Mobile Application Development

Lecture 4: Android App Development Fundamentals

Course Lecturer: Menard Phiri, Dept of CSIT, MUST

Lesson outcomes

- Understanding Android programming languages
- Understanding the platform architecture of Android OS

Android Programming Languages

- Android natively supports three programming languages Java, Kotlin and C++.
- Additionally, Android apps can also be developed using crossplatform frameworks like Flutter, React Native and Xamarin.
- The Android SDK tools compile your code along with any data and resource files into an APK, an Android package.

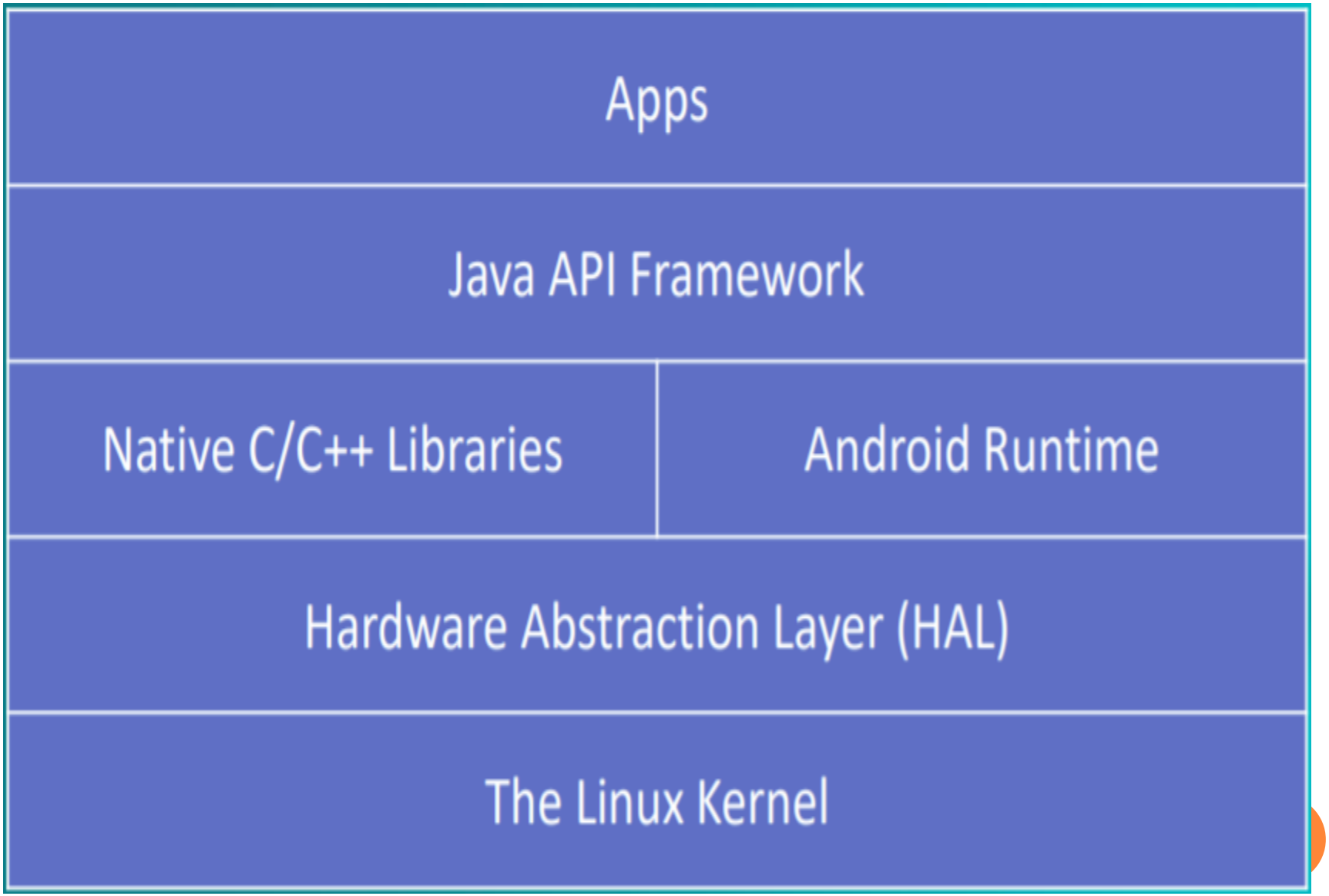
Android Programming Languages

- The Android package (APK) is an archive file with an .apk suffix.
- The APK is the file that Android-powered devices use to install the app.
- Each APK file contains all the contents necessary for running a given Android app.

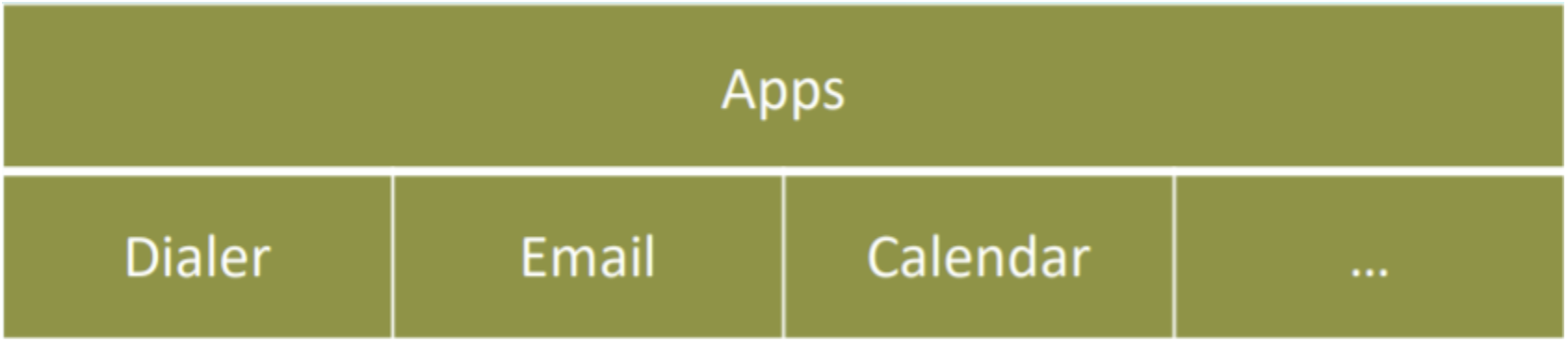
Android Platform Architecture

- The Android operating system follows a layered architecture approach.
- It is a stack of software components which is roughly divided into five main layers with six different sections.
- All these layers are responsible for different roles and have different features.

Android Platform Architecture



Apps

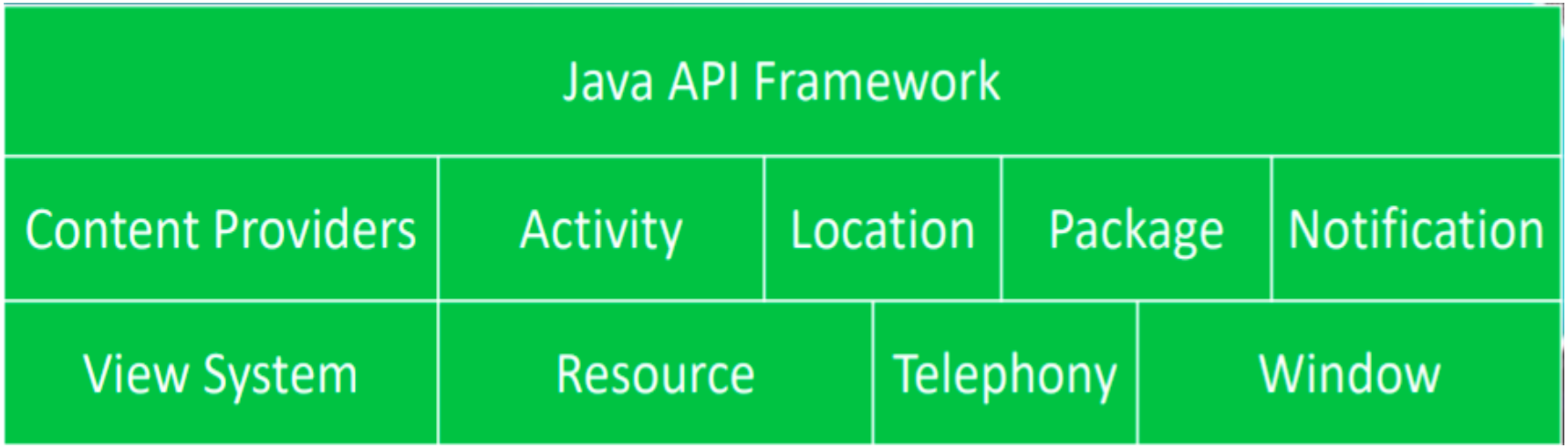


- The top of the Android OS Stack is occupied by apps that users use.
- This layer uses all the layers below it for the proper functioning of these mobile apps.
- Some apps, called system apps, are preinstalled in the device.
- Other apps, called third-party apps, are downloaded from Android supported app stores like Google Play Store.

Apps

- The system apps function both as apps for users and to provide key capabilities that developers can access from their app.
- For instance, if a third-party app would like to deliver an SMS message, the developer is not required to build that functionality.
- Instead, developer can just invoke whichever SMS app is already installed to deliver a message to the recipient they specify.

Java API Framework



- It is a collection of APIs written in Java, which gives developers access to the complete feature set of Android OS.
- Developers have full access to the same framework APIs used by the core applications to enhance the functionalities of their applications.
- These APIs form the building blocks you need to create Android apps by enabling and simplifying the reuse of core components and services.

Java API Framework

1. A View System:

– A rich and extensible system is used to build an app's UIs.

2. A Resource Manager:

– A service that provides access to non-code resources such as localized strings, graphics, and layout files.

3. Package Manager:

– A service that manages installation of apps on the device.

Java API Framework

4. Notification Manager:

– A service that enables all apps to display custom alerts in the status bar.

5. An Activity Manager:

– A service that manages the lifecycle of apps and provides a common navigation stack.

6. Content Providers:

– A group of resources that enable apps to access or share data within Android.

6. Telephony Manager:

– A service that enables app to use phone capabilities of the device.

3. Native C/C++ Libraries and Android Runtime

Native C/C++ Libraries			Android Runtime
Webkit	OpenGL	Libc	Android Runtime (ART)
Media Framework	SSL	SQLite	Core Libraries

- All Android system components and services are built from native libraries written in C and C++.
- This layer contains those C/C++ libraries used by the various components of the system.

Native C/C++ Libraries

- These libraries are exposed to the developer through Java framework APIs in the upper layer of the system stack.
- For those developers using C/C++ to develop apps, they have to use the Android NDK to access these native platform libraries directly.
- Examples of features include SQLite, WebKit, the surface manager library, the media framework library, and the Open Graphics Library (OpenGL).

Android Runtime

- This section provides a Java virtual machine that is specially designed and optimized for running Android apps.
- The virtual machine enables every Android app to run in its own process with its own instance of the virtual machine.
- The virtual machine runs the apps by executing bytecode files called Dalvik Executable format (DEX) files.

Android Runtime

- Prior to Android version 5.0 (API level 21), Android was using Dalvik as the Android runtime.
- Since Android version 5.0 (API level 21), Android uses Android Runtime (ART) virtual machine.
- Due to these changes, an app that runs well on ART should work on Dalvik as well, but the reverse may not be true.
- On Android 9 (API level 28) and higher, conversion of an app package's DEX files to more compact machine code.

Android Runtime

- The Android Runtime virtual machine is responsible for:

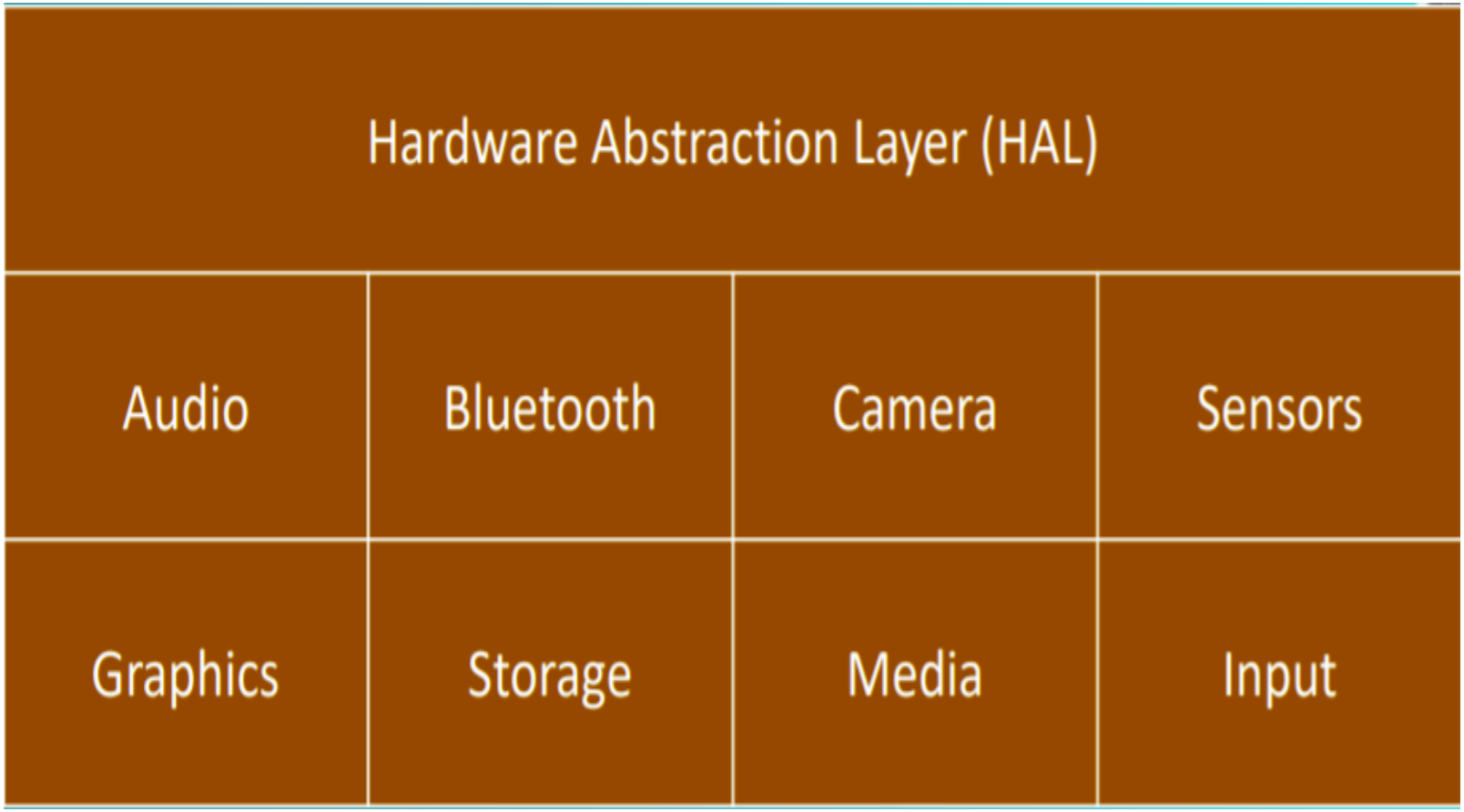
– Providing a set of core libraries for developing Android apps.

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

– Optimized garbage collection (GC).

– Provision of debugging support tools.

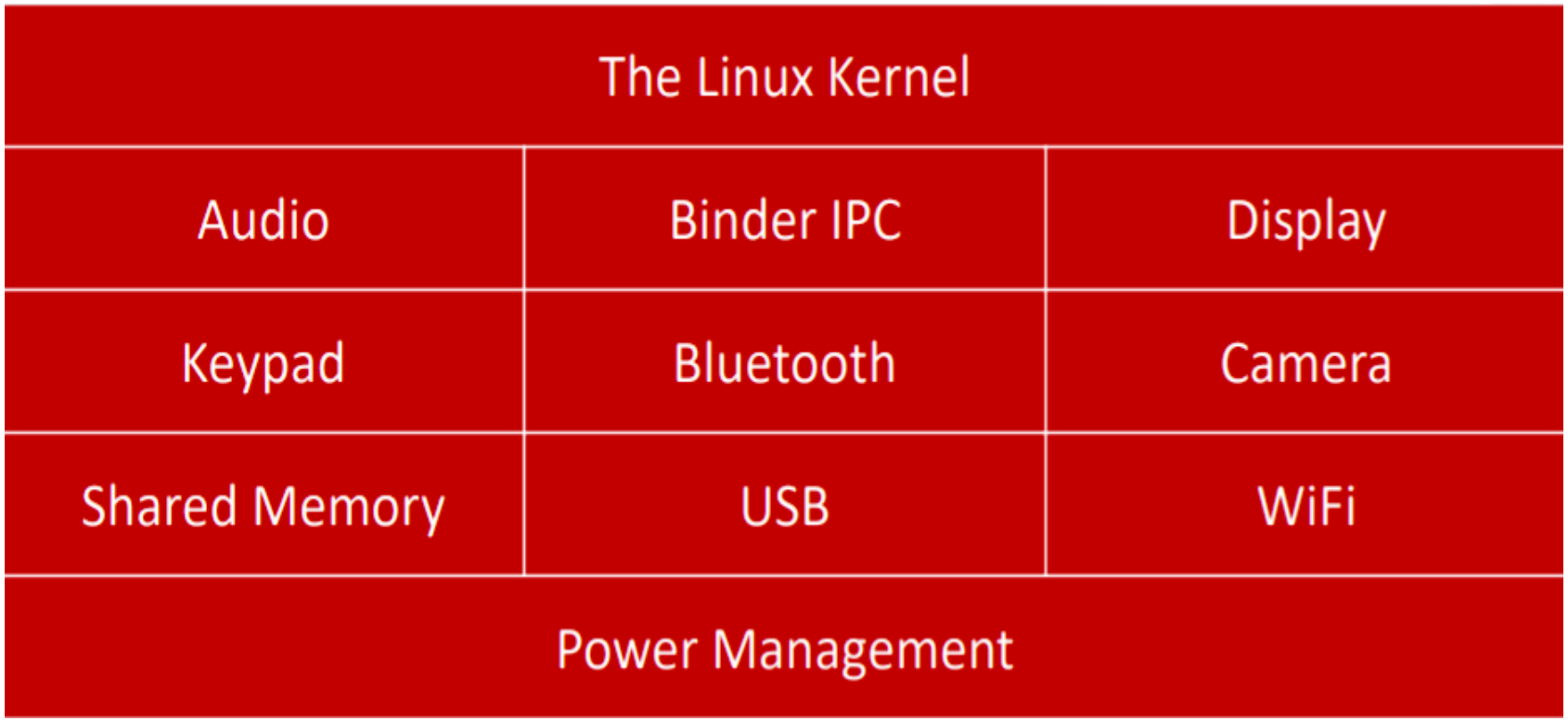
Hardware Abstraction Layer (HAL)



Hardware Abstraction Layer (HAL)

- Hardware Abstraction Layer (HAL) is a logical layer of the Android OS that serves as an abstraction layer between the device's physical hardware and the software.
- It contains hardware specific implementations of interfaces between system services (such as the camera software) and the hardware drivers for that service (such as the camera driver).
- It uses the functions provided by the lower-layer Linux kernel to serve the request from the Android application/framework.
- HAL is a vendor-specific layer implemented in the C/C++ language.
- HAL modules are packaged into shared libraries (.so files) and loaded by the Android system at the appropriate time.

The Linux Kernel



The Linux Kernel

- This layer is the foundation of the Android Platform.
- The Linux kernel acts as an abstraction layer between the software and hardware present on the device.
- It contains all low level drivers for that control the underlying various hardware components support.
- It is responsible for managing the core functionality of Android.

The Linux Kernel

- Android Runtime relies on Linux Kernel for core system services, such as:

–Process Management

–Memory Management

–Security Management

–Networking

–Threading

–Hardware Drivers