

Android: Time Tracker

- ▼ Florian Diederichs
- ▼ Version 2011-10-20 (Production)

Android

- ▼ Warum Android?
 - ▼ Bin Java Coder
 - ▼ Hat mich schon immer interessiert
- ▼ Ist nicht prüfungsrelevant
- ▼ Zurücklehnen und entspannt anhören
- ▼ Begegnet einem öfters

Gliederung

- ▼ Android
- ▼ Time Tracker
- ▼ Entwicklungsumgebung
- ▼ Aufgaben

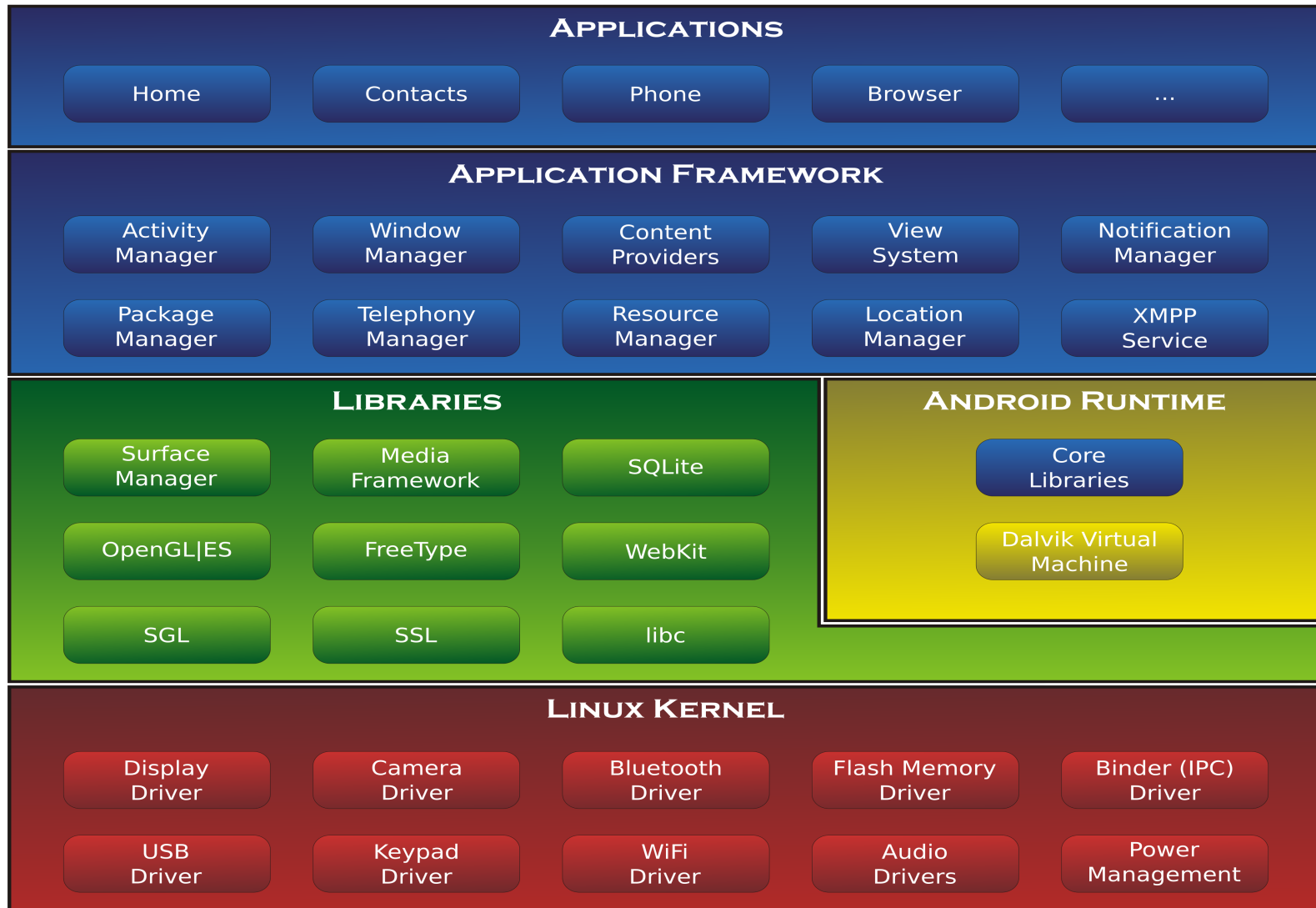
1. Android

- ▼ Einleitung
- ▼ Hello World
- ▼ Architektur
- ▼ Komponentenmodell
- ▼ Activities
- ▼ Services
- ▼ Content Providers

Android Einleitung

- ▼ Betriebssystem und Software-Plattform
- ▼ Entwickelt von Google
- ▼ Ursprünglich Plattform für mobile Endgeräte, vor allem Smartphones
- ▼ Anfangs nur für ARM-Prozessoren, mittlerweile auch andere wie X86 und MIPS
- ▼ Einsatz heute auch auf
 - ▼ Tablets
 - ▼ Netbooks
 - ▼ Auto-Infotainment
 - ▼ Spielekonsolen
 - ▼ Set-Top-Boxen

Android Architektur



Quelle: <http://de.wikipedia.org/wiki/Datei:Android-System-Architecture.svg>

Dalvik Virtual Machine

- ▼ Ersetzt die JVM
- ▼ Hat einen eigenen Bytecode → Cross-Compiler
- ▼ Optimiert für mobile Prozessoren
- ▼ Ressourcenschonend → Jede App bekommt eine eigene Instanz

Android Anwendungen

▼ Terminologie

- ▼ KGSE Komponenten != Android Komponenten
- ▼ KGSE Komponenten == Android Anwendungen
- ▼ Android Komponenten == Android Bausteine

▼ Android Bausteine

- ▼ Activities
- ▼ Services
- ▼ Content Providers
- ▼ Broadcast Receivers

▼ Weitere wichtige Bestandteile

- ▼ Datenbankschnittstelle
- ▼ Intents
- ▼ Android-Manifest
- ▼ Layouts
- ▼ Strings
- ▼ Grafiken

Activities

- ▼ Sichtbare Teile einer Android Anwendung
- ▼ Eine Bildschirmseite entspricht einer Activity
- ▼ Erstellen einer Activity durch Erstellen einer Klasse welche von der Klasse *Activity* abgeleitet ist
- ▼ Implementieren bestimmter Event-Handler
- ▼ Einführung eigener Event-Handler für Reaktion auf Benutzereingaben
- ▼ Layout-Konfiguration in XML-Dateien

HelloWorld App



Aufruf mehrerer Activities

- ▼ Die meisten Apps bestehen aus mehreren Activities
- ▼ Activities können andere Activities starten
 - ▼ Derselben Anwendung
 - ▼ Anderer Anwendungen

MultiActivity App



Intents

- ▼ Mechanismus zum Austausch zwischen Android Bausteinen
- ▼ Drei Hauptanwendungsfälle
 - ▼ Starten einer Activity
 - ▼ Starten eines Service
 - ▼ Absenden eines systemweiten Broadcasts
- ▼ Intents sind der „Klebstoff“ zwischen den Bausteinen einer oder mehrerer Android Anwendungen
- ▼ Intents können unterschiedliche Zusatzinformationen erhalten und an den Zielbaustein übergeben
- ▼ Explizite und Implizite Intents

Intents App



Activity Start durch Explizite Intents

- ▼ Zum Aufruf einer Activity innerhalb derselben Anwendung
- ▼ Direkte Angabe der Klasse

```
Intent intent = new Intent(this, SecondActivity.class);
```

Activity Start durch Implizite Intents

- ▼ Aufgerufene Activity kann sich innerhalb oder ausserhalb der eigenen Anwendung befinden
- ▼ Angabe eines *ACTION* Parameters
- ▼ Für den individuellen *ACTION* Parameter können mehrere Activities in Frage kommen
- ▼ Bereitstellung passender Activities durch Android anhand der Activity-Filter in Android-Manifest Dateien
- ▼ Auswahl der gewünschten Activity durch den User

```
Intent intent = new Intent(ACTION);
```

```
<activity android:name=".SecondActivity"/>  
  <intent-filter>  
    <action android:name="android.activities.action.VIEW"/>  
    <category android:name="android.intent.category.DEFAULT"/>  
  </intent-filter>  
</activity>
```


Implizite Intents App



Activities und Prozesse

- ▼ Jede App läuft in einem eigenen Prozess (mit eigener DVM)
- ▼ Activities derselben App werden in der Regel im selben Prozess gestartet
- ▼ Geöffnete Activities bilden einen Keller
- ▼ Navigation über den “Zurück”-Button
- ▼ Bei Ressourcenknappheit schießt Android Prozesse ab
- ▼ Bei Rückkehr zu abgeschossenen Activities sorgt Android für die Wiederherstellung

Threads und Hintergrundoperationen

- ▼ Jeder Prozess hat einen MainThread
- ▼ Methoden aller Androidbausteine laufen in der Regel im MainThread
- ▼ GUI Blockade durch längere Methodenaufrufe
- ▼ ANR nach ca. 5 – 10 Sekunden
- ▼ Auslagerung längerer Operationen in andere Threads
- ▼ Verschiedene Mechanismen für Threadauslagerung
- ▼ Mechanismen für GUI-Zugriff aus anderen Threads
 - ▼ Selbst programmierte Callback-Mechanismen
 - ▼ Hilfsklassen und -methoden
 - ▼ Async-Task
- ▼ Aufräumen von Threads ist Verantwortung des Entwicklers

Threads und Hintergrundoperationen

- ▼ Jeder Prozess hat einen MainThread
- ▼ Methoden aller Androidbausteine laufen in der Regel im MainThread
- ▼ GUI Blockade durch längere Methodenaufrufe
- ▼ ANR nach ca. 5 – 10 Sekunden
- ▼ Auslagerung längerer Operationen in andere Threads
- ▼ Verschiedene Mechanismen für Threadauslagerung
- ▼ Mechanismen für GUI-Zugriff aus anderen Threads
 - ▼ Selbst programmierte Callback-Mechanismen
 - ▼ Hilfsklassen und -methoden
 - ▼ Async-Task
- ▼ Aufräumen von Threads ist Verantwortung des Entwicklers

Services

- ▼ Ableitung aus Klasse *Service*
- ▼ Keine Benutzeroberfläche
- ▼ Local/Remote Services
- ▼ Start über explizite oder implizite Intents
- ▼ Auslagerung von Quellcode in eine eigene Komponente
- ▼ Service-Methoden laufen per Standard im MainThread
- ▼ Gut geeignet für Hintergrundoperationen
- ▼ Andere Prozess-Priorität als einfache Thread-Auslagerung
- ▼ Gebundene/Ungebundene Services

Broadcast Receivers

- ▼ Reagieren auf Broadcast Intents
- ▼ Broadcast Intents sind Systemnachrichten:
 - ▼ Hochfahren (ACTION_BOOT_COMPLETED)
 - ▼ Niedriger Batteriestand (ACTION_BATTERY_LOW)
 - ▼ viele weitere
- ▼ Können über die Laufzeit der App hinweg existieren
- ▼ Konfiguration über Intent-Filter

Content Provider

- ▼ Austausch strukturierter Daten über Anwendungs- und Prozessgrenzen hinweg
- ▼ Zugriff auf Dateninhalt, nicht die Daten selbst → Abstraktion über internes Datenmodell
- ▼ Hauptanwendungen: Datenbank und Dateizugriffe
- ▼ Berechtigungssystem für Lese- und Schreibzugriffe (normale Android-Berechtigungen)
- ▼ Konfiguration über Android-Manifest
- ▼ Klassen welche auf einen CP zugreifen möchten nutzen ein Content Resolver (CR) Objekt aus ihrem eigenen Adressraum
- ▼ CR stellt die API-Methoden des zugehörigen CP zur Verfügung
- ▼ Interprozesskommunikation zwischen CR und CP

Content Provider

- ▼ Beispiele für CP:
 - ▼ Calendar Provider
 - ▼ Contacts Provider
 - ▼ Media Store Provider
- ▼ Verschiedene Operationen möglich:
 - ▼ insert / delete / update / query
 - ▼ openFile
 - ▼ ...
- ▼ Zugriff über Content-URIs

```
$scheme://$authority/$dataDescriptor[/ $id]  
content://com.example.app.provider/contacts  
content://com.example.app.provider/contacts/1
```


Android und Datenbanken

- ▼ Warum Datenbanken für Android?
 - ▼ Persistenz
 - ▼ Zwischenspeicherung
 - ▼ Kombination mit Web-Synchronisation
- ▼ SQLite im Lieferumfang enthalten

SQLite

- ▼ Schlankes DBMS, optimiert für mobile Plattformen
- ▼ Kein Server, standalone
- ▼ Viele Features trotz geringer Größe
- ▼ Datenbank in einer einzelnen Datei gespeichert
- ▼ Nur für eine einzelne App erreichbar
- ▼ Konvention: Datenbank-Manager-Klasse (DMK), erweitert SQLiteOpenHelper
- ▼ DMK dient der Schemaverwaltung und Versionierung
- ▼ Direkter Zugriff auf die Datenbank über die Klasse SQLiteDatabase
- ▼ Software-Schicht abstrahiert den Zugriff:
 - ▼ Data Access Objects (DAOs)
 - ▼ Content Providers

2. Time Tracker

- ▼ Einleitung
- ▼ Ziele und Anforderungen
- ▼ Demo
- ▼ Activities
- ▼ Geolocation
- ▼ AsyncRequest
- ▼ Datenbank-Anbindung

Anforderungen

- ▼ Time Tracking
- ▼ Benutzereingaben „in Echtzeit“
- ▼ Was/Wann/Wo
- ▼ Einfaches Starten und Stoppen
- ▼ Nur ein Task gleichzeitig
- ▼ Nachbearbeitung und Verwaltung von Tasks
- ▼ Statistiken
- ▼ Verwaltung von Kategorien

Praesentation

Datenbank: Definition und Verwaltung

- ▼ SQLite
- ▼ DatabaseHelper
- ▼ TrackTable und CategoryTable
- ▼ Versionierung für Schemaänderungen

Datenbank: Definition und Verwaltung

- ▼ SQLite
- ▼ DatabaseHelper
- ▼ TrackTable und CategoryTable
- ▼ Versionierung für Schemaänderungen

Category	Cursor
Cursor	
DAO	Content Provider
SQLite Database	
<i>SQLite Datenbank</i>	

Quelle: Eigene Darstellung

Main Activity

Edit Track Activity

Geodaten und Adressen Lookup

- ▼ Koordinaten
 - ▼ Bereitgestellt durch Android Framework
 - ▼ Listener
- ▼ Adressen-Lookup
 - ▼ Geocoding-API
 - ▼ AsyncTask
 - ▼ Rechtliche Probleme
 - ▼ Alternative: Geocode-Klasse

Listenansichten

Thank you ...

- ▼ ... for using this template!
- ▼ ... for supporting LibreOffice!



All text and image content in this document is licensed under the [Creative Commons Attribution-Share Alike 3.0 License](#) (unless otherwise specified). "LibreOffice" and "The Document Foundation" are registered trademarks. Their respective logos and icons are subject to international copyright laws. The use of these therefore is subject to the [trademark policy](#).