FACHHOCHSCHULE ERFURT UNIVERSITY OF APPLIED SCIENCES
Angewandte Informatik

Mobile Meetup

Android

Prof. Dr. Steffen Avemarg

October 19th, 2017

Outline

Mobile Ecosystem

Platforms

BlackBerry

iOS

android

webOS

bada

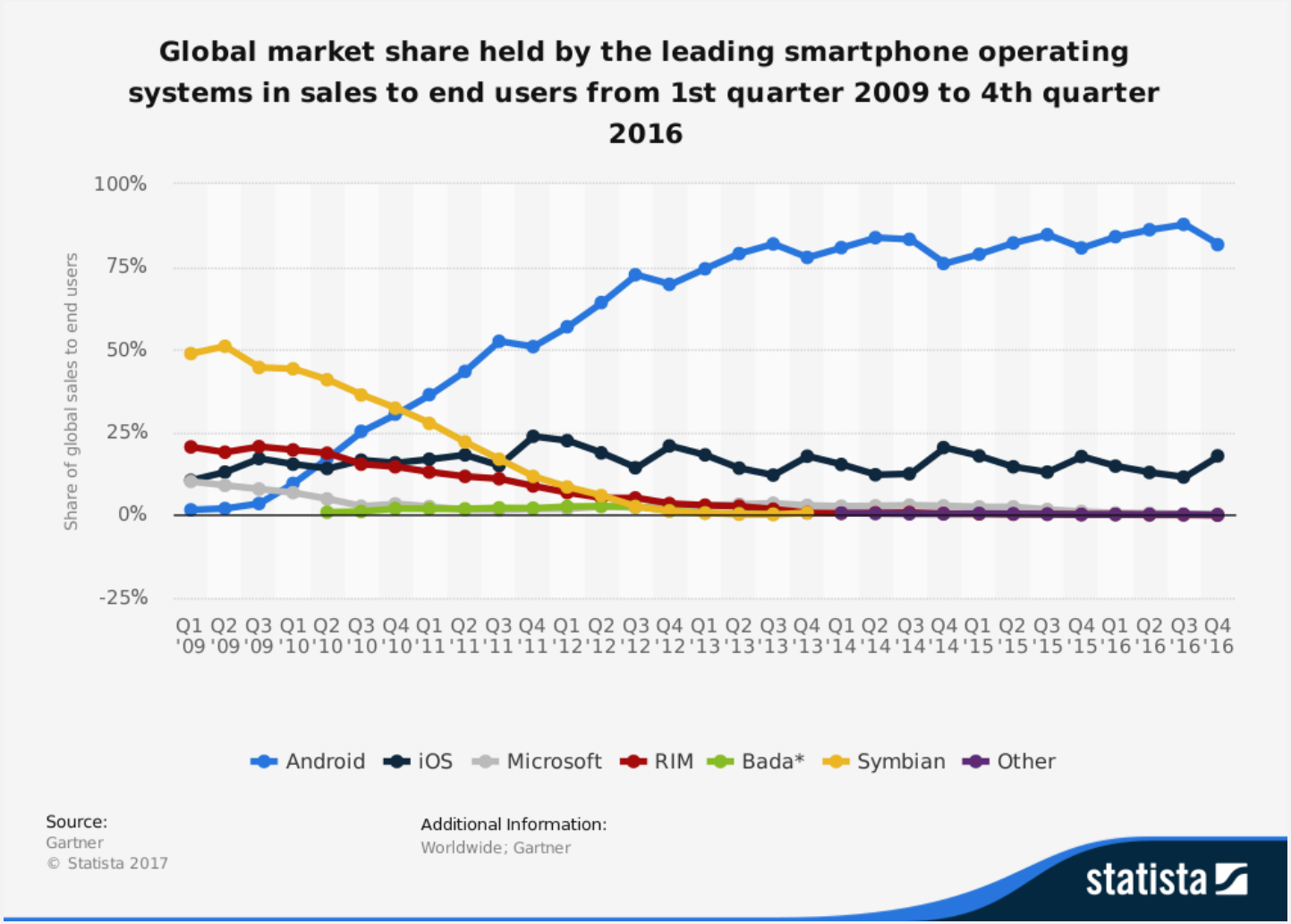
TIZEN

symbian

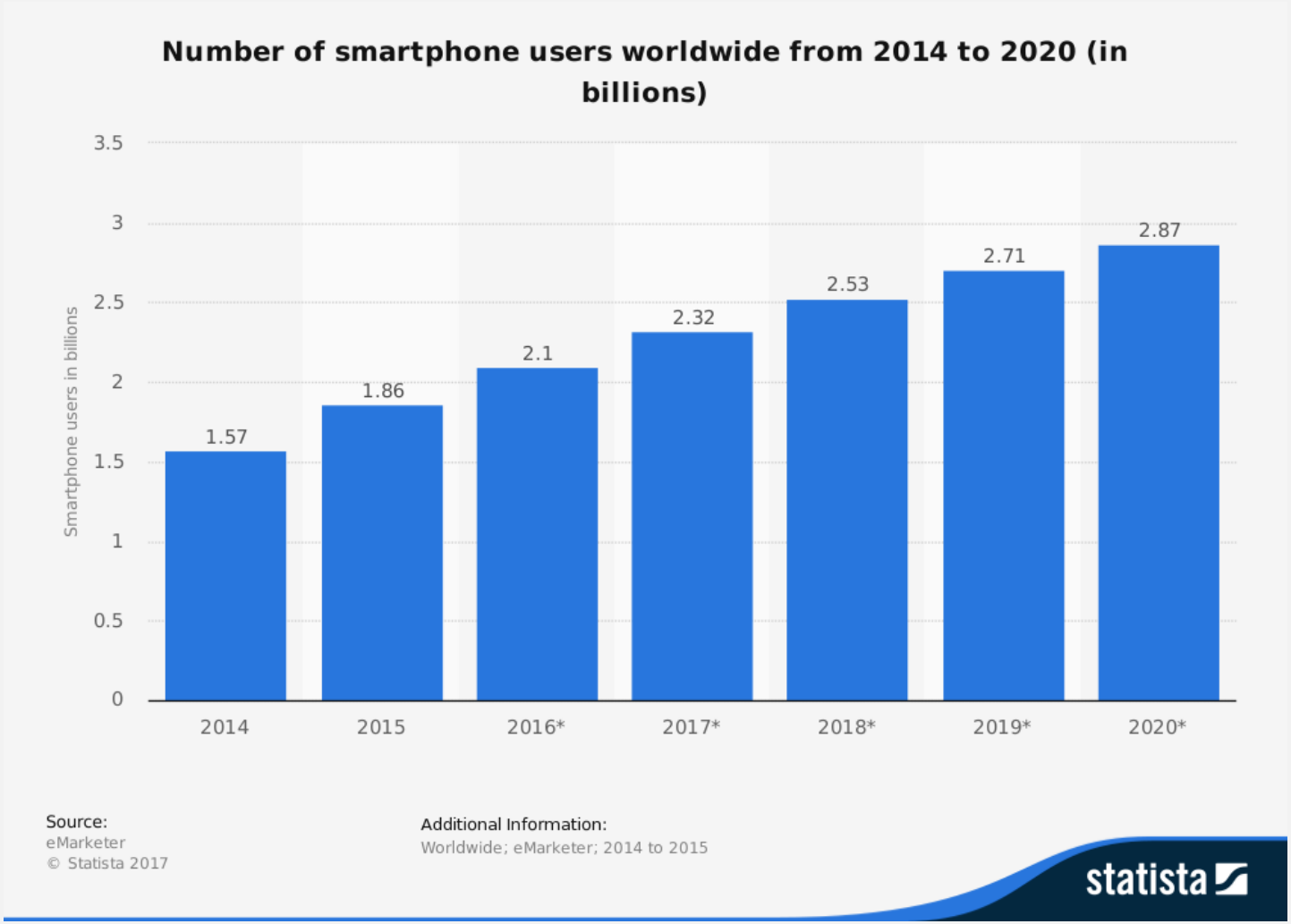
Windows 10 Mobile

ubuntu

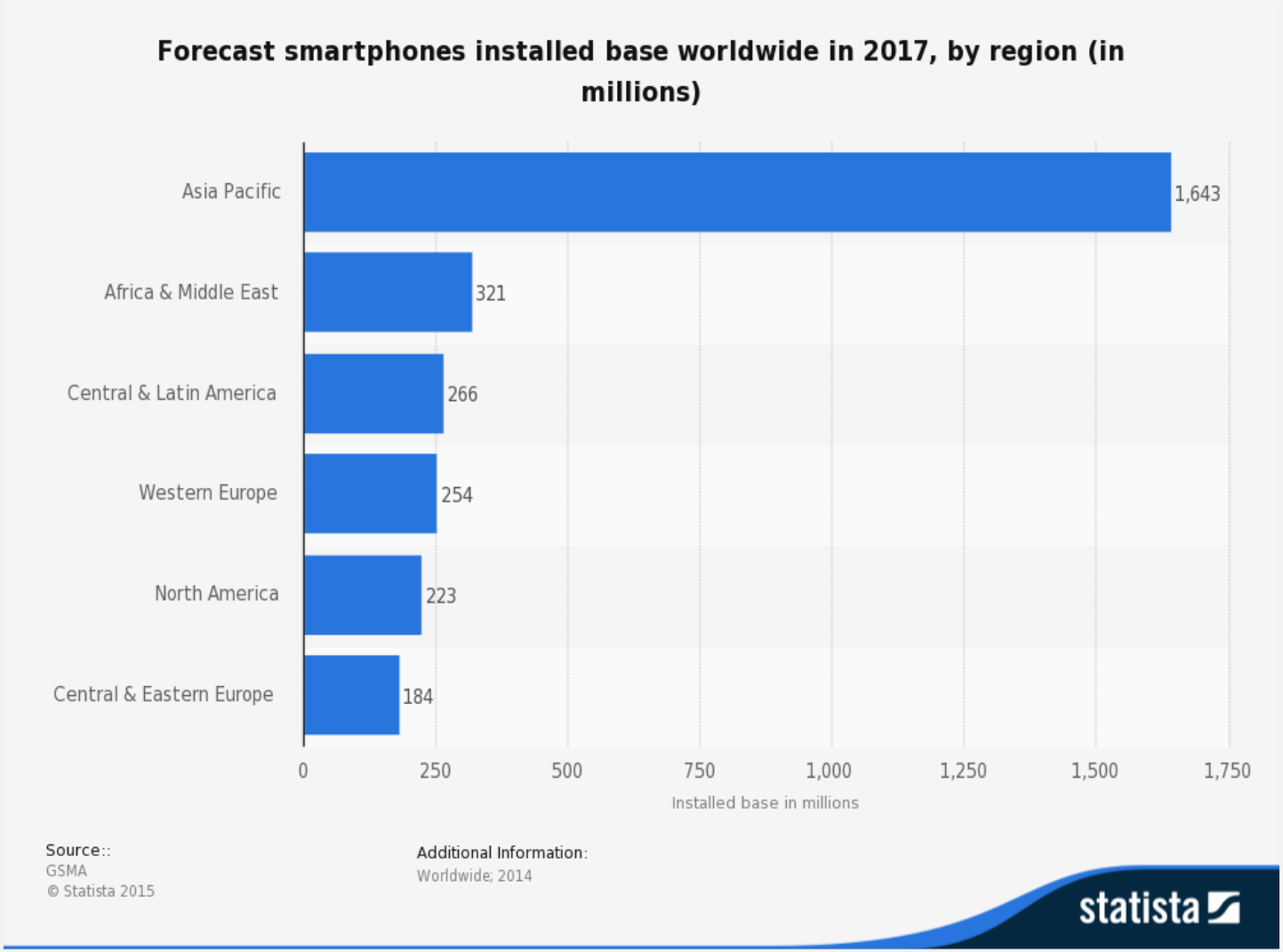
Market Statistics I



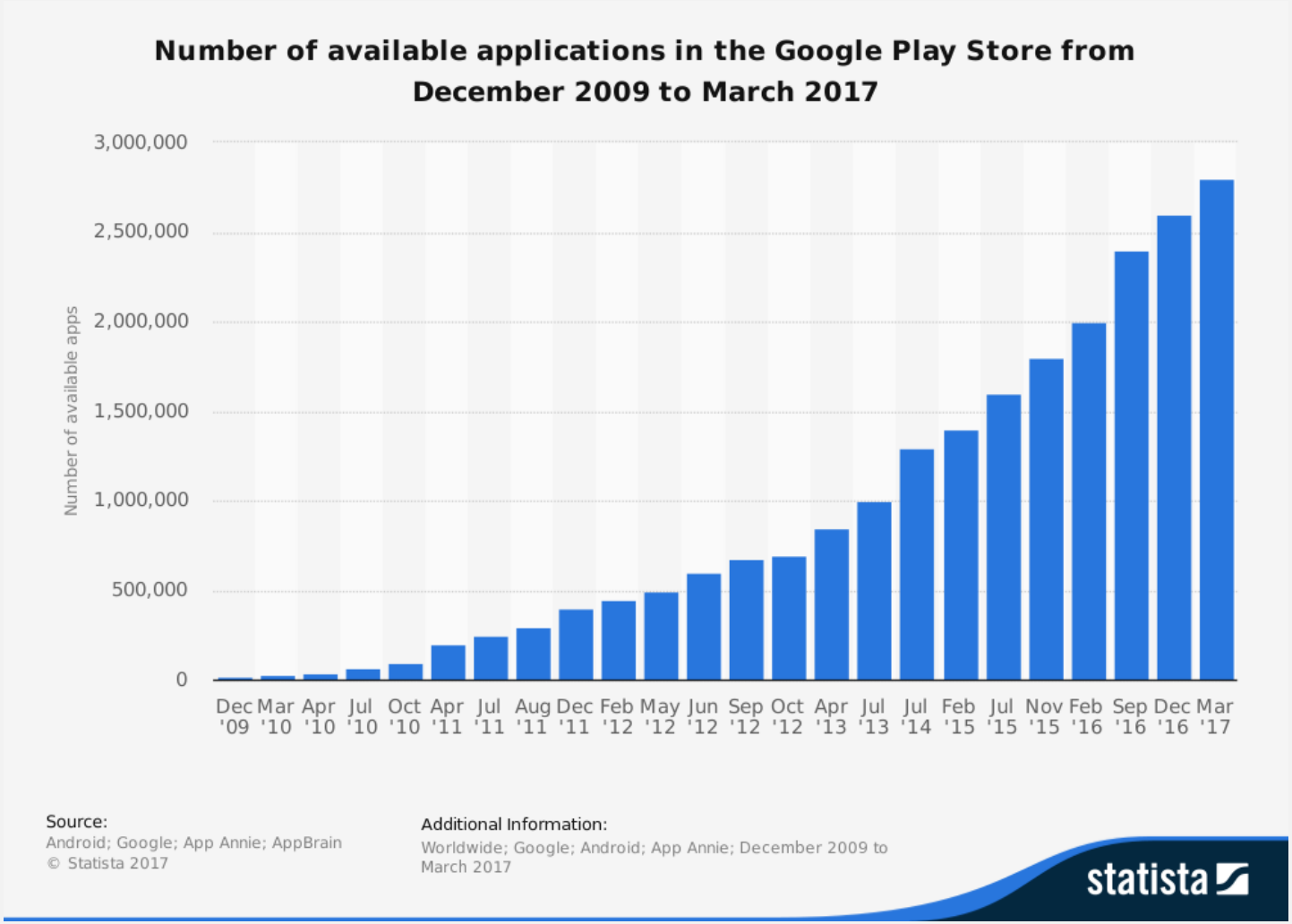
Market Statistics II



Market Statistics III



Market Statistics IV



Mobile Application Types

Application Types I

Native Applications

- Specific for just one platform
- Using development tools provided by platform maintainer
- Distribution using the platform-specific stores

Pros

- Full access to all platform capabilities
- Best performance
- Probably best UX

Cons

- Huge efforts when targeting more than one platform
- Dependent on platform

Application Types II

Mobile Web Applications

- Using web languages and tools like HTML, Javascript, CSS
- Browser is your engine, thus nearly all devices can be reached
- You can apply all your web knowhow

Pros

- Not dependent on any specific platform
- Easier maintenance & fast updates
- No installation

Cons

- Only capabilities offered by all platforms can be use, e.g. lowest common denominator

Application Types III

Hybrid Applications

- Mobile web app wrapped in a native browser container
- Access to some device capabilities
- Examples: Adobe PhoneGap resp. Apache Cordova

Pros

- A mobile web app will be wrapped in a native container
- Container provides access to some device capabilities

Cons

- Only capabilities offered by all platforms can be use, e.g. lowest common denominator

Application Types IV

Cross-Platform Applications

- Using different than the native tools and frameworks
- However, compile/transpile to machine code, e.g. no browser
- Examples: Xamarin, React Native, NativeScript

Pros

- One code base
- Native performance
- Access to some device capabilities

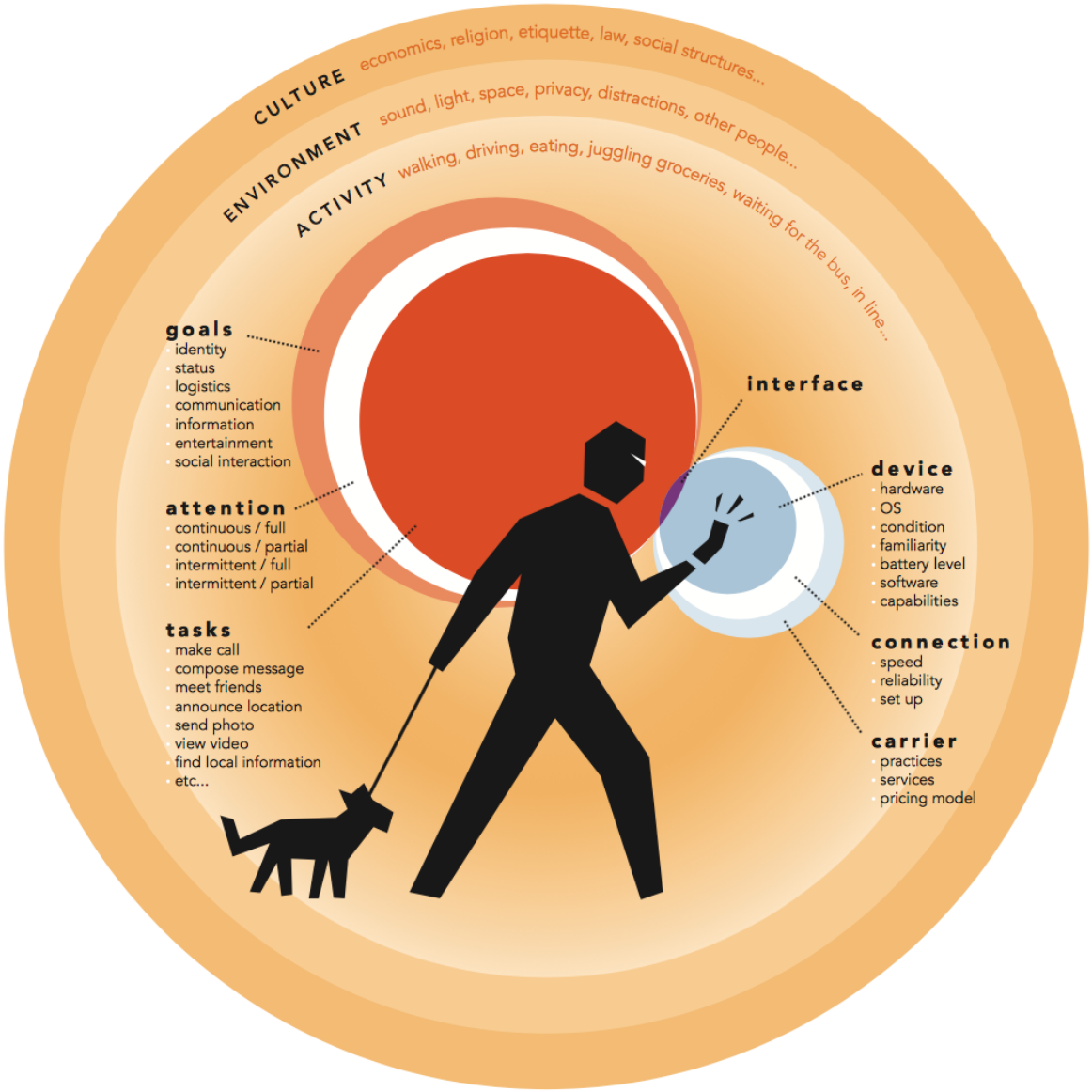
Cons

- Some frameworks use own UI library
- Dependent on toolchain

Mobile Development

Mobile Interaction

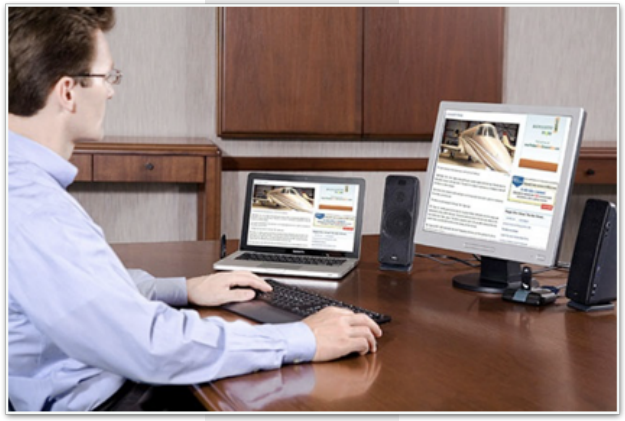
FACHHOCHSCHULE
ERFURT UNIVERSITY
OF APPLIED SCIENCES
Angewandte
Informatik



Nadav Savio
Design Sketch: The Context of Mobile Interaction
http://www.giantant.com/antenna/2007/06/design-sketch-the-context-of-m.html

Mobile vs. Desktop

- A mobile app is not a desktop app!
  - Different usage context
  - Different means of interaction
  - Different user goals
  - Different device capabilities
  - Different usage times
  - Different attention span



Android Ecosystem

Origin

- 2003: Development starts at Android, Inc.
- 2005: Google acquires Android, Inc.[1]
- 2007: Beta release
- 2008: Version 1.0 release (Sep 23th)[2]

- 2017: Andy Rubin (Android, Inc. founder) starts new foray into phone business with Essential[3]

[1] https://www.cnet.com/news/google-buys-android/

[2] https://en.wikipedia.org/wiki/Android_version_history

[3] https://tinyurl.com/gqv6ufb

Alpha Version Devices

FACHHOCHSCHULE ERFURT UNIVERSITY OF APPLIED SCIENCES
Angewandte Informatik

https://arstechnica.com/gadgets/2016/10/building-android-a-40000-word-history-of-googles-mobile-os/

What happened

https://everystevejobsvideo.com/original-iphone-introduction-macworld-sf-2007/

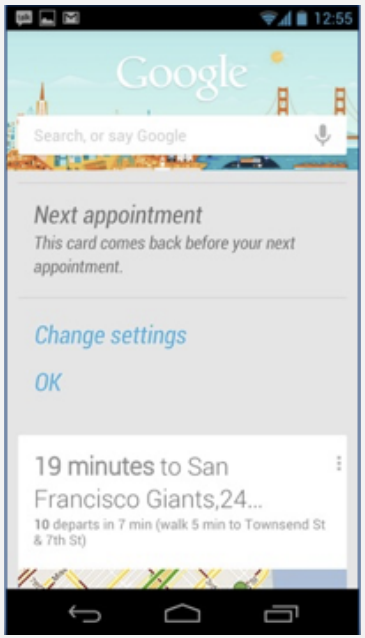
First Device: HTC Dream

FACHHOCHSCHULE
ERFURT UNIVERSITY
OF APPLIED SCIENCES
Angewandte
Informatik

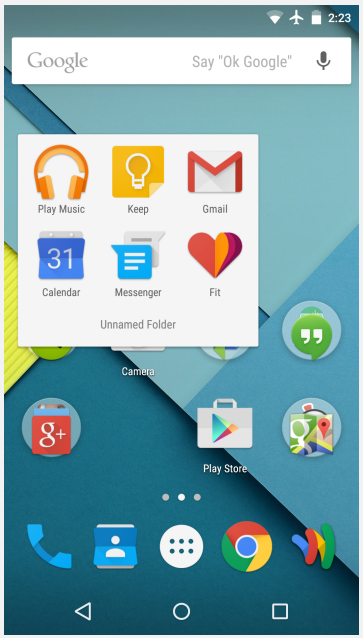
https://tinyurl.com/ydzb25o6

Impressions

Android 1.0



Android 4.1

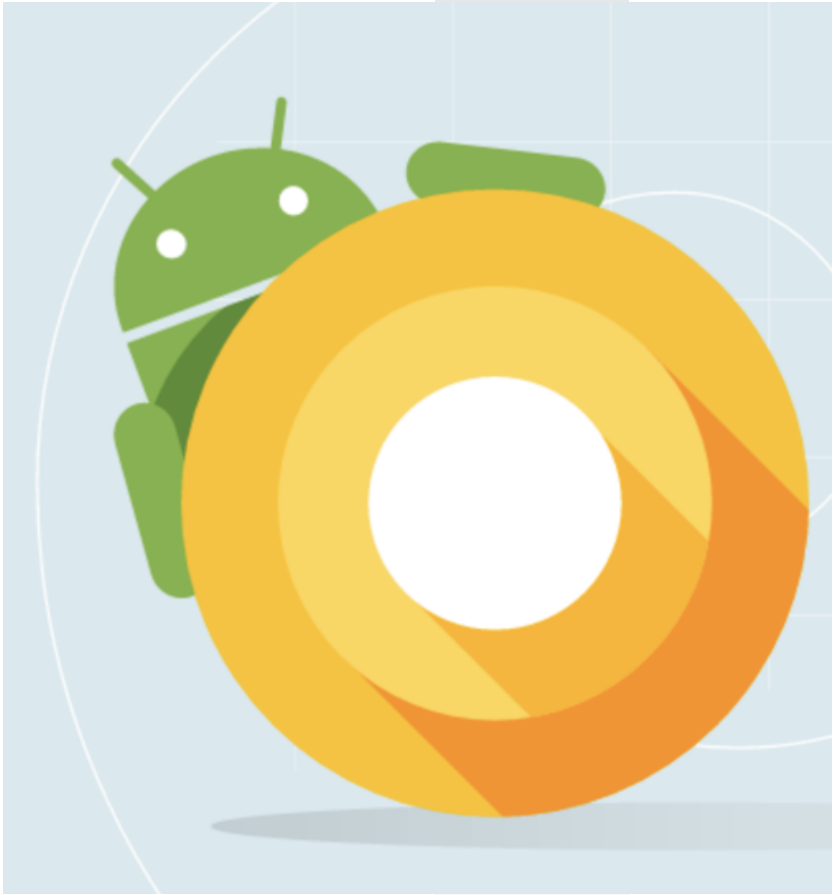


Android 5.0

Latest Version: 8.0 Oreo

- August 2017
- Distribution: 0.2%

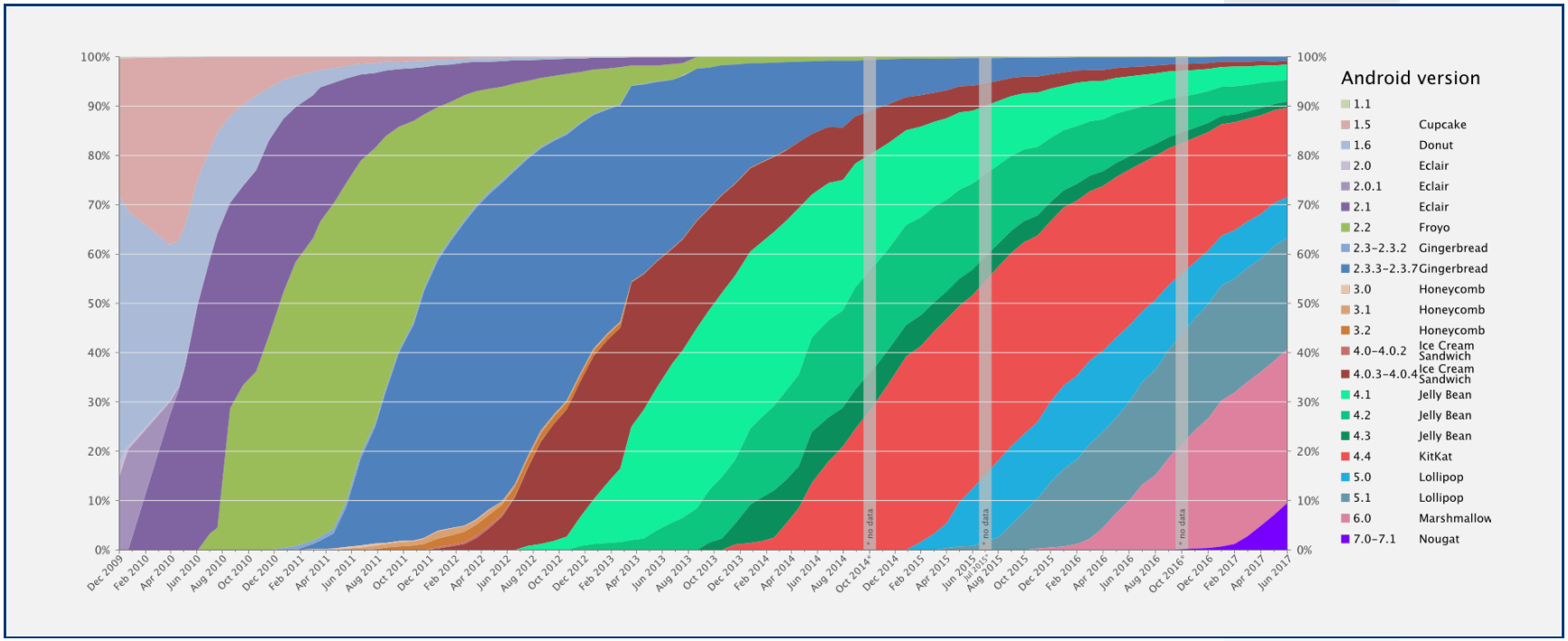
- Improved Notifications
- Picture-in-Picture
- Java 8 APIs
- Adaptive Icons
- Multidisplay Support



https://developer.android.com/about/dashboards/index.html

https://developer.android.com/studio/write/java8-support.html

Version Distribution



Source: https://en.wikipedia.org/wiki/Android_version_history

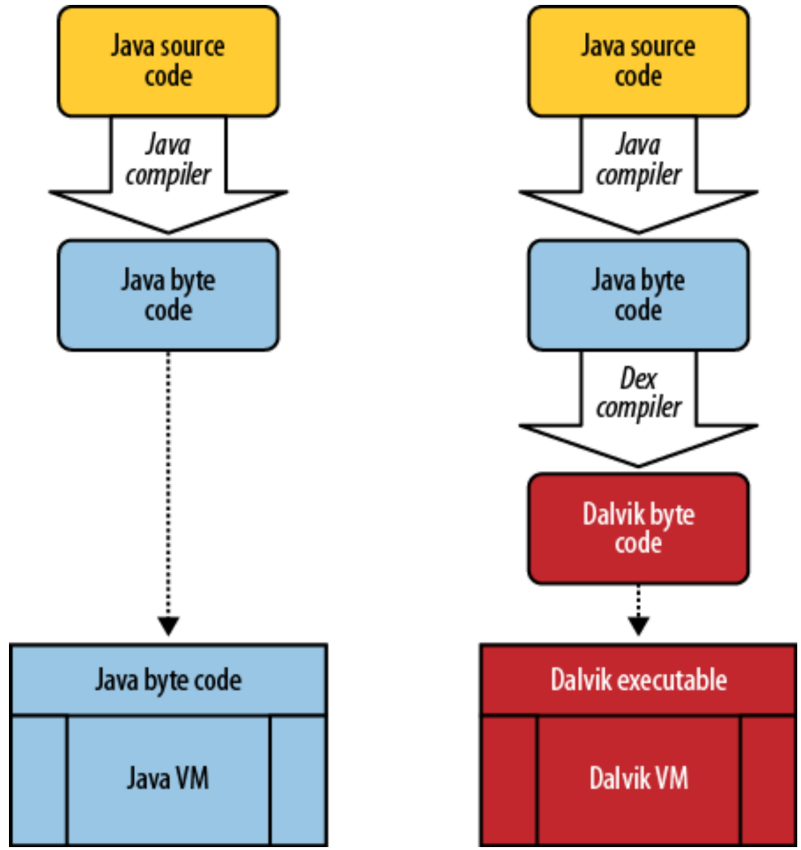
Android Basics

Java Runtime

- Apps usually written in Java
- Code and resources packed into *APK* file
- Dalvik = Android Java VM up to Android 4.x

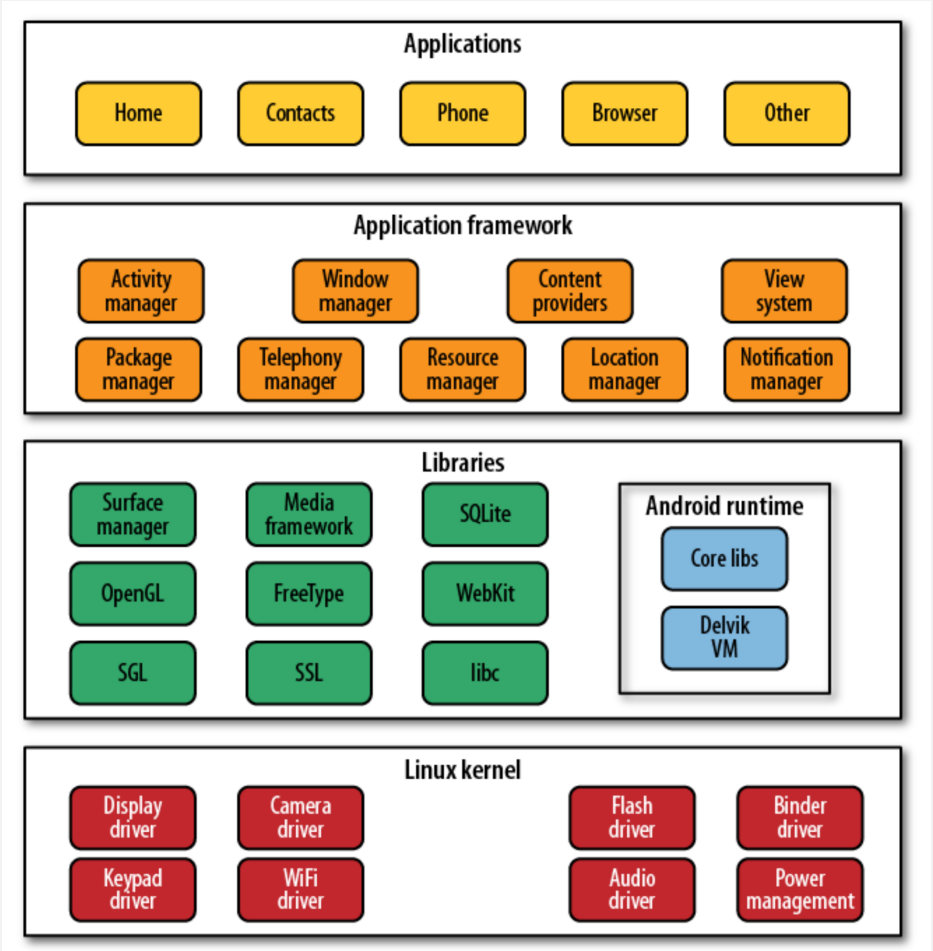
  http://code.google.com/p/dalvik
- ART Runtime since Android 5

  https://source.android.com/devices/tech/dalvik



Source: *Learning Android*, Marko Gargenta, O'Reilly 2011

System Stack



Source: *Learning Android*, Marko Gargenta, O'Reilly 2011 / https://developer.android.com/guide/platform

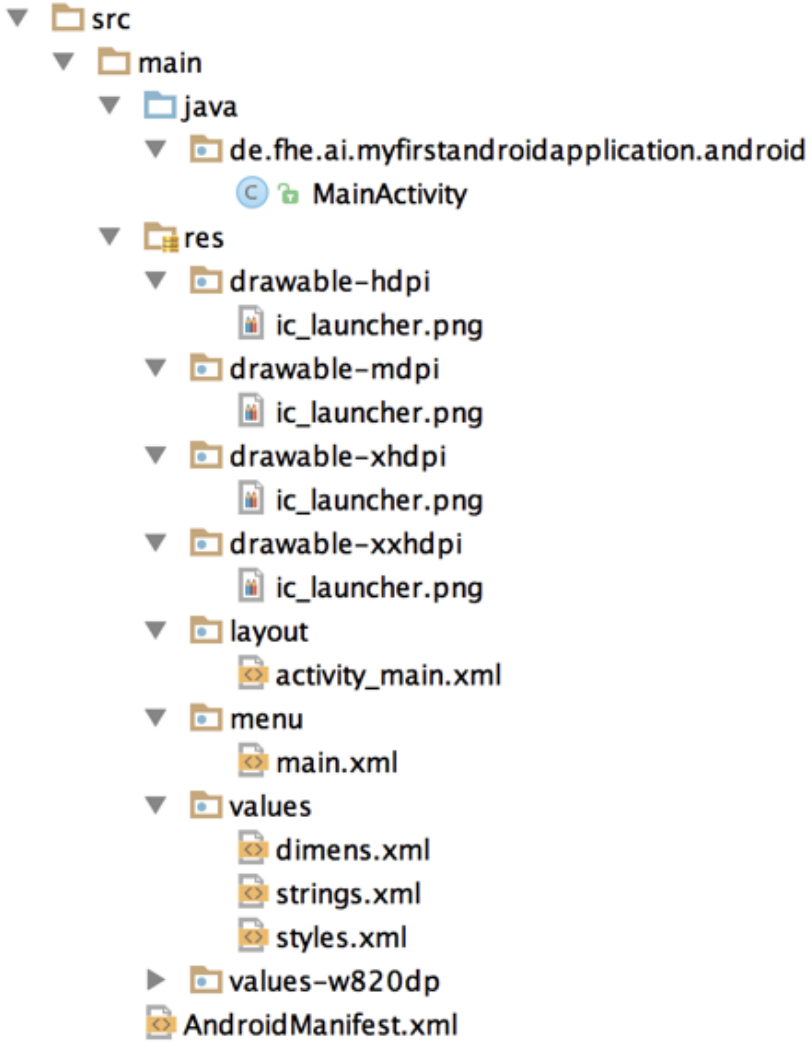
Android Dev feels like...

Android Dev feels like...

XML is used for

- App Descriptor, e.g. AndroidManifest.xml
- UI structure
- Style definitions
- SVG-like graphics
- Static data
- Localized strings
- Intent filters
- ...

Project Structure



- Typical Maven/Gradle structure
- *res* contains resources like images or xml files
- Folder suffixes for device/system specific classification
- Runtime will select resources accordingly
- Think twice before providing to many resources

Referencing Resources II


```
1  public final class R {
2
3    public static final class color {
4      public static final int dark=...;
5      public static final int blue=...;
6    }
7
8    public static final class id {
9      public static final int bar=...;
10     public static final int text=...;
11   }
12
13   public static final class layout {
14     public static final int grid=...;
15     public static final int item=...;
16   }
17
18 }
```

- Android Studio maintains an *R* class for your project
- That class contains inner classes for different types of resources
- Those inner classes contain constants that map to file names or IDs

Referencing Resources II


```
1  public class MyActivity extends Activity {
2
3      public void onCreate( Bundle savedInstanceState ) {
4          super.onCreate( savedInstanceState );
5
6          // Set UI by referencing grid.xml from layout folder
7          setContentView( R.layout.grid );
8
9          // Find two widgets using their IDs
10         GridView mGridView = (GridView) findViewById( R.id.grid );
11         TextView mTextView = (TextView) findViewById( R.id.text );
12     }
13
14 }
```

App Components

Activity

- Main component
- Responsible for UI/UX
- May use *Fragments*

Service

- Background work
- Don’t define a UI
- May run indefinitely

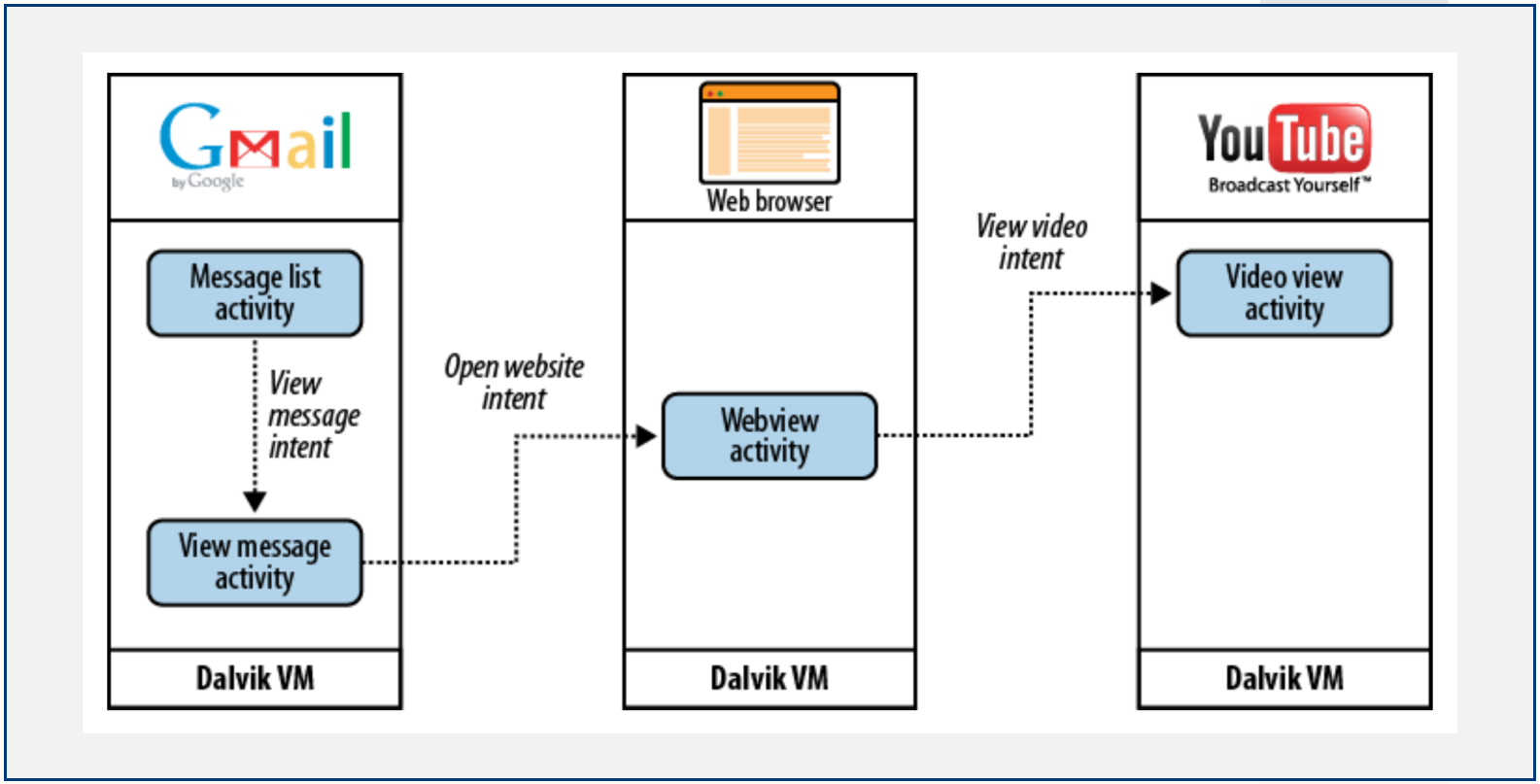
Broadcast Receiver

- Listen to events
- Act accordingly
- No specific order

Content Provider

- Access to local data
- Can be used by all installed apps
- Usually backed by a database

Component Interaction



Source: *Learning Android*, Marko Gargenta, O'Reilly 2011

Intents

Explicit

- Targets a specific component
- Start activity or stop a service
- May contain additional key-value data
- Handled by the system

Implicit

- No specific target
- Describe what is needed, e.g. open a webpage
- System determines matching components
- If more than one component matches, the user decides
- May also contain key-value data

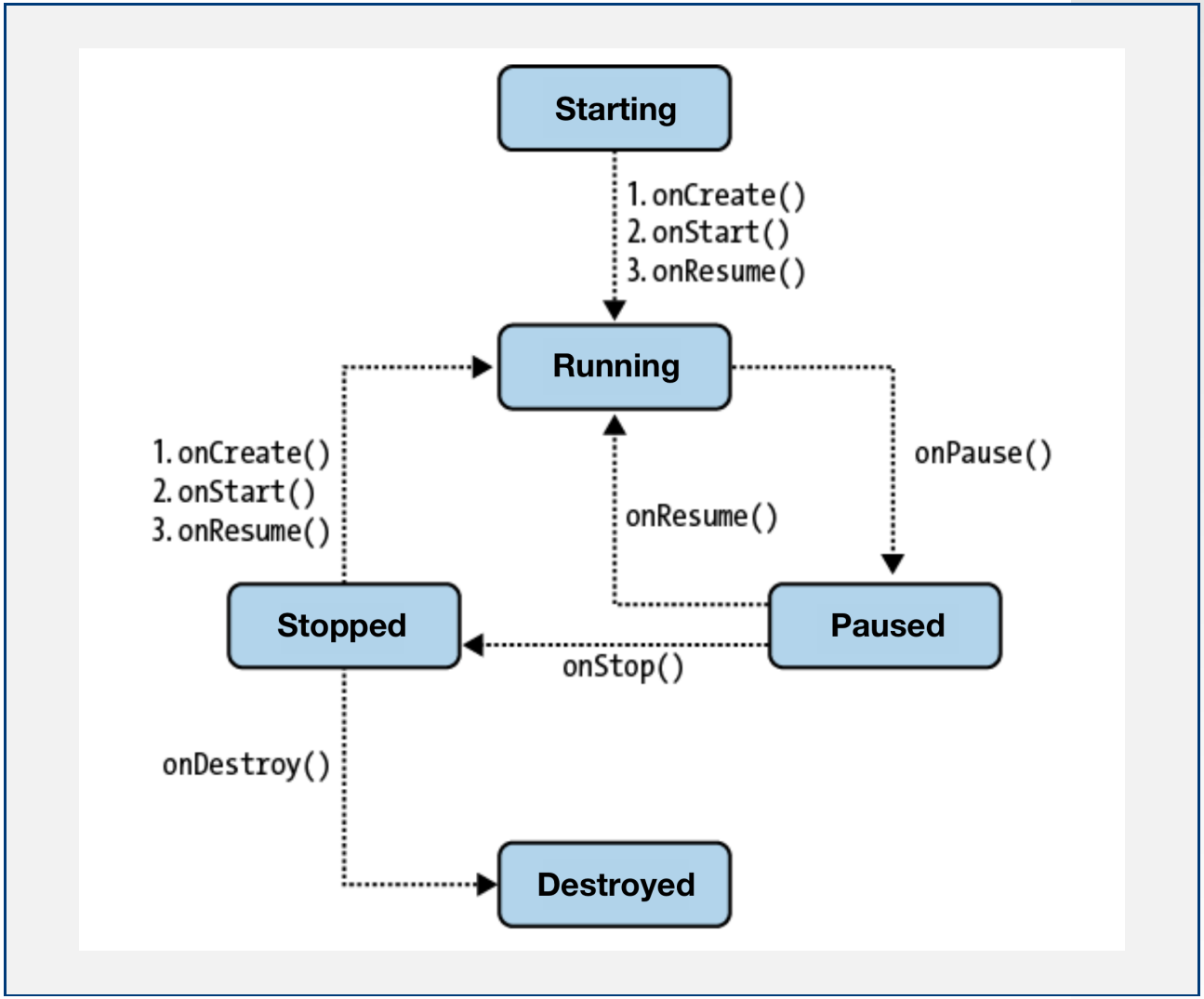
Intent Examples


```
1  // Explicit Intent
2  Intent i = new Intent( this, OtherActivity.class );
3  i.putExtra( "value", "Hello World!" );
4
5  this.startActivity( i );
6
7
8  // Implicit Intent
9  Intent i = new Intent( Intent.ACTION_VIEW );
10 i.setData( Uri.parse( "geo:0,0?z=4&q=fh+erfurt" ) );
11
12 this.startActivity( i );
13
14
15 // /this/ references instance of /Context/ like Activity or Service
```

AndroidManifest.xml

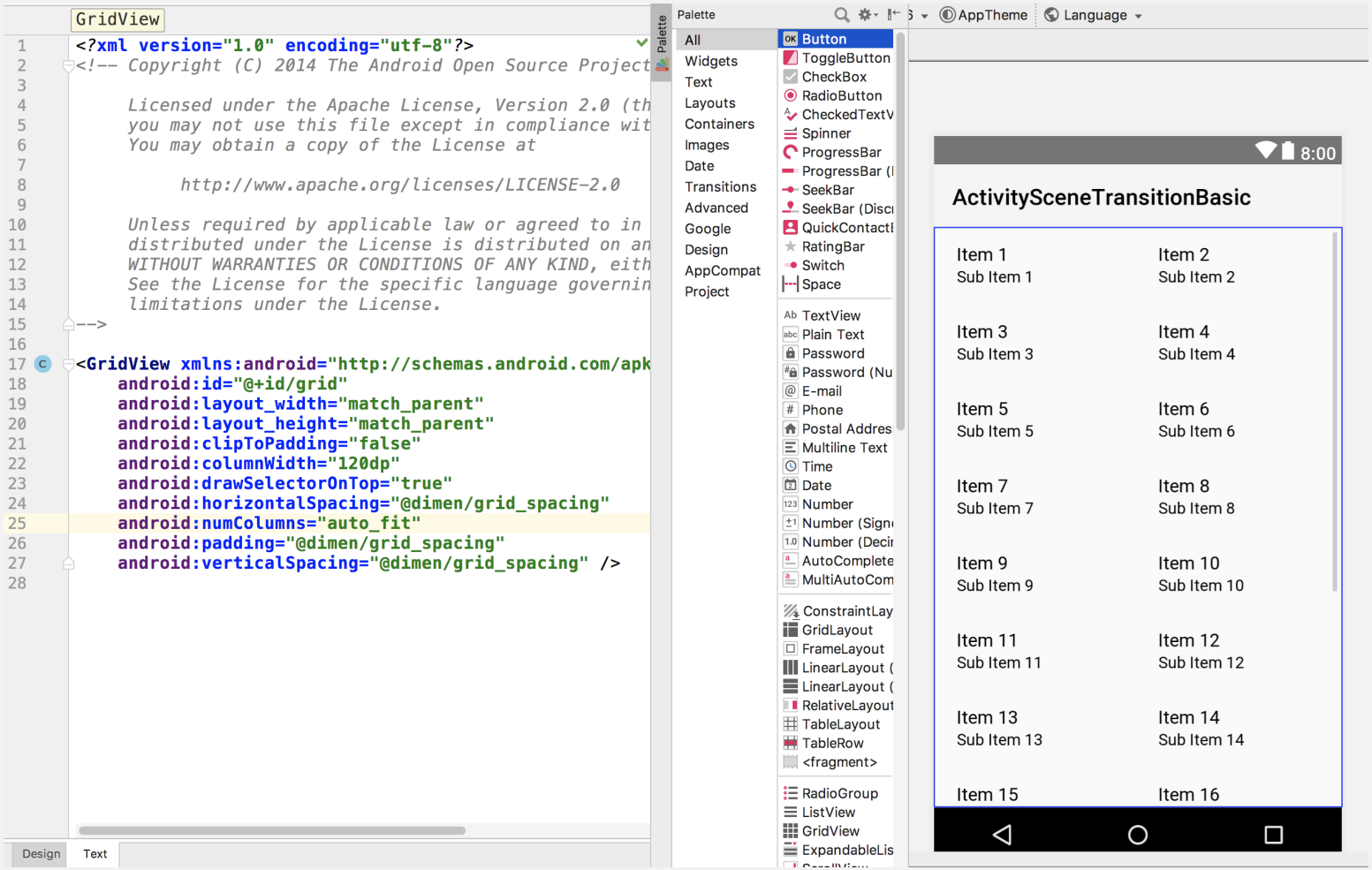

```
1  <manifest ... >
2    <application android:label="@string/app_name"
3      android:icon="@drawable/ic_launcher"
4      android:theme="@style/AppTheme">
5
6      <activity android:name=".MainActivity"
7        android:label="@string/app_name">
8        <intent-filter>
9          <action android:name="android.intent.action.MAIN" />
10         <category android:name="android.intent.category.LAUNCHER" />
11       </intent-filter>
12     </activity>
13
14     <activity android:name=".DetailActivity" />
15
16   </application>
17
18   <uses-permission android:name="android.permission.INTERNET" />
19 </manifest>
```

Activity Lifecycle

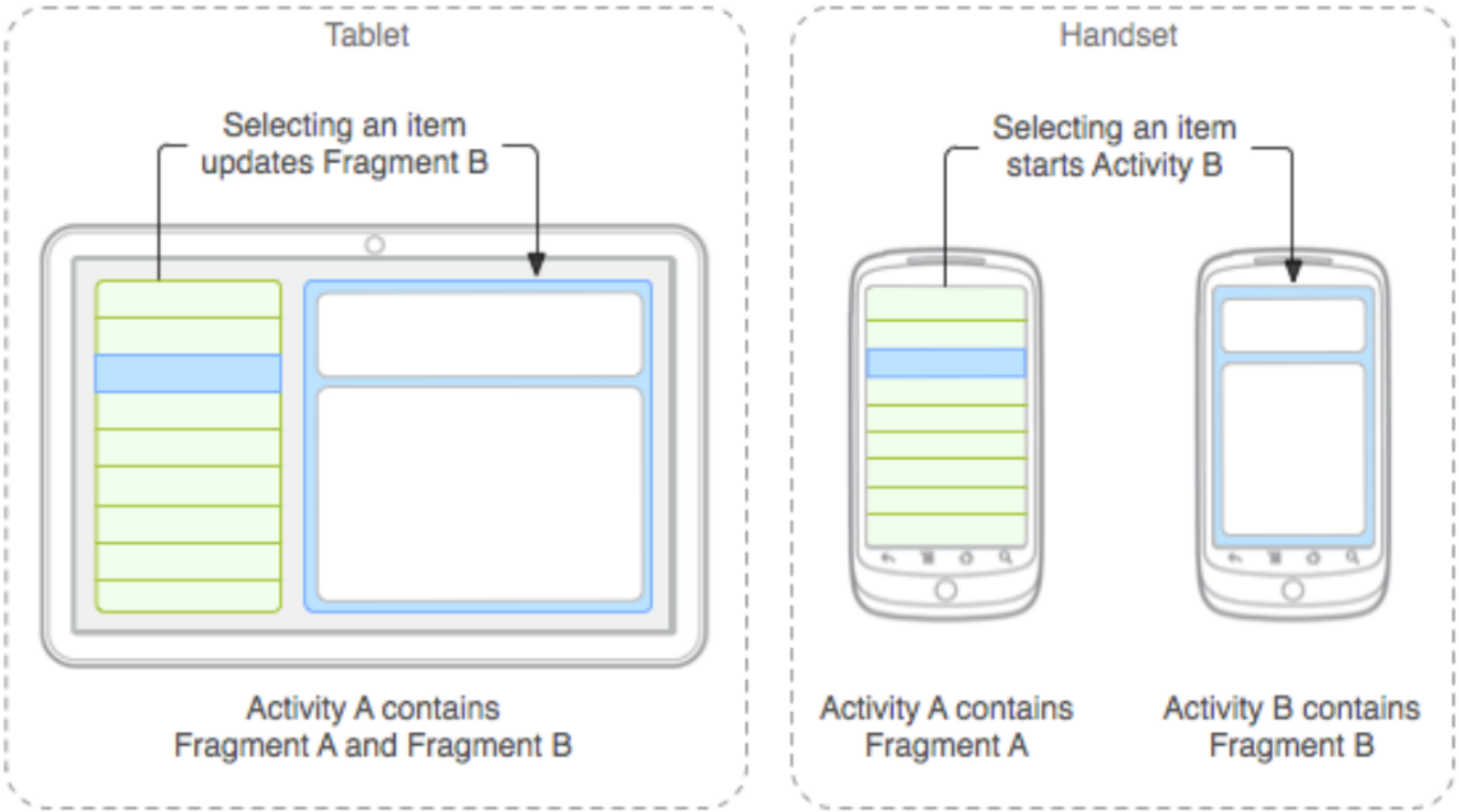


Source: *Learning Android*, Marko Gargenta, O'Reilly 2011

UI Designer

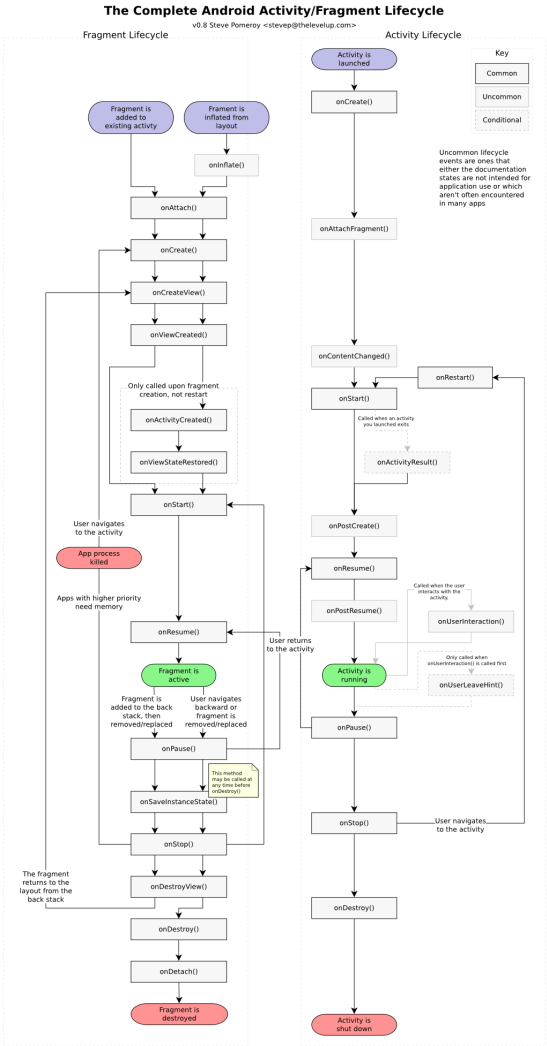

GridView
<?xml version="1.0" encoding="utf-8"?>
<!-- Copyright (C) 2014 The Android Open Source Project
Licensed under the Apache License, Version 2.0 (th
you may not use this file except in compliance wit
You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in
distributed under the License is distributed on an
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, eith
See the License for the specific language governin
limitations under the License.
-->
<GridView xmlns:android="http://schemas.android.com/apk
android:id="@+id/grid"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:clipToPadding="false"
android:columnWidth="120dp"
android:drawSelectorOnTop="true"
android:horizontalSpacing="@dimen/grid_spacing"
android:numColumns="auto_fit"
android:padding="@dimen/grid_spacing"
android:verticalSpacing="@dimen/grid_spacing" />
Design
Text
Palette
All
Widgets
Text
Layouts
Containers
Images
Date
Transitions
Advanced
Google
Design
AppCompat
Project
Button
ToggleButton
CheckBox
RadioButton
Spinner
SeekBar
Switch
Space
TextView
Plain Text
Password
E-mail
Phone
Postal Addres
Multiline Text
Time
Date
Number
AutoComplete
GridLayout
FrameLayout
TableLayout
TableRow
RadioGroup
ListView
GridView
AppTheme
Language
8:00
ActivitySceneTransitionBasic
Item 1
Sub Item 1
Item 2
Sub Item 2
Item 3
Sub Item 3
Item 4
Sub Item 4
Item 5
Sub Item 5
Item 6
Sub Item 6
Item 7
Sub Item 7
Item 8
Sub Item 8
Item 9
Sub Item 9
Item 10
Sub Item 10
Item 11
Sub Item 11
Item 12
Sub Item 12
Item 13
Sub Item 13
Item 14
Sub Item 14
Item 15
Item 16

Fragments



Source: https://developer.android.com/guide/components/fragments.html

Fragment Lifecycle



Source: https://github.com/xxv/android-lifecycle

Context

- Interface to global information about an app
- Access to app-specific resources and classes
- Calls to operations like launching activities, broadcasting and receiving intents
- Provided by the Android system
- Most components implement that interface (e.g. Activities or Services)
- However, don’t simply reference those components when in need of a Context instance -> DI!

Support Libraries

- Many features which are only available to newer Android versions have been back-ported to older versions via *Support Libraries*
- Prominent examples include: Fragments, Action-/AppBar, Material Design, ViewPager, RecyclerView, SwipeToRefresh, DrawerLayout

- https://developer.android.com/topic/libraries/support-library

Google Play Services

- A lot of new Google-related features are not part of the *Android Open Source Project* (AOSP)
- Instead, they have been added to *Google (Play) Services*
- Standalone APK that gets updated silently
- Examples: In-App Billing, Access to Google Services like Drive, Analytics or Ads, Wear Communication
- Seem to move to Firebase

- https://developers.google.com/android/guides/overview

App Architecture

- Up to Android 8, not that much help from Google
- Lot of bad apps out there with massive, unstable Activities, broken network code, bad UX, performance issues, ...
- You should definitely choose an architecture like MVC, MVVM or MVP - see https://github.com/googlesamples/android-architecture

- Since Android 8, *Android Architecture Components* are in beta - see https://developer.android.com/topic/libraries/architecture

What else?

- Android Wear, Android Auto, Android TV
- Android Things
- NDK
- Product Flavors
- Expansion Files for Play Store
- Other App Stores
- ARCore
- Instant Apps
- Kotlin

Links And References

Third Party Libs

- Retrofit (Networking)[4]
- Dagger 2 (Dependency Injection)[5]
- EventBus (Message-based In-App Communication)[6]
- RxAndroid (Reactive)[7]
- Glide (Image Handling)[8]
- Firebase (Push, Storage, Authentication, Analytics, ...)[9]

[4] http://square.github.io/retrofit/

[5] https://google.github.io/dagger/

[6] http://greenrobot.org/eventbus/

[7] https://github.com/ReactiveX/RxAndroid

[8] http://bumptech.github.io/glide/

[9] https://firebase.google.com

Docs, Tutorials & Co.

- Google Dev Docs[10]
- Futurice Best Practices[11]
- Material Design Guidelines[12]
- Kotlin and Android[13]
- Meet Android Studio[14]
- Configure Your Build[15]
- Navigation - Back vs. Up[16]

[10] https://developer.android.com

[11] https://github.com/futurice/android-best-practices

[12] https://developer.android.com/design/material/index.html

[13] https://kotlinlang.org/docs/tutorials/kotlin-android.html

[14] https://developer.android.com/studio/intro/index.html

[15] https://developer.android.com/studio/build/index.html

[16] https://developer.android.com/design/patterns/navigation.html

Hackaton @ FH Erfurt

FACHHOCHSCHULE ERFURT UNIVERSITY OF APPLIED SCIENCES
Angewandte Informatik

- 15-17 Dec 2017
- By Fachschaft AI
- For students and practitioners
- Advertising & registration will start in a couple of days