Trường Đại học Bách Khoa Hà Nội
Hanoi University of Science and Technology

Chapter 9

Intent and Notifications

© HUST 2012

What is an Intent

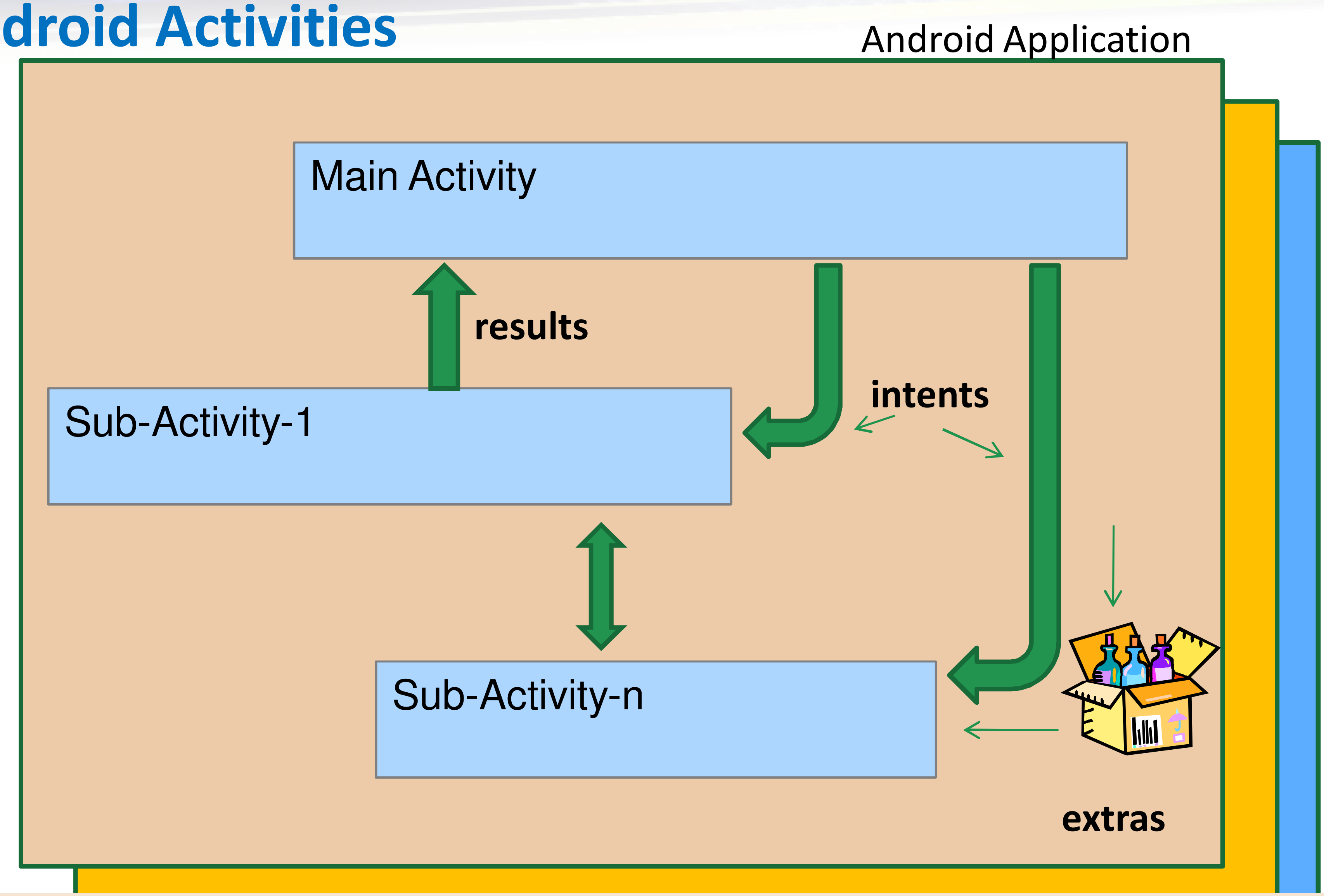
- Intents represent an action plus context

An intent is basically a message that you pass to Android saying,

"Yo! I want to do...er...something! Yeah!“

What is an Intent

Android Activities



Intents represent an action plus context

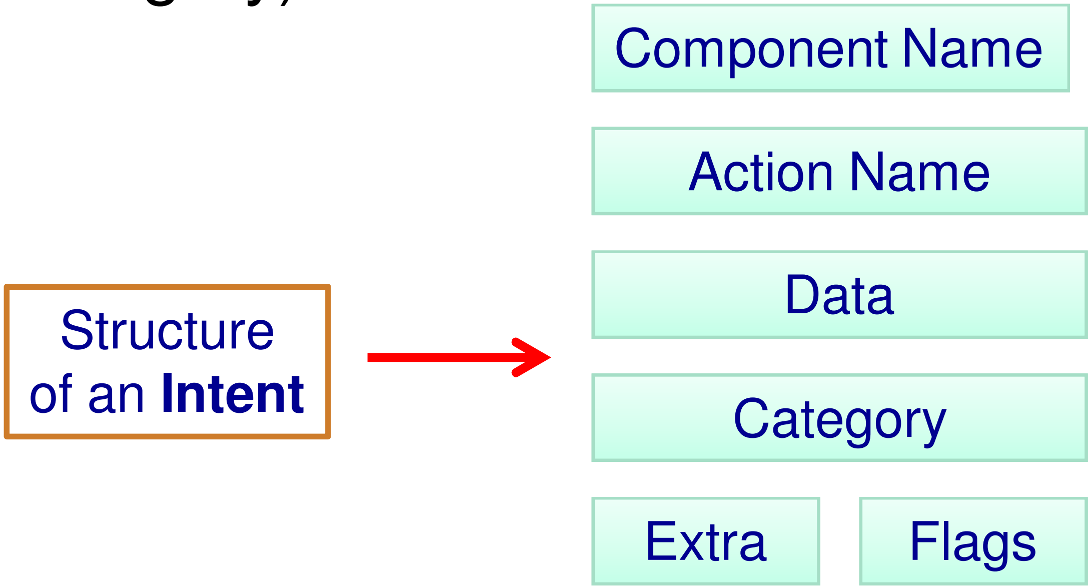
Intent Definition

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 **<u>that</u>** with **<u>this</u>** 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 <u>interests for the receiver </u>(e.g. data)
  - Information of <u>interests for the Android system </u>(e.g. category).



Structure of an **Intent**

Component Name

Action Name

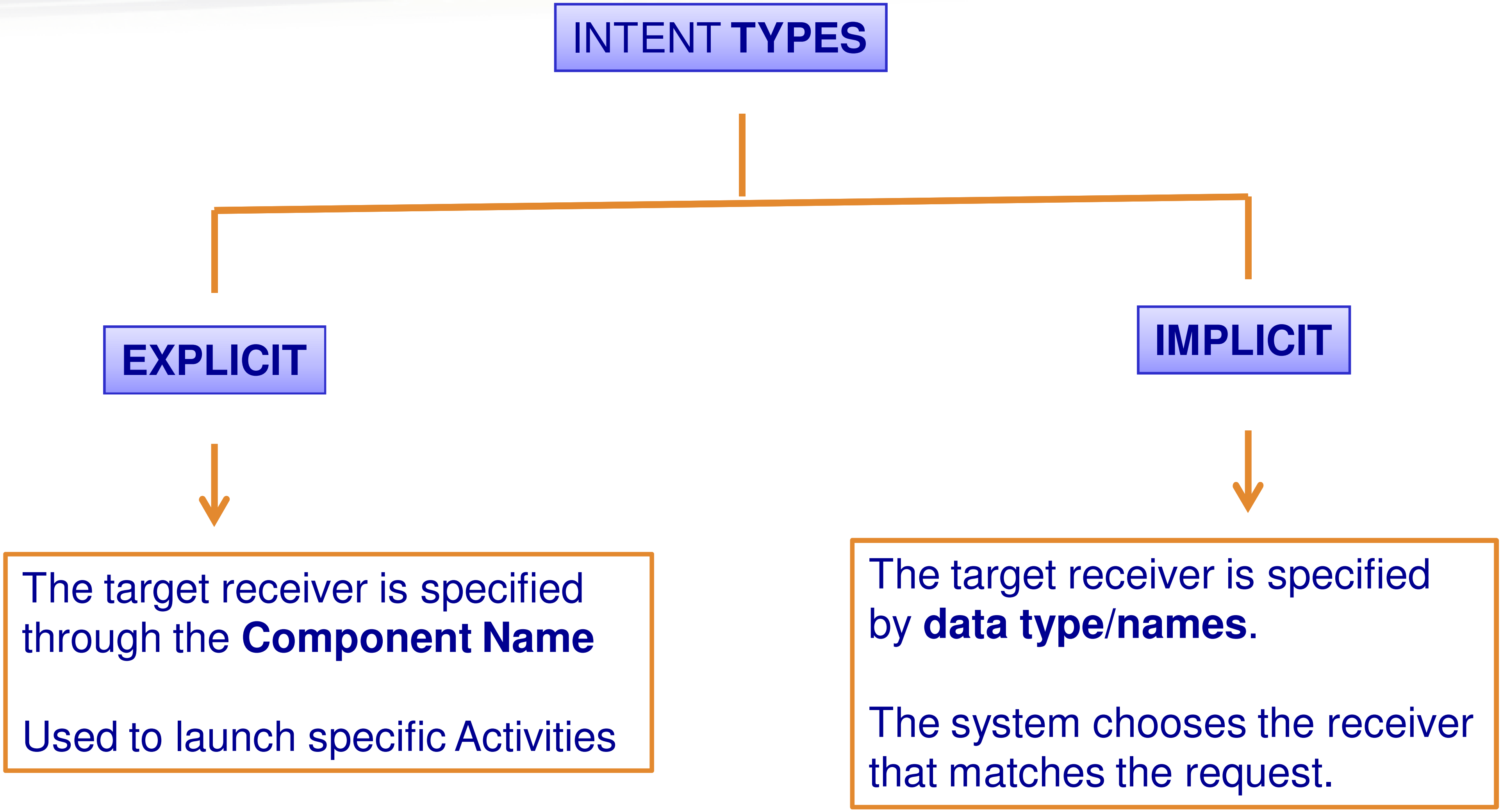
Data

Category

Extra

Flags

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_V
        IEW,
        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

Invoking Intents

***startActivity* (intent)**

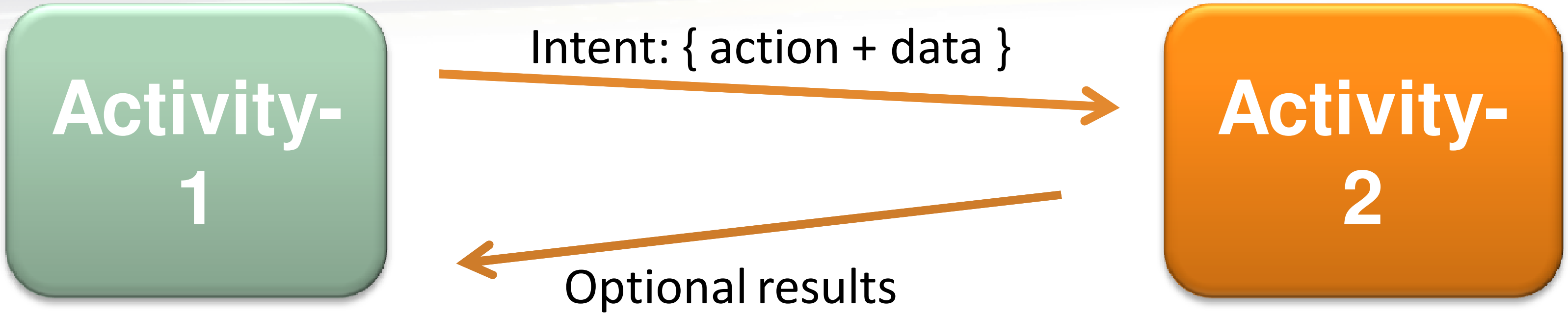
***sendBroadcast* (intent)**

***startService*(intent)**

or

***bindService*(intent, …)**

Intent primary attributes



```
Intent  myActivity = new Intent (action, data);

startActivity (myActivity);
```

Built-in or user-created activity

Primary data (as an URI)
tel://
http://
sendto://

Standard Activity Actions

Built-in Standard Actions

ACTION_MAIN

ACTION_VIEW

ACTION_ATTACH_DATA

ACTION_EDIT

ACTION_PICK

ACTION_CHOOSER

ACTION_GET_CONTENT

ACTION_DIAL

ACTION_CALL

ACTION_SEND

ACTION_SENDTO

ACTION_ANSWER

ACTION_INSERT

ACTION_DELETE

ACTION_RUN

ACTION_SYNC

ACTION_PICK_ACTIVITY

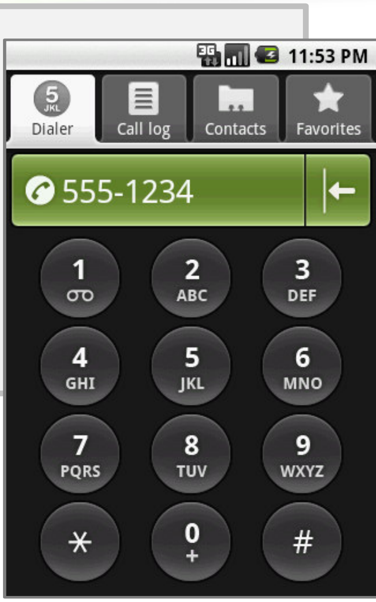
ACTION_SEARCH

ACTION_WEB_SEARCH

ACTION_FACTORY_TEST

Example: Display phone dialer

```
Intent myActivity2 = new Intent (Intent.ACTION_DIAL,
                        Uri.parse( "tel:555-1234"));
startActivity(myActivity2);
```



Secondary Attributes

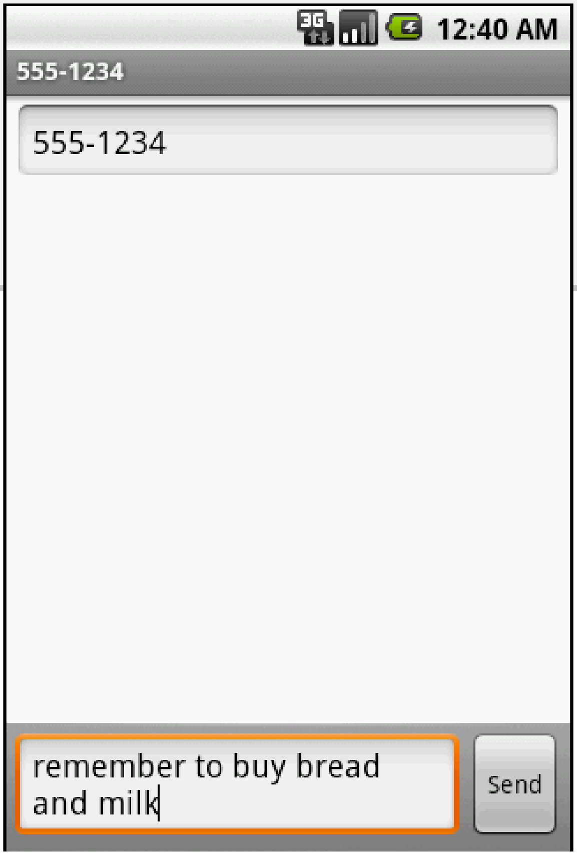
1. Category
2. Components
3. Type
4. Extras

```
Intent intent = new Intent (Intent.ACTION_WEB_SEARCH );

intent.putExtra(SearchManager.QUERY,
               "straight hitting golf clubs");

startActivity(intent);
```

Secondary data

Example: sending an SMS

```
Intent intent = new Intent( Intent.ACTION_SENDTO,
                     Uri.parse("sms:5551234"));

intent.putExtra("sms_body", "are we playing golf next
Saturday?");

startActivity(intent);
```



12:40 AM

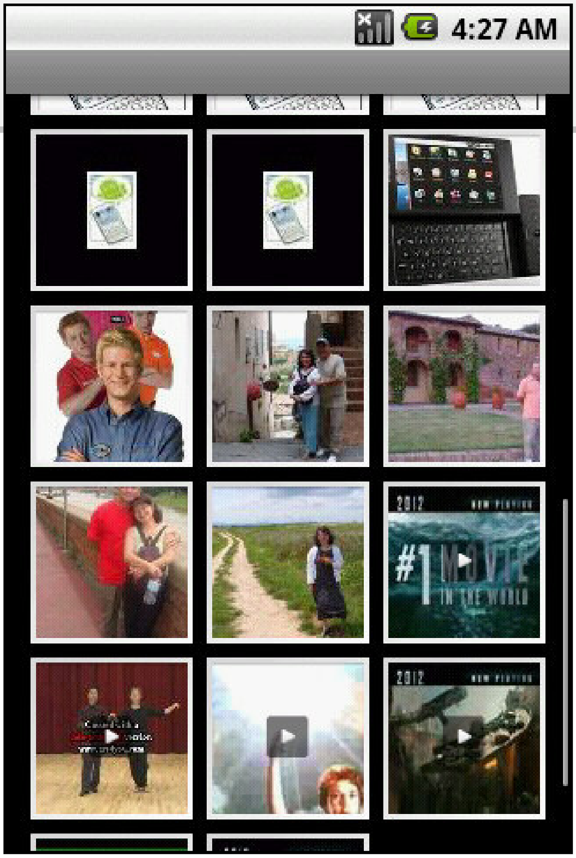
555-1234

555-1234

remember to buy bread and milk

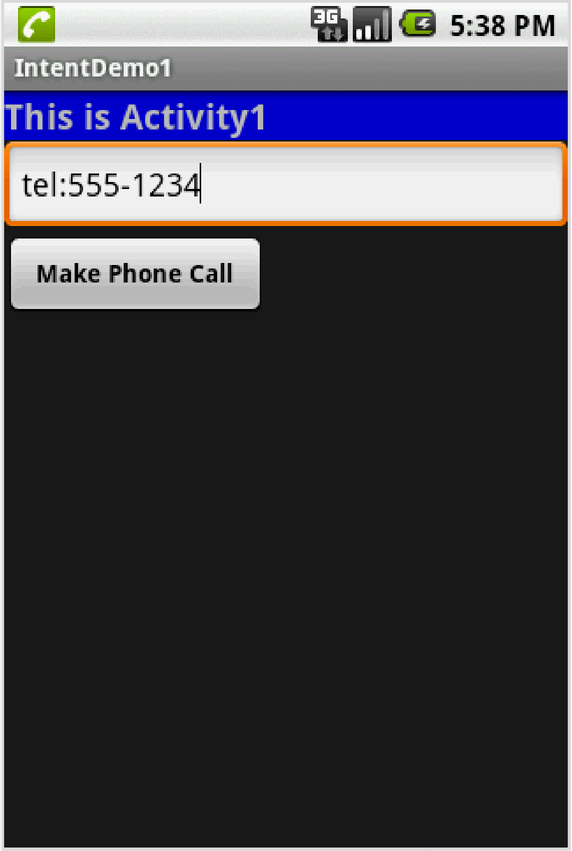
Example: Displaying an image

```
Intent myIntent = new Intent();

 myIntent.setType("image/pictures/*");
 myIntent.setAction(Intent.ACTION_GET_CONTENT);

 startActivity(myIntent)
```



Complete Example

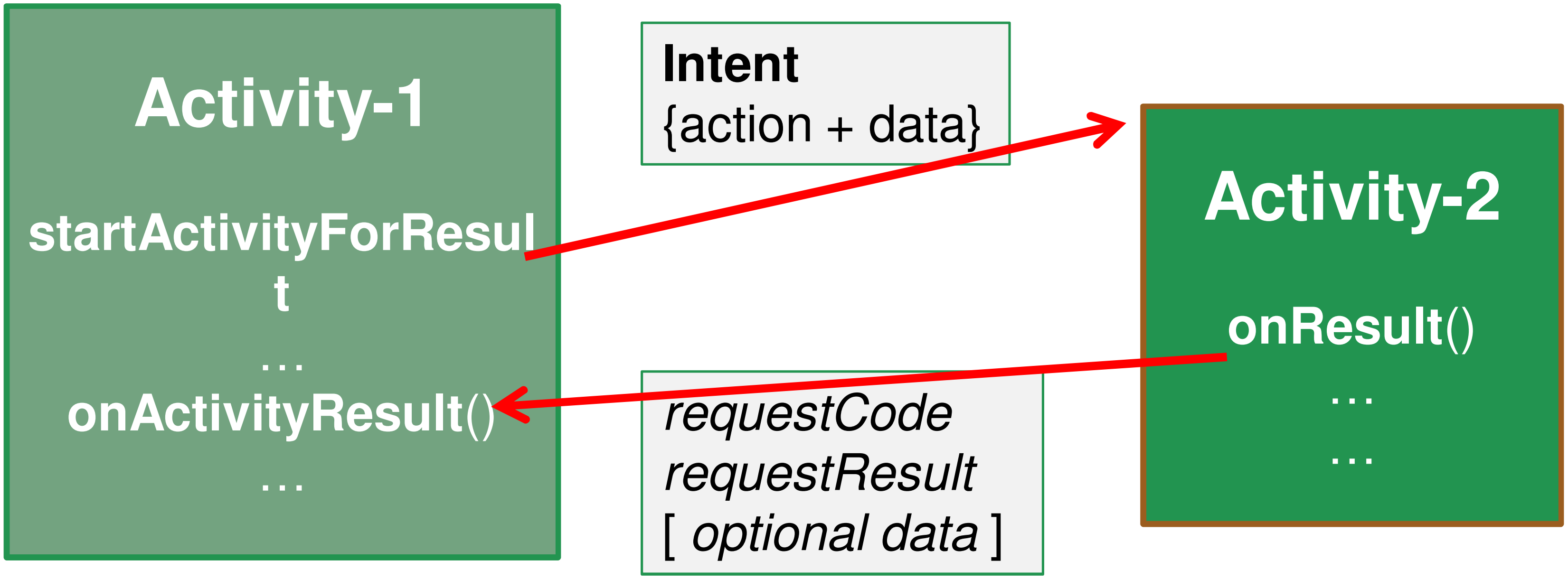
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"  >
<TextView
     android:id="@+id/label1"
     android:layout_width="fill_parent"
     android:layout_height="wrap_content"
     android:background="#ff0000cc"
     android:text="This is Activity1"
     android:textStyle="bold"
     android:textSize="20sp"   />
<EditText
     android:id="@+id/text1"
     android:layout_width="fill_parent"
     android:layout_height="54px"
     android:text="tel:555-1234"
     android:textSize="18sp"  />

<Button
     android:id="@+id/btnCallActivity2"
     android:layout_width="149px"
     android:layout_height="wrap_content"
     android:text="Make Phone Call"
     android:textStyle="bold" />
</LinearLayout>
```



Android intents

An *activity* usually presents a single visual user interface from which a number of actions could be performed.

Moving from one activity to another is accomplished by having the current activity start the next one through so called ***intents***.



Android Bundles

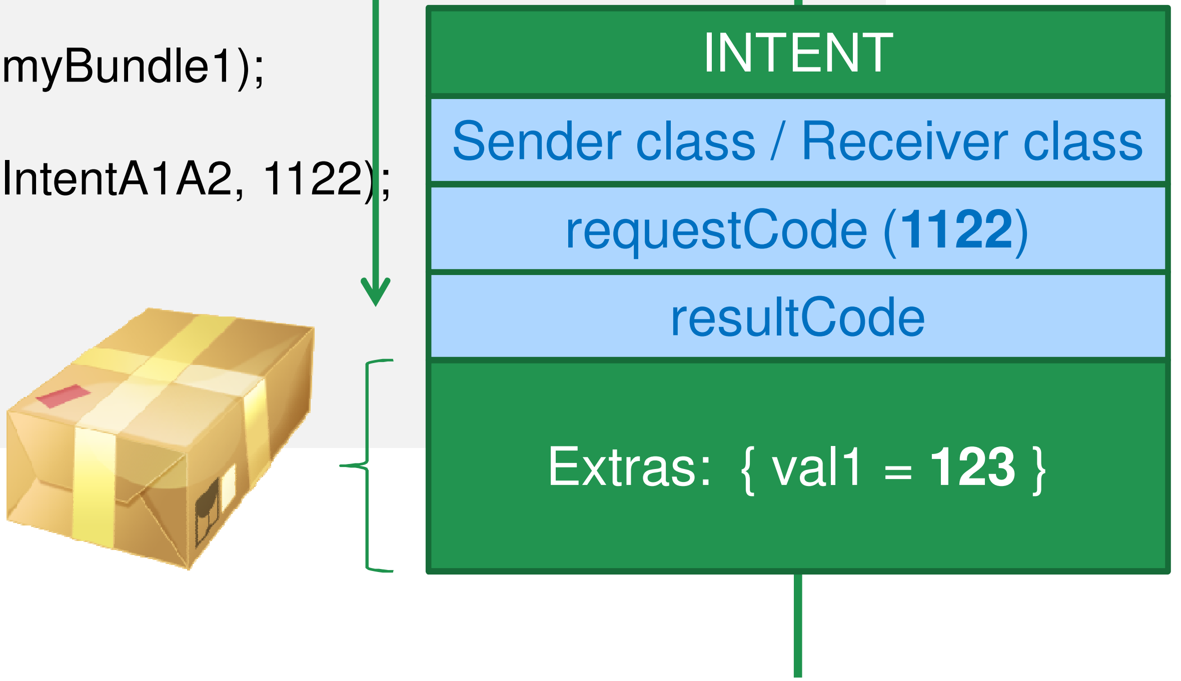
```
Bundle myBundle = new Bundle();
    myBundle.putDouble (“var1”, 3.1415);
    ...
    Double v1 = myBundle.getDouble(“var1”);
```

Android Intents & Bundles

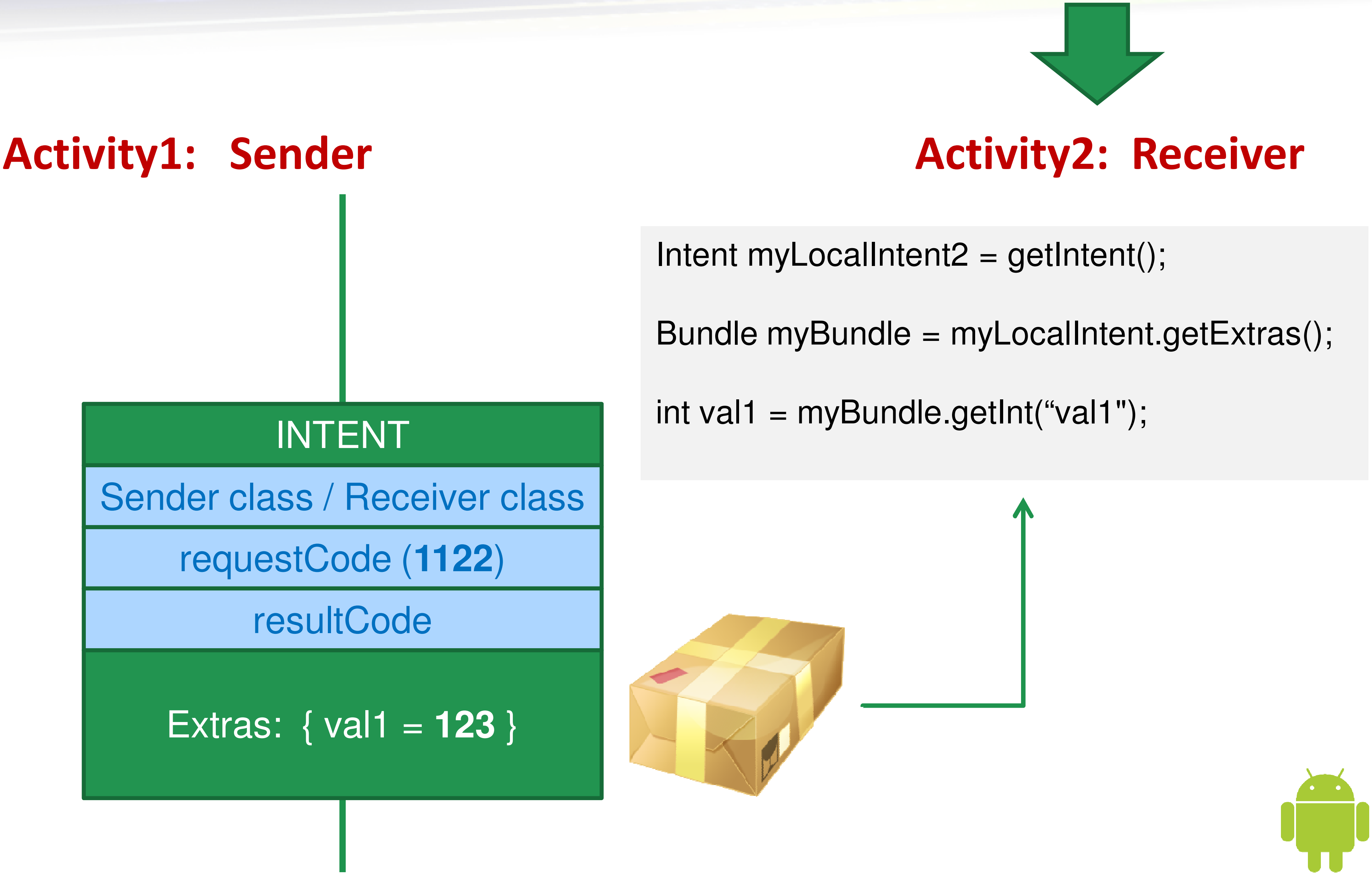
Activity1: Sender

Activity2: Receiver

```
Intent myIntentA1A2 = new Intent  (Activity1.this, Activity2.class);

Bundle myBundle1 = new Bundle();
myBundle1.putInt ("val1", 123);

myIntentA1A2.putExtras(myBundle1);

startActivityForResult(myIntentA1A2, 1122);
```



Android Intents & Bundles



Activity1: Sender

Activity2: Receiver

```
Intent myLocalIntent2 = getIntent();

Bundle myBundle = myLocalIntent.getExtras();

int val1 = myBundle.getInt(“val1");
```

INTENT

Sender class / Receiver class

requestCode (**1122**)

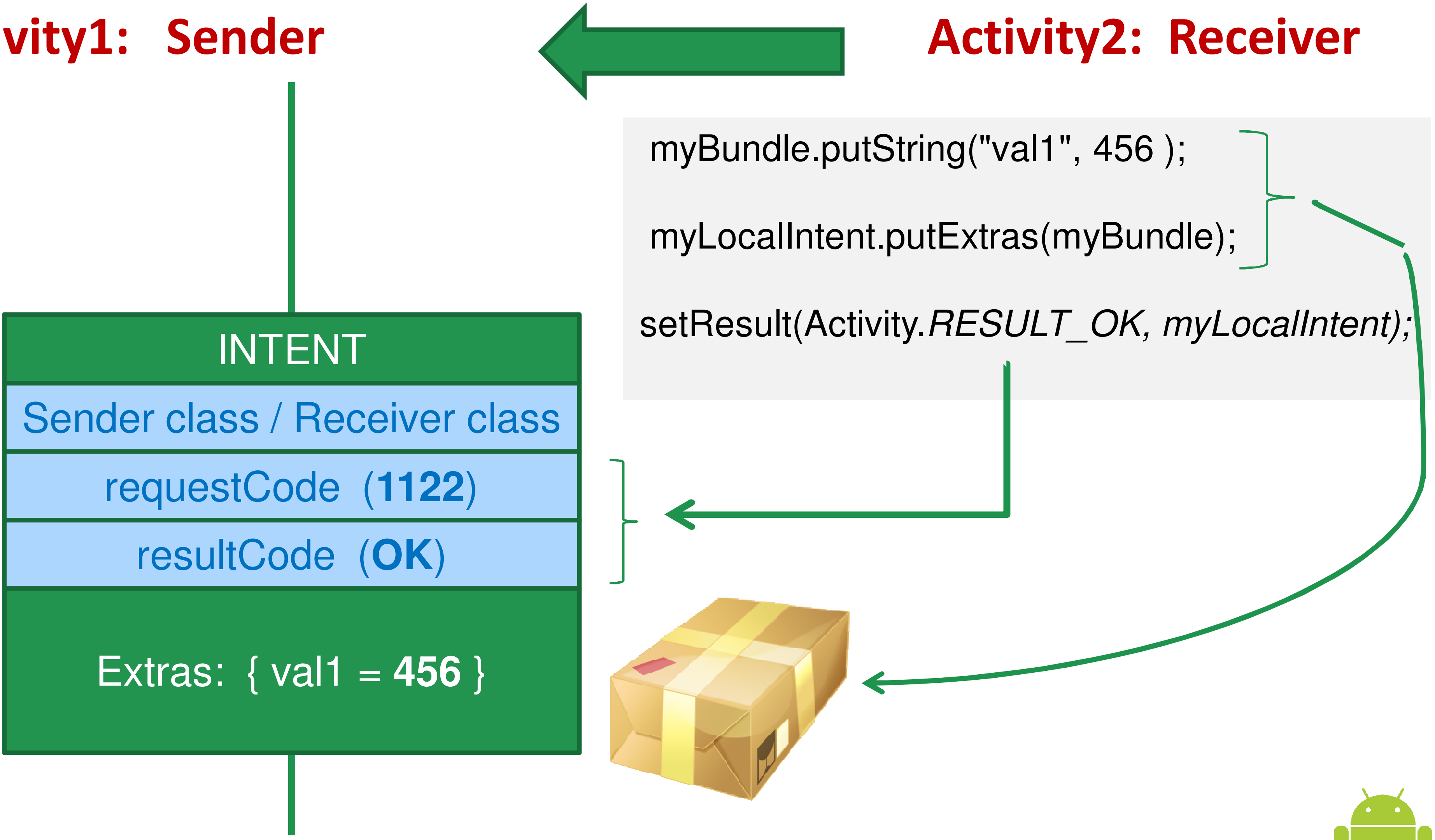
resultCode

Extras: { val1 = **123** }

Android Intents & Bundles

Activity1: Sender

Activity2: Receiver

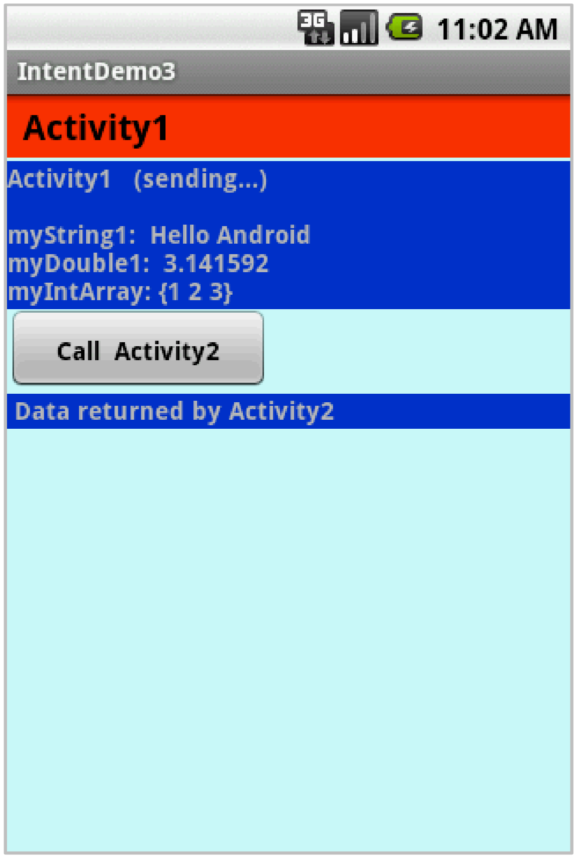


```
myBundle.putString("val1", 456 );

myLocalIntent.putExtras(myBundle);

setResult(Activity.RESULT_OK, myLocalIntent);
```

INTENT
Sender class / Receiver class
requestCode (**1122**)
resultCode (**OK**)
Extras: { val1 = **456** }

Intents

Example: Activity1 invokes Activity2 using an Intent. A bundle containg a set of values is sent back-and-forth between both activities.



11:02 AM

IntentDemo3

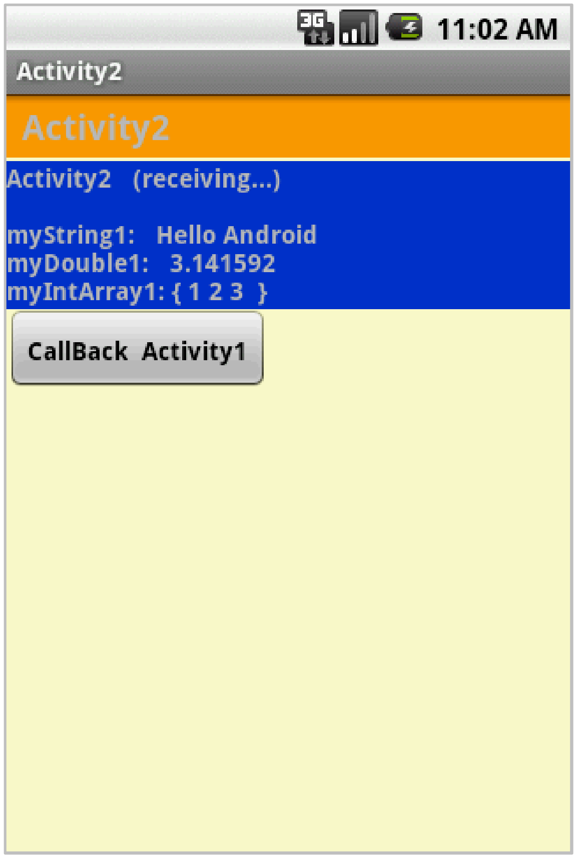
Activity1

Activity1 (sending...)

myString1: Hello Android
myDouble1: 3.141592
myIntArray: {1 2 3}

Call Activity2

Data returned by Activity2



11:02 AM

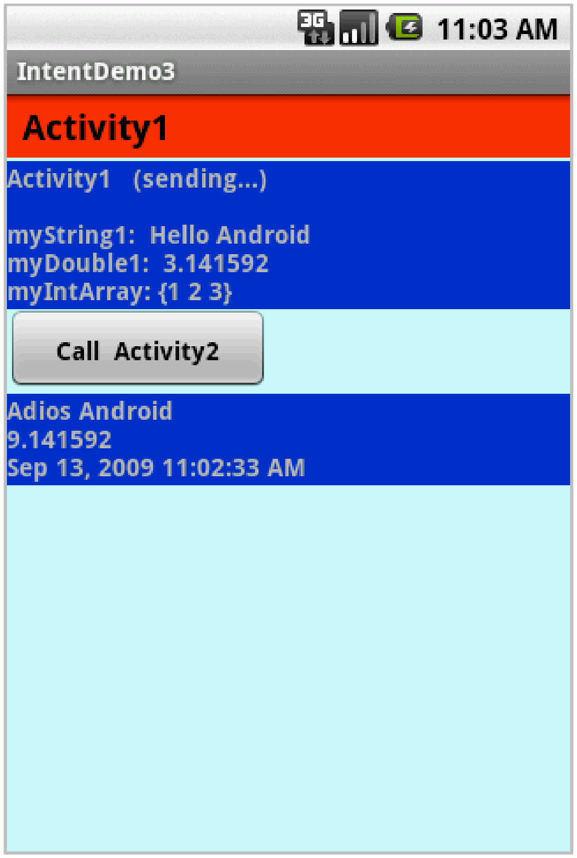
Activity2

Activity2

Activity2 (receiving...)

myString1: Hello Android
myDouble1: 3.141592
myIntArray1: { 1 2 3 }

CallBack Activity1



11:03 AM

IntentDemo3

Activity1

Activity1 (sending...)

myString1: Hello Android
myDouble1: 3.141592
myIntArray: {1 2 3}

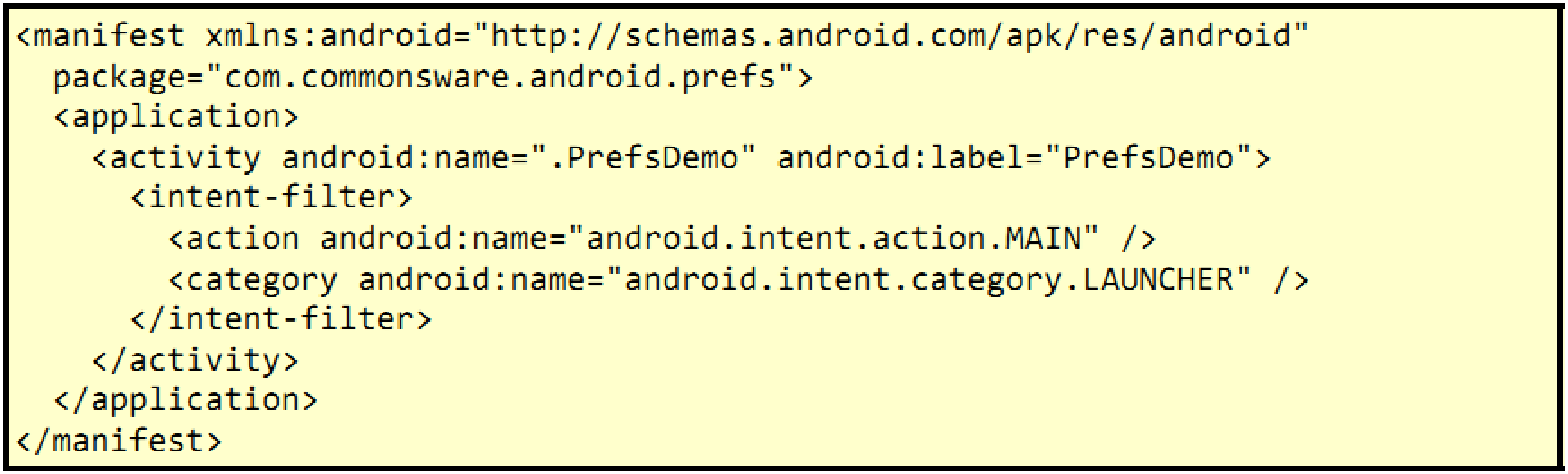
Call Activity2

Adios Android
9.141592
Sep 13, 2009 11:02:33 AM

Intent-Filter

- All Android components that wish to be notified via intents must declare intent filters, so Android knows which intents should go to that component.

- Need to add intent-filter elements to the AndroidManifest.xml file.

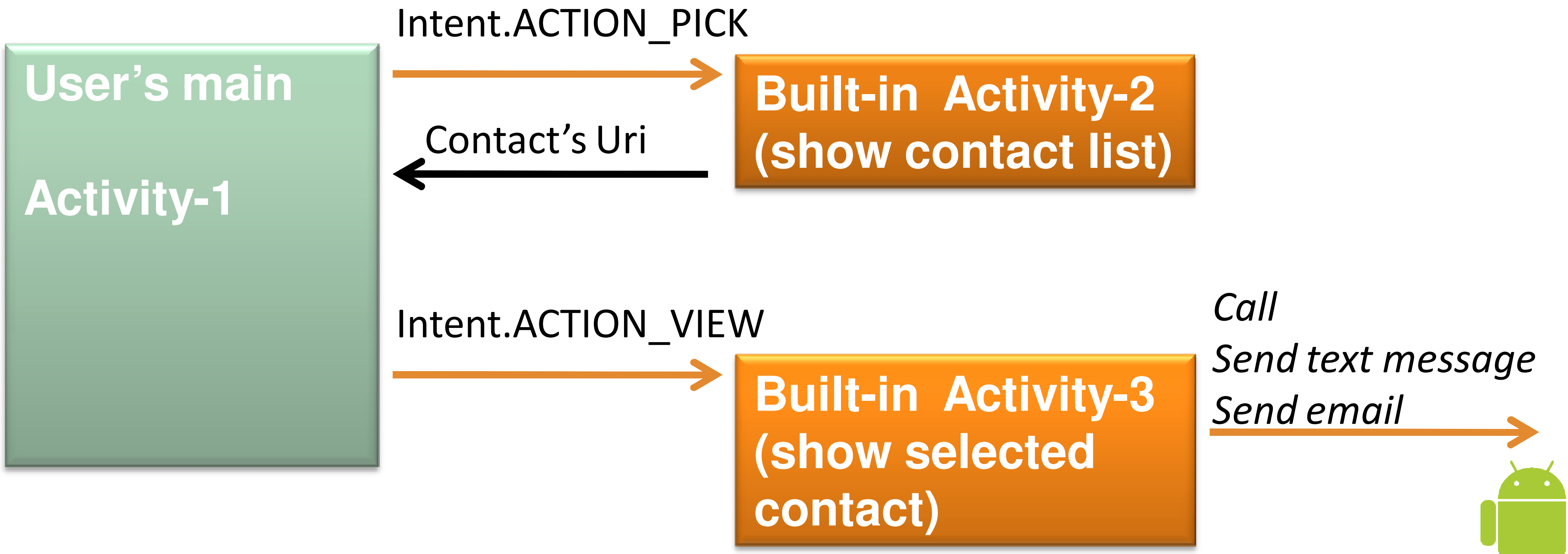


```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
  package="com.commonsware.android.prefs">
  <application>
    <activity android:name=".PrefsDemo" android:label="PrefsDemo">
      <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
    </activity>
  </application>
</manifest>
```

Intents

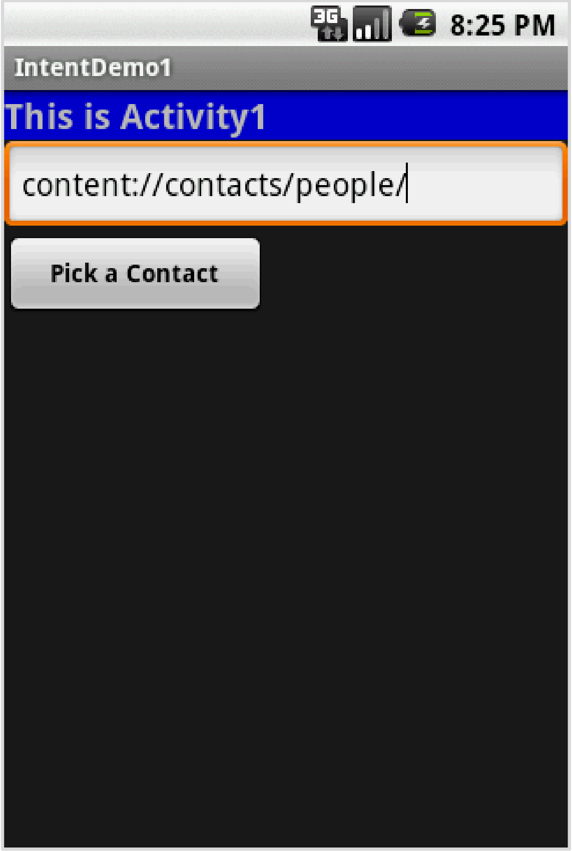
Example1. Let's play golf - Call for a tee-time.

1. Show all contacts and pick a particular one (*Intent.ACTION_PICK*).
2. For a successful interaction the main-activity accepts the returned URI identifying the person we want to call (*content://contacts/people/n*).
3. 'Nicely' show the selected contact's entry allowing calling, texting, emailing actions (Intent.ACTION_VIEW).



Intents

Example1. Let’s play golf - *Call for a tee-time*.



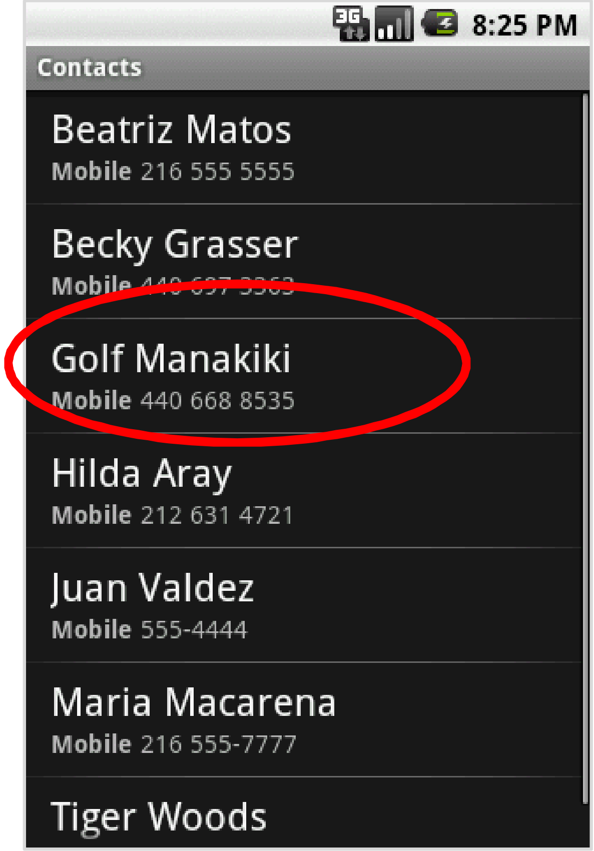
8:25 PM

IntentDemo1

This is Activity1

content://contacts/people/

Main Activity



8:25 PM

Contacts

Beatriz Matos
Mobile 216 555 5555

Becky Grasser
Mobile 440 697 3363

Golf Manakiki
Mobile 440 668 8535

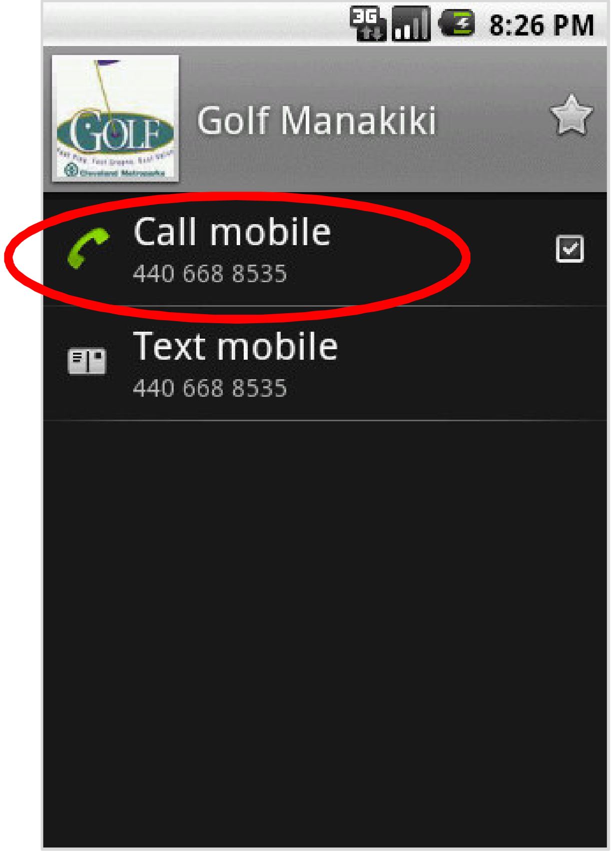
Hilda Aray
Mobile 212 631 4721

Juan Valdez
Mobile 555-4444

Maria Macarena
Mobile 216 555-7777

Tiger Woods

Intent.ACTION_PICK



8:26 PM

Golf Manakiki

Call mobile ☑
440 668 8535

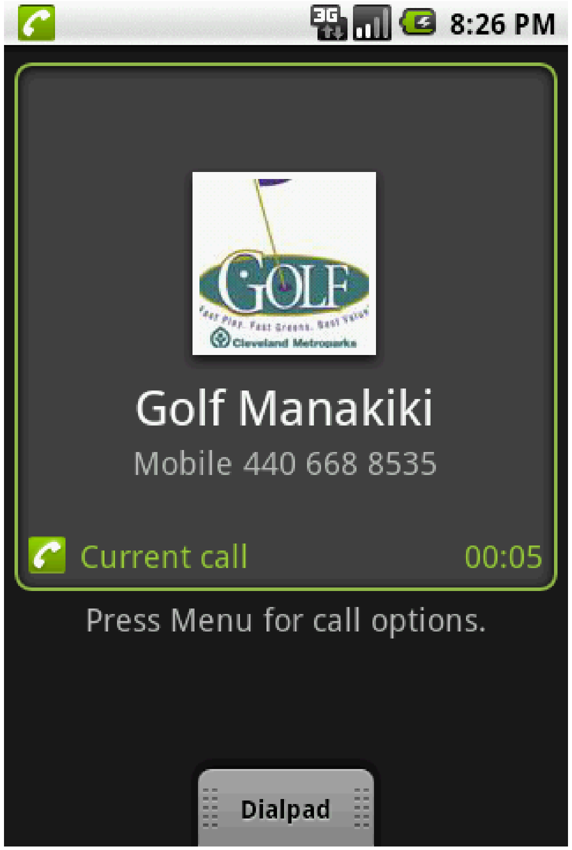
Text mobile
440 668 8535

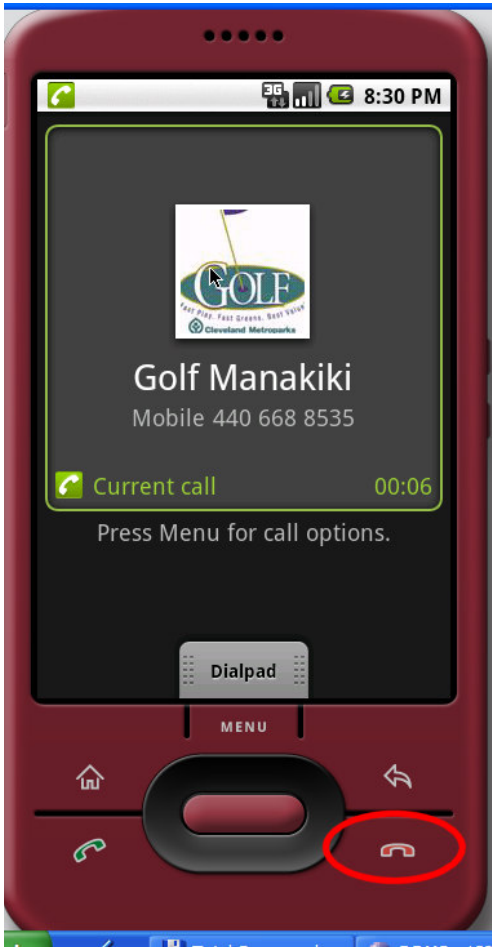
Intent.ACTION_VIEW

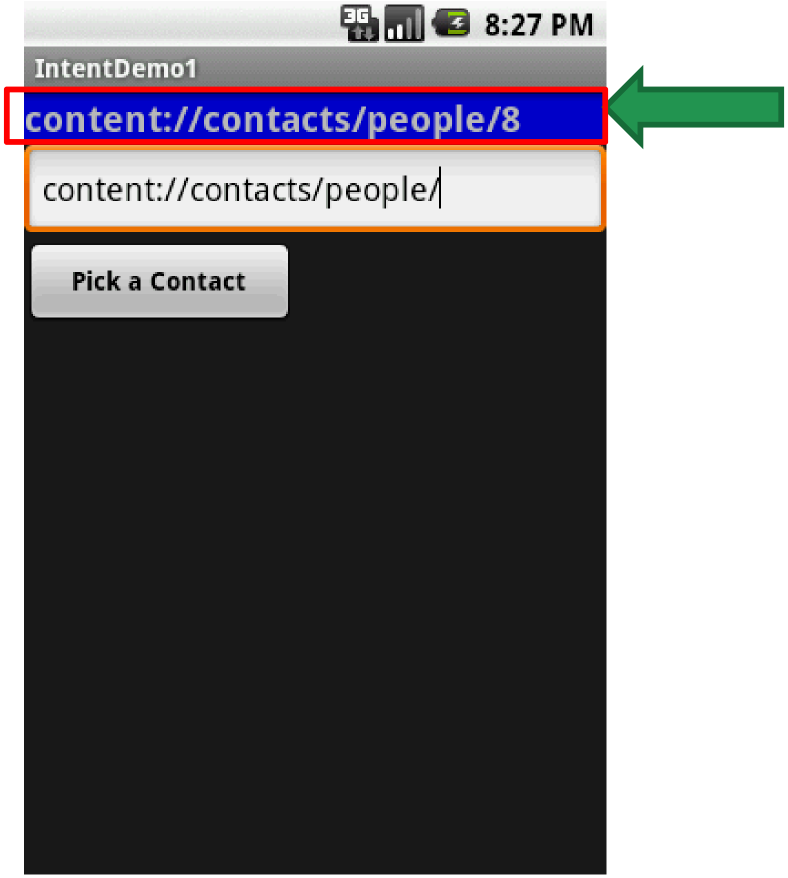
Cont.

Intents

Example1 (cont.) Let's play golf - *Call for a tee-time*







content://contacts/people/8

content://contacts/people/

Place the call

Terminate the call

Selected contact's URI

Intents

Example1. *Calling a sub-activity, receiving results.*

```
//IntentDemo2_Intent: making a phone call
//receiving results from a sub-activity
package cis493.intents;
import android.app.Activity;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.*;

public class IntentDemo2 extends Activity {
    TextView label1;
    EditText text1;
    Button   btnCallActivity2;
```

Intents

Example1. *Calling a sub-activity, receiving results.*

```
@Override
   public void onCreate(Bundle savedInstanceState) {
   super.onCreate(savedInstanceState);
   try {
           setContentView(R.layout.main);
           label1 = (TextView)findViewById(R.id.label1);
           text1 = (EditText)findViewById(R.id.text1);

           btnCallActivity2 = (Button)findViewById(R.id.btnPickContact);
           btnCallActivity2.setOnClickListener(new ClickHandler());
    }
    catch (Exception e) {
            Toast.makeText(getBaseContext(),
            e.getMessage(), Toast.LENGTH_LONG).show();
    }
 }//onCreate
```

Intents

Example1. *Calling a sub-activity, receiving results.*

```
private class ClickHandler implements OnClickListener {
    @Override
    public void onClick(View v) {
        try {
             // myData refer to: content://contacts/people/
             String myData = text1.getText().toString();

             //you may also try ACTION_VIEW instead
             Intent myActivity2 = new Intent(Intent.ACTION_PICK,
                                         Uri.parse(myData));
            // start myActivity2.
            // Tell it that our requestCodeID (or nickname) is 222
            startActivityForResult(myActivity2, 222);

            // Toast.makeText(getApplicationContext(),
            //                 "I can't wait for you", 1).show();
        }
        catch (Exception e) {
            label1.setText(e.getMessage());
    }
    }//onClick
  }//ClickHandler
```

Intents

Example1. *Calling a sub-activity, receiving results.*

```
@Override
protected void onActivityResult(int requestCode,
                                int resultCode,
                                Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    try {
        // use requestCode to find out who is talking back to us
        switch (requestCode){
            case (222): {
                // 222 is our friendly contact-picker activity
                if (resultCode == Activity.RESULT_OK) {
                    String selectedContact = data.getDataString();
                    // it will return an URI that looks like:
                    // content://contacts/people/n
                    // where n is the selected contacts' ID
                    label1.setText(selectedContact.toString());

                    //show a 'nice' screen with the selected contact
                    Intent myAct3 = new Intent (Intent.ACTION_VIEW,
                                          Uri.parse(selectedContact));
                    startActivity(myAct3);
                }
```

Listener

Intents

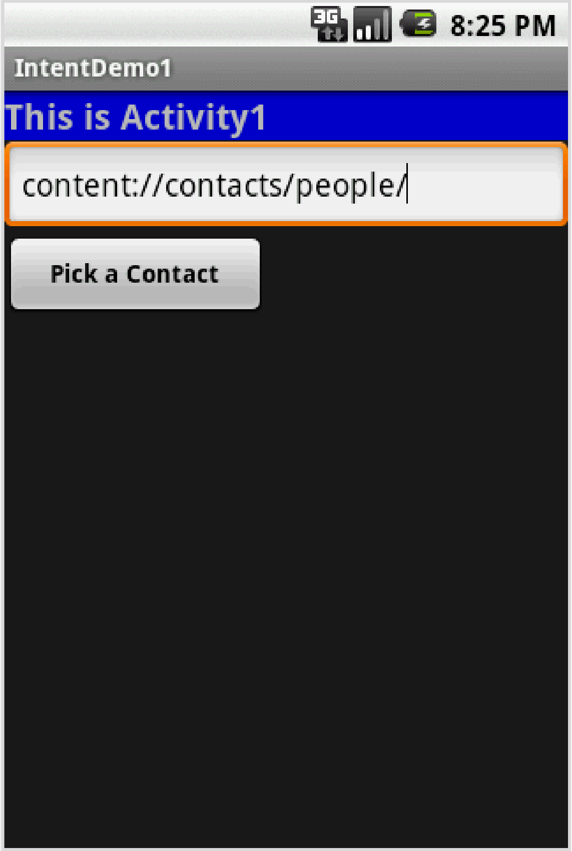
Example1. *Calling a sub-activity, receiving results.*

```
                 else {
                     //user pressed the BACK button
                     label1.setText("Selection CANCELLED "
                                    + requestCode + " " + resultCode);
                     }
                     break;
                 }
        }//switch
    }
    catch (Exception e) {
        Toast.makeText(getBaseContext(), e.getMessage(),
                       Toast.LENGTH_LONG).show();
    }
  }// onActivityResult

}//IntentDemo2
```

Intents

Example1. *Calling a sub-activity, receiving results.*

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent“ >
     <TextView
          android:id="@+id/label1"
          android:layout_width="fill_parent"
          android:layout_height="wrap_content"
          android:background="#ff0000cc"
          android:text="This is Activity1"
          android:textStyle="bold"
          android:textSize="20sp“/>
     <EditText
          android:id="@+id/text1"
          android:layout_width="fill_parent"
          android:layout_height="54px"
          android:text="content://contacts/people/"
          android:textSize="18sp”  />
     <Button
          android:id="@+id/btnPickContact"
          android:layout_width="149px"
          android:layout_height="wrap_content"
          android:text="Pick a Contact"
          android:textStyle="bold“ />
</LinearLayout>
```



8:25 PM

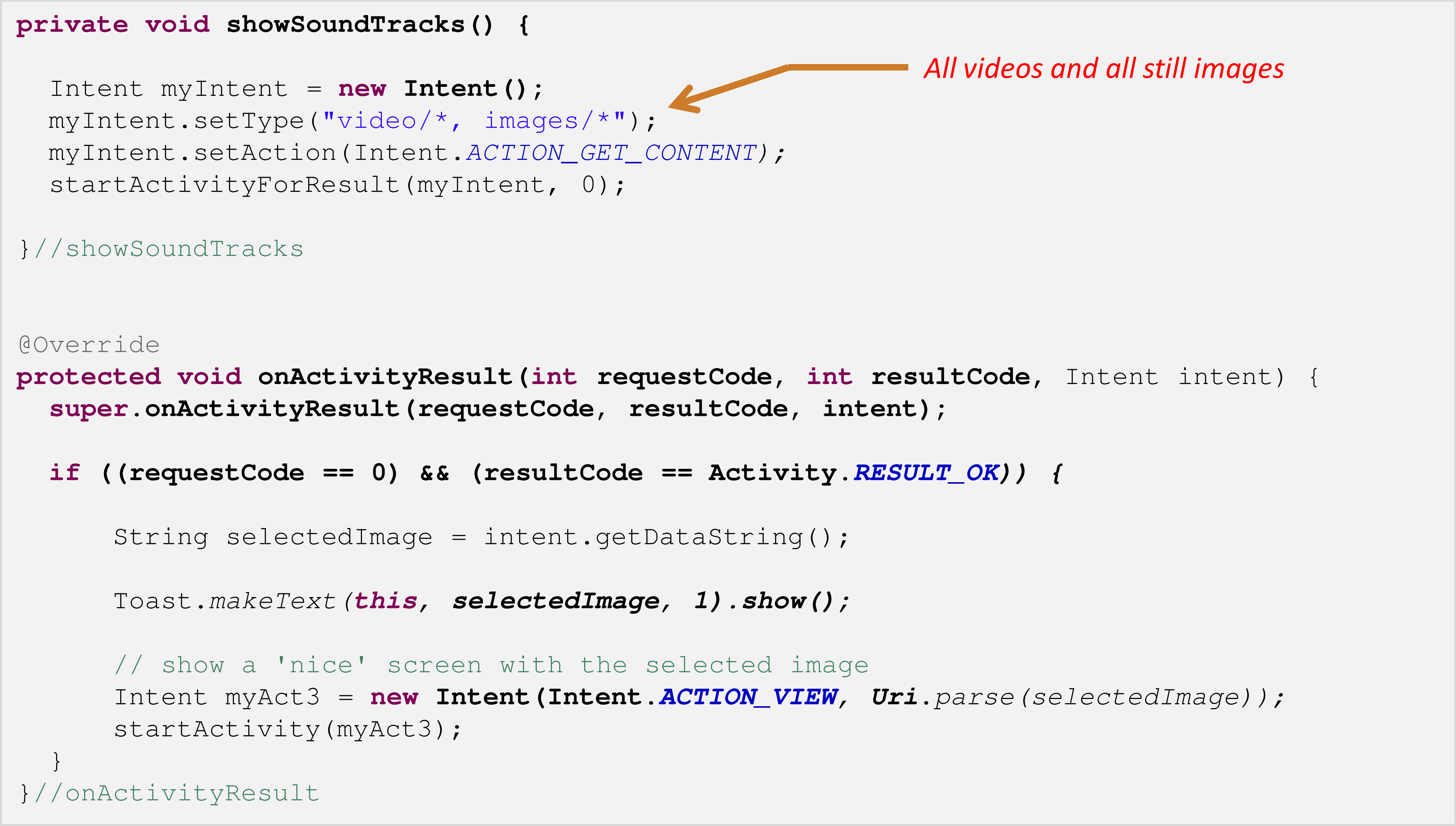
IntentDemo1

This is Activity1

content://contacts/people/

Intents

Example2. Showing Pictures and Video - Calling a sub-activity, receiving results.

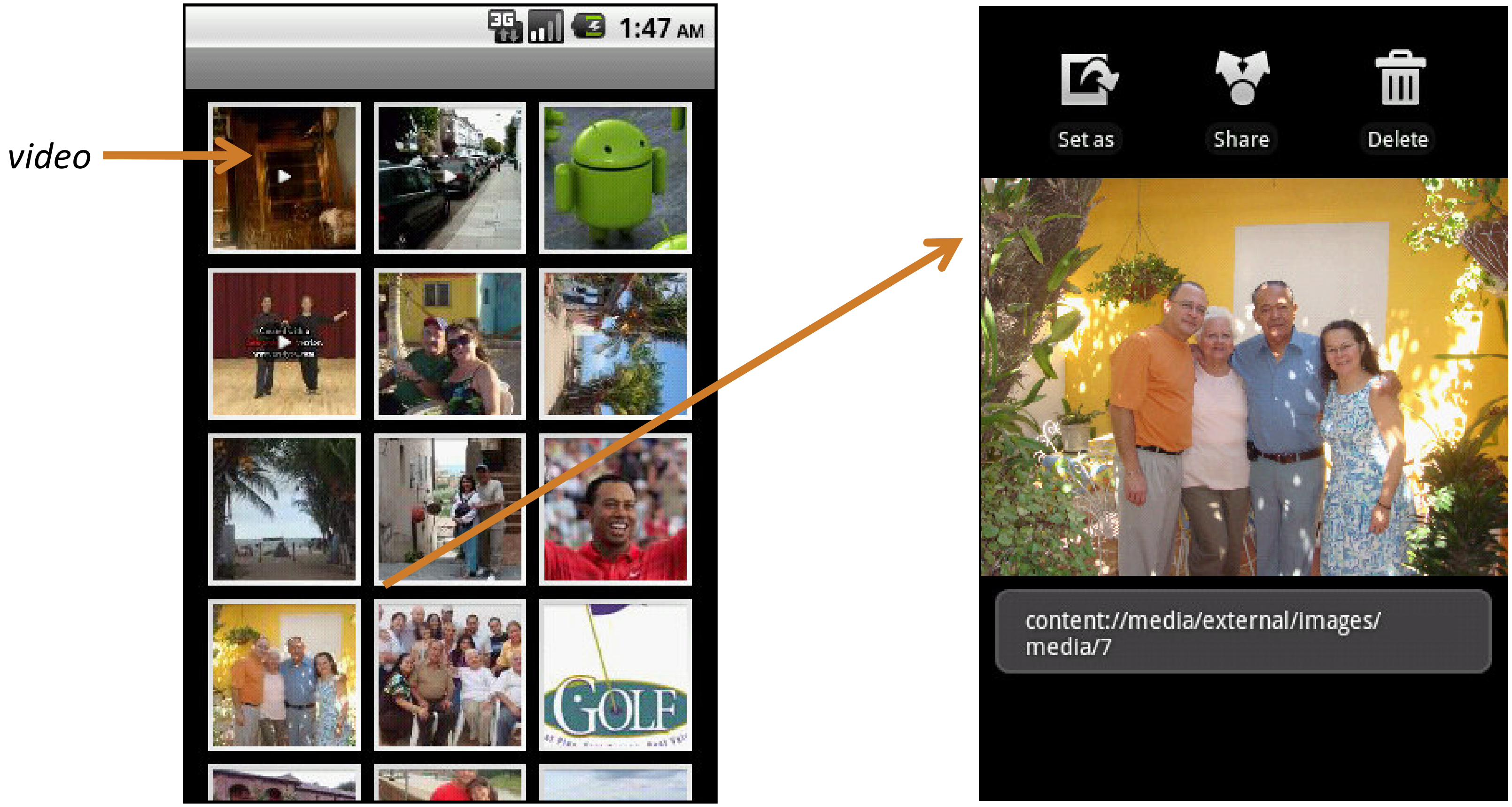


```
private void showSoundTracks() {

  Intent myIntent = new Intent();
  myIntent.setType("video/*, images/*");
  myIntent.setAction(Intent.ACTION_GET_CONTENT);
  startActivityForResult(myIntent, 0);

}//showSoundTracks


@Override
protected void onActivityResult(int requestCode, int resultCode, Intent intent) {
  super.onActivityResult(requestCode, resultCode, intent);

  if ((requestCode == 0) && (resultCode == Activity.RESULT_OK)) {

      String selectedImage = intent.getDataString();

      Toast.makeText(this, selectedImage, 1).show();

      // show a 'nice' screen with the selected image
      Intent myAct3 = new Intent(Intent.ACTION_VIEW, Uri.parse(selectedImage));
      startActivity(myAct3);
  }
}//onActivityResult
```

All videos and all still images

Intents

Example2. Showing Pictures and Video - Calling a sub-activity, receiving results.



content://media/external/images/media/7

End of Lecture

Questions ?