Introduction to Android

Dilman A. Salih

University of Halabja

Slides: 3,4,15,16,17,18, … to the end.

Outline

- ✓ What is Android?

- ✓ History of Android

- ✓ iPhone vs. Android

- ✓ Developer Revenues

- ✓ Android Architecture

- ✓ Android Features

What is Android?

❑ Android is a ***software stack*** for mobile devices that includes an *operating system*, *middleware* and *key applications*.

❑ Android is an operating system based on the **Linux Operating System**.

❑ The project responsible for developing the Android system is called the Android Open Source Project (**AOSP**) and is primarily lead by Google.

What is Android? (2)

❑ Uses Linux to provide core system services:

    ❑ Security

    ❑ Memory management

    ❑ Process management

    ❑ Power management

    ❑ Hardware drivers

❑ An Open Handset Alliance (**OHA**) project

    OHA: The Open Handset Alliance is a consortium of 84 firms

    to develop open standards for mobile devices

A Short History of Android

- **2001** Palm Kyocera 6035, combing PDA and phone
- **2003** - Blackberry Smartphone released
- **2005**
    - Google acquires startup Android Inc. to start Android platform
    - Work on Dalvik VM begins
- **2007**
    - Open Handset Alliance announced
    - Early look at SDK
    - June, iPhone released
- **2008**
    - Google sponsors 1st Android Developer Challenge
    - T-Mobile G1 announced, released fall
    - SDK 1.0 released
    - Android released open source (Apache License)
    - Android Dev Phone 1 released

A Short History of Android (2)

❑ **2009**

    ❑ SDK 1.5 (*Cupcake*)

        ❑ new soft keyboard with "autocomplete" feature

    ❑ SDK 1.6 (*Donut*)

        ❑ support Wide VGA

    ❑ SDK 2.0/2.0.1/2.1 (*Éclair*)

        ❑ revamped UI, browser

❑ **2010**

    ❑ Nexus One released to the public

    ❑ SDK 2.2 (*Froyo*)

        ❑ flash support, tethering

    ❑ SDK 2.3 (*Gingerbread*)

        ❑ UI update, system-wide copy-paste

A Short History of Android (3)

- **2011**
  - SDK 3.0 (***Honeycomb***) for tablets only
    - new UI for tablets, support multi-core processors, fragments
  - SDK 3.1 and 3.2
    - hardware support and UI improvements
  - SDK 4.0 (***Ice Cream Sandwich***)
    - for Q4, combination of Gingerbread and Honeycomb

Android powers hundreds of millions of mobile devices in more than ***190*** countries around the world

A Short History of Android (4)

❑ **2012**
  - ❑ Android 4.1, "*Jelly Bean*" announced late June 2012

❑ **2013**
  - ❑ Google launched Android 4.4 "*KitKat*" last September 2013

❑ **2014**
  - ❑ On November 2014, Android 4.5 or 5.0 "*Lollipop*" was released

❑ **2015**
  - ❑ On October 2015 a new version of Android 6.0 and 6.0.1 was released which is called "*Marshmallow*"

Android 4.1 (Jelly Bean)

A Short History of Android (5)
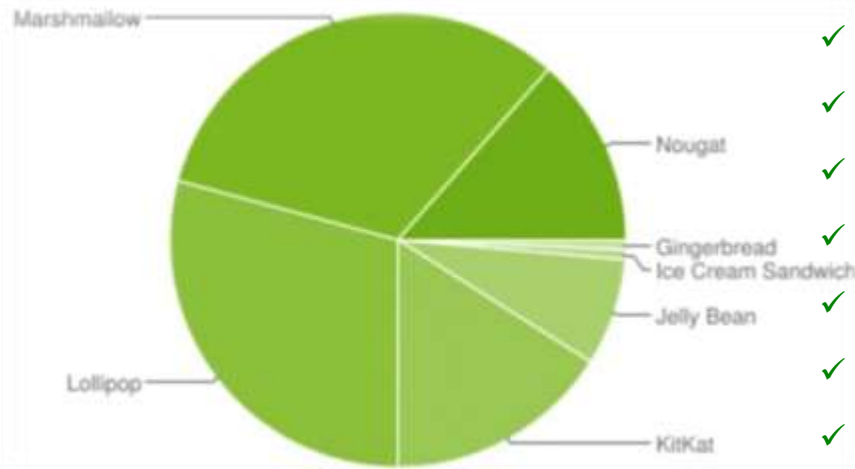
❑ **2016**

  ❑ Version 7.0 and 7.1.2 "***Nougat***" was released on 22
  August 2016

❑ **2017**

  ❑ The latest Android SDK version is 8.0 "***Oreo***" was
  announced on 21 August 2017

Device Distribution August 2016

- ✓ 2.3 "API 10" Gingerbread 0.7%
- ✓ 4.x "API 15" Ice Cream Sand. 0.7%
- ✓ 4.1.X "API 16" Jelly Bean 2.7%
- ✓ 4.2.X "API 17" Jelly Bean 3.8%
- ✓ 4.3.X "API 18" Jelly Bean 1.1%
- ✓ 4.4 "API 19" KitKat 16.0%
- ✓ 5.0 "API 21" Lollipop 7.4%
- ✓ 5.1 "API 22" Lollipop 21.8%
- ✓ 6.0 "API 23" Marshmallow 32.3%
- ✓ 7.0 "API 24" Nougat 12.3%
- ✓ 7.1 "API 25" Nougat 1.2%

❑ Based on *active devices*

❑ Forward compatible

Devices and Apps
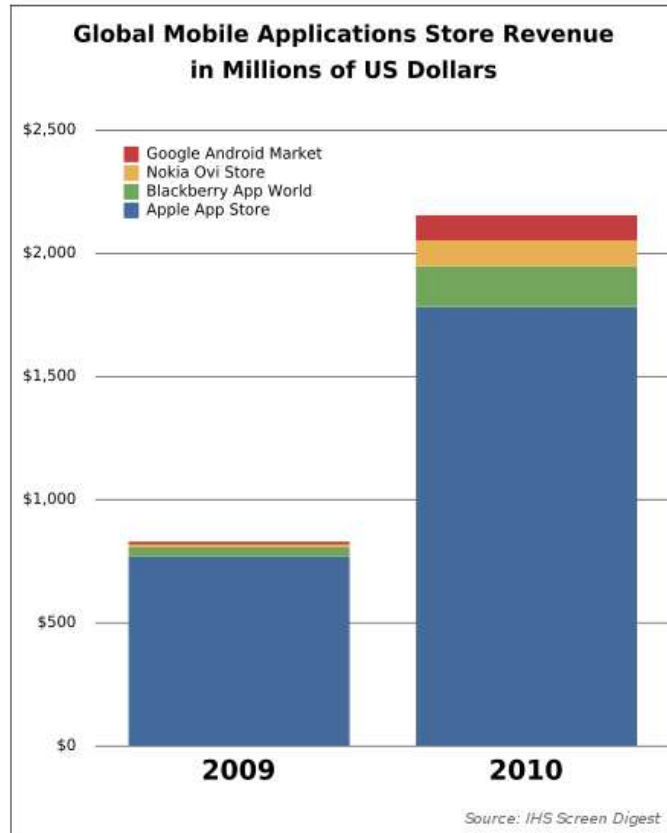
❑ Estimated 400M activated devices

❑ 1M new activations per day

❑ <u>Google Play</u> (formerly **Android Market**)

    ❑ 3 million **apps** in March 2017, after surpassing 1 million **apps** in July 2013

    ❑ 2/3 free, 1/3 paid

❑ <u>App Store</u> (formerly **Apple Market**)

    ❑ 1 million **app** mark at the end of 2013, and Apple currently has around 1.4 million iOS **apps** available for download

❑ What's old is new - Mac vs. PC

    iPhone vs. Android???

iPhone vs. Android

Developer Revenues

Global Mobile Applications Store Revenue in Millions of US Dollars

- Google Android Market
- Nokia Ovi Store
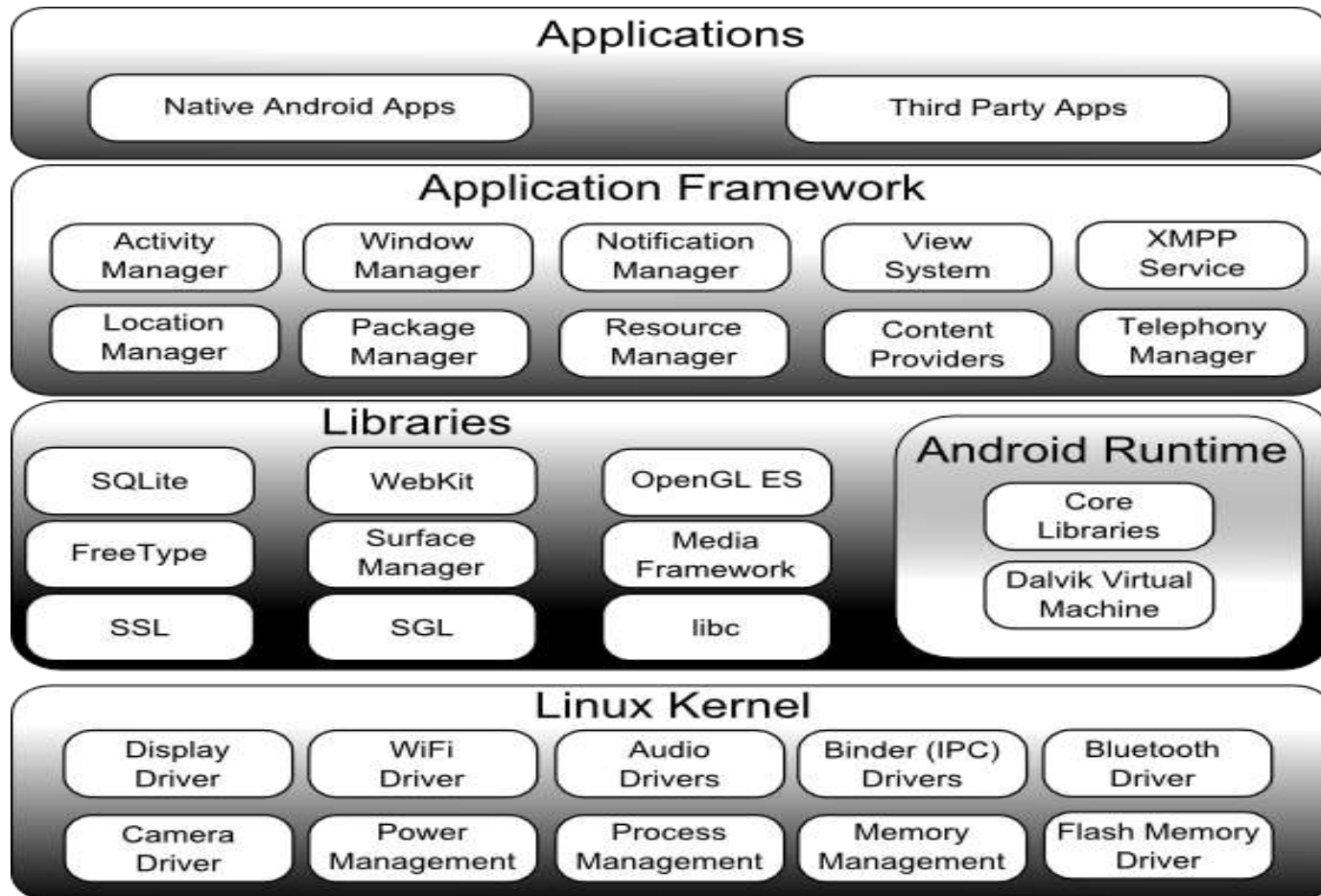- Blackberry App World
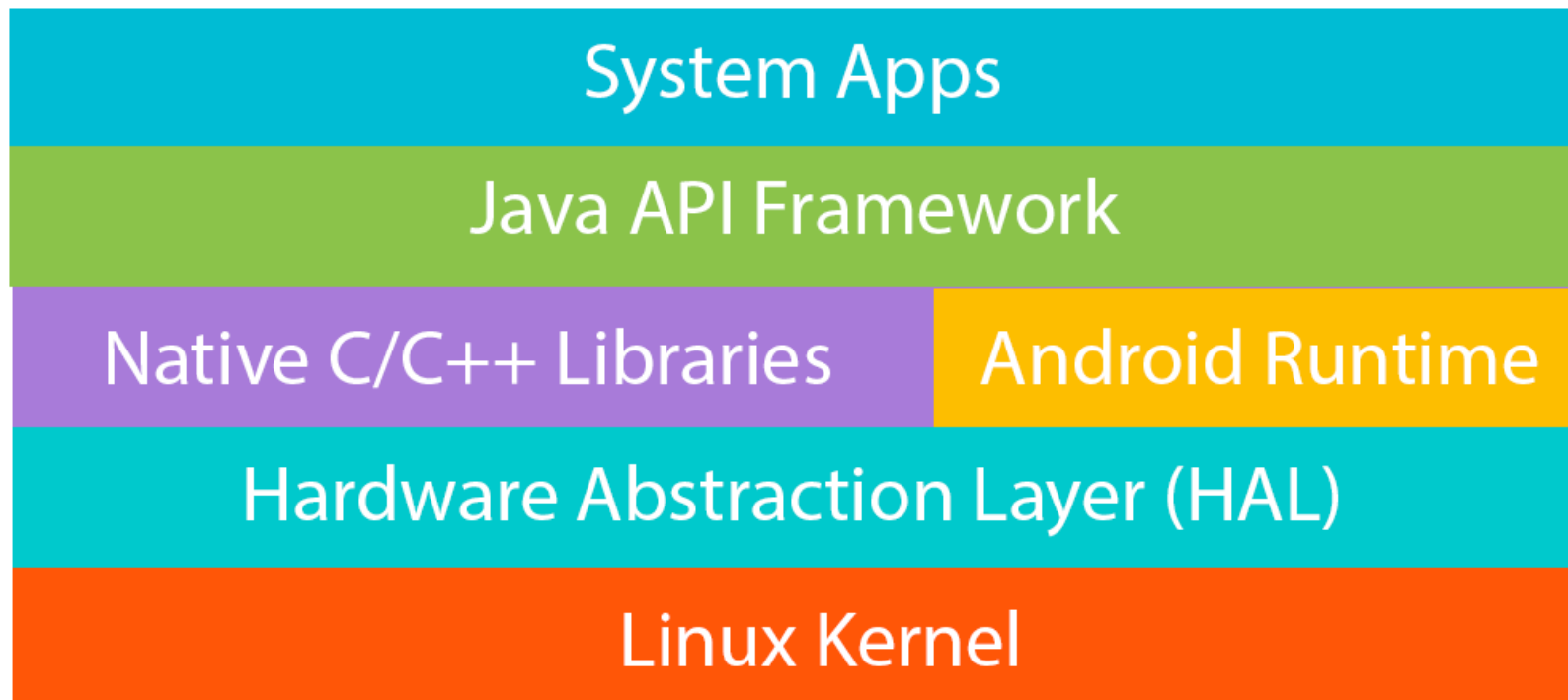- Apple App Store

Source: IHS Screen Digest

- ❑ Business Strategy:
    - ❑ Attract developers with comparison of revenue generated by applications, average revenue per user, ..etc
- ❑ Apple has the most high revenue in 2010
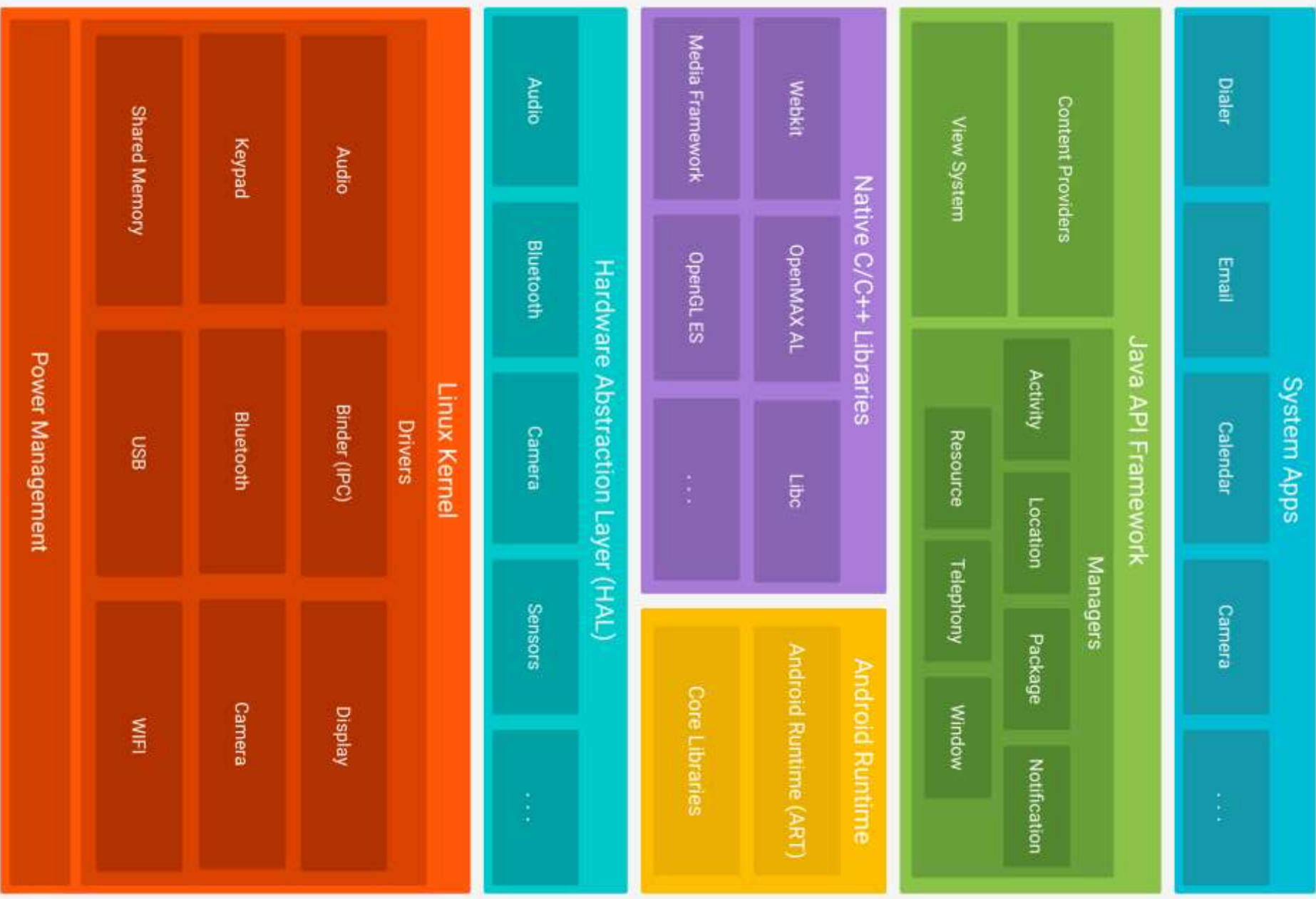
Android Architecture

Android Platform 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.**

System Apps
Java API Framework
Native C/C++ Libraries · Android Runtime
Hardware Abstraction Layer (HAL)
Linux Kernel

The Android software stack architecture diagram showing the following layers and components:

Linux Kernel
- Power Management
- Drivers:
  - Audio, Keypad, Shared Memory
  - Binder (IPC), Bluetooth, USB
  - Display, Camera, WIFI

Hardware Abstraction Layer (HAL)
- Audio
- Bluetooth
- Camera
- Sensors
- ...

Native C/C++ Libraries
- Media Framework
- Webkit
- OpenGL ES
- OpenMAX AL
- Libc
- ...

Android Runtime
- Android Runtime (ART)
- Core Libraries

Java API Framework
- View System
- Content Providers
- Managers:
  - Activity, Location, Package, Notification
  - Resource, Telephony, Window

System Apps
- Dialer
- Email
- Calendar
- Camera
- ...

16

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. Therefor, 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).

System Apps

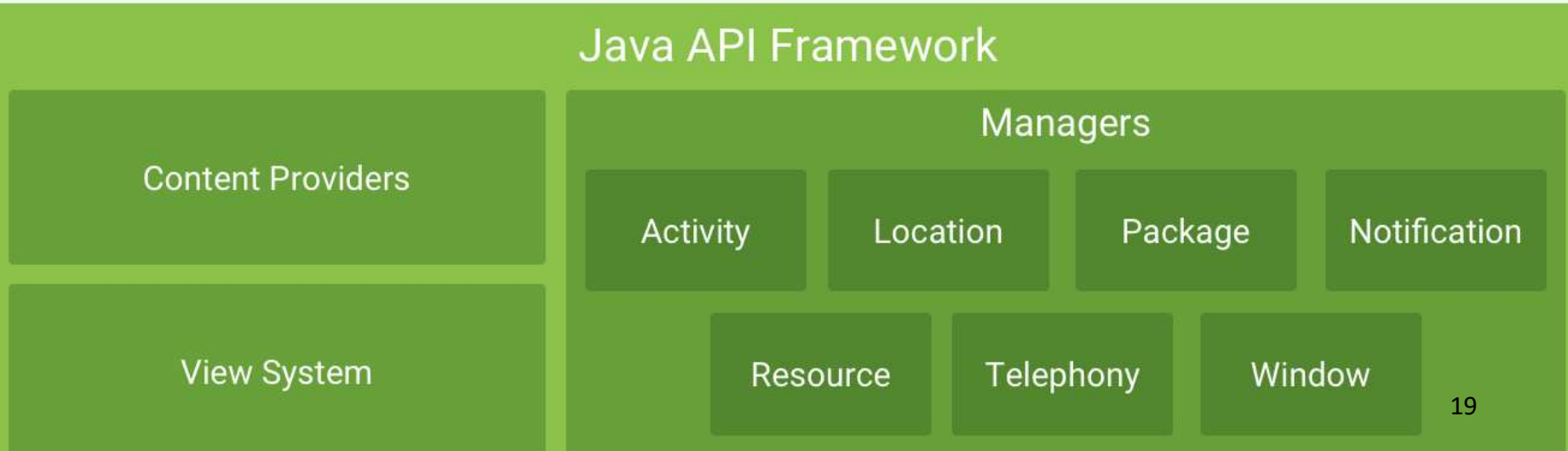| Dialer	Email	Calendar	Camera	. . .

System Apps

❑ 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 yourself—you can instead invoke whichever SMS app is already installed to deliver a message to the recipient you specify.

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**.

Java API Framework

Content Providers	Managers			
	Activity	Location	Package	Notification
View System	Resource	Telephony	Window	

Java API Framework: System components and services

❑ **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**

Native C/C++ Libraries

Many core Android system components and services, such as ART and HAL, are built from native code that require 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.

Native C/C++ Libraries			Android Runtime	
Webkit	OpenMAX AL	Libc	Android Runtime (ART)	
Media Framework	OpenGL ES	. . .	Core Libraries	

Android Runtime (ART)

❑ Homework!

Submit your answer to the Google Classroom assignment.

Native C/C++ Libraries			Android Runtime
Webkit	OpenMAX AL	Libc	Android Runtime (ART)
Media Framework	OpenGL ES	. . .	Core Libraries

Android Runtime (ART)

ART:
Android runtime

Runs Android framework and applications
Interpreter, Compiler

Manages application memory
Memory allocator, Garbage Collector

Taken From: The Evolution of ART - Google I/O 2016

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.
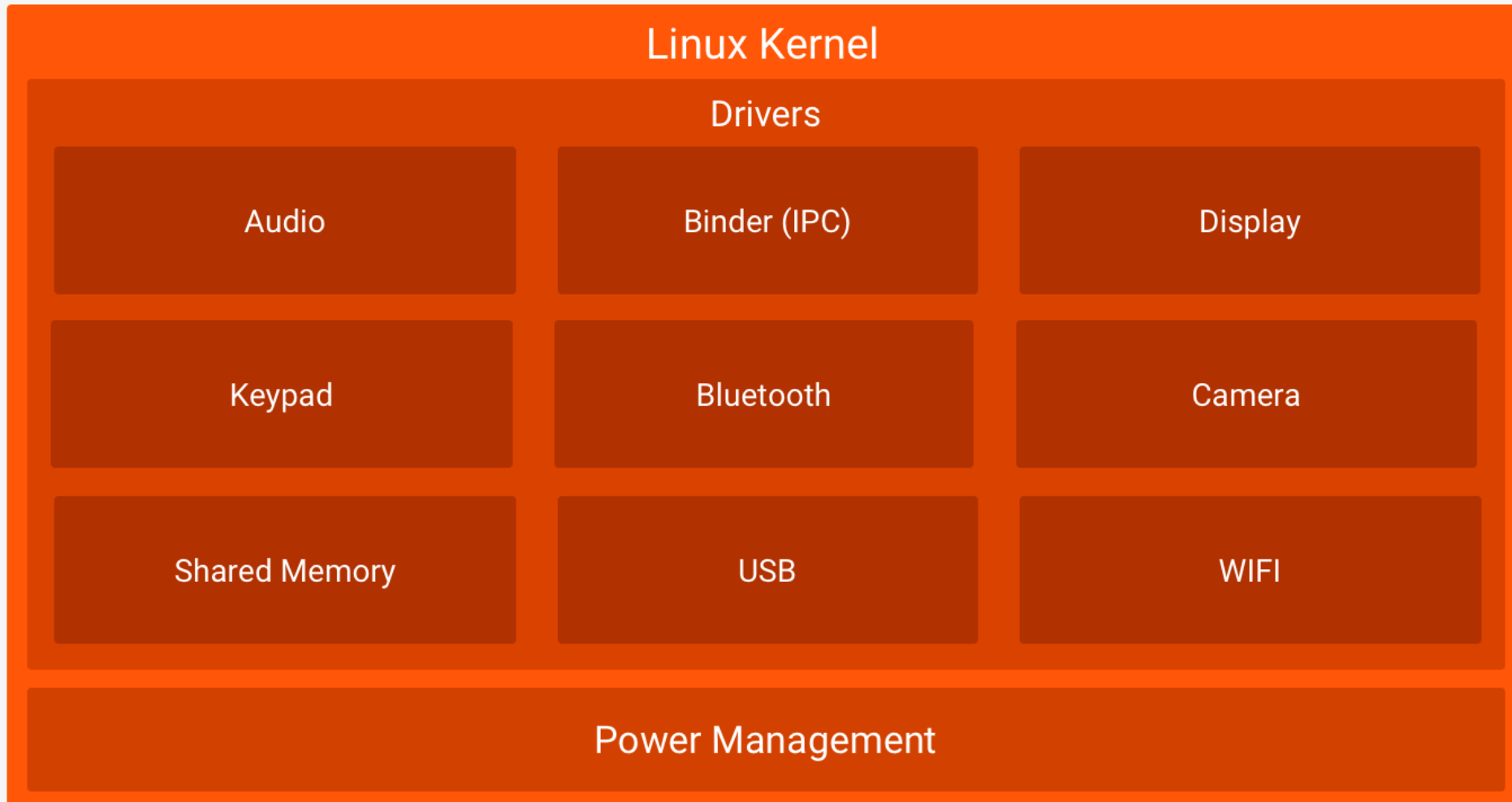
Hardware Abstraction Layer (HAL)

Audio	Bluetooth	Camera	Sensors	. . .

Kernel

Linux Kernel

Drivers

Audio	Binder (IPC)	Display
Keypad	Bluetooth	Camera
Shared Memory	USB	WIFI

Power Management

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.

Android Features

❑ Application framework enabling **reuse** and

 replacement of components

❑ Optimized Java virtual machine: **Dalvik**

❑ Optimized Graphics Processing, supporting 2D

 and 3D graphics (OpenGL ES 1.0)

❑ Integrated open source web browser: **WebKit**

❑ **SQLite** for structured data storage

Android Features (2)

❑ Multimedia capability, supporting varieties of audio, video and still image formats

❑ GSM Telephony

❑ Bluetooth, EDGE, 3G and Wi-Fi support

❑ Camera, GPS, compass, accelerometer and other sensors support

❑ Rich development environment, including an emulator, debugging tools, memory probe tools, log tools and powerful eclipse plug-ins