


Как укротить виджеты в супер-аппе?

Егор Болдырев

 Старший разработчик

 Дизайн-развитие (UX)
Мобильный Банк



 e.boldyrev@tinkoff.ru

 eboldyrev

- В iOS разработке 4.5 года;
- Интервьюер;
- За образовательный движ.

Что нас сегодня ждёт?



«Oh *#!@, here were go again»

Пройдемся по ключевым
моментам технологии виджетов



«Зачем? Да.»

Какие виджеты можно сделать?
Зачем они нужны? Какие задачи
решают?



«Приключение на 20 минут...»

Покажу, как мы разрабатывали
виджеты и что узнали во время
разработки



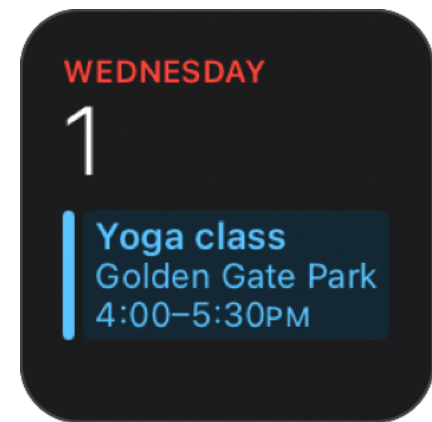
«Документация? Это что?»

Посмотрим на примеры, когда
нужной информации нет в
официальной документации

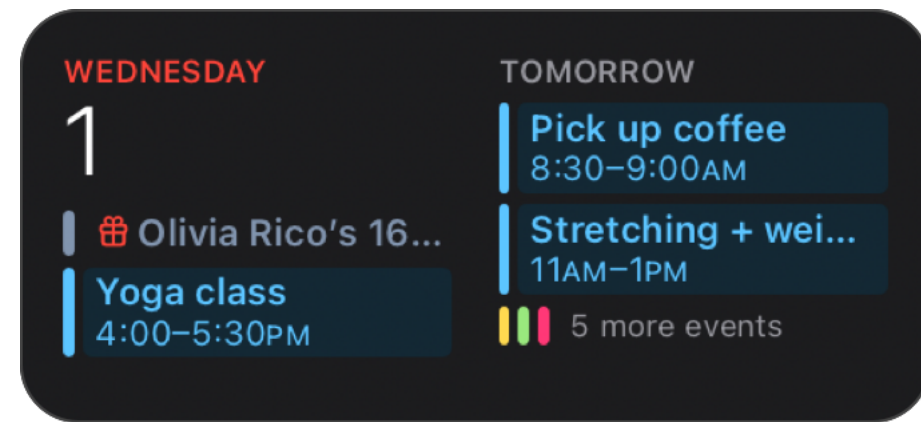
Теория



ВИДЫ ВИДЖЕТОВ



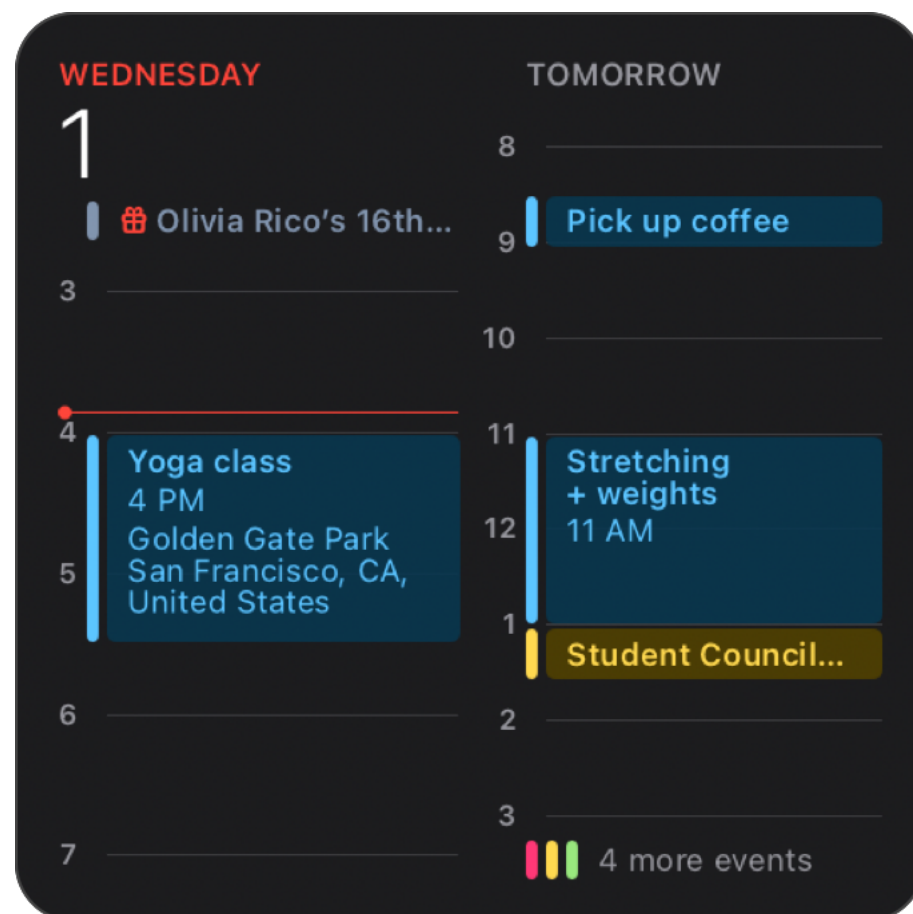
.systemSmall



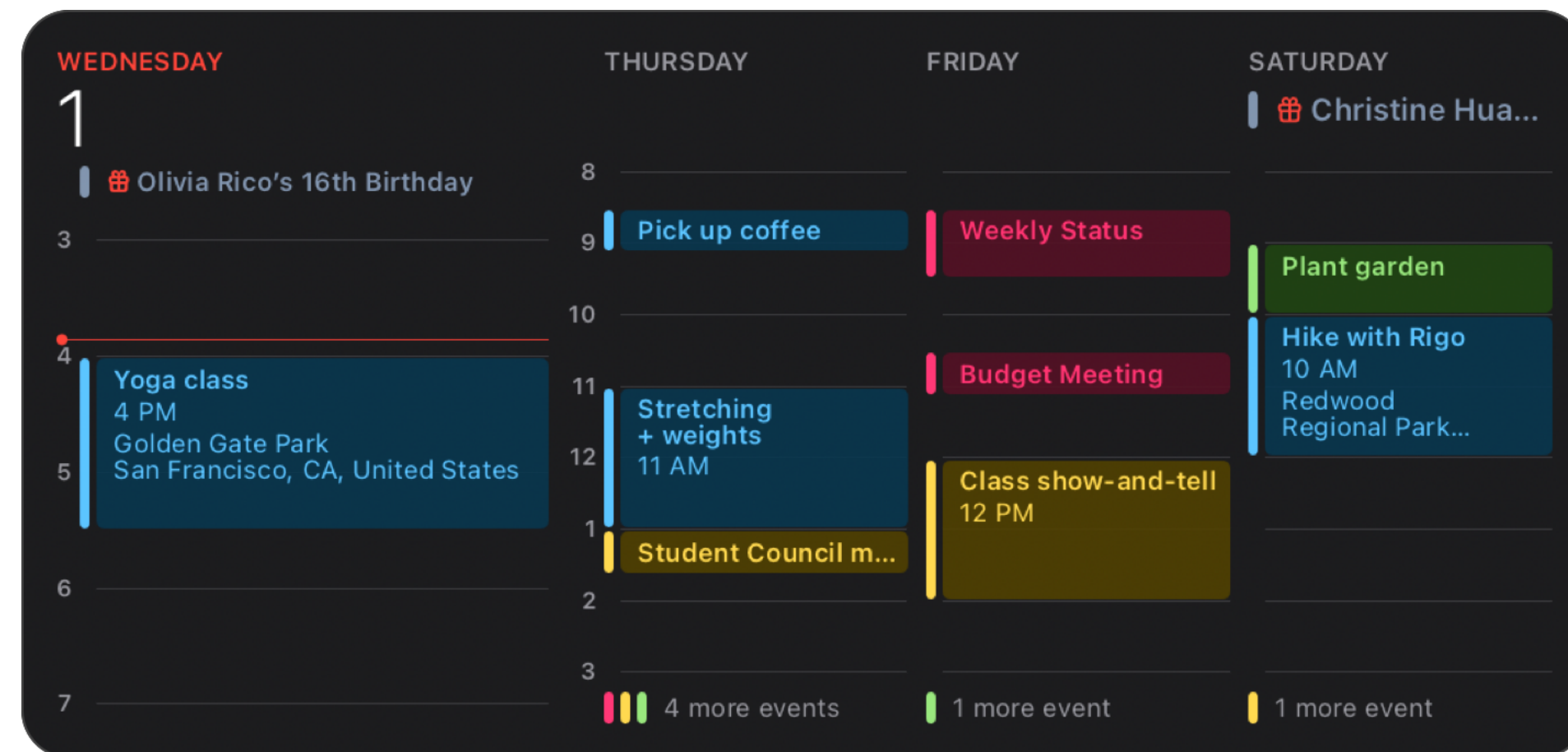
.systemMedium



.accessoryInline



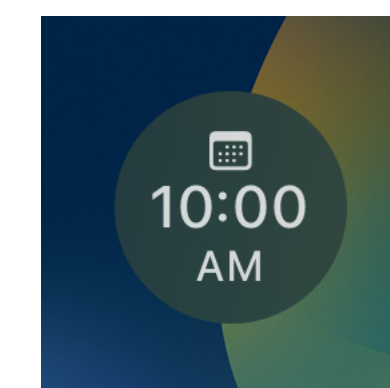
.systemLarge



.systemExtraLarge



.accessoryRectangular



.accessoryCircular

Виджеты должны смотреться гармонично



В каждом варианте виджет отображается только та информация, которая напрямую связана с его основным назначением.



Предлагайте свой виджет в нескольких размерах - это повышает его ценность.



Избегайте создания виджета, который запускает только ваше приложение.



Отдавайте предпочтение динамической информации, которая меняется в течение дня.

Виджеты должны смотреться гармонично



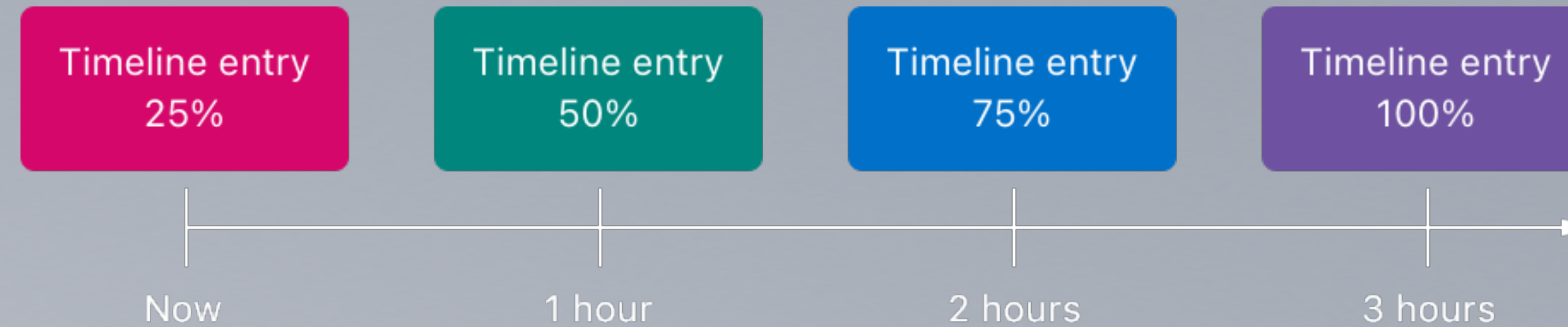
Ищите возможности удивить и порадовать пользователя.



Сообщите пользователям, что аутентификация в вашем приложении повышает ценность виджета.

Как устроен виджет

- Виджет - последовательность entry с некоторой периодичностью.
- Структура, порождающая entry - TimelineProvider.



Обновление виджета: что на это влияет?



Частота просмотра

Сколько раз и как долго по времени пользователь видит виджет?



Время обновления

Когда в последний раз виджет обновился?



Активность приложения

Пользователь использует основное приложение прямо сейчас?

Окно бюджета - 24 часа

Окно не обязательно соответствует обычным суткам

В среднем на виджет приходится 40-70 обновлений

iOS самостоятельно определит бюджет и частоту обновления

Система не будет считать обновлением:

01

Основное приложение виджета в состоянии foreground;

02

У основного приложения активна сессия навигации или аудио;

03

Виджет выполняет пользовательское действие через intent;

04

Виджет выполняет анимацию;

05

Система меняет локаль;

06

Меняются настройки accessibility и dynamic type.

Обновление виджета предсказуемо



.atEnd

Система запрашивает следующий массив entry после отрисовки последнего элемента;



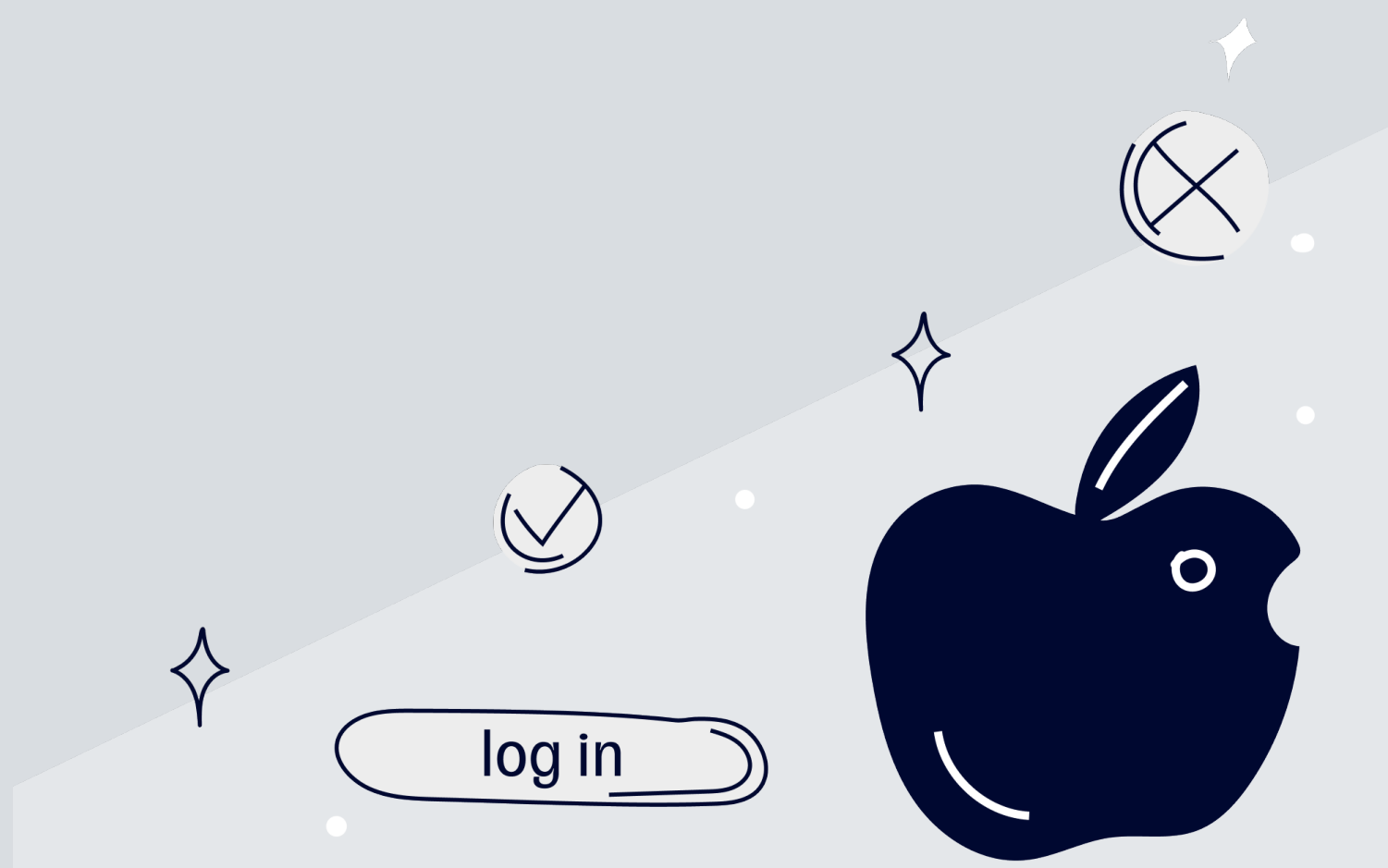
.after(Date)

Система запрашивает следующий массив entry после определенной даты;



.never

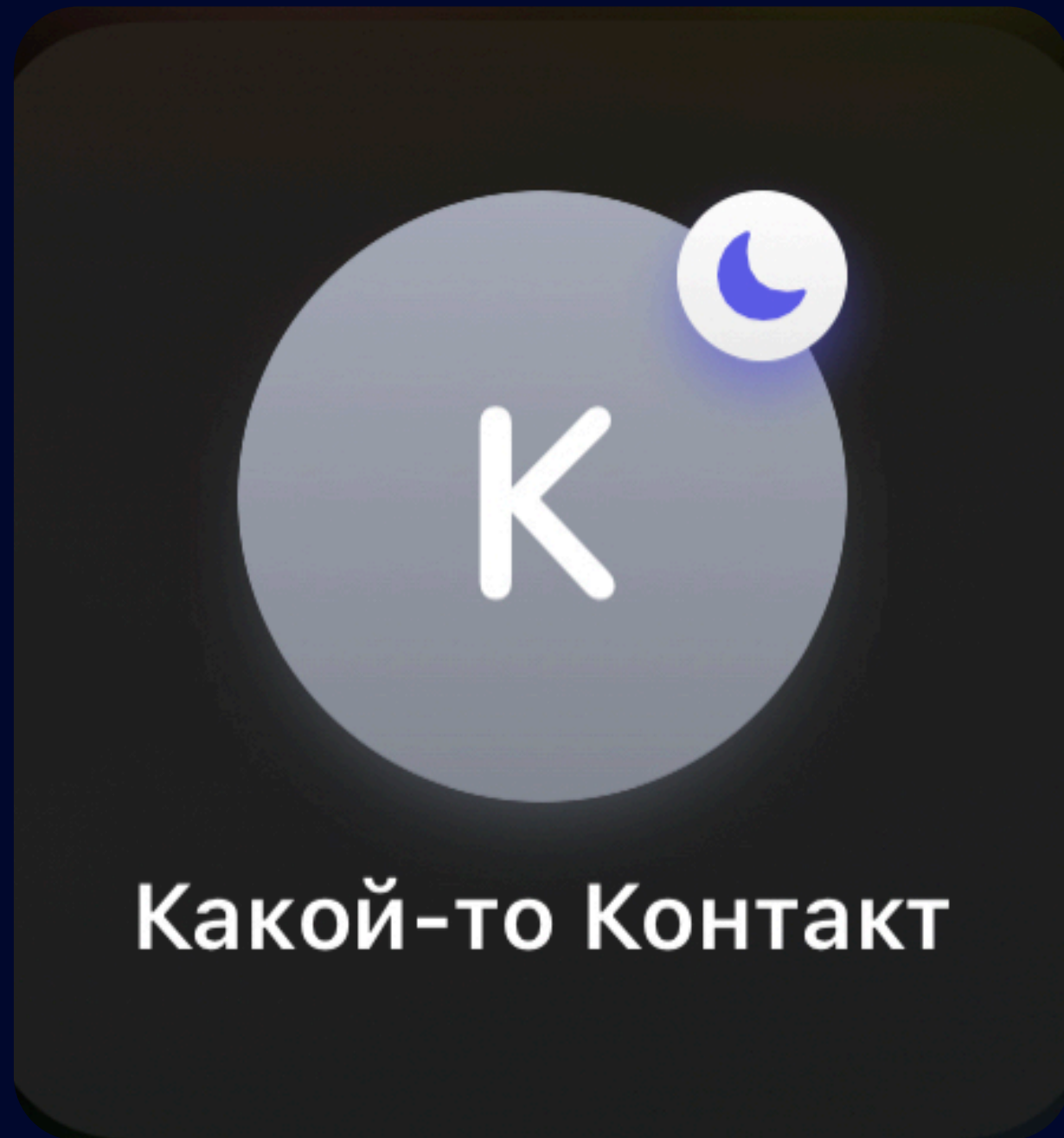
Система не запрашивает следующий массив entry, пока ее не попросят об этом принудительно.



Опыт разработки

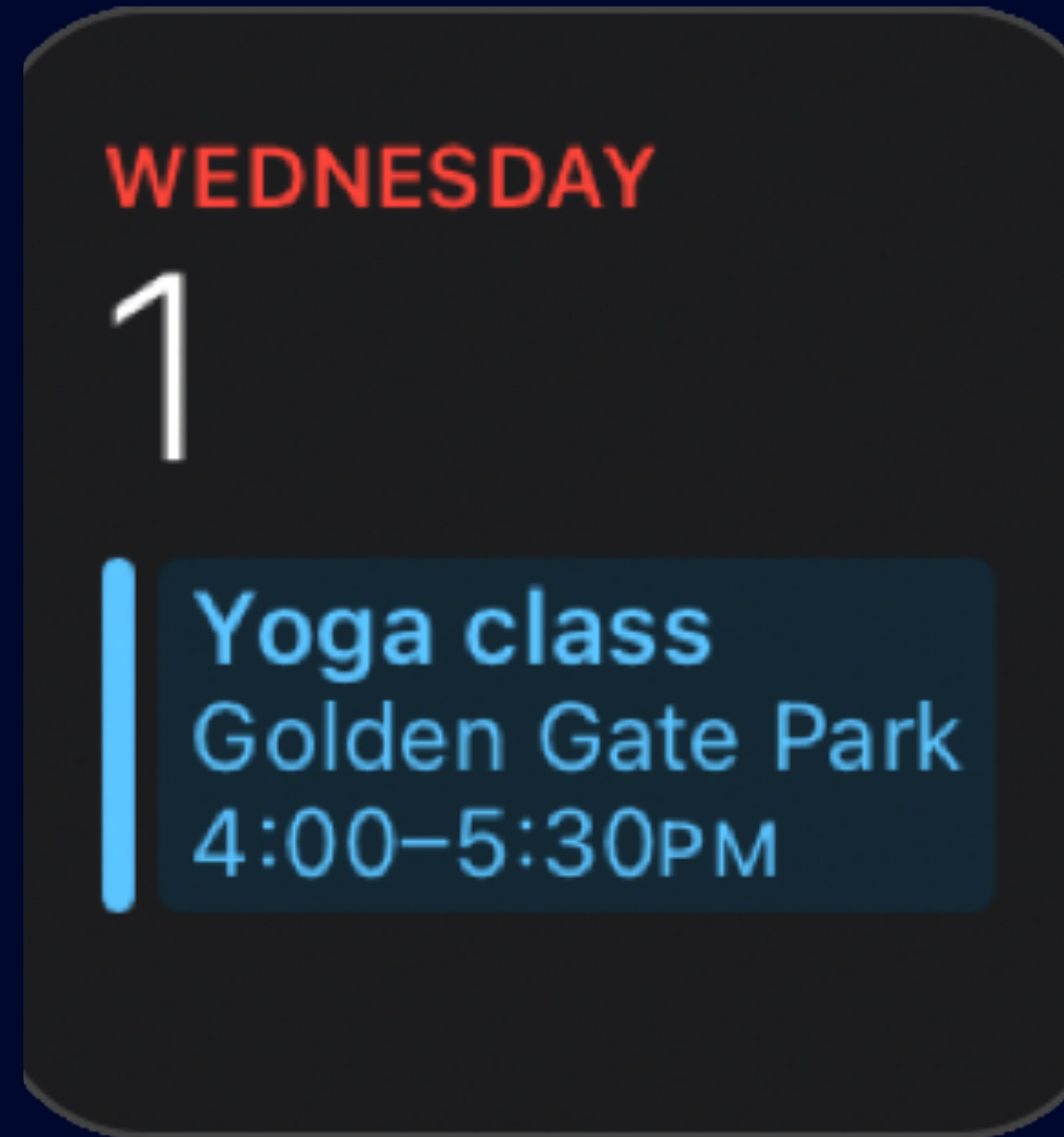


Виджеты



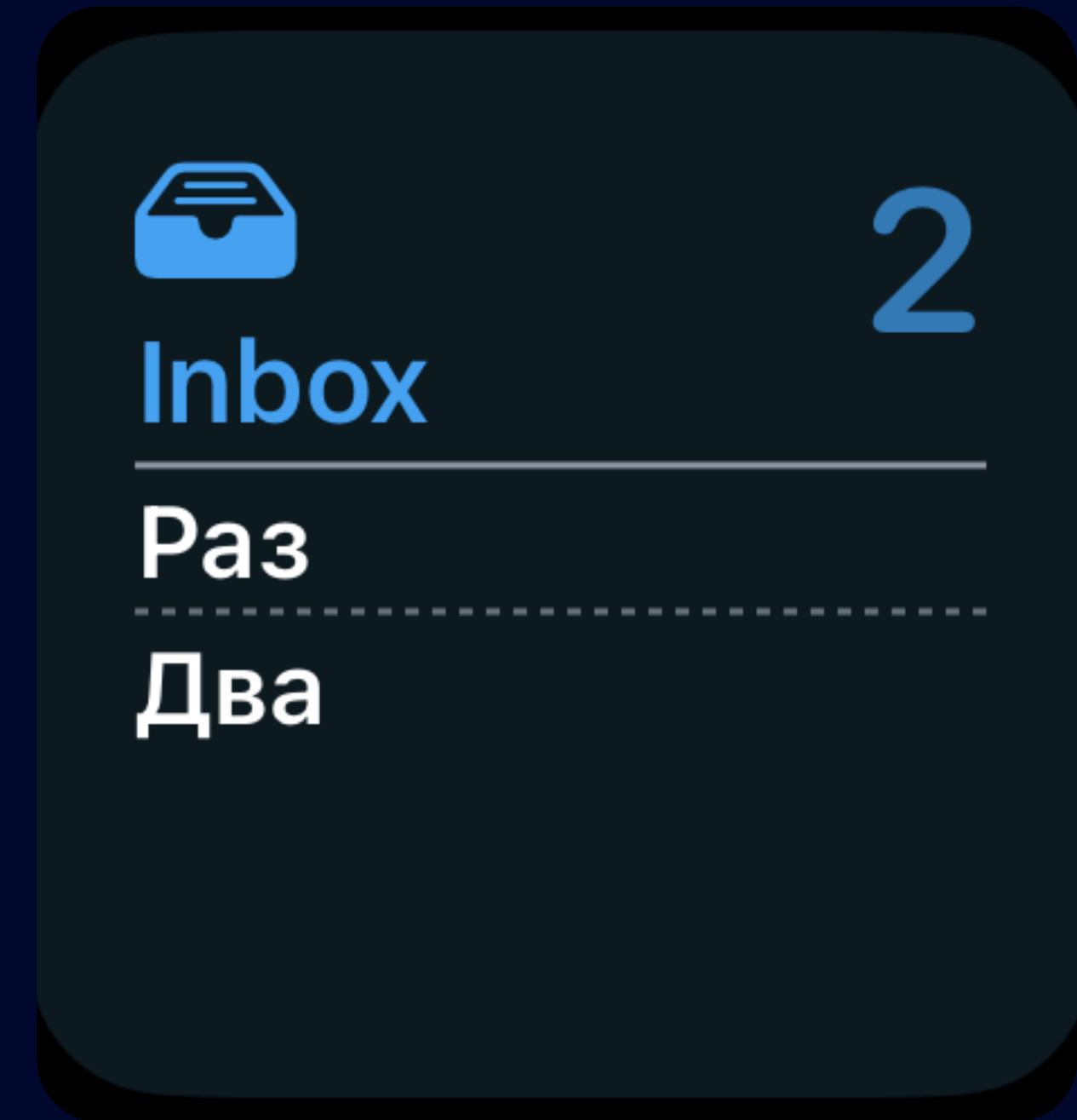
Виджет-deerlink

Ведет в некоторую точку приложения



Информационный

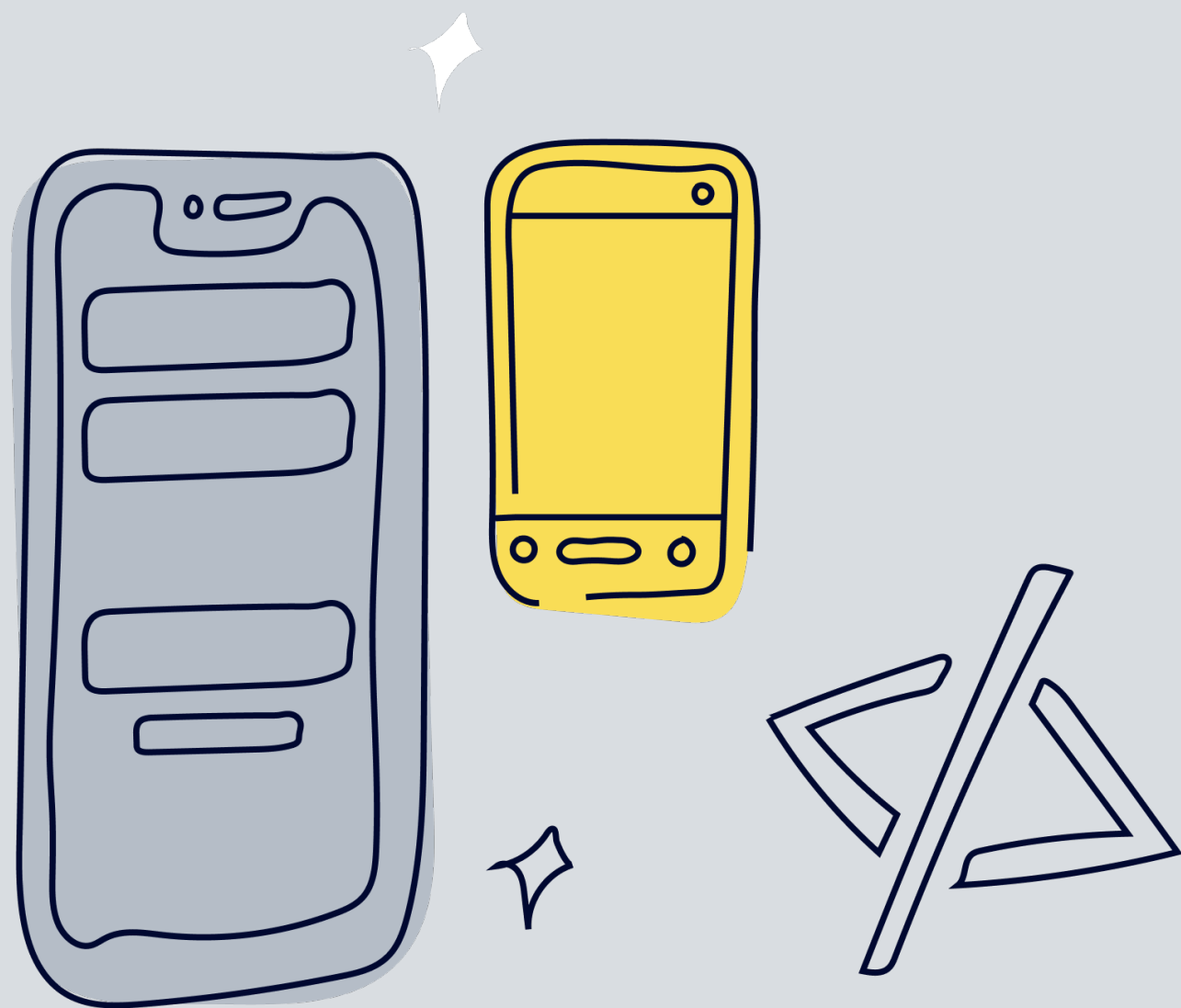
Показывает полезную и актуальную информацию



Интерактивный

Как информационный, но позволяет взаимодействовать

План разработки ВИДЖЕТОВ- ДИПЛИКОВ



Реализовать UI

Только SwiftUI, уже сколько лет-то прошло!



Договориться о диплинке

Виджет использует старый-добрый механизм для перехода на нужный экран



Подготовить таймлайн

Один entry, стратегия обновления `.never`

widgetURL

- ✓ **Охватывает весь виджет**
Без выбора элементов, не больше одного
- ✓ **Доступен всем виджетам**
Независимо от его размера
- ✓ **URL опционален**
Если его нет, то откроется текущий экран приложения

```
struct SomeWidget: View {
  let title: String
  let icon: Image

  var body: some View {
    ZStack {
      Color.red
      HStack {
        Spacer()
        VStack {
          Image(systemName: "folder")
            .resizable()
            .scaledToFit()
          Spacer()
        }
      }
      VStack {
        Spacer()
        HStack { /* дизайн */ }
      }
    }
    .widgetURL(URL(string: "ваш диплинк"))
  }
}
```

```

struct SomeWidget: View {
    var body: some View {
        ZStack {
            Color.red
            HStack {
                Spacer()
                VStack {
                    Image(systemName: "folder")
                    .linkIfPossible(string: "ваш диплинк")
                    Text("Привет! Как дела?")
                }
            }
        }
        VStack {
            Spacer()
            VStack {
                Image(systemName: "xmark")
                .linkIfPossible(string: "ваш диплинк")
            }
        }
    }
}

extension View {
    @ViewBuilder
    func linkIfPossible(string: String) → some View {
        if let url = URL(string: string) {
            Link(destination: url) {
                self
            }
        } else {
            self
        }
    }
}

```

Link



Можно повесить на отдельный элемент

И переходить с одного виджета на разные экраны



Доступен не всем виджетам

Не для `.systemSmall`



URL неопционален

И точно должен существовать на момент перехода

Виджеты-диплинки в Мобильном Банке



Перевести
по телефону

Перевод по телефону



Сканировать
QR или фото

Оплата по QR

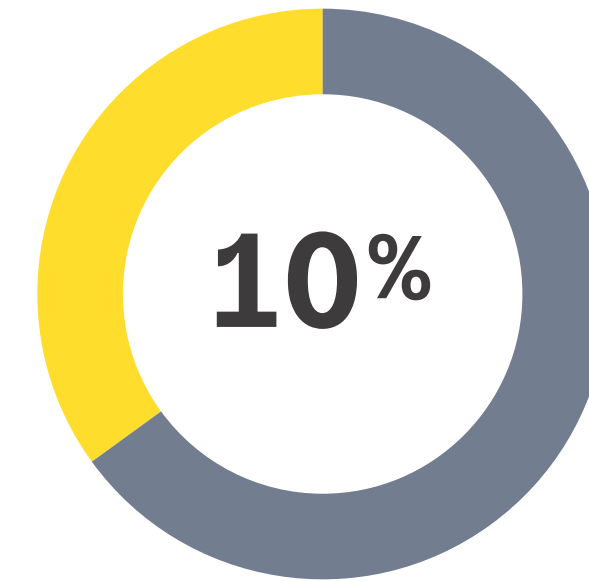


Заправиться
с кэшбэком

Топливо

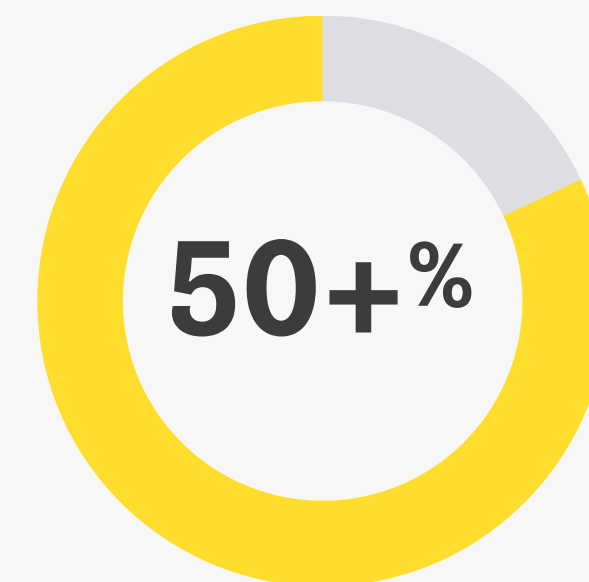
Чем помогают виджеты-диплинки?

Польза таких виджетов - сокращение времени до использования фичи.



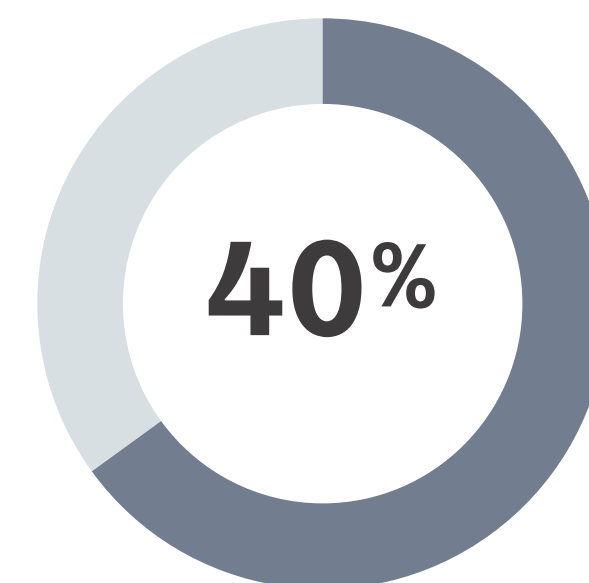
Процент всех клиентов

Виджеты - не самая используемая часть приложения...



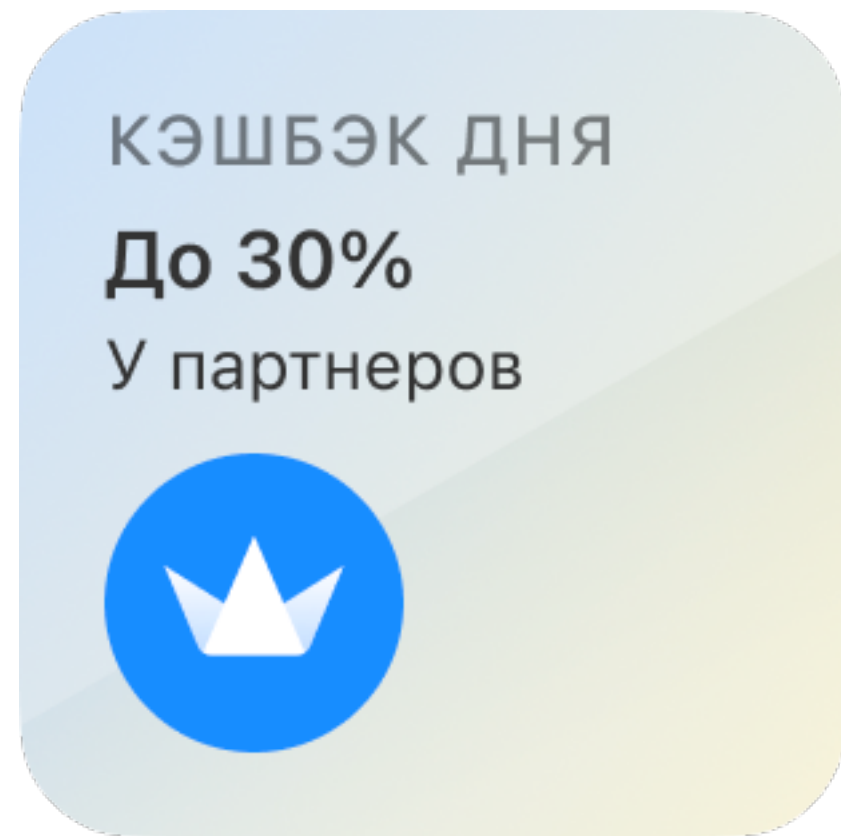
Удержание

...но больше половины продолжают пользоваться виджетом

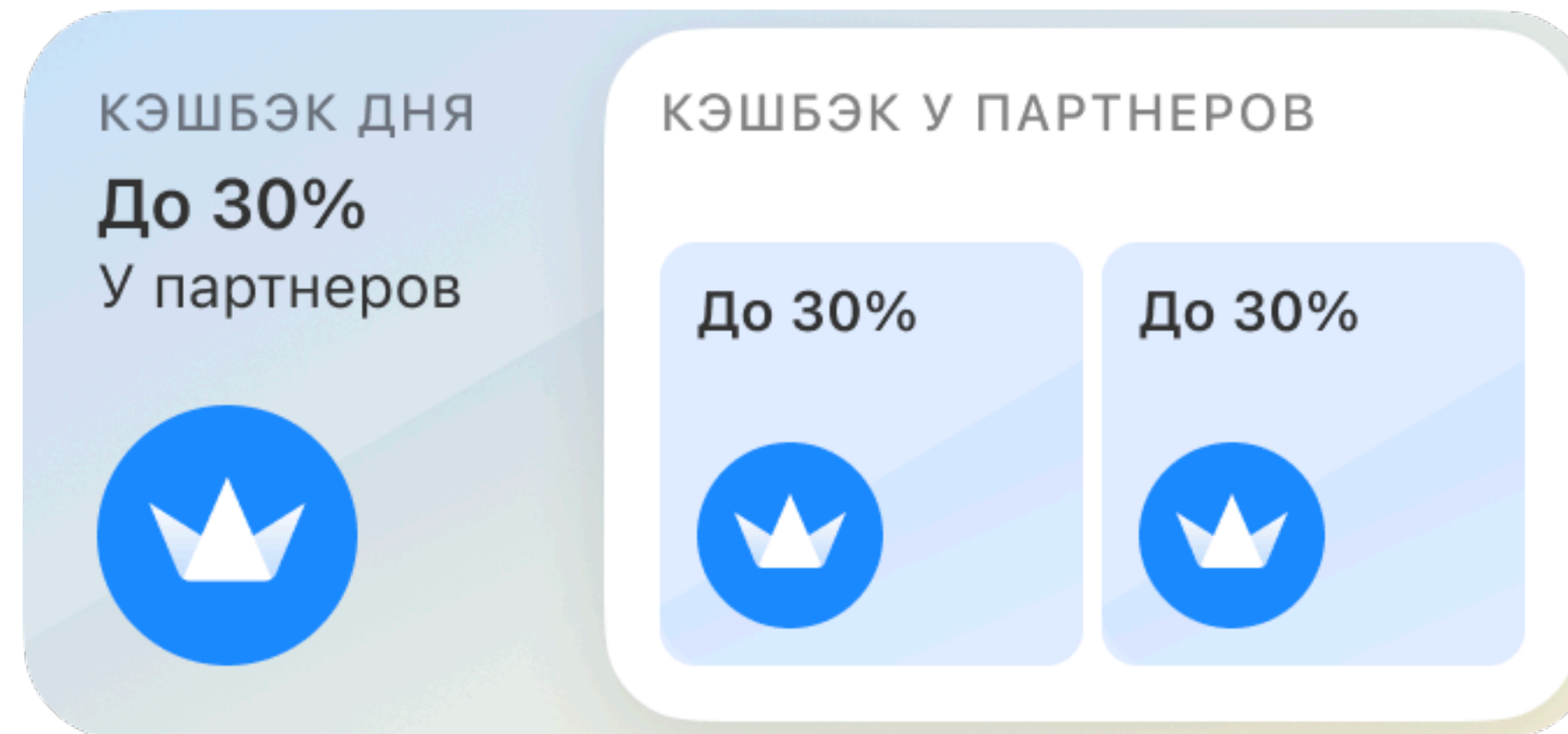


Экономия времени

На виджете «Топливо» удалось сократить время использования заправки



Кэшбэк дня



Кэшбэк у партнёров

План-капкан

01

Информация: как хранить?
Связь приложение-виджет?

02

Виджет умеет сам ходить в
сеть?

03

Как можно сделать
ничёшный UI?

04

Доступен ли механизм
фича-тогглов?

05

Можно ли обработать
аналитику действий?

06

Как организовать код в
проекте?

07

Можно ли прикрутить тесты
к виджетам?

План-капкан

01

**Информация: как хранить?
Связь приложение-виджет?**

02

Виджет умеет сам ходить в сеть?

03

Как можно сделать
ничёшный UI?

04

Доступен ли механизм
фича-тогглов?

05

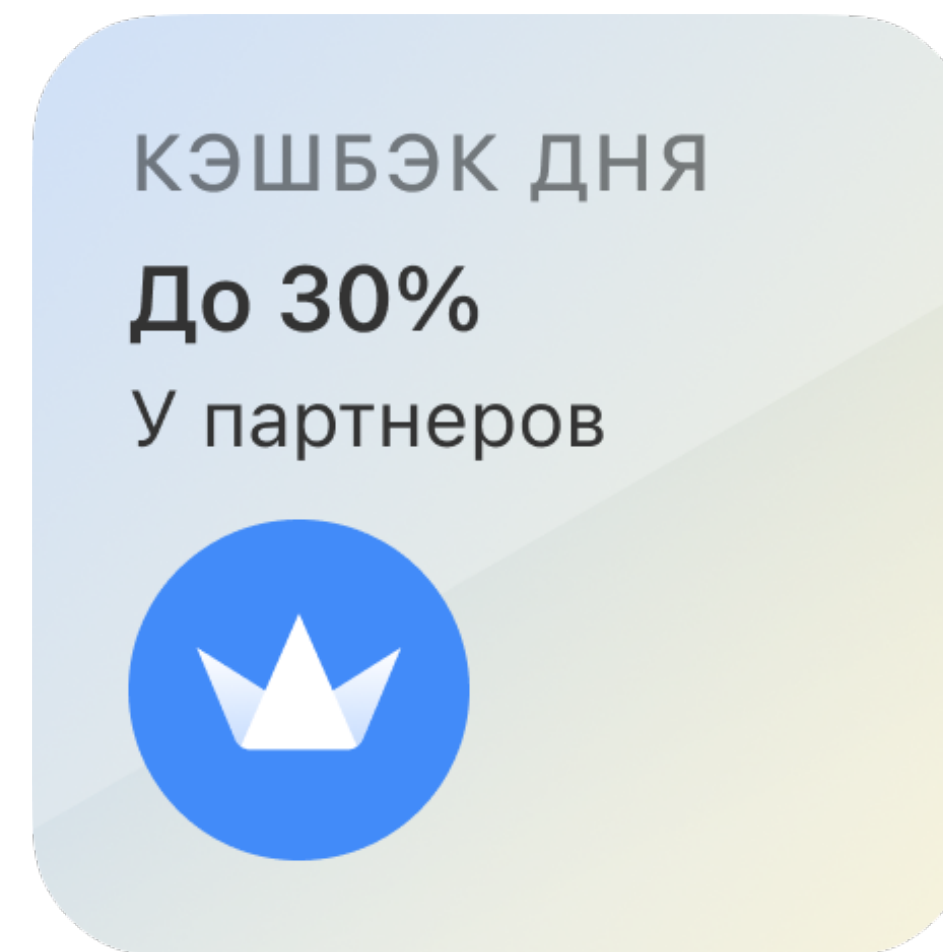
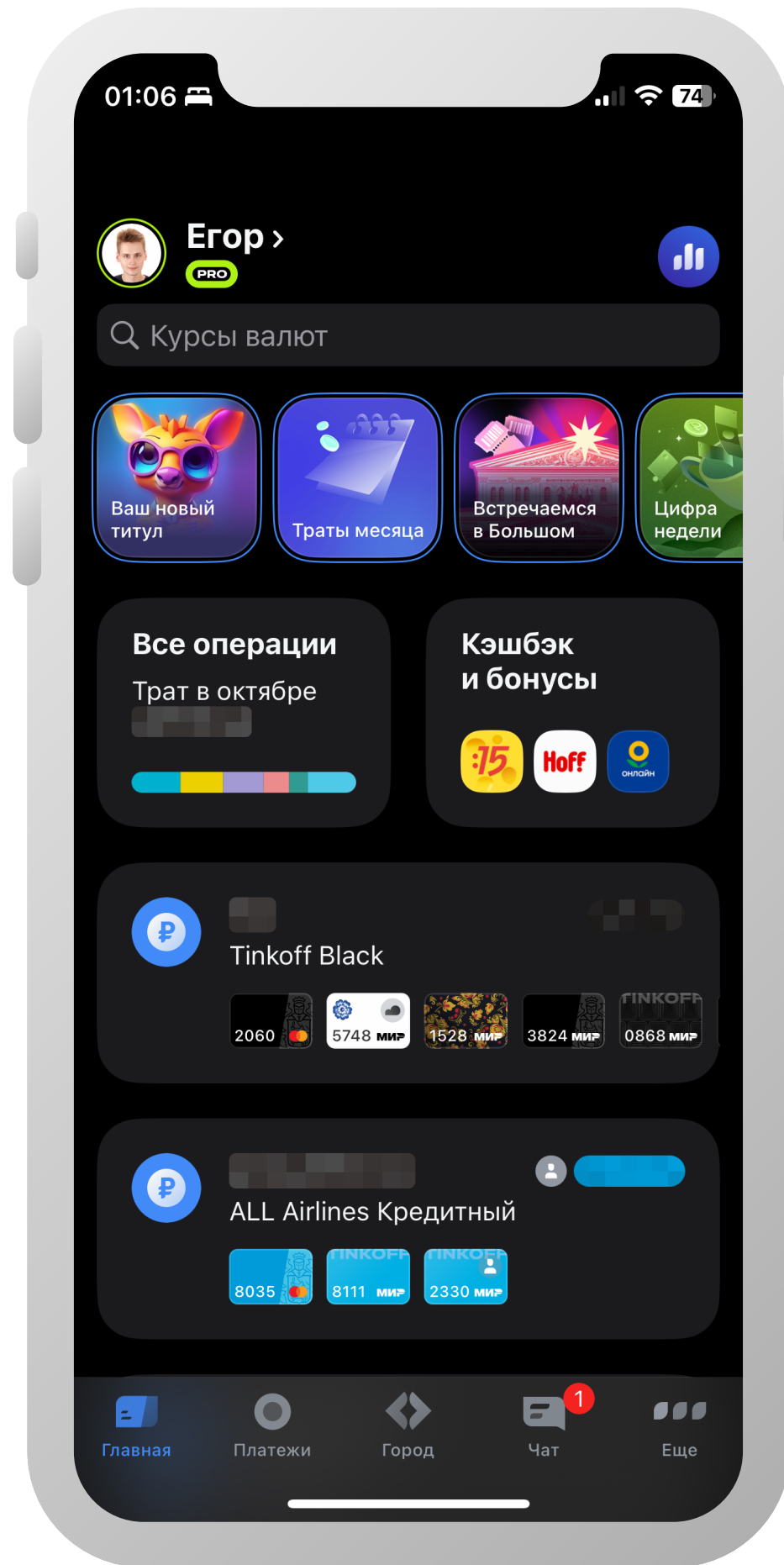
Можно ли обработать
аналитику действий?

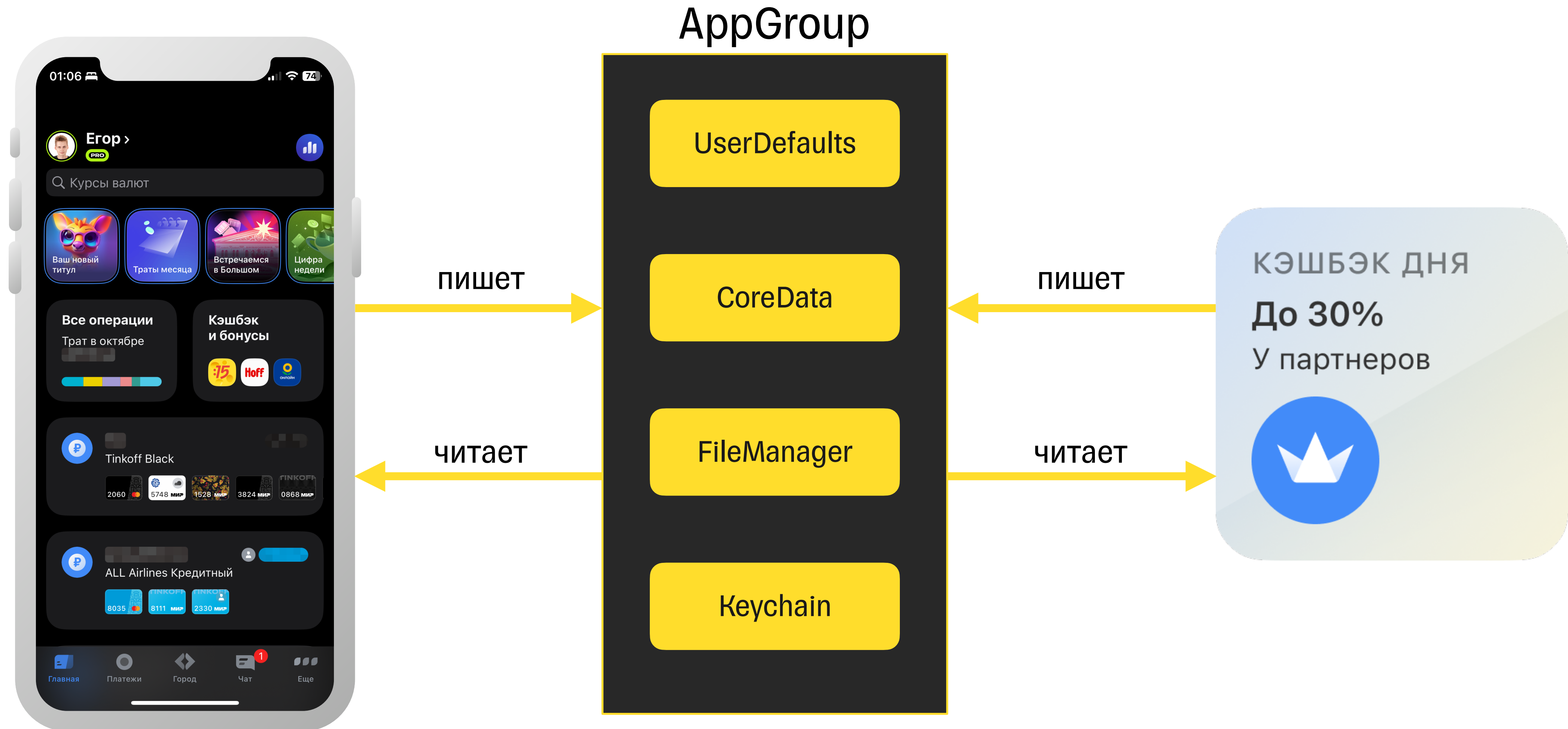
06

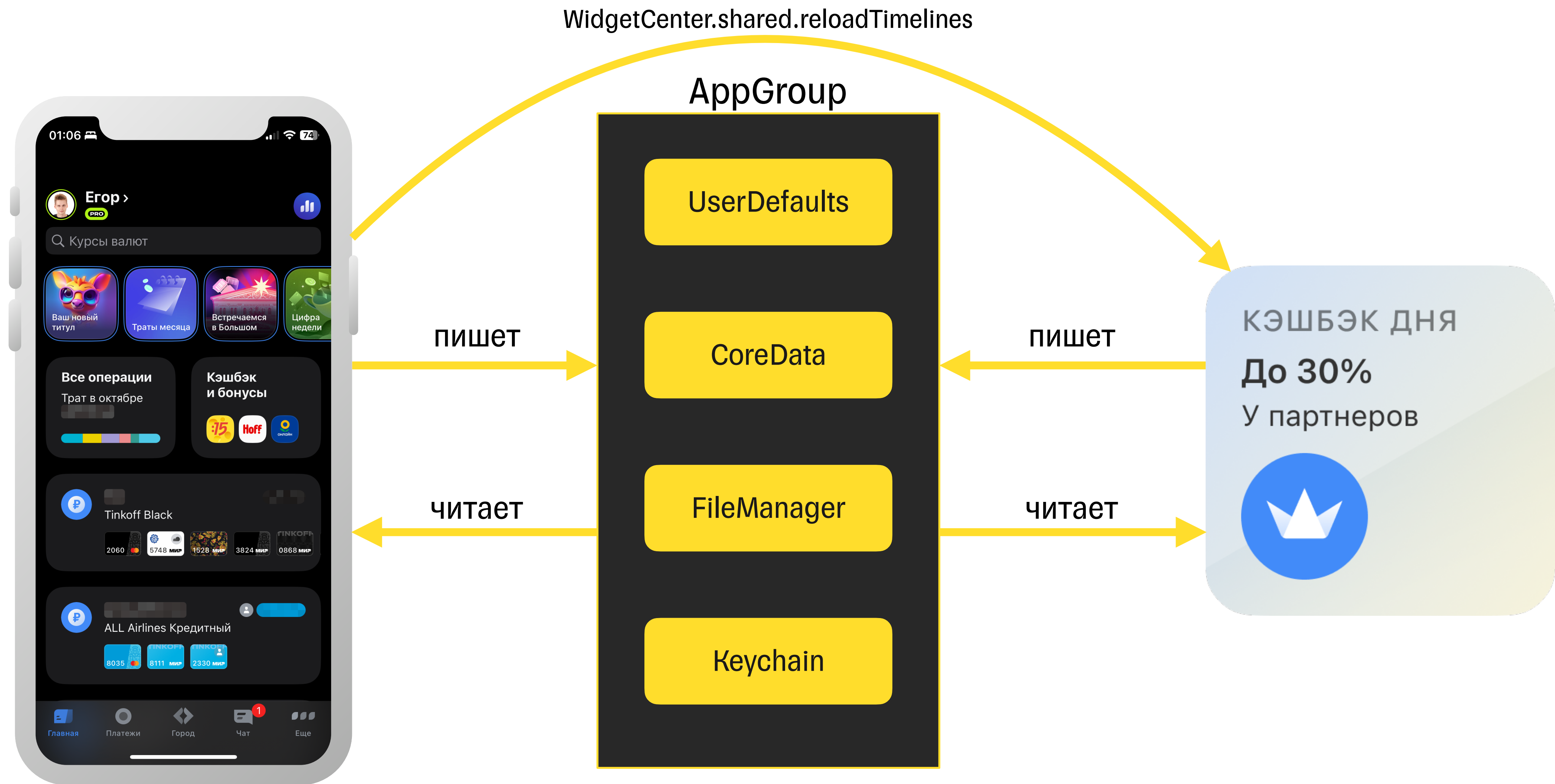
Как организовать код в
проекте?

07

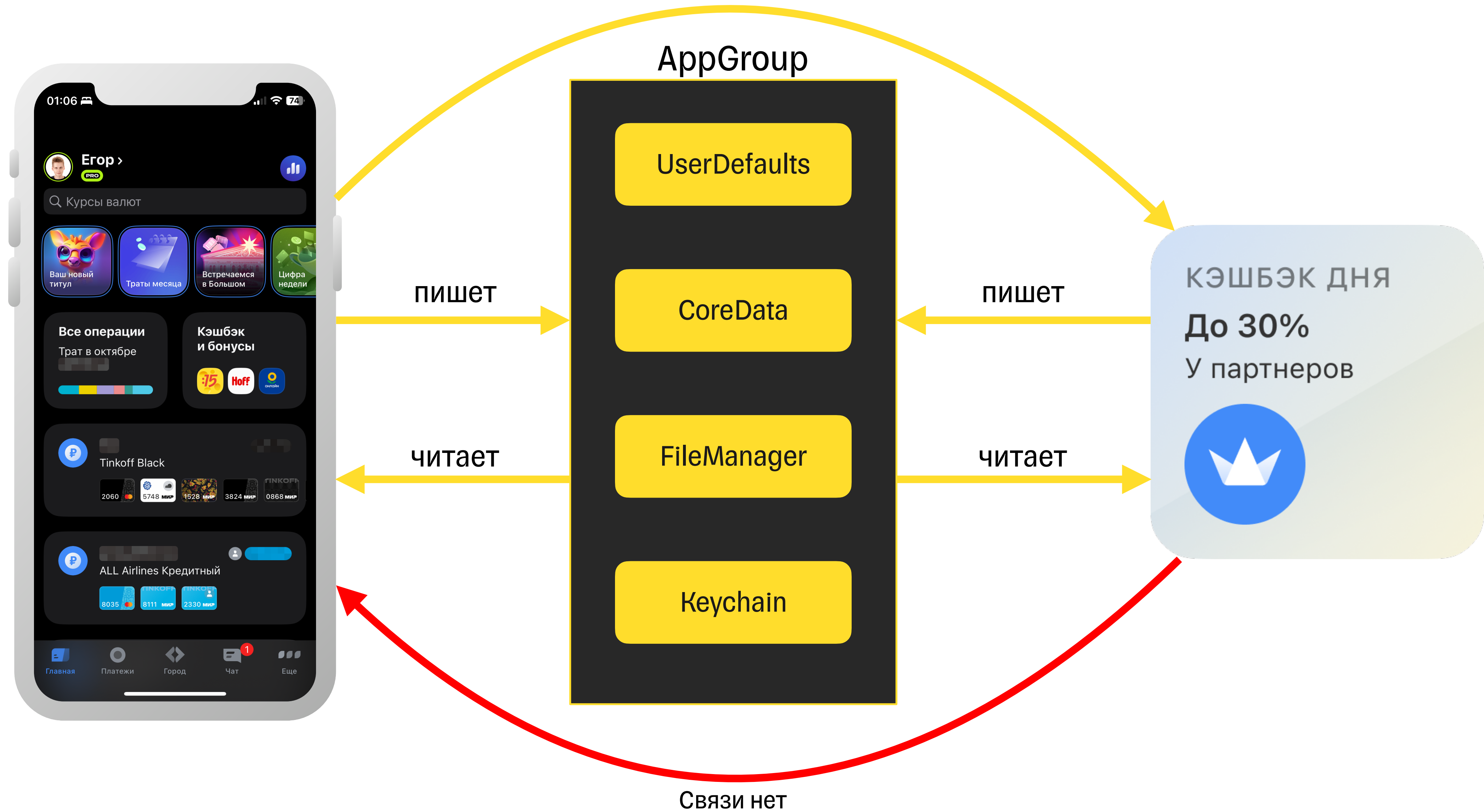
Можно ли прикрутить тесты
к виджетам?







WidgetCenter.shared.reloadTimelines



План-капкан

01

Информация: как хранить?
Связь приложение-виджет?

02

**Виджет умеет сам
ходить в сеть?**

03

Как можно сделать
ничёшный UI?

04

Доступен ли механизм
фича-тогглов?

05

Можно ли обработать
аналитику действий?

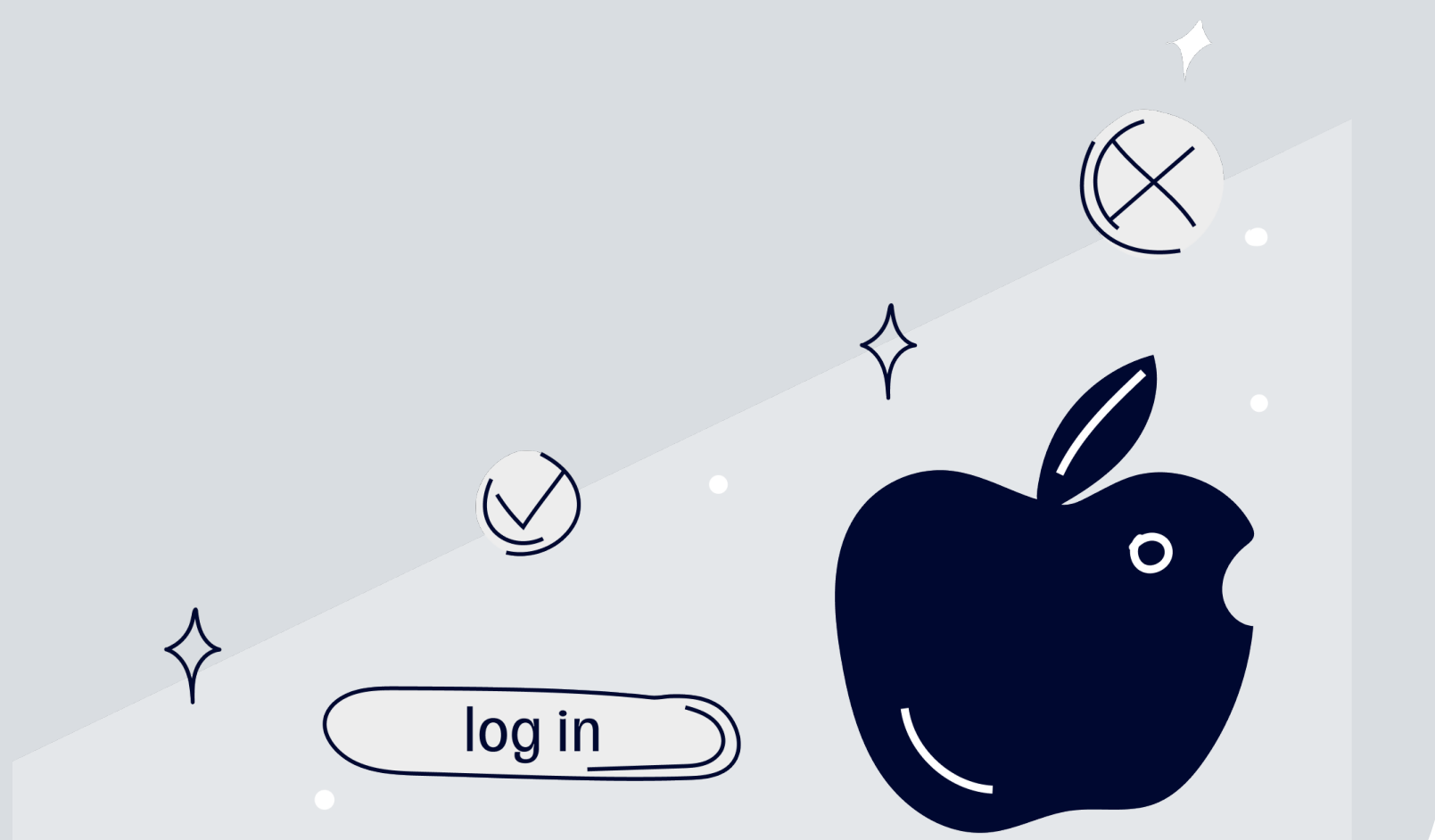
06

Как организовать код в
проекте?

07

Можно ли прикрутить тесты
к виджетам?

Возможности сетевого слоя



Никакого «кастома»

В виджетах доступен URLSession и всего его стандартные возможности



Виджет сам ходит в сеть

И ему абсолютно не нужно основное приложение



Доступна background загрузка

Виджет может пробуждаться и грузить данные, даже если основное приложение спит

Мобильный банк – многомодульное приложение. Особенности?



Линковка

Статическая, а значит любой модуль вшивается на стадии компиляции



Возможности

Проект развитый и есть куча модулей, решающие любые задачи



Непредсказуемость

Неизвестно, какие еще модули затягиваются транзитивно

Эксперимент с размером расширения виджета

На сколько вырастет размер?

Подготовка

Для сравнения два модуля:

- Сетевой слой чисто под виджеты;
- Базовый сетевой слой приложения.

Измерение

Сборка конечного приложения с модулями:

- Сетевой слой для виджетов ~ **2.67 мегабайт**;
- Сетевой слой приложения ~ **2.86 мегабайт**.

Сравнение

Насколько выросло расширение?

- Разница ~ **0.19 мегабайт** на один модуль;
- В процентах ~ **7 процентов** на один модуль.



Выводы эксперимента

Независимость

Виджет не должен
зависеть от
модулей основного
приложения

Ничего лишнего

Подключение
модулей строго в
рамках
необходимого

Ожидание

.after(Date)

КЭШБЭК ДНЯ
До 30%
У партнеров



загружает

Сеть

Ожидание

.after(Date)

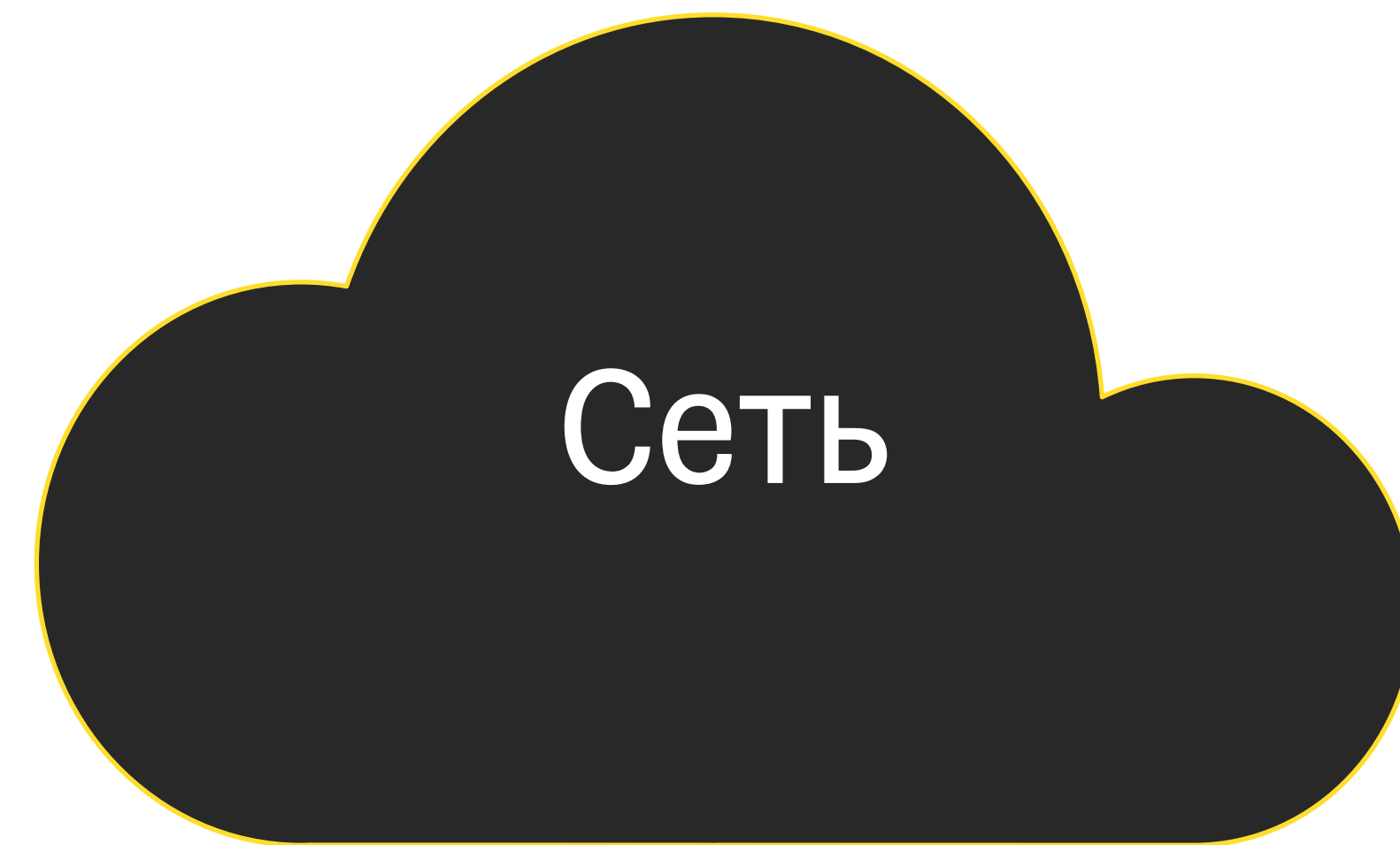
КЭШБЭК ДНЯ
До 30%
У партнеров



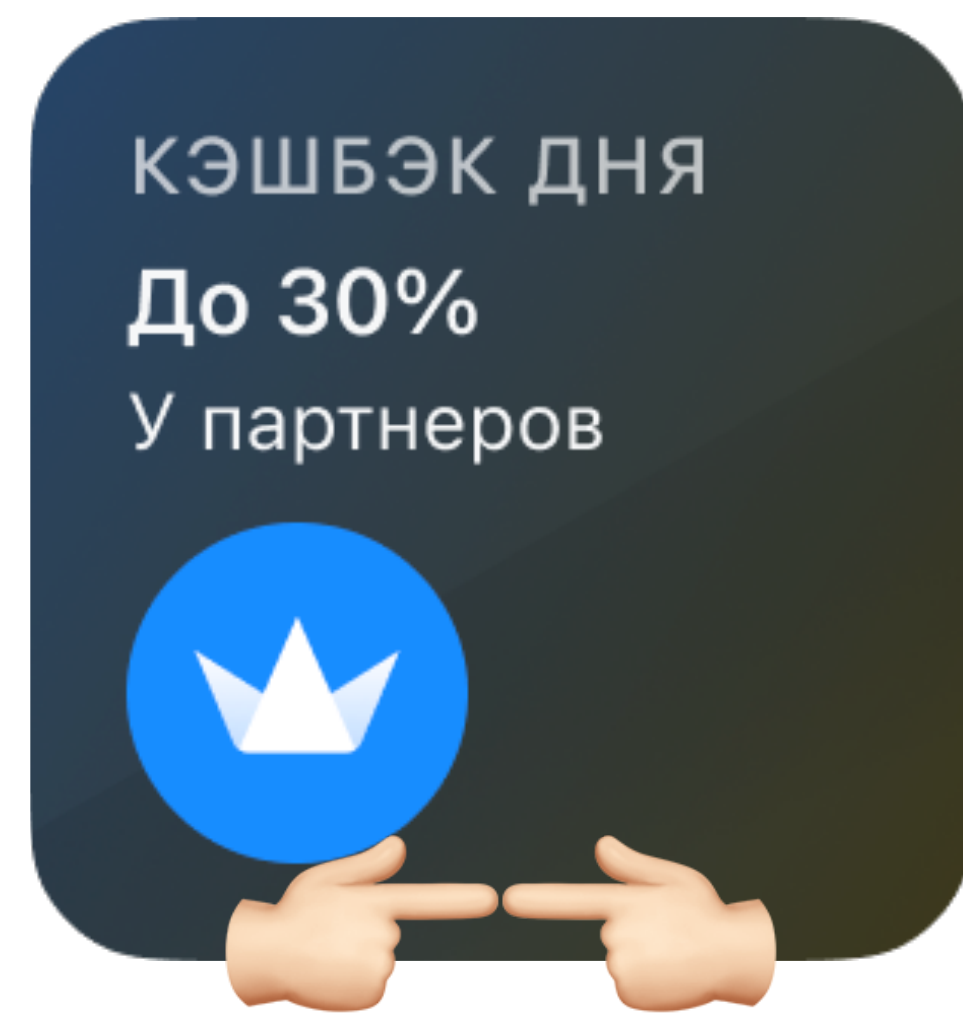
отдаёт

Сеть

Ожидание




Ожидание



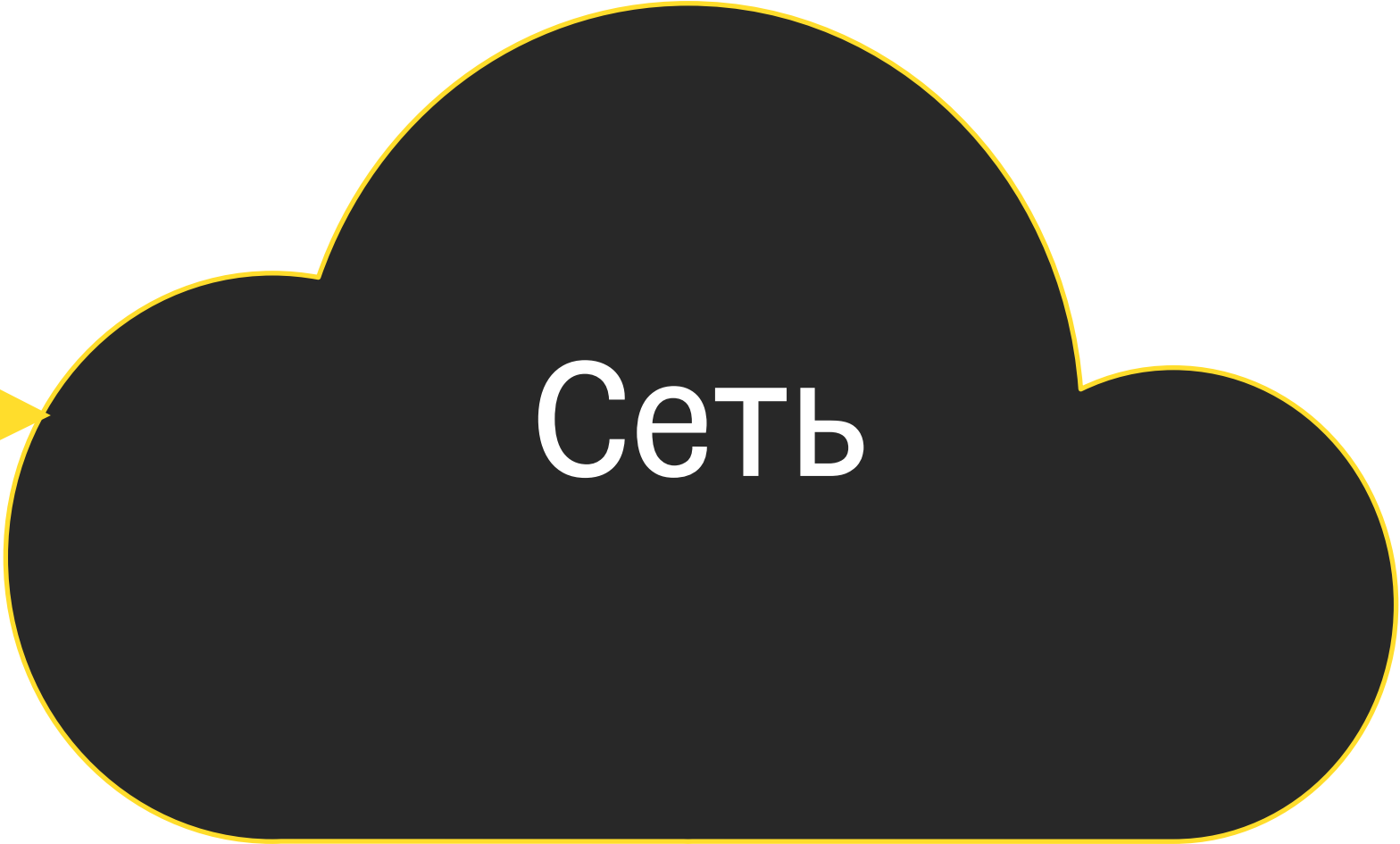
Реальность

`.after(Date)`

КЭШБЭК ДНЯ
До 30%
У партнеров



загружает



Реальность

.after(Date)

КЭШБЭК ДНЯ
До 30%
У партнеров



загружает



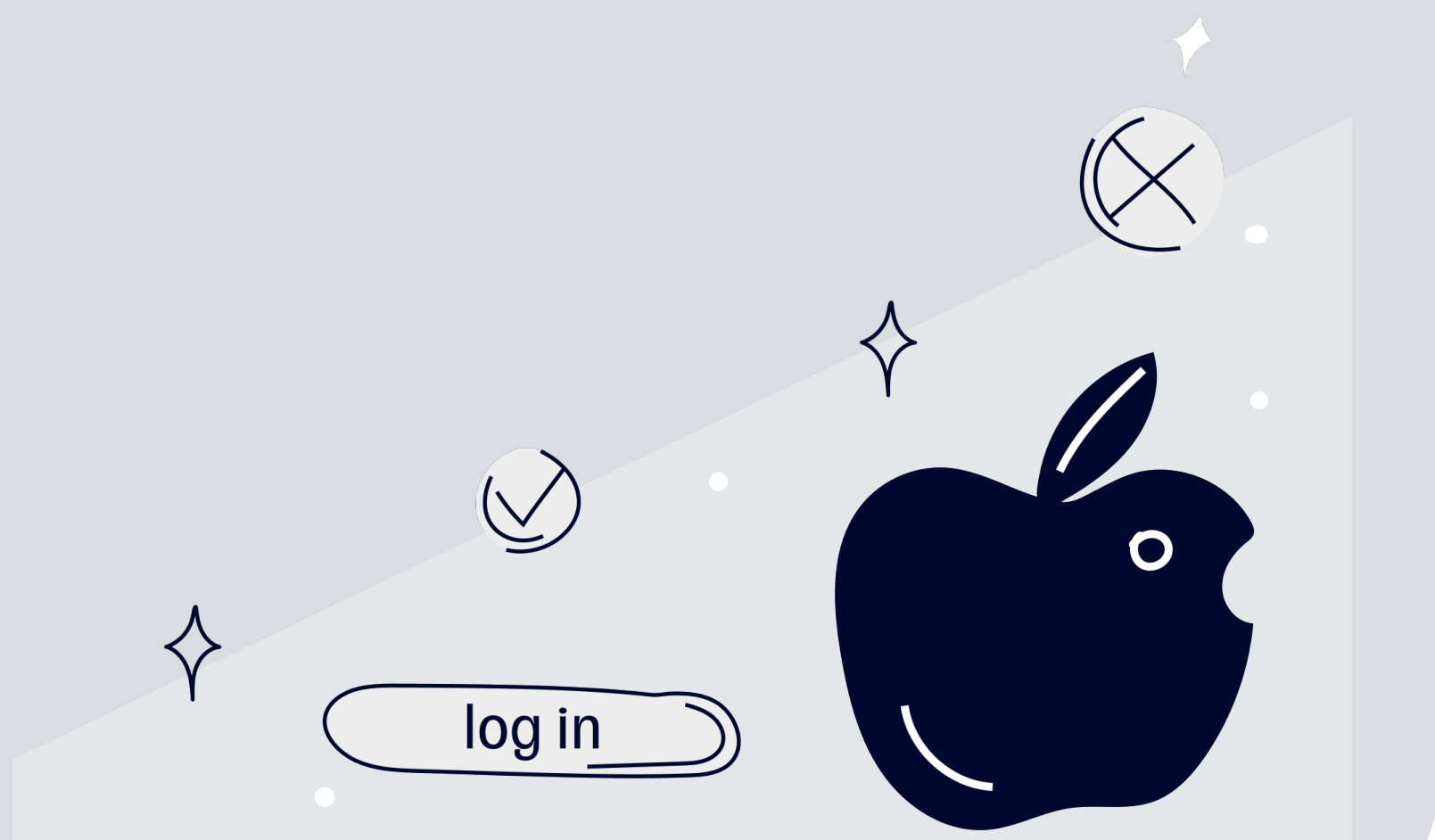
Реальность



Реальность



Как сделать правильную загрузку?



Background запрос

Виджет должен пробудиться, выполнить запрос и обновить виджет



Немного дополнительной работы

Адаптировать функцию `onBackgroundURLSessionEvents(matching:_:)` в конфигурации виджета



Не забыть понастроить проект

Добавить capability Background Fetch в настройках проекта

План-капкан

01

Информация: как хранить?
Связь приложение-виджет?

02

Виджет умеет сам ходить в
сеть?

03

**Как можно сделать
ничёшный UI?**

04

Доступен ли механизм
фича-тогглов?

05

Можно ли обработать
аналитику действий?

06

Как организовать код в
проекте?

07

Можно ли прикрутить тесты
к виджетам?

КЭШБЭК ДНЯ

До 30%

У партнеров



КЭШБЭК ДНЯ

Карусель

Кэшбэк 10%



Сегодня нет
кэшбэка дня

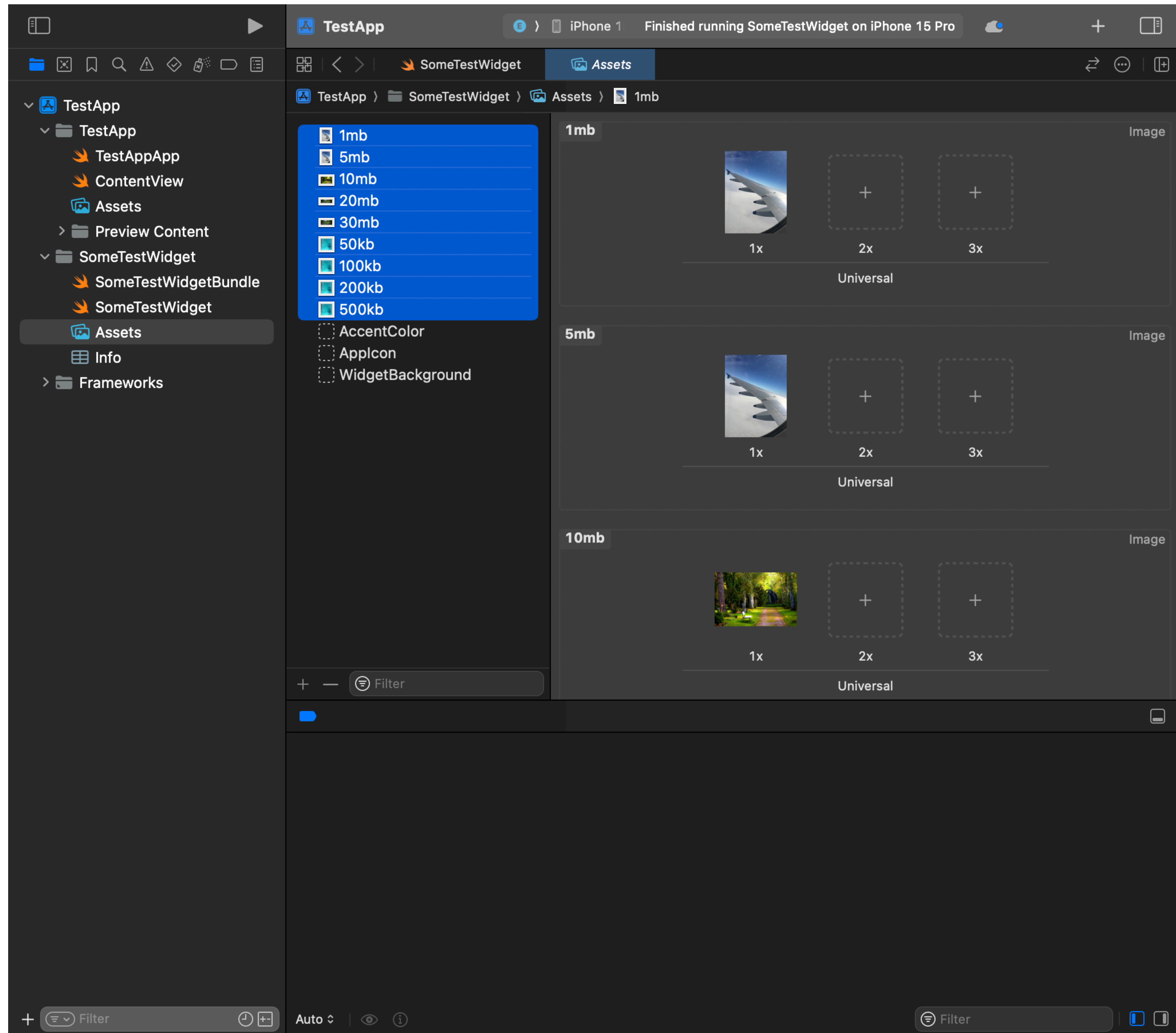
Карта для вас

С кэшбэком
до 30%



Проверьте соединение
с интернетом или войдите
в приложение

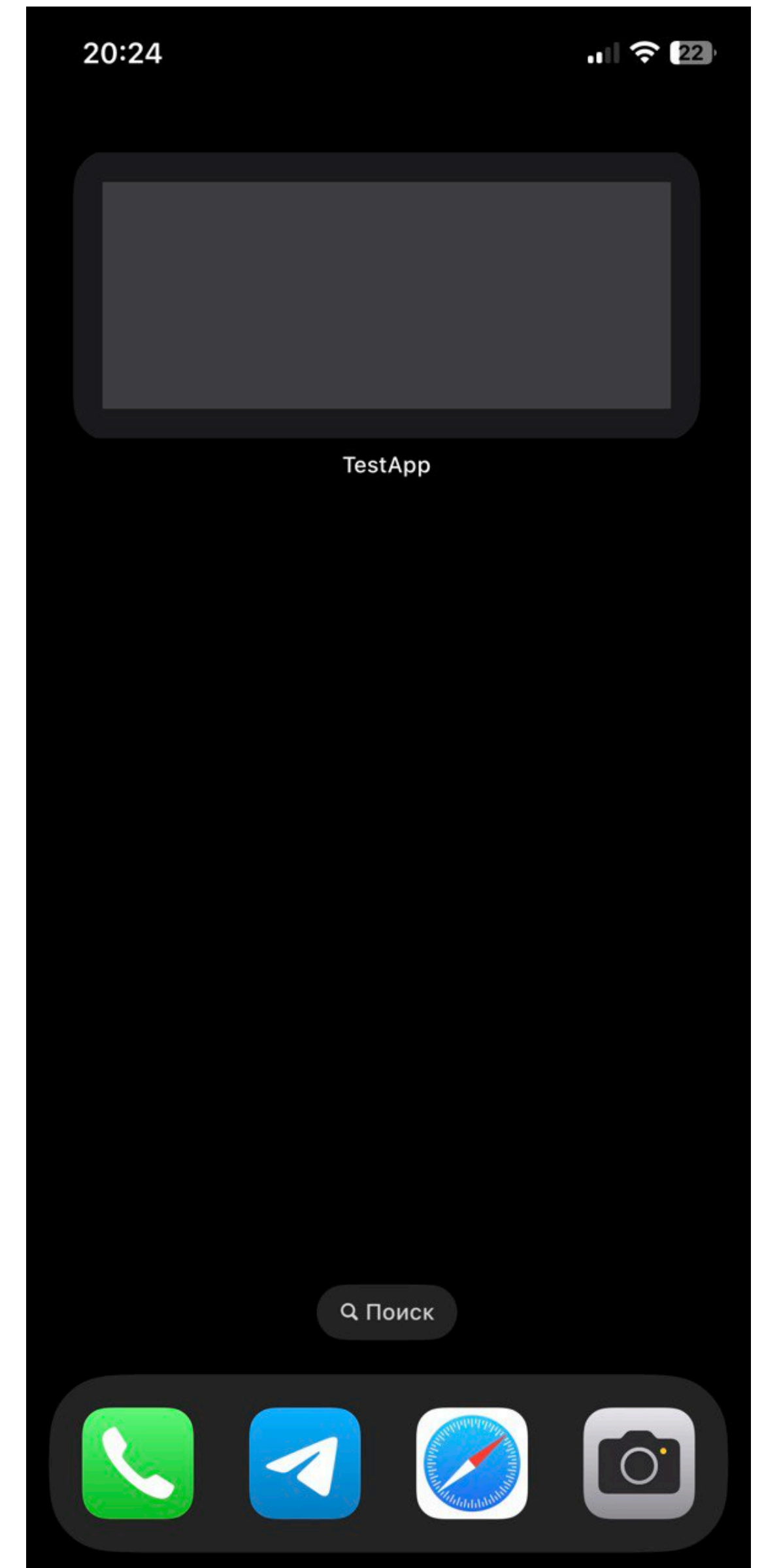
Войти



```
35 }
36
37 struct SimpleEntry: TimelineEntry {
38     let date: Date
39     let emoji: String
40 }
41
42 struct SomeTestWidgetEntryView : View {
43     var entry: Provider.Entry
44
45     var body: some View {
46         Image("30mb").resizable().scaledToFit()
47     }
48 }
49
50 struct SomeTestWidget: Widget {
51     let kind: String = "SomeTestWidget"
52
53     var body: some WidgetConfiguration {
54         StaticConfiguration(kind: kind, provider: Provider()) { entry in
55             if #available(iOS 17.0, *) {
56                 SomeTestWidgetEntryView(entry: entry)
57                     .containerBackground(.fill.tertiary, for: .widget)
58             } else {
59                 SomeTestWidgetEntryView(entry: entry)
60                     .padding()
61                     .background()
62             }
63         }
64         .configurationDisplayName("Mv Widget")

```

2023-10-21 22:15:37.516405+0700 SomeTestWidgetExtension[44432:4802768] [archiving]
Widget archival failed due to image being too large [1] - (13583, 5417).
2023-10-21 22:15:37.549221+0700 SomeTestWidgetExtension[44432:4802768] [archiving]
Widget archival failed due to image being too large [1] - (13583, 5417).



Untitled3

iPhone 15 Pro (17.0.1) > SomeTestWidgetExtension.appex Run 1 of 1 | 00:00:00

Track Filter: All Tracks Duplicate

Allocations: All Heap & Anonymous VM

Leaks: Leak Checks

Statistics

Graph	Category	Persistent Bytes	# Persistent	# Transient	Total Bytes	# Total	Persistent/Total Bytes
<input checked="" type="checkbox"/>	All Heap & Anonymou...	6,03 MiB	2 743	6 969	6,85 MiB	9 712	
<input type="checkbox"/>	All Heap Allocations	328,09 KiB	2 741	6 969	1,15 MiB	9 710	
<input type="checkbox"/>	All Anonymous VM	5,71 MiB	2	0	5,71 MiB	2	
<input type="checkbox"/>	Malloc 48 Bytes	13,31 KiB	284	1 670	91,59 KiB	1 954	
<input type="checkbox"/>	Malloc 32 Bytes	20,00 KiB	640	1 280	60,00 KiB	1 920	
<input type="checkbox"/>	Malloc 64 Bytes	31,06 KiB	497	1 162	103,69 KiB	1 659	
<input type="checkbox"/>	Malloc 16 Bytes	544 Bytes	34	657	10,80 KiB	691	
<input type="checkbox"/>	CFString (immutable)	5,61 KiB	95	365	21,34 KiB	460	
<input type="checkbox"/>	Malloc 80 Bytes	3,28 KiB	42	406	35,00 KiB	448	
<input type="checkbox"/>	Malloc 96 Bytes	1,50 KiB	16	393	38,34 KiB	409	
<input type="checkbox"/>	Malloc 128 Bytes	26,50 KiB	212	53	33,12 KiB	265	
<input type="checkbox"/>	Malloc 256 Bytes	44,00 KiB	176	8	46,00 KiB	184	
<input type="checkbox"/>	Malloc 272 Bytes	39,05 KiB	147	4	40,11 KiB	151	
<input type="checkbox"/>	__NSArrayM	240 Bytes	5	135	6,56 KiB	140	
<input type="checkbox"/>	__NSArrayI	1,25 KiB	32	66	4,00 KiB	98	
<input type="checkbox"/>	__NSMallocBlock__	5,12 KiB	91	1	5,17 KiB	92	
<input type="checkbox"/>	Malloc 288 Bytes	18,00 KiB	64	15	22,22 KiB	79	
<input type="checkbox"/>	__NSSingleObjectSetI	784 Bytes	49	6	880 Bytes	55	
<input type="checkbox"/>	Malloc 112 Bytes	1,31 KiB	12	36	5,25 KiB	48	
<input type="checkbox"/>	CFBasicHash (value-stor...	1,30 KiB	9	27	5,36 KiB	36	
<input type="checkbox"/>	Malloc 192 Bytes	192 Bytes	1	34	6,56 KiB	35	
<input type="checkbox"/>	__NSSingleObjectArrayI	400 Bytes	25	8	528 Bytes	33	
<input type="checkbox"/>	NSMethodSignature	1,03 KiB	33	0	1,03 KiB	33	
<input type="checkbox"/>	CFBasicHash (key-store)	1,09 KiB	6	22	4,89 KiB	28	
<input type="checkbox"/>	OS_os_log	1,75 KiB	28	0	1,75 KiB	28	
<input type="checkbox"/>	NSSet	1,52 KiB	26	1	1,64 KiB	27	

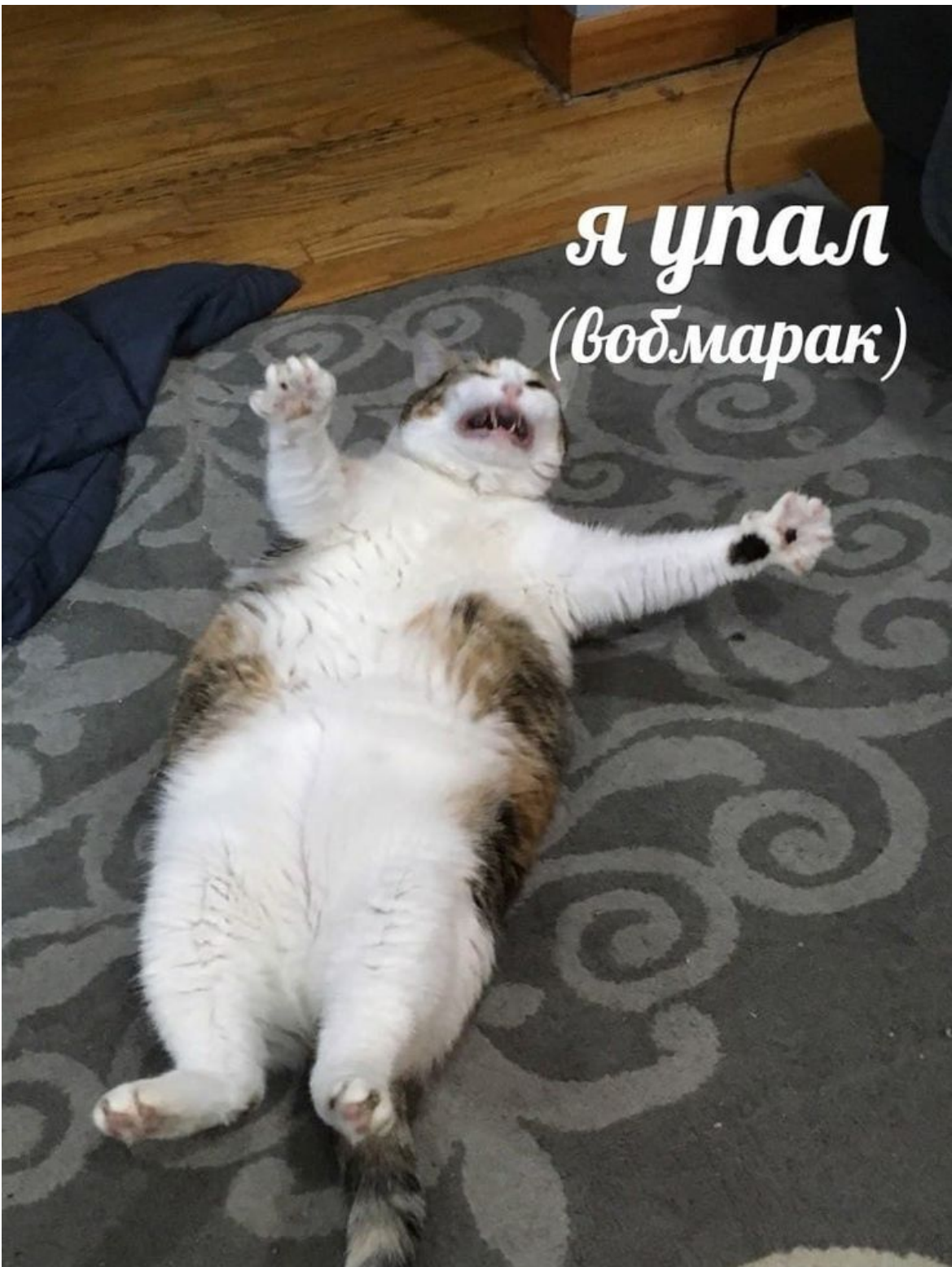
Detail Filter: Mark Generation Created & Persistent All Heap & Anonymous VM


```
// MARK: - View

var body: some View {
    let backgroundView2 = BackgroundView2(
        darkColor: LinearGradient(
            stops: [
                Gradient.Stop(color: Color(red: 0.23, green: 0.31, blue: 0.41), location: 0.00),
                Gradient.Stop(color: Color(red: 0.35, green: 0.36, blue: 0.28), location: 1.00)
            ],
            startPoint: .topLeading,
            endPoint: .bottomTrailing
        ),
        lightColor: Color.white
    )
    .cornerRadius(.cornerRadius)

    RoundedRectangle(cornerRadius: .cornerRadius)
        .fill(.clear)
        .background(backgroundView2)
        .padding([.top, .bottom, .leading], .zero)
        .padding(.trailing, .trailingPadding)
        .frame(width: .width, height: .height)
        .shadow(color: .black.opacity(.shadowColorOpacity), radius: .shadowRadius, x: .shadowXCoord, y: .shadowYCoord)
}
```

Failed to produce diagnostic for expression; please submit a bug report (<https://swift.org/contributing/#reporting-bugs>) and include the project



10:29

98



Please adopt
container-
Background API

Тинькофф



Please adopt
container-
Background API

Тинькофф



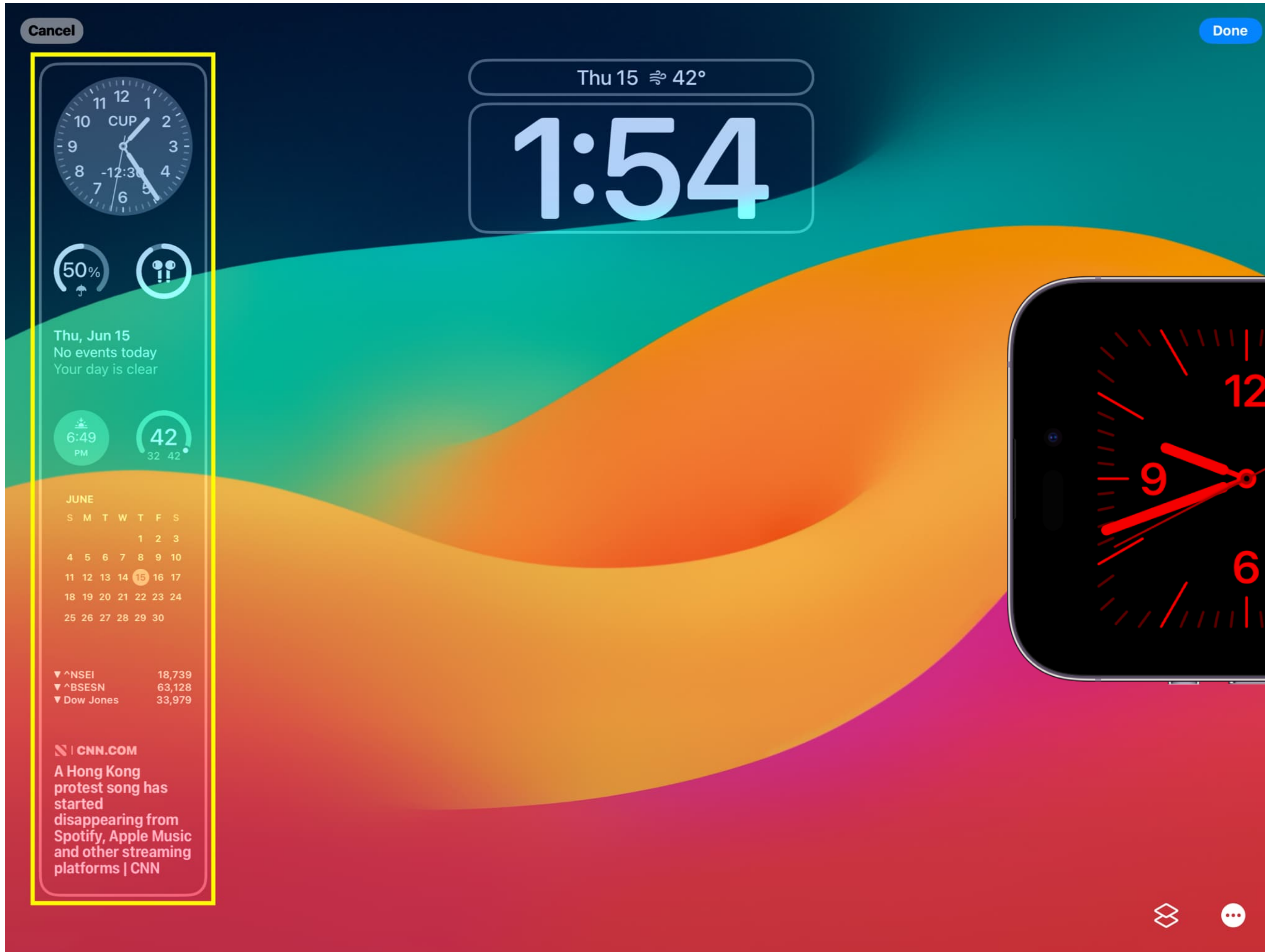
Сканировать
QR или фото

КЭШБЭК ДНЯ

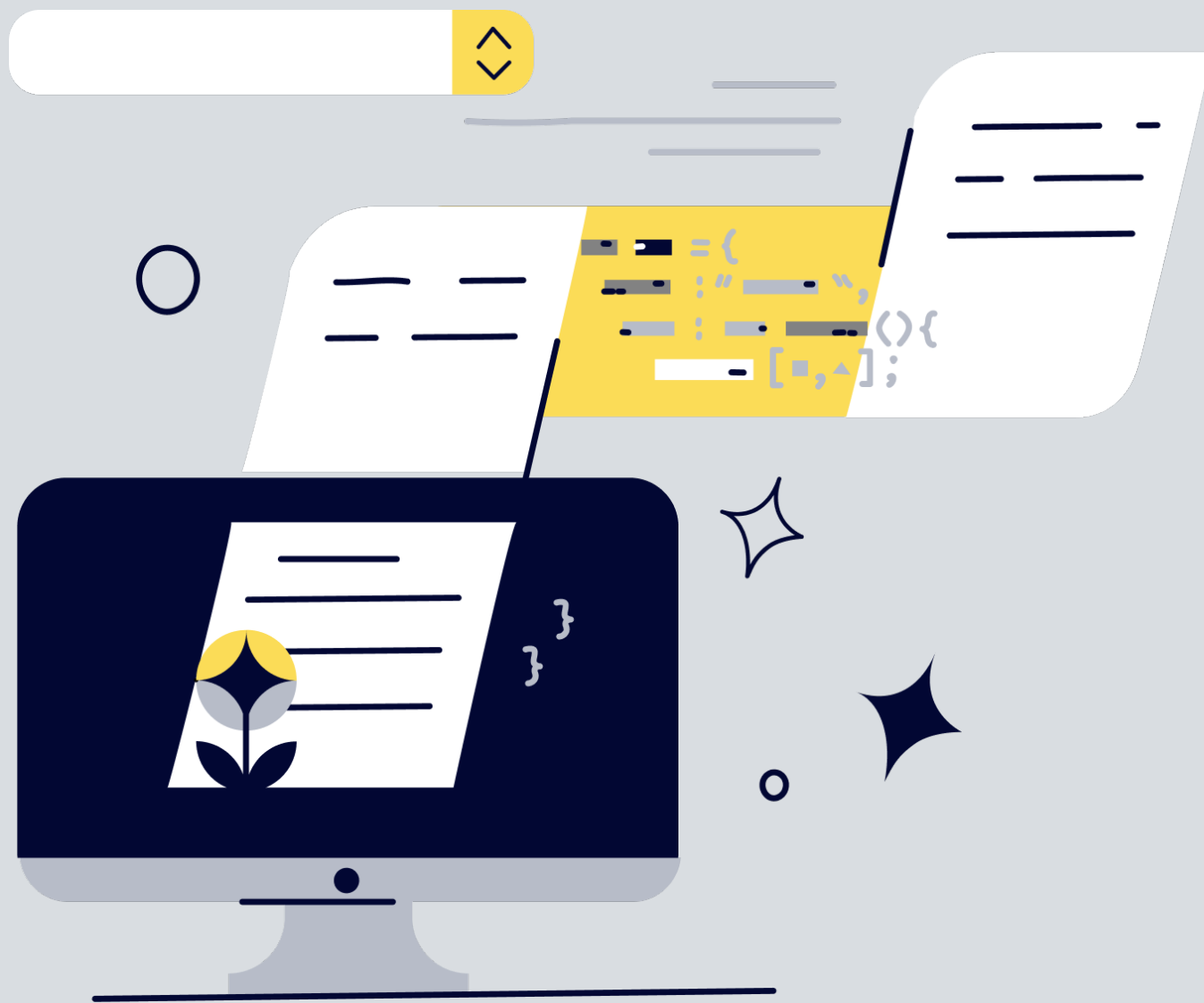
СТОЛОТО

Кэшбэк 20%





Адаптация нового API



```
public struct SomeWidget: Widget {  
  
    public init() {}  
  
    public var body: some WidgetConfiguration {  
        StaticConfiguration(kind: "someWidget", provider: SimpleTimelineProvider()) { _ in  
            SomeWidgetView().containerBackgroundIfNeeded()  
        }  
        .configurationDisplayName("Title")  
        .description("Subtitle")  
        .supportedFamilies([.systemSmall])  
        .containerBackgroundRemovable(false)  
    }  
}  
  
extension View {  
  
    func containerBackgroundIfNeeded() → some View {  
        if #available(iOS 17.0, *) {  
            return self.containerBackground(for: .widget) { }  
        } else {  
            return self  
        }  
    }  
}
```

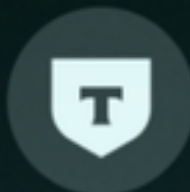



17.2 iPad Pro (12.9-inch)
6th gen – iOS 17.2




Carrier

100%



Monday, May 6

6:49


Check your internet
connection or log in
to the app
[Log in](#)

План-капкан

01

Информация: как хранить?
Связь приложение-виджет?

02

Виджет умеет сам ходить в сеть?

03

Как можно сделать
ничёшный UI?

04

**Доступен ли механизм
фича-тогглов?**

05

Можно ли обработать
аналитику действий?

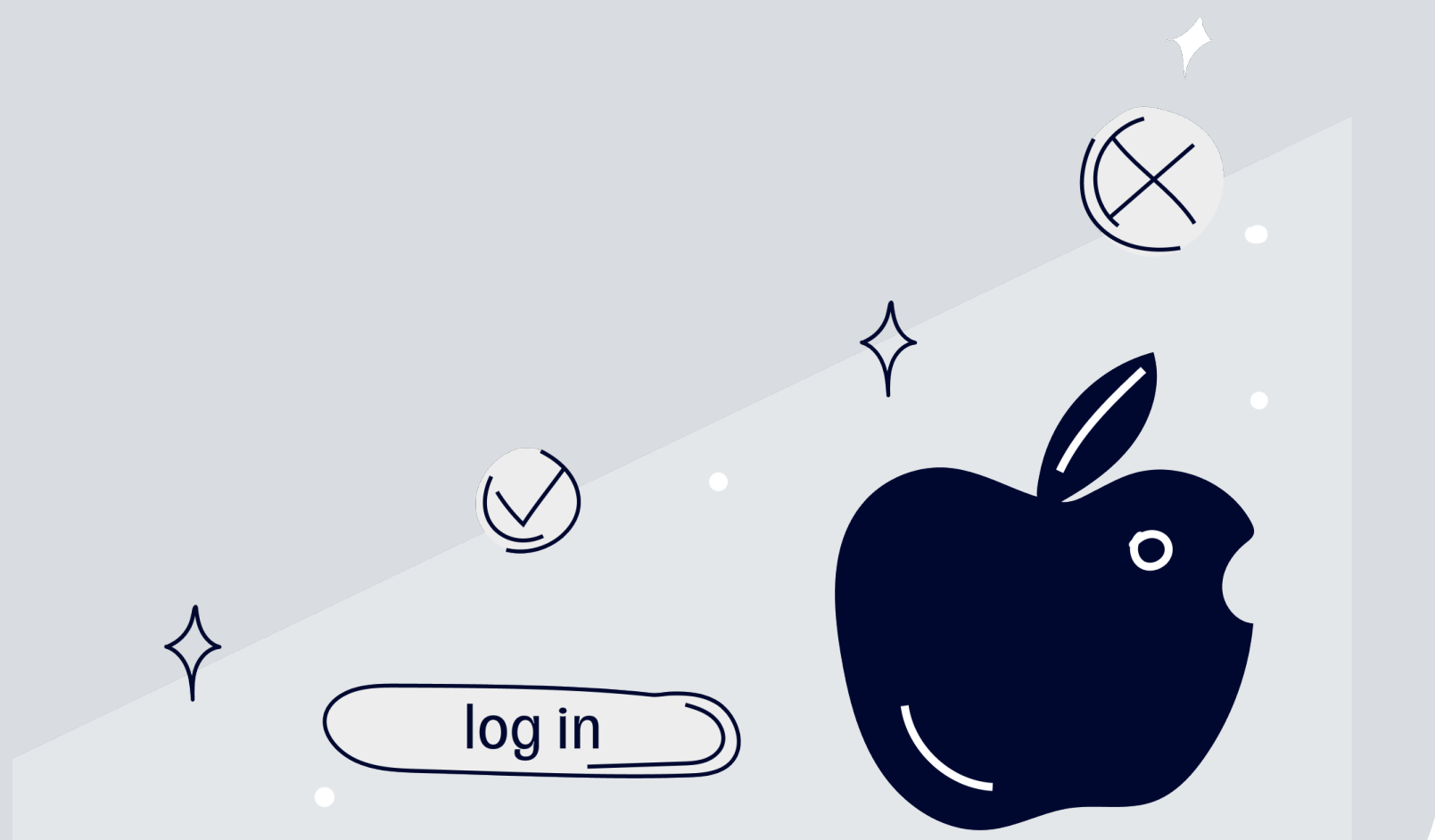
06

Как организовать код в
проекте?

07

Можно ли прикрутить тесты
к виджетам?

Фича-тогглы



Локальные

Для скрывтия недоделанных фич от пользователей



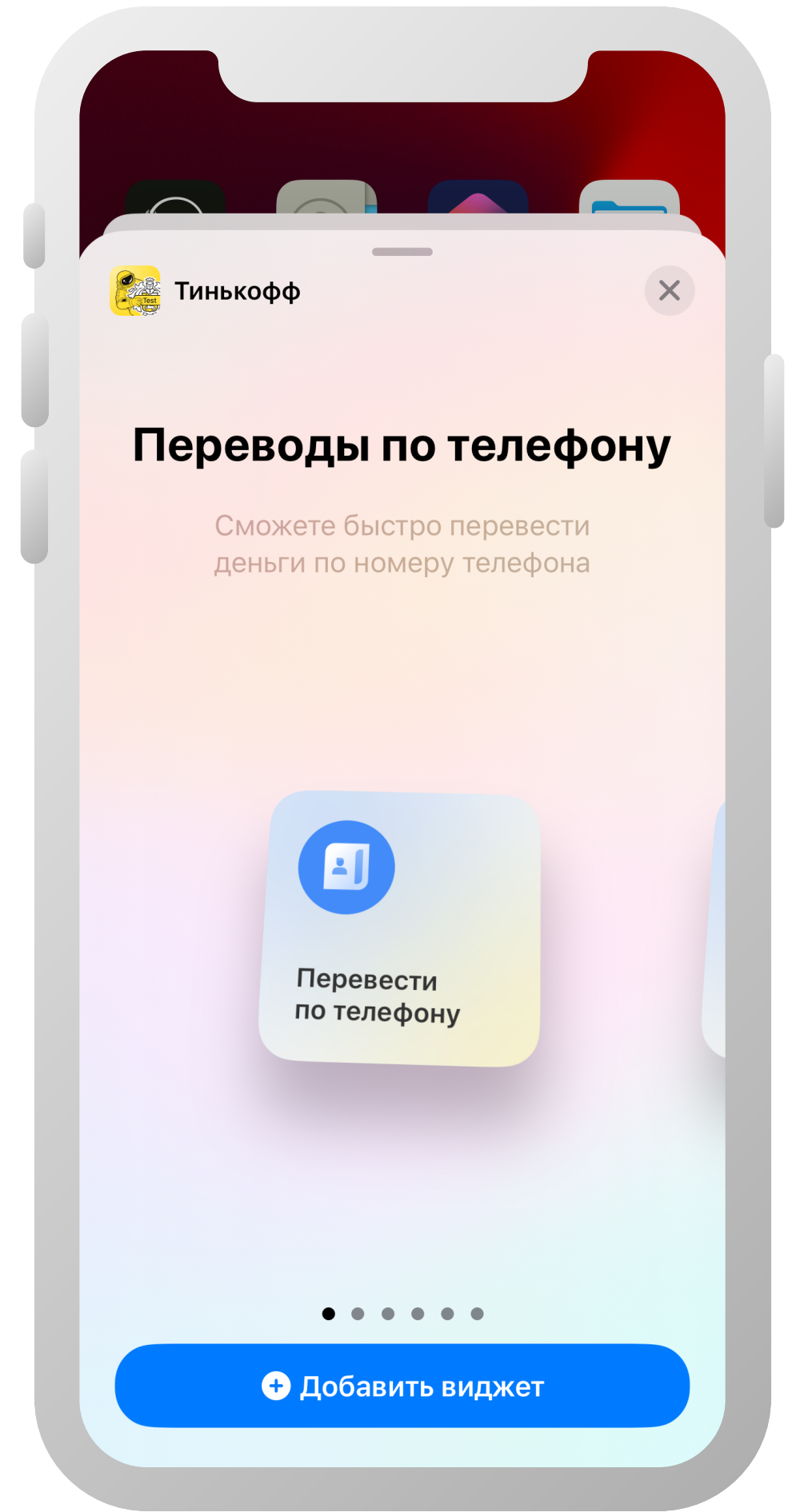
Ремоут

Удаленное управление фичами, A/B тестирование









Попробуем обойти ограничение

```
class Storage {  
  
    static let shared = Storage()  
  
    var toggle: Bool? {  
        get { UserDefaults.standard.value(forKey: "toggle") as? Bool }  
        set { UserDefaults.standard.setValue(newValue, forKey: "toggle") }  
    }  
}  
  
class FeatureToggle {  
  
    static let shared = FeatureToggle()  
  
    let lock = NSLock()  
    var toggle: Bool = false  
    let session = URLSession.shared  
    var task: URLSessionTask?  
  
    func fetch() {  
  
        guard Storage.shared.toggle == nil else { return }  
  
        lock.lock()  
  
        task = session.dataTask(  
            with: URLRequest(url: URL(string: "tinkoff.ru/widgetToggle.json"))!  
        ), completionHandler: { [lock] data, res, err in  
            Storage.shared.toggle = String(data: data, encoding: .utf8).flatMap(Bool.init)  
            lock.unlock()  
        })  
        task?.resume()  
  
        lock.lock()  
        lock.unlock()  
    }  
}
```

Попробуем обойти ограничение

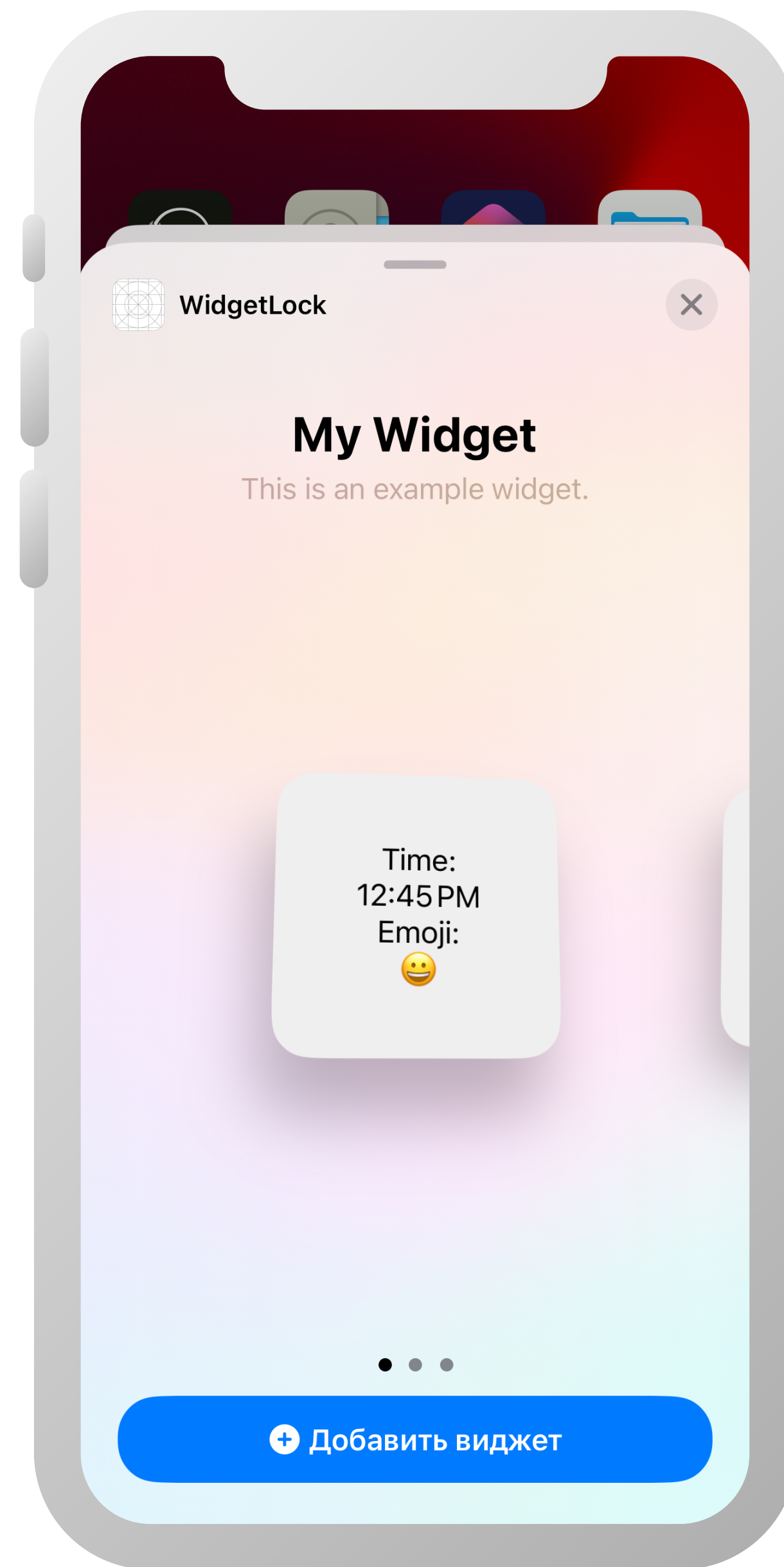
```
import WidgetKit
import SwiftUI

@main
struct LockedWidgetMain {
    static func main() {
        FeatureToggle.shared.fetch()
        LockedWidgetBundle.main()
    }
}

struct LockedWidgetBundle: WidgetBundle {
    var body: some Widget {
        LockedWidget()
    }
}
```

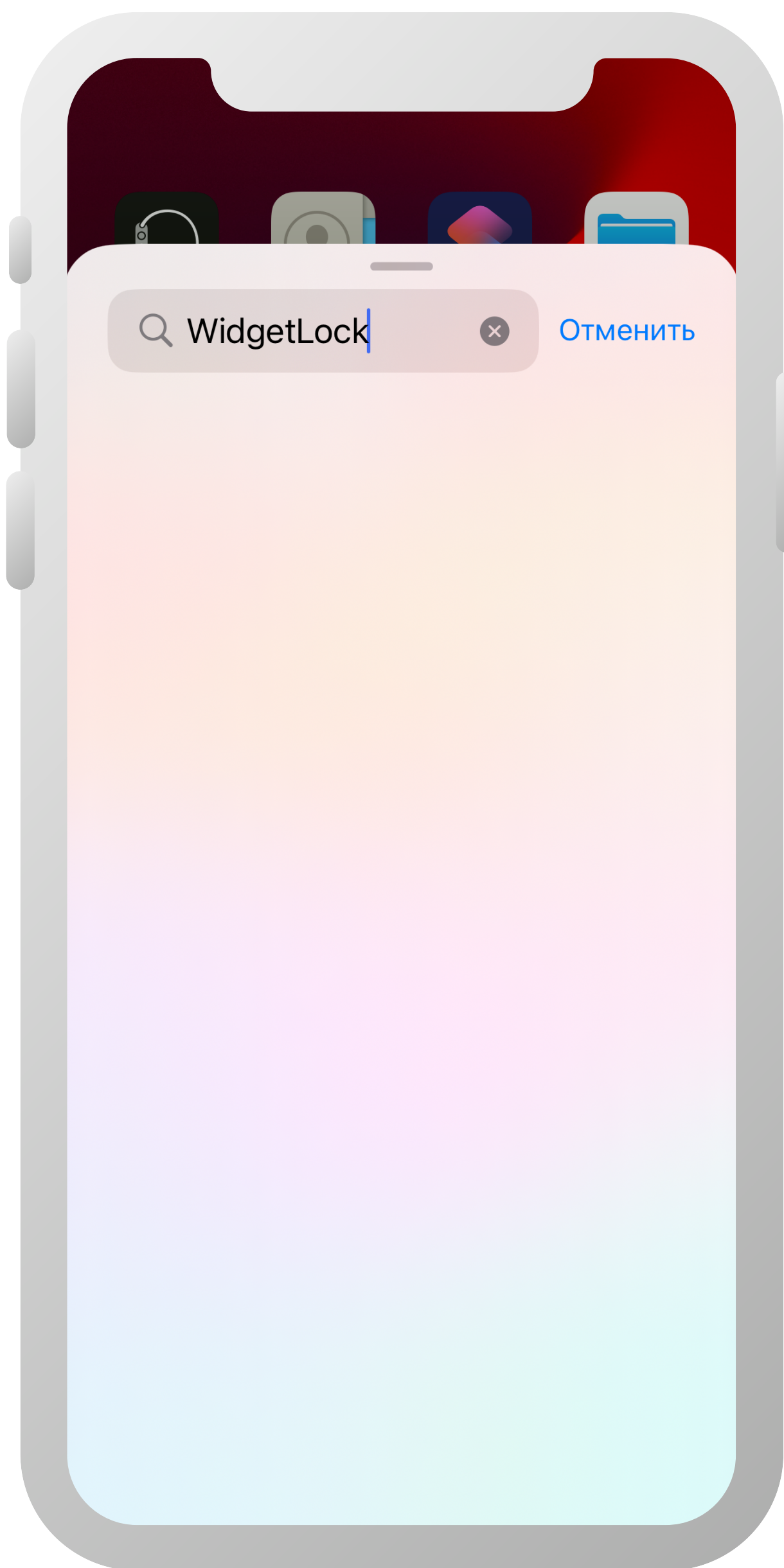


```
WidgetLock
17.2 iPhone Running WidgetLock on 17.2 iPhone 12 Pro
LockedWidget AppDelegate SceneDelegate LockedWidgetBundle
WidgetLock > LockedWidget > LockedWidgetBundle > No Selection
1 //
2 // LockedWidgetBundle.swift
3 // LockedWidget
4 //
5
6 import WidgetKit
7 import SwiftUI
8
9 @main
10 struct LockedWidgetMain {
11     static func main() {
12         FeatureToggle.shared.fetch()
13         LockedWidgetBundle.main()
14     }
15 }
16
17 struct LockedWidgetBundle: WidgetBundle {
18     var body: some Widget {
19         LockedWidget()
20     }
21 }
22
```



Попробуем обойти ограничение

```
class Storage {  
  
    static let shared = Storage()  
  
    var toggle: Bool? {  
        get { UserDefaults.standard.value(forKey: "toggle") as? Bool }  
        set { UserDefaults.standard.setValue(newValue, forKey: "toggle") }  
    }  
}  
  
class FeatureToggle {  
  
    static let shared = FeatureToggle()  
  
    let lock = NSLock()  
    var toggle: Bool = false  
    let session = URLSession.shared  
    var task: URLSessionTask?  
  
    func fetch() {  
  
        guard Storage.shared.toggle == nil else { return }  
  
        lock.lock()  
  
        task = session.dataTask(  
            with: URLRequest(url: URL(string: "tinkoff.ru/widgetToggle.json"))!  
        ), completionHandler: { [lock] data, res, err in  
            Storage.shared.toggle = String(data: data, encoding: .utf8).flatMap(Bool.init)  
            /* lock.unlock() */  
        })  
        task?.resume()  
  
        lock.lock()  
        lock.unlock()  
    }  
}
```

План-капкан

01

Информация: как хранить?
Связь приложение-виджет?

02

Виджет умеет сам ходить в
сеть?

03

Как можно сделать
ничёшный UI?

04

Доступен ли механизм
фича-тогглов?

05

**Можно ли обработать
аналитику действий?**

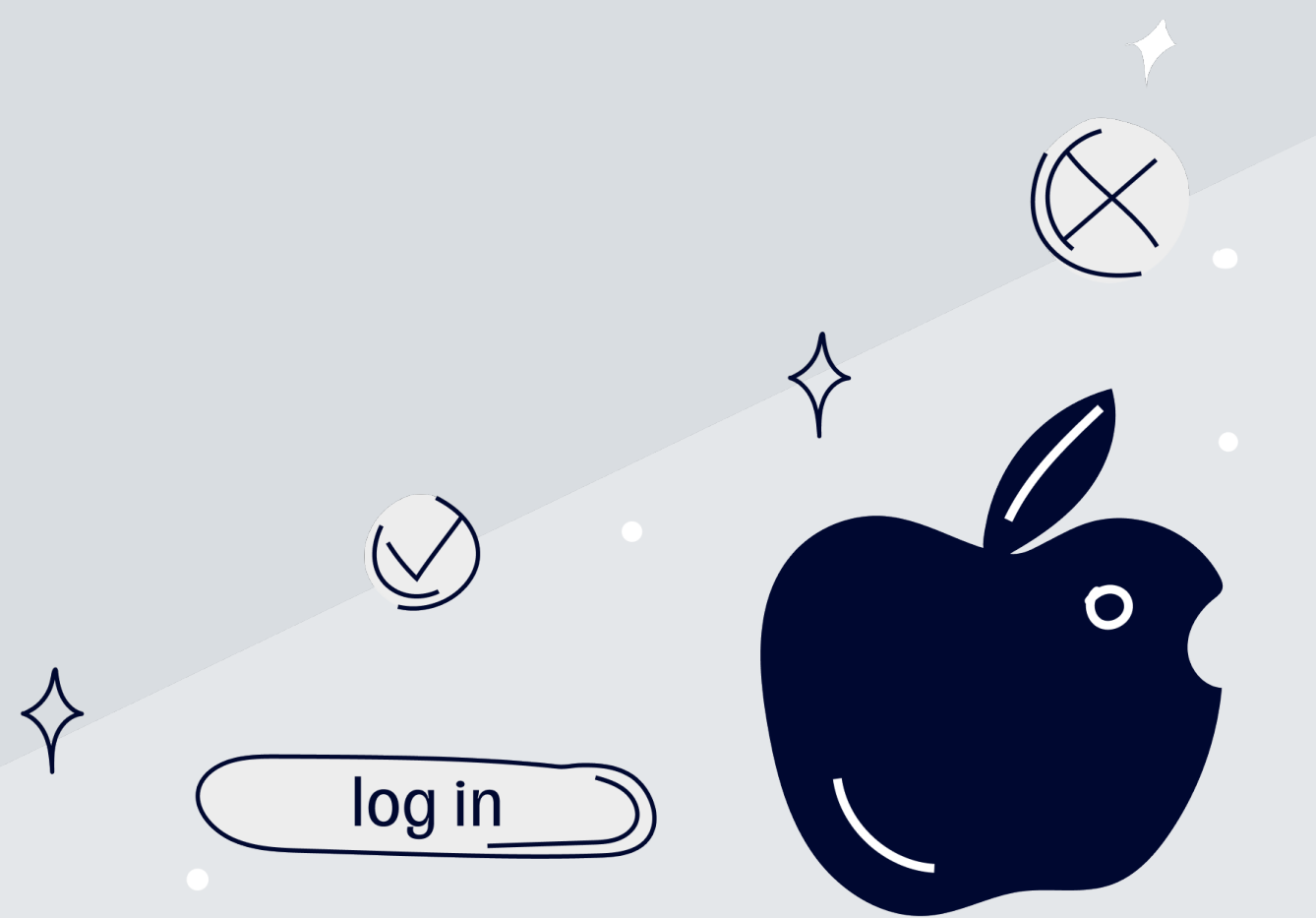
06

Как организовать код в
проекте?

07

Можно ли прикрутить тесты
к виджетам?

Аналитика



Переходы на экраны приложения

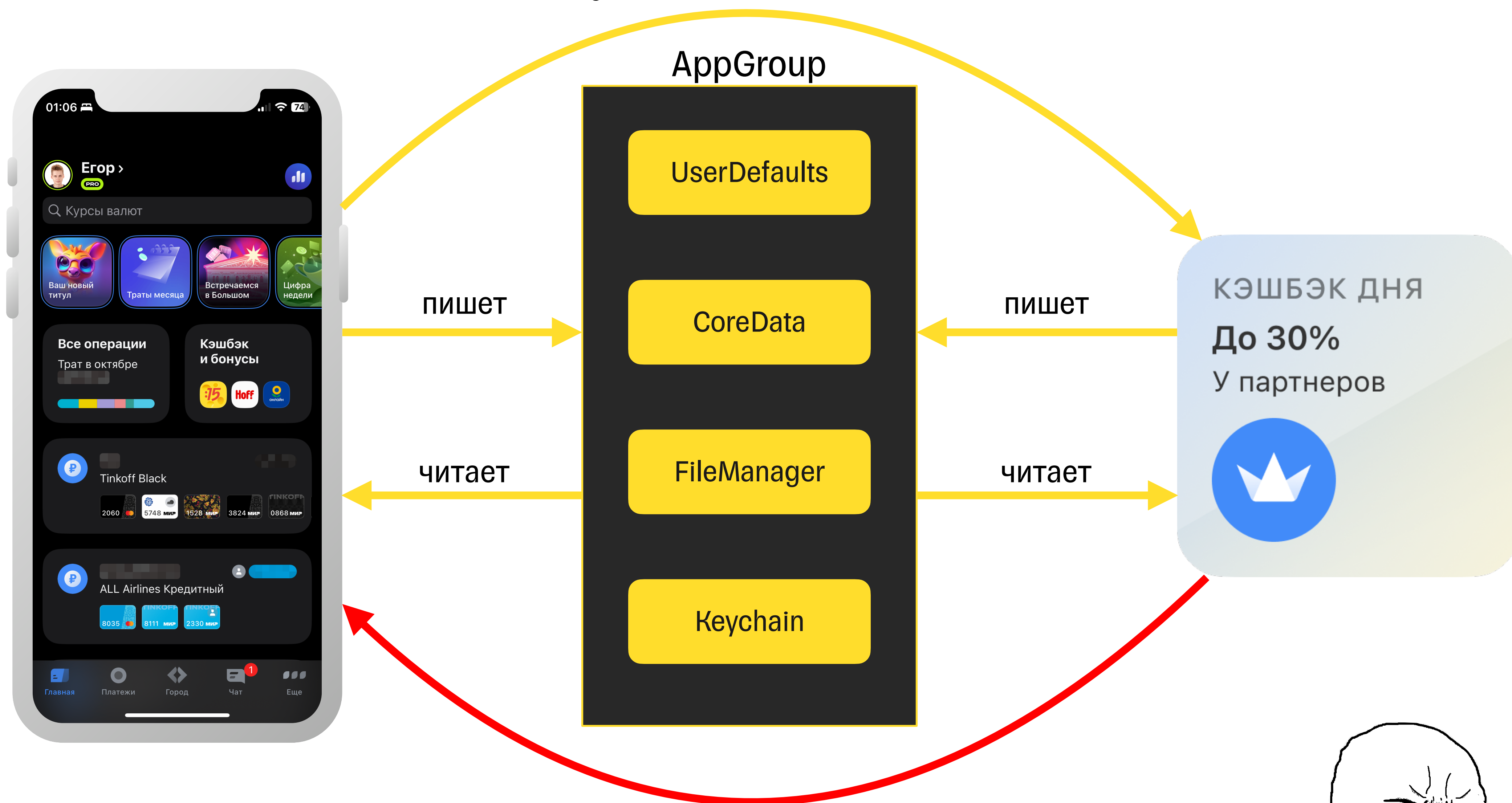
Могут трекаться аналитикой основного приложения, тут проблем нет



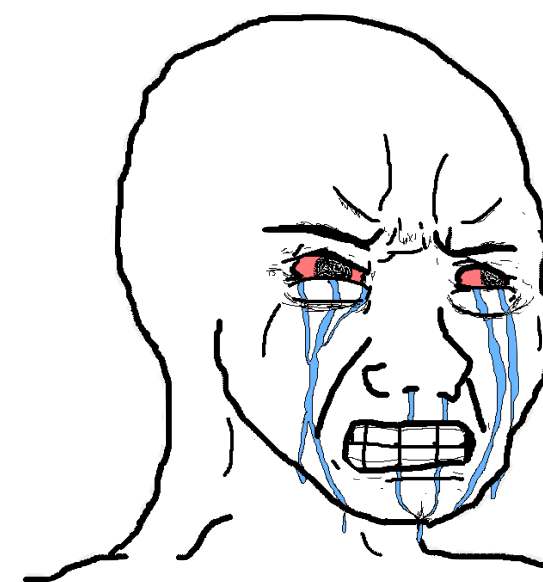
А аналитика самого виджета?

Тапы? Факт установки/удаления? Смена состояний?

WidgetCenter.shared.reloadTimelines

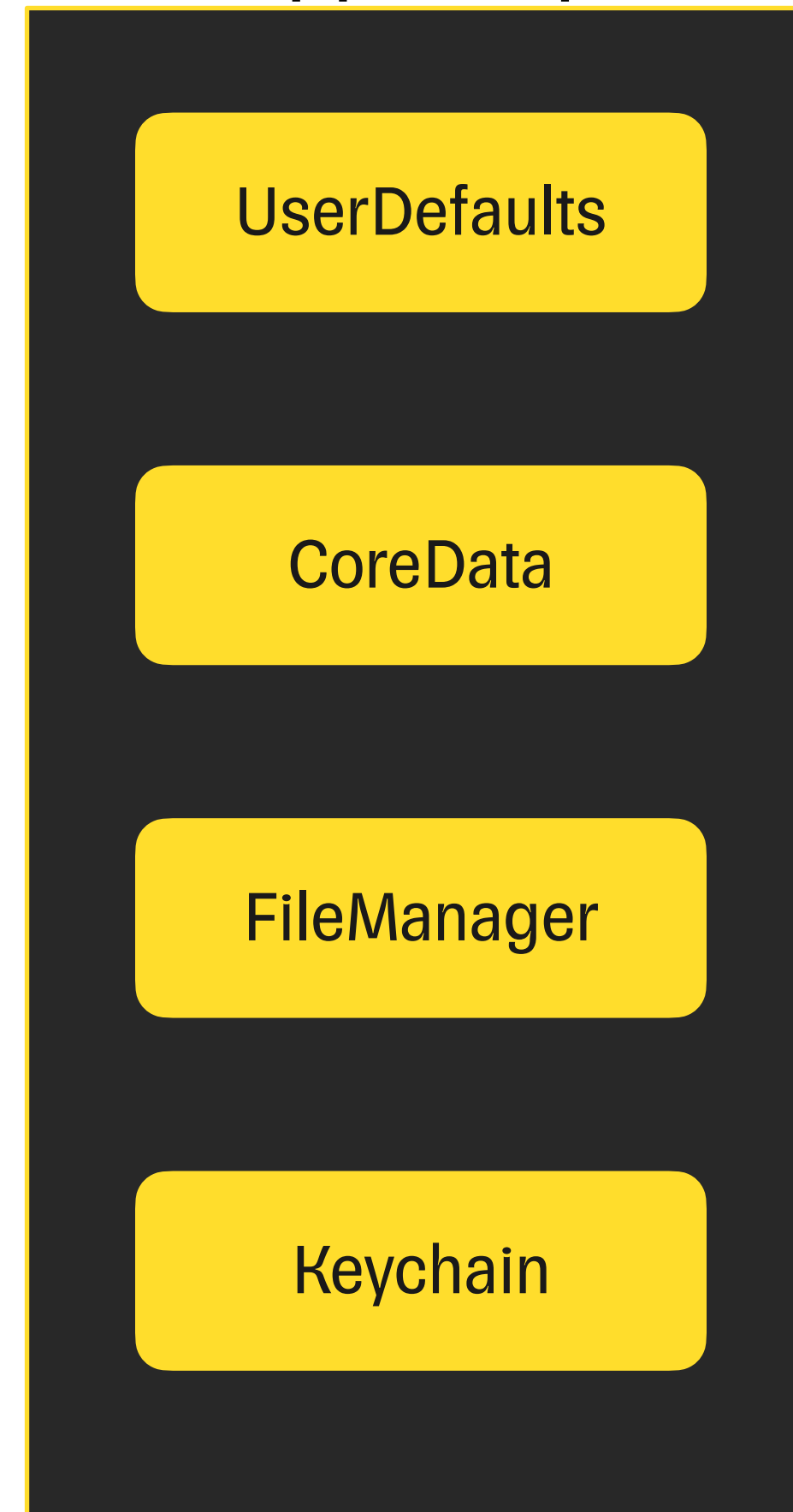


Связи нет

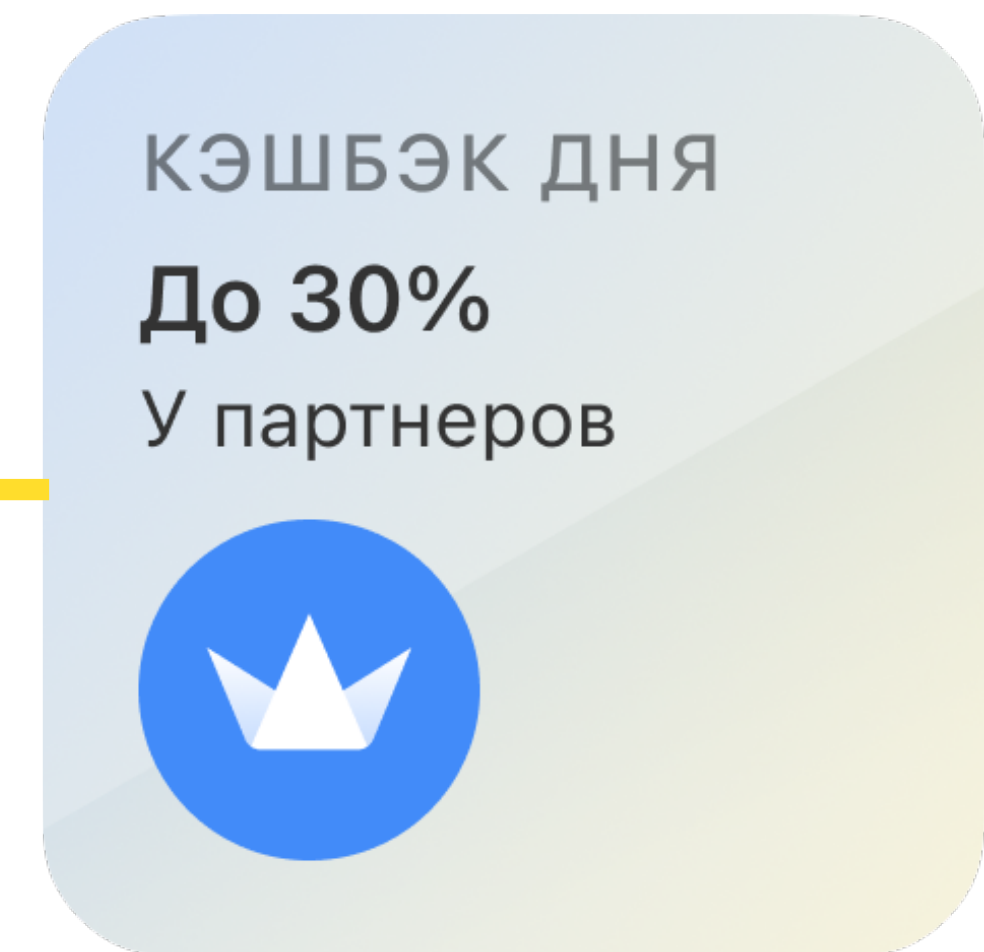




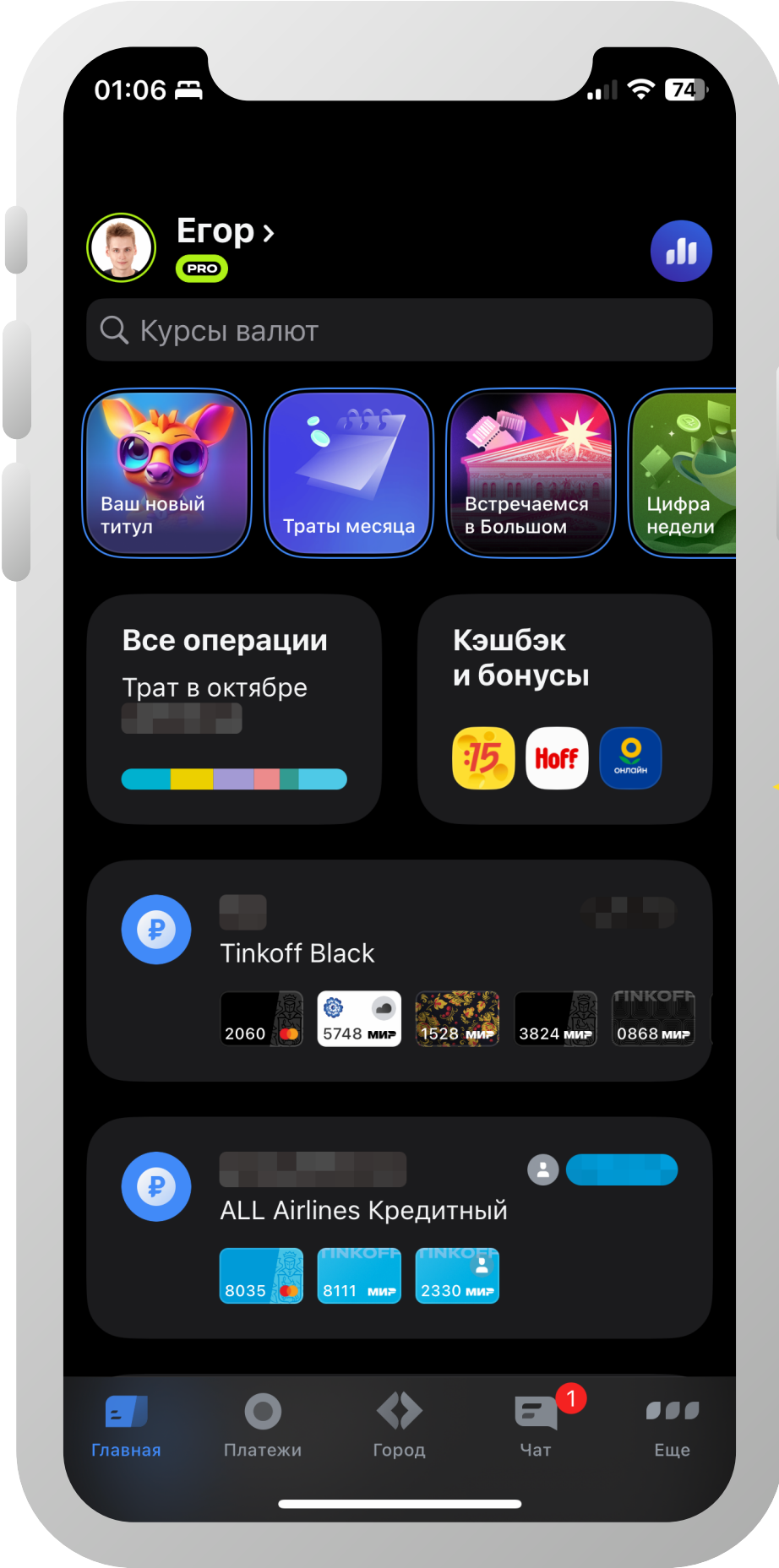
AppGroup



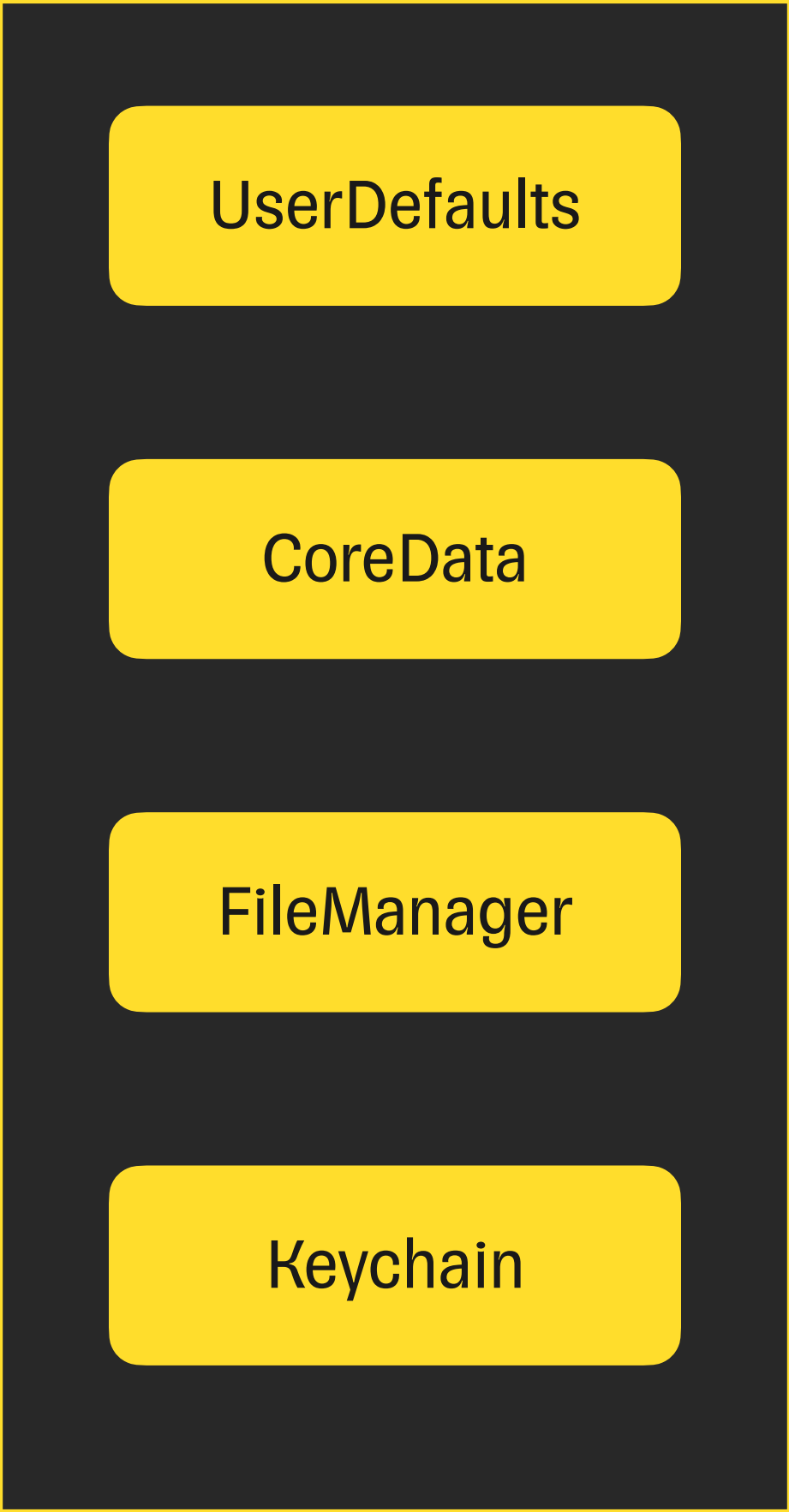
пишет



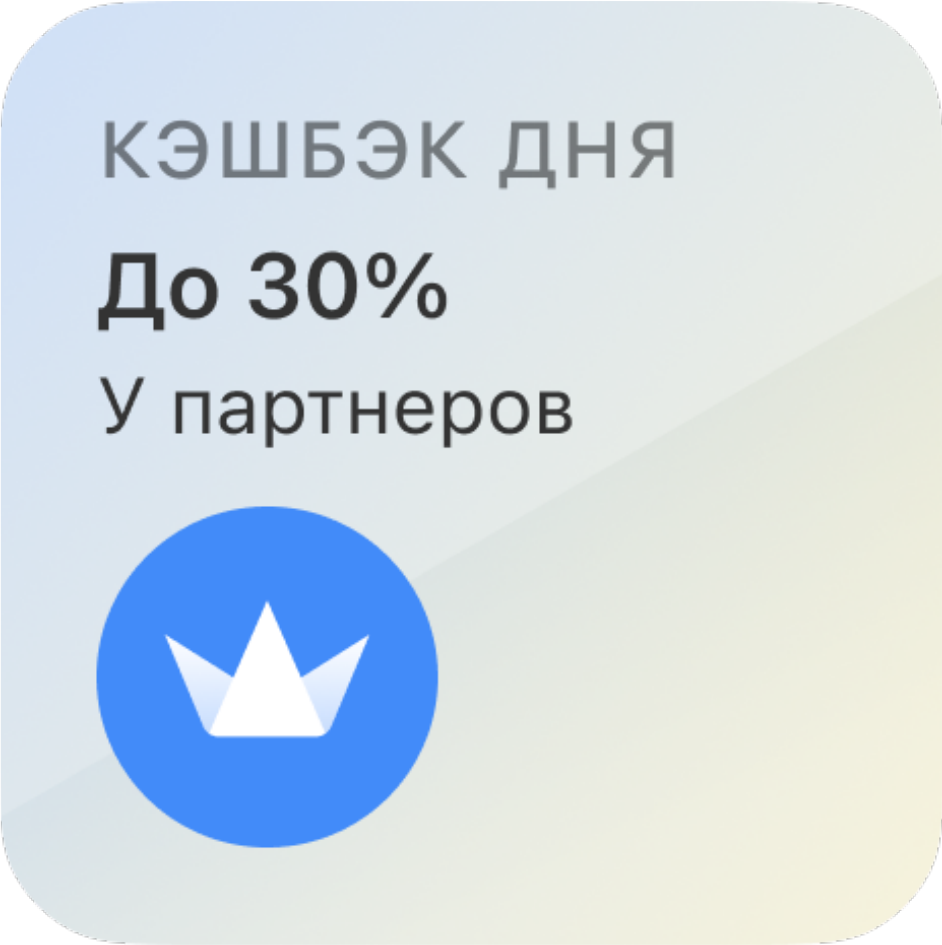
applicationDidBecomeActive



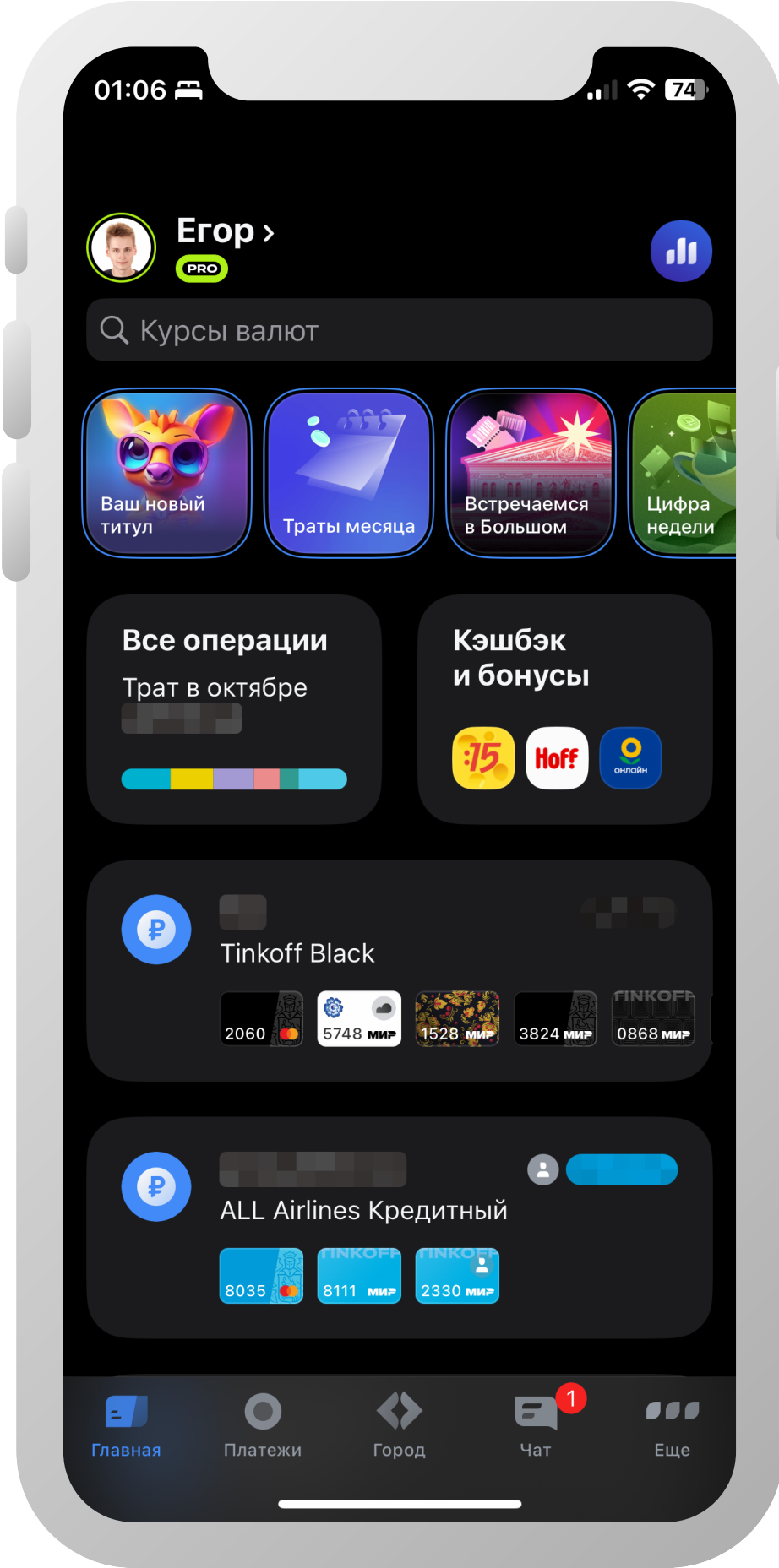
AppGroup



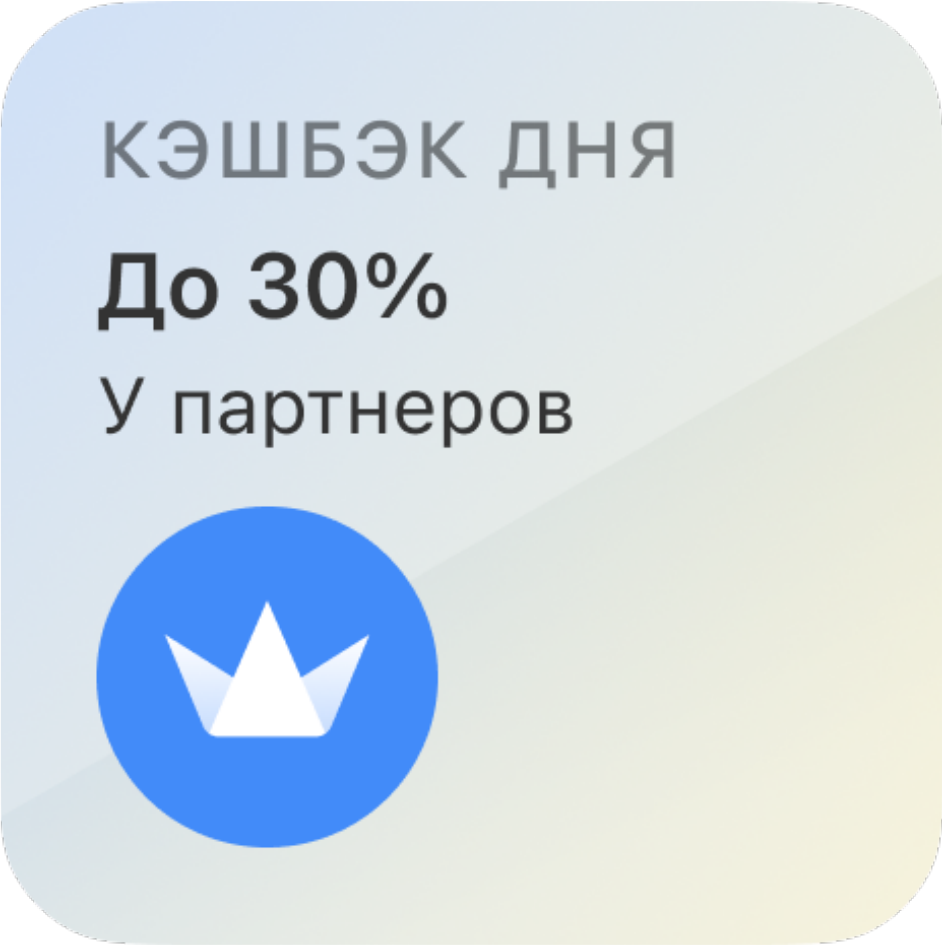
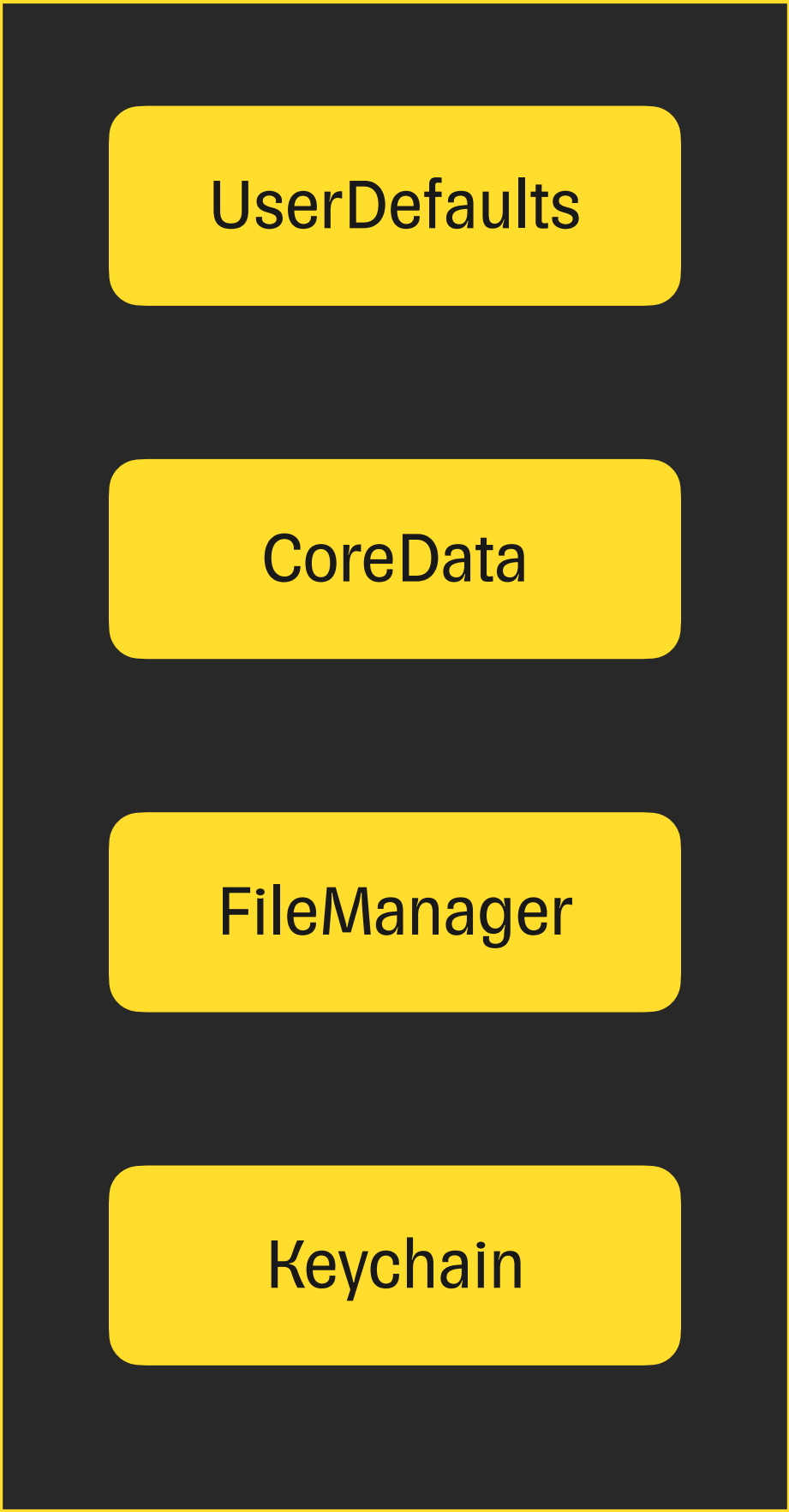
ЧИТАЕТ



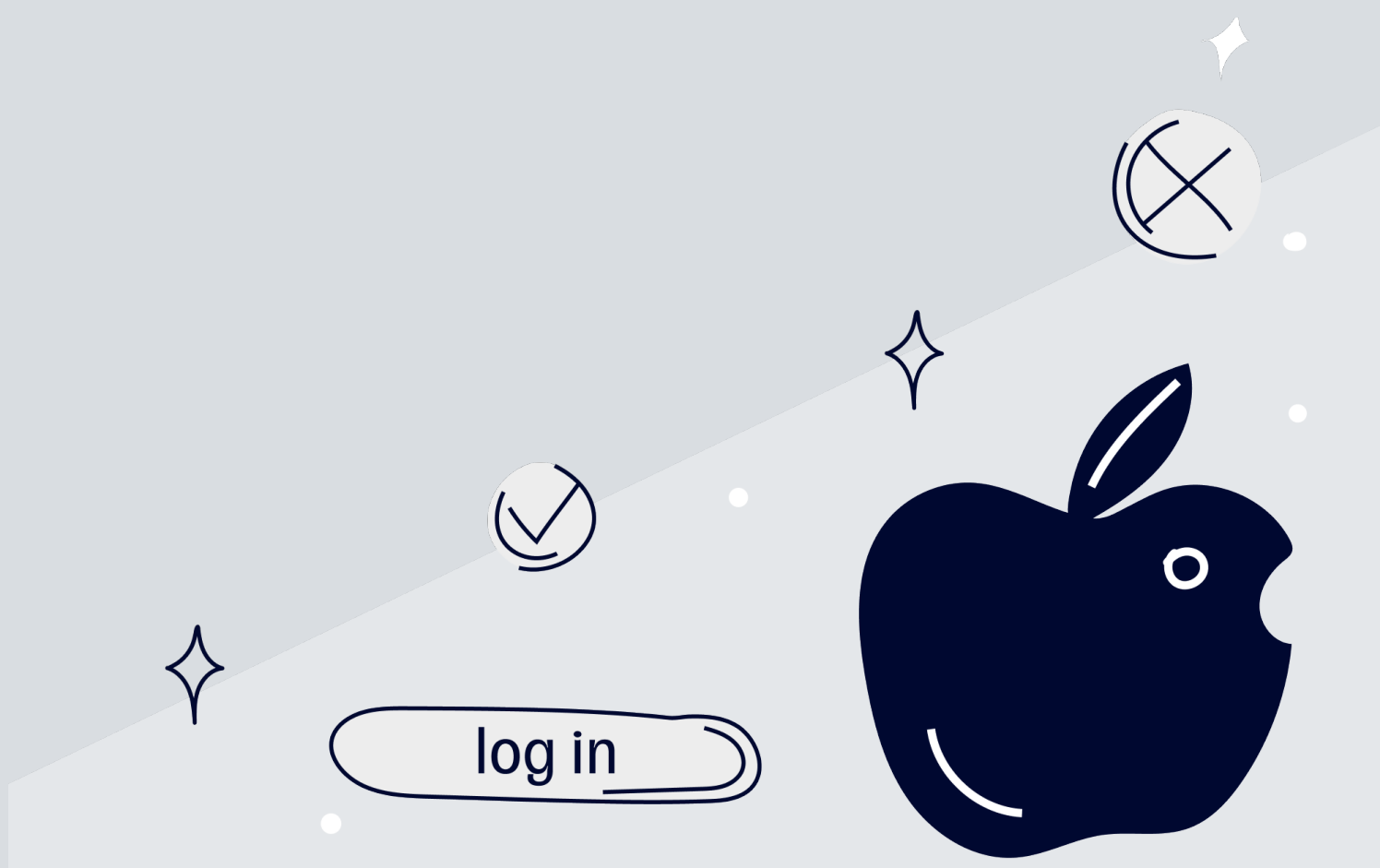
applicationDidBecomeActive



AppGroup



факты установки/ удаления виджета



WidgetCenter.shared.getCurrentConfigurations

Спрашивает у системы список текущих
активных виджетов



Аналитика только по факту открытия приложения

Поскольку нет связи «виджет → основное
приложение»

План-капкан

01

Информация: как хранить?
Связь приложение-виджет?

02

Виджет умеет сам ходить в
сеть?

03

Как можно сделать
ничёшный UI?

04

Доступен ли механизм
фича-тогглов?

05

Можно ли обработать
аналитику действий?

06

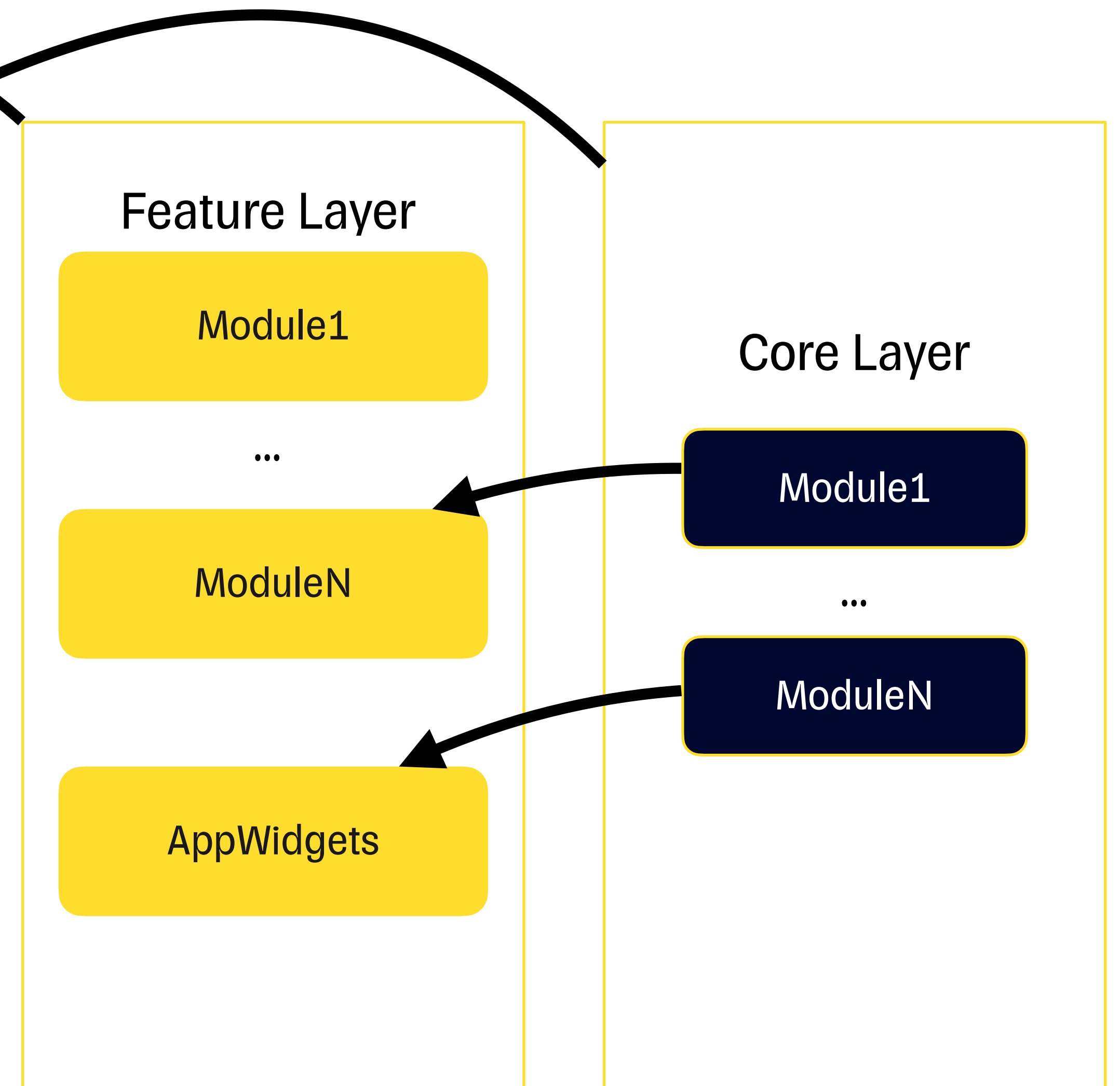
**Как организовать код в
проекте?**

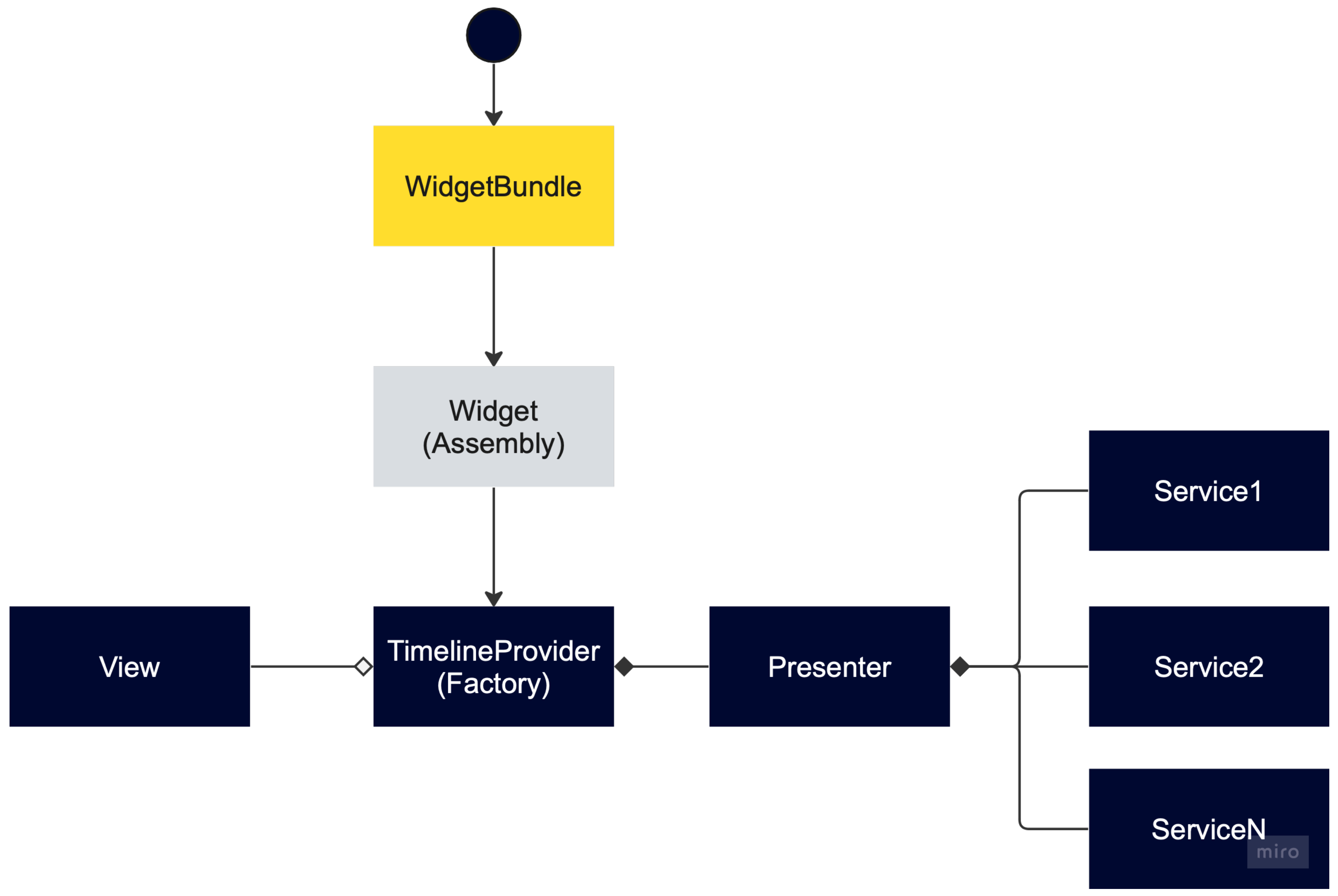
07

Можно ли прикрутить тесты
к виджетам?

MobileBank target

- Стартовая точка виджета - вынести ее нельзя
- Зависимости виджета, зависящие от фичей, которые не подключены к модулю виджетов





План-капкан

01

Информация: как хранить?
Связь приложение-виджет?

02

Виджет умеет сам ходить в сеть?

03

Как можно сделать
ничёшный UI?

04

Доступен ли механизм
фича-тогглов?

05

Можно ли обработать
аналитику действий?

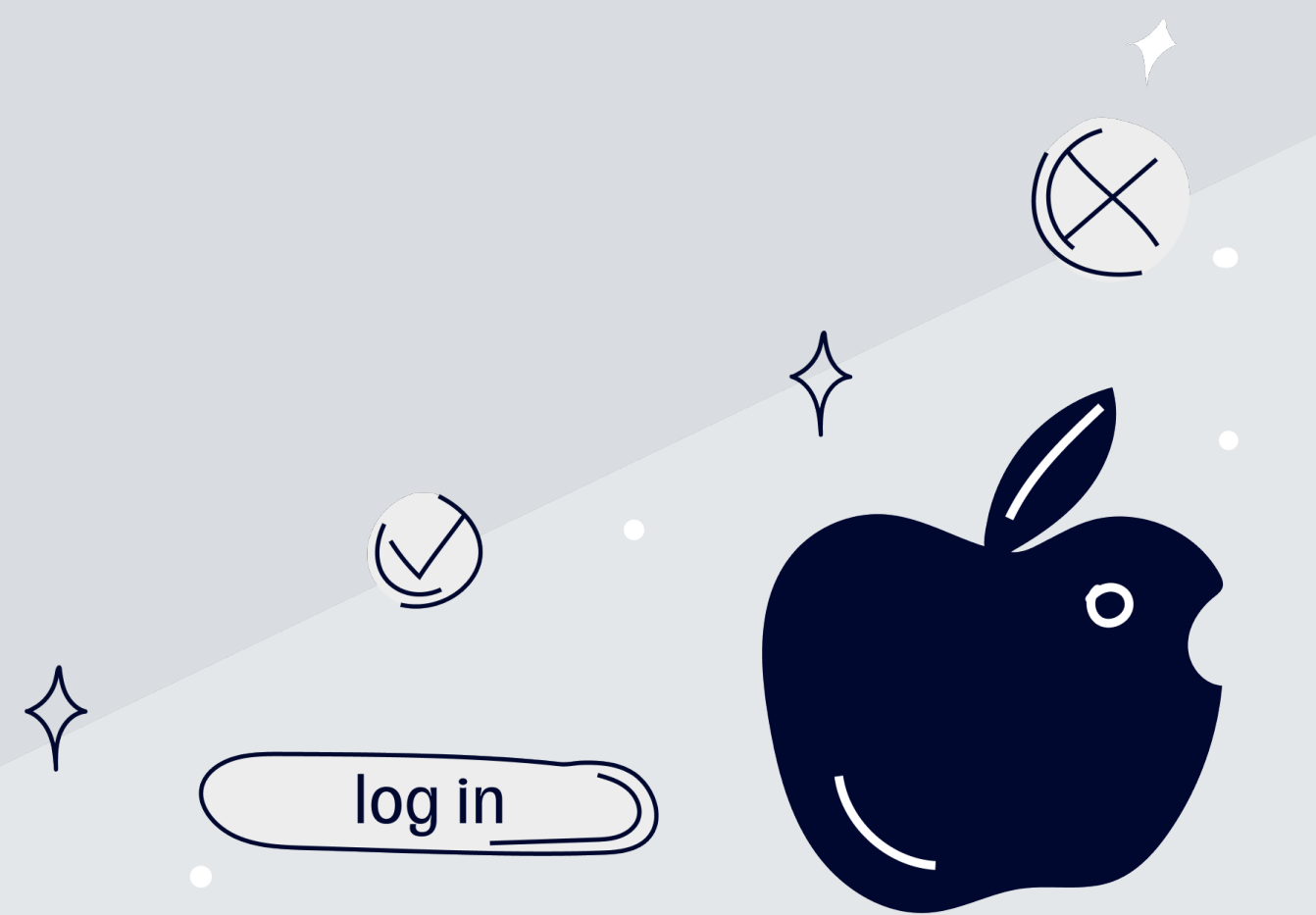
06

Как организовать код в
проекте?

07

**Можно ли прикрутить
тесты к виджетам?**

Тестирование



Unit-тесты

Отлично заходят, бизнес-логика тестируется, можно покрыть коробочные протоколы и структуры



UI-тесты

Доступны, можно автоматизировать открытие приложения через виджет



Snapshot-тесты

Доступны, можно делать слепки отдельных View и проверять, что ничего не поехало



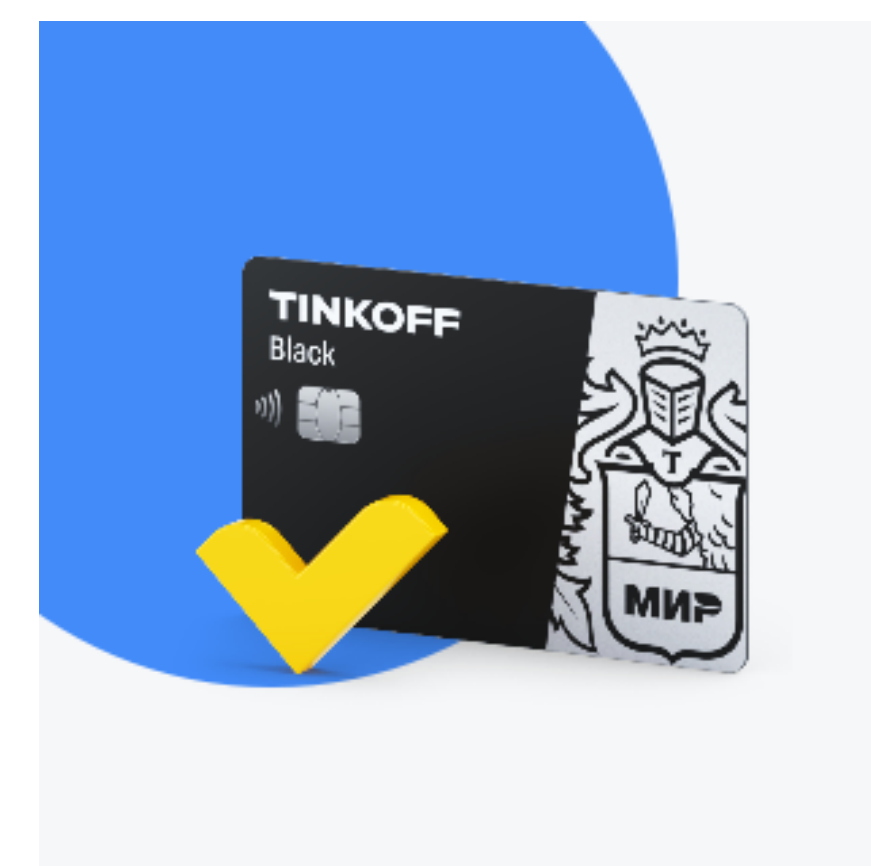
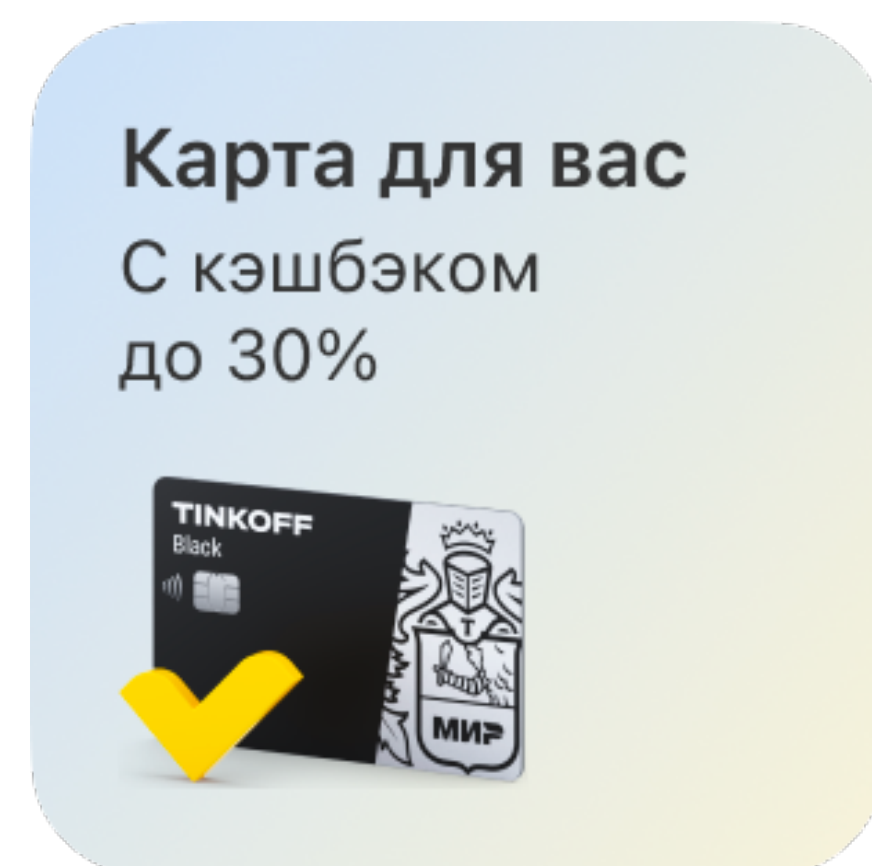
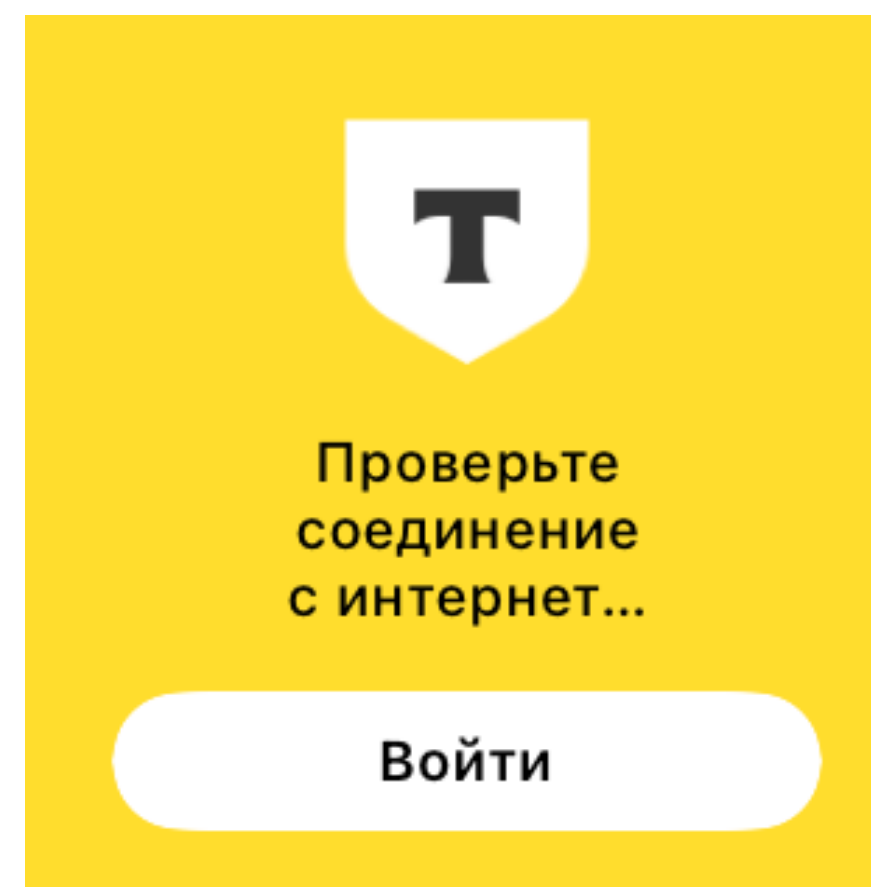
Manual-тесты

Применение ограничено: не все аспекты виджетов можно проверить вручную

Как должно быть



Snapshot-тест



Как полечить?

```
public var body: some WidgetConfiguration {
    StaticConfiguration(kind: "SomeWidget", provider: timelineProvider) { entry in
        SomeWidgetView().environment(\.widgetFamily, .systemSmall)
    }
    .configurationDisplayName("Some title")
    .description("Some subtitle")
    .supportedFamilies([.systemSmall])
}
```

```
extension WidgetFamily: EnvironmentKey {
    public static var defaultValue: WidgetFamily = .systemMedium
}
```

```
extension EnvironmentValues {
    var widgetFamily: WidgetFamily {
        get { self[WidgetFamily.self] }
        set { self[WidgetFamily.self] = newValue }
    }
}
```



Выводы

Так стоят ли виджеты того?

Виджеты-диплинки

Просто

- Ноль бизнес-логики;
- SwiftUI несложен.

Быстро

- Срок разработки ~ 3 дня;*
■ SwiftUI осваивается быстро.

Эффективно

- Быстрый доступ к экранам;
- Стоит затраченных ресурсов.

Информационные виджеты

Вообще не просто

- Проект может быть не готов к виджету;
- Простых решений может не быть.

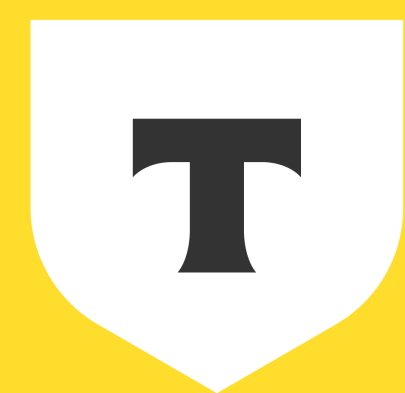
Ни разу не быстро

- Требуется вовлечение множества коллег;
- Требуются ресурсы на разработку.

Эффективно и эффектно

- См. виджеты-диплинки :)
- Вы в авангарде технологий.





ТИНЬКОФФ