

3.– 6. September 2012  
in Nürnberg



# Herbstcampus

Wissenstransfer  
par excellence

## Roboter-Baukasten

Einführung in Android

Manfred Borzechowski

LambdaLogic Informationssysteme GmbH

# Wir gratulieren Ihnen, daß Sie sich für einen Android-Baukasten entschieden haben

- Volle Java-Sprachunterstützung
  - Bis Compiler-Level 6
- Reichhaltiges API
- Breite Geräteauswahl
- Vielfältige Möglichkeiten
  - Organizer, Social, Player, Recorder, Navigation, Spiele, Barcode Scanner, Stimmgerät, Taschenlampe, u.v.a.m.
- Ideal für die Realisierung eigener Ideen

# Verhaltensmaßregeln

- Vermeiden Sie unnötige Instanziierungen
  - Heap für Ihre Prozesse begrenzt
  - Garbage Collection langsam
- Vermeiden Sie häufige Schreibvorgänge
  - Unperformant wegen Flash-basiertem Speicher
- Vermeiden Sie Blockierung des UI-Threads
  - „Die Anwendung reagiert nicht“
- Vermeiden Sie häufige Netzwerkzugriffe über den Mobilfunkbetreiber
  - Verbraucht hohe Akku-Leistung

# Der Baukasten

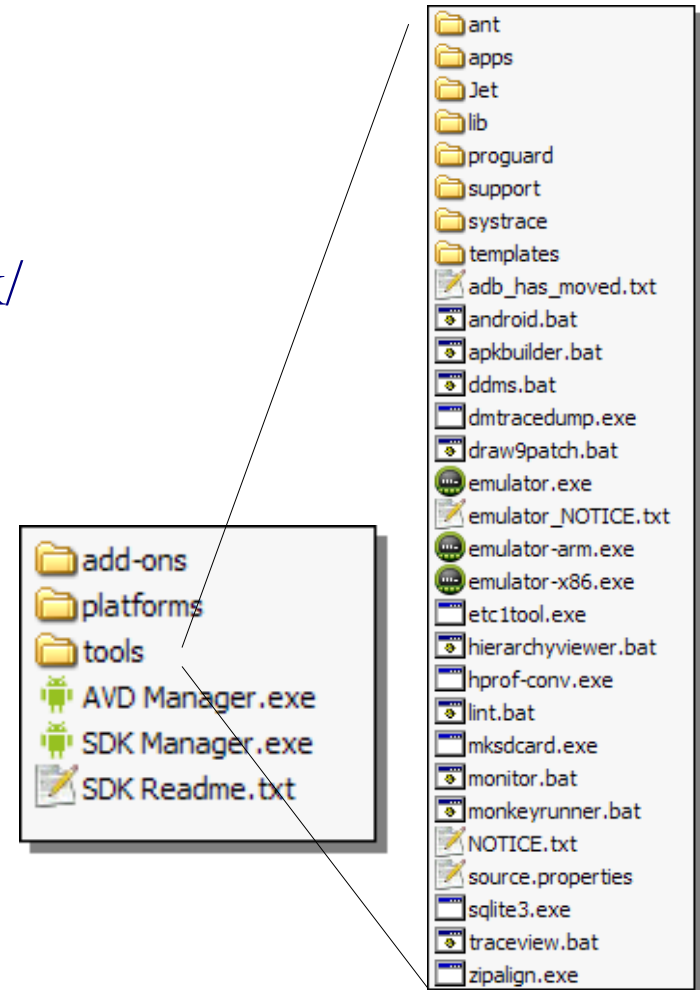


# Die Bodenplatte: Ein Java JDK

- Haushaltsübliche Version
  - (nicht mitgeliefert)
- Nur für IDE und SDK
- Android bringt sein eigenes „Java“ mit
  - Eigene Laufzeitklassen und Dalvik VM

# Das Android SDK

- Bezugsquelle  
<http://developer.android.com/sdk/>
  - Bei Bestellung angeben ob für Windows, Mac OS X, Linux
- Bestandteile kontrollieren
  - Tools
  - SDK Manager
  - AVD Manager



SDK Manager

SDK

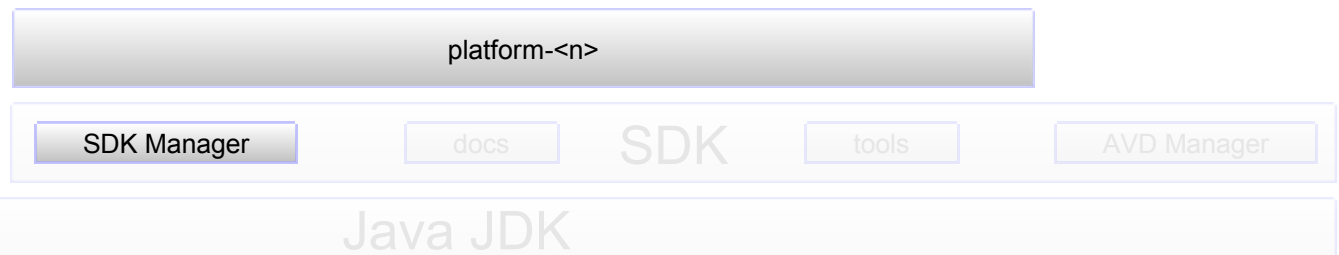
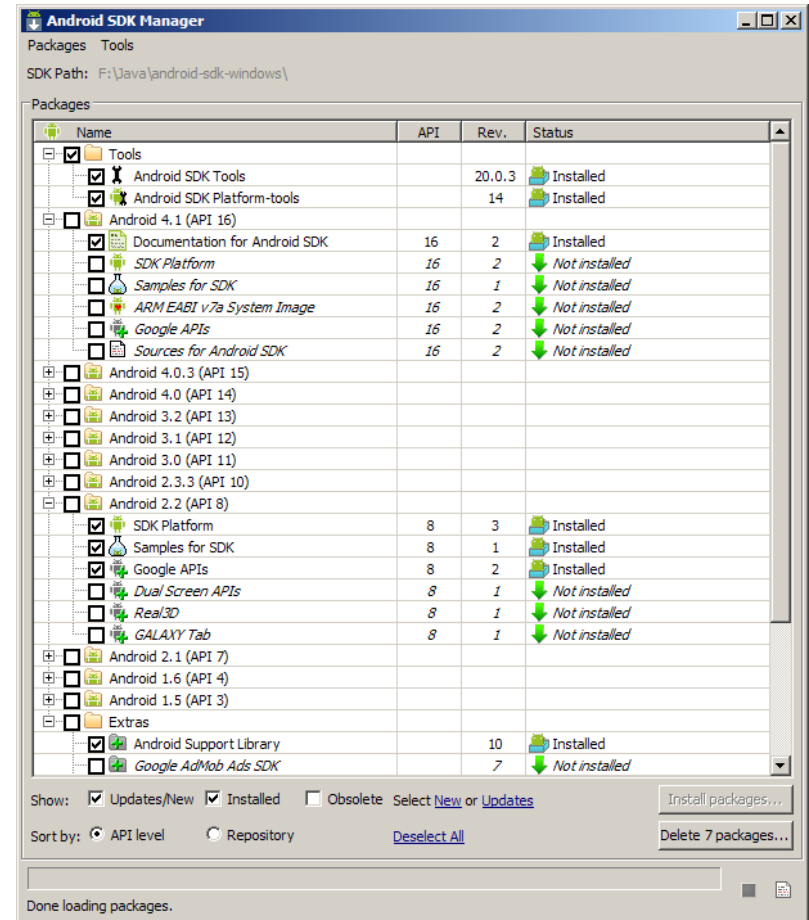
tools

AVD Manager

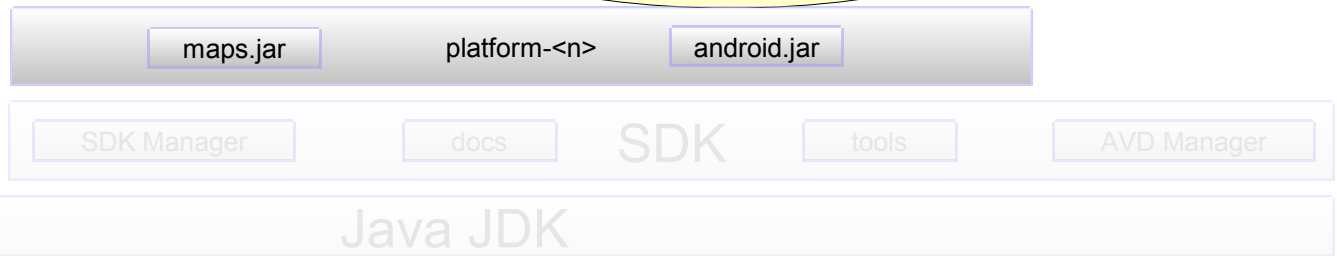
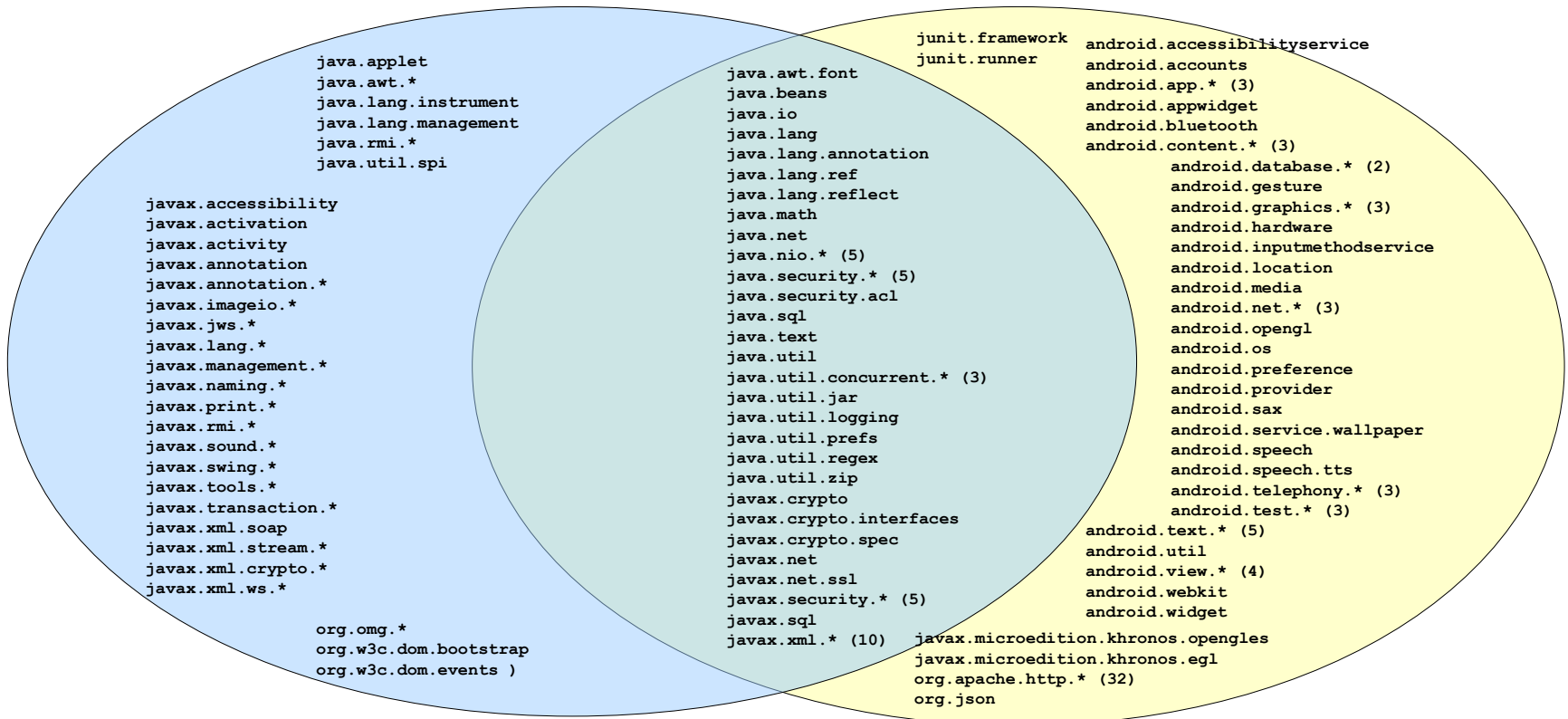
Java JDK

# Der SDK Manager

- Installieren und aktualisieren von
  - Plattformen und -Tools
  - Dokumentation
  - Quellen
  - Ergänzende Bibliotheken



# Platform – android.jar

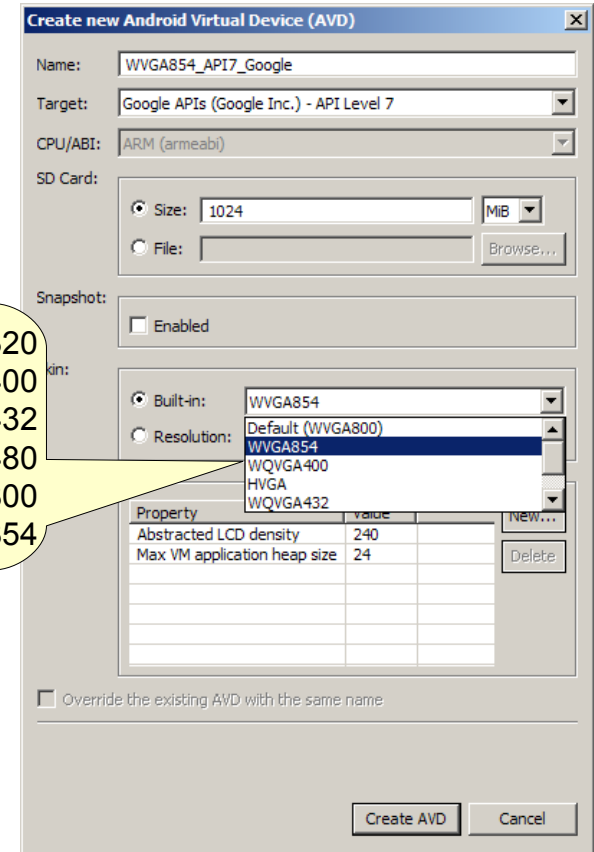




# Der AVD Manager

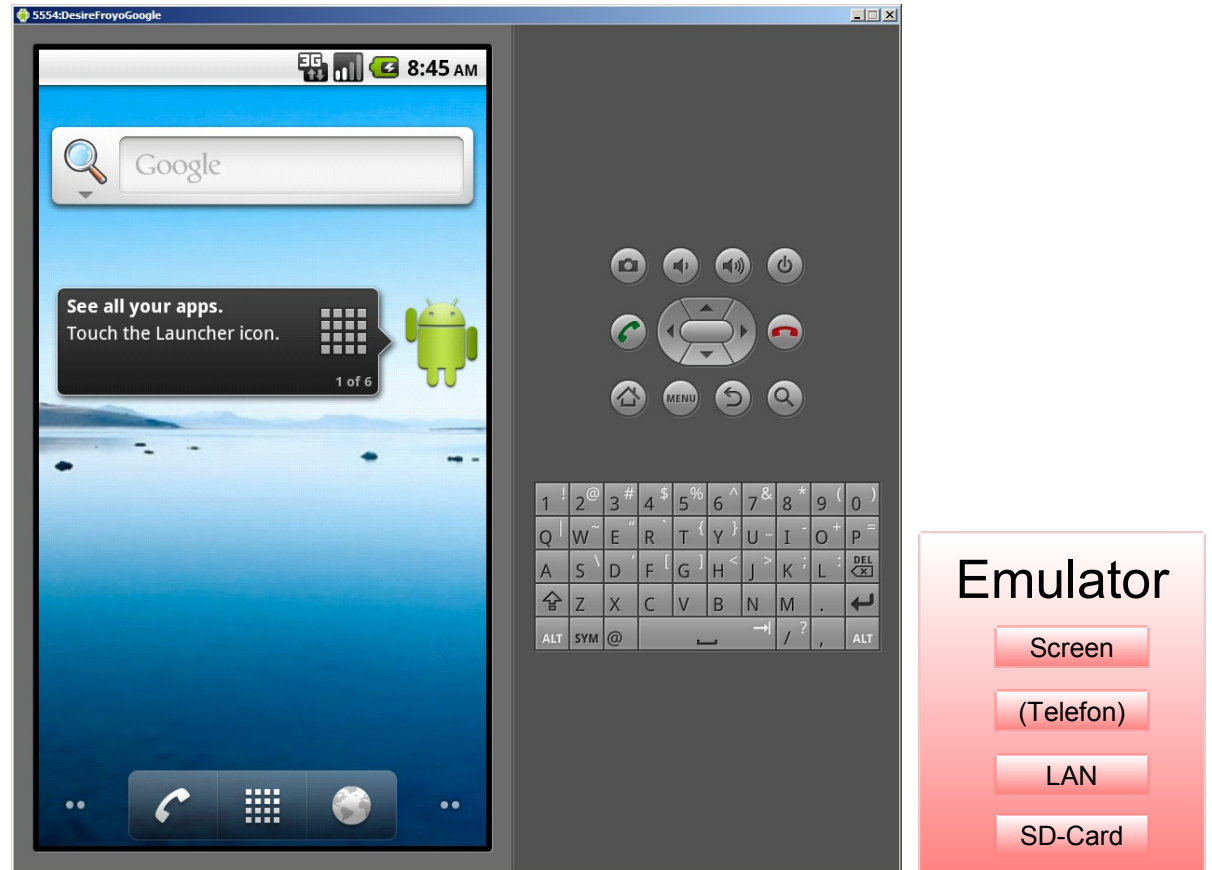
- Erstellen Virtueller Devices
- Images und Konfigurationen
- Varianten von
  - Bildschirmgröße
  - Pixeldichte
  - Hardwareausstattung
  - Speicherkartengröße und -Inhalt

QVGA	240x320
WQVGA400	240x400
WQVGA432	240x432
HVGA	320x480
WVGA800	480x800
WVGA854	480x854



# Emulator

- Führt ein AVD aus
- Aber langsam



```
>emulator -avd DesireWUGA800
```



# Mit Smartphone/Tablet entwickeln

- Geeignete USB-Treiber installieren
- Am Gerät
  - Einstellungen >
  - Anwendungen >
  - Entwicklung >
  - USB-Debugging aktivieren



## Device

Screen

Sensoren

GPS

Camera

Telefon

WLAN

SD-Card

## Emulator

Screen

(Telefon)

LAN

SD-Card

maps.jar

platform-<n>

android.jar

AVD

SDK Manager

docs

SDK

tools

AVD Manager

Java JDK

# ADB – Android Debug Bridge

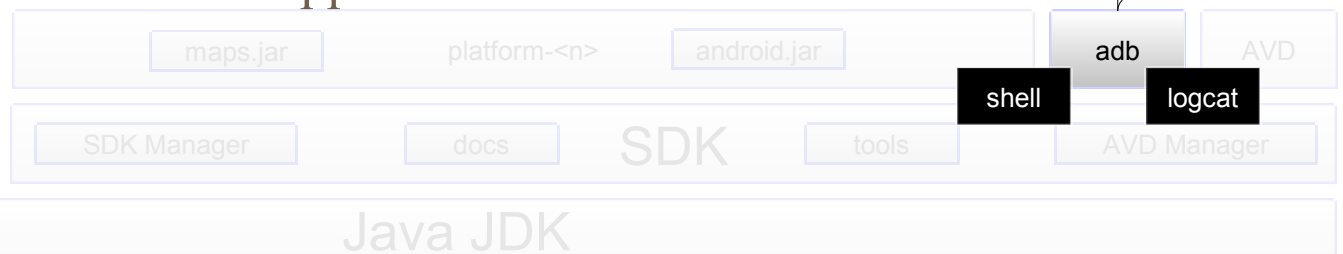
- Für Kommunikation mit Emulatoren und echten Geräten
  - Deviceliste
  - Logausgaben

```
F:\>adb devices
List of devices attached
emulator-5554    device
HT041PL10695   device
```

```
D/SurfaceFlinger< 92>: Layer::requestBuffer(this=0x6285b8), index=0, pid=92, w=480, h=762 success
D/SurfaceFlinger< 92>: Layer::requestBuffer(this=0x6285b8), index=1, pid=92, w=480, h=762 success
W/InputManagerService< 92>: Window already focused, ignoring focus gain of: com.android.internal.view.IInputMethodClient$Stub$Proxy@46703b78
I/ActivityManager< 92>: Starting activity: Intent < act=android.intent.action.MAIN cat=[android.intent.category.HOME] flg=0x10200000 cmp=com.htc.launcher/.Launcher >
D/Sensors < 92>: close_akn, fd=180
D/SurfaceFlinger< 92>: Layer::setBuffers(this=0x509730), pid=201, w=480, h=800
D/SurfaceFlinger< 92>: Layer::setBuffers(this=0x509730), pid=201, w=480, h=800
D/SurfaceFlinger< 92>: Layer::requestBuffer(this=0x509730), index=0, pid=201, w=480, h=800 success
D/SurfaceFlinger< 92>: Layer::requestBuffer(this=0x509730), index=1, pid=201, w=480, h=800 success
D/AR8973 < 72>: Compass CLOSE
```

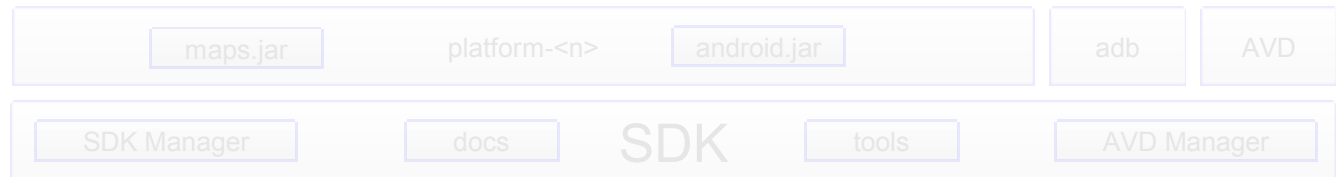
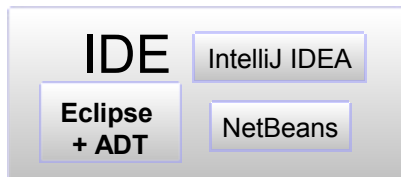
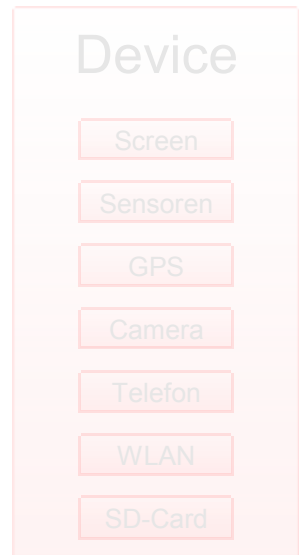
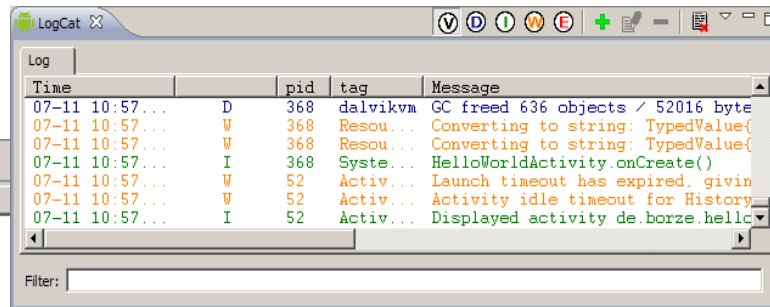
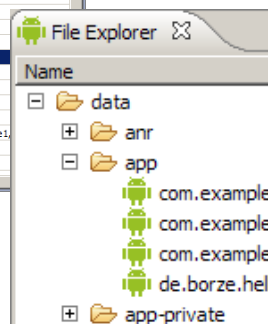
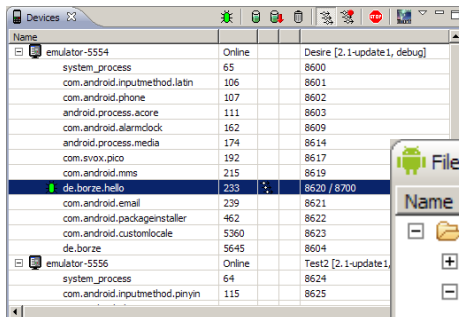
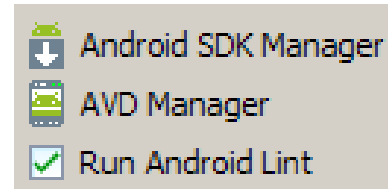
- Shell
- Dateiaustausch
- (De)Installation von Apps

```
F:\>adb -e shell
# ls
ls
config
cache
sdcard
```

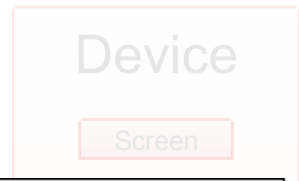


## IDE – Unterstützung

- Für alle namhaften IDEs
- Für Eclipse direkt von Google
- ADT – Android Developer Tools



# Die AndroidManifest.xml



- Identifizierung
  - Package
- Voraussetzungen
  - Hardware
  - Berechtigungen
  - Versionen
- Bestandteile

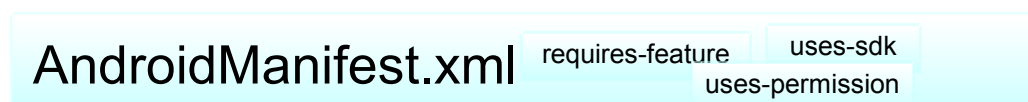
```

<manifest
  xmlns:android="http://schemas.android.com/apk/res/android"
  package="de.borze.http"
  android:versionCode="1"
  android:versionName="1.0" >

  <uses-sdk
    android:minSdkVersion="7" />

  <uses-permission
    android:name="android.permission.INTERNET" />
  <uses-permission
    android:name="android.permission.WRITE_EXTERNAL_STORAGE" />

  <uses-feature
    android:name="android.hardware.camera" />
    
```



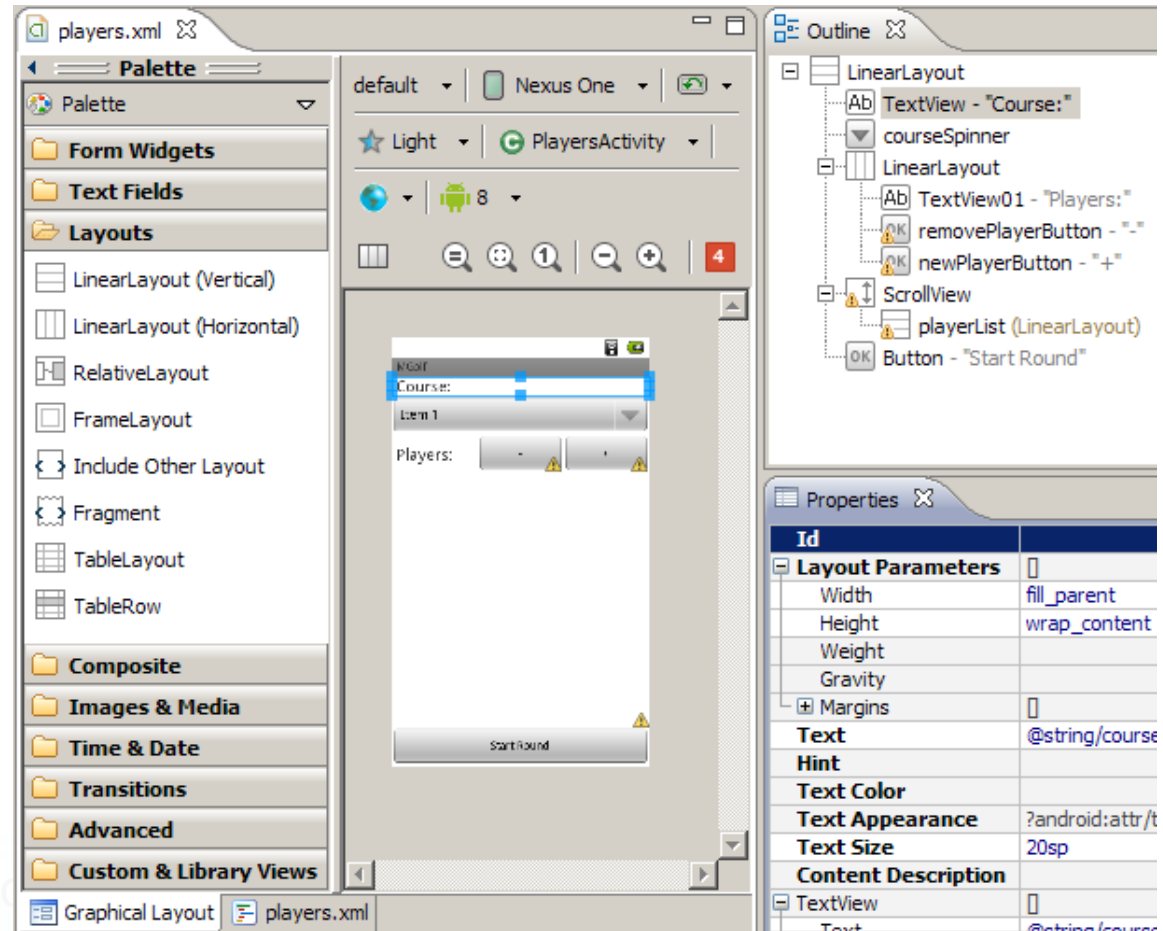
- A** Activity
- P** Provider
- R** Receiver
- S** Service



## View

- Oberflächen und -elemente
- Layouts
- Widgets
- CustomViews
- WYSIWYG

View



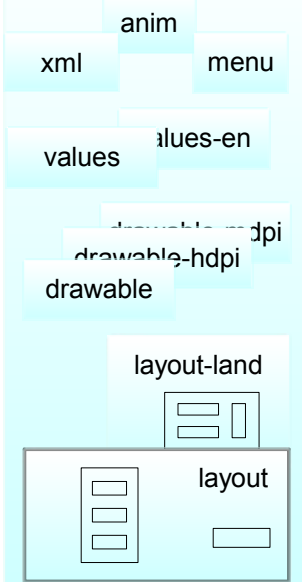
IDE  
Eclipse + ADT

maps.jar platform-<n> android.jar adb AVD  
SDK Manager docs SDK tools AVD Manager

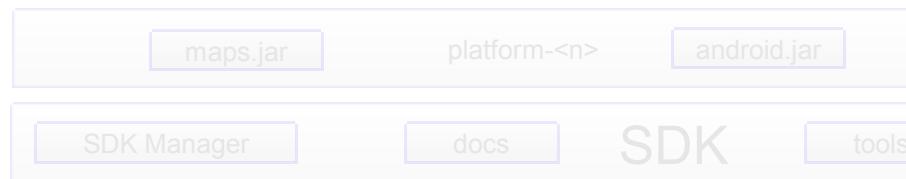
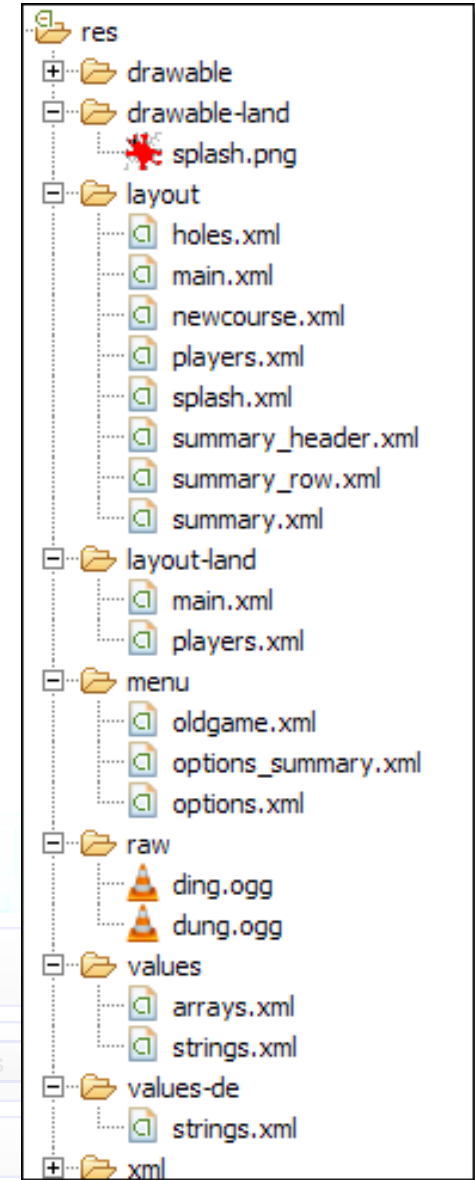
Java JDK

# Ressourcen

## Ressourcen



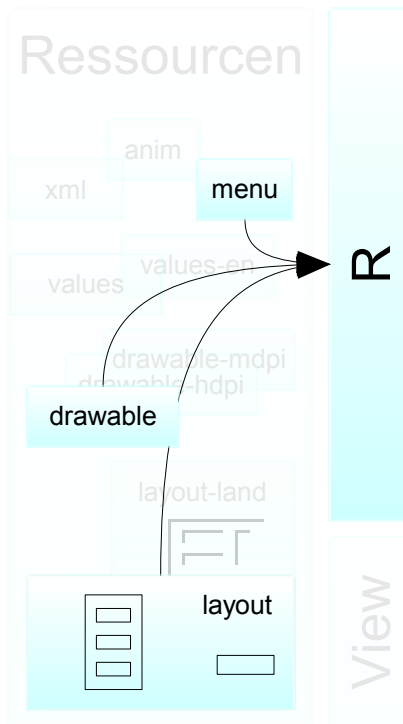
- Arten:
  - Texte, Bilder, Layouts, Menüs, Farben, Größen, Animationen
- Varianten:
  - Sprache, Land, Auflösung, Größe, Orientierung, Version





## R

- Zugriff auf Ressourcen
- Auswahl zur Laufzeit abhängig von Konfiguration



```

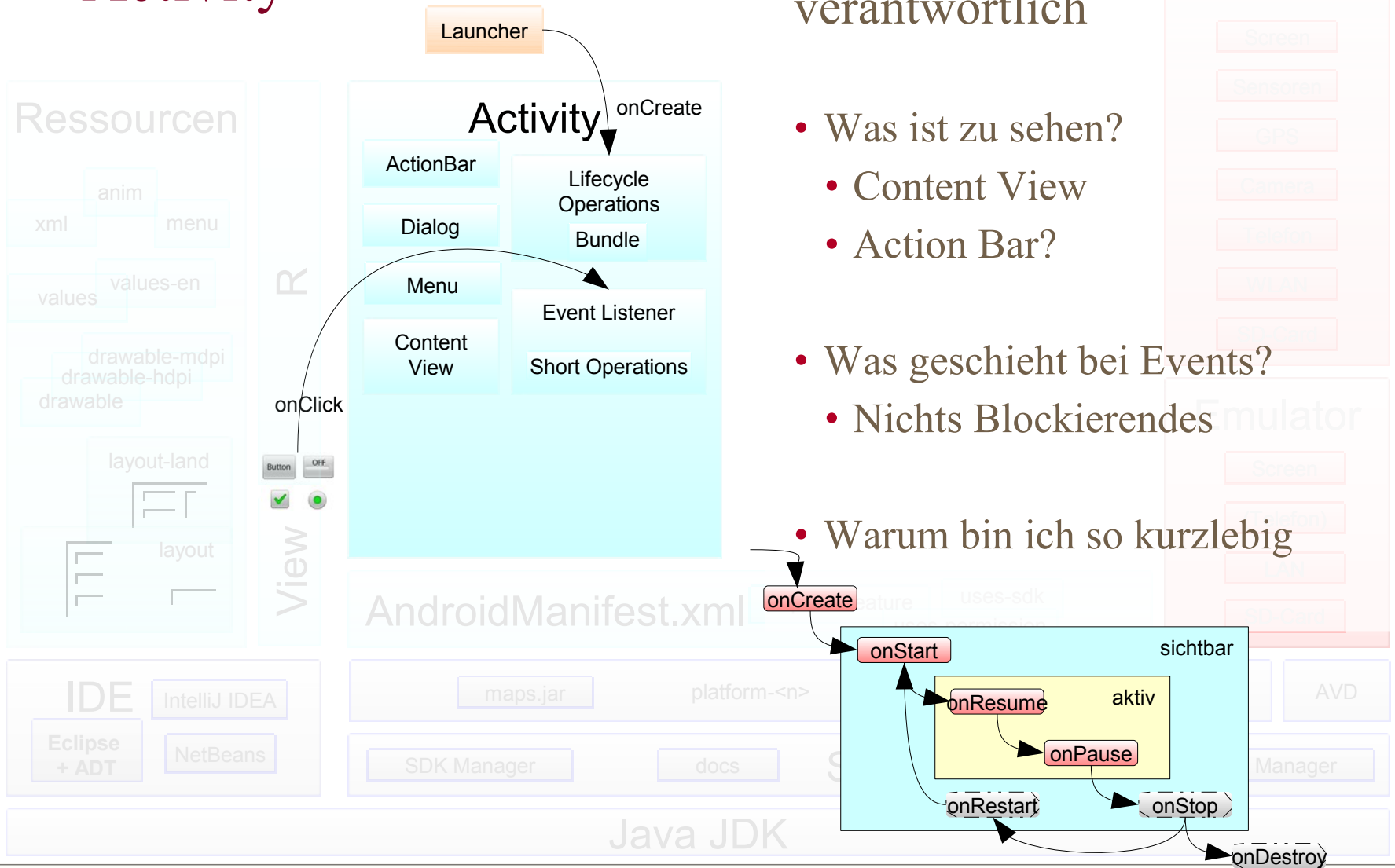
setContentView(R.layout.main);

Button button = (Button) findViewById(R.id.prefsButton);

MenuInflater menuInflater = getMenuInflater();
menuInflater.inflate(R.menu.menu, menu);
    
```



## Activity



- Für User Interface verantwortlich

- Was ist zu sehen?
  - Content View
  - Action Bar?

- Was geschieht bei Events?
  - Nichts Blockierendes

- Warum bin ich so kurzlebig

## Service

```
startService(new Intent(this, SynchronizerIntentService.class));
```

- Hintergrundoperationen
  - Startbar von User oder bei Systemereignissen
- Öffentliche Dienste
  - AIDL
  - Aufrufbar von anderen Prozessen
  - Generierte Stub + Proxy



```
package de.borze.service;

interface IAdderService {

    int add(int op1, int op2 );
}
```

```
case TRANSACTION_add:
{
    data.enforceInterface (DESCRIPTOR);
    int _arg0;
    _arg0 = data.readInt ();
    int _arg1;
    _arg1 = data.readInt ();
    int _result = this.add(_arg0, _arg1);
    reply.writeNoException ();
    reply.writeInt (_result);
    return true;
}
```

App ← Laufzeitumgebung

Action	Scheme	Startet Activity für
Intent.ACTION_DIAL	tel:	Telefon
Intent.ACTION_VIEW	http:	Internet
Intent.ACTION_SENDTO	mailto:	Email
Intent.ACTION_VIEW	geo:	Map

```
new Intent(Intent.ACTION_DIAL, Uri.parse("tel:22 22 22 "));
```

# Intent

Intent

Service

```
new Intent(context, UpdateService.class);
```

- Nachricht zum Start einer Activity oder Service
- Dedizierte Klasse
- Standard-Action und Uri
  - Objekt oder Content Ressource
  - Wird von Laufzeitumgebung aufgelöst
- Custom Action mit Extra-Daten

"content://media/external/audio/media/" + id

App

Laufzeitumgebung

Device

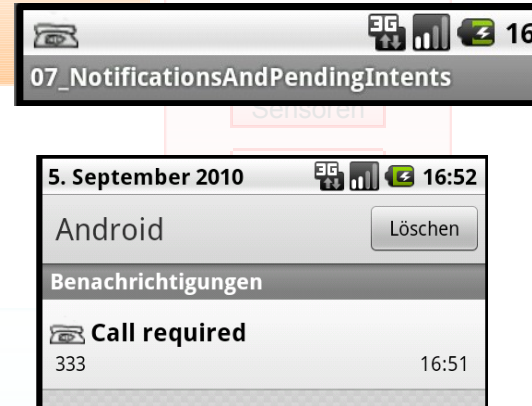
NotificationManager

# Notification und Pending Intent

- Hinweis an Benutzer
- Ignorieren
- Aktivität starten



Intent



```

// Erzeuge mit Icon, Text und Vibrationsmuster für StatusBar
Notification callNotification = new Notification(
    R.drawable.tel, "Call", System.currentTimeMillis());
callNotification.defaults |= Notification.DEFAULT_SOUND;
callNotification.flags = Notification.FLAG_AUTO_CANCEL;
callNotification.vibrate = new long[]{500,500,500,500};

// Definiere später auszulösende Aktion....
Intent callIntent = new Intent(Intent.ACTION_DIAL, Uri.parse("tel:" + number));
PendingIntent callPendingIntent =
    PendingIntent.getActivity(this, 0, callIntent, Intent.FLAG_ACTIVITY_NEW_TASK);

// Setze Titel, Text, auszulösende Aktion für geöffnete Benachrichtigung
callNotification.setLatestEventInfo(this, "Call required", number, callPendingIntent);

// Versende Benachrichtigung
NotificationManager notificationManager =
    (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);
notificationManager.notify(1, callNotification);
  
```

## Laufzeitumgebung

### Broadcast Receiver

- Erhalten Nachricht bei Systemereignissen

- Boot completed
- SMS received
- WiFi connected

- Battery low

- Können Services starten

### Broadcast Receiver



### Service

Long Operations

AIDL

## Device

Screen

Sensoren

GPS

Camera

Telefon

WLAN

SD-Card

## Emulator

Screen

(Telefon)

LAN

SD-Card

## Laufzeitumgebung



## Widget

- Injizieren RemoteView in Oberfläche

- Periodisch aktualisierbar

- Broadcast

- Alarm

```
for (int widgetId : appWidgetIds) {
    RemoteViews remoteViews =
        new RemoteViews(context.getPackageName(), R.layout.widget_text);
    remoteViews.setTextViewText(R.id.appwidget_text, date);
    appWidgetManager.updateAppWidget(widgetId, remoteViews);
}
```

```
remoteViews.setOnClickPendingIntent(R.id.imageViewStop, pendingStopIntent);
```

- Stark eingeschränkte Interaktion

## Laufzeitumgebung





## Laufzeitumgebung

android.R

App

Notifications

AlarmManager

SQLite

```
String titleColumn = cursor.getColumnIndex(Browser.HISTORY_PROJECTION_TITLE_INDEX);
String urlColumn = cursor.getColumnIndex(Browser.HISTORY_PROJECTION_URL_INDEX);

ListAdapter adapter = new SimpleCursorAdapter(
    this,
    android.R.layout.simple_list_item_2,
    cursor,
    new String[] {titleColumn, urlColumn},
    new int[] {android.R.id.text1, android.R.id.text2});
setListAdapter(adapter);
```

## Device

Screen

Sensoren

GPS

Camera

Telefon

WLAN

SD-Card

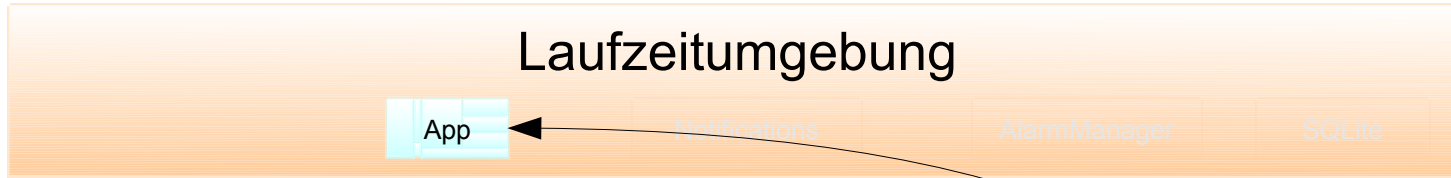
ListActivity

Adapter

## ListActivity

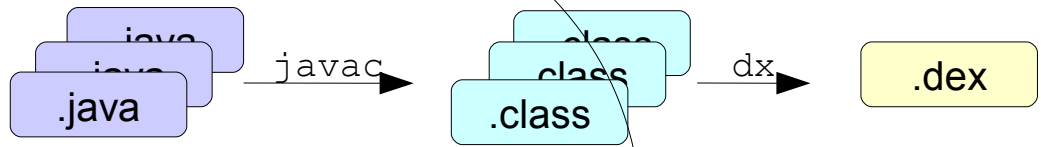
- Adapter verbindet
  - Views eines Teil-Layouts
  - Projektion eines Cursors
  - Custom oder android.R

44 SQLite Database	
Vanille	193
Erdbeer	165
Schokolade	204
Schokosplitter	128
Karamel	215
Marzipan	365



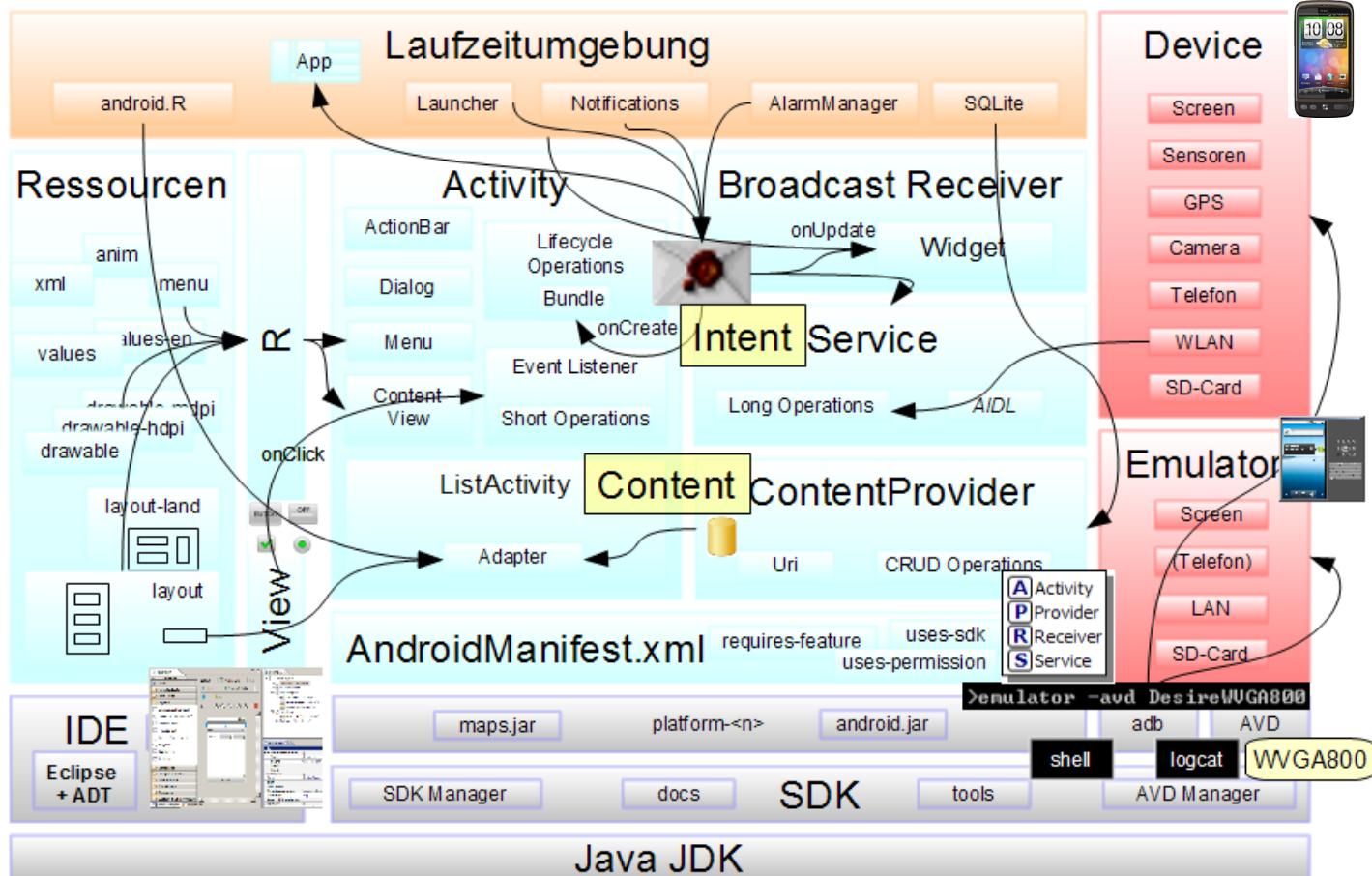
# APK

- Klassen (und Libraries) werden ge-dex
- Ressourcen und Dex werden zu APK-Datei gezippt, signiert und aligned
- Direkt installiert, über Firmen-Server, über GooglePlay oder alternativem „Store“



# Q & A

- Wie Sie sehen, ist Android ganz einfach...



3.– 6. September 2012  
in Nürnberg



# Herbstcampus

Wissenstransfer  
par excellence

Vielen Dank!

Manfred Borzechowski

LambdaLogic Informationssysteme GmbH