

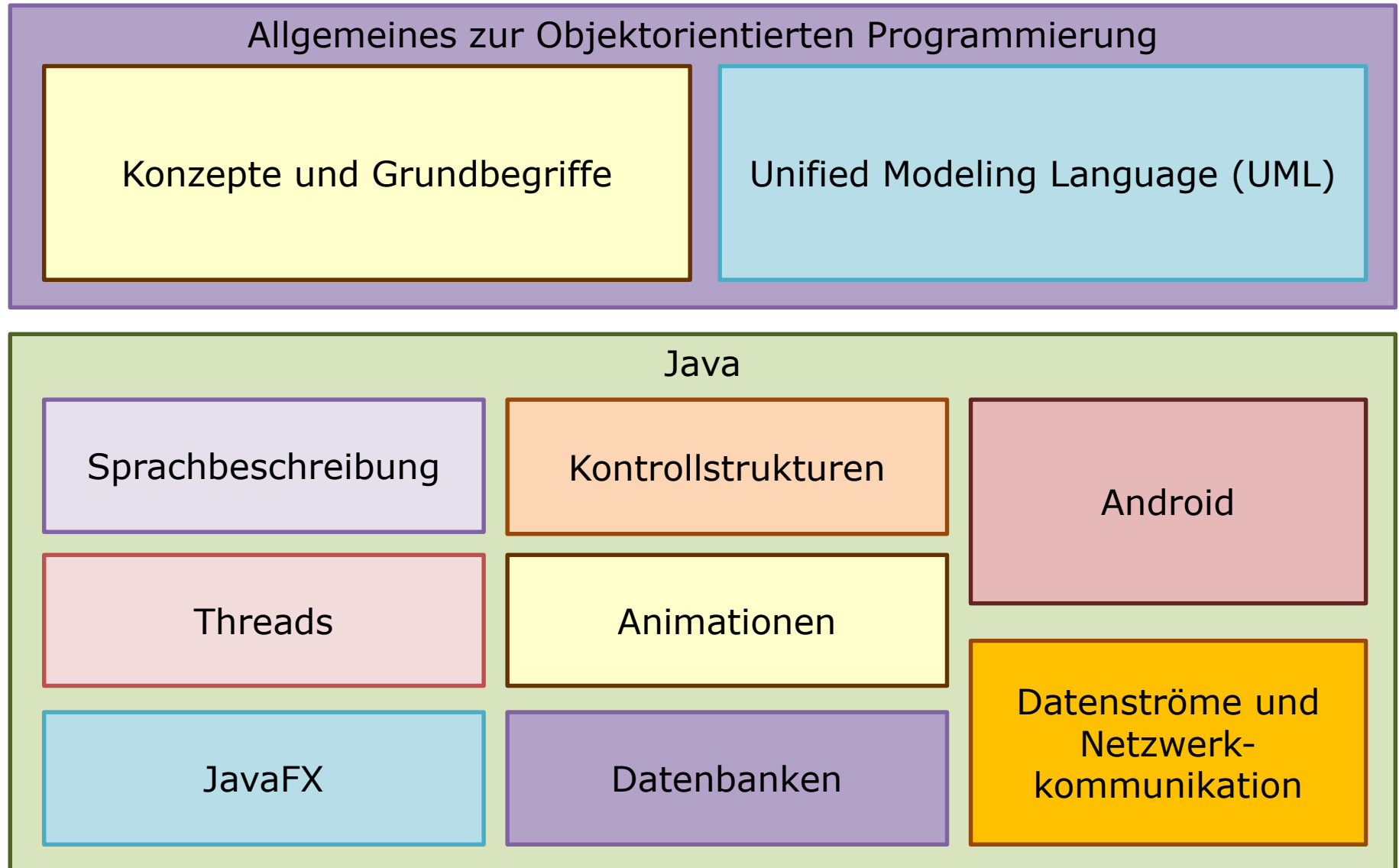
Objektorientierte Programmierung

Android

Prof. Dr.-Ing. Tenshi Hara
tenshi.hara@ba-sachsen.de

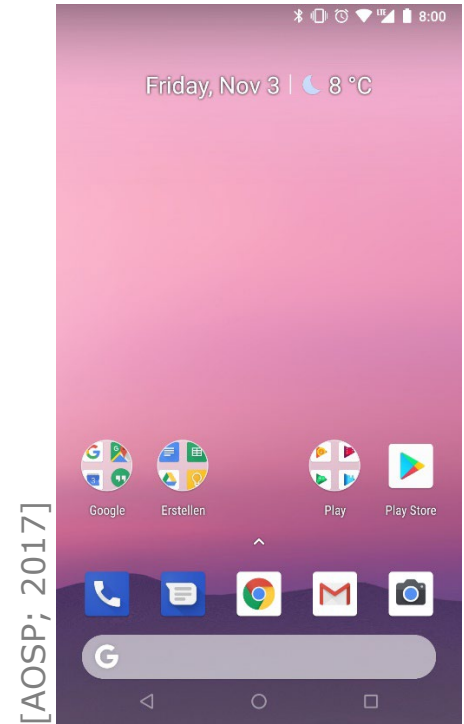


AUFBAU DER LEHRVERANSTALTUNG



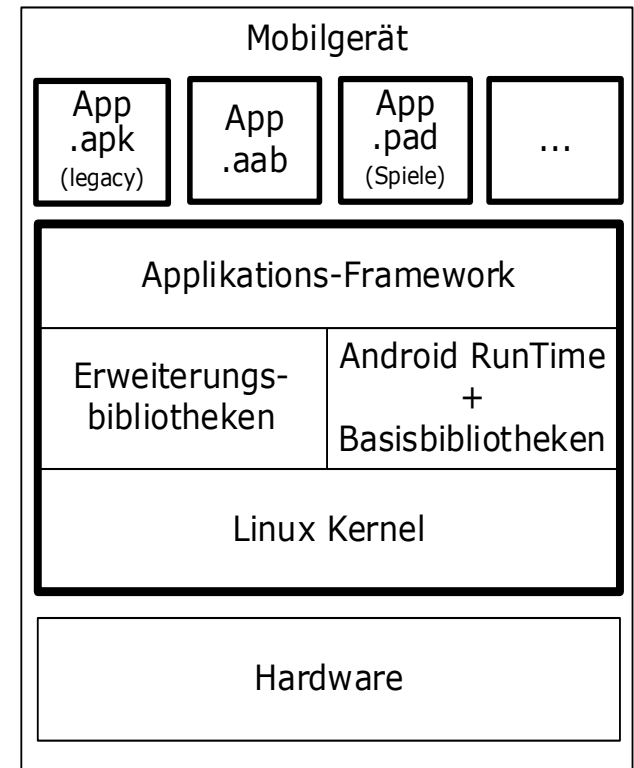
ALLGEMEINES

- Entwickelt von der Open Handset Alliance, getrieben von Google
- Zielplattform: Smartphones und Tablets
- Unterstützung heterogener Hardware
- zugrunde liegendes Betriebssystem: Linux 3.x
- Open Source (Apache 2.0, GPL 2.0)
- oft mit vorinstallierter proprietärer Google Software
- Programmierung von Apps für Android in Java mittels Android Studio
- Android 1.0 (Beta) im November 2007; aktuell 12L bzw. 13 DP2 (T)



ARCHITEKTUR

- Framework mit wiederverwendbaren und austauschbaren Komponenten
- Bibliotheken
 - für viele gängige Medienformate
 - SQLite als relationale Datenbank
 - Schnittstellen für Positionierung
 - integrierte Browser-Engine
 - Grafikoptimierung
- Android RunTime
- Linux Kernel 3.x

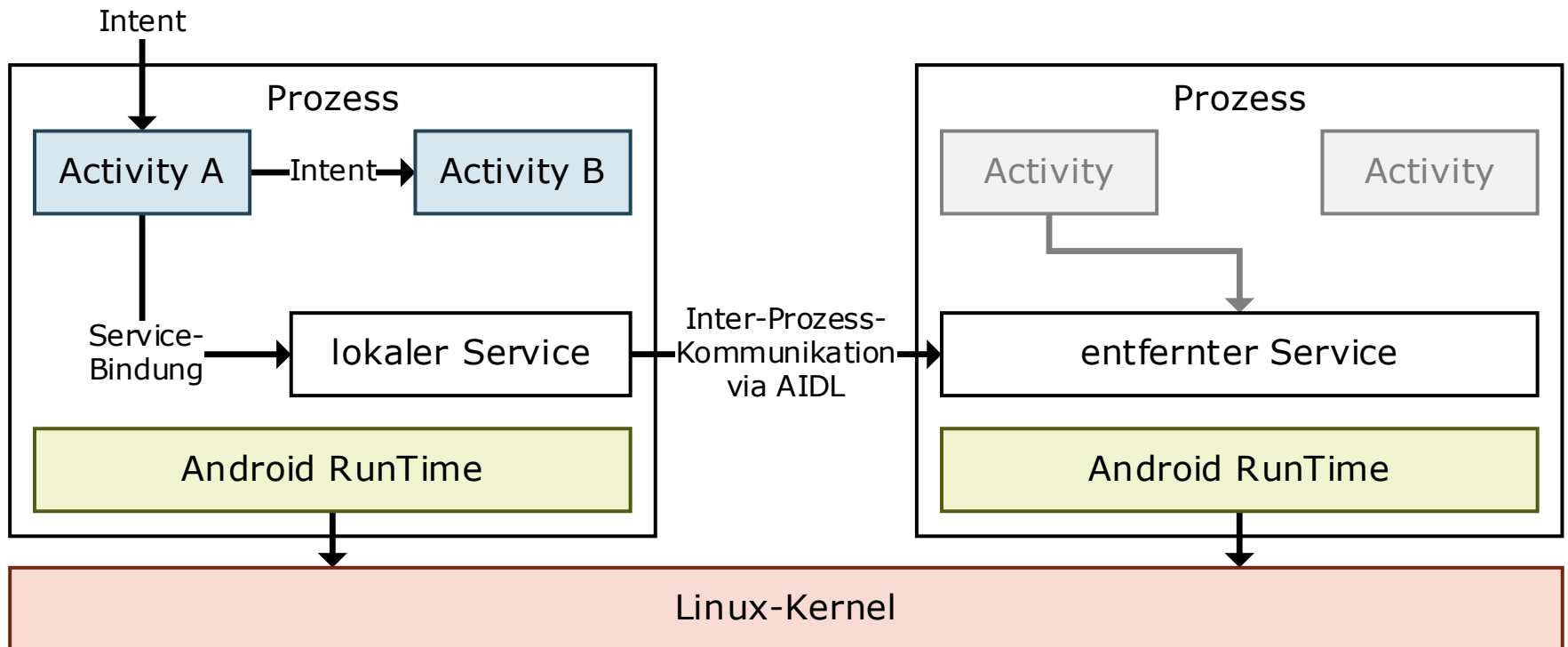


ANDROID RUNTIME

- Standard Laufzeitumgebung seit Android 5.0 (Lollipop)
- Ahead-of-Time-Kompilierung (AOT)
 - beim Installieren entstehender Overhead (Speicher und Zeit)
 - bessere Effizienz und Leistung zur Laufzeit
- verbesserte Garbage Collection und Debugging

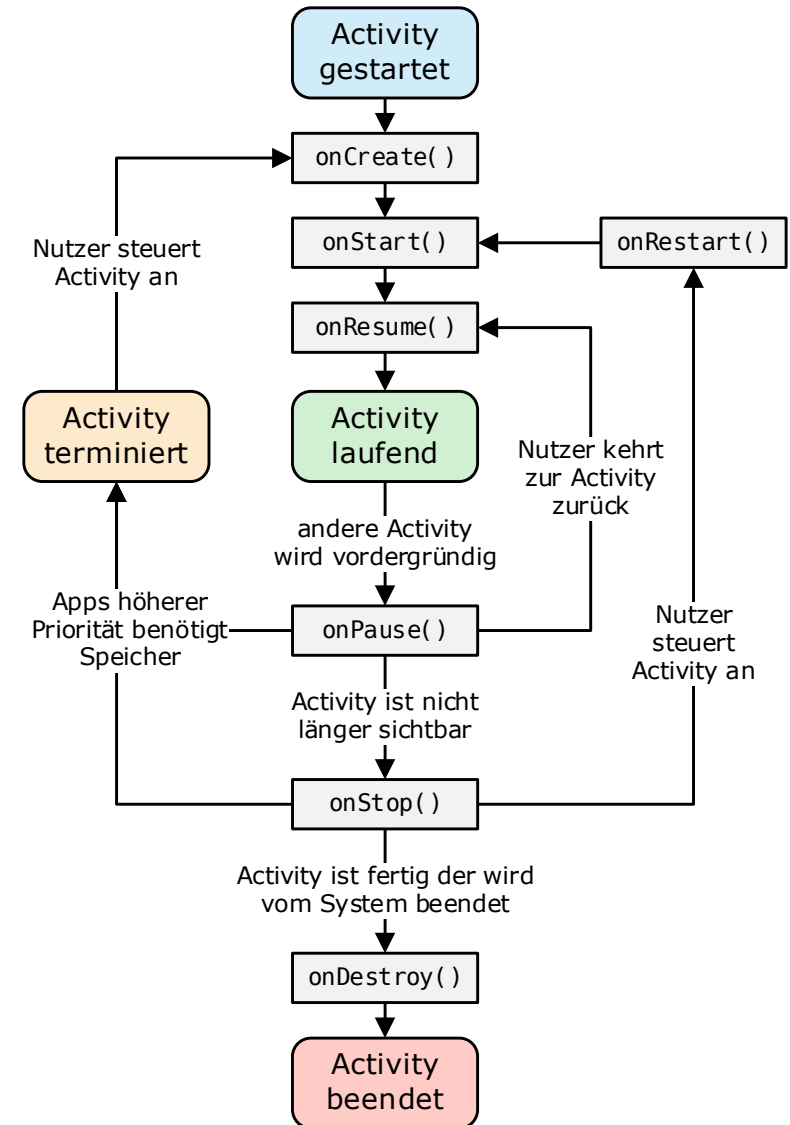
AUFBAU EINER ANDROID-APP

- drei Hauptbestandteile: **Activities**, **Services**, **Intents**
- Android-Apps werden in getrennten Prozessen ausgeführt
 - Inter-Prozess-Kommunikation via AIDL-Schnittstellen
 - verwendete Komponenten müssen in Manifest deklariert werden



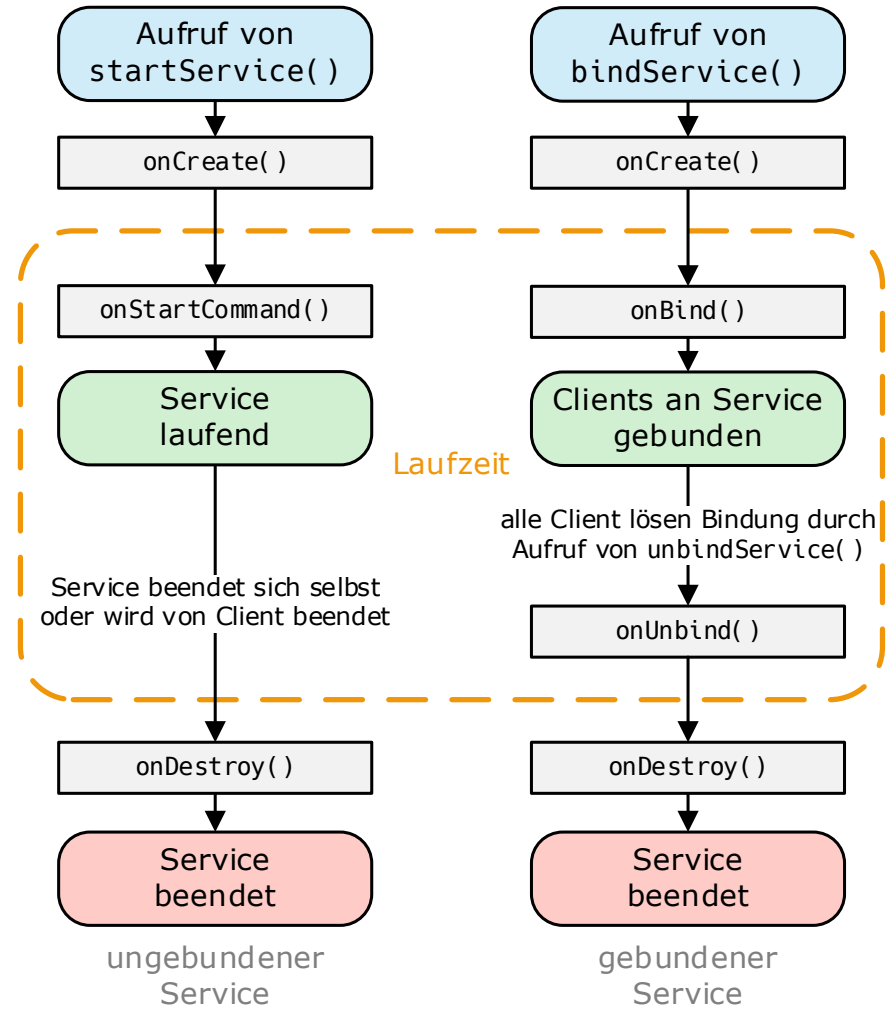
ACTIVITY

- repräsentiert eine einzelne Anzeige (**Screen**) einer App
- Umsetzung der Activity-Klasse (Erweiterung um eigene Methoden)
- enthält Nutzerschnittstellenelement (**Views**), die auf Ereignisse reagieren
- Android hält Activity-Stapel vor (bspw. für „zurück“)
- Methoden einer Activity spiegeln den Lebenszyklus wieder
- vier grundlegende Zustände
 - laufend (**Running**)
 - pausiert (**Paused**)
 - angehalten (**Stopped**)
 - beendet (**Destroyed**)



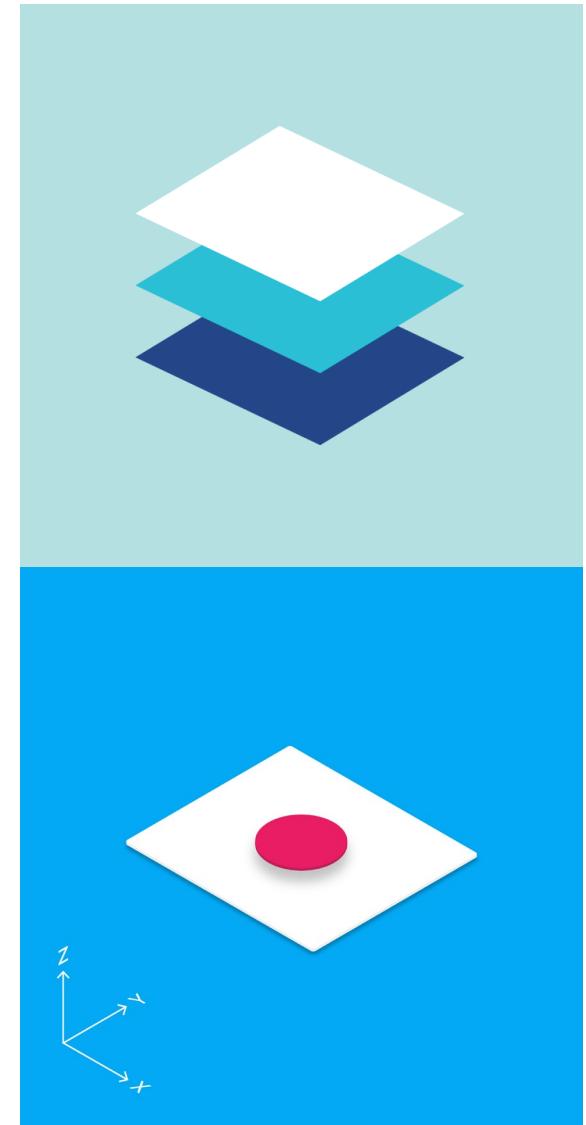
SERVICE

- Ausführung von Hintergrundaktivitäten (startService)
- Offenlegung von Funktionalität ggü. anderen Apps (bindService)
- läuft im Prozess erzeugender App
- **kein Thread**
- lokal und entfernt nutzbar
- wenn verbunden, wird via offenliegender Schnittstelle kommuniziert
 - lokal: einfacher Java-Ruf
 - entfernt: Android Interface Definition Language (**AIDL**)



MATERIAL DESIGN: NATIVE OBERFLÄCHE VON ANDROID 5 BIS 11

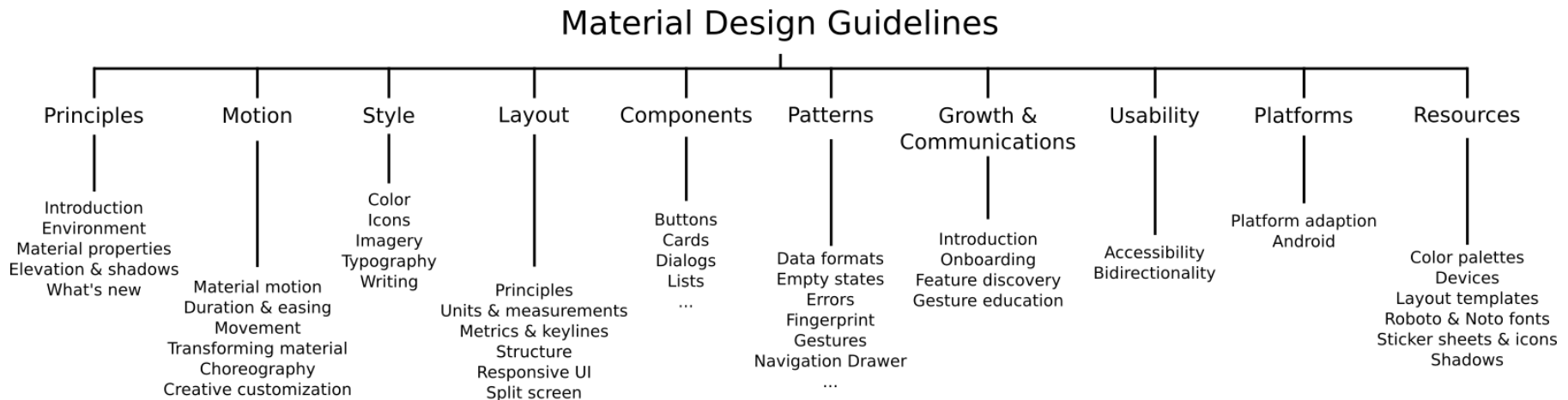
- seit 2014 Vorgabe von Google
→ <https://material.io>
- wurde mit der Einführung von Android 12 („Red Velvet Cake“) durch *Material You* abgelöst
- Material als Metapher (Stift und Papier)
- GUI-Elemente erben Papiercharakteristiken
- flaches Design mit Ebenen (z-Achse)
- z-Position wird durch Erhöhungen und Schatten dargestellt
- Erhöhung wird durch Nutzerinteraktion geändert (bspw. Schaltfläche betätigen)
- umfangreiche Nutzung von Bildern und Animationen



MATERIAL DESIGN: NATIVE OBERFLÄCHE VON ANDROID 5 BIS 11

Material Design Styleguides

- unterteilt in Hauptkategorien
 - Principles – spezifizieren Grundlagen des Material Designs
 - Components – spezifizieren GUI-Elemente (Dialoge, Menüs, ...)
- enthält auch Beispiele und Warnungen (Dos und Don'ts)



MATERIAL DESIGN: NATIVE OBERFLÄCHE VON ANDROID 5 BIS 11

 MATERIAL DESIGN

Material Design

Motion

Style

Layout

Components

Bottom navigation

Bottom sheets

Buttons

Buttons: Floating Action Butt...

Cards

Chips

Data tables

Dialogs

Dividers

Expansion panels

Grid lists

Lists

Lists: Controls

Menus

Pickers

Progress & activity

Components – Cards

Cards

A card is a sheet of material that serves as an entry point to more detailed information.

Cards may contain a photo, text, and a link about a single subject. They may display content containing elements of varying size, such as photos with captions of variable length.

A card collection is a layout of cards on the same plane.

Usage

Cards display content composed of different elements whose size or supported actions vary.

Supported gestures

Swipe

Pick-up-and-move

Related components

Grid lists

For developers

Android cards

Polymer cards

Contents

Usage

Content

Behavior

Actions

Content blocks



MATERIAL YOU: NATIVE OBERFLÄCHE AB ANDROID 12

- Anpassungen in den Farbpaletten (Pastelltöne)
 - dynamische Farbpalette
 - abgerundete Ecken
 - vergrößerte Tap Targets
 - neue (dynamische) Elementformen und -abstände
- <https://material.io/blog/announcing-material-you>

NETZWERKKOMMUNIKATION

- `Android.net.ConnectivityManager`
 - überwacht Verbindungszustand
 - sendet Broadcast-Intent sobald sich Zustand ändert
 - stellt Methoden zum Zugriff auf Netzwerkzustand bereit
 - `getActiveNetworkInfo( )`
 - `getAllNetworkInfo( )`
 - `getNetworkInfo(int networkType)`
- `Java.net.*`-API können genutzt werden
 - werden als Teil der Android-Plattform vorgehalten
 - `HttpClient` als Basis für HTTP-Verbindungen
- alternative Bibliotheken auch verwendbar (bspw. Google Volley)

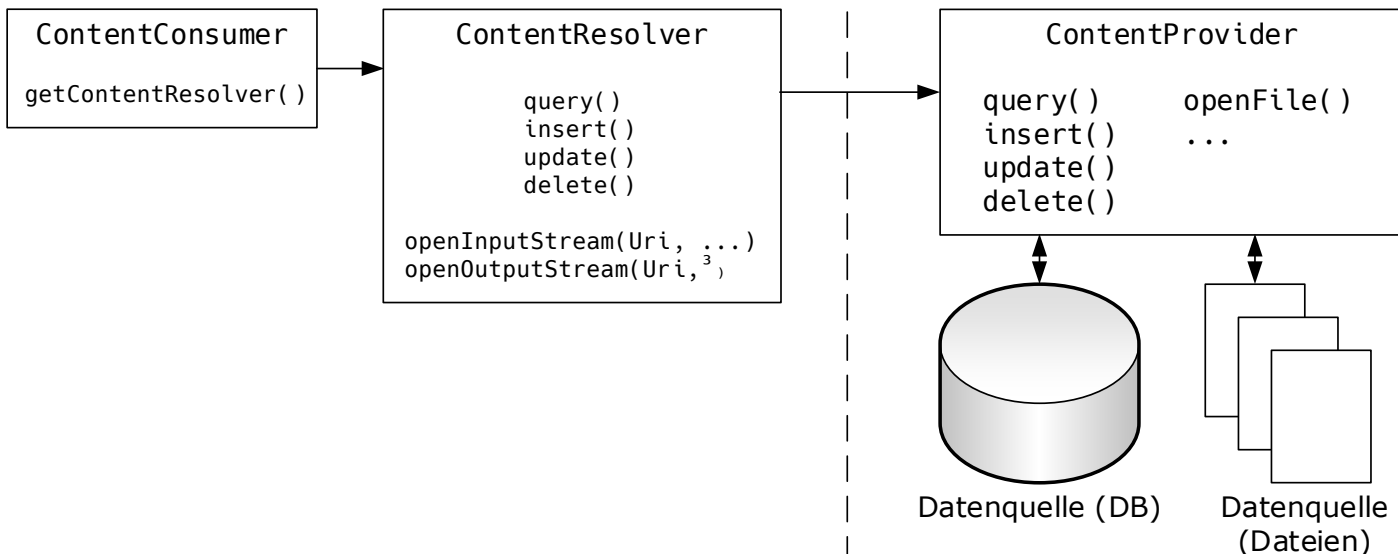
PERSISTENTER SPEICHER

- SQLite
 - lokale Datenbank mit SQL- und Transaktionsunterstützung (siehe Kapitel 7) in einzelner Datei
 - Ausgangsklasse: `android.database.sqlite.SQLiteDatabase`
 - Abfragen mittels `SQLiteDatabase.query( )`
- gemeinsame Voreinstellungen
 - Ausgangsklasse: `android.content.SharedPreferences`
 - speichert primitive Daten in Schlüssel/Wert-Paaren
- interner Speicher für private Daten in Dateien (getrennt nach App-Id)
- (virtueller) externer Speicher für gemeinsame Daten in gemeinschaftlich zugreifbaren Dateien

PERSISTENTER SPEICHER – CONTENTPROVIDER

Teilen von Daten zwischen Applikationen

- Abstraktionsebene oberhalb der DB oder von Dateien (Interface-Klasse `android.content.ContentProvider`)
- Inhalt wie auf Web-Server organisiert – URIs für Zugriff
- Standardmethoden für lesenden/schreibenden Datenzugriff
- `ContentProvider`-Implementierung für gängige Datentypen
- Zugriff via `ContentResolver`



Mehr zu Android und zur App-Programmierung
in den Lehrveranstaltungen

Programmierung mobiler Anwendungen

(Wahlpflicht 3IT-PMA-40)

Web- und App-Programmierung

(Wahlpflicht 3MI-WEBPR-50)

REFERENZEN

<https://developer.android.com>

<https://developer.android.com/guide/components/aidl.html>

<https://material.io>