

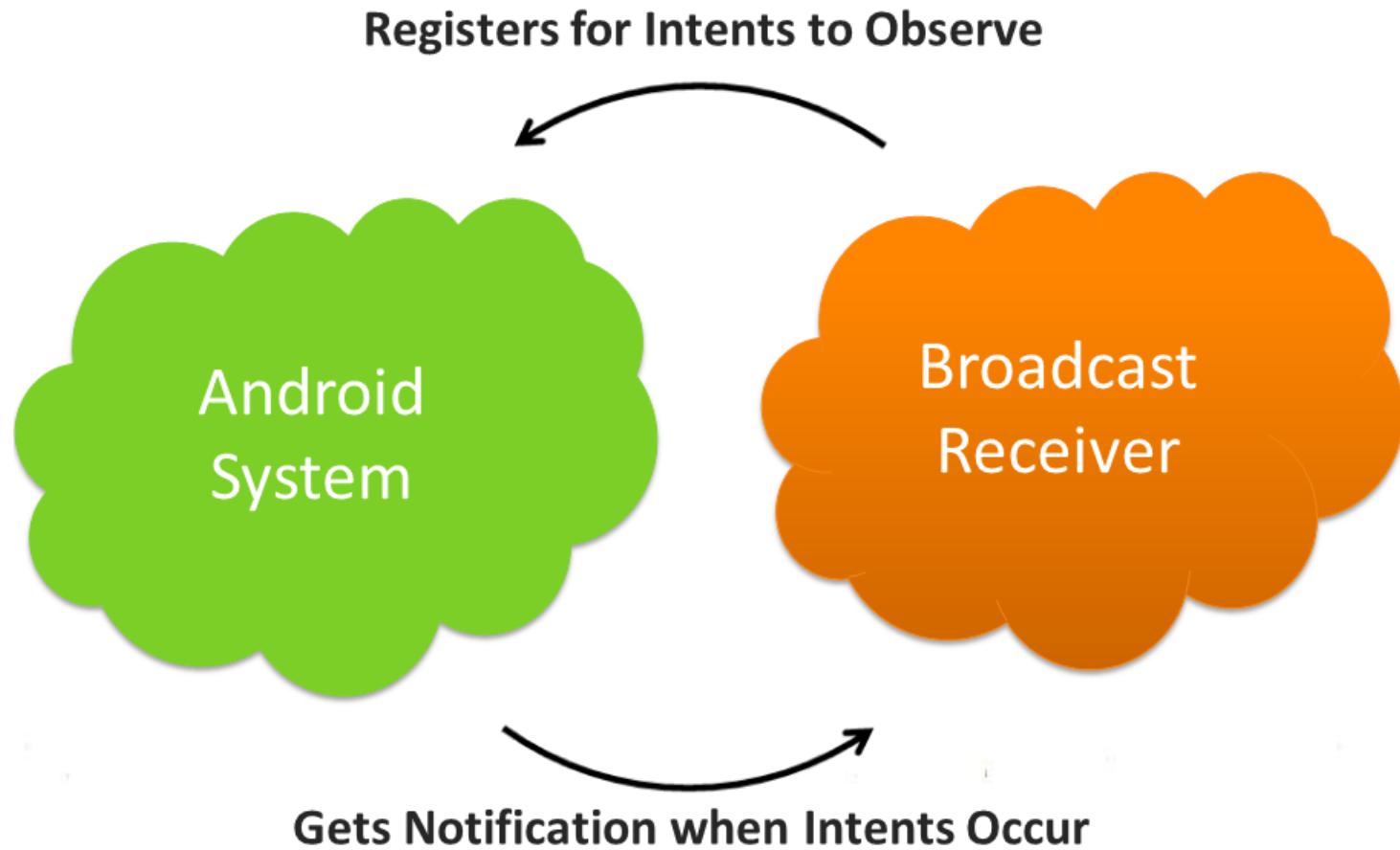


Broadcast Receivers



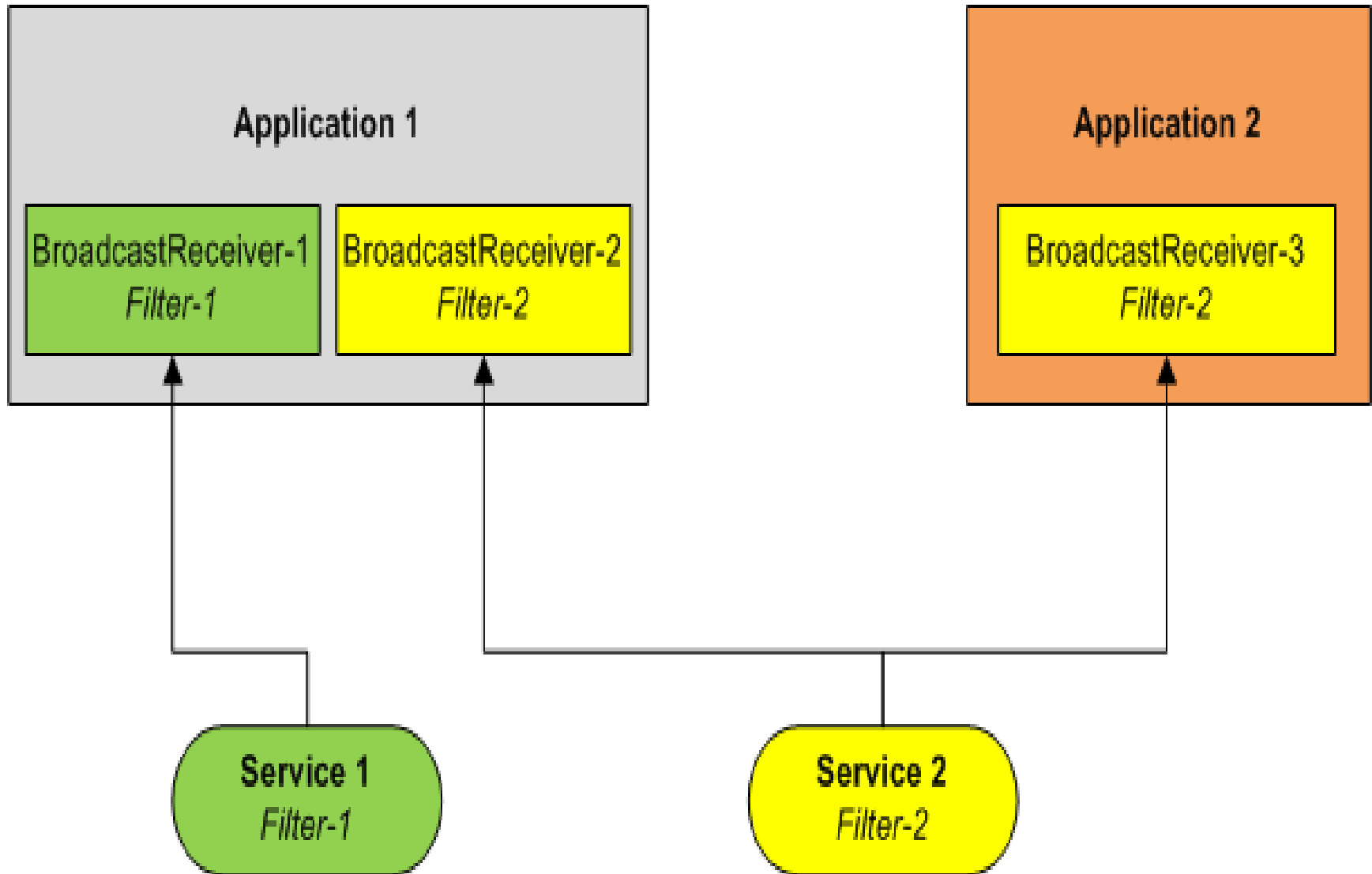
Sisoft Technologies Pvt Ltd
SRC E7, Shipra Riviera Bazar, Gyan Khand-3, Indirapuram, Ghaziabad
Website: www.sisoft.in Email: info@sisoft.in
Phone: +91-9999-283-283

Broadcast Receiver



Broadcast Receiver

- A Broadcast Receiver is an application class that is a dormant component of Android System.
- It registers for specific event using Intent filter action
- Once the event occurs, the system delivers a broadcast Intent to all interested broadcast receivers, which handle the Intent *sequentially*



Registering a Broadcast Receiver

- Static: Use `<receiver>` tag in your `AndroidManifest.xml`
 - In addition an `<intent-filter>` entry is needed to declare the actual filter the service and the receiver use.
- Dynamic: Use ***Context.registerReceiver()*** method to dynamically register an instance of the class



BroadcastReceivers in AndroidManifest

```
<receiver android:name="SisoftReceiver">  
  <intent-filter>  
    <action  
      android:name="android.intent.action.BOOT_C  
      OMPLETED">  
    </action>  
  </intent-filter>  
</receiver>
```

Broadcast Receiver Lifecycle

- A broadcast receiver has a single callback method:**void onReceive()**
- When a broadcast message arrives for the receiver, Android calls its onReceive()method and passes it the Intent object containing the message.
- The broadcast receiver is considered to be active only while it is executing this method.
- When onReceive()returns, it is inactive.

Send Broadcasts

- There are two major classes of broadcasts that can be sent:
 - 1. Normal broadcasts**(sent with `Context.sendBroadcast`) are completely asynchronous. All receivers of the broadcast are run in an undefined order, often at the same time.
 - 2. Ordered broadcasts**(sent with `Context.sendOrderedBroadcast`) are delivered to one receiver at a time. As each receiver executes in turn, it can propagate a result to the next receiver, or it can completely abort the broadcast so that it won't be passed to other receivers. The order receivers run in can be controlled with the `android:priority` attribute of the matching intent-filter; receivers with the same priority will be run in an arbitrary order.