

# Как интегрировать legacy сервисы в мобильное приложение дешево и быстро

Кейс Альфа-Банка

Без регистрации и смс...

# Артём Зяблицев

Team Lead в Альфа-Банке

Проектирую архитектуру  
Отвечаю за интеграции  
В прошлом - системный  
аналитик



# О чем расскажу

1. Наш контекст
2. Почему нам не подошла интеграция по API
3. Наше решение
4. Лайфхаки для внедрения



Контекст

# Вводные



## Много приложений и сервисов

Непонятно, в какой  
сервис и по какому  
вопросу идти



## Много разных бекэндов

Сложно  
поддерживать,  
интегрировать



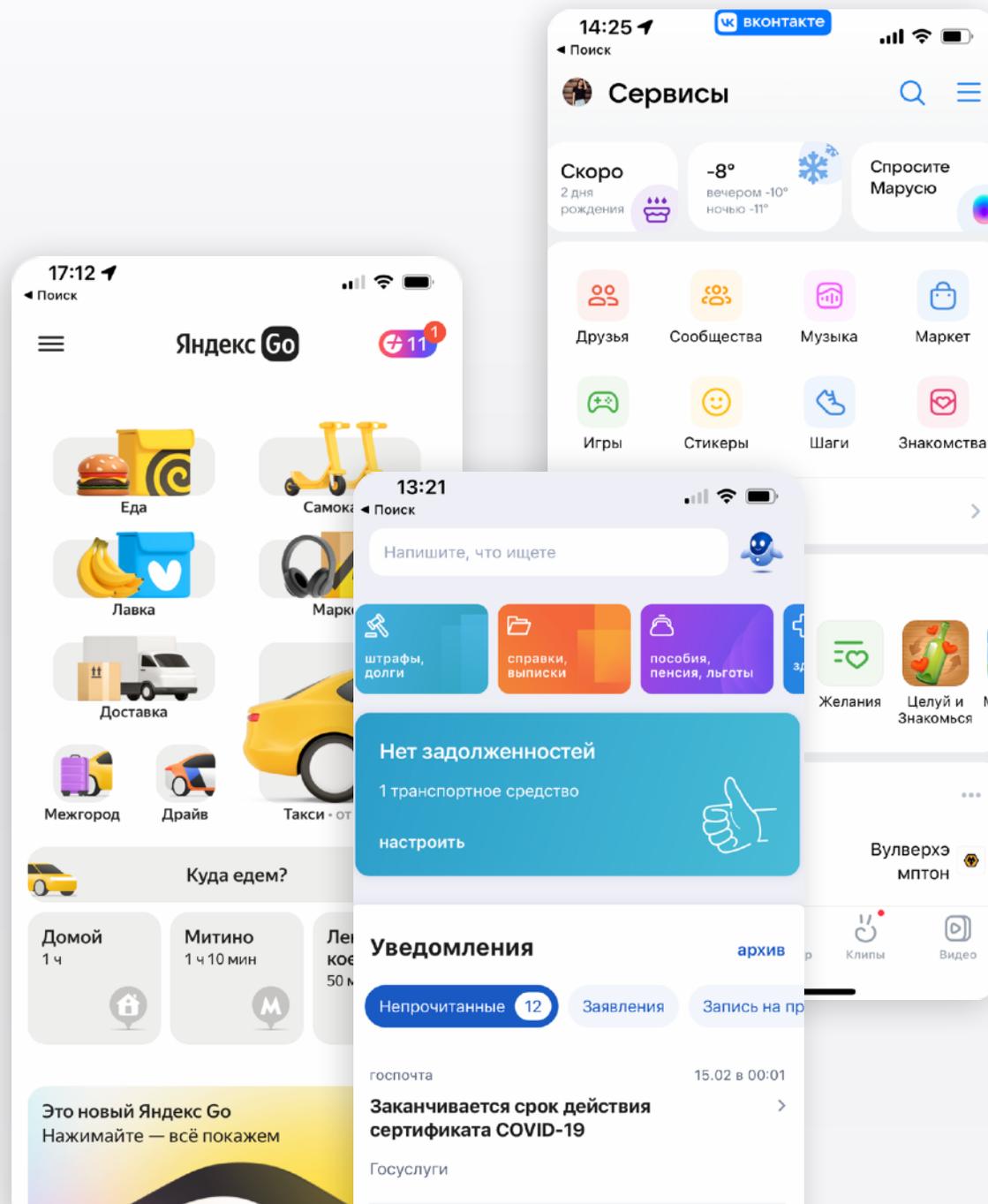
## Небольшой бюджет

Не можем  
нанять толпу  
разработчиков

# Контекст

## Бенчмарки

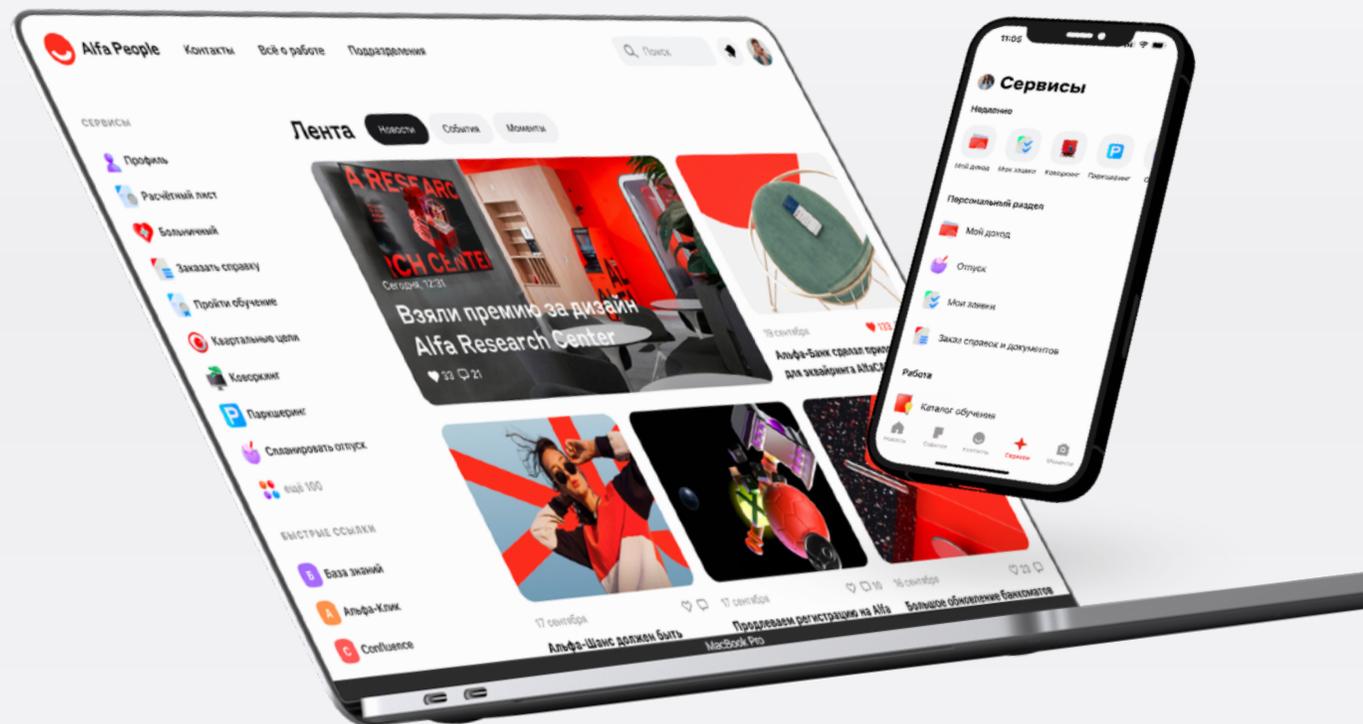
Посмотрели на другие клиентские приложения и захотели сделать также круто для сотрудников



Наша цель

# Продукт Alfa People

Единая точка контакта  
сотрудника со всеми  
HR-сервисами

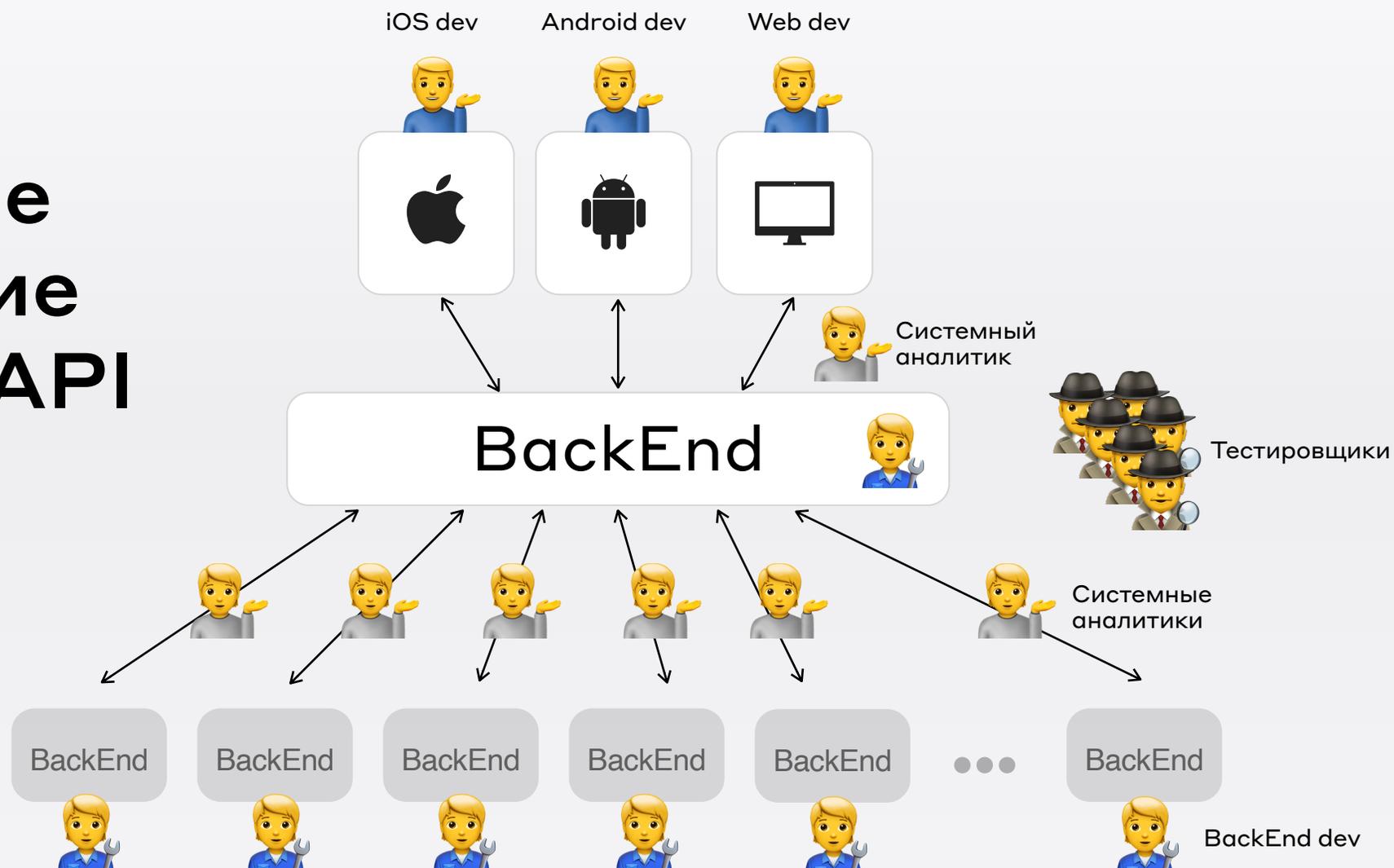


Чтобы сделать  
такой продукт,  
нужно спроектировать  
**подходящую**  
**архитектуру**

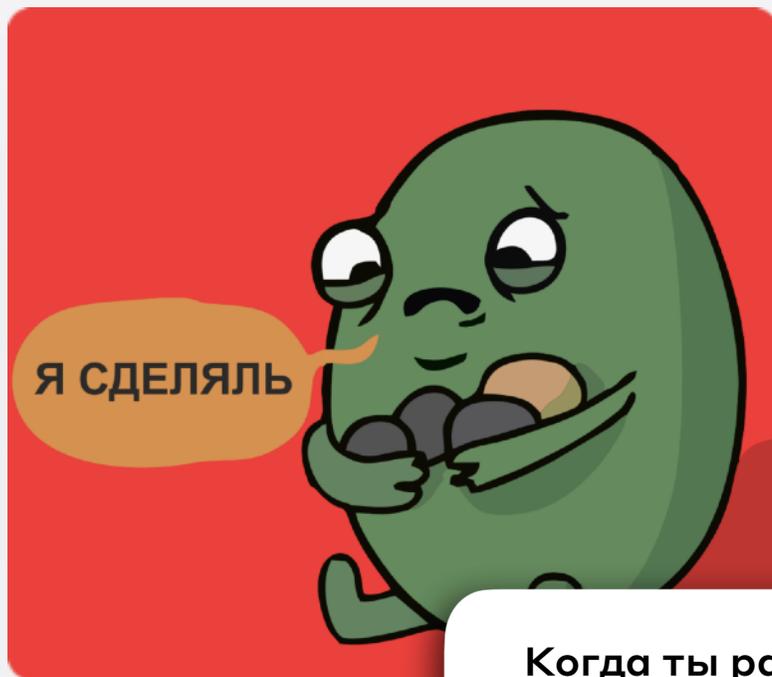


Классическая  
схема интеграций

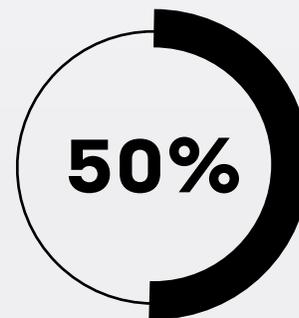
# Встраивание в приложение с помощью API



# Половина пользователей Alfa People на Android не обновляет приложение



Когда ты разработал фичу, а пользователи не обновляются



пользователей Android на старых версиях приложения Alfa People

**>10**

Необходимо поддерживать много версий приложения

# Почему это нам не подходит?



**Дорого**

Нужна большая  
DevTeam



**Больно**

Нужно полностью  
пересобрать команду



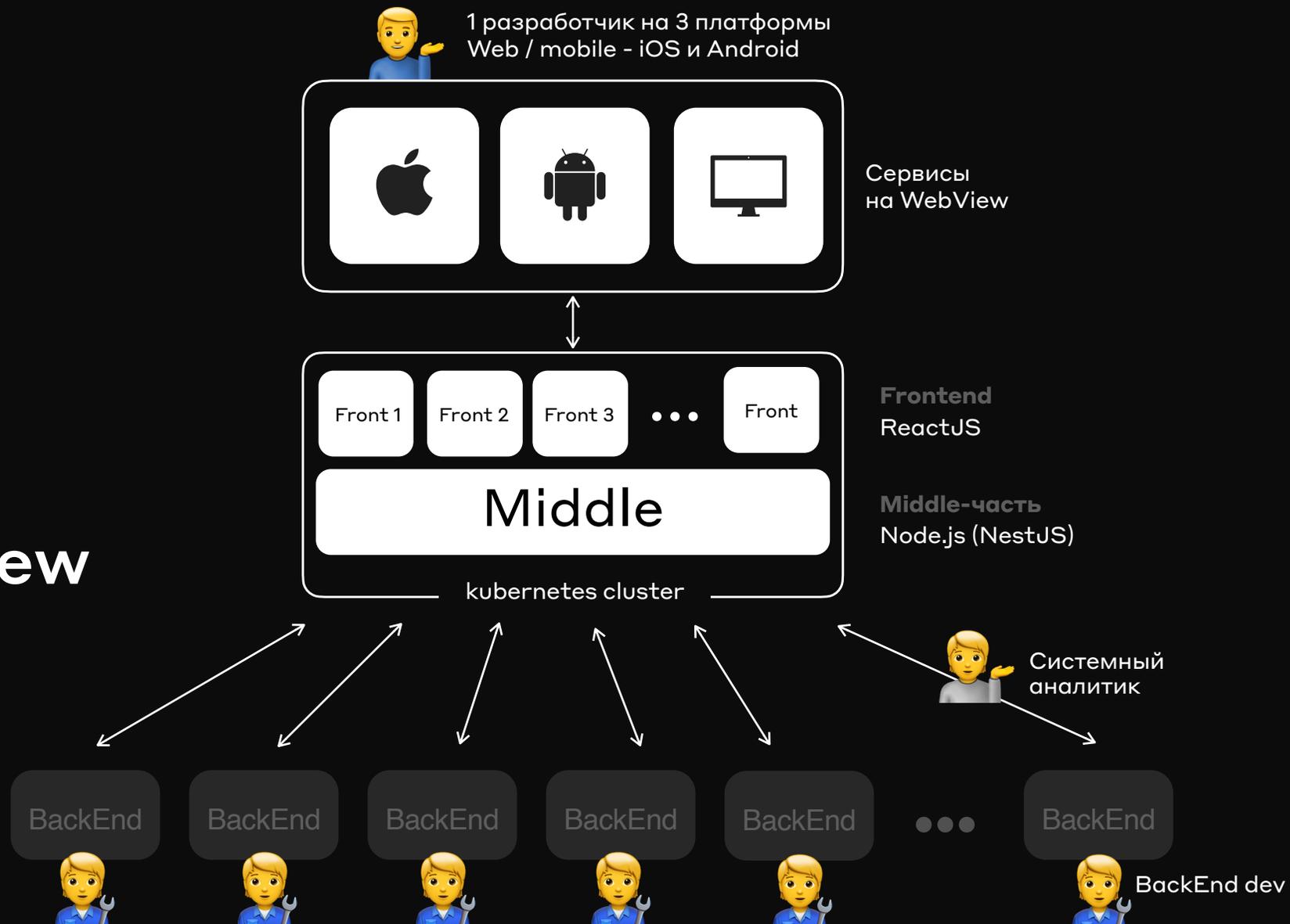
**Неудобно**

Нужно постоянно  
обновлять приложение

## **Наше решение**

**Встраивание сервисов в мобильное приложение с помощью WebView и микрофронтонтов**

# Встраивание в приложение с помощью сервисов **WebView**



# Плюсы такого решения



## Дешевле

Не нужно много Android  
и iOS разработчиков

На рынке больше  
JS-разработчиков



## Старое Legacy в новом дизайне

Большие возможности  
для объединения старых  
беков в красивое  
приложение



## Проще доставлять новые сервисы

Сервисы появляются  
у клиента без обновления  
приложения

# Особенности работы с WebView

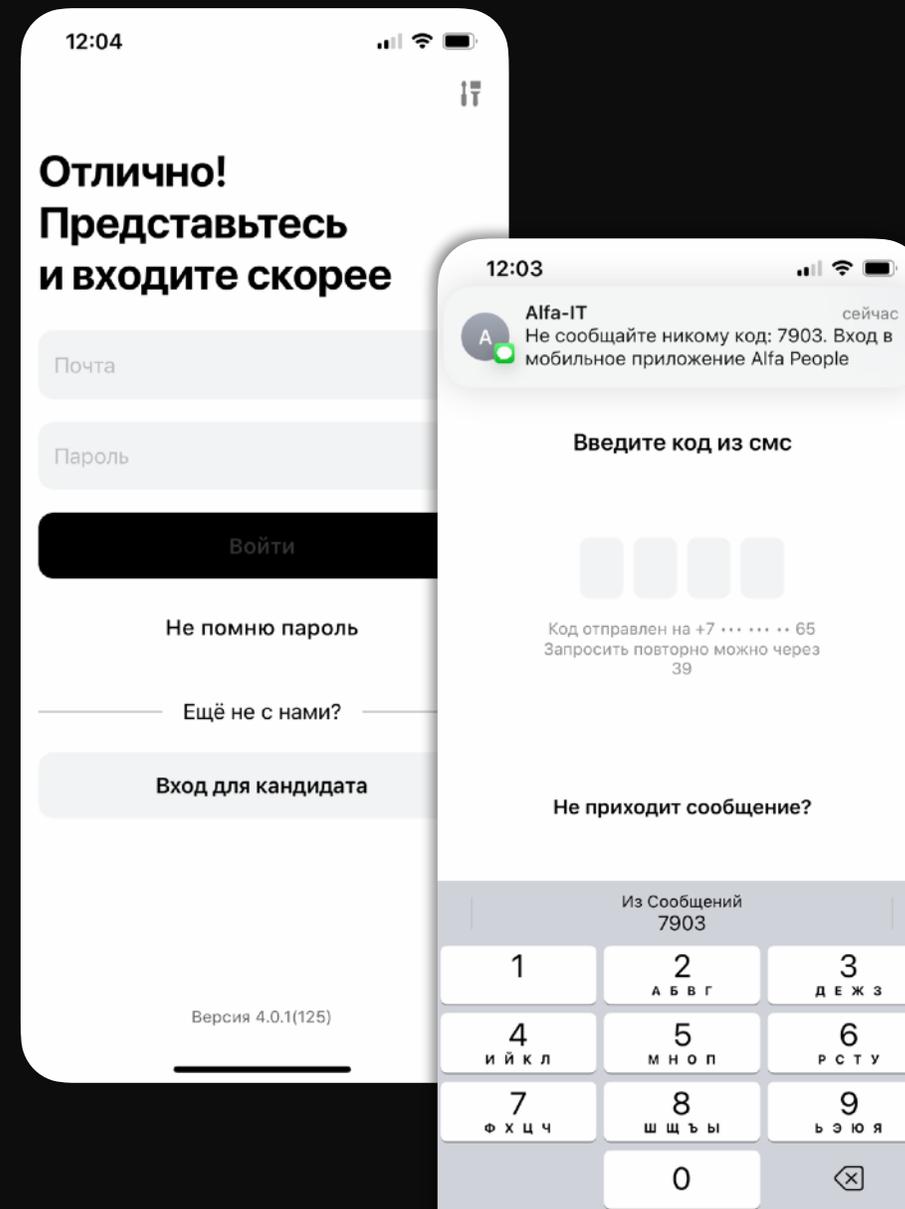


1. Как настроили аутентификацию и авторизацию
2. Какие фичи лучше разрабатывать на нативе
3. Как помогает единая дизайн-система
4. Как можно доставлять новые фичи без обновления

# Настроили аутентификацию и авторизацию

Вход по почте и паролю от рабочего компьютера

Используем двухфакторную аутентификацию с помощью – OTP (one time password)



# Как мы снижаем риски несанкционированного доступа



**Соблюдаем безопасный порядок ввода данных**

Имя учетная записи  
-> OTP-код  
-> пароль от учётки



**Блокировка при неправильных попытках ввода пароля**

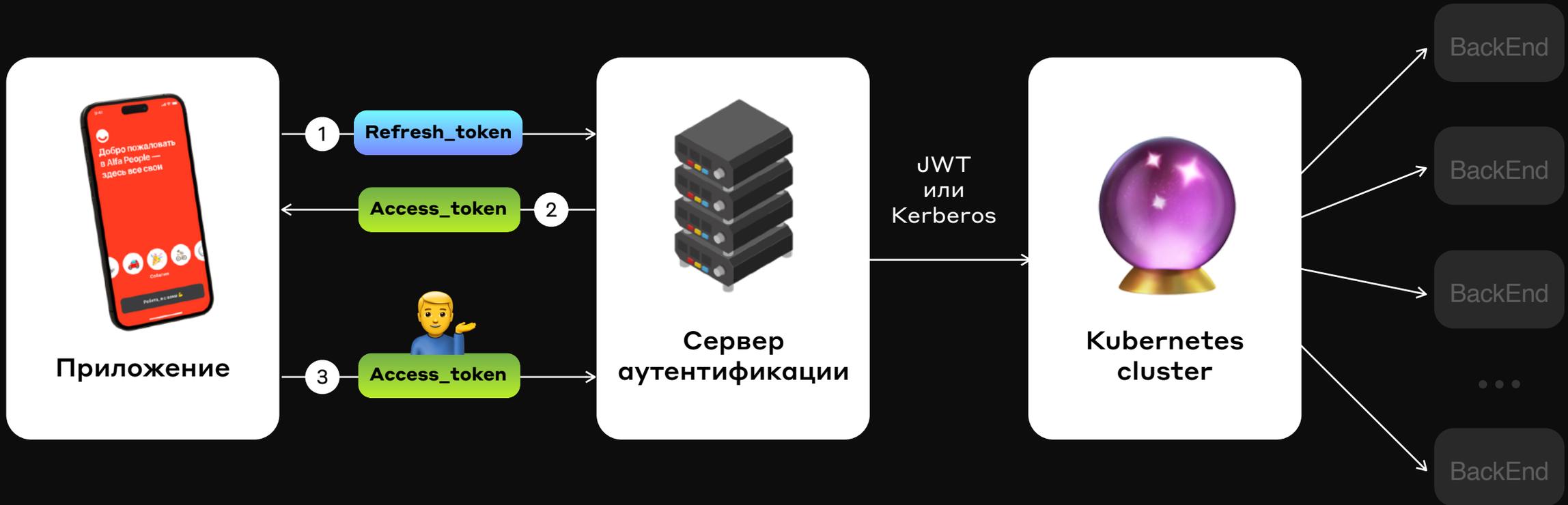
Если пользователь ввел 10 и более раз некорректный пароль. Снять блокировку можно только через обращение в поддержку



**Установили системы предотвращения вторжений**

Используем системы защиты от угроз без ущерба для производительности

# Как работает аутентификация



1. Аутентификация

2. Разработка на нативе

3. Дизайн-система

4. Доставка фич

Разрабатываем  
**без нативных**  
разработчиков



Ну не совсем...

# Почему некоторые сервисы лучше разрабатывать на нативе

Экономия ресурсов разработки



**Есть SDK для подключения сторонних сервисов**

Пример: интеграции с данными фитнес-приложений



**Если фича уже на нативе, не нужно её переписывать**

Пример: аутентификация или смена пароля

1. Аутентификация

2. Разработка на нативе

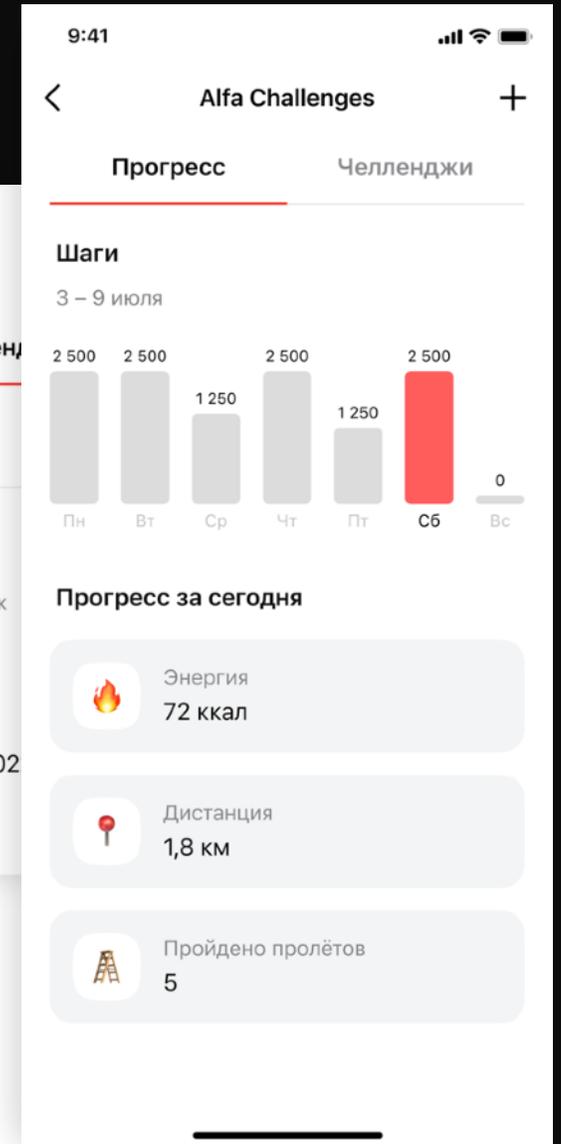
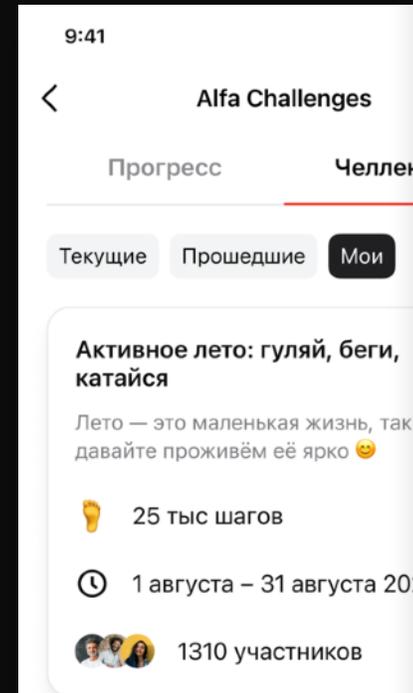
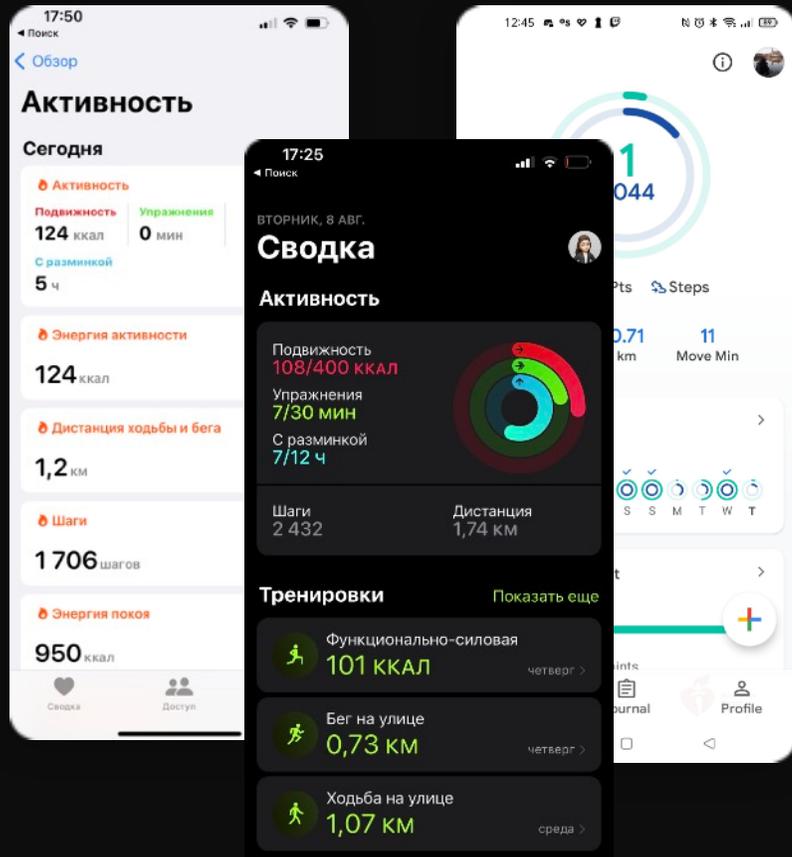
3. Дизайн-система

4. Доставка фич

# Кейс. Alfa Challenges



Здоровье & фитнес – iOS  
Google Fit – Android



1. Аутентификация

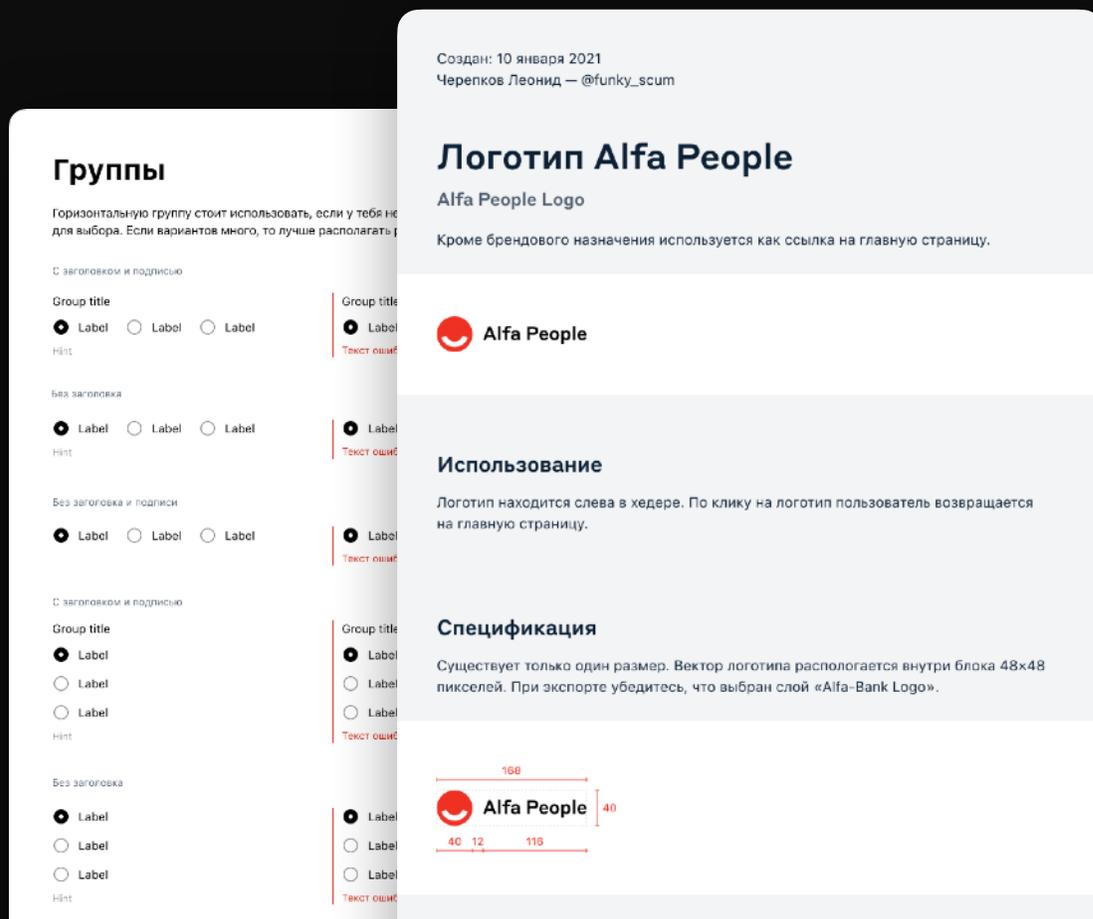
2. Разработка на нативе

3. Дизайн-система

4. Доставка фич

# Используем единую дизайн-систему Альфа-Банка

Native



WebView

Компоненты  
выглядят  
одинаково  
как на native,  
так и на WebView

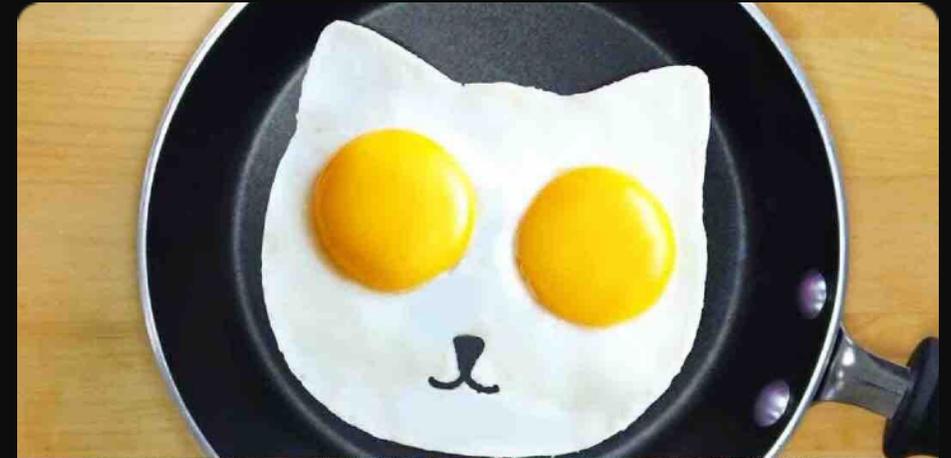
1. Аутентификация

2. Разработка на нативе

3. Дизайн-система

4. Доставка фич

**Но** ожидания  
и реальность иногда  
не совпадают



1. Аутентификация

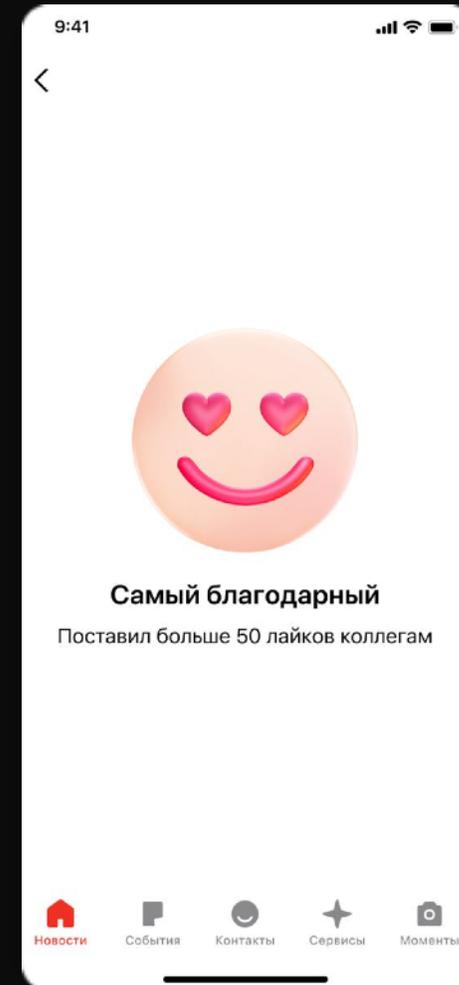
2. Разработка на нативе

3. Дизайн-система

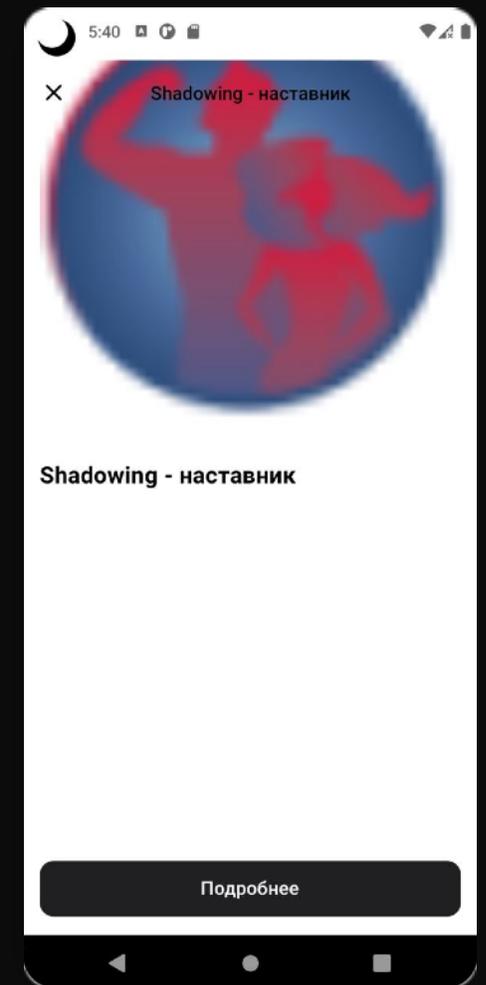
4. Доставка фич

# Случаются баги в компонентах дизайн-системы

Дизайн 🥰



Реализация 🤪



1. Аутентификация

2. Разработка на нативе

3. Дизайн-система

4. Доставка фич

Доставляем  
новые сервисы  
без обновления  
приложения



# Не совсем без обновления

Сценарии, где нужен релиз в сторы:



Переход  
по внешним  
ссылкам



Переходы на другие  
сервисы и нативные  
экраны



Скрытие сервисов  
WebView



Загрузка  
файлов



Принимайте решения, используя ваши ограничения

Ограничения – скрытые преимущества

Не бойтесь интегрировать сервисы с помощью WebView

Проработайте как можно  
больше сценариев заранее



# Буду рад вопросам

## Артём Зяблицев

Team Lead в Альфа-Банке



@EXXYO