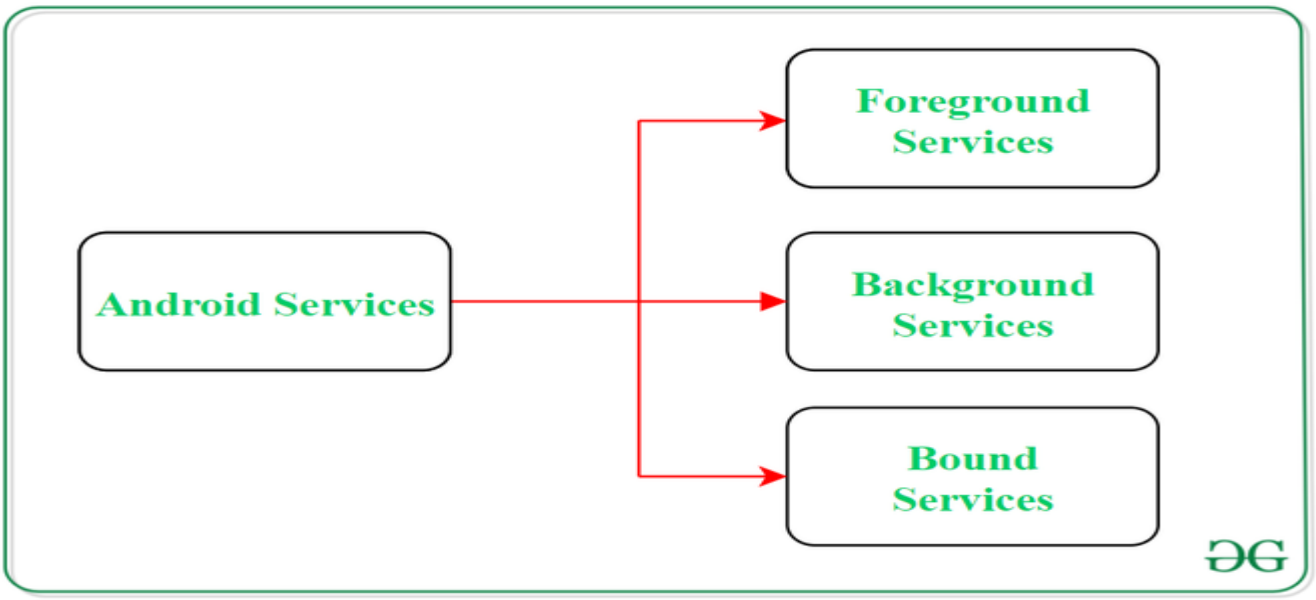
App Component: Services in Android

Services in Android are a special component that facilitates an application to run in the background in order to perform long-running operation tasks. The prime aim of a service is to ensure that the application remains active in the background so that the user can operate multiple applications at the same time. A user-interface is not desirable for android services as it is designed to operate long-running processes without any user intervention. A service can run continuously in the background even if the application is closed or the user switches to another application. Further, application components can bind itself to service to carry out Inter Process Communication (IPC). There is a major difference between android services and threads, one must not be confused between the two. Thread is a feature provided by the Operating system to allow the user to perform operations in the background. While service is an android component that performs a long-running operation about which the user might not be aware of as it does not have UI.

Types of Android Services



1. Foreground Services:
Services that notify the user about its ongoing operations are termed as Foreground Services. Users can interact with the service by the notifications provided about the ongoing task. Such as in downloading a file, the user can keep track of the progress in downloading and can also pause and resume the process.

2. Background Services:
Background services do not require any user intervention. These services do not notify the user about ongoing background tasks and users also cannot access them. The process like schedule syncing of data or storing of data fall under this service.

3. Bound Services:
This type of android service allows the components of the application like activity to bound themselves with it. Bound services perform their task as long as any application component is bound to it. More than one component is allowed to bind themselves with a service at a time. In order to bind an application component with a service bindService() method is used.

The Life Cycle of Android Services

In android, services have 2 possible paths to complete its life cycle namely Started and Bounded.

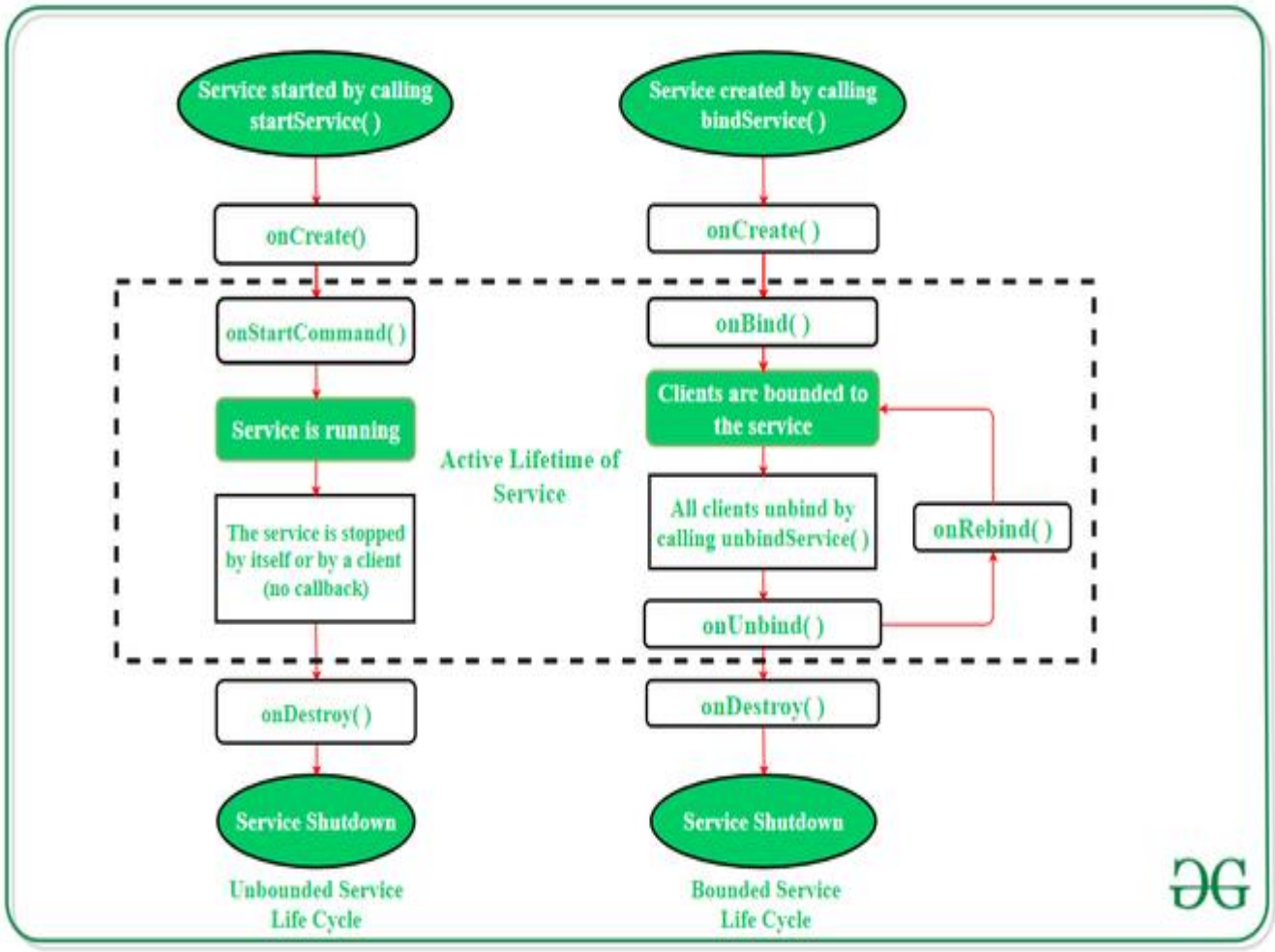
1. Started Service (Unbounded Service):

By following this path, a service will initiate when an application component calls the startService() method. Once initiated, the service can run continuously in the background even if the component is destroyed which was responsible for the start of the service. Two option are available to stop the execution of service:

- By calling stopService() method,
- The service can stop itself by using stopSelf() method.

2. Bounded Service:

It can be treated as a server in a client-server interface. By following this path, android application components can send requests to the service and can fetch results. A service is termed as bounded when an application component binds itself with a service by calling bindService() method. To stop the execution of this service, all the components must unbind themselves from the service by using unbindService() method.



Fundamentals of Android Services

A user-defined service can be created through a normal class which is extending the class Service. Further, to carry out the operations of service on applications, there are certain callback methods

which are needed to be overridden. The following are some of the important methods of Android Services:

onStartCommand()	The Android service calls this method when a component(eg: activity)requests to start a service using startService(). Once the service is started,it can be stopped explicitly using stopService() or stopSelf() methods.
onBind()	This method is mandatory to implement in android service and is invoked whenever an application component calls the bindService() method in order to bind itself with a service. User-interface is also provided to communicate with the service effectively by returning an IBinder object. If the binding of service is not required then the method must return null.
onUnbind()	The Android system invokes this method when all the clients get disconnected from a particular service interface.
onRebind()	Once all clients are disconnected from the particular interface of service and there is a need to connect the service with new clients, the system calls this method.
onCreate()	Whenever a service is created either using onStartCommand() or onBind(), the android system calls this method. This method is necessary to perform a one-time-set-up.
onDestroy()	When a service is no longer in use, the system invokes this method just before the service destroys as a final clean up call. Services must implement this method in order to clean up resources like registered listeners, threads, receivers, etc.

Example of Android Services

Playing music in the background is a very common example of services in android. From the time when a user starts the service, music play continuously in the background even if the user switches to another application. The user has to stop the service explicitly in order to pause the music.