International Journal of Engineering Research-Online
A Peer Reviewed International Journal
Articles available online http://www.ijoer.in

Vol.3., Issue.4., 2015
(July-Aug)

REVIEW ARTICLE

ISSN: 2321-7758

OVERVIEW OF ANDROID SYSTEM: EVOLUTION WITH STRUCTURE

SONU RANI
Research Scholar
Computer Science and Engineering, Singhania University, Pacheri Bari
Jhunjhunu (Raj.) India

SONU RANI

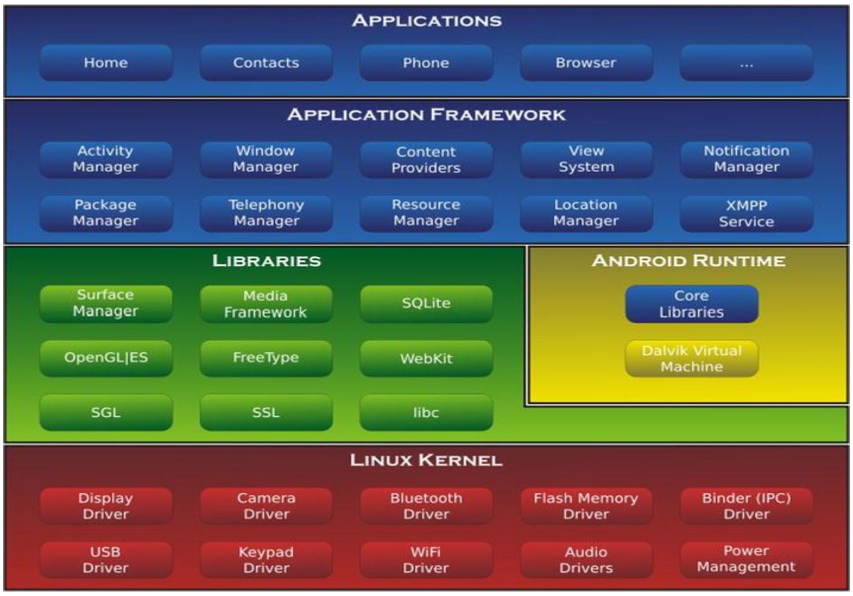
ABSTRACT

Due to the technological development, mobile terminals have evolved into functionally sophisticated devices such as smartphones and tablets. The Android platform has become one of the most popular operating system with millions of new users each year. Despite many studies, none have provided a comprehensive description of this operating system. In this paper we present a review of the Android OS. We describe the platforms history including improvements involved in each release as well as the systems architecture and project structure. It is mobile operating system that uses a modified version of the Linux kernel 2.6. Google developed Android as part of the Open Handset Alliance, a group of more than 30 mobile and technology companies working to open up the mobile handset environment.

KEY WORDS- Android OS, AOSP, ICS , IPC, ROM

I.INTRODUCTION

In this Proposed work, we will discuss about Android which is a software platform and operating system for mobile devices, based on the Linux kernel, and developed by Google and later the Open Handset Alliance. It allows developers to write managed code in the Java language, controlling the device via Google-developed Java libraries. Android is available as open source. Android is a freely downloadable open source software stack for mobile devices that includes an operating system, middleware and key applications based on Linux and Java. Google purchased the developer of Android in 2005, and Android was unveiled in 2007. Google released the Android code as open-source under the Apache License. Android has numerous developers writing applications (apps) all over the world. First of all the developers write their script in Java, and then download the apps from the third party sites or online stores. In February 2012, 450,000 apps were available for Android but the estimated number of downloads since December, 2011 was more than 10 billion. There are over 300 million Androids in use and over 850,000 devices activated every day. Android is the one of the most used mobile operating system with a market share of 48% and Over 400,000 applications available in Google play store. Android apps have been installed over 10 billion times and cover a vast range of categories from games and entertainment to financial and business services. Android software development and the Google Play Market are relatively open and unrestricted. This offers both developers and users more flexibility and freedom, but also creates significant security challenges.



Architecture of Android OS

(Source: http://link.springer.com/chapter/10.1007/978-1-4302-6131-5_1/fulltext.html)

II. The Open Handset Alliance

The Open Handset Alliance In November 2007, the Open Handset Alliance (OHA) was established by 34 founding members dedicated to development of open mobile standards, including Google, mobile device manufacturers, application developers, embedded systems developers, and commercialization companies. The goal of this alliance as described in the web site is as follows: OHA members are primarily mobile operators, handset manufacturers, software development firms, semiconductor companies and commercialization companies. Members share a commitment to expanding the commercial viability of open platform development.

OHA member companies back the open platform concept for a number of reasons, as follows:

- Lower overall handset costs: Opens up resources, which facilitates the focus on creating innovative applications, solutions and services.
- Developer-friendly environment: In the open-source community, developers share notes to expedite application development.
- Post-development: Provides an ideal channel for application marketing and distribution.

III. Android Open Source Project

Android is an open source software stack created for a wide array of devices with different form factors. The primary purposes of Android are to create an open software platform available for carriers, OEMs, and developers to make their innovative ideas a reality and to introduce a successful, real-world product that improves the mobile experience for users.

We also wanted to make sure there was no central point of failure, where one industry player could restrict or control the innovations of any other. The result is a full, production-quality consumer product with source code open for customization and porting.

The goal of the Android Open Source Project is to create a successful real world product that improves the mobile experience for end users. Android is designed and maintained with backward capability in mind. This means that new devices can run applications developed all the way back to Android's Cupcake (1.5). Official support for the Android SDK only goes back to Cupcake (1.5), so applications written for pre-Cupcake devices aren't guaranteed to run on the latest Android devices. Through the course of the AOSP, there have been many different versions of Android released for mobile devices. When new Android versions are released, mobile device owners are allowed to choose whether to upgrade their OS. With every iteration of Android, a new SDK is made available to developers and various new features are added to supported devices. Software developers need to stay mindful of the legacy features from previous versions when developing new applications.

Android 1.5 (Cupcake)

Salient features of Android 1.5 are listed below :

- Universal search box
- Revamped Android market : Browsing categories and filters (Top free, Top paid, Just in)
- Toggle between camera and videos modes, Video recording and playback in MPEG-4 and 3GP formats.
- Faster Camera start-up and image capture, integrated photo gallery.
- Much faster acquisition of GPS location (powered by SUPL AGPS).
- Auto-pairing and stereo support for Bluetooth added (A2DP and AVRCP profiles) .
- On-screen soft keyboard .
- Directly upload videos to youtube and picassa.

Android 1.6 (Donut)

Next version of Android, 1.6, had following enhanced features:

- Virtual On-screen keyboard .
- Camcorder mode for recording (and watching) video
- Stereo Bluetooth
- Expanded Gesture framework and new GestureBuilder development tool
- Home screen widgets and folders
- Quick Search Box and Voice Search
- Toggle between still and video captures modes, Support for WVGA screen resolutions Copy/Paste and search within the browser.
- Direct upload to Youtube and Picasa .
- User can select multiple photos for deletion.
- Battery usage indicator
- Updated technology support for CDMA/EVDO, 802.1x,
- VPNs, and a text-to-speech engine .
- Multilingual text-to-speech function.

Android 2.0 (Eclair)

Salient features of Android 2.0 are as follows:

- Multiple accounts for email and contact synchronization.
- Microsoft Exchange Support for syncing of e-mail
- Quick Contact pop-up widget
- Search saved SMS and MMS messages
- Camera improvement include support for flash and digital zoom, scene mode, white balance, color effect and macro focus
- Keyboard improvement : Adaptive Dictionary
- Bluetooth 2.1 support
- New browser User Interface and support for HTML5 Improved Google Maps 3.1.2
- Besides live wallpapers, it's allowing Home screen background images to be animated to show movement Motion Event class enhanced to track multi-touch events.

Android 2.0.1

- Minor API changes.
- Bug fixes and framework behavioral changes.

Android 2.1

- This version had minor amendments to the API and few bug fixes.
- Android Matures
- With the feature related to connectivity and enhanced mobile developments, android has also added growth in the technical aspect such as speed and performance increment.

Android 2.2 (Froyo)

- ➢ Multiple user account
- Speed, memory, and performance optimizations
- .Android market update : Batch and automatic updates, installing apps to the SD card- New tips widget for homescreen.
- Support for the Android Cloud to Device Messaging (C2DM) service
- File uploading in the browser.
- Improved Microsoft Exchange support .
- USB tethering and Hotspot support, Voice dialing and contact sharing over Bluetooth
- Multiple keyboard languages Adobe Flash 10.1.

Android 2.3 (Gingerbread)

- Native support for more sensors .
- UI refinements for simplicity and speed
- New keyboard for faster text input
- One-touch word selection and copy/paste .
- Near Field Communication (NFC), Improved power management .
- New Download Manager, Internet Calling.

- New audio effects such as reverb, equalization etc.
- Support for WebM/VP8 video playback, and AAC audio encoding.

Android 3.0 (Honeycomb)

Specifically optimized for tablets and devices with larger screen sizes Hardware acceleration, Support for multi-core processors, Ability to encrypt all user data Refined multitasking, rich notifications, home screen customization, widgets, Redesigned keyboard Support for video chat, Multiple browser tabs, form auto fill, new “incognito” mode allowing anonymous browsing Bluetooth tethering Built-in support for Media/Picture Transfer Protocol New two plane contacts and Email UI Ability to view albums and other collections in full-screen mode.

Android 4.0 (Ice Cream Sandwich)

Ice Cream Sandwich is the dessert-themed Android codename for version 4.0 of the open source Android mobile operating system. Ice Cream made its debut in October 2011 as Google’s “everywhere” operating system for smartphones, tablets and other mobile devices.

With Ice Cream Sandwich Google worked to unify the v2.x/v3.x forks of Android development while also adapting the Android framework and adding new APIs to help put an end to the issue of Android fragmentation. Ice Cream Sandwich also brought many of the tablet-centric features in Honeycomb (Android v3.0) to smartphones, including video chat support based on Google Talk protocols, a new System Bar for global status and notifications and an Action Bar for application control, tabbed Web browsing support, an optimized soft keyboard and a new email interface.Ice Cream follows a number of earlier updates for Android, including Cupcake (v1.5), Donut (v1.6), Eclair (v2.0),FroYo (v2.2), Gingerbread (v2.3) and Honeycomb (v3.0). The successor to Ice Cream Sandwich, Jelly Bean (v5.0) is expected to become available in late 2012.

Android 4.1 (Jelly Bean)

It is the dessert-themed Android codename for the 4.1 and 4.2 updates of the open source Android mobile operating system. Jelly Bean 4.1 debuted in June 2012 as the successor to the v4.0 "Ice Cream Sandwich" release, and was followed by Jelly Bean 4.2 in October 2012.

New additions in Android Jelly Bean include interface enhancements and improved overall responsiveness via "Project Butter," Google Now support, advanced natural language voice command capabilities akin to Apple's Siri, an improved Web browser with constant on-screen tabs, enhanced file management capabilities and better text input options.

Jelly Bean follows a number of earlier updates for Android, including all old version and involves advance features and merits .

Android 4.4 (KitKat)

KitKat is the dessert-themed Android codename for the 4.4 update of the open source Android mobile operating system. KitKat debuted in November 2013 as the successor to the "Jelly Bean" Android releases. The Android 4.4 update was widely expected to carry the Key Lime Pie codename, but Google instead decided to go with the iconic candy bar for the new update's codename.

Google first had to get permission to use the KitKat name from the Nestle. Nestle also released a limited edition KitKat bar with Android-decorated packaging in conjunction with the new mobile OS release.

Android 5.0 (Lollipop)

The most noticeable change in Android Lollipop 5.0 is an enhanced user interface called Material Design that utilizes bright colors, layering and subtle shadows to improve the overall user experience. The new Material Design interface is complemented by feature additions like multiple user support and a guest user account option, a new notification system, support for 64-bit CPUs, Project Volta for delivering battery life improvements, and more.

Continuity across Android devices is another key development in Android Lollipop. The new mobile operating system automatically transfers information and provides seamless integration between Android Lollipop smartphones, tablets, Android Wear and other mobile devices.

Lollipop follows a number of earlier updates for Android, including all versions and advance feature as compare others..

Android Release History

Version	*Name*	*Release Date*
1.5	Cupcake	April 2009

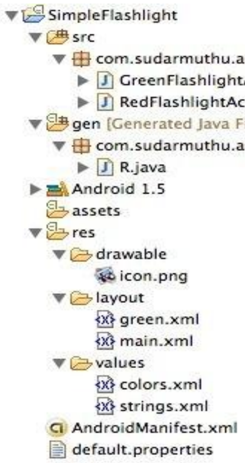
1.6	Donut	September 2009
2.0-2.1	Éclair	October 2009
2.2-2.2.3	Froyo	May 2010
2.3-2.3.7	Gingerbread	December 2010
3.0-3.2.6	Honeycomb	February 2011
4.0-4.0.4	Ice Cream Sandwich	October 2011
4.1-4.3.1	Jelly Bean	June 2012
4.4-4.4.4	KitKat	October 2013
5.0-5.0.2	Lollipop	November 2014

IV. Backgroud of Android OS

In order to develop an Android project, you have to install the Android plug in for Eclipse. In order to do that, choose (in the top menu) Help -> Install new software, and in the window that opens, insert the address: "https://dl-ssl.google.com/android/eclipse/". The installation wizard will guide you to download the full Android SDK if it is necessary.

A. Method of Basic Android project structure

After you have installed all the plug-ins necessary for the development of an Android file, you can now begin to develop an Android application. From the top menu, choose File -> Project, and from the "New Project window", choose "Android Project". Follow the project setup wizard and after finishing the wizard, you will have a basic Android application. Every Android project contains several folders:



```
SimpleFlashlight
  src
    com.sudarmuthu.android.flashlight
      GreenFlashlightActivity.java
      RedFlashlightActivity.java
  gen [Generated Java Files]
    com.sudarmuthu.android.flashlight
      R.java
  Android 1.5
  assets
  res
    drawable
      icon.png
    layout
      green.xml
      main.xml
    values
      colors.xml
      strings.xml
  AndroidManifest.xml
  default.properties
```

Figure- Basic Android project structure

Each Android project consists of key elements included in the root directory, as shown above figure

src: This folder contains the Java source files.

gen: Generated Java library, this library is for Android internal use only.

Res: Here we can store resource files such as pictures, XML files for defining layouts, and so forth. Within this folder there are additional folders such as *Drawable*, *Layout*, and *Values*.

Drawable: Here we store the various graphic files. We can see three types of drawable folders. This is because there are many Android devices with different screen resolutions. By default, there are several versions of this folder such as: *Drawable-mdpi*, *drawable-hdpi*, and so forth. This is required in order to adapt to different screen resolutions.

Layout: This is the place for XML layout files. Layout files are XML files which define how various Android objects (such as textboxes, buttons, etc.) are organized on the screen.

Values: XML files which store various string values (titles, labels, etc.).

B. Major files in the Android project:

AndroidManifest.xml: This is the Android definition file. It contains information about the Android application such as minimum Android version, permission to access Android device capabilities such as internet access permission, ability to use phone permission, etc.

MainLayout.xml: This file describes the layout of the page. This means the placement of every component (such as textboxes, labels, radio buttons, user defined components, etc.) on the app screen.

Activity class: Every application that occupies the entire device screen needs at least one class which inherits from the Activity class. One major method is called OnCreate. This method initiates the app and loads the layout page.

C. The Android manifest file-:***A typical Android. Manifest file looks like this***:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
xmlns:android="http://schemas.android.com/apk/re
s/android"
   package="com.firstproject"
   android:versionCode="1"
   android:versionName="1.0" >

   <uses-sdk android:minSdkVersion="7" />
```

```
  <application
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name" >
    <activity
      android:name=".MynewprojectActivity"
      android:label="@string/app_name" >
      <intent-filter>
        <action
android:name="android.intent.action.MAIN" />
                    <category
android:name="android.intent.category.LAUNCHER"
/>
      </intent-filter>
    </activity>
  </application>
</manifest>
```

Some important definitions in this file:

Android:versionCode: This is a numerical value. This is an important attribute because that is how the Android device knows when to alert the user to upgrade the application to a newer version.

Android:sdkMinVersion: Defines what is the earliest version of the Android operating system available to this application.

Activity element: Defines what the activities available in this application are and which activity has to be loaded at startup.

D. Layout file declaring components

The primary task of the manifest is to inform the system about the app's components. For example, a manifest file can declare an activity as follows:

A typical layout file looks like this:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/re
s/android"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent"
  android:orientation="vertical" >

  <TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/hello" />

</LinearLayout>
```

This layout defines one object, a TextView. This is a text label and it is used when a programmer wants to place a text title in your application.

E. Activity class

One major class is the Activity class. We already defined this class in the *android.manifest* file as the main activity class. Which is the first class to be loaded and executed right after the Android application will be launched:

```
package com.firstproject;

import android.app.Activity;
import android.os.Bundle;

public class MynewprojectActivity extends Activity {
  /** Called when the activity is first created. */
  @Override
  public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
  }
}
```

We can see here one major method called "onCreate" which is called when the Activity class is activated. In our specific example, we define a layout file by the line set Content View (R.layout.main)

V.CONCLUSION

The wide support from large companies, especially Google, have made Android one of the most important contestants in the mobile sector. The widespread and availability of smartphones and tablets allow manufacturers to modify the system in order to fit their needs, including both hardware and software layers. However, the inconvenient aspect of this platform, which is fragmentation, still remains. It takes a while for manufacturers to adopt a new version of Android to already released devices available on the market. Usually they do not provide a continuous support for all. Despite all difficulties, the release of a new version takes place roughly once a year. The development itself may broaden as more hardware and software companies get involved in the project. It is worth mentioning, that the platform is well supported by people outside the main Android project.

VI.REFERENCE

[1]. Kundu T.K., Paul K. (2010) *Android on Mobile Devices: An Energy Perspective*, CIT.

[2]. Lee W.M. (2011). *Beginning Android Application Development*. Wiley Publishing

[3]. Google. (n.d.). *Android Open Source Project*. Retrieved Sept 2011, from Android Open Source – Android Security Overview http://source.android.com/tech/security/index.html

[4]. Android Open Source Project. "Security Overview." Tech Info. N.p., 2012. Web. 18 June 2012. http://source.android.com/tech/security/index.html

[5]. Android Developers webpage, http://developer.android.com/index.html [access: 02.2015].

[6]. Google. (2012, April 28). What is Android. Retrieved April 28, 2012, from Android Developers: http://developer.android.com/guide/basics/what-isandroid.html

[7]. http://developer.android.com/tools/help/index.html

[8]. http://www2.dcsec.uni-hannover.de/files/android/p50-fahl.pdf

[9]. http://digitalforensicssolutions. com/papers/android-memory-analysis-DI.pdf

[10]. http://www.uandistar.org/2011/ 06/paper-presentation-on-android.html

[11]. http://www.webopedia.com/

[12]. http://www.tech2crack.com/history-android/

[13]. http://001mobileappdevelopment.blogspot. in/2012/01/trendsthat-impact-android-application.html