

# Introduction

## App Fundamentale

Android Applikationen werden mit der Java Programmiersprache geschrieben.

Sobald ein App installiert ist gibt es für das App:

- Einen einzigartigen Linux User ID: Sodass die Applikation auf keine anderen App-Daten zugreifen kann
- Eine eigene Virtuelle Maschine: Damit die Rechte eingeschränkt bleiben

### App Components

#### Activities

Aktivitäten repräsentieren eine einzelne Anzeige mit einem User-Interface. So kann man zB. eine Activity für die Anzeige der Liste der Emails benutzen und eine andere für die Anzeige des Emails.

#### Services

Services sind Komponenten welche im Hintergrund laufen um länger laufende Operationen durchzuführen.

Beispiel: Musik im Hintergrund hören und eine Textnachricht senden.

#### Content Providers

Content Providers managed das File System, in welchem man Dateien speichern und gebrauchen kann. So managen diese SQLite Datenbanken oder andere Methoden um Daten zu speichern.

#### Broadcast Receivers

Die Broadcast Receivers erhalten Informationen vom System wie zum Beispiel, dass der Akku niedrig ist.

#### Manifest File

Bevor man eine App Komponente startet muss das Android System wissen, dass die Komponente existiert. Dazu gebraucht man das App Manifest. In dieser werden auch andere Dinge festgelegt wie der App-Name, Kompatibilität und die Berechtigungen die es gebraucht.

## Device Compatibility

Android ist darauf ausgelegt auf sehr vielen verschiedenen Geräten zu funktionieren, deshalb ist es wichtig, dass man eine App einschränkt und sie nur für gewisse Geräte, bei denen das auch Sinn macht, zur Verfügung stellt. Diese Einschränkung wird über drei Hauptkriterien gemacht.

### Features

Wenn eine App ein Feature zwingen braucht dann kann man sie nur für Geräte welche über dieses Feature verfügen freigeben. Dazu gibt Android jedem nicht zwingend zu verfügbarem Feature eine ID als Beispiel der Kompass also `FEATURE_SENSOR_COMPASS`.

### Platform version

Bei neuen Versionen von Android werden immer neue APIs hinzugefügt und wenn die App ein gewisses API braucht kann man das auch so einschränken, dies wird mit API Levels gemacht.

### Screen configuration

Beim Bildschirm wird zwischen zwei Einheiten unterschieden, als erstes die Grösse des Bildschirms da gibt es die vier Grössen `small`, `normal`, `large`, und `xlarge`. Als zweites die Pixeldichte auch DPI da gibt es einige standardisierte Dichten `mdpi` (medium), `hdpi` (hdpi), `xhdpi` (extra high), `xxhdpi` (extra-extra high).

### Nicht Technische Gründe

Ausserdem gibt es noch die Möglichkeit eine App Verfügbarkeit zu filtern z.B. nach Standort. Dies wird im Gegensatz zu der technischen Kompatibilität, welche mit APK Files gehandelt wird, über die Google Play developer Console eingestellt.

## Permissions – Rechte

Android benutzt ein Rechtssystem, sodass man ohne diese keine anderen Apps, das OS oder den Benutzer beeinträchtigen kann (Kontakte, Emails, Netzwerkzugriff, etc.). Um auf diese Features zugreifen zu können muss man sie im App-Manifest festlegen:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.android.app.myapplication" >
    <uses-permission android:name="android.permission.RECEIVE_SMS" />
    ...
</manifest>
```

<https://developer.android.com/reference/android/Manifest.permission.html>

### Normale und Gefährliche Permissions

Es gibt Permissions die als Normal und Gefährlich eingestuft sind. Normale Permissions werden bei der Installation der App vom Benutzer gewährt. Gefährliche Permissions werden bei «Run-Time», also wenn die Applikation sie benutzen will, spezifisch vom Benutzer gewährt (ab API 23). «»»»

### Permission-Groups

Es gibt Gruppen von Permissions, zB. `CONTACTS` in der es zB. Das Lesen und Schreiben als spezifische Permission drin sind. Wenn der Benutzer nach diesen Permissions gefragt wird, wird er nur zur Gruppe der Permission gefragt, falls er diese zulässt sind automatisch alle Permissions dieser Gruppe gewährt. → [https://developer.android.com/reference/android/Manifest.permission\\_group.html](https://developer.android.com/reference/android/Manifest.permission_group.html)

### Application Signing

Jedes .apk muss von Autor der Applikation «signed» werden, dh. Unterzeichnet, mittels eines Private Keys. Die Keys können auch Self-Signed sein und nicht von einer CA verifiziert sein.