iOS Architecture with Multiplatform

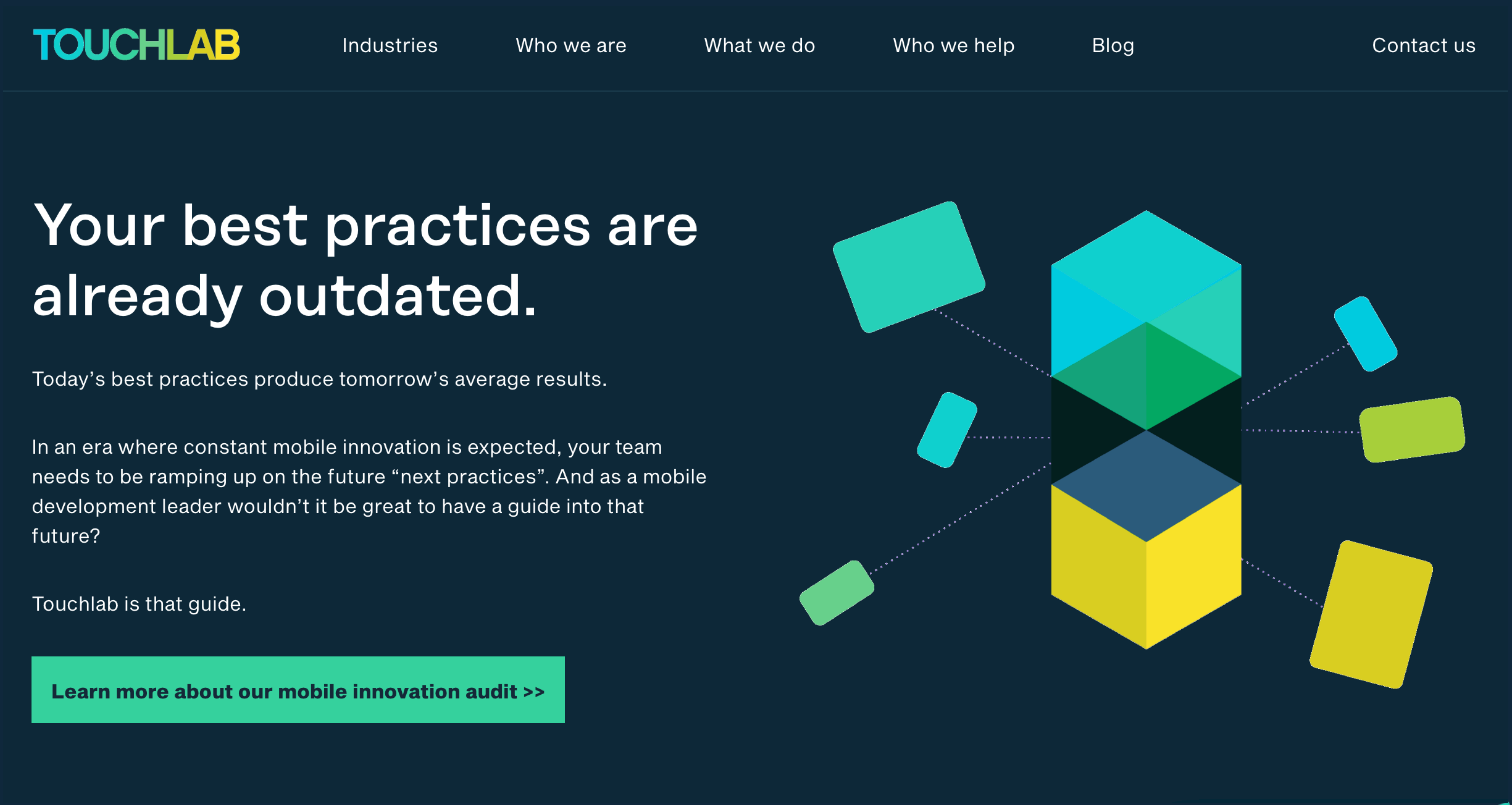
Kevin Galligan

TOUCHLAB

Kevin Galligan

President of Touchlab

President of Touchlab. Have been coding Android since before the G1. We run the big Android meetup in NYC, and Droidcon NYC. Currently obsessed with platform convergence topics.



Your best practices are already outdated.

Today’s best practices produce tomorrow’s average results.

In an era where constant mobile innovation is expected, your team needs to be ramping up on the future “next practices”. And as a mobile development leader wouldn’t it be great to have a guide into that future?

Touchlab is that guide.

community!

2016

doppl

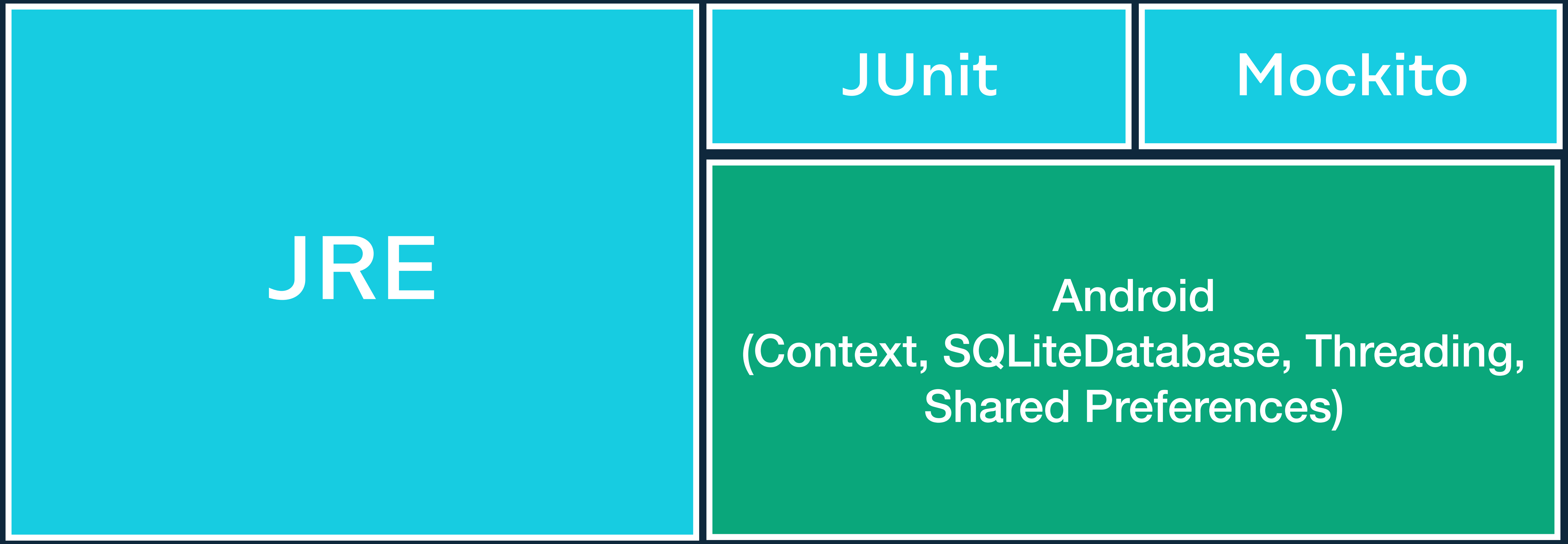
J2ObjC

JRE

JUnit

Mockito

J2ObjC



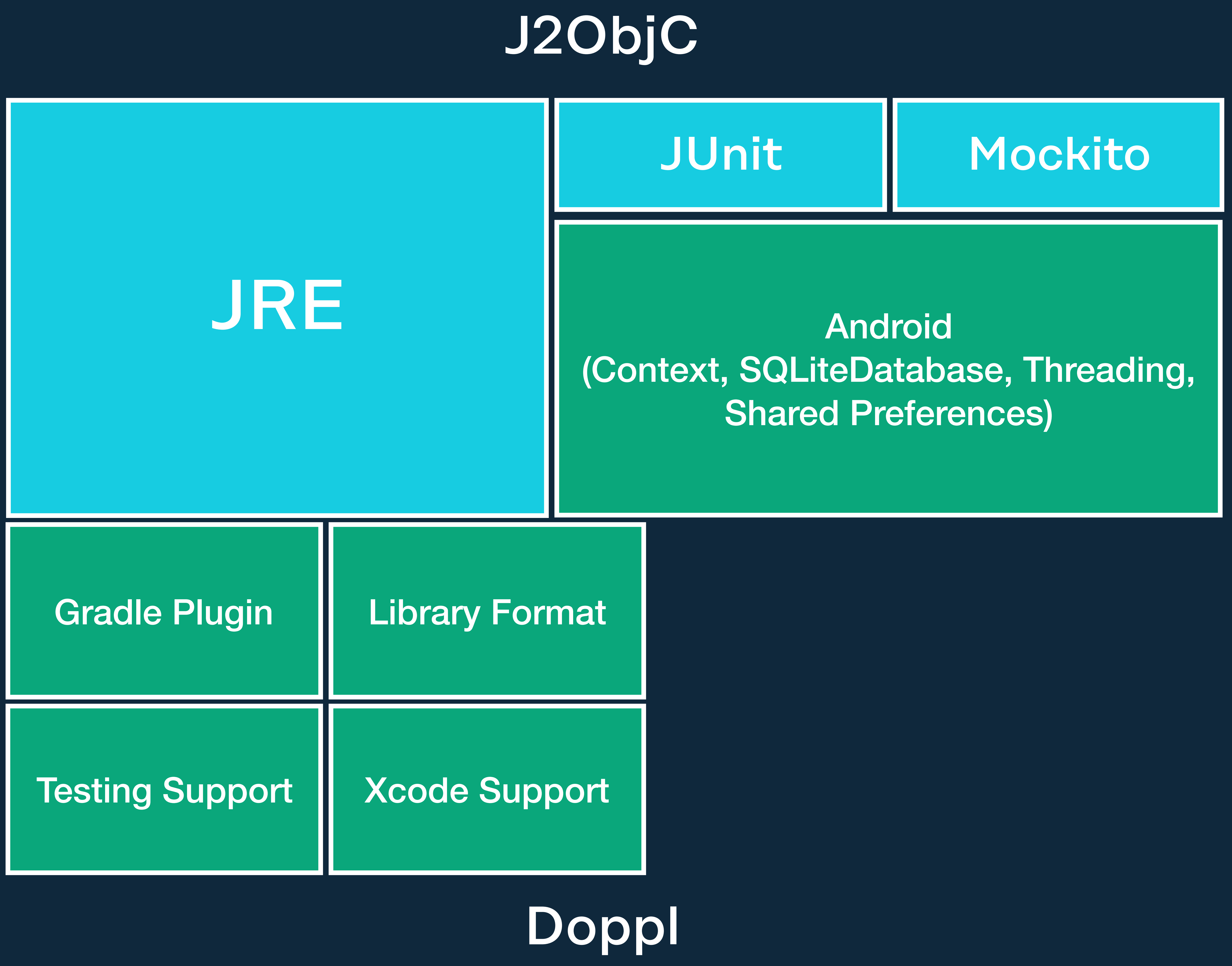
JRE

JUnit

Mockito

Android
(Context, SQLiteDatabase, Threading,
Shared Preferences)

Doppl



J2ObjC

JRE

JUnit

Mockito

Android
(Context, SQLiteDatabase, Threading,
Shared Preferences)

Gradle Plugin

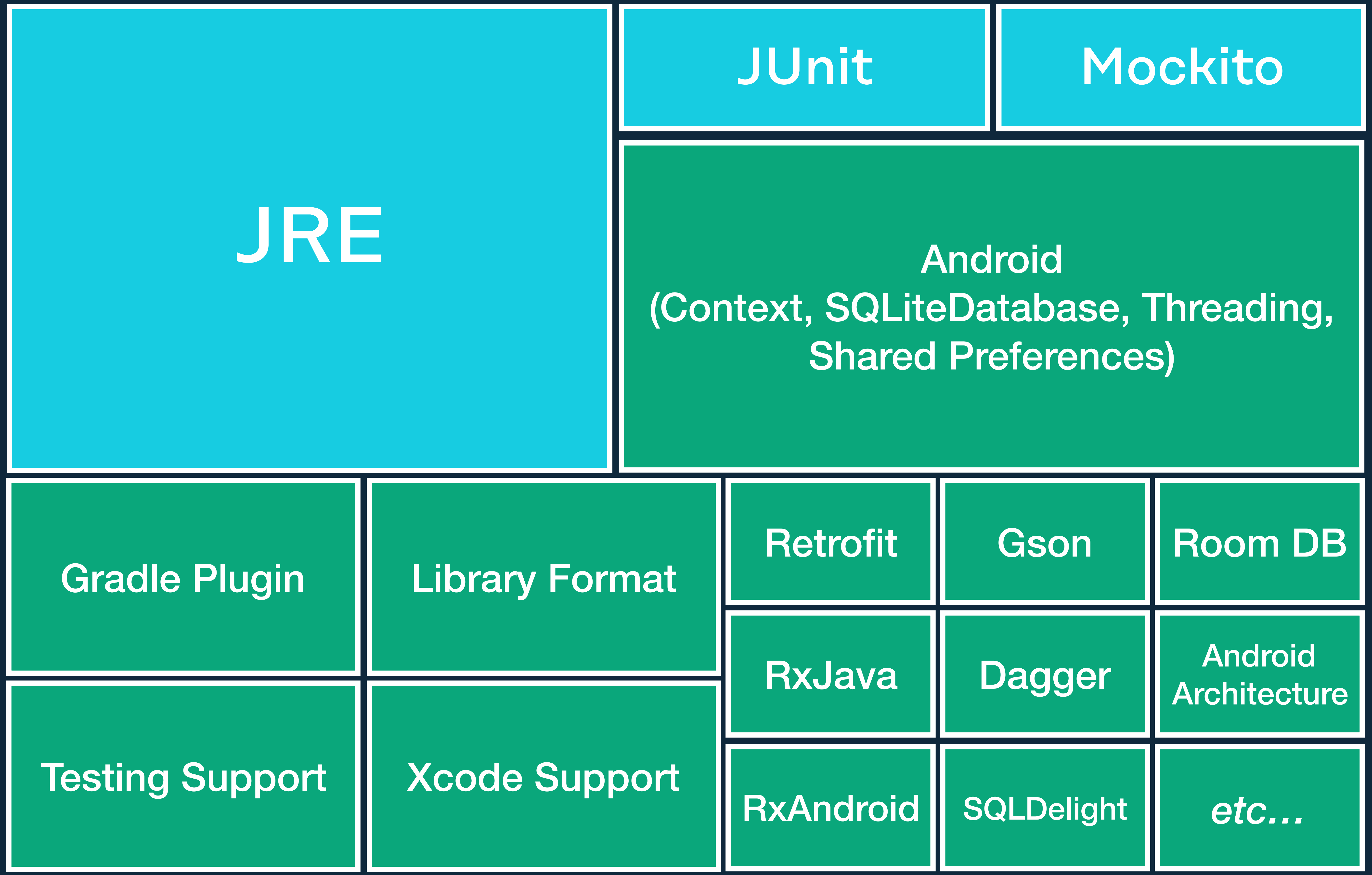
Library Format

Testing Support

Xcode Support

Doppl

J2ObjC



JRE

JUnit

Mockito

Android
(Context, SQLiteDatabase, Threading,
Shared Preferences)

Gradle Plugin

Library Format

Retrofit

Gson

Room DB

RxJava

Dagger

Android Architecture

Testing Support

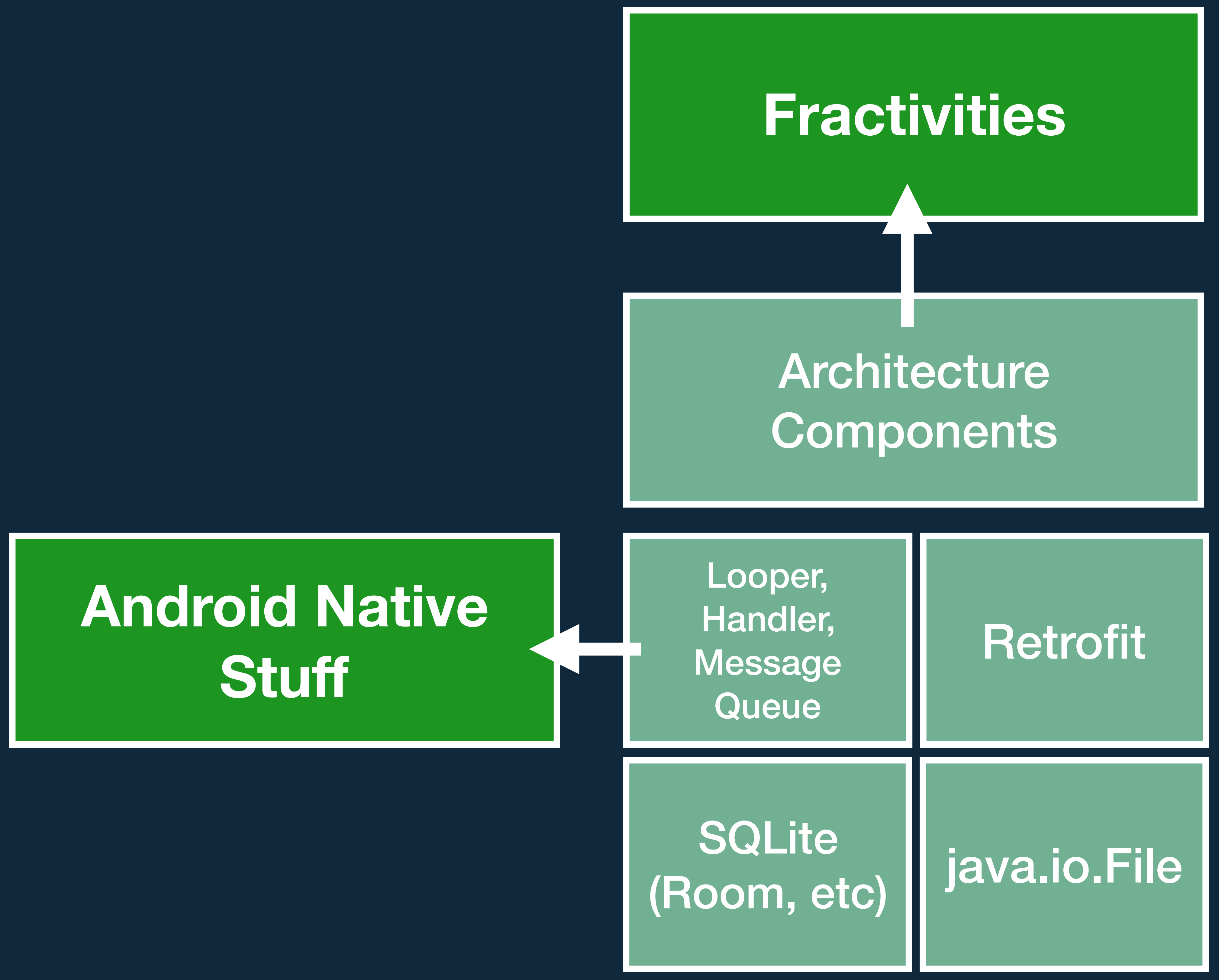
Xcode Support

RxAndroid

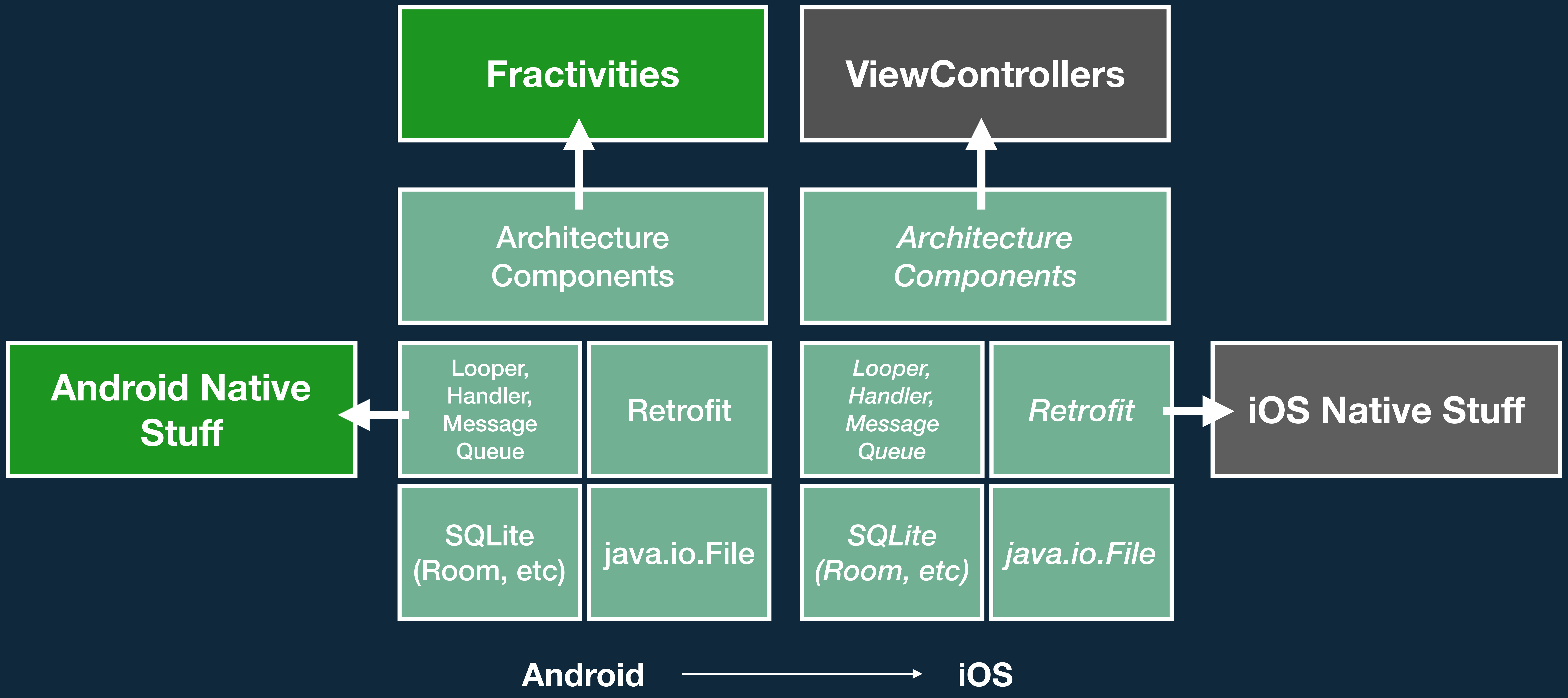
SQLDelight

etc...

Doppl


Fractivities
Architecture Components
Android Native Stuff
Looper, Handler, Message Queue
Retrofit
SQLite (Room, etc)
java.io.File

Android



Fractivities

ViewControllers

Architecture Components

Architecture Components

Android Native Stuff

Looper, Handler, Message Queue

Retrofit

Looper, Handler, Message Queue

Retrofit

iOS Native Stuff

SQLite (Room, etc)

java.io.File

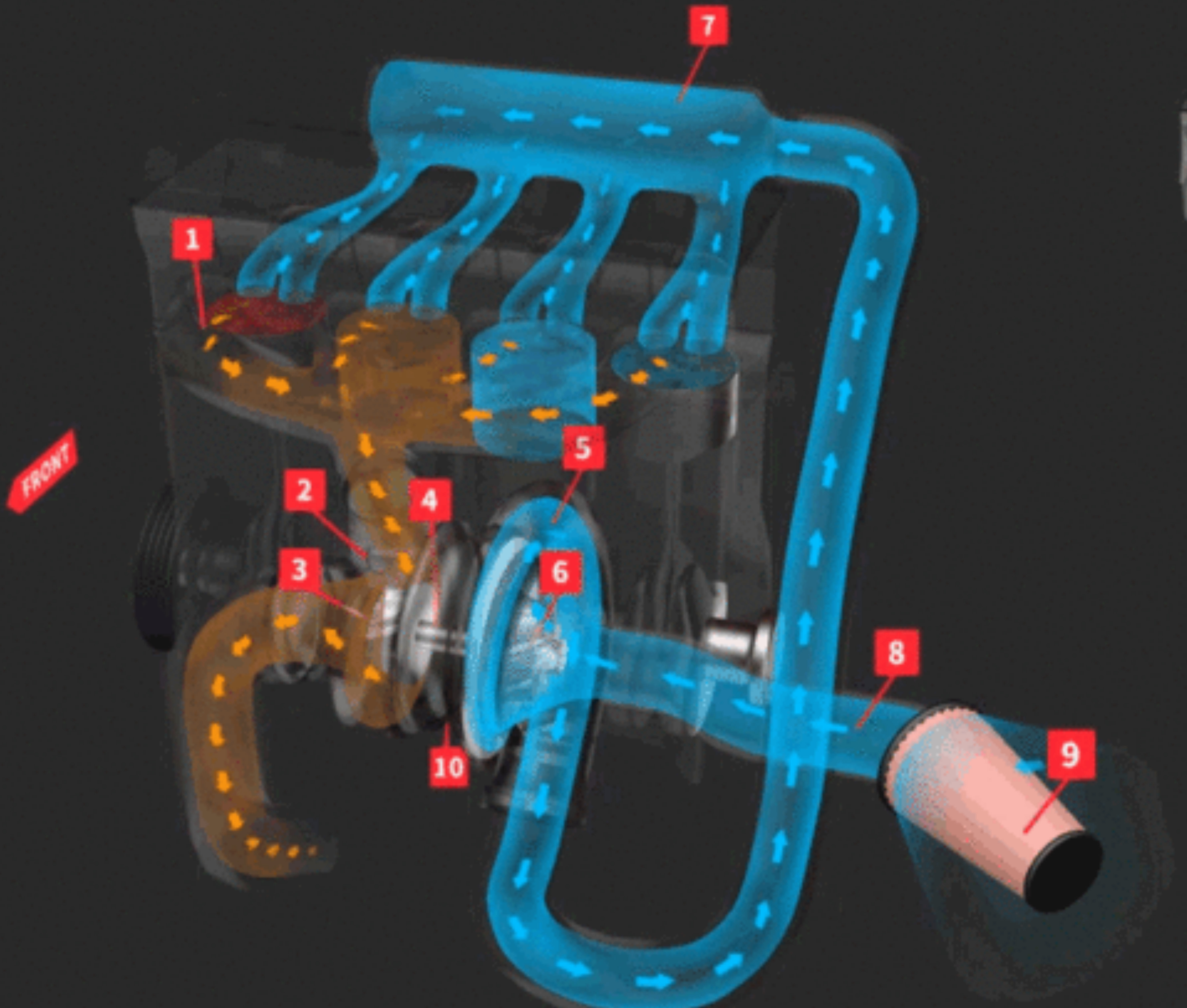
SQLite (Room, etc)

java.io.File

Android → iOS

no thanks

Turbo



https://medium.com/@kpgalligan/the-future-of-shared-code-is-kotlin-multiplatform-9aac94517f95

M
PHSM
I approve!

Andrey Breslav
@abreslav

Following

I tend to trust people who have no trouble admitting they are wrong (and are fixing their mistakes). Acknowledging that someone else is right gains them extra points.

7:12 AM - 25 Sep 2018

3 Retweets 8 Likes

3
8

Replying to @abreslav

Was just baking a bit about this into a talk I'm working on...

GIF
Reply

evin Galligan
kpgalligan

ollowing 03
Followers 1,875

you · Change

erRegistrationDay
788.683 to register to

HBO

eHitWonderDay

ughts

close to buying Versace in $2
rts say

ivation

Vho to follow · Refresh · Vie

Scott Klein @kleinm

Brian Boyer @brian

erin mccann | subscr

Find people you know

2018 Twitter About Help Center
rivacy policy Cookies Ads info Br
log Status Apps Jobs Marketing
usinesses Developers

Advertise with Twitter

still no thanks

swift is life!!!

the future?

now

reach out for repos

chief hacking officer

kotlin

SHARED ARHICTECTURE

Mobile & Web

Architecture, not UI

Write once, Run Anywhere

Java Source
Software.java
Java Compiler
javac.exe
Java Code
Software.class
JVM for Windows
Windows OS
JVM for Linux
Linux OS
KVM for Palm
Palm OS

Shared UI == Failure!

Write once, Run Anywhere

Java Source

Software.java

Java Compiler

javac.exe

Java Code

Software.class

JVM for Windows

Windows OS

JVM for Linux

Linux OS

KVM for Palm

Palm OS

Shared Loigc == Computers

kotlin-native-latest [~/temp4/kotlin-native-latest] - .../runtime/src/main/kotlin/kotlin/native/concurrent/Worker.kt [runtime]
kotlin-native-latest
runtime
src
main
kotlin
kotlin
native
concurrent
Worker.kt
Add Configuration...
Project
1: Project
Z: Structure
Annotations.kt
gradle-wrapper.properties
Worker.cpp
Atomics.kt
Freezing.kt
Internal.kt
Worker.kt
Future.kt
posix.kt
llvmDebugInfoC
performance
platformLibs
runtime
build
src
launcher
main
cpp
js
kotlin
kotlin
annotation
collections
comparisons
coroutines
internal
io
jvm
math
native
concurrent
Atomics.kt
Freezing.kt
Future.kt
Internal.kt
Lazy.kt
Lock.kt
ObjectTransfer.kt
Worker
internal
ref
Annotations.kt
BitSet
Blob.kt
Compatibility.kt
Runtime.kt
Text.kt
TypedArrays.kt
random
ranges
reflect
sequences
system
IJ
IntelliJ IDEA
ULTIMATE 2018.2
JET BRAINS

```
 * object graph belongs to one worker at the time, but can be disconnected and reconnected as needed.
 * See 'Object Transfer Basics' and [TransferMode] for more details on how objects shall be transferred.
 * This approach ensures that no concurrent access happens to same object, while data may flow between
 * workers as needed.
 */

/**
```

IntelliJ IDEA 2018.2.1 (Ultimate Edition)

Build #IU-182.3911.36, built on August 6, 2018

Licensed to Kevin Galligan

Subscription is active until December 14, 2018

JRE: 1.8.0_152-release-1248-b8 x86_64
JVM: OpenJDK 64-Bit Server VM by JetBrains s.r.o

Powered by open-source software

THE DRIVE TO DEVELOP

Copyright © 2000–2018 JetBrains s.r.o.

```
execute` interface.
re, for IO it may be
s processed,
n:
 refers to
ing added to jobs queue
 that whole state is
ecuted by the worker,
ph. Whoever will consume
bda job: (T1) -> T2): Future<T2> =
        /*
         * This function is a magical operation, handled by lowering in the compiler, and replaced with call to
         *   executeImpl(worker, mode, producer, job)
         * but first ensuring that `job` parameter  doesn't capture any state.
         */
        throw RuntimeException("Shall not be called directly")

    override public fun equals(other: Any?): Boolean = (other is Worker) && (id == other.id)

    override public fun hashCode(): Int = id

    override public fun toString(): String = "worker $id"
```

Web is more difficult

No SQL :(

Advocate for new standards

AKA The Long Game

PEPSI
E IN MEXICO · HECHO EN MÉXICO
L IMPORT · IMPORTACIÓN ESPECIAL
12 FL OZ
(355 ml)
Coca-Cola
MR

mobile is MUCH simpler

DROIDCON

WITH

KOTLIN MULTIPLATFORM

Search or jump to...
Pull requests
Issues
Marketplace
Explore
Unwatch
13
Unstar
Code
Issues 7
Pull requests 0
Projects 1
Wiki
Insights
Settings
Branch: master
New pull request
Upload files
Find file
Schedule

touchlab / DroidconKotlin

No description, website, or topics provided.

Manage topics

76 commits | 3 branches | 0 releases | 2 contributors

kpgalligan Update README — Latest comm

gradle/wrapper	Import	
iosApp	Update README	
libs	Import	3 months ago
sessionize	Added atomic.fu native implementation	6 days ago
[illegible]	Import	[illegible] ago
.gitignore	Update to native 0.9.1	6 days ago
LICENSE.txt	Readme update	2 months ago
README.md	Update README	5 days ago
build.gradle	Added atomic.fu native implementation	6 days ago

7:58

Droidcon NYC

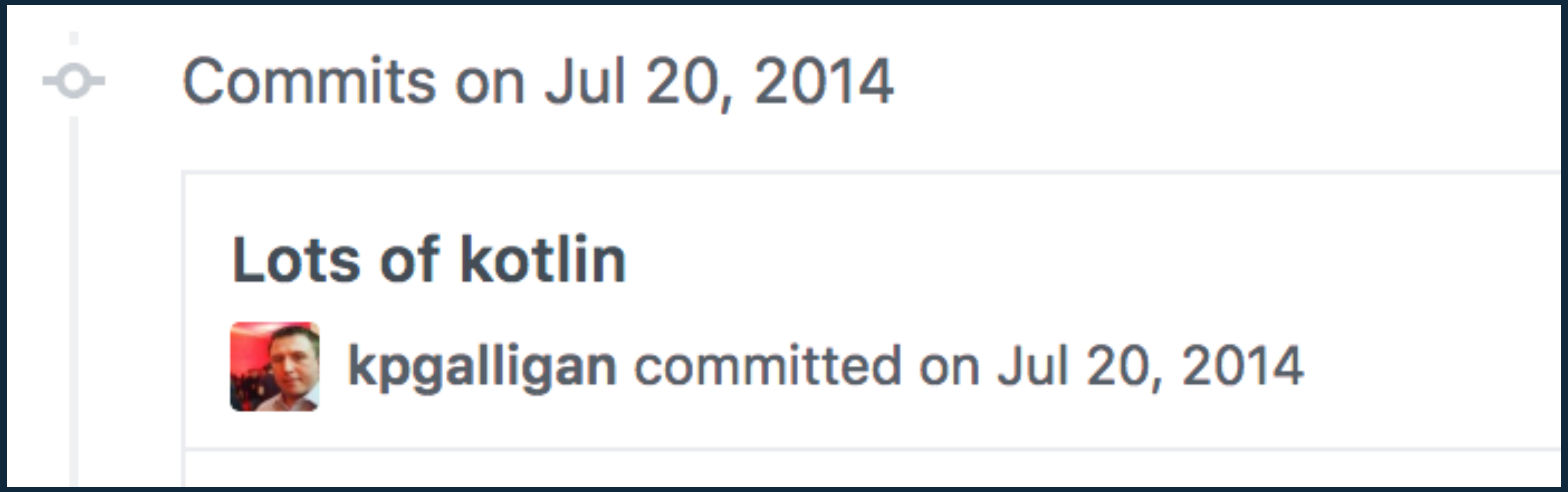
AUG 27 | AUG 28

9:00AM	Keynote: The future of our community is YOU Dan Kim
9:50AM	Digging into D8 and R8 Jake Wharton
	Scaling Instagram's Launcher Activity Rahul Dandamudi
●	Don't Sweat the Small Stuff Zarah Dominguez
	Why Jetpack? Kurt Nelson
10:40AM	Adventures in Navigation Lyla Fujiwara, Daniel Galpin
●	Bringing your app to life - Motion Animations everywhere Marcos Paulo Damasceno
	Building a Multiplatform Kotlin Library Russell Wolf

Funky Code Testbed

Funky Code Testbed

Kotlin in 2014!



Commits on Jul 20, 2014

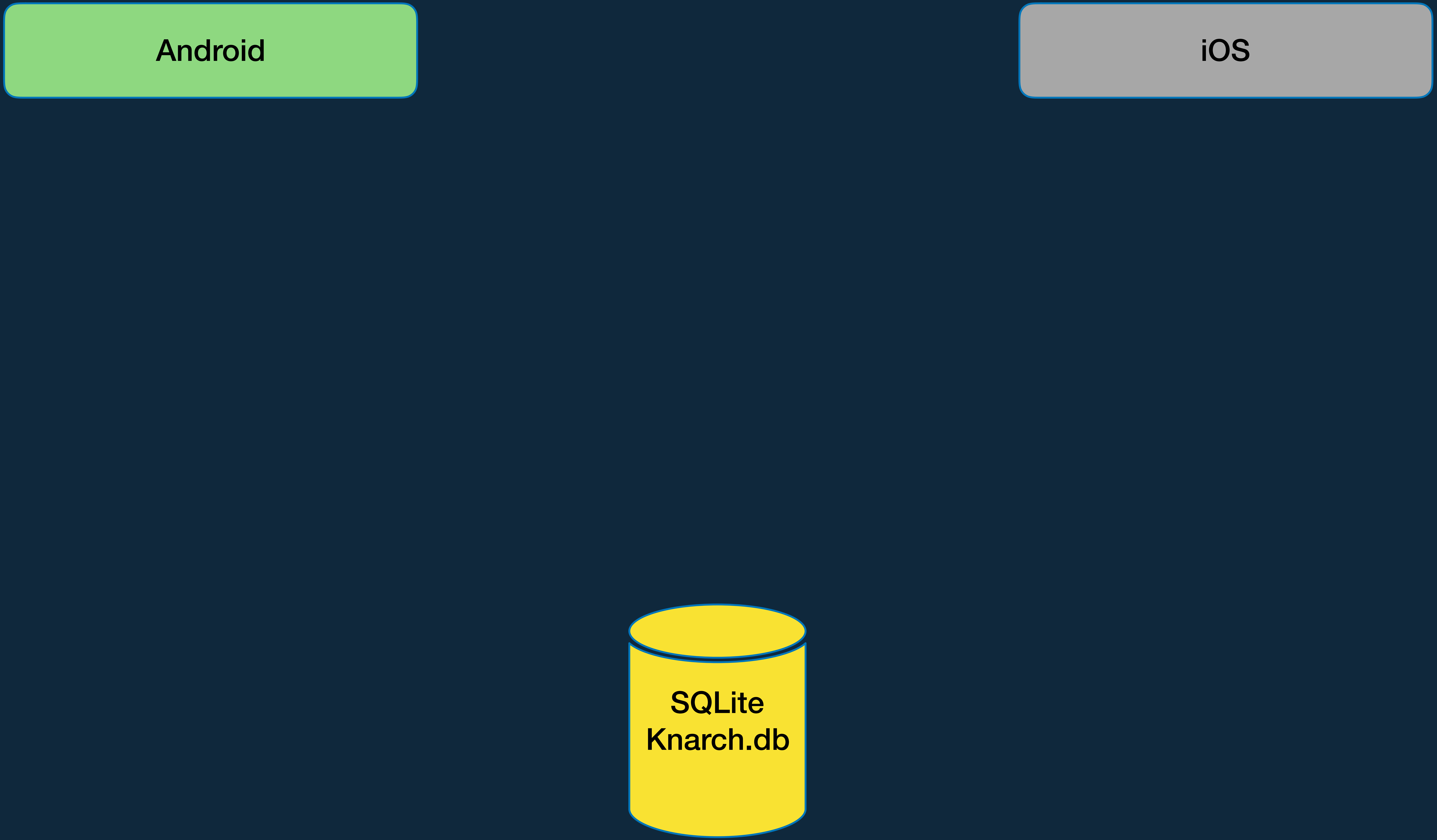
Lots of kotlin

kpgalligan committed on Jul 20, 2014

Droidcon NYC & SF

Droidcon NYC & SF

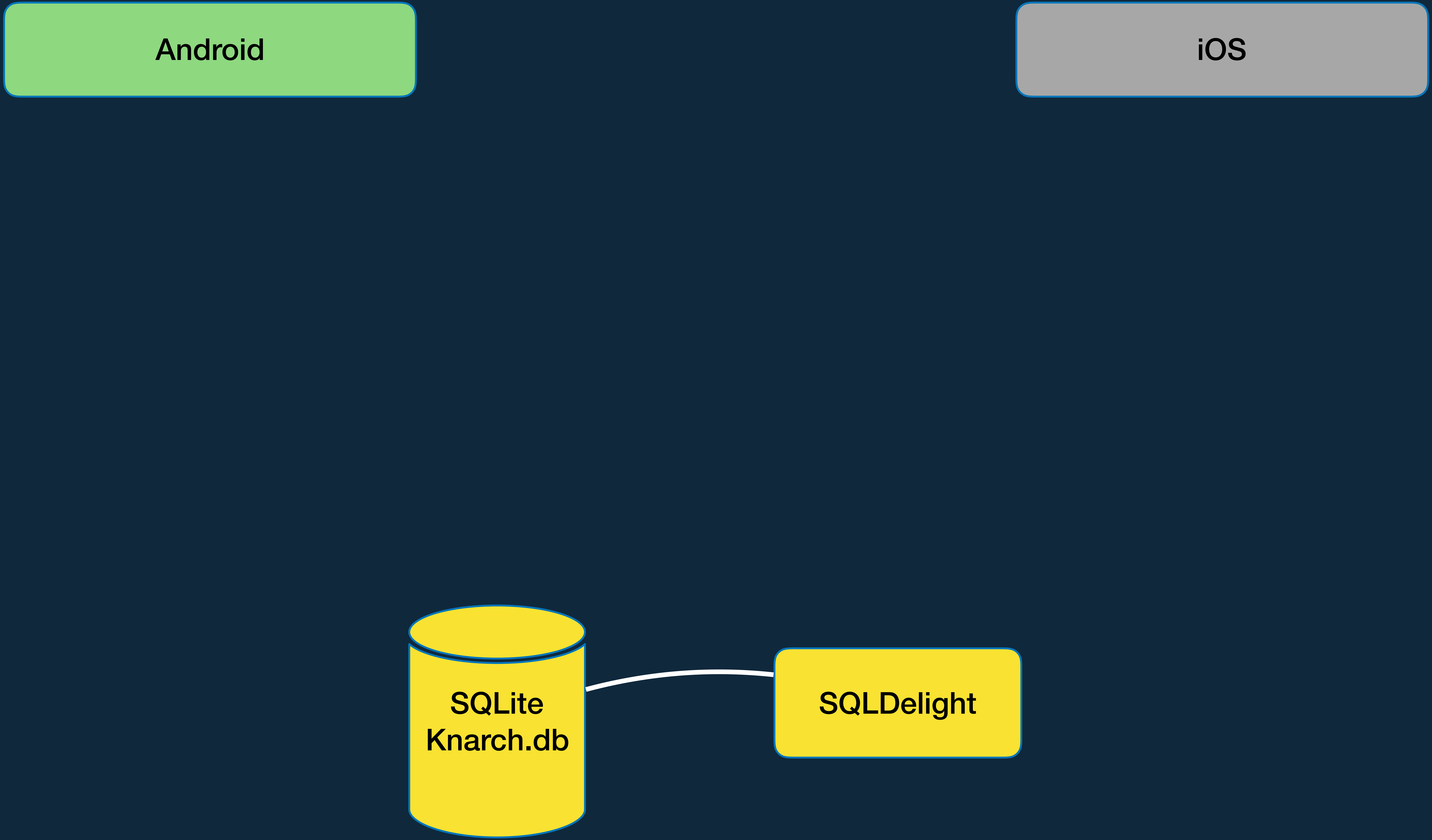
KotlinConf fork, but use the official :)



Android

iOS

SQLite
Knarch.db

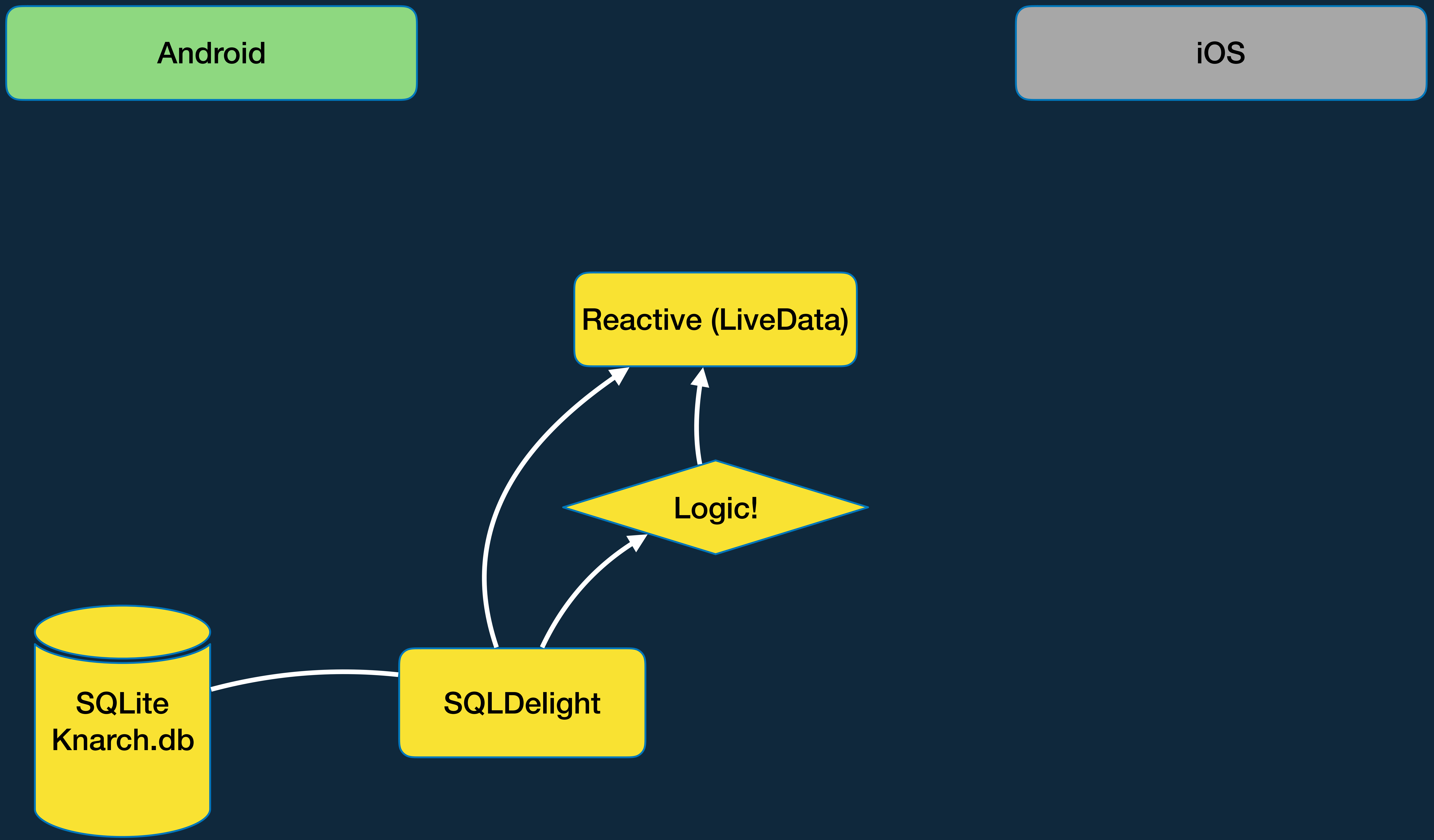


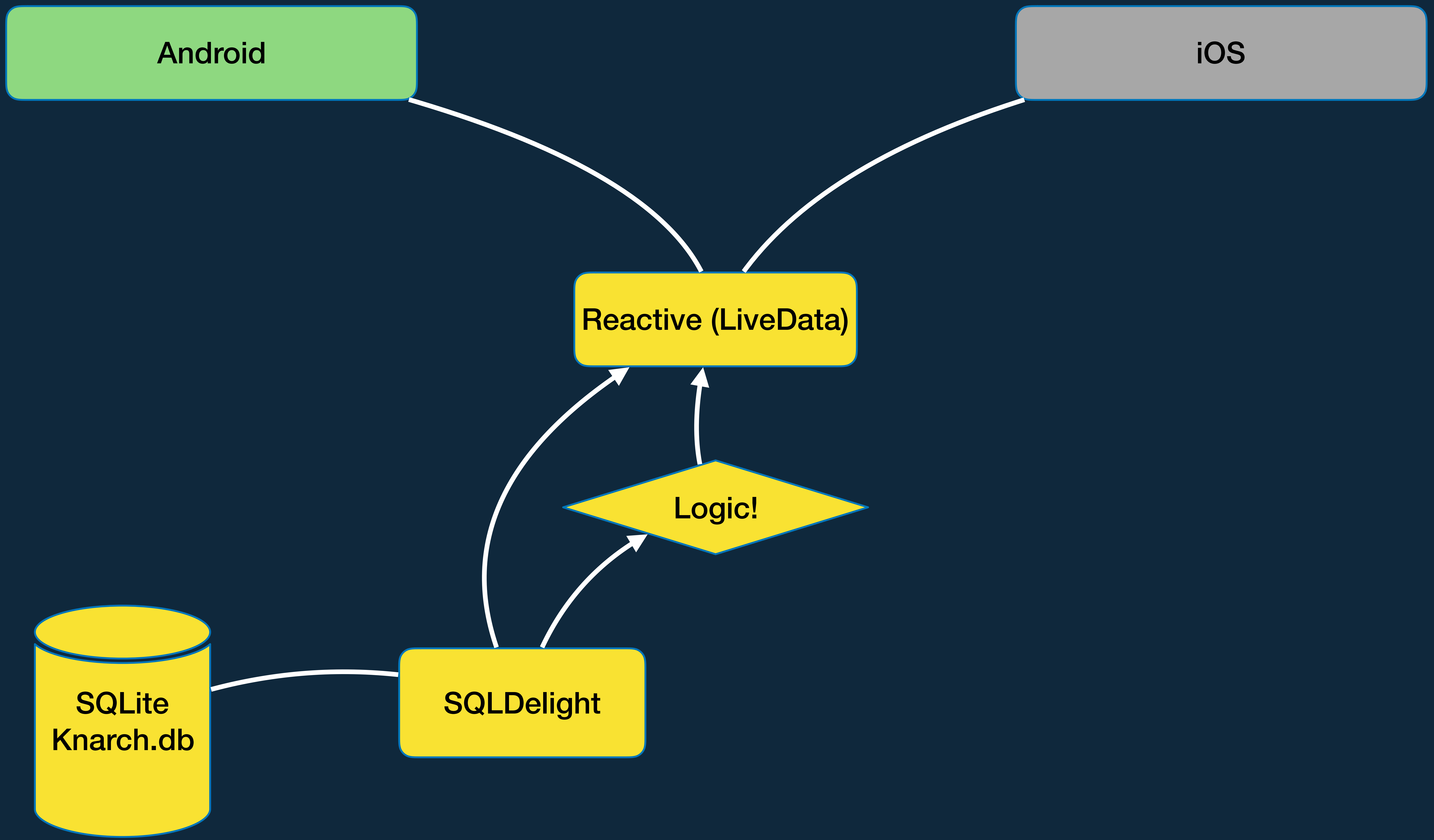
Android

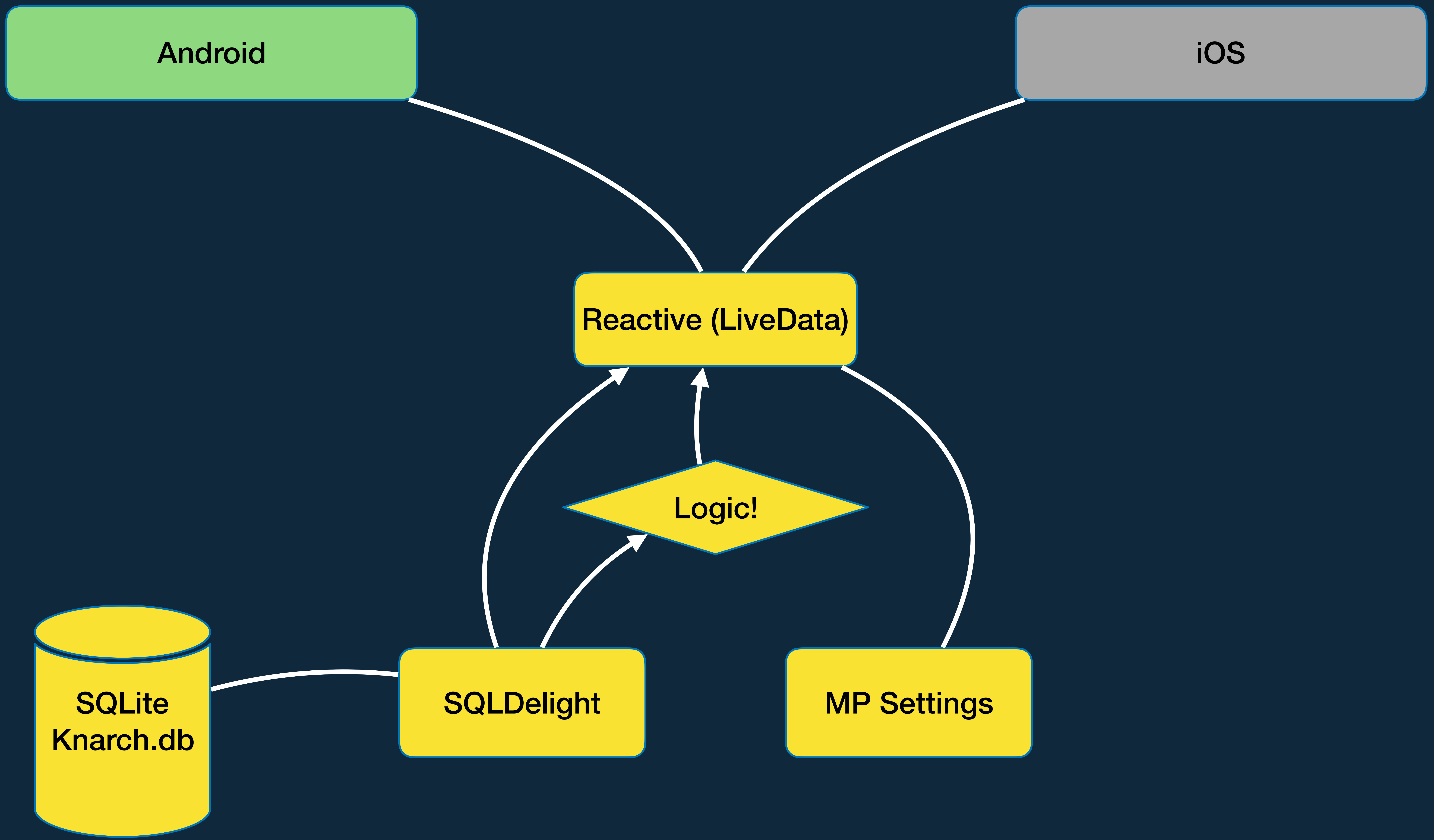
iOS

SQLite
Knarch.db

SQLDelight


Android
iOS
Reactive (LiveData)
Logic!
SQLite
Knarch.db
SQLDelight


Android
iOS
Reactive (LiveData)
Logic!
SQLite
Knarch.db
SQLDelight


Android
iOS
Reactive (LiveData)
Logic!
SQLite
Knarch.db
SQLDelight
MP Settings

```
val evenLiveData:EventLiveData

init {
    val query = goFreeze(AppContext.dbHelper.
            queryWrapper.sessionQueries.
            sessionById(sessionId))
    evenLiveData = EventLiveData(query)
}

fun shutDown(){
    evenLiveData.removeListener()
}
```

```kotlin
val evenLiveData:EventLiveData

init {
    val query = goFreeze(AppContext.dbHelper.
            queryWrapper.sessionQueries.
            sessionById(sessionId))
    evenLiveData = EventLiveData(query)
}

fun shutDown(){
    evenLiveData.removeListener()
}
```

```
override fun onCreateView(inflater: LayoutInflater,
container: ViewGroup?, savedInstanceState: Bundle?): View?
{
    eventViewModel.eventModel.evenLiveData.
            observe(viewLifecycleOwner,
                    Observer { dataRefresh(it) })

    return initView(inflater, container)
}
```

```
override fun onCreateView(inflater: LayoutInflater,
container: ViewGroup?, savedInstanceState: Bundle?): View?
{
    eventViewModel.eventModel.evenLiveData.
            observe(viewLifecycleOwner,
                    Observer { dataRefresh(it) })

    return initView(inflater, container)
}
```

```kotlin
fun registerForChanges(proc:(sessionInfo:SessionInfo)->Unit){
    eventObserver = object : Observer<SessionInfo>{
        override fun onChanged(t: SessionInfo?){
            if(t != null)
                proc(t)
        }
    }

    eventModel.evenLiveData.observeForever(eventObserver!!)
}
```

```
viewModel = EventViewModel(sessionId: sessionId)
viewModel.registerForChanges(proc: updateUi)
```

```swift
viewModel = EventViewModel(sessionId: sessionId)
viewModel.registerForChanges(proc: updateUi)
```

```swift
func updateUi(sessionInfo:SessionInfo) -> KotlinUnit{
    self.sessionInfo = sessionInfo
    styleButton()
    updateAllUi()
    return KotlinUnit()
}
```

Now with 0.9.3!

Droidcon App Kotlin Multiplatform

https://www.youtube.com/watch?v=YAeDK3Ei0Lk

https://github.com/touchlab/DroidconKotlin/

KOTLINCONF
WITH
KOTLIN MULTIPLATFORM
(OBV)

Search or jump to...

Pull requests Issues Marketplace Explore

JetBrains / kotlinconf-app

Watch 101 Star 1,

Code · Issues 8 · Pull requests 0 · Projects 0 · Wiki · Insights

KotlinConf Schedule Application

153 commits · 7 branches · 0 releases · 5 contributors

Branch: new-mpp · New pull request · Create new file · Upload files · Find file

This branch is 145 commits ahead, 4 commits behind master. Pull

4u7 Restore tests in common module · Latest comm

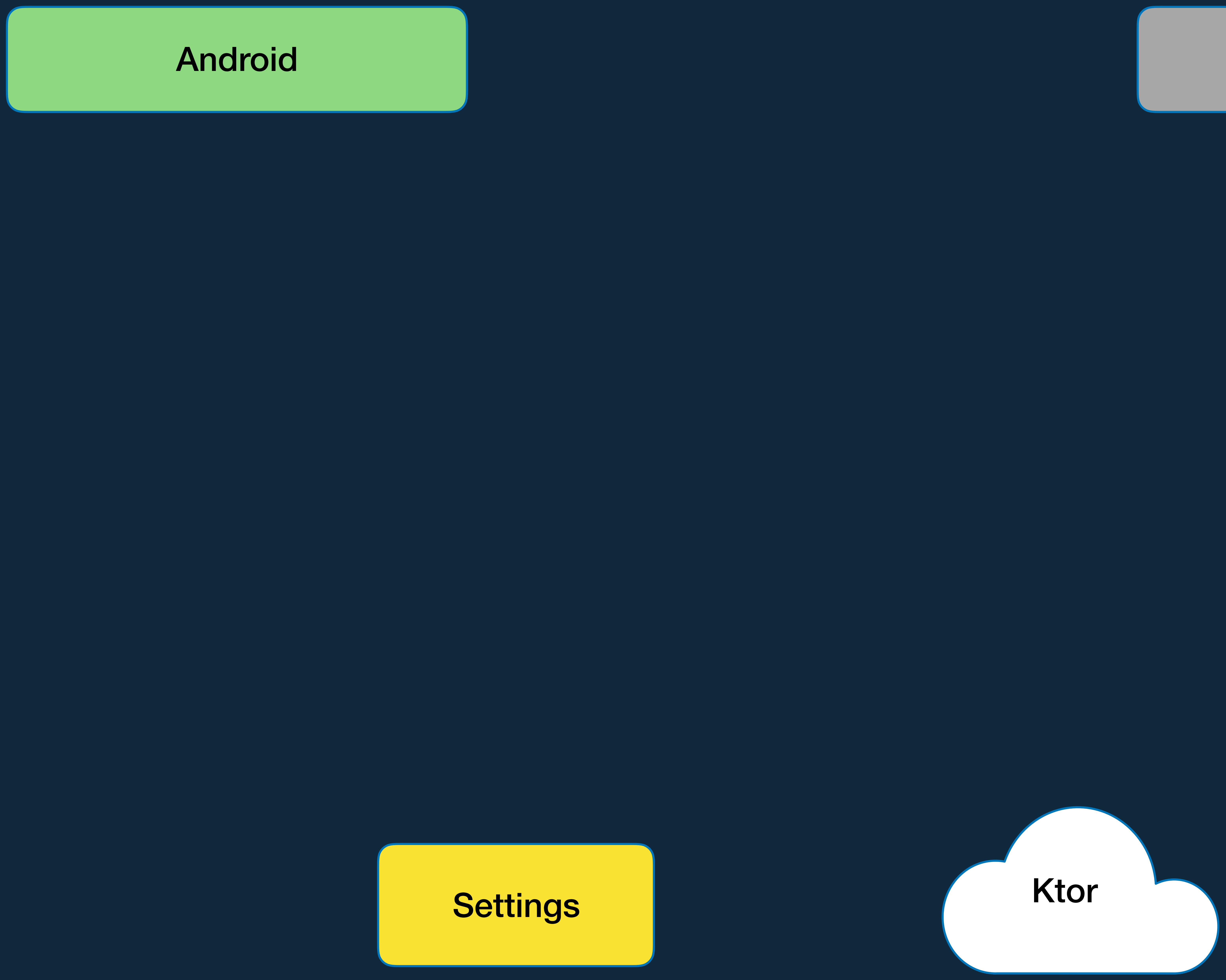
.idea	Add logic to code verification prompt and improve update error handling	11 days ago
android	Restore tests in common module	3 days ago
backend	Merge branch 'develop' into new-mpp	3 days ago
common	Restore tests in common module	3 days ago
gradle/wrapper	Revert back to gradle	2 months ago
konfios	Move konfios to common as source set and build it to framework	3 days ago
.gitignore	Initial sources	11 months ago
LICENSE	Initial commit	11 months ago
Procfile	Added a Heroku button for one-click deploy (#1)	11 months ago

12:56
◂ Droidcon N...

KotlinConf
Amsterdam
2018

To proceed please accept the Privacy Policy

Accept

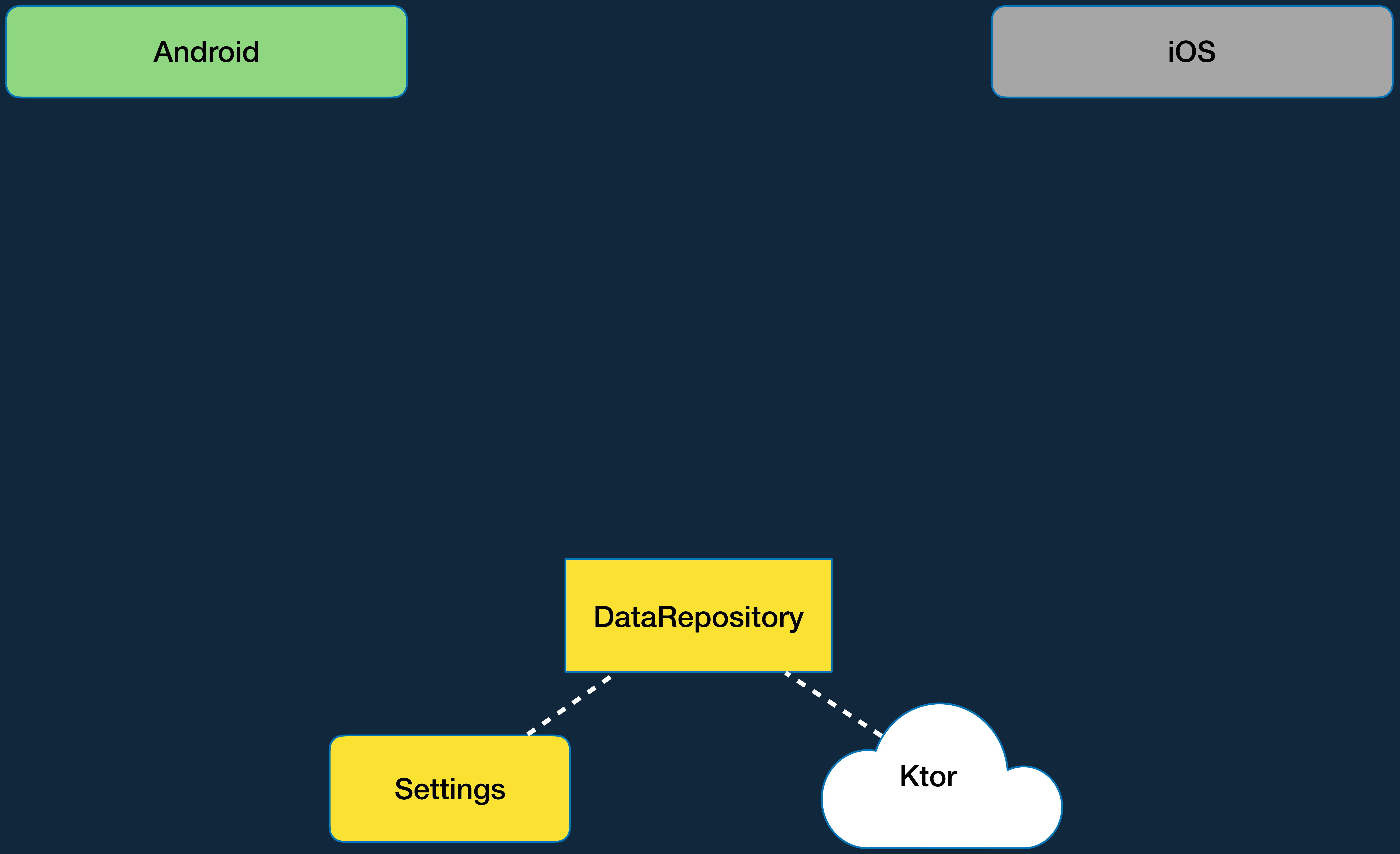


Android

iOS

Settings

Ktor



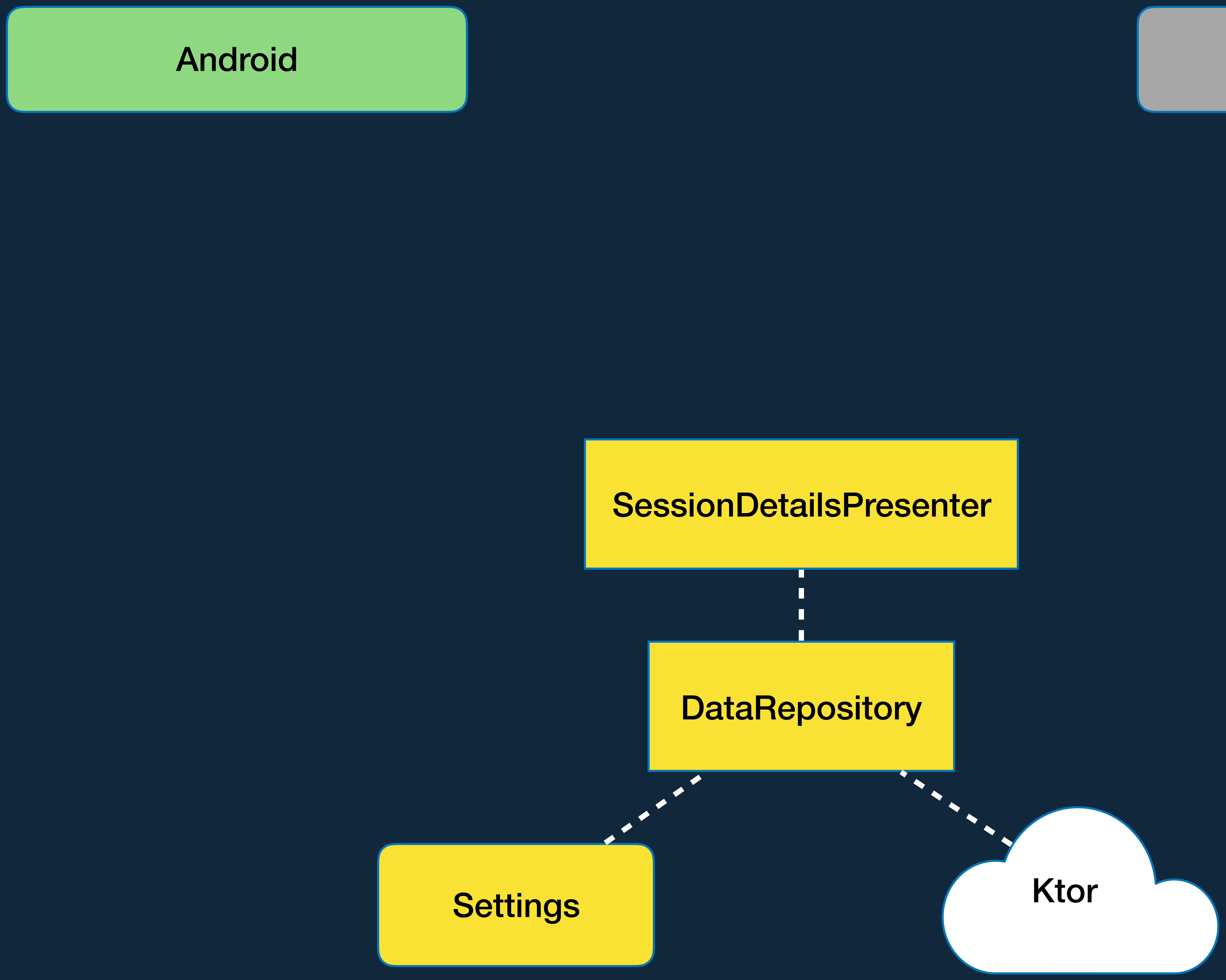
Android

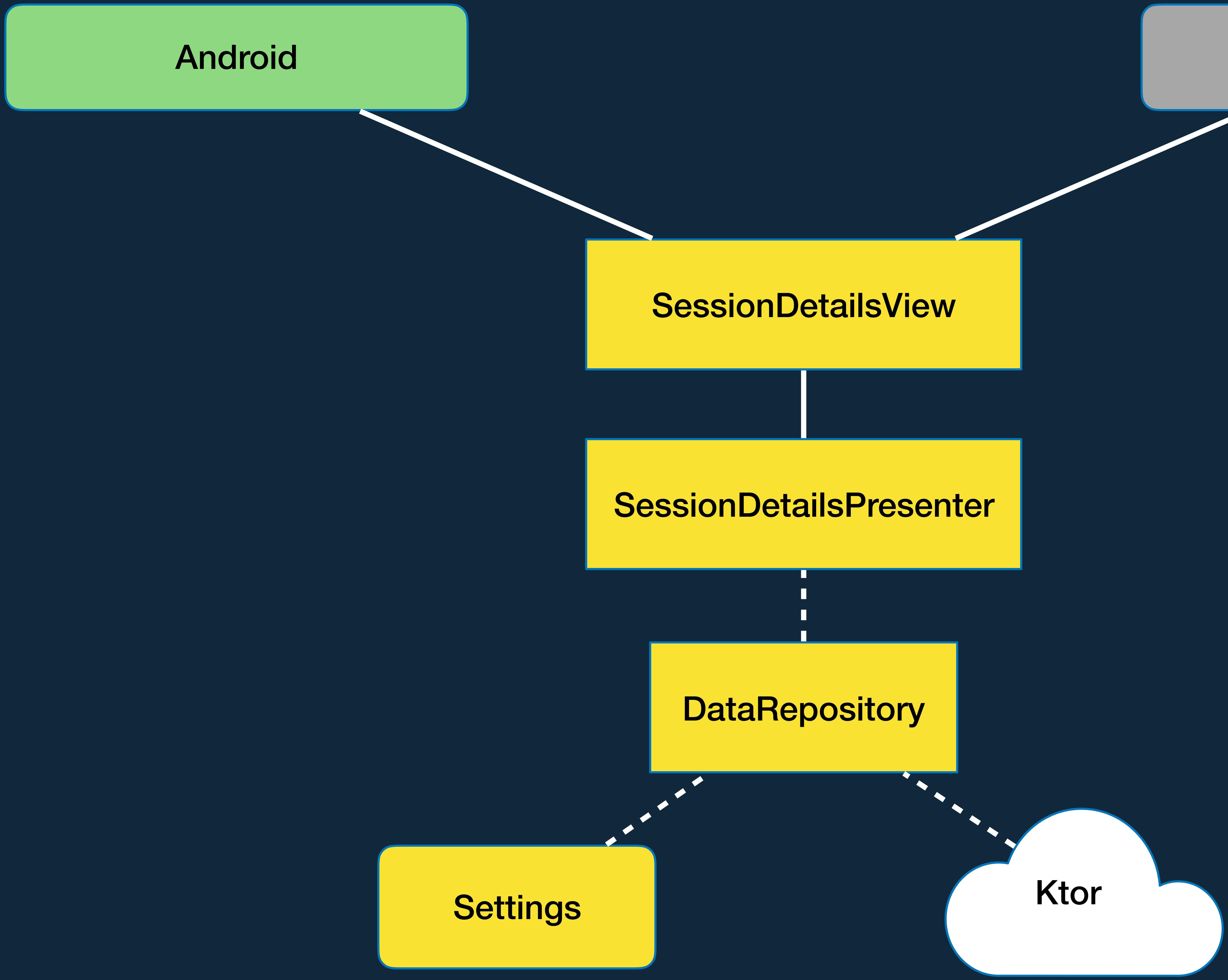
iOS

DataRepository

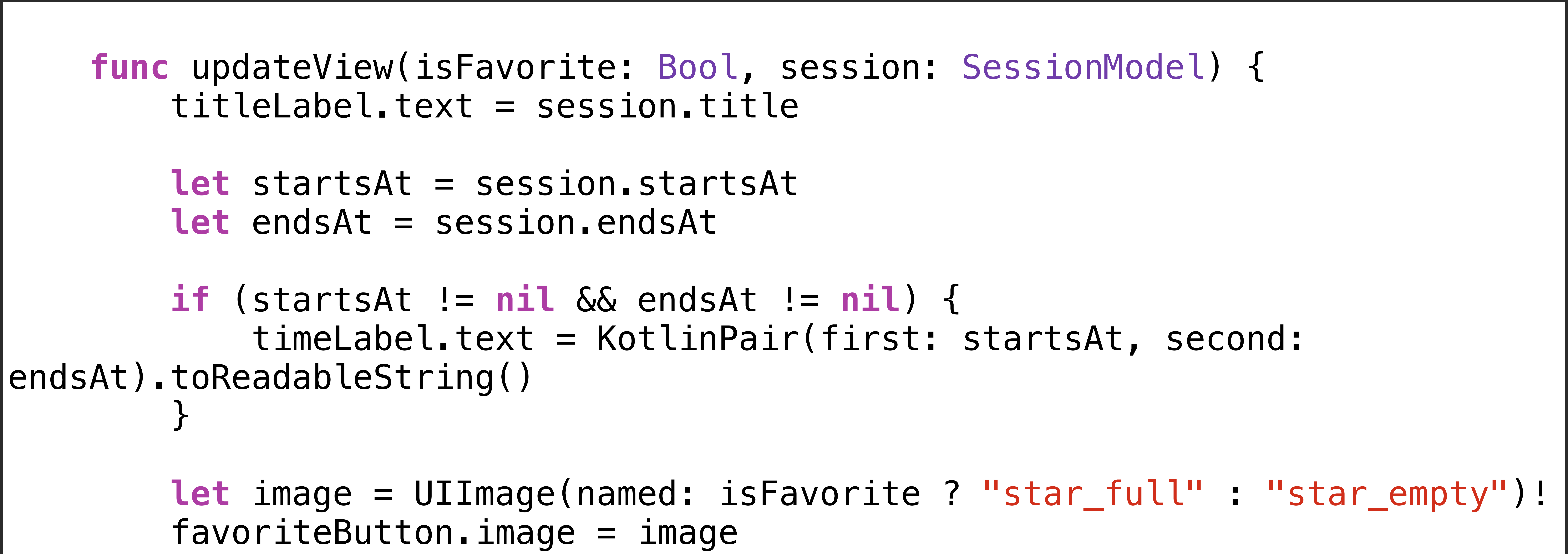
Settings

Ktor

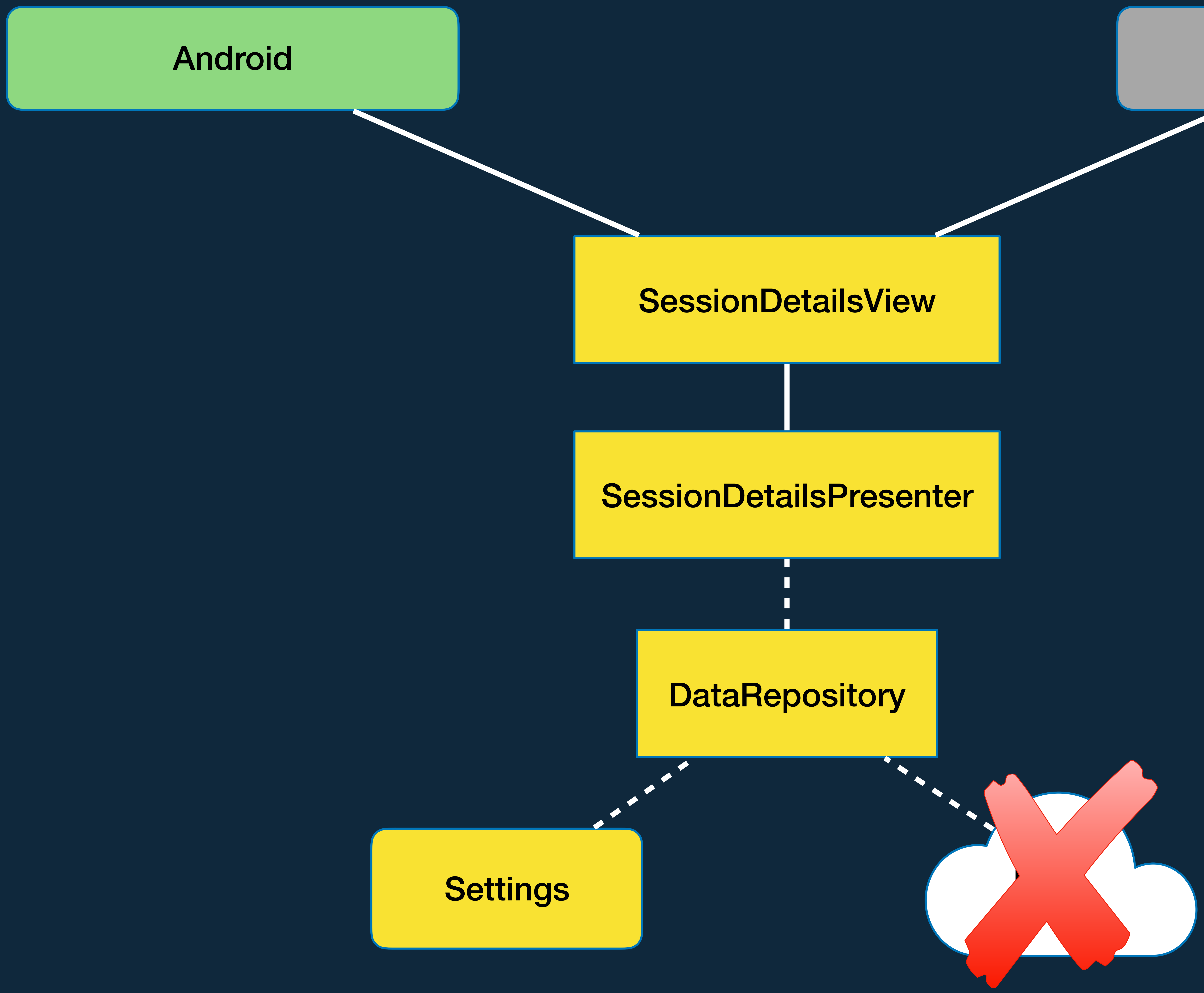

Android
iOS
SessionDetailsPresenter
DataRepository
Settings
Ktor


Android
iOS
SessionDetailsView
SessionDetailsPresenter
DataRepository
Settings
Ktor

```
interface SessionDetailsView : BaseView {
    fun updateView(isFavorite: Boolean, session: SessionModel)
    fun setupRatingButtons(rating: SessionRating?)
    fun setRatingClickable(clickable: Boolean)
}
```



```
    func updateView(isFavorite: Bool, session: SessionModel) {
        titleLabel.text = session.title

        let startsAt = session.startsAt
        let endsAt = session.endsAt

        if (startsAt != nil && endsAt != nil) {
            timeLabel.text = KotlinPair(first: startsAt, second:
endsAt).toReadableString()
        }

        let image = UIImage(named: isFavorite ? "star_full" : "star_empty")!
        favoriteButton.image = image
```


Android
iOS
SessionDetailsView
SessionDetailsPresenter
DataRepository
Settings

KotlinConf App

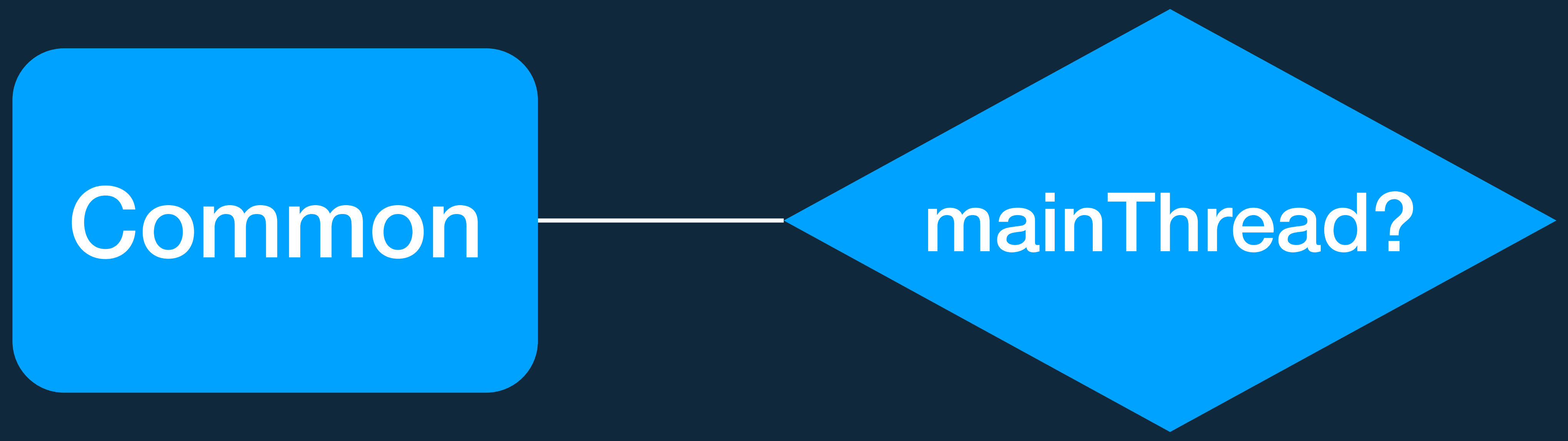
https://github.com/JetBrains/kotlinconf-app

SHARED CODE

FOR

ANDROID & IOS

Common



Common

mainThread?

```
expect val mainThread:Boolean
expect val mainThread:Boolean


actual val mainThread: Boolean
    get() = Looper.myLooper() === Looper.getMainLooper()
```

```
expect val mainThread:Boolean


actual val mainThread: Boolean
    get() = Looper.myLooper() === Looper.getMainLooper()

actual val mainThread: Boolean
    get() = NSThread.isMainThread()
```

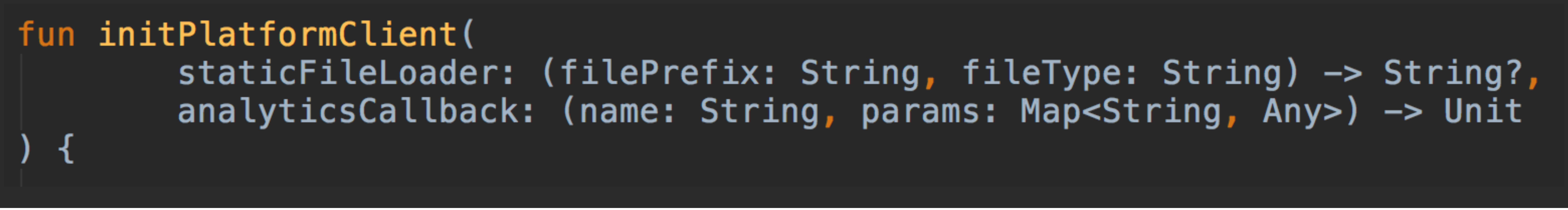
```
expect val mainThread:Boolean


actual val mainThread: Boolean
    get() = Looper.myLooper() === Looper.getMainLooper()

actual val mainThread: Boolean
    get() = NSThread.isMainThread()

actual val mainThread: Boolean = true
```

```kotlin
expect fun currentTimeMillis():Long

expect fun <B> backgroundTask(backJob:()-> B, mainJob:(B) -> Unit)

expect fun backgroundTask(backJob:()->Unit)

expect fun networkBackgroundTask(backJob:()->Unit)

expect fun initContext():NativeOpenHelperFactory

expect fun <T> goFreeze(a:T):T

expect fun <T> T.freeze2(): T

expect fun simpleGet(url:String):String

expect fun logException(t:Throwable)

expect fun settingsFactory(): Settings.Factory

expect fun createUuid():String
expect class Date {
    fun toLongMillis():Long
}

expect class DateFormatHelper(format:String){
    fun toDate(s:String):Date
    fun format(d:Date):String
}
```

```kotlin
actual class Date(val date:java.util.Date) {
    actual fun toLongMillis(): Long = date.time
}

actual class DateFormatHelper actual constructor(format: String) {
    val dateFormatter = object : ThreadLocal<DateFormat>(){
        override fun initialValue(): DateFormat = SimpleDateFormat(format)
    }

    actual fun toDate(s: String): Date = Date(dateFormatter.get()!!.parse(s))

    actual fun format(d: Date): String = dateFormatter.get()!!.format(d.date)
}
```

```kotlin
fun initPlatformClient(
        staticFileLoader: (filePrefix: String, fileType: String) -> String?,
        analyticsCallback: (name: String, params: Map<String, Any>) -> Unit,
        clLogCallback: (s: String) -> Unit) {
```

```
fun initPlatformClient(
        staticFileLoader: (filePrefix: String, fileType: String) -> String?,
        analyticsCallback: (name: String, params: Map<String, Any>) -> Unit,
        clLogCallback: (s: String) -> Unit) {


        AppContext.initPlatformClient ({filePrefix, fileType ->
            loadAsset("${filePrefix}.${fileType}")},
                {name: String, params: Map<String, Any> ->

                    val event = CustomEvent(name)
        //Loop
                    Answers.getInstance().logCustom(event)
                },
                { Log.w("MainApp", it) })
```

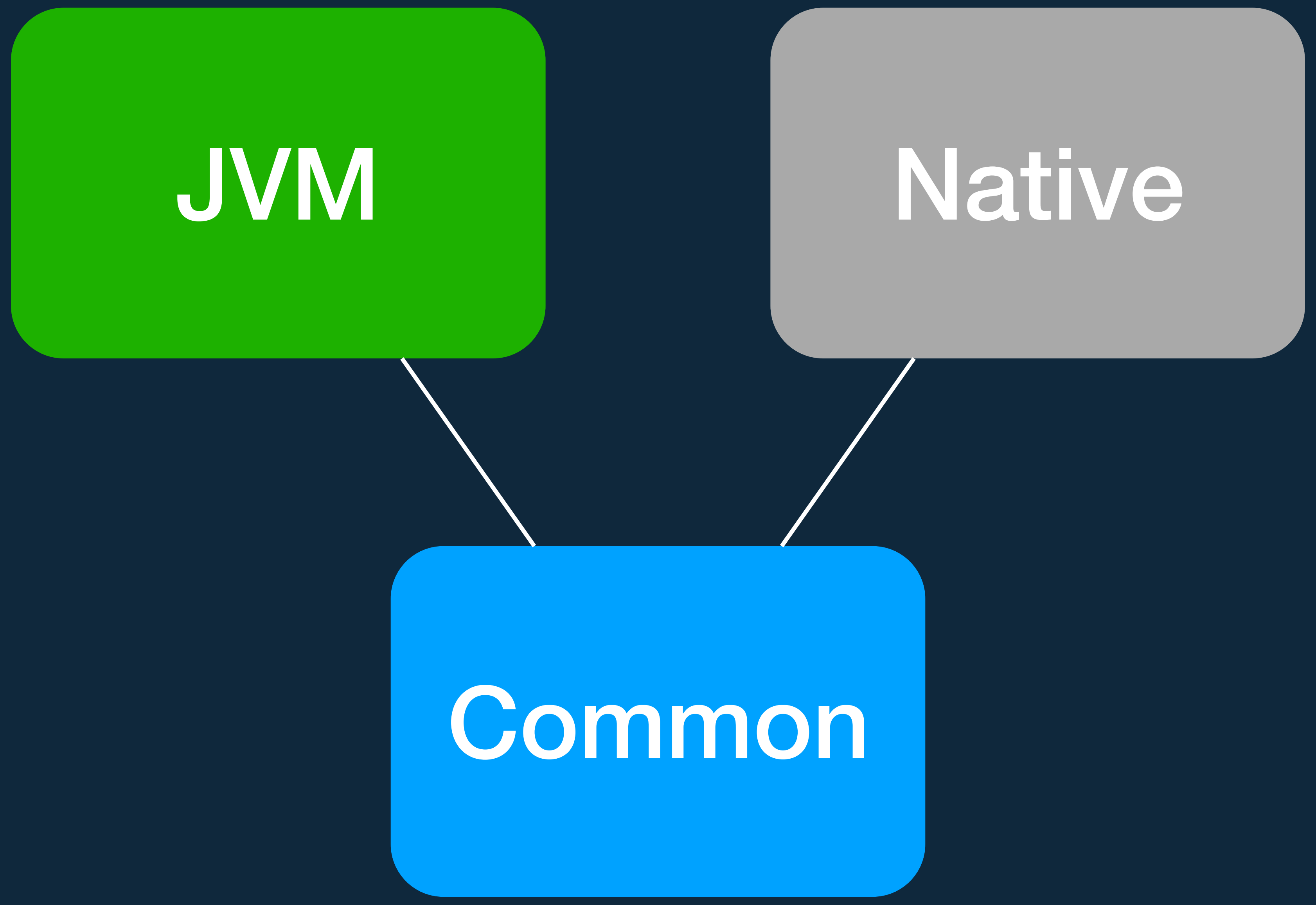


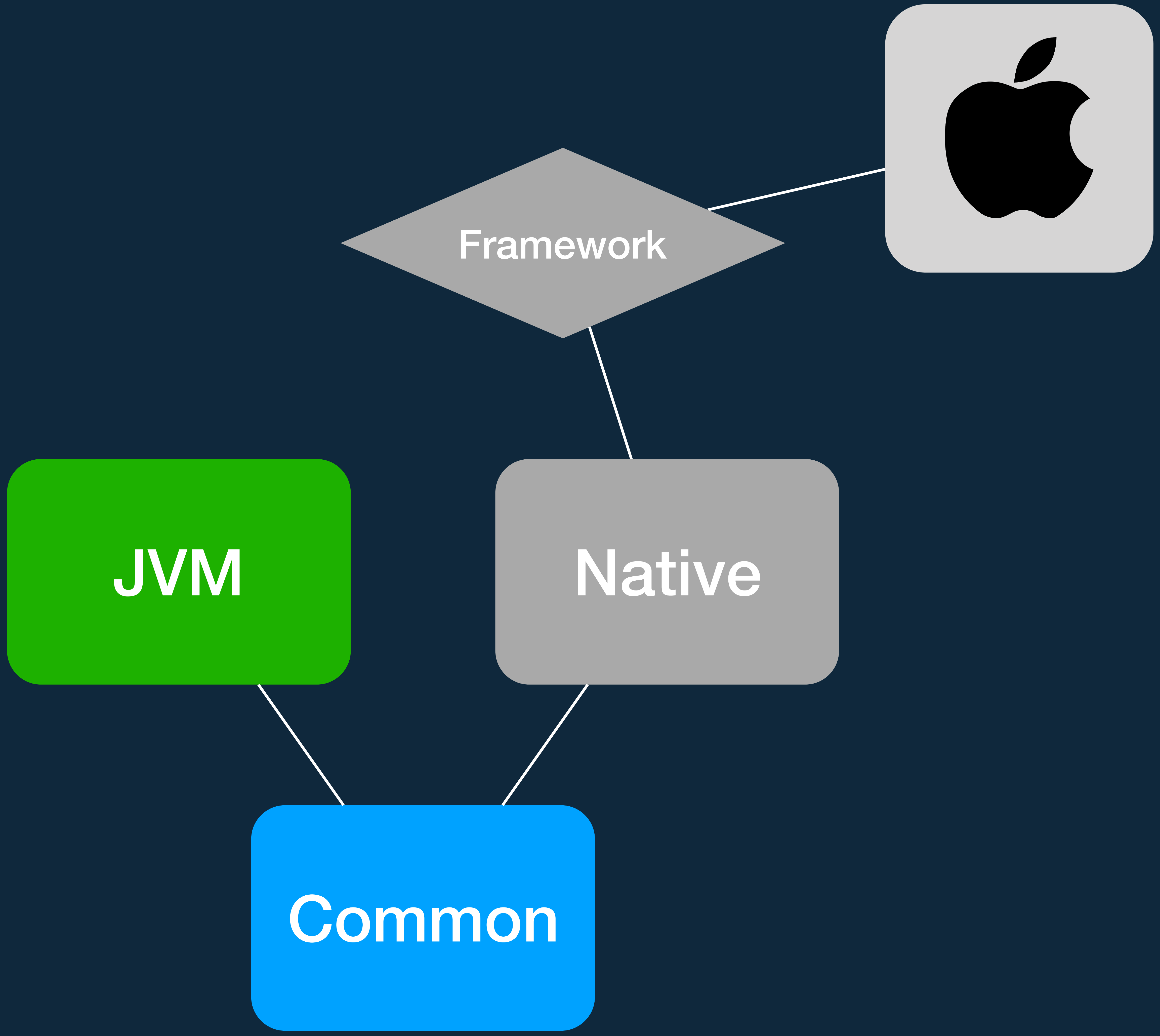
```
fun initPlatformClient(
        staticFileLoader: (filePrefix: String, fileType: String) -> String?,
        analyticsCallback: (name: String, params: Map<String, Any>) -> Unit
) {
```

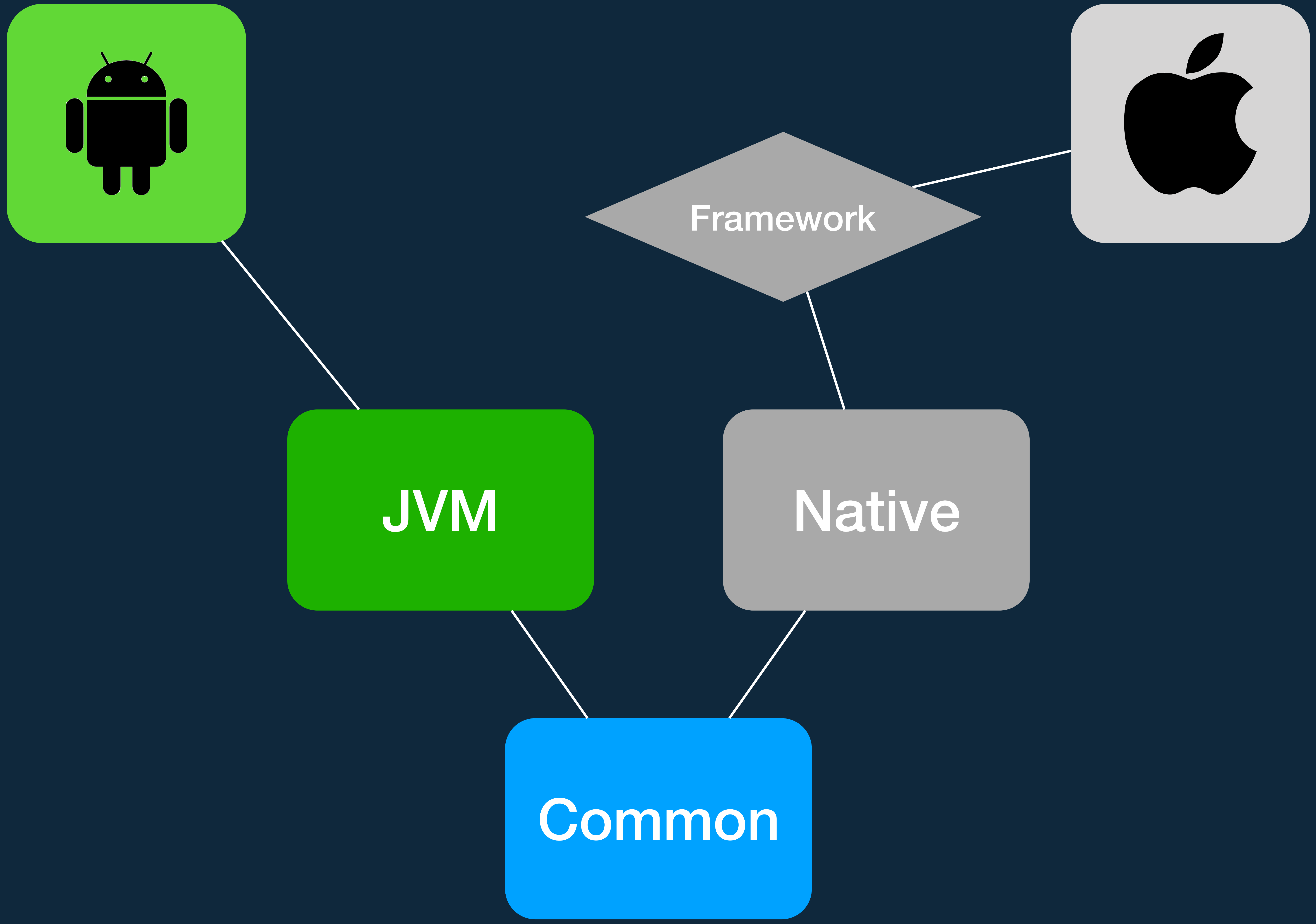
```
let appContext = AppContext()
        appContext.doInitPlatformClient(staticFileLoader: loadAsset,
                                                        analyticsCallback:
analyticsCallback,
                                                        clLogCallback: csLog)
```

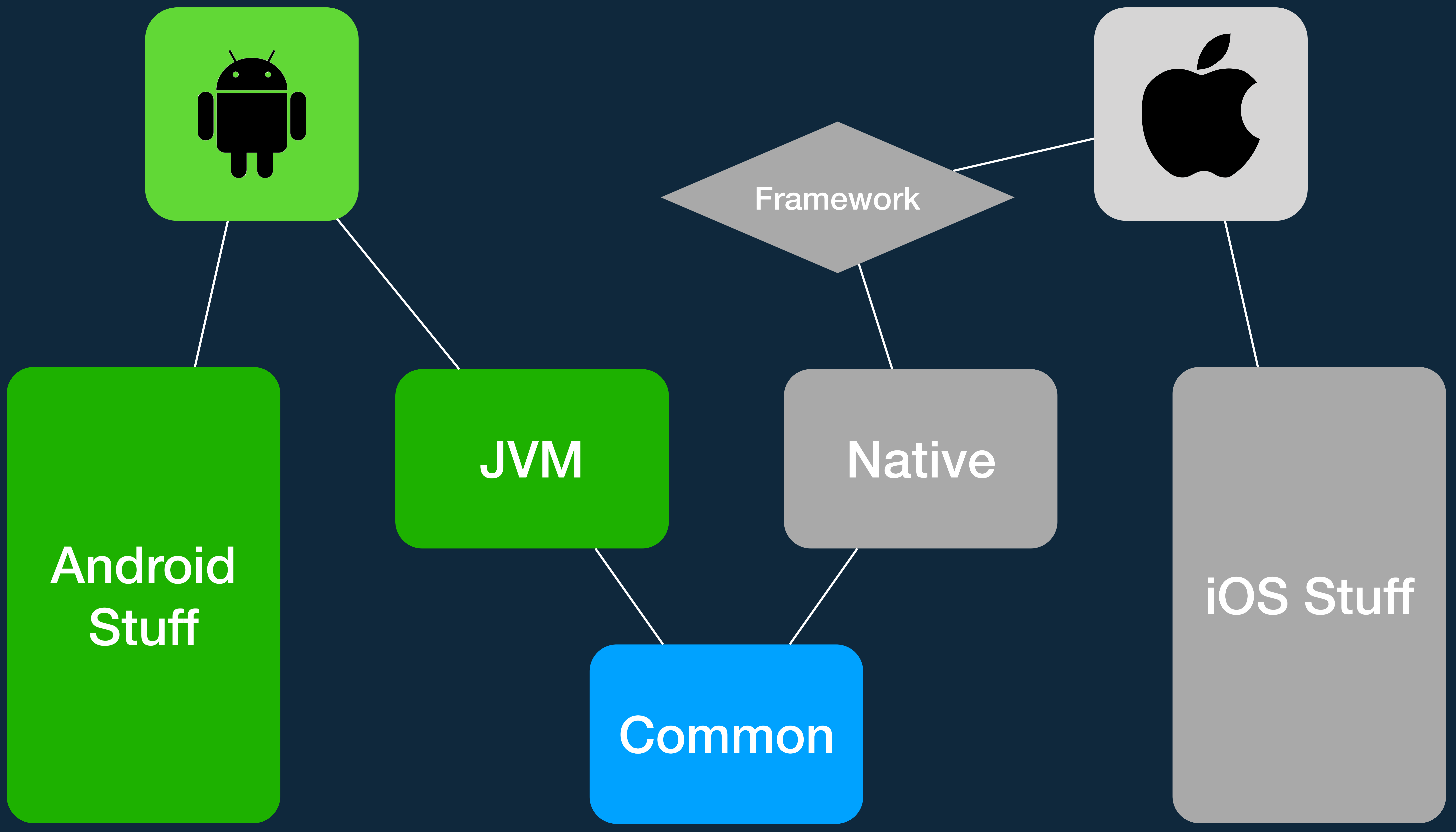
```
func loadAsset(filePrefix:String, fileType:String) -> String?{
        do{
            let bundleFile = Bundle.main.path(forResource: filePrefix,
ofType: fileType)
            return try String(contentsOfFile: bundleFile!)
        } catch {
            return nil
        }
    }
```

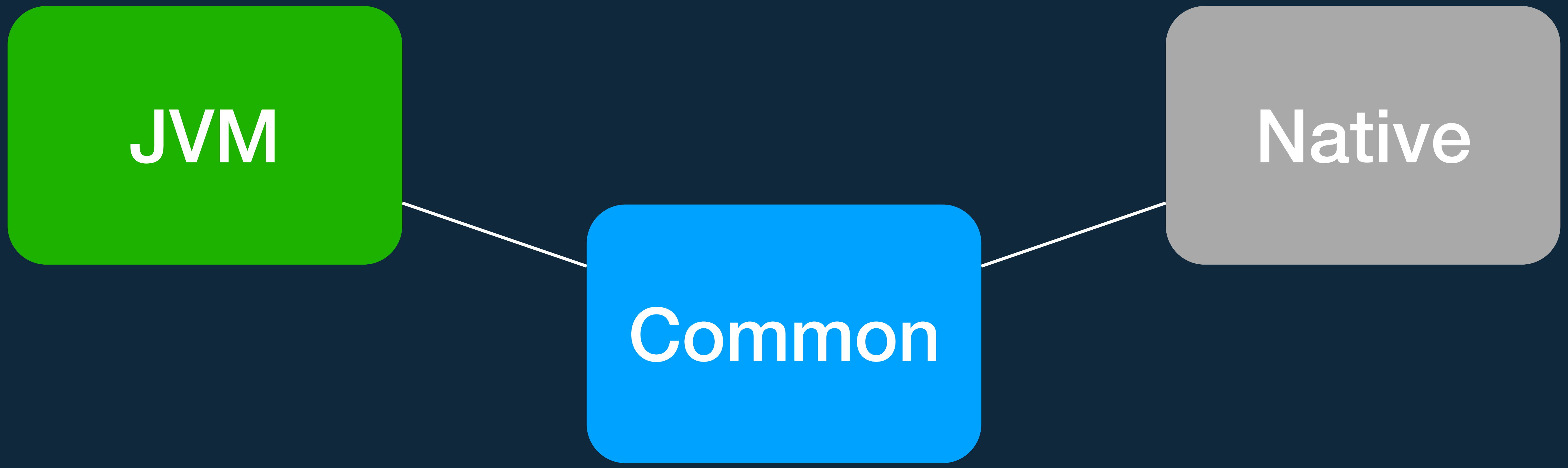
Common


JVM
Native
Common

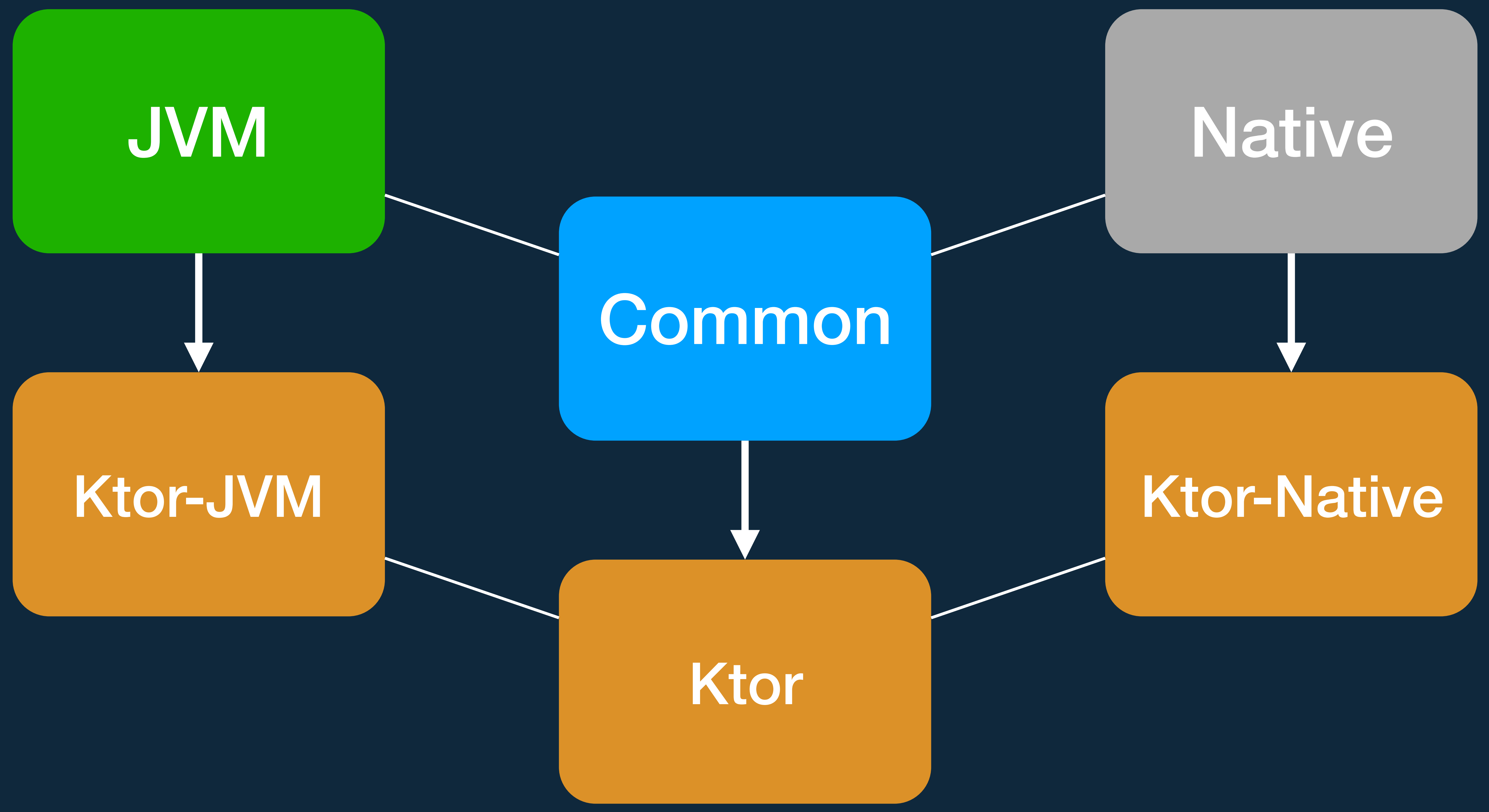

Framework
JVM
Native
Common


Framework
JVM
Native
Common


Framework
JVM
Native
Android Stuff
Common
iOS Stuff



JVM
Native
Common



JVM
Native
Common
Ktor-JVM
Ktor-Native
Ktor

📱iOS Dev Info

Reference Counting

but not ***your*** reference counting

No Reference Cycles

Can call from Swift

although some complaints

No bitcode support

yet...

Threading is Different

that’s for everybody

Nikolay Igotti

JetBrains, Kotlin/Native tech lead

Worked on various system level software (Hotspot JVM, VirtualBox, Native Client) at Sun, EMC, Oracle and Google. Now implementing native backend and the runtime for Kotlin programming language.

Kotlin/Native concurrency model

Kotlin/Native runtime is designed to minimize possible programmers mistakes related to concurrency and undesired mutable state. To achieve this goal, while keeping the source-level compatibility with Kotlin, runtime and standard library was carefully designed to avoid concurrently accessible mutable state. In this talk we will discuss both reasoning behind this design, design and implementation details of the runtime and compiler. Also generic topics of automated memory management in the compiled language are covered.

STRANGER THREADS

https://medium.com/@kpgalligan/kotlin-native-stranger-threads-c0cf0e0fb847

STRANGER THREADS

Episode 2 soon!

Kotlin Conference 04 OCT

Saner concurrency

Effectenbeurzaal

KotlinConf
LIVE
YouTube

IDE TOOLS

&

GRADLE PLUGINS

¯_(ツ)_/¯

Multiplatform IDE

Intellij community and Android Studio!

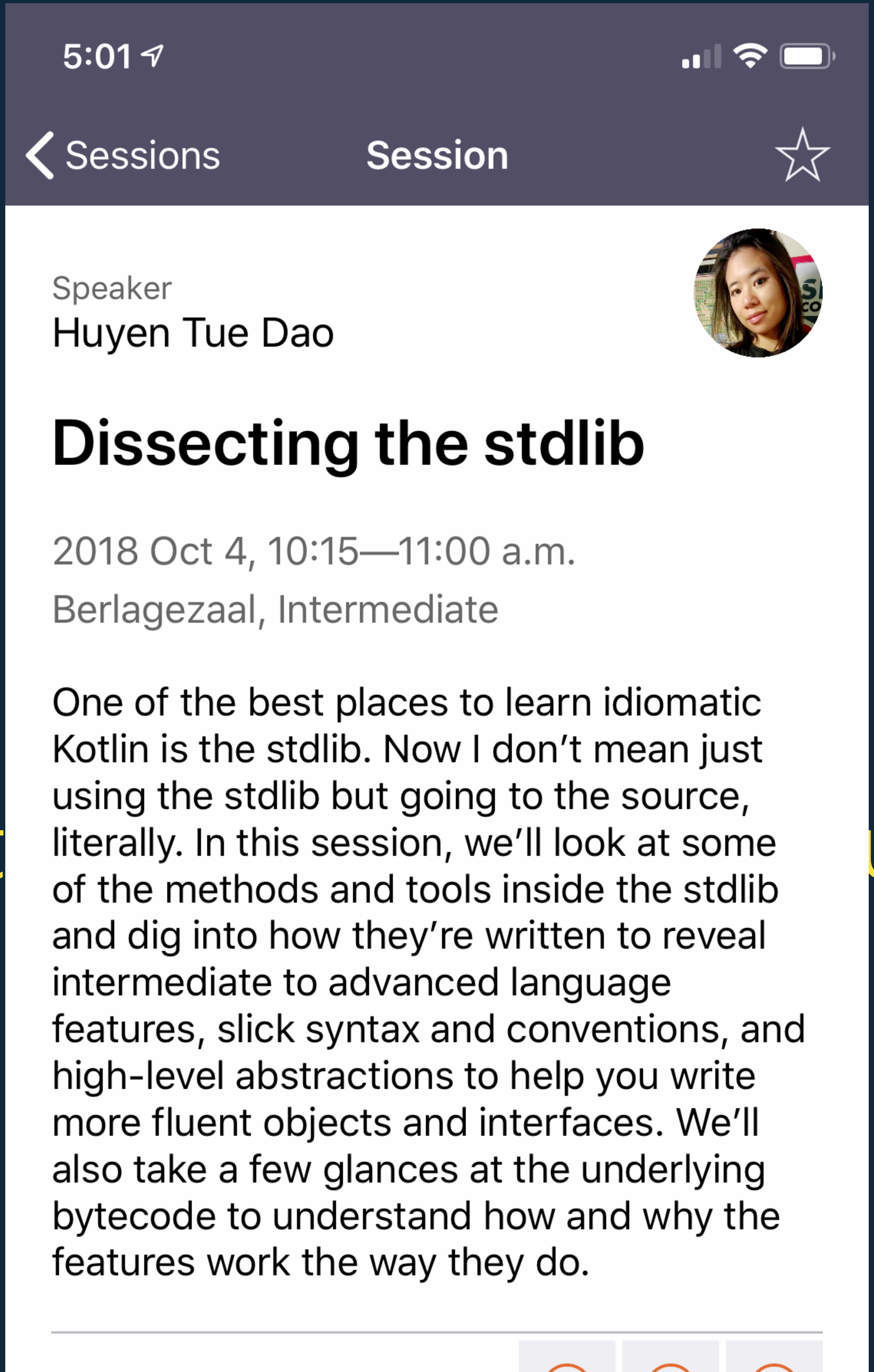
Multiplatform Gradle

new and changing

Other Plugins?

Rivers of Eden
Managing Water for Peace in the Middle East
The Water Sellers
THE INDUS RIVER
Water and the Ganges Valley
Towards Water Wisdom
Water and power
CHAPMAN AND THOMPSON
RURAL DEVELOPMENT INSTITUTE
The Jonglei Canal
America's Water

LIBRARIES



Speaker
Huyen Tue Dao

Dissecting the stdlib

2018 Oct 4, 10:15—11:00 a.m.
Berlagezaal, Intermediate

One of the best places to learn idiomatic Kotlin is the stdlib. Now I don’t mean just using the stdlib but going to the source, literally. In this session, we’ll look at some of the methods and tools inside the stdlib and dig into how they’re written to reveal intermediate to advanced language features, slick syntax and conventions, and high-level abstractions to help you write more fluent objects and interfaces. We’ll also take a few glances at the underlying bytecode to understand how and why the features work the way they do.

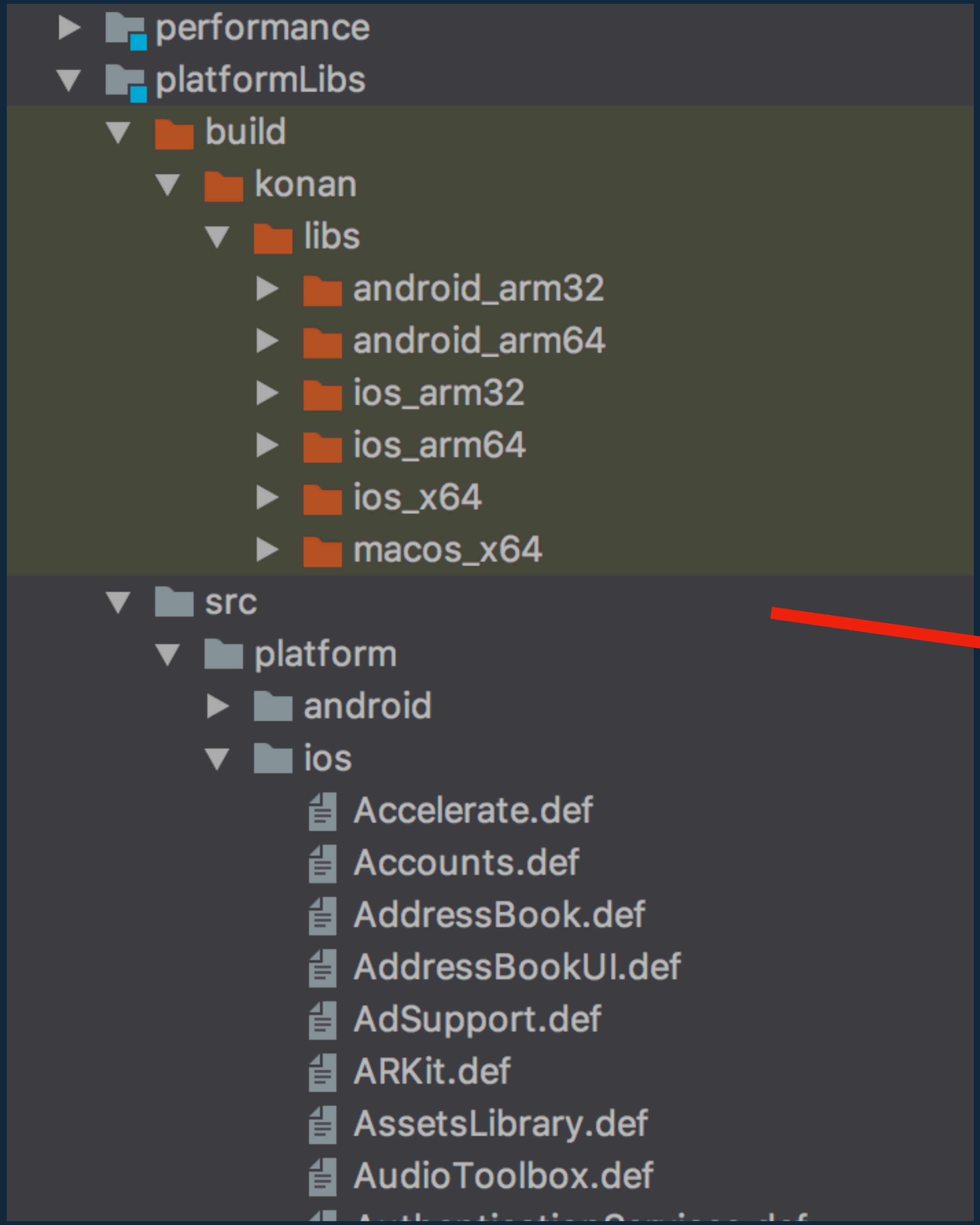
not ut...

Kotlin/Native Runtime

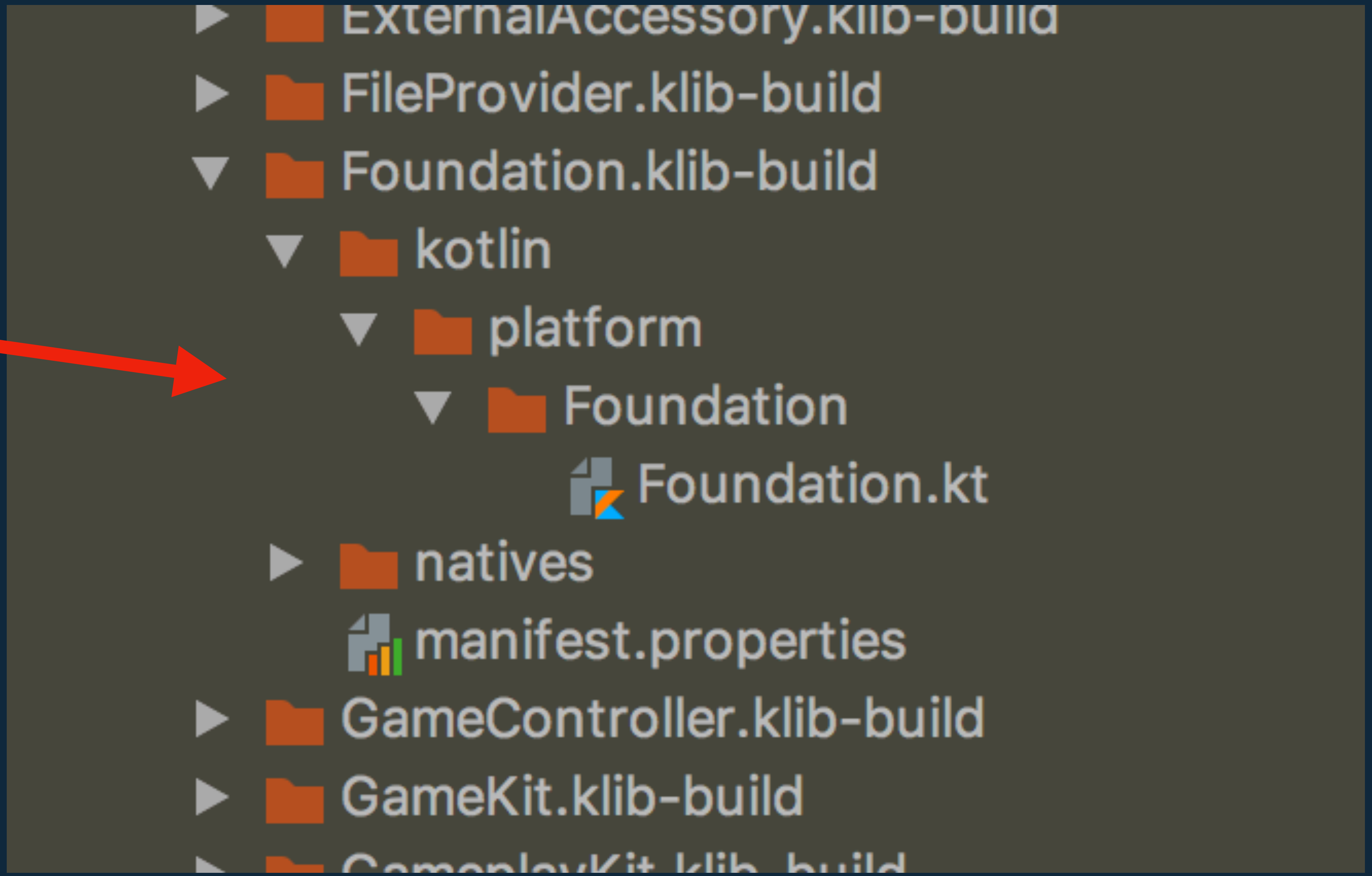
also not a ***library***, still…

https://github.com/JetBrains/kotlin-native

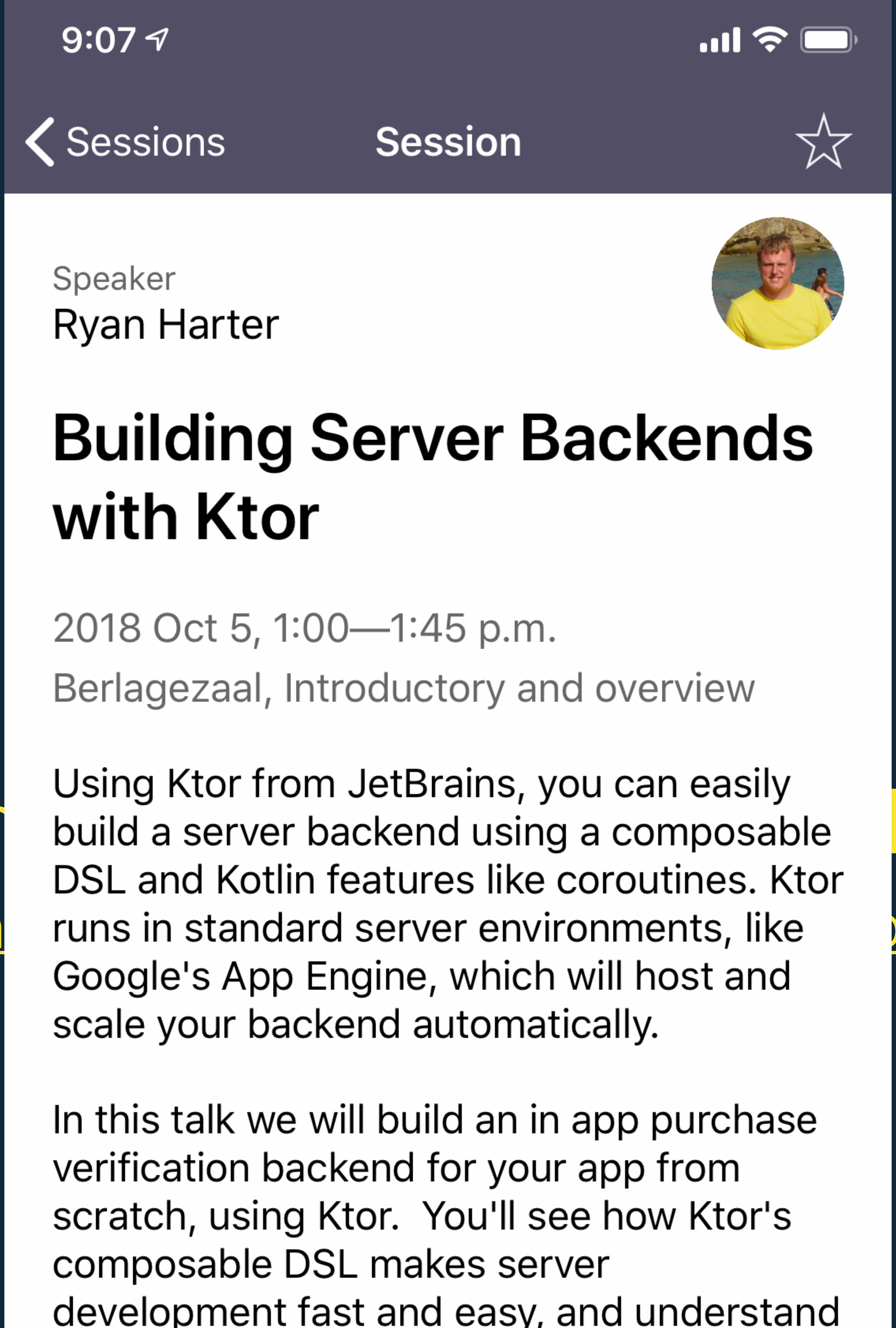
- kotlin/native/concurrent/Freezing.kt
- kotlin/native/Annotations.kt
- kotlin/native/concurrent/Worker.kt (maybe)



```
▶ performance
▼ platformLibs
  ▼ build
    ▼ konan
      ▼ libs
        ▶ android_arm32
        ▶ android_arm64
        ▶ ios_arm32
        ▶ ios_arm64
        ▶ ios_x64
        ▶ macos_x64
  ▼ src
    ▼ platform
      ▶ android
      ▼ ios
          Accelerate.def
          Accounts.def
          AddressBook.def
          AddressBookUI.def
          AdSupport.def
          ARKit.def
          AssetsLibrary.def
          AudioToolbox.def
          AuthenticationServices.def
```



```
▶ ExternalAccessory.klib-build
▶ FileProvider.klib-build
▼ Foundation.klib-build
  ▼ kotlin
    ▼ platform
      ▼ Foundation
          Foundation.kt
  ▶ natives
    manifest.properties
▶ GameController.klib-build
▶ GameKit.klib-build
▶ GameplayKit.klib-build
```



Session

Speaker
Ryan Harter

Building Server Backends with Ktor

2018 Oct 5, 1:00—1:45 p.m.

Berlagezaal, Introductory and overview

Using Ktor from JetBrains, you can easily build a server backend using a composable DSL and Kotlin features like coroutines. Ktor runs in standard server environments, like Google's App Engine, which will host and scale your backend automatically.

In this talk we will build an in app purchase verification backend for your app from scratch, using Ktor. You'll see how Ktor's composable DSL makes server development fast and easy, and understand

asynch lient(s)

h or

Kotlinx.serialization

cross-platform / multi-format reflectionless serialization

https://github.com/Kotlin/kotlinx.serialization

Kotlinx.coroutines

makes coroutines usable

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

Code
Issues 87
Pull requests 14
Projects 0
Wiki
Insights
New issue
Open
Collaborator
18
Closed
Subscribe

Support multi-threaded coroutines on Kotlin/Native #462

elizarov opened this issue on Jul 27 · 3 comments

elizarov commented on Jul 27 • edited

You can have multiple threads in Kotlin/Native. Each thread can have its own event loop with `runBlocking` and have number of coroutines running there. However, currently communication between those threads via coroutine primitives (like channels) is not supported. This issue it to track enhancement of Kotlin/Native in `kotlinx.coroutines` library so that all the following becomes possible:

- Launching coroutines from one thread with a dispatcher on another thread
- Await/join coroutine running on another thread
- Send/Receive elements to/from coroutines on other threads

elizarov added the enhancement label on Jul 27

elizarov referenced this issue on Jul 27

Backgroud execution with Delay in Kotlin/Native #461

gwwdfsad referenced this issue on Jul 31

Assignees

No one assigned

Labels

enhancement

Projects

None yet

Milestone

No milestone

Notifications

You're not receiving notifications from this thread.

3 participants

Code
Issues 87
Pull requests 14
Projects 0
Wiki
Insights

mohit-gurumuk... commented on Aug 2

Second that. Can we please get a rough estimate?

elizarov commented on Aug 3 — Collaborator

We're in the design phase now. I'll update you on the status in couple of weeks.

👍 4

brettwillis referenced this issue 12 days ago

EXC_BAD_ACCESS when releasing lambda #2052 — Open

kpgalligan referenced this issue 6 days ago

what is something better should be coming really soon #1 — Closed

Backgroud execution with Delay in Kotlin/Native #461 — Closed

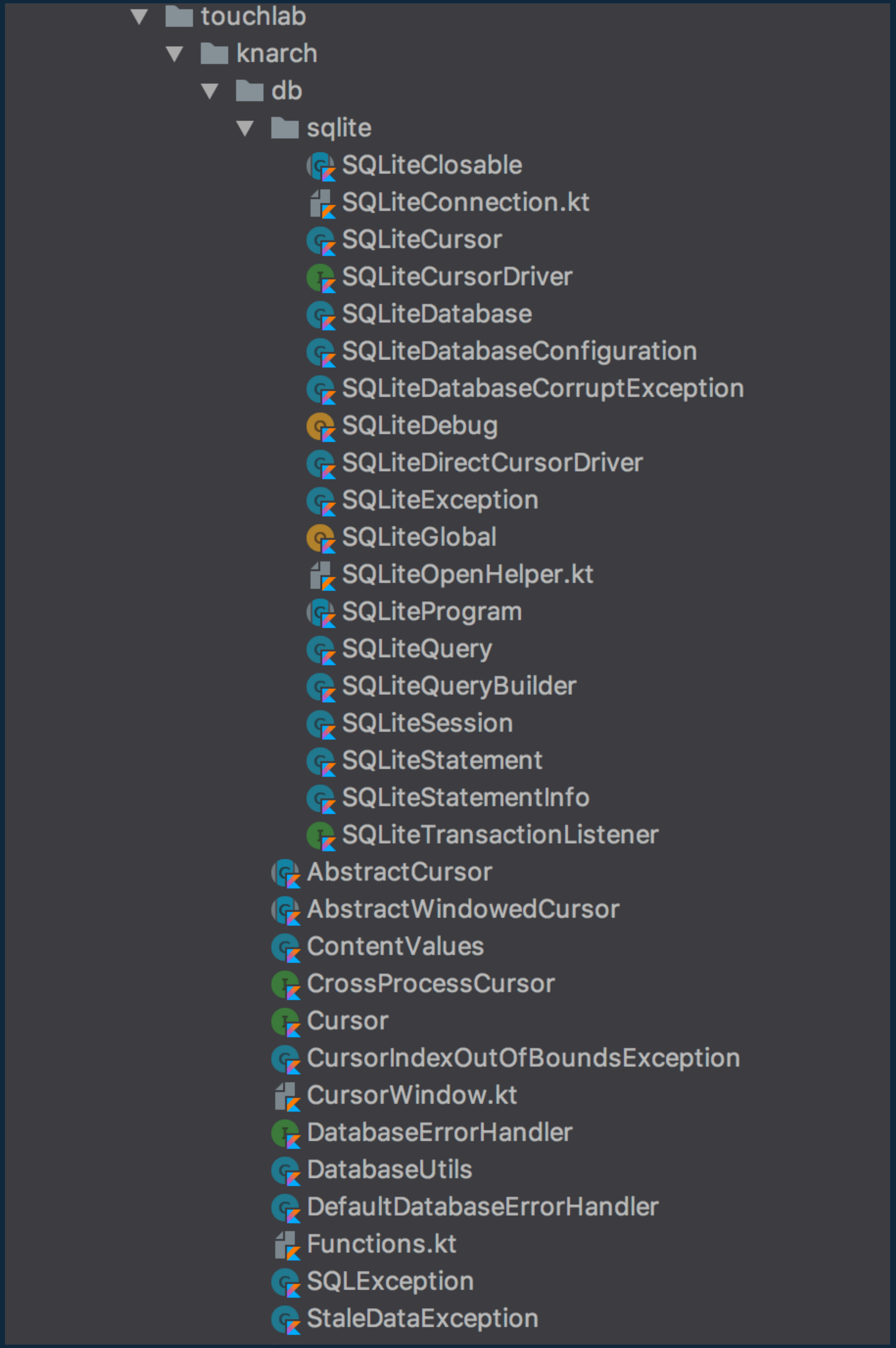
3 participants

gwwdfsad referenced this issue on Jul 31

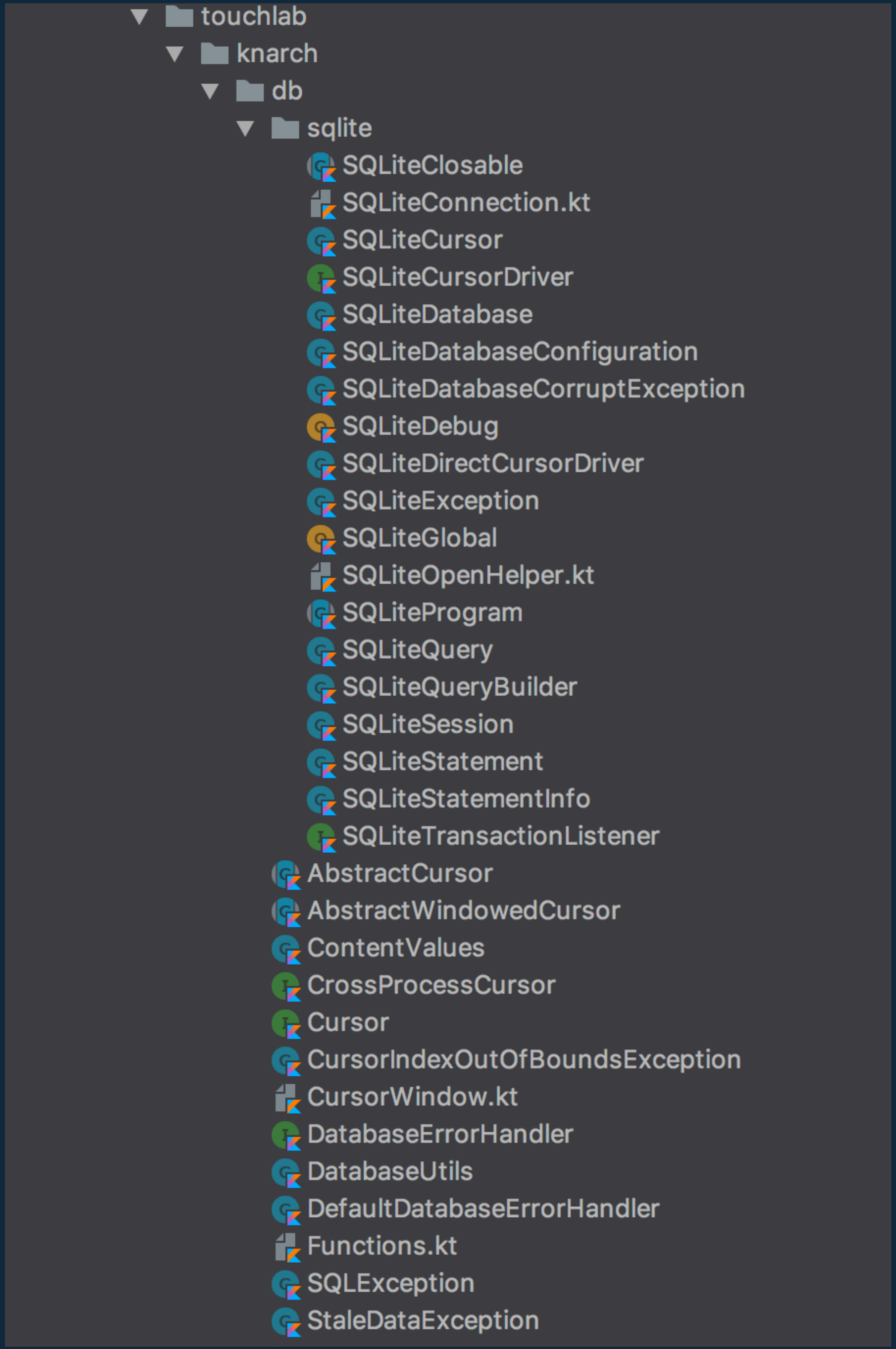
KNArch.db

Kotlin Native Architecture - Database

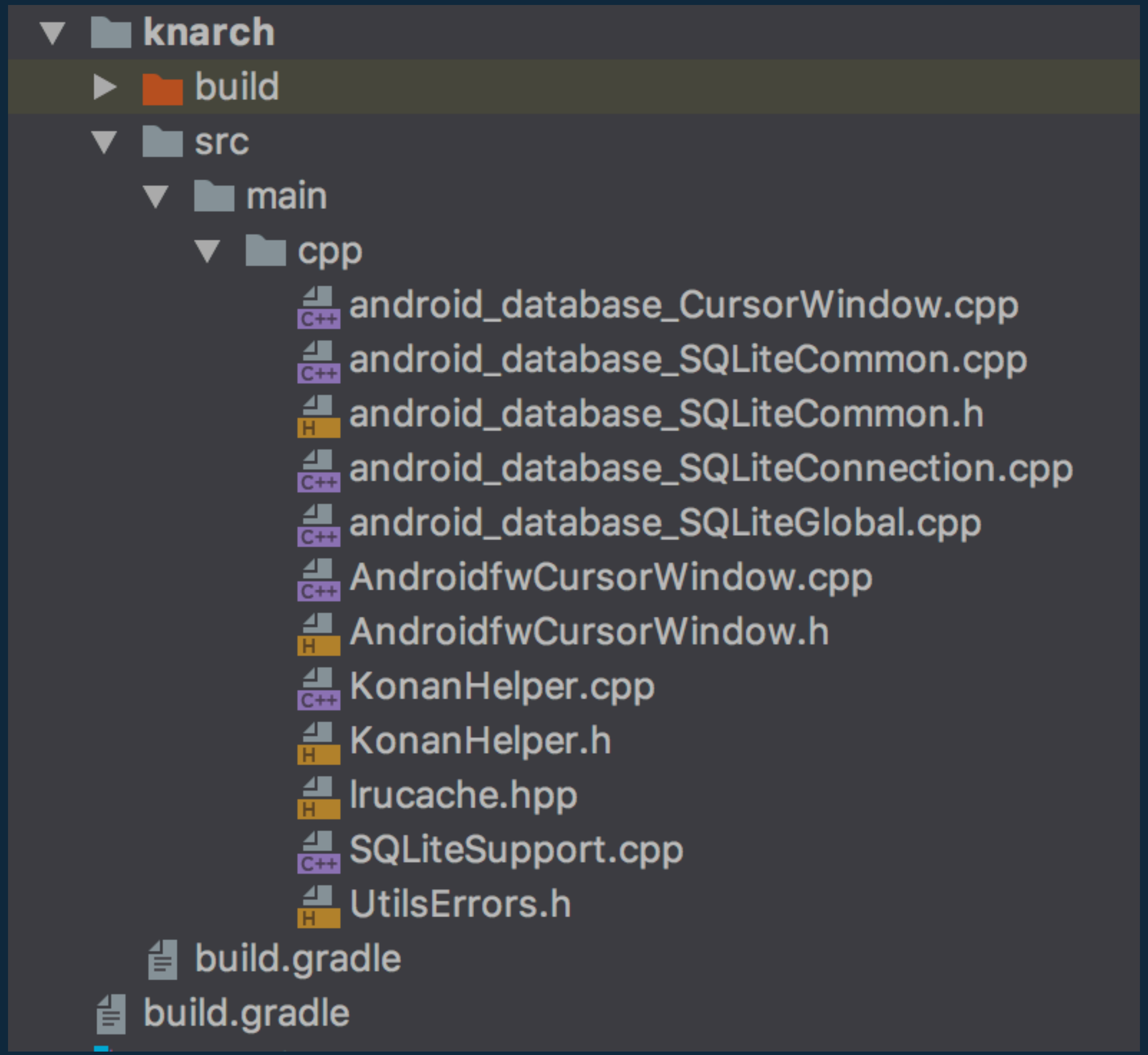
https://github.com/touchlab/knarch.db



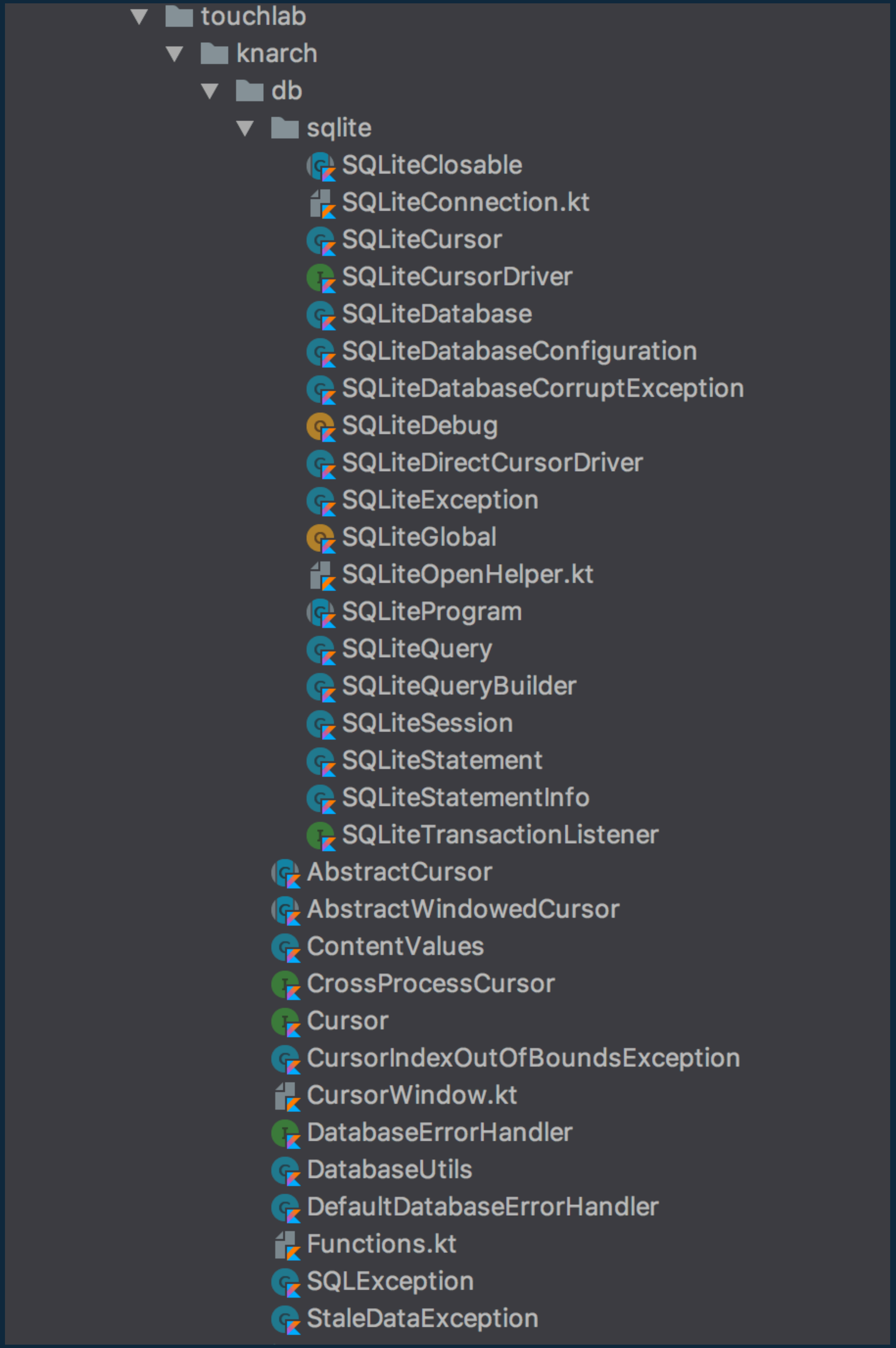
```
▼ touchlab
  ▼ knarch
    ▼ db
      ▼ sqlite
          SQLiteClosable
          SQLiteConnection.kt
          SQLiteCursor
          SQLiteCursorDriver
          SQLiteDatabase
          SQLiteDatabaseConfiguration
          SQLiteDatabaseCorruptException
          SQLiteDebug
          SQLiteDirectCursorDriver
          SQLiteException
          SQLiteGlobal
          SQLiteOpenHelper.kt
          SQLiteProgram
          SQLiteQuery
          SQLiteQueryBuilder
          SQLiteSession
          SQLiteStatement
          SQLiteStatementInfo
          SQLiteTransactionListener
        AbstractCursor
        AbstractWindowedCursor
        ContentValues
        CrossProcessCursor
        Cursor
        CursorIndexOutOfBoundsException
        CursorWindow.kt
        DatabaseErrorHandler
        DatabaseUtils
        DefaultDatabaseErrorHandler
        Functions.kt
        SQLException
        StaleDataException
```



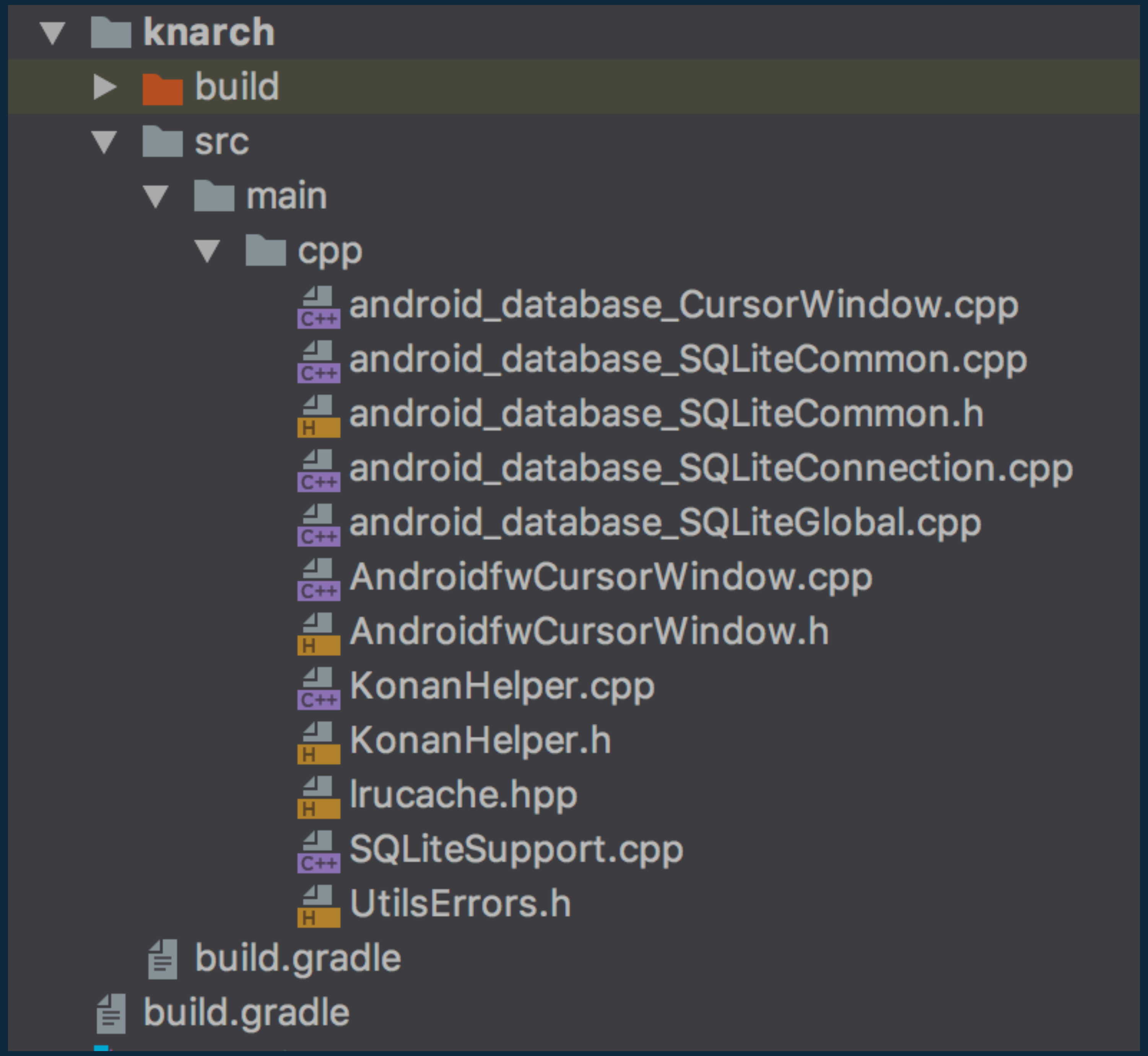
```
▼ touchlab
  ▼ knarch
    ▼ db
      ▼ sqlite
          SQLiteClosable
          SQLiteConnection.kt
          SQLiteCursor
          SQLiteCursorDriver
          SQLiteDatabase
          SQLiteDatabaseConfiguration
          SQLiteDatabaseCorruptException
          SQLiteDebug
          SQLiteDirectCursorDriver
          SQLiteException
          SQLiteGlobal
          SQLiteOpenHelper.kt
          SQLiteProgram
          SQLiteQuery
          SQLiteQueryBuilder
          SQLiteSession
          SQLiteStatement
          SQLiteStatementInfo
          SQLiteTransactionListener
        AbstractCursor
        AbstractWindowedCursor
        ContentValues
        CrossProcessCursor
        Cursor
        CursorIndexOutOfBoundsException
        CursorWindow.kt
        DatabaseErrorHandler
        DatabaseUtils
        DefaultDatabaseErrorHandler
        Functions.kt
        SQLException
        StaleDataException
```



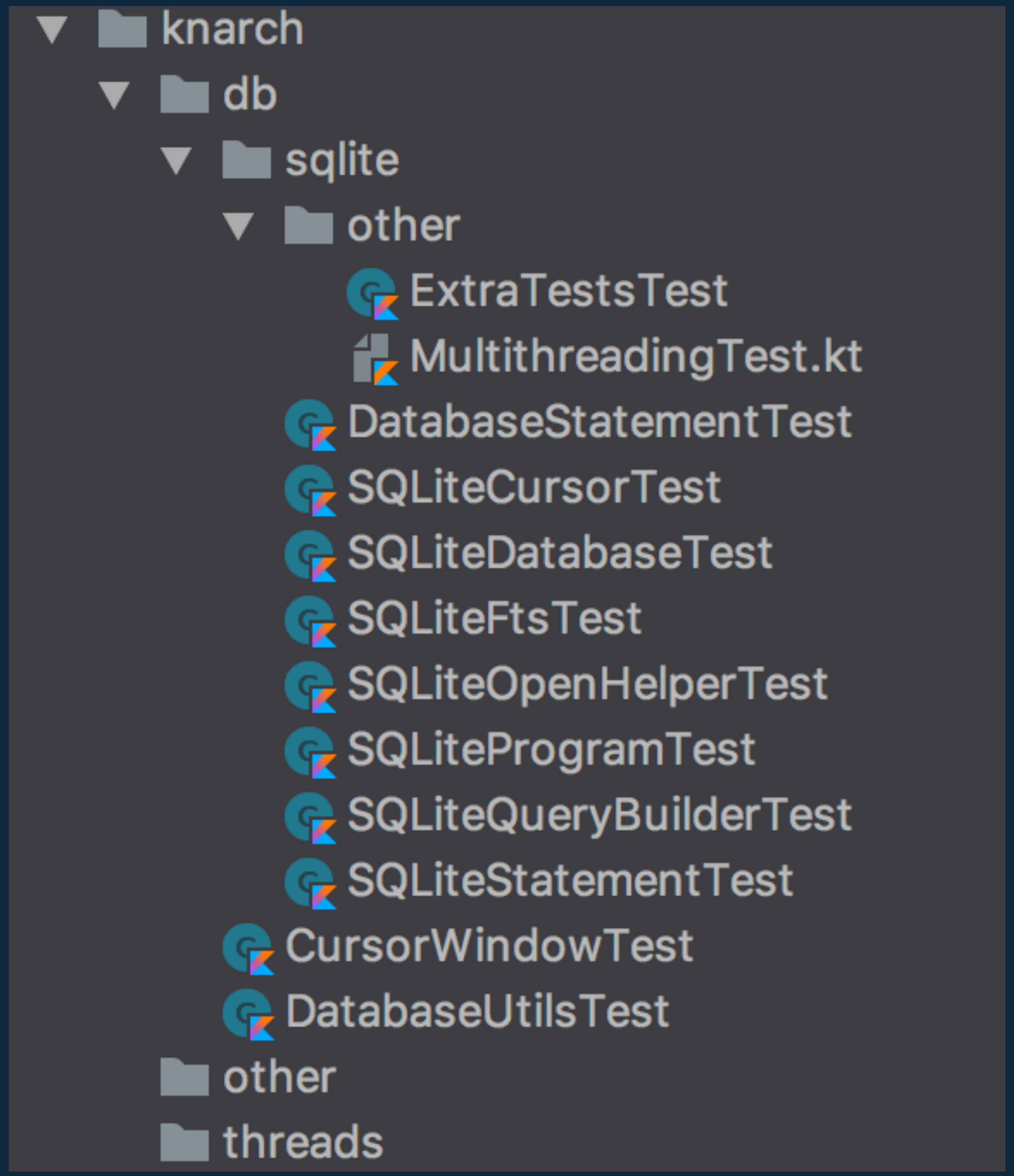
```
▼ knarch
  ▶ build
  ▼ src
    ▼ main
      ▼ cpp
          android_database_CursorWindow.cpp
          android_database_SQLiteCommon.cpp
          android_database_SQLiteCommon.h
          android_database_SQLiteConnection.cpp
          android_database_SQLiteGlobal.cpp
          AndroidfwCursorWindow.cpp
          AndroidfwCursorWindow.h
          KonanHelper.cpp
          KonanHelper.h
          lrucache.hpp
          SQLiteSupport.cpp
          UtilsErrors.h
    build.gradle
  build.gradle
```



```
▼ touchlab
  ▼ knarch
    ▼ db
      ▼ sqlite
          SQLiteClosable
          SQLiteConnection.kt
          SQLiteCursor
          SQLiteCursorDriver
          SQLiteDatabase
          SQLiteDatabaseConfiguration
          SQLiteDatabaseCorruptException
          SQLiteDebug
          SQLiteDirectCursorDriver
          SQLiteException
          SQLiteGlobal
          SQLiteOpenHelper.kt
          SQLiteProgram
          SQLiteQuery
          SQLiteQueryBuilder
          SQLiteSession
          SQLiteStatement
          SQLiteStatementInfo
          SQLiteTransactionListener
        AbstractCursor
        AbstractWindowedCursor
        ContentValues
        CrossProcessCursor
        Cursor
        CursorIndexOutOfBoundsException
        CursorWindow.kt
        DatabaseErrorHandler
        DatabaseUtils
        DefaultDatabaseErrorHandler
        Functions.kt
        SQLException
        StaleDataException
```



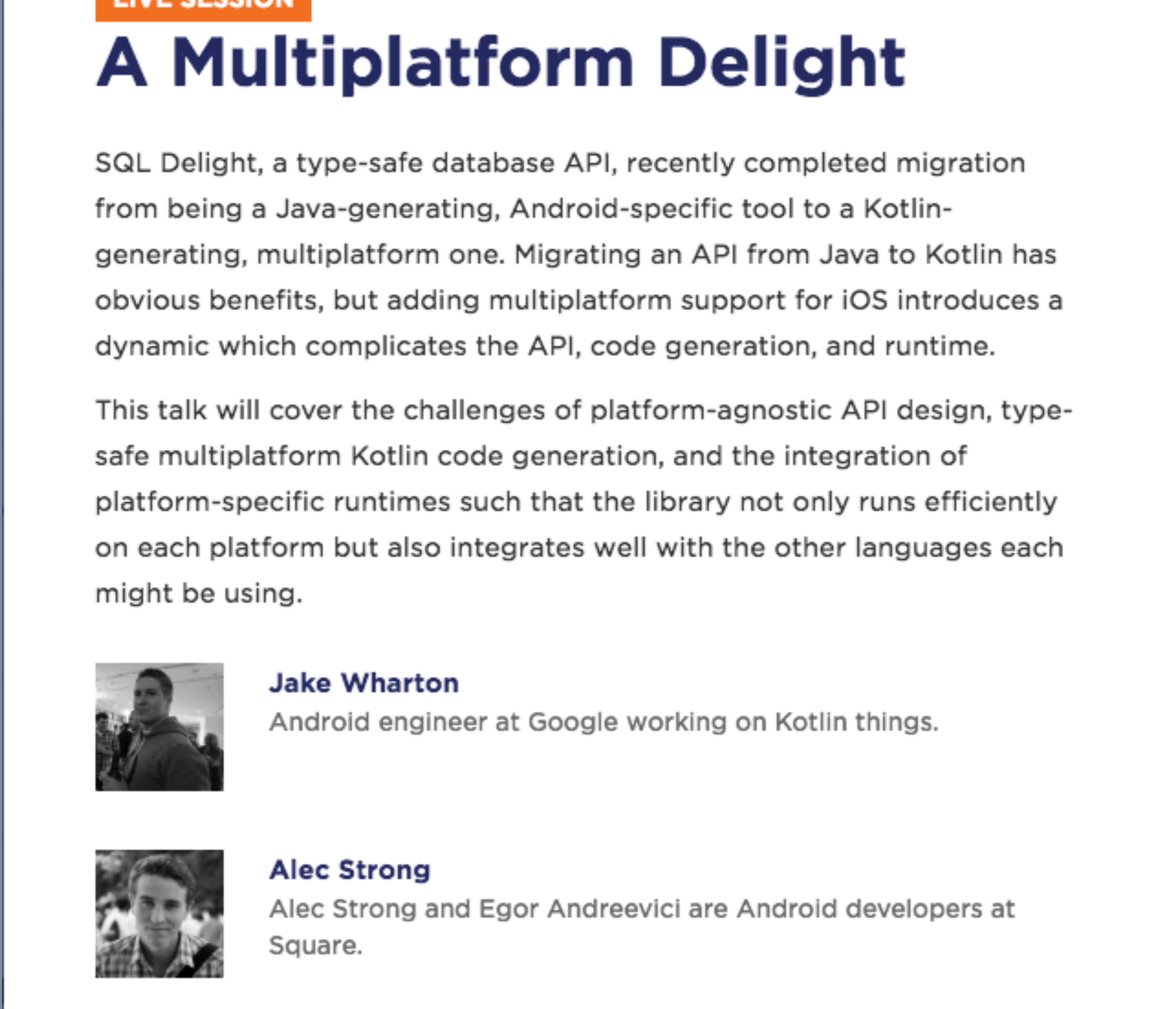
```
▼ knarch
  ▶ build
  ▼ src
    ▼ main
      ▼ cpp
          android_database_CursorWindow.cpp
          android_database_SQLiteCommon.cpp
          android_database_SQLiteCommon.h
          android_database_SQLiteConnection.cpp
          android_database_SQLiteGlobal.cpp
          AndroidfwCursorWindow.cpp
          AndroidfwCursorWindow.h
          KonanHelper.cpp
          KonanHelper.h
          lrucache.hpp
          SQLiteSupport.cpp
          UtilsErrors.h
    build.gradle
  build.gradle
```



```
▼ knarch
  ▼ db
    ▼ sqlite
      ▼ other
          ExtraTestsTest
          MultithreadingTest.kt
        DatabaseStatementTest
        SQLiteCursorTest
        SQLiteDatabaseTest
        SQLiteFtsTest
        SQLiteOpenHelperTest
        SQLiteProgramTest
        SQLiteQueryBuilderTest
        SQLiteStatementTest
      CursorWindowTest
      DatabaseUtilsTest
    other
    threads
```

Future Changes

- Add multithreaded reads and WAL support
- Coroutines aware api
- CursorWindow?
- Other stuff



LIVE SESSION

A Multiplatform Delight

SQL Delight, a type-safe database API, recently completed migration from being a Java-generating, Android-specific tool to a Kotlin-generating, multiplatform one. Migrating an API from Java to Kotlin has obvious benefits, but adding multiplatform support for iOS introduces a dynamic which complicates the API, code generation, and runtime.

This talk will cover the challenges of platform-agnostic API design, type-safe multiplatform Kotlin code generation, and the integration of platform-specific runtimes such that the library not only runs efficiently on each platform but also integrates well with the other languages each might be using.

Jake Wharton
Android engineer at Google working on Kotlin things.

Alec Strong
Alec Strong and Egor Andreevici are Android developers at Square.

```
CREATE TABLE session(
id TEXT NOT NULL PRIMARY KEY,
title TEXT NOT NULL,
description TEXT NOT NULL,
startsAt TEXT AS Date NOT NULL,
endsAt TEXT AS Date NOT NULL,
serviceSession INTEGER NOT NULL DEFAULT 0,
rsvp INTEGER NOT NULL DEFAULT 0,
roomId INTEGER,
FOREIGN KEY (roomId) REFERENCES room(id)
);

insert:
INSERT INTO session(id, title, description, startsAt, endsAt, serviceSession, roomId)
VALUES (?,?,?,?,?,?,?)
;

update:
UPDATE session SET title = ?, description = ?, startsAt = ?,
endsAt = ?, serviceSession = ?, roomId = ?, rsvp = ?
WHERE id = ?;

deleteById:
DELETE FROM session WHERE id = ?;

allSessions:
SELECT * FROM session;

sessionById:
SELECT * FROM session WHERE id = ?;
```

```sql
--Special query for schedule view
sessionWithRoom:
SELECT session.id, session.title, session.description, session.startsAt,
session.endsAt,
session.serviceSession, session.rsvp, session.roomId, room.name AS roomName,
speakers.allNames
FROM session
LEFT JOIN (
SELECT sessionId,group_concat(fullName, ', ') AS allNames
FROM sessionSpeaker
JOIN userAccount ON userAccount.id = sessionSpeaker.userAccountId
GROUP BY sessionId
) AS speakers ON speakers.sessionId = session.id
JOIN room ON session.roomId = room.id
;
```

```
--Special query for schedule vie
sessionWithRoom:
SELECT session.id, session.title                              tartsAt, session.en
session.serviceSession, session.                              AS roomName,
speakers.allNames
FROM session
LEFT JOIN (
SELECT sessionId,group_concat(fu
FROM sessionSpeaker
JOIN userAccount ON userAccount.                              d
GROUP BY sessionId
) AS speakers ON speakers.sessio
JOIN room ON session.roomId = ro
;
```

```kotlin
interface SessionWithRoom {
    val id: String

    val title: String

    val description: String

    val startsAt: Date

    val endsAt: Date

    val serviceSession: Long

    val rsvp: Long

    val roomId: Long?

    val roomName: String

    val allNames: String?
```

```
/**
 * Provide for "ORM-like" associated query
 */
internal fun UserAccount.sessionsAsync(): Deferred<List<Session>> {
    val id = this.id
    return async(ApplicationDispatcher) { AppContext.dbHelper.queryWrapper.sessionQueries.user
}

internal fun Session.roomAsync(): Deferred<Room> {
    val id = this.roomId!!
    return async(ApplicationDispatcher) {
        AppContext.dbHelper.queryWrapper.
                roomQueries.selectById(id).executeAsOne()
    }
}
```

SQLDelight + KNArch.db

KNArch.threads

Kotlin Native Architecture - Threads

https://github.com/touchlab/knarch.threads

KNArch.threads

- Temporary-ish until better tools emerge
- Atomic support (deprecated)
- ThreadLocal
- LiveData

Kotlin / kotlinx.coroutines
Watch 177
Star 3,062
Fork 384
Code
Issues 87
Pull requests 14
Projects 0
Wiki
Insights
New issue
Open
Collaborator
Assignees
No one assigned
Labels
enhancement
Projects
None yet
Milestone
No milestone
Notifications
Subscribe
You're not receiving notifications from this thread.

Provide abstraction for cold streams #254

elizarov opened this issue on Feb 21 · 48 comments

elizarov commented on Feb 21

All the currently provided channel abstractions in `kotlinx.coroutines` are *hot*. The data is being produced regardless of the presence of subscriber. This is good for data sources and applications that are inherently hot, like incoming network and UI-events.

However, hot streams are not an ideal solution for cases where data stream is produced on demand. Consider, for example, the following simple code that produces `ReceiveChannel<Int>`:

```
produce<Int> {
    while (true) {
        val x = computeNextValue()
        send(x)
    }
}
```

One obvious downside is the `computeNextValue()` is invoked before `send`, so even when receiver is not ready, the next value gets computed. Of course, it gets suspended in `send` if there is no receiver, but it is not as lazy as you get with cold reactive Publisher/Observable/Flowable/Flux/Flow.

We need the abstraction for cold streams in `kotlinx.coroutines` that is going to be just as lazy

multiplatform ✕

KT-19848 Created by Leonid Startev a year ago Updated by Stanislav Erokhin a year ago Visible to All Users

★ Multiplatform projects: incompatibility between Kotlin's protected visibility and Java's protected visibility

Subtask of: KT-17909 ×

I'm trying to make common abstraction over `java.io.Writer` class. Given the

```
header abstract class Writer protected constructor()
```

in common module and

```
impl typealias Writer = java.io.Writer
```

in jvm module, it fails to compile with:

```
The following declaration is incompatible because some members are not implement
    public typealias Writer = Writer
No implementations are found for members listed below:

    protected constructor Writer()

    The following declaration is incompatible because visibility is different:
        protected/*protected and package*/ constructor Writer()
    The following declaration is incompatible because number of value parameters
        protected/*protected and package*/ constructor Writer(p0: Any!)
```

Seems currently there is no way to implement Kotlin's protected functions with Java's protected functions, because Java's protected visibility is wider and includes also package scope. But it shouldn't

Priority	Major
Type	Problem
Target versions	No Target versions
State	Spec Needed
Assignee	Alexander Udalov
Subsystems	Frontend. Declarations
Affected versions	1.1.4
Tester (Kotlin)	No tester
Change processed	No

Multiplatform Settings

Really Shared Preferences

https://github.com/russhwolf/multiplatform-settings

```kotlin
public expect class PlatformSettings : Settings {

    /**
     * A factory that can produce [Settings] instances.
     */
    public class Factory : Settings.Factory {
        public override fun create(name: String?): Settings
    }

    public override fun clear()
    public override fun remove(key: String)
    public override fun hasKey(key: String): Boolean
    public override fun putInt(key: String, value: Int)
    public override fun getInt(key: String, defaultValue: Int): Int
    public override fun putLong(key: String, value: Long)
    public override fun getLong(key: String, defaultValue: Long): Long
    public override fun putString(key: String, value: String)
    public override fun getString(key: String, defaultValue: String): String
    public override fun putFloat(key: String, value: Float)
    public override fun getFloat(key: String, defaultValue: Float): Float
    public override fun putDouble(key: String, value: Double)
    public override fun getDouble(key: String, defaultValue: Double): Double
    public override fun putBoolean(key: String, value: Boolean)
    public override fun getBoolean(key: String, defaultValue: Boolean): Boolean
}
```

Timber

Multiplatform logging

https://github.com/JakeWharton/timber

Atomic Fu

Atomic operation support

https://github.com/Kotlin/kotlinx.atomicfu

Kotlinx.io

multiplatform I/O library

https://github.com/Kotlin/kotlinx-io

OKIO2 MULTIPLATFORM

Jesse Wilson
Android and jokes.
Aug 27 · 3 min read

Announcing Okio 2: Our fast + simple I/O library, Okio, has a new release that supports Kotlin.

MY WISH LIST

Stable Gradle Plugins

I know, but 😢

Significant Library Examples

With publishing, for all targets

Multithreaded Native Coroutines

If I get 1 thing for Christmas...

A Reactive Library

Or maybe just coroutines?

Xcode Debugging?

Asking a lot, but still

COMMUNITY WISH LIST

Mocking Library

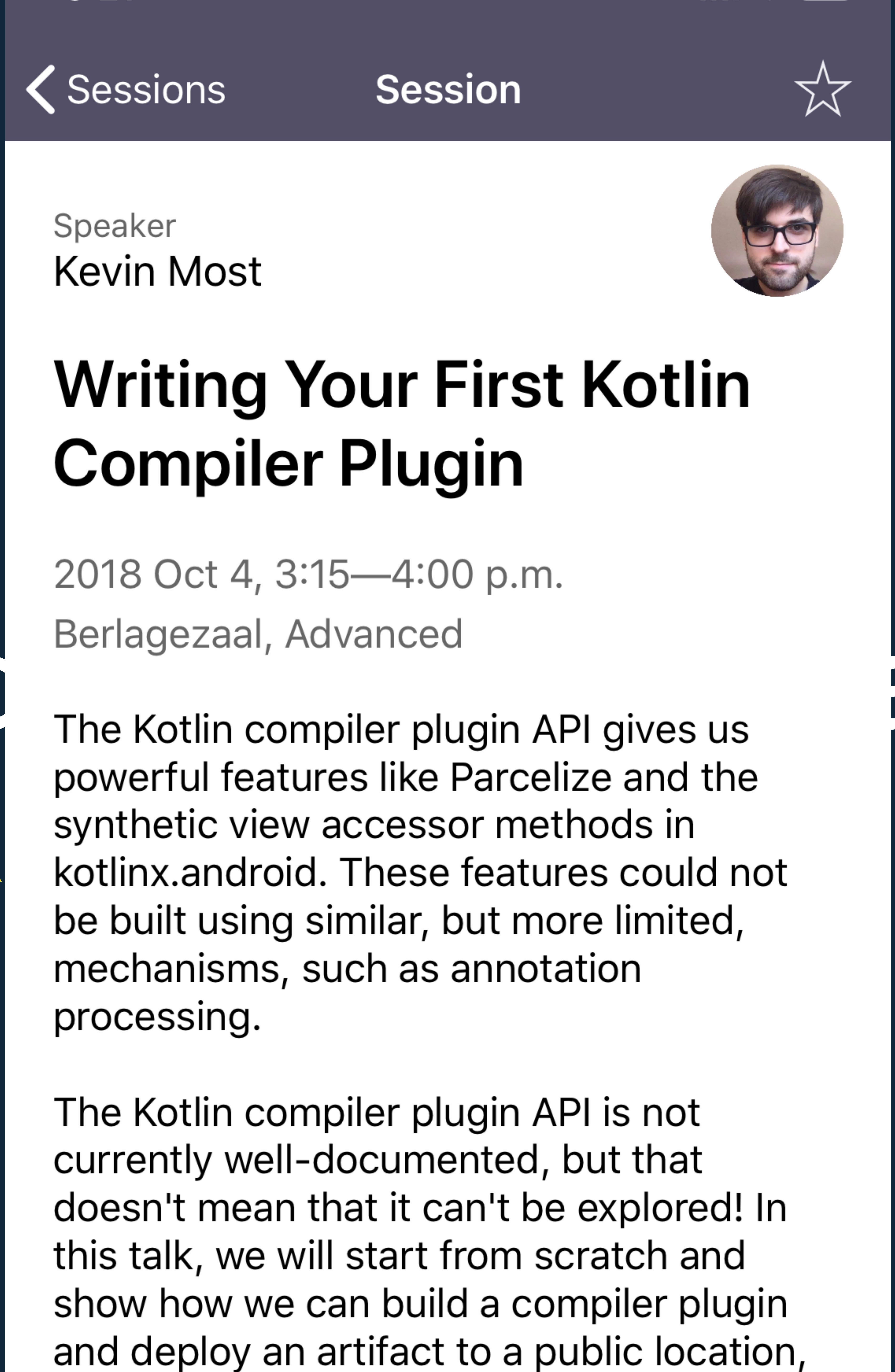
See mockk repo

Dependency Injection

Or service locator I guess...

Buil ation

K s



Speaker
Kevin Most

Writing Your First Kotlin Compiler Plugin

2018 Oct 4, 3:15—4:00 p.m.

Berlagezaal, Advanced

The Kotlin compiler plugin API gives us powerful features like Parcelize and the synthetic view accessor methods in kotlinx.android. These features could not be built using similar, but more limited, mechanisms, such as annotation processing.

The Kotlin compiler plugin API is not currently well-documented, but that doesn't mean that it can't be explored! In this talk, we will start from scratch and show how we can build a compiler plugin and deploy an artifact to a public location,

Date Support

JSR 310 or similar

UI Stuff

Here be dragons

Getting Started

Build Samples

Conference apps, several others

Kotlin/Native Docs

Learn threads and state

For Libraries?

Multiplatform Settings (then the rest)

Join the Kotlin Slack

When?

v0.9
Coroutines
(and other libs)

.6

v0.7

v0.8

v0.8.2

Q2

Q3

Q4

Q1

Q2

2018

2019

TOUCHLAB

v0.9.3

Coroutines?

.6
v0.7
v0.8
v0.8.2
Q2
Q3
Q4
Q1
Q2
2018
2019

TOUCHLAB

v0.9.3

Coroutines?
Other Libraries
Docs/tutorials

v0.7

v0.8.2

.6

v0.8

Q2 Q3 Q4 Q1 Q2

2018

2019

Thanks for the Images!

Source Links at: https://bit.ly/2O0c469

TOUCHLAB

kevin@touchlab.co

@kpgalligan

Join the team!

TOUCHLAB

kevin@touchlab.co

@kpgalligan