

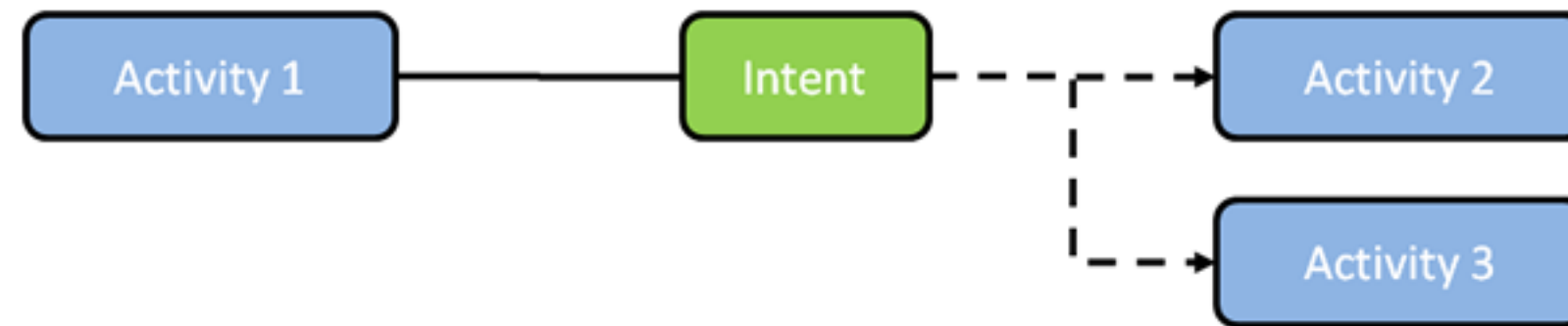


ANDROID

LEZIONE 3 - 26/04/2017

<https://github.com/giuseppegrosso/android.git>

- ▶ Activity. Componenti che sovrintendono alla gestione della UI e controllano l'interazione con l'utente.
- ▶ Fragment. Componenti che consentono un'approccio modulare nella realizzazione e gestione di interfacce utente.
- ▶ **Intent**
- ▶ BroadcastReceiver
- ▶ Service
- ▶ ContentProvider



- ▶ Un Intent è un oggetto il cui scopo principale è quello di veicolare un messaggio fra due componenti del sistema
- ▶ Esistono Intent espliciti ed impliciti
- ▶ Un Intent può avere extra, intesi come oggetti che implementano l'interfaccia Parcelable

INTENT e INTENT FILTER

- ▶ Un Intent è definito in modo più generale come un oggetto il cui scopo è richiedere ad una componente terza di eseguire una operazione.
- ▶ Gli use case principali sono 3:
 - ▶ Avviare una **Activity**, attraverso il metodo *startActivity()* o il metodo *startActivityForResult()*
 - ▶ Avviare un **Service** tramite *startService()* o *bindService()*.
 - ▶ Notificare un evento a un **BroadcastReceiver** attraverso i metodi *sendBroadcast()*, *sendOrderedBroadcast()*, *sendStickyBroadcast()*

► Esistono due tipi di Intent, espliciti e impliciti:

► Un intent **esplicito** indica il componente che si desidera venga avviato in maniera...esplicita. Use case tipici sono avviare una Activity della propria app in risposta ad un input utente o un Service al verificarsi di un evento.

// intent to home

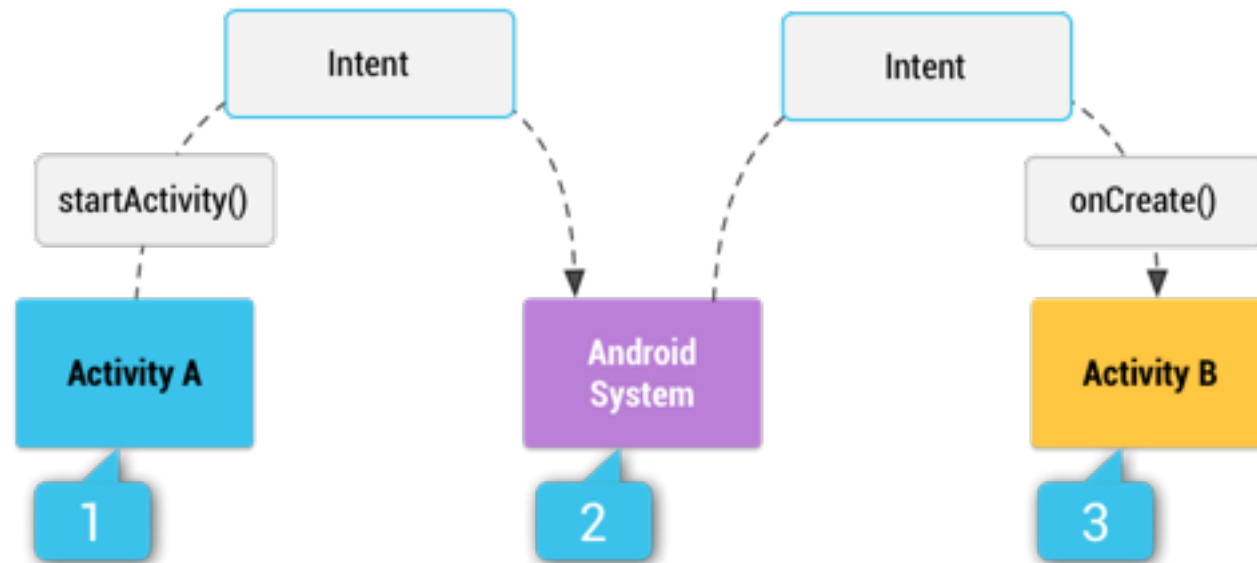
```
Intent i = new Intent(this, HomeActivity.class);  
startActivity(i);
```

► Un intent **implicito** non indica una componente specifica ma una azione. Esempi tipici sono la condivisione social, la visualizzazione della posizione, l'invio di una mail, etc.

// this runs, for example, after a button click

```
Intent intent = new Intent(Intent.ACTION_SEND);  
intent.setType("text/plain");  
intent.putExtra(android.content.Intent.EXTRA_TEXT, "News for you!");  
startActivity(intent);
```

► La propagazione di un Intent implicito avviene come segue.



Quando un intent implicito viene creato il sistema utilizza gli *Intent Filter* dichiarati da ciascuna delle *app* presenti sul device per vedere quali sono in grado di gestire la specifica action.

<http://developer.android.com/guide/components/intents-filters.html>

Se c'è più di un match il sistema mostra un dialog all'utente che consente di selezionare l'applicazione (es. ACTION_SEND mostra Whatsapp, Facebook, Telegram, etc).

Una volta selezionata l'applicazione il sistema avvia l'activity dichiarata nel filter passando l'Intent al suo metodo *onCreate()*.

- ▶ Costruire un Intent richiede di conoscere le componenti che è possibile configurare.
- ▶ **Component Name**, opzionale permette di distinguere fra Intent impliciti ed espliciti
- ▶ **Action**, azione da eseguire (opzionale)
- ▶ **Data**, dati su cui operare (in relazione all'azione se presente)
- ▶ **Type**, MIME type dei dati se presenti
- ▶ **Category**, informazioni aggiuntive sul tipo di componente a cui l'Intent è destinato (es. CATEGORY_LAUNCHER)
- ▶ **Extras**, payload custom legato allo scenario operativo
- ▶ **Flags**, indicano al sistema operativo come avviare il componente a cui l'Intent è destinato.

- ▶ Activity. Componenti che sovrintendono alla gestione della UI e controllano l'interazione con l'utente.
- ▶ Fragment. Componenti che consentono un'approccio modulare nella realizzazione e gestione di interfacce utente.
- ▶ Intent. Componenti che modellano in modo flessibile la comunicazione e le richieste di task fra componenti del sistema.
- ▶ **BroadcastReceiver**
- ▶ Service
- ▶ ContentProvider

- ▶ Un BroadcastReceiver è un componente che permette di *ricevere* in modo asincrono notifiche dal sistema (anche per conto di altre componenti) circa il verificarsi di particolari eventi.
- ▶ Un register può essere registrato attraverso AndroidManifest.xml

```
<application  
    <receiver android:name="MyReceiver">  
        <intent-filter>  
            <action android:name="android.intent.action.BOOT_COMPLETED">  
            </action>  
        </intent-filter>  
    </receiver>  
</application>
```

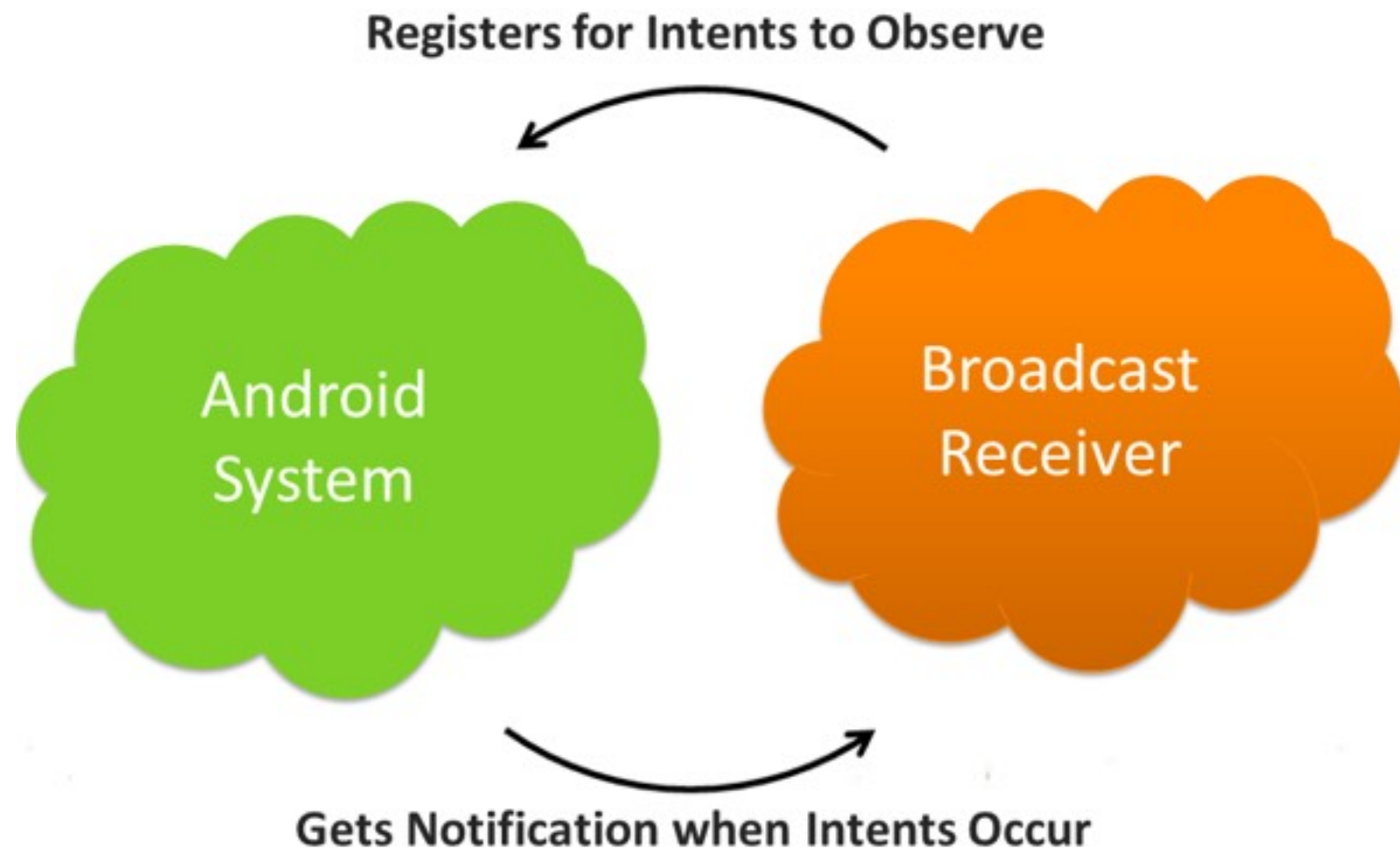
- ▶ ...oppure registrato (e deregistrato) programmaticamente a *runtime*

► ...oppure registrato ...

```
private final BroadcastReceiver myReceiver = new BroadcastReceiver() {  
  
    @Override  
    public void onReceive(Context c, Intent intent)  
    {  
  
        // handle intent  
    }  
  
};  
  
registerReceiver(myReceiver, new IntentFilter(WifiManager.SCAN_RESULTS_AVAILABLE_ACTION));
```

► ... (e deregistrato) programmaticamente a *runtime*

```
unregisterReceiver(myReceiver);
```



Quando si verifica un evento per il quale esistono `BroadcastReceiver` registrati il sistema li istanzia e invoca il loro metodo `onReceive()`

L'istanza di `BroadcastReceiver` viene riciclata al termine dell'esecuzione del metodo `onReceive()`. **I BC sono *stateless***

`onReceive()` non può operare in modo asincrono. Se necessario effettuare operazione lunghe si può invocare `goAsync()` e ricevere il risultato come `PendingIntent`.

- Una lista degli eventi generati dal sistema si trova qui

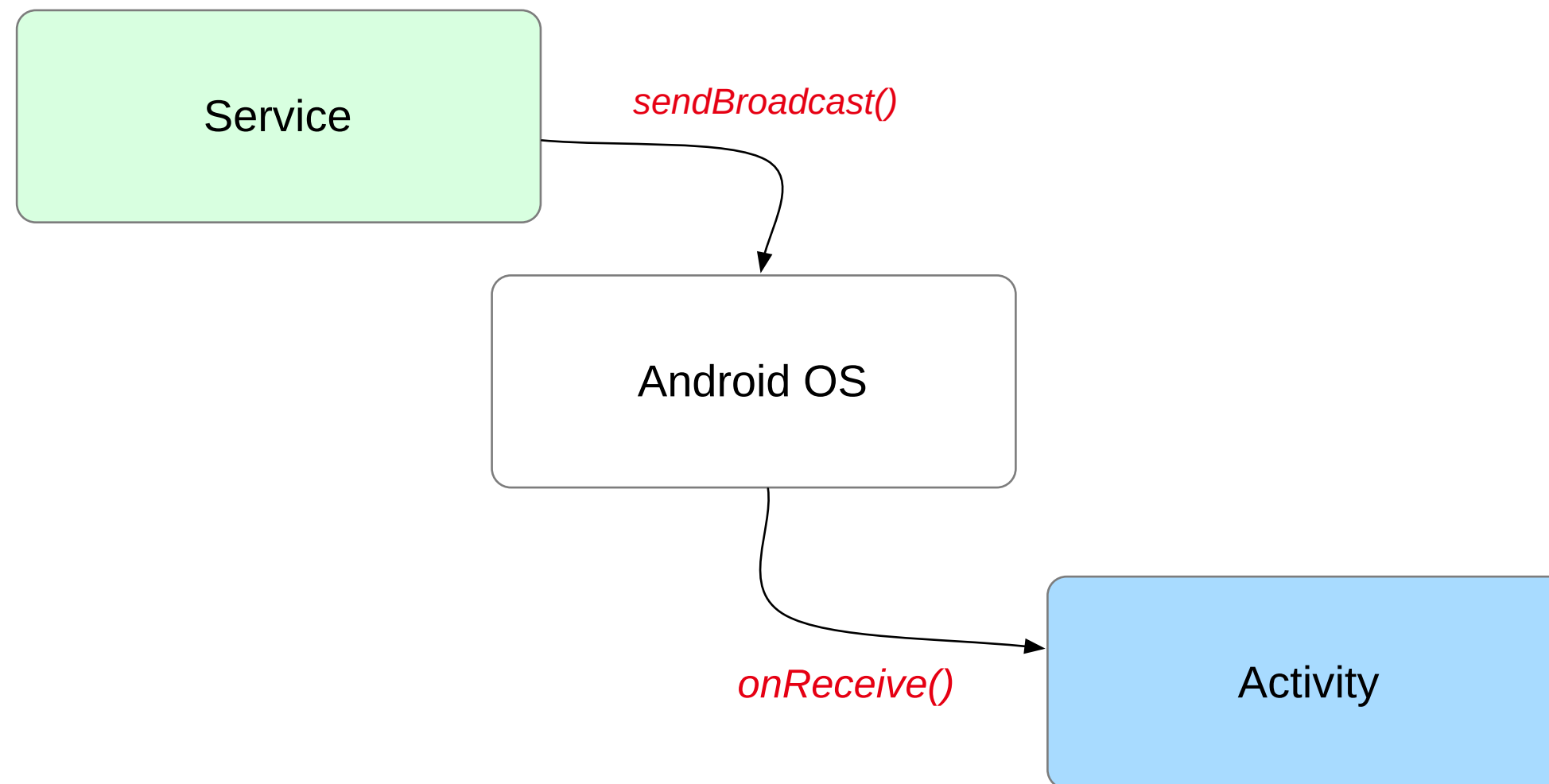
<http://developer.android.com/reference/android/content/Intent.html>

- Lo sviluppatore può aggiungere eventi propri e gestirne l'invio in broadcast e la ricezione.

```
Intent intent = new Intent();  
intent.setAction("biz.aeffegroup.broadcastevent");  
sendBroadcast(intent);
```

```
<receiver android:name="MyReceiver" >  
    <intent-filter>  
        <action android:name="biz.aeffegroup.broadcastevent" />  
    </intent-filter>  
</receiver>
```

- I broadcast receiver sono uno dei metodi, probabilmente il più semplice, che consentono una comunicazione (benchè unidirezionale) fra una Activity ed un Service. Disaccoppiamento fra componenti.



- ▶ Activity. Componenti che sovrintendono alla gestione della UI e controllano l'interazione con l'utente.
- ▶ Fragment. Componenti che consentono un'approccio modulare nella realizzazione e gestione di interfacce utente.
- ▶ Intent. Componenti che modellano in modo flessibile la comunicazione e le richieste di task fra componenti del sistema.
- ▶ BroadcastReceiver. Componenti che consentono la ricezione di notifiche riguardo ad eventi di sistema o stabiliti dallo sviluppatore
- ▶ **Service**
- ▶ ContentProvider

- Layout: i principali layout sono descritti alla pagina seguente:

<https://developer.android.com/guide/topics/ui/declaring-layout.html>

LinearLayout : vertical / Horizontal

RelativeLayout

GridLayout