iOS Development Accelerator Week 5 Day 4

- Local Notifications
- IBeacons

Notifications

- “Local and push notifications are ways for an application that isn’t running in the foreground to let its users know it has information for them”
- Local and push look and sound the same.
- Can be displayed as an alert message and/or badge icon.
- Can play a sound.
- Not related to NSNotificationCenter!

Push vs Local

- Local notifications are scheduled by an app and delivered on the same app. Everything is done locally.
- Push notifications are sent by your server to the Apple Notification service, which pushes it to the device(s).
- While they appear the exact same to the user, they appear different to your app.
- If your app is in the foreground, you will receive either application:didRecieveRemoteNotification: or application:didRecieveLocalNotification: in the app delegate. If your app is not in the foreground or not running, your app will launch , and then you need to check the launch dictionary.

Local notifications

- Suited for time based or location based behaviors.
- Local notifications are instances of UILocalNotification
- 3 Properties:
  - Scheduled Time: Known as the fire date. Can set the time zone as well. Can also request it be rescheduled at regular intervals.
  - Notification Type: The alert message, the title of action button, the icon badge number, and a sound to play.
  - Custom Data: dictionary of custom data
- Each app limited to 64 scheduled local notifications.

Local notifications work flow

1. Create an instance of UILocalNotification
2. Set the fireData property.
3. Set the alertBody (message) property, alertAction property(title of button or slider), applicationIconBadgeNumber property, and soundName property.
4. Optionally set any custom data you want with userInfo property
5. Schedule the delivery by calling scheduleLocalNotification: on UIApplication. Or you can fire it immediately by calling presentLocalNotificationNow:

- You can cancel local notifications with cancelLocalNotifcation: or cancel all with cancelAllLocalNotifications:

Reacting to a Notification when your app is not in the foreground.

1. The system presents the notification, displaying the alert, badge, and/or playing the sound.
2. As a result, the user taps the action button of the alert, or taps the applications icon.
3. If the user tapped the action button, the app is launched and the app calls its delegate’s application:DidFinishLaunchingWithOptions: method. It passes in the notification payload in the info dictionary.

Reacting to a Notification when your app is in the foreground.

1. The application calls its delegate application:didReceiveRemoteNotification: method or application:didReceiveLocalNotification method and passes in the notification payload.

iBeacons

John Clem
iOS Instructor | CodeFellows

Workshop Demo Code

https://github.com/johnnyclem/iBeacons-Workshop

John Clem
iOS Instructor | CodeFellows

What we’ll cover

What iBeacons is

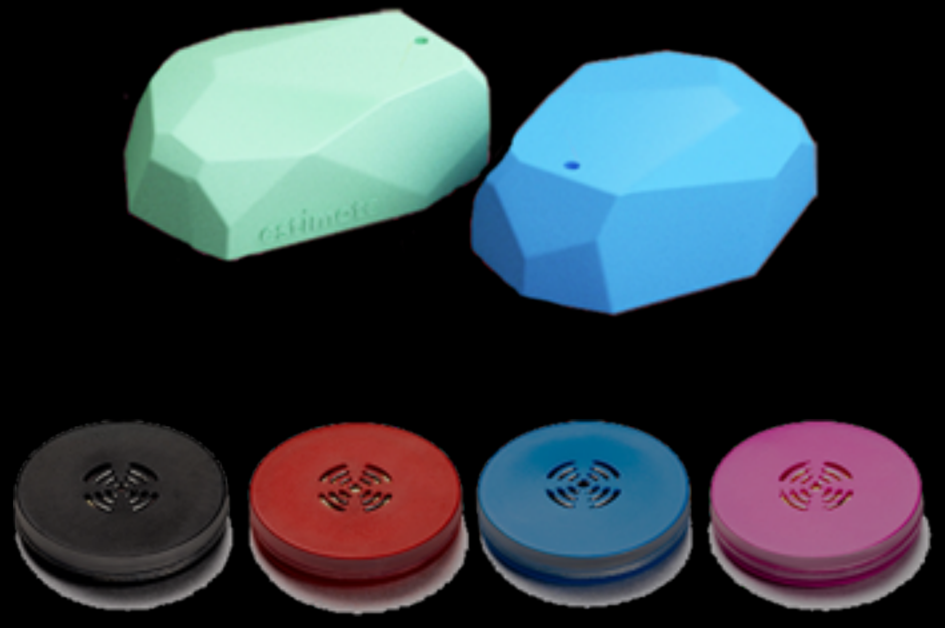
What iBeacons isn’t

Example uses of iBeacons

Current real-world usage of iBeacons

Potential uses for iBeacons

iBeacons Devices



stand-alone device
estimote
stickNfind

iPhone (4s or later)
iPad (3rd gen or later)
iPad mini
iPod Touch (5th gen or later)

Mac/Win/Linux
with Bluetooth LE

Android w/BLE
Windows Phone w/BLE
XYZ Platform w/BLE

iBeacons is not

Indoor GPS

Apple-Only Technology

A specific device

A specific piece of software

iBeacons is

Distance ranging and region monitoring

An open-source specification

Compatible with any Bluetooth LE device

Available on nearly every platform

What can you do with iBeacons

Beacon ranging / broadcasting

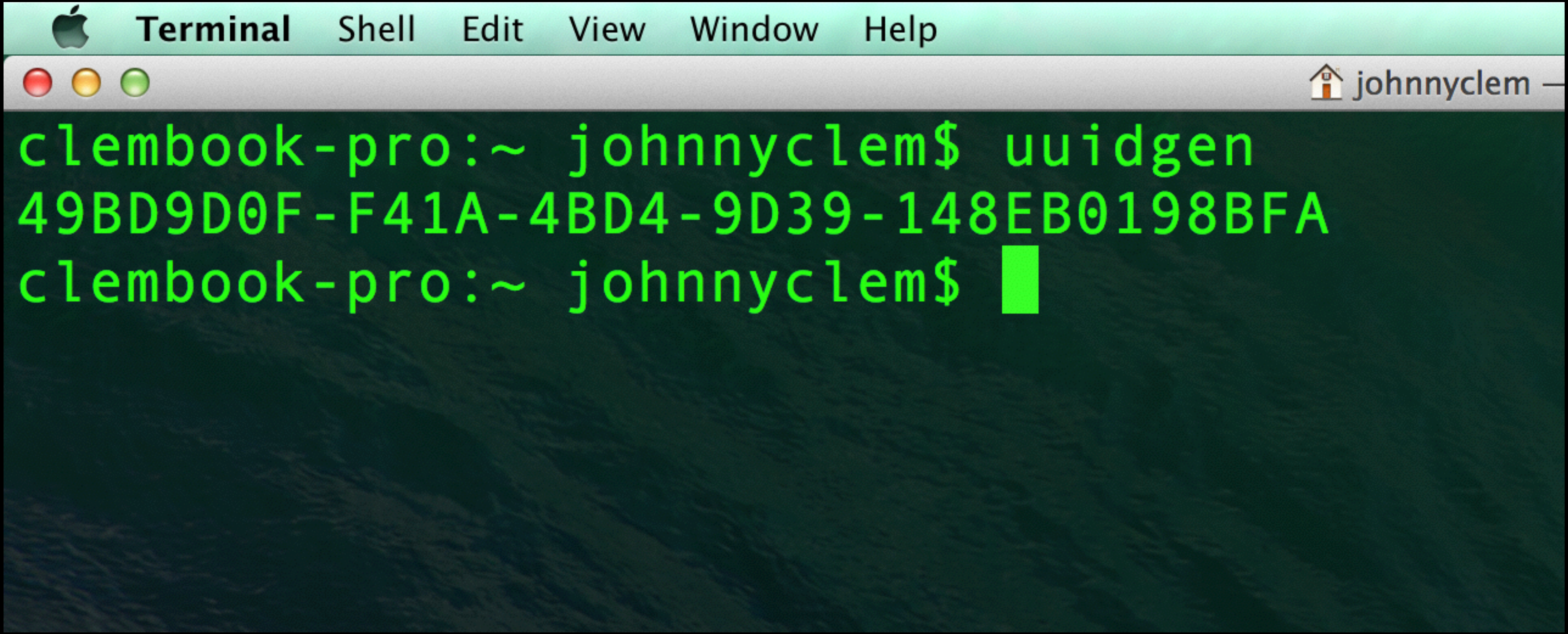
Region monitoring

Proximity-Based Positioning

Store and Retrieve a small amount of data

Be The Beacon

Generate a UUID for your app



```
clembook-pro:~ johnnyclem$ uuidgen
49BD9D0F-F41A-4BD4-9D39-148EB0198BFA
clembook-pro:~ johnnyclem$ 
```

Be The Beacon

Use your UUID to create an NSUUID

```
let myUUID = NSUUID(UUIDString: “5E145790-AC19-463A-A7D7-5EF29CB2A571")
```

Create a unique region identifier

```
let myIdentifier = "com.codefellows.beacons.the_east_room"
```

Be The Beacon

Use your NSUUID to create a CLBeaconRegion

```
let myUUID = NSUUID(UUIDString: “5E145790-AC19-463A-A7D7-5EF29CB2A571”)

let myIdentifier = "com.codefellows.beacons.the_east_room"

var region = CLBeaconRegion(proximityUUID: myUUID, identifier: myIdentifier)
```

Be The Beacon

Create your beacon peripheral data (typically passing nil for the measured power)

```
let beaconData = region.peripheralDataWithMeasuredPower(nil)
```

Start broadcasting your iBeacon using CBPeripheralManager

```
let peripheralManager = CBPeripheralManager(delegate: self, queue: nil)
```

Be The Beacon

DEMO

Beacon Ranging

Range all nearby beacons

Range all beacons with a given UUID

Range beacons with a given UUID, Major (integer) & Minor (integer)

Region Monitoring

Inside/Outside the region

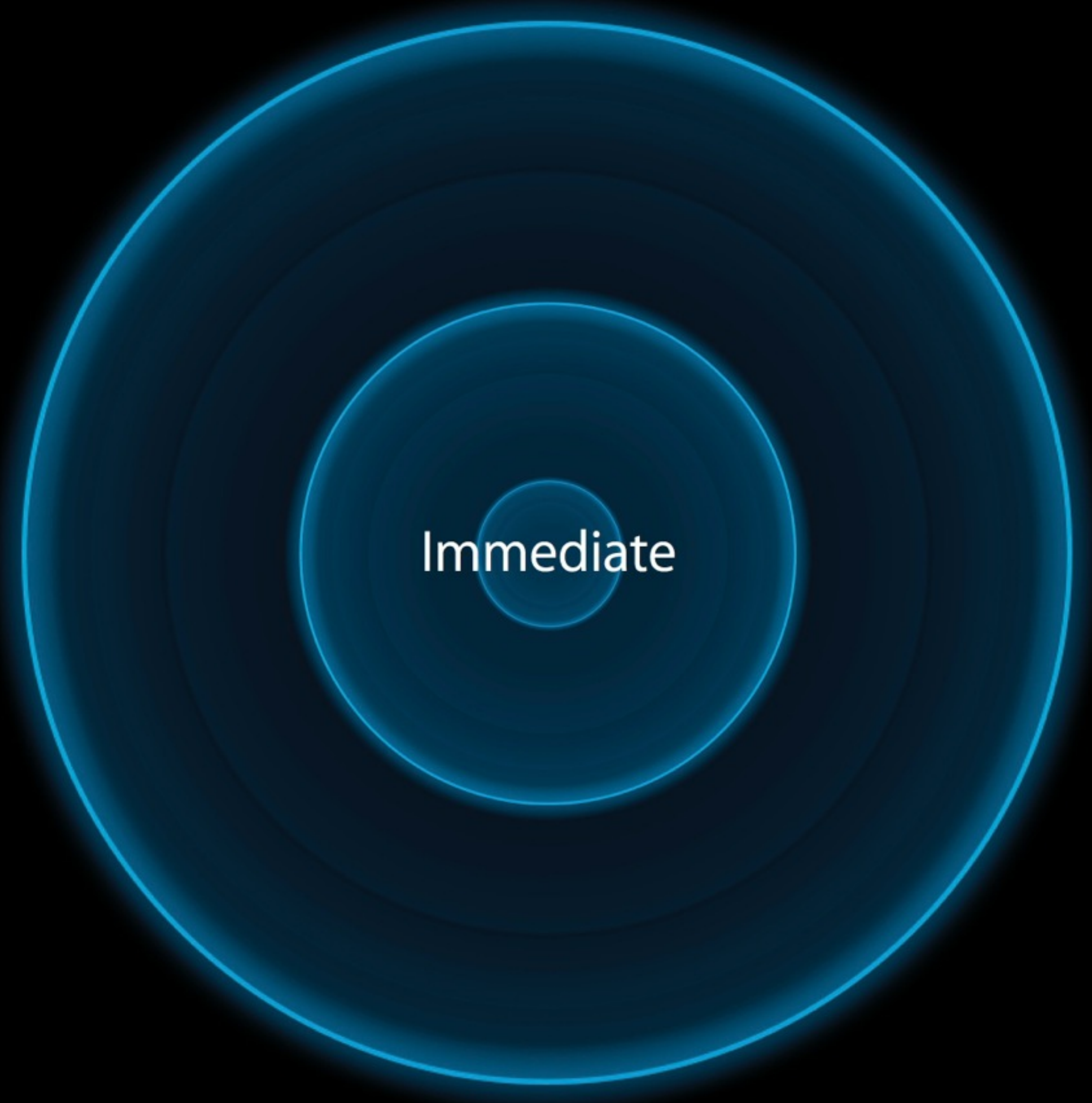
Notify on entry/exit

Notify entry display

iBeacons

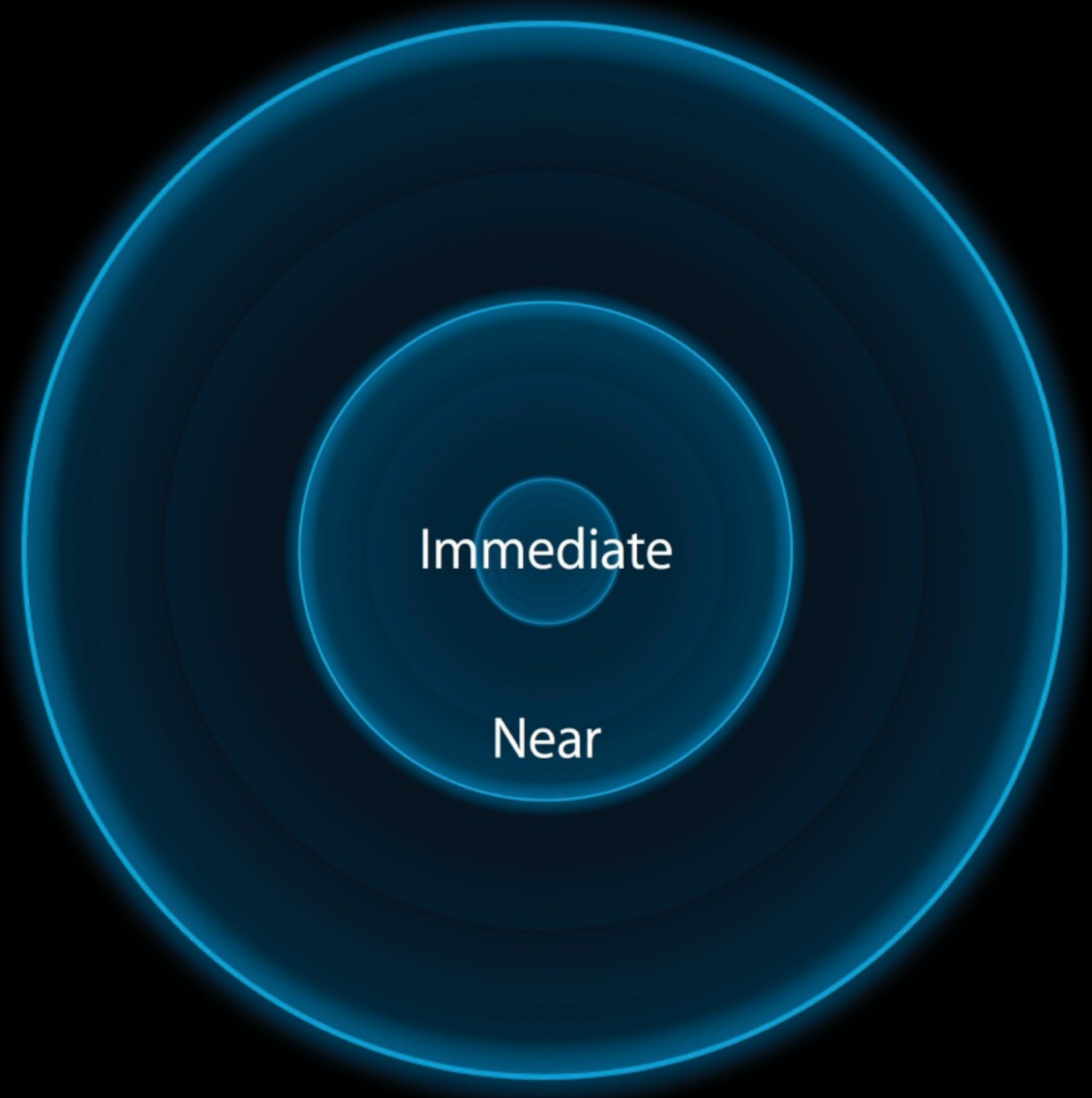
Ranging

iBeacons
Ranging

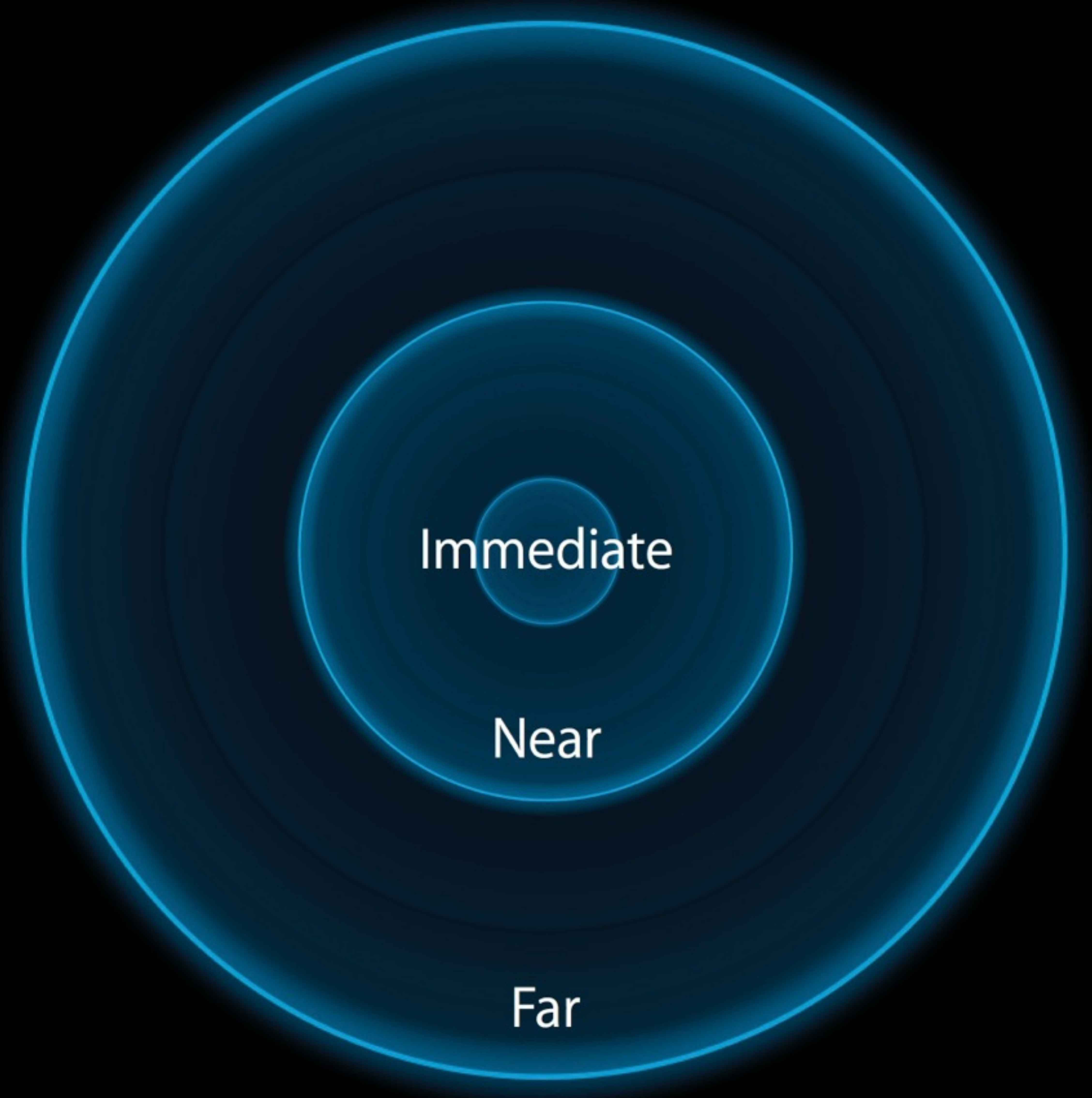


iBeacons

Ranging

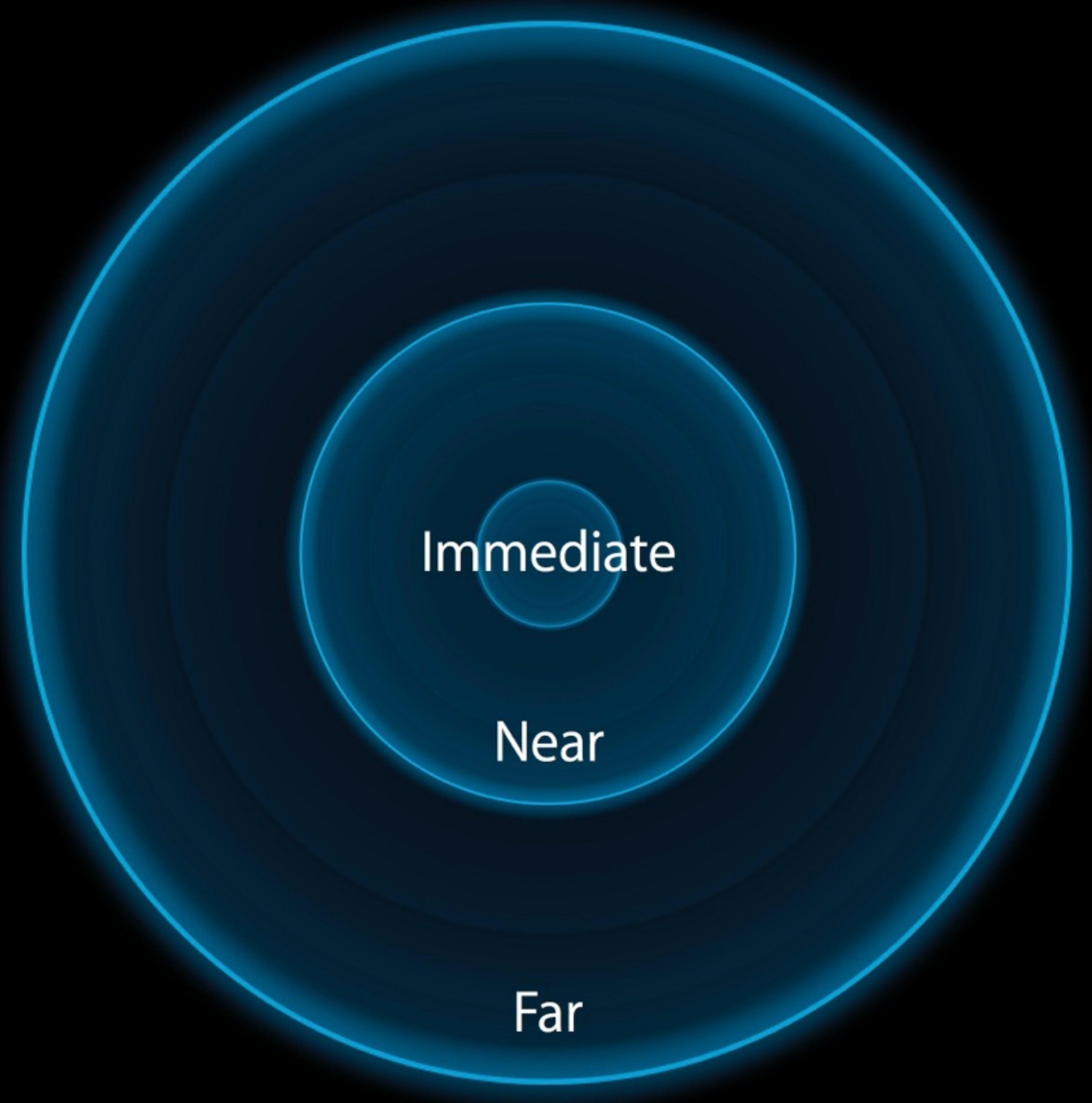


iBeacons
Ranging



iBeacons
Ranging

Unknown



Art Gallery

DEMO

Store and Retrieve Data

UUID / Major / Minor

RSSI / Accuracy / Proximity

Ambient Air Temperature (estimote, stickNfind, etc.)

Choose your own sensor (Arduino, Beaglebone, Raspberry Pi)

Proximity-Based Positioning

RSSI - Received Signal Strength Indicator

Proximity / Accuracy

Immediate - Near - Far - Unknown

Power levels vary from device to device

iBeacons Jukebox

DEMO