Programming with Android:
Activities and Intents

Luca Bedogni **Marco Di Felice**

Dipartimento di Scienze dell'Informazione

Università di Bologna

Outline

- What is an **intent**?
- **Intent** description
- Handling **Explicit Intents**
- Handling **implicit Intents**
- **Intent-Resolution** process
- **Intent with results**: Sender side
- **Intent with results**: Receiver side

More on Activities: Activity states

- **Active** (or running)
  - Foreground of the screen (top of the stack)

- **Paused**
  - Lost focus but still visible
  - Can be killed by the system in extreme situations

- **Stopped**
  - Completely obscured by another activity
  - Killed if memory is needed somewhere else

More on Activities: Saving resources

- An activity lifecycle flows between onCreate and onDestroy
- Create, initialize everything you need in onCreate
- Destroy everything that is not used anymore, such as background processes, in onDestroy
- It is fundamental to save the data used by the application inbetween the state-transitions …

Activities and AndroidManifest.xml

- An Android application can be composed of **multiple Activities** …
- Each activity should be declared in the file: **AndroidManifest.xml**
- Add a **child element** to the <application> tag:

```
<application>
    <activity android:name=".MyActivity" />
    <activity android:name=".SecondActivity" />
</application>
```

Activities and AndroidManifest.xml

- Each activity has its **Java** class and **layout** file.

```
public class FirstActivity extends Activity {
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_first);
}
```

```
public class SecondActivity extends Activity {
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_two);
}
```

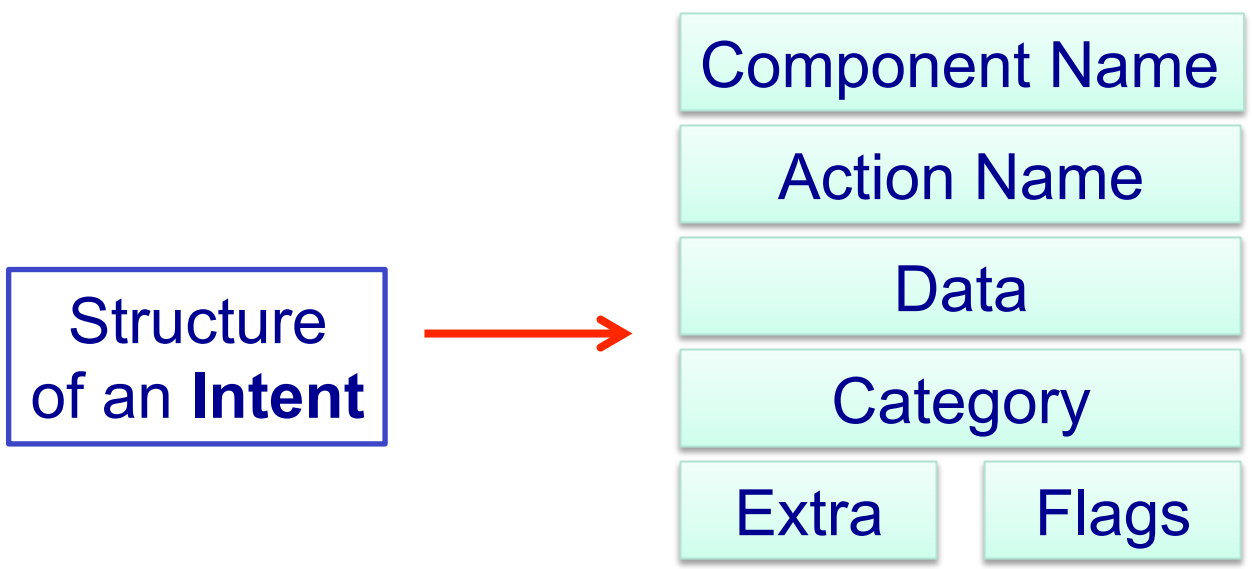
Intent Definition

Intent: facility for late run-time binding between components in the same or different applications.

- **Call** a component from another component
- Possible to **pass data** between components
- Components: **Activities**, *Services*, *Broadcast receivers* …
- Something like:
  - "Android, please do **that** with **this** data"
- **Reuse** already installed applications and components

Intent Definition

- We can think to an "**Intent**" object as a **message** containing a bundle of information.
  - Information of interests for the receiver (e.g. name)
  - Information of interests for the Android system (e.g. category).



Intent types

INTENT **TYPES**

EXPLICIT

The target receiver is specified through the **Component Name**

Used to launch specific Activities

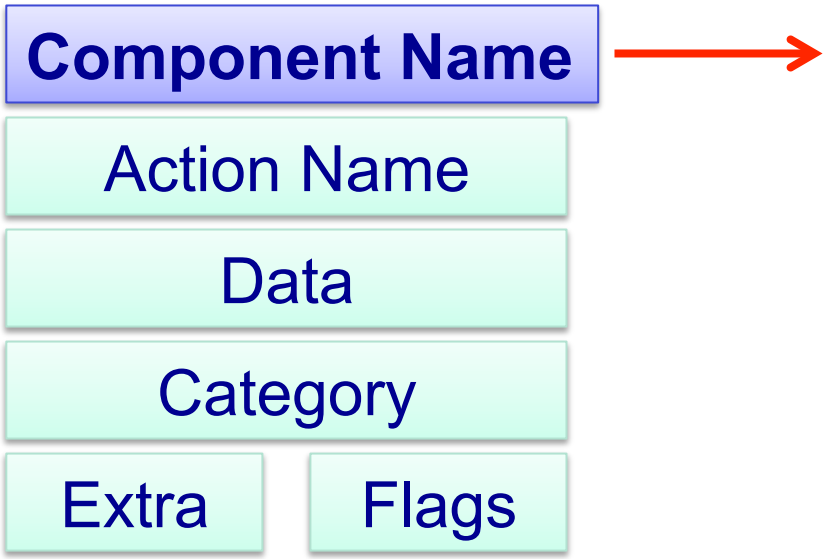
IMPLICIT

The target receiver is specified by **data type/names**.

The system chooses the receiver that matches the request.

Intent Components

- We can think to an "**Intent**" object as a **message** containing a bundle of information.
  - Information of interests for the receiver (e.g. data)
  - Information of interests for the Android system (e.g. category).



Component that should handle the intent (i.e. the **receiver**).

It is **optional** (implicit intent)

void **setComponent**(ComponentName)

Intent types: Explicit Intents

- **Explicit** Intent: Specify the name of the Activity that will handle the intent.

```
Intent intent=new Intent(this, SecondActivity.class);
startActivity(intent);
```

```
Intent intent=new Intent();
ComponentName component=new
ComponentName(this,SecondActivity.class);
intent.setComponent(component);
startActivity(intent);
```

Intent with Results

- Activities can return results (e.g. data)
- Sender side: invoke the **startActivityForResult**()
  - **onActivityResult**(int requestCode, int resultCode, Intent data)
  - **startActivityForResult**(Intent intent, int requestCode);

```
Intent intent = new Intent(this, SecondActivity.class);
startActivityForResult(intent, CHOOSE_ACTIVITY_CODE);
…
public void onActivityResult(int requestCode, int resultCode, Intent data)
{
    // Invoked when SecondActivity completes its operations …
}
```

Intent with Results

- Activities can return results (e.g. data)
- Receiver side: invoke the **setResult()**
  - void **setResult**(int resultCode, Intent data)

```
Intent intent=new Intent();
setResult(RESULT_OK, intent);
intent.putExtra(”result", resultValue);
finish();
```

- The result is delivered to the caller component only after invoking the **finish()** method!

Intent types

INTENT **TYPES**

EXPLICIT

The target receiver is specified through the **Component Name**

Used to launch specific Activities

IMPLICIT

The target receiver is specified by **data type/names**.

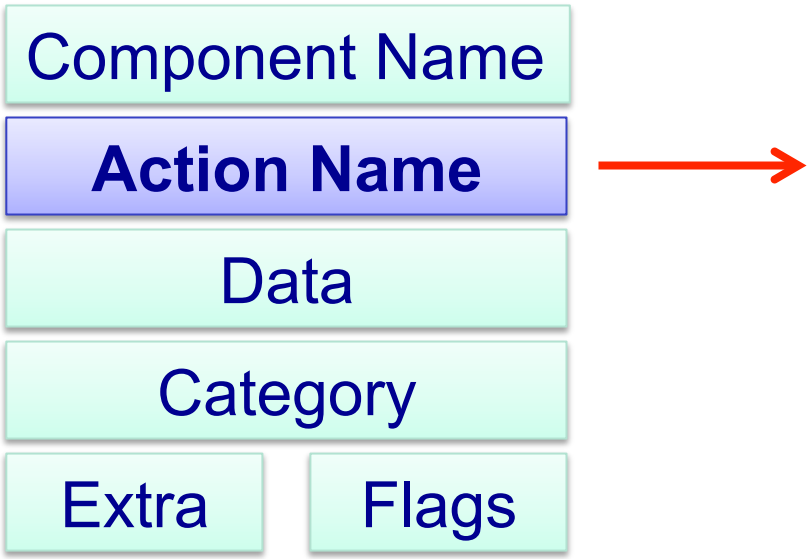
The system chooses the receiver that matches the request.

Intent types: Implicit Intents

- **Implicit** Intents: do not name a target (*component name is left blank*) …
- When an Intent is launched, Android checks out which activies might answer to the Intent …
- If at least one is found, then that activity is started!
- Binding does not occur at compile time, nor at install time, but at run-time …(***late run-time binding***)

Intent Components

- We can think to an "**Intent**" object as a **message** containing a bundle of information.
  - Information of interests for the receiver (e.g. data)
  - Information of interests for the Android system (e.g. category).



A string naming the **action** to be performed.

Pre-defined, or can be specified by the programmer.

void **setAction**(String)

Intent Components

- **Predefined** actions (http://developer.android.com/reference/android/content/Intent.html)

Action Name	Description
ACTION_EDIT	Display data to edit
ACTION_MAIN	Start as a main entry point, does not expect to receive data.
ACTION_PICK	Pick an item from the data, returning what was selected.
ACTION_VIEW	Display the data to the user
ACTION_SEARCH	Perform a search

- **Defined** by the programmer
  - it.example.projectpackage.FILL_DATA (package prefix + name action)

Intent Components

- **Special** actions (http://developer.android.com/reference/android/content/Intent.html)

Action Name	Description
ACTION_IMAGE_CAPTION	Open the camera and receive a photo
ACTION_VIDEO_CAPTION	Open the camera and receive a video
ACTION_DIAL	Open the phone app and dial a phone number
ACTION_SENDTO	Send an email (email data contained in the extra)
ACTION_SETTINGS	Open the system setting
ACTION_WIRELESS_SETTINGS	Open the system setting of the wireless interfaces
ACTION_DISPLAY_SETTINGS	Open the system setting of the display

Intent Components

- Example of Implicit Intent that **initiates a web search**.

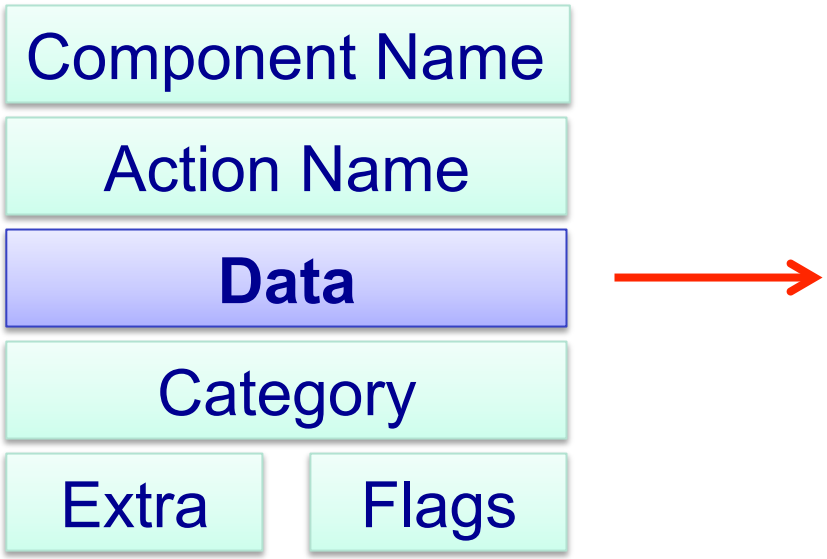
```
public void doSearch(String query) {
    Intent intent =new Intent(Intent.ACTION_SEARCH);
    Intent.putExtra(SearchManager.QUERY,query);
    if (intent.resolveActivity(getPackageManager()) !=null)
        startActivity(intent)
}
```

- Example of Implicit Intent that **plays a music file**.

```
public void playMedia(Uri file) {
    Intent intent =new Intent(Intent.ACTION_VIEW);
    if (intent.resolveActivity(getPackageManager()) !=null)
        startActivity(intent)
}
```

Intent Components

- We can think to an "**Intent**" object as a **message** containing a bundle of information.
  - Information of interests for the receiver (e.g. data)
  - Information of interests for the Android system (e.g. category).



Data passed from the caller to the called Component.

Def. of the data (**URI**) and Type of the data (**MIME** type)

void **setData**(Uri)
void **setType**(String)

Intent Components

- Each data is specified by a **name** and/or **type**.
- **name**: Uniform Resource Identifier (**URI**)
- **scheme://host:port/path**

EXAMPLEs

tel://003-232-234-678

content://contacts/people

http://www.cs.unibo.it/

Intent Components

- Each data is specified by a **name** and/or **type**.
- **type**: **MIME** (Multipurpose Internet Mail Extensions)-type
- Composed by two parts: a type and a subtype

EXAMPLEs

Image/gif image/jpeg image/png image/tiff
text/html text/plain text/javascript text/css
video/mp4 video/mpeg4 video/quicktime video/ogg
application/vnd.google-earth.kml+xml

Intent Components

- We can think to an "**Intent**" object as a **message** containing a bundle of information.
  - Information of interests for the receiver (e.g. data)
  - Information of interests for the Android system (e.g. category).

A string containing information about the **kind of component** that should handle the Intent.

> 1 can be specified for an Intent

void **addCategory**(String)

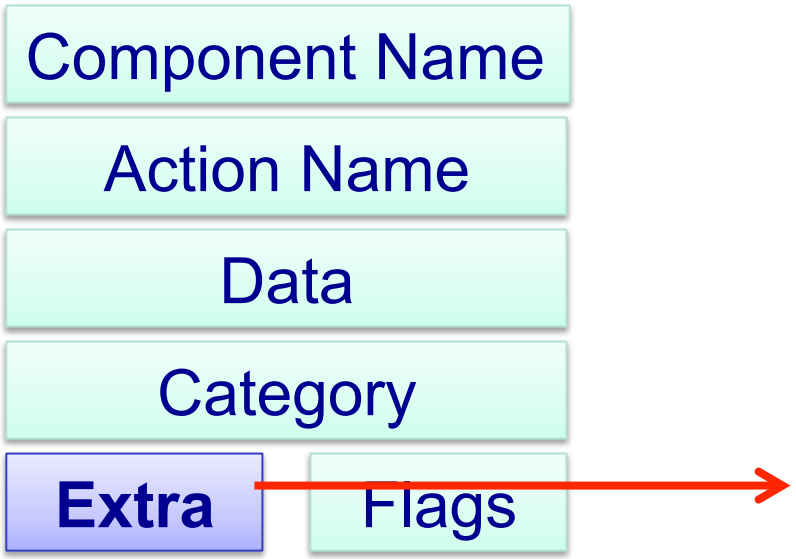
Intent Components

- **Category**: string describing the **kind of component** that should handle the intent.

Category Name	Description
CATEGORY_HOME	The activity displays the HOME screen.
CATEGORY_LAUNCHER	The activity is listed in the top-level application launcher, and can be displayed.
CATEGORY_PREFERENCE	The activity is a preference panel.
CATEGORY_BROWSABLE	The activity can be invoked by the browser to display data referenced by a link.

Intent Components

- We can think to an "**Intent**" object as a **message** containing a bundle of information.
  - Information of interests for the receiver (e.g. data)
  - Information of interests for the Android system (e.g. category).



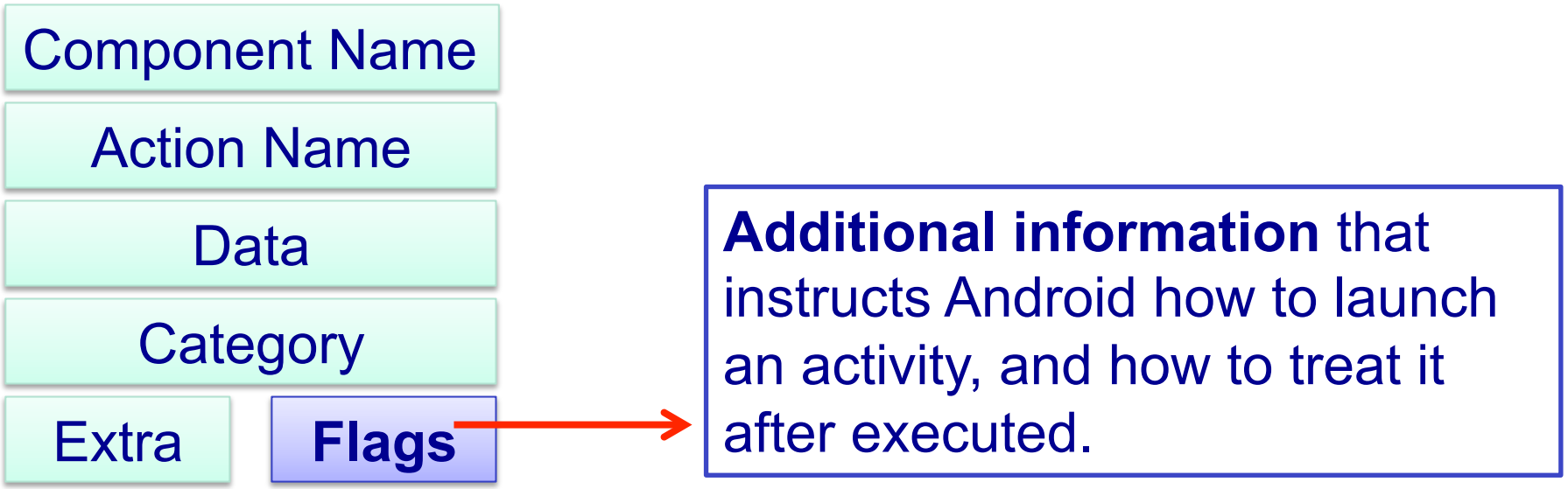
Additional information that should be delivered to the handler(e.g. parameters).

Key-value pairs

void **putExtras**() **getExtras()**

Intent Components

- We can think to an "**Intent**" object as a **message** containing a bundle of information.
  - Information of interests for the receiver (e.g. data)
  - Information of interests for the Android system (e.g. category).



Intent types: Implicit Intents

```
Intent i = new
    Intent(android.content.Intent.ACTION_VIEW,
    Uri.parse("http://informatica.unibo.it"));
startActivity(i);
```

Action to perform

Data to perform the action on

- Implicit intents are very useful to **re-use code** and to **launch external applications …**
- *More than a component can match the Intent request …*
- **How to define the target component**?

Intent types: Implicit Intents

- How to declare what intents I'm able to handle?

  <**intent-filter**> tag in AndroidManifest.xml

- How?

  ```
  <intent-filter>
      <action android:name="my.project.ACTION_ECHO" />
  </intent-filter>
  ```

- If a component creates an Intent with "my.project.ACTION_ECHO" as action, the corresponding activity will be executed …

Intent types: Intent Resolution

- The **intent resolution** process resolves the Intent-Filter that can handle a given Intent.
- Three tests to be passed:
  - **Action field** test
  - **Category field** test
  - **Data field** test
- If the Intent-filter passes all the three test, then it is selected to handle the Intent.

Intent types: Intent Resolution

- (**ACTION** Test): The action specified in the Intent must match one of the actions listed in the filter.
  - If the filter does not specify any action → **FAIL**
  - An intent that does not specify an action → **SUCCESS** as as long as the filter contains at least one action.

```
<intent-filer … >
    <action android:name="com.example.it.ECHO"/>
</intent-filter>
```

Intent types: Intent Resolution

- (**CATEGORY** Test): Every category in the Intent must match a category of the filter.

- If the category is not specified in the Intent → Android assumes it is CATEGORY_DEFAULT, thus the filter must include this category to handle the intent

```
<intent-filer … >
    <category android:name="android.intent.category.DEFAULT"/>
</intent-filter>
```

Intent types: Intent Resolution

- (**DATA** Test): The URI of the intent is compared with the parts of the URI mentioned in the filter (this part might be incompleted).

```
<intent-filer … >
    <data android:mimeType="audio/*  android:scheme="http"/>
    <data android:mimeType="video/mpeg  android:scheme="http"/>
</intent-filter>
```

- Both URI and MIME-types are compared (4 different sub-cases …)