

Renault@lrde.epita.fr







Quick Tour of Android

Renault@lrde.epita.fr



Some Statistics


 **Nearly 86 % of all new smartphones (1st quarter 2018)**

 **2 billion monthly active Android users**

 **Smartphones, TV, cars, tablets, watches**

 **3 800 000 applications**

 94.5 % free

 5.5% paid



 **30 % of Android users are Millennials**



Brief History



2003: Android Inc. by A. Rubin



-  A. Rubin had previously worked for Apple and MSN
-  Idea: create an Operating System for digital camera/phone



2005: Google acquired Android for \$50 million



2007: Open Handset Consortium Alliance

-  35 technology and mobile company: Google, Samsung ... but not Blackberry, Nokia or Microsoft
-  Compete against Apple iPhone





2008: Android SDK-1 on T-mobile G1



Brief History



2010: Android is profitable for Google

-  Introduction of Nexus One and S
-  Samsung introduces the first Android Tablet



2010-2011: TVs, autoradio (Parrot), Netbooks



2012: Google Play

-  Merge Google Music and Google Market
-  MyPlay first Gaming Tablet (DEA)



2014: Wearable / Auto



2015: Brillo Project / Weave Protocols

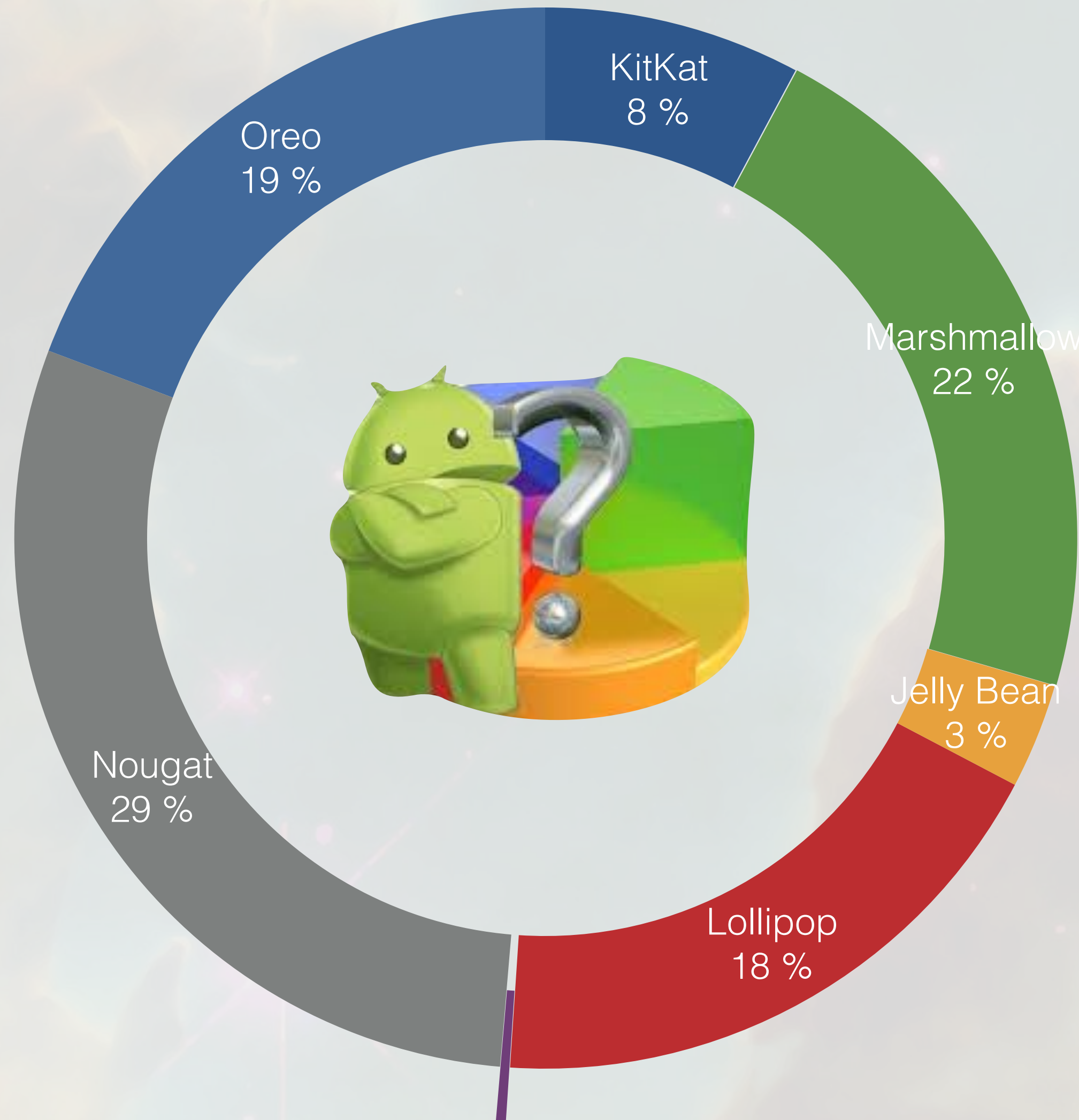
Bugdroïd: a tribute to Gauntlet: the 3rd Encounter



Versions

	API Level	Version	Features	Logo
April'09	1 - 3	Cupcake	Widgets	
September'09	4	Donut	Gesture	
October'09	5 -7	Eclair	Multitouch	
May'10	8	Froyo	Pinch to zoom	
Décember'10	9 - 10	Gingerbread	NFC, 4G	
July'11	11 - 13	Honeycomb	Multitask	
December'11	14 -15	Icecream	Virtual buttons	
July'12	16 -17	Jelly Bean	Notifications (better)	
October'13	19	Kitkat	Better Ram usage	
June'14	21	Lollipop	Compilation, Volta, new UI	
October'15	23	Marshmallow	Autonomy	
August'16	24	Nougat	Performances	
August'17	26	Oreo	Android Go	
March'18	28	Pie	Adaptative battery	

Android OS Distribution



Royalties, Licences & Open-Source



Microsoft makes money with Android

- \$5 to 15\$ for each phone
- 300 Software Patents



Licenses

- Gnu General Public License, Version 2.0

▶ For the Linux Kernel

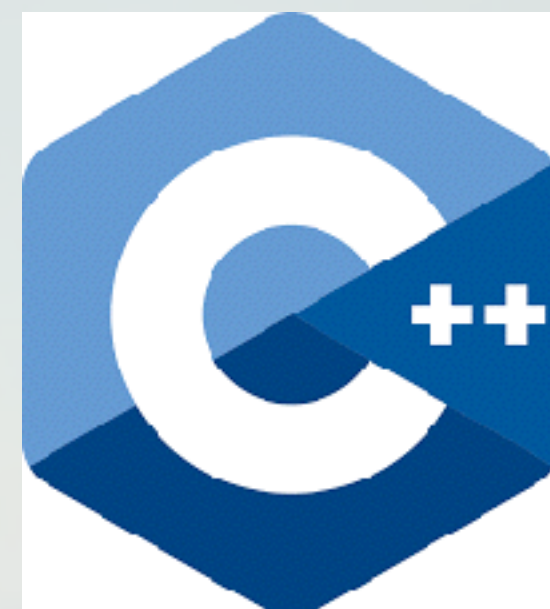
- Apache Software License, Version 2.0

▶ All that is not Kernel

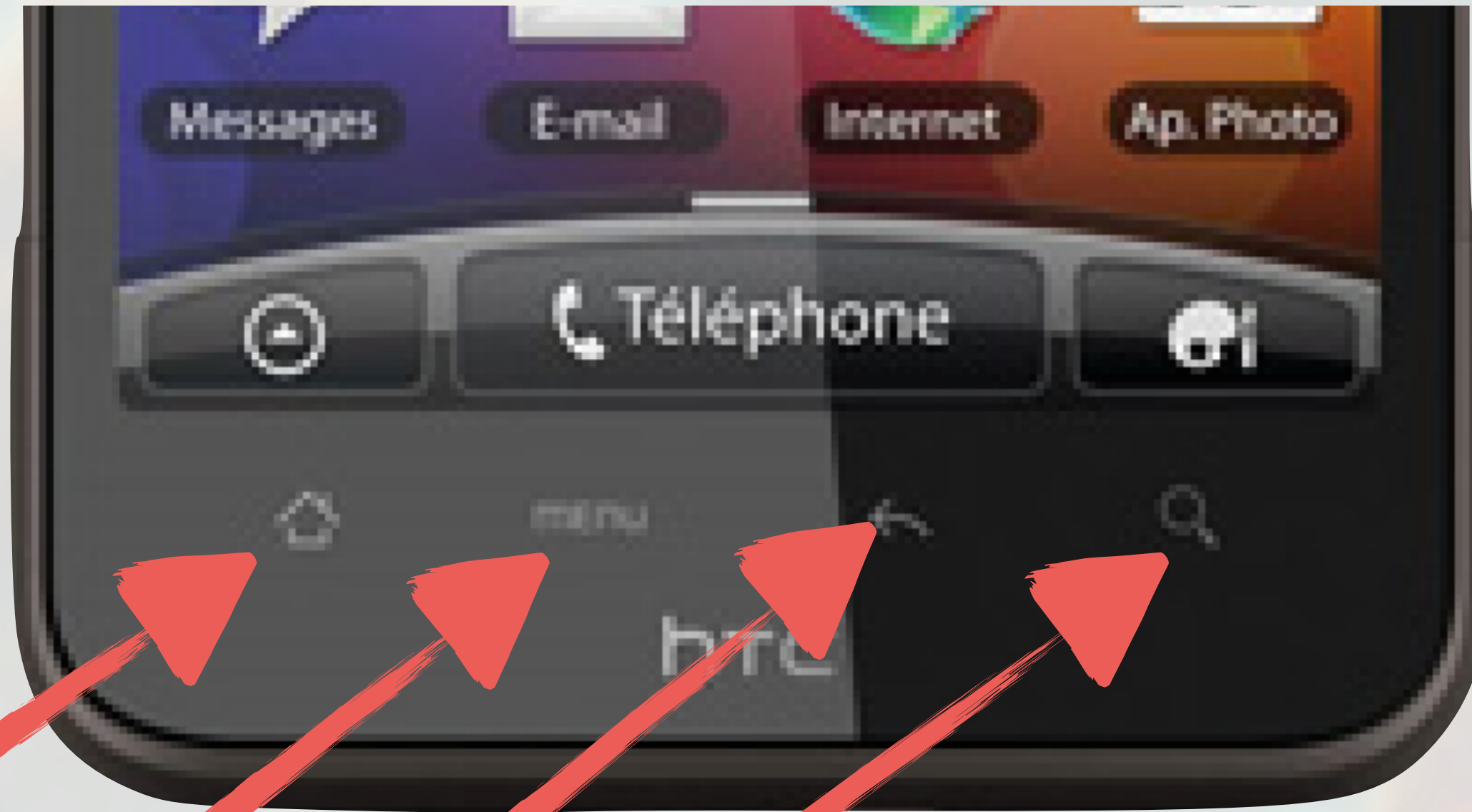


Developpement Environment

- Java, XML, C++, ...



Android Phone



HOME: back to the home

MENU: optional menu

CANCEL: discard current screen

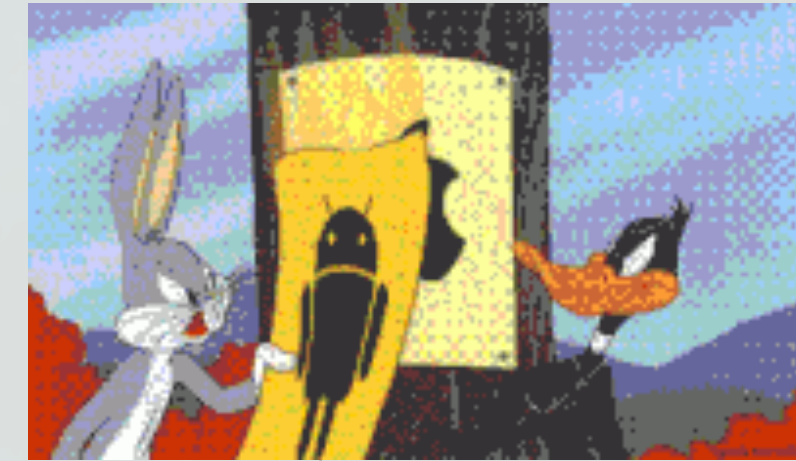
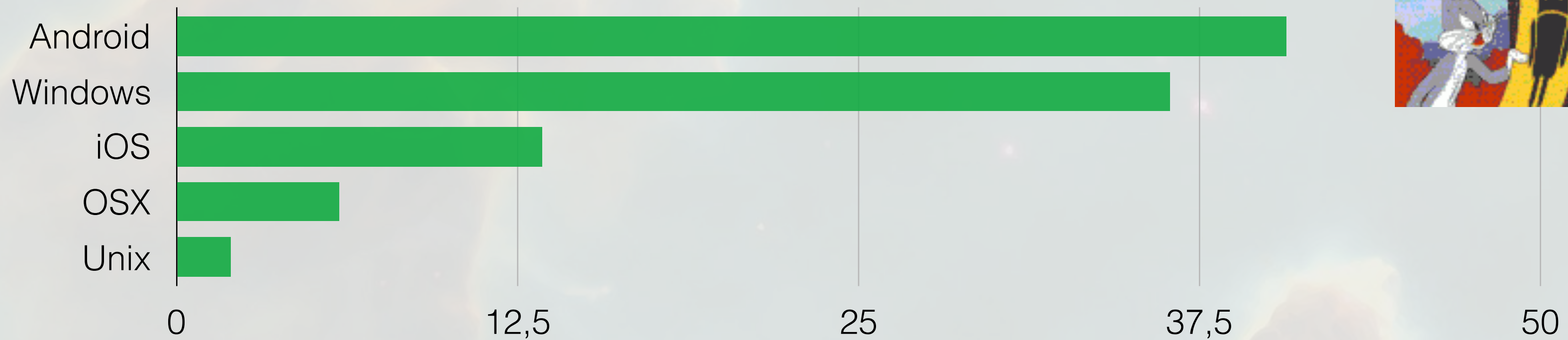
SEARCH: lookup



Summary



OS Distribution - world wide marketshare

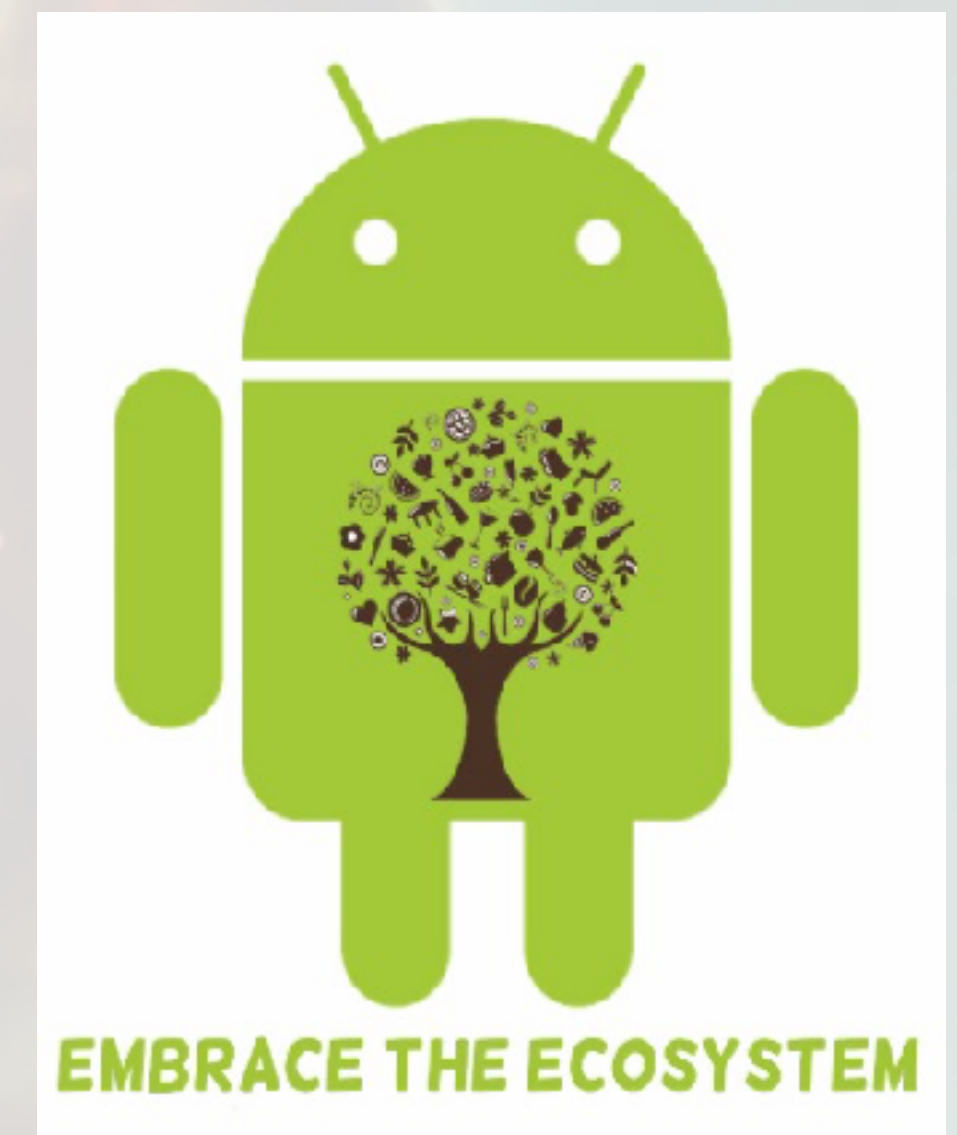


A lot of phones to handle

- Different API versions
- Different hardware
- ... Handle this in in single App!



Don't reinvent the wheel!





How to Develop & Deploy an Android Application?

Renault@lrde.epita.fr

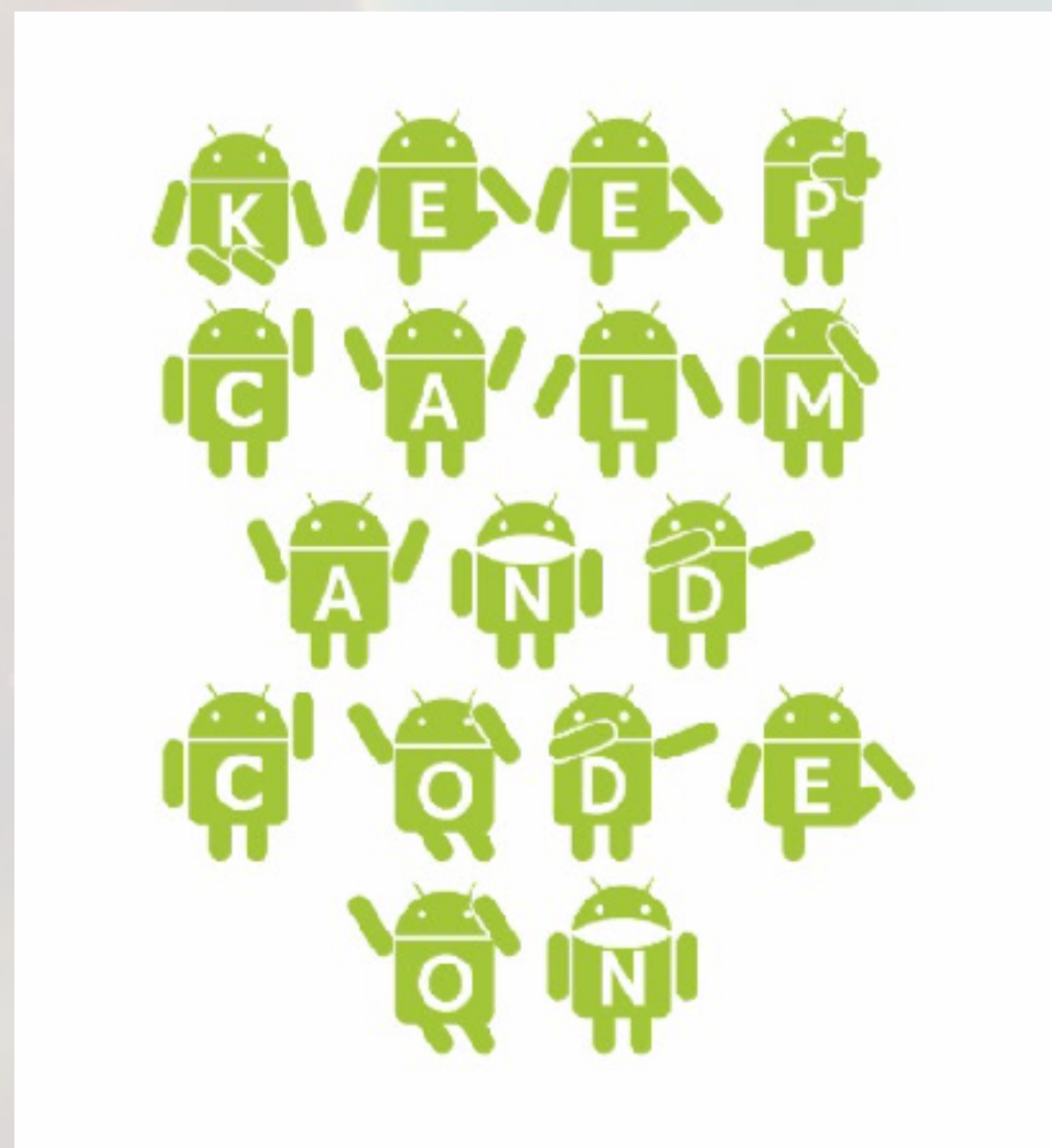


Dev. Tools

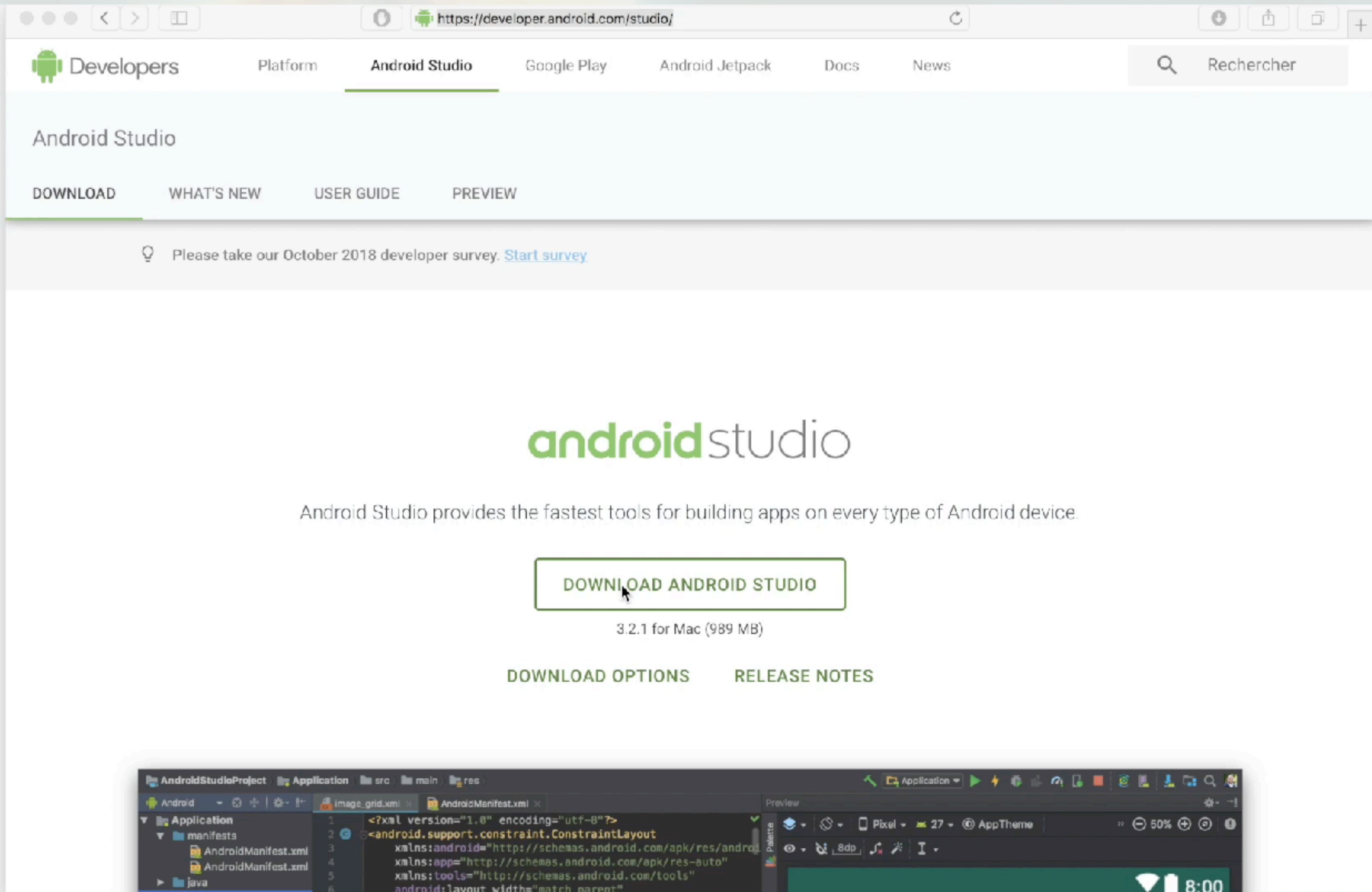


Android Studio 3.2

- Based on IntelliJ IDEA
 - ▶ **Decompiler, Debugger, Inspector, Refactor**
- Graddle based on Groovy
 - ▶ **Configure dependencies, Compilation chains**
- Lint code analyser
 - ▶ **Bug détection, security report, improvement suggestions**
- Android Virtual Device Manager
 - ▶ **Simulator**
- Android Debug Bridge for remote debugging
- Logcat: filter logs





Install & Run





Main Files and Directories





Manifests/AndroidManifest.xml

-  Permissions of the application
-  Declaration of the main components of the application

java/

-  One package for the application itself
-  One package for the tests





res/

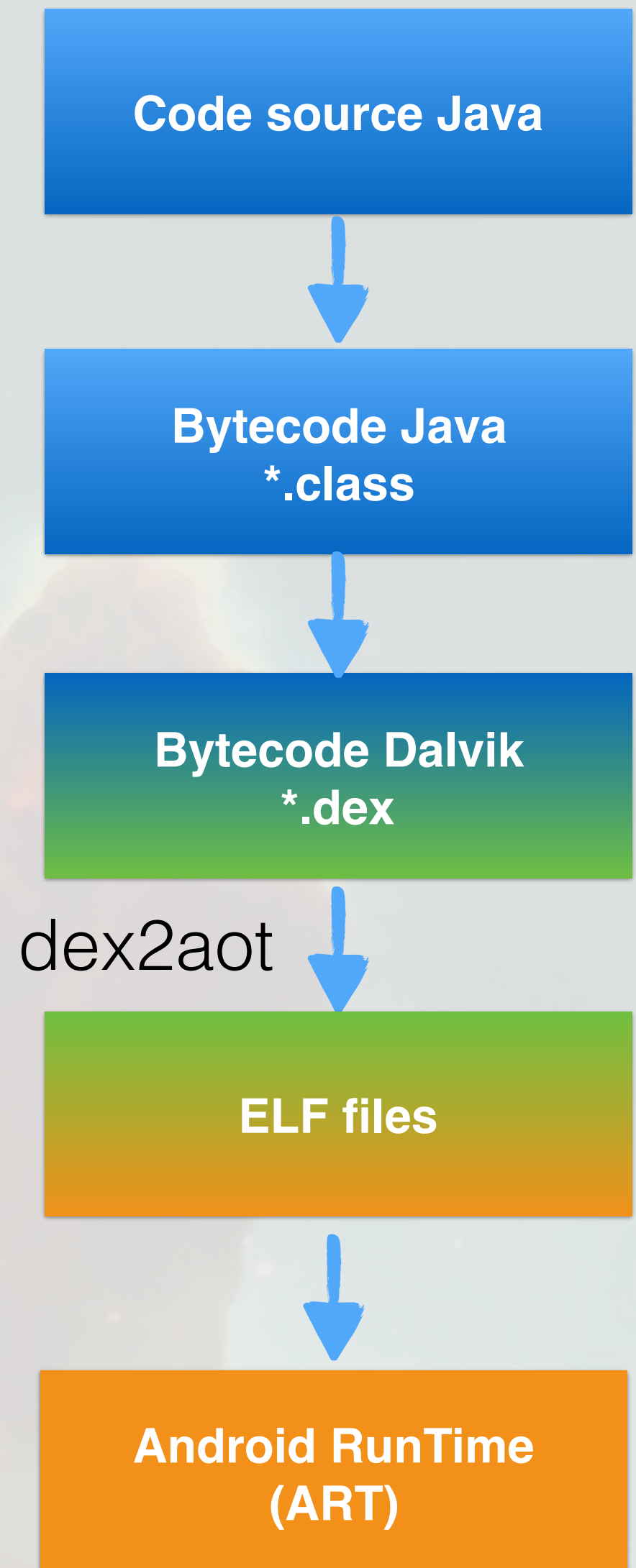
-  layout: the "views" of the application
-  menu: description of the various menus inside the application
-  values: the static values of the application (magic numbers, etc)
-  drawable, raw: images, music, etc. required by the application

ART : Android RunTime

 **Not available for oldest devices**

 **An Alternative to Dalvik**

-  dex2oat: compile the application during the installation
-  Minimise the number of Garbage Collection
-  Reduce memory consumption
-  Better performance



Run an Application



On the emulator

- 📱 Use the AVD manager to create (as many as you want) devices
- 📱 You can emulate:
 - ▶ **Phone calls, rotation/orientation, network speed, specific location, etc.**



On a device

- 📱 Connect you device to your computer
- 📱 Activate the "developper mode" in you device
 - ▶ **Tap a certain amount (7) of time on the build number**
- 📱 Activate USB deployment



Summary



You must test your application on multiple devices

- Use the AVD Manager



Click and run to have a remote run/debug



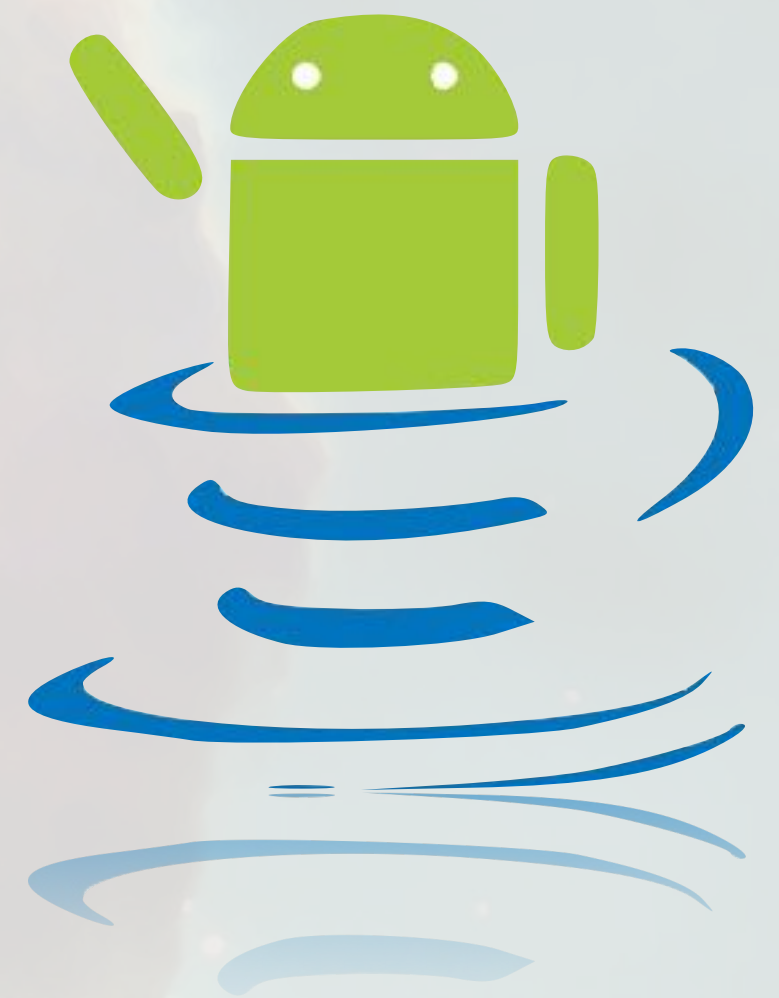
Compilation scheme

- Different from a traditional Java application
- Dalvik bytecode, Odex files, ART



Deployment

- Using the Playstore
 - ▶ **25\$ once, then 15\$ per app**
- Using your own aptoid server (debian)



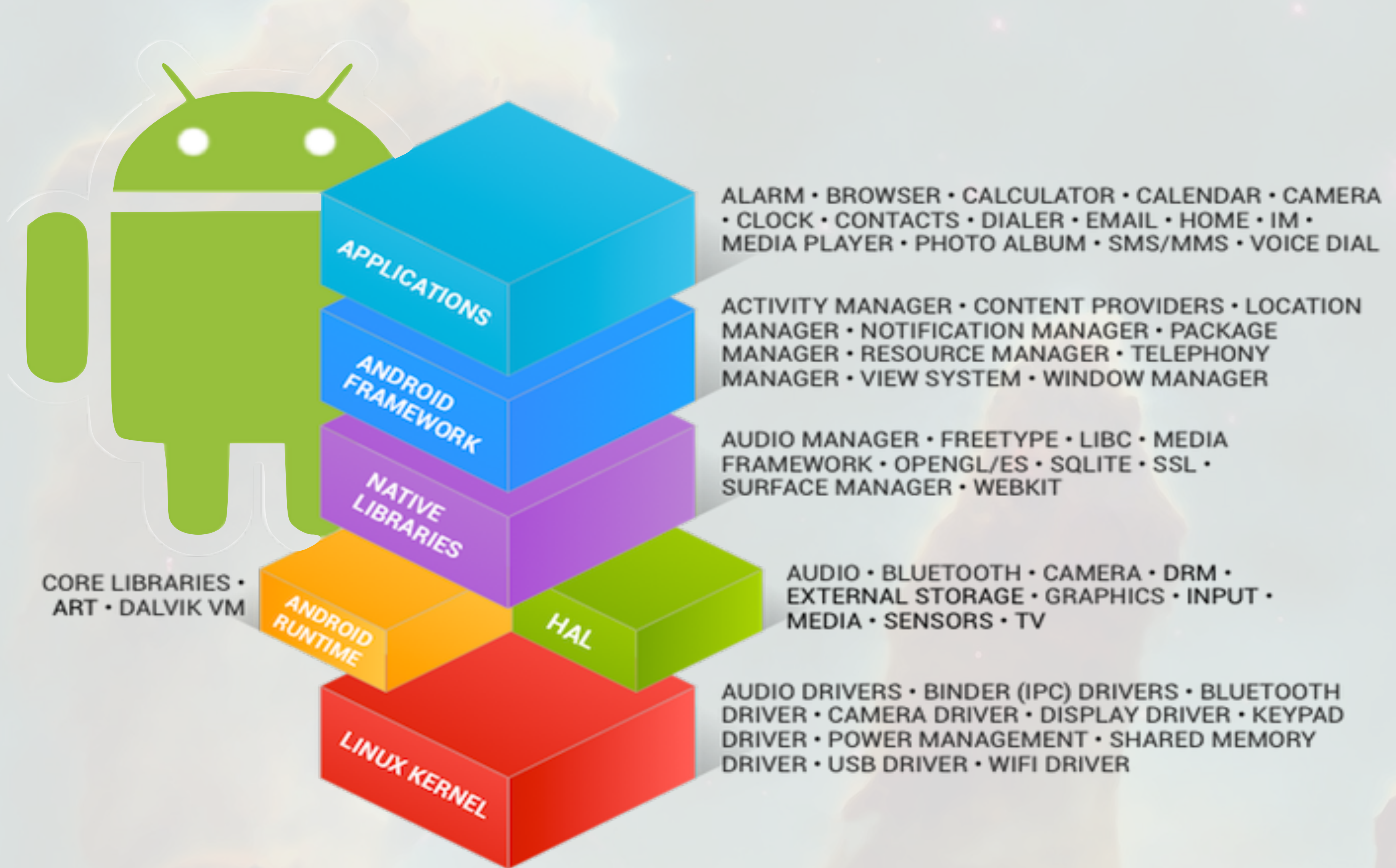


How to build a static application?

Renault@lrde.epita.fr



Android Applicative Stack





One sandbox per Android Application

- 🎧 One application = One linux user
- 🎧 One application = One process
- 🎧 One application = One virtual machine



Permissions

- 🎧 Declared in AndroidManifest.xml
- 🎧 The user may actively authorize some permissions (later in the lecture)



Two applications can share UID to exchange information

Applications & Activities



One application is composed of activities

For instance, a mail reader :

- ▶ One activity for displaying emails
- ▶ One activity for reading emails
- ▶ One activity for writing emails
- ▶ ...



An activity is:

- A simple screen with an user interface
- Must inherits from Activity class



An activity may launched from another application

- A mail reader can launch the camera

How activities are working ?

30



Model-View-Controller scheme



GUI is defined through XML files



res/layout: for views



res/menu: for the associated menus



Controller derived from the Activity class



Load the GUI



Register to events



Update the view



Model is updated by the controlling according to user actions

Building a GUI

Easy, use a WISIWYG editor

 Use simple drag-and-drops

res/layout/activity_main.xml:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent"
  android:orientation="horizontal"
  android:baselineAligned="false"
  android:weightSum="1">
```

<TextView

```
  android:layout_width="wrap_content"
  android:layout_height="635dp"
  android:textAppearance="?android:attr/textAppearanceLarge"
  android:text="My First Hello World!"
  android:id="@+id/textView"
  android:layout_weight="1.05"
  android:gravity="center_vertical|center_horizontal"
  android:textSize="100dp" />
```

```
</LinearLayout>
```



Layouts

Setup for the GUI structure

 ViewGroup implementation

Three main layout

 Linear Layout

- ▶ Elements are ordered horizontally or vertically
- ▶ A scrollbar is generated for invisible elements

 Relative Layout

- ▶ Elements are ordered according to their relative position with other elements

 Webview

- ▶ Display a webpage



```
<html>  
  <!-- web page -->  
</html>
```


Views Hierarchy



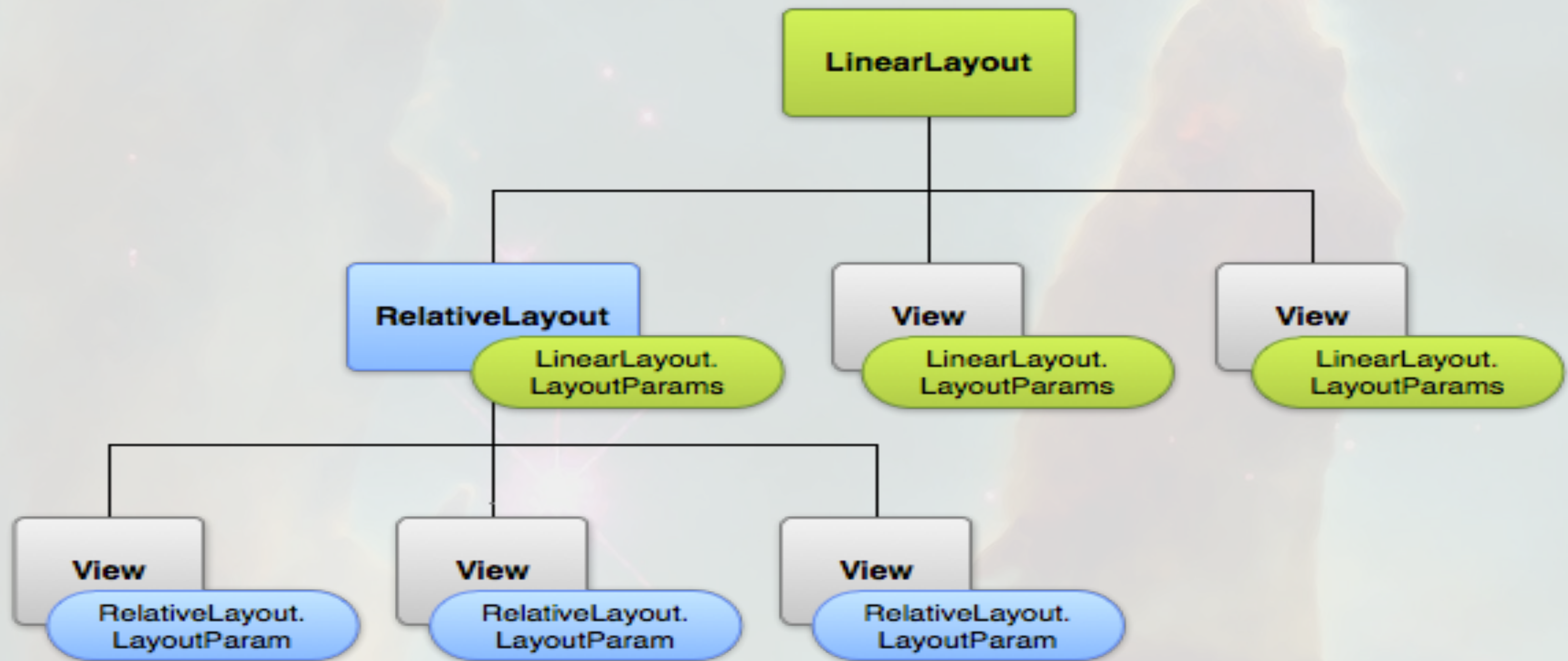
A screen can be seen as a tree view

ViewGroup

- ▶ views containers (internal nodes of the tree)

Views

- ▶ TextView, Buttons, etc. (leaf nodes of the tree)



Display a GUI



In the controller

- modify `java/.../MainActivity.java`
- Setup in the `onCreate` method
- Specify which GUI XML file we want to use

```
public class MainActivity extends Activity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
    }  
  
    // More stuff here ...  
}
```

The "mysterious" Ressource class

35



Automatically generated during the compilation

- Binding between the view and the controller



Associate an ID to all elements

- Layout
- Views
- Images
- ...

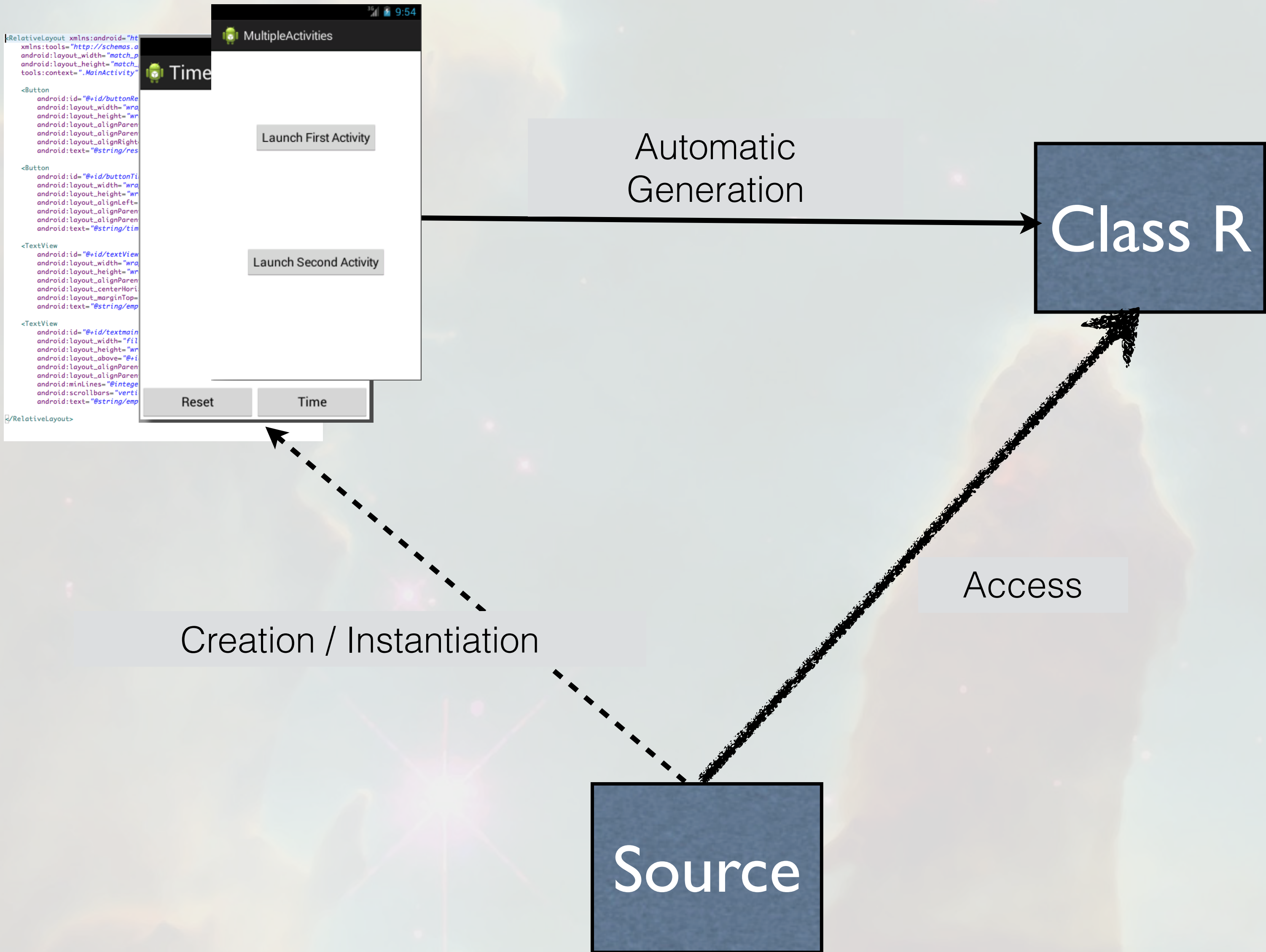


How to get the ID of an element

```
findViewById (R.id.my_fancy_object)
```

```
findViewById (R.layout.my_fancy_object)
```

The relation between the elements



Device orientation



We can build a view per orientation

- One XML for the landscape view
- One XML for the portrait view
- The two XML must have the same name

Only for static application

Summary



Overview of an Android Application

- The applicative stack
- Sandboxing



How to define a GUI

- Through XML files
- View hierarchy
- Landscape / Portrait pre-defined views



How the R class is working

- Automatic Generation of ID for each elements



How to build a dynamic application?

Renault@lrde.epita.fr



Switching between Activities



An application is composed of many activities

- One activity is a user screen



An activity can launch another activity

- ... that has been written by the developer himself/herself
- ... or that is already available on the device
- ▶ one can decide to launch the camera from its own application



Only one activity is visible at time t



How to switch between activities?

- Intents: an explicit way to communicate
- Backstack: the navigation between activities

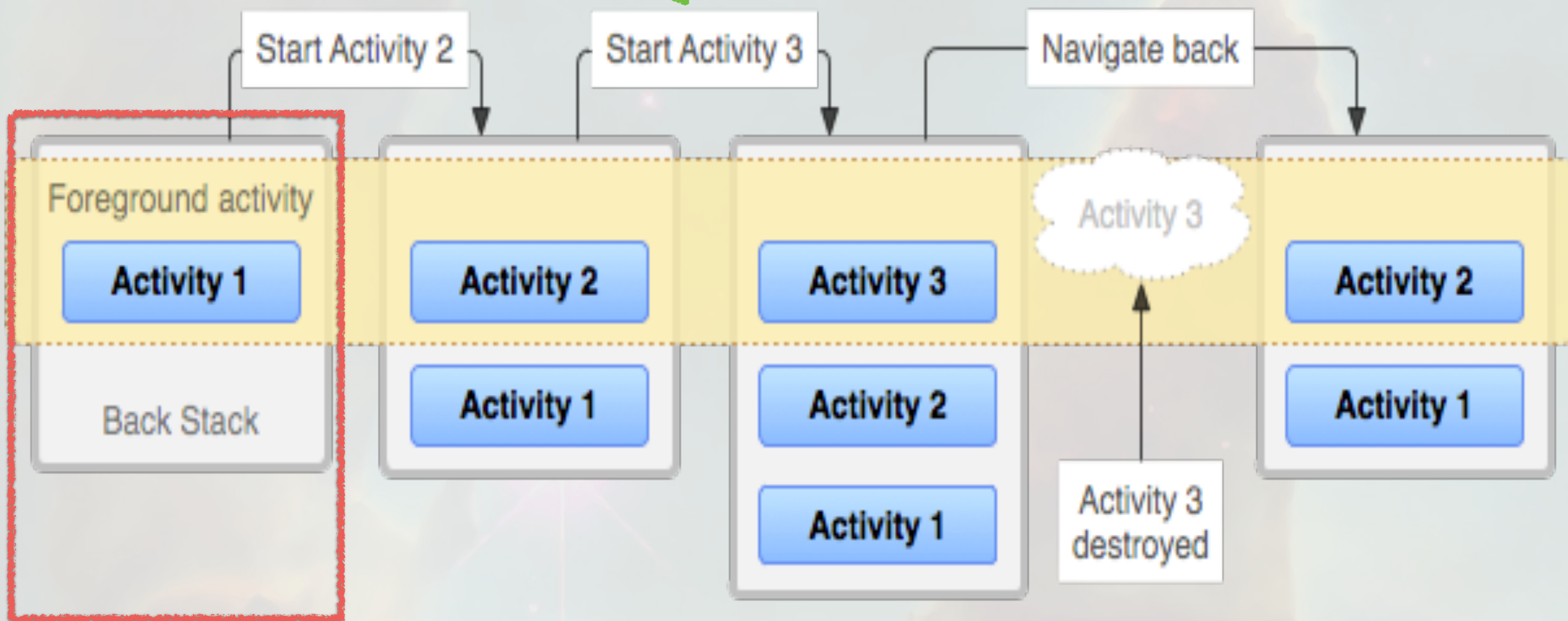
The Backstack



The home screen is the main entry point

- it corresponds to an empty backstack
 - ▶ **not exactly true, more information later in this lecture**
- A new activity is pushed into the backstack and become visible
- Press the back button to remove the pushed activity

Intents



Intents & Intents Bus



An Intent:

- 📱 is an asynchronous message asking for launching another component
- 📱 is launched from the activity at the top of the back stack (let's ignore services for now)
- 📱 can launch an activity that is not a part of the current activity



The Intent Bus:

- 📱 dedicated to the circulation of Intents
- 📱 communication with other activities (inner to the application or outer to the application)
- 📱 All activities listen this bus
 - ▶ **we can restrict this listening only to "interesting" messages**

How to work with Intents?



Start an Activity with arguments

```
Intent intent = new Intent(this, SecondActivity.class);
intent.putExtra("My_Additional_data", 42);
startActivity(intent);
```



Get the value associated to an Intent

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_second);
    Intent intent = getIntent();
    int value = intent.getIntExtra("My_Additional_data", 0);
    // Some stuff with value...
}
```

AndroidManifest.xml & Intents



An activity can be started by another only if it has been declared in AndroidManifest.xml

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

Otherwise your program will silently fails

Activities and Intents



When an Activity receive an activation intent

- it becomes active (visible)
- it has been pushed on the top of the backstack
- it can get the arguments for the activity in the onCreate method



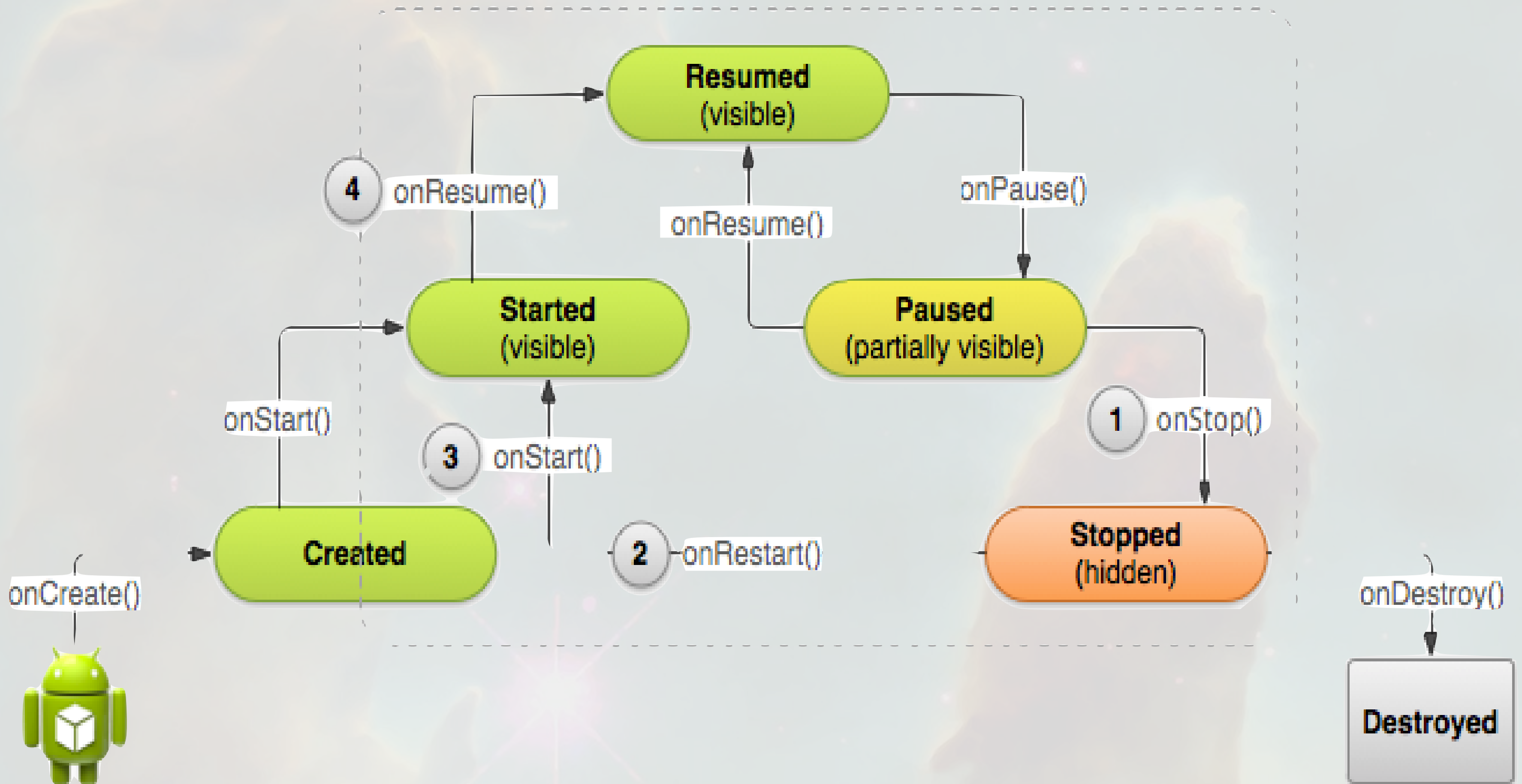
As soon as the activation intent has been received, the sender Activity becomes inactive (invisible)

Understanding lifecycles in Android is important to understand visibility and backstack management

Activity Lifecycle



6 states , 9 methods



Details of the methods

<u>Methods</u>	<u>Description</u>	<u>Visible</u>	<u>Killable</u>
onCreate	Instanciacion + binding evt	no	no
onStart onRestart	Initialization Load persistent data Restart Sensors	no	no
onResume	Display UI	yes	no
onPause	Save / Persistence Stop animations ... Stop service (GPS , ...)	no	yes
onStop	Free RAM	no	yes
onDestroy	Delete UI Free Ressources	no	yes

Logs



Dalvik Debug Monitor System (ddms)

- 🎧 Console in AndroidStudio
- 🎧 Logs/Displays event according to a specific level
 - ▶ **log.d: Debug**
 - ▶ **log.v: Verbose**
 - ▶ **log.i: Info**
 - ▶ **log.w: Warning**
 - ▶ **log.e: Errors**



Best practice: define a TAG in your activities:

```
private final String TAG = "SecondActivity"
```






And use it for the logger:

```
Log.v(TAG, "Intent contains value:" + value);
```

Summary




The role of the backstack

-  the activity on the top is the visible (active) one
-  press the return button pop the head of the backstack
-  the home screen represents an empty (kind of) backstack



Quick overview of Intents

-  They can do a lot more than just launching activities (later in this lecture)



Lifecycle of an Activity

-  Numerous states with restricted permissions



DDMS: logger for debugging and tracing applications



MultipleActivities Example

Renault@lrde.epita.fr



BeerApp Example

Three activities

 MainActivity

▶ Start screen with a "register button"

 InscriptionActivity

▶ Fields required for registration




 ResultActivity

▶ Display a text according to your age



Use XML to store static values

Group constants in XML files

-  Good practice since you will not use magic static values in your code
-  Avoid to repeat multiple time same text
-  Put all constants into **res/value/string.xml**

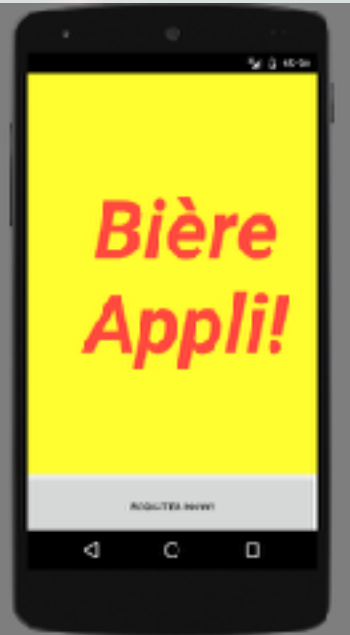
```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <string name="app_name">Beer App!</string>
  <string name="register_now">Register now!</string>
  <string name="inscription">Inscription</string>
  <string name="name">Name</string>
  <string name="firstName">First Name</string>
  <string name="age">Age</string>
  <string name="submit">Submit</string>
  <string name="okDrink">Drink with me!</string>
  <string name="koDrink">"Don't drink!"</string>
</resources>
```


Build an App based on Localisation



Localisation

- Adapt your texts based on the language of the user
- Copy the **res/value** directory into **res/value-fr**
- Edit **res/value-fr/strings.xml**



```
<resources>
  <string name="app_name">Beer App!</string>
  <string name="...">...r now!</string>
  <string name="...">...ion</string>
  <string name="...">...</string>
  <string name="...">...</string>
  <string name="...">...>
  <string name="ORDRINK">Drink with me!</string>
  <string name="koDrink">"Don't drink!"</string>
</resources>
```

You can change the local of the device through System > Preferences > Language

- Android will select the correct file based on the Locale

Main GUI (1/2)



Three elements in the hierarchy of views

- A LinearLayout
- A TextView
- A Button



GUI description: activity_main.xml

```
<LinearLayout xmlns:android=  
    "http://schemas.android.com/apk/res/android"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:layout_width="fill_parent"  
    android:layout_height="fill_parent"  
    android:orientation="vertical"  
    android:baselineAligned="false"  
    android:weightSum="1">
```

Main GUI (2/2)

<TextView

```
android:layout_width="396dp"
android:layout_height="497dp"
android:textAppearance=
    "?android:attr/textAppearanceLarge"
android:text="@string/app_name"
android:id="@+id/display_name"
android:gravity="center_vertical|center_horizontal"
android:textStyle="bold|italic"
android:textColor="#ffff4a45"
android:textSize="100dp"
android:background="#ffffff53" />
```

<Button

```
android:layout_width="match_parent"
android:layout_height="79dp"
android:text="@string/register_now"
android:id="@+id/button_register" />
```

```
</LinearLayout>
```

Registration Form GUI (1/2)



Five elements in the hierarchy of views

- A LinearLayout
- A TextView
- Three EditText
- A Button



GUI description: activity_inscription.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="fill_parent"    android:layout_height="fill_parent"
    android:orientation="vertical"    android:baselineAligned="false"
    android:weightSum="1">
```

<TextView

```
    android:layout_width="match_parent"    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="@string/inscription"    android:id="@+id/textView"
    android:layout_weight="0.43"    android:gravity="center_horizontal"
    android:textStyle="bold|italic"    android:textSize="70dp"
```

```
/>
```

Registration Form GUI (2/2)

<EditText

```
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:inputType="textPersonName"  
android:ems="10"  
android:layout_weight="0.04" />
```

```
android:text="@string/firstName"  
android:id="@+id/text_firstname"
```

<EditText

```
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:inputType="number"  
android:id="@+id/text_age"  
android:layout_gravity="center_horizontal"  
android:layout_weight="0.04"
```

```
android:ems="10"
```

```
android:text="@string/age" />
```

<Button

```
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:text="@string/submit"  
android:layout_gravity="center_horizontal"  
android:layout_weight="0.04" />
```

```
android:id="@+id/button_submit"
```

```
</LinearLayout>
```

GUI post inscription



Two elements in the hierarchy of views

- A LinearLayout
- A TextView with runtime initialization



GUI description: activity_result.xml

<LinearLayout

```
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="fill_parent"           android:layout_height="fill_parent"
android:orientation="vertical"              android:baselineAligned="false"
android:weightSum="1">
```

<TextView

```
    android:layout_width="match_parent"    android:layout_height="match_parent"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="Large Text"              android:id="@+id/text_display"
    android:textSize="70dp"
    android:gravity="center_vertical|center_horizontal" />
```

</LinearLayout>

MainActivity



Must build the GUI and track clicks on button

- Build the GUI during the call to the `onCreate` callback
- Action when the button is clicked
 - ▶ Implement `View.OnClickListener` Load the GUI XML during creation
 - ▶ set `this` as listener

```
public class MainActivity extends Activity implements View.OnClickListener {
    Button btGotoInscription;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        btGotoInscription = (Button) findViewById(R.id.button_register);
        btGotoInscription.setOnClickListener(this);
    }
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(this, InscriptionActivity.class);
        startActivity(intent);
    }
}
```

Registration Form



Action when the user click on the button

- We must capture the age field
- Always implement View.OnClickListener

```
public class InscriptionActivity
    extends Activity implements View.OnClickListener {
    private Button buttonSubmit;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_inscription);
        buttonSubmit = (Button) findViewById(R.id.button_submit);
        buttonSubmit.setOnClickListener(this);
    }
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(this, ResultActivity.class);
        TextView age = (TextView) findViewById(R.id.text_age);
        intent.putExtra("AGE", Integer.parseInt(age.getText().toString()));
        startActivity(intent);
    }
}
```


Last Screen: display result

 Must grab the age from the intent

 Must instantiate the XML

```
public class ResultActivity extends Activity{
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_result);
        int age = getIntent().getIntExtra("AGE", 0);
        TextView display =
            (TextView)findViewById(R.id.text_display);
        if (age <= 20)
            display.setText(R.string.koDrink);
        else
            display.setText(R.string.okDrink);
    }
}
```

Setup Android Permissions



Modify the AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.admin.myfirstapplication" >
    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".InscriptionActivity" />
        <activity android:name=".ResultActivity" />
    </application>
</manifest>
```

Summary



End to end application

- Three Activities
- Manage simple events (clicks)
- Simple Navigation through the back stack
- Intents
 - ▶ **With or Without parameters**
- Static value management
 - ▶ **Through res/values/*.xml**



Localisation

- On simple value type
- Can also be applied on all ressources
- Application name can also be changed!



Handle Device Orientation

Renault@lrde.epita.fr



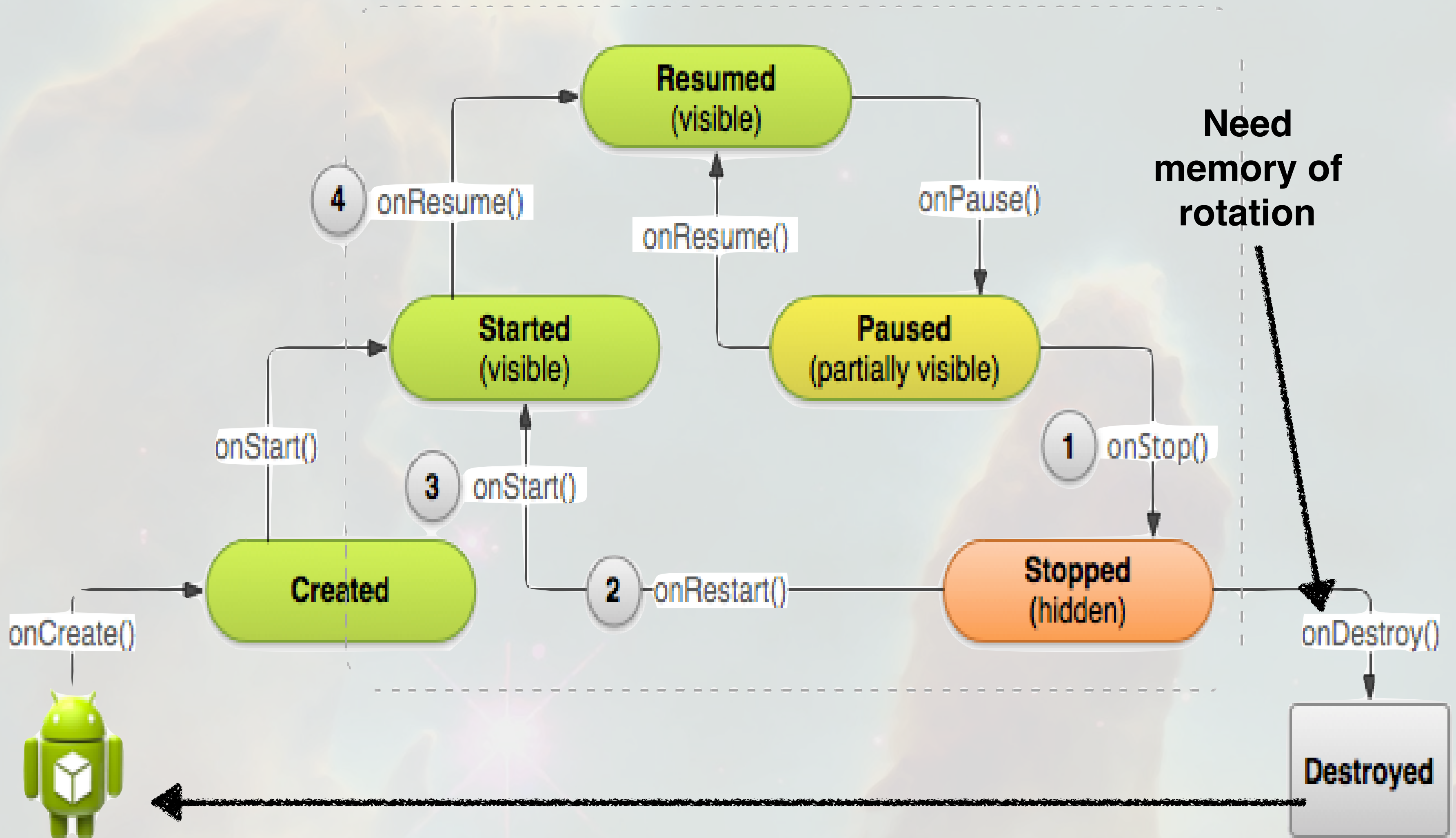
Motivation

Let us consider a simple application

-  Only count the number of clicks on a button



The Origins of the Problem



Re-build the application !



How to solve this problem?



We can fix the orientation of the activity

- 🎧 Technique used in video-games
- 🎧 Easy, a single modification is enough

```
<activity android:name=".MainActivity"  
          android:screenOrientation="landscape"  
          android:label="@string/app_name" />
```

```
<activity android:name=".MainActivity"  
          android:screenOrientation="portrait"  
          android:label="@string/app_name" />
```



We can define static views for landscape and portrait

- 🎧 res/layout directory for portrait static layout
- 🎧 res/layout-land directory for landscape static layout

How to detect a rotation?

Handle configuration changes in AndroidManifest.xml

```
<activity android:name=".MainActivity"
  android:configChanges="orientation|screenSize"
  android:label="@string/app_name" />
```

Then override the `onConfigurationChanged` method

```
public void onConfigurationChanged(Configuration newConfig) {
    super.onConfigurationChanged(newConfig);

    // Checks the orientation of the screen
    if (newConfig.orientation == Configuration.ORIENTATION_LANDSCAPE) {
        // do something...
    }
    else if (newConfig.orientation == Configuration.ORIENTATION_PORTRAIT) {
        // do something...
    }
}
```

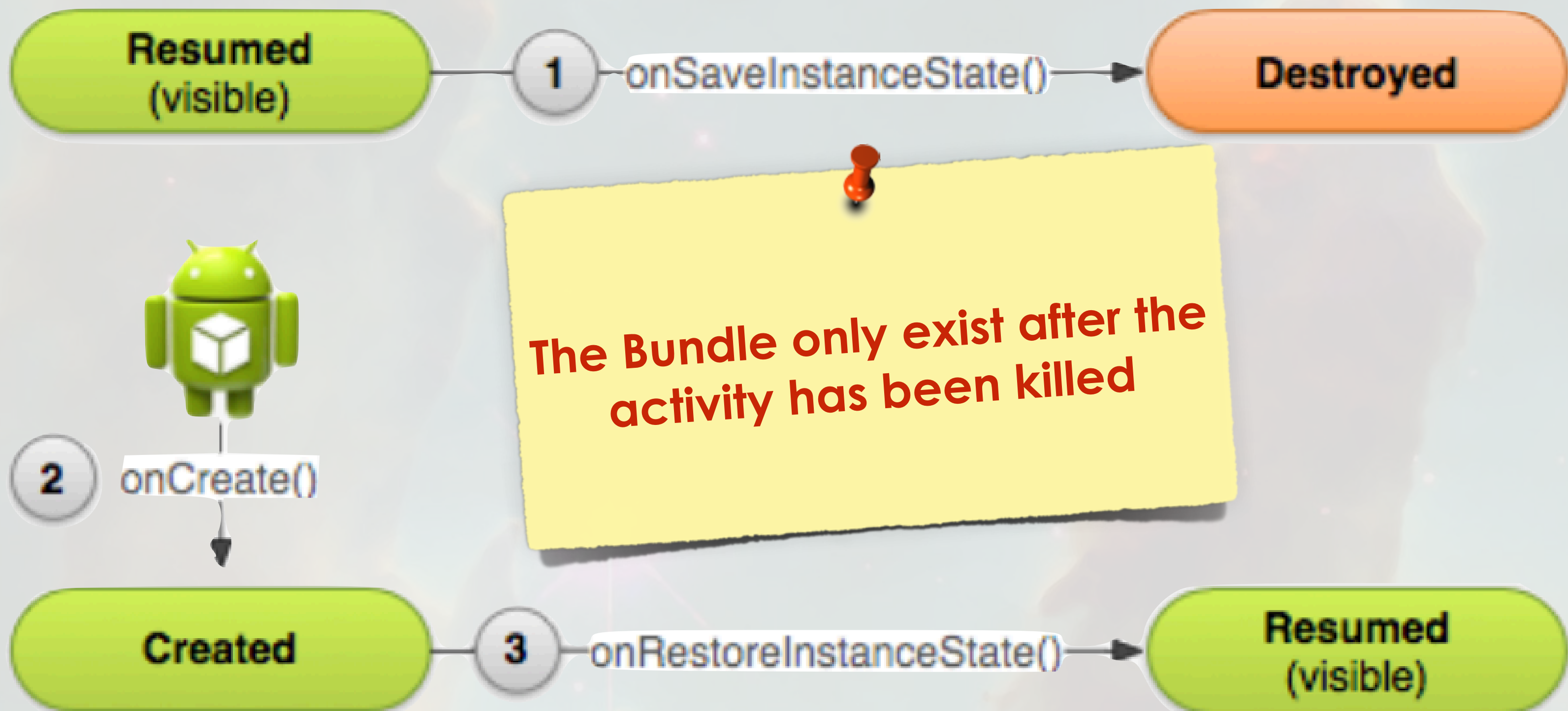
Handling configuration changes does not avoid data loss...

How to avoid data loss?



Use the Bundle

- Associative <key, values> with heterogeneous values
- onSaveInstanceState: save data before a rotation
- onRestoreInstanceState: restore data after a rotation
- Bundle is also passed to onCreate (but may be null)



Save Primitive types into the Bundle (1/2)

Save primitive types

```
@Override
public void onSaveInstanceState(Bundle savedInstanceState) {
    super.onSaveInstanceState(savedInstanceState);
    savedInstanceState.putInt("NB_CLICK", nbClick);
}
```

Restore saved values

```
@Override
protected void
onRestoreInstanceState(Bundle savedInstanceState) {
    super.onRestoreInstanceState(savedInstanceState);
    nbClick = savedInstanceState.getInt("NB_CLICK", 0);
    // more stuff...
}
```

Save Primitive types into the Bundle (2/2)



Restoration can also be done in onCreate (but may be null)

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    if (savedInstanceState != null)
        nbClick = savedInstanceState.getInt("NB_CLICK", 0);
    // more stuff...
}
```

🔊 getInt / putInt / putIntegerArrayList / ...

🔊 getChar / putChar / ...

🔊 getByte / putByte / ...

🔊 ...

Save more complex types

78



The Parcelable interface helps to serialize objects

- Parcelable object can be stored and retrieved into / from the Bundle
 - ▶ **putParcelable / getParcelable**



How to implement a Parcelable Object?

- writeToParcel(Parcel out, int flag): serialize the object
 - ▶ **out: the receiver object for ou serialization**
 - ▶ **flags: various set of options for the serialization**
- MyClass(Parcel in): unserialize object
- describeContents(): useless unless we want to serialize file descriptor and when all bits are relevant
- Creator to define loading one or multiple objects

How to implement Parcelable Interface? (1/2)

```
public class Student implements Parcelable {
    private String name;
    public int grade;

    // Simple Constructor
    Student(String theName, int theGrade) {
        name = theName;          grade = theGrade;
    }

    // How to load this Object from a parcel
    private Student(Parcel in) {
        grade = in.readInt();
        name = in.readString();
    }

    // How to save this Object
    public void writeToParcel(Parcel out, int flags) {
        out.writeInt(grade);
        out.writeString(name);
    }
}
```

How to implement Parcelable Interface? (2/2)

```
// Unused here
public int describeContents() {
    return 0;
}

// Creator to define loading one or multiple objects
public static final Parcelable.Creator<Student> CREATOR
    = new Parcelable.Creator<Student>() {
    public Student createFromParcel(Parcel in) {
        return new Student(in);
    }

    public Student[] newArray(int size) {
        return new Student[size];
    }
};
}
```


Load and Save Parcelable Objects

Save during a rotation

```
@Override
public void onSaveInstanceState(Bundle savedInstanceState) {
    super.onSaveInstanceState(savedInstanceState);
    savedInstanceState.putInt("Student", student);
}
```

Load after a rotation

 in onCreate for the purpose of this slide!

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    if (savedInstanceState != null)
        student = savedInstanceState.getParcelable("Student");
    else
        student = new Student("etienne", 0);
}
```

Summary



You must deal with the orientation problem



Fix device orientation



OR use dedicated views



OR detect configuration changes



OR use the Bundle

▶ the last one is the preferred one

▶ One should note that configuration changes detects more than only rotation



The Bundle is only a temporary storage



It does not survive when an application exits



It can handle any types as long as they implement parcelable





Notifications

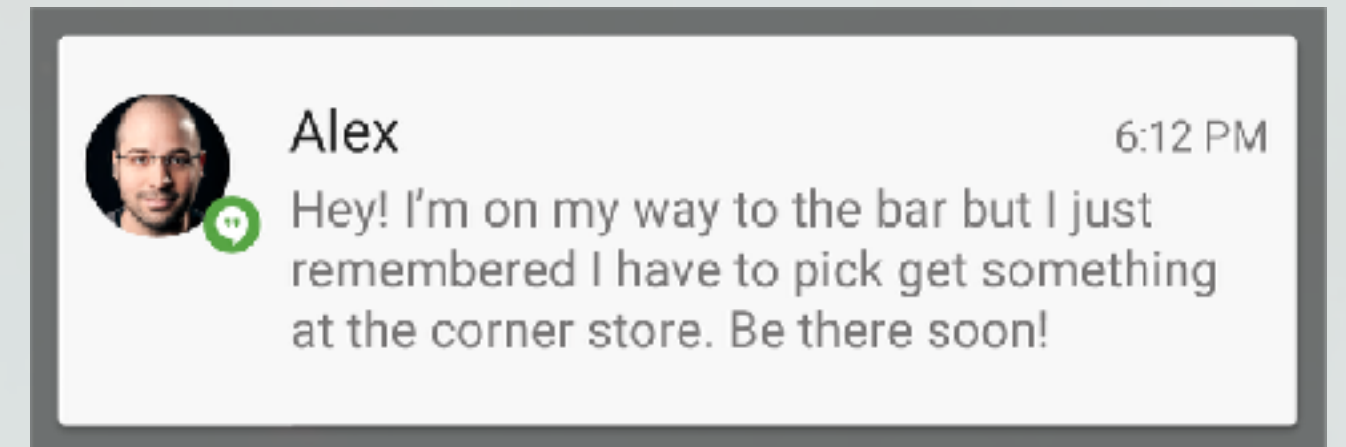
Renault@lrde.epita.fr








Inform the User that something

-  just happened (SMS, mail, etc)
-  will happen (reminders, meetings, etc.)



Only visible for a certain amount of time

-  they do not stop the current activity
-  they use dedicated (restricted) space for their display
-  the user is free to interrupt its activity through an action



Lollipop (and higher version) provide new notifications that are more aesthetic and allow more actions

Relative Importance of Notifications

MAX

Critical: something requires an immediate attention

HIGH

Communication: SMS, mail, telephony, ...

Notifications are ordered according to their priorities

Your application is not more important than all the other ones

LOW

Non-important: Software upgrades

MIN

Contextual informations: weather, etc.

Manage Notifications (1/2)

Building a notification

```
NotificationCompat.Builder mBuilder =  
    new NotificationCompat.Builder(getBaseContext())  
        .setSmallIcon(R.drawable.ic_launcher)  
        .setContentTitle("My notification")  
        .setContentText("Hello World!")  
        .setPriority(Notification.PRIORITY_MAX);
```

Grab the notification manager

```
// Gets an instance of the NotificationManager service  
NotificationManager mNotifyMgr = (NotificationManager)  
    getSystemService(NOTIFICATION_SERVICE);
```

Publish it:

```
mNotifyMgr.notify(mNotificationId, mBuilder.build());
```

For later updates

Manage Notifications (2/2)

Suppress a notification

-  cancel(int): delete a notification using its identifier
-  cancelAll(): delete all notifications from the current application

Fix the visibility on the home screen

-  setVisibility: fix the visibility using above criteria

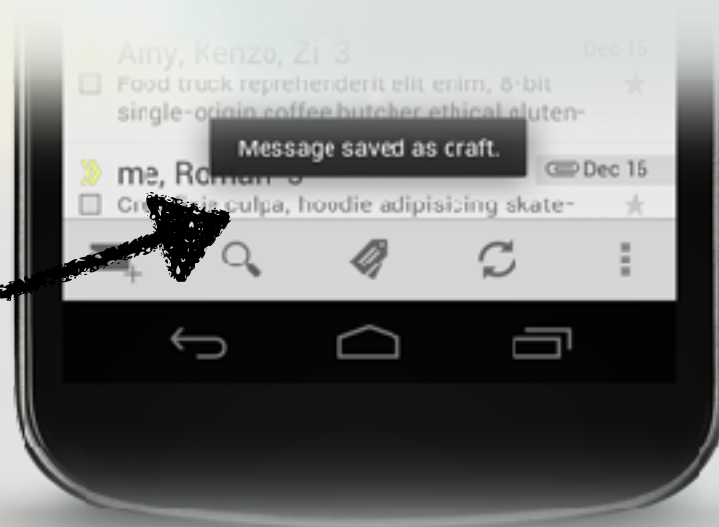
VISIBILITY_PUBLIC	Display all the content of the notification
VISIBILITY_PRIVATE	Only display basic informations (application, etc.)
VISIBILITY_SECRET	Minimum display: not even the name of the originated application

Toasts



Toast

Contextual
Message



- Immediate message popping like a toast



```
Toast.makeText(getApplicationContext(),  
    "MyText", Toast.LENGTH_SHORT).show();
```

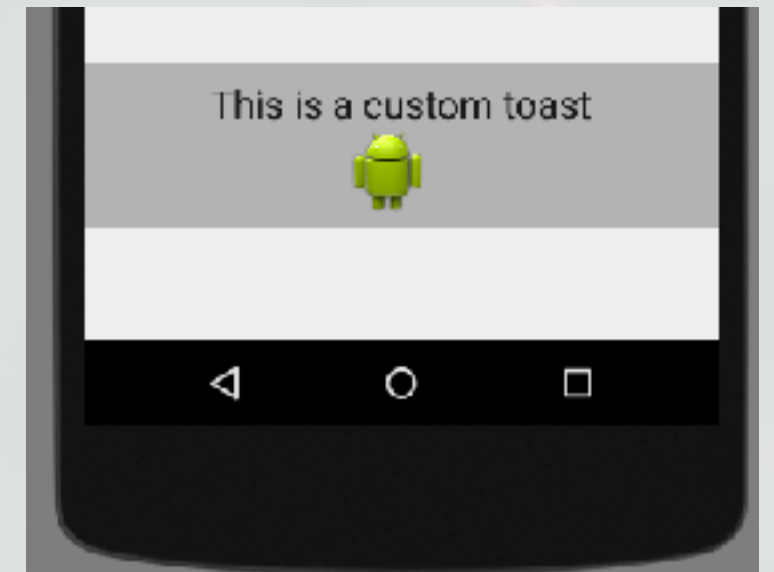
- The length of the display can be fixed
 - ▶ **Toast.LENGTH_SHORT: 2 secondes**
 - ▶ **Toast.LENGTH_LONG: 3.5 secondes**
- The position over the screen can be fixed using `setGravity`
 - ▶ **Gravity.TOP, Gravity.BOTTOM, Gravity.RIGHT, Gravity.LEFT**

```
toast.setGravity(Gravity.TOP | Gravity.LEFT, xoffset, yoffset);
```

Customized Toasts

Toasts can be customized

-  Define a layout (here TextView + ImageView)
-  Use the layout inflater to build it







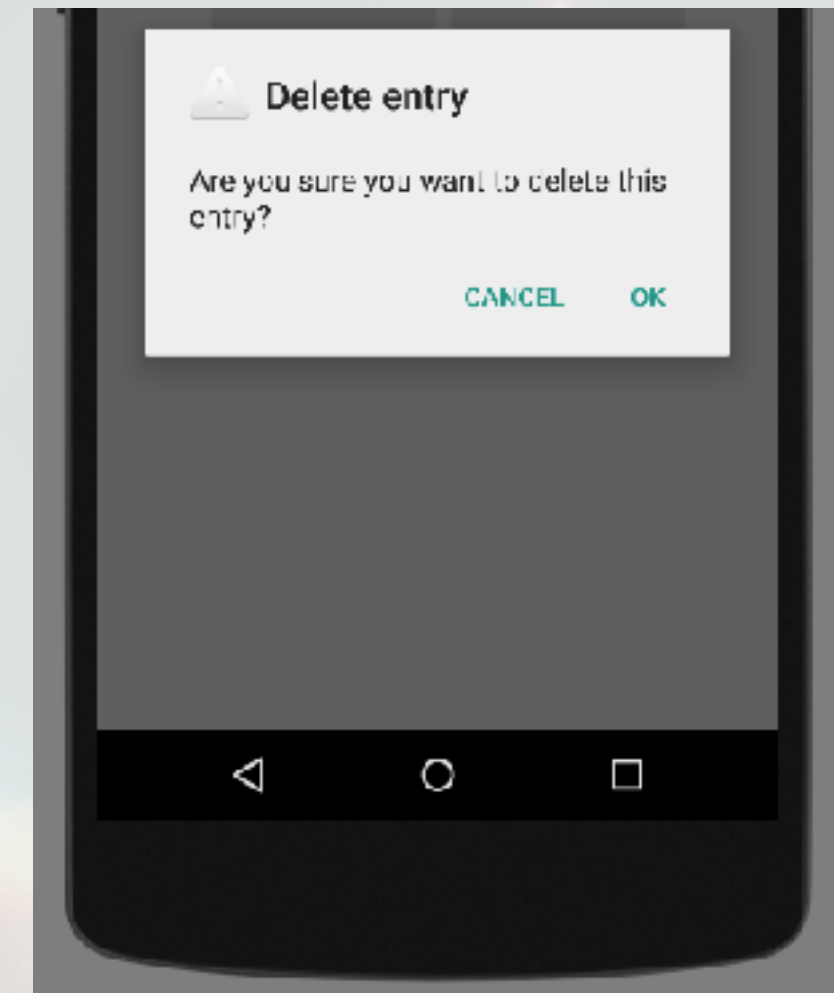
```
LayoutInflater inflater = getLayoutInflater();
View layout = inflater.inflate(R.layout.toast_layout,
    (ViewGroup) findViewById(R.id.toast_layout_root));
TextView text = (TextView)
    layout.findViewById(R.id.textDisplay);
text.setText("This is a custom toast");
Toast toast = new Toast(getApplicationContext());
toast.setDuration(Toast.LENGTH_LONG);
toast.setView(layout);
toast.show();
```

-  Use constructor only if there is a View in the customized Toast!
Otherwise makeText if enough!

AlertDialog

Generic user interaction


-  A title
-  Three button maximum
 - ▶ **with customizable actions**
-  Clickable list of elements
-  Customized Layout

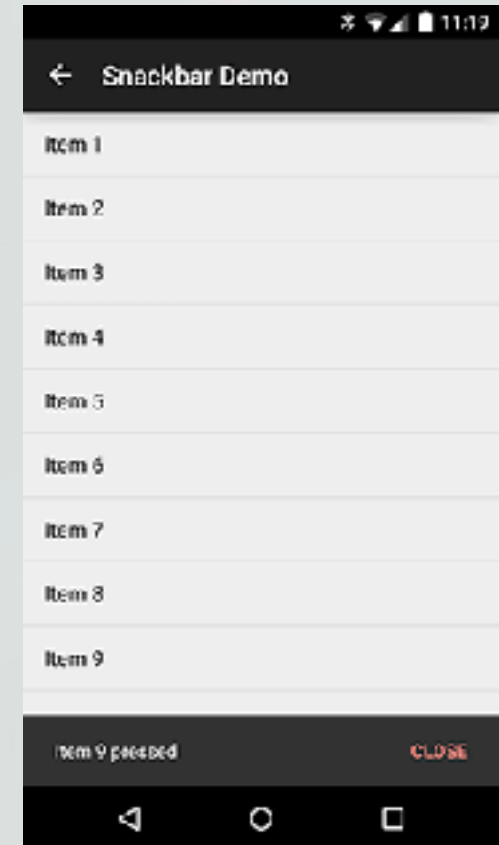


```
new AlertDialog.Builder(MainActivity.this)
    .setTitle("Delete entry")
    .setMessage("Are you sure you want to delete this entry?")
    .setIcon(android.R.drawable.ic_dialog_alert)
    .setPositiveButton(android.R.string.yes,
        new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int which) {
                // continue with delete
            }
        }
    ).show();
```

Snackbar

Non-intrusive notifications

-  One button (i.e. action) maximum
-  Customizable



Build.graddle may be updated

```
compile 'com.android.support:design:23.1.1'
```

```
Snackbar.make(findViewById(R.id.container),  
    "Item 9 pressed", Snackbar.LENGTH_INDEFINITE)  
    .setAction("close", new View.OnClickListener() {  
        @Override  
        public void onClick(View v) {  
            }  
    }).show();
```

Summary



Many components can be used to interact with the user without building a with with dedicated buttons

it helps to have a homogeneous environment across applications



Notifications warn that something happened

Priority can be fixed



Contextual notifications

Toast: transient information

Dialog: require a user action



Other kind of notification exist

For instance, `ProgressView` notify that something is actually in progress



Gesture Detection

Renault@lrde.epita.fr



Touch Detection



Implement onTouchEvent(MotionEvent)

Return true if the event has been capture, false otherwise

Event	Description
<code>MotionEvent.ACTION_DOWN</code>	New touch
<code>MotionEvent.ACTION_MOVE</code>	The touch is moving
<code>MotionEvent.ACTION_UP</code>	The finger is up
<code>MotionEvent.ACTION_CANCEL</code>	Current touch is cancelled (an another touch as the focus)
<code>MotionEvent.ACTION_POINTER_DOWN</code>	New touch
<code>MotionEvent.ACTION_POINTER_UP</code>	End of a touch (multitouch)

How to build a Drawing Canvas? (1/3)

There is no drawing canvas

- 🗣️ But building a fresh one is easy!
 - ▶ Detect touches down, up and move
 - ▶ updates the view according to these
 - ▶ View.onDraw() allows to draw on a View

Extend the View Class

```
public class DrawCanvas extends View {
    private Paint paint = null;
    private Path path = null;

    public DrawCanvas(Context context, AttributeSet attrs) {
        super(context, attrs);
        reset();
    }
}
```

How to build a Drawing Canvas? (2/3)

```
public void reset(){
    path = new Path();
    paint = new Paint();
    paint.setAntiAlias(true);
    paint.setStrokeWidth(6f);
    paint.setColor(Color.BLACK);
    paint.setStyle(Paint.Style.STROKE);
    paint.setStrokeJoin(Paint.Join.ROUND);
    invalidate();
}

@Override
protected void onDraw(Canvas canvas) {
    canvas.drawPath(path, paint);
}

@Override
protected void onDraw(Canvas canvas) {
    canvas.drawPath(path, paint);
}
```

How to build a Drawing Canvas? (3/3)

```
@Override
public boolean onTouchEvent(MotionEvent event) {
    float eventX = event.getX();
    float eventY = event.getY();
    switch (event.getAction()) {
        case MotionEvent.ACTION_DOWN:
            path.moveTo(eventX, eventY);
            return true;
        case MotionEvent.ACTION_MOVE:
            path.lineTo(eventX, eventY);
            break;
        case MotionEvent.ACTION_UP:
            // nothing to do
            break;
        default:
            return false;
    }
    // Schedules a repaint.
    invalidate();
    return true;
}
}
```

Instanciate the Component



How to instantiate programmatically a *new* Component

```
setContentView(new DrawCanvas(this, null));
```



Though the graphical interface (XML)

- 🔊 Declare the new component in res/values/attr.xml

```
<resources>  
    <declare-styleable name="DrawCanvas">  
    </declare-styleable>  
</resources>
```

- 🔊 The component is now available for drag-and-drop in the Visual Editor
Custom > CustomView

Handle Multitouch

Implement onTouchEvent(MotionEvent event)

 Get the action

```
int maskedAction = event.getActionMasked();
```

 Get the number of actions in progress

```
event.getPointerCount();
```

 Get the index of ACTION_POINTER_1, ACTION_POINTER_2, ACTION_POINTER_3, ACTION_POINTER_4, ACTION_POINTER_5

```
event.getActionIndex();
```

 Get the coordinates of an event

```
event.getX(i);  
event.getY(i);
```

Works as in single touch mode

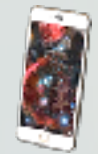
You only have to keep touches in memory!

Predefined gestures



GestureDetector helps to catch predefined gestures

- All events detected by onTouchEvent must be forwarded to GestureDetector
- **Warning:** onTouchEvent must call the super constructor to trigger the callback



Implement GestureDetector.OnGestureListener

- onDown: detects a single touch
- onShowPress: detects a touch not yet up/cancelled
- onSingleTapUp: detects a single tap
- onScroll: detects a scrolling move
- onLongPress: detects a long press
- onFling: detects a dynamic move

Implement Swipe Up-Down-Left-Right

```
private static final int SWIPE_MIN_DISTANCE = 120;
private static final int SWIPE_THRESHOLD_VELOCITY = 200;

@Override
public boolean onFling(MotionEvent e1, MotionEvent e2,
    float velocityX, float velocityY) {
    try {
        if(e1.getX() - e2.getX() > SWIPE_MIN_DISTANCE &&
            Math.abs(velocityX) > SWIPE_THRESHOLD_VELOCITY)
            // Left Swipe ...
        else if (e2.getX() - e1.getX() > SWIPE_MIN_DISTANCE &&
            Math.abs(velocityX) > SWIPE_THRESHOLD_VELOCITY)
            // Right Swipe ...
        if(e1.getY() - e2.getY() > SWIPE_MIN_DISTANCE &&
            Math.abs(velocityY) > SWIPE_THRESHOLD_VELOCITY)
            // Swipe up ...
        else if (e2.getY() - e1.getY() > SWIPE_MIN_DISTANCE &&
            Math.abs(velocityY) > SWIPE_THRESHOLD_VELOCITY) {
            // Swipe down ...
        } catch (Exception e) { }
    }
    return false;
}
```



Scaling effects

- 📱 ScaleGestureDetector: works similarly to `GestureDetector` and must analyse events received in `onTouchEvent`
- 📱 Implement `ScaleGestureDetector.OnScaleGestureListener`
 - ▶ **`onScale`, `onScaleBegin`, `onScaleEnd`**
- 📱 There is a default listener `ScaleGestureDetector.OnScaleGestureListener` if we only want to track special events



Double tap

- 📱 `ScaleGestureDetector.OnDoubleTapGestureListener`
 - ▶ **`onDoubleTap`, `onDoubleTapEvent`, `onSingleTapConfirmed`**

Summary



Detect single or multiple touches

- implement `onTouchEvent`
- For multi-touches the developer must keep track of previous events



Detecting touches helps to define a drawing canvas

- A new component can then be used directly in the GUI Editor



Predefined events can be easily detected

- `onFling` can capture almost all events
- Scaling, tap, doubletap, can easily be capture without reinventing the wheel

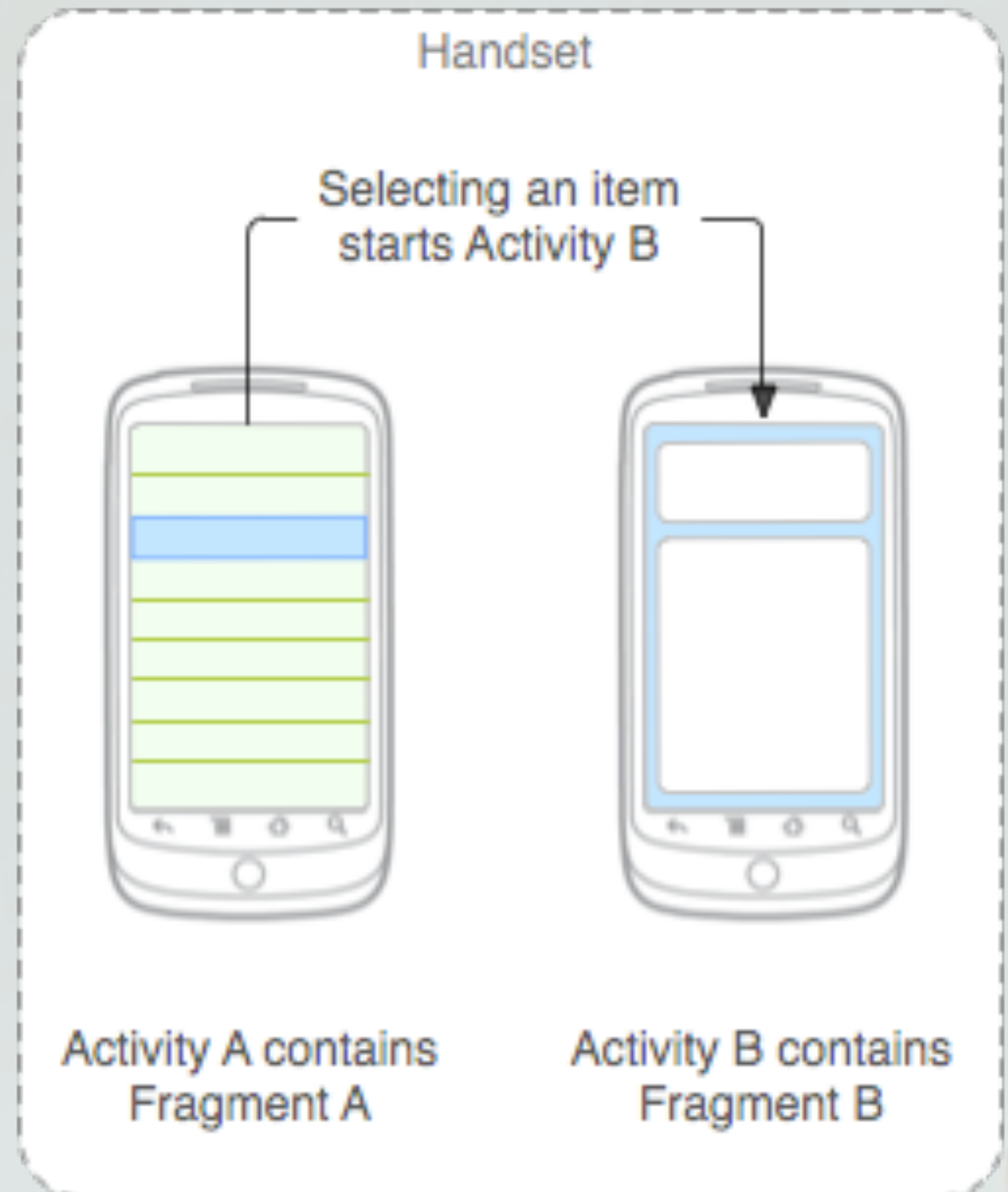
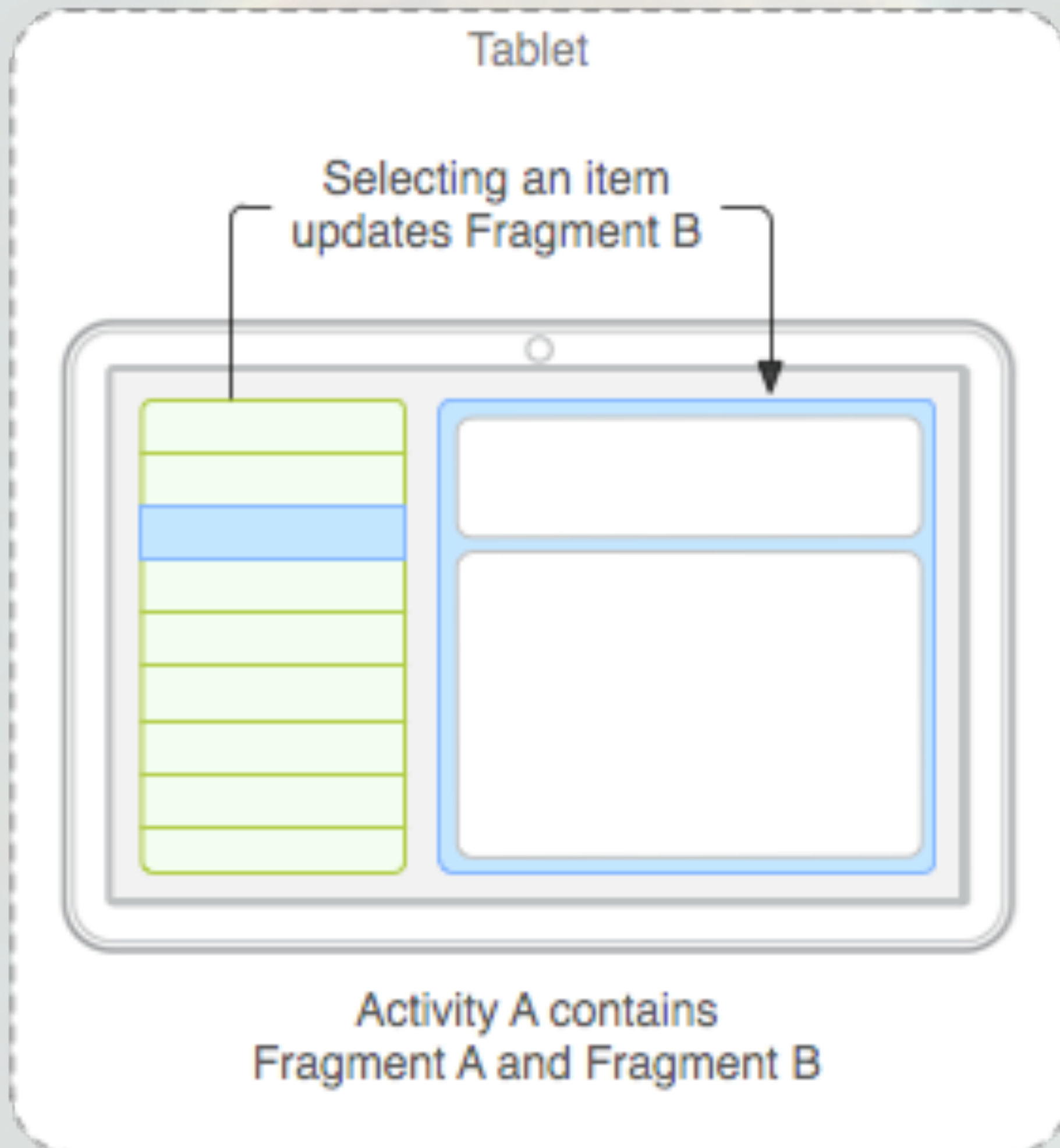


Fragments

Renault@lrde.epita.fr



Purpose of Fragments



Definition



Modular part of an activity (FragmentActivity)

- Easily reusable component
- Allows to build panels of views
- An activity can have as many fragments as desired



Adding or removing a fragment can be done dynamically

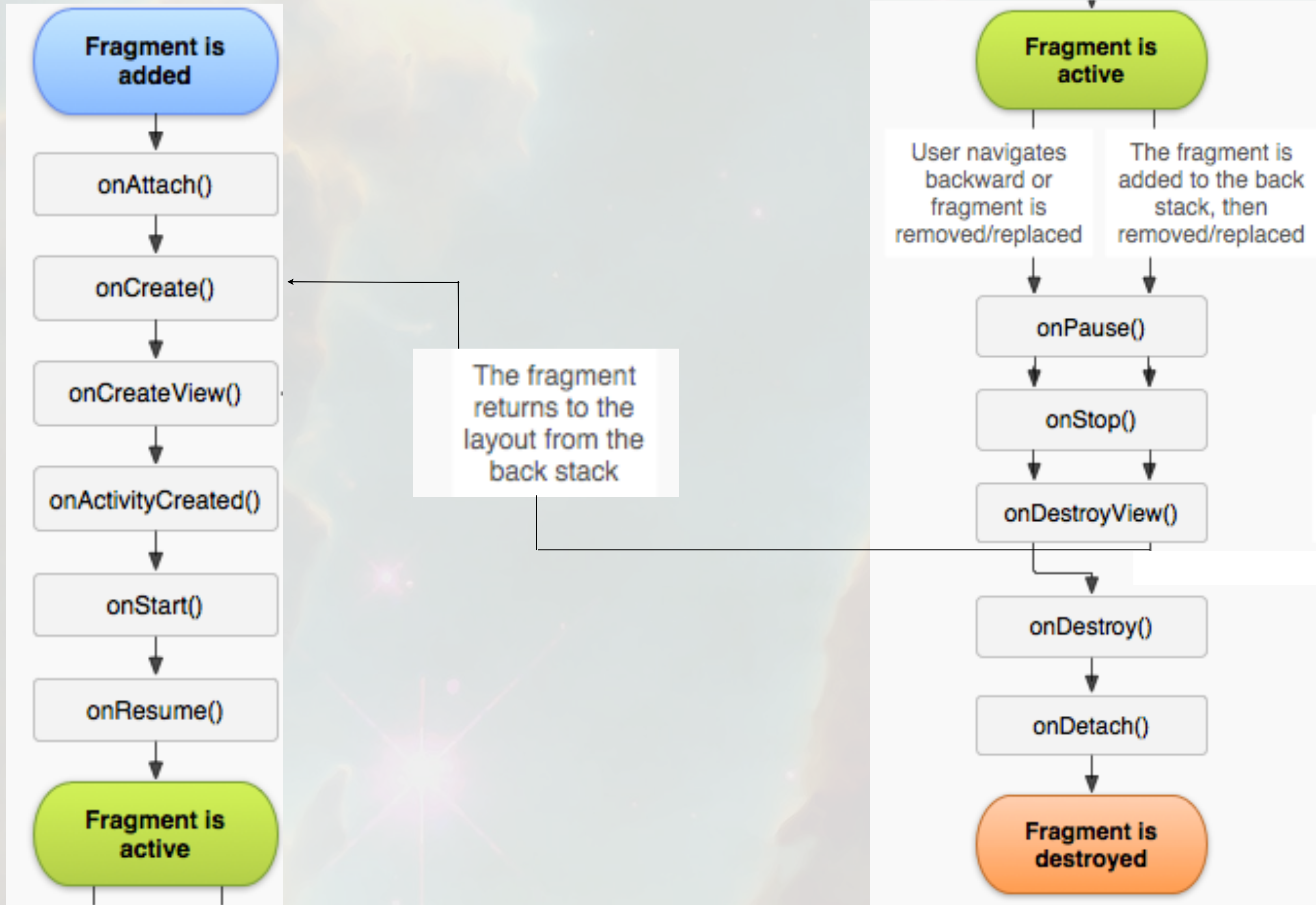
- Use well suited component adapted to the current device / orientation
- Keep the back stack unmodified



Dedicated Lifecycle

- Linked to the activity lifecycle

Fragment's Lifecycle



Example

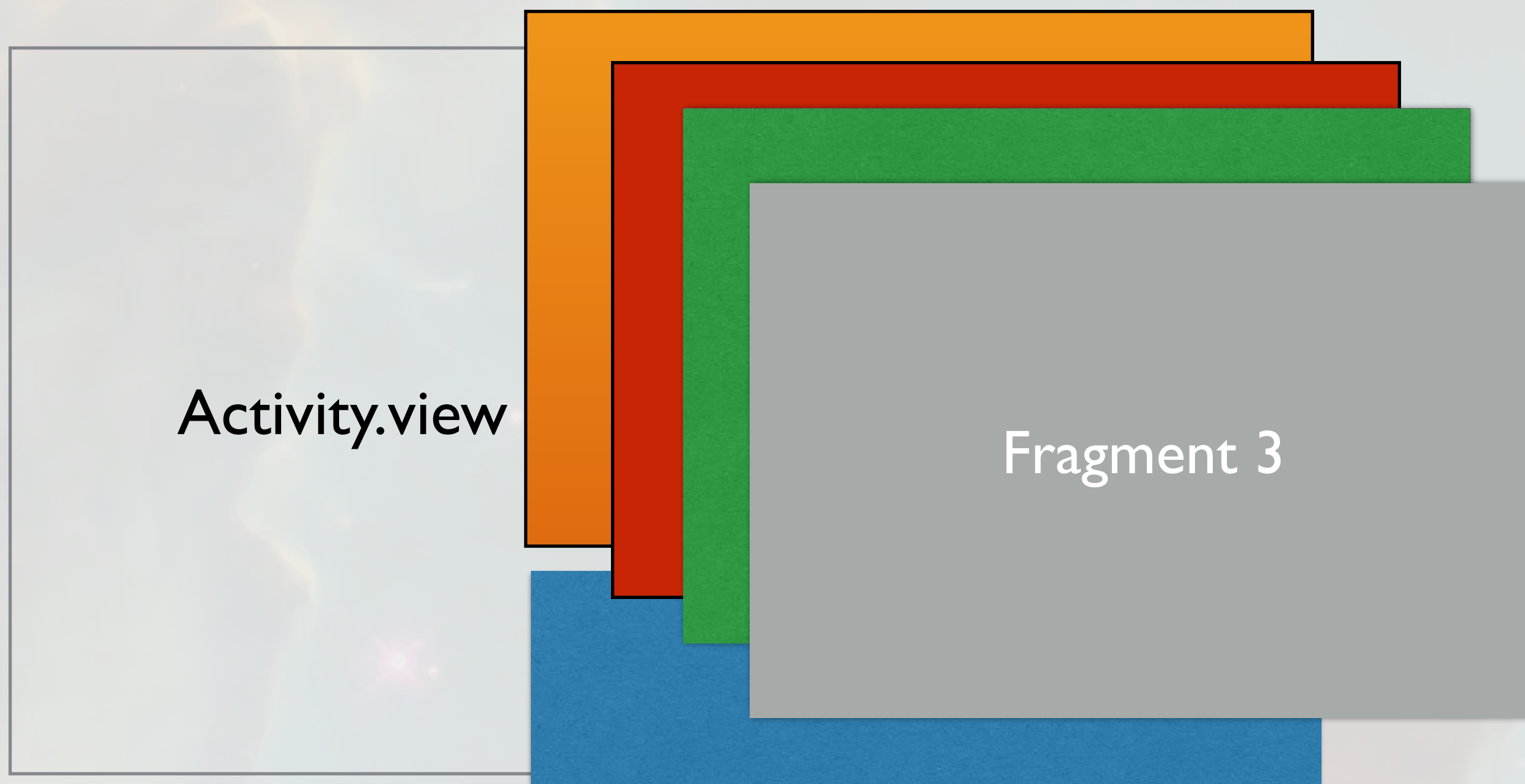


Example details (1/2)



- **Fragments use area to display**
- **Usually we use empty layout for this specific area**

Example details (2/2)



Fragments can be nested

Define the container



Only define a ViewGroup in the main's activity xml

<FrameLayout

```
xmlns:android="http://schemas.android.com/apk/res/android"  
xmlns:tools="http://schemas.android.com/tools"  
android:id="@+id/container"  
android:layout_width="match_parent"  
android:layout_height="518dp"  
tools:context=".MainActivity"  
tools:ignore="MergeRootFrame" >
```

</FrameLayout>

The area used to display fragments

How to build a Fragment



Define the GUI

- Through the GUI Editor (as seen for Activities)
- Using existing components: ListFragments, etc.



Add a fragment

```
getSupportFragmentManager()  
    .beginTransaction()  
    .add(R.id.container, new MyFragment())  
    .commit();
```



Replace a fragment

```
getSupportFragmentManager()  
    .beginTransaction()  
    .replace(R.id.container, new MyFragment())  
    .commit();
```

Animations

Defined through an XML

 Enter ([res/anim/enter.xml](#))

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android" >
  <translate
    android:duration="700"
    android:fromXDelta="0%"
    android:toXDelta="100%" >
  </translate>
</set>
```

call then:

```
setCustomAnimations(R.anim.enter,
                   R.anim.exit)
```

on a transaction




 Exit ([res/anim/exit.xml](#)),

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">
  <translate
    android:duration="700"
    android:fromXDelta="0%"
    android:toXDelta="100%" >
  </translate>
</set>
```

Summary



Objectives of Fragments

-  Build modular applications
-  Specify animations
-  Do not overload Intent Bus



Fragments have their own lifecycle



Build an great user experience whatever the considered device

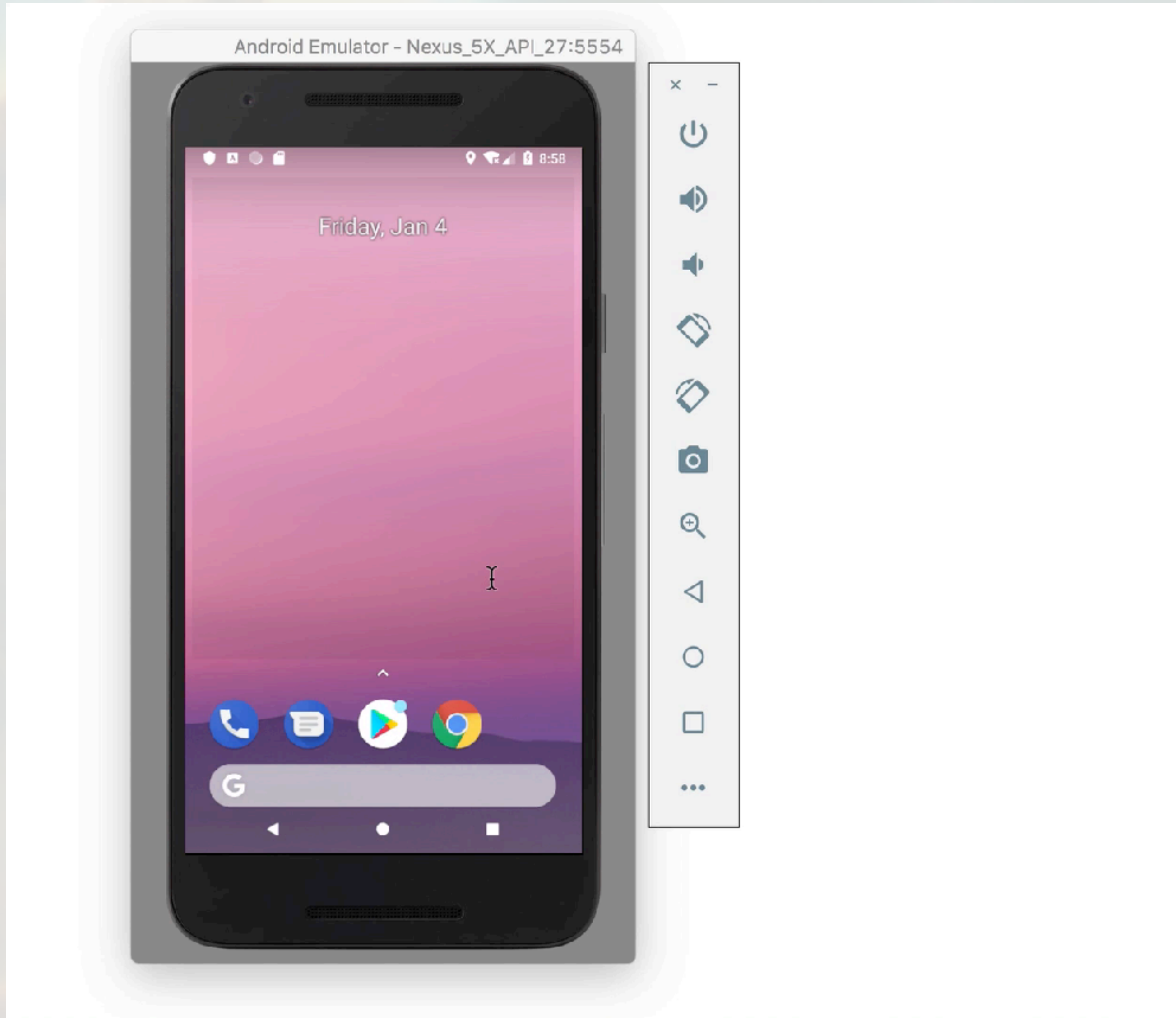


Exercise: HelloBye

Renault@lrde.epita.fr



Demo Video



Summary



Simple App!



Simple GUI to build



A Button



A TextView



Handle rotation



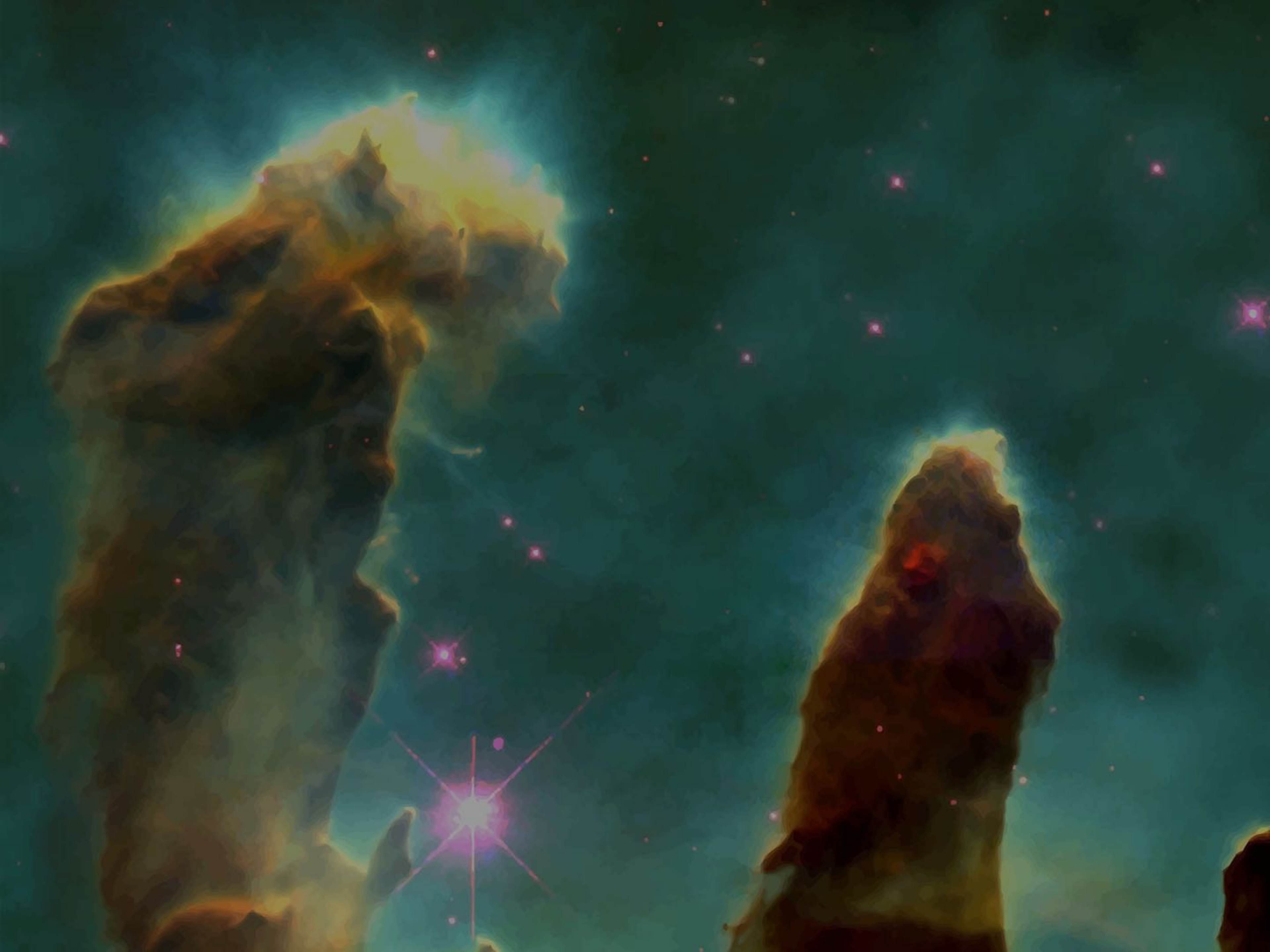
onSaveInstanceState



Learn to detect button click



setOnClickListener



Exercise: Gem Puzzle

Renault@lrde.epita.fr



The Game



Also called **Game of 15**, **Boss Puzzle** or **Taquin** (in french)

- Invented by Sam Loyd around 1870
- Rubick's cube ancestor



Objective: reorder numbers by sliding tiles on the empty tile

Some configurations cannot be solved.

We only consider 8 tiles and one empty room

13	2	3	
9	11	1	10
	6	4	14
15	8	7	5

	3	4	
5	6	7	8
9	10	11	12
13	14	15	

Implementation Details



Capture *swipe gesture*

- And move the tiles consequently
- Use GestureDetector
- Implement GestureDetector.OnGestureListener



Two options for the GUI

- Use TextView that will be updates according to user moves
- Use containers and fragments for each tile
 - ▶ **preferred option since it allows to implements animations for moving tiles**



Do not forget to shuffle randomly your tiles



Do not forget to detect the end of the Game

GUI



Initial state




- Yellow rectangles represent tiles to move
- Grey tile is the empty room
- A swipe move the **only** available tile for this move
- A Toast notify the user that its move has been detected



Summary

 An dialog displays when the user end the game

 With this simple game you can elaborate a valuable App

-  Temple Run
-  1024
-  CandyCrush
-  ...



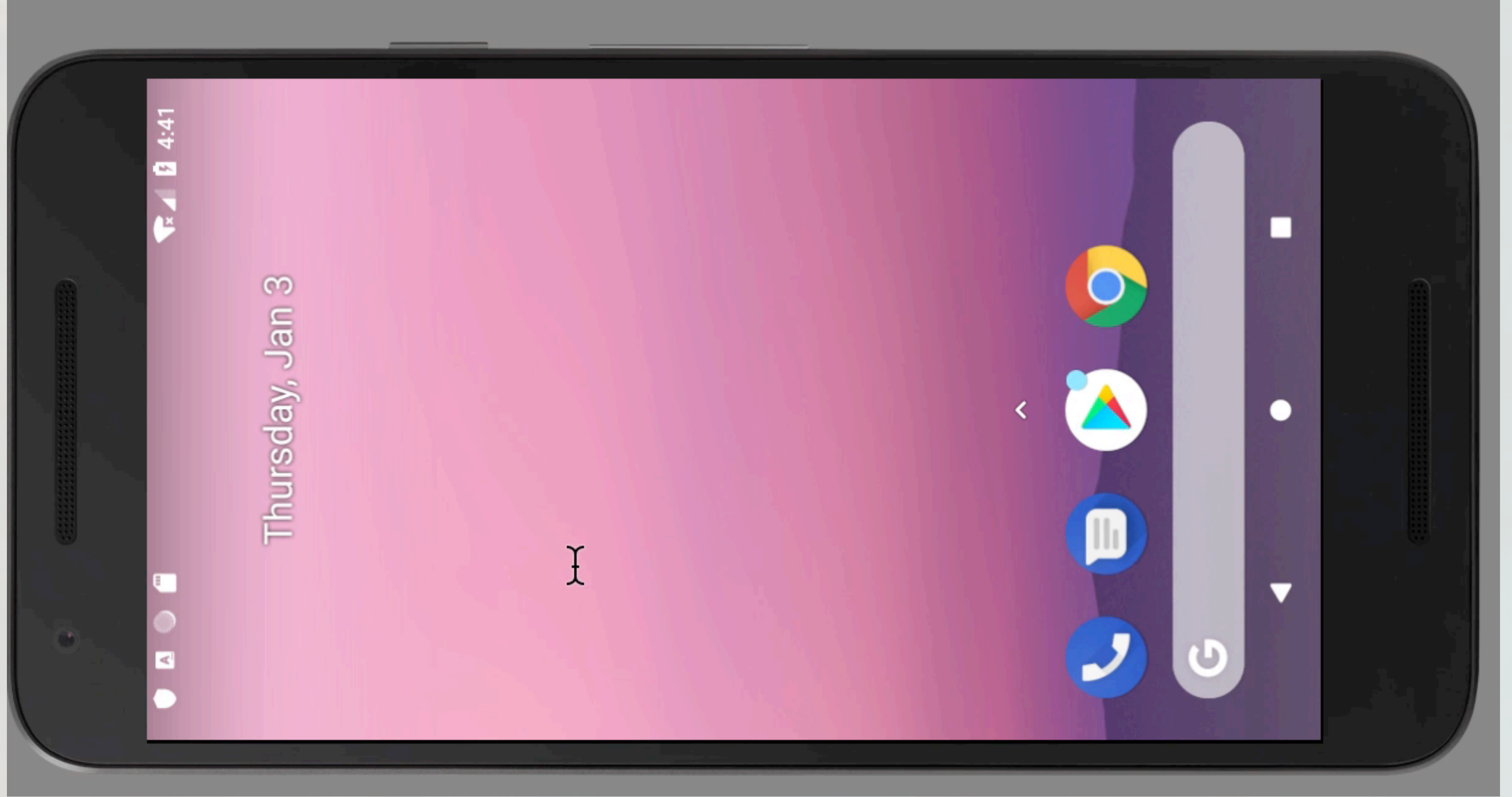


Project: Crossing Roads

Renault@lrde.epita.fr



Demo Video



Global Description



Objectives

- 📱 Teach you Android by coding a simple Game (~25 hours of work)
- 📱 Make you familiar with Map and (reverse) geolocalisation
 - ▶ **Remember to register for Google API**
 - ▶ **and activate geolocalisation and reverse geolocalisation**
- 📱 Force you to deal with Web communication
 - ▶ **You will have to deal with UI thread**



Mandatory (15 pts)

- 📱 What you can see on the demo video
- 📱 4 screens
 - ▶ **the world**
 - ▶ **the game**
 - ▶ **the profile**
 - ▶ **the statistics**

World's Details



Level Description

- 📍 A Level is composed of
 - ▶ The name of the level (levelname): for instance Paris
 - ▶ The latitude
 - ▶ The longitude
 - ▶ The difficulty (an integer between 1 and 5)
 - ▶ The url of the background
 - ▶ The url of the car (left to right)
 - ▶ The url of the car (right to left)
 - ▶ The url of the pin (for the map)

Data can be fetched using asynchronous task or threads

<https://www.lrde.epita.fr/~renault/teaching/ppm/game.txt>

- 📍 Each field is separated by a #
- 📍 Each level description is in only one line!



This file will be update for the evaluation!

- 📍 Only new levels can be added, no modifications for prior levels

Game Details



Important: the roads will be always positioned in exactly the same position



You have to generate and move cars

- 🕒 Best idea: have a list of cars that are repeatedly updated
- 🕒 User Timer.scheduleAtFixedRate method to move cars



You must deal with the bird

- 🕒 Cannot go backward!
- 🕒 Detect long press and move the bird consequently
- 🕒 Detect X-axis moving to force left or right



Detect Collision (smooth mode is ok)

Result's Details



A list of 4 elements

- name
- hometown
- time
- level name

<https://www.lrde.epita.fr/~renault/teaching/ppm/results.txt>

- Each field is separated by a #
- Each result description is in only one line!



The user must be geolocated to display results






Note that a player can play multiple time before accessing the results list




Profile Details and Music

141

Profile is the Simplest screen

-  A picture
-  A name
-  A motto

The music is playing all around the game

-  No restart when changing screen (please!)
-  Use a Service to do that
-  You can have songs for each level if you want it

You are free to choose the bird (or other) you like the most

Tips

- Reverse geocoding from your actual position

https://maps.googleapis.com/maps/api/geocode/json?latlng=48.9167,2.2&key=YOUR_APIKEY

- FloatingButton for actions in the map
- Fix the orientation to landscape so that you don't have to handle rotation
- Use Handler for moving the bird and onTouchEvent
- Fix the size of your images
- If you opt for multiples activity startActivityForResult could be a good choice

Options (1/2)

143



Support rotation on all screens of the App. (2.5 pts)

- ▶ During the game your bird must be correctly relocated



Persistence across re-opening (2.5 pts)

- ▶ Save results, name, and motto and other stuff



Score sharing (2.5 pts)

- ▶ At the end of the game or by selection on the score screen



Widget for displaying the bests results (2.5 pts)



Allow user to create its own level (2.5 pts)

- ▶ Create background, fix image for cars, etc.

Options (2/2)

144



Sort list of results (2.5 pts)

- ▶ According to user name, levels, time, location (all options must be available)



Offer the possibility to record a game (2.5 pts)

- ▶ And allows the user to browse the videos
- ▶ Be careful to lag...



Connect with iOS through "Bonjour" (2.5 pts)

- ▶ Share results with "bonjour"
- ▶ Develop an iOS application that displays these results



Other ideas welcome !

- ▶ Just ask me so I can validate your option

Submission

145


An archive *.tar.gz containing

 README.txt: this file must be at the root of your archive

- ▶ Explain your architecture
- ▶ Describe what are the implemented options
- ▶ Describe the problems you have encountered

 The APK of your application: this file must be at the root of your archive

- ▶ You can find it

 The source of your application

- ▶ Clean them!
- ▶ Document them!

The name of your project must be Student1_Student2

Summary



A simple application

- ▶ Less than one day of work for an experimented developer
- ▶ Regroup all elements required by a modern application
 - ▶ **Connection to Google API**
 - ▶ **Parsing**
 - ▶ **Touches events**
 - ▶ **Multiple Sceens**



From this application you can build your own

- ▶ **Pokemon Go**
- ▶ **Ingress**
- ▶ **Crossy Road**
- ▶ **Temple Run**



Lists





Renault@lrde.epita.fr



Using Lists



The most used component in Android Applications




-  Contacts
-  Mails
-  Pictures
-  Social Network



Ready-to-use component



Model-View-Controller

-  Model: the data to be displayed
-  View: the GUI of the cell
-  Controller: the relation between a data and its display

ListActivity



A ListActivity is the simplest component to build a list

- A unique ListView
- No need to define a layout for each cell
- Abstractions to manipulation the underlying list
 - ▶ notifyDataSetChanged: datas have been modified, the GUI must be refreshed
 - ▶ notifyDataSetInvalidated: invalidate the underlying datas





An Adapter is responsible of the management between the model and the view

- use a DataSetObserver to track modifications
- implement android.widget.adapter

ArrayAdapter

A ready-to-use adapter

-  requirements: your data must be organized as an array
-  Your data will be displayed using the toString method

```
public class MainActivity extends ListActivity {
```

```
    ArrayAdapter<String> adapter;
```

```
    String[] values = {
```

```
        "Rambo I", "Rambo II", "Rambo III", "Rambo V",
```

```
        "Die Hard I", "Die Hard 2", "Die Hard 3",
```

```
        "Die Hard 4"};
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

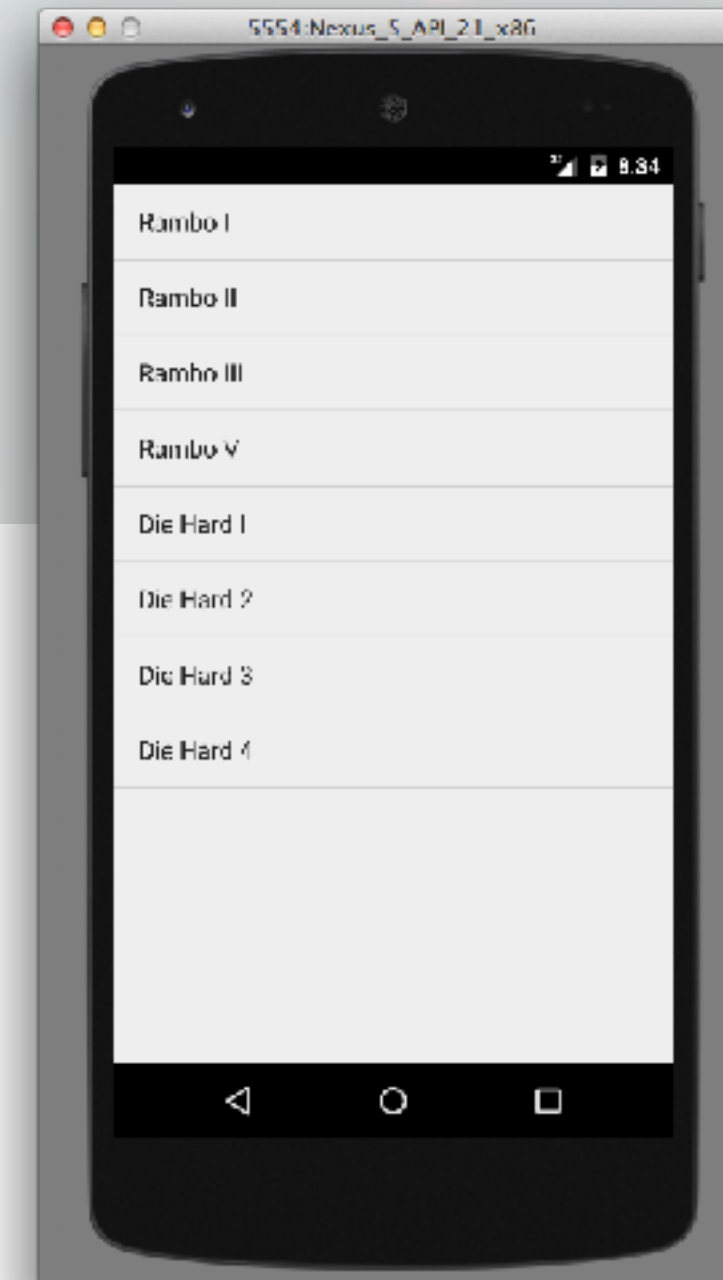
```
        adapter = new ArrayAdapter<String>(this,
```

```
            android.R.layout.simple_list_item_1, values);
```

```
        setListAdapter(adapter);
```

```
    }
```

```
}
```



ListActivity and User Event



Detect a click

```
@Override
public void onItemClick(ListView l, View v,
                        int position, long id) {
    // do something with the data
    Log.v("MainActivity", values[position]);
}
```



Detect a long click

```
getListView().setOnItemLongClickListener(
    new AdapterView.OnItemLongClickListener() {
        @Override
        public boolean onItemLongClick(AdapterView<?> parent,
                                       View view, int position, long id) {
            return true;
        }
    });
```

Slightly More complex Lists



Handle lists with multiple text fields

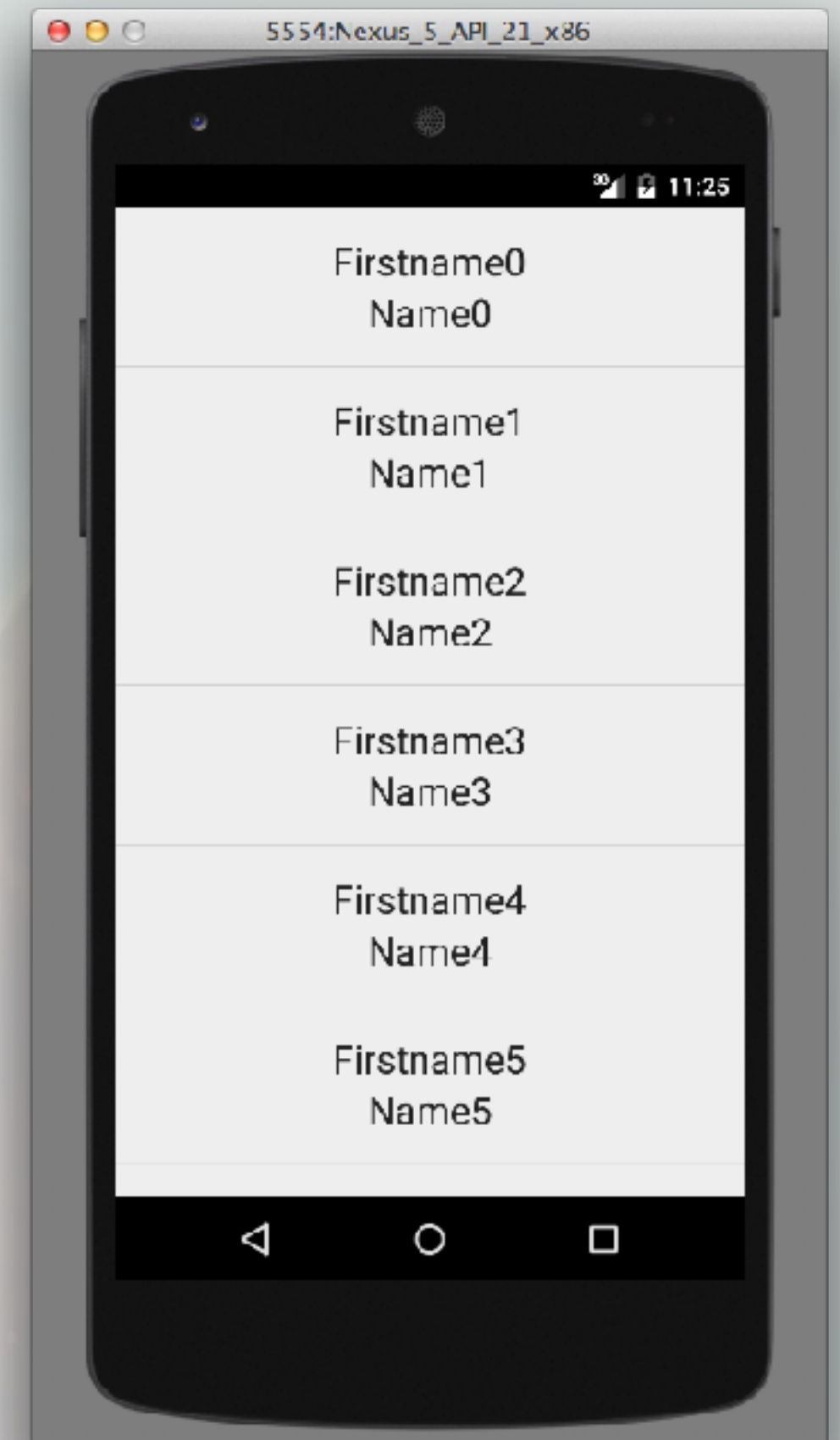


A GUI for each cell



A specific adapter for matching the elements of the view

**SimpleAdapter is
the solution**



SimpleAdapter

 Binding between static data and static GUI description

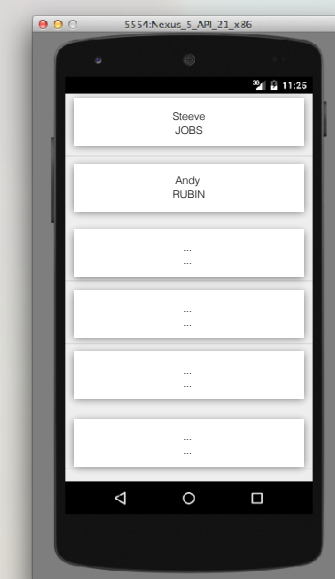
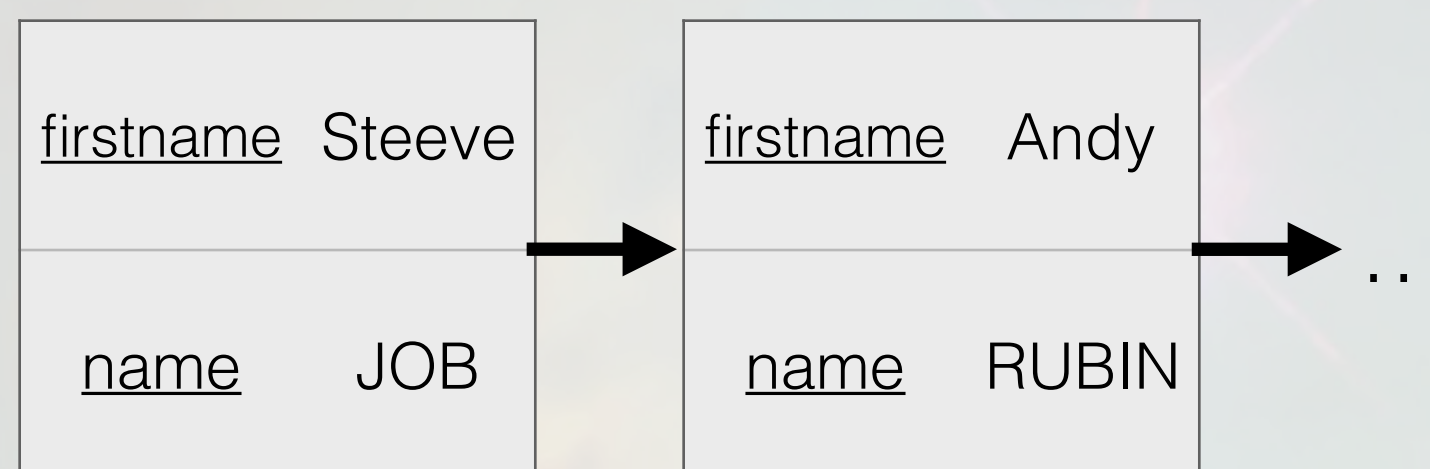
 Data must be a list of associative containers

 Each container is one row to display

 Each container contains the description of a cell

INPUT

OUTPUT



Define a Cell GUI

<LinearLayout

```
xmlns:android="http://schemas.android.com/apk/res/android"  
android:layout_width="match_parent"  
android:layout_height="match_parent"  
android:orientation="vertical">
```

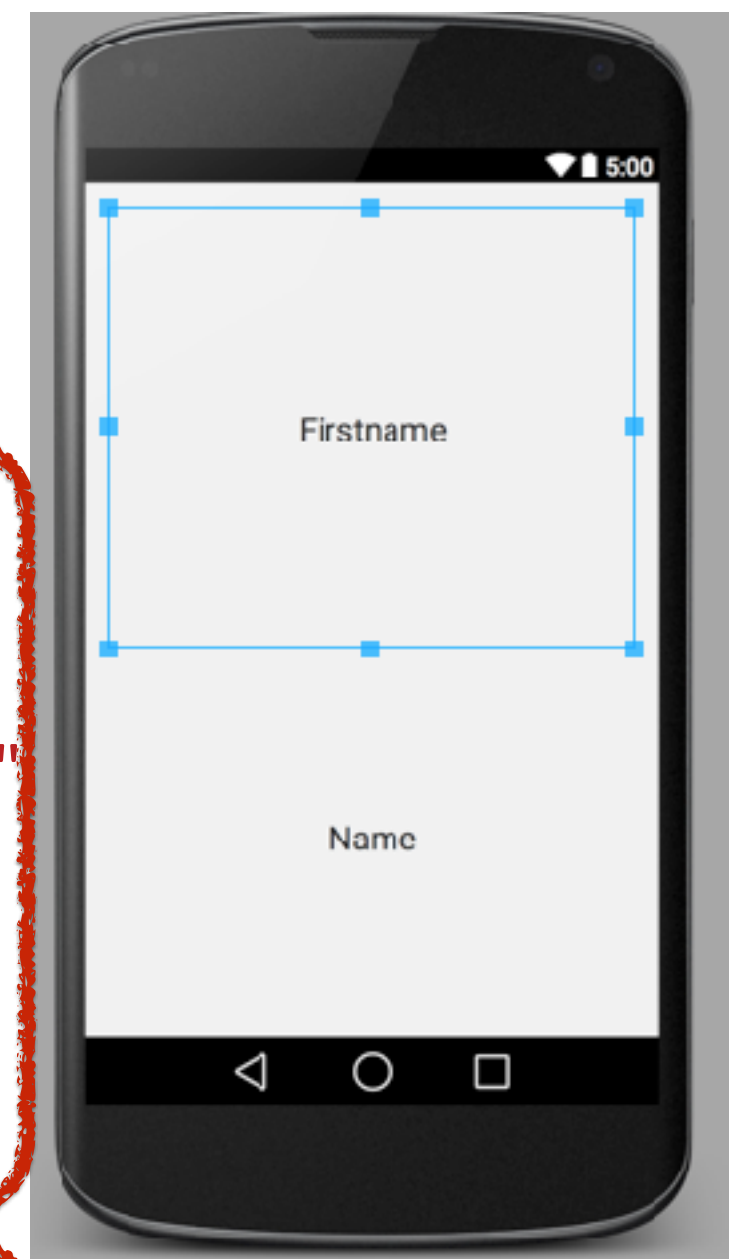
<TextView

```
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:textAppearance="?android:attr/textAppearanceLarge"  
android:text="Firstname"  
android:id="@+id/textView_firstname"  
android:layout_weight="0.50"  
android:gravity="center_vertical|center_horizontal" />
```

<TextView

```
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:textAppearance="?android:attr/textAppearanceLarge"  
android:text="Name"  
android:id="@+id/textView_name"  
android:layout_weight="0.50"  
android:gravity="center_vertical|center_horizontal" />
```

</LinearLayout>



Populate the List

```
SimpleAdapter adapter;  
ArrayList<HashMap<String, String>> data =  
    new ArrayList<HashMap<String, String>>();
```

Declaration

```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    for (int i = 0; i < 20; ++i)  
        addItem("Name"+i, "Firstname"+i);
```

Prepare Data

```
adapter = new SimpleAdapter(  
    this,  
    data,  
    R.layout.cell_content,  
    new String[]{"name", "firstname"},  
    new int[]{R.id.textview_name, R.id.textview_firstname});
```

Instantiation

```
setListAdapter(adapter);
```

setup

```
private void addItem(String record_name, String record_fname) {  
    HashMap<String, String> item = new HashMap<String, String>();  
    item.put("name", record_name);  
    item.put("firstname", record_fname);  
    data.add(item);
```

Prepare
Data

More Complex Lists

Overload ArrayAdapter

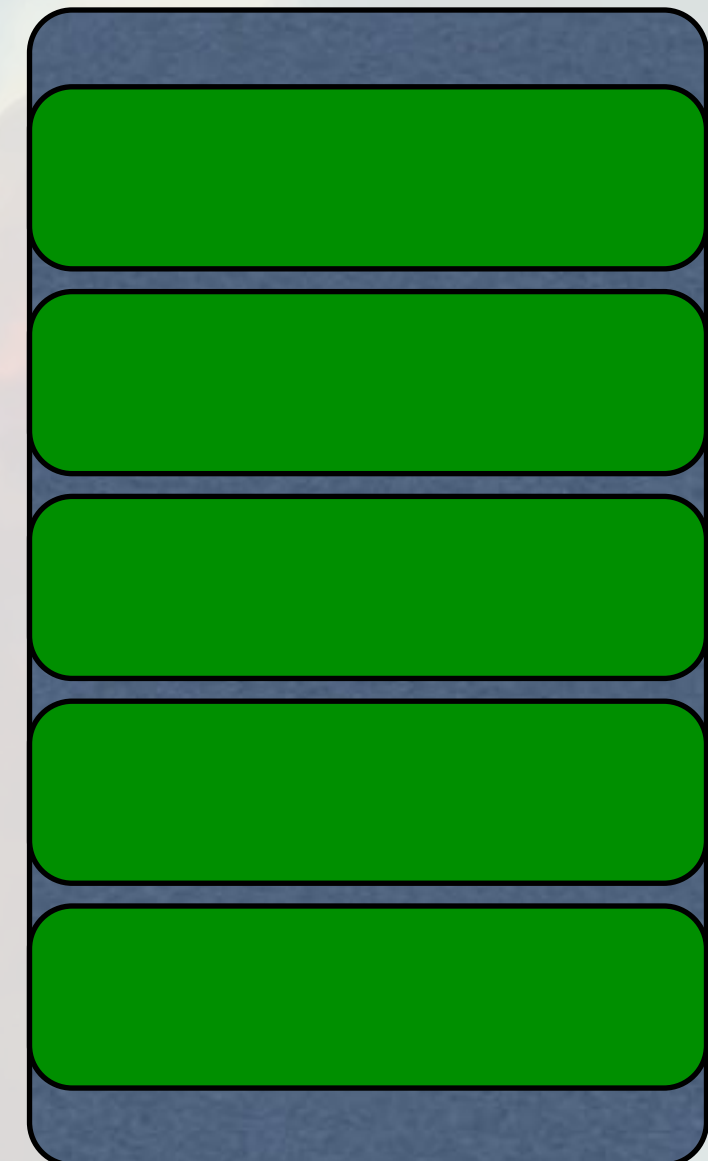
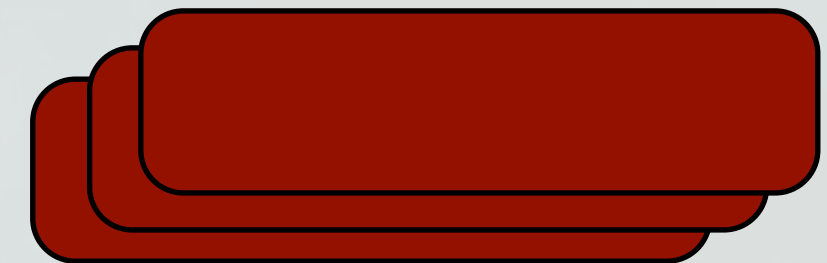
Specify dynamically each cell

 getView(int position, View cV, View parent):

- ▶ position: index of the element in the list
- ▶ cV: reusable View, if different from null

Recycle Views:

-  Avoid excessives allocations
-  Reset the view before usage!



Define a complex Cell GUI (rowlayout)

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <ImageView
        android:contentDescription="row"
        android:id="@+id/icon"
        android:layout_width="60dp"
        android:layout_height="60dp"
        android:layout_marginLeft="6dp"
        android:layout_marginRight="6dp"
        android:layout_marginTop="5dp"
        android:layout_marginBottom="5dp"
        android:src="@drawable/ic_launcher_foreground" >
    </ImageView>

    <TextView
        android:id="@+id/label"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginTop="20dp"
        android:layout_marginBottom="20dp"
        android:layout_marginLeft="10dp"
        android:text="text"
        android:textSize="26sp" >
    </TextView>

</LinearLayout>
```

Overload ArrayAdapter

```
public class MyAdapter extends ArrayAdapter<String> {
```

```
    private Integer[] images = {  
        R.drawable.one, R.drawable.two, R.drawable.three  
    };
```

Import these
ressources !

```
@Override
```

```
public View getView(int position, View convertView, ViewGroup parent) {  
    LayoutInflater inflater = (LayoutInflater)  
        getContext().getSystemService(Context.LAYOUT_INFLATER_SERVICE);
```

```
    View rowView = inflater.inflate(R.layout.rowlayout, parent, false);
```

```
    TextView textView = (TextView) rowView.findViewById(R.id.label);  
    ImageView imageView = (ImageView) rowView.findViewById(R.id.icon);
```

```
    textView.setText(getItem(position));
```

```
    if(convertView == null )  
        imageView.setImageResource(images[position]);
```

```
    else  
        rowView = (View)convertView;
```

```
    return rowView;
```

```
public MyAdapter(Context context, String[] values) {  
    super(context, R.layout.rowlayout, values);
```

```
}
```

call the super
constructor

```
}
```


Setup the Adaptator

```
public class MainActivity extends ListActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        String[] values = new String[] { "one", "two", "three" };  
        MyAdapter a = new MyAdapter(this, values);  
        setListAdapter(adaptateur);  
    }  
}
```



**Your List is ready to
be used !**

Header for the view



Adding an Header for your list is easy

- Just define R.layout.header
- and use `addHeaderView`

```
ListView lv = getListView();  
LayoutInflater inflater = getLayoutInflater();  
View header = inflater.inflate(R.layout.header, lv, false);  
lv.addHeaderView(header, null, false);
```



You can also use `addFooterView` to define a footer

Summary

A lot of adapter exist

 SimpleAdapter:

- ▶ **static binding between data and GUI of a row**


 ArrayAdapter:

- ▶ **If not overloaded, acts like a simple adapter**
- ▶ **Otherwise can be specified to handle complex views**

 SimpleCursorAdapter and CursorAdapter

- ▶ **Create lists from data base**

These Adapter works the same in every activity even if not ListActivity.

 You just have to grab the reference to the listView and apply setListAdapter



Persistent Data

Renault@lrde.epita.fr






Persistent Data

The bundle used during rotation cannot be used!

-  Only focuses on currently active datas
-  It does not survive to application closing/reopening

You have to chose between on of these

-  SharedPreferences
 - ▶ **might be slow!**
-  FileSystem
 - ▶ **Internal: the memory of the device**
 - ▶ **External: the SD card (if available)**
 - ▶ **Cloud: on the internet**
-  Database
 - ▶ **Sqlite3**
 - ▶ **...**

SharedPreferences

Key-value pair of primitive data types

 String / some_basic_type

One XML file per activity

 Name can be fixed

 Options for this file

- ▶ **MODE_PRIVATE**: access only for the activity
- ▶ **MODE_WORLD_READABLE**: read access for everyone
- ▶ **MODE_WORLD_WRITABLE**: write access for everyone

Acts like Windows registry

Name may be confusing!

 Not for saving ringtone selected by the user but to save any kind of simple data

Manipulating PreferencesFile (1/2)

Naming your Preferences file

 File will be saved at com.example.myapp.MyPreferenceFile

```
public static final String PREFS_NAME = "MyPreferenceFile";
```

```
Context context = getActivity();  
SharedPreferences sharedPref =  
    context.getSharedPreferences(PREFS_NAME,  
    Context.MODE_PRIVATE);
```

Reading in the file

```
SharedPreferences settings =  
    getSharedPreferences(PREFS_NAME, Context.MODE_PRIVATE);  
advertising = settings.getBoolean("withAdvertising", false);
```

Manipulating PreferencesFile (2/2)

Writing in the file

```
SharedPreferences.Editor editor = settings.edit();
editor.putBoolean("withAdvertising", advertising);
// Commit the edits!
editor.commit();
```

 Use apply to propagate your changes atomically

 Use commit to propagate your changes asynchronously

Use the internal Memory

171



Default behavior: files are private to the application

- Not accessible from the user
- Not accessible from other applications
- When an application is removed, all the associated files are also removed



A dedicated directory per application

- Created when the application is installed
- All files created by the application are in this directory
- We can have access to this directory programmatically
 - ▶ **getFilesDir(): the directory**
 - ▶ **getFileList(): the list of all files created by the application**
 - ▶ ...

Writing in a file (internal memory)

172

Fix read / write mode

```
String FILENAME = "hello_file";
String string_hello = "hello world!";
FileOutputStream fos = null;
try {
    fos = openFileOutput(FILENAME, Context.MODE_PRIVATE);
    fos.write(string_hello.getBytes());
    fos.close();
} catch (Exception e) {
    e.printStackTrace();
}
```

Context.MODE_APPEND for appending a current file

getCacheDir: helps to manipulate temporary files

Reading from a file (internal memory)

```
private String readFromFile() {
    String ret = "";
    try {
        InputStream input = openFileInput(FILENAME);
        if ( input != null ) {
            InputStreamReader inputSR = new InputStreamReader(input);
            BufferedReader bufferedReader = new BufferedReader(inputSR);
            String receiveString = "";
            StringBuilder stringBuilder = new StringBuilder();
            while ( (receiveString = bufferedReader.readLine()) != null )
            {
                stringBuilder.append(receiveString);
            }
            input.close();
            ret = stringBuilder.toString();
        }
    }
    catch (Exception e) {
    }
    return ret;
}
```

Memory that can be addressed by the User

174

 Application's directories are private

 Some directories are shared

 DIRECTORY_PICTURES

 DIRECTORY_RINGTONES

 DIRECTORY_MUSIC

 ...

```
public File getAlbumStorageDir(String albumName) {  
    // Get the dir. for the user's public pictures directory.  
    File file =  
        new File(Environment.getExternalStoragePublicDirectory(  
            Environment.DIRECTORY_PICTURES), albumName);  
    if (!file.mkdirs()) {  
        Log.e(LOG_TAG, "Directory not created");  
    }  
    return file;  
}
```

External Memory

 Can be removed anytime!

 Specify AndroidManifest.xml

```
<uses-permission  
    android:name=« android.permission.WRITE_EXTERNAL_STORAGE »  
>
```

 Test if media is mounted or not

```
String state = Environment.getExternalStorageState();  
if (Environment.MEDIA_MOUNTED.equals(state)) {  
    // some stuff...  
}  
else if (Environment.MEDIA_MOUNTED_READ_ONLY.equals(state)) {  
    // some other stuff  
}  
else {  
    // What to do here?  
}
```



Data can be stored in a database

- Full support for SQLite
- All data base are private
- Have a look to
 - ▶ **SQLDataBase**
 - ▶ **SQLiteQueryBuilder**
- Android suggests to add unique ID for each record



Data can be stored in the cloud

- Warning! Internet is not always available
- Use java.net and android.net
- Think to require permissions in AndroidManifest.xml

Summary



Use SharedPreferences for saving small datas

- 📱 Limited to basic types



Use Filesystem otherwise

- 📱 Temporary files
- 📱 Internal memory
 - ▶ **Always available**
- 📱 External memory
 - ▶ **Not always available**



Use database

- 📱 When you have a lot of structured data to store



Use Cloud and related API if your app consume a lot of memory



User Preferences

Renault@lrde.epita.fr



Configuration of your Application



A dedicated screen to configure the application



Great user experience



Can be built by-hand



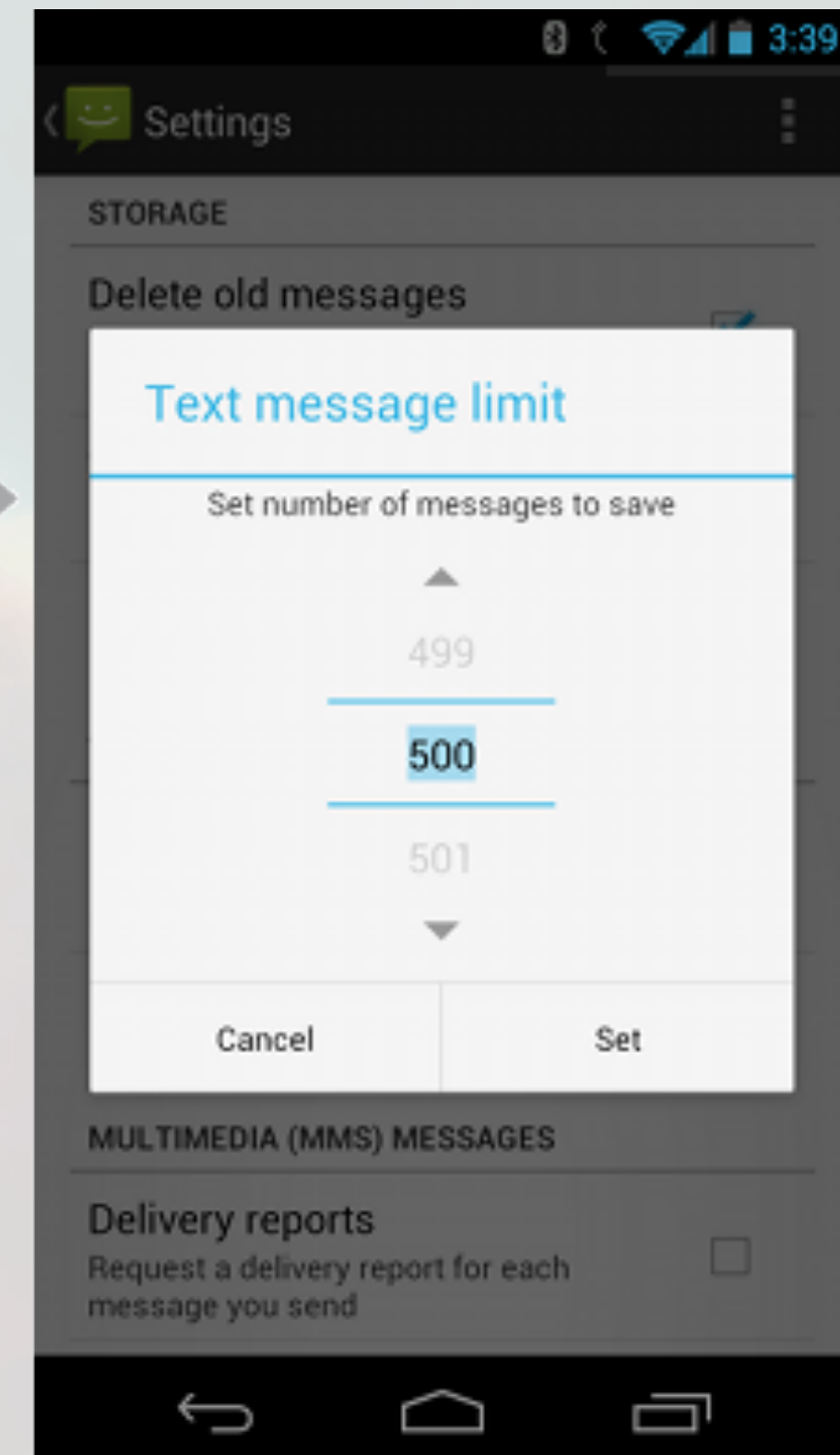
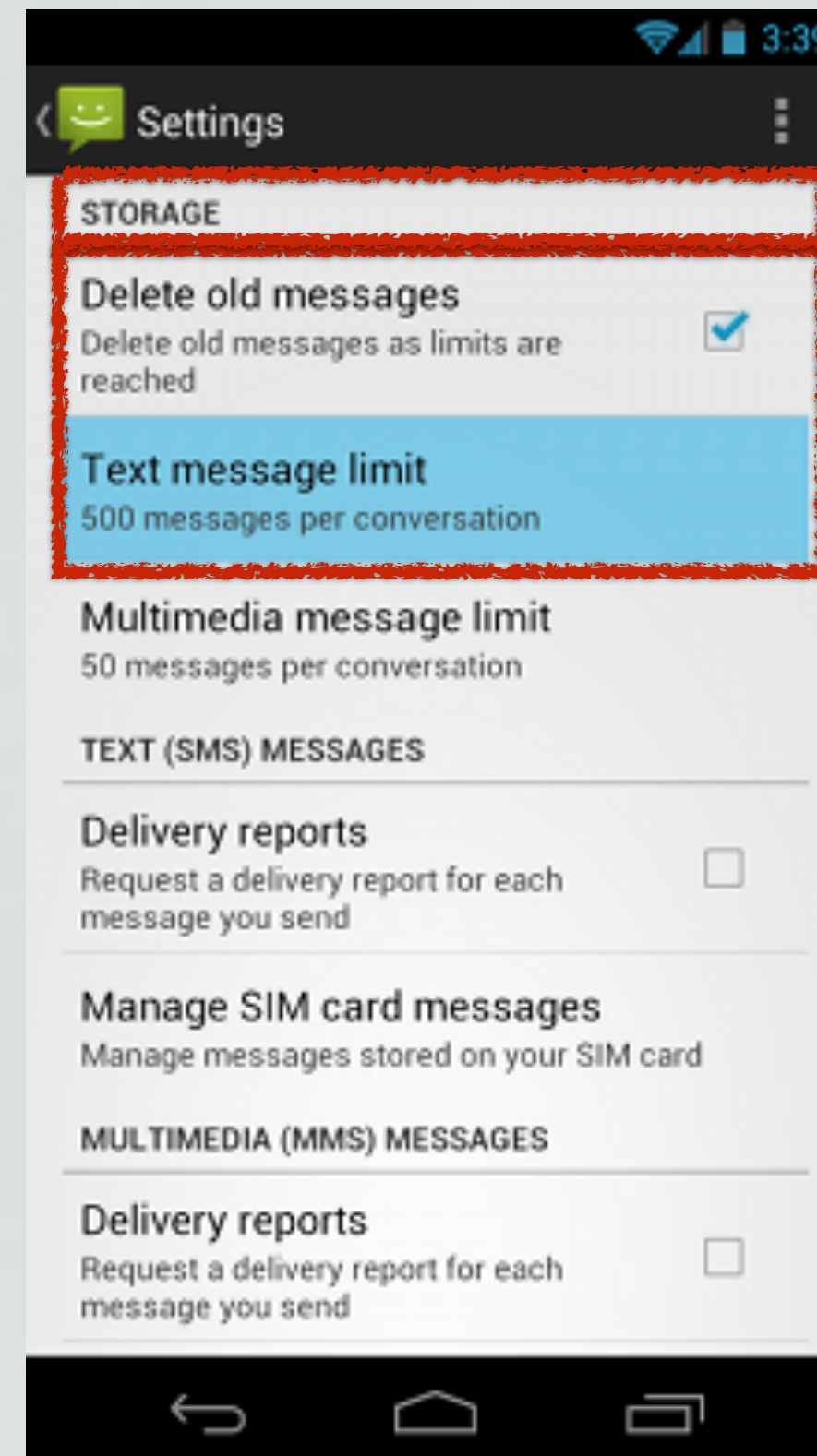
Not homogeneous with the other applications



BUT ...



Android is already providing necessary frameworks



Guidelines

Everything is configurable

 BUT you have to restrict configuration for your User

▶ **Otherwise your application will be too complex**

Guidelines according to API Level

< API 7	No groups
API 8 à API 10	Groups of two maximum
API 11 à API 15	Groups four maximum
API 16 +	Create a dedicated scree for groups of more than four elements

PreferenceActivity



Base class for building a hierarchy of preferences

- 📱 On small screen only displays titles
- 📱 On large screen, a two views are displayed side by side



Preferences are saved in SharedPreferences







Implement onBuildHeaders to separate headers and group of preferences

- 📱 A preference may have no group!

Simple Preferences (1/2)

A preference is composed of

-  a title: describing the preference
-  a summary: explaining what is the purpose of the preference
-  a key: that will be used by SharedPreferences
-  a default value: for the first launch of the application

Define a preference through an XML file

```
<?xml version="1.0" encoding="utf-8"?>
<PreferenceScreen
    xmlns:android="http://schemas.android.com/apk/res/android">
  <CheckBoxPreference
    android:key="@string/prefs_synchro"
    android:title="@string/synchro"
    android:summary="Manage Synchronisation"
    android:defaultValue="true" />
</PreferenceScreen>
```


Simple Preferences (2/2)



Inherits from Preference

```
public class SettingsAc
    @Override
    public void onCreate
        super.onCreate
        addPreferences
    }
}
```



```
PreferenceActivity{
    (onPreferenceState) {
    };
    (addPreferences);
}
```



Declare the class

```
<activity android:name="
```

AndroidManifest.xml

```
"></activity>
```



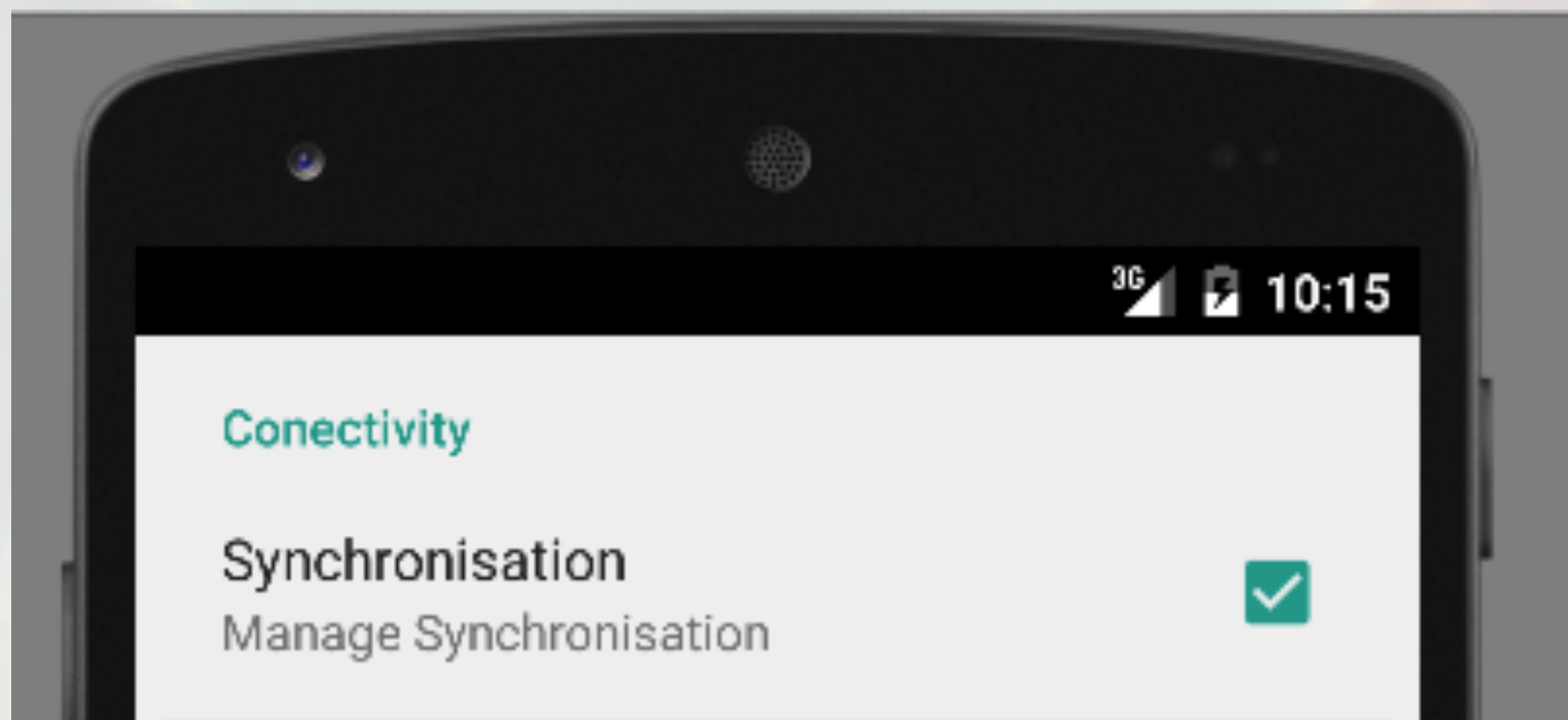
Run this class from

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

Groups of Preferences

Define a group of Preferences

```
<PreferenceCategory
    android:key="pref_key_storage_settings"
    android:title="Conectivity">
    ...
</PreferenceCategory>
```



Run Intents from Preferences

Only specify it into the XML

```
Preference android:title="@string/prefs_web_page" >  
    <intent android:action="android.intent.action.VIEW"  
        android:data="http://www.example.com" />  
</Preference>
```

You can specify any kind of Intents

-  Just remember to declare the target activity into the AndroidManifest.xml

Manipulating Preferences (1/2)

Grab the value of a preference

```
SharedPreferences sharedPref =  
    PreferenceManager.getDefaultSharedPreferences(this);  
String syncConnPref =  
    sharedPref.getString(String.valueOf(R.string.synchro), "");
```

Register to watch modifications

 registerOnSharedPreferenceChangeListener:

▶ in the onStart method

 unregisterOnSharedPreferenceChangeListener:

▶ in the onStop method

 Implement OnSharedPreferenceChanged

Manipulating Preferences (2/2)



Callback when a Preference is changed

```
public void onSharedPreferenceChanged
    (SharedPreferences sharedPreferences, String key) {
    if (key.equals(R.string.synchro)) {
        Preference connectionPref = findPreference(key);
        // Set summary to be the user-description
        // for the selected value
        connectionPref.setSummary
            (sharedPreferences.getString(key, ""));
    }
}
```

Fragments of Preferences (1/2)

190



Define headers

```
<preference-headers
  xmlns:android="http://schemas.android.com/apk/res/android">
<header
  android:fragment="com.example.admin.preferenceapplication.SettingsA
  ctivity$Prefs1FragmentInner"
  android:title="Prefs 1"
  android:summary="Summary prefs 1" />
</preference-headers>
```



Define contents

```
<?xml version="1.0" encoding="utf-8"?>
<PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/
  android">
  <CheckBoxPreference
    android:defaultValue="true"
    android:key="@string/prefs_synchro2"
    android:summary="Manage Synchronisation"
    android:title="@string/synchro2" />
</PreferenceScreen>
```

Fragments of Preferences (2/2)

Implement the fragment

```
public static class Prefs1FragmentInner
    extends PreferenceFragment {
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        addPreferencesFromResource(R.xml.fragmented_preferences);
    }
}
```

Modify the class extending PreferencesActivity to handle onBuildHeader and isValidFragment

```
public void onBuildHeaders(List<Header> target) {
    super.onBuildHeaders(target);
    loadHeadersFromResource(R.xml.preferences_headers, target);
}
protected boolean isValidFragment(String fragmentName) {
    return Prefs1FragmentInner.class
        .getName().equals(fragmentName);
}
```

Summary



Having the Android'Preferences look-and-feel is easy

- Just build an activity that inherits from PreferencesActivity
- The SharedPreferences component is used in background to save it



Split preferences to respect guidelines

- A title
- A content



DO NOT abuse of Preferences

- If an action tends to be frequently modified it may be useful to build a menu for that



Menus

Renault@lrde.epita.fr



Better User Experience

-  All action on the GUI are grouped

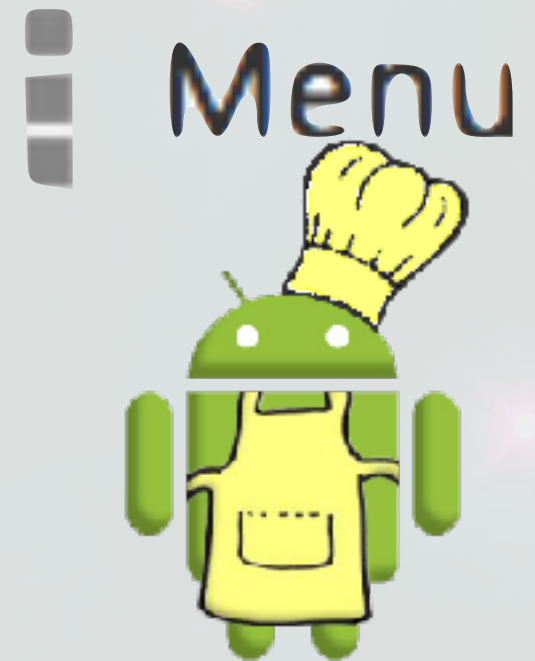
Since Android 3.0 the menu button is no longer mandatory

-  Before it was used to display 6 elements (maximum)

A menu has a GUI so it is described through an XML

A lot of menu exists

- | | |
|---|--|
|  ContextMenu |  OptionMenu |
|  PopupMenu |  ActionBar |
|  NavigatinDrawer |  ... |



Menu's Structure



Define an XML resource file to describe the structure

```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:app="http://schemas.android.com/apk/res-auto"
      xmlns:tools="http://schemas.android.com/tools"
      tools:context=".MainActivity">
  <item android:id="@+id/edit"    android:title="Edit" />
  <item android:id="@+id/remove"  android:title="Remove" />
</menu>
```

Then you have to use
an Inflater to build the
View

Popup Menu

 **The menu appears above the specified View**

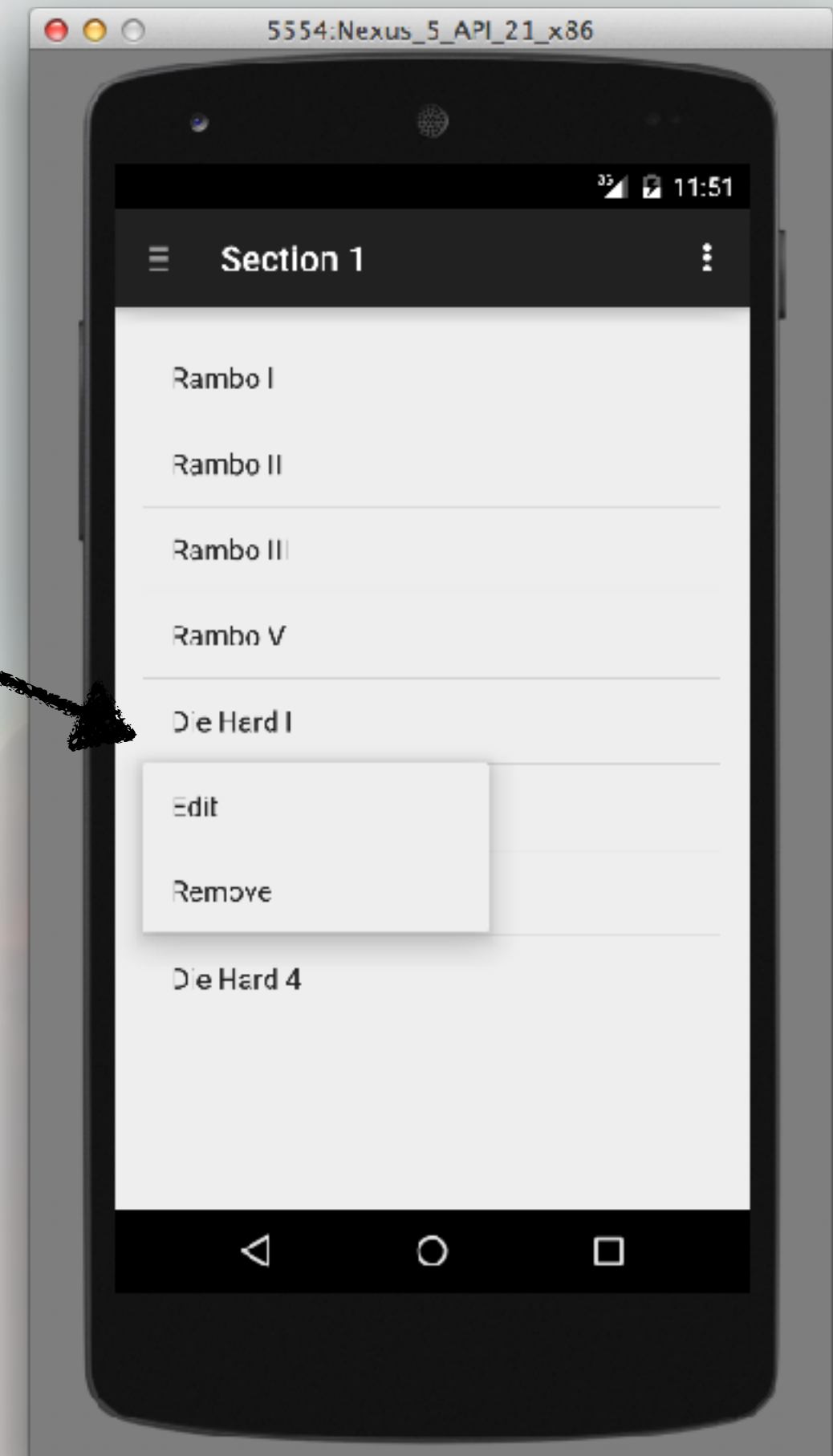
 **The simplest menu to build**

 **Trigger an action to display the menu**

 Long click for instance

 **Register then for clicks**

 Implements [MenuItem.OnMenuItemClickListener](#)



Popup menu: implementation

```
ListView listView = (ListView) rootView.findViewById(R.id.listView);
adapter = new ArrayAdapter<String>(getActivity(),
                                   android.R.layout.simple_list_item_1, values);
listView.setAdapter(adapter);
listView.setOnItemLongClickListener(new AdapterView.OnItemLongClickListener() {
    @Override
    public boolean onItemLongClick(AdapterView<?> parent,
                                    View view, int position, long id) {
        PopupMenu popupMenu = new PopupMenu(getActivity(), view);
        MenuInflater menuInflater = popupMenu.getMenuInflater();
        menuInflater.inflate(R.menu.mymenu, popupMenu.getMenu());
        popupMenu.getMenu().findItem(R.id.edit)
            .setOnMenuItemClickListener(
                new MenuItem.OnMenuItemClickListener() {
                    @Override
                    public boolean onMenuItemClick(MenuItem item) {
                        Toast.makeText(getActivity(),
                                      item.getTitle(), Toast.LENGTH_LONG).show();
                        return false;
                    }
                }
            );
        popupMenu.show();
        return true;
    }
});
```

Contextual Menus



Dedicated actions for specific views

Can be associated to any kind of Views



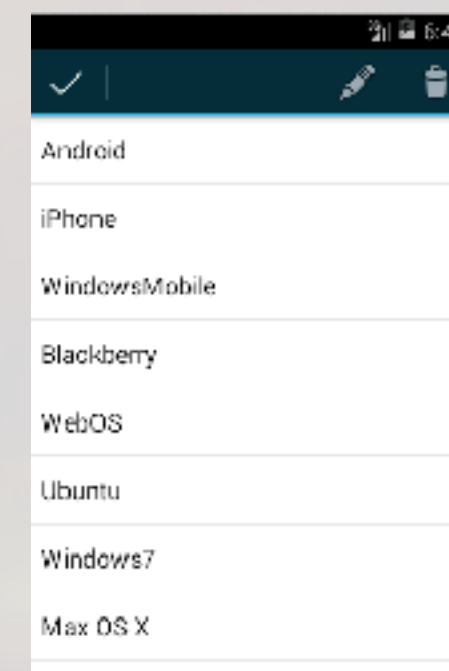
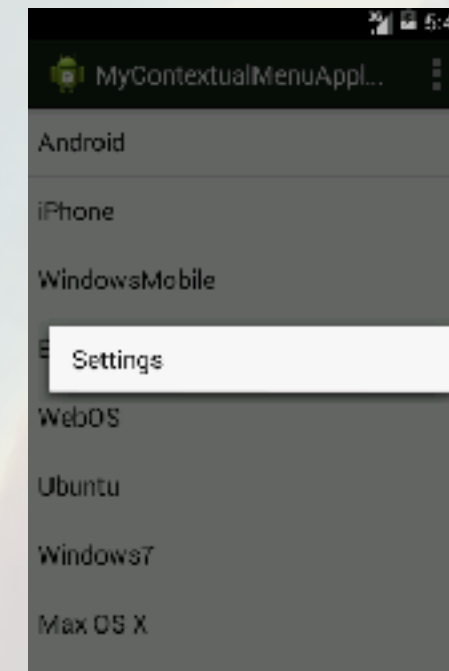
Two kind of contextual menus

Floating menus:

- ▶ close to popup menu
- ▶ the menu is displayed by a long press

ActionMode:

- ▶ one action bar is displayed with all possible actions



Floating Menu



Register the view as a receiver for context menus

```
registerForContextMenu(listView);
```



Overload onCreateContextMenu to trigger the contextual menu

 onContextItemSelected is useful to know which item has been clicked

```
@Override
public void onCreateContextMenu(ContextMenu menu, View v,
                               ContextMenu.ContextMenuInfo menuInfo)
{
    super.onCreateContextMenu(menu, v, menuInfo);
    getActivity().getMenuInflater()
        .inflate(R.menu.mymenu, menu);
}
```

Contextual Menu with ActionBar



Implement ActionMode.Callback

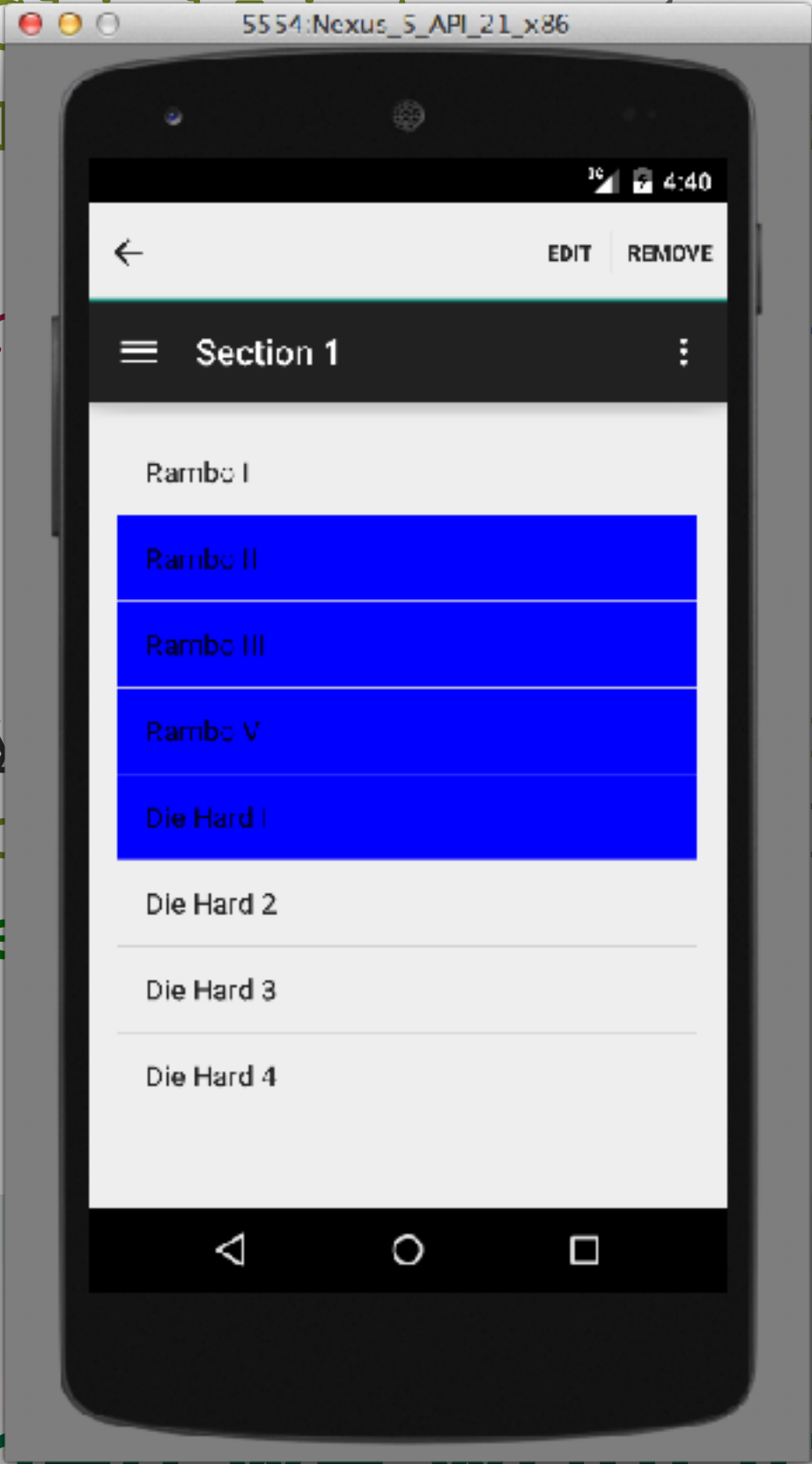
```
private ActionMode.Callback modeCallback = new ActionMode.Callback()
{
    @Override
    public boolean onCreateActionMode(ActionMode mode, Menu menu) {
        MenuInflater inflater = getActivity().getMenuInflater();
        inflater.inflate(R.menu.mymenu, menu);
        return true;
    }
    @Override
    public boolean onPrepareActionMode(ActionMode mode, Menu menu) {
        return false;
    }
    @Override
    public boolean onActionItemClicked(ActionMode mode, MenuItem item)
    {
        return false;
    }
    @Override
    public void onDestroyActionMode(ActionMode mode) {
    }
};
```

Use the Callback



Run the callback with a long press

```
listView.setOnItemClickListener(  
    new AdapterView.OnItemClickListener() {  
        @Override  
        public boolean onItemClick(  
            AdapterView.OnItemClickListener listener,  
            AdapterView parent,  
            View view,  
            int position, long id)  
        {  
            getActivity().  
            view.setBackgroundColor(  
                Color.BLUE);  
            return true;  
        }  
    });
```



```
listener() {  
    <(AdapterView<?> parent,  
    View view,  
    int position, long id)  
    > modeCallback);  
    Color.BLUE);
```



Very interesting when we want to perform an action on multiple views

Navigation Drawer



Hidden panel



Can be revealed



By a click



By a left-to-right swipe



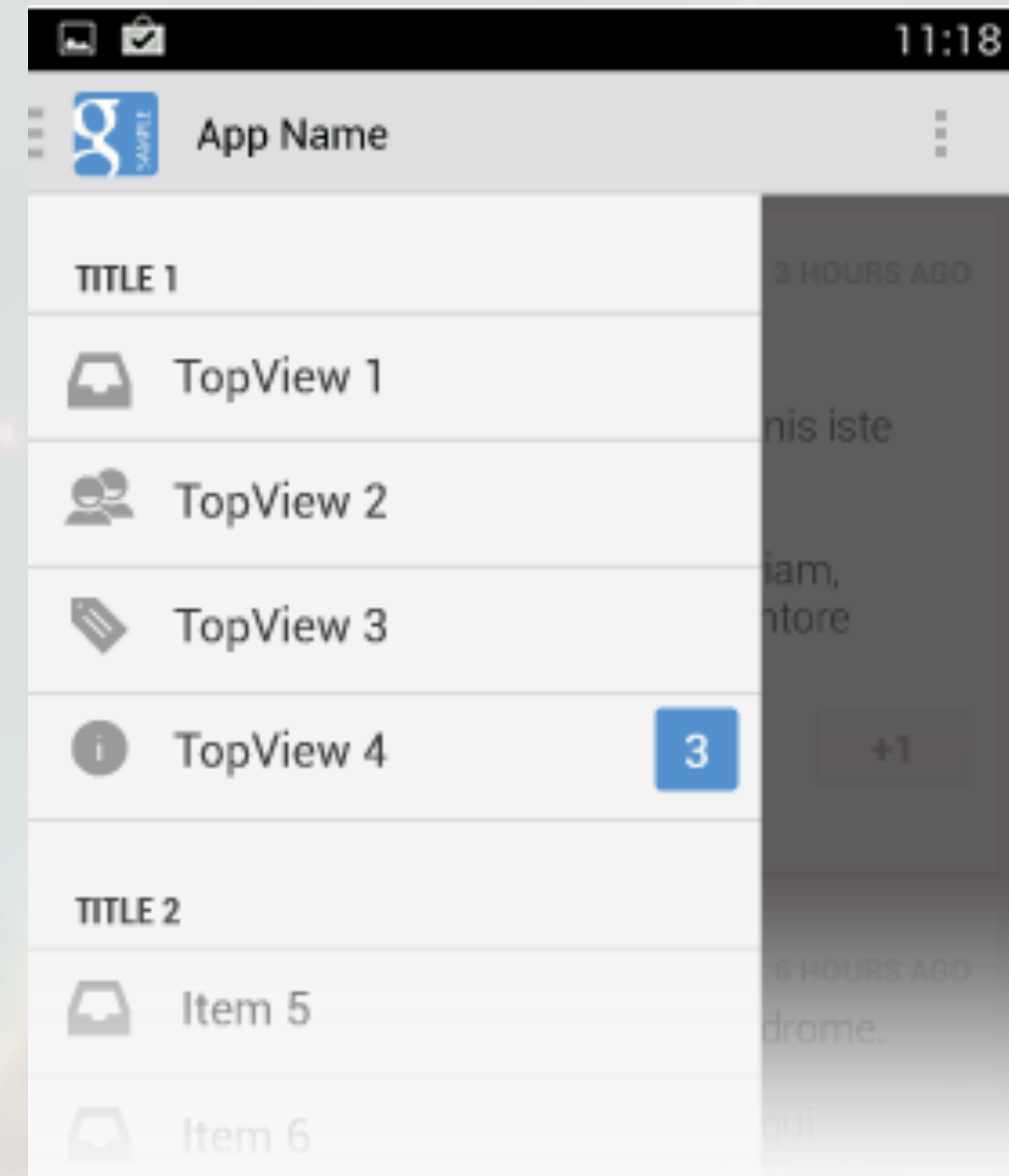
Group options, actions, informations



A full fragment



Guideline: the drawer must be opened during the first opening of the application



The main View

```
<android.support.v4.widget.DrawerLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/drawer_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <FrameLayout android:id="@+id/container"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

    <!-- The drawer is given a fixed width in dp and extends the full
        height of the container. -->
    <fragment android:id="@+id/navigation_drawer"
        android:layout_width="@dimen/navigation_drawer_width"
        android:layout_height="match_parent"
        android:layout_gravity="start"
        android:name="com.example.admin.mynavigationdrawerapplication
            .NavigationDrawerFragment"
        tools:layout="@layout/fragment_navigation_drawer" />
</android.support.v4.widget.DrawerLayout>
```

Navigation Drawer's View



Use a List for instance as the main View of the drawer

<ListView

```
xmlns:android="http://schemas.android.com/apk/res/android"  
xmlns:tools="http://schemas.android.com/tools"  
android:layout_width="match_parent"  
android:layout_height="match_parent"  
android:choiceMode="singleChoice"  
android:divider="@android:color/transparent"  
android:dividerHeight="0dp"  
android:background="#cccc"  
tools:context=".NavigationDrawerFragment" />
```

Manipulate the NavigationDrawer



Use openDrawer from the main activity

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
    mNavigationDrawerFragment =  
        (NavigationDrawerFragment) getSupportFragmentManager()  
            .findFragmentById(R.id.navigation_drawer);  
    DrawerLayout mDrawerLayout =  
        (DrawerLayout) findViewById(R.id.drawer_layout);  
    mDrawerLayout.openDrawer(Gravity.START);  
}
```



Dont forget to add dependencies in build.graddle

```
dependencies {  
    implementation 'com.android.support:appcompat-v7:28.0.0'  
    implementation 'com.android.support:design:28.0.0'  
}
```

Add Header to the NavigationDrawer



Specify the headerLayout attribute

```
<android.support.design.widget.NavigationView  
    android:id="@+id/nav_view"  
    android:layout_width="wrap_content"  
    android:layout_height="match_parent"  
    android:layout_gravity="start"  
    android:fitsSystemWindows="true"  
    app:menu="@menu/drawer_view"  
    app:headerLayout="@layout/nav_header" />
```



And the just define layout/nav_header

Integration with the ActionBar



A fine grained integration with the ActionBar is possible



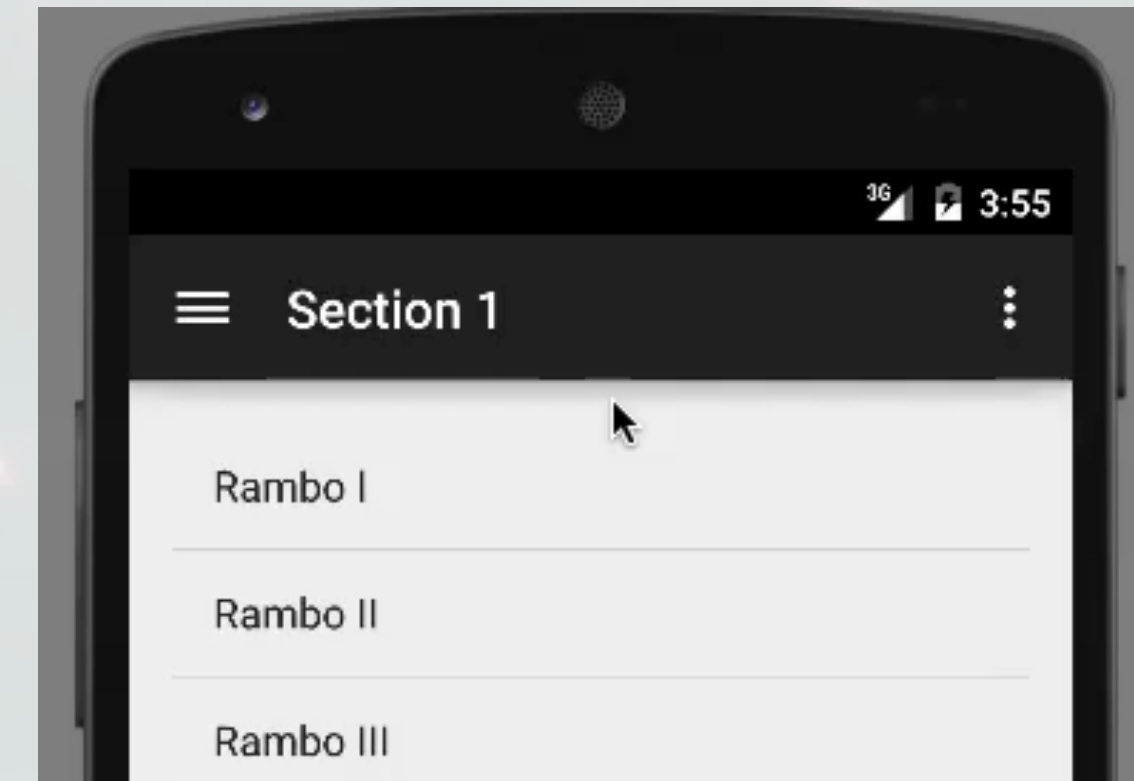
Use `ActionBarDrawerToggle`

Some effects are available only from [android.support.v7.app.ActionBarDrawerToggle](#)

Objectives: ease the interaction with the action bar

Ease the manipulation from the main activity

Android Studio is providing canvas for these projects



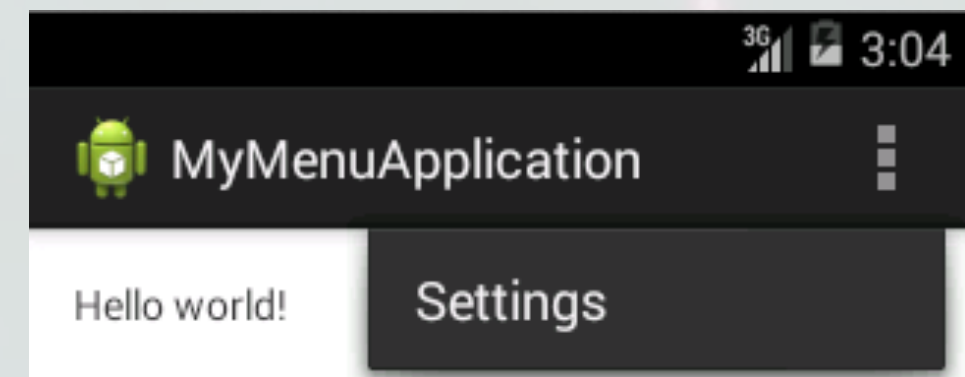
Setup DrawerToggle into NavigationDrawer

```
mDrawerToggle = new ActionBarDrawerToggle(  
    getActivity(),                               /* host Activity */  
    mDrawerLayout,                               /* DrawerLayout object */  
    //R.drawable.ic_drawer,                       /* nav drawer image to  
                                                replace 'Up' caret */  
    R.string.navigation_drawer_open,            /* "open drawer"  
                                                description for accessibility */  
    R.string.navigation_drawer_close           /* "close drawer"  
                                                description for accessibility */  
)  
{  
    @Override  
    public void onDrawerClosed(View drawerView) {  
        super.onDrawerClosed(drawerView);  
        // do something...  
    }  
  
    @Override  
    public void onDrawerOpened(View drawerView) {  
        super.onDrawerOpened(drawerView);  
        // do something  
    }  
}
```

Summary

211

 **The action Bar offers another menu which is highly configurable**



 **There are a lot of menus**

-  that can be mixed
-  almost fully configurable
 - ▶ **layout, highlight some elements**

 **Do not overload your application with menus**

 **Some menus are useful**

-  for navigation (NavigationDrawer)
-  for multiple editing (ActionMode.Callback)






ActionBar Activity

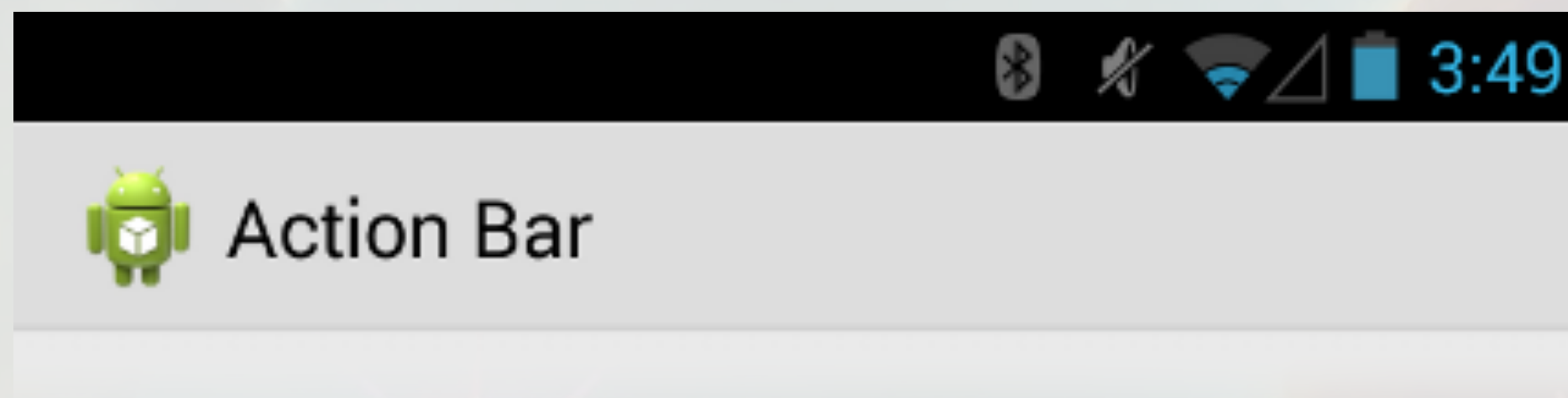
Renault@lrde.epita.fr




Description

Not a "classical" activity

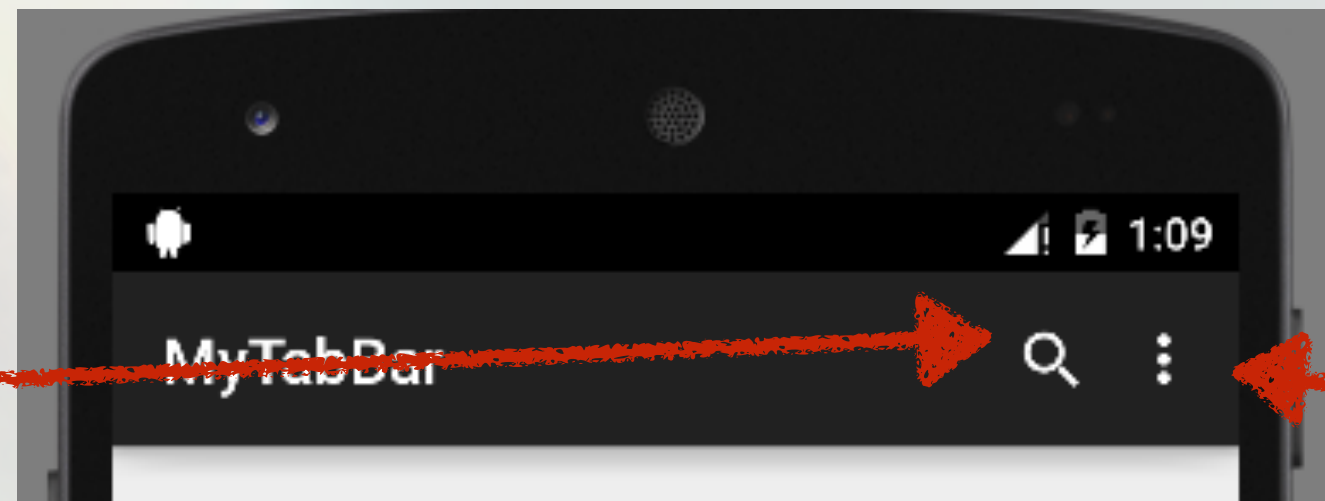
-  No full screen mode
-  Some informations stay visible
 - ▶ **Time**
 - ▶ **Battery**
 - ▶ **Connectivity**
-  Presents contextual informations: title, icon, ...



-  Easily configurable
 - ▶ **Add buttons**
 - ▶ **Add menus**

Adding Buttons

A button



Option Menu

Adding a button is simple

Just modify res/menu/menu_main.xml

```
<item android:id="@+id/action_settings_promote"  
      android:title="@string/action_settings"  
      android:icon="@drawable/abc_ic_search_api_mtrl_alpha"  
      android:orderInCategory="100"  
      app:showAsAction="ifRoom" />
```

Order them by importance

Buttons may be displayed only if rooms are available otherwise they are displayed with their names in the Option Menu

Order Buttons



Automatic layout for buttons according to two criteria



android:orderInCategory:

- ▶ **priority between the different elements**
- ▶ **smallest weights are low in the hierarchy**



app:showsAsAction:

- ▶ **main positioning options**
- ▶ **always: the button must appears in the Action Bar**
- ▶ **never: the button will never appears**
- ▶ **ifroom: button may appears**
 - **depends on the other button**
 - **depends on device's screen size**



Guidelines: do not force a button to be in the ActionBar

OptionsMenu



Inflate it when the user wants to inspect it

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action
    // bar if it is present.
    getMenuInflater().inflate(R.menu.menu_main, menu);
    return true;
}
```



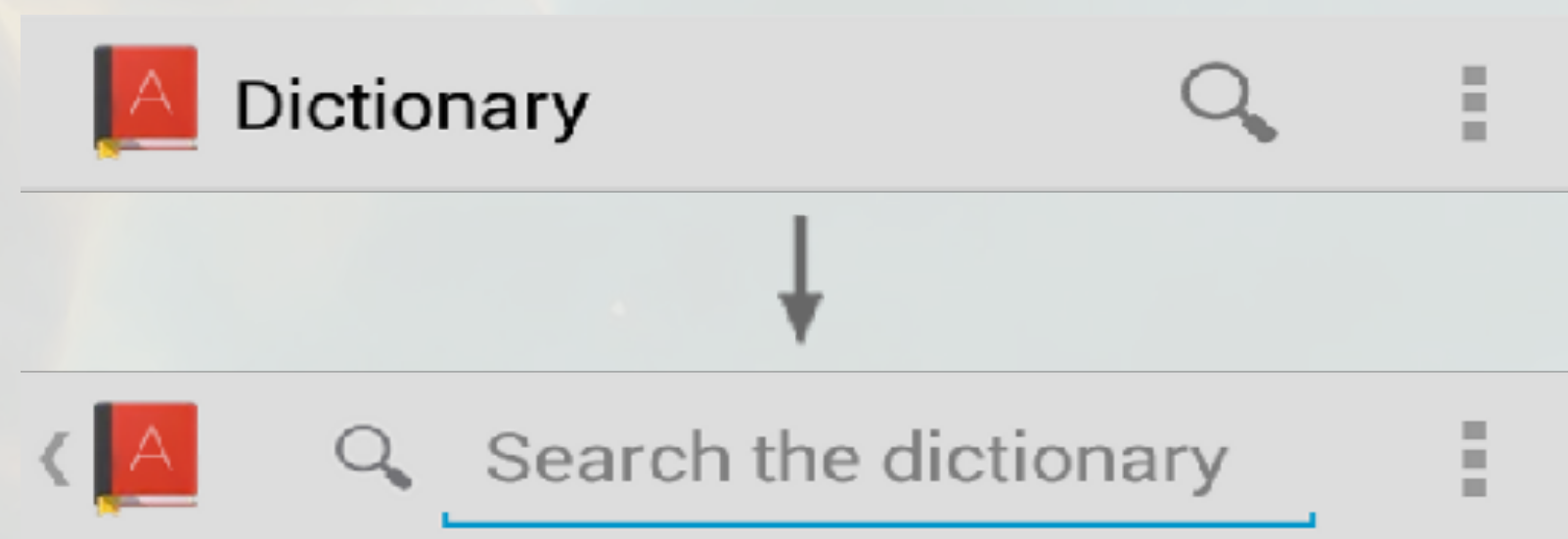
Get clicks on items

```
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here.
    int id = item.getItemId();
    if (id == R.id.action_settings)
        return true;
    return super.onOptionsItemSelected(item);
}
```

Sophisticated Buttons

ActionsViews

- 🔊 View is changing when we click the button



- 🔊 Use a SearchView

```
<item android:id="@+id/action_search"  
    android:title="@string/action_search"  
    android:icon="@drawable/ic_action_search"  
    app:showAsAction="ifRoom|collapseActionView"  
    app:actionViewClass="android.support.v7.widget.SearchView"  
>
```

Move a button, if necessary, i.e. if a button requires space

How to manipulate an ActionView

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu_main, menu);
    SearchView searchView =
        (SearchView)menu.findItem(R.id.action_search).getActionView();
    searchView.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            // Click when view is expanded ...
        }
    });
    searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
        @Override
        public boolean onQueryTextSubmit(String s) {
            // Submit Text ...
            return false;
        }
        @Override
        public boolean onQueryTextChange(String s) {
            // Detect that the text has changed ..
            return false;
        }
    });
    return true;
}
```

Even more complex buttons

221



Action Providers

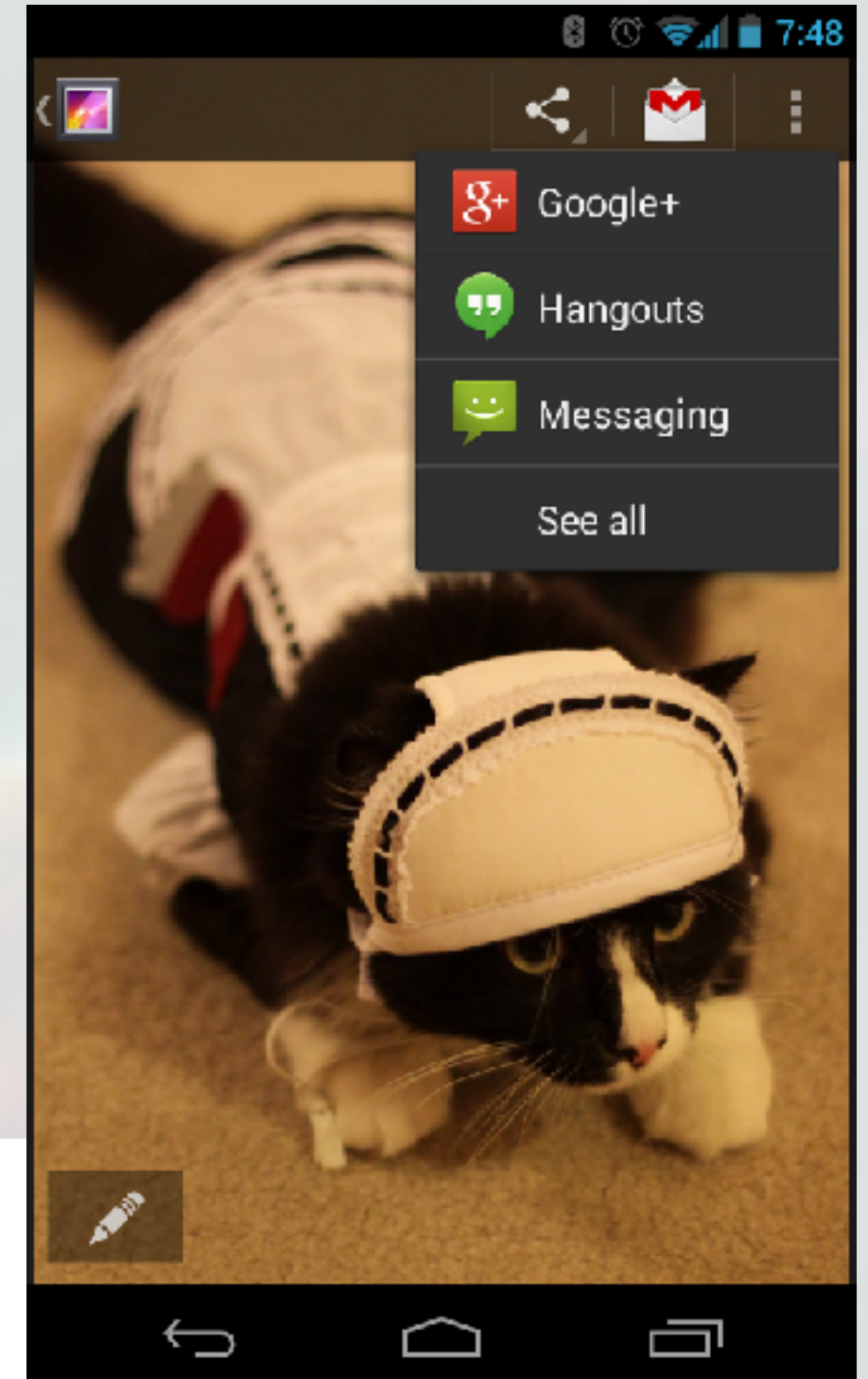
- Close to ActionViews
- Can have their own layout
- Overlap all current actions
- Trigger other Activities



Add the new menu

```
<item
```

```
    android:id="@+id/action_share"  
    android:title="Share"  
    app:showAsAction="ifRoom|collapseActionView"  
    app:actionProviderClass=  
        "android.support.v7.widget.ShareActionProvider" />
```



Manipulate ActionProvider

222



Build the intent you want to launch

```
private Intent getDefaultShareIntent() {
    Intent intent = new Intent(Intent.ACTION_SEND);
    intent.setType("text/plain");
    intent.putExtra(Intent.EXTRA_SUBJECT, "SUBJECT");
    intent.putExtra(Intent.EXTRA_TEXT, "Extra Text");
    return intent;
}
```



Launch it in the onCreateOptionsMenu



```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    MenuItem item = menu.findItem(R.id.menu_share);
    provider = (ShareActionProvider)
        MenuItemCompat.getActionProvider(item);
    provider.setShareIntent(getDefaultShareIntent());
    return super.onCreateOptionsMenu(menu);
}
```

Summary

223





ActionBarActivity helps to customize your Application

-  Add menus
-  Add ContentProvider



Integration of Fragments is possible

-  Helps to create dynamic views
-  Add scrolling effect or reusable components



Building its own ActionProvider

-  Only specify the onCreateActionView method



Before API 21, stack bar action helps to build Bar with tabs



Building Tabbed Applications

Renault@lrde.epita.fr



TabHost Component



Build a view with tabs is interesting

- Offers more features to the users
- without overloading the GUI

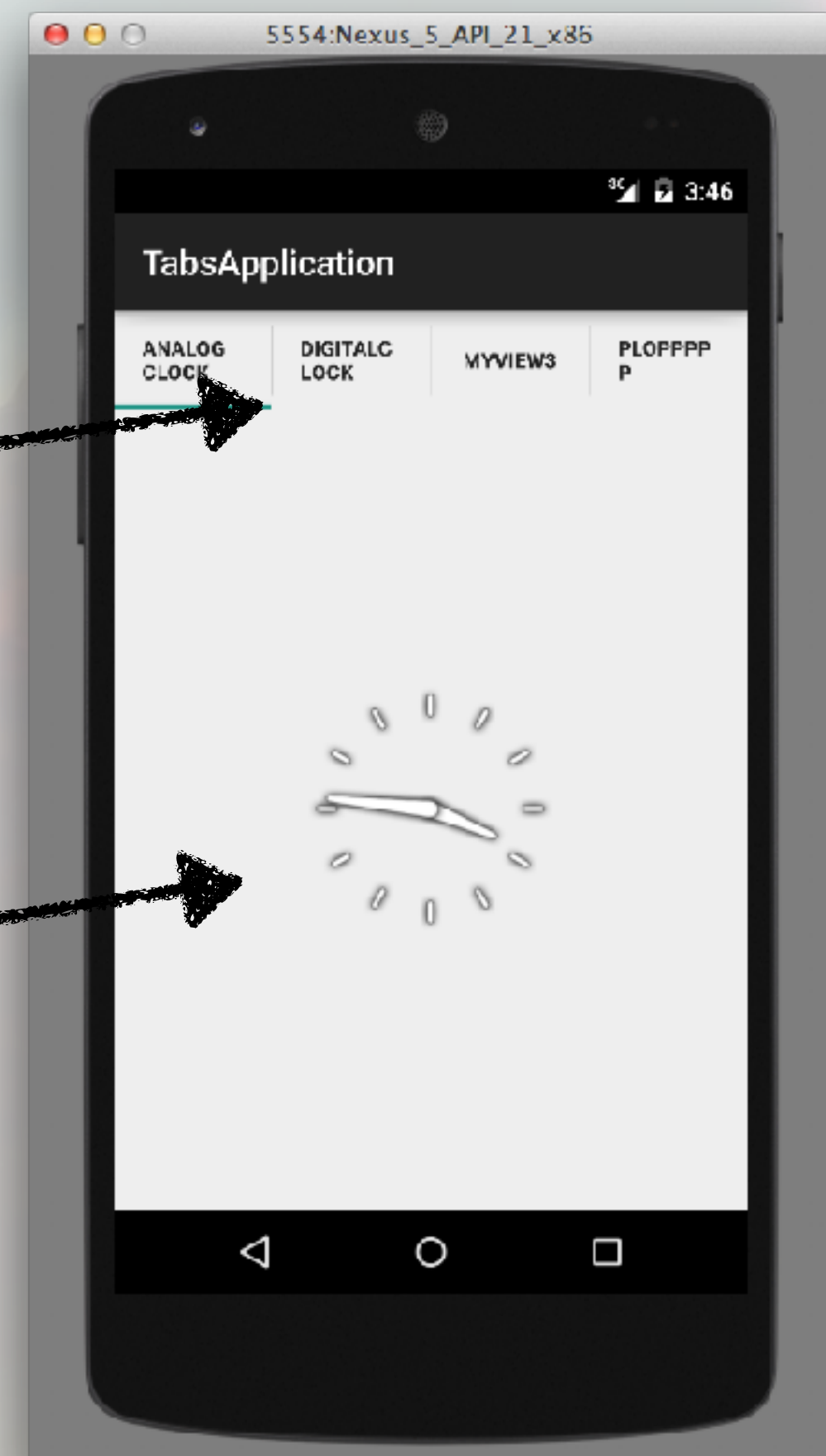


Two main elements

- A set of clickable tabs
- A frame layout used to display contents



These elements are manipulated transparently inside of the component



Define GUI (1/2)



Two GUI elements

- TabHost: the main component
- TabWidget : Displays a list of tab labels representing each page in the parent's tab collection (only for tweaking things)

<TabHost

```
xmlns:android="http://schemas.android.com/apk/res/android"  
android:id="@+id/tabhost"  
android:layout_width="fill_parent"  
android:layout_height="fill_parent">
```

<LinearLayout

```
android:orientation="vertical"  
android:layout_width="fill_parent"  
android:layout_height="fill_parent">
```

```
<TabWidget android:id="@android:id/tabs"  
    android:layout_width="fill_parent"  
    android:layout_height="wrap_content"/>
```

Define GUI (2/2)



The main part of the GUI

```
<FrameLayout android:id="@android:id/tabcontent"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">

    <AnalogClock android:id="@+id/tab1"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent" />

    <TextView android:id="@+id/tab3"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:text="@string/nothing" />

</FrameLayout>

</LinearLayout>
</TabHost>
```

Setup the TabHost

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    TabHost tabs=(TabHost)findViewById(R.id.tabhost);

    tabs.setup();

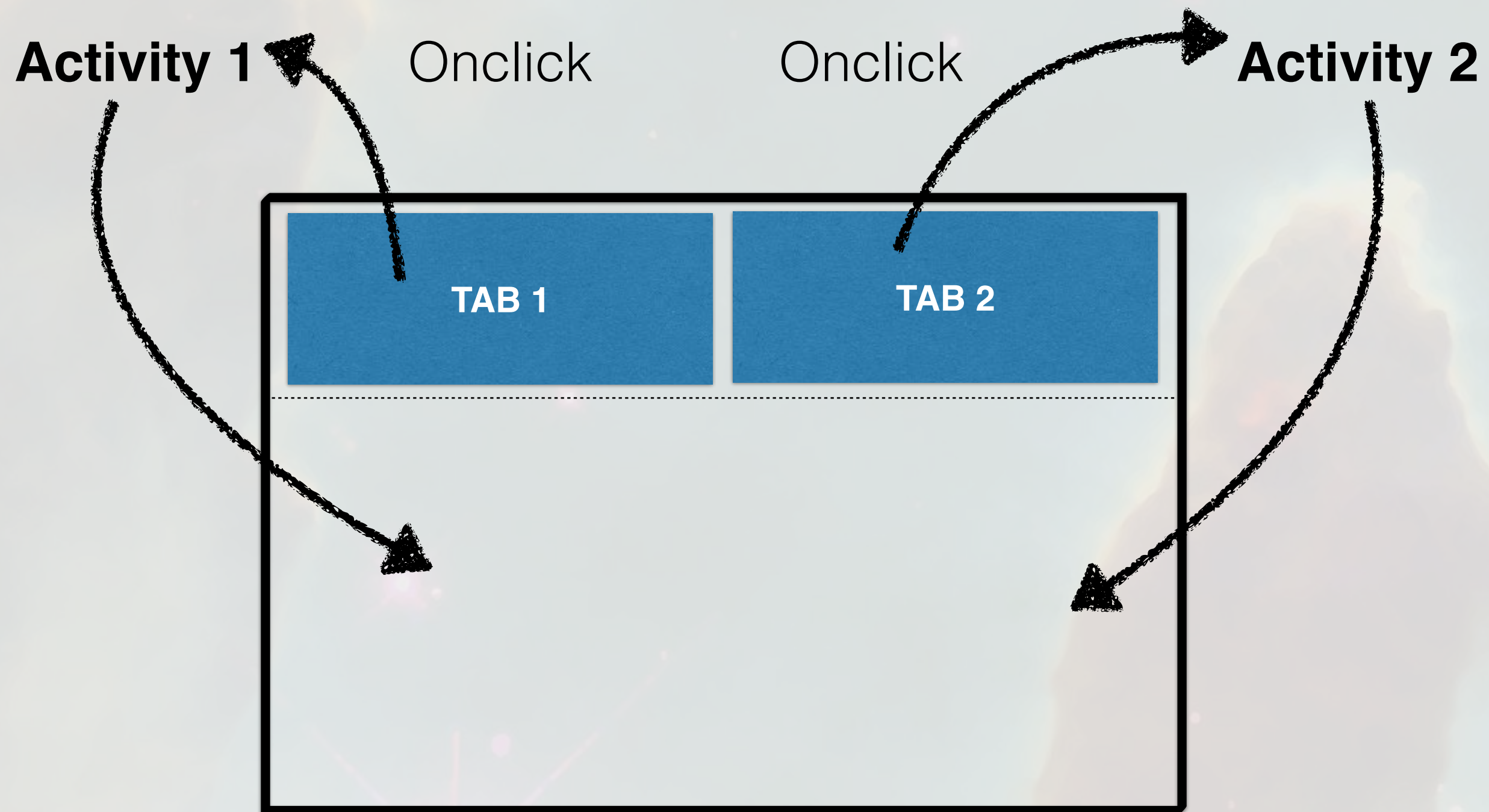
    TabHost.TabSpec spec=tabs.newTabSpec("tag1");
    spec.setContent(R.id.tab1);
    spec.setIndicator("Analog Clock");
    tabs.addTab(spec);

    spec=tabs.newTabSpec("tag2");
    spec.setContent(R.id.tab3);
    spec.setIndicator("MyTextView");
    tabs.addTab(spec);
}
```

Building Dynamic Tabs



TabHost and Intents can be mixed!



Manipulating TabSpec

Trigger an intent onClick

```
TabHost.TabSpec spec = tabs.newTabSpec("tag2");  
spec.setContent(new Intent()  
    .setClass(MainActivity.this, Mydedicatedclass.class));  
spec.setContent(R.id.tab3);  
spec.setIndicator("MyTextView");  
tabs.addTab(spec);
```

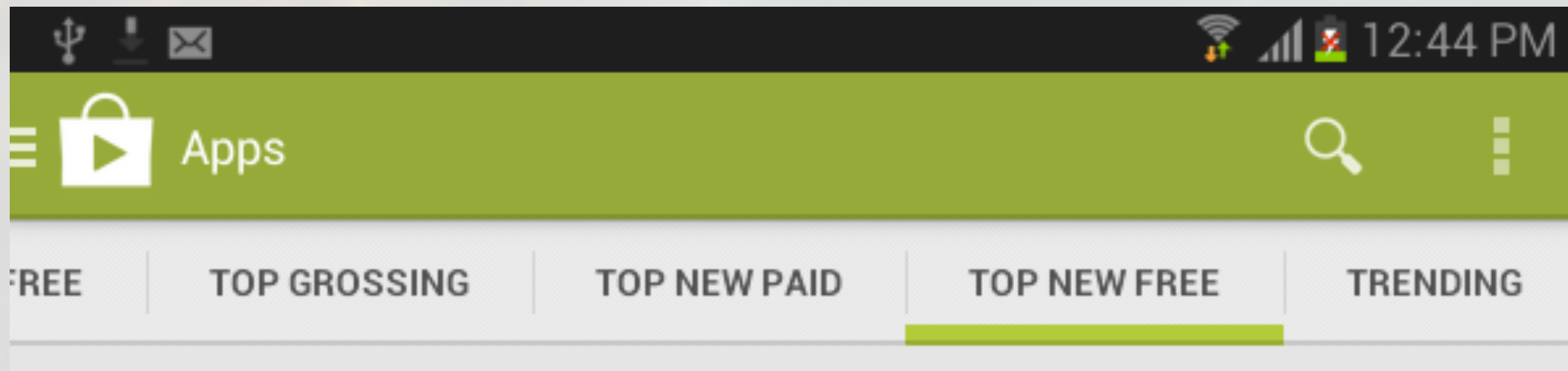
Replace tabs by some view

```
TextView tv = new TextView(this);  
tv.setText("Tab 1");  
tv.setBackgroundColor(Color.RED);  
tv.setIndicator(txt);
```


Scrolling Tabs



Sometime, applications provide horizontally (or vertically scrolling tabs)



Simply encapsulate TabWidget in a scroll layout

```
<HorizontalScrollView android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:fillViewport="true"
    android:scrollbars="none">
    <TabWidget android:id="@android:id/tabs"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content" />
</HorizontalScrollView>
```

Summary



TabHost makes applications more user-friendly

- We can have as many tabs we want
- We can build scrollable applications
- Configurations through
 - ▶ **Dedicated views (prefer this one)**
 - ▶ **Dedicated activity**



Compatibility with fragments

- Component TabHostFragment helps to embed fragments
- Same workflow
- More flexible
 - ▶ **Tab bar can be positioned at the bottom part of the screen**
 - ▶ **Guidelines don't recommend that**



ViewPager

Renault@lrde.epita.fr



View Pager



An horizontal swipe change the main view



Used for slideshows, or UI setup wizards



Also used for homescreen



Alternative to tabbed application when only few tabs are required



Based on fragments



Each view is a dedicated fragment



The parent view will display fragments and animations



You don't need to define your own animations

Define the Fragment (1/2)



Define the class that inherits from Fragment

```
<ScrollView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/content"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

<TextView style="?android:textAppearanceMedium"
    android:padding="16dp"
    android:lineSpacingMultiplier="1.2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/lorem_ipsum" />
</ScrollView>
```

Define the Fragment (2/2)



Define the core of your fragment

```
public class ScreenSlidePageFragment extends Fragment {  
  
    @Override  
    public View onCreateView(LayoutInflater inflater,  
                             ViewGroup container,  
                             Bundle savedInstanceState) {  
        ViewGroup rootView = (ViewGroup) inflater.  
            inflate(R.layout.some_page, container, false);  
        return rootView;  
    }  
}
```


Setup the main activity

241



Use a ViewPager to display your pages

```
<android.support.v4.view.ViewPager
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/pager"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />
```



You now have to :

- 📱 Create a pager adapter extending FragmentStatePagerAdapter
- 📱 setup the view pager with your own adapter

Defining a PagerAdapter



A simple adapter with 5 screens

Using different fragment can be done in getItem

```
private class ScreenSlidePagerAdapter
    extends FragmentStatePagerAdapter {

    public ScreenSlidePagerAdapter(FragmentManager fm) {
        super(fm);
    }

    @Override
    public Fragment getItem(int position) {
        return new ScreenSlidePageFragment();
    }

    @Override
    public int getCount() {
        return 5;
    }
}
```

Setup altogether

```
public class ScreenSlidePagerActivity extends FragmentActivity
{
    private ViewPager mPager;
    private PagerAdapter mPagerAdapter;

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_screen_slide);

        mPager = (ViewPager) findViewById(R.id.pager);
        mPagerAdapter = new ScreenSlidePagerAdapter
            (getSupportFragmentManager());
        mPager.setAdapter(mPagerAdapter);
    }

    public void onBackPressed() {
        if (mPager.getCurrentItem() == 0)
            super.onBackPressed();
        else
            mPager.setCurrentItem(mPager.getCurrentItem() - 1);
    }
}
```

Animating transitions



Animations can be specified by implementing ViewPager.PageTransformer

- This method is called once for each visible page
- The previous page is also considered as visible for the animation
 - ▶ **Moving from page 2 to 3 will imply page 1, page 2 and page**
- transformPage(): is doing the job



Some predefined animations exits

- The page grows while getting close to the center



```
ViewPager mPager = (ViewPager) findViewById(R.id.pager);  
mPager.setPageTransformer(true, new ZoomOutPageTransformer());
```

Building your transitions



Implementation of ViewPager.PageTransformer is easy!

```
public class ParallaxPageTransformer
    implements ViewPager.PageTransformer {

    public void transformPage(View view, float position) {
        int pageWidth = view.getWidth();
        if (position < -1) { // [-Infinity,-1)
            // This page is way off-screen to the left.
            view.setAlpha(1);
        } else if (position <= 1) { // [-1,1]
            //Half the normal speed
            view.setTranslationX(-position * (pageWidth / 2));
        } else { // (1,+Infinity]
            // This page is way off-screen to the right.
            view.setAlpha(1);
        }
    }
}
```

Summary



ViewPager

- reduce the amount of displayed elements
- without removing actions



Easily customizable

- Specification of animations is simple
 - ▶ **use predefined ones**
 - ▶ **use your own**



Alternative to tabbed applications

- useful for content displaying
 - ▶ **Video**
 - ▶ **News**
 - ▶ **...**



Android Web App

Renault@lrde.epita.fr



Why developing a WebApp

250

If you already have a website

-  Easy to develop of Proof-of-Concept for your application

If you want to display third party contents

-  It may be useful to connect with webviews

If you want to use javascript

-  For using existing framework



If you want a quick display of some information

-  Displays array for instance

If you want to offer the ability to surf on web while staying in your application

How to use WebViews

A webpage can be display

-  on an explorer
-  on an application



WebViews



Require an access to the internet

- 🔊 Declare it in the AndroidManifest.xml

```
<manifest ... >
  <uses-permission android:name="android.permission.INTERNET" />
  ...
</manifest>
```



Load an URL

```
webView = (WebView) findViewById(R.id.webView);
webView.setWebViewClient(new WebViewClient());
webView.loadUrl("http://google.com");
```

Loading Static HTML

Load a static HTML

```
String summary = "<html><body>You scored <b>192</b> points.</body></html>";  
webView.loadData(summary, "text/html", null);
```

You can also write into a file a load directly this file

```
WebView webView = (WebView) findViewById(R.id.webView);  
File file = new File(Environment.getExternalStorageDirectory()  
    + "<FOLDER_PATH_TO_FILE>/<FILE_NAME>");  
webView.loadUrl("file://" + file.getAbsolutePath());
```

OR

```
webView.loadUrl("file:///android_asset/filename.html");
```

Using Javascript (1/2)



Allows the web view to use Javascript

 Warning: javascript may be memory/energy consumer

```
WebSettings webSettings = myWebView.getSettings();  
webSettings.setJavaScriptEnabled(true);
```



Connect Javascript and Java

 Annotate your API with @JavascriptInterface

```
public class WebAppInterface {  
    Context mContext;  
    WebAppInterface(Context c) {mContext = c; }  
  
    @JavascriptInterface  
    public void showToast(String toast) {  
        Toast.makeText(mContext, toast, Toast.LENGTH_SHORT).show();  
    }  
}
```

Using Javascript (2/2)



Register the interface to the web view

- First parameter: the Java class
- Second parameter: the name that can be used from HTML webpages

```
webView.addJavascriptInterface(  
    new WebAppInterface(this),  
    "Android"  
);
```



Build an HTML using this interface

```
<input type="button" value="Say hello »  
    onClick="showAndroidToast('Hello Android!')" />  
<script type="text/javascript">  
    function showAndroidToast(toast) {  
        Android.showToast(toast);  
    }  
</script>
```

Restrictions (1/2)



The app must restrict accessibles URL

- 🔊 Capture outgoing links
 - ▶ So that the user cannot escape your application
- 🔊 Define your own WebViewClient

```
private class MyWebViewClient extends WebViewClient {
    @Override
    public boolean shouldOverrideUrlLoading(WebView view,
                                             String url) {
        if (Uri.parse(url).getHost().equals("www.example.com")) {
            return false;
        }
        Intent intent =
            new Intent(Intent.ACTION_VIEW, Uri.parse(url));
        startActivity(intent);
        return true;
    }
}
```


Restrictions (2/2)

Pass then this client to the web view

```
myWebView.setWebViewClient(new MyWebViewClient());
```

History management

 canGoBack; goBack; canGoForward; goForward

Capture keydown for implementing back

```
public boolean onKeyDown(int keyCode, KeyEvent event) {  
    if ((keyCode == KeyEvent.KEYCODE_BACK) &&  
        myWebView.canGoBack()) {  
        myWebView.goBack();  
        return true;  
    }  
    return super.onKeyDown(keyCode, event);  
}
```

Summary



Building applications based on web contents

- is easy
- helps to offer better user experience than if we only use an explorer
 - ▶ **Use of javascript**
 - ▶ **Use of CSS**
- Still compatible with Android mechanisms



How to debug such can application

- Use WebChromeClient
- Use onConsoleMessage
- Logcat will then display your logs
- OR your can define your own API



Permissions

Renault@lrde.epita.fr



Permissions


 **Before installing an App, all the required permissions are displayed to the user**

 The user may accept and download

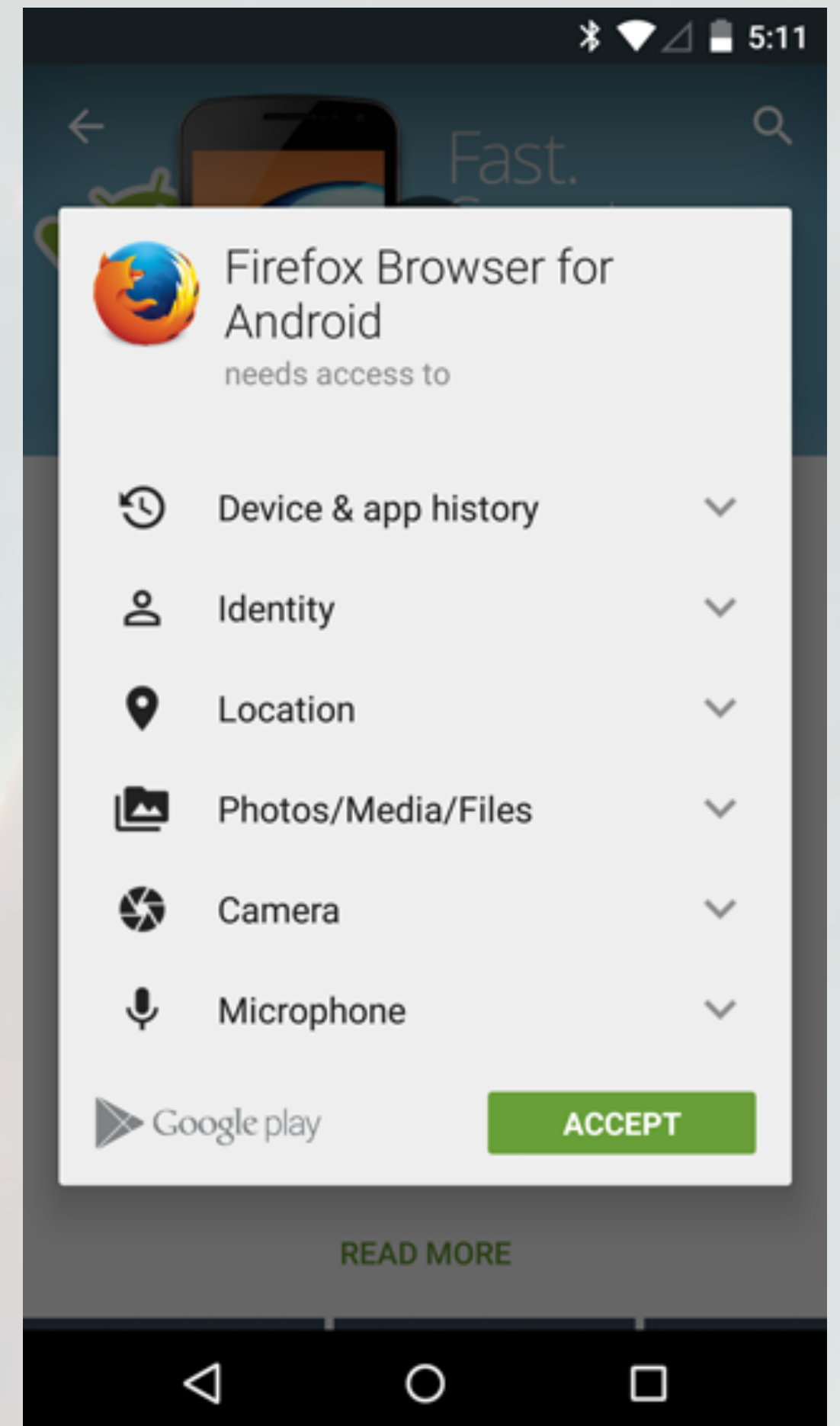
 ...or not!

 Theses permissions are describe in the Android Manifest.xml

 **At runtime**

 The user may choose to authorize or not some permissions

▶ **but the app is already installed and may have already explored all your personal data.**



AndroidManifest.xml



All permissions must be declared in AndroidManifest

ACCESS_COARSE_LOCATION

Approximative localisation

ACCESS_FINE_LOCATION

Precise localisation

BATTERY_STATS

Collect information on battery

BLUETOOTH

Connect to already paired bluetooth

BLUETOOTH_ADMIN

Discover bluetooth devices

CALL_PHONE

Call without displaying keyboard

CALL_PRIVILEGE

Call anybody including Emergency

CAMERA

Camera access

CAPTURE_AUDIO_OUTPUT

Audio capture

BROADCAST_SMS

Receive

cont'd

CAPTURE_VIDEO_OUTPUT

Capture video

INTERNET

Require access to the Internet

READ_CONTACT

Read all your contact

READ_CALENDAR

Read your calendar

....

...

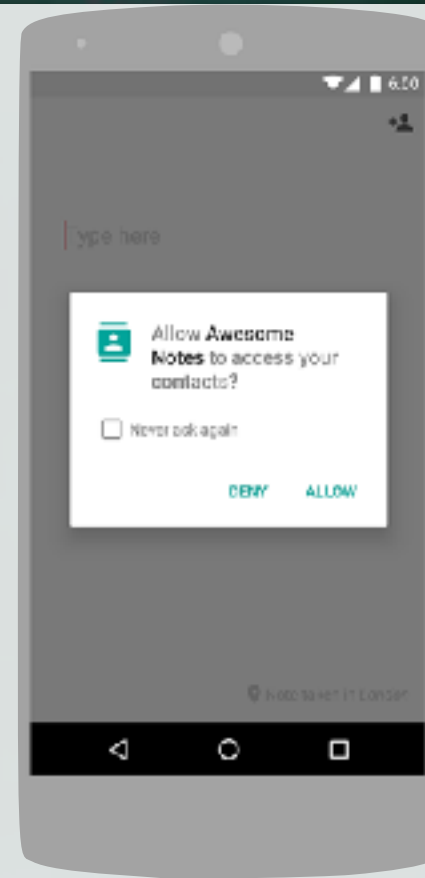
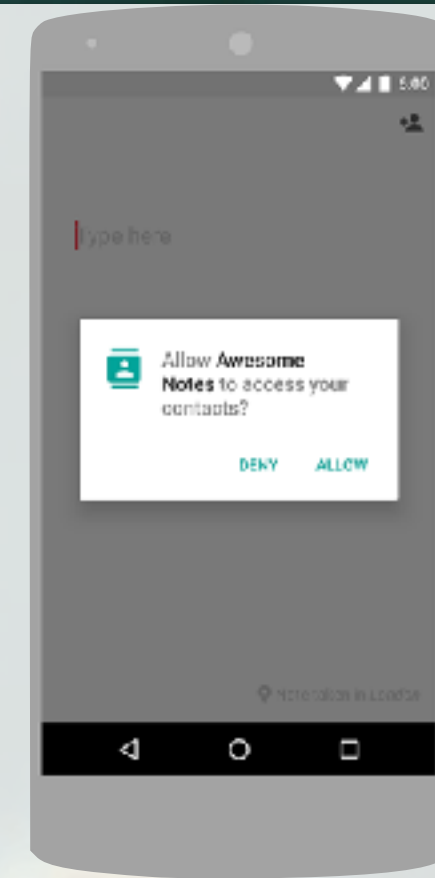
A Lot more!

System Permissions



Only available since Android API 23

- Goal: have more secure applications
- Ask before first usage for some permission



Two steps process

- (1) declare permissions into your AndroidManifest.xml
- (2) check before using something that you have permissions
 - ▶ If yes: use the feature
 - ▶ otherwise you have to request permission



Permissions must be checked frequently since the user may changed them at anytime

How to request for permissions?

267



Define the permissions you are interested in

an define an ID for this set of permission (here permsCode)

```
String[] perms = {  
    Manifest.permission.ACCESS_FINE_LOCATION,  
    Manifest.permission.ACCESS_COARSE_LOCATION,  
    Manifest.permission.INTERNET  
};  
final int permsCode = 200;
```



Request then for this permission

Don't forget to check the API level

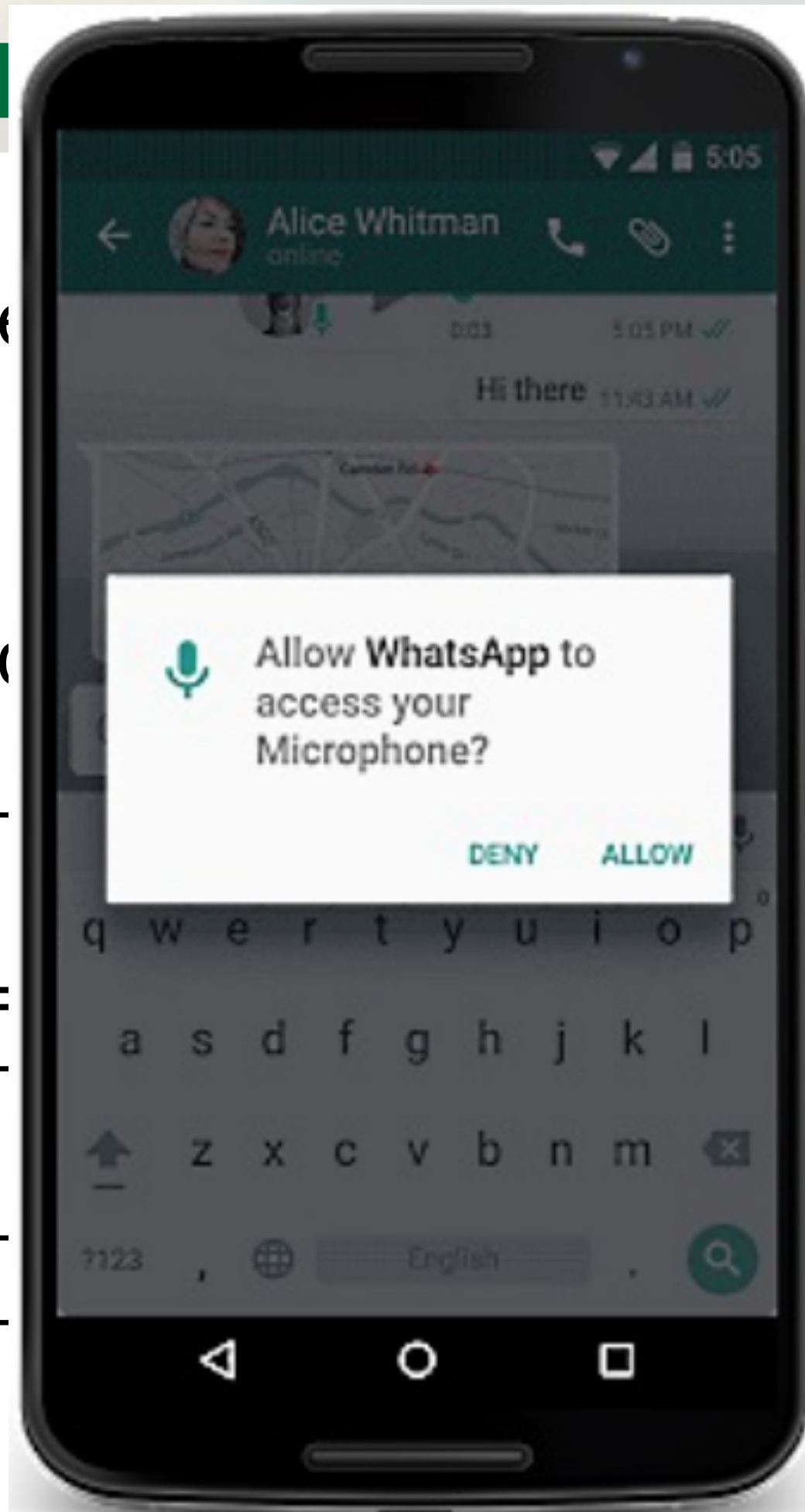
```
// Check For permissions  
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {  
    requestPermissions(perms, permsCode);  
}
```

Permissions authorized callback



The onRequestPermissionsResult callback will be run as a response to req

```
public void  
onRequestPermissionsResult(  
  
    if (permsRequestCode ==  
        boolean fine =  
            grantResults[0] ==  
  
        boolean coarse =  
            grantResults[1] ==  
  
        boolean internet =  
            grantResults[2] ==  
            // Do something  
    }  
}
```



```
requestCode,  
permissions,  
results){
```

```
per.PERMISSION_GRANTED;
```

```
per.PERMISSION_GRANTED;
```

```
per.PERMISSION_GRANTED;
```

Summary

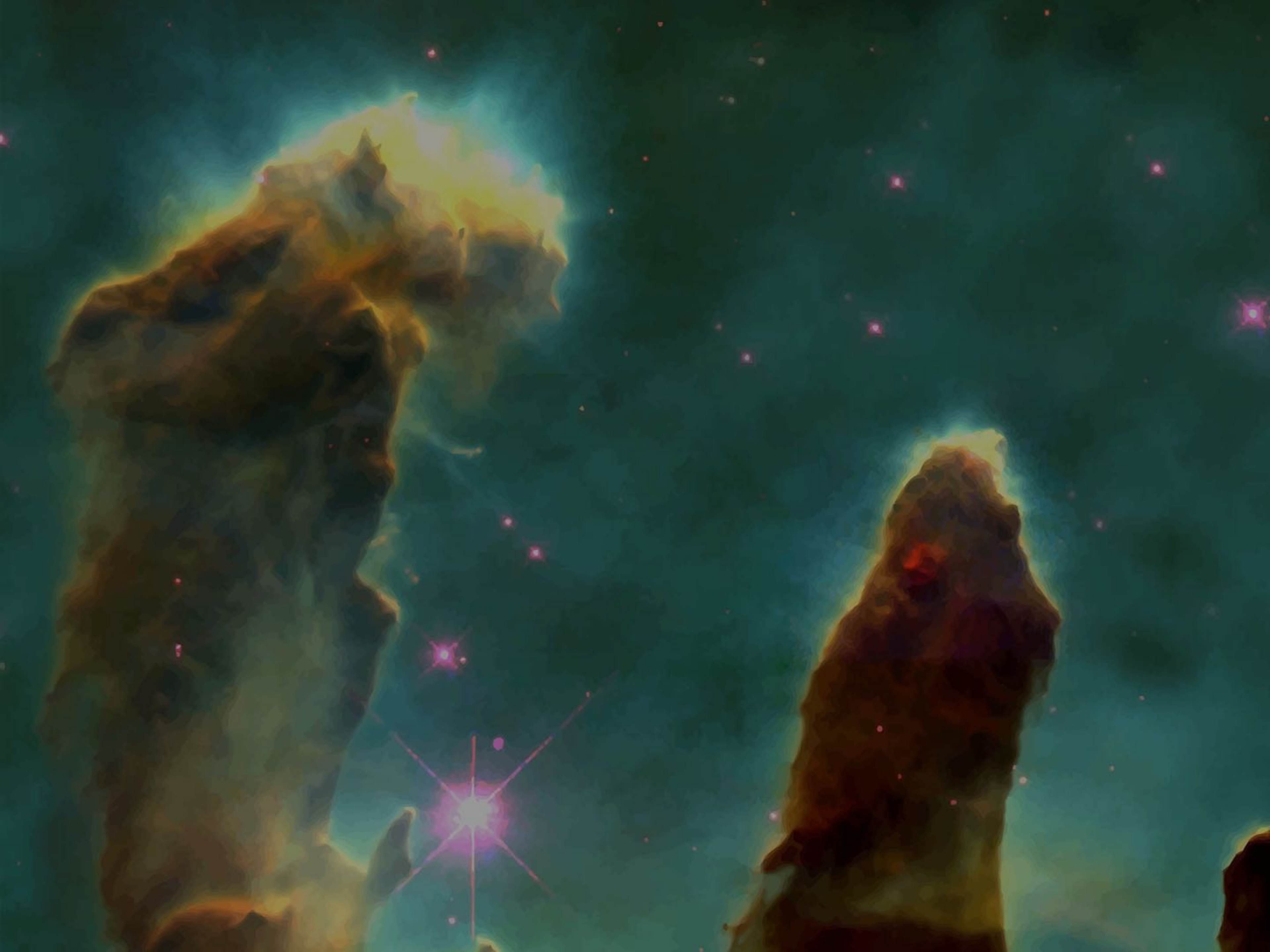
Check permission at runtime

```
if (ActivityCompat.checkSelfPermission(getActivity(),  
    Manifest.permission.ACCESS_FINE_LOCATION) !=  
    PackageManager.PERMISSION_GRANTED) {  
    ...  
}
```

AndroidManifest covers all permissions

More security is available since Android M

Think to check the permission frequently



Maps and Localisation

Renault@lrde.epita.fr



GoogleMap

273

Not a part of Android

-  It's a project from Google
-  <https://console.developers.google.com>

Add your key in res/values/google_map_api.xml

```
<resources>
    <string name="google_maps_key_instructions"
        templateMergeStrategy="replace">
    <string name="google_maps_key"
        templateMergeStrategy="preserve">
        MY_KEY
    </string>
</resources>
```

Dependencies




274



Android Studio provide ready-to-use project



Install dependencies

-  Tools/Android/SDK Manager
-  Install Google Play Services
-  Install Google Play Repository



Modify Graddle.build

```
compile 'com.google.android.gms:play-services:8.4.0'
```



You can now display a map

GoogleMap: details



Composed of the SupportMapFragment

- 📍 A fragment that can display a map
- 📍 Acquiring the map should be done through

```
getMapAsync ( OnMapReadyCallback ) ;
```

- 📍 Implement then the callback



Can be integrated directly

```
<fragment  
  xmlns:android="http://schemas.android.com/apk/res/android"  
  android:name=  
    "com.google.android.gms.maps.SupportMapFragment"  
  android:id="@+id/map"  
  android:layout_width="match_parent"  
  android:layout_height="match_parent"/>
```

Manipulating Markers



Add a marker

```
private GoogleMap mMap;  
  
// ...  
Marker m = mMap.addMarker(new MarkerOptions()  
    .position(point)  
    .icon(BitmapDescriptorFactory  
        .defaultMarker(BitmapDescriptorFactory.HUE_ORANGE))  
    .draggable(true)  
    .alpha(0.7f)  
    .visible(true)  
    .title("Fancy title"));  
    .snippet("Snippet for this marker"));
```



Remove a marker

```
m.remove();
```

Grab User Events

277



Click on the map

 `setOnMapClickListener`



Long click on the map

 `setOnMapLongClickListener`



Click on a marker

 `setOnMarkerClickListener`

 You must keep track of existing markers



Moving a marker

 `setOnMarkerDragListern`

**Many other actions
can be captured!**

Localisation (Where Am I?)

Exact localisation is complex

- Multiple sources:
 - ▶ **GPS, Wifi, Cellular network**
- Various kind of precision
- The user is moving
- Energy saver with best precision

The 3 forms are based on LocationManager
...but Google Play API helps to build efficient applications

Many ways to get the User Localisation

- LocationManager framework
- GooglePlay location API
- Use existing features

```
mMap.setMyLocationEnabled(true);
```

Using Localisation

Modify AndroidManifest.xml

```
<uses-permission  
    android:name="android.permission.ACCESS_COARSE_LOCATION" />  
  
<uses-permission  
    android:name="android.permission.ACCESS_FINE_LOCATION" />
```

Two Kind of usages:

 ACCESS_COARSE_LOCATION:

- ▶ **approximative localisation**
- ▶ **Give you the current block**

 ACCESS_FINE_LOCATION

- ▶ **precise location**
- ▶ **position in the street**

Use Google Localisation API



Based on the latest known position

- 📱 enough in most cases
- 📱 can specify brand with / energy policy
- 📱 Specify Connection Failed
- 📱 Intensive use of Callbacks



Must be created and instantiated in onCreate method

```
GoogleApiClient mGoogleApiClient;  
protected synchronized void buildGoogleApiClient() {  
    mGoogleApiClient = new GoogleApiClient.Builder(this)  
        .addConnectionCallbacks(this)  
        .addOnConnectionFailedListener(this)  
        .addApi(LocationServices.API)  
        .build();  
}
```


Following User's Moves (1/2)



Build requests

```
LocationRequest mLocationRequest;
protected void createLocationRequest() {
    mLocationRequest = new LocationRequest();
    mLocationRequest.setInterval(1000);
    mLocationRequest.setFastestInterval(1000);
    mLocationRequest.
        setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);
}
```



Specify priority

PRIORITY_BALANCED_POWER_ACCURACY

Performances and Precisions
Wifi + réseau cellulaire

PRIORITY_HIGH_ACCURACY

Extreme Localisation
GPS

PRIORITY_LOW_POWER

City precision
energy saving

PRIORITY_NO_POWER

No Consomation
Localisation deduced from the other activities

Following User's Moves (2/2)



Then start requesting the position

```
protected void startLocationUpdates() {  
    createLocationRequest();  
    LocationServices.FusedLocationApi.requestLocationUpdates(  
        mGoogleApiClient, mLocationRequest, this);  
}
```



The callback onLocationChanged will then be called

You have to do something in this callback

Using Location Manager (1/2)

```
// Acquire a reference to the system Location Manager
LocationManager locationManager = (LocationManager)
this.getSystemService(Context.LOCATION_SERVICE);

// Define a listener that responds to location updates
LocationListener locationManager = new LocationListener() {
    public void onLocationChanged(Location location) {
        // Called when a new location is found by the
        // network location provider.
        makeUseOfNewLocation(location);
    }

    public void onStatusChanged(String provider,
                               int status, Bundle extras) {}

    public void onProviderEnabled(String provider) {}

    public void onProviderDisabled(String provider) {}
};
```

Using Location Manager (2/2)



Build requests

```
// Register the listener with the Location Manager
// to receive location updates
locationManager.requestLocationUpdates
    (LocationManager.NETWORK_PROVIDER, 0, 0,
     locationManagerListener);
```



You must then



- Define the source : GPS, WIFI, ...
- Define the frequency
- Define refreshing
- ...

Summary





Manipulation of maps is easy

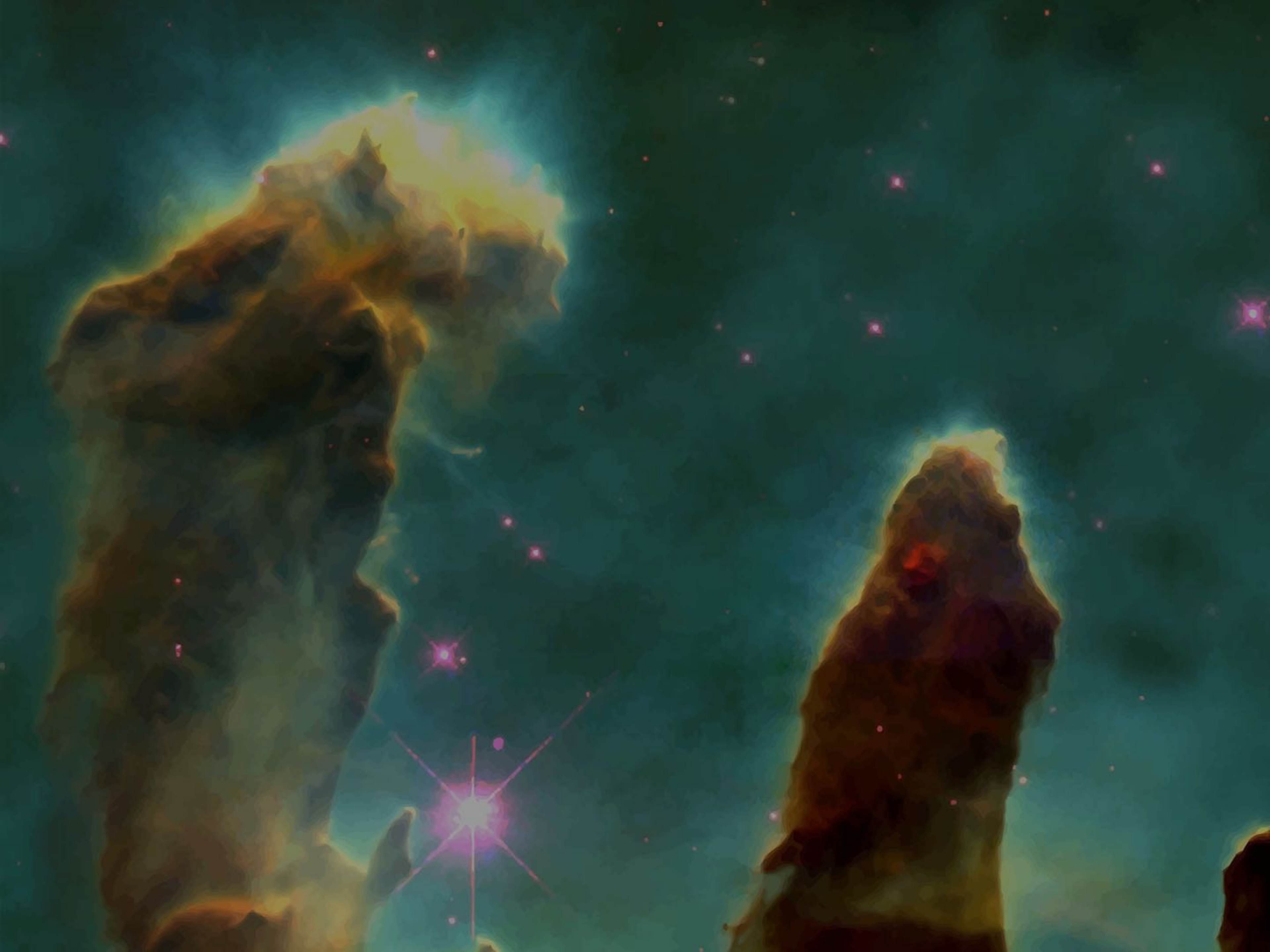
-  When you use Google API

In the emulator you can virtually move

-  telnet 127.0.0.1 5554
-  geo fix 12 40

LocationManager framework

-  Is complex to use
-  Does not allow to specify energy policies
-  BUT ...
-  ... independent from Google !



Wearables

Renault@lrde.epita.fr





Wearable

- Current trends for Application
- provide quick access to your app
- Easily customizable
- Objectives: ensure that the user will not miss any notifications from its favorite app



Many kind of wearable exists

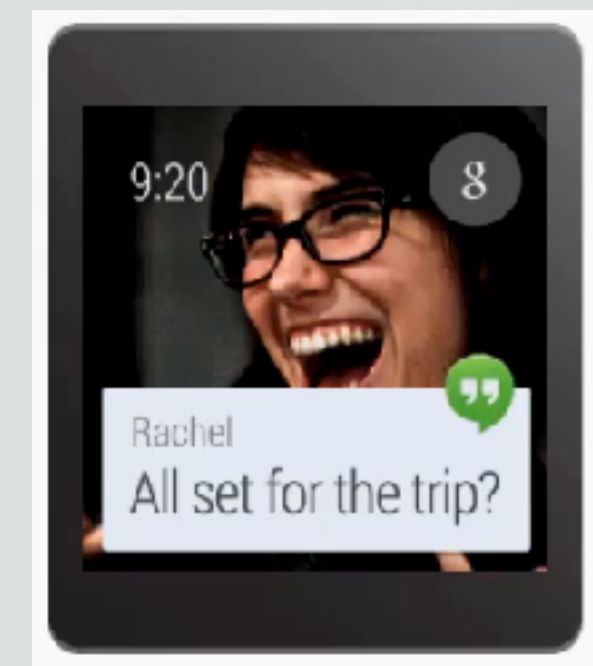
- Activity trackers
- Watches
- Glasses
- ...

ContextStream and CueCard



ContextStream

- Smarter Notifications than in the phone
- Notifications are displayed vertically
- On a card, a right swipe displays more informations



CueCard

- open when saying "ok google" or when tapping the home screen
- Is triggered through voice intents



How to build a wearable activity?

291



Similar to a classical application



Things to know about:

After some time the operating system is sleeping

- ▶ when the user will go back, its application will no longer be the active one
- ▶ Home screen will be displayed

Small screen size !

- ▶ **Not easy to manipulate a complex UI**

Some frameworks are not supported

Apps are not downloaded on the device

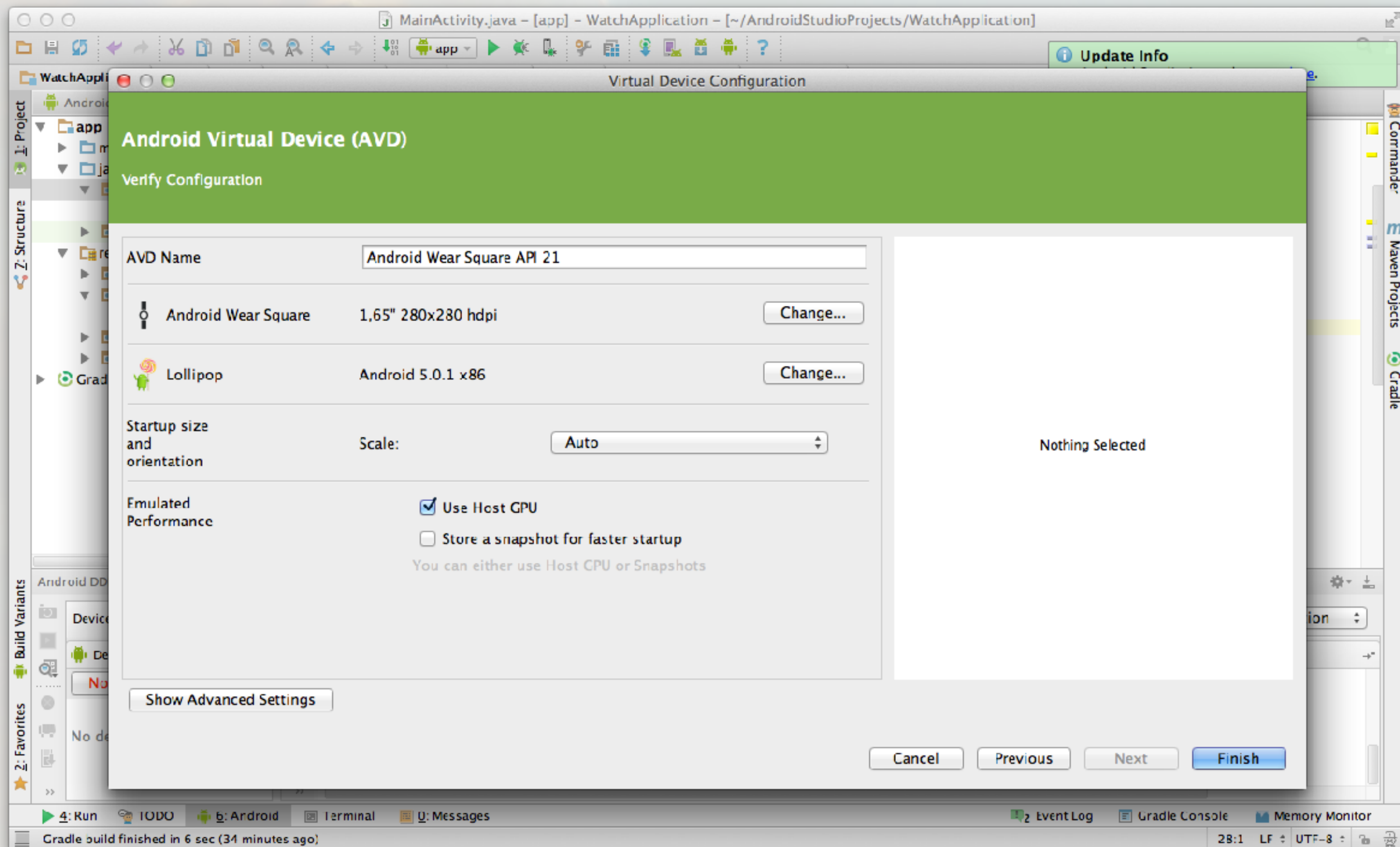
- ▶ The mother app is running on the phone
- ▶ The wearable app is displaying on the wearable

Building a test device



Building a virtual wearable is easy

- AVD manager provides a lot of predefined skills

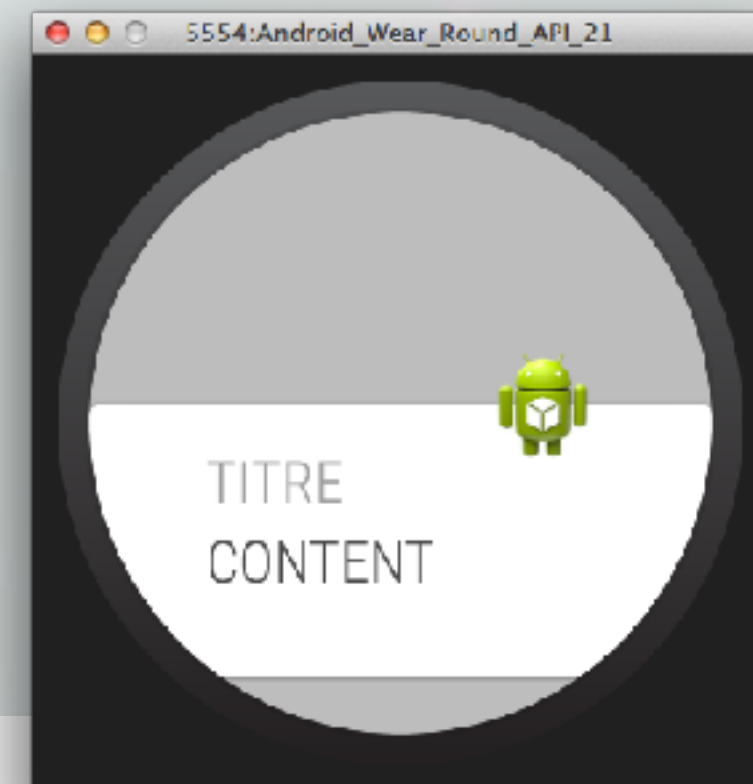


Building a notification



Notifications are displayed after a top-down move

- An action can be defined when tapping a notification



```
Builder notificationBuilder = new Builder(this)
    .setSmallIcon(R.drawable.ic_launcher)
    .setContentTitle("TITRE")
    .setContentText("CONTENT");
```

```
// Get an instance of NotificationManager Service
NotificationManagerCompat notificationManager =
    NotificationManagerCompat.from(this);
```

```
// Build the notification and issues it with
//notification manager.
notificationManager.notify(notificationId,
notificationBuilder.build());
```

Using voice to launch Applications (1/2)



Fundamental mechanisms for wearable

 Running predefined apps:

▶ Take a note, Set an alarm

 Running third party apps:

▶ Reaction to the "Start" command

▶ The application is then launched





Running predefined apps

```
<activity android:name="MyNoteActivity">
  <intent-filter>
    <action android:name="android.intent.action.SEND" />
    <category
      android:name="com.google.android.voicesearch.SELF_NOTE" />
  </intent-filter>
</activity>
```

Using voice to launch Applications (2/2)



Running third party apps

-  Same mechanism
-  Only specify the name that must be speak

```
<application>
  <activity android:name="StartRunActivity"
            android:label="My Running App">
    <intent-filter>
      <action android:name="android.intent.action.MAIN" />
      <category
            android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
  </activity>
</application>
```

Voice interaction (1/2)

The app may interact with voice

-  Build an intent to capture user voice
-  Result is provided as a list of string

Start listening

```
private static final int SPEECH_REQUEST_CODE = 0;

// Create an intent that can start the Speech Recognizer
private void displaySpeechRecognizer() {
    Intent intent = new
    Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
    intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
        RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
    // Start the activity, the intent will be
    // populated with the speech text
    startActivityForResult(intent, SPEECH_REQUEST_CODE);
}
```


Voice interaction (2/2)



Getting results

```
// This callback is invoked when the Speech Recognizer
// returns. This is where you process the intent and extract
// the speech text from the intent.
@Override
protected void onActivityResult(int requestCode,
                                int resultCode,
                                Intent data) {
    if (requestCode == SPEECH_REQUEST_CODE
        && resultCode == RESULT_OK) {
        List<String> results = data.getStringArrayListExtra(
            RecognizerIntent.EXTRA_RESULTS);
        String spokenText = results.get(0);
        // Do something with spokenText
    }
    super.onActivityResult(requestCode, resultCode, data);
}
```

Summary



Developing a wearable activity

- is similar to developing a classical one
- Android studio also provides builtin tools
- can't be done without a phone-side application



Some specificities must be considered

- ContextStream

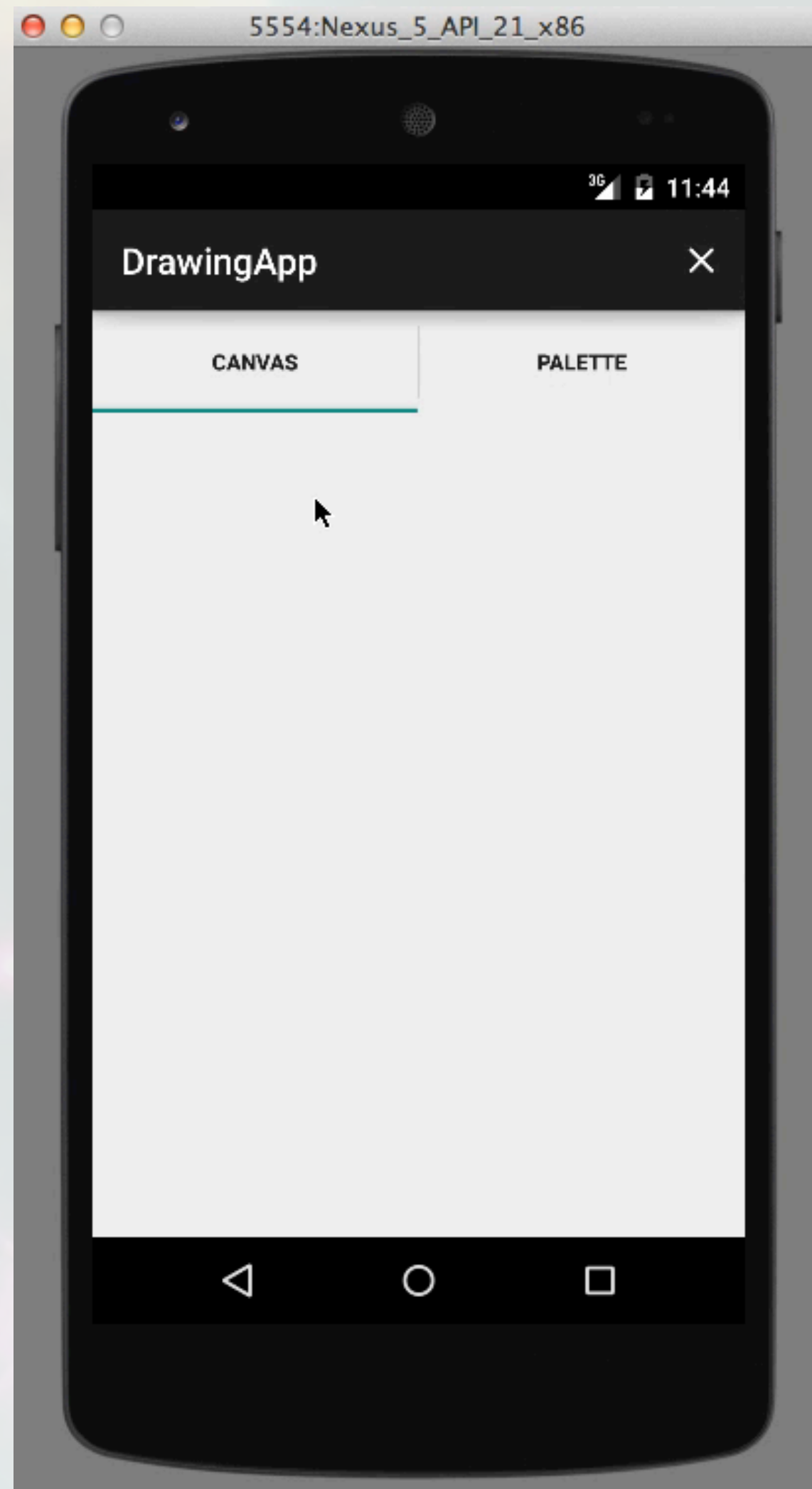


Exercise: Drawing Application

Renault@lrde.epita.fr



Demo Video



GUI and Implementation details

303



Multiples implementations for the layout

- Use button and fragments
 - ▶ **Two buttons canvas and palette**
 - ▶ **A click on a button update the main view with the DrawingCanvas or the List**
- OR Use ActionBar and ViewPager
- OR Use TabHost and TabWidgets



Define a DrawCanvas

- Look to classes Paint and Path



Grab User touches and clicks

Summary



Toward an end-to-end application

- 📱 Add persistency for rotation
- 📱 Add persistency even when the application is closed
- 📱 Add the ability to draw predefined shapes
- 📱 Add the ability to share the drawing with the ActionBar
 - ▶ **Useful for many applications**
 - **Bltstrip**
 - **Snapshot**
 - **doodle.ly**
 - ...
- 📱 Add the ability to create movies from a sequence of images
- 📱 Add the ability to explore and save previous drawing



This App teach you how to deal with view and how to move them through the UI

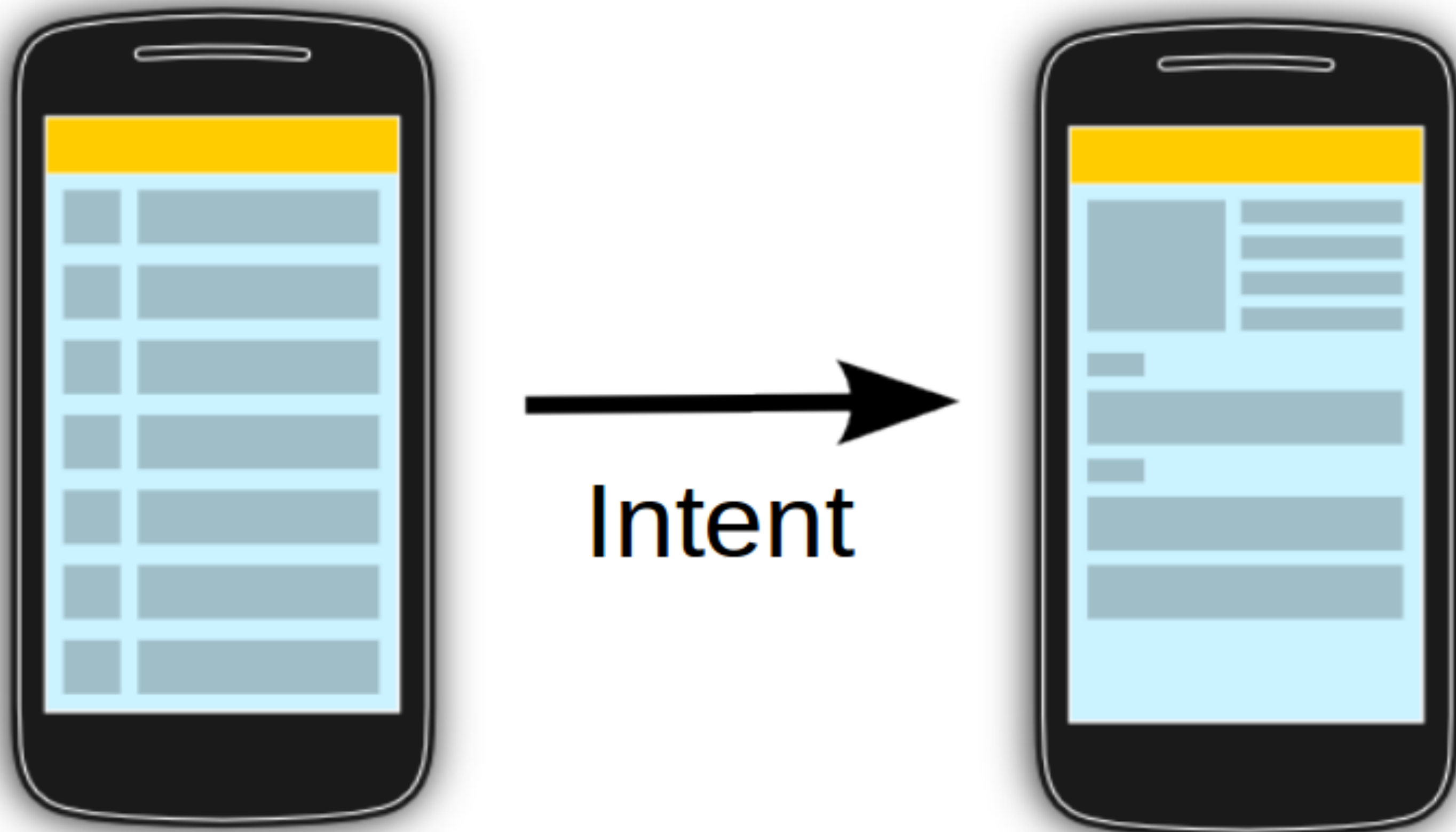


Intents and Intent Bus

Renault@lrde.epita.fr



Main Idea



What is the goal of Intents?



Two kind of components in Android

- Activities: GUI for the user
 - ▶ **Game, mails, etc.**
- Services: background tasks
 - ▶ **music player, sync., etc.**



Goals of the Intents

- Ease the communication between these components
- Provide functionalities for the other apps



How intents are managed by the system?

- Asynchronous communication bus
- Intent filters

Structure of an Intent

action

data



Action examples

action	data	description
ACTION_VIEW	<u>content://contact/people/1</u>	Information about people with UID 1
ACTION_VIEW	tel:123	Display keyboard with pre-composed 123 numero
ACTION_DIAL	<u>content://contact/people/1</u>	Display keyboard already filled with the phone number of people with UID1

Extra parameters for Intents

<u>action</u>	<u>data</u>	category	type	component	extra
---------------	-------------	----------	------	-----------	-------



category: extra informations for the actions

 ACTION_MAIN+CATEGORY_HOME: launch home screen



type: the MIME type in data

 may be deduced sometime by the system



component: the name of the component targeted by the intent



extra: a bundle of informations (e.g. subject for a mail)

Target of an Intent



Explicit target:

- The name is specified in the Intent
- The specified target will be triggered

```
Intent intent = new Intent(getApplicationContext(),  
                             MyClass.class);  
startActivity(intent);
```



Implicit target

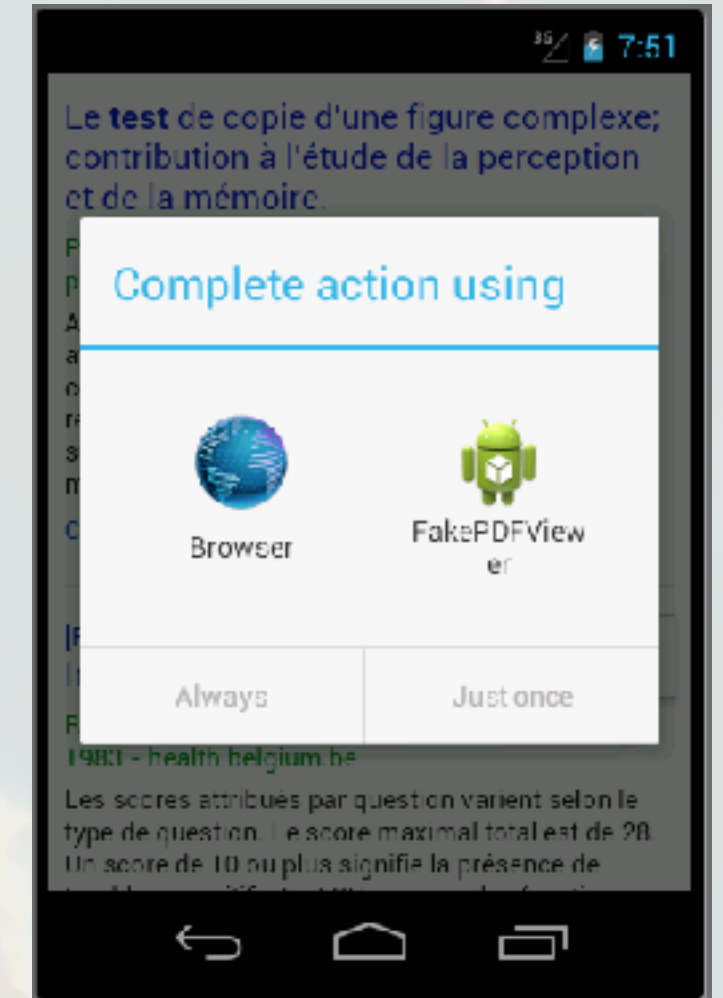
- The system have a lookup to find the best component to trigger
- This lookup is called Intent Resolution

```
Intent intent = new Intent(Intent.ACTION_DIAL);  
intent.setData(Uri.parse("tel:"+phone));  
startActivity(intent);
```

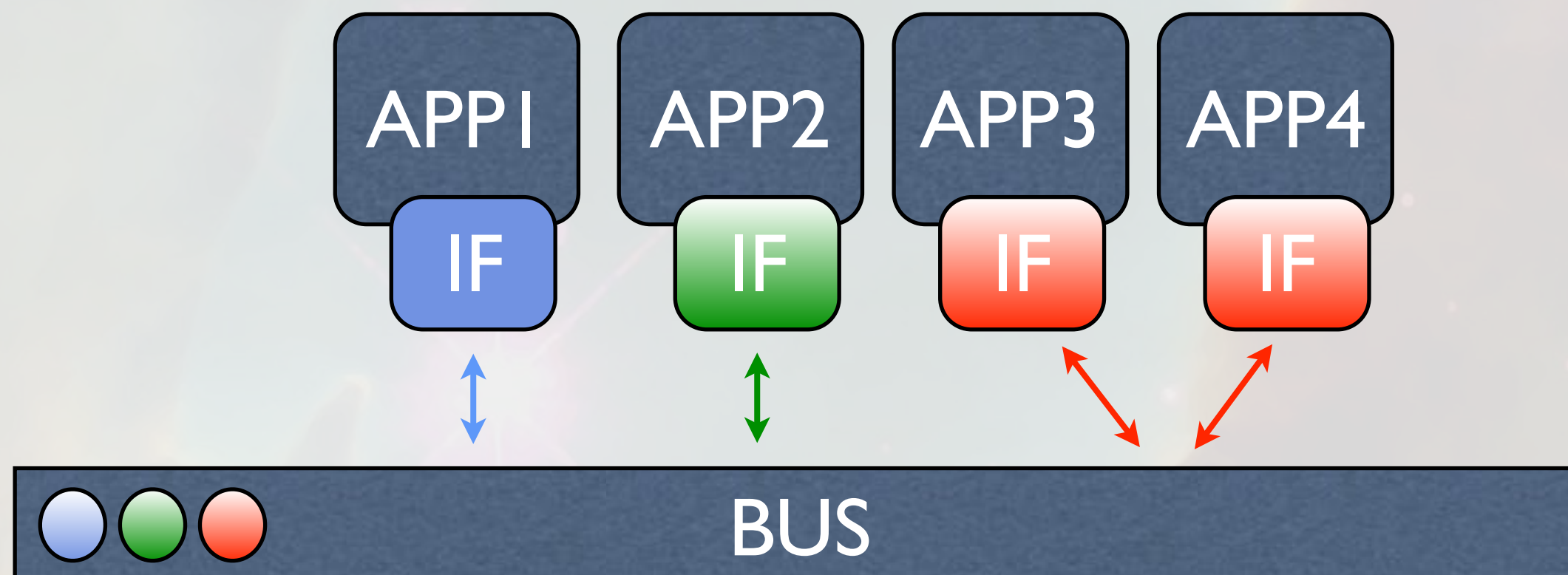

Intent Resolution

Intents Filters

- Each class define its realizable actions
- In the AndroidManifest.xml



The lookup explores all Intent Filters



Declare an Intent Filter

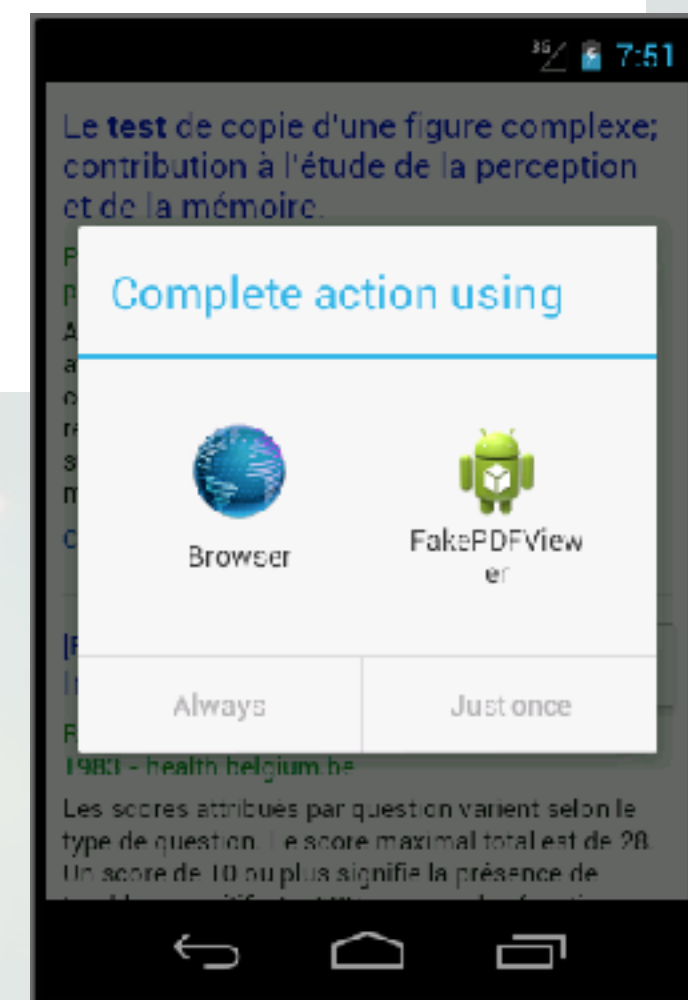
314



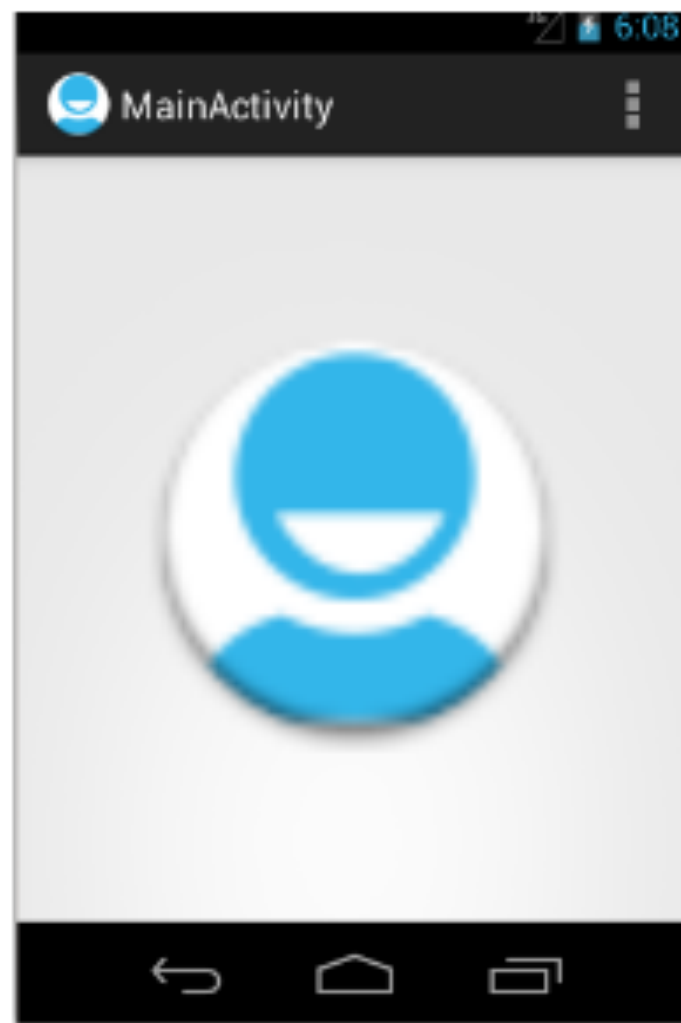
For instance, build an application that may read PDF

Modify the AndroidManifest.xml

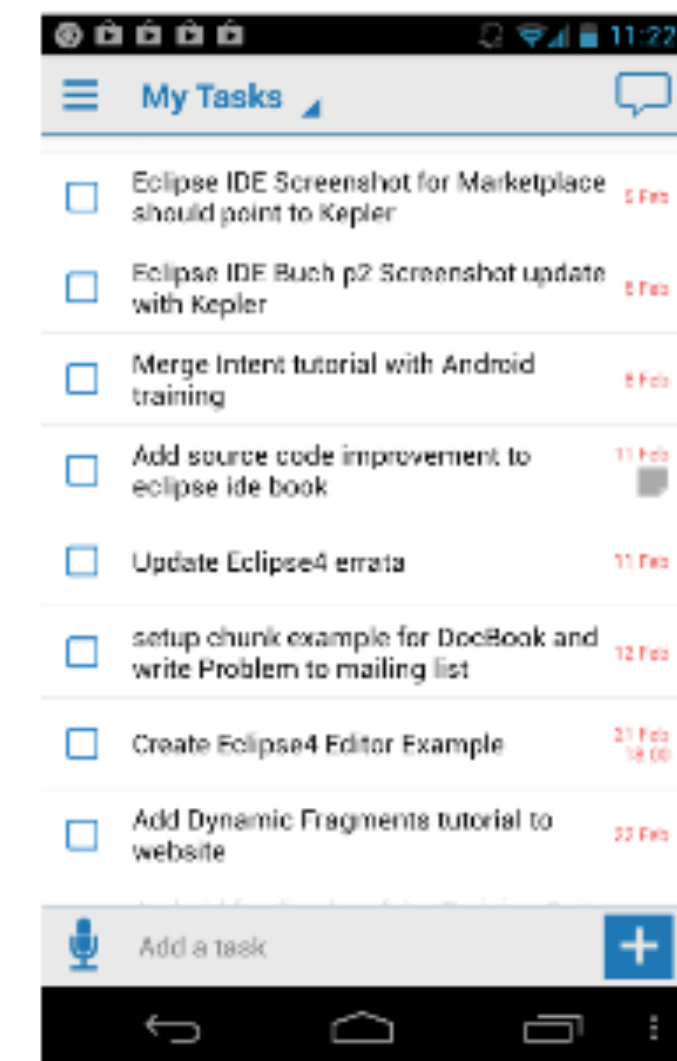
```
<intent-filter>  
  <action  
    android:name="android.intent.action.VIEW" />  
  <category  
    android:name="android.intent.category.DEFAULT" />  
  <category  
    android:name="android.intent.category.BROWSABLE" />  
  <data  
    android:mimeType="application/pdf"  
</intent-filter>
```



Result from an Intent



Intent + resultCode
provided by called
activity



`onActivityResult(requestCode, resultCode, intent)`

requestCode
provided by Android to
identify which activity
type was started

startActivityForResult:

- Identify a call
- Result is provided in onActivityResult

Summary



Intent Bus

- Helps the communication between components
- Circulation of asynchronous messages



The target of an Intent can be

- Explicit: if the name of the component is specified
- Implicit: the system tries to find the best application
 - ▶ **A component can have multiple Intent Filters**
 - ▶ **If multiple component can answer, the user choose!**



We can obtain results after the return of a a component triggered by an Intent



Services

Renault@lrde.epita.fr



What is the goal of Services?

320



Background Tasks

- Similar to services on Linux
- No GUI
- Useful for running a background task

▶ **Music for instance**



Lifecycle is simplified



A Service can be controlled by an activity or an widget

- in other words, we (most-of-the-time) needs an UI to interact




Services can be shared among multiple components


More details

321

Warning!

-  By default, the service run in the process that triggered the service
-  Neither a thread, nor a separated processus by default!

The system only

-  instantiate the service
-  run the callbacks
-  the developer has to allocate ressources, use dedicated thread, etc.

Two kind of services exists

 Bounded:

- ▶ a single instance
- ▶ runs continuously

 Unbounded:

- ▶ runs as long as some component use it

Unbounded Services

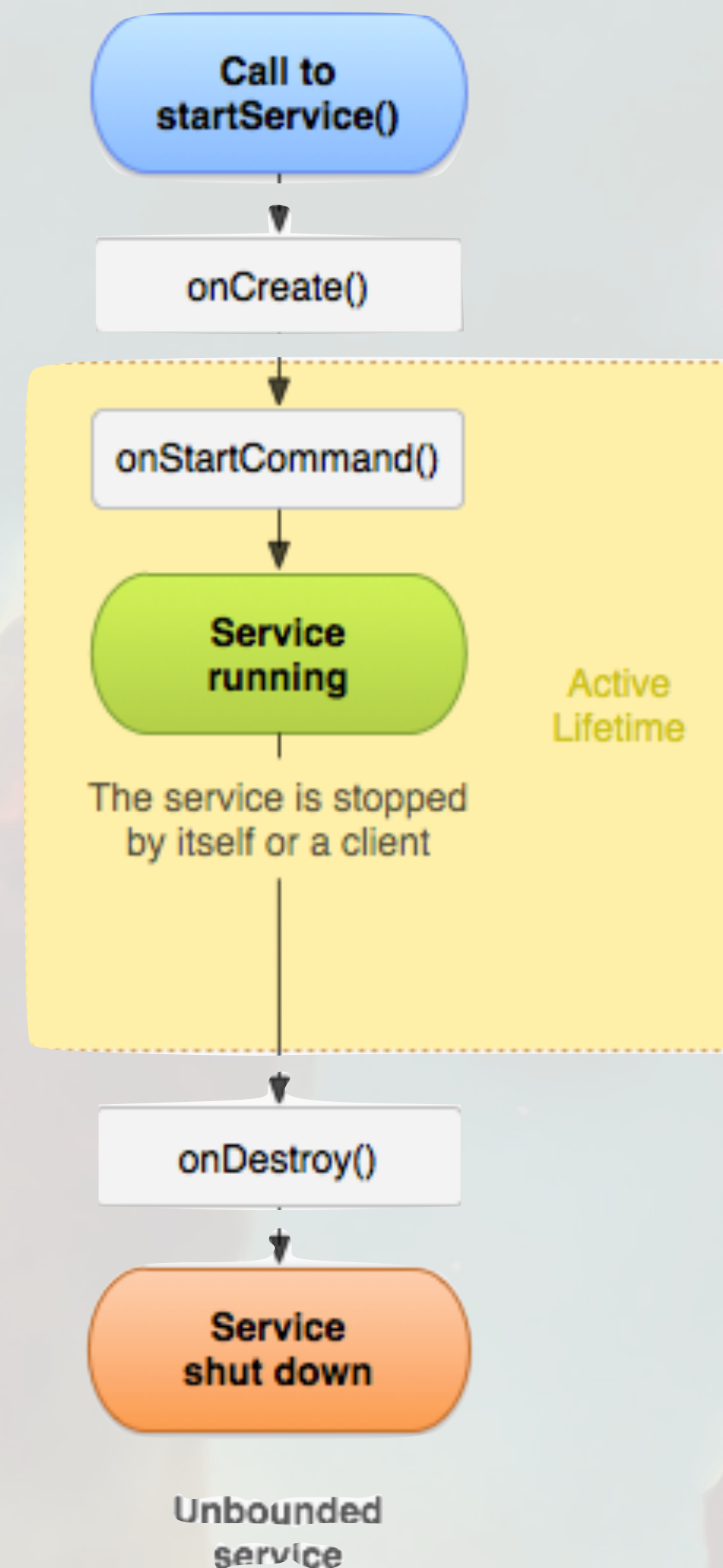
Running the service

- onStartService
- Parameters are passed through Intents

One-way communication

Stopping the service

- stopService
- The number of start request is not taking in account for stopping the service



Unbounded Service: Media Player Example (1/2)

323

```
public class UnboundedService extends Service {
    MediaPlayer mediaPlayer;

    @Override
    public IBinder onBind(Intent intent) {
        return null;
    }

    @Override
    public void onCreate() {
        mediaPlayer = MediaPlayer.create(this, R.raw.mymusic);
        mediaPlayer.setLooping(false);
    }

    @Override
    public void onDestroy() {
        mediaPlayer.stop();
    }
}
```

Unbounded Service: Media Player Example (2/2)

324

```
@Override
public int onStartCommand(Intent intent, int flags, int
startID) {
    mediaPlayer.start();
    // START_STICKY to continue until the
    // service is stopped
    return START_STICKY;
}
}
```



Declaration in the AndroidManifest.xml

```
<service
    android:name=".UnboundedService"
    android:enabled="true">
</service>
```

Bounded Services

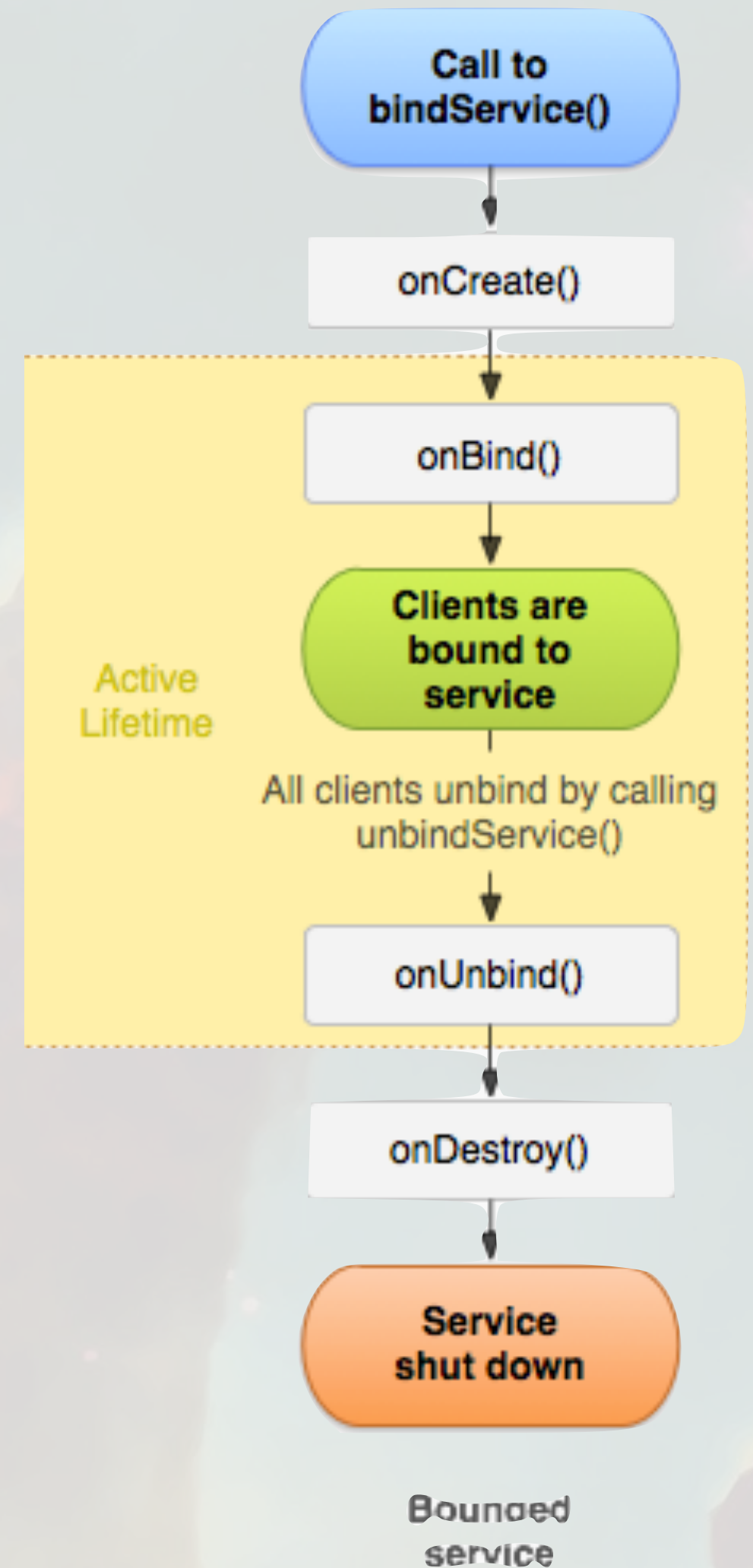
Starting service

- onBind
- returns a binder, close to notion of client/server
- close to IPC
- First client creates the service

The service is manipulated through the binder

Stopping the service

- Clients disconnect
- Last client destroy the service



Bounded Service Example (1/3)

326

Construct the binder (IBinder)

```
public class IDBinder extends Binder {
    int pseudoRnd = 0;

    public int getNewID() {
        ++pseudoRnd;
        return pseudoRnd;
    }
}
```

Building the service

```
public class BoundedService extends Service {
    @Override
    public IBinder onBind(Intent intent) {
        return myIDBinder;
    }
    IDBinder myIDBinder = new IDBinder();
}
```

Bounded Service Example (2/3)



Connecting with the bounded service

```
boolean isBinded = false;
IDBinder idService;
ServiceConnection mConnection = new ServiceConnection() {
    @Override
    public void onServiceConnected(ComponentName name,
                                   IBinder service) {
        idService = (IDBinder) service;
        isBinded = true;
    }
    @Override
    public void onServiceDisconnected(ComponentName name) {
        isBinded = false;
    }
};
```

Bounded Service Example (3/3)

328

Fix the AndroidManifest.xml

```
<service android:name=".BoundedService" ></service>
```

Connect to the service

```
getApplicationContext()  
    .bindService(new Intent(getApplicationContext(),  
                             BoundedService.class),  
                mConnection, BIND_AUTO_CREATE);
```

Disconnect to the service

```
getApplicationContext().unbindService(mConnection);
```



Call the service

```
idService.getNewID();
```


Summary



Two kind of Services

-  Bounded or not depending on your needs
-  Can be started with different policies in case of early terminaison (START_STICKY)



Services work well with widgets to build background tasks



Services have their own lifecycle



Must handle threads

-  in the example, music player handle that



Broadcast Receiver

Renault@lrde.epita.fr



Definition



Component that responds to events



Most of the events are triggered by the system

- Low battery, screen locked, picture taken...



... BUT some may be triggered by applications

- To inform that something is now available



BroadcastReceivers:

- Don't have a GUI

- But can launch some notifications

Building a BroadcastReceiver

334



Modify the AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
    xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.admin.mybroadcastapplication" >

    <uses-permission
        android:name="android.permission.READ_SMS" />

    <uses-permission
        android:name="android.permission.RECEIVE_SMS" />

    <receiver android:name=".MyReceiver">

        <intent-filter>
            <action android:name="android.permission.READ_SMS" />
            <action android:name="android.permission.RECEIVE_SMS" />
        </intent-filter>

    </receiver>
```

Registering & Understanding Lifecycle

335

Register the BroadcastReceiver

- Specify programmatically
 - ▶ the receiver
 - ▶ the IntentFilter

```
registerReceiver(new MyReceiver(), new IntentFilter  
                ("android.provider.Telephony.SMS_RECEIVED"));
```

Lifecycle

- Simplest lifecycle possible
- Only one action onReceive
 - ▶ In this method, computing must be short and without asynchronous processing

Implementation of a Broadcast Receiver

```
public class MyReceiver extends BroadcastReceiver {
    @Override
    public void onReceive(Context context, Intent intent) {
        NotificationCompat.Builder mBuilder =
            new NotificationCompat.Builder(context);
        mBuilder.setSmallIcon(R.drawable.ic_launcher);
        mBuilder.setContentTitle("Notification Detail!");
        // Create Notification
        Notification notification = new NotificationCompat.Builder(context)
            .setContentTitle("Notification Detail!")
            .setSmallIcon(R.drawable.ic_launcher)
            .setContentText("Notification Detail!")
            .setPriority(NotificationCompat.PRIORITY_DEFAULT)
            .build();
        // Build Notification with Notification Manager
        notificationmanager.notify(0, mBuilder.build());
    }
}
```



No direct access to the UI

Broadcasting its own Events

337

Defining a new event

```
<receiver android:name=".MyReceiver">  
  <intent-filter>  
    <action android:name="com.example.mybroadcastapp.CUSTOM_INTENT" />  
  </intent-filter>
```

Register to an event

```
registerReceiver(new MyReceiver(), new  
    IntentFilter("com.example.mybroadcastapp.CUSTOM_INTENT"));
```

Broadcasting

Normal Broadcast

(sendBroadcast)

- Asynchronous messages
- No ordering between receivers
- Efficient

```
Intent intent = new Intent();  
intent.setAction("com.example.mybroadcastapp.CUSTOM_INTENT");  
sendBroadcast(intent);
```

Ordered Broadcast

(sendOrderedBroadcast)

- One receiver at a time
- Priority can be fixed through android:priority
- Receiver can stop diffusion

```
Intent intent = new Intent();  
intent.setAction("com.example.mybroadcastapp.CUSTOM_INTENT");  
sendOrderedBroadcast(intent);
```

A word on Security



BroadcastReceiver use Context that

- allows access to application-specific resources & classes
- etc.



Ensure that you only work on Intents or string that are in your namespace



Some applications do not respect IntentFilter

- This can be fight using `android:exported=false`



All applications can target a broadcast receiver

- Fix that by using Android permissions

Summary



BroadcastReceiver offer a way to trace all the system events



Two kind of broadcasting



ordered



asynchronous



LocalBroadcastManager



Do not use shared memory (inter-processes)



High security: we can share sensitive information



Other application cannot target these managers



Content Providers

Renault@lrde.epita.fr



Inter-processes communication

344

How to exchange information
between processes?

Send Intents

- fill the extra with a lot of informations
 - ▶ **Management can be hard**

Share information through database

- Not available for inter-process communication

Use files

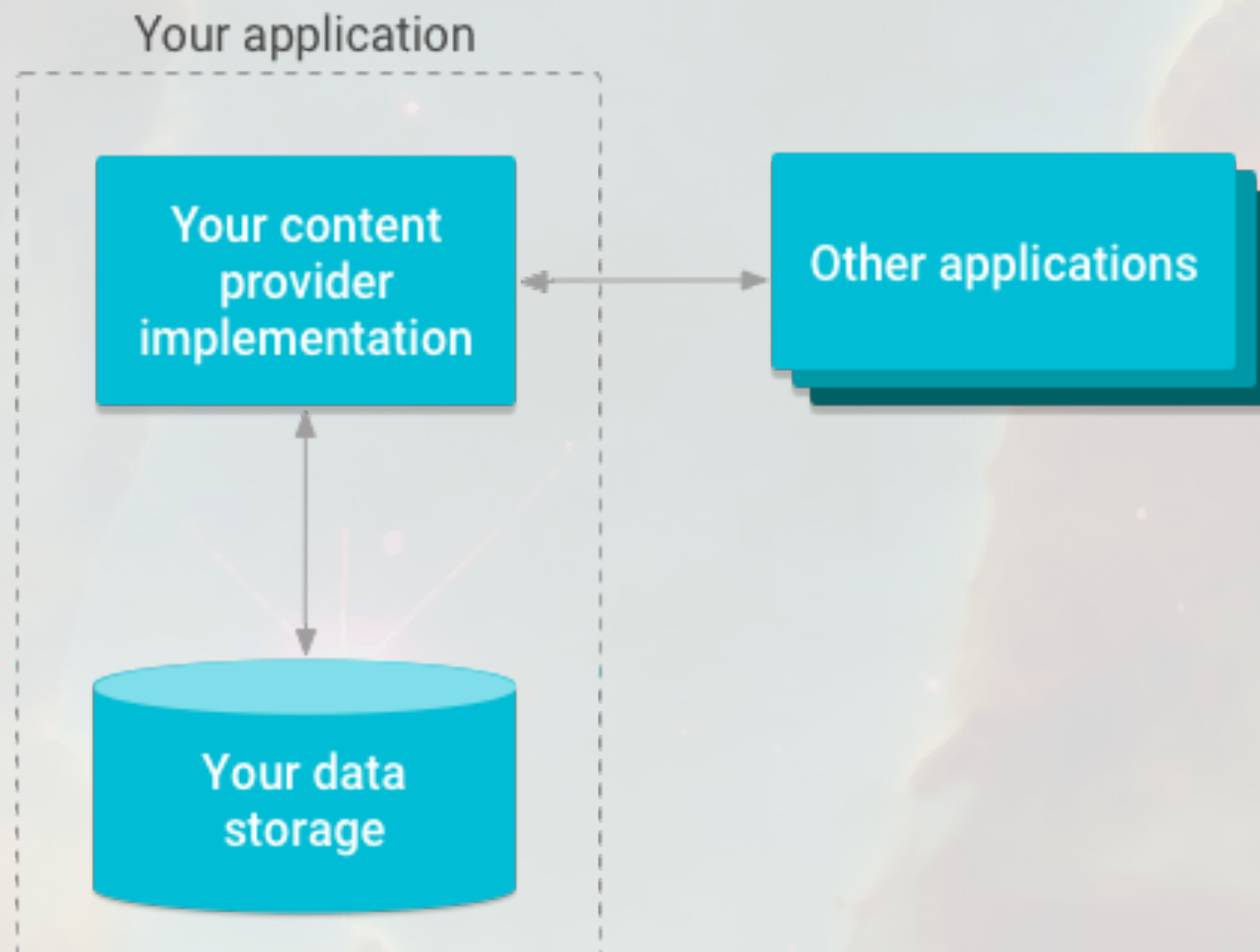
- Read / Write must be synchronized
- The format must be fixed
- cannot restrict to a specific application

ContentProviders



A provider

- Handle some content (data)
- Manage this content through a database (for instance)
- Must ensure the validity of its database
- May not know its clients

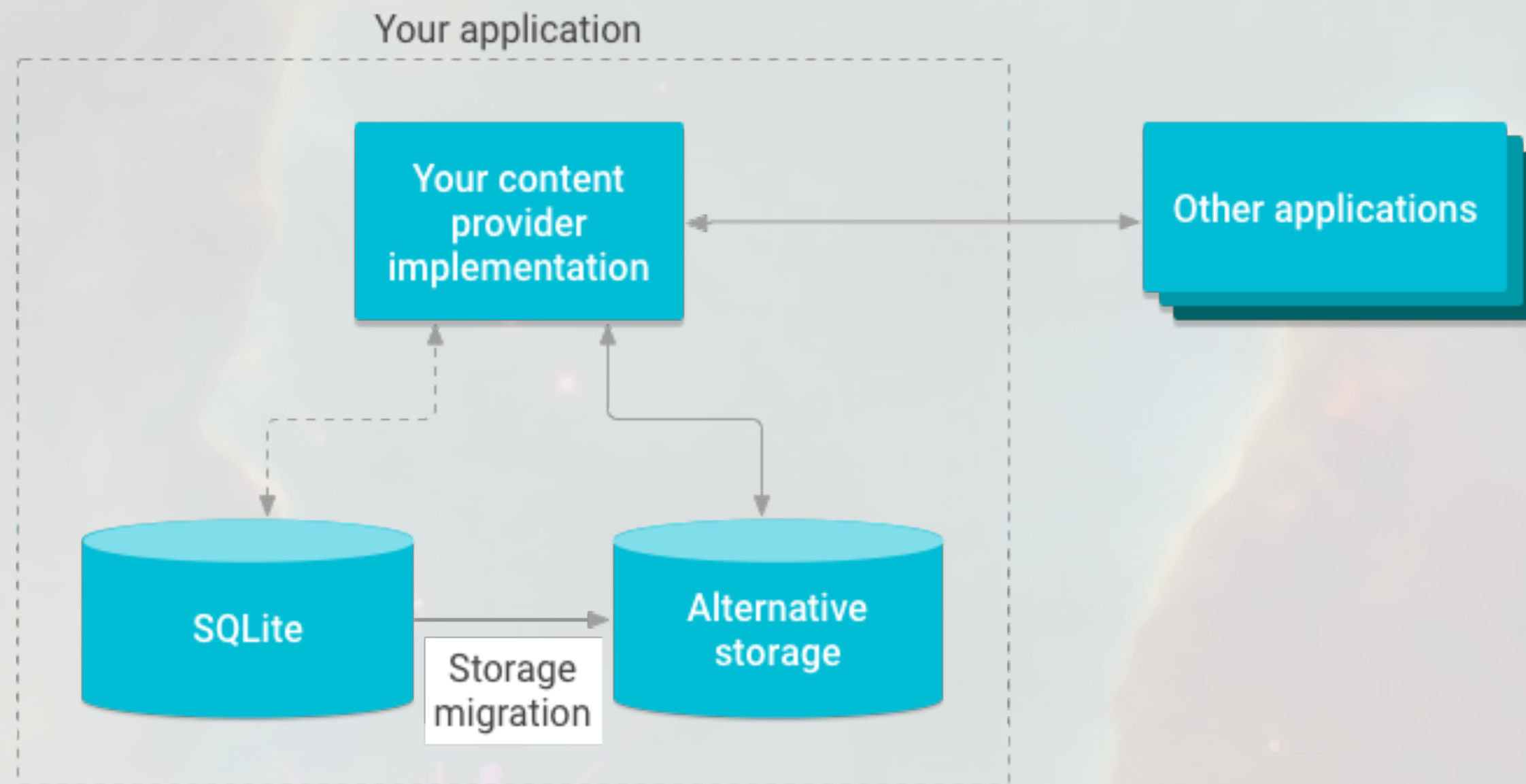


Advantages of ContentProviders? (1/2)



Nice abstraction

- Changing the underlying representation does not affect other applications



You can use it even if you don't plan to share with other Applications

Advantages of ContentProviders? (2/2)

347



Granular control over the permissions for accessing data

- Restrict access
- Grant blanket
- Configure read / write permissions



Can be used to mask many sources

- Your data appears as a single one




Nice pattern that abstract your data

- SQL or raw files are accessed the same way







Wrap around read / write data

-  Data is manipulated without knowing the underlying structure
-  Can be seen as a direct access to a database row



Offers CRUD features

-  Create: create a new data
-  Remove: remove some existing data
-  Update: update existing data
-  Delete: delete existing data



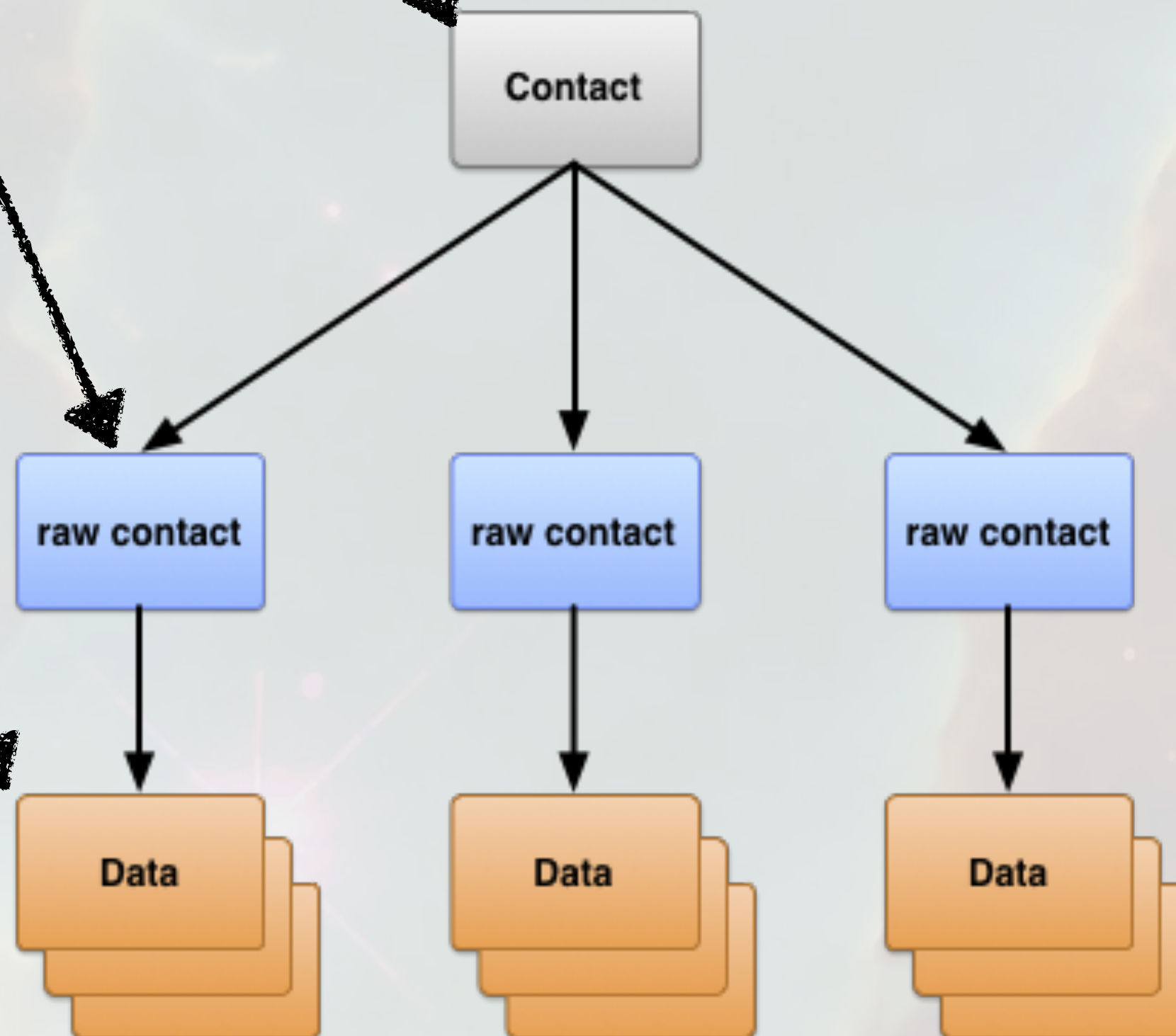
Implement android.database.Cursor

Using existing ContentProvider (Contacts)



Contacts structure

- ContactContract.Contracts: store all contacts
- ContactContract.RawContact: store summary for each contact
- ContactContract.Data: SMS, mails, etc.



Access to Contacts



Modify the AndroidManifest.xml

```
<manifest
  xmlns:android="http://schemas.android.com/apk/res/android"
  package="com.example.admin.mycontactapplication"
>
<uses-permission
  android:name="android.permission.READ_CONTACTS"
/>
```



Build the request

```
ContentResolver cr = getContentResolver();
Cursor cur = cr.query(ContactsContract.Contacts.CONTENT_URI,
  null, null, null, null);
```



Parameters for filtering etc.

Exploiting the cursor

```
if (cur.getCount() > 0) {
    while (cur.moveToNext()) {
        String name = cur.getString(
            cur.getColumnIndex(
                ContactsContract.Contacts.DISPLAY_NAME));

        Toast.makeText(getApplicationContext(), name,
            Toast.LENGTH_SHORT).show();
    }
}
```



We can then look in the other bases if there are additional informations

Define a ContentProvider

Define your own URI

```
contents://com.example.admin.mycontactapplication
```

Implement ContentProvider

onCreate()

Prepare ContentProvider

getType(Uri)

Return the MIME type of the URI

delete(...)

Suppress an entry

insert(...)

Insert a new entry

Update(...)

Update a data

Create Database (1/2)

```
private SQLiteDatabase db;

static final String DATABASE_NAME = "mydb";

static final String TABLE_NAME = "names";

static final int DATABASE_VERSION = 1;

static final String CREATE_DB_TABLE =
    " CREATE TABLE "
    + TABLE_NAME
    + " (id INTEGER PRIMARY KEY AUTOINCREMENT, "
    + " name TEXT NOT NULL);";

private static class DatabaseHelper extends SQLiteOpenHelper {
    DatabaseHelper(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }
}
```

Create Database (2/2)

```
@Override
public void onCreate(SQLiteDatabase db) {
    db.execSQL(CREATE_DB_TABLE);
}

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion,
    int newVersion) {
    db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME);
    onCreate(db);
}
}
```

Define Constants

```
public class MyProvider extends ContentProvider {
    static final String PROVIDER_NAME =
        "com.example.admin.mycontactapplication.MyProvider";
    static final String URL =
        "content://" + PROVIDER_NAME + "/cte";

    static final Uri CONTENT_URI = Uri.parse(URL);

    static final String id = "id";
    static final String name = "name";
    static final int uriCode = 1;

    static final UriMatcher uriMatcher;
    private static HashMap<String, String> values;

    static {
        uriMatcher = new UriMatcher(UriMatcher.NO_MATCH);
        uriMatcher.addURI(PROVIDER_NAME, "cte", uriCode);
        uriMatcher.addURI(PROVIDER_NAME, "cte/*", uriCode);
    }
}
```

Implement methods (1/4)

```
@Override
public String getType(Uri uri) {
    switch (uriMatcher.match(uri)) {
        case uriCode:
            return "vnd.android.cursor.dir/cte";
        default:
            throw new IllegalArgumentException
                ("Unsupported URI: " + uri);
    }
}
```

```
@Override
public boolean onCreate() {
    Context context = getContext();
    DatabaseHelper dbHelper = new DatabaseHelper(context);
    db = dbHelper.getWritableDatabase();
    if (db != null)
        return true;
    return false;
}
```

Implement methods (2/4)

```
@Override
public int delete(Uri uri, String selection,
                  String[] selectionArgs) {
    int count = 0;
    switch (uriMatcher.match(uri)) {
        case uriCode:
            count = db.delete(TABLE_NAME,
                              selection,
                              selectionArgs);

            break;
        default:
            throw new IllegalArgumentException
                ("Unknown URI " + uri);
    }
    getContext().getContentResolver().notifyChange(uri, null);
    return count;
}
```

Implement methods (3/4)

```
@Override
public Uri insert(Uri uri, ContentValues values) {
    long rowID = db.insert(TABLE_NAME, "", values);

    if (rowID > 0) {
        Uri _uri =
            ContentUris.withAppendedId(CONTENT_URI, rowID);
        getContext().getContentResolver()
            .notifyChange(_uri, null);
        return _uri;
    }

    throw new SQLException("Failed to add a record into " + uri);
}
```

Implement methods (4/4)

```
@Override
public Cursor query(Uri uri, String[] projection,
                   String selection, String[] selectionArgs,
                   String sortOrder) {
    SQLiteQueryBuilder qb = new SQLiteQueryBuilder();
    qb.setTables(TABLE_NAME);
    switch (uriMatcher.match(uri)) {
        case uriCode:
            qb.setProjectionMap(values);
            break;
        default:
            throw new IllegalArgumentException("Unknown URI " + uri);
    }
    if (sortOrder == null || sortOrder == "") {
        sortOrder = name;
    }
    Cursor c = qb.query(db, projection, selection,
                       selectionArgs, null, null, sortOrder);
    c.setNotificationUri(getContext().getContentResolver(), uri);
    return c;
}
```

Add element to the database

360

New element to the database

```
public void onClickAddName(View view) {
    ContentValues values = new ContentValues();
    values.put(MyProvider.name, ((EditText)
        findViewById(R.id.txtName)).getText().toString());
    Uri uri = getContentResolver()
        .insert(MyProvider.CONTENT_URI, values);
}
```

Modify the AndroidManifest.xml

```
<provider
    android:name=".MyProvider"
    android:authorities=
        "com.example.admin.mycontactapplication.MyProvider"
    android:exported="true"
    android:multiprocess="true">
</provider>
```


Client (1/2)



Implement LoaderManager.LoaderCallbacks<Cursor>

```
CursorLoader cursorLoader;

public void onClickDisplayNames(View view) {
    getSupportLoaderManager().initLoader(1, null, this);
}

@Override
public Loader<Cursor>
onCreateLoader(int arg0, Bundle arg1) {
    cursorLoader =
        new CursorLoader(this, Uri.parse("content://
com.example.admin.mycontactapplication.MyProvider/cte"),
            null, null, null, null);
    return cursorLoader;
}
```

Client (2/2)

362



Implement LoaderManager.LoaderCallbacks<Cursor>

```
@Override
public void onLoadFinished(Loader<Cursor> arg0,
                          Cursor cursor) {
    cursor.moveToFirst();
    StringBuilder res = new StringBuilder();
    while (!cursor.isAfterLast()) {
        res.append("\n"
            + cursor.getString(cursor.getColumnIndex("id"))
            + "_"
            + cursor.getString(cursor.getColumnIndex("name")));
        cursor.moveToNext();
    }
    resultView.setText(res);
}
```

Summary

363



Content Providers

- Are used on databases
 - ▶ **But not necessarily**
- Efficient way to share informations between threads
 - ▶ **permissions can be fixed**
- May be used by a single application to provide an abstraction layer



Access is done through ContentResolver

- Predefined ones: CursorLoader, ...
- URI must be known



You can now build your own database

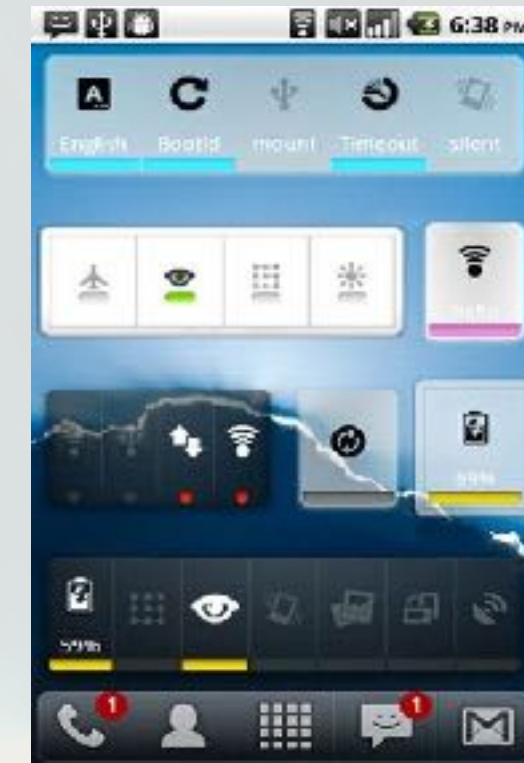
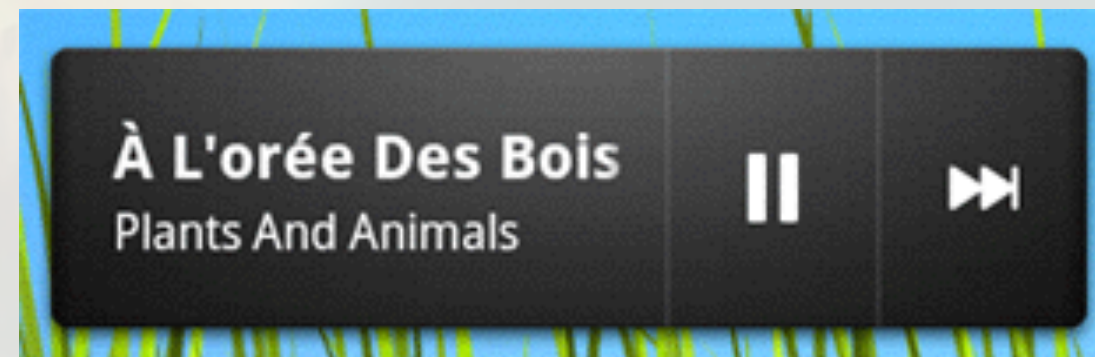


Widgets

Renault@lrde.epita.fr



Description



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!

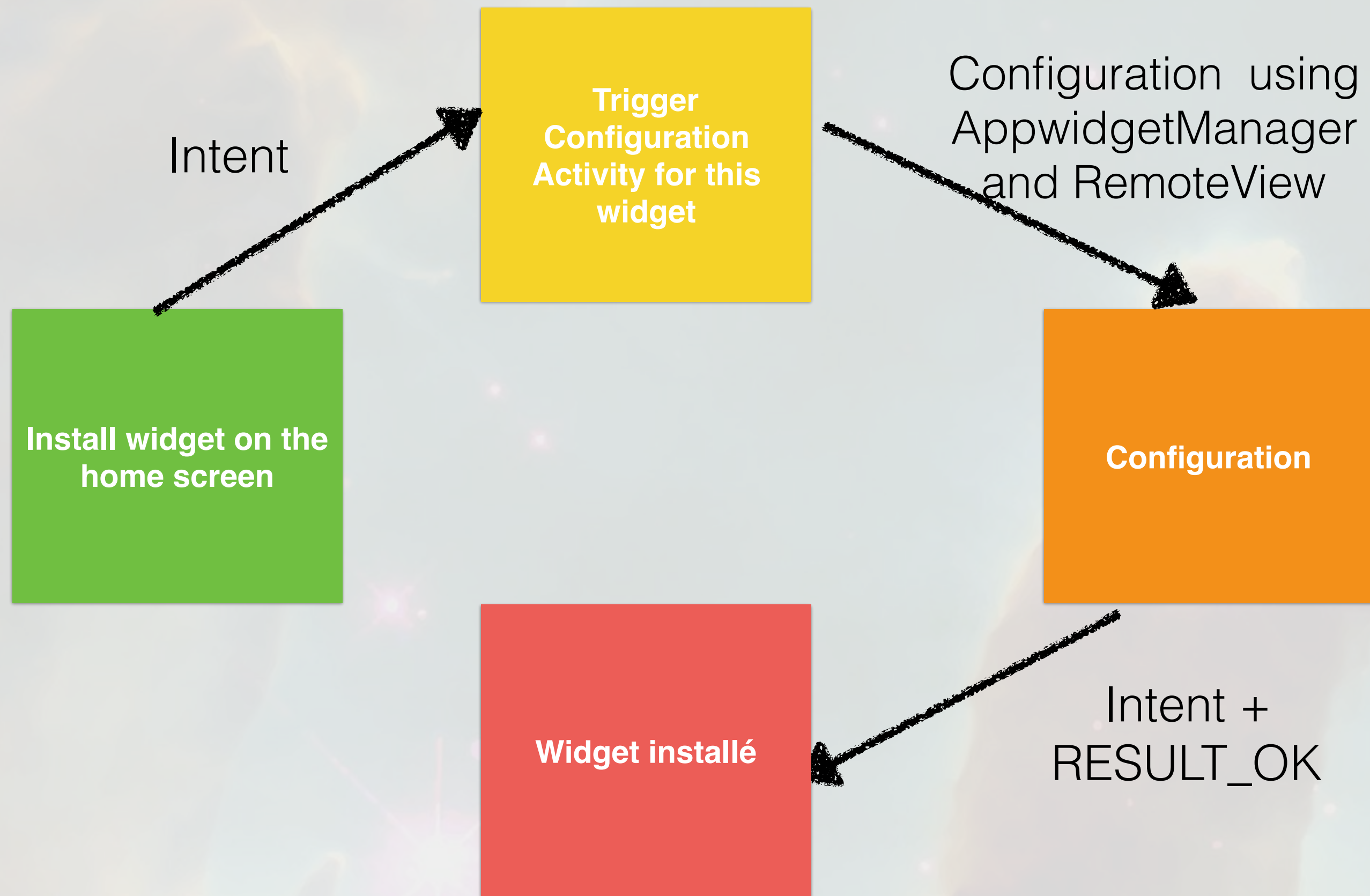
Widgets Lifecycle



Widgets works by callbacks

onEnabled()	Called when installed on the HomeScreen (<u>first instance only</u>)
onDeleted()	Called when a widget is removed from the HomeScreen
onDisabled()	Called when the last widget is removed from the HomeScreen
onUpdate()	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 widget 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?

376



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();
```

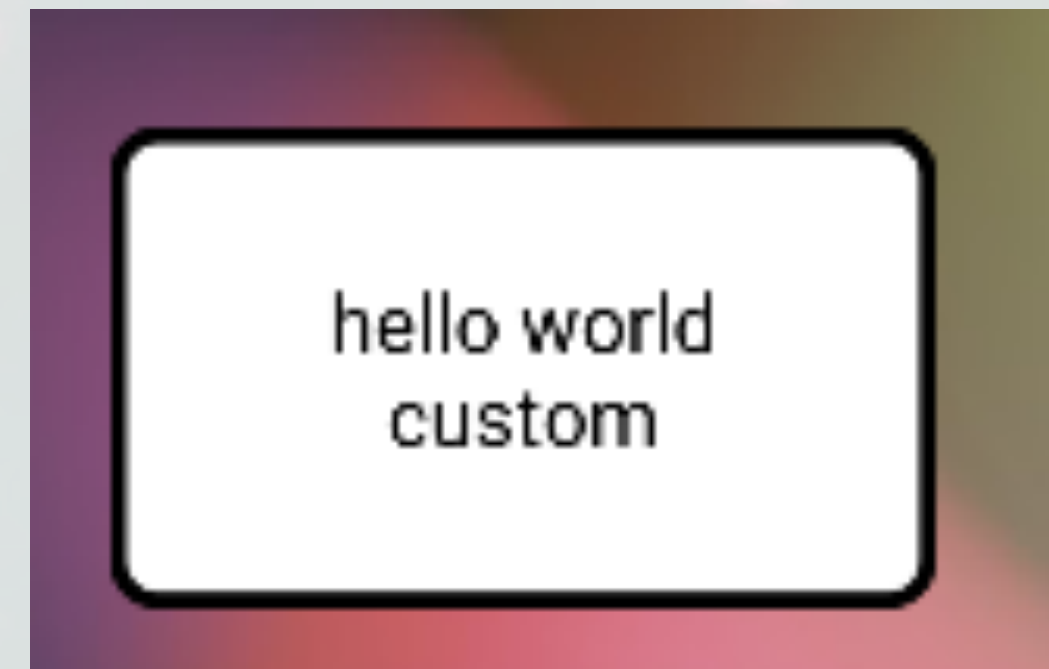

How to build friendly widgets?

377



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)

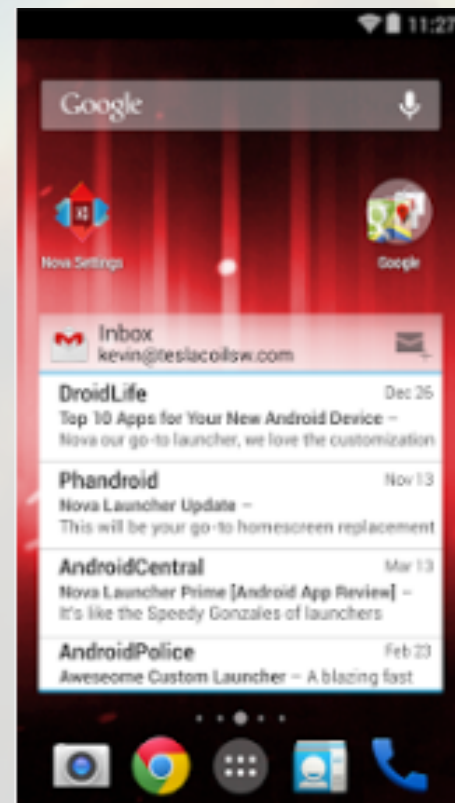


Packages and HomeScreen

Renault@lrde.epita.fr



Home Screen



Goals

- Displays Widgets
- Provides access to applications
- Configurable by the user

Offers a screen that display "important" informations

- An important information depend of the user

How to Access to Installed Application?

383



PackageInstaller

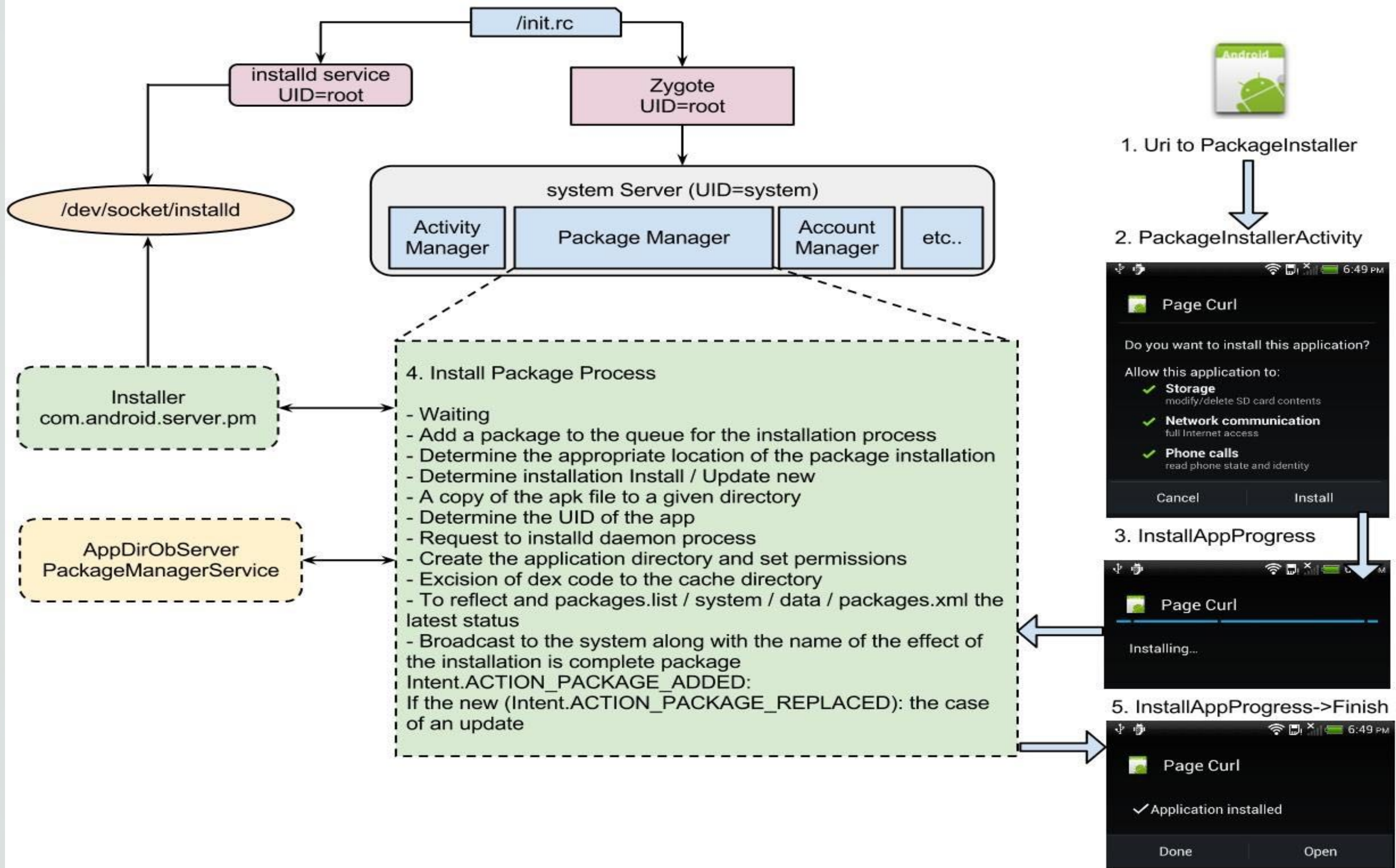
- Interface for installing applications
- Use PackageManager to manage applications



PackageManager

- Linux Daemon
- During the install of an application
 - ▶ **Open the APK**
 - ▶ **Grab all valuable information**
- Applications are installed in directories
 - ▶ **/system/app: the preinstalled applications**
 - ▶ **/data/app: user applications**
 - ▶ **/data/data/<app_name>: data of each application**

Lifecycle



PackageManager Details

Access to a lot of informations

 QueryBroadcastReceiver:

▶ **What are the BroadcastReceiver that can reply to an Intent?**

 QueryIntentActivities:

▶ **What are the Activities that can reply to an Intent?**

 QueryIntentService:

▶ **What are the Service that can reply to an Intent?**

 GetApplicationEnabled Settings:

▶ **Read permission of an Application**

 SetApplicationSettings:

▶ **Modify existing permissions**

 ...

Accessing to Application's Basic Informations

```
public class AppDetail {
    CharSequence label; CharSequence name; Drawable icon;
}
private PackageManager manager;
private List<AppDetail> apps;

private void loadApps() {
    manager = getPackageManager();
    apps = new ArrayList<AppDetail>();
    Intent i = new Intent(Intent.ACTION_MAIN, null);
    i.addCategory(Intent.CATEGORY_LAUNCHER);
    List<ResolveInfo> availableActivities =
        manager.queryIntentActivities(i, 0);
    for(ResolveInfo ri:availableActivities){
        AppDetail app = new AppDetail();
        app.label = ri.loadLabel(manager);
        app.name = ri.activityInfo.packageName;
        app.icon = ri.activityInfo.loadIcon(manager);
        apps.add(app);
    }
}
```

Define a GUI for displaying this information

387



Build a simple View

Here a button that will launch a listView on click



... and modify AndroidManifest.xml

```
<activity
    android:name=".MainActivity"
    android:label="@string/app_name"
    android:theme=
        "@android:style/Theme.Wallpaper.NoTitleBar.Fullscreen"
    android:launchMode="singleTask"
    android:stateNotNeeded="true" >
<intent-filter>
    <action android:name="android.intent.action.MAIN" />
    <category android:name="android.intent.category.HOME" />
    <category android:name=
        "android.intent.category.DEFAULT" />
    <category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
```

The "Activity" HomeScreen



Build a simple App to access all existing Apps

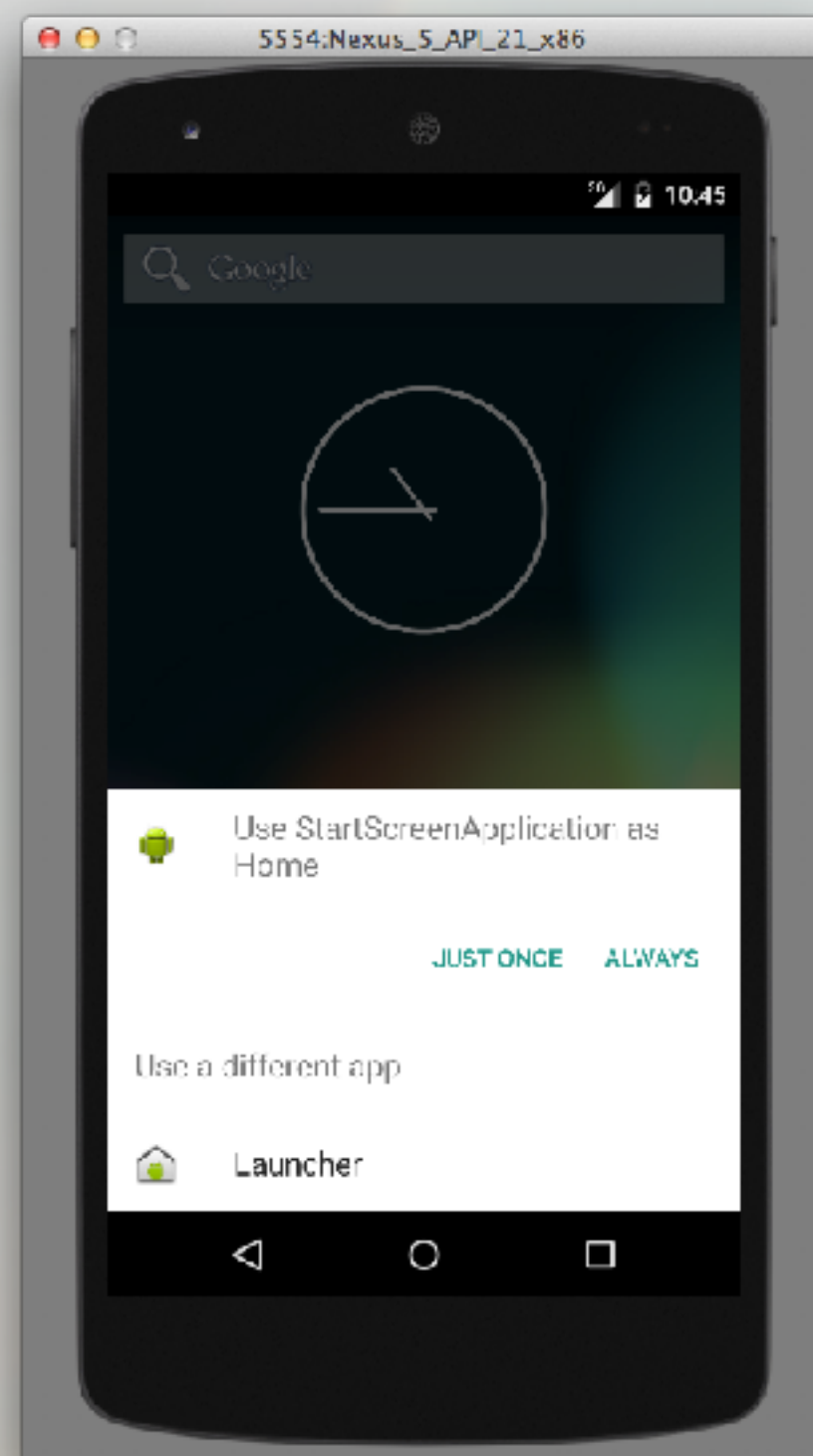
- Fill the list view with some information

```
private ListView list;
private void loadListView(){
    list = (ListView)findViewById(R.id.apps_list);
    ArrayAdapter<AppDetail> adapter =
        new ArrayAdapter<AppDetail>(this, R.layout.list_item, apps) {
        @Override
        public View getView(int position, View convertView, ViewGroup parent) {
            if(convertView == null){
                convertView = getLayoutInflater().inflate(R.layout.list_item, null);
            }
            ImageView appIcon =
                (ImageView)convertView.findViewById(R.id.item_app_icon);
            appIcon.setImageDrawable(apps.get(position).icon);
            TextView appLabel =
                (TextView)convertView.findViewById(R.id.item_app_label);
            appLabel.setText(apps.get(position).label);
            TextView appName =
                (TextView)convertView.findViewById(R.id.item_app_name);
            appName.setText(apps.get(position).name);
            return convertView;
        }
    };
};
```

Setup the Home Screen



When you launch the Application, you are asked to decide if you want to change your default home screen



Summary



PackageManager helps to get informations about installed applications



This informations are useful when we want to build an HomeScreen



Defining Lock-Screen is possible



Same mechanism



Register to Events:

- ▶ **BOOT_COMPLETED**: to trigger lock-screen after a reboot
- ▶ **ACTION_SCREEN_OFF, ACTION_SCREEN_ON**: to trigger lock-screen activation
- ▶ ... other events



Be careful with security

- ▶ **You have to propose security schemes**



Asynchronous Tasks

Renault@lrde.epita.fr



Technical Considerations

394



In an Application, only the UI Thread can update the GUI

Must **NOT** be blocked



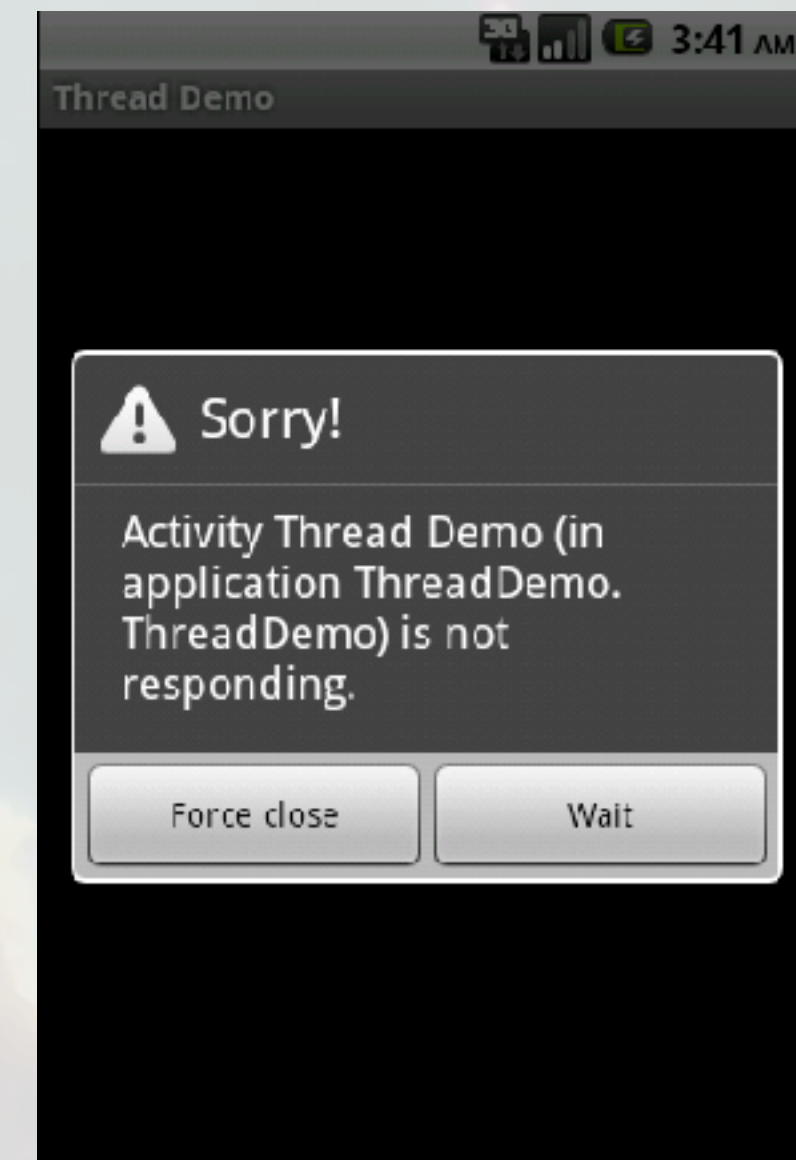
For long processing, 2 options

Build a new thread that will perform computation

- ▶ You have to communicate with UI thread if you want to display the result
- ▶ may be hard
- ▶ Not in this chapter

Use asynchronous tasks

- ▶ Dedicated component for performing background operation with GUI updates



Asynchronous Tasks



Ease manipulation of the UI thread

- Background operation
- Builtin callbacks to update the GUI
 - ▶ Useful for progress bar for instance



MUST NEVER REPLACE THREADS

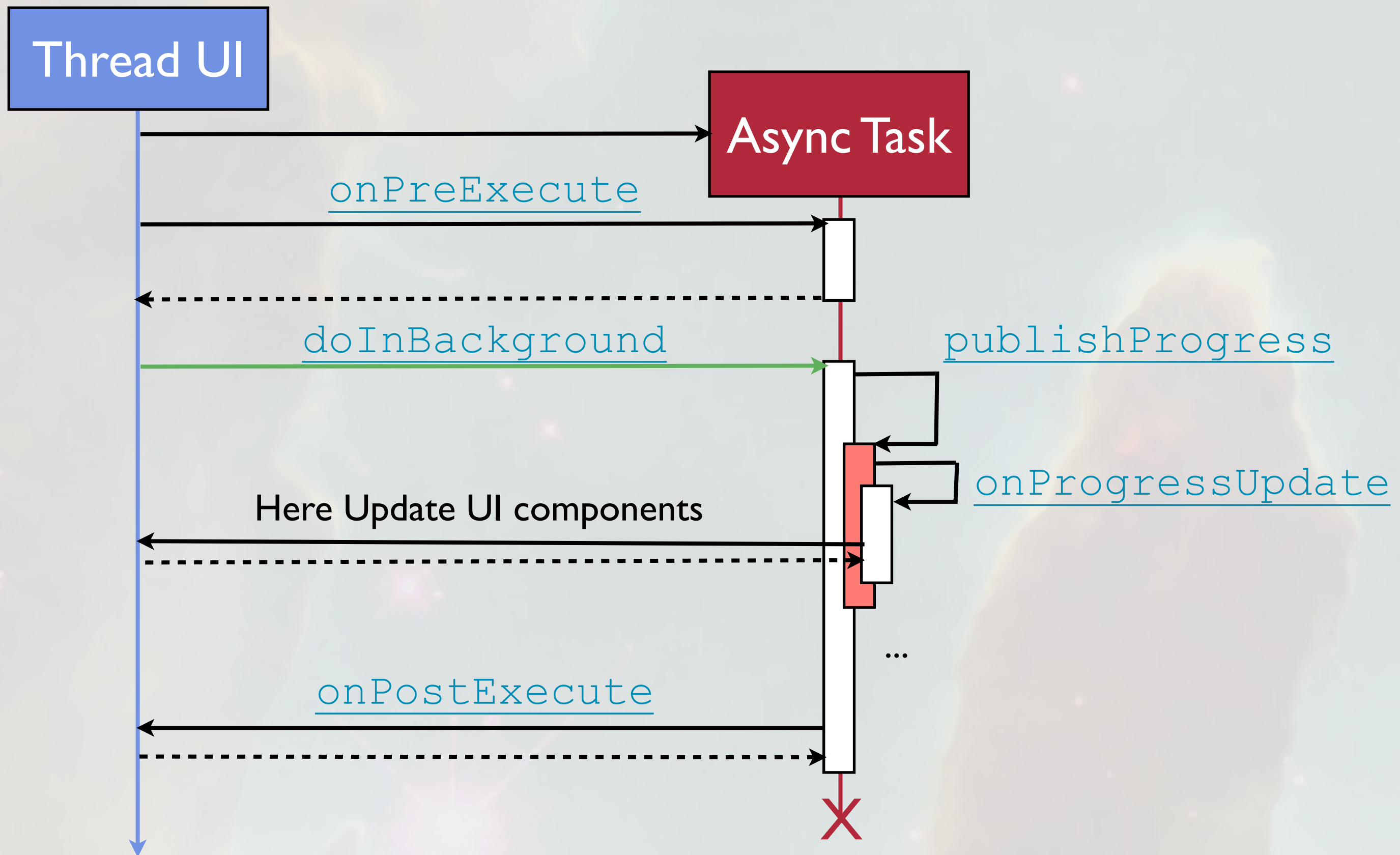
- Should not be used for more than few seconds operations



Use three parameters

- Param:
 - ▶ The type of the parameter used for the computation
- Progress:
 - ▶ The unit used to notify a progress
- Results:
 - ▶ The type of the result of an asynchronous task

Lifecycle



AsyncTask and ProgressBar (1/2)

```
public class BigCompute
    extends AsyncTask<Void, Integer, Void> {
    Context mContext;
    ProgressBar mProgressBar;

    BigCompute(Context c, ProgressBar p) {
        mContext = c;
        mProgressBar = p;
    }

    @Override
    protected void onPreExecute() {
        Toast.makeText(mContext, "OnPreExecute",
            Toast.LENGTH_SHORT).show();
        super.onPreExecute();
    }

    @Override
    protected void onPostExecute(Void aVoid) {
        super.onPostExecute(aVoid);
    }
}
```

AsyncTask and ProgressBar (2/2)

```
@Override
protected void onProgressUpdate(Integer... values) {
    super.onProgressUpdate(values);
    mProgressBar.setProgress(values[0]);
}

@Override
protected Void doInBackground(Void... params) {
    for (int progress = 0; progress < 100; ++progress) {
        try {
            Thread.currentThread().sleep(1000);
        } catch (Exception e) {}
        publishProgress(progress);
    }
    return null;
}
```

Building an AsyncTask



Instantiation

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Button b = (Button) findViewById(R.id.button_launch);
    b.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            BigCompute bc = new BigCompute(getApplicationContext(),
                (ProgressBar) findViewById(R.id.progressBar));
            bc.execute();
        }
    });
}
```




AsyncTask and Rotation



AsyncTask does not support rotation as-is



Possible workarounds

-  Define the asynchronous task in a Fragment and Apply a setRetainedInstance
-  Associate AsyncTask to Services
-  Relaunch the AsyncTask



No best solutions

Summary

401



Asynchronous Tasks

- Easy way to release UI Thread
- Already predefined handlers
- Not for more than few seconds computing
- Well suited for mixing with ProgressBar



Mix bad with rotation

- Even if some solution exist



Should never replace threads



Threads

Renault@lrde.epita.fr



Overview

405



An Android Application is executed by a process

- A thread, called **main thread** or **UI thread**, is in charge of updating GUI



One thread per component

- The UI Thread of the component at the top of the back stack runs
- Thread UI manage callbacks



An application performing a lot of computing must use different threads

Rules

 **1 - Do not block UI Thread**

 **2 - Only UI Thread can modify UI**

 Android Toolkit UI is not thread safe

```
public void onClick(View v) {  
    new Thread(new Runnable() {  
        public void run() {  
            Bitmap b =  
                loadImageFromNetwork("http://example.com/image.png");  
            mImageView.setImageBitmap(b);  
        }  
    }).start();  
}
```



How to update GUI then ?

407



To update GUI from another Thread

- Use Asynchronous tasks
- Use dedicated methods that take a thread as parameter
 - ▶ **Activity.onRunUI(Runnable)**
 - ▶ **View.post(Runnable)**
 - ▶ **View.postDelayed(Runnable, long)**

```
public void onClick(View v) {
    new Thread(new Runnable() {
        public void run() {
            final Bitmap bitmap =
                loadImageFromNetwork("http://example.com/image.png");
            mImageView.post(new Runnable() {
                public void run() {
                    mImageView.setImageBitmap(bitmap);
                }
            });
        }
    }).start();
}
```

Message Queue & Looper

408



A thread holds a message queue

For all action and callback to perform later



This queue is thread-safe



Messages from this queue will be flushed by the Looper

When a message is received, the looper treat it

To do so, handlers are used

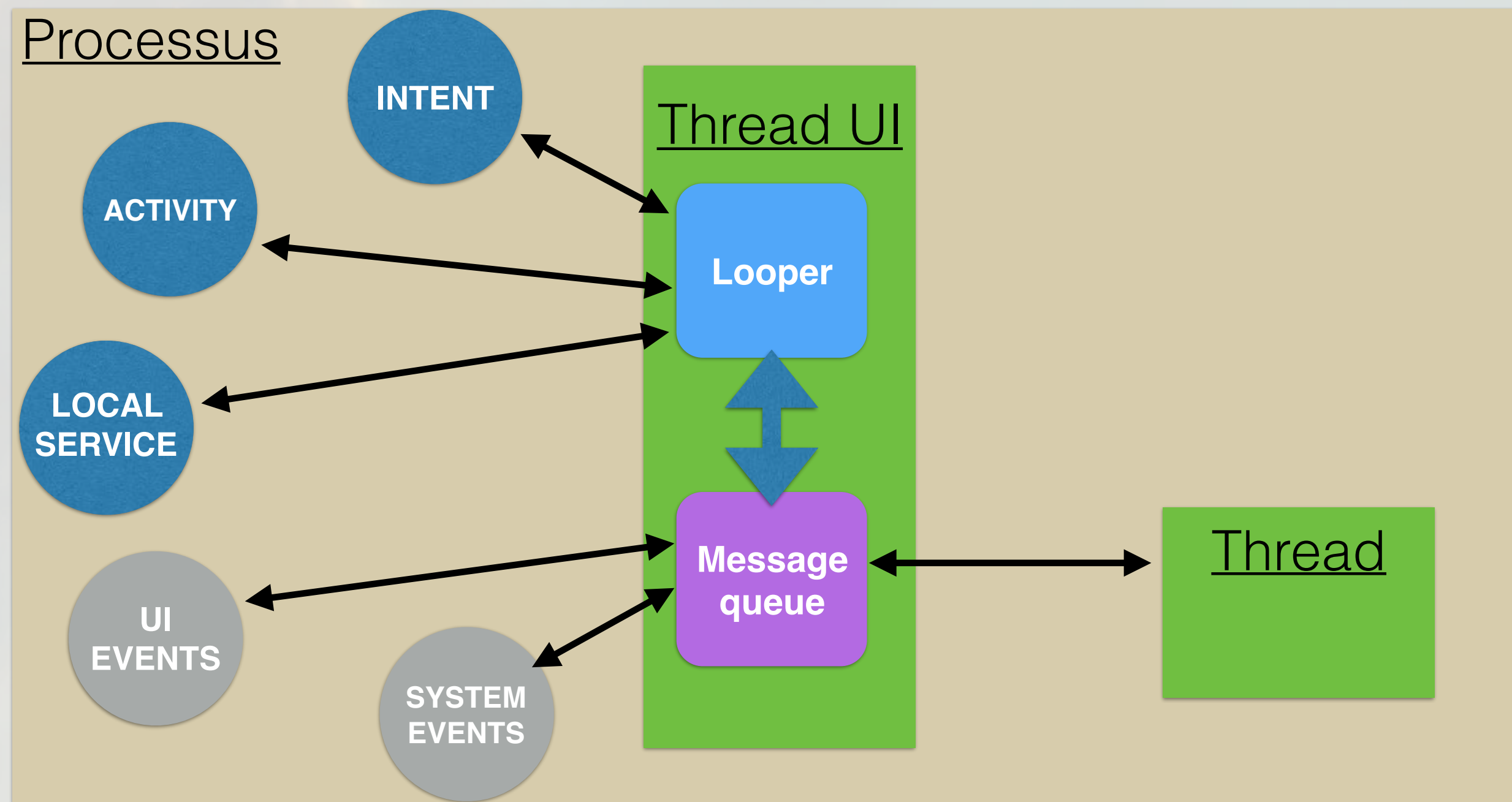


By default, only the UI Thread have a looper and a message queue

How does the looper work?



We can force the UI Thread to do something thanks to the looper



Creating its own Looper



We can build a callback looper in every thread

To do so, use Handlers

```
class LooperThread extends Thread {
    public Handler mHandler;
    public void run() {
        //Initialize the current thread as a looper.
        Looper.prepare();

        //instance a Handler of the current thread
        mHandler = new Handler() {
            // process incoming messages here
            public void handleMessage(Message msg) {
            }
        };

        //Run the message queue in this thread.
        Looper.loop();
    }
}
```

Define Handler in the UI Thread



The handler is associated to a given thread

```
private ThreadCompute mThread;
private Handler mHandler;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    mHandler = new Handler(Looper.getMainLooper()) {
        @Override
        public void handleMessage(Message inputMessage) {
            Toast.makeText(getApplicationContext(),
                inputMessage.toString(), Toast.LENGTH_SHORT)
                .show();
        }
    };
    mThread = new ThreadCompute();
    mThread.start();
}
```

Define Handler in the UI Thread

412



The handler is associated to a given thread

```
void notifyUI() {
    Message completeMessage = mHandler.obtainMessage();
    completeMessage.sendToTarget();
}

class ThreadCompute extends Thread {
    @Override
    public void run() {
        try {
            Thread.sleep(10000);
            notifyUI();
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
};
```

Priorities

THREAD_PRIORITY_AUDIO

THREAD_PRIORITY_URGENT_AUDIO

THREAD_PRIORITY_BACKGROUND

THREAD_PRIORITY_LOWEST

THREAD_PRIORITY_DISPLAY

THREAD_PRIORITY_URGENT_DISPLAY

THREAD_PRIORITY_FOREGROUND

THREAD_PRIORITY_MOST_FAVORABLE

THREAD_PRIORITY_LESS_FAVORABLE

....

Summary



Only the UI Thread can modify the UI



If another component want to modify the UI, it has to trigger an action on the UI Thread



with AsyncTask



with predefined methods



with handler



An application can handle multiple threads



onPrepare should be called to setup the looper



Knowing how to manage threads is important



Battery Management

Renault@lrde.epita.fr



Goals and Problems



Battery is a sensitive component in mobile devices

- Think to it when writing applications
- The application must adapt to battery level
- Restrict interaction with networks



If all applications were sensitive to battery problem, the lifetime of a charge will no longer be a problem



How to build such an application?

Monitor the Battery Status



BatteryManager broadcast information about battery

- 🔊 Prefer updates when the device is charging

```
IntentFilter ifilter =
    new IntentFilter(Intent.ACTION_BATTERY_CHANGED);
Intent batteryStatus = context.registerReceiver(null, ifilter);
// Are we charging / charged?
int status =
    batteryStatus.getIntExtra(BatteryManager.EXTRA_STATUS, -1);
boolean isCharging =
    status == BatteryManager.BATTERY_STATUS_CHARGING ||
    status == BatteryManager.BATTERY_STATUS_FULL;
// How are we charging?
int chargePlug =
    batteryStatus.getIntExtra(BatteryManager.EXTRA_PLUGGED, -1);
boolean usbCharge =
    chargePlug == BatteryManager.BATTERY_PLUGGED_USB;
boolean acCharge =
    chargePlug == BatteryManager.BATTERY_PLUGGED_AC;
```

Detect the exact level of the battery

420



Exact level

```
int level = batteryStatus
    .getIntExtra(BatteryManager.EXTRA_LEVEL, -1);
int scale = batteryStatus
    .getIntExtra(BatteryManager.EXTRA_SCALE, -1);
float batteryPct = level / (float)scale;
```



Modify AndroidManifest.xml

```
<receiver android:name=".PowerConnectionReceiver">
  <intent-filter>
    <action android:name=
      "android.intent.action.ACTION_POWER_CONNECTED" />
    <action android:name=
      "android.intent.action.ACTION_POWER_DISCONNECTED" />
  </intent-filter>
</receiver>
```

Detect only important variations

421

Modify AndroidManifest.xml

```
<intent-filter>  
  <action android:name=  
    "android.intent.action.ACTION_BATTERY_LOW" />  
  <action android:name=  
    "android.intent.action.ACTION_BATTERY_OKAY" />  
</intent-filter>
```

 ACTION_BATTERY_LOW

▶ the battery is low, you should stop energy consumer activities

 ACTION_BATTERY_OKAY

▶ the battery is OK

Detecting Docks



You can trigger when the device is docked

- Car, Desk, etc. : to perform specific actions like swapping networks, silent mode, etc

```
IntentFilter ifilter =
    new IntentFilter(Intent.ACTION_DOCK_EVENT);
Intent dockStatus = context.registerReceiver(null, ifilter);
int dockState = battery.getIntExtra(EXTRA_DOCK_STATE, -1);

boolean isDocked =
    dockState != Intent.EXTRA_DOCK_STATE_UNDOCKED;
boolean isCar = dockState == EXTRA_DOCK_STATE_CAR;
boolean isDesk = dockState == EXTRA_DOCK_STATE_DESK ||
    dockState == EXTRA_DOCK_STATE_LE_DESK ||
    dockState == EXTRA_DOCK_STATE_HE_DESK;
```

Summary



Managing the battery is really important

- to have responsible behavior
- to modify the behavior of the application



Trace also connectivity to optimize battery

- CONNECTIVITY_CHANGE: helps to detect such situations
- Can be combined with connectivity manager



Some receiver can be deactivated runtime

- using PackageManager
- to avoid extra energy consumptions



Networks

Renault@lrde.epita.fr



Goals

427



Nowadays, most applications require an access to the internet

- to advertise
- to update data
- to publish on social network
- ...



Android provides frameworks for that

- Classical HTTP clients
- Connectivity Manager
- Volley Framework
- ...

Network Connection

428

Modify AndroidManifest.xml

```
<uses-permission  
    android:name="android.permission.INTERNET" />  
<uses-permission  
    android:name="android.permission.ACCESS_NETWORK_STATE" />
```

Choose an HTTP Client

 HttpURLConnection:



- ▶ Most used component
- ▶ Compress Cache
- ▶ Post-Froyo applications

 HttpClient:

- ▶ Before Froyo and Eclair
- ▶ Less configurable

Test Connectivity

ConnectivityManager

-  Check all possible sources (GPS, Wifi, ...)
-  Send Intents when connectivity changes

```
ConnectivityManager connMgr = (ConnectivityManager)
    getSystemService(Context.CONNECTIVITY_SERVICE);
NetworkInfo networkInfo = connMgr.getActiveNetworkInfo();
if (networkInfo != null && networkInfo.isConnected()) {
    // Network available ...
} else {
    // Network not available ...
}
```

 isConnected:

▶ **check if a connection exists**

 getActiveNetworkInfo:

▶ **informations about existing networks**

How to download a Webpage? (1/2)

430

Use asynchronous task

```
private class DownloadWebpageTask
    extends AsyncTask<String, Void, String> {
    @Override
    protected String doInBackground(String... urls) {
        // params comes from the execute() call:
        // params[0] is the url.
        try {
            return downloadUrl(urls[0]);
        } catch (IOException e) {
            return "Unable to retrieve URL (maybe invalid).";
        }
    }

    // onPostExecute displays the results of the AsyncTask.
    @Override
    protected void onPostExecute(String result) {
    }
```

How to download a Webpage? (2/2)

431

```
private String downloadUrl(String myurl) throws IOException {
    InputStream is = null;
    // Only display the first 500 characters of the retrieved web page content.
    int len = 500;
    try {
        URL url = new URL(myurl);
        HttpURLConnection conn = (HttpURLConnection) url.openConnection();
        conn.setReadTimeout(10000 /* milliseconds */);
        conn.setConnectTimeout(15000 /* milliseconds */);
        conn.setRequestMethod("GET");
        conn.setDoInput(true);
        // Starts the query
        conn.connect();
        int response = conn.getResponseCode();
        is = conn.getInputStream();
        // Convert the InputStream into a string
        String contentAsString = readIt(is, len);
        return contentAsString;
    }
    // Makes sure that the InputStream is closed after the app is
    // finished using it.
    finally {
        if (is != null)
            is.close();
    }
}
```

Volley Framework



Google project dedicated to Android

- Automatic request scheduling
- Simultaneous communications
- Caches
- Request's priority
- Easy-configuration



Adapted for applications

- with a massive use of RPC
- using massively structured datas



Ready-to-use component

- in graddle: 'compile com.mcxiaoke.volley:library:1.0.6'

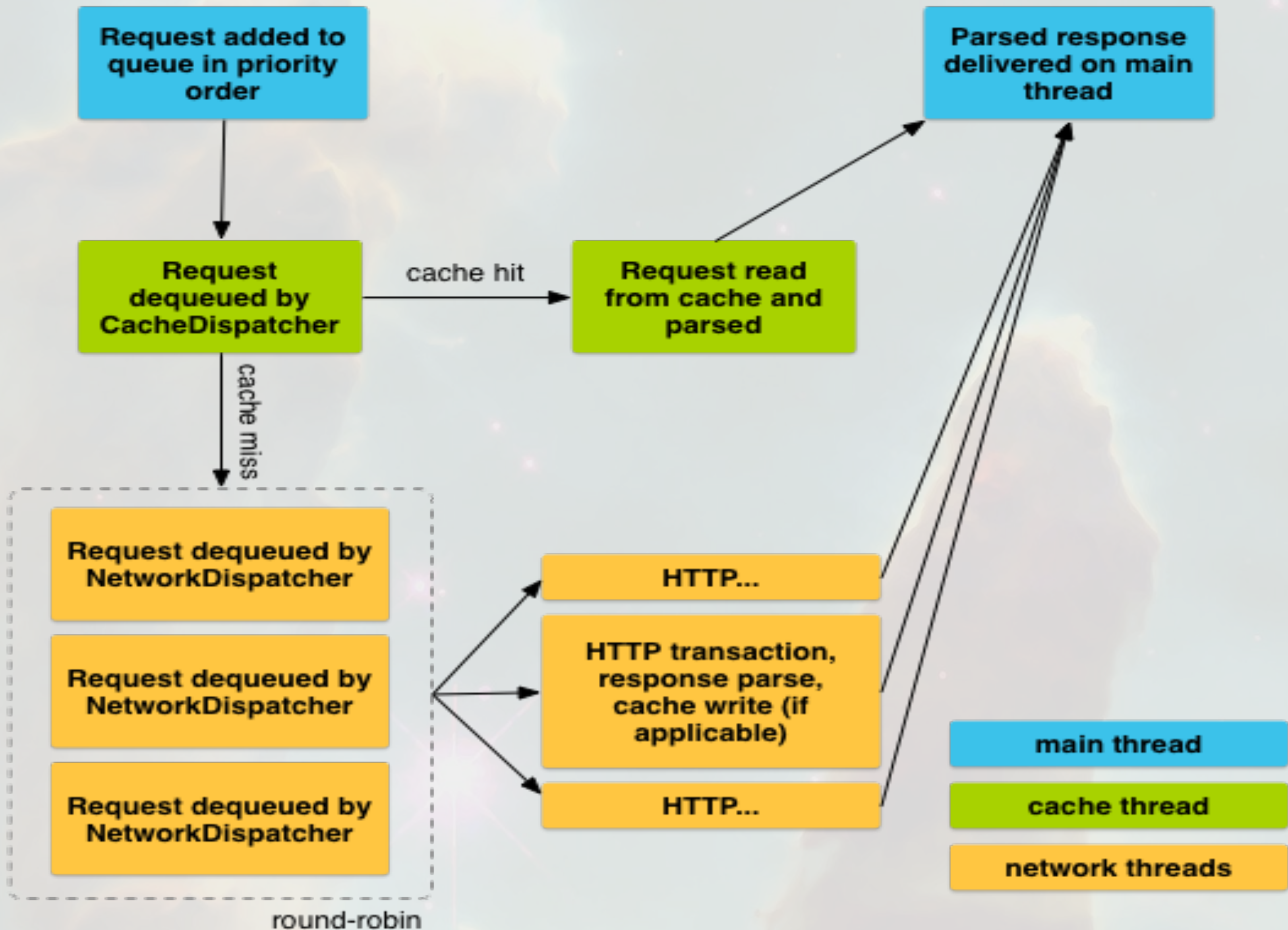
Send/Receive Request

```
RequestQueue queue = Volley.newRequestQueue(this);
String url = "http://www.google.com";

// Request a string response from the provided URL.
StringRequest stringRequest =
    new StringRequest(Request.Method.GET, url,
        new Response.Listener() {
            @Override
            public void onResponse(Object response) {
                // the response without using AsyncTask!
            }
        }, new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {
                // Error!
            }
        });

// Add the request to the RequestQueue.
queue.add(stringRequest);
```

Request Lifecycle



Summary

435



Many ways to connect to network

- 🎧 Classical HTTP clients
- 🎧 Modern Frameworks



Do not have long computing in the UI Thread

- 🎧 Otherwise the application will crash



Think to check connectivity before sending a request



Manage connectivity all around you application



Sensors

Renault@lrde.epita.fr





Sensors

- are device dependent
- produce raw data



Many sensors may be available on devices

- Based on user's movements
 - ▶ **Accelerometer, Gyro, rotation sensors,**
- Based on the environment
 - ▶ **Temperature, barometer, ...**
- Based on positions
 - ▶ **Magnetometers, GPS, ...**

Sensor Framework

SensorManager

Helps to calibrate and detect constants

Sensor

An instance of a given sensor

SensorEvent

Information stored after an event on the sensor

SensorEventListener





Listener on the sensor

Detecting Device Sensors

List available sensors

```
SensorManager mSensorManager =  
    (SensorManager) getSystemService(Context.SENSOR_SERVICE);  
List<Sensor> deviceSensors =  
    mSensorManager.getSensorList(Sensor.TYPE_ALL);  
for (Sensor s: deviceSensors)  
    Toast.makeText(getApplicationContext(), s.toString(),  
        Toast.LENGTH_SHORT).show();
```

Types of Sensors

-  Sensor.TYPE_GYROSCOPE
-  Sensor.TYPE_LINEAR_ACCELERATION
-  Sensor.TYPE_GRAVITY
-  Sensor.TYPE_ALL

Using Light Sensor (1/2)

```
public class MainActivity extends ActionBarActivity
    implements SensorEventListener {
    private SensorManager mSensorManager;
    private Sensor mLight;

    @Override
    public final void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mSensorManager =
            (SensorManager) getSystemService(Context.SENSOR_SERVICE);
        mLight = mSensorManager.getDefaultSensor(Sensor.TYPE_LIGHT);
    }

    @Override
    public final void onAccuracyChanged(Sensor sensor,
        int accuracy) {
        // Do something here if sensor accuracy changes.
    }
}
```

Using Light Sensor (2/2)

```
@Override
public final void onSensorChanged(SensorEvent event) {
    // The light sensor returns a single value.
    // Many sensors return 3 values, one for each axis.
    float lux = event.values[0];
    TextView v = (TextView) findViewById(R.id.display);
    v.setText("LUX : " + lux );
}

@Override
protected void onResume() {
    super.onResume();
    mSensorManager.registerListener(this, mLight,
                                    SensorManager.SENSOR_DELAY_NORMAL);
}

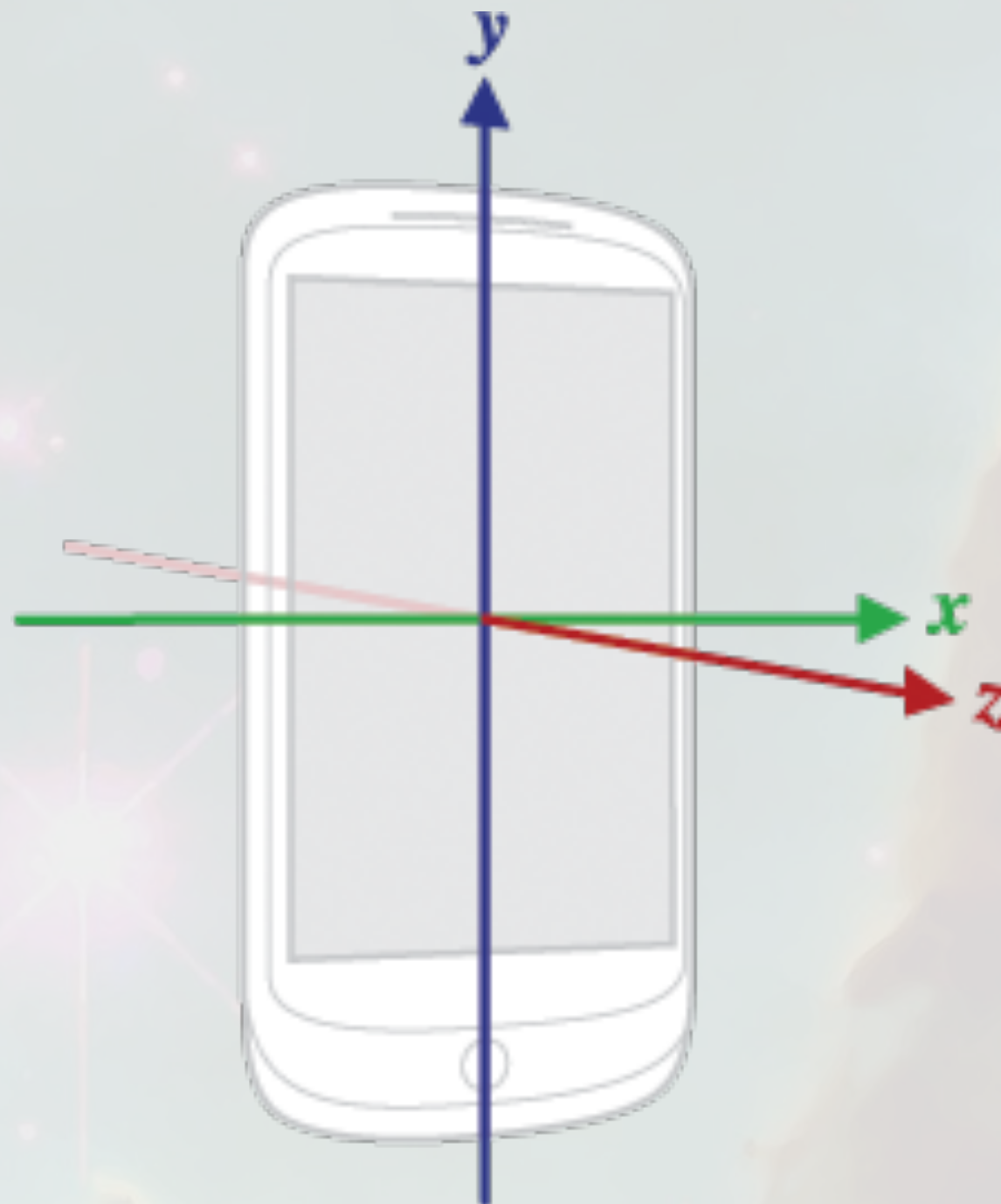
@Override
protected void onPause() {
    super.onPause();
    mSensorManager.unregisterListener(this);
}
```

Motion Sensor



Works similarly

- Register to `SensorEventListener` in `onResume`
- Unregister in `onStop`
- Register for accelerometer event (for instance)
- `onSensorChange` provides informations
 - ▶ `x` in `event.values[0]`
 - ▶ `y` in `event.values[1]`
 - ▶ `z` in `event.values[2]`



Runtime Detection of Sensors

445

 Sensors can be plugged and unplugged anytime

 Sensors may be broken!

 Detect sensors at runtime

```
private SensorManager mSensorManager;
...
mSensorManager =
    (SensorManager) getSystemService(Context.SENSOR_SERVICE);
if (mSensorManager.getDefaultSensor(Sensor.TYPE_PRESSURE)
    != null) {
    // Success! There's a pressure sensor.
}
else {
    // Failure! No pressure sensor.
}
```

Summary



Do not forget to declare in AndroidManifest.xml

```
<uses-feature android:name=  
    "android.hardware.sensor.accelerometer"  
    android:required="true"  
>
```



Multiple sensors are available on devices BUT

-  Choose low refreshing rate to be friendly with battery



All sensors use the same interface

-  one can register to all sensors



Peer2Peer

Renault@lrde.epita.fr



Main Goals



Use local network to exchange information between multiple devices

- 📱 Chats
- 📱 Content Sharing Applications
- 📱 LAN Games

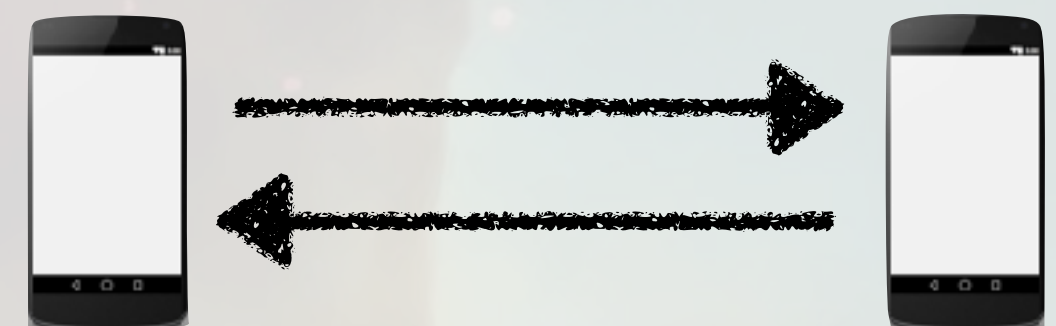


How does it work?

- 📱 A device declares a service on the local network
- 📱 Look for devices connected to the network
- 📱 Peer-2-Peer connection



In this kind of connection devices are services and clients



Connection to local network



Multiple local networks



NFC (Near Field Communication)

- ▶ **Small distance 10 cm**
- ▶ **Contactless payment**



Bluetooth

- ▶ **Medium distance 10 m**
- ▶ **Connection with wearable**
- ▶ **Low speed**



Wifi

- ▶ **Large distance**
- ▶ **High speed**
- ▶ **Internet**



**Here we focus on
P2P over Wifi**

WifiP2PManager



Methods to interact with the Wifi device and to detect and connect with other peer

initialize()

Register a service on a Wifi network

connect()

Start a P2P connection with another device

discoverPeers()

Lookup for other peers

createGroup()


Construct a group with the current device as the group owner

removeGroup()

Remove the current group

How to declare a network service?

Modify AndroidManifest.xml

 ACCESS_WIFI_STATE

▶ access to wifi informations

 CHANGE_WIFI_STATE

▶ modify Wifi connectivity for the device

 INTERNET

▶ allows the application to open sockets

```
<uses-permission
  android:required="true"
  android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission
  android:required="true"
  android:name="android.permission.CHANGE_WIFI_STATE" />
<uses-permission
  android:required="true"
  android:name="android.permission.INTERNET" />
```

Registering a Service

```
int SERVER_PORT = 9000;
WifiP2pManager mManager;
WifiP2pManager.Channel channel;
private void startRegistration() {
    Map record = new HashMap();
    record.put("listenport", String.valueOf(SERVER_PORT));
    record.put("buddyname", "John doe (on "
        + android.os.Build.MODEL + ")");
    record.put("available", "visible");
    WifiP2pDnsSdServiceInfo serviceInfo = WifiP2pDnsSdServiceInfo
        .newInstance("_test", "_presence_tcp", record);
    mManager.addLocalService(channel, serviceInfo,
        new WifiP2pManager.ActionListener() {
            @Override
            public void onSuccess() {
            }
            @Override
            public void onFailure(int arg0) {
            }
        });
}
```

Detecting existing service (1/3)

```
final HashMap<String, String> buddies =
    new HashMap<String, String>();
private void discoverService() {
    WifiP2pManager.DnsSdTxtRecordListener txtListener =
        new WifiP2pManager.DnsSdTxtRecordListener() {
        @Override
        /* Callback includes:
        * fullDomain: full domain name:
        *           e.g "tt._ipp._tcp.local."
        * record: TXT record data as a map of key/value pairs.
        * device: The device running the advertised service.
        */
        public void onDnsSdTxtRecordAvailable(
            String fullDomain, Map record,
            WifiP2pDevice device) {
            buddies.put(device.deviceAddress,
                (String) record.get("buddyname") +
                "--" + (String) record.get("available"));
        }
    };
};
```

Detecting existing service (2/3)

```
WifiP2pManager.DnsSdServiceResponseListener servListener =
    new WifiP2pManager.DnsSdServiceResponseListener() {
        @Override
        public void onDnsSdServiceAvailable(String instanceName,
            String registrationType,
            WifiP2pDevice
resourceType) {

            // Update the device name with the human-friendly version
            from
            // the DnsTxtRecord, assuming one arrived.
            resourceType.deviceName = buddies
                .containsKey(resourceType.deviceAddress) ?
                buddies.get(resourceType.deviceAddress) :
                resourceType.deviceName;
        }
    };

mManager.setDnsSdResponseListeners(channel,
                                    servListener, txtListener);
```


Detecting existing service (3/3)

```
WifiP2pDnsSdServiceRequest serviceRequest =
    WifiP2pDnsSdServiceRequest.newInstance();
mManager.addServiceRequest(channel,
    serviceRequest,
    new WifiP2pManager.ActionListener() {
        @Override
        public void onSuccess() {
            // Success!
        }
        @Override
        public void onFailure(int code) {
            // Command failed.
            // Check for P2P_UNSUPPORTED, ERROR, or BUSY
        }
    });
mManager.discoverServices(channel, new WifiP2pManager.ActionListener() {
    @Override
    public void onSuccess() {
    }
    @Override
    public void onFailure(int reason) {
    }
});
}
```

Summary



We can now discover, create and connect to services



To exchange with a device

 just open a socket and exchange data



Over other network P2P communication works the same way

 Just read the documentation



Think to remove your service when you leave the Wifi or the application



Advertising

Renault@lrde.epita.fr



Advertising

462



What for?

- Most of Android Applications are free
- advertising is one way to have money



Paid Applications can also embed advertising

- 15% to 20% more gains



Advertising must be controlled

- Limit the impact on the advertising
- Control and restrict according to your users



Choose your advertising network

- AdMob, DoubleClickForPublisher, AdWhirl, MobClix, MobFox, MobPub...



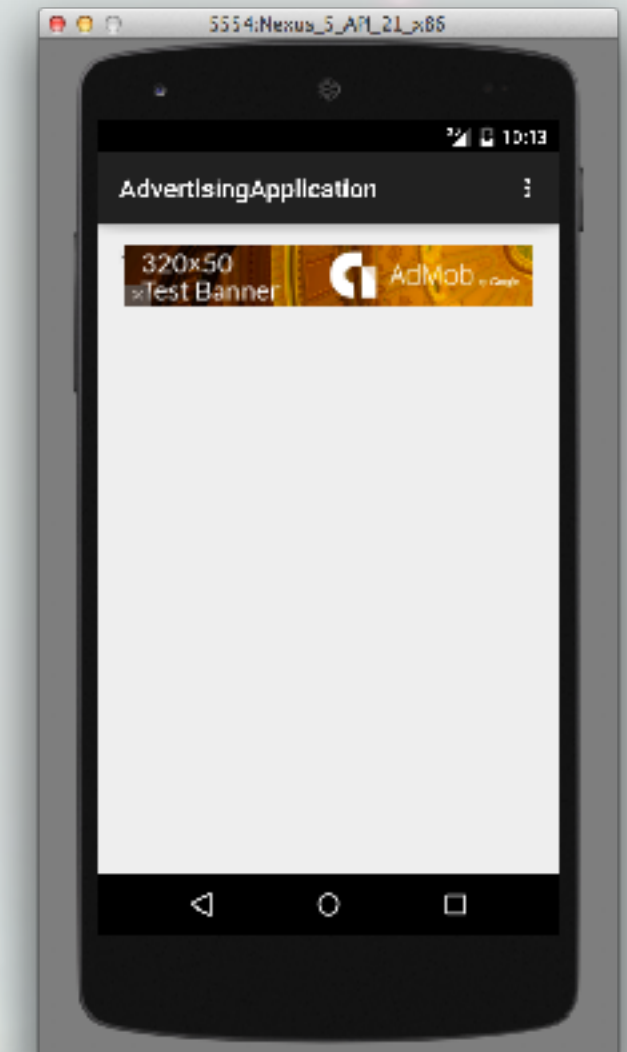
Widely used service on Android

- More than 700 000 application use it
- Reports for the developers
- Easy filtering



Two kind of advertising



- Banners
 - ▶ A banner is displayed on the top of the screen
 - ▶ Small impact on the user experience
- Interstitial
 - ▶ Full screen HTML5
 - ▶ The user choose to close the ad
 - ▶ Usually displayed when swapping activities



Setup (1/2)

464

Register into AdMob

-  Declare then each advertising
-  And get AD_UNIT_ID for each

Modify AndroidManifest.xml

-  Internet permission

```
<uses-permission  
    android:name="android.permission.INTERNET" />  
<uses-permission  
    android:name="android.permission.ACCESS_NETWORK_STATE" />
```

-  Clickable advertising

```
<activity android:name="com.google.android.gms.ads.AdActivity"  
    android:configChanges="keyboard|keyboardHidden|orientation|  
        screenLayout|uiMode|screenSize|smallestScreenSize"  
    android:theme="@android:style/Theme.Translucent" />
```


Setup (2/2)

465

Modify AndroidManifest.xml

 Connect to GooglePlay

```
<meta-data  
    android:name="com.google.android.gms.version"  
    android:value="@integer/google_play_services_version" />
```

Modify build.gradle

```
compile 'com.google.android.gms:play-services:6.5.+'
```

Declaring a Banner



Integrate AD_UNIT_ID into res/values/string.xml

```
<string name="banner_ad_unit_id">  
    ca-app-pub-XXX/YYY  
</string>
```



Modify main layout to add the ad

```
<com.google.android.gms.ads.AdView  
    android:id="@+id/adView"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_centerHorizontal="true"  
    ads:adSize="BANNER"  
    ads:adUnitId="@string/banner_ad_unit_id">  
</com.google.android.gms.ads.AdView>
```

Trigger a Banner (1/2)

```
private AdView mAdView;
private AdRequest adRequest;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    mAdView = (AdView) findViewById(R.id.adView);
    adRequest = new AdRequest.Builder()
        // .addTestDevice(AdRequest.DEVICE_ID_EMULATOR)
        // .tagForChildDirectedTreatment(true)
        // .setGender(AdRequest.GENDER_FEMALE)
        // .setBirthday(new GregorianCalendar(1985, 1, 1).getTime())
        .build();
    mAdView.loadAd(adRequest);
}
/** Called when leaving the activity */
@Override
public void onPause() {
    if (mAdView != null) mAdView.pause();
    super.onPause();
}
```

Trigger a Banner (2/2)

```
/** Called when returning to the activity */  
@Override  
public void onResume() {  
    super.onResume();  
    if (mAdView != null) {  
        mAdView.resume();  
    }  
}
```

```
/** Called before the activity is destroyed */  
@Override  
public void onDestroy() {  
    if (mAdView != null) {  
        mAdView.destroy();  
    }  
    super.onDestroy();  
}
```



Interface `com.google.android.gms.ad.AdListener`



`onAdLoaded`

▶ **triggered when the ad is loaded**



`onAdOpened`

▶ **triggered when the ad is open**



`onAdClosed`

▶ **triggered when the ad is closed**



`onAdLeftApplication`

▶ **triggered when the ad is dismissed**



`onAdFailedToLoad`

▶ **triggered when a problem occurs during the loading of an app**

Interstitial Advertising



Work the same!



Instantiation

```
interstitial = new InterstitialAd(this);
interstitial.setAdUnitId("ca-app-pub-XXX/YYYY");
adRequest = new AdRequest.Builder().build();

// Begin loading your interstitial.
interstitial.loadAd(adRequest);
displayInterstitial();
```



Trigger the Ad

```
// Invoke displayInterstitial() when you are ready
// to display an interstitial.
public void displayInterstitial() {
    if (interstitial.isLoaded()) interstitial.show();
    else Toast.makeText(this, "Ad did not load",
        Toast.LENGTH_SHORT).show();
}
```

Summary

471



Easy to integrate into your application



To note

 Do not overload your application with ads

 Choose your ad network

 Define the kind of ad you want to have

 Log the lifecycle of your activity



Free-to-play game use a lot of ads





