kaspersky

KOTLIN / Everywhere Gorky

Mobile Applications Development with Kotlin Multiplatform

Andrey Beryukhov

About Me

- **Android developer**
  - Kaspersky Internet Security for Android
- **HSE master student**

Agenda

Why cross-platform?

Proven solutions

Kotlin MP basics

App prototype

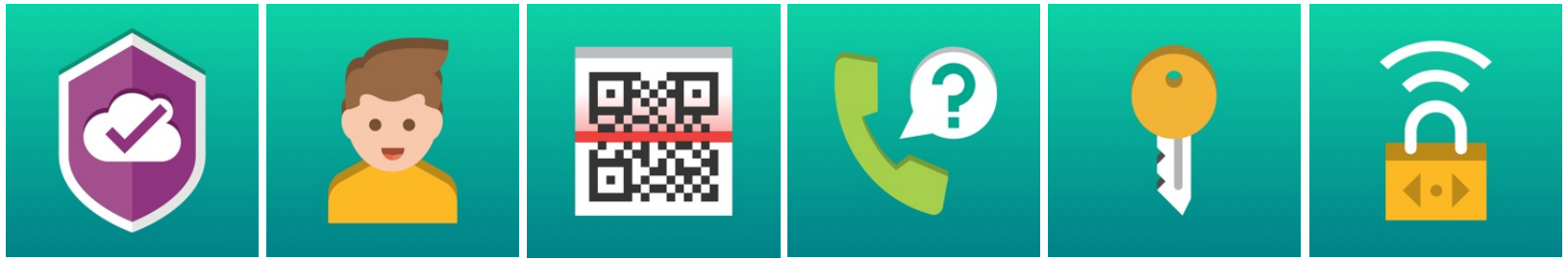
Useful links/examples

Summary

Why we need cross-platform?

- **Android and iOS developers write similar code for both platform**
- **Twice (or more) bigger team**

Kaspersky Mobile Apps



iOS

Xamarin, React Native, …

- WebView or Bridge for UI-components rendering
- Slow
- Mono garbage collection

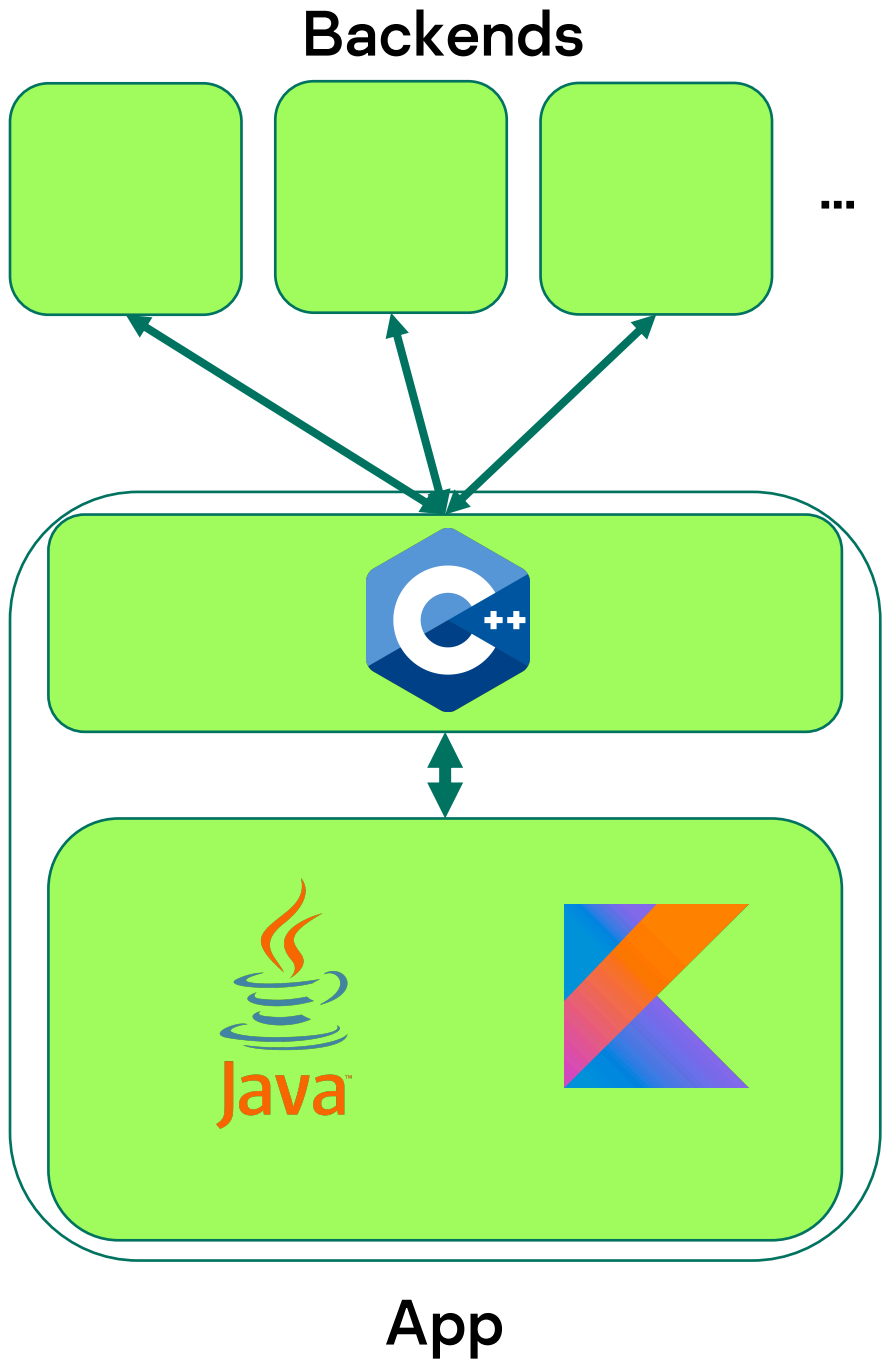
https://youtu.be/NjDC2DXh34c?t=4674 (RUS) Путешествие в кроссплатформенность с **Xamarin.Forms**

C++

Common transport library for
backends communication in Kaspersky apps

- Low-level
- Too complex to understand & maintain

- Mobile UI with Qt



kaspersky

My learning curve

On my pet projects

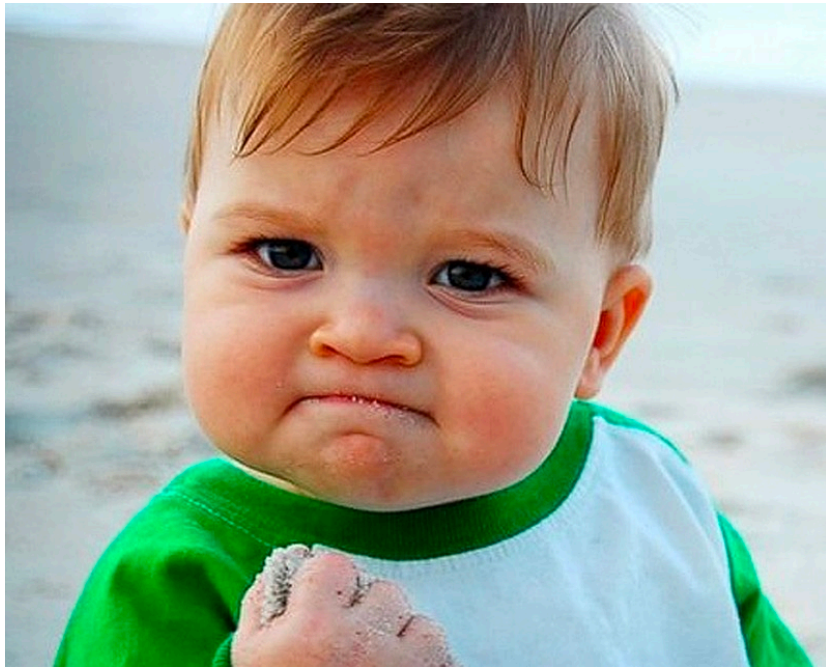
Me in 2018

Junior Android developer

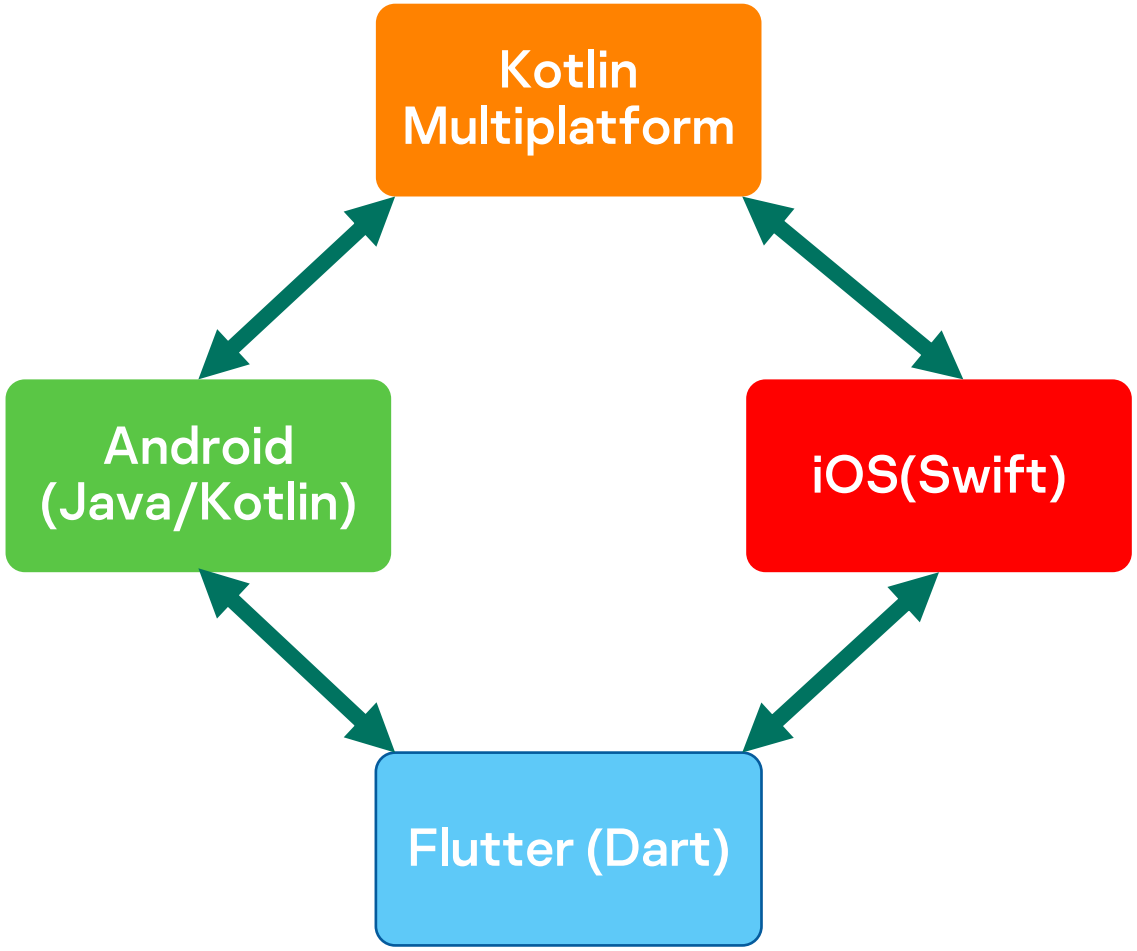
Java

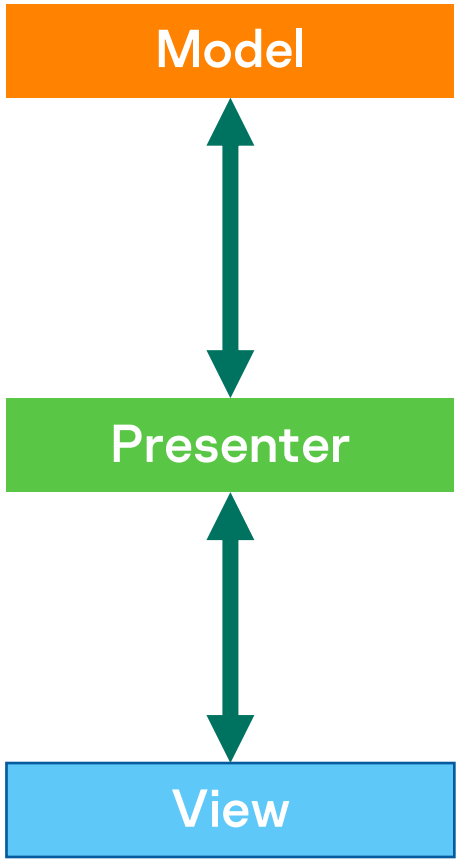
Gradle – a bit

Kotlin – not used in project in 2018



Article «Fast Prototypes with Flutter + Kotlin/Native»

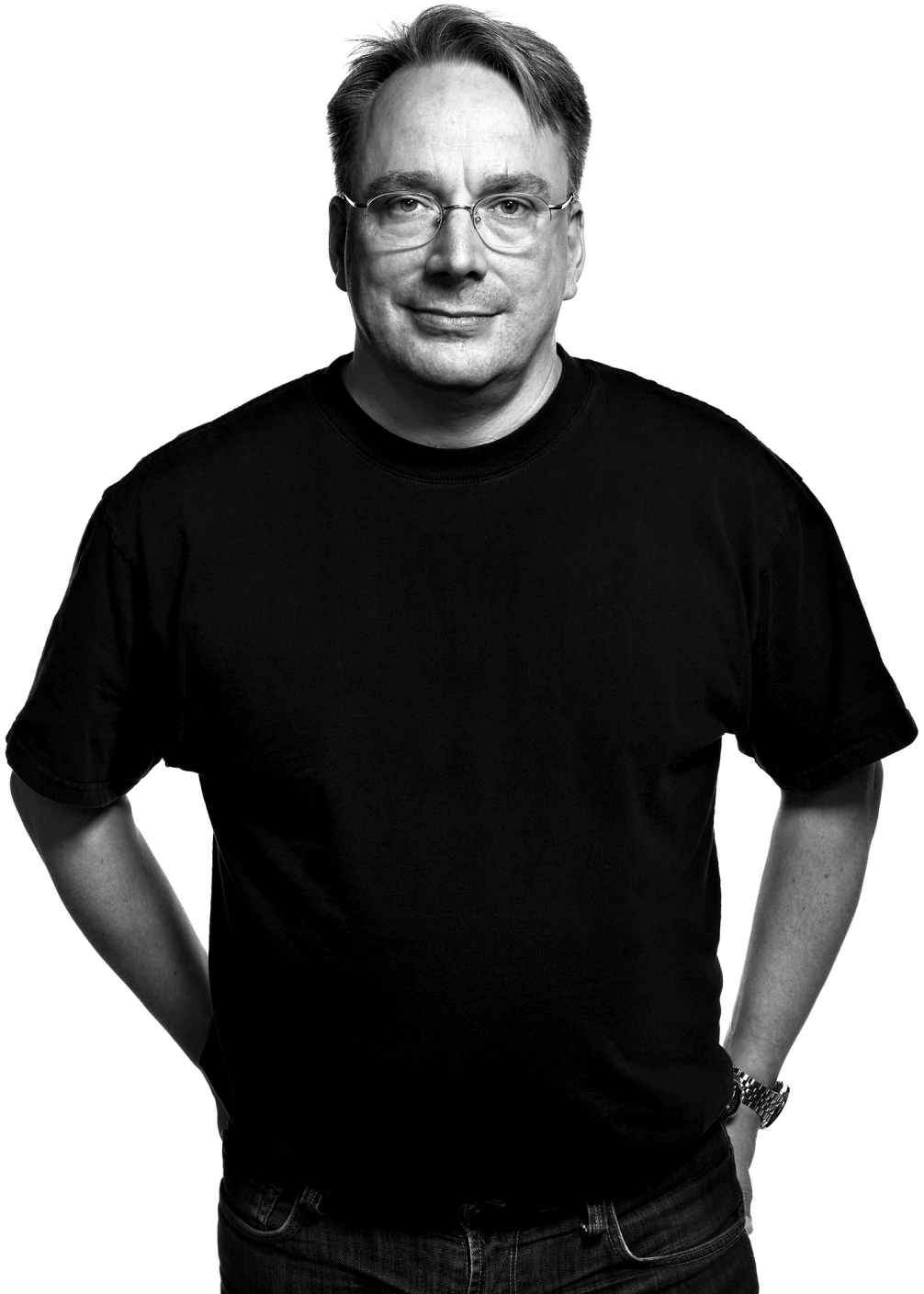


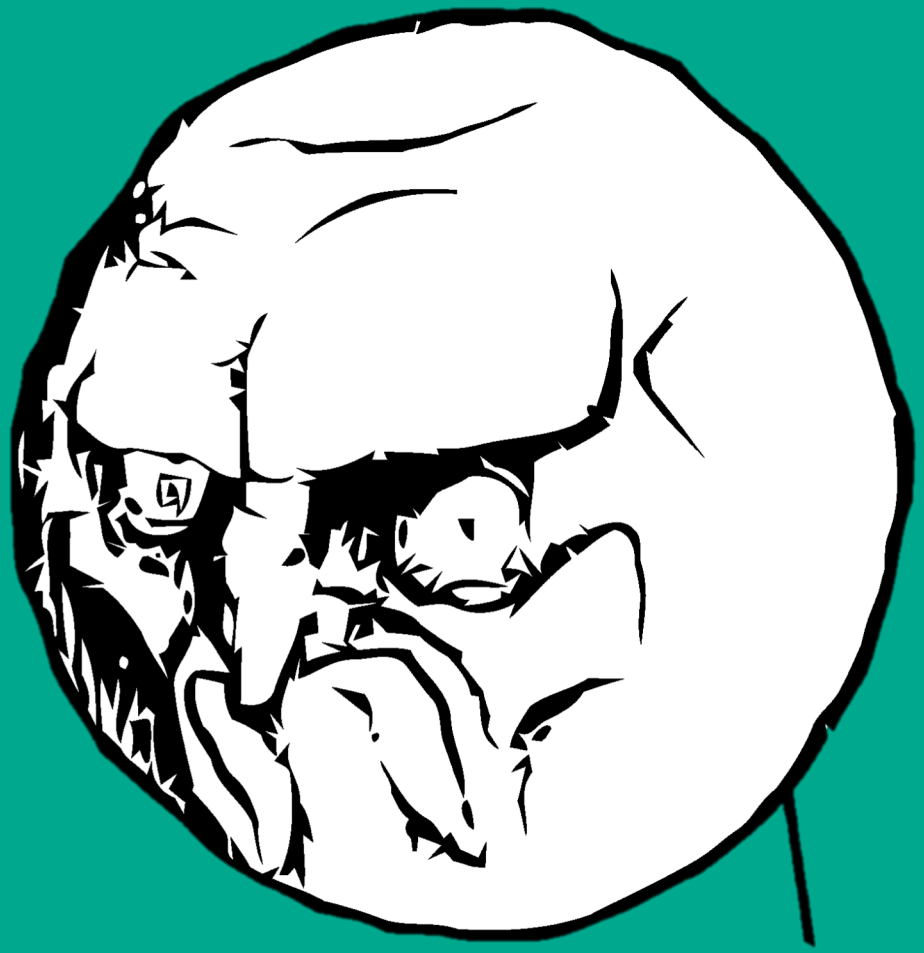


https://tech.olx.com/fast-prototypes-with-flutter-kotlin-native-d7ce5cfeb5f1

Sounds good

Show me the code



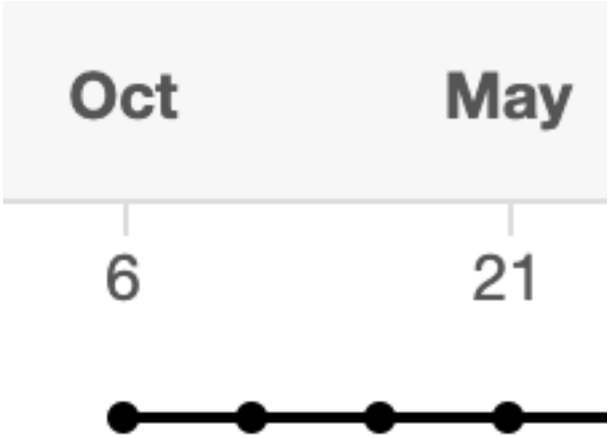


NO.


Ok


I’ll try
myself

October 2018



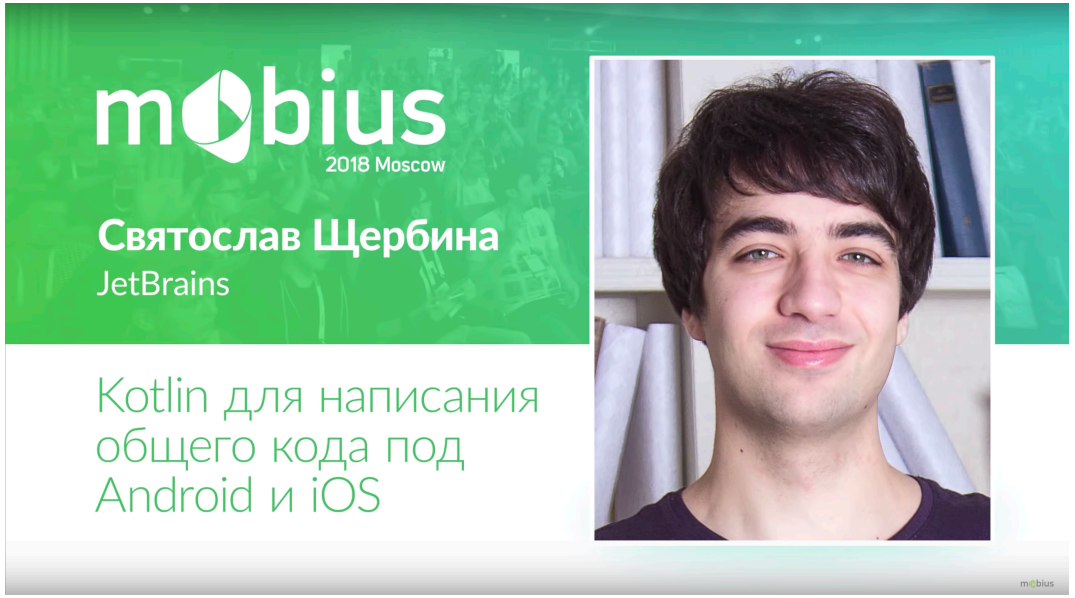
Kotlin MP official docs & example

Kotlin 1.3.0-rc-57

```
package org.kotlin.mpp.mobile

import platform.UIKit.UIDevice

actual fun platformName(): String {
    return UIDevice.currentDevice.systemName() + " " + UIDevice.currentDevice.systemVersion
}
```

Mobius Moscow 2018

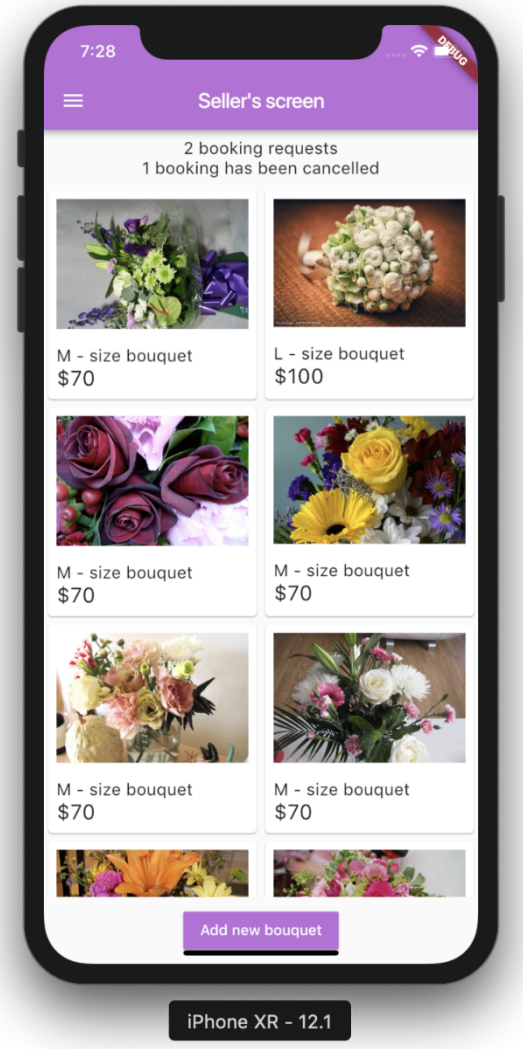
”Kotlin for common code in Android & iOS”



https://youtu.be/BDTgkXUgXPk

Flutter in parallel

https://github.com/AndreySBer/flutter_app_example

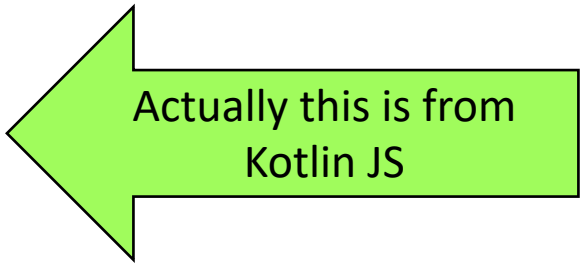


Kotlin Native

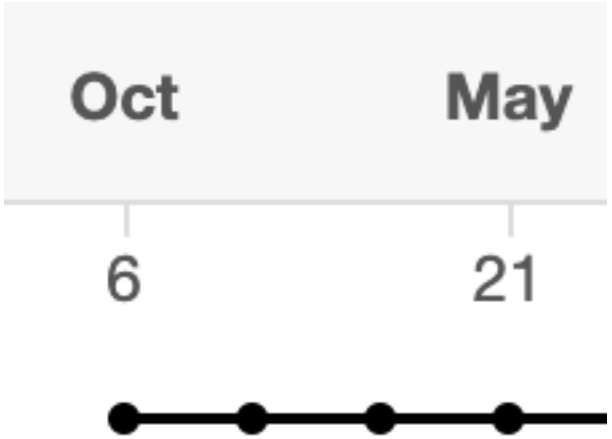
The abi versions don't match. Expected '[9]', found '14'
w: The compiler versions don't match either. Expected '[1.3.1]',
found '1.3.50-release-11850'

kotlin.IllegalStateException: There is no event loop.
Use runBlocking { ... } to start one.
at
0 SharedCode 0x000000010c44d367
kfun:kotlin.Throwable.<init>(kotlin.String?)kotlin.Thro
wable + 23
at...

ERROR in ../_imported/kotlinx-io/0.1.12/kotlinx-io.js
Module not found: Error: Can't resolve 'text-encoding' in
'/Users/beryukhov/SE/KotlinIosAndroid/build/js/packages_imp
orted/kotlinx-io/0.1.12'



git log



Try again

Kotlin 1.3.31

Combination of KMP libraries with same ABI (for a week)

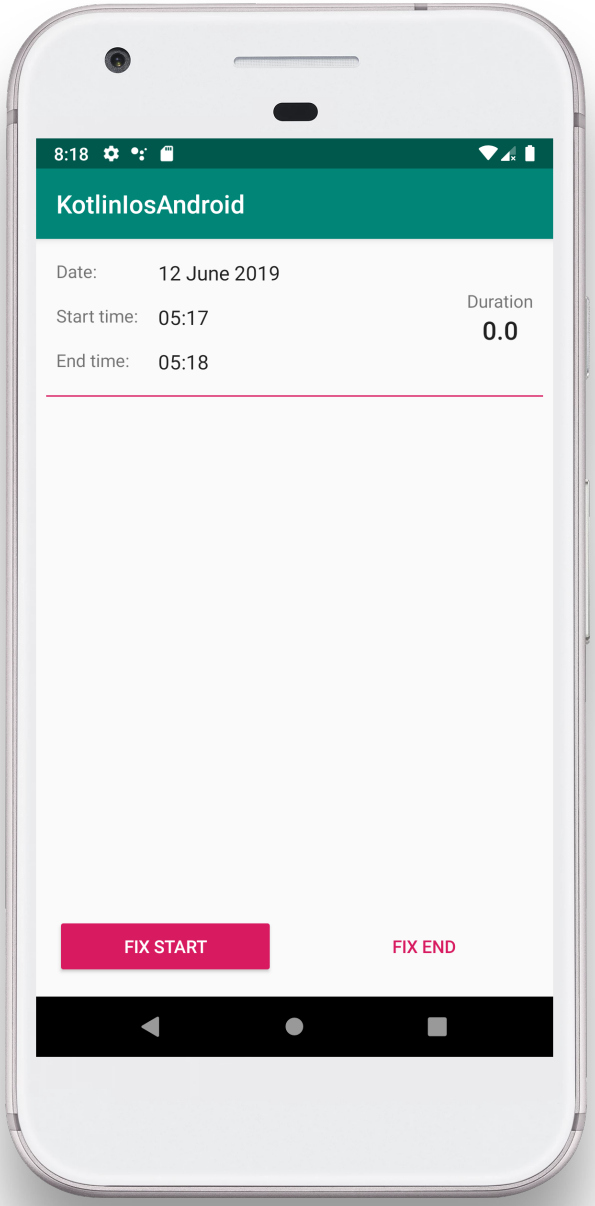
Simple Android layout

Simple MVP logic

Kotlin MP application prototype

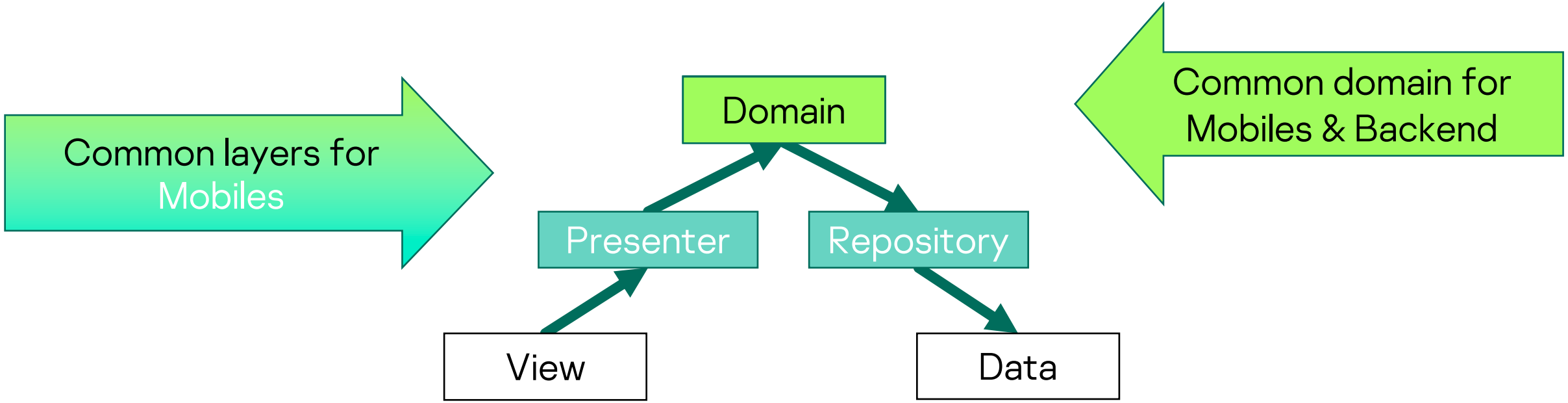
Client application for working time tracking

- 1 screen per each OS
  - Dates list;
  - Fix start, Fix end buttons
- iOS + Android + JS
- Simple Backend



https://github.com/AndreySBer/KotlinIosAndroid

MVP + Clean Architecture



iOS

Storyboards)

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

Readable headers

```
__attribute__((objc_subclassing_restricted))
__attribute__((swift_name("TimeSheetPresenter.Companion")))
@interface SharedCodeTimeSheetPresenterCompanion : KotlinBase
+ (instancetype)alloc __attribute__((unavailable));
+ (instancetype)allocWithZone:(struct _NSZone *)zone
__attribute__((unavailable));
+ (instancetype)companion __attribute__((swift_name("init()")));
- (double)getStartDate
__attribute__((swift_name("getStartDate()")));
@end;
```

Expect + Actual Mechanism Workaround 1

```
expect class TimeSheetPresenterKmp
(timeSheetView: TimeSheetView, timeSheetRepository: TimeSheetRepository) {
    fun onCreateView()
    fun onFixStart()
    fun onFixEnd()
    fun onItemClick(item: DateModel)
}
```

Expect + Actual Mechanism

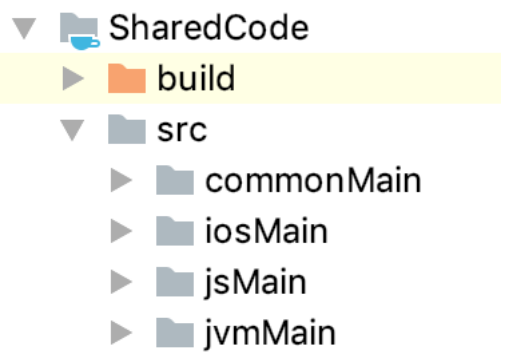

```
actual class TimeSheetPresenterKmp
        actual constructor(timeSheetView: TimeSheetView, timeSheetRepository: TimeSheetRepository) {

val timeSheetPresenter: TimeSheetPresenter

   init { timeSheetPresenter = TimeSheetPresenter(timeSheetView,
timeSheetRepository) }

   actual fun onCreateView() = timeSheetPresenter.onCreateView()
   actual fun onFixStart() = timeSheetPresenter.onFixStart()
   actual fun onFixEnd() = timeSheetPresenter.onFixEnd()
   actual fun onItemClick(item: DateModel) =
timeSheetPresenter.onItemClick(item)
}
```

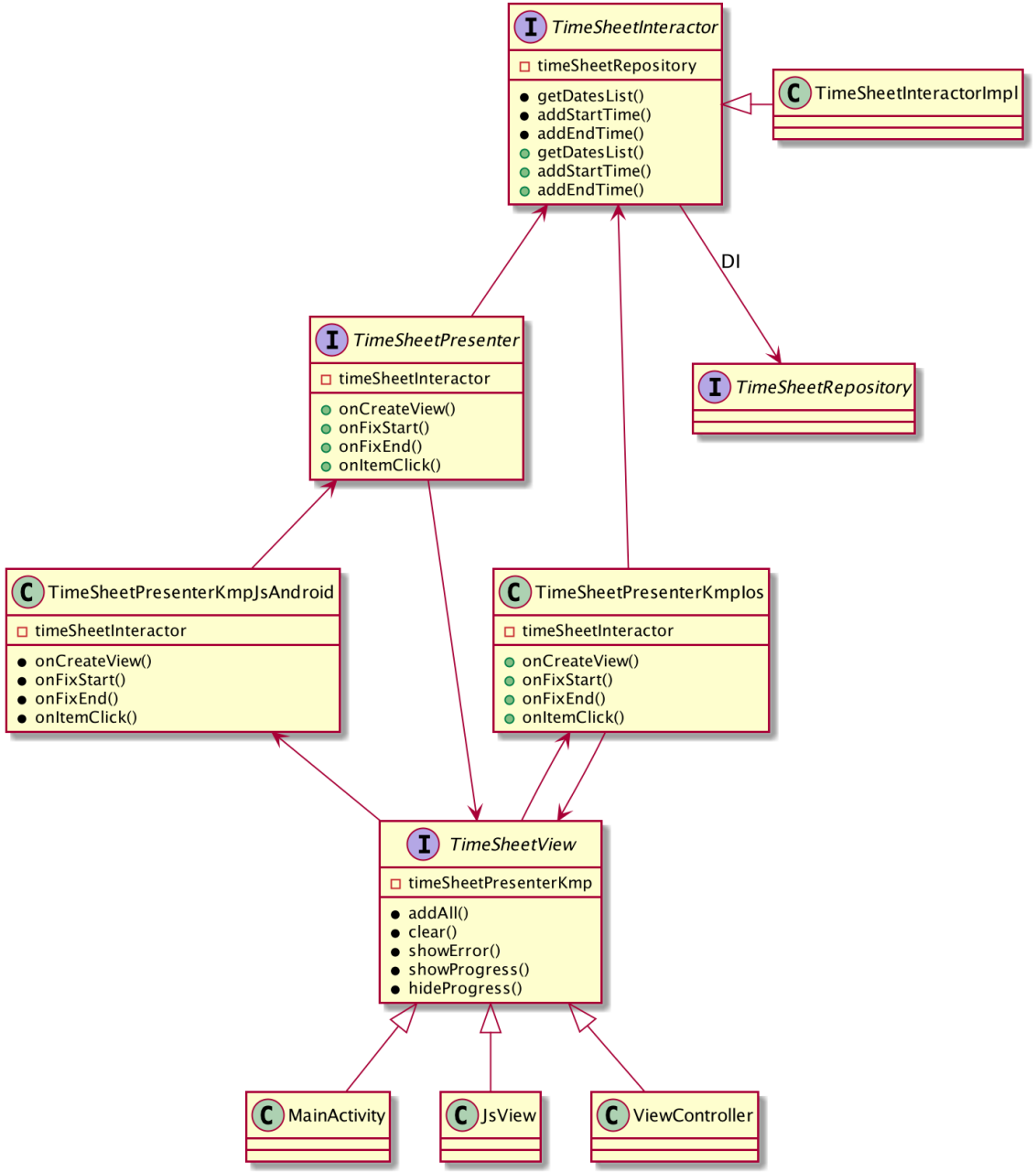
Expect + Actual Mechanism

iOS

```
actual class TimeSheetPresenterKmp
        actual constructor(timeSheetView: TimeSheetView, timeSheetRepository: TimeSheetRepository) {

    val timeSheetInteractor: TimeSheetInteractor
    val timeSheetView: TimeSheetView

    init {
        timeSheetInteractor = TimeSheetInteractorImpl(timeSheetRepository)
        this.timeSheetView = timeSheetView
    }

    actual fun onCreateView() {...}
    actual fun onFixStart() {...}
    actual fun onFixEnd() {...}
    actual fun onItemClick(item: DateModel) {...}
}
```

Workaround 1 Scheme





Try again

Kotlin 1.3.41

Combination of KMP libraries with same ABI for just half a day

Add JS client View

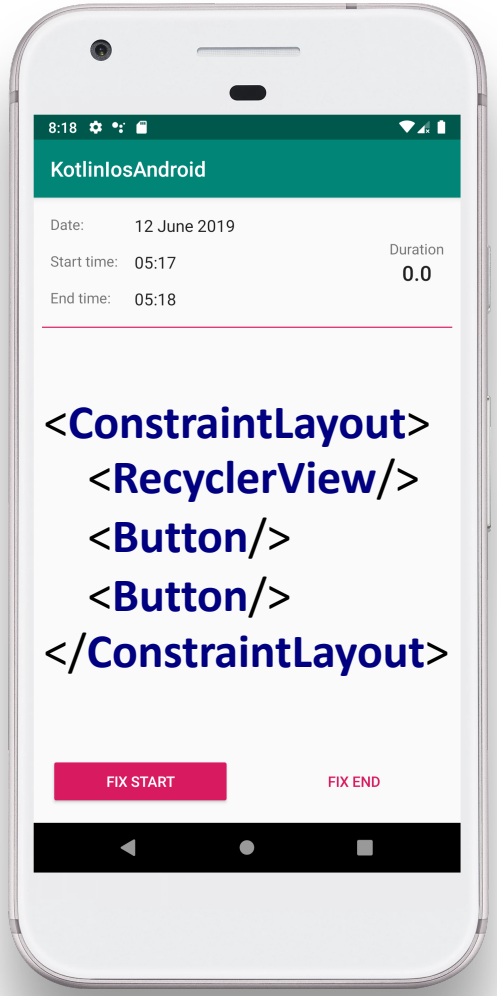
+ backend

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() {...}
  ...
}
```

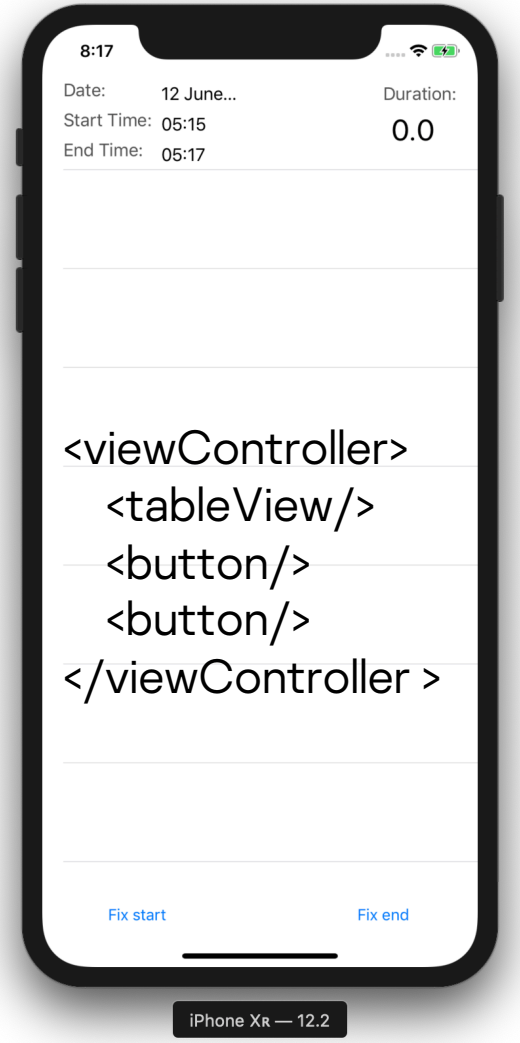


Jetpack Compose

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() {…}
    …
}
```

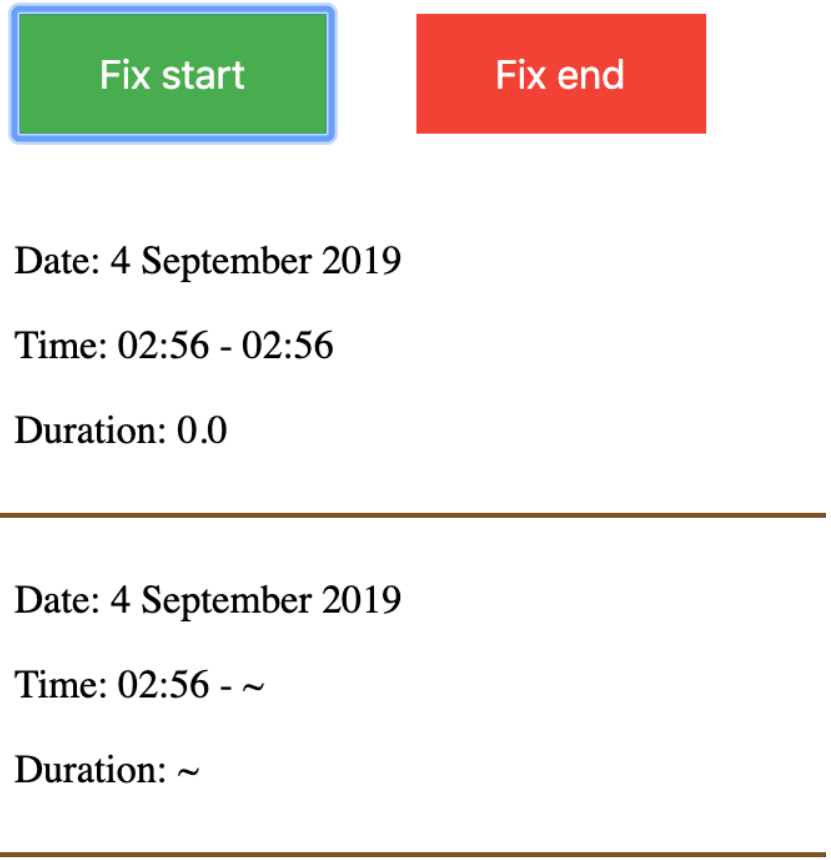
Swift UI



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



See also: https://github.com/Kotlin/kotlin-frontend-plugin
https://github.com/JetBrains/create-react-kotlin-app

Backend (JVM)

Netty Server

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

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

Expect + Actual Mechanism

Workaround 2

- E **expect val** *uiScope* : CoroutineScope
- E **expect val** *processScope* : CoroutineScope
- E **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() {...}
```

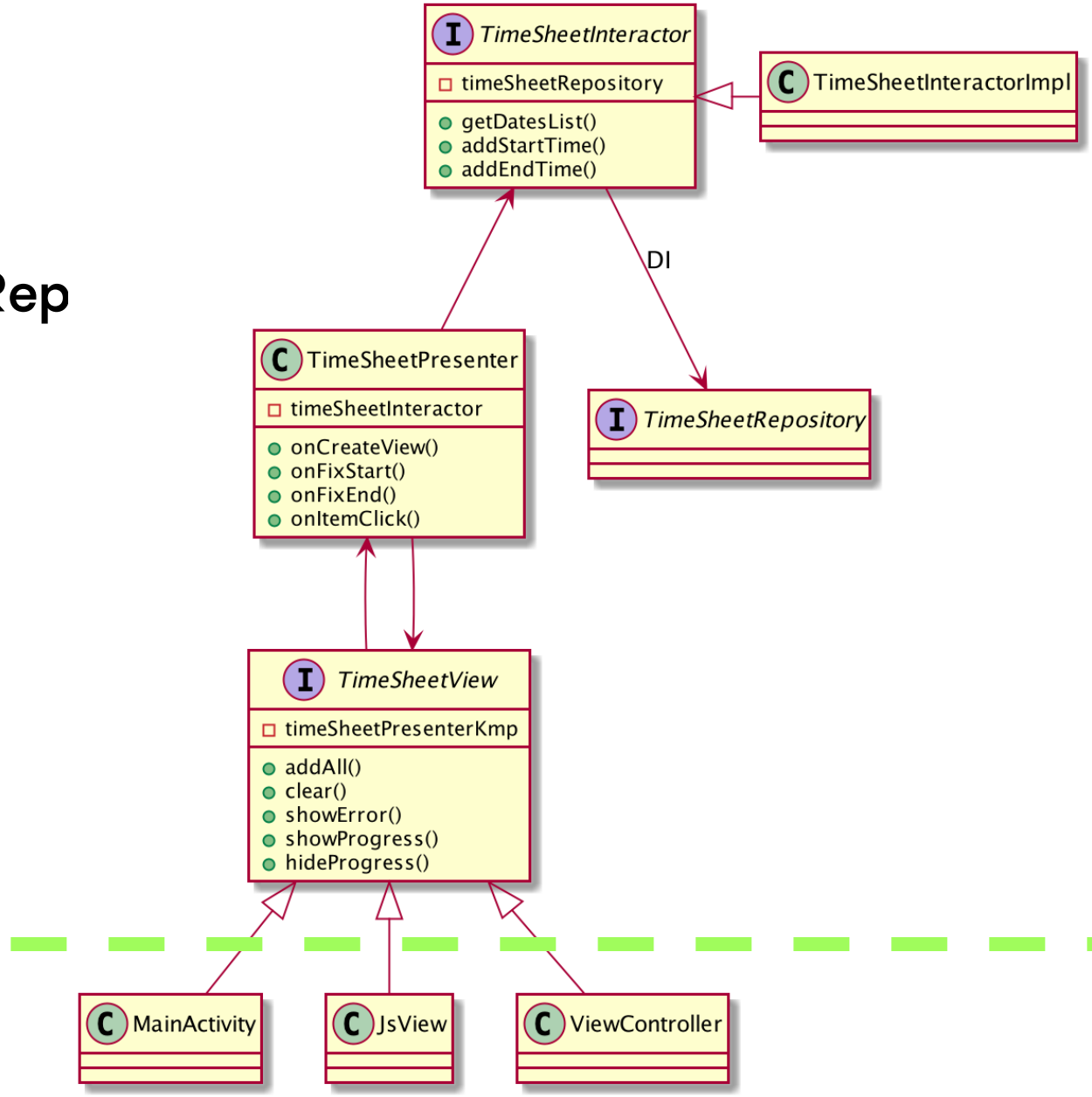
Common code

Add new client == Implement View + Rep

Native Views

Access to platform services

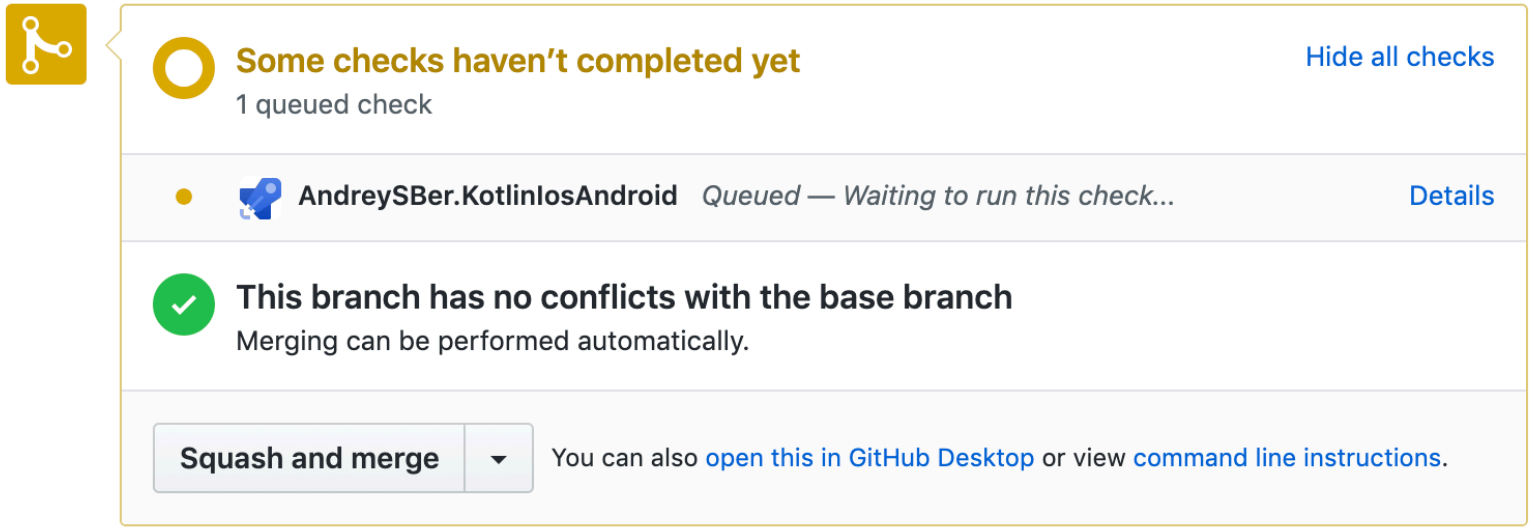
Business logic reuse



CI

https://dev.azure.com

tasks: 'app:assembleDebug packForXCode jsBrowserWebpack backend:jar'



Gradle + Artefacts

implementation project(':SharedCode') ~35KB

iOS

task packForXCode => SharedCode.framework ~6.8MB

jsBrowserWebpack => SharedCode.js ~12MB

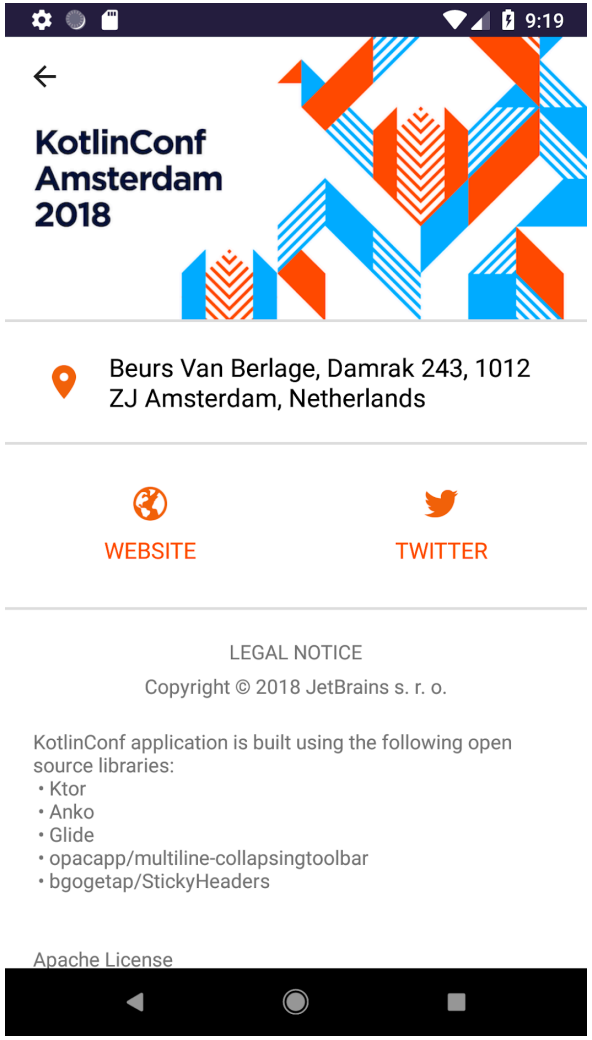
kaspersky

Useful links/examples

KotlinConf Schedule Application <-MVP

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

- Android
- iOS
- Backend



MPP with MVI sample

https://github.com/badoo/Reaktive

- Android
- iOS
- JS
- Linux X64

- Interesting lib for Rx

Ktor Samples of MPP

https://github.com/ktorio/ktor-samples

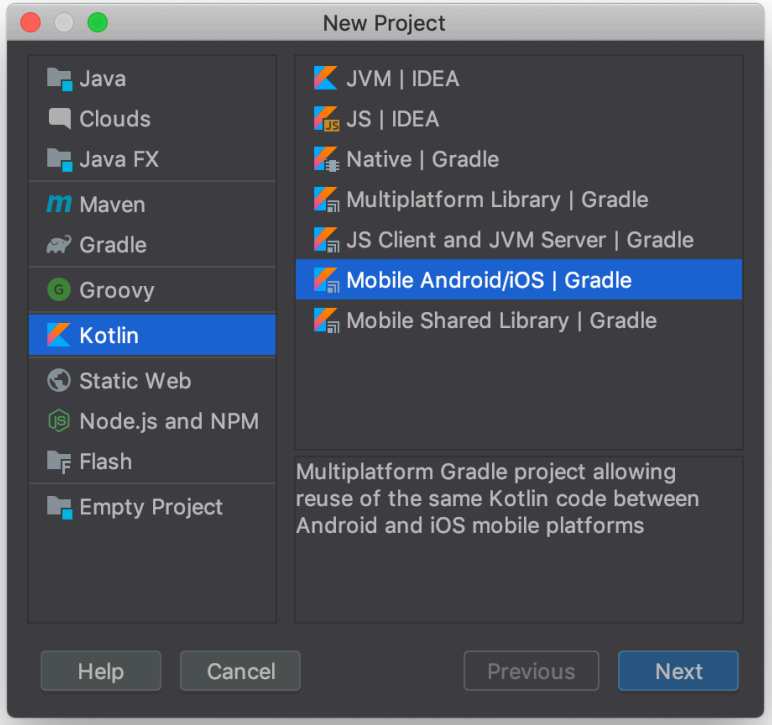
https://github.com/ktorio/ktor/tree/master/ktor-client

https://ktor.io/

IntelliJ IDEA project templates

Samples of MPP projects

Different in different versions of IDE



Kotlin Multiplatform

Kotlin compiler

Compiles into:
- JVM bytecode (Android, Backend),
- JavaScript,
- Native (iOS, ...)

Native UI

UI is made by native tools on each platform
+ system functions availability

Experimental

Set of multiplatform libraries
- Coroutines,
- Ktor,
- Serialization,...

Developers

Android developers

+ Backend developers (Java)

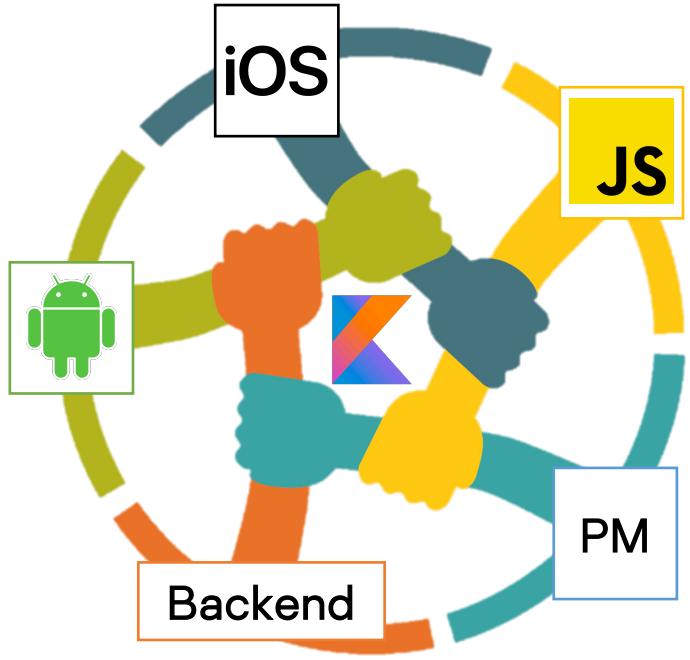
+ Full Java interoperability

44

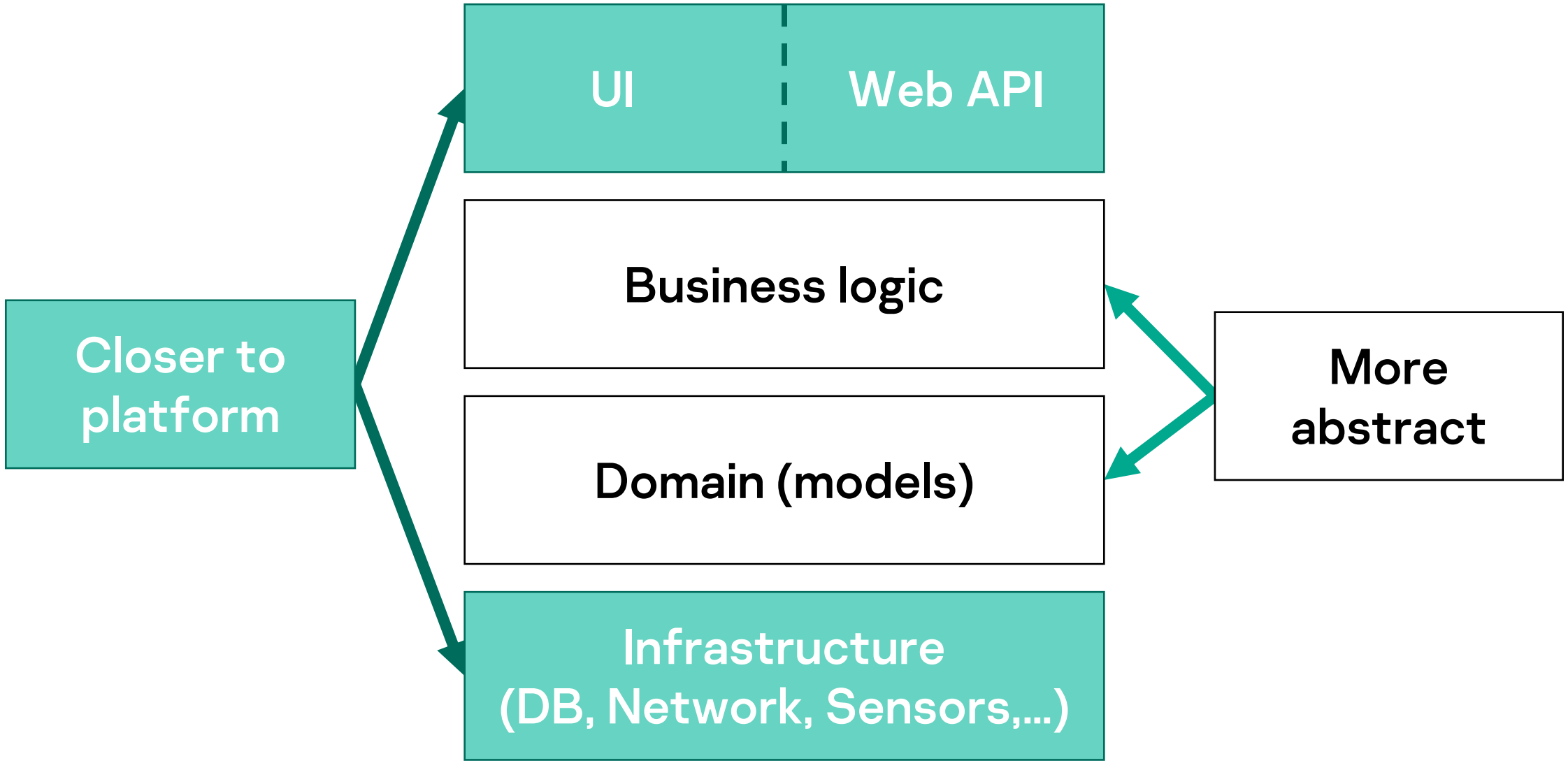
Comparison to Native Approach

Native layers: UI and Repository => Full access to platform features

- **Android/Backend => All the same**
- **Web => Developers seem happy**
- **iOS => Another world + Pain**



High-level architecture of applications



Comparison to Flutter

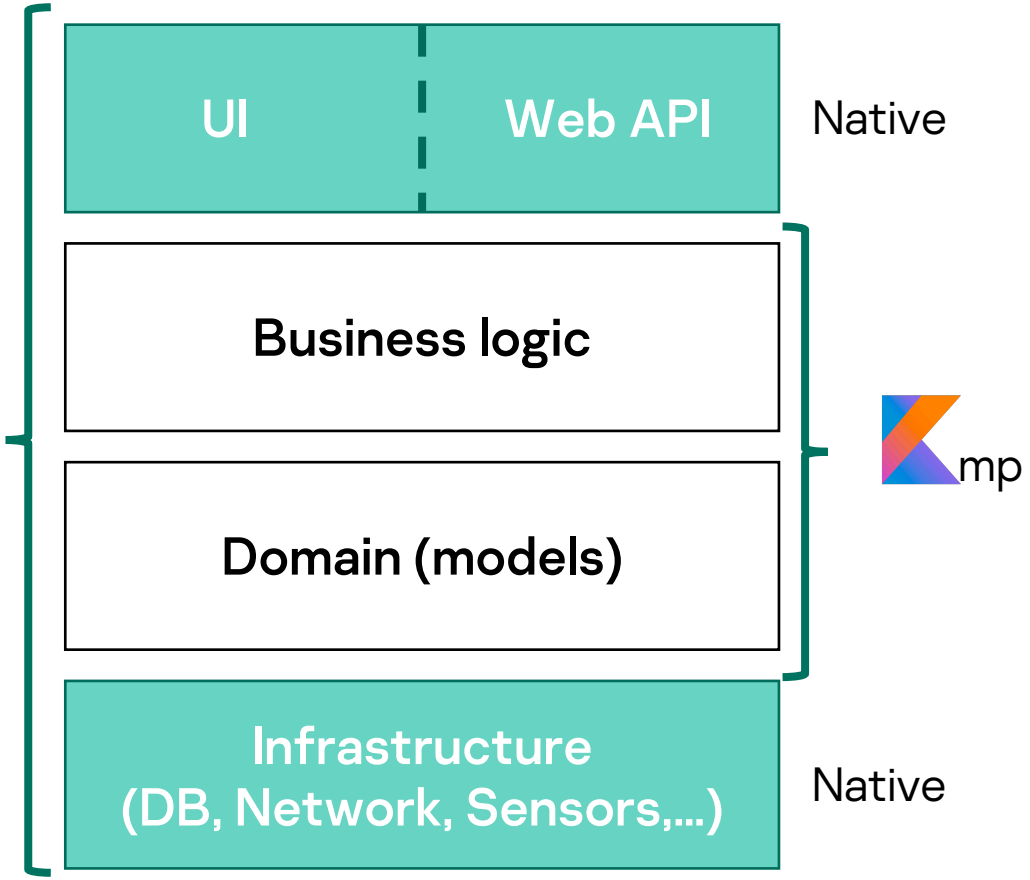
UI Widgets library

No minSdk limit

Why Dart; not Kotlin?

```
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")

            } } } }
```

Plugins & Pain



https://hackernoon.com/why-flutter-uses-dart-dd635a054ebf
https://blog.kotlin-academy.com/flutter-and-kotlin-multiplatform-relationship-890616005f57

kaspersky

Thank you!

Andrey Beryukhov

beryukhov.ru
@Phansier

kaspersky

Kaspresso

Android UiTest framework based
on Espresso, UIAutomator and Kakao

https://github.com/KasperskyLab/Kaspresso
https://youtu.be/cTykctRSmuA