

UNIT I - Introduction to Mobile Applications: Native and web applications - Mobile operating systems and applications - Mobile Databases. Android: History of Android - Android Features – OSS – OHA - Android Versions and compatibility - Android devices - Prerequisites to learn Android -- Setting up software – IDE - XML. Android Architecture: Android Stack - Linux Kernel - Android Runtime - Dalvik VM - Application Framework - Android emulator - Android applications.

1.1 Native and web applications

A **Native App** is an app developed essentially for one particular mobile device and is installed directly onto the device itself. Users of native apps usually download them via app stores online or the app marketplace, such as the Apple App Store, the Google Play store and so on. An example of a native app is the Camera+ app for Apple's iOS devices.

A **Web App**, on the other hand, are basically Internet-enabled apps that are accessible via the mobile device's web browser. They need not be downloaded onto the user's mobile device in order to be accessed. The Safari browser is a good example of a mobile Web app.

The Difference Between Native and Web Apps

In order to know which type of app is better suited to your needs, you need to compare each one of them. Here is a quick comparison between native apps and web apps.

User Interface

From the point of the mobile device user, some native and web apps look and work much the same way, with very little difference between them. The choice between these two types of apps has to be made only when you have to decide whether to develop a user-centric app or an application-centric app. Some companies develop both native and web apps, so as to widen the reach of their apps, while also provide a good overall user experience.

App Development Process

The app development process of these two types of apps is what distinguishes them from each other.

- Each mobile platform that the native app is developed for, stipulates its own unique development process. In the case of web apps running on a mobile device's web browser, the problem that arises is that each of these mobile devices have unique features and come with their unique problems as well.
- Every mobile platform uses a different native programming language. While iOS uses Objective-C, Android uses Java, Windows Mobile uses C++ and so on. Web apps, on the other hand, use languages such as JavaScript, HTML 5, CSS3 or other web application frameworks as per the developer's preferences.

- Each mobile platform offers the developer its own standardized SDK, development tools and other user interface elements, which they can use to develop their native app with relative ease. In the case of web apps, though, there is no such standardization and the developer has no access to SDKs or tools of any sort.

Accessibility

- A native app is totally compatible with the device's hardware and native features, such as an accelerometer, camera and so on. Web apps, on the other hand, can access only a limited amount of a device's native features.
- While a native app works as a standalone entity, the problem is that the user has to keep downloading updates. A web app, on the other hand, updates itself without the need for user intervention. However, it necessarily needs to be accessed via a mobile device's browser.

1.2 Mobile operating systems and applications

A mobile operating system (Mobile OS) is a software platform on top of which other programs called application programs, can run on mobile devices such as personal digital assistant (PDA), tablets, cellular phones, smartphones and so on. Over the years, Mobile OS design has experienced a three-phase evolution: from the PC-based operating system to an embedded operating system to the current smartphone-oriented operating system in the past decade. Throughout the process, Mobile OS architecture has gone from complex to simple to something in-between. The evolution process is naturally driven by the technology advancements in hardware, software, and the Internet.

1.2.1 Hardware

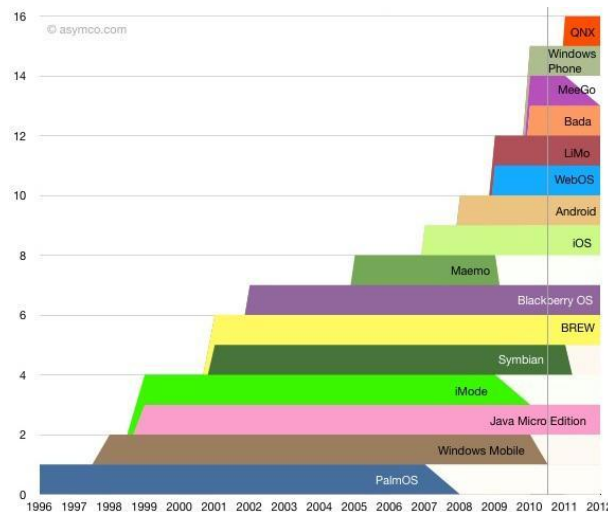
The industry has been reducing the factor size of microprocessors and peripherals to design actual mobile devices. Before the form factor size was reduced enough, the mobile device could not achieve both small size and processing capability at the same time. We had either a PC-sized laptop computer or a much weaker personal data assistant (PDA) in phone size. Mobile operating systems for PDAs usually did not have full multitasking or 3D graphics support. Features like sensors, such as accelerometers, and capacitor-based touch screens were not available in the past mobile operating systems.

1.2.2 Software

With a laptop computer, the software is mainly focused on the user's productivity, where support for keyboard and mouse that have precise inputs are essential. The software for a personal data assistant, as its name implies, helps the user to manage personal data such as contacts information, email, and so on. The mobile operating systems were not designed for good responsiveness or smoothness with a rich user interface (UI) including both touch screen and other sensors.

1.2.3 Internet

Along with Internet development, especially after Web 2.0, there is abundant information in the network waiting to be searched, organized, mined, and brought to users. People are increasingly living with the Internet instead of just browsing the Web. More and more people are involved in the development, including information contribution, application development, and social interactions. The mobile operating systems cannot be self-contained, but have to be open systems.



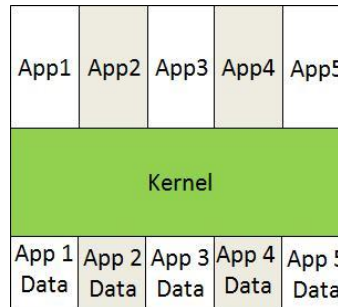
1.2.4 Mobile OS: Android

Android is an open source operating system for mobile devices developed by Google and the Open Handset Alliance. With 22,7% it is the second most used operating system for mobile devices worldwide behind Symbian. The system architecture consists of

- a modified Linux Kernel
- open source Libraries coded in C and C++
- the Android Runtime, which considers core libraries that disposes the most core functions of Java. As virtual machine it uses Dalvin, which enables to execute Java applications.

- an Application Framework, which disposals services and libraries coded in Java for the application
- and the Applications, which operate on it

Android, iOS and Windows Phone use the same model of application sandboxing, which is shown in below Fig. Each application owns a unique identity and any data, process or permission belongs to it. For example, the data assigned to one application identity has no access to any other data of another applications identity. This sandboxing model will have discussed closer in the following by using the example of Android, to underlay the understatement for it.



Android, iOS, Windows Phone Sandboxing Model

1.2.5 Mobile OS: BlackBerry OS

Blackberry OS is the proprietary mobile operating system, developed by the Canadian company Research in Motion and is used for Blackberry devices only. Instead of all the other regarded mobile operating systems, it is mainly developed for business usage. Gartner says that it is one of the most popular mobile operating system today with 16,0% market share, but they also predict a decreasing relevance in the future.

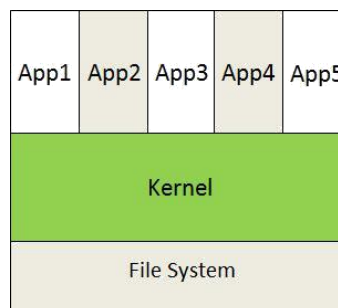


Fig. 4. BlackBerry OS Sandboxing Mode

BlackBerry OS uses an older model for application sandboxing, which can be seen in above Fig. It uses different trust roles for assignments and applications have full access to the complete device and data. It is also requiring to sign an application via Certificate Authorities (CA) or generated (selfsigned) certificate to run code on the device.

Furthermore, the signature provides information about the privileges for an application, which is necessary because applications have full access on BlackBerry devices, because of its sandboxing model.

1.2.6 Mobile OS: iOS

The proprietary mobile operating system iOS is only used for Apple devices like the iPhone and is a further development of the operating system Mac OSX. Its market share grew continuously over the last year to 15,7

The system architecture is identical to the Mac OSX architecture and consists of the following components:

- Core OS: The kernel of the operating system
- Core Services: Fundamental system-services, which are subdivided in different frameworks and based on C and Objective C. For example, offers the CF Network Frame work the functionality to work with known network protocols.
- Media: Considers the high-level frameworks, which are responsible for using graphic-, audio- and video technologies.
- Coca Touch: Includes the UIKIT, which is an Objective based framework and provides a number of functionalities, which are necessary for the development of an iOS Application like the User Interface Management

Like in the Android section mentioned, iOS uses a similar sandboxing model.

Furthermore, applications must be signed with an issued certificate. This ensures that application have not been manipulated and ensures the runtime to check if an application has not become untrusted since it was last used. Uneven Android applications, iOS applications can only be signed with an official certification

1.2.7 Windows Phone

- Windows Phone is a successor of the operating system Windows Mobile of the software developer Microsoft. By comparison to the other discussed mobile operating systems, the market share is low with only 4,2%.
- Windows Phone uses technologies and tools, which are also used in the station based application development, like the development environment Visual Studio and the Frameworks Silverlight, XNA and .NET Compact. Furthermore, Windows Phone considers a complete integration with the Microsoft Services Windows Live, Zune, Xbox Live and Bing.
- For sandboxing Windows Phone uses the same model like Android and iOS. Furthermore 3rd party applications can only be signed with an official certification, like iOS Application.

1.2.8 Mobile Applications

The diagram shows four basic apps (App 1, App 2, App 3 and App 4), just to give the idea that there can be multiple apps sitting on top of Android.

- Mobile applications (also known as mobile apps) are software programs developed for mobile devices such as smartphones and tablets. They turn mobile devices into miniature powerhouses of function and fun.
- Some devices come preloaded with some mobile apps courtesy of their manufacturers or the mobile service providers with which they're associated (for example, Verizon, AT&T, T-Mobile, etc.), but many more apps are available through device-specific app stores.
- These apps are like any user interface you use on Android; for example, when you use a music player, the GUI on which there are buttons to play, pause, seek, etc is an application.
- Similarly, is an app for making calls, a camera app, and so on. All these apps are not necessarily from Google.
- Anyone can develop an app and make it available to everyone through Google Play Store. These apps are developed in Java, and are installed directly, without the need to integrate with Android OS.

1.3 Mobile Databases

1.3.1 What is Mobile Database?

A mobile database is a database that can be connected to by a mobile computing device over a mobile network. And it is portable and physically separate from the corporate database server.

But mobile database is capable of communicating with that corporate database server from remote sites allowing the sharing of corporate database.

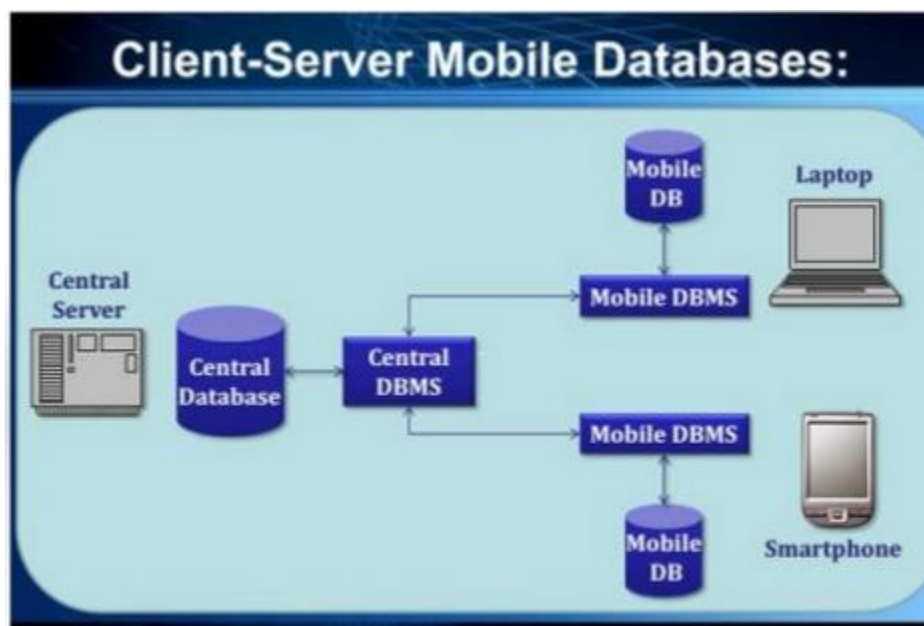
With mobile databases, users have access to corporate data on their laptop, PDA, or other Internet access device that is required for applications at remote sites.

1.3.2 The components of a mobile database environment

- **Corporate database server** and DBMS that deals with and stores the corporate data and provides corporate applications

- **Remote database** and DBMS usually manages and stores the mobile data and provides mobile applications
- **mobile database platform** that includes a laptop, PDA, or other Internet access devices
- Two-way communication links between the corporate and mobile DBMS.

Based on the particular necessities of mobile applications, in many of the cases, the user might use a mobile device may and log on to any corporate database server and work with data there, while in others the user may download data and work with it on a mobile device or upload data captured at the remote site to the corporate database.



The communication between the corporate and mobile databases is usually discontinuous and is typically established or gets its connection for a short duration of time at irregular intervals. Although unusual, some applications require direct communication between the mobile databases. The two main issues associated with mobile databases are the management of the mobile database and the communication between the mobile and corporate databases.

1.3.3 Features of mobile database

- Communicate with centralized database server through modes such as wireless or internet access
- Replicate data on centralized database server and mobile device
- Synchronize data on centralized database server and mobile device

- Capture data from various sources such as internet
- Manage/analyze data on the mobile device
- Create customized mobile applications

1.4 Android: History of Android

1.4.1 What is Android?

Android is a Linux based operating system it is designed primarily for touch screen mobile devices such as smart phones and tablet computers. Android is an operating system and programming platform developed by Google for Smartphone and other mobile devices (such as tablets). It can run on many different devices from many different manufacturers. Android includes a software development kit for writing original code and assembling software modules to create apps for Android users. It also provides a marketplace to distribute apps. Altogether, Android represents an ecosystem for mobile apps.

These applications are more comfortable and advanced for the users. The hardware that supports android software is based on ARM architecture platform. The android is an open source operating system means that it's free and any one can use it.

The android development supports with the full java programming language. Even other packages that are API and JSE are not supported. The first version 1.0 of android development kit (SDK) was released in 2008 and latest updated version is jelly bean.

Android: History of Android

The history and versions of android are interesting to know. The code names of android ranges from A to J currently, such as **Aestro, Blender, Cupcake, Donut, Eclair, Froyo, Gingerbread, Honeycomb, Ice Cream Sandwich, Jelly Bean, KitKat** and **Lollipop**. Let's understand the android history in a sequence.

1) Initially, **Andy Rubin** founded Android Incorporation in Palo Alto, California, United States in October, 2003.

2) In 17th August 2005, Google acquired android Incorporation. Since then, it is in the subsidiary of Google Incorporation.

3) The key employees of Android Incorporation are **Andy Rubin, Rich Miner, Chris White** and **Nick Sears**.

- 4) Originally intended for camera but shifted to smart phones later because of low market for camera only.
- 5) Android is the nick name of Andy Rubin given by coworkers because of his love to robots.
- 6) In 2007, Google announces the development of android OS.
- 7) In 2008, HTC launched the first android mobile.

Android Versions, Codename and API

Let's see the android versions, codenames and API Level provided by Google.

Version	Code name	API Level
1.5	Cupcake	3
1.6	Donut	4
2.1	Eclair	7
2.2	Froyo	8
2.3	Gingerbread	9 and 10
3.1 and 3.3	Honeycomb	12 and 13
4.0	Ice Cream Sandwich	15
4.1, 4.2 and 4.3	Jelly Bean	16, 17 and 18
4.4	KitKat	19
5.0	Lollipop	21
6.0	Marshmallow	23
7.0	Nougat	24-25
8.0	Oreo	26-27

1.5 Features of Android

Android is a powerful operating system competing with Apple 4GS and supports great features.

Few of them are listed below –

Beautiful UI Android OS basic screen provides a beautiful and intuitive user interface.

Connectivity GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, NFC and WiMAX.

Storage SQLite, a lightweight relational database, is used for data storage purposes.

Media support H.263, H.264, MPEG-4 SP, AMR, AMR-WB, AAC, HE-AAC, AAC 5.1, MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF, and BMP.

Messaging SMS and MMS

Web browser Based on the open-source WebKit layout engine, coupled with Chrome's V8 JavaScript engine supporting HTML5 and CSS3.

Multi-touch Android has native support for multi-touch which was initially made available in handsets such as the HTC Hero.

Multi-tasking User can jump from one task to another and same time various application can run simultaneously.

Resizable widgets Widgets are resizable, so users can expand them to show more content or shrink them to save space.

Multi-Language Supports single direction and bi-directional text.

GCM Google Cloud Messaging (GCM) is a service that lets developers send short message data to their users on Android devices, without needing a proprietary sync solution.

Wi-Fi Direct A technology that lets apps discover and pair directly, over a high-bandwidth peer-to-peer connection.

Android Beam A popular NFC-based technology that lets users instantly share, just by touching two NFC-enabled phones together.

1.6 OSS

OSS is IT for running a communication network. OSS is either Operational Support Systems or Operations Support Systems.

It is a set of programs that help a communications service provider for monitor, control, analyze, and manage a telephone or computer network.

OSS is software (Occasionally hardware) applications that support back-office activities which operate a telecommunication network, provision and maintain customer services

It is traditionally used by network planners, service designers, operations, architects, support, and engineering teams in the service provider. Increasingly product managers and senior staff under the CTO or COO may also use or rely on OSS to some extent.

OSS break down:

Operational/operations:

Relating to the day-to-day tasks of supplying and supporting communication services. Getting technical and infrastructure jobs done.

Running the network and services. As opposed to the business of selling, marketing or billing.

Support:

Enabling and improving the service provider's operational activities:

Automating operational tasks; executing them faster; making them consistent; and tracking progress/results.

Systems

One or more distinct software applications, that is responsible for doing specific OSS jobs, running on servers, or on devices installed in the network, or executed in the Cloud.

1.6.1 Functions of OSS

- Network management systems
- Service delivery
- Service fulfillment, including the network inventory, activation and provisioning
- Service assurance
- Customer Care

1.6.2 Advantages of OSS

- Support communications
- Support collaborations
- Control industrial process
- Savings through processes redesign
- Savings through processes integration
- Efficiently process business transactions
- Update corporate data base
- Centralized data repository for strategic decisions
- Competitive advantages through the ability of adapt business easier and faster.

1.7 OHA

The **Open Handset Alliance** (OHA) is a business alliance that was created for the purpose of developing open mobile device standards. The OHA has approximately 80 member companies, including HTC, Dell, Intel, Motorola, Qualcomm and Google. The OHA's main product is the Android platform - the world's most popular smartphone platform.

It was established on 5th November, 2007, led by Google. It is committed to advance open standards, provide services and deploy handsets using the Android Platform.

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 develop technologies that will significantly lower the cost of developing and distributing

mobile devices and services.

OHA devoted to advancing open standards for mobile devices.

Each Android Device Manufacturer of OHA partners can customize Android to suit their needs.

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.

Following are some Open Handset Alliance Firms.



1.8 Android Versions



1- Android 1.0 and 1.1(Unnamed)

Both versions are first commercial versions. They are officially released publicly in 2008 and 2009. The first android commercial version was placed on HTC dream device. These versions were released without codename. The features in

Android 1.0 :

- Google Maps
- Camera
- Gmail, Contacts and Google Synchronization
- Web Browser
- Wireless supports – WiFi and Bluetooth

Android 1.1 :

- Add Save attachment in message
- Provides reviews and details when user search business on maps

2- Android 1.5 (CupCake)

Android 1.5 was released in April 2009. This is first released codename with official name “**CupCake**”. It brought features in UI design and update several new features are.

- New upload service on YouTube and Picasa like Uploading Videos and Photos.
- Supporting in MPEG-4, Video recording
- Improving Web Browser-Copy and Paste facility

3- Android 1.6(Donut)

The Android Version 1.6, Codename is **Donut**. It was released in Sept 2009. It including various features

- It supports large screen size
- Providing Gallery and Camera features.
- Improve speed in system apps

4- Android 2.0 and 2.1(Eclair)

The codename for Android 2.0 is **Eclair**. It was brought in Oct 2009 and 2.1 version released in Dec 2009. There features:

- Update UI
- Support Live Wallpaper
- Support Bluetooth 2.1
- Improve Google map
- Minor API Changes

5- Android 2.2(Froyo)

It was released in May 2010 with the codename is **Froyo**. There features :

- Support Animated GIF
- WiFi Support Hotspot functionality
- Speed improvements
- Upload file support in browser
- Support numeric and alphanumeric password

6- Android 2.3 and 2.4 (Gingerbread)

Gingerbread came out on the market in December 2010. It was officially announced in Nexus S android phone which is Google co-developed with Samsung.

- Improve Copy-Paste Facility
- Updated UI design
- Social Networking Supports

- Easy use of keyboard

7- Android 3.0, 3.1 and 3.2 (Honeycomb)

Android 3.0 was released in February 2011 after that quickly followed by 3.1 and 3.2 in July and August of 2011.

- Gmail App improvements
- Updated 3D UI
- Media Sync from SD Card
- Google eBooks
- Google Talk Video Chat
- Support Adobe Flash in Browser
- High-performance WiFi Connections and Lock
- Chinese handwriting

8-Android 4.0 (Ice-Cream Sandwich)

Ice-Cream Sandwich was released in October 2011. It was Google's attempt to synthesize Honeycomb. There are some features

Improved text input and spelling check

- WiFi direct
- Photo Decor facility
- Improve in keyboard correction
- Face Lock
- Improve in video recording resolution
- Camera performance
- Up to 16 tabs in web browser

9- Android 4.1, 4.2 and 4.3 (Jelly Bean)

Android 4.1 came out in July 2012. Its codename is **Jelly Bean**. '**Google now**' is the main feature of Jelly Bean. It is used to whenever we want to search data from your google account and location data from your Android device to compile the information you need it. Many others features are

- Voice search
- Smooth UI
- Improve camera application
- Security enhancement
- Voice typing
- Multiple user accounts on tablet only
- 4k resolution support
- Supporting Bluetooth Low Energy
- Bi-directional text and other language support
- Support USB audio

- Lock screen improvement
- Set the volume of incoming calls and showing message alert
- Native emoji support

10- Android 4.4 (KitKat)

Android 4.4 as a name of **KitKat** announced by Google in September 2013. Features are:

- Screen Recording
- KitKat adds a feature in 'Google now'. Its name is '**OK Google**'. "OK Google" allows access google now to the user without touching your mobile phone.
- GPS Support
- Offline music support
- UI updates for google map navigation and alarm.
- Also, introduce 'Emoji' to google keyboard.

11- Android 5.0 and 5.1 (Lollipop)

Android 5.0 is called **Lollipop**. It was released in November 2014. Support ART (Android RunTime)

- Save battery on some device
- Improvement in UI
- New material design
- Bug fixes
- Multiple sim card support
- High definition voice call

12- Android 6.0 (Marshmallow)

Marshmallow came out in May 2015. It's new features are

- Fingerprint authentication
- USB Type C support
- Save battery-'Sleep Mode'
- App permission model-OPT(send request for permission)
- New Emoji's

13- Android 7.0 (Nougat)

Android **Nougat** was released in August 2016. It was announced with native split-screen mode and data saver feature.

- Provide multitasking
- Providing multi-window mode
- Improve in storage manager
- Display touch improvement

14- Android 8.0 (Oreo)

After some of the version released in the market recently released Android 8.0 named is **Oreo** in August 2017. There some updated new features

- Support PIP(Picture-in-Picture)
- Multi-display support
- Google Play support
- Adaptive icons
- Improve notification system

1.8.1 Android compatibility

Android is designed to run on many different types of devices, from phones to tablets and televisions. As a developer, the range of devices provides a huge potential audience for your app. In order for your app to be successful on all these devices, it should tolerate some feature variability and provide a flexible user interface that adapts to different screen configurations.

Because Android is an open source project, any hardware manufacturer can build a device that runs the Android operating system. Yet, a **device is "Android compatible"** only if it can correctly run apps written for the *Android execution environment*. The exact details of the Android execution environment are defined by the Android compatibility program and each device must pass the Compatibility Test Suite (CTS) in order to be considered compatible.

A app developer, need not worry, whether a device is Android compatible, because only devices that are Android compatible include Google Play Store. So we can rest assured that users who install your app from Google Play Store are using an Android compatible device.

However, we do need to consider whether our app is compatible with each potential device configuration. Because Android runs on a wide range of device configurations, some features are not available on all devices. For example, some devices may not include a compass sensor. If your app's core functionality requires the use of a compass sensor, then our app is compatible only with devices that include a compass sensor.

Android's purpose is to establish an open platform for developers to build innovative apps.

- The Android Compatibility program defines technical details of the Android platform and provides tools for OEMs to ensure developer applications run on a variety of devices.

- The Android SDK provides built-in tools for developers to clearly state the device features required by their applications.
- Google Play shows applications only to those devices that can properly run those applications.

To build an Android-compatible mobile device, follow this three-step process:

- Obtain the Android software source code. This is the source code for the Android platform that you port to your hardware.
- Comply with the Android Compatibility Definition Document (CDD) (PDF, HTML). The CDD enumerates the software and hardware requirements of a compatible Android device.
- Pass the Compatibility Test Suite (CTS). Use the CTS as an ongoing aid to evaluate compatibility during the development process.

After complying with the CDD and passing the CTS, your device is Android compatible, meaning Android apps in the ecosystem provide a consistent experience when running on your device.

Just as each version of the Android platform exists in a separate branch in the source code tree, there is a separate CTS and CDD for each version as well. The CDD, CTS, and source code are — along with your hardware and your software customizations — everything you need to create a compatible device.

1.9 Android devices

An Android device is a device that runs on the Android operating system. Android is an array of software intended for mobile devices that features an operating system, core applications and middleware.

An Android device may be a smartphone, tablet PC, e-book reader or any type of mobile device that requires an OS.

Within a short period, the Android platform became so popular that it surpassed Windows Mobile and Symbian for a number of applications. Various mobile device manufacturers embraced the Android platform due to its overwhelming popularity. The reasons behind this success are as follows:

- Cutting-edge technology offered by Google
- Extremely user friendly platform
- Can be used in smartphones as well as tablets
- Any user can do modifications to the platform as the Android SDK is open to users
- Availability of huge volume of applications

1.10 Setting up software

1.10.1 Set-up Java Development Kit (JDK)

You can download the latest version of Java JDK from Oracle's Java site – [Java SE Downloads](#). You will find instructions for installing JDK in downloaded files, follow the given instructions to install and configure the setup. Finally set PATH and JAVA_HOME environment variables to refer to the directory that contains **java** and **javac**, typically java_install_dir/bin and java_install_dir respectively.

If you are running Windows and installed the JDK in C:\jdk1.8.0_102, you would have to put the following line in your C:\autoexec.bat file.

```
set PATH=C:\jdk1.8.0_102\bin;%PATH%
set JAVA_HOME=C:\jdk1.8.0_102
```

Alternatively, you could also right-click on *My Computer*, select *Properties*, then *Advanced*, then *Environment Variables*. Then, you would update the PATH value and press the OK button.

On Linux, if the SDK is installed in /usr/local/jdk1.8.0_102 and you use the C shell, you would put the following code into your **.cshrc** file.

```
setenv PATH /usr/local/jdk1.8.0_102/bin:$PATH
setenv JAVA_HOME /usr/local/jdk1.8.0_102
```

Alternatively, if you use Android studio, then it will know automatically where you have installed your Java.

1.10.2 Android IDEs

There are so many sophisticated Technologies are available to develop android applications, the familiar technologies, which are predominantly using tools as follows

- [Android Studio](#)
- Eclipse IDE(Deprecated)

Android Studio Installation:

- 1) First of all, Download android studio from this link: <https://developer.android.com/studio/index.html>
- 2) JDK 8 is required when developing for Android 5.0 and higher (JRE is not enough). To check if you have JDK installed (and which version), open a terminal and type javac -version. If the JDK is not

available or the version is lower than 6, download it from this link.

1.10.3 To set up Android Studio on Windows

1. Launch the .exe file you just downloaded.
2. Follow the setup wizard to install Android Studio and any necessary SDK tools.

On some Windows systems, the launcher script does not find where Java is installed. If you encounter this problem, you need to set an environment variable indicating the correct location.

Select Start menu > Computer > System Properties > Advanced System Properties. Then open Advanced tab > Environment Variables and add a new system variable JAVA_HOME that points to your JDK folder, for example C:\Program Files\Java\jdk1.8.x.(where x is version number).

1.10.4 To set up Android Studio on Mac OSX

1. Launch the .dmg file you just downloaded.
2. Drag and drop Android Studio into the Applications folder.
3. Open Android Studio and follow the setup wizard to install any necessary SDK tools.

Depending on your security settings, when you attempt to open Android Studio, you might see a warning that says the package is damaged and should be moved to the trash. If this happens, go to System Preferences > Security & Privacy and under Allow applications downloaded from, select Anywhere. Then open Android Studio again.

If you need use the Android SDK tools from a command line, you can access them at:

```
/Users/<user>/Library/Android/sdk/
```

1.10.5 To set up Android Studio on Linux

1. Unpack the downloaded the Android SDK Manager in one of the following ways:
 - In Android Studio, click SDK Manager in the toolbar.
 - If you're not using Android Studio:
2. Windows: Double-click ZIP file into an appropriate location for your applications.
3. To launch Android Studio, navigate to the android-studio/bin/ directory in a terminal and execute studio.sh. You may want to add android-studio/bin/ to your PATH environmental variable so that you can start Android Studio from any directory.

4. Follow the setup wizard to install any necessary SDK tools.

Android Studio is now ready and loaded with the Android developer tools, but there are still a couple packages you should add to make your Android SDK complete.

3) The SDK separates tools, platforms, and other components into packages you can download as needed using the Android SDK Manager. Make sure that you have downloaded all these packages.

To start adding packages, launch the SDK Manager.exe file at the root of the Android SDK directory.

1.10.6 XML

XML stands for Extensible Markup Language. XML is a markup language much like HTML used to describe data. XML tags are not predefined in XML. We must define our own Tags. Xml as itself is well readable both by human and machine. Also, it is scalable and simple to develop. In Android we use xml for designing our layouts because xml is lightweight language so it doesn't make our layout heavy.

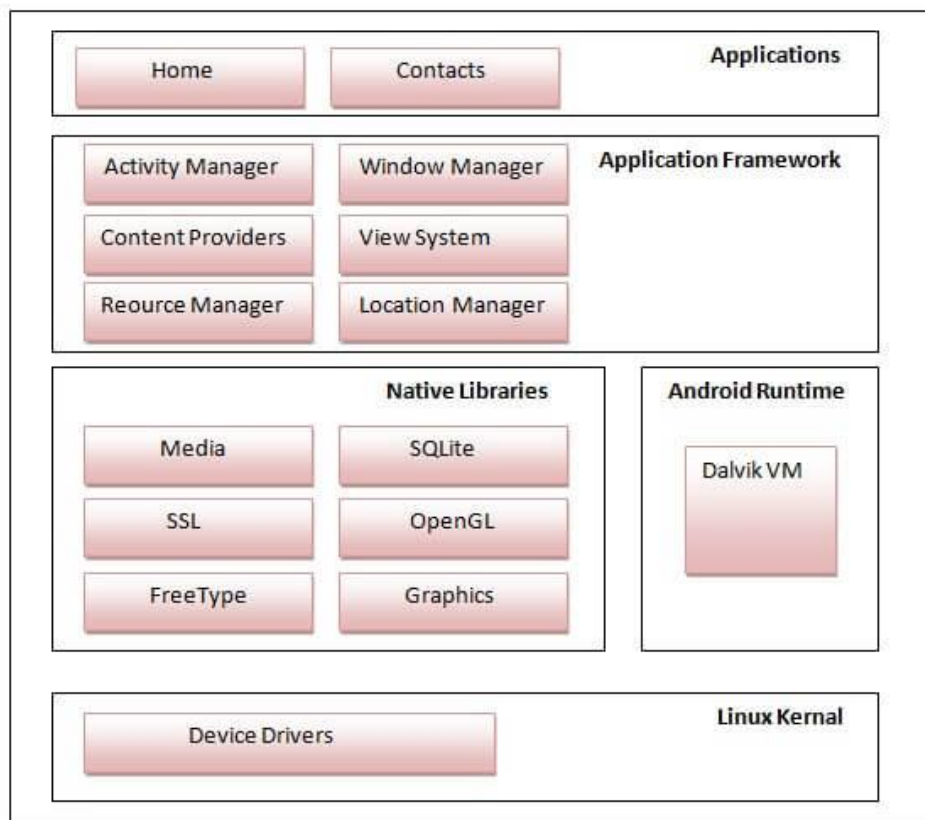
In this article we will go through the basic concepts of xml in Android and different XML files used for different purpose in Android. This will help you in writing a UI code to design your desired user interface.

The whole concept of Android User Interface is defined using the hierarchy of View and ViewGroup objects. A ViewGroup is an invisible container that organizes child views. These child views are other widgets which are used to make the different parts of UI. One ViewGroup can have another ViewGroup as a child element as shown in the figure given below:

1.11 Android Architecture

Android architecture or Android software stack is categorized into five parts:

1. Linux Kernel
2. Native Libraries (middleware),
3. Android Runtime
4. Application Framework
5. Applications



1) Linux kernel

It is the heart of android architecture that exists at the root of android architecture. **Linux kernel** is responsible for device drivers, power management, memory management, device management and resource access.

2) Native Libraries

On the top of linux kernel, there are **Native libraries** such as WebKit, OpenGL, FreeType, SQLite, Media, C runtime library (libc) etc.

The WebKit library is responsible for browser support, SQLite is for database, FreeType for font support, Media for playing and recording audio and video formats.

1. The Android runtime:

The Android runtime consists of core libraries of Java and ART (the Android RunTime). Older versions of Android (4.x and earlier) had Dalvik runtime.

2. Open GL (graphics library):

This cross-language, cross-platform application program interface (API) is used to produce 2D and 3D computer graphics.

3. WebKit:

This open source web browser engine provides all the functionality to display web content and to simplify page loading.

4. Media frameworks:

These libraries allow you to play and record audio and video.

5. Secure Socket Layer (SSL):

These libraries are there for Internet security.

3) Android Runtime

In android runtime, there are core libraries and DVM (Dalvik Virtual Machine) which is responsible to run android application. DVM is like JVM but it is optimized for mobile devices. It consumes less memory and provides fast performance.

This is the third section of the architecture and available on the second layer from the bottom. This section provides a key component called **Dalvik Virtual Machine** which is a kind of Java Virtual Machine specially designed and optimized for Android.

The Dalvik VM makes use of Linux core features like memory management and multi-threading, which is intrinsic in the Java language. The Dalvik VM enables every Android application to run in its own process, with its own instance of the Dalvik virtual machine.

The Android runtime also provides a set of core libraries which enable Android application developers to write Android applications using standard Java programming language.

4) Android Framework

On the top of Native libraries and android runtime, there is android framework. Android framework includes **Android API's** such as UI (User Interface), telephony, resources, locations, Content Providers (data) and package managers. It provides a lot of classes and interfaces for android application development.

The Android framework includes the following key services –

- **Activity Manager** – Controls all aspects of the application lifecycle and activity stack.
- **Content Providers** – Allows applications to publish and share data with other applications.
- **Resource Manager** – Provides access to non-code embedded resources such as strings, color settings and user interface layouts.
- **Notifications Manager** – Allows applications to display alerts and notifications to the user.
- **View System** – An extensible set of views used to create application user interfaces.

5) Applications

On the top of android framework, there are applications. All applications such as home, contact, settings, games, browsers are using android framework that uses android runtime and libraries. Android runtime and native libraries are using linux kernal.

Android applications

Android applications are usually developed in the Java language using the Android Software Development Kit.

Once developed, Android applications can be packaged easily and sold out either through a store such as **Google Play, Slide ME, Opera Mobile Store, Mobango, F-droid** and the **Amazon Appstore**.

Android powers hundreds of millions of mobile devices in more than 190 countries around the world. It's the largest installed base of any mobile platform and growing fast. Every day more than 1 million new Android devices are activated worldwide.

This tutorial has been written with an aim to teach you how to develop and package Android application. We will start from environment setup for Android application programming and then drill down to look into various aspects of Android applications.

Categories of Android applications

There are many android applications in the market. The top categories are

