kaspersky

Насколько Kotlin Multiplatform готов для эффективной разработки мобильных приложений

Андрей Берюхов

Обо мне

- **Android**-разработчик
  - **Kaspersky Internet Security for Android**
- Студент магистратуры ВШЭ
- Статьи и доклады (**KMP, Flutter, Compose,...**)

План

Зачем нам кроссплатформа?

Что было до Kotlin MP

Основы Kotlin MP

Пример архитектуры

Эффективность

Сравнение с Flutter

Давайте

знакомиться

Кто нативный Android-разработчик?

Кто нативный iOS-разработчик ?

Кто использовал Kotlin Multiplatform?

Кто использовал Flutter?

Мобильные приложения Лаборатории Касперского

Что можно улучшить?

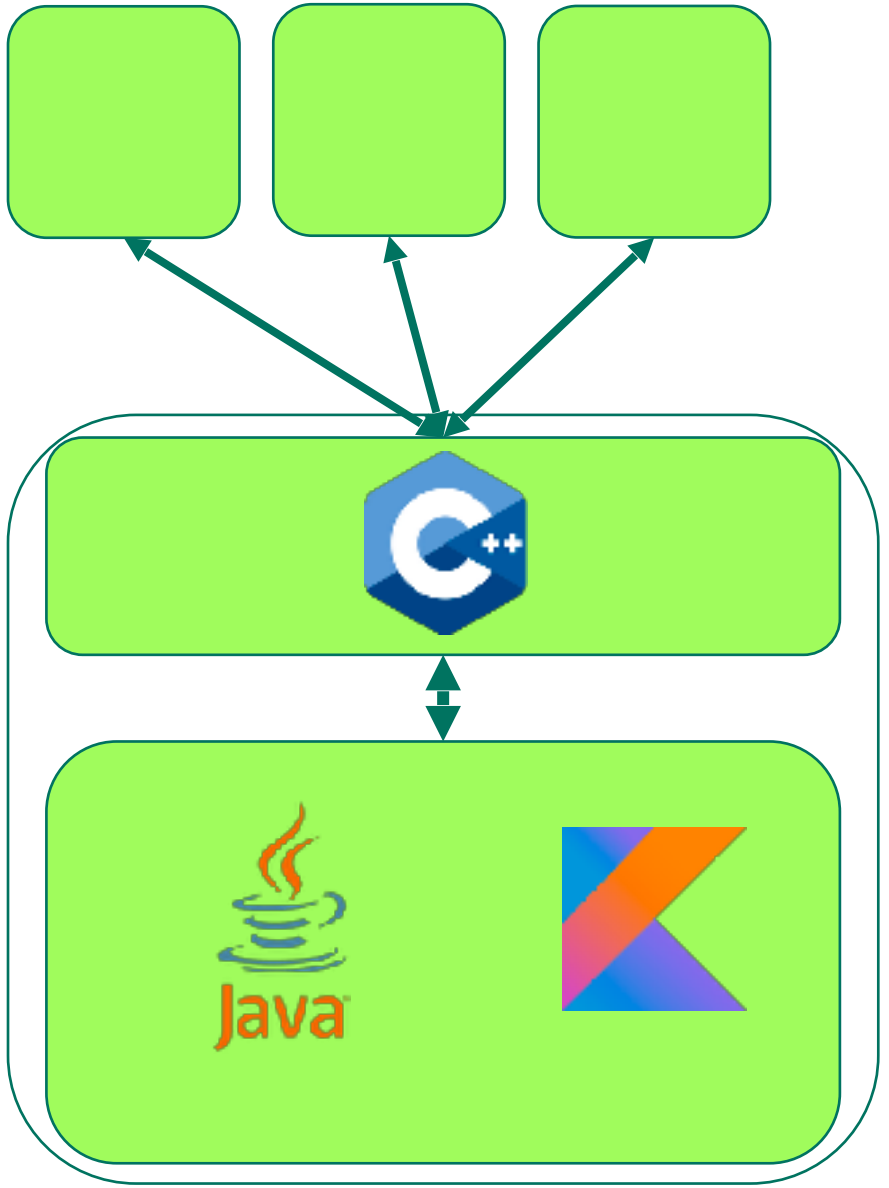
+ Переиспользование кода между платформами
+ Быстрее выпуск фич
+ Меньше багов

C++

Общая библиотека для общения с **backend** в приложениях **Kaspersky**

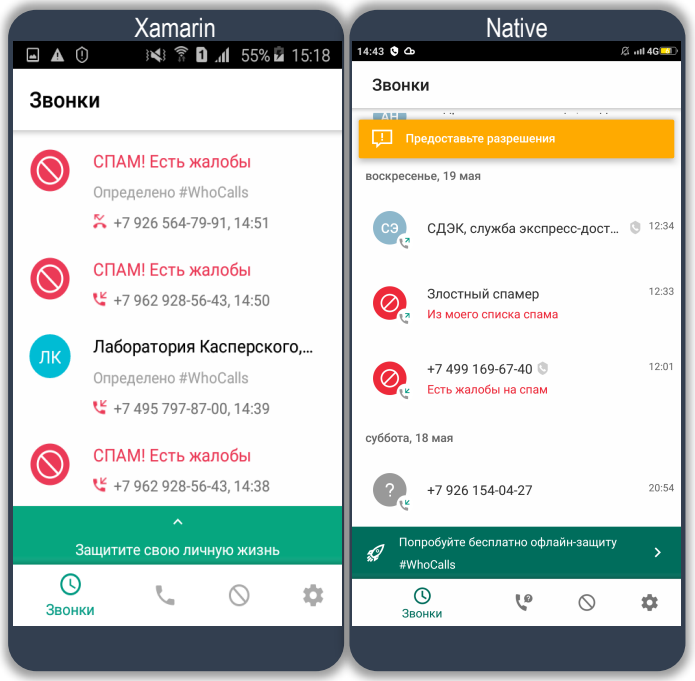
- Низкоуровневость
- Нестандартные внутренние компоненты
- Сложно вникать и поддерживать

+ Мобильный UI на Qt



Xamarin

- **Bridge** для рендеринга **UI**-компонентов
- Медленно
- **Mono GC**

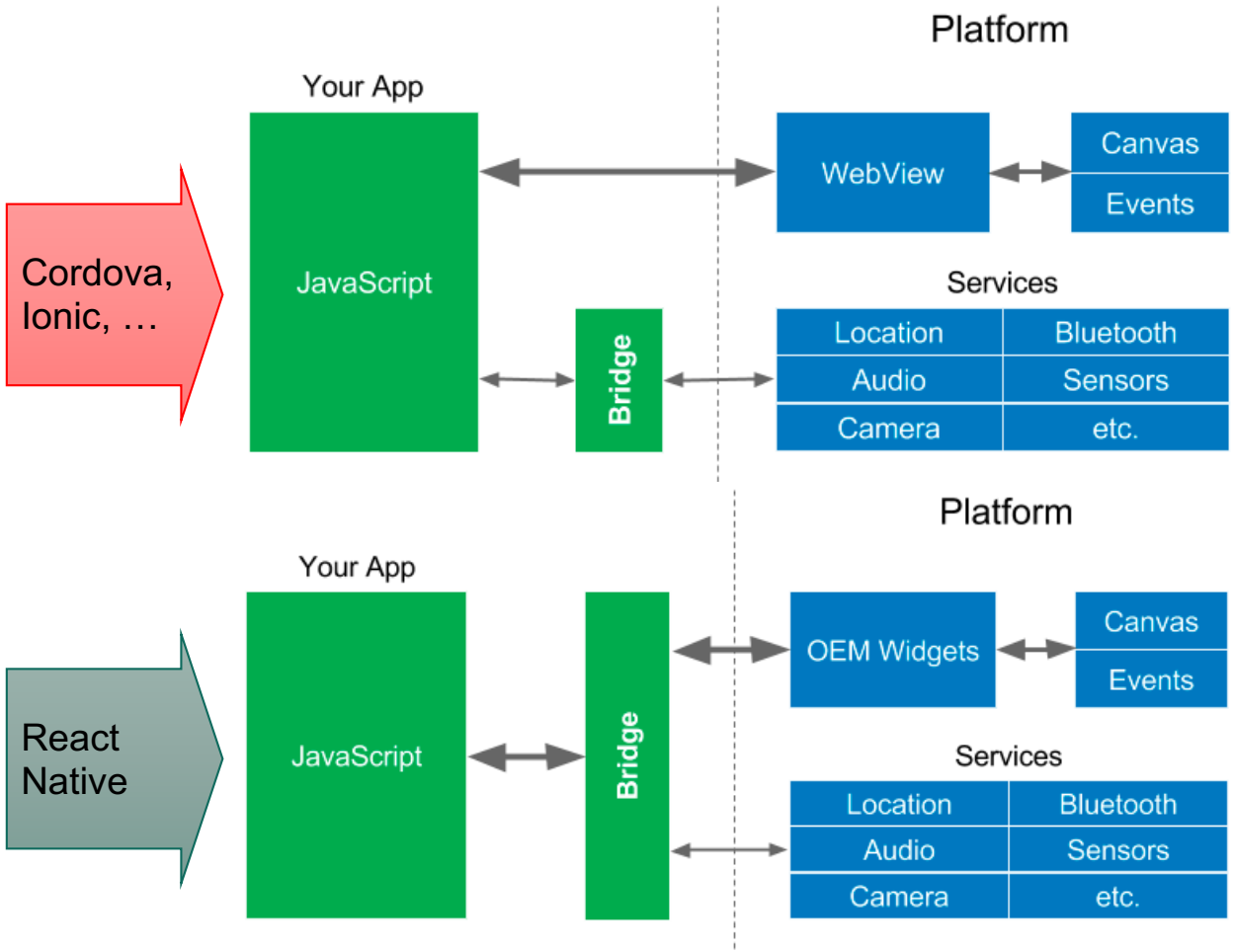


https://youtu.be/NjDC2DXh34c?t=4674 **(RUS)** Путешествие в кроссплатформенность с **Xamarin.Forms**

JS + ReactNative

- JavaScript (JIT-компиляция)

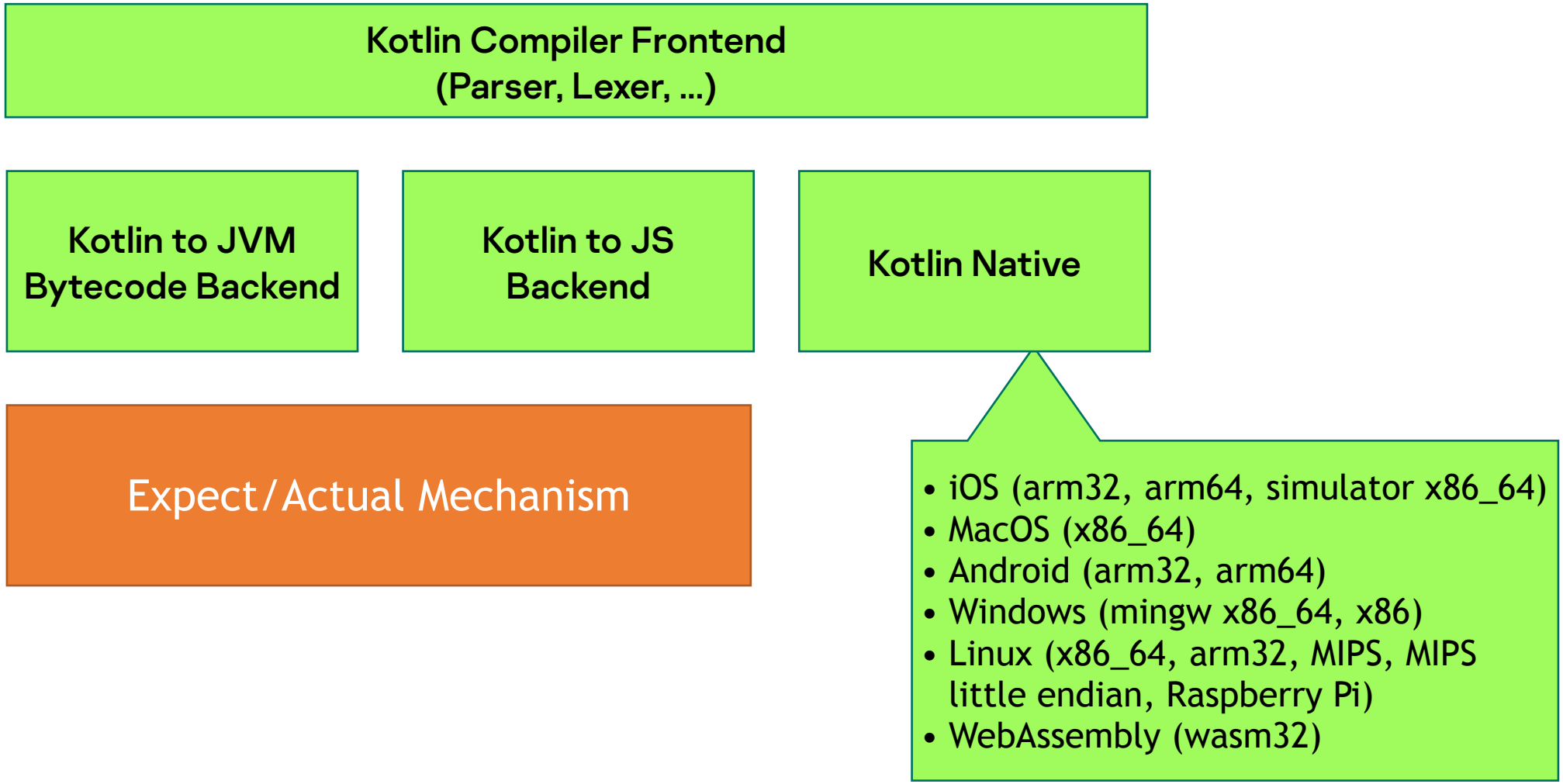
- WebView либо бридж для отрисовки UI-компонентов



kaspersky

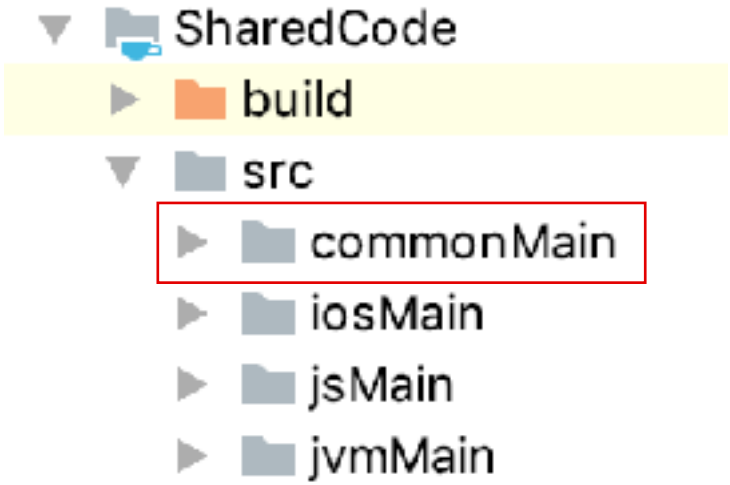
Основы Kotlin Multiplatform

Kotlin Multiplatform



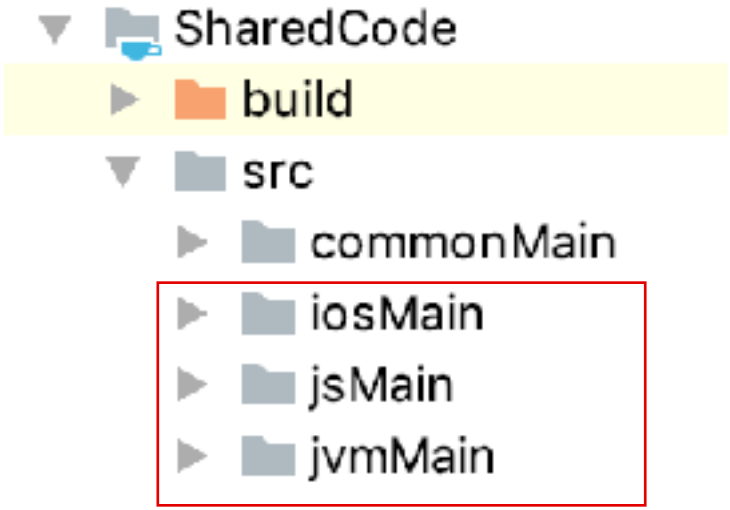
Expect + Actual Mechanism

```
expect val someVal : SomeClass

expect class ExpectSmth{
    expect fun doSmth()
}
```



Expect + Actual Mechanism

```
actual val someVal : SomeClass = SomeClass()

actual class ExpectSmth{
    actual fun doSmth() { /*some logic*/ }
}
```



Coroutines in Native (iOS)

https://github.com/Kotlin/kotlinx.coroutines/issues/462

kotlin.IllegalStateException: There is no event loop. Use runBlocking { ... } to start one.

at 0 SharedCode 0x000000010c44d367 kfun:kotlin.Throwable.<init>(kotlin.String?)kotlin.Throwable + 23

at…

Expect + Actual Mechanism

- **expect val** *uiScope* : CoroutineScope
- **expect val** *processScope* : CoroutineScope
- **expect val** *netScope* : CoroutineScope

https://github.com/chris-hatton/kotlin-multiplatform-template/

Expect + Actual Mechanism

- **actual val** *uiScope* : CoroutineScope = *MainScope* ()
- **actual val** *processScope* : CoroutineScope = GlobalScope
- **actual val** *netScope* : CoroutineScope = GlobalScope

Expect + Actual Mechanism

iOS

```
actual val uiScope : CoroutineScope = createMainScope ()

actual val processScope : CoroutineScope = createMainScope ()

actual val netScope : CoroutineScope = createMainScope ()


private fun createMainScope() {…}
```

Kotlin Coroutines

```
val job = Job()

fun onCreateView() {
    timeSheetView.showProgress()
    netScope.launch(context = job) {
        val dates = async {
            timeSheetInteractor.getDatesList(getStartDate())
        }
        withContext(uiScope.coroutineContext) {
            timeSheetView.clear()
            timeSheetView.addAll(dates.await())
            timeSheetView.hideProgress()
        }
    }
}

fun onDestroyView() {
    job.cancel()
}
```

Kotlin Native

Нужно подобрать версии библиотек, собранные с одинаковой версией компилятора

The abi versions don't match. Expected '[9]', found '14'

w: The compiler versions don't match either. Expected '[1.3.1]', found '1.3.50-release-11850'

Kotlin Native (1.3.60)

Kotlin/Native

The Kotlin/Native compiler has acquired a few new capabilities:

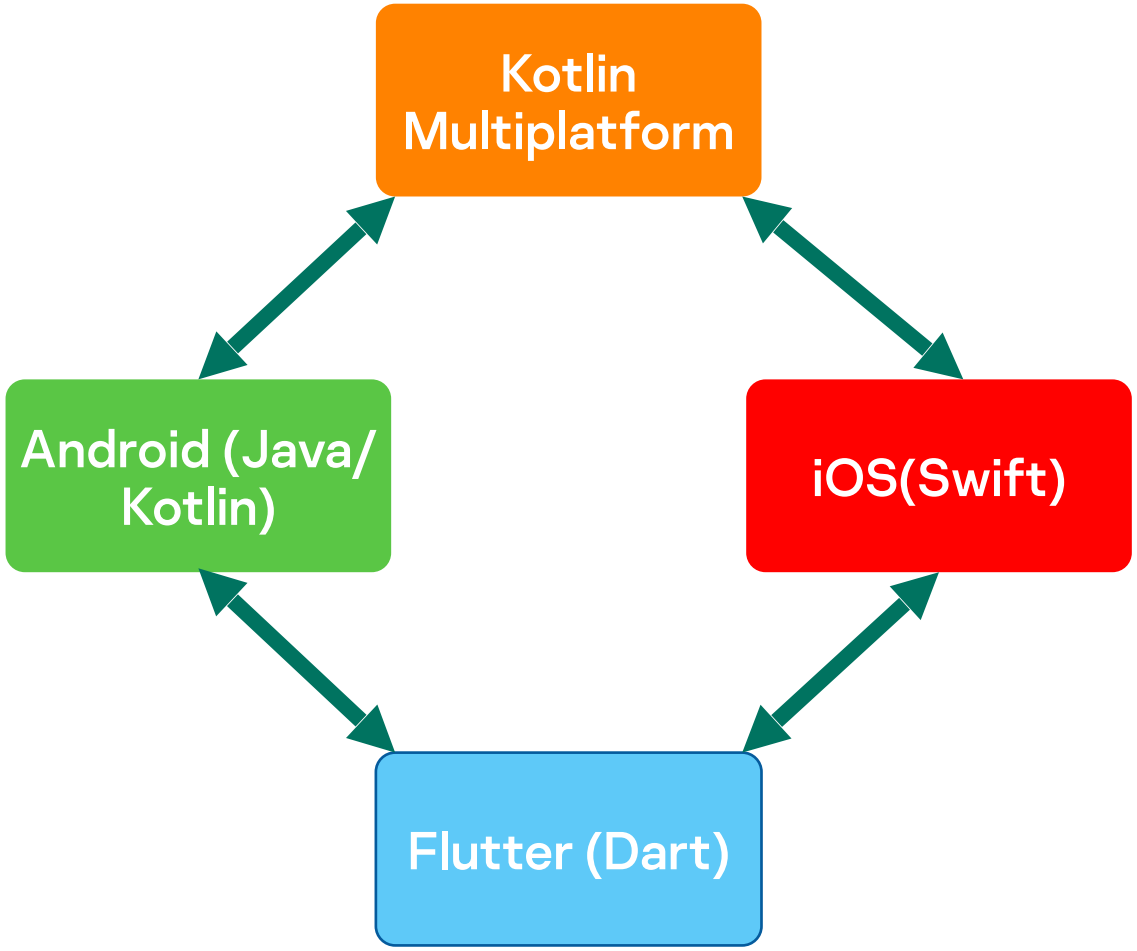
- Compatibility with the latest tooling bits: **XCode 11** and **LLVM 8.0**.
- A plethora of new platforms/targets:
  - *watchOS*
    `watchos_x86`
    `watchos_arm64`
    `watchos_arm32`
  - *tvOS*
    `tvos_x64`
    `tvos_arm64`
  - *Android (native)*
    `android_x86`
    `android_x64`
- Experimental symbolication of iOS crash reports for release binaries (including LLVM-inlined code, which is one step further than what XCode is able to decode).
- Thread-safe tracking of Objective-C weak/shared references to Kotlin objects.
- Support for `suspend` callable references.
- Functions with "big arity" (on par with the JVM limit)
- The ability to associate a work queue with any context/thread, not just the ones created ad-hoc through `Worker.start`.

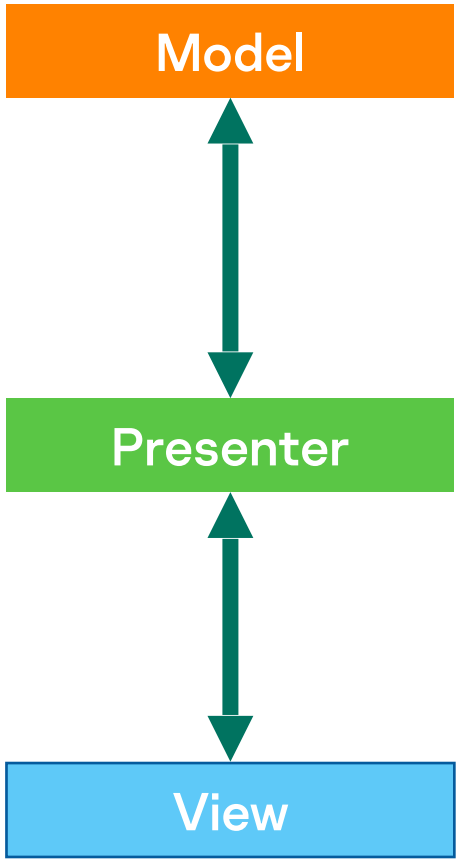
https://blog.jetbrains.com/kotlin/2019/11/kotlin-1-3-60-released/

kaspersky

Про архитектуру

«Fast Prototypes with Flutter + Kotlin/Native»

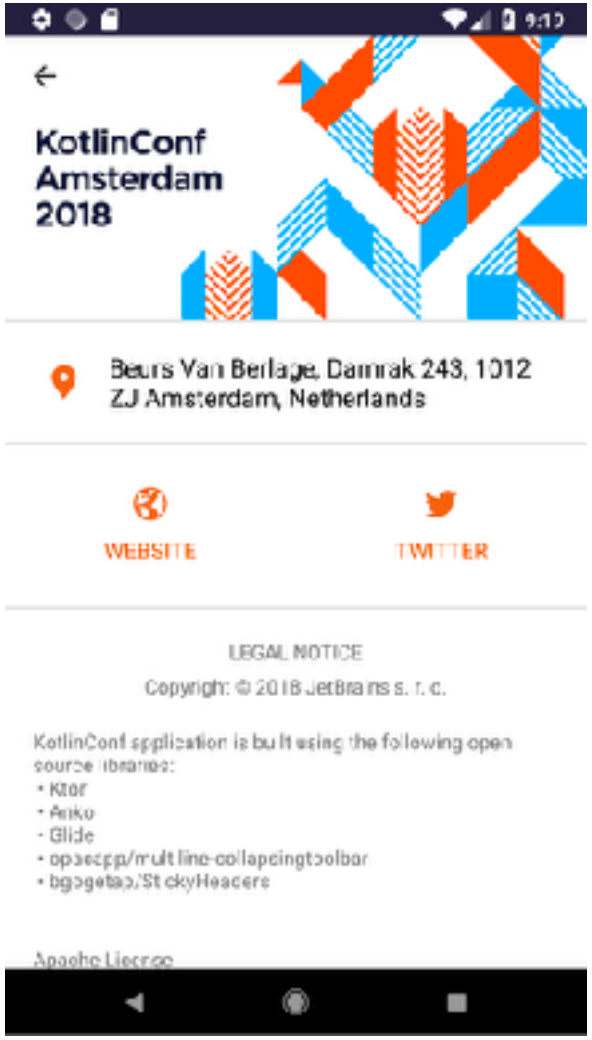




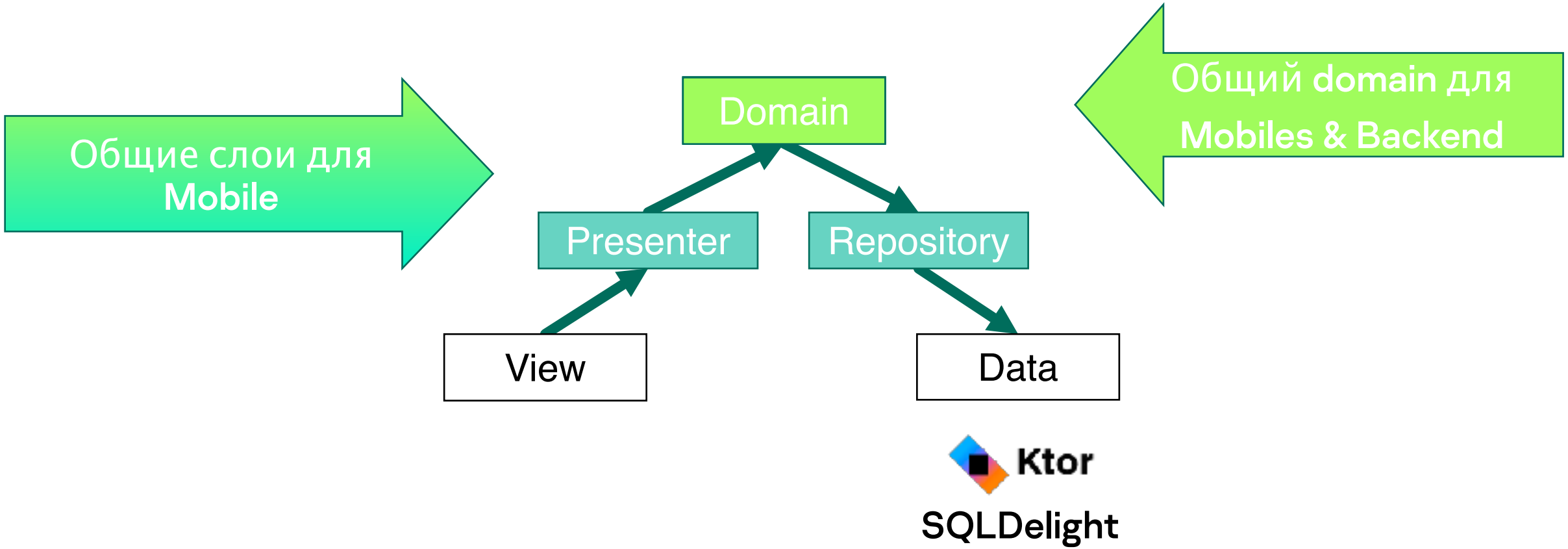
https://tech.olx.com/fast-prototypes-with-flutter-kotlin-native-d7ce5cfeb5f1

KotlinConf Schedule Application <-MVP

https://github.com/JetBrains/kotlinconf-app



MVP + Clean Architecture



Прототип приложения на Kotlin MP

Приложение для фиксирования времени

- 1 экран для каждой ОС
  - Список дат;
  - Кнопки: зафиксировать начало и конец
- **iOS + Android + JS**
- Простой **Backend**

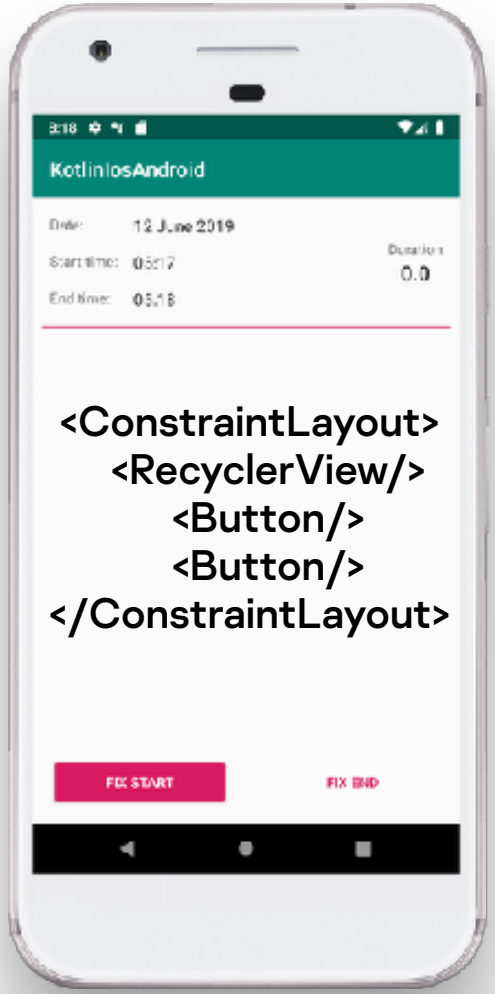
https://github.com/AndreySBer/KotlinIosAndroid

View Interface

```
interface TimeSheetView {
    fun addAll(list:List<DateModel>){}
    fun clear(){}
    fun showError(message:String){}
    fun showProgress(){}
    fun hideProgress(){}
}
```

Android View Implementation

```
class MainActivity : AppCompatActivity(), TimeSheetView {

    private lateinit var adapter: SimpleListAdapter

    override fun onCreate(savedInstanceState: Bundle?) {
        ...
        val presenter = TimeSheetPresenterKmp(this, TimeSheetRepositoryImpl())
        presenter.onCreateView()
        setUpButtons(presenter)
    }

    override fun addAll(list: List<DateModel>) = adapter.add(list.map { ... })
    }

    override fun clear() = adapter.clearAll()

    override fun showError(message: String) {
        Toast.makeText(this, message, LENGTH_SHORT).show()
    }

    override fun showProgress() {…}

    override fun hideProgress() {…}
    ...
```

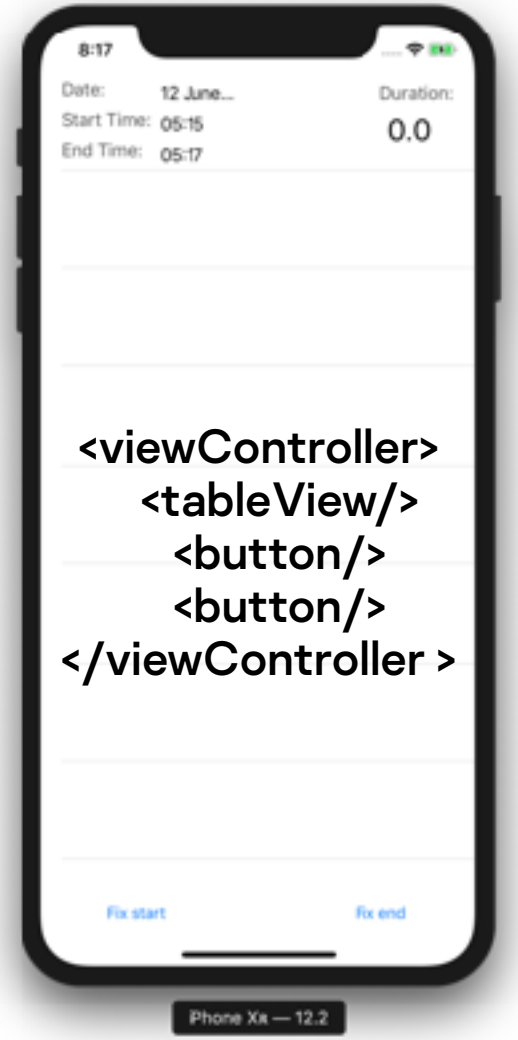


Jetpack Compose

iOS View Implementation

```
class ViewController: UIViewController, TimeSheetView {
    @IBAction func onFixStartButton(_ sender: UIButton) { self.presenter?.onFixStart() }
    @IBAction func onFixEndButton(_ sender: UIButton) { self.presenter?.onFixEnd() }

    override func viewDidLoad() {
        …
        presenter = TimeSheetPresenterKmp(timeSheetView:self,
                                          timeSheetRepository:TimeSheetRepositoryImplSwift())
        presenter?.onCreateView()
    }
    func addAll(list: [DateModel]) {
        self.datesList.append(contentsOf: list); tableView.reloadData()
    }
    func clear() {
        self.datesList.removeAll(); tableView.reloadData()
    }
    func showError(message: String) {…}
    func showProgress() {…}
    func hideProgress() {…}
    …
}
```

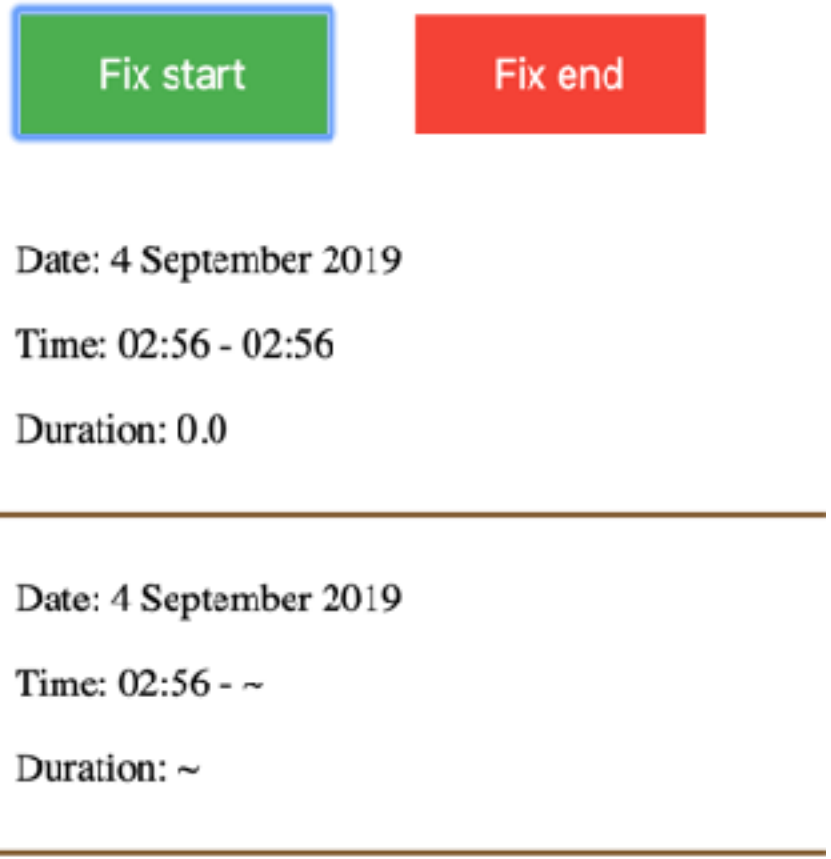
Swift UI



Web (JS) View Implementation

```
fun main() {
    JsView.init()
}

object JsView : TimeSheetView {
    fun init() {
        window.onload = {

            val presenter = TimeSheetPresenterKmp(this, TimeSheetRepositoryImpl()) //create presenter
            presenter.onCreateView()

            document.body!!.append.div {//create buttons
                button {
                    onClickFunction = presenter.onFixStart()
                    style = greenButtonStyle
                    +"Fix start"
                }
            }
            //create list
        }
    }

    override fun addAll(list: List<DateModel>) {
        list.forEach {
            document.body?.append?.div(classes = divClass) {
                style = divStyle
                p(classes = spanClass) {
                +"Date: ${it.date}"
                }…
            }
        }
    }
}
```

<- kotlinx.html



Полезные ссылки: https://github.com/Kotlin/kotlin-frontend-plugin

https://github.com/JetBrains/create-react-kotlin-app

Backend (JVM)


```
routing {
    get("/") {
        call.respondHtml {
            head {
                title("Hello from Ktor!")
            }
            body {
                script(src = "/static/KotlinIosAndroid-SharedCode.js") {}
            }
        }
    }

    static("/static") {
        resource("KotlinIosAndroid-SharedCode.js")
    }
}
```

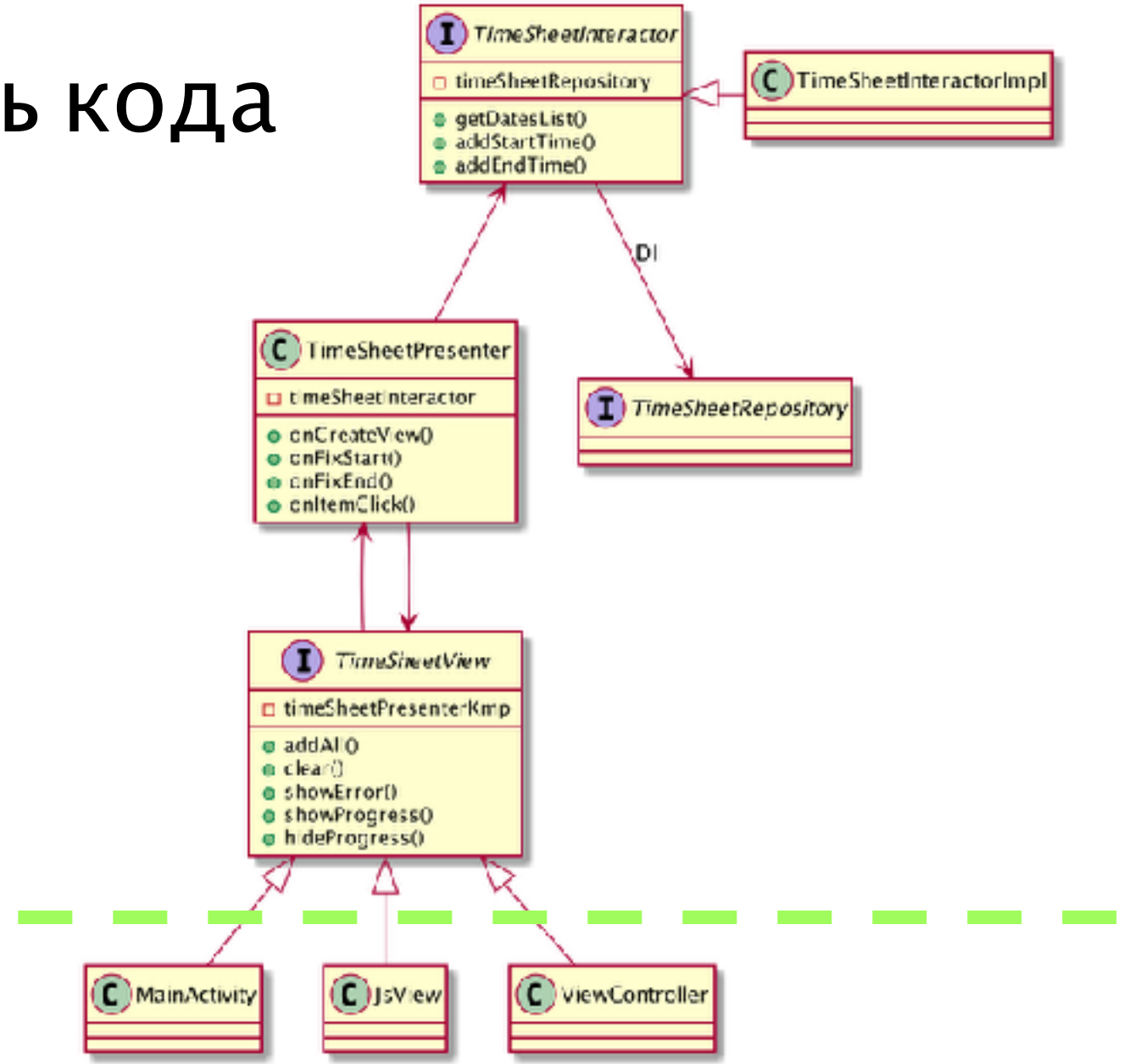
Переиспользуемость кода

Добавление нового клиента ==
имплементация View + Repository

Нативный UI (Look & Feel)

Нативный доступ к платформе

Общая бизнес логика



kaspersky

Эффективность

=Effective

Сценарии использования

Новые прототипы/
приложения

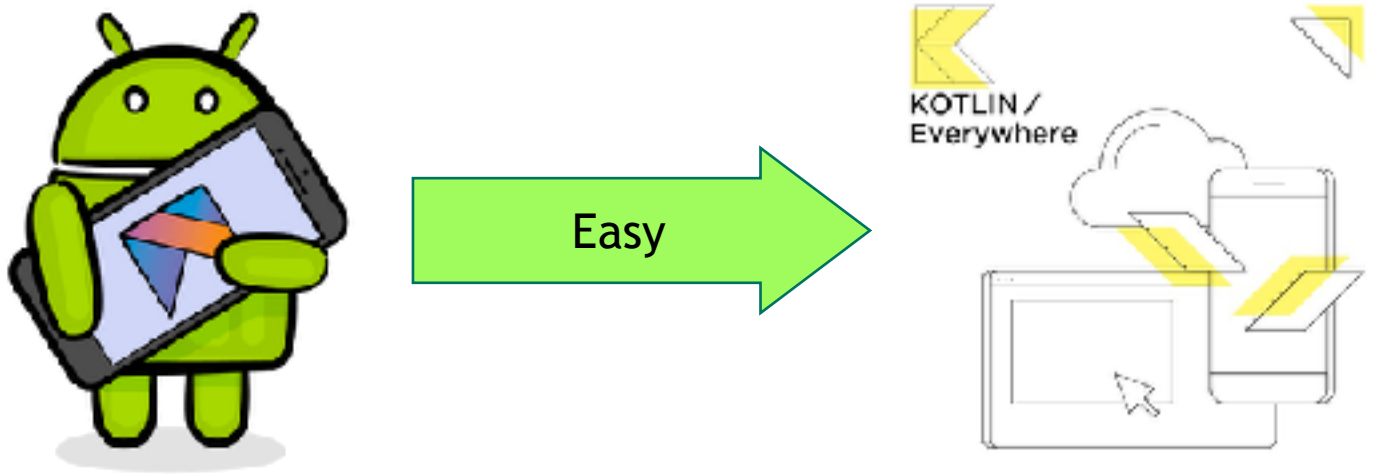
В существующих
приложениях

Критерии

- Стоимость внедрения
- Скорость компиляции и работы приложения
- Размер сборки
- Тестирование
- **Continuous Integration**
- Переучивание разработчиков
- Комьюнити (библиотеки, статьи, ...)

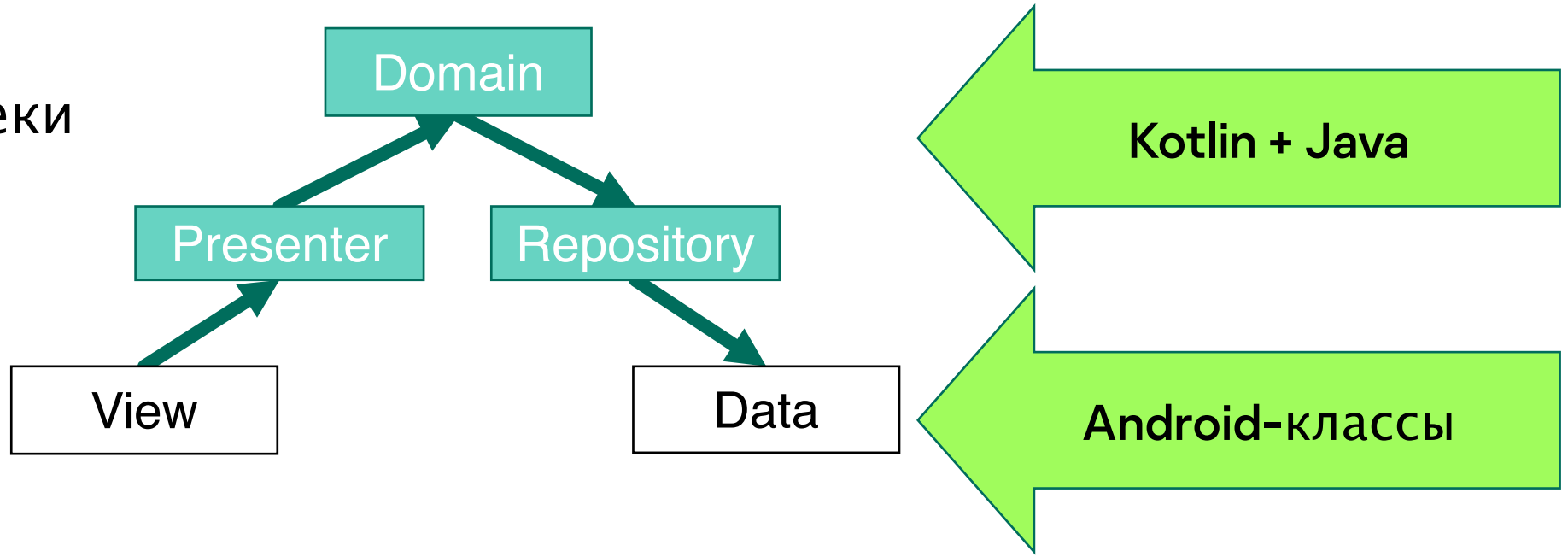
Стоимость внедрения (в **Android** приложение)

- Правильная архитектура решает
- Роберт Мартин: архитектура не должна зависеть от имплементации
- Роберт Мартин: **Clean Architecture**



Текущее состояние в KISA

+ Clean Architecture + MVP
+ Почти не используем Android-классы в Presenter, Repository; Domain
+ Kotlin во всех новых файлах
- RxJava2
- Java библиотеки



Готовимся к KMP в KISA

+ Clean Architecture + MVP
+ ~~Почти~~ не используем Android-классы в Presenter, Repository; Domain
+ Kotlin во всех ~~новых~~ файлах
- ~~RxJava2~~ => Coroutines + Flows
- Java библиотеки => Используем через интерфейсы и заменяем

Скорость компиляции и работы

- Нативные слои: **UI** и **Repository** => Полный доступ к платформе
- **Native** -> долго собирается
- **Experimental** =

Performance

Even though the Kotlin/Native compiler is yet to be deeply optimized for performance, this release brings a few improvements that result in some impressive speedups.

The compilation speed, especially for large projects, has been increased by producing native libraries directly from klibs (instead of sources).

The runtime performance has also been improved: interface calls are now up to **5x** faster, and type checks up to **50x** faster!

https://blog.jetbrains.com/kotlin/2019/11/kotlin-1-3-60-released/

Gradle + Размер сборки

implementation project(':SharedCode')

~35KB

iOS

task packForXCode => SharedCode.framework

~6.8MB

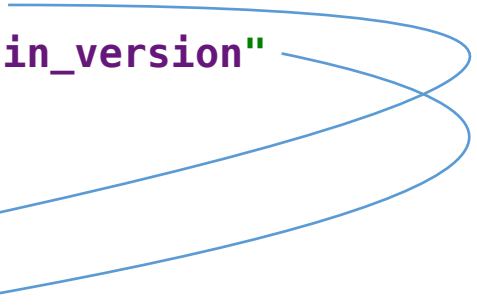
jsBrowserWebpack => SharedCode.js

~12MB

Тестирование

Expect

```
commonTest.dependencies {
    implementation "org.jetbrains.kotlin:kotlin-test-common:$kotlin_version"
    implementation "org.jetbrains.kotlin:kotlin-test-annotations-common:$kotlin_version"
}

jvmTest.dependencies {
    implementation "org.jetbrains.kotlin:kotlin-test:$kotlin_version"
    implementation "org.jetbrains.kotlin:kotlin-test-junit:$kotlin_version"
}
```



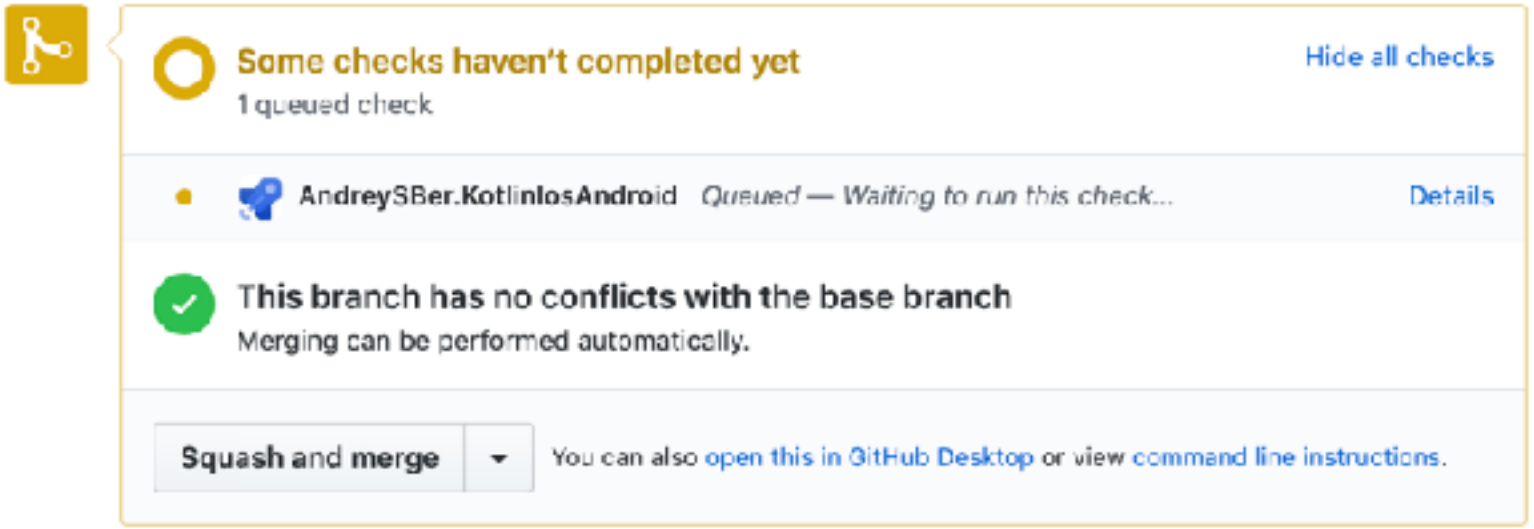
Actual

```
class SampleTest {
    @Test
    fun test() {
        assertTrue(2 + 2 == 4)
    }
}
```

CI (Continuous Integration)

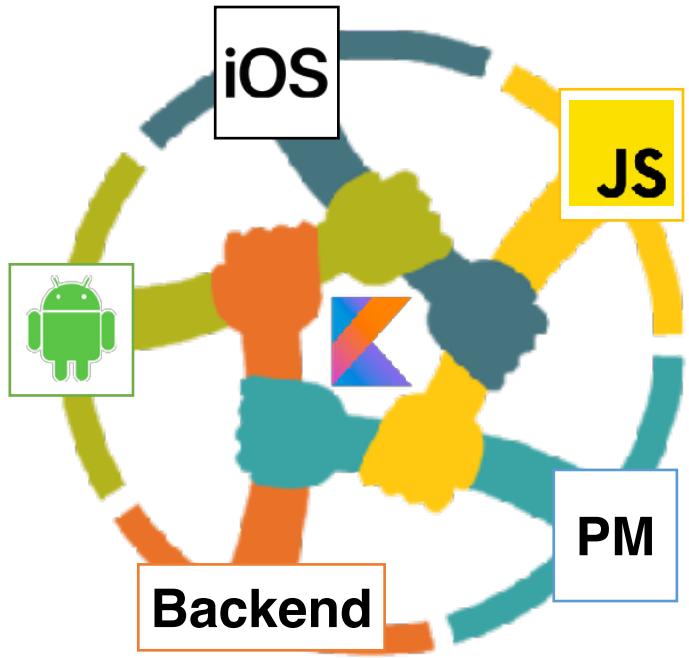
https://dev.azure.com

tasks: 'app:assembleDebug packForXCode jvmTest'



Переучивание разработчиков

- **Android/Backend =>** Всё то же самое
- **Web =>** кажется, ок
- **iOS =>** Другой мир + Боль



Комьюнити

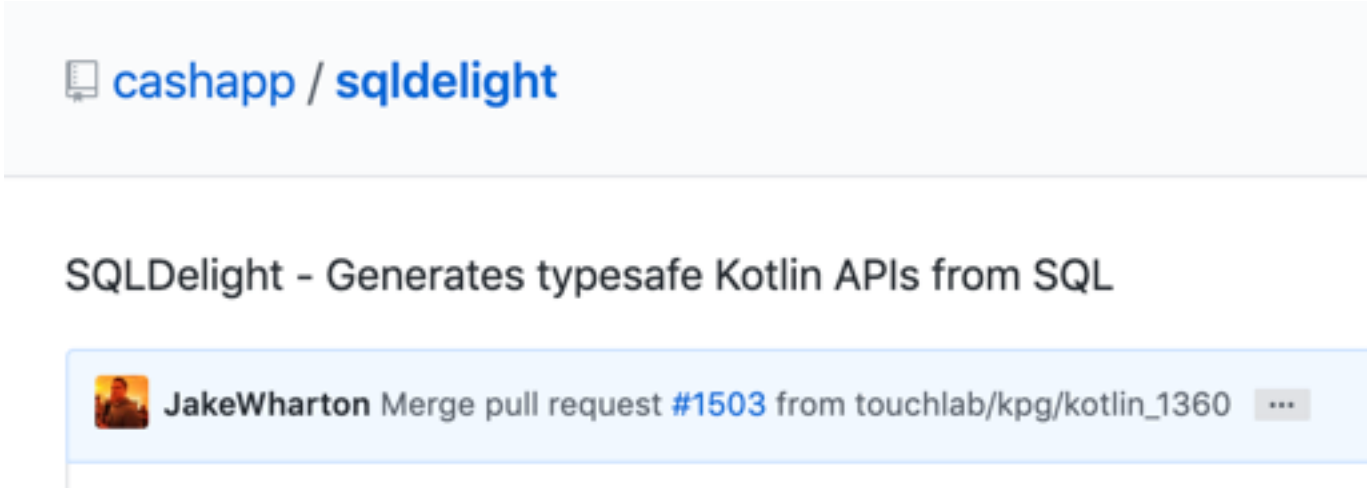
https://github.com/badoo/Reaktive - Rx-библиотека и пример MVI

https://moko.icerock.dev/ - набор KMP библиотек для Android и iOS

https://t.me/kotlinmpp - Kotlin Mobile MultiPlatform Russia (канал + чат)

https://korlibs.soywiz.com/ - набор KMP библиотек для всего

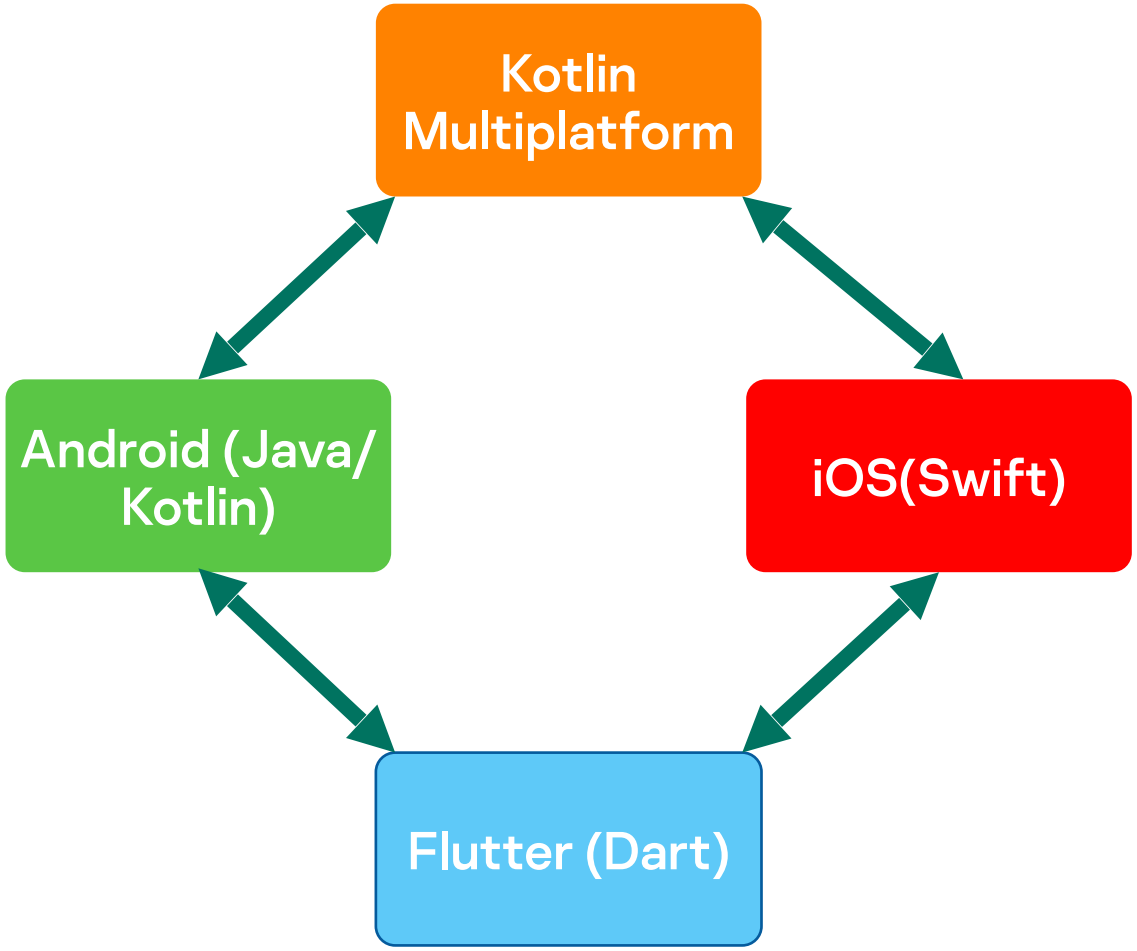
https://touchlab.co/touchlab-square-collaborating-on-kotlin-multiplatform/

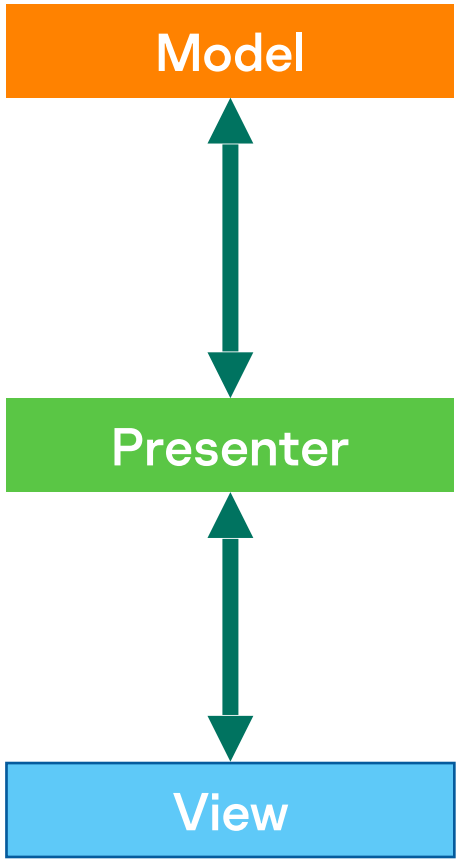


MSK

t.me/KotlinMoscow

«Fast Prototypes with Flutter + Kotlin/Native»



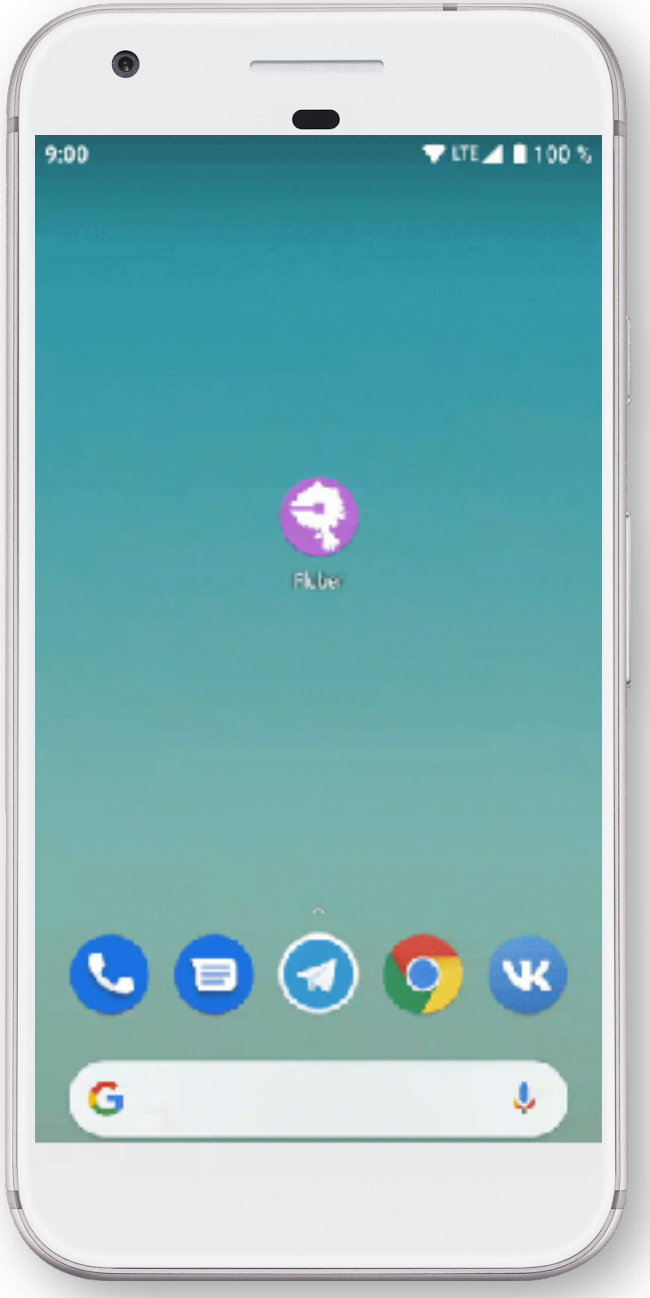


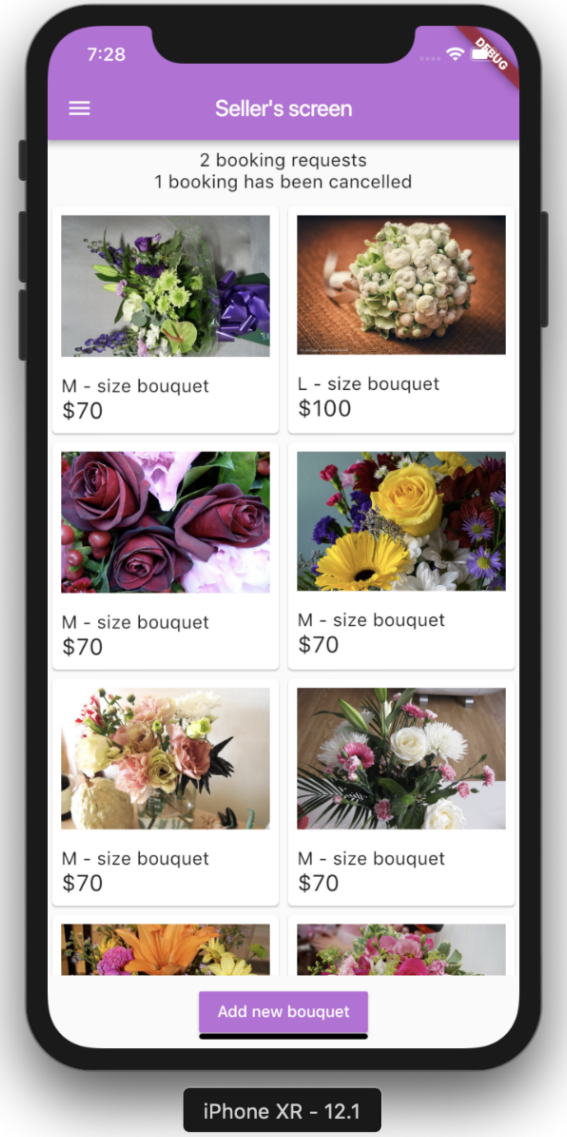
https://tech.olx.com/fast-prototypes-with-flutter-kotlin-native-d7ce5cfeb5f1

Flutter – UI-first

Fluber – Uberisation of Flowers delivery

4 экрана
3 дня на разработку





https://github.com/AndreySBer/FlutterPresentation

Сравнение с Flutter

Нет лимита на minSdk

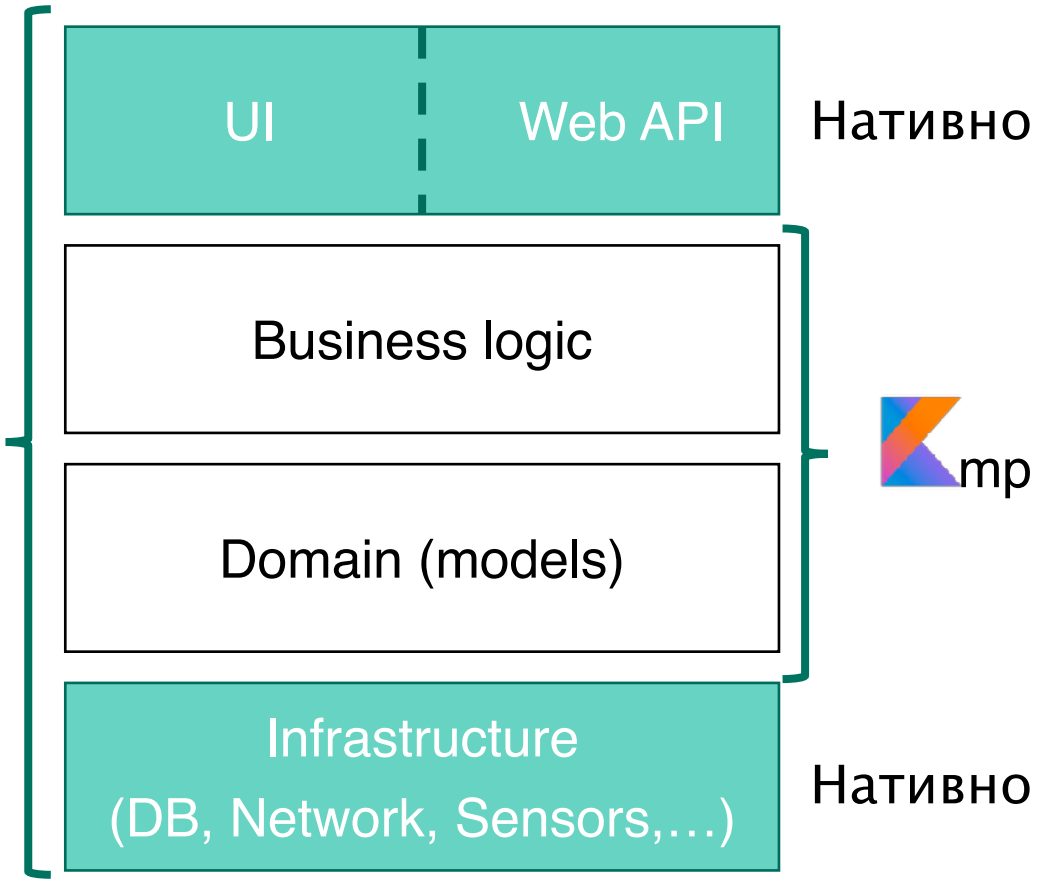
Почему Dart; не Kotlin?

Библиотека UI виджетов


```
import flutter.material.*

fun main() = runApp(MyApp())

class MyApp : StatelessWidget {
    fun build(context: BuildContext) = materialApp {
        title = "Welcome to Flutter"
        scaffold {
            appBar = AppBar(title = "Welcome to Flutter")
            center {
                text("Hello World")
            }}}}
```

Плагины и Боль

Высокоуровневая архитектура приложений



https://hackernoon.com/why-flutter-uses-dart-dd635a054ebf

https://blog.kotlin-academy.com/flutter-and-kotlin-multiplatform-relationship-890616005f57

kaspersky

Спасибо. Вопросы?

Андрей Берюхов

beryukhov.ru
@Phansier