

#WWDC19

What's New in Core Bluetooth

Duy Phan, Bluetooth Engineer

Yilok Wong, Bluetooth Engineer

Meghna Lav, Bluetooth Engineer

Introduction

Low energy 2 Mbps

Advertising extensions

Core Bluetooth for BR/EDR

Core Bluetooth dual-mode

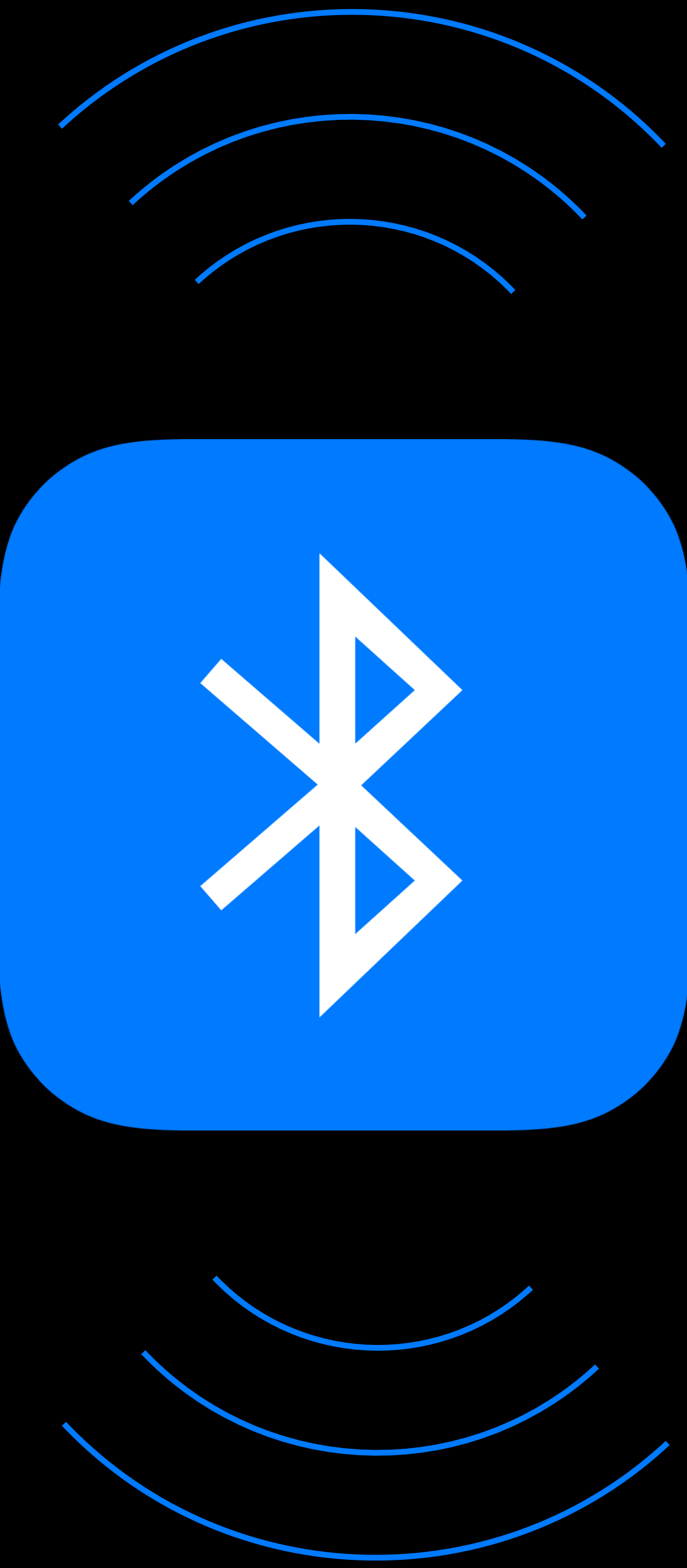
User privacy

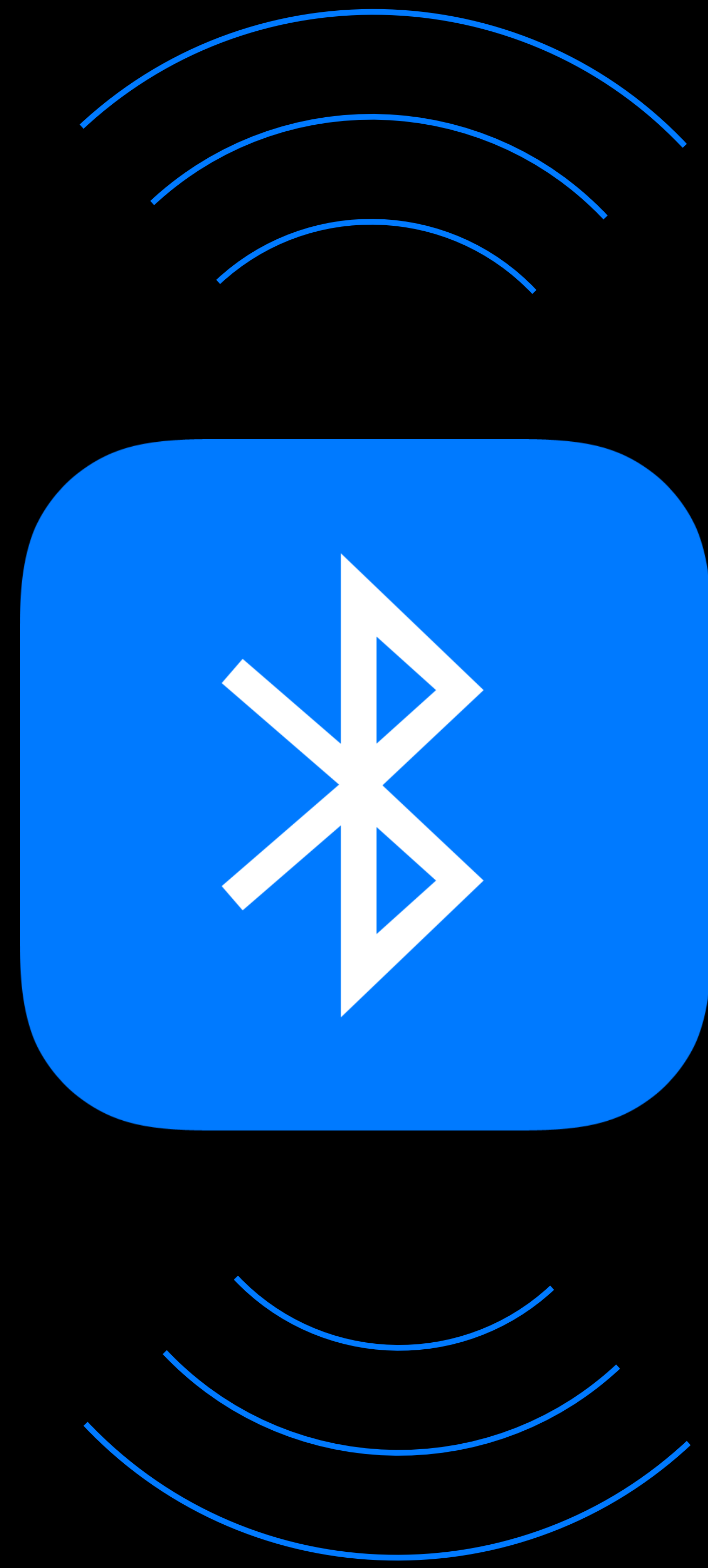
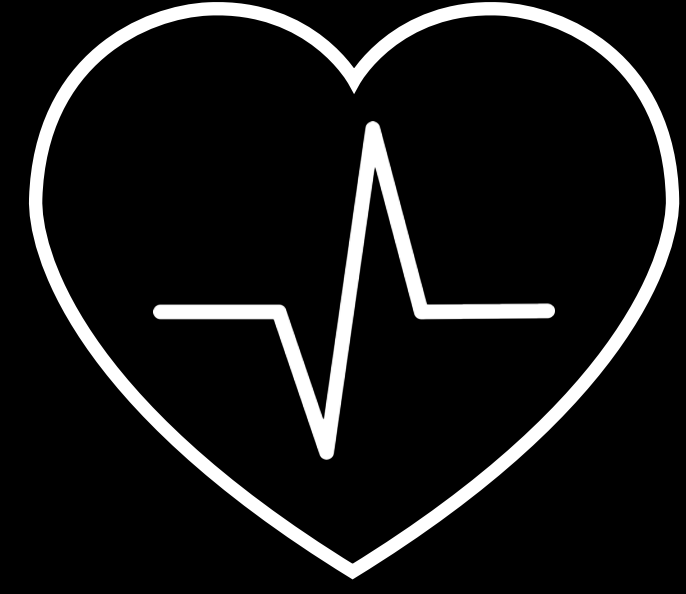
PacketLogger

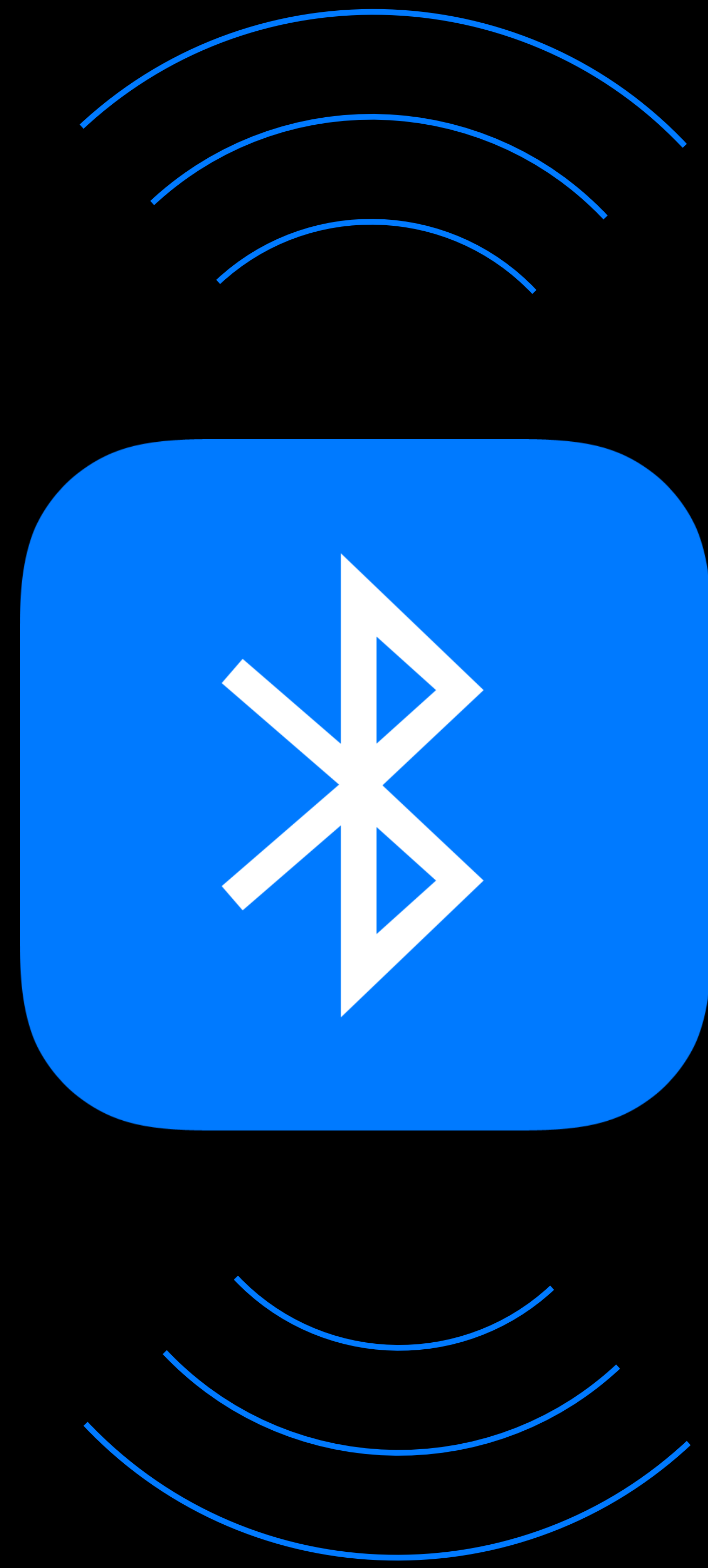
Summary

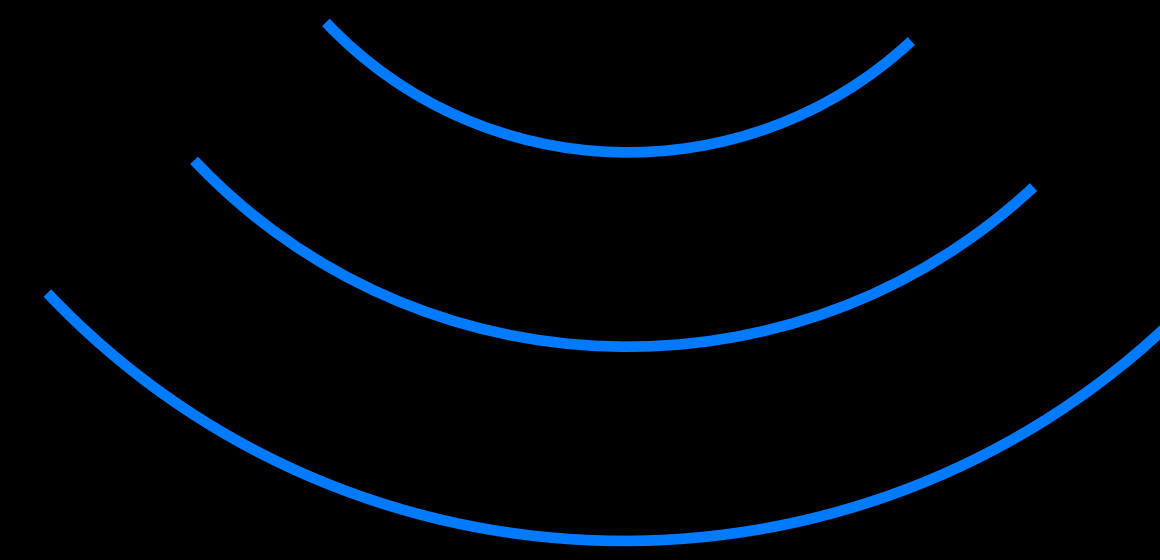
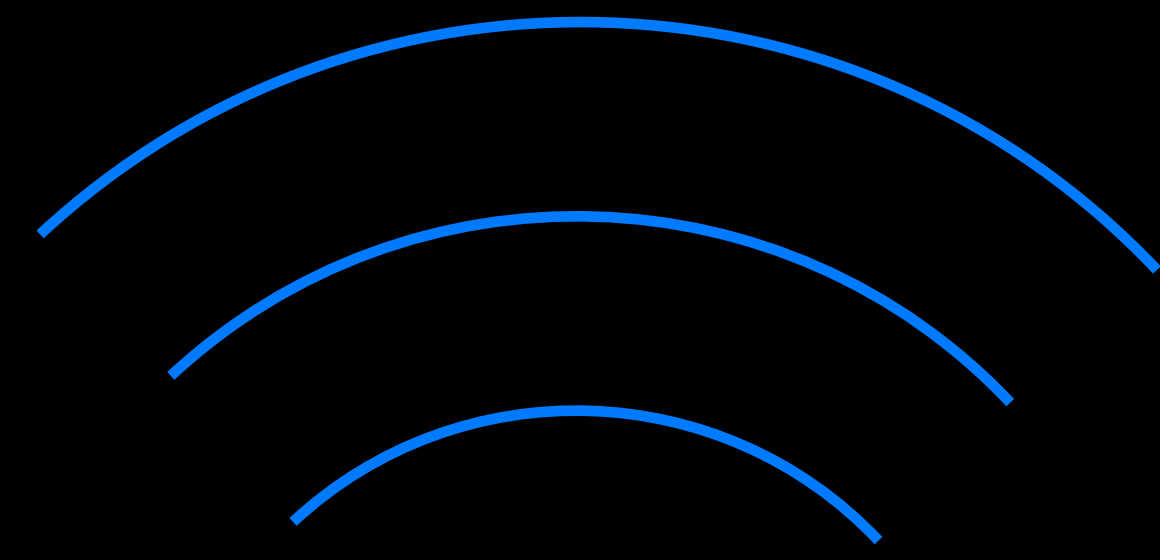
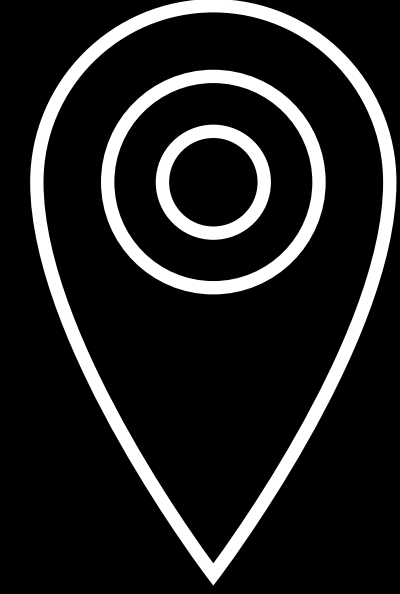
Introduction

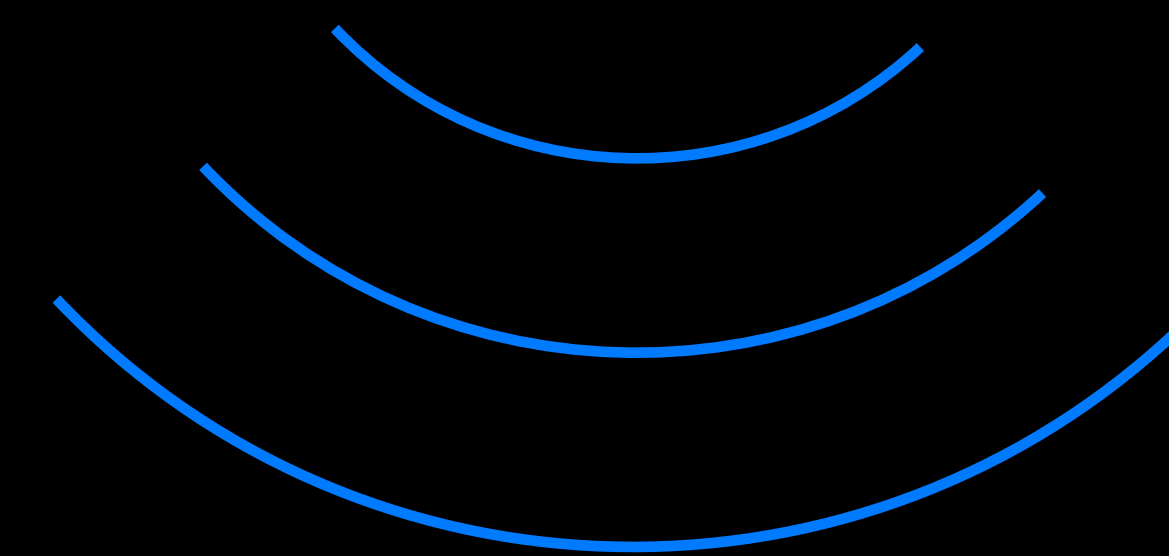
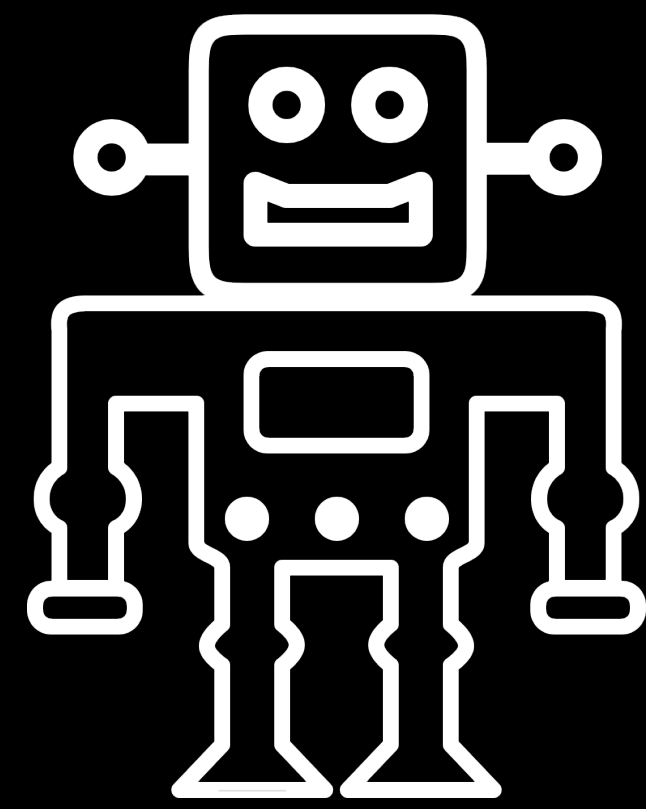
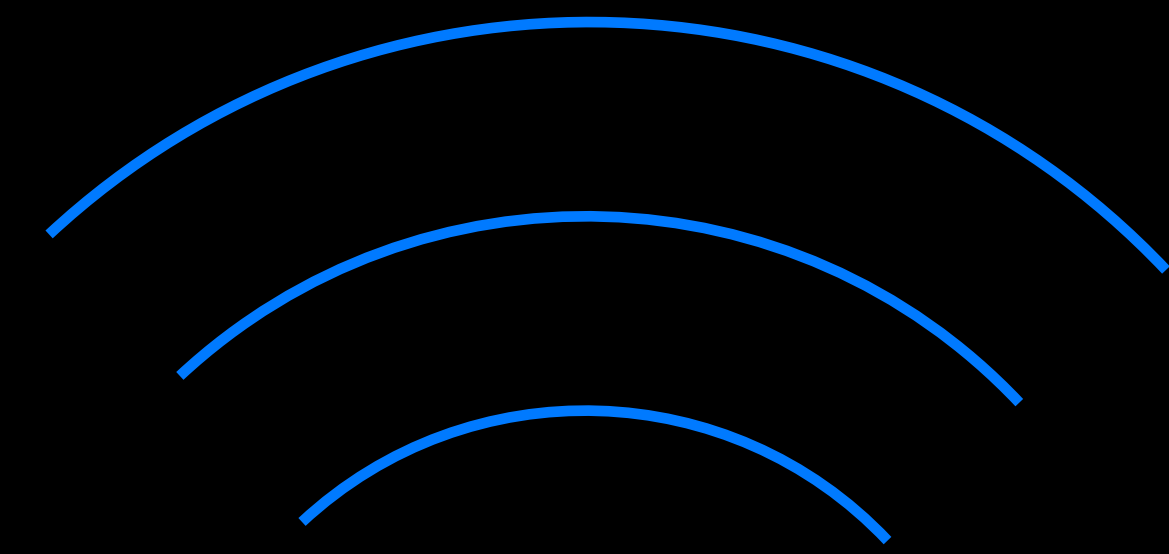
Duy Phan, Bluetooth Engineer

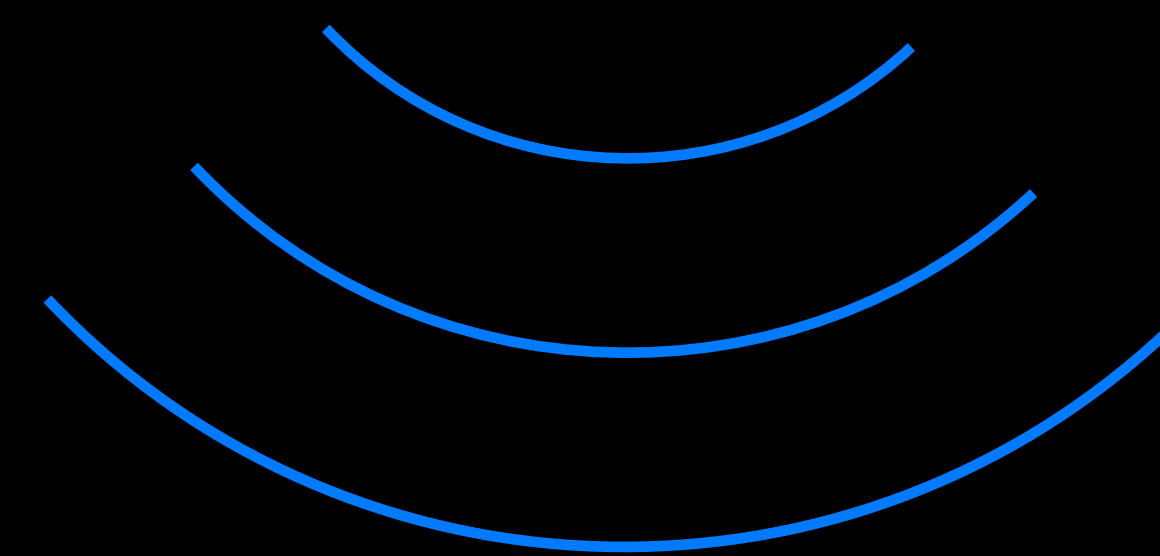
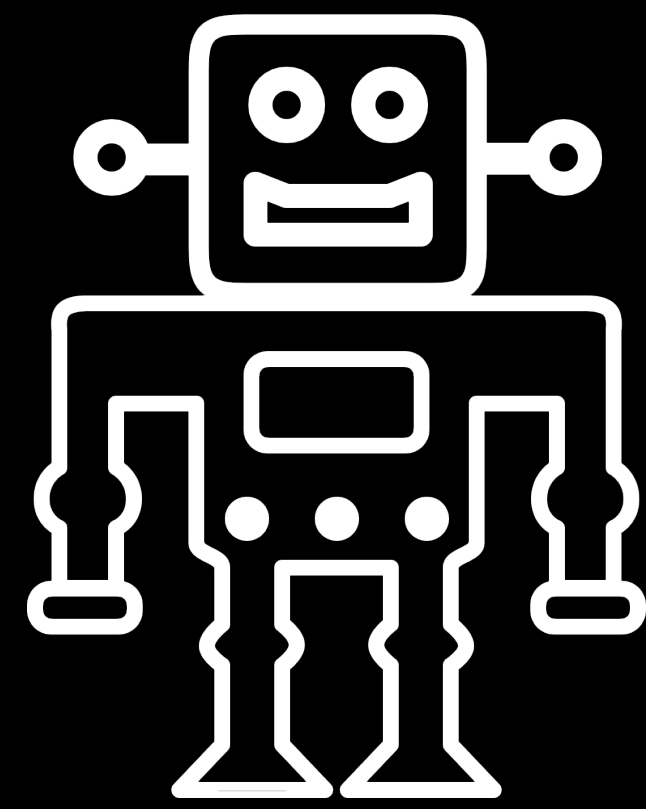
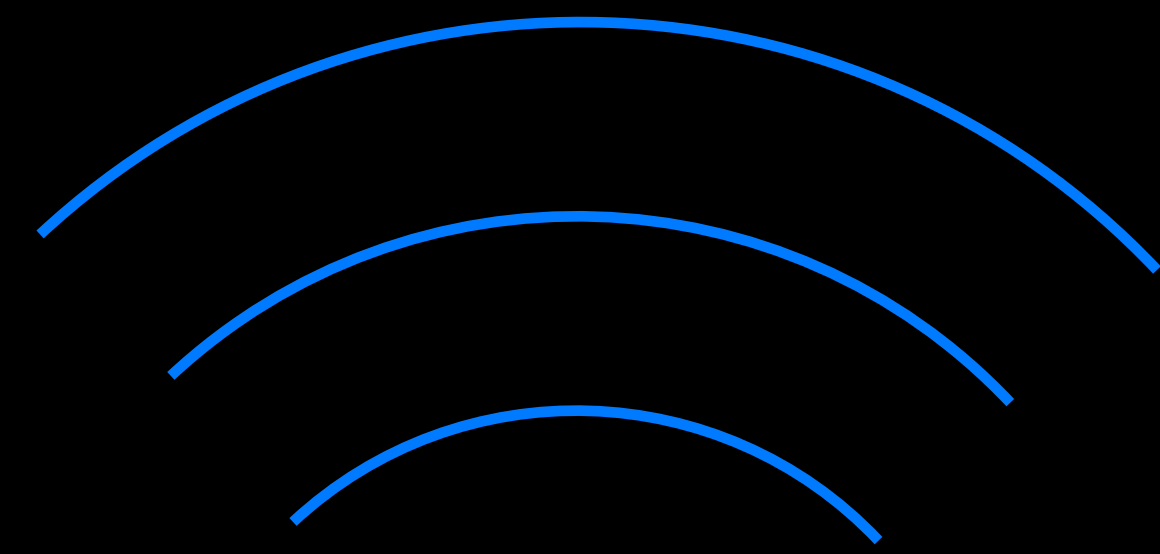


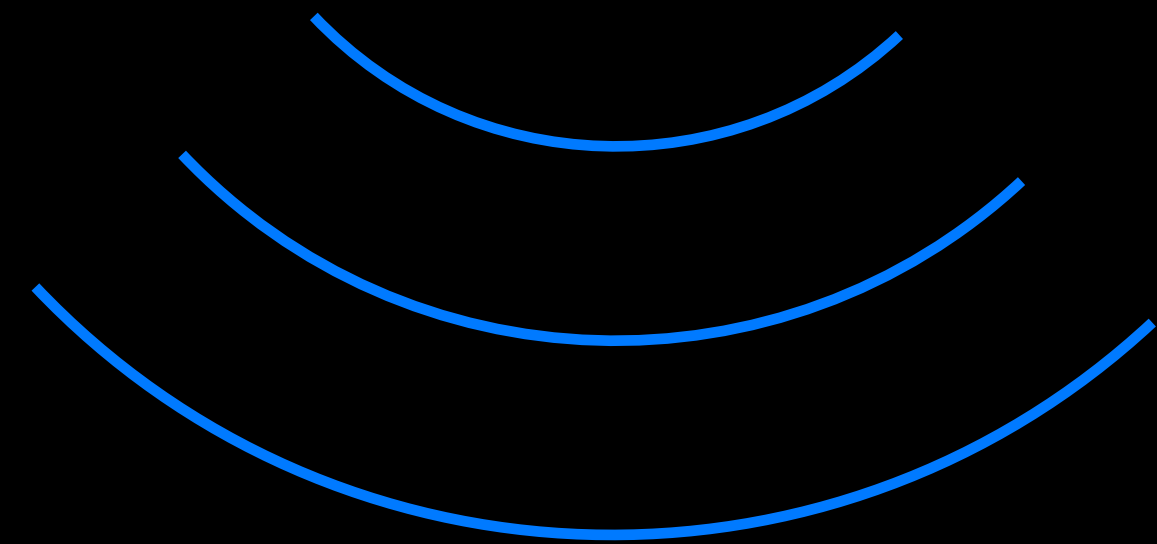
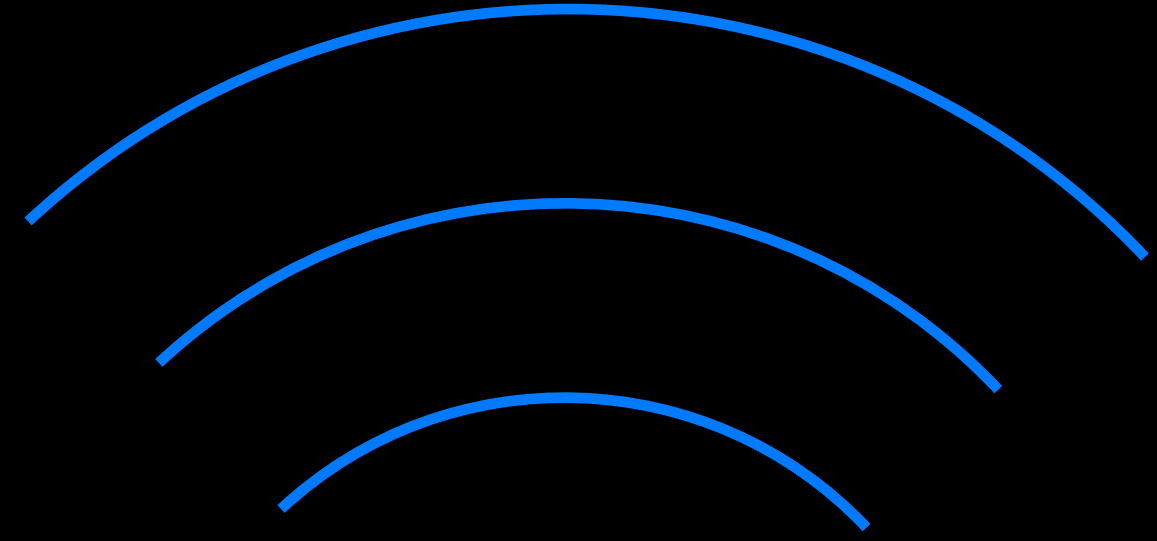
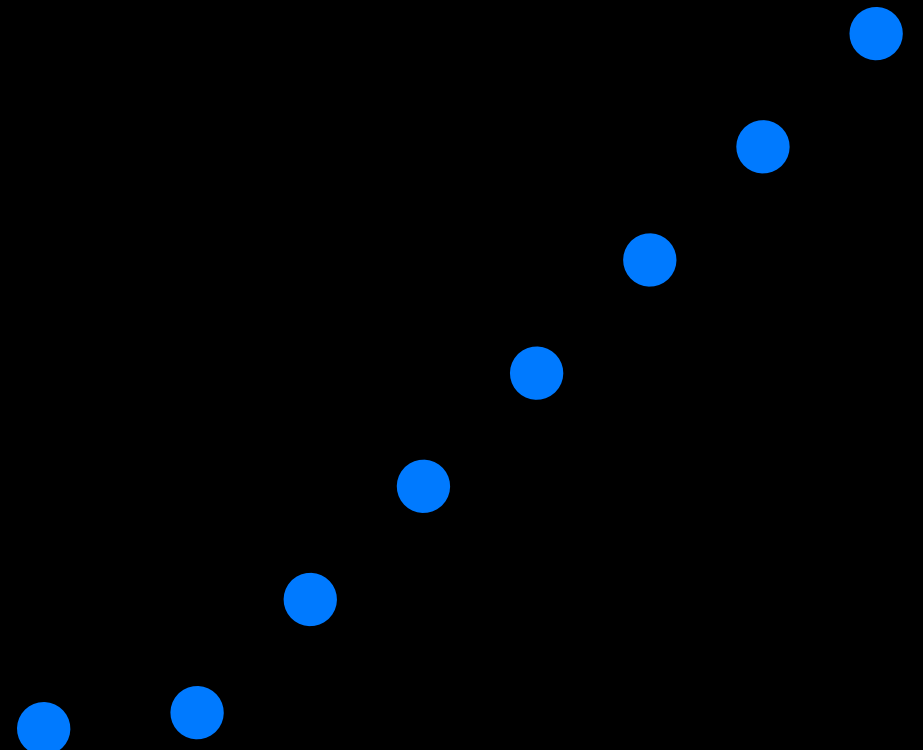
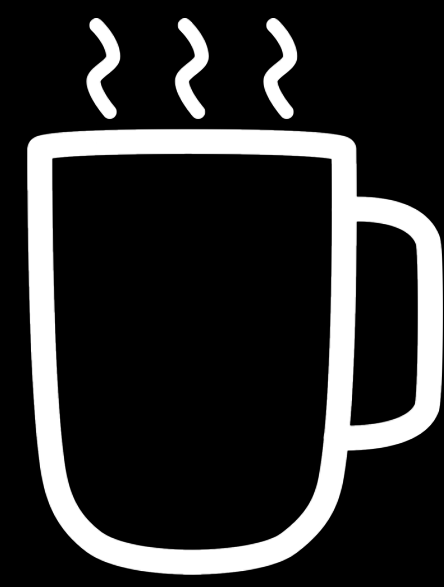
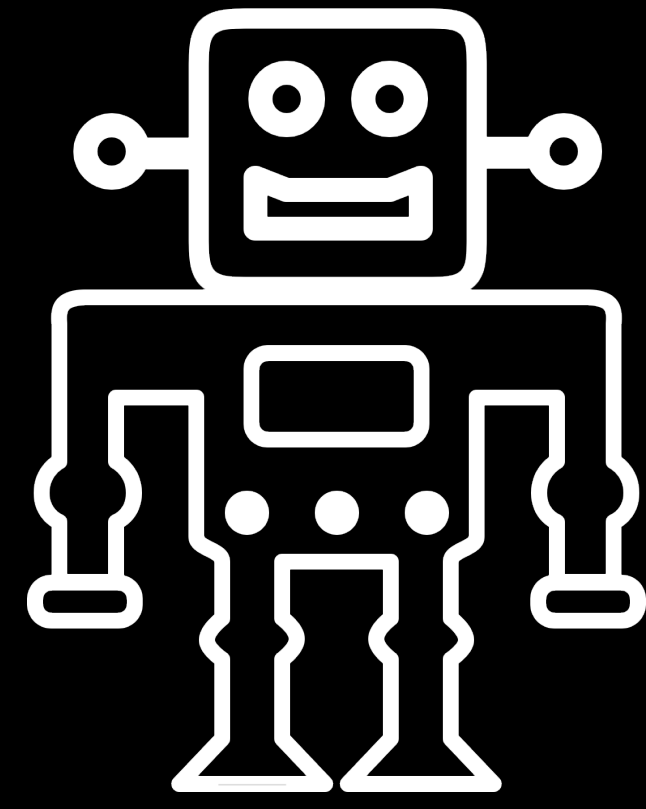
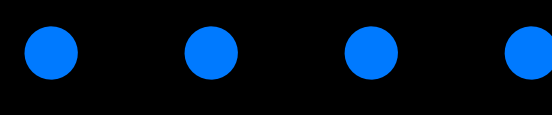
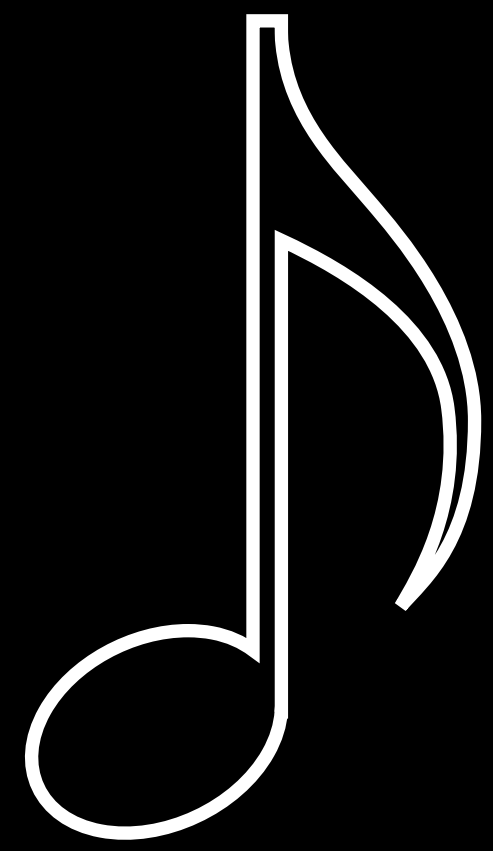
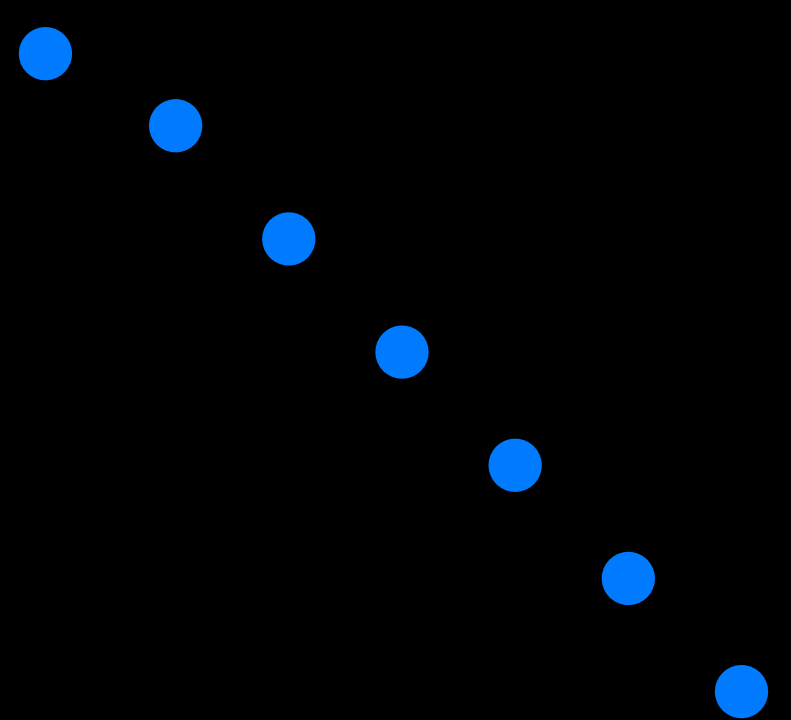
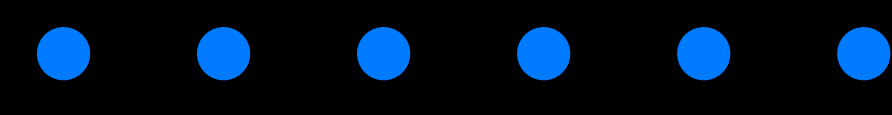


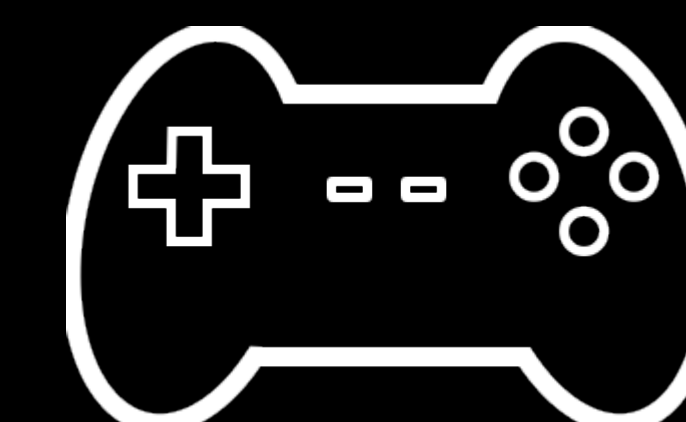
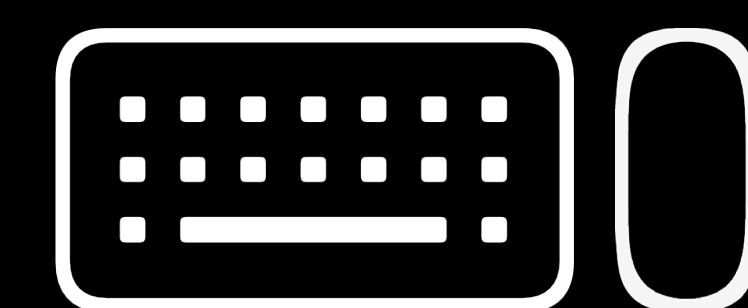
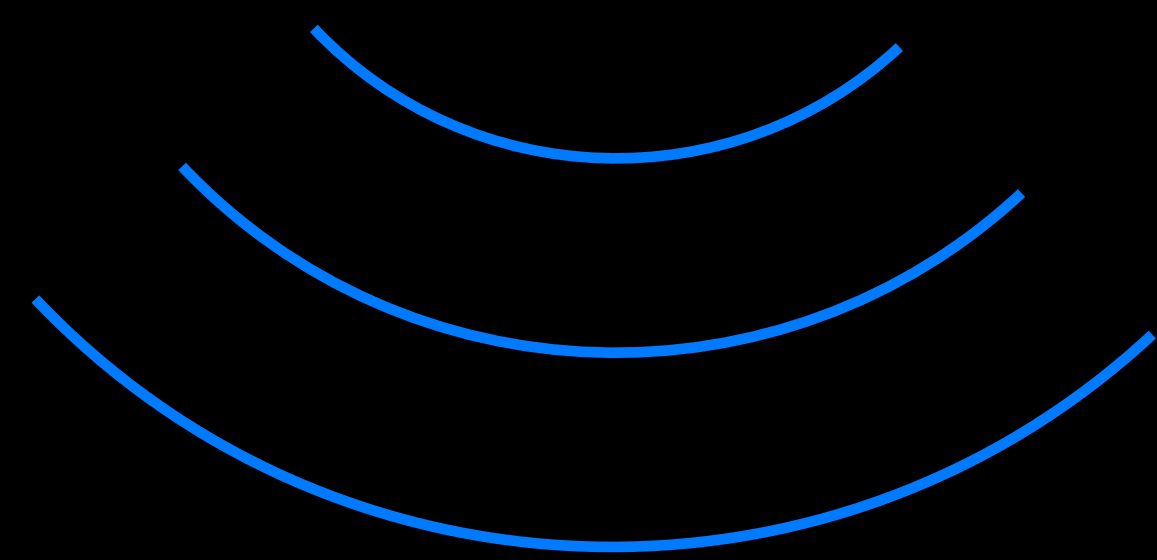
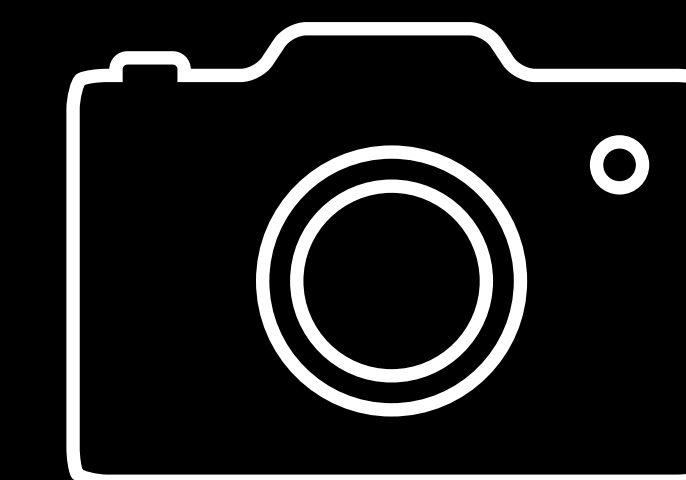
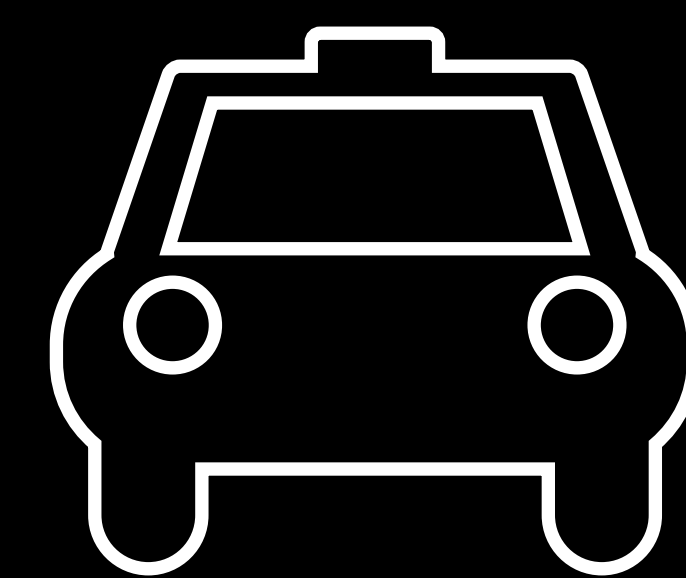
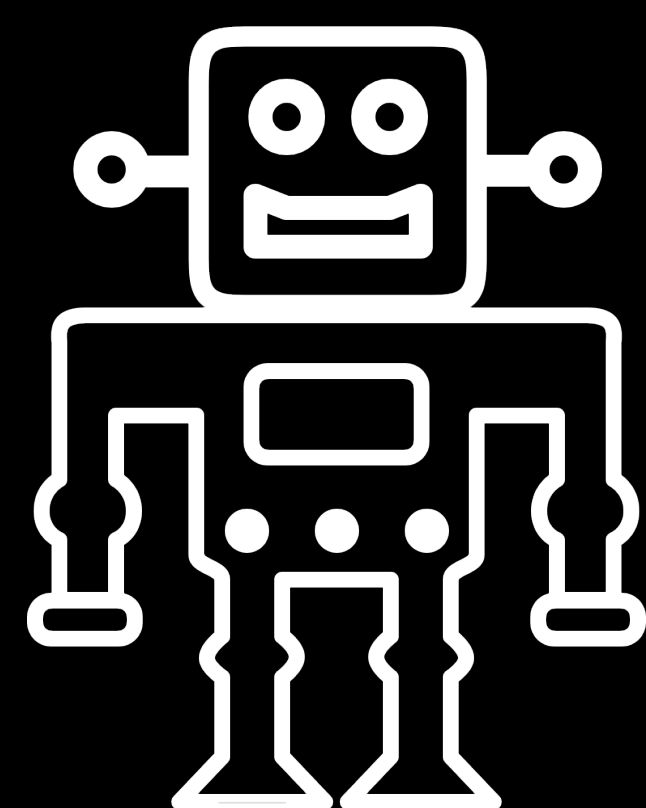
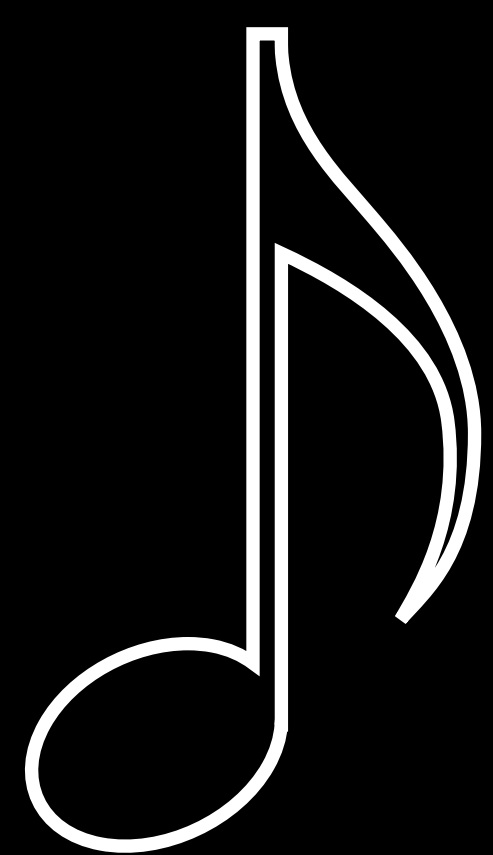
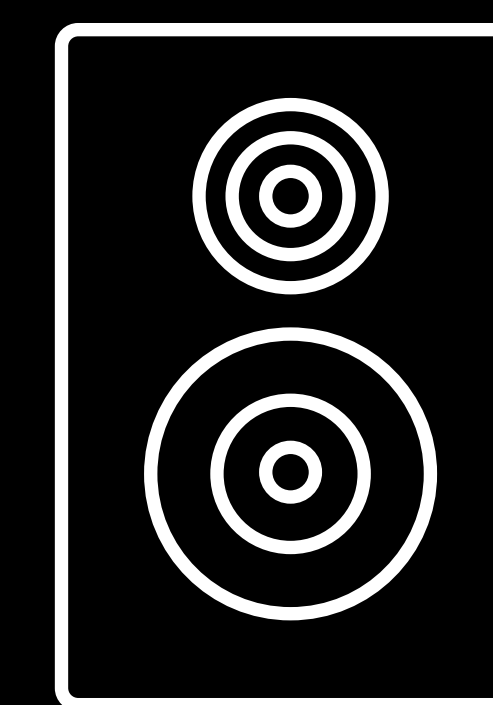
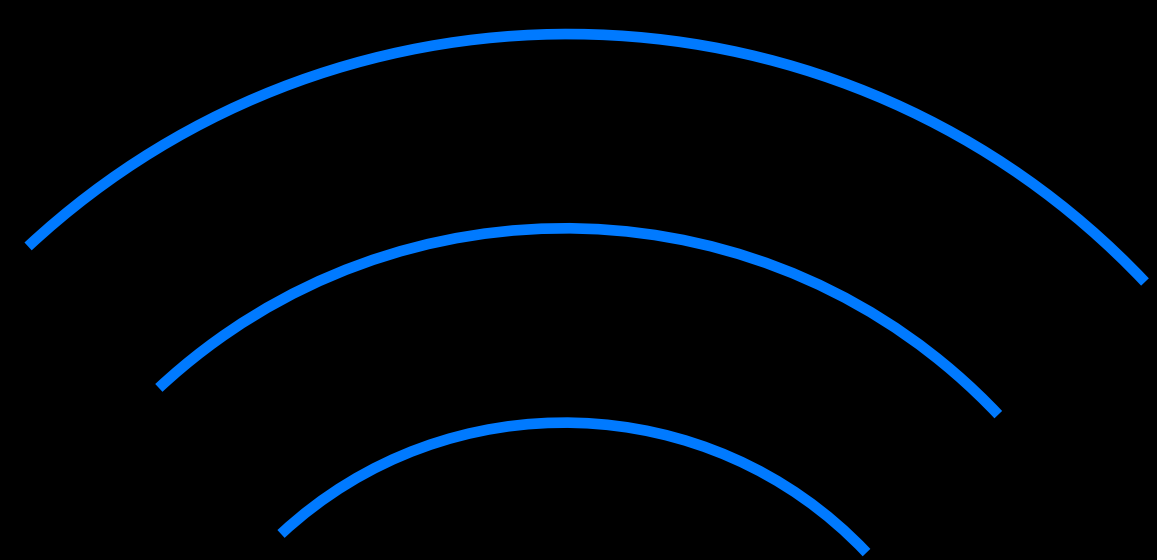
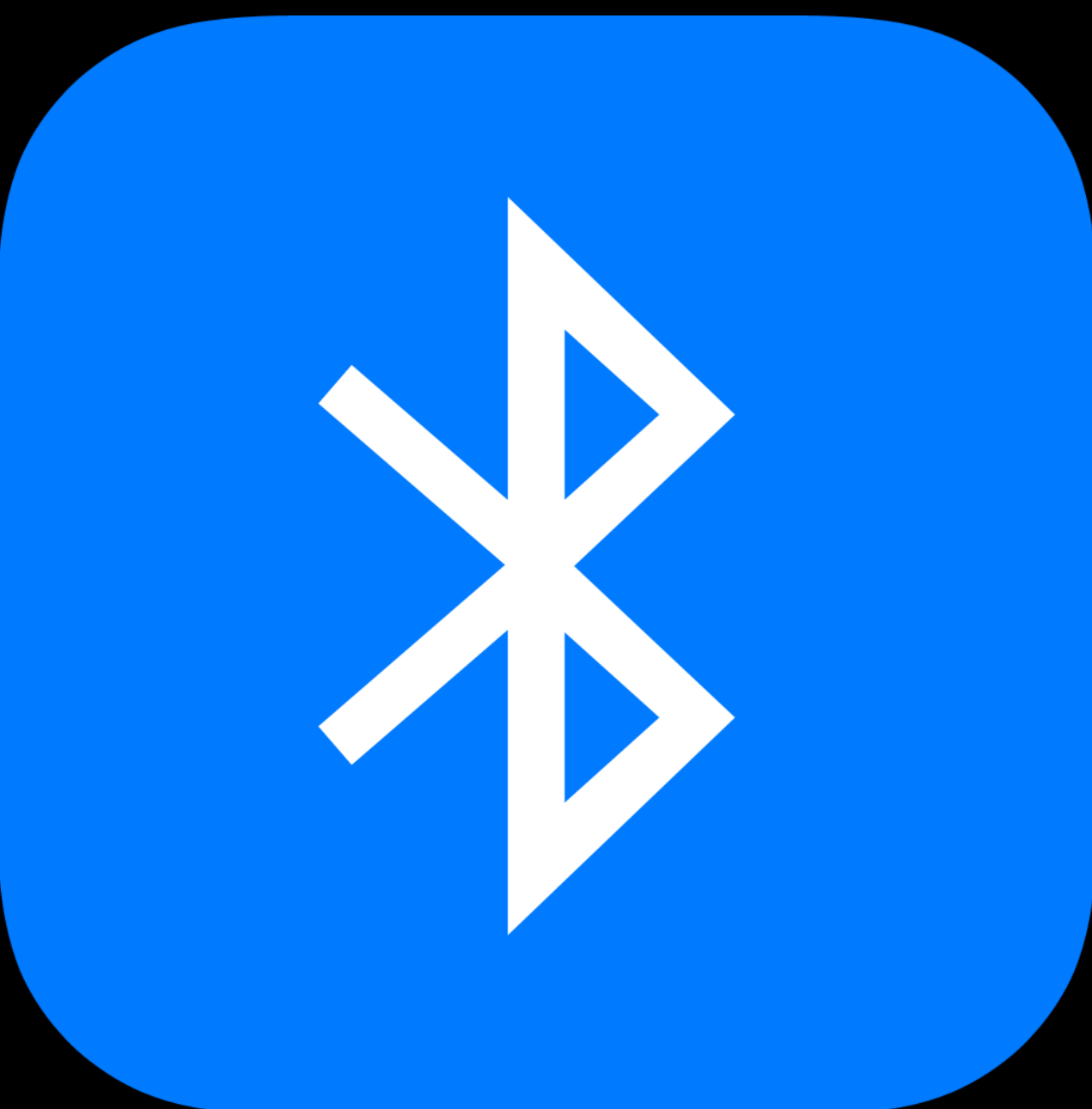
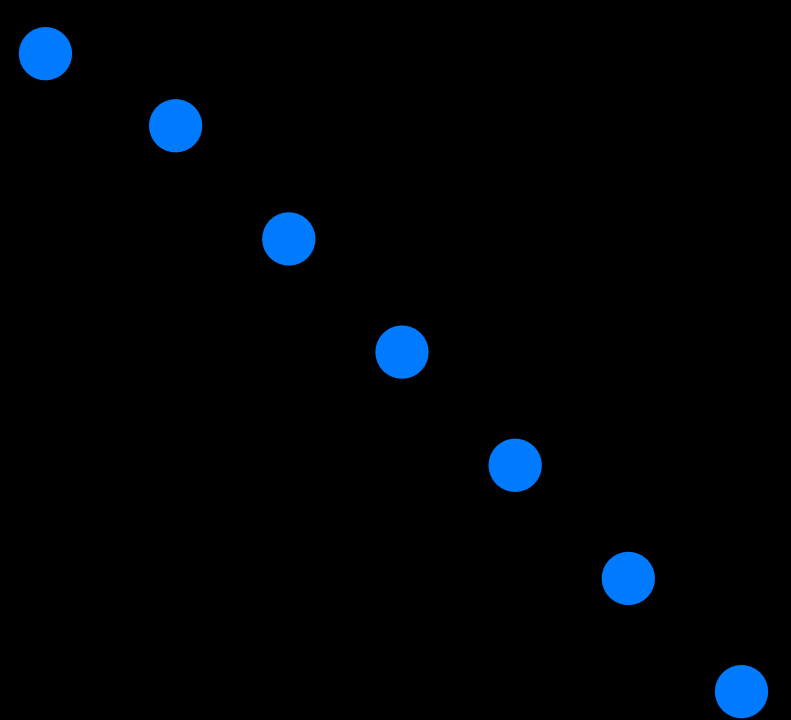


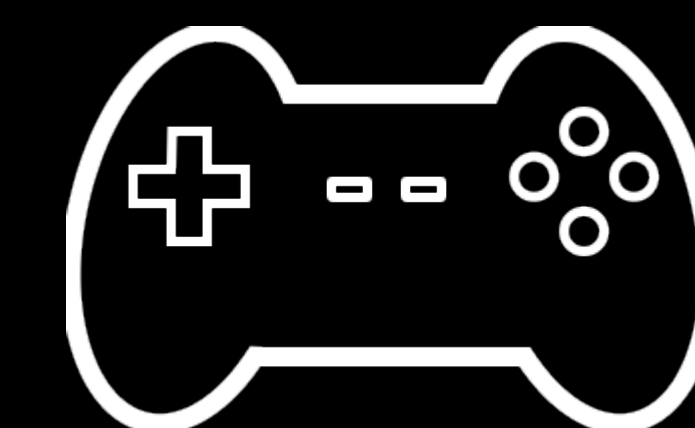
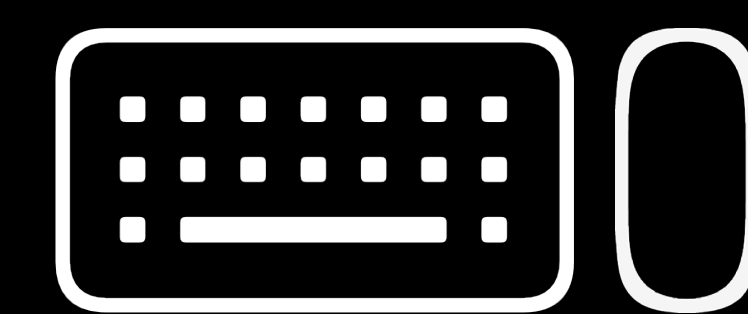
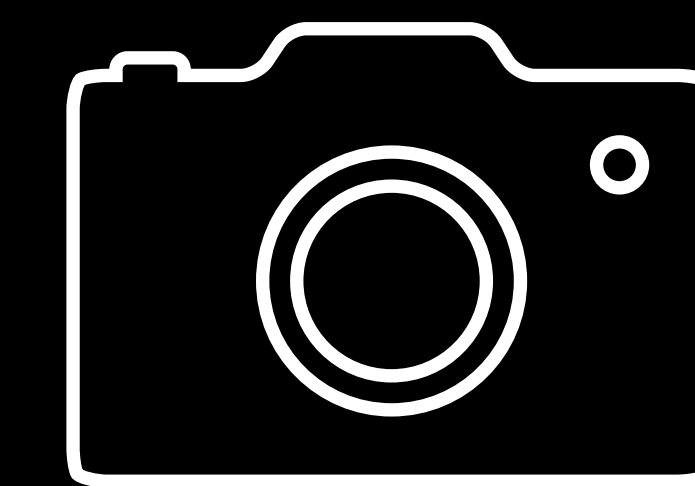
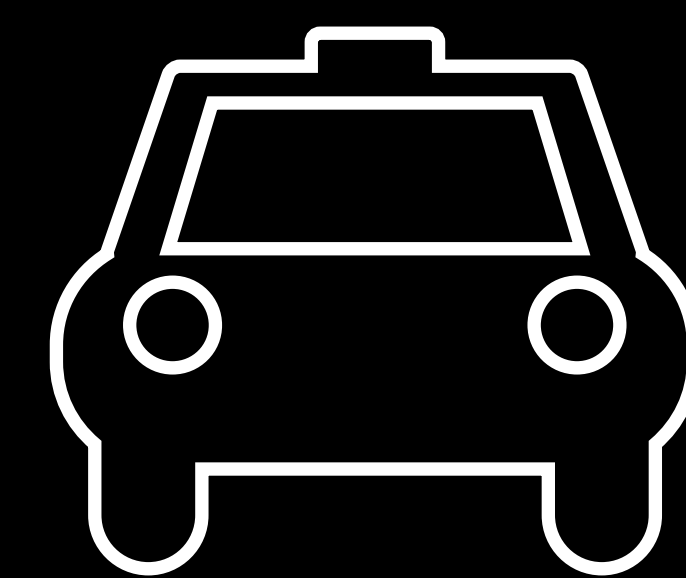
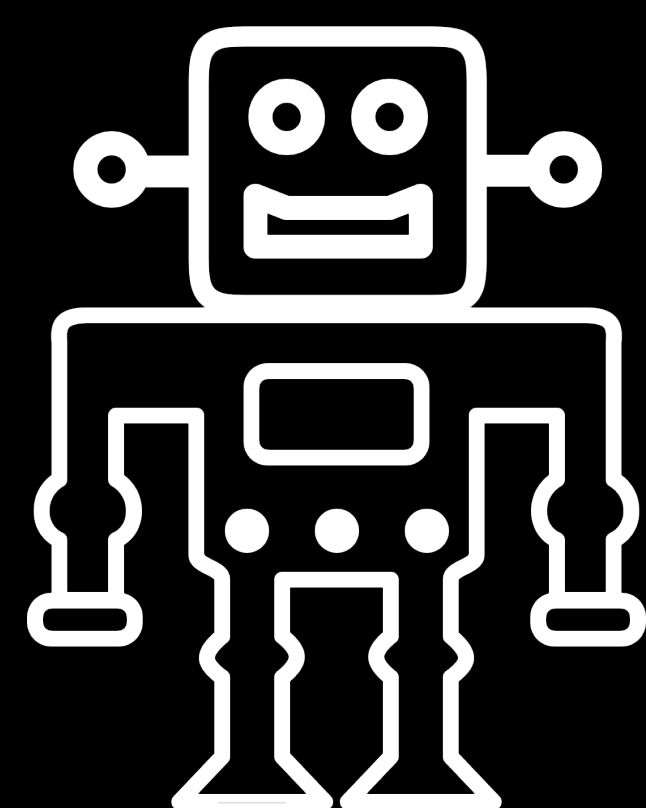
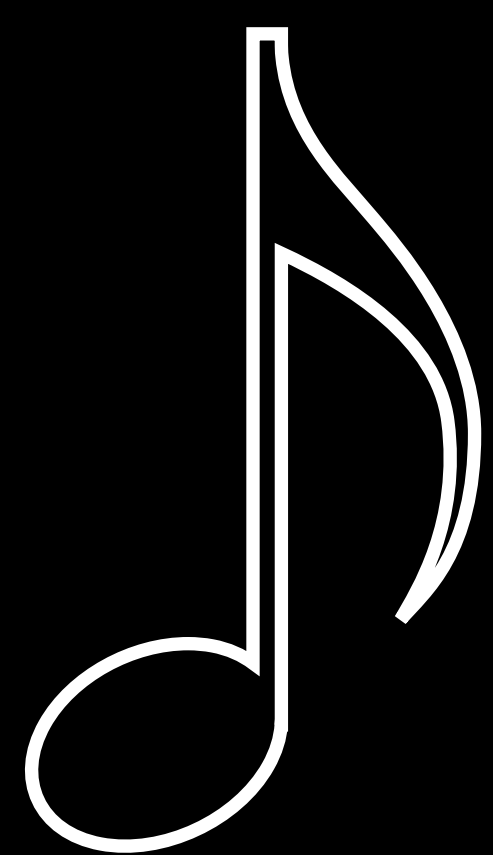
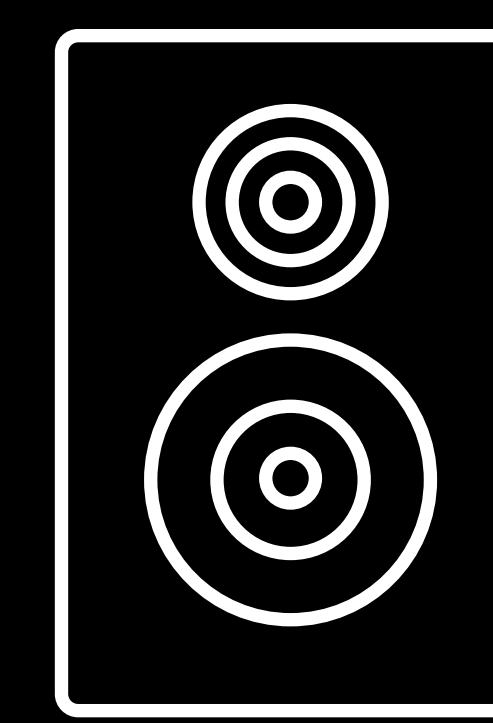


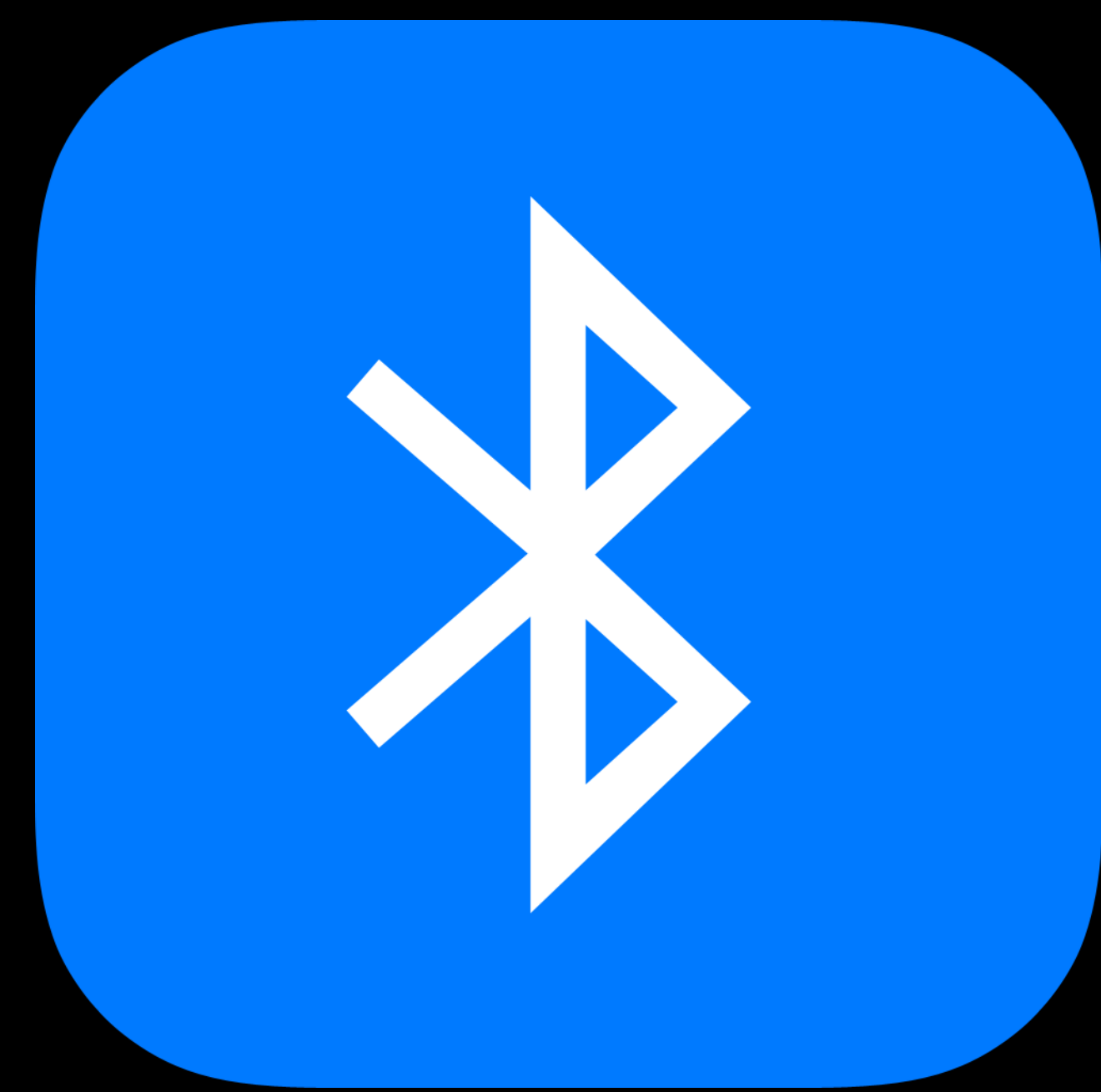


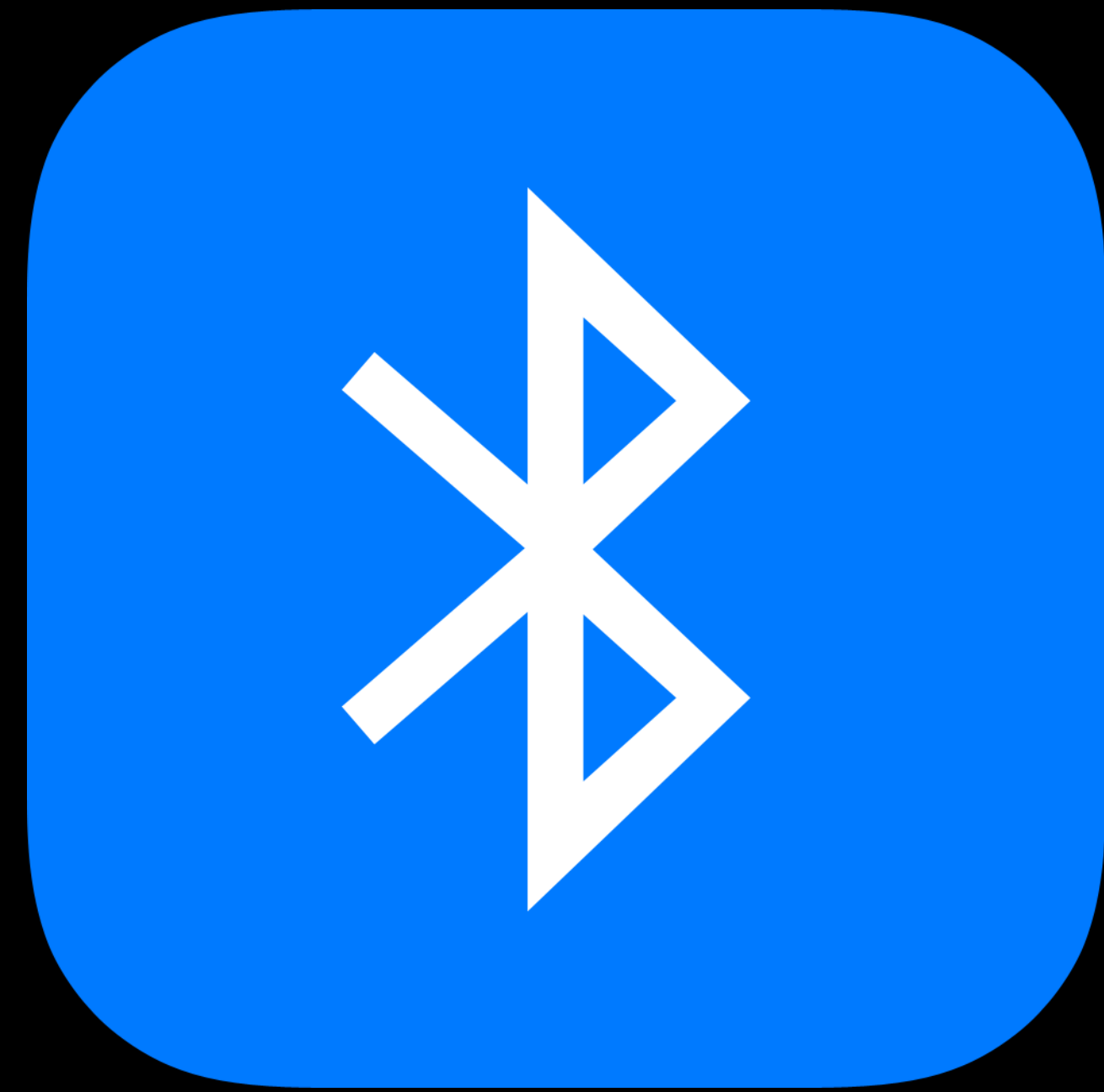


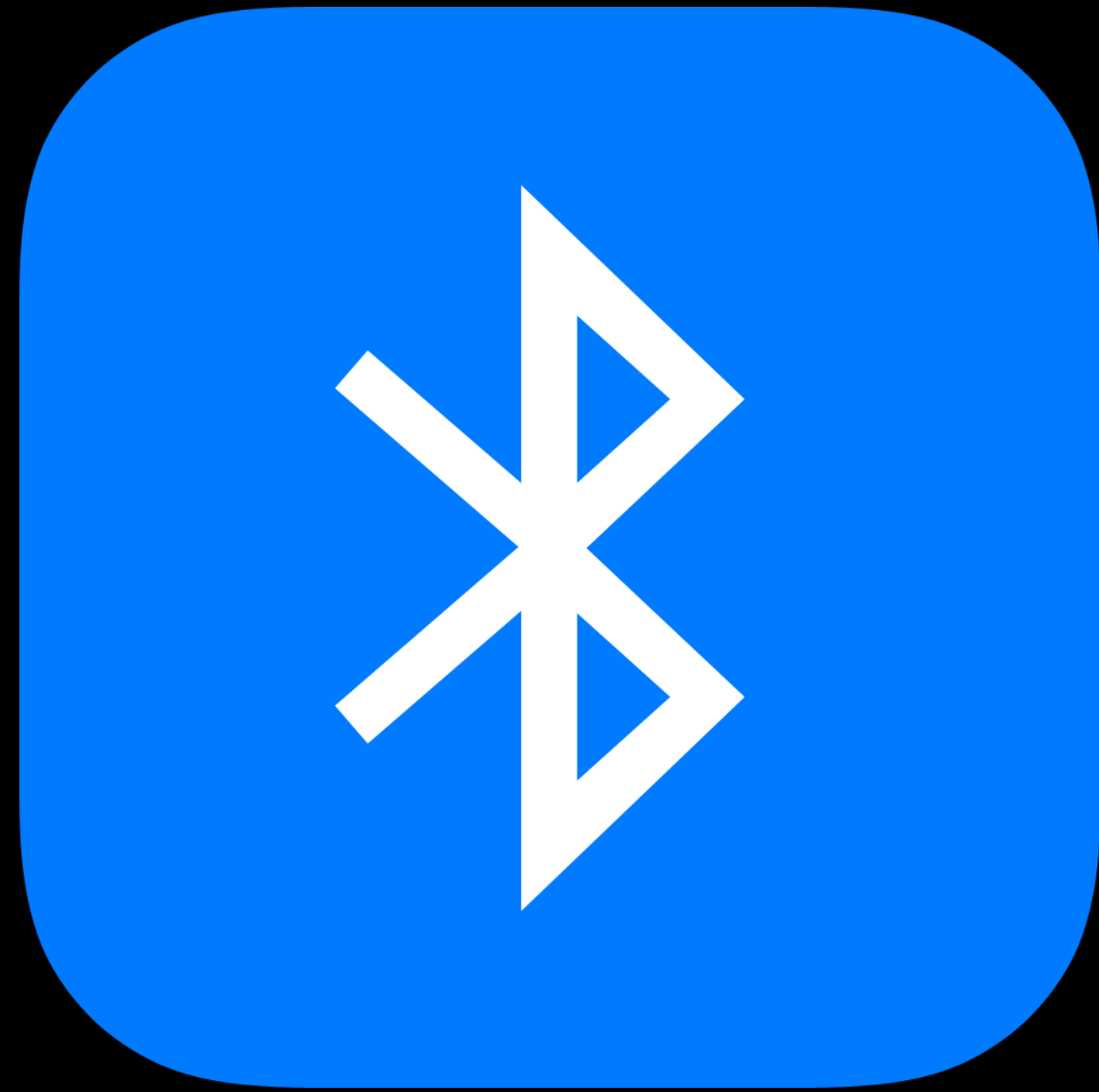


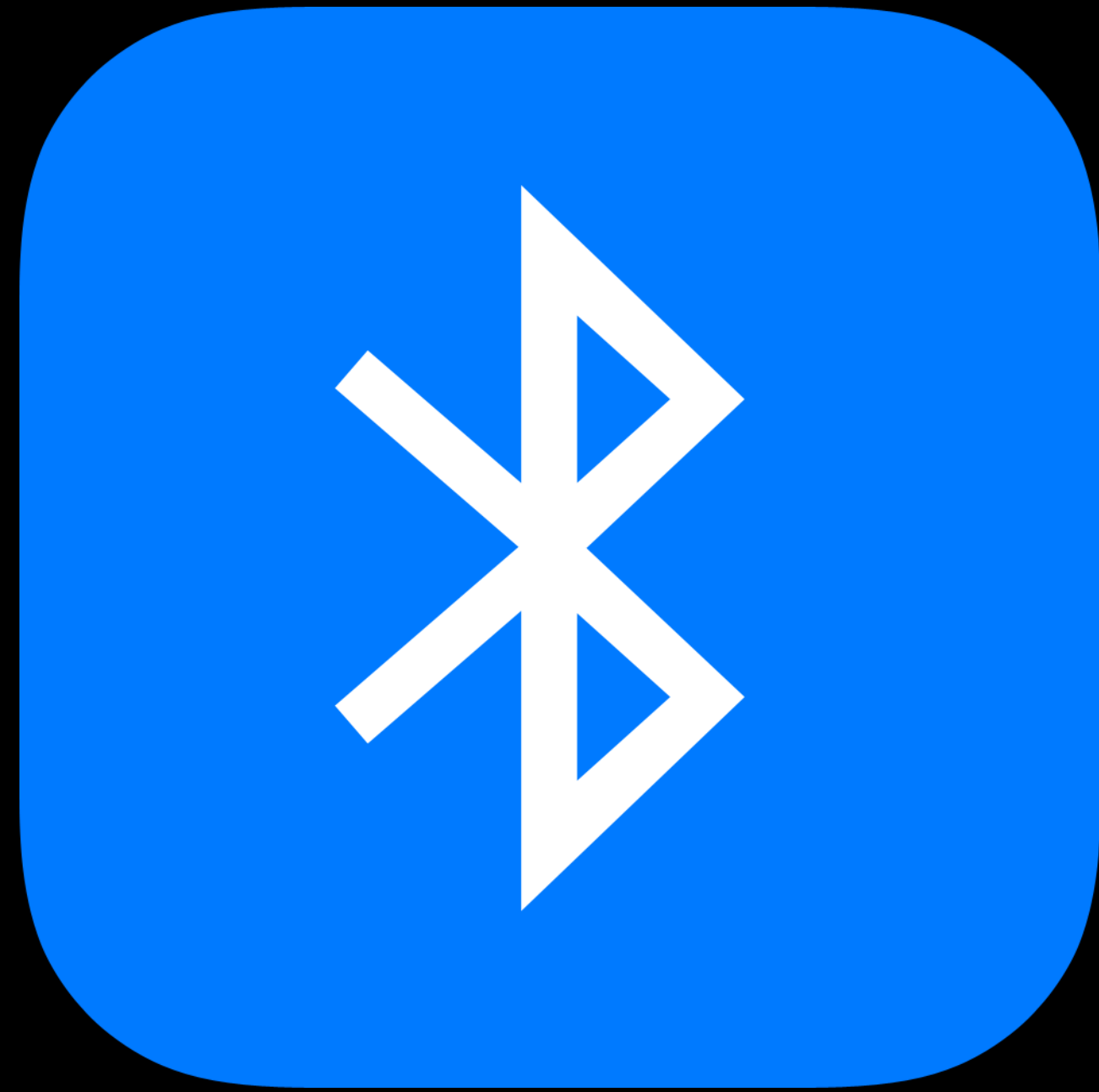












Low Energy 2 Mbps

LE 2 Mbps

NEW

LE 2 Mbps



NEW

New feature in Bluetooth 5.0

LE 2 Mbps



NEW

New feature in Bluetooth 5.0

Physical layer rate increased from 1 Mbps to 2 Mbps

LE 2 Mbps

NEW

New feature in Bluetooth 5.0

Physical layer rate increased from 1 Mbps to 2 Mbps

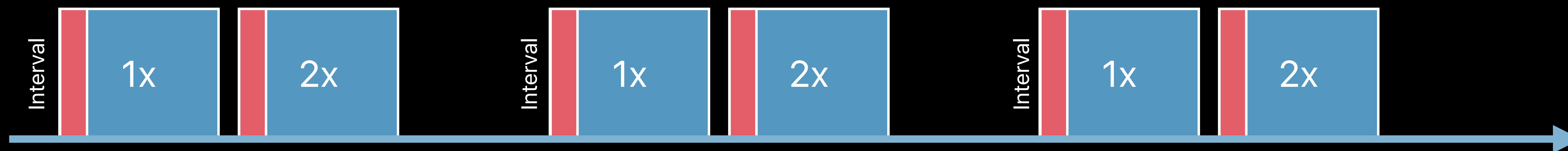


LE 2 Mbps

NEW

New feature in Bluetooth 5.0

Physical layer rate increased from 1 Mbps to 2 Mbps



LE 2 Mbps

NEW

New feature in Bluetooth 5.0

Physical layer rate increased from 1 Mbps to 2 Mbps

Faster and more power efficient connection

LE 2 Mbps

NEW

New feature in Bluetooth 5.0

Physical layer rate increased from 1 Mbps to 2 Mbps

Faster and more power efficient connection

Transparent to the application

LE 2 Mbps



NEW

New feature in Bluetooth 5.0

Physical layer rate increased from 1 Mbps to 2 Mbps

Faster and more power efficient connection

Transparent to the application

Accessories must support LE 2 Mbps

LE 2 Mbps

NEW

New feature in Bluetooth 5.0

Physical layer rate increased from 1 Mbps to 2 Mbps

Faster and more power efficient connection

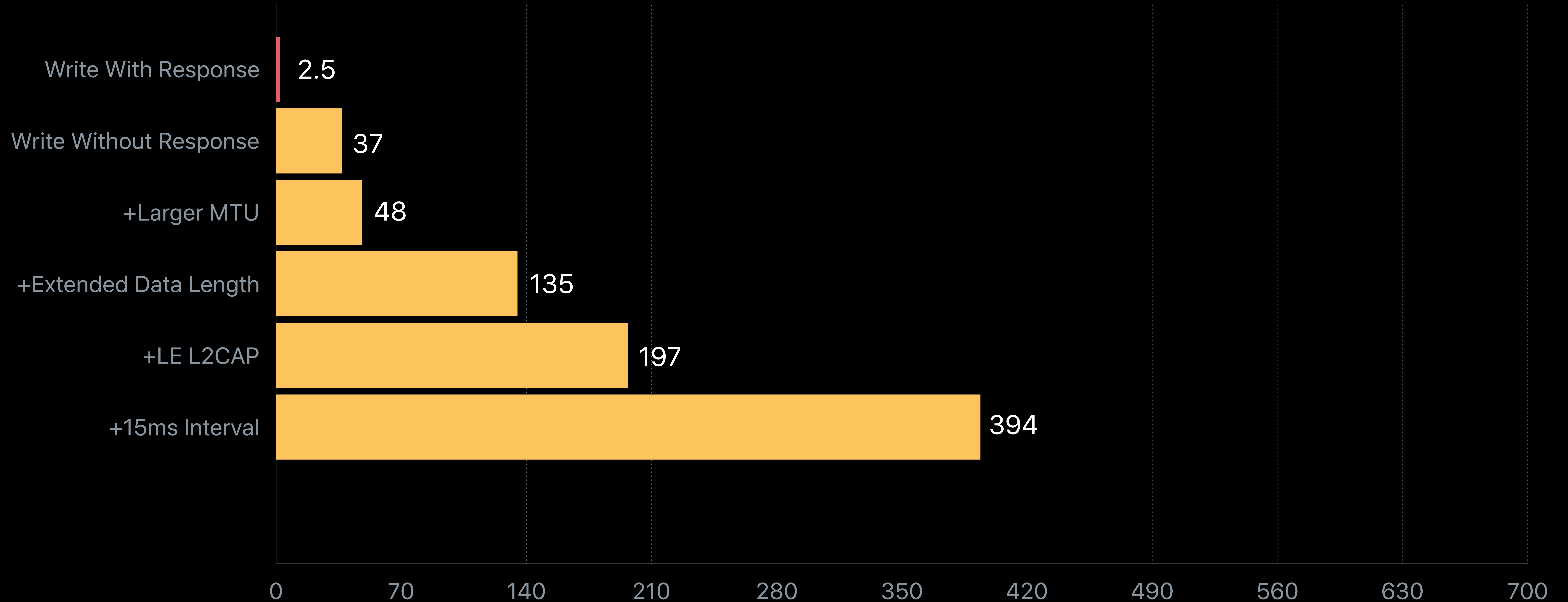
Transparent to the application

Accessories must support LE 2 Mbps

Available starting with iPhone 8, Apple TV 4K, and Apple Watch Series 4

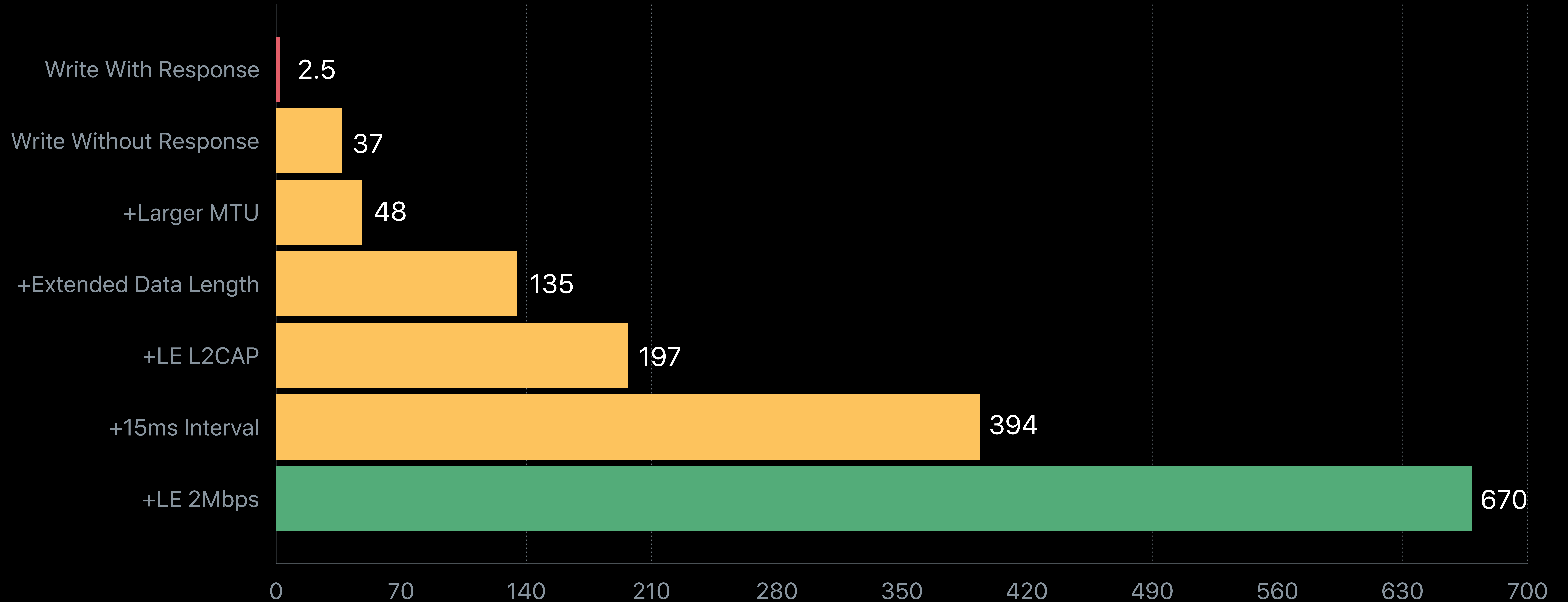
LE 2 Mbps Throughput (kbps)

NEW



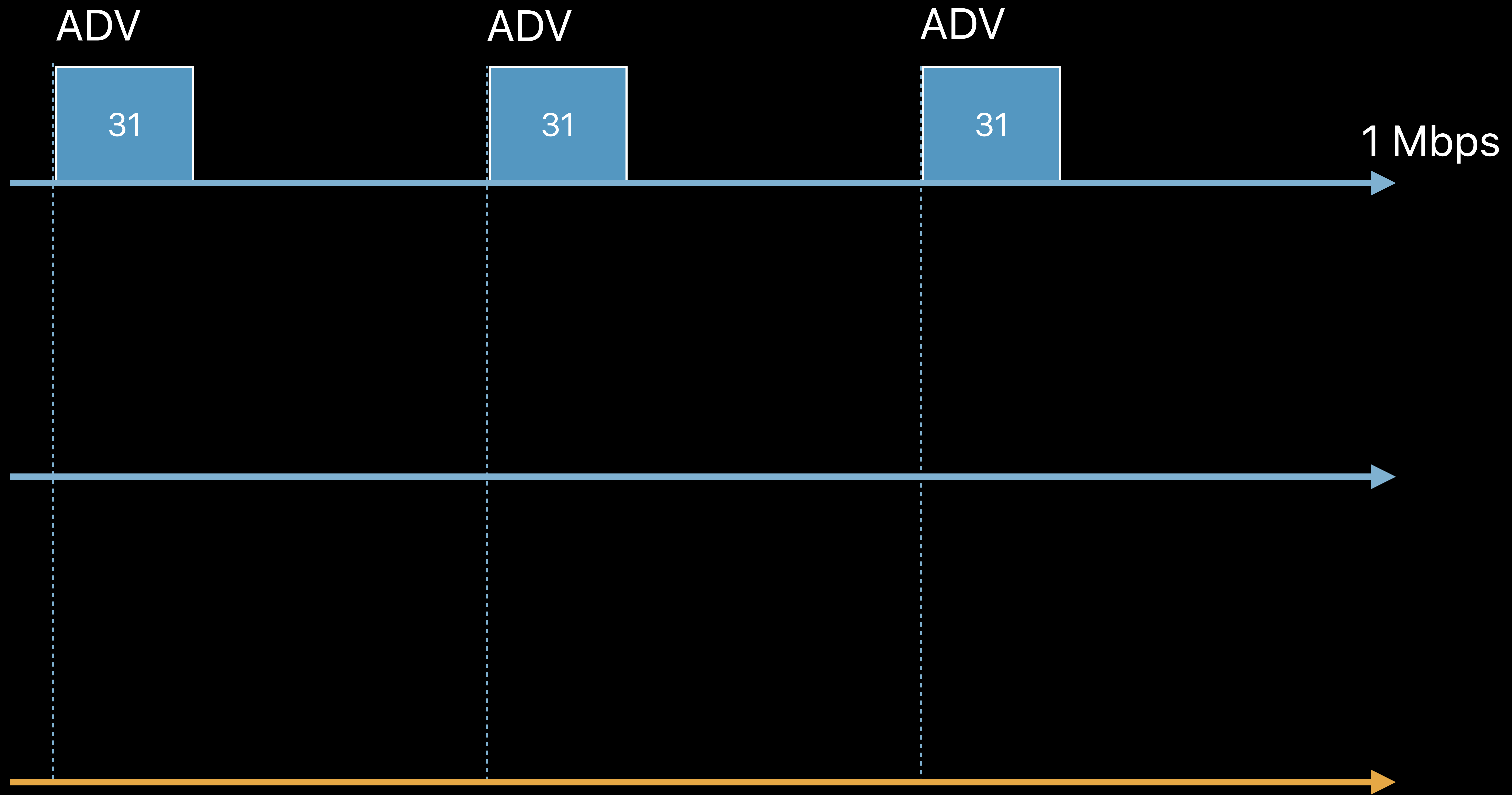
LE 2 Mbps Throughput (kbps)

NEW

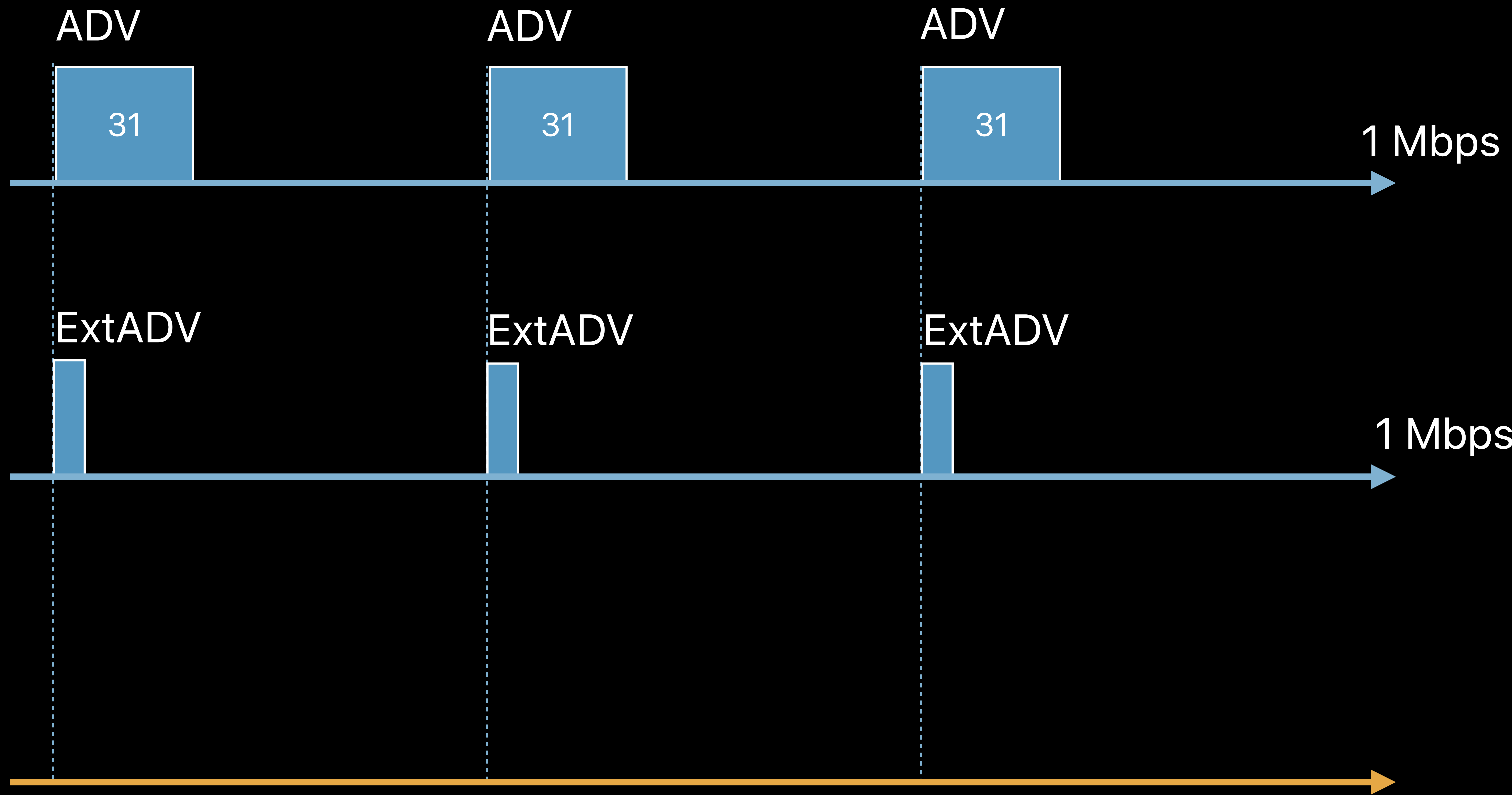


Advertising Extensions

Advertising Extensions

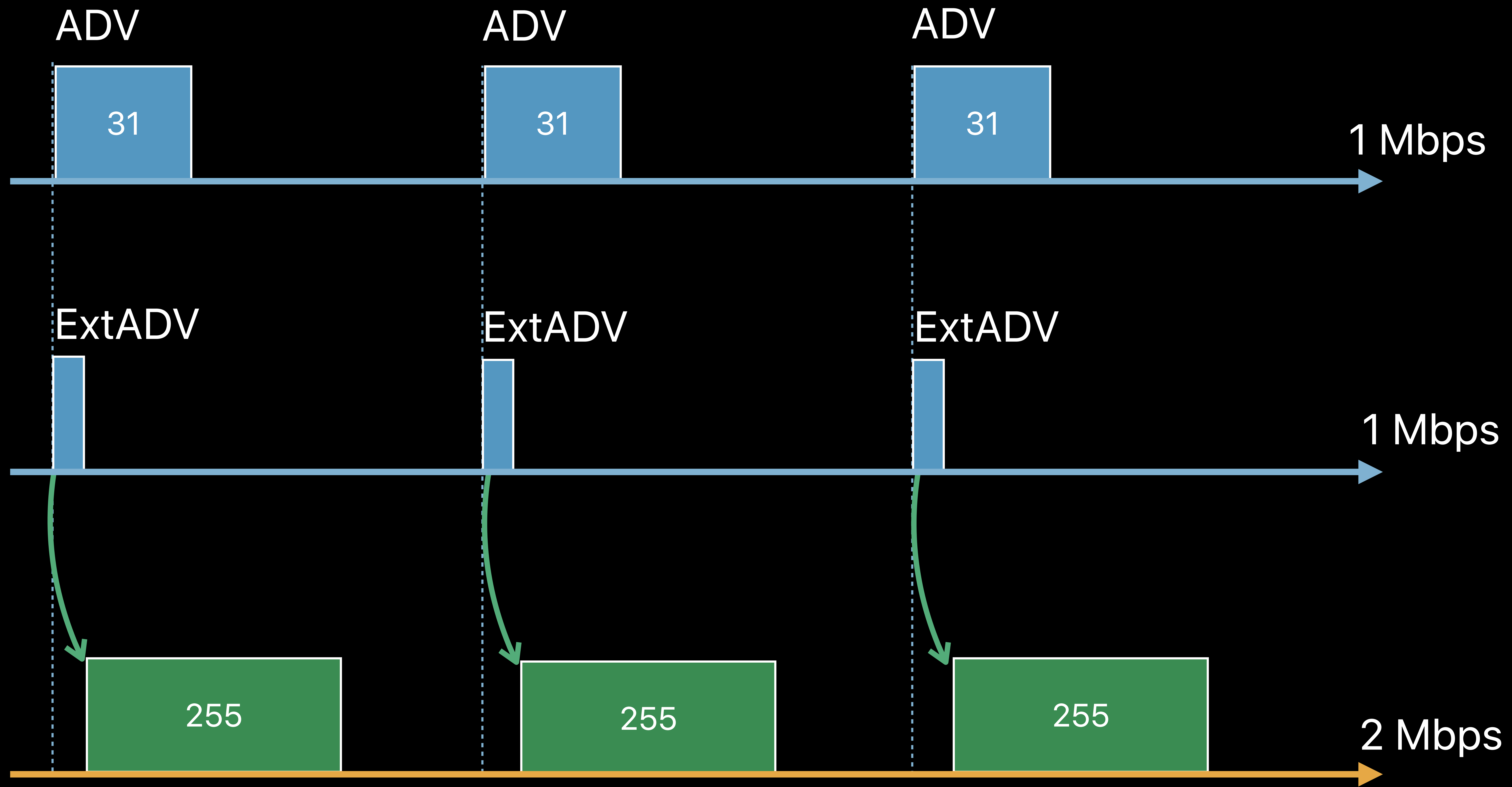


Advertising Extensions



Advertising Extensions

NEW



Extended Scan

NEW

Extended Scan

NEW

Scans for extended advertisements

Extended Scan



NEW

Scans for extended advertisements

Accessories must support extended advertisements with LE 2 Mbps

Extended Scan



NEW

Scans for extended advertisements

Accessories must support extended advertisements with LE 2 Mbps

Support extended advertisement payload up to 124 bytes

Extended Scan

NEW

Scans for extended advertisements

Accessories must support extended advertisements with LE 2 Mbps

Support extended advertisement payload up to 124 bytes

4 times the advertisement data that an accessory can send today

Extended Scan

NEW

Scans for extended advertisements

Accessories must support extended advertisements with LE 2 Mbps

Support extended advertisement payload up to 124 bytes

4 times the advertisement data that an accessory can send today

Transparent to application

Extended Scan

NEW

Scans for extended advertisements

Accessories must support extended advertisements with LE 2 Mbps

Support extended advertisement payload up to 124 bytes

4 times the advertisement data that an accessory can send today

Transparent to application

New API to query for platform support

```
class func supports(_ features: CBCentralManager.Feature) -> Bool
static var extendedScanAndConnect: CBCentralManager.Feature { get }
```

Extended Connections

NEW

Extended Connections

NEW

Supports connections to connectable extended advertisements

Extended Connections



NEW

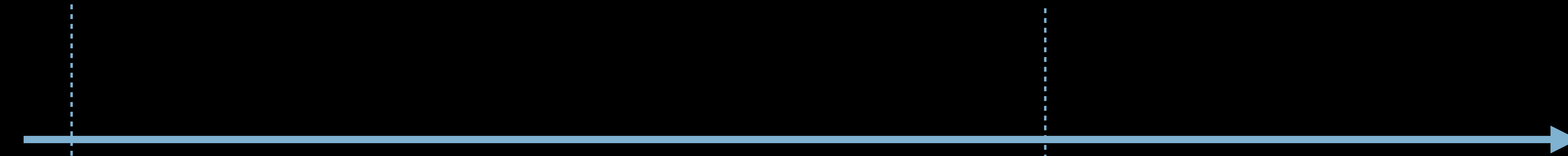
Supports connections to connectable extended advertisements

Improves existing connection exchange protocol

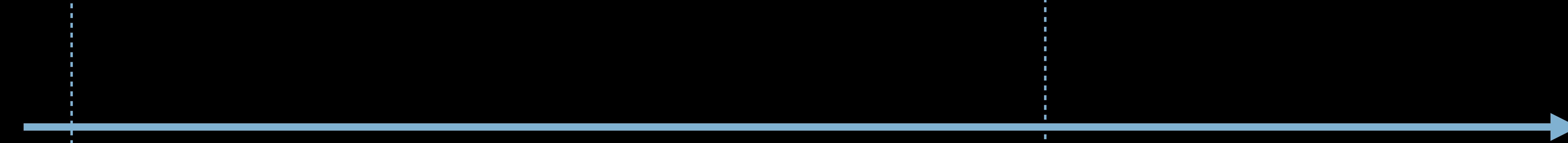
Legacy Connections

NEW

Advertiser



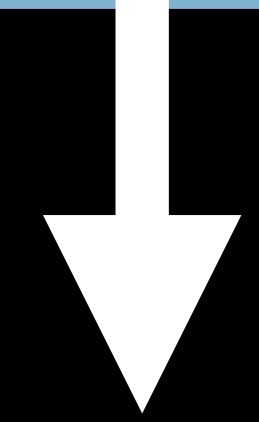
Scanner



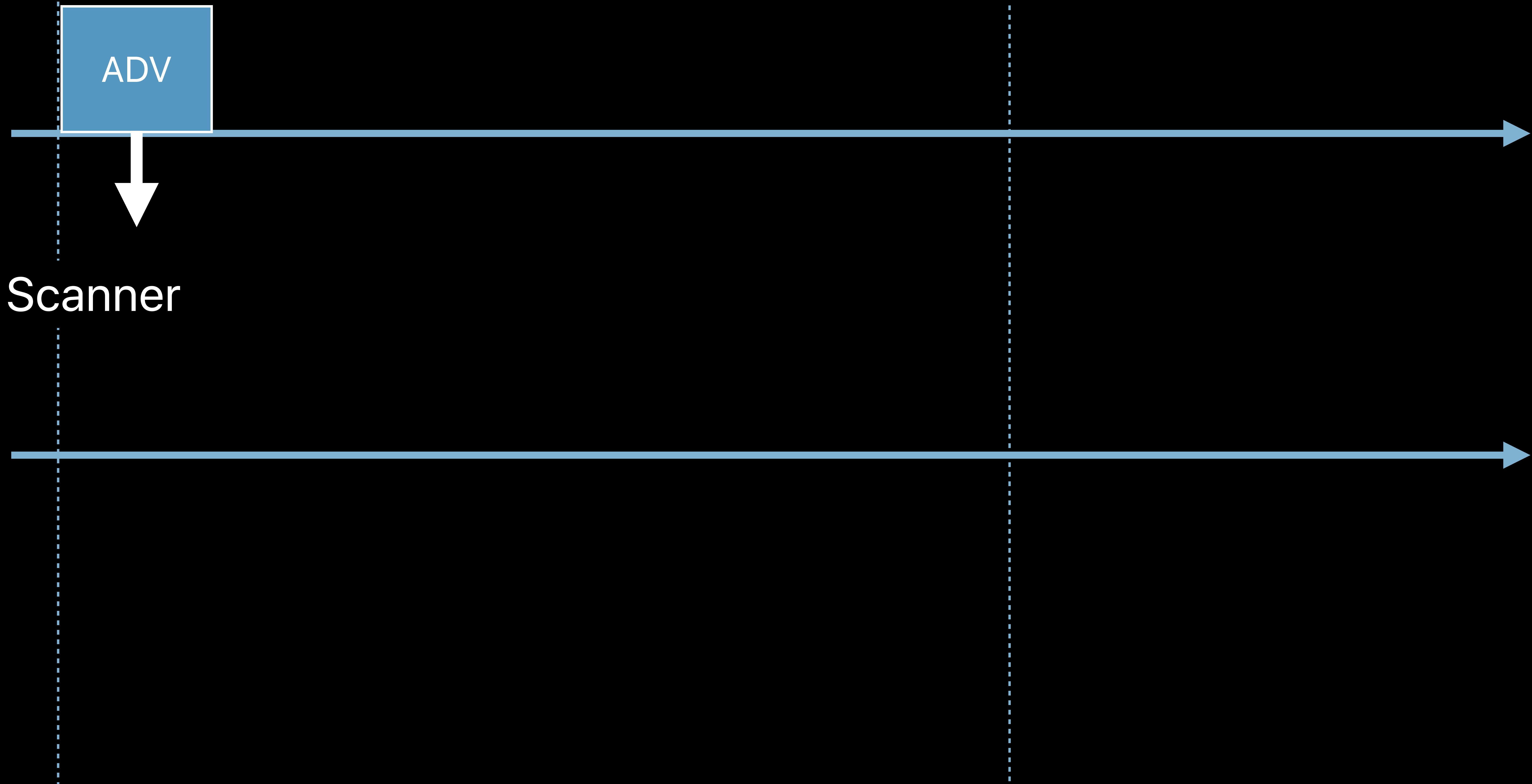
Legacy Connections

NEW

Advertiser



Scanner



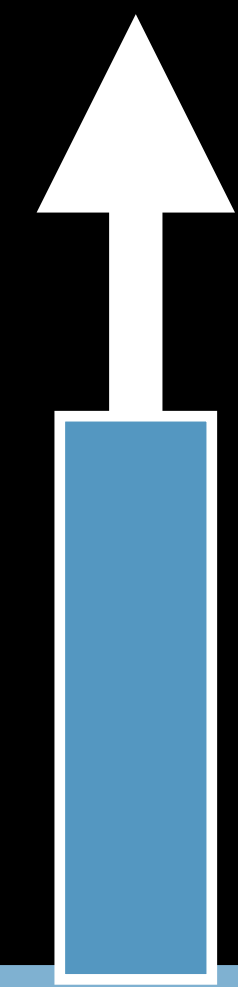
Legacy Connections

NEW

Advertiser



Scanner



Conn_Indication



Legacy Connections

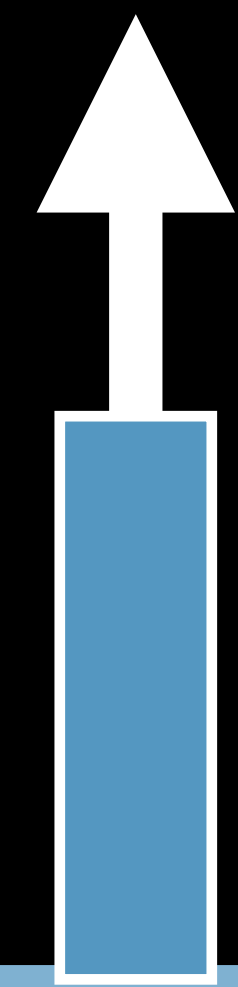


Advertiser



X No ACK

Scanner



Conn_Indication



Legacy Connections

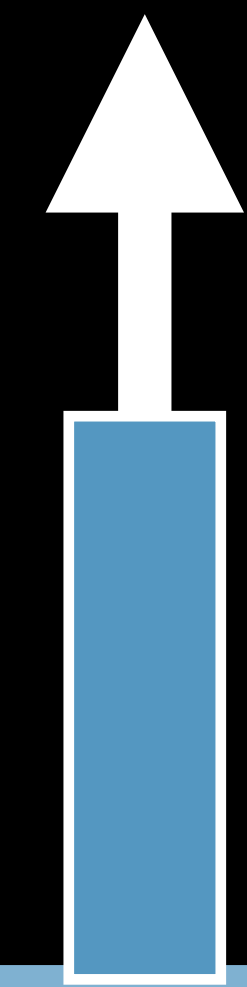


Advertiser



X No ACK

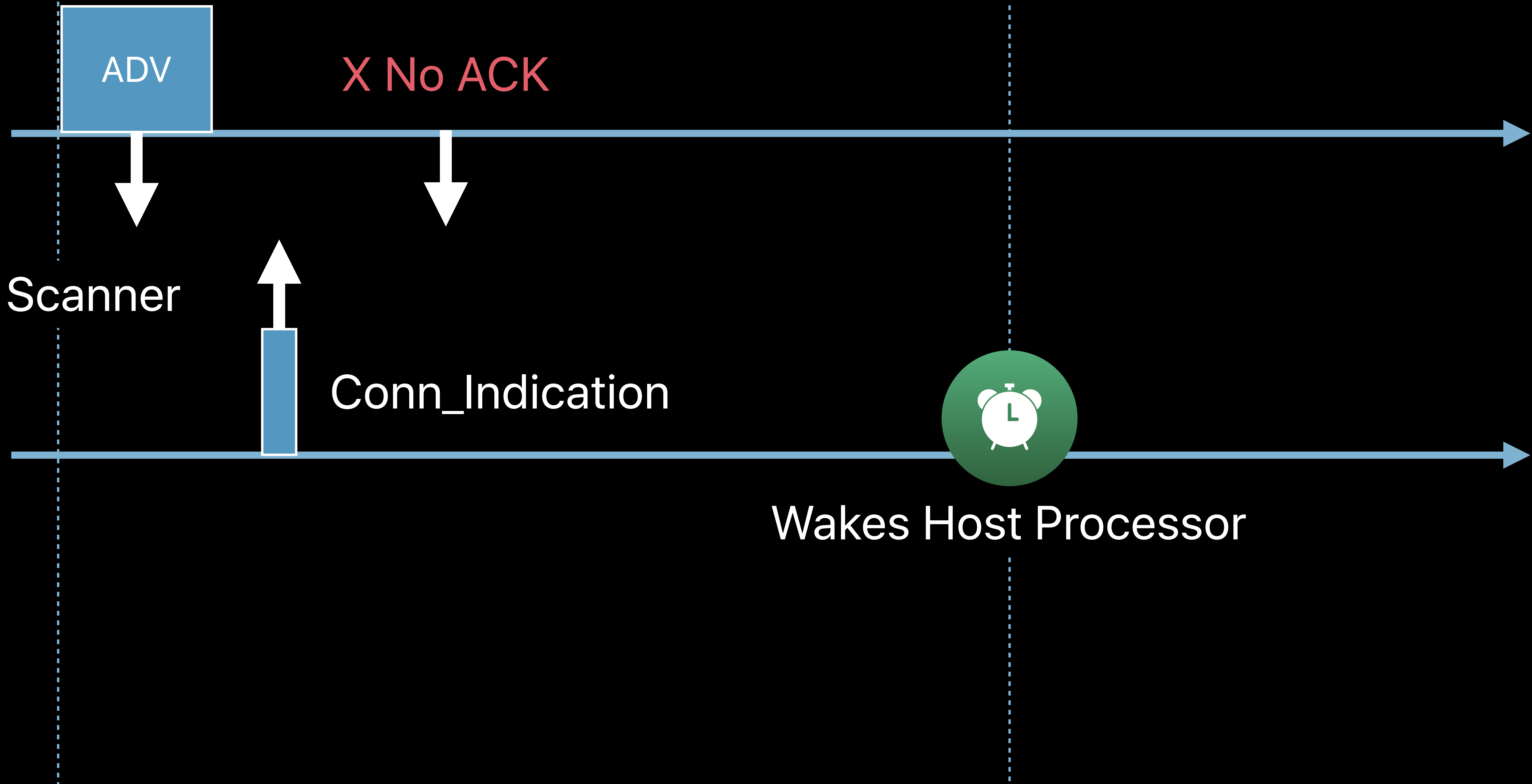
Scanner



Conn_Indication

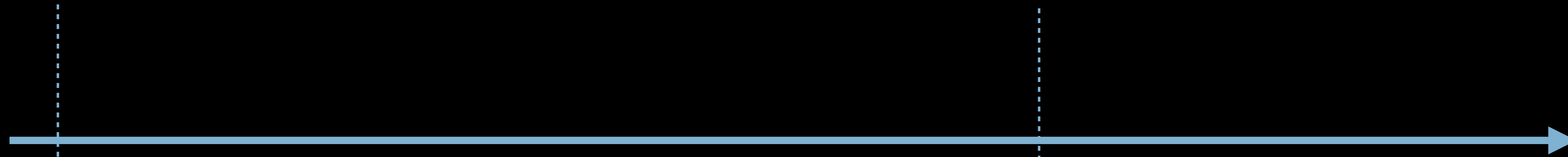


Wakes Host Processor



Extended Connections

Advertiser



Advertiser

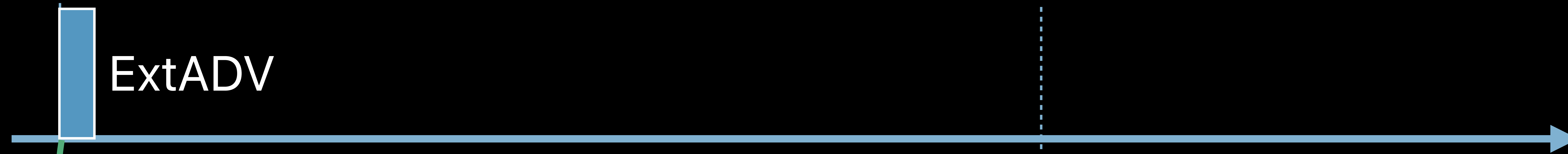


Scanner



Extended Connections

Advertiser



Advertiser

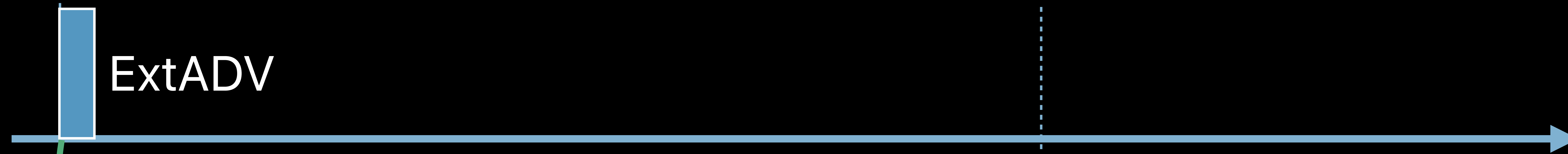


Scanner



Extended Connections

Advertiser



Advertiser

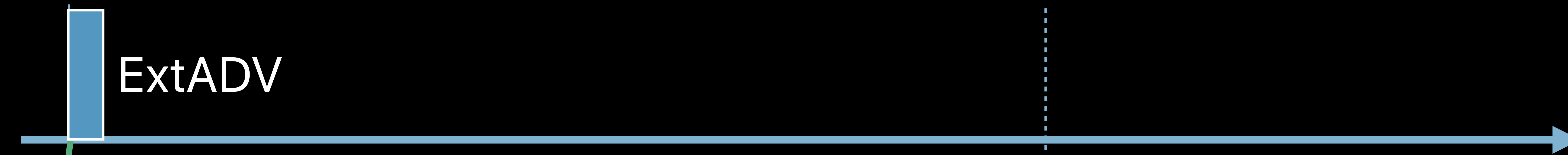


Scanner



Extended Connections

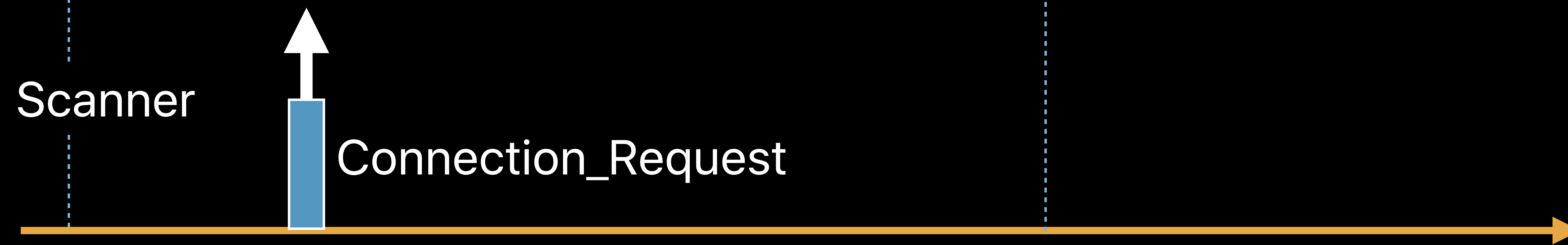
Advertiser



Advertiser

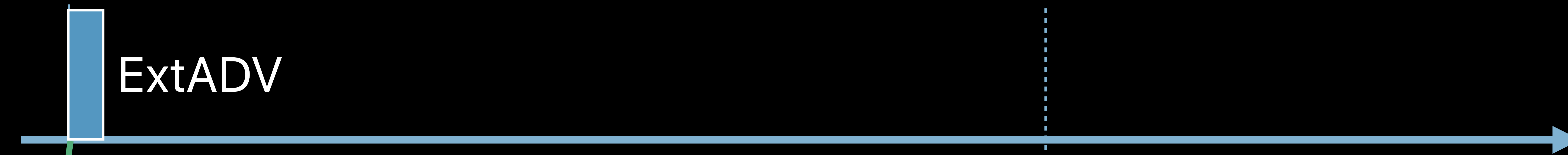


Scanner



Extended Connections

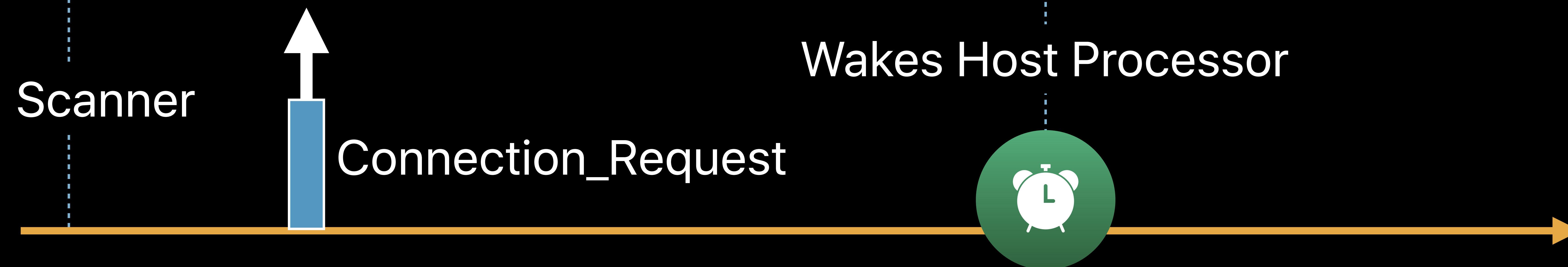
Advertiser



Advertiser

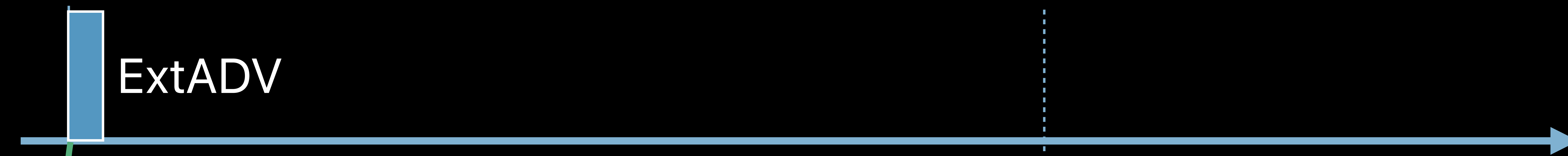


Scanner



Extended Connections

Advertiser



Advertiser



Scanner



Extended Connections

NEW

Supports connections to peripheral connectable extended advertisements

Improves on connection protocol exchange

More robust and power efficient

Extended Connections

NEW

Supports connections to peripheral connectable extended advertisements

Improves on connection protocol exchange

More robust and power efficient

Transparent to application

Extended Connections

NEW

Supports connections to peripheral connectable extended advertisements

Improves on connection protocol exchange

More robust and power efficient

Transparent to application

Accessories must support connectable extended advertisements

Extended Connections

NEW

Supports connections to peripheral connectable extended advertisements

Improves on connection protocol exchange

More robust and power efficient

Transparent to application

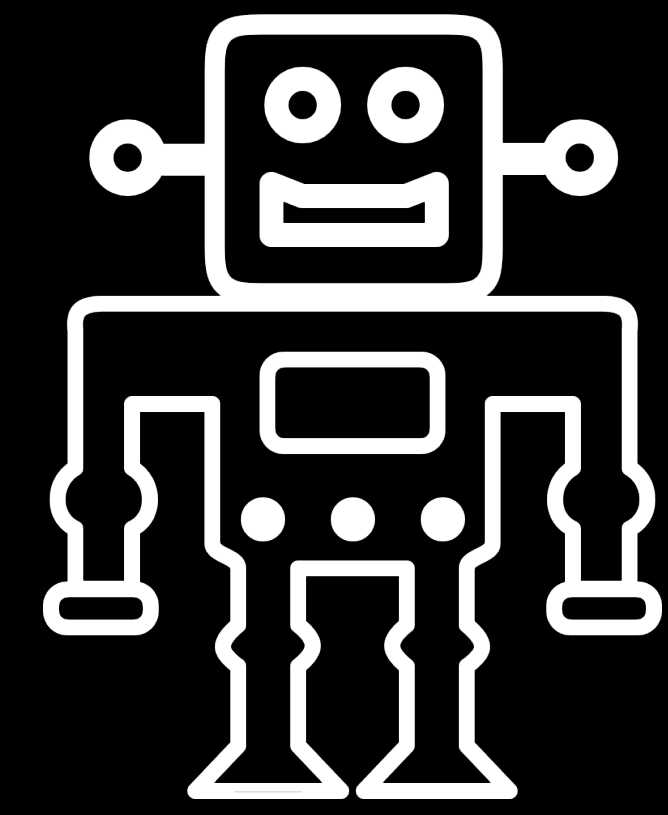
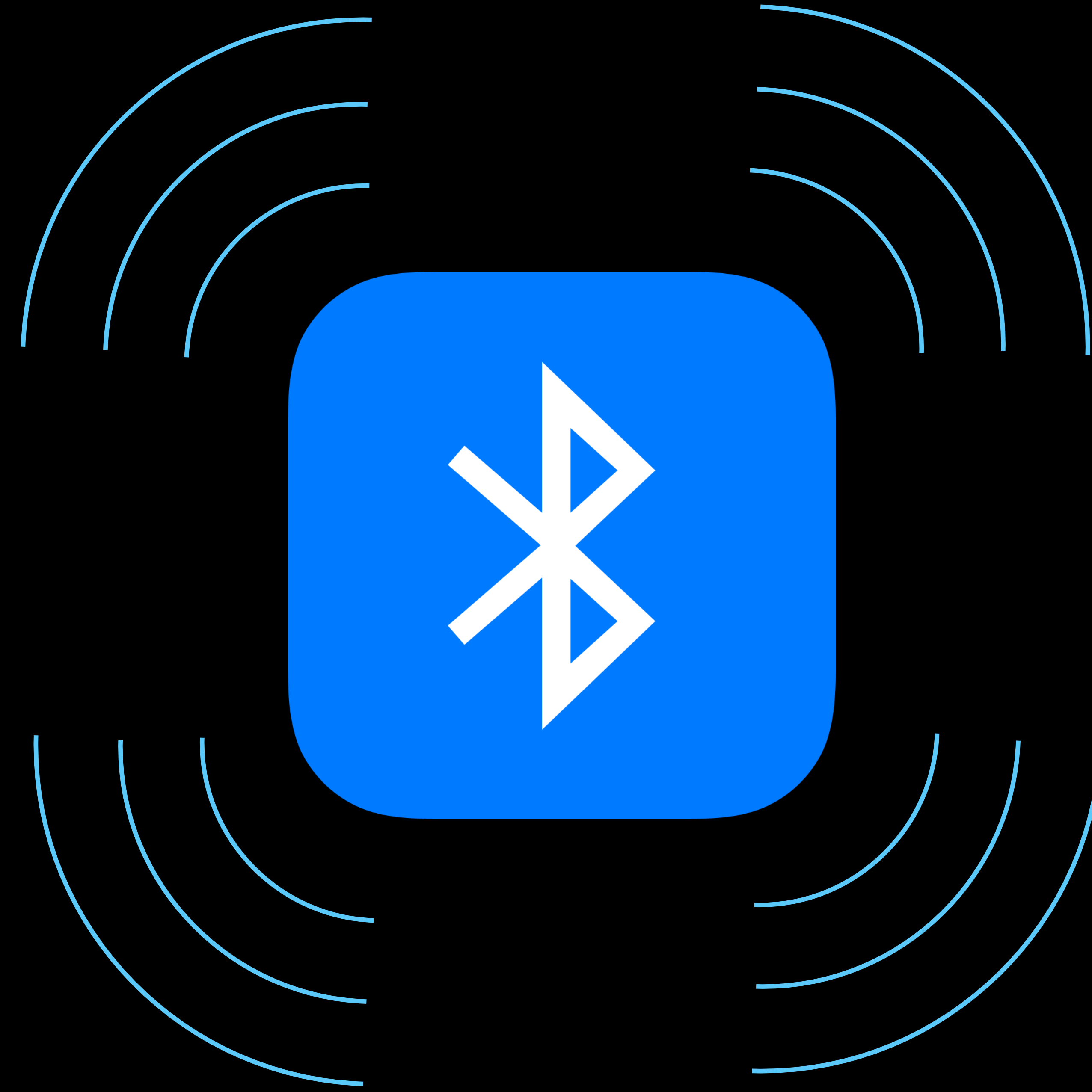
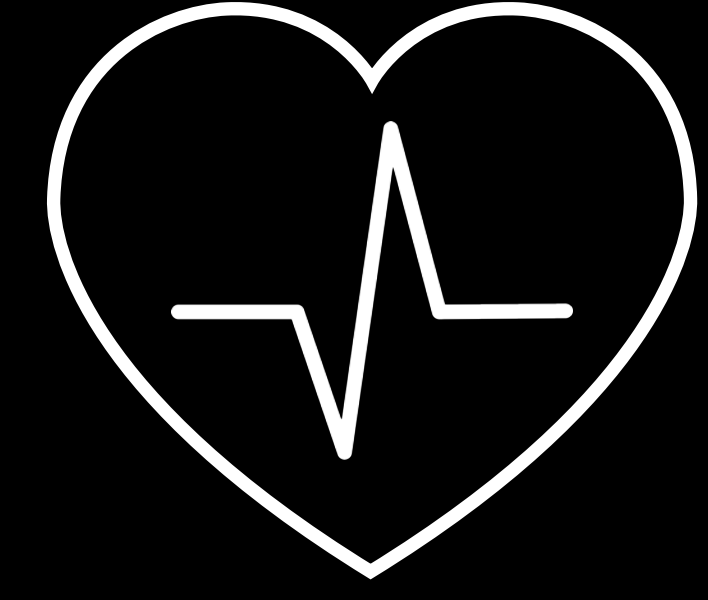
Accessories must support connectable extended advertisements

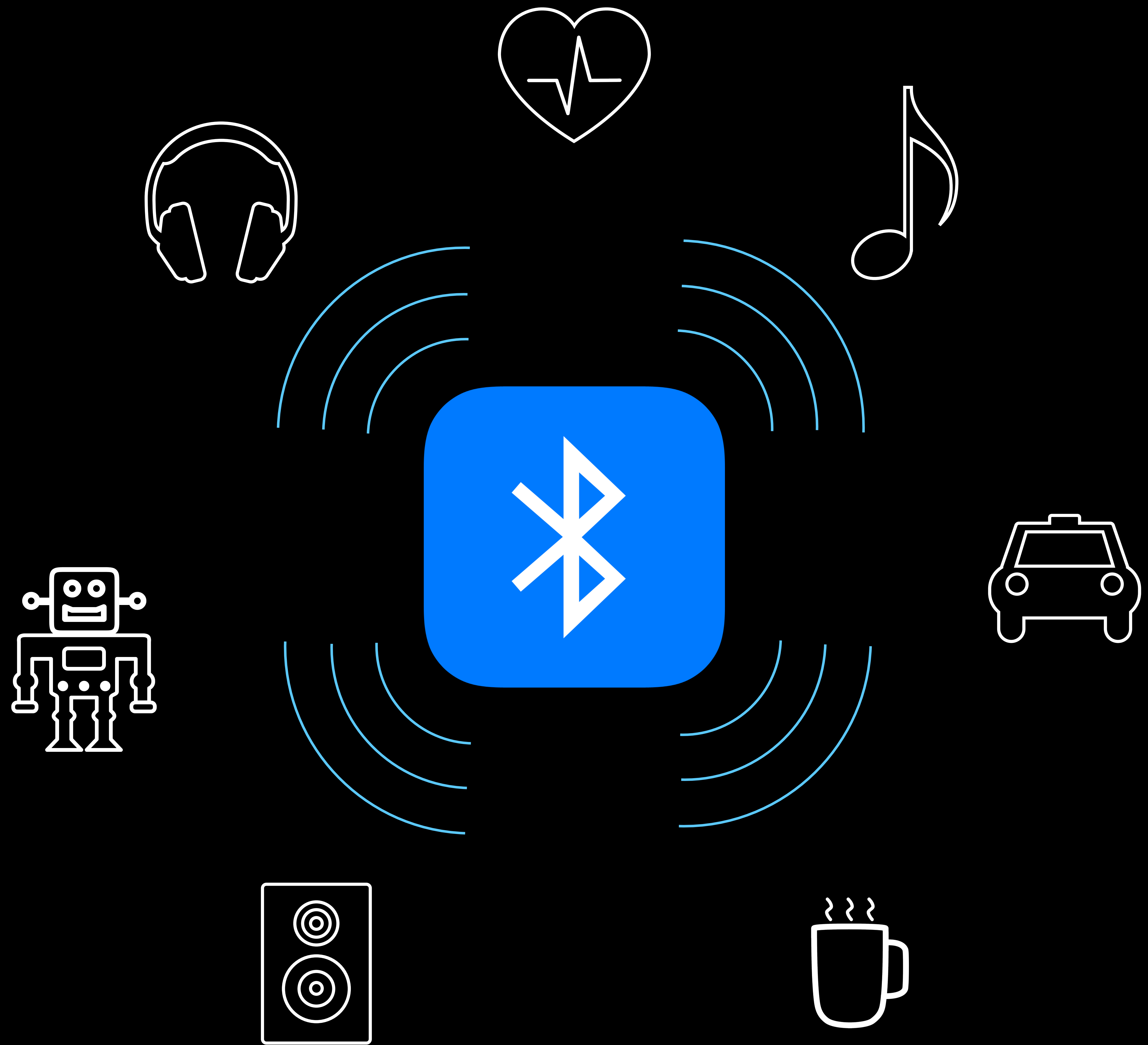
New API to query for platform support

```
class func supports(_ features: CBCentralManager.Feature) -> Bool
static var extendedScanAndConnect: CBCentralManager.Feature { get }
```

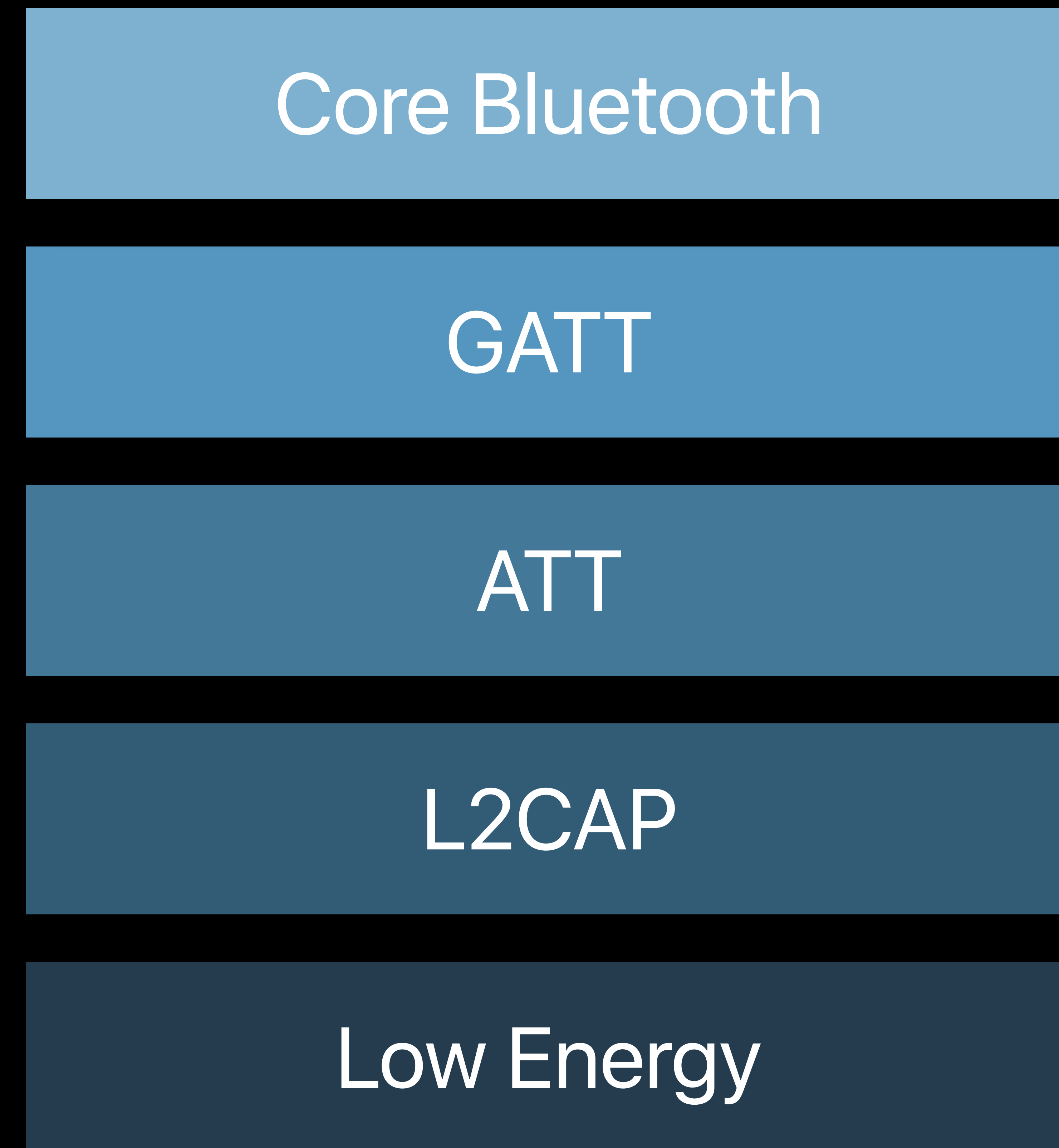
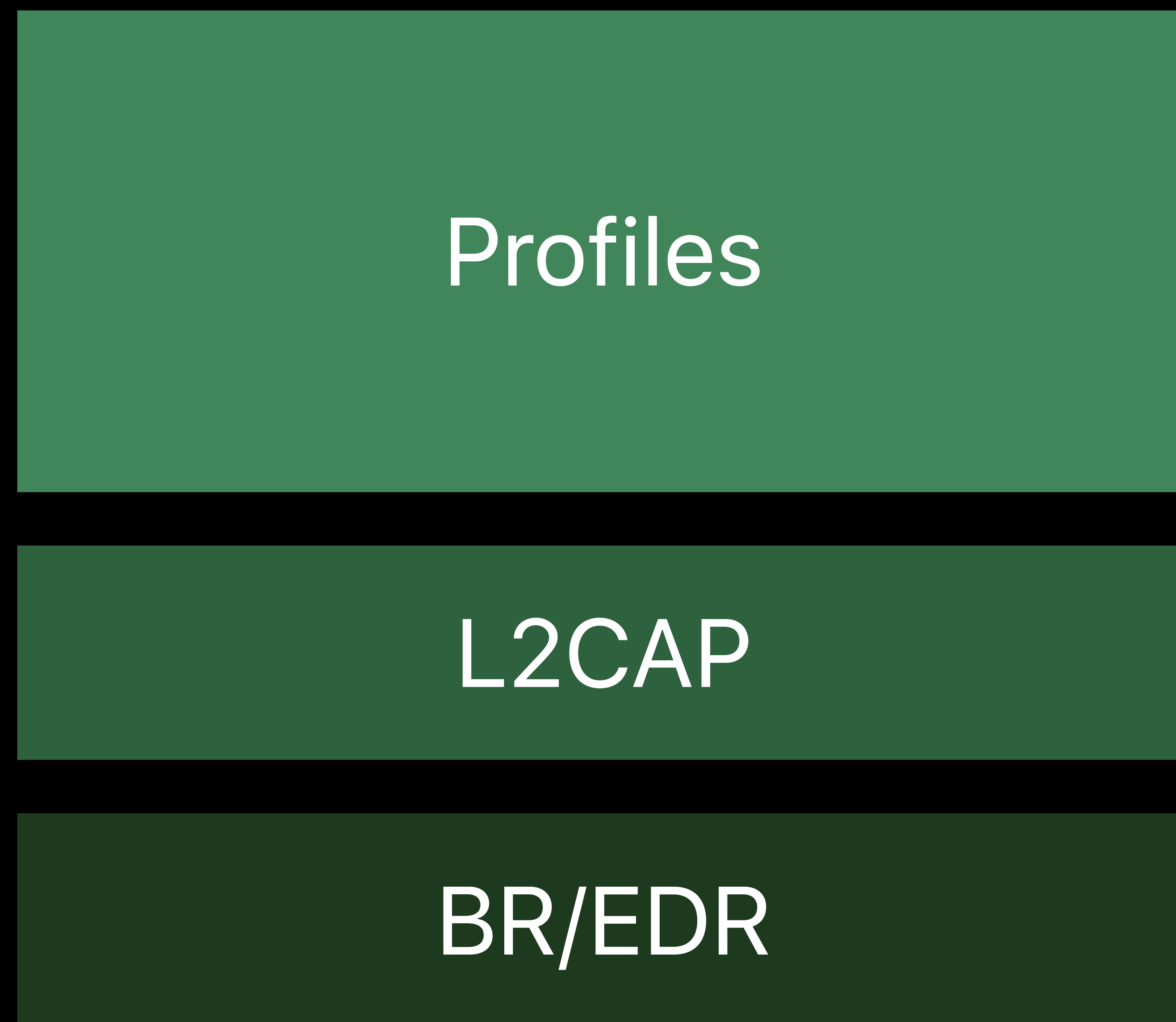
Core Bluetooth for BR/EDR

Yilok Wong, Bluetooth Engineer

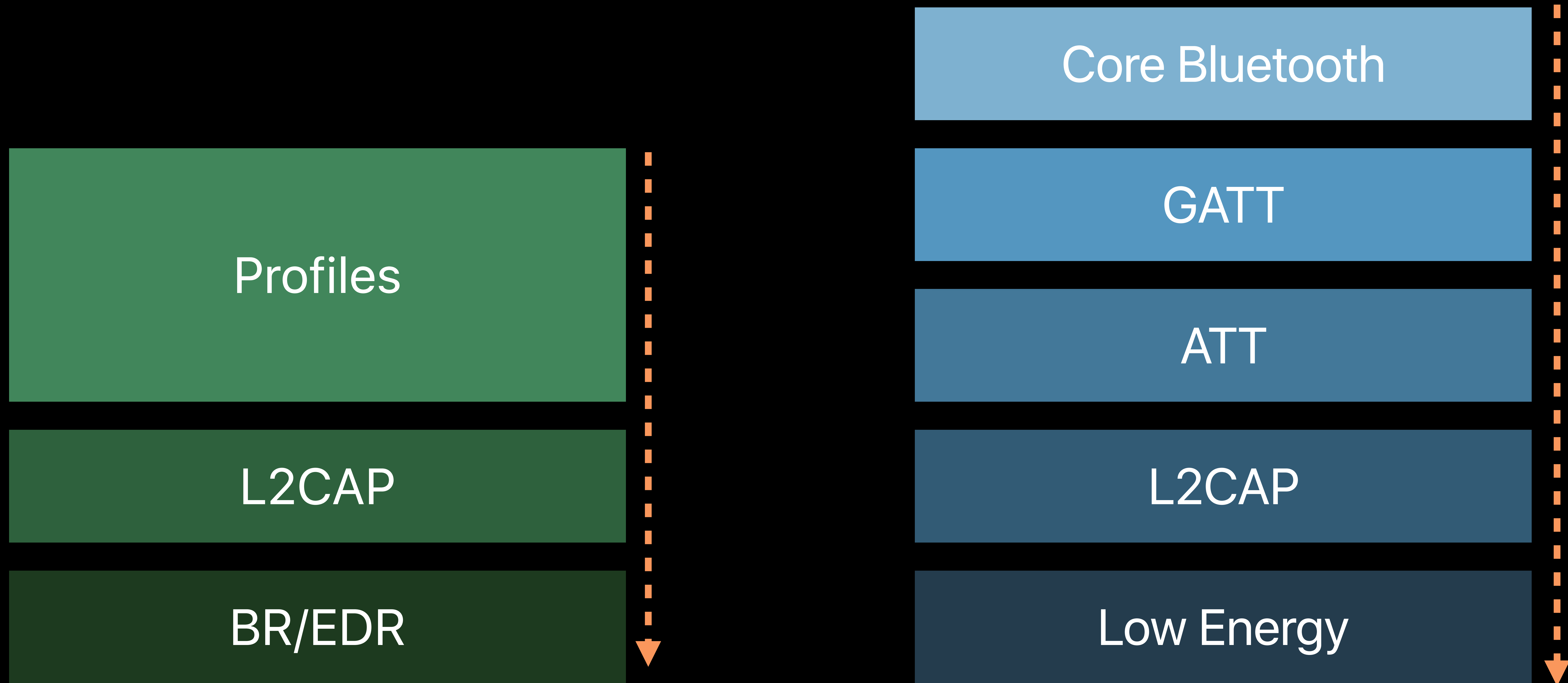




Core Bluetooth 2018

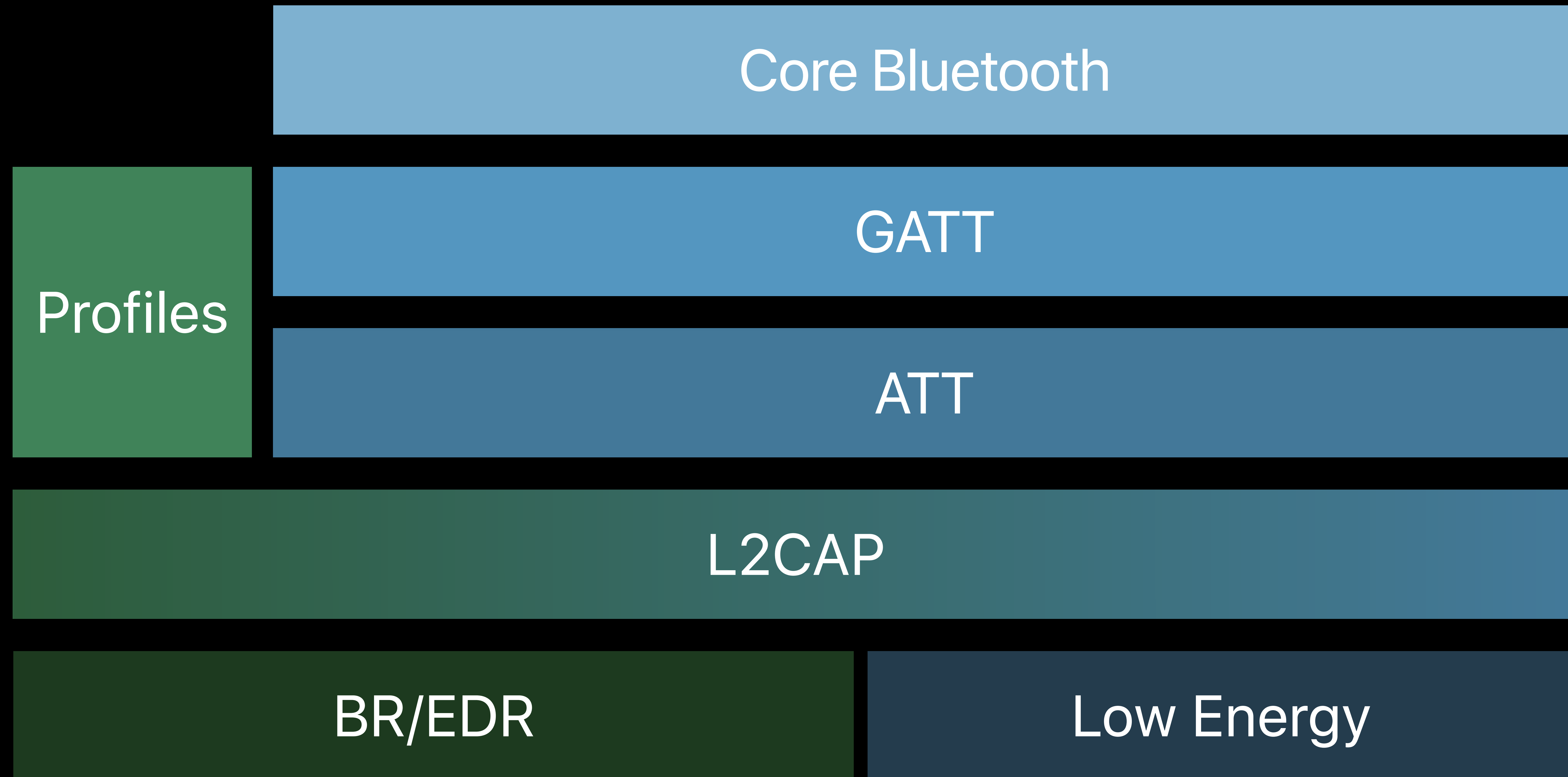


Core Bluetooth 2018



Core Bluetooth 2019

NEW



Core Bluetooth BR/EDR

NEW

Core Bluetooth BR/EDR

NEW

Use Core Bluetooth with BR/EDR Bluetooth devices

Core Bluetooth BR/EDR

NEW

Use Core Bluetooth with BR/EDR Bluetooth devices

Bluetooth SIG GATT protocol running over BR/EDR

Core Bluetooth BR/EDR



NEW

Use Core Bluetooth with BR/EDR Bluetooth devices

Bluetooth SIG GATT protocol running over BR/EDR

Same CBPeripheral APIs

Core Bluetooth BR/EDR



NEW

Use Core Bluetooth with BR/EDR Bluetooth devices

Bluetooth SIG GATT protocol running over BR/EDR

Same CBPeripheral APIs

New API in CBCentralManager

Core Bluetooth BR/EDR



NEW

Use Core Bluetooth with BR/EDR Bluetooth devices

Bluetooth SIG GATT protocol running over BR/EDR

Same CBPeripheral APIs

New API in CBCentralManager

Available today with latest iOS, watchOS, and tvOS

Core Bluetooth BR/EDR

NEW

Use Core Bluetooth with BR/EDR Bluetooth devices

Bluetooth SIG GATT protocol running over BR/EDR

Same CBPeripheral APIs

New API in CBCentralManager

Available today with latest iOS, watchOS, and tvOS

Add support to your accessory

Registering for Connection Events

NEW

Registering for Connection Events



NEW

Connection registration by the Central

- Register by Service
- Register by Peripheral

Registering for Connection Events

NEW

Connection registration by the Central

- Register by Service
- Register by Peripheral

```
open class CBCentralManager : CBManager {  
    open func registerForConnectionEvents(options:[CBCConnectionEventMatchingOption:Any]?)  
}
```

Registering for Connection Events

NEW

Connection registration by the Central

- Register by Service
- Register by Peripheral

```
open class CBCentralManager : CManager {  
    open func registerForConnectionEvents(options:[CBCConnectionEventMatchingOption:Any]?)  
}
```

```
extension CBCConnectionEventMatchingOption {  
    public static let serviceUUIDs: CBCConnectionEventMatchingOption  
    public static let peripheralUUIDs: CBCConnectionEventMatchingOption  
}
```

Connection Event

NEW

Connection Event



NEW

Delegate callback

- Sent on matching connection
- Sent after registration if a matching connection already established

Connection Event

NEW

Delegate callback

- Sent on matching connection
- Sent after registration if a matching connection already established

```
optional func centralManager(_ central: CBCentralManager, connectionEventDidOccur event:
CBCConnectionEvent, for peripheral: CBPeripheral)
```

Incoming Connection

Initialization

```
private var central: CBCentralManager!  
central = CBCentralManager(delegate: self, queue: nil)
```

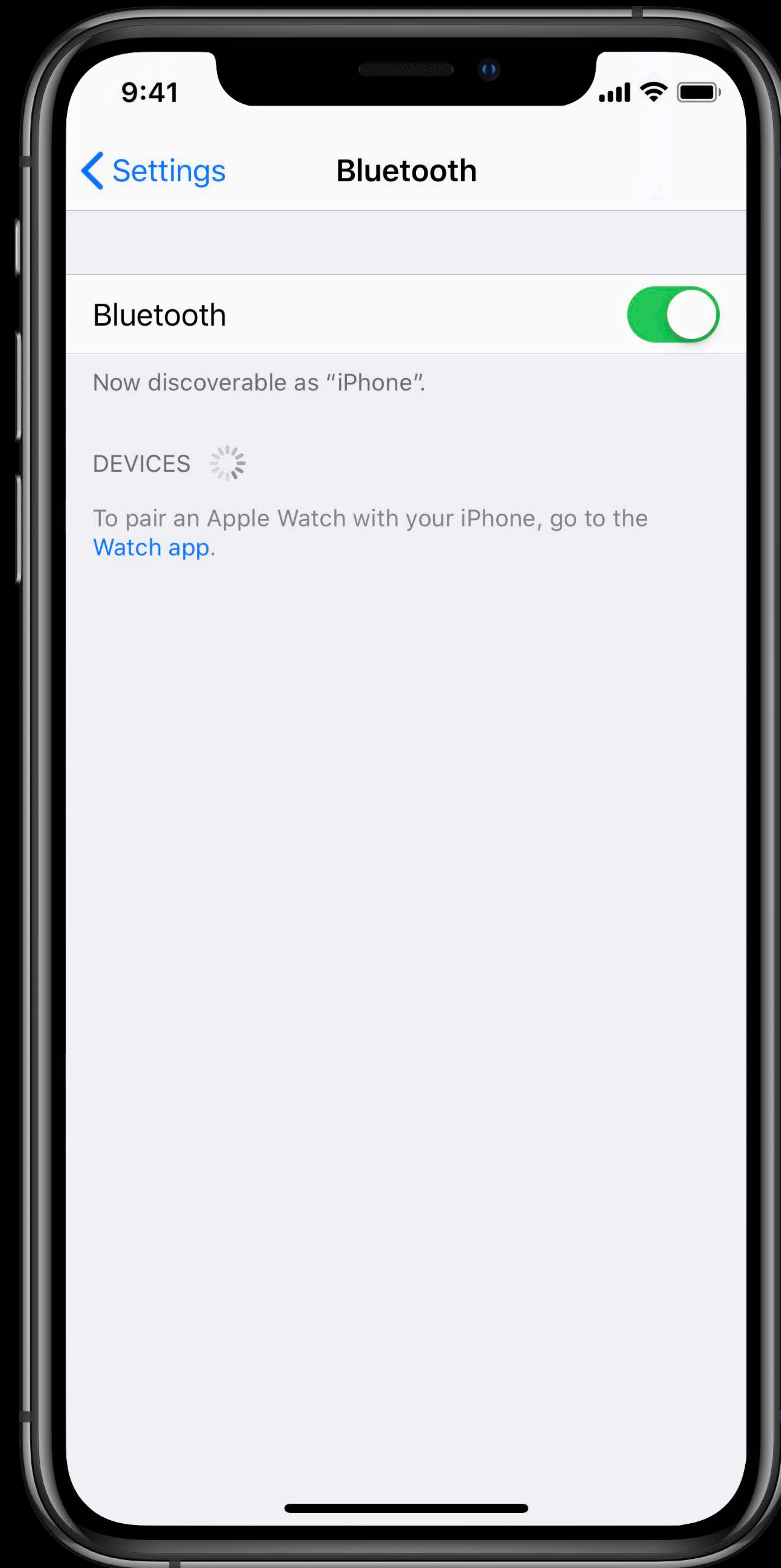


Registration

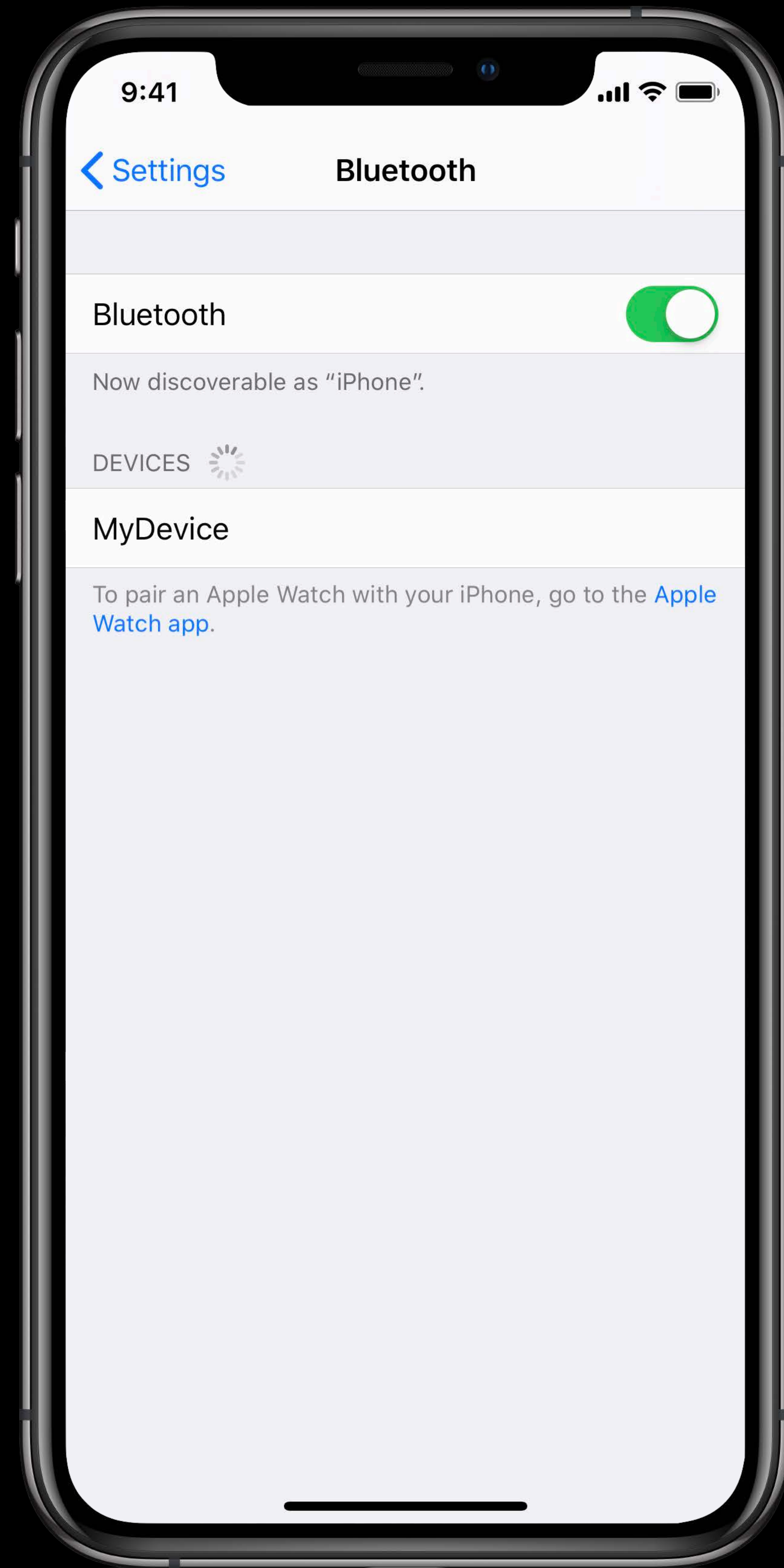
```
let matchingOptions =  
[CBCConnectionEventMatchingOption.serviceUUIDs :  
[myServiceUUID]]  
  
central.registerForConnectionEvents(options:  
matchingOptions)
```



Discover



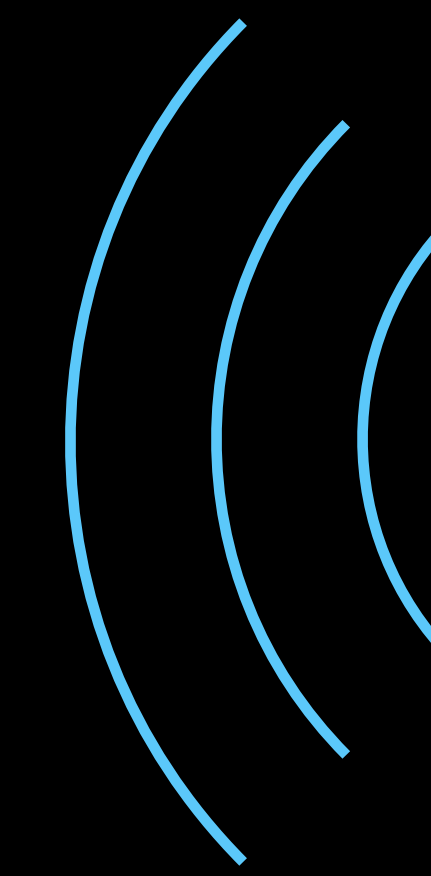
Discover



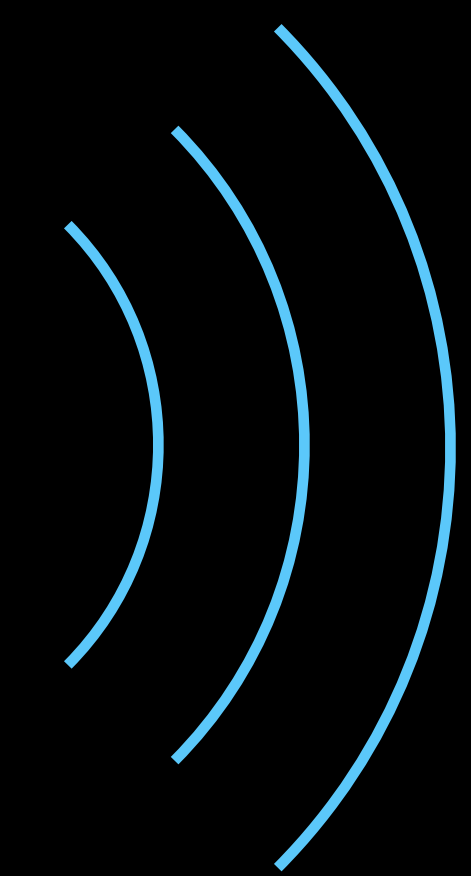
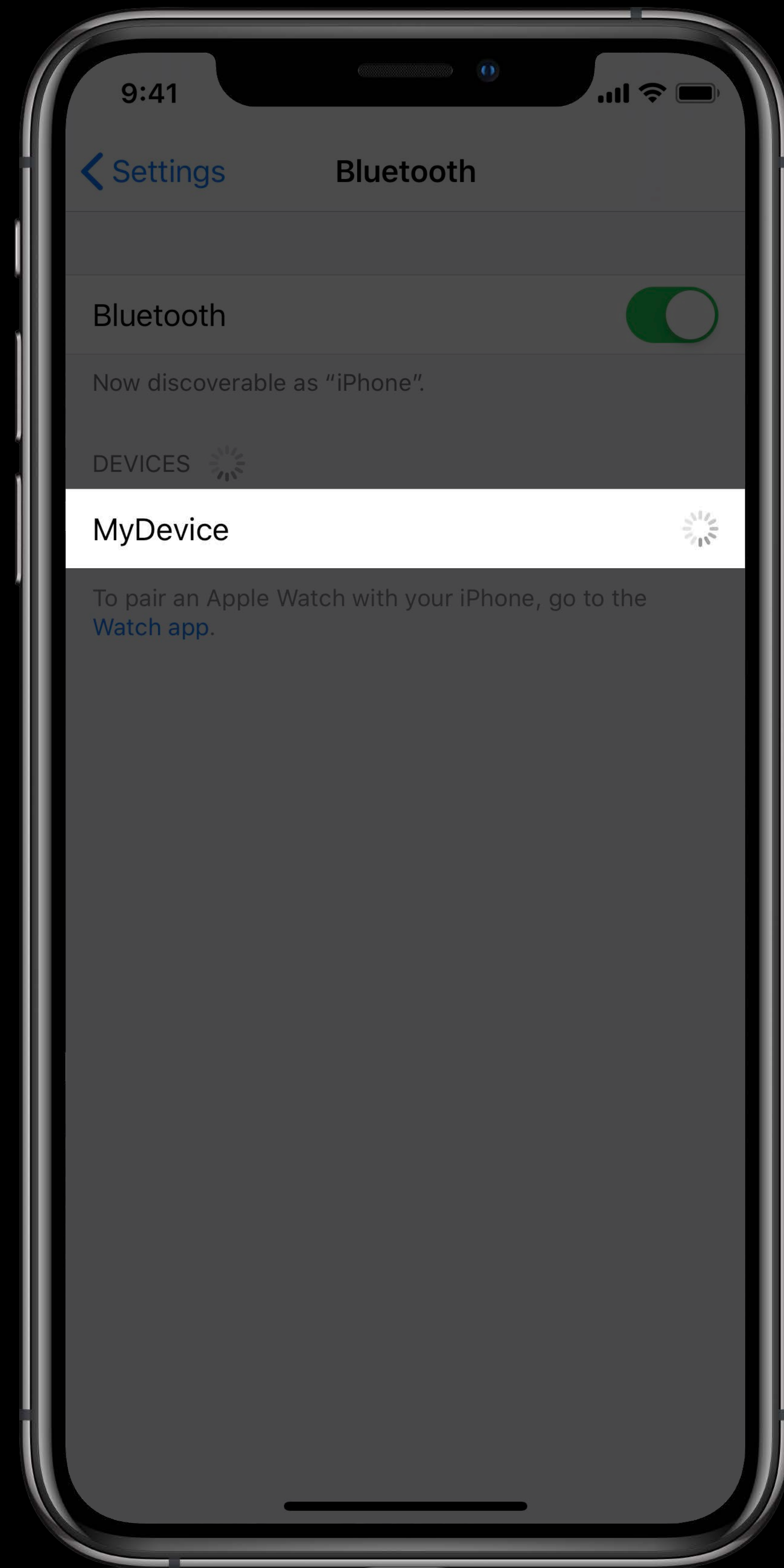
Inquiry Scan



Inquiry Response



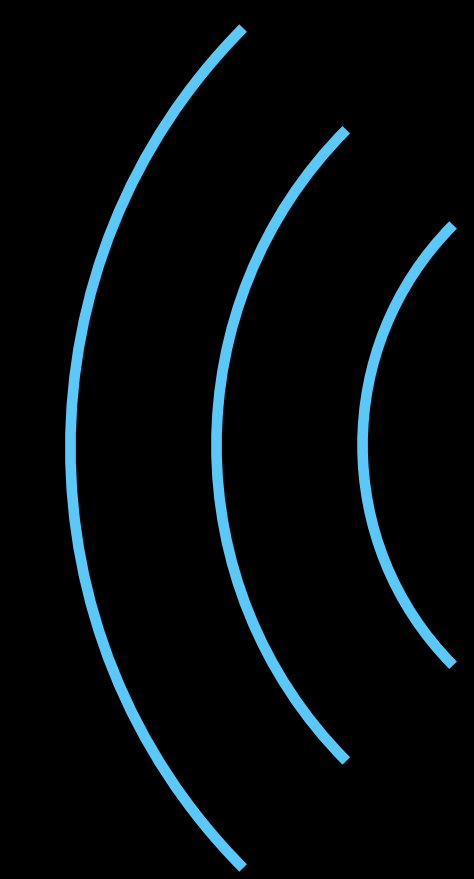
Discover



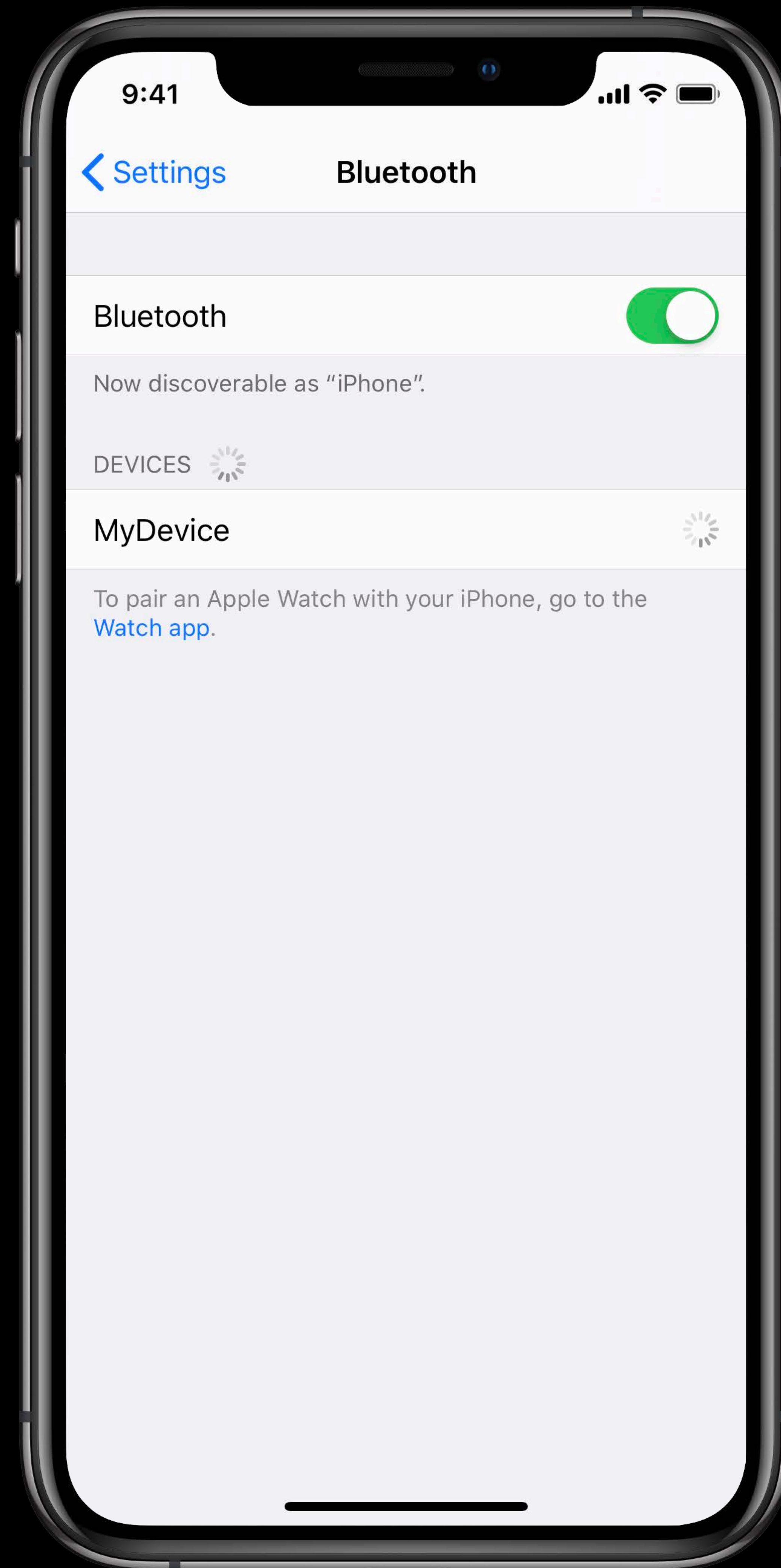
Inquiry Scan



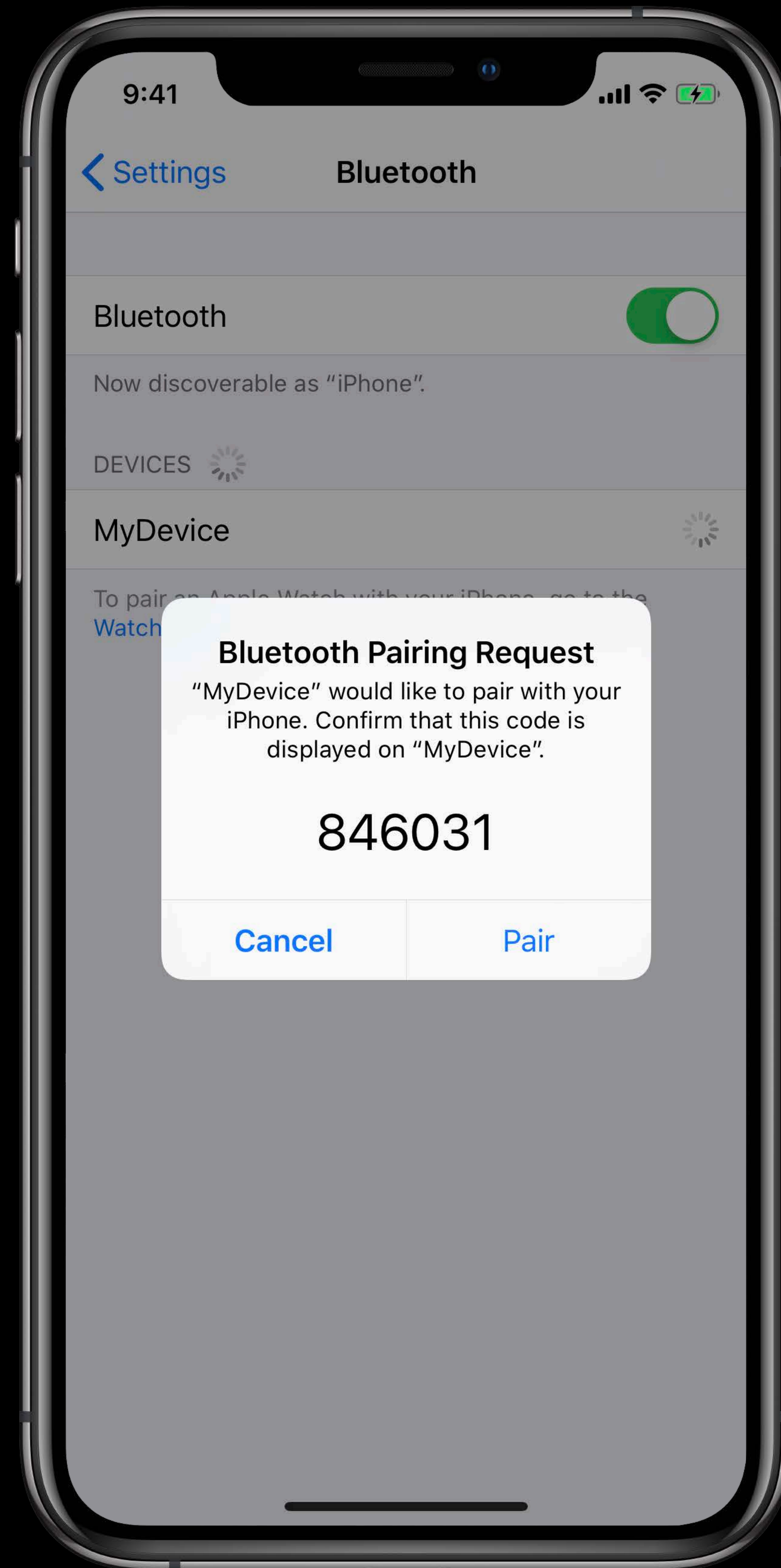
Inquiry Response



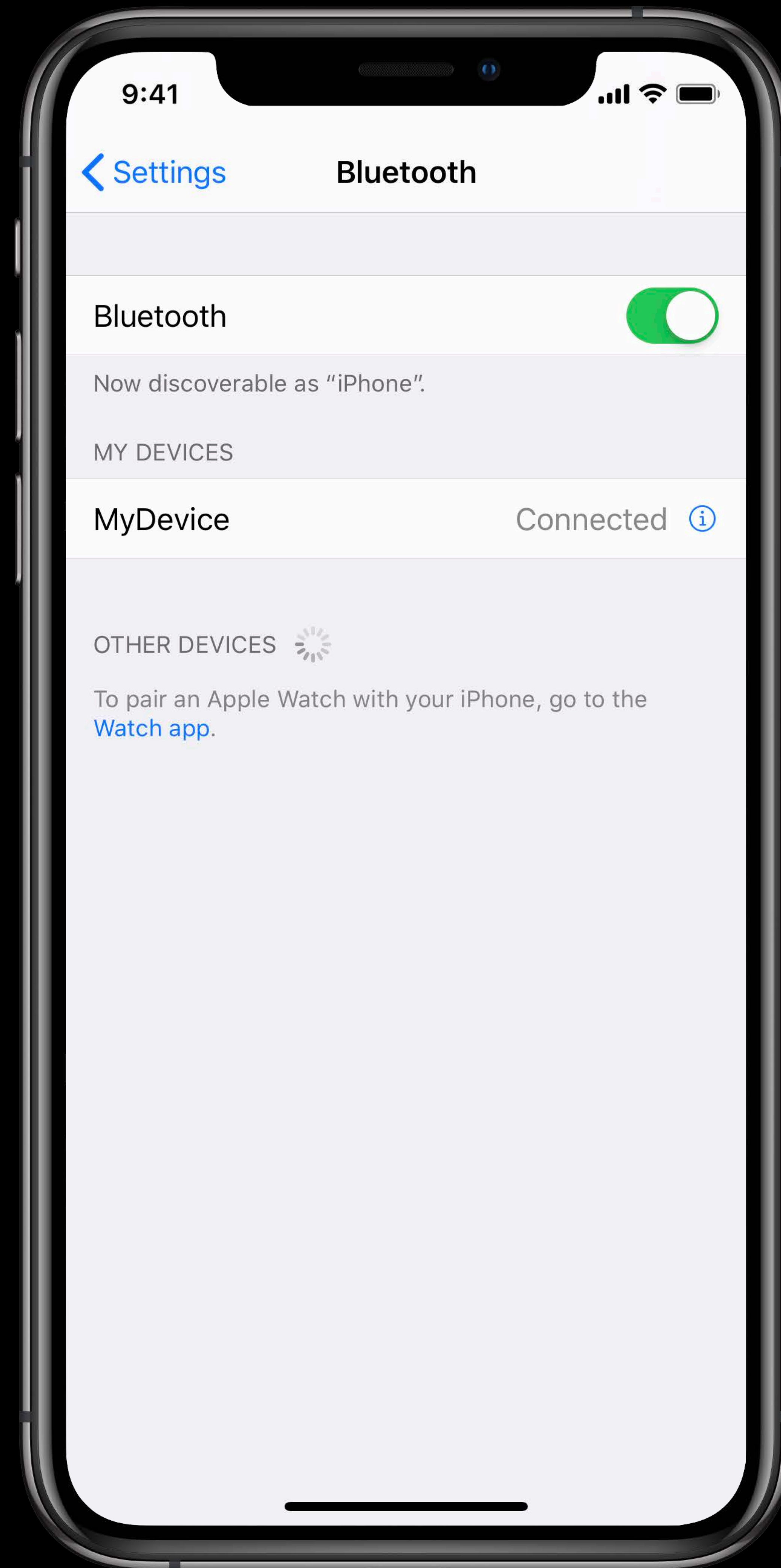
Connect/Pair



Connect/Pair



Connect/Pair



BR/EDR Connected and Paired



Delegate Callback



Delegate Callback

```
// Connection Event  
func centralManager(_ central: CBCentralManager,  
connectionEventDidOccur event: CBConnectionEvent,  
for peripheral: CBPeripheral) {  
  
}  
}
```



Delegate Callback

```
// Connection Event
func centralManager(_ central: CBCentralManager,
connectionEventDidOccur event: CBConnectionEvent,
for peripheral: CBPeripheral) {

    // Handle connection event

}
```



```
// Initialization
private var cbManager: CBCentralManager!
cbManager = CBCentralManager(delegate: self, queue: nil)

// Registering for Gatt Connection events
let matchingOptions = [CBCentralManagerMatchingOption.serviceUUIDs : [myServiceUUID]]()
cbManager.registerForConnectionEvents(options: matchingOptions)

// Delegate callback
func centralManager(_ central: CBCentralManager, connectionEventDidOccur event:
CBCentralManagerConnectionEvent, for peripheral: CBPeripheral) {
    switch event {
    case .peerConnected:
        // We are interested in this peripheral, clear registration
        cbManager.connect(peripheral, options: nil)
        cbManager.registerForConnectionEvents(options: nil)
    default:
        // Not interested
    }
}
```

```
// Initialization
private var cbManager: CBCentralManager!
cbManager = CBCentralManager(delegate: self, queue: nil)

// Registering for Gatt Connection events
let matchingOptions = [CBCentralManagerMatchingOption.serviceUUIDs : [myServiceUUID]]()
cbManager.registerForConnectionEvents(options: matchingOptions)

// Delegate callback
func centralManager(_ central: CBCentralManager, connectionEventDidOccur event:
CBCentralManagerConnectionEvent, for peripheral: CBPeripheral) {
    switch event {
    case .peerConnected:
        // We are interested in this peripheral, clear registration
        cbManager.connect(peripheral, options: nil)
        cbManager.registerForConnectionEvents(options: nil)
    default:
        // Not interested
    }
}
```

```
// Initialization
private var cbManager: CBCentralManager!
cbManager = CBCentralManager(delegate: self, queue: nil)

// Registering for Gatt Connection events
let matchingOptions = [CBCConnectionEventMatchingOption.serviceUUIDs : [myServiceUUID]]()
cbManager.registerForConnectionEvents(options: matchingOptions)

// Delegate callback
func centralManager(_ central: CBCentralManager, connectionEventDidOccur event:
CBCConnectionEvent, for peripheral: CBPeripheral) {
    switch event {
    case .peerConnected:
        // We are interested in this peripheral, clear registration
        cbManager.connect(peripheral, options: nil)
        cbManager.registerForConnectionEvents(options: nil)
    default:
        // Not interested
    }
}
```

```
// Initialization
private var cbManager: CBCentralManager!
cbManager = CBCentralManager(delegate: self, queue: nil)

// Registering for Gatt Connection events
let matchingOptions = [CBCentralManagerMatchingOption.serviceUUIDs : [myServiceUUID]]()
cbManager.registerForConnectionEvents(options: matchingOptions)
```

```
// Delegate callback
func centralManager(_ central: CBCentralManager, connectionEventDidOccur event:
CBCentralManagerConnectionEvent, for peripheral: CBPeripheral) {
    switch event {
    case .peerConnected:
        // We are interested in this peripheral, clear registration
        cbManager.connect(peripheral, options: nil)
        cbManager.registerForConnectionEvents(options: nil)
    default:
        // Not interested
    }
}
```

```
// Initialization
private var cbManager: CBCentralManager!
cbManager = CBCentralManager(delegate: self, queue: nil)

// Registering for Gatt Connection events
let matchingOptions = [CBCConnectionEventMatchingOption.serviceUUIDs : [myServiceUUID]]()
cbManager.registerForConnectionEvents(options: matchingOptions)

// Delegate callback
func centralManager(_ central: CBCentralManager, connectionEventDidOccur event:
CBCConnectionEvent, for peripheral: CBPeripheral) {
    switch event {
    case .peerConnected:
        // We are interested in this peripheral, clear registration
        cbManager.connect(peripheral, options: nil)
        cbManager.registerForConnectionEvents(options: nil)
    default:
        // Not interested
    }
}
```

Outgoing Connection

Connecting Out



Connecting Out

```
private var central: CBCentralManager?  
central = CBCentralManager(delegate: self, queue: nil)
```



Connecting Out

```
private var central: CBCentralManager?  
central = CBCentralManager(delegate: self, queue: nil)  
  
central?.connect(myPeripheral, options: nil)
```



BR/EDR Page



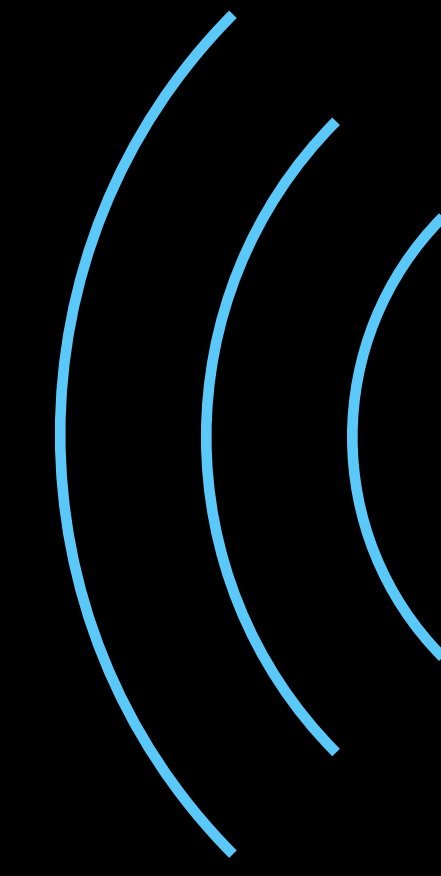
BR/EDR Page



Page Scan



Page



Connected



Connected

```
optional func centralManager(_ central:
CBCentralManager, didConnect peripheral: CBPeripheral)
{
    // Handle connection
}
```

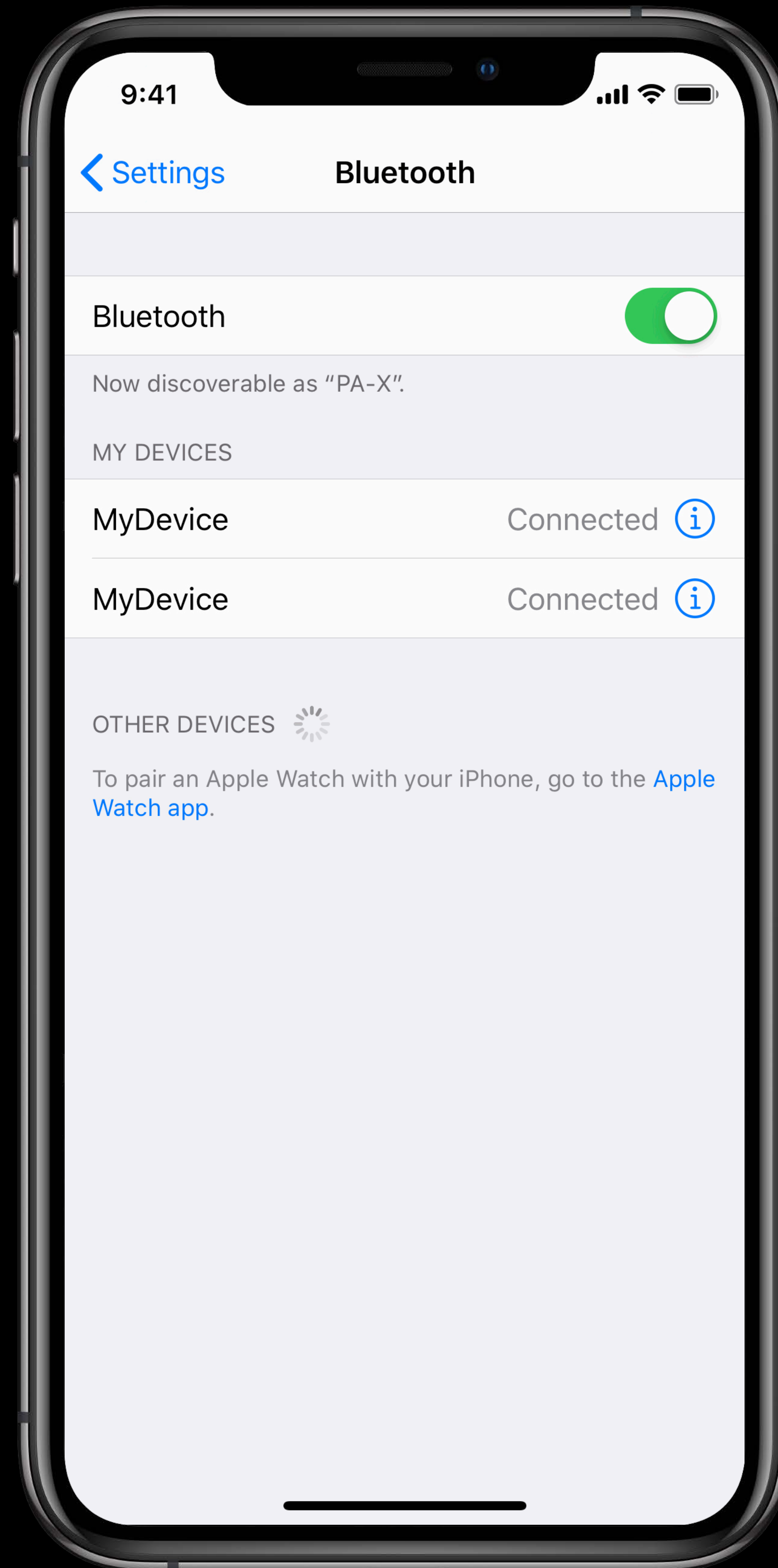


BR/EDR Connected and Paired

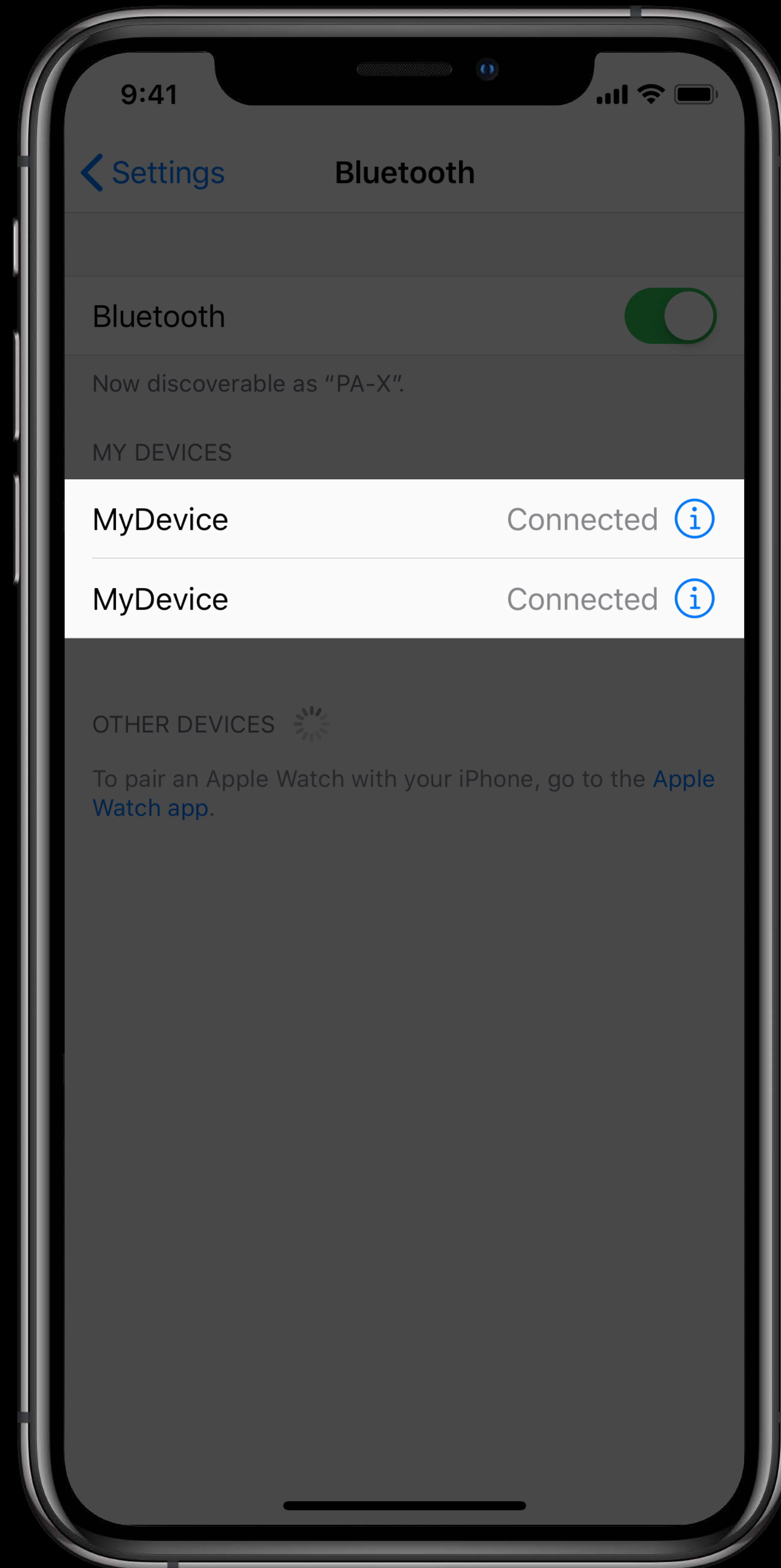


Core Bluetooth Dual-Mode

Improving Dual-Mode Pairing



Improving Dual-Mode Pairing



Cross Transport Key Derivation

NEW



Cross Transport Key Derivation

NEW

Bluetooth 4.2 SIG Specification



Cross Transport Key Derivation

NEW

Bluetooth 4.2 SIG Specification

Single pairing process



Cross Transport Key Derivation

NEW

Bluetooth 4.2 SIG Specification

Single pairing process

Same CBPeripheral identifier



Cross Transport Key Derivation

NEW

Bluetooth 4.2 SIG Specification

Single pairing process

Same CBPeripheral identifier

Transparent to the application



Cross Transport Key Derivation

NEW

Bluetooth 4.2 SIG Specification

Single pairing process

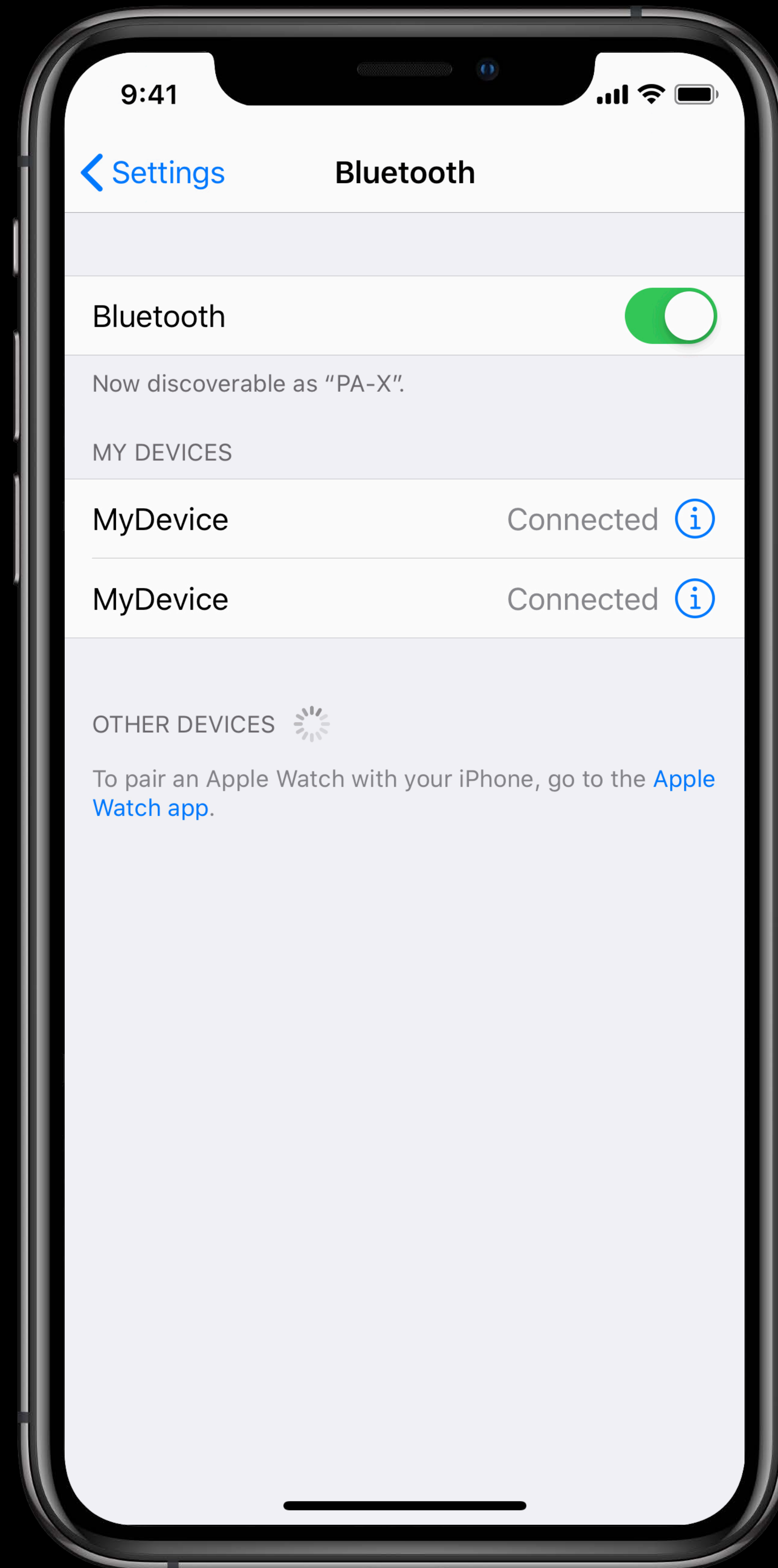
Same CBPeripheral identifier

Transparent to the application



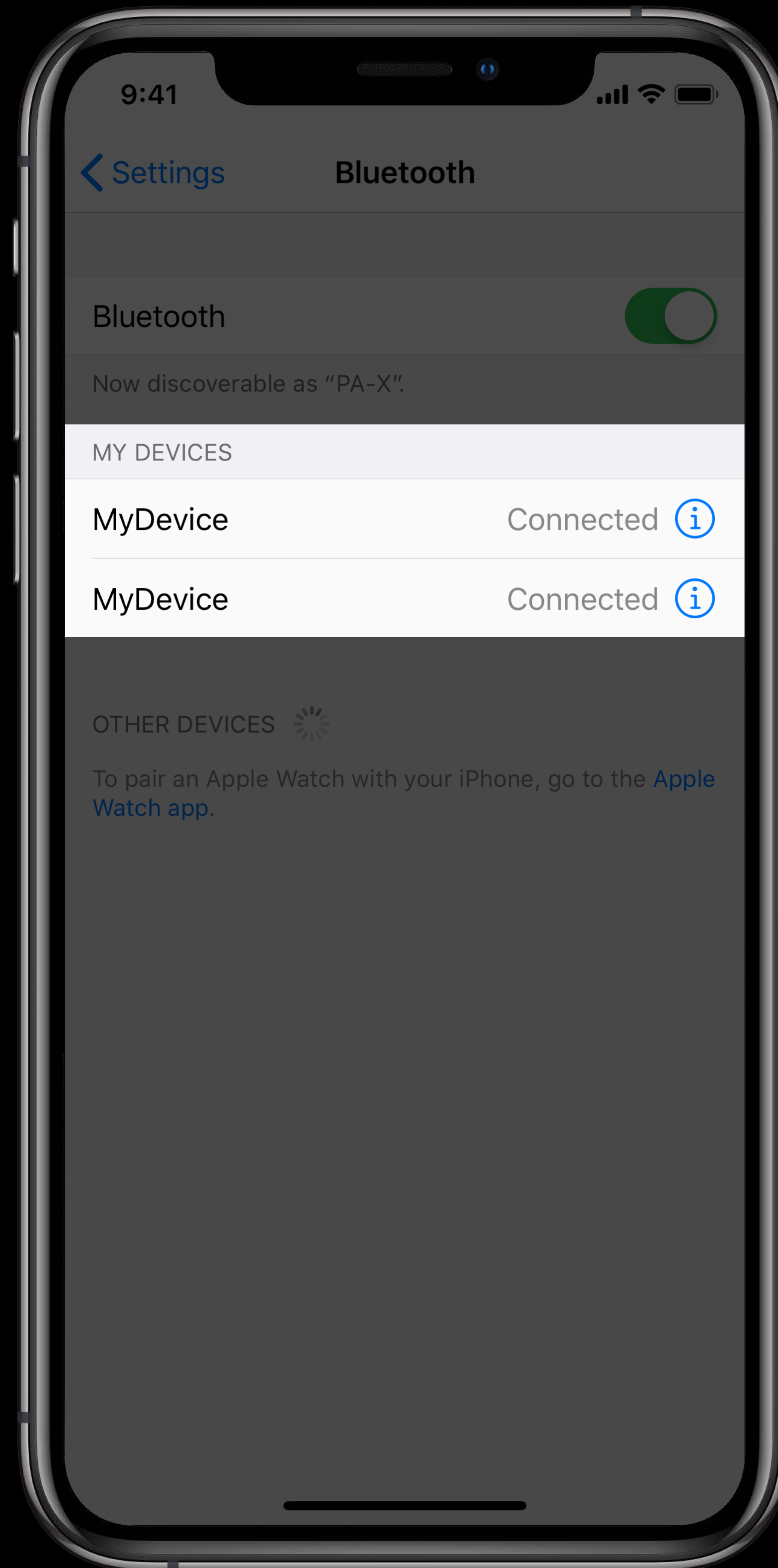
Cross Transport Key Derivation

NEW



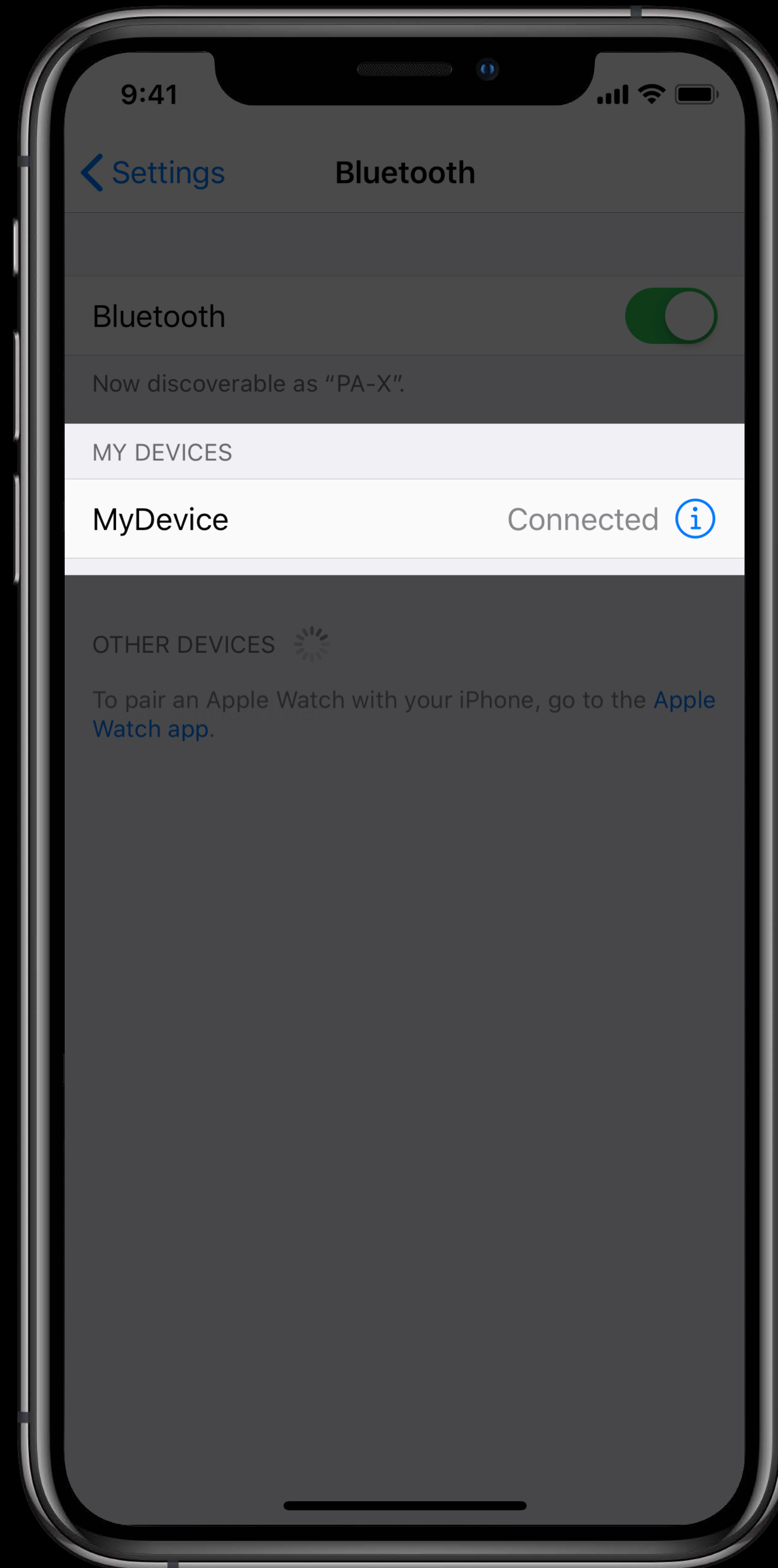
Cross Transport Key Derivation

NEW

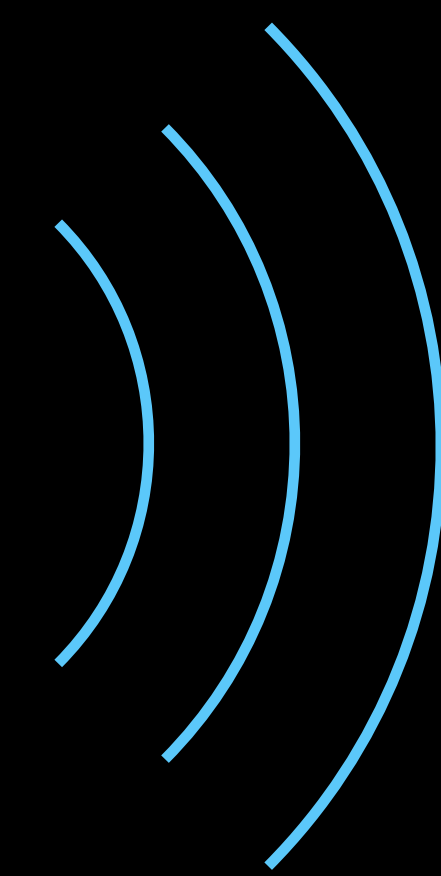
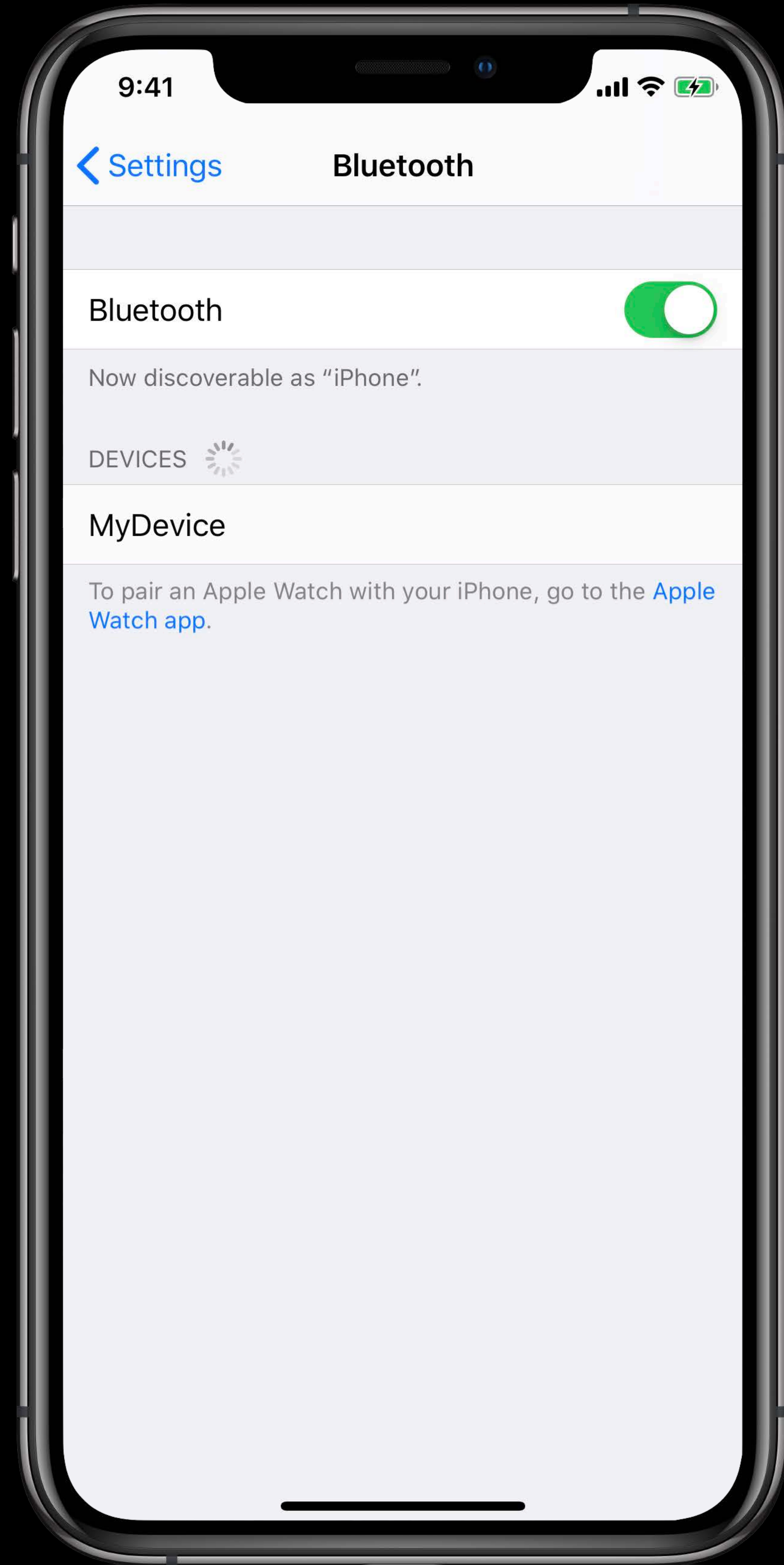


Cross Transport Key Derivation

NEW



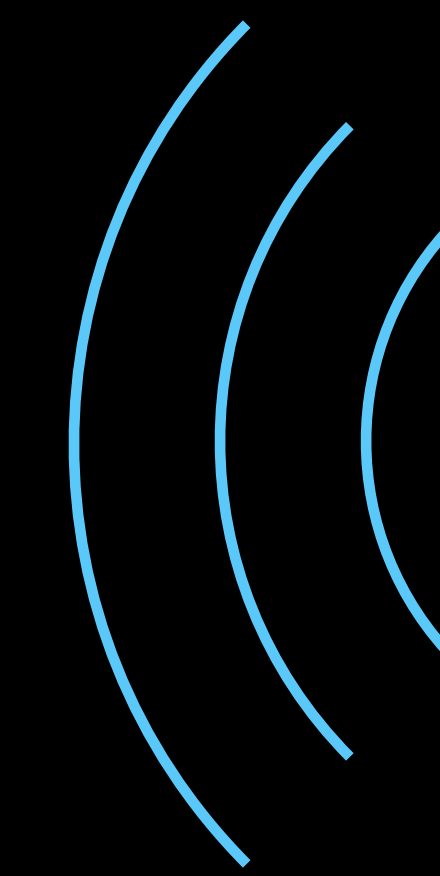
Instead of Inquiry



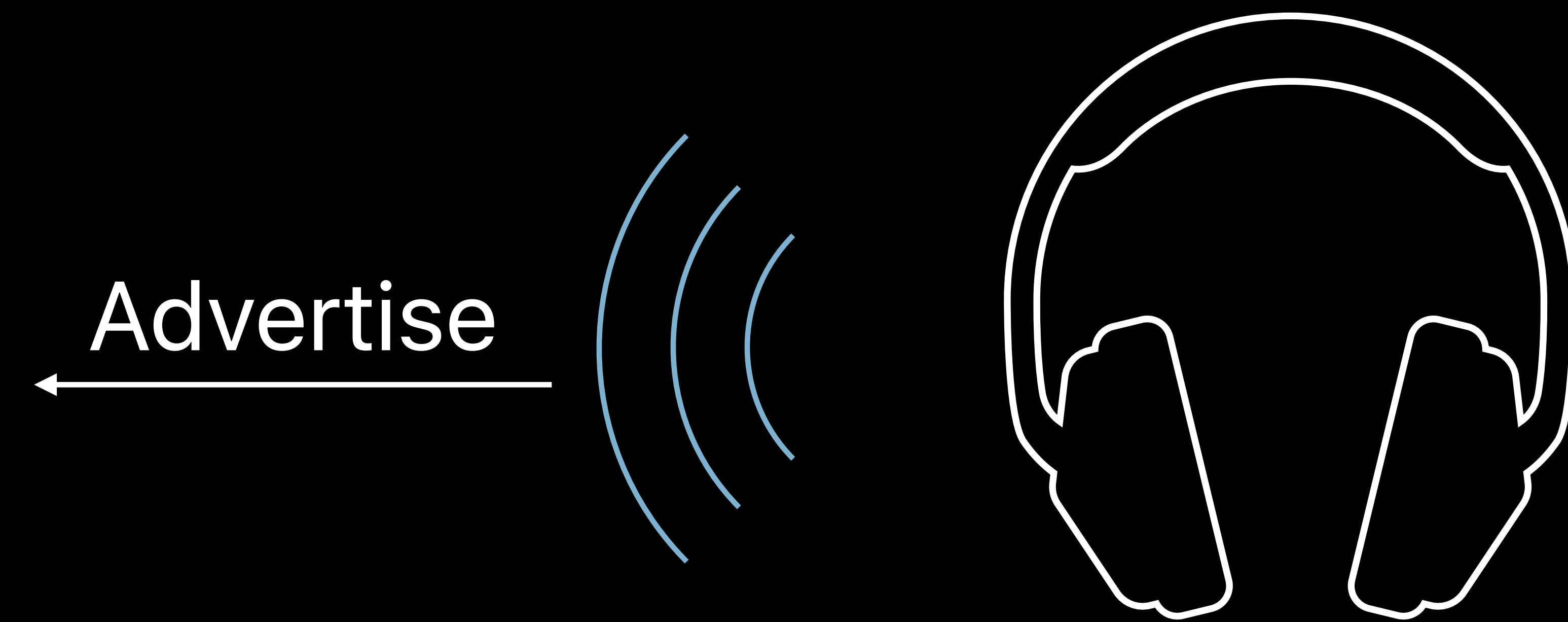
Inquiry Scan



Inquiry Response



Low Energy Scan



CTKD — Pairing



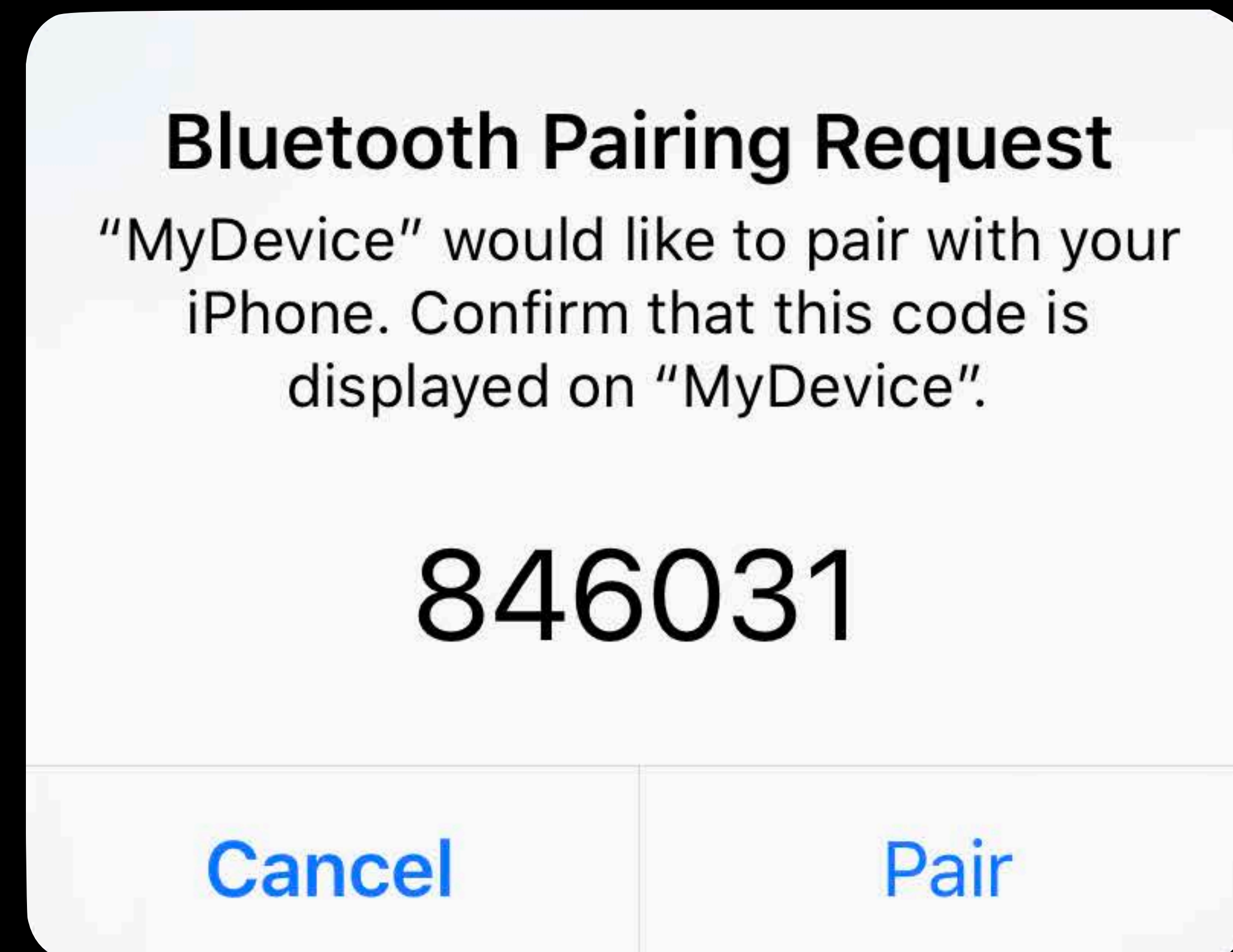
CTKD — Pairing



← LE Connected →



CTKD — Pairing



← LE Connected →



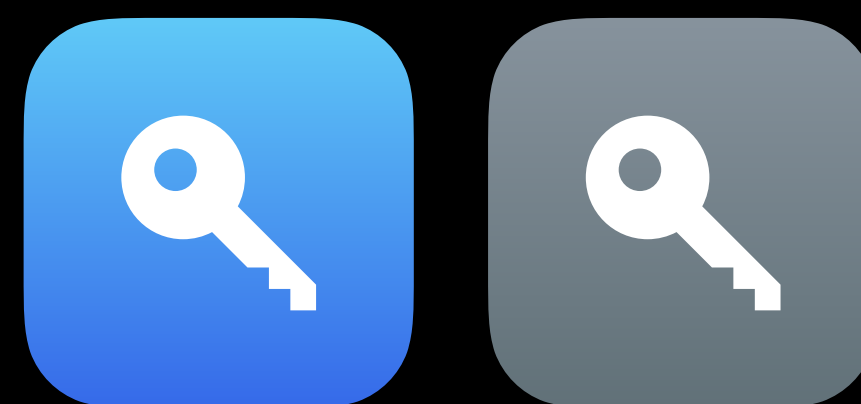
Key Derivation



← LE Connected and Paired →



Key Derivation



← LE Connected and Paired →



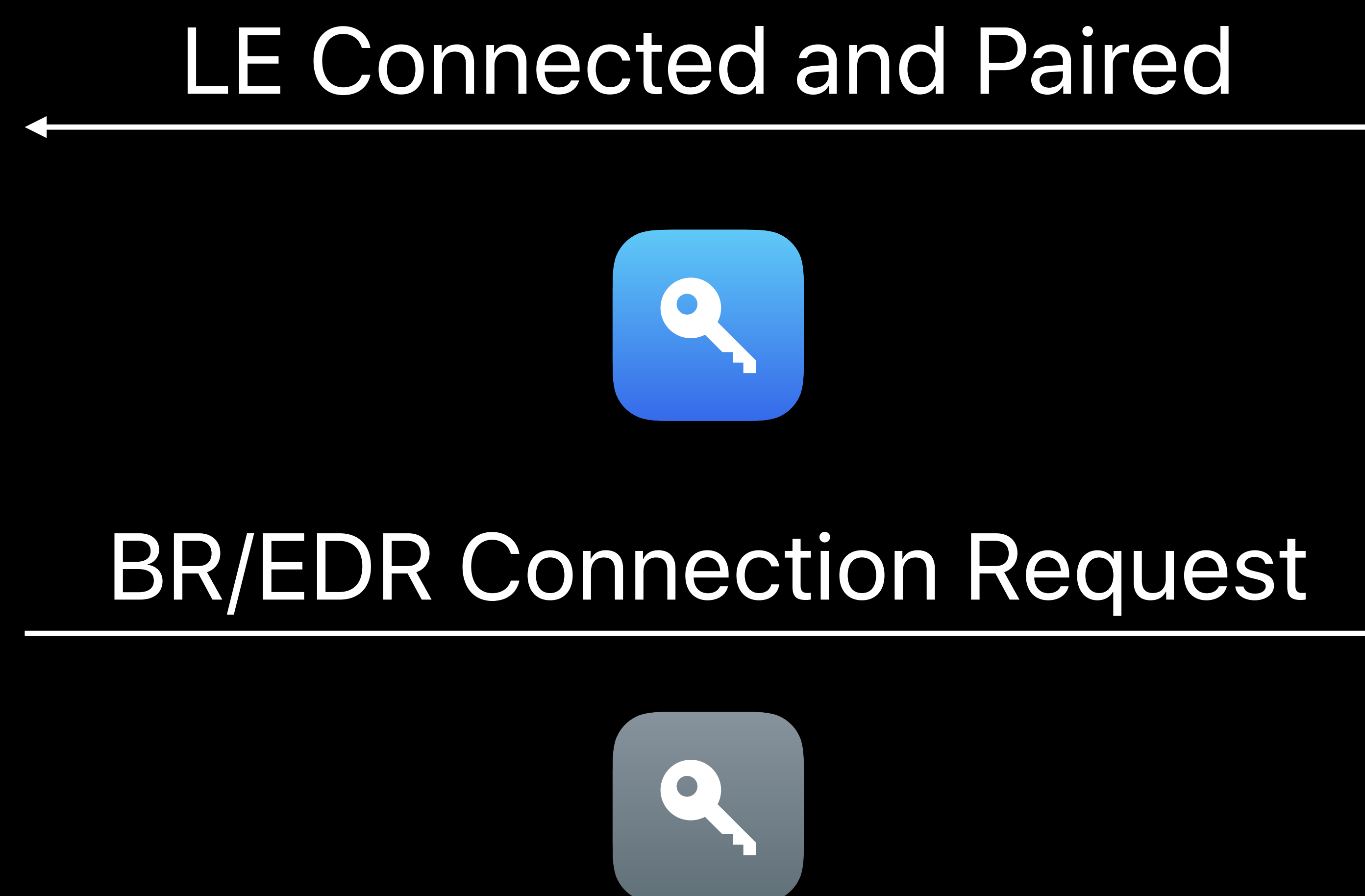
Key Derivation



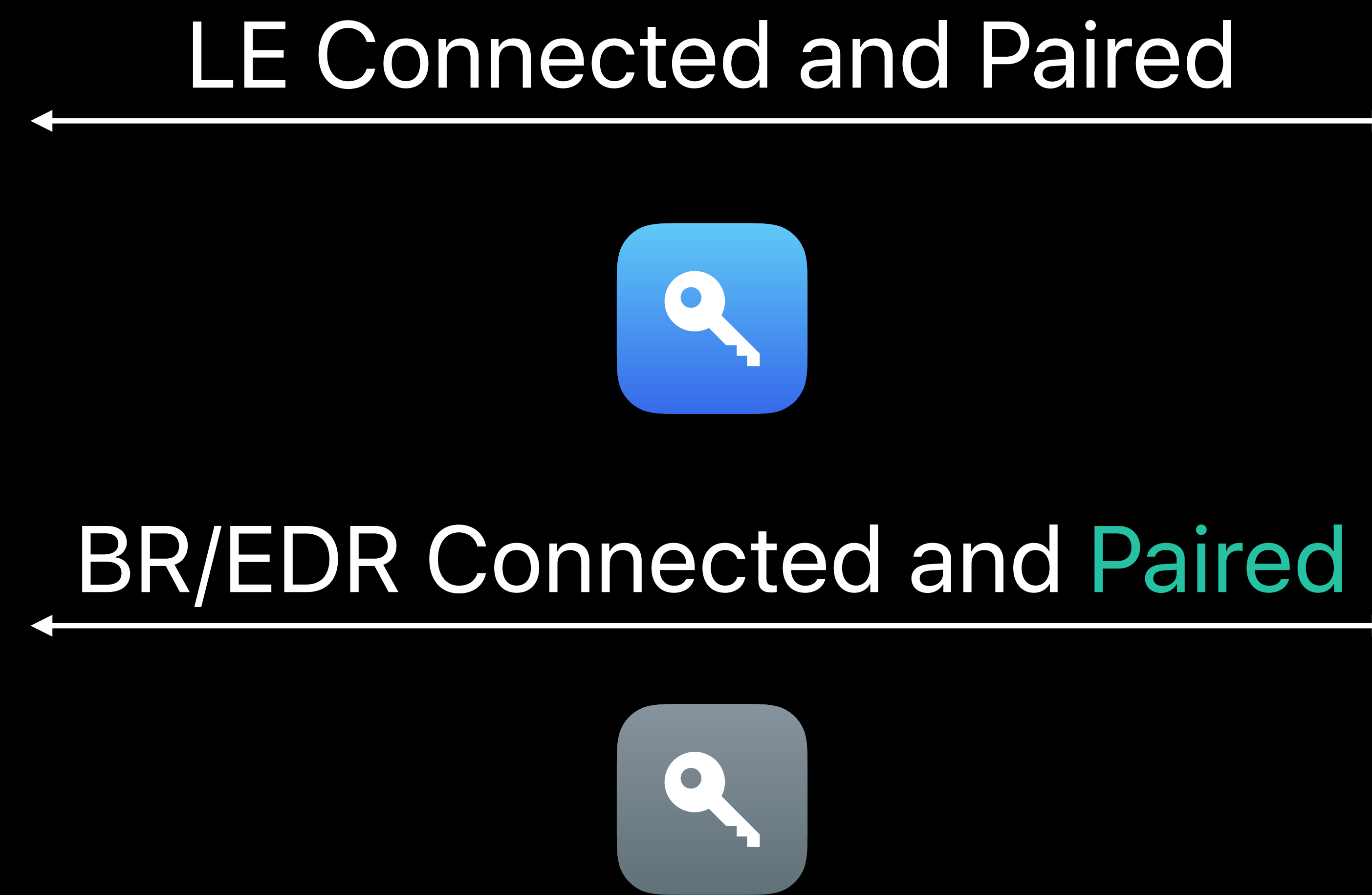
← LE Connected and Paired →



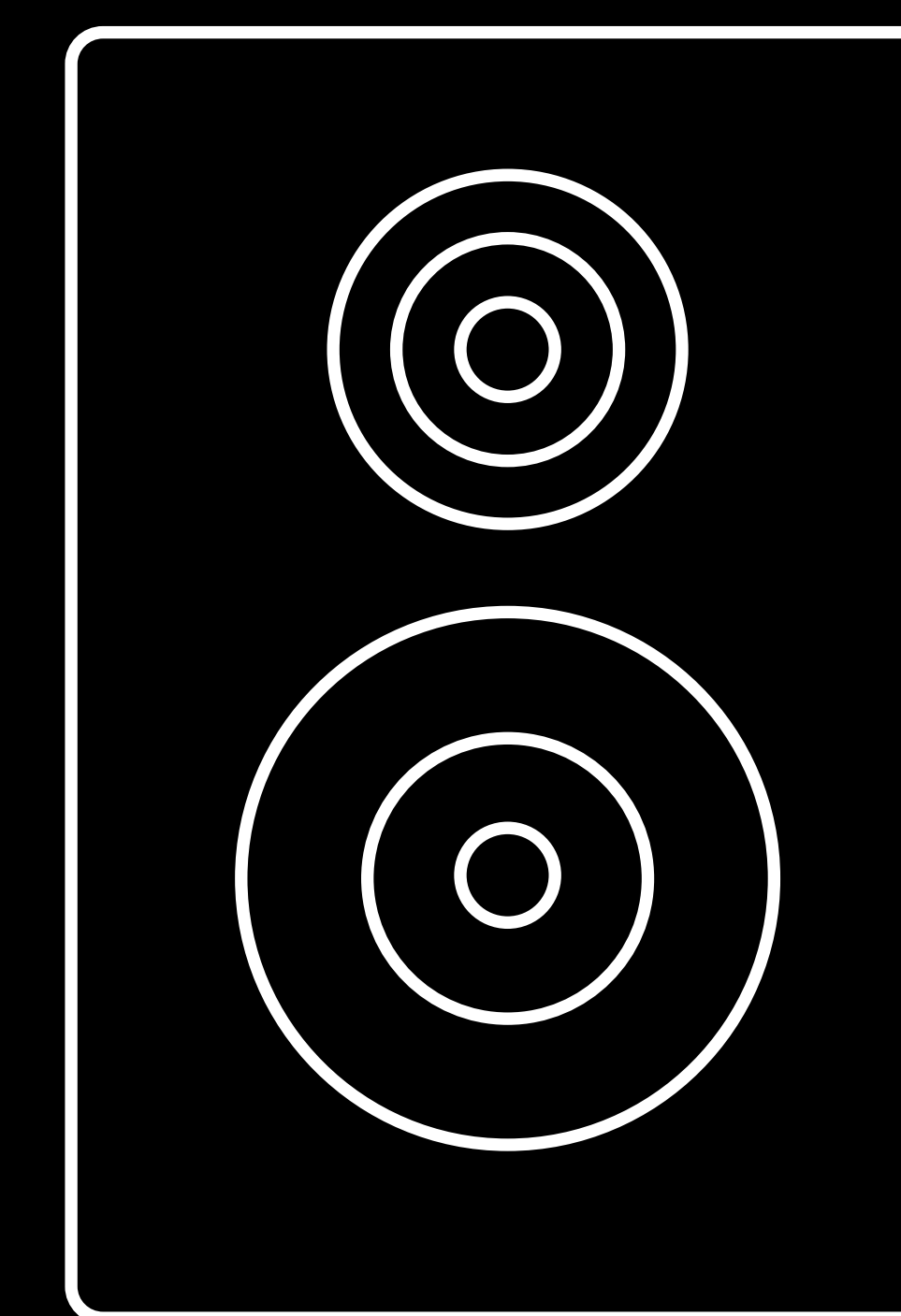
CTKD — BR/EDR Connected



CTKD — BR/EDR Connected



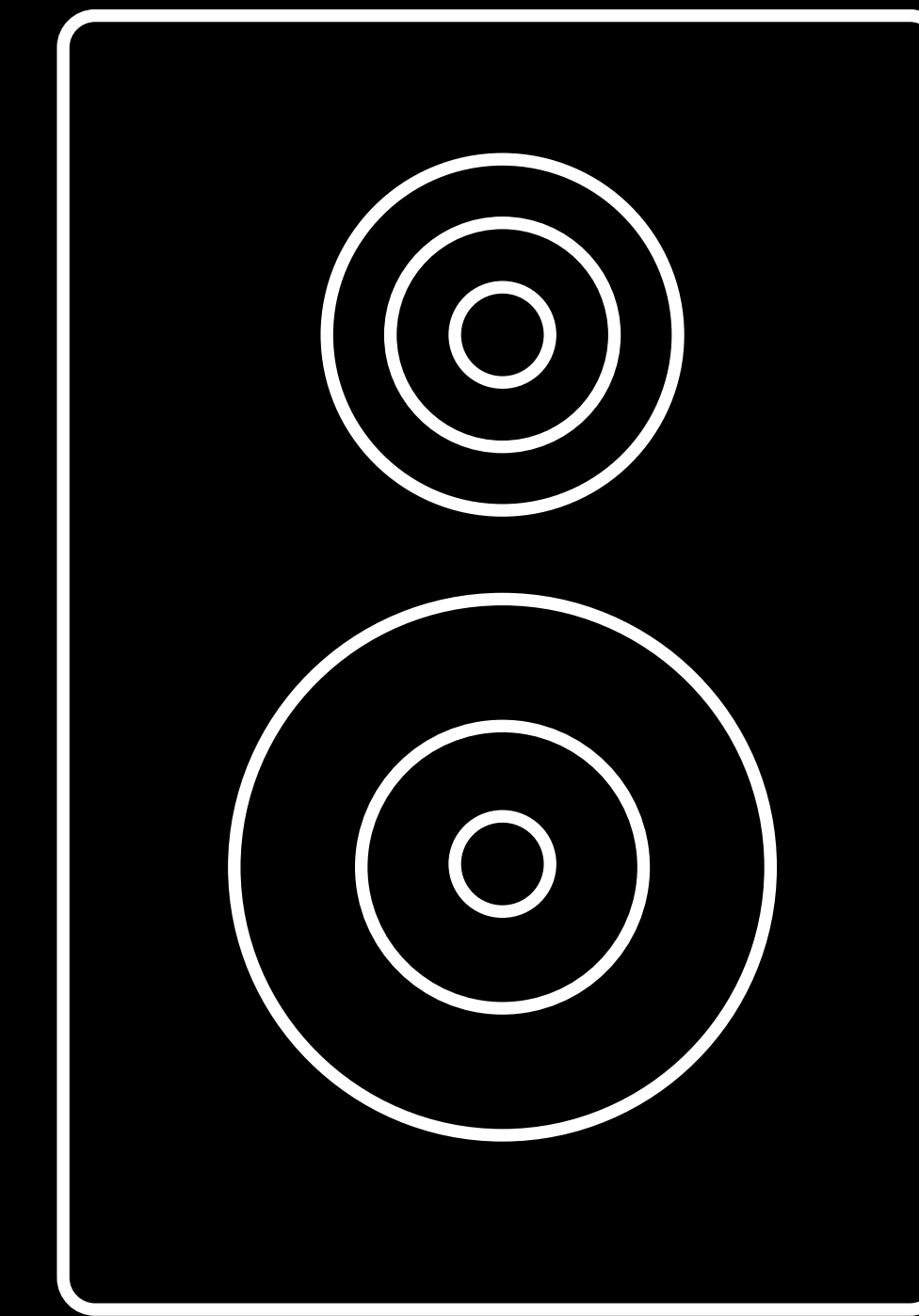
Improving Dual-Mode Connections



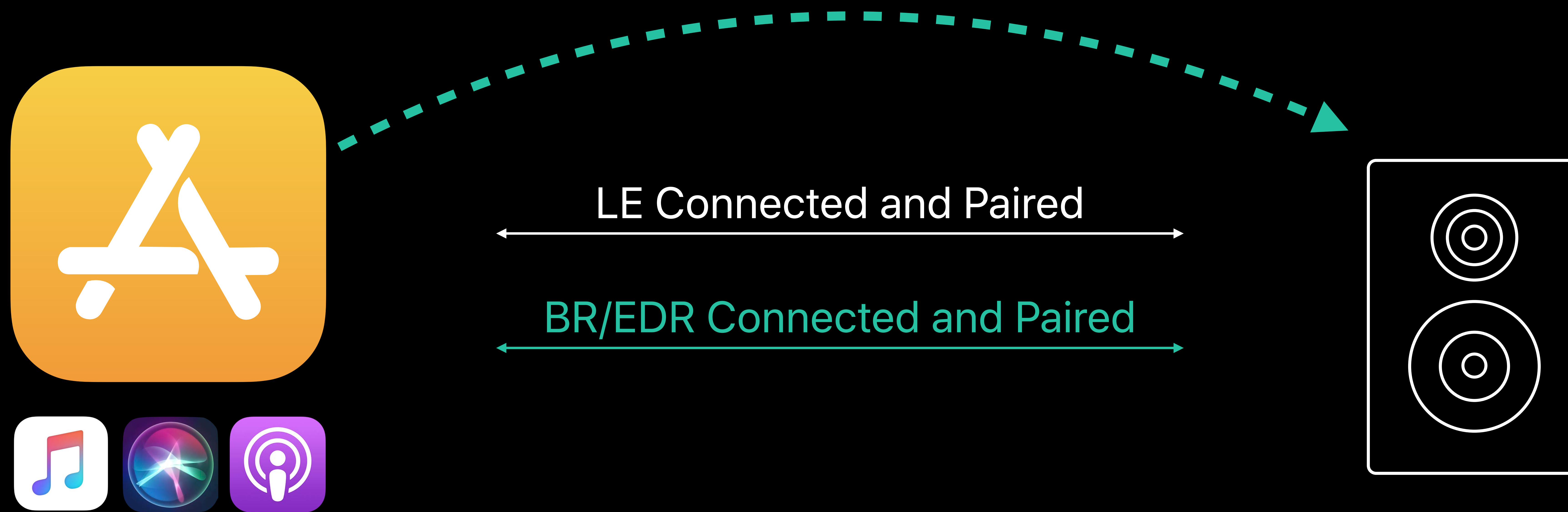
Improving Dual-Mode Connections



← LE Connected and Paired →



Improving Dual-Mode Connections



Bridging

NEW

Bridging

NEW

Low Energy proximity triggers BR/EDR connection

Bridging

NEW

Low Energy proximity triggers BR/EDR connection

Works on devices supporting CTKD

Bridging

NEW

Low Energy proximity triggers BR/EDR connection

Works on devices supporting CTKD

```
public let CBCentralManagerOptionEnableTransportBridgingKey: String
```

Bridging

NEW

Low Energy proximity triggers BR/EDR connection

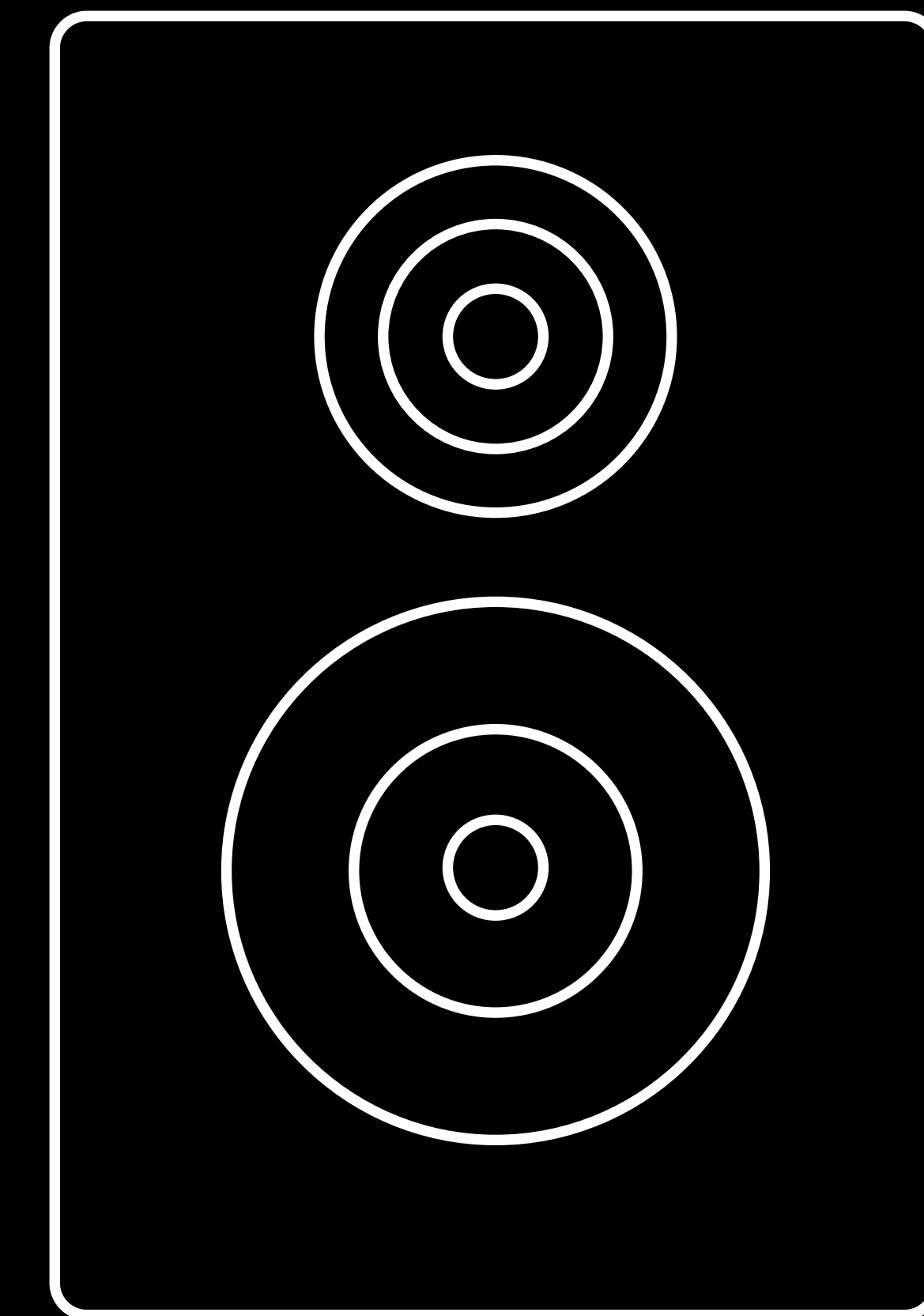
Works on devices supporting CTKD

```
public let CBCentralManagerOptionEnableTransportBridgingKey: String
```

```
cbCentralManager.connect(cbPeripheral,  
options:[CBCentralManagerOptionEnableTransportBridgingKey : true])
```

Bridging

NEW

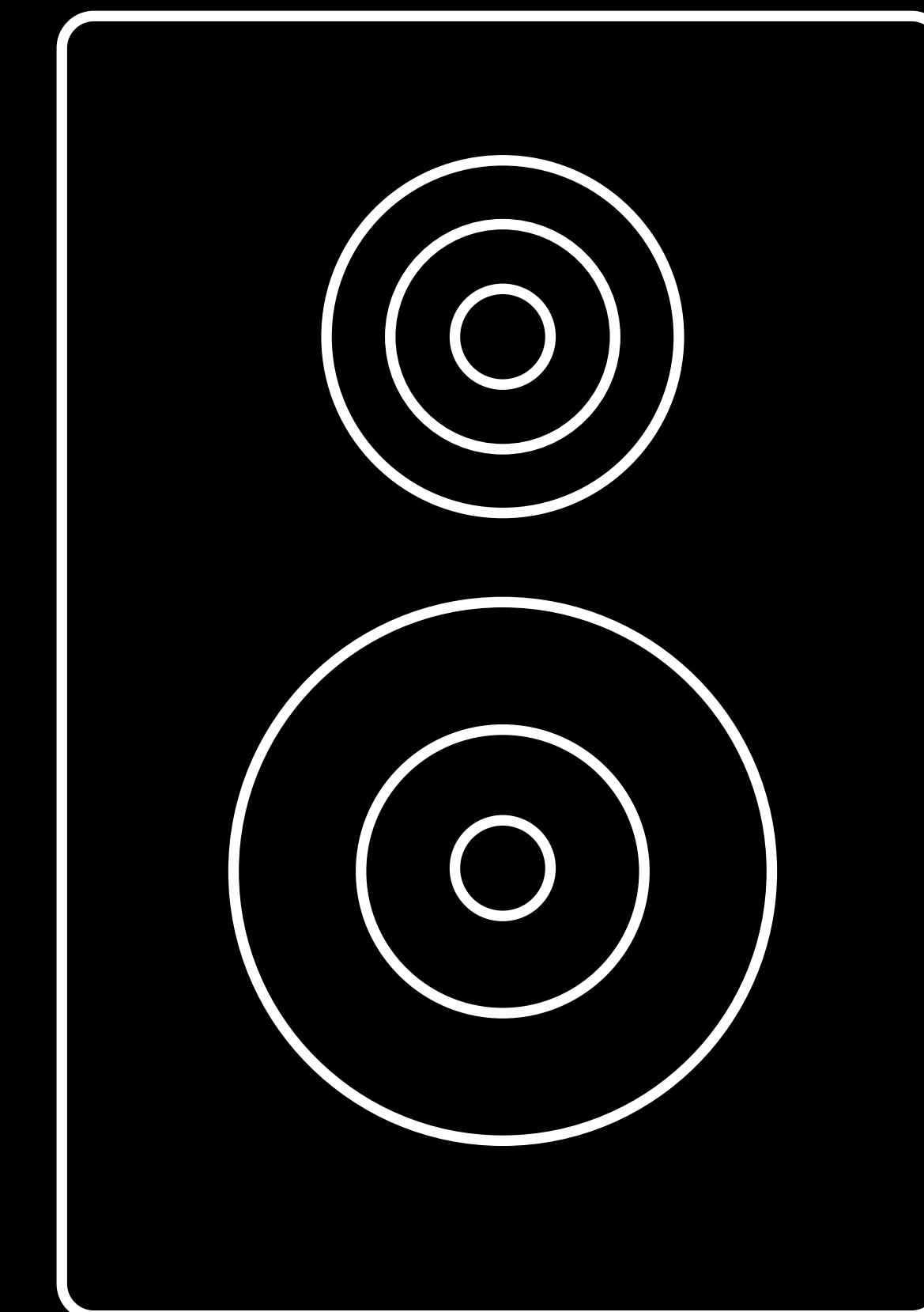


Bridging

NEW



← LE Connected and Paired →



```
cbCentralManager.connect(cbPeripheral, options:  
[CBCentralManagerOptionEnableTransportBridgingKey : true])
```

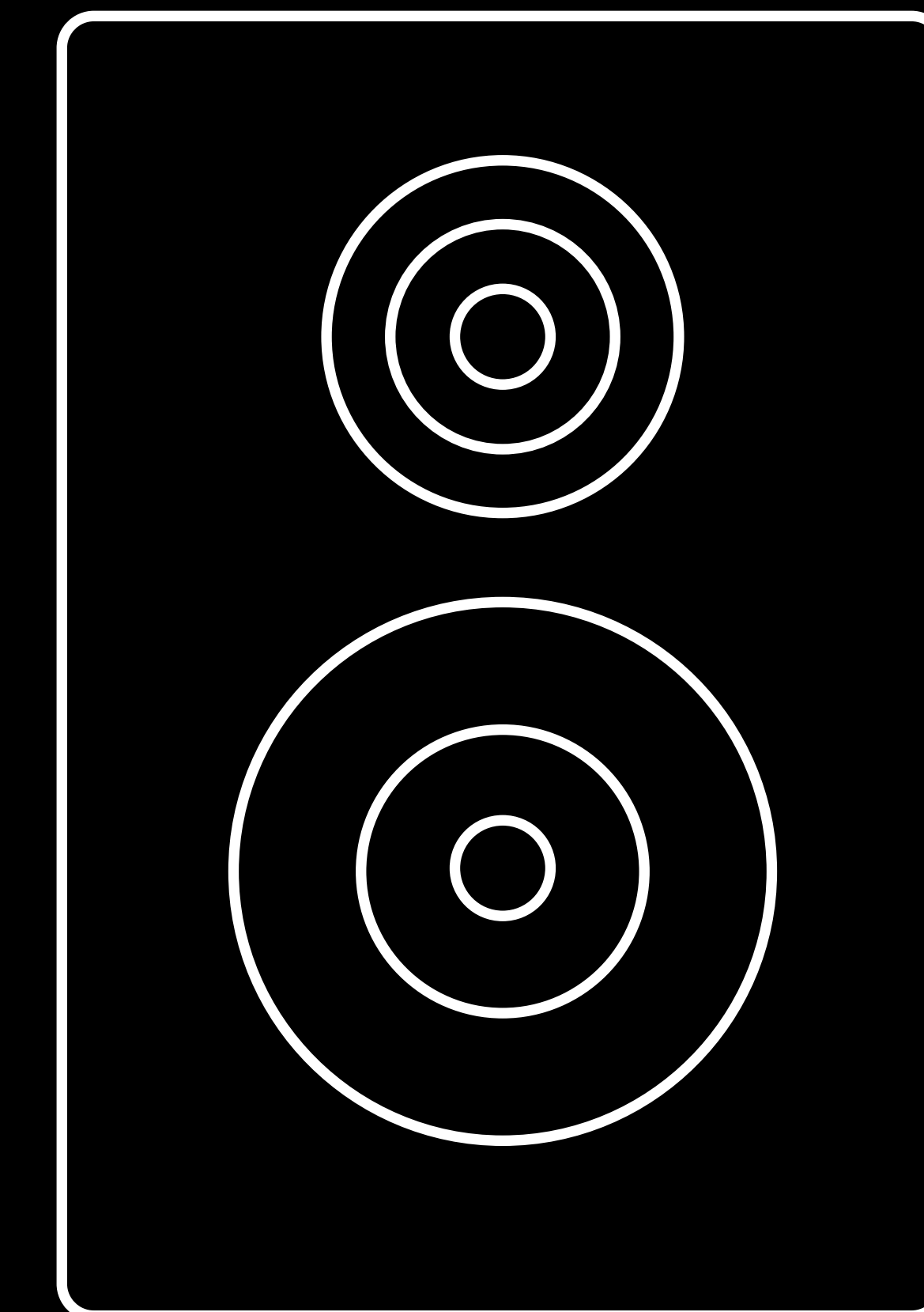
Bridging

NEW



BR/EDR Connection Request

LE Connected and Paired



```
cbCentralManager.connect(cbPeripheral, options:  
[CBCentralManagerOptionEnableTransportBridgingKey : true])
```

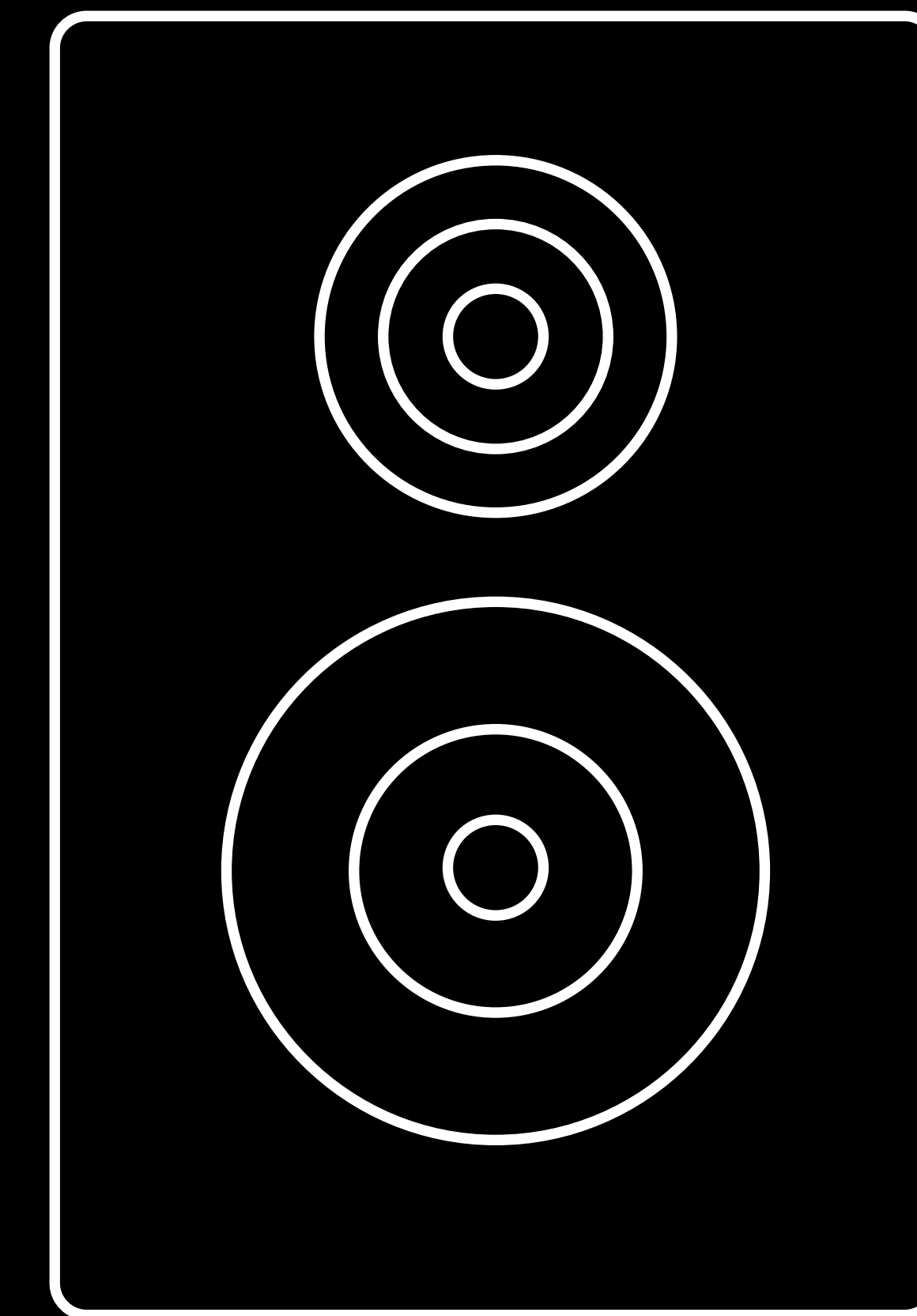
Bridging



BR/EDR Connected and Paired



LE Connected and Paired



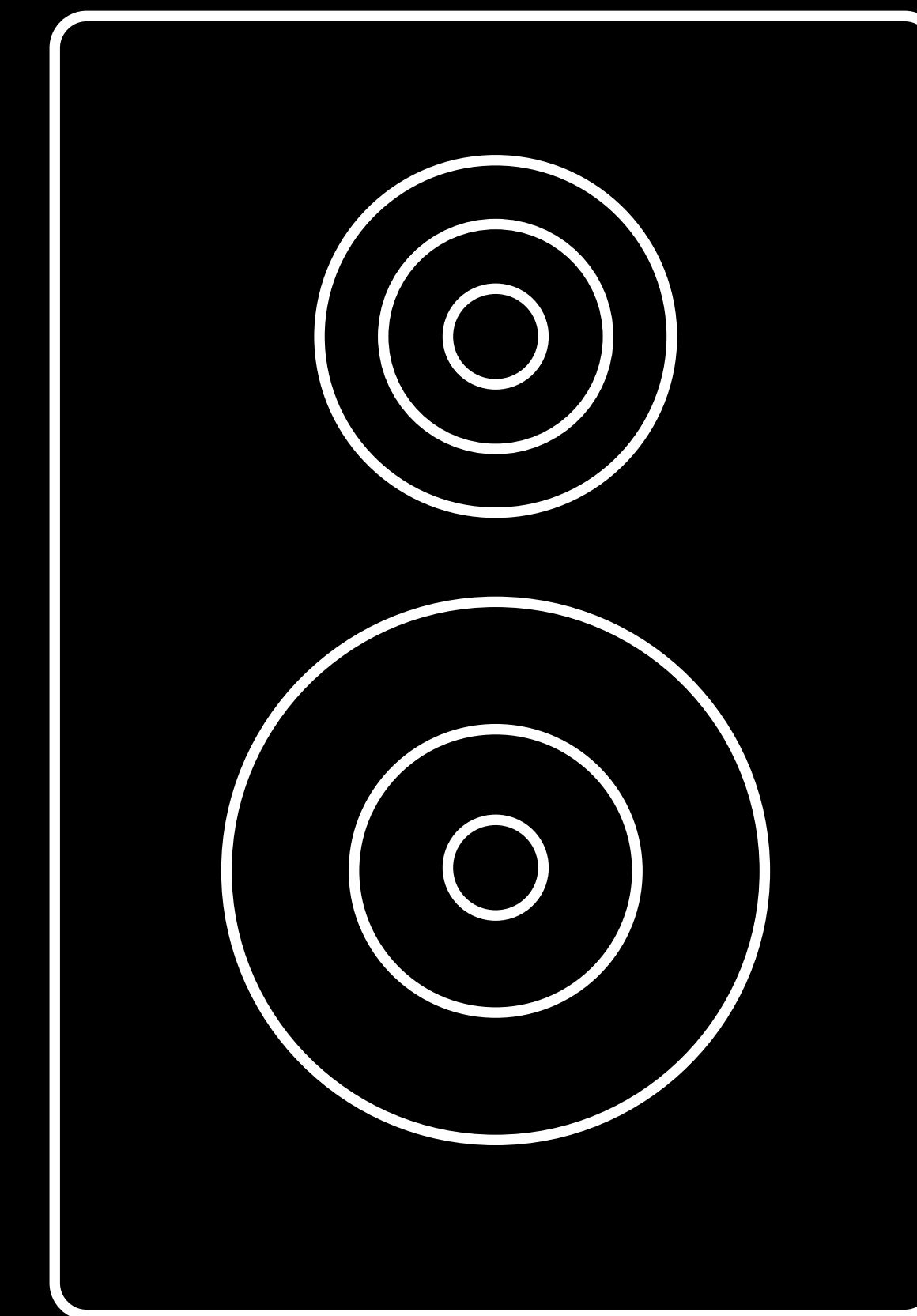
Bridging

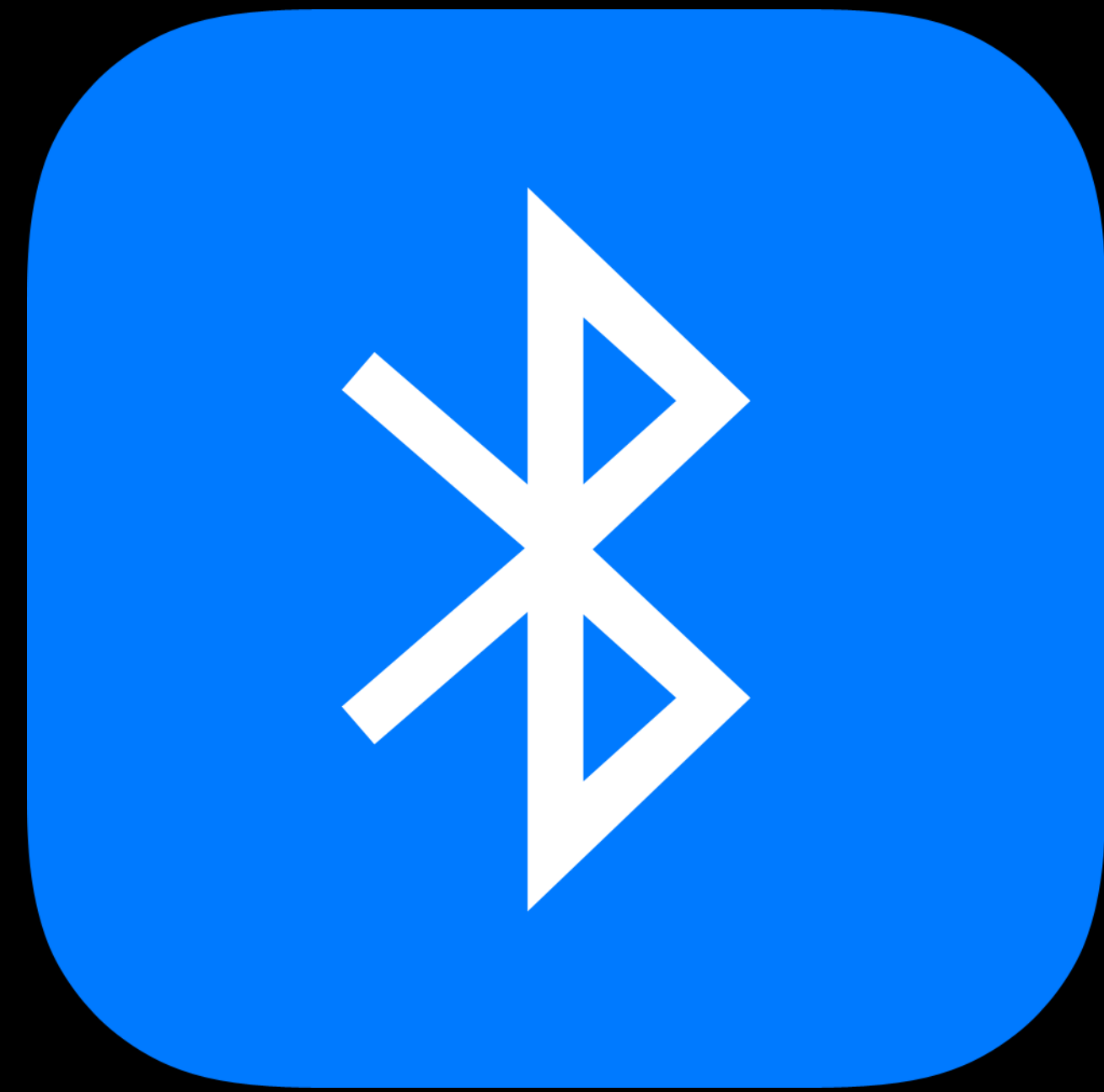


BR/EDR Connected and Paired



LE Connected and Paired







Privacy Update

Meghna Lav, Bluetooth Engineer

A hand holding a gold iPhone X on a dark wooden table. The phone is held vertically, showing its back with the Apple logo and dual-camera system. The background is a dark, textured wooden surface with various objects: a glass of dark liquid in the top left, a metal bowl in the top right, a pair of glasses in the bottom left, and a decorative metal object in the bottom right.

Privacy

matters

Enhancements

Enhancements

User authorization

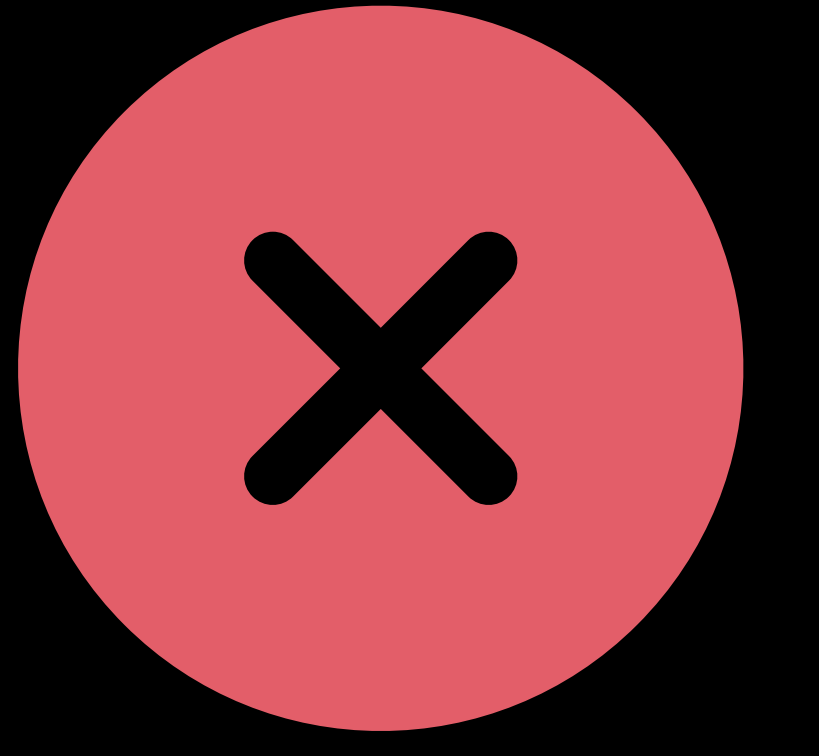
Enhancements

User authorization

Accessory notifications

User Authorization

How it currently works



Required only for background advertising

User Authorization

NEW

User Authorization

NEW

User consent required for all Core Bluetooth API's

**"CBCClassic" Would Like to
Use Bluetooth**

We use Bluetooth to discover, connect
to, and share information with nearby
devices

Don't Allow

OK

User Authorization

NEW

User consent required for all Core Bluetooth API's

Applies to apps linked with any SDK

**"CBCClassic" Would Like to
Use Bluetooth**

We use Bluetooth to discover, connect
to, and share information with nearby
devices

Don't Allow

OK

User Authorization

NEW

User consent required for all Core Bluetooth API's

Applies to apps linked with any SDK

Can be modified in Settings

**"CBCClassic" Would Like to
Use Bluetooth**

We use Bluetooth to discover, connect
to, and share information with nearby
devices

Don't Allow

OK

User Authorization

NEW

User consent required for all Core Bluetooth API's

Applies to apps linked with any SDK

Can be modified in Settings

Required on iOS, watchOS, and tvOS

**"CBCClassic" Would Like to
Use Bluetooth**

We use Bluetooth to discover, connect
to, and share information with nearby
devices

Don't Allow

OK

User Authorization on watchOS

NEW



User Authorization on watchOS

NEW

Shared between iOS and watchOS



User Authorization on watchOS

NEW

Shared between iOS and watchOS

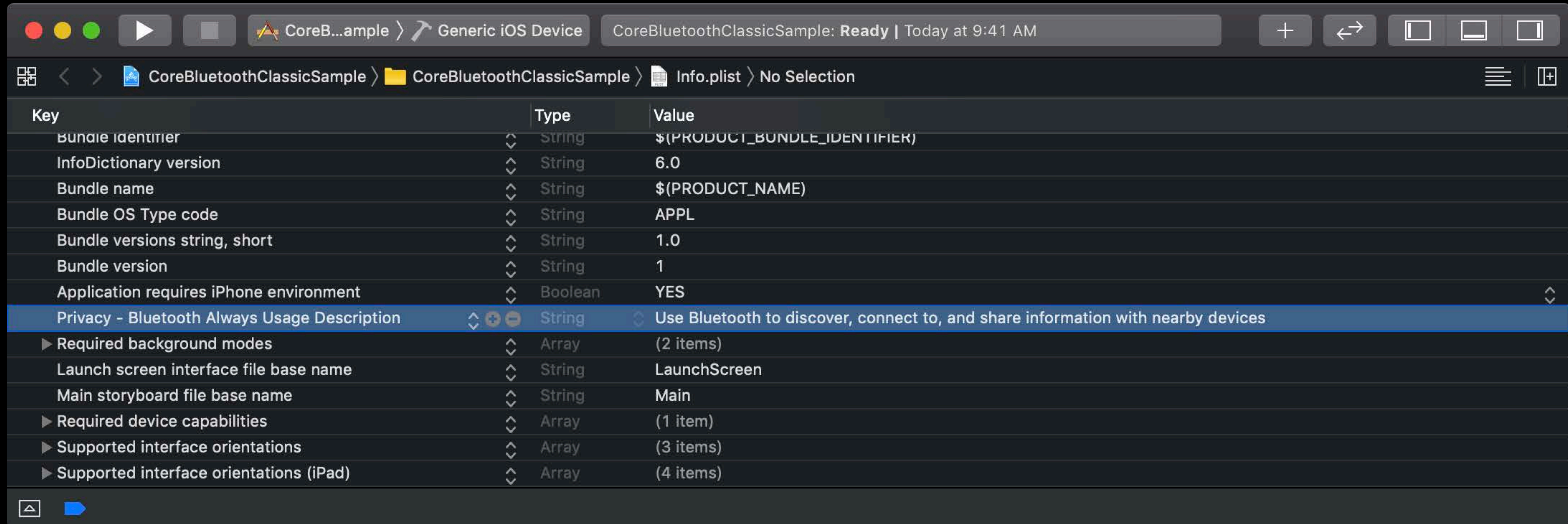
Except for standalone watchOS applications



Adoption

Purpose string

NEW



CoreBluetoothClassicSample: Ready | Today at 9:41 AM

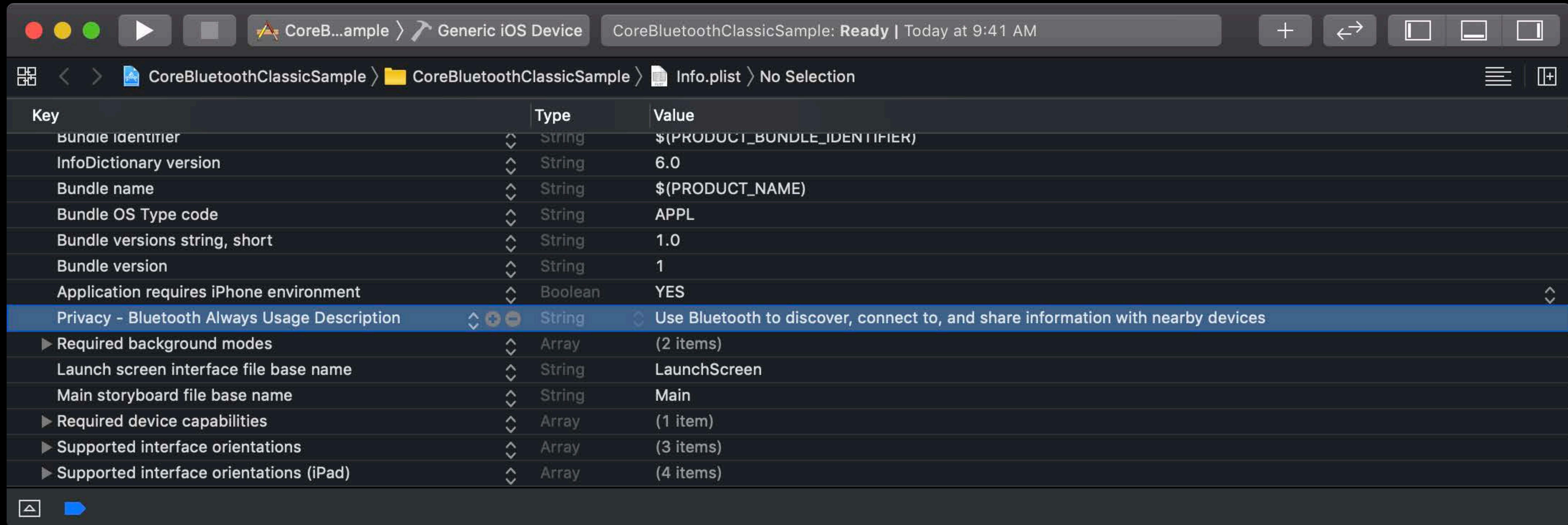
CoreBluetoothClassicSample > Info.plist > No Selection

Key	Type	Value
Bundle identifier	String	\$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version	String	6.0
Bundle name	String	\$(PRODUCT_NAME)
Bundle OS Type code	String	APPL
Bundle versions string, short	String	1.0
Bundle version	String	1
Application requires iPhone environment	Boolean	YES
Privacy - Bluetooth Always Usage Description	String	Use Bluetooth to discover, connect to, and share information with nearby devices
▶ Required background modes	Array	(2 items)
Launch screen interface file base name	String	LaunchScreen
Main storyboard file base name	String	Main
▶ Required device capabilities	Array	(1 item)
▶ Supported interface orientations	Array	(3 items)
▶ Supported interface orientations (iPad)	Array	(4 items)

Adoption

Purpose string

NEW



The screenshot shows the Xcode interface for editing the Info.plist file of a project named 'CoreBluetoothClassicSample'. The 'Privacy - Bluetooth Always Usage Description' key is selected, and its value is 'Use Bluetooth to discover, connect to, and share information with nearby devices'. The table below lists the keys and their values.

Key	Type	Value
Bundle identifier	String	\$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version	String	6.0
Bundle name	String	\$(PRODUCT_NAME)
Bundle OS Type code	String	APPL
Bundle versions string, short	String	1.0
Bundle version	String	1
Application requires iPhone environment	Boolean	YES
Privacy - Bluetooth Always Usage Description	String	Use Bluetooth to discover, connect to, and share information with nearby devices
▶ Required background modes	Array	(2 items)
Launch screen interface file base name	String	LaunchScreen
Main storyboard file base name	String	Main
▶ Required device capabilities	Array	(1 item)
▶ Supported interface orientations	Array	(3 items)
▶ Supported interface orientations (iPad)	Array	(4 items)



Core BluetoothClassicSample: [access] This app has crashed because it attempted to access privacy-sensitive data without a usage description. The app's Info.plist must contain an NSBluetoothAlwaysUsageDescription key with a string value explaining to the user how the app uses this data.

Adoption

New property



NEW

```
var authorization: CBManagerAuthorization { get }
```

```
enum CBManagerAuthorization : Int {
```

```
    init?(rawValue: Int)
```

```
    var rawValue: Int { get }
```

```
    case notDetermined
```

```
    case restricted
```

```
    case denied
```

```
    case allowedAlways
```

```
}
```

Adoption

New property



NEW

```
var authorization: CBManagerAuthorization { get }
```

```
enum CBManagerAuthorization : Int {  
    init?(rawValue: Int)  
    var rawValue: Int { get }  
    case notDetermined  
    case restricted  
    case denied  
    case allowedAlways  
}
```

Adoption

New property



NEW

```
var authorization: CBManagerAuthorization { get }
```

```
enum CBManagerAuthorization : Int {  
    init?(rawValue: Int)  
    var rawValue: Int { get }  
    case notDetermined  
    case restricted  
    case denied  
    case allowedAlways  
}
```

Adoption Flow



NEW

```
func centralManagerDidUpdateState(_ central: CBCentralManager)
func peripheralManagerDidUpdateState(_ peripheral: CBPeripheralManager)

open var state: CBManagerState { get }

// if (state == CBManagerState.unauthorized)
open var authorization: CBManagerAuthorization { get }
```

Adoption

Flow

NEW

```
func centralManagerDidUpdateState(_ central: CBCentralManager)
func peripheralManagerDidUpdateState(_ peripheral: CBPeripheralManager)

open var state: CBManagerState { get }

// if (state == CBManagerState.unauthorized)
open var authorization: CBManagerAuthorization { get }
```


Adoption Flow



NEW

```
func centralManagerDidUpdateState(_ central: CBCentralManager)
func peripheralManagerDidUpdateState(_ peripheral: CBPeripheralManager)

open var state: CBManagerState { get }

// if (state == CBManagerState.unauthorized)
open var authorization: CBManagerAuthorization { get }
```

Adoption

Flow

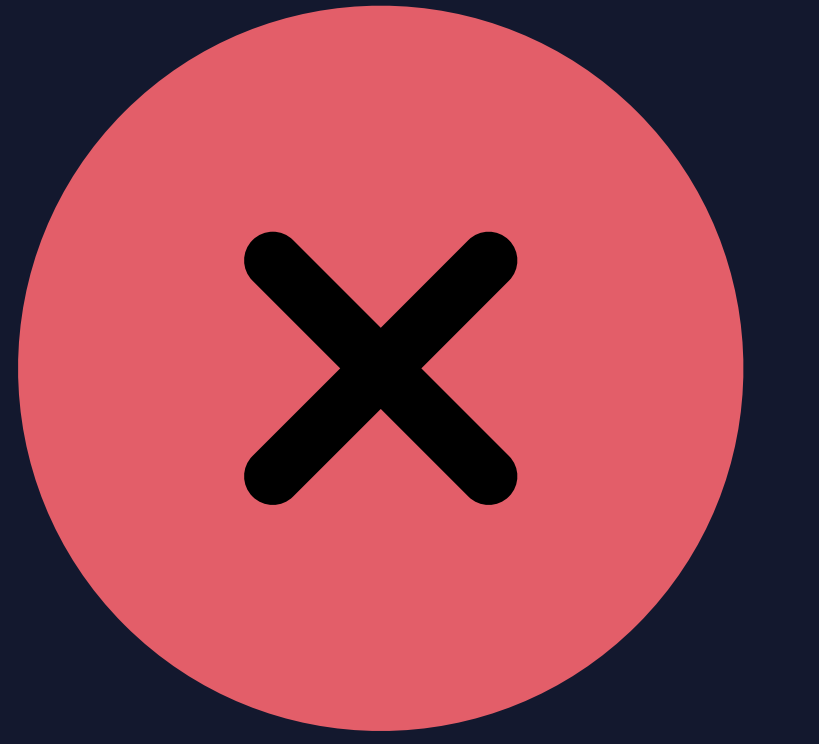
NEW

```
func centralManagerDidUpdateState(_ central: CBCentralManager)
func peripheralManagerDidUpdateState(_ peripheral: CBPeripheralManager)

open var state: CBManagerState { get }

// if (state == CBManagerState.unauthorized)
open var authorization: CBManagerAuthorization { get }
```

```
// Old managerDidUpdateState
func centralManagerDidUpdateState(_ central: CBCentralManager) {
    if (cbState == CBManagerState.poweredOn) {
        // Kick-off bluetooth functionality
    }
}
```



```
// Updated managerDidUpdateState
func centralManagerDidUpdateState(_ central: CBCentralManager) {
    switch central.state {
    case .unknown:
        // Handle state
    case .resetting:
        // Handle state
    case .unsupported:
        // Handle state
    case .unauthorized:
        if (central.authorization != CBUManagerAuthorization.allowedAlways) {
            // Prompt user to give permission
        }
    case .poweredOn:
        // Handle state
    case .poweredOff:
        // Handle state
    }
}
```



Enhancements

User authorization

Accessory notifications

Accessory Notifications

Accessory Notifications

Apple Notification Center Service

Accessory Notifications

Apple Notification Center Service

GATT server service

Accessory Notifications

Apple Notification Center Service

GATT server service

Allows accessories to receive notifications from iOS

ANCS Privacy Update

NEW

ANCS Privacy Update

NEW

User permission required to share notifications

ANCS Privacy Update

NEW

User permission required to share notifications

Prompted when accessory registers for notifications



ANCS Privacy Update

NEW

User permission required to share notifications

Prompted when accessory registers for notifications

Permissions can be changed in Settings



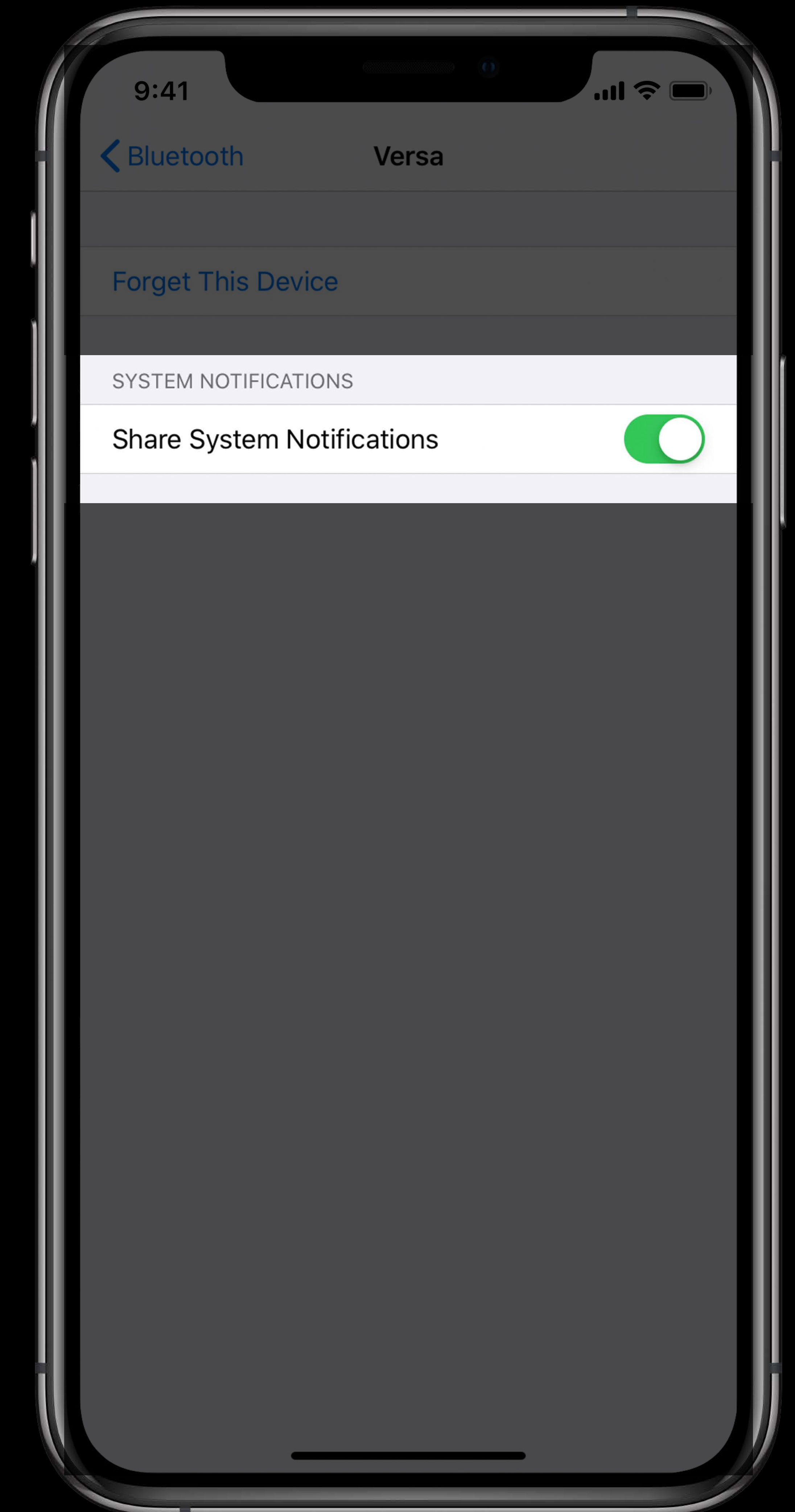
ANCS Privacy Update

NEW

User permission required to share notifications

Prompted when accessory registers for notifications

Permissions can be changed in Settings



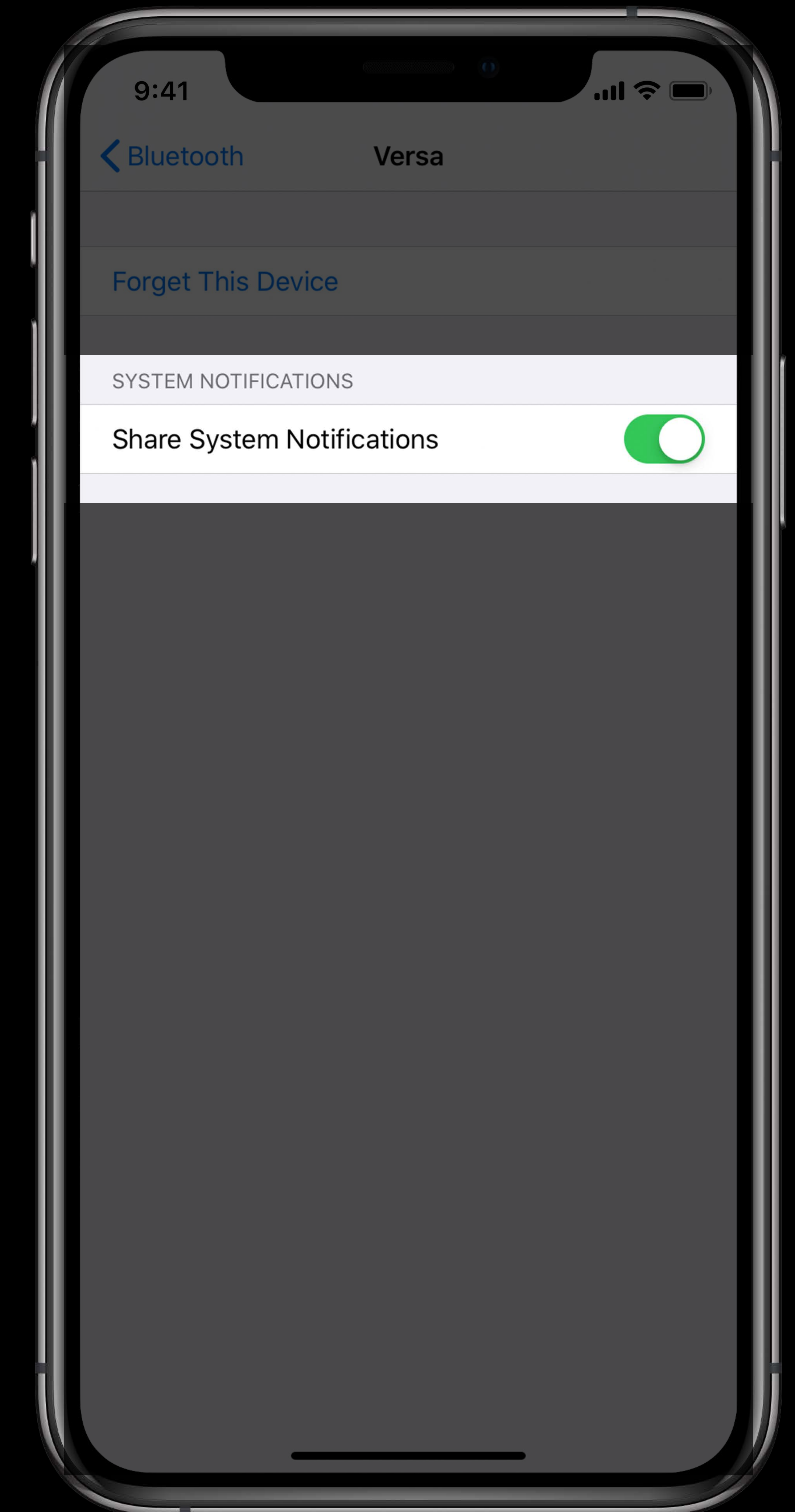
ANCS Privacy Update

NEW

User permission required to share notifications

Prompted when accessory registers for notifications

Permissions can be changed in Settings



New ANCS Privacy API

NEW

```
public let CBCConnectPeripheralOptionRequiresANCS: String
```

```
optional func centralManager(_central: CBCentralManager, didUpdateANCSAuthorizationFor  
peripheral: CBPeripheral)
```

```
open var ancsAuthorized: Bool { get }
```


New ANCS Privacy API

NEW

```
public let CBCentralManagerOptionRequiresANCS: String
```

```
optional func centralManager(_central: CBCentralManager, didUpdateANCSAuthorizationFor  
peripheral: CBPeripheral)
```

```
open var ancsAuthorized: Bool { get }
```

New ANCS Privacy API

NEW

```
public let CBCConnectPeripheralOptionRequiresANCS: String
```

```
optional func centralManager(_central: CBCentralManager, didUpdateANCSAuthorizationFor  
peripheral: CBPeripheral)
```

```
open var ancsAuthorized: Bool { get }
```

New ANCS Privacy API

NEW

```
public let CBCConnectPeripheralOptionRequiresANCS: String
```

```
optional func centralManager(_central: CBCentralManager, didUpdateANCSAuthorizationFor  
peripheral: CBPeripheral)
```

```
open var ancsAuthorized: Bool { get }
```

Best Practices



Best Practices



Invoke Core Bluetooth APIs only when required

Best Practices



Invoke Core Bluetooth APIs only when required

Scan and advertise for a limited duration

Best Practices



Invoke Core Bluetooth APIs only when required

Scan and advertise for a limited duration

Scan for specific service UUID(s)

Best Practices

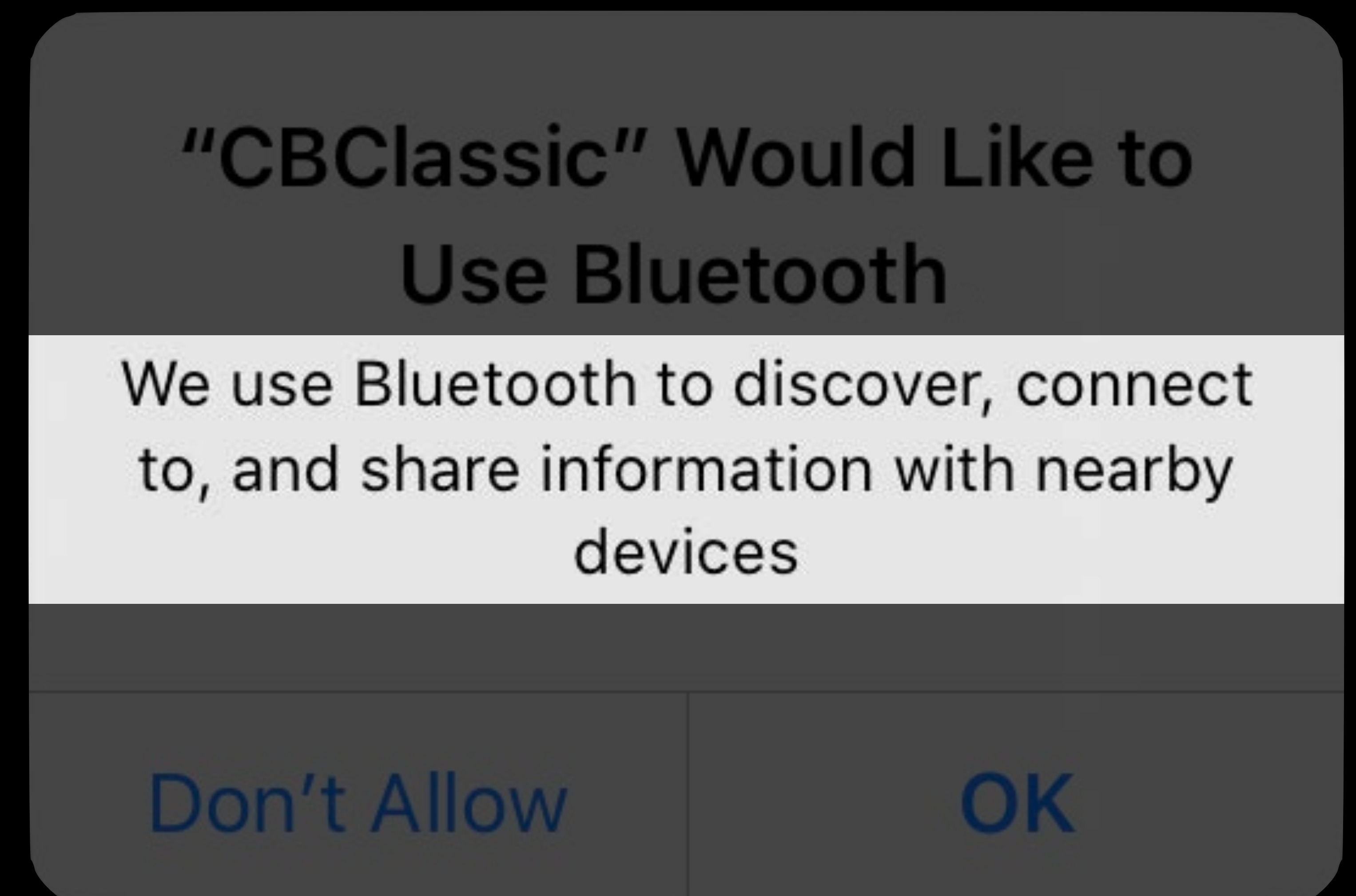


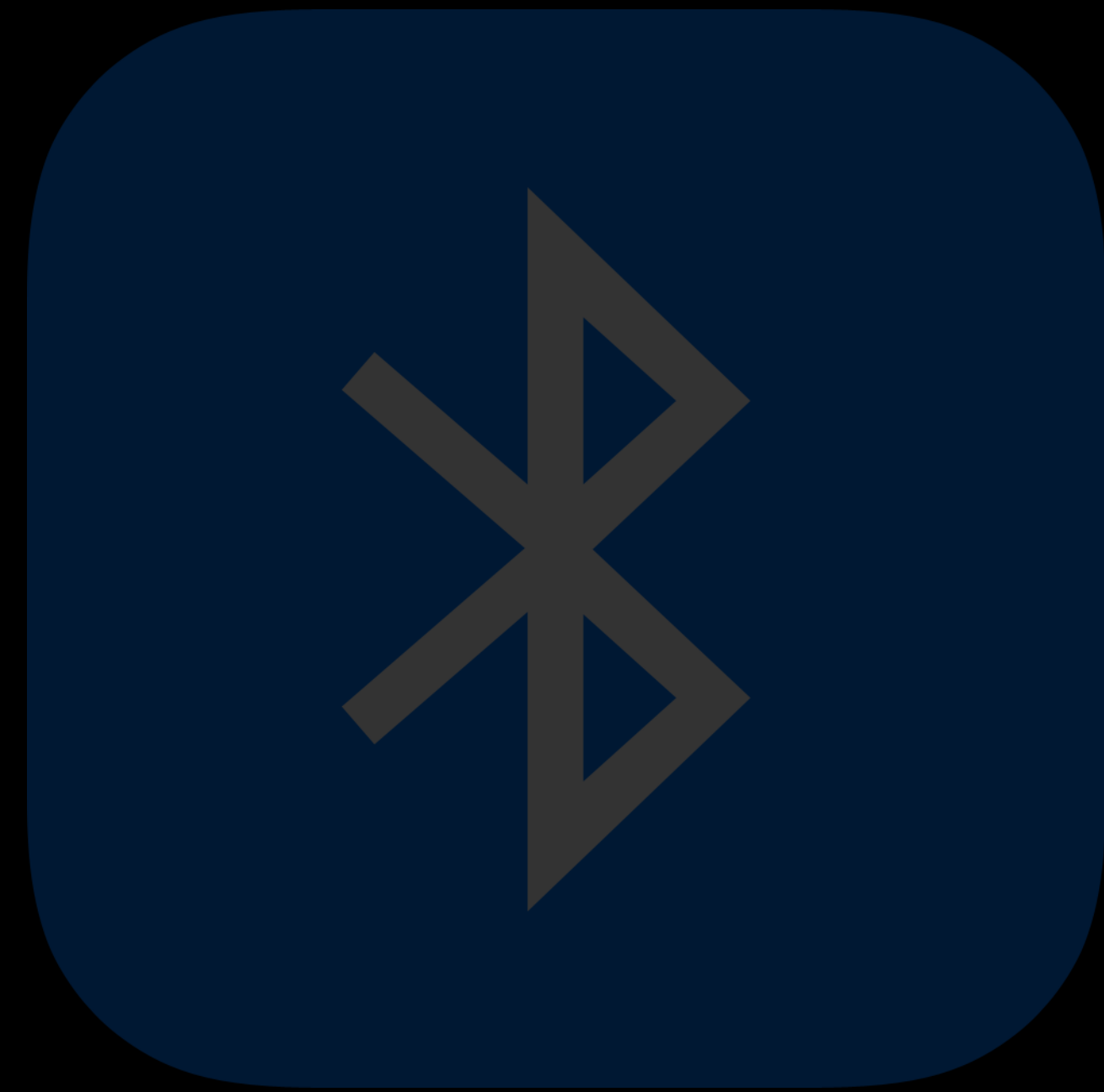
Invoke Core Bluetooth APIs only when required

Scan and advertise for a limited duration

Scan for specific service UUID(s)

Be transparent







Core Bluetooth PacketLogger

Duy Phan, Bluetooth Engineer

Overview

Overview

Bluetooth packet analysis application

Overview

Bluetooth packet analysis application

Visualizer for packet logs inside sysdiagnose

Overview

Bluetooth packet analysis application

Visualizer for packet logs inside sysdiagnose

Decode all protocols defined by Bluetooth SIG and Apple

Overview

Bluetooth packet analysis application

Visualizer for packet logs inside sysdiagnose

Decode all protocols defined by Bluetooth SIG and Apple

Rich filtering options

Overview

Bluetooth packet analysis application

Visualizer for packet logs inside sysdiagnose

Decode all protocols defined by Bluetooth SIG and Apple

Rich filtering options

Search by text or regex

Overview

Bluetooth packet analysis application

Visualizer for packet logs inside sysdiagnose

Decode all protocols defined by Bluetooth SIG and Apple

Rich filtering options

Search by text or regex

Comment and flag packets

Overview

Bluetooth packet analysis application

Visualizer for packet logs inside sysdiagnose

Decode all protocols defined by Bluetooth SIG and Apple

Rich filtering options

Search by text or regex

Comment and flag packets

Export raw data for analysis

Top Level View

Top Level View

WWDC-Dual

Start Clear Priorities Throughput Comment Flag Packet Multiple Packet Types All Devices All Handles Find Search Filter

Decode Packets 1109 total (0 Err / 0 HCI / 0 ACL / 0 SCO / 0 Misc) Δ 0.443 s

Time	Type	Handle	Addr	Decoded Packet
May 08 23:04:39.824	ATT Send	0x0044	00:09:A7:24:26:38	▶ Read By Type Request - Start Handle: 0x0016 - End Handle: 0x0017 - UUID: GATT Cl
May 08 23:04:39.825	HCI Event		24:18:97:6D:C0:AB	▶ LE - Advertising Report - 1 ReportNormal - Random - 24:18:97:6D:C0:AB -67 dBm
May 08 23:04:39.827	HCI Event		49:8B:4F:0D:B7:D0	▶ LE - Advertising Report - 1 ReportNormal - Random - 49:8B:4F:0D:B7:D0 -68 dBm
May 08 23:04:39.827	HCI Event		49:8B:4F:0D:B7:D0	▶ LE - Advertising Report - 1 ReportNormal - Random - 49:8B:4F:0D:B7:D0 -68 dBm
May 08 23:04:39.854	ATT Receive	0x0044	00:09:A7:24:26:38	▶ Find Information Request - Handle: 0x0009 - Service Changed
May 08 23:04:39.854	ATT Send	0x0044	00:09:A7:24:26:38	▶ Find Information Response
May 08 23:04:39.883	ATT Receive	0x0044	00:09:A7:24:26:38	▶ Error Response - Attribute Handle: 0x0016 - Error Code: Attribute Not Found (0x0
May 08 23:04:39.883	HCI Event		55:89:7B:4A:8A:AF	▶ LE - Advertising Report - 1 ReportNormal - Random - 55:89:7B:4A:8A:AF -79 dBm
May 08 23:04:39.884	ATT Send	0x0044	00:09:A7:24:26:38	▶ Read By Type Request - Start Handle: 0x0018 - End Handle: 0xFFFF - UUID: GATT Cl
May 08 23:04:39.897	HCI Event		00:DB:70:00:75:5A	▶ LE - Advertising Report - 1 ReportNormal - Public - 00:DB:70:00:75:5A -67 dBm
May 08 23:04:39.897	HCI Event		00:DB:70:00:75:5A	▶ LE - Advertising Report - 1 ReportNormal - Public - 00:DB:70:00:75:5A -67 dBm
May 08 23:04:39.914	ATT Receive	0x0044	00:09:A7:24:26:38	▶ Write Request - Handle: 0x0009 - Service Changed - Configuration - Indication
May 08 23:04:39.914	ATT Send	0x0044	00:09:A7:24:26:38	▶ Write Response
May 08 23:04:39.929	HCI Event		D0:03:4B:3B:7E:AB	▶ LE - Advertising Report - 1 ReportNormal - Public - D0:03:4B:3B:7E:AB -87 dBm
May 08 23:04:39.929	HCI Event		D0:03:4B:3B:7E:AB	▶ LE - Advertising Report - 1 ReportNormal - Public - D0:03:4B:3B:7E:AB -87 dBm
May 08 23:04:39.933	HCI Event		7E:75:7D:F9:82:03	▶ LE - Advertising Report - 1 ReportNormal - Random - 7E:75:7D:F9:82:03 -47 dBm
May 08 23:04:39.933	HCI Event		7E:75:7D:F9:82:03	▶ LE - Advertising Report - 1 ReportNormal - Random - 7E:75:7D:F9:82:03 -47 dBm
May 08 23:04:39.945	ATT Receive	0x0044	00:09:A7:24:26:38	▶ Read By Type Response - Packets: 4
May 08 23:04:39.947	ATT Send	0x0044	00:09:A7:24:26:38	▶ Read By Type Request - Start Handle: 0x0023 - End Handle: 0xFFFF - UUID: GATT Cl

Top Level View

WWDC-Dual

Start Clear Priorities Throughput Comment Flag Packet Packet Type Filter All Devices All Handles Find Search Filter

Decode Packets 1109 total (0 Err / 0 HCI / 0 ACL / 0 SCO / 0 Misc) Δ 0.443 s

Time	Type	Handle	Addr	Decoded Packet
May 08 23:04:39.824	ATT Send	0x0044	00:09:A7:24:26:38	▶ Read By Type Request - Start Handle: 0x0016 - End Handle: 0x0017 - UUID: GATT Cl
May 08 23:04:39.825	HCI Event		24:18:97:6D:C0:AB	▶ LE - Advertising Report - 1 ReportNormal - Random - 24:18:97:6D:C0:AB -67 dBm
May 08 23:04:39.827	HCI Event		49:8B:4F:0D:B7:D0	▶ LE - Advertising Report - 1 ReportNormal - Random - 49:8B:4F:0D:B7:D0 -68 dBm
May 08 23:04:39.827	HCI Event		49:8B:4F:0D:B7:D0	▶ LE - Advertising Report - 1 ReportNormal - Random - 49:8B:4F:0D:B7:D0 -68 dBm
May 08 23:04:39.854	ATT Receive	0x0044	00:09:A7:24:26:38	▶ Find Information Request - Handle: 0x0009 - Service Changed
May 08 23:04:39.854	ATT Send	0x0044	00:09:A7:24:26:38	▶ Find Information Response
May 08 23:04:39.883	ATT Receive	0x0044	00:09:A7:24:26:38	▶ Error Response - Attribute Handle: 0x0016 - Error Code: Attribute Not Found (0x0
May 08 23:04:39.883	HCI Event		55:89:7B:4A:8A:AF	▶ LE - Advertising Report - 1 ReportNormal - Random - 55:89:7B:4A:8A:AF -79 dBm
May 08 23:04:39.884	ATT Send	0x0044	00:09:A7:24:26:38	▶ Read By Type Request - Start Handle: 0x0018 - End Handle: 0xFFFF - UUID: GATT Cl
May 08 23:04:39.897	HCI Event		00:DB:70:00:75:5A	▶ LE - Advertising Report - 1 ReportNormal - Public - 00:DB:70:00:75:5A -67 dBm
May 08 23:04:39.897	HCI Event		00:DB:70:00:75:5A	▶ LE - Advertising Report - 1 ReportNormal - Public - 00:DB:70:00:75:5A -67 dBm
May 08 23:04:39.914	ATT Receive	0x0044	00:09:A7:24:26:38	▶ Write Request - Handle: 0x0009 - Service Changed - Configuration - Indication
May 08 23:04:39.914	ATT Send	0x0044	00:09:A7:24:26:38	▶ Write Response
May 08 23:04:39.929	HCI Event		D0:03:4B:3B:7E:AB	▶ LE - Advertising Report - 1 ReportNormal - Public - D0:03:4B:3B:7E:AB -87 dBm
May 08 23:04:39.929	HCI Event		D0:03:4B:3B:7E:AB	▶ LE - Advertising Report - 1 ReportNormal - Public - D0:03:4B:3B:7E:AB -87 dBm
May 08 23:04:39.933	HCI Event		7E:75:7D:F9:82:03	▶ LE - Advertising Report - 1 ReportNormal - Random - 7E:75:7D:F9:82:03 -47 dBm
May 08 23:04:39.933	HCI Event		7E:75:7D:F9:82:03	▶ LE - Advertising Report - 1 ReportNormal - Random - 7E:75:7D:F9:82:03 -47 dBm
May 08 23:04:39.945	ATT Receive	0x0044	00:09:A7:24:26:38	▶ Read By Type Response - Packets: 4
May 08 23:04:39.947	ATT Send	0x0044	00:09:A7:24:26:38	▶ Read By Type Request - Start Handle: 0x0023 - End Handle: 0xFFFF - UUID: GATT Cl

Top Level View

WWDC-Dual

Start Clear Priorities Throughput Comment Flag Packet Packet Type Filter Device Filter ACL Filter Find Filter

Decode Packets 1109 total (0 Err / 0 HCI / 0 ACL / 0 SCO / 0 Misc) Δ 0.443 s

Time	Type	Handle	Addr	Decoded Packet
May 08 23:04:39.824	ATT Send	0x0044	00:09:A7:24:26:38	▶ Read By Type Request - Start Handle: 0x0016 - End Handle: 0x0017 - UUID: GATT Cl
May 08 23:04:39.825	HCI Event		24:18:97:6D:C0:AB	▶ LE - Advertising Report - 1 ReportNormal - Random - 24:18:97:6D:C0:AB -67 dBm
May 08 23:04:39.827	HCI Event		49:8B:4F:0D:B7:D0	▶ LE - Advertising Report - 1 ReportNormal - Random - 49:8B:4F:0D:B7:D0 -68 dBm
May 08 23:04:39.827	HCI Event		49:8B:4F:0D:B7:D0	▶ LE - Advertising Report - 1 ReportNormal - Random - 49:8B:4F:0D:B7:D0 -68 dBm
May 08 23:04:39.854	ATT Receive	0x0044	00:09:A7:24:26:38	▶ Find Information Request - Handle: 0x0009 - Service Changed
May 08 23:04:39.854	ATT Send	0x0044	00:09:A7:24:26:38	▶ Find Information Response
May 08 23:04:39.883	ATT Receive	0x0044	00:09:A7:24:26:38	▶ Error Response - Attribute Handle: 0x0016 - Error Code: Attribute Not Found (0x0
May 08 23:04:39.883	HCI Event		55:89:7B:4A:8A:AF	▶ LE - Advertising Report - 1 ReportNormal - Random - 55:89:7B:4A:8A:AF -79 dBm
May 08 23:04:39.884	ATT Send	0x0044	00:09:A7:24:26:38	▶ Read By Type Request - Start Handle: 0x0018 - End Handle: 0xFFFF - UUID: GATT Cl
May 08 23:04:39.897	HCI Event		00:DB:70:00:75:5A	▶ LE - Advertising Report - 1 ReportNormal - Public - 00:DB:70:00:75:5A -67 dBm
May 08 23:04:39.897	HCI Event		00:DB:70:00:75:5A	▶ LE - Advertising Report - 1 ReportNormal - Public - 00:DB:70:00:75:5A -67 dBm
May 08 23:04:39.914	ATT Receive	0x0044	00:09:A7:24:26:38	▶ Write Request - Handle: 0x0009 - Service Changed - Configuration - Indication
May 08 23:04:39.914	ATT Send	0x0044	00:09:A7:24:26:38	▶ Write Response
May 08 23:04:39.929	HCI Event		D0:03:4B:3B:7E:AB	▶ LE - Advertising Report - 1 ReportNormal - Public - D0:03:4B:3B:7E:AB -87 dBm
May 08 23:04:39.929	HCI Event		D0:03:4B:3B:7E:AB	▶ LE - Advertising Report - 1 ReportNormal - Public - D0:03:4B:3B:7E:AB -87 dBm
May 08 23:04:39.933	HCI Event		7E:75:7D:F9:82:03	▶ LE - Advertising Report - 1 ReportNormal - Random - 7E:75:7D:F9:82:03 -47 dBm
May 08 23:04:39.933	HCI Event		7E:75:7D:F9:82:03	▶ LE - Advertising Report - 1 ReportNormal - Random - 7E:75:7D:F9:82:03 -47 dBm
May 08 23:04:39.945	ATT Receive	0x0044	00:09:A7:24:26:38	▶ Read By Type Response - Packets: 4
May 08 23:04:39.947	ATT Send	0x0044	00:09:A7:24:26:38	▶ Read By Type Request - Start Handle: 0x0023 - End Handle: 0xFFFF - UUID: GATT Cl

Hierarchical View

WWDC-Dual

Start Clear Priorities Throughput Comment Flag Packet Packet Type Filter Device Filter ACL Filter Find Filter

Multiple Packet Types All Devices All Handles

Q Search

Decode Packets 1109 total (0 Err / 0 HCI / 0 ACL / 0 SCO / 0 Misc) Δ 0.443 s

Time	Type	Handle	Addr	Decoded Packet
May 08 23:04:39.884	ATT Send	0x0044	00:09:A7:24:26:38	▼Read By Type Request - Start Handle: 0x0018 - End Handle: 0xFFFF - UUID: GATT Read By Type Request - Start Handle: 0x0018 - End Handle: 0xFFFF - UUID: GATT Opcode: 0x0008 Starting Handle: 0x0018 Ending Handle: 0xFFFF Attribute Type: 2803 (GATT Characteristic Declaration)
May 08 23:04:39.884	L2CAP Send	0x0044	00:09:A7:24:26:38	▼Channel ID: 0x0004 Length: 0x0007 (07) [08 18 00 FF FF 03 28] Channel ID: 0x0004 Length: 0x0007 (07) [08 18 00 FF FF 03 28] L2CAP Payload: 00000000: 0818 00FF FF03 28(
May 08 23:04:39.884	ACL Send	0x0044	00:09:A7:24:26:38	▼Data [Handle: 0x0044, Packet Boundary Flags: 0x0, Length: 0x000B (11)] Packet Boundary Flags: [00] 0x00 - Reserved For Future Use Broadcast Flags: [00] 0x00 - Point-to-point Data (0x000B Bytes)
May 08 23:04:39.884	ACL Send	0x0044	00:09:A7:24:26:38	▼00000000: 4400 0B00 0700 0400 0818 00FF FF03 28 D.....(00000000: 4400 0B00 0700 0400 0818 00FF FF03 28 D.....(
May 08 23:04:39.933	HCI Event		7E:75:7D:F9:82:03	▶LE - Advertising Report - 1 ReportNormal - Random - 7E:75:7D:F9:82:03 -47 dBm
May 08 23:04:39.945	ATT Receive	0x0044	00:09:A7:24:26:38	▶Read By Type Response - Packets: 4
May 08 23:04:39.947	ATT Send	0x0044	00:09:A7:24:26:38	▶Read By Type Request - Start Handle: 0x0023 - End Handle: 0xFFFF - UUID: GATT Cl

Hierarchical View

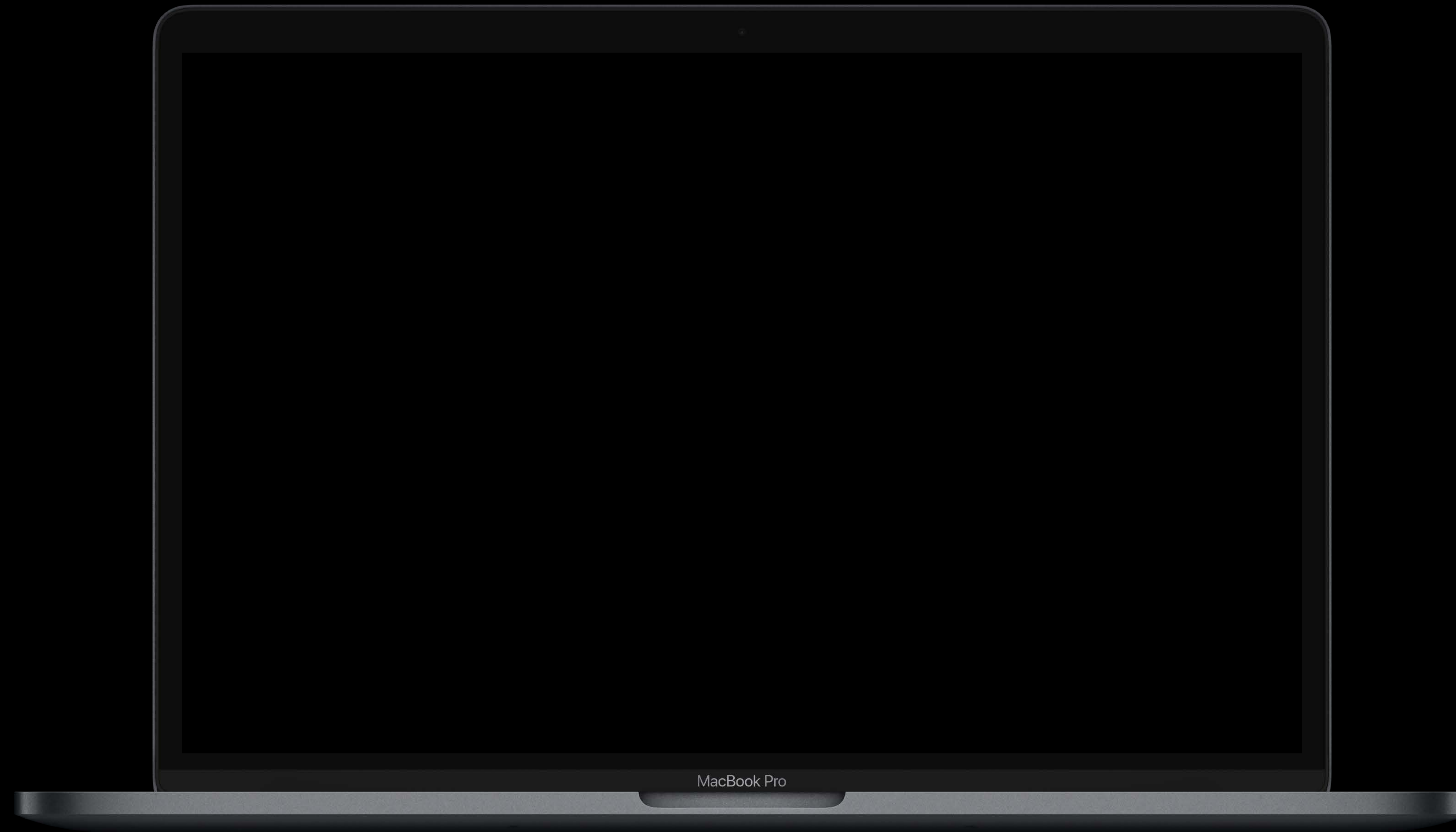
May 08 23:04:39.884	ATT Send	0x0044	00:09:A7:24:26:38	▼Read By Type Request - Start Handle: 0x0018 - End Handle: 0xFFFF - UUID: GATT Read By Type Request - Start Handle: 0x0018 - End Handle: 0xFFFF - UUID: GATT Opcode: 0x0008 Starting Handle: 0x0018 Ending Handle: 0xFFFF Attribute Type: 2803 (GATT Characteristic Declaration)
May 08 23:04:39.884	L2CAP Send	0x0044	00:09:A7:24:26:38	▼Channel ID: 0x0004 Length: 0x0007 (07) [08 18 00 FF FF 03 28] Channel ID: 0x0004 Length: 0x0007 (07) [08 18 00 FF FF 03 28] L2CAP Payload: 00000000: 0818 00FF FF03 28(
May 08 23:04:39.884	ACL Send	0x0044	00:09:A7:24:26:38	▼Data [Handle: 0x0044, Packet Boundary Flags: 0x0, Length: 0x000B (11)] Packet Boundary Flags: [00] 0x00 - Reserved For Future Use Broadcast Flags: [00] 0x00 - Point-to-point Data (0x000B Bytes)
May 08 23:04:39.884	ACL Send			▼00000000: 4400 0B00 0700 0400 0818 00FF FF03 28 D.....(00000000: 4400 0B00 0700 0400 0818 00FF FF03 28 D.....(

Hierarchical View

May 08 23:04:39.884	ATT Send	0x0044	00:09:A7:24:26:38	▼Read By Type Request - Start Handle: 0x0018 - End Handle: 0xFFFF - UUID: GATT Read By Type Request - Start Handle: 0x0018 - End Handle: 0xFFFF - UUID: GATT Opcode: 0x0008 Starting Handle: 0x0018 Ending Handle: 0xFFFF Attribute Type: 2803 (GATT Characteristic Declaration)
May 08 23:04:39.884	L2CAP Send	0x0044	00:09:A7:24:26:38	▼Channel ID: 0x0004 Length: 0x0007 (07) [08 18 00 FF FF 03 28] Channel ID: 0x0004 Length: 0x0007 (07) [08 18 00 FF FF 03 28] L2CAP Payload: 00000000: 0818 00FF FF03 28(
May 08 23:04:39.884	ACL Send	0x0044	00:09:A7:24:26:38	▼Data [Handle: 0x0044, Packet Boundary Flags: 0x0, Length: 0x000B (11)] Packet Boundary Flags: [00] 0x00 - Reserved For Future Use Broadcast Flags: [00] 0x00 - Point-to-point Data (0x000B Bytes)
May 08 23:04:39.884	ACL Send			▼00000000: 4400 0B00 0700 0400 0818 00FF FF03 28 D.....(00000000: 4400 0B00 0700 0400 0818 00FF FF03 28 D.....(

Live Capture

NEW



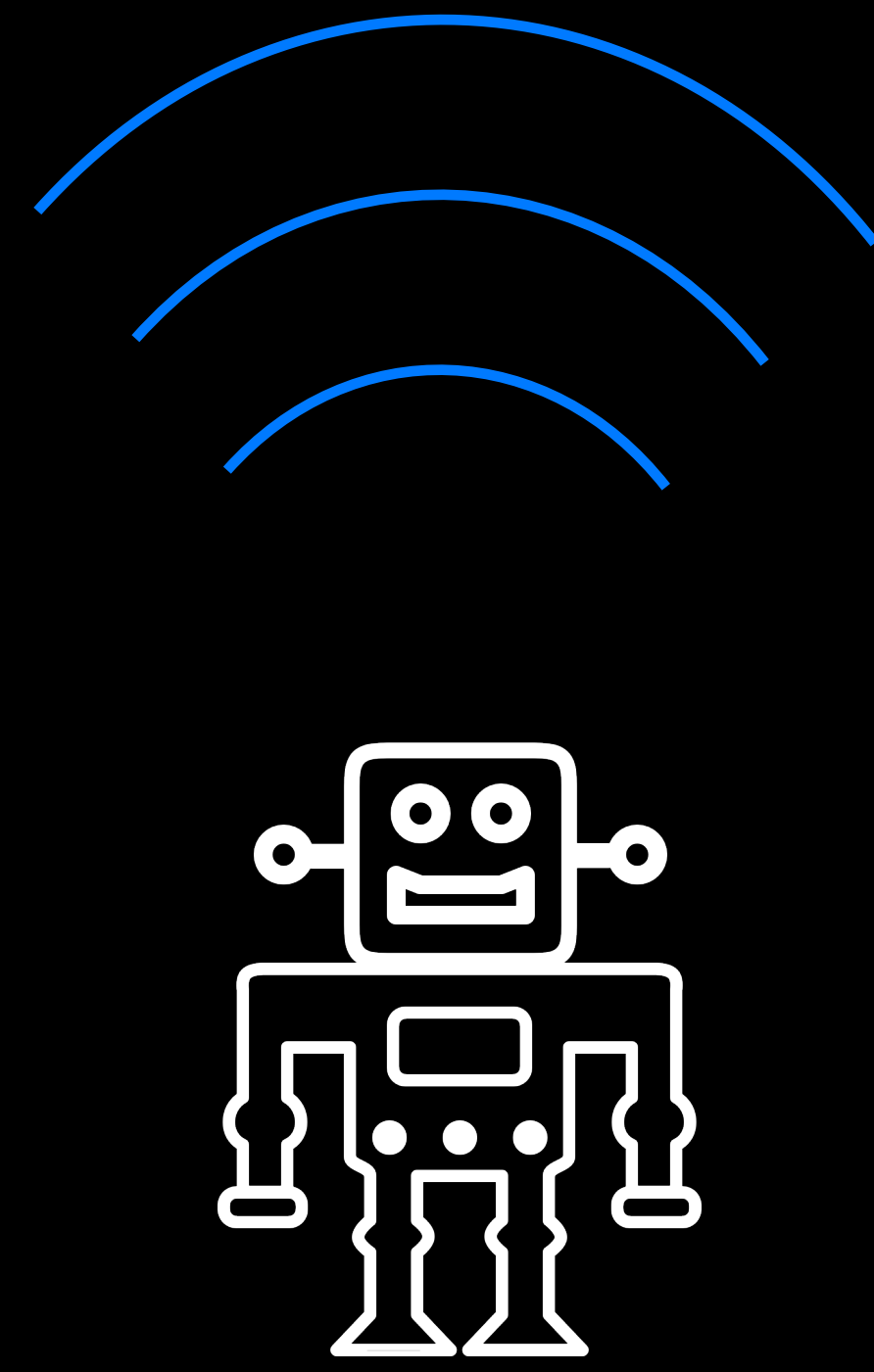
Live Capture

NEW



Live Capture

NEW



PacketLogger File Edit Control View Window Help

WWDC-Dual

Start Clear Priorities Throughput Comment Flag Packet Packet Type Filter Device Filter ACL Filter Find Q-Search Filter

1111 total (0 Err / 0 HCI / 0 ACL / 0 SCO / 0 Misc) 0.030 s

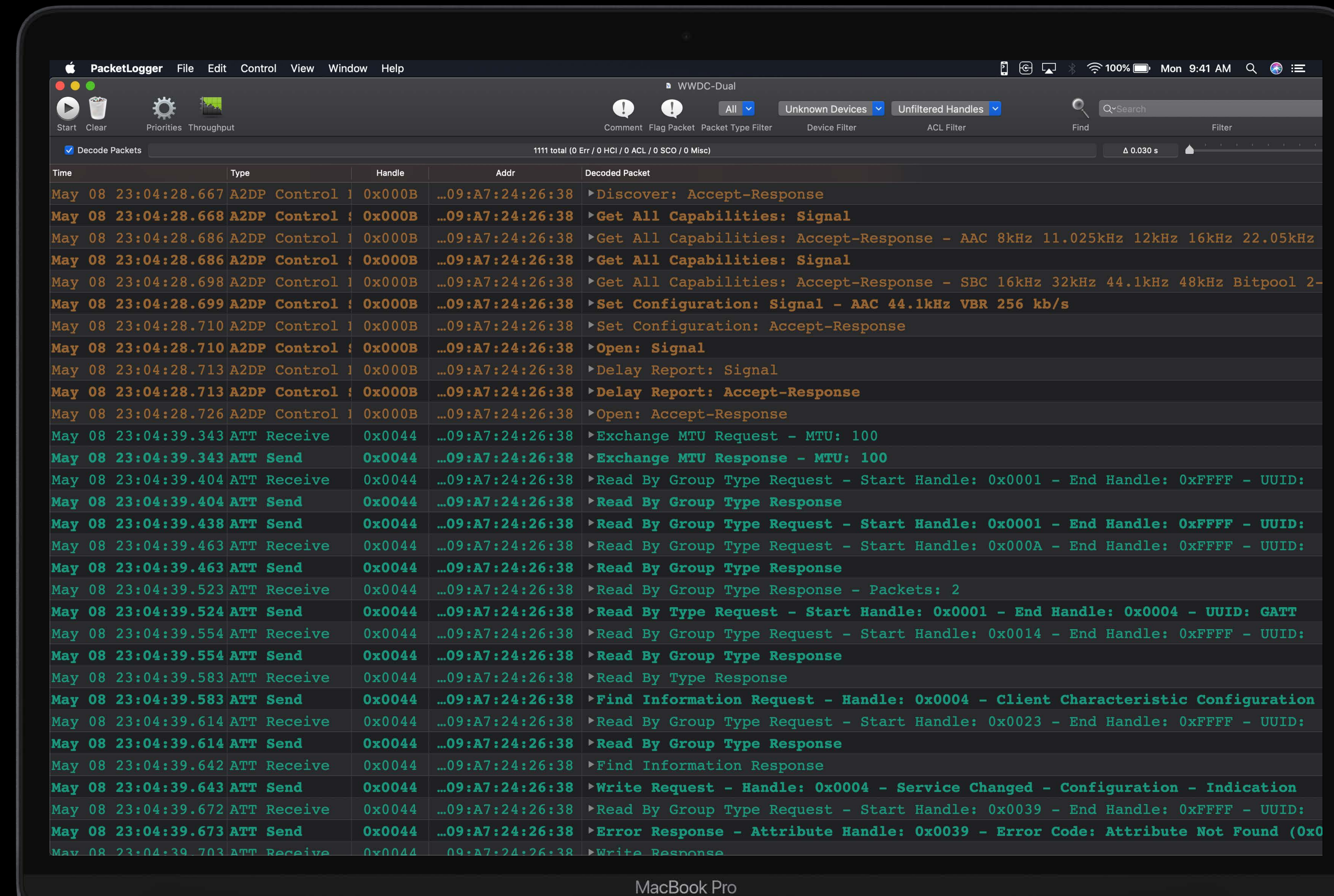
Time	Type	Handle	Addr	Decoded Packet
May 08 23:04:28.667	A2DP Control	0x000B	...09:A7:24:26:38	Discover: Accept-Response
May 08 23:04:28.668	A2DP Control	0x000B	...09:A7:24:26:38	Get All Capabilities: Signal
May 08 23:04:28.686	A2DP Control	0x000B	...09:A7:24:26:38	Get All Capabilities: Accept-Response - AAC 8kHz 11.025kHz 12kHz 16kHz 22.05kHz
May 08 23:04:28.686	A2DP Control	0x000B	...09:A7:24:26:38	Get All Capabilities: Signal
May 08 23:04:28.698	A2DP Control	0x000B	...09:A7:24:26:38	Get All Capabilities: Accept-Response - SBC 16kHz 32kHz 44.1kHz 48kHz Bitpool 2-
May 08 23:04:28.699	A2DP Control	0x000B	...09:A7:24:26:38	Set Configuration: Signal - AAC 44.1kHz VBR 256 kb/s
May 08 23:04:28.710	A2DP Control	0x000B	...09:A7:24:26:38	Set Configuration: Accept-Response
May 08 23:04:28.710	A2DP Control	0x000B	...09:A7:24:26:38	Open: Signal
May 08 23:04:28.713	A2DP Control	0x000B	...09:A7:24:26:38	Delay Report: Signal
May 08 23:04:28.713	A2DP Control	0x000B	...09:A7:24:26:38	Delay Report: Accept-Response
May 08 23:04:28.726	A2DP Control	0x000B	...09:A7:24:26:38	Open: Accept-Response
May 08 23:04:39.343	ATT Receive	0x0044	...09:A7:24:26:38	Exchange MTU Request - MTU: 100
May 08 23:04:39.343	ATT Send	0x0044	...09:A7:24:26:38	Exchange MTU Response - MTU: 100
May 08 23:04:39.404	ATT Receive	0x0044	...09:A7:24:26:38	Read By Group Type Request - Start Handle: 0x0001 - End Handle: 0xFFFF - UUID:
May 08 23:04:39.404	ATT Send	0x0044	...09:A7:24:26:38	Read By Group Type Response
May 08 23:04:39.438	ATT Send	0x0044	...09:A7:24:26:38	Read By Group Type Request - Start Handle: 0x0001 - End Handle: 0xFFFF - UUID:
May 08 23:04:39.463	ATT Receive	0x0044	...09:A7:24:26:38	Read By Group Type Request - Start Handle: 0x000A - End Handle: 0xFFFF - UUID:
May 08 23:04:39.463	ATT Send	0x0044	...09:A7:24:26:38	Read By Group Type Response
May 08 23:04:39.523	ATT Receive	0x0044	...09:A7:24:26:38	Read By Group Type Response - Packets: 2
May 08 23:04:39.524	ATT Send	0x0044	...09:A7:24:26:38	Read By Type Request - Start Handle: 0x0001 - End Handle: 0x0004 - UUID: GATT
May 08 23:04:39.554	ATT Receive	0x0044	...09:A7:24:26:38	Read By Group Type Request - Start Handle: 0x0014 - End Handle: 0xFFFF - UUID:
May 08 23:04:39.554	ATT Send	0x0044	...09:A7:24:26:38	Read By Group Type Response
May 08 23:04:39.583	ATT Receive	0x0044	...09:A7:24:26:38	Read By Type Response
May 08 23:04:39.583	ATT Send	0x0044	...09:A7:24:26:38	Find Information Request - Handle: 0x0004 - Client Characteristic Configuration
May 08 23:04:39.614	ATT Receive	0x0044	...09:A7:24:26:38	Read By Group Type Request - Start Handle: 0x0023 - End Handle: 0xFFFF - UUID:
May 08 23:04:39.614	ATT Send	0x0044	...09:A7:24:26:38	Read By Group Type Response
May 08 23:04:39.642	ATT Receive	0x0044	...09:A7:24:26:38	Find Information Response
May 08 23:04:39.643	ATT Send	0x0044	...09:A7:24:26:38	Write Request - Handle: 0x0004 - Service Changed - Configuration - Indication
May 08 23:04:39.672	ATT Receive	0x0044	...09:A7:24:26:38	Read By Group Type Request - Start Handle: 0x0039 - End Handle: 0xFFFF - UUID:
May 08 23:04:39.673	ATT Send	0x0044	...09:A7:24:26:38	Error Response - Attribute Handle: 0x0039 - Error Code: Attribute Not Found (0x0
May 08 23:04:39.703	ATT Receive	0x0044	...09:A7:24:26:38	Write Response

MacBook Pro



Live Capture

NEW



Live Capture

NEW

Live Capture

NEW

Install iOS 13 developer beta

Live Capture

NEW

Install iOS 13 developer beta

Install iOS Bluetooth developer logging profile

Live Capture

NEW

Install iOS 13 developer beta

Install iOS Bluetooth developer logging profile

Launch PacketLogger

Live Capture

NEW

Install iOS 13 developer beta

Install iOS Bluetooth developer logging profile

Launch PacketLogger

Connect your iOS device to your Mac

Live Capture



NEW

Install iOS 13 developer beta

Install iOS Bluetooth developer logging profile

Launch PacketLogger

Connect your iOS device to your Mac

Select File and "New iOS Trace"

Live Capture

NEW

Install iOS 13 developer beta

Install iOS Bluetooth developer logging profile

Launch PacketLogger

Connect your iOS device to your Mac

Select File and "New iOS Trace"

Indicator will appear on iOS device



Getting PacketLogger

Getting PacketLogger

Download "Additional Tools for Xcode"

Getting PacketLogger

Download "Additional Tools for Xcode"

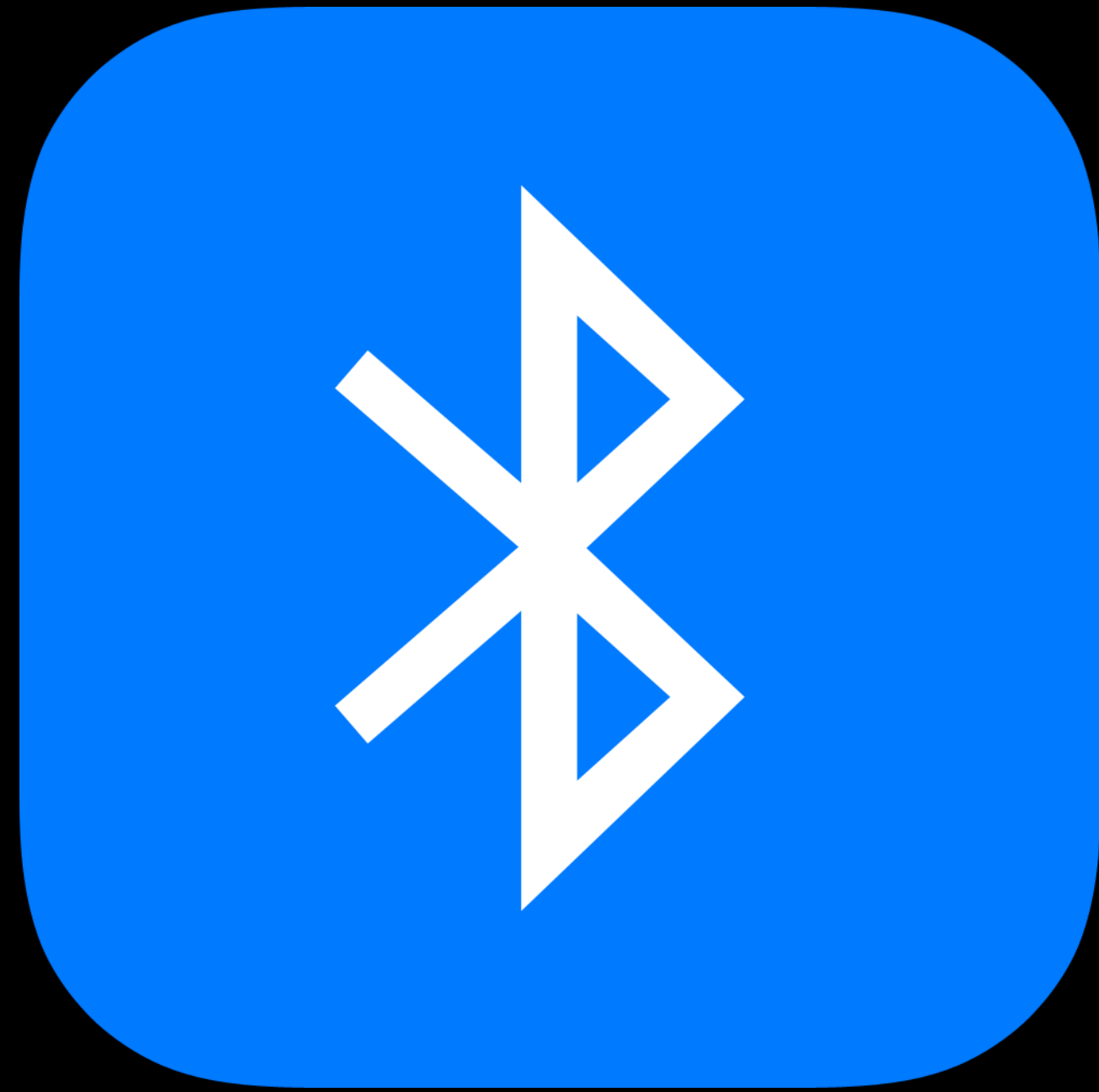
PacketLogger is inside the Hardware folder

Getting PacketLogger

Download "Additional Tools for Xcode"

PacketLogger is inside the Hardware folder

For best performance run with macOS Catalina



Summary

Summary

Use chipset with latest Bluetooth standard

Summary

Use chipset with latest Bluetooth standard

Build Core Bluetooth apps for BR/EDR devices

Summary

Use chipset with latest Bluetooth standard

Build Core Bluetooth apps for BR/EDR devices

Protect user privacy

Summary

Use chipset with latest Bluetooth standard

Build Core Bluetooth apps for BR/EDR devices

Protect user privacy

Take advantage of the developer beta

Summary

Use chipset with latest Bluetooth standard

Build Core Bluetooth apps for BR/EDR devices

Protect user privacy

Take advantage of the developer beta

Refer to [Accessory Design Guidelines](#) for Apple devices

Summary

Use chipset with latest Bluetooth standard

Build Core Bluetooth apps for BR/EDR devices

Protect user privacy

Take advantage of the developer beta

Refer to Accessory Design Guidelines for Apple devices

Apple is here to help

More Information

developer.apple.com/wwdc19/901

Core Bluetooth Lab

Friday, 4:00

