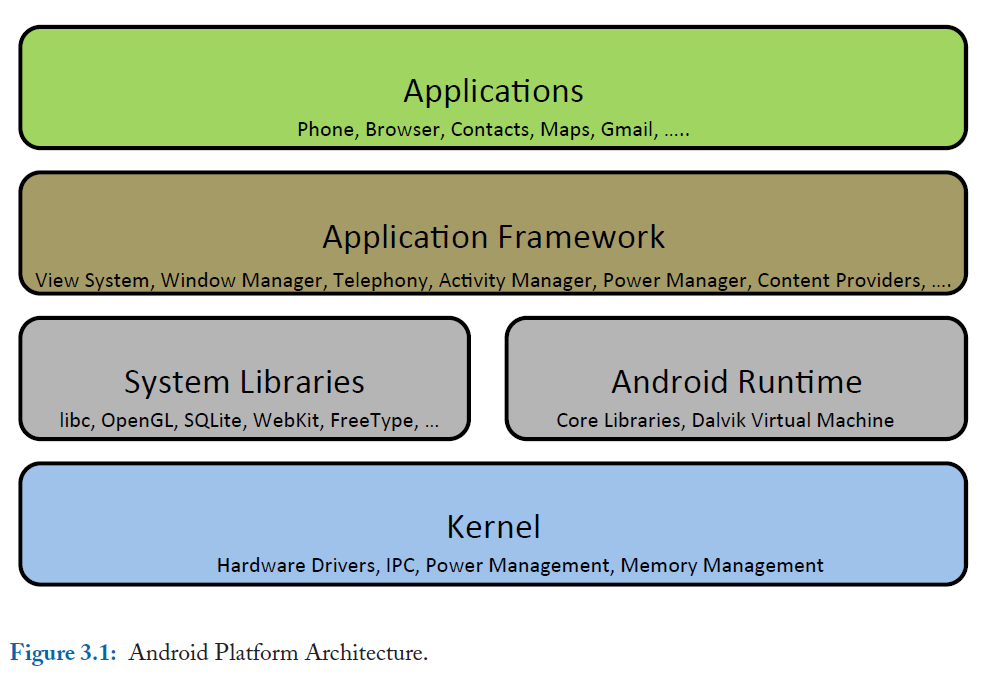
**Android platform architecture**



1. Kernel:

* Android uses a customized Linux kernel based on version 2.6.
* The kernel provides hardware abstraction, drivers, and lowlevel functionalities.
* It introduces WakeLocks to prevent devices from going into low power states, enhancing user experience.
* Android utilizes the Binder mechanism for interprocess communication and memory management.

2. Android Runtime:

* Android uses the Dalvik Virtual Machine (VM) for executing applications.
* Applications are written in a dialect of Java, compiled into .dex files, and run on the Dalvik VM.
* Each application runs in its own instance of the Dalvik VM for security and isolation.
* Android apps can request permissions to access device data during installation.

3. System Libraries:

* Android provides C/C++ libraries accessible through the Application Framework.
* Libraries include libc, FreeType, SQLite, OpenGL/ES, Scalable Graphics Library (SGL), and more.
* LibWebCore powers the Android browser engine, important for web applications.
* The Android Media Library handles multimedia content, including audio and video.

4. Application Framework:

* The Application Framework offers highlevel APIs for developers to use Android's capabilities.
* The View System is used to create user interfaces, with views, widgets, and view groups.
* ContentProvider enables data sharing between applications.
* NotificationManager manages notifications for events.
* Resources are managed separately for customization and efficiency.
* LocationManager handles locationbased services and GPS.
* InputMethodService allows developers to create custom input methods.
* TelephonyManager provides telephony service information.
* Utility classes simplify tasks like phone number formatting.
* PowerManager controls device power states and includes WakeLocks for responsiveness.

**Android Anatomy**

An Android application consists of four types of components:

1. Activity:

* Represents a screen with a visual user interface.
* Defined by the Activity class.
* An app can have multiple activity components for different screens.
* Activities can be started in other applications asynchronously through Intents.

2. Service:

* Used for background tasks and functionalities that don't require direct user interaction.
* Defined by the Service class.
* Does not have a user interface.
* Can be started based on requests from other applications.

3. Content Provider:

* Enables data sharing between applications.
* Defined by the ContentProvider class.
* Accessed using the ContentResolver interface.
* Other apps can query and, if allowed, modify data provided by a content provider.
* Interaction with a content provider is not initiated through Intent objects.

4. Broadcast Receiver:

* Responds to system-wide broadcast announcements.
* Defined by the BroadcastReceiver class.
* Handles system status messages and events.
* Represents an Intent object that carries the broadcast message.
* Can create notifications on the device's user interface using the Notification Manager.

All these components must be declared in the AndroidManifest.xml file, including any declared Intent Filters to specify their capabilities. The manifest file also defines required user permissions, hardware and software services, external libraries, and the API Level used by the application. It plays a crucial role in specifying the application's configuration and capabilities.