

# Mobile Computing Android

## Gei



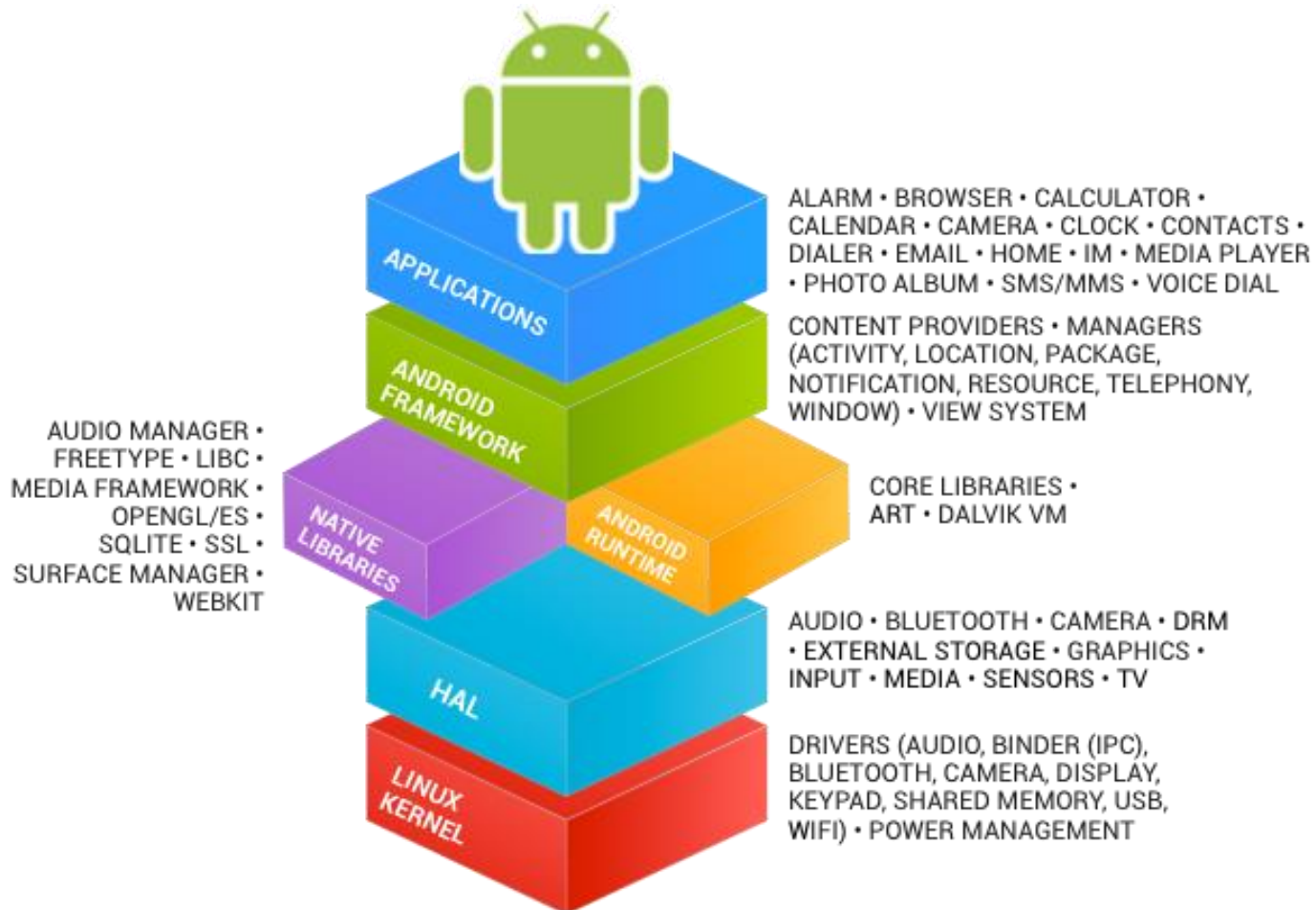
# Googles Android

- Linux Plattform
- Laufzeitumgebung an Java angelehnt
  - Dalvik VM (Virtual Machine)
  - ART (Android Runtime) ab Version 5.0
- Plattform als Open Source
- Verschiedene Hersteller (Samsung, ZTE,...)
- Vertriebskanal: Google Play Store
- Verwendung: Smartphones, Tablets, Googles TV-Plattform, Automobilbereich

# Gründe für die Verwendung

- Plattform ist Open-Source-Software
- verschiedene Hersteller von Geräten
- unterschiedliche Geräteklassen
- relativ rasche Einarbeitung
- kostenlose Tools
- Anzahl der Apps über 2000000
- starke Verbreitung
- echtes Multitasking

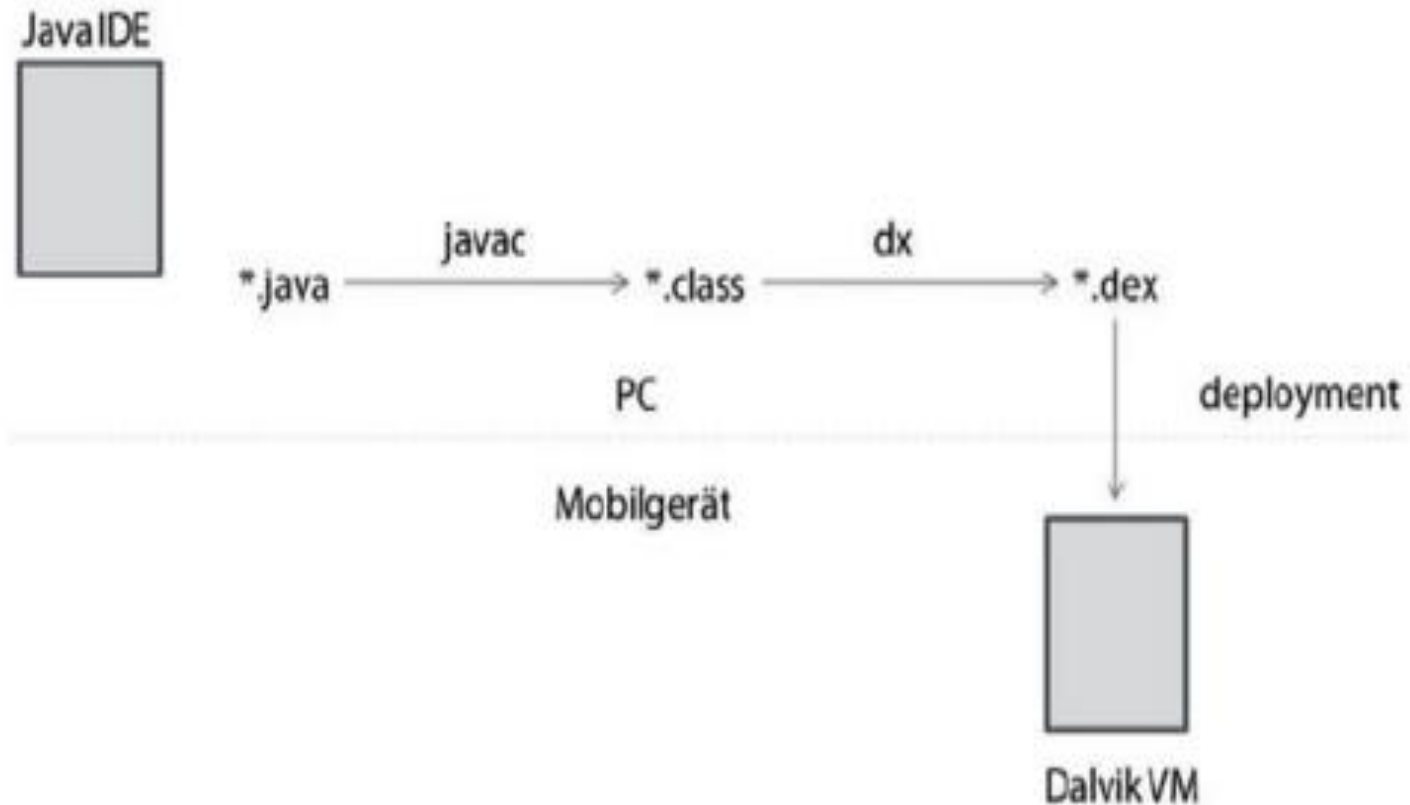
# Android Stack



# Dalvik VM (Virtual Machine)

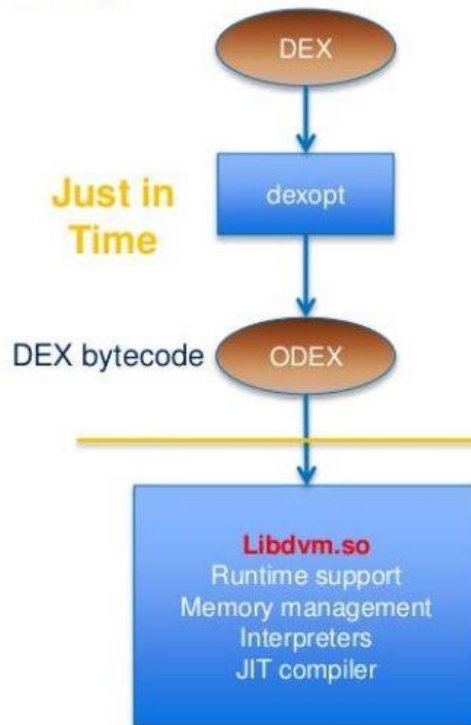
- Basis quelloffene Java VM Apache Harmony
- klassische JVM basiert auf einer Stapel-  
die DVM auf einer Registermaschine
- mehrere Instanzen können auch auf  
kleinen Systemen parallel laufen
- Lizenztrick: Die JVM und der original Java-  
Bytecode sind lizenziertlich geschützt,  
die Programmiersprache Java nicht.
- Ab 5.0 Android Runtime ART

# Vom Java-Code zur Dalvik VM

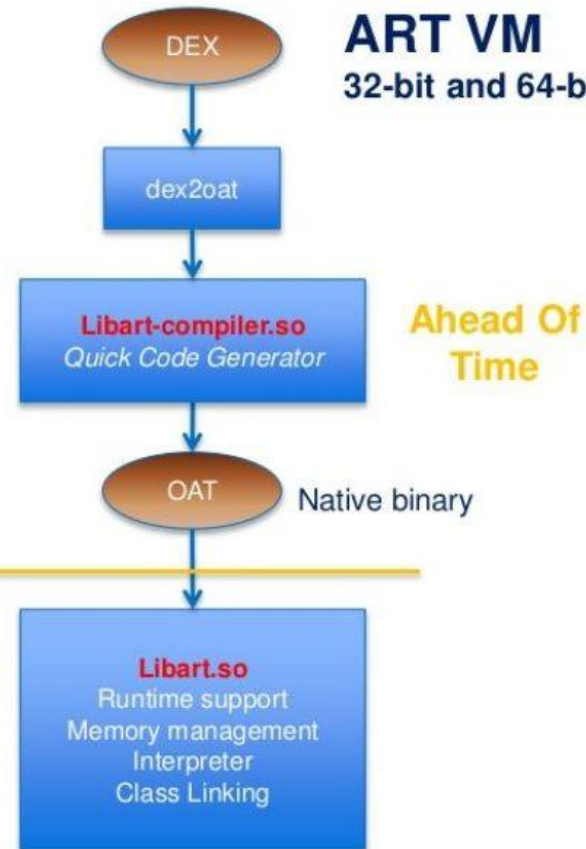


# Dalvik VM vs. ART VM

## Dalvik VM 32-bit only



## ART VM 32-bit and 64-bit



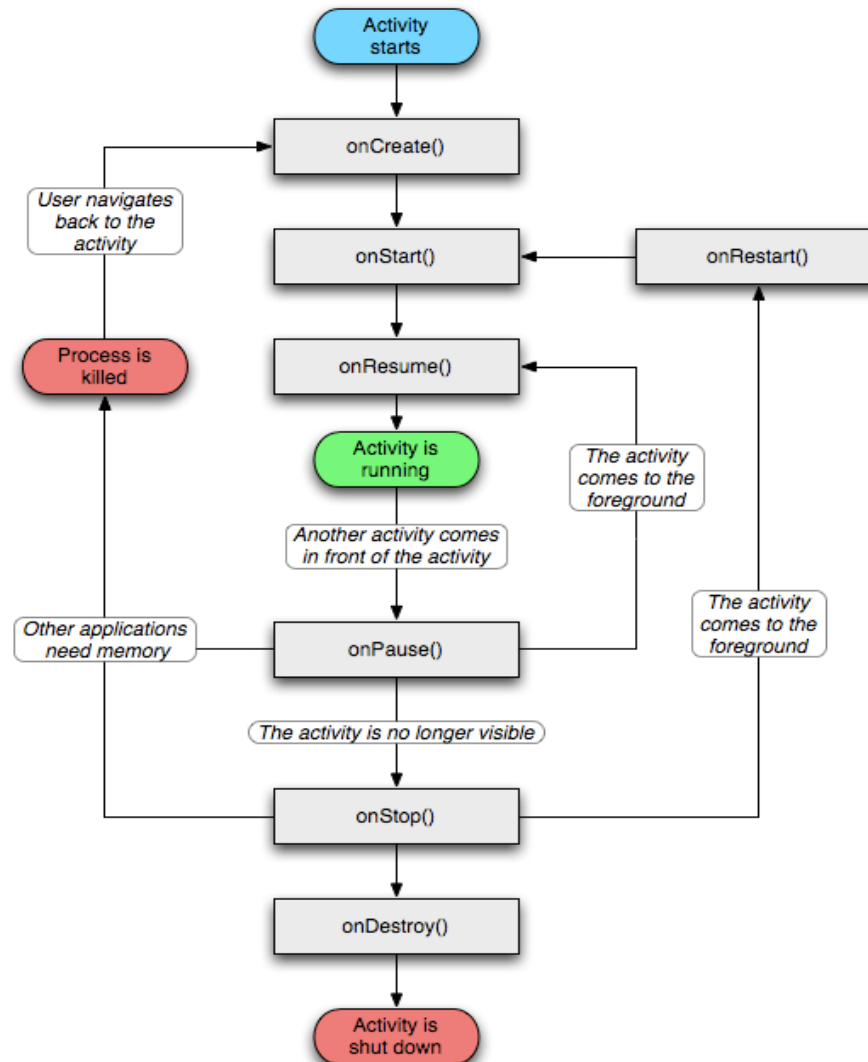
Runtime

# Android Komponenten

- **Activity**: Activities dienen zur Darstellung und Verwaltung von Oberflächen.
- **Service**: Ein Service erledigt Hintergrundprozesse.
- **Content Provider**: Ein Content Provider verwaltet Daten und abstrahiert die darunterliegende Persistenzschicht.
- **Broadcast Receiver**: Broadcast Receiver empfängt Systemnachrichten und ermöglicht darauf zu reagieren.



# Lebenszyklus einer Activity



- Android führt alle Anwendungen in einer Sandbox aus.
- Eine Sandbox ist eine eingeschränkte Laufzeitumgebung.
- Eine Anwendung besitzt einen eigenen Prozess, Betriebssystem-User, DVM/ART, Bereich im Hauptspeicher, Bereich im Dateisystem

# Signieren einer Anwendung

- Der Anwendung wird mit Hilfe eines **Zertifikats** eine **digitale Unterschrift** hinzugefügt.
- Das Android SDK erlaubt das Signieren mit unbeglaubigten Zertifikaten.
- Durch das Zertifikat wird daher nur sichergestellt, dass verschiedenen Anwendungen vom selben Hersteller stammen.
- Standardzertifikat reicht für Entwicklung

# Berechtigungen

- Durch die **Berechtigung** wird der Anwendung der **Zugriff auf Systemfunktionen** und Ressourcen **außerhalb der Sandbox** ermöglicht.
- Die **Berechtigungen** werden im **Android-Manifest eingetragen**. (AndroidManifest.xml)
- **sharedUserId** um **mehrere Anwendungen** in **derselben Sandbox** auszuführen

# Elemente zur Oberflächengestaltung

- **View:** Eingabefelder, Buttons und Listen werden als Views bezeichnet.
- **Layout:** Views werden mittels Layout am Bildschirm angeordnet. (z.B. LinearLayout, TableLayout)
- **Activity:** Die Activity dient zur Darstellung und Verwaltung von Oberflächen

- 1. Woche: Android-Studio, Kotlin for Android
- 2. Woche: Eingangswiederholung  
Jetpack Compose, Projektideen
- 3. Woche: Jetpack Compose Next Level, Projekt
- 4. Woche: Projekt  
Abschlusspräsentation

# Voraussetzungen

- Grundbegriffe der objektorientierten Programmierung (Klasse, Objekt, ...)
- Android Studio vorinstalliert  
<https://developer.android.com/studio>
- Android Virtual Device vorinstalliert  
<https://developer.android.com/studio/run/managing-avds#createavd>
- Kopfhörer für Videos

# more information

<https://developer.android.com/>

<https://developer.android.com/jetpack/compose>

<https://developer.android.com/jetpack/compose/tutorial>

<https://kotlinlang.org/docs/android-overview.html>