

A Quick Overview of Everything Else

CS193W - Spring 2016 - Lecture 10

Agenda

1. [tvOS] TVML
2. [tvOS] Game Controllers
3. [watchOS] Heart rate monitoring

tvML

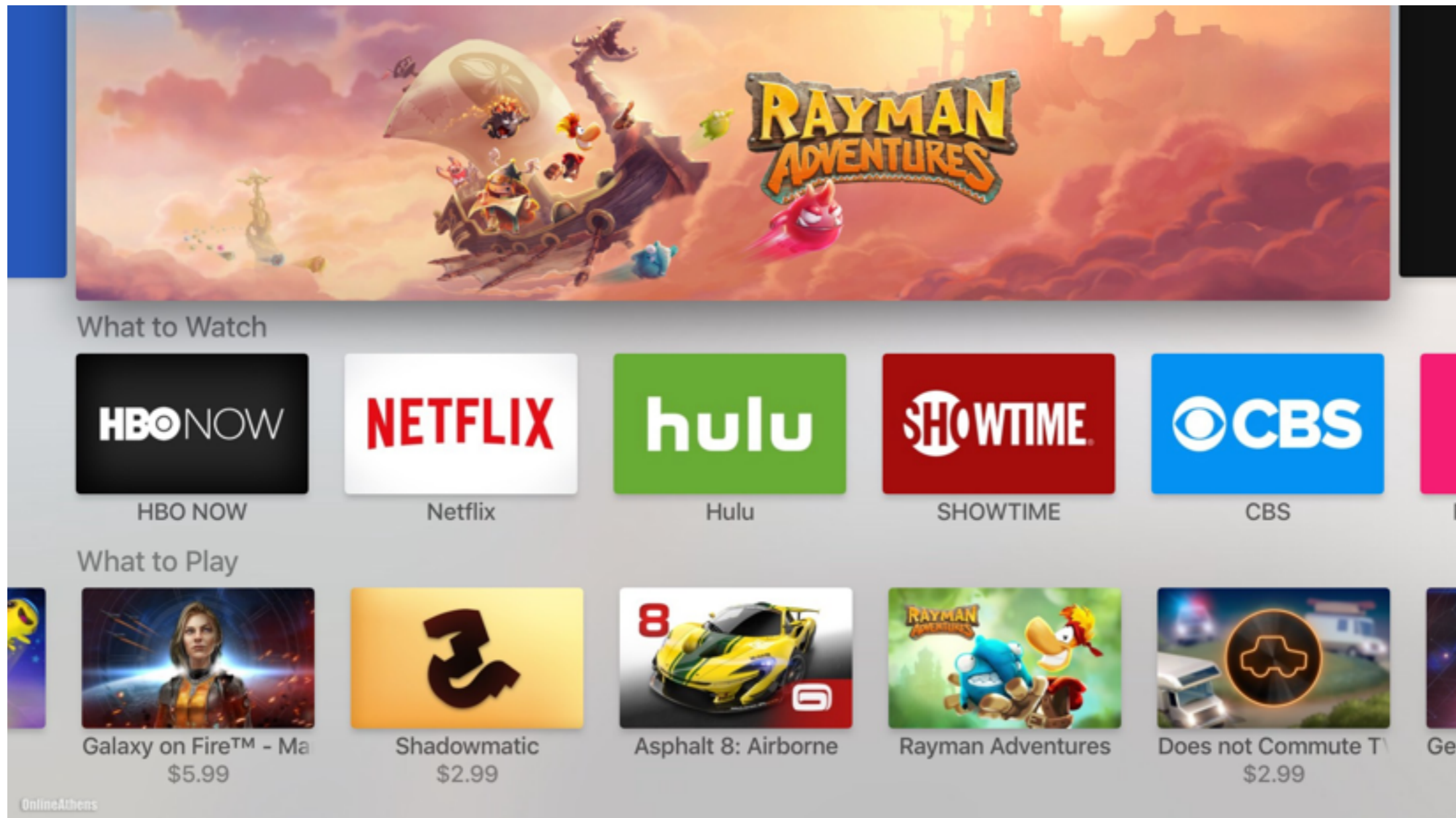
TVML

- TVML is an XML dialect that allows you to specify native UI
- You use Javascript together with TVML to create your app

TVML and Javascript files are Fetched from a Server

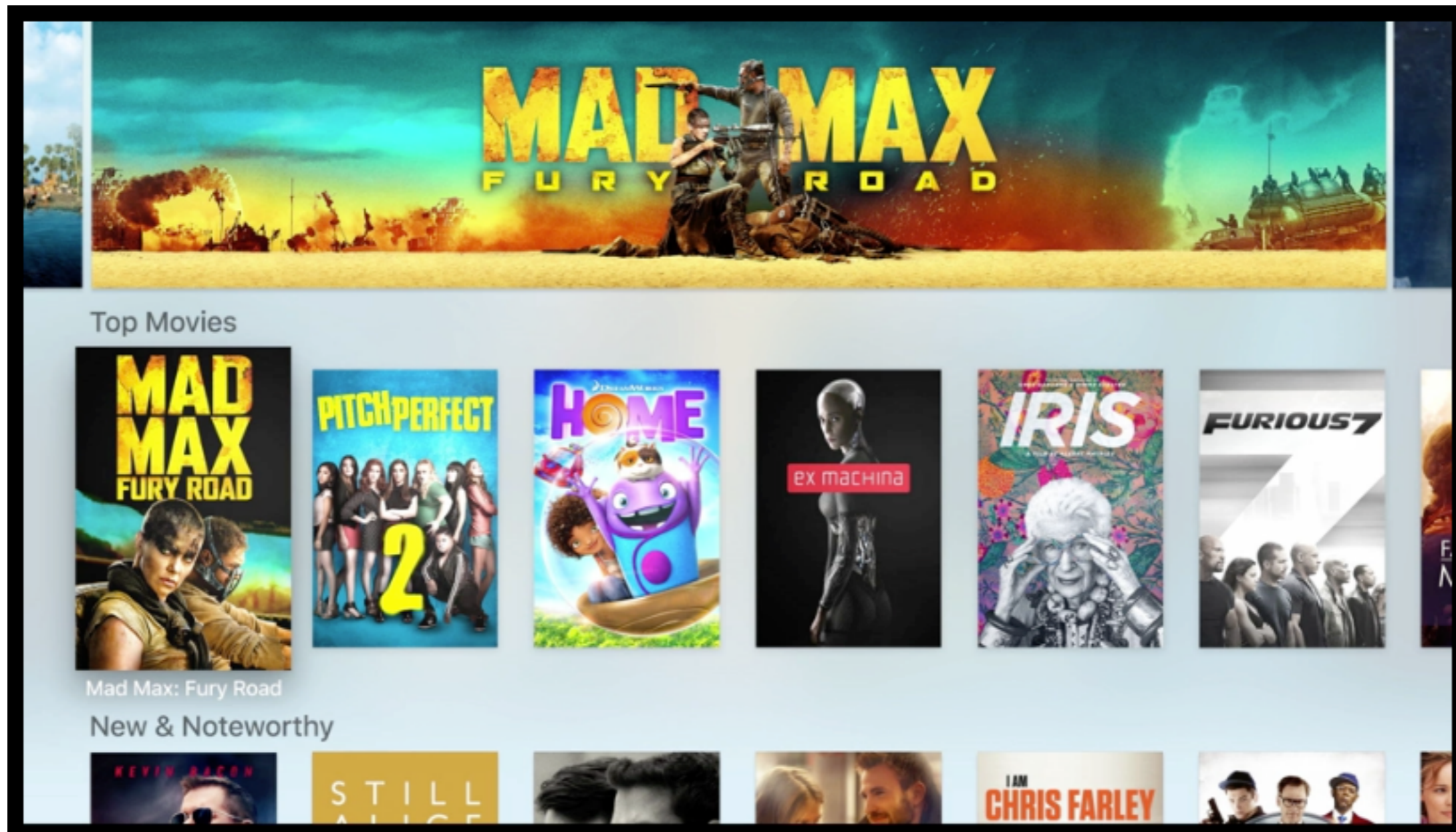
- This means that a TVML app can be very small (technically, just an App Delegate)
- The rest of the code, data and UI is fetched dynamically from the server

Example TVML Apps



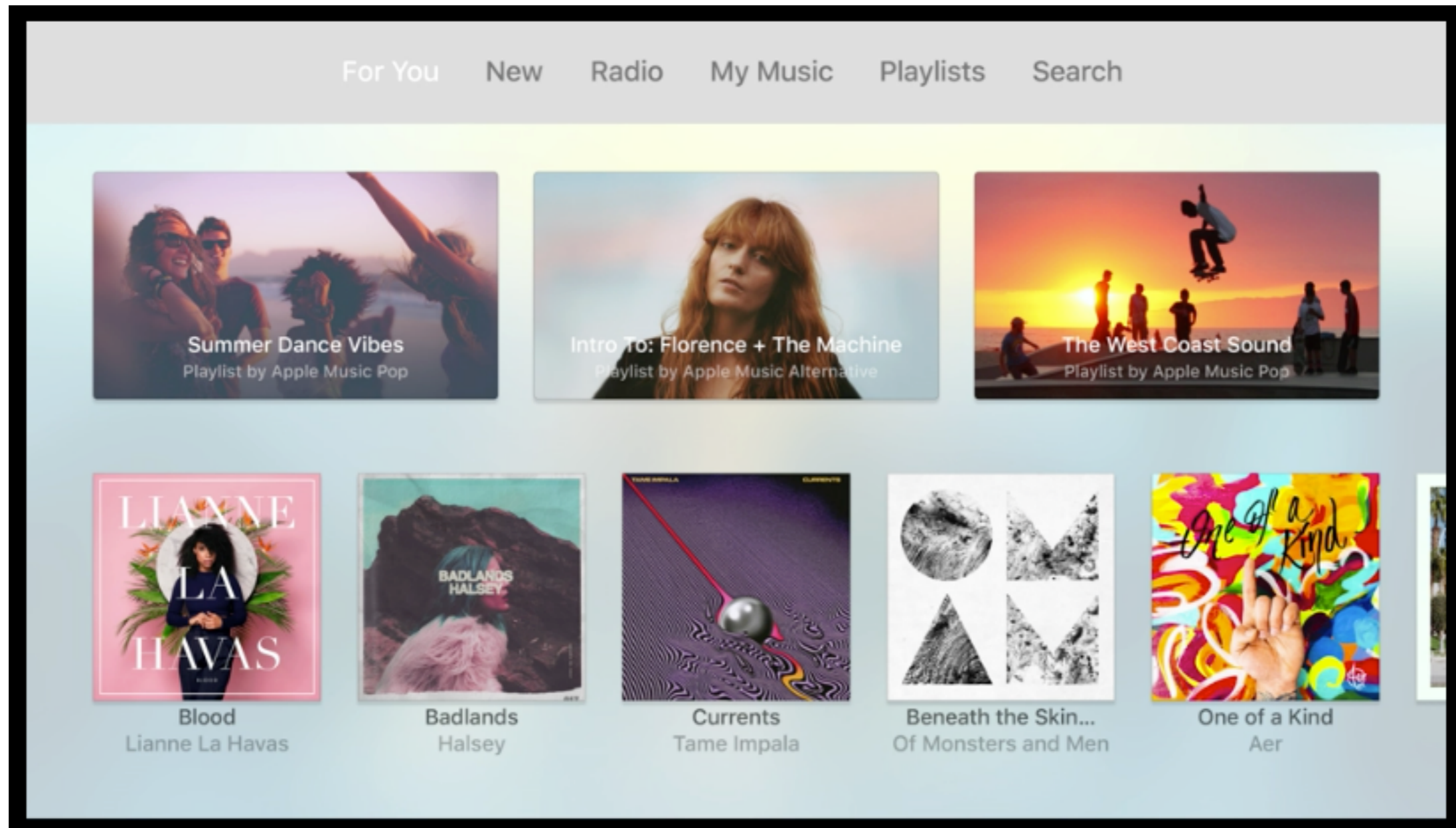
App Store

Example TVML Apps



iTunes Video Player

Example TVML Apps



Apple Music

Why Use TVML?

- Consistent, native look and feel
- Quick development time
- Server-based deployment means easy to make changes without going through App Store approval
- Many development shops have more expertise in Javascript than Swift / Objective C

Starting a Web Server for Development

cd to the root directory and then run:

```
python -m SimpleHTTPServer 9001
```

Which will run an HTTP server on port 9001.

Telling Your Application to use TVML

1. Create a `TVApplicationControllerContext`
2. Configure it with the URL to your Webserver and the URL of your main Javascript file.
3. Create a `TVApplicationController` using the context.
4. TVMLKit will then connect to the server and execute the main Javascript file.

Sample UIApplication Delegate Code

```
var window: UIWindow?
var appController: TVApplicationController?

static let tvBaseURL = "http://localhost:9001/"
static let tvBootURL = "\(AppDelegate.tvBaseURL)/js/application.js"

func application(application: UIApplication, didFinishLaunchingWithOptions launchOptions: [NSObject: AnyObject]?) -> Bool {
    window = UIWindow(frame: UIScreen.mainScreen().bounds)

    let appControllerContext = TVApplicationControllerContext()

    if let javascriptURL = NSURL(string: AppDelegate.tvBootURL) {
        appControllerContext.javascriptApplicationURL = javascriptURL
    }

    appControllerContext.launchOptions["BASEURL"] = AppDelegate.tvBaseURL

    if let launchOptions = launchOptions as? [String: AnyObject] {
        for (kind, value) in launchOptions {
            appControllerContext.launchOptions[kind] = value
        }
    }

    appController = TVApplicationController(context: appControllerContext, window: window, delegate: self)

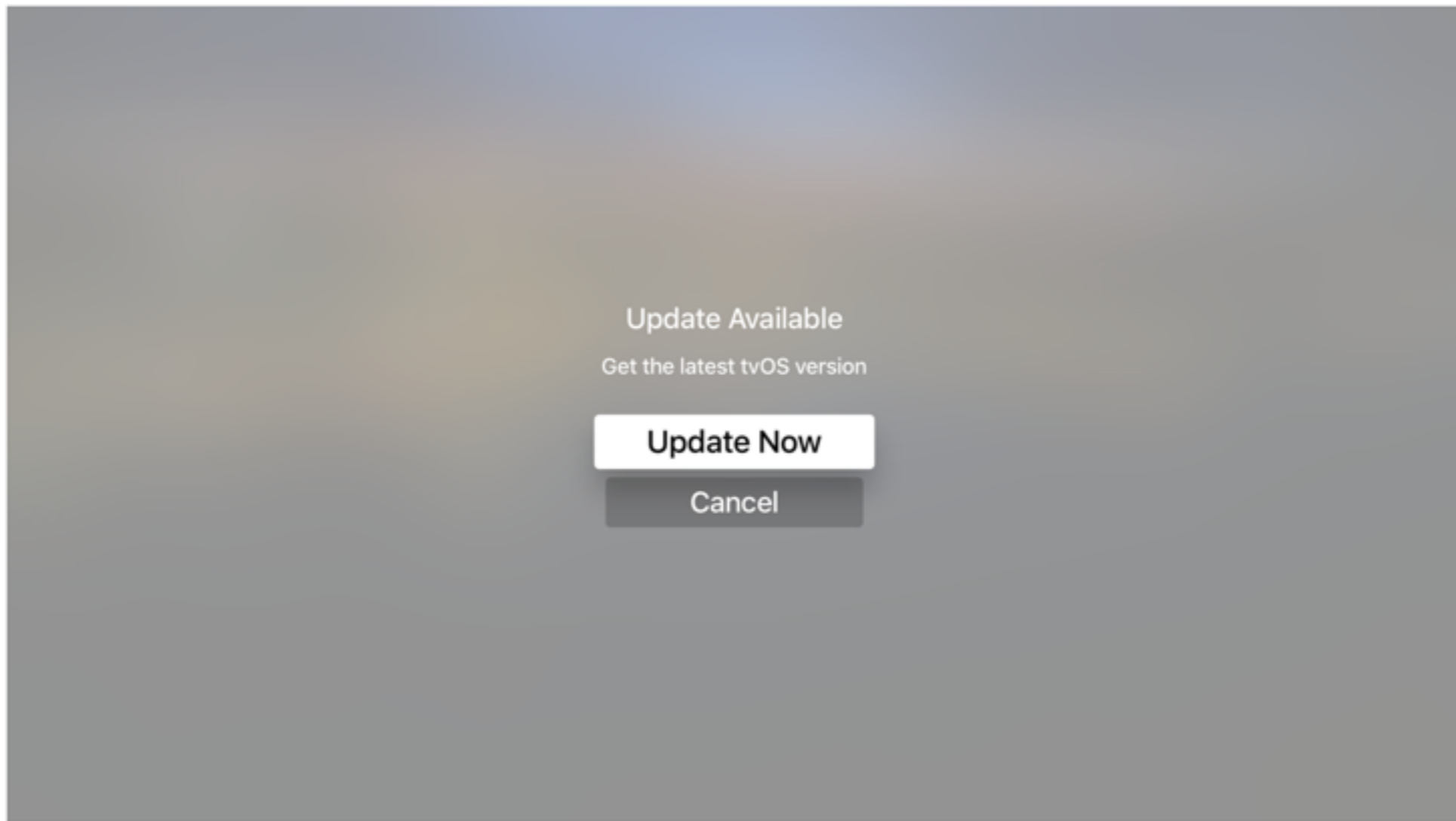
    return true
}
```

TVML Templates

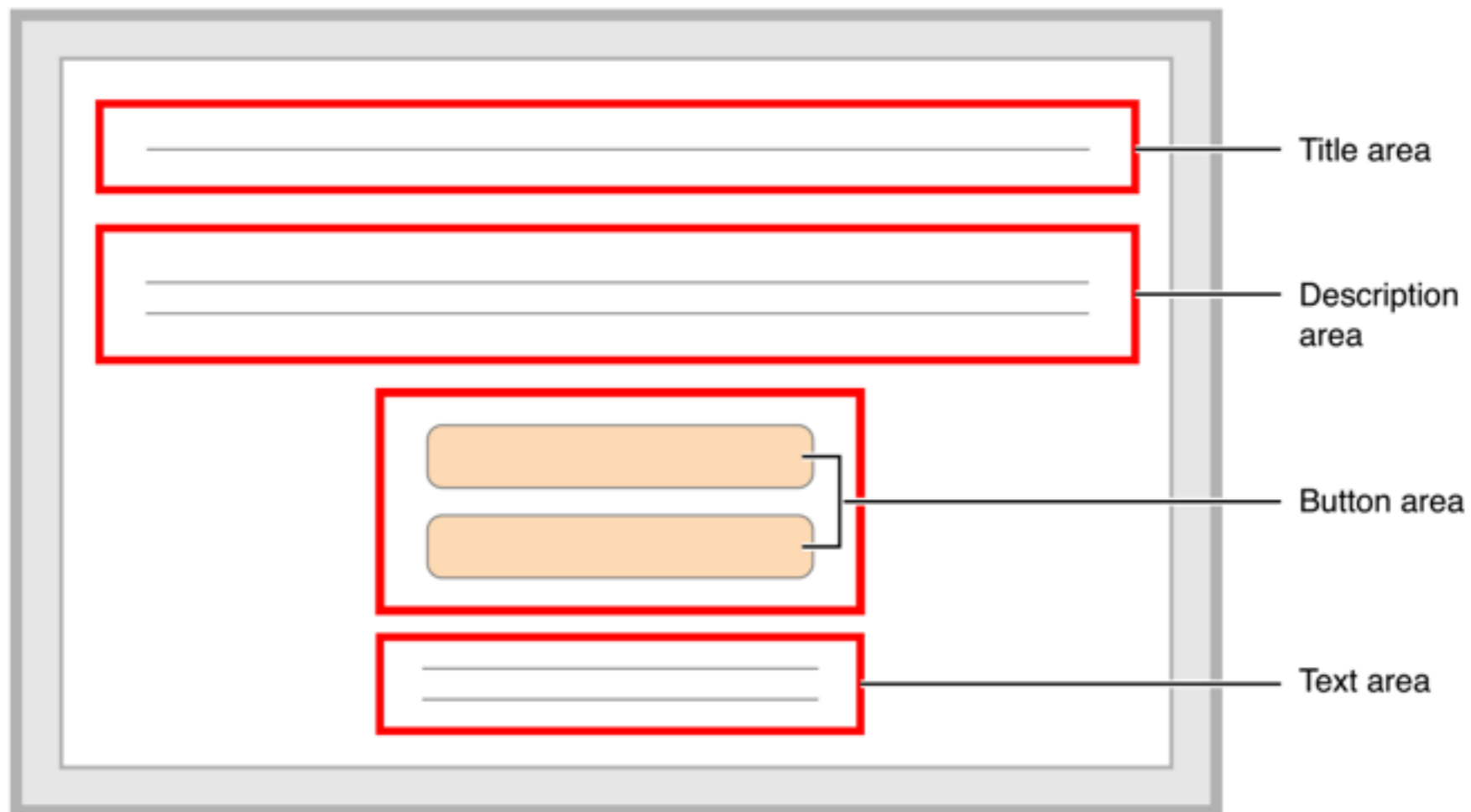
Apple provides many templates for you to start with. For example, here is the alert template:

```
<document>
  <alertTemplate>
    <title>Update Available</title>
    <description>Get the latest tvOS version</description>
    <button>
      <text>Update Now</text>
    </button>
    <button>
      <text>Cancel</text>
    </button>
  </alertTemplate>
</document>
```

What it Generates



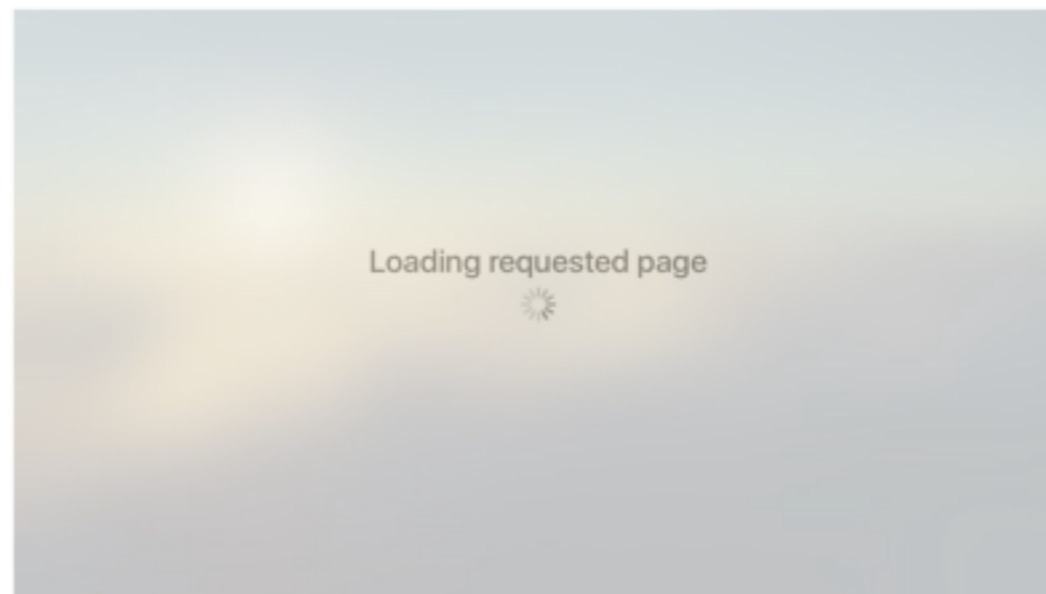
How the TVML corresponds to the generated UI



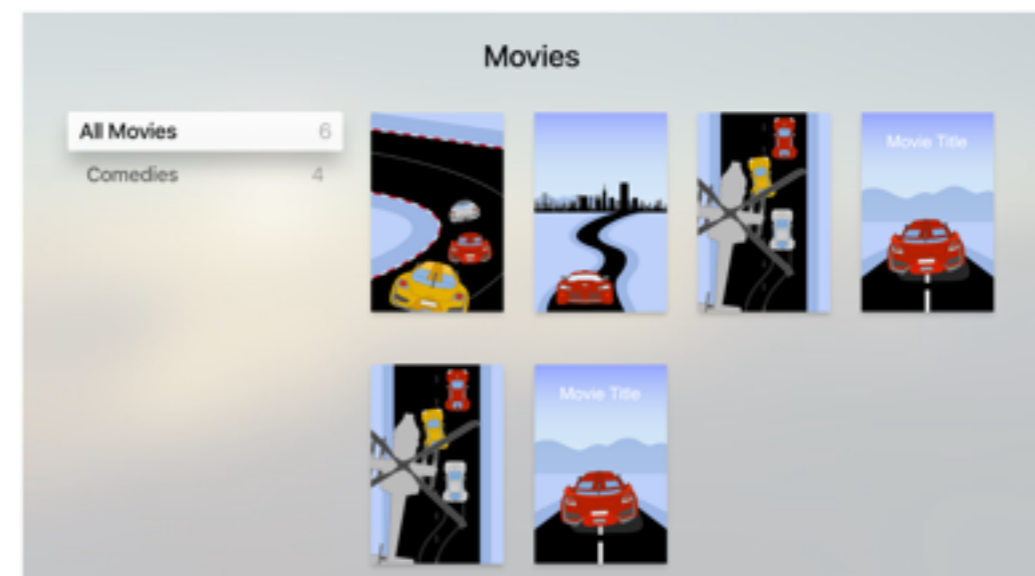
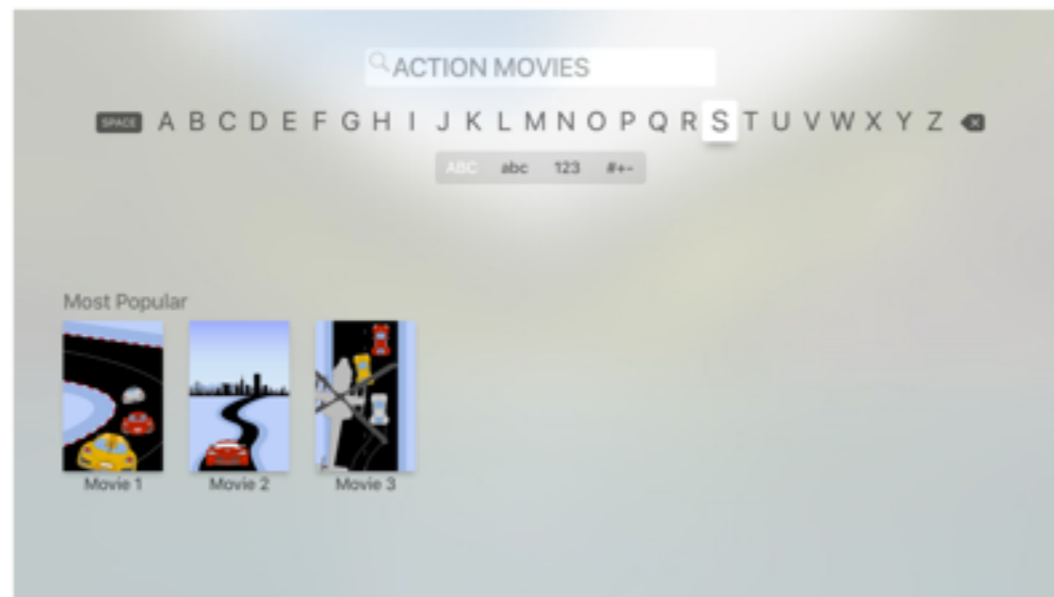
Another Simple Example

A loading template:

```
<document>  
  <loadingTemplate>  
    <activityIndicator>  
      <title>Loading requested page</title>  
    </activityIndicator>  
  </loadingTemplate>  
</document>
```



You can do most anything with TVML



Game Controllers

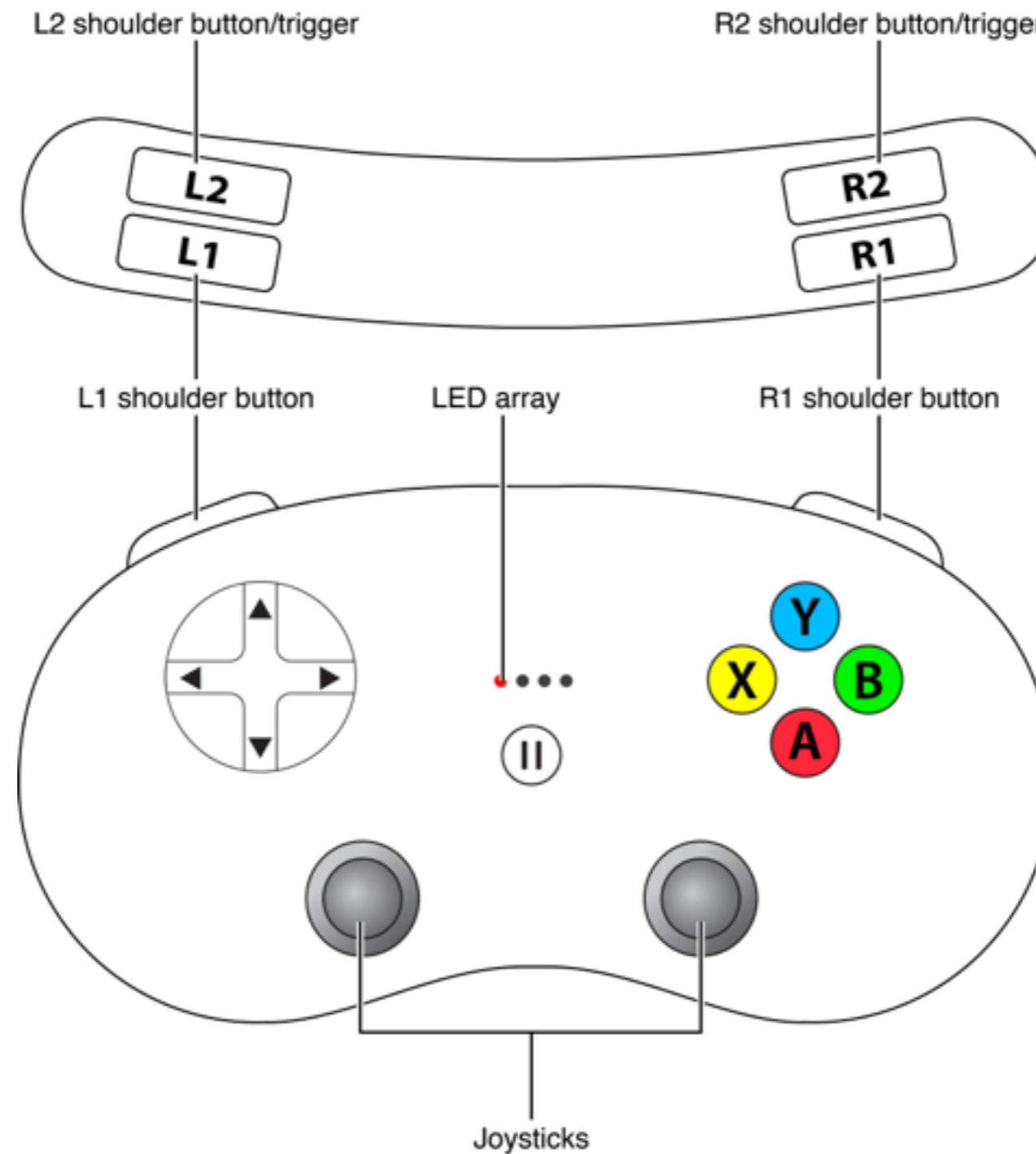
MFi Game Controllers

- Along with iOS 7, Apple introduced a specification for game controllers called “MFi”
- MFi stands for “Made For iPhone/iPad/iPod”

MFi Controllers

- Apple specifies that MFi controllers must conform to one of handful of predefined layouts
- Standard Layout - D-pad, A,B,X,Y buttons, left / right shoulder buttons, pause button, player indicator lights
- Extended Layout - All of the above, plus two triggers and two joysticks
- The Apple TV supports Extended Layout controllers but not Standard Layout controllers

Extended Game Controller



Example Extended Controller



The Siri Remote as a Game Controller

- The Apple TV requires that games also support the Apple TV remote as a controller
- The controllers on the Apple TV remote map to a subset of the controls on the Game Controller

Apple TV Remote / MFi Controller Mapping

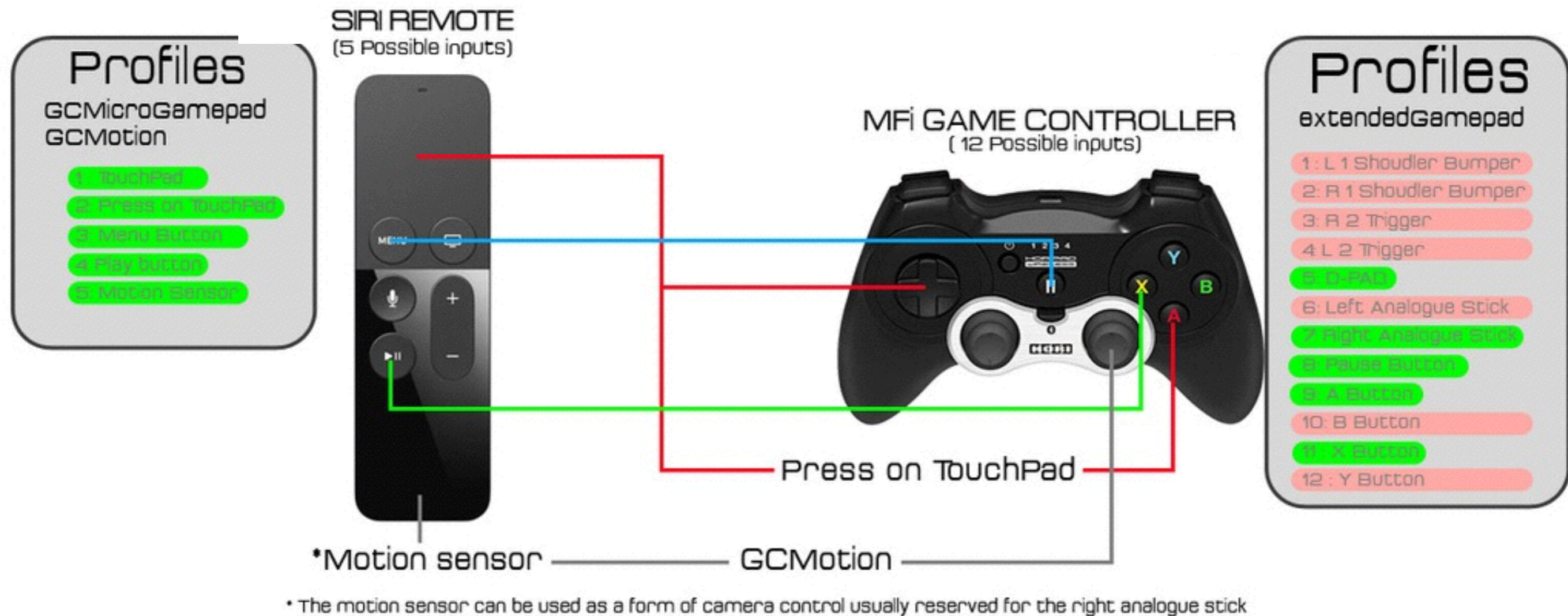


Image Credit: afterpad.com

Motion Control

- Optionally, Game Controllers can support motion control.
- The Siri Remote supports motion control.
- You can detect both the current acceleration placed by gravity and the acceleration caused by the user's motion

Supporting Multiple Controllers

- The Apple TV supports up to 2 simultaneous game controllers in addition to the the Siri Remote
- Your app must handle these controllers (and the Siri remote) connecting and disconnecting at any moment.

GCCController

- The class which represents a game controller
- Lets you see which controllers are currently connected and the state of their inputs

Getting Inputs from Game Controllers

- Three methods:
 - Polling
 - Registering for callbacks
 - Taking snapshots

Heart Rate Monitoring

HealthKit

- Heart rate information is available through HealthKit.
- Heart rate information is health information, which is *private*.
- It is more private than your contacts or your photos.

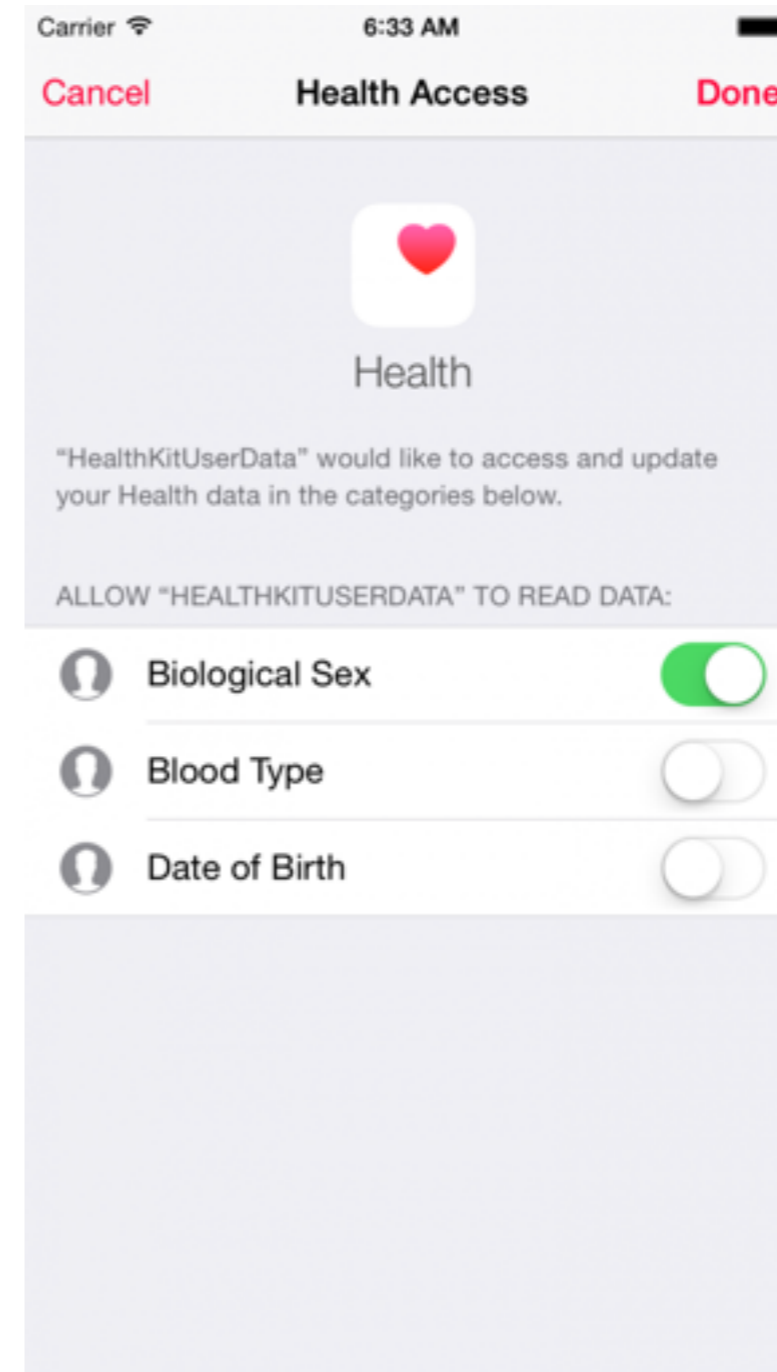
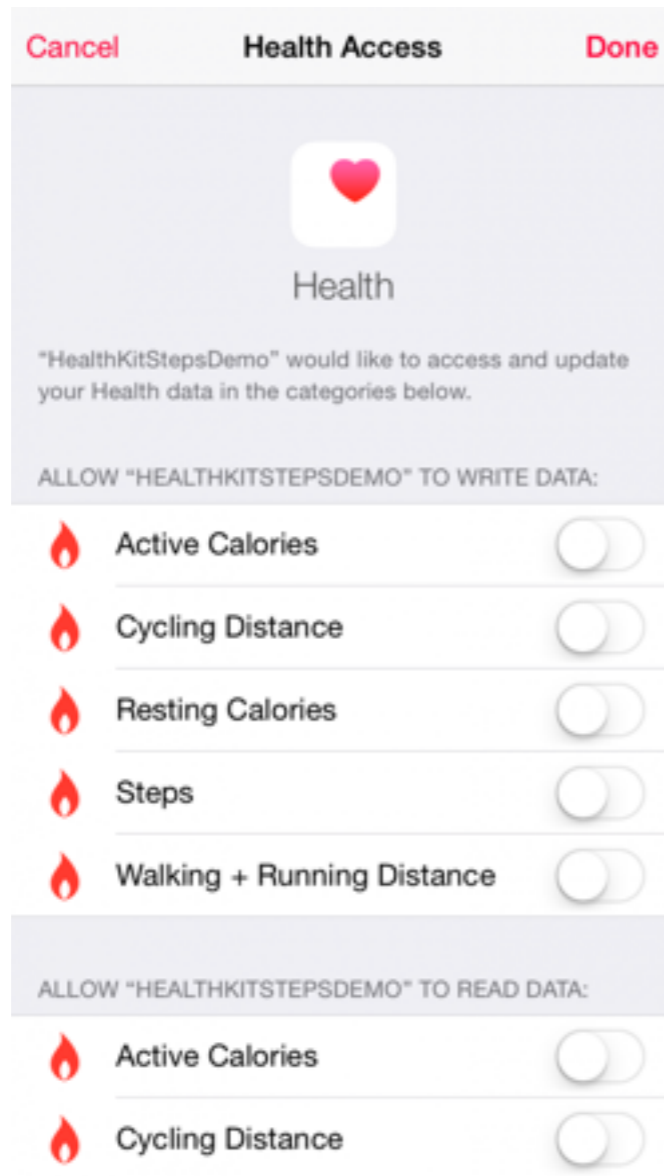
Privacy

- To access health information, you need to be a primarily a health-focused app
 - e.g. Birthdate, Skin Color, Gender are stored in HealthKit, but any old app just can't request them.
 - You need explicit privacy policy for Health info
 - You can't share Health information with third-parties unless they are also Health-focused
 - You can't use Health information to serve ads

Permissions

- You need to ask for permission to obtain Health information
- Permission is granted on the iPhone and is fine grained. The app specifies exactly what info it can accept from the user and the user can pick and choose what to share.
- If the user does not grant access to some info, the app doesn't even know. It just looks like unpopulated data. Privacy again.

Permissions Examples



The HealthKit Store (HKHealthStore)

- All data is stored locally, in the HealthKit Store
- An encrypted database
- Shared between all apps

Characteristic Data

- Properties of HKHealthStore
- Entered using the Health app.

`biologicalSex()`

`bloodType()`

`dateOfBirth()`

`fitzpatrickSkinType()`

Sample Data (HKSample)

- *Category samples.* Semantic data such as “in bed”, “asleep”
- *Quantity Samples.* Numerical data such as heart rate or steps taken.
- *Correlations.* Used to group together related samples, such as blood pressure (systolic and diastolic).
- *Workouts.* Represents a physical activity, can have associated sub-samples. Has properties like *type*, *duration*, *distance* and *energy*.

Using the Health Store

1. Collect samples and save them into the Health Store
2. Query the Health Store

Streaming Queries

- You can set up a streaming query to get notified when new data of a certain type is available.
- An *anchor point* is used to specify “give me data points that are new since this anchor point”

In Summary

- To monitor heart rate
 1. Get permission to the user's heart rate on the iPhone
 2. Start a workout
 3. Create a streaming query to get notified about new heart rate data
 4. End the workout

Thanks!

WESTERN CONFERENCE CHAMPIONS

San Francisco Chronicle

SFChronicle.com | Tuesday, May 31, 2016 | 0.50 *****

COMEBACK COMPLETED



Stephen Curry celebrates a three-pointer with the fans in Oracle Arena in the final minute of the Warriors' 96-88 win over the Oklahoma City Thunder.

Scott Strazzante / The Chronicle