Kotlin Multiplatform: Effizientes Code-Sharing – jetzt auch mit KI!

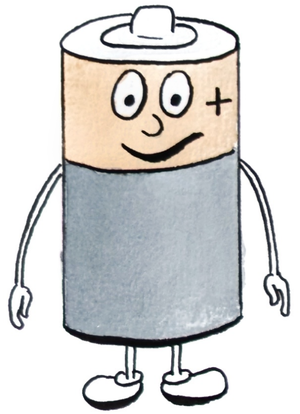
Alexander von Below
Michal Harakal

Deutsche Telekom AG, 2025

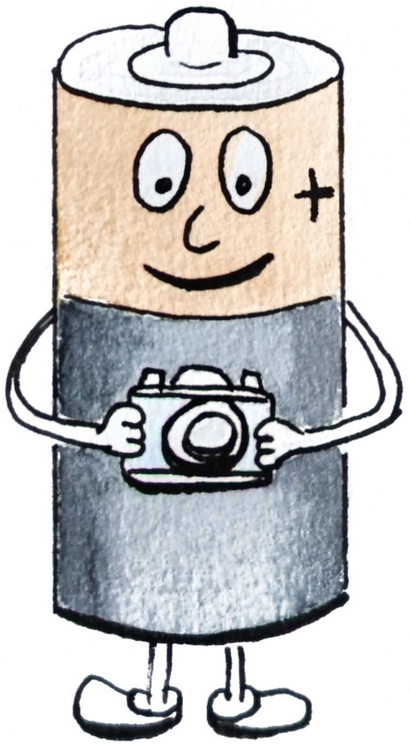
Kotlin Multiplatform: Effizientes Code-Sharing – jetzt auch mit KI!

Alexander von Below
Michal Harakal

Deutsche Telekom AG, 2025

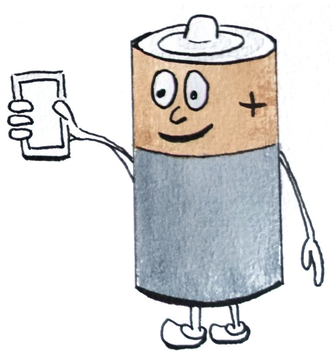


(C) Adriana Harakalova, 2024



CODE SHARING

KOTLIN



(C) Adriana Harakalova, 2024

KOTLIN

Code writing and reading

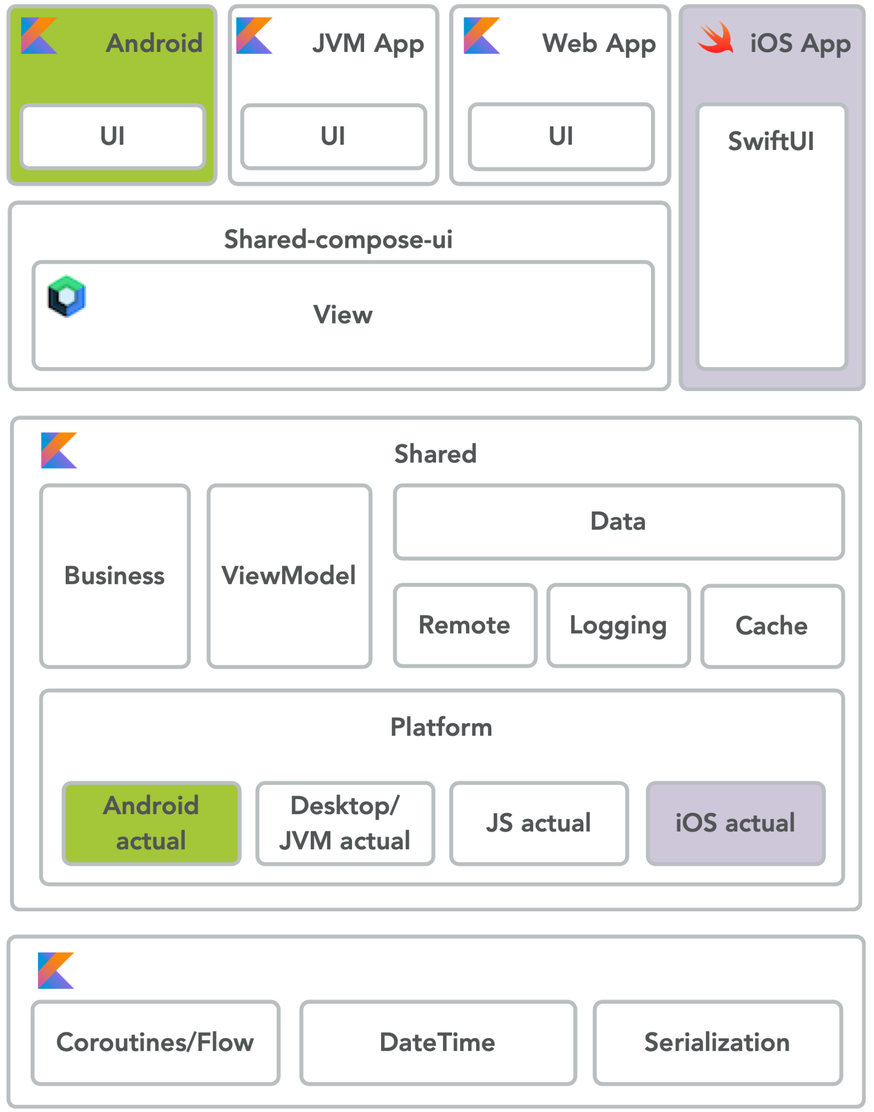
- Hier ist die Übersetzung auf Deutsch:
- Knappe und ausdrucksstarke Syntax
- Null-Sicherheit
- Datenklassen
- Vollständige Java-Interoperabilität
- Koroutinen und Unterstützung für asynchrone Programmierung

KOTLIN MULTIPLATFORM

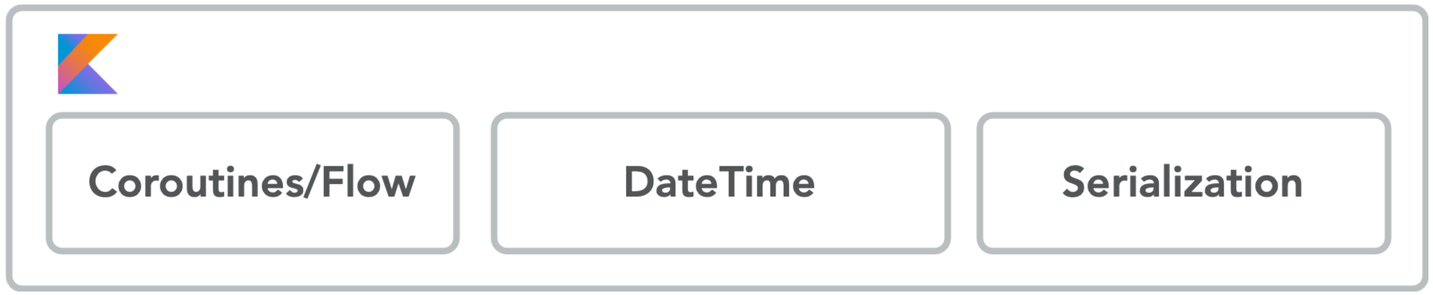
Source code sharing

- Plattformübergreifende Entwicklung
- Unterstützung nativer Performance – Kotlin Native
- Plattformübergreifende Benutzeroberflächen mit Compose Multiplatform

ARCHITECTURE

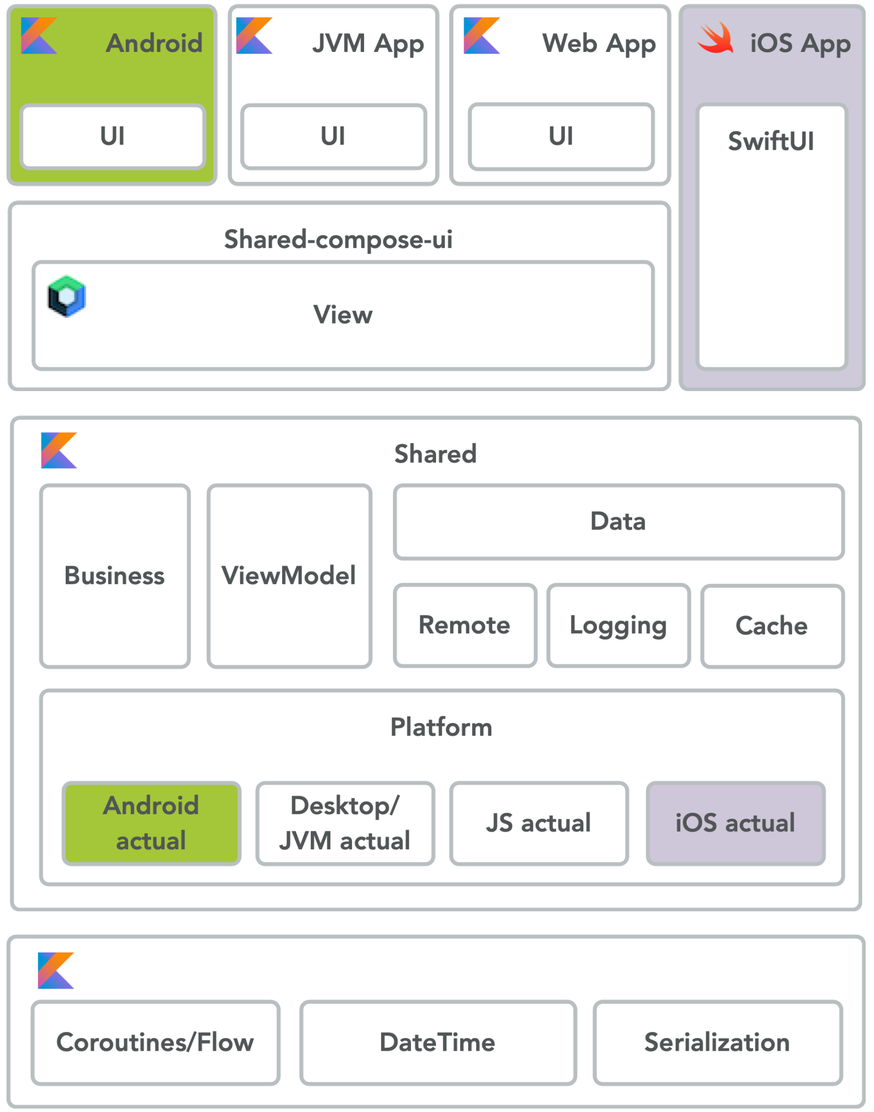


FOUNDATION

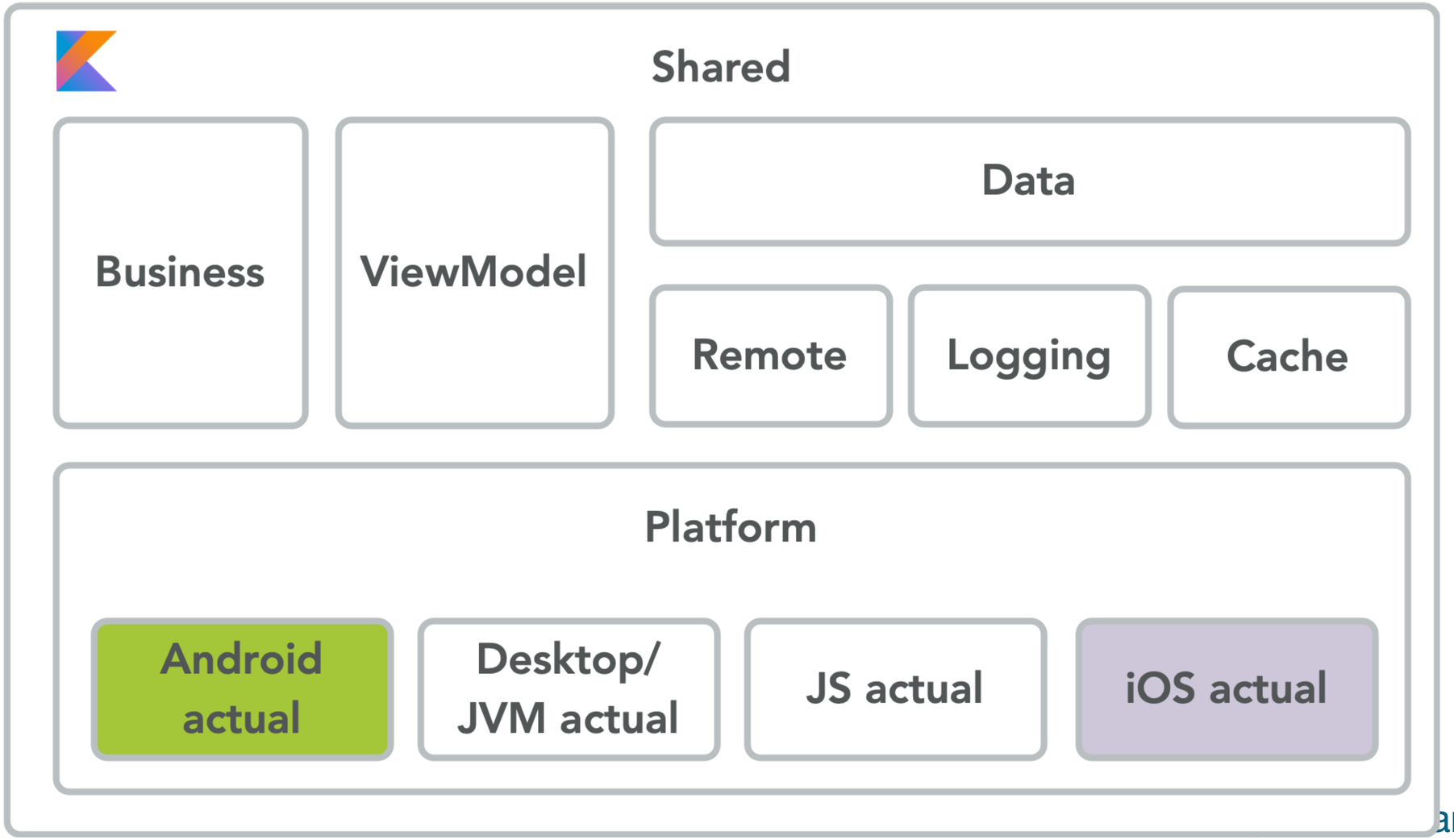


- Kotlin Standard Library
- Kotlin/Native
- Kotlinx libraries

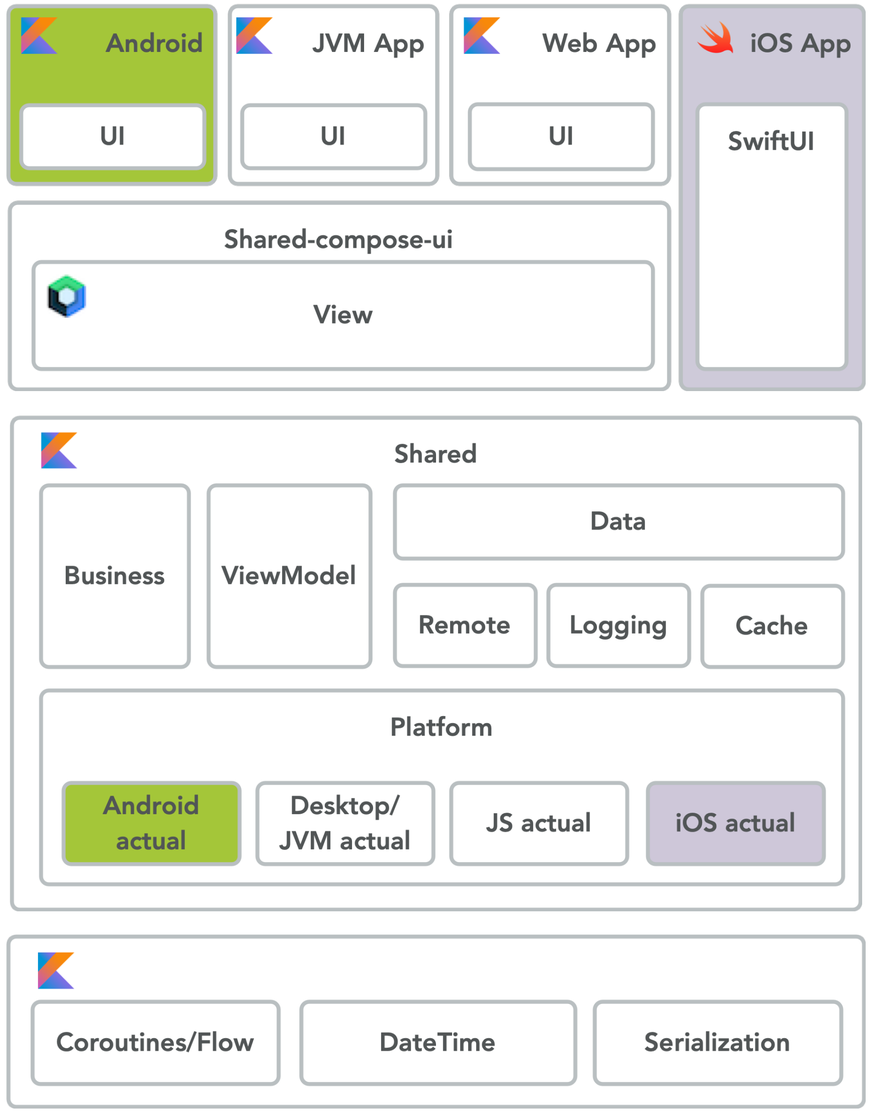
ARCHITEKTUR



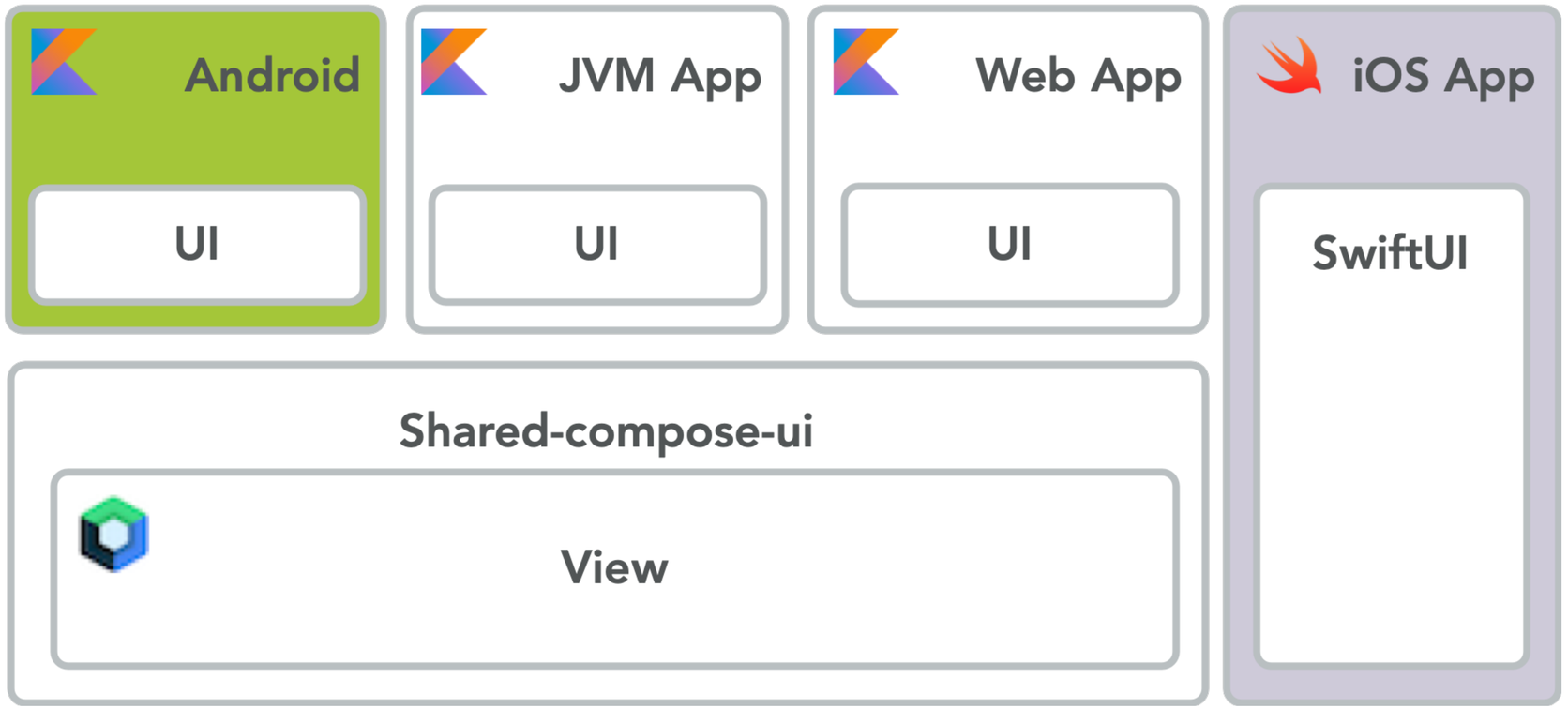
SHARED MODULE



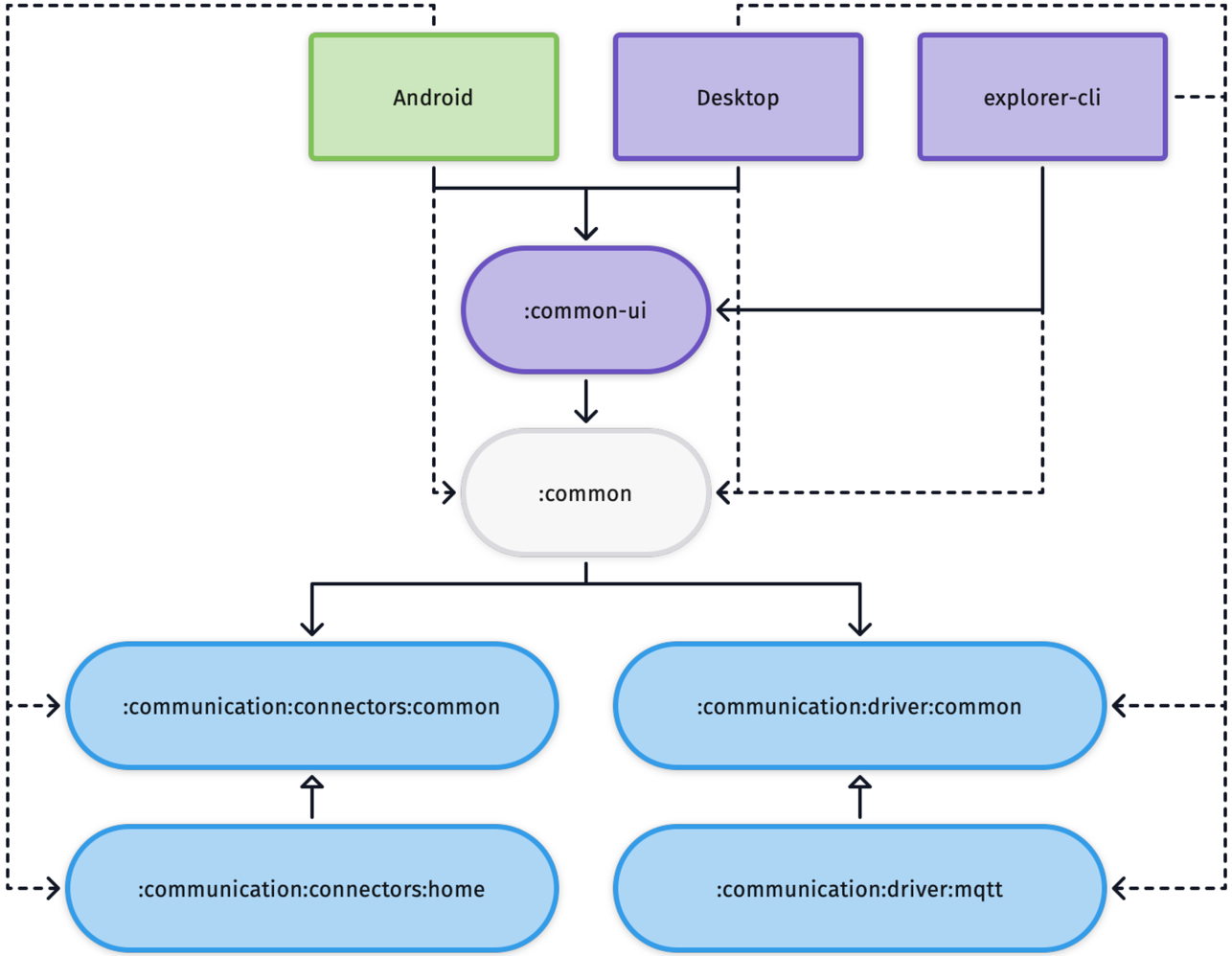
ARCHITECTURE

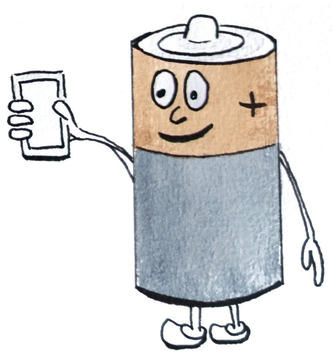


APPS


Android
UI
JVM App
UI
Web App
UI
iOS App
SwiftUI
Shared-compose-ui
View

GRADLE


Android
Desktop
explorer-cli
:common-ui
:common
:communication:connectors:common
:communication:driver:common
:communication:connectors:home
:communication:driver:mqtt



(C) Adriana Harakalova, 2024

Alexander von Below, Michal Harakal, 2025

Aber … warum ich?

„The dark side is a pathway to many abilities some consider to be unnatural“

Sheev Palpatine

KOTLIN

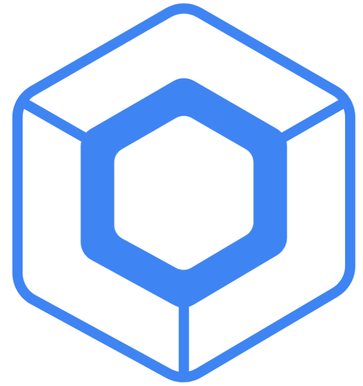
Das alte Ziel: Cross Platform

- „Write Once, Run Anywhere“
- Geringere Kosten
- Schnellere Entwicklung
- Keine Probleme

KOTLIN

Anforderungen

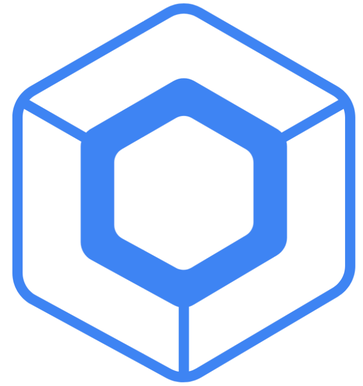
- Performance
- Optimale Gestaltung des User Interface
- Unterstützung der Platform



KOTLIN

Lernkurve

- Neue IDEs
- Neue Sprachen
- Neue Toolchains
- Neue Patterns



KOTLIN

Code writing and reading

- Modern, wie Swift
- Viele Sprachfeatures ähnlich zu Swift
- Liest sich fast wie Swift

KOTLIN

It’s magic!

```
suspend fun loadModel() { … }
```

```
try await sineCalc.loadModel()
```

KOTLIN

Kotlin Native

- Nicht mehr Java Virtual Machine
- Konan Compiler
- Frontend für llvm

K2

KOTLIN

Was brauche ich?

- Xcode
- Android Studio

✅

KOTLIN

Integration

- Direct als BuildPhase
- SwiftPackageManager
- Cocoapods

KOTLIN

SwiftPackage Manager

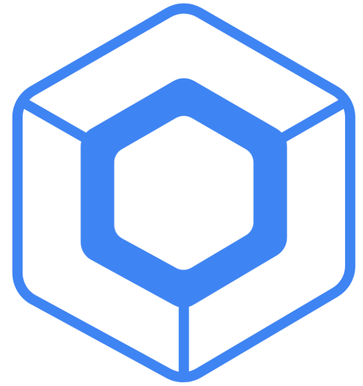
✓Binary: xcframework

✓Versionskontrolle

✓Besonders gut für grosse Projekte

- Schlecht zu debuggen

- Schwierig zu ändern



KOTLIN

Direkte Integration

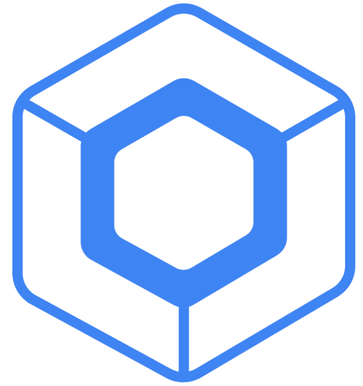
✓Wird in Xcode gebaut

✓Gut zu debuggen

✓Leicht zu ändern

✓Besonders für Mono-Repositories

- Setup schwer zu ändern



KOTLIN

Cocoapods

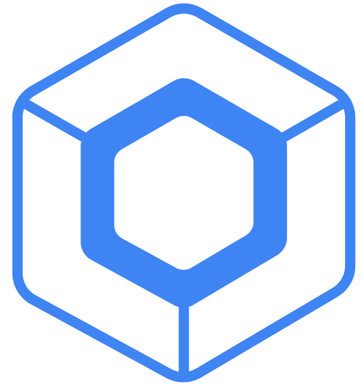
✓Wird in Xcode gebaut

✓Gut zu debuggen

✓Leicht zu ändern

✓Mögliche Versionskontrolle

- Cocoapods: Maintenance Mode (2024)

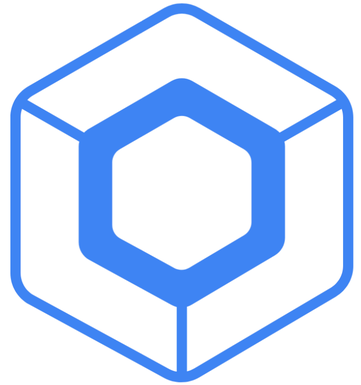


KOTLIN

Geheime Zutat

https://github.com/touchlab/xcode-kotlin

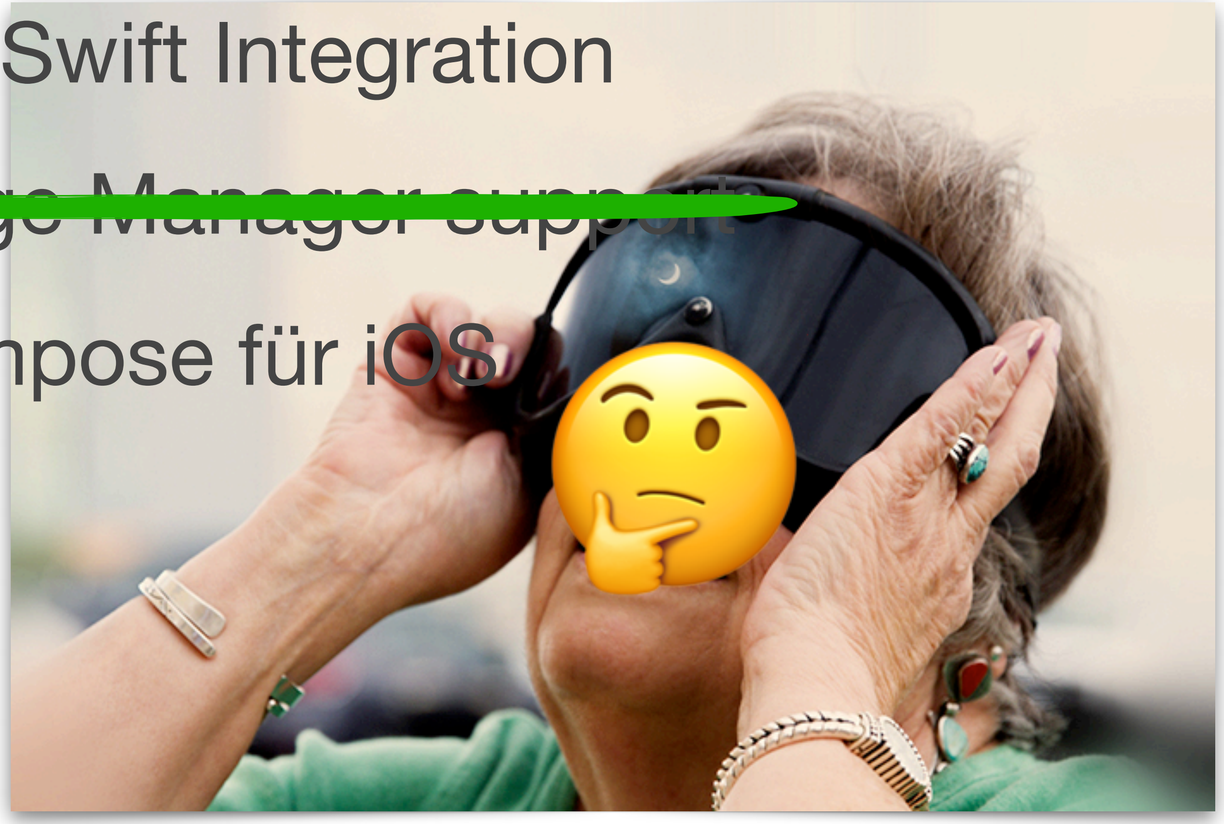
```
brew install xcode-kotlin
```



KOTLIN

The Future

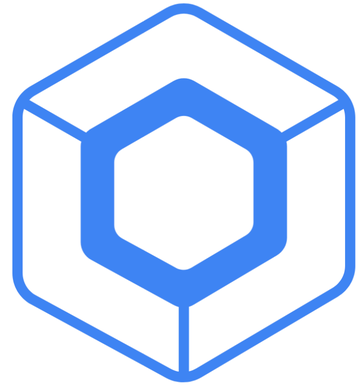
- Viel bessere Swift Integration
- ~~Swift Package Manager support~~
- Jetpack Compose für iOS



KOTLIN

Konklusion

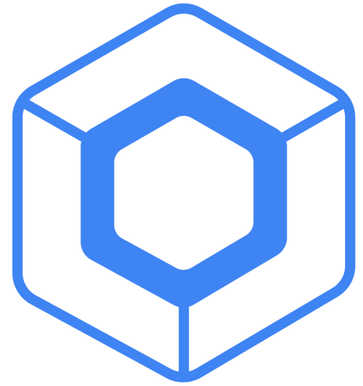
- ✓ Geringe Lernkurve
- ✓ Platformexpertise ist da
- ✓ Bekannte Toolchains
- ✓ Bekannte Patterns



KOTLIN

Ready for Production?

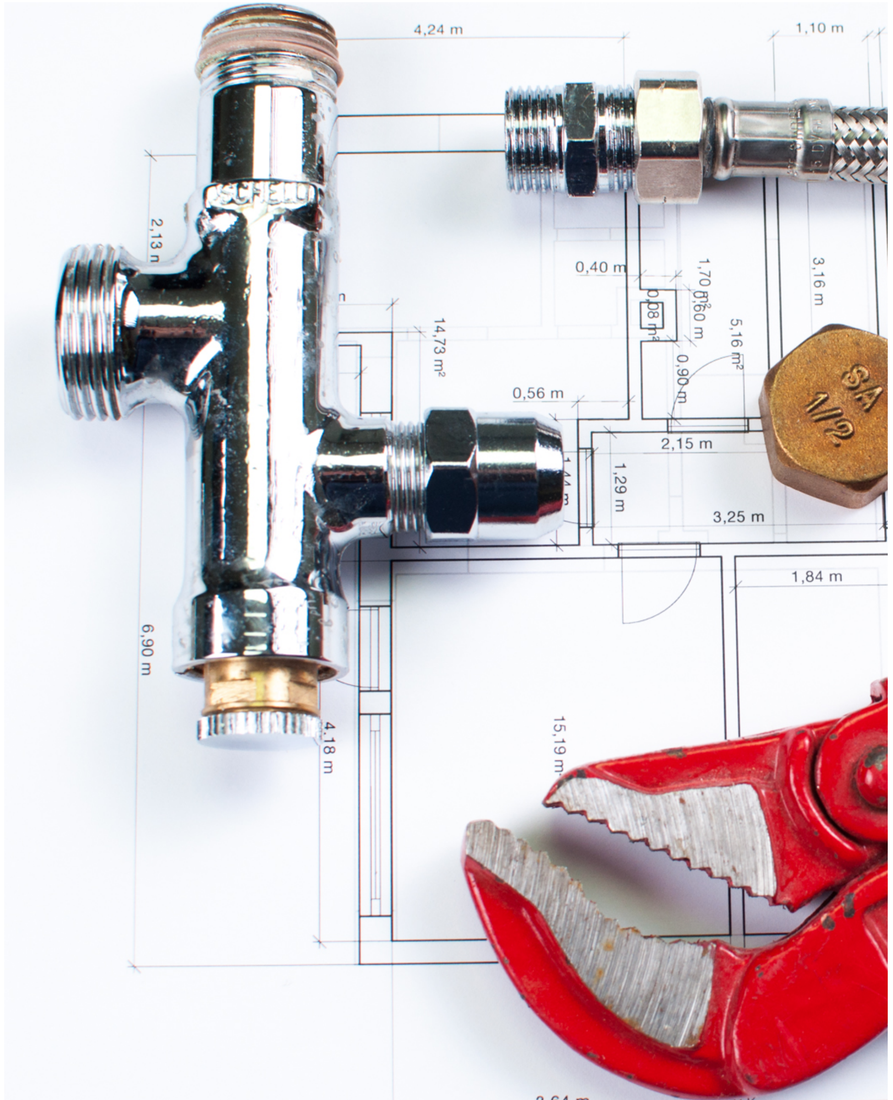
- Multiplatform projects are now **Stable**!
- Einige Libraries, kotlinx.io, noch experimental
- Toller Support durch die Community und JetBrains



HTTPS://KLIBS.IO => 1600+

KMP BIBLIOTHEKEN

VERDRAHTUNG



- Dependency Injection
  - **Koin**
  - Kodein

ZEIT

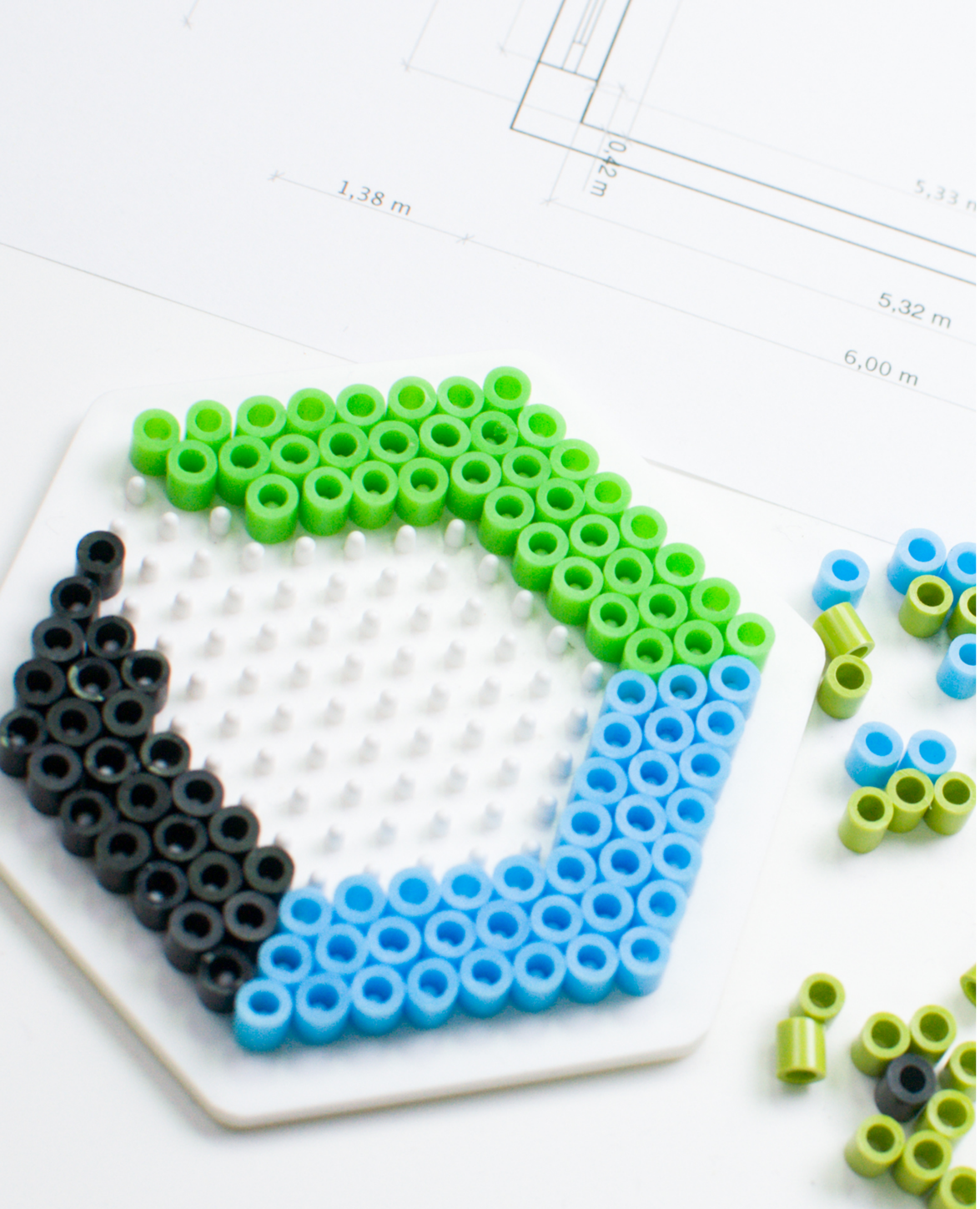
- kotlinx.datetime

LOCALE STORAGE

- Speichert serialised JSON objects als String
  - **SharedPreferences** auf Android
  - **NSUserDefaults** auf iOS
- Für größere Datenmengen
  - **Sqldelight** - Square
  - **Room** - Google
- **kotlinx.io -** 0.7.0

NETZWERK KOMMUNIKATION

- **KTOR** networking client
  - Natives HTTP Klient
- JSON serialisieren/deserialisieren
  kotlinx.serializations



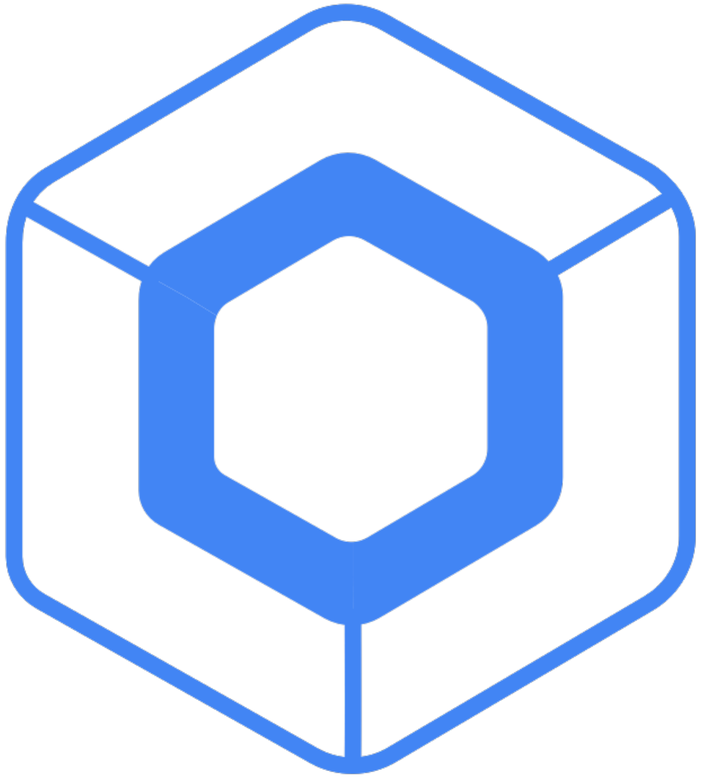
UI CODE SHARING

JETPACK COMPOSE

JETPACK COMPOSE

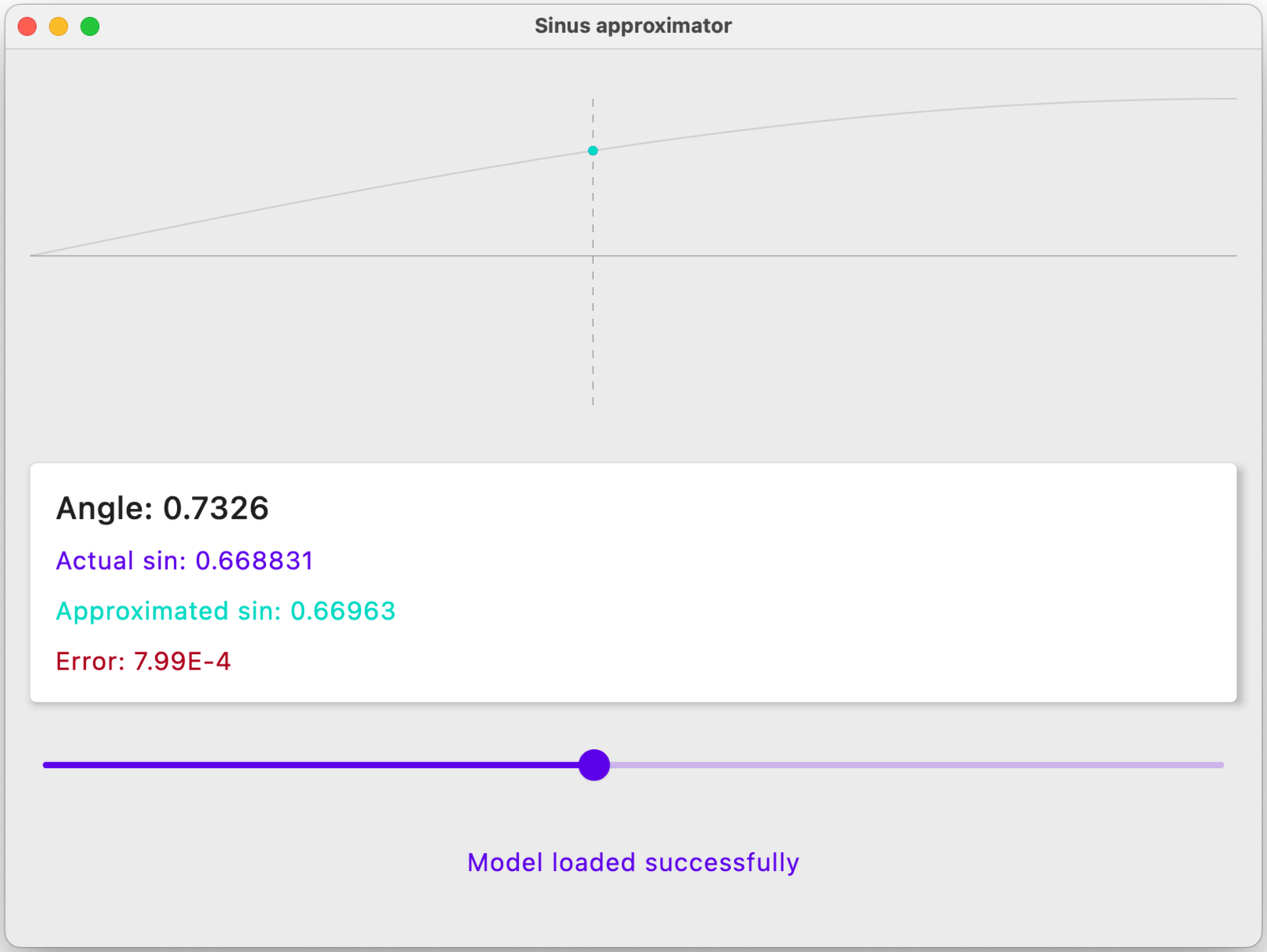
- modern reactive declarative UI framework by Google
- stable since Juli 2021
- Powered by Kotlin Compiler Plugin
- Portable

COMPOSE MULTIPLATFORM



- Android
- iOS
- desktop
- web

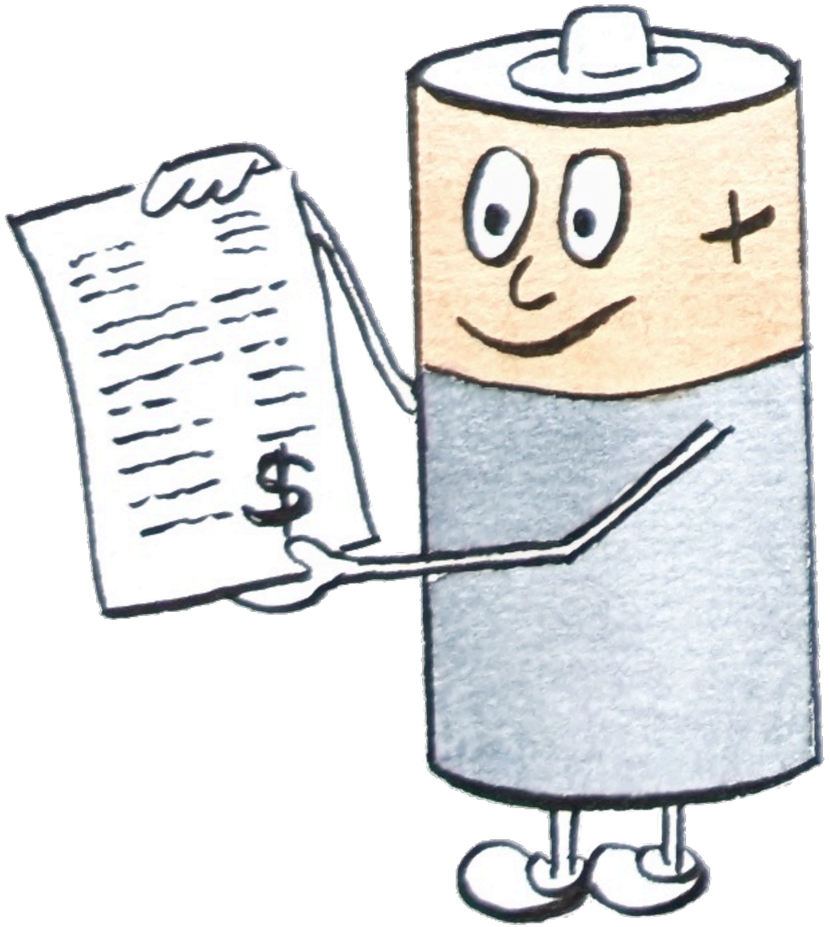
COMPOSE MULTIPLATFORM



WIR MÜSSEN ÜBER CHATGPT & CO REDEN

Alexander von Below, Michal Harakal, 2025

WIR MÜSSEN ÜBER CHATGPT & CO REDEN



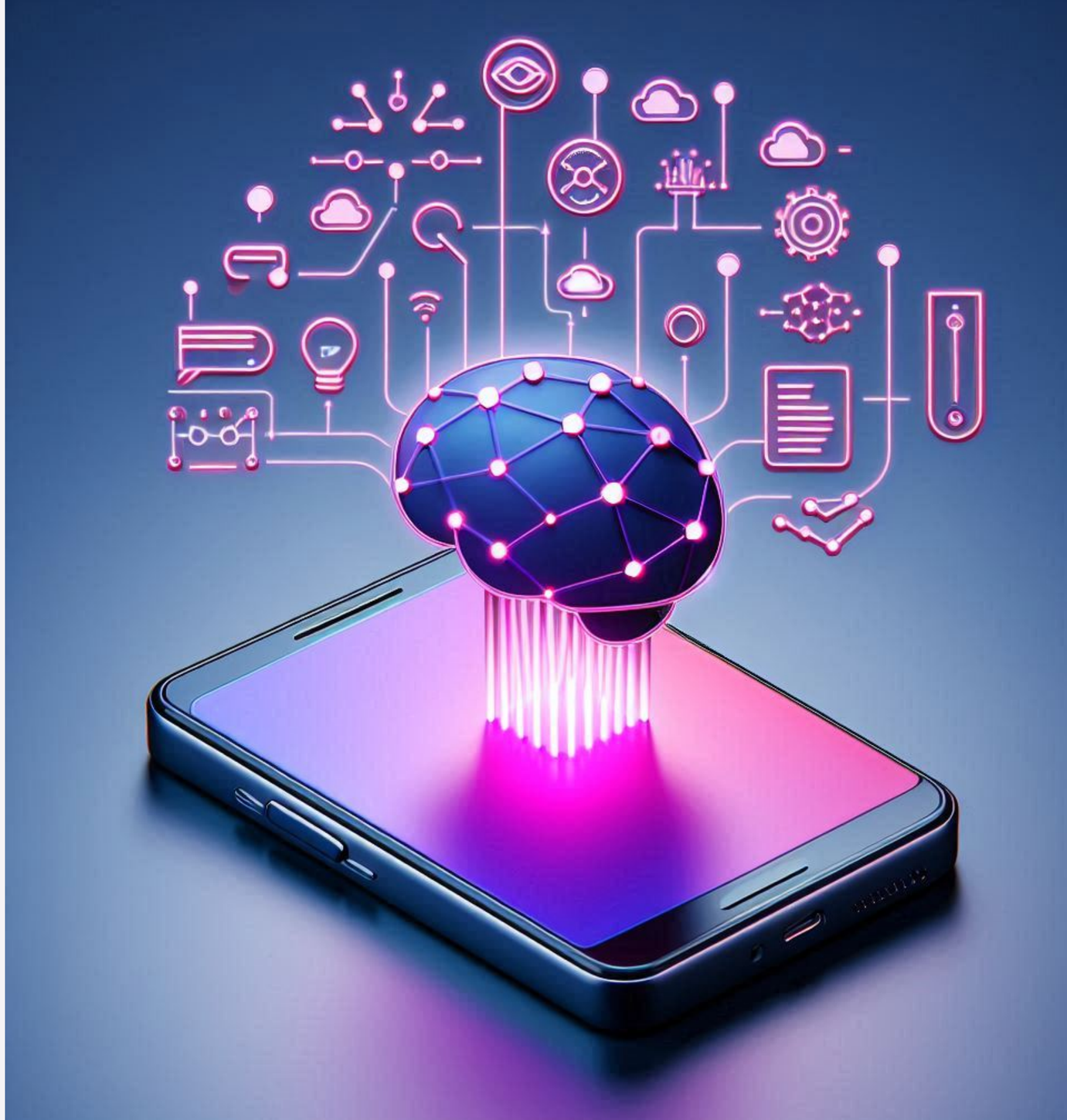
(C) Adriana Harakalova, 2024

NACHTEILE VON CLOUD KI

BLERP

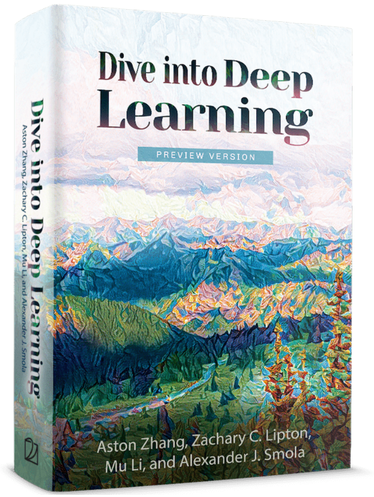
- Bandbreite
- Latenz
- Economics /Wirtschaftlichkeit
- Reliability/Zuverlässigkeit
- Privacy /Datenschutz

ON-DEVICE AI



KI MIT JAVA

Jlama



Deep Java Library(DJL)

Llama3.java

Eclipse Deeplearning4j

Alexander von Below, Michal Harakal, 2025

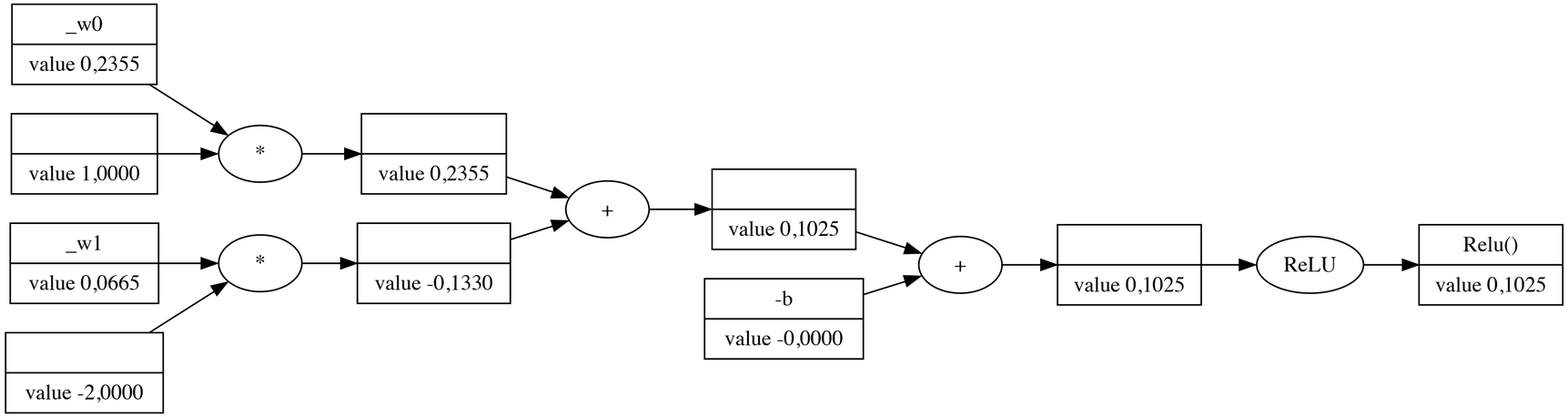
ML MIT KOTLIN MULTIPLATFORM

MIKROGRAD

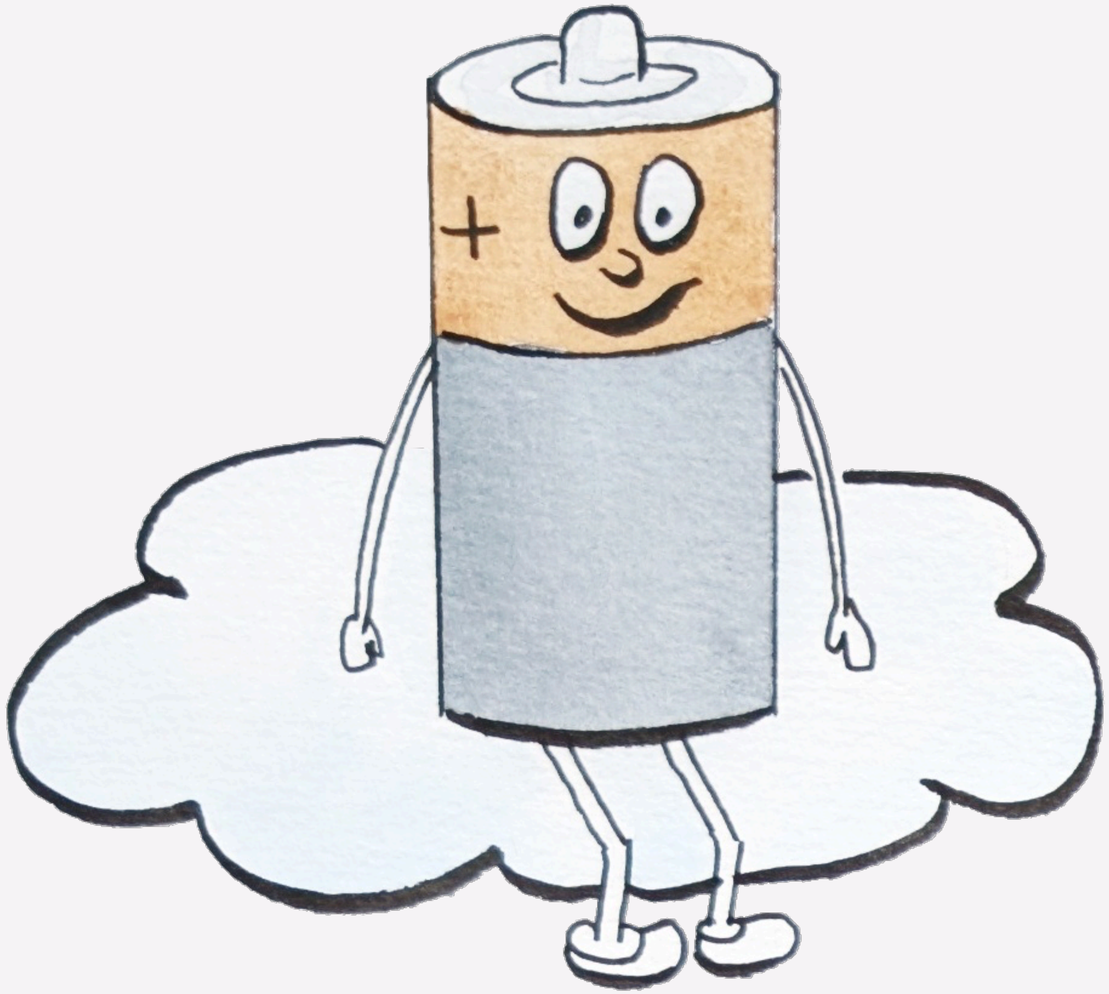
```
import org.mikrograd.diff.Value
import org.mikrograd.diff.div
import org.mikrograd.diff.plus

val a = Value(-4.0)
val b = Value(2.0)
var c = a + b
var d = a * b + b.pow(3.0)
c += c + 1
c += 1.0 + c + (-a)
d += d * 2 + (b + a).relu()
d += d * 3.0 + (b - a).relu()
val e = c - d
val f = e.pow(2.0)
var g = f / 2
g += 10.0 / f
println("$g")  // prints 24.7041, the outcome of this forward pass
g.backward()
println("${a.grad}") // prints 138.8338, i.e. the numerical value of dg/da
println("${b.grad}") // prints 645.5773, i.e. the numerical value of dg/db```
```

MIKROGRAD

Compute graph



DEMO




+
Battery

Danke

https://github.com/michalharakal/javaland-2025

Alexander von Below, Michal Harakal, 2025

LINKS

Sources and links

- Attention Is All You Need - https://arxiv.org/pdf/1706.03762
- https://github.com/michalharakal/micrograd
- https://lmos-ai.github.io/arc/
- https://github.com/kindxiaoming/pykan
- https://commons.wikimedia.org/wiki/File:The-Transformer-model-architecture.png
- https://github.com/michalharakal/gradienttracer
- https://deeplearning4j.konduit.ai/
- https://github.com/tjake/Jlama
- https://docs.quarkiverse.io/quarkus-langchain4j/dev/llama3.html
- https://github.com/deeplearning4j

HINWEIS ZU LOGOS UND MARKENZEICHEN

Sources and links

Alle verwendeten Logos, Markenzeichen und Markenabbildungen sind Eigentum ihrer jeweiligen Inhaber. Ihre Verwendung erfolgt ausschließlich zur Illustration und Darstellung in diesem Kontext.

Bilder auf Slides 32, 33, 34, 35, 36 - Adriana Harakalova (c) 2022

Slide 43 - generiert mit ChatGPT