Audio Recipes for iOS

AVFoundation / MPMediaPlayer / CoreAudio / OpenAL

Javier Sánchez

@jsanchezsierra

http://ccrma.stanford.edu/~jsanchez

NSSpain, September 17-19, Logroño, Spain

Agenda

- SystemSound (Audio Toolbox)
- Media Player (MPMusicPlayerController)
- AVFoundation (AVAudioPlayer)
- Audio Sessions

demo 1

- CoreAudio / Audio Units

demo 2

- Open AL

demo 3

Media Player

OpenAL

AV Foundation

Audio Toolbox

Audio Unit

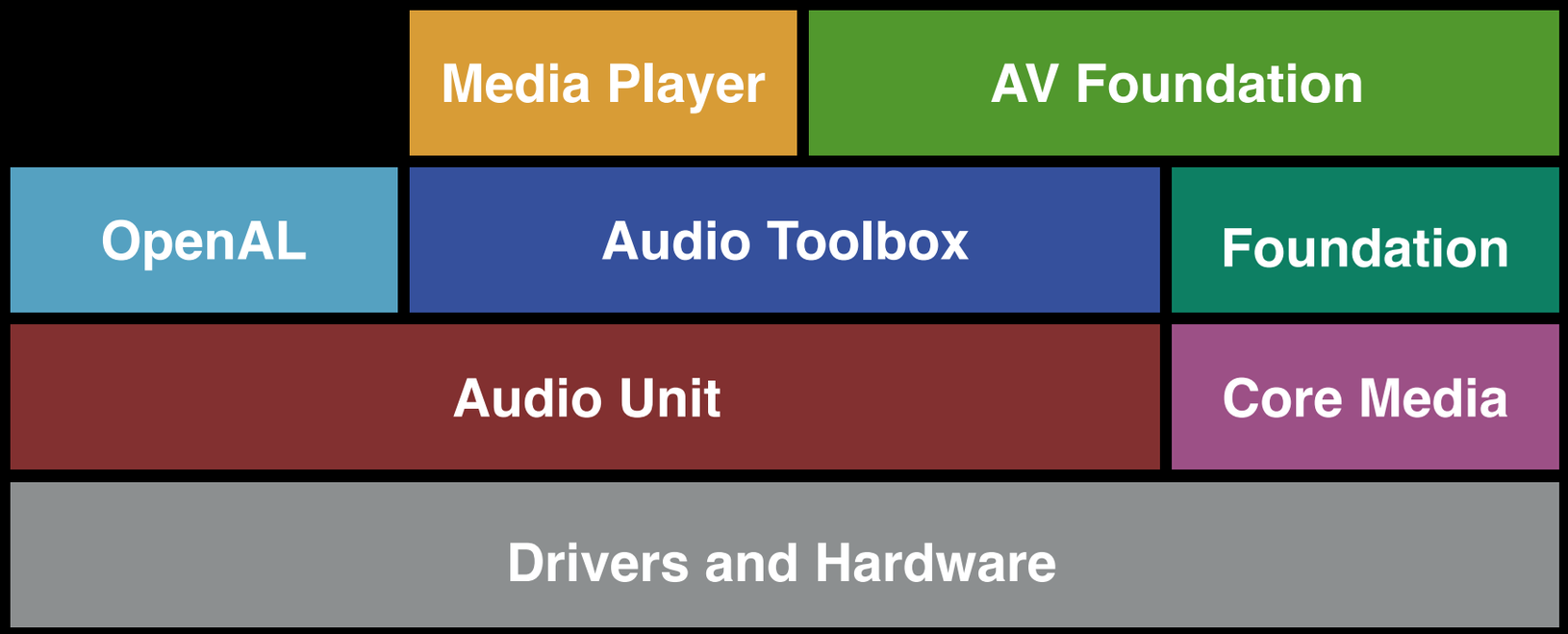
@jsanchezsierra
http://ccrma.stanford.edu/~jsanchez

Demo I

SystemSound / Media Player / AVAudioPlayer

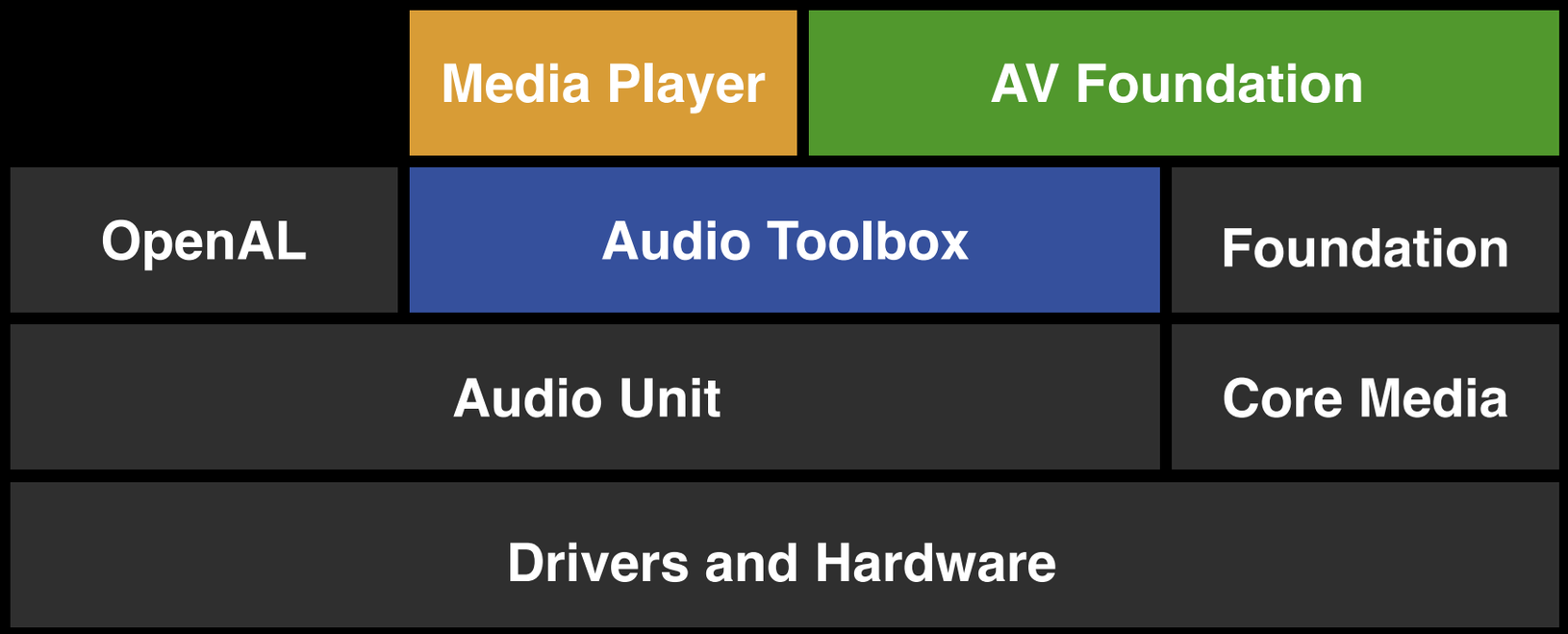
https://github.com/jsanchezsierra/AudioLab

Introduction



@jsanchezsierra
http://ccrma.stanford.edu/~jsanchez

Introduction



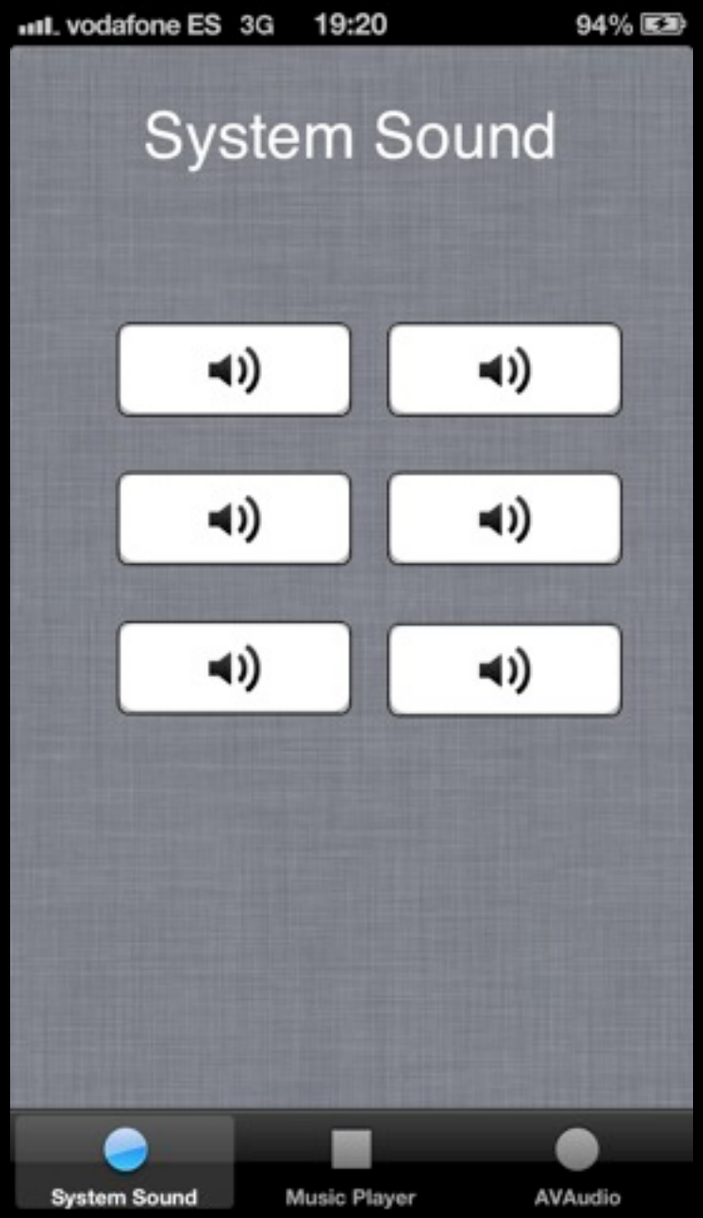
@jsanchezsierra
http://ccrma.stanford.edu/~jsanchez

SystemSound
Audio Toolbox

@jsanchezsierra
http://ccrma.stanford.edu/~jsanchez

SystemSound

Audio Toolbox



@jsanchezsierra
http://ccrma.stanford.edu/~jsanchez

SystemSound

Audio Toolbox

- Add AudioToolbox framework

```
#import <AudioToolBox/AudioToolbox.h>
```

- Create the URL to your sound file (ex: sound.mp3)

```
NSString *soundPath = [[NSBundle mainBundle] pathForResource:@”sound" ofType:@”mp3” inDirectory:@"/"];
CFURLRef SoundPathPathURL = (CFURLRef) [[NSURL alloc]  initFileURLWithPath: soundPath];
```

- Create the sound and play

```
SystemSoundID systemSound;
AudioServicesCreateSystemSoundID ( SoundPathPathURL  ,&systemSound);
AudioServicesPlaySystemSound(systemSound);
```

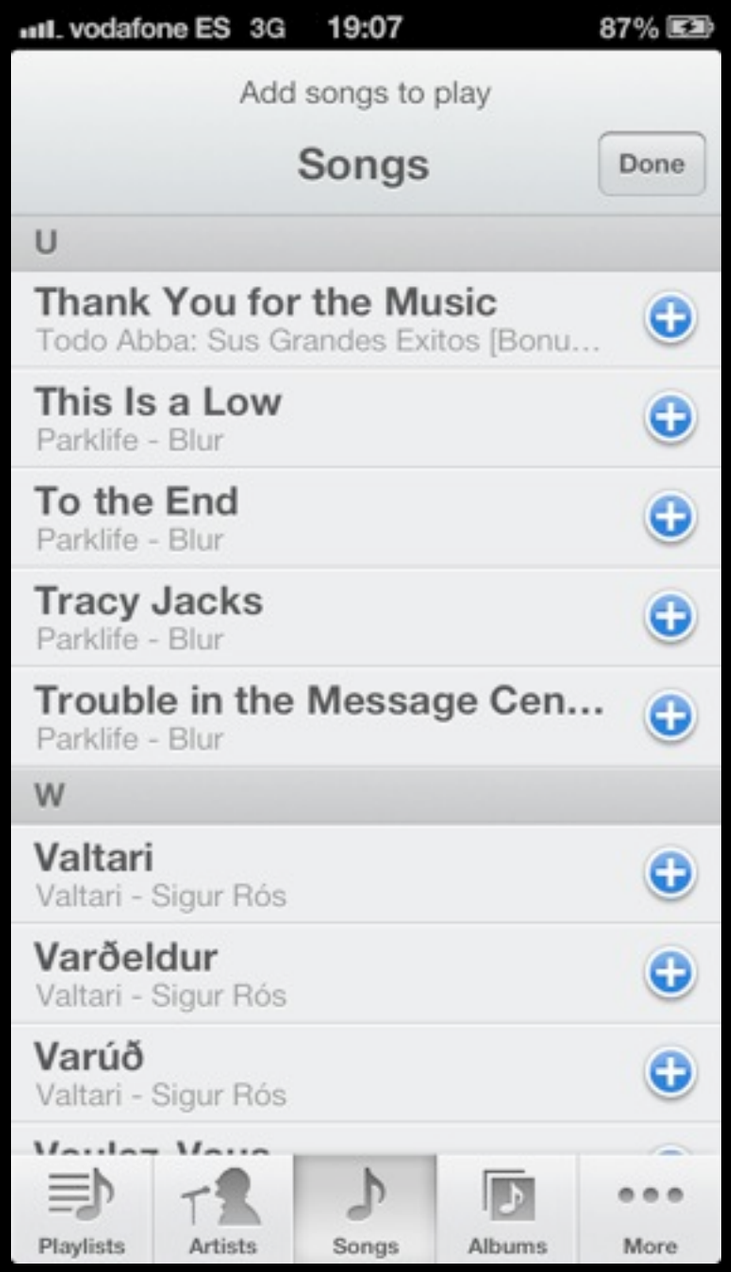
- No longer than 30 seconds in duration
- In linear PCM or IMA4 (IMA/ADPCM) format
- Packaged in a `.caf`, `.aif`, or `.wav` file

@jsanchezsierra
http://ccrma.stanford.edu/~jsanchez

Media Player

MPMediaPickerController / MPMusicPlayerController

Media Player

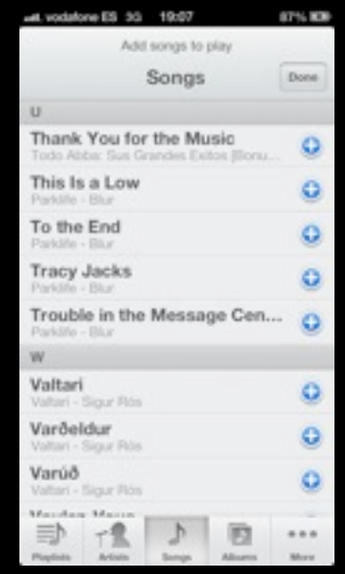


MPMediaPickerController

MPMusicPlayerController

MPMediaPicker Controller

Media Player



- Add MediaPlayer framework

```
#import <MediaPlayer/MediaPlayer.h>
```

- MPMediaPickerController allows you to access your music Library

```
MPMediaPickerController *picker =[[MPMediaPickerController alloc] initWithMediaTypes: MPMediaTypeMusic];
picker.delegate = self;   // MPMediaPickerControllerDelegate
picker.allowsPickingMultipleItems = YES;
picker.prompt  = @"Añade las canciones a reproducir";
```

- Present the MPMediaPickerController in a modal ViewController

```
UINavigationController *theNavController = [[UINavigationController alloc]
                  initWithRootViewController:picker];
[self presentModalViewController:theNavController animated: YES];
```

- Delegate Methods --> MPMediaPickerControllerDelegate.

```
- (void) mediaPicker: (MPMediaPickerController *) mediaPicker
                        didPickMediaItems: (MPMediaItemCollection *) mediaItemCollection
{
   ...
   [myPlayer setQueueWithItemCollection: mediaItemCollection];
   ...
}
```

@jsanchezsierra
http://ccrma.stanford.edu/~jsanchez

MPMusicPlayerController

Media Player

- Add MediaPlayer framework

```
#import <MediaPlayer/MediaPlayer.h>
```

- Create an instance of MPMusicPlayerController

```
MPMusicPlayerController *myPlayer =[MPMusicPlayerController applicationMusicPlayer ];
[myPlayer setQueueWithItemCollection: mediaItemCollection];
[myPlayer play];
```

- Get metadata (título, artista, album, imagen...)

```
MPMediaItem *myTrack = [ myPlayer nowPlayingItem];
NSString *title=[myTrack valueForProperty: MPMediaItemPropertyTitle]];
NSString *artist=[myTrack valueForProperty: MPMediaItemPropertyArtist]];
UIImage *artworkImage= [[myTrack valueForProperty: MPMediaItemPropertyArtwork]
    imageWithSize: CGSizeMake (120, 120) ]
```

- Methods and properties

```
[myPlayer play];
[myPlayer pause];
[myPlayer stop];
[myPlayer skipToNextItem];
[myPlayer skipToPreviousItem];
[myPlayer skipToBeginning];
```

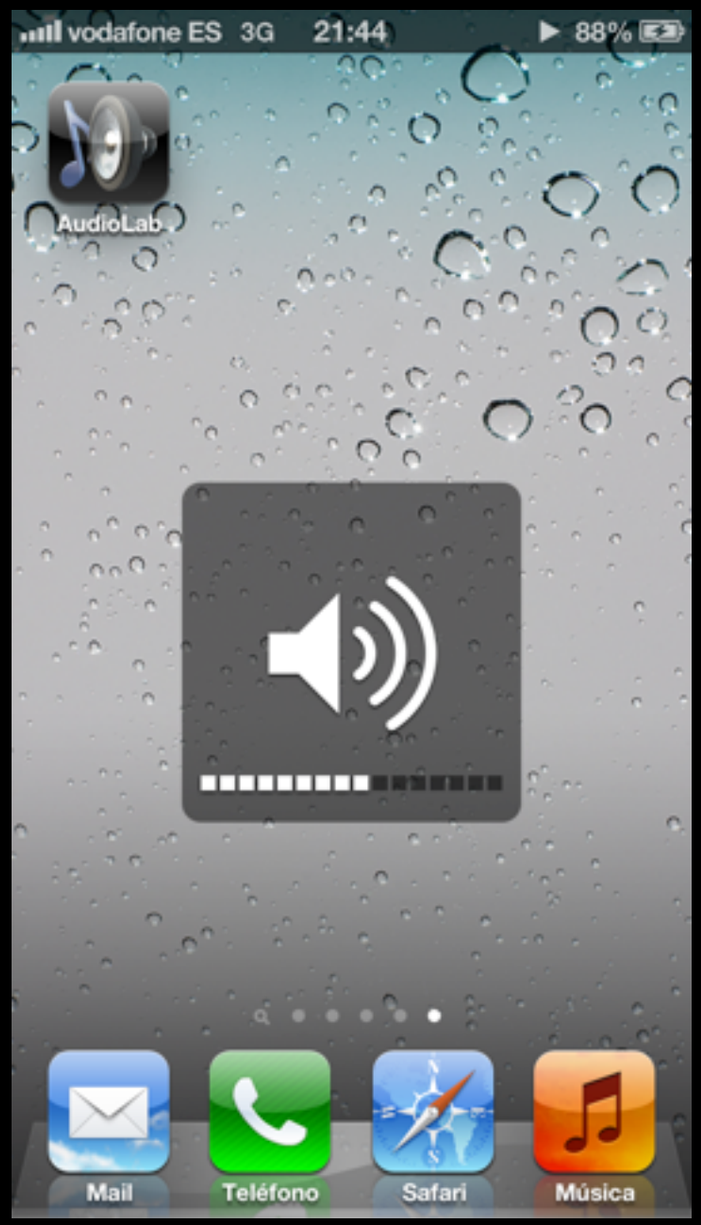
```
[myPlayer nowPlayingItem];
[myPlayer currentPlaybackTime];
[myPlayer repeatMode];
[myPlayer shuffleMode];
[myPlayer playbackState];
[myPlayer volumen];
```

@jsanchezsierra
http://ccrma.stanford.edu/~jsanchez

MPVolumeView

@jsanchezsierra
http://ccrma.stanford.edu/~jsanchez

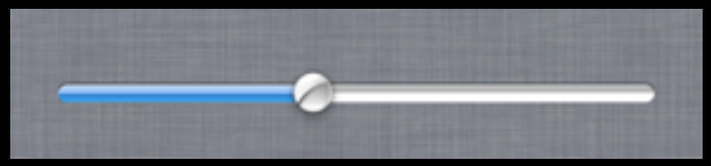
MPVolumeView



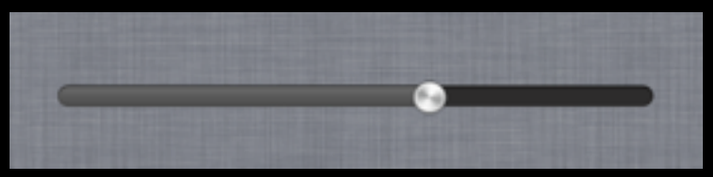
@jsanchezsierra
http://ccrma.stanford.edu/~jsanchez

MPVolumeView

```
MPVolumeView *myVolumeView = [[MPVolumeView alloc] initWithFrame: CGRectMake(20, 450, 280, 20)];
[self.view addSubview: myVolumeView];
```



```
UIImage* knobImage = ...
UIImage* volumeViewMinImage = ...
UIImage* volumeViewMaxImage = ...

[myVolumeView setVolumeThumbImage: KnobImage forState:UIControlStateNormal];
[myVolumeView setMinimumVolumeSliderImage: volumeViewMinImage forState:UIControlStateNormal];
[myVolumeView setMaximumVolumeSliderImage: volumeViewMaxImage forState:UIControlStateNormal];
```



knobImage

volumeViewMinImage

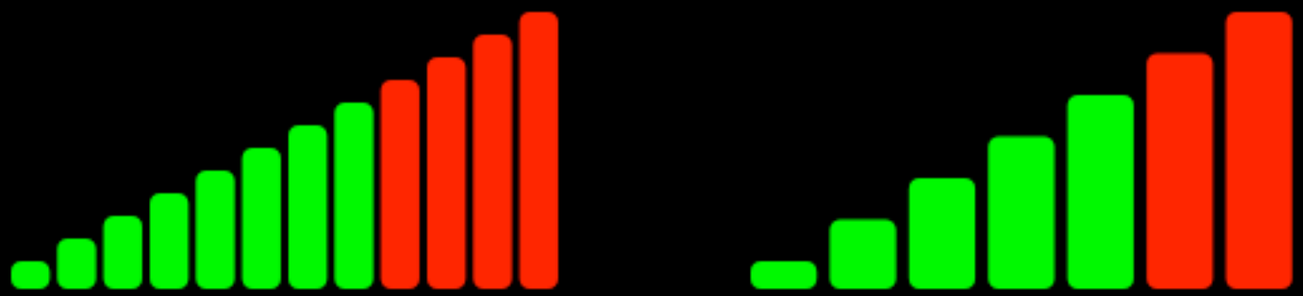
volumeViewMaxImage

@jsanchezsierra
http://ccrma.stanford.edu/~jsanchez

VolumeView

My Custom Volume View Class (colors, number of lines)

https://github.com/jsanchezsierra/VolumeView



```
VolumeView *volumeView= [[VolumeView alloc] initWithFrame:CGRectMake(175, 425, 110, 55 ) ];
[volumeView setNumberOfBars:7];
[volumeView setBarsColorMin:[UIColor greenColor]];
[volumeView setBarsColorMax:[UIColor redColor]];
[self.view addSubview: volumeView];
```

@jsanchezsierra
http://ccrma.stanford.edu/~jsanchez

AVFoundation

AVAudioPlayer

@jsanchezsierra
http://ccrma.stanford.edu/~jsanchez

AVAudioPlayer

@jsanchezsierra
http://ccrma.stanford.edu/~jsanchez

AVFoundation

AVAudioPlayer

- Add AVFoundation framework

```
#import <AVFoundation/AVFoundation.h>
```

- Create a AVAudioPlayer from a local file

```
NSURL * urlTrack = [[NSURL alloc] initFileURLWithPath:
                        [[NSBundle mainBundle] pathForResource:@"track" ofType:@"mp3"]];

AVAudioPlayer *track = [[AVAudioPlayer alloc] initWithContentsOfURL: urlTrack  error: nil];
```

- Create a AVAudioPlayer from NSData

```
AVAudioPlayer *track = [[AVAudioPlayer alloc] initWithData: [NSData ...]  error: nil];
```

- Methods and properties

```
[track duration]; (read only)
[track numberOfChannels]; (read only)
[track isPlaying]; (read only)
[track prepareToPlay];
[track play];
[track pause];
[track stop];
[track playAtTime];
[track volumen];
[track pan];
```

```
[track numberOfLoops];
[track rate];
[track enableRate];
[track currentTime];
[track meteringEnabled];
[track averagePowerForChannel];
[track peakPowerForChannel];
[track url];
[track data];
[track settings];
```

@jsanchezsierra
http://ccrma.stanford.edu/~jsanchez

AVFoundation

AVAudioPlayer Delegates

```
-(void) audioPlayerDidFinishPlaying:(AVAudioPlayer *)player successfully:(BOOL)flag
{
    //Update UI
    /...
}

-(void) audioPlayerBeginInterruption:(AVAudioPlayer *)player
{
    [player pause];

    //Update UI
    /...

}

-(void) audioPlayerEndInterruption:(AVAudioPlayer *)player withOptions:(NSUInteger)flags
{

    [player play];

    //Update UI
    /...


}
```

@jsanchezsierra
http://ccrma.stanford.edu/~jsanchez

Audio Session

AVAudioSession

@jsanchezsierra
http://ccrma.stanford.edu/~jsanchez

Audio Session

AVAudioSession

- Categorize the App audio type
- Mix with background audio
- Handle interruptions and routing changes

- Objective-C alternative to many features from the C-based Audio Session services

AVAudioSession	**Audio Session Services**
<AVFoundation/AVAudioSession.h>	<AudioToolbox/AudioSession.h>
High-level wraper with most common functionalities	C-Based, lower-level, all the implementation

@jsanchezsierra
http://ccrma.stanford.edu/~jsanchez

Audio Session

instance/delegate/category/activate

```objc
#import <AVFoundation/AVFoundation.h>

//Create an instance of AVAudioSession
AVAudioSession *session = [AVAudioSession sharedInstance];

//Registers the delegate of the audio session.
[session setDelegate: self];

//set the playback category for the sesision
[session setCategory: AVAudioSessionCategoryPlayback error: nil];

// Activates the audio session.
[session setActive: YES error: nil];
```

@jsanchezsierra
http://ccrma.stanford.edu/~jsanchez

Audio Session

2.-Choose a category

Based on the role of audio in your app

Category Name	Intended Usage	Obey Ringer Switch	Obey Screen Lock	Mix with Others	Audio Input	Audio Output	Allowed In Background
Playback	Audio Players, Video Players			Optional		●	●
Record	Audio Recorders, Voice Capture				●		●
Play and Record	VOIP, Voice Chat			Optional	●	●	●
Audio Processing	Offline Conversion, Offline Processing						●
Ambient	Games, Productivity Apps	●	●	●		●	
Solo Ambient	Games, Productivity Apps	●	●			●	

WWDC 2010 - Session 412 - Audio Development for iOS, Part1

@jsanchezsierra
http://ccrma.stanford.edu/~jsanchez

Audio Session

Handle interruptions

AVAudioSessionDelegate

```
// Called after your audio session is interrupted
-(void) beginInterruption
{
    // Update UI ...
    // Playback stopped

}

// Called after your audio session interruption ends
// flags indicate the state of the audio session
-(void) endInterruptionWithFlags:(NSUInteger)flags
{
    // Update UI ...
    // Make session active
    // Resume playback

}
```

@jsanchezsierra
http://ccrma.stanford.edu/~jsanchez

Audio Session

Handle route changes

What is the user experience?

Pluggin in the headphone

- routed to headphone
- audio continues playing, no pause

Unpluggin the headphone

- routed to output
- audio pause

@jsanchezsierra
http://ccrma.stanford.edu/~jsanchez

Audio Session

Queryng route

Audio Session Services

Querying the route --> kAudioSessionProperty_AudioRoute

```
CFStringRef currentRoute;
UInt32 size = sizeof(currentRoute);
AudioSessionGetProperty( kAudioSessionProperty_AudioRoute, &size, &currentRoute);
NSLog (@”Current route is %@”, currentRoute);
```

route values -> @“speaker”, @“Headphone”, @“receiver”, @“”...

Overriting the output audio route --> kAudioSessionProperty_OverrideAudioRoute

```
UInt32 override = kAudioSessionOverrideAudioRoute_Speaker;
AudioSessionSetProperty (kAudioSessionProperty_OverrideAudioRoute,
                         sizeof(override), & override );
```

routes output to speaker

@jsanchezsierra
http://ccrma.stanford.edu/~jsanchez

Audio Session

Handle route changes

Audio Session Services

listening to route changes --> kAudioSessionAddPropertyListener

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

    Register for notifications when route changes (reason/old route).
```

AudioRoute change CallBack --> kAudioSessionProperty_OverrideAudioRoute

```
    void MyPropListener (void* clientData, AudioSessionPropertyID inID,
                         UInt32 dataSize, const void* inData)
    {
        CFDictionaryRef dict = (CFDictionaryRef)inData;
        CFNumberRef reason = CFDictionaryGetValue(dict,
                                CFSTR(kAudioSession_AudioRouteChangeKey_Reason));

        CFStringRef oldRoute = CFDictionaryGetValue(dict,
                                CFSTR(kAudioSession_AudioRouteChangeKey_OldRoute));

    }
```

@jsanchezsierra
http://ccrma.stanford.edu/~jsanchez

Playing in Background

info.plist file --> set Required background modes

Key	Type	Value
▼ Information Property List	Dictionary	(16 items)
Localization native development reg	String	en
Bundle display name	String	${PRODUCT_NAME}
Executable file	String	${EXECUTABLE_NAME}
▶ Icon files	Array	(0 items)
Bundle identifier	String	Javier-Sanchez.${PRODUCT_NAME:rfc1034identifier}
InfoDictionary version	String	6.0
Bundle name	String	${PRODUCT_NAME}
Bundle OS Type code	String	APPL
Bundle versions string, short	String	1.0
Bundle creator OS Type code	String	????
Bundle version	String	1.0
Application requires iPhone environ	Boolean	YES
▶ Required device capabilities	Array	(1 item)
▼ Supported interface orientations	Array	(1 item)
Item 0	String	Portrait (bottom home button)
Icon file	String	icon.png
▼ Required background modes	Array	(1 item)
Item 0	String	App plays audio

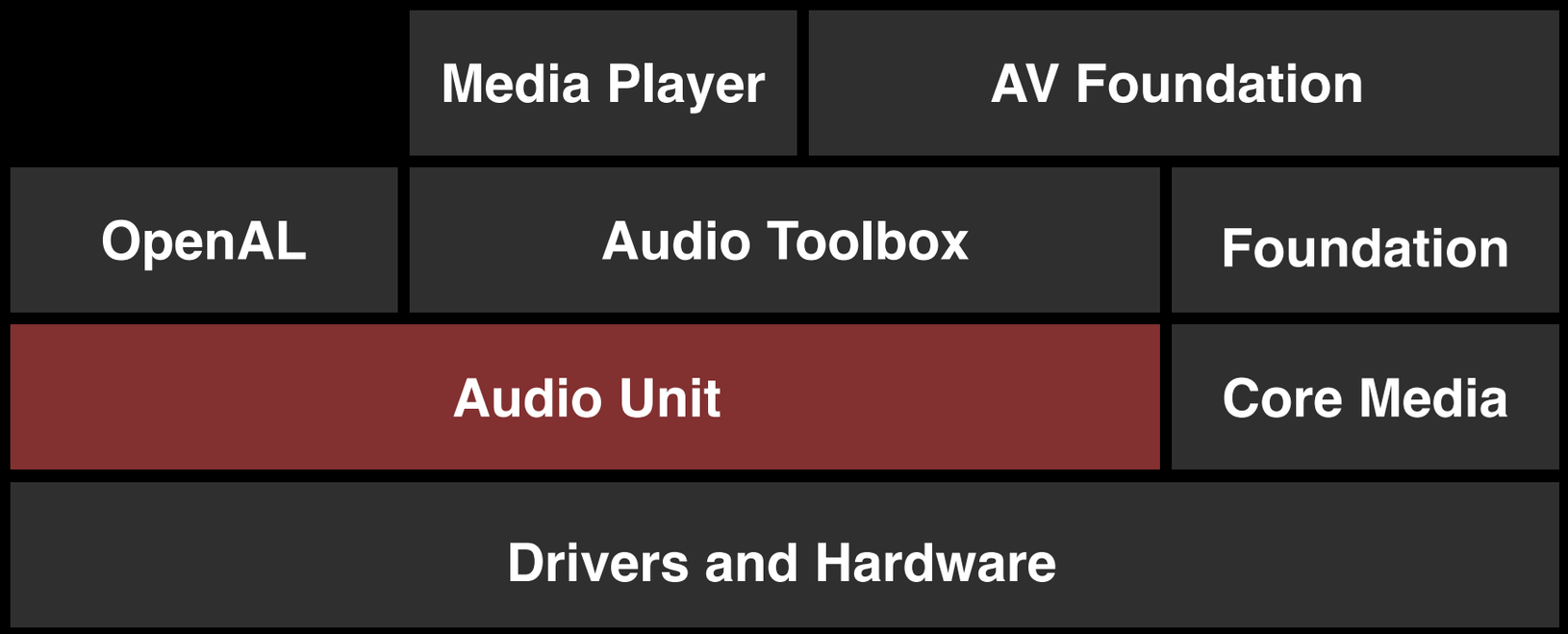
- App plays audio
- App registers for location updates
- App provides Voice over IP services
- App processes Newsstand Kit downloads
- App communicates with an accessory
- App communicates using CoreBluetooth
- App shares data using CoreBluetooth

@jsanchezsierra
http://ccrma.stanford.edu/~jsanchez

Core Audio

Core Audio

Audio Units



@jsanchezsierra
http://ccrma.stanford.edu/~jsanchez

Audio Units

Core Audio

When to use the Audio Units

- Very specific needs
- Low Latency I/O
- Responsive playback of synthesized sounds
- Use of built-in features (echo cancelation, mixing, paning...)

Where to use Audio Units

- VoIP Apps (using Voice Processing I/O unit)
- Interactive music apps (mixer unit)
- For real time I/O processing

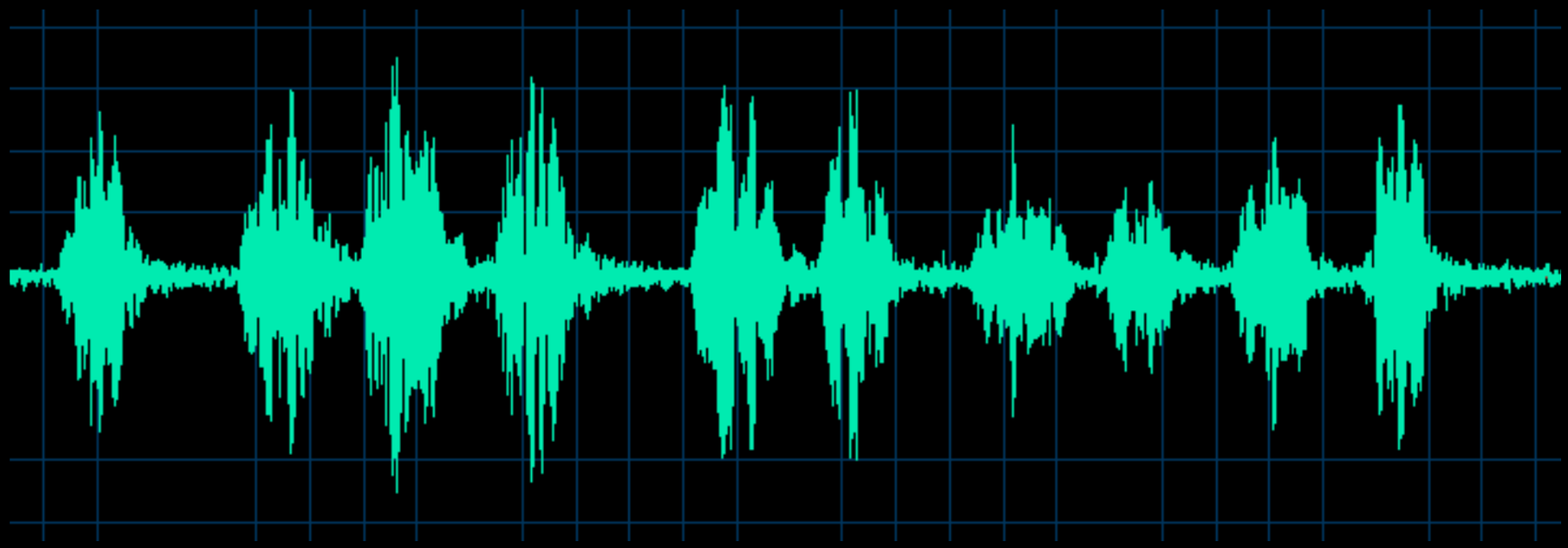
@jsanchezsierra
http://ccrma.stanford.edu/~jsanchez

Digital audio basic concepts

Digital Audio

Basics

Digital audio representation of a sound



An analog waveform is represented digitally using sines and cosines expressions
Discrete representation -> sampling rate, bit sample

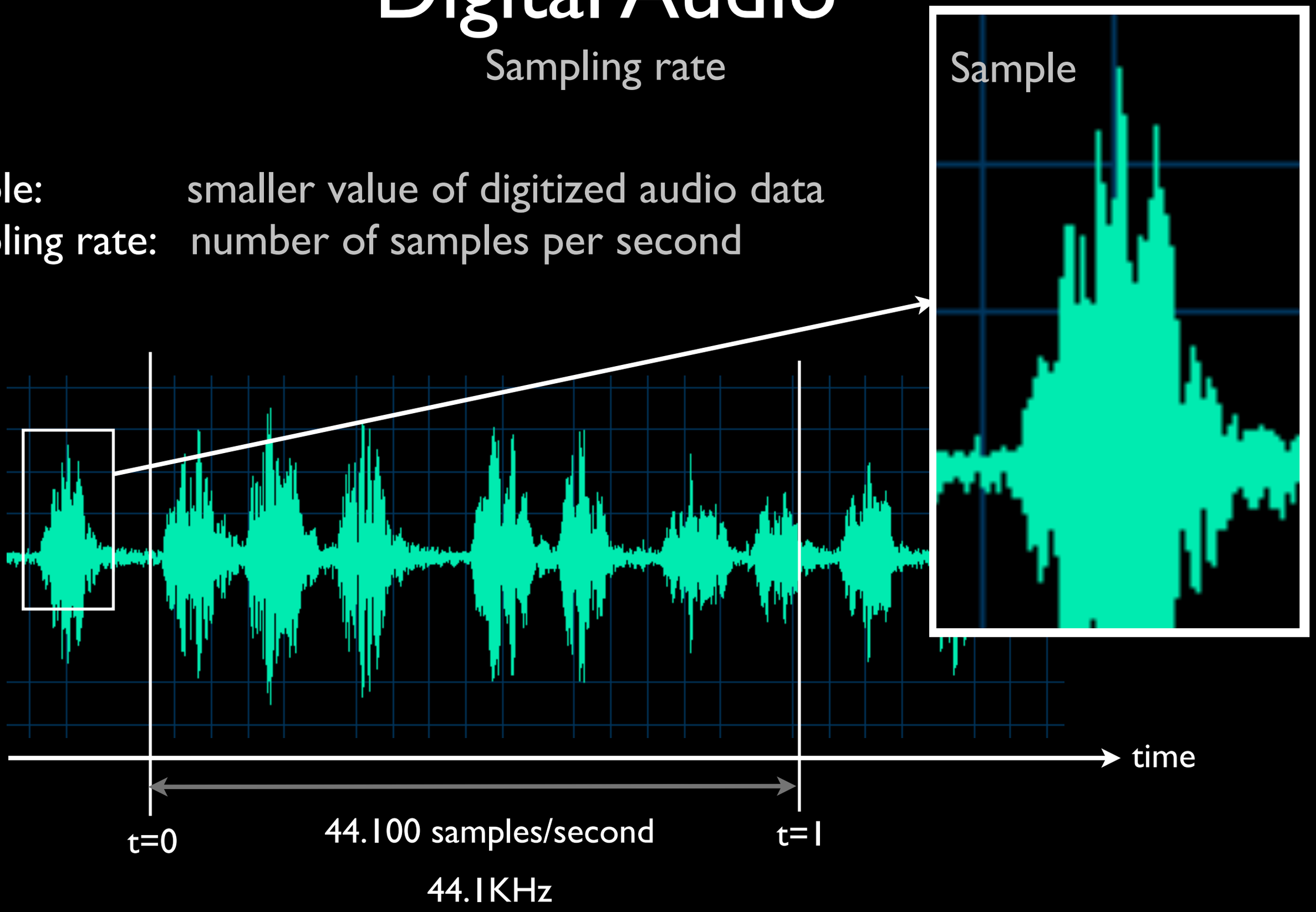
@jsanchezsierra
http://ccrma.stanford.edu/~jsanchez

Digital Audio

Sampling rate

Sample: smaller value of digitized audio data

Sampling rate: number of samples per second



@jsanchezsierra
http://ccrma.stanford.edu/~jsanchez

Digital Audio

Sampling rate

Sampling rate	Quality
8 KHz	Narrow-band speech
16 KHz	Narrow-band speech
44.1 KHz	CD quality
48 KHz	Digital Audio Tape
96 KHz	Pro Quality
192 KHz	Ultimate marketing quality

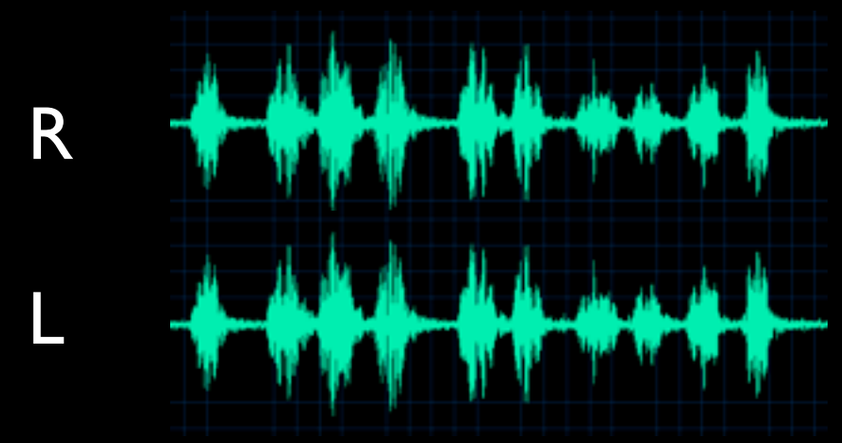
@jsanchezsierra
http://ccrma.stanford.edu/~jsanchez

Digital Audio

Channels, frames

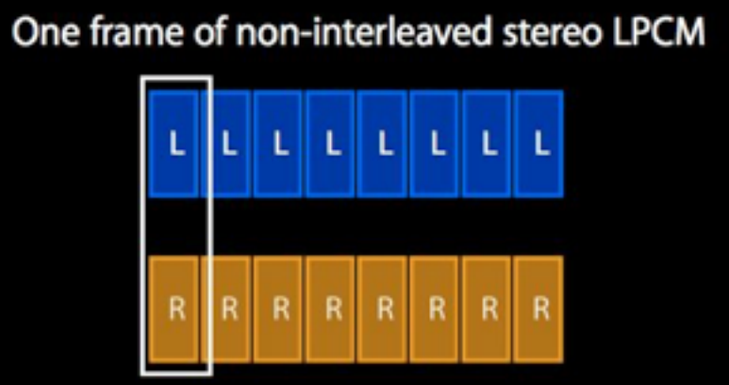
Stereo sound: 2 channels: R & L

Frames: A collection of samples for each channel. Stereo has 2 frames



same buffer for both channel

two buffers, one per channel

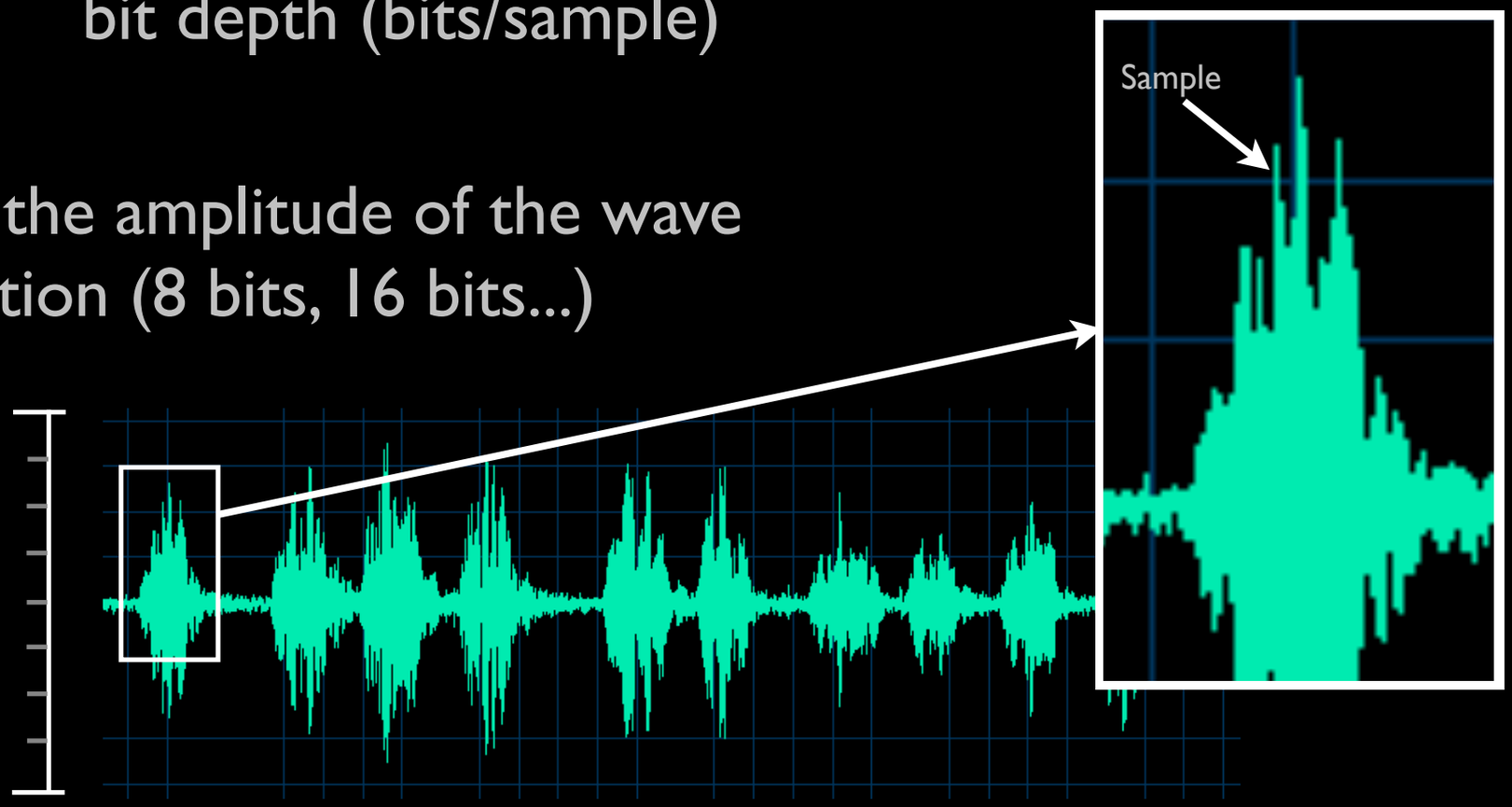


@jsanchezsierra
http://ccrma.stanford.edu/~jsanchez

Digital Audio

bit depth (bits/sample)

-Each sample represents the amplitude of the wave
-Bit depth: sample resolution (8 bits, 16 bits...)



if the difference between two consecutive sounds is smaller than the sample resolution, the difference is lost

bit rate (bits/second)= number of channels x bit depth (bits/sample) x sampling rate (samples/second)
bit rate= 2 x16 bits/sample x 44100 samples/second = 1,411,200 bits/second = 1,411 Kbits/second

80 minutes of audio, stereo 16bits, 44.1Khz, CD Quality
bit rate= 80 minutes x 60 ssconds/minute x 2 channels x16 bits/sample x 44100 samples/second / 8 bits/byte = 846,720,000 bytes = 847 Mb

80 minutes = 847 Mb Hardware limitations!!

@jsanchezsierra
http://ccrma.stanford.edu/~jsanchez

Audio Formats

File Format / Data Formats

File Format	Data Formats
AAC (.aac, .adts)	'aac '
AC3 (.ac3)	'ac-3'
AIFC (.aif, .aiff,.aifc)	BEI8, BEI16, BEI24, BEI32, BEF32, BEF64, 'ulaw', 'alaw', 'MAC3', 'MAC6', 'ima4' , 'QDMC', 'QDM2', 'Qclp', 'agsm'
AIFF (.aiff)	BEI8, BEI16, BEI24, BEI32
Apple Core Audio Format (.caf)	'.mp3', 'MAC3', 'MAC6', 'QDM2', 'QDMC', 'Qclp', 'Qclq', 'aac ', 'agsm', 'alac', 'alaw', 'drms', 'dvi ', 'ima4', 'lpc ', BEI8, BEI16, BEI24, BEI32, BEF32, BEF64, LEI16, LEI24, LEI32, LEF32, LEF64, 'ms\x00\x02', 'ms\x00\x11', 'ms\x001', 'ms\x00U', 'ms\x00', 'samr', 'ulaw'
MPEG Layer 3 (.mp3)	'.mp3'
MPEG 4 Audio (.mp4)	'aac '
MPEG 4 Audio (.m4a)	'aac ', alac'
NeXT/Sun Audio (.snd, .au)	BEI8, BEI16, BEI24, BEI32, BEF32, BEF64, 'ulaw'
Sound Designer II (.sd2)	BEI8, BEI16, BEI24, BEI32
WAVE (.wav)	LEUI8, LEI16, LEI24, LEI32, LEF32, LEF64, 'ulaw', 'alaw'

Audio Formats

LPCM / compressed formats

-LPCM (Linear Pulse Code Modulation)

Uncompressed format

One packet = one frame

Constant bit rate (CBR)

number of frames per packet = 1

-Packetize compressed formats

A group of frames of LPCM is compressed into a packet. Packets have dependencies on preceding packets.

Variable bit rate (VBR)

number of frames per packet (AAC has1024 frames/packet)

@jsanchezsierra
http://ccrma.stanford.edu/~jsanchez

Audio Formats

Get audio file information

> afinfo mySong.mp3

```
> afinfo mySong.mp3

File:           mySong.mp3
File type ID:   MPG3
Num Tracks:     1
----
Data format:     2 ch,  44100 Hz, '.mp3' (0x00000000) 0 bits/
channel, 0 bytes/packet, 1152 frames/packet, 0 bytes/frame
                no channel layout.
estimated duration: 274.745600 sec
audio bytes: 5494912
audio packets: 10517
bit rate: 160000 bits per second
packet size upper bound: 1052
maximum packet size: 523
audio data file offset: 0
optimized
```

```
> afinfo Sound2.caf

File:           Sound2.caf
File type ID:   caff
Num Tracks:     1
----
Data format:     1 ch,  44100 Hz, 'lpcm' (0x0000000C) 16-bit little-endian
signed integer
                no channel layout.
estimated duration: 0.328345 sec
audio bytes: 28960
audio packets: 14480
bit rate: 705600 bits per second
packet size upper bound: 2
maximum packet size: 2
audio data file offset: 4096
optimized
audio 14480 valid frames + 0 priming + 0 remainder = 14480
source bit depth: 116
sound check:
    approximate duration in seconds          0.328
sound check volume normalization gain: 0.00 dB
----
```

@jsanchezsierra
http://ccrma.stanford.edu/~jsanchez

Audio Stream Basic Description

format properties of a stream of audio data

```
struct AudioStreamBasicDescription
{
    Float64 mSampleRate;
    UInt32  mFormatID;
    UInt32  mFormatFlags;
    UInt32  mBytesPerPacket;
    UInt32  mFramesPerPacket;
    UInt32  mBytesPerFrame;
    UInt32  mChannelsPerFrame;
    UInt32  mBitsPerChannel;
    UInt32  mReserved;
};
```

```
enum {
   kAudioFormatLinearPCM             = 'lpcm',
   kAudioFormatAC3                   = 'ac-3',
   kAudioFormat60958AC3              = 'cac3',
   kAudioFormatAppleIMA4             = 'ima4',
   kAudioFormatMPEG4AAC              = 'aac ',
   kAudioFormatMPEG4CELP             = 'celp',
   kAudioFormatMPEG4HVXC             = 'hvxc',
   kAudioFormatMPEG4TwinVQ           = 'twvq',
   kAudioFormatMACE3                 = 'MAC3',
   kAudioFormatMACE6                 = 'MAC6',
   kAudioFormatULaw                  = 'ulaw',
   kAudioFormatALaw                  = 'alaw',
   kAudioFormatQDesign               = 'QDMC',
   kAudioFormatQDesign2              = 'QDM2',
   kAudioFormatQUALCOMM              = 'Qclp',
   kAudioFormatMPEGLayer1            = '.mp1',
   kAudioFormatMPEGLayer2            = '.mp2',
   kAudioFormatMPEGLayer3            = '.mp3',
   kAudioFormatTimeCode              = 'time',
   kAudioFormatMIDIStream            = 'midi',
   kAudioFormatParameterValueStream  = 'apvs',
   kAudioFormatAppleLossless         = 'alac'
   kAudioFormatMPEG4AAC_HE           = 'aach',
   kAudioFormatMPEG4AAC_LD           = 'aacl',
   kAudioFormatMPEG4AAC_ELD          = 'aace',
   kAudioFormatMPEG4AAC_ELD_SBR      = 'aacf',
   kAudioFormatMPEG4AAC_HE_V2        = 'aacp',
   kAudioFormatMPEG4AAC_Spatial      = 'aacs',
   kAudioFormatAMR                   = 'samr',
   kAudioFormatAudible               = 'AUDB',
   kAudioFormatiLBC                  = 'ilbc',
   kAudioFormatDVIIntelIMA           = 0x6D730011,
   kAudioFormatMicrosoftGSM          = 0x6D730031,
   kAudioFormatAES3                  = 'aes3'
};
```

```
Compressed audio (VBR)
   mBytesPerPacket = 0;
   mBytesPerFrame = 0;
   mBitsPerChannel = 0;
```

@jsanchezsierra
http://ccrma.stanford.edu/~jsanchez

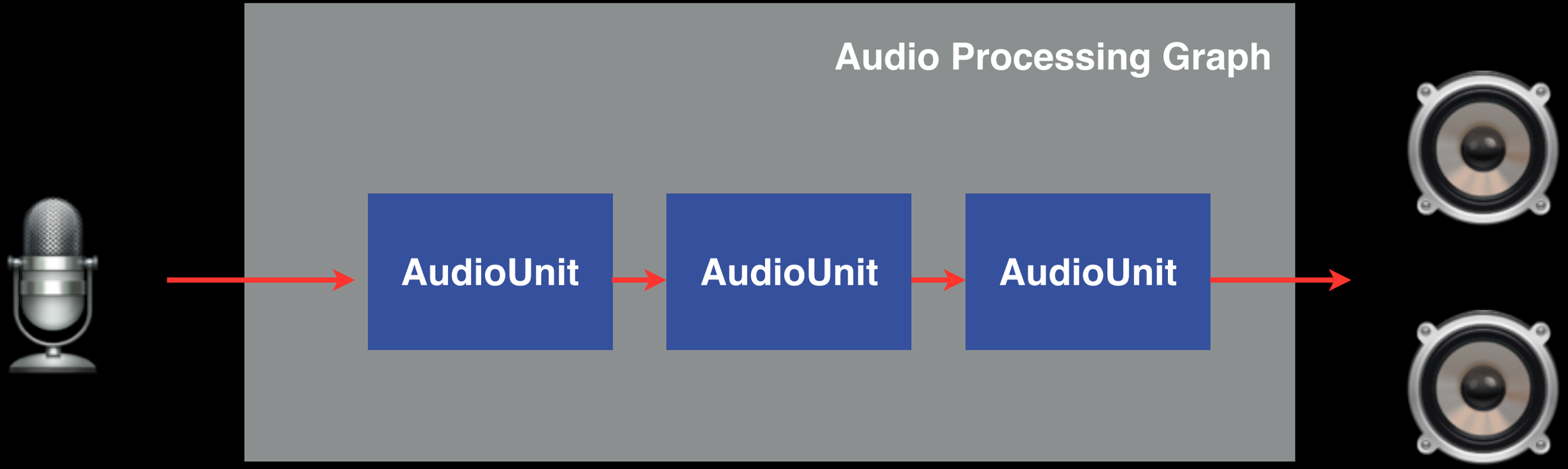
Audio Units

Core Audio

Audio Units

Some definitions

- Audio Processing Graph: An object that manages a network of audio unit nodes
- Audio Unit: audio processing plug-in component
- Audio Node: representation of Audio Unit in the context of an Audio Processing Graph



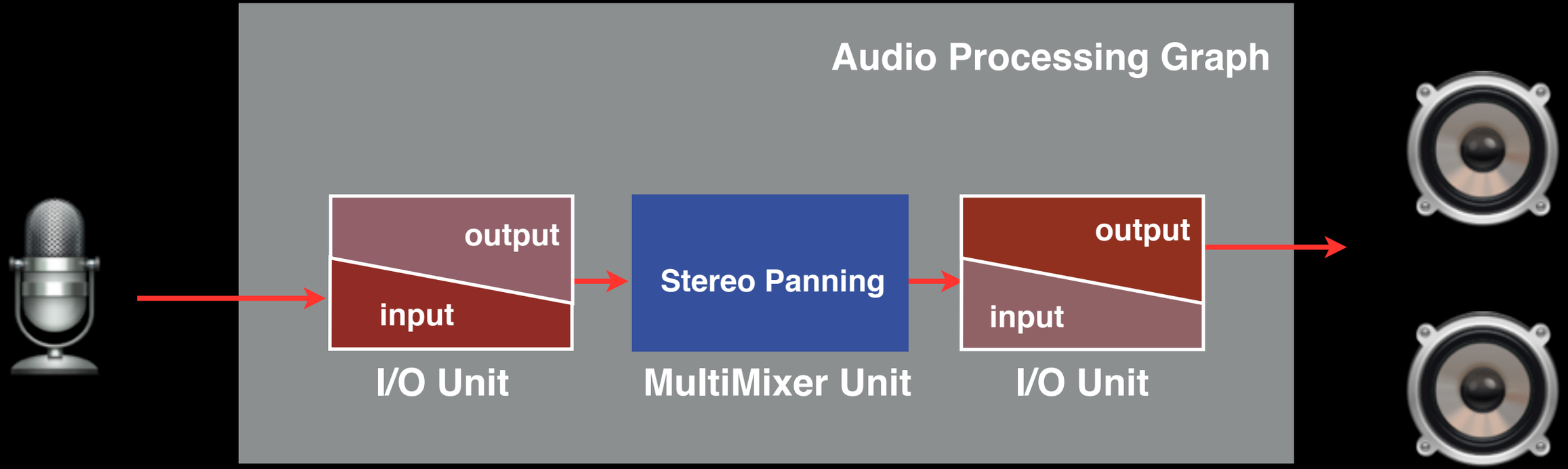
Property / Parameter

@jsanchezsierra
http://ccrma.stanford.edu/~jsanchez

Audio Units

Some definitions

- Audio Processing Graph: An object that manages a network of audio unit nodes
- Audio Unit: audio processing plug-in component
- Audio Node: representation of Audio Unit in the context of an Audio Processing Graph



@jsanchezsierra
http://ccrma.stanford.edu/~jsanchez

Audio Units

Available Units for OSX

- Generator units: stream of audio from files, network, memory
- Instrument units: stream of synthesized audio from MIDI data.
- Mixer units: Combine multiple streams into one or more streams.
- Effect units: digital signal processing, reverb, pitch change, noise filtering...
- Converter units: Perform transformations (change sample rate, bit depth, adjust playback speed...)
- Output units: Interface with audio input and/or output hardware. Input/output

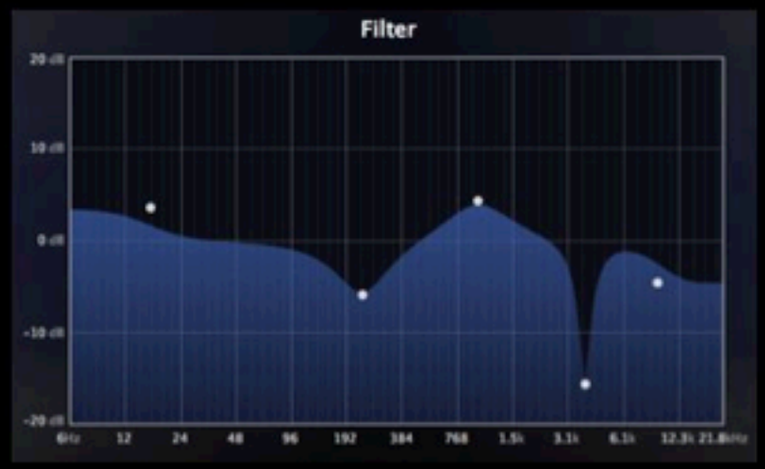
Available Units for iOS

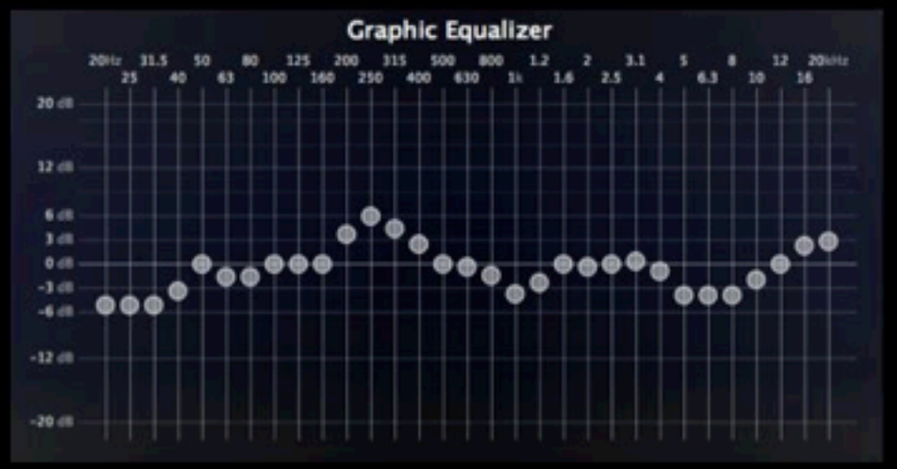
- Effect units: iPod Equalizer
- Mixing units : 3D Mixer / Multichannel Mixer
- I/O Unit: Remote I/O, Voice-Processing I/O, Generic Output
- Format conversion: Format converter

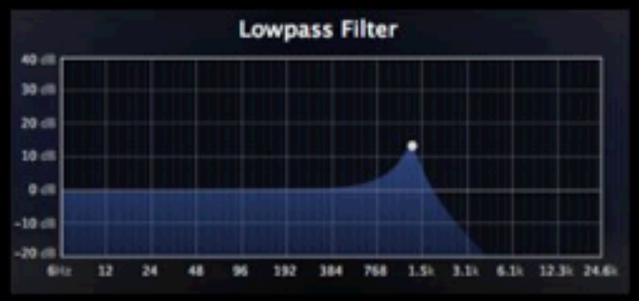
@jsanchezsierra
http://ccrma.stanford.edu/~jsanchez

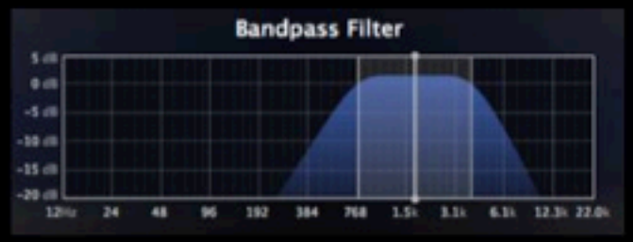
Filter Units

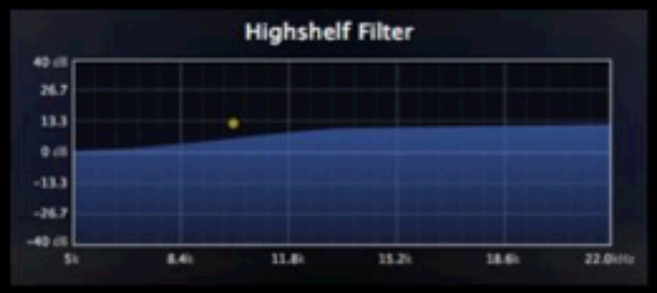
OSX

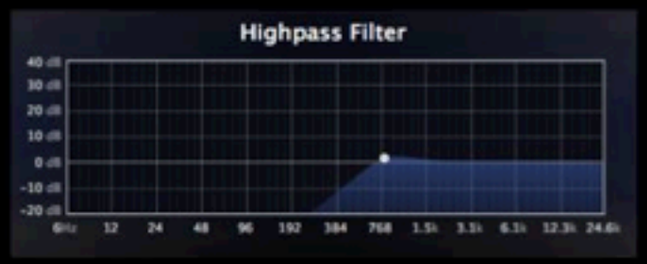


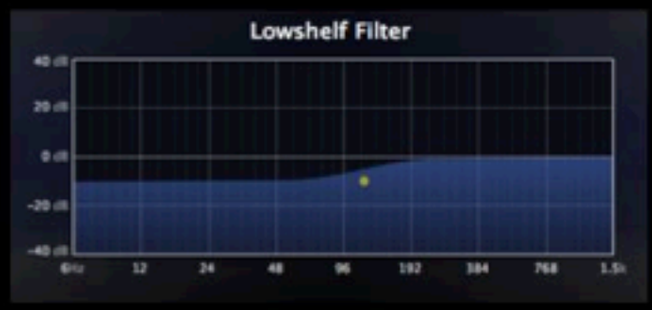








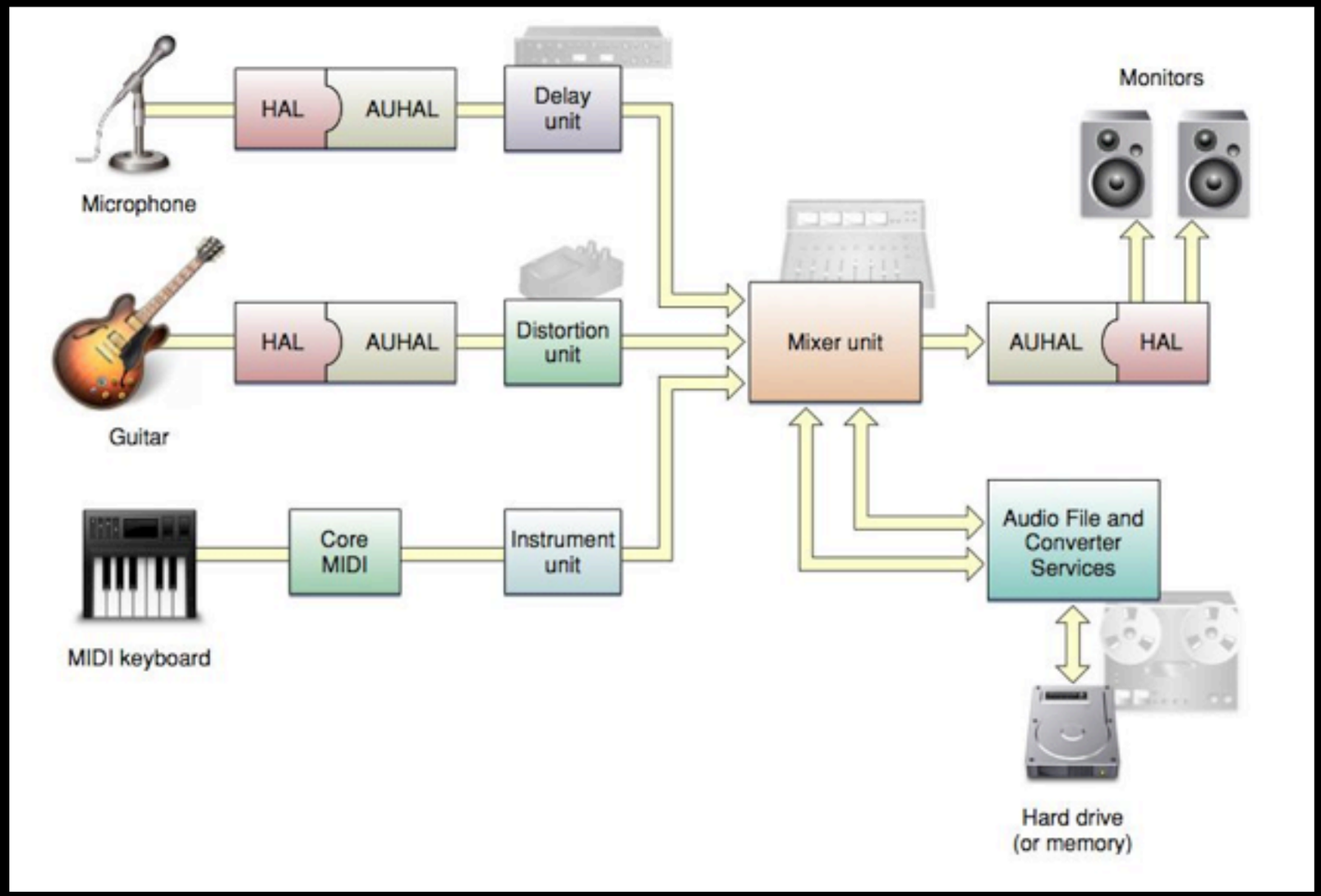




@jsanchezsierra
http://ccrma.stanford.edu/~jsanchez

Audio Units

other examples



@jsanchezsierra
http://ccrma.stanford.edu/~jsanchez

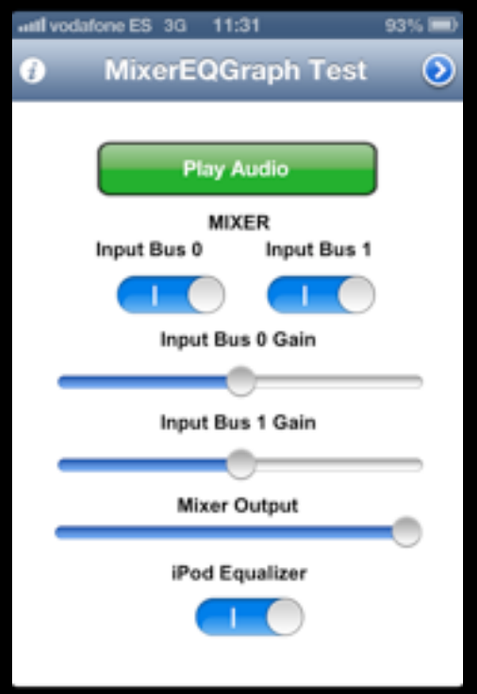
Demo 2

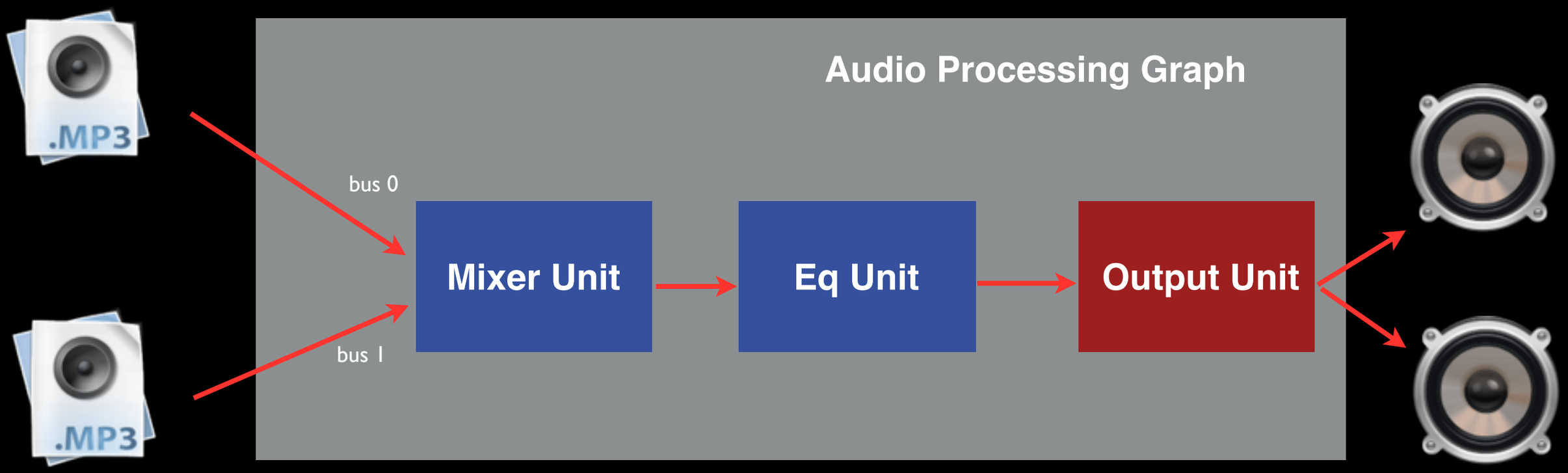
iPhoneMixerEQGraphTest (Apple code sample)

https://developer.apple.com/library/ios/samplecode/iPhoneMixerEQGraphTest/Introduction/Intro.html

Audio Unit example

iPhoneMixerEQGraphTest (apple code sample)





Audio Units

Creating an Audio Unit App

1. Create audio session
2. Specify audio units
3. Create a graph, then obtain the audio units
4. Configure the audio units
5. Connect the nodes
6. Provide a user interface
7. Initialize and then start the graph

@jsanchezsierra
http://ccrma.stanford.edu/~jsanchez

Audio Units

1.-Configure Audio Session

```
self.graphSampleRate=44100.0;

AVAudioSession *mySession = [AVAudioSession sharedInstance];
[mySession setPrederredHardwareSampleRate: graphSampleRate error: nil];
[mySession setCategory: AVAudioSessionCategoryPlayAnddRecord error: nil];
[mySession setActive: YES error: nil];
  self.graphSampleRate = [mySession currentHardwareSampleRate];
```

Sets the preferred hardware sample rate for input and output.

@jsanchezsierra
http://ccrma.stanford.edu/~jsanchez

Audio Units

2.-Specify Audio Units

```
// multichannel mixer unit
AudioComponentDescription mixer_desc;
mixer_desc.componentType        = kAudioUnitType_Mixer;
mixer_desc.componentSubyype     = kAudioUnitSubType_MultiChannelMixer;
mixer_desc.componentManufacturer =   kAudioUnitManufacturer_Apple;
mixer_desc.componentFlags       = 0;
mixer_desc.componentFlashsMask  = 0;

 // iPodEQ unit
AudioComponentDescription eq_desc;
eq_desc.componentType        = kAudioUnitType_Effect;
eq_desc.componentSubyype     = kAudioUnitSubType_AUiPodEQ;
eq_desc.componentManufacturer =   kAudioUnitManufacturer_Apple;
eq_desc.componentFlags       = 0;
eq_desc.componentFlashsMask  = 0;

// output unit
AudioComponentDescription output_desc;
output_desc.componentType       = kAudioUnitType_Output;
output_desc.componentSubyype    = kAudioUnitSubtype_RemoteIO;
output_desc.componentManufacturer =  kAudioUnitManufacturer_Apple;
output_desc.componentFlags      = 0;
output_desc.componentFlashsMask    = 0;
```

@jsanchezsierra
http://ccrma.stanford.edu/~jsanchez

Audio Units

3.- Create a graph, then obtain the audio units

```
// create a new AUGraph
AUGraph   mGraph;
result = NewAUGraph(&mGraph);

// Add Audio Nodes to graph
AUNode outputNode;
AUNode eqNode;
AUNode mixerNode;
AUGraphAddNode(mGraph, & mixer_desc, &mixerNode);
AUGraphAddNode(mGraph, & eq_desc, &eqNode);
AUGraphAddNode(mGraph, & output_desc, &outputNode);

// open the graph AudioUnits (but not initialized)
result = AUGraphOpen(mGraph);

// grab the audio unit instances from the nodes
AudioUnit mEQ;
AudioUnit mMixer;
result = AUGraphNodeInfo(mGraph, mixerNode, NULL, &mMixer);
result = AUGraphNodeInfo(mGraph, eqNode, NULL, &mEQ);
```

Audio Units

4.- Configure the audio units (AudioUnitSetProperty / AudioUnitGetProperty)

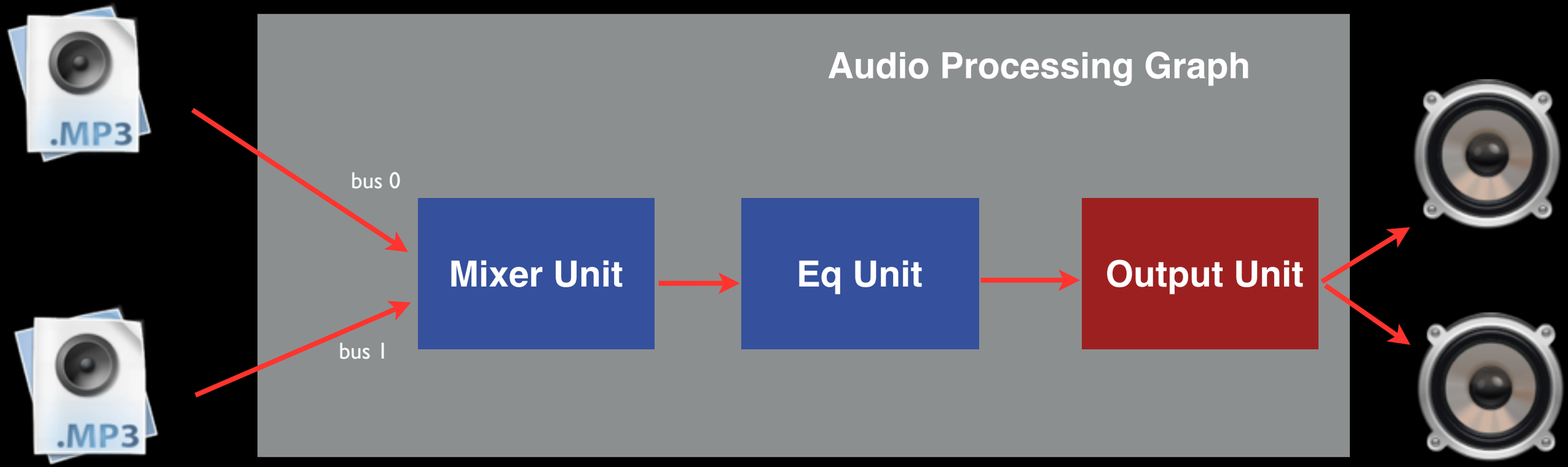
```
// set number of input buses for the mixer Audio Unit
UInt32 numbuses = 2;
AudioUnitSetProperty ( mMixer, kAudioUnitProperty_ElementCount,
                kAudioUnitScope_Input, 0, &numbuses, sizeof(numbuses));


// set a callback for the specified node's specified input
AURenderCallbackStruct rcbs=...;
AUGraphSetNodeInputCallback(mGraph, mixerNode, busNumber, &rcbs);
```

```
//Set mixer input & output format
CAStreamBasicDescription mClientFormat= ...;
CAStreamBasicDescription mOutputFormat= ...;
AudioUnitSetProperty(mMixer, kAudioUnitProperty_StreamFormat,
          kAudioUnitScope_Input, i, &mClientFormat, sizeof(mClientFormat));
AudioUnitSetProperty(mMixer, kAudioUnitProperty_StreamFormat,
          kAudioUnitScope_Output, 0, &mOutputFormat, sizeof(mOutputFormat));


 // get the equalizer factory presets list
 CFArrayRef mEQPresetsArray;
 UInt32 sizeof = sizeof(mEQPresetsArray);
 AudioUnitGetProperty(mEQ, kAudioUnitProperty_FactoryPresets,
                kAudioUnitScope_Global, 0, &mEQPresetsArray, &size);
```

Audio Units

5.- Connect the nodes

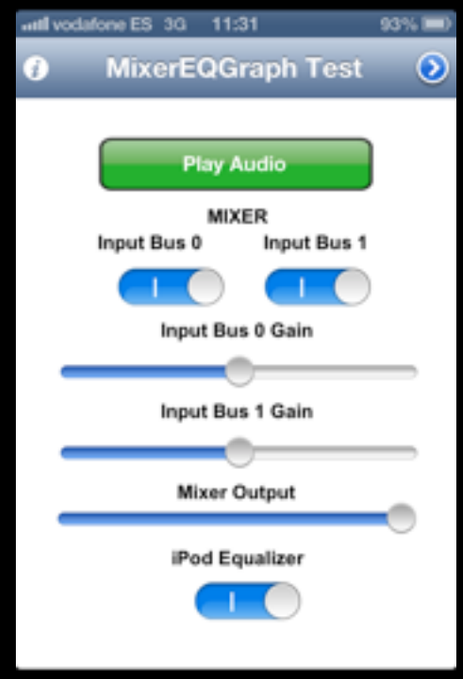
```
// connect a node's output to a node's input
// mixer -> eq -> output
result = AUGraphConnectNodeInput(mGraph, mixerNode, 0, eqNode, 0);
result = AUGraphConnectNodeInput(mGraph, eqNode, 0, outputNode, 0);
```



@jsanchezsierra
http://ccrma.stanford.edu/~jsanchez

Audio Units

6.- Provide a user interface




```
// Mixer Unit - Change input volumen for inputNum bus
AudioUnitSetParameter(mMixer, kMultiChannelMixerParam_Volume,
                      kAudioUnitScope_Input, busNumber, value, 0);


// Mixer Unit - Change output volumen for bus 0
AudioUnitSetParameter(mMixer, kMultiChannelMixerParam_Volume,
                      kAudioUnitScope_Output, 0, value, 0);


// Mixer Unit - Enable bus
AudioUnitSetParameter(mMixer, kMultiChannelMixerParam_Enable,
                      kAudioUnitScope_Input, busNumber, isONValue, 0);


// Equalizer Unit - Change equalizer preset from mEQPresetArray
AUPreset *aPreset = (AUPreset*)CFArrayGetValueAtIndex(mEQPresetsArray, presetIndex);
AudioUnitSetProperty (mEQ, kAudioUnitProperty_PresentPreset,
            kAudioUnitScope_Global, 0, aPreset, sizeof(AUPreset));
```


@jsanchezsierra
http://ccrma.stanford.edu/~jsanchez

Audio Units

7.- Initialize and start the graph

```
AUGraphInitialize(mGraph);

AUGraphStart(mGraph);

...
```

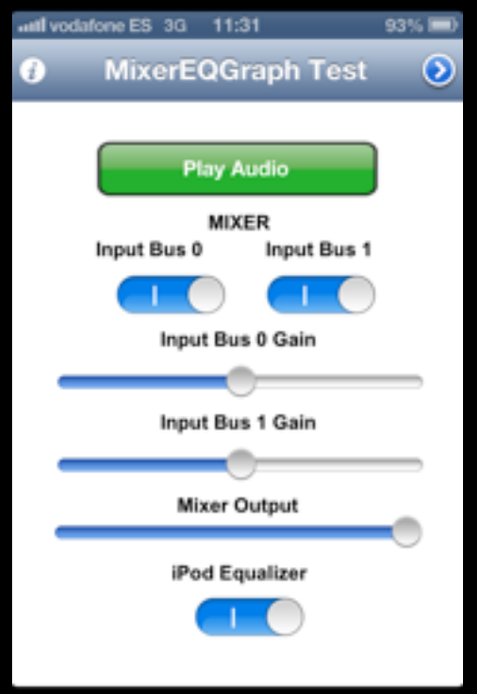
```
AUGraphStop(mGraph);


...
```

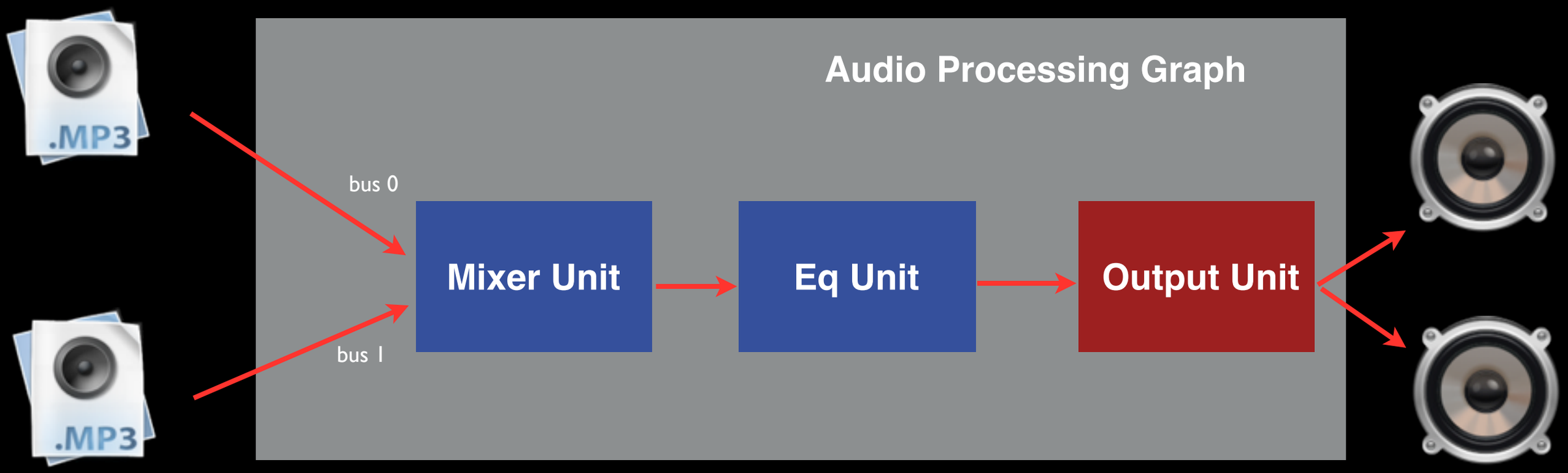
```
Boolean isRunning;
AUGraphIsRunning(mGraph, &isRunning);
```

@jsanchezsierra
http://ccrma.stanford.edu/~jsanchez

Audio Unit example

iPhoneMixerEQGraphTest (apple code sample)





Spatial Audio

3DMixer Unit / OpenAL

Demo 3

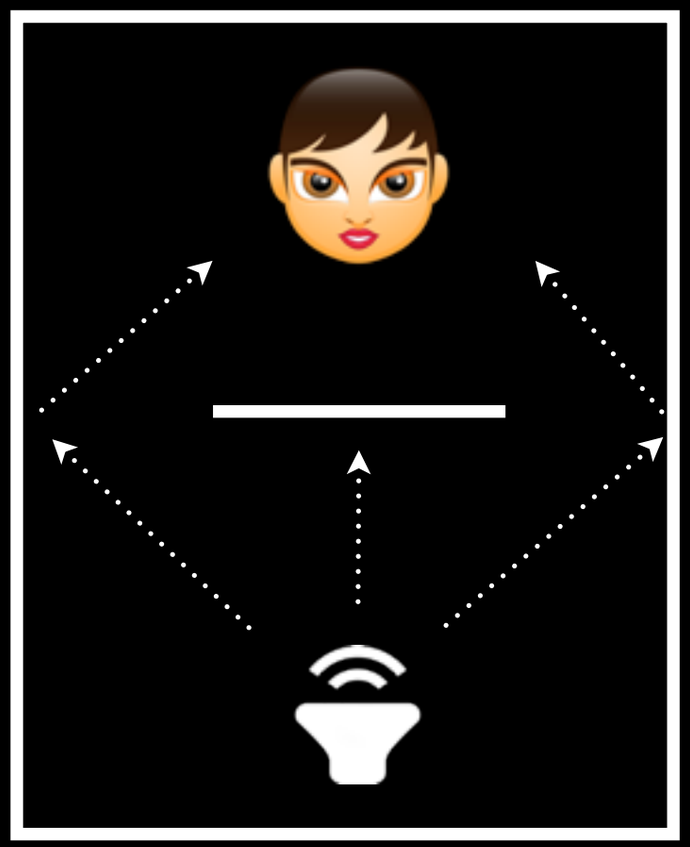
oalTouch (Apple sample code)

https://developer.apple.com/library/ios/samplecode/oalTouch/Introduction/Intro.html

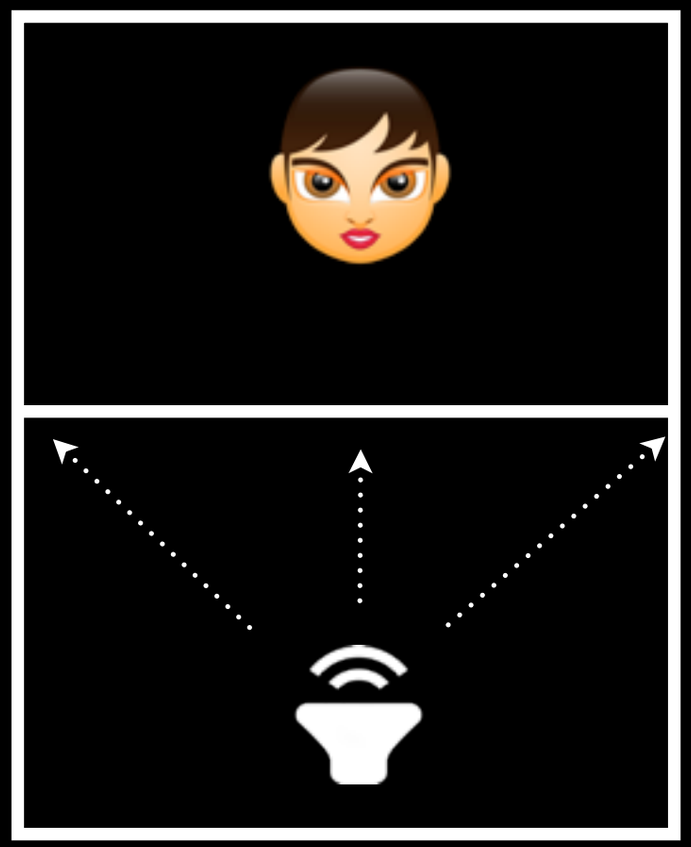
3DMixer Audio Unit

Fundamentals

- One listener and multiple sound sources
- 3D Audio (panning, audio cues, reverb, obstruction, occlusion)



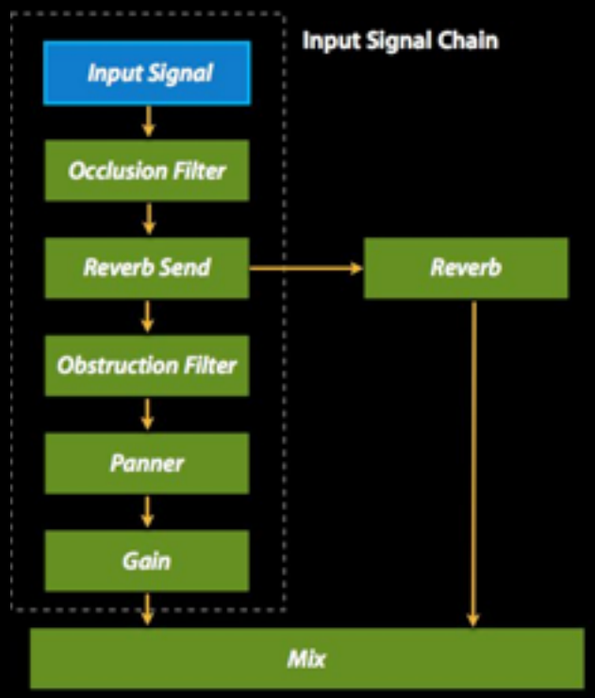
obstruction



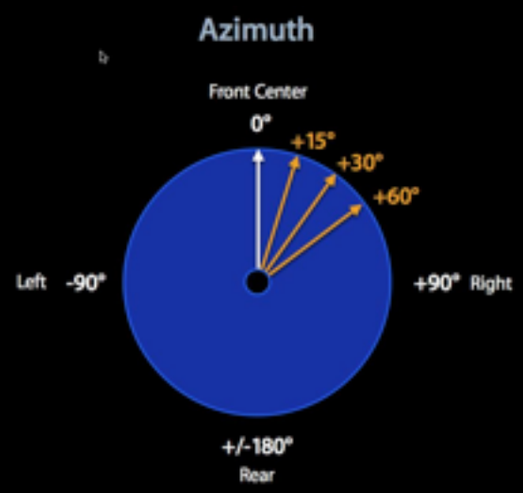
occlusion

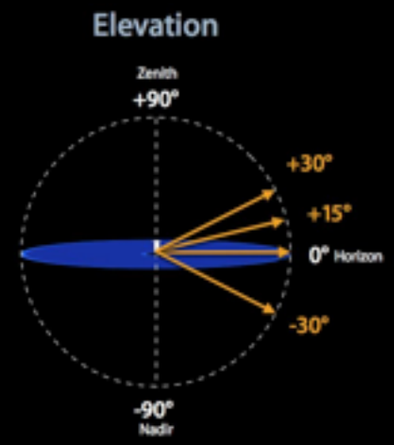
3DMixer Audio Unit

Parameters




```
enum {
    k3DMixerParam_Azimuth                = 0,
    k3DMixerParam_Elevation              = 1,
    k3DMixerParam_Distance               = 2,
    k3DMixerParam_Gain                   = 3,
    k3DMixerParam_PlaybackRate           = 4,
    k3DMixerParam_Enable                 = 5,
    k3DMixerParam_MinGain                = 6,
    k3DMixerParam_MaxGain                = 7,
    k3DMixerParam_ReverbBlend            = 8,
    k3DMixerParam_GlobalReverbGain       = 9,
    k3DMixerParam_OcclusionAttenuation   = 10,
    k3DMixerParam_ObstructionAttenuation = 11
};
```





OpenAL

Fundamentals

- Open standard audio API for 3D spatial audio (complement to Open GL)
- Available on OSX and iOS.

```
// Device Management
alcOpenDevice( const ALCchar *devicename );
alcCloseDevice( ALCdevice *device );

// Create Open AL Context
alcCreateContext (ALCdevice *device, <#const ALCint *attrlist#>)
alcMakeContextCurrent (<#ALCcontext *context#>)


// Creating a source
alGenSources(<#ALsizei n#>, <#ALuint *sources#>)

// Creating a buffer and fill it
alGenBuffers(<#ALsizei n#>, <#ALuint *buffers#>)


// Attach OpenAL buffer to OpenAL source
alSourcei (source, AL__BUFFER, buffer);
```

OpenAL

Fundamentals

```
// Set source attributes
alSourcefv(source, AL_POSITION, 0.8);
alSourcef (source, AL_REFERENCE_DISTACE, 4);
alSourcei (source, AL__BUFFER, 3);

// Set listener attributes
alSourcefv(AL_POSITION ,listener_position);
alSourcefv(AL_POSITION ,listener_orientation);


// Play a sound
alSourcefv(AL_POSITION ,listenerposiont);


// Move source&listener position
alSourcefv(source,AL_POSITION ,2.0);
alSourcefv(AL_POSITION ,listener_position);
```

OpenAL Extensions

Reverb, Occlusion, Obstruction

ASA Extension (Apple Spatial Audio)

```
ALC_ASA_REVERB_ON
ALC_ASA_REVERB_GLOBAL_LEVEL
ALC_ASA_REVERB_ROOM_TYPE
    ALC_REVERB_ROOM_TYPE_SmallRoom
    ALC_REVERB_ROOM_TYPE_MediumRoom
    ALC_REVERB_ROOM_TYPE_LargeRoom
    ALC_REVERB_ROOM_TYPE_MediumHall
    ALC_REVERB_ROOM_TYPE_LargeHall
    ALC_REVERB_ROOM_TYPE_Cathedral
    ALC_REVERB_ROOM_TYPE_Plate
    ALC_REVERB_ROOM_TYPE_MediumChamber
    ALC_REVERB_ROOM_TYPE_LargeChamber
    ALC_REVERB_ROOM_TYPE_LargeRoom2
    ALC_REVERB_ROOM_TYPE_MediumHall2
    ALC_REVERB_ROOM_TYPE_MediumHall3
    ALC_REVERB_ROOM_TYPE_LargeHall2
ALC_ASA_EQ_GAIN
ALC_ASA_EQ_BANDWIDTH
ALC_ASA_EQ_FREQ
ALC_ASA_REVERB_SEND_LEVEL
ALC_ASA_OCCLUSION
ALC_ASA_OBSTRUCTION
```

```
//Set a listener property
ALuint setting = 1;
alcASASetListenerProc(alcGetEnumValue(NULL, "ALC_ASA_REVERB_ON"), &setting,sizeof(setting));

//Set a source property
ALfloat level = 0.4;
alcASASetSourceProc(alcGetEnumValue(NULL, "ALC_ASA_REVERB_SEND_LEVEL"),source, &level, sizeof(level));
```

Source Notifications Extension

```
AL_SOURCE_STATE
    AL_INITIAL
    AL_PLAYING
    AL_PAUSED
    AL_STOPPED
AL_BUFFERS_PROCESSED
AL_QUEUE_HAS_LOOPED
```

Core Audio

other interesting projects

AudioGraph

https://github.com/tkzic/audiograph

Novocaine

https://github.com/alexbw/novocaine

NVDSP (with novocaine)

https://github.com/bartolsthoorn/NVDSP

AurioTouch2

https://developer.apple.com/library/ios/samplecode/aurioTouch2/Introduction/Intro.html

DSP (Digital signal processing), FFT (Fast Fourier Transform), DFT (Discrete Fourier Transform)

Core Audio

Now you have the basics to start discovering the power of this framework...

Apple Documentation

AudioSession Programming Guide
https://developer.apple.com/library/ios/documentation/Audio/Conceptual/AudioSessionProgrammingGuide/AudioSessionProgrammingGuide.pdf

iPod Library Access Programming Guide
https://developer.apple.com/library/iosdeveloper.apple.com/documentation/Audio/Conceptual/iPodLibraryAccess_Guide/iPodLibraryAccess_Guide.pdf

AudioQueue Programming Guide
https://developer.apple.com/library/mac/documentation/MusicAudio/Conceptual/AudioQueueProgrammingGuide/AudioQueueProgrammingGuide.pdf

CoreAudio Overview
https://developer.apple.com/library/mac/documentation/MusicAudio/Conceptual/CoreAudioOverview/CoreAudioOverview.pdf

CoreAudio Framework Reference
https://developer.apple.com/library/ios/documentation/MusicAudio/Reference/CACoreAudioReference/CACoreAudioReference.pdf

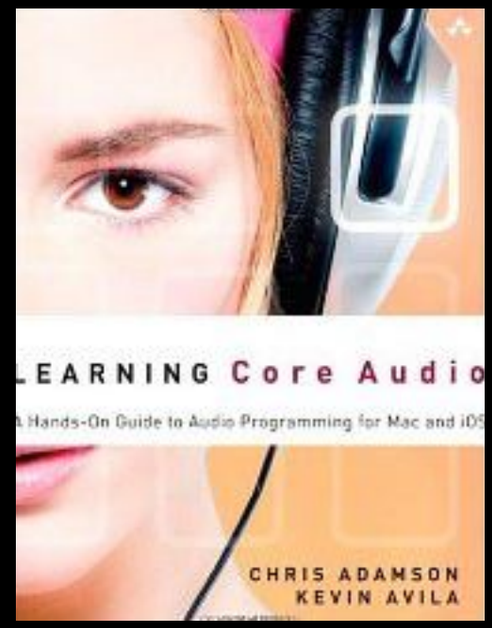
Core Audio Format Specification
https://developer.apple.com/library/mac/documentation/MusicAudio/Reference/CAFSpec/CAFSpec.pdf

AudioUnitHostingGuideForiOS
https://developer.apple.com/library/ios/documentation/MusicAudio/Conceptual/AudioUnitHostingGuide_iOS/AudioUnitHostingGuideForiOS.pdf

@jsanchezsierra
http://ccrma.stanford.edu/~jsanchez

Book

Learning Core Audio: A hands-on Guide to Audio Programming for Mac and iOS
http://www.amazon.com/Learning-Core-Audio-Hands-On-Programming/dp/0321636848
Book Sample Code
http://www.informit.com/content/images/9780321636843/downloads/9780321636843learning-core-audio-xcode4-projects-jan-23-2013.zip



WWDC 2010

Session 411 - Fundamentals of Digital Audio for Mac OS X and iPhone OS
Session 412 - Audio Development for iPhone OS, Part 1
Session 413 - Audio Development for iPhone OS, Part 2

WWDC 2011

Session 404 - Audio Development for Games
Session 411 - Music in iOS and Lion
Session 413 - Audio Session Management for iOS

WWDC 2012

Session 505 - Audio Session and Multiroute in iOS

WWDC 2013

Session 602-What's New in Core Audio for iOS

@jsanchezsierra
http://ccrma.stanford.edu/~jsanchez

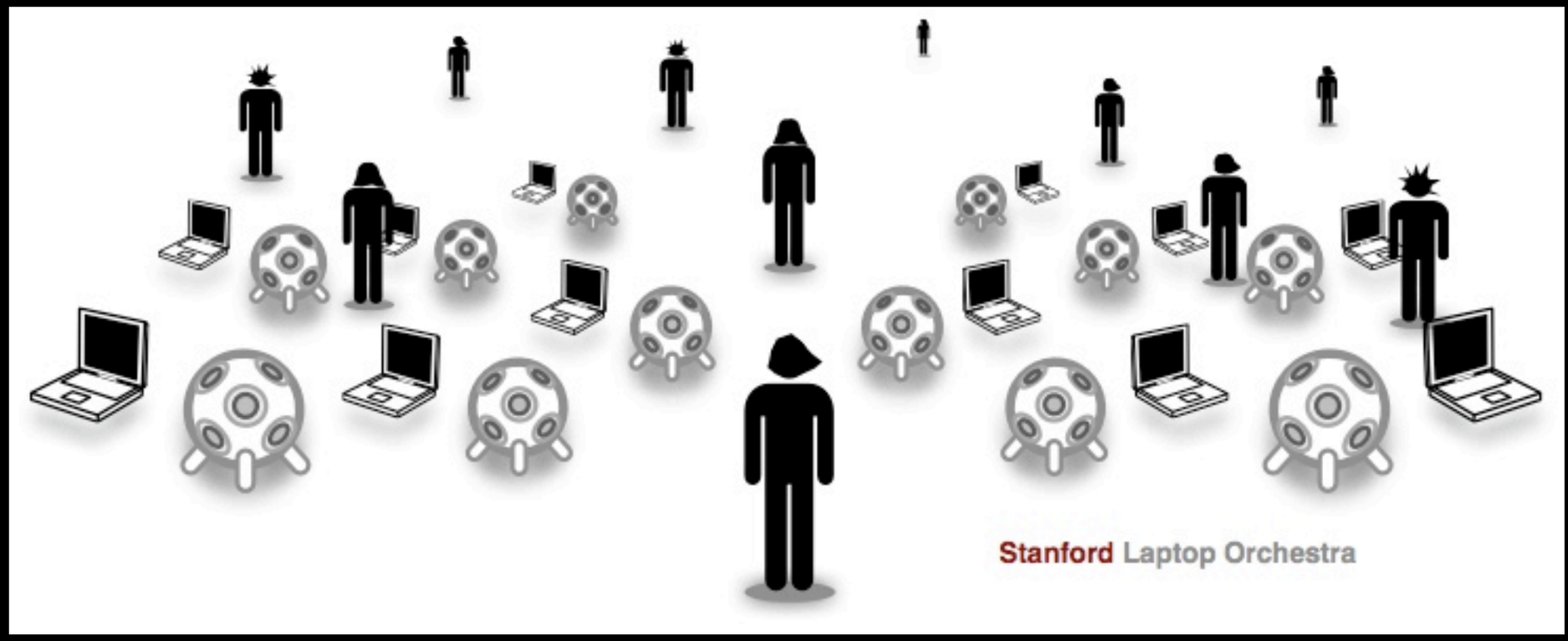
CCRMA / Stanford University

Center for Computer Research in Music and Acoustics

Visiting scholar from 2008-2011

SLOrk

Stanford Laptop Orchestra



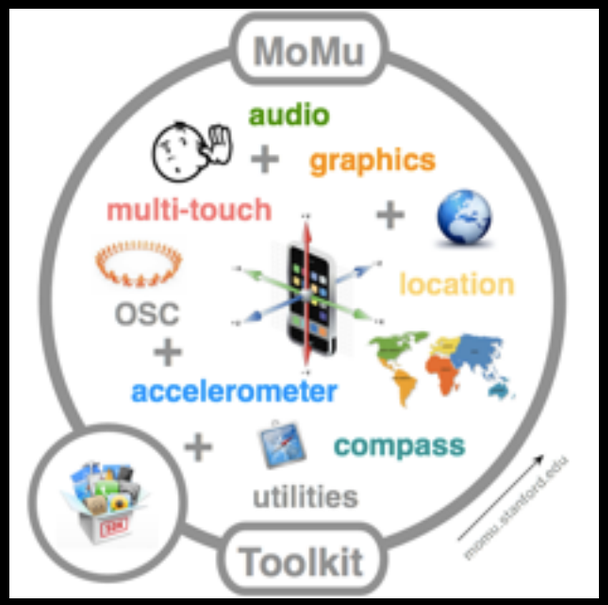
Spring 2008

MoMu Toolkit (Mobile Music at Stanford)

http://momu.stanford.edu/toolkit/

MoMu is a light-weight software toolkit for creating musical instruments and experiences on mobile device, and currently supports the iPhone platform (iPhone, iPad, iPod Touches).

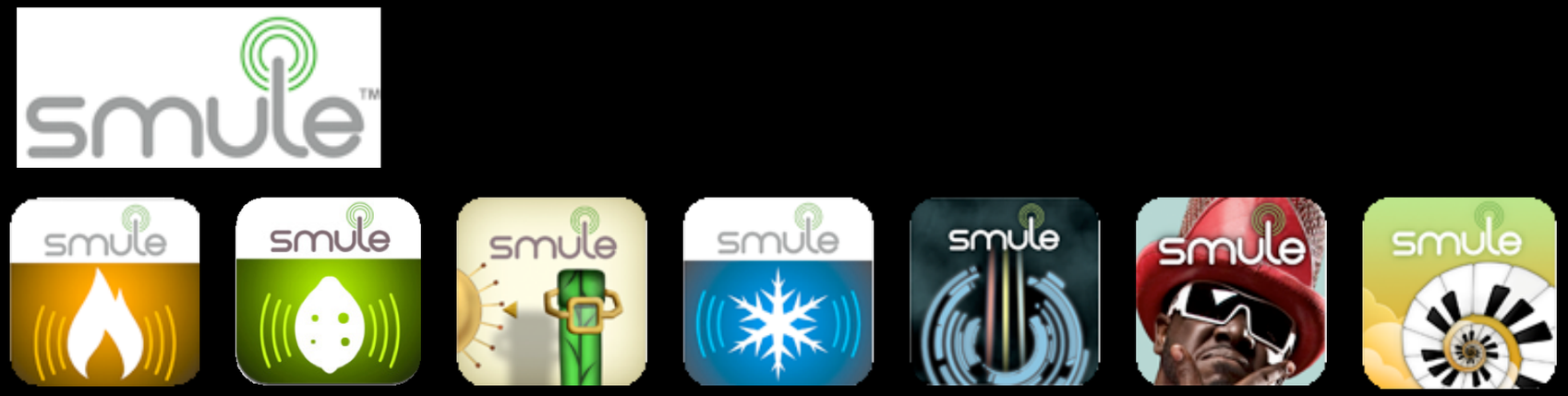
MoMu provides API's for real-time full-duplex audio, accelerometer, location, multi-touch, networking (via OpenSoundControl), graphics, and utilities.



The Synthesis Toolkit (STK)

http://momu.stanford.edu/stk/

@jsanchezsierra
http://ccrma.stanford.edu/~jsanchez



Stanford Mobile Phone Orchestra (MoPhO) at CCRMA, Ge Wang

Core Audio, low-level, DSP

Funding: $25.5M since 2008

Ocarina for iPhone - Nov 2008

http://www.youtube.com/watch?v=kfrONZjakRY

Magic Piano

Lang Lang, San Francisco Symphony Hall, April 19th, 2010

http://www.youtube.com/watch?v=HvpIGbCBaLA

Double Company - smule / Kush

http://www.youtube.com/watch?v=B9AUad-HEIE

Thanks!

Audio Recipes for iOS

AVFoundation / MPMediaPlayer / CoreAudio / OpenAL

Javier Sánchez

@jsanchezsierra

http://ccrma.stanford.edu/~jsanchez

Presentation: http://ccrma.stanford.edu/~jsanchez/NSSpain.pdf

NSSpain, September 17-19, Logroño, Spain