Mobile Application Development

01 - Architecture

Objectives

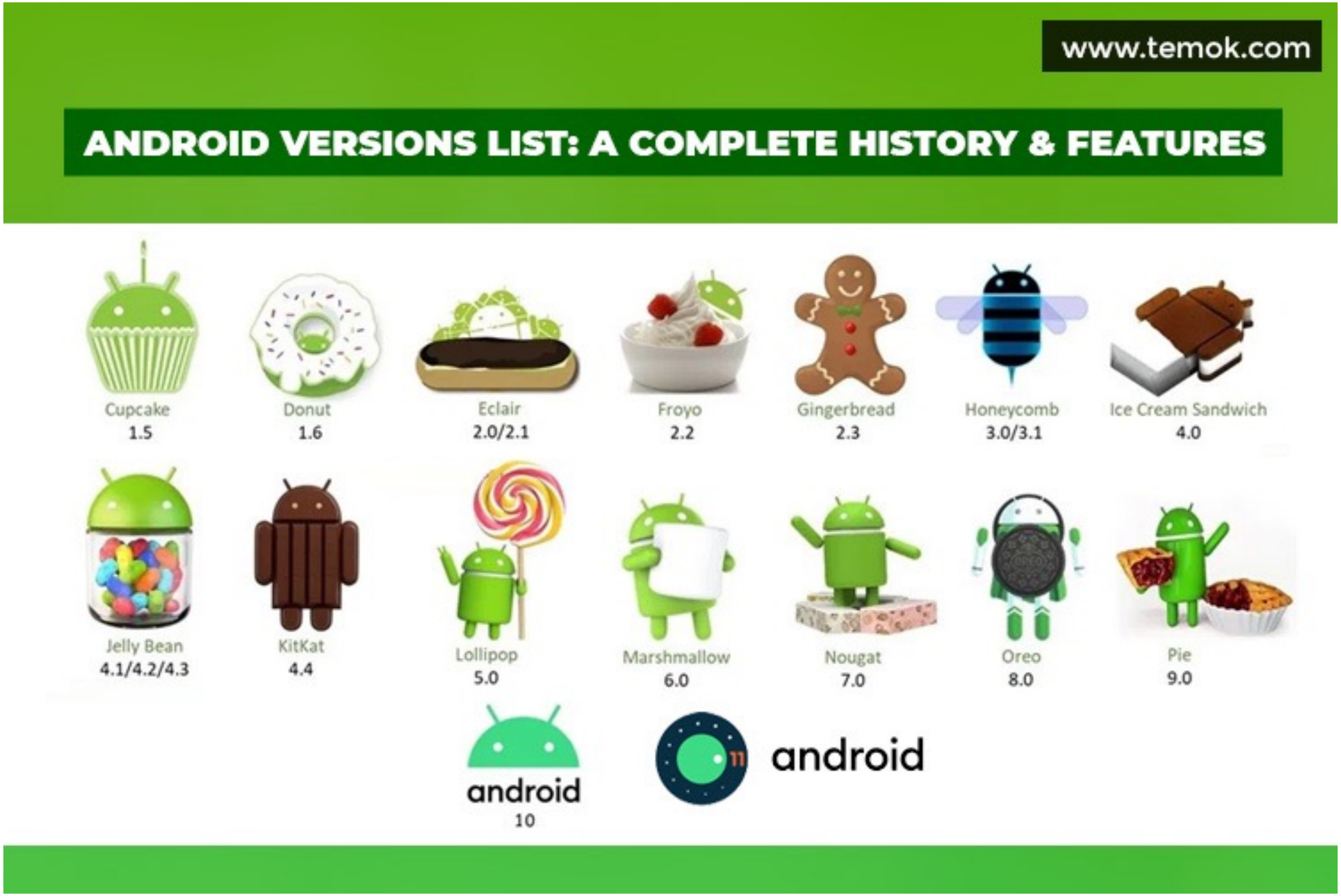
- Overview of Android
- Android Architecture
- Android Application & Components

Overview of Android

- Mobile operating system based on modified version of Linux, designed primarily for devices such as smartphones and tablets.
- Originally developed by a small startup and was acquired by Google in 2005.
- Officially unveiled in 2007 by Open Handset Alliance (handset and chip manufacturers, application developers, mobile carriers).
- Open source nature allows the software to be freely modified and distributed by device manufacturers.
- Android SDK provides the tools and APIs for application development using the Java programming language.

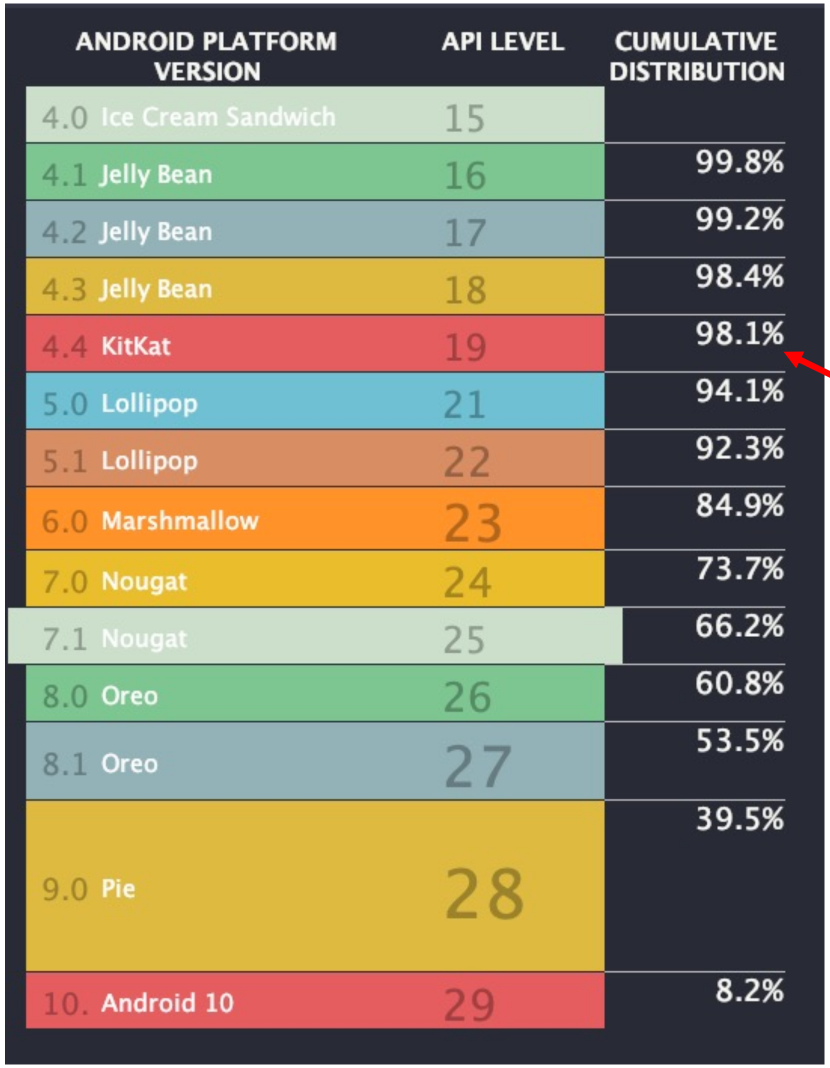
Evolution of Android OS

Name	Version number(s)	Initial stable release date	API level
No official codename	1.0	September 23, 2008	1
	1.1	February 9, 2009	2
Cupcake	1.5	April 27, 2009	3
Donut	1.6	September 15, 2009	4
Eclair	2.0 – 2.1	October 26, 2009	5 – 7
Froyo	2.2 – 2.2.3	May 20, 2010	8
Gingerbread	2.3 – 2.3.7	December 6, 2010	9 – 10
Honeycomb	3.0 – 3.2.6	February 22, 2011	11 – 13
Ice cream sandwich	4.0 – 4.0.4	October 18, 2011	14 – 15
Jelly Bean	4.1 – 4.3.1	July 9, 2012	16 – 18
Kitkat	4.4 – 4.4.4	October 31, 2013	19 – 20
Lollipop	5.0 – 5.1.1	November 12, 2014	21 – 22
Marshmallow	6.0 – 6.0.1	October 5, 2015	23
Nougat	7.0 – 7.1.2	August 22, 2016	24 – 25
Oreo	8.0 – 8.1	August 21, 2017	26 – 27
Pie	9	August 6, 2018	28
Android 10	10	September 3, 2019	29
Android 11	11	September 8, 2020	30



https://www.temok.com/blog/android-version-list/

Distribution

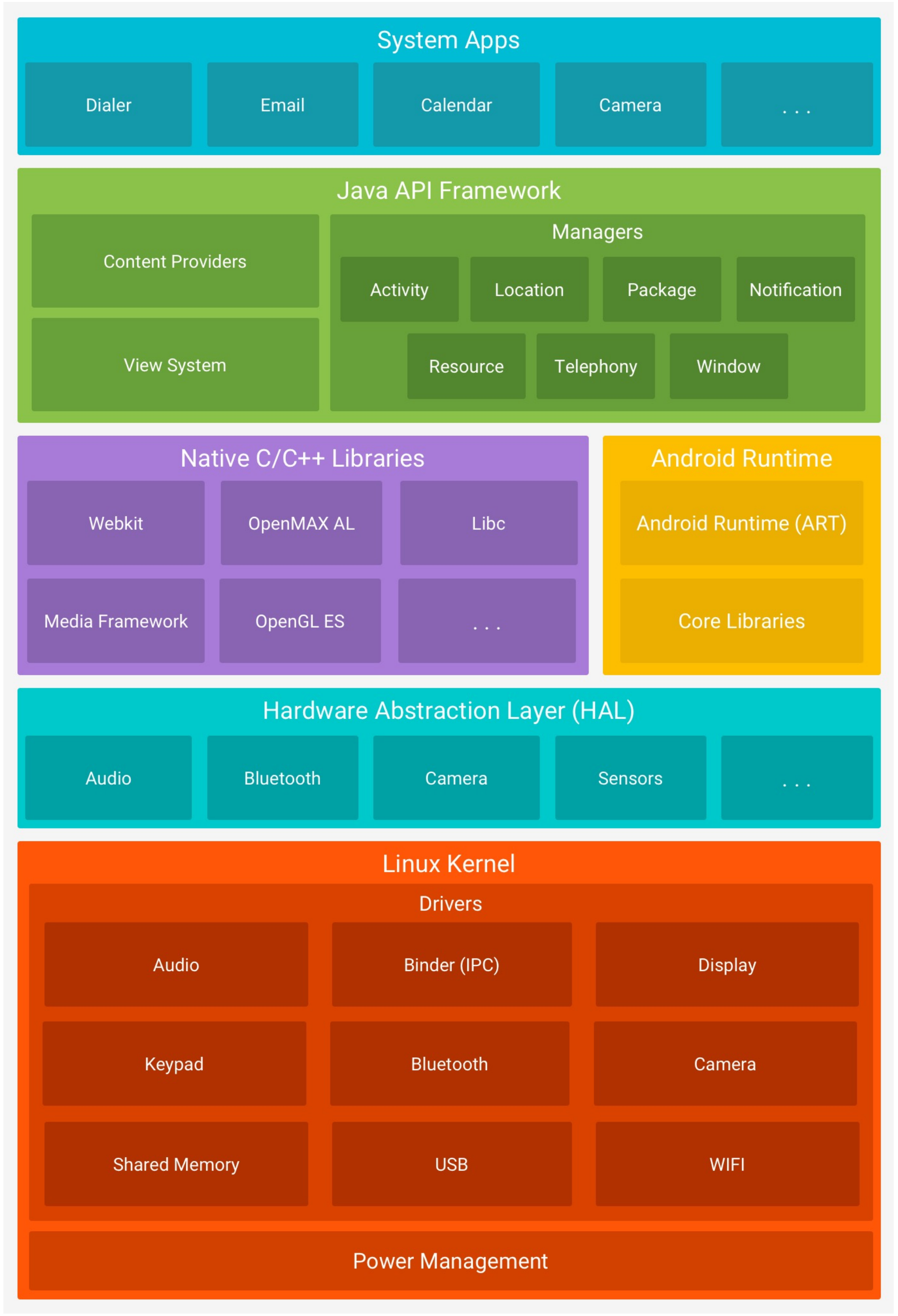


ANDROID PLATFORM VERSION	API LEVEL	CUMULATIVE DISTRIBUTION
4.0 Ice Cream Sandwich	15	
4.1 Jelly Bean	16	99.8%
4.2 Jelly Bean	17	99.2%
4.3 Jelly Bean	18	98.4%
4.4 KitKat	19	98.1%
5.0 Lollipop	21	94.1%
5.1 Lollipop	22	92.3%
6.0 Marshmallow	23	84.9%
7.0 Nougat	24	73.7%
7.1 Nougat	25	66.2%
8.0 Oreo	26	60.8%
8.1 Oreo	27	53.5%
9.0 Pie	28	39.5%
10. Android 10	29	8.2%

% of users that are able to run your app.

Android Software stack

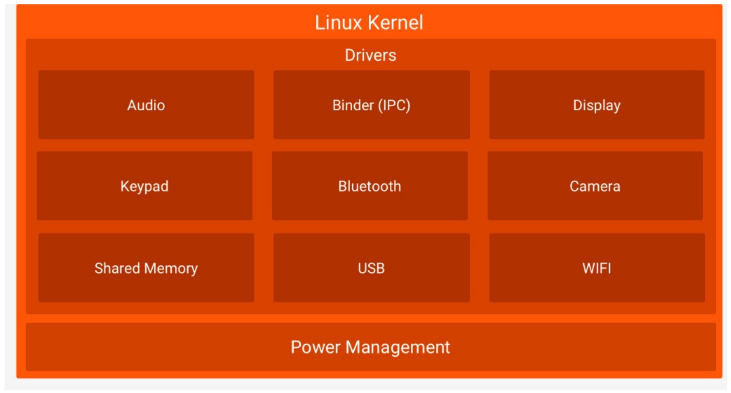
- Android is a complete software stack that includes:
  - Operating System (OS)
  - Middleware
  - Applications



Linux Kernel

- The foundation of the Android platform is the Linux kernel.

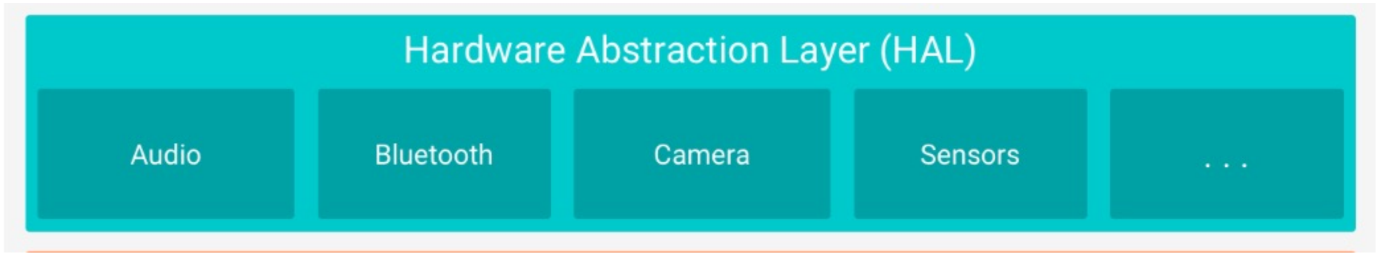
- Core system services such as
  - Security
  - Memory management
  - Process management
  - Network stack
  - Driver model



- 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.

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.



Android Runtime

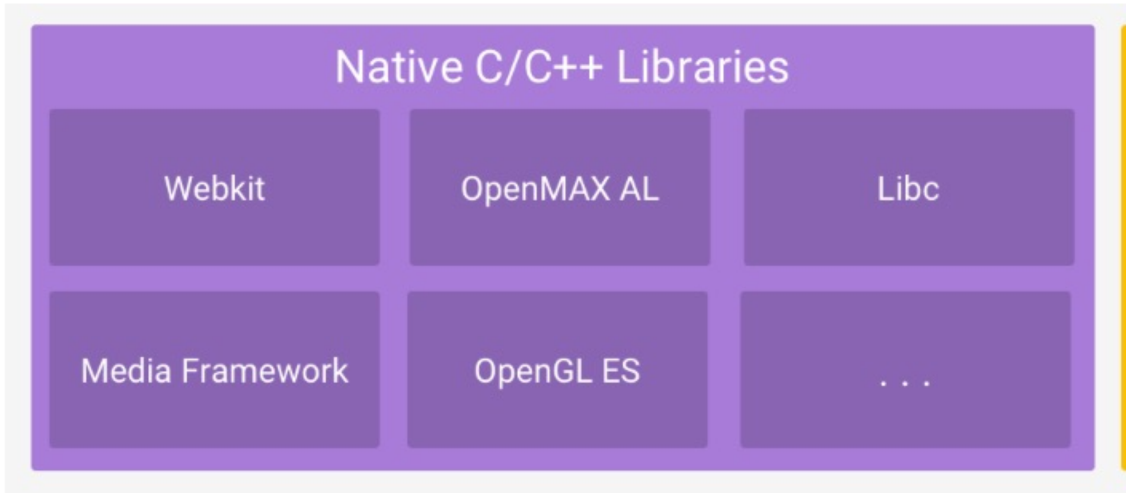
- For devices running Android version 5.0 (API level 21) or higher, each app runs in its own process and with its own instance of the Android Runtime (ART).
- ART is written to run multiple virtual machines on low-memory devices by executing DEX files, a bytecode format designed specially for Android that's optimized for minimal memory footprint. Build toolchains, such as Jack, compile Java sources into DEX bytecode, which can run on the Android platform.
- Some of the major features of ART include the following:
  - Ahead-of-time (AOT) and just-in-time (JIT) compilation
  - Optimized garbage collection (GC)
  - Better debugging support, including a dedicated sampling profiler, detailed diagnostic exceptions and crash reporting, and the ability to set watchpoints to monitor specific fields

Android Runtime

- Prior to Android version 5.0 (API level 21), Dalvik was the Android runtime. If your app runs well on ART, then it should work on Dalvik as well, but the reverse may not be true.
- Android also includes a set of core runtime libraries that provide most of the functionality of the Java programming language, including some Java 8 langauge features, that the Java API framework uses.

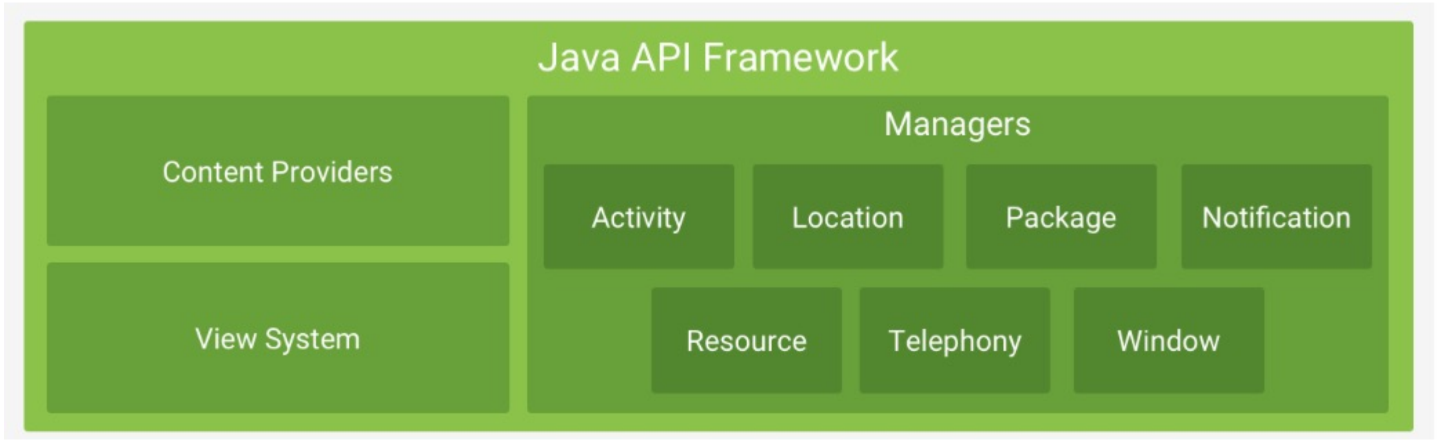
Native C / C++ Libraries

- 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.



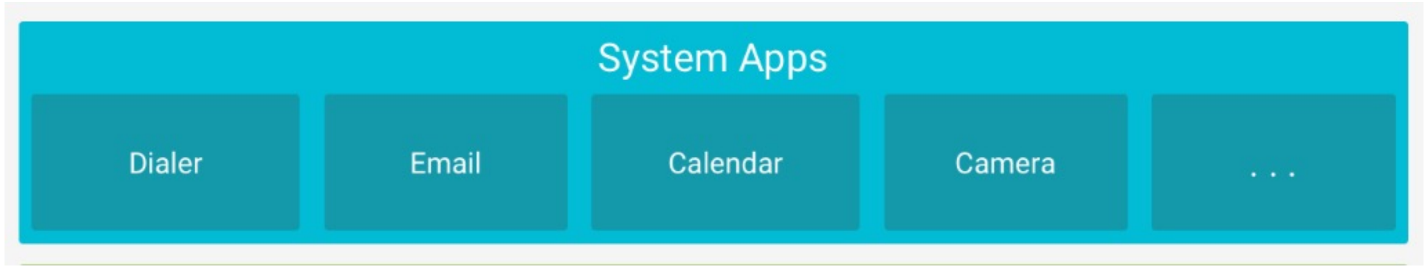
Java API Framework

- 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, which include the following:
  - 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
- Developers have full access to the same framework APIs that Android system apps use.



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.
    - So 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).
- 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.


System Apps
Dialer
Email
Calendar
Camera
. . .