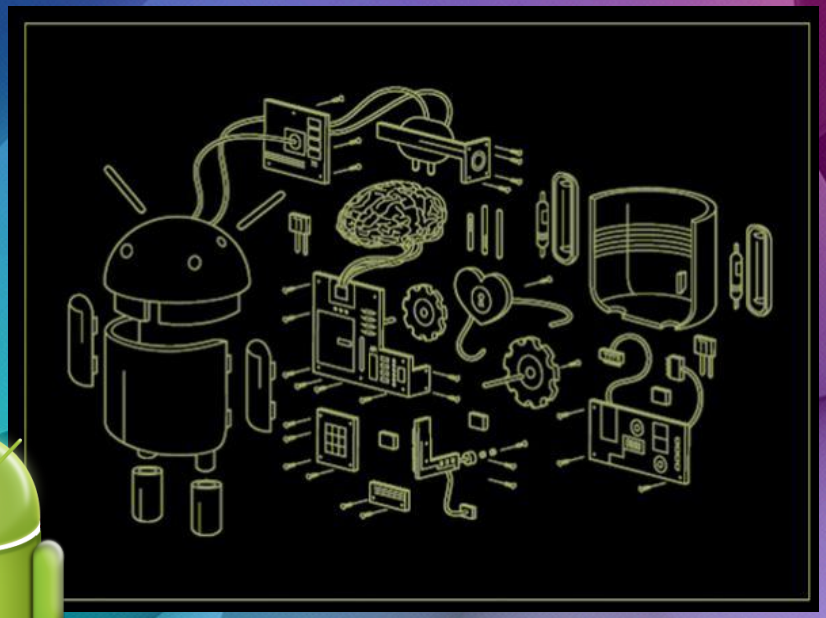
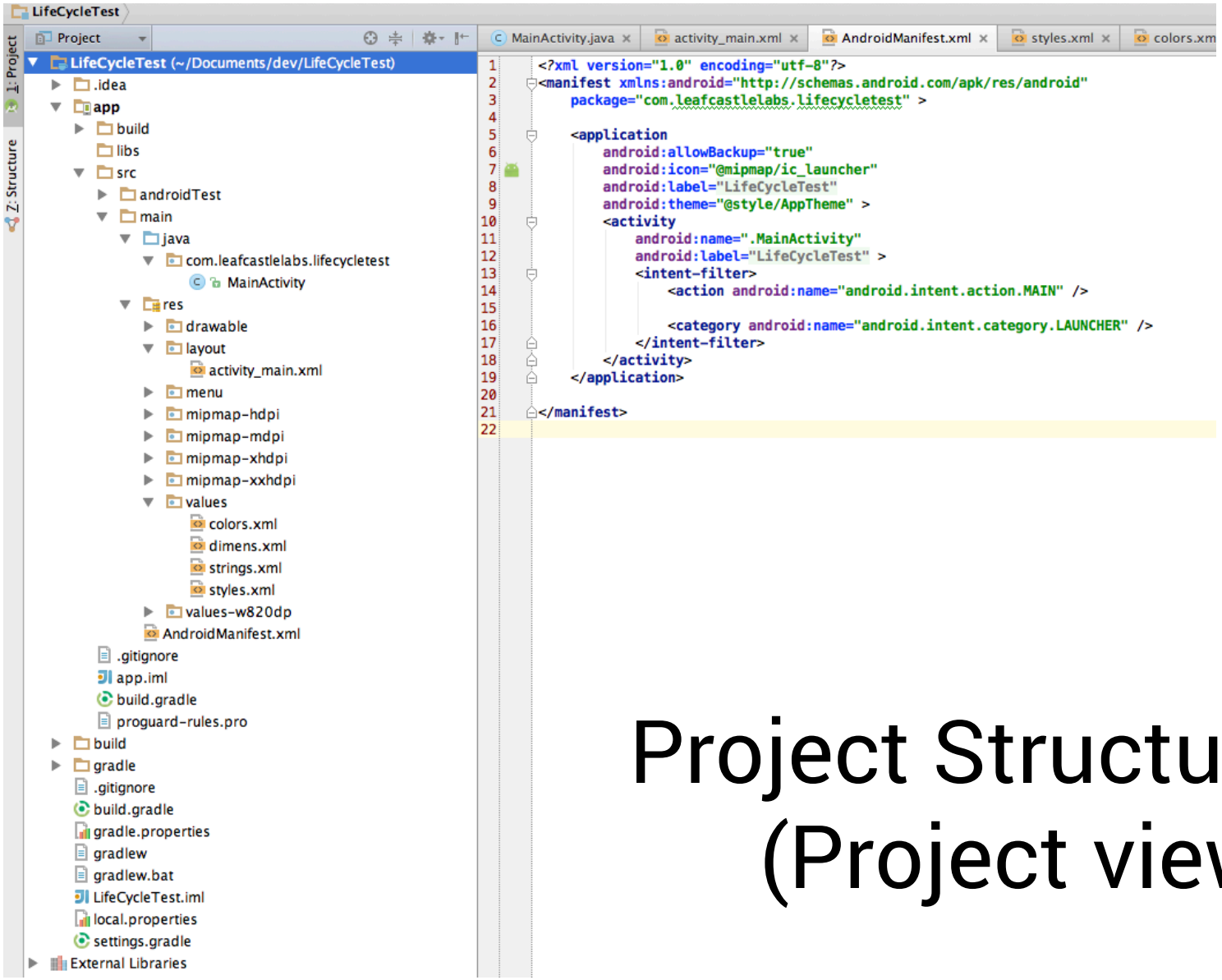
THE ANATOMY OF AN APP



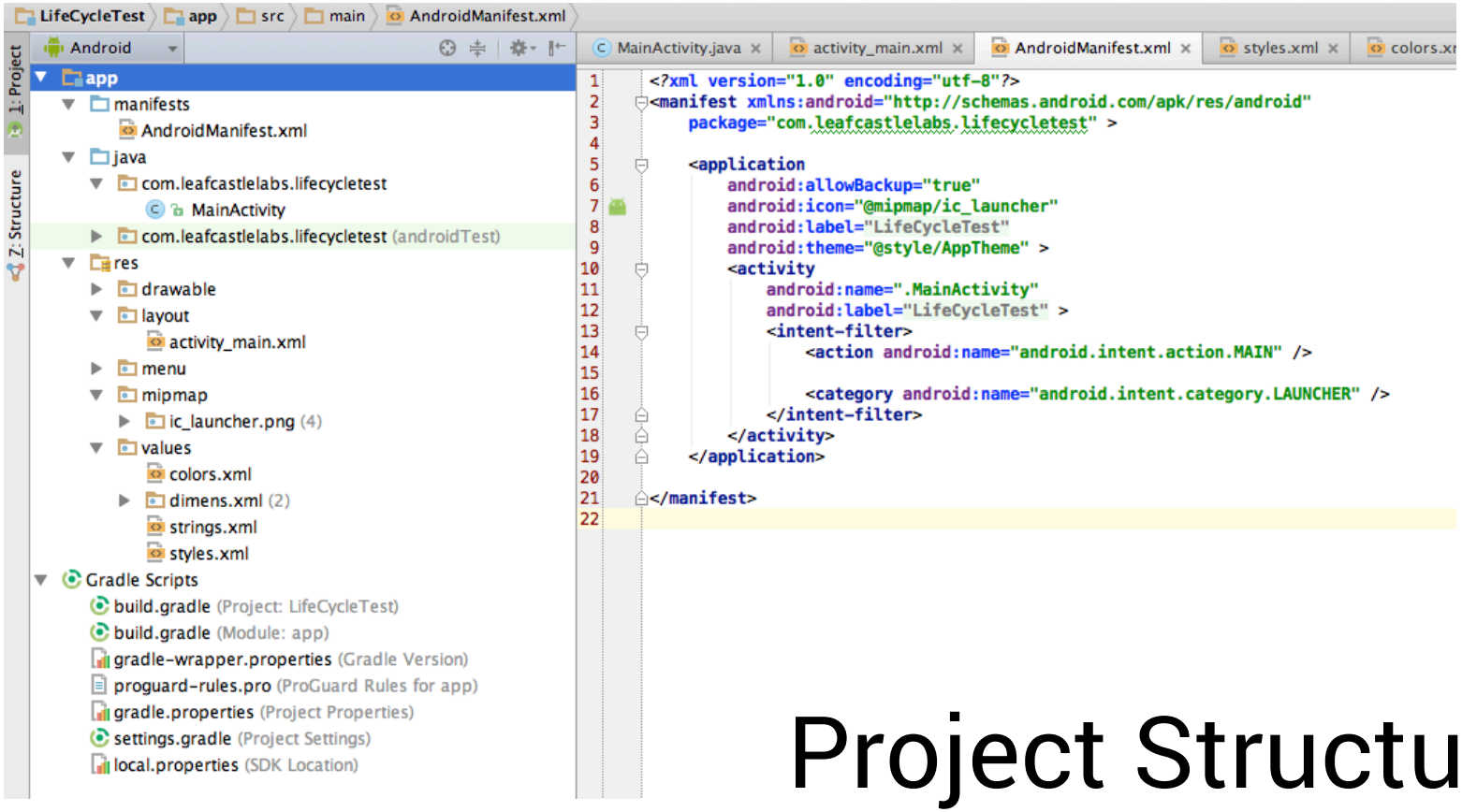
Outline

- Project structure
- Manifest
- Build system
- Core components
- Application
- Activity
- Development Flow

PROJECT STRUCTURE



Project Structure (Project view)



Project Structure (Android view)

Modules

- **Android Application Modules**
- Test Modules
- Library Modules
- App Engine Modules (Cloud backend)

App Module Files

```
app/
  | build/
  | libs/
  | src/
  |   | androidTest/
  |   | main/
  |   |   | java/
  |   |   |    <package name>/
  |   |   |       … code files …
  |   |   | res/
  |   |   |    … resources files and xml …
  |   |   | manifest.xml
  |   | build.gradle
```

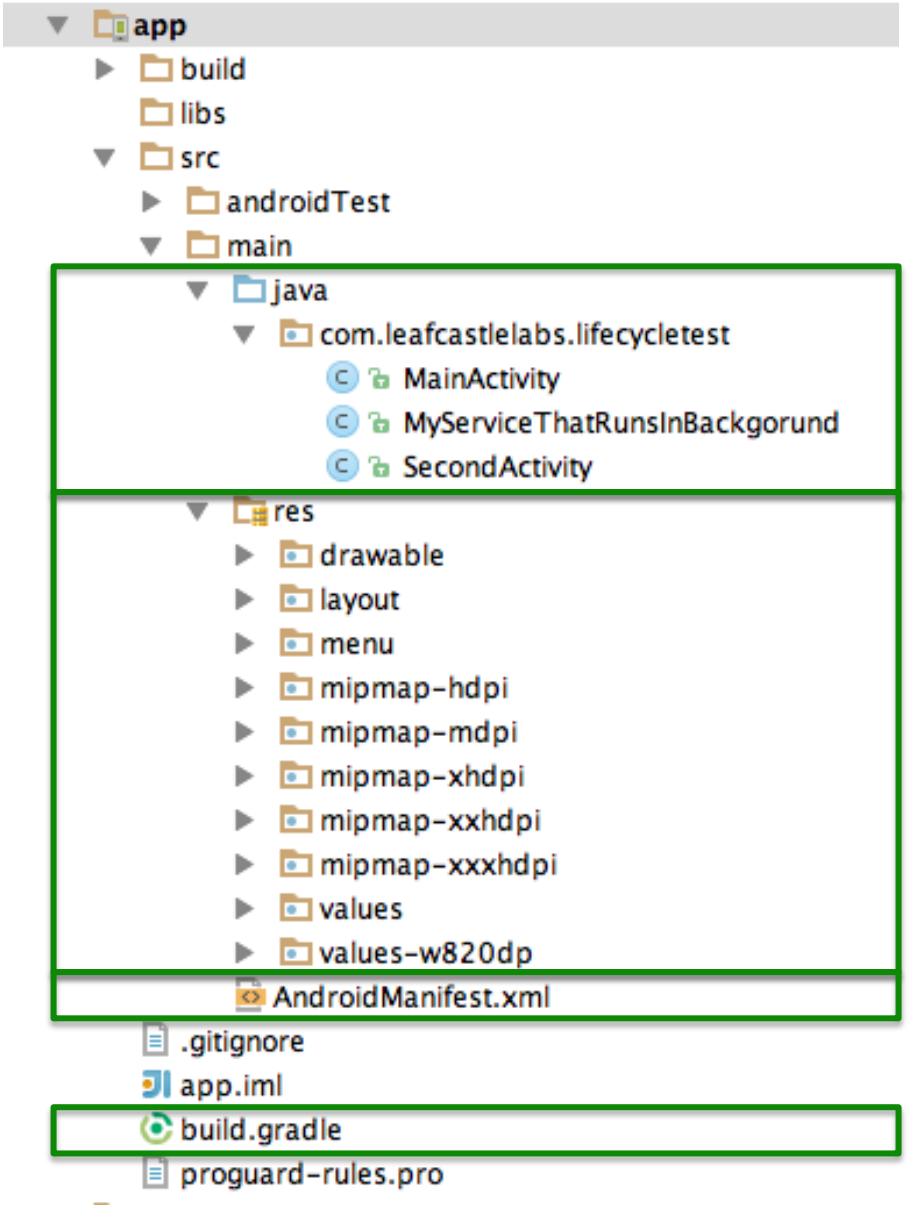
App Module Files

Code

Resources

Manifest

Gradle




THE MANIFEST

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.leafcastlelabs.lifecycletest" >

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name=".SecondActivity"
            android:label="@string/title_activity_second" >
        </activity>
        <service
            android:name=".MyServiceThatRunsInBackgorund"
            android:enabled="true"
            android:exported="true" >
        </service>
    </application>

</manifest>
```

This is the app name

Manifest.xml

Core Components

- Application There will only be one Application per. app
- Activity "Unlimited amounts" of these
- Intents and Intent-Filters Like Signals and Slots
- Service Can work in the background
- Content Provider
- Broadcast Receiver
- Widget

Manifest Elements

<action>
<activity>
<activity-alias>
<application>
<category>
<data>
<grant-uri-permission>
<instrumentation>
<intent-filter>
<manifest>
<meta-data>

<permission>
<permission-group>
<permission-tree>
<provider>
<receiver>
<service>
<supports-screens>
<uses-configuration>
<uses-feature>
<uses-library>
<uses-permission>
<uses-sdk>

http://developer.android.com/guide/topics/manifest/manifest-intro.html

THE BUILD SYSTEM

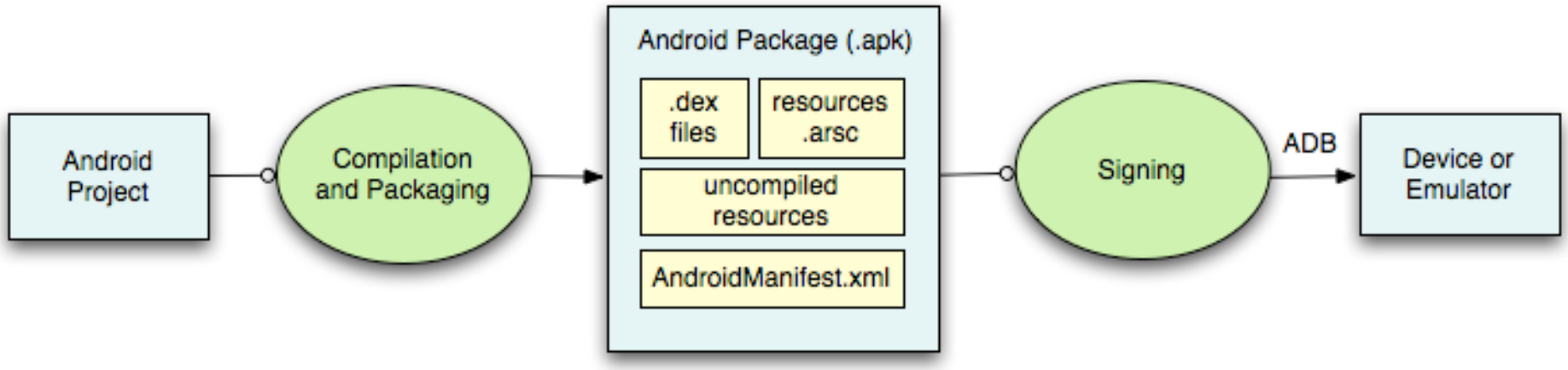
The app can be build with another compiler with the manifest file

```
1   apply plugin: 'com.android.application'
2
3   android {
4       compileSdkVersion 22
5       buildToolsVersion "22.0.1"
6
7       defaultConfig {
8           applicationId "com.leafcastlelabs.lifecycletest"
9           minSdkVersion 14
10          targetSdkVersion 22
11          versionCode 1
12          versionName "1.0"
13      }
14      buildTypes {
15          release {
16              minifyEnabled false
17              proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
18          }
19      }
20  }
21
22  dependencies {
23      compile fileTree(dir: 'libs', include: ['*.jar'])
24      compile 'com.android.support:appcompat-v7:22.0.0'
25  }
26
```

Package name -> Has to be Unique

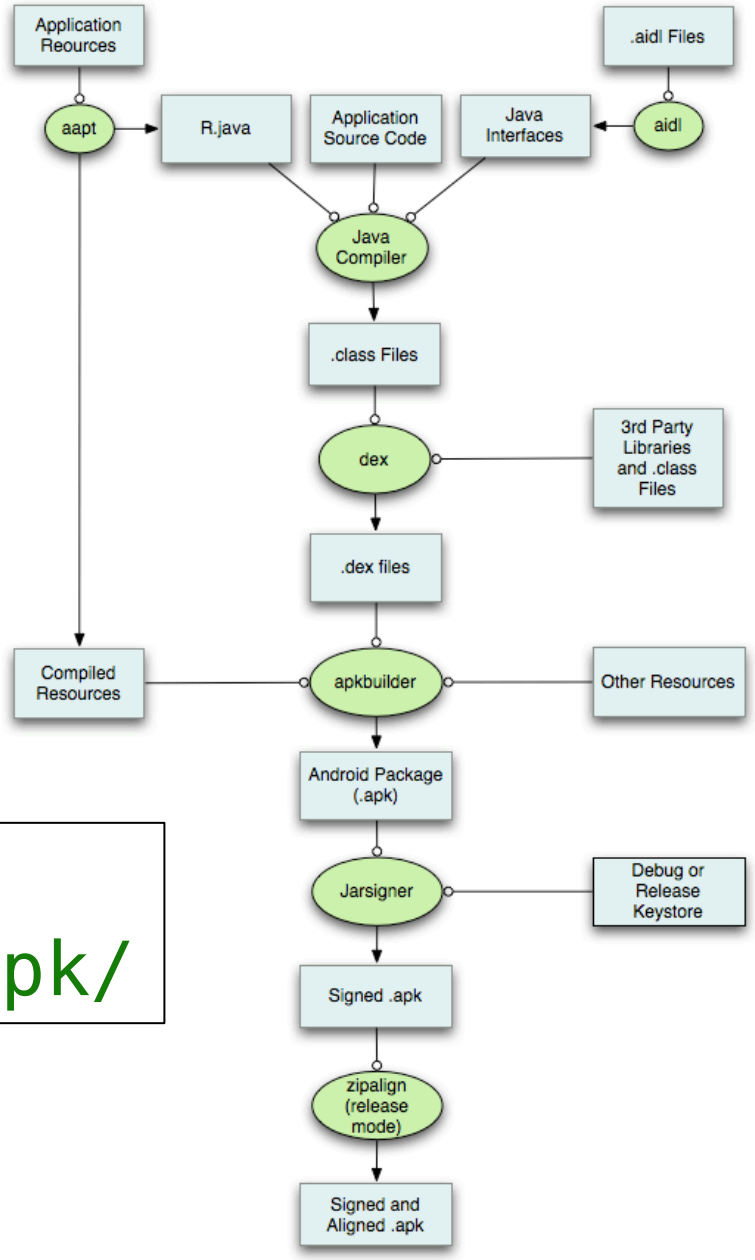
Here SDK versions can be edited -> Target should always be the newest version.

Building Apps



https://developer.android.com/tools/building/

Build System under the Hood



APK files output:
`app/build/outputs/apk/`

Source: http://developer.android.com/

Resources in `res/`

- Layouts
- Simple values e.g. strings, string arrays, dimensions
- App icon
- Images
- Menus
- Colors
- Animations
- Arbitrary raw assets e.g. media files
- Misc. XML

```
layout/
values/

mipmap/
drawable/
menu/
color/
anim/
raw/

xml/
```

App Icon

- Support various dpi levels
- New standard is: **mipmap**
- Can be generated by e.g. **Android Asset Studio**

http://romannurik.github.io/AndroidAssetStudio/icons-launcher.html

ProGuard

“**shrinks**, **optimizes**, and **obfuscates** your code by **removing unused code** and **renaming** classes, fields, and methods with semantically obscure names.

The result is a **smaller sized .apk file** that is **more difficult to reverse engineer**”

http://developer.android.com/tools/help/proguard.html

Project and Manifest

ACTIVITIES

Tasks and Activities

- Task

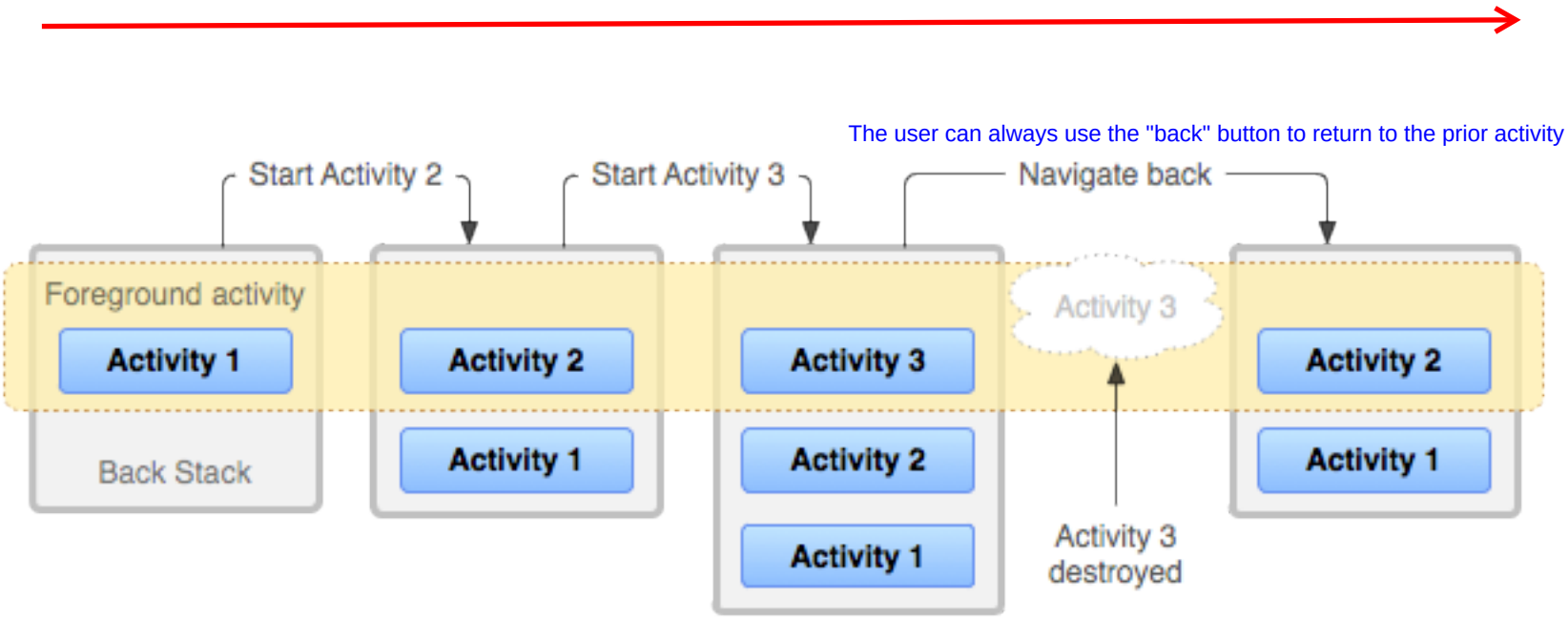
  “A task is a **collection of activities** that users interact with when **performing a certain job**”

- Activity

  “Each activity should be designed around a **specific kind of action** the user can perform and can start other activities”

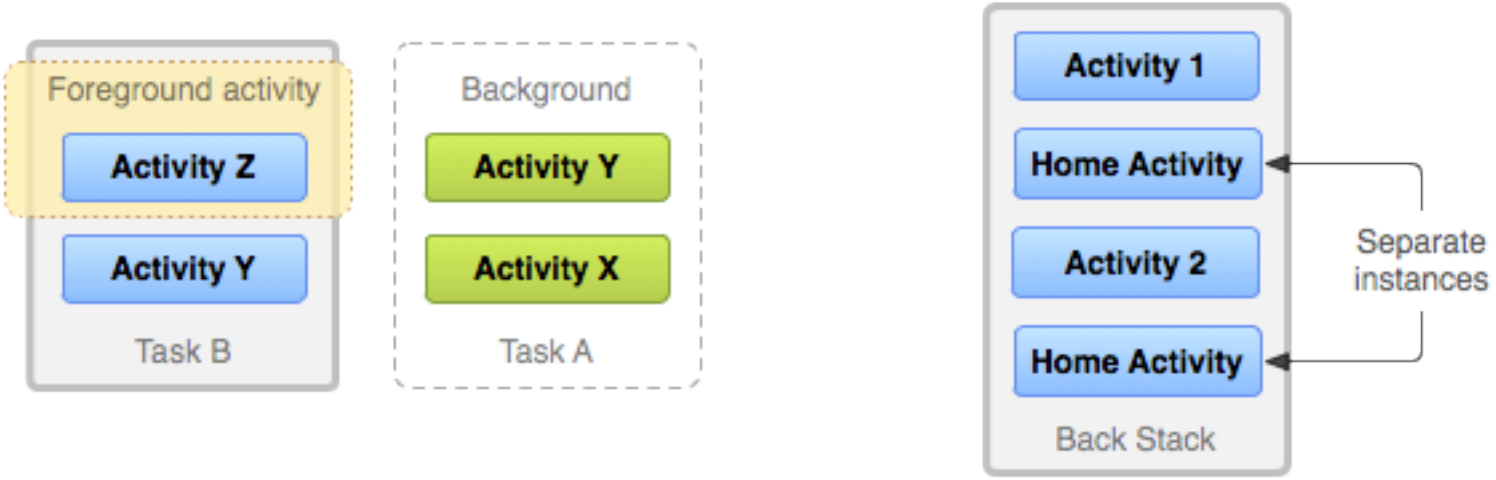
http://developer.android.com/guide/components/tasks-and-back-stack.html

Back Stack



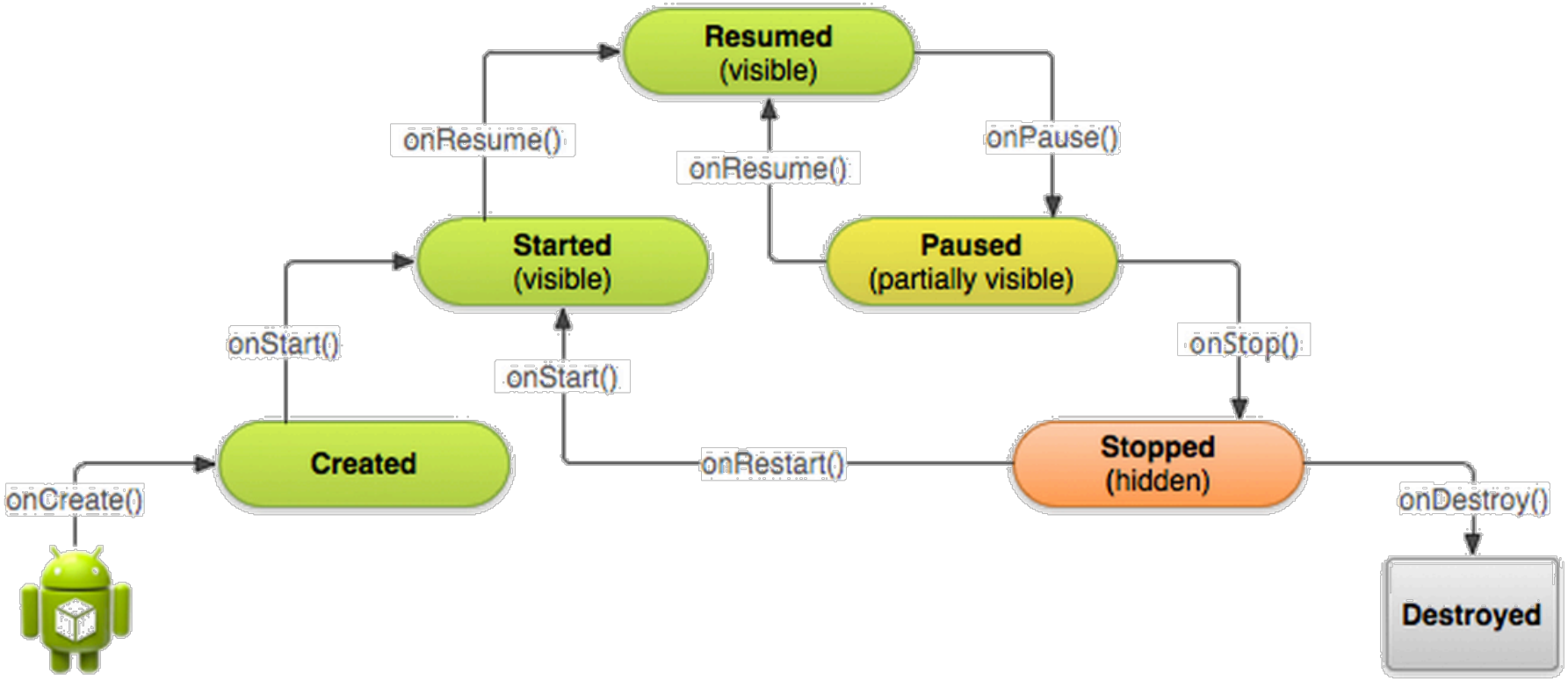
http://developer.android.com/guide/components/tasks-and-back-stack.html

Multitasking vs Multi-instance



http://developer.android.com/guide/components/tasks-and-back-stack.html

Activity Life Cycle Diagram

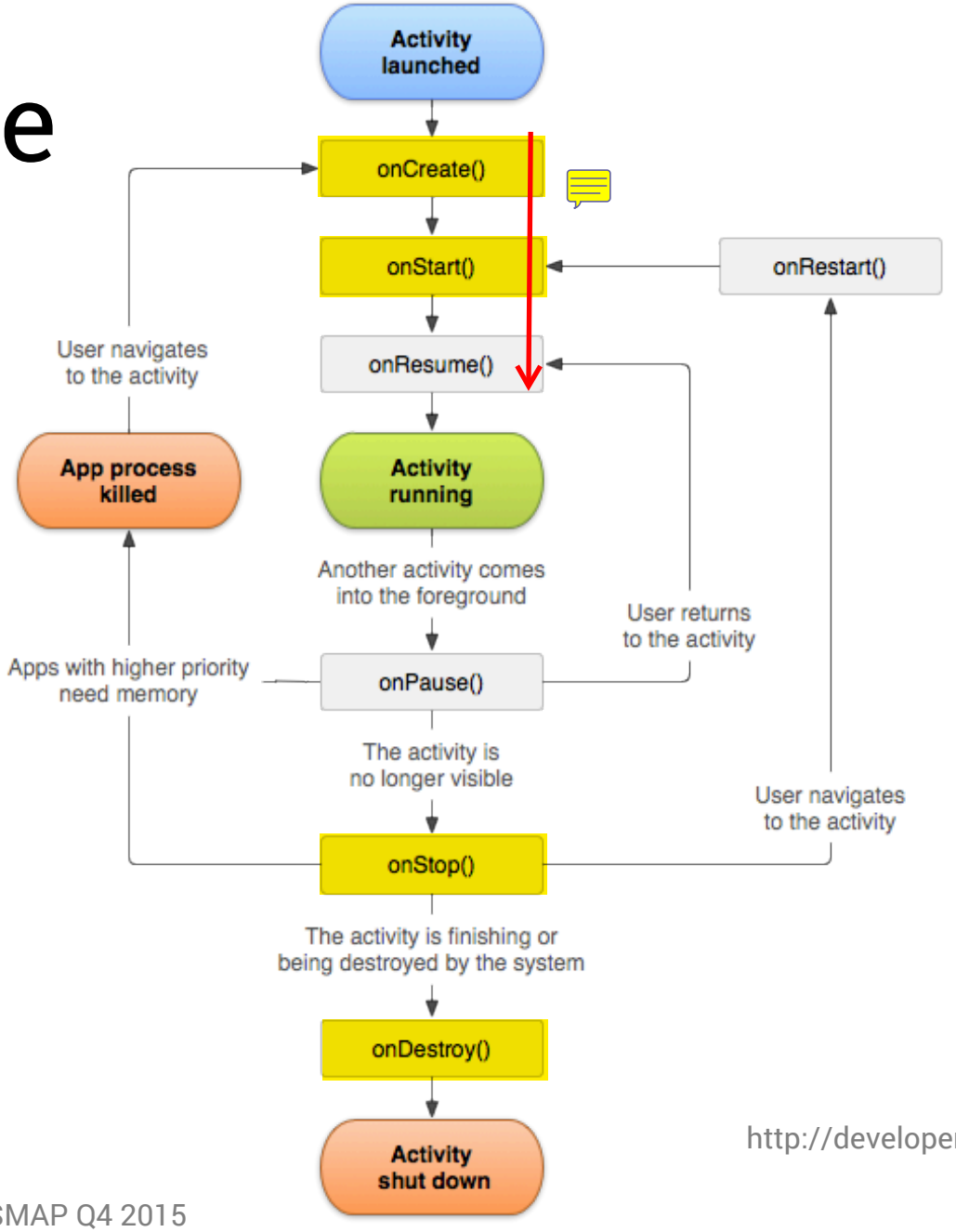


http://developer.android.com/

“Paranoid Android”

- Android is in charge!
- Once you activity is not in the foreground it can potentially be shutdown e.g. to free system memory
- Android can also intervene if your app is too slow on the main thread / UI

Life Cycle



http://developer.android.com/

Activity Life Cycle Methods

- `@Override` these methods to handle the main state changes
  - `onCreate()`
  - `onDestroy()`
  - `onStart()`
  - `onStop()`
  - `onPause()`
  - `onResume()`

savedInstanceState

- Activities used a `Bundle` to persist information about the instance state
- Parsed to the Activity on creation as the Bundle savedInstanceState parameter
- Android does a lot of saving automatically, but ultimately it is YOUR responsibility as a programmer
  - Think of the user experience!


APPLICATION

`Application`

- An app always has a “root level” singleton from the base class `Application`
- Activities can share global data and state by extending this, but in most cases it is more clean to use intents and/or static singletons

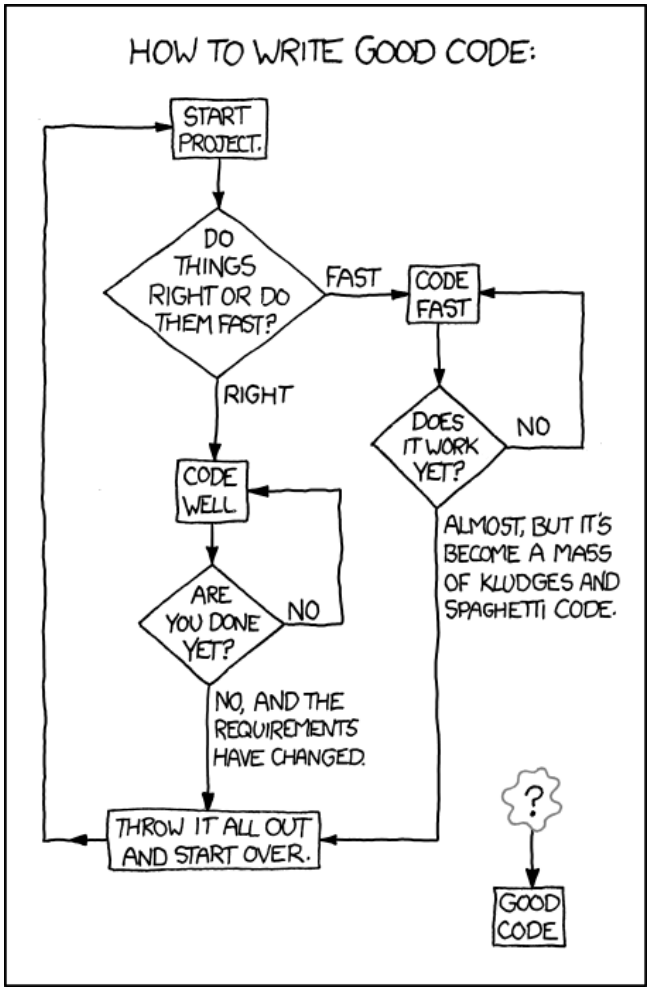
Context

- “**Interface to global information about an application environment**. This is an abstract class whose implementation is provided by the Android system. It **allows access to application-specific resources and classes, as well as up-calls for application-level operations** such as launching activities, broadcasting and receiving intents, etc.“

http://developer.android.com/reference/android/content/Context.html

DEVELOPMENT FLOW

Android Development Flow



https://xkcd.com/844/

Pro Tip

- Beware of the tools
  - Features like code completion and project wizards are awesome for general productivity
  - But sometimes they “outsmart” you and do things without you noticing
    - Importing the wrong library (version)
    - Using unexpected derived classes
  - Can be really hard to track down!
- The code will never lie!
  - (comments might)

When Things go Wrong…

- …and they will!
  - LogCat
  - Toasts
  - Debugger
  - DDMS / ADM

Debugger

Logging and Debugging

EXERCISE 1