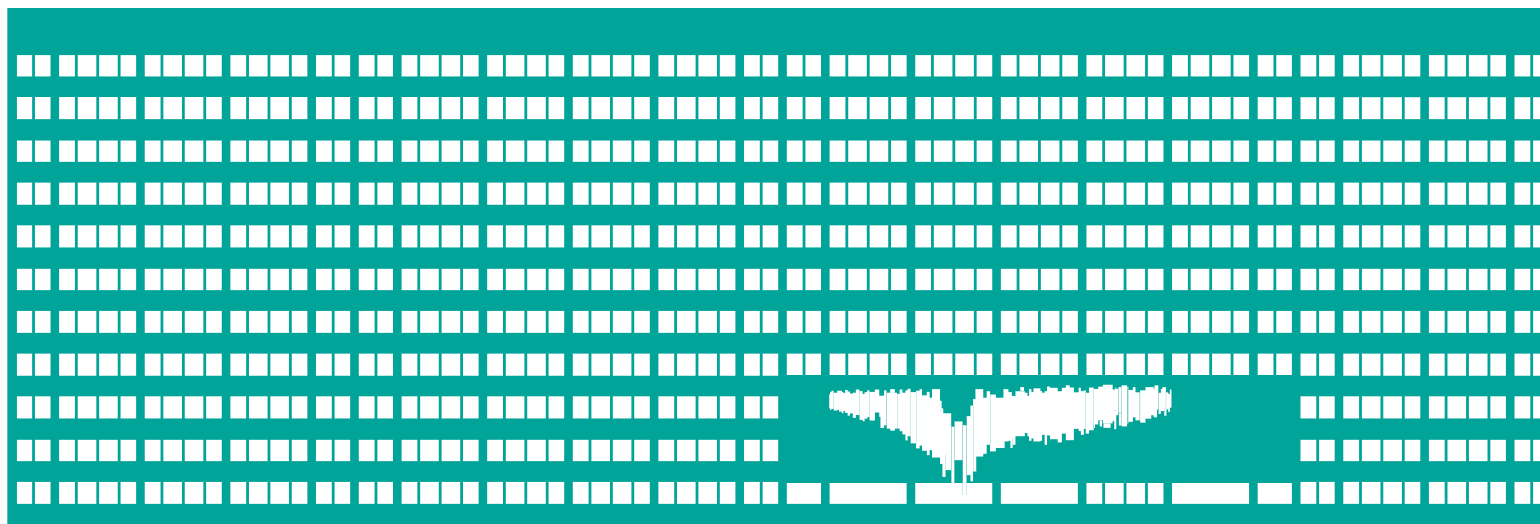


VŠB TECHNICKÁ
UNIVERZITA
OSTRAVA

VSB TECHNICAL
UNIVERSITY
OF OSTRAVA



www.vsb.cz

Android - Introduction

Michal Krumnikl

Android

- Software platform and operating system
 - **Operating system** (with modified Linux kernel)
 - **Middleware**
 - **Key applications**
- Initially developed by Android Inc., which was purchased by Google and later came under the Open Handset Alliance (OHA)
 - Allows writing managed code in the **Java/Kotlin** language.
 - Applications in other languages (**C/C++**) and compiled to ARM native code (NDK)
- **Android Open Source Project**
 - Apache Software License (ASL v2)
 - Opensource: <https://source.android.com>
- Open Handset Alliance
 - Business alliance of 84 firms for developing open standards for mobile devices.



Android Versions and Modifications

- **Official**

- Initial release Android 1.0 September 23, 2008
- Latest release **Android 15 / September 3, 2024**

- **Unofficial Distributions**

- Fire OS
- MIUI ROM
- Kali Nethunter
- Paranoid Android
- LineageOS
- Resurrection Remix OS
- Android Open Kang Project
- ...

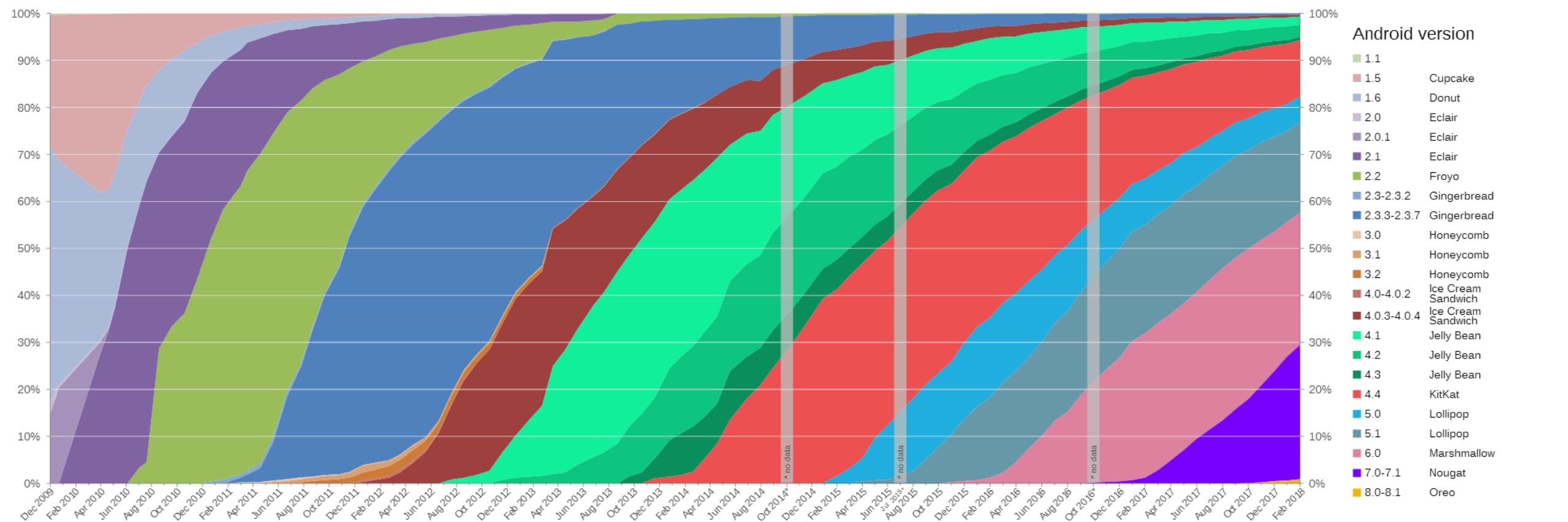
https://en.wikipedia.org/wiki/List_of_custom_Android_distributions



Android Versions

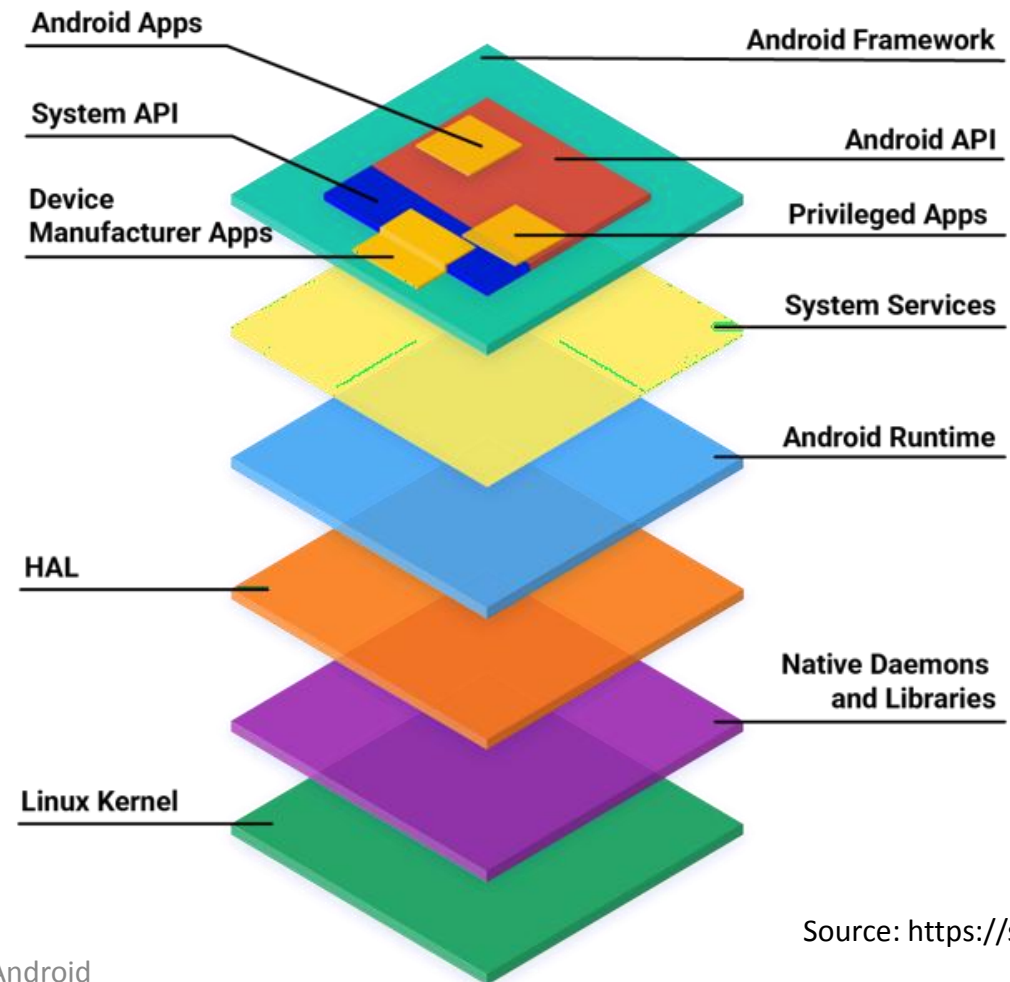
Code name	Version	API level	Code name	Version	API level
Vanilla Ice Cream	15	API level 35	Ice Cream Sandwich	4.0.3 - 4.0.4	API level 15, NDK 8
Upside Down Cake	14	API level 34	Ice Cream Sandwich	4.0.1 - 4.0.2	API level 14, NDK 7
Tiramisu	13	API level 33	Honeycomb	3.2.x	API level 13
Snow Cone	12	API level 31/32	Honeycomb	3.1	API level 12, NDK 6
Red Velvet Cake	11	API level 30	Honeycomb	3.0	API level 11
Quince Tart	10	API level 29	Gingerbread	2.3.3 - 2.3.7	API level 10
Pie	9	API level 28	Gingerbread	2.3 - 2.3.2	API level 9, NDK 5
Oreo	8.0/8.1	API level 26/27	Froyo	2.2.x	API level 8, NDK 4
Nougat	7.0/7.1	API level 24/25	Eclair	2.1	API level 7, NDK 3
Marshmallow	6.0	API level 23	Eclair	2.0.1	API level 6
Lollipop	5.0/5.1	API level 21/22	Eclair	2.0	API level 5
KitKat	4.4 - 4.4.4	API level 19	Donut	1.6	API level 4, NDK 2
Jelly Bean	4.3.x	API level 18	Cupcake	1.5	API level 3, NDK 1
Jelly Bean	4.2.x	API level 17	(no code name)	1.1	API level 2
Jelly Bean	4.1.x	API level 16	(no code name)	1.0	API level 1

Android Historical Version Distribution



Android Platform

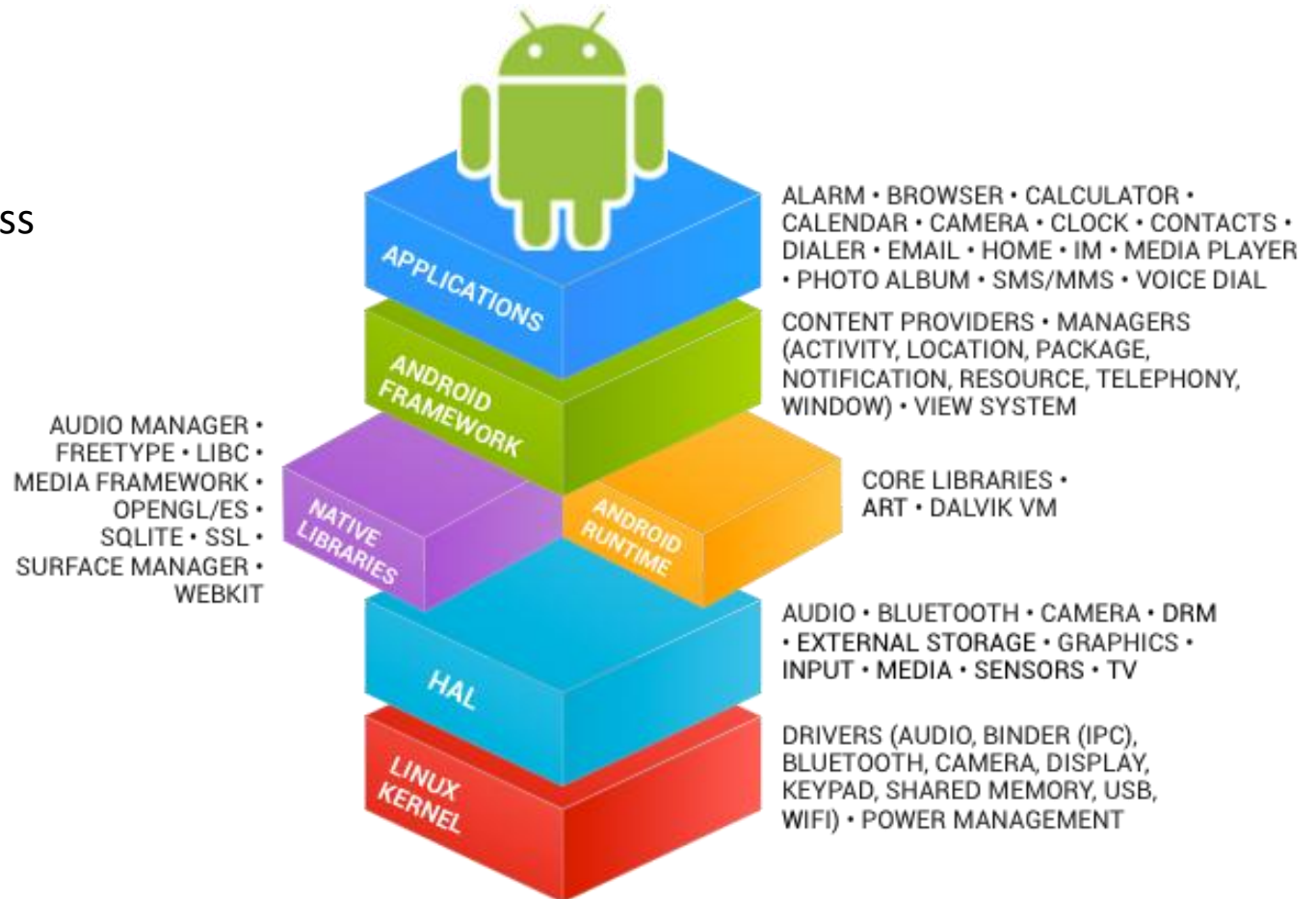
- **Linux Kernel**
 - Device drivers, memory management, process management, and networking.
- **Native Libraries**
 - Written in C/C++ internally, but called from Java interfaces.
 - Surface Manager, 2D and 3D graphics, Media codecs, SQLite, WebKit.
- **Android Runtime**
 - **Android Runtime (ART)**
 - **Dalvik Virtual Machine (< Android 4.4)**
- **Application Framework**



Source: <https://source.android.com/>

Android Platform

- **Linux Kernel**
 - Device drivers, memory management, process management, and networking.
- **Native Libraries**
 - Written in C/C++ internally, but called from Java interfaces.
 - Surface Manager, 2D and 3D graphics, Media codecs, SQLite, WebKit.
- **Android Runtime**
 - **Android Runtime (ART)**
 - **Dalvik Virtual Machine (< Android 4.4)**
- **Application Framework**



Android Kernel

- **Linux Kernel** (started from 2.6.24)
 - 1.5 (Cupcake, 2.6.27), 1.6 (Donut, 2.6.29)
 - 2.0/2.1 (Eclair, 2.6.29), 2.2 (Froyo, 2.6.32)
 - 4.1 (Jelly Bean, 3.1.10)
 - 6.0.1 (Lollipop, 3.18.10)
 - 9 (Pie, 4.4.107, 4.9.84, and 4.14.42)
 - 11 (4.14, 4.19)
 - 13 (5.10, 5.15)
 - **14 (5.10, 5.15, 6.1)**
 - **15 (6.1, 6.6)**
- Some features are removed
 - No GLIBC support (GNU C Library)
 - No native window system (no X/Wayland)
- Android components added
 - Alarm, Android Shared Memory
 - Kernel Memory Killer
 - Kernel Debugger
 - Logger
 - ...

Android or Linux Kernel ?

- **Standard Features**

- Process & Memory Management
- File & Network I/O
- Device Drivers
- Security

- **Android-specific**

- Power Management
 - Energy Aware Scheduler task placement optimizations
- Android Share Memory (ashmem)
- Low Memory Killer
- Interprocess Communication

Android Platform Specifics

- **Power Management**

- Not using standard Linux power management
- Application uses user space library to inform framework about its constraints. Implemented by locking mechanism

- **Binder**

- Driver to facilitate inter-process communication between applications and services. A pool of threads is associated to each application to process incoming IPC.

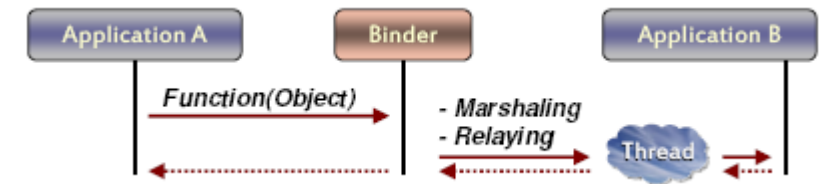


Image Source: Esmertec – OpenExpo2008

Android Platform Libraries

- **System C library**

- **Bionic Libc** - BSD-derived implementation of the standard C system library (libc), tuned for embedded Linux-based devices
- <https://android.googlesource.com/platform/bionic/>

- **Surface Manager**

- Render all surfaces to a framebuffer device. Supports 2D and 3D surfaces. Can exploit OpenGL ES acceleration.

- **Graphics Libraries**

- SGL - the underlying 2D graphics engine.
- FreeType - bitmap and vector font rendering
- 3D libraries, based on OpenGL ES 1.0 APIs

- **Audio Manager**

- Processes multiple audio streams into PCM
- Handles several input and output streams

Android Application Framework

- **Package Manager**
 - Keeps track of app packages on device
- **Window Manager**
 - Manages the windows of the applications
- **View System**
 - Provides common user interface elements
- **Resource Manager**
 - Manages non-compiled resources
- **Activity Manager**
 - Manages App lifecycle and navigation stack
- **Content Provider**
 - Provides inter-application data sharing

Android SDK Features

- No licensing, distribution, or development fees
- Wi-Fi hardware access
- GSM, EDGE, 3G, 4G networks for telephony and data
- API for location based services (e.g. GPS)
- Access to Bluetooth libraries
- IPC message passing, shared data stores
- Background processes and applications
- Widgets, Live Folders, Live Wallpaper
- HTML5 WebKit based browser
- 2D and 3D graphics (OpenGL ES)
- Multimedia support
- Localization using a dynamic resource framework
- Access to hardware, including camera, GPS and accelerometer
- Native Google Maps and Geocoding
- SQLite Database
- Optimized Memory and Process Management
- ...

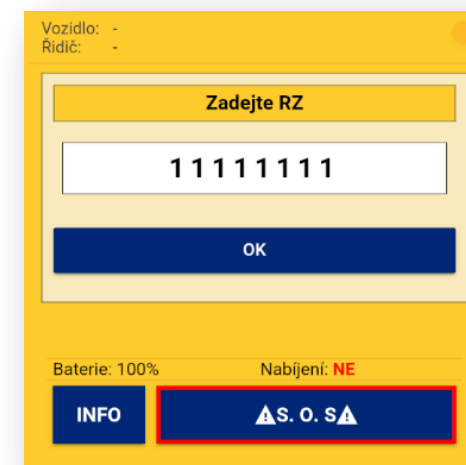
Android Standard Applications

- **Home** – main screen
- **Contacts** – contacts database
- **Phone** – dial phone numbers
- **Browser** – view web pages
- **Email reader** – compose & read email messages

You can substitute any of them with your own or 3rd party application



<https://github.com/FossifyOrg>

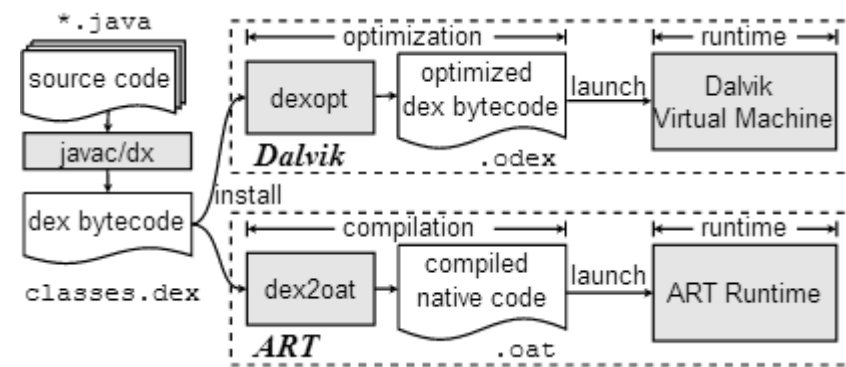


Android Runtime before Lollipop (deprecated)

- Every Android application runs in its own process, with its own instance of the **Dalvik** virtual machine.
- Dalvik has been written so that a device can run multiple VMs efficiently. The Dalvik VM executes files in the **Dalvik Executable (.dex)** format which is optimized for minimal memory footprint.
- The VM is **register-based**, and runs classes compiled by a Java language compiler that

have been transformed into the .dex format by the included "dx" tool.

- The Dalvik VM relies on the Linux kernel for underlying functionality such as threading and low-level memory management.



Source: Mingshen Sun, Tao Wei, and John C.S. Lui, TaintART

Dalvik Virtual Machine (deprecated)

- **Designed for slower CPU, less RAM and limited battery life**
 - Dalvik VM is a **register-based** architecture.
 - An uncompressed .dex file is typically a few percent smaller in size than a compressed .jar
 - The VM was slimmed down to use less space.
- It uses its own bytecode, not Java bytecode.
 - Tool dx is used to convert Java .class
- Dalvik does not align to Java SE nor Java ME Class Library profiles.

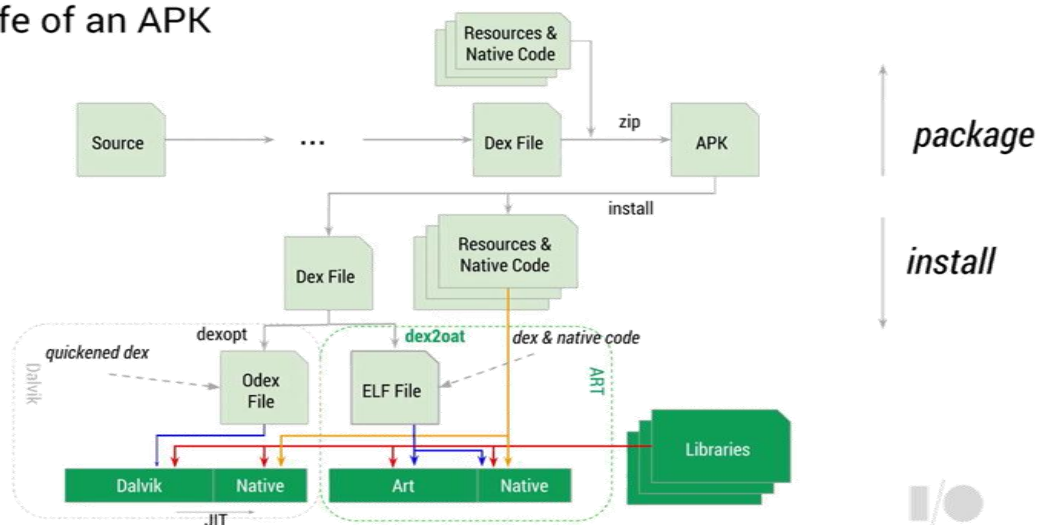
Dalvik Executable Format

<https://source.android.com/docs/core/runtime/dex-format>

ART

- **Entirely replaced Dalvik** since Android 5.0
- Hybrid combination of AOT, just-in-time (JIT) compilation, and profile-guided compilation.
<https://source.android.com/docs/core/runtime/configure>

The life of an APK

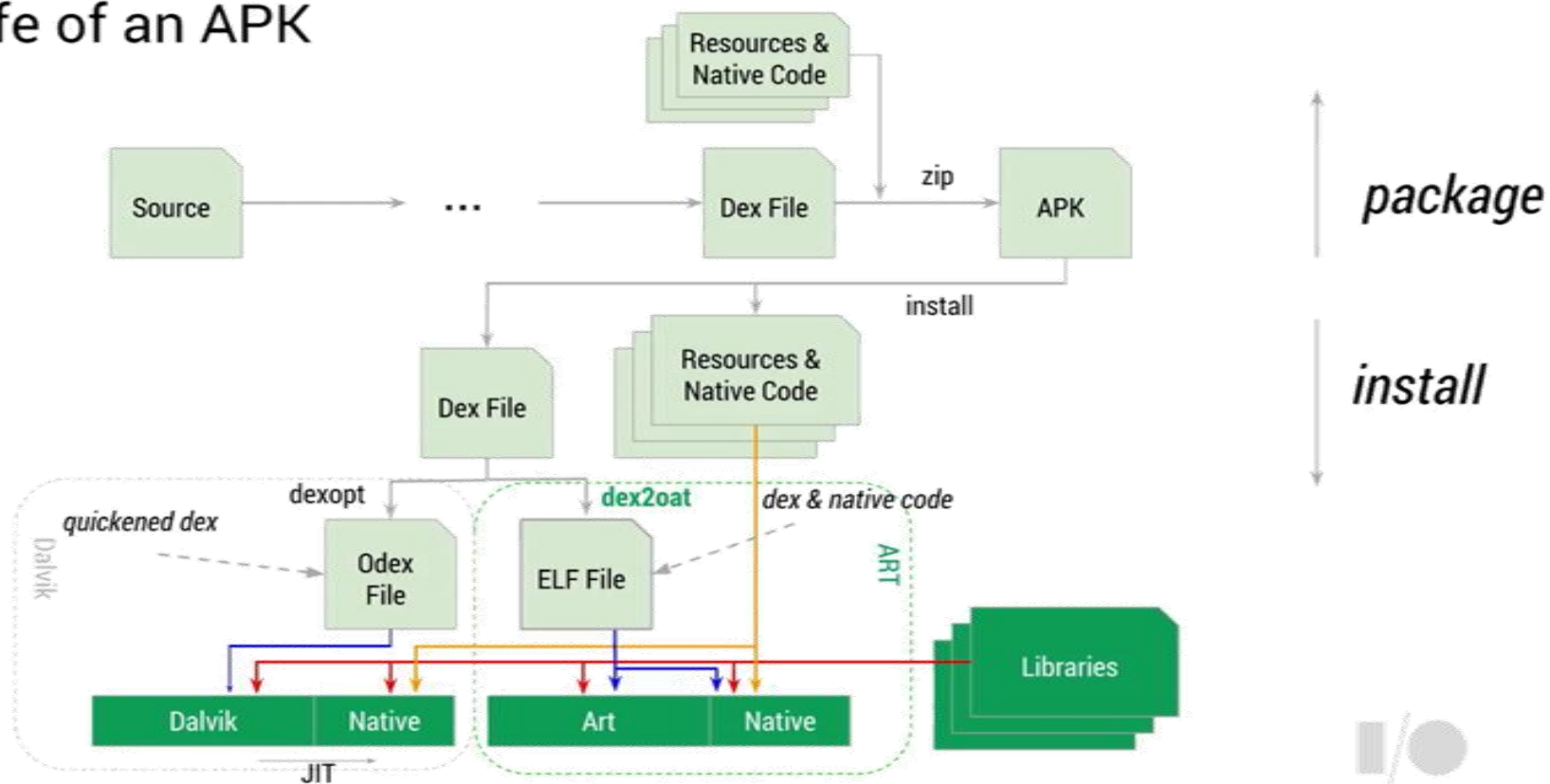


1. An application is initially installed without any AOT compilation. It will be interpreted, and methods frequently executed will be JIT compiled.
2. When the device is idle and charging, a compilation daemon runs to AOT-compile frequently used code based on a profile generated during the first runs.
3. The next restart of an application will use the profile-guided code and avoid doing JIT compilation at runtime for methods already compiled.

ART gradually improves - <https://source.android.com/docs/core/runtime/improvements>

Android Runtime Dalvik vs. ART

The life of an APK



Source: Anandtech, <http://www.anandtech.com/show/8231/a-closer-look-at-android-runtime-art-in-android-l>

Android Bootup

- **Boot ROM** - Loads the first stage of the boot loader into internal RAM.
- **Bootloader** - Inits memories, verifies for security, and loads kernel.
- **Kernel** - Sets up interrupt controllers, memory protection, caches, and scheduling; launches user space processes.
- **Init process** - Parses init.rc scripts, mounts file systems, launches Zygote, launches system process.

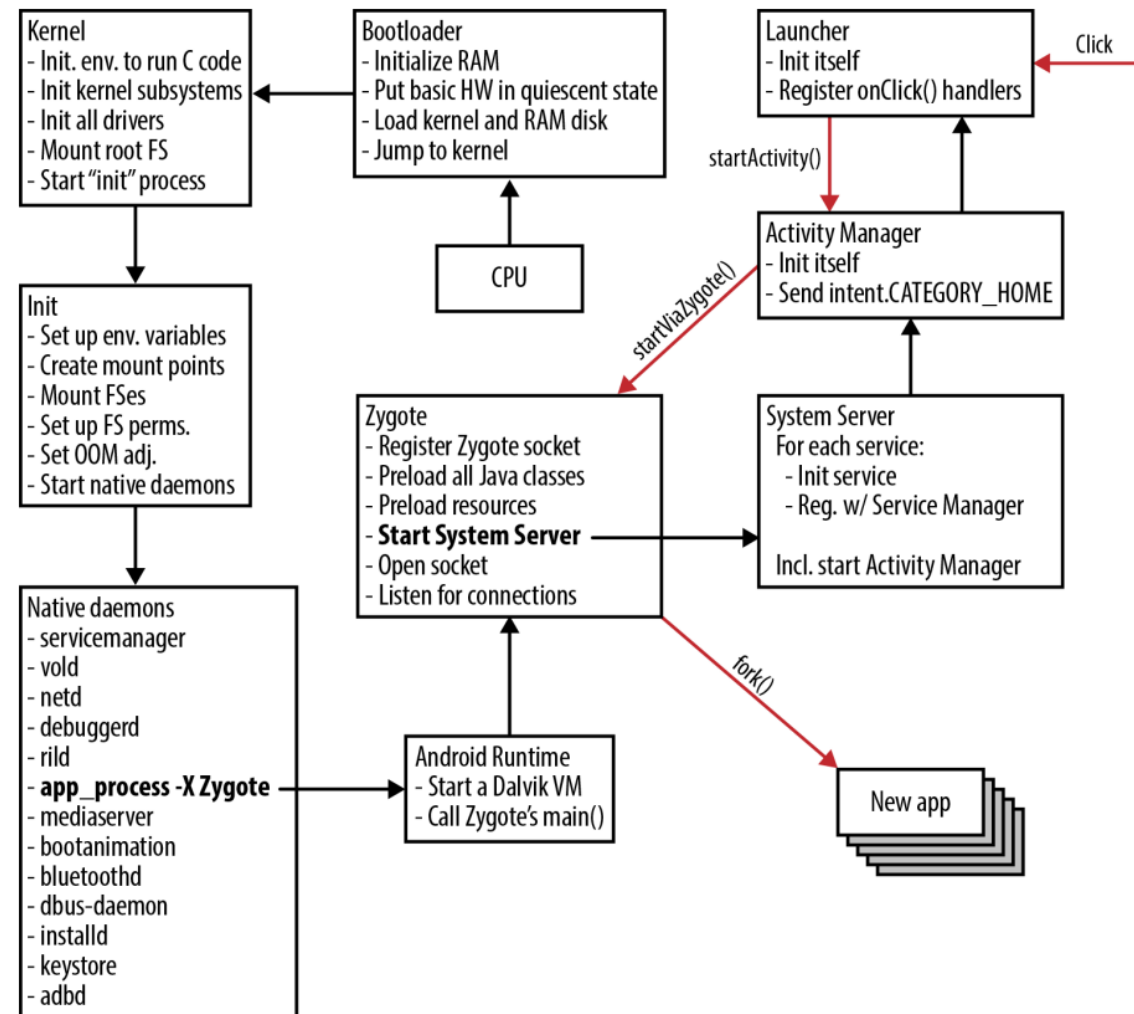


Image Source: zybuluo.com

Android Bootup cont.

- **Zygote** - Sets up Java Runtime and init memory for Android objects.
- **System server** - First Java component in the system, starts core Android services.

Similar boot of Android in Automotive

https://source.android.com/docs/automotive/power/boot_time

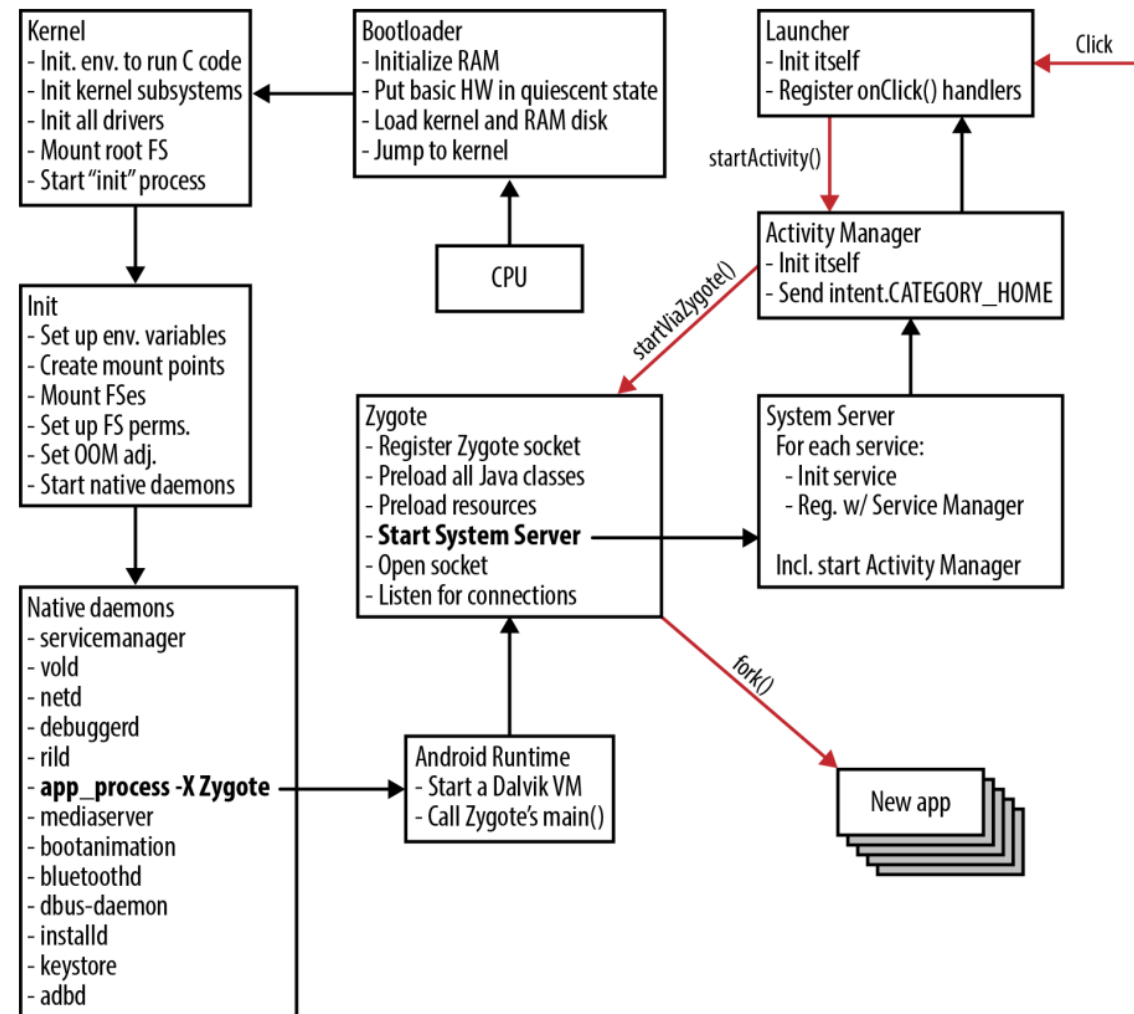


Image Source: zybuluo.com

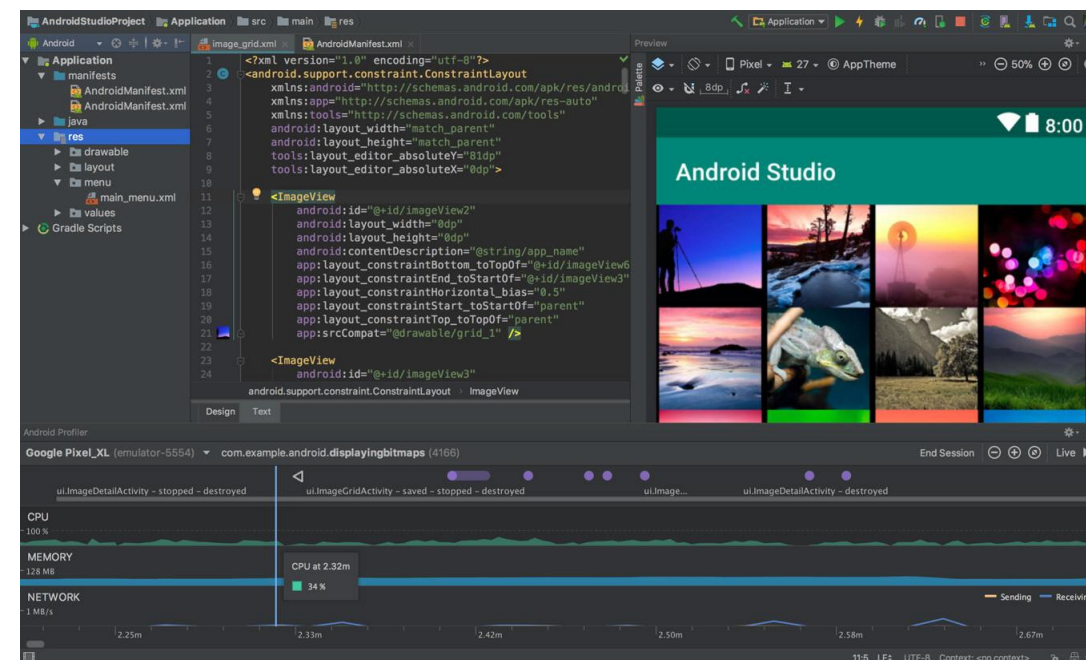
Development Tools

• Android SDK

- Class Library
- Developer Tools
 - **dx** – Dalvik Cross-Assembler. Converts Java Class files into "dex" (Dalvik Executable) files.
 - **aapt** – Android Asset Packaging Tool. Applications are packed into an .apk (Android Package) file.
 - **adb** – Android Debug Bridge
 - **ddms** – Dalvik Debug Monitor Service
- Emulator and System Images
- Documentation and Sample Code

• Android NDK

• Android Studio



Development Tools

- Dalvik Debug Monitor Server (deprecated)

- **Android Device Monitor**

Allows performing debugging and performance trace of applications.

- **LogCat**
 - Logs including `System.out.print()` statements.
- **File explorer**
- **Traceview**
 - Display method traces taken from running application

- **Hierarchyview**

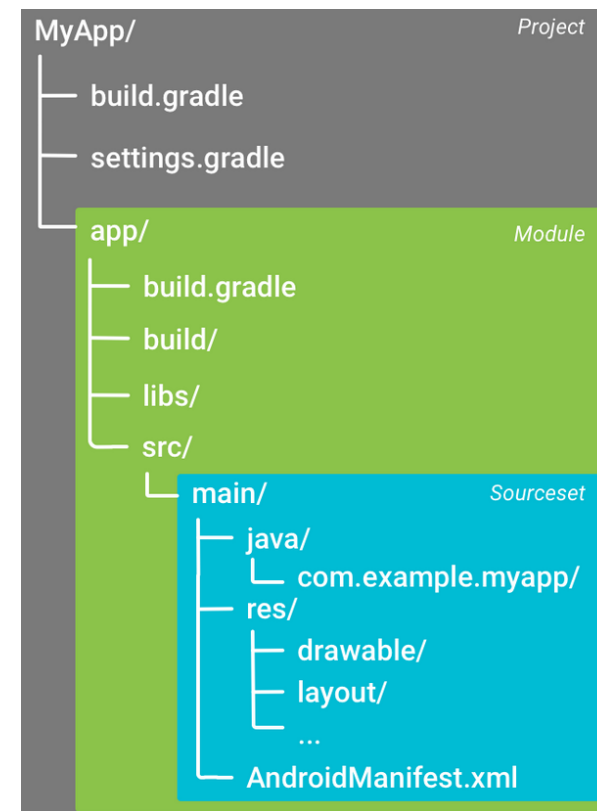
- Shows the runtime organization of the user interface

- **Android Debugging Bridge / Shell**

- Access Android emulator via the text console.

Android Application Package - APK

- **Distribution and Installation**
- Contains program's code (such as .dex files), resources, assets, certificates, and manifest file.
- Manually install APK files only after turning on an "Unknown Sources" setting
- Build configuration files, *build.gradle*, are plain text files use Domain Specific Language (DSL) to describe and manipulate the build logic using Groovy, which is a dynamic language for the Java Virtual Machine (JVM).



APK Analyzer

Build > Analyze APK

- Analyze file and size information
- Content of AndroidManifest.xml
- View and browse content of DEX files
- Show bytecode, find usages
- **Compare two different APK**

Application-release-unsigned.apk (old) vs Application-debug.apk (new)

File	Old Size	New Size	Diff Size
Application-release-unsigned.apk	954.2 KB	3 MB	2 MB
classes.dex	1.1 MB	6.1 MB	5 MB
> META-INF/	340 B	86.5 KB	86.2 KB
> res/	245.3 KB	280.8 KB	35.5 KB
resources.arsc	248.7 KB	262.3 KB	13.6 KB
AndroidManifest.xml	3.9 KB	4 KB	52 B
> kotlin/	24.1 KB	24.1 KB	0 B
DebugProbesKt.bin	1.7 KB	1.7 KB	0 B
> assets/	1.1 KB	0 B	-1.1 KB

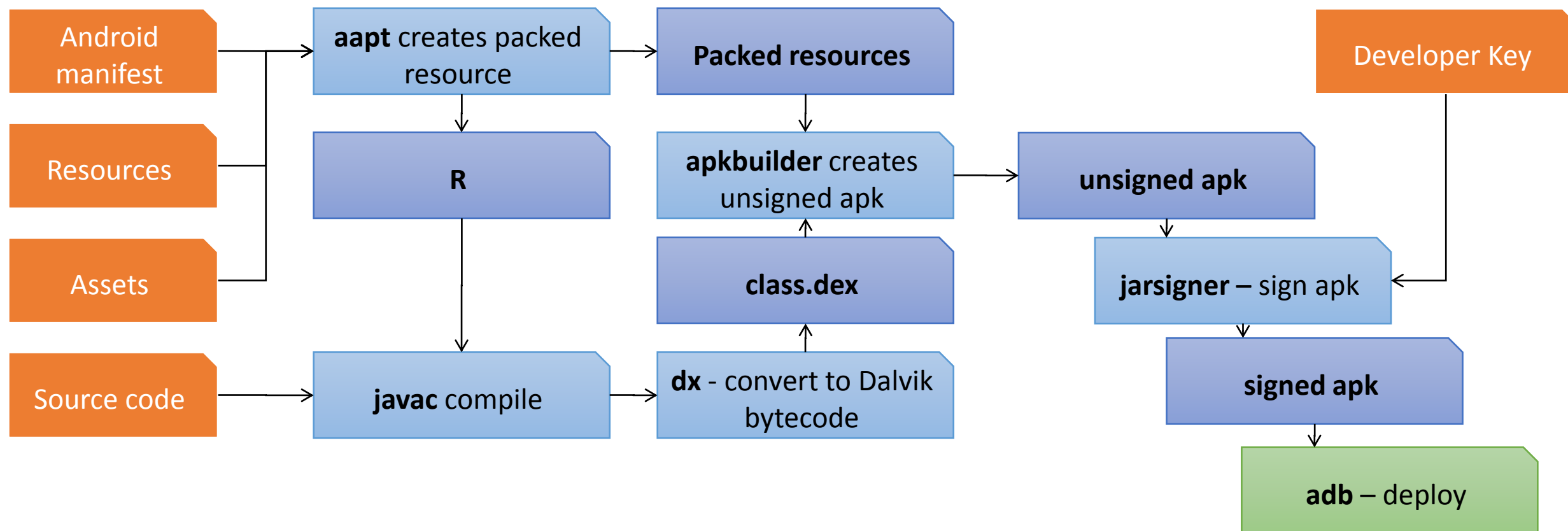
Application-debug.apk

com.example.android.displayingbitmaps (Version Name: 1.0, Version Code: 1)

APK size: 3 MB, Download Size: 2.7 MB Compare with previous APK...

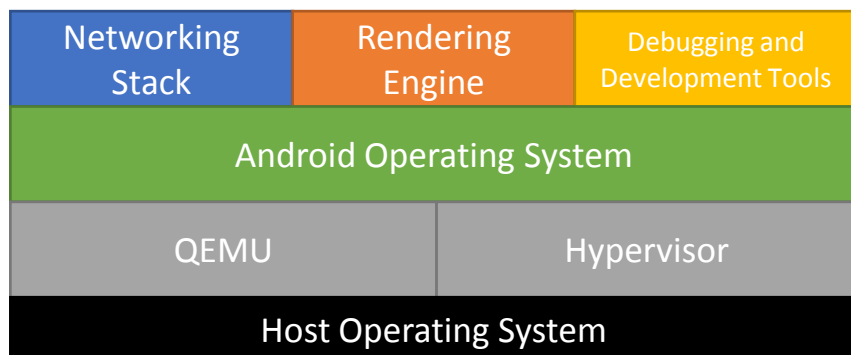
File	Raw File Size	Download Size	% of Total Download
classes.dex	2.4 MB	2.2 MB	89.5%
> res	174 KB	166 KB	6.7%
resources.arsc	262.3 KB	57.1 KB	2.3%
> META-INF	28.4 KB	25.8 KB	1%
> kotlin	9.2 KB	9.2 KB	0.4%
AndroidManifest.xml	1.3 KB	1.3 KB	0.1%

APK Creation Process

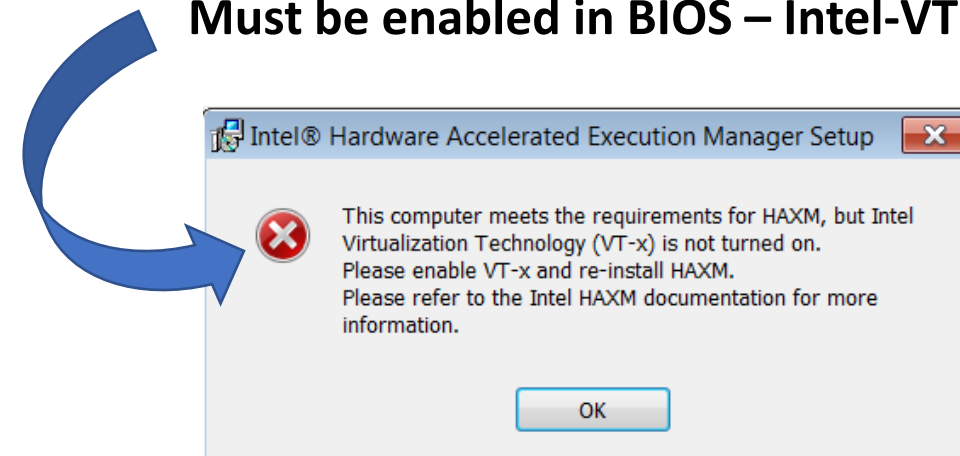


Android Emulator

- **Emulation Engine (QEMU)**
 - KVM (Kernel-based Virtual Machine) or HAXM (Intel Hardware Accelerated Execution Manager)
- **Android Operating System**
- **Host Operating System Integration**
- Recommended more than 8GB RAM
- Uses Intel Hardware Accelerated Execution Manager



Must be enabled in BIOS – Intel-VT



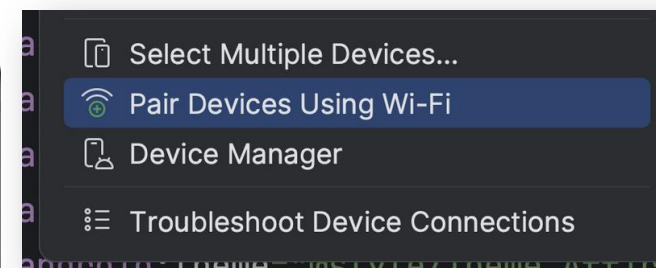
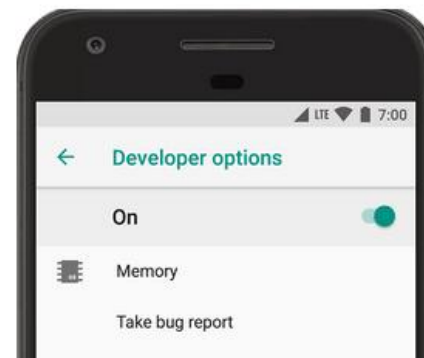
Android Emulator

- **Advanced Features**

- Can emulate many different device/user characteristics
 - Network speed/latencies
 - Battery power
 - Location coordinates
- Emulate incoming phone calls and SMS messages
- Can interconnect multiple emulators

- **Installing on Real Device**

- <https://developer.android.com/tools/adb>
- Turn on "**USB Debugging**" on your device in the settings.
 - Android 9 (API level 28) and higher: Settings > About Phone > Build Number (tap 7x)
- Install ADB driver for mobile phone.
 - On Linux edit udev



References

- <http://developer.android.com/>
- <https://developer.android.com/tools/adb>
- Esmertec – OpenExpo2008 presentation
 - http://www.openexpo.ch/fileadmin/documents/2008Zuerich/Slides/33_Printemps.pdf
- Krzysztof Bzowski, Android lectures
 - http://home.agh.edu.pl/~kbzowski/2016_ANDROID/

Thank you for your attention

Mgr. Ing. **Michal Krumnikl**, Ph.D.

+420 597 325 867

michal.krumnikl@vsb.cz

www.vsb.cz