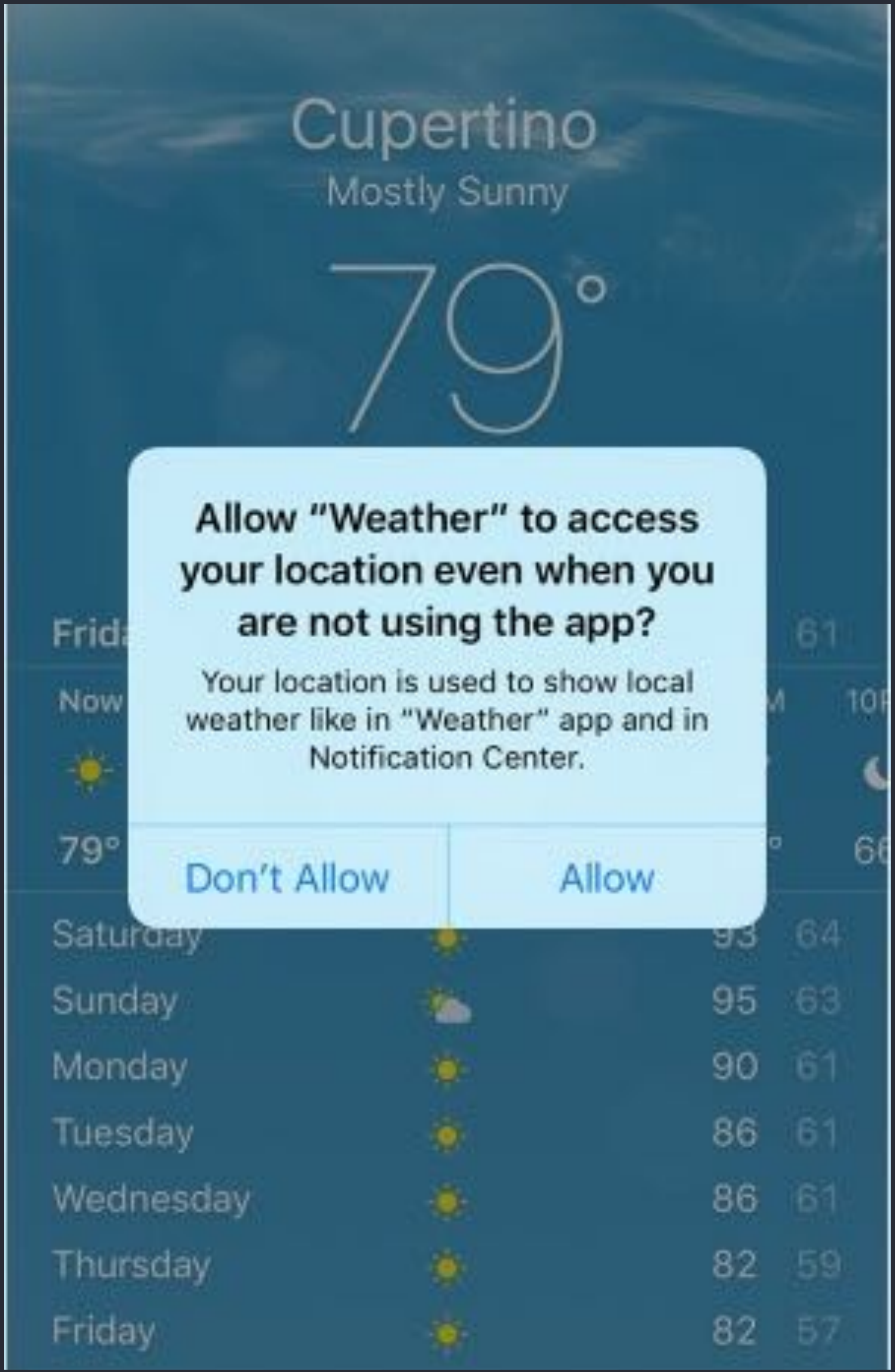
Core Location and background APIs

Gustavo Nascimento
@gusta_nas

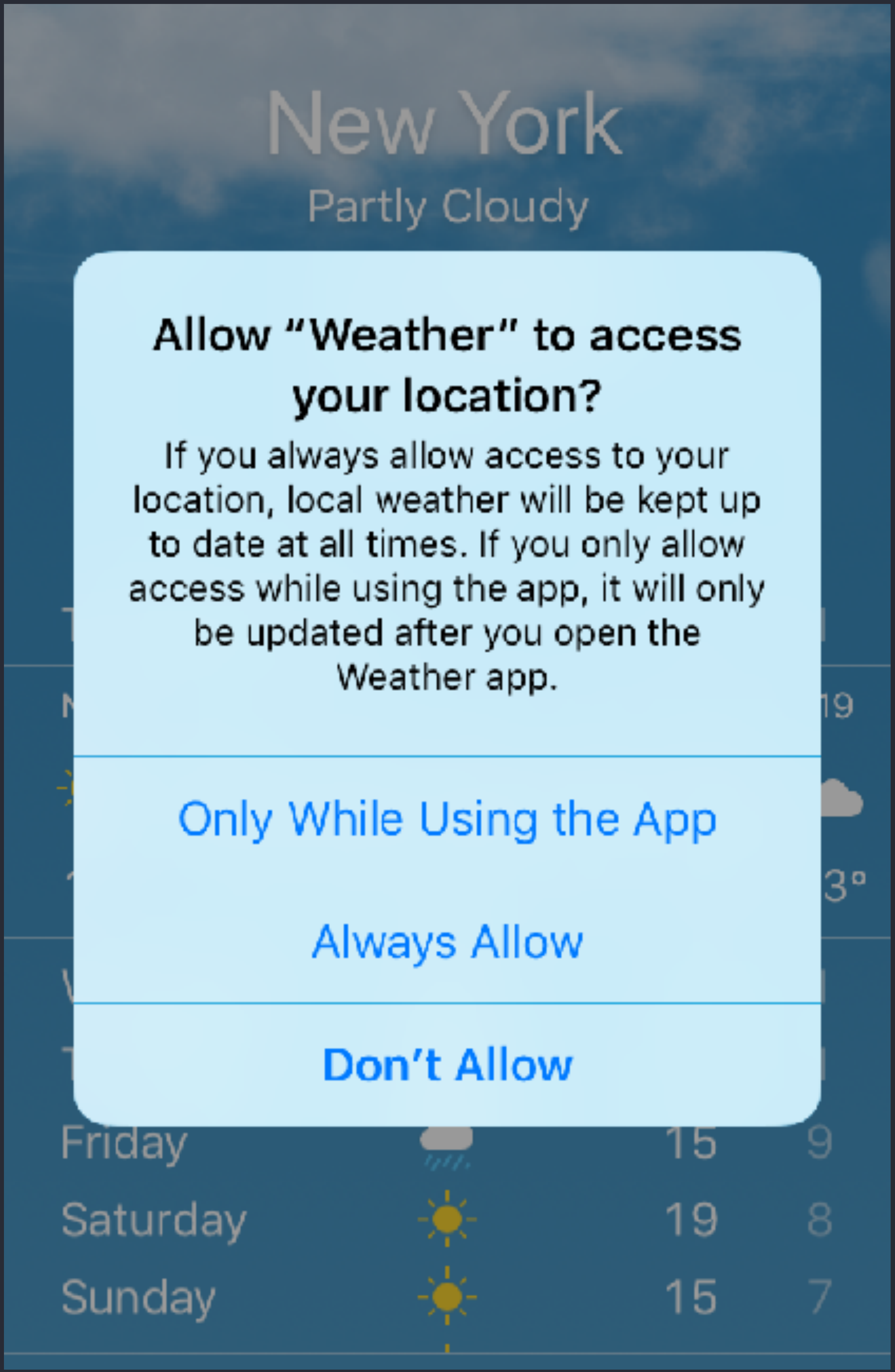
Introduction

- Once an app has moved to the background, it can be killed or suspended anytime.
- Request for extra time using a backgroundTask (up to 3 minutes)
- But, Apple allows some specific apps to run for an extended time in the background when necessary
- Location based apps



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

Your location is used to show local weather like in "Weather" app and in Notification Center.

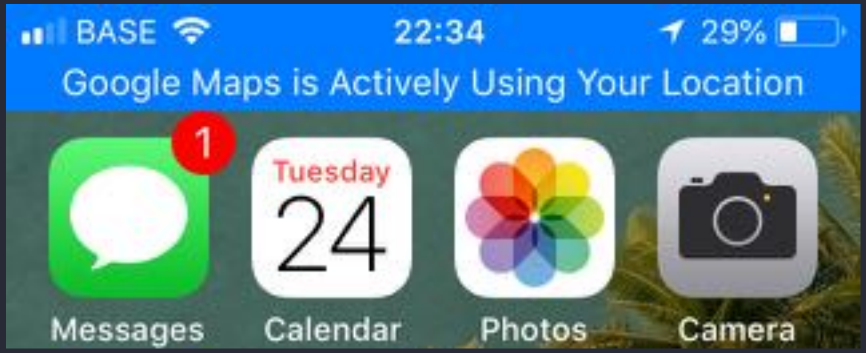


Allow "Weather" to access your location?

If you always allow access to your location, local weather will be kept up to date at all times. If you only allow access while using the app, it will only be updated after you open the Weather app.

A word about Privacy

- Different levels of location access
  - Always
  - When-In-Use
  - Never
- Xcode project correctly configured



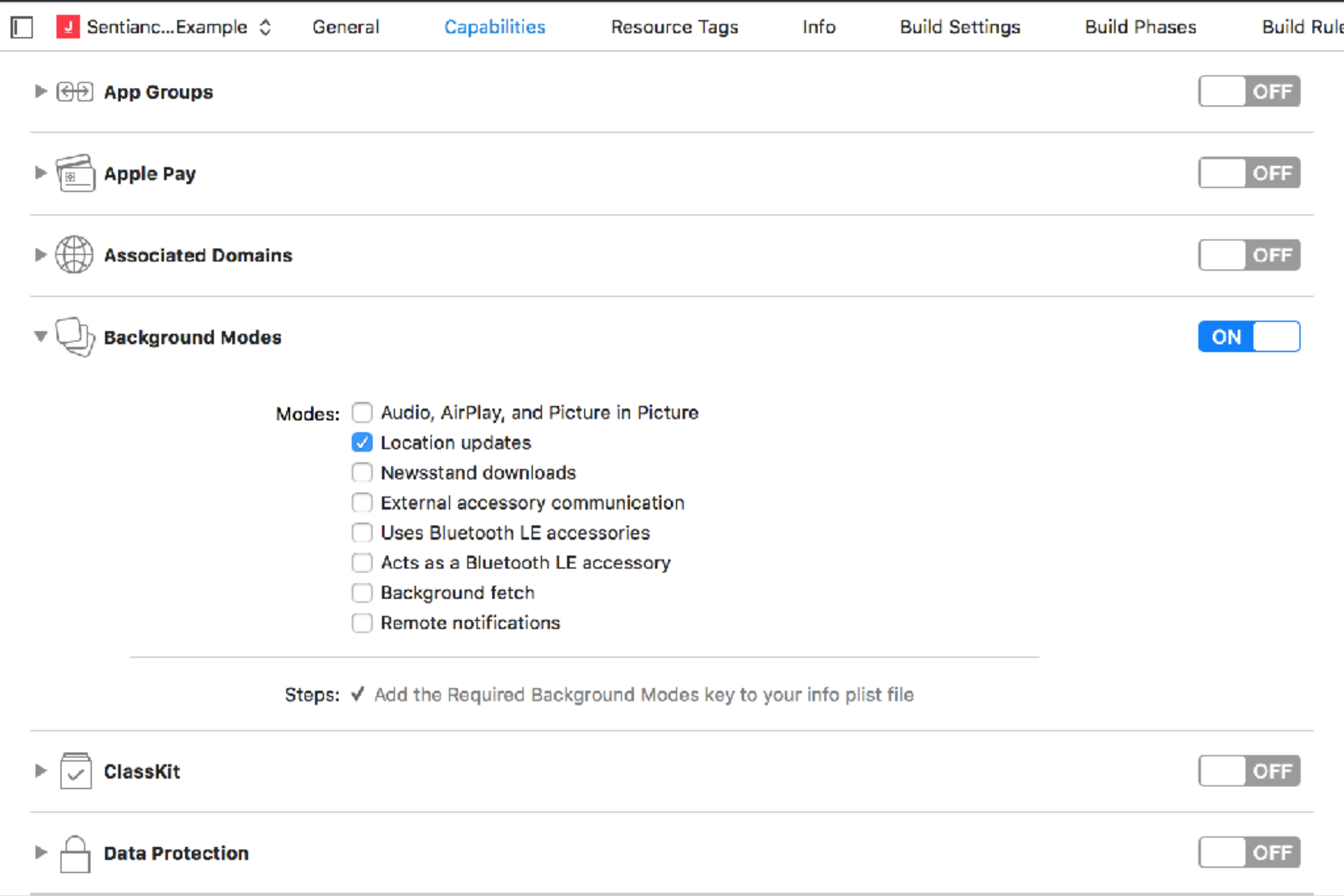
Google Maps is Actively Using Your Location

Info.plist

- NSLocationAlwaysAndWhenInUseUsageDescription
- NSLocationAlwaysUsageDescription
- NSLocationUsageDescription
- NSLocationWhenInUseUsageDescription

Requirements checklist

- Info.plist configured
- Request always authorization
- Xcode capabilities



App Groups OFF

Apple Pay OFF

Associated Domains OFF

Background Modes ON

Modes:
☐ Audio, AirPlay, and Picture in Picture
☑ Location updates
☐ Newsstand downloads
☐ External accessory communication
☐ Uses Bluetooth LE accessories
☐ Acts as a Bluetooth LE accessory
☐ Background fetch
☐ Remote notifications

Steps: ✓ Add the Required Background Modes key to your info plist file

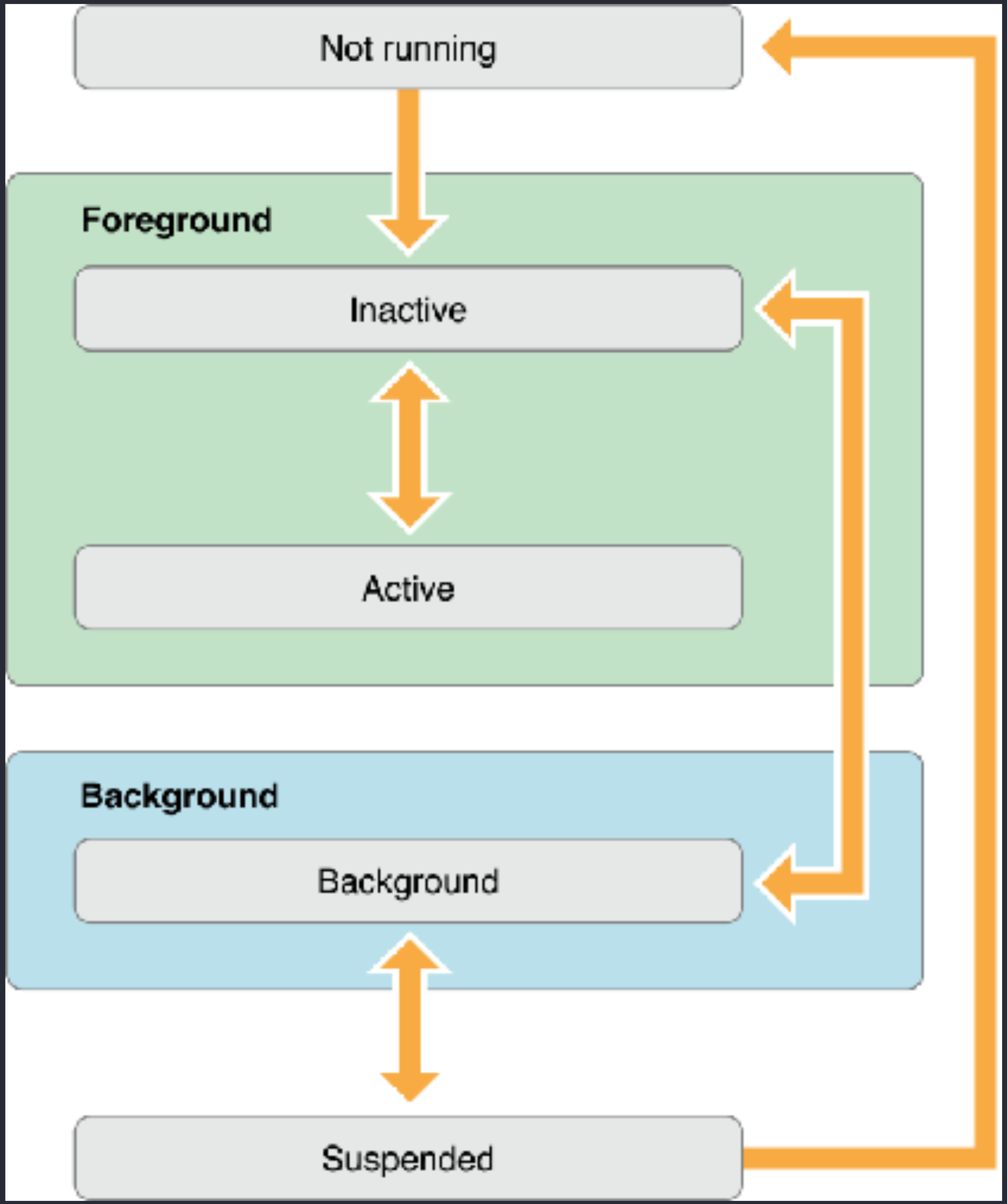
ClassKit OFF

Data Protection OFF

```
import CoreLocation

let locationManager = CLLocationManager()
locationManager.requestAlwaysAuthorization()
locationManager.allowsBackgroundLocationUpdates = true
locationManager.delegate = self
locationManager.startUpdatingLocation()


func locationManager(_ manager: CLLocationManager,  didUpdateLocations locations: [CLLocation]) {
    let lastLocation = locations.last!

    // Do something with the location.
}
```

- We will now receive location updates in the background

The execution state



Background APIs

- Significant-change location monitoring
- Region monitoring
- Visits monitoring

Significant-change location

- Will deliver location update every ~500m
- More battery friendly (relies on Wi-Fi and cellular information)
- But less accurate

```swift
import CoreLocation

let locationManager = CLLocationManager()
locationManager.requestAlwaysAuthorization()
locationManager.allowsBackgroundLocationUpdates = true
locationManager.delegate = self
locationManager.startMonitoringSignificantLocationChanges()
```

Region Monitoring

Entering
Exiting

Region Monitoring

- Determine when the user enters or leaves a geographic region. But, without knowing the exact location of the user
- Coordinates (center) + radius (in meters).
- Max 20 regions at the same time
- Can get notified of an exit event only if we entered the region before!

```
let region = CLCircularRegion(center: CLLocationCoordinate2D(latitude: 50, longitude: 50),
radius: 100, identifier: “myRegion")

region.notifyOnExit = false
region.notifyOnEntry = true

locationManager.startMonitoring(for: region)

func locationManager(_ manager: CLLocationManager, didEnterRegion region: CLRegion) {
    if let region = region as? CLCircularRegion {
        //
    }
}

func locationManager(_ manager: CLLocationManager, didExitRegion region: CLRegion) {
    if let region = region as? CLCircularRegion {
        //
    }
}
```

Visits

- Very power-efficient but less frequent than other services
- Will inform the app every time we have left (or have entered) a frequently visited place.

```
locationManager.startMonitoringVisits()

func locationManager(_ manager: CLLocationManager, didVisit visit: CLVisit) {
    // Do something with the visit.
    print(visit.departureDate)
    print(visit.arrivalDate)
}
```

- Will not always contain both the departureDate and the arrivalDate!

But what if my app is not running?

How will my app be launched?

- In AppDelegate.swift:

```
func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions:
[UIApplicationLaunchOptionsKey: Any]?) -> Bool {

    if (launchOptions?[UIApplicationLaunchOptionsKey.location] != nil) {
        // App was launched by a location event
    }

    return true
}
```

How will my app be launched?

- Create a new CLLocationManager object, configure it with a delegate, and start location services again to receive the update.

- Your didFinishLaunchingWithOptions method shouldn’t take more than 10s to return.

- No UI related code is called!

Battery

How to reduce battery usage?

```
locationManager.desiredAccuracy = kCLLocationAccuracyThreeKilometers

//Use the highest possible accuracy and combine it with additional sensor data.
let kCLLocationAccuracyBestForNavigation: CLLocationAccuracy

//Use the highest level of accuracy.
let kCLLocationAccuracyBest: CLLocationAccuracy

//Accurate to within ten meters of the desired target.
let kCLLocationAccuracyNearestTenMeters: CLLocationAccuracy

//Accurate to within one hundred meters.
let kCLLocationAccuracyHundredMeters: CLLocationAccuracy

//Accurate to the nearest kilometer.
let kCLLocationAccuracyKilometer: CLLocationAccuracy

//Accurate to the nearest three kilometers.
let kCLLocationAccuracyThreeKilometers: CLLocationAccuracy
```

How to reduce battery usage?

- Let the OS pause the updates when it thinks we are not moving anymore.

```
locationManager.pausesLocationUpdatesAutomatically = true
locationManager.activityType = .automotiveNavigation

//The location manager is being used for an unknown activity.
case other

//The location manager is being used specifically during vehicular navigation to track
location changes to the automobile.
case automotiveNavigation

//The location manager is being used to track fitness activities such as walking,
running, cycling, and so on.
case fitness

//The location manager is being used to track movements for other types of vehicular
navigation that are not automobile related.
case otherNavigation
```

How to reduce battery usage?

- Request a quick update and stop the location service right after.

```
locationManager.requestLocation()
```

- Defer location updates

```
locationManager.allowDeferredLocationUpdates(untilTraveled: CLLocationDistance,
timeout: TimeInterval)
```

- Or, only rely on regions/visits/significant-change location services

Challenges

- App can still be suspended
- Location can be cached
- Accuracy not reliable
- Location not updating when you’re not moving
- Network/Geo conditions
- iOS updates
- Regions/Visits/SCL delayed
- Visits timestamps not reliable

Thank you!