

iOS

years **> 9**

countries **155**

apps **1,500,000**

tvOS

**<1** year

**100+** countries

**5,000** apps

# Introduction to tvOS

---



Gabriela Zagarova

[gaby.zagarova@gmail.com](mailto:gaby.zagarova@gmail.com)

[LinkedIn](#)



Siyana Slavova

[siyanaslavova@gmail.com](mailto:siyanaslavova@gmail.com)

[LinkedIn](#)

# Agenda

- Introduction
- Design
- Break
- TVMLKit
- Native tvOS Apps
- Exercises
- Break
- App distribution
- Questions



#MobConEU

# Comparison

	1st gen (2007)	2nd gen (2010)	3rd gen (2012)	4th gen (2015)
Processor	1 GHz Intel "Crofton" Pentium M	Apple A4 (ARM Cortex-A8)	Apple A5 (Single core ARM Cortex-A9, dual core with one core locked)	<b>Apple A8 (dual core)</b>
Memory	256 MB of 400 MHz DDR2 SDRAM	256 MB	512 MB	<b>2GB</b>
Storage	40 or 160 GB hard disk	8 GB NAND Flash for cache	8 GB NAND Flash for cache	32 or 64 GB NAND Flash
Connectivity	USB 2.0, infrared receiver, HDMI, component video,optical audio	Bluetooth, Micro-USB, HDMI, infrared receiver, optical audio	Bluetooth, Micro-USB, HDMI, infrared receiver, optical audio	Bluetooth, USB-C, HDMI, infrared receiver
OS	Apple TV Software 1.0	Apple TV Software 4.0	Apple TV Software 5.0	<b>tvOS 9.0</b>
Apps	No App store	No App store	No App store	<b>App store</b>

# What's the difference?

- New Hardware
  - 32GB or 64GB storage
  - 64-bit A8 chip processor
  - 2 GB of RAM
  - USB-C
  - HDMI 1.4
  - 1080p resolution
  - Bluetooth 4.0

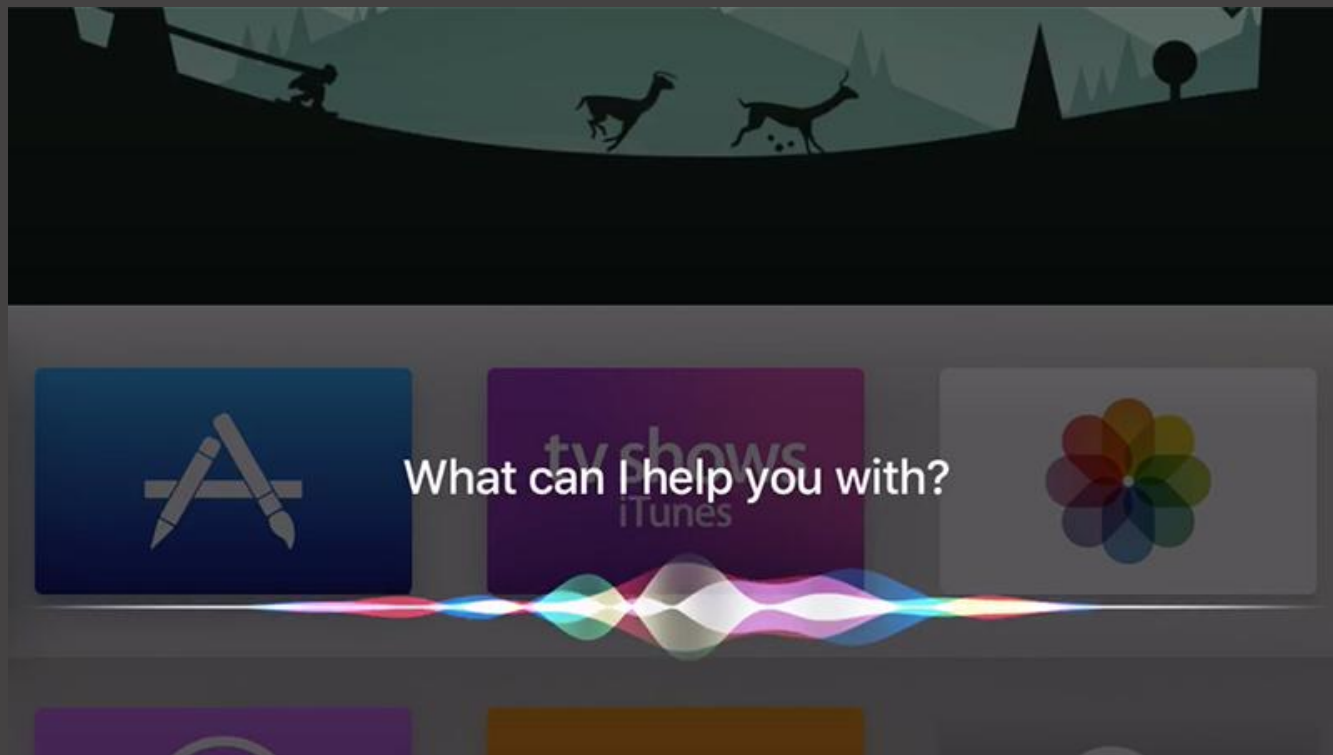


- New Remote

- Accelerometer
- Gyroscope
- Landscape mode
- Dual microphones for Siri
- Glass Touch surface
- Lightning connector for charging
- Buttons
  - Menu
  - Home
  - Siri
  - Play/Pause
  - Volume up/down



# Ask Siri



# Latest News



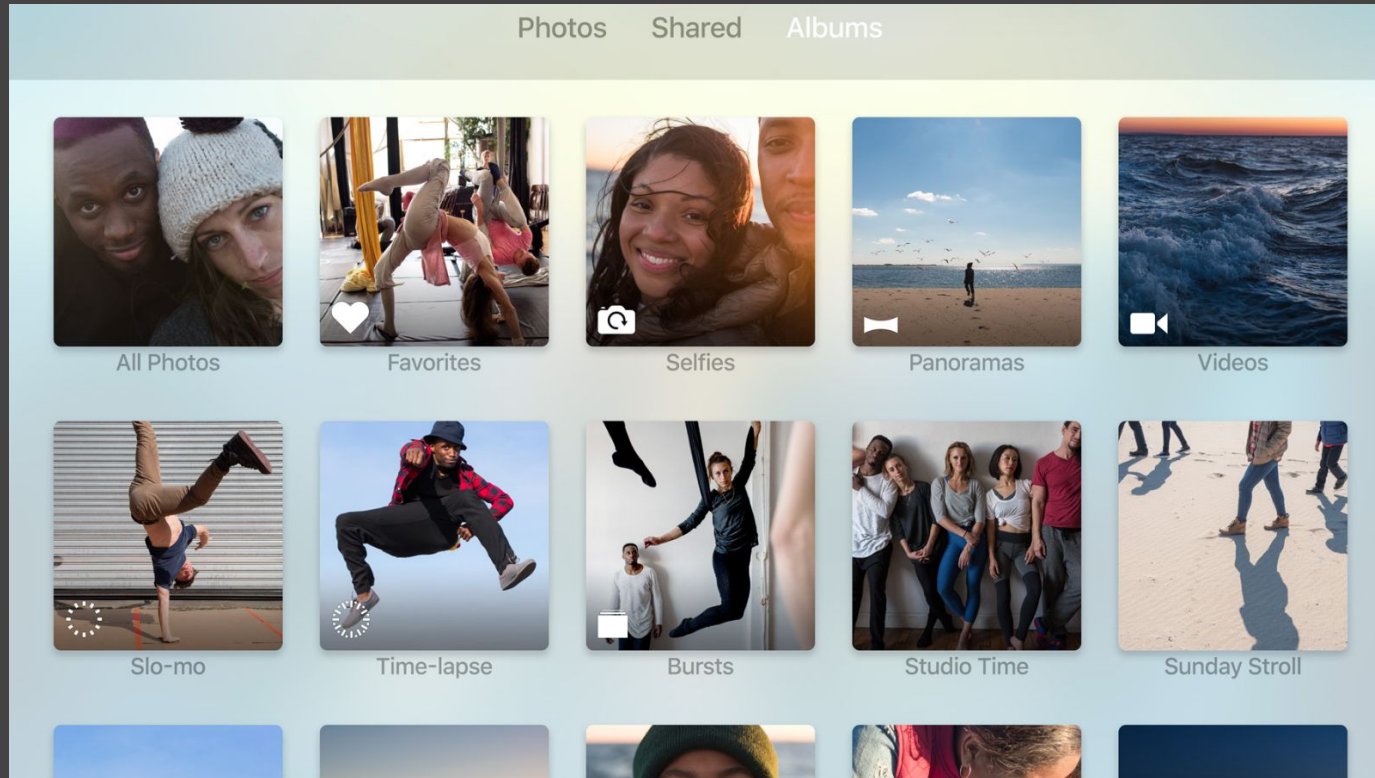


# What's new in tvOS 9.2?

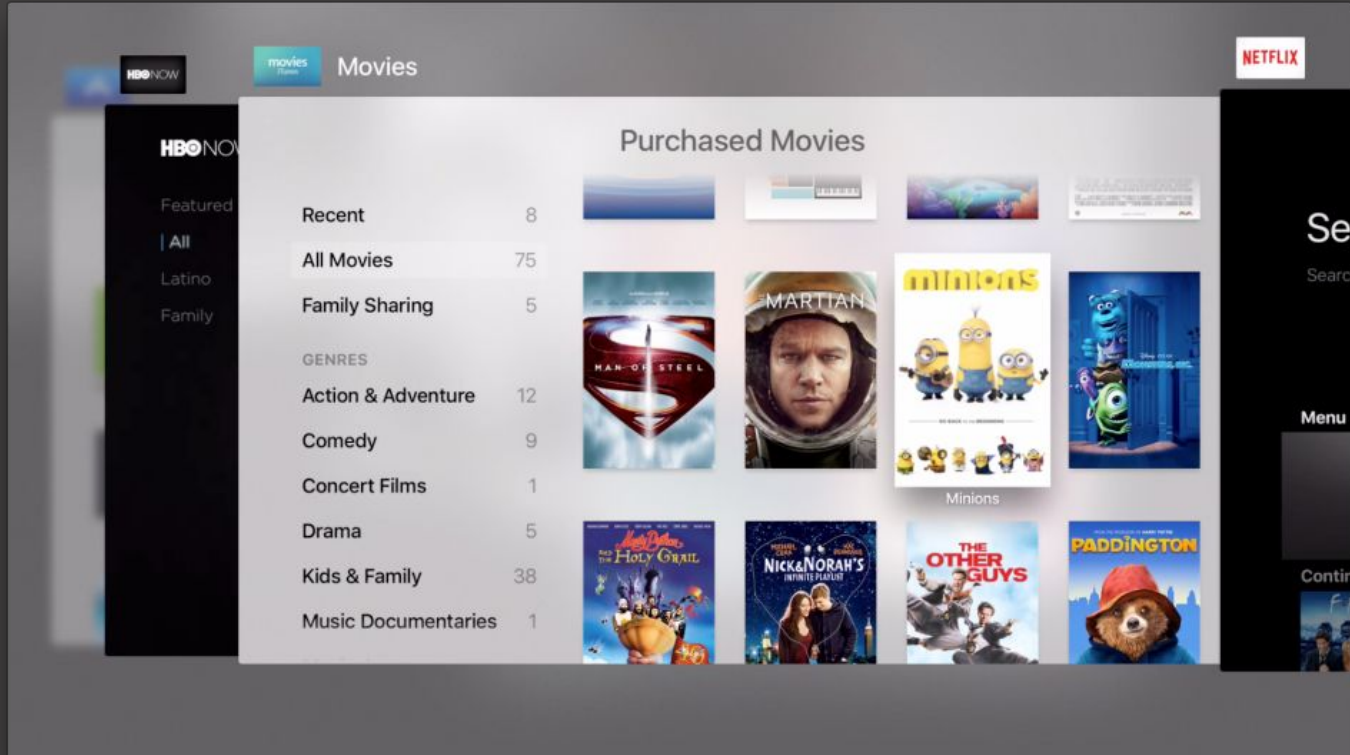
- Siri Remote dictation
- Siri for App Store Content
- Bluetooth keyboard support



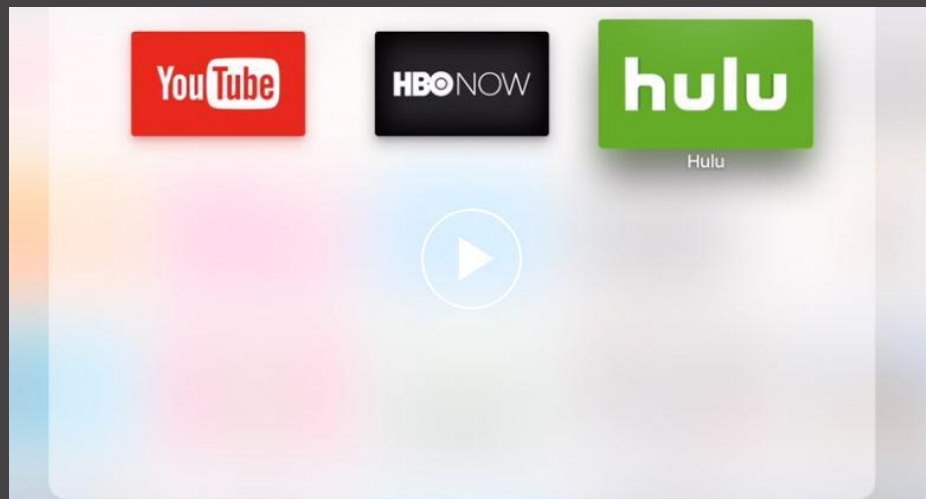
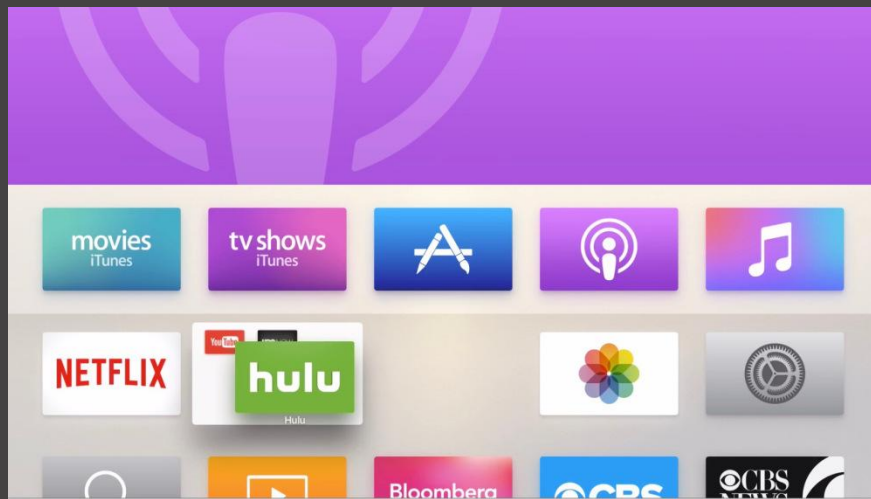
# iCloud Photo Library & Live Photos



# App Switcher UI



# App Folders





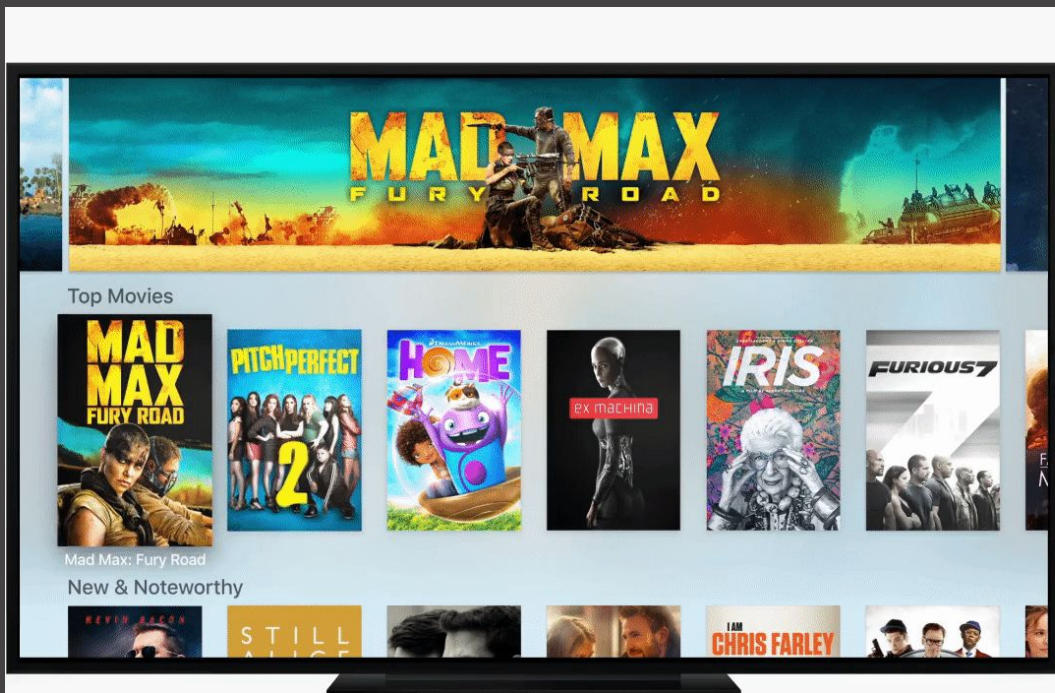
# Design For Apple TV

# Design Principles

Connected

Clear

Immersive



# New Gestures

Swipe



Tap



Click



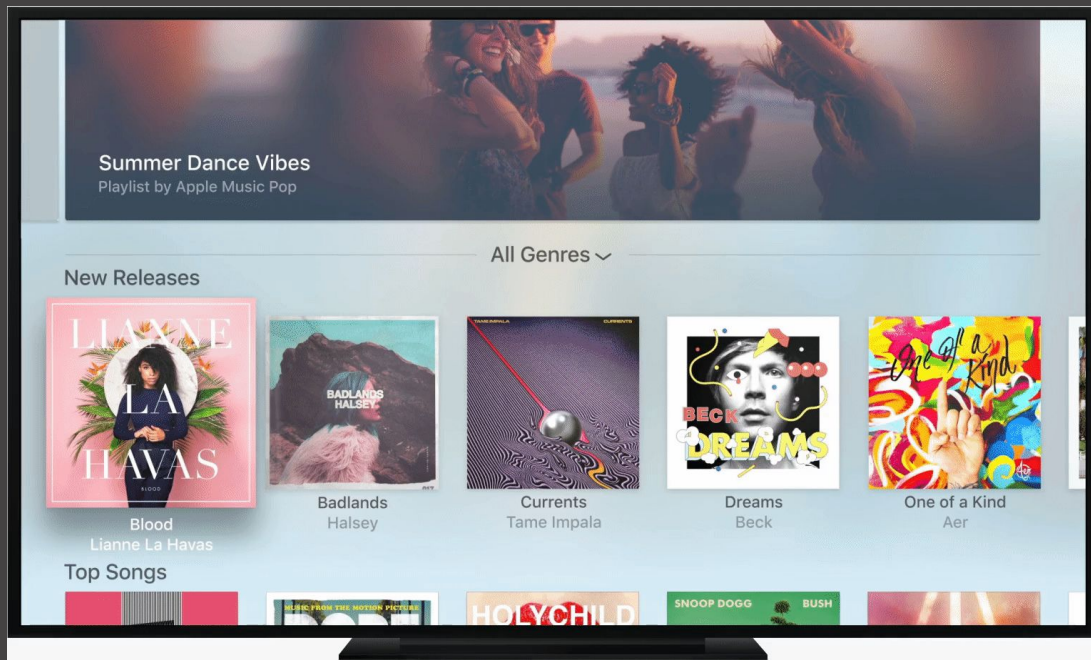


# Design Principles

Connected

Clear

Immersive





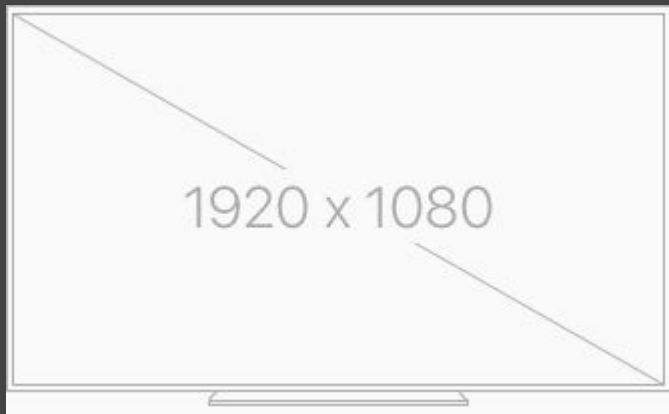
# Design Principles

Connected

Clear

**Immersive**

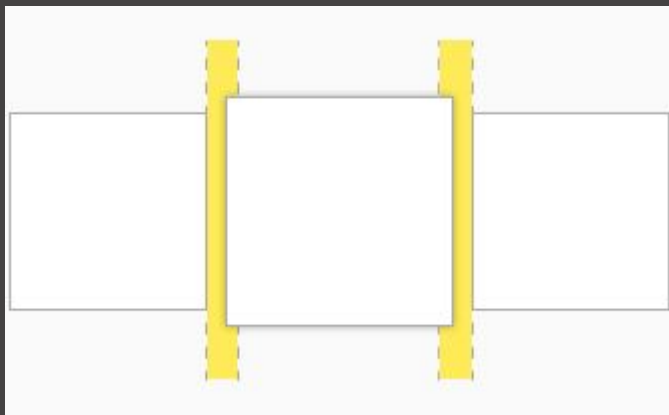




- Optimize assets



- Away from the edges



- Padding



- Partially displaying

# Best practices

- Consider loading experience
  - Use loading indicator and placeholders
- Authentication
  - Delay authentication as long as possible and collect minimal information
- Onboarding
  - Resist the temptation to explain every feature
  - Teach gradually and in context
- **Avoid using cursor!**



# Parallax

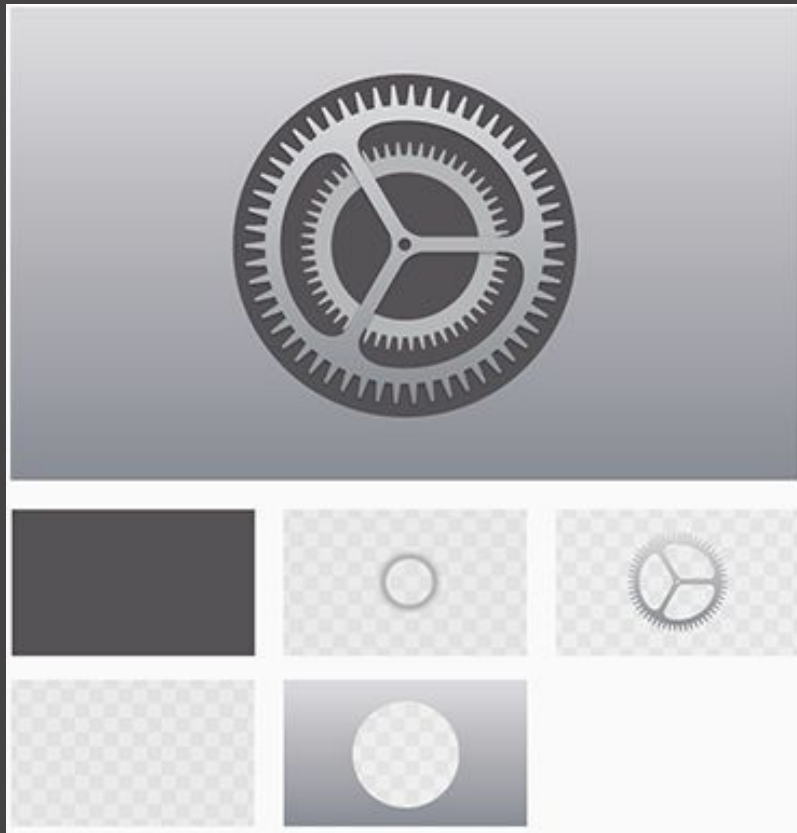
# Parallax effect

- What is parallax?
  - Layers
  - .LSR
- How to create?
  - Parallax Explorer - Requires OS X 10.9+, Photoshop CC 2014, 8 GB RAM
  - Parallax Preview app - Requires OS X Yosemite 10.10.5+
  - Xcode



# App Icon

- A small and a large icon
  - 400x240px for Apple TV Home Screen
  - 1280x768px for App Store
- Layering
- Effects
- Sizes



# Let's try it

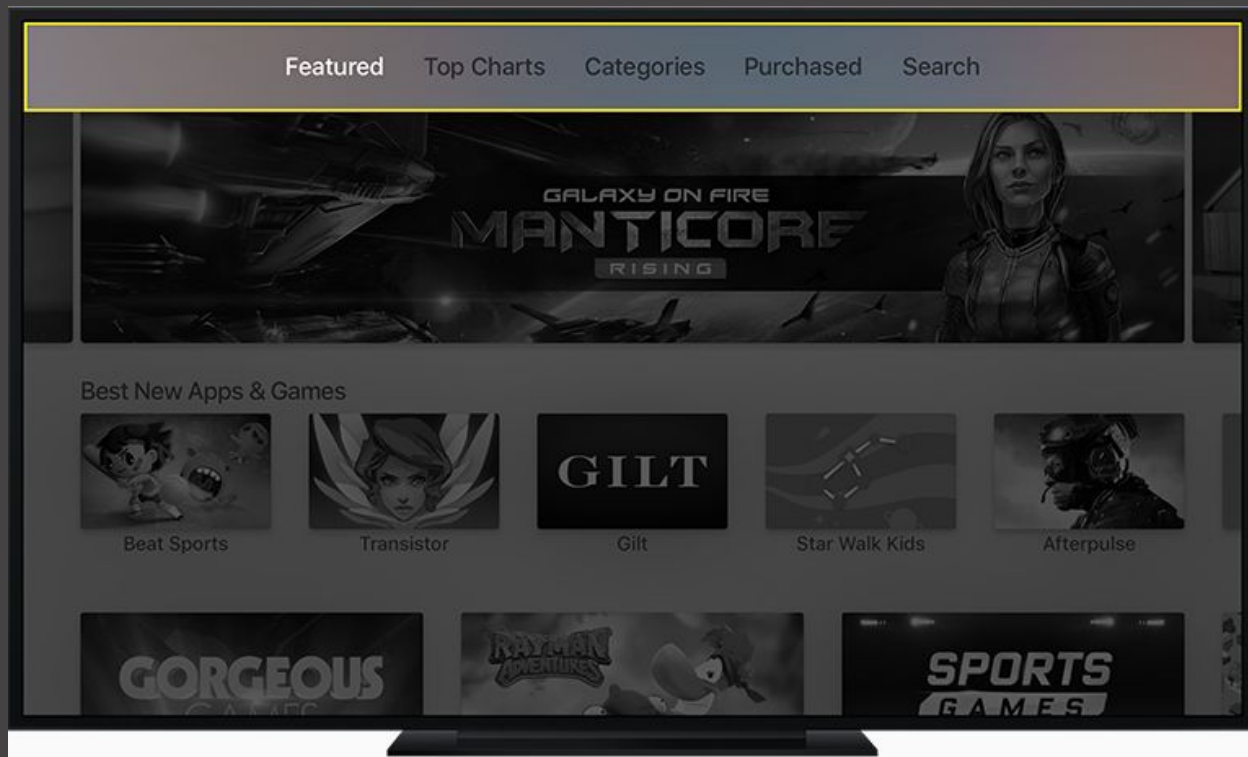
Download resources  
~10 min



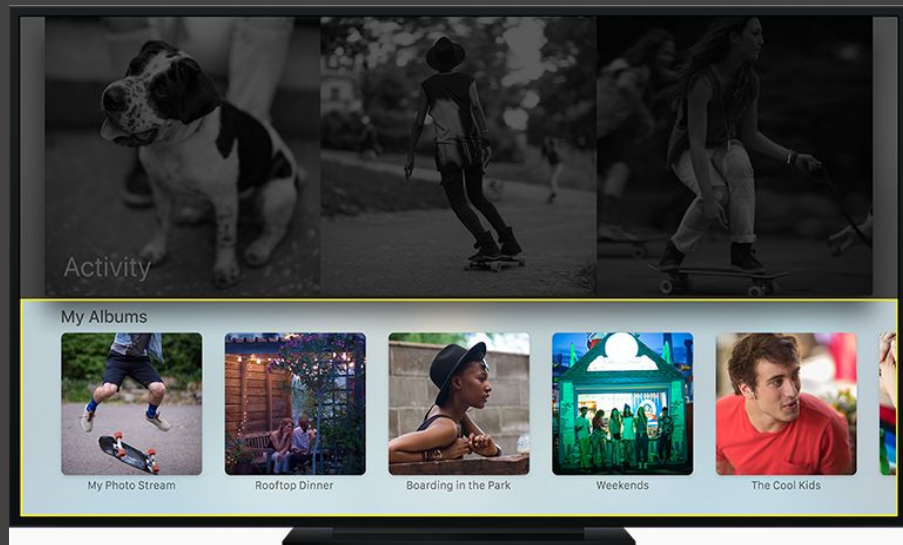
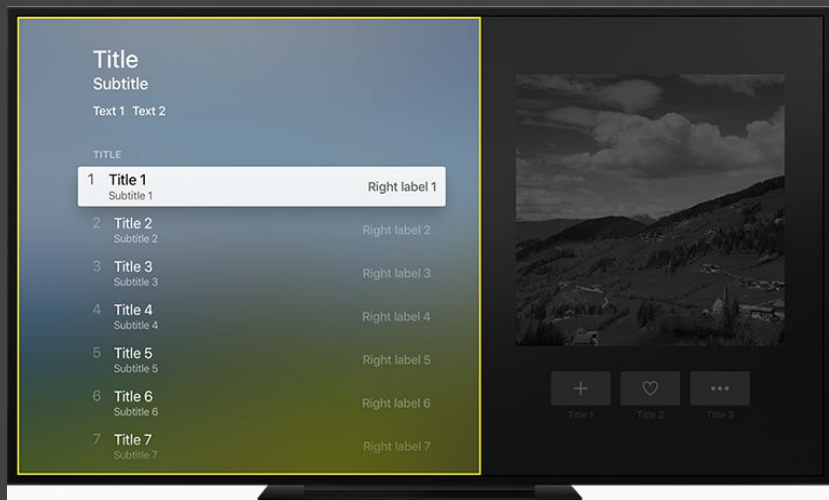
# User Interface Elements



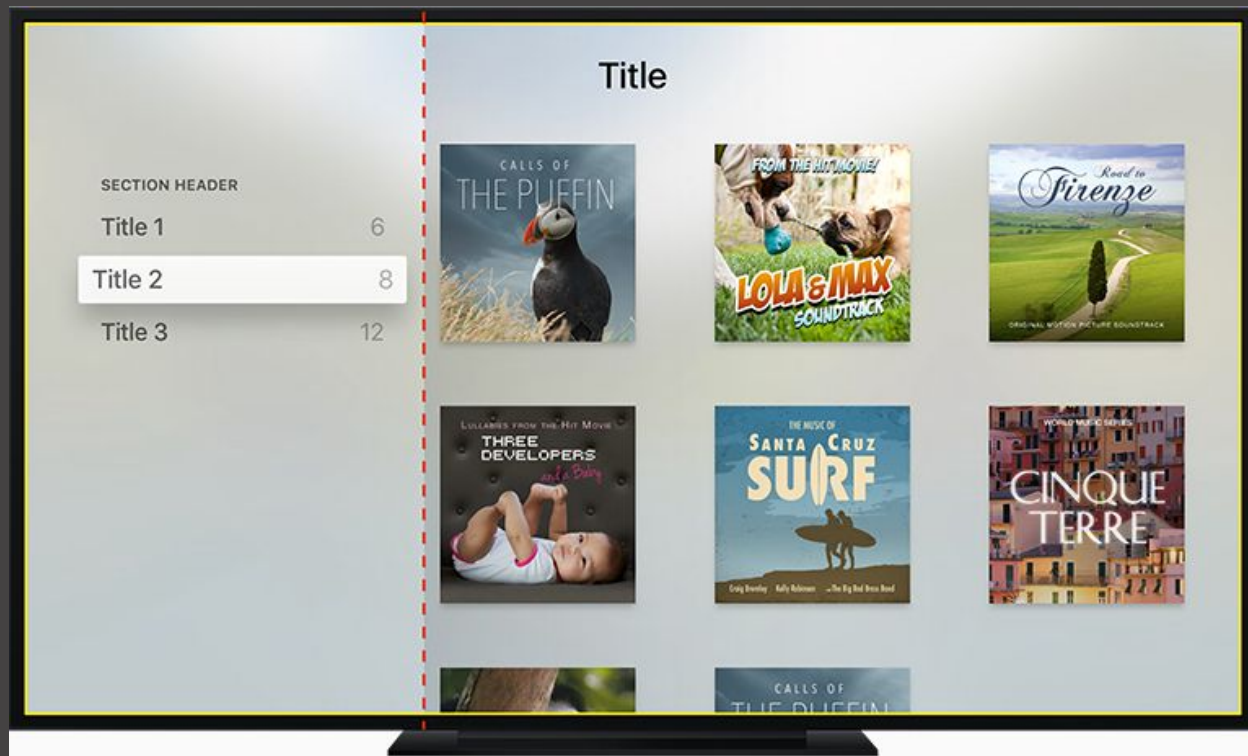
# Tab Bars



# Tables and Collections



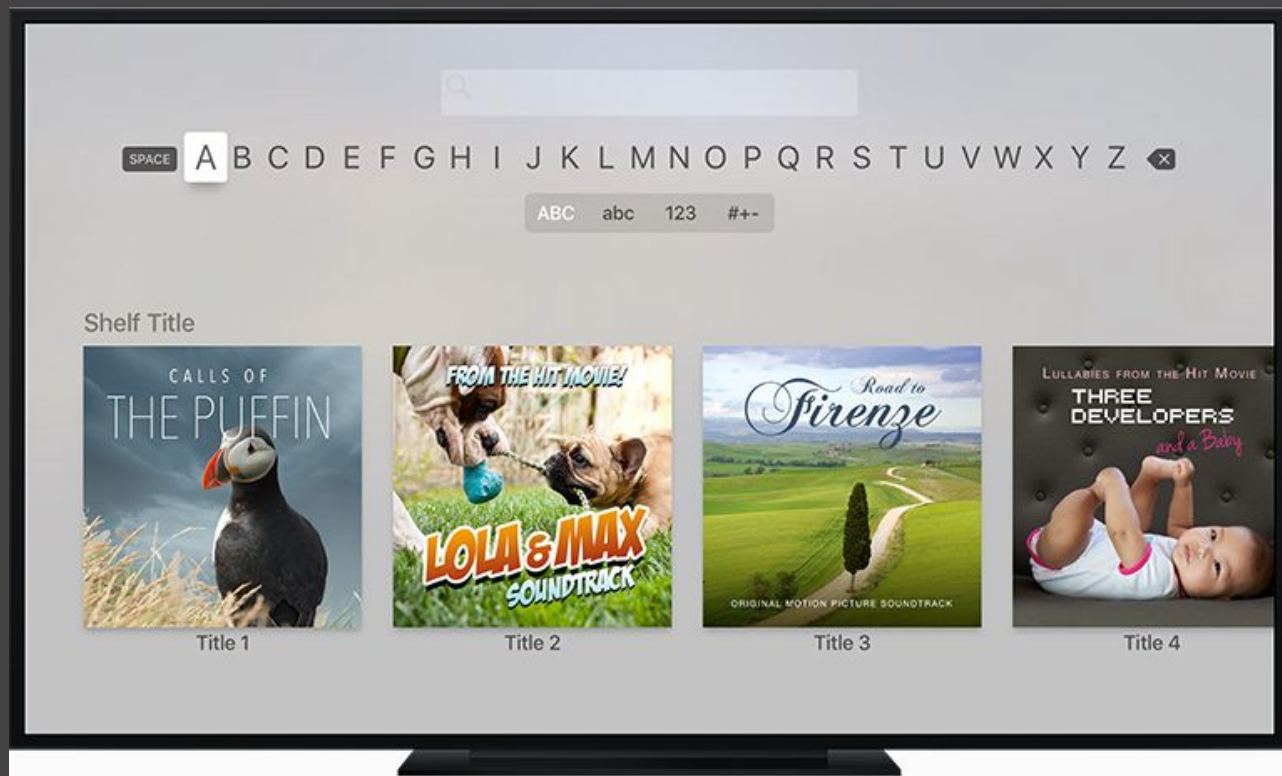
# Split Views



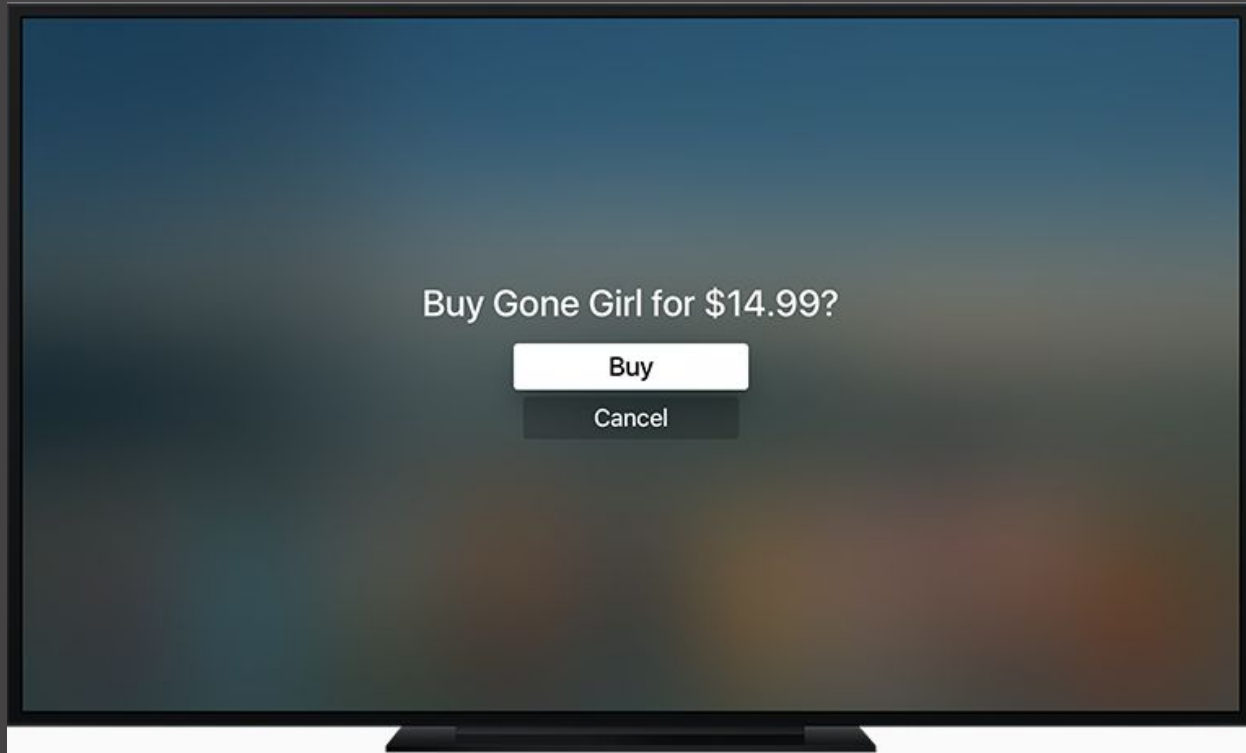
# Keyboards



# Search



# Alerts



# Tips

- Avoid displaying a back button
- Avoid creating radical new designs
- Show a collection of content on a single screen
- Favor horizontal navigation of content
- Use a tab bar to logically organize content
- Make it easy to focus on an item
- Avoid using alerts very often

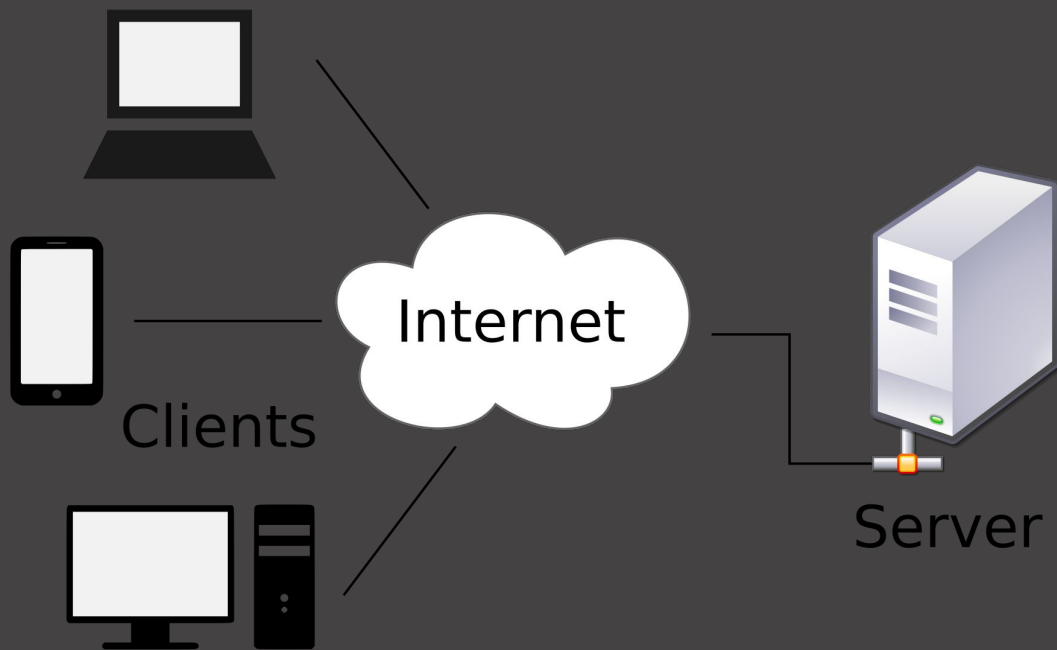


Take a break.  
**You deserve it!**





# TVMLKit



# TVMLKit Framework

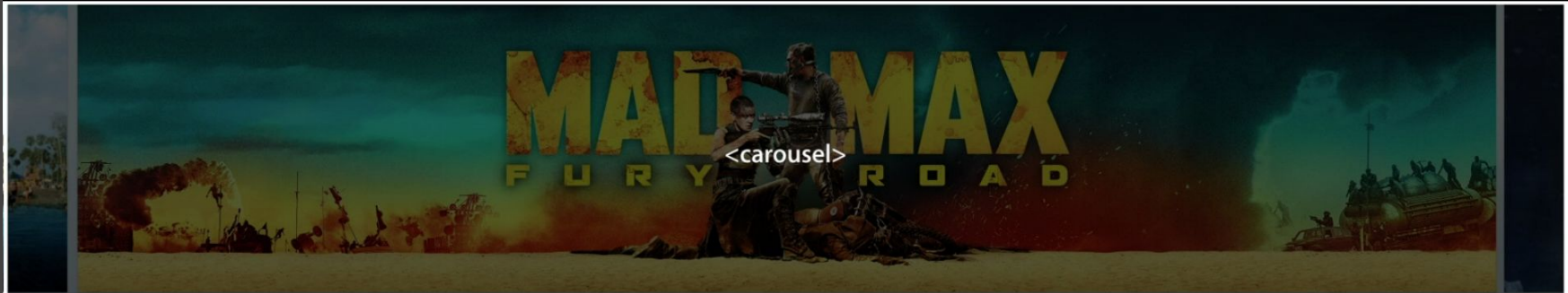
- Layout Templates
  - highly customizable
- Client Server Architecture
- TVML - TV Markup Language
- TVJS - Framework based on JavaScript
- Some native code is required



# TVML - Simple and Compound elements

- background
- banner
- buttonLockup
- description
- heroImg
- productInfo
- row
- shelf
- stack
- subtitle / text / title





<carousel>

## Top Movies



<img>

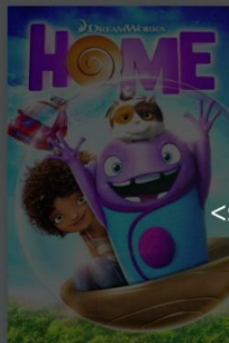
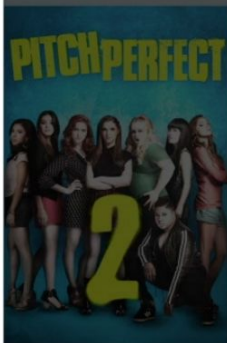


<title>

Mad Max: Fury Road

<lockup>

New & Noteworthy



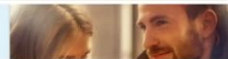
<shelf>



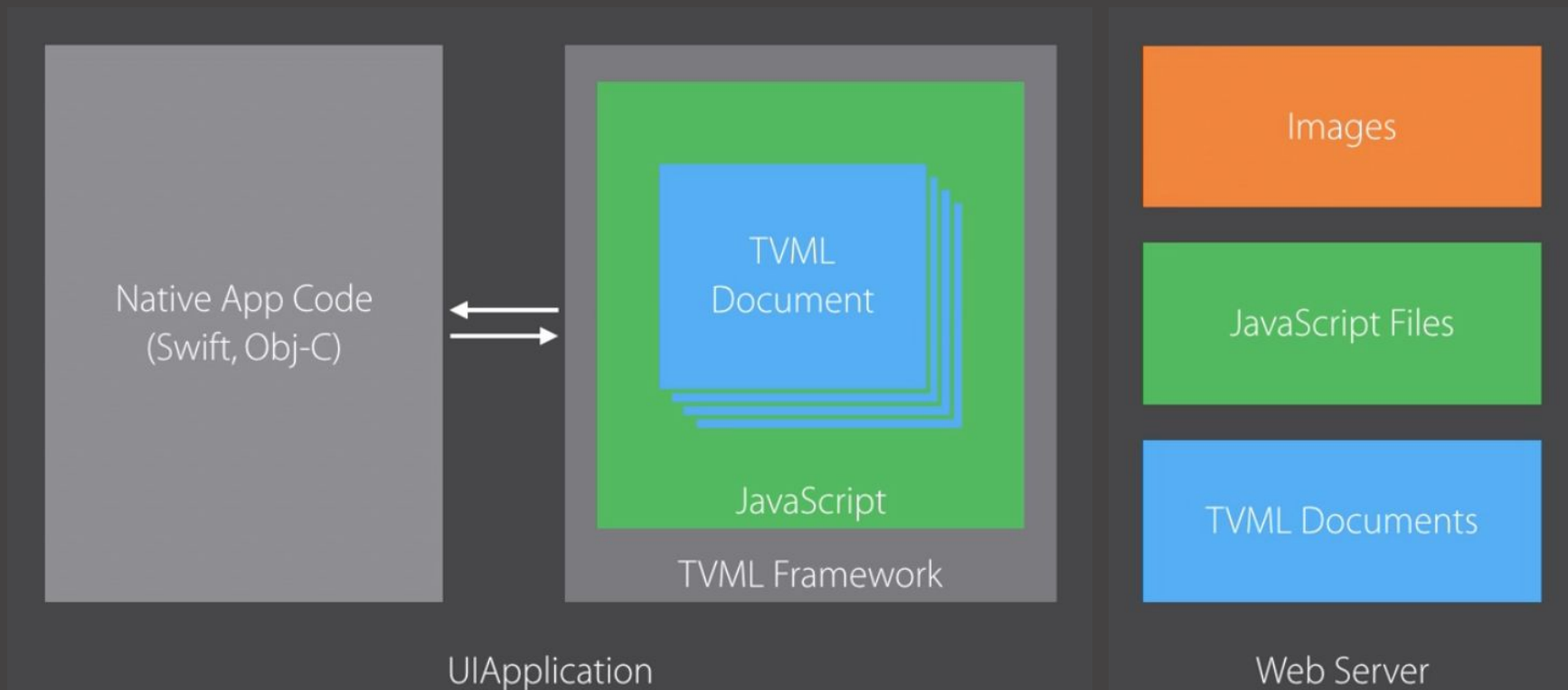
ex machina



<collectionList>



# TVML Application Structure



```
import TVMLKit
class AppDelegate: TVApplicationControllerDelegate {
    func application(application: UIApplication, didFinishLaunchingWithOptions launchOptions: ....

        let appControllerContext = TVApplicationControllerContext()

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

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

        appController = TVApplicationController(context: appControllerContext, window: window, delegate: self)
        return true
    }
```

# Summary

- Not a web browser - cannot display a web page
- Especially geared for media apps
- Reduces development efforts
- Shorter time-to-market
- Consistency between different apps





Let's jump into the real world of  
native apps



# What's available?

The image displays a collection of Apple frameworks and APIs, categorized by color:

- Blue:** CloudKit, SpriteKit, StoreKit, TVMLJS, Foundation, UIKit, CoreGraphics, QuartzCore.
- Pink:** CoreMedia, AVKit, AVFoundation, JavaScriptCore, MediaPlayer.
- Green:** TVMLKit, CoreBluetooth, TVServices.
- White:** Metal, GameKit, CoreAudio, GameController, OpenGL ES, GameplayKit.

On Demand Resources & Data Storage

# Data Storage

- Limited persistent local storage
  - UserDefaults (500 KB max)
  - Keychain
- Network
  - iCloud Key-Value Store (up to 1 GB per user, 1024 keys)
  - CloudKit
  - Developer - hosted server
- Temporary data storage
  - `NSCachesDirectory`, `NSTemporaryDirectory` (subject to purging)

# On-demand resources

- Dynamically loaded content (up to 20 GB) - hosted on Apple servers
- Downloadable during app install or by request
- No executable content
- Identified by tags (string identifiers)
- Apps request tags, not individual resources
- Purged when OS needs more local storage

# Numbers to remember

App size	Up to 200 MB
ODR content	Up to 20 GB
ODR content in run time	Up to 2 GB
Initial Install Tag	Up to 2 GB (including the app size)
Prefetched Tags	Up to 4 GB (including Initial Install tag)
Asset pack	Up to 512 MB (64 recommended)
Total number of asset packs	1000



# Time to write some code

*I will not write bad code any more!*  
*I will not write bad code any more!*  
*I will not write bad code any more!*  
*I will not write bad code any more!*  
*I will not write bad code any more!*



match  
me

Demo



# What can we improve?

- Clear
  - Show the user what is on focus
- Immersive
  - Provide Top Shelf
- Connected
  - Leverage the features of remote



# Focus Engine

# Focus



Scenery three



Top Left Button

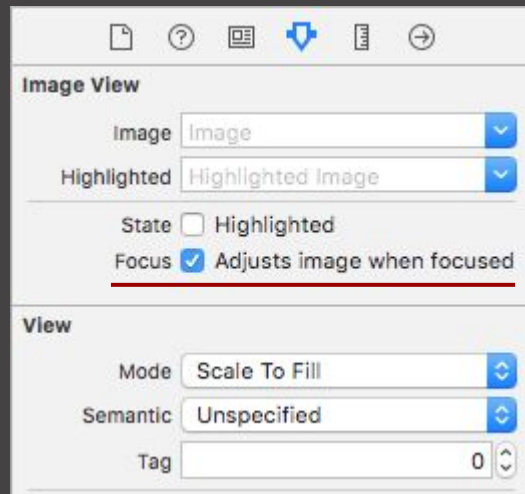
Top Right Button

# The Focus Model

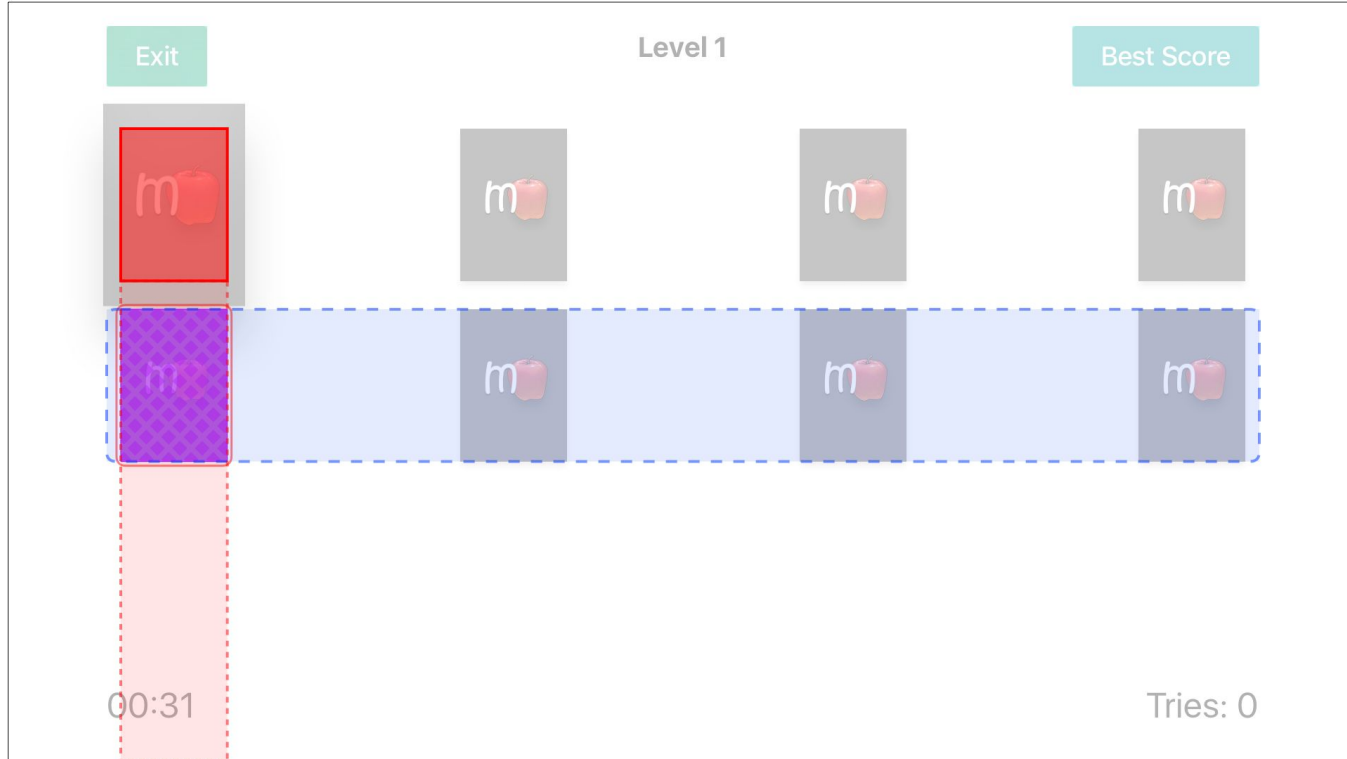
- Make focus obvious
- Make different control states distinct
- Only one view in focus
- Add parallax to current focus view
- Use click instead of tap

# Focus API

- Focusable views
  - `func canBecomeFocused( ) -> Bool`
- Initial focus - "Preferred focus chain"
  - `UIFocusEnvironment` (protocol)
  - `weak var preferredFocusedView: UIView?`
- Current focus
  - `var focused: Bool { get }`
  - `weak var focusedView: UIView? { get }`
- `func tableView(tableView: UITableView, didUpdateFocusInContext context: UITableViewFocusUpdateContext, withAnimationCoordinator coordinator: UIFocusAnimationCoordinator) -> Void`



# Testing



# Focus guides

- Part of the view hierarchy
- Not visible
- The focus engine can see it as focusable element
- `weak var preferredFocusedView: UIView?`
- Override it in
  - `func didUpdateFocusInContext(_ context: UIFocusUpdateContext, withAnimationCoordinator coordinator: UIFocusAnimationCoordinator)`

# Summary

- Implementing focus appearance
- Responding to focus changes
- Using focus guides
- Use debugging tips





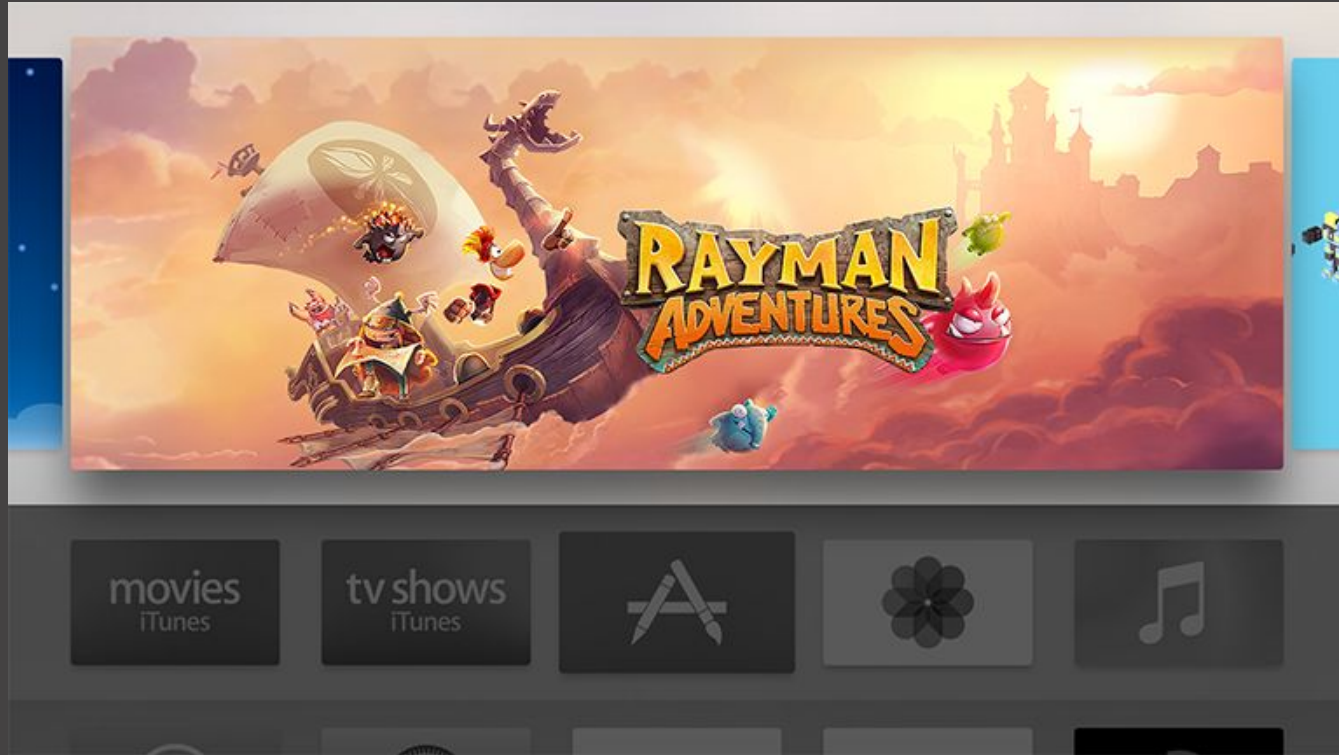
Coffeebreak

Top Shelf

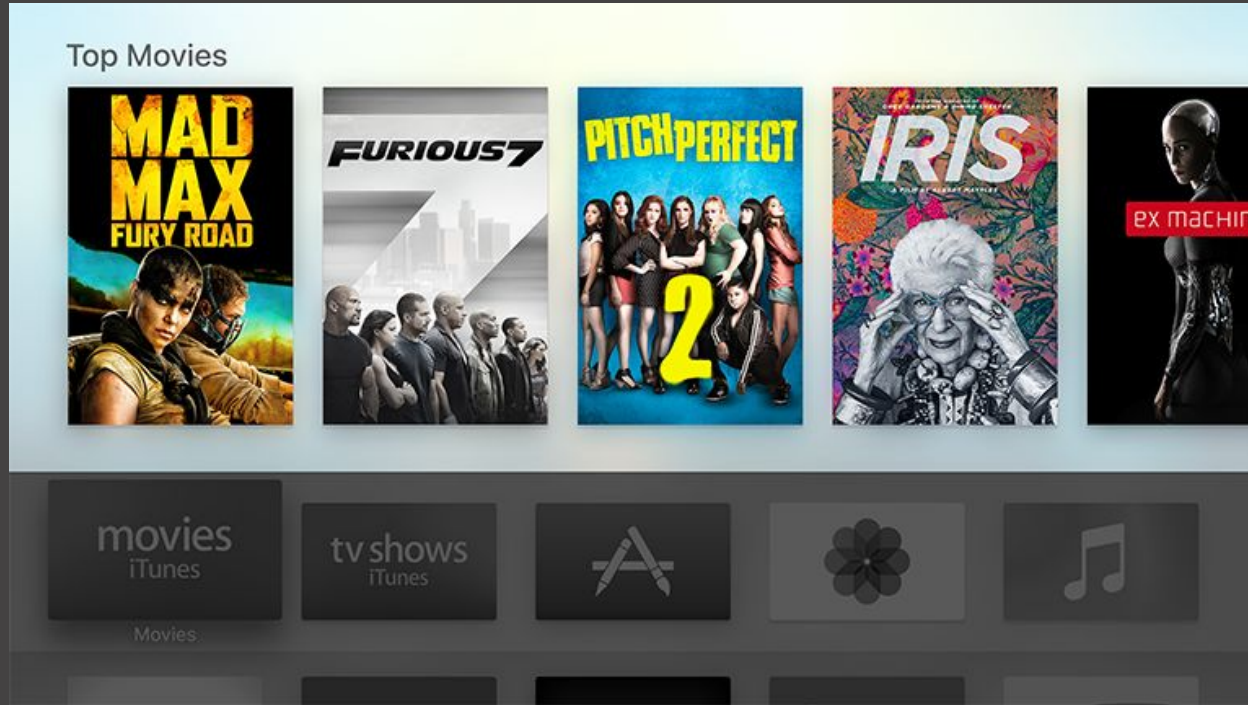
# Static Top Shelf



# Dynamic Top Shelf: Scrolling Inset Banner



# Dynamic Top Shelf: Sectioned Content Row



# TV Services Extension

```
protocol TVTopShelfProvider {  
    var topShelfStyle: TVTopShelfContentStyle { get }  
    var topShelfItems: [TVContentItem] { get }  
}
```

# TVContentItem

```
class TVContentItem : NSObject, NSCopying, NSSecureCoding {  
    var contentIdentifier: TVContentIdentifier { get }  
    var imageURL: NSURL?  
    var imageShape: TVContentItemImageShape  
    var title: String?  
    ★ var topShelfItems: [TVContentItem]?  
}
```

- 
- *None*
  - *Poster*
  - *Square*
  - *SDTV*
  - *HDTV*
  - *Wide*
  - *ExtraWide*

# Linking to Content

- Register custom URL scheme for your app
- `application(_openURL:options:)`
- `TVContentItem.displayURL`
- `TVContentItem.playURL`



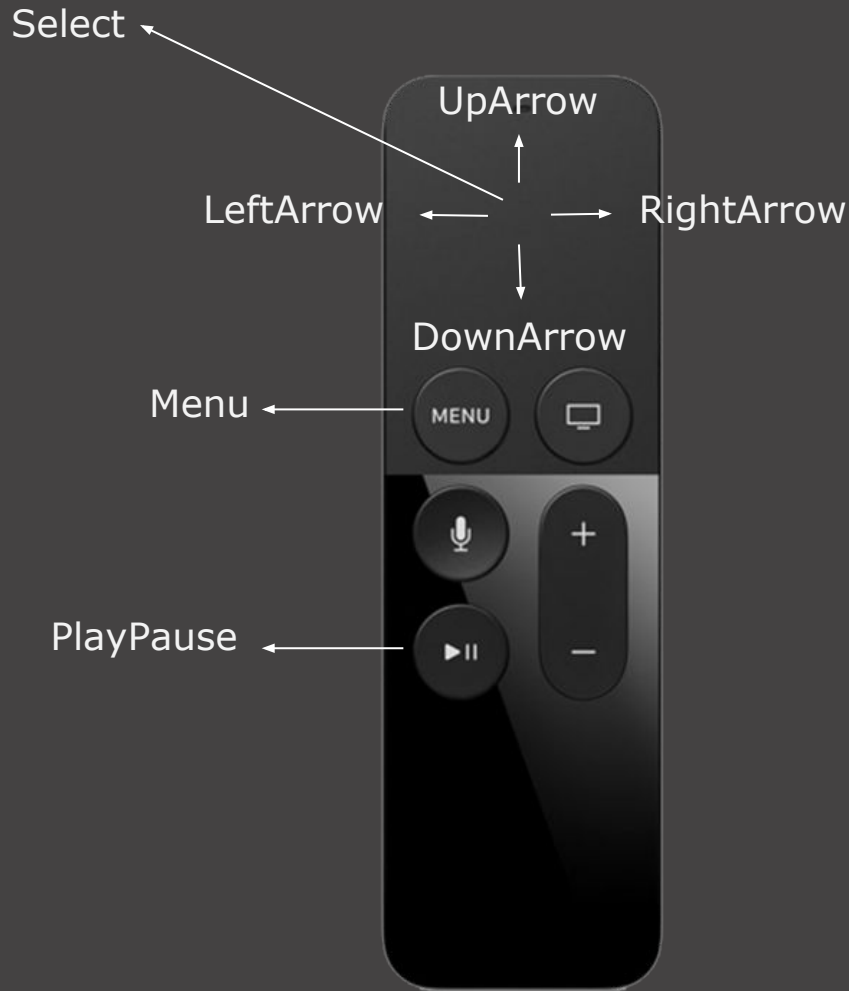
# Tips for Top Shelf

- Always provide at least static image top shelf
- Top shelf is an app extension
- Provide both `displayURL` and `playURL` for all content
- Post `TVTopShelfItemsDidChangeNotification`
- Use `NSUserDefaults` to share data between app and extension

# Remote and Gestures

# API

- Using gesture recognizers
  - `var allowedPressTypes: [NSNumber]`
  - `UIPressType`
- Low - level event handling:
  - `pressesBegan(_:withEvent:)`
  - `pressesCancelled(_:withEvent:)`
  - `pressesChanged(_:withEvent:)`
  - `pressesEnded(_:withEvent:)`

















App submission

# App Store Submission

- Profiles
- Ids
  - AppName.ExtensionName
- Submit extension

Demo transition from iOS to tvOS

# Business Models

Free	Paid	Freemium	Paymium
			
			
			

# Universal Purchase

- User purchase content ones
- Enjoy apps both on iOS and tvOS
- Discovery through Purchased Apps tab
- Share metadata
- **Decide once!**



# Apple marketing advices

- Think global, be local
- Consider top Markets
- Be sure to localize
- Product Page
- Use apple approved TV images
- Let Apple know



<https://developer.apple.com/>

*“The future of television is  
apps.”*

