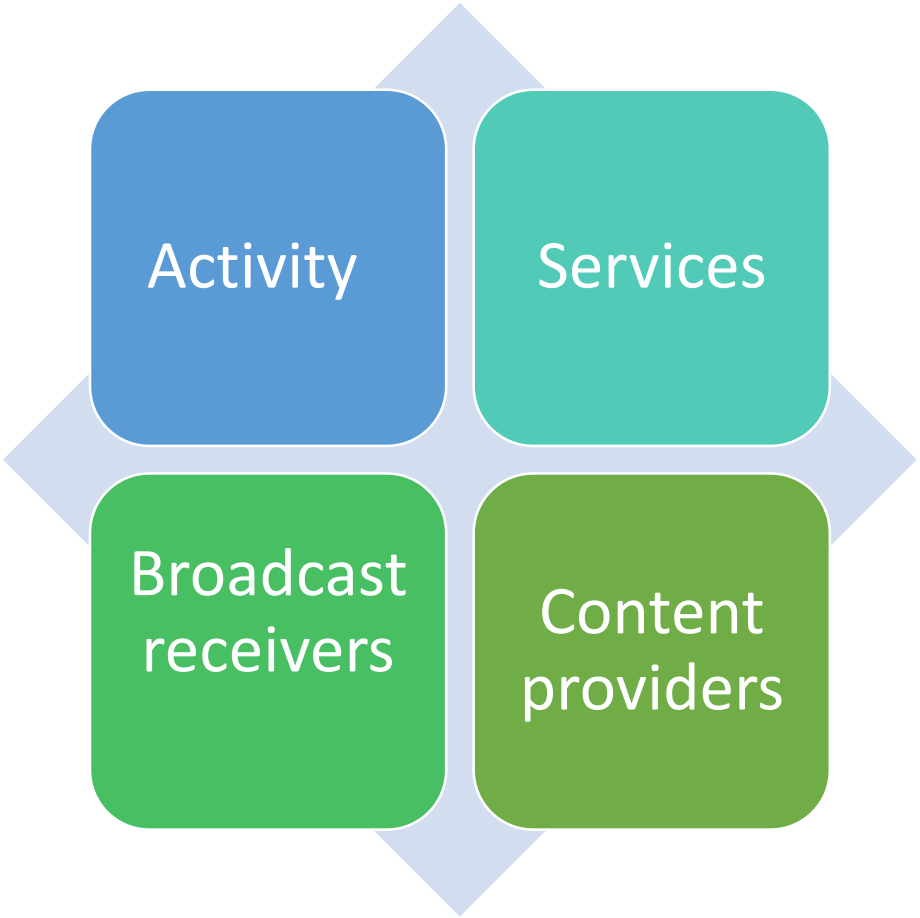
Summer 2, 2019 - CS 4520/CS5520 – Mobile Application Development

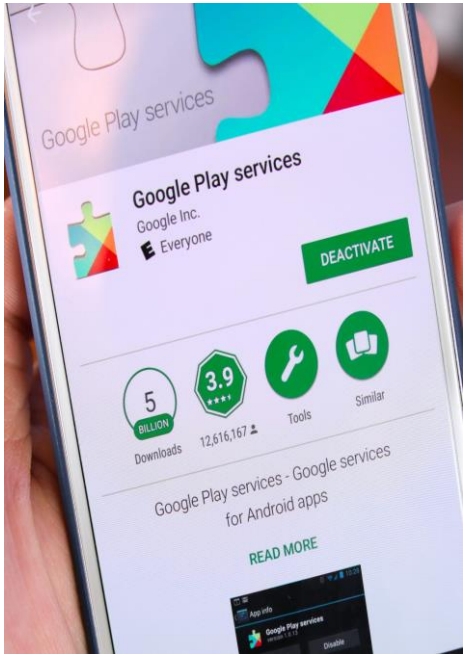
Pratheep Kumar Paranthaman, Ph.D.,

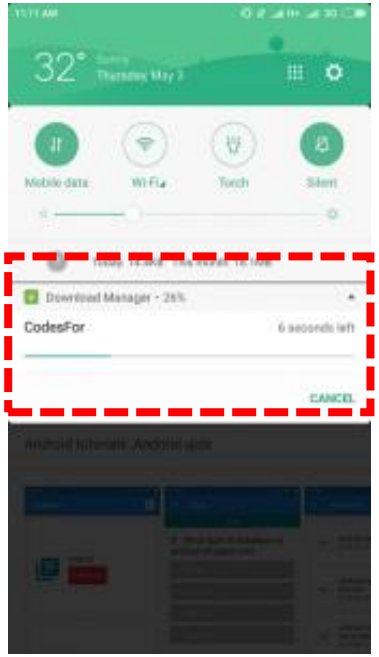
Android Components



Services in Android

- Android component that runs in **background** to perform long-running processes.
- No need to present a UI for the services
- By default the services run on the Main Thread





[codesfor.in/downloading-files-in-background-using-download-manager-in-android/]

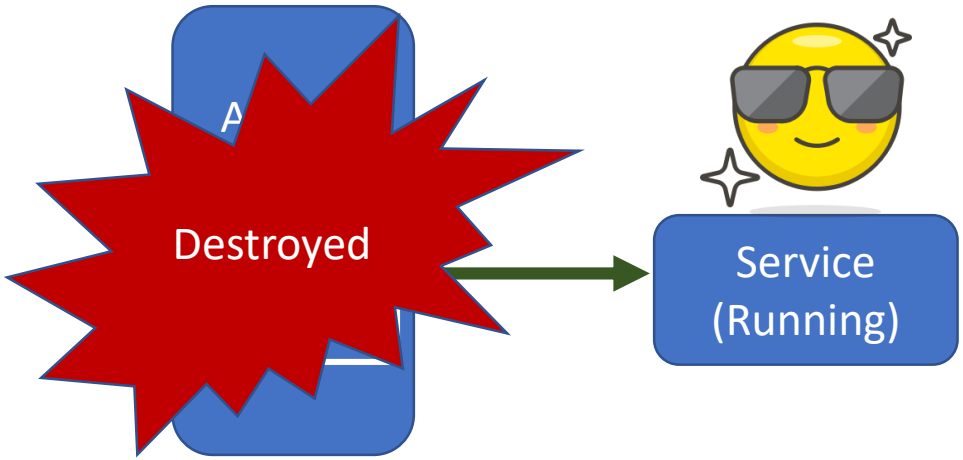
[9to5google.com/2018/02/01/how-to-update-google-play-services-android-basics/]

Service Types

- Unbound(or **Started**) Services
  - Background
  - Foreground
- Bound Services

Unbound(or started) service

- Started by Android Components(Activity, Broadcast Receivers, Content providers or Service).
- Runs even after the component that started it has been destroyed(Example: Destroying an activity)
- Once **started runs indefinitely until it's stopped manually**/ killed by the Android System.

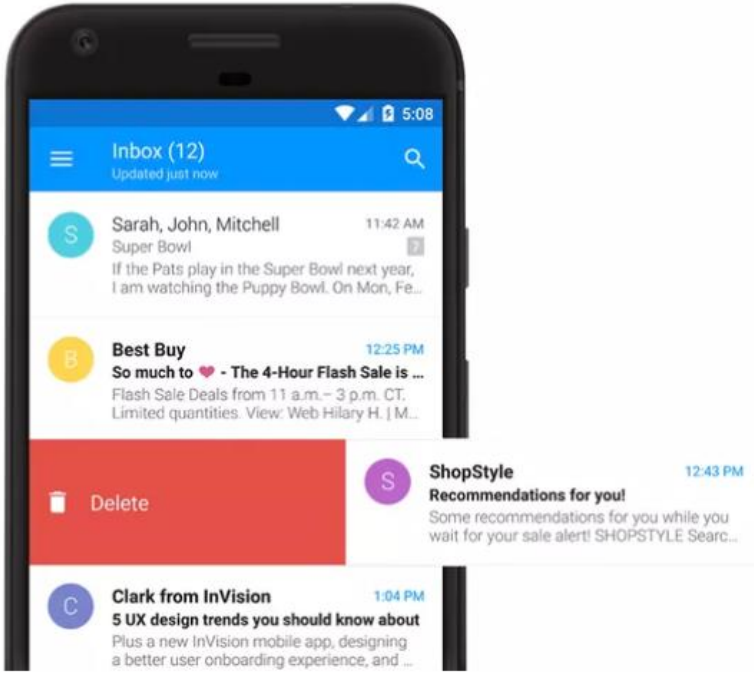


Background Services

- ***Operations performed are not notified to the users. – Totally background***

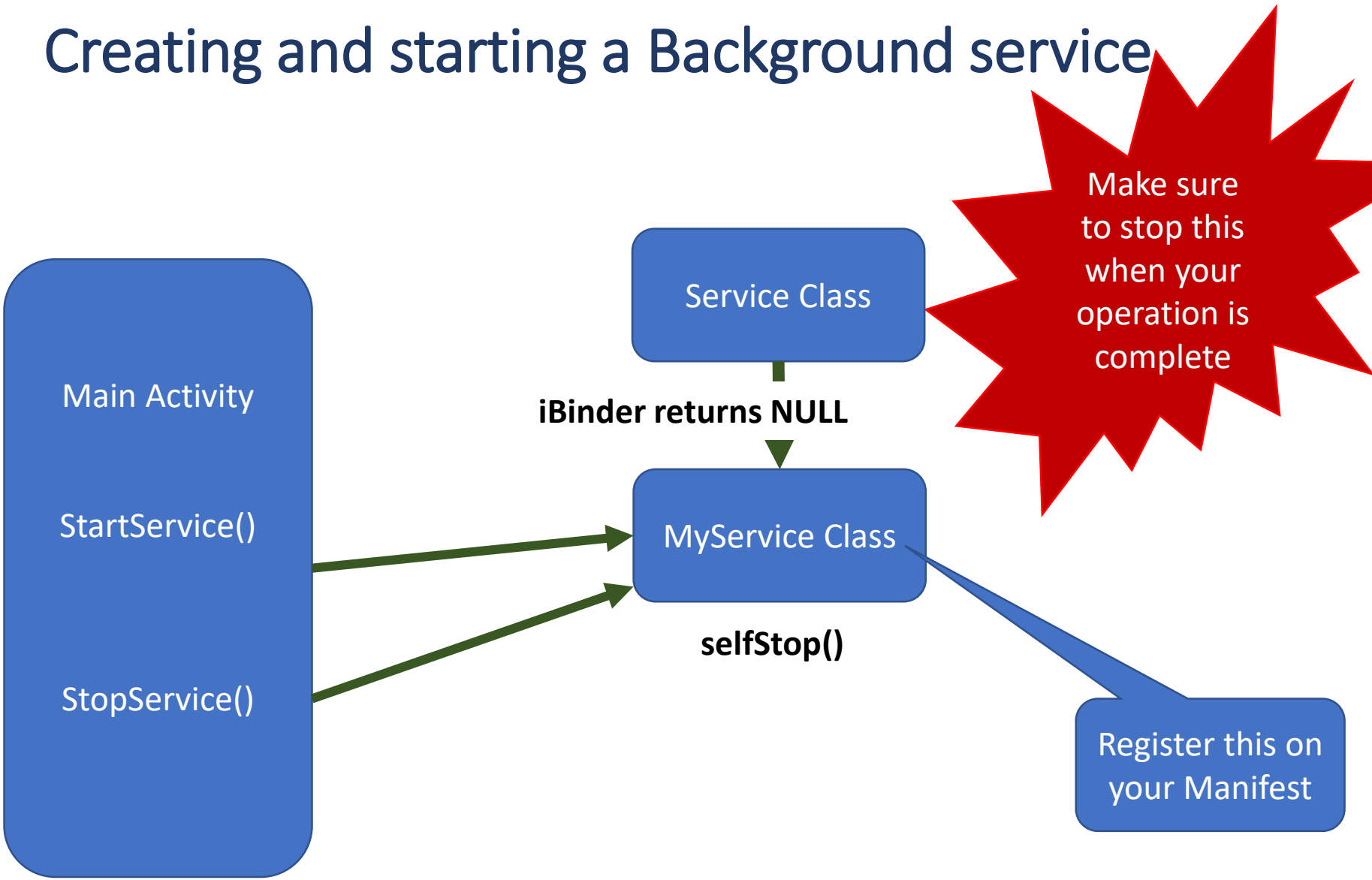
- Mail Application – Constantly listens for any new updates on the background

Example: Software updates, optimization services(clearing redundant data)



[theverge.com/2017/2/16/14642944/easilydo-email-available-android-play-store]

Creating and starting a Background service



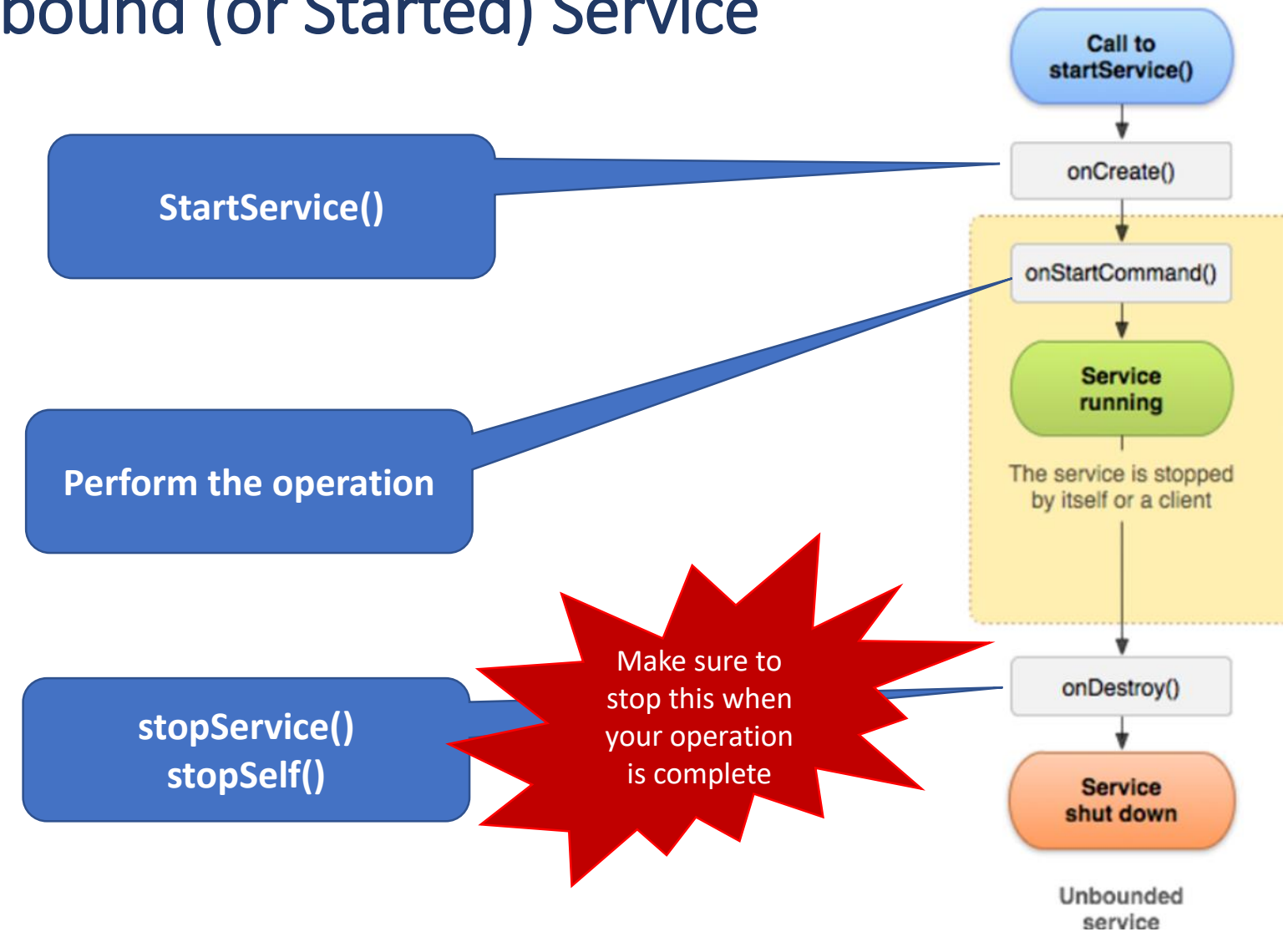
Unbound (or Started) Services – How to create one?

- Create a Class, which extends Services
- Register in Manifest File
- Implement the Methods (iBind is mandatory)
- Started by Any Application Component(Activity, Broadcast receivers, Content Providers or Services)
  - **startService()**

- **Service Running state -> onStartCommand()**

- **Service is Destroyed**
  - **stopService()**
  - **stopSelf() -> can be called only within a Service**

Unbound (or Started) Service



Demo Background Service

State of a Service?

- ***Service Flags***

  - **START_STICKY** -> If the system kills service, then **it recreates the service again and runs onStartCommand(). Intent Not Delivered (Data= NULL)**

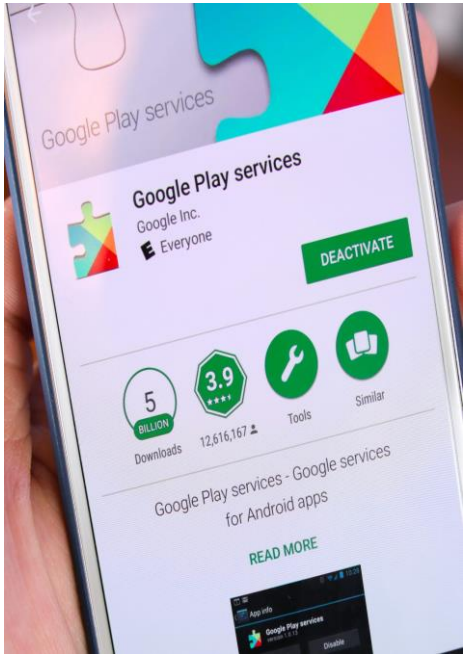
  - **START_REDELIVER_INTENT** -> if the system kills service, **then it recreates the service again and runs onStartCommand**(). **Intent is delivered** -> good to resume any background activities (**DATA= Restored**)

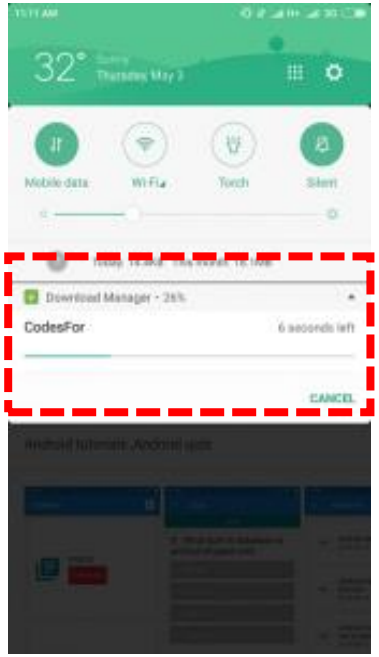
  - **START_NOT_STICKY ->** if the system kills service, then **it doesn't recreate the service again.** (Service is killed along with the Android component)
    - **Intent Not Delivered and DATA = NULL**

Issues with UI?

Services in Android

- Android component that runs in **background** to perform long-running processes.
- No need to present a UI for the services
- By default the services run on the Main Thread



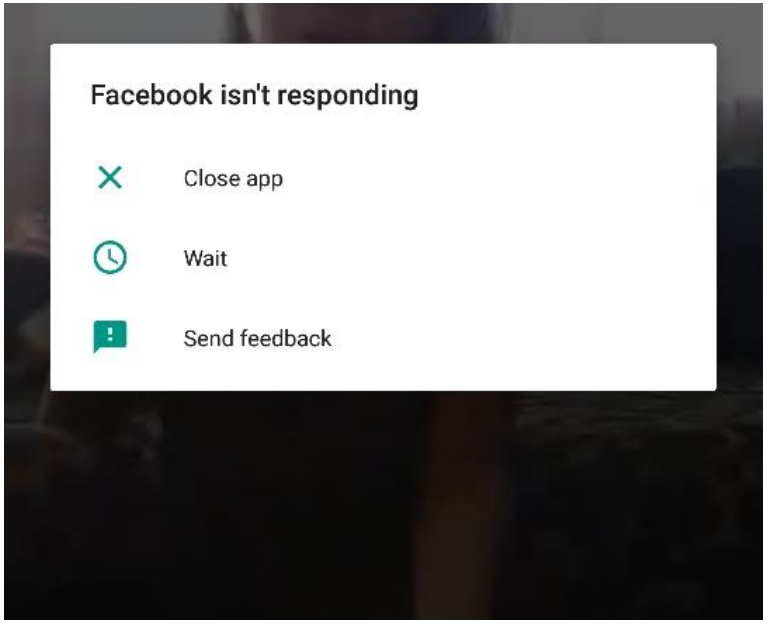


[codesfor.in/downloading-files-in-background-using-download-manager-in-android/]

[9to5google.com/2018/02/01/how-to-update-google-play-services-android-basics/]

Running services

On Main UI thread



[forums.oneplus.com/threads/facebook-app-not-responding.508764/]

On Worker Thread

[commons.wikimedia.org/wiki/File:Android_robot_skateboarding.svg]

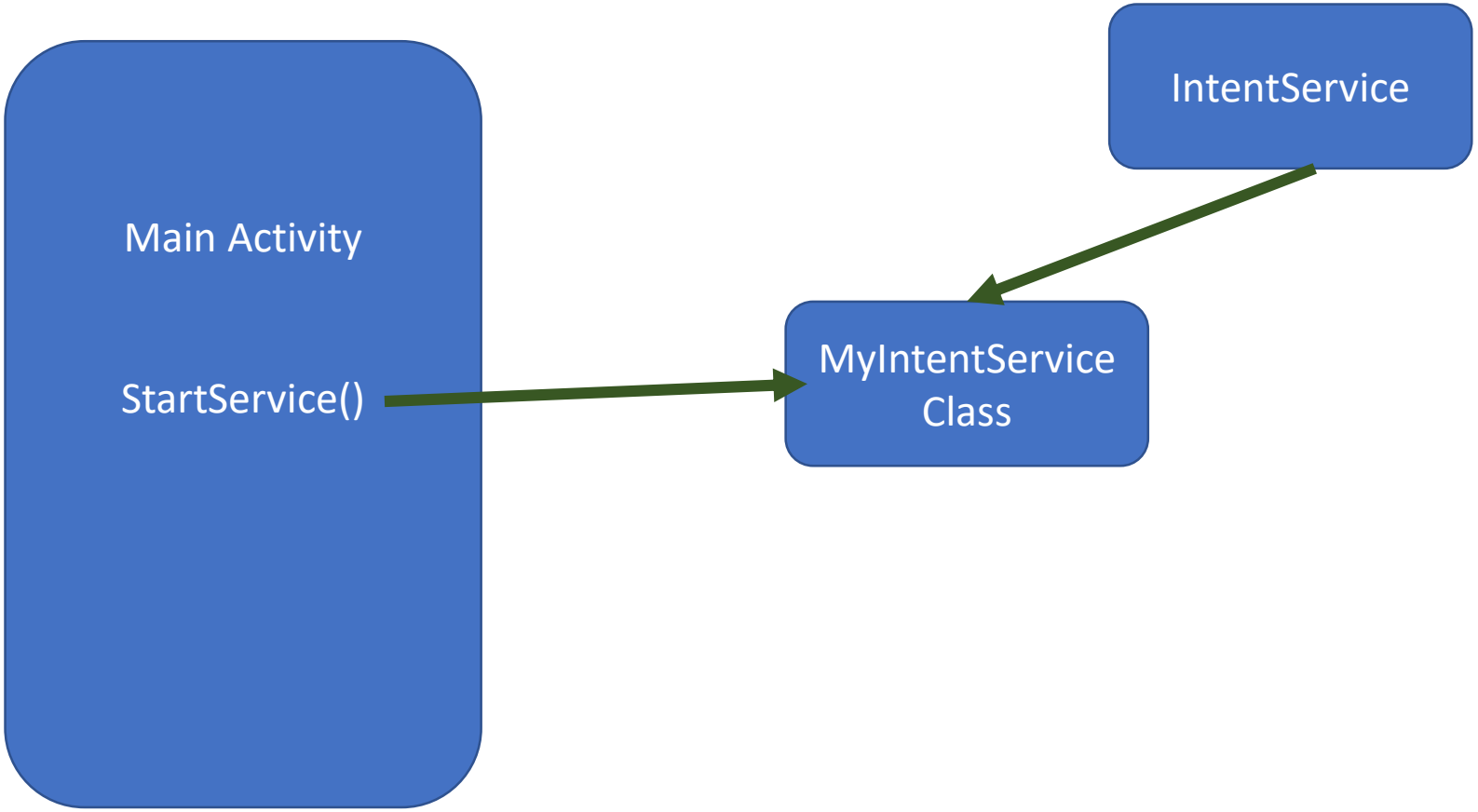
Creating a Worker Thread – Intent Service

Intent Service

- Sub Class of **Service**
- By default runs on a Worker Thread
- Performs one task at a time -> maintains queue for upcoming tasks
- Service stops when the task completes

- **Three Requirements**
  - Extend IntentService class
  - Create a custructor and pass in the name of the service
  - Override onHandleIntent()

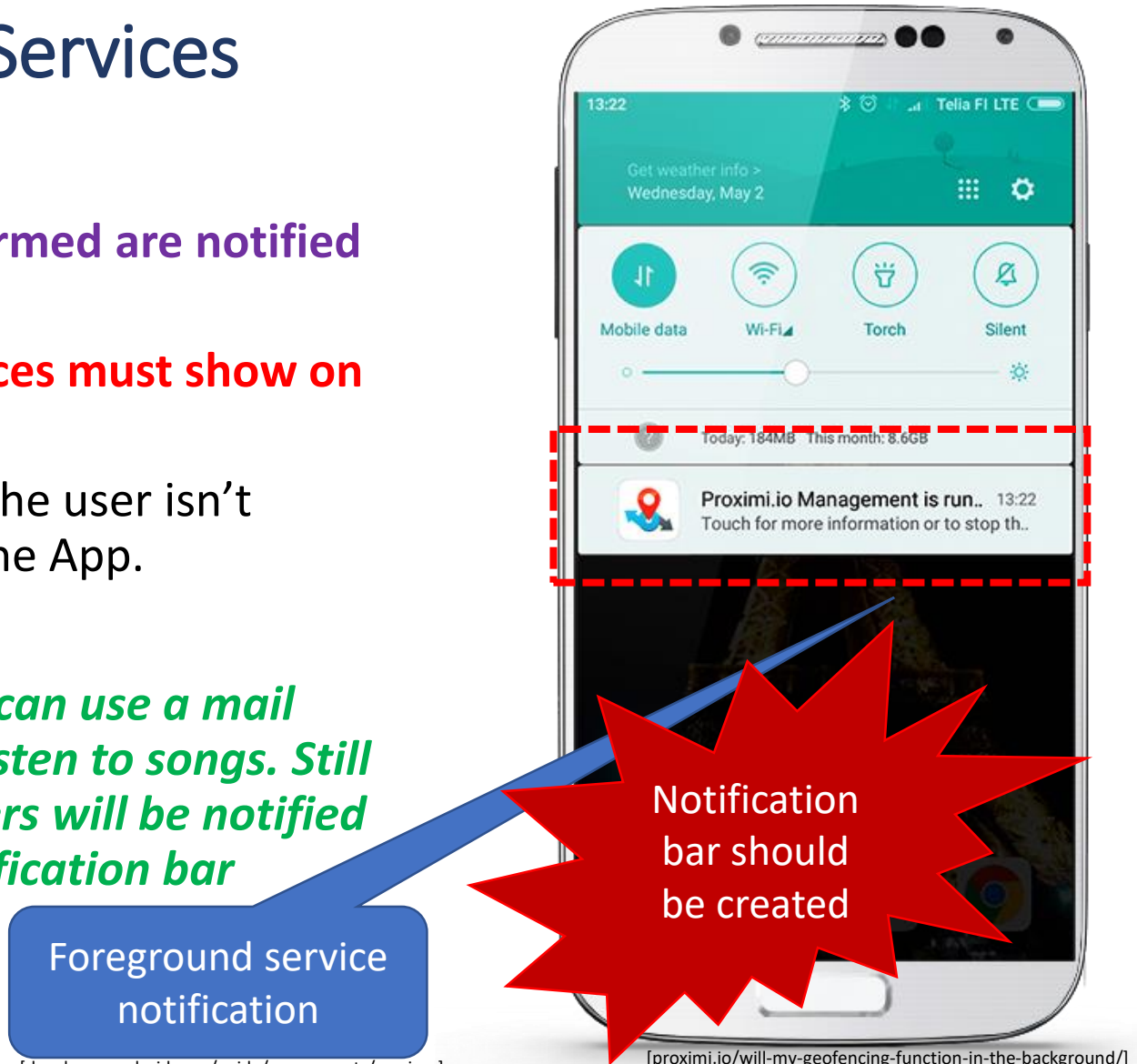
IntentService



Foreground Services

- **Operations performed are notified to the users.**
- **Foreground services must show on notification bar.**
- Runs even when the user isn't interacting with the App.

Example: You can use a mail application and listen to songs. Still your track numbers will be notified on the notification bar

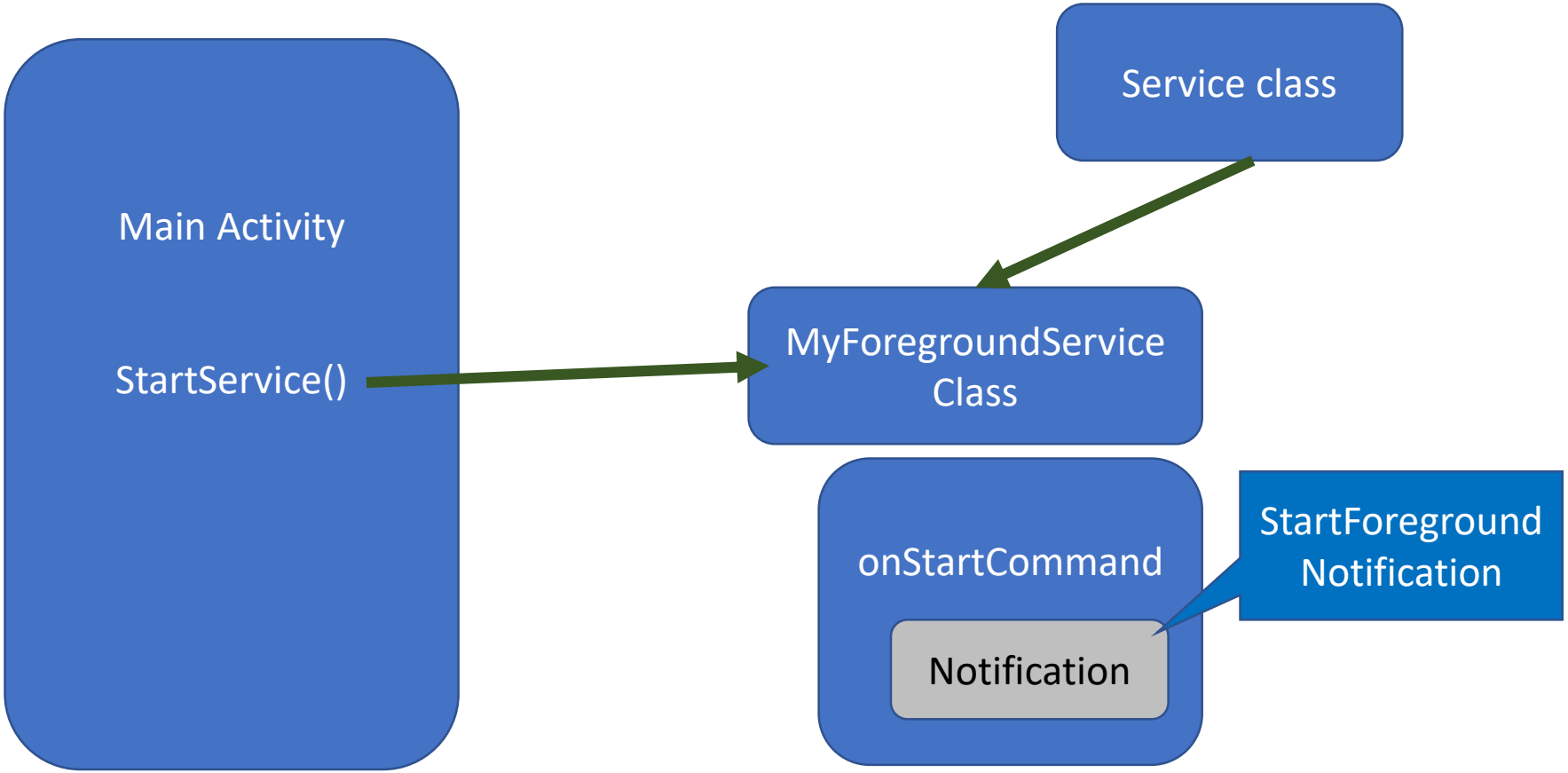


[developer.android.com/guide/components/services]

[proximi.io/will-my-geofencing-function-in-the-background/]

Creating Foreground Services

Foreground Service



Note: the methods for creating Foreground services differ in Android 9.0

Foreground Services

- Create a Foreground services class and extend Service
- Create a ***Notification Builder*** in onStartCommand()

Start Foregroundservice Notification in onStartCommand()

- Start Foreground by triggering the notification -> ***startForeGround(Unique identifier, Notification)***

- ***Start the service from your MainUI***

Creating Notification

- Create ***Notification builder*** and pass in the Context

```
Notification notification = new Notification.Builder(this)
```

- Set the following parameters
  - SetContentTitle
  - SetContentText
  - setSmallIcon
  - SetContentIntent**(Pending Intent)** -> Intent to be opened when user clicks on the Notification
  - Build

Pending Intent

- Gives permission to other application(notification) to execute an operation -> such as opening an activity

- Create a definition of the action

```
Intent notificationIntent = new Intent(this, MainActivity.class);
```

- Pass in the definition to Pending Intent

MainActivity opens up when the user taps on Notification

```
PendingIntent pendingIntent =
        PendingIntent.getActivity(this, 0, notificationIntent, 0);
```