

ANDROID

Android App Development

COS216
AVINASH SINGH
DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF PRETORIA



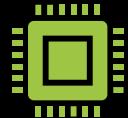
HISTORY

- Andy Rubin (veteran Apple engineer) was the main founder of Android in 2003
- Android was the nickname given to Andy by his co-workers due to his love of robots
- The logo is officially named “Bugdroid”
- Android was initially intended for cameras

HISTORY



In 2005 Android was incorporated as a subsidiary of Google



The development of the Android OS was announced in 2007 by OHA



HTC launched the first Android mobile in 2008

HISTORY

- OHA (Open Handset Alliance) is a business alliance of hardware, software and telecom companies
- The mandate of OHA is to advance the cause of open sourcing for mobile phones
- The initial members included
 - Google, HTC, Dell, Intel, Motorola
 - Qualcomm, Texas Instruments, Samsung
 - LG, T-Mobile, Nvidia, and Wind River Systems

OVERVIEW



Open source (Apache license)



OS primarily used by mobile devices



Other devices include televisions, tablets, e-readers, media boxes, fridges, etc



Based on the Linux kernel version 2.6

Newer versions based on Linux kernel version 4.4 to 4.14



Written in

C/C++ for the core and kernel
Java for the UI and apps

VERSIONS

- Cupcake (1.5), Donut (1.6)
- Eclair (2.0), Froyo (2.2), Gingerbread (2.3)
- Honeycomb (3.0)
- Ice Cream Sandwich (4.0), Jelly Bean (4.1), KitKat (4.4)
- Lollipop (5.0)
- Marshmallow (6.0)
- Nougat (7.0)
- Oreo (8.0)
- Pie (9.0)
- Android Q (10.0)
- Red Velvet Cake (11)
- Snow Cone (12)
- Tiramisu (13)

ARCHITECTURE

- Android's main hardware platform is the ARM architecture
 - Various versions of ARM (eg ARMv6, ARMv7, ARMv8)
 - Versions are typically not compatible with each other
- In 2012 Intel-based devices began to appear
- Since Lollipop (5.0), 64-bit and 32-bit variants for mobile platforms are supported

COMPILATION

- From KitKat, the memory requirement is less than previously
- This is due to the change in the runtime environment
 - From Dalvik: compiled apps Just In Time (JIT)
 - To Android Runtime (ART): compiled Ahead of Time (AOT)
 - AOT compilation which reduces startup time and naturally memory usage

CAMERA



Initially Android required an autofocus camera



This was relaxed to a fixed-focus camera

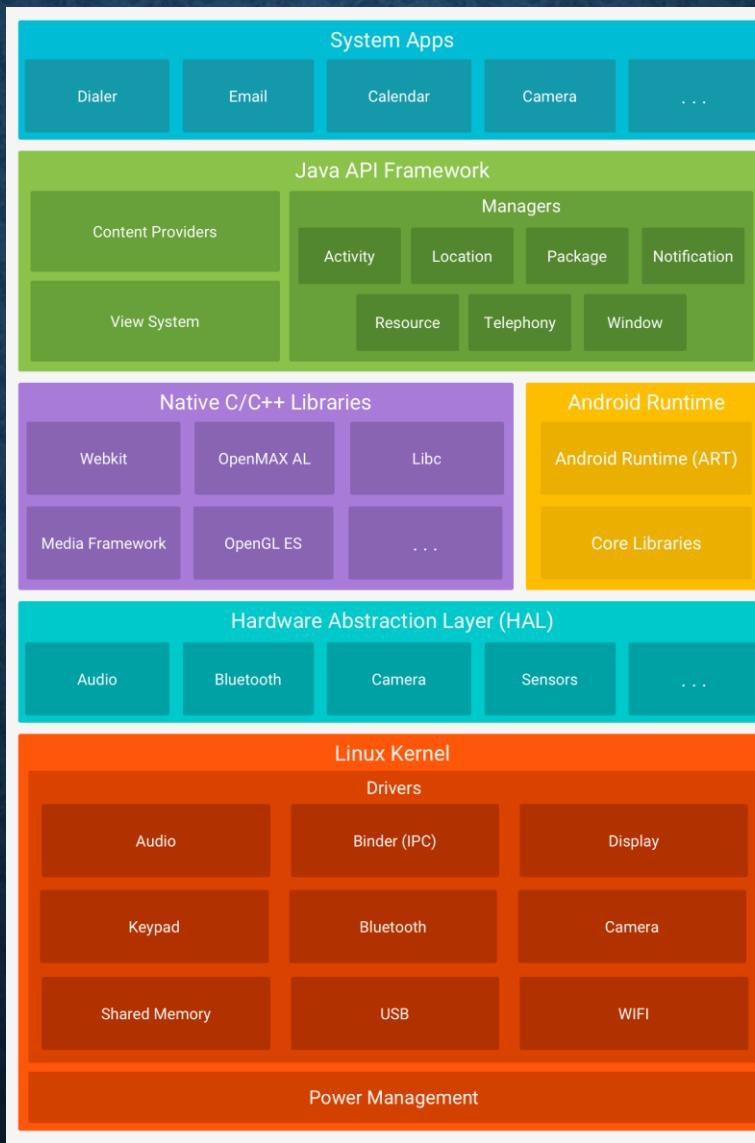


Later dropped entirely for devices other than smartphones.

SOFTWARE STACK

- Android software stack is categorised into
 - Linux kernel and drivers (C/C++)
 - Hardware Abstraction Layer (HAL) (C/C++)
 - Native libraries (middleware) (C/C++)
 - Android Runtime (C++ for ART and C for Dalvik)
 - Application/API Framework (Java)
 - Applications (Mostly Java, but other languages possible such as C/C++)

SOFTWARE STACK



SOFTWARE STACK - APPS

- Two categories, system and user apps
- System Apps
 - Come default installed with Android
 - Can sometimes not be uninstalled
 - Provide basic functionality
 - Examples: phone, contacts, clock, calendar, camera, etc
- User Apps
 - Written by some third-party developers
 - Can be downloaded and installed from the Google Play store
 - Examples: Chrome, Firefox, Google Maps, Dropbox, Banking apps, games, etc

SOFTWARE STACK – API FRAMEWORKS

- API Framework comprises
 - Content Providers
 - Allows applications to publish and share data with other applications
 - View System
 - Set of views used to create application user interfaces
 - Managers
 - 7 in total (discussed next)

SOFTWARE STACK – MANAGERS



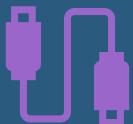
Activity Manager

Controls the application lifecycle and activity stack



Location Manager

Allows the application to receive updates about location changes



Package Manager

Enables an application to find information about other application installed on the device



Notification Manager

Allows applications to display alerts and notifications to the user

SOFTWARE STACK – MANAGERS



Resource Manager

Provides access to non-code embedded resources

These include strings, colour settings, user interface layouts, translations, etc



Telephony Manager

Telephony services available on the device

Status and subscriber information is provided to the application



Window Manager

Manages the z-ordered list of windows

Which windows are visible
Window layouts

BUILDING BLOCKS

- Fundamental components of Android
 - Activity
 - View
 - Intent
 - Services
 - Content Providers
 - Fragments
 - Manifest

BUILDING BLOCKS

- **Activity**
 - A class which represents a single screen
- **View**
 - An UI element (button, label, etc) that you see
- **Intent**
 - Used to invoke components, such as
 - Start the service
 - Launch an activity
 - Display a web page
 - Display a list of contacts
 - Broadcast a message
 - Dial a phone call

BUILDING BLOCKS

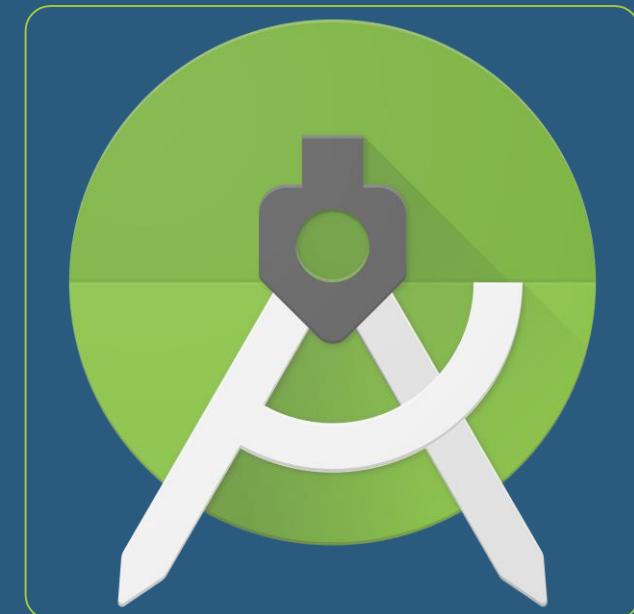
- Services
 - Runs in the background without direct user interaction
 - It tells the system what the application wants to do in the background
 - Enables the application to expose some of its functionality to other applications
- Content Providers
 - Are used to share data between applications
- Fragments
 - Are parts of an activity
 - Fragments can be displayed by an activity on the screen at the same time
- Android Manifest
 - Contains information about activities, content providers, permissions etc
 - Written in XML (AndroidManifest.xml)

PROGRAMMING LANGUAGE

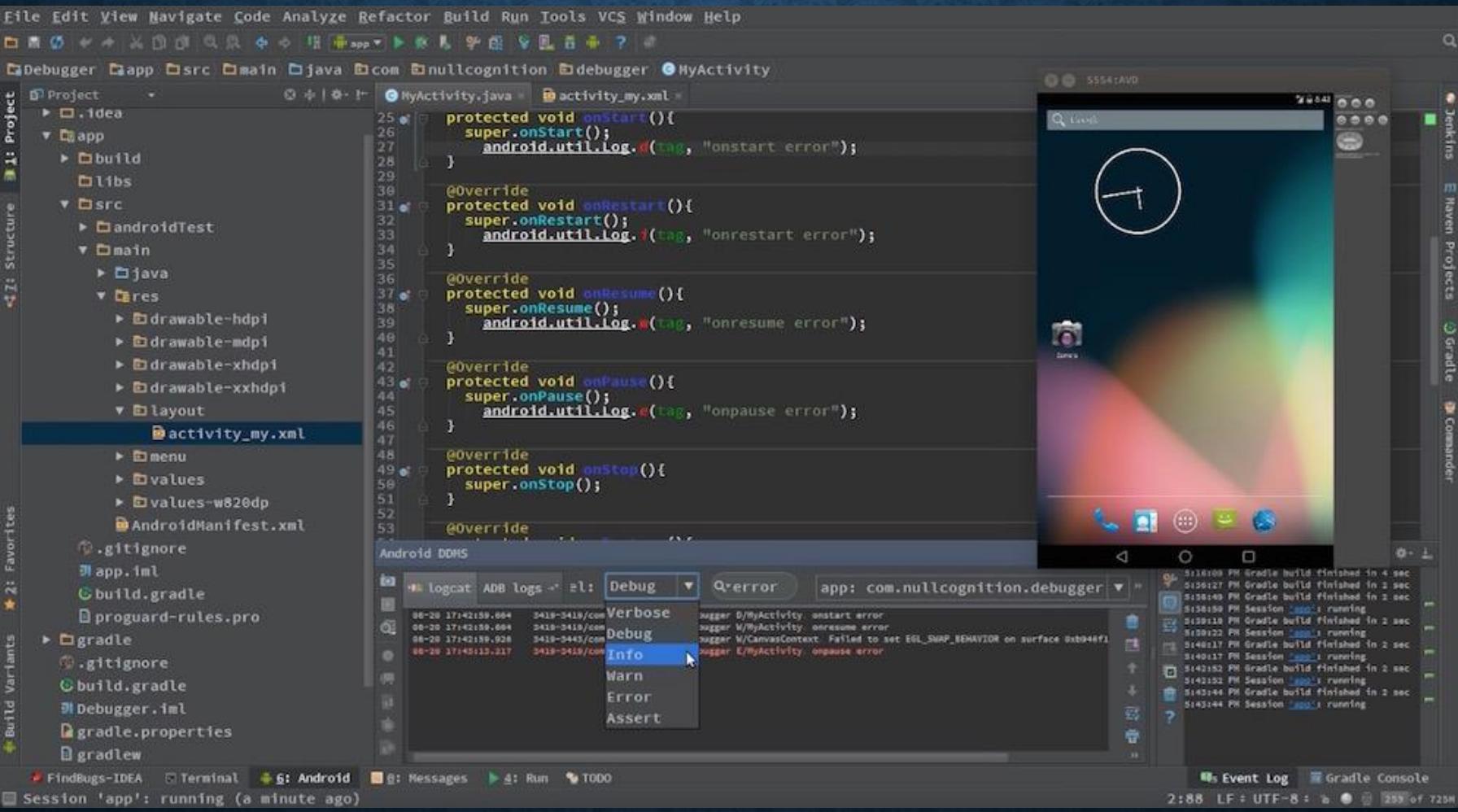
- To program an Android application knowledge of Java is required
- It is advisable to understand XML as well
- Other programming languages are supported
 - Often do not have access to all the nice features and libraries in Java
 - Often requires calling certain Java code through language bindings
 - Examples: C, C++, Python

PROGRAMMING ENVIRONMENT

- Many possible ways to write and test Android apps
- The recommended way
 - IDE: Android Studio (code editor, debugger, and compiler)
 - SDK: Android SDK (all the frameworks and libraries available in Android)
 - Emulator: Android Emulator (test your app in an emulated Android VM on your computer). Can emulate a range of different Android devices
 - OS Images: Image files (eg ISO) containing the entire Android OS. The image is loaded using the emulator. You can have different images with different Android versions/flavours
 - Physical Device (optional): Test on your own Android phone



PROGRAMMING ENVIRONMENT



PROGRAMMING ENVIRONMENT

System requirements

Windows

- Microsoft® Windows® 7/8/10 (32- or 64-bit)
- 3 GB RAM minimum, 8 GB RAM recommended; plus 1 GB for the Android Emulator
- 2 GB of available disk space minimum, 4 GB Recommended (500 MB for IDE + 1.5 GB for Android SDK and emulator system image)
- 1280 x 800 minimum screen resolution

Mac

- Mac® OS X® 10.10 (Yosemite) or higher, up to 10.13 (macOS High Sierra)
- 3 GB RAM minimum, 8 GB RAM recommended; plus 1 GB for the Android Emulator
- 2 GB of available disk space minimum, 4 GB Recommended (500 MB for IDE + 1.5 GB for Android SDK and emulator system image)
- 1280 x 800 minimum screen resolution

Linux

- GNOME or KDE desktop
Tested on Ubuntu® 14.04 LTS, Trusty Tahr (64-bit distribution capable of running 32-bit applications)
- 64-bit distribution capable of running 32-bit applications
- GNU C Library (glibc) 2.19 or later
- 3 GB RAM minimum, 8 GB RAM recommended; plus 1 GB for the Android Emulator
- 2 GB of available disk space minimum, 4 GB Recommended (500 MB for IDE + 1.5 GB for Android SDK and emulator system image)
- 1280 x 800 minimum screen resolution

FIRST APP

- Follow the instructions at
<https://developer.android.com/training/basics/firstapp/index.html>

