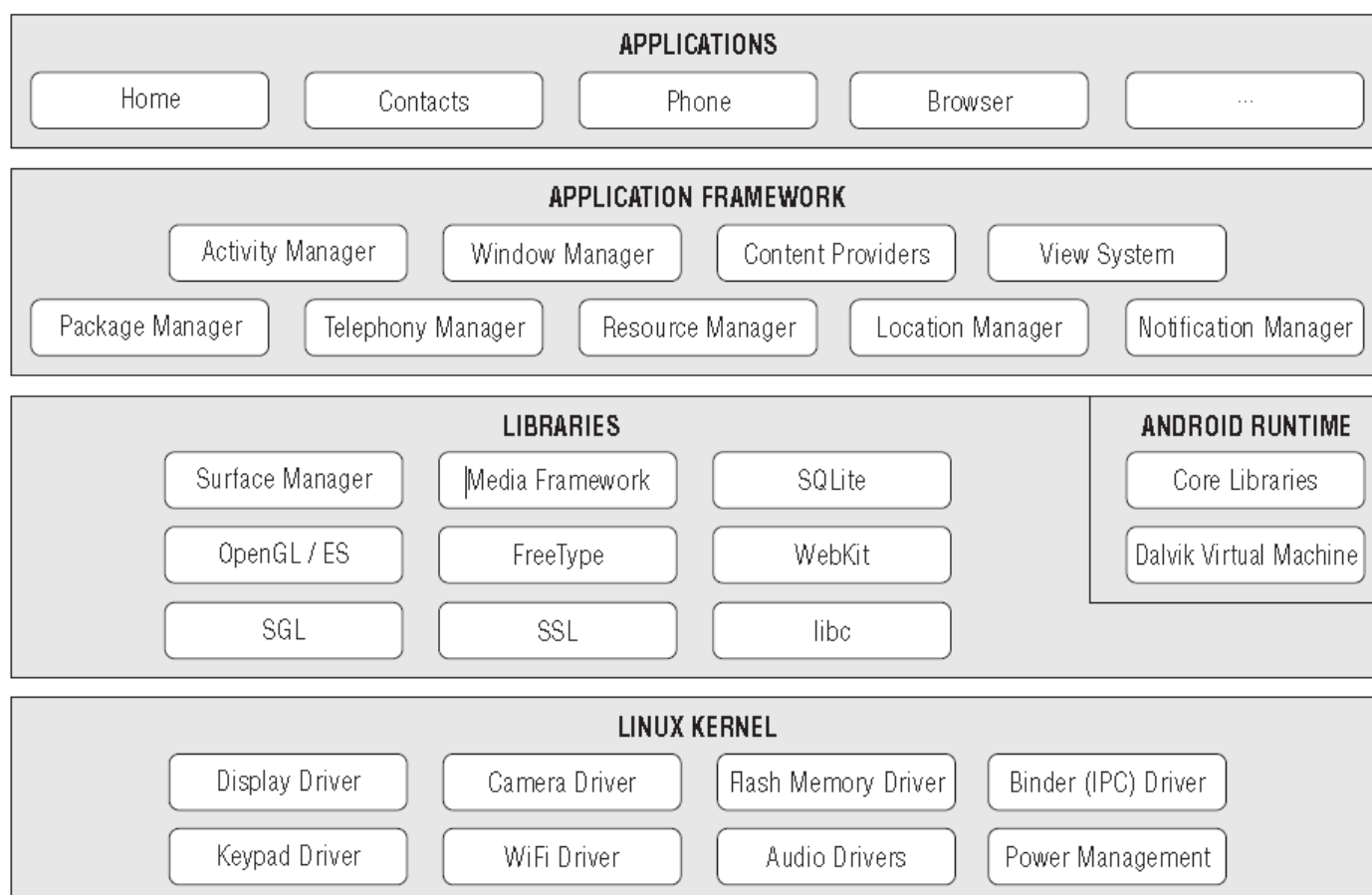


Android: Android is a mobile operating system that is based on a modified version of Linux. Android's Linux operating system kernel provides low level interface with hardware, memory management and process control which is optimized for mobile devices.

Architecture of Android: Android's architecture can be divided into four layers.

1. Linux Kernel
2. Libraries + Android Runtime
3. Application Framework
4. Applications



1. Linux Kernel: Android is based on Linux Kernel layer. It contains all low level device drivers for various hardware components of an Android device like display driver, camera driver, memory driver, keypad driver etc. It also provides some core services like security service, network service, power management and process management for android devices.

2. Libraries: These are basic libraries developed in C/C++ and provide functionality to interact with Android. Some basic libraries are Surface Manager (for display management), SQLite (for database support), SSL and WebKit (for web browsing and internet security) etc.

3. Android Runtime: Android runtime provides a set of core libraries that enable developers to write android applications using Java programming language. The android runtime also includes **Dalvik Virtual Machine**, which is responsible to run android applications.

Dalvik is a specialized virtual machine designed specifically for Android and optimized for battery powered mobile devices with limited memory and cpu. Dalvik uses the device's underlying Linux kernel to handle low level functionality including security, threading, process and memory management.

4. Application Framework: Application framework provides classes for hardware access, managing user interface and application resources.

5. Applications: At this layer all Android factory and third party applications resides.

Types of Android Applications: Android application can be divided into following categories.

- **Foreground:** These applications provide their functionality when they are visible on device's screen. For example games and maps applications.
- **Background:** These applications spend most of their lifetime hidden and have very little interaction from user. For example call responding service, messaging service.
- **Intermittent:** These applications expect some interaction but do most of their work in background. These applications often run silently and notify users when appropriate. Example : media player.
- **Widget:** These are home screen components.

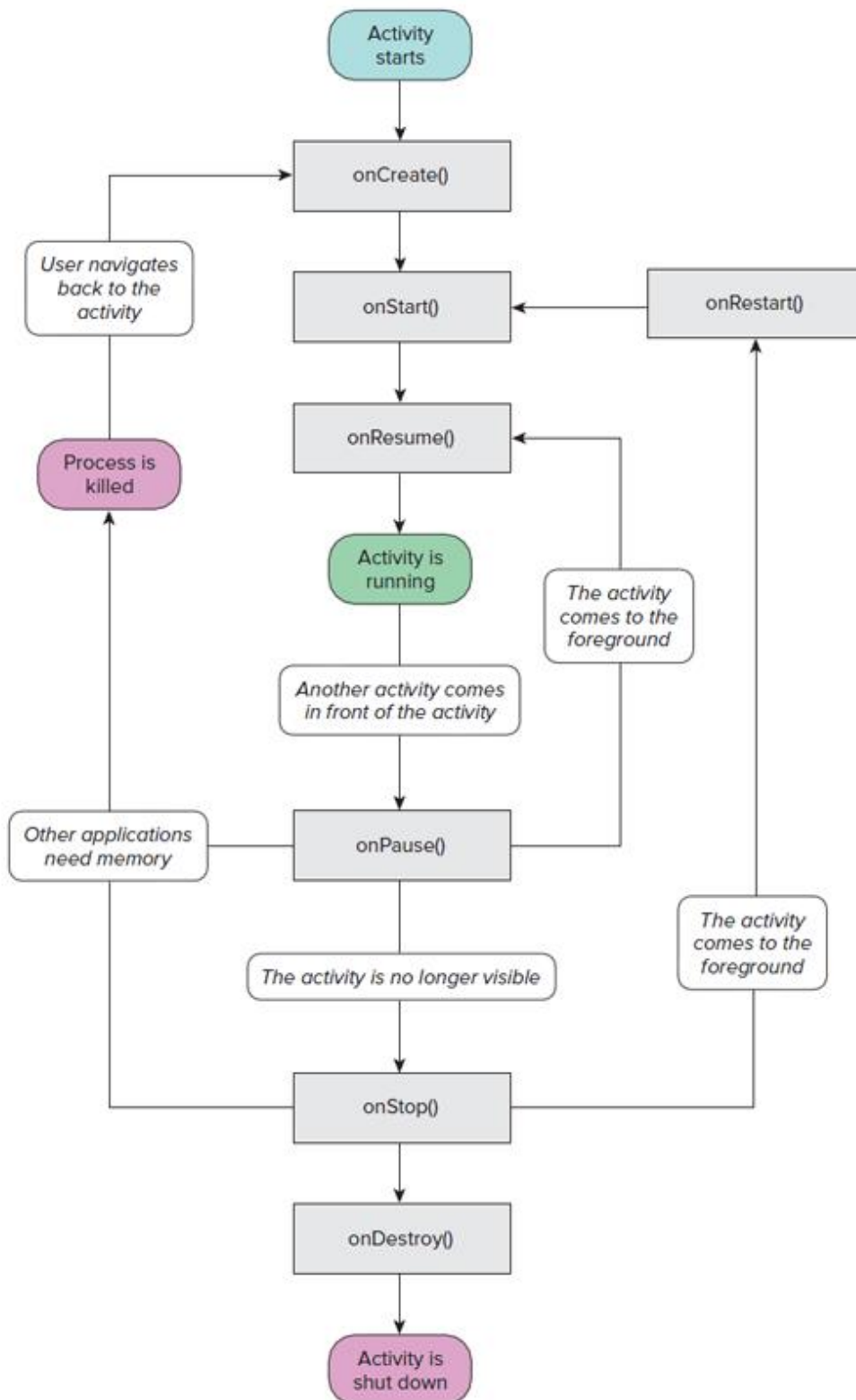
Components of Android Application: Following six components provide basic building blocks for an android application:

- **Activities:** An activity represents a single screen with a user interface. An activity is implemented as a subclass of **android.app.Activity** class. It's like a frame in awt(java).
- **Services:** A service is a component that runs in the background. They are used to perform regular processing that needs to continue even when our application is not active or visible.
- **Content Providers:** A content provider manages a shared set of application data. We can store the data in the file system, SQLite database, on the web, or any other persistent storage location our application can access. Content providers are also useful for reading and writing data that is private to our application and not shared.
- **Broadcast Receivers:** A broadcast receiver is a component that responds to system-wide broadcast announcements. Many broadcasts originate from the system—for example, a broadcast announcing that the screen has turned off, the battery is low, or a picture was captured. Applications can also initiate broadcasts—for example, to let other applications know that some data has been downloaded to the device and is available for them to use.
- **Intents:** Intent is used to invoke components. An Intent is exactly what it describes. It's an "intention" to do an action. An Intent is basically a message to say you did or want something to happen. For example using Intents we can broadcast message system wide or to a specific activity or service.
- **Notifications:** Notifications let us signal users without interrupting their current Activities. These are preferred to get user's attentions for background attention.

Activity Class (android.app package): An Activity represents user interface of our application. An application can have zero or more activities. We must extend **android.app.Activity** class to create user interface of our applications. Activity class defines some methods, which represents various stages of an activity we have to override them in order to provide functionality in our application. These methods are:

1. **onCreate():** it is called when activity is first created.
2. **onStart():** it is called when activity becomes visible to user.
3. **onResume():** it is called when starts interaction with user.
4. **onPause():** it is called when current activity is being paused and other activity is being resumed.
5. **onStop():** it is called when activity is no longer visible to user.
6. **onDestroy():** it is called when activity is destroyed by the system.
7. **onRestart():** it is called when activity has been stopped and restarting again.

Activity Life Cycle: An activity's life span is represented by seven methods defined in Activity class. Following figure represents life cycle of an Activity.



Android Runtime [Application Management Software(AMS)]

Mobile devices, whether emulators or real, interact with an application using their own software, which is called Android Runtime. The Android Runtime is responsible for initializing, starting, pausing, resuming, and destroying an application. Besides these services, Android Runtime is responsible for installing and removing an application, as well. Android Runtime gracefully handles all the errors that occur during the installation and execution of applications without crashing the system.