Introduction to Android

Ref: Wei-Meng Lee, "BEGINNING ANDROID™ 4 APPLICATION DEVELOPMENT ", Ch1 , John Wiley & Sons , 2012

1. What is Android

Android is a mobile operating system that is based on a modified version of Linux. It was originally developed by a startup of the same name, Android, Inc. In 2005, as part of its strategy to enter the mobile space, Google purchased Android and took over its development work (as well as its development team).

Google wanted Android to be open and free; hence, most of the Android code was released under the open source Apache License, which means that anyone who wants to use Android can do so by downloading the full Android source code. Moreover, vendors (typically hardware manufacturers) can add their own proprietary extensions to Android and customize Android to differentiate their products from others.

The main advantage of adopting Android is that it offers a unified approach to application development. Developers need only develop for Android, and their applications should be able to run on numerous different devices, as long as the devices are powered using Android.

2. Android Versions

Android has gone through quite a number of updates since its first release. Table 1-1 shows the various versions of Android and their codenames.

TABLE 1-1: A Brief History of Android Versions

ANDROID VERSION	RELEASE DATE	CODENAME
1.1	9 February 2009	
1.5	30 April 2009	Cupcake
1.6	15 September 2009	Donut
2.0/2.1	26 October 2009	Eclair
2.2	20 May 2010	Froyo
2.3	6 December 2010	Gingerbread
3.0/3.1/3.2	22 February 2011	Honeycomb
4.0	19 October 2011	Ice Cream Sandwich
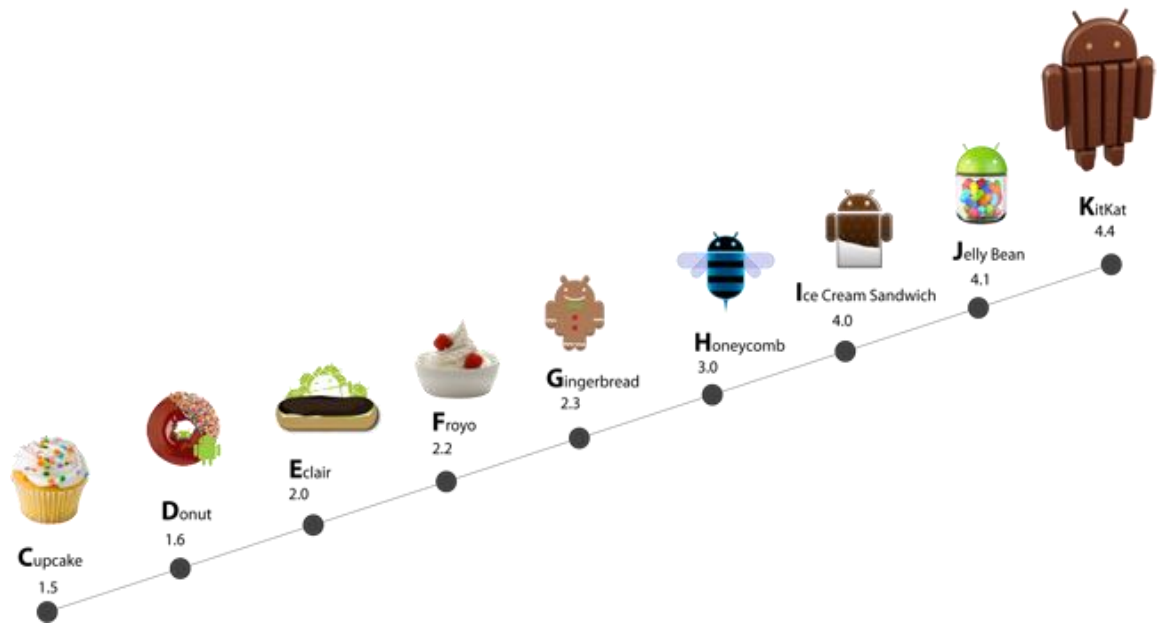
In February 2011, Google released Android 3.0, a tablet-only release supporting widescreen devices. The key changes in Android 3.0 are as follows.

1. New user interface optimized for tablets
2. 3D desktop with new widgets
3. Refi ned multi-tasking
4. New web browser features, such as tabbed browsing, form auto-fill, bookmark synchronization, and private browsing
5. Support for multi-core processors.

In October 2011, Google released Android 4.0, a version that brought all the features introduced in Android 3.0 to smartphones, along with some new features such as facial recognition unlock, data usage monitoring and control, Near Field Communication (NFC), and more.
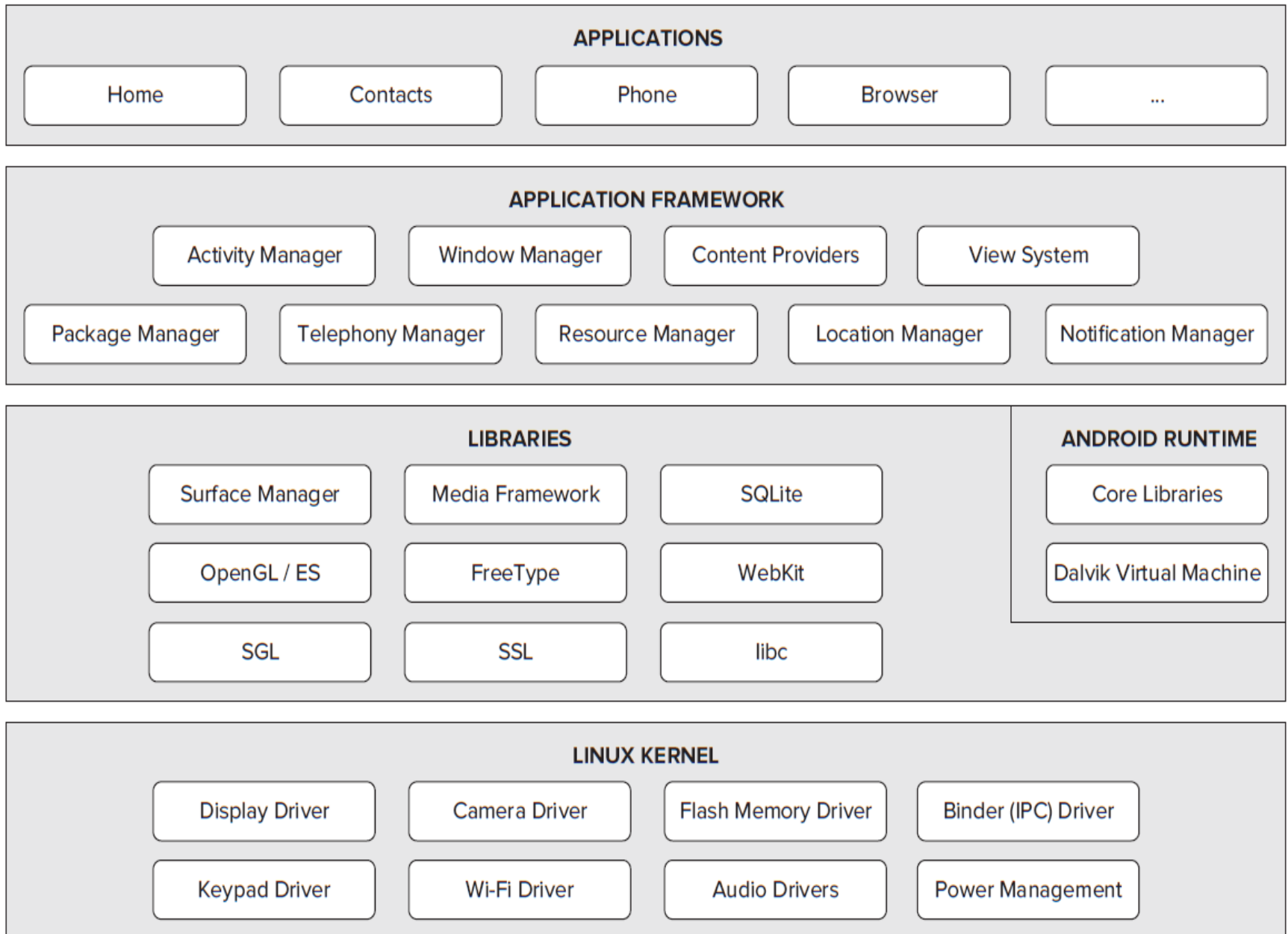
3. Features of Android

Because Android is open source and freely available to manufacturers for customization, there are no fixed hardware or software configurations. However, Android itself supports the following features:

1. **Storage** — Uses SQLite, a lightweight relational database, for data storage.
2. **Connectivity** — Supports GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth (includes A2DP and AVRCP), Wi-Fi, LTE, and WiMAX.
3. **Messaging** — Supports both SMS and MMS.
4. **Web brow**ser — Based on the open source WebKit, together with Chrome's V8 JavaScript engine
5. **Media support** — Includes support for the following media: H.263, H.264 (in 3GP or MP4 container), MPEG-4 SP, AMR, AMR-WB (in 3GP container), AAC, HE-AAC (in MP4 or 3GP container), MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF, and BMP.
6. **Hardware suppor**t — Accelerometer Sensor, Camera, Digital Compass, Proximity Sensor, and GPS
7. **Multi-touch** — Supports multi-touch screens
8. **Multi-tasking** — Supports multi-tasking applications
9. **Flash support** — Android 2.3 supports Flash 10.1.
10. **Tethering** — Supports sharing of Internet connections as a wired/wireless hotspot.

4. Architecture of Android

In order to understand how Android works, take a look at Figure 1-1, which shows the various layers that make up the Android operating system (OS). The Android OS is roughly divided into five sections in four main layers:

1. Linux kernel — This is the kernel on which Android is based. This layer contains all the low level device drivers for the various hardware components of an Android device.

2. Libraries — These contain all the code that provides the main features of an Android OS. For example, the SQLite library provides database support so that an application can use it for data storage. The WebKit library provides functionalities for web browsing.

3. Android runtime — At the same layer as the libraries, the Android runtime provides a set of core libraries that enable developers to write Android apps using the Java programming language. The Android runtime also includes the **Dalvik virtual machine**, which enables every Android application to run in its own process, with its own instance of the Dalvik virtual machine (Android applications are compiled into Dalvik executables).

4. Application framework — Exposes the various capabilities of the Android OS to application developers so that they can make use of them in their applications.

5. Applications — At this top layer, you will find applications that ship with the Android device (such as Phone, Contacts, Browser, etc.), as well as applications that you download and install from the Android Market. Any applications that you write are located at this layer.

APPLICATIONS

| Home | Contacts | Phone | Browser | ... |

APPLICATION FRAMEWORK

| Activity Manager | Window Manager | Content Providers | View System |

| Package Manager | Telephony Manager | Resource Manager | Location Manager | Notification Manager |

LIBRARIES

| Surface Manager | Media Framework | SQLite |

| OpenGL / ES | FreeType | WebKit |

| SGL | SSL | libc |

ANDROID RUNTIME

| Core Libraries |

| Dalvik Virtual Machine |

LINUX KERNEL

| Display Driver | Camera Driver | Flash Memory Driver | Binder (IPC) Driver |

| Keypad Driver | Wi-Fi Driver | Audio Drivers | Power Management |

5. **Android Devices in the Market**

Android devices come in all shapes and sizes. As of late November 2011, the Android OS powers the following types of devices:

- ➤ Smartphones

- ➤ Tablets

- ➤ E-reader devices

- ➤ Netbooks

- ➤ MP4 players

- ➤ Internet TVs

6. **The Android Market**

As mentioned earlier, one of the main factors determining the success of a smartphone platform is the applications that support it. Using the Market application that is preinstalled on their Android device, users can simply download third-party applications directly onto their devices. Both paid and free applications are supported on the Android Market, though paid applications are available only to users in certain countries due to legal issues. Similarly, in some countries, users can buy paid applications from the Android Market, but developers cannot sell in that country.

7. **The Android Developer Community**

With Android in its fourth version, there is a large developer community all over the world. It is now much easier to get solutions to problems, and find like-minded developers

to share app ideas and exchange experiences. Here are some developer communities/sites that you can turn to for help if you run into problems while working with Android:

- Stack Overflow (www.stackoverflow.com) — Stack Overflow is a collaboratively edited question and answer site for developers.

- Google Android Training (http://developer.android.com/training/index .html) — Google has launched the Android Training site that contains a number of useful classes grouped by topics.

- Android Discuss (http://groups.google.com/group/android-discuss) — Android Discuss is a discussion group hosted by Google using the Google Groups service.

8. OBTAINING THE REQUIRED TOOLS

The Android SDK makes use of the **Java SE Development Kit (JDK).** If your computer does not have the JDK installed, you should start by downloading it from www.oracle.com/technetwork/java/javase/downloads/index.html and installing it prior to moving to the next section.

Android SDK The first and most important piece of software you need to download is, of course, the Android SDK. The Android SDK contains a debugger, libraries, an emulator, documentation, sample code, and tutorials. You can download the Android SDK from http://developer.android.com/sdk/index.html.

The Android SDK Manager manages the various versions of the Android SDK currently installed on your computer. When it is launched, you will see a list of items and whether or not they are currently installed on your computer (see Figure 1-12). Check the relevant tools, documentation, and platforms you need for your project. Once you have selected the items you want, click the Install button to download them.

For Android development, you should download the **Eclipse IDE for Java EE Developers** (www.eclipse.org/downloads/). Six editions are available: Windows (32- and 64-bit), Mac OS X (Cocoa 32- and 64), and Linux (32- and 64-bit). Simply select the relevant one for your operating system. All the examples in this book were tested using the 32-bit version of Eclipse for Windows. For Android development, you should download the Eclipse IDE for Java EE Developers (www.eclipse.org/downloads/). Six editions are available: Windows (32- and 64-bit), Mac OS X (Cocoa 32- and 64), and Linux (32- and 64-bit). Simply select the relevant one for your operating system. All the examples in this book were tested using the 32-bit version of Eclipse for Windows.

When Eclipse is launched, select Help ⇨ Install New Software (see Figure 1-15) to install the **Android Development Tools (ADT)** plug-in for Eclipse. The ADT is an extension to the Eclipse IDE that supports the creation and debugging of Android applications. Using the ADT, you will be able to do the following in Eclipse:

➤ Create new Android application projects.

➤ Access the tools for accessing your Android emulators and devices.

➤ Compile and debug Android applications.

➤ Export Android applications into Android Packages (APKs).

➤ Create digital certificates for code-signing your APK.

The next step is to create an **Android Virtual Device (AVD)** to be used for testing your Android applications. An AVD is an emulator instance that enables you to model an actual device. Each AVD consists of a hardware profile; a mapping to a system image; as well as emulated storage, such as a secure digital (SD) card. You can create as many AVDs as you want in order to test your applications with several different configurations. This testing is important to confirm the behavior of your application when it is run on different devices with varying capabilities.