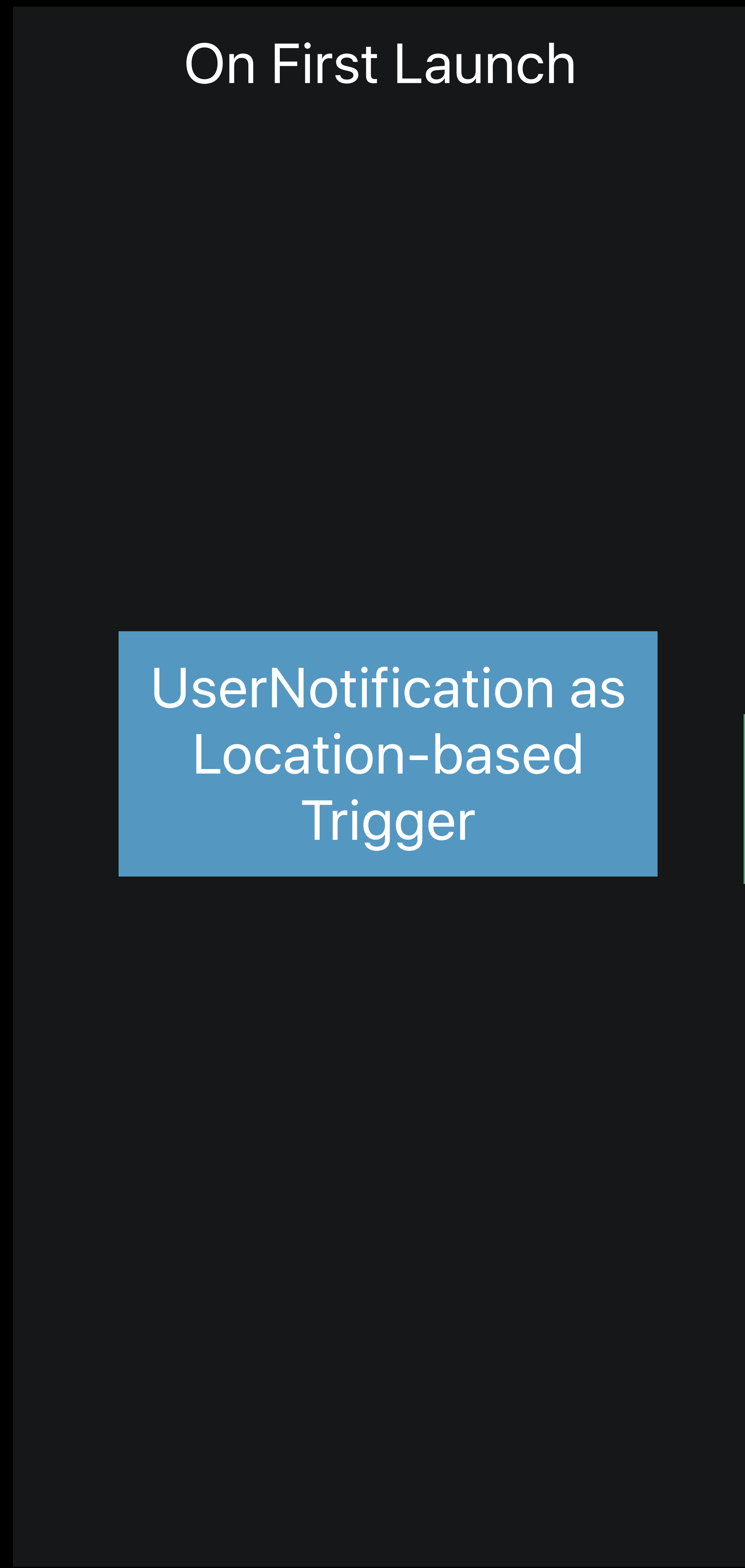
When in Use



When in use



```
// Status update for what beacons are in proximity
func locationManager(_ manager: CLLocationManager,
                    didRange Beacons: [CLBeacon],
                    satisfying beaconConstraint: CLBeaconIdentityConstraint)
{
    /* Beacons are categorized by proximity. */
    for range in [CLProximity.immediate, .near, .far, .unknown] {
        let proximityBeacons = beacons.filter { $0.proximity == range }
        // TODO: Do something with the matching beacons
    }
}
```




```
// Setup a location-triggered user notification
```

```
let center = CLLocationCoordinate2D(latitude: 37.335400, longitude: -122.009201)
```

```
let region = CLCircularRegion(center: center, radius: 2000.0, identifier: "Museum")
```

```
region.notifyOnEntry = true
```

```
region.notifyOnExit = false
```

```
let trigger = UNLocationNotificationTrigger(region: region, repeats: false)
```

Recap

Beacon Ranging

API Updates

Example: Ranging for beacons in a museum

Summary

We have some new authorization behaviors

Test them

Ranging opens a new way to deliver location-aware experiences

More Information

developer.apple.com/wwdc19/705

Core Location Lab

Wednesday, 11:00

Designing for Privacy

Wednesday, 2:00

Core Location Lab

Friday, 1:00

 WWDC19