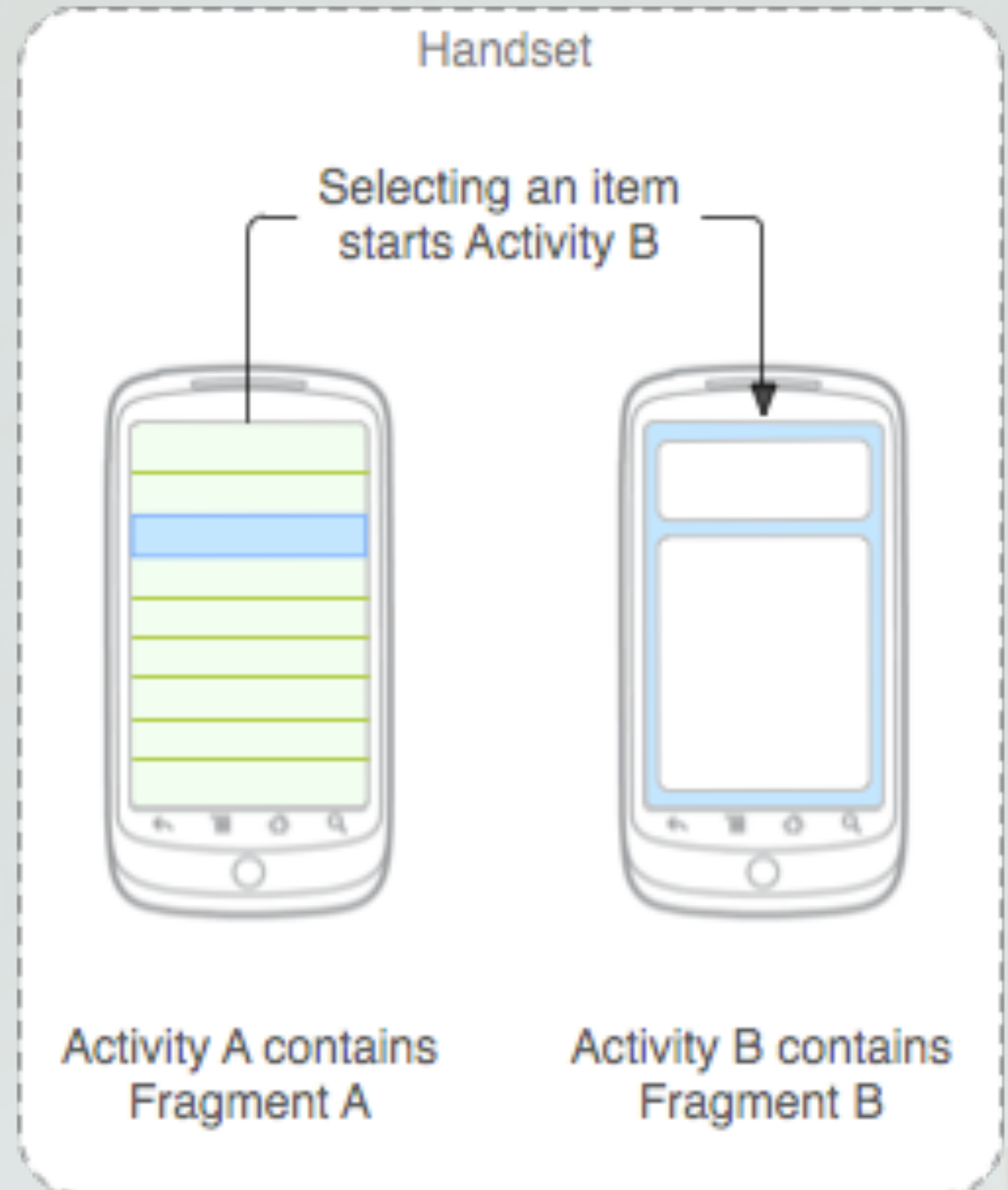
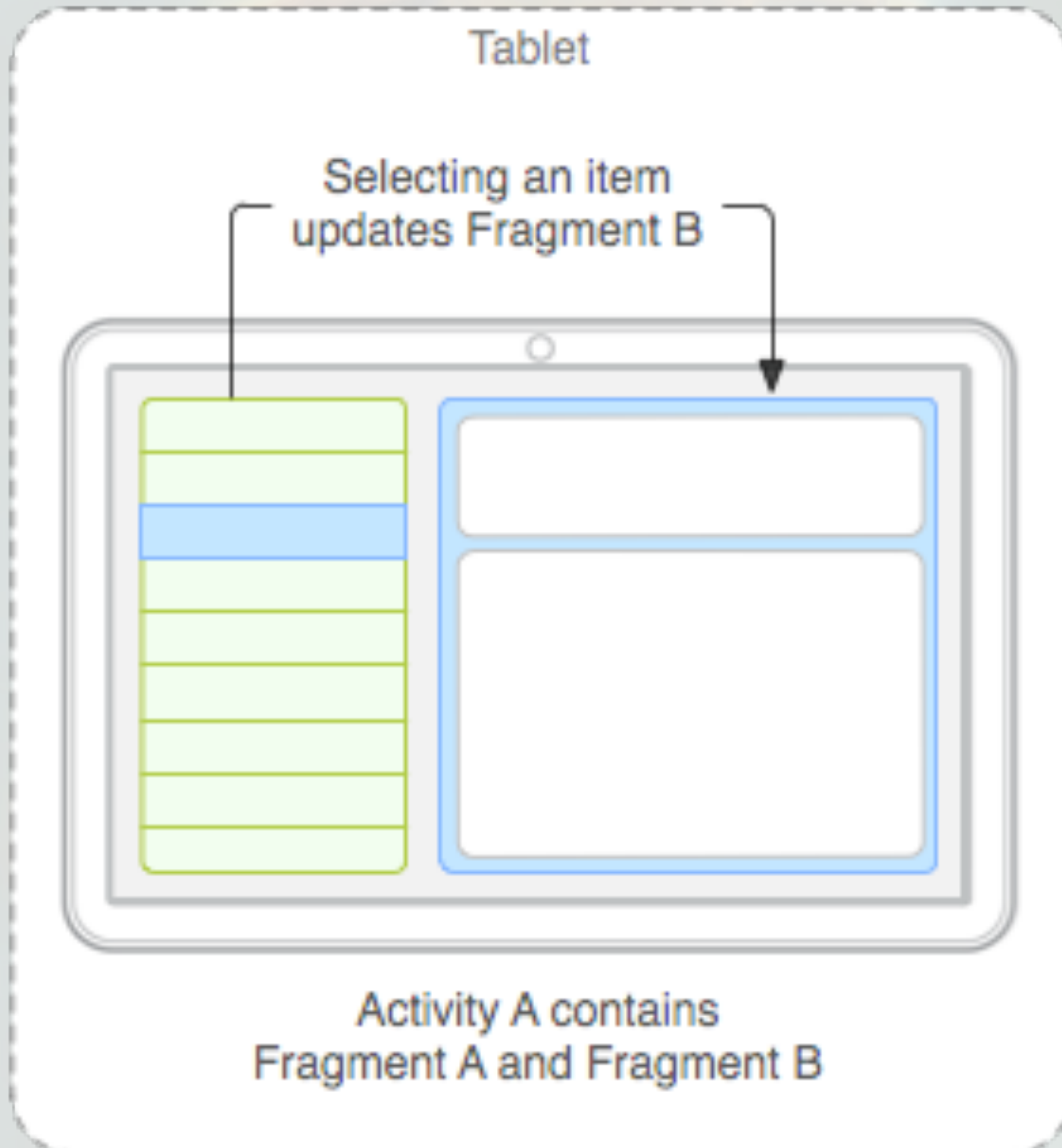


Fragments

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

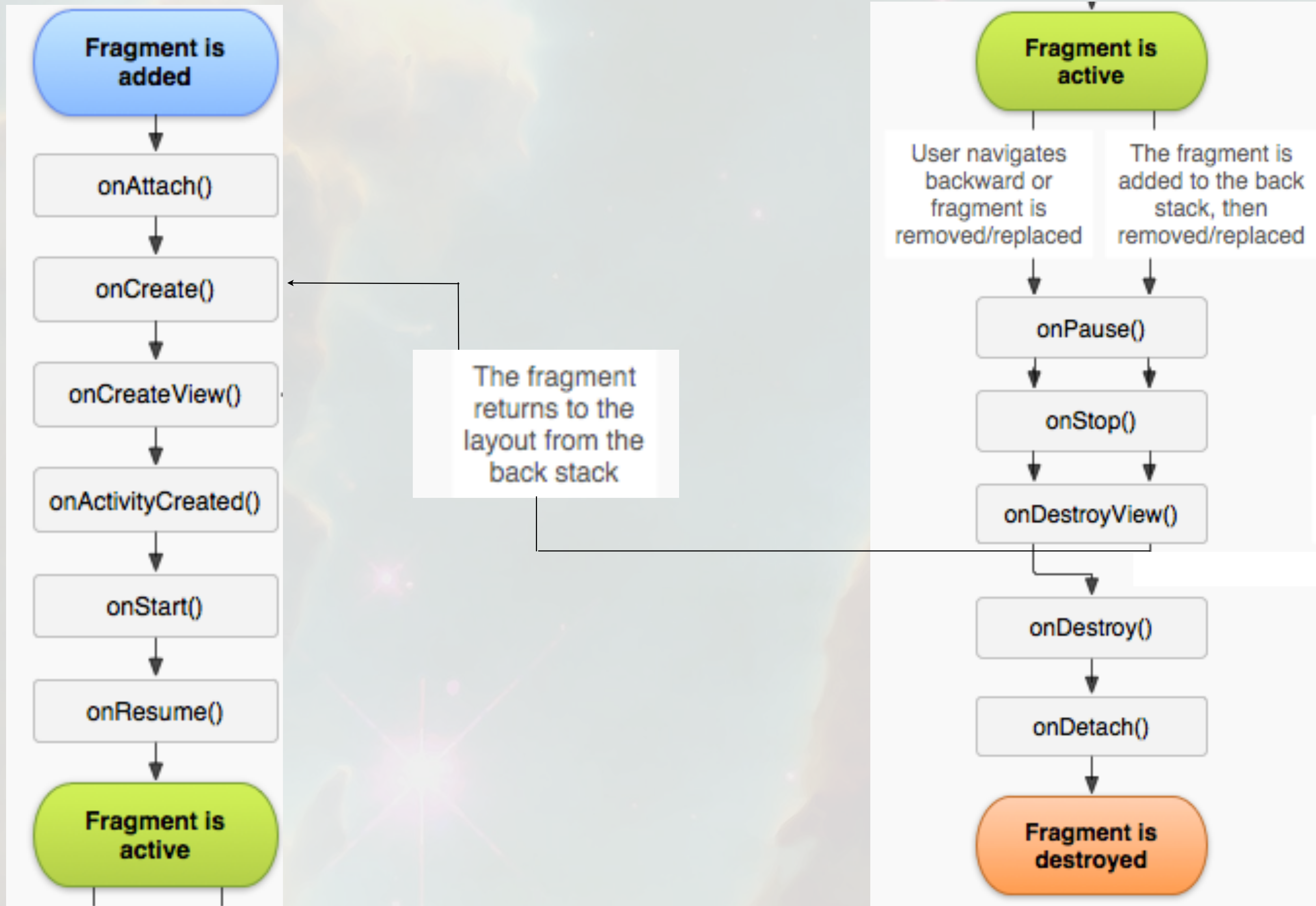
- Use well suited component adapted to the current device / orientation
- Keep the back stack unmodified



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**

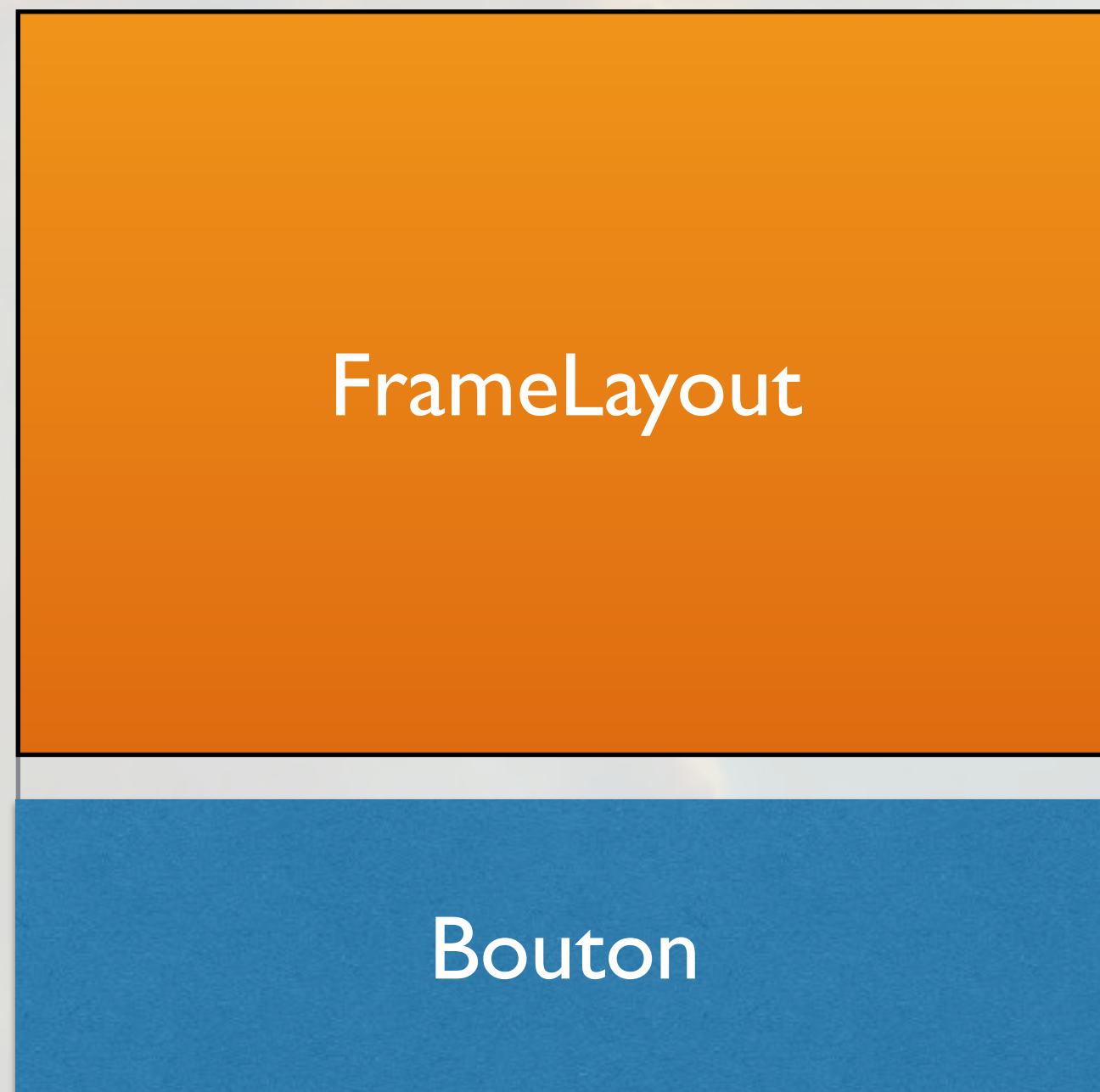
Example details (1/2)



Activity.view

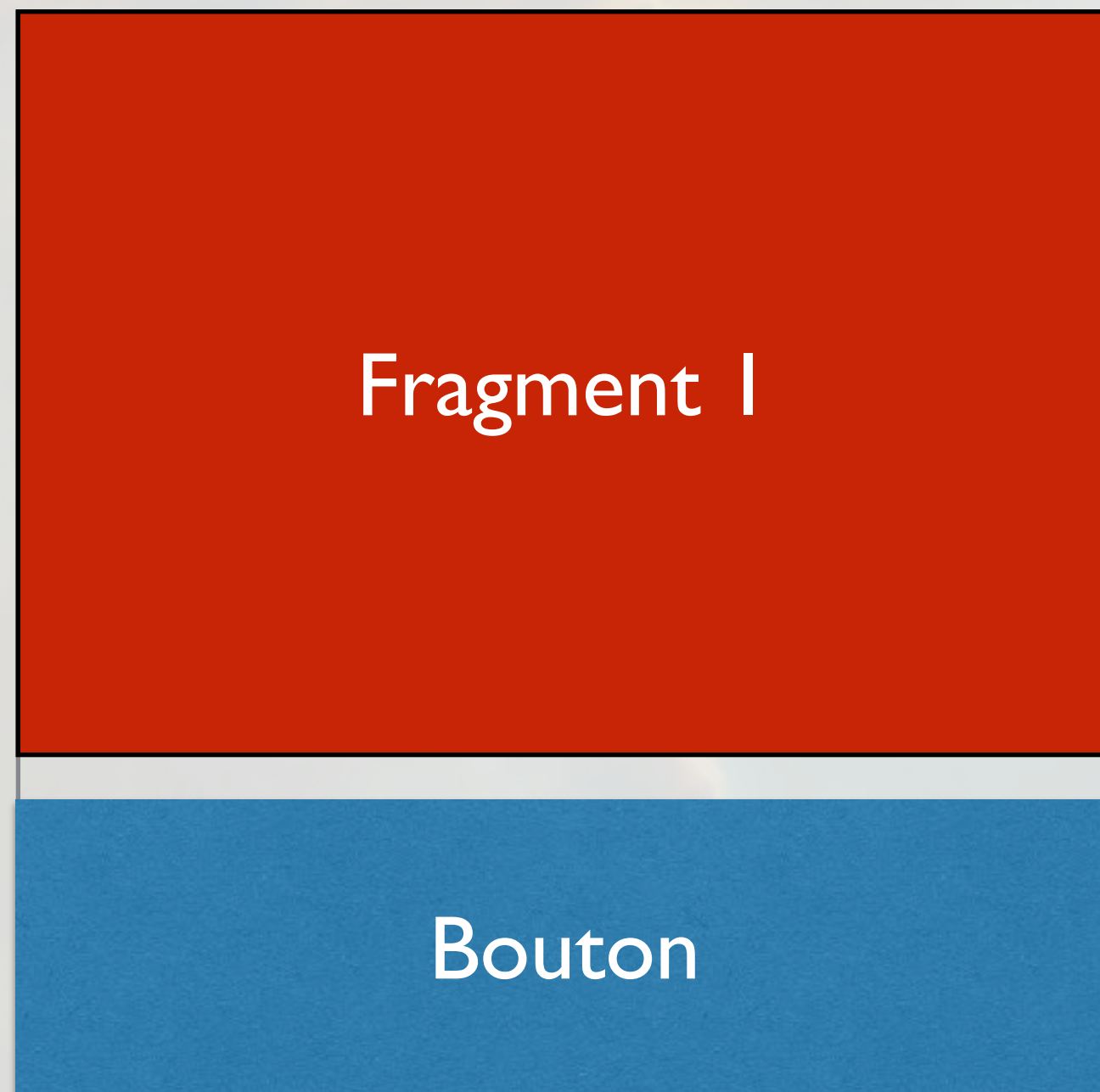
- **Fragments use area to display**
- **Usually we use empty layout for this specific area**

Example details (1/2)



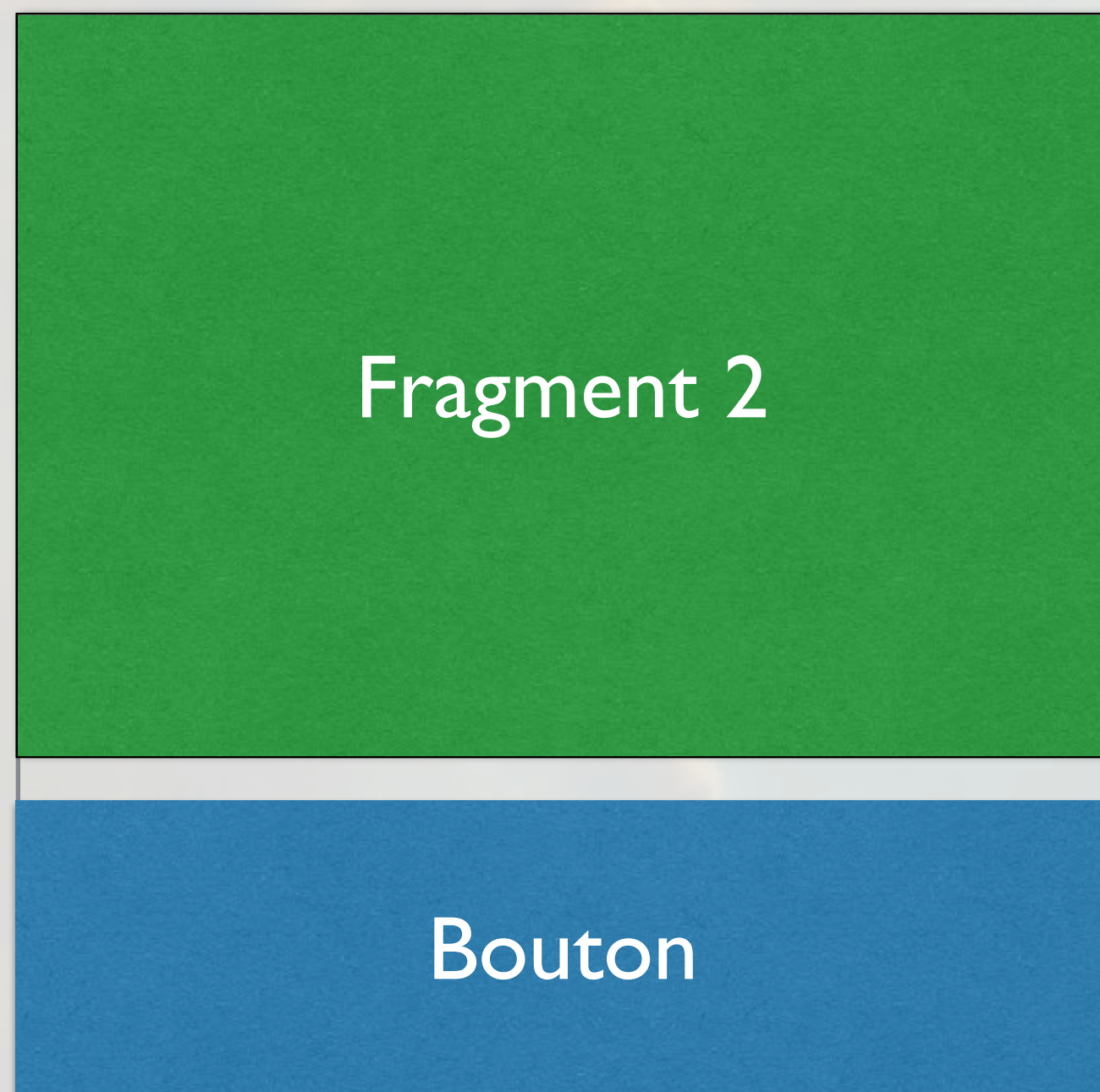
- **Fragments use area to display**
- **Usually we use empty layout for this specific area**

Example details (1/2)



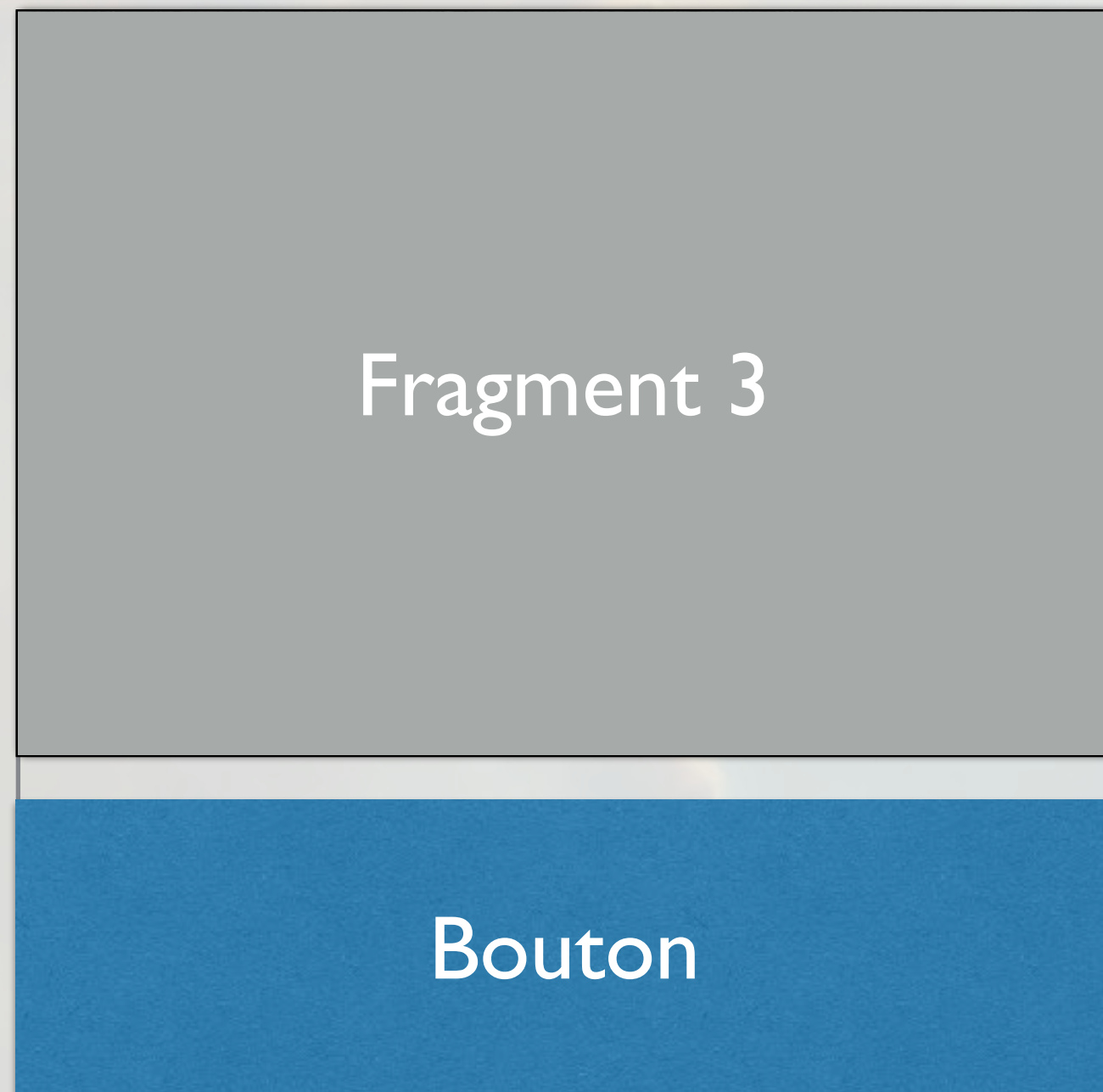
- **Fragments use area to display**
- **Usually we use empty layout for this specific area**

Example details (1/2)



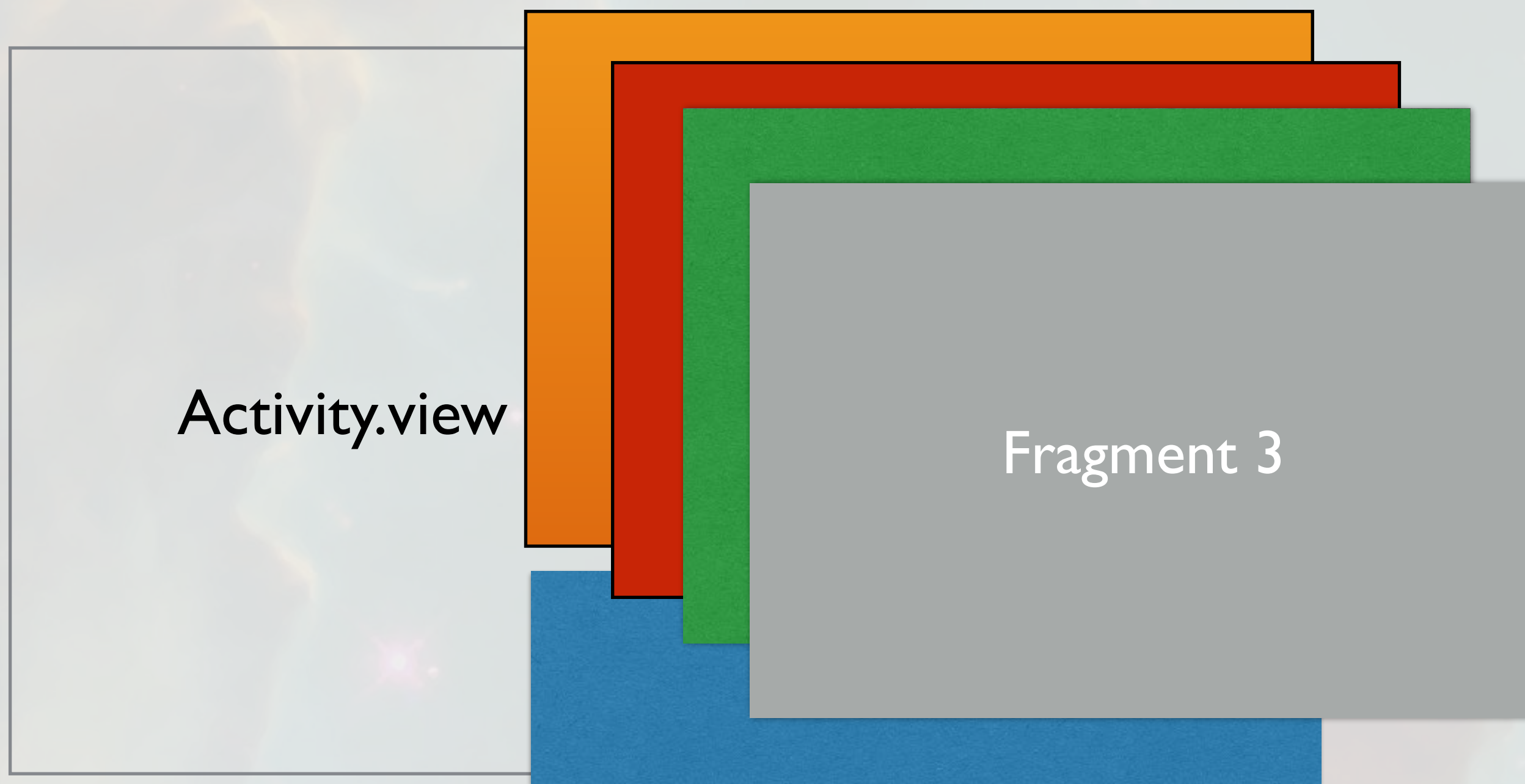
- **Fragments use area to display**
- **Usually we use empty layout for this specific area**

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="-100%"
    android:toXDelta="0%" >
  </translate>
</set>
```

 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>
```

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



