Real-time audio using Swift

Cocoaheads Berlin
April 15, 2015
Ariel Elkin
http://arielelkin.github.io

Plan

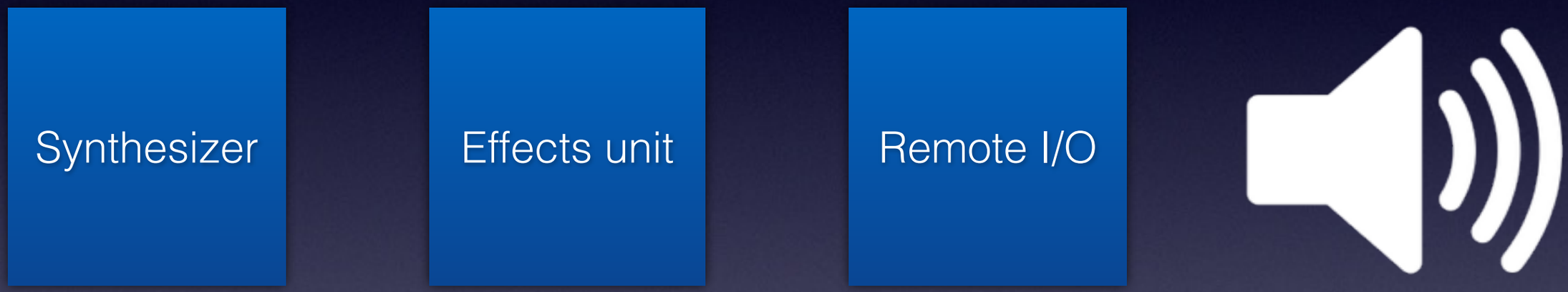
- Basics of real-time audio
  - Why?
  - How?
  - Core Audio
- Real-time audio using Swift
- Best Practices

Basics of real-time audio

Real-time audio: Why?

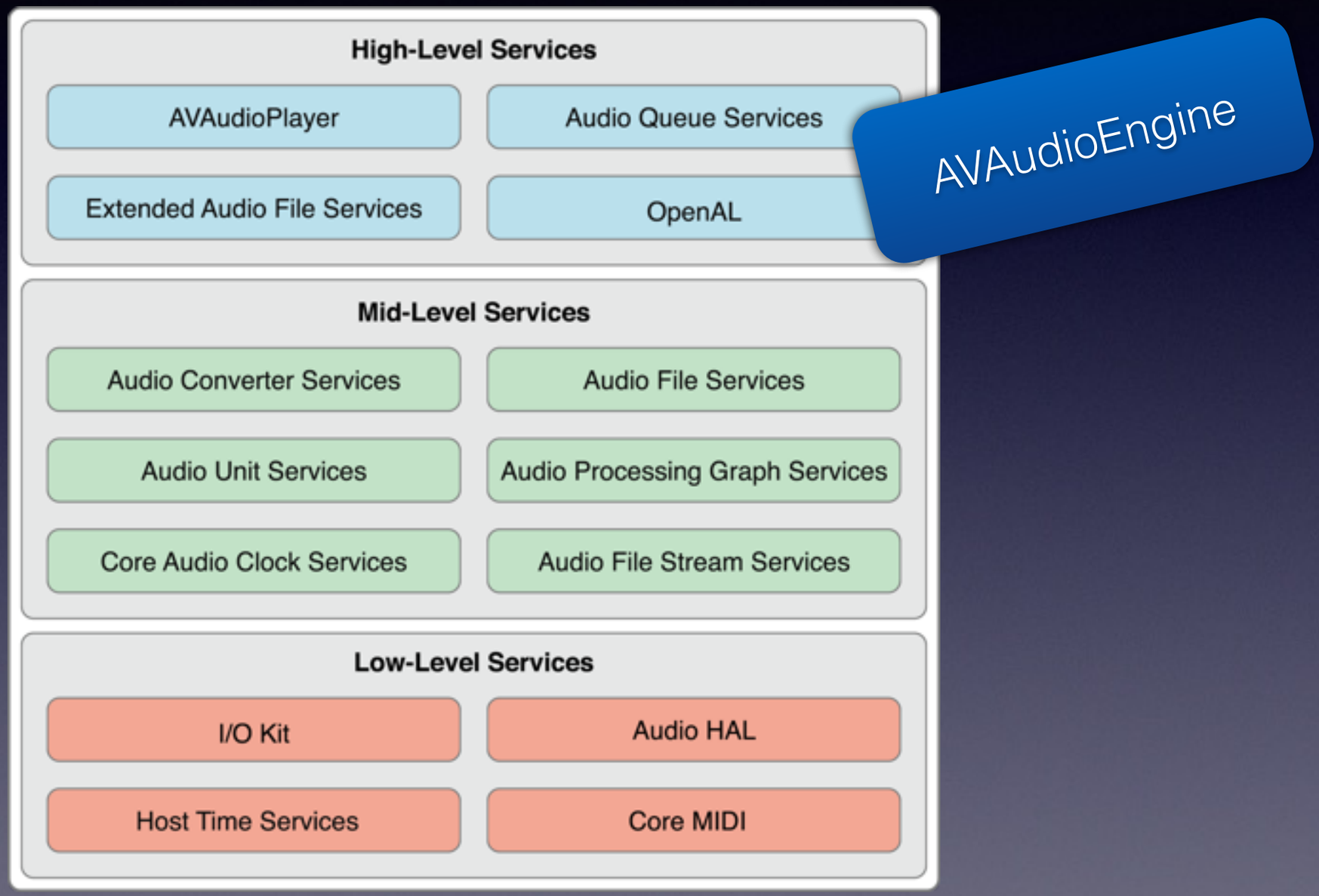
- Versatility
- Complexity
- Immediacy

Real-time audio: How?



- **Graph** of audio-processing nodes (Audio Units)
- The buffers are passed around through a **pull mechanism**.
- The pull mechanism works with **render callbacks**.

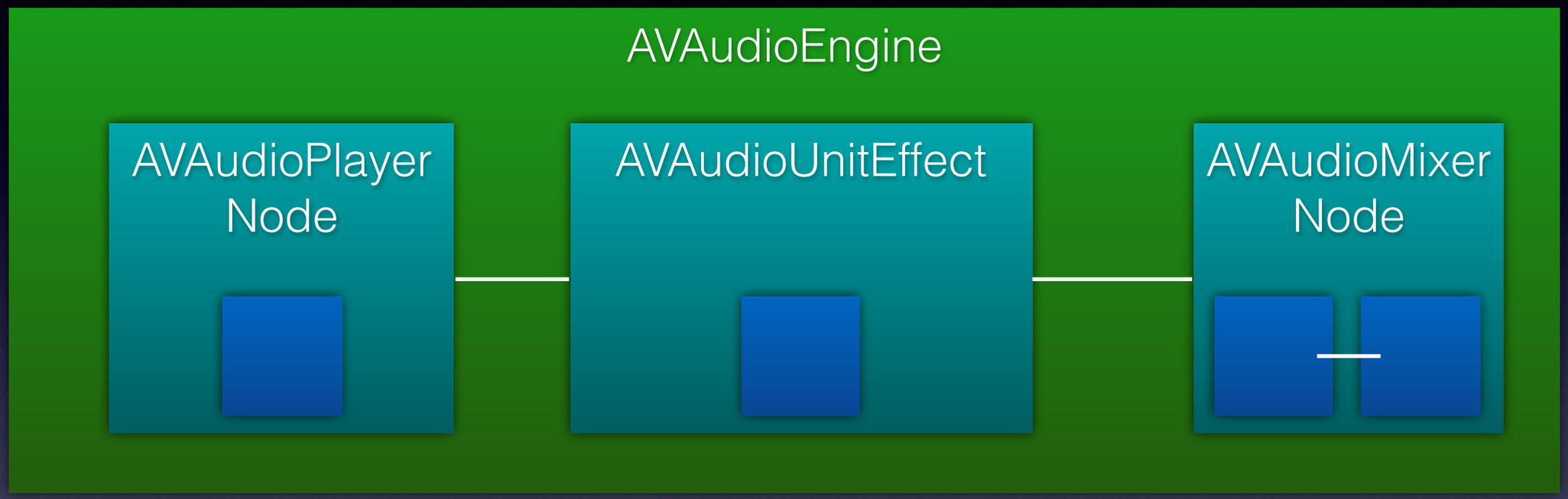
Real-time audio: Core Audio



From Apple's *Core Audio Overview*

AVAudioEngine

AVAudioEngine

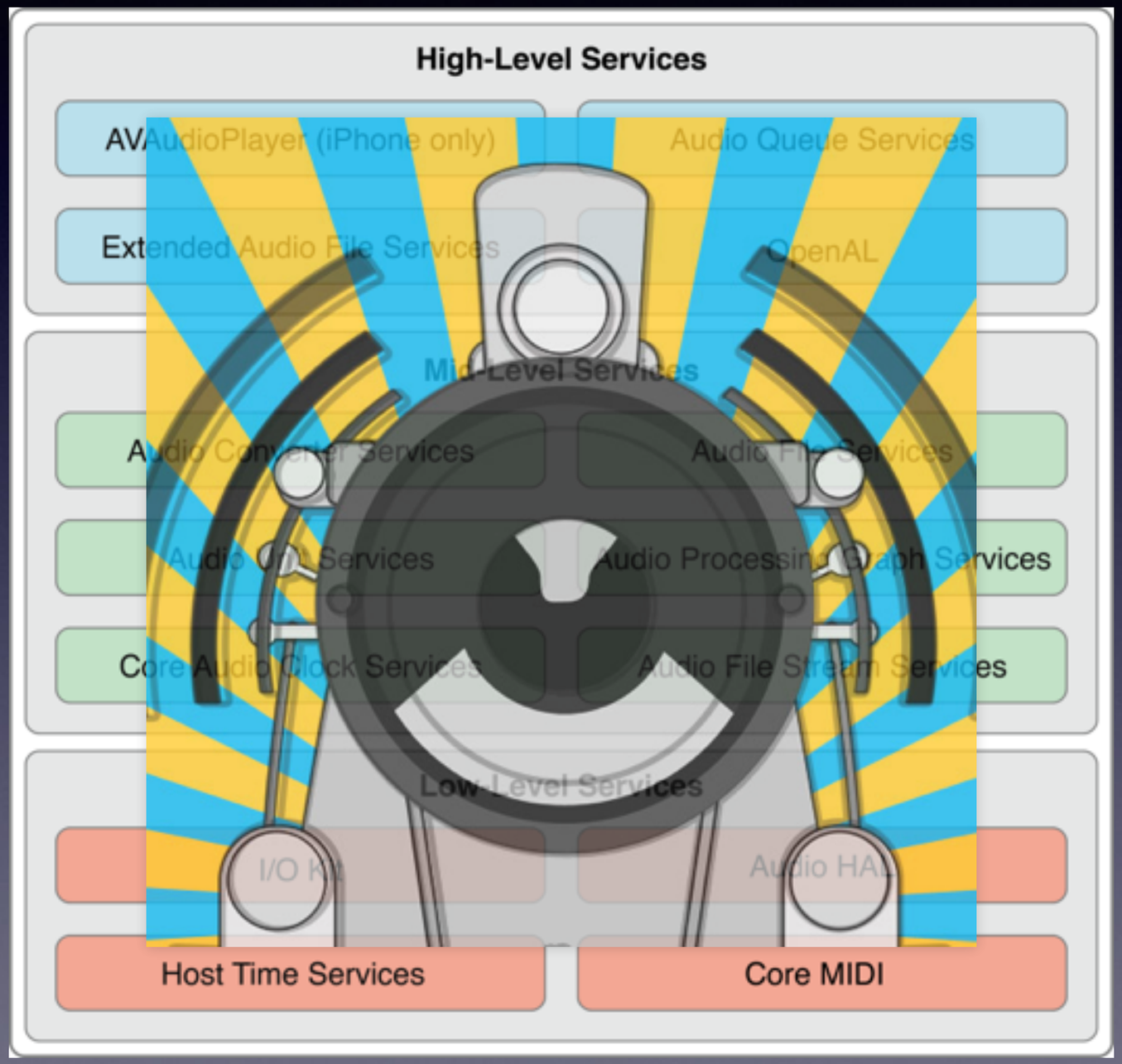


- Control over real time audio processing parameters
- Dynamically configure audio-processing nodes
- Input and output monitoring
- Much more (3D Mixing, recording, MIDI, …)

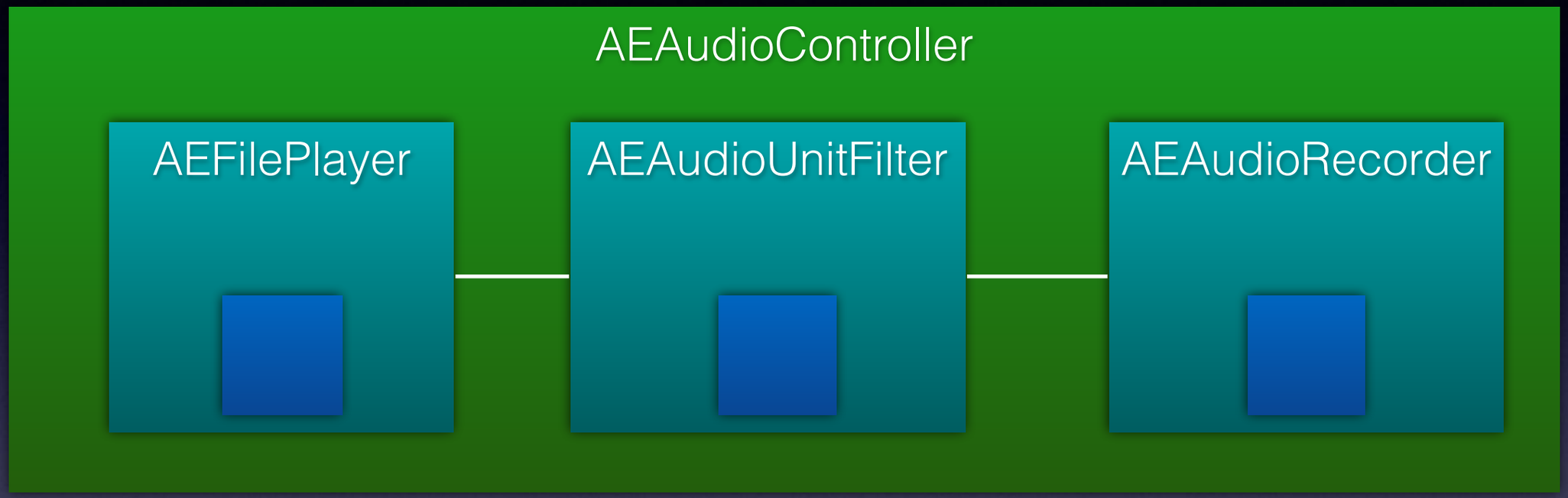
Demo

The Amazing Audio Engine

The Amazing Audio Engine



The Amazing Audio Engine



- *Lots* of functionality in common with AVAudioEngine
  - Control over real time audio processing parameter
  - Dynamically configure audio-processing nodes
  - Input and output monitoring
- Simple wrapper for AudioUnit
- Support for iOS 6.1+

Demo: The Amazing Sequencer

Best Practices

- Represent your Audio Engine as a class, not a struct.
- Let your app delegate manage the Audio Engine.
- Setup your AVAudioSession adequately.

Goodies

faturl.com/swiftyaudio

Thank You!

Questions?

Ariel Elkin
http://arielelkin.github.io
@AriVocals
arielelkin@gmail.com