Unit - 1

OS → Operating System

↓

initially loaded via (Boot program)

interaction via (Command line) or (GUI)

API → Application Program Interface

↳ used for making requests for services by Appl. Program.

Types :

Android	Bada	Symbian
Apple	Blackberry	Web

Android

↳ (mobile OS)

↳ developed via (Google)

↳ based on (Linux) kernel

Founder : Andy Rubin & Palo Alto

Date : Oct, 2003

& (17th Aug. 2005)

↑

Google acquired android

Ryx

Versions of Android:

1.0 → Alpha

2.0 → Beta

3.0 → Cupcake → 1.5

4.0 → Donut → 1.6

5.0 → Eclair → 2.0

6.0 → Froyo → 2.2

7.0 → Gingerbread → 2.3

Honeycomb → 3.0

Icecream Sandwich → 4.0

Jelly bean → 4.1

kitkat → 4.4

Lolipop → 5.0

Marshmallow → 6.0

Nought → 7.0

Oreo → 8.0

Pie → 9.0

Q → 10.0

R → 11.0

SAOWcom → 12.0

Tiramishu → 13.0

Features:

- memory management
- Process management
- Device management
- file management
- Security

Annexure No :

OHA : Open Handset Alliance

→ organization of (84) companies

→ led by Google.

→ develop open Standard for mobile

For promoting Android.,

members of OHA

↓ Prohibited

producing devices that are → incompatible forks of Android

Android architecture :

Applications : HOME, CONTACTS, PHONE, BROWSER

App. frame work : Activity manager, Window manager, Content Provider, Resource manager, Notification manager

Libraries : SGL, SSL, SQLite, OpenGL, libc

Android runtime : DVM, Core libraries

Linux kernel : USB Driver, Camera Driver, Bluetooth driver, Audio driver, Power management

Android components:

① Activities

- always used in App
- multiple activity in app
- each activity has various UI component
- Activity class — all work is done there.

② Services

- background processes — not displayed
- can be started & managed from activities
- higher priority
- eg. playing music, change it automatically, etc.

③ Intent

- provides a developer with the option to move from one screen to other
- consists of the operation to be performed & the activity class name has to be declared.

(4) Intent Reciever:

↳ when a fxn has to be executed in response to an event.

↳ alert via notification.

(5) Content Provider:

↳ Sharing data within the app.

↳ Apps can store data in shared preferences.

eg. SQLite, files, etc

(6) Widgets & notification:

→ Widgets for information : only info is displayed | Dynamic | Home screen manipulation.

→ widgets for collection : Collection of info. of same kind | Email App

→ Control panel widget : display most used features | Pause, Play

Hybrid widget → Combines above all 3's

Broadcast reciever:

↳ responds to Broadcast Messages from other applications.

↳ eg. App. can initiate brodcast saying / stating the info. about some data getting downloaded.

Broadcast Reciever will intercept this communication & will initiate appropriate action

↳ implemented as subclass of Broadcast Reciever class

ex. public class MyReciever extends BroadcastReciever {
public void onRecieve (Context, intent) { }
}

Drawable Resources:

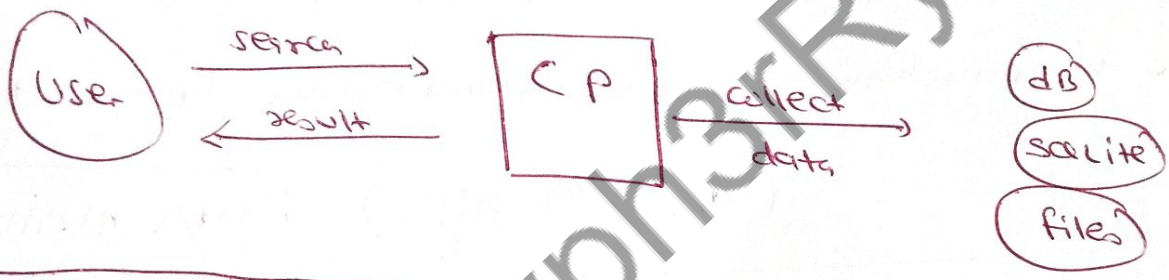
↳ graphic

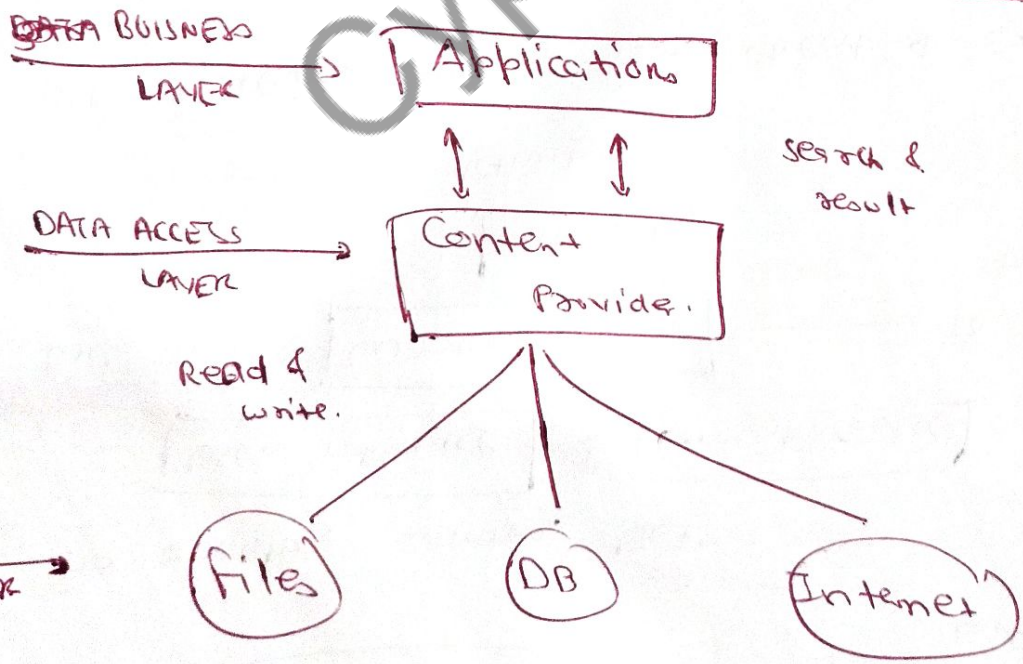
↳ Bitmap file represented via a BitmapDrawable class.

↳ Every drawable is stored as individual files in one of the res/drawable folder.

Content Provider

- share data bet^n Apps
- supply data on basis of request.
  - handled by "ContentResolverClass"
- implemented as → subclass of ContentProvider } must implement a standard set of API's that enable other apps to perform transactions.





Access data with Content Provider:

① Use (Content Resolver) → to communicate with CP for DATA ACCESS

② for enabling comm. beth [User interface] & [Content Resolver]

we use [Cursor Loader] → to run query asynchronously.

↓ called by Activity

③ CP recieves query from client & execute result.

DALVIK VM : (Register based VM)

↳ providing environment in which every android app. runs.

↳ each android app runs its own process.

↳ in own instance of

[multiple VMs] ← can run ← (Dalvik VM)

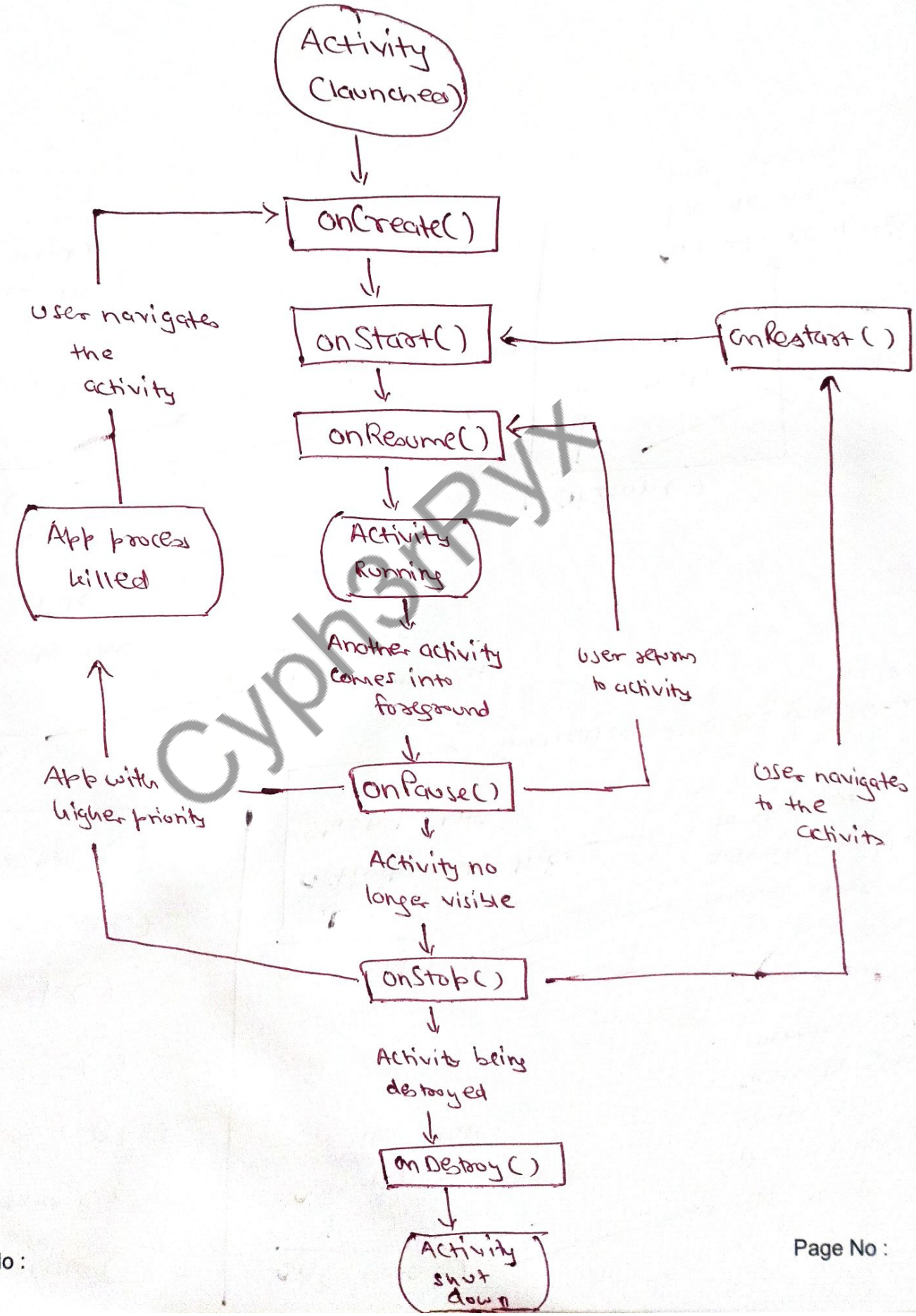
[.dex] format

↳ optimized for minimal memory footprint

Compilation:

[.java] → [.class] → [.dex]

Activity Life cycle :



Ryx

1. onCreate()
2. onStart()
3. onResume()
4. onPause()
5. onStop()
6. onRestart()
7. onDestroy()

onCreate()

↓

activity is created in the state.

→ OS calls this method when an activity gain memory.

onStart()

↓

activity enters the started state.

→ makes the activity visible to the user

→ initializes the UI maintaining code.

onResume()

↓

activity enters the foreground state

→ app communicates with user

onPause()

↓

user leaving your activity (no longer in foreground)

→ still be visible in multi window mode.

onStop()

↓

Any activity is on halt

→ hidden from user

→ App will cease to provide the user with any useful info.

onRestart()

- ↓ After stopped state, activity is called in this stage
- → User returns to the screen or resumes the activity that was stopped.
- → Once activity is stopped then & only then it will be restart.

onDestroy()

- ↓ App not in background then it reaches this stage.
- → User clicks back button & the activity will end after it completes the pause & stop lifecycle.

Intent:

- message passed b/w components
- used with startActivity() to involve activity, broadcast reciever..

→ used for

1. start service
2. launch an activity
3. Display a webpage
4. Dial on phone call
5. Broadcast a message
6. Display contacts.

(I) Implicit Intent: (doesn't specify the component)

↳ intent provides info of available components provided by the system to be involved.

eg. call, mail, phone, see any website.

```
Intent i = new Intent();
i.setAction(Intent.ACTION_VIEW);
i.setData(uri.parse("www.pu.ac.in"));
startActivity(i);
```

Ryx

Explicit Intent:

- specifies the component
- provides the external class to be invoked.

```
Intent i = new Intent (getApplicationContext(), Activity2.class);
startActivity(i);
```

Manifest Attributes of Android Activity:

- To implement activities in our App.
- we need to register them in the manifest file.
- for activities to work properly, we must manage their lifecycle properly.
  - Declare Activities
  - Declare Intent filters
  - Declare Permission

Annexure No :

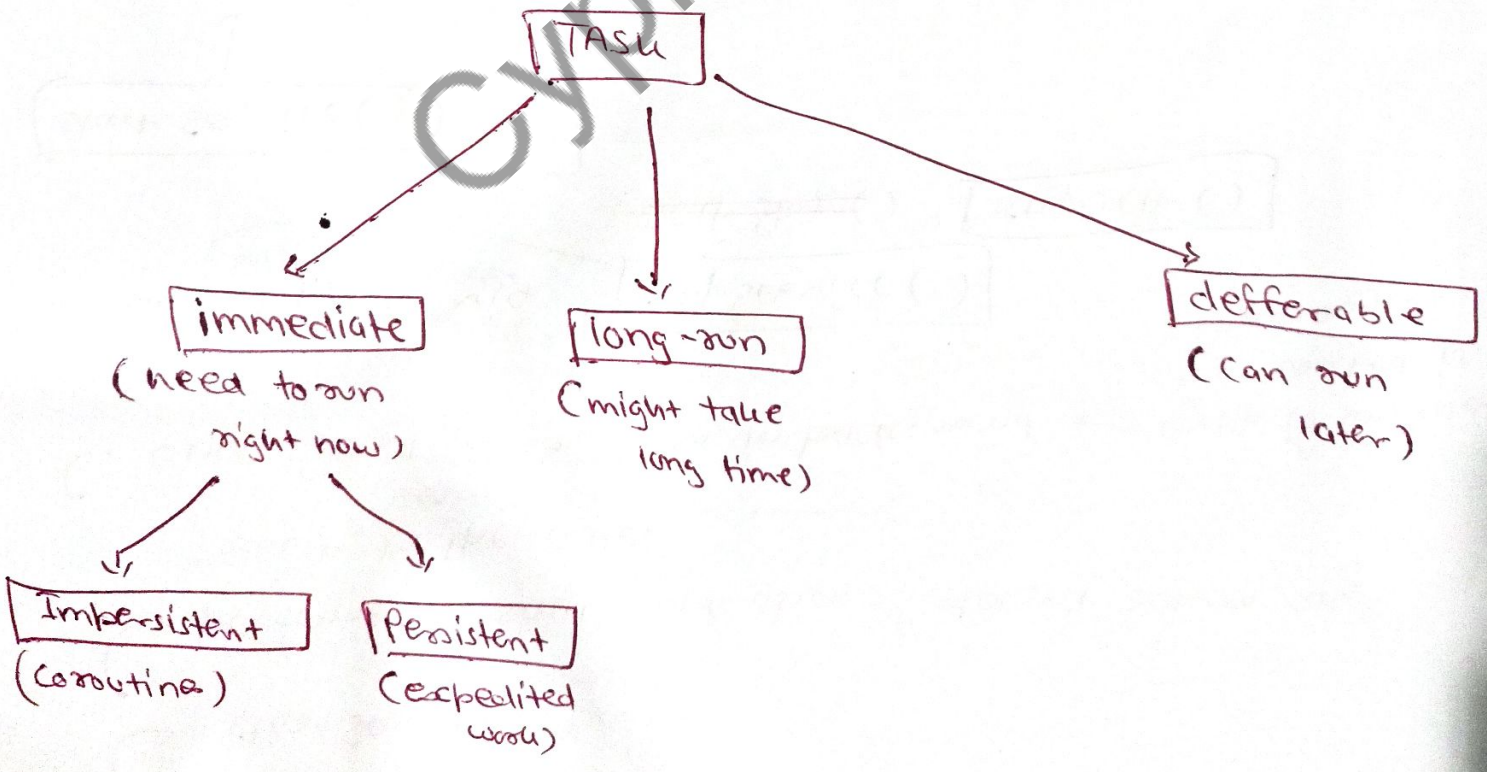
Types of services :

① Foreground services :

- service that notify user about its ongoing operations.
- visible service to users.
- eg. Music Player & Downloads.

② Background service :

- can't see them on screen.
- don't need the user to know them.
- eg. Syncing data & storing data



Lifecycle of Android Services:

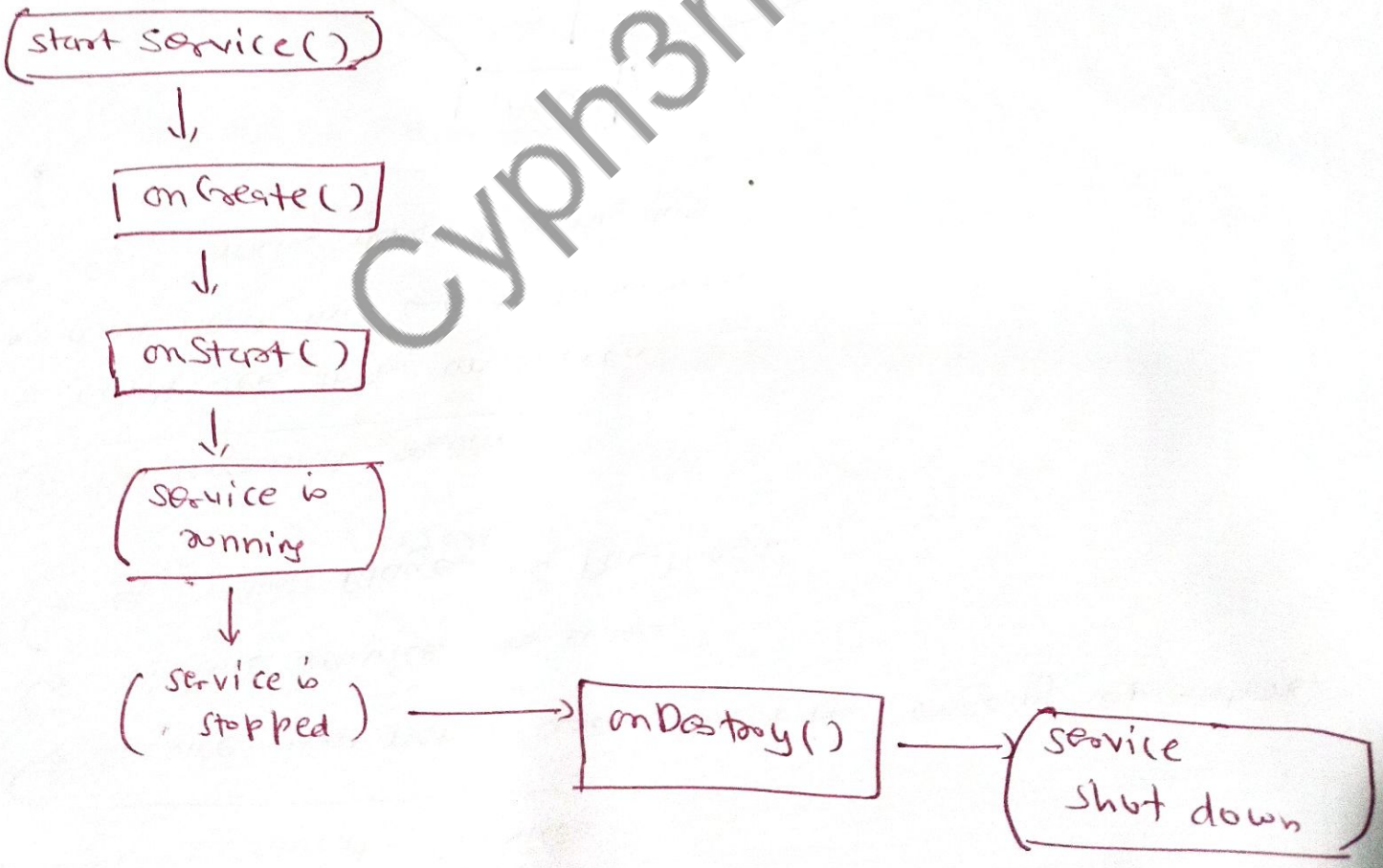
① Started service:

↳ startService()

↳ performs a single operation & doesn't return any result to the caller.

↳ once start → runs in background → even if component that created is destroyed.

can be stopped via stopService() / ~~self stop()~~ stopSelf()



② Bound Service :

↳ service runs in background

bindService ( ) ⟶ to bound the application component

unbind service ( ) ⟶ to unbind the