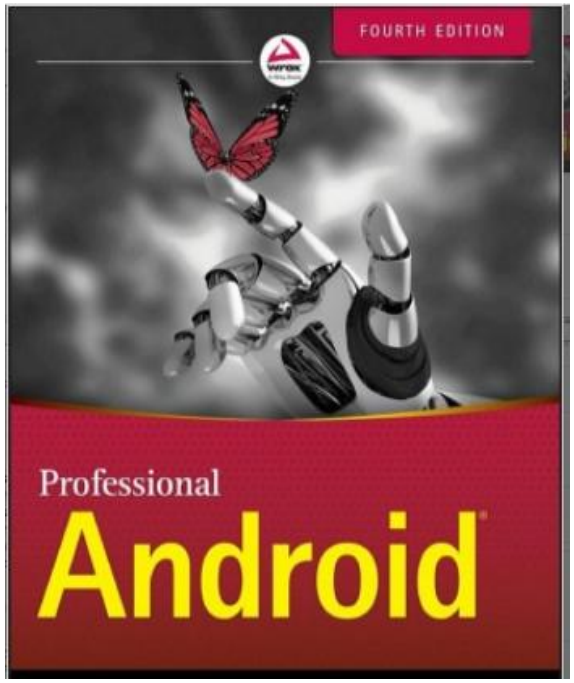
Mobile Phone Security

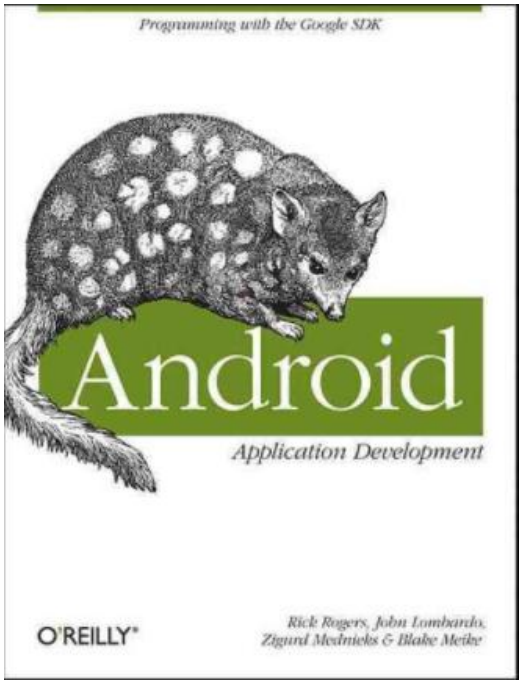
Dr. Digvijaysinh Rathod
Associate Professor
(Cyber Security and Digital Forensics)
Institute of Forensic Science
Gujarat Forensic Sciences University

digvijay.rathod@gfsu.edu.in

Android Service

Reference





www.developer.google.com

https://data-flair.training/blogs/android-service-tutorial/

The application framework : Explicit Intent with Service

✓Android Services are the application components that **run in the background**.

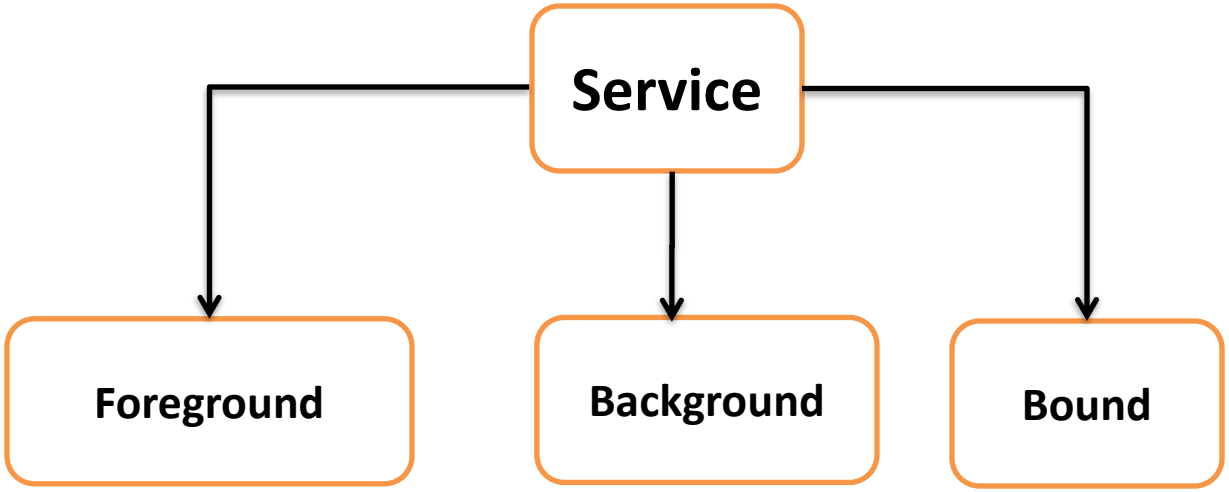
✓We can understand it as a process that **doesn't need any direct user interaction.**

✓As they perform **long-running processes** without user intervention, they have no User Interface.

✓They can be connected to other components and do inter-process communication (IPC).

✓Types of Android Services

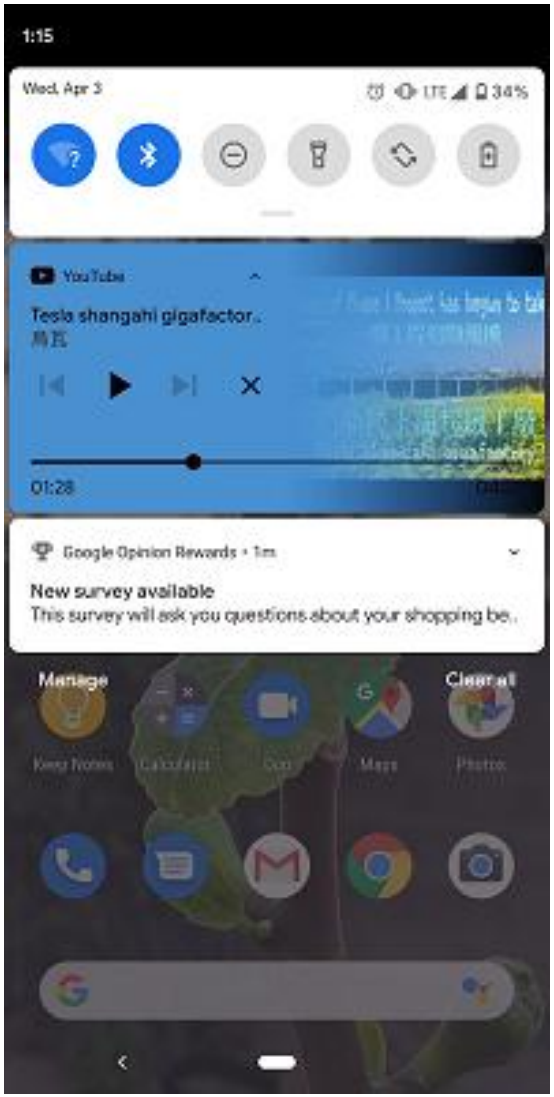
✓When we talk about services, they can be of three types as shown in the figure below:



- ✓The working of these three services is below:
  - ✓**Foreground services** are those services that are **visible to the users**.
  - ✓The users can interact with them at ease and track what's happening.
  - ✓These services continue to **run even when users are** using other applications.
  - ✓Example: The perfect example of this is **Music Player and Downloading.**

- ✓The working of these three services is below:
  - ✓**Background Services:** These services run in the **background,** such that the user **can't see or access them**.
  - ✓These are the tasks that don't need the user to know them.
  - ✓**Syncing and Storing data can be the best example.**

The application framework : Explicit Intent with Service



- ✓The working of these three services is below:
  - ✓**Bound Services :** Bound service runs as long as some other application component is bound to it.
  - ✓Many components can bind to one service at a time, but once they all unbind, the service will destroy.
  - ✓To bind an application component to the service, bindService() is used.

✓Lifecycle of Android Services:

✓Android services life-cycle can have two forms of services and they follow two paths, that are:

1. Started Service
2. Bounded Service

1. Started Service

✓A service becomes started only when an **application component calls startService()**.

✓It performs a single operation and doesn't return any result to the caller.

✓Once this service starts, it runs in the background even if the component that created it destroys.

✓This service can be stopped only in one of the two cases:

- ✓By using the **stopService()** method.
- ✓By stopping itself using the **stopSelf()** method.

2. Bound Service

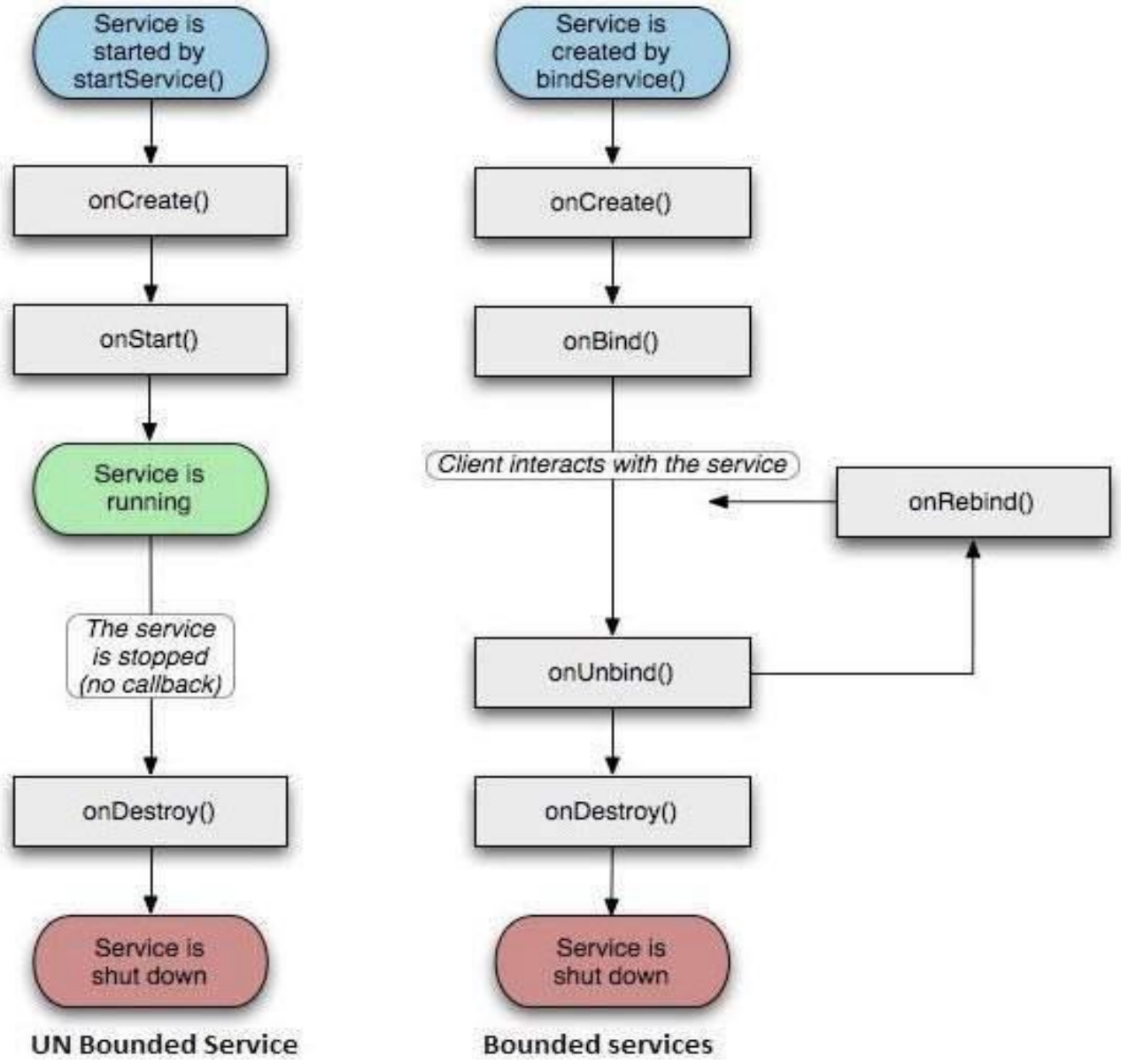
✓A service is bound only if an application component binds to it **using bindService().**

✓It gives a **client-server relation** that lets the components interact with the service.

✓The components can send requests to services and get results.

✓This service runs in the background as long as another application is bound to it. Or it can be unbound according to our requirement by using the **unbindService() method.**

Life Cycle of Android Service



✓ Methods of Android Services

- ✓ The service base class defines certain callback methods to perform operations on applications. When we talk about Android services it becomes quite obvious that these services will do some operations and they'll be used. The following are a few important methods of Android services :
- ✓ onStartCommand()
- ✓ onBind()
- ✓ onCreate()
- ✓ onUnbind()
- ✓ onDestroy()
- ✓ onRebind()

1. onStartCommand()

- ✓ The system calls this method whenever a component, say an activity requests **'start' to a service, using startService().**
- ✓ Once we use this method it's our duty to stop the service using **stopService() or stopSelf().**

2. onBind()

- ✓ This is invoked when a component wants to bind with the service by calling **bindService()**.
- ✓ In this, we must provide an interface for clients to communicate with the service. For **interprocess communication,** we use the IBinder object.
- ✓ It is a must to implement this method.
- ✓ If in case binding is not required, we should return null as implementation is mandatory.

3. onUnbind()

- ✓ The system invokes this when all the clients disconnect from the interface published by the service.

4. onRebind()

- ✓ The system calls this method when new clients connect to the service. The system calls it after the onBind() method.

5. onCreate()

- ✓ This is the first callback method that the system calls when a **new component starts the service.**
- ✓ We need this method for a one-time set-up.

6. onDestroy()

- ✓ This method is the final clean up call for the system.
- ✓ The system invokes it just before the service destroys.
- ✓ It cleans up resources like threads, receivers, registered listeners, etc.

Life Cycle of Android Service

- ✓ Implementation of Android Services and discussion of source code. (Service_Example)

Mobile Phone Security

Dr. Digvijaysinh Rathod
Associate Professor
(Cyber Security and Digital Forensics)
Institute of Forensic Science
Gujarat Forensic Sciences University

digvijay.rathod@gfsu.edu.in