SCIENCES SORBONNE UNIVERSITÉ

Handle Device Orientation

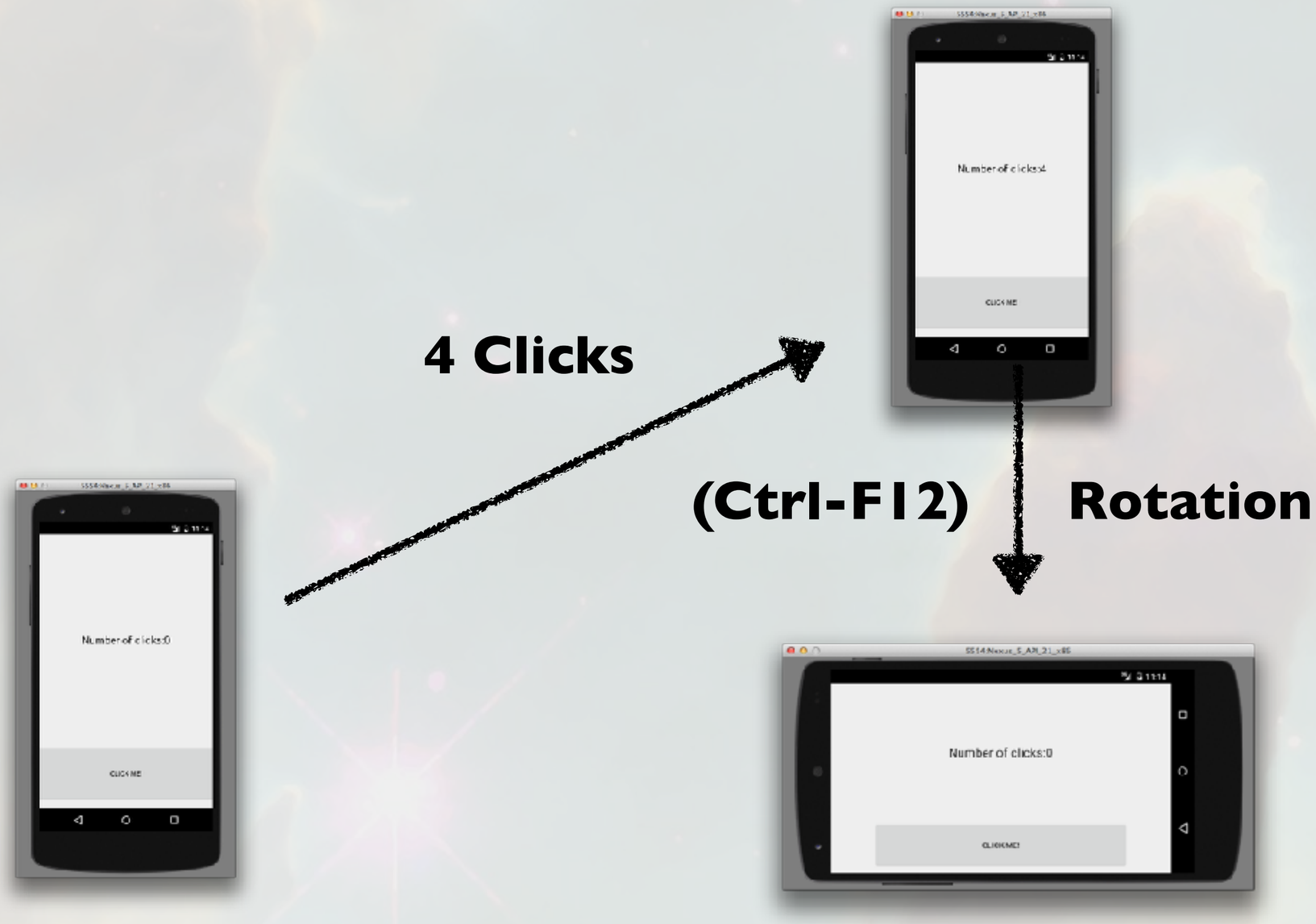
F. Kordon - Sorbonne Université - CC2018

Renault@lrde.epita.fr

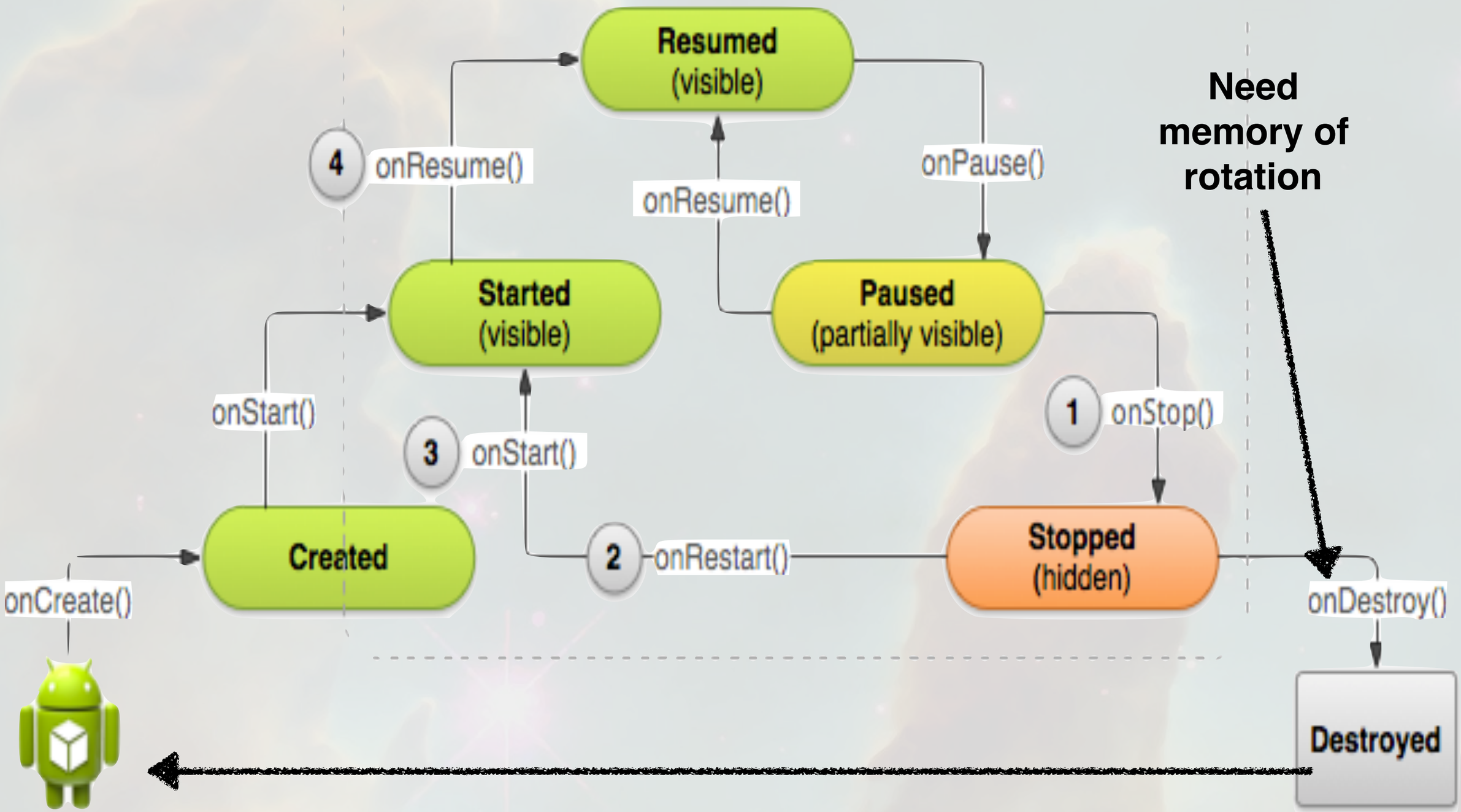
Motivation

Let us consider a simple application

- Only count the number of clicks on a button



The Origins of the Problem



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 overload 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...
   }
}
```

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 overlo anged method

Handling configuration changes does not avoid data loss...

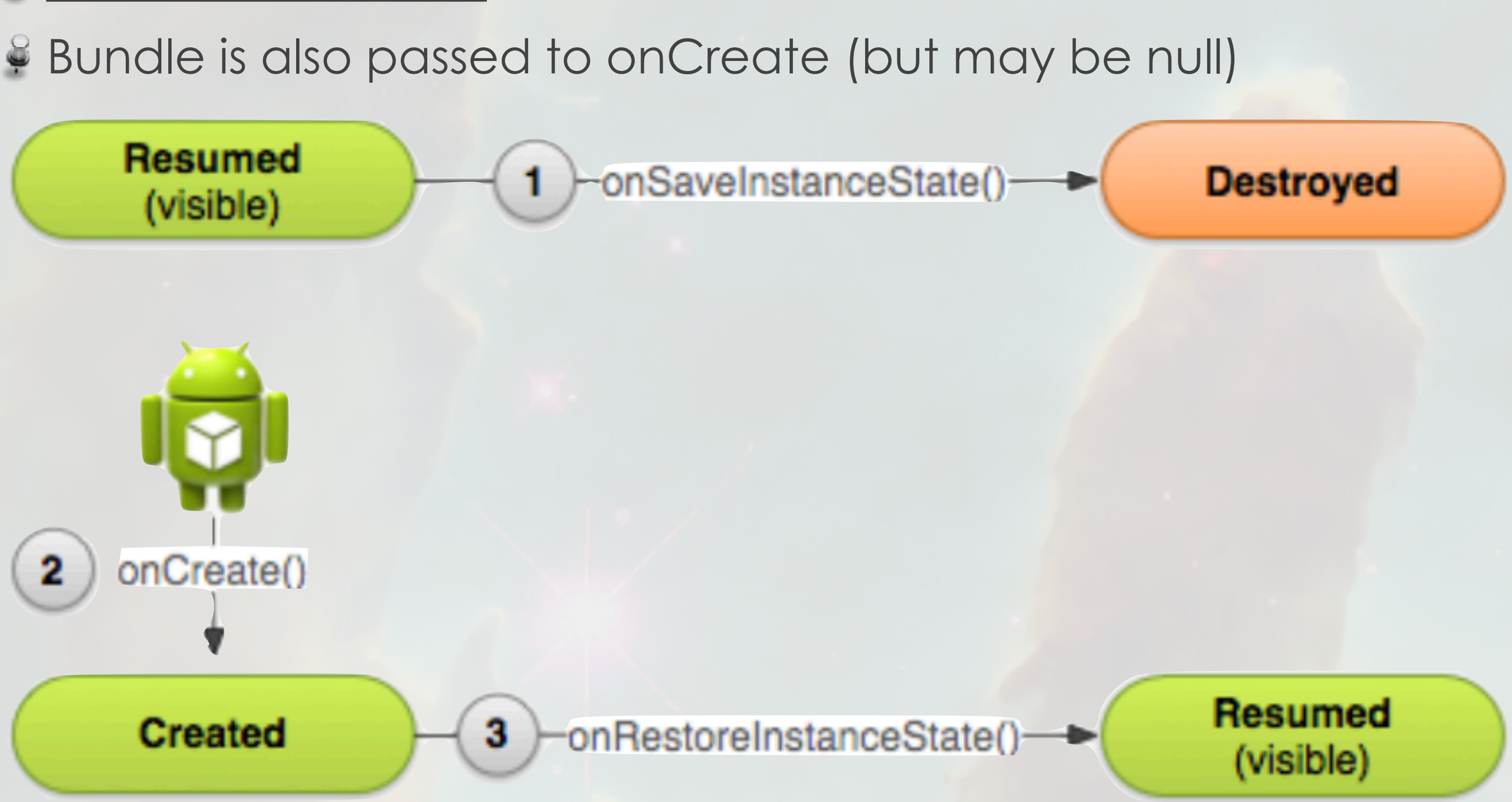
```
public void onConfi newConfig) {
    super.onConfigu

    // Checks the or
    if (newConfig.ori ion.ORIENTATION_LANDSCAPE) {
    // do something...
    }
    else if (newConfig.orientation == Configuration.ORIENTATION_PORTRAIT){
    // do something...
    }
}
```

How to avoid data loss?

Use the Bundle

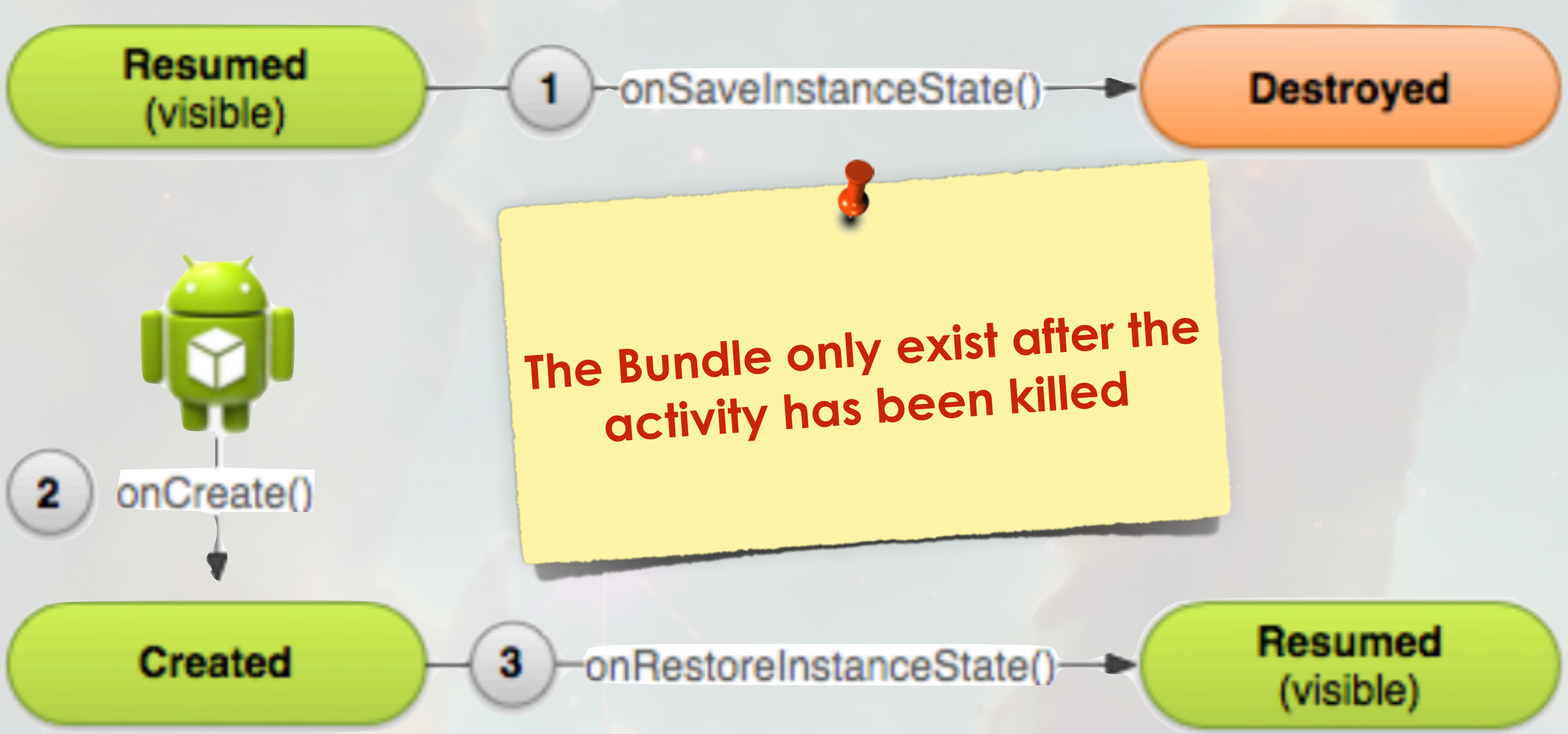
- Associative <key, values> with heterogeneous values
- OnSaveInstance: save data before a rotation
- OnRestoreInstance: restore data after a rotation
- Bundle is also passed to onCreate (but may be null)



How to avoid data loss?

Use the Bundle

- Associative <key, values> with heterogeneous values
- OnSaveInstance: save data before a rotation
- OnRestoreInstance: 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 / putIntegerArrayLit / ...
- getChar / putChat / ...
- getByte / putByte / ...
- ...

Save more complex types

The Parcelable interface helps to serialize objects

- Parcelable object can be stored and retreived 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 relevants
- 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