

Platform-Based Development: Android Programming – Architecture

BS UNI studies, Spring 2019/2020

Dr Veljko Pejović
Veljko.Pejovic@fri.uni-lj.si



The World of Android

- The Android Platform
 - A mobile operating system + libraries + application frameworks + key apps
 - Based on Linux
 - Open source
 - Runs on a range of devices
 - Some with OEM versions
- Market share ~ 75% worldwide
- Android SDK for creating apps
 - Lots of documentation
 - Huge community



Android Versions



Cupcake
1.5



Donut
1.6



Eclair
2.0/2.1



Froyo
2.2



Gingerbread
2.3



Honeycomb
3.0/3.1



Ice Cream Sandwich
4.0

Android
Versions
List
#hickart



Jelly Bean
4.1/4.2/4.3



KitKat
4.4



Lollipop
5.0



Marshmallow
6.0



Nougat
7.0



Oreo
8.0



Pie
9.0

API 21

API 22

API 23

API 24

API 25

API 26

API 27

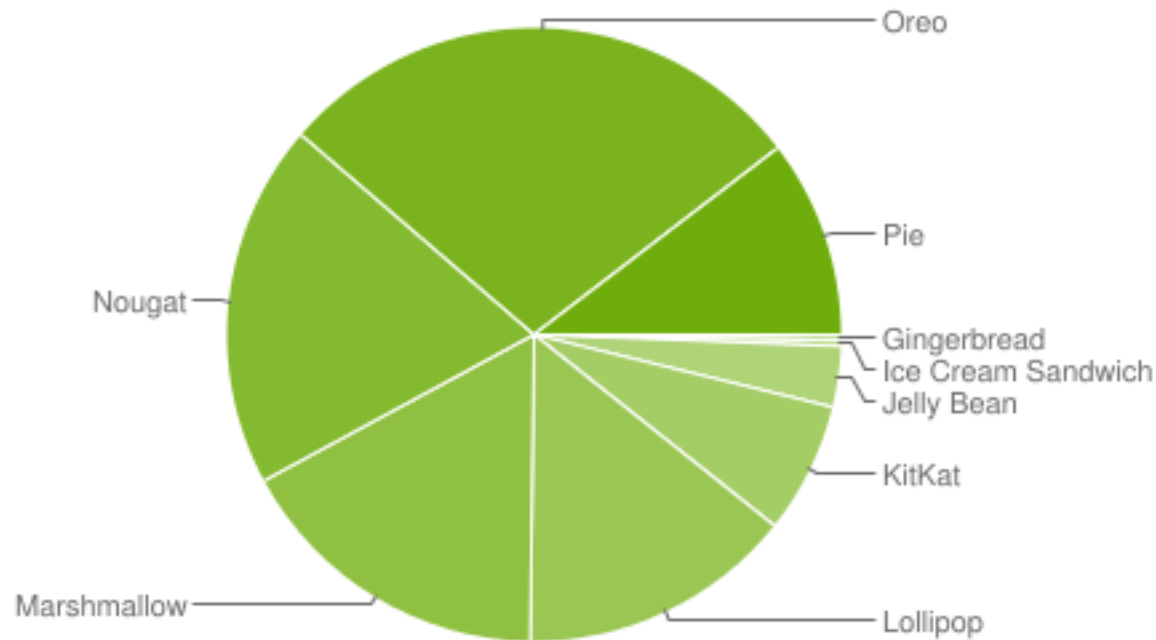
API 28

API 29

Android
10



Android Version Market Share



Data collected during a 7-day period ending on May 7, 2019.

Any versions with less than 0.1% distribution are not shown.

Check current stats at <https://developer.android.com/about/dashboards>

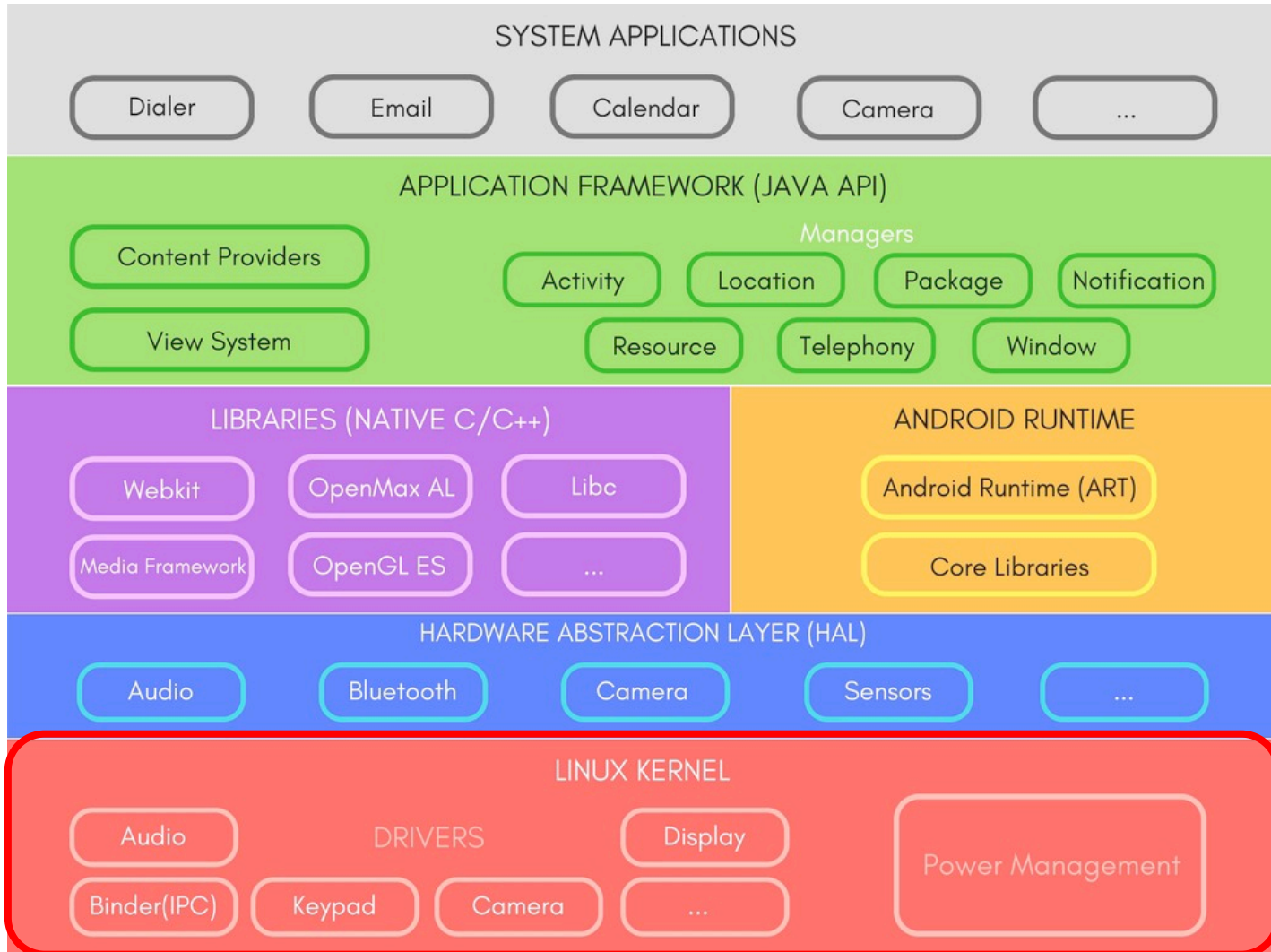


Key Android Features

- Process management specifically tailored for battery-powered devices
 - When an app is not used, it gets suspended by Android
- Process management specifically tailored for low-memory devices
 - When the memory is low, suspended apps are terminated
- Support for direct manipulation interfaces
 - Touchscreen gestures, sensors, notifications
- Open ecosystem of applications
 - Support for developing and distributing Android apps



Android Architecture



Linux Kernel Services

- Memory and process management
 - Usually one process per app
 - Processes are allocated a certain amount of memory (you may get `OutOfMemoryError` in your app)
 - **Android automatically manages the process lifecycle**
 - Apps can be in the foreground (visible) or in the background
 - Background apps can be terminated when the device needs more memory
 - Apps that use more memory are the first to be terminated
 - Interprocess communication



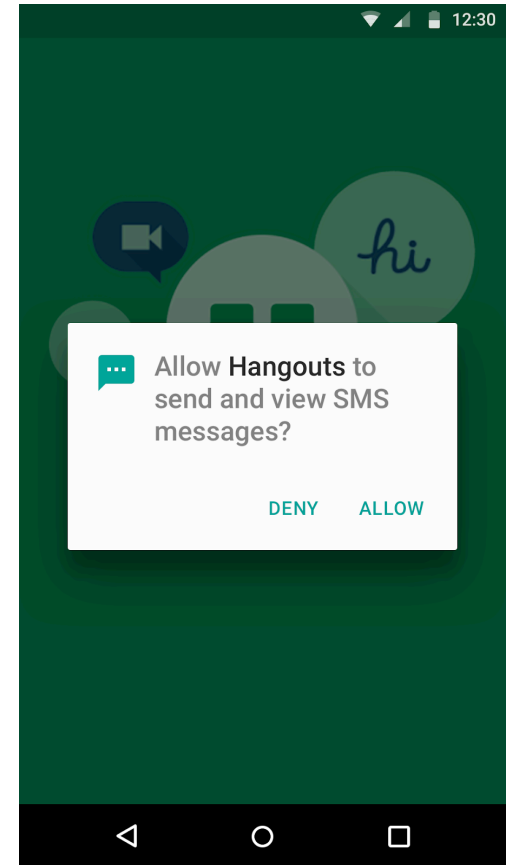
Linux Kernel Services

- Security management
 - Based on Security-Enhanced Linux
 - Only the kernel and a few core applications run as a root
 - Each app is assigned a unique UID
 - Each app runs in its own **sandbox**
 - Private memory for the app
 - Apps cannot access each other's data
 - Android also ensures that the memory is used in a fair manner



Linux Kernel Services

- Security management
 - Each app is given a set of permissions (an app needs to ask the user!)
 - Users can restrict access to system features and user data, encrypt files
- Bad programming is the biggest threat
 - Exposing data to other apps, insecure networking, buggy native code, dynamic code loading

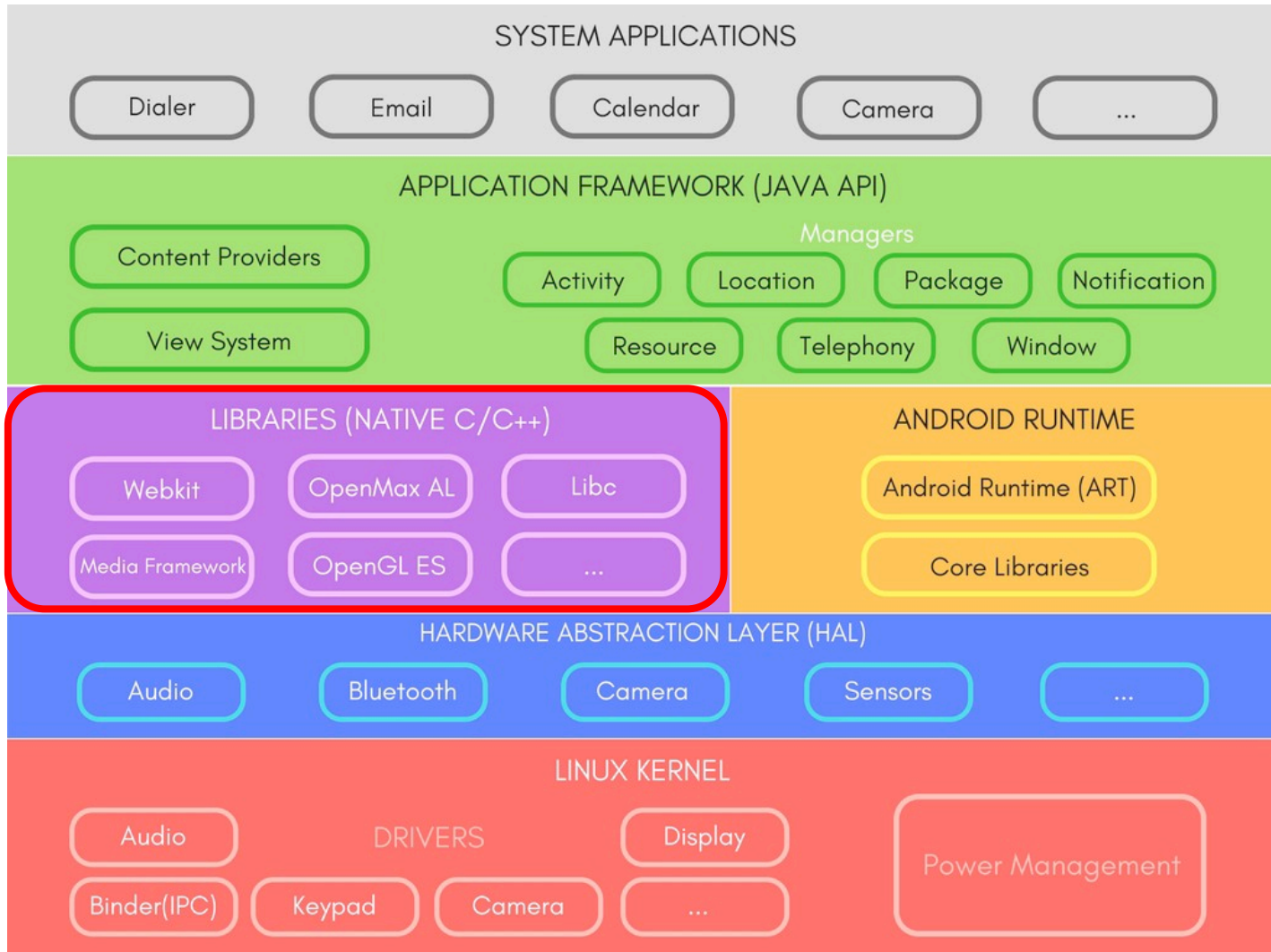


Linux Kernel Services

- Power management
 - Screen dimming, process killing
 - Wakelocks – prevent the device from going to “sleep”
 - Can have a big negative impact on battery life
 - Project Volta – from Lollipop onwards, the OS takes care of scheduling periodic jobs for the apps
 - JobScheduler API: **you might tell your app to do something every 15 minutes, but the OS might schedule this differently!**
- File and network I/O
- Device drivers



Android Architecture

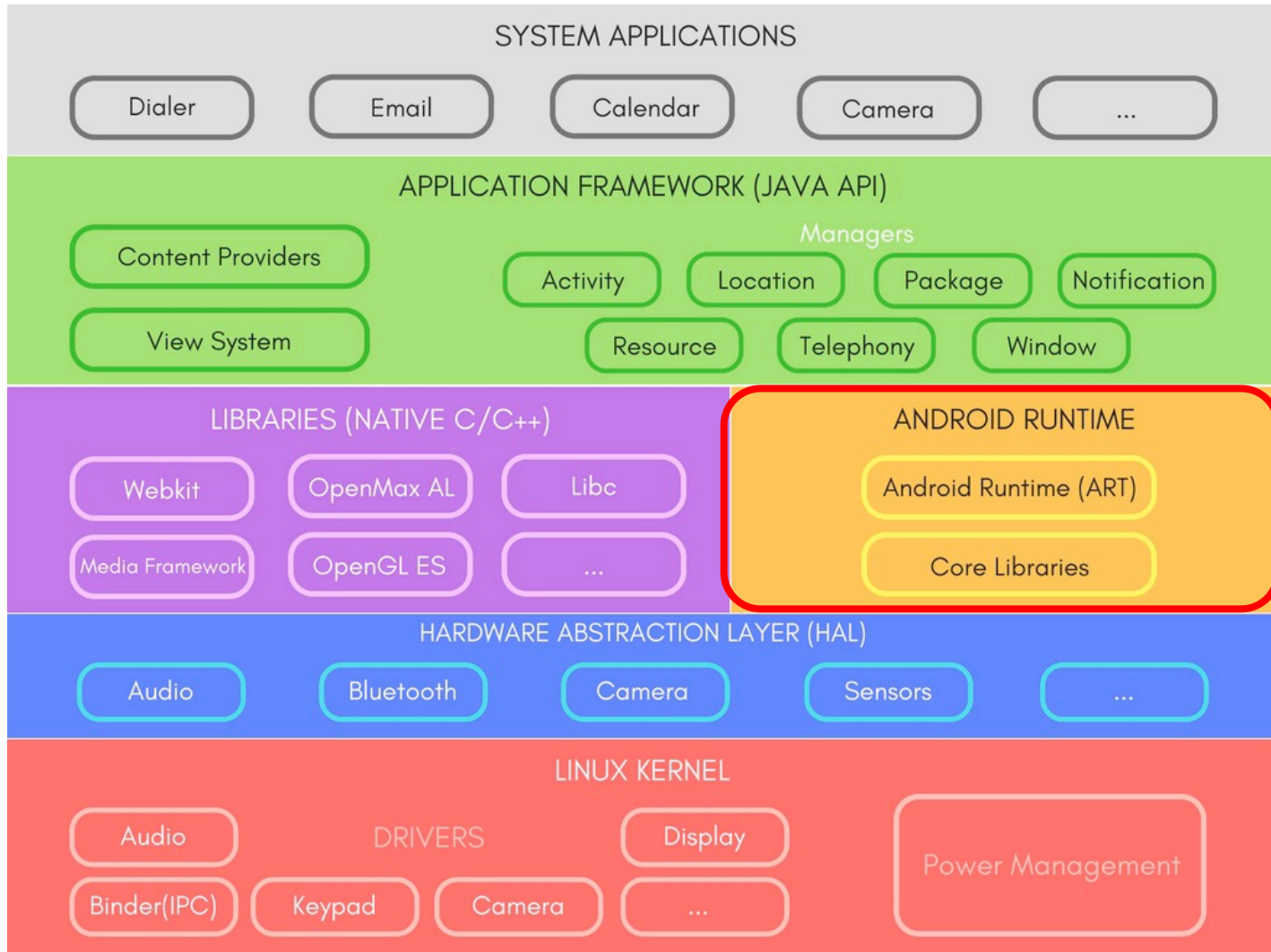


Native Libraries

- System C library
 - Bionic libc
- Surface manager
 - Composing windows on the screen
- Open GL
 - 3D graphics
- OpenMAX
 - Audio, video, and image processing
- Media Framework
 - Recording and playback of audio/video/photos
- SQLite
 - Relational database engine
- Webkit
 - Browser engine
- Neural Networks API



Android Architecture



Android Runtime

- Android core libraries
 - Besides standard Java libraries for tasks such as file handling, Strings, etc., Android includes specific libraries for the mobile environment
 - basic java classes - java.*, javax.*
 - app lifecycle, db management - android.*
 - Internet/Web services - org.*
 - Unit testing - junit.*
- Process virtual machine (VM):
 - Dalvik (until Android 4.4 KitKat)
 - Android Runtime – ART (starting with 5.0 Lollipop)

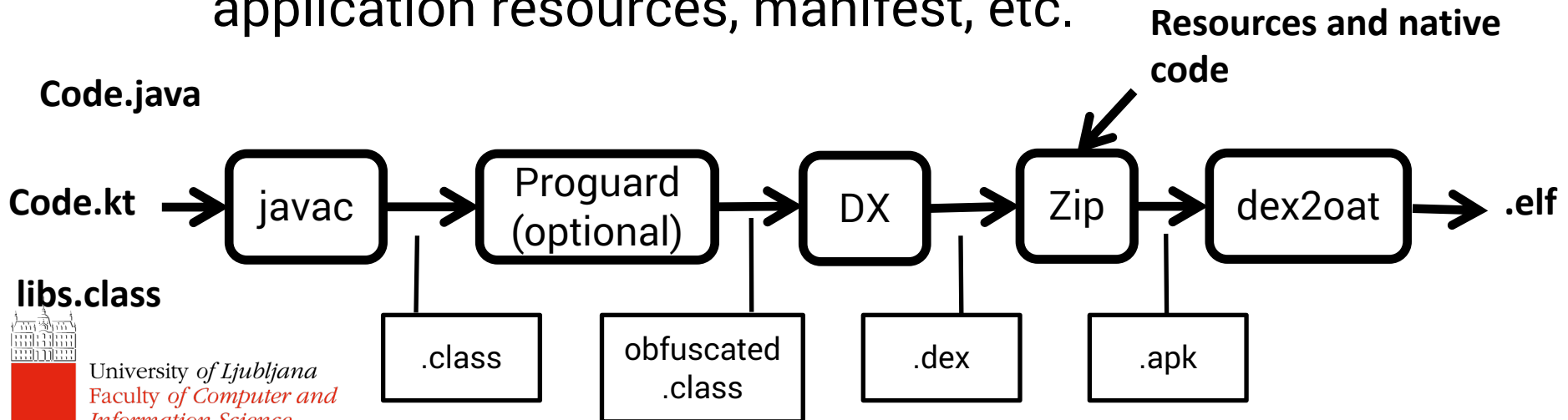


Mostly wrap native libraries



Android Runtime

- Compilation and workflow (with ART)
 - App written in Java or Kotlin
 - Compiled to Java bytecode files (i.e. .class files)
 - DX converts Java bytecode files to a single DEX bytecode file (.dex file) optimised for space
 - .apk file is generated with the dex file and all the application resources, manifest, etc.



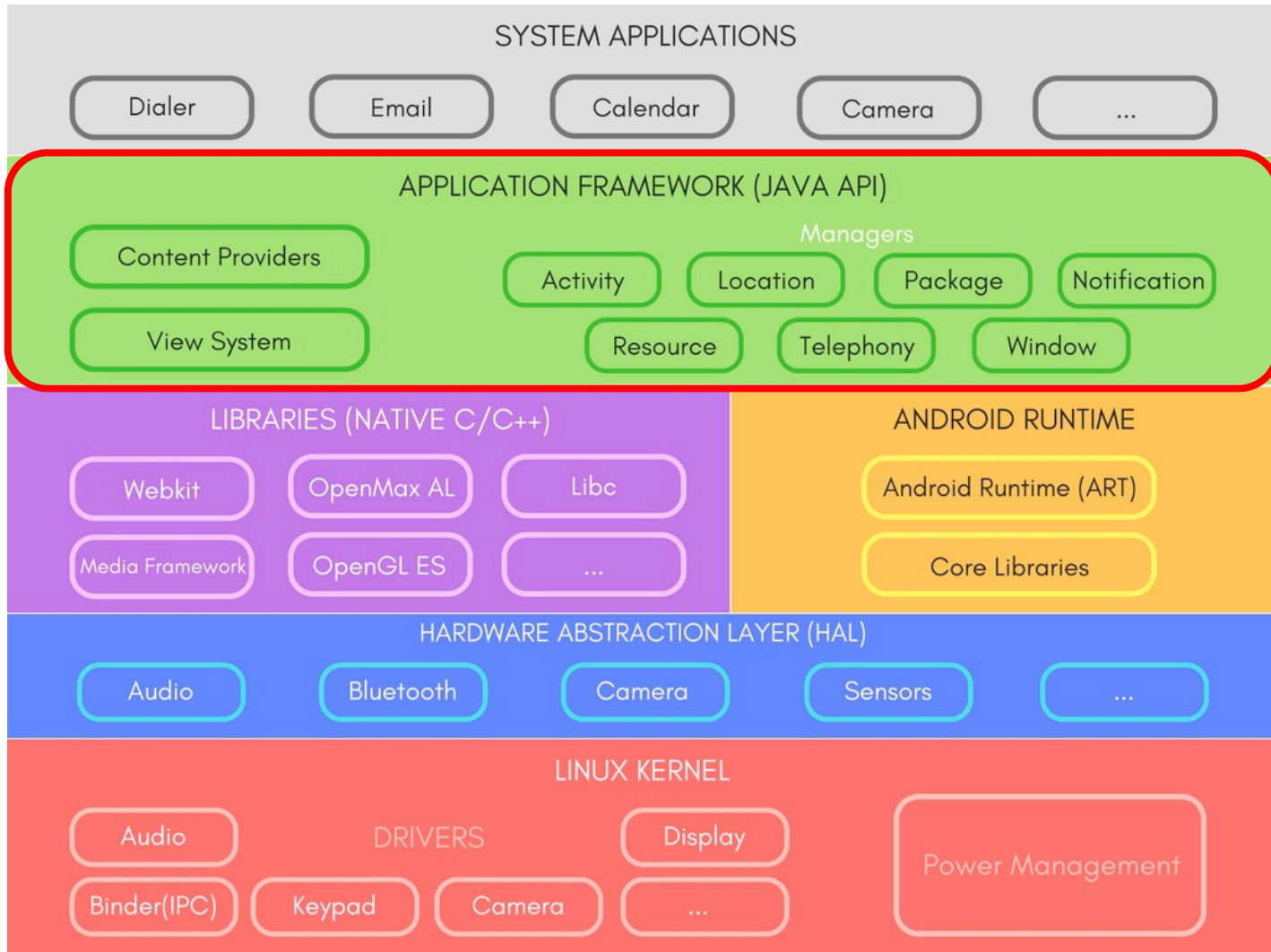
Android Runtime

- **Compilation and workflow (with ART)**
 - .apk is the file you download and install on a device
 - ART:
 - When the.apk is **installed**, ART uses ahead-of-time (**AOT**) compilation to **convert** it and save it as **native machine code**.
 - **Every other time**, the app runs from the **native code**
 - Dalvik:
 - **Trace-based just-in-time (JIT) compilation**: continuously profile apps each time they run and dynamically compile frequently executed short segments of the bytecode into native machine code

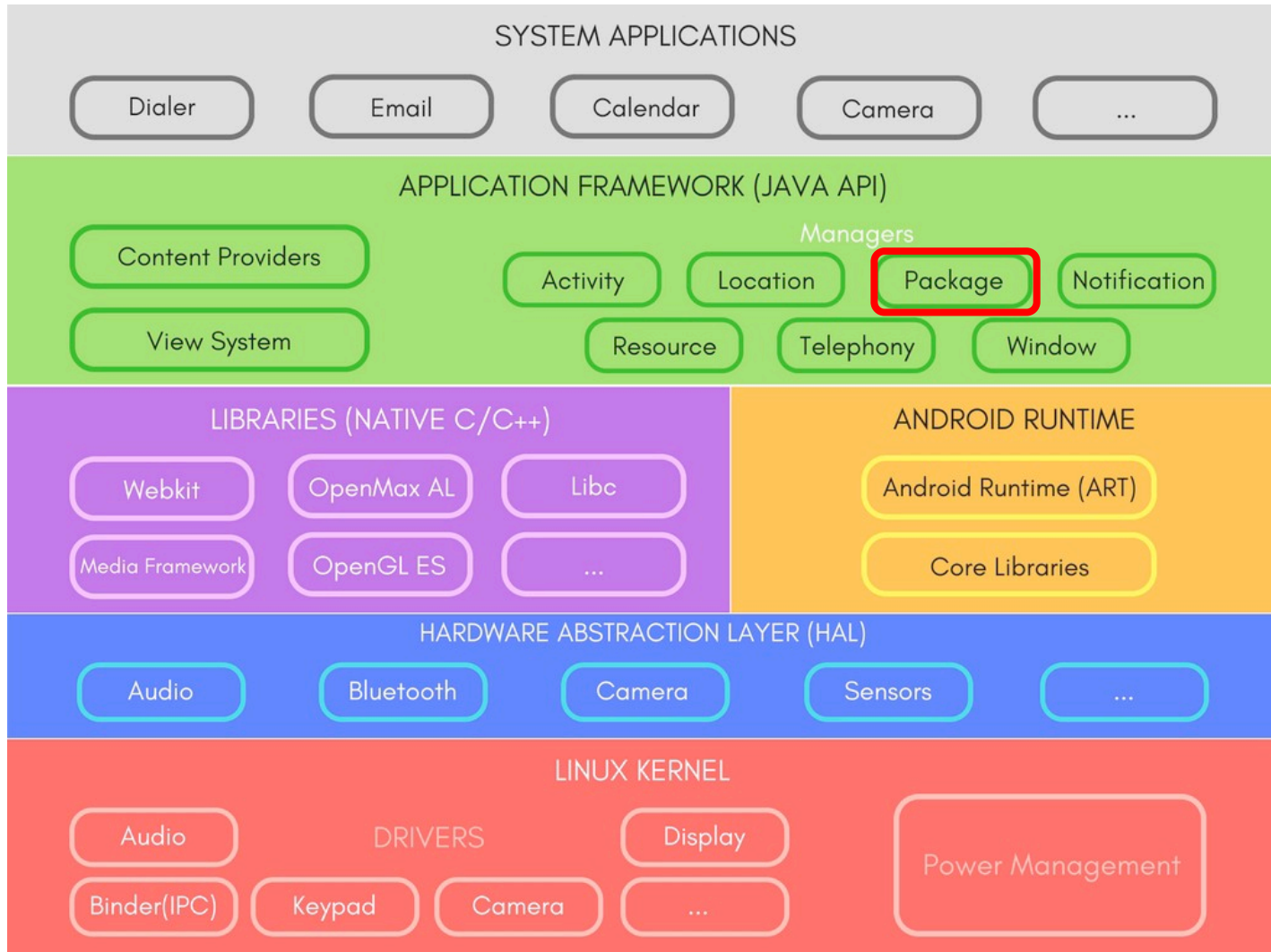
Why did Google engineers change this?



Android Architecture



Android Architecture



Android Architecture

- Package Manager
 - Keeps track of the installed applications
 - E.g. camera invocation code:

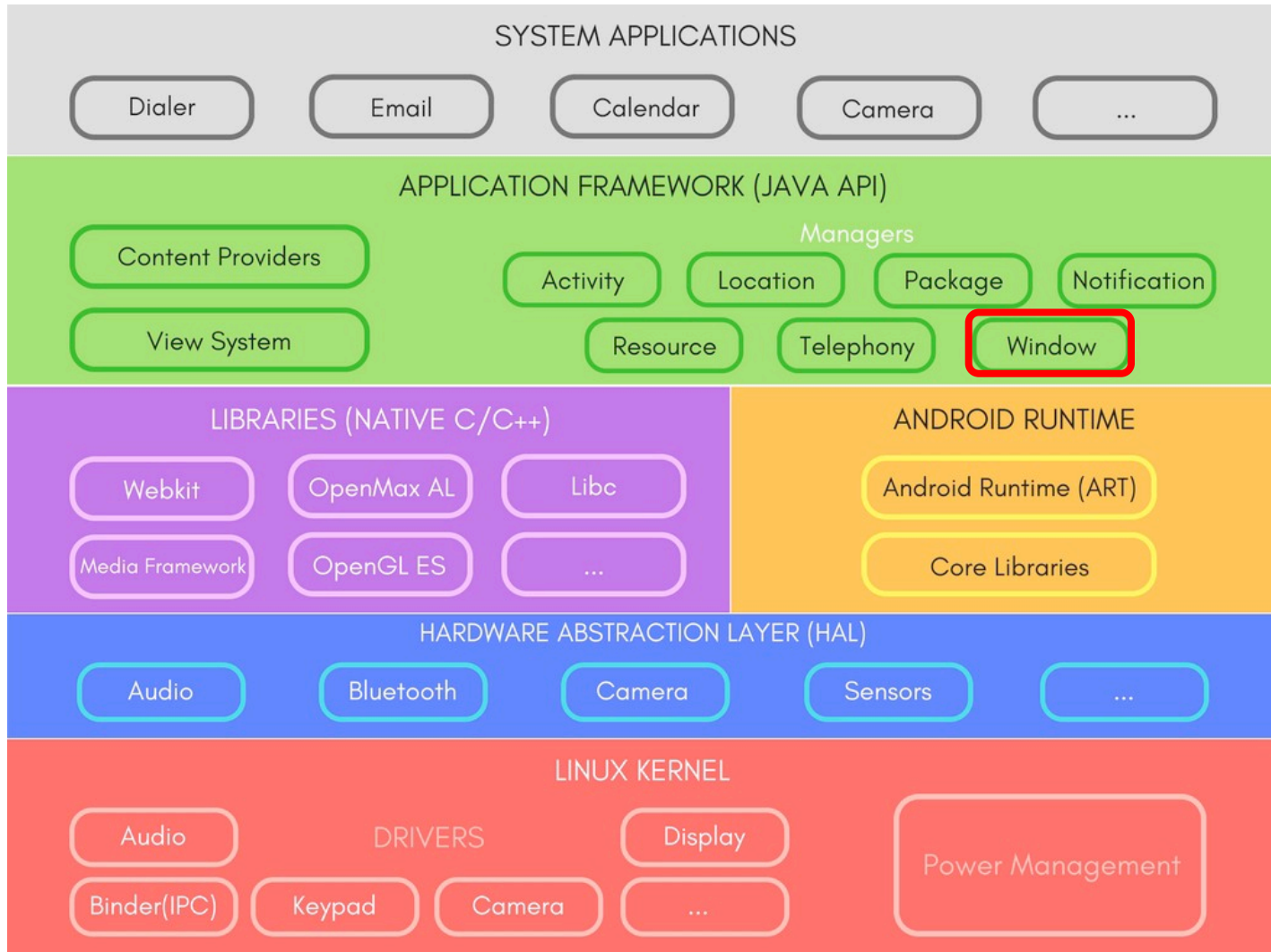
```
Intent takePictureIntent = new  
Intent(MediaStore.ACTION_IMAGE_CAPTURE);
```

```
takePictureIntent.resolveActivity(getPackageManager());
```

- Some other uses:
 - `getInstalledApplications()`
 - `queryContentProviders()`



Android Architecture



Android Architecture

- Window Manager
 - Manages application's windows
 - Ensures that `setContentView` connects the given View with the activity's Window
 - Ensures that your activity's window spans full screen
 - Example usage: floating icon over any app, e.g. Facebook chat heads (bubbles)



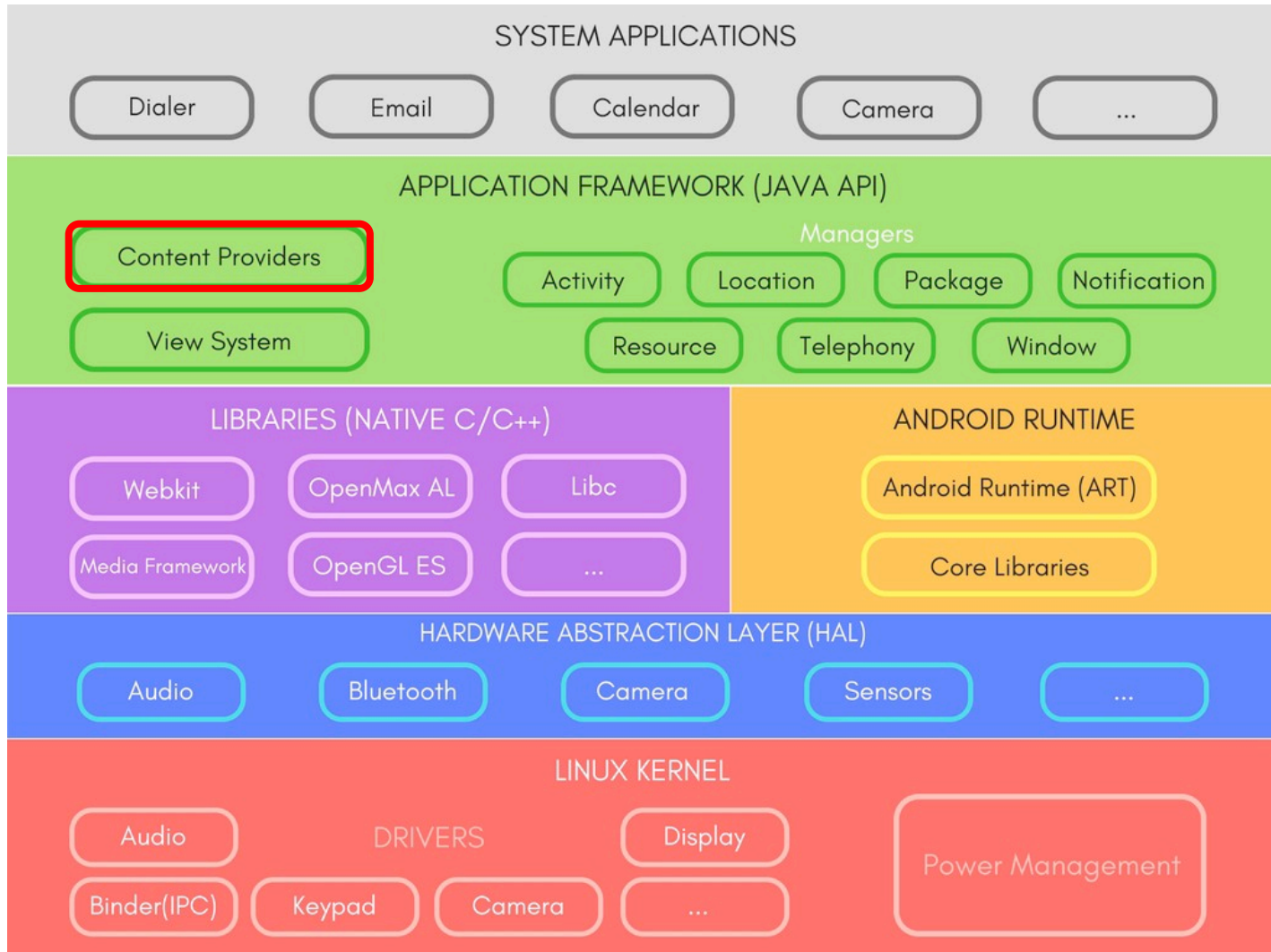
Android Architecture

```
WindowManager.LayoutParams p = new WindowManager.LayoutParams(  
    // Shrink the window to wrap the content  
    // rather than filling the screen  
    WindowManager.LayoutParams.WRAP_CONTENT,  
    WindowManager.LayoutParams.WRAP_CONTENT,  
    // Display it on top of other application windows,  
    //but only for the current user  
    WindowManager.LayoutParams.TYPE_SYSTEM_ALERT,  
    // Don't let it grab the input focus  
    WindowManager.LayoutParams.FLAG_NOT_FOCUSABLE,  
    // Make the underlying application window visible  
    // through any transparent parts  
    PixelFormat.TRANSLUCENT);  
    // Define the position of the window within the screen  
p.gravity = Gravity.TOP | Gravity.RIGHT;  
p.x = 0; p.y = 100;
```

```
WindowManager windowManager =  
    (WindowManager) getSystemService(WINDOW_SERVICE);  
windowManager.addView(myView, params);
```



Android Architecture

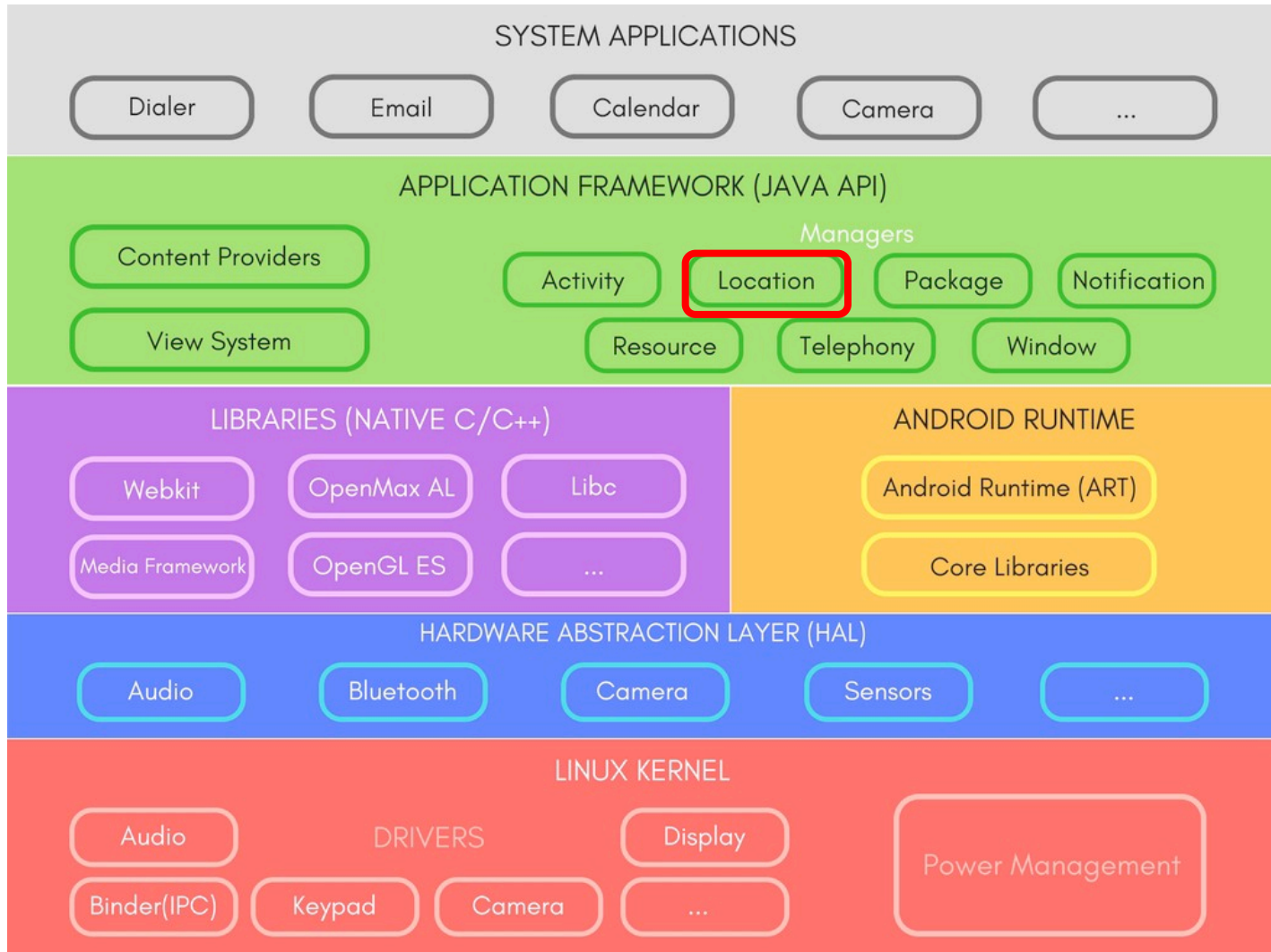


Android Architecture

- Content Providers
 - Applications are sandboxed – you cannot access another app's data, unless explicitly shared
 - Content providers manage access to data that is exposed for inter-application sharing
 - Write your own content providers to:
 - Share data
 - Provide an extra layer of abstraction
 - Example: using existing CP to get contacts for your chat app
 - Example: write a CP to expose a To-Do list created in your app to other apps on the device

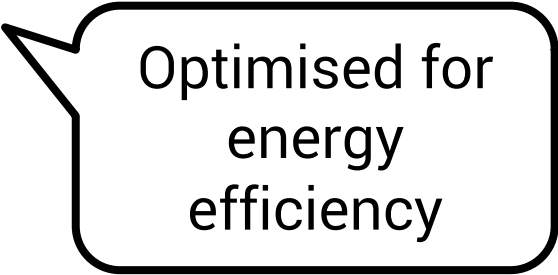


Android Architecture



Android Architecture

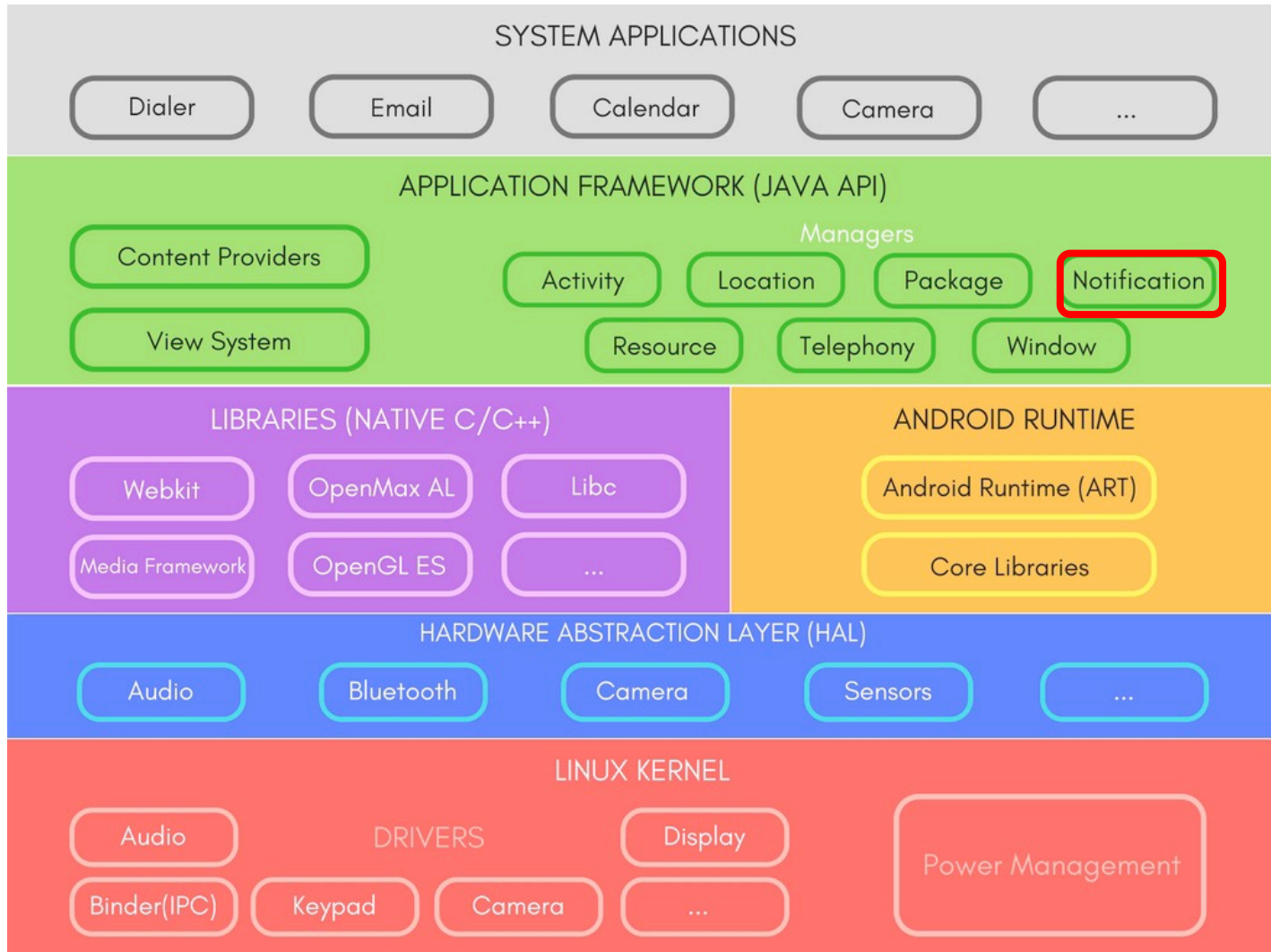
- Location Manager
 - Provides location and movement information
 - Example: obtain periodic updates of the device's geographical location, or to fire an application-specified `Intent` when the device enters the proximity of a given geographical location.
 - Ways to access location info:
 - Directly through Location Manager
 - **Google Play Location Services** – a preferred way of accessing location information.



Optimised for
energy
efficiency

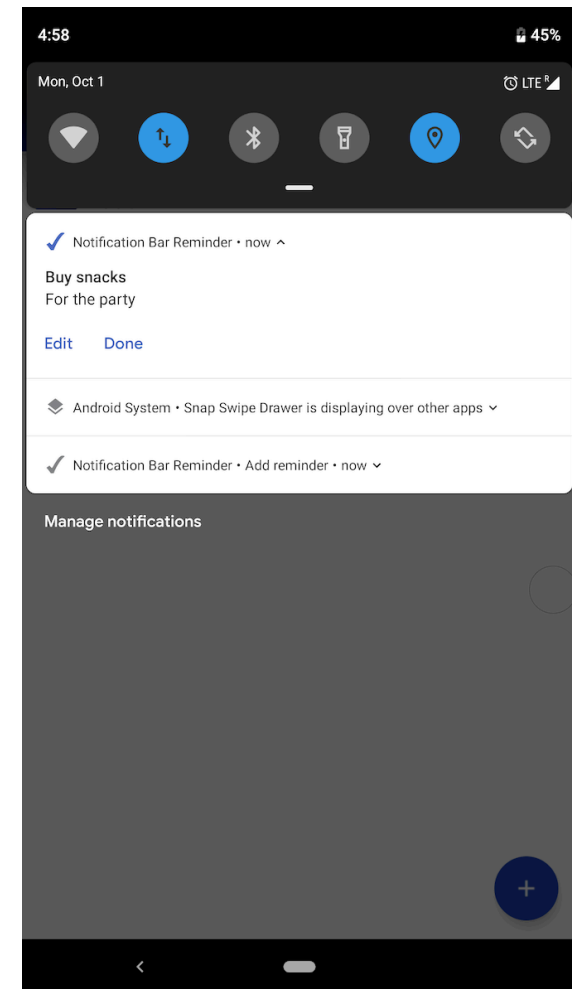


Android Architecture

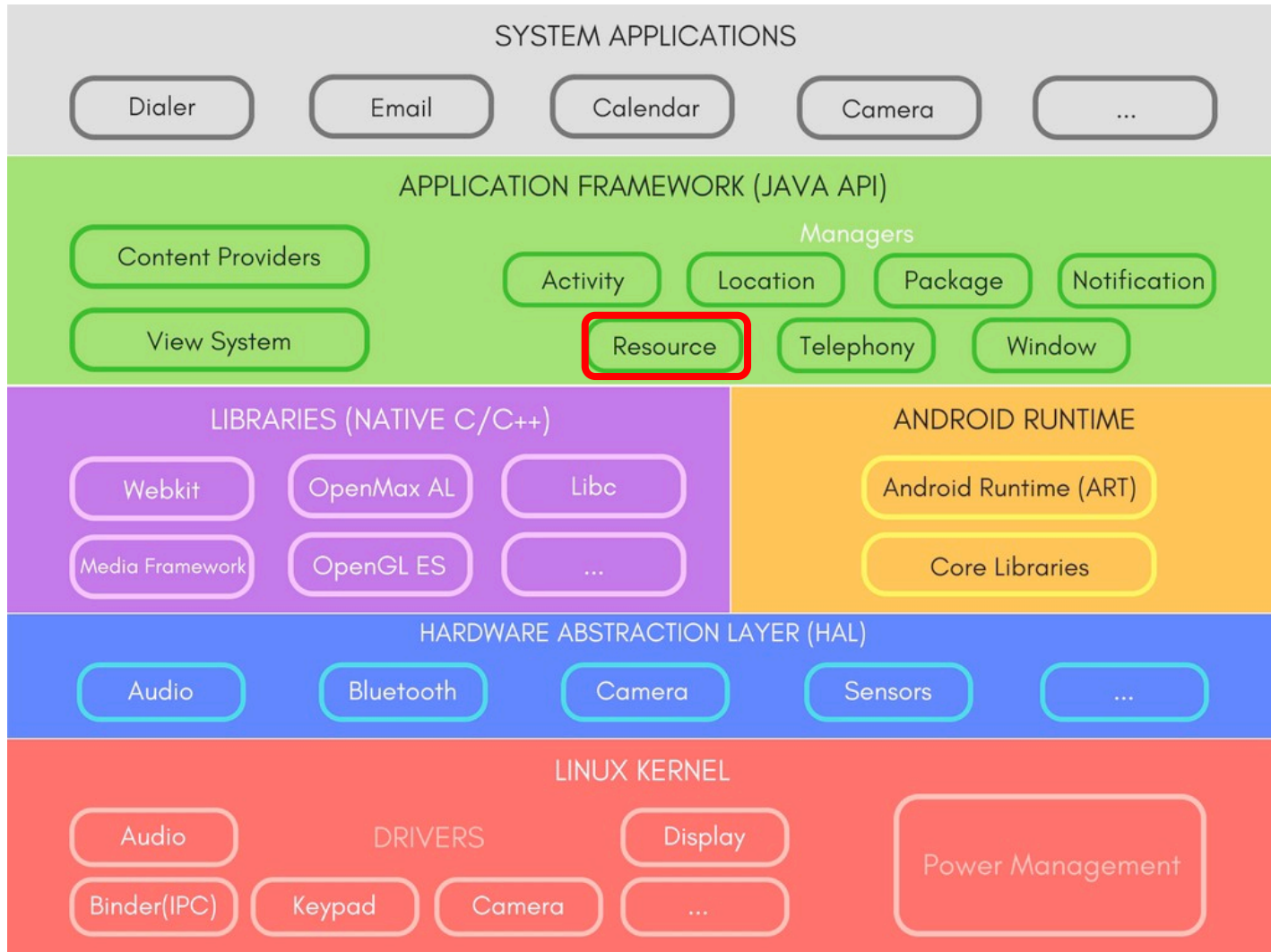


Android Architecture

- Notification Manager
 - Notifications are a means for your application to initiate interaction with a user
 - Show an icon in **the notification bar**
 - Play alert sound, or vibration, flash LED
 - Use carefully: remember the Weiser's vision
 - Notification Manager also allows you to peek into other applications' notifications

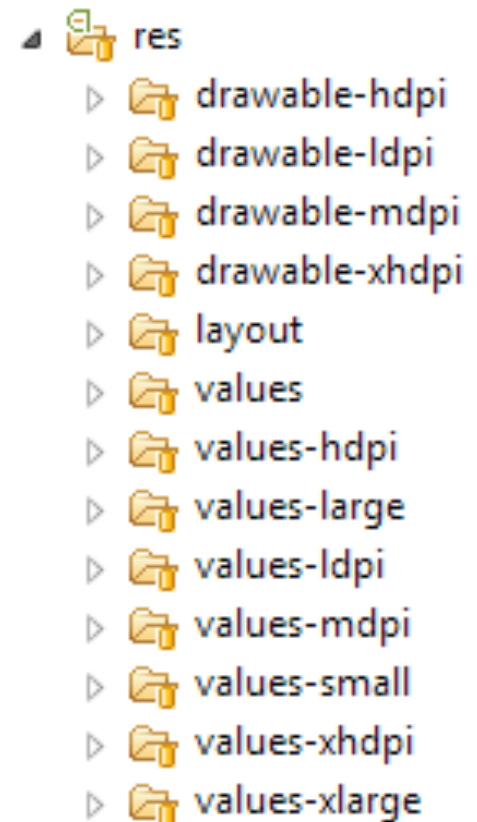


Android Architecture



Android Architecture

- Android application resources:
 - Non-compiled static content of your application
 - See “res” folder created by Android Studio
 - Examples:
 - String values
 - Bitmaps (e.g. backgrounds, icons)
 - Layout files
 - Styles’ definitions



Android Architecture

- Resource Manager
 - Manages these resources
 - Support different screen sizes and orientations
 - Support different languages
 - Support different platform versions
- Resource files are programmatically accessible via the automatically-generated **R** file

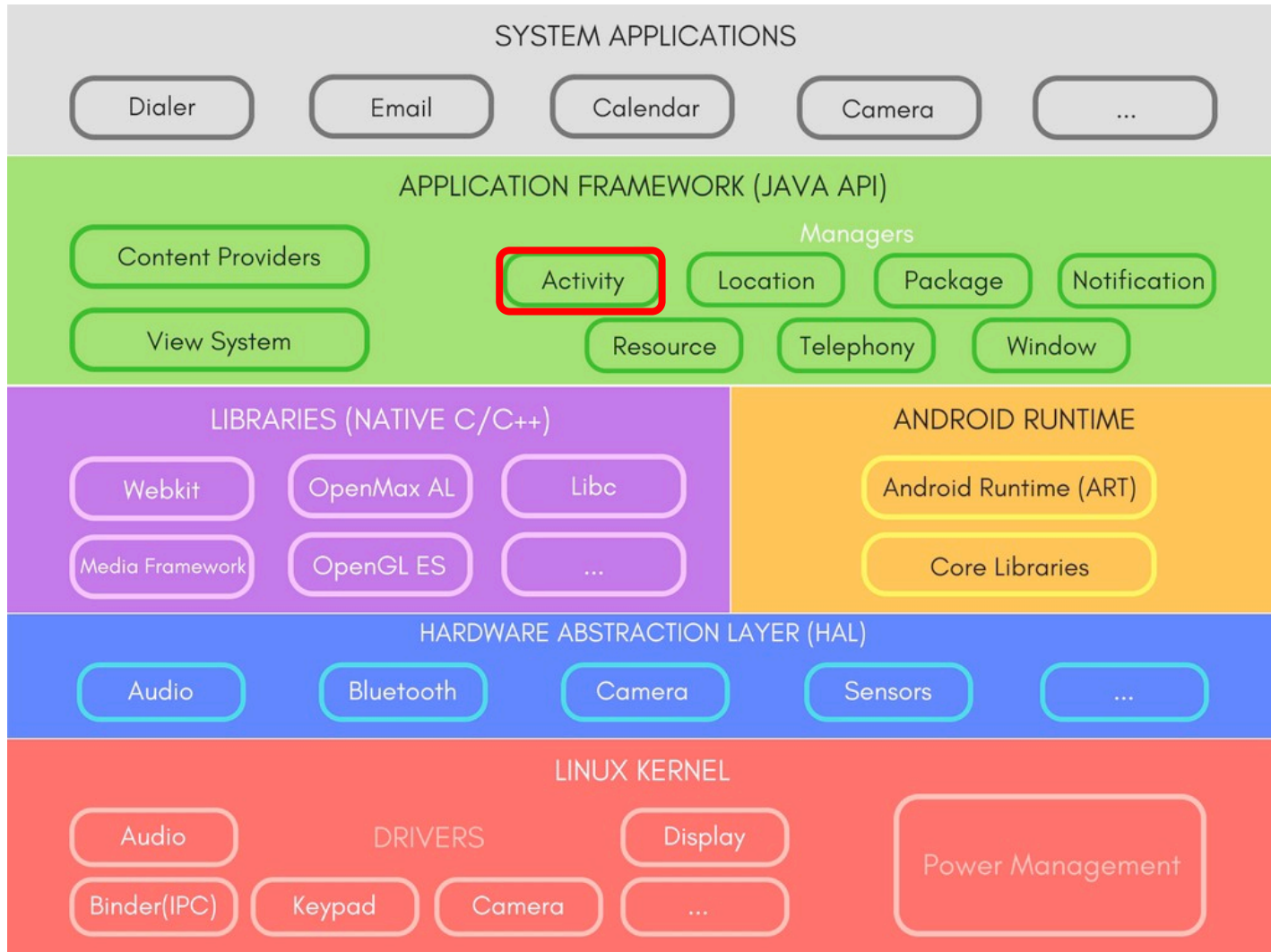
```
String mystring = getResources()  
                .getString(R.string.mystring);
```



Resource ID



Android Architecture

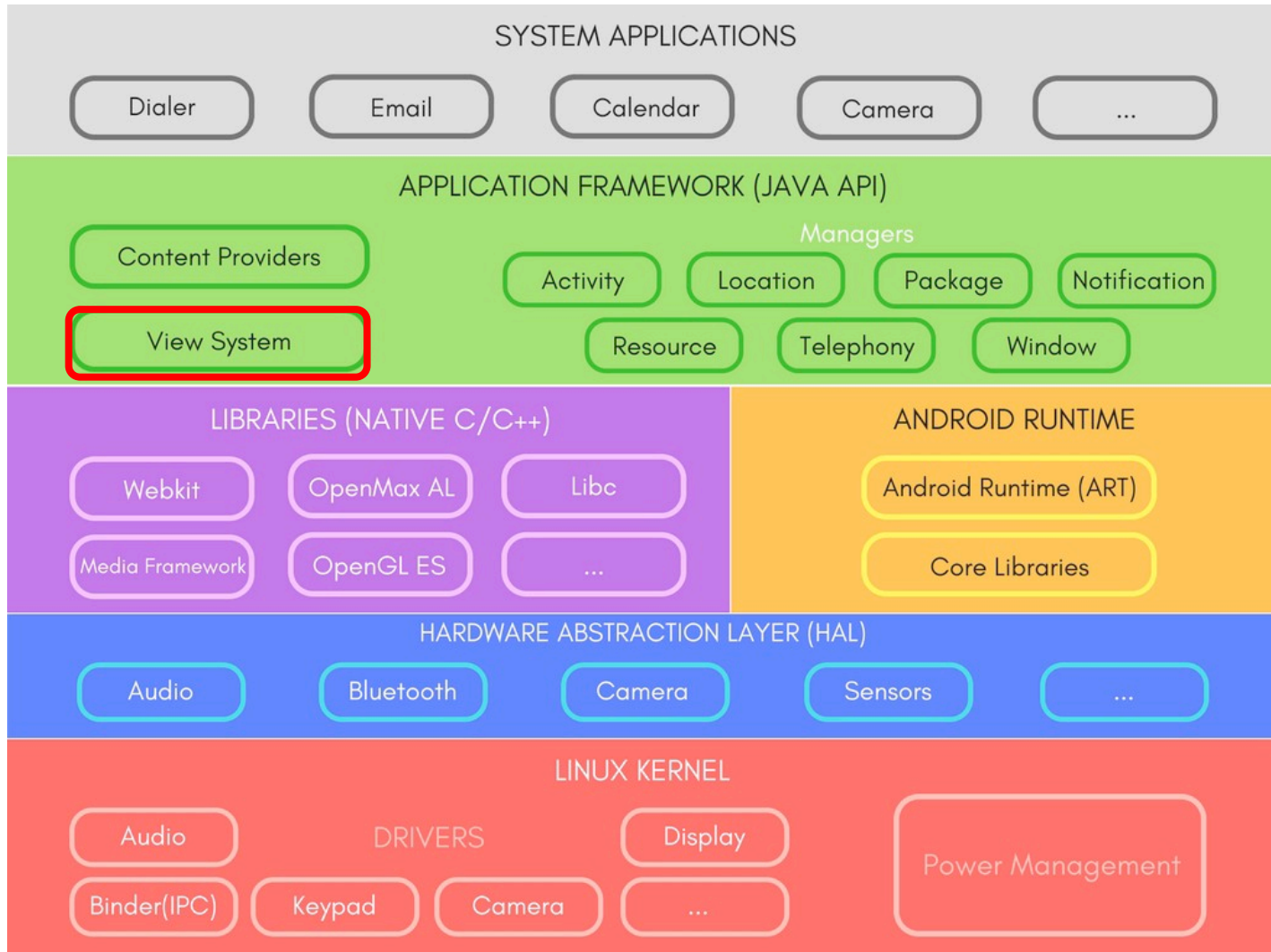


Android Architecture

- Activity Manager
 - Manages the application lifecycle and navigation through the stack of application pages that a user sees
 - Mostly used for debugging purposes or app running configuration adaptation:
 - `clearApplicationUserData()`
 - `isLowRamDevice()`
 - `isUserAMonkey()`

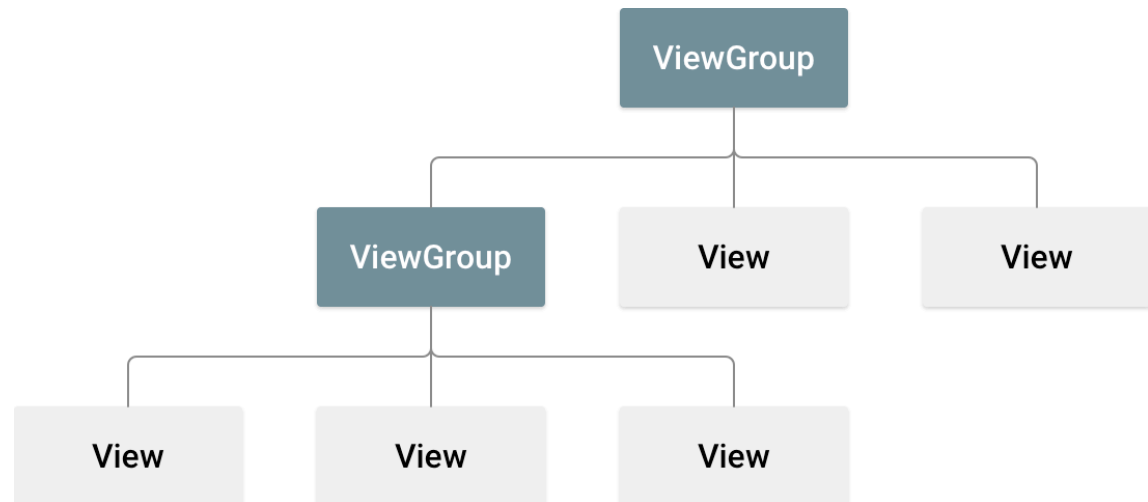


Android Architecture

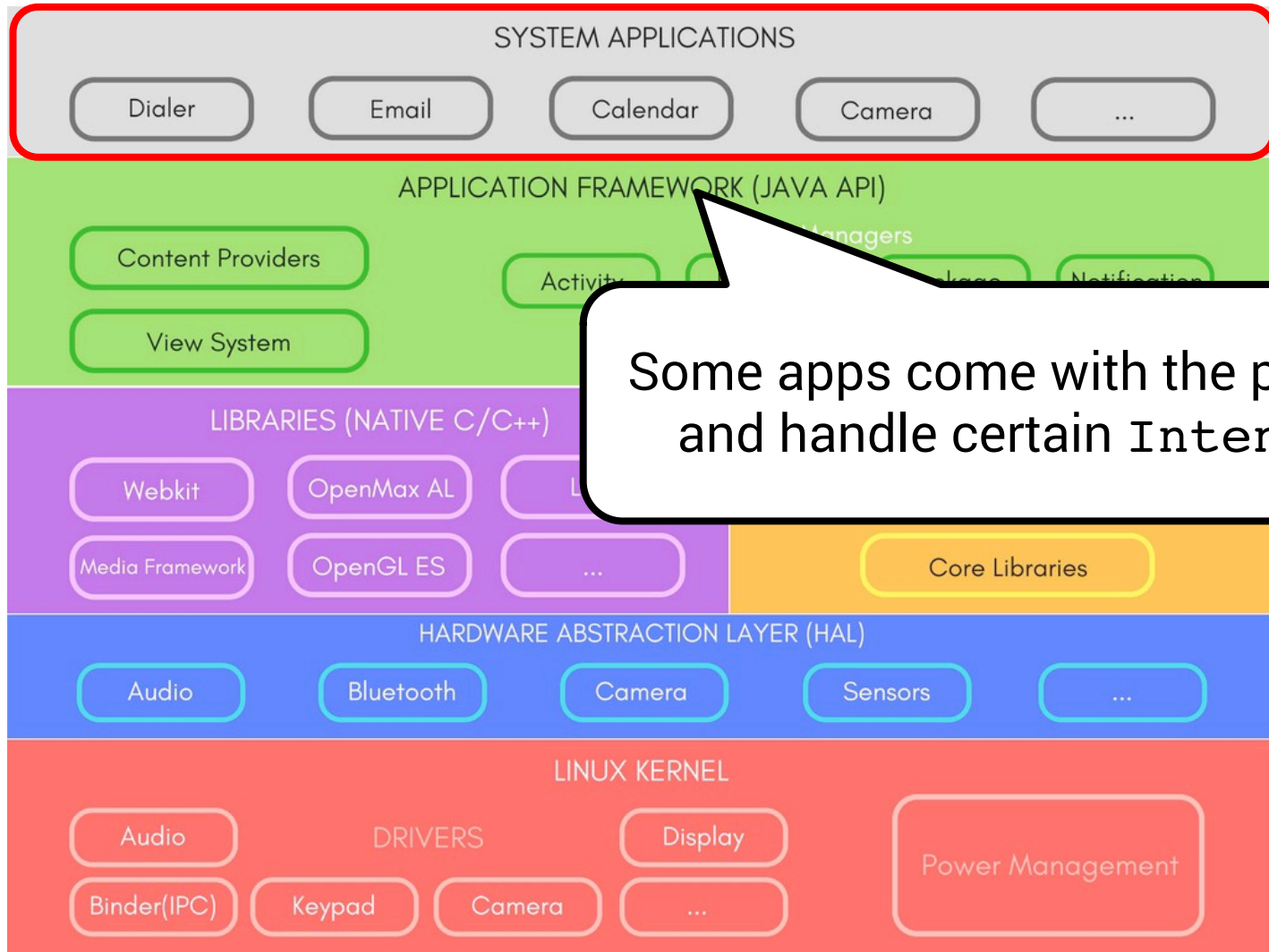


Android Architecture

- View System
 - For building the app's User Interface (UI)
 - UI is represented as a hierarchy of Views
 - Such a structure is called a **layout** and is defined by an XML file



Android Architecture



Some apps come with the phone and handle certain Intents

