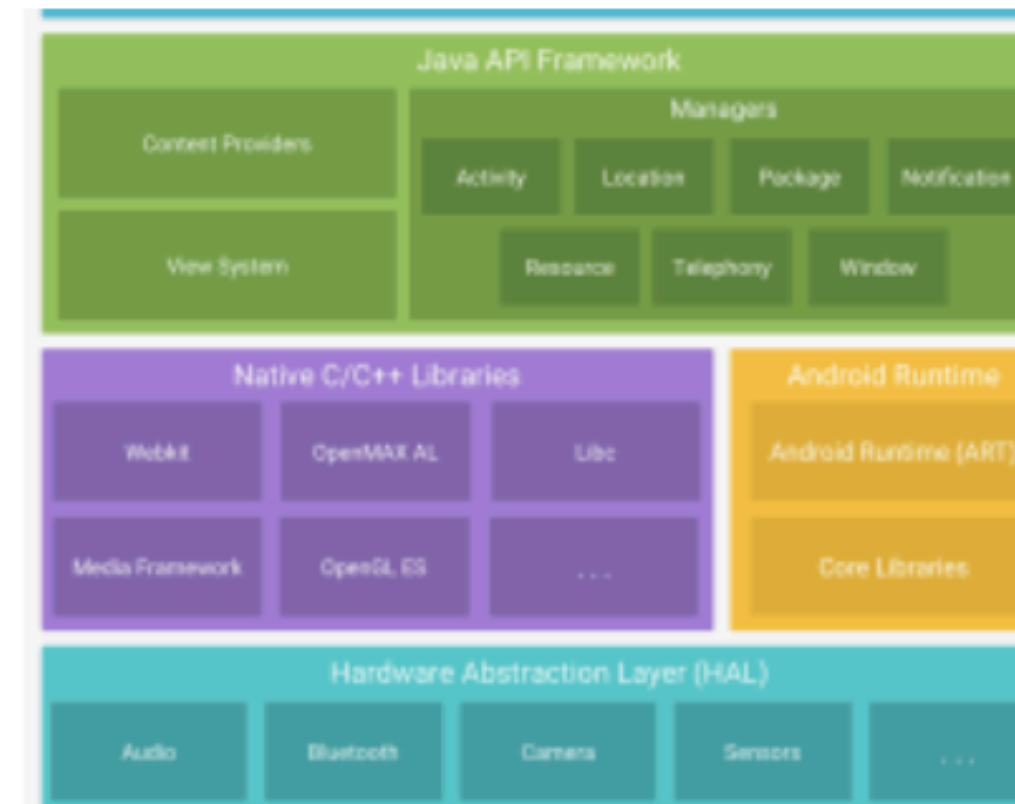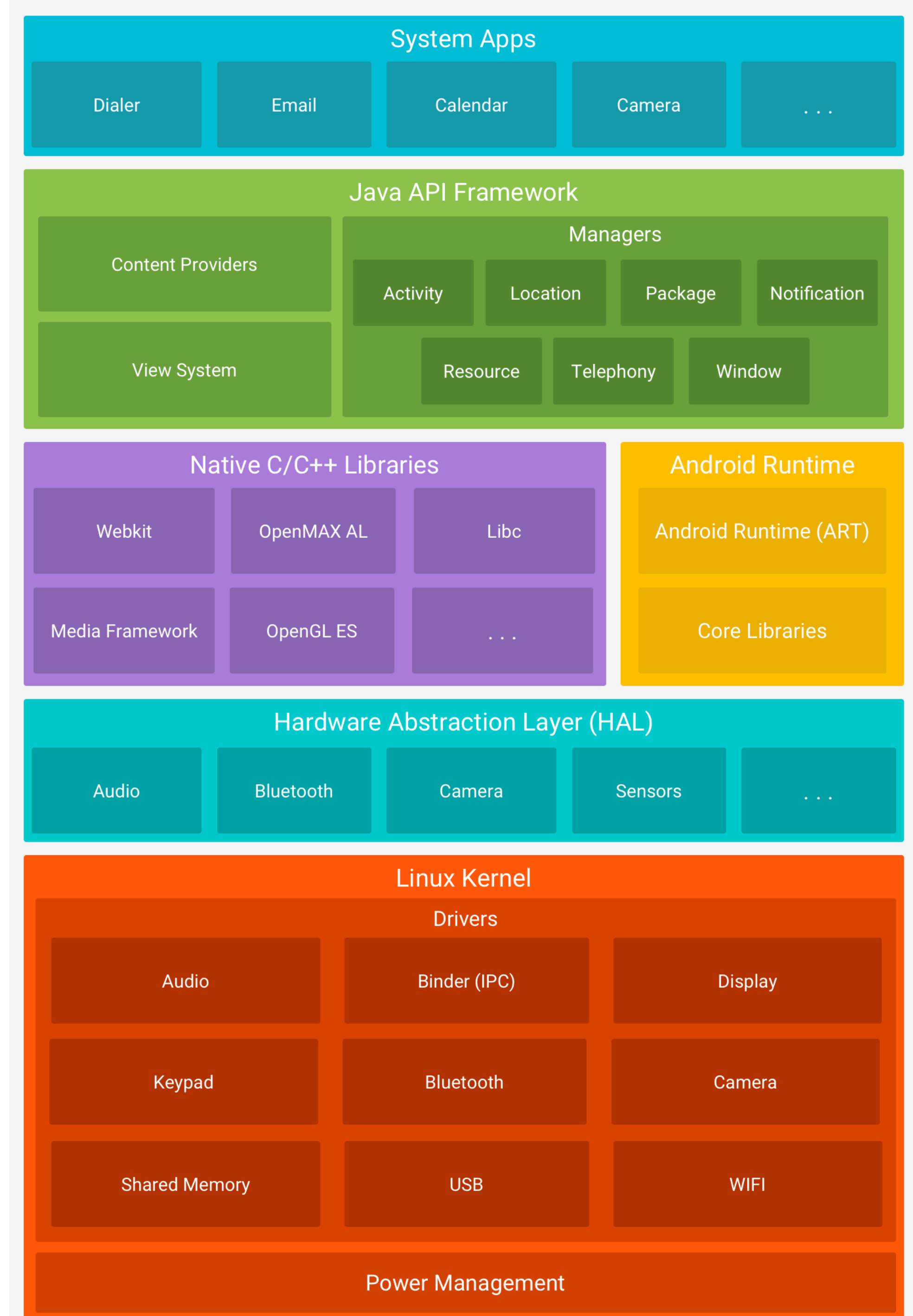# Android Platform, Compnents & Activities

Android Platform, Components & Activities

Key features of the platform, the principle components + the activity model

Platform Fundamentals

Android is an open source, Linux-based software stack created for a wide array of devices and form factors.
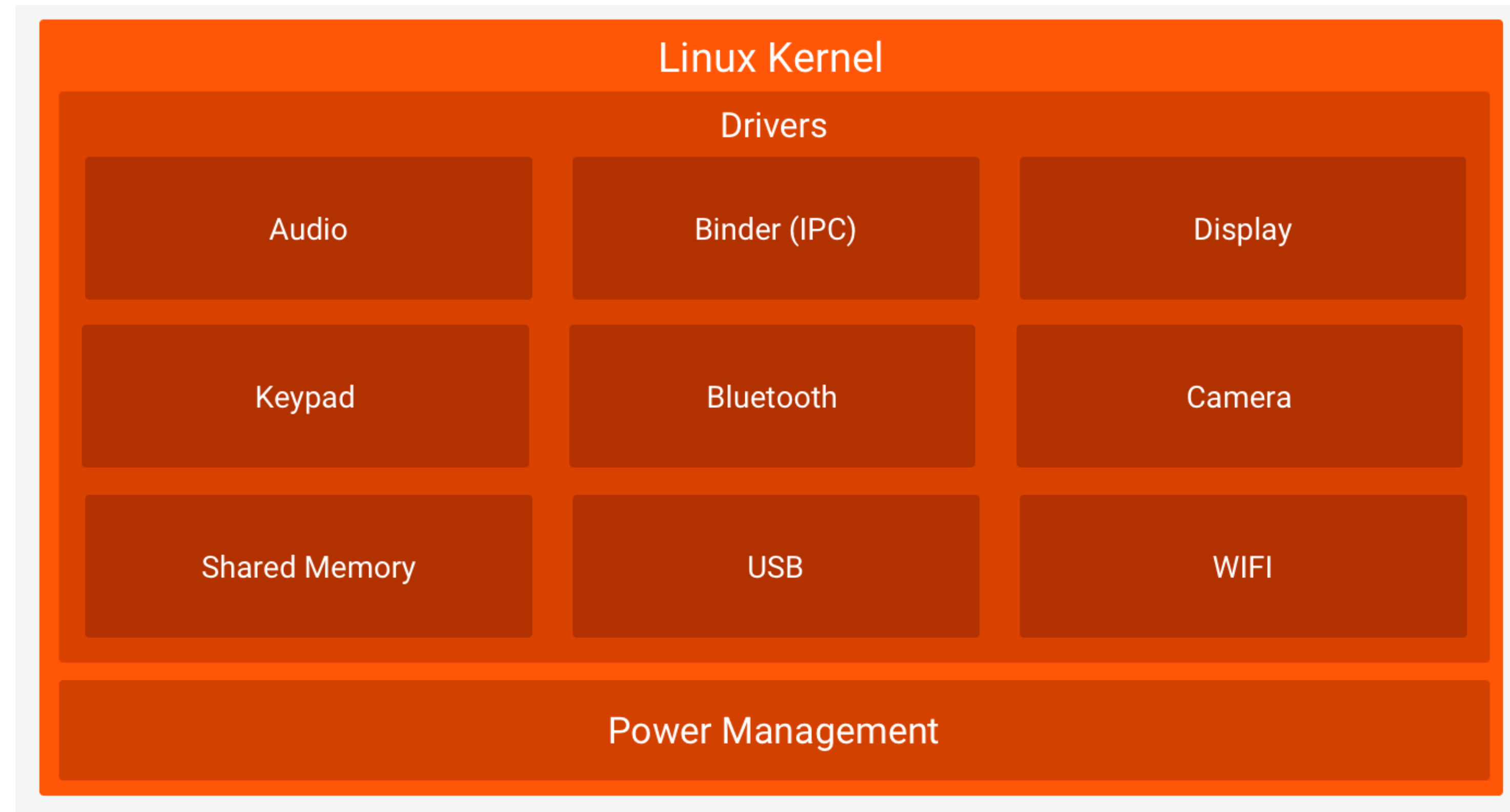
The Linux Kernel

The foundation of the Android platform

Allows Android to take advantage of key security features.

Allows device manufacturers to develop hardware drivers for a well-known kernel

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

Hardware Abstraction Layer (HAL)

Standard interfaces that expose device hardware capabilities to the higher-level Java API framework.

Multiple modules, each implementing an interface for a specific type of hardware component.

Hardware Abstraction Layer (HAL)

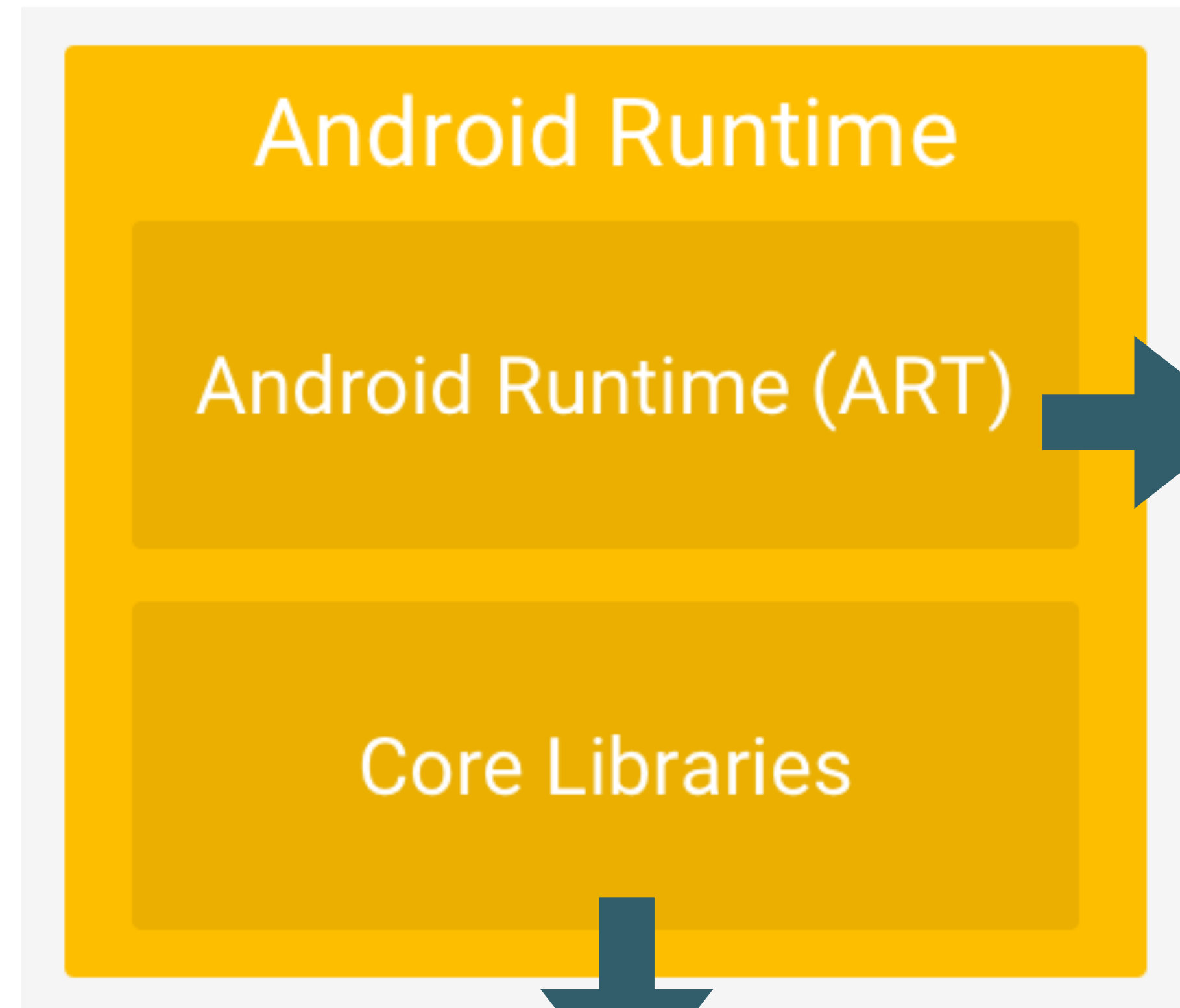| Audio	Bluetooth	Camera	Sensors	. . .
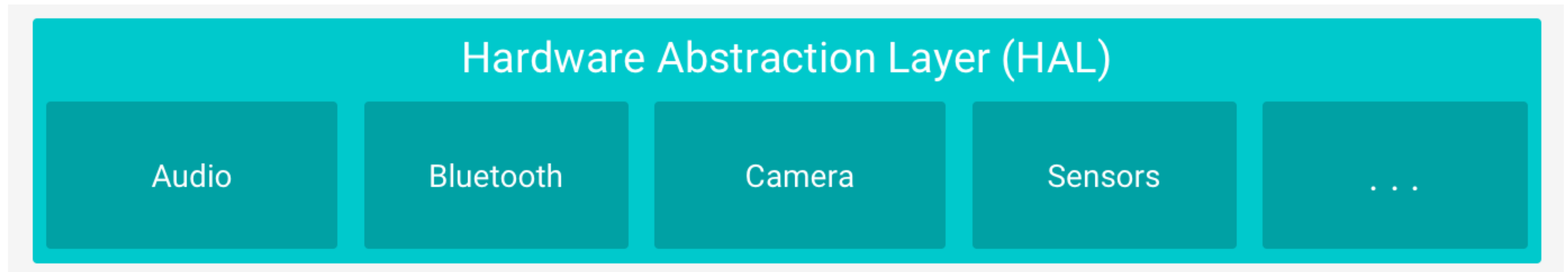
Module loaded on demand

Android Runtime - ART

ART Features

Each app runs in its own process and with its own instance of the ART.

ART executes DEX files (Dalvik Executable format)

Dex is a bytecode format designed optimized for minimal memory footprint

Android Runtime

Android Runtime (ART)

Core Libraries

- Converts DEX files to more compact machine code via Ahead-of-time (AOT) and just-in-time (JIT) compilation

- Optimized garbage collection (GC)

- Debugging support, profiler, diagnostic and crash reporting

Provide most of the JDK features

Native C/C++ Libraries

Many core components are built from native code that require native libraries written in C and C++.

Some of these exposed as Java framework APIs
E.G:

Native C/C++ Libraries

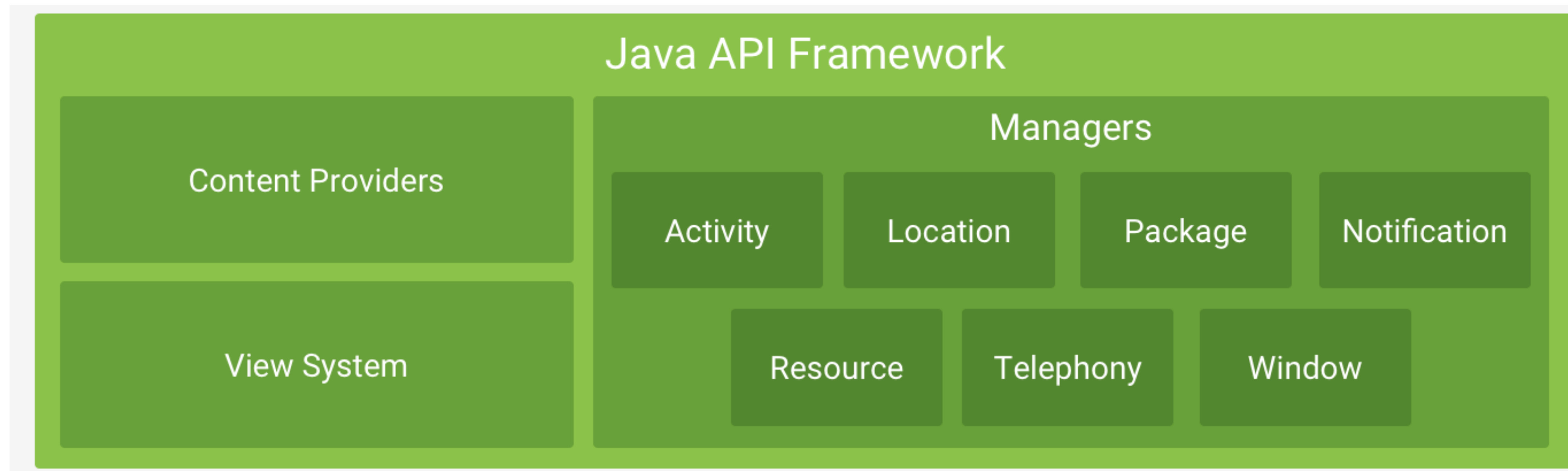| Webkit | OpenMAX AL | Libc |
| Media Framework | OpenGL ES | . . . |

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.
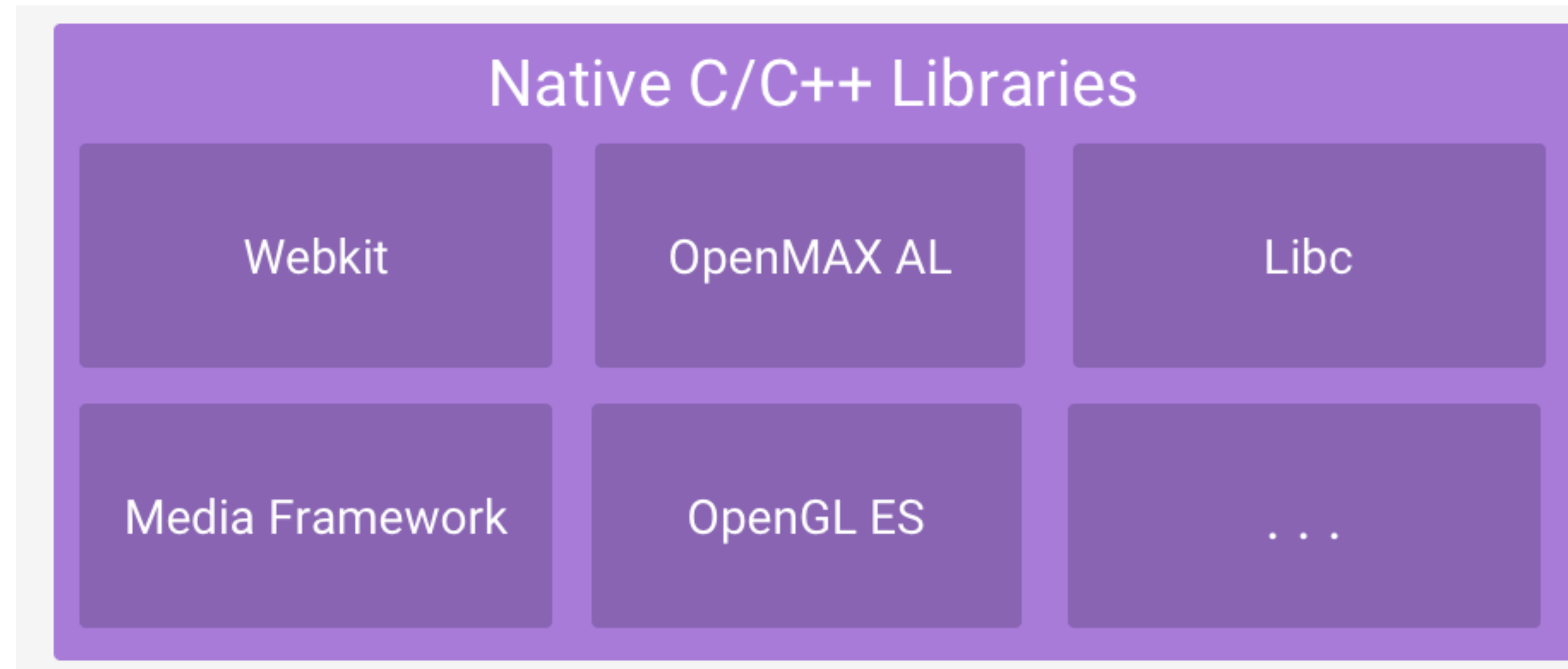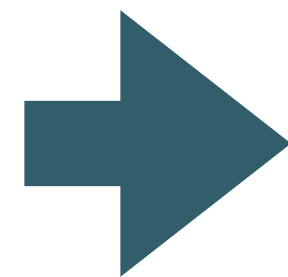
Java API Framework

Android OS APIs

The building blocks for Android apps

Encapsulates lower level components and services

- **View System:** to build an app's UI, including lists, grids, text boxes, buttons, and even an embeddable web browser
- **Resource Manager:** providing access to non-code resources such as localized strings, graphics, and layout files
- **Notification Manager** : enables all apps to display custom alerts in the status bar
- **Activity Manager:** manages the lifecycle of apps and provides a common navigation back stack
- **Content Providers:** enable apps to access data from other apps (eg)Contacts app) or to share their own data
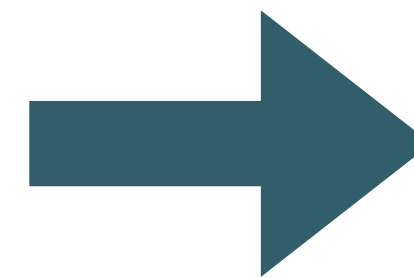
System Apps
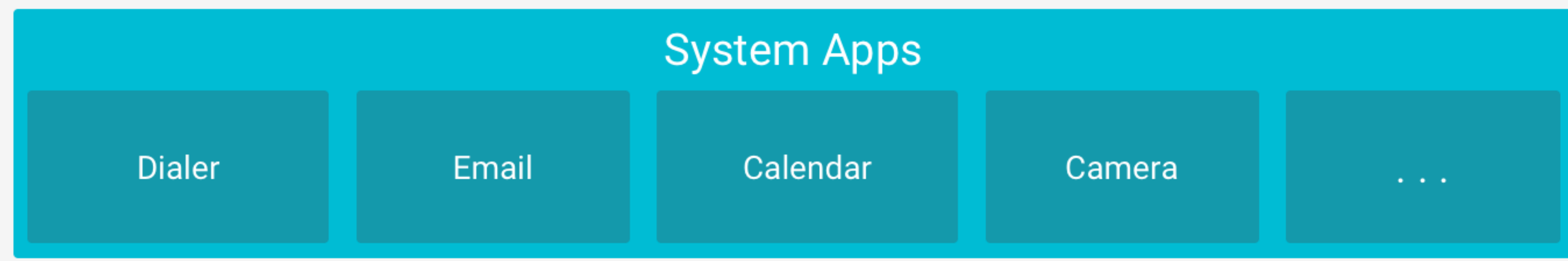
A set of core apps for email, SMS messaging, calendars, internet browsing, contacts, etc..

Apps have no special status among the apps the user chooses to install.

These are app for users + they provide capabilities that developers can access.
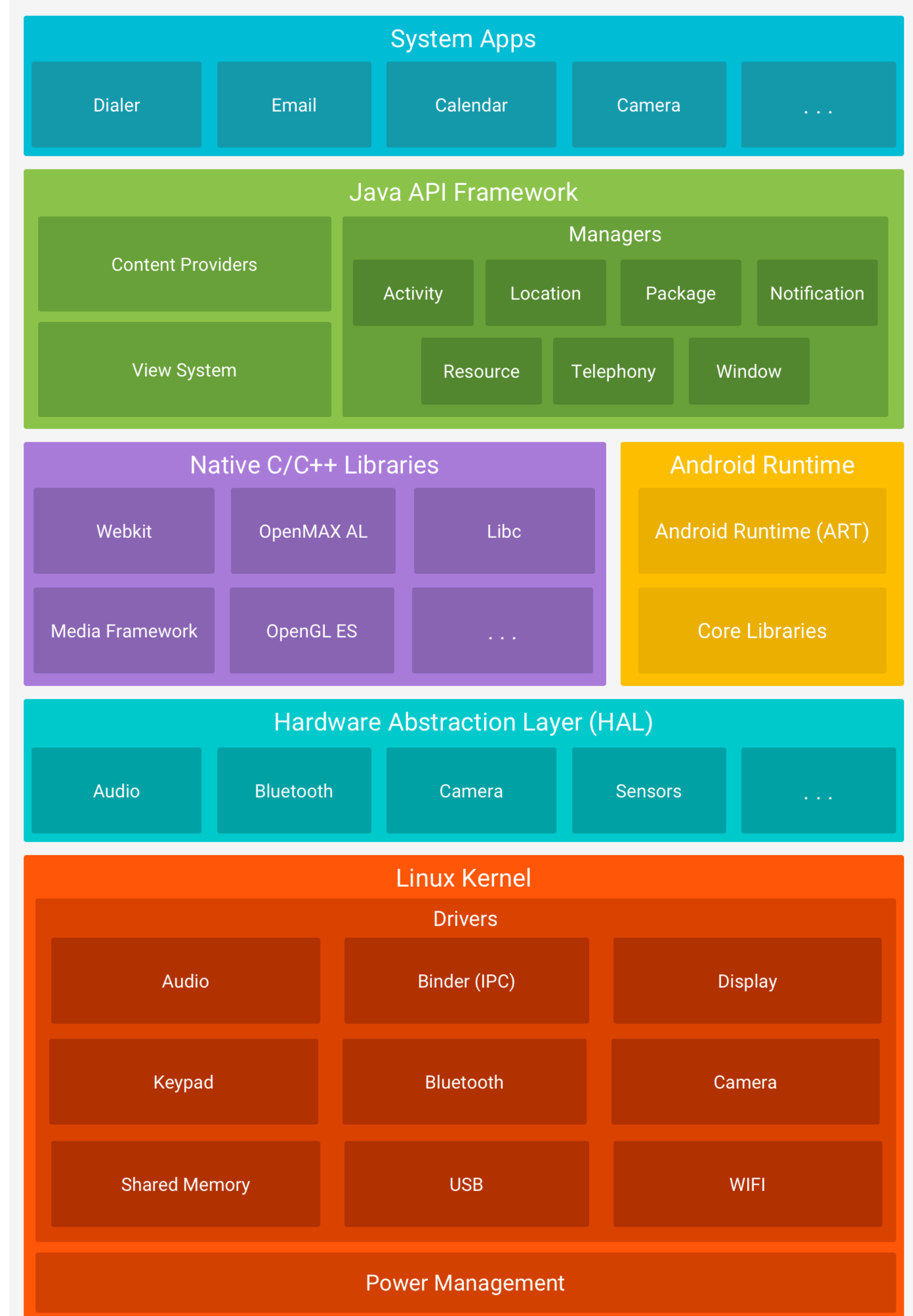
A third-party app can become the user's default web browser, SMS messenger, or even the default keyboard (some exceptions apply, such as the system's Settings app).

Platform Fundamentals

Android is an open source, Linux-based software stack created for a wide array of devices and form factors.
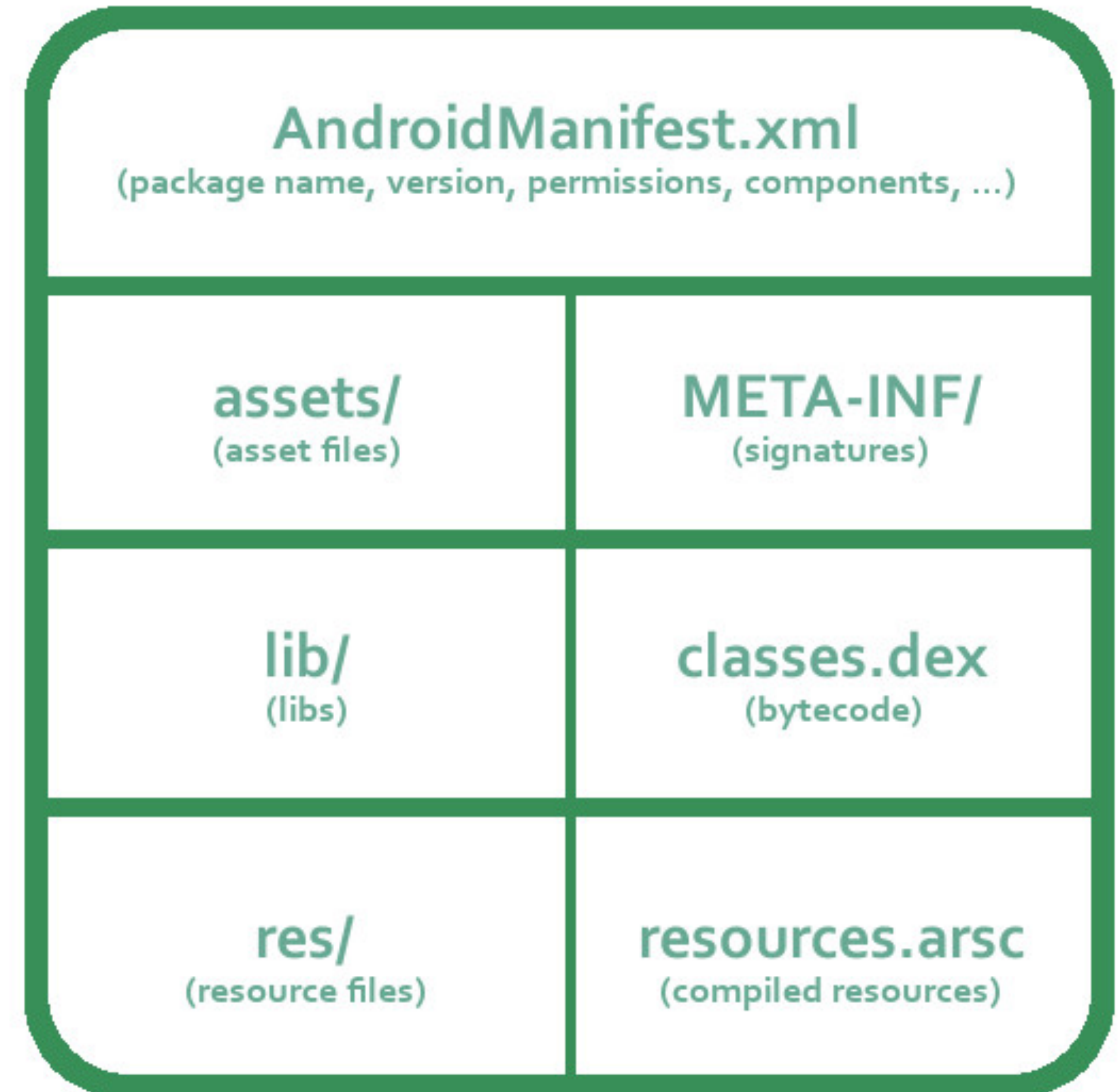
Android Application Components

Android Application

Written using Kotlin, Java, and C++ languages.

The Android SDK tools compile code along with any data and resource files into an APK, an Android package, which is an archive file with an .apk suffix.

One APK file contains all the contents of an Android app and is the file that Android-powered devices use to install the app.

APK

AndroidManifest.xml
(package name, version, permissions, components, …)

assets/	META-INF/
(asset files)	(signatures)
lib/	classes.dex
(libs)	(bytecode)
res/	resources.arsc
(resource files)	(compiled resources)

Security Model

Each app lives in its own security sandbox, protected a range of security features.

- Assigned a unique Linux user ID. The system sets permissions for all the files in an app so that only the user ID assigned to that app can access them.

- Wach app es a process with its own virtual machine (VM), so an app's code runs in isolation from other apps.

App is started when any of its components needs to be executed

App is shutdown when it's no longer needed or when the system must recover memory for other apps.

Principle of Least Privilege.

Each app, by default, has access only to the components that it requires to do its work and no more.

This creates a secure environment in which an app cannot access parts of the system for which it is not given permission.

However, there are ways for an app to share data with other apps and for an app to access system services:

- An app can request permission to access device data such as the user's contacts, SMS messages, the mountable storage (SD card), camera, and Bluetooth.

- The user has to explicitly grant these permissions.
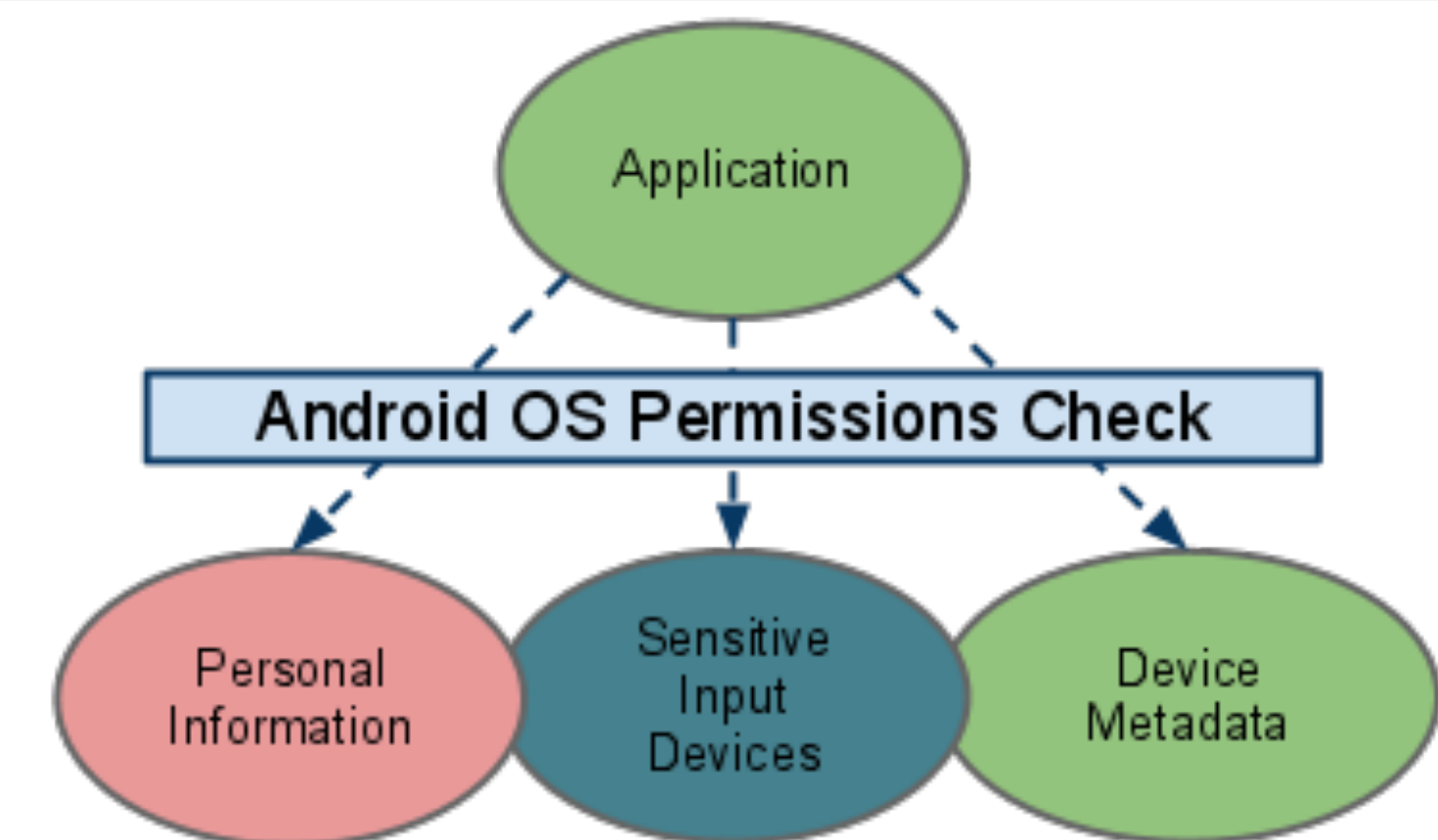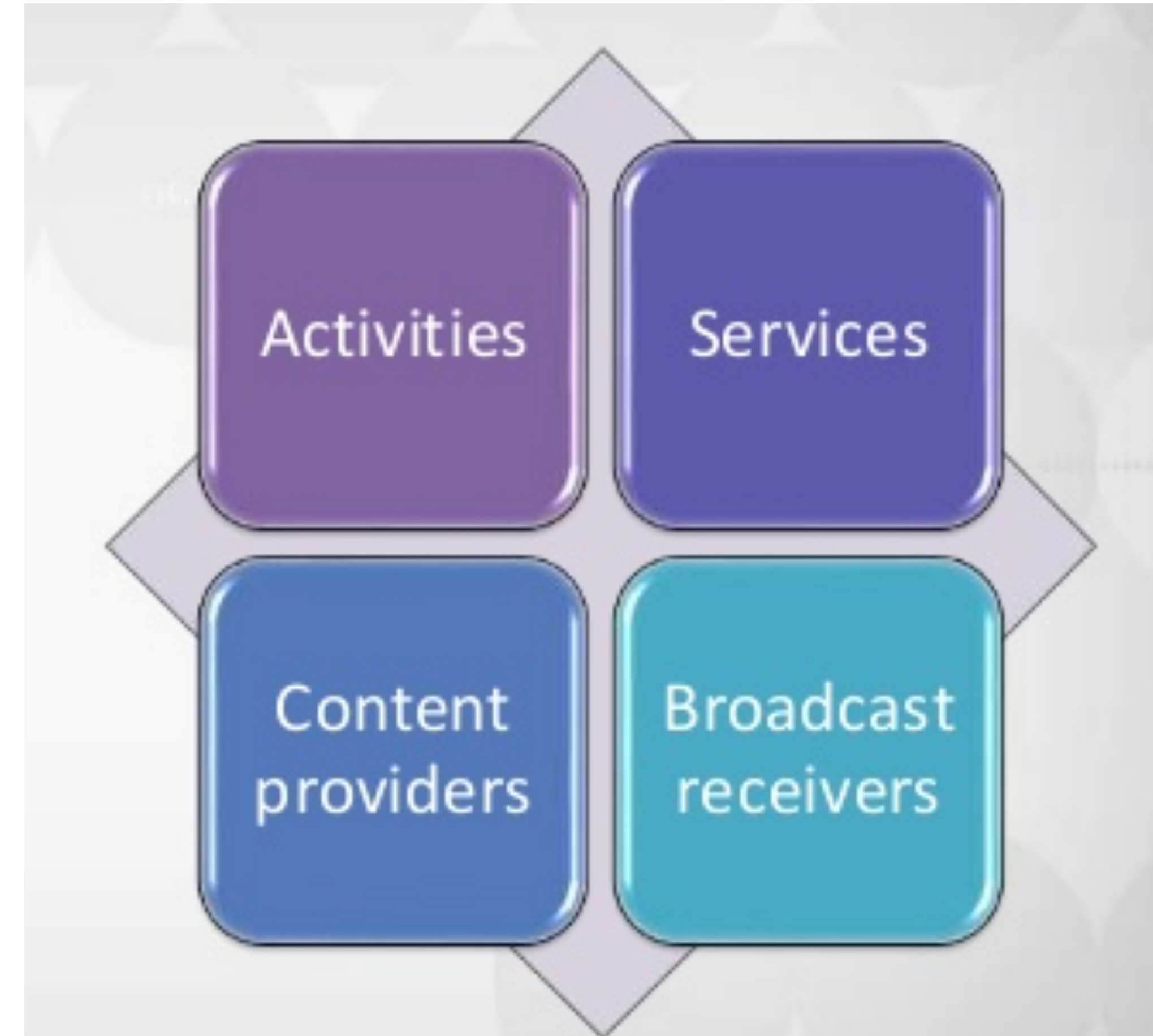

Types of Sensitive User Data

Application Components

4 Types

App components are the essential building blocks of an Android app.

Each component is an entry point through which the system or a user can enter the app

Each type serves a distinct purpose and has a distinct lifecycle that defines how the component is created and destroyed

Android Components

Application components are the essential building blocks of an Android application.

Activities

An activity represents a single screen with a user interface, in-short Activity performs actions on the screen.

Services

A service is a component that runs in the background to perform long-running operations.

Broadcast Receivers

Broadcast Receivers simply respond to broadcast messages from other applications or from the system.

Content Providers

A content provider component supplies data from one application to others on request.

Additional Components

There are additional components which will be used in the construction of above mentioned entities.
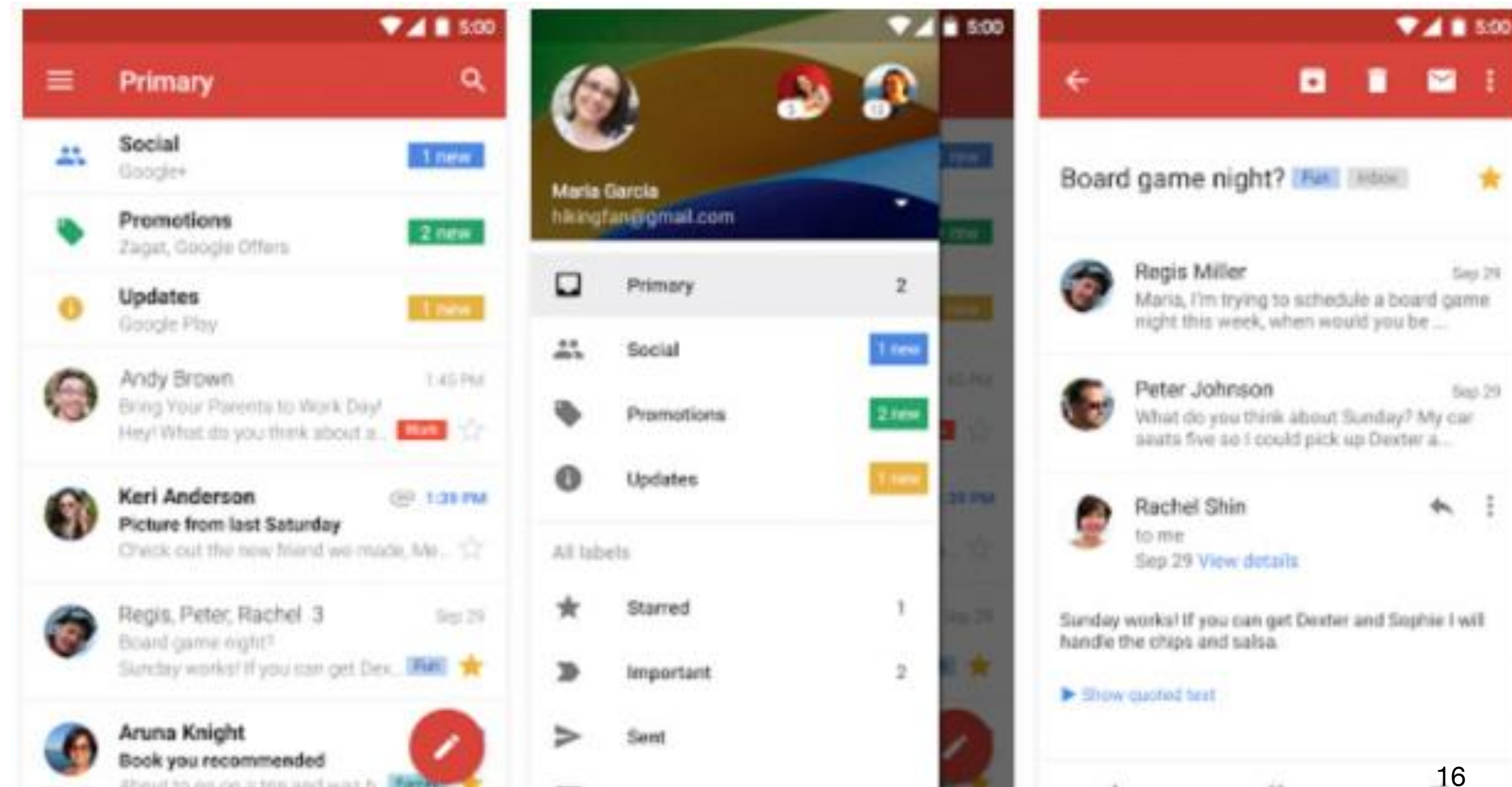
Activities I

An activity is the entry point for interacting with the user.

It represents a single screen with a user interface.

Activities work together to form a cohesive user experience

An email app might have one activity that shows a list of new emails, another activity to compose an email, and another activity for reading emails

16

Activities II

Although the activities work together to form a cohesive user experience each one is independent of the others.

If permitted, a different app can start activity in another app.

A camera app can start the activity in the email app that composes new mail to allow the user to share a picture

Activity Lifecycle

To navigate transitions between stages of the activity lifecycle, the Activity class provides a core set of six callbacks:

onCreate(),
onStart(),
onResume(),
onPause(),
onStop(),
onDestroy().

The system invokes each of these callbacks as an activity enters a new state.