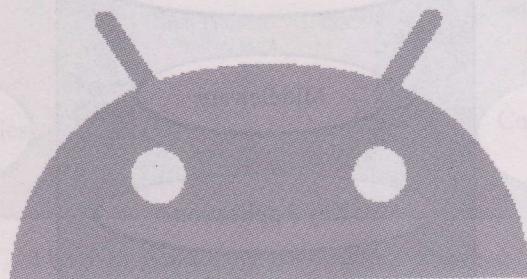


1

Introduction to Android Operating System



- What is Android?
- What is “Open Handset Alliance”?
- Android Ecosystem
- Why Android?
- Android Version
- Features of Android
- Architecture of Android
- Stack Overview

Chapter Overview:

In this chapter, a new learner will be able to understand the history of Android, the architecture, framework and various tools which are used in Android framework. You will also learn about the Android system overview with topics like Preferences and various supported File System in Android.

Operating Systems have developed a lot in last 15 years. It started from black and white phones to smart phones or mini computers. Nowadays one of the most widely used mobile OS is **ANDROID**.



Android Inc was founded in Palo Alto of California, U.S. by Andy Rubin, Rich Miner, Nick Sears and Chris White during 2003. After that it was acquired by Google in 2005.

1.1 What is Android?

- Android is a stack of software for mobile devices that are an Operating System, Middleware and Key Applications.

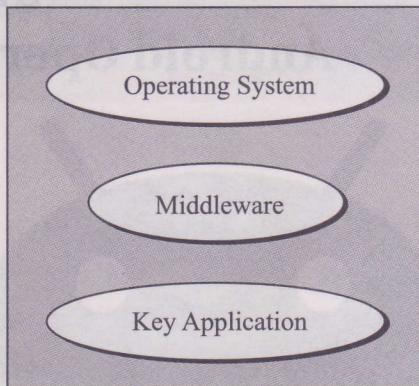


Figure 1.1: Android Operating System

- Android is a **Linux-based operating system** which is designed for **touchscreen** mobile devices like **smartphones and tablet computers**.
- It is an **open source** technology that allows the software to be freely modified and distributed by device manufacturers, wireless carriers and developers.
- Android was unveiled during 2007 along with the **founding of the Open Handset Alliance**.

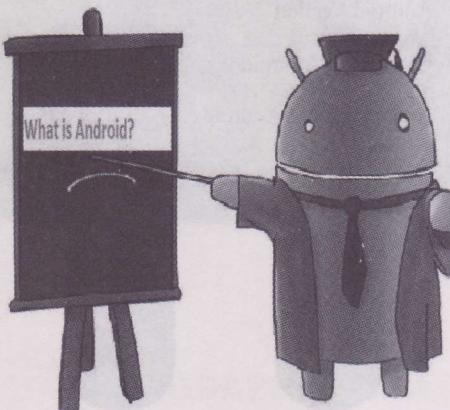


Figure 1.2: Android Logo

1.2 What is “Open Handset Alliance”?

Open Handset Alliance (OHA) was formed in November 2007. The OHA is the group that is in charge of the **Android smartphones operating system**. It was created by Google. The Open Handset Alliance (OHA) is a **business alliance** that consists of **47 companies** for developing open standard platform for



mobile devices. The members of OHA include handset manufacturers, chip makers, commercialization companies, software companies, and mobile operators.

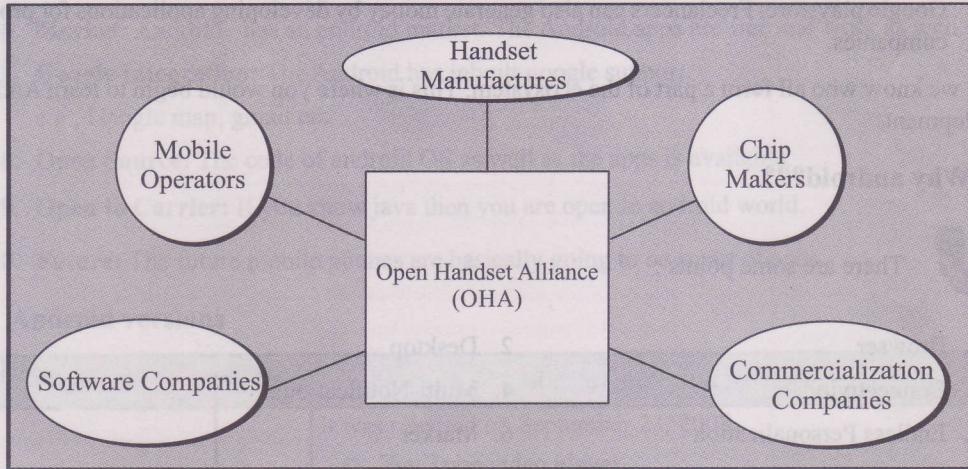


Figure 1.3 : Android Open Handset Alliance

1.3 Android Ecosystem

If we want to learn about Android and develop Android applications, it is very important for us to understand the entire ecosystem of Android and the stakeholders of this ecosystem.

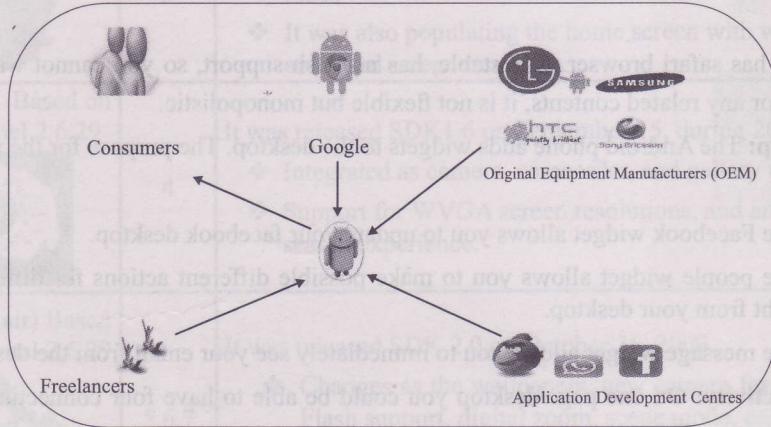


Figure 1.4: Android Ecosystem

The stakeholders are the **consumers that own Android devices**. But there are others as well:

- **Google:** It develops Android.
- **OEMs (Original Equipment Manufacturers):** They **manufacture the hardware**, and as well the custom application components?
- **Application Development Companies:** They are the **major contributors** to the ecosystem and employ Android developers, and also contract out the product development to services companies.



- **Freelance Android developers:** Developers have the skill-set to contribute to the ecosystem for Android development, **they are who create their own applications**, and publish them on **Google playstore**. Freelancers **can also generate money** by developing applications for product companies.

Now, we know who all form a part of the ecosystem. This is where you would begin to learn Android development.

1.4 Why android???



There are some points :

1. **Browser**
2. **Desktop**
3. **Connectivity**
4. **Multi-Notification**
5. **Endless Personalization**
6. **Market**
7. **Google Integration**
8. **Open Source**
9. **Open to Carrier**
10. **Future**

1. **Browser:** The **Android browser** is one of the best browsers on the mobile market. It generally loads pages faster than Safari or any other browser, has flash support and simply does everything a browser should do.

e.g.,

iPhone has safari browser. It is stable, has no flash support, so you cannot watch YouTube videos or any related contents, it is not flexible but monopolistic.

2. **Desktop:** The Android phone adds **widgets to the desktop**. The purpose for the widget

e.g.,

- ◆ The **Facebook widget** allows you to update your facebook desktop.
- ◆ The **people widget** allows you to make possible different actions for different contacts right from your desktop.
- ◆ The **message widget** allows you to immediately see your email from the desktop.

3. **Connectivity:** On one page/desktop you could be able to have **four connecting** device tool button.

- ◆ Turn on/off Bluetooth
- ◆ Turn on/off WiFi
- ◆ Turn on/off mobile network
- ◆ Turn on/off GPS

These buttons let you switch on/off instantly which will help you to **conserve battery life**.

4. **Multi-Notification:** **Android phones have multi-notification system.** With android the app have access to the notification system and call all report.

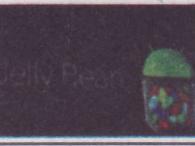


5. **Endless Personalization:** The Android cell phone allows client to configuration their mobile to look and behave exactly like they want.
6. **Market:** Android has an android market. The Android apps are free and work as well.
7. **Google Integration:** The Android has inbuilt google support.
e.g., Google map, gmail etc.
8. **Open Source:** The code of android OS as well as the apps is available.
9. **Open to Carrier:** If you know java then you are open to android world.
10. **Future:** The future mobile phones are basically going to be smart phones.

1.5 Android versions

Versions	APK	Release date and Activity
1.0	1	<p>It was released on September 23, 2008.</p> <ul style="list-style-type: none"> ❖ You Tube video player ❖ Wi-Fi and Bluetooth support ❖ Camera maintain this version
1.5(Cupcake)	3	<p>It was released on April 29, 2009.</p> <ul style="list-style-type: none"> ❖ Recording and watching videos in MPEG-4 and 3GP formats. ❖ It was also populating the home screen with widget and animated screen transition.
1.6 (Donut) Based on Linux Kernel 2.6.29	4	<p>It was released SDK1.6 on September 15, during 2009.</p> <ul style="list-style-type: none"> ❖ Integrated as camera, camcorder, and gallery interface. ❖ Support for WVGA screen resolutions, and an updated search experience.
2.0/2.1 (Eclair) Based on Linux Kernel 2.6.29	5,6,7	<p>It was released SDK 2.0 on October 26, 2009..</p> <ul style="list-style-type: none"> ❖ Changes as the wallpapers, new camera features that are:- Flash support, digital zoom, scene mode, colour effect etc. ❖ Improved typing speed on virtual keyboard, a smarter dictionary that learns from word usage
2.2 (Froyo) Based on 2.6.32	8	<p>It was released SDK 2.2 on May 20, 2009.</p> <ul style="list-style-type: none"> ❖ Changes as the integration of chrome's V8 java script engine into the browser app, voice dialling and contact sharing over Bluetooth, adobe flash support, speed implements through JIT compilation, USB tethering and Wi-Fi function.



2.3(Gingerbread) Based on 2.6.35		9/10	<p>It was released SDK 2.3 on December 6, during 2010.</p> <ul style="list-style-type: none"> ❖ Support as web M/VP8 video playback and AAC audio encoding, near field communication, and copy/paste functionality that select a word by press-hold, copy and paste.
3.0 (Honeycomb) Based on 2.6.36		14	<p>It was released as SDK 3.0 on February 22, during 2011.</p> <ul style="list-style-type: none"> ❖ This version focuses on tables, such as Motorola Xoom, the first tablet to be released. ❖ It improves multitasking. ❖ Supports multicore processor, hardware accelerations. ❖ It provides a 3D desktop with redesigned widget..
4.0 (Icecream Sandwich) Based on 3.0.1			<p>It was released SDK 4.0.1 on October 19, during 2011</p> <ul style="list-style-type: none"> ❖ SDK 4.0.1 as and 4.x successors unify the 2.3.x Smartphone and 3.x tablet SDKs. ❖ Include 1080P recording and a customizable launcher.
4.1(JellyBean)			<p>It was released SDK 4.1 on June 27, 2012.</p> <ul style="list-style-type: none"> ❖ Include as triple buffering, automatically resizable app widgets. ❖ Improved voice search and multichannel audio

1.6. Android Activity

	Activity
2001	search service for wireless device
2005	<ul style="list-style-type: none"> -Acquire Android (Andy Rubin: Danger of CEO, Development Sidekick of T-Mobile) -Acquire Skia (2D Graphics for mobile device) -Acquire RegWireless (Browser and Email for mobile device) -Move Engineers from PalmSource (Dianne Hack born, etc...)
2007 (Nov 5)	Android announced
2007 (Nov 12)	Android SDK is released by OHA
2007 (Dec 14)	Bug-fix SDK released



2008 (Jan 3)	Android Developer Challenge I starts accepting submissions
2008 (Feb 13)	m5-rc15 SDK released
2008 (Apr 14)	1788 is the total submissions for Challenge
2008 (May 12)	Top 50 Applications is announced in Challenge I
2008 (Nov)	Android Phone (G1 Phone by HTC/T-mobile)
2008 (Nov)	Full Source Open
2009 (Apr)	HTC Magic
2009 (July)	HTC Hero, Samsung i7500, Android Netbook , Set-top.....
2009 (Aug)	Android Developer Challenge

1.7. Features of Android

- 1. Open Source
- 2. Storage
- 3. Media support
- 4. Streaming media support
- 5. Multitouch
- 6. Web browser
- 7. Video calling
- 8. Multitasking
- 9. Accessibility
- 10. Voice based features
- 11. External storage

1. **Open Source:** Android is an open-source operating system. This way that the source code for Android is open to the public, **dissimilar iOS**, which is kept secret by Apple. This way that anyone can work on the operating system, not only one company developers. Therefore app developers for Android devices are able to implement extra features of their apps, due to the access they have to contain the source code. Again, these are only some of the unique features to Android. iOS also have many key abilities that are missing on Android devices. In the end, it is up to you to decide which operating system you would rather have. Anyone of you choose will be the right decision for you.

- 2. **Storage:** **SQLite**, a lightweight relational database, is used for data storage purposes.
- 3. **Media support:** Android supports the following audio/video/still media formats: WebM, H.263, H.264, AAC, HE-AAC (in 3GP or MP4 container), MPEG-4 SP, AMR, AMR-WB (in 3GP container), MP3, MIDI, WAV, JPEG, PNG, GIF, Ogg Vorbis, FLAC, BMP, WebP.
- 4. **Streaming media support:** **RTP/RTSP streaming** (3GPP PSS, ISMA), **HTML progressive download** (HTML5 <video> tag), **Adobe Flash Streaming** (RTMP) and **HTTP Dynamic**



Streaming are supported by the Flash plugin. Apple HTTP Live Streaming is support through RealPlayer for Android, and through the operating system for the duration Android 3.0 (Honeycomb).

5. **Multitouch:** Android has native support for multi-touch which was initially made available in handsets such as the HTC Hero. These features was originally disabled at the kernel level (possibly to avoid infringing Apple's patents on touch-screen technology at the time).Google has released an update version for the Nexus One and the Motorola Droid which enables multi-touch natively.
6. **Web browser:** The web browser available in Android is based on the open-source WebKit layout engine, attached with Chrome's V8 JavaScript engine. The browser scores 100/100 on top of the Acid3 test on Android 4.0.
7. **Video calling:** Android does not support local video calling, but several handsets have a customized version of the operating system that supports it, either through the UMTS network or ended IP. Video calling through Google Talk is available within Android 2.3.4 and later. Gingerbread allows Nexus S to place Internet calls with a SIP account. This allows for enhanced VoIP dialling to other SIP accounts and even phone numbers. Skype 2.1 offer video calling in Android 2.3, include front camera support. Users with the Google+ android app can video chat with other google+ users through hangouts.
8. **Multitasking:** Multitasking of application, with unique handling of memory allocation, will be existing.
9. **Accessibility:** Built in text to speech is provided by Talk back for people with low or no vision. Enhancements for people through hearing disabilities are available as is other aids.
10. **Voice based features:** Google search through voice has been available since opening release. Voice activities for navigation, calling, texting, etc. are supported on Android 2.2 forwards. As of Android 4.1, Google has expanded Voice Actions with the ability to talk back and read answers from Google's Knowledge Graph when queried with specific commands. The ability to control hardware have not yet been implemented.
11. **External storage:** Most Android devices include microSD slot and can read microSD cards formatted with FAT32, Ext3or Ext4 file system. To allow use of high-capacity storage media such as USB flash drives and USB HDDs, many Android tablets also include USB 'A' receptacle. Storage format with FAT32 is handled by Linux Kernel VFAT driver, As 3rd party solutions are necessary to handle other popular file systems such as NTFS, HFS Plus and ex FAT.

1.8 Android Architecture:

The Android OS can be referred to as a software stack of different layers, where every layer is a group of several programs components. It includes operating system, middleware and important applications. Every layer in the architecture provides different services.

Mostly Android has the following layers:

1. Applications
2. Application Framework



3. Libraries

4. Android Runtime

5. Linux Kernel

- When we are having **blue color** in architecture applications and framework it is written in **Java, execute in Dalvik**)
- When the color is **green** then it goes to c and C++ code libraries. If we want to communicate between java application and C, C++ libraries then we have used **JNI (java native interface)**.
- If the color is **yellow** then it is developed, which executes a **virtual machine** whose name is **dalvik virtual machine, native libraries, daemons and services (written in C or C++)**
- If the color is **red** it means it is a **linear kernel** which includes drivers for hardware, networking, file system access and inter-process communication.



Figure 1.5: Android Architecture

Applications:



Figure 1.6: Applications

All applications in the Application layer are written using Java Language. The various core applications that an Android device provides include:

- Email Client
- SMS program
- Maps



- ◆ Browser
- ◆ Calender
- ◆ Contacts
- ◆ Parallel running
- ◆ No compulsory applications

Application Framework:

Android, being open, offers developers the ability to build extremely rich and innovative applications. Developers are free to take benefit of the device hardware, access location information, run set alarms, background services, add notifications to the status bar, and much, much more.

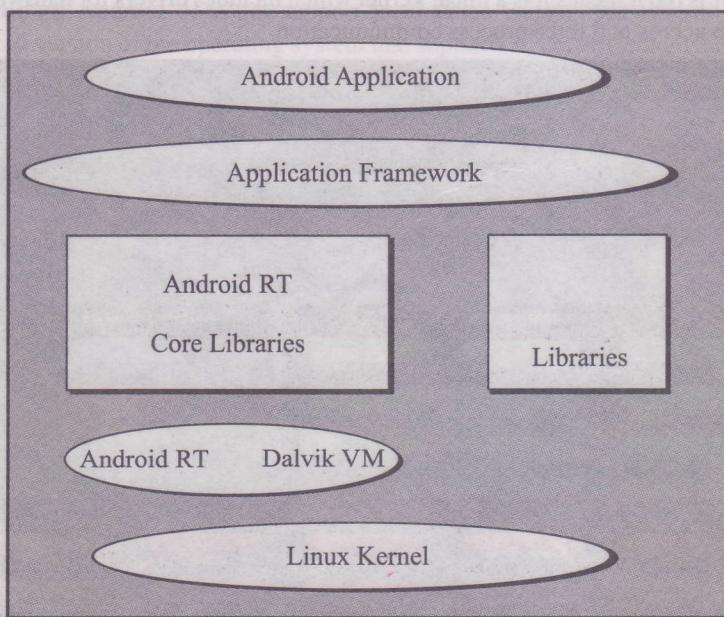


Figure 1.7: Android Application and Framework

Developers have full access to the same framework APIs used by the core applications. The application architecture is designed to simplify the reuse of components; any application can publish its capabilities and any other application may then make use of those capabilities. They can also be replaced by the user.

The set of services and systems include:

- ◆ A rich and extensible set of Views that can be used to build an application, including lists, buttons, grids, text boxes, and even an embeddable web browser
- ◆ Content Providers that allow applications to access data from other applications (such as Contacts), or to share their own data.
- ◆ A Resource Manager which provides access to non-code resources such as localized strings, graphics, and layout files.



- A Notification Manager that enables all applications to display custom alerts in the status bar
- An Activity Manager that manages the **lifecycle of applications** and provides a common navigation backstack



Figure 1.8: Application Framework

This layer is the second layer to the application layer. It contains **10 basic components**.

Following are the components this layer:

1. Activity Manager
 2. Window Manager
 3. View Manager
 4. Contents Manager
 5. Notification Manager
 6. Package Manager
 7. Telephone Manager
 8. Resource Manager
 9. Location Manager
 10. XMPP service Manager
- 1. Activity Manager:** This layer manages the **lifecycle of applications** and provides a common **navigation backstack**.
- 2. Window Manager:** As the name suggests it manages the **window surface**. Then it organizes the screen layout and locates the drawing surface and also performs other windows related jobs.
- 3. View Manager:** Views the window.
- 4. Contents Manager:** Enables applications to access data from other applications or to share their own data.
- 5. Notification Manager:** We get the notification from the system when the battery is low. If the programmer wants, he can also enable all applications to display customer alerts in the status bar.
- 6. Package Manager:** This manager handles other packages if it is installed in the machine. Sometimes we may download some application which is not our own. This manages other packages.
- 7. Telephony Manager:** Telephony manager is to handle the receiver call or voice calls.
- 8. Resource Manager:** This manager provides access to the non-code resources (Graphics, localized strings, and layout files).
- 9. Location Manager:** This manager helps to locate the mobile devices at which the location of the mobile device can be searched. Ex- Map, GPS or cell tower.
- 10. XMPP Service Manager:** Every mobile has some service like music application, browser, ringtone, services etc. **It is managed by service manager.**



Linux Kernel:

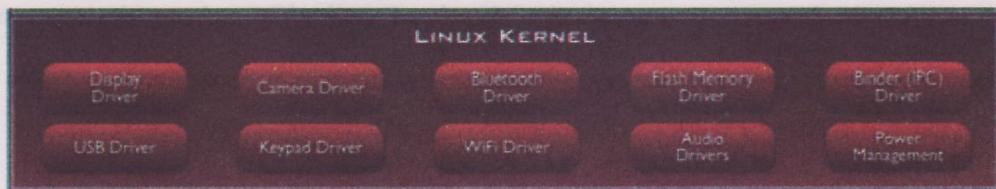


Figure 1.9: Linux Kernel

- The basic layer is the **Linux kernel**. The entire Android OS is built on top of the Linux 2.6 Kernel with the architectural changes made by Google.
- Here it uses Linux kernel for the process and the version is 2.6.
- It is highly trusted for years together for giving security giving better and networking etc.
- This layer has **10 major** components which were nothing but drivers.
- Android runtime gives **.dex** file which converted from the class file. .dex file goes to the linux kernel and calls the suitable drivers so that it can communicate with the corresponding drivers.
- Following are the drivers:
 - ◆ Display driver
 - ◆ Camera driver
 - ◆ Bluetooth driver
 - ◆ Flash memory driver
 - ◆ Binder driver(IPS)
 - ◆ USB driver
 - ◆ WiFi driver
 - ◆ Keypad driver
 - ◆ Audio driver
 - ◆ Power management driver

Libraries:

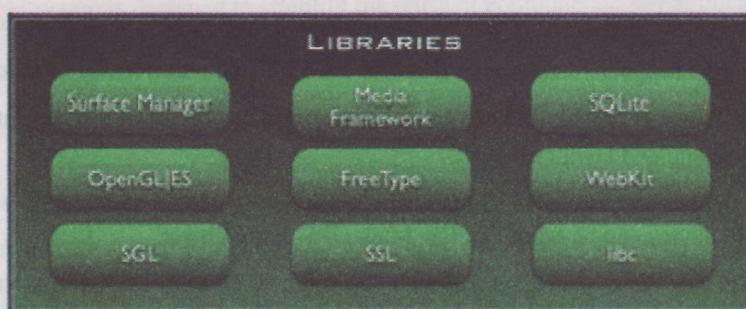


Figure 1.10: Libraries



- The 3rd layer of android architecture is the libraries layer. The color of this layer is green; it means it is written in C and C++ libraries.
- This layer contains 9 components. This layer is communicated through application framework layer. The communication happens through the Java Native Interface(JNI) package.
- This package of Java is represented for communication C and C++ language.
- Following are the 9 major components
 - ❖ Surface Manager
 - ❖ Open GLIES
 - ❖ SQLite database
 - ❖ Media Framework
 - ❖ SGL(Scalable graphics libraries)
 - ❖ Free Type
 - ❖ SSL(Secured socket layer)
 - ❖ Webkit
 - ❖ lib c(C libraries)

Surface Manager: We have seen in the application framework for following manager

1. Activity manager
2. Window manager
3. View system manager
 - ❖ It gives a frame, button, text field etc.
 - ❖ If we are setting the size for the frame, button, image (2D, 3D),
 - ❖ We have to interact with this manager called as surface manager.
 - ❖ This component handles all the surfaces rendered by each component of the frame.

Open GLIES: This component provides mostly 3D graphics libraries which have to be implemented by Java application and it is rendered by the application layer when they want some 3D graphics.

SQLite: When our application wants to connect the database then we can think of some popular RDBMS like mySQL, Oracle. But they are heavy weight database server as our applications are light weight. So it is better to have light weight RDBMS.

- ❖ So we can use SQLite which is an opensource RDBMS. It is not exactly tied with android.
- ❖ We can also use file system to store the data.



Media Framework: When we are using any mobile set, what comes our mind are

- ❖ Picture quality should be good.
- ❖ Multimedia action should good which include the sounds
- ❖ So to get those activities or application Android API has to call the media framework component to play and view the video-audio.

SGL(Scalable graphics libraries): It is responsible for implementing low level graphics by using JNI.

Free type: It is the responsible for to support the font quality, the image (bitmap images).

SSL(Secured socket layer): It stands for secure socket layer. It is mostly used in internal application. Suppose online through mobile we want to access our online through mobile then at that time we need some secure socket layer. It is also required in case of chat and message.

- It supports http and tcp protocol.
- There are some s/w company are available which are providing some 3rd party s/w to make the layer more secured and those s/w are:
 - Cyber cash
 - Gauntlet

Webkit: It is responsible for giving the browser supports. It also support well to following browser

- Google Chromes
- Apple Safari
- It also support JavaScript, jQuery, HTML, CSS, Ajax.

Lib c: It stands for C libraries .When we want to access some code C libraries headers then this components is useful.

Android Runtime:-

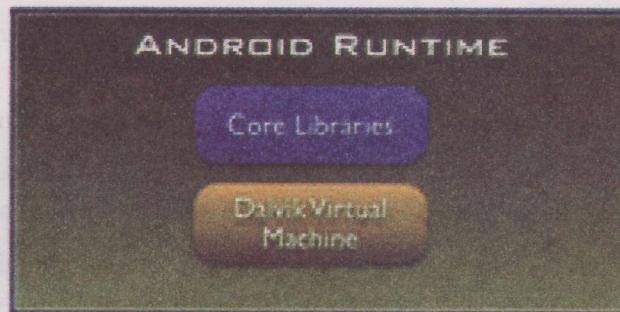


Figure 1.11: Android Runtime

Android Runtime consists of Dalvik Virtual Machine and Core Libraries.

DVM(Dalvik Virtual Machine)

- It is a type of JVM used in android devices to run apps and is optimized for low processing power and low memory environments.
- Unlike the JVM, the Dalvik Virtual Machine does not run .class files, instead it runs .dex files. .dex files are built from .class file at the time of compilation and provide higher efficiency in low resource environments.
- The DVM allows memory management, security, isolation, and threading support. It is also developed by Dan Bornstein of Google.

Core libraries:

These are different from Java SE and Java ME libraries. But these libraries provide most of the functionalities defined in the Java SE libraries.

- ✦ Data Structure
- ✦ File Access
- ✦ Network Access
- ✦ Utilities
- ✦ Graphics

1. 9. Stack Linux Kernel

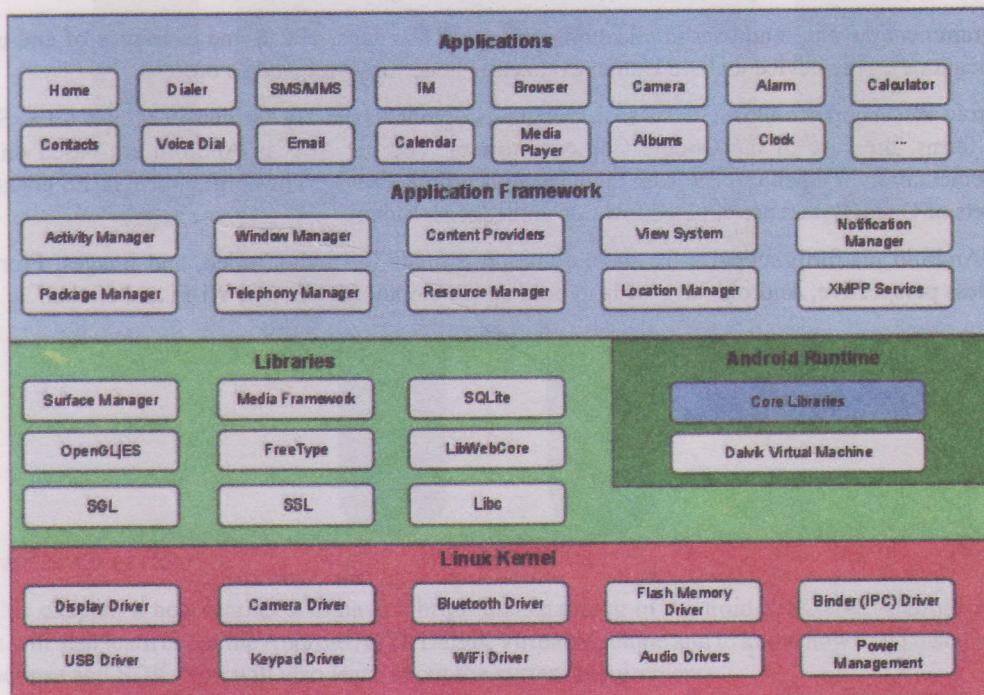


Figure 1.12: Stack Linux Kernel



The core Android platform has a Linux Kernel responsible for device drivers, resource access, power management, and OS duties. The complete device drivers include Display, Camera, Keypad, Audio, Wi-Fi, Flash Memory, and IPC (inter-process communication). Majority of the apps are developed in Java and run during the Dalvik VM.

Above the kernel are a number of C/C++ libraries such as OpenGL, WebKit, Free Type, Secure Sockets Layer (SSL), the C runtime library (libc), SQLite, and Media. The structure C library based on Berkeley Software Distribution (BSD) is tuned (to roughly half its original size) for surrounded Linux-based devices.

The media libraries are based on Packet Video's (www.packetvideo.com/) Open CORE. These libraries are responsible for recording and playback of audio and video formats. A library which is called Surface Manager Controls access to the display system and supports 2D and 3D.

More of these native libraries are likely to be added with new releases. The WebKit library is responsible for browser support; it is the same library that supports Google Chrome and Apple's Safari. The Free Type library is responsible for font support. SQLite (www.sqlite.org/) is a relational databases and not directly tied to Android. You can obtain and use tools meant for SQLite for Android databases as well.

Most of the application framework accesses these core libraries through the Dalvik VM, the doorway to the Android platform. Dalvik is optimized to run several instances of VMs. As Java applications access these core libraries, every application gets its own VM instance.

The Android Java API's chief libraries include telephony, resources, locations, UI, content providers (data), and package managers (installation, security, and so on).

Programmers develop end-user applications on top of this Java API. Some examples of end-user applications on the device include Home, Contacts, Phone, Browser ,and so on.

Android also supports Skia – a custom Google 2D graphics library written in C and C++. Skia also forms, the core of the Google Chrome browser. The 3D APIs in Android are based on an implementation of Open GL ES from the Khronos group(www.khronos.org). OpenGL ES contains subsets of OpenGL that are targeted towards embedded systems.

The Android platform supports the most common formats for audio, video, and images. From a wireless perspective, Android have APIs to hold up Bluetooth, EDGE, 3G, Wi-Fi and Global.
