

Реализуем CI в iOS

улучшаем, ускоряем, масштабируем!

Игорь Савельев

igor.savelev@corp.badoo.com





Игорь Савельев

 igor.savelev@corp.badoo.com

 twitter.com/leonspok

Содержание

- Введение
- Модуляризация UI
- Visual Regression Testing
- Cosmos Design System
- Заключение

Введение



<https://www.youtube.com/watch?v=mTV45Qa8ZRI>

Введение

Vadoo И НЕ ТОЛЬКО



Модуляризация III

Проблема в 2017 году:
Сборка проекта отнимает очень много рабочего времени у разработчика

Clean build

8-9m

Incremental

1-5m

2 Building Badoo: Badoo | Building 2459 of 3252 tasks

 11

```
// MARK: Set Up

private func setUpBackgroundImageView() {
    self.addSubview(self.backgroundImage)
    self.backgroundImage.contentMode = .scaleAspectFit
}

private func setUpBadgeView() {
    self.addSubview(self.label)
    self.styleLabel()
}

private func styleLabel() {
    self.label.layer.borderWidth = self.style.borderWidth
    self.label.layer.borderColor = self.style.borderColor.cgColor
    self.label.bpui_textStyle = self.style.badgeTextStyle
    self.label.backgroundColor = self.style.badgeBackground
    self.label.clipsToBounds = true
}
```

1. Находим проблему
2. Правим константу
3. Перезапускаем
4. **Ждем 5 минут**



VadooUIKit

Содержит UI компоненты для приложения Vadoo

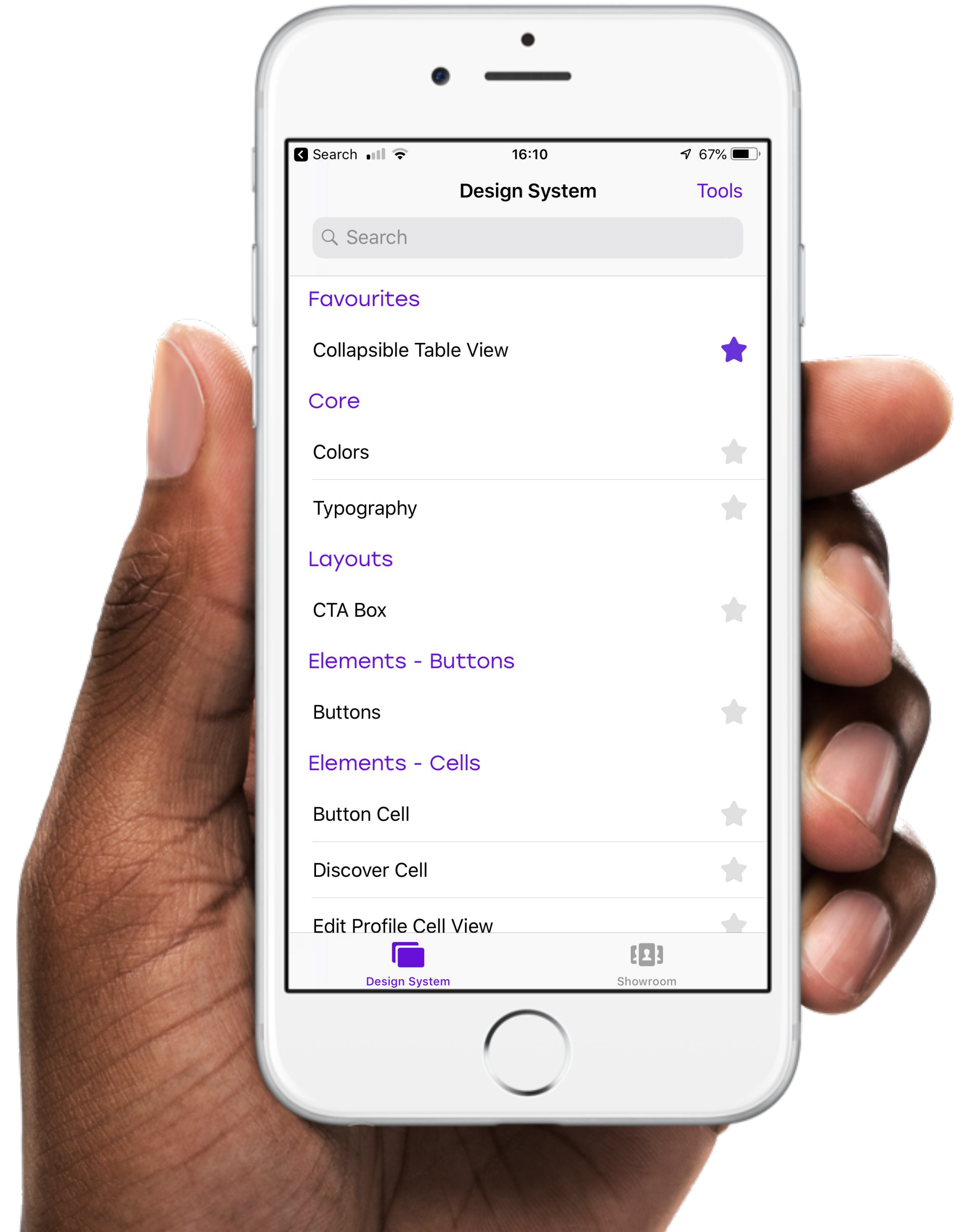


Модуляризация UI



Badoo Gallery

Демонстрирует UI компоненты из BadooUIKit



Badoo

Clean build

8-9m

Incremental

1-5m

Badoo Gallery

Clean build

40s

Incremental

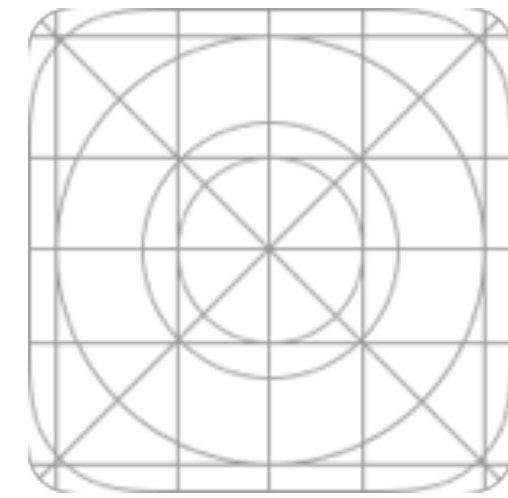
5-10s

-90%

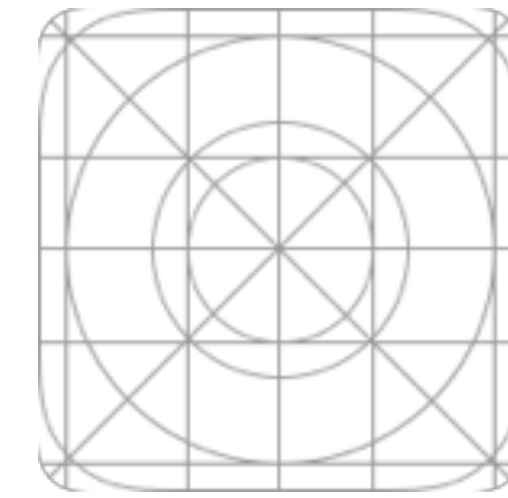
-95%



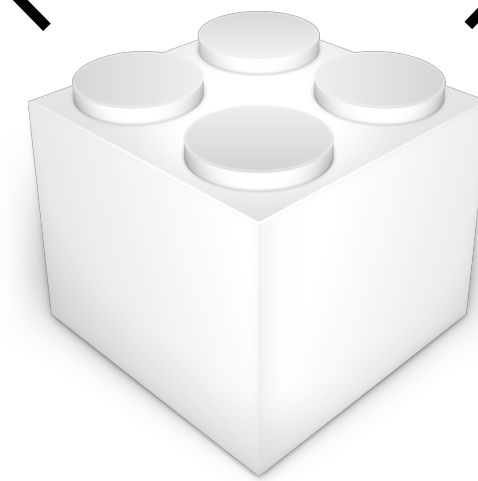
Задача в 2018 году:
Модуляризация UI для всей
группы приложений Badoo



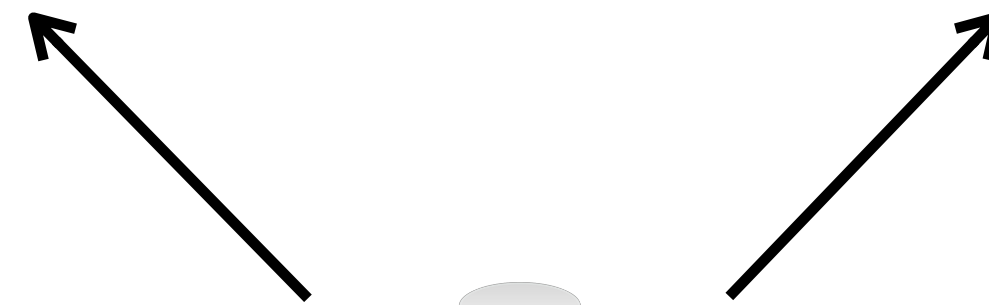
Badoo

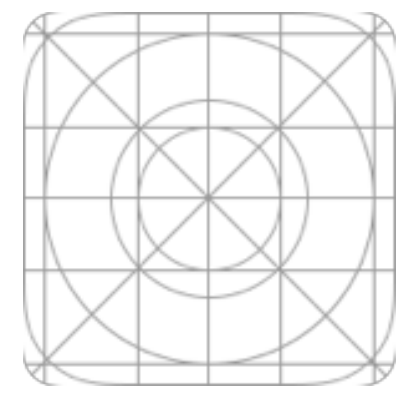


Badoo Gallery

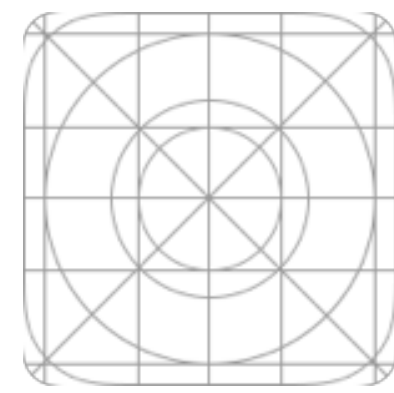


BadooUIKit

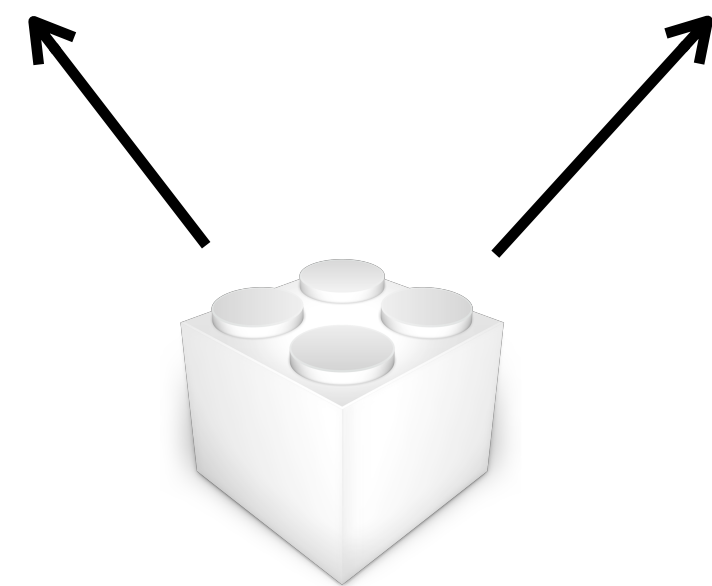




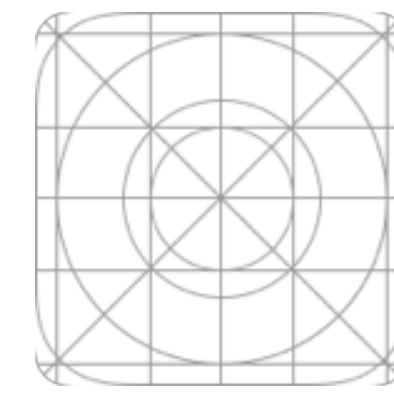
Badoo



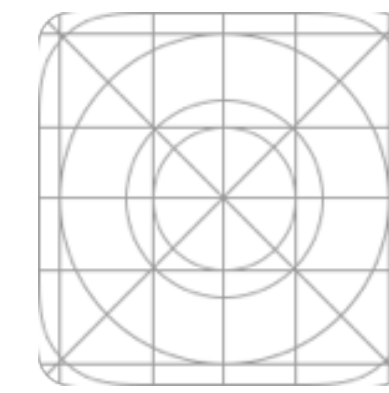
Badoo Gallery



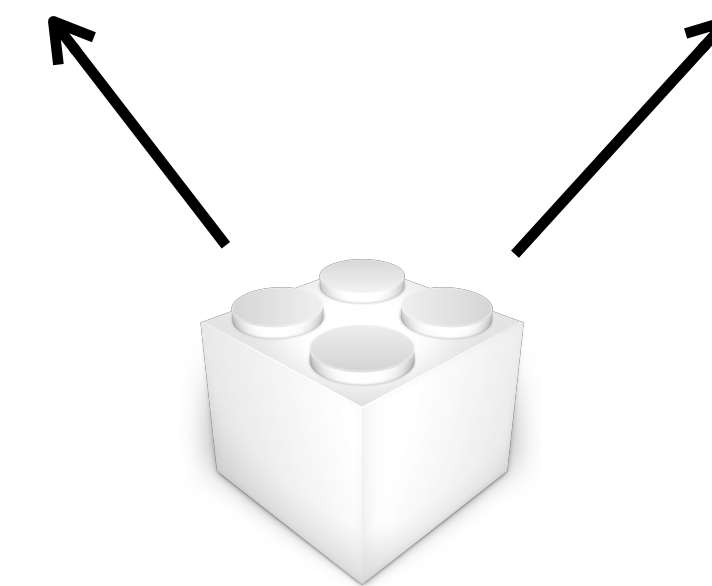
BadooUIKit



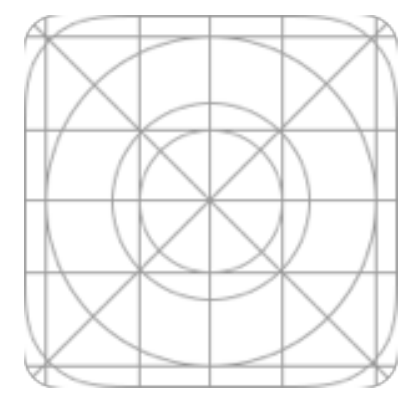
Bumble



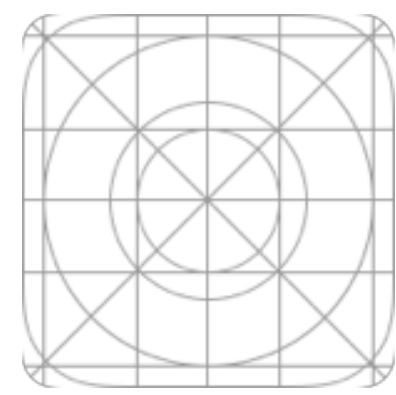
Bumble Gallery



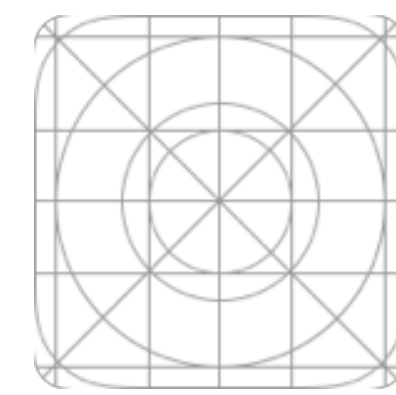
BumbleUIKit



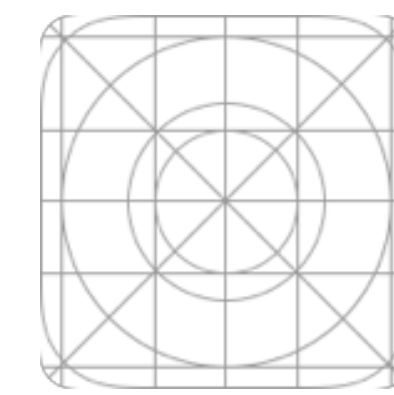
Badoo



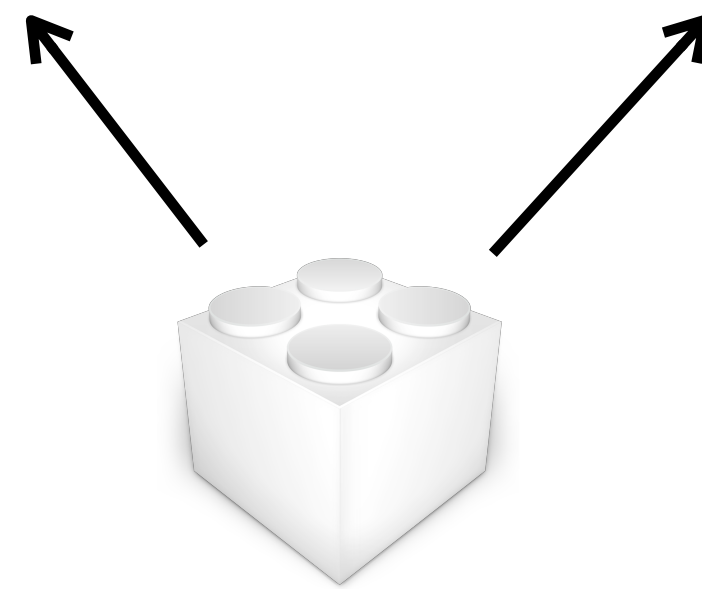
Badoo Gallery



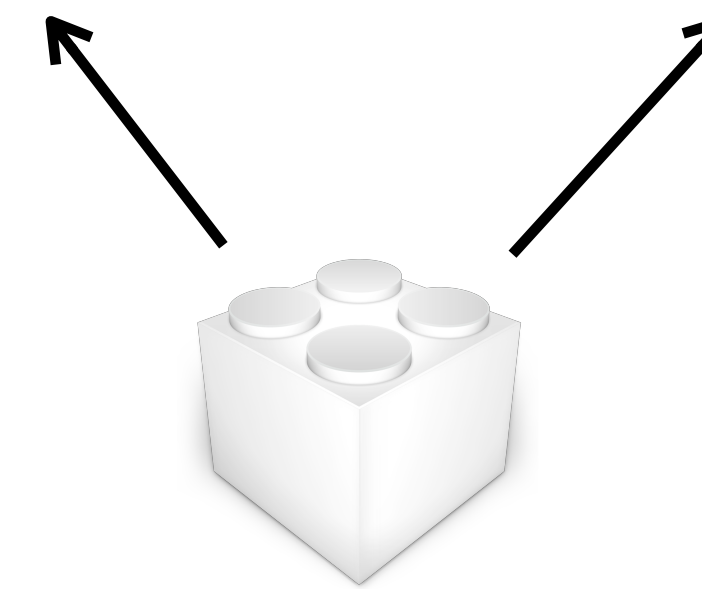
Bumble



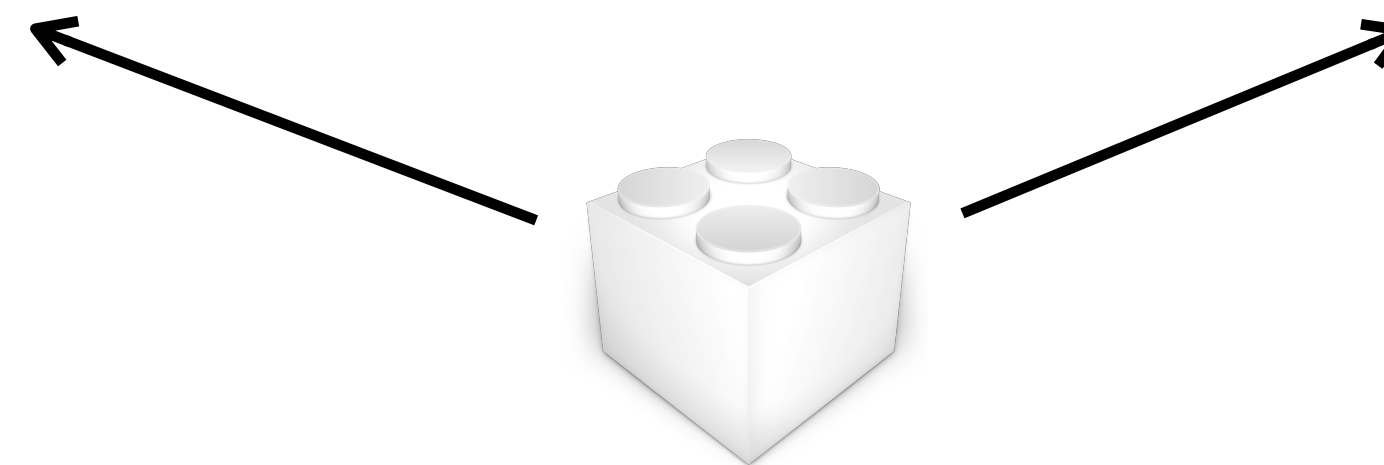
Bumble Gallery



BadooUIKit



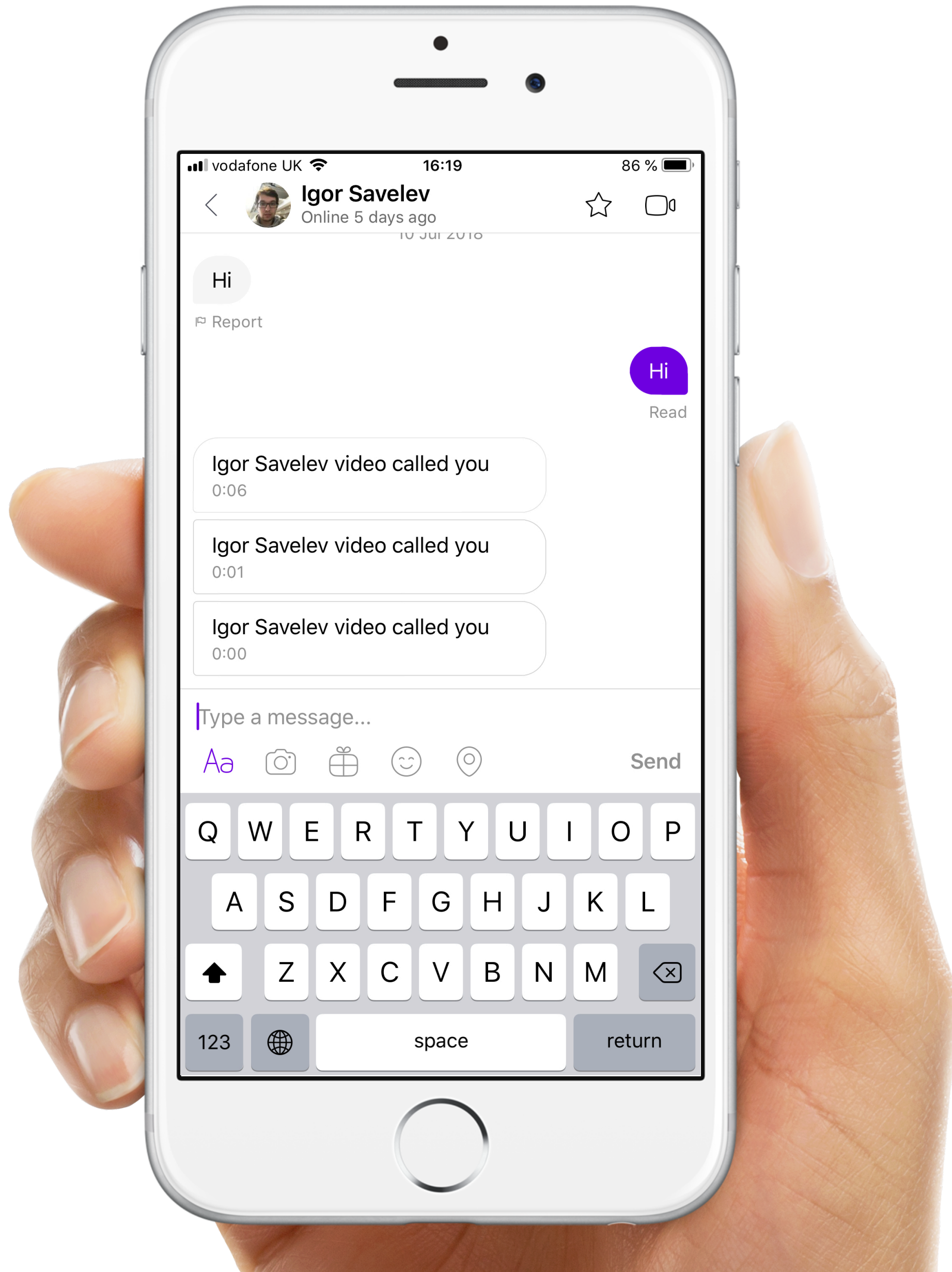
BumbleUIKit



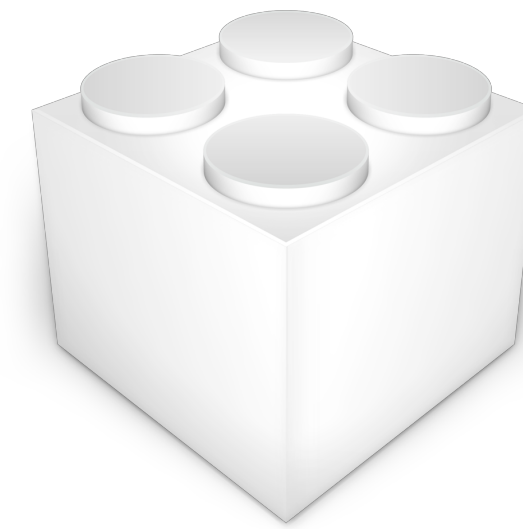
PlatformUIKit

Модуляризация UI

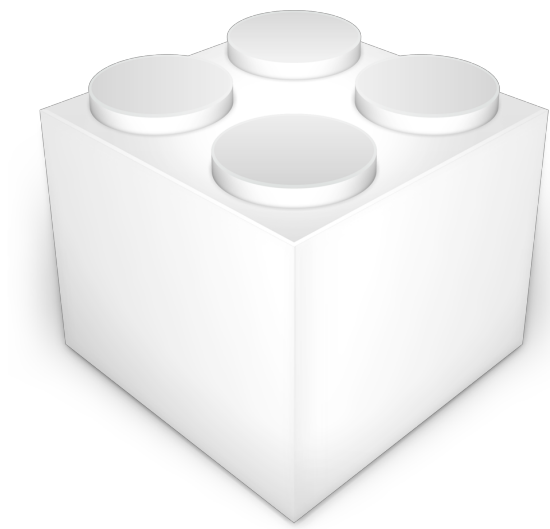
badoo_tech



Chat

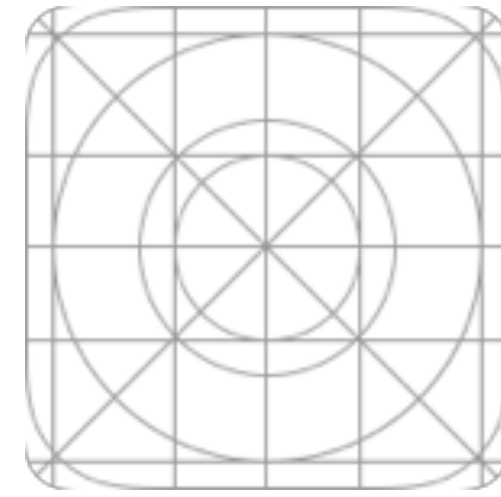


+

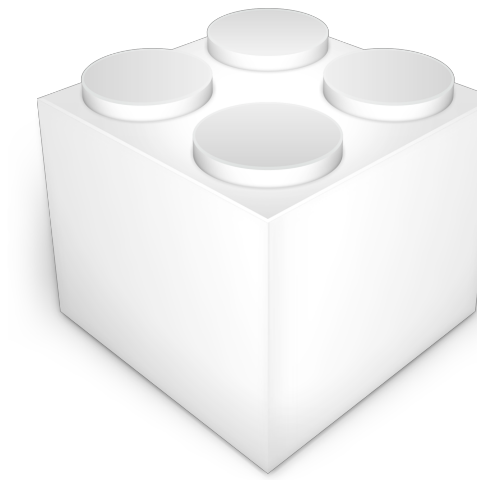
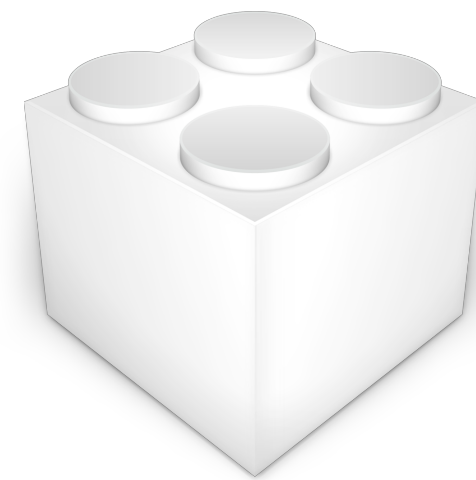
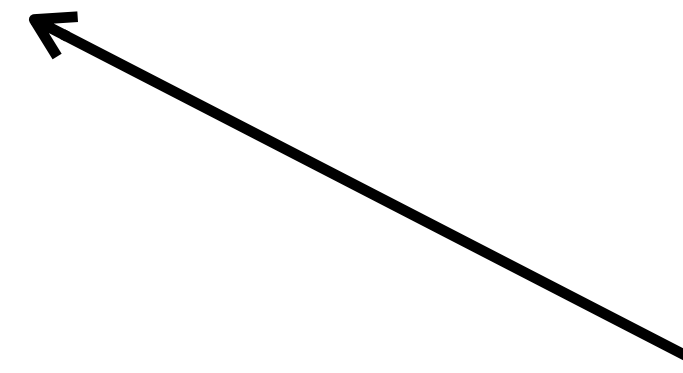


Platform_Chat

Platform_ChatUI



Badoo Gallery



BadooUIKit

Platfrom_ChatUI



Антон Щукин

**Монорепо: путешествие обратно.
История модуляризации в Vadoo**

Mobius, 9 декабря, 10:30

Зал 1

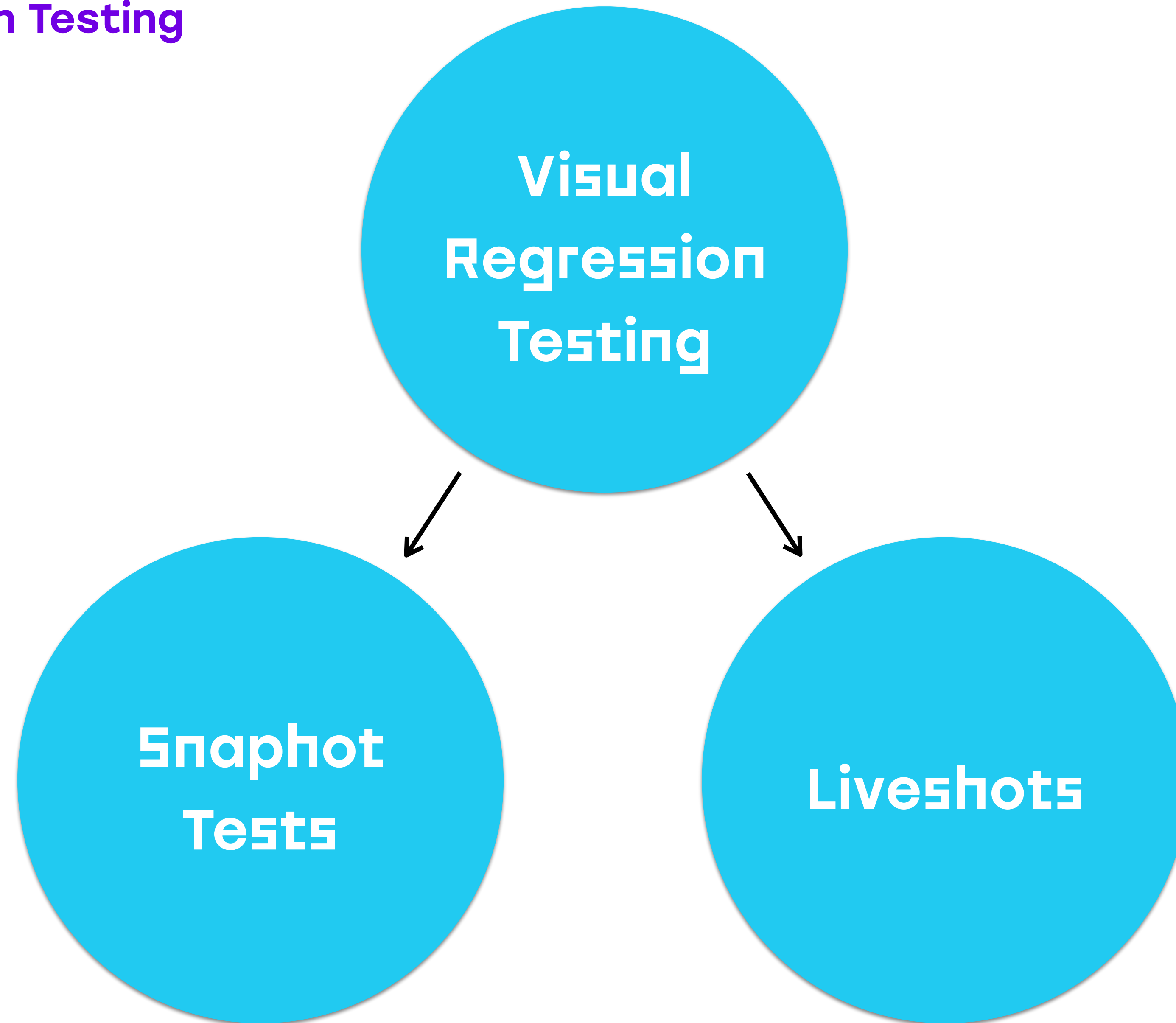
Platform UIKit
+
Platform_Chat & Platform_ChatUI
+
Registration & RegistrationUI
+
2 разработчика
=
Новое приложение за 2 месяца

Visual Regression Testing

Задача:

**Минимизировать количество
ресурсов, которые мы
вкладываем в ручное
тестирование**

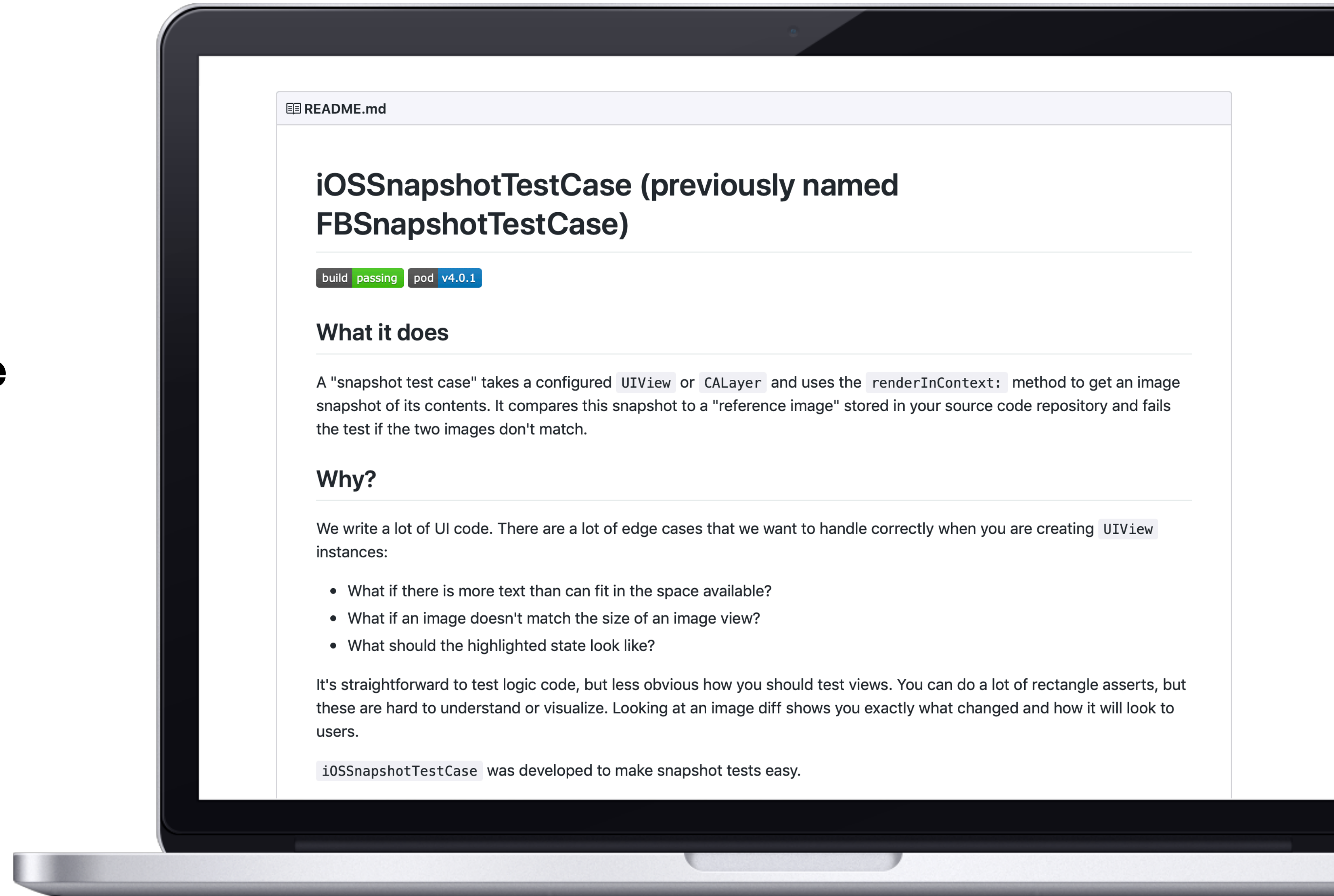
**Visual Regression Testing –
регрессионное тестирование UI
КОМПОНЕНТОВ НА ВЫЯВЛЕНИЕ
ВИЗУАЛЬНЫХ багов**



Snapshot Tests

iOSSnapshotTestCase

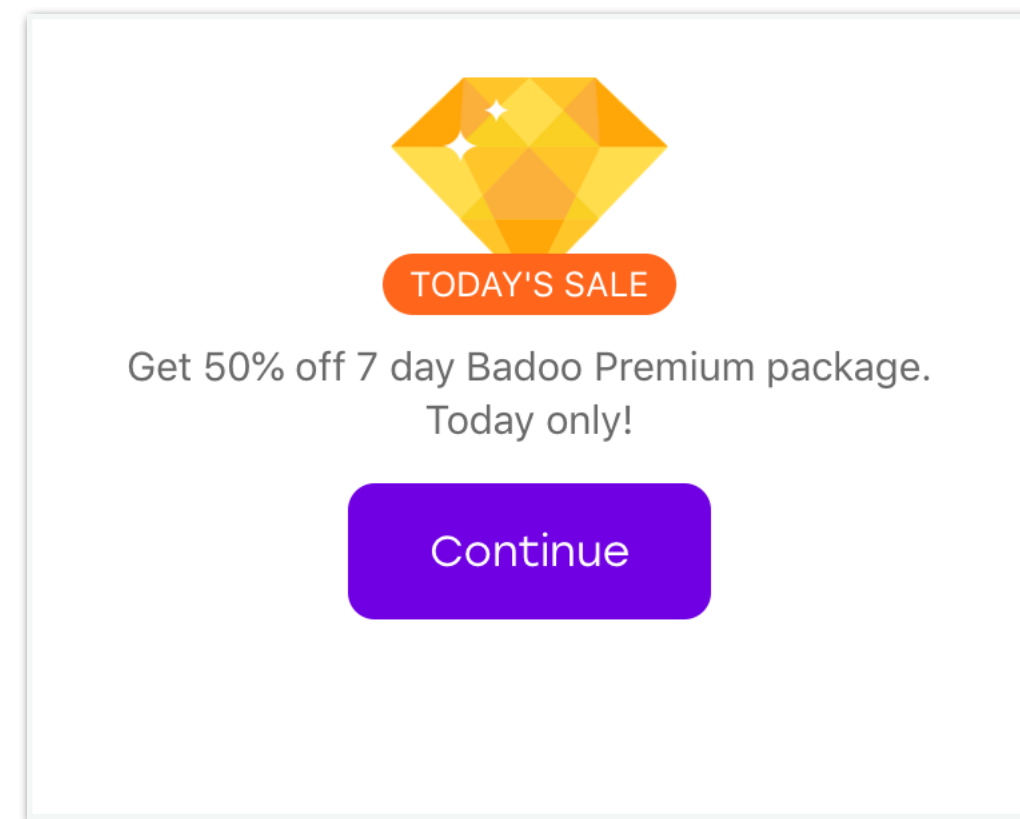
<https://github.com/uber/ios-snapshot-test-case>



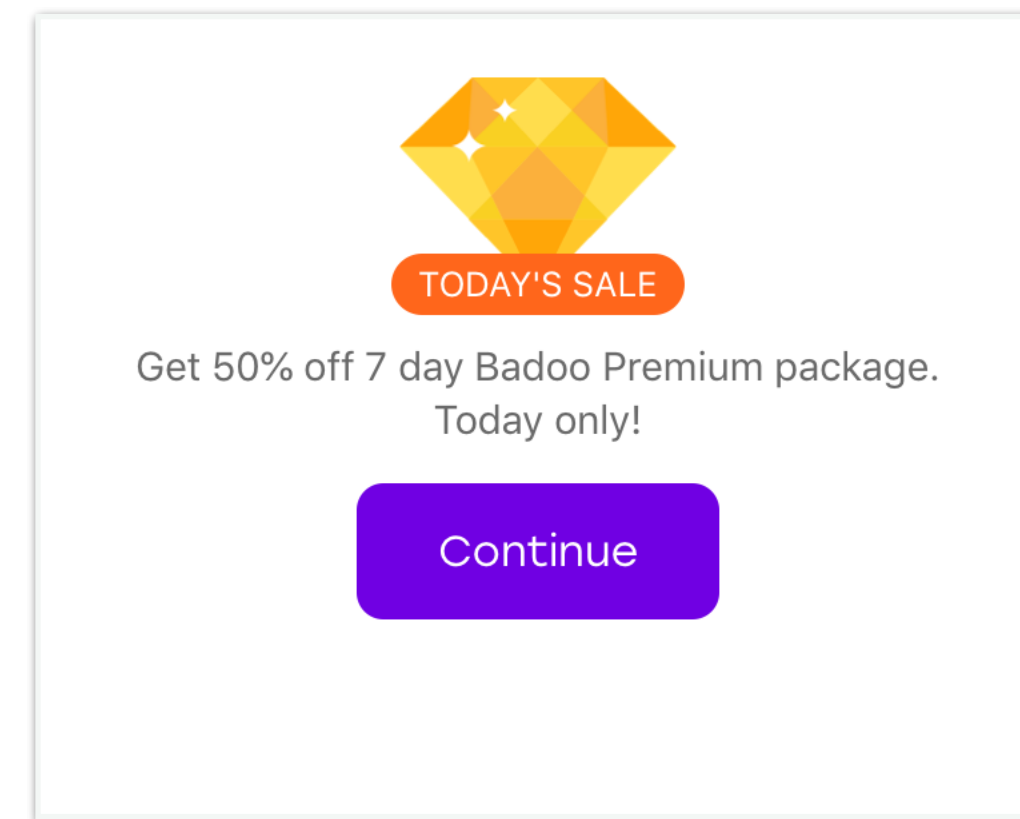
iOSSnapshotTestCase –
наследник класа **TestCase** из
Xcode Unit Tests

VRT / Snapshot Tests

badoo_tech



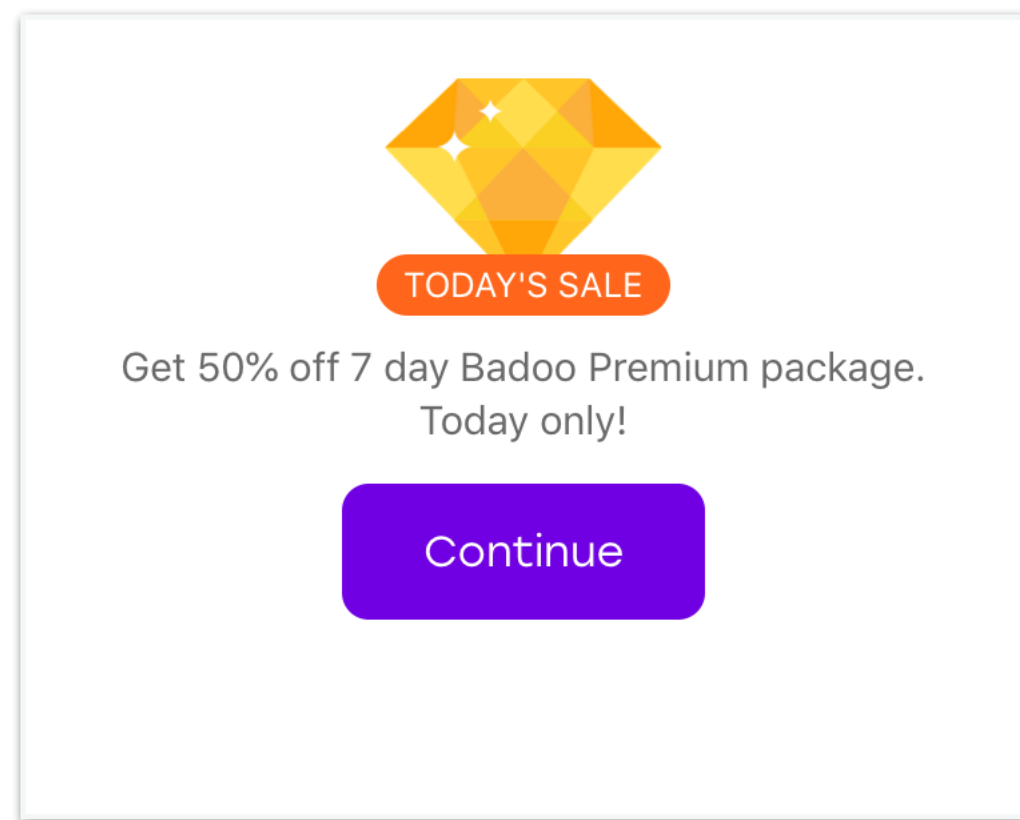
+



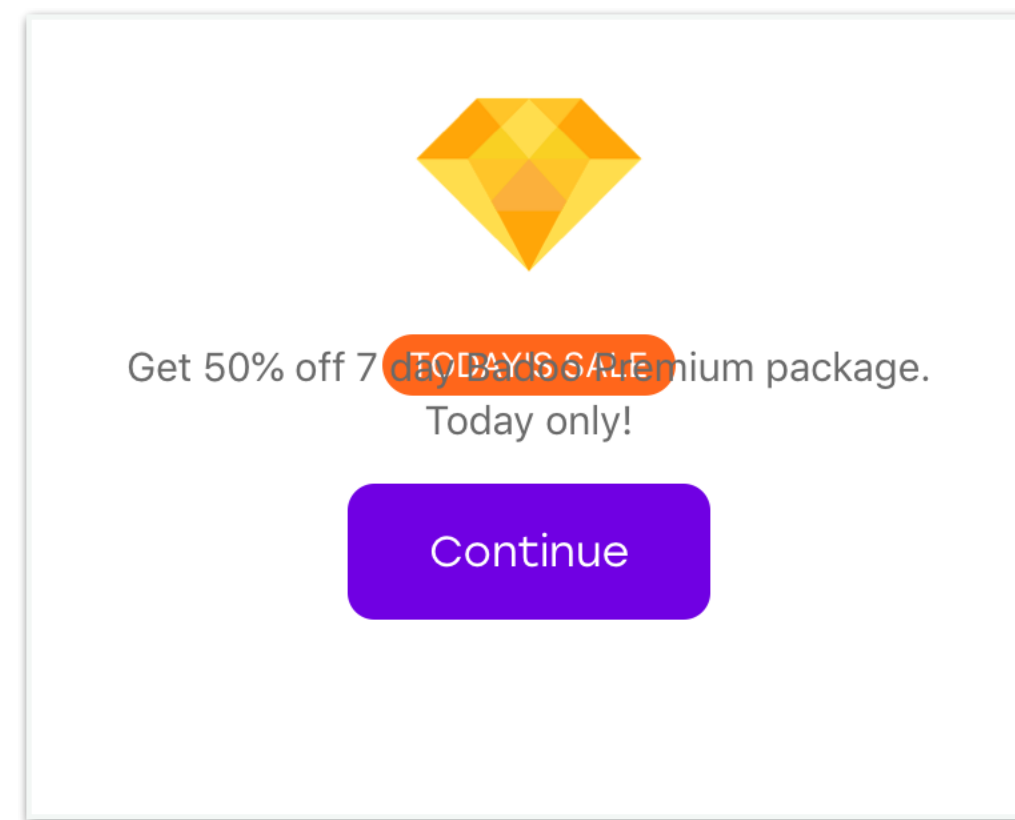
=



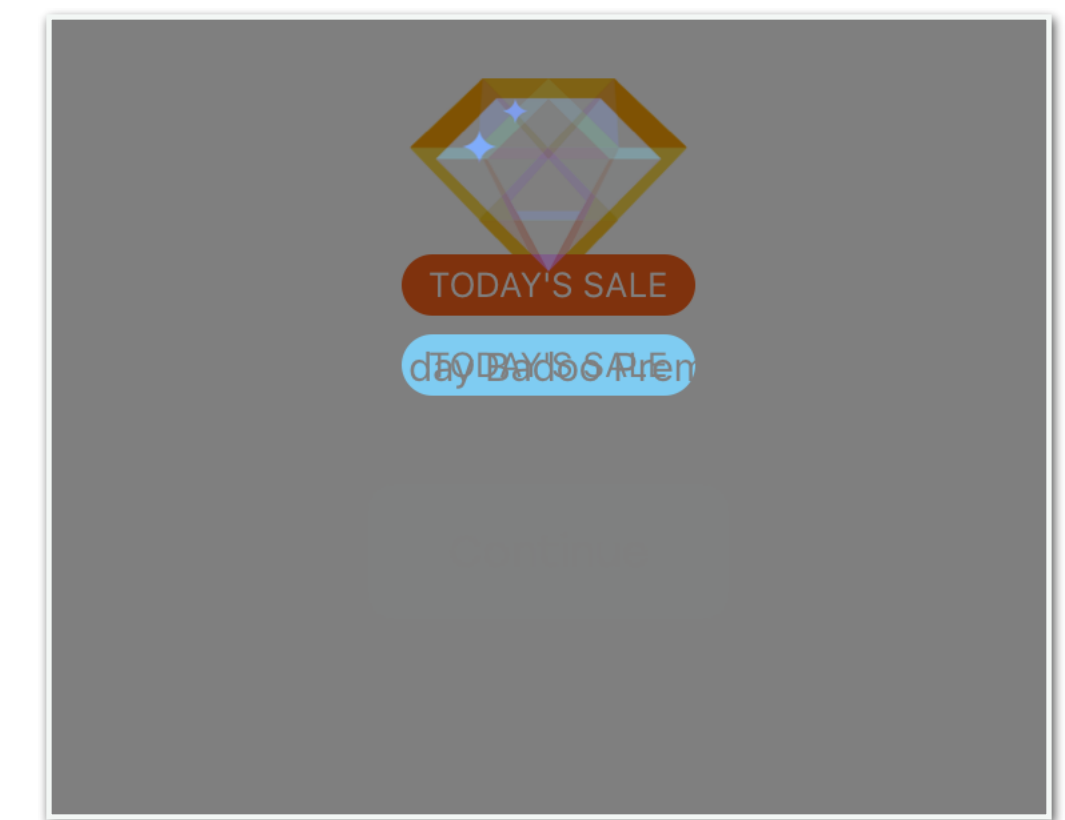
VRT / Snapshot Tests



+

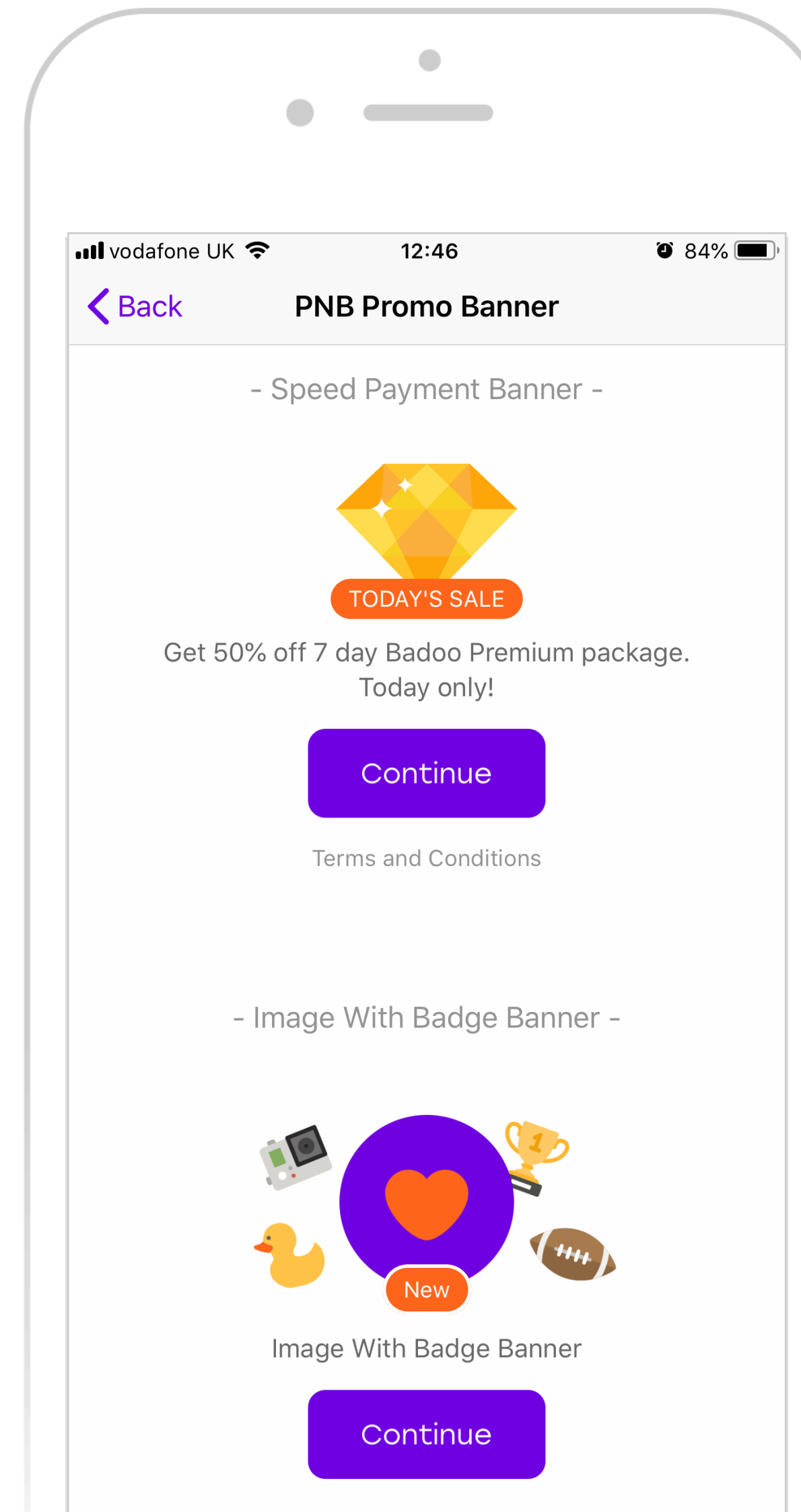
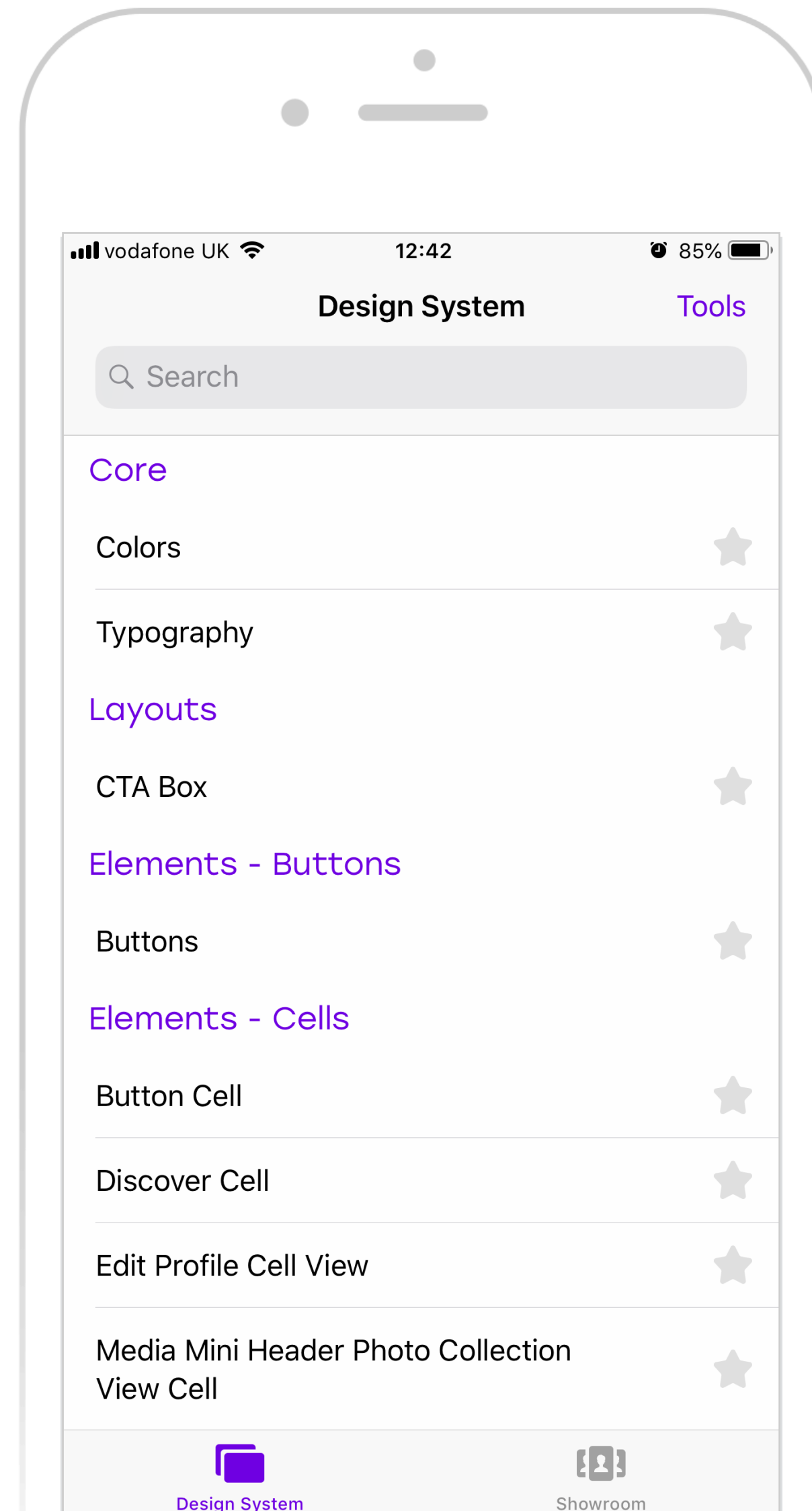


=

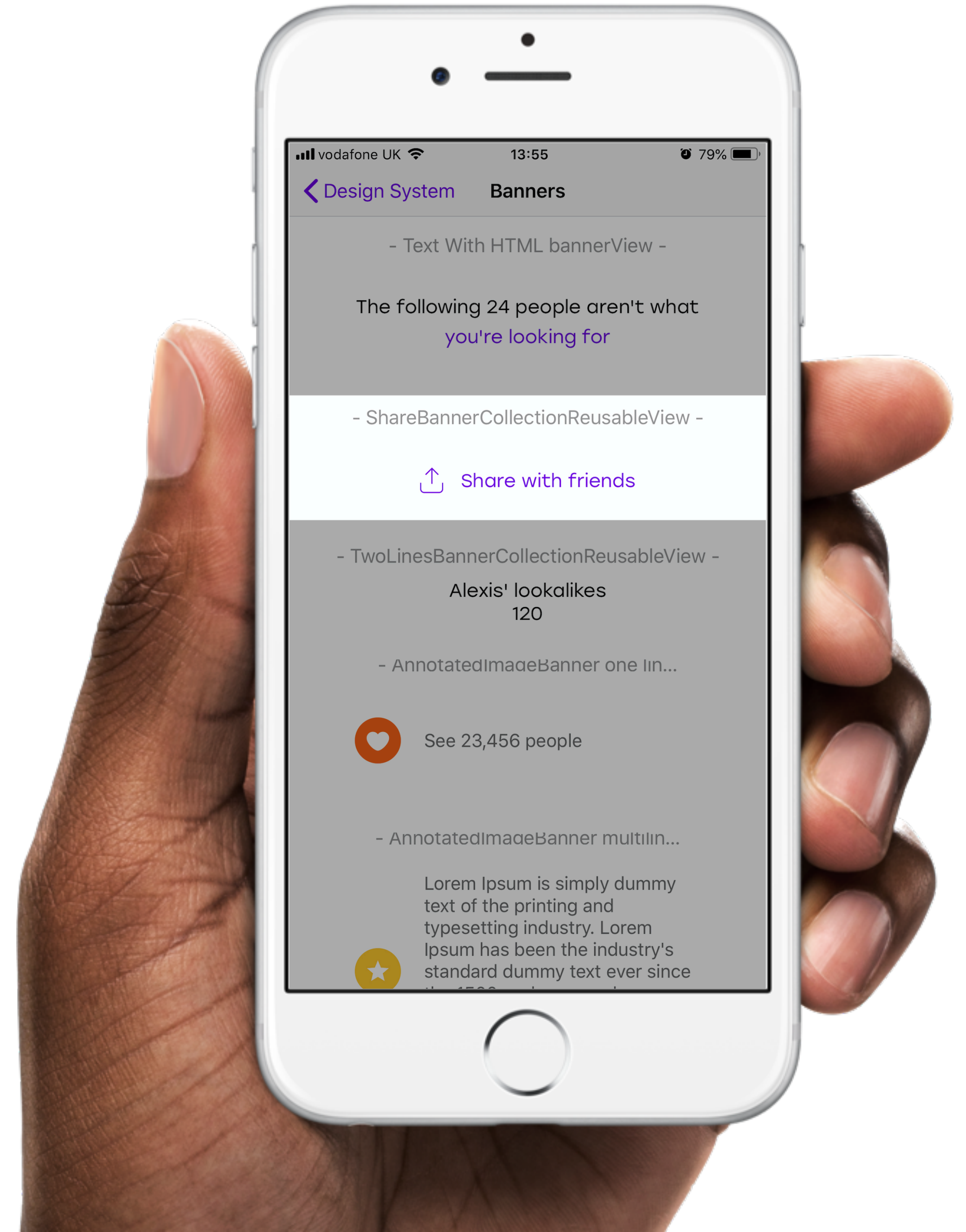


Gallery App + Snapshot Tests

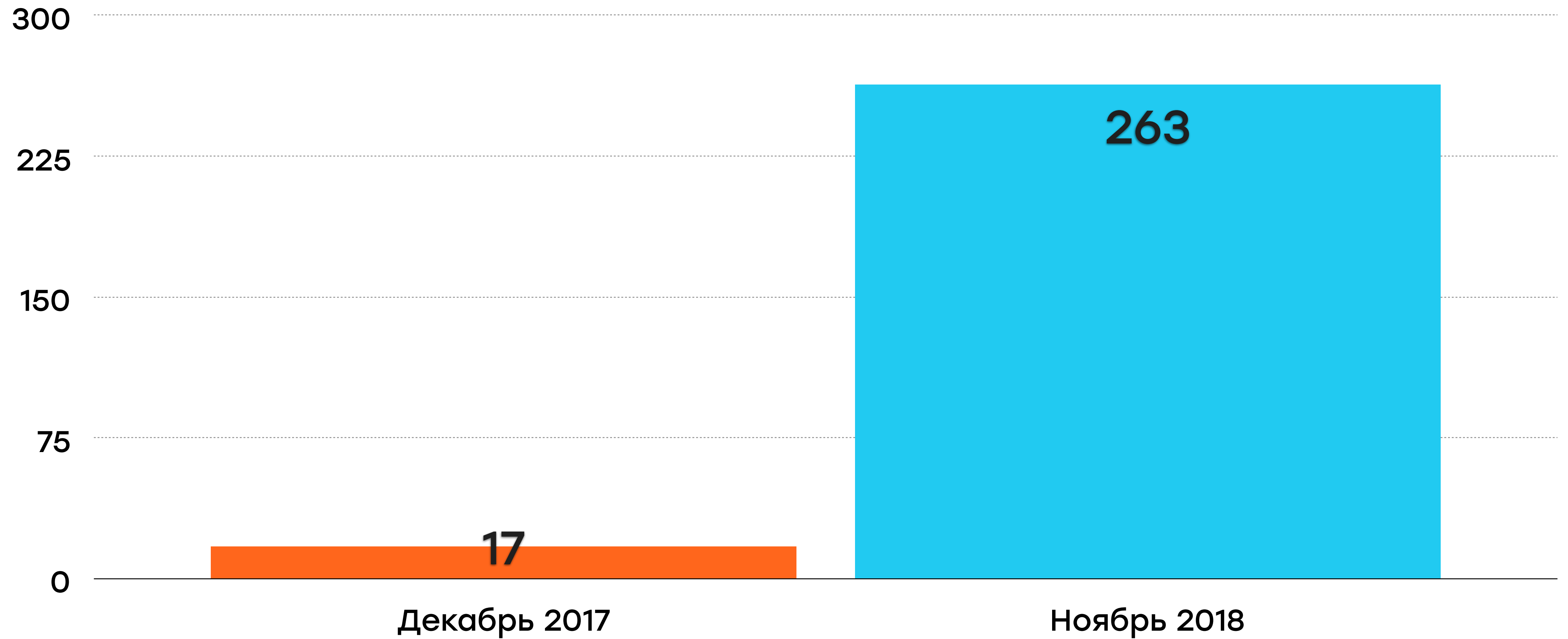
Running title Subtitle



Каждое состояние –
это отдельный
Snapshot тест



Количество Snapshot тестов



```
// MARK: - Input View

func testCaptchaInputView() {
    self.setupAndTest()
}

func testClientOnboardingNameInputView_1() {
    self.setupAndTest()
}
```

**Мы переиспользовали в
Snapshot тестах код, который
был создан для наполнения
Badoo Gallery**

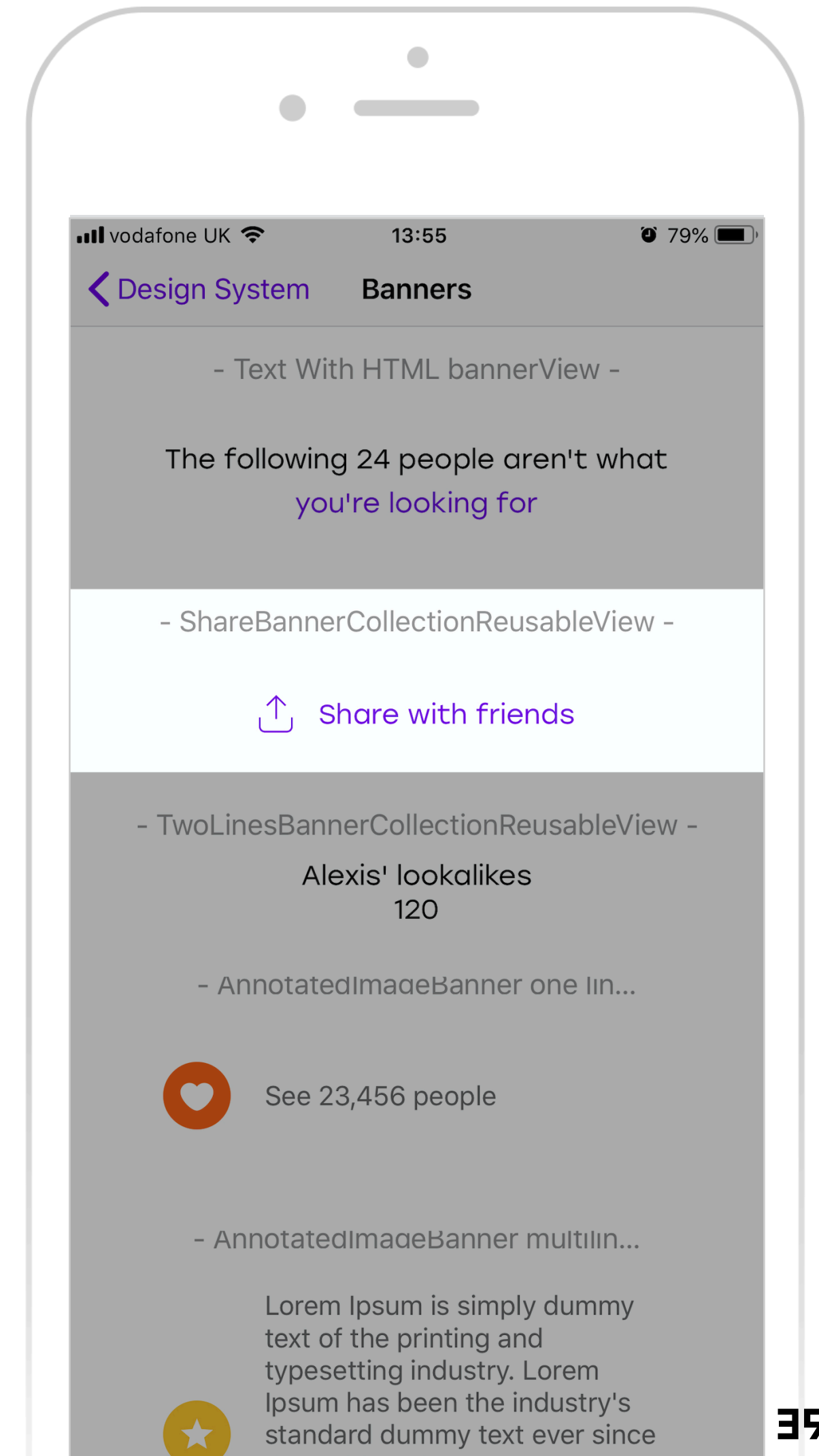
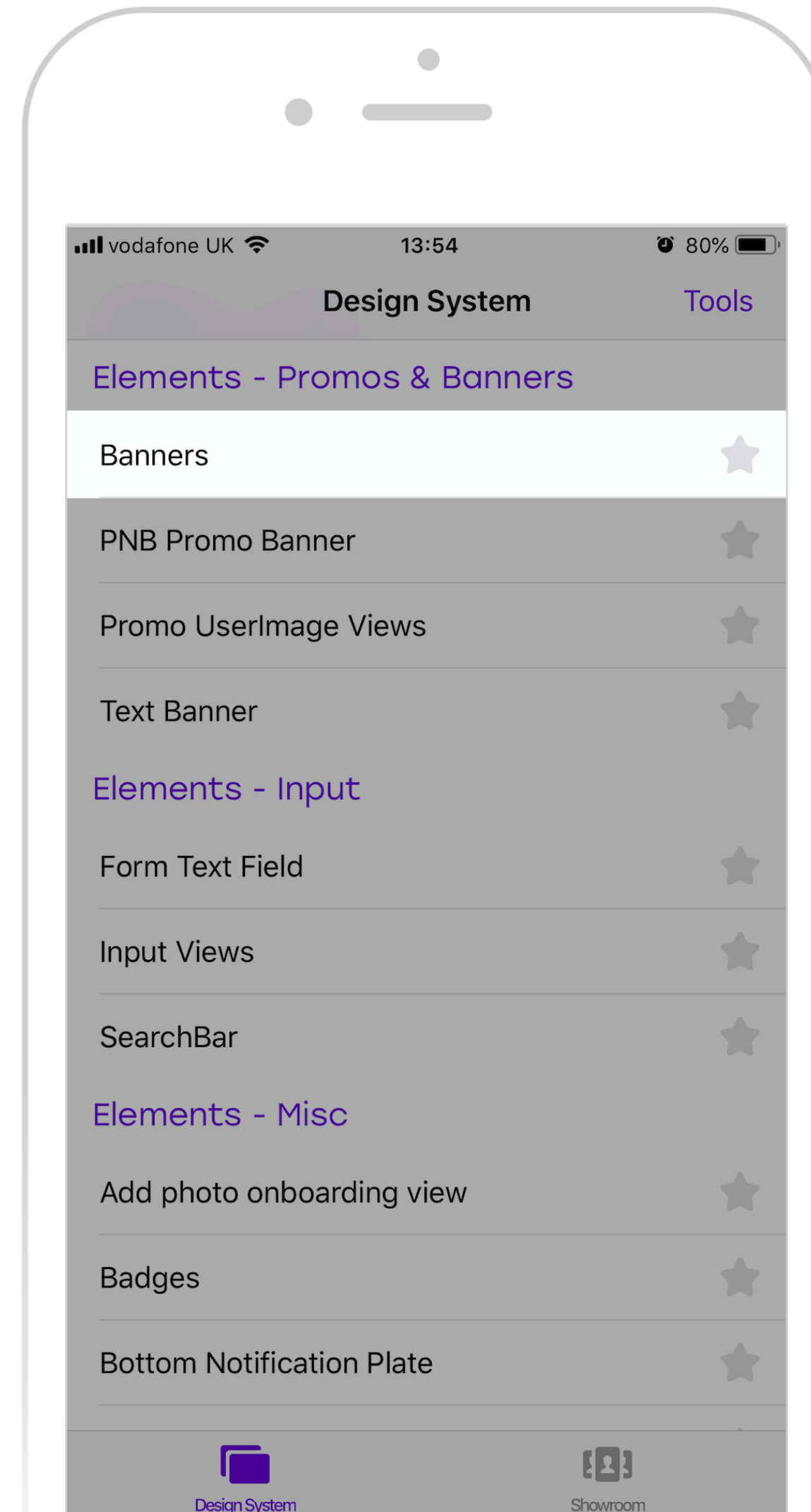
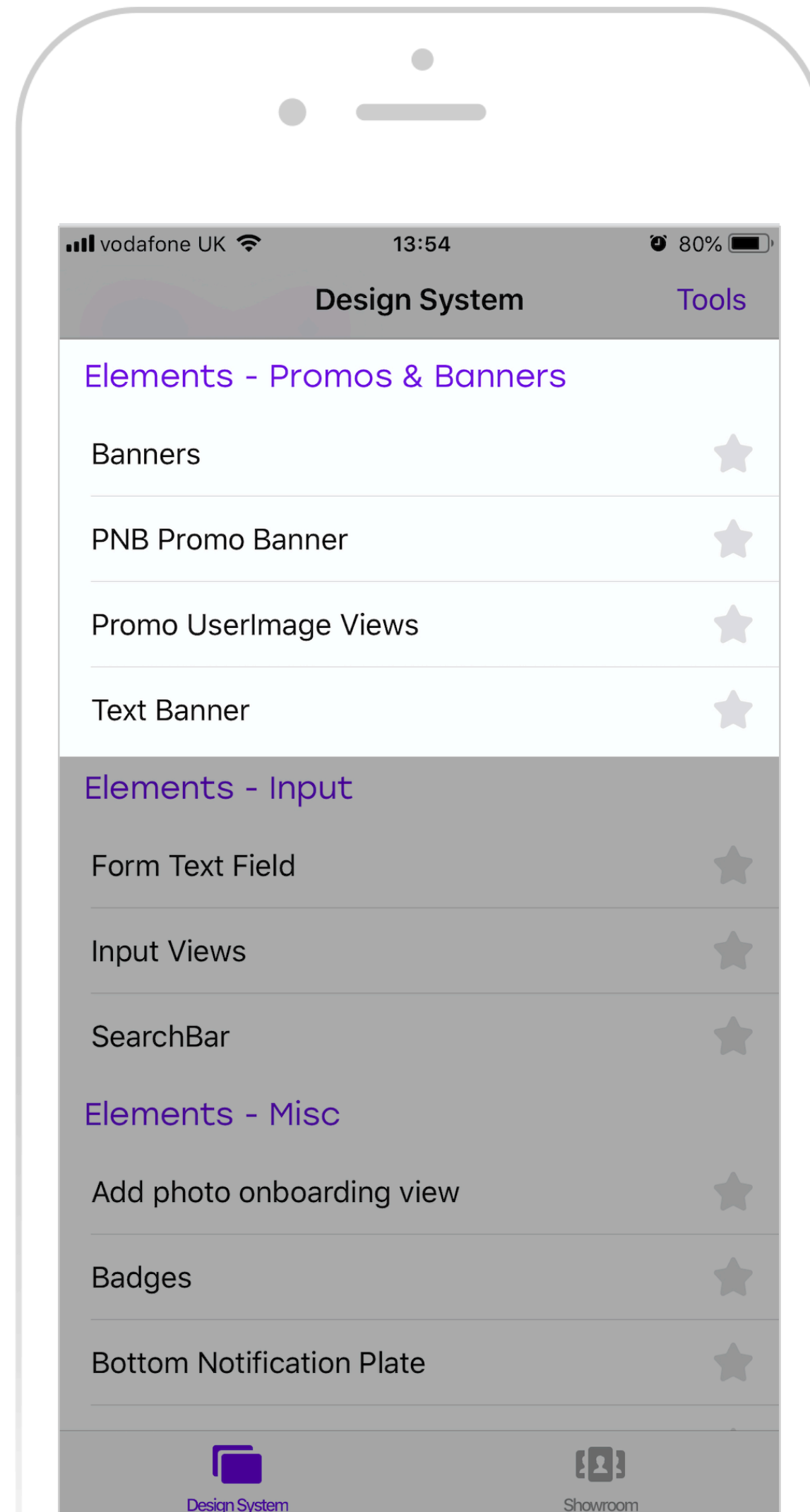
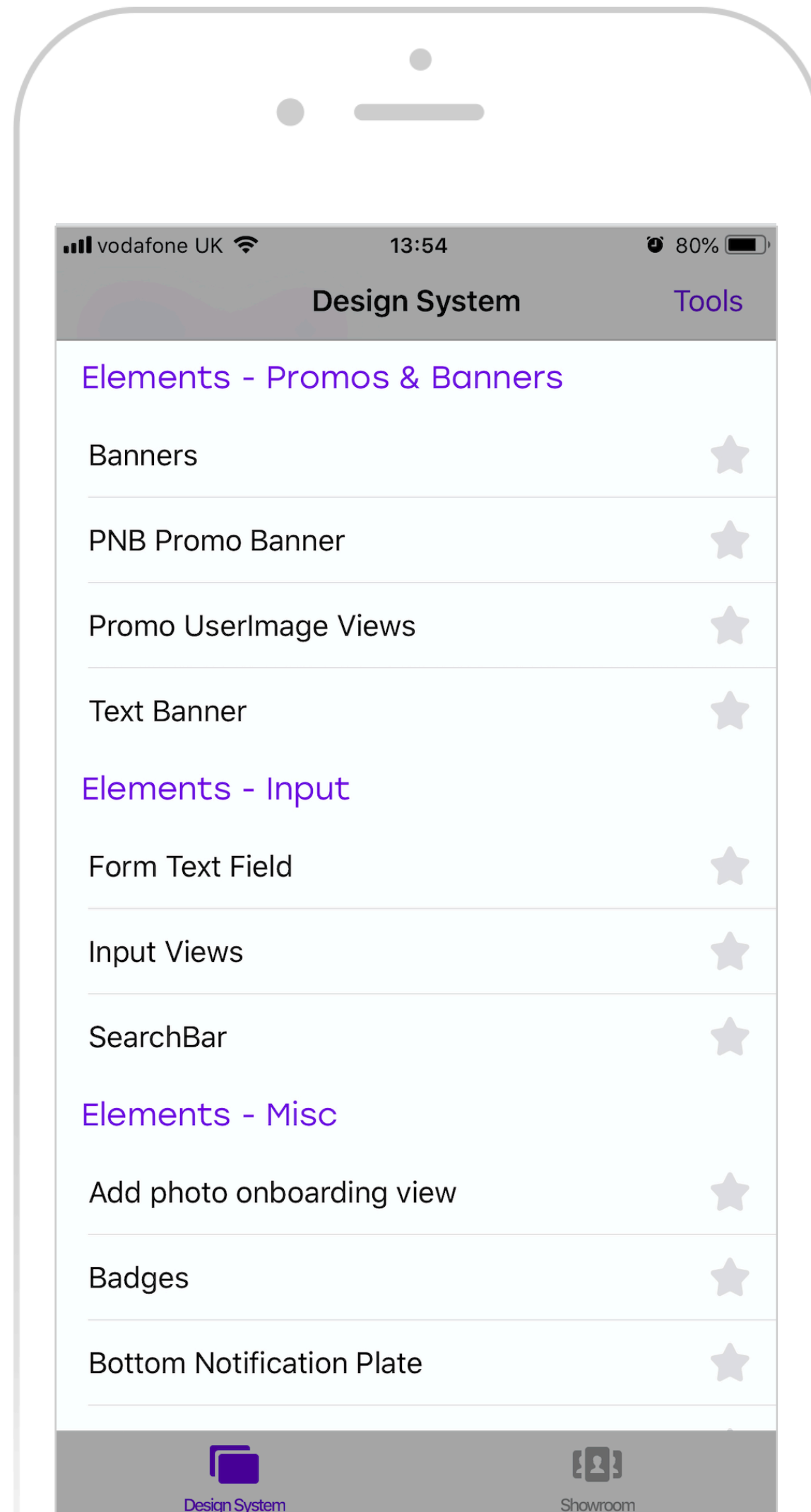
VRT / Snapshot Tests

Store

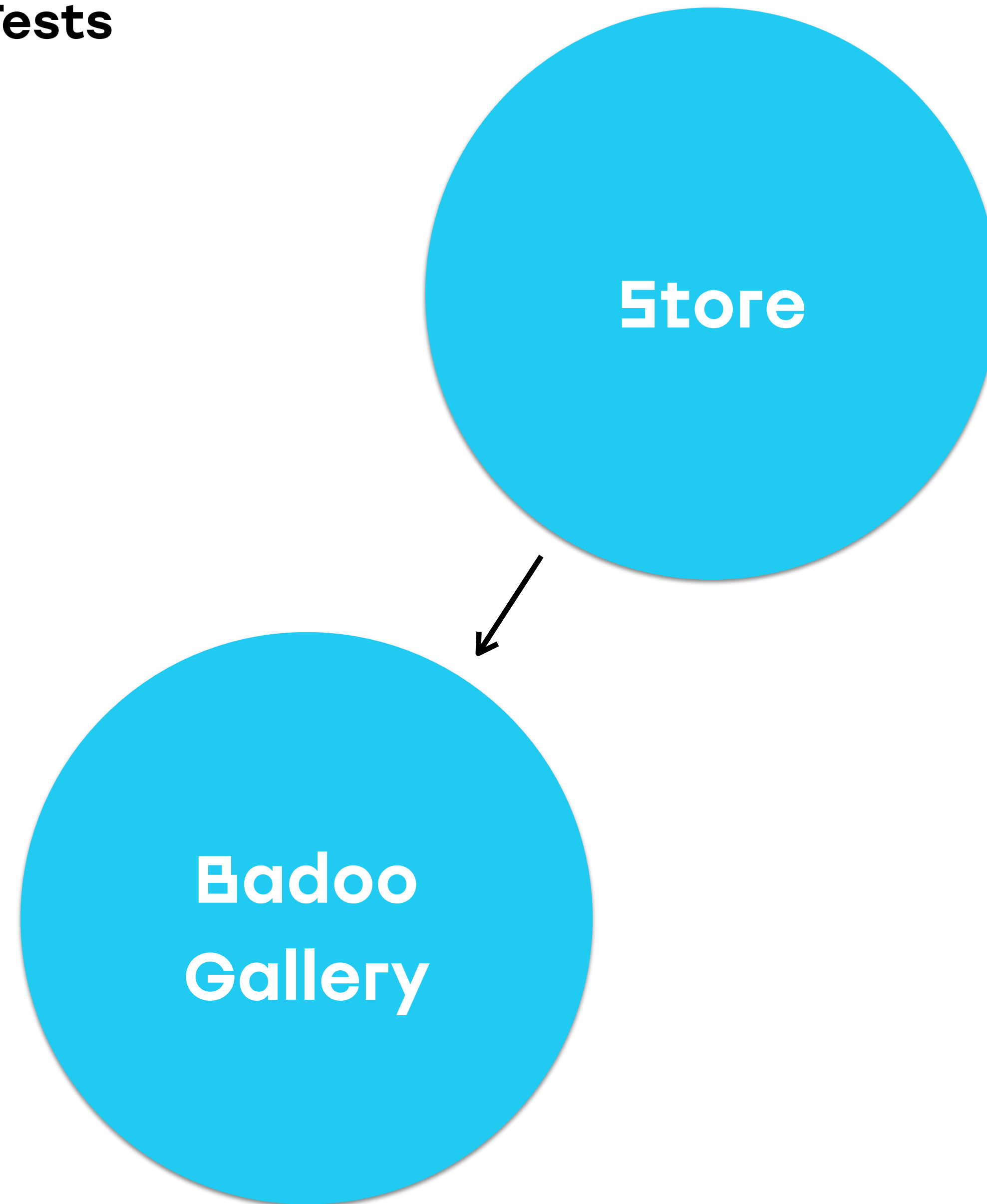
Section

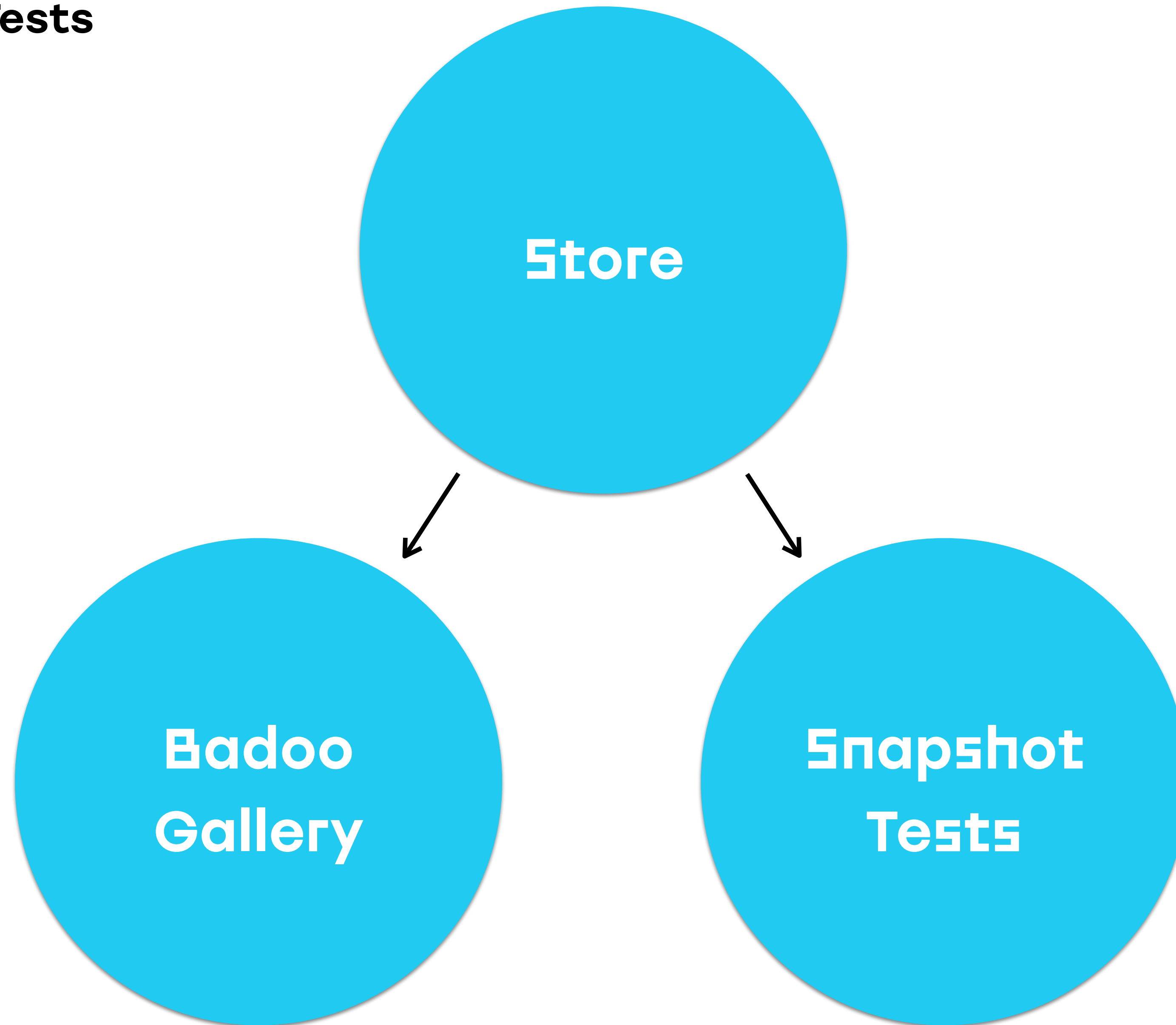
ElementsDataModel

Element

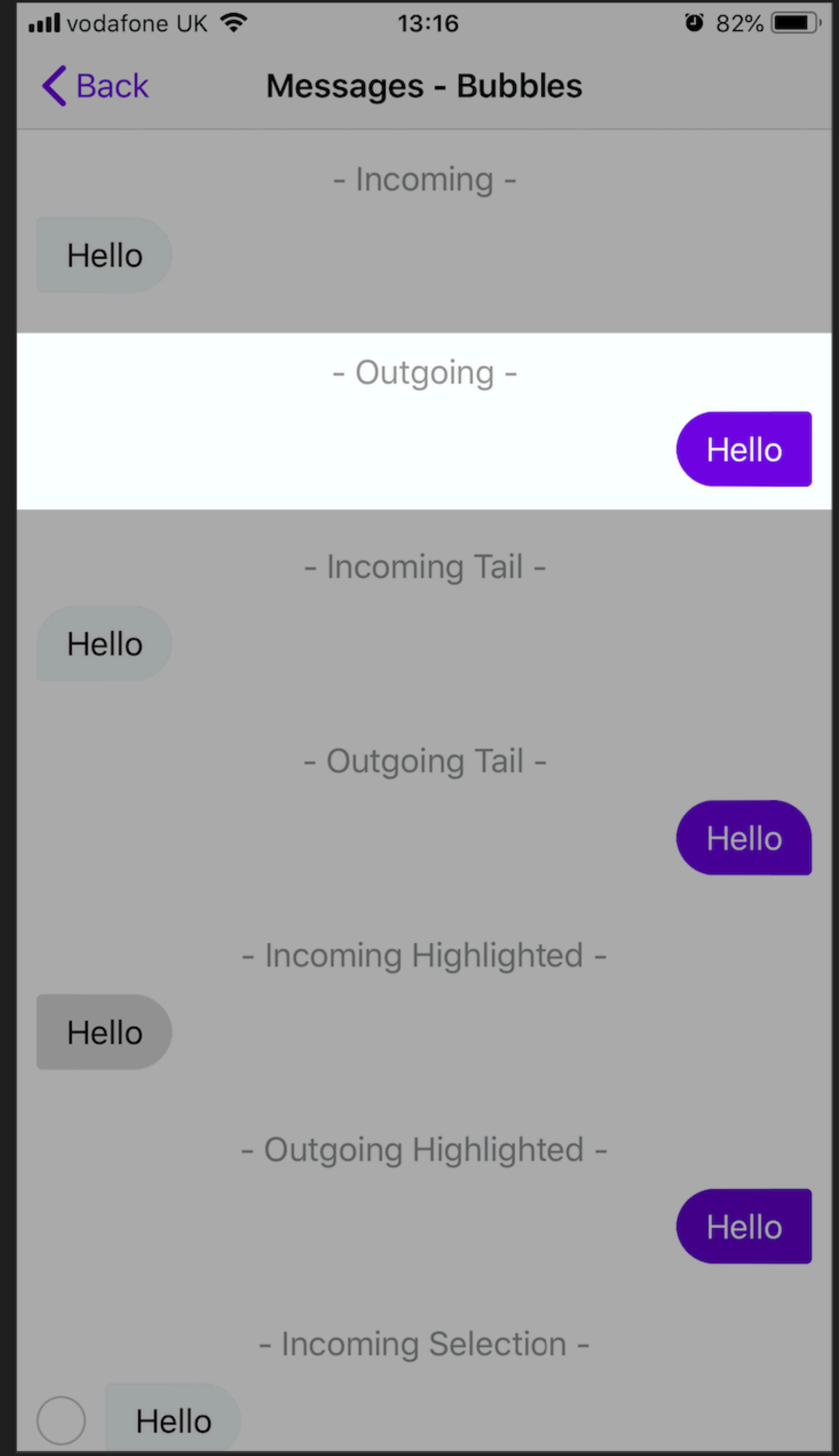


**Из Store мы можем получить
доступ ко всем состояниям всех
UI компонентов**





```
struct Element {
  let title: String
  let view: UIView
  let snapshotTestName: String
  let size: CGSize?
  let tags: [String]?
}
```



```
// Element

let banner = PeopleNearbyBanner()
banner.applyStyle(style)
banner.applyViewModel(viewModel)

let element = Element(title: "People nearby banner",
                      view: banner,
                      snapshotTestName: "testPeopleNearbyBanner")

// Test

func testPeopleNearbyBanner() {
    self.setupAndTest()
}
```

```
// Element
```

```
let banner = PeopleNearbyBanner()  
banner.applyStyle(style)  
banner.applyViewModel(viewModel)
```

```
let element = Element(title: "People nearby banner",  
                      view: banner,  
                      snapshotTestName: "testPeopleNearbyBanner")
```

```
// Test
```

```
func testPeopleNearbyBanner() {  
    self.setupAndTest()  
}
```

```
// Element

let banner = PeopleNearbyBanner()
banner.applyStyle(style)
banner.applyViewModel(viewModel)

let element = Element(title: "People nearby banner",
                      view: banner,
                      snapshotTestName: "testPeopleNearbyBanner")

// Test

func testPeopleNearbyBanner() {
    self.setupAndTest()
}
```

```
func setupAndTest(file: StaticString = #file,  
                 line: UInt = #line,  
                 function: String = #function) {  
    let testName = function.replacingOccurrences(of: "()", with: "")  
    guard let view = self.view(forSnapshotTestCase: testName) else {  
        XCTFail("...")  
        return  
    }  
  
    self.folderName = testName  
    self.present(view: view, sideSizing: .selfSized)  
    self.snapshotAndTest(view: view, file: file, line: line)  
}
```

```
func setupAndTest(file: StaticString = #file,  
                 line: UInt = #line,  
                 function: String = #function) {  
    let testName = function.replacingOccurrences(of: "()", with: "")  
    guard let view = self.view(forSnapshotTestCase: testName) else {  
        XCTFail("...")  
        return  
    }  
  
    self.folderName = testName  
    self.present(view: view, sideSizing: .selfSized)  
    self.snapshotAndTest(view: view, file: file, line: line)  
}
```



```
func setupAndTest(file: StaticString = #file,  
                 line: UInt = #line,  
                 function: String = #function) {  
    let testName = function.replacingOccurrences(of: "()", with: "")  
    guard let view = self.view(forSnapshotTestCase: testName) else {  
        XCTFail("...")  
        return  
    }  
  
    self.folderName = testName  
    self.present(view: view, sideSizing: .selfSized)  
    self.snapshotAndTest(view: view, file: file, line: line)  
}
```

```
func setupAndTest(file: StaticString = #file,  
                 line: UInt = #line,  
                 function: String = #function) {  
    let testName = function.replacingOccurrences(of: "()", with: "")  
    guard let view = self.view(forSnapshotTestCase: testName) else {  
        XCTFail("...")  
        return  
    }  
  
    self.folderName = testName  
    self.present(view: view, sideSizing: .selfSized)  
    self.snapshotAndTest(view: view, file: file, line: line)  
}
```

Snapshot тесты в **Continuous Integration**:

- Unit Tests target для Badoo Gallery
- часть Entry Point

Liveshots

The image displays a Badoo mobile application interface. On the left, a 'Flow Explorer' sidebar allows users to select a flow (iOS, Badoo), device (iPhone 6 @2x), version (0.0.0-ios-18070), and language (English (UK)). It also includes toggle switches for 'Lexemes' and 'HotPanel', and buttons for 'Compare flows' and 'Download Flow'. The main area shows three app screens: 1. 'encounters' (5:45 16th Nov 18) showing a profile for 'Alex, 72' with a photo and a heart icon. 2. 'own_profile' (5:45 16th Nov 18) showing a profile for 'Alex, 72' with a notification to 'Switch on notifications and never miss a match', 'Add credits', and 'Badoo Premium Activate'. 3. 'own_profile_edit' (5:45 16th Nov 18) showing an 'Edit profile' screen with fields for Name, Work, Education, About you, Appearance, Children, and Drinking.

**Для создания Liveshots мы
используем часть наших
функциональных тестов**



Функциональные тесты в Badoo:

- **создаются отдельной командой QA**
- **выполняются при помощи Calabash**
- **более 1500 тестов**



Liveshot в Continuous Integration:

- **выделены в отдельную секцию**
- **запускаются по ночам**

Результаты

**QA=NO flow тикеты
закрываются без ручной
проверки QA инженером,
верификация происходит при
ПОМОЩИ ТЕСТОВ**

Работа над QA=NO ТИКЕТОМ

1. QA и iOS определяют тестовые сценарии

Работа над QA=NO ТИКЕТОМ

1. QA и iOS определяют тестовые сценарии
2. iOS разработчик пишет Unit, Integration и **Snapshot** тесты

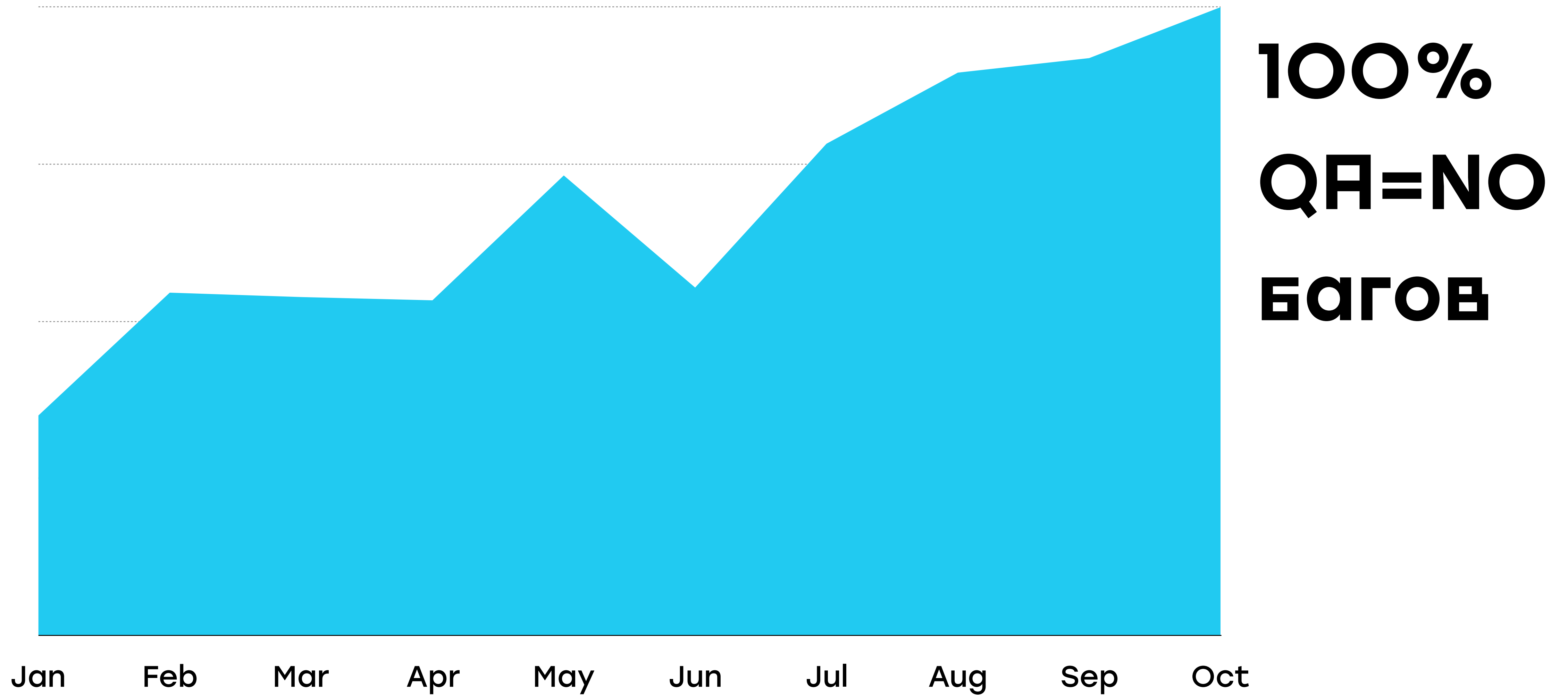
Работа над QA=NO ТИКЕТОМ

1. QA и iOS определяют тестовые сценарии
2. iOS разработчик пишет Unit, Integration и **Snapshot** тесты
3. QA пишет тесты для Calabash

Работа над QA=NO ТИКЕТОМ

1. QA и iOS определяют тестовые сценарии
2. iOS разработчик пишет Unit, Integration и **Snapshot** тесты
3. QA пишет тесты для Calabash

Scenario	Unit tests?	Integration tests?	Visual snapshot tests?	Needs Calabash
[Final stream screen, Tokens earned] Streaming works after you convert all	x	x	x	x
[Insufficient tokens] attempting to convert tokens to credits will show "Not e	done	x	x	y
[Convert to credits tab] tooltip for Tokens can be dismissed	y	x	x	x
[Convert to credits tab] send billing question via contact support	x	y	x	x
[Convert to cash tab] tooltip for Pending tokens can be dismissed	y	x	x	x
[Convert to cash tab] send billing question via contact support	x	y	x	y
[Convert to cash tab] cash out rules screen is rendered correctly	x	x	y	x
[Convert to cash tab] terms and conditions screen is rendered correctly	x	x	y	x
[Not verified, Rules not met, Sufficient tokens] can not cash out after verific	x	x	x	y



Преимущества QA=NO flow

- Стабильность изменений обеспечивается в течение долгого времени

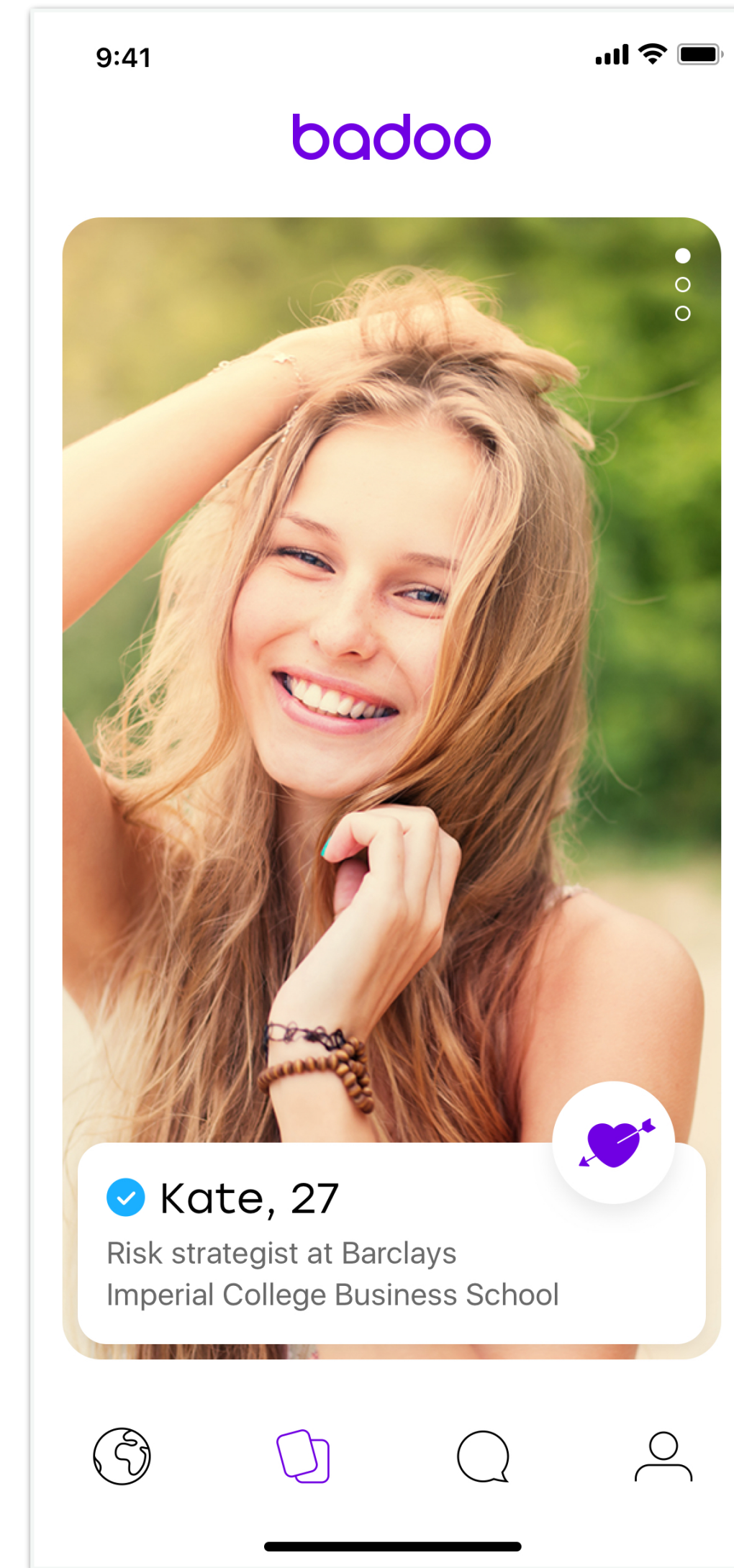
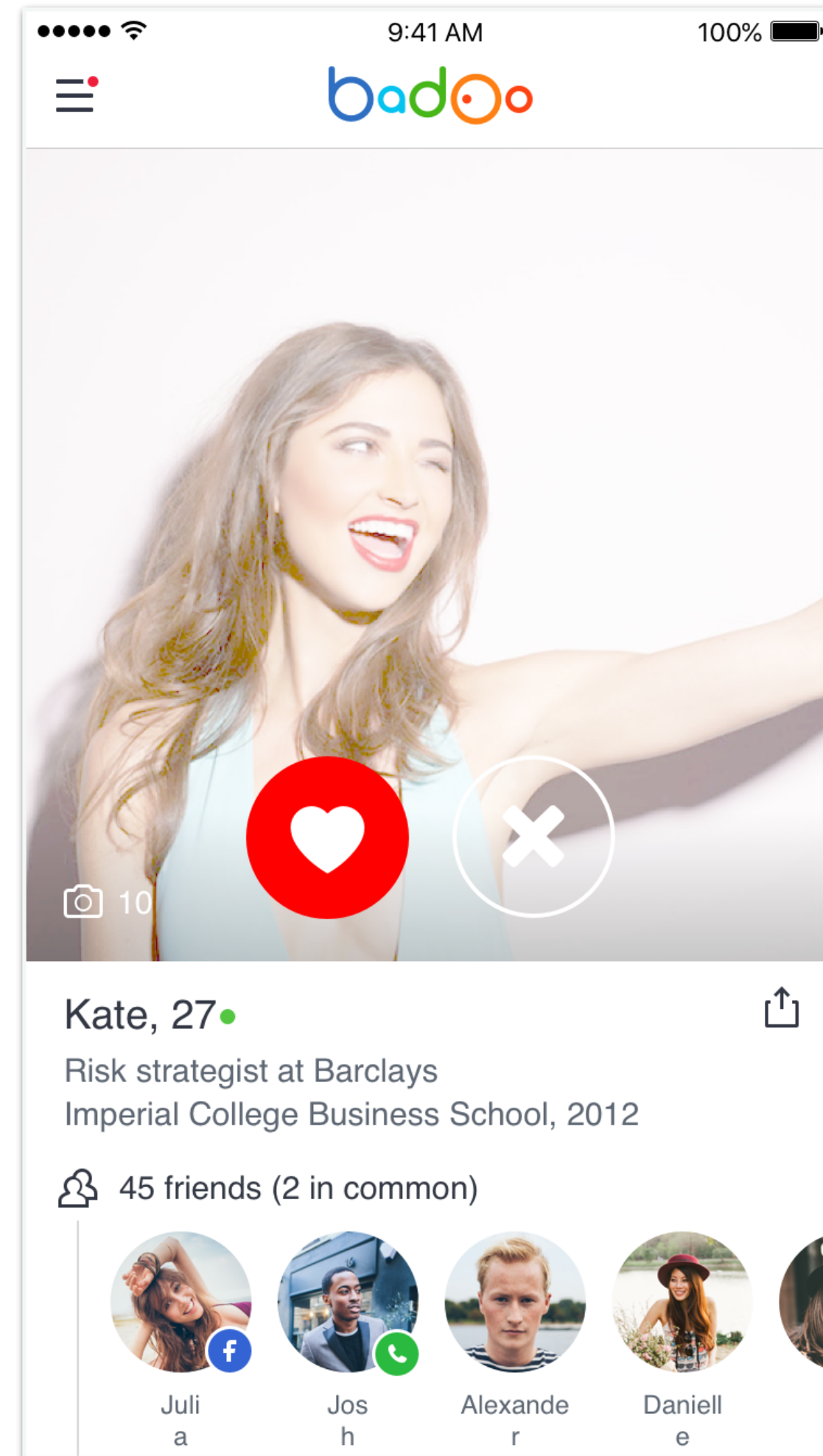
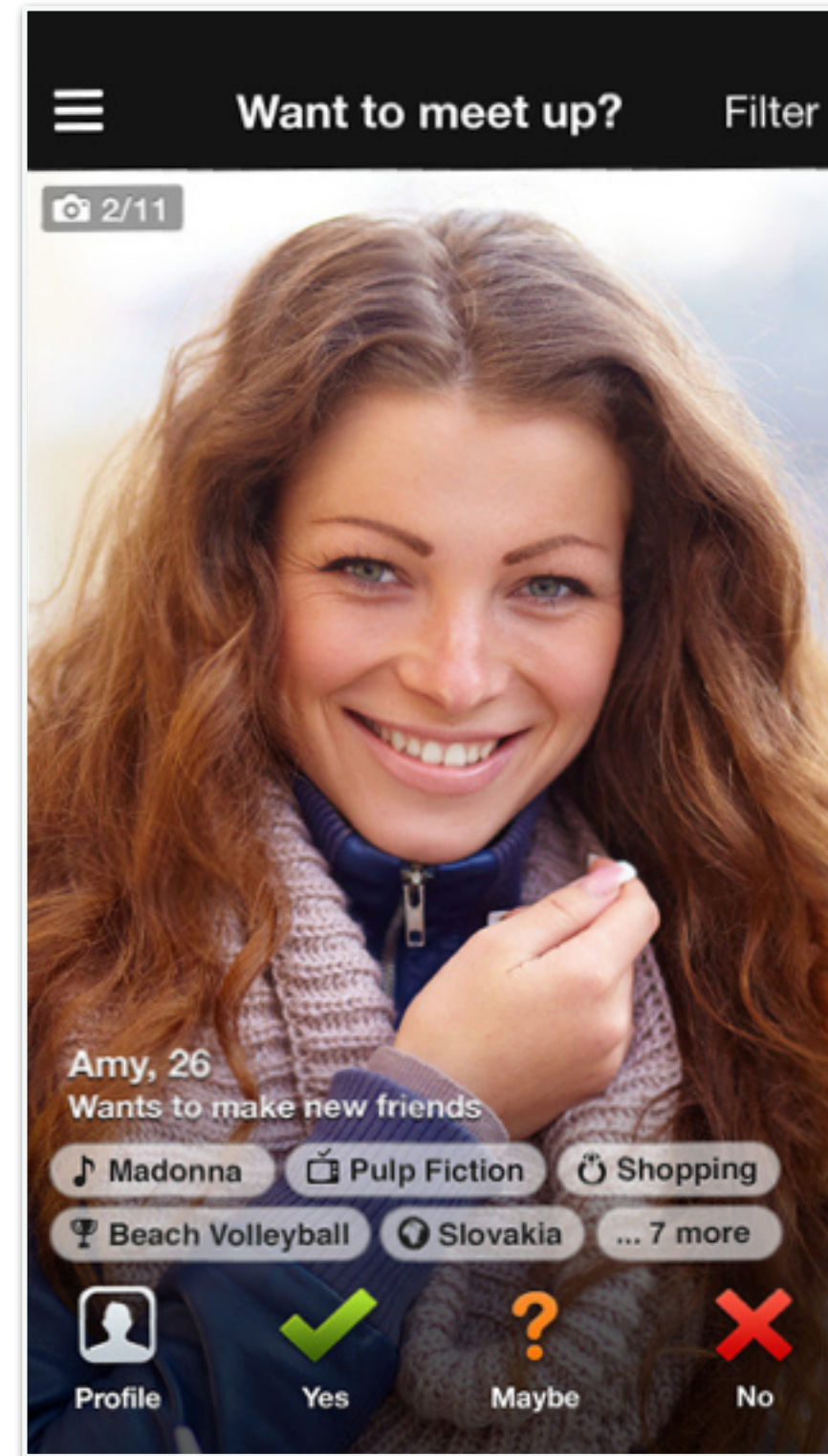
Преимущества QA=NO flow

- Стабильность изменений обеспечивается в течение долгого времени
- QA инженер использует свое время более эффективно

Cosmos Design System

Cosmos Design System

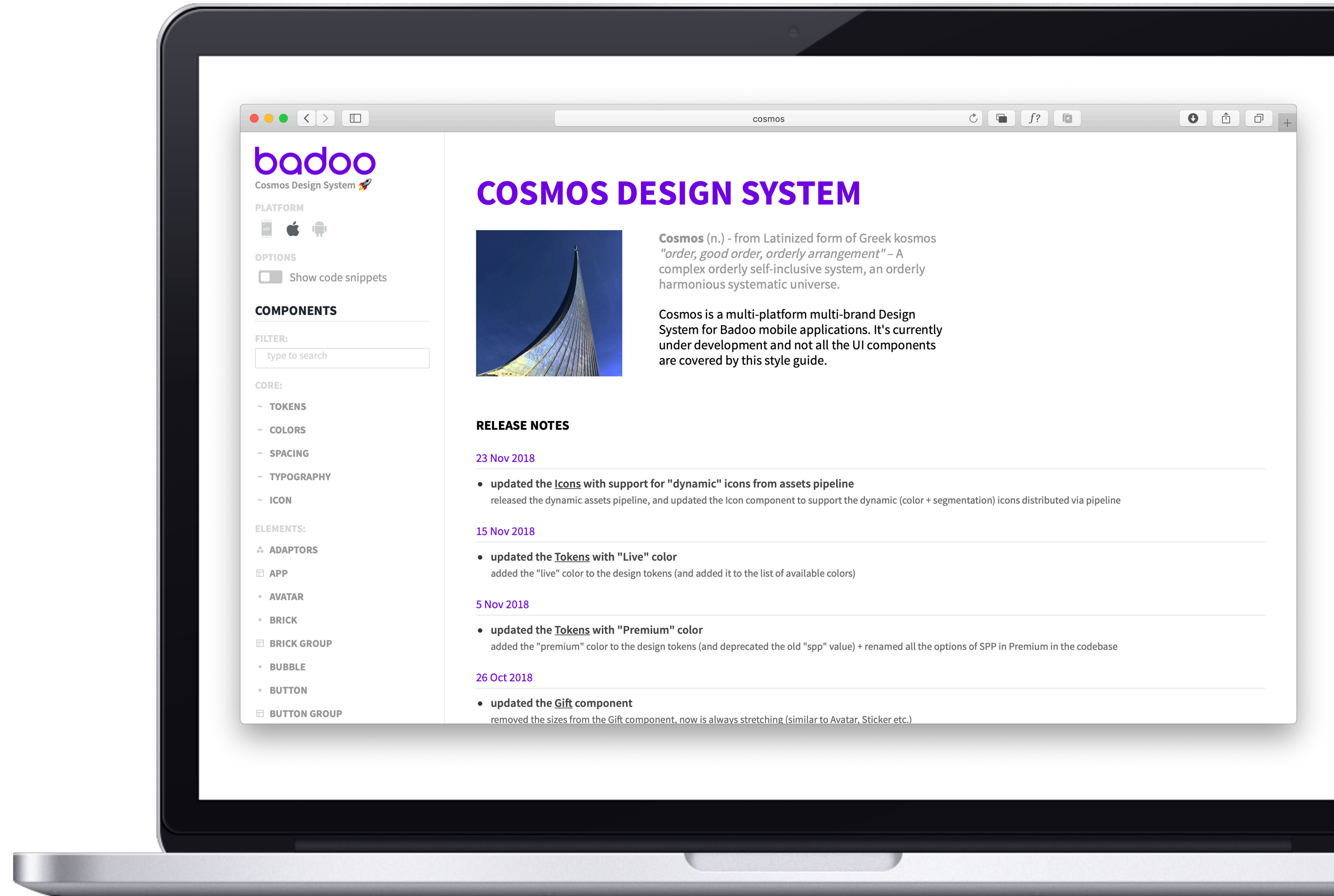
badoo_tech



Cosmos Design System

СОСМОС

Дизайн-система Badoo



badoo
Cosmos Design System

PLATFORM
[Icons for iOS and Android]

OPTIONS
 Show code snippets

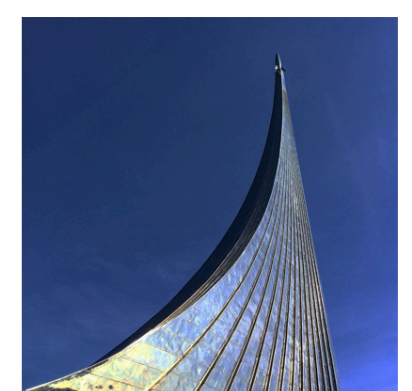
COMPONENTS

FILTER:
type to search

CORE:
~ TOKENS
~ COLORS
~ SPACING
~ TYPOGRAPHY
~ ICON

ELEMENTS:
ADAPTORS
APP
AVATAR
BRICK
BRICK GROUP
BUBBLE
BUTTON
BUTTON GROUP

COSMOS DESIGN SYSTEM

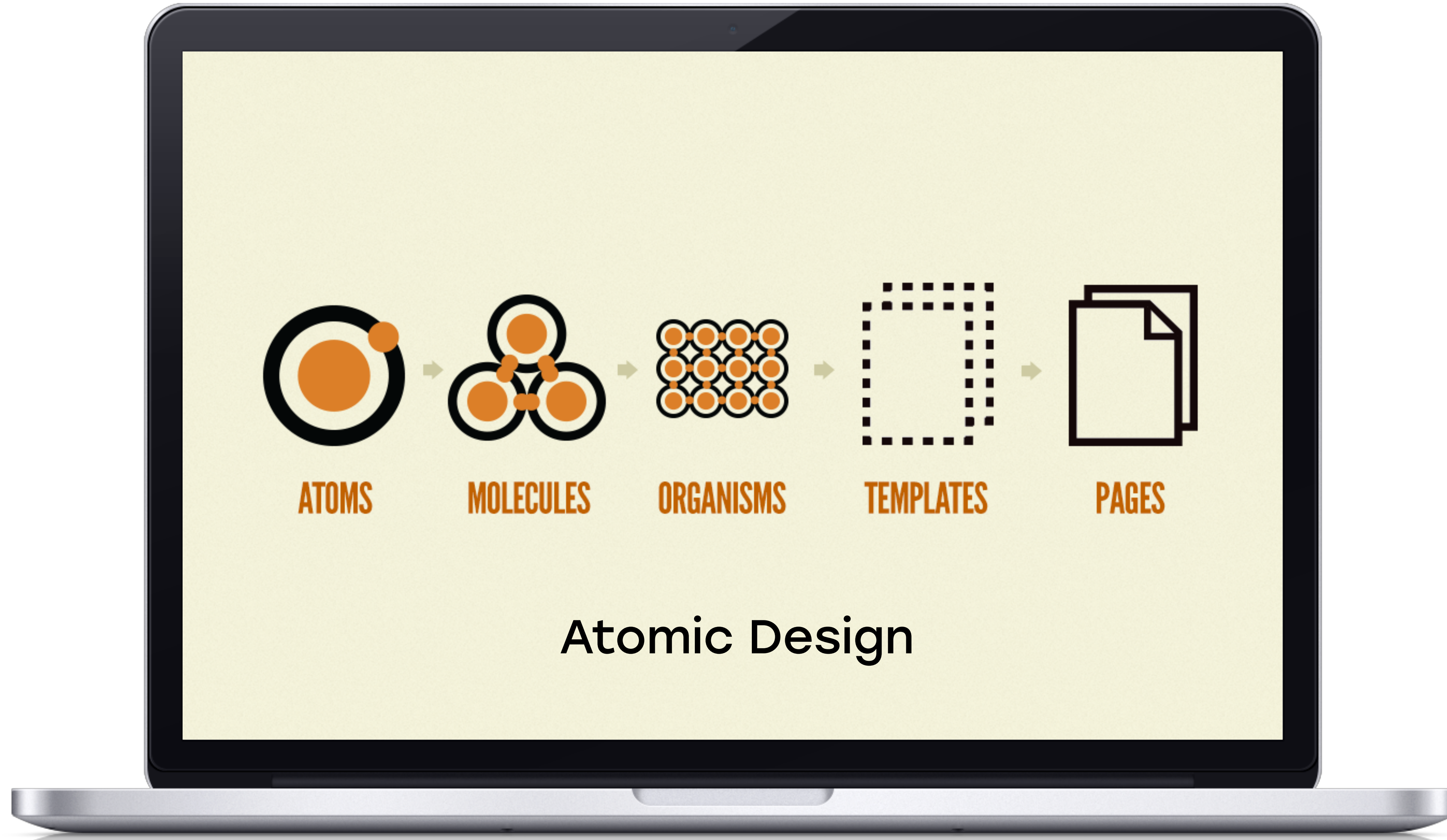


Cosmos (n.) - from Latinized form of Greek kosmos "order, good order, orderly arrangement" - A complex orderly self-inclusive system, an orderly harmonious systematic universe.

Cosmos is a multi-platform multi-brand Design System for Badoo mobile applications. It's currently under development and not all the UI components are covered by this style guide.

RELEASE NOTES

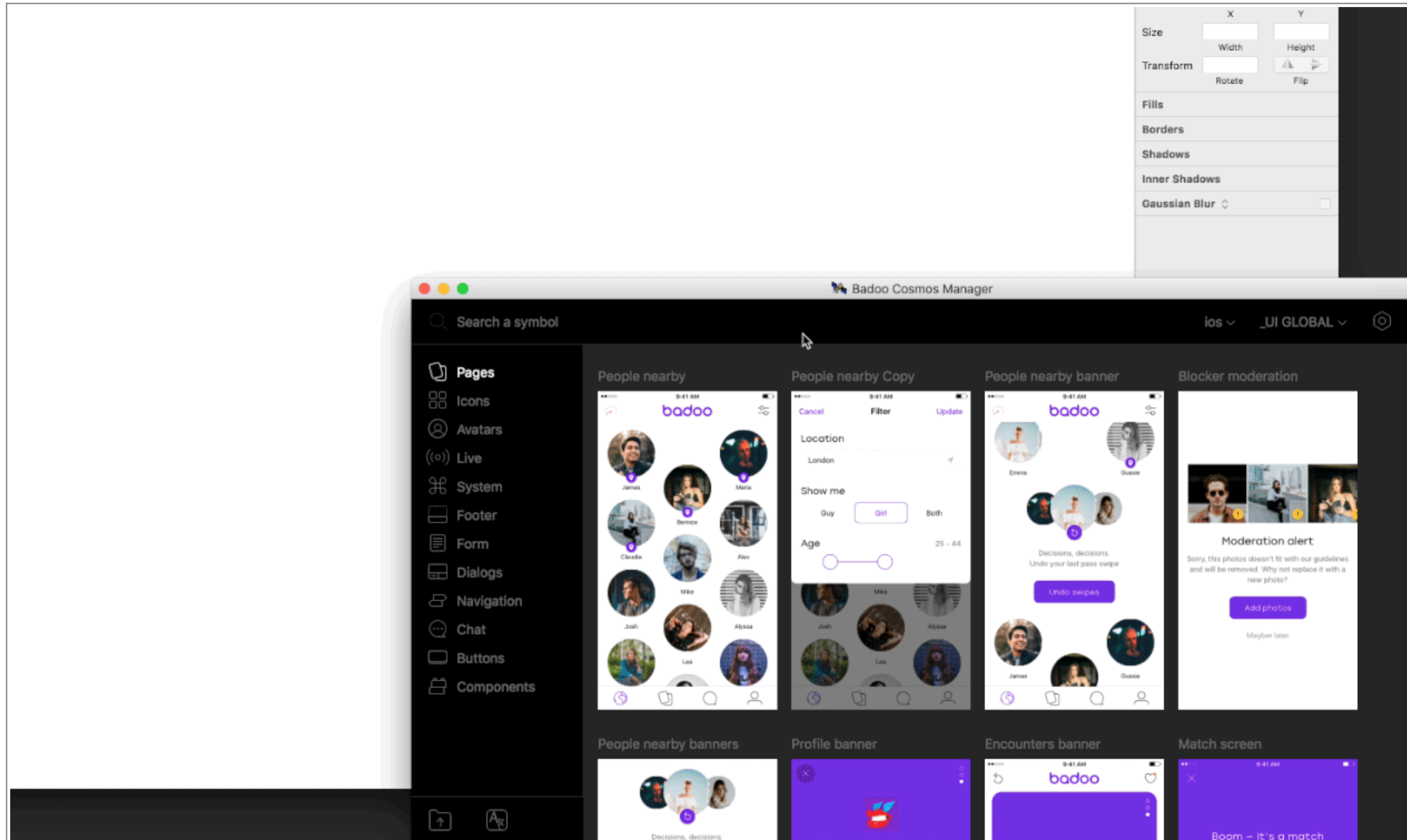
- 23 Nov 2018**
 - updated the **Icons** with support for "dynamic" icons from assets pipeline
released the dynamic assets pipeline, and updated the Icon component to support the dynamic (color + segmentation) icons distributed via pipeline
- 15 Nov 2018**
 - updated the **Tokens** with "Live" color
added the "live" color to the design tokens (and added it to the list of available colors)
- 5 Nov 2018**
 - updated the **Tokens** with "Premium" color
added the "premium" color to the design tokens (and deprecated the old "spp" value) + renamed all the options of SPP in Premium in the codebase
- 26 Oct 2018**
 - updated the **Gift** component
removed the sizes from the Gift component, now is always stretching (similar to Avatar, Sticker etc.)

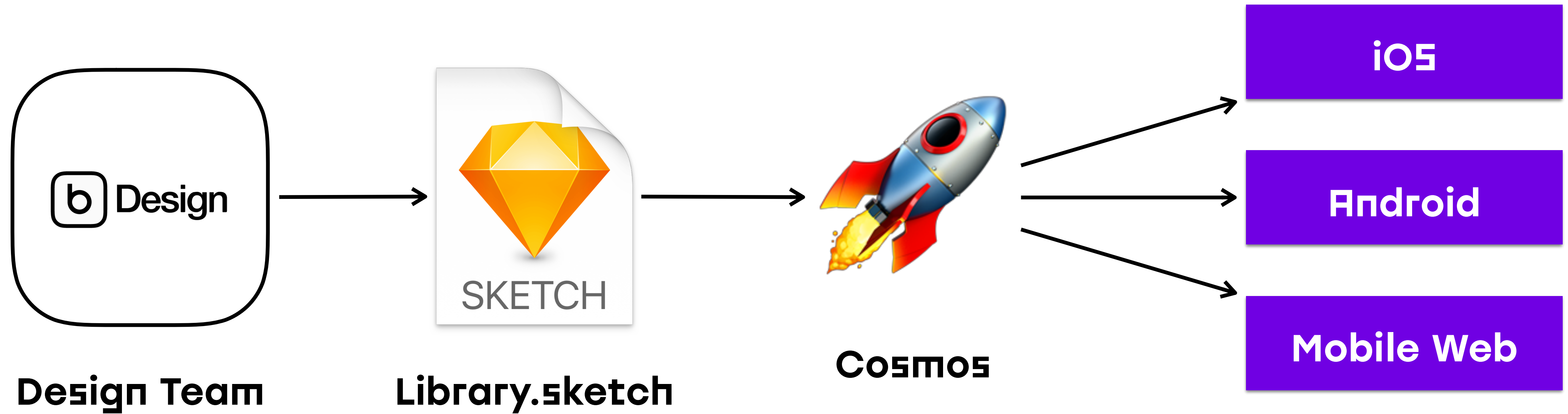




Library.sketch

– содержит все UI компоненты,
использующиеся для макетов экранов
приложений





Преимущества дизайн-системы

- Консистентность дизайна

Преимущества дизайн-системы

- Консистентность дизайна
- Повышается
переиспользование
компонентов

Задача:

**Быстро адаптировать
приложения под изменения в
брендинге**

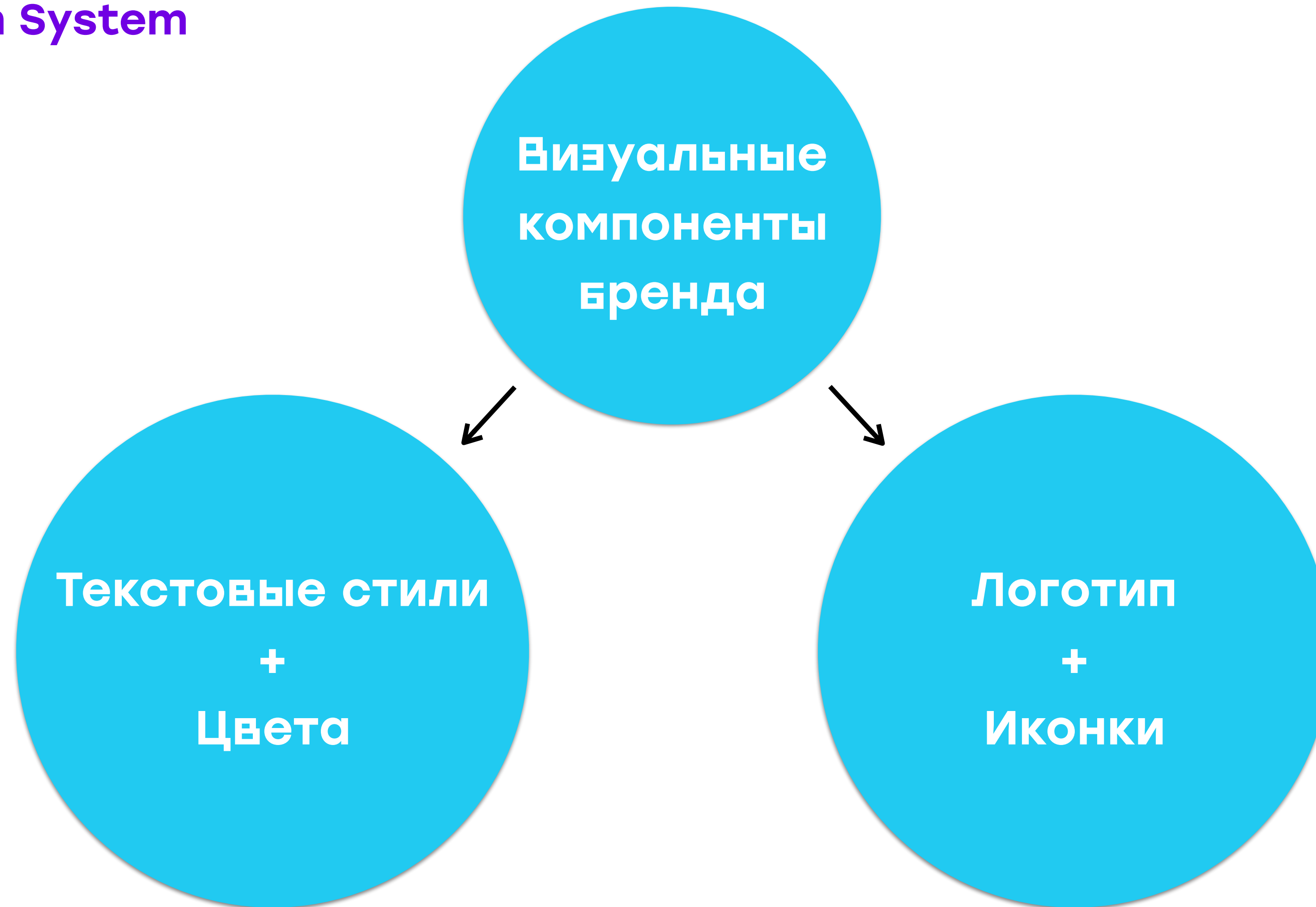
Визуальные составляющие бренда

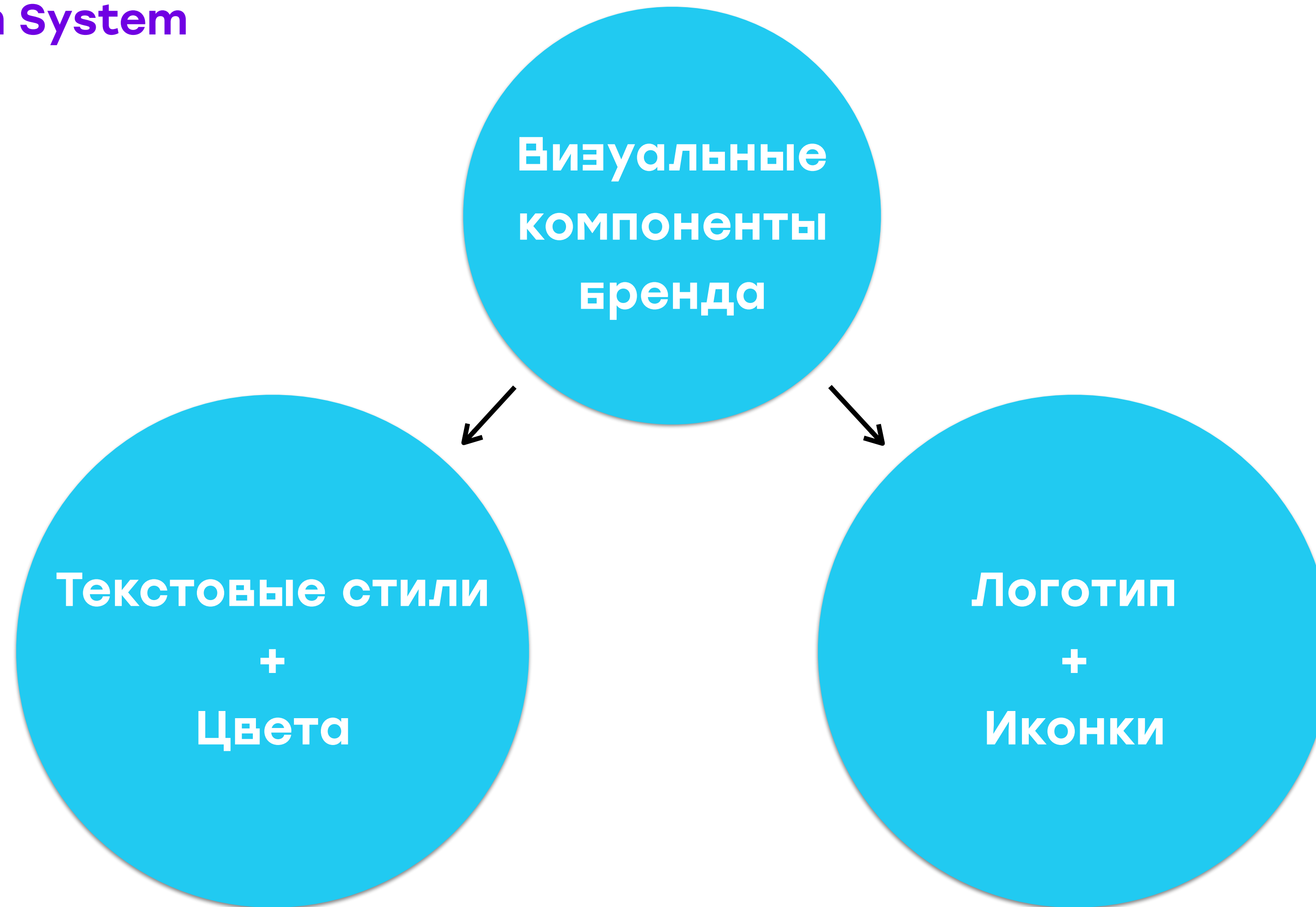
- Логотип
- Цветовая палитра
- Текстовые стили
- Иконки

Проблема:

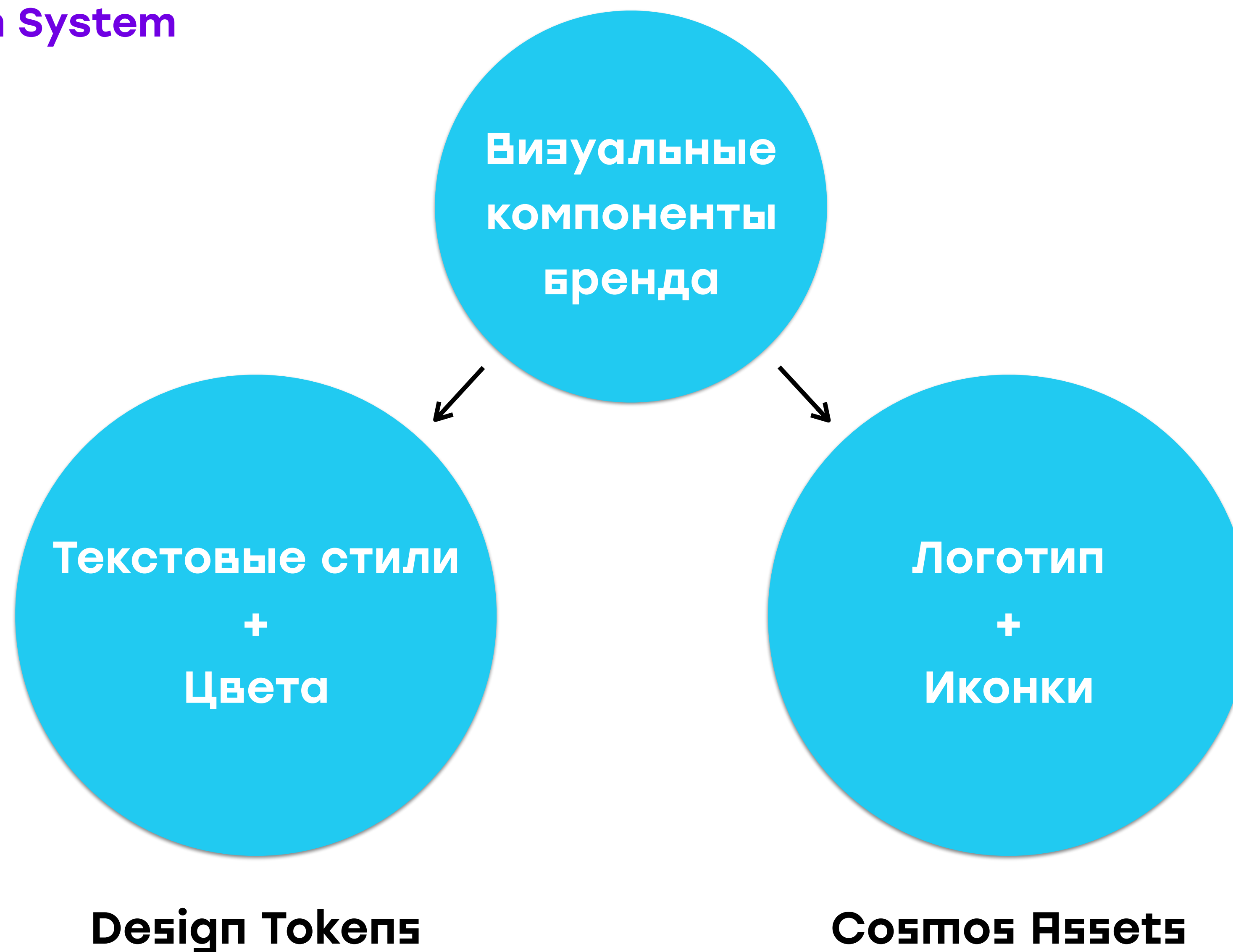
Чтобы применить изменения в брендинге, нужно обновить все «атомы» в нашей дизайн-системе, а это СЛИШКОМ ДОЛГО



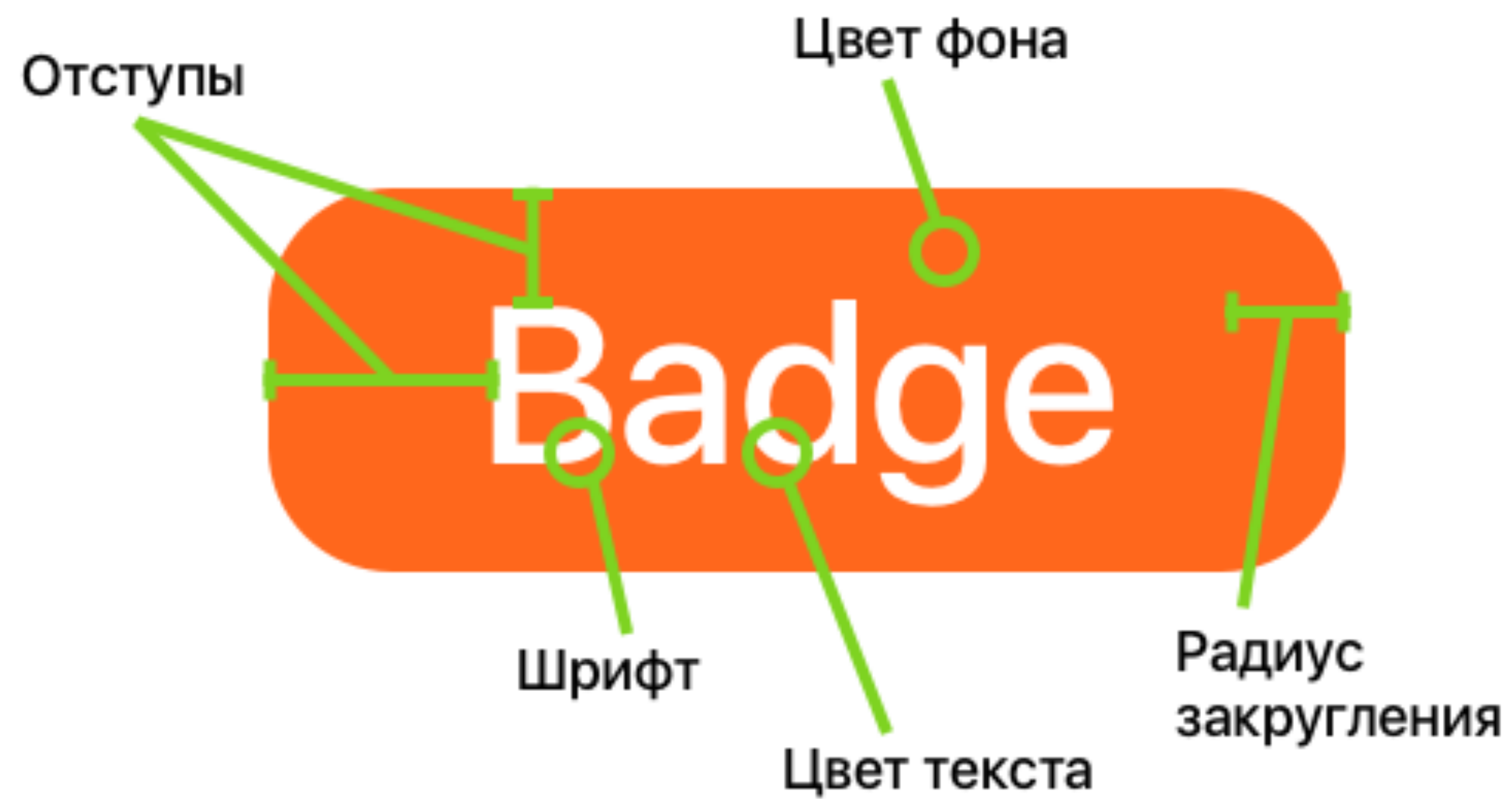


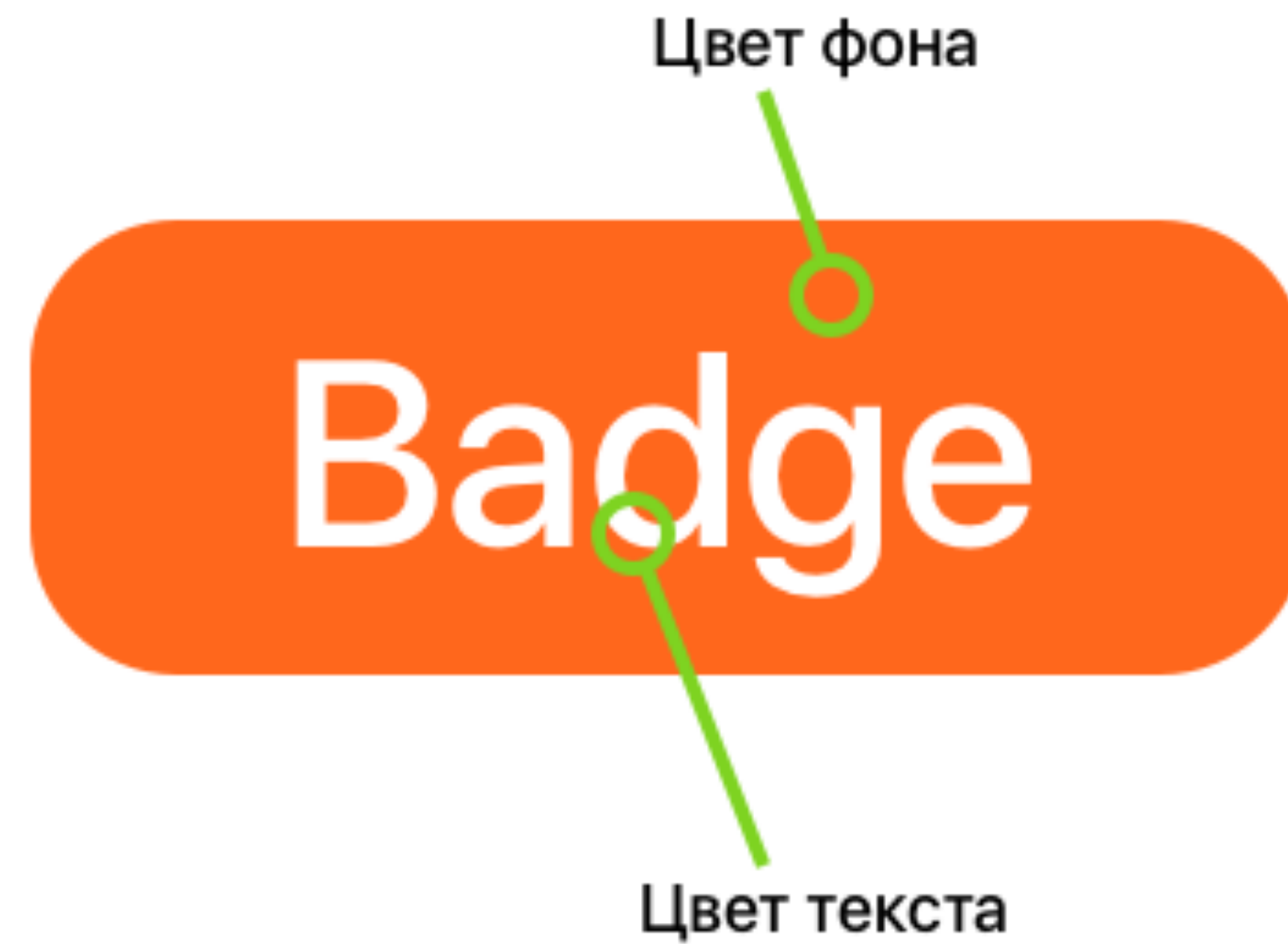


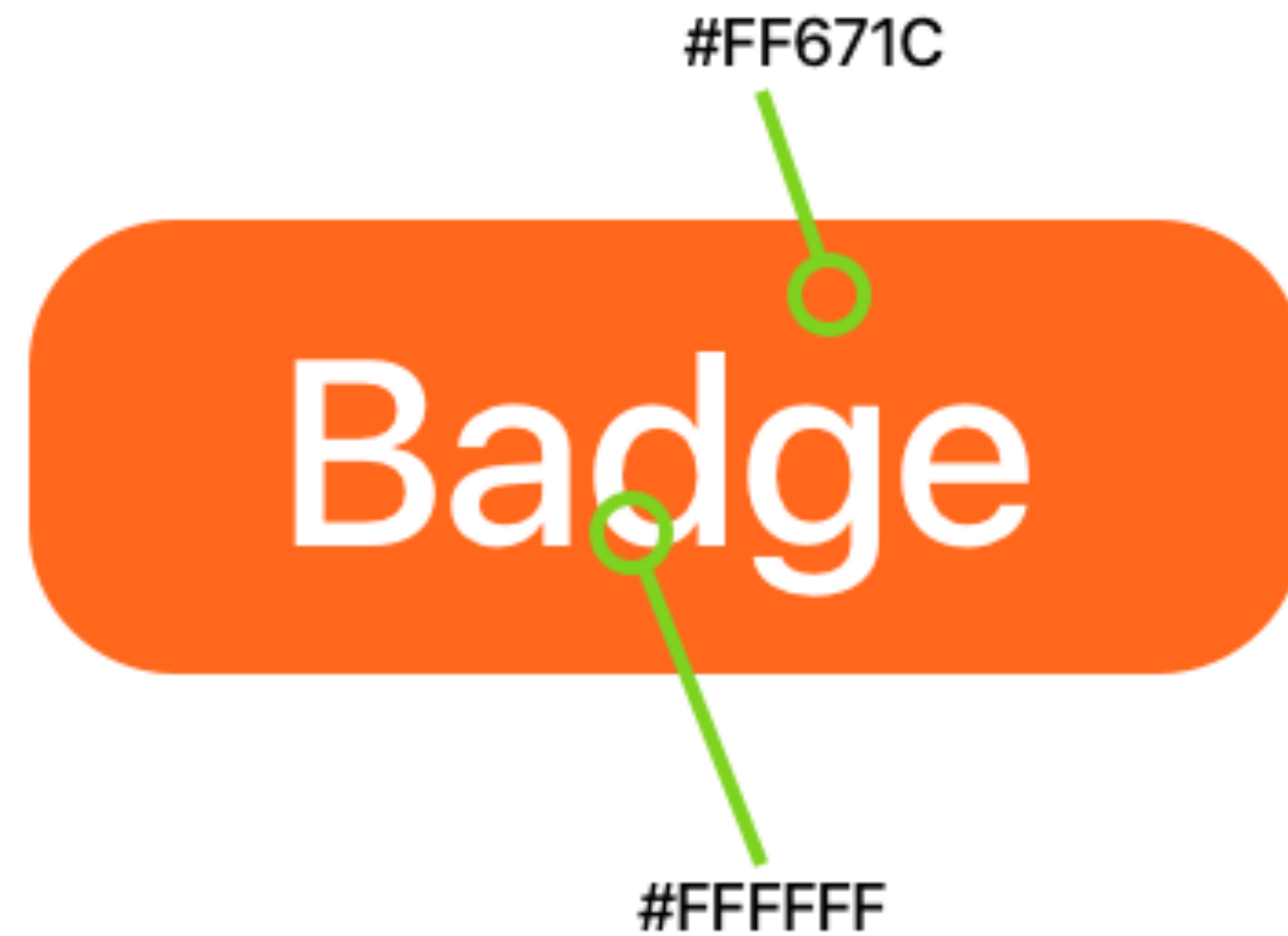
Design Tokens



Design Tokens

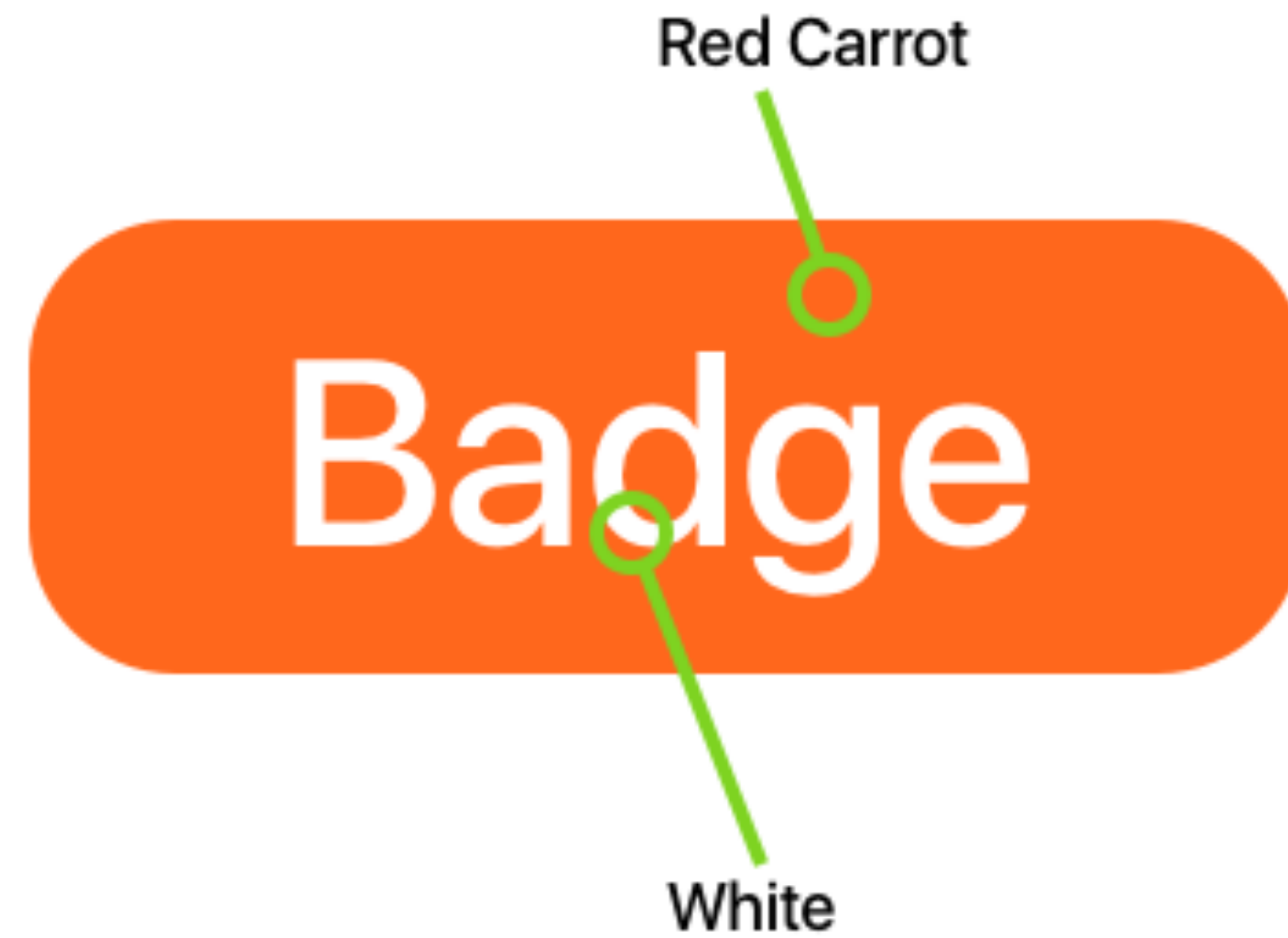






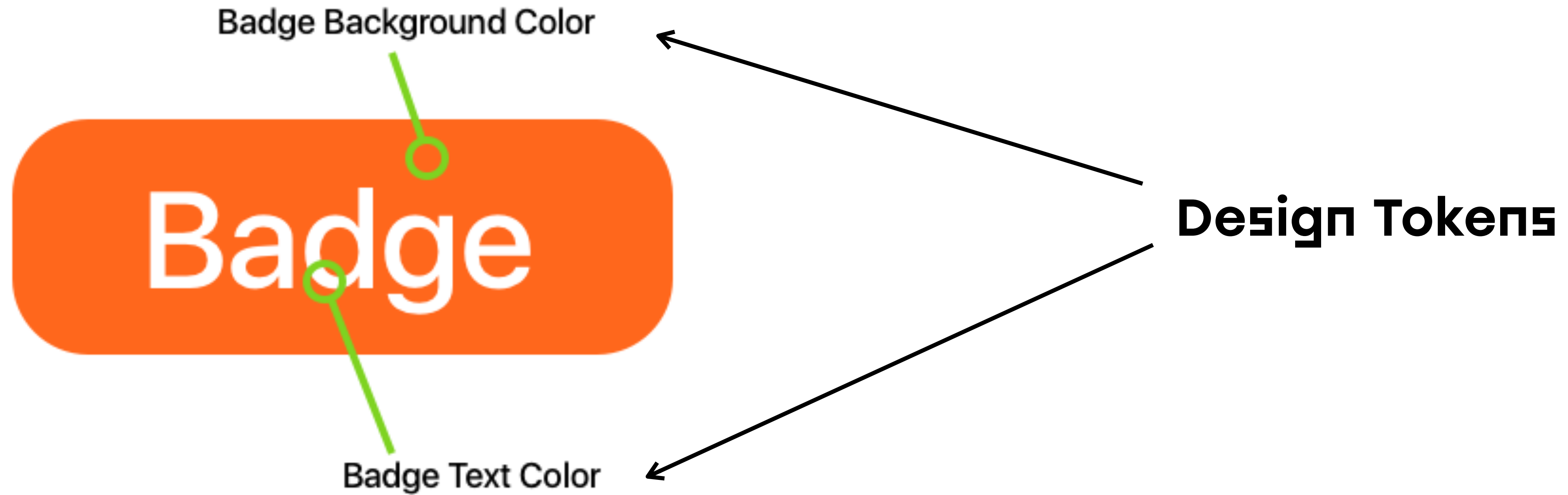
Проблема:

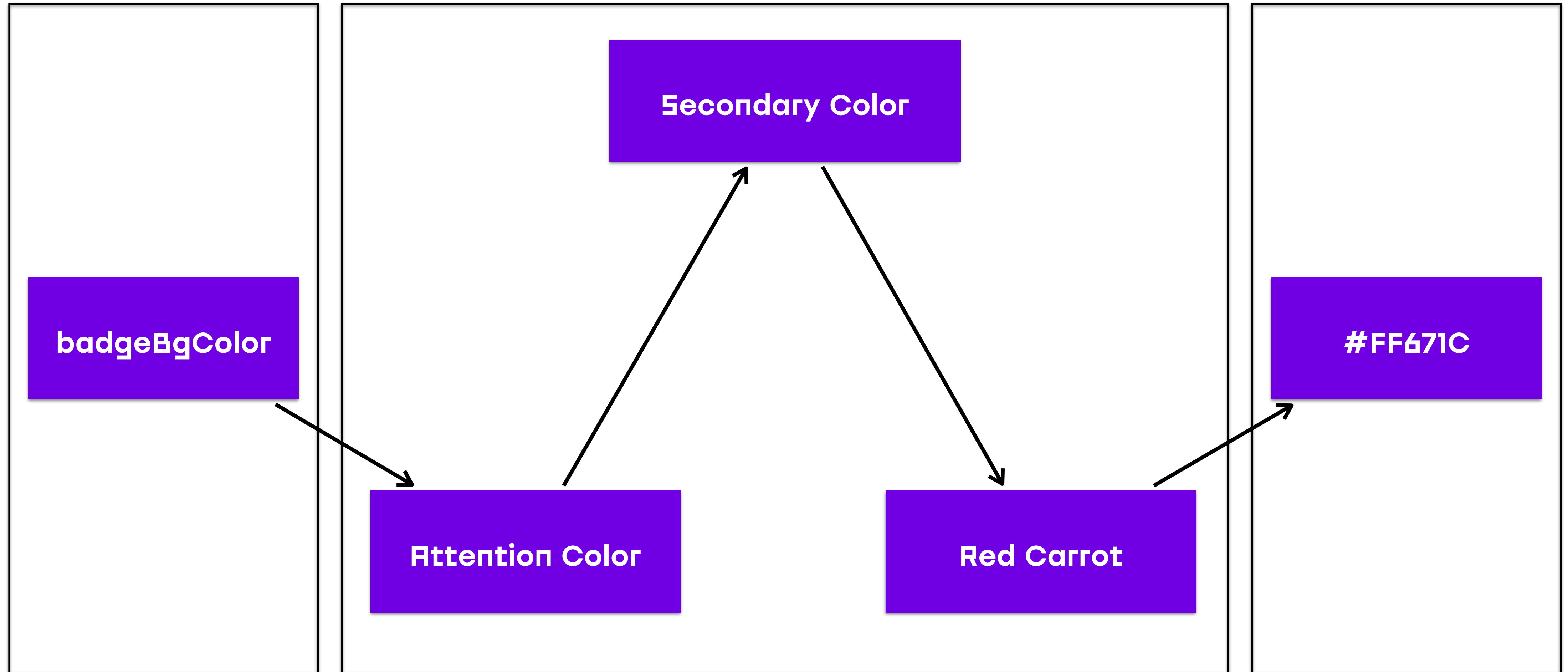
При любых изменениях в оттенках цветов в палитре бренда нам потребуются ручную заменить в проекте все вхождения старого оттенка



Проблема:

В случае сокращения или расширения палитры снова может потребоваться рефакторинг всех UI КОМПОНЕНТОВ








iOS

Cosmos

Library.sketch

CHAT

tokenChatBubblePrimaryTextColor*	#ffffff	▼
tokenChatBubblePrimaryBackgroundColor*	 #1db6cb	▼
tokenChatBubbleGrayTextColor*	 #000000	▼
tokenChatBubbleGrayBackgroundColor*	 #f6f6f6	▼
tokenChatToolbarHeight	48	▼
tokenChatToolbarItemGap	4	▼
tokenChatToolbarSendIconSize	32	▼
tokenChatSwitcherItemSize	40	▼
tokenChatSwitcherIconSize	24	▼
tokenChatMiniHeight	36	▼
tokenChatMiniSendIconSize	32	▼

Реализация ДИЗАЙН-ТОКЕНОВ

 / [resources](#) / [tokens](#) / [1.1.6](#)

- ↑ ..
- ▣ [android](#)
- ▣ [ios](#)
- ▣ [mw](#)
- ▣ [website-gui](#)



tokens-all-unified.plist

tokens-all-unified.plist > No Selection

Key	Type	Value
▶ tokenBrickSizeXlg	Dictionary	(2 items)
▶ tokenBrickBorderRadiusCircle	Dictionary	(2 items)
▶ tokenBrickBorderRadiusSquared	Dictionary	(2 items)
▶ tokenButtonBorderRadius	Dictionary	(2 items)
▶ tokenButtonBorderWidth	Dictionary	(2 items)
▶ tokenButtonHeight	Dictionary	(2 items)
▶ tokenButtonHeightSmall	Dictionary	(2 items)
▶ tokenChatBubblePrimaryTextColor	Dictionary	(2 items)
▼ tokenChatBubblePrimaryBackgroundC...	Dictionary	(2 items)
type	String	color
value	String	#1f63dc
▶ tokenChatBubbleGrayTextColor	Dictionary	(2 items)
▶ tokenChatBubbleGrayBackgroundColor	Dictionary	(2 items)
▶ tokenChatToolbarHeight	Dictionary	(2 items)
▶ tokenChatToolbarItemGap	Dictionary	(2 items)
▶ tokenChatToolbarSendIconSize	Dictionary	(2 items)
▶ tokenChatSwitcherItemSize	Dictionary	(2 items)
▶ tokenChatSwitcherIconSize	Dictionary	(2 items)
▶ tokenChatMiniHeight	Dictionary	(2 items)
▶ tokenChatMiniSendIconSize	Dictionary	(2 items)

```
extension DesignTokens {  
    public static let badoo = DesignTokens(  
        ...  
        chatBubbleGrayTextColor: UIColor(...),  
        chatBubblePrimaryBackgroundColor: UIColor(...),  
        chatBubblePrimaryTextColor: UIColor(...),  
        chatMiniHeight: 36,  
        chatMiniSendIconSize: 32,  
        chatSwitcherIconSize: 24,  
        chatSwitcherItemSize: 40,  
        chatToolbarHeight: 48,  
        chatToolbarItemGap: 4,  
        chatToolbarSendIconSize: 32  
        ...  
    )  
}
```

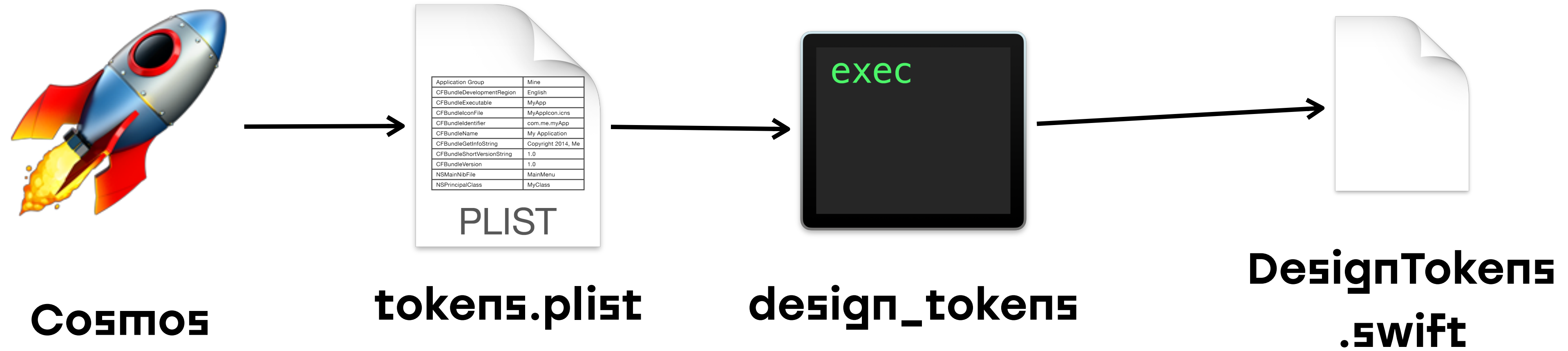
```
func createDefaultStyle() -> Style {  
    var style = Style()  
    style.backgroundColor = Cosmos.designTokens.badgeBackgroundColor  
    style.textColor = Cosmos.designTokens.badgeTextColor  
    ...  
}
```



```
Badoo-secondary — -bash — 100x30
Igor-Savelevs-MacBook-Pro:Badoo-secondary igorsavelev$ ./scripts/cosmos_resources/tokens/design_tokens -m ./scripts/cosmos_resources/manifest.json -b badoo -t badoo_ui_kit_tokens -v 1.1.7

Brands: Badoo
Targets: BadooUIKit Design Tokens
New version: 1.1.7

==== Fetching tokens ====
Fetching tokens for Badoo
  Downloading tokens file from: http://cosmos/resources/tokens/1.1.7/ios/badoo/tokens-all-unified.plist
  Parsing tokens
  ⚠ Token with name tokenFontFamilyCustom doesn't have type. Skipping.
==== Verifying tokens ====
Total amount of tokens found: 180
Using Badoo as a reference
Verification successful
🎯 Running steps for target BadooUIKit Design Tokens
==== Generating swift tokens ====
Building contexts
Rendering
Rendering DesignTokensProtocol.swift
Rendering DesignTokens.swift
Rendering DesignTokens+%BRAND%.swift for Badoo
✅ Done
Igor-Savelevs-MacBook-Pro:Badoo-secondary igorsavelev$
```



Преимущества

- Code-completion

Преимущества

- Code-completion
- Компилятор выдаст ошибку, если в очередной версии токенов пропадет какой-то из них

Преимущества

- Code-completion
- Компилятор выдаст ошибку, если в очередной версии токенов пропадет какой-то из них
- Возможность верифицировать значения перед этапом генерации Swift файлов



When she likes you back...

You liked Clarissa, if they like you back - it's a match!

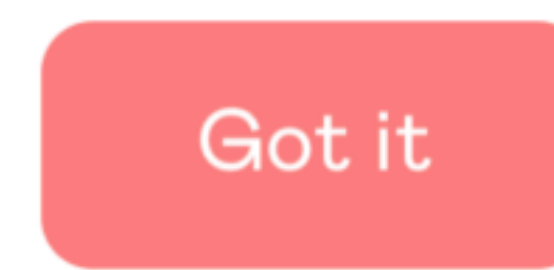


Текущий бренд



When she likes you back...

You liked Clarissa, if they like you back - it's a match!



Эксперимент



When she likes you back...

You liked Clarissa, if they like you back - it's a match!

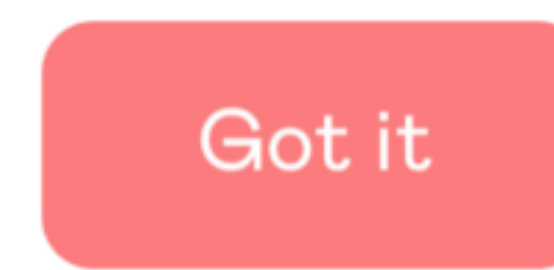


Текущий бренд



When she likes you back...

You liked Clarissa, if they like you back - it's a match!



Эксперимент

Cosmos Assets

ICONS

DESIGN

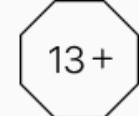












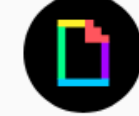





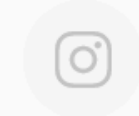











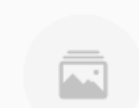




DEVELOPMENT

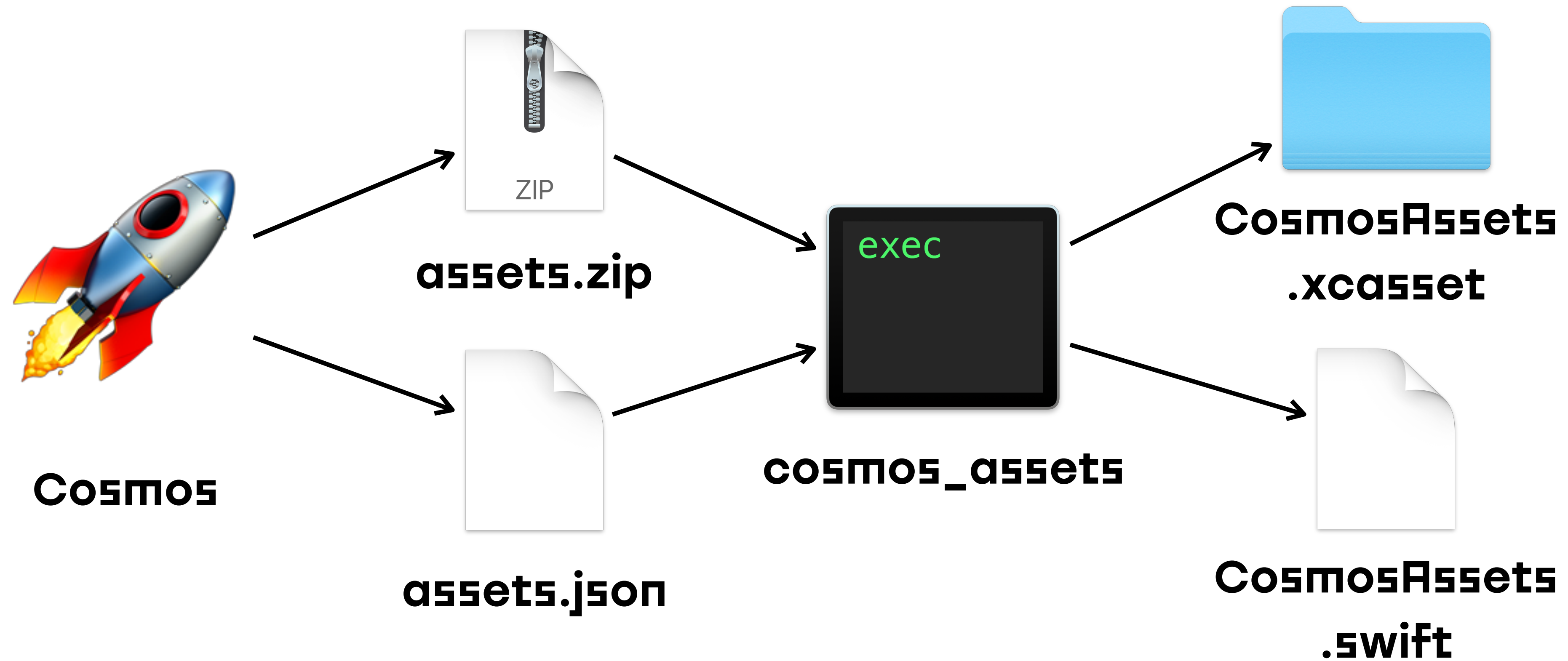
PLAYGROUND

DYNAMIC ASSETS (COMING FROM THE PIPELINE)



Filter:

 account-abuse-13only	 account-abuse-18only	 account-abuse-blocked	 account-abuse-warning	 adaptor-alert	 adaptor-camera
 adaptor-card	 adaptor-copy	 adaptor-copy-inactive	 adaptor-email	 adaptor-email-inactive	 adaptor-facebook
 adaptor-facebook-inactive	 adaptor-giphy	 adaptor-gmail	 adaptor-gmail-inactive	 adaptor-google	 adaptor-google-inactive
 adaptor-instagram	 adaptor-instagram-inactive	 adaptor-linkedin	 adaptor-linkedin-inactive	 adaptor-messenger	 adaptor-messenger-inactive
 adaptor-odnoklassniki	 adaptor-odnoklassniki-inactive	 adaptor-other	 adaptor-other-inactive	 adaptor-phone	 adaptor-phone-inactive
 adaptor-image	 adaptor-image-inactive	 adaptor-diamond	 adaptor-diamond-inactive	 adaptor-download	 adaptor-check



```
// CosmosAssets.swift

final class CosmosAssets {
    ...
    let accountAbuse13only = CosmosAsset(assetName: "account-abuse-13only", experiments: [])
    let accountAbuse18only = CosmosAsset(assetName: "account-abuse-18only", experiments: [])
    let accountAbuseBlocked = CosmosAsset(assetName: "account-abuse-blocked", experiments: [])
    let accountAbuseWarning = CosmosAsset(assetName: "account-abuse-warning", experiments: [])
    let adaptorAlert = CosmosAsset(assetName: "adaptor-alert", experiments: [])
    ...
}

// Usage

let icon = Cosmos.imageFor(cosmosAsset: .adaptorAlert)
```

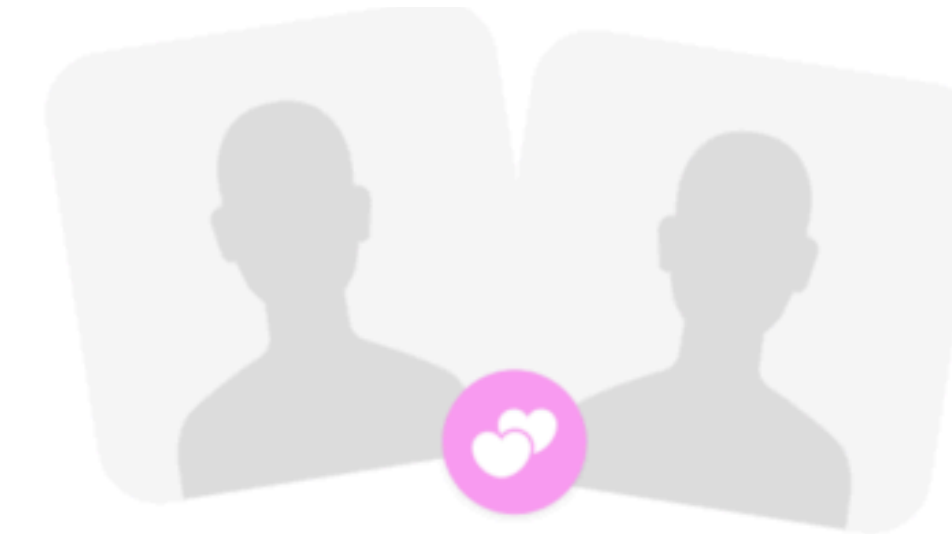


When she likes you back...

You liked Clarissa, if they like you back - it's a match!

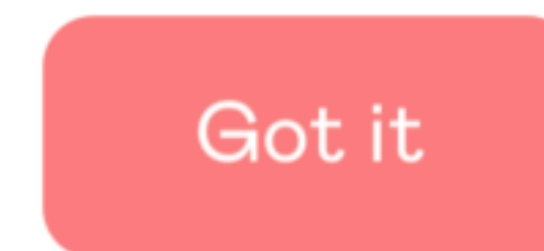


Текущий бренд



When she likes you back...

You liked Clarissa, if they like you back - it's a match!

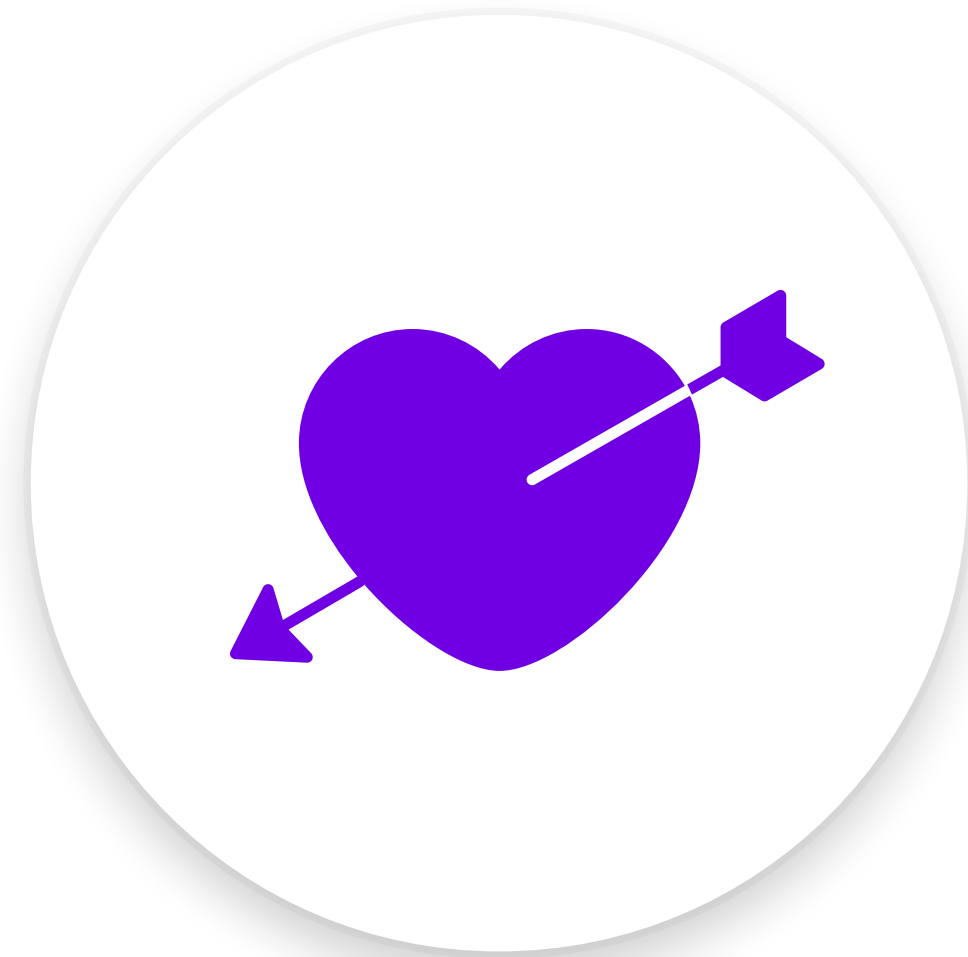


Эксперимент

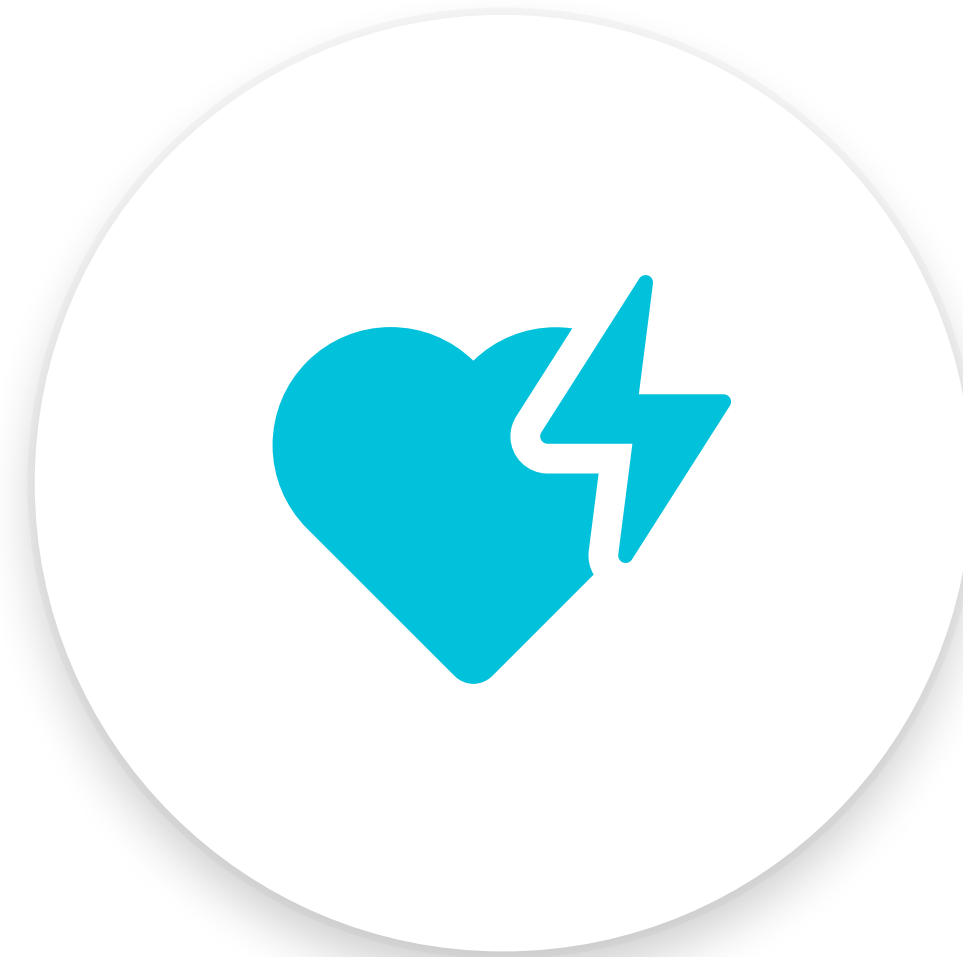
Задача со звездочкой:

**Реализовать сплит-тестирование
отдельной иконки при помощи
Cosmos Assets**

Crush icon test



control



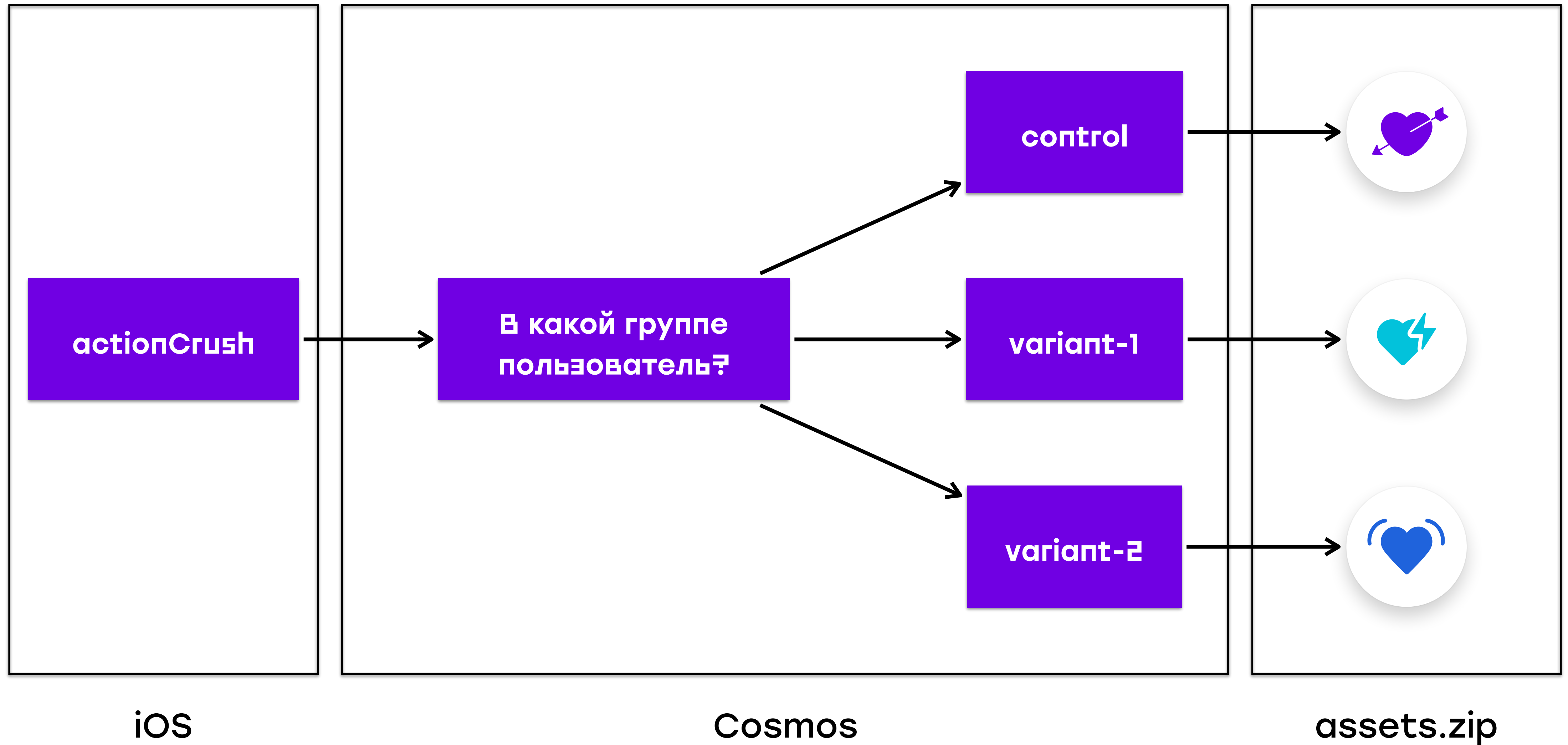
variant-1



variant-2

```
// assets.json

{
  "floating-action-crush": {
    "base-name": "floating-action-crush.pdf",
    "split-test": {
      "experiment-id": "crush-icon-test",
      "groups": {
        "control": "crush-icon-test/control.pdf",
        "variant-1": "crush-icon-test/variant-1.pdf",
        "variant-2": "crush-icon-test/variant-3.pdf"
      }
    }
  }
}
```



Преимущества

- Быстрая адаптация приложения под новый брендинг

Преимущества

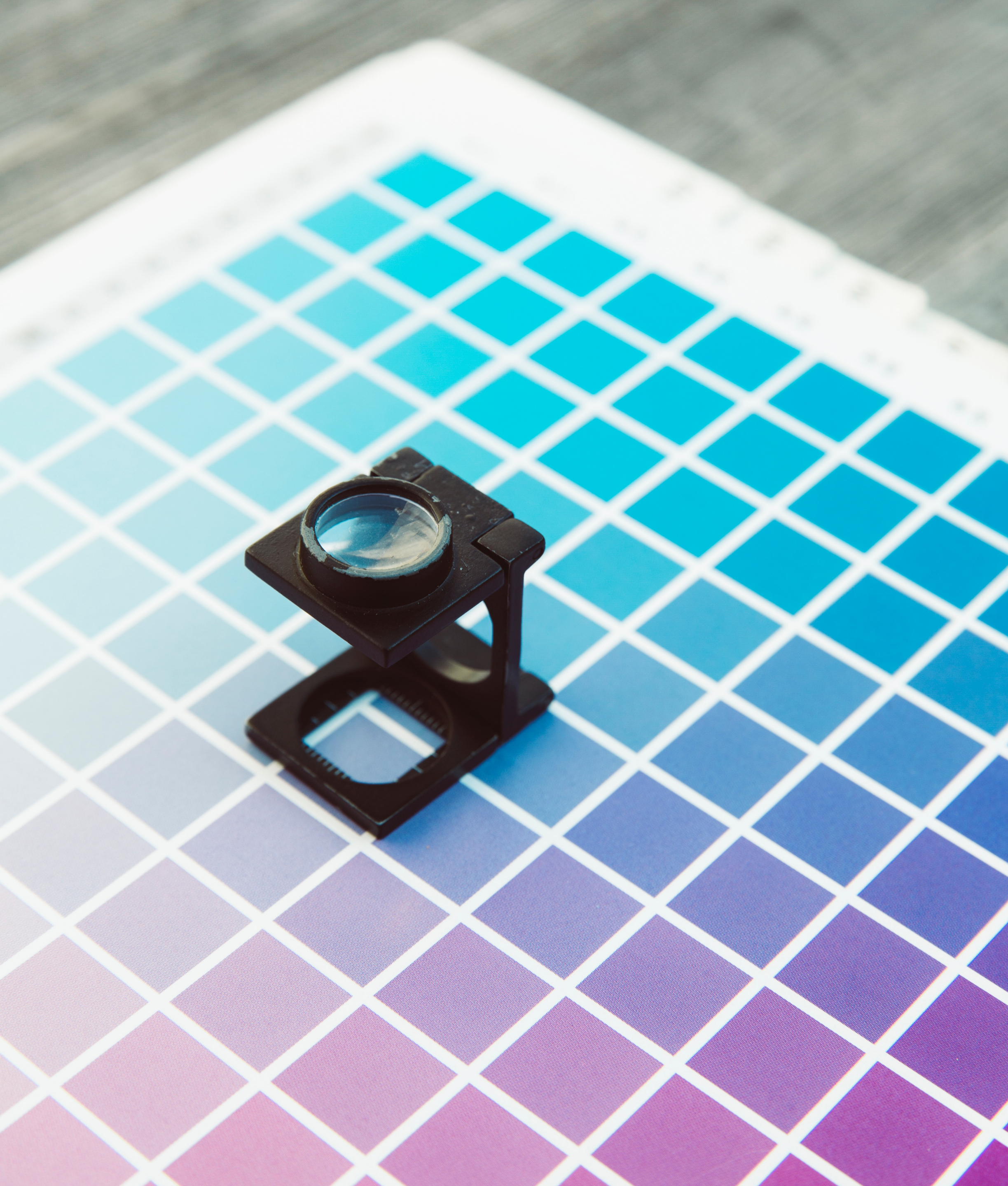
- Быстрая адаптация приложения под новый брендинг
- Прозрачное тестирование отдельных ассетов

Заключение

Заключение

Модуляризация UI
позволяет нам
создавать новые
приложения гораздо
быстрее

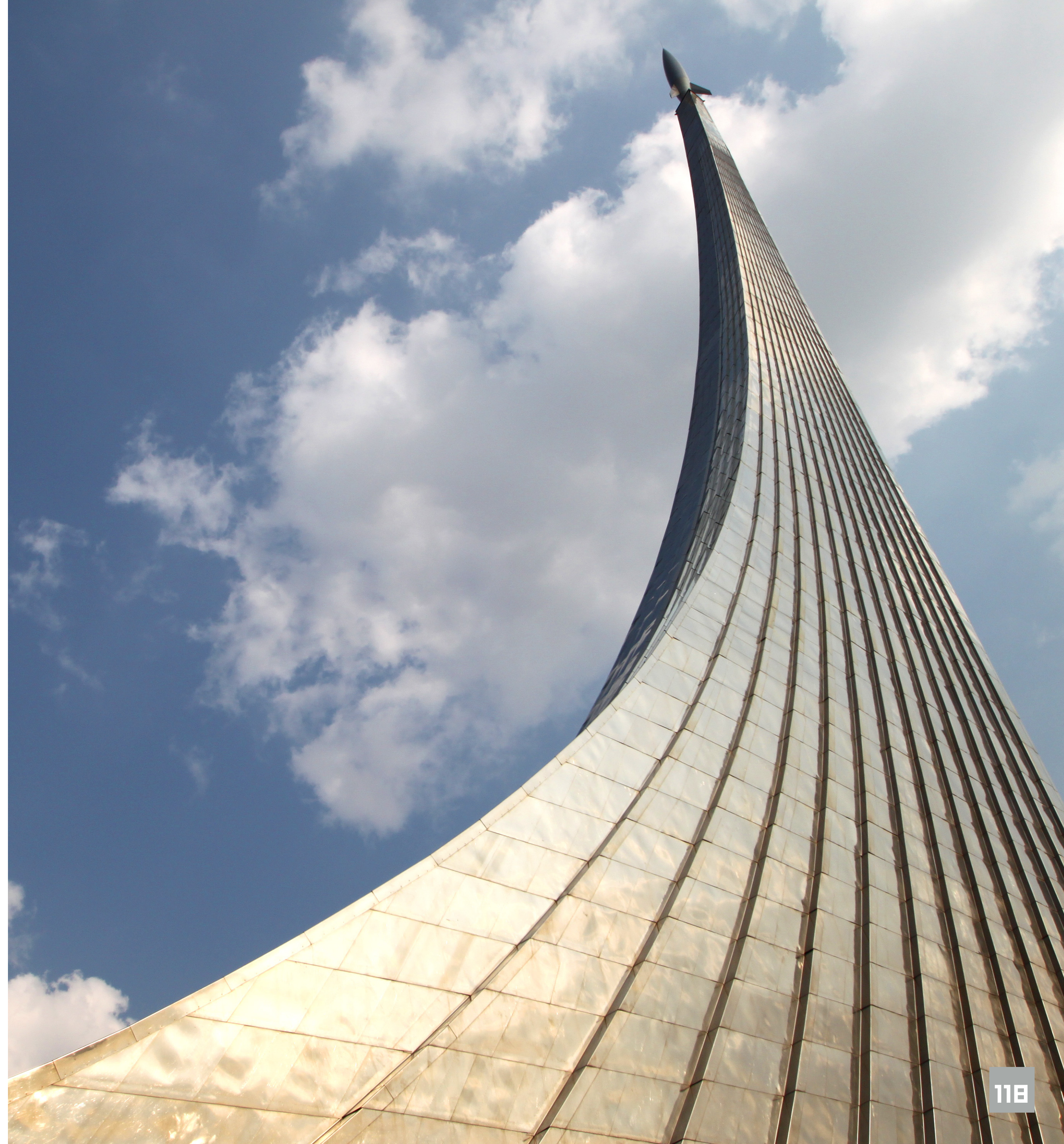




Visual Regression Testing
помогает эффективнее
использовать время QA
инженеров

Заключение

С дизайн-системой
Cosmos нам легче
адаптироваться к новым
трендам



Статьи

Implementing UI in iOS: Better, faster, and it scales!

<https://badootech.badoo.com/implementing-ui-in-ios-better-faster-and-it-scales-9788659e482>



Alexis Santos

iOS developer

Компонентная архитектура UI в iOS-приложении

<https://habr.com/company/badoo/blog/421559/>



Валерий Чевтаев

iOS developer

From zero to Cosmos: Parts 1-3

<https://badootech.badoo.com/from-zero-to-cosmos-part-1-2d080fe35bf2>



Cristiano Rastelli

Mobile Web developer

СПАСИБО

Игорь Савельев
igor.savelev@corp.badoo.com



Q&A



Презентация