Ios-gst-(audio)player

Swift wrapper for using GstPlayer on iOS

Stephan Hesse, Emliri
stephan@emliri.com

Using GstPlayer on iOS

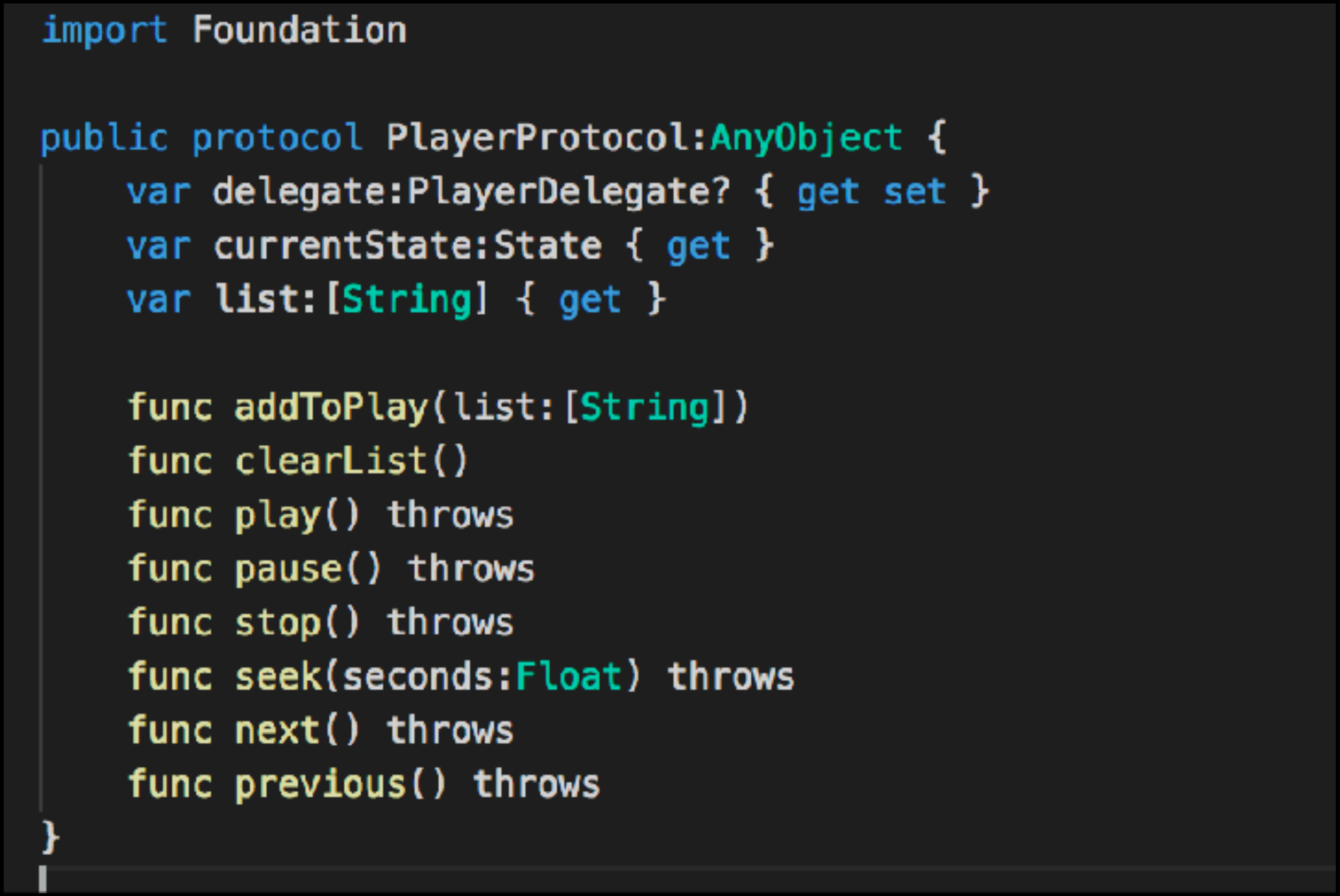
- Use-case: Play *mp4a* from plain HTTP (progressive download)
- Needed as a replacement for iOS AVPlayer
- Main motivation: gain more control of playback internals
- Display buffering progress to user
- Measure/optimize network transport issues

And also: May allow to use other codecs (OPUS), or use HLS/DASH on iOS using an implorentation with full control…

How to use GstPlayer from Swift?

- Swift: Nice modern language, built-in framework and a runtime for async
- Swift <-> C bridge can be built
- But not straightforward to pass callback data from Swift into Gstreamer and recover the context => a lot of “unsafe” calls and casts
- => Using Objective-C wrapper as intermediate layer between

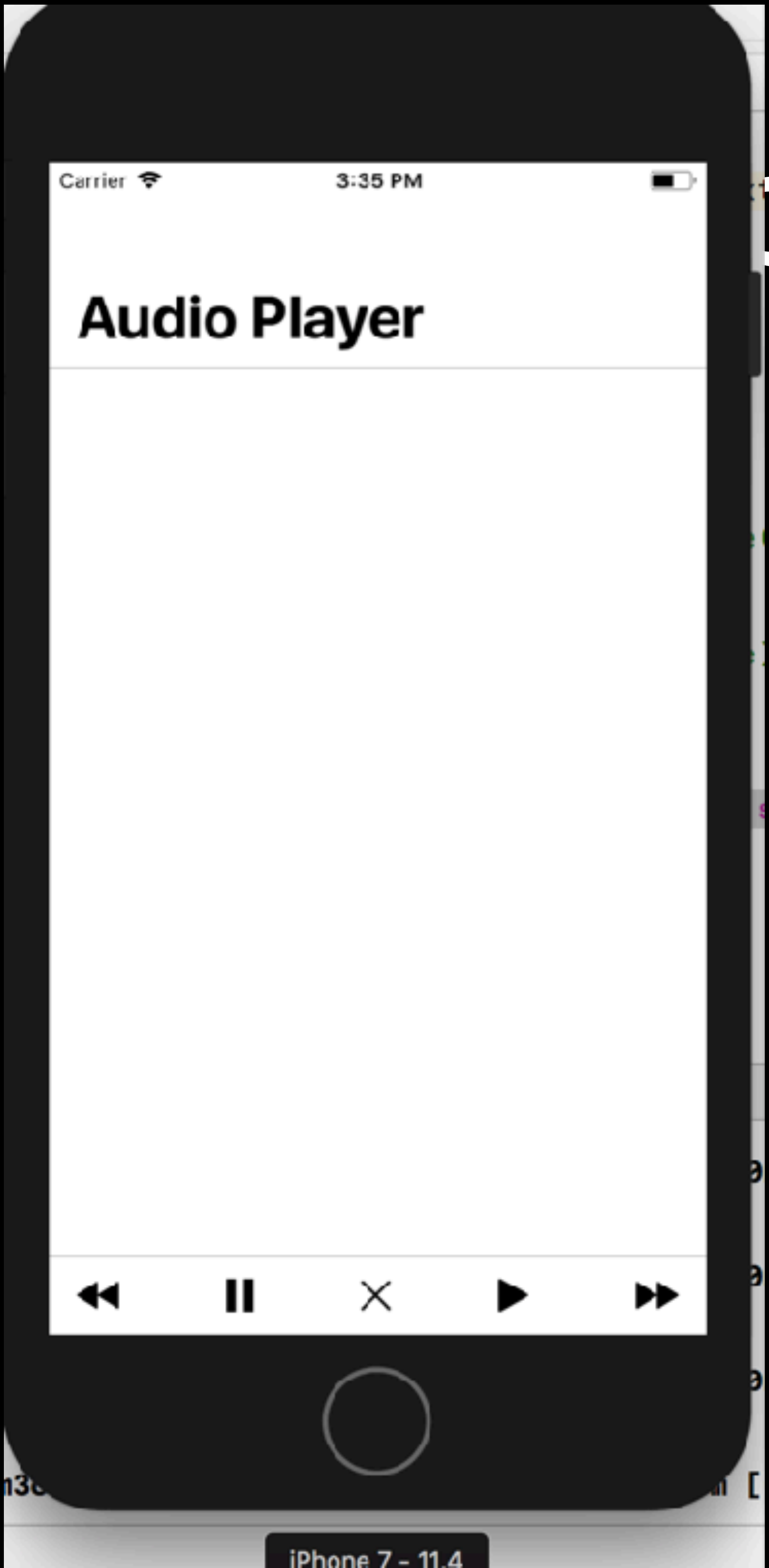
Swift player API



```
import Foundation

public protocol PlayerProtocol:AnyObject {
    var delegate:PlayerDelegate? { get set }
    var currentState:State { get }
    var list:[String] { get }

    func addToPlay(list:[String])
    func clearList()
    func play() throws
    func pause() throws
    func stop() throws
    func seek(seconds:Float) throws
    func next() throws
    func previous() throws
}
```


Carrier
3:35 PM
Audio Player
iPhone 7 - 11.4

Future

- https://github.com/emliri/ios-gst-audioplayer
- What’s next?
  - First create complete 1-to-1 GstPlayer API wrapper, then simplified abstractions may be done for certain use-cases
  - Use built-in (coming) play-queue functionality
  - Add video canvas to demo