The Android Platform

LEARNING OBJECTIVE

After studying this module student should be able to:

- Know History of mobile application development
- What Android is (and what it isn't)
- List different version of Android
- Describe Android SDK features
- Understand Android architecture
- Enumerate factors that affect Mobile Application development

INTRODUCTION

Android is an operating system that powers mobile devices and is one of the most popular mobile platforms today. Android platform runs on hundreds of millions of mobile devices throughout the world. It's the largest installed operating system of any mobile operating system and growing rapidly day by day.

You can develop apps and games using Android and it gives you an open marketplace for distributing your apps and games instantly.

Android is the operating system for powering screens of all sizes. Android version is named after a dessert. The latest version of android is Android 9.0 – Pie.

HISTORY OF MOBILE APPLICATION DEVELOPMENT

To understand what makes Android so convincing, you must study how mobile development has evolved and how Android differs from other mobile platforms.

The Motorola DynaTAC 8000X was the first commercially available cell phone and it is of brick size. First-generation mobiles were expensive, not particularly full featured and has Proprietary software.

As mobile phone prices dropped, batteries improved, and reception areas grew, more and more people began carrying these handy devices. Customers began pushing for more features and more games. They needed some way to provide a

portal for entertainment and information services without allowing direct access to the handset.

Early phone has postage stamp-sized low-resolution screens and limited storage and processing power, these phones couldn't handle the data-intensive operations required by traditional web browsers. The bandwidth requirements for data transmission were also costly to the user.

Wireless Application Protocol

The Wireless Application Protocol (WAP) standard emerged to address above concerns. WAP was stripped-down version of HTTP. WAP browsers were designed to run within the memory and bandwidth constraints of the phone. Third-party WAP sites served up pages written in a mark-up language called Wireless Markup Language (WML). The WAP solution was great for handset manufacturers and mobile operators. Phone users can access the news, stock market quotes, and sports scores on their phones.

WAP fell short of commercial expectations due to following reasons and Critics began to call WAP "Wait and Pay."

- Handset screens were too small for surfing.
- WAP browsers, especially in the early days, were slow and frustrating.
- Reading a sentence fragment at a time, and then waiting seconds for the next segment to download, ruined the user experience, especially because every second of downloading was often charged to the user.
- Mobile operators who provided the WAP portal often restricted which WAP sites were accessible.

Proprietary Mobile Platforms

Writing robust applications with WAP, such as graphic-intensive video games, was nearly impossible. Memory was getting cheaper, batteries were getting better, and PDAs and other embedded devices were beginning to run compact versions of common operating systems such as Linux and Windows. A variety of different proprietary platforms emerged and developers are still actively creating applications for them. Some of the examples of proprietary mobile platform are:

- Palm OS (now Garnet OS)
- RIM BlackBerry OS
- Java Micro Edition [Java ME]
- Binary Runtime Environment for Wireless (BREW)
- Symbian OS
- OS X iPhone

Each platform has benefits and drawbacks.

OPEN HANDSET ALLIANCE (OHA)

The Open Handset Alliance (OHA) was formed in November 2007, comprised of many of the largest and most successful mobile companies on the planet. Its members include chip makers, handset manufacturers, software developers, and service providers. The entire mobile supply chain is well represented.

In their own words, the OHA represents the following: “A commitment to openness, a shared vision for the future, and concrete plans to make the vision a reality, to accelerate innovation in mobile and offer consumers a richer, less expensive, and better mobile experience”

The OHA hopes to deliver a better mobile software experience for consumers by providing the platform needed for innovative mobile development at a faster rate and with higher quality than existing platforms, without licensing fees for either software developers or handset manufacturers.

THE ANDROID PLATFORM

Andy Rubin has been credited as the father of the Android platform. His company, Android Inc., was acquired by Google in 2005. Working together, OHA members, including Google, began developing a non-proprietary open standard platform based upon technology developed at Android Inc. that would aim to solve the problems hindering the mobile community. The result is the Android project.

Most Android platform development is completed by Rubin’s team at Google, where he acts as VP of Engineering and manages the Android platform roadmap. Google hosts the Android open-source project and provides online Android documentation,

tools, forums, and the Software Development Kit (SDK) for developers. All major Android news originates at Google.

What is android?

Android is called as "the first complete, open, and free mobile platform":

Complete: allows for rich application development opportunities.

Open: It is provided through open-source licensing.

Free: Android applications are free to develop. Android applications can be distributed and commercialized in a variety of ways.

Features of Android

- Free and Open Source
- Familiar and inexpensive development tools
- Freely available SDK
- Familiar Language, Familiar Development Environments
- Reasonable learning curve for developers
- Enabling development of powerful applications
- Rich, secure application integration
- No costly obstacles to publication
- Free "Market" for application
- A new growing platform

What it is not?

Android is not:

- *A Java ME implementation:* Android applications are written in the Java language, but they are not run within a Java ME virtual machine, and Java-compiled classes and executable will not run natively in Android.
- *Part of the Linux Phone Standards Forum or the Open Mobile Alliance:* Android runs on an open-source Linux kernel, but, while their goals are similar, Android's complete software stack approach goes further than the focus of these standards-defining organizations.
- *Simply an application layer (like UIQ or S60):* While Android does include an application layer, ''Android'' also describes the entire software stack

encompassing the underlying operating system, the API libraries, and the applications themselves.

- A mobile phone handset Android includes a reference design for mobile handset manufacturers, but there is no single ‘‘Android phone.’’ Instead, Android has been designed to support many alternative hardware devices.
- *Google’s answer to the iPhone:* The iPhone is a fully proprietary hardware and software platform released by a single company (Apple), while Android is an open-source software

THE ANDROID VERSION

Android is the operating system for powering screens of all sizes. Android version is named after a dessert. The latest version of android is Android 9.0 – Pie. Following table shows how the android platform evolves.

Android Version	Name	Feature	API Level
1.0	Alpha	Web browser, Camera, Synchronization of Gmail, Contact and Calendar, Google Maps, Google Search, Google Talk, Instant Messaging, Text Messaging and MMS, Media Player, Notification, Voice Dialer, YouTube Video Player Other applications include: Alarm Clock, Calculator, Dialer (Phone), Home screen (Launcher), Pictures (Gallery), and Settings.	1
1.1	Beta	The update resolved bugs, changed the Android API and added a number of features such as Details and reviews available when a user searches for businesses on Maps, Ability to show/hide dial pad and save attachments in messages.	2
1.5	Cupcake	Virtual keyboards with text prediction and user dictionary for custom words, widgets, video	3

Android Version	Name	Feature	API Level
		recording and playback, Bluetooth, Copy and Paste, animated screen transition, auto rotation, upload video on YouTube, upload photo to Picasa.	
1.6	Donut	Voice and text entry search, Multi-lingual speech synthesis, updated technology support for CDMA/EVDO, 802.1x, VPNs, and a text-to-speech engine, WVGA screen resolutions, Expanded Gesture framework and new Gesture Builder development tool	4
2.0 **2.0.1** **2.1**	Éclair	Customize your home screen just the way you want it. Arrange apps and widgets across multiple screens and in folders. Stunning live wallpapers respond to your touch.	5 6 7
2.2-2.2.3	Froyo	Voice Typing lets you input text, while Voice Actions allow you to control your phone, just by speaking.	8
2.3	Gingerbread	New sensors make Android great for gaming – so you can touch, tap, tilt and play away.	9-10
3.0	Honeycomb	Optimized for tablets.	11-13
4.0	Ice Cream Sandwich	A new, refined design. Simple, beautiful and beyond smart.	14-15
4.1-4.3	Jelly Bean	Fast and smooth with slick graphics. With Google Now, you get just the right information at the right time.	16-18
4.4	Kit Kat	A more polished design, improved performance and new features.	19-20
5.0	Lolipop	Get the smarts of Android on screens big and small with the right information at the right moment.	21-22
6.0	Marshmallow	New App Drawer, Doze mode, Native finger	23

Android Version	Name	Feature	API Level
		print support, Android pay, USB type-C and USB 3.1 support, Direct share.	
7.0	Nougat	Revamped notification, Split-screen use, file based encryption, direct boot, data saver	24-25
8.0	Oreo	Picture in picture, Google play protect, emoji	26-27
9.0	Pie	Adaptive Battery, adaptive brightness, intuitive navigation, dashboard, App timers, Wind down and do not disturb, Digital wellbeing.	28

Table-1 Android Versions

NATIVE ANDROID APPLICATIONS

Android phones will normally come with a suite of generic preinstalled applications that are part of the Android Open-Source Project (AOSP), including, but not necessarily limited to:

- An e-mail client
- An SMS management application
- A full PIM (personal information management) suite including a calendar and contacts list
- A Web Kit-based web browser
- A music player and picture gallery
- A camera and video recording application
- A calculator
- The home screens
- An alarm clocks

In many cases Android devices will also ship with the following proprietary Google mobile applications:

- The Android Market client for downloading third-party Android applications
- A fully-featured mobile Google Maps application including Street View, driving directions and turn-by-turn navigation, satellite view, and traffic conditions

- The Gmail mail client
- The Google Talk instant-messaging client
- The YouTube video player

ANDROID SDK FEATURES

The true appeal of Android as a development environment lays in the APIs it provides. As an application-neutral platform, Android gives you the opportunity to create applications that are as much a part of the phone as anything provided out of the box. The following list highlights some of the most noteworthy Android features:

- No licensing, distribution, or development fees or release approval processes
- Wi-Fi hardware access
- GSM, EDGE, and 3G networks for telephony or data transfer, enabling you to make or receive calls or SMS messages, or to send and retrieve data across mobile networks
- Comprehensive APIs for location-based services such as GPS
- Full multimedia hardware control, including playback and recording with the camera and microphone
- APIs for using sensor hardware, including accelerometers and the compass
- Libraries for using Bluetooth for peer-to-peer data transfer
- IPC message passing
- Shared data stores
- Background applications and processes
- Home-screen Widgets, Live Folders, and Live Wallpaper
- The ability to integrate application search results into the system search
- An integrated open-source HTML5WebKit-based browser
- Full support for applications that integrate map controls as part of their user interface
- Mobile-optimized hardware-accelerated graphics, including a path-based 2D graphics library and support for 3D graphics using OpenGL ES 2.0
- Media libraries for playing and recording a variety of audio/video or still image formats
- Localization through a dynamic resource framework

- An application framework that encourages reuse of application components and the replacement of native applications

ANDROID ARCHITECTURE

Android is an open source, Linux-based software stack created for a wide array of devices and form factors. The following diagram shows the major components of the Android platform.

The Linux Kernel

The foundation of the Android platform is the Linux kernel. For example, the Android Runtime (ART) relies on the Linux kernel for underlying functionalities such as threading and low-level memory management. Using a Linux kernel allows Android to take advantage of key security features and allows device manufacturers to develop hardware drivers for a well-known kernel.

Hardware Abstraction Layer (HAL)

The hardware abstraction layer (HAL) provides standard interfaces that expose device hardware capabilities to the higher-level Java API framework. The HAL consists of multiple library modules, each of which implements an interface for a specific type of hardware component, such as the camera or Bluetooth module. When a framework API makes a call to access device hardware, the Android system loads the library module for that hardware component.

Android Runtime

For devices running Android version 5.0 (API level 21) or higher, each app runs in its own process and with its own instance of the Android Runtime (ART). ART is written to run multiple virtual machines on low-memory devices by executing DEX files, a byte code format designed especially for Android that's optimized for minimal memory footprint. Build tool chains, such as Jack, compile Java sources into DEX byte code, which can run on the Android platform.

Some of the major features of ART include the following:

- Ahead-of-time (AOT) and just-in-time (JIT) compilation
- Optimized garbage collection (GC)

- On Android 9 (API level 28) and higher, conversion of an app package's Dalvik Executable format (DEX) files to more compact machine code.
- Better debugging support, including a dedicated sampling profiler, detailed diagnostic exceptions and crash reporting, and the ability to set watch points to monitor specific fields

Prior to Android version 5.0 (API level 21), Dalvik was the Android runtime. If your app runs well on ART, then it should work on Dalvik as well, but the reverse may not be true. Android also includes a set of core runtime libraries that provide most of the functionality of the Java programming language, including some Java 8 language features that the Java API framework uses.

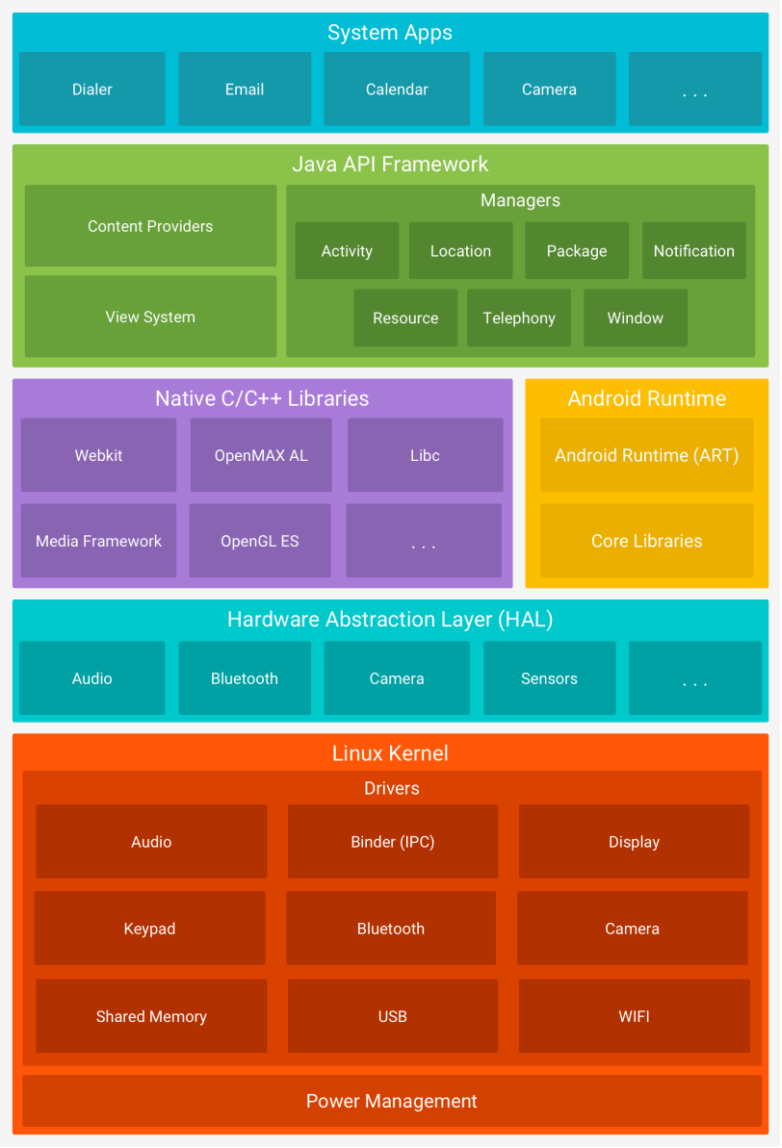

Figure-1: Android Software Stack

Native C/C++ Libraries

Many core Android system components and services, such as ART and HAL, are built from native code that requires native libraries written in C and C++. The Android platform provides Java framework APIs to expose the functionality of some of these native libraries to apps. For example, you can access OpenGL ES through the Android framework's Java OpenGL API to add support for drawing and manipulating 2D and 3D graphics in your app. If you are developing an app that requires C or C++ code, you can use the Android NDK to access some of these native platform libraries directly from your native code.

Java API Framework

The entire feature-set of the Android OS is available to you through APIs written in the Java language. These APIs form the building blocks you need to create Android apps by simplifying the reuse of core, modular system components and services, which include the following:

- A rich and extensible View System you can use to build an app's UI, including lists, grids, text boxes, buttons, and even an embeddable web browser
- A Resource Manager, providing access to non-code resources such as localized strings, graphics, and layout files
- A Notification Manager that enables all apps to display custom alerts in the status bar
- An Activity Manager that manages the lifecycle of apps and provides a common navigation back stack
- Content Providers that enable apps to access data from other apps, such as the Contacts app, or to share their own data

Developers have full access to the same framework APIs that Android system apps use.

System Apps

Android comes with a set of core apps for email, SMS messaging, calendars, internet browsing, contacts, and more. Apps included with the platform have no special status among the apps the user chooses to install. So a third-party app can

become the user's default web browser, SMS messenger, or even the default keyboard (some exceptions apply, such as the system's Settings app).

The system apps function both as apps for users and to provide key capabilities that developers can access from their own app. For example, if your app would like to deliver an SMS message, you don't need to build that functionality you—you can instead invoke whichever SMS app is already installed to deliver a message to the recipient you specify.

Factors that affect Mobile Application development

You should keep in mind the following factors while developing mobile application:

- Low processing power
- Limited RAM
- Limited permanent storage capacity
- Small screens with low resolution
- High costs associated with data transfer
- Slow data transfer rates with high latency
- Unreliable data connections
- Limited battery life

Following is some of the factors that affect app development time:

- User Interface & User Experience.
- Custom application
- Resource availability
- App security and publishing the app
- App designing
- Number of screens/devices/platforms/operating systems
- Third party integration
- Features
- Understanding the business logic
- Complexity of the App

Let us sum up

In this module we have discussed the history of mobile application development, you have got detailed understanding of android and its version, android SDK features and architecture, and we also elaborate factors that should be given consideration while developing mobile application

References

- Android™ Wireless Application Development, Second Edition, Shane Conder Lauren Darcey, Addison-Wesley, 2010
- Beginning Android™ Application Development, Wei-Meng Lee, Wiley Publishing Inc., 2011

Acknowledgement: "The content in this module is modifications based on work created and shared by the Android Open-Source Project and used according to terms described in the Creative Commons 2.5 Attribution License."