

Quadrant II – Transcript and Related Materials

Programme: Bachelor of Science (Third Year)

Subject: Computer Science

Course Code: CSC108

Course Title: Mobile Application Development

Unit: 03

Module Name: Services: Services life cycle & Inter Process Communication (AIDL Services)

Name of the Presenter: Ms Bhakti Gawas

Notes:

Life Cycle of Service

- The service lifecycle is the process from when it is created to when it is destroyed can follow either of the below two paths:
 - A started service.
 - A bound service.

Life Cycle of Service: A started service

- The service is created when an application component calls `startService()`.
- The service then runs indefinitely and must stop itself by calling `stopSelf()`.
- Another component can also stop the service by calling `stopService()`.
- When the service is stopped, the system destroys it.
- A service is started when component (like activity) calls **`startService()`** method, and it runs in the background indefinitely.
- A service is stopped by **`stopService()`** method. The service can stop itself by calling the **`stopSelf()`** method.

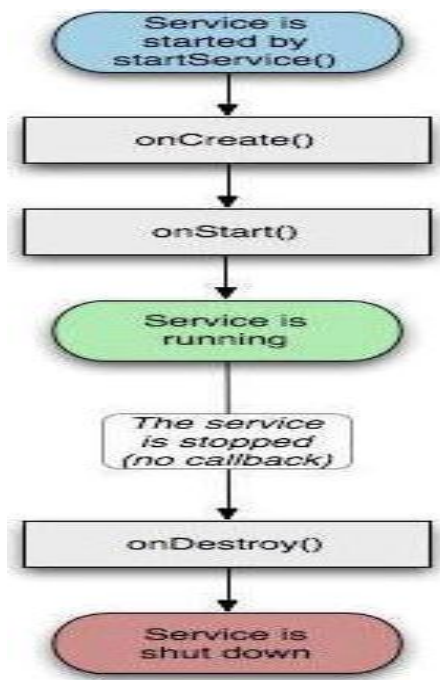


Fig 1: Life cycle of service: A started service

Life Cycle of Service: A bound service

- A service is termed as bounded when an application component bind itself with a service by calling `bindService()`.
- The service is created when another component (a client) calls `bindService()`.
- The client then communicates with the service through an `IBinder` interface.
- The client can close the connection by calling `unbindService()`.
- Multiple clients can bind to the same service and when all of them unbind, the system destroys the service.
- A Bound Service allows components (e.g. Activity) to bind to the services, send requests, and receive response.
- Bound Service can serve components running on different processes.

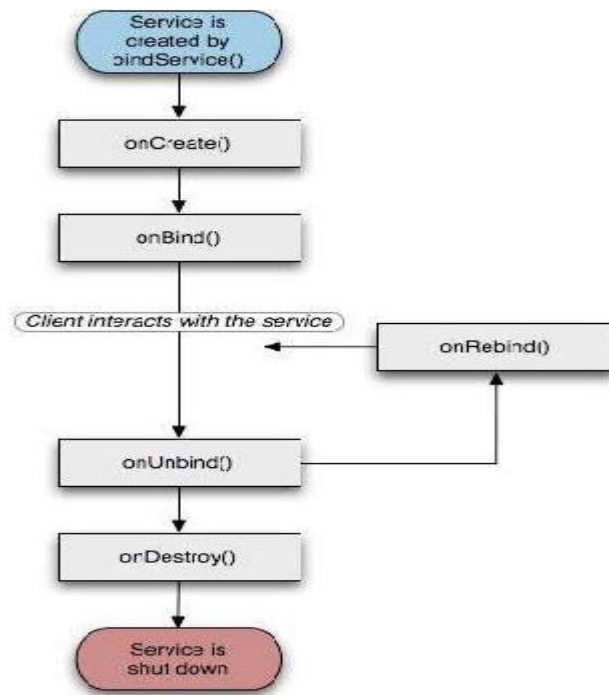


Fig 2: Life cycle of service: A Bound service

Inter Process Communication (AIDL Services)

- Android supports mechanism to establish an inter process communication between the android service (in process A) and the activity (in process B).
- To facilitate inter process communication, there is need to decompose the objects into primitives that the operating system can understand. In android this is done using AIDL(Android Interface Definition Language).
- The way to establish interprocess communication (IPC) is through, the use of AIDL.

Android interface Definition Language (AIDL)

- AIDL allows you to define the programming interface that both the client and service agree upon in order to communicate with each other using interprocess communication (IPC).
- On Android System, one process cannot normally access the memory of another process.

- To communicate between the processes there is a need to define programming interface to decompose the objects into primitives that the operating system can understand.
- Using AIDL is necessary only if the clients from different applications are allowed to access the service for IPC.

Defining an AIDL Interface

- ✓ To create a bounded service using AIDL, follow below steps.
- ✓ Create the .aidl file:
 - This file defines the interface with method signatures.
- ✓ Implement the interface:
 - The Android SDK tools generate an interface based on .aidl file.
 - This interface has an abstract class named Stub that extends Binder and implements methods from AIDL interface, extend Stub class and implement methods.
- ✓ Expose the interface to clients:
 - Implement a service and override onBind() to return implementation of the Stub class.

Summary:

- The service lifecycle process follows either of two paths: A started service or a bound service.
- The service is created when an application component calls **startService()**.
- The service is created when another component (a client) calls **bindService()**.
- Android supports mechanism to establish an inter process communication between processes.
- To establish interprocess communication (IPC) android supports the use of Android Interface Definition Language (AIDL).

