Open Handset Alliance

With its user-centric, democratic design philosophies, Google has led amovement to turn the existing closely guarded wireless market into one where phone users can move between carriers easily and have unfettered access to applications and services. With its vast resources, Google has taken a broad approach, examining the wireless infrastructure—from the FCC wireless spectrum policies to the handset manufacturers' requirements, application developer needs, and mobile operator desires.

Manufacturers: Designing Android Devices

More than half the members of the OHA are device manufacturers, such as Samsung, Motorola, Dell, Sony Ericsson, HTC, and LG, and semiconductor companies, such as Intel, Texas Instruments, ARM, NVIDIA, and Qualcomm. Manufacturers continue to create new generations of Android devices—from phones with HD displays, to watches for managing exercise programs, to dedicated e-book readers, to full-featured televisions, netbooks, and almost any. other "smart" device

Mobile Operators: Delivering the Android Experience

After you have the devices, you have to get them out to the users. Mobile operators from North, South, and Central America as well as Europe, Asia, India, Australia, Africa, and the Middle East have joined the OHA, ensuring a worldwide market for the Android movement. With almost half a billion subscribers alone, telephony giant China Mobile is a founding member of the alliance. Sprint launched the Evo 4G to much fanfare and record one-day sales (http://j.mp/uuWVIQ).

Apps Drive Device Sales: Developing Android Applications

Initially, Google led the pack in developing Android applications, many of which, such as the email client and web browser, are core features of the platform. They also developed the first successful distribution platform for third-party Android applications: The Android Market. The Android Market remains the primary method for which users download apps, but it is no longer the only distribution mechanism for Android apps.

Taking Advantage of All Android Has to Offer
Android's open platform has been embraced by much of the mobile development community—extending far beyond the members of the OHA.
The Android Marketplace: Where We're at Now

The Android marketplace continues to grow at an aggressive rate on all fronts (devices, developers, and users). Lately, the focus has been on several topics:
• **Competitive hardware and software feature upgrades:** The Android SDK developers have focused on providing APIs for features that are available on competing platforms, to bring them into parity. For example, recent releases of the Android SDK have featured significant improvements from an enterprise perspective.
• **Expansion beyond smartphones:** Android was almost exclusively a smartphone platform prior to Android 3.x (Honeycomb). Android 4.0 (Ice Cream Sandwich) merges (dare we say "sandwiches") the smartphone platform features and the tablet/other device features in one seamless SDK.
• **Free*:** Android applications are free to develop. There are no licensing or royalty fees to develop on the platform. No required membership fees. No required testing fees. No required signing or certification fees. Android applications can be distributed and commercialized in a variety of ways. (The term "Free*" implies there might actually be costs to development, but they are not mandated by the platform. Costs for designing, developing, testing, marketing, and maintaining are not included. If you provide all of these, you may not be laying out cash, but there is a cost associated with them.)

Improved user-facing features and marketing: The Android development team has shifted focus from feature implementation to providing user-facing usability upgrades and "chrome." Holographic user interfaces are the result. Google app developers have followed suit, updating the core apps to reflect better designs. The Android Market has received a number of sophisticated new features (including a web-based app store) and more interesting categories and organization.

Android Platform Differences

The Android platform itself is hailed as "the first complete, open, and free mobile platform":

• **Complete:** The designers took a comprehensive approach when they developed the Android platform. They began with a secure operating system and built a robust software framework on top that allows for rich application development opportunities.

• **Open:** The Android platform is provided through open-source licensing. Developers have unprecedented access to the device features when developing applications.

• **Free:** Android applications are free to develop. There are no licensing or royalty fees to develop on the platform. No required membership fees. No required testing fees. No required signing or certification fees. Android applications can be distributed and commercialized in a variety of ways.

Android: A Next-Generation Platform

Although Android has many innovative features not available in existing mobile platforms, its designers also leveraged many tried-and-true approaches proven to work in the wireless world. It's true that many of these features appear in existing proprietary platforms, but Android combines them in a free and open fashion while simultaneously addressing many of the flaws on these competing platforms.

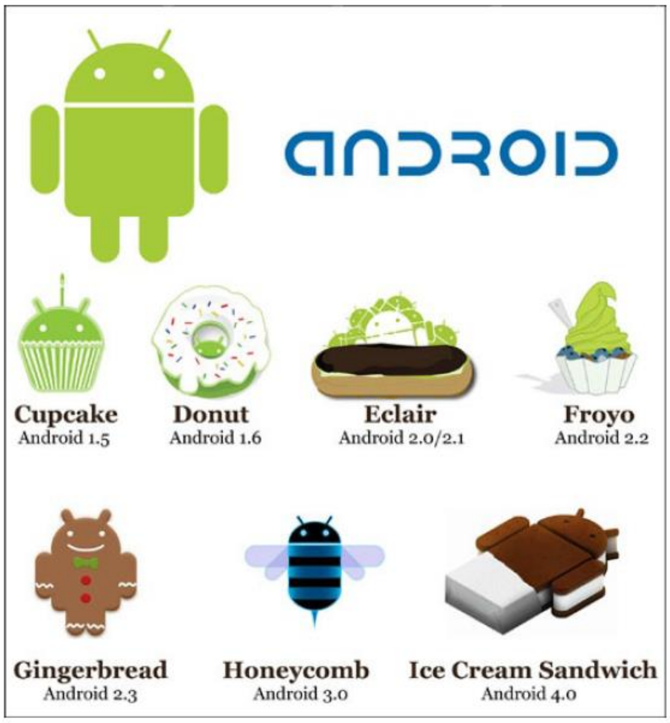

Figure 1.6 Some Android SDKs and their codenames.

Free and Open Source

Android application developers have the ability to distribute their applications under whatever licensing scheme they prefer. Developers can write open-source freeware or traditional licensed applications for profit and everything in between.

Familiar and Inexpensive Development Tools

Freely Available Software Development Kit

The Android SDK and tools are freely available. Developers can download the Android SDK from the Android website after agreeing to the terms of the Android Software Development Kit License Agreement.

Familiar Language, Familiar Development Environments

Developers have several choices when it comes to integrated development environments (IDEs). Many developers choose the popular and freely available Eclipse IDE to design and develop Android applications. Eclipse is the most popular IDE for Android development, and an Android plug-in is available for facilitating Android development.

Reasonable Learning Curve for Developers

Android applications are written in a well-respected programming language: Java.

Enabling Development of Powerful Applications

Rich, Secure Application Integration

Recall from the bat story I previously shared that I accessed a variety of phone applications in the course of a few moments: text messaging, phone dialer, camera, email, picture messaging, and the browser. Each was a separate application running on the phone—some built in and some purchased. Each had its own unique user interface. None were truly integrated.

No Costly Obstacles to Publication

Android applications have none of the costly and time-intensive testing and certification programs required by other platforms such as BREW and Symbian.

A "Free Market" for Applications

Android developers are free to choose any kind of revenue model they want. They can develop freeware, shareware, trial-ware or ad-driven applications, and paid applications. Android was designed to fundamentally change the rules about what kind of wireless applications could be developed Because developers have a variety of application distribution mechanisms to choose from, they can pick the methods that work for them instead of being forced to play by others' rules.

Android developers can distribute their applications to users in a variety of ways:

1. Google developed the Android Market, a generic Android application store with a revenue-sharing model. Android Market now has a web store for browsing and buying apps online (see Figure 1.7). Android Market also sells movies, music, and books, so your application will be in good company.

2. Amazon Appstore for Android launched in 2011 with a lineup of exciting Android applications using its own billing and revenue-sharing models. A unique feature of Amazon Appstore for Android is that it allows users to demo certain applications in the web browser before purchasing them. Amazon uses an environment similar to the emulator that runs right in the browser.

3. Numerous other third-party application stores are available. Some are for niche markets; others cater to many different mobile platforms.

4. Developers can come up with their own delivery and payment mechanisms

Growing Platform

Early Android developers have had to deal with the typical roadblocks associated with a new platform: frequently revised SDKs, lack of good documentation, and market uncertainties.

The Android Platform

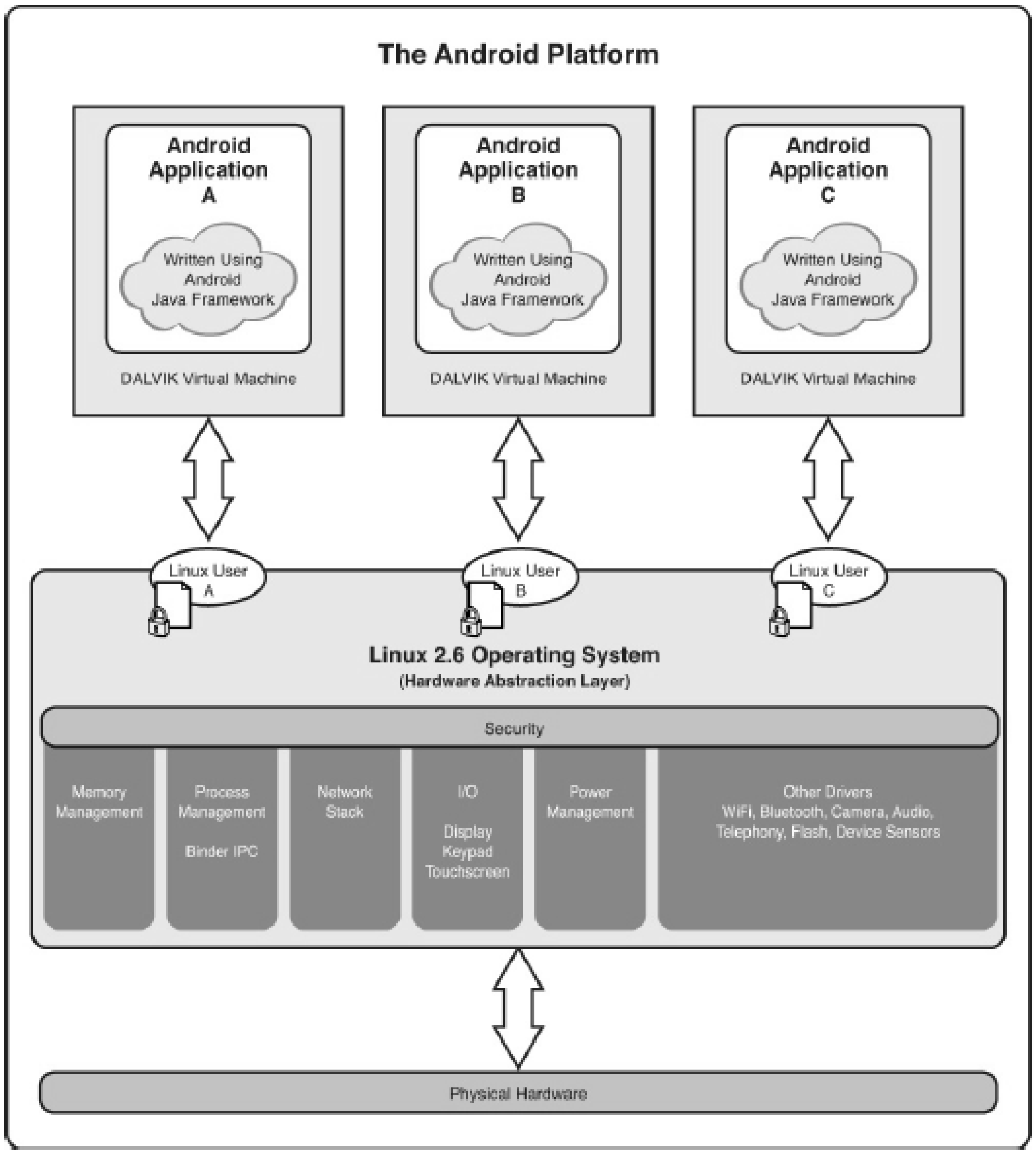

Figure 1.8 Diagram of the Android platform architecture.

The Linux Operating System

The Linux 2.6 kernel handles core system services and acts as a hardware abstraction layer (HAL) between the physical hardware of the device and the Android software stack.

Some of the core functions the kernel handles include

- Enforcement of application permissions and security
- Low-level memory management
- Process management and threading
- The network stack
- Display, keypad input, camera, Wi-Fi, Flash memory, audio, and binder (IPC) driver access

Android Application Runtime Environment

Each Android application runs in a separate process, with its own instance of the Dalvik virtual machine (VM). Based on the Java VM, the Dalvik design has been optimized for mobile devices. The Dalvik VM has a small memory footprint and optimized application loading, and multiple instances of the Dalvik VM can run concurrently on the device.

Security and Permissions

The integrity of the Android platform is maintained through a variety of security measures. These measures help ensure that the user's data is secure and that the device is not subjected to malware.

Applications as Operating System Users

When an application is installed, the operating system creates a new user profile associated with the application. Each application runs as a different user, with its own private files on the file system, a user ID, and a secure operating environment. The application executes in its own process with its own instance of the Dalvik VM and under its own user ID on the operating system.

Explicitly Defined Application Permissions

To access shared resources on the system, Android applications register for the specific privileges they require. Some of these privileges enable the application to use device functionality to make calls, access the network, and control the camera and other hardware sensors. Applications also require permission to access shared data containing private and personal information, such as user preferences, user's location, and contact information.

Limited Ad-Hoc Permissions

Applications that act as content providers might want to provide some on-the-fly permissions to other applications for specific information they want to share openly. This is done using ad-hoc granting and revoking of access to specific resources using Uniform Resource Identifiers (URIs).

Application Signing for Trust Relationships

All Android applications packages are signed with a certificate, so users know that the application is authentic. The private key for the certificate is held by the developer. This helps establish a trust relationship between the developer and the user. It also enables the developer to control which applications can grant access to one another on the system. No certificate authority is necessary; self-signed certificates are acceptable.

Marketplace Developer Registration

To publish applications on the popular Android Market, developers must create a developer account. The Android Market is managed closely and no malware is tolerated.

Developing Android Applications

The Android SDK provides an extensive set of application programming interfaces (APIs) that is both modern and robust. Android device core system services are exposed and accessible to all applications. When granted the appropriate permissions, Android applications can share data among one another and access shared resources on the system securely.

Android Programming Language Choices

Commonly Used Packages

With Android, mobile developers no longer have to reinvent the wheel. Instead, developers use familiar class libraries exposed through Android's Java packages to perform common tasks involving graphics, database access, network access, secure communications, and utilities. The Android packages include support for the following:

• A wide variety of user interface controls (Buttons, Spinners, Text input)
• A wide variety of user interface layouts (Tables, Tabs, Lists, Galleries)
• Secure networking and web browsing features (SSL, WebKit)
• XML support (DOM, SAX, XMLPullParser)
• Structured storage and relational databases (App Preferences, SQLite)
• Powerful 2D and 3D graphics (including SGL, OpenGL ES, and RenderScript)
• Multimedia frameworks for playing and recording standalone or network streaming (MediaPlayer, JetPlayer, SoundPool)
• Extensive support for many audio and visual media formats (MPEG4, MP3, Still Images)
• Access to optional hardware such as location-based services (LBS), USB, Wi-Fi, Bluetooth, and hardware sensors

Android Application Framework

The Android application framework provides everything necessary to implement your average application. The Android application lifecycle involves the following key components:

• Activities are functions the application performs.
• Groups of views define the application’s layout.
• Intents inform the system about an application’s plans.
• Services allow for background processing without user interaction.
• Notifications alert the user when something interesting happens.
• Content providers facilitate data transmission among different applications.

Android Platform Services

Android applications can interact with the operating system and underlying hardware using a collection of managers. Each manager is responsible for keeping the state of some underlying system service. For example:

• The LocationManager facilitates interaction with the location-based services available on the device.
• The ViewManager and WindowManager manage display and user interface fundamentals related to the device.
• The AccessibilityManager manages accessibility events, facilitating device support for users with physical impairments.
• The ClipboardManager provides access to the global clipboard for the device, for cutting and pasting content.
• The AudioManager provides access to audio and ringer controls.

References and More Information

Android Development:
http://developer.android.com
Open Handset Alliance:
http://www.openhandsetalliance.com
Official Android Developers Blog:
http://android-developers.blogspot.com
This Book’s blog:
http://androidbook.blogspot.com