International Journal of General Engineering and Technology (IJGET)
ISSN(P): 2278-9928; ISSN(E): 2278-9936
Vol. 6, Issue 4, Jun - Jul 2017; 1 - 6
© IASET

International Academy of Science, Engineering and Technology
Connecting Researchers; Nurturing Innovations

# ANDROID: THE ARCHITECTURE AND APPLICATION ENVIRONMENT

**ANAND**

Assistant Professor, Department of Computer Science, Keshav Mahavidyalaya,

University of Delhi, New Delhi, India

## ABSTRACT

This article presents an overview of the Android OS, covering a detailed discussion on its architecture and framework. The author has compiled a detailed description of the Android to facilitate a broader understanding of its various features and its internal composition. The significant reach of the Android in the various computing devices and its enhanced usage by a variety of people, professional and hobbyists presents a need for a holistic description of its environment for the end users. Hence the article presents a wide ranging discussion on the architecture, framework and the development environment of the Android.

**KEYWORDS:** Android OS, Android Structure, Android Architecture, Android Framework

## INTRODUCTION

Since the middle 20th Century the computers have seen a constant improvement and evolution in their functions and roles. The most exceptional feature has been the reduction in their size, as they have become smaller, they use very less power and now they are capable of performing more advanced, faster and accurate calculations. Apple's iPhone represents one such revolution in computing. It is also termed as a Smartphone which is a handheld computer. While Apple was the first company to make available these Smartphones to the reach of common people, the revolution is now spearheaded by the software manufacturers and computer companies. In this context, the Android OS appears as a competitor of Apple iPhone, which is developed by Google. It is stimulating that many members contribute to Android, such as handset manufacturers, mobile operators, software companies, semiconductor companies, and some commercial units too. Interestingly, Android has revolutionized the development of applications which can be run on the Android OS, using its Android Software Development Kit (driven through an open source license). The common non-proprietary techniques such as Java programming language, in combination with Extensible Markup Language (XML) are used for the development of Android applications, hence making it easy and open tool highly preferred by the developer community. Hall and Anderson (2009) in their article have compared different mobile operating systems and suggested the uniqueness of Android in facilitating all features of the Smart phones (and similar devices).

### What is Android?

Android is an open system which can be used freely by any person. It is a software which includes an Operating System (OS). The Android OS is a Linux kernel based OS hosting the Dalvik virtual machine (optimized for mobile devices) which runs the various android applications like instances of the virtual machine. Android includes a user interface, multimedia support, an application framework and also Java class libraries. It has found multiple uses in many devices such as desktop computers, laptops, games, cameras, virtual keyboards, Android TV user interface, android wearable and other android touch based devices. The benefits of its design that suits the touch screen input makes it useful for many PCs, tablets, and electronic devices. Android is a platform which supports many diverse applications which are made available through the Android Stores. The users may develop their own applications, they can install them and use them on the Android platform.

**The Interface**

The default user interface of Android is based on ‘direct manipulation’. It means that it is based on touch inputs and involves the real-world actions such as taping, swiping, pinching and reverse pinching which are used to manipulate the on-screen objects. It also contains a virtual keyboard and by using Bluetooth and USB many other physical devices (such as gaming devices and the physical keyboard, etc.) are also supported by the user interface. The response to the user input is immediate and it provides a fluid touch interface (which also include vibration capabilities of the device). The device boot to a home screen which contains widgets and apps. The widgets are live and they contain content ‘auto-update’ mechanisms. The apps (developed by third-party) are available on google play store (Android Market) which are useful to upgrade the home screen. The apps are available on an All Apps screen. The notifications are expandable from the top bar, making it suitable to use.

**Android Structure, Java and XML**

A Java class library is used to build applications for the Android software environment. It makes use of XML for interface design such as for controlling the layout and style of an application. To exemplify the connection, if Java defines the functionality of the button, then the XML is used to define its text, color and font etc.

**Table 1: Android Versions**

| Code Name | Version Number | API Level |
|---|---|---|
| No Codename | 1.0 | 1 |
| Internally known as Petit Four | 1.1 | 2 |
| Cupcake | 1.5 | 3 |
| Donut | 1.6 | 4 |
| Éclair | 2.0-2.1 | 5-7 |
| Froyo | 2.2-2.2.3 | 8 |
| Gingerbread | 2.3-2.3.7 | 9-10 |
| Honeycomb | 3.0-3.2.6 | 11-13 |
| Ice Cream Sandwich | 4.0-4.0.4 | 14-15 |
| Jelly Bean | 4.1-4.3.1 | 16-18 |
| Kitkat | 4.4-4.4.4 | 19-20 |
| Lollipop | 5.0-5.1.1 | 21-22 |
| Marshmallow | 6.0-6.0.1 | 23 |
| Noughat | 7.0-7.1.2 | 24-25 |
| O | 8.0 | 26 |

**The Android Architecture**

- **Linux Kernel:** Android uses the Linux kernel. The various versions of the Linux kernel (such as 3.4, 3.10 or 3.18) are used by Android devices depending on the actual device and its hardware. While it is said that the basics of the Android kernel are similar to Linux, there is still a debate on the other aspects such as Android does not qualify to be a Linux distribution. Android has made minimal modifications in the Linux kernel for its use, such as below:

- **Android Shared Memory (Ashmem):** It is a filed based shared memory which better support a low-memory device. The ashmem sub-system is shared memory allocator which is similar to POSIX SHM, but has a different behavior displaying a simple file based API (Application Programming Interface).
- **Binder:** Is an android specific IPC (inter process communication) mechanism and RPC (remote procedure call) system similar to DBus (in Linux).
- **Pmem:** Is process memory allocator, it is similar to ashmem but the difference being that it uses the physical contiguous memory. It needs to maintain a physical to virtual mapping, hence, the required process allows the pmem to hold the file descriptor until all other references are closed.
- **Logger:** It is a system logging facility, which is a kernel support for the ‘logcat’ command.
- **Wakelocks:** Are used for power management files. It holds the machine awake on a per event basis.
- **Oom Handling:** It kills the processes when available memory becomes low. It is also known as Viking killer. The information about memory levels and associated classes are written in it.
- **Alarm Timers:** It allows the user space to tell the kernel when to wake up. It supports Android alarm manager. POSIX alarm times are used to implement them functionally.
- **Timed Output/GPIO:** It allows changing a gpio pin and its automatic restoration after a specified timeout.
- **RAM_CONSOLE:** It allows to save the kernel print messages to buffer in RAM. In case of kernel panic these messages can be seen in the next invocation of the kernel.

- **The Dalvik Virtual Machine (DVM):** It is a virtual machine which runs the codes and applications written in Java. The actual programming is written in Java, which is translated to a format which runs on the DVM. Programs which are written in Java are translated to Dalvik bytecode and stored in.dex (Dalvik Executable) and.odex (optimized Dalvik Executable) files. The DVM is actually to remove the constraints of the platform such as slow CPU, relatively low RAM, OS without swap space and powered by a battery. The DVM is also designed to eliminate the constraint of multiple and independent processes by providing the separate address space and separate memory. This helps in enhancing the application security.
- **The Android Runtime (ART):** It is the successor of Dalvik which uses the .dex files but not the .odex files. This is aimed at the enhancement of performance of the applications. The application’s bytecode is translated by ART into the native instruction which are executed by the runtime environment of the particular device. The ART uses ahead-of-time compilation through a process of compiling the entire application into the native machine code when they are installed. This is different from the Dalvik which is based on trace-based just-in-time (JIT) compilation. The JIT optimizes the execution by constantly profiling the applications each time they run. The frequently executed segments of their bytecodes are compiled into the native machine code. On the other hand, as described above ART eliminates the Dalvik’s JIT compilation and therefore, enhances the performance and reduces the power consumption. It also increases the efficiency, faster execution of applications and improved memory. This allows for a better garbage collection and debugging mechanisms and high-level profiling of the applications.

- Libraries
  - The Standard C library of Android, Bionic, was developed by Google. It is a derivation of standard C BSD’s library code. Its benefits are smaller runtime footprint and optimized for a low-frequency CPU. The Android does not use either the GNU C libraries (as used on other standard Linux distributions) or uClibc. The X server is also not included in Android. The libraries are written in C or C++ and they offer capabilities which are similar to the application layer, with sitting on top of the kernel. Some major native libraries are Surface manager, media framework, Web kit, system C libraries, open GL ES libraries and SQLite.

## APPLICATION FRAMEWORK

The framework offers many interfaces which are used by the developers for diverse standards. There is no need to code each and every basic task. The application framework contains different entities as given below.

- **Activity Manager:** They are the UI components typically corresponds to one screen. They can be faceless, can be in a floating window and return a value. Manages activity life cycle of the applications.
- **Package Manager:** It keeps track of installed applications. The apps can communicate with other apps on the device.
- **Window Manager:** Manages main window that comprise applications.
- **Notification Manager:** Allows applications to display alerts.
- **Views:** They provide common user interface elements. They are given to create layouts (which includes components such as grids, lists and buttons).
- **Resource Manager:** Manages various types of resources used in different applications. Allows access to non-code embedded resources (such as strings, color settings, UI layout)
- **Content Provider:** They enable to share data across applications (such as photo gallery, phonebook). They also provide uniform APIs for querying; delete, update and insert row. Content is represented by the URI or MIME type.

## ANDROID DEVELOPMENT ENVIRONMENT

- **Android SDK:** The applications which enhance the functionality of the devices are written using an Android software development kit (SDK), which makes use of the Java programming language. It is called Java, Android Library. The advantage of using Java is developer’s familiarity with programming languages belonging to family C. Hence, Java may be combined with C/C++ making the development of apps easy and simple. The use of the Android SDK is mandatory for the developers; for it includes a comprehensive set of development tools. It contains all the packages, class libraries, and application packages needed for the Application development. Debugger, software libraries and handset emulators (based on QEMU), documentation, sample code, tutorial are also included in the SDK.

- **Integrated Development Environment (IDE):** Initially the applications were developed on Eclipse IDE, which used Android Development Tools (ADT) plugin; but in 2014 Google shifted to Android Studio, the newly released primary IDE (by Google) for Android application development which replaced Eclipse. The Android Studio is compatible for windows, Mac OSX and Linux. Its main features are Gradle-based build support, Lint tool for performance, usability and compatibility, pro-guard integration and app-signing capabilities. It has template based wizards to facilitate common Android designs. It also contains a rich layout editor which allows drag and drop UI components, and preview of the layouts on multiple screens. It also facilitates a built-in support for Google cloud platform allowing an integration with Firebase cloud messaging.
- **Android Virtual Device (Emulator):** It is a virtual device which runs on the development machine, it is also called an emulator. It allows to prototype, develop and test the Android applications. The emulator mimics the software and hardware features of the device such as a Smartphone but it cannot physically approve of the real phone calls. It provides navigation and control keys which can be used to generate events for the developed application. The AVDs help in better modeling of an actual device. The idea is to develop and test over an emulator. It has the following constituents:
  - **A Hardware Profile:** It helps in defining the hardware features of the virtual device.
  - **System Image Mapping:** It helps in deciding the features what version of the Android platform will work on the virtual device. A system image packed with SDK add-on is useful for the developers or a standard Android platform can be used.
  - **Options:** To control the screen dimensions and appearances, the AVD can also specify the emulator skin. An emulated SD card can also be defined for the AVD.
  - **A Storage Area:** On which he device's user data is stored (such as settings, installed application). An emulated SD card option is also provided.

## CONCLUSIONS

The Android software environment was released as an open source project to be the first open, complete and free platform which was created specifically for the mobile devices. The Android is preferred for its various advantages such as it supports 2D and 3D platforms, supports different languages and easier to adopt more than 100 languages. The Java supporting features of Android enable the developers to improve more features. Any hardware support is easily connected to the Android devices. The evolving benefits of new generation networks make Android preferable for faster web browsing and video calling facilities. Above all the open source framework allows the users to make changes in apps required for themselves. This benefits not only the developer community, but also the users of diverse interests, professionals and hobbyists who wish to use and integrate Android in their list of interests and work. Another significant potential of this open source project is to enable the developers to add improved features to the existing applications or even develop the new ones without waiting for the manufacturer's improvements in apps and devices. However, the risks involved are the possible exposures to various malicious codes. In case of a large number of devices getting infected by the malicious codes or virus, the project may suffer its goal. The growing popularity and usage of Android is likely to face associated risks. The Android community provides continuous updates and security patches to counter the security loopholes.

There are some disadvantages of Android such as its slow response as compared to the new generation and latest computational devices and OS. The slow response is encountered and observed when apps are opened in the different platforms. Second, as compared to other OS, it makes use of the processes very efficient, which makes the processor to heat up. Third, usually when the free Android apps are used, several advertisements are encountered in between the application is. It is because anyone can make the advertisement by inserting some logic in the app, it may also put the phone's information at risk.

In continuation of the above limitations, the rapid release of the new Android versions also demands attention of the developers. A closely connected community of developers and their interaction is important to keep attending to these demands. The rise in number of blogs, open source support among the community is positively related to keep managing the above demands. It is expected that as the older apps will be updated as per the new versions the Android will also mature as a project overall, offering multiple and endless new possibilities. As the development environment and tools of development will improve the development of apps will also enhance and become better.

## REFERENCES

1. Hall S.P. And Anderson E., Operating Systems for Mobile Computing, December 2009. Journal of Computing Sciences in Colleges, ISSN:1937-4771
2. Google's Android OS: Past, Present and Future. Posted 17 August 2011. Retrieved May 28, 2017. Retrieved from http://www.phonearena.com/news/Googles-Android-OS-Past-Present-and-Future_id21273.
3. Android Kernel Features. Retrieved on May 29 2017. Retrieved from http://elinux.org/Android_Kernel_Features.
4. Harrison A, Saxton J., Android Architecture. Retrieved on 10 May 2017. Retrieved from

   http://meseec.ce.rit.edu/551-projects/fall2015/1-3.pdf