Bluetooth

Josh Sy, Marshall Sprigg, Yucheng Lin

What is Bluetooth?

- A universal wireless technology standard used to transmit data over short distances in an ad-hoc network
- Short range (~10 m)
- Low power consumption
- 1 Mbit/s data rate

What is Bluetooth used for?

- Connecting devices
  - Hands free headsets
  - Wireless headphones
  - Wireless speakers
- Data transmission
  - Airdrop
  - Wireless hotspots
- Location tracking
  - Beacons (Gimbal and Estimote)
    - iBeacon, Eddystone
  - Key/Wallet tracking devices
    - TrackR, Tile

Core Bluetooth Framework

Two Sides

Central

Wants Data
Can scan or listen for data
Can request for data

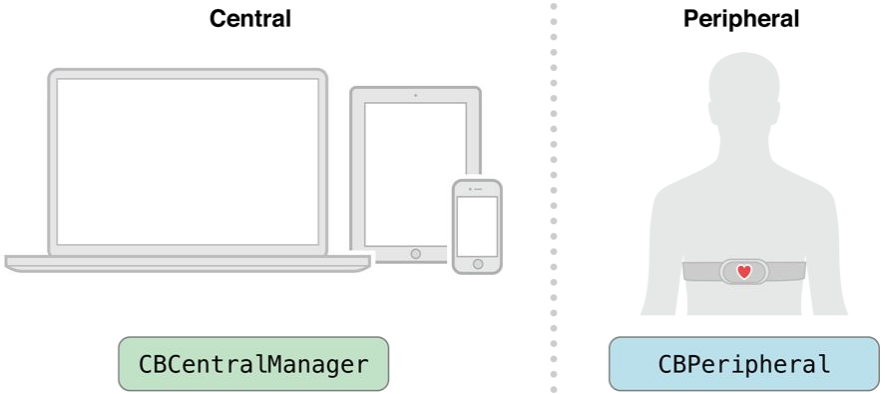
Peripheral

Has Data
Sends advertising packets
Contains services and characteristics

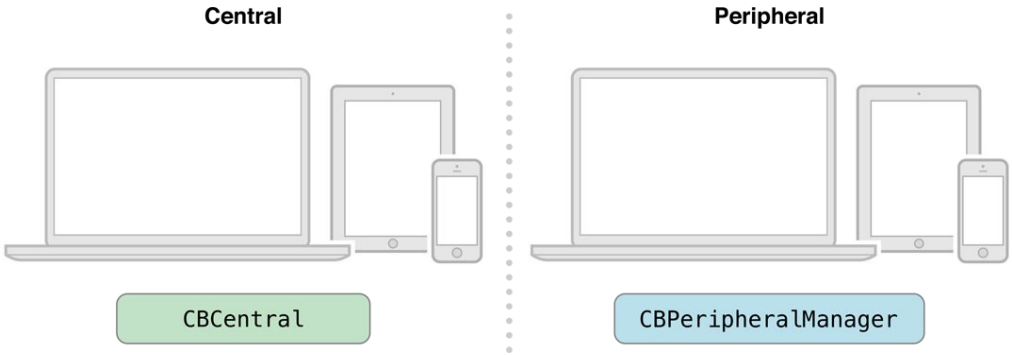
Performing Actions on the Central Side

Initialize

Look for a Peripheral Device



Performing Actions on the Peripheral Side

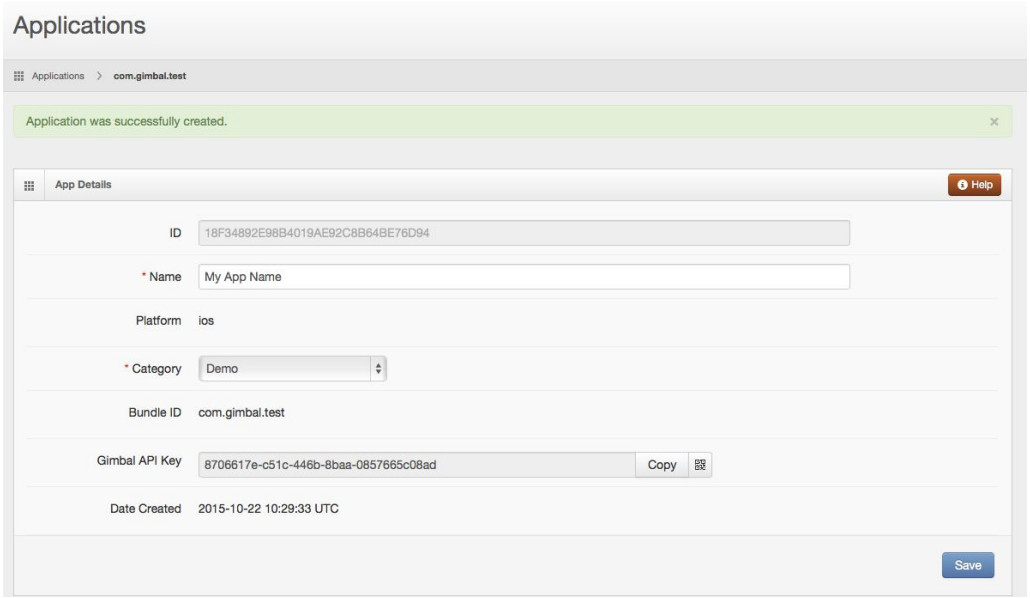


Gimbal Beacons

- Utilizes Bluetooth Low Energy (BLE) technology to provide phones with contextual data once they are in range
- Range distances are measured in RSSI (Received Signal Strength Indicator)
- Once the beacons sends data to the phone, the data is also uploaded to the Gimbal Cloud Platform for more advanced analytics
- Gimbal beacons can be configured using the online Gimbal Manager

Gimbal Framework and iOS SDK

- After installing the gimbal framework, register application and get API Key using online Gimbal Manager



Set API Key like this...

```
class AppDelegate: UIResponder, UIApplicationDelegate {

    var window: UIWindow?


    func application(application: UIApplication, didFinishLaunchingWithOptions launchOptions:
        [NSObject: AnyObject]?) -> Bool {
        // Override point for customization after application launch.
        Gimbal.setAPIKey("Your API KEY", options: nil);
        return true
    }
```

Add GMBLPlaceManagerDelegate to your ViewController

And fufill requirements...

```
var placeManager: GMBLPlaceManager
...

placeManager = GMBLPlaceManager()
...

placeManager.delegate = self
```

```
func placeManager(manager: GMBLPlaceManager, didBeginVisit visit: GMBLVisit) {
    // this will be invoked when a place is entered
}

func placeManager(manager: GMBLPlaceManager, didEndVisit visit: GMBLVisit) {
    // this will be invoked when a place is exited
}
```

But you’re not done yet

Just need to start monitoring

```
if (!GMBLPlaceManager.isMonitoring()){
    GMBLPlaceManager.startMonitoring();
    print("Start Monitoring");
}
```

DEMO

Questions?

For more info: https://docs.gimbal.com/iosdocs/v2/swiftguide.html