USETECH

Swift для Android и кроссплатформы. Разбираем на практике

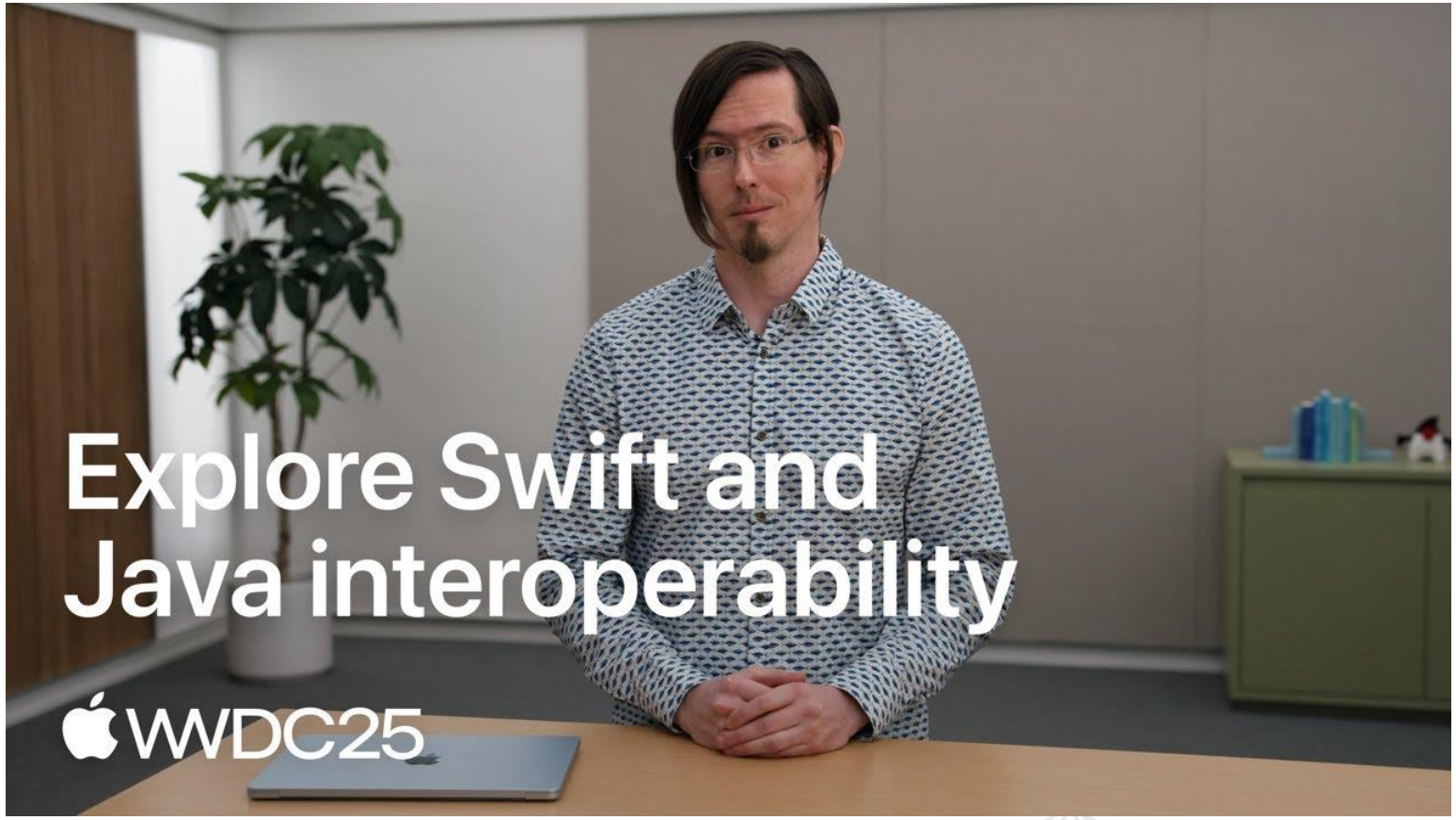
Анна Жаркова
Lead Mobile developer

Обо мне

- В мобильной разработке с 2013
- Ведущий мобильный разработчик в Usetech
- Нативная разработка под iOS и Android (Swift/Objective-C, Kotlin/Java) кросс-платформа (Xamarin, Kotlin multiplatform)
- Ментор, управляю командой направления
- Спикер на конференциях AppsConf, Mobius, TechTrain, DroidCon (2022)
- Преподаватель в Otus (iOS Pro и базовый)
- Автор статей по мобильной разработке (SwiftUI, iOS, KMM)
- Автор книги «Kotlin Multiplatform на практике»

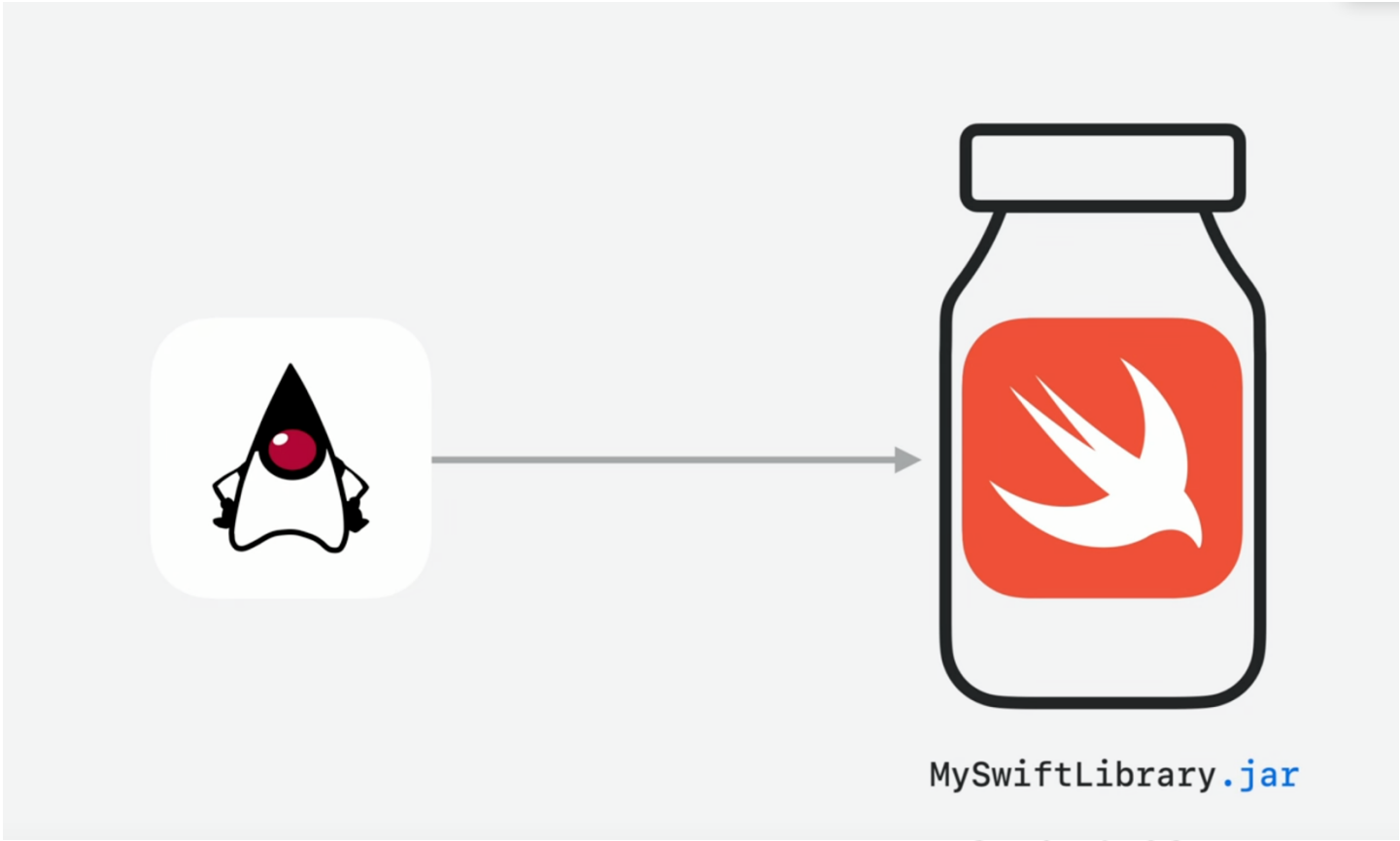
Swift-Java Interoperability. WWDC 2025

https://github.com/swiftlang/swift-java



Swift-Java Interoperability. WWDC 2025

https://github.com/swiftlang/swift-java



Swift-Java Interoperability. WWDC 2025

Swift Package

- **JavaKit** – Swift libraries and macros

Java Library

- **SwiftKit** – Java library for efficient Swift interoperability

Кросс-платформа для натива

ДОКЛАД 19.11 / 19:30 – 20:15 (UTC+7) Under the Hood

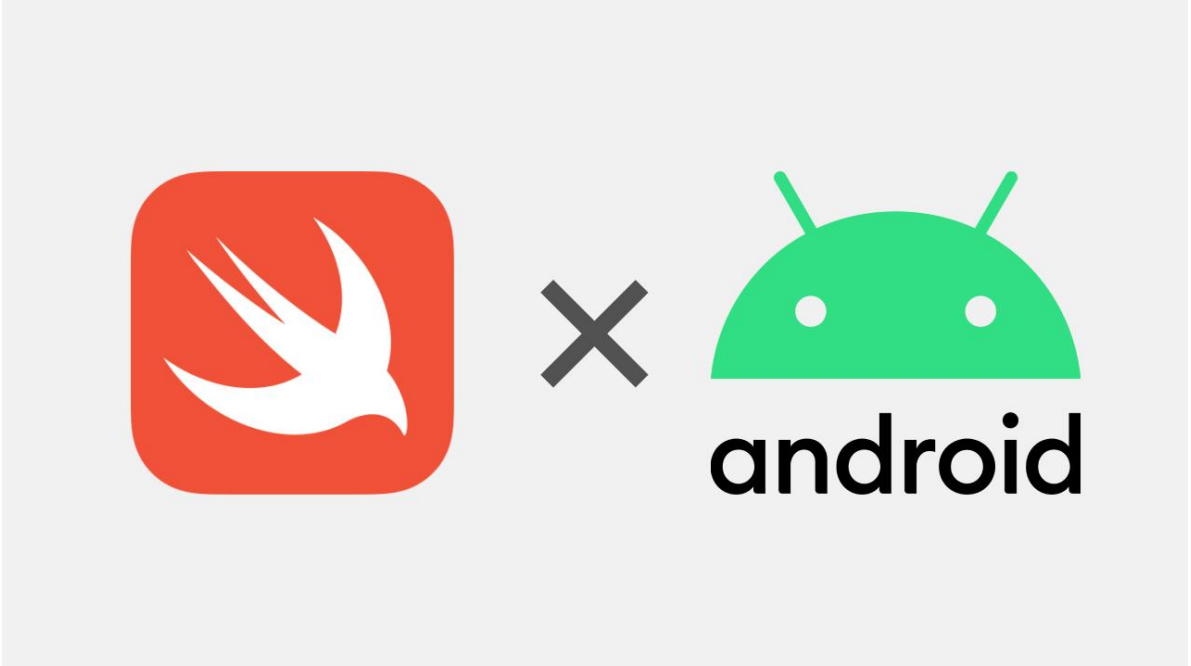
Java-Swift interop vs Kotlin-Swift export. Путь к сингулярности

Спикеры

Анна Жаркова
ГК Юзтех

Swift 6.3 Официальный релиз SDK для Android

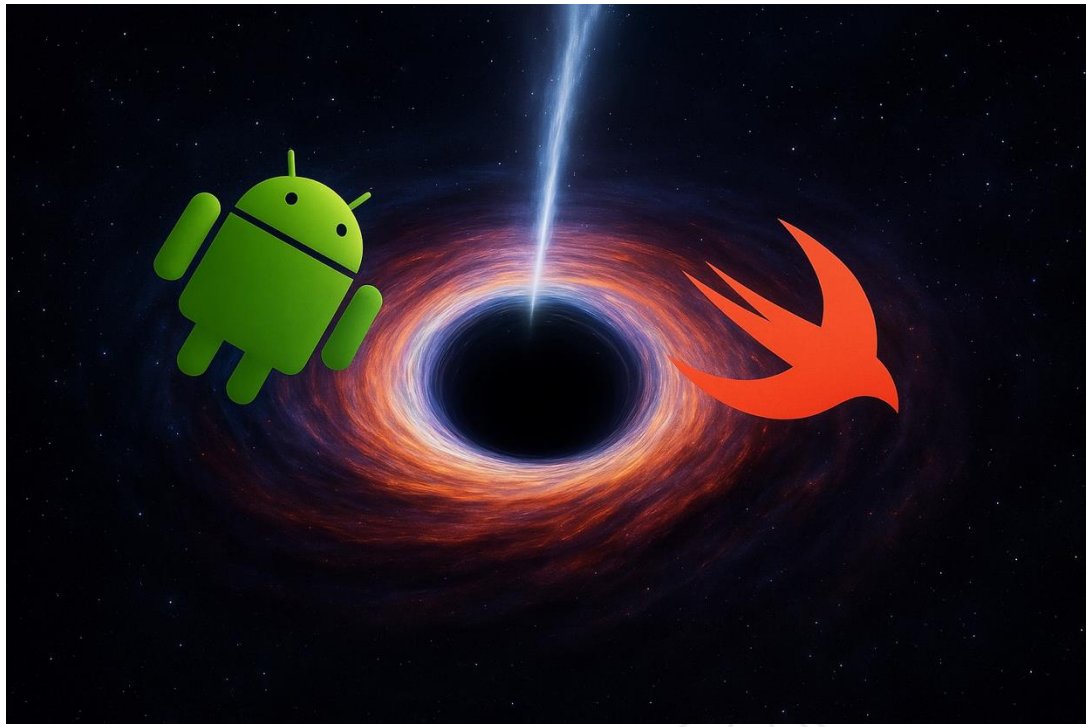
https://www.swift.org/documentation/articles/swift-sdk-for-android-getting-started.html



Обсудим:

- Swift-Java 2026, что изменилось
- Базовая архитектура решения в Android приложении
- Особенности интеропа
- Работаем с памятью и многопоточностью
- Работаем с UI ???

Swift-> Java для Android, как работает



JavaKit RIP. Swift->Java

Основное направление развития

- https://github.com/swiftlang/swift-java/

Generates hard-to-get-right code.

Генерирует код, который сложно написать правильно.

SwiftJava. Что улучшилось

- Интеграция с Java и Kotlin, Interop типов
- Кросс-компиляция
- Компиляция в нативные бинарные файлы для Android
- Интеграция с Gradle
- Поддержка отладки

Проблемы SwiftJava

- JNIExtract produces code incompatible with .enableUpcomingFeature("InternalImportsByDefault") feature:jextract
  #705 · by tkrajacic was closed 2 weeks ago — 1 — 2
- `IfConfigDecl` support broke Android builds android
  #676 · by madsodgaard was closed last month — 6
- error: the target 'SwiftJava' in product 'JExtractSwiftPlugin' contains unsafe build flags
  #636 · by gmondada was closed on Mar 28 — 6
- jextract: Evaluate `IfConfigDecl` for simple patterns feature:jextract
  #633 · by sidepelican was closed 3 weeks ago — 5
- Adding `JExtractSwiftPlugin` links Full Foundation build
  #619 · by madsodgaard was closed on Mar 17 — 1
- jextract/jni: Support native Swift Set use from Java feature:jextract mode:jni
  #612 · by ktoso was closed on Mar 17 — 1
- jextract/jni: Support native Swift Dictionary use from Java feature:jextract mode:jni
  #609 · by ktoso was closed on Mar 16 — 1
- Runtime crash on Android after `toString` refactorings android bug
  #604 · by madsodgaard was closed on Mar 11 — 1 — 3
- jextract/ffm: tuples support feature:jextract mode:ffm
  #603 · by ktoso was closed on Mar 19 — 1
- Discrepancy between files in generated/ directories and wrap-java output build
  #599 · by sidepelican was closed on Mar 18 — 2
- jextract/jni: tuple support feature:jextract mode:jni
  #596 · by ktoso was closed on Mar 10 — 1
- Add support for objects in `@JavaImplementation` feature:javakit-macros
  #595 · by madsodgaard was closed on Mar 10 — 2
- Use Android's api-versions.xml to inform availability of Android APIs android feature:wrap-java
  #592 · by ktoso was closed on Mar 9 — 1

110 пофикшенных тикетов

Проблемы SwiftJava

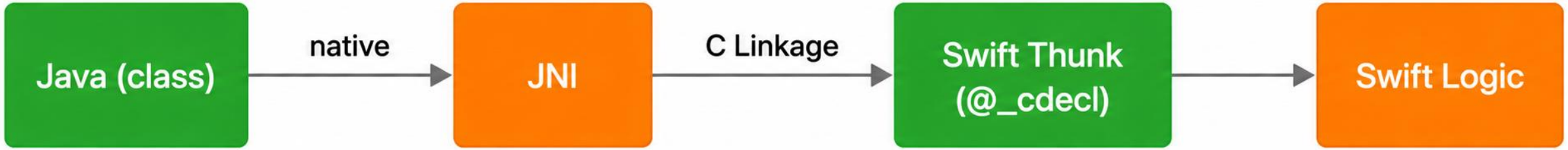
is:issue state:open | Labels | Milestones | New issue

Open 117 | Closed 110 | Author | Labels | Projects | Milestones | Assignees | Types | Newest

- **Multiple Swift targets, single dynamic library**
  #715 · mortenbekditlevsen opened 2 weeks ago
- **jextract: Support Foundation.Decimal** 2
  #711 · tkrajacic opened 2 weeks ago
- **JNIExtract can't handle type-name/module-name conflicts** feature:jextract 2
  #706 · tkrajacic opened 2 weeks ago
- **Building a package for Apple platform shouldn't use JExtractSwiftPlugin plugin and swift-java dependency at all**
  #694 · tkrajacic opened 3 weeks ago
- **Docs/guide about debugging crashes for java developers** documentation
  #683 · ktoso opened last month
- **Reenable nightly builds in CI**
  #681 · ktoso opened last month
- **Attempt to remove non-JNI local reference errors on Android** android 2
  #663 · lhoward opened on Apr 1
- **Generate version script file for FFM mode in jextract**
  #653 · madsodgaard opened on Mar 28
- **Move Foundation support into a common package**
  #638 · sidepelican opened on Mar 23
- **Switch wrap java to JEP467 markdown docs style**
  #637 · ktoso opened on Mar 21
- **Add CI job that checks/regenerates JavaStdlib etc with wrap-java** 3
  #628 · ktoso opened on Mar 18
- **jextract/ffm: Support native Swift Set use from Java** feature:jextract mode:ffm
  #611 · ktoso opened on Mar 12

117 открытых тикетов

SwiftJava под капотом



SwiftJava под капотом

- Java-привязки (Java Bindings): классы Java, которые отражают типы Swift и предоставляют объявления нативных методов.
- Swift-переходники (Swift Thunks): функции Swift с аннотацией @_cdecl, точки входа для вызовов JNI.
- Списки экспорта компоновщика (Linker Export Lists): скрипты, обеспечивающие экспорт только необходимых символов JNI в итоговой разделяемой библиотеке.

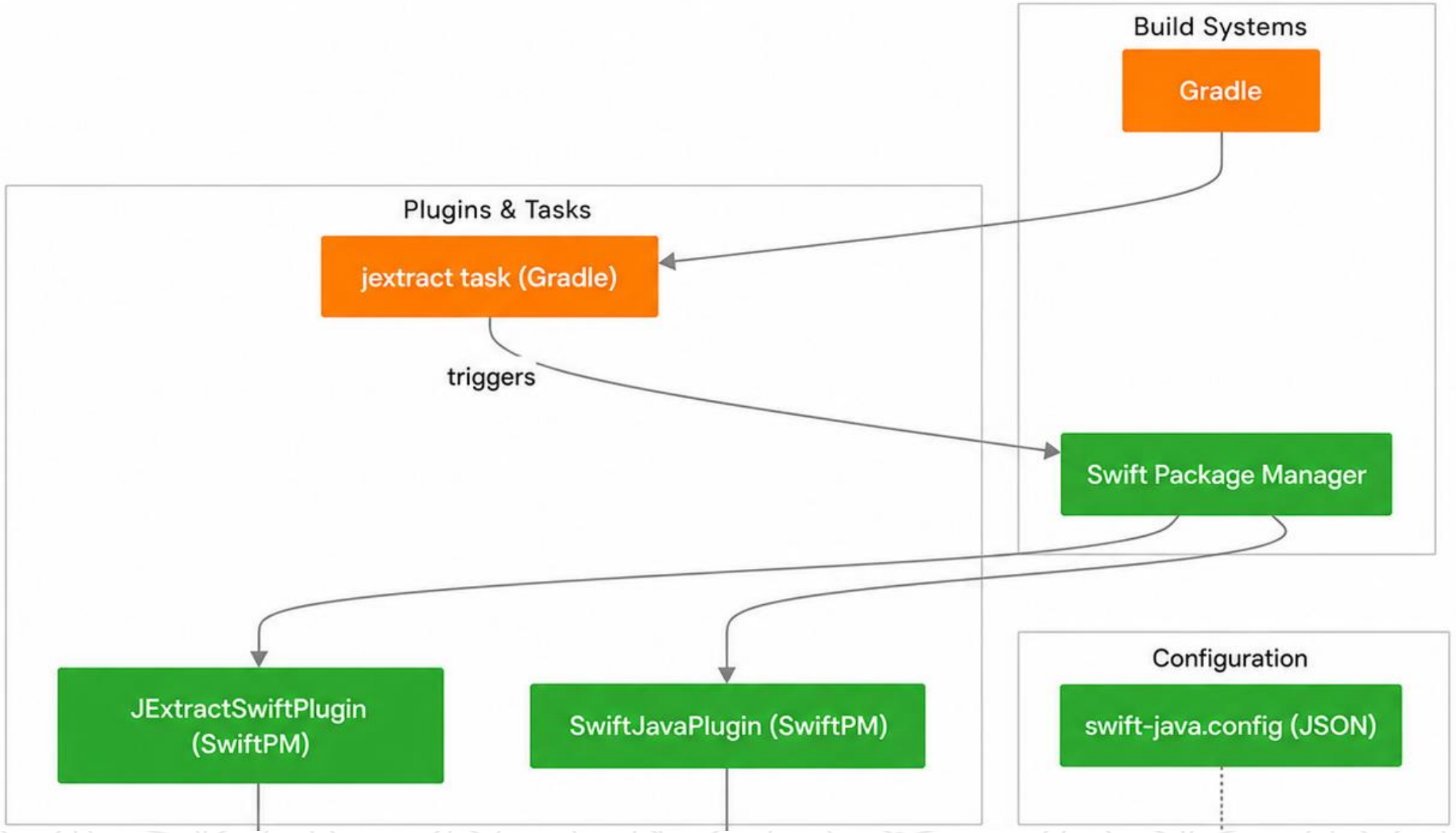
Swift -> Java

Jextract – Java код из Swift

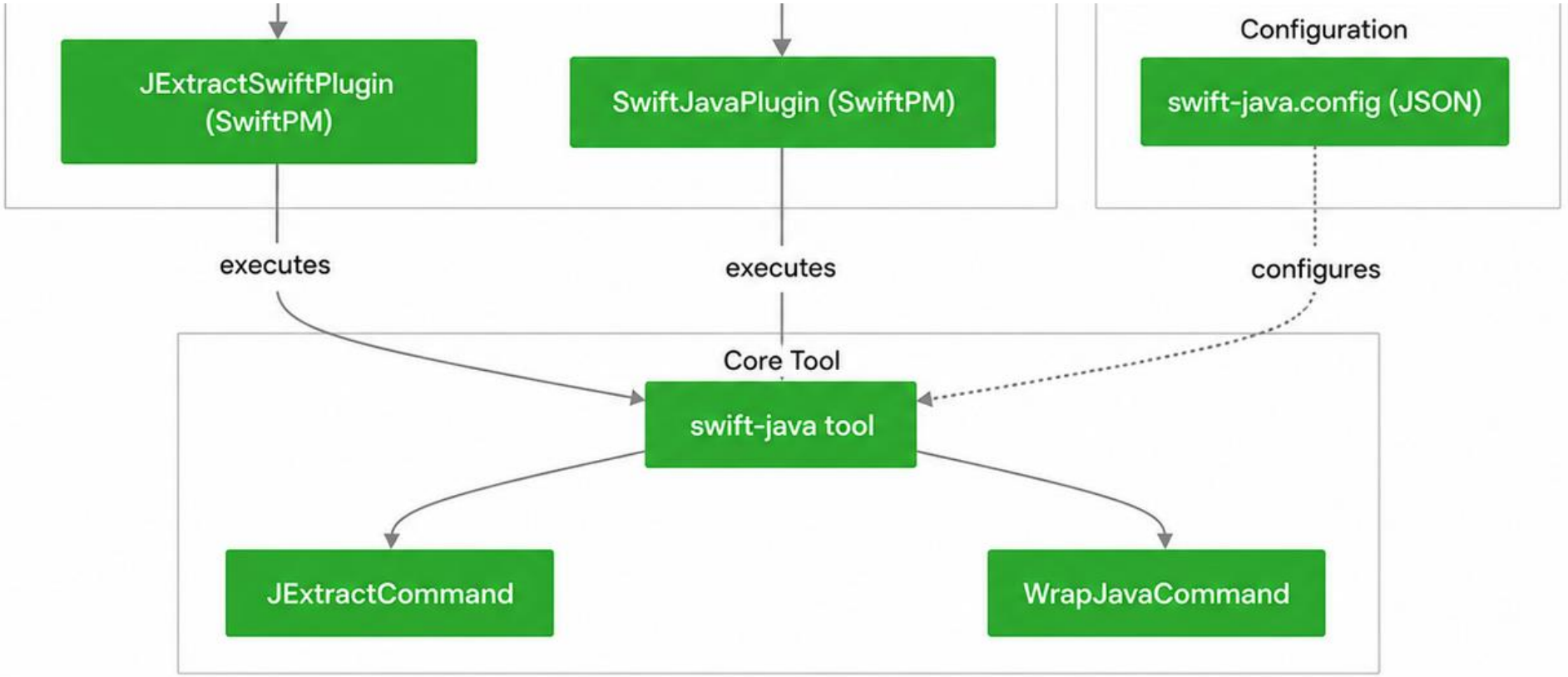
- ffm (Swift 6.2+, JDK 25)
- jni (Swift 6.2+, JDK 7+)

```
swift-java jextract --mode=ffm (default)

swift-java jextract --mode=jni
```

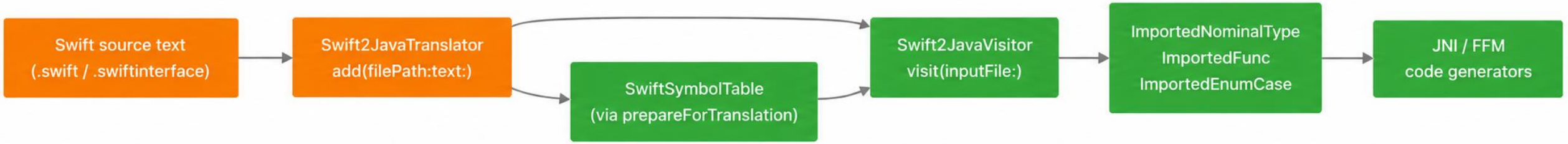
SwiftJava под капотом



SwiftJava под капотом



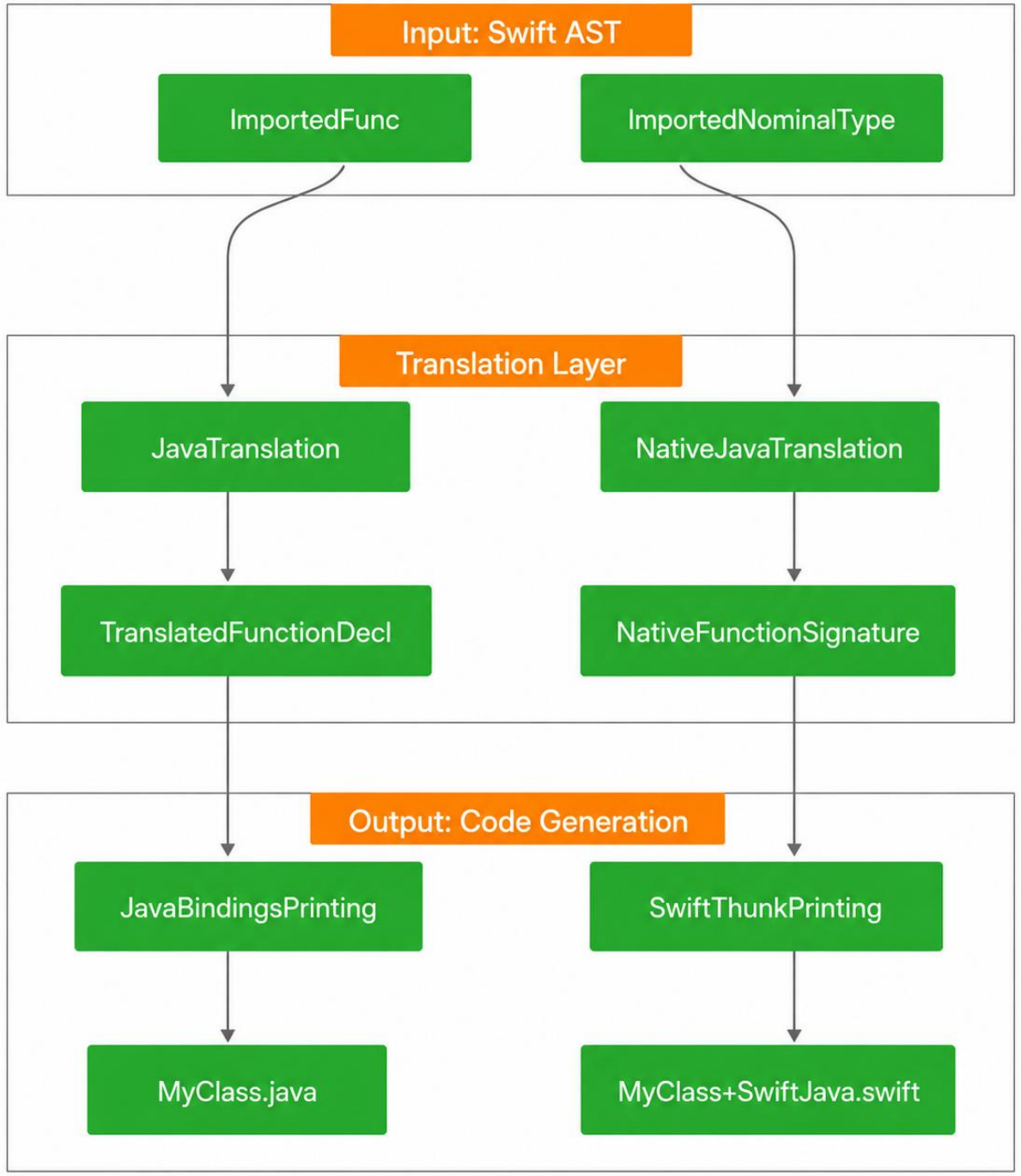
SwiftJava под капотом



SwiftJava под капотом

JNI поддерживает комплексные Swift-feature:

- async/await (CompletableFuture)
- перехват исключений
- Arc (через SwiftArena)



JNI vs FFM

JNI	FFM
Java 11+	Java 22+
Android Compability	For modern platforms
More copies	Better native memory support
Wider set of language features available	“Basic” features supported

JNI vs FFM

JEP 454: Foreign Function & Memory API

Owner	Maurizio Cimadamore
Type	Feature
Scope	SE
Status	Closed / Delivered
Release	22
Component	core-libs / java.lang.foreign
Discussion	panama dash dev at openjdk dot org
Relates to	JEP 472: Prepare to Restrict the Use of JNI
	JEP 442: Foreign Function & Memory API (Third Preview)
Reviewed by	Alex Buckley, Jorn Vernee
Endorsed by	Alan Bateman
Created	2023/06/22 09:36
Updated	2024/10/07 14:18
Issue	8310626

Summary

Introduce an API by which Java programs can interoperate with code and data outside of the Java runtime. By efficiently invoking foreign functions (i.e., code outside the JVM), and by safely accessing foreign memory (i.e., memory not managed by the JVM), the API enables Java programs to call native libraries and process native data without the brittleness and danger of JNI.

FFM с Android не работает

Работаем с JNI

Что поддерживаем

Jextract
https://swiftpackageindex.com/swiftlang/swift-java/main/documentation/swiftjavadocumentation/supportedfeatures

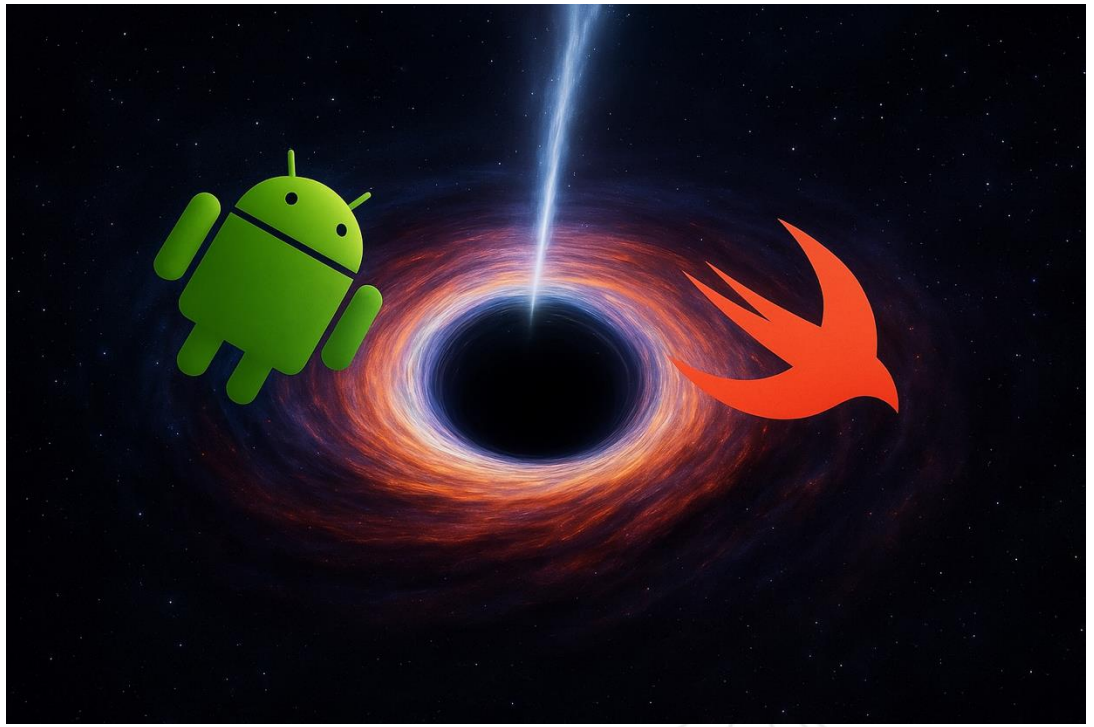
Swift Feature	FFM	JNI
Initializers: `class`, `struct`	✅	✅
Optional Initializers / Throwing Initializers	❌	✅
Deinitializers: `class`, `struct`	✅	✅
`enum`	❌	✅
`actor`	❌	❌
Global Swift `func`	✅	✅
Class/struct member `func`	✅	✅
Throwing functions: `func x() throws`	❌	✅
Typed throws: `func x() throws(E)`	❌	❌
Stored properties: `var`, `let` (with `willSet`, `didSet`)	✅	✅
Computed properties: `var` (incl. `throws`)	✅ / TODO	✅
Async functions `func async` and properties: `var { get async {} }`	❌	✅
Arrays: `[UInt8]`	✅	✅
Arrays: `[MyType]`, `Array<Int64>` etc	❌	✅
Dictionaries: `[String: Int]`, `[K:V]`	❌	✅
Generic type: `struct S<T>`	❌	✅
Functions or properties using generic type param: `struct S<T> { func f(_: T) {} }`	❌	❌

USE TECH

JExtract. Что улучшилось

- Поддержка исключений (JNI)
- Многопоточность (CompletableFuture, через JNI)
- Работа с generics
- Управление памятью (через SwiftArena)

Swift-> Java для Android на бою



Swift Android SDK. Было

Написать все приложение Android на Swift пока не получится

Does this mean I can write Android applications in Swift?

No. Although the Swift compiler is capable of compiling Swift code that runs on an Android device, it takes a lot more than just the Swift stdlib to write an app. You'd need some sort of framework to build a user interface for your application, which the Swift stdlib does not provide.

Swift Android SDK. Стало

Сложно, но возможно и не особо рекомендуется

Can I build a complete Android app entirely in Swift now?

Technically yes, but practically not recommended. You can write logic, networking, and data layers in Swift, but UI still requires Jetpack Compose or Android Views. There's no SwiftUI equivalent for Android in the official SDK.

Swift Android SDK. Что можно

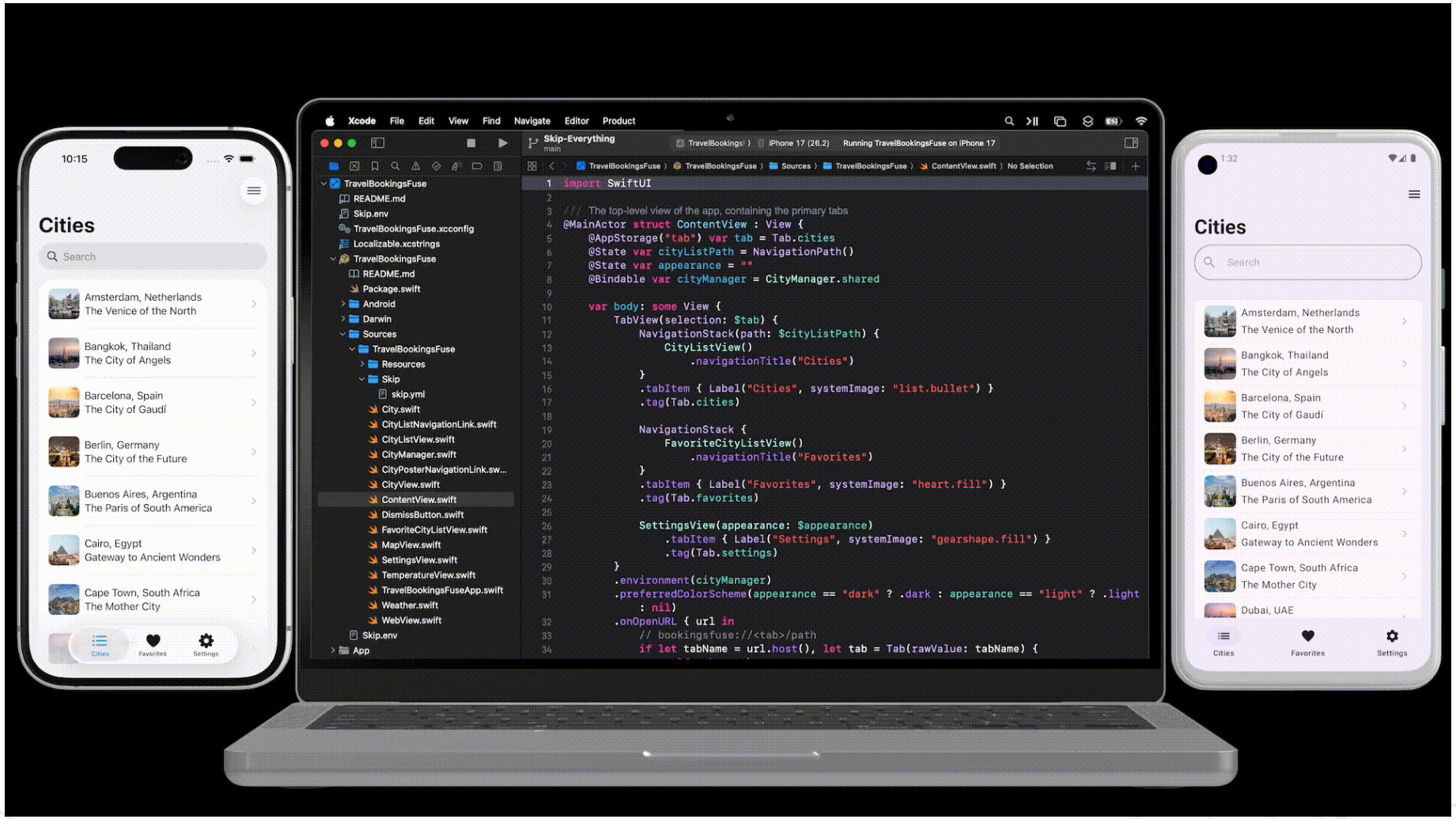
- Бизнес-логика
- Модели данных
- Сеть, хранилище
- Архитектура
- Фоновые процессы

Swift Android SDK. Что нельзя

- Общий UI (UI должен быть нативным - Jetpack Compose, Xml)
- SwiftUI (только на iOS и только со стороны нативного приложения)
- «Пиши один раз – запускай везде»

Swift Android SDK. Если нужен SwiftUI

Skip.Tools
https://skip.tools/



Swift Android SDK. Для чего

- Just for fun, пет-проекты, эксперименты, улучшение технологии...

- Расширение существующего iOS приложения на Android

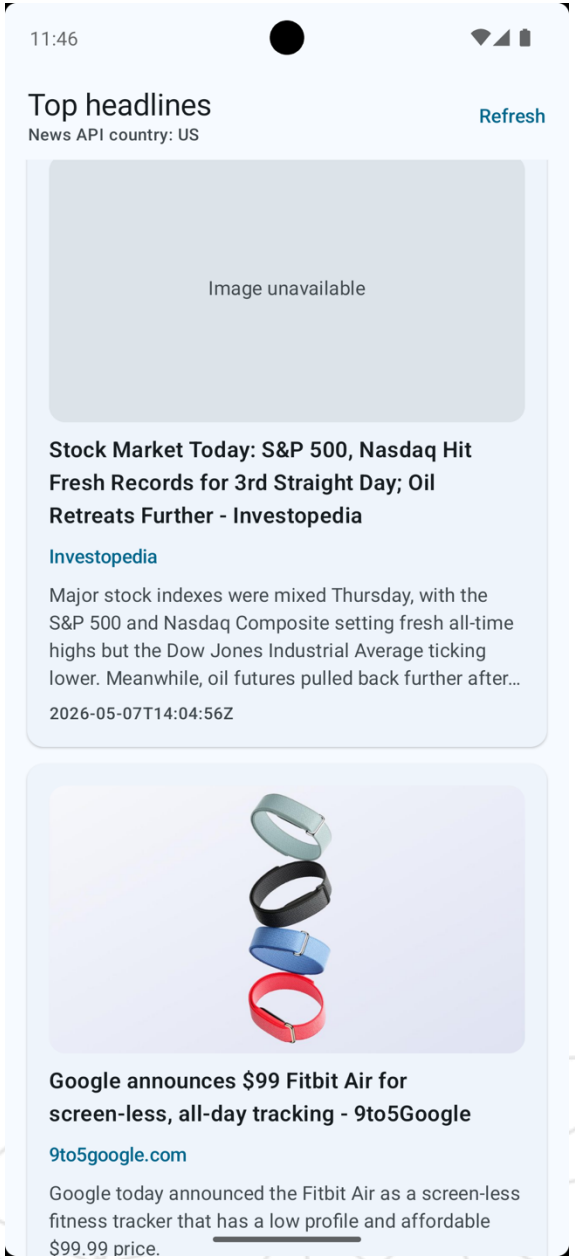
Swift Android SDK. Когда не стоит

- Новый проект кросс-платформенного приложения – KMP, Flutter
- Серьезный большой проект (с CI/CD, Unit-тестами, UI-тестами...)

Реальный пет-пример. NewsApi.org

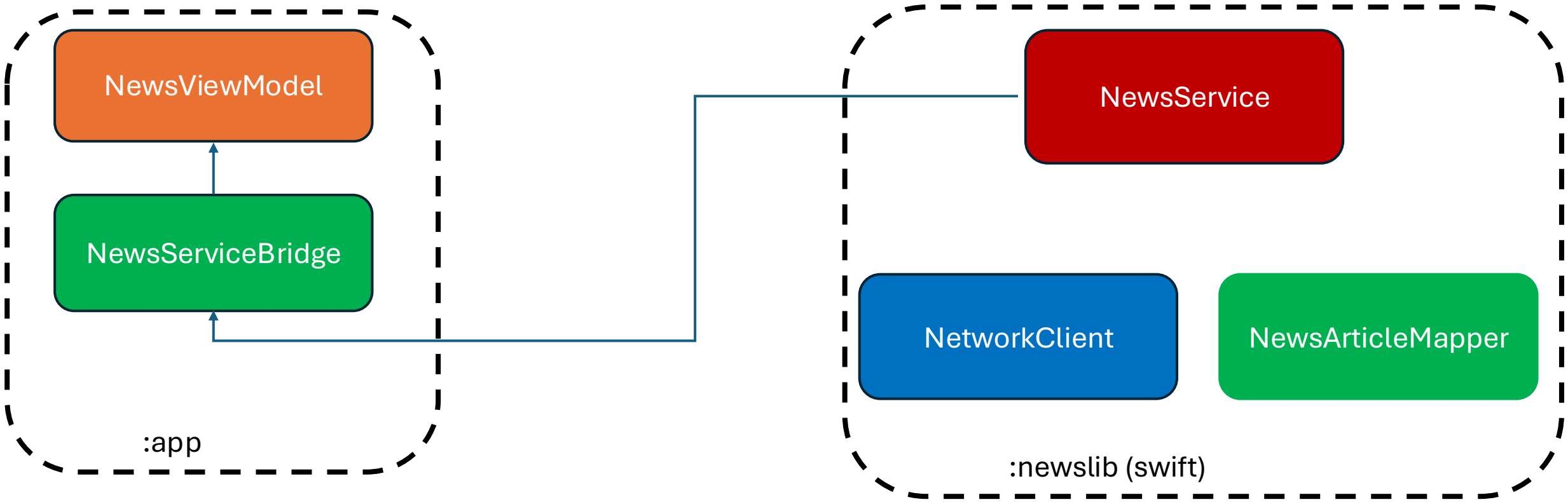
AndroidApp:
- NewsViewModel + UI

NewsLib:
- Network
- Data models
- Business logic



USETECH

Структура проекта



План-капкан

1 Установим и настроим Swift Android SDK

2 Подготовим структуру проекта и swift-android.build.gradle.kts

3 SPM модуля логики

4 Сетевой слой и бизнес-логика

5 Bridge

6 Решаем проблемы: память и многопоточность

Что понадобится

- swiftly
- Swift Toolchain
- Android Swift SDK
- Android NDK

```
% swiftly install latest
Fetching the latest stable Swift release...
Installing Swift 6.3.1

Installing package in user home directory...
Swift 6.3.1 is installed successfully!
% swiftly use latest
The global default toolchain has been set to `Swift 6.3.1`

% swift --version
Apple Swift version 6.3.1 (swift-6.3.1-RELEASE)
Target: arm64-apple-macosx26.0
```

Что понадобится

- Swift Toolchain
- Android Swift SDK
- Android NDK

```
% swift sdk install https://download.swift.org/swift-6.3.1-release/android-sdk/swift-6.3.1-
RELEASE/swift-6.3.1-RELEASE_android.artifactbundle.tar.gz --checksum
8193a4e96538635131a154736c8896fba0e5a1c30e065524f00ed78719bac35a
```

Что понадобится

- Swift Toolchain
- Android Swift SDK
- Android NDK

Если NDK у вас уже есть, просто установите ANDROID_NDK_HOME

```
% mkdir ~/android-ndk
% cd ~/android-ndk
% curl -fSLO https://dl.google.com/android/repository/android-ndk-r27d-$(uname -s).zip
% unzip -q android-ndk-r27d-*.zip
% export ANDROID_NDK_HOME=$PWD/android-ndk-r27d
```

Важно! Линкуем NDK с SwiftPM

```
% cd ~/Library/org.swift.swiftpm/swift-sdks/swift-6.3.1-RELEASE_android.artifactbundle/swift-android/
% ./scripts/setup-android-sdk.sh
```

План-капкан

~~1 Установим и настроим Swift Android SDK~~

2 Подготовим структуру проекта и swift-android.build.gradle.kts

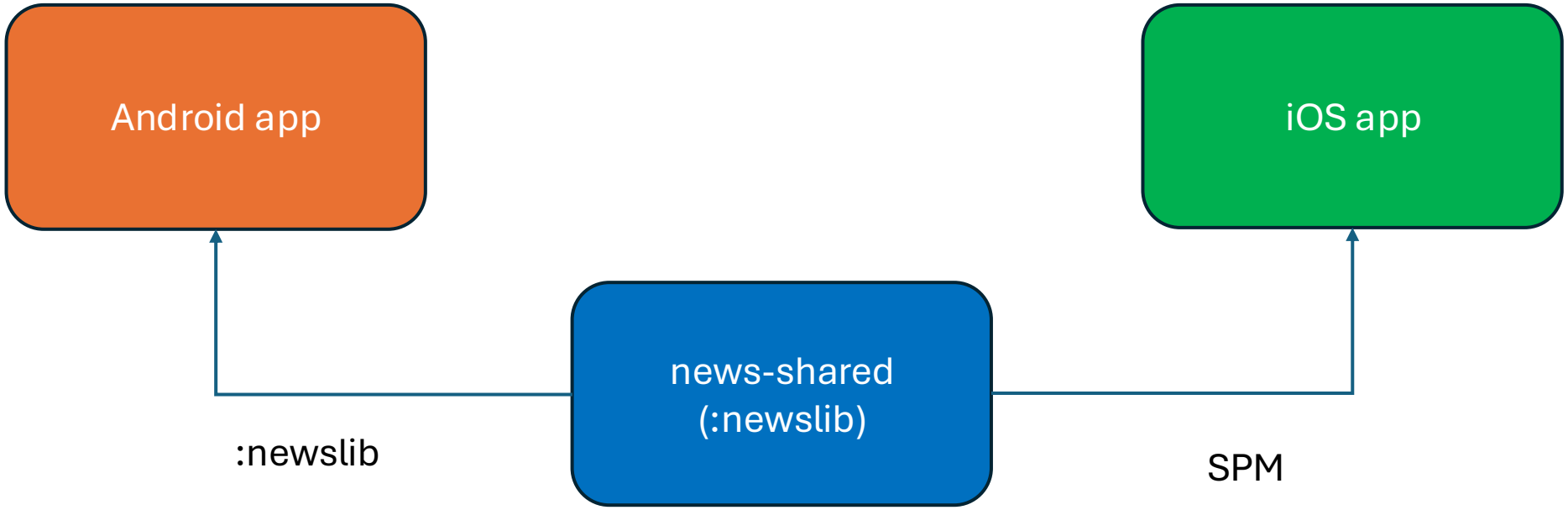
3 SPM модуля логики

4 Сетевой слой и бизнес-логика

5 Bridge

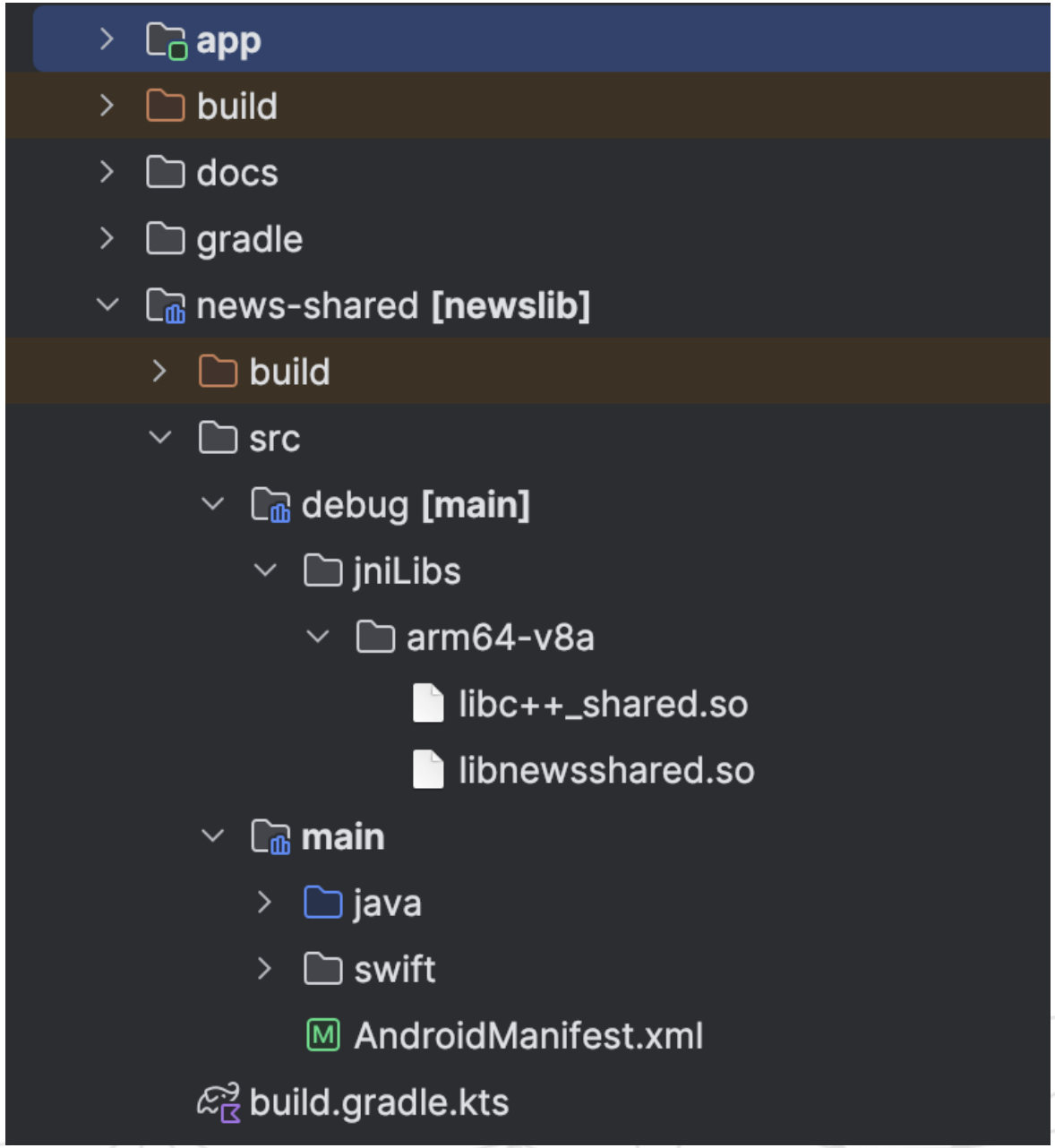
6 Решаем проблемы: память и многопоточность

Структура проекта



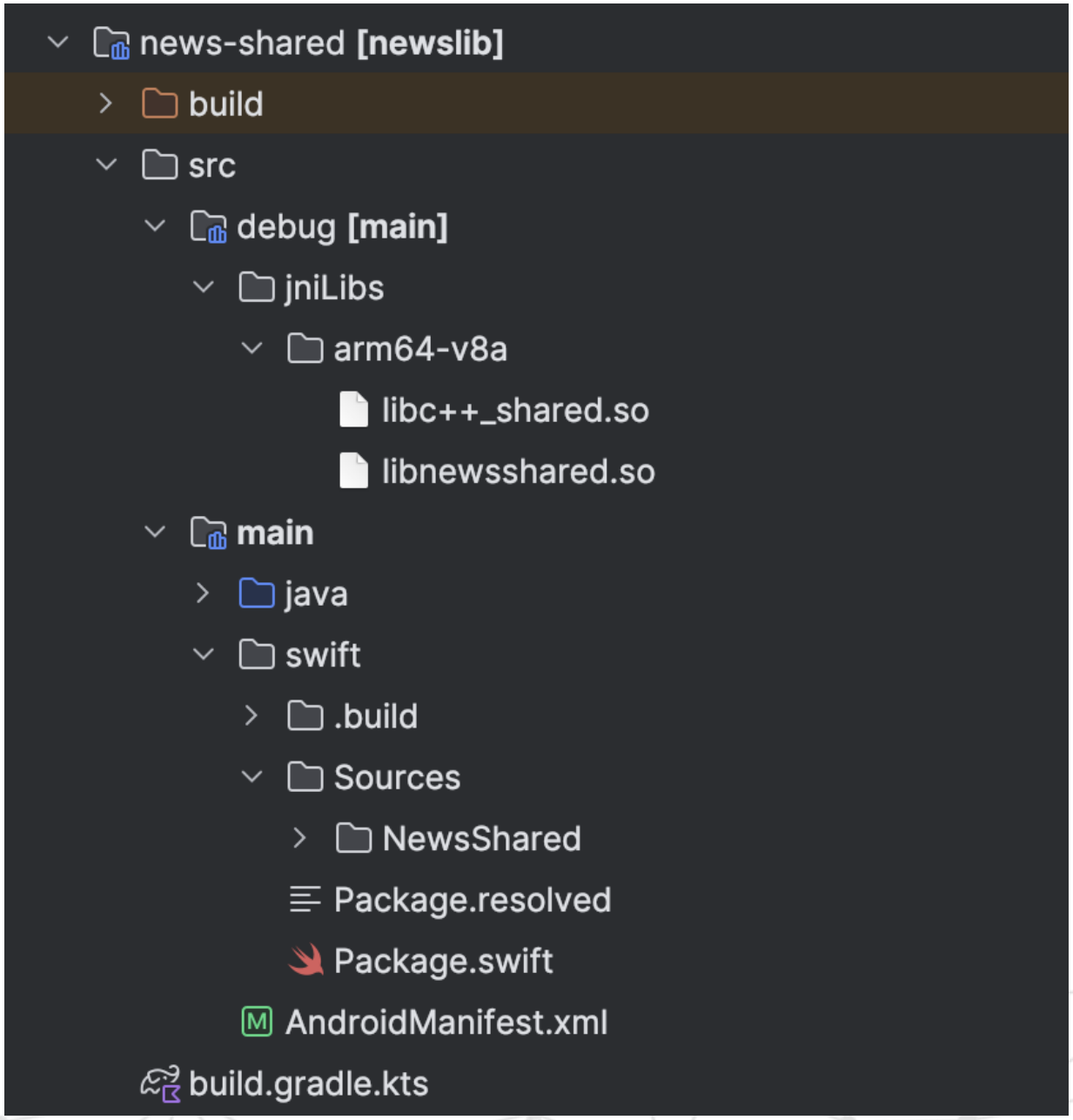
Структура проекта

- App
- News-shared:
- debug –jniLibs (генерируется)



Структура проекта

- News-shared:
- debug –jniLibs (генерируется)
- java (бриджи)
- swift (исходники)
- Package



SPM настройки

- Dynamic library
- SwiftJava

```swift
import CompilerPluginSupport
import PackageDescription

let package = Package(
    name: "newslib",
    products: [
        .library(name: "newslib", type: .dynamic, targets: ["NewsShared"])
    ],
    dependencies: [
        .package(url: "https://github.com/swiftlang/swift-java", from: "0.2.0")
    ],
    targets: [
        .target(
            name: "NewsShared",
            dependencies: [ .product(name: "SwiftJava", package: "swift-java")],
            exclude: ["swift-java.config"],
            swiftSettings: [
                .swiftLanguageMode(.v6),
                .unsafeFlags([
                    "-Xfrontend", "-strict-concurrency=minimal"
                ])
            ],
            plugins: [
                .plugin(name: "JExtractSwiftPlugin", package: "swift-java")
            ]
        )
    ]
)
```

USETECH

SPM настройки

- Dynamic library
- SwiftJava

```swift
import CompilerPluginSupport
import PackageDescription

let package = Package(
    name: "newslib",
    products: [
        .library(name: "newslib", type: .dynamic, targets: ["NewsShared"])
    ],
    dependencies: [
        .package(url: "https://github.com/swiftlang/swift-java", from: "0.2.0")
    ],
    targets: [
        .target(
            name: "NewsShared",
            dependencies: [ .product(name: "SwiftJava", package: "swift-java")],
            exclude: ["swift-java.config"],
            swiftSettings: [
                .swiftLanguageMode(.v6),
                .unsafeFlags([
                    "-Xfrontend", "-strict-concurrency=minimal"
                ])
            ],
            plugins: [
                .plugin(name: "JExtractSwiftPlugin", package: "swift-java")
            ]
        )
    ]
)
```

USETECH

SPM настройки

Dependencies:

- AsyncHttpClient

```
let package = Package(
    ...
    dependencies: [
        .package(url: "https://github.com/swift-server/async-http-client.git", from: "1.33.1"),
        .package(url: "https://github.com/swiftlang/swift-java", from: "0.2.0")
    ],
    targets: [
        .target(
            name: "NewsShared",
            dependencies: [
                .product(
                    name: "AsyncHTTPClient",
                    package: "async-http-client",
                    condition: .when(platforms: [.android])
                ),
                .product(
                    name: "SwiftJava",
                    package: "swift-java"
                )
            ],
            exclude: ["swift-java.config"],
            swiftSettings: [...],
            plugins: [
                .plugin(name: "JExtractSwiftPlugin", package: "swift-java")
            ]
        )
    ]
)
```

USETECH

Настройки swift-android.build.gradle.kts

Задача для сборки Swift-исходников

```
fun createSwiftBuildTask(buildTypeName: String, arch: Arch, isDebug: Boolean) =
    tasks.register(
        "swiftBuild${arch.variantName}${buildTypeName.replaceFirstChar { it.uppercaseChar() }}",
        Exec::class.java
    ) {
        workingDir("src/main/swift")
        executable(getSwiftlyPath())

        val sdkName = "${arch.targetTriple}${swiftConfig.apiLevel}"
        args(...)

        inputs.file(layout.projectDirectory.file("src/main/swift/Package.swift"))
        inputs.dir(layout.projectDirectory.dir("src/main/swift/Sources"))
        outputs.dir(layout.projectDirectory.dir(buildDirectoryFor(arch, isDebug)))
    }
```

Настройки swift-android.build.gradle.kts

Копирование generated -> libs

```
fun createCopySwiftLibrariesTask(buildTypeName: String, arch: Arch, isDebug: Boolean,
swiftBuildTask: TaskProvider<Exec>) =
    tasks.register(
        "copySwift${arch.variantName}${buildTypeName.replaceFirstChar { it.uppercaseChar() }}",
        Copy::class.java
    ) {
        dependsOn(swiftBuildTask)

        val generatedJniLibsDir = layout.buildDirectory.dir("generated/jniLibs")
        val swiftSdkBundlePath = getSwiftSdkBundlePath()

        //... само копирование
        duplicatesStrategy = DuplicatesStrategy.INCLUDE
    }
```

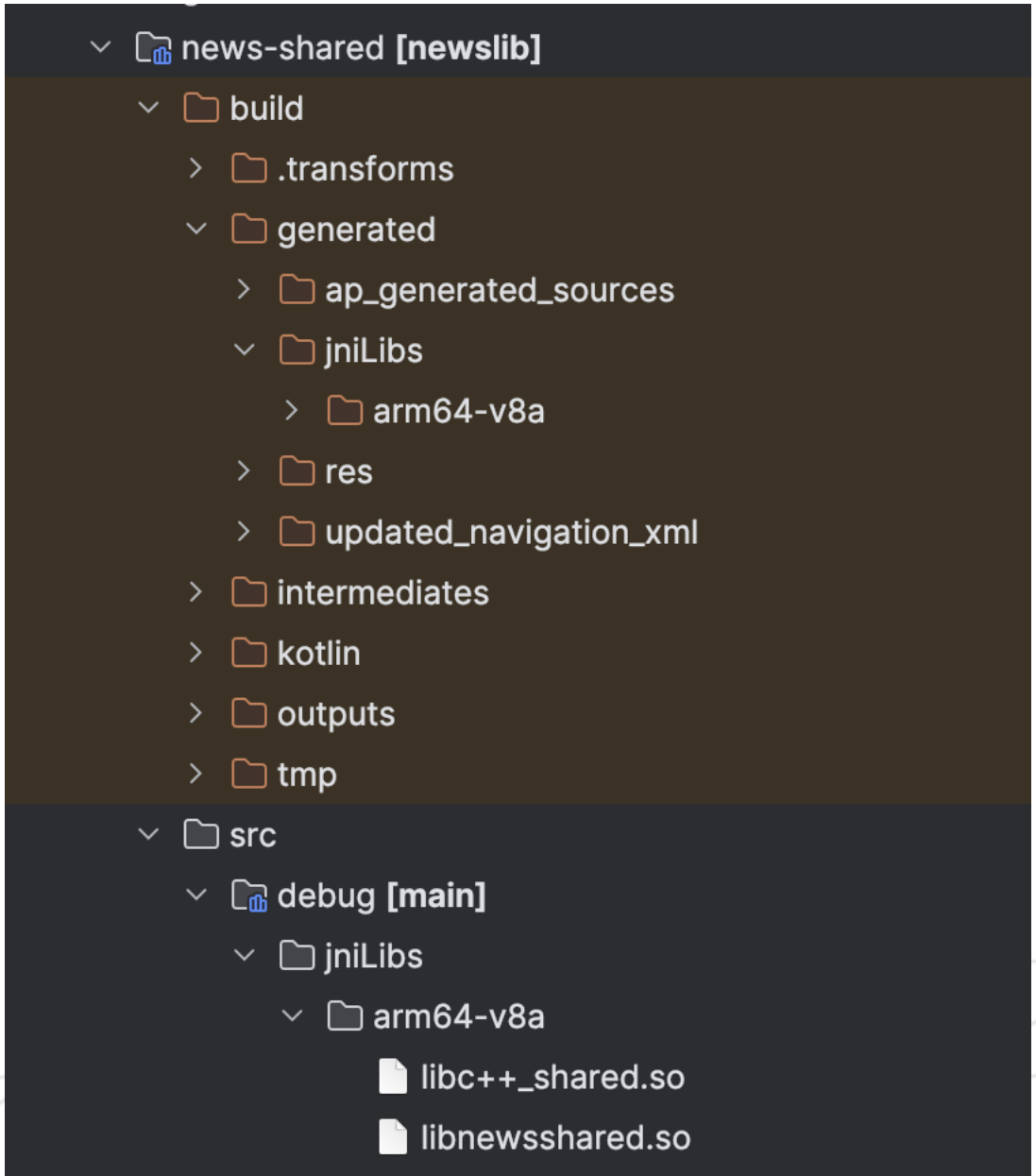
Настройки swift-android.build.gradle.kts

Копирование generated -> libs

```
from(layout.projectDirectory.dir(buildDirectoryFor(arch, isDebug))) {
            include("*.so")
            into(arch.androidAbi)
        }

        from(file("$swiftSdkBundlePath/ndk-sysroot/usr/lib/${arch.ndkTriple}/libc++_shared.so")) {
            into(arch.androidAbi)
        }

        from(
            swiftRuntimeLibs.map { libName ->
                "$swiftSdkBundlePath/swift-
resources/usr/lib/${arch.swiftRuntimeDir}/android/lib$libName.so"
            }
        ) {
            into(arch.androidAbi)
        }

        into(generatedJniLibsDir)
```

Настройки swift-android.build.gradle.kts

Копирование generated -> libs



Настройки build.gradle.kts

Настраиваем пути для исходников

```
sourceSets {
        getByName("main") {
            java.srcDir("src/main/swift/.build/plugins/outputs/swift/" +
                "NewsShared/destination/JExtractSwiftPlugin/src/generated/java")
            jniLibs.srcDir(layout.buildDirectory.dir("generated/jniLibs"))
        }
        getByName("debug") {
            jniLibs.setSrcDirs(emptyList<String>())
        }
        getByName("release") {
            jniLibs.setSrcDirs(emptyList<String>())
        }
    }
```

Настройки build.gradle.kts

Путь к swift-droid.gradle.kts
org.swift.swiftkit:swiftkit-core

```
apply(from = "../swift-android.gradle.kts")

dependencies {
    implementation(libs.androidx.core.ktx)
    api(libs.swiftkit.core)
}
```

Настройки swift-java.config

Наименование библиотек
Режим JNI

```
// swift-java.config
{
  "javaPackage": "ru.azharkova.sampleapp.newslib.swift",
  "nativeLibraryName": "newslib",
  "mode": "jni",
  "asyncFuncMode": "legacyFuture"
}
```

План-капкан

~~1 Установим и настроим Swift Android SDK~~

~~2 Подготовим структуру проекта и swift-android.build.gradle.kts~~

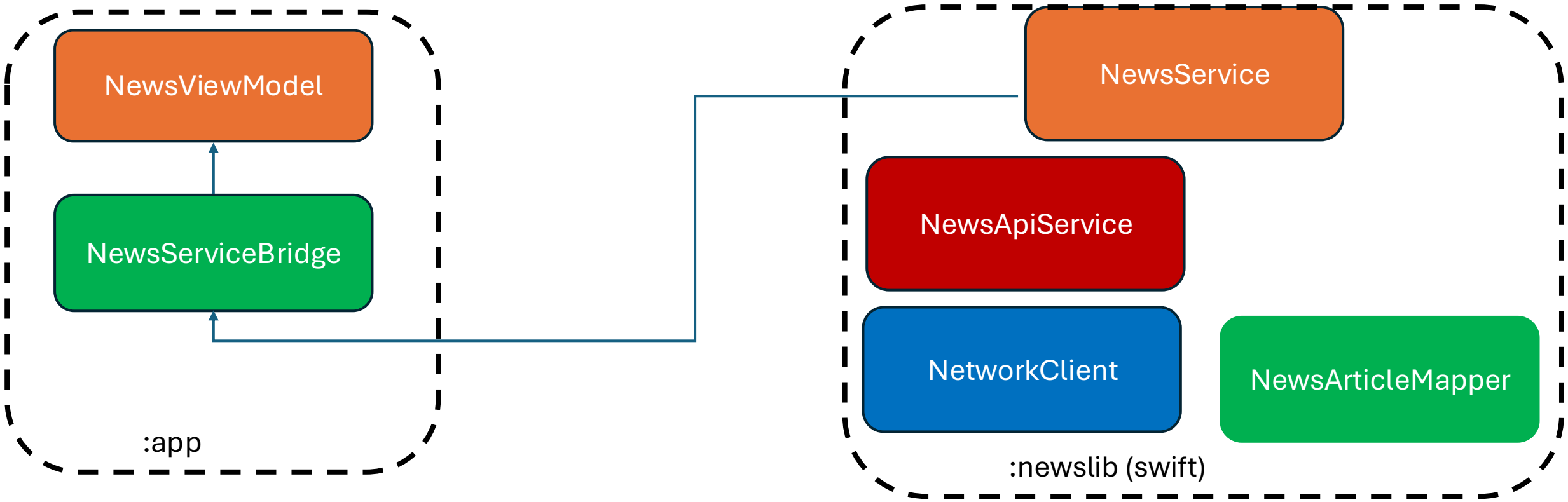
~~3 SPM модуля логики~~

4 Сетевой слой и бизнес-логика

5 Bridge

6 Решаем проблемы: память и многопоточность

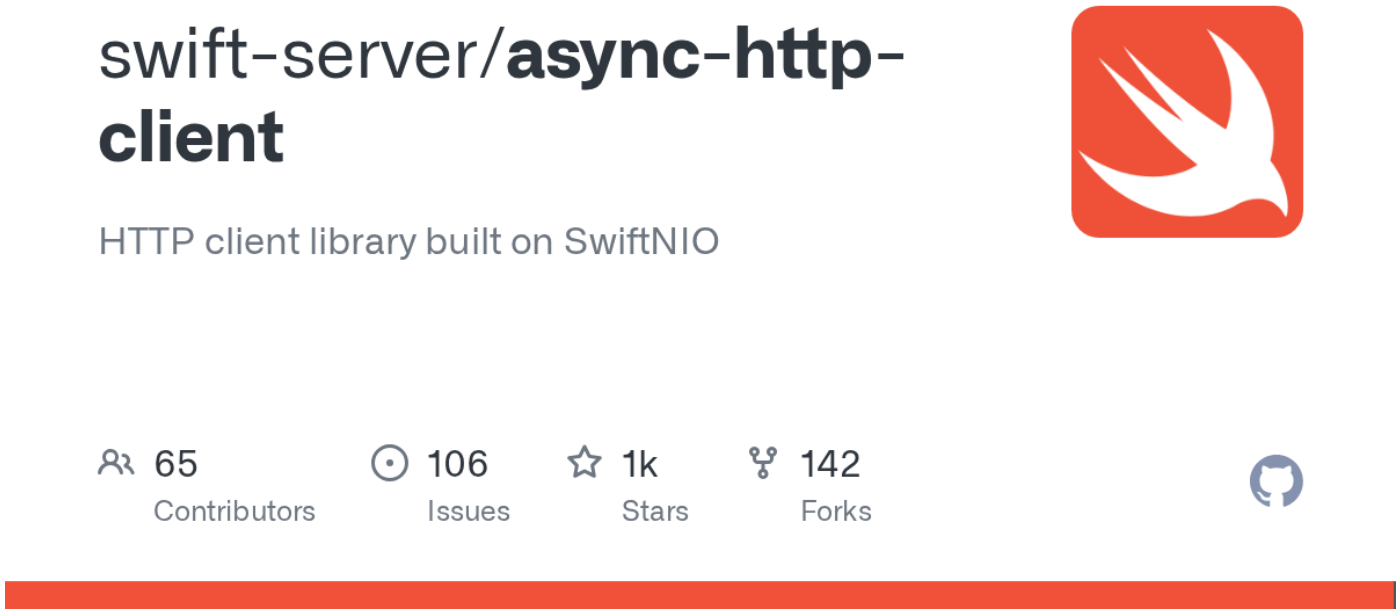
Структура проекта



Сетевой клиент для SwiftJava. Транспорт

https://github.com/swift-server/async-http-client

SwiftNIO поддерживает Android NDK



Сетевой клиент для SwiftJava. Транспорт

#if os(Android) – для Android-таргета
#if canImport(Foundation) для ios/macOS

```
   private static func makeTransport() -> some ClientTransport {
#if os(Android)
        return AsyncHTTPClientTransport(configuration: .init())
#else
        return URLSessionTransport()
#endif
    }
```

Сетевой клиент для SwiftJava

Обычный сетевой клиент. Создаем запрос через фабрику транспорта

```swift
final class NetworkClient {
    private let transport: any NetworkTransport

    init(
        transport: any NetworkTransport = makeTransport()
    ) {
        self.transport = transport
    }

    /// Выполняет запрос и возвращает сырой HTTP-ответ.
    func execute<Request: NetworkRequest>(_ request: Request) async throws -> NetworkResponse {
        let rawRequest = try makeRawRequest(from: request)
        let response = try await transport.execute(rawRequest)

        return response
    }
```

NewsApiService

Вызывает клиент и маппперы

```swift
final class NewsApiService: NewsServicing {
    private let apiKey: String
    private let client: NetworkClient
    private let responseDecoder: NetworkResponseDecoder
    private let articleMapper: NewsArticleMapper
    private let errorMapper: NewsServiceErrorMapper

    func fetchTopHeadlines(country: String) async throws -> [NewsArticle] {
        //...
        do {
            let response = try await client.execute(TopHeadlinesRequest(...))
            let payload = try await responseDecoder.decode(
              NewsAPIResponse.self, from: response.data)

            return payload.articles.map(articleMapper.map)
        } catch {
            throw await errorMapper.map(error)
        }
    }
}
```

SwiftDroidNewsSample

Полный код смотрите здесь

NewsService

Прокси и основная входная точка

```
public final class NewsService {
    private let service: NewsApiService

    public init(apiKey: String) throws {
        self.service = try NewsApiService(apiKey: apiKey)
    }

    public func fetchTopHeadlines(country: String) async throws -> [NewsArticle] {
        try await service.fetchTopHeadlines(country: country)
    }
}
```

План-капкан

~~1 Установим и настроим Swift Android SDK~~

~~2 Подготовим структуру проекта и swift-android.build.gradle.kts~~

~~3 SPM модуля логики~~

~~4 Сетевой слой и бизнес-логика~~

5 Bridge

6 Решаем проблемы: память и многопоточность

NewsServiceBridge. Бридж между app и swift lib

Вызывает Swift-instance NewsService

```
class NewsServiceBridge(
    apiKey: String
) {
    // Здесь хранится именно generated Swift instance, а не Kotlin service
    private val swiftNewsService: NewsService = NewsService.init(apiKey)

    suspend fun fetchTopHeadlines(country: String): List<NewsArticleUi> {
        val articles = swiftNewsService.fetchTopHeadlines(country).awaitSwift()
        return articles.map(::toUiArticle)
    }
}
```

NewsViewModel

Обычная Android ViewModel

```
class NewsViewModel(
    private val apiKey: String = BuildConfig.NEWS_API_KEY,
    private val country: String = BuildConfig.NEWS_COUNTRY
) : ViewModel() {
    private val newsServiceBridge: NewsServiceBridge?
      //...

    fun refresh() {

        viewModelScope.launch {
            _uiState.value = _uiState.value.copy(isLoading = true, errorMessage = null)

            val session = newsServiceBridge

            try {
                val articles = session.fetchTopHeadlines(country = country)
                _uiState.value = NewsUiState(articles = articles)
            } catch (throwable: Throwable) {...}
        }
    }
}
```

NewsViewModel

Обычная Android ViewModel

НЕ СОБЕРЕТСЯ

```
class NewsViewModel(
    private val apiKey: String = BuildConfig.NEWS_API_KEY,
    private val country: String = BuildConfig.NEWS_COUNTRY
) : ViewModel() {
  private val newsServiceBridge: NewsServiceBridge?
     //...

    fun refresh() {

        viewModelScope.launch {
            _uiState.value = _uiState.value.copy(isLoading = true, errorMessage = null)

            val session = newsServiceBridge

            try {
                val articles = session.fetchTopHeadlines(country = country)
                _uiState.value = NewsUiState(articles = articles)
            } catch (throwable: Throwable) {...}
        }
    }
}
```

План-капкан

~~1 Установим и настроим Swift Android SDK~~

~~2 Подготовим структуру проекта и swift-android.build.gradle.kts~~

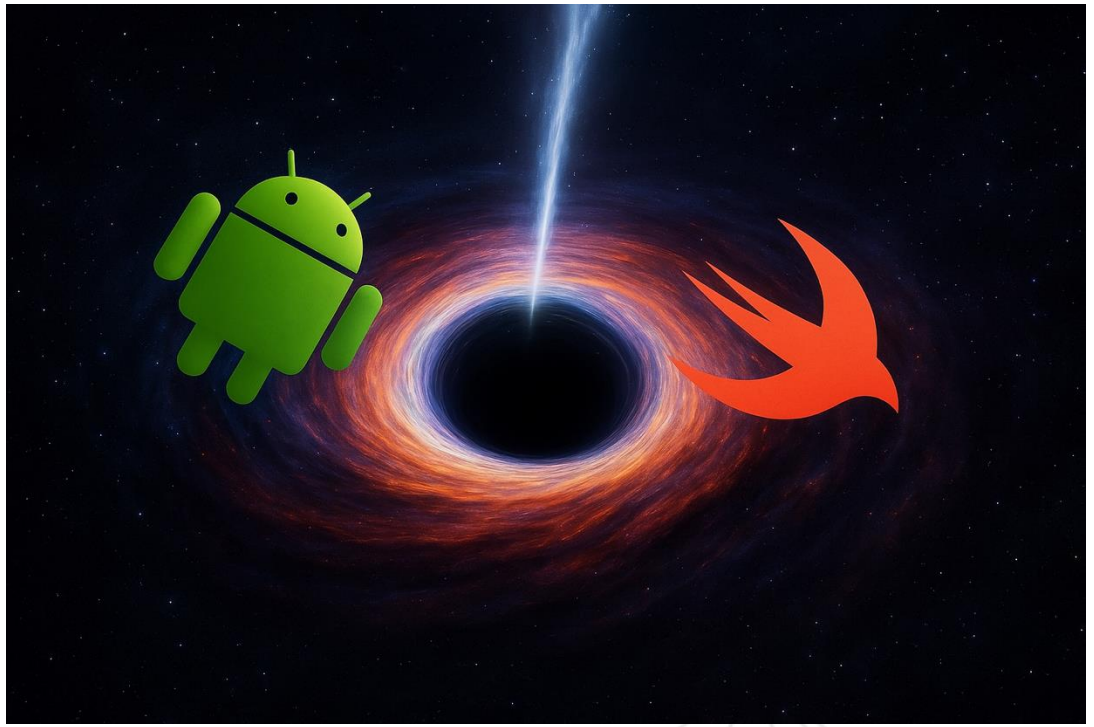
~~3 SPM модуля логики~~

~~4 Сетевой слой и бизнес-логика~~

~~5 Bridge~~

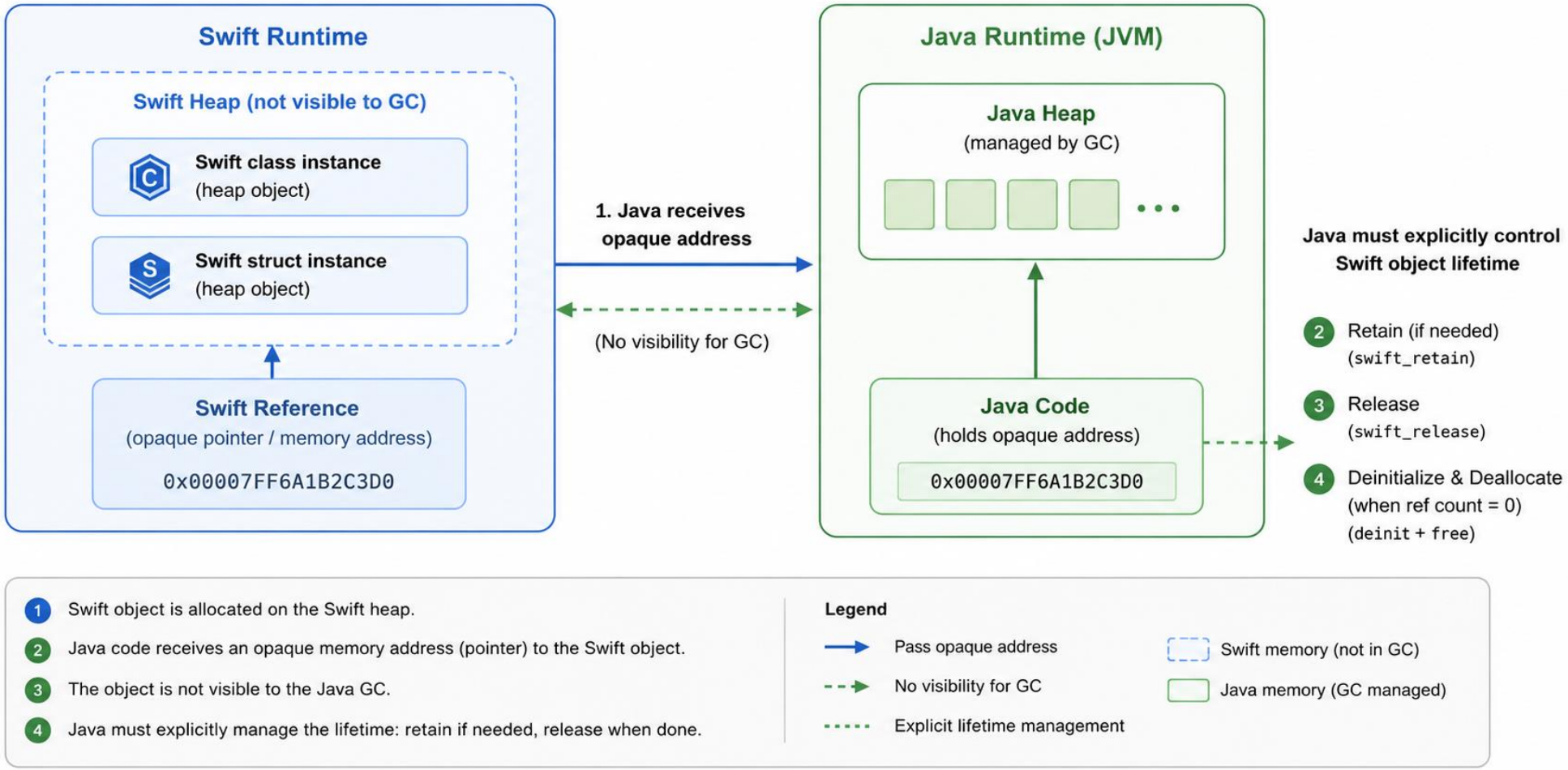
6 Решаем проблемы: память и многопоточность

Проработаем проблемы

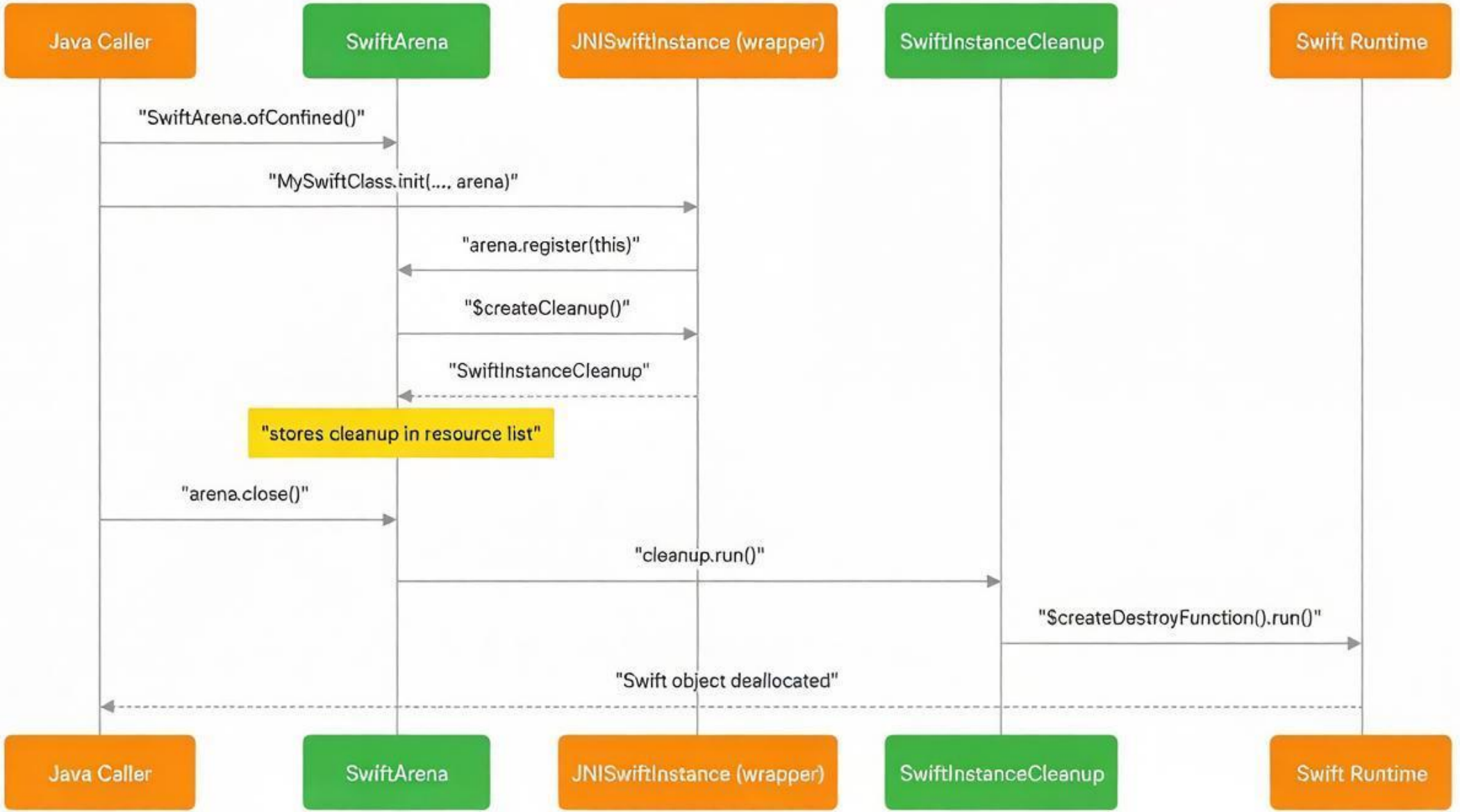


Проблема. Управление памятью

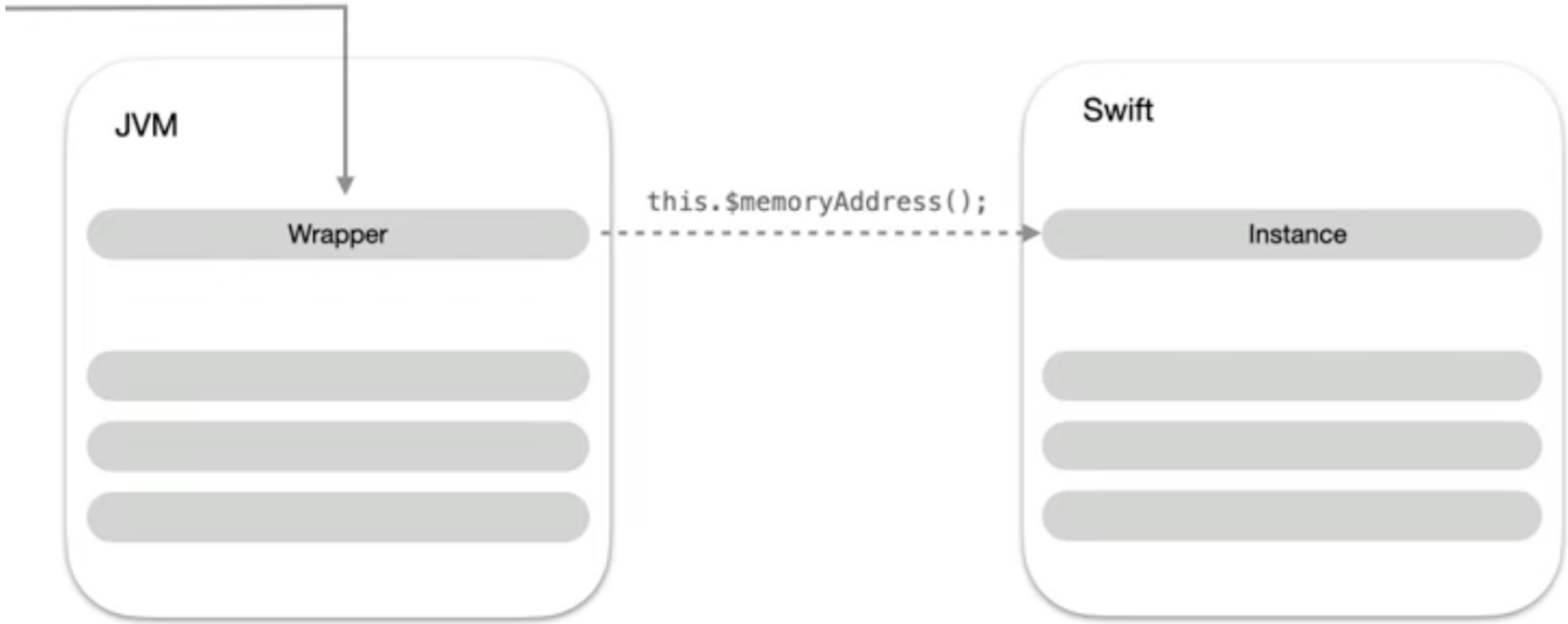
- Java GC не видит Swift классы и структуры в Swift-куче



~~Quake 3.~~ SwiftArena



~~Quake 3.~~ SwiftArena



~~Quake 3.~~ SwiftArena

Интерфейс / Класс	Ответственность	Жизненный цикл
ClosableSwiftArena	Расширяет SwiftArena и AutoCloseable для использования try-with-resources.	Явный (close())
ConfinedSwiftMemorySession	ClosableSwiftArena отслеживает ресурсы в списке.	Явный (close())
AutoSwiftMemorySession	SwiftCleaner запускает очистку через GC.	Неявный (GC-driven)
AllocatingSwiftArena	FFM-реализация, объединяющая жизненный цикл Swift с выделением MemorySegment FFM.	Явный или GC

~~Quake 3.~~ SwiftArena

```
public interface SwiftArena {
    /**
     * Register a Swift object.
     * Its memory should be considered managed by this arena, and be destroyed when the arena is
closed.
     */
    void register(SwiftInstance instance);

    static ClosableSwiftArena ofConfined() {
        return new ConfinedSwiftMemorySession();
    }

    static SwiftArena ofAuto() {
        ThreadFactory cleanerThreadFactory = r -> new Thread(r, "AutoSwiftArenaCleanerThread");
        return new AutoSwiftMemorySession(cleanerThreadFactory);
    }
}
```

~~Quake 3.~~ SwiftArena

```
final class AutoSwiftMemorySession implements SwiftArena {
    private final SwiftCleaner swiftCleaner;

    public AutoSwiftMemorySession(ThreadFactory cleanerThreadFactory) {
        this.swiftCleaner = SwiftCleaner.create(cleanerThreadFactory);
    }

    @Override
    public void register(SwiftInstance instance) {
        Objects.requireNonNull(instance, "value");

        // Гарантируем, что instance не захватится во время очистки,
        // поэтому можем игнорировать предупреждение ниже.
        var cleanupAction = instance.$createCleanup();
        swiftCleaner.register(instance, cleanupAction);
    }
}
```

Поддержим в коде. JNISwiftInstance

Инициализатор со стороны Java/Kotlin требует передачи SwiftArena

```
// Инстанс Swift
public final class NewsService {

    public init(apiKey: String) {}
}

// JNISwiftInstance
public static NewsService init(java.lang.String apiKey, SwiftArena swiftArena) throws Exception {
    return NewsService.wrapMemoryAddressUnsafe(NewsService.$init(apiKey), swiftArena);
  }
```

Поддержим в коде. JNISwiftInstance

Если нет привязки к реальной SwiftArena, то запрос объекта по указателю может привести к крашу

```
private NewsService(long selfPointer, SwiftArena swiftArena) {
    SwiftObjects.requireNonZero(selfPointer, "selfPointer");
    this.selfPointer = selfPointer;

    // Only register once we have fully initialized the object since this will need the object
pointer.
    swiftArena.register(this);
  }

  public static NewsService wrapMemoryAddressUnsafe(long selfPointer, SwiftArena swiftArena) {
    return new NewsService(selfPointer, swiftArena);
  }

  public static NewsService wrapMemoryAddressUnsafe(long selfPointer) {
    return new NewsService(selfPointer, SwiftMemoryManagement.DEFAULT_SWIFT_JAVA_AUTO_ARENA);
  }
```

Поддержим в коде

ClosableSwiftArena управляем через ViewModel

```
class NewsViewModel(
    private val swiftArena: ClosableSwiftArena = SwiftArena.ofConfined()
) : ViewModel() {


    private val newsServiceBridge: NewsServiceBridge?

    init {
        var createdSession: NewsServiceBridge? = null
        try {
            createdSession = NewsServiceBridge(
                swiftArena = swiftArena
            )
        } catch (throwable: Throwable) {
            initializationError = throwable
        }

        newsServiceBridge = createdSession
    }


    }
    override fun onCleared() {
        newsServiceBridge?.close()
        super.onCleared()
    }

}
```

Поддержим в коде

ClosableSwiftArena управляем через ViewModel

```
class NewsViewModel(
    private val swiftArena: ClosableSwiftArena = SwiftArena.ofConfined()
) : ViewModel() {


    private val newsServiceBridge: NewsServiceBridge?

    init {
        var createdSession: NewsServiceBridge? = null
        try {
            createdSession = NewsServiceBridge(
                swiftArena = swiftArena
            )
        } catch (throwable: Throwable) {
            initializationError = throwable
        }

        newsServiceBridge = createdSession
    }


    }
    override fun onCleared() {
        newsServiceBridge?.close()
        super.onCleared()
    }

}
```

Поддержим в коде. Bridge Autoclosable

Bridge -> Service

Autoclosable

```kotlin
class NewsServiceBridge(
    private val swiftArena: ClosableSwiftArena
) : AutoCloseable {
    // Здесь хранится именно generated Swift instance, а не Kotlin service.
    private val swiftNewsService: NewsService = NewsService.init(swiftArena)

    suspend fun fetchTopHeadlines(country: String): List<NewsArticleUi> {
        val articles = swiftNewsService.fetchTopHeadlines(swiftArena).awaitSwift()
        return articles.map(::toUiArticle)
    }

    override fun close() {
        swiftArena.close()
    }
}
```

Поддержим в коде. Bridge Autoclosable

Bridge -> Service

```
class NewsServiceBridge(
    private val swiftArena: ClosableSwiftArena
) : AutoCloseable {
    // Здесь хранится именно generated Swift instance, а не Kotlin service.
    private val swiftNewsService: NewsService = NewsService.init(swiftArena)

    suspend fun fetchTopHeadlines(country: String): List<NewsArticleUi> {
        val articles = swiftNewsService.fetchTopHeadlines(swiftArena) awaitSwift()
        return articles.map(::toUiArticle)
    }

    override fun close() {
        swiftArena.close()
    }
}
```

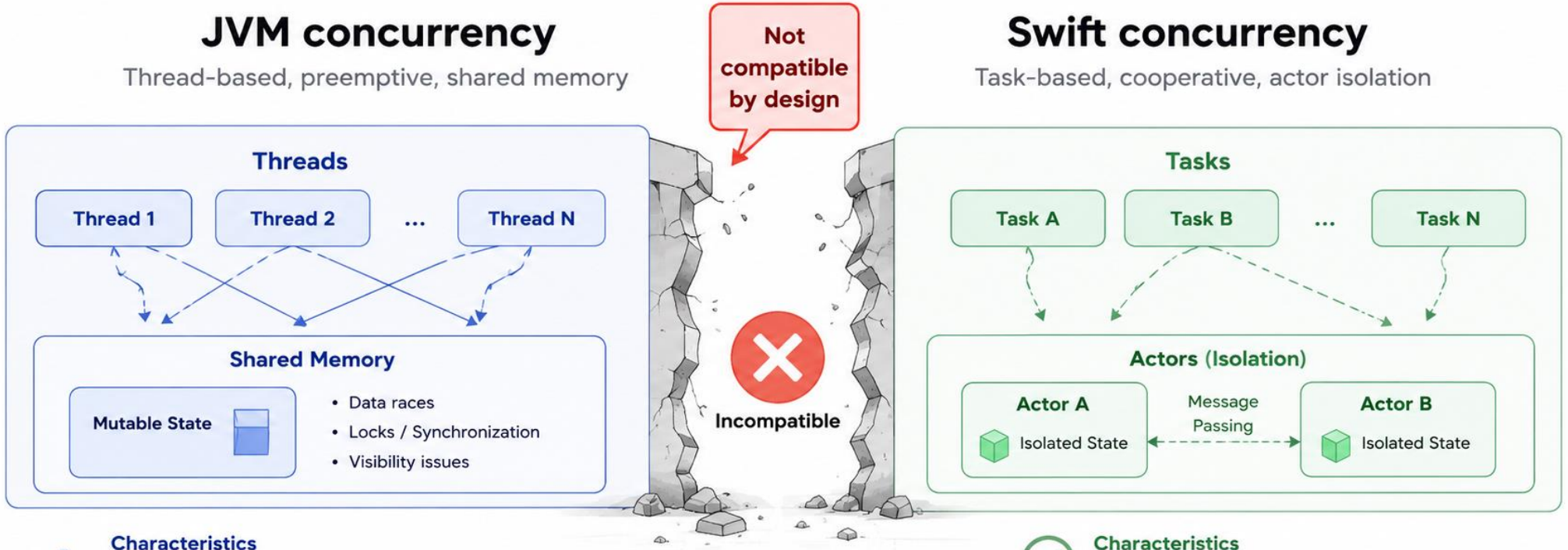
Поддержим в коде. Очистка

Очистка объекта в памяти (видимый для GC)

```
@Override
  public Runnable $createDestroyFunction() {
    long self$ = this.$memoryAddress();
    long selfType$ = this.$typeMetadataAddress();
    if (CallTraces.TRACE_DOWNCALLS) {
      CallTraces.traceDowncall("NewsService.$createDestroyFunction",
          "this", this,
          "self", self$);
    }
    return new Runnable() {
      @Override
      public void run() {
        if (CallTraces.TRACE_DOWNCALLS) {
          CallTraces.traceDowncall("NewsService.$destroy", "self", self$);
        }
        SwiftObjects.destroy(self$, selfType$);
      }
    };
  }
```

Проблема. Поддержка многопоточности

- JVM concurrency изначально не совместима со Swift-concurrency



Проблема. Поддержка многопоточности

```swift
@_cdecl("Java_ru_azharkova_sampleapp_sharednews_NewsBridge_fetchTopHeadlinesAsync")
public func newsBridgeFetchTopHeadlinesAsync(
    env: UnsafeMutablePointer<JNIEnv?>?,
    _: jobject?,
    requestId: jlong
) {
    guard let env else {
        return
    }

  // С переключением между потоками
    guard let requestContext = makeNewsJNIAsyncRequestContext(env: env, requestId: requestId) else {
        throwRuntimeException(env: env, message: "Unable to prepare async")
        return
    }

    DispatchQueue.global(.) {
        do {
            let articles = try await loadTopHeadlines()
            dispatchNewsLoadSuccess(requestContext: requestContext,
                articles: articles
            )
        } catch {..}
    }
}
```

Проблема. Поддержка многопоточности

AttachCurrentThread/DetachCurrentThread

```
private func withAttachedJNIEnv(
    javaVM: UnsafeMutablePointer<JavaVM?>,
    _ body: (UnsafeMutablePointer<JNIEnv?>) -> Void
) {
    var attachedEnv: UnsafeMutablePointer<JNIEnv?>?
    let attachStatus = javaVM.pointee?.pointee.AttachCurrentThread(javaVM, &attachedEnv, nil) ??
JNI_ERR
    guard attachStatus == JNI_OK, let attachedEnv else {
        return
    }

    body(attachedEnv)
    _ = javaVM.pointee?.pointee.DetachCurrentThread(javaVM)
}
```

Проблема. Поддержка многопоточности

AttachCurrentThread/DetachCurrentThread – напоминает DetachedGraph Kotlin Native

```
//Sample from Supranatural
fun <T, V> execute(jobInput: T, job: (T) -> V): Future<V> {
        val deferred = DeferredFuture<V>()
        val detached = DetachedObjectGraph { Triple(jobInput, job, deferred)
                            .toImmutable() }.asCPointer()

        dispatch_async_f(dispatch_get_main_queue(), detached, staticCFunction {
            it: COpaquePointer? ->

            val attached = DetachedObjectGraph<Triple<T, (T) -> V,
                           DeferredFuture<V>>>(it).attach()
            val result = attached.second(attached.first)
            attached.third.setValue(result)
        })
        return deferred
```

Проблема. Поддержка многопоточности

DispatchWorkItem

```
@_cdecl("Java_org_example_helloswift_MainActivity_startTicks")
public func MainActivity_startTicks(env: UnsafeMutablePointer<JNIEnv?>, thiz: jobject) {
    guard let globalRef = env.pointee!.pointee.NewGlobalRef(env, thiz) else { return }
    guard let cls = env.pointee!.pointee.GetObjectClass(env, thiz) else { return }
    defer { env.pointee!.pointee.DeleteLocalRef(env, cls) }
    guard let mid = env.pointee!.pointee.GetMethodID(env, cls, "updateTimer", "()V") else { return }

    let activityHandle = JGlobalObject(ref: globalRef)
    let methodHandle = JMethodID(id: mid)

    queue.async {
        workItem?.cancel()
        workItem = DispatchWorkItem {
            getEnvForCurrentThread { env in
                env?.pointee!.pointee.CallVoidMethodA(env, activityHandle.ref, methodHandle.id, nil)
            }
            if let workItem = workItem, workItem.isCancelled == false {
                queue.asyncAfter(deadline: .now() + 1, execute: workItem)
            }
        }
        queue.async(execute: workItem!)
    }
}
```

CompletableFuture <-> Async/await

https://github.com/swiftlang/swift-java/blob/main/Sources/SwiftJavaDocumentation/Documentation.docc/SupportedFeatures.md

Async function mode: completable-future (default)

In this mode `async` functions in Swift are extracted as Java methods returning a `java.util.concurrent.CompletableFuture`. This mode gives the most flexibility and should be prefered if your platform supports `CompletableFuture`.

CompletableFuture <-> Async/await

https://github.com/swiftlang/swift-java/blob/main/Sources/SwiftJavaDocumentation/Documentation.docc/SupportedFeatures.md

Поддержка включается в конфиге

```
// swift-java.config
{
  "javaPackage": "ru.azharkova.sampleapp.newslib.swift",
  "nativeLibraryName": "newslib",
  "mode": "jni",
  "asyncFuncMode": "legacyFuture"
}
```

CompletableFuture <-> Async/await

Насыплем сахарку

```
private suspend fun <T> Future<T>.awaitSwift(): T = withContext(Dispatchers.IO) {
    get()
}
```

```
suspend fun fetchTopHeadlines(country: String): List<NewsArticleUi> {
        val articles = swiftNewsService.fetchTopHeadlines(country, swiftArena).awaitSwift()
        return articles.map(::toUiArticle)
    }
```

CompletableFuture <-> Async/await

Async/await Swift автоматически переводится в CompletableFuture

```
public java.util.concurrent.Future<NewsArticle[]> fetchTopHeadlines(java.lang.String country,
SwiftArena swiftArena) {
    org.swift.swiftkit.core.SimpleCompletableFuture<long[]> future$ = new
org.swift.swiftkit.core.SimpleCompletableFuture<long[]>();
    NewsService.$fetchTopHeadlines(country, this.$memoryAddress(), future$);
    return future$.thenApply((futureResult$) -> {
      return Arrays.stream(futureResult$).mapToObj((pointer) -> {
        return NewsArticle.wrapMemoryAddressUnsafe(pointer, swiftArena);
      } // render(_:_:) @ JExtractSwiftLib/JNISwift2JavaGenerator+JavaTranslation.swift:2024
      ).toArray(NewsArticle[]::new);
    } // render(_:_:) @ JExtractSwiftLib/JNISwift2JavaGenerator+JavaTranslation.swift:2024
    );
  }
```

План-капкан

~~1 Установим и настроим Swift Android SDK~~

~~2 Подготовим структуру проекта и swift-android.build.gradle.kts~~

~~3 SPM модуля логики~~

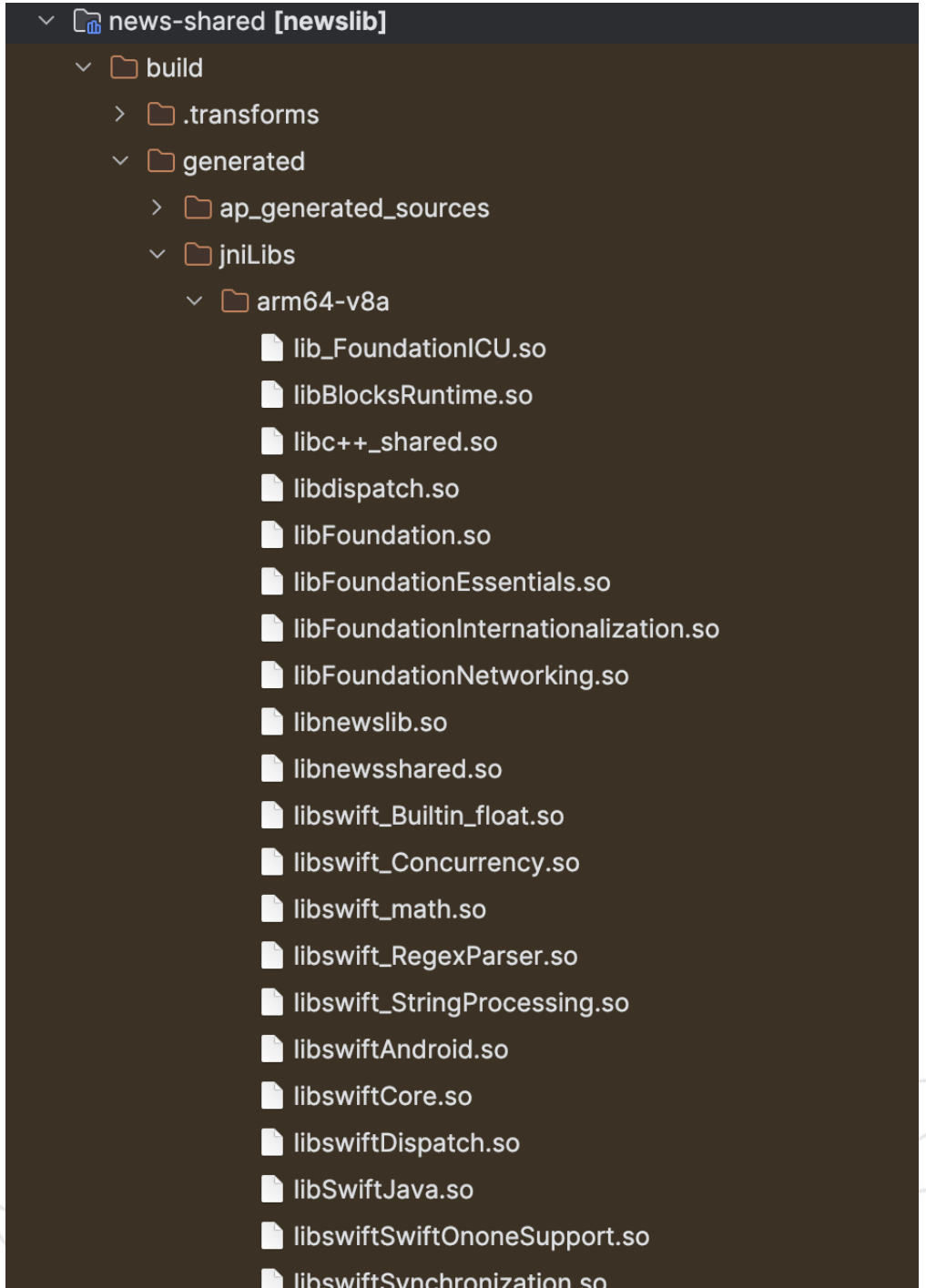
~~4 Сетевой слой и бизнес-логика~~

~~5 Bridge~~

~~6 Решаем проблемы: память и многопоточность~~

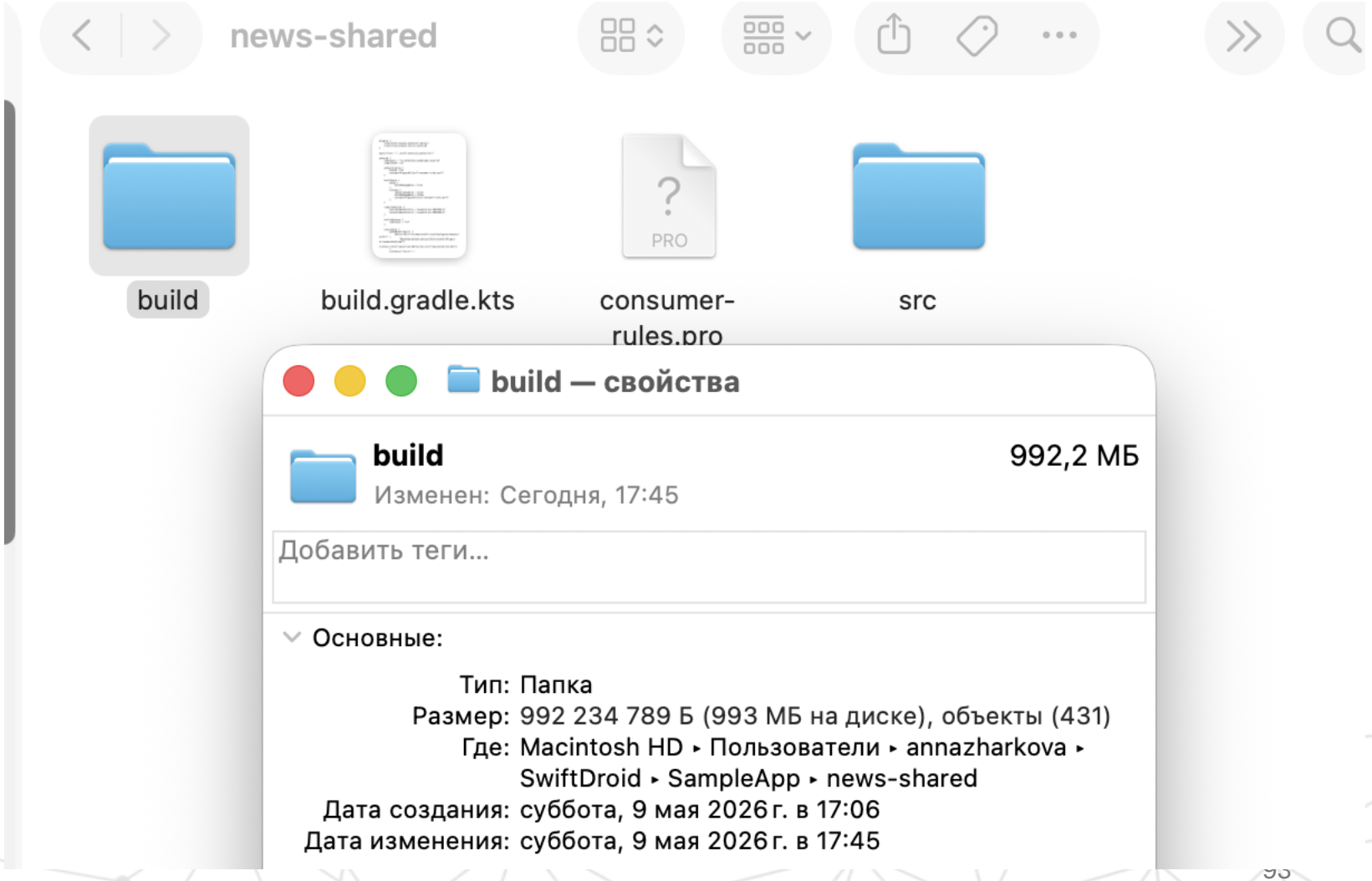
Запускаем билд

- Сгенерированные компоненты



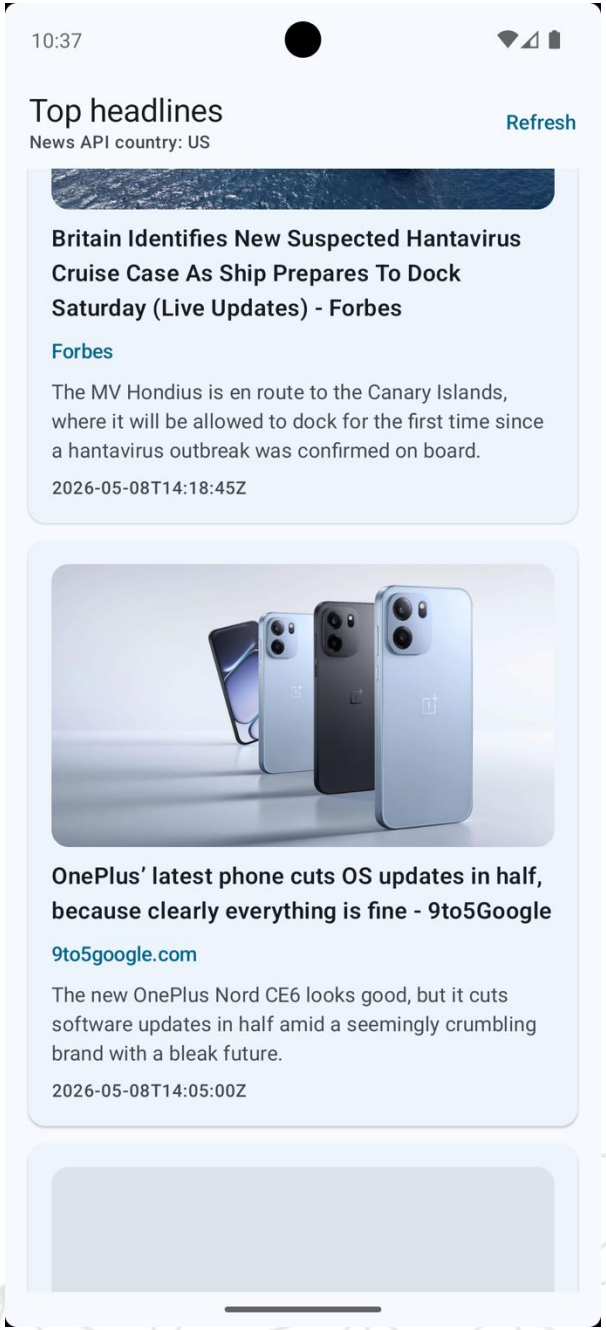
Сгенерировалось

- Что-то много



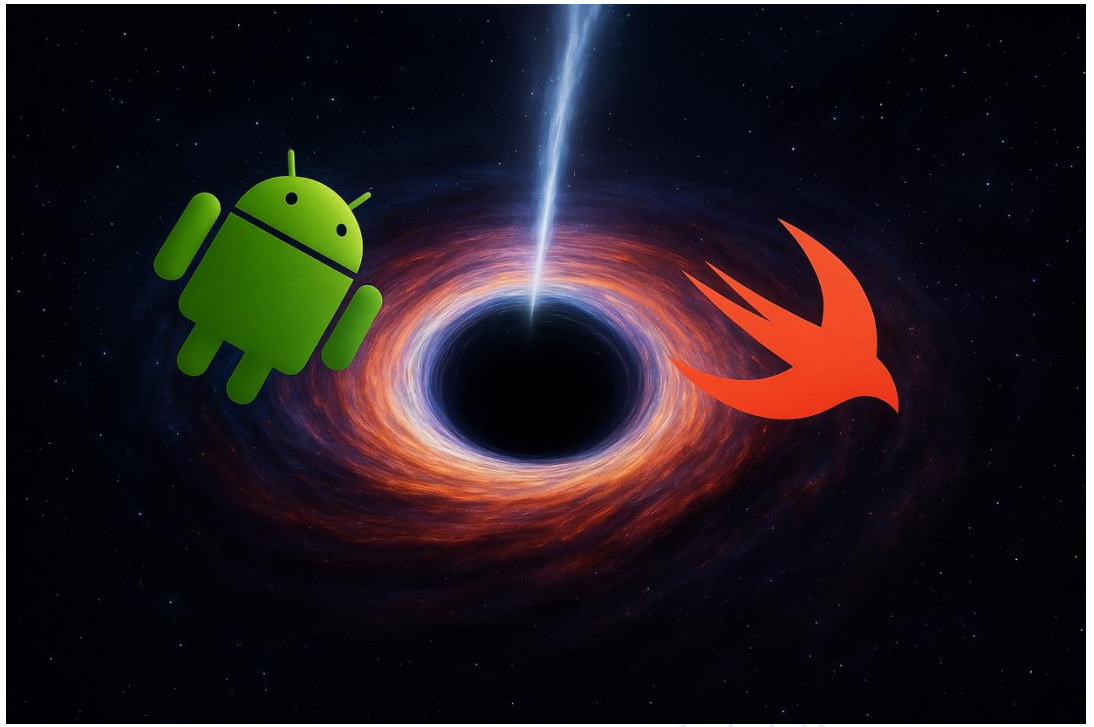
Проверяем результат

https://gitverse.ru/azharkova/swift-droid-news-sample



USETECH

Забыли про UI



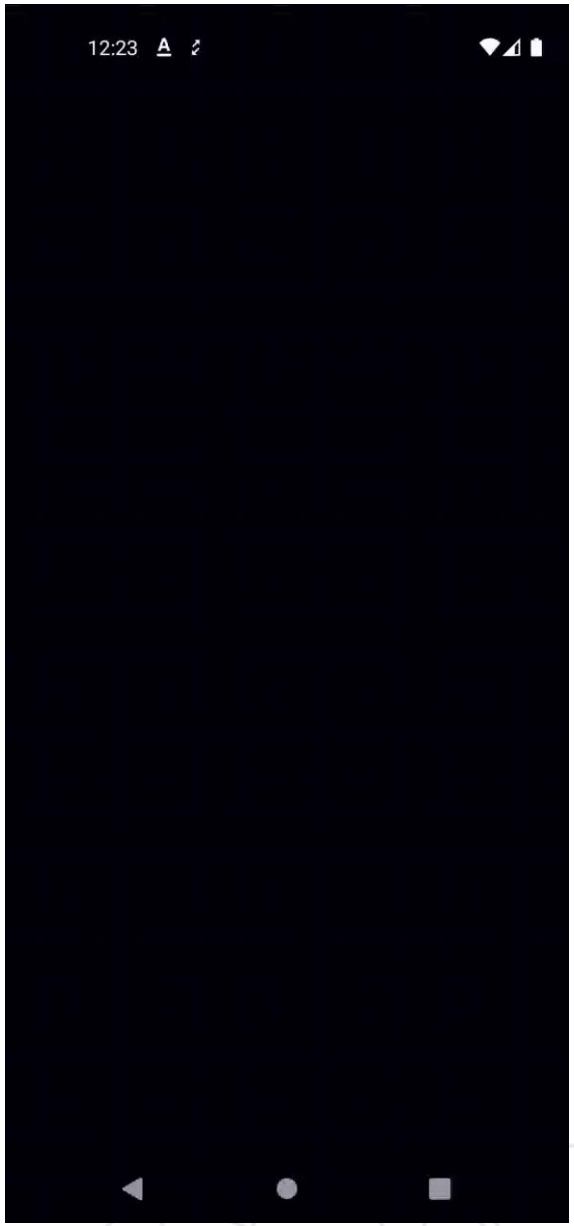
UI для такого приложения

- UI нативный
- Skip Tools
- NDK Activity

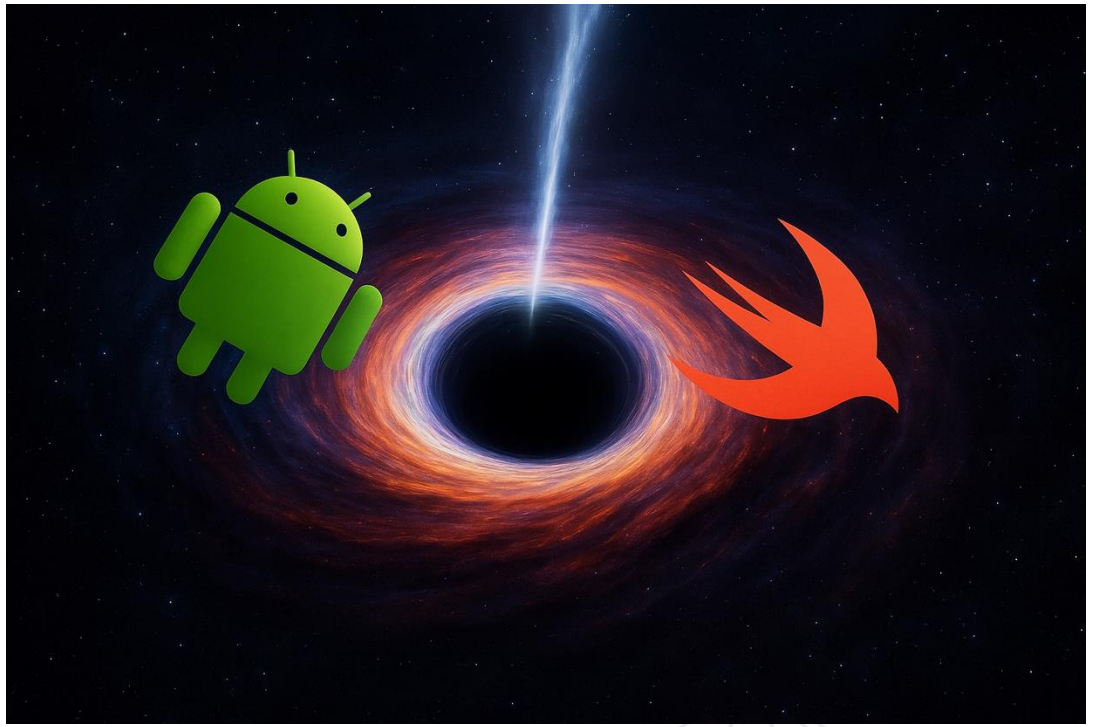
UI для такого приложения

- NDK Activity

C++ -> JNI -> Swift -> Android



Подведем итог



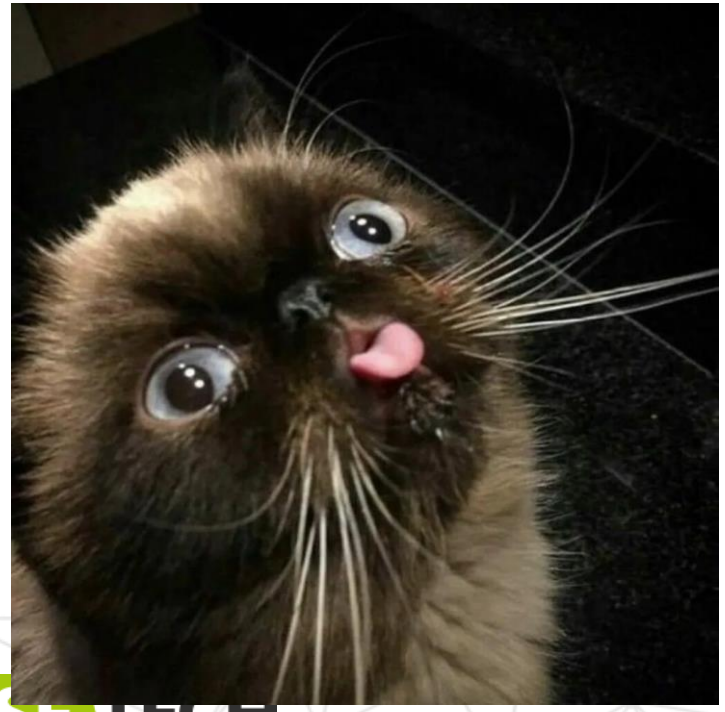
Swift Android SDK. Для чего

- Just for fun, пет-проекты, эксперименты, улучшение технологии…

- Расширение существующего iOS приложения на Android

Swift Android SDK. Когда не стоит

- Новый проект кросс-платформенного приложения – KMP, Flutter
- Серьезный большой проект (с CI/CD, Unit-тестами, UI-тестами…) - натив



Swift Android SDK. Что ожидаем

- Полная поддержка Async/await для Android
- Кросс-компиляция и поддержка тестов
- Расширение для VSCode
- https://github.com/orgs/swiftlang/projects/17/views/1

Список полезных источников

https://www.swift.org/documentation/articles/swift-sdk-for-android-getting-started.html

https://www.swift.org/blog/nightly-swift-sdk-for-android/

https://deepwiki.com/swiftlang/swift-java/

https://docs.oracle.com/en/java/javase/21/core/foreign-function-and-memory-api.html

https://github.com/swiftlang/swift-android-examples

https://gitverse.ru/azharkova/swift-droid-news-sample

https://github.com/swiftlang/swift-java

Спасибо за внимание!

@anioutkajarkova

azharkova
prettygeeknotes