

# Working with iOS Accessories

Session 201

**Emily Schubert**

Manager, Accessory Interface  
iPod | iPhone | iPad Accessories

These are confidential sessions—please refrain from streaming, blogging, or taking pictures

# Introduction

- iOS provides interfaces for applications to interact with external devices
- New opportunities in iOS 5

# Working with iOS Accessories

- AirPlay
- Bluetooth
- Accessory updates

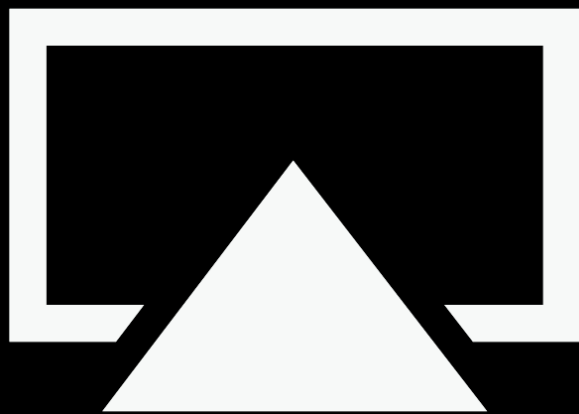
# AirPlay

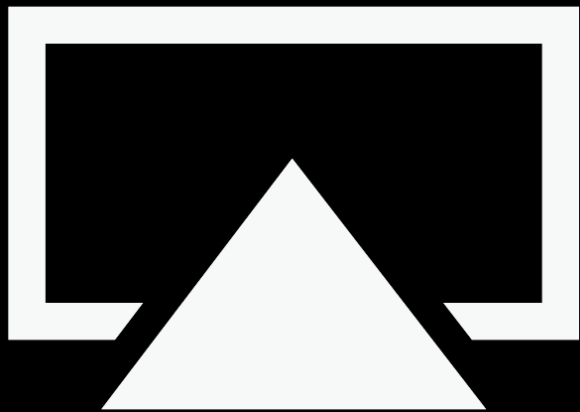
**Peter Langenfeld**

Feature Manager

iPod | iPhone | iPad Accessories







**Audio**

**Video**

**Photos**

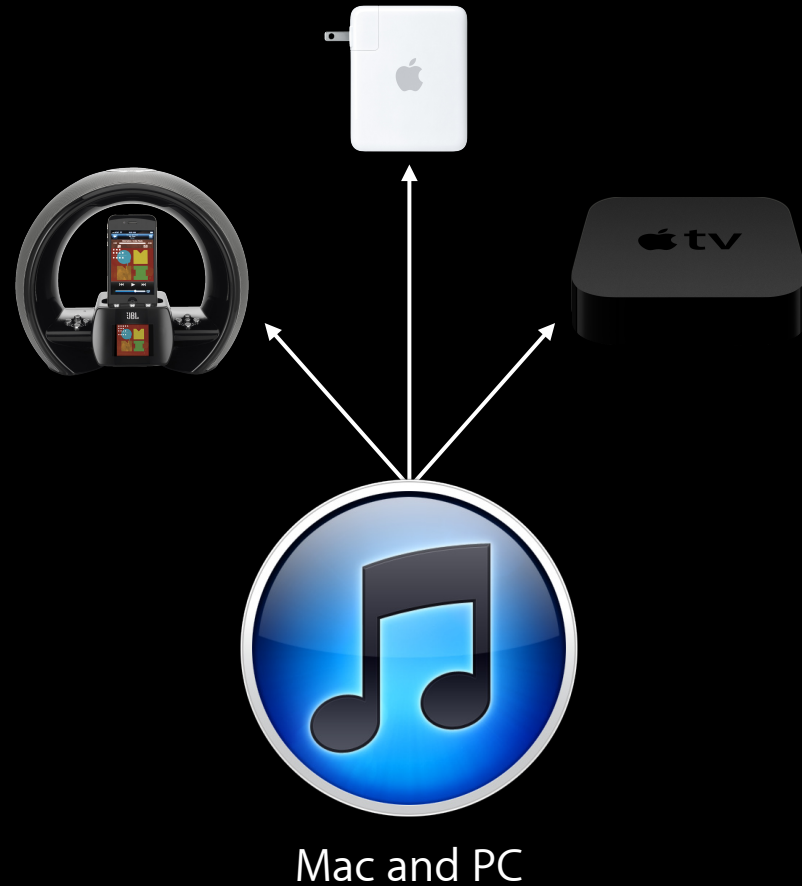
**Mirroring**



**Supports standard Wi-Fi**

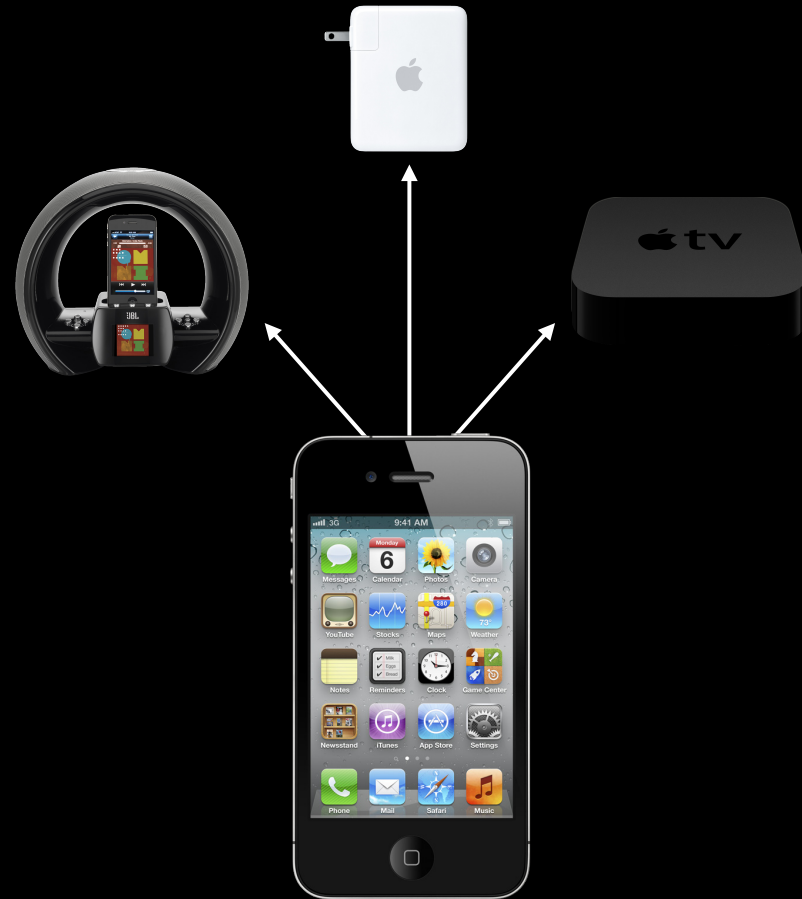
# Audio

- Multiple endpoints in sync
- Discrete volume control
- Remote app
- ALAC and AAC
  - ~2 second latency



# Audio

- Stream audio
- Volume control
- Any app can route audio
- ALAC and AAC
  - ~2 second latency



# Video

- Stream video
- Any app can route video
  - Local content
  - Web apps



# Photos

- Stream photos
- Play slideshows
- Use transitions



# AirPlay Mirroring

- Just like wired mirroring
- Games, web, apps
- Rotate screen on TV
- Any app can route dedicated screen UI
  - ~100ms latency



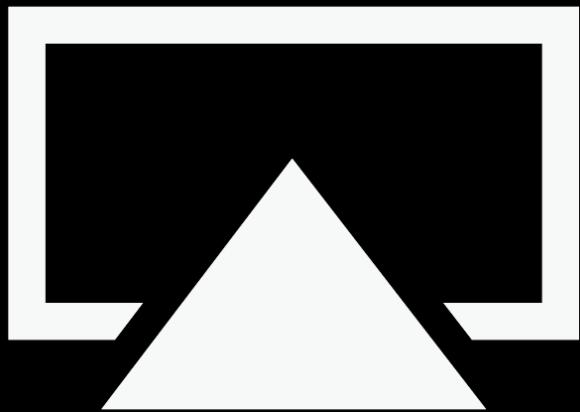


# AirPlay Accessories



# AirPlay Accessories





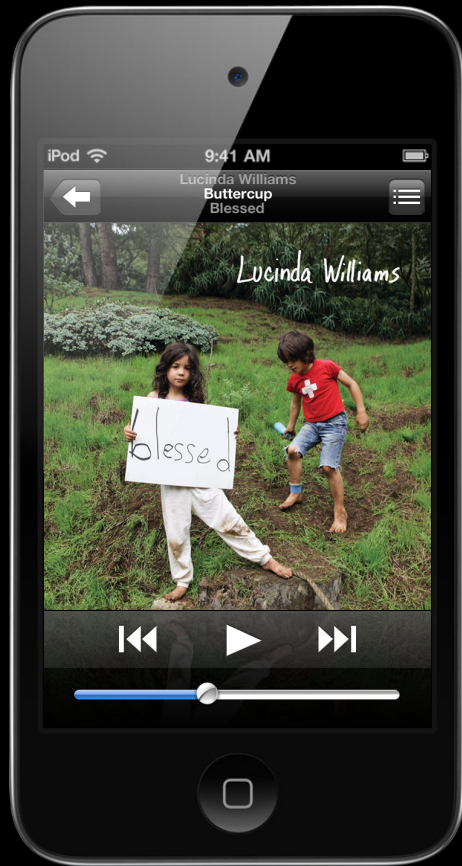
**Audio**

**Video**

**Photos**

**Mirroring**

# Using AirPlay



# Using AirPlay



# Using AirPlay

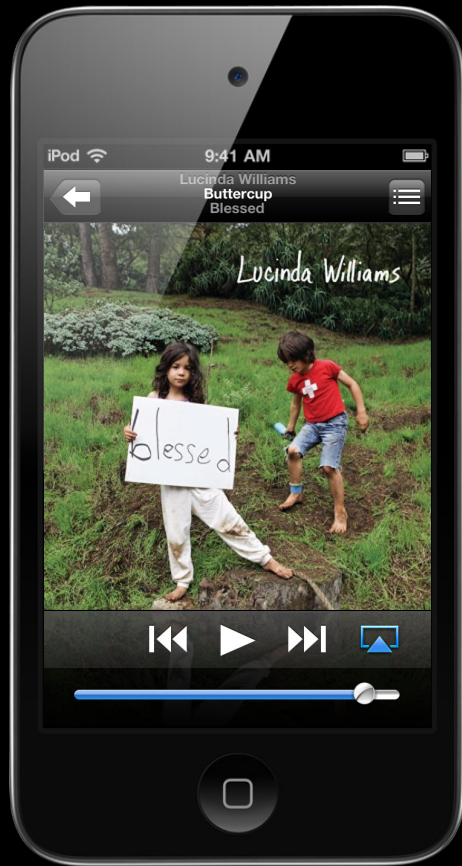


# Using AirPlay





# Using AirPlay





# Using AirPlay



# Using AirPlay in Your App

- Audio
  - Use `MPVolumeView`
- Video
  - Use `MPMoviePlayerController` or AV Foundation
- System handles the rest
  - Presence of AirPlay devices triggers route button appearance
  - User can access system volume levels, video routes, and audio routes

# Adding a Volume View to Your Hierarchy

- Make sure you **do not** hide the route button
  - The route button is visible by default, don't override this behavior

```
myVolumeView.showsRouteButton = NO;
```

# Allowing AirPlay Video

- MPMoviePlayerController

```
setAllowsAirPlay:YES
```

- QT Plug-in

```
airplay="allow"
```

- HTML5 video element

```
x-webkit-airplay="allow"
```

# Using Mirroring

5



# Using Mirroring

5



# Using Mirroring

5



# Using Mirroring

5





# Using Mirroring

5



# Using Mirroring

5



# Using Mirroring

5



# Using Mirroring

5



# Using Mirroring

5



# Using Mirroring in Your App

- Use `UIScreen` to give the user a tailored experience with your app
- System handles mirroring automatically
  - Presence of AirPlay devices triggers route button appearance
  - User can access system volume levels, video routes, and audio routes



# Using Metadata

5



# Using Metadata

5





# Using Metadata





# Providing Metadata From Your App



- Provide all available Now Playing information
- System displays information in a consistent manner for all apps
  - No direct control over which information is displayed, or its formatting

# Providing Now Playing Information



- Create dictionary with metadata

```
myinfo = [NSDictionary dictionaryWithObjectsAndKeys:  
    @"Lucinda Williams", MPMediaItemPropertyArtist,  
    @"Buttercup", MPMediaItemPropertyTitle,  
    nil];
```

- Pass dictionary to Now Playing Info Center

```
MPNowPlayingInfoCenter *infoCenter;  
  
infoCenter = [MPNowPlayingInfoCenter defaultCenter];  
infoCenter.nowPlayingInfo = myInfo;
```

# Providing Now Playing Information



- Create dictionary with metadata

```
myinfo = [NSDictionary dictionaryWithObjectsAndKeys:  
    @"Lucinda Williams", MPMediaItemPropertyArtist,  
    @"Buttercup", MPMediaItemPropertyTitle,  
    nil];
```

- Pass dictionary to Now Playing Info Center

```
MPNowPlayingInfoCenter *infoCenter;  
  
infoCenter = [MPNowPlayingInfoCenter defaultCenter];  
infoCenter.nowPlayingInfo = myInfo;
```

# Metadata

Album Title

Album Track Count

Album Track Number

Artist

Artwork

Composer

Disc Count

Disc Number

Genre

Persistent ID

Playback Duration

Title

Elapsed Playback Time

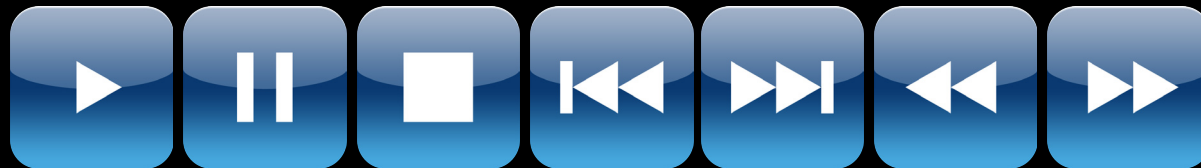
Playback Rate

Playback Queue Index

Playback Queue Count

# Using Remote Playback Controls

- AirPlay
- Bluetooth
- MFi accessories
- Headphone remote



# Supporting Remote Controls in Your App

- Ask for remote control events

```
[[UIApplication sharedApplication] beginReceivingRemoteControlEvents];
```

- Add remote control event handler

```
-(void)remoteControlReceivedWithEvent:(UIEvent *)event {  
    switch(event.subType) {  
        case UIEventSubtypeRemoteControlPlay:  
            // Respond to Play event  
            break;  
    }  
}
```

- Events are only sent to Now Playing app



# Supporting Remote Controls in Your App

- Ask for remote control events

```
[[UIApplication sharedApplication] beginReceivingRemoteControlEvents];
```

- Add remote control event handler

```
-(void)remoteControlReceivedWithEvent:(UIEvent *)event {  
    switch(event.subType) {  
        case UIEventSubtypeRemoteControlPlay:  
            // Respond to Play event  
            break;  
    }  
}
```

- Events are only sent to Now Playing app

# Remote Playback Controls

Play

Pause

Play/Pause Toggle

Stop

Next Track

Previous Track

Begin Seeking Backward

End Seeking Backward

Begin Seeking Forward

End Seeking Forward

# Summary

- Use standard audio and video controls
  - **Do not** hide the route button
  - Allow AirPlay video
- Provide as much metadata as possible to the system
- Handle remote playback controls

# AirPlay for 3rd Party Accessories



# AirPlay for 3rd Party Accessories

- License AirPlay audio through MFi Program  
<http://developer.apple.com/mfi/>



# Product Best Practices

- Take full advantage of registered information
- Be available to the user whenever possible
- Getting on the network is key

# iOS Bluetooth

**Brian Tucker**

Senior Software Engineering Manager  
Mobile Bluetooth Technologies

# Agenda



- Audience
  - Bluetooth accessory manufacturers
  - iOS application developers
- Topics
  - iOS 5 Updates
  - State of iOS Bluetooth
  - Best Practices for working with Bluetooth and iOS
- Goal
  - Create the best possible Bluetooth customer experience



# New in iOS 5



## Audio Visual Remote Control Profile (AVRCP) 1.4

- Now Playing metadata
  - Artist
  - Title
  - Playback Duration
  - Album Title
  - Album Track Number
  - Album Track Count
  - Genre
- Volume control and reporting
- API for AVRCP control and metadata reporting
  - Same API used for AirPlay

# New in iOS 5



- Wide Band Speech (WBS) codec for Handsfree Profile (HFP)
  - 16 kHz audio samples vs. 8 kHz with no increase in bandwidth
  - Much better audio quality between iOS and a BT HFP accessory
  - Improved voice recognition, recording and FaceTime
- Apps can now send and receive audio from handsfree devices
  - Standard CoreAudio device
- Multiple concurrent Bluetooth MFi accessories

# State of iOS Bluetooth

## Standard protocols

Hands-Free Profile	HFP 1.6
Phone Book Access Profile	PBAP 1.0
Advanced Audio Distribution Profile	A2DP 1.2
Audio Visual Remote Control Profile	AVRCP 1.4
Human Interface Device Profile	HID 1.0
Personal Area Networking	PAN 1.0
Device ID Profile	DID 1.2

## Custom protocols

iPod Accessory Protocol

# Bluetooth Best Practices and Apple Guidelines



Download from iOS Dev Center

# Best Practices Overview

- Use
  - BT 2.1 + EDR
  - BT 3.0
- Use
  - Device ID Profile (DID)
  - Secure Simple Pairing (SSP)
  - Sniff Mode when possible
  - Extended Inquiry Response when possible

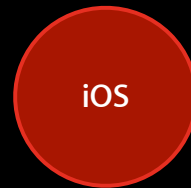
# “Help Me Help You”

## Role and topology management

- Bluetooth Master controls timing
- Bluetooth Slave must adhere to the Master's timing

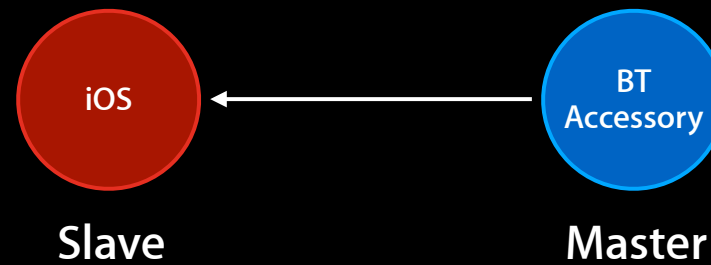
# “Help Me Help You”

Role and Topology Management



# “Help Me Help You”

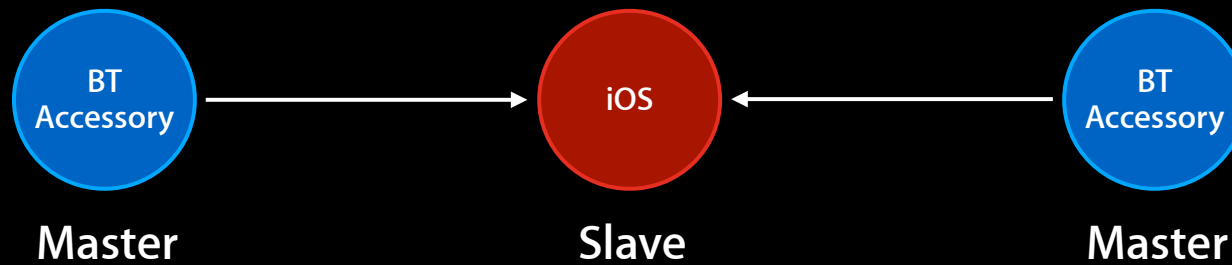
## Role and Topology Management





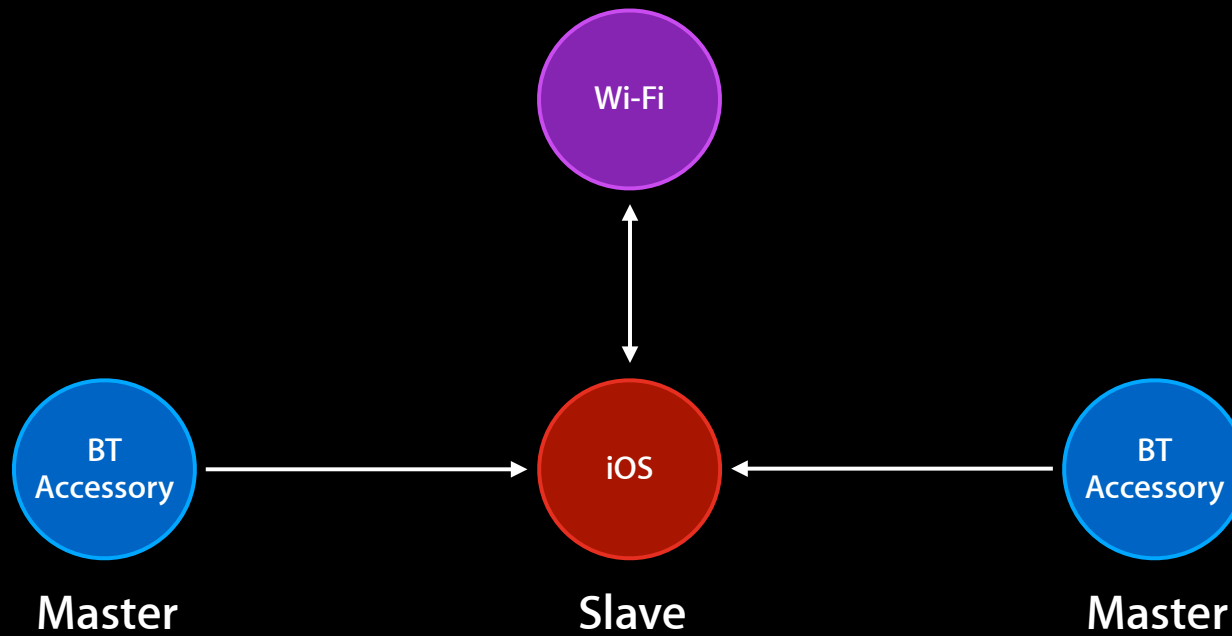
# “Help Me Help You”

## Role and Topology Management



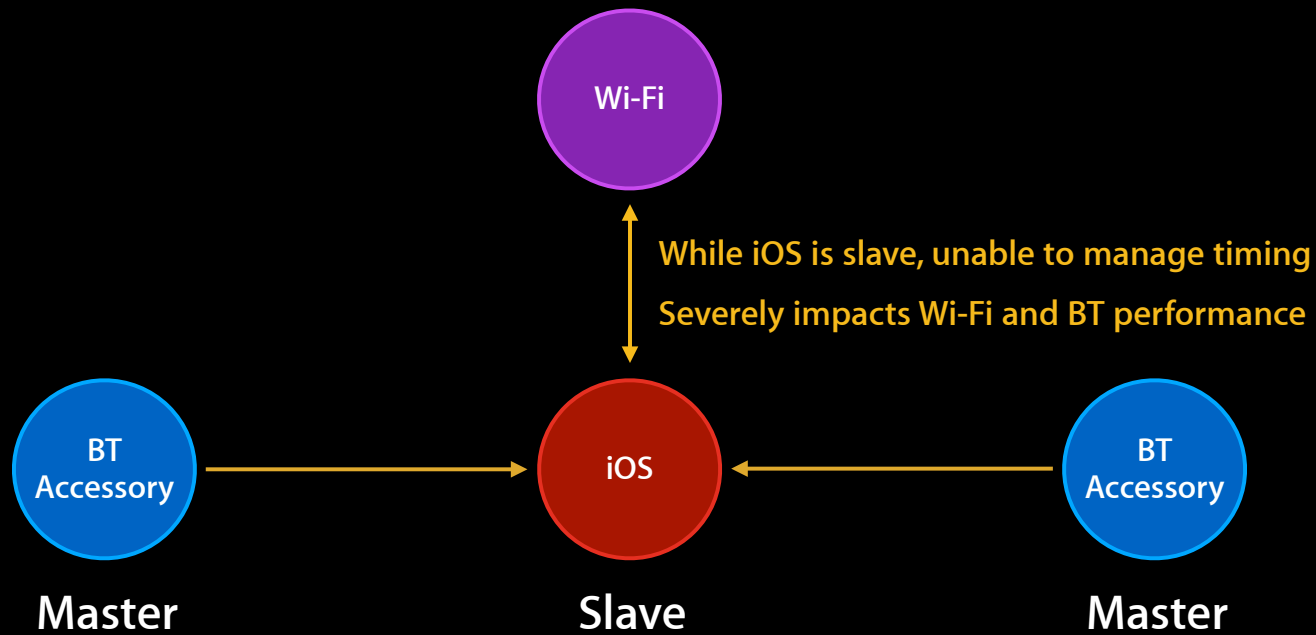
# “Help Me Help You”

## Role and Topology Management



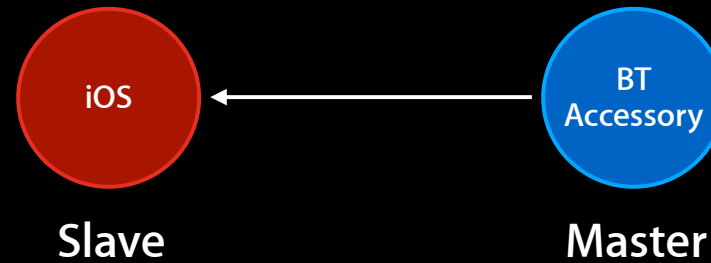
# “Help Me Help You”

## Role and Topology Management



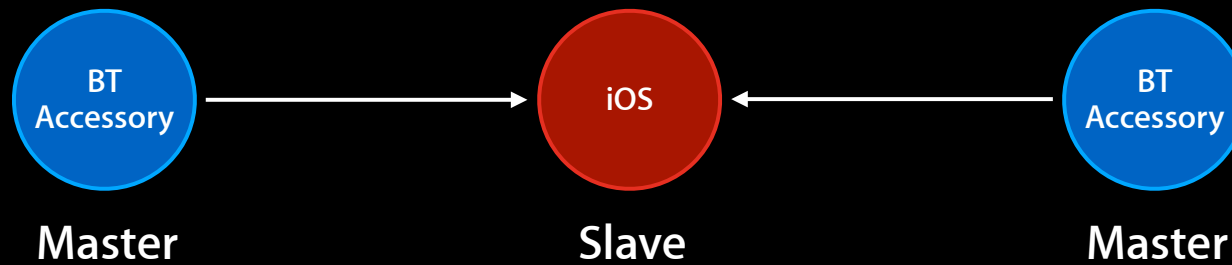
# “Help Me Help You”

## Role and Topology Management



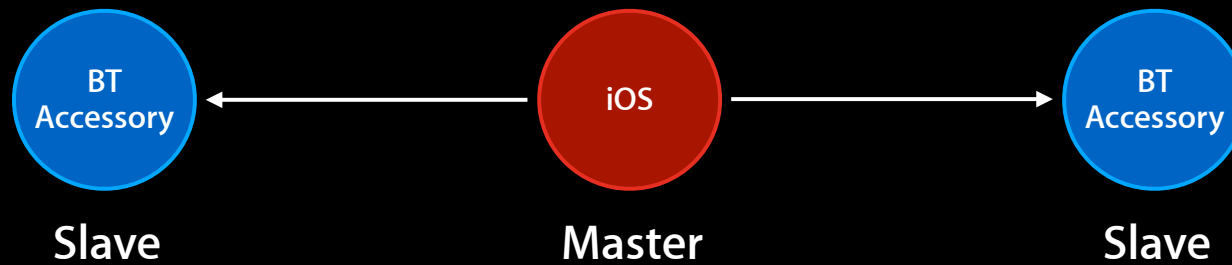
# “Help Me Help You”

## Role and Topology Management



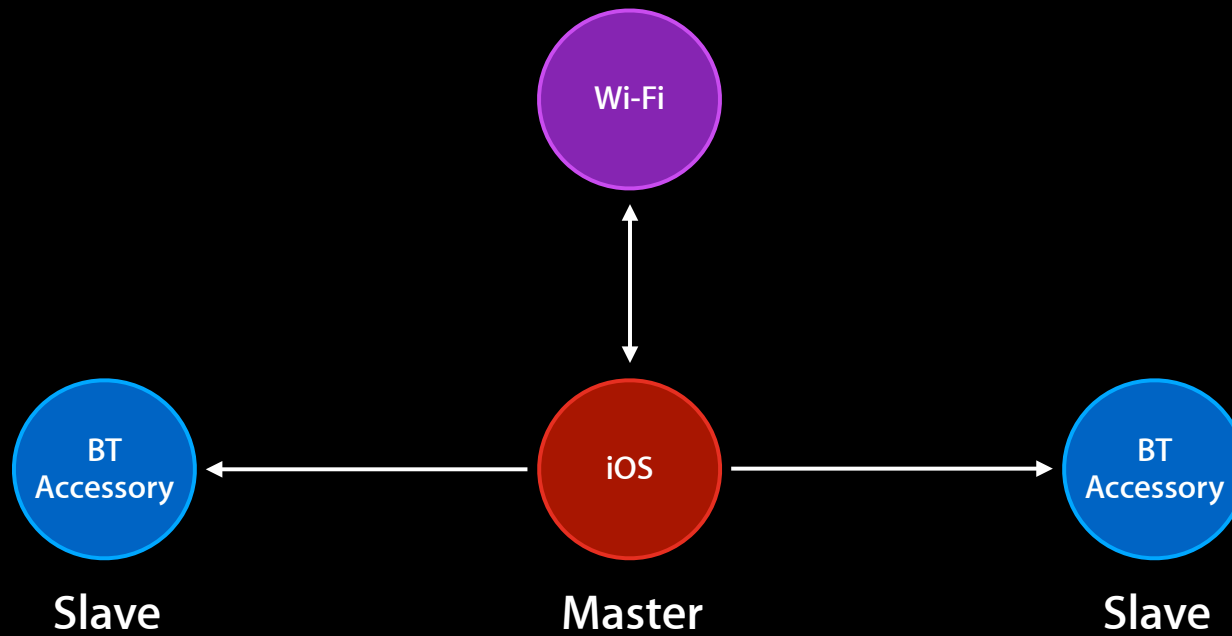
# “Help Me Help You”

## Role and Topology Management



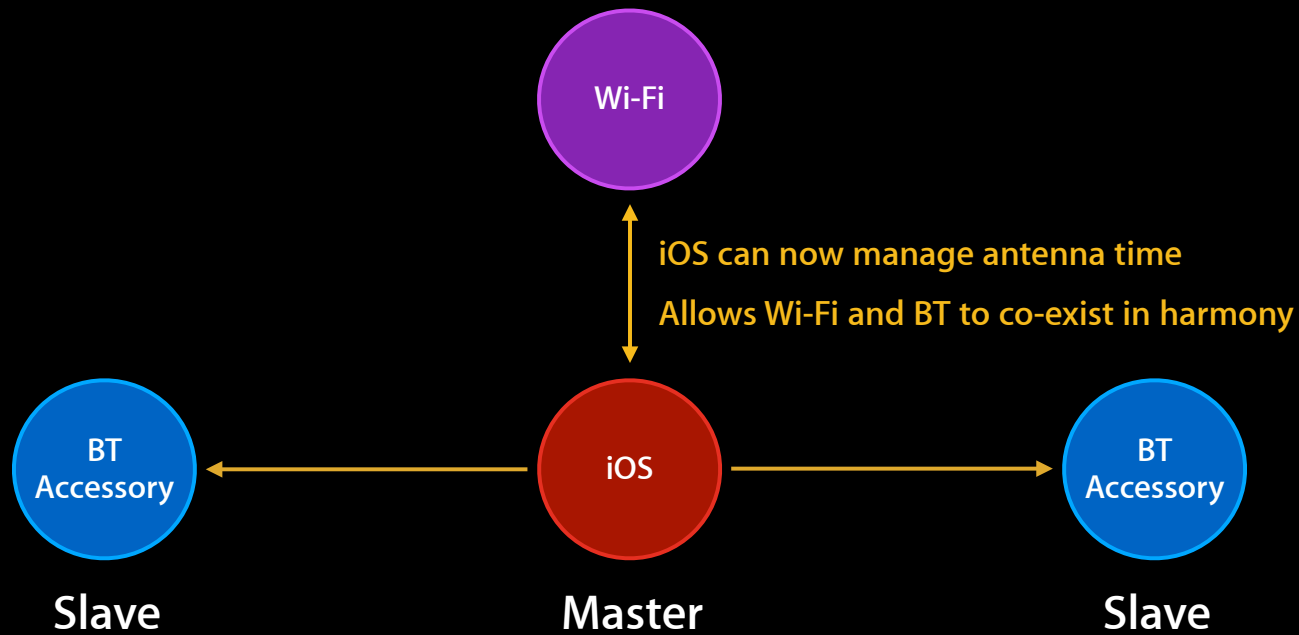
# “Help Me Help You”

## Role and Topology Management



# “Help Me Help You”

## Role and Topology Management

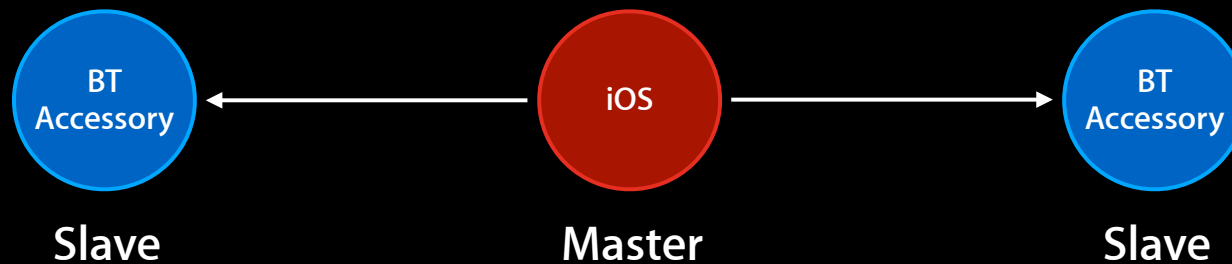




# “Help Me Help You”

## Role and Topology Management

- Allow iOS to role switch when requested
- It is OK to ask for role switch, but sometimes iOS will politely decline



# “Help Me Help You”

## Focus on Audio Quality

- Implement Wide Band Speech
- Use High Quality SBC parameters

Sample Frequency	44,100 Hz
Channel Mode	Stereo
Block Length	16
Subbands	8
Allocation method	Loudness
Bitpool Range	48 to 53 Shoot for 53

- Use MPEG 2/4 AAC codec

Object Type	MPEG-2 AAC LC
Sampling Frequency	44,100 Hz
Channels	2
Bit rate	131,072 bps
VBR	0

# Summary

- Bluetooth Design Guidelines

Download from iOS Dev Center

- Ask Questions

Bluetooth Developer Mailing List <[bluetooth-dev@lists.apple.com](mailto:bluetooth-dev@lists.apple.com)>

# Accessory Updates

# iPod Accessory Protocol (iAP)

- Allows accessories to communicate with and control iPod, iPhone, and iPad
- Details available through MFi Program  
<http://developer.apple.com/mfi/>



# External Accessory Framework

## App-Accessory Communication

# External Accessory Framework

## App-Accessory Communication



# External Accessory Framework

## App-Accessory Communication





# External Accessory Framework

## App-Accessory Communication



# External Accessory Framework



- Multitasking apps can now use External Accessory framework
  - Background audio
  - Voiceover IP
  - Background location
  - Task finishing
- Accessories can trigger a notification to launch an app

# Notifications

5

- New accessory command





# Media Metadata

Your app can now provide metadata to accessories

Album Title	Genre
Album Track Count	Persistent ID
Album Track Number	Playback Duration
Artist	Title
Artwork	Elapsed Playback Time
Composer	Playback Rate
Disc Count	Playback Queue Index
Disc Number	Playback Queue Count

# Accessibility

5



# AssistiveTouch

Accessible input for mobility

5



iCloud



# Summary

- External Accessory framework notifications
- Media metadata to accessories
- AssistiveTouch
- iCloud considerations



# More Information

## Stephen Chick

iPhone Evangelism  
[chick@apple.com](mailto:chick@apple.com)

## Craig Keithley

MFi Technology Evangelist  
[keithley@apple.com](mailto:keithley@apple.com)

## Developer Programs

MFi Program  
<http://developer.apple.com/mfi/>

## Bluetooth Mailing List

[bluetooth-dev@lists.apple.com](mailto:bluetooth-dev@lists.apple.com)

## Bluetooth Accessory Design Guidelines

<http://developer.apple.com/hardware/drivers/>

## Reporting Bugs

<http://bugreport.apple.com>

## Apple Developer Forums

<http://devforums.apple.com>

# Related Sessions

AirPlay and External Displays in iOS apps

Presidio  
Tuesday 3:15PM

iOS Accessibility

Marina  
Thursday 11:30AM

# Labs

Accessories Lab	Core OS Lab A Today 2:00PM
AirPlay Lab	Graphics, Media & Games Lab B Wednesday 9:00AM
USB, Bluetooth and FireWire Lab	Core OS Lab A Wednesday 2:00PM
Accessibility Lab	Application Frameworks Lab D Thursday 4:30PM

