

# Bluetooth LE, CoreBluetooth

@zachdennis 

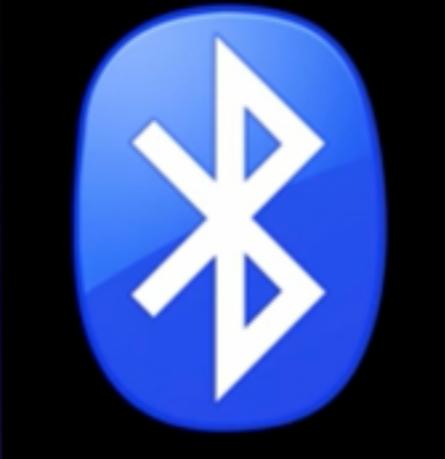
@zdennis 

[mutuallyhuman.com](http://mutuallyhuman.com)

# Bluetooth LE

# What is BTLE?

A short range, ultra-low  
power consuming  
wireless technology.



Shares the “Bluetooth” name, but has different design goals in mind.

# Power Consumption

Years, not hours or  
days.



# Short range

~ 50m

# Packet-based

Short bursts of data.

# Intervals

Ad

Ad

Data

Ad

Data

# Multiple channels

#1



#2

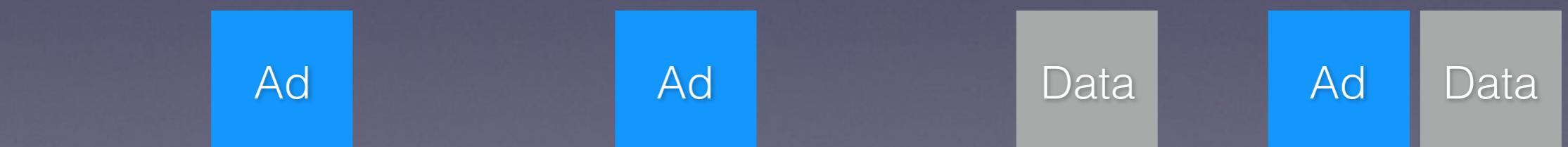
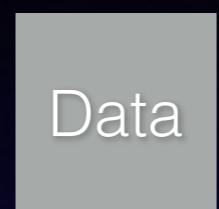
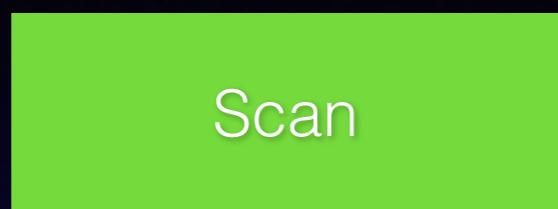
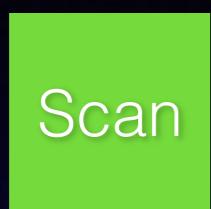


#3



# Device A is looking.

---



# Device B is advertising.

# Frequency hopping

# Connection-less

Devices do not need to maintain connections.

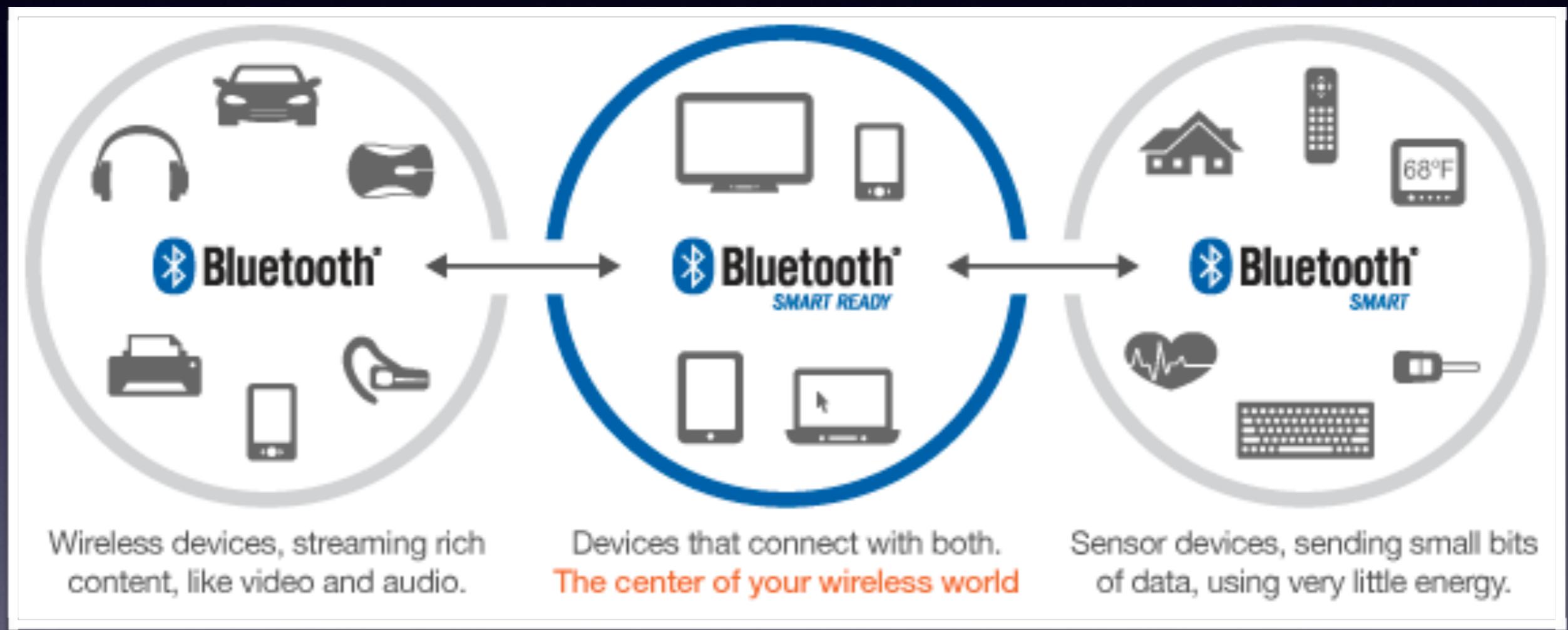
# Pairing

e.g. iOS - happens when Insufficient Authentication error code is sent from the peripheral.

# Security

Devices pair, keys are distributed, and the connection is encrypted.

Encryption is AES-128.



# Why BTLE?



Health Care



Sports/Fitness



Security



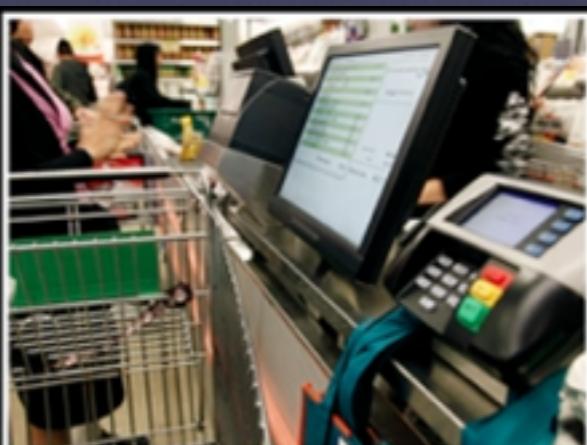
Automation



Entertainment



Toys



Pay Systems



Time Services

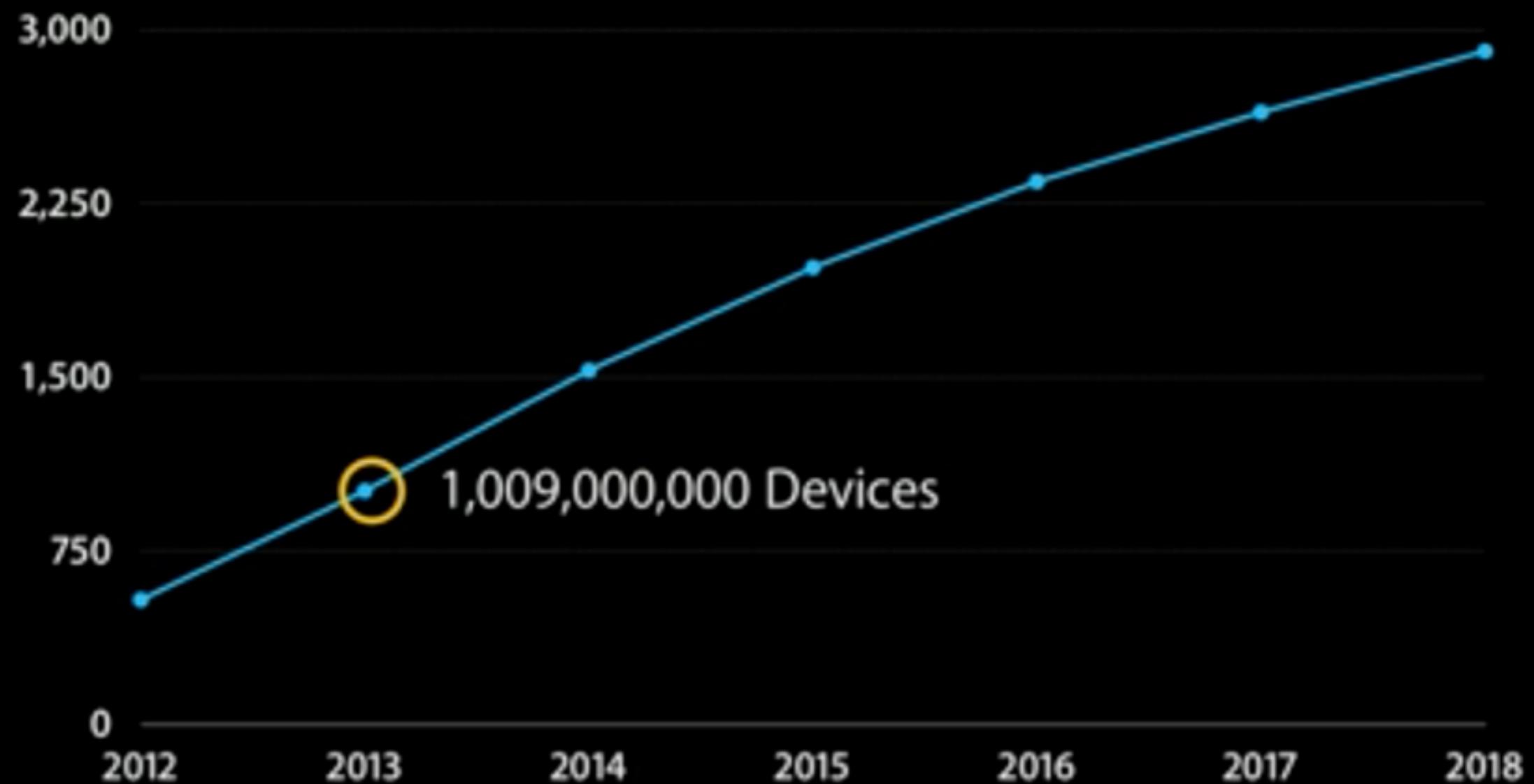


Proximity

\*Borrowed from 2012 WWDC CoreBluetooth Talk

1 billion+ devices

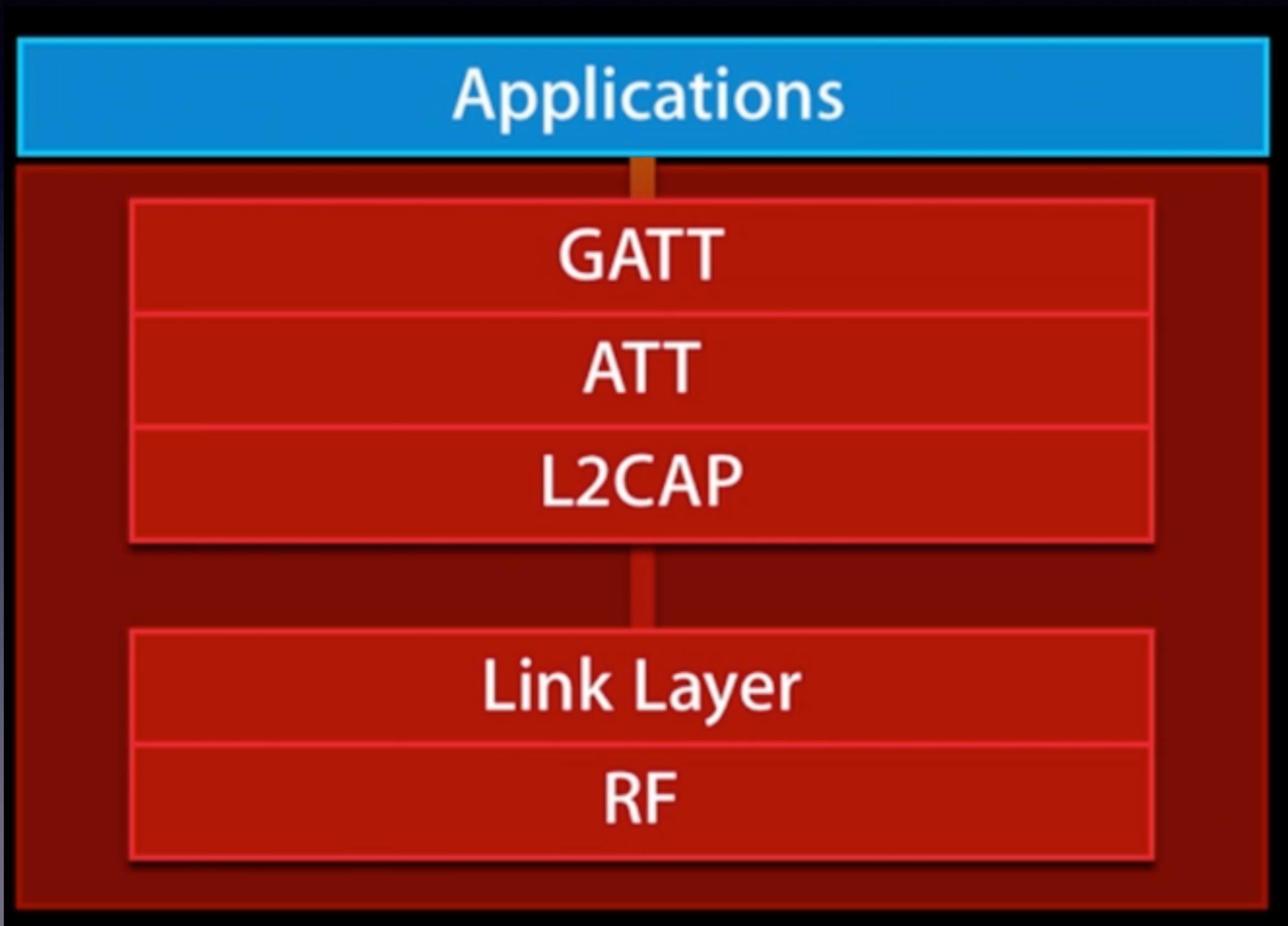
# Projected BLE Accessories Shipped



\*Borrowed from 2013 Apple WWDC talk

# How BTLE Works

# The Stack



# Key Terms

- Central
- Peripheral
- Service
- Characteristic
- Descriptor

# Central

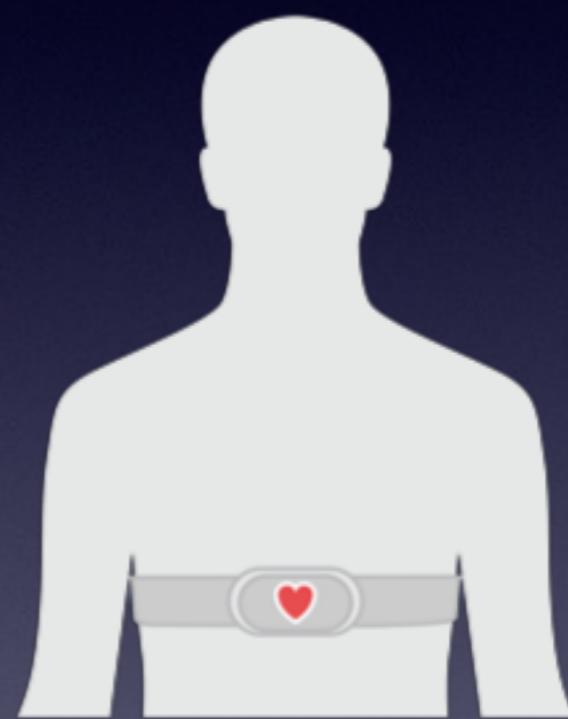
(wants data)



Central

# Peripheral

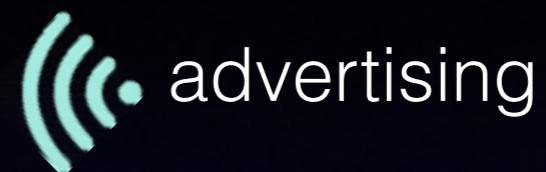
(has data)



Peripheral

Central

Peripheral



advertising

**Central**

scan

**Peripheral**



advertising

**Central**

scan

**Peripheral**



# Central

# Peripheral



scan



discover services



Heart Monitor

# Central

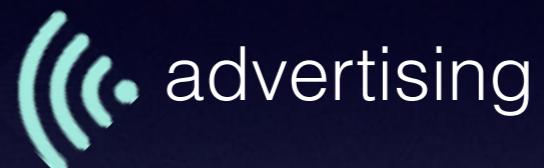
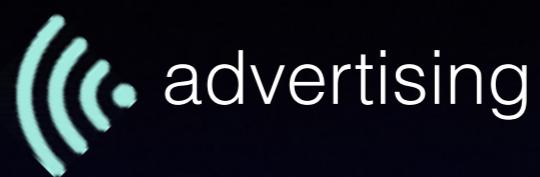
scan ..... •

discover services • .....

discover characteristics • .....



# Peripheral

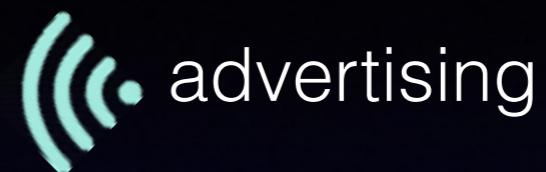


Heart Monitor

BPM

# Central

# Peripheral



scan ..... advertising



discover services ..... Heart Monitor

discover characteristics ..... BPM

read value ..... ➤ 95



# Central

# Peripheral

scan ..... advertising



# advertising

discover services • • • • • • • • • • • • • •

# Heart Monitor

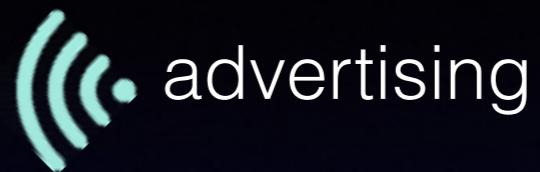
# discover characteristics • • • • • • • • • • •

BPM

read value ..... 95

# Central

# Peripheral



scan ..... advertising

discover services ..... Heart Monitor

discover characteristics ..... BPM

read value ..... 95

observe value ..... BPM 95

BPM 95

value changed < ..... BPM 98

# Service

A service is a human-readable specification of a set of **characteristics** and their associated behavior.

# Two kinds of services

There are primary services and secondary services.

# Nested services

Services can contain other services.

# Characteristic

A characteristic is a bit of data that has a known format labeled with a UUID.

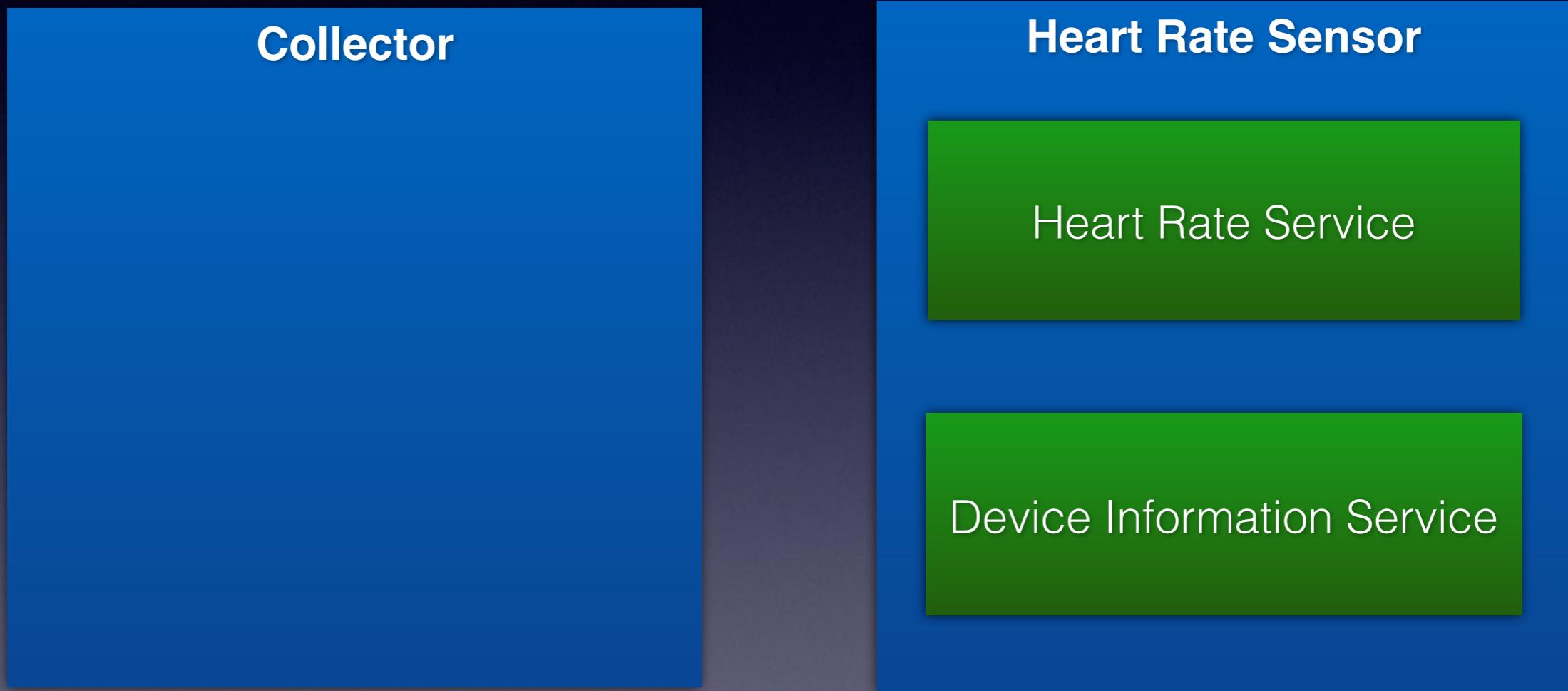
They are intended for computer-readable format as opposed to human-readable text.

# Profiles

A profile is a specification that describe two or more devices, with one or more services on each device, how they discover each other, connect, and otherwise interact.

Profiles define roles for devices to play.

# Heart Rate Profile



# Heart Rate Profile

**GATT CLIENT**

Collector

**GATT SERVER**

Heart Rate Sensor

Heart Rate Service

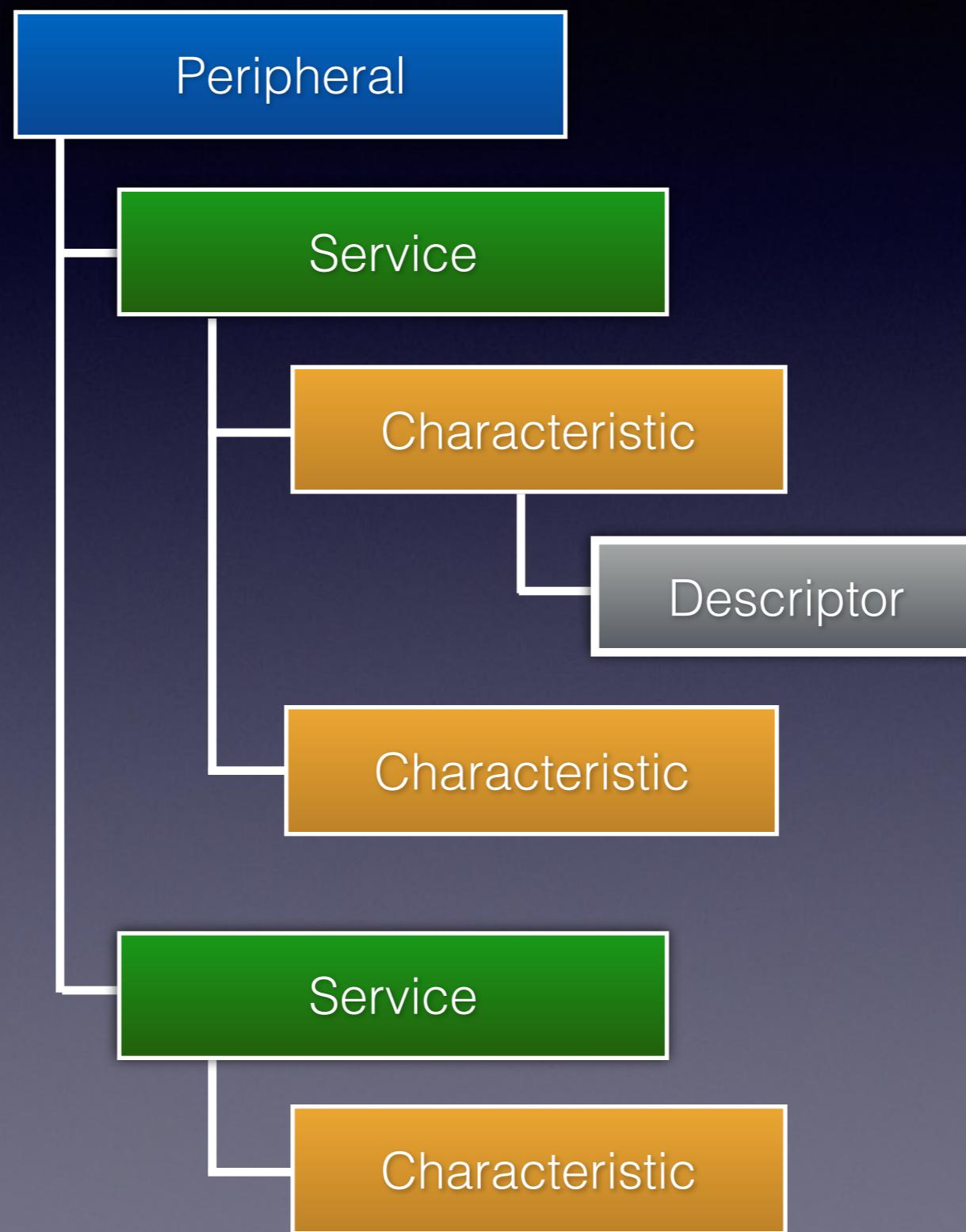
Device Information Service

# Profiles & Services

Profiles contain services.

Services can be contained by multiple profiles.

# Anatomy of a Peripheral



# Apple's Approach to BTLE

Simple  
Powerful

# Technology Stack

Applications

Core Bluetooth

GATT

ATT

L2CAP

Link Layer

RF

\*Borrowed from 2012 WWDC CoreBluetooth Talk

# Supported Profiles

- Generic Attribute Profile Service
- Generic Access Profile Service
- Bluetooth Low Energy HID Service
- Battery Service
- Time Service
- Apple Notification Center Service

# CoreBluetooth

# Object Model

# Main Objects

CBCentralManager

CBCentralManagerDelegate

CBPeripheralManager

CBPeripheralManagerDelegate

# Data Objects

CBPeripheral

CBPeripheralDelegate

CBService

CBCharacteristic

CBCentral

CBMutableService

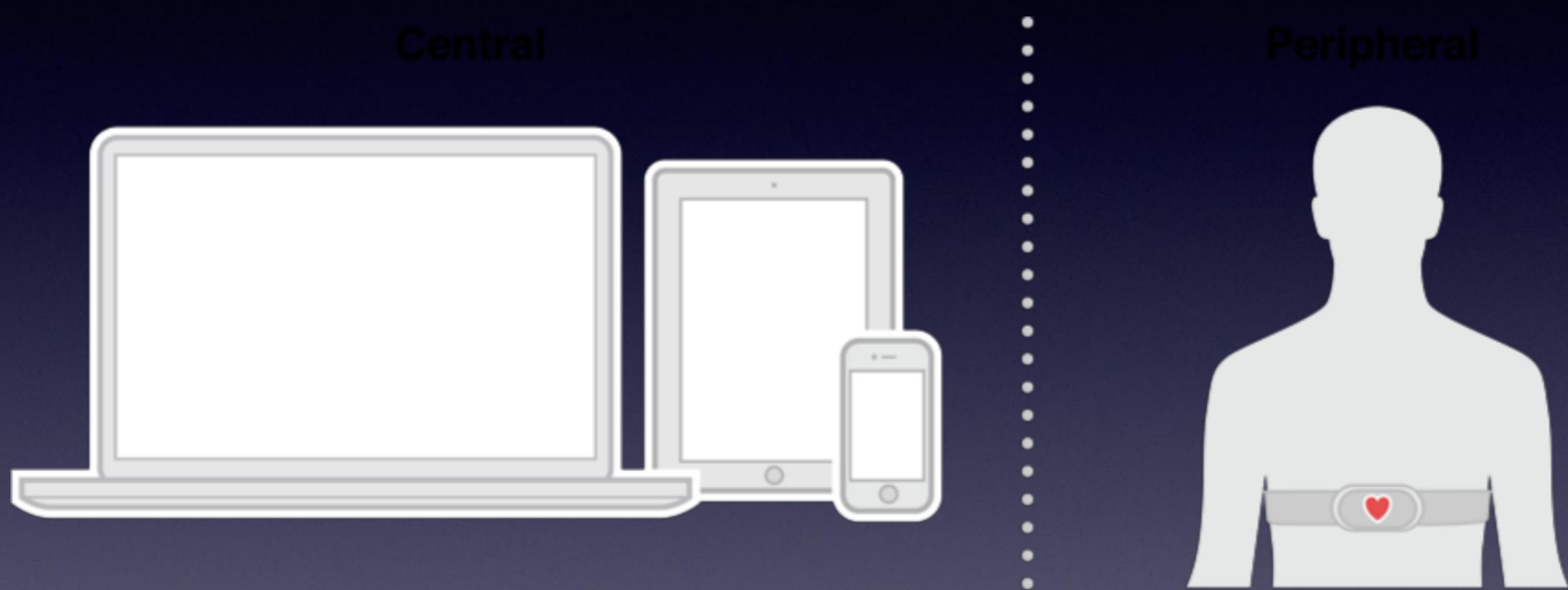
CBMutableCharacteristic

# Helper Objects

CBUUID

CBATTRequest

Being a Central



Central

Peripheral

# CBCentralManager

## Scanning

- scanForPeripheralsWithServices:options

## Stop scanning

- stopScan

## Connecting to peripherals

- connectPeripheral:options
- cancelPeripheralConnection:

## Retrieving known peripherals

- retrieveConnectedPeripheralsWithServices:
- retrievePeripheralsWithIdentifiers:

# CBCentralManagerDelegate

## Monitoring Connections with Peripherals

- centralManager:didConnectPeripheral:
- centralManager:didDisconnectPeripheral:error:
- centralManager:didFailToConnectPeripheral:error:

## Discovering and Retrieving Peripherals

- centralManager:didDiscoverPeripheral:advertisementData:RSSI:
- centralManager:didRetrieveConnectedPeripherals:
- centralManager:didRetrievePeripherals:

## Monitoring Changes to the Central Manager's State

- centralManagerDidUpdateState:
- centralManager:willRestoreState:

# CBPeripheralDelegate

## Discovering Services

- peripheral:didDiscoverServices:
- peripheral:didDiscoverIncludedServicesForService:error:

## Discovering Characteristics and Characteristic Descriptors

- peripheral:didDiscoverCharacteristicsForService:error:
- peripheral:didDiscoverDescriptorsForCharacteristic:error:

## Retrieving Characteristic and Characteristic Descriptor Values

- peripheral:didUpdateValueForCharacteristic:error:
- peripheral:didUpdateValueForDescriptor:error:

## Writing Characteristic and Characteristic Descriptor Values

- peripheral:didWriteValueForCharacteristic:error:
- peripheral:didWriteValueForDescriptor:error:

## Managing Notifications for a Characteristic's Value

- peripheral:didUpdateNotificationStateForCharacteristic:error:

## Retrieving a Peripheral's Received Signal Strength Indicator (RSSI) Data

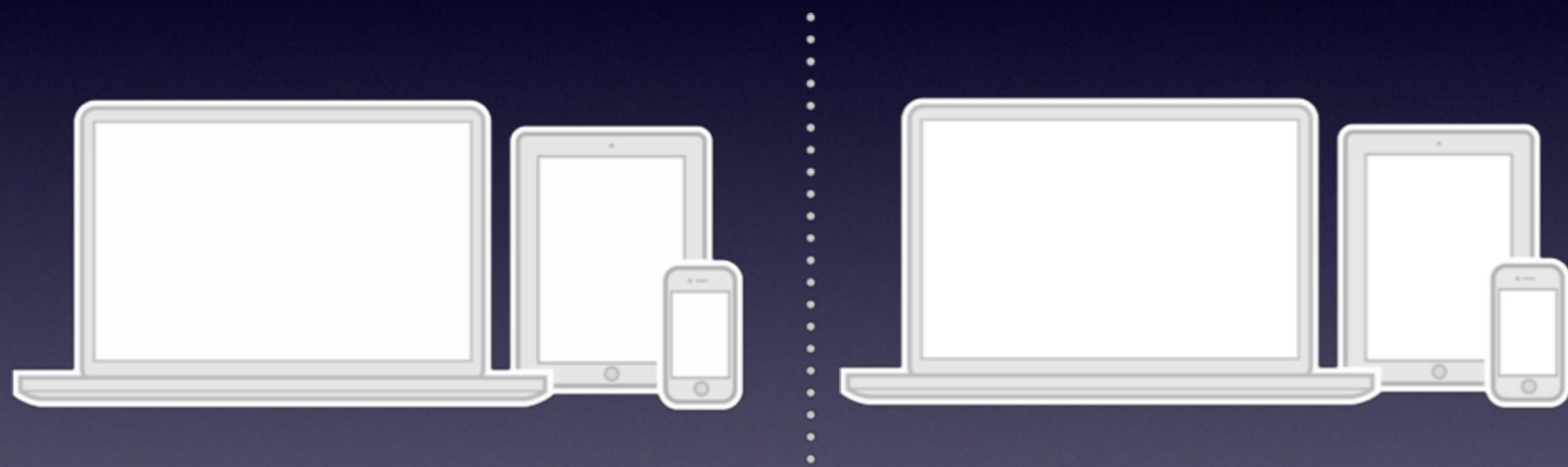
- peripheralDidUpdateRSSI:error:

## Monitoring Changes to a Peripheral's Name or Services

- peripheralDidUpdateName:
- peripheral:didModifyServices:

> code <

# Being a Peripheral



**Central**

**Peripheral**

# CBPeripheralManager

## Services

- addService:
- removeService:
- removeAllServices:

## Advertising

- startAdvertising:
- stopAdvertising
- isAdvertising

## Notifying observing devices of updates

- updateValue:forCharacteristic:onSubscribedCentrals:

## Responding to Read/Write Requests

- respondToRequest:withResult

# **CBPeripheralManagerDelegate**

## **Monitoring Changes to the Peripheral Manager's State**

- peripheralManagerDidUpdateState:
- peripheralManager:willRestoreState:

## **Adding Services**

- peripheralManager:didAddService:error:

## **Advertising Peripheral Data**

- peripheralManagerDidStartAdvertising:error:

## **Monitoring Subscriptions to Characteristic Values**

- peripheralManager:central:didSubscribeToCharacteristic:
- peripheralManager:central:didUnsubscribeFromCharacteristic:
- peripheralManagerIsReadyToUpdateSubscribers:

## **Receiving Read and Write Requests**

- peripheralManager:didReceiveReadRequest:
- peripheralManager:didReceiveWriteRequests:

# **CBMutableService**

## **Initializing a Mutable Service**

- initWithType:primary:

## **Managing a Mutable Service**

- UUID
- isPrimary
- characteristics
- includedServices

# **CBMutableCharacteristic**

## **Initializing a Mutable Characteristic**

- initWithType:properties:value:permissions:

## **Managing a Mutable Characteristic**

- UUID
- value
- descriptors
- properties
- permissions
- subscribedCentrals

> code <

Foreground vs.  
Background

Two background  
modes.

# bluetooth-central

“Uses Bluetooth LE accessories” in Xcode

# bluetooth-peripheral

“Acts as Bluetooth LE accessory” in Xcode

Specify background  
modes in Info.plist.

Nuances of  
backgrounding

# Central Scanning

- Scanning (as a Central) acts differently in the background.
  - Scan options are ignored. Multiple discoveries of a peripheral are coalesced into one.
  - The scan interval may increase and your app may take longer to discover a peripheral.

# Peripheral Advertising

- Advertising in the background differs from foreground mode:
  - The `CBAdvertisementDataLocalNameKey` is not advertised.
  - The frequency at which your app advertises may decrease.
  - Service UUIDs may not be advertised. Apple does best effort.

# Peripheral Events

- iOS will wake up your app to receive events: read, write, and subscribe events.

# Caching

- Services, characteristics and characteristic descriptors are cached
- Characteristic value is kind of cached. When discovered the last read value will be provided, but it's up to you to use it (static values) or read the value (dynamic values) from the peripheral.

# State Preservation and Restoration

- Optional feature.
- Why? if your app is background it can be terminated by the OS
- iOS will store the state of the application and act on behalf of it as a proxy. When it receives an event your app is waiting for it will start the app back up in the background to allow it to process it
- Single method for you to implement to restore the state of your app.

# iBeacons

# iBeacons

## Ranging

Unknown

Immediate

Near

Far



A part of CoreLocation

Just a data format in the advertising packets.

Apple to release actual profile on \_\_\_\_\_.

# Tips

CoreBluetooth lives in  
IOBluetooth for Mac  
apps.

# No more simulator support as of iOS7

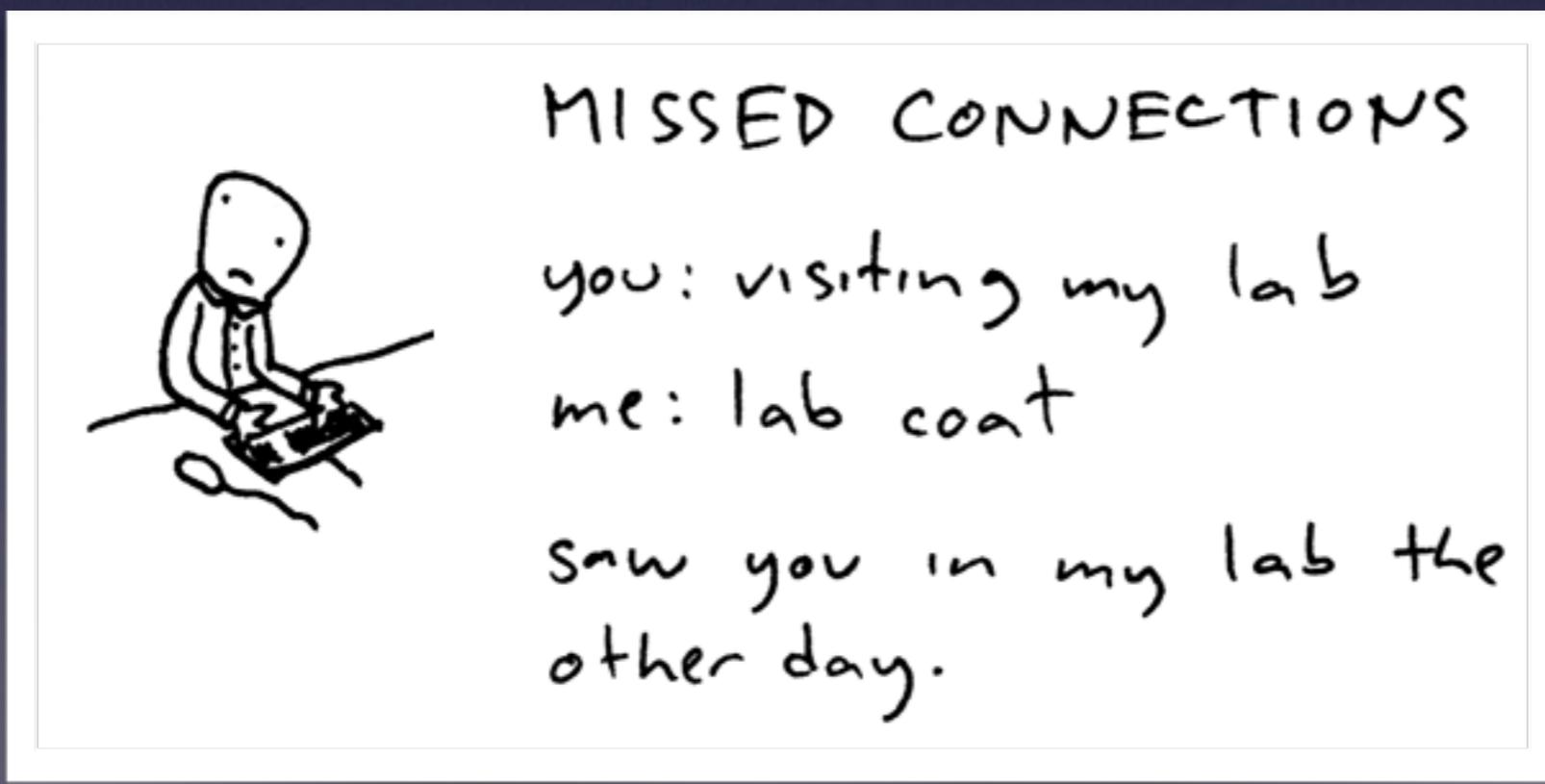


Keep references to  
CBPeripheral and CBCentrals  
if you plan on using them.



**YOSO  
YOCO  
DSWYDNT**

# Done with peripheral, disconnect.



**CBPeripheral** and  
**CBCentral** objects can be  
dictionary keys.

If you're a peripheral  
support characteristic  
**notifications.**

Require paired  
connections to acquire  
sensitive information



Finding peripherals the system  
already knows about

## CentralManager

- retrieveConnectedPeripheralsWithServices:
- retrievePeripheralsWithIdentifiers:

# MTU Exchange Requests

## Performance Improvements

Sending data

Default MTU  
23 bytes



Increased MTU  
??? bytes



# MTU Exchange Requests

- Allows more data to be sent in one go
- Less packet overhead
- up to 20% increase in throughput
- Free.

# App store recommendations from Apple

- include the device with your submission
- be explicit about services the device provides
- provide instructions for how to use the device/  
app
- or don't and cross your fingers



@zachdennis 

@zdennis 

[mutuallyhuman.com](http://mutuallyhuman.com)