

# КМР в X5 Tech: переход бигтеха на новые технологии

Задавайте вопросы в чат



## Вопрос 1

**Расскажите о себе, своей роли в X5  
и продукте, над которым вы работаете**

# Юрий Гришин



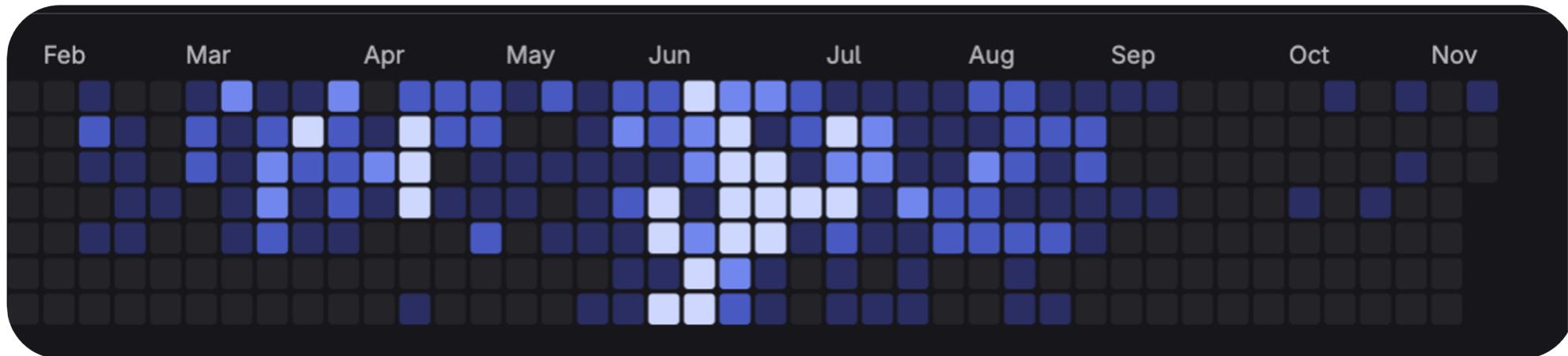
Руководить группой разработки сервиса для сотрудников



Мобильное приложение СМР



BDUI на DivKit + Kotlin Spring



## Сфера Х5 – единое цифровое пространство для сотрудников

Приложение, которое объединяет ключевые сервисы, коммуникации и персональную информацию в одном месте

### С помощью приложения любой сотрудник может:

- Отслеживать и анализировать начисление заработной платы

- Контролировать свой график работы

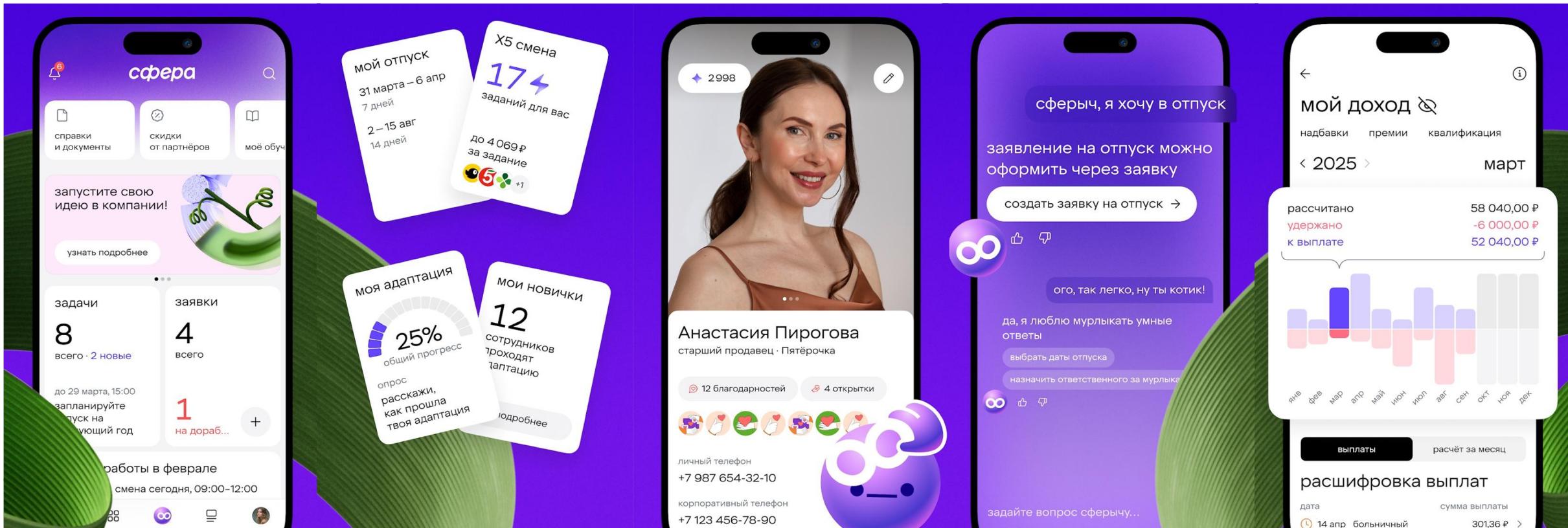
- Работать с сервисами портала HRO

- Отслеживать отпуска и больничные

- Заказывать справки и кадровые документы

- Получить электронную подпись

- Подписывать документы электронно



# Алексеенков Дмитрий



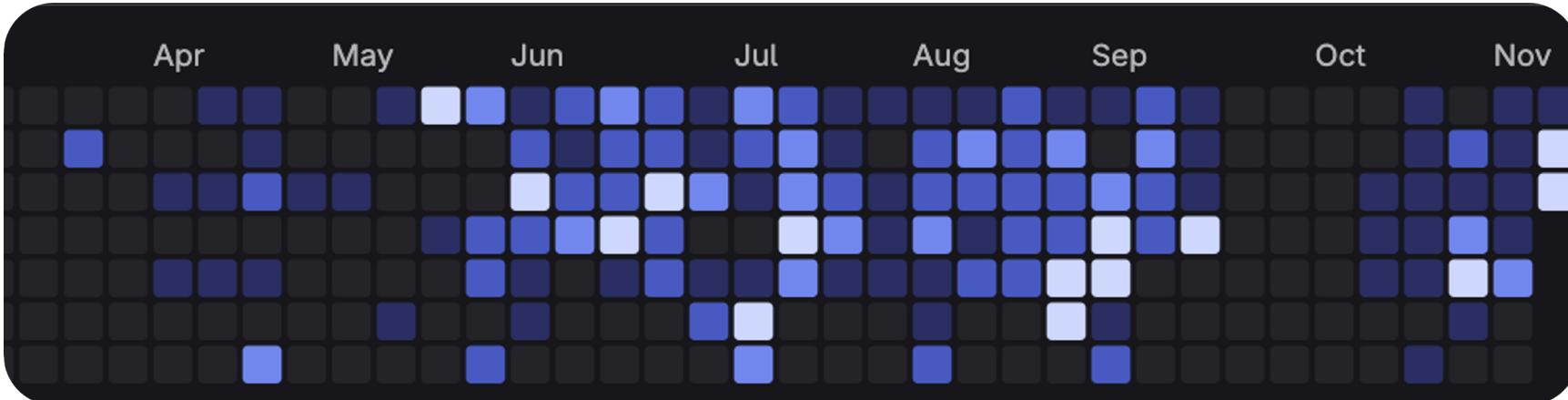
12 лет в мобильной разработке



6 лет в X5 Tech



КМР проектов в проде - 2



# Баранов Павел



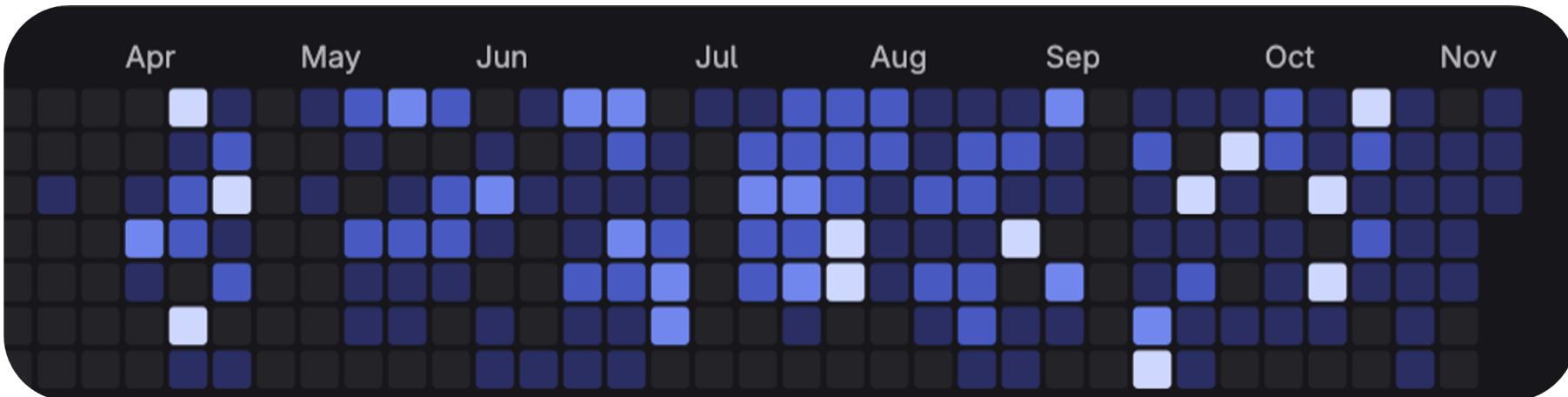
15 лет в разработке



8 лет в мобильной разработке



5 лет в ритейле



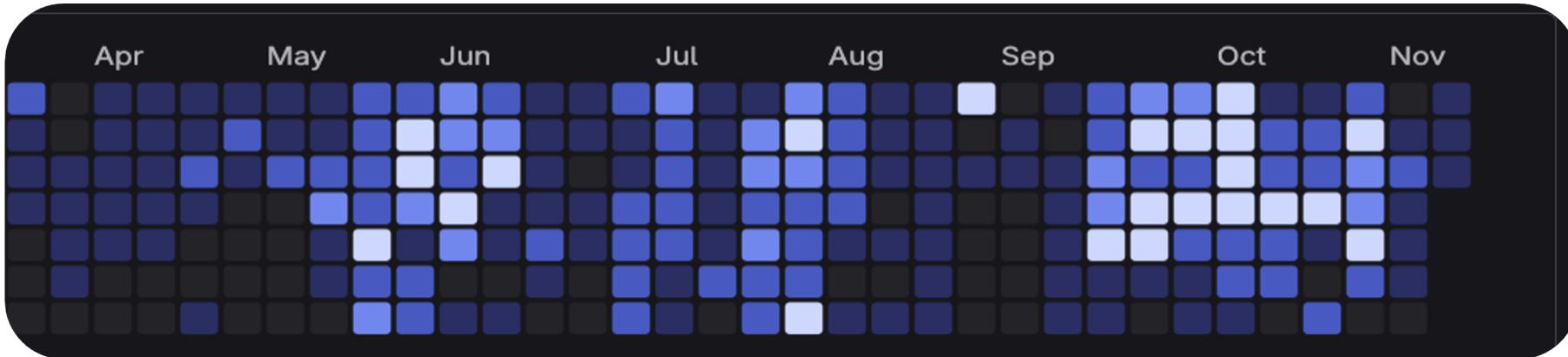
# Денис Александров



Технический менеджер в приложениях  
торговых сетей

Занимаюсь объединением опыта мобилок  
разных торговых сетей с помощью КМР

Помогаю разрабатывать технологическую  
стратегию в X5 со стороны мобилок



# Анна Жаркова



Руководитель мобильной практики в Usetech



Эксперт по Kotlin Multiplatform



Спикер на конференциях AppsConf, Mobius, TechTrain, DroidCon (2022)



Член ПК Mobius, Codefest, «Стачка»



Преподаватель в Otus (iOS Pro и базовый)



Автор статей по мобильной разработке (SwiftUI, iOS, KMM)



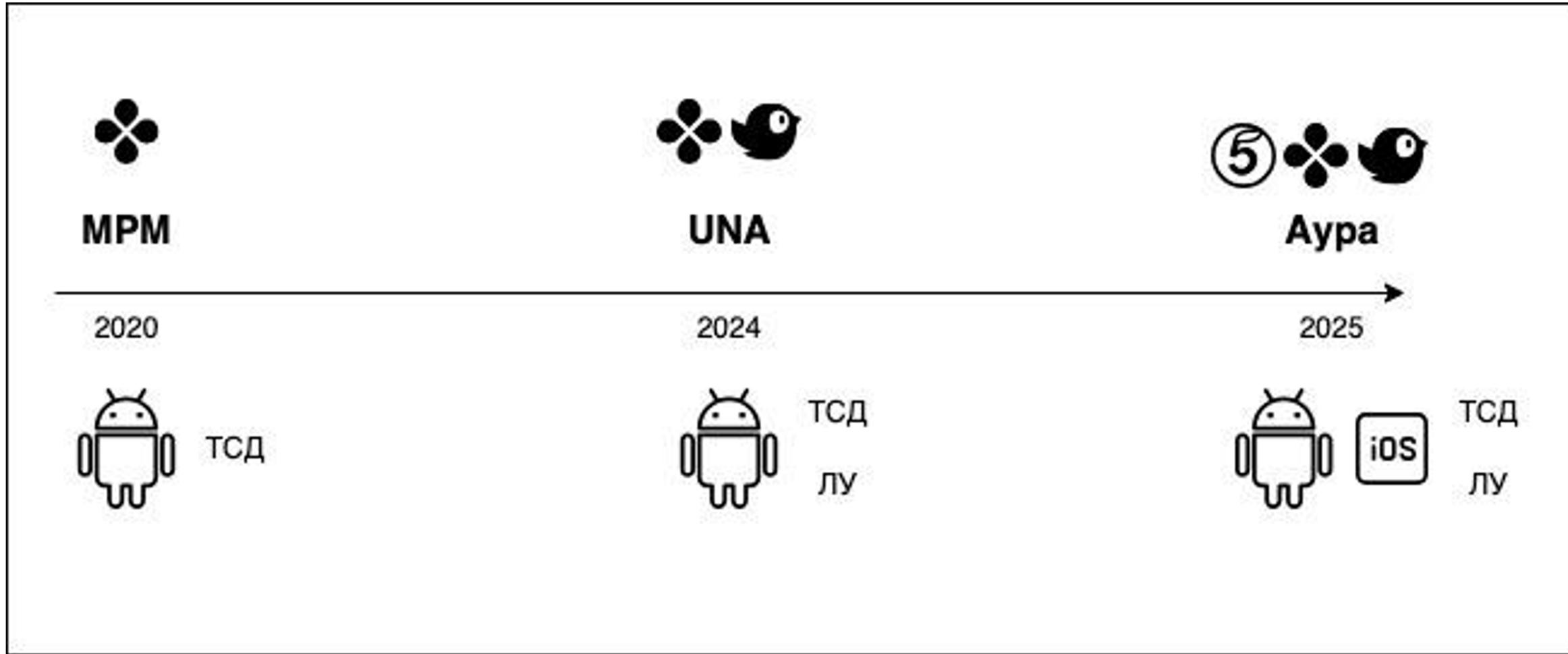
Автор книги «Kotlin Multiplatform на практике»



## Вопрос 2

Почему вы выбрали КМР?  
Как переехали?

# Почему вы выбрали КМР? Как переехали?

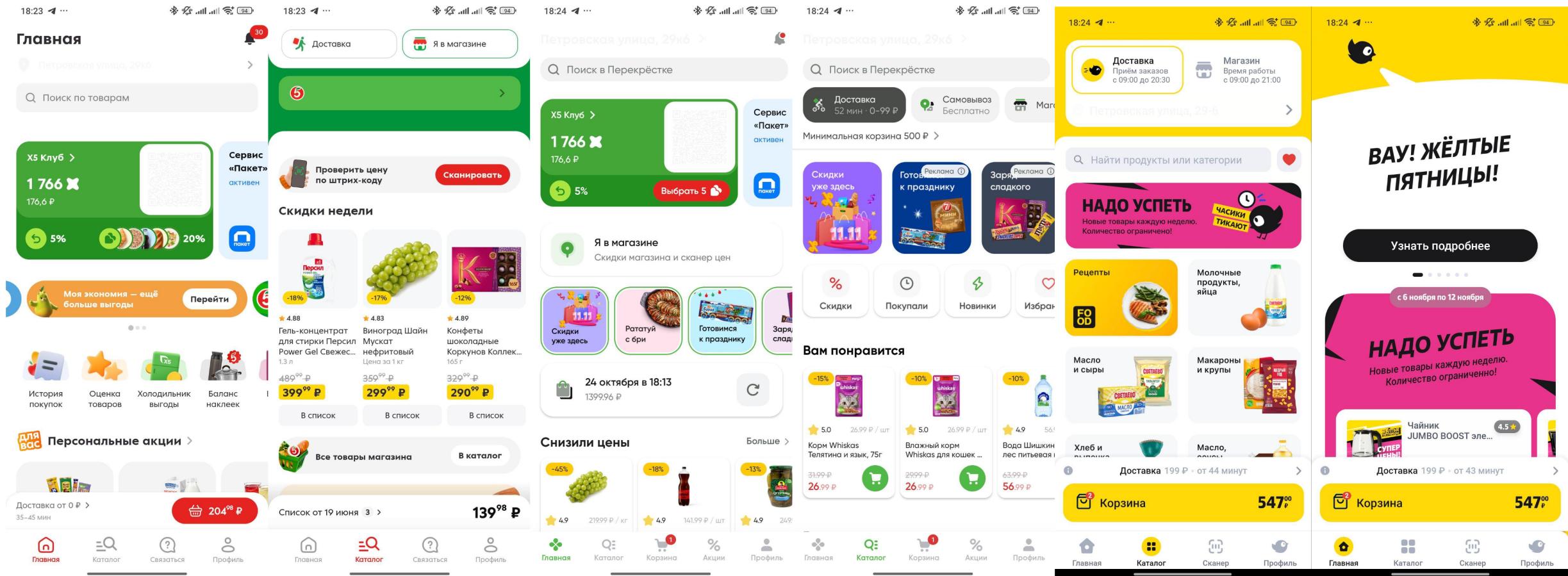


# КМР в X5 Tech: переход бигтеха на новые технологии

Задавайте вопросы в чат



# Почему вы выбрали КМР? Как переехали?

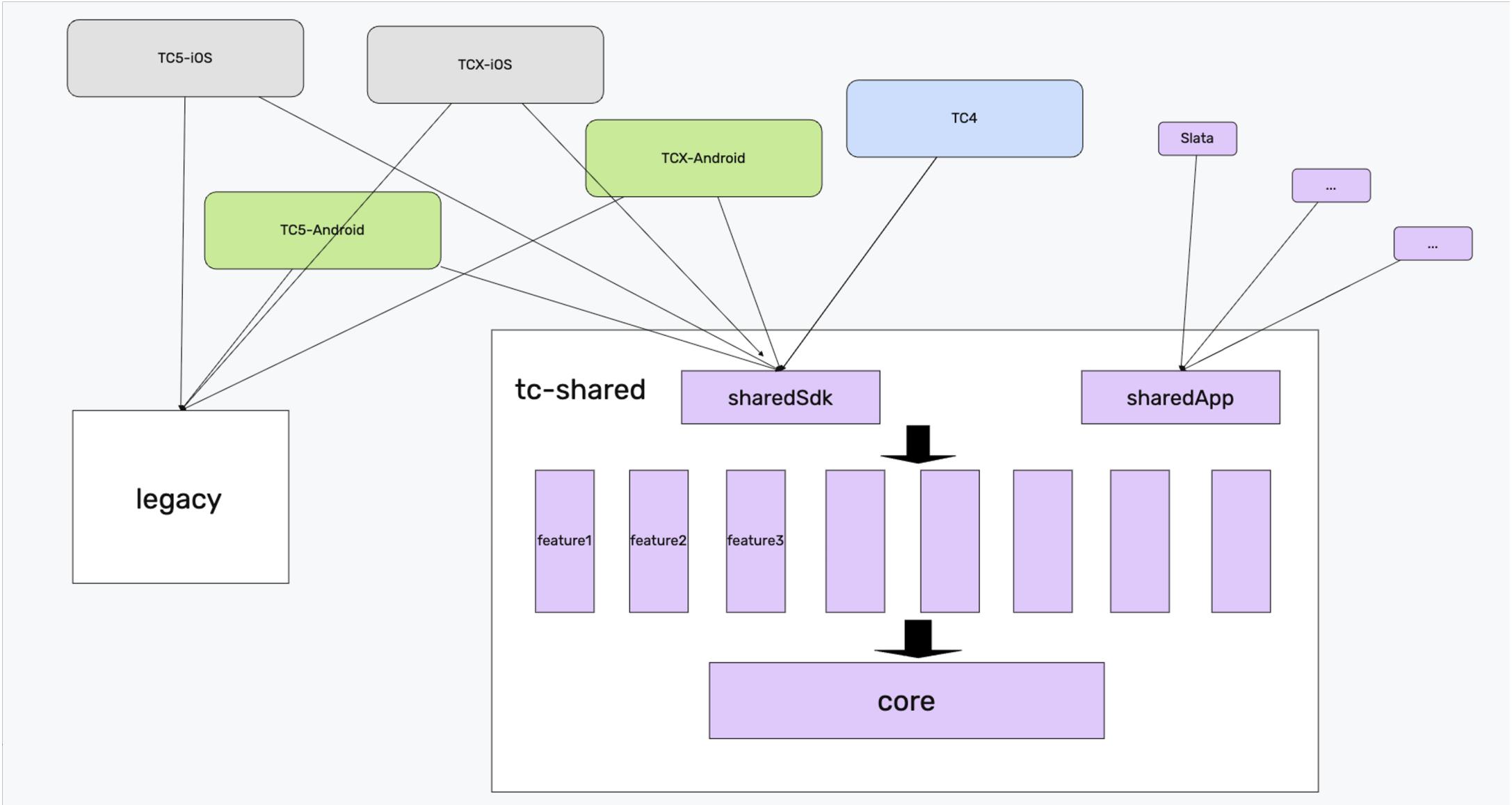


# Почему вы выбрали КМР? Как переехали?

А еще Слата и Vprok



# Почему вы выбрали КМР? Как переехали?



# Почему вы выбрали КМР? Как переехали?

## Задача

-  Объединить приложения в единый сервис
-  Сократить время выхода фич
-  Убрать барьеры между Backend и Mobile

## Техническое решение

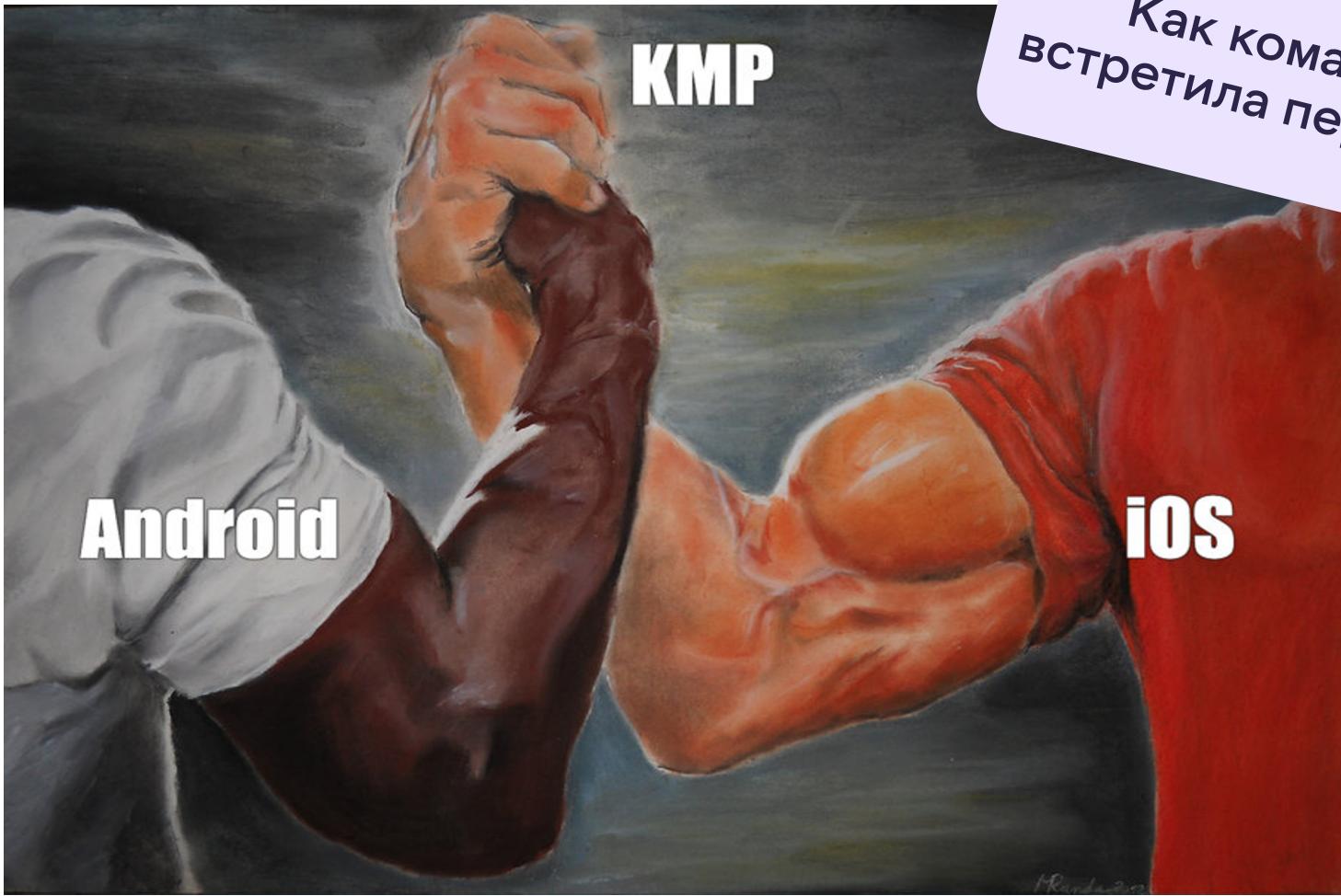
-  BDUI/SDUI подход
-  DivKit для динамических интерфейсов
-  Единый язык разработки

## Выгода

-  Единая точка входа, согласованный UX
-  Обновления без публикации в AppStore/GooglePlay
-  Скорость разработки и общая экспертиза

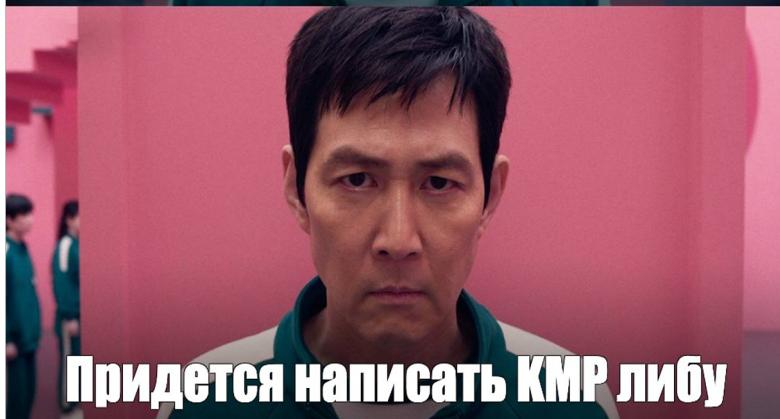
## Вопрос 3

**Как команда встретила переезд?  
Что болело больше всего?**



# Как команда встретила переезд?

## Проблемки



Cocoapods

SPM



Карл

Все-таки надо на Swift

# Как команда встретила переезд?



# Как команда встретила переезд?

## Неочевидные ошибки

try-catch на iOS не работает без Coroutine Error Handler

```
// Так падает на iOS!
val coroutineScope = rememberCoroutineScope()
coroutineScope.launch(Dispatchers.IO) {
    try {
        throw Exception("Some error!")
    } catch (e: Exception) {
        // Не сработает!
    }
}
```

## Неожиданные NPE на iOS

В своем приложении мы используем апметрику. При передаче в параметре map с вложенными map возникал NPE

```
AppMetrica.reportEvent(
    name: name,
    parameters: map
)
```

## Решение

Обязательно использовать supervisorScope или устанавливать CoroutineExceptionHandler

## Решение

Используем примитивные типы в map параметров события аналитики

# Как команда встретила переезд?



Проблема 1: ресурсы в релизе iOS

▼ Copy Bundle Resources (4 items)

- Assets.xcassets
- Preview Assets.xcassets ...in Preview Content
- GoogleService-Info.plist
- compose-resources ...in ../../app-ios-shared/build/kotlin-multiplatform-resources/aggregated-resources/iosSimulatorArm64

+ -

# Как команда встретила переезд?

## Проблема 2: Coroutine Exception handler



### Coroutine context and behavior

Launching a coroutine from a `CoroutineScope` creates a context that governs its execution. Builder functions like `.launch()` and `.async()` automatically create a set of elements that define how the coroutine behaves:

- The `Job ↗` interface tracks the coroutine's lifecycle and enables structured concurrency.
- `CoroutineDispatcher ↗` controls where the coroutine runs, such as on a background thread or the main thread in UI applications.
- `CoroutineExceptionHandler ↗` handles uncaught exceptions.

These, along with other possible elements, make up the `coroutine context`, which is inherited by default from the coroutine's parent. This context forms a hierarchy that enables structured concurrency, where related coroutines can be `canceled` together or `handle exceptions` as a group.

# Как команда встретила переезд?



## Проблема 3: UIViewControllerRepresentable

Next, you need a structure that represents Compose Multiplatform in SwiftUI. Create the following structure that converts a `UIViewController` instance to a SwiftUI view:

```
struct ComposeViewController: UIViewControllerRepresentable {  
    func makeUIViewController(context: Context) -> UIViewController {  
        return Main_iosKt.MainViewController()  
    }  
  
    func updateUIViewController(_ uiViewController: UIViewController, context:  
    Context)  
}
```

## Вопрос 4

**Какую технологию выберете  
в следующий раз?**

## Вопрос 4

**Какую технологию выберете  
в следующий раз?**

Какую технологию  
выберете  
в следующий  
раз?



## Вопрос 4

**Какую технологию выберете  
в следующий раз?**

Какую технологию  
выберете  
в следующий  
раз?



Я так больше  
не могу

Но буду

# КМР в X5 Tech: переход бигтеха на новые технологии

Задавайте вопросы в чат

