Smartphone Applications (ITSMAP-01)

Lecture 1

Android Platform

Jesper Rosholm Tørresø

- Understanding The Android Platform

Introduction to Android

- Android is an open-source software stack that includes:
  - Operating system,
  - Middleware,
  - Mobile applications,
  - API libraries that can shape the look-n-feel and write business logic of mobile applications.

- Android represents an exciting new opportunity to write innovative applications for an increasingly wide range of devices

- Small, stylish, and versatile, modern mobile devices have become powerful tools that incorporate touchscreen, camera, media player, GPS receiver, and NFC hardware

Introduction to Android

- In Android, native and third-party applications are written with the same APIs and executed on the same run time.

- The APIs feature:
  - Hardware access,
  - video recording,
  - location-based services, map-based activities,
  - support for background services, relational databases, and inter-application communication (i.e., AIDL for IPC),
  - Bluetooth, NFC,
  - 2D and 3D graphics

Brief History

- Smartphone application developers (coding in low-level C or C++) have needed to understand the specific hardware for a range of devices from different manufacturers.
- Later on, Symbian provided to developers to develop rich applications that better leveraged the hardware available.
- The biggest advances in mobile phone development was the introduction of Java- hosted MIDlets (mobile information device profile).
- MIDlets are executed on a Java virtual machine (JVM), a process that abstracts the underlying hardware and lets developers create applications that run on the wide variety of devices that support the Java run time. Unfortunately, this convenience comes at the price of restricted access to the device hardware.

Current Status

- Platforms like Microsoft's Windows Phone and the Apple iPhone also provide a richer, simplified development environment for mobile applications; however, unlike Android, they're built on **proprietary** operating systems.
- In some cases they **prioritize** native applications over those created by third parties, restrict communication among applications and native phone data, and restrict or control the distribution of third-party applications to their platforms.
- Android offers an open development environment built on an **open-source Linux kernel**.
- Hardware access is available to all applications through a series of API libraries, and application interaction, while **carefully controlled**.

Current Status

Google's Andy Rubin describes Android as follows:

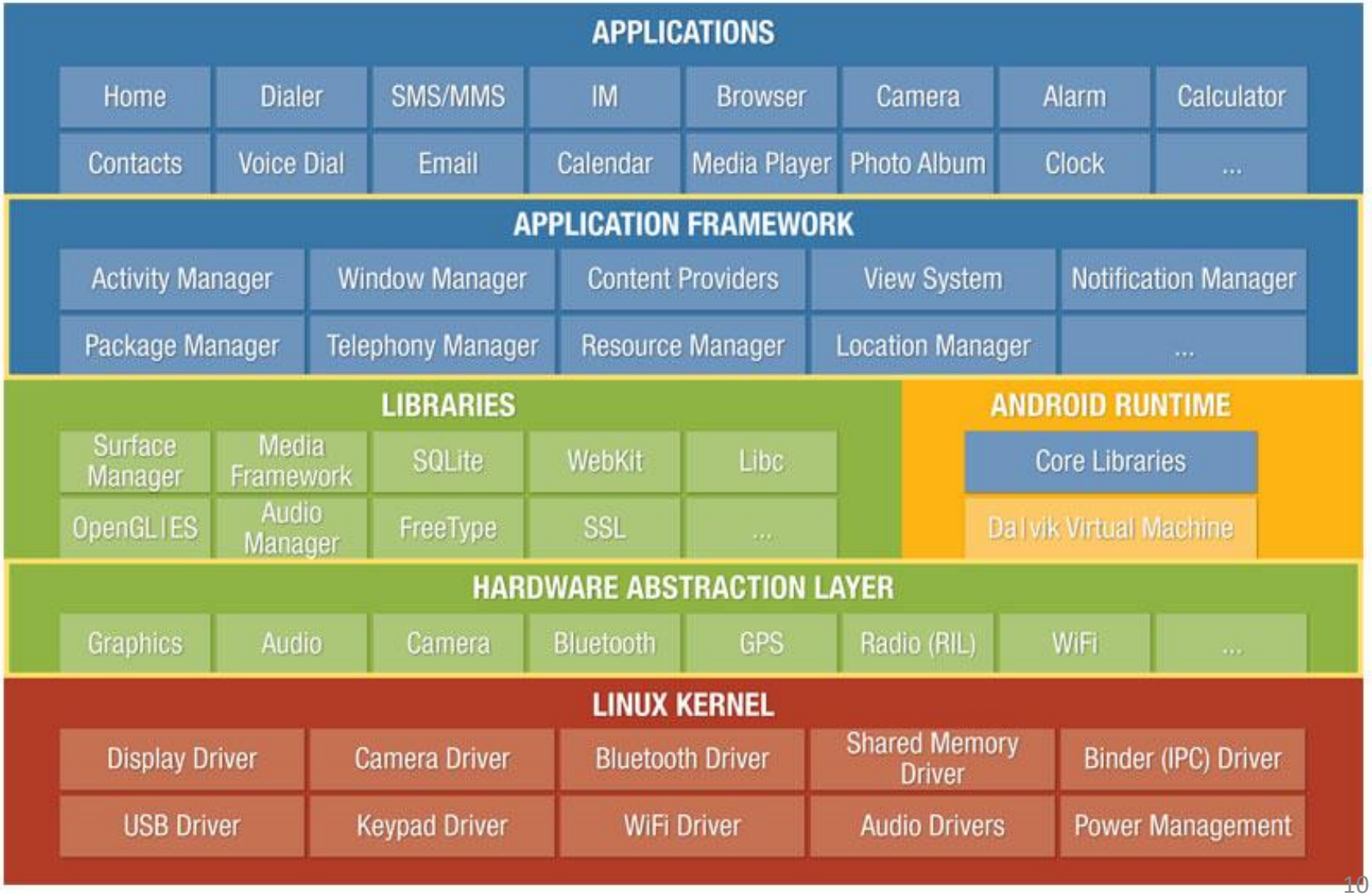
The first truly open and comprehensive platform for mobile devices. It includes an operating system, user-interface and applications — all of the software to run a mobile phone but without the proprietary obstacles that have hindered mobile innovation.

Where's My Gphone? (http://googleblog.blogspot.com/2007/11/wheres-my-gphone.html)

Components of Android Platform

- A **Compatibility Definition Document (CDD)** and **Compatibility Test Suite (CTS)** that describe the capabilities required 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 runtime is designed to be small and efficient for use on mobile devices.
- **An application framework** that agnostically exposes system services to the application layer, including the window manager and location manager, databases, telephony, and sensors.
- A **user interface framework** used to host and launch applications.
- A set of core **pre-installed applications**.
- A **software development kit (SDK)** used to create applications, including the related tools, plug-ins, and documentation.

Android Software Stack



Android Software Stack

Linux kernel & Hardwar Abstraction Layer

- Core services including
  - Hardware drivers,
  - Process and memory management,
  - Security,
  - Network, and
  - Power management
  - An abstraction layer between the hardware and the remainder of the stack

Android Software Stack

Libraries:

- Running on top of the kernel,
- Android includes various C/C++ core libraries such as libc and SSL, as well as the following:
  - A media library for playback of audio and video media
  - A surface manager to provide display management
  - Graphics libraries that include SGL and OpenGL for 2D and 3D graphics
  - SQLite for native database support
  - SSL and WebKit for integrated web browser and Internet security

Android Software Stack

Android Runtime:

- Application Runtime Environment
- Includes the **Core Libraries** and the **Dalvik Virtual Machine (DVM**)
- The **core Android libraries** provide most of the functionality available in the core Java libraries, as well as the Android-specific libraries.
- Although most Android application development is written using the Java language, **Dalvik is not a Java VM**.
- DVM is optimized to ensure that a **device can run multiple instances** efficiently. It relies on the Linux kernel for threading and low-level memory management.

Android Software Stack

Dalvik VM :

- It's also possible to write C/C++ applications that run closer to the underlying Linux OS. Although you can do this, in most cases there's no reason you should need to.
- If the speed and efficiency of C/C++ is required for your application, Android provides a **native development kit (NDK).** The NDK is designed to enable you to create C++ libraries using the libc and libm libraries, along with native access to OpenGL.
- The Dalvik VM executes Dalvik executable files, a format optimized to ensure minimal memory footprint. You create .dex executables by transforming Java language compiled classes using the tools supplied within the SDK.

Android Software Stack

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.

Application layer:

- All applications, both native and third-party, are built on the application layer by means of the same API libraries.
- The applications in the this layer run within the Android runtime, using the classes and services made available from the application framework

Android Application Architecture

- Encourages component reuse, enabling you to publish and share Activities, Services, and data with other applications, with access managed by the security restrictions you define.

- Enables to produce a replacement of native contact manager or phone dialer

- lets you expose your application's components in order to let other developers build on them by creating new UI front ends or functionality extensions.

Android Application Architecture/ Application Framework

Architectural cornerstones of all Android applications framework:

- **Activity Manager** and **Fragment Manager:** Control the lifecycle of Activities and Fragments
- **Views:** Construct user interfaces for Activities and Fragments
- **Notification Manager:** Provides a consistent and nonintrusive mechanism to notify events to users
- **Content Providers:** Lets applications share data
- **Resource Manager:** Enables non-code resources, such as strings, styles, colors, dimension and graphics to be **externalized**
- **Intents:** Provides a mechanism for transferring data between applications and their components
- **Android Libraries:** APIs for developing android applications

Native Android Applications

Native Applications:

- e-mail client
- SMS management application
- PIM (personal information management) suite, including a calendar and contacts list
- WebKit-based web browser
- Music player and picture gallery
- Camera and video recording application
- Calculator
- Home screen
- Alarm clock

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

- The Google Play Store for downloading third-party Android applications
- A fully featured mobile Google Maps application, including StreetView, driving directions and turn-by-turn navigation, satellite views, and traffic conditions
- The Gmail email client
- The Google Talk instant-messaging client
- The YouTube video player

Native Android Applications

- The exact makeup of the applications available on new Android phones is likely to vary based on the hardware manufacturer and/or the phone carrier or distributor.
- The open-source nature of Android means that carriers and OEMs can customize the user interface and the applications bundled with each Android device.
- Several OEMs have done this, including HTC with Sense, Motorola with MotoBlur, and Samsung with TouchWiz.
- It’s important to note that for compatible devices, the underlying platform and SDK remain consistent across OEM and carrier variations. The look and feel of the user interface may vary, but the applications will function in the same way across all compatible Android devices.