

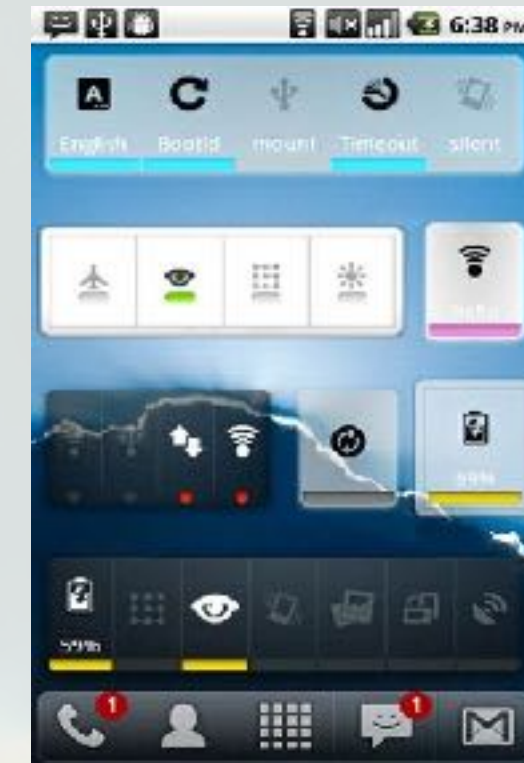
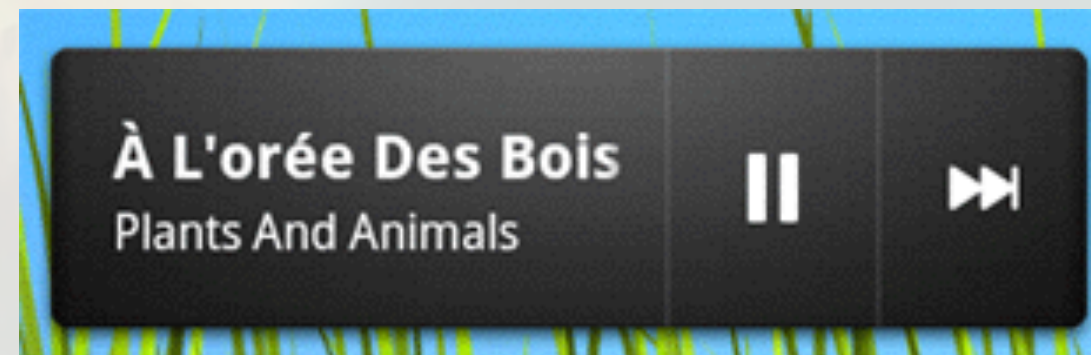
# Widgets

[Renault@lrde.epita.fr](mailto:Renault@lrde.epita.fr)



# Description

367



## Miniature Application

- embedded in other application (homescreen for instance)
- that receive periodically updates
- that offer the best GUI for BroadcastReceiver
  - ▶ One can define a widget that count the number of received SMS

# Non-technical details

368



**A widget may be associated to an activity that helps its configuration**

- 🔊 Refresh rate
- 🔊 Appearance
- 🔊 Informations to display



**The widget must be not too small or too large**



**For the installation**

- 🔊 Application -> Widgets -> drag-and-drop on the home screen



**Multiple instances of the same widgets can exist at the same time!**

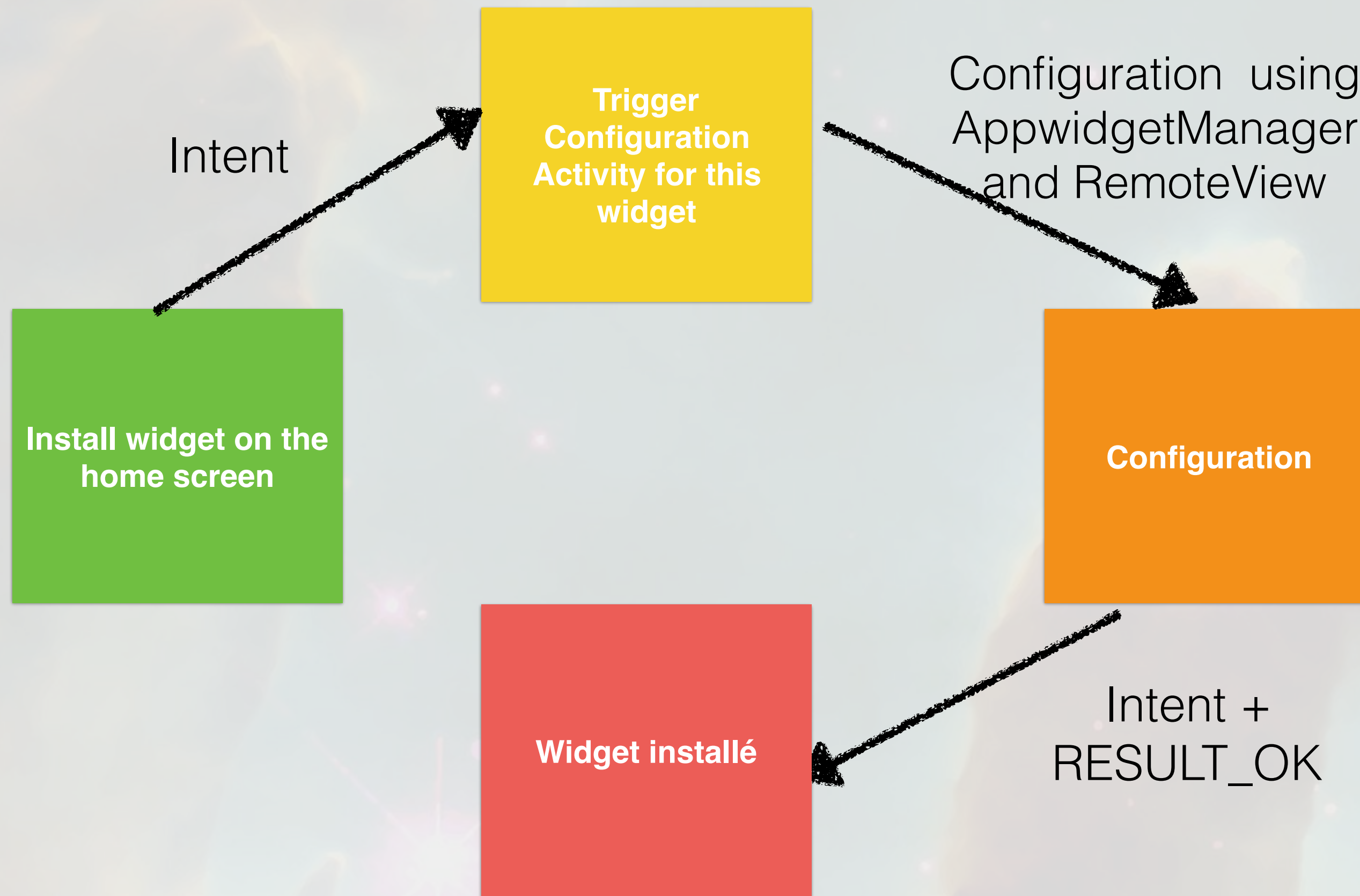
# Widgets Lifecycle



## Widgets works by callbacks

<b>onEnabled()</b>	Called when installed on the HomeScreen ( <u>first instance only</u> )
<b>onDeleted()</b>	Called when a widget is removed from the HomeScreen
<b>onDisabled()</b>	Called when the last widget is removed from the HomeScreen
<b>onUpdate()</b>	Called foreach updates. An identifier helps to detect which instance is concerned

# Lifecycle Details



# Defining a widget

371

## Modify AndroidManifest.xml

```
<receiver android:name="MyExampleAppWidgetProvider" >
  <intent-filter>
    <action android:name="
      android.appwidget.action.APPWIDGET_UPDATE" />
  </intent-filter>
  <meta-data android:name="android.appwidget.provider"
    android:resource="@xml/my_example_appwidget_info" />
</receiver>
```

-  MyExampleAppWidgetProvider:
  - ▶ **Entry point for the widget**
-  xml/my\_example\_appwidget\_info:
  - ▶ **configuration file for the widget**

# Widget's Configuration

372



## Configuration File

```
<?xml version="1.0" encoding="utf-8"?>
<appwidget-provider
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:minWidth="40dp"
  android:minHeight="40dp"
  android:updatePeriodMillis="86400000"
  android:initialLayout="@layout/my_example_appwidget"
  android:resizeMode="horizontal|vertical"
  android:widgetCategory="home_screen|keyguard">
</appwidget-provider>
```



Updates cannot be less than 10 000 milliseconds!

# Defining the Main class

373



## Option 1:

- Use a BroadcastReceiver



## Option 2:

- Use an AppWidgetProvider

- ▶ Facilities to build widgets
- ▶ Parse automatically relevant fields of the Intent
- ▶ Call hook methods with extras
- ▶ Load the GUI

```
public class MyExampleAppWidgetProvider
    extends AppWidgetProvider {
}
```

- Note: this widgets does nothing except loading its UI



# Widget's Configuration Activity (1/2)

374

## Modify AndroidManifest.xml

```
android:configure=  
    "com.example.admin.widgetapplication.MainActivity"
```

 This activity will be triggered automatically when installing the widget

 Think to declare this activity sensible to widget configuration in AndroidFile.xml

```
<intent-filter>  
    <action android:name=  
        "android.appwidget.action.APPWIDGET_CONFIGURE" />  
</intent-filter>
```

# Widget's Configuration Activity (2/2)



## Get the ID of the widget

```
private int mAppWidgetId;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Intent intent = getIntent();
    Bundle extras = intent.getExtras();
    if (extras != null) {
        mAppWidgetId = extras.getInt(
            AppWidgetManager.EXTRA_APPWIDGET_ID,
            AppWidgetManager.INVALID_APPWIDGET_ID);
    }
}
```

# How to instanciate a new Widget?



## Create a RemoteView and specify the layout

```
AppWidgetManager appWidgetManager =  
    AppWidgetManager.getInstance(getApplicationContext());  
  
RemoteViews views = new RemoteViews(  
    getApplicationContext().getPackageName(),  
    R.layout.my_example_appwidget_custom);  
  
appWidgetManager.updateAppWidget(mAppWidgetId, views);
```



## Notify the widget that the configuration is done and finish the configuration activity

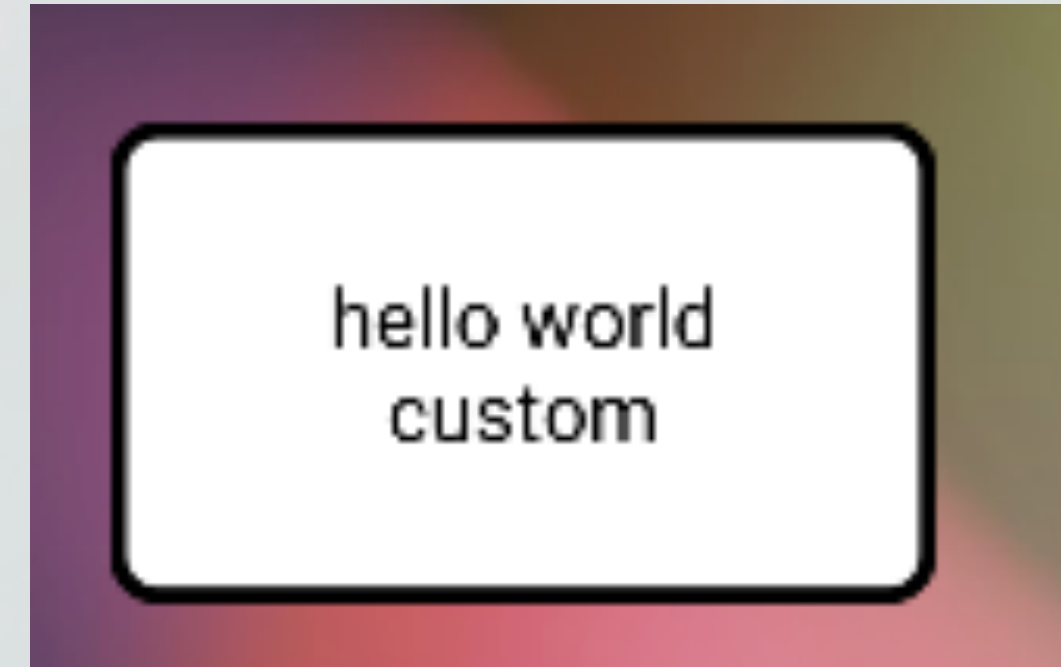
```
Intent resultValue = new Intent();  
resultValue.putExtra(AppWidgetManager.EXTRA_APPWIDGET_ID,  
    mAppWidgetId);  
setResult(RESULT_OK, resultValue);  
finish();
```

# How to build friendly widgets?



## Rounded Widgets are more aesthetic

🔧 Create a file [res/drawable/rounded.xml](#)



```
<?xml version="1.0" encoding="UTF-8"?>
<shape
  xmlns:android="http://schemas.android.com/apk/res/android">
  <solid android:color="#FFFFFF" />
  <stroke android:width="3dip" android:color="#B1BCBE" />
  <corners android:radius="10dip" />
  <padding android:left="0dip" android:top="0dip"
    android:right="0dip" android:bottom="0dip" />
</shape>
```

🔧 [and set this as background for the widget layout!](#)

# Summary



## Widgets are often a betterment for you application

- Quick access to informations
  - ▶ **Can be installed in the HomeScreen**
- Can handle buttons to trigger other android components
- Mix well with BroadcastReceivers
  - ▶ **Easy to build a counter for some system events**



## Guidelines

- ( $70-n*30$ ) for a cell in the home screen
- Android is providing existing shapes for widgets (among the others)



