



# В ЧЕМ РАЗРАБОТКА ПОД APPLE TV ОТЛИЧАЕТСЯ ОТ РАЗРАБОТКИ ПОД IOS

На основе реального проекта онлайн-кинотеатра Okko

АЛЕКСЕЙ СУХОВ

# ökkö

## Немного о себе

- Начал свою профессиональную карьеру j2me разработчиком
- Под iOS начал писать еще до появления App Store
- Первое приложение вышло под iOS 2.0
- В Okko с 2011 года

# öккo

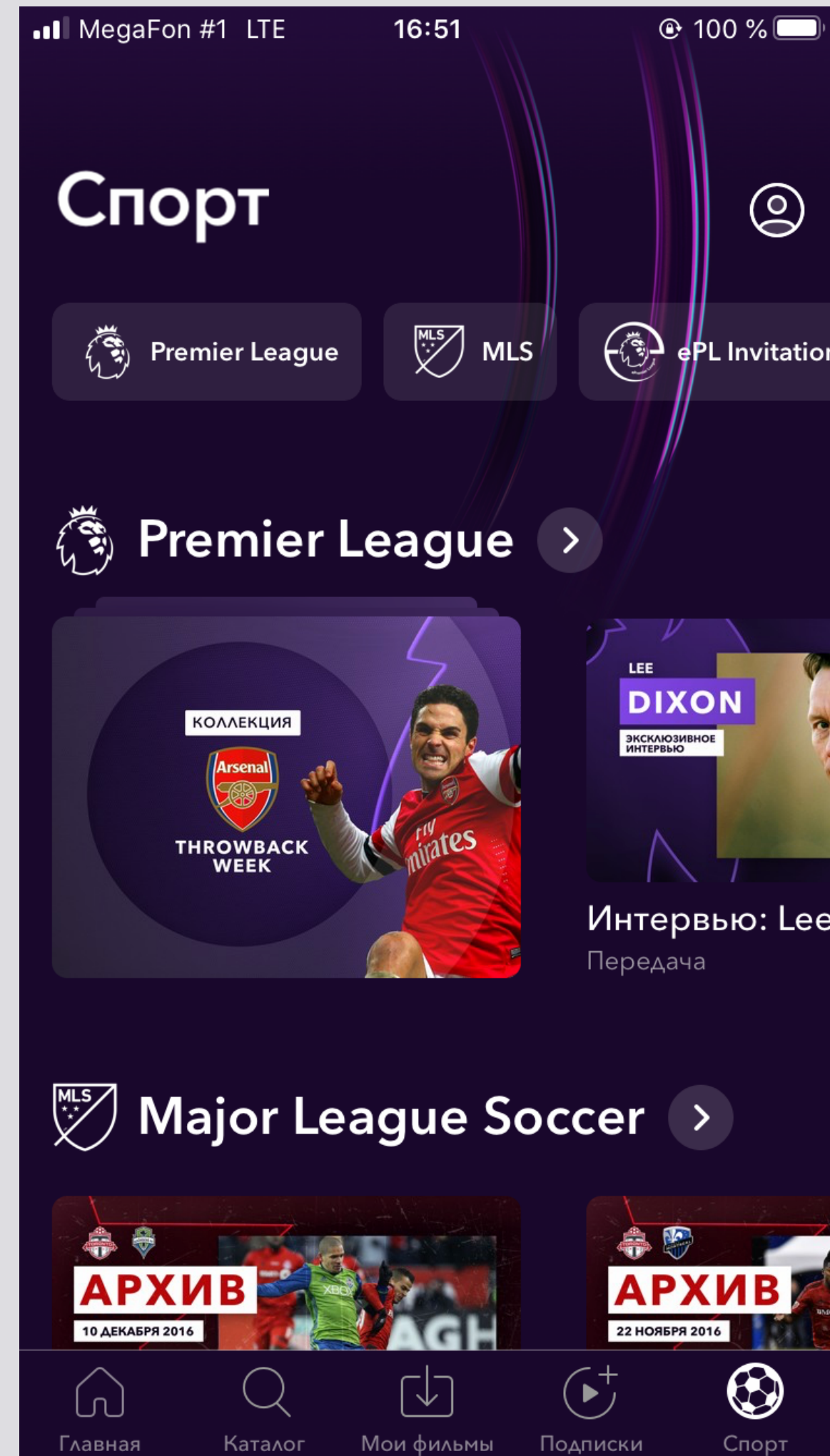
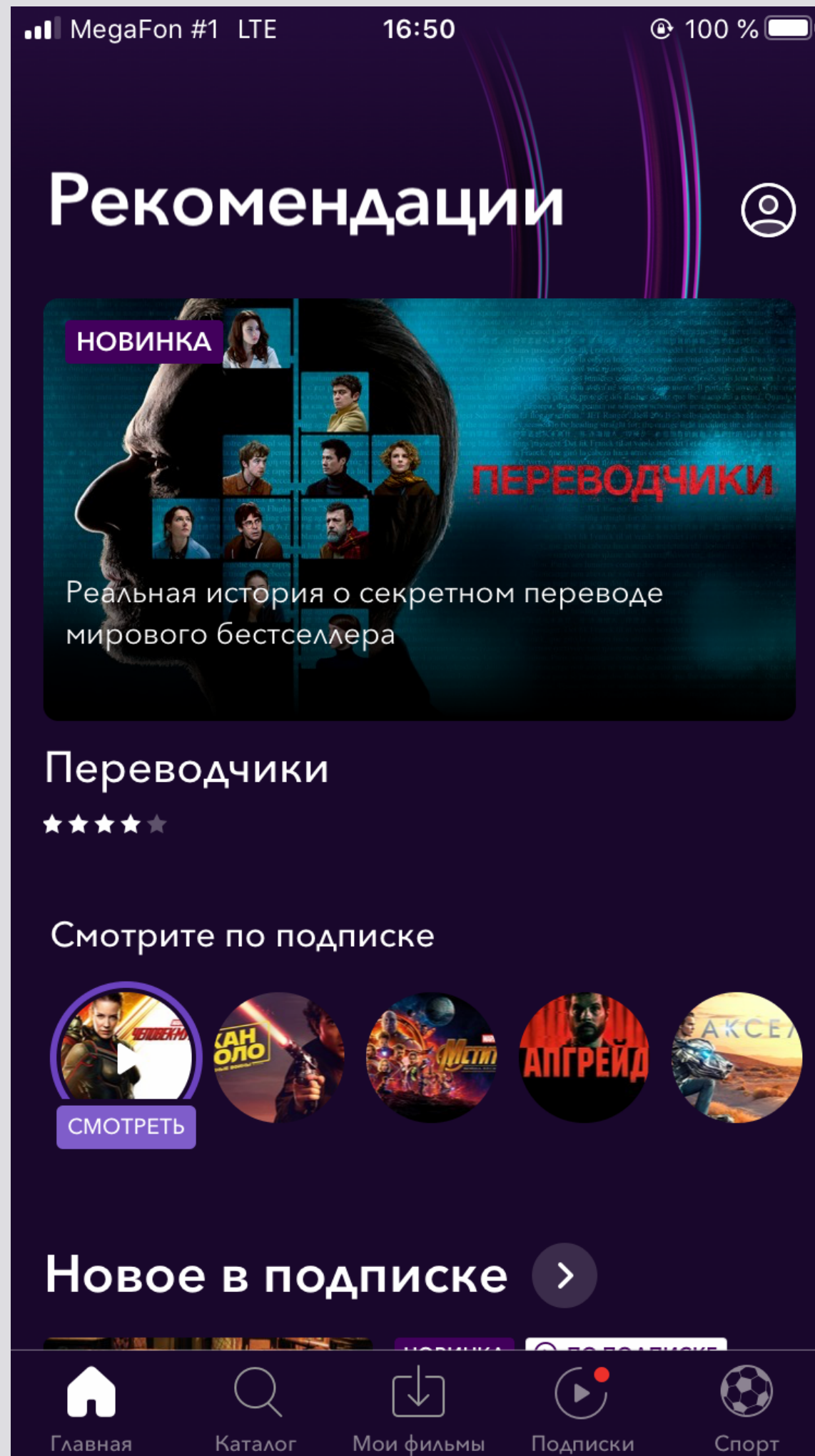
## Сeгoднeя бyдeм гoвoрить:

1. Отличия tvOS и iOS
2. Примеры реальных проблем на этапе разработки
3. Нативное приложение

# ökkö

## Исходные данные:

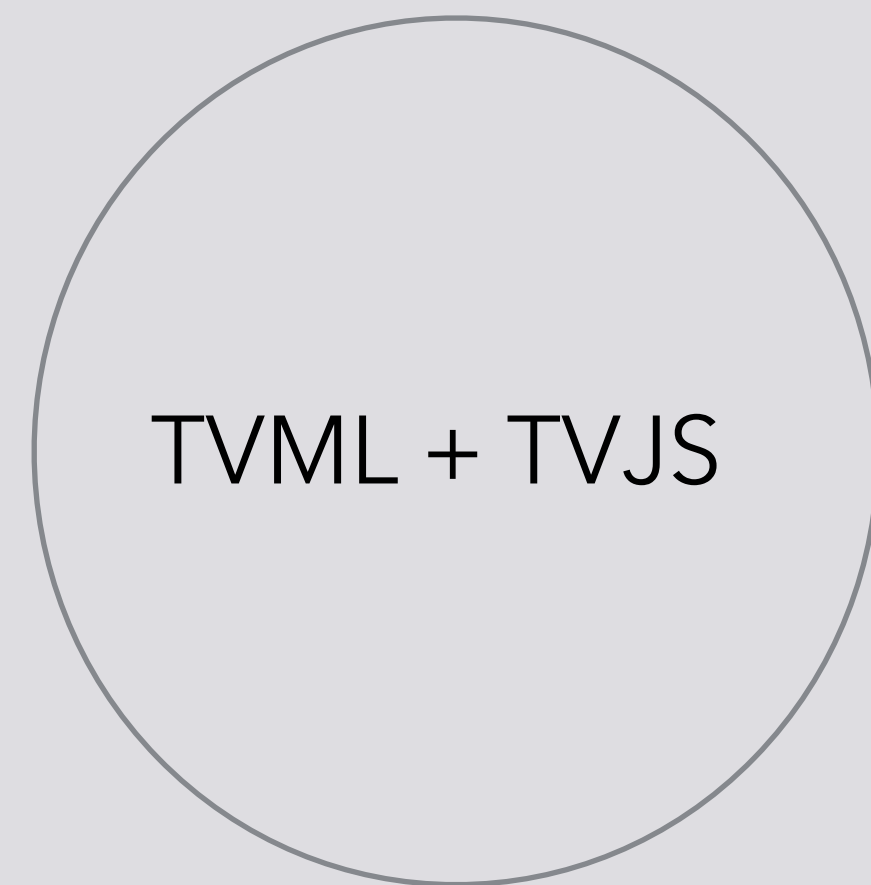
1. Готовое iOS приложение
2. Команда: 2 iOS разработчика, дизайнер и тестировщик.
3. Срок реализации 4 месяца



- Objective-C
- Почти все экраны на UICollectionView
- iPhone + iPad
- MVC
- Минимальная зависимость от внешних библиотек

# ökko

## Варианты для разработки под tvOS



**VS**



# ökko

## TVML + TVJS

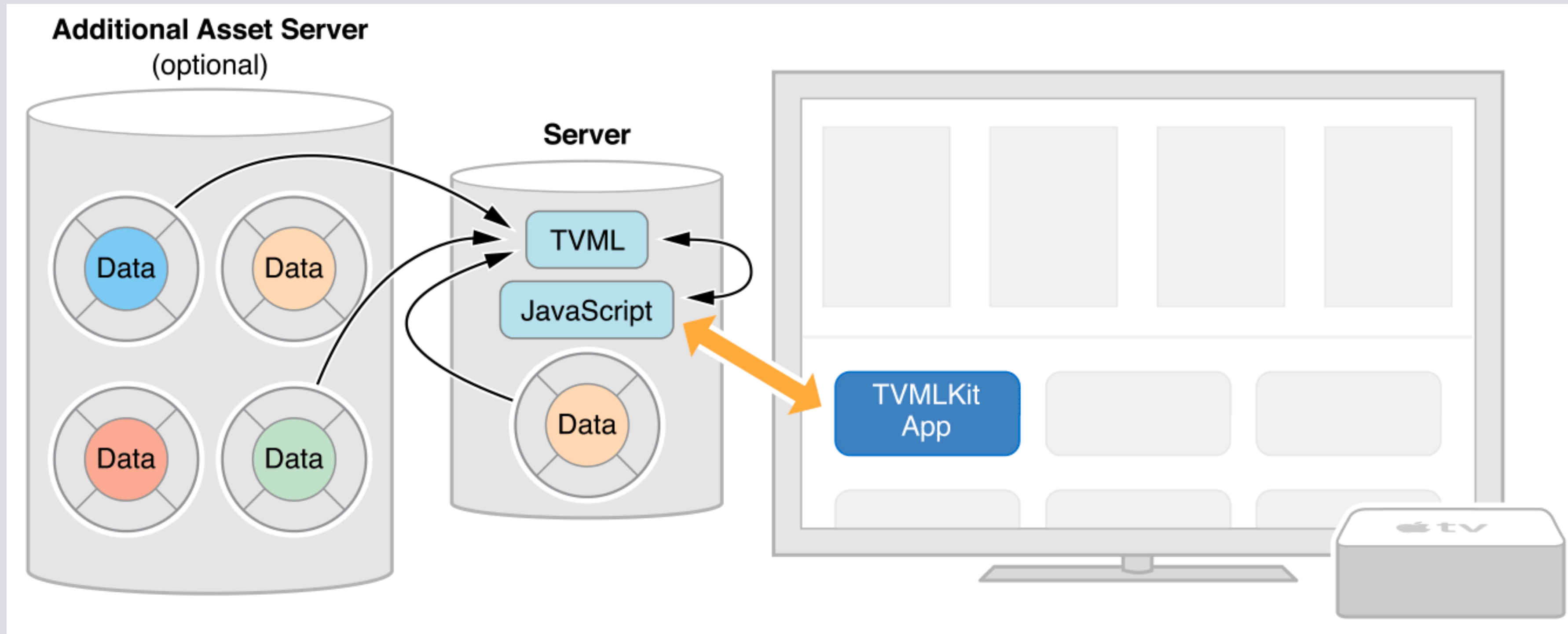
- Подходит для клиент-серверных приложений
- Ограниченный набор шаблонов
- Ограниченная кастомизация
- Быстро

Подходит для тех, кто знает JS лучше Swift/Objective-C

Может сэкономить время

```
<document>
  <alertTemplate>
    <title>Update to premium</title>
    <description>Go ad free by updating to the premium version</
description>
    <button>
      <text>Update Now</text>
    </button>
    <button>
      <text>Cancel</text>
    </button>
  </alertTemplate>
</document>
```





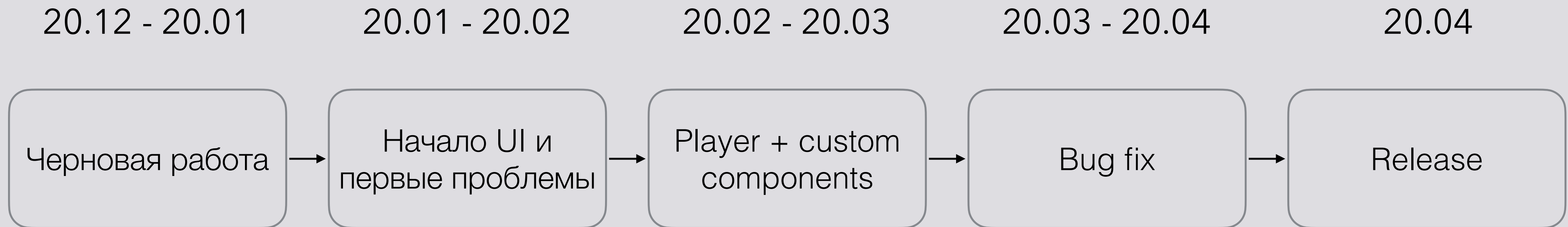
# ökko

## Native application

- Swift/Objective-C
- Переиспользование компонент с iOS приложения
- Неограниченная кастомизация

# ökko

## Составим план действий



## Локальное хранилище

- Локальное хранилище ограничено
- iCloud Key-Value Storage (KVS) и CloudKit
- UserDefaults на 500KB
- Cache directory
- Максимальный размер приложения 4GB(200MB)

## CoreData

```
40  
41     try persistentStoreCoordinator.addPersistentStore(ofType:  
        NSSQLiteStoreType, configurationName: config, at: url)
```

```
40  
41     try persistentStoreCoordinator.addPersistentStore(ofType:  
        NSInMemoryStoreType, configurationName: config, at: url)
```

# Keychain & UserDefaults



# Network Layer & Model

JSON Schema:

```
1. {
2.   "type" : "object",
3.   "$schema" : "http://json-schema.org/draft-04/schema",
4.   "description" : "Profile request",
5.   "definitions" : {
6.     "user" : {
7.       "type" : "object",
8.       "properties" : {
9.         "id" : { "type" : "string" }
12.      }
13.    }
14.  },
15.  "properties" : {
16.    "status" : { "type" : «number" },
19.    "user" : { "$ref" : "#/definitions/user" }
22.  }
23. }
```

## Не все фреймворки готовы для работы на tvOS

- Аналитика
- Социальные сети
- DRM



## Только на tvOS:

- TVServices(Top Shelf)
- Parallax Effect
- Focus Engine
- TVMLKit
- TVJS
- Light/Dark



## Focus Engine

- Только Focus Engine может явно обновить фокус
- Протокол UIFocusEnvironment обеспечивает связь между Focus Engine и приложением
- UIView, UIViewController и другие классы поддерживают протокол

```

176
177 public protocol UIFocusEnvironment : NSObjectProtocol {
178     var preferredFocusEnvironments: [UIFocusEnvironment] { get }
179     func setNeedsFocusUpdate()
180     func updateFocusIfNeeded()
181     func didUpdateFocus(in context: UIFocusUpdateContext, with coordinator:
        UIFocusAnimationCoordinator)
182 }

```

## Поддержка фокуса в UICollectionView

```
// Support for Focus
open var remembersLastFocusedIndexPath: Bool

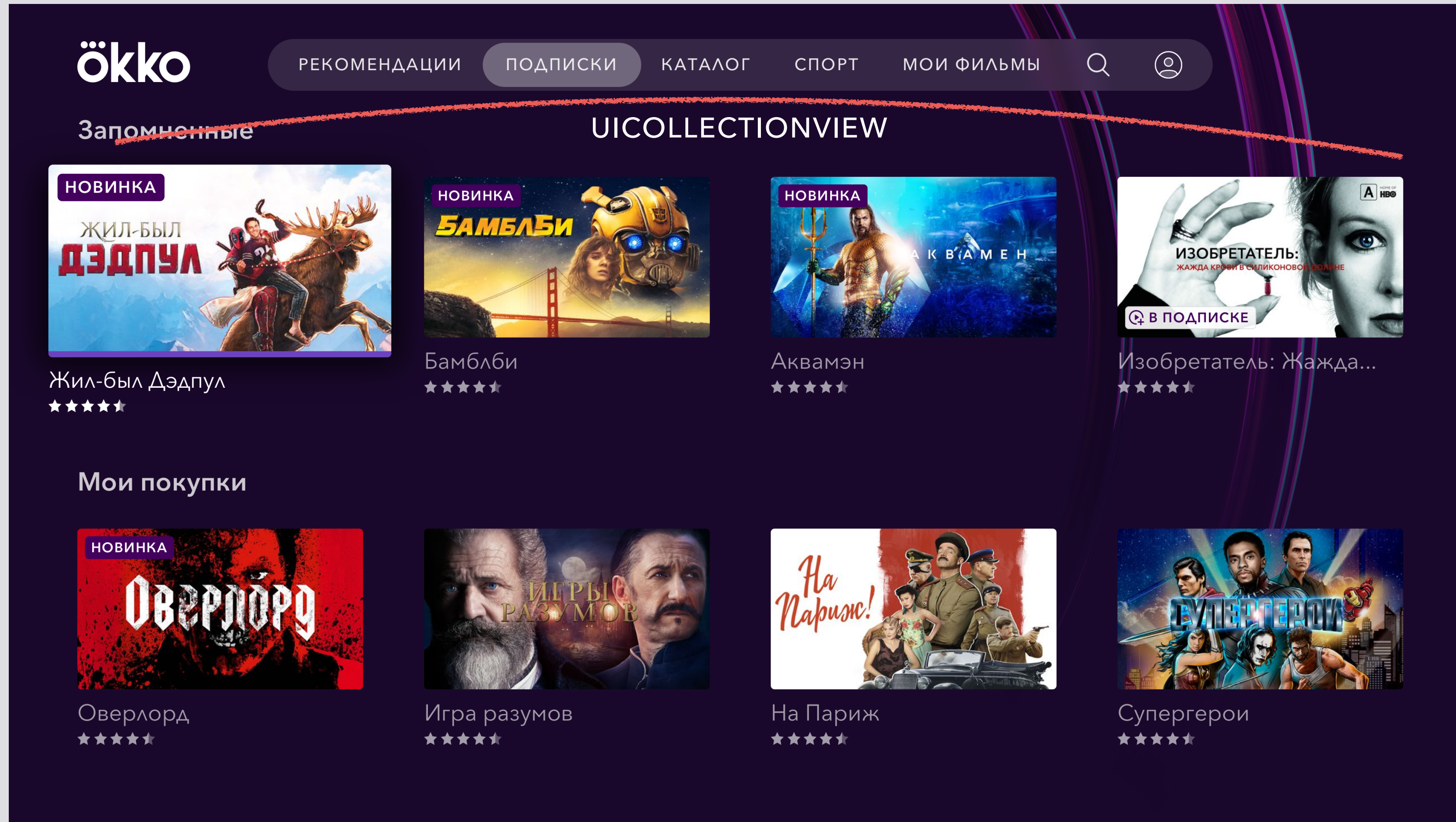
// Focus
@available(tvOS 9.0, *)
optional func collectionView(_ collectionView: UICollectionView, canFocusItemAt indexPath: IndexPath) -> Bool

@available(tvOS 9.0, *)
optional func collectionView(_ collectionView: UICollectionView, shouldUpdateFocusIn context:
UICollectionViewFocusUpdateContext) -> Bool

@available(tvOS 9.0, *)
optional func collectionView(_ collectionView: UICollectionView, didUpdateFocusIn context:
UICollectionViewFocusUpdateContext, with coordinator: UIFocusAnimationCoordinator)

@available(tvOS 9.0, *)
optional func indexPathForPreferredFocusedView(in collectionView: UICollectionView) -> IndexPath?
```

# Первая проблема с Focus



СЕКЦИЯ 1:  
CELL1

СЕКЦИЯ 2:  
CELL1

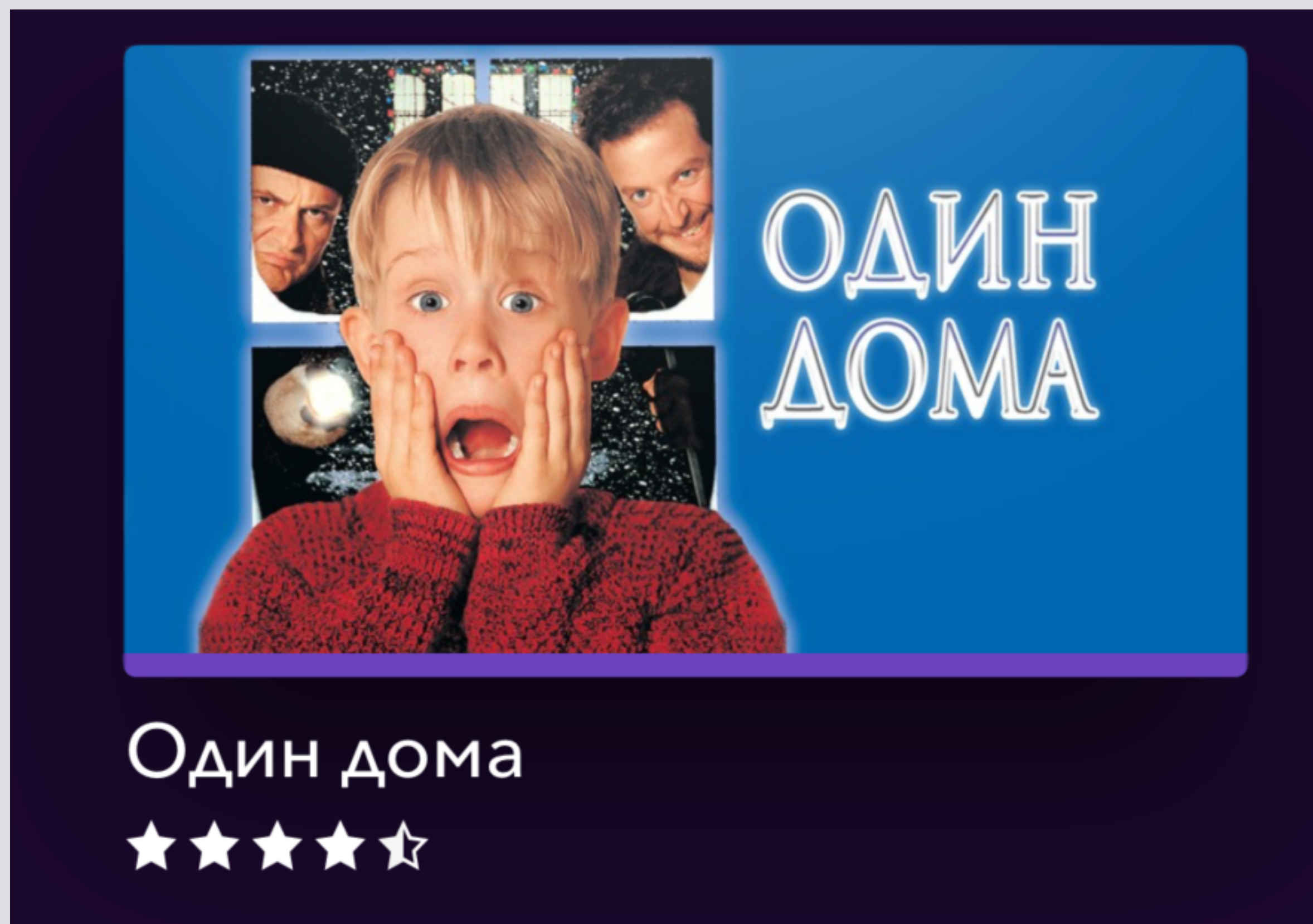
## Первая проблема с Focus

```
func collectionView(_ collectionView: UICollectionView, canFocusItemAt
    indexPath: IndexPath) -> Bool {
    return collectionView != self.collectionView
}
```

## Первая проблема с Focus



## Сделай красиво



## Делаем красиво

```

    override fun didUpdateFocus(in context: UIFocusUpdateContext, with coordinator:
UIFocusAnimationCoordinator) {
        super.didUpdateFocus(in: context, with: coordinator)
        if self == context.nextFocusedView {
            coordinator.addCoordinatedFocusingAnimations({ (context) in

        })
        } else if (self == context.previouslyFocusedView) {
            coordinator.addCoordinatedUnfocusingAnimations({ (context) in

        })
        }
    }
}

```



```

titleLabel.isHighlighted = isFocused
ratingControl.alpha = isFocused ? 1 : 0.6

var transform = CGAffineTransform.identity

if isFocused {
    let focusedSize = imageView.focusedFrameGuide.layoutFrame.size
    transform = CGAffineTransform(translationX:
(imageView.frame.size.width - focusedSize.width) / 2, y: (focusedSize.height -
imageView.frame.size.height) / 2)
}

titleLabel.transform = transform
ratingControl.transform = transform

```

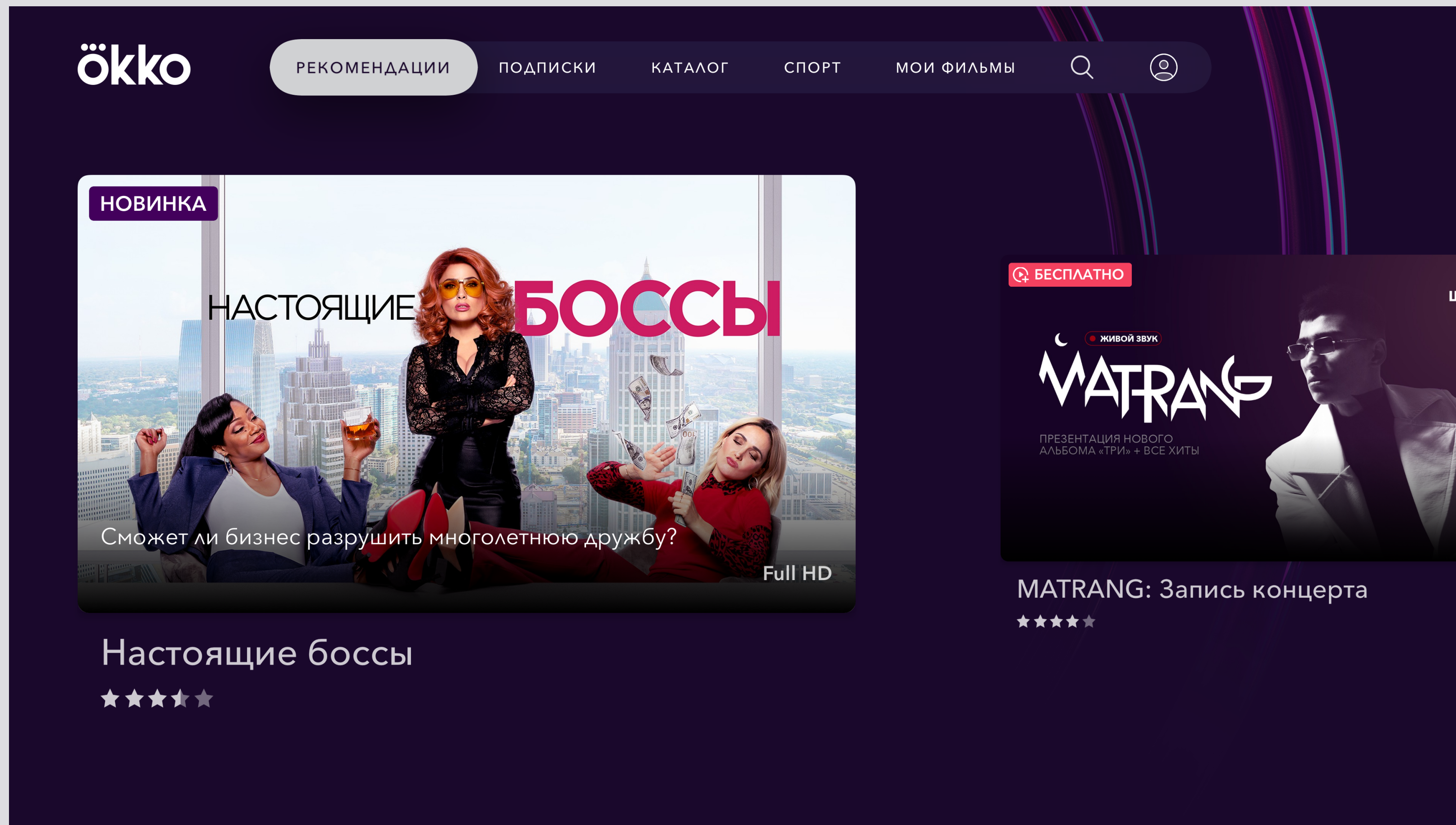
# Запомненные



Один дома



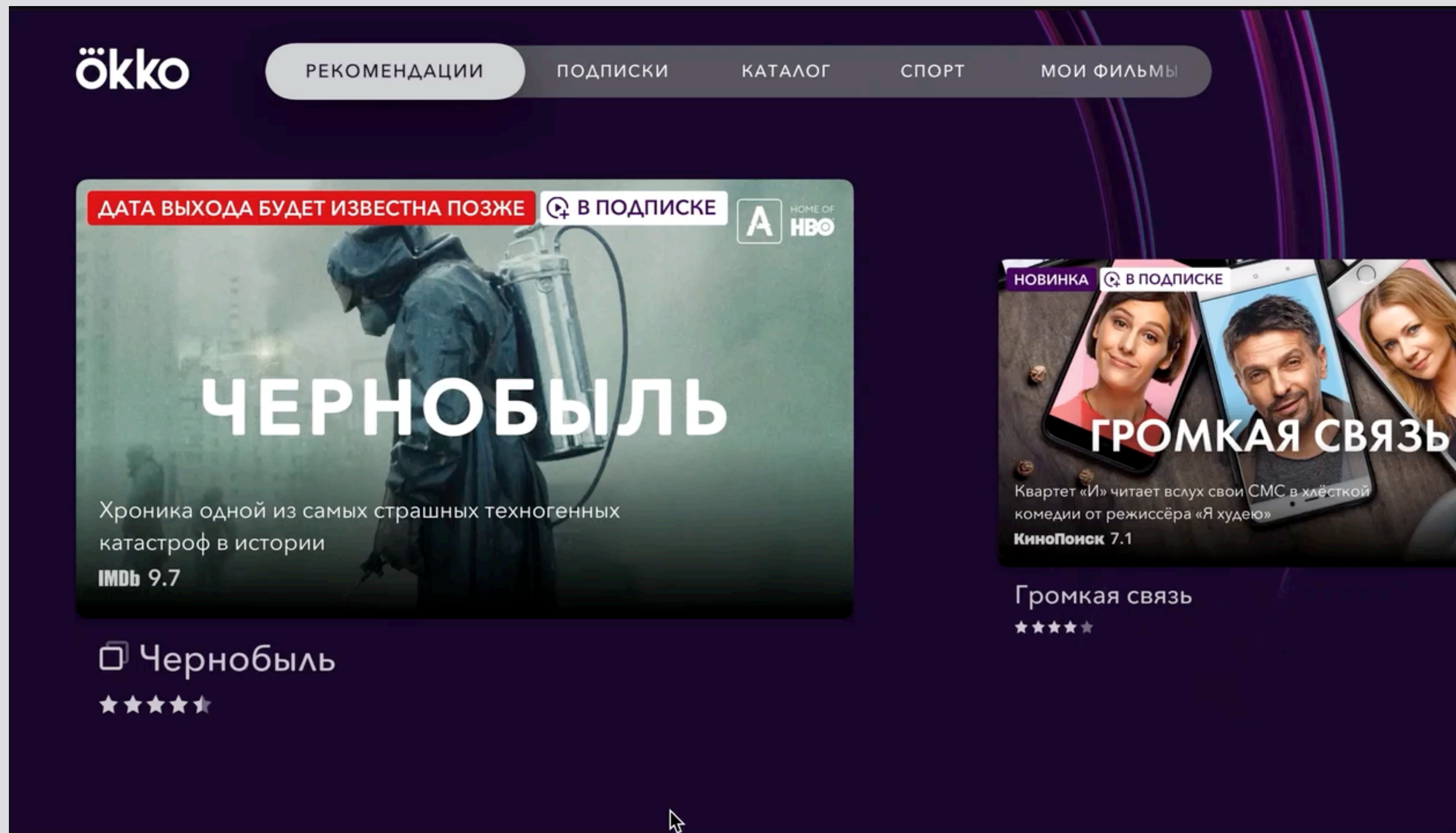
# Custom Focus Transform



## Первый шаг

```
        if imageView.adjustsImageWhenAncestorFocused {
            if isFocused {
                let scale = imageView.bounds.size.height /
imageView.focusedFrameGuide.layoutFrame.size.height
                imageView.transform = .init(scaleX: scale, y: scale)
            } else {
                imageView.transform = .identity
            }
        }
    }
```

# Custom UICollectionViewLayout с фокусом



## Custom UICollectionViewLayout с фокусом

- Проставляем в layout focus index path для того, чтобы правильно вычислять оффсет
- На элемент в фокусе не накладывается никаких трансформаций
- Инвалидируем layout при каждой смене оффсета
- Оффсет меняется, так как только 1 элемент может находиться полностью в области видимости
- Все элементы расположены последовательно с реальным размером и далее на них накладывается трансформация
- На все элементы без фокуса накладывается трансформацию

## Custom UICollectionViewLayout с фокусом

```
    optional func collectionView(_ collectionView:
UICollectionView, shouldUpdateFocusIn context:
UICollectionViewFocusUpdateContext) -> Bool
```

## Custom UICollectionViewLayout с фокусом

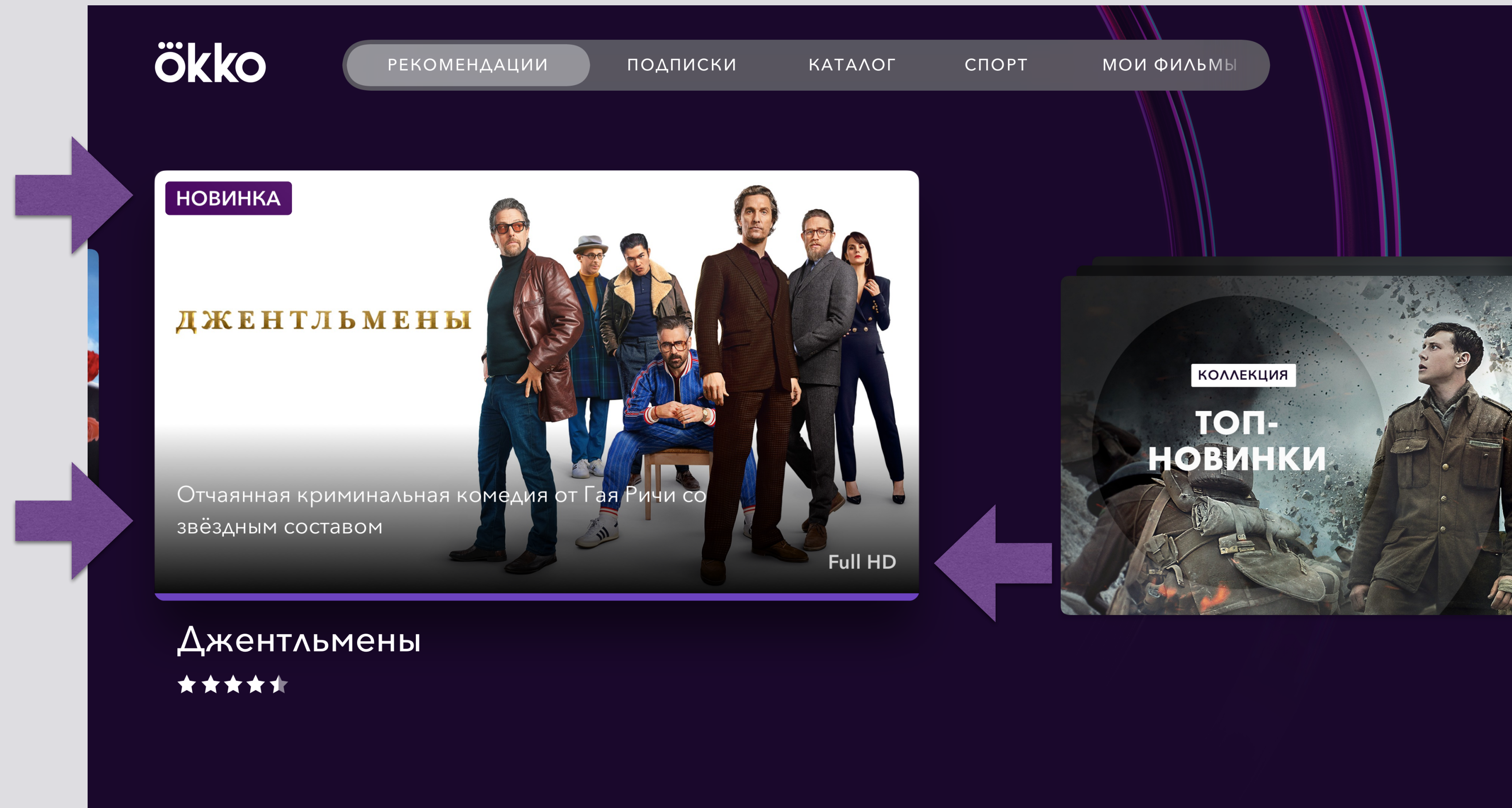
```
    override func targetContentOffset(forProposedContentOffset
proposedContentOffset: CGPoint, withScrollingVelocity velocity: CGPoint) ->
CGPoint {
    let smallWidth = CGFloat(ceilf(Float(itemScale * itemSize.width)))
    var positionX = ...
    return CGPoint(x: max(0, positionX), y: proposedContentOffset.y)
}
```



## Custom UICollectionViewLayout с фокусом

```
override func prepare() {  
    super.prepare()  
    ...  
    attribute.transform = CGAffineTransform(scaleX: scale, y: scale)  
    attribute.bounds = CGRect(origin: CGPoint.zero, size: itemSize)  
    let centerX = CGFloat(positionX + attribute.frame.size.width / 2))  
    attribute.center = CGPoint(x: centerX, y: positionY)  
    ...  
}
```


# UIImageView внутри UICollectionViewCell



# UICollectionViewCell собрали, но не работает

НОВИНКА

**ДЖЕНТЛЬМЕНЫ**



Отчаянная криминальная комедия от Гая Ричи со звёздным составом

КиноПоиск 7.0 Full HD

Джентльмены

★★★★★

# Иерархия



## Поиск решения

Загрузка многослойной картинки с сервера

Создание картинки со всей необходимой информацией на клиенте

Использование дополнительного слоя у UIImageView

## Решение

```
@available(tvOS 11.0, *)  
open var overlayContentView: UIView { get }  
  
imageView.overlayContentView.addSubview(overlayView)
```

# Результат

НОВИНКА

**ДЖЕНТЛЬМЕНЫ**

Отчаянная криминальная комедия от Гая Ричи со звёздным составом

Full HD

Джентльмены

★★★★★

# öko

## Поддержка тем в tvOS

- tvOS 10
- Важно поддерживать темную тему



# Top Shelf

```

@available(tvOS 13.0, *)
open class TVTopShelfContentProvider : NSObject {

    /// Load Top Shelf content.
    ///
    /// If the completion handler is not called in a timely manner or is called with empty content the
    system will display the static image instead.
    ///
    /// – parameter completionHandler: You must call this handler and should do so as soon
    as possible. The handler can be called on any queue.
    open func loadTopShelfContent(completionHandler: @escaping
(TVTopShelfContent?) -> Void)

    /// Can be called from an application or extension to tell the system that your Top Shelf content has
    changed.
    open class func topShelfContentDidChange()
}

```

# Top Shelf

```

    override fun loadTopShelfContent(completionHandler: @escaping
(TVTopShelfContent?) -> Void) {

        //load content...

        let size = TVTopShelfSectionedContent.imageSize(for: .hdtv)
        let items = elements.compactMap { self.makeShelfItem(from: $0, size:
size) }
        //item.displayAction = TVTopShelfAction(url: deeplink)

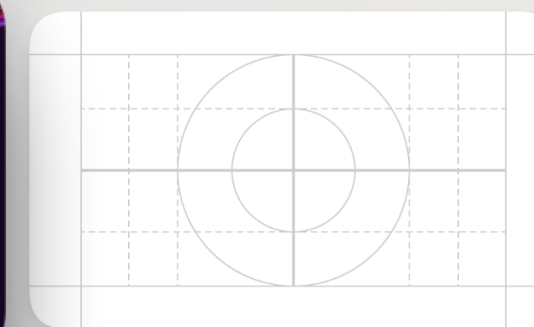
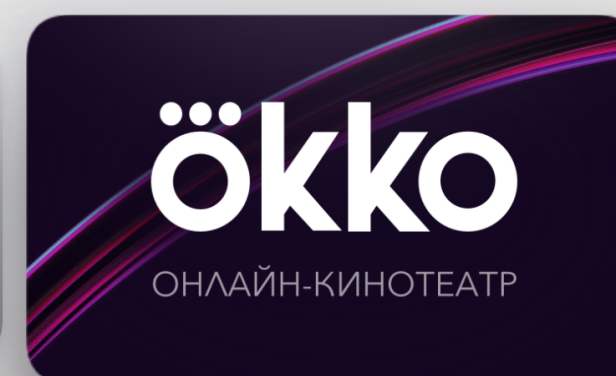
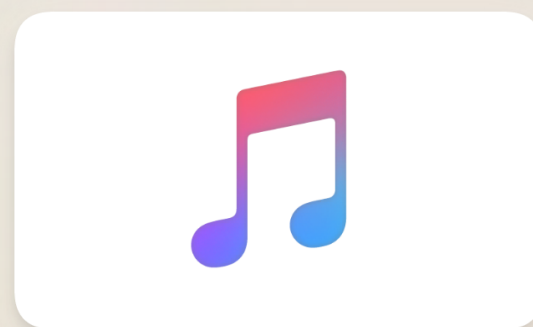
        let section = TVTopShelfItemCollection(items: items)
        let content = TVTopShelfSectionedContent(sections: [section])
        completionHandler(content)
    }

```

## Top shelf

```
func application(_ app: UIApplication, open url: URL, options:
[UIApplication.OpenURLOptionsKey: Any] = [:]) -> Bool {
    if let params = deeplinkManager.parse(url: url) {
        return deeplinkManager.processParams(params)
    }
    return false
}
```

# Результат



## Player

- Перешли на FairPlay
- Используем нативные контролы управления
- Preroll movie

# ökko

## tvOS

- Устройства устаревают медленно
- Платформа для экспериментов
- Удачные решения внедряются в iOS

**СОЗДАВАЙТЕ КРУТЫЕ  
ПРИЛОЖЕНИЯ!**

**ВОПРОСЫ?**



**СПАСИБО**