

# **Tutorial Day – Android Session**

**22.06.2013, Susanne Braun**

- Diplom-Informatikerin
- Senior Developer
- Committer Android Apps bei der Deutschen Bahn
- Studien und Workshops im Bereich Mobile/Android
- Autorin von Artikeln für Fachzeitschriften Mobile Technology und android 360

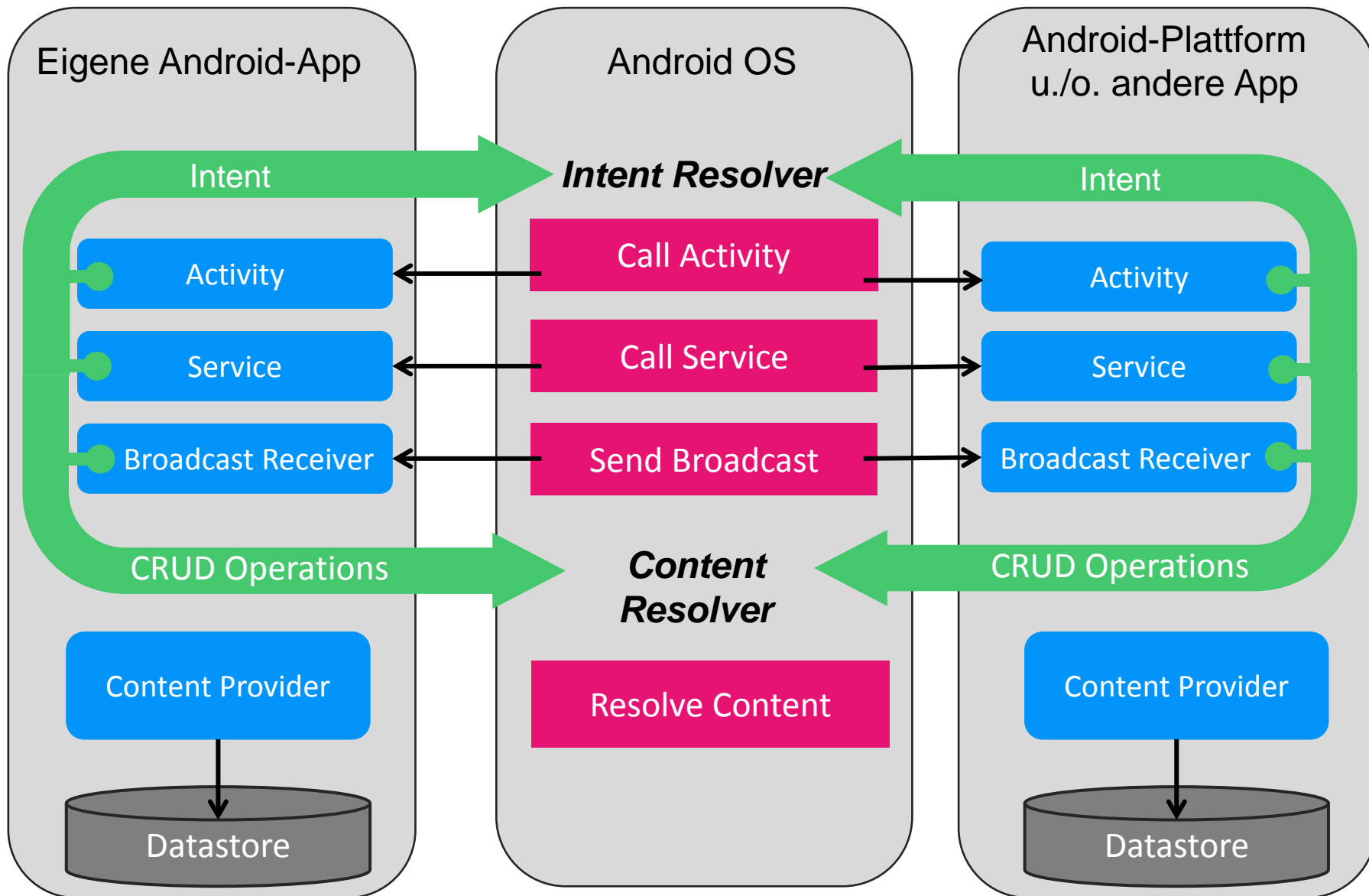


***Susanne Braun***

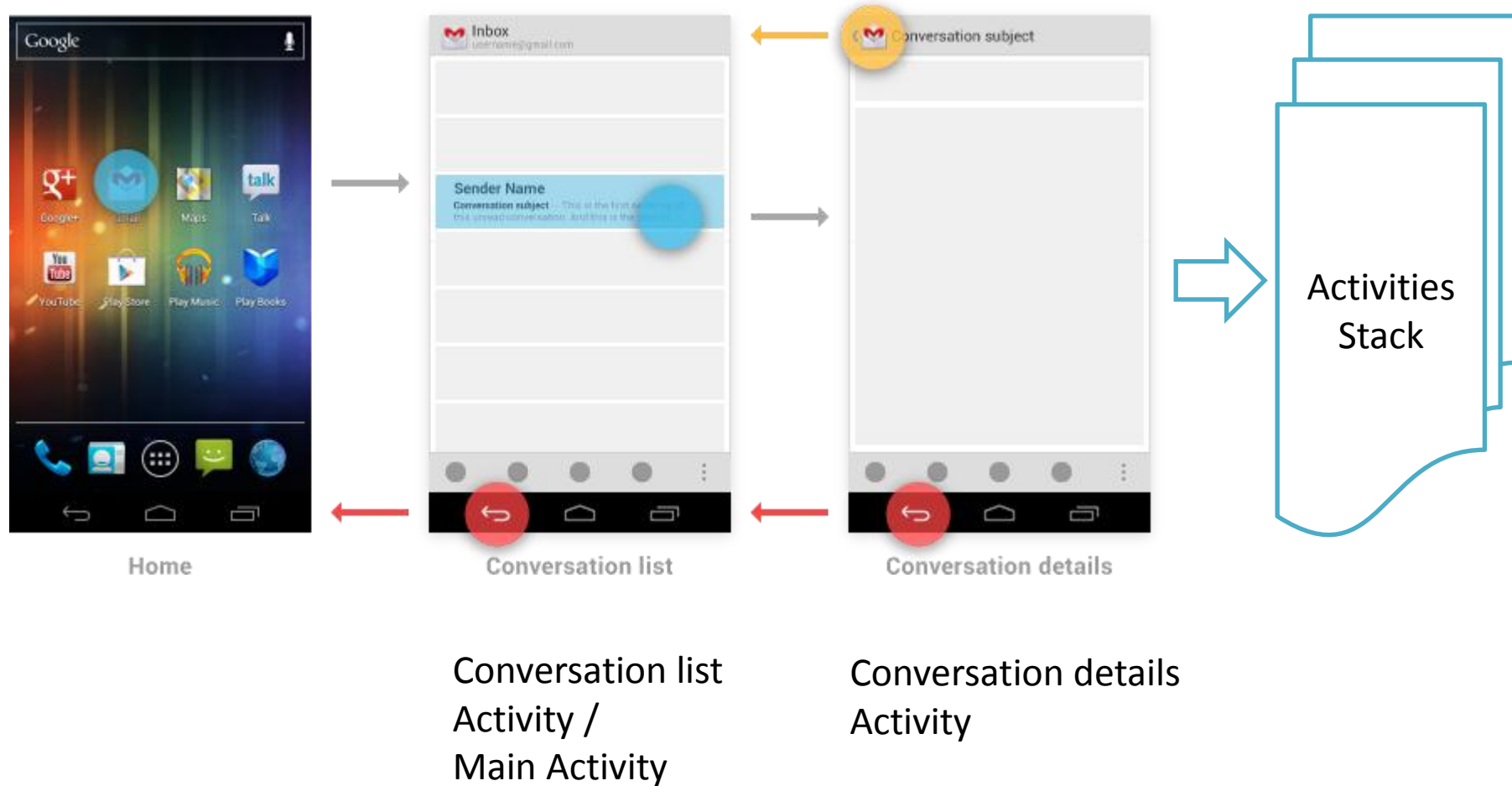
- Die Firma Android wurde 2003 von Andy Rubin gegründet
- 2005 übernimmt Google Android für 50 Mio USD
- Rubin wird zum Vice President of Engineering bei Google
- Seit Oktober 2008 ist Android offiziell verfügbar
- Page & Brin: „Beste Übernahme aller Zeiten“
- Google hat Schätzungen zufolge im Jahr 2012 bereits 1,3 Milliarden USD mit Werbung durch Android-Devices umgesetzt



- Das Android-OS basiert auf einem 2.6-er Linux-Kernel
- Jede App wird in einer Application Sandbox ausgeführt
- Dabei bekommt jede App eine eindeutige User-ID und wird stets mit dieser in einem eigenen Prozess ausgeführt
- Normalfall: App wird in Java programmiert (Compiler Compliance SE 5.0 oder SE 6.0)
- Übersetzung von javac-class-Files in Dalvik-VM-Bytecode durch „dex“
- Java API ist Teilmenge des JDKs (zum Beispiel keine Swing oder AWT-Klassen)



- Activity korrespondiert i.d.R mit einem bestimmten „Screen“ der App
- Anwendungskomponente zur visuellen Interaktion mit dem Nutzer
- Activity ist für Darstellungslogik des UI und die Entgegennahme und Verarbeitung der Nutzereingaben verantwortlich (View + Controller)
- App besteht i.d.R. aus mehreren Activities, die **loose gekoppelt** sind.
- Interaktion/Kopplung der Activities erfolgt über **Intents**.
- Activities besitzen einen Lebenszyklus (Lifecycle), der von der Android Laufzeitumgebung gemanaged wird





## Activity-Class



## Resources:

- Layout
- Menu
- Strings
- Styles
- Drawables
- ...

```
public class NotesActivity extends Activity {  
  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_notes);  
    }  
  
    @Override  
    public boolean onCreateOptionsMenu(Menu menu) {  
        getMenuInflater().inflate(R.menu.activity_notes,  
            menu);  
        return true;  
    }  
}
```

```
<RelativeLayout  
    xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent" >  
  
    <EditText  
        android:id="@+id/notesTextEdit"  
        android:layout_width="match_parent"  
        android:layout_height="match_parent"  
        android:ems="10"  
        android:gravity="top/center_vertical"  
        android:hint="@string/enter_notes"  
        android:inputType="textMultiLine" >  
  
        <requestFocus />  
    </EditText>  
  
</RelativeLayout>
```





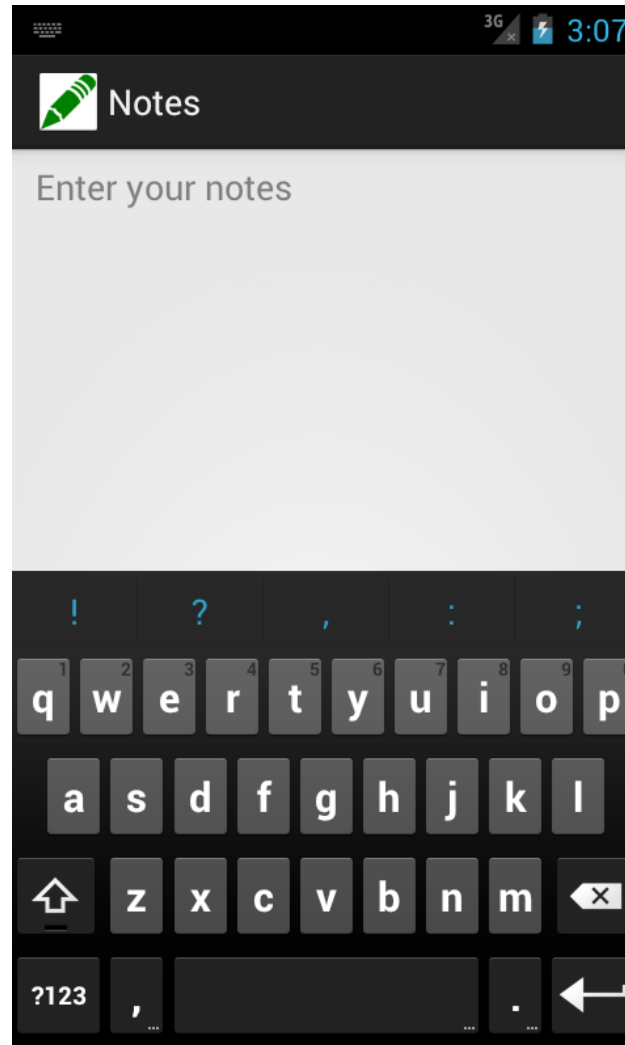
aapt-Generator  
(Android Dev Tools)



R.java Class

```
<resources>
  <string name="app_name">Notes</string>
  <string name="hello_world">Hello world!</string>
  <string name="menu_settings">Settings</string>
  <string name="title_activity_notes">Notes</string>
  <string name="enter_notes">Enter your notes</string>
</resources>
```

```
public final class R {
  ...
  public static final class string {
    public static final int app_name=0x7f040000;
    public static final int enter_notes=0x7f040004;
    public static final int hello_world=0x7f040001;
    public static final int menu_settings=0x7f040002;
    public static final int title_activity_notes=0x7f040003;
  }
  ...
}
```



strings.xml (String-Ressource):

```
<resources>

    <string name="app_name">Notes</string>
    <string name="hello_world">Hello world!</string>
    <string name="menu_settings">Settings</string>
    <string name="title_activity_notes">Notes</string>
    <string name="enter_notes">Enter your notes</string>

</resources>
```

activity\_notes.xml (Layout-Ressource):

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

    <EditText
        android:id="@+id/notesTextEdit"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:ems="10"
        android:gravity="top/center_vertical"
        android:hint="@string/enter_notes"
        android:inputType="textMultiLine" >

        <requestFocus />
    </EditText>

</RelativeLayout>
```

activity\_notes.xml (Menu-Ressource):

```
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:id="@+id/menu_settings"
          android:title="@string/menu_settings"
          android:orderInCategory="100"
          android:showAsAction="never" />
</menu>
```

styles.xml (Style-Ressource):

```
<resources>  
    <style name="AppTheme" parent="android:Theme.Light" />  
</resources>
```

R.java:

```
public final class R {  
    public static final class id {  
        public static final int menu_settings=0x7f070001;  
        public static final int notesTextEdit=0x7f070000;  
    }  
    public static final class layout {  
        public static final int activity_notes=0x7f030000;  
    }  
    public static final class menu {  
        public static final int activity_notes=0x7f060000;  
    }  
    public static final class string {  
        public static final int app_name=0x7f040000;  
        public static final int enter_notes=0x7f040004;  
        public static final int hello_world=0x7f040001;  
        public static final int menu_settings=0x7f040002;  
        public static final int title_activity_notes=0x7f040003;  
    }  
    public static final class style {  
        public static final int AppTheme=0x7f050000;  
    }  
}
```

## NotesActivity.java (Activity-Class):

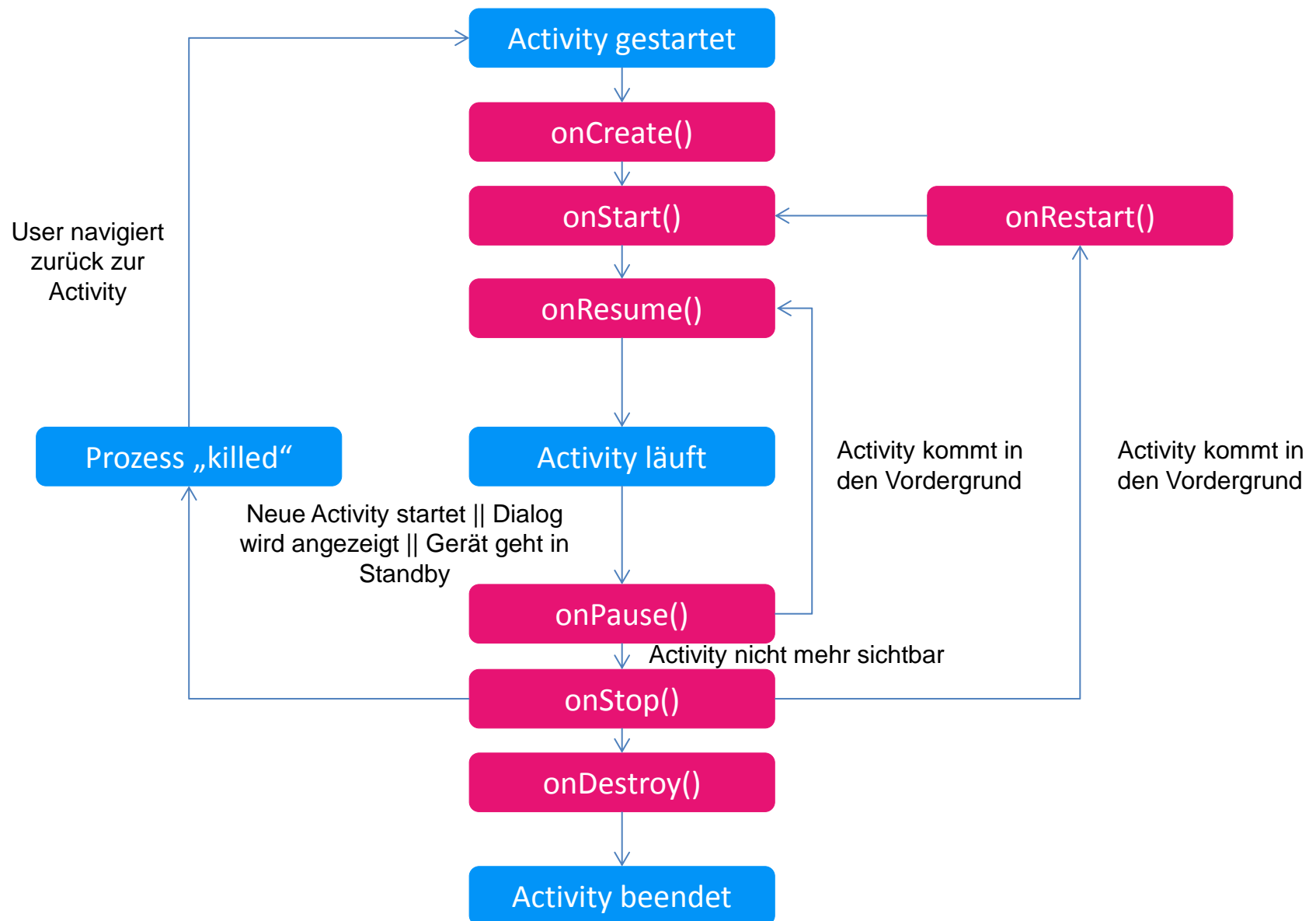
```
public class NotesActivity extends Activity {  
  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_notes);  
    }  
  
    @Override  
    public boolean onCreateOptionsMenu(Menu menu) {  
        getMenuInflater().inflate(R.menu.activity_notes, menu);  
        return true;  
    }  
}
```

Transienten Zustand  
wiederherstellen

View initialisieren

Menü initialisieren





## Wichtige Callbacks:

| Callback  | Zweck  |
|---|--|
| <code>onCreate(Bundle savedInstanceState)</code>  | Initialisierung oder transienten Zustand wiederherstellen                        |
| <code>onPause()</code>                            | Alle persistenten ungesicherten Daten speichern                                  |
| <code>onSaveInstanceState(Bundle outState)</code> | Transienten Zustand im Bundle sichern, komplexe Objekte ggf. statisch speichern. |
| <code>onDestroy()</code>                          | Ressourcen releasen (z.B. Threads stoppen, Cursor schließen etc.)                |

- Realisierung von looser Kopplung zwischen den Laufzeit-Komponenten eines Android-Systems
- und Wiederverwendbarkeit von (Fremd-) Komponenten
- Komponenten sind z.B. **Activities**, Services, Broadcast-Receiver...
- *„Ich hätte gerne den Zucker“*
- Beschreiben eine Absichtserklärung
- ...oder das Eintreten eines bestimmten Ereignisses

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

    <uses-sdk
        android:minSdkVersion="15"
        android:targetSdkVersion="15" />

    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".NotesActivity"
            android:label="@string/title_activity_notes" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

- Kontext der App
- Klasse *android.content.Context*
- Die Activity-Klasse ist eine Subklasse von *Context*
- Referenz auf Context-Objekt wird oft benötigt um
  - neue Objekte zu erzeugen
  - Zugriff auf Ressourcen und wichtige Systemdienste zu erhalten
  - Zugriff auf andere Komponenten (Activities, Services) zu erhalten

## Beispiele

```
TextView tv = new TextView(ctx)

ctx.startActivity(intent)

ctx.getSystemService(LOCATION_SERVICE)

ctx.getContentResolver().query(uri, ...)

ctx.getResources()

ctx.getString(resId)

ctx.getAssets()

ctx.openOrCreateDatabase(...)

...
```



## Hintergrund: GUI-Thread nicht blockieren!

```
public class DownloadFileTask extends AsyncTask<URL, Integer, Long> {

    protected Long doInBackground(URL... urls) {
        int count = urls.length;
        long totalSize = 0;
        for (int i = 0; i < count; i++) {
            totalSize += Downloader.downloadFile(urls[i]);
            publishProgress((int) ((i / (float) count) * 100));
            // Escape early if cancel() is called
            if (isCancelled()) break;
        }
        return totalSize;
    }

    protected void onPostExecute(Long result) {
        showDialog("Downloaded " + result + " bytes");
    }

    @Override
    protected void onProgressUpdate(Integer... progress) {
        setProgressPercent(progress[0]);
    }
}
```

...

- // [braun@unix-ag.uni-kl.de](mailto:braun@unix-ag.uni-kl.de)
- // [twitter.com/susannebraun](https://twitter.com/susannebraun)
- // [github.com/susannebraun](https://github.com/susannebraun)