

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

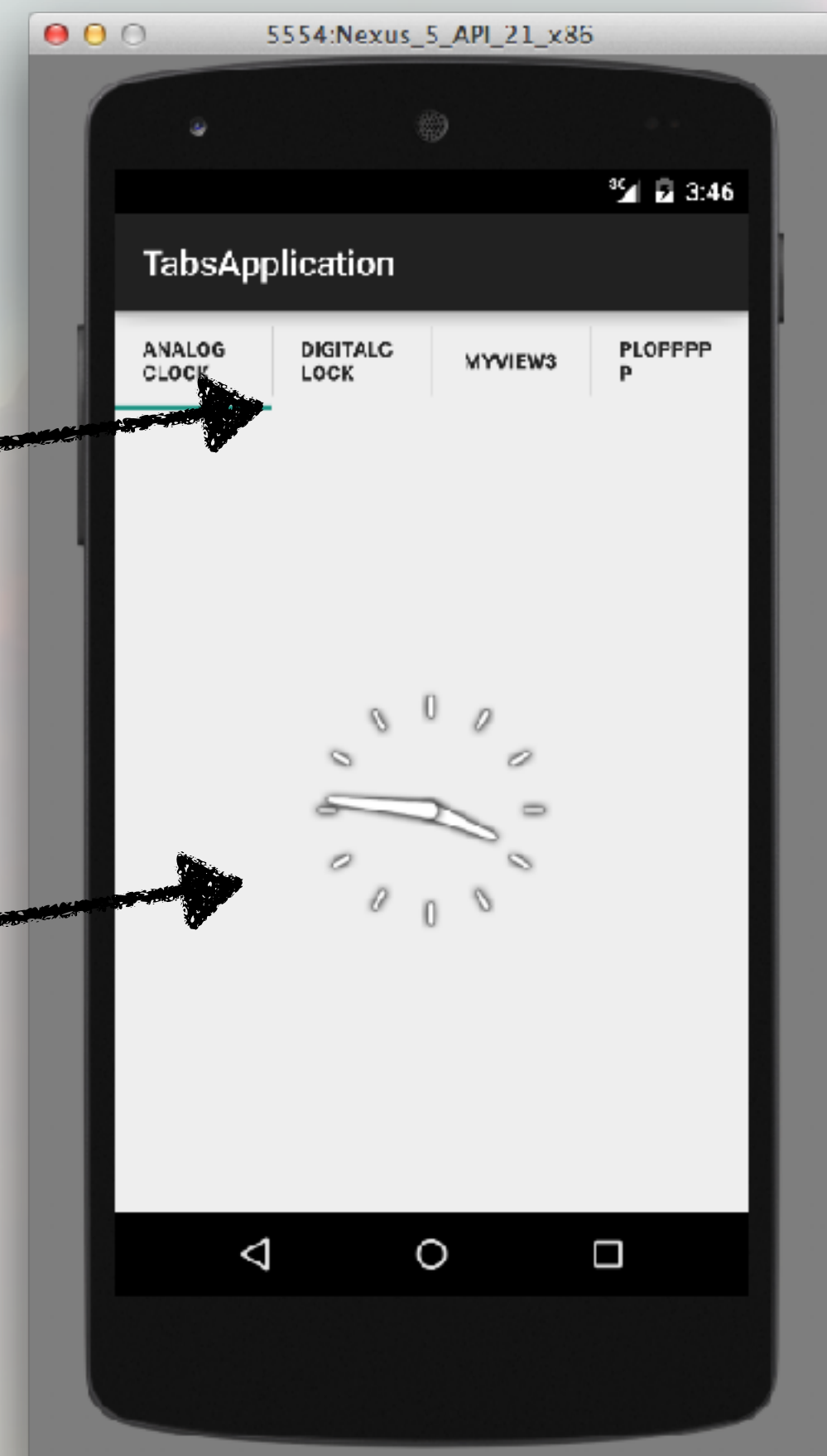


Two main elements

- 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();

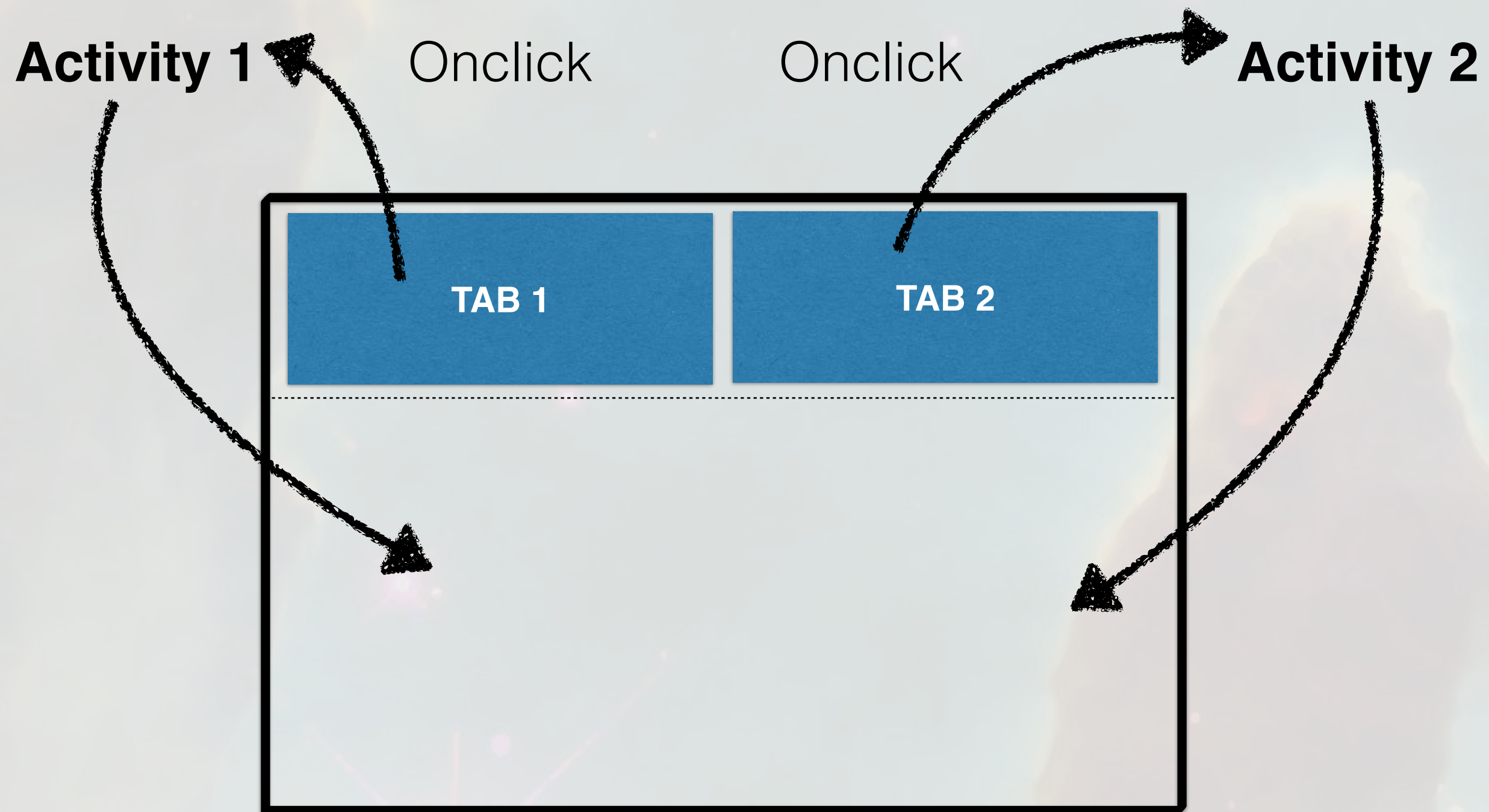
    TabHost.TabSpec spec=tabs.newTabSpec("tag1");
    spec.setContent(R.id.tab1);
    spec.setIndicator("Analog Clock");
    tabs.addTab(spec);

    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);
```

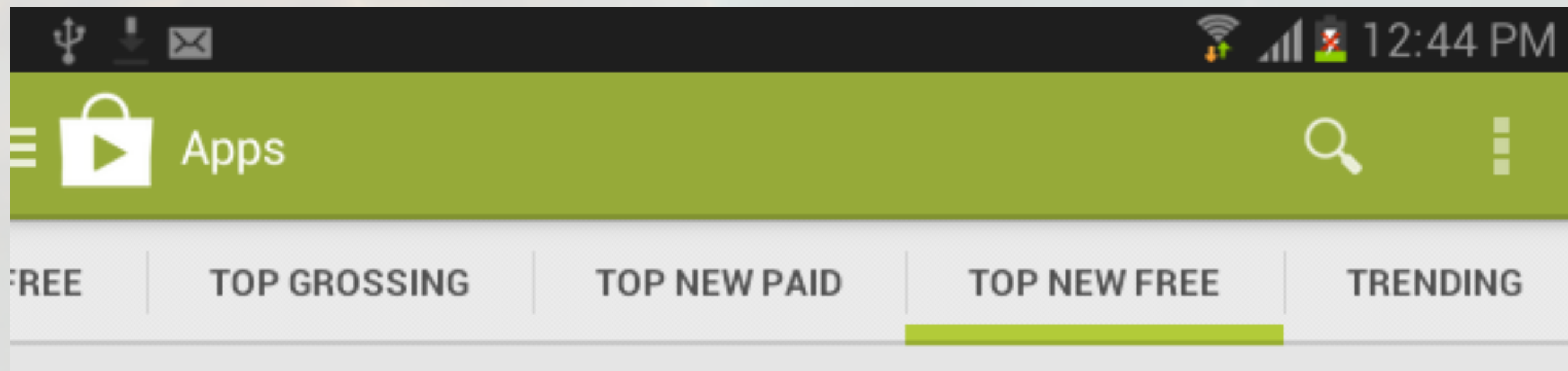
Replace tabs by some view

```
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**



