Audio Session Management for iOS

Session 413

Eric Johnson
Core Audio Engineering

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

Outline

- Managed audio experience on iOS
- Audio session
  - Modes
  - Background audio and mixable vs. nonmixable
  - Routing properties and new behavior
  - Voice-processing audio unit
- Codecs

Audio Session Interaction

- Prepare session
- Handle begin interruption
- Handle end interruption

Prepare Session

```
...

// Retrieve session instance
AVAudioSession *session = [ AVAudioSession sharedInstance ];

// Request the “Ambient” category
[ session setCategory:AVAudioSessionCategoryAmbient error:&errRet ];

// Set our session to be active
[ session setActive:YES error:&errRet ];

// play audio ...
```

Handle Begin Interruption

```
- (void)beginInterruption
{
    // Playback has stopped.  Session is inactive
    // Update UI
}
```

- Global notification
  - AVAudioSessionDelegate
- Per instance notification
  - AVAudioPlayerDelegate
  - AVAudioRecorderDelegate

Handle End Interruption

```
- (void)endInterruptionWithFlags
{
    // Resume playback or recording
    // Update UI
    // Reactivate
}
```

- Global notification
  - AVAudioSessionDelegate
- Per-instance notification
  - AVAudioPlayerDelegate
  - AVAudioRecorderDelegate

Managed Audio Experience on iOS

Managed Audio Experience on iOS

- Users carry iOS devices everywhere
- Goal: Consistent user experience
- Mobile-device market is fast moving
- Choose the right APIs and communicate app's intentions

Audio Session Management

Focus on the audio user experience

- How to make your app's sounds
  - Behave according to user's expectations
  - Be consistent with built-in apps
- Categorize your application
- Mix with background audio
- Respond to interruptions
- Handle routing changes

Two Audio Session APIs

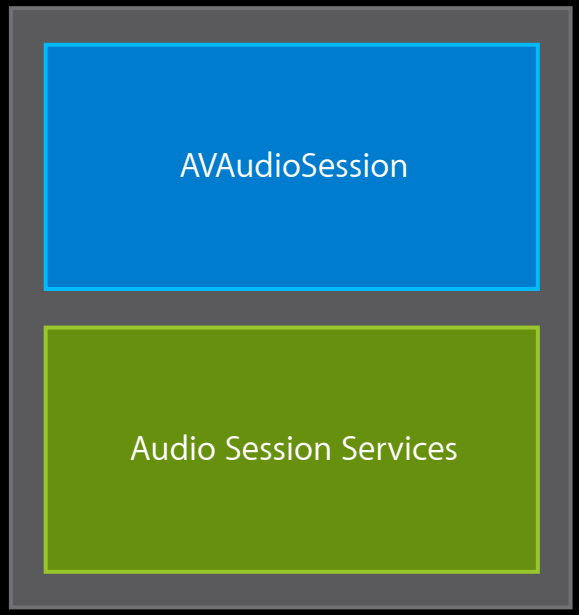
- AVAudioSession class

  `<AVFoundation/AVAudioSession.h>`

  - High-level wrapper for the most common functionality
- Audio Session Services

  `<AudioToolbox/AudioServices.h>`

  - All of the implementation
  - C Based, lower-level
- Mix and match OK



Using Audio Session

Five tasks

1. Set up the session and delegate
2. Choose and set a category
   - Choose and set mode
3. Make session active
4. Handle interruptions
5. Handle route changes

1. Set Up the Session

- Retrieve the AVAudioSession instance

```
AVAudioSession *session = [ AVAudioSession sharedInstance ];
```

- Set a delegate for notifications

```
session.delegate = myDelegate;
```

2. Choose and Set a Category

Based on role of audio in your app

Playback

Play and Record

Ambient

Record

Audio Processing

Solo Ambient

Categories

Audio applications

Category Name	Intended Usage	Mix with Others	Audio Input	Audio Output
 Playback	Audio Players, Video Players	Optional		●
 Record	Audio Recorders, Voice Capture		●	
 Play and Record	VoIP, Voice Chat	Optional	●	●

- Do not obey screen lock or ringer switch
- Allowed in background

Categories

Games, general applications

Category Name	Intended Usage	Mix with Others	Audio Input	Audio Output
 Ambient	Games, Productivity Apps	●		●
 Solo Ambient	Games, Productivity Apps			●

- Obey screen lock and ringer switch
- Not allowed in background

Category

Offline processing

Category Name	Intended Usage	Mix with Others	Audio Input	Audio Output
 Audio Processing	Offline Conversion, Offline Processing			

- Does not obey screen lock and ringer switch
- Allowed in background

2. Choose and Set a Category

```
// Retrieve session instance
AVAudioSession *session = [ AVAudioSession sharedInstance ];

// set a delegate for interruptions & state changes
session.delegate = myDelegate;

// Request the “Play and Record” category
[ session setCategory:AVAudioSessionCategoryPlayAndRecord error:&errRet ];

// Set our session to be active
[ session setActive:YES error:&errRet ];

// Set up AVAudioPlayer or OpenAL or ..
...

// Handle interruptions
...
```

2. Choose and Set a Mode

Voice Chat

Measurement

Video Recording

[Default]

Voice Chat Mode

- Works with play-and-record category
- Best microphone choice for current route
- Signal processing optimized for voice
- Manages routing
  - Restricts allowed routes
  - Bluetooth headsets available in routes
- Encourage use of Voice Processing Audio Unit

Phone-to-Ear Orientation on iPhone 4

Speaker-Phone Orientation on iPhone 4

Top Microphone

Bottom Speaker

Video-Recording Mode

- Supported categories
  - Play and record
  - Record
- Best microphone choice for current usage
- Best output route for current usage

Video-Recording Mode

Measurement Mode

- Supported categories
  - Record
  - Play and record
  - Playback
- Uses primary microphone
- Signal path
  - Flat EQ
  - Minimal signal processing

Default Mode

- Works with all categories
- Selects primary microphone
- Device configured for general usage

2. Choose and Set a Mode

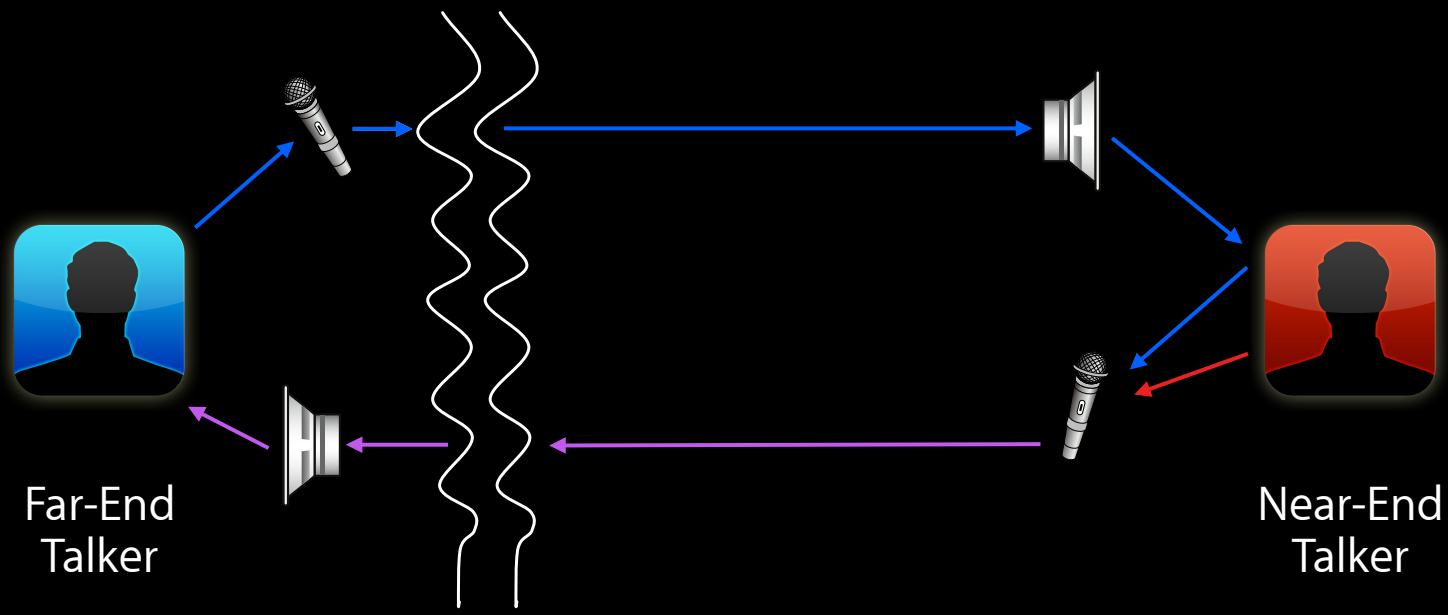
```
// Retrieve session instance
AVAudioSession *session = [ AVAudioSession sharedInstance ];

// set a delegate for interruptions & state changes
session.delegate = myDelegate;

// Request the “Play and Record” category
[ session setCategory:AVAudioSessionCategoryPlayAndRecord error:&errRet ];

// Request the “Voice Chat” mode
[ session setMode:AVAudioSessionModeVoiceChat error:&errRet ];

// Set our session to be active
[ session setActive:YES error:&errRet ];

// Set up AVAudioPlayer or OpenAL or ..
...

// Handle interruptions
...
```

Voice Processing Audio Unit

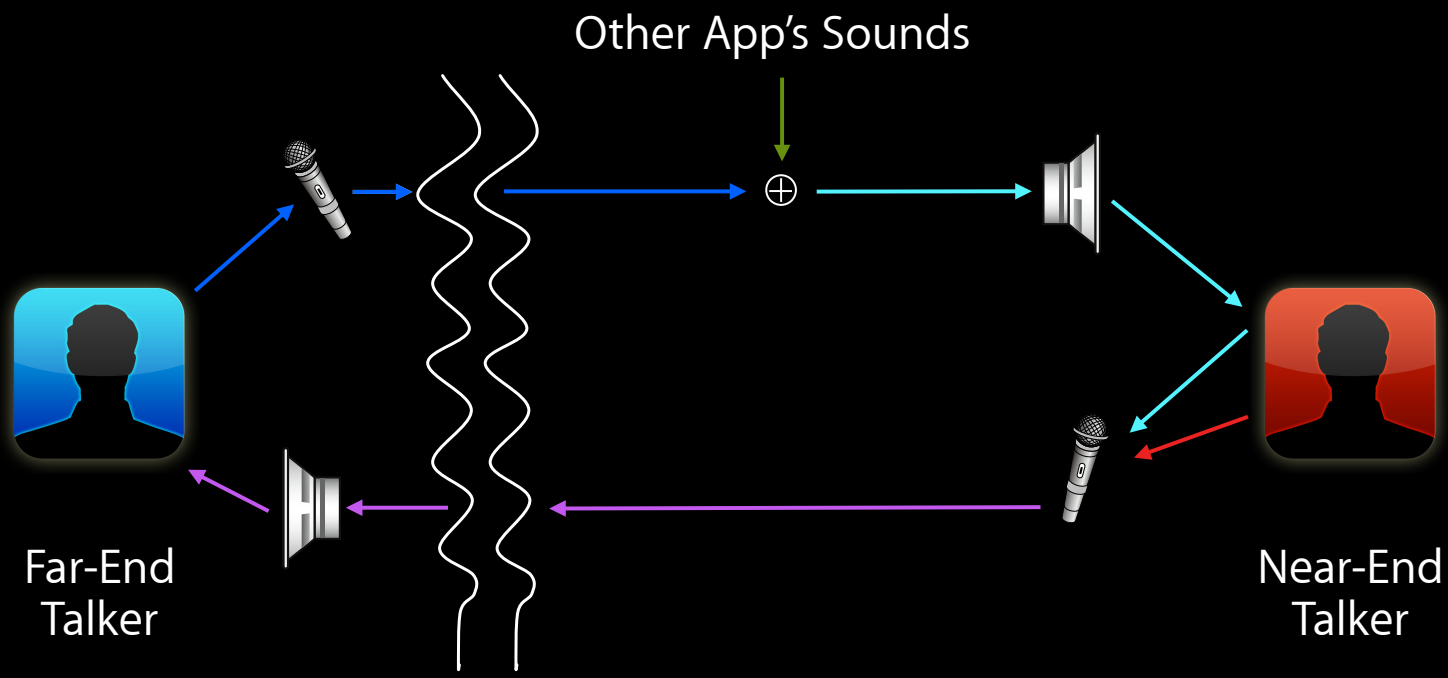
Voice Processing Audio Unit

- AURemoteIO with Acoustic Echo Canceler
- Designed for high-quality chat
- Two configurations available
  - Highest quality
  - Lowest complexity
- Available on iOS
- Available on OS X Lion New

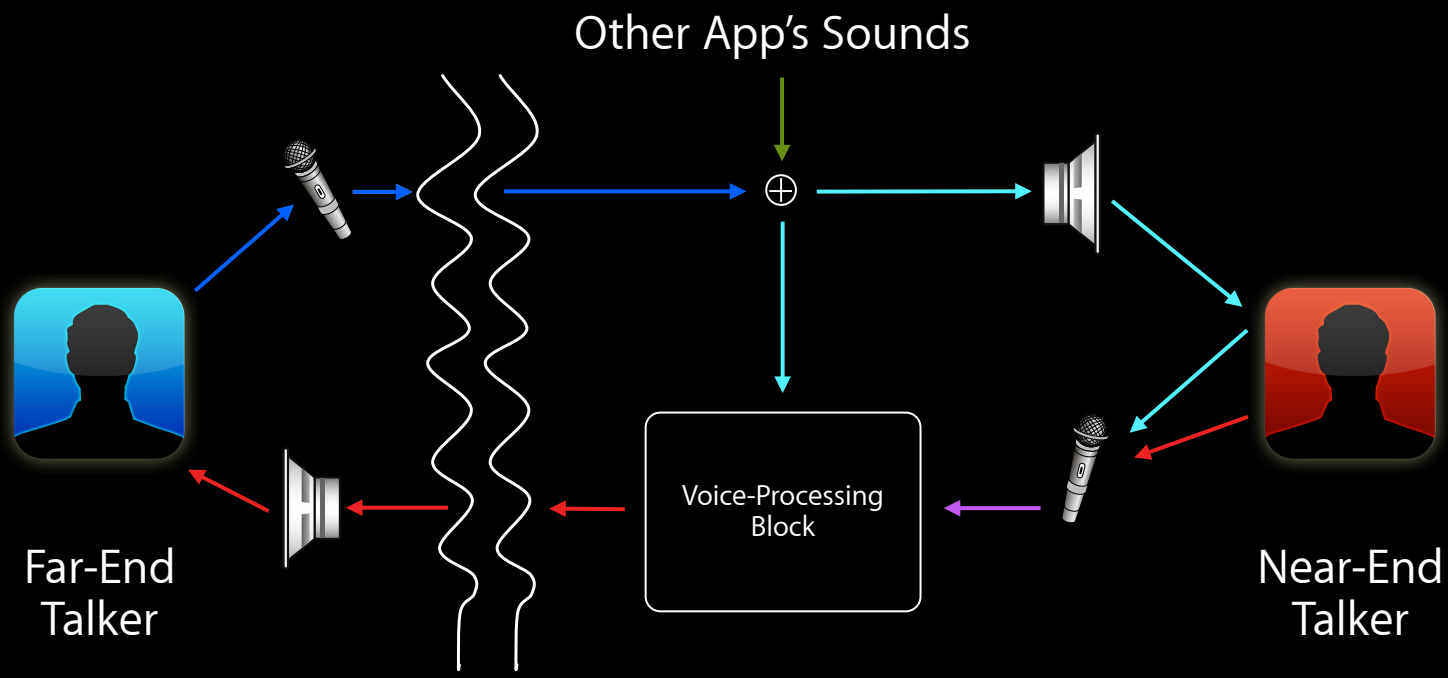
Without an Acoustic Echo Canceler



Without an Acoustic Echo Canceler



With an Acoustic Echo Canceler



Voice Processing Audio Unit Properties

- Defined in the `<AudioUnit/AudioUnitProperties.h>` header file

  `kAUVoiceIOProperty_BypassVoiceProcessing`

  `kAUVoiceIOProperty_VoiceProcessingEnableAGC`

  `kAUVoiceIOProperty_DuckNonVoiceAudio`

  `kAUVoiceIOProperty_VoiceProcessingQuality`

  `kAUVoiceIOProperty_MuteOutput`

3. Make Session Active

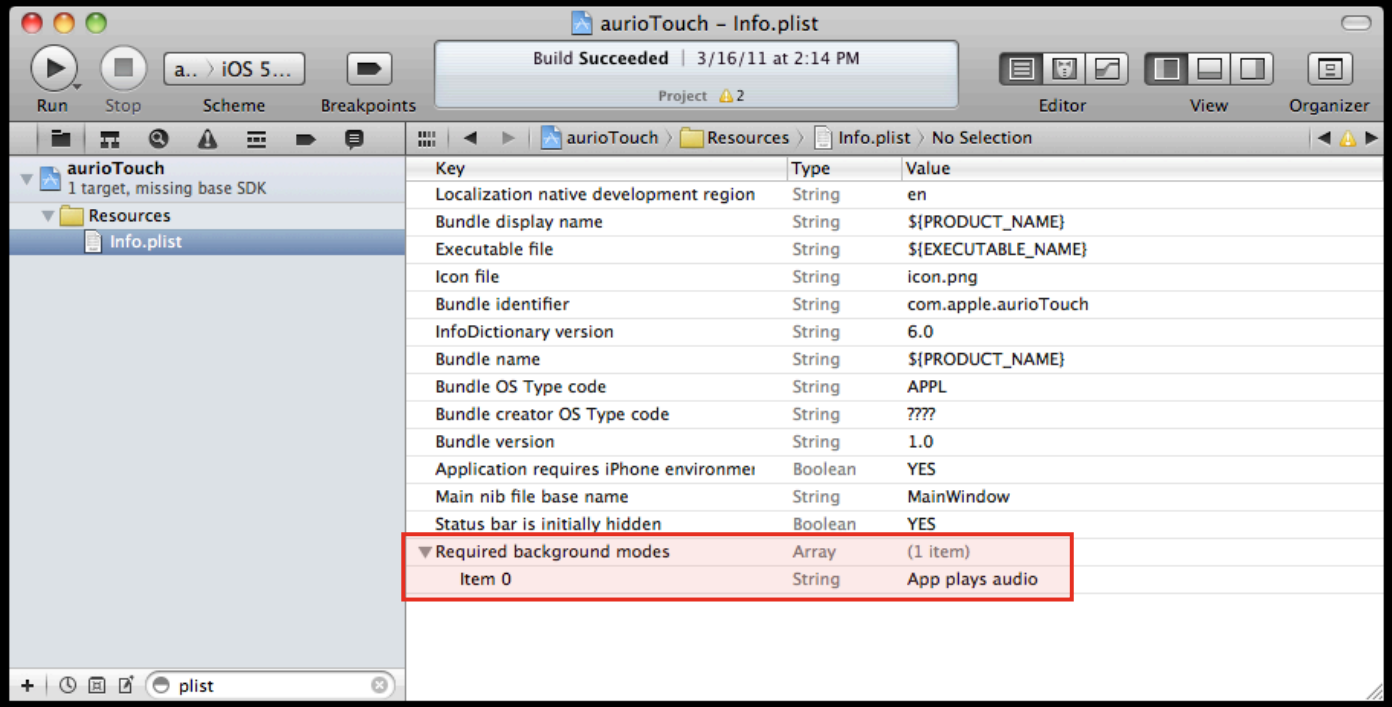
```
// Retrieve session instance
AVAudioSession *session = [ AVAudioSession sharedInstance ];

// set a delegate for interruptions & state changes
session.delegate = myDelegate;

// Request the “Play and Record” category
[ session setCategory:AVAudioSessionCategoryPlayAndRecord error:&errRet ];

// Request the “Voice Chat” mode
[ session setMode:AVAudioSessionModeVoiceChat error:&errRet ];

// Set our session to be active
[ session setActive:YES error:&errRet ];

// Set up AVAudioPlayer or OpenAL or ..
...

// Handle interruptions
...
```

3. Make Session Active

```
// Retrieve session instance
AVAudioSession *session = [ AVAudioSession sharedInstance ];

// set a delegate for interruptions & state changes
session.delegate = myDelegate;

// Request the “Play and Record” category
[ session setCategory:AVAudioSessionCategoryPlayAndRecord error:&errRet ];

// Request the “Voice Chat” mode
[ session setMode:AVAudioSessionModeVoiceChat error:&errRet ];

// Set our session to be active
[ session setActive:YES error:&errRet ];

// Set up AVAudioPlayer or OpenAL or ..
...

// Handle interruptions
...
```

Background Audio

Enabling Background Audio

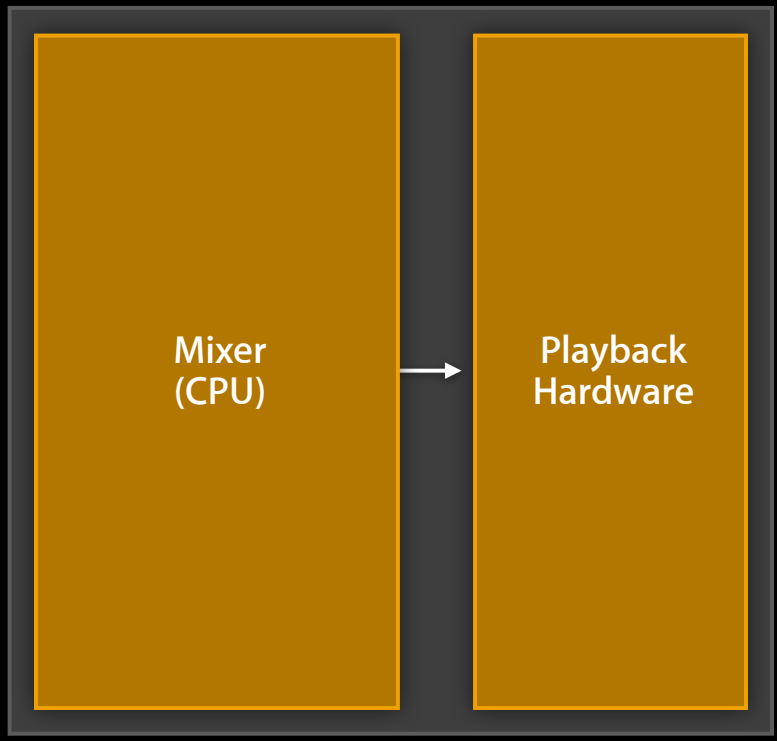


Mixing with Background Audio

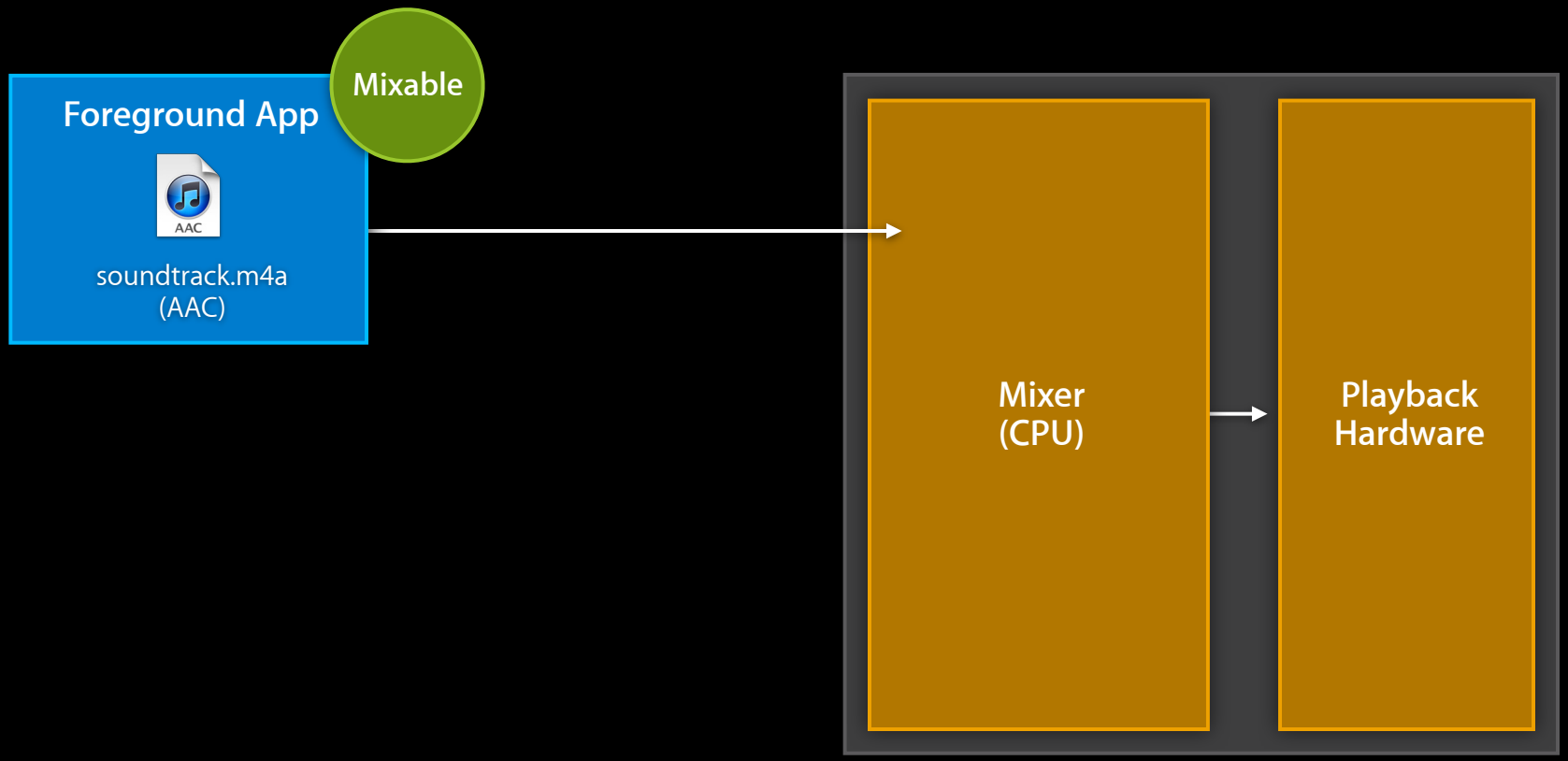
- More than one app wants to play audio
- What is heard?
  - Depends on both apps
    - "Mixable" or "nonmixable"
- "Nonmixable"
  - HW Resources
  - Mode assertion
  - May interrupt other audio

Category Name	Mix with Others
Playback	Optional
Record	
Play and Record	Optional
Audio Processing	
Ambient	●
Solo Ambient	

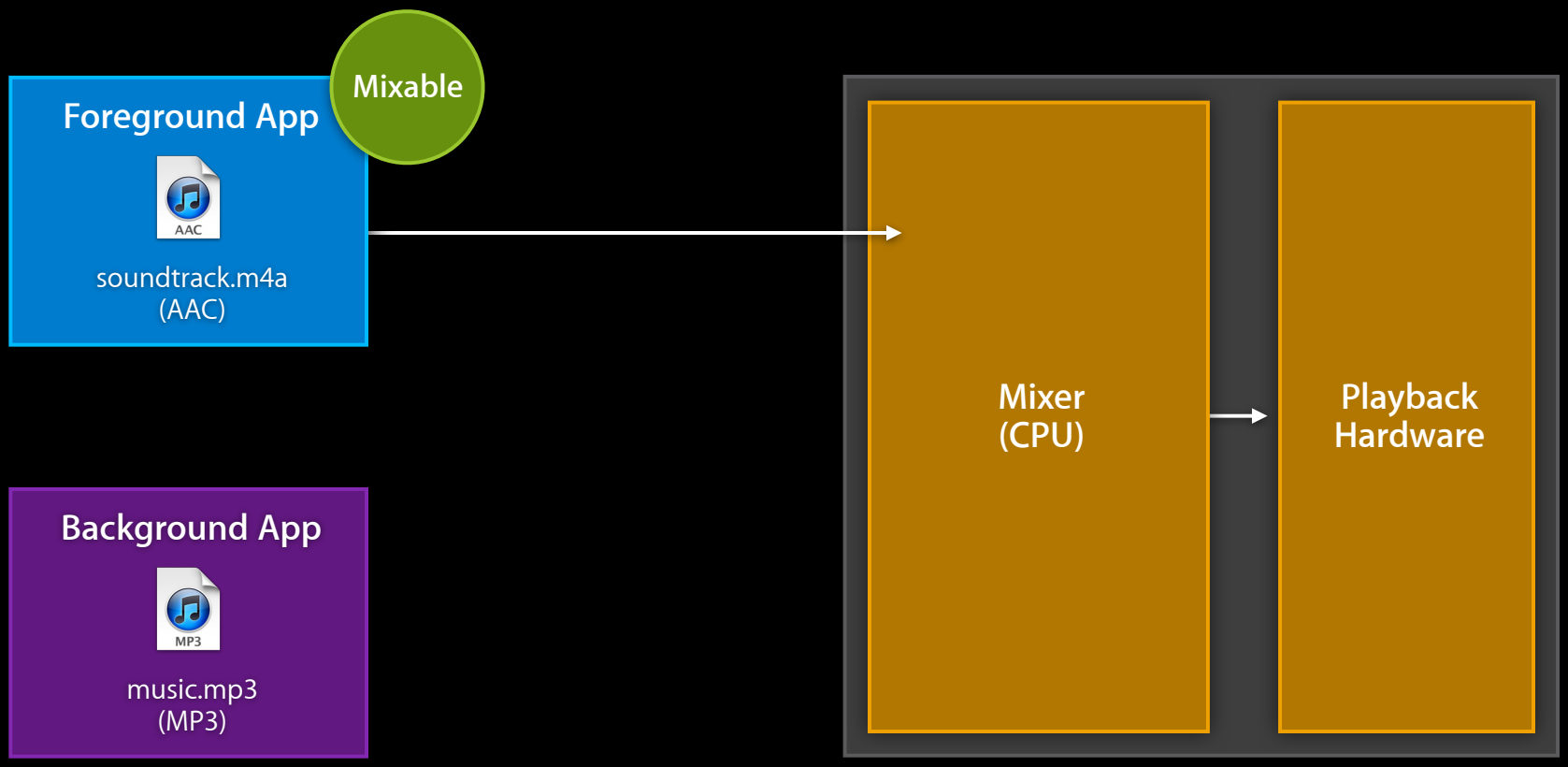
Mixing with Background Audio



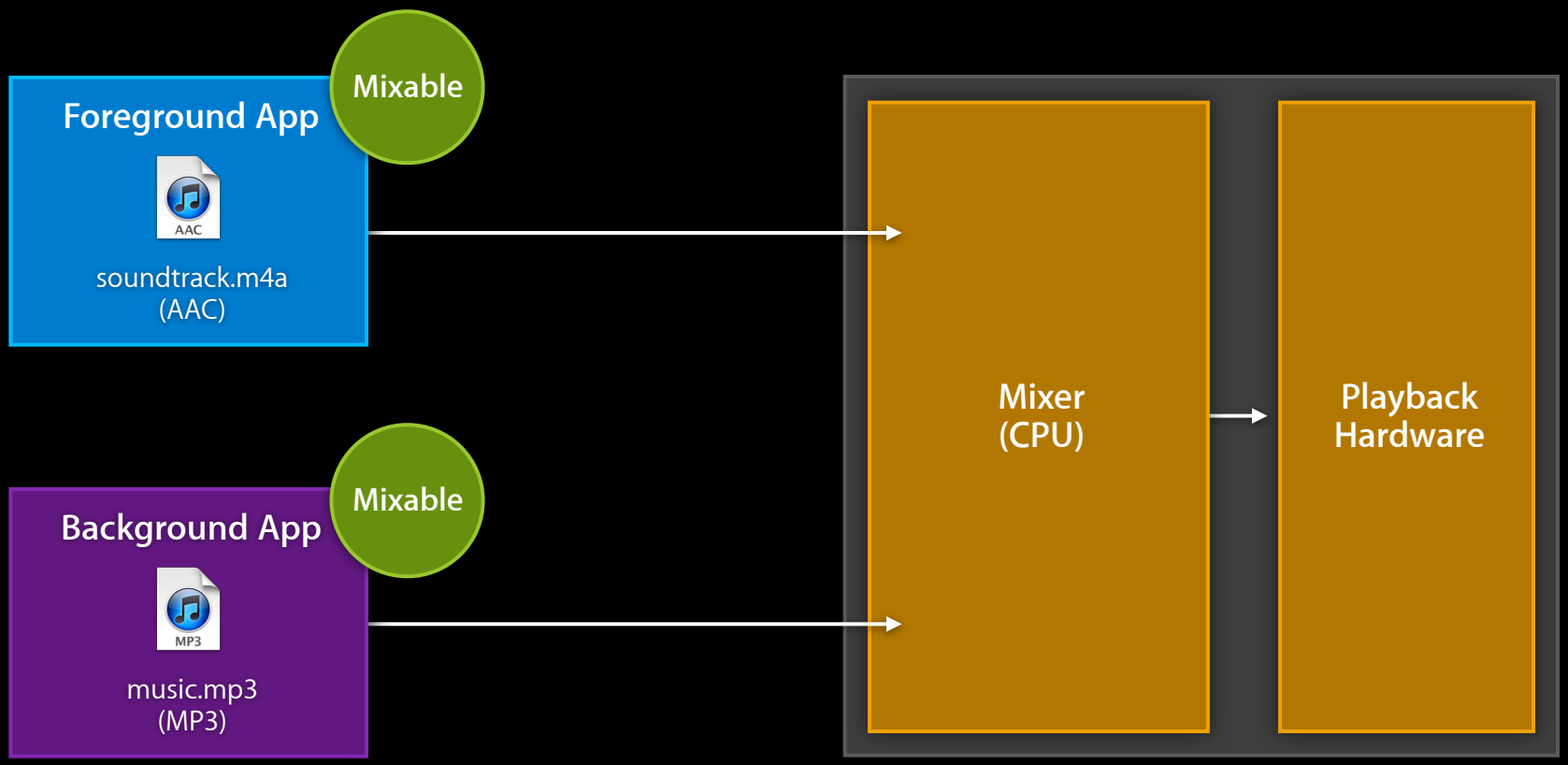
Mixing with Background Audio



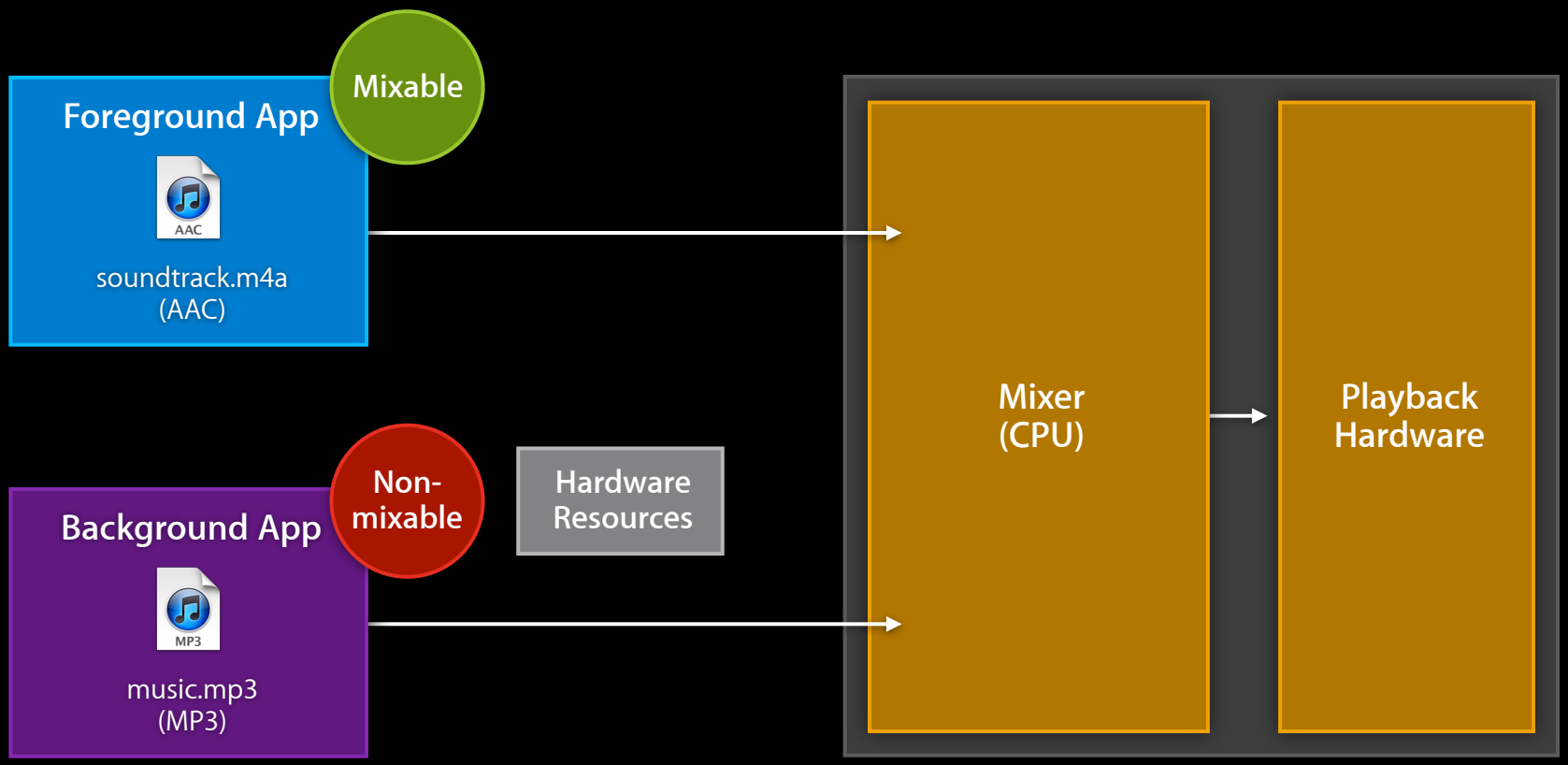
Mixing with Background Audio



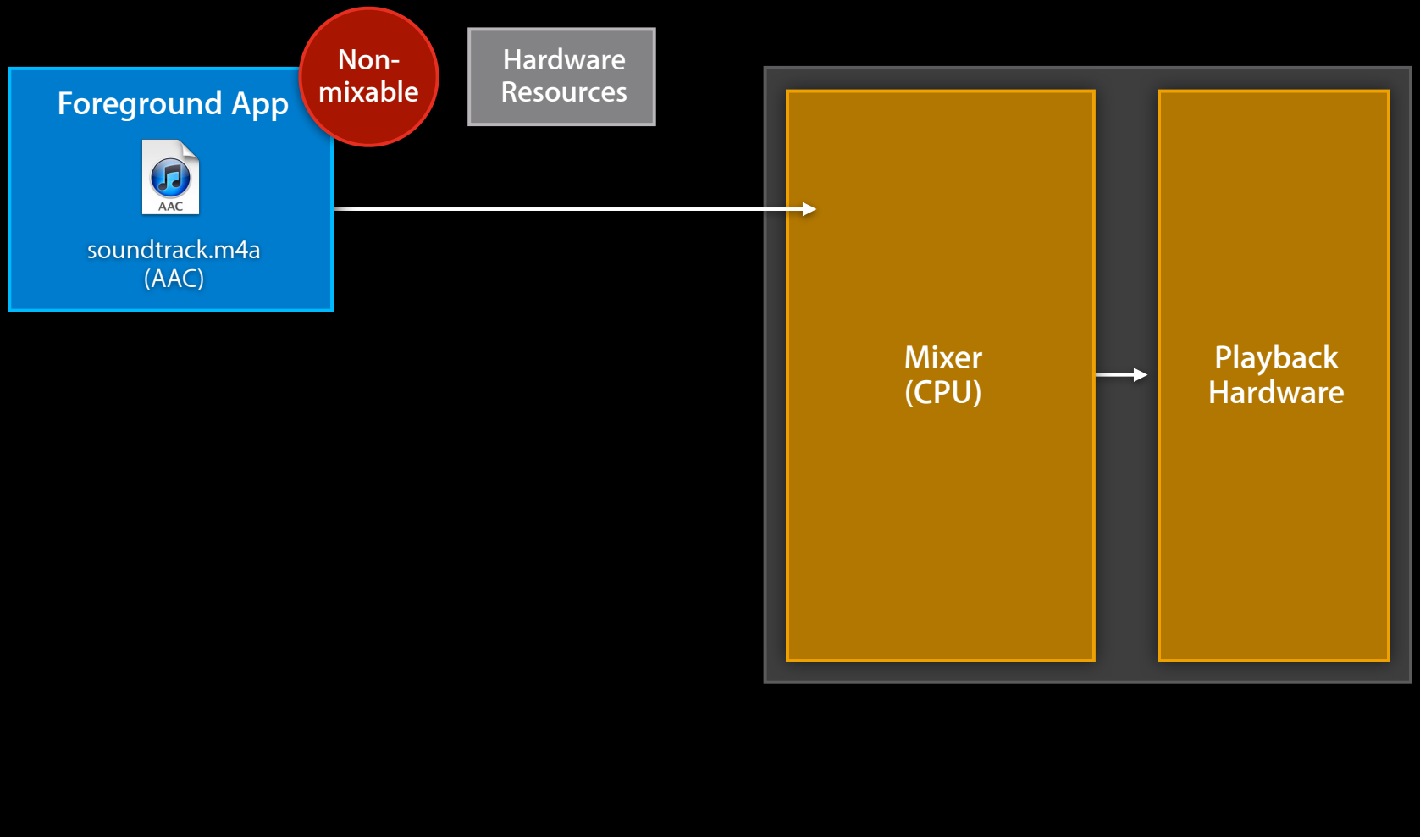
Mixing with Background Audio



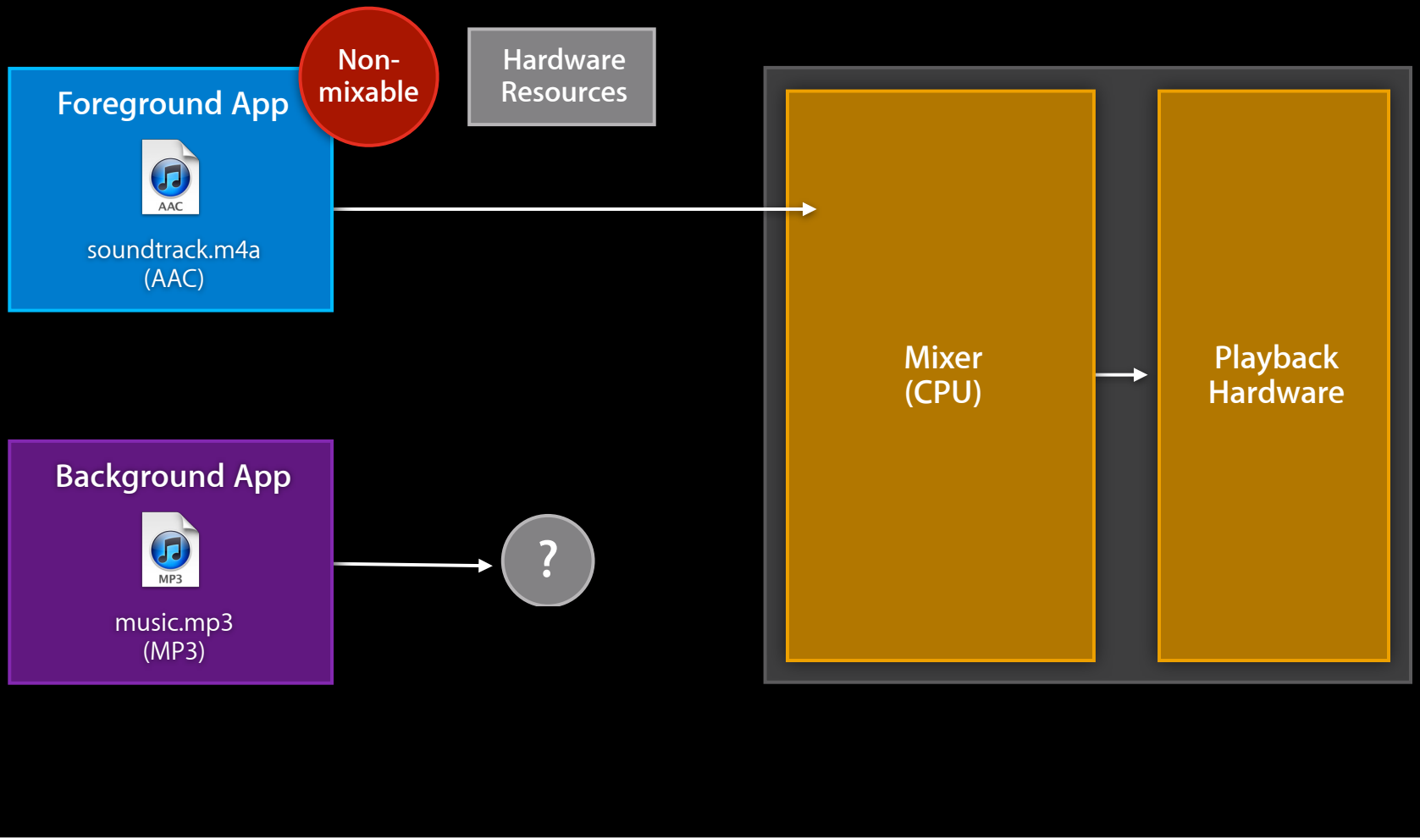
Mixing with Background Audio



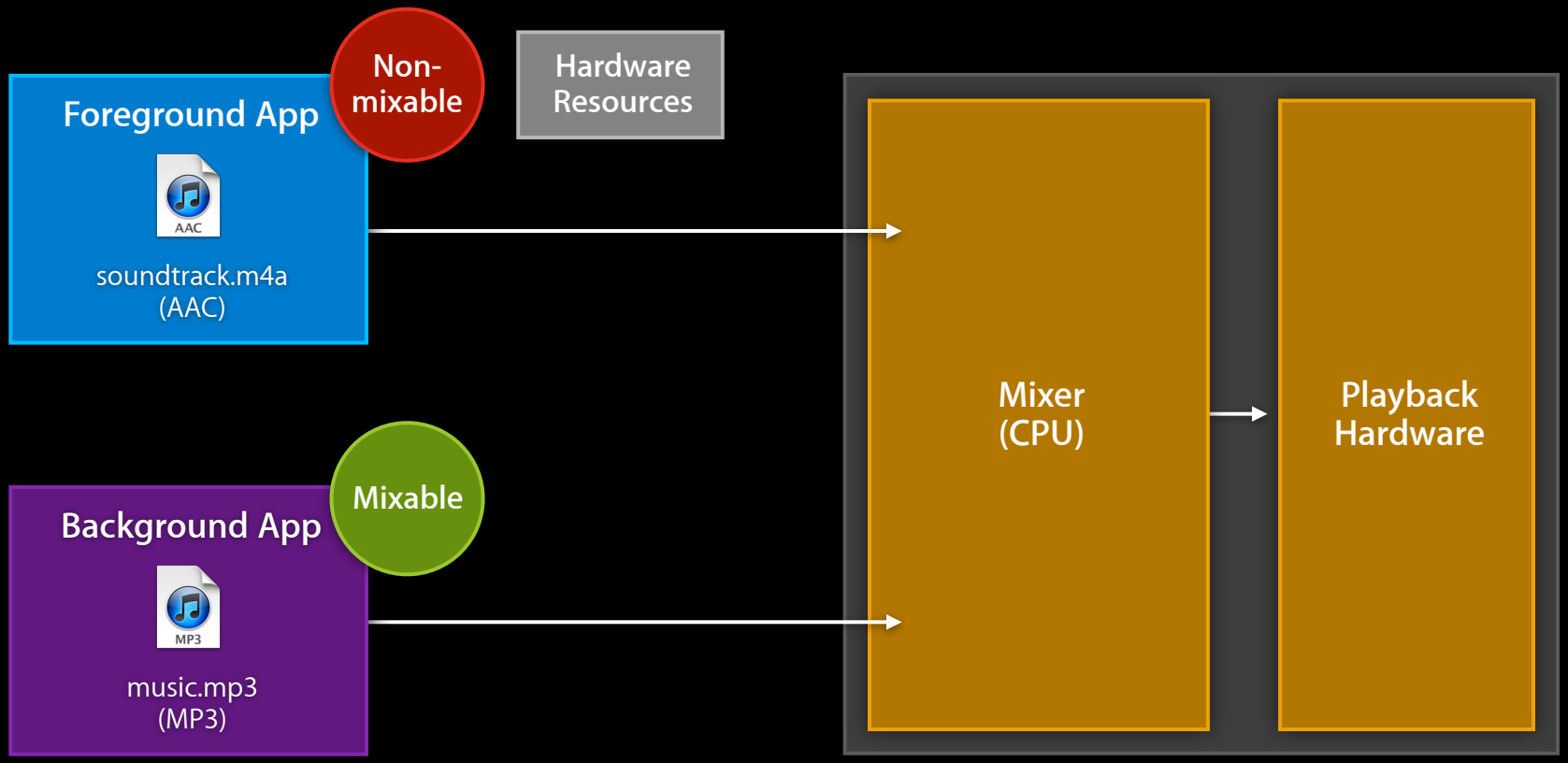
Mixing with Background Audio



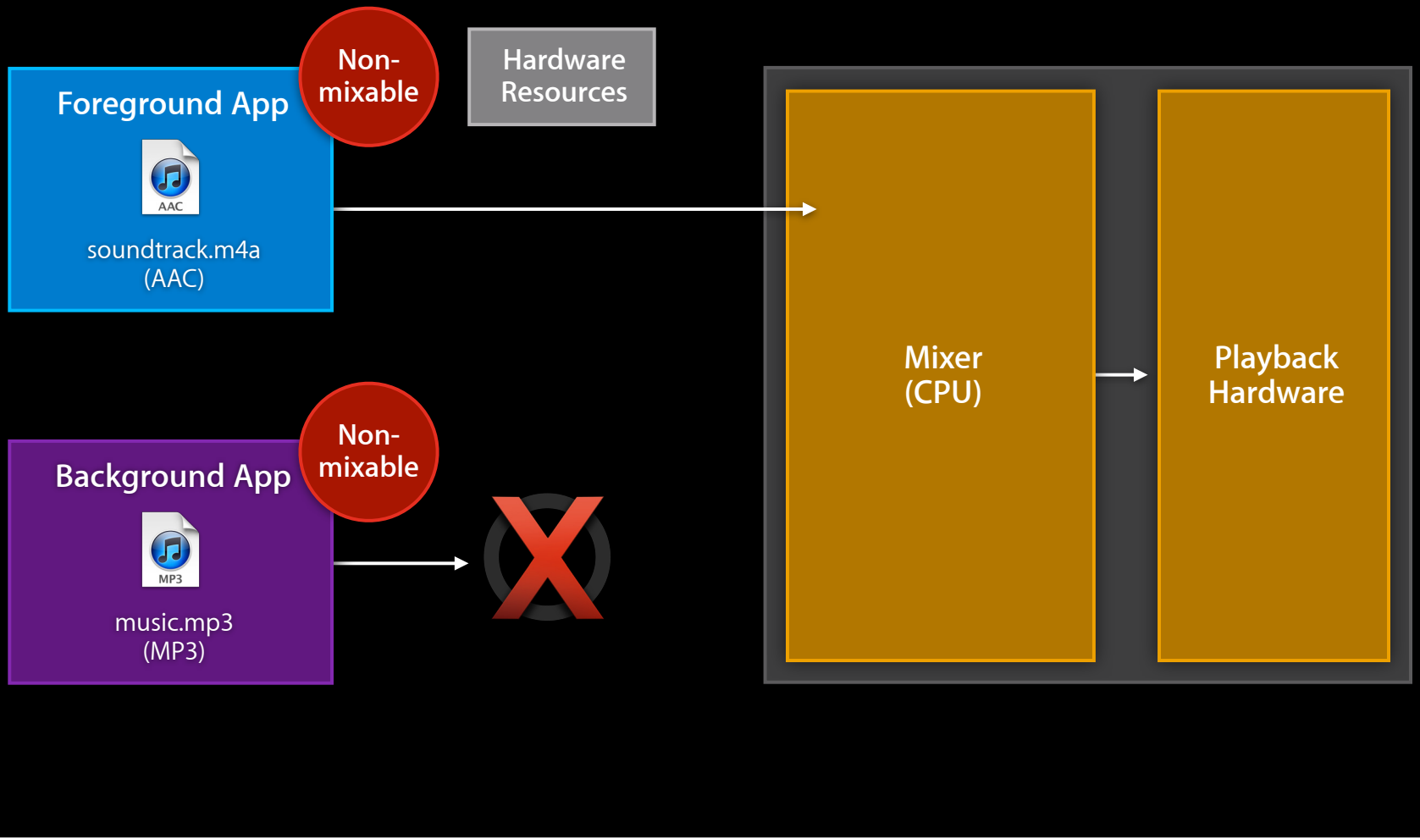
Mixing with Background Audio



Mixing with Background Audio



Mixing with Background Audio



Quick Tip

Interaction with background audio

- Most apps can just go "Active" and stay that way
- Some apps should only be "Active" while in use
  - Recorders
  - VoIP apps
  - Turn-by-turn navigation apps
  - Nonmixable apps

Quick Tip

VoIP/voice chat app

- Go active when the call begins
- Interrupts background app playback
- Go inactive when the call ends
- System will notify background app it can resume

```
- (void)myCallDidFinish
{
    int flags = AVAudioSessionSetActiveFlags_NotifyOthersOnDeactivation;

    // Call is done
    [ session setActive:NO withFlags:flags error:&errRet ];

    // update UI, ...
}
```

Turn-by-Turn GPS Activation

```
- (void)setup
{
    UInt32 mix = 1, duck = 1;

    [ session setCategory:AVAudioSessionCategoryPlayback
                    error:&errRet ];

    AudioSessionSetProperty (
        kAudioSessionProperty_OverrideCategoryMixWithOthers,
        sizeof(mix), &mix );

    AudioSessionSetProperty (
        kAudioSessionProperty_OtherMixableAudioShouldDuck,
        sizeof(duck), &duck );
}
```

Turn-by-Turn GPS Activation

```
- (void)playThePreloadedInstructions
{
    [ session setActive:YES error:&errRet ];   // duck other audio
    [ myAudioPlayer play ];
}


- (void)audioPlayerDidFinishPlaying:(AVAudioPlayer *)player
                       successfully:(BOOL)flag
{
    [ session setActive:NO error:&errRet ];   // un-duck
}
```

4. Handle Interruptions

- Session may be interrupted by higher-priority audio
  - Phone call, clock alarm, foreground app
- Interruption makes your session inactive
  - Currently playing audio is stopped
- After the interruption is over
  - Reactivate certain state (API-specific)
  - Then become active again (if appropriate)

Responding to Interruptions

AVAudioSessionDelegate methods

```
- (void)beginInterruption
```

- Audio has stopped, already inactive
- Change state of UI, etc., to reflect nonplaying state
  - For instance, change Play button to its stopped state

```
- (void)endInterruptionWithFlags:
  AVAudioSessionInterruptionFlags_ShouldResume
```

- Make session active
- Update user interface
- Resume playback or recording

Responding to Interruptions

AVAudioPlayerDelegate methods

```
- (void) audioPlayerBeginInterruption
```

- Playback has stopped, already inactive
- Change state of UI, etc., to reflect nonplaying state
  - For instance, change Play button to its stopped state

```
- (void) audioPlayerEndInterruption:withFlags:
  AVAudioSessionInterruptionFlags_ShouldResume
```

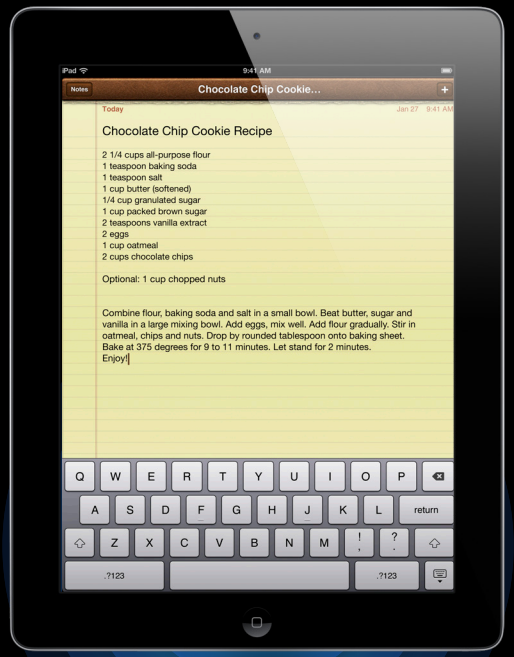
- Update user interface
- Resume playback

5. Handle Route Changes

What users expect

- "Last in wins"
  - Plugging in
    - Routed to headset/headphone
    - Continues playing without pause
  - Unplugging
    - Routed to previous output
    - Audio playback should pause

New Routing Behavior



Audio Routing

Topics

- Querying the route
- Listening for changes

Querying the Route

What is the current route?

`kAudioSessionProperty_AudioRouteDescription`

- CFDictionaryRef with detailed information about the route

`kAudioSession_AudioRouteKey_Inputs`

- Value is an array of input descriptions

`kAudioSession_AudioRouteKey_Outputs`

- Value is an array of output descriptions

Detailed Route Description—Inputs

```
extern const CFStringRef   kAudioSessionInputRoute_LineIn
extern const CFStringRef   kAudioSessionInputRoute_BuiltInMic
extern const CFStringRef   kAudioSessionInputRoute_HeadsetMic
extern const CFStringRef   kAudioSessionInputRoute_BluetoothHFP
extern const CFStringRef   kAudioSessionInputRoute_USBAudio
```

Detailed Route Description—Outputs

```
extern const CFStringRef kAudioSessionOutputRoute_LineOut
extern const CFStringRef kAudioSessionOutputRoute_Headphones
extern const CFStringRef kAudioSessionOutputRoute_BluetoothHFP
extern const CFStringRef kAudioSessionOutputRoute_BluetoothA2DP
extern const CFStringRef kAudioSessionOutputRoute_BuiltInReceiver
extern const CFStringRef kAudioSessionOutputRoute_BuiltInSpeaker
extern const CFStringRef kAudioSessionOutputRoute_USBAudio
extern const CFStringRef kAudioSessionOutputRoute_HDMI
extern const CFStringRef kAudioSessionOutputRoute_AirPlay
```

Listening for Changes

Where did the route go?

`AudioSessionAddPropertyListener()`

- Register for notifications for when a route changes

`kAudioSessionProperty_AudioRouteChange`

- Notification for when the route changes
  - Reason why route changed
  - What was the old route?

```
AudioSessionAddPropertyListener (
    kAudioSessionProperty_AudioRouteChange,
    MyPropListener, &clientData );
```

Audio Route Change Notification

```
void MyPropListener (void* clientData,
                     AudioSessionPropertyID inID,
                     UInt32 dataSize,
                     const void* inData)
{
  CFDictionaryRef dict = (CFDictionaryRef)inData;

  CFNumberRef reason = CFDictionaryGetValue(dict,
      kAudioSession_RouteChangeKey_Reason);

  CFDictionaryRef oldRoute = CFDictionaryGetValue(dict,
      kAudioSession_AudioRouteChangeKey_PreviousRouteDescription);

  CFDictionaryRef newRoute = CFDictionaryGetValue(dict,
      kAudioSession_AudioRouteChangeKey_CurrentRouteDescription);

}
```

Using Audio Session

Summary

1. Set up the session and delegate
2. Choose and set a category
   - Choose and set mode
3. Make session active
4. Handle interruptions
5. Handle route changes

Audio Codecs

Audio Codecs

- CODEC = enCOder + DECoder
- Compresses and decompresses linear PCM audio signals
- Lossy and lossless codecs
- Core technology in digital audio

Lossless Audio Codecs

- No loss of information
- Typical compression factor: 1.5–2.0

Lossy Audio Codecs

- Relies on a perceptual model of the human auditory system
- Quality varies with the bit rate
- Typical compression factor: 6–24

Availability of Popular Codecs

Category Name	Encoder	Decoder
MP3		●
ALAC	●	●
AAC Low Complexity	●	●
AAC High Efficiency		●
AAC High Efficiency v2		●
AAC (Enhanced) Low Delay	●	●
AAC Enhanced Low Delay + SBR	●	●

AAC Variants

- AAC Low Complexity
  - Core technology of AAC family
  - High quality
  - General use
- AAC High Efficiency
  - Streaming audio
  - Low bit rates
- AAC Enhanced Low Delay
  - VoIP, conference applications
  - Low delay

Using Codecs

- AVFoundation
  - AVAudioPlayer
  - AVAudioRecorder
- ExtendedAudioFile
- AudioQueue
- AudioConverter

Summary

- Managed audio experience on iOS
- Audio Session
  - Modes
  - Background audio and mixable vs. nonmixable
  - Routing properties and new behavior
  - Voice Processing Audio Unit
- Codecs

Labs

Audio Lab	Graphics, Media & Games Lab C Wednesday 2:00PM

More Information

Eryk Vershen
Media Technologies Evangelist
eversheN@apple.com

Audio Programming Guides
iPhone Dev Center
http://developer.apple.com/devcenter/ios

Apple Developer Forums
http://devforums.apple.com


WWDC2011