Hasso Plattner Institut

IT Systems Engineering | Universität Potsdam

Android

Frank Feinbube

Operating Systems and Middleware

Prof. Dr. rer. nat. Andreas Polze

Android Overview

Android is

- a software platform and operating system for mobile devices
- based on the Linux kernel
- developed by Google / the Open Handset Alliance

Android Apps are

- managed code in the Java* language
- controlling the device via Google-developed Java libraries.

*C and other languages can be compiled to ARM native code and run, but this is not officially supported

2003
Android Inc.

2005
Google

2007
Open Handset Alliance

- Google, HTC, Intel, LG, Motorola, Nvidia, Samsung, T-Mobile,...
- Goal: develop open standards for mobile devices

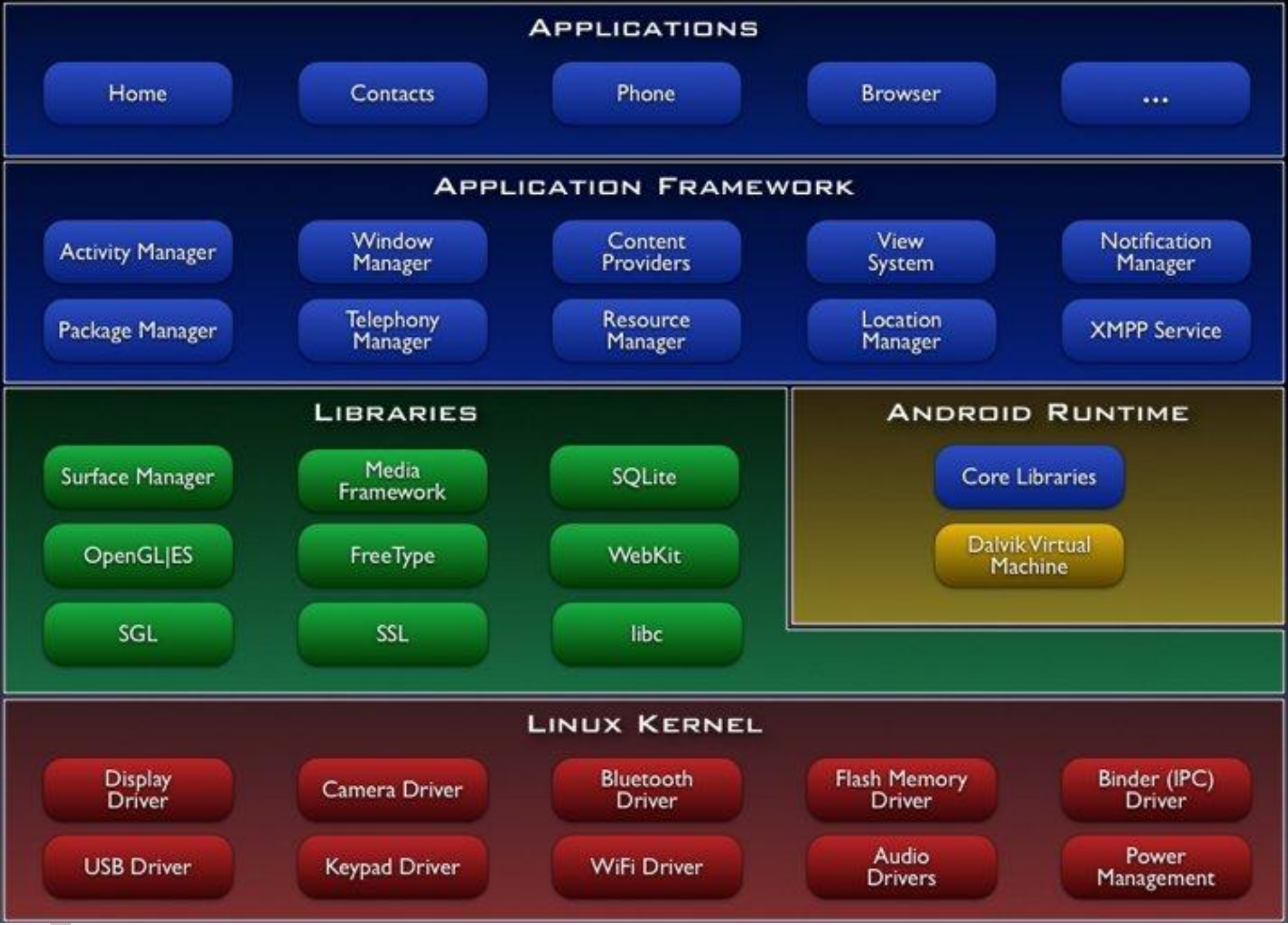
Android Features

Software Features

- Integrated browser based on the open source WebKit engine
- SQLite for relational data storage
- Media support for common audio, video, and still image formats (MPEG4, H.264,MP3, AAC, AMR,JPG, PNG,GIF)
- Dalvik Virtual Machine optimized for mobile device

Hardware Features

- Cellular networking : GSM, EDGE, 3G (hardware dependent)
- LAN : Bluetooth, and Wi-Fi (hardware dependent)
- Graphics Hardware Acceleration (OpenGL)
- Camera, GPS, Compass, Thermometer, … (hardware dependent)
- Touch screen and accelerometer for motion sensing


Applications
Home
Contacts
Phone
Browser
...
Application Framework
Activity Manager
Window Manager
Content Providers
View System
Notification Manager
Package Manager
Telephony Manager
Resource Manager
Location Manager
XMPP Service
Libraries
Surface Manager
Media Framework
SQLite
OpenGL|ES
FreeType
WebKit
SGL
SSL
libc
Android Runtime
Core Libraries
Dalvik Virtual Machine
Linux Kernel
Display Driver
Camera Driver
Bluetooth Driver
Flash Memory Driver
Binder (IPC) Driver
USB Driver
Keypad Driver
WiFi Driver
Audio Drivers
Power Management

Android Software Stack

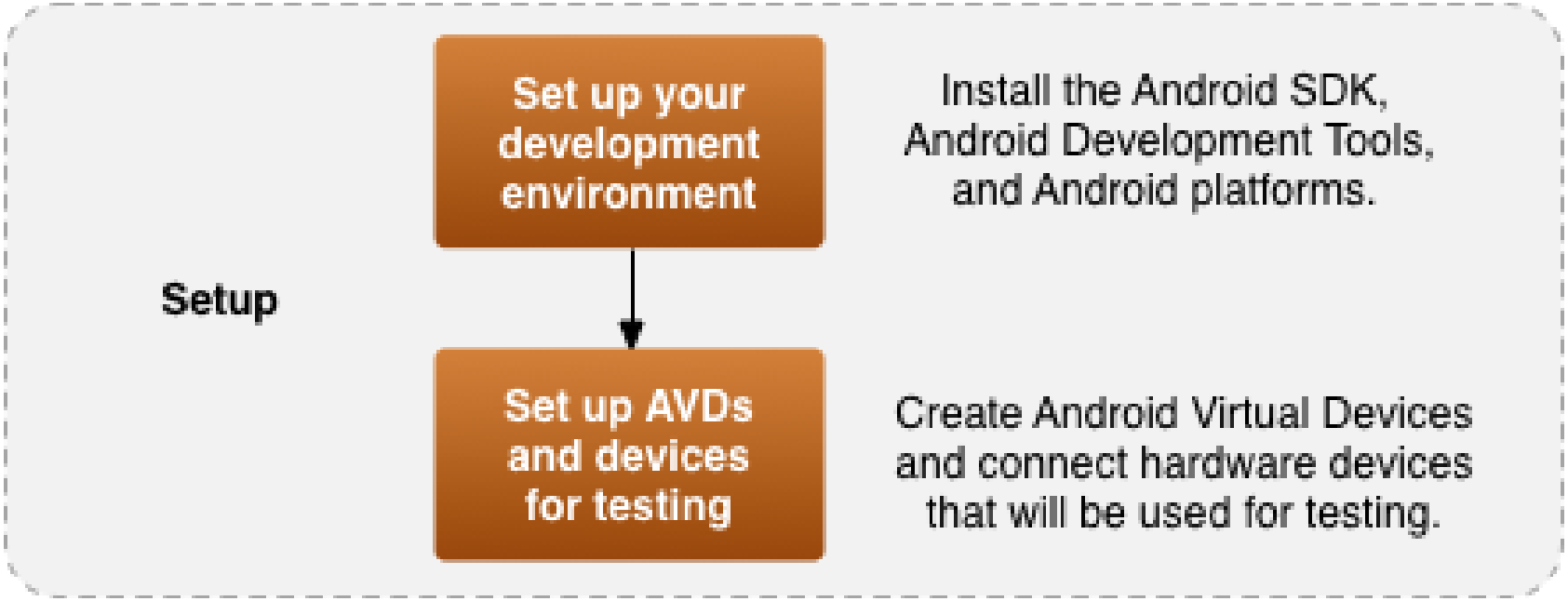
Application Framework

- Content Providers: data sharing between application
- Resource Manager: providing access to non-code resources
- Notification Manager: display alerts in the status bar
- Activity Manager: manages the lifecycle of applications

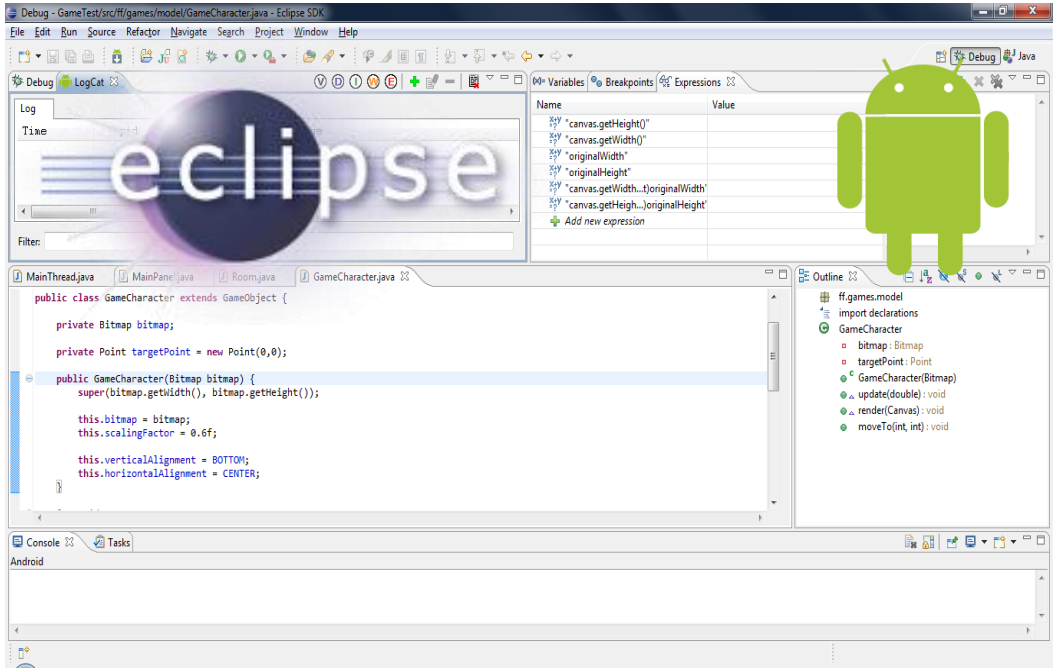
Android Runtime

- core libraries that provides most of the Java-functionality
- every Android application runs in its own process
- Dalvik VM executes files in the (.dex) format
- device can run multiple VMs efficiently

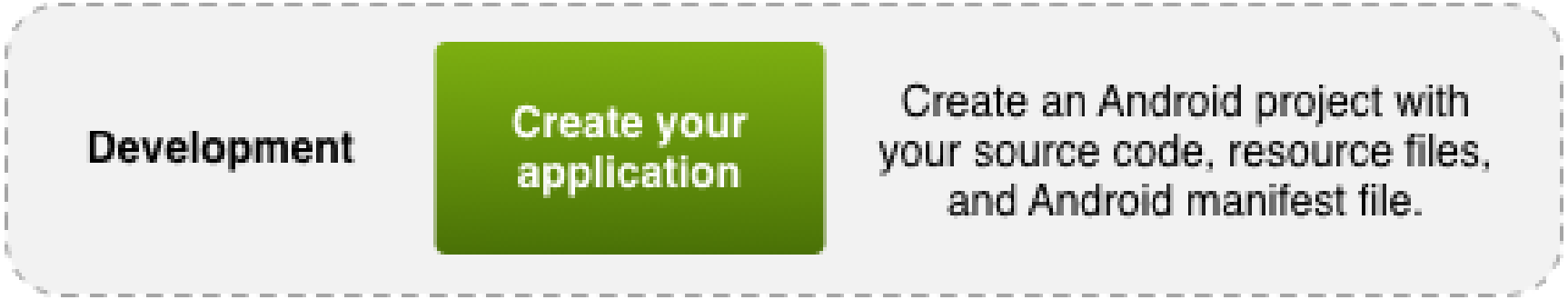
Android Development Process
Step 1: Setup



Android SDK and eclipse IDE



Android Development Process: Step 2: Development



Building blocks for applications

- **Activity**: a single screen
- **Intent Receiver**: to execute in reaction to an external event (Phone Ring)
- **Service**: code that is long-lived and runs without a UI (Media Player)
- **Content Provider**: an application‘s data to be shared with other applications
- **Widgets**: interactive component primary used on the Android homescreen

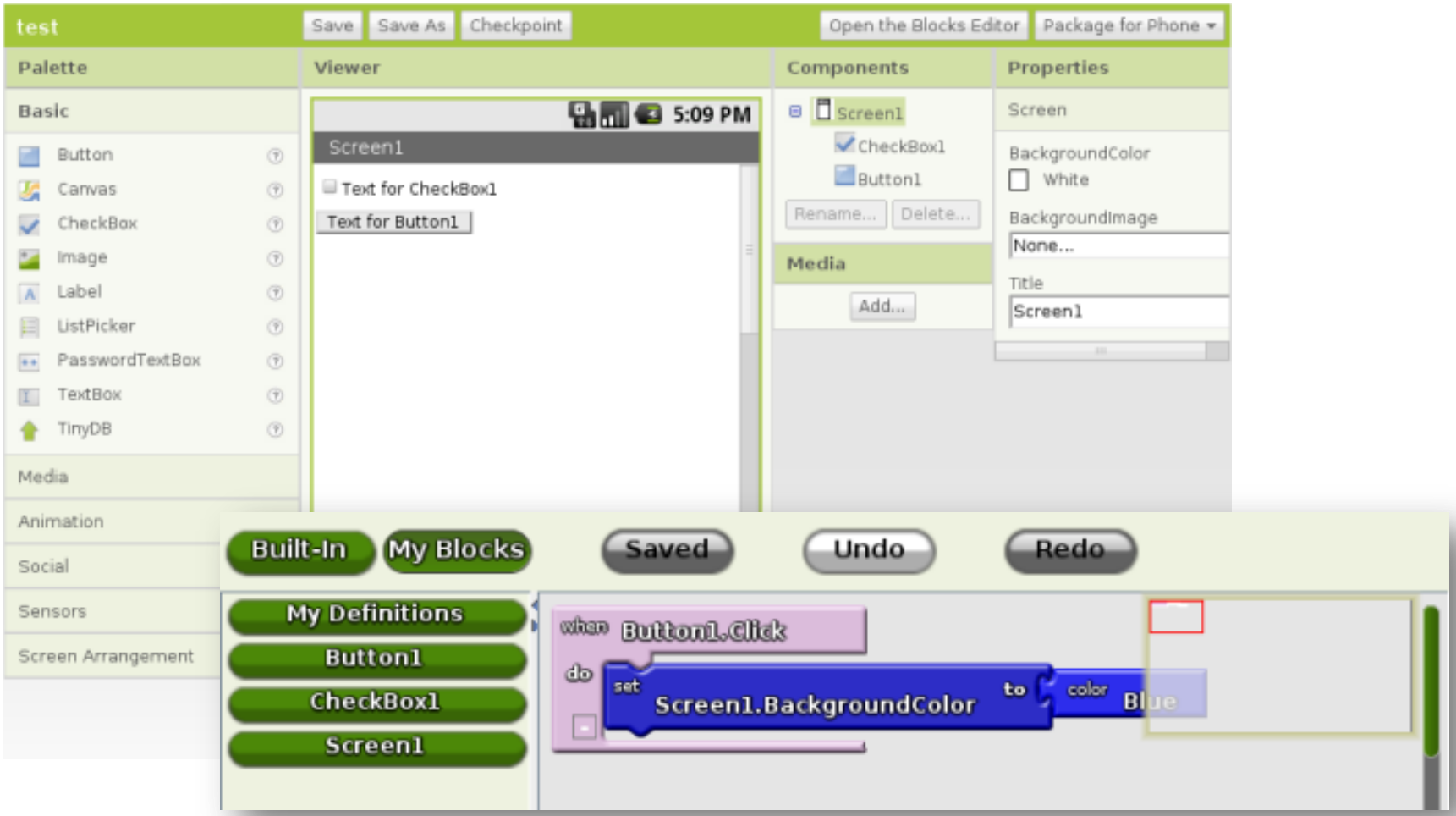
Application parts

- **AndroidManifest.xml**: the control file-tells the system what to do with the top-level components
- **Activities**: an object that has a life cycle-is a chunk of code that does some work
- **Views**: an object that knows how to draw itself to the screen
- **Intents**: a simple message object that represents an "intention" to do something
- **Notifications**
  - is a small icon that appears in the status bar(SMS messages)
  - for alerting the user
- **Services**

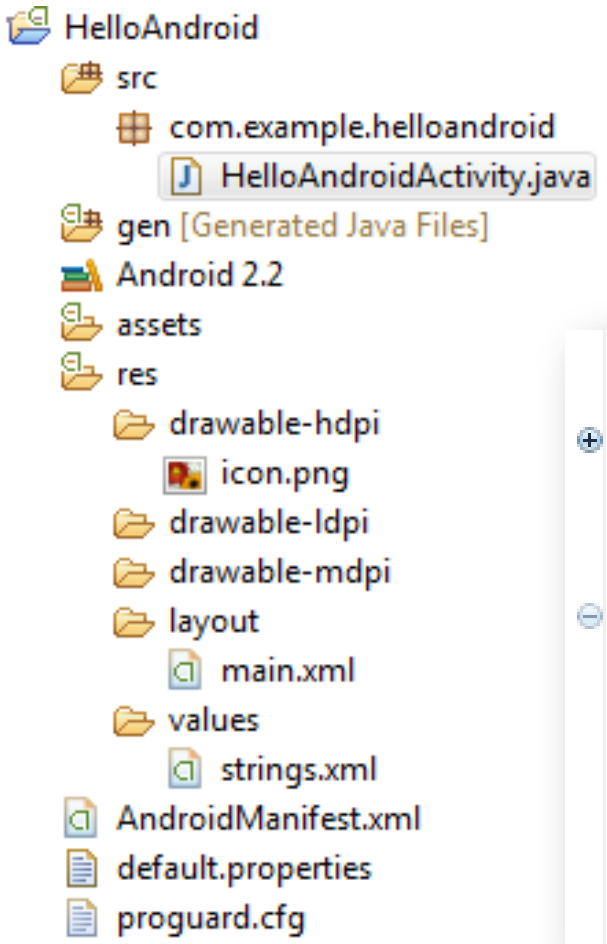
Process termination

- Lifetime is **not** directly controlled by the application itself!
- Low memory -> processes with low importance are killed
  - **Foreground Process** -required for what the user is currently doing
  - **Visible Process** -holding an Activity - visible to the user on-screen but not in the foreground(on pause)
  - **Service Process** -holding a Service - not directly visible to the user- relevant tasks
  - **Background Process** -holding an Activity - not visible to the user - can kill at any time (stopped)
  - **Empty Process** -doesn't hold any active application components (as a cache to improve start-up time)

Google App Inventor

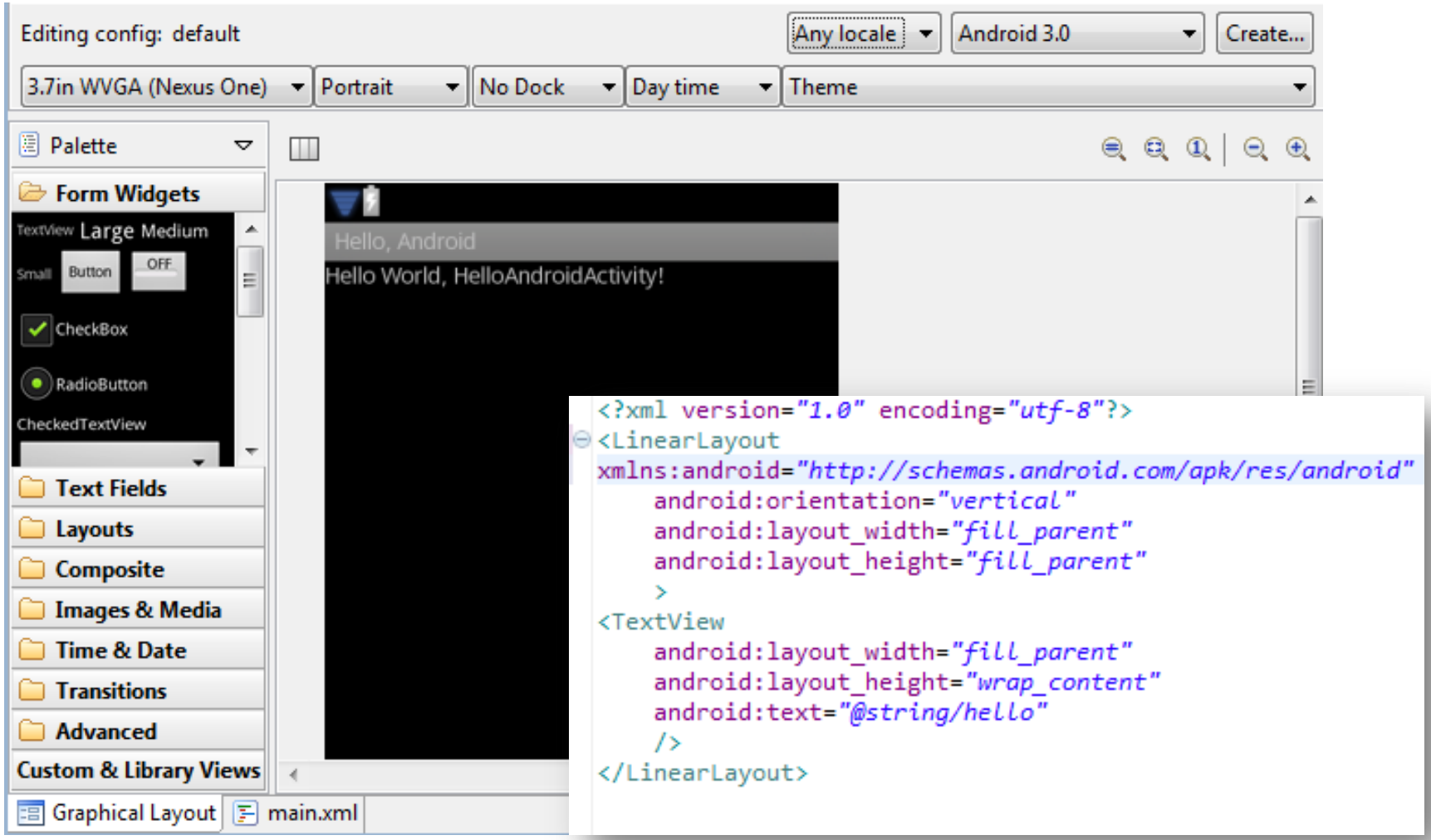


Hello Android: Project & Code

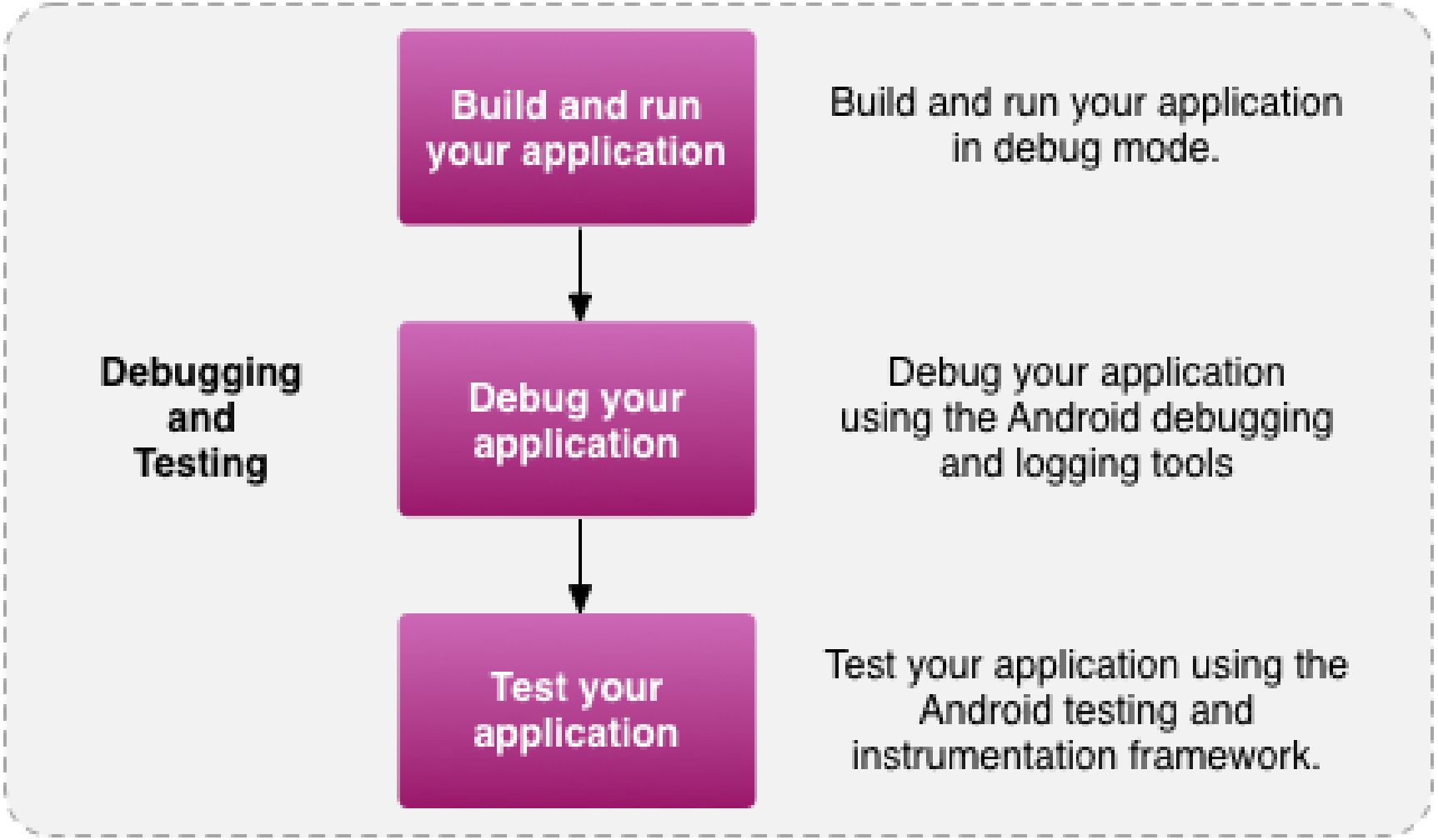



```java
package com.example.helloandroid;

import android.app.Activity;

public class HelloAndroidActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        TextView tv = new TextView(this);
        tv.setText("Hello, Android");
        setContentView(tv);
    }
}
```

Hello Android: UI Designer



Android Development Process
Step 3: Debugging and Testing



Debugging and Logging

- Break-Point Debugging

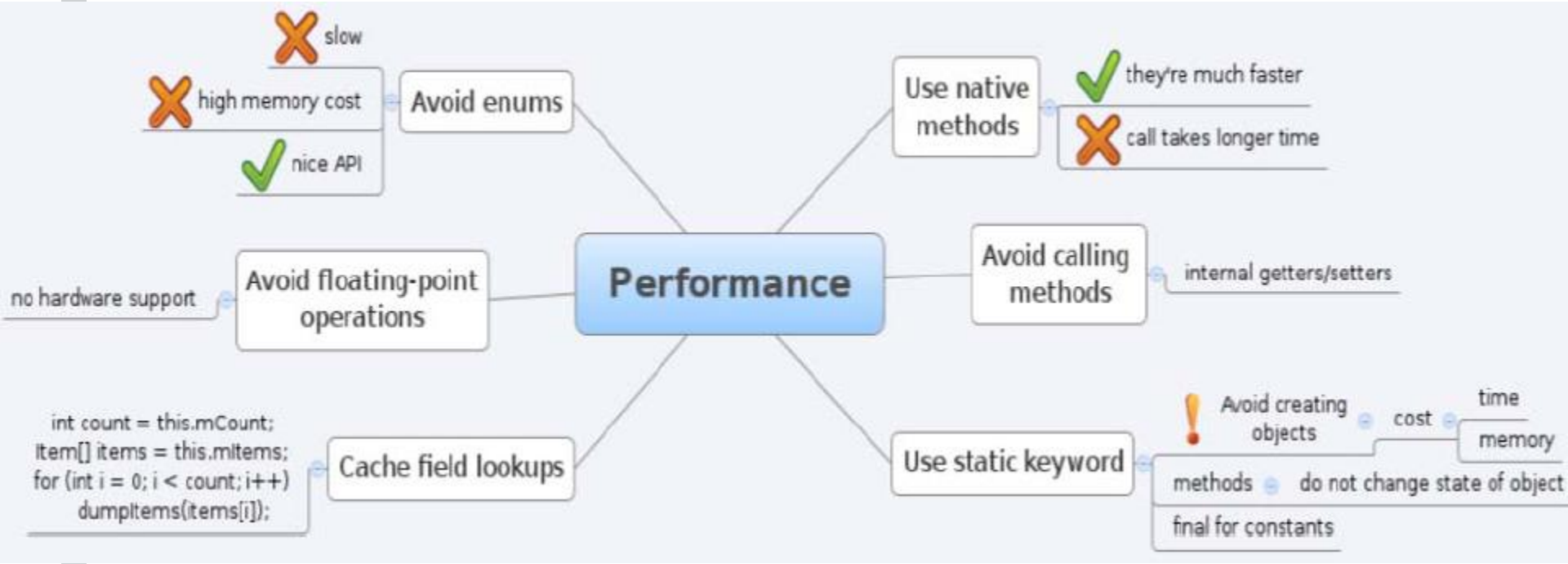
```
        super.onCreate(savedInstanceState);
        TextView tv = new TextView(this);
        tv.setText("Hello, Android");
```

- Logging with LogCat

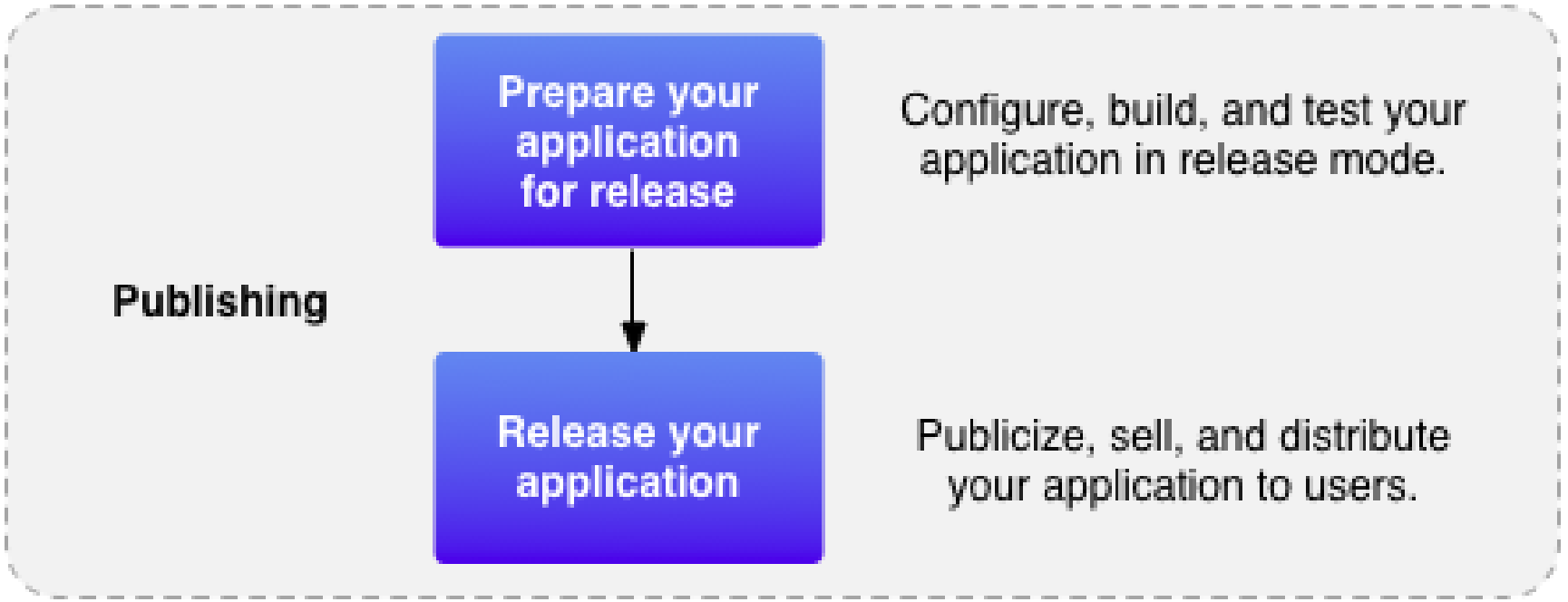
```
Log.d(TAG, "View added");
```

```
11-16 11...  I  282  System.out          waiting for debugger to settle...
11-16 11...  I  282  System.out          debugger has settled (1366)
11-16 11...  D  282  GameTestActivity    View added
11-16 11...  D  282  MainThread          Starting game loop
11-16 11...  D  282  MainThread.init...  Timing elements for stats initialised
11-16 11...  I  59   ActivityManager     Displayed activity ff.games/.GameTestActivity: 5718 ms
11-16 11...  D  282  dalvikvm            GC_FOR_MALLOC freed 9212 objects / 384240 bytes in 80ms
```

Performance Tips



Android Development Process
Step 4: Publishing



Preparing for Release

"As a best practice, your application should meet all of your release criteria for functionality, performance, and stability before you perform the tasks outlined in this document."

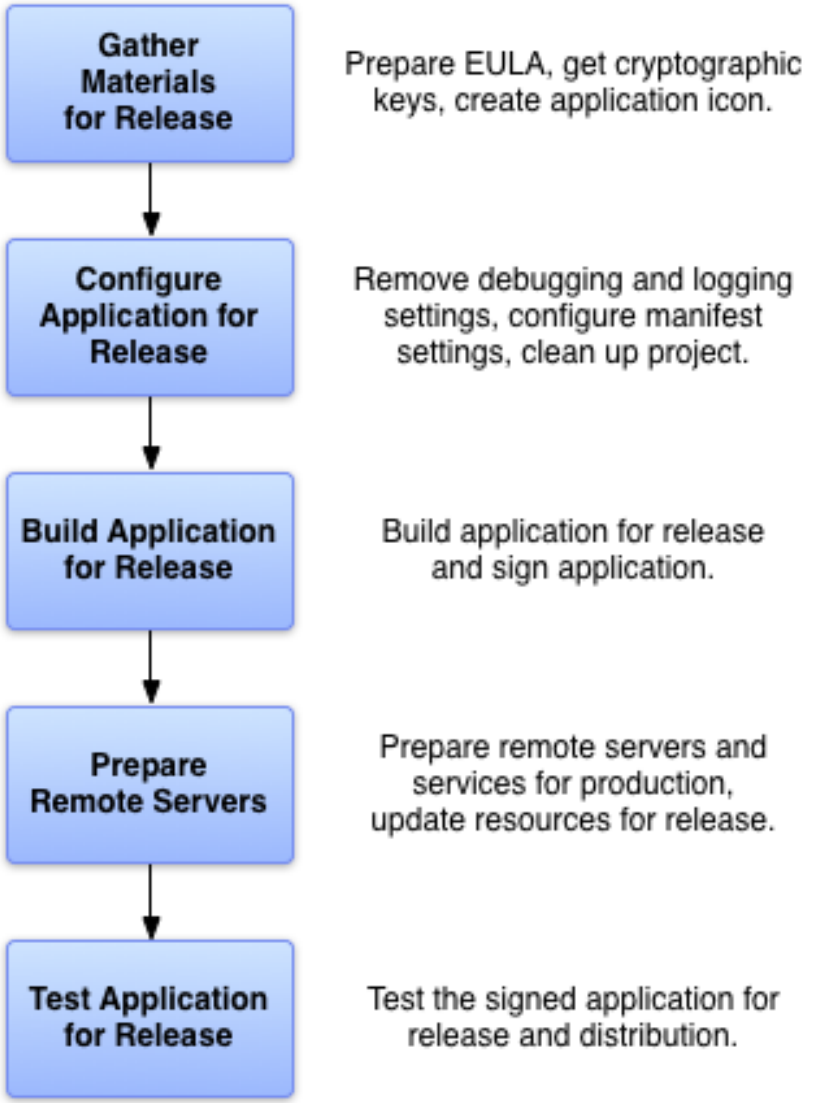
Build with Ant or eclipse IDE

Address compatibility issues

Must be signed (key valid till 2033)

Must define version code and name

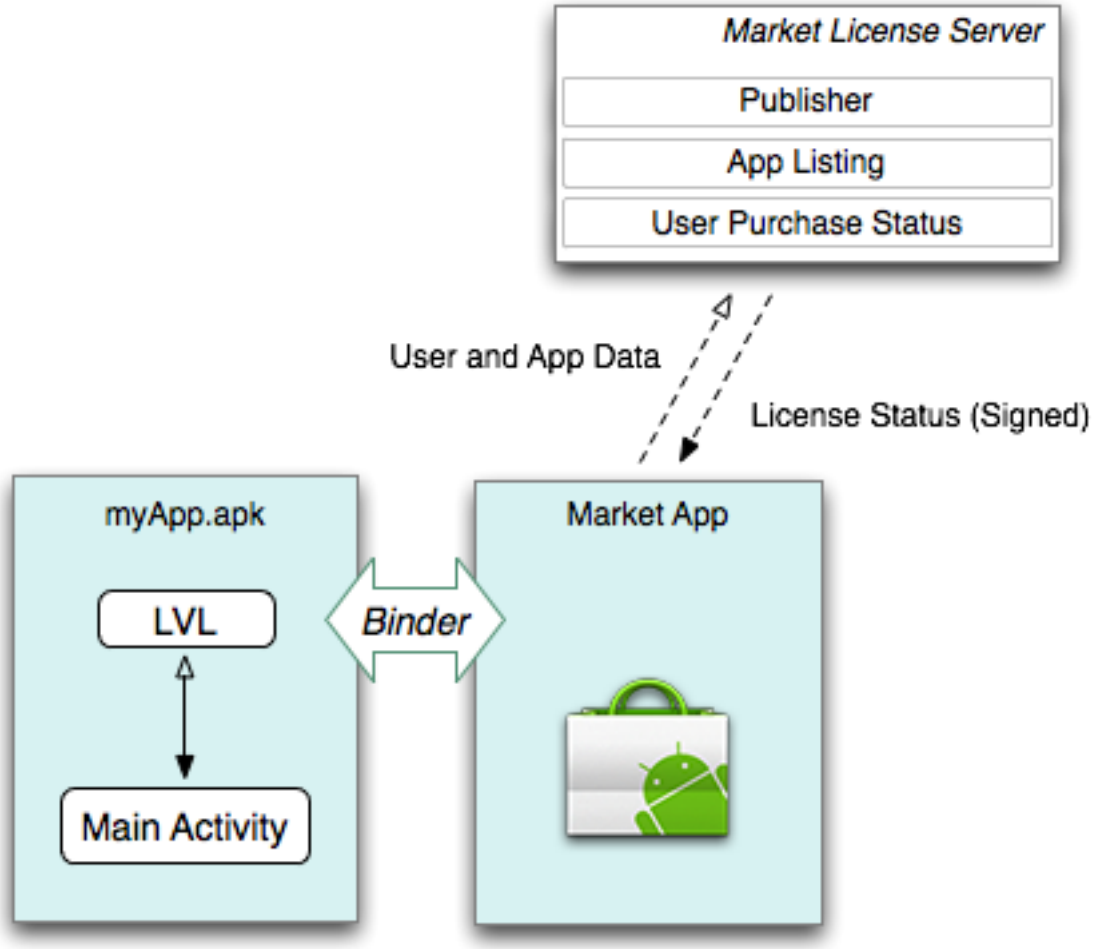
Must define icon and label



Licensing

query a trusted licensing server, to determine whether the application is licensed to the current device user

licensing server is to provide the license status for the current user; the application itself is responsible for querying the server and conditionally granting access to the application



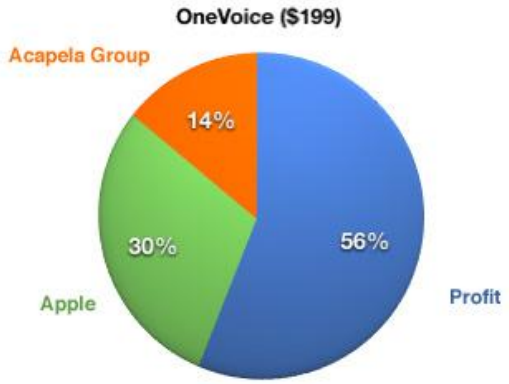
Publishing

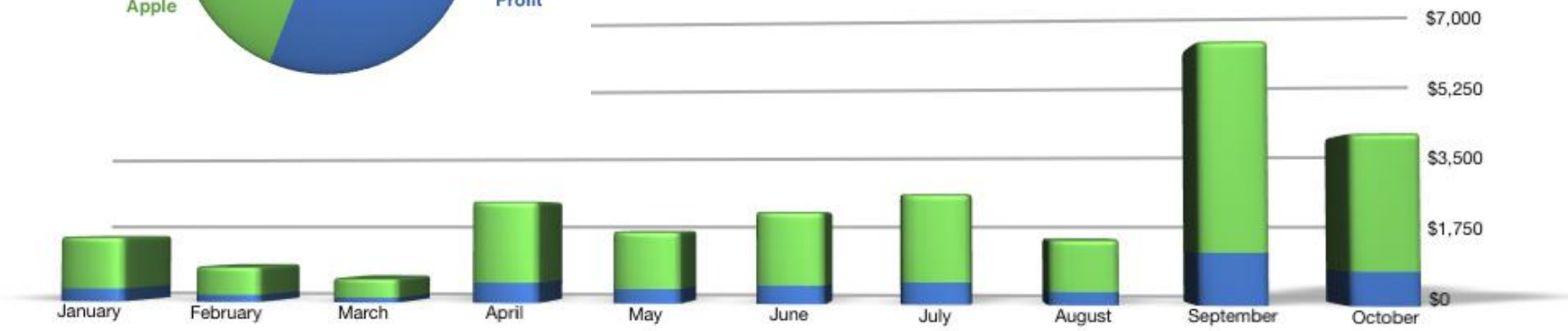
Android Market publisher site: http://market.android.com/publish

Upload Assets

- **APK file size:** Maximum supported size is 50MB.
  - package names for app files are unique and permanent
- **Screenshots:** 2 screenshots are required (additional 6 are optional)
- **High-Resolution Application Icon:** Required.
- **Promotional Graphic, Feature Graphic, Promotional Video:** Optional.

Making Money with Android #1





	January	February	March	April	May	June	July	August	September	October
Expenses	$310	$168	$116	$477	$336	$427	$509	$300	$1,203	$780
Profit	$1,240	$672	$448	$1,908	$1,344	$1,710	$2,038	$1,200	$4,814	$3,122
Revenue	$1,550	$840	$564	$2,385	$1,680	$2,137	$2,547	$1,500	$6,017	$3,902

http://nathanbarry.com/how-i-made-19000-on-the-app-store-while-learning-to-code/

Making Money with Android #2

- $155.37 (**+$124.02**) Amazon Affiliate Program – sold a lot of it equipment! It looks that it was a crazy month as it is much above my average. What is really interesting I sent no more traffic than usual. Just a good month here.
- 111.77 € (**+9.17€**) AdSense – another good month of some extra revenue. Looks to be growing each month.
- $265.27 (**-$1.28**) iStockphoto – looks to be very stable income source. My photos are still selling. As mentioned in my last reports I need to give little more attention to it as I have not been updating my portfolio for ages!
- $1,179.41 (**+$699.65**) Sold 59 copies of my "Make Money on Android" eBook! That is my record! On 59 copies! Really thank you to all of you my friends! I am even more happy as most of you seems to enjoy this eBook a lot! More personal thanks goes in replies to your emails! Anyway thank you!
- $1,985.89 (**+$436.87**) Ads in my free Android applications. As usual this is my best money maker and as you can see it is still growing! This month such increase has happened because of one of described marketing methods that is very often overlooked but provided by Google itself! Just see my eBook for more details.

http://www.kreci.net/reports/developer-income-report-14/