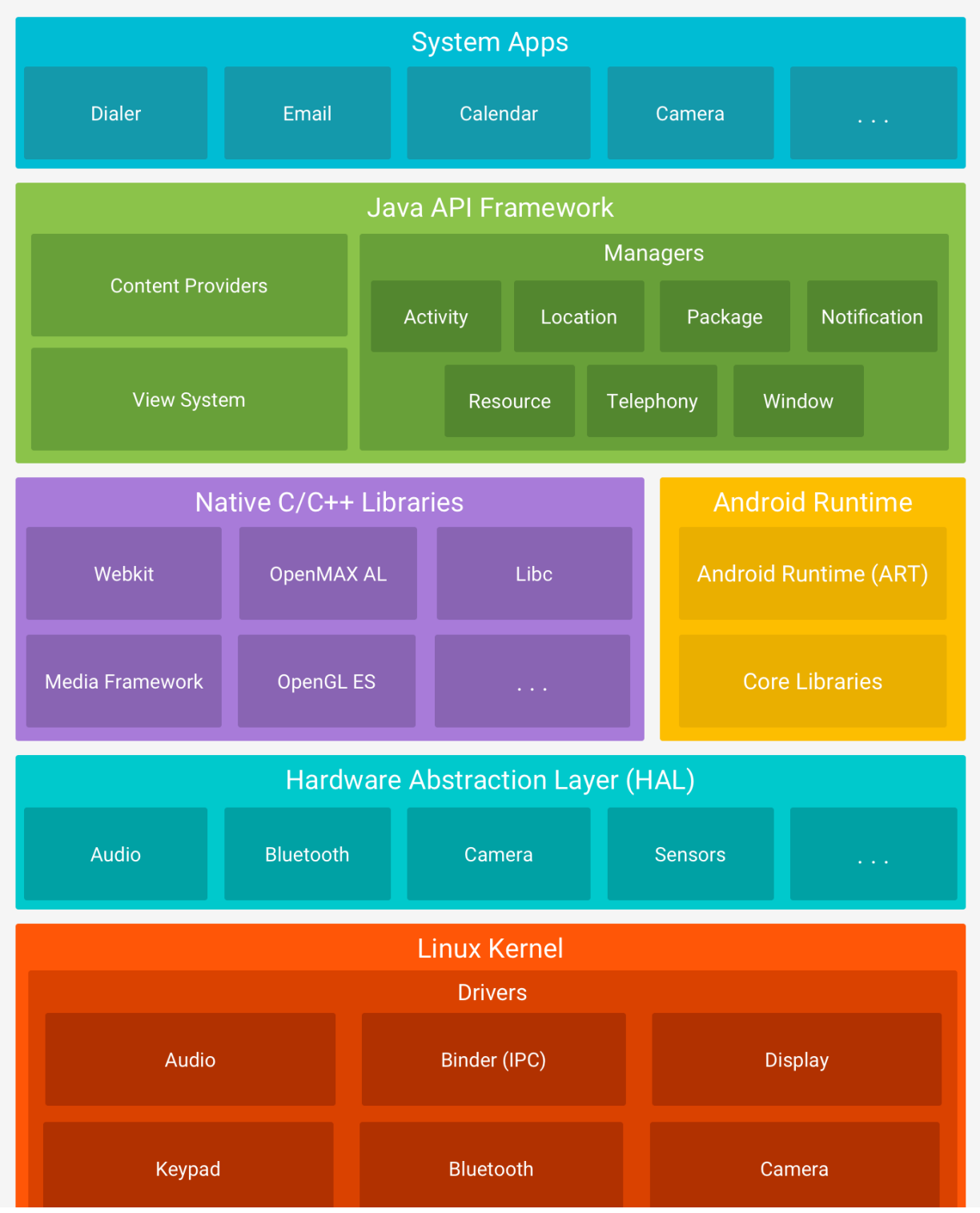
ANDROID PLATFORM ARCHITECTURE AND ACTIVITY: -



KERNEL

The foundation of the Android platform is the Linux kernel , kernel facilitates interactions between hardware and software components

HAL :

The hardware abstraction layer (HAL) provides standard interfaces that expose device hardware capabilities to the higher-level Java API framework

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 tools, such as d8, compile Java sources into DEX bytecode, which can run on the Android platform.

- Ahead-of-time (AOT) and just-in-time (JIT) compilation

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++.

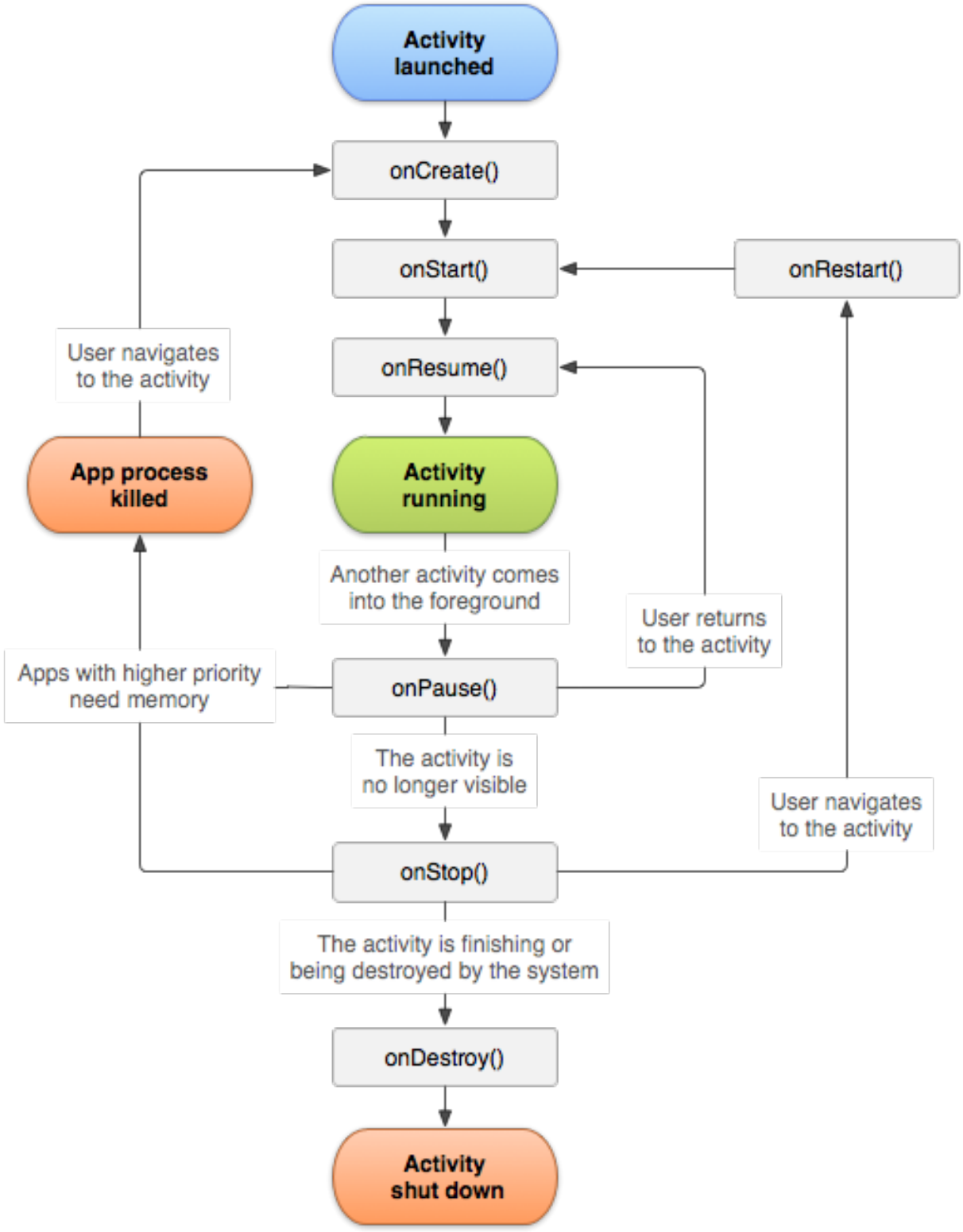
Java API Framework

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

System Apps
Android comes with a set of core apps for email, SMS messaging, calendars, internet browsing, contacts, and more.

ACTIVITY:
An *activity* is the entry point for interacting with the user. It represents a single screen with a user interface.
An activity provides the window in which the app draws its UI.



ACTIVITY NAVIGATION:

The Navigation component consists of three key parts that are

described below:

- Navigation graph: An XML resource that contains all navigation-related information in one centralized location. This includes all of the individual content areas within your app, called *destinations*, as well as the possible paths that a user can take through your app.
- NavHost: An empty container that displays destinations from your navigation graph. The Navigation component contains a default NavHost implementation, NavHostFragment, that displays fragment destinations.
- NavController: An object that manages app navigation within a NavHost. The NavController orchestrates the swapping of destination content in the NavHost as users move throughout your app.
