

# **Titel**

Untertitel

## **Bachelorarbeit**

Eingereicht in teilweiser Erfüllung der Anforderungen zur Erlangung des akademischen Grades:

### **Bachelor of Science in Engineering**

an der FH Campus Wien

Studienfach: Computer Science and Digital Communications

#### **Autor:**

Markus Mayer

#### **Matrikelnummer:**

52006537

#### **Betreuer:**

MSc René Goldschmid

#### **Datum:**

25.09.2022

Erklärung der Urheberschaft:

Ich erkläre hiermit diese Bachelorarbeit eigenständig verfasst zu haben. Ich habe keine anderen Quellen, als die in der Arbeit gelisteten verwendet, noch habe ich jegliche unerlaubte Hilfe in Anspruch genommen

Ich versichere diese Bachelorarbeit in keinerlei Form jemandem Anderen oder einer anderen Institution zur Verfügung gestellt zu haben, weder in Österreich noch im Ausland.

Weiters versichere ich, dass jegliche Kopie (gedruckt oder digital) identisch ist.

Datum:

Unterschrift:

# Abstract

(E.g. “This thesis investigates...”)

# Kurzfassung

(Z.B. “Diese Arbeit untersucht...”)

# Abkürzungen

OHA	Open Handset Alliance
OAA	Open Automotive Alliance
LiPS	Linux Phone Standards Forum
OMA	Open Mobile Alliance
IDE	Integrated Development Enviroment

# Schlüsselbegriffe

- 1
- 2
- 3

# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Relevanz . . . . .	1
<b>2</b>	<b>Hauptteil</b>	<b>2</b>
2.1	Android-Übersicht . . . . .	2
2.2	Android Jetpack-Übersicht . . . . .	3
2.3	Android Jetpack-Komponenten . . . . .	5
<b>3</b>	<b>Schluss</b>	<b>6</b>
3.1	Zusammenfassung . . . . .	6
3.2	Bewertung und Ausblick . . . . .	6
	<b>Bibliographie</b>	<b>7</b>
	<b>Abbildungen</b>	<b>8</b>
	<b>Tabellen</b>	<b>9</b>
	<b>Appendix</b>	<b>10</b>

# **1 Einführung**

## **1.1 Motivation**

Warum ist diese Arbeit interessant?

## **1.2 Relevanz**

Inwiefern ist diese Arbeit neu und wichtig?



## 2 Hauptteil

### 2.1 Android-Übersicht

Um einen Überblick über Android zu bekommen, beziehungsweise wo der Fokus des am weitest verbreitetsten Betriebssystem für mobile Geräte liegt, muss man etwas in die Vergangenheit blicken. Android ist ein Open-Source-Betriebssystem, welches auf einem Linux-Kernel basiert. Es wurde von der Firma Android Inc. entwickelt, die im Jahr 2005 von Google übernommen wurde [1].

#### 2.1.1 Rückblick

Im Jahr 2006 hatten Hersteller, die ein Smartphone auf den Markt bringen wollten, nur zwei ziemlich teure Möglichkeiten. Entweder Lizenzgebühr für ein Betriebssystem bezahlen oder ein eigenes Betriebssystem entwickeln. Google gründet 2007 die Open Handset Alliance (OHA) und führt Android als Open Source-Plattform ein. Im Jahr 2008 wird Android auf die Version mit dem Namen Cupcake aktualisiert. Bei dieser Version können Hersteller wie HTC und Samsung, aber auch Mobilfunkanbieter das Betriebssystem ihrer Smartphones anpassen. Laut Google kommt 2010 Android auf 34 Mobilgerätetypen in 49 Ländern zum Einsatz und bietet damit den Nutzern eine größere Auswahl als je zuvor. 2011 veröffentlicht Google mit Android 3.0 Honeycomb ein für Tablets optimiertes Betriebssystem, das als Basis für das Betriebssystem Fire OS des Kindle Fire-Tablets von Amazon dient. Der sogenannte Android Market wird 2012 in Google Play umbenannt und Entwickler können weiterhin ihre Apps in wenigen Stunden veröffentlichen. Zusätzlich zur Unterstützung von Mobilgeräten wird 2014 die Open Automotive Alliance (OAA) gegründet, deren Ziel es ist Android auch für Autos zu nutzen. Heute sind 45 führende Automarken Mitglied der OAA. Durch die Innovationskraft einiger Hersteller sind 2015 inzwischen Android Geräte für weniger als 50 US-Dollar verfügbar. Mit Stand 2016 gibt es fast 24.000 Android Geräte von über 1.300 Marken. Außerdem gibt es heute dutzende weltweit operierende Stores für Android-Apps. [2]

Androids Ziele sind zwar ähnlich zu denen des Linux Phone Standards Forum (LiPS) oder der Open Mobile Alliance (OMA), da aber der Ansatz des kompletten Software-Stacks von Android weit über den Fokus dieser standarddefinierenden Organisationen hinausgeht, ist Android kein Teil dieser Organisationen. Verglichen mit dem iPhone, dass eine vollständig herstellerspezifische Hardware- und Softwareplattform ist, die von einem einzigen Unternehmen herausgegeben wird, ist Android ein Open Source-Software-Stack, der von der OHA entwickelt und unterstützt wird und auf jedem kompatiblen Gerät funktionieren soll. [3]

#### 2.1.2 Programmiersprachen

Bis 2019 war Java die offizielle Standardprogrammiersprache für die Entwicklung von Android Apps. Als 2017 Kotlin als unterstützte Sprache für die Android Entwicklung angekündigt wurde, gab es zu Beginn große Aufregung unter den Entwicklern. Jedoch ist die Anzahl der Entwickler die Kotlin verwenden in kürzester Zeit so gestiegen, dass im Jahr 2019 Kotlin, Java als offizielle Standardprogrammiersprache ablöste. [4]

Mit Javas Ablöse als Standardprogrammiersprache wurde seitens Android jedoch weiterhin der offizielle Support für Java und C++ zugesichert [5]. Das Feedback, dass direkt von den Entwicklern kommt hebt folgende Vorteile bei der Verwendung von Kotlin hervor [6].

- **Ausdrucksstark und prägnant**  
Es kann mit weniger Code mehr erreicht werden. Codesegmente, die ohne Änderung immer wieder verwendet werden, fachsprachlich auch Boilerplate-Code genannt, können merkbar reduziert werden. Somit kann Kotlin die Produktivität der Entwickler erhöhen.
- **Sicherer Code**  
Kotlin enthält viele Sprachfunktionen, die dabei helfen häufige Programmierfehler, wie zum Beispiel Null Pointer Exceptions zu vermeiden. Android Apps mit Kotlin Code haben eine 20-prozentige geringer Absturzwahrscheinlichkeit.
- **Interoperabel**  
Kotlin zu 100 Prozent interoperabel mit der Programmiersprache Java, das heißt das Java Code in Kotlin verwendet werden kann und Kotlin Code auch problemlos von Java aus verwendet werden kann. Damit können die Entwickler so viel oder so wenig Kotlin Code in ihren Projekten verwenden, wie sie möchten.
- **Strukturierte Gleichzeitigkeit**  
Mit Kotlin Coroutines ist das Arbeiten mit asynchronem Code genauso einfach wie mit blockierendem Code. Diese Coroutines vereinfachen die Verwaltung von diversen Hintergrundaufgaben.

### 2.1.3 Android Studio

Android Studio ist seit 2013 die offizielle Integrated Development Environment (IDE) von Google für die Android-Softwareentwicklung. Um Android Studio zu entwickeln, ging Google eine Kooperation mit der tschechischen Firma JetBrains ein, die zu diesem Zeitpunkt mit IntelliJ IDEA eine der fortschrittlichsten Java IDEs am Markt hatten und auch heute noch haben. Android Studio basiert auf dieser leistungsstarken Entwicklungsumgebung, die von Google mit speziellen Funktionen für die Android-Entwicklung erweitert und angepasst wurde. Eine dieser Funktionen ist ein Build-System, das auf Gradle basiert und mit dem Android Gradle Plugin um einige Features erweitert wurde. Die aktuelle Version ist Android Studio Dolphin mit dem Versionscode 2021.3.1. Android Studio ist ein Open Source Projekt und steht auch frei zum Download zur Verfügung. [7]

### 2.1.4 Gradle

## 2.2 Android Jetpack-Übersicht

Android Jetpack wurde von der Support Library inspiriert, dabei handelt es sich um eine Reihe von Komponenten, die es erleichtern neue Android Funktionen zu nutzen und dabei trotzdem eine Abwärtskompatibilität aufrecht zu erhalten. Nahezu jede App im Play Store verwendete diese Support Library. Aufgrund dieser Beliebtheit wurden 2017 die Architecture Components eingeführt. Sie erleichtern den Umgang mit Daten bei Änderungen und Komplikationen im Lebenszyklus einer App. Android Jetpack bringt die existierende Support Library und die Architecture Components zusammen und gliedert sie in vier Kategorien, wie in Abbildung 2.1 zu sehen ist. [8]



Abbildung 2.1: Überblick über die Komponenten von Android Jetpack [8]

Android Jetpack ist eine Sammlung von Bibliotheken und soll Entwicklern dabei helfen, Best Practices zu befolgen, Boilerplate-Code zu reduzieren und Code zu schreiben, der über alle Android-Versionen und -Geräte hinweg konsistent funktioniert [9]. Da die Komponenten so aufgebaut sind, dass ihre Funktionalität versionsunabhängig zur Verfügung steht, ist die Abwärtskompatibilität gewährleistet [8]. Damit haben Entwickler mehr Zeit sich auf den Code zu konzentrieren, der für ihre App wichtig ist, um leichter und schneller, robuste und hochwertige Apps zu entwickeln.

Android Jetpack verwaltet also Aktivitäten wie Hintergrundaufgaben, Navigation und Lebenszyklusmanagement. Jetpack ist für eine gute Zusammenarbeit mit Kotlin konzipiert und zusätzlich kann mit der Komponente Android KTX einiges an Code eingespart werden. Die von Jetpack verwalteten Aktivitäten, Kotlin und die Komponenten Android KTX tragen daher gemeinsam dazu bei den Boilerplate-Code zu eliminieren. [10]

Die Android Jetpack Komponenten sind als sogenannte „unbundled“ Libraries bereitgestellt und sind nicht Teil der Android-Plattform. Dadurch können Updates der Libraries auch unabhängig und öfter durchgeführt werden [11]. Diese Libraries wurden alle in den neuen androidx.\* Namespace verschoben und können einzeln oder auch in Kombination verwendet werden. [8] Daher ist AndroidX eine wesentliche Verbesserung zu der ursprünglichen Android Support Library, welche nicht mehr gewartet wird [12].

## **2.3 Android Jetpack-Komponenten**

### **2.3.1 Core**

### **2.3.2 Activity**

### **2.3.3 Lifecycle**

### **2.3.4 Compose**

### **2.3.5 Navigation**

### **2.3.6 Room**

## **3 Schluss**

### **3.1 Zusammenfassung**

Wichtigsten Aussagen der Arbeit wiederholen und miteinander in Beziehung bringen

### **3.2 Bewertung und Ausblick**

Bewertung der verschiedenen Faktoren und Erkenntnisse Ideen für weitere Arbeiten

# Literaturverzeichnis

- [1] R. Hrmo, “Android mobile media platform.” 2
- [2] Android, “Der einfluss von android im laufe der jahre,” zugriff am 29.11.2022. [Online]. Available: [https://www.android.com/intl/de\\_de/everyone/](https://www.android.com/intl/de_de/everyone/) 2
- [3] R. Meier, *Professional Android 4 Application Development*. Wrox, 2012, ch. Hello, Android, p. 3–4. 2
- [4] Android, “Android’s commitment to kotlin,” Dez 2019, zugriff am 01.12.2022. [Online]. Available: <https://android-developers.googleblog.com/2019/12/androids-commitment-to-kotlin.html> 2
- [5] —, “11 weeks of android: Languages,” zugriff am 01.12.2022. [Online]. Available: <https://android-developers.googleblog.com/2020/07/11-weeks-of-android-languages.html> 3
- [6] —, “Android’s kotlin-first approach : Android developers,” zugriff am 03.12.2022. [Online]. Available: <https://developer.android.com/kotlin/first> 3
- [7] —, “Android studio: An ide built for android,” Mai 2013, zugriff am 04.12.2022. [Online]. Available: <https://android-developers.googleblog.com/2013/05/android-studio-ide-built-for-android.html> 3
- [8] —, “Use android jetpack to accelerate your app development,” Mai 2018, zugriff am 03.12.2022. [Online]. Available: <https://android-developers.googleblog.com/2018/05/use-android-jetpack-to-accelerate-your.html> 3, 4, 8
- [9] —, “Android jetpack dev resources,” zugriff am 04.12.2022. [Online]. Available: <https://developer.android.com/jetpack> 4
- [10] —, “Google i/o 2018: What’s new in android,” Mai 2018, zugriff am 04.12.2022. [Online]. Available: <https://android-developers.googleblog.com/2018/05/google-io-2018-whats-new-in-android.html> 4
- [11] —, “Androidx releases,” zugriff am 04.12.2022. [Online]. Available: <https://developer.android.com/jetpack/androidx/versions> 4
- [12] —, “Androidx overview,” zugriff am 04.12.2022. [Online]. Available: <https://developer.android.com/jetpack/androidx> 4

# Abbildungsverzeichnis

2.1	Überblick über die Komponenten von Android Jetpack [8] . . . . .	4
-----	--	---

# Tabellenverzeichnis



# Appendix

(Hier können Schaltpläne, Programme usw. eingefügt werden.)