Kotlin Multiplatform

Code sharing einfach gemacht

Daniel Bälz & Robert Zetzsche

GDG DevFest Karlsruhe 2023

Daniel Bälz

Freiberuflicher Android Entwickler

https://dbaelz.de

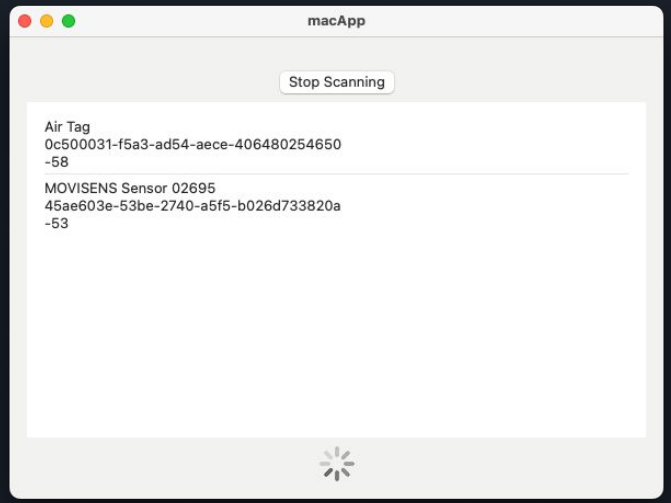
Robert Zetzsche

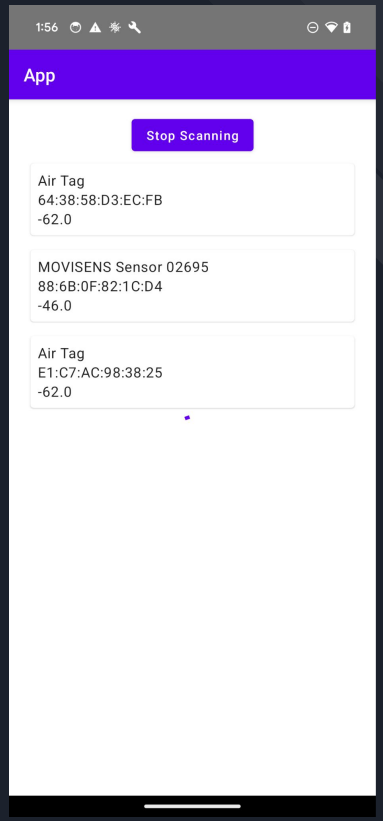
Senior Software Developer

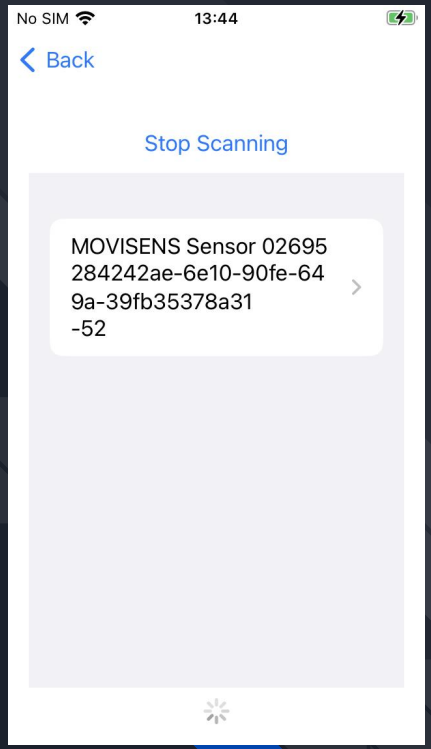
https://movisens.com

WERBUNG: Mobile Development Karlsruhe

https://www.meetup.com/de-DE/karlsruhe-mobile-development-meetup/


macApp
Stop Scanning
Air Tag
0c500031-f5a3-ad54-aece-406480254650
-58
MOVISENS Sensor 02695
45ae603e-53be-2740-a5f5-b026d733820a
-53


1:56
App
Stop Scanning
Air Tag
64:38:58:D3:EC:FB
-62.0
MOVISENS Sensor 02695
88:6B:0F:82:1C:D4
-46.0
Air Tag
E1:C7:AC:98:38:25
-62.0


No SIM
13:44
Back
Stop Scanning
MOVISENS Sensor 02695
284242ae-6e10-90fe-64
9a-39fb35378a31
-52

Kotlin Multiplatform

- Ziel: Code plattformübergreifend teilen
- Entwickelt von JetBrains
- Android, iOS, JVM, Native (macOS, Windows, Linux) und Web

Seit 1. November Stable

Kotlin Multiplatform

Current platform stability levels for Core Kotlin Multiplatform technology

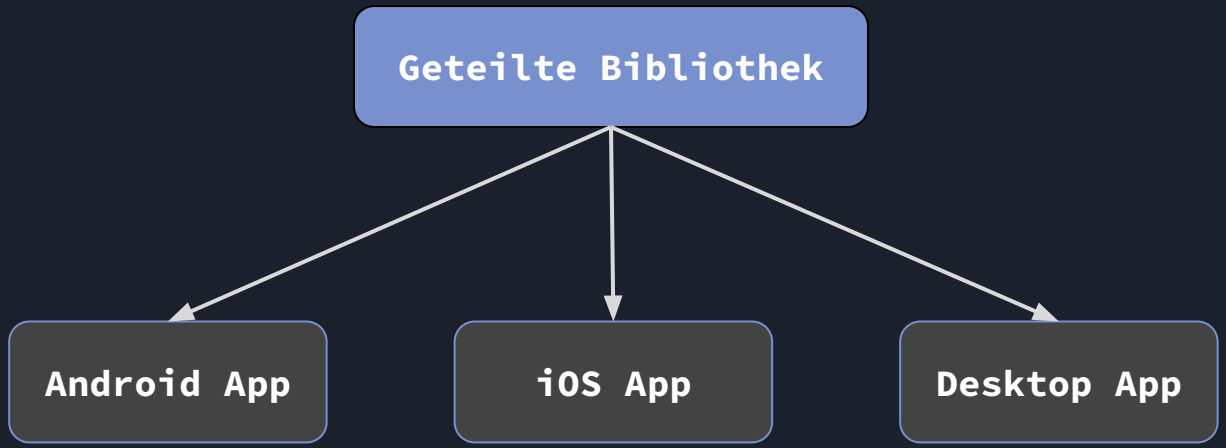
Platform	Stability level
Android	Stable
iOS	Stable
Desktop (JVM)	Stable
Server-side (JVM)	Stable
Web based on Kotlin/Wasm	Experimental
Web based on Kotlin/JS	Stable
watchOS	Best effort
tvOS	Best effort

Kotlin Multiplatform supports more native platforms than are listed here. To understand the level of support for each of them, see Kotlin/Native target support ↗.

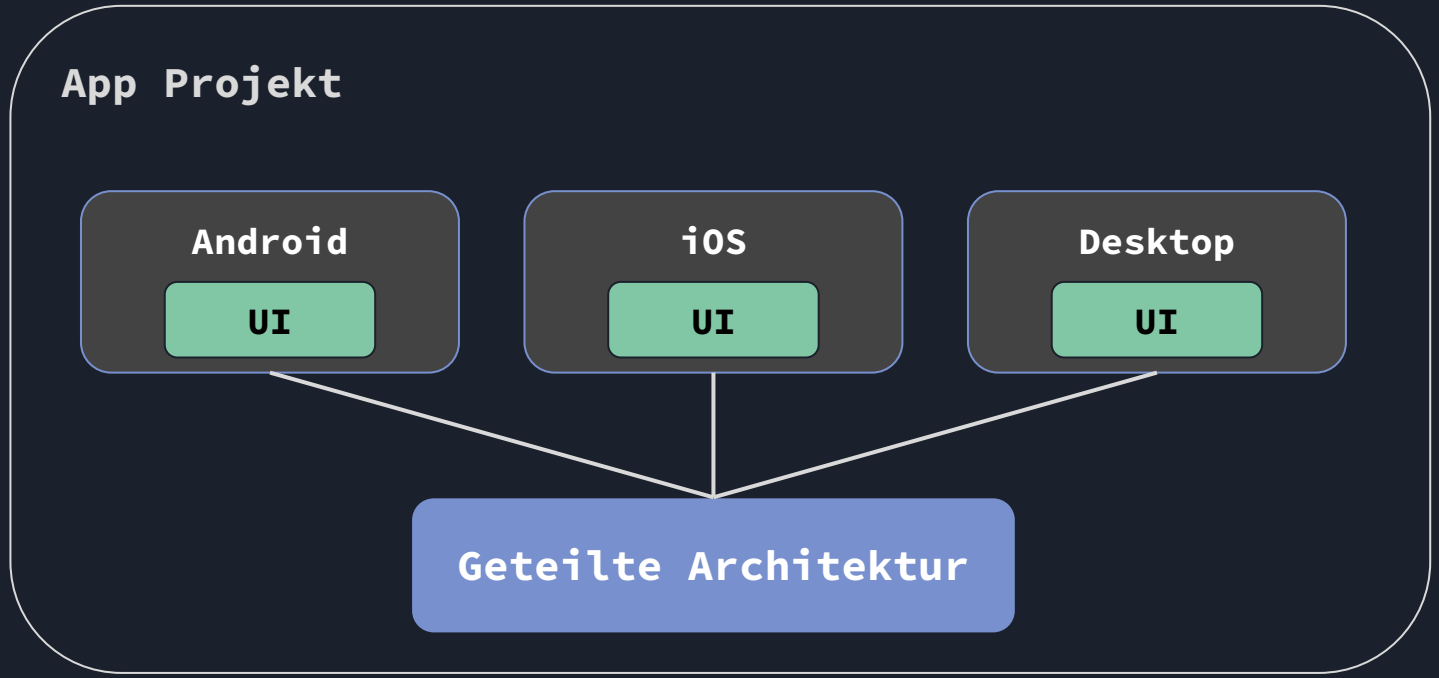
Quelle: https://www.jetbrains.com/help/kotlin-multiplatform-dev/supported-platforms.html

Umsetzungsstrategien Kotlin Multiplatform

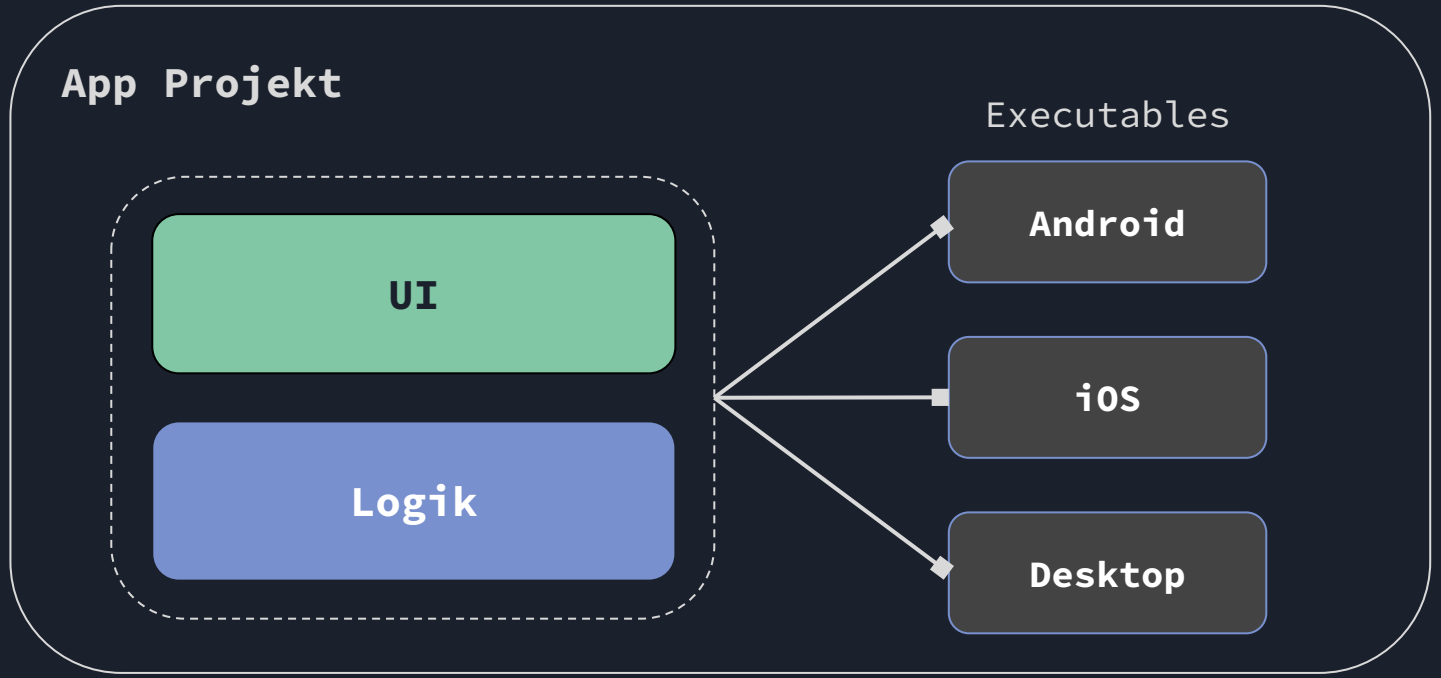
Geteilte Bibliothek/Modul



Geteilte Architektur



(Teilweise) Geteilte UI



WERBUNG:

13:30 Uhr - Raum 1

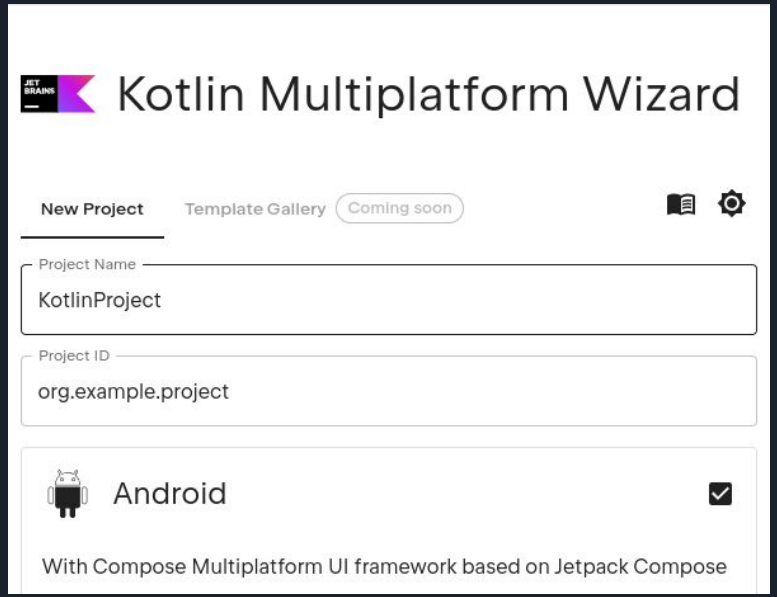
Compose Multiplatform:
UI sharing einfach gemacht

Getting started

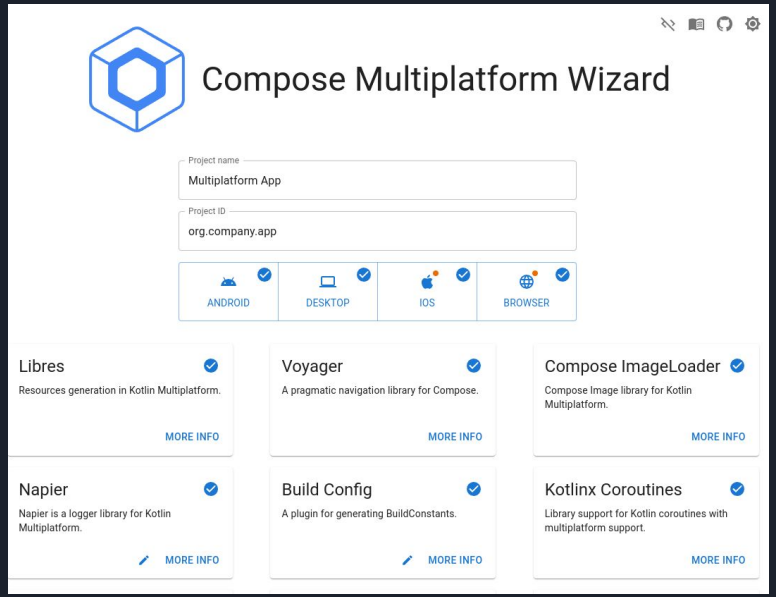
Getting started

- Android Studio/Intellij IDEA: Kotlin Multiplatform Mobile Plugin
- iOS & macOS Entwicklung
  - macOS zum Kompilieren der Anwendung mit Xcode
  - XCode Kotlin Plugin
  - Neue IDE: Fleet mit KMP Plugin
- Hilfreich: KMP Get Started Guide
- Open Source Projekte mit KMP

Neues Projekt starten



Offizielle KMP Wizard Webseite



Compose MP Wizard

...weitere Optionen

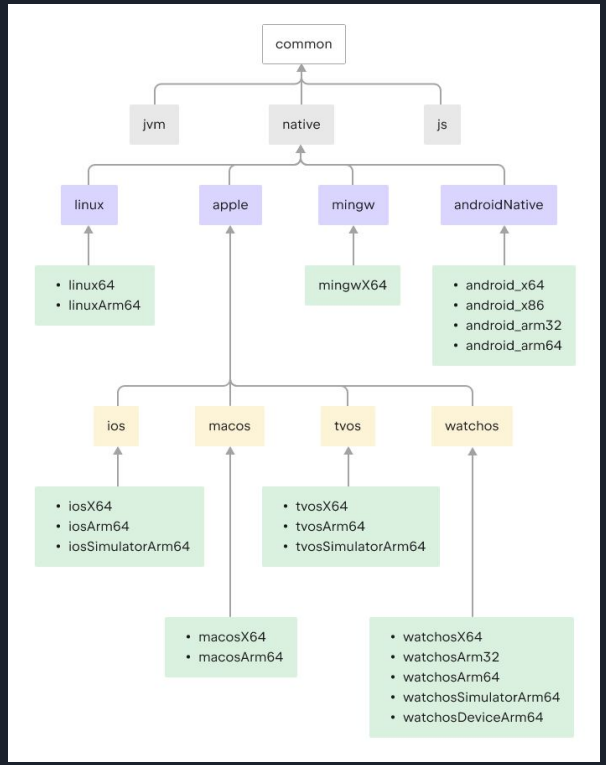
- Multiplatform Kickstarter Template
- KaMP Kit Template*
- Android Studio/IDEA Projekt Templates*

- Neu: Amper
  - Experimentelles Tool zur Projektkonfiguration für IDEA und Fleet
  - Gradle Plugin mit YAML Konfiguration

* noch nicht auf Kotlin 1.9.20

Kotlin Multiplatform Projekt

Standard Hierarchie



Kotlin/Native Zielplattformen

- Unterstützt große Anzahl von Zielplattformen, aber Unterschiede beim Support
  - Kompilierung und Ausführbarkeit auf CI getestet
  - Source und Binary Kompatibilität zwischen Compiler Releases
- Abstufung
  1. MacOS (X64, Arm64), iOS (X64), iOS Simulator (Arm64)
  2. iOS (Arm64), Linux (X64, Arm64), watchOS (X64, Arm32, Arm64_32), tvOS (X64, Arm64), watchOS Simulator (Arm64), tvOS Simulator (Arm64)
  3. Android Native (X86, X64, Arm32, Arm64), Windows (X64), watchOS (Arm64)

Verhalten pro Plattform

`Zwei Mechanismen`

- **`expect`** und **`actual`**
  - Beta
  - Implementierung pro Plattform auf Kotlin Seite
- **`Interface`**
  - Kann auf Kotlin oder Swift Seite implementiert werden
  - Setzt Service Locator oder DI voraus

expect und **actual**

```
// commonMain

expect fun getPlatform(): String


// androidMain

actual fun getPlatform() = "Android"


// iosMain

actual fun getPlatform() = "iOS"
```

Interoperabilität

Interoperabilität iOS

- Kotlin ↔ Obj-C
  - Kein direkter Swift Interop
- Nachteile:
  - Sealed Classes/Enums
  - Suspend Interop/Flows
  - Default Arguments
  - Generics
- Checked vs Unchecked Exceptions
- Lösung: Bibliotheken zum Interop
  - SKIE
  - KMP-Native-Coroutines
  - @HiddenFromObjC und @ShouldRefineInSwift

Memory Model und Garbage Collection

- Eigener Garbage Collector mitgeliefert
  - Stop-the-World Mark and Concurrent Sweep
  - Manuell steuerbar

- Einige Besonderheiten
  - Suspend Functions
  - Swift/Obj-C Objekte leben länger als sie sollten, wenn sie die Swift/Kotlin Grenze übertreten
    - Warten auf GC

Nachteile von Kotlin Multiplatform in iOS

- Langsame Kompilierung
- Nur nativer Obj-C Interop
  - Mit all den Nachteilen
  - Neue Libraries auf iOS Seite sind teilweise Swift only und daher nicht benutzbar
- Coroutines und Flows nur mit externen Libraries sinnvoll nutzbar
- Zusätzliche Garbage Collection Runtime durch KMP

Veröffentlichen von Artefakten

- Android
  - Android Projekt
  - Einbindung als jar oder aar möglich
- iOS
  - Zusätzliches XCode Projekt zum Bauen von ipa
  - Verschiedene Ansätze um shared Framework zu konsumieren
    - SPM, Cocoapod oder direkt im Projekt
- Desktop mittels Kotlin/Native
  - Einbindung als Bibliothek
  - Zielformate für App
    - Linux: AppImage, deb und rpm
    - macOS: dmg und pkg
    - Windows: exe und msi

Bibliotheken & Ökosystem

Bibliotheken und Ökosystem

- Immer größer werdende Anzahl an Kotlin Multiplatform Bibliotheken
  - Kotlin Foundation's Grants Program

- Zusätzliche Option plattformspezifische Lösungen zu verwenden

- Awesome Kotlin Multiplatform mit umfangreicher Liste

Bibliotheken und Ökosystem

Dependency Injection

Koin Kodein

Key/Value Store

Multiplatform Settings

Networking

Ktor kotlinx.serialization Okio

Logging

Kermit Napier

Architecture

KMM-ViewModel Decompose

Storage & Database

KStore SQLDelight Realm

Interoperability

SKIE KMP-NativeCoroutines MOKO KSwift

Roadmap

- Developer Experience
  - Fleet
  - Amper
- iOS
  - Kotlin ↔ Swift Interop
  - Beschleunigen des Kotlin Native Compilers
  - Verbesserung für Cocoapod und SPM
- Verbessern beim Entwickeln von Libraries
  - Rückwärtskompatibilität
  - Besserer Veröffentlichungsprozess

Fazit

- Kotlin Multiplatform hat viele Nutzungsszenarien
- Technologie mittlerweile stabil und gut einsetzbar
- Ökosystem wächst stetig
- Adaption ist abhängig von der Akzeptanz in (cross-funktionalen) Entwicklungsteams
- Kotlin/Native ist stabil, aber
  - Langsame Kompilierung
  - Zusätzliche Garbage Collection Runtime
  - Obj-C Interop
- Würden wir es schon einsetzen?

Ressourcen

- Get Started Guide
- Awesome Kotlin Multiplatform
- Webinare von JetBrains (21., 23., 28. und 30. November)

Bluetooth Kotlin Multiplatform

https://github.com/rzetzsche/BluetoothKotlinMultiplatform

Fragen?!