1.2 INFRASTRUCTURE AND MANAGING RESOURCES

ANDROID: AN OPEN PLATFORM FOR MOBILE DEVELOPMENT

More recently, Android has expanded beyond a pure mobile phone platform toprovide a development platform for an increasingly wide range of hardware, including tablets and televisions.

Put simply, Android is an ecosystem made up of a combination of three components:

a. **A free, open-source operating system for embedded devices**
b. **An open-source development platform for creating applications**
c. **Devices**

More specifically, Android is made up of several necessary and dependent parts,including the following:

- **A Compatibility Definition Document (CDD) and Compatibility Test Suite (CTS)**
  that describe the capabilitiesrequired for a device to support the software stack.

- A linux operating system kernel that provides a low-level interface with the hardware, memory management, and process control, all optimized for mobile and embedded devices.

- Open source libraries for application development, including SQLite, WebKit, OpenGL, and a media manager.

- A runtime used to execute and host Android applications, including **the Dalvik Virtual Machine (DVM)** and the core libraries that provide Android- specific functionality. The run time is designed to be small and efficient for use on mobile devices.

- A user interface framework used toho stand launch applications.
- A set of core installed applications.

- A SDK (software development kit) used to create applications including the relatedtools, plugins and documentation.

WHAT DOES ANDROID RUN ON?

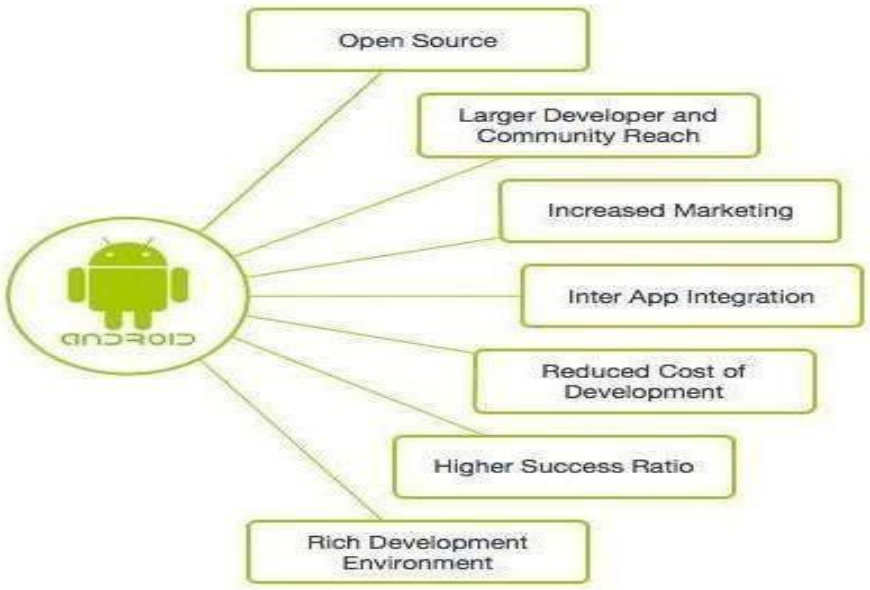
- **The first Android mobile handset, theT-MobileG1, was released in the United States in October 2008.**
- **Android is created to support variety of platform and applications.**

WHY DEVELOP FOR ANDROID?

Android represents a clean break, a mobile framework based on thereality of modern mobile devices designed by developers, for developers.

The barrier to entry for new Android developers is minimal:

- No certification is required to become an Android developer.
- Google Play provides free, up-front purchase, and in-app billing options for distribution and monetization of your applications.
- There is no approval process for application distribution.



- Developers have total control over their brand

WHAT ANDROID HAS THAT OTHER PLATFORMS DON'THAVE

- Google Maps applications
- Background services and applications
- Shared data and inter-process communication
- All application are created equal
- Wi-Fi Direct and Android Beam

INTRODUCING THE DEVELOPMENT FRAMEWORK

- Android applications normally are written using Java as the programming language but executed by means of a custom VM called **Dalvik, rather than a traditional JavaVM.**

- Each Android application runs in a separate process with in its own Dalvik instance, relinquishing all responsibility for memory and process management to the Android runtime, which stops and kills processes as necessary to manage resources.

WHAT COMES IN THE BOX

- **TheAndroidAPIs —** The core of the SDK is the Android API libraries that

provide developer access to the Android stack. These are the same libraries that Google uses to create native Android applications.

- **Development tools —** The SDK includes several development tools that let youcompile and debug your applications so that you can turn Android source code into executable applications.

- **The Android Virtual Device Manager and emulator—**The Android emulator is a fully interactive mobile device emulator featuring several alternative skins. The emulator runs within an Android Virtual Device (AVD)that simulates a device hardware configuration.
- **Fulldocumentation**
- **Samplecode**
- **Onlinesupport**.

UNDERSTANDING THE ANDROID SOFTWARE STACK

TheAndroid software stack is,put simply,a Linux kernel and a collectionof C/ C++libraries exposed through an application framework that provides service for, andmanagement of, the runtime and applications.

- **Linuxkernel**

  Coreservices(includinghardwaredrivers,processandmemorymanagement,security
  ,network,andpo wermanagement) are handled by a Linux2.6 kernel.
  The kernel also **provides an abstraction layer between the hardwareand the remainder of the stack.**

  Libraries—Running on top of the kernel, Android includes various C/C++ corelibraries such as libc and SSL.

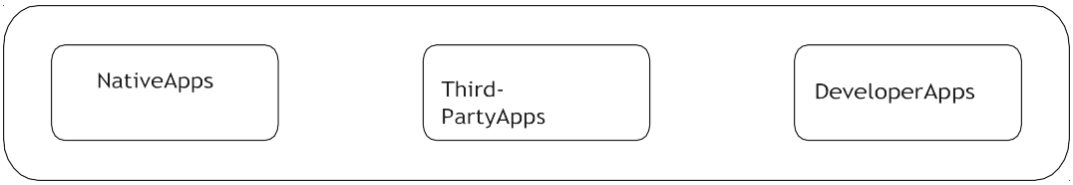
- **Androidruntime—**The runtime is what makes an Android phone an Android phone rather than a mobile Linux implementation. Including the core libraries and the DalvikVM, the **Android runtime is the engine that powers your**

  applications and, along with the libraries, forms the basis for the application framework.

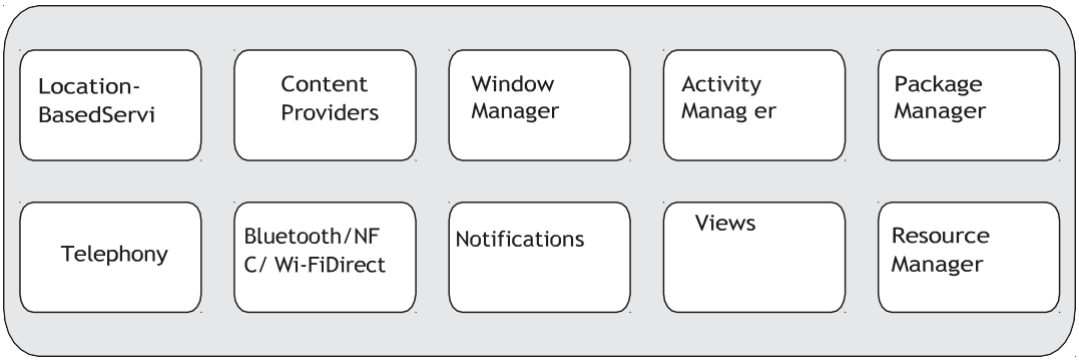
- **Corelibraries—**Although most Android application development is **written using the Java language**, Dalvik is not a JavaVM. The core Android libraries provide most of the functionality **available in the core Java libraries, as well as the Android- specific libraries**.

- **DalvikVM—**Dalvik is a register-based Virtual Machine that's been optimized to ensure that a **device can run multiple instances efficiently**. It relies onthe Linux kernel for threading and low-level memory management.

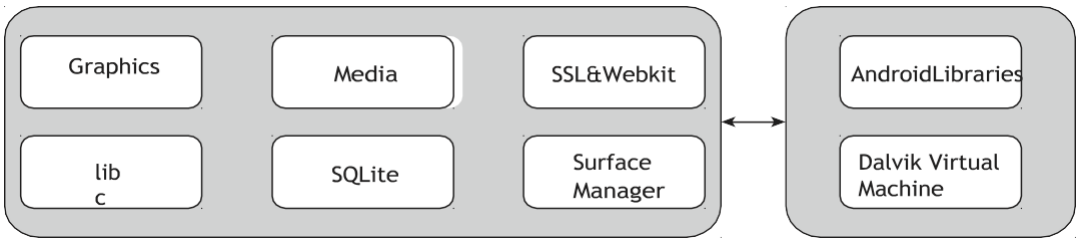
- **Application framework —** The application framework provides the classes used to create Android applications. It also provides a **generic abstraction for hardware access and manages the user interface and application resources.**
- **Applicationlayer—**All applications, both native and third-party, are built on the application layer by means of the same API libraries. The application layer runs with in the Android runtime, using the classes and services made available from the application framework.



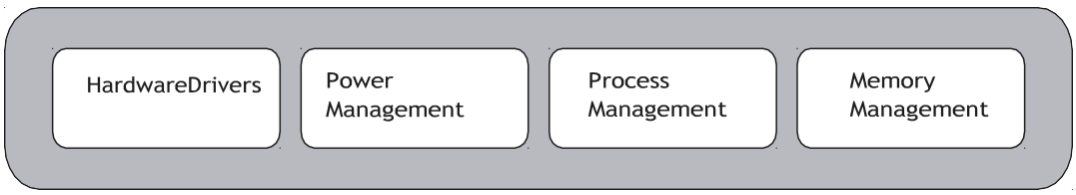
ApplicationLayer



ApplicationFramework



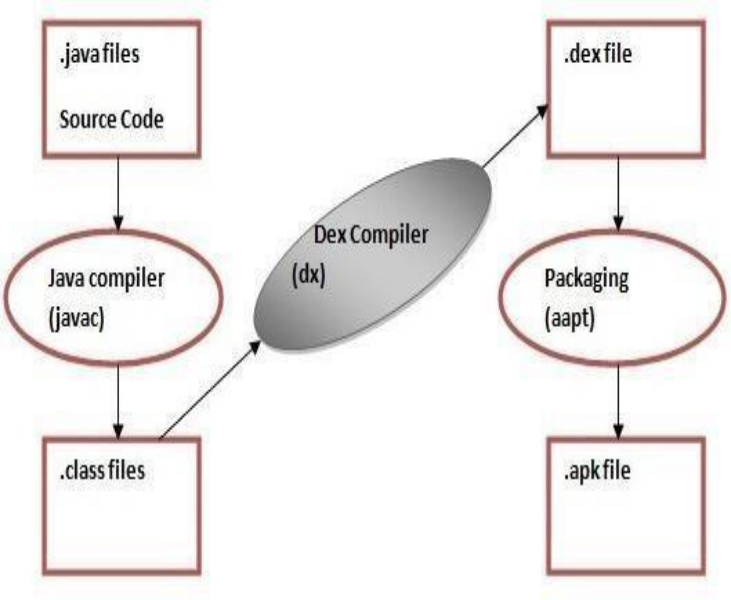
Libraries Android Run Time



Linux Kernel

The Dalvik Virtual Machine

- The **Dalvik Virtual Machine (DVM)** is an android virtual machine optimized for mobile devices. It **optimizes the virtual machine for *memory, battery life* and *performance*.**
- Dalvik is a name of a town in Iceland. The Dalvik VM was written by Dan Bornstein.
- The **Dex compiler converts the class files into the .dex file that run on the Dalvik VM**. Multiple class files are converted into one dex file.
- Let's see the compiling and packaging process from the source file:



Android Application Architecture

- Android's architecture encourages component reuse, enabling you to publish and share Activities, Services, and data with other applications, with access managed by the security restrictions.

Activity Manager and Fragment Manager

- **Views**—Used to construct the user interfaces for your Activities and Fragments.
- **Notification Manager**—provides a consistent and non intrusive mechanism for signaling your users.
- **Content Providers**—Lets your applications share data.
- **Resource Manager**—Enables non-code resources, such as strings and graphics, to be externalized.
- **Intents**—Provides a mechanism for transferring data between applications

and their components.

MANAGING RESOURCES

A mobile device is a specialized piece of hardware. It has several different types of resources that may require special monitoring and management activities that require scalable yet uniform designs. Android resources are basically files or external data supporting our app‘s operation. These files can be images, strings, colours, styles, etc.

KINDS OF RESOURCES FOR ANDROID APP DEVELOPMENT

1. Animation Resources

   - Their purpose is to **set default animations.**
   - They are stored in the res/drawable/ folder under the R.drawable identifier. They support bitmap files (.png, .jpg or .gif), PNG files in Nine-patch (.9.png) format and XML files with graphics descriptors.

2. Colour State List Resources

   - For determining a **component’s colour depending on its state.**

   - Stored in res/color/ and accessed through R.color.

3. Drawable Resources

   - For **defining bitmap or XML graphics**. They are stored in the res/drawable/ folder under the R.drawable identifier.

4. Layout Resources

   - **For defining an app’s interfaces. They contain XML files** and are stored in res/layout/ under ID R.layout.

5. Menu resources

   - For defining the contents of **an app’s menus**. They are stored in res/menu/ and accessed through R.menu.

6. String Resources

   - **They contain XML files with data embedded in strings or in string arrays**
     (including the string‘s format and style).

   - They can be sorted into strings.xml, colors.xml, dimens.xml, styles.xml or arrays.xml and are stored in res/values/ and include R.string, R.array, and R.plurals.

7. Style Resources

   - They define a **number of attributes that can be applied to a view or an activity**. Also used for **defining an app’s style and design.**

They are stored in res/values and accessed through R.style.

8. Other types:

- We‘ll be defining values as being Boolean, integer, dimensions, colours and other arrangements.

- They are stored in res/values/ but each of them is accessed through unique R subclasses (such as R.bool, R.integer, R.dimen, etc.).