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-field** description
- Handling **Explicit Intents**
- Handling **implicit Intents**
- **Intent-Resolution** process
- **Intent with results**: Sender side
- **Intent with results**: Receiver side

More on AndroidManifest.xml

- Applications should declare everything needed on the the **AndroidManifest.xml** file …

- One AndroidManifest.xml for application ..

- What's contained in it?
  - *Permissions*
  - *Hw and Sw resources used by the Application*
  - *Activities*
  - *Intent-filters*

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
- Fundamental to save the data used by the application between 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 of <application>:

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

AndroidManifest.xml example

```
<?xml version="1.0" encoding="utf-8"?>
<manifest>

    <application android:icon="@drawable/icon.png" >

        <activity
                android:name="com.example.project.MyActivity"
                android:label="@string/label">
        </activity>

    </application>

</manifest>
```

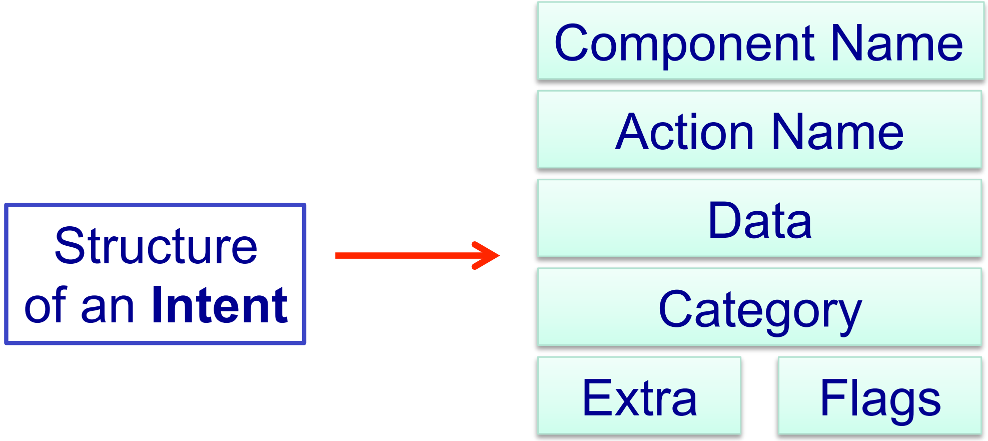
Intent Definition

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
- Something like:
  - "Android, please do **that** with **this** data"
- **Reuse** already installed applications
- Components: **Activities**, *Services*, *Broadcast receivers* …

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. data)
  - Information of interests for the Android system (e.g. category).



Structure of an **Intent**

Component Name

Action Name

Data

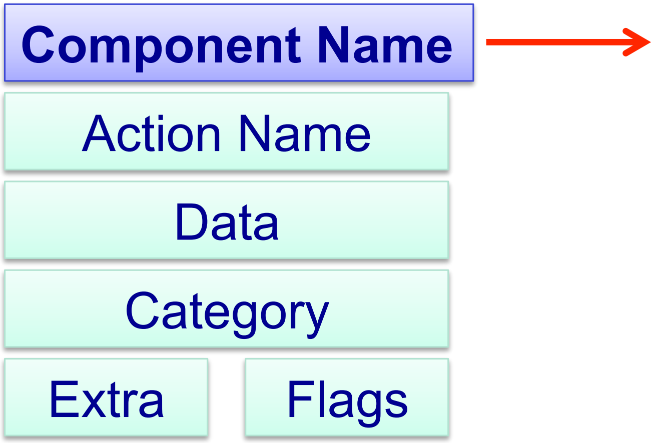
Category

Extra

Flags

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 Name

Action Name

Data

Category

Extra

Flags

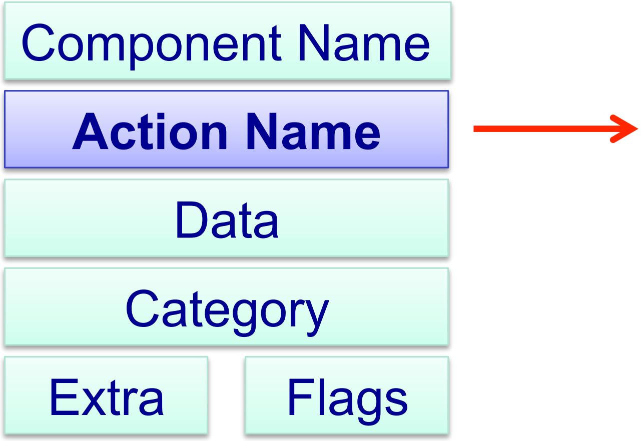
Component that should handle the intent (i.e. the **receiver**).

It is **optional** (implicit intent)

void **setComponent**()

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**()

Intent Components

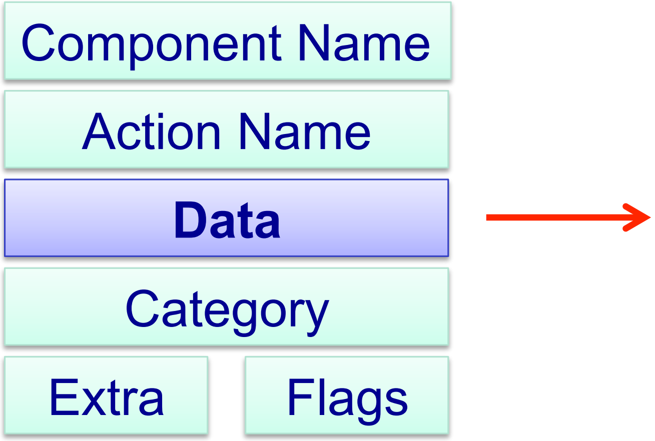
- **Predefined** actions (http://developer.android.com/reference/android/content/Intent.html)

Action Name	Description
ACTION_CALL	Initiate a phone call
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

- 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 Name

Action Name

Data

Category

Extra

Flags

Data passed from the caller to the called Component.

Location of the data (**URI**) and Type of the data (**MIME** type)

void **setData**()

Intent Components

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

EXAMPLEs

content://com.example.project:200/folder

content://contacts/people

content://contacts/people/1

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

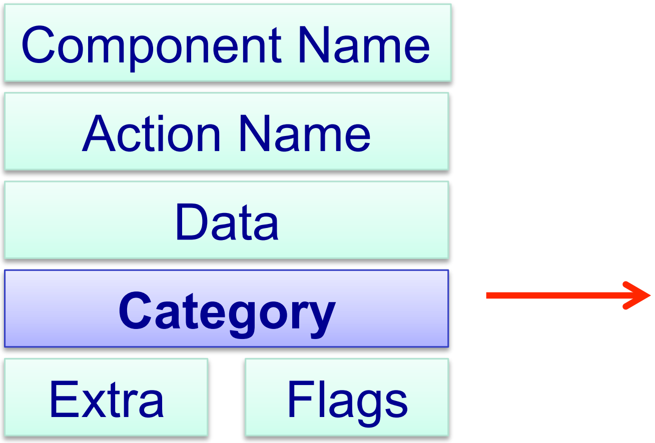
text/html text/plain text/javascript text/css

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).



Component Name

Action Name

Data

Category

Extra

Flags

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

> 1 can be specified for an Intent

void **addCategory**()

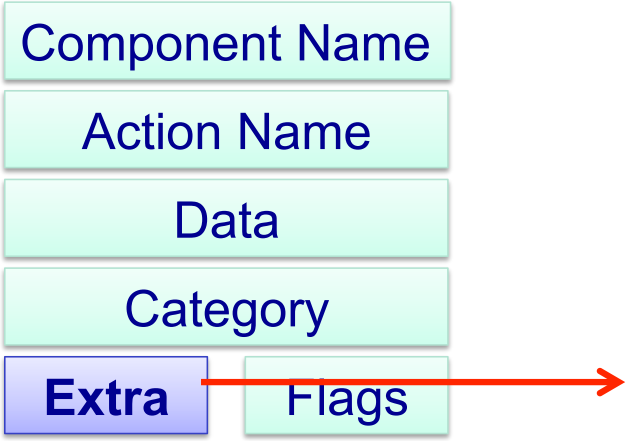
Intent Components

- **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).



Component Name

Action Name

Data

Category

Extra

Flags

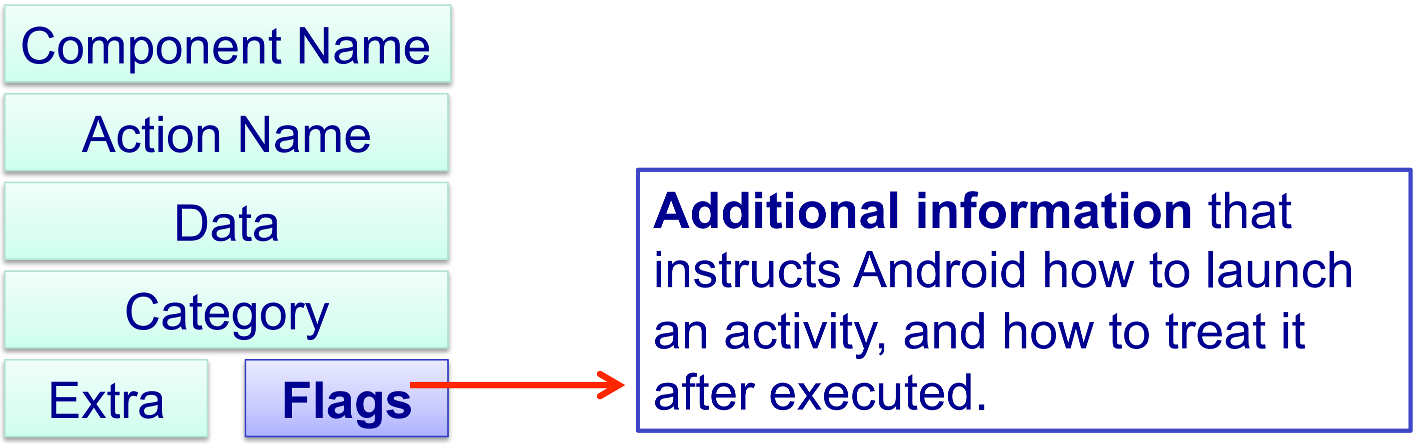
Additional information that should be delivered to the handler(e.g. parameters).

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).



Component Name

Action Name

Data

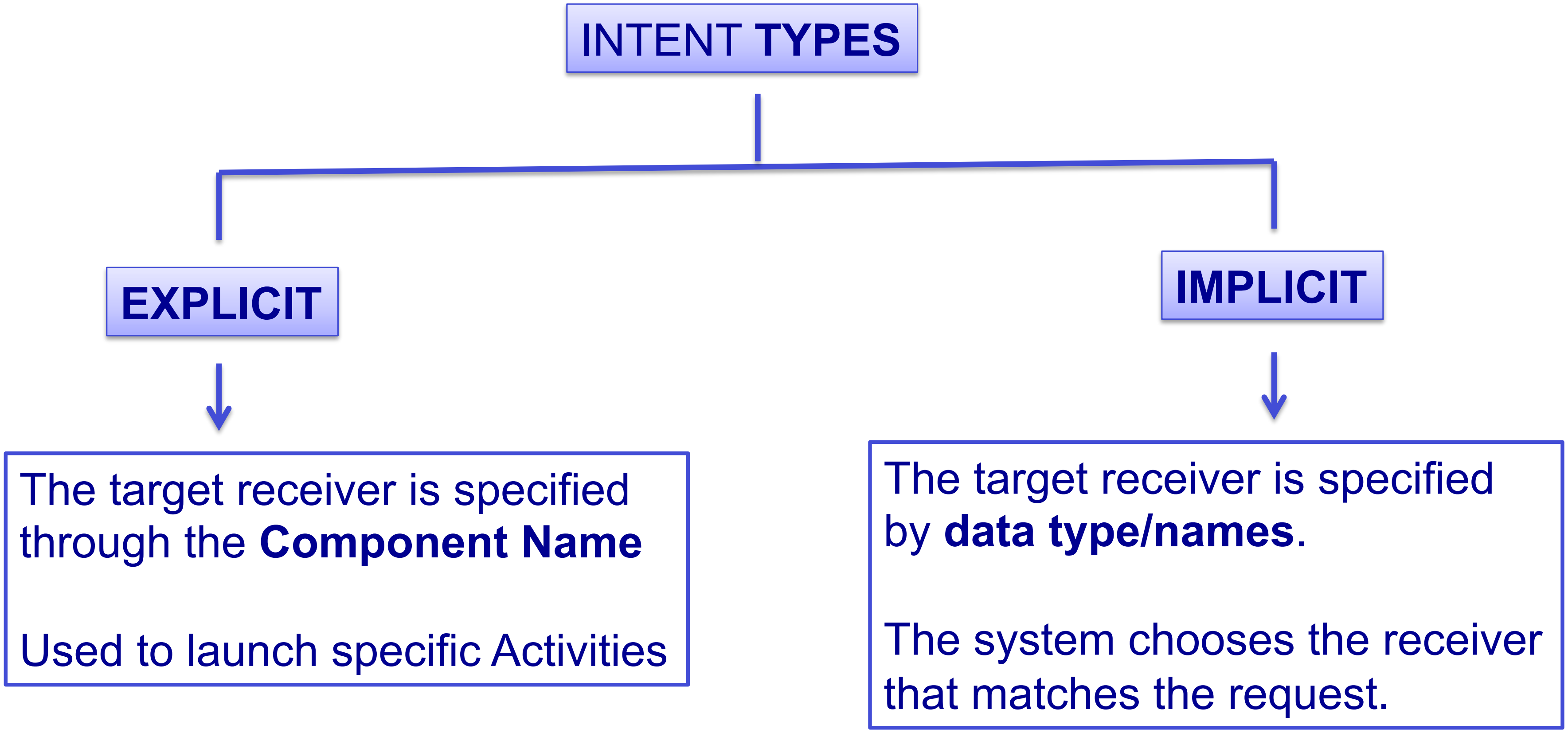
Category

Extra

Flags

Additional information that instructs Android how to launch an activity, and how to treat it after executed.

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: Explicit Intents

- **Explicit** Intent: Specify 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 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 could answer to such 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 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 anyone calls an Intent with “my.project.ACTION_ECHO” as action, our activity will be called

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 …)

Intent **with Results**

- Activities could return results ....Useful for calling activities and have some data back!
- Sender side: invoke the **startActivityForResult**()
- **onActivityResult**(int requestCode, int resultCode, Intent data)
- **startActivityForResult**(Intent intent, int requestCode);

```
Intent intent = new Intent(Intent.ACTION_PICK);
intent.setData(android.provider.Contacts.People.CONTENT_URI);
startActivityForResult(intent,CHOOSE_ACTIVITY_CODE);
```

Intent with Results

- Activities could return results ....Useful for calling activities and have some data back
- Receiver side: invoke the **setResult()**
- void **setResult**(int resultCode, Intent data)

```
void setResult(int resultCode, Intent data);
finish();
```

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