uc3m | Universidad Carlos III de Madrid
Grupo de investigación:
Computer Security Lab

Mobile Devices Security

Bachelor Degree in Informatics Engineering

2019

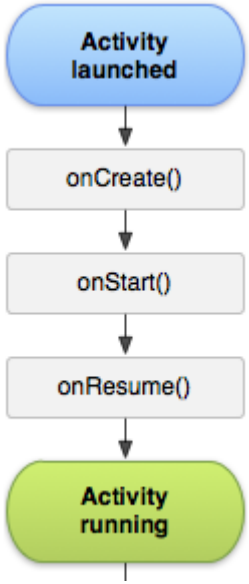
Agenda

- Life Cycle of an Activity
- Broadcast Receivers
- Services
- Content Providers

Life Cycle

Once an application has been launched, the main activity is called:

- onCreate() – the activity is called
- onStart() – the activity passes from invisible to visible, but it is not completely visible
- onResume() – it is completely visible and it is possible to interact with the activity
- Activity running – the activity interact with the user




```
public class MainActivity extends Activity {

     protected void onCreate(Bundle savedInstanceState) {
          super.onCreate(savedInstanceState);
          setContentView(R.layout.activity_main)
          Log.i("Lifecycle", "Se ha llamado al onCreate()");
     }

     protected void onStart() {
          super.onStart();
          Log.i("Lifecycle", "Se ha llamado al onStart()");
     }

     protected void onResume() {
          super.onResume();
          Log.i("Lifecycle", "Se ha llamado al onResume()");
     }

} /*  Fin de the class MainActivity*/
```

In order to see visually how all these methods are triggered, we use a "Log" object and its methods. They allow us to capture their entries in a console.

Life Cycle

During the actual activity (game, map etc.) we press “home” or “someone calls”, the activity won´t be destroyed but it hides:

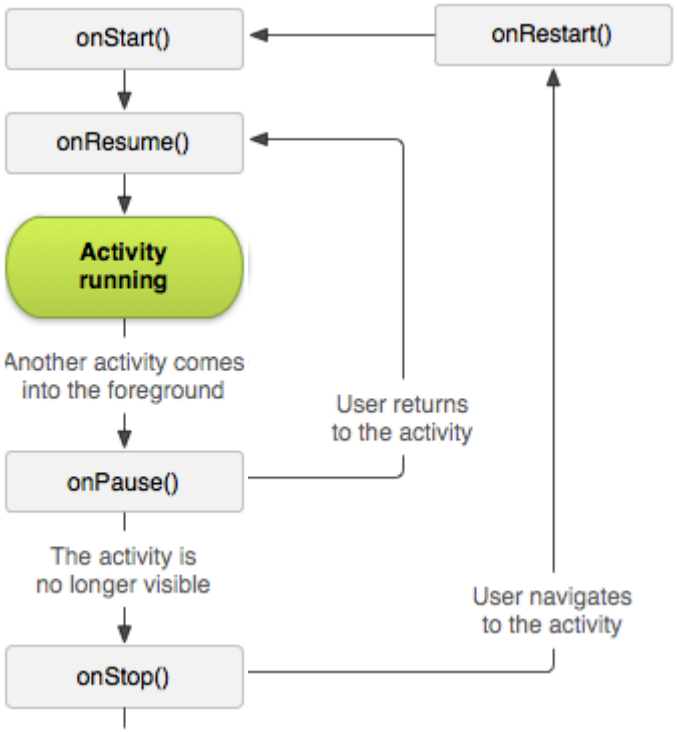
- onPause() – the activity passes to invisible. It takes like 2 seconds to pass to onStop(). This will give enough time to save the state of the application, save data etc.
- onStop() – the activity stops but it won´t be destroyed. It takes very short period of time if the application jumps to onDestroy() and about 2 seconds to pass to onRestart().

If the user returns to activity:

- onResume() reactivates the application.

If the application is stopped for other reasons and the user returns to the activity

- onRestart() and onStart() are called to restart the application



Life Cycle

If an activity is onPause() or onStop() and another with higher priority and/or higher memory resources is called, the first activity will be destroyed.

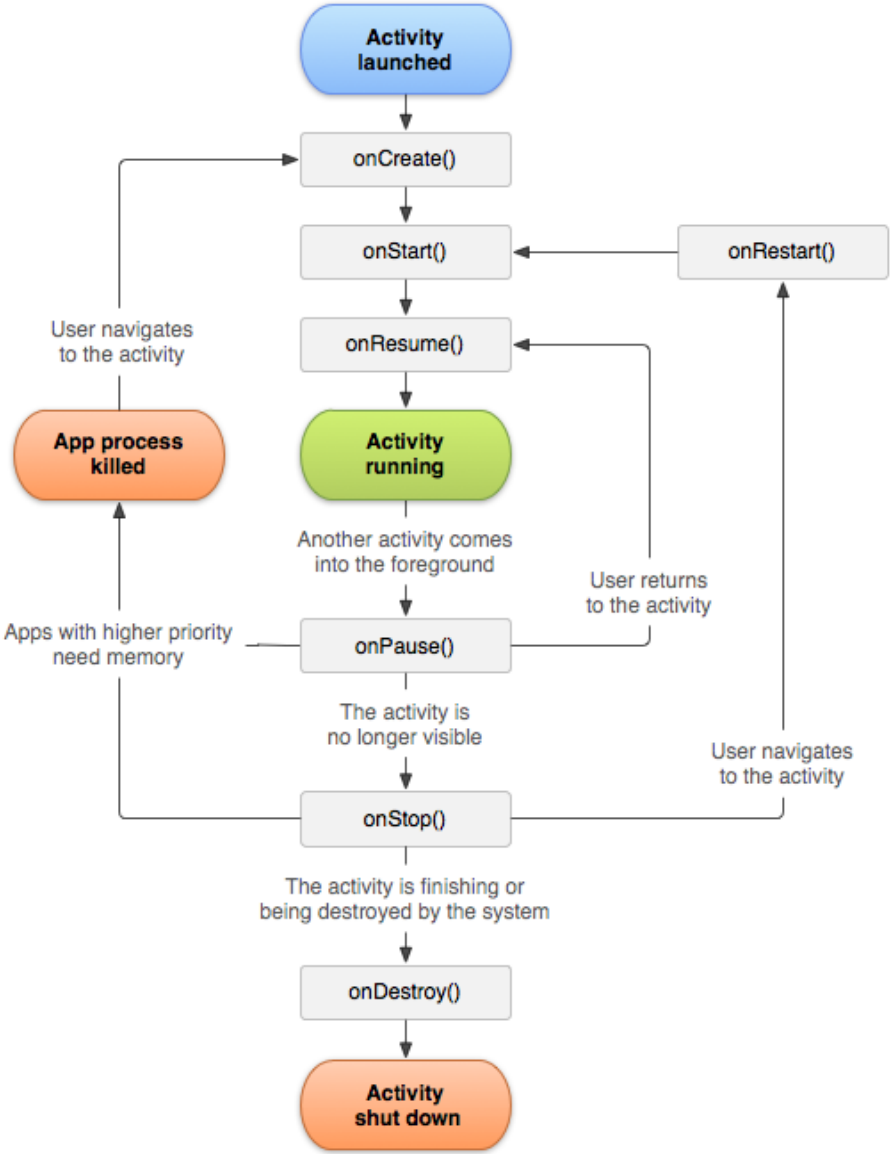
- onCreate() – to start the activity newly.

If the application has been stopped for other reasons and the user returns to the activity

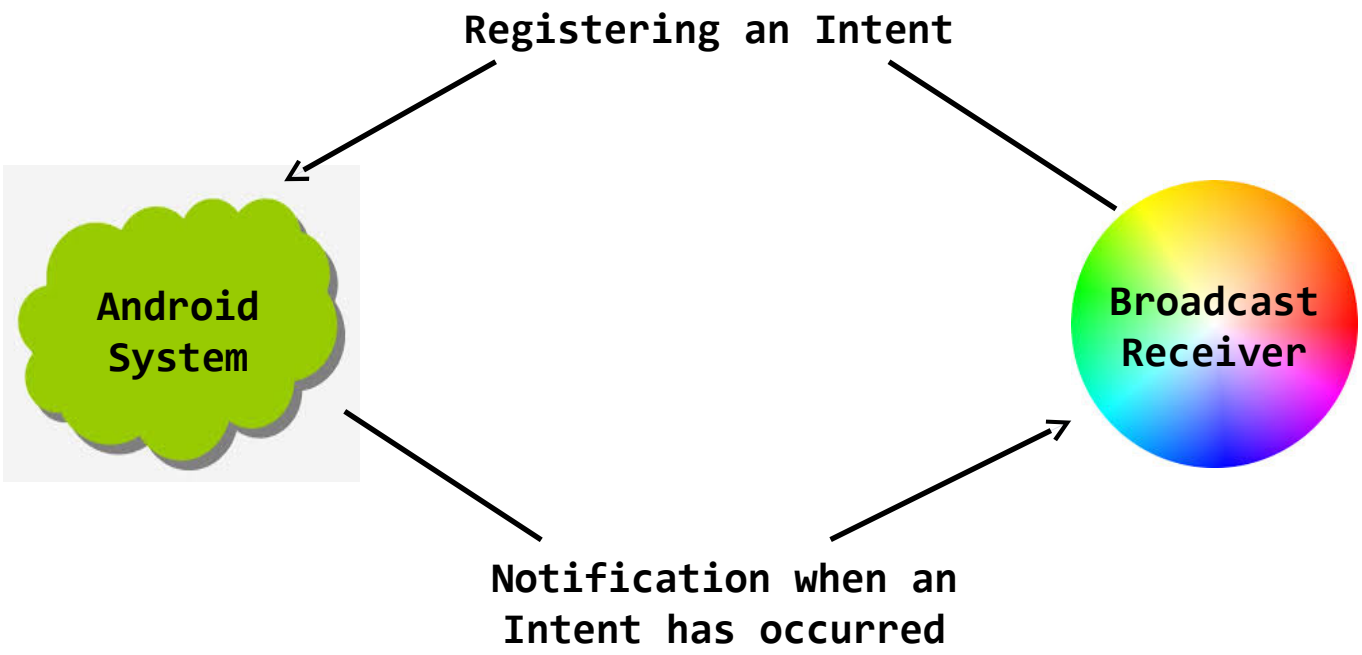
- onRestart() and then onStart() are called to restart the application

If the user finally wants to finish the activity:

- onDestroy()



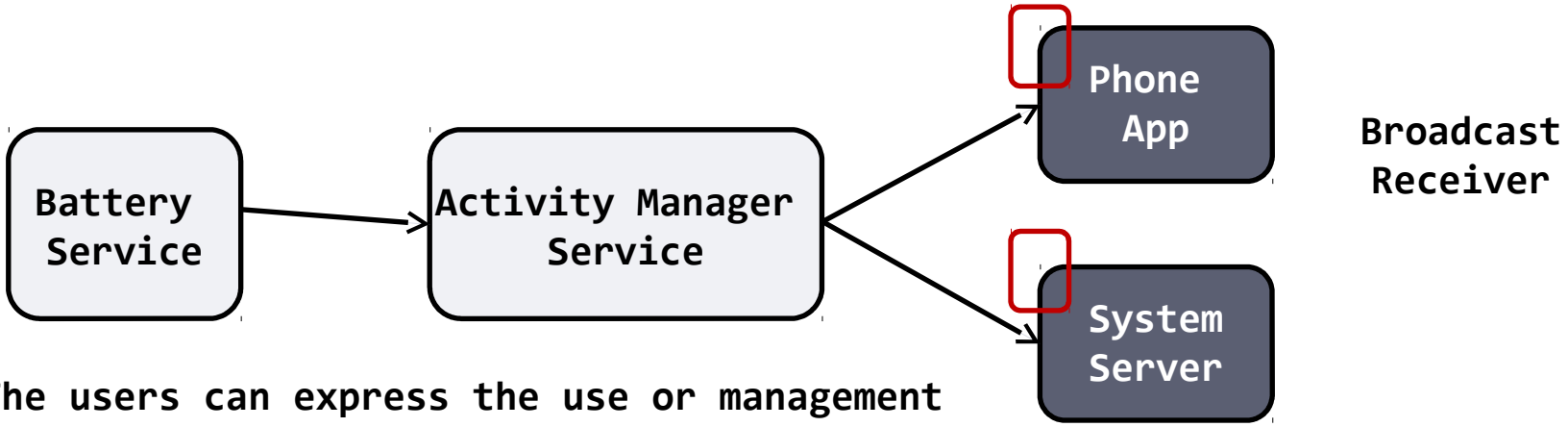
Broadcast Receivers



An Intent Subscribe-Notification mechanism
Used to listen for event that happens in the system

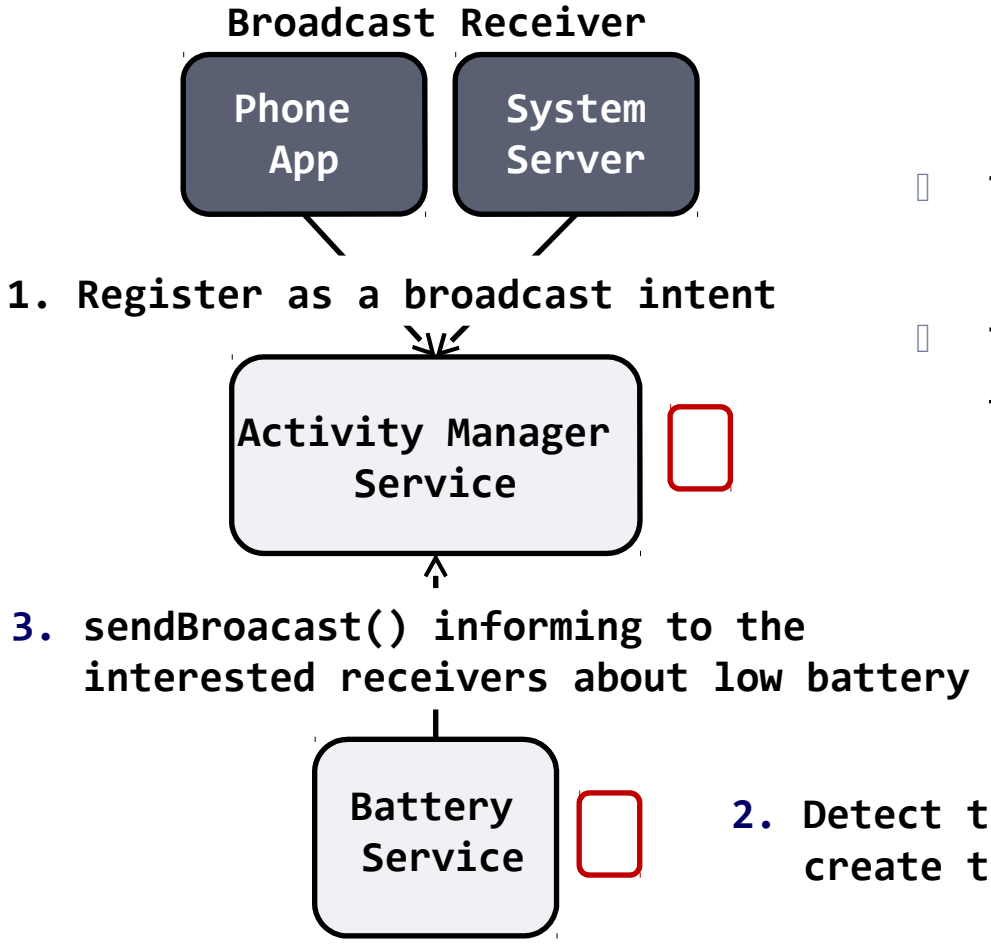
Broadcast Receivers

- Receiver (short) – allows to register actions/events from the system or from the applications. They are notified in real time once an event has occurred
- Are used to respond to the event message diffusion from all the system



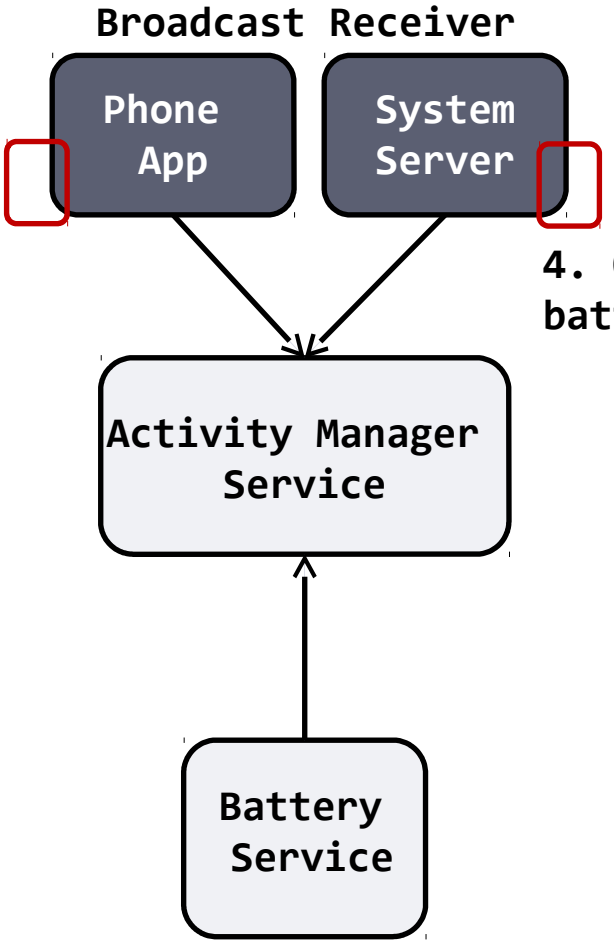
The users can express the use or management of petitions: dynamically or statically

Broadcast Receivers



- User creates a receiver to register events from the system or from other applications
- The Events are implemented as Intents
- The Events are disseminated through the whole system

Broadcast Receivers



- When an event occurs, an intent is sent to all interested receivers: we get the bar notification, etc.

Broadcast Receiver Implementation

1. extends `BroadcastReceiver`
2. We implement (override) `onReceive()`
3. Perhaps we have to implement other methods depending on the need and/or what has to be done
4. A Broadcast Receiver has to be declared within the AndroidManifest.xml if we want it visible

```
public class BRExample extends BroadcastReceiver {
     @Override
     public void onReceive(Context rcvCtx, Intent rcvIntent) {
        if (rcvIntent.getAction().equals(Intent.ACTION_CAMERA_BUTTON)) {
          rcvCtx.startService(new Intent(rcvCtx, SomeService.class));
          }
     }
} /** Fin de the class BRExample ***/
```

Broadcast Receiver Implementation

- Generated by the OS
  - Low battery
  - When the Camera button has been pressed
  - New app installed
  - WiFi – connection established
- Generated by the user
  - Starting some calculation
  - End of some operation

Broadcast Receiver Implementation

- The OS uses the Activity Manager in order to disseminate the intents to the receivers using some filter that apply some criteria
- Two forms to register a receiver:
  1. Statically: publishing it through the tag <receiver> in the AndroidManifest.xml file
  2. Dynamically via Context.registerReceiver() through its own class at real time

Broadcast Intents

- System Services:

ACTION_AIRPLANE_MODE_CHANGED	Broadcast Action: The user has switched the phone into or out of Airplane Mode.
ACTION_APPLICATION_RESTRICTIONS_CH	Broadcast Action: Sent after application restrictions are changed.
ACTION_APP_ERROR	Activity Action: The user pressed the "Report" button in the crash/ANR dialog.
ACTION_BATTERY_CHANGED	Broadcast Action: This is a sticky broadcast containing the charging state, level, and other information about the battery.
ACTION_BATTERY_LOW	Broadcast Action: Indicates low battery condition on the device.
ACTION_BATTERY_OKAY	Broadcast Action: Indicates the battery is now okay after being low.
ACTION_BOOT_COMPLETED	Broadcast Action: This is broadcast once, after the system has finished booting.
ACTION_BUG_REPORT	Activity Action: Show activity for reporting a bug.
ACTION_CALL	Activity Action: Perform a call to someone specified by the data.
ACTION_CALL_BUTTON	Activity Action: The user pressed the "call" button to go to the dialer or other appropriate UI for placing a call.
ACTION_CAMERA_BUTTON	Broadcast Action: The "Camera Button" was pressed.

- They can be originated from application components:
  - e.g. image file download petition completed
  - A document has been opened etc.

Services

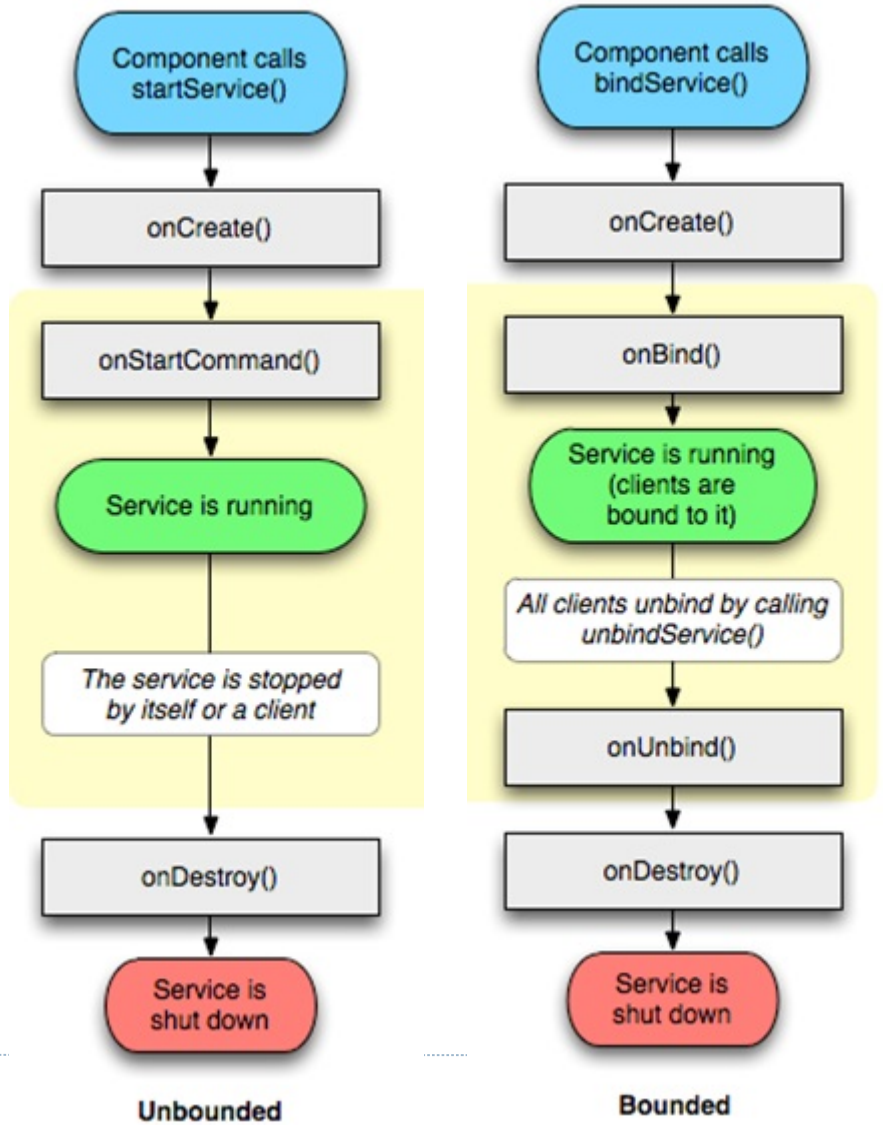
- It is an application that is executed in the background, when there is no need for user interaction
  - It continues even when the activity that has launched it is destroyed
  - It terminates when it is disconnected from all applications
  - It is used when something that has to be done/executed while the user has no interaction with the application, otherwise threads are used
- Allows multiple applications to communicate through a common interface
- It has to be declared in the “manisfest.xml”
- Similar to the services in Windows or daemon in Unix No UI: check events. e.g.: new email etc.
- It has its own life cycle structure

Services

The life cycle of a Service is much simpler than that of an activity

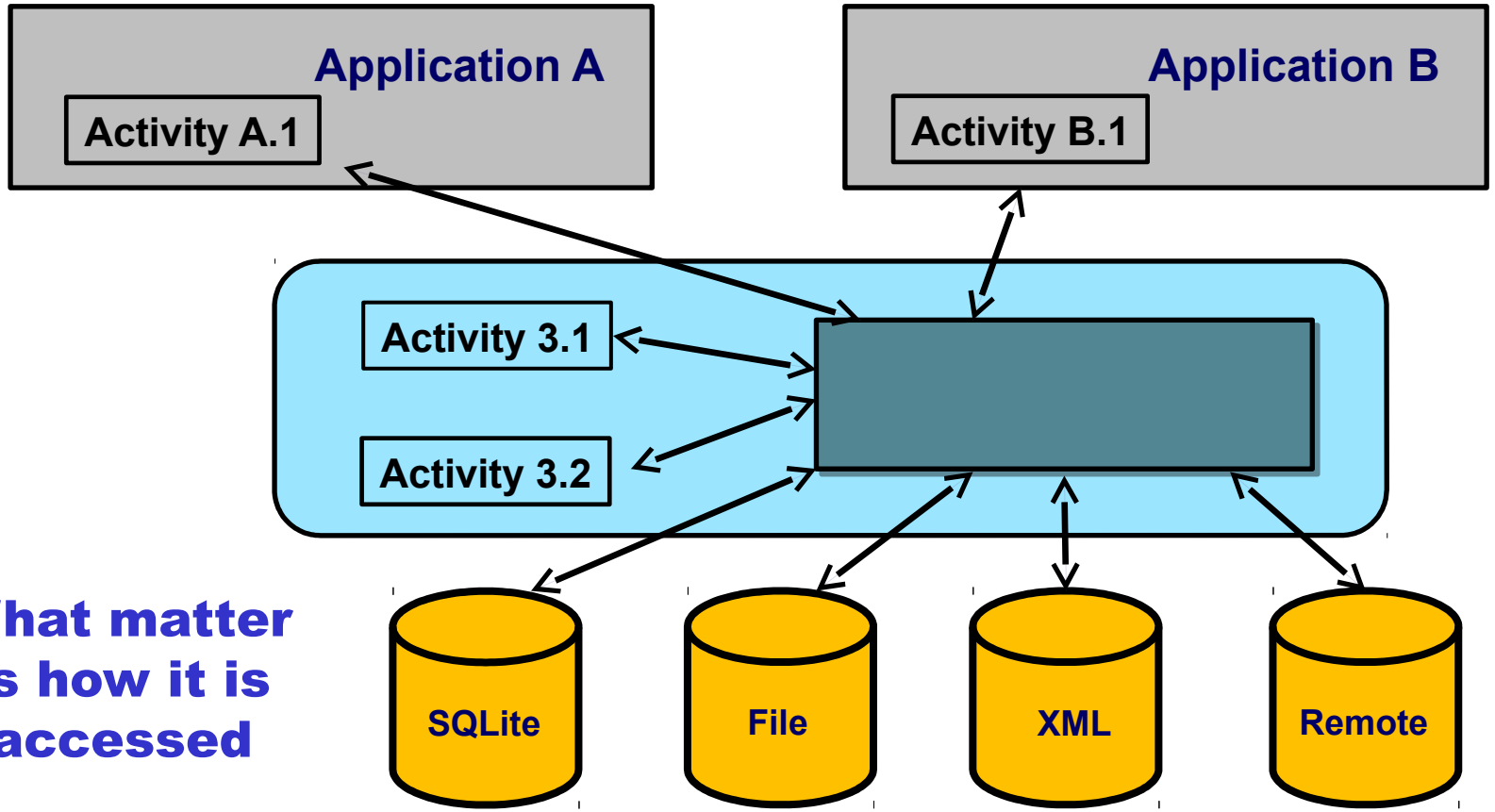
An activity normally starts and stops a service to execute some action behind the scenes: play music, see the new tweets, etc.

Services can be bounded or unbounded



Content Providers

- Used primarily for data sharing between packages, regardless of how they are saved



What matter is how it is accessed

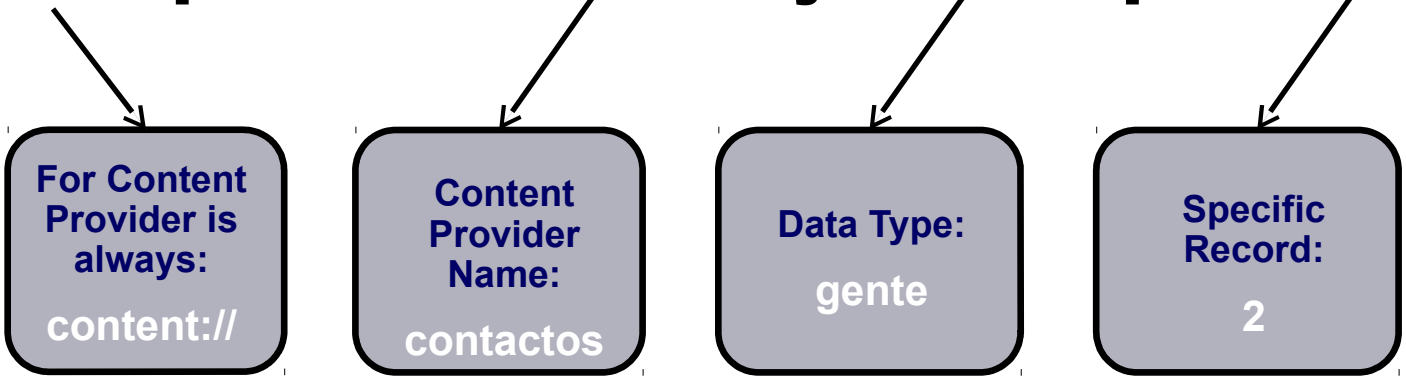
Content Providers

- CP's behaves like databases:
  - Do queries
  - Edit content
  - Add record
  - Delete record
- Data can be stored as files , in a network or in a DB
- Android has some built-in CP's:
  - Browser – store bookmarks, browse history
  - CallLog – store the lost calls
  - Contacts – store the contacts info
  - Media store – store different media files
  - Settings – store configuration and preferences

Content Providers

- To do a query to a CP you have to specify the query string as a URI :

<standard_prefix>://<authority>/<data_path>/<id>



content://contactos/gente/2

URI – Uniform Resource Identifier
Authority – com.SDM.setiapp
data_path – specify the queried data type
id – specify a defined record

uc3m | Universidad Carlos III de Madrid

Grupo de investigación:
Computer Security Lab

Mobile Devices Security

Bachelor Degree in Informatics Engineering

2019