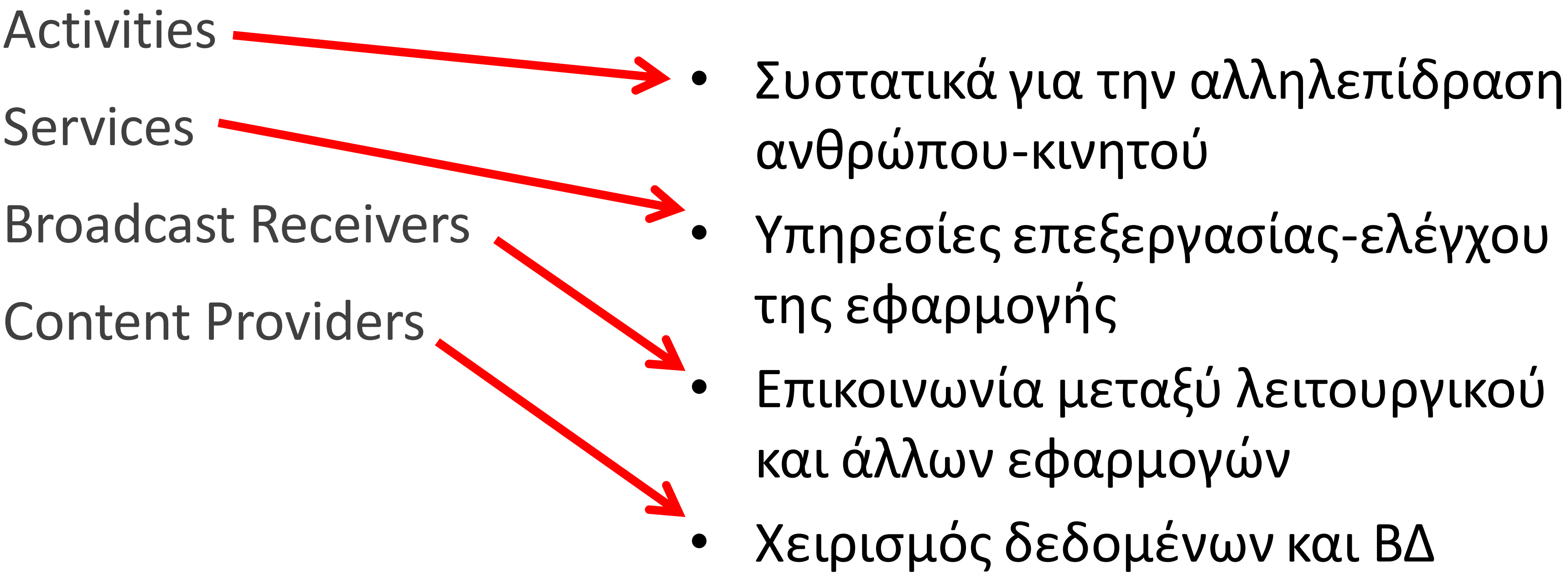
Android Programming

Basic Android Components



Activities

Services

Broadcast Receivers

Content Providers

- Συστατικά για την αλληλεπίδραση ανθρώπου-κινητού
- Υπηρεσίες επεξεργασίας-ελέγχου της εφαρμογής
- Επικοινωνία μεταξύ λειτουργικού και άλλων εφαρμογών
- Χειρισμός δεδομένων και ΒΔ

+ resources

Activities

```
public class MainActivity extends Activity { }
```

Services

```
public class MyService extends Service { }
```

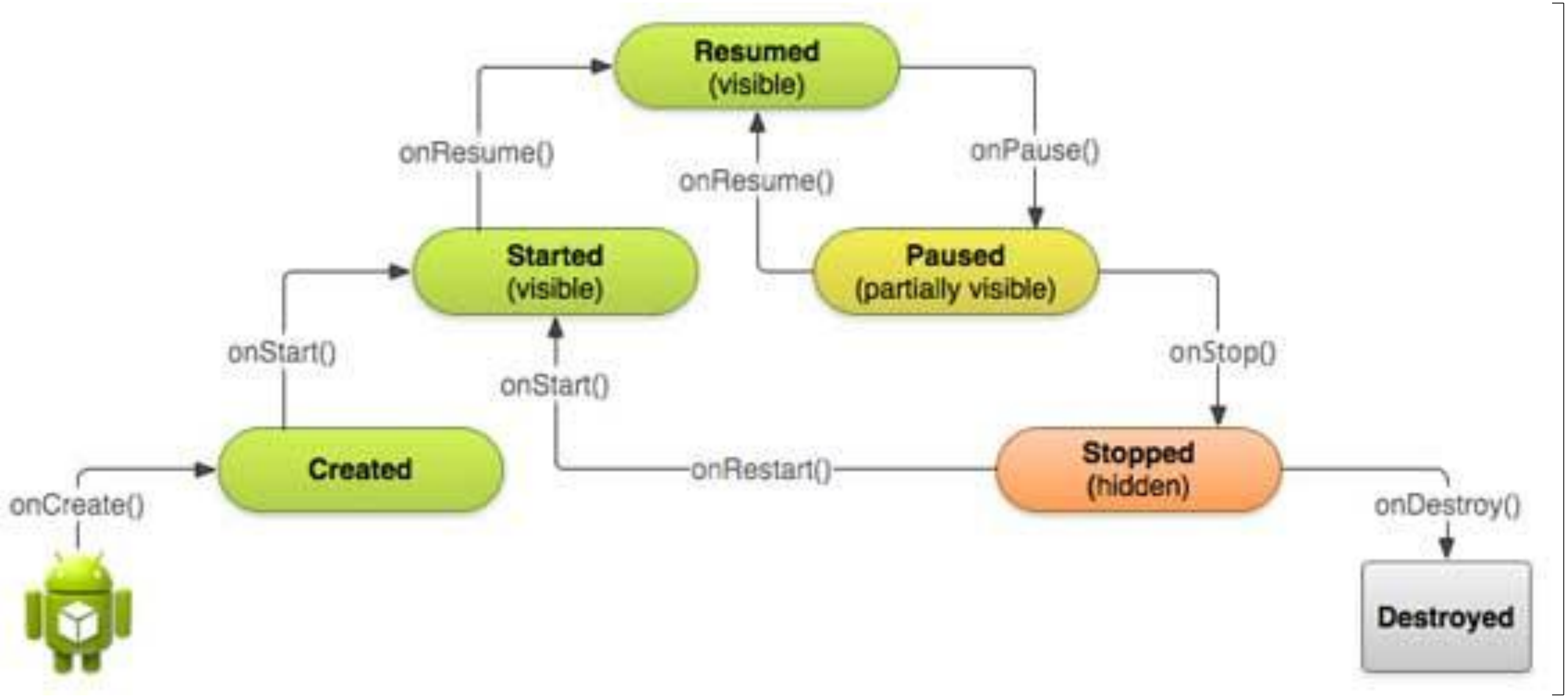
Broadcast Receivers

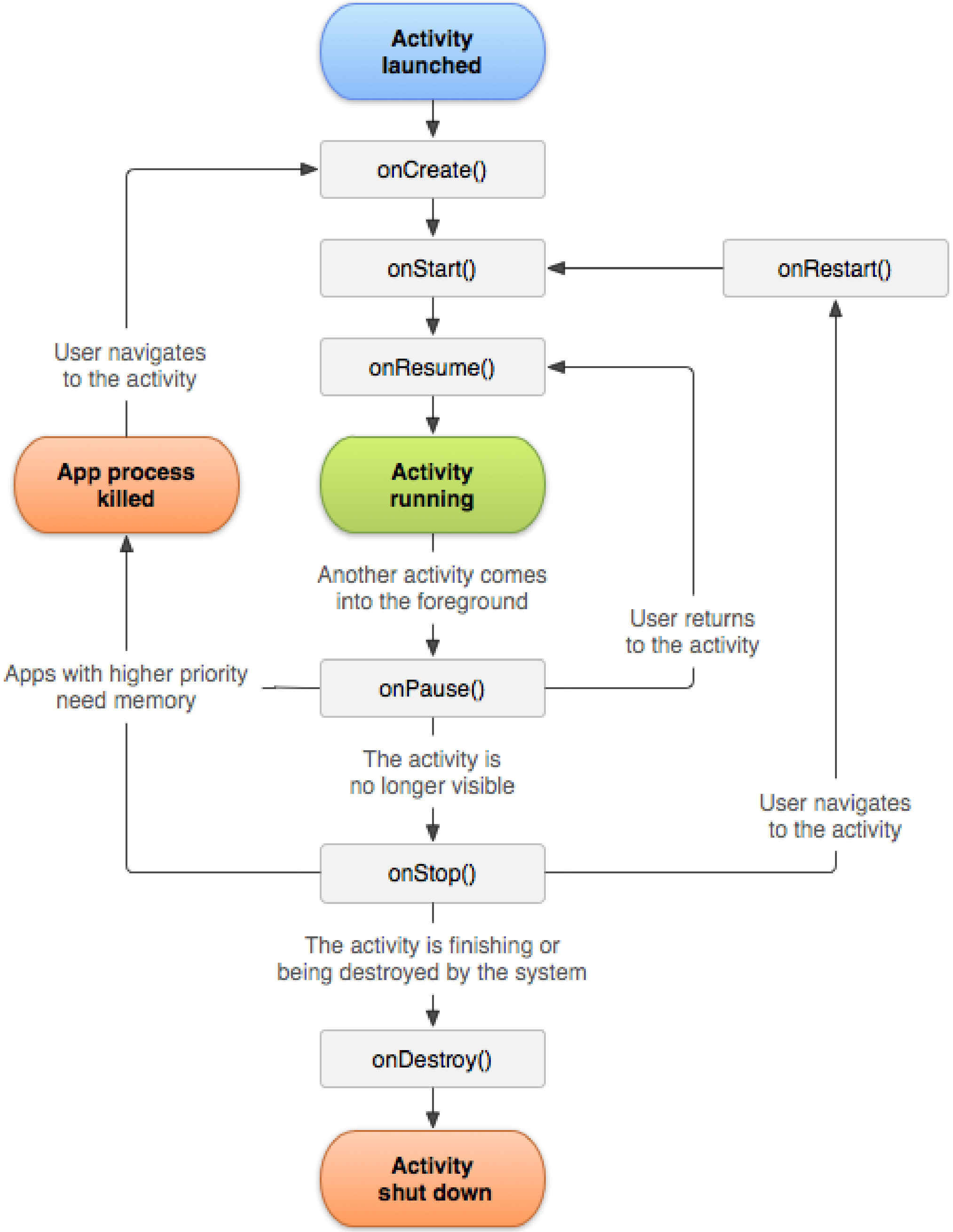
```
public class MyReceiver extends BroadcastReceiver { }
```

Content Providers

```
public class MyContentProvider extends
ContentProvider { }
```

Activity Lifecycle





Activity launched

onCreate()

onStart()

onRestart()

User navigates to the activity

onResume()

App process killed

Activity running

Another activity comes into the foreground

User returns to the activity

Apps with higher priority need memory

onPause()

The activity is no longer visible

User navigates to the activity

onStop()

The activity is finishing or being destroyed by the system

onDestroy()

Activity shut down

onCreate()

Called when the activity is first created. This is where you should do all of your normal static set up: create views, bind data to lists, etc. This method also provides you with a Bundle containing the activity's previously frozen state, if there was one. Always followed by onStart().

onStart()

lled when the activity is becoming visible to the user. Followed by onResume() if the activity comes to the foreground, or onStop() if it becomes hidden.

onResume()

Called when the activity will start interacting with the user. At this point your activity is at the top of the activity stack, with user input going to it. Always followed by onPause().

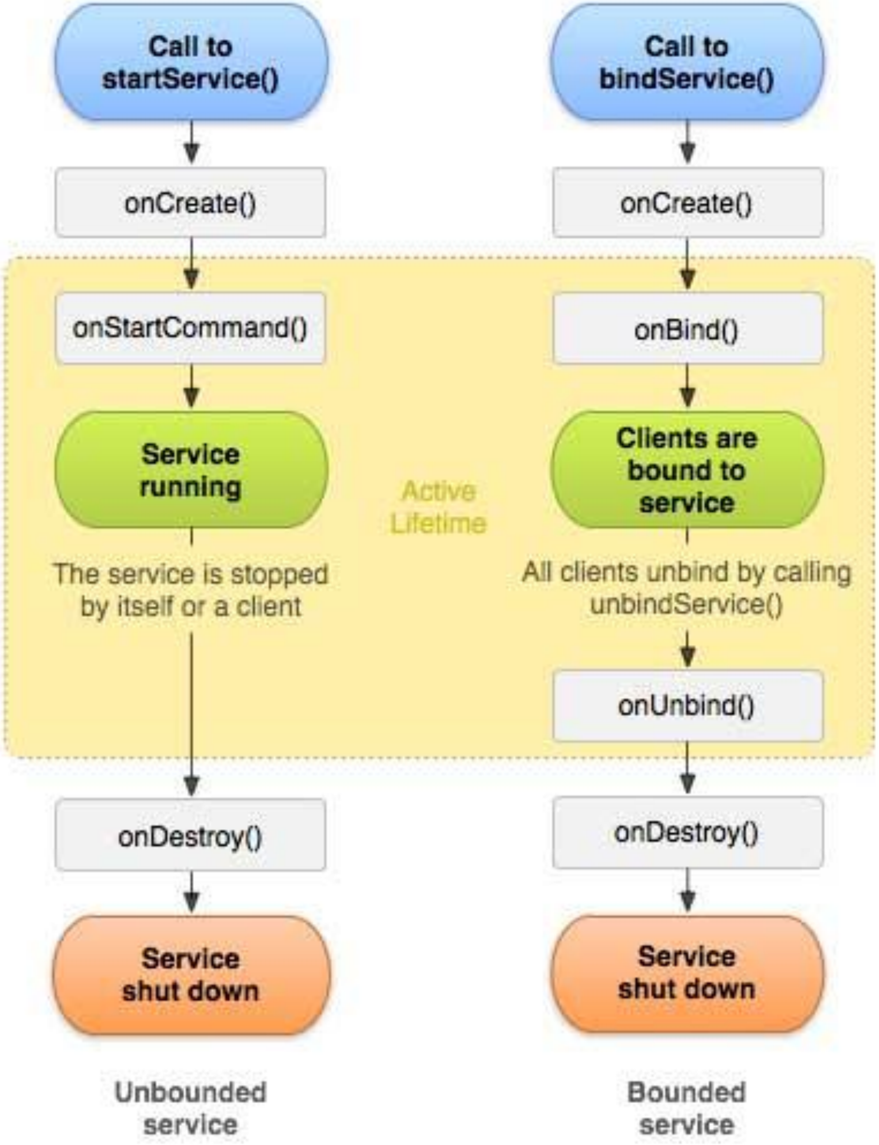
onPause()

Called when the system is about to start resuming a previous activity. This is typically used to commit unsaved changes to persistent data, stop animations and other things that may be consuming CPU, etc. Implementations of this method must be very quick because the next activity will not be resumed until this method returns. Followed by either onResume() if the activity returns back to the front, or onStop() if it becomes invisible to the user.

onDestroy()

The final call you receive before your activity is destroyed. This can happen either because the activity is finishing (someone called finish() on it, or because the system is temporarily destroying this instance of the activity to save space. You can distinguish between these two scenarios with the isFinishing() method.

Service Lifecycle



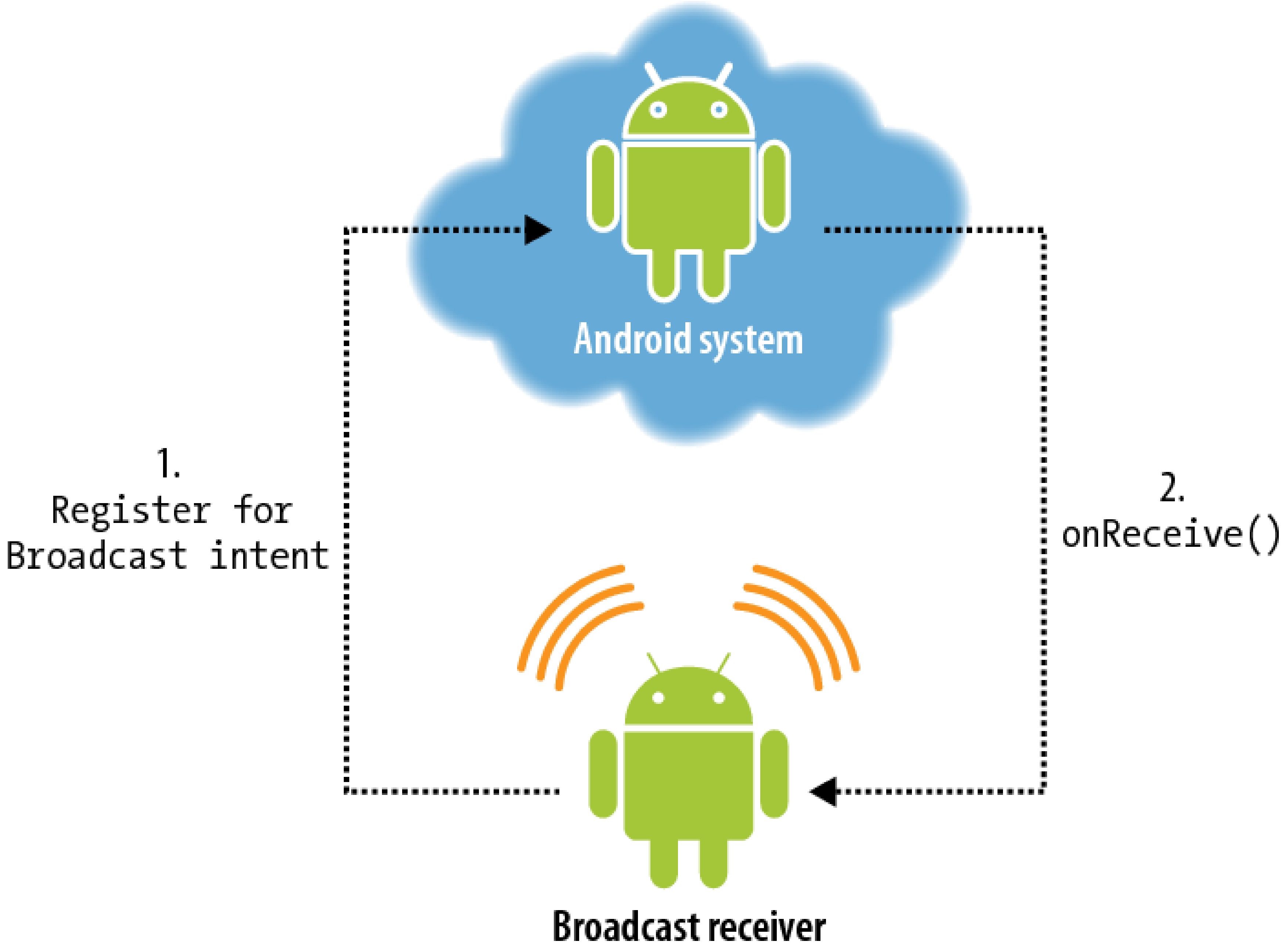
Creating a Broadcast Receiver

"Broadcast Receivers respond to broadcast messages from other applications or from the system itself"

```
public class MyReceiver extends BroadcastReceiver {

  @Override
  public void onReceive(Context context, Intent intent) {
    Toast.makeText(context, "Intent Detected.", Toast.LENGTH_LONG).show();
  }

}
```

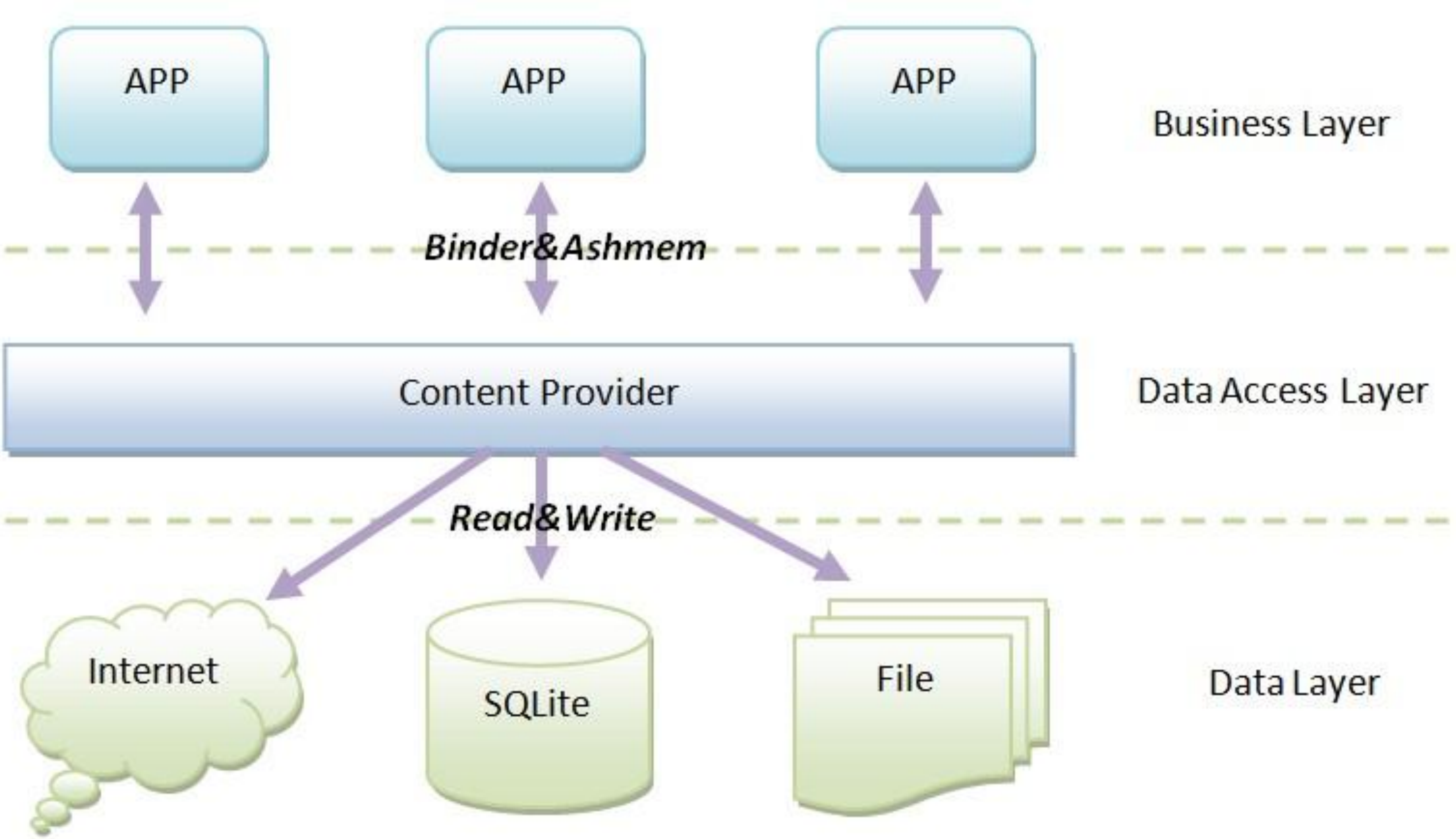


Android system

1.
Register for
Broadcast intent

2.
onReceive()

Broadcast receiver


APP
APP
APP
Business Layer
Binder&Ashmem
Content Provider
Data Access Layer
Read&Write
Internet
SQLite
File
Data Layer

Android Content Providers

A content provider component supplies data from one application to others on request. In most cases this data is stored in an SQlite database.

```
public class MyContentProvider extends  ContentProvider {


}
```

Android Events

Event Listeners

Event Handlers

Event Listeners Registration

What is the difference?

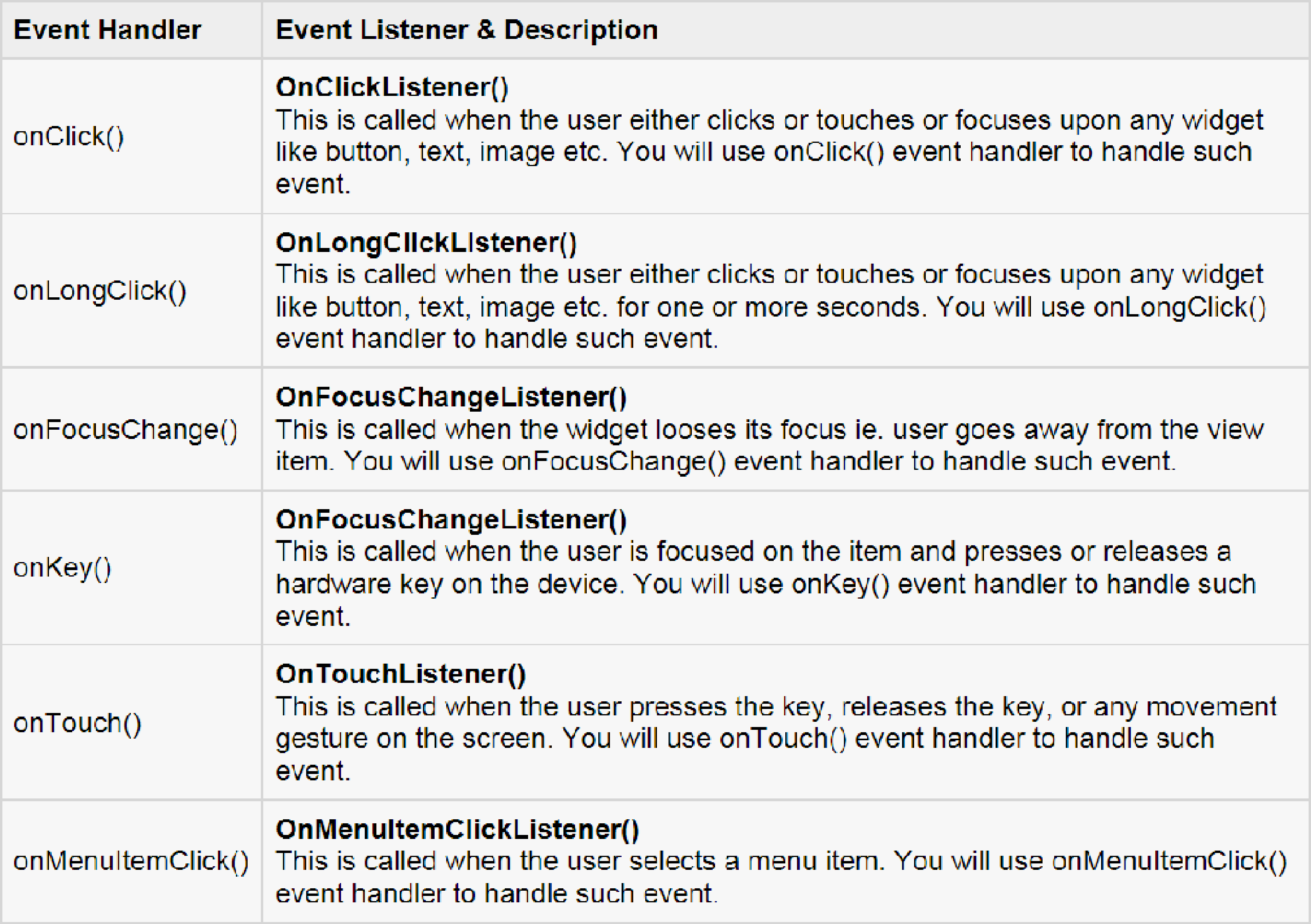
The listener is the object (or the mechanism) that receives notification

- A Listener is associated with Event Source

The handler is the method that actually handles the notification

- A Handler is associated with an Event

Event Registration is the process by which an Event Handler gets registered with an Event Listener so that the handler is called when the Event Listener fires the event

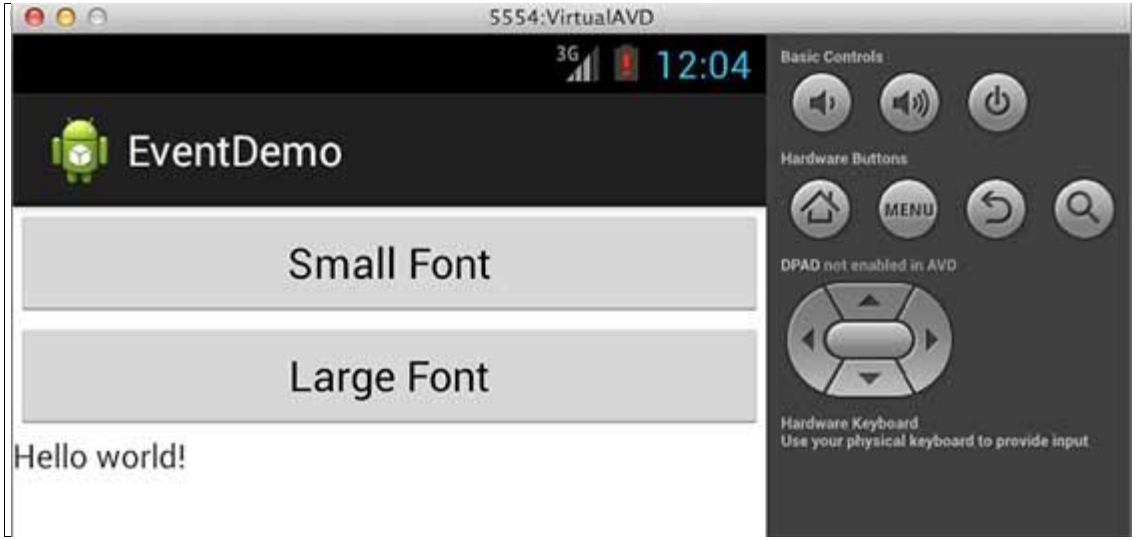


Event Handler	Event Listener & Description
onClick()	**OnClickListener()** This is called when the user either clicks or touches or focuses upon any widget like button, text, image etc. You will use onClick() event handler to handle such event.
onLongClick()	**OnLongClickListener()** This is called when the user either clicks or touches or focuses upon any widget like button, text, image etc. for one or more seconds. You will use onLongClick() event handler to handle such event.
onFocusChange()	**OnFocusChangeListener()** This is called when the widget looses its focus ie. user goes away from the view item. You will use onFocusChange() event handler to handle such event.
onKey()	**OnFocusChangeListener()** This is called when the user is focused on the item and presses or releases a hardware key on the device. You will use onKey() event handler to handle such event.
onTouch()	**OnTouchListener()** This is called when the user presses the key, releases the key, or any movement gesture on the screen. You will use onTouch() event handler to handle such event.
onMenuItemClick()	**OnMenuItemClickListener()** This is called when the user selects a menu item. You will use onMenuItemClick() event handler to handle such event.

Event Listeners Registration

- Using an Anonymous Inner Class (file events1.java)
- Using an Activity Class that implements the OnClickListener interface (file events2.java)
- Using Layout file activity_main.xml to specify event handler directly (file events3.java)

Demo



Hello world!

Drag and Drop Process

Η διαδικασία Drag and Drop στο Android αποτελείται από 4 βασικά στάδια:

- Στάδιο γεγονότος «Started»
- Στάδιο γεγονότος «Continuing»
- Στάδιο γεγονότος «Dropped»
- Στάδιο γεγονότος «Ended»

Started event

Το γεγονός αυτό πυροδοτείται όταν ξεκινάμε να «σύρουμε» ένα αντικείμενο μέσα σε ένα layout

Κατά το γεγονός αυτό, η εφαρμογή καλεί τη μέθοδο startDrag()

Εν συνεχεία το σύστημα προβάλει μια σκιά στη συσκευή που αντιστοιχεί στο «συρόμενο» αντικείμενο

Continuing event

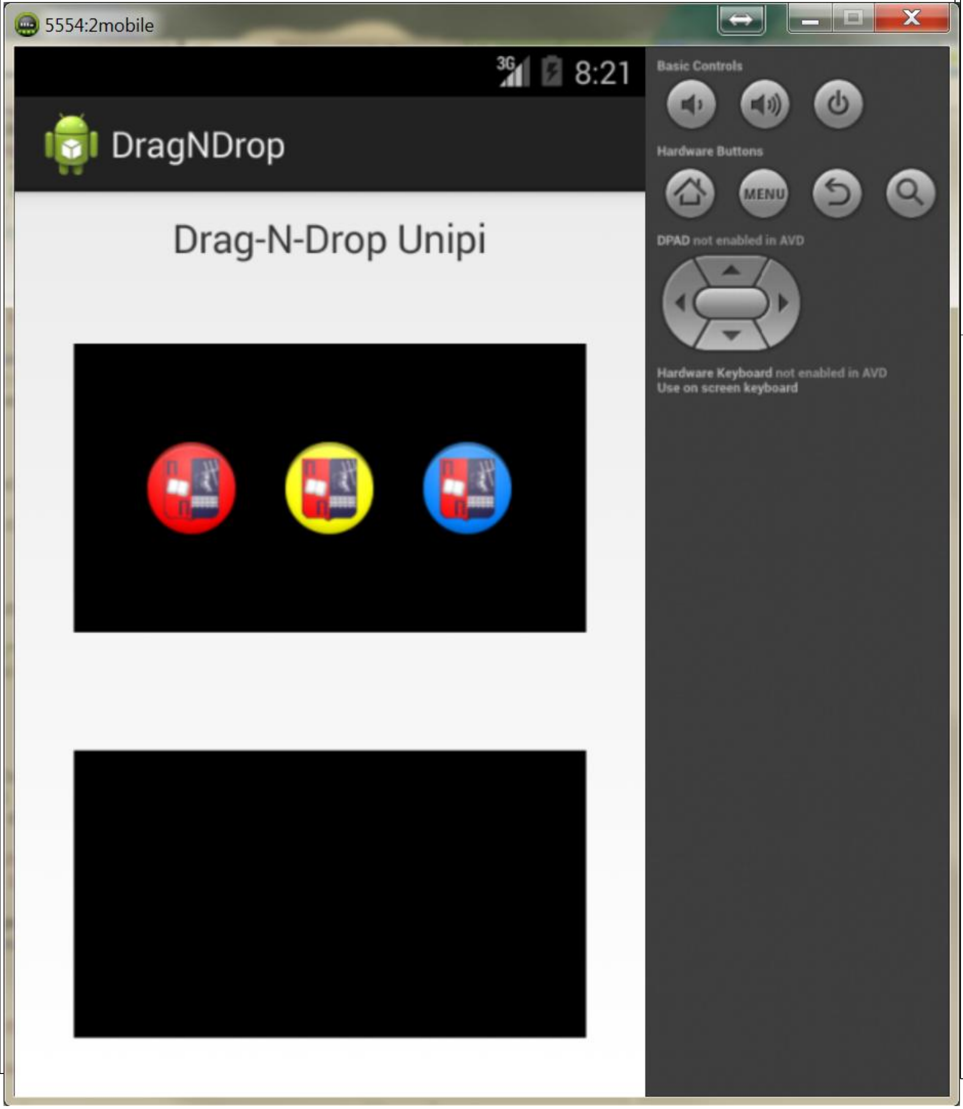
Μετά την εκκίνηση του γεγονότος «Started» και εφόσον ο χρήστης συνεχίζει τη διαδικασία, έχουμε το continuing event

Το σύστημα στέλνει την ενέργεια ACTION_DRAG_ENTERED και εν συνεχεία την ενέργεια ACTION_DRAG_LOCATION

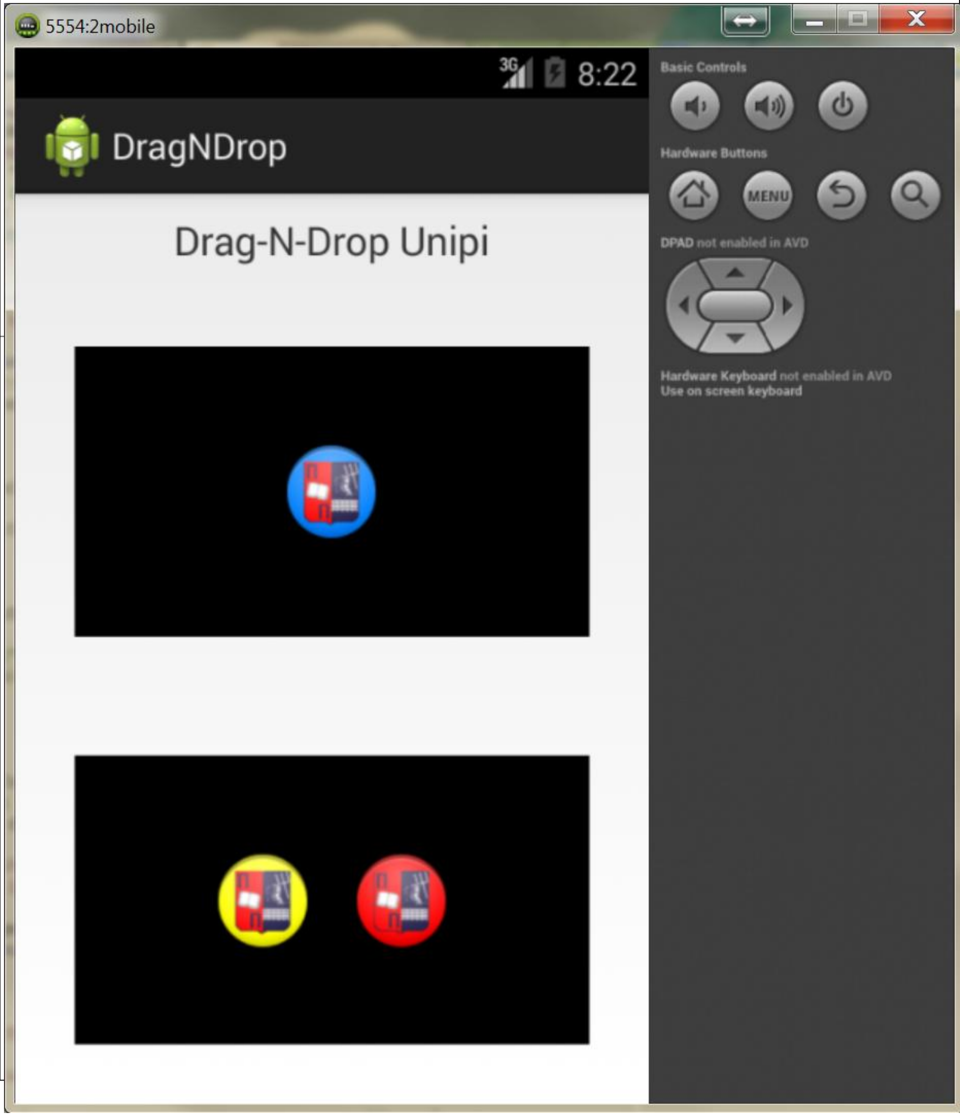
Dropped event/Ended event

- Έχουμε το γεγονός dropped μόλις ο χρήστης απελευθερώσει το συρόμενο αντικείμενο μέσα στα περιθώρια ενός View
- Το σύστημα ενημερώνει για το εν λόγω event ενεργοποιώντας την ενέργεια ACTION_DROP
- Αμέσως μετά το γεγονός dropped το σύστημα πυροδοτεί το γεγονός ended με την ενέργεια ACTION_DRAG_ENDED

The image part with relationship ID rId2 was not found in the file.



The image part with relationship ID rId2 was not found in the file.


5554:2mobile
8:22
DragNDrop
Drag-N-Drop Unipi
Basic Controls
Hardware Buttons
MENU
DPAD not enabled in AVD
Hardware Keyboard not enabled in AVD
Use on screen keyboard