



JÖNKÖPING UNIVERSITY

School of Engineering

ANDROID FRAGMENTS

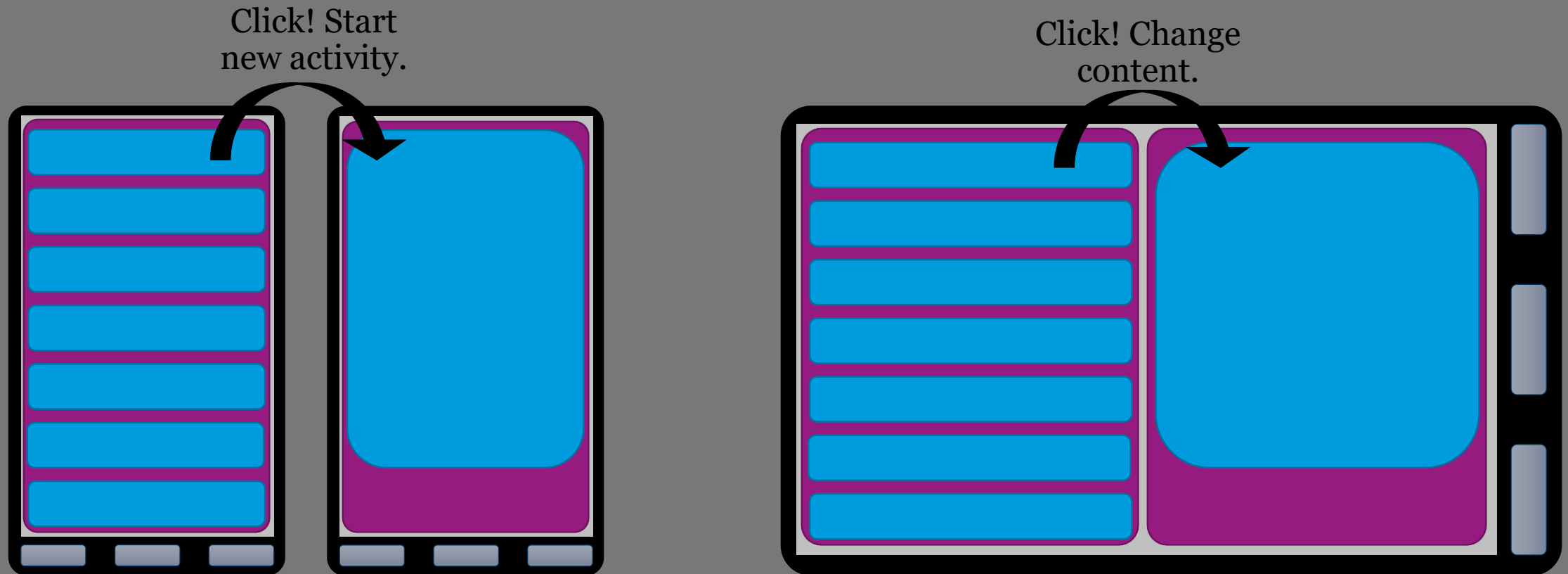
Peter Larsson-Green

Jönköping University

Spring 2020

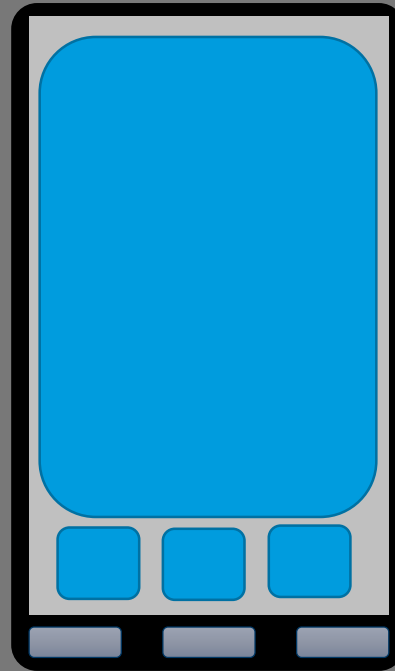
WHY WERE FRAGMENTS ADDED?

To re-use parts of the GUI.



WHY WERE FRAGMENTS ADDED?

To encapsulate a part of the GUI.



CREATING A FRAGMENT

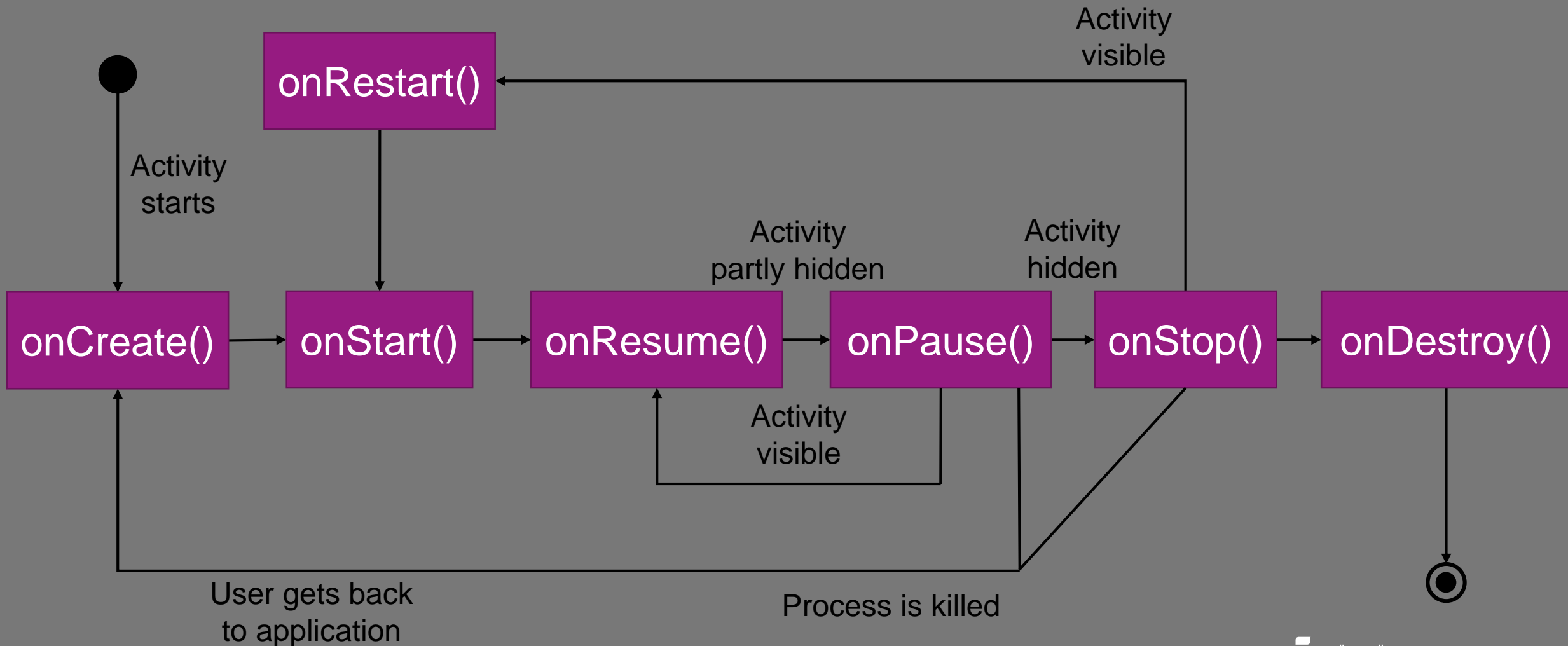
```
public class MyFragment extends Fragment{  
    @Override  
    public View onCreateView(LayoutInflater inflater,  
                             ViewGroup container,  
                             Bundle savedInstanceState) {  
        return inflater.inflate(R.layout.my_layout, container, false);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

INSERTING STATIC FRAGMENTS

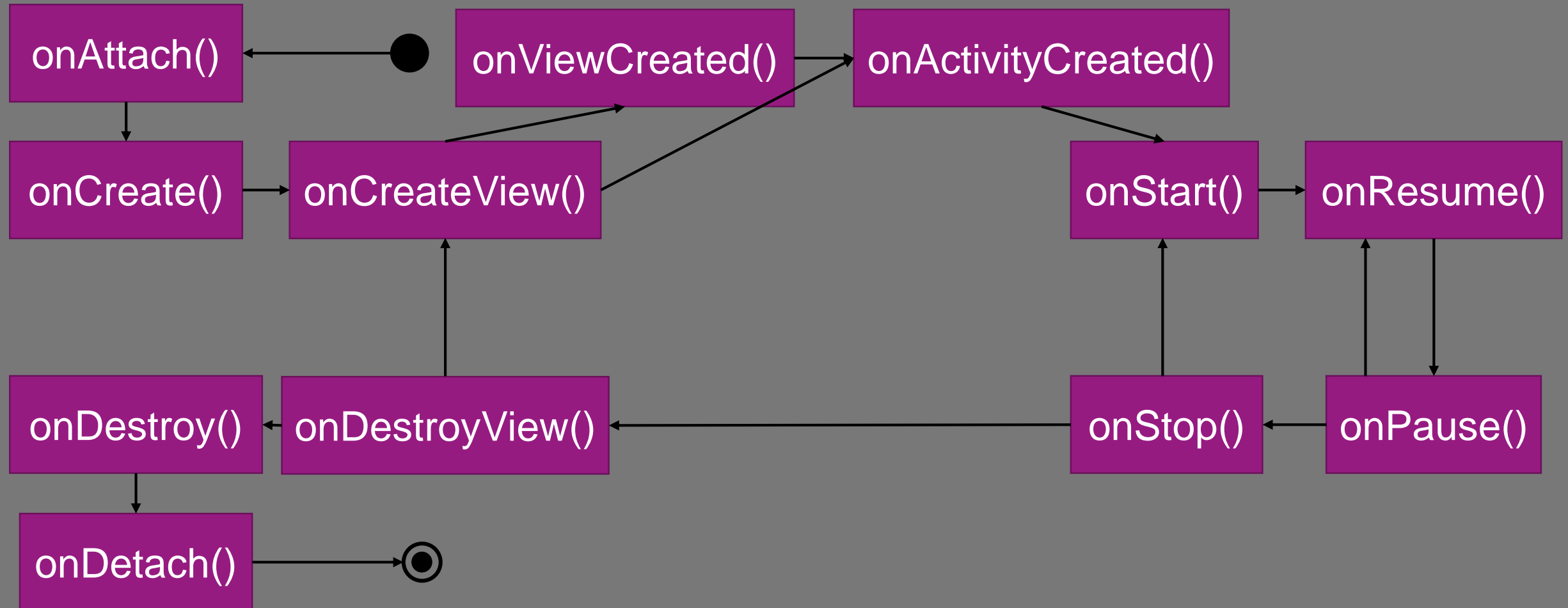
```
<fragment  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:name="your.package.YourFragmentClass" />
```

Can easily insert the fragment in different layouts/activities 😊

AN ACTIVITY'S LIFE CYCLE



A FRAGMENT'S LIFECYCLE



INSERTING DYNAMIC FRAGMENT

```
FragmentManager fm = anActivity.getFragmentManager();  
FragmentTransaction ft = fm.beginTransaction();  
ft.add(R.id.containerViewId, theFragment, "theTag");  
ft.commit();
```

```
Fragment theFragment = fm.findFragmentByTag("theTag");
```

```
ft.remove(theFragment);
```

```
ft.replace(R.id.containerViewId, theNewFragment);
```

```
ft.addToBackStack("theName");
```

CONSTRUCTORS IN FRAGMENTS

```
public class MyFragment extends Fragment{  
    public String name;  
    public MyFragment(String name) {  
        this.name = name;  
    }  
}
```

```
MyFragment fragment = new MyFragment("Alan Turing");
```

Runtime configuration change?

- Android will re-create the fragment as needed.
- The default constructor will be used.

USE ARGUMENTS INSTEAD

When creating the fragment:

```
Bundle args = new Bundle();  
args.putString("name", "Alan Turing")  
MyFragment fragment = new MyFragment();  
fragment.setArguments(args);
```

Inside the fragment:

```
String name = getArguments().getString("name");
```

FRAGMENT CREATION PATTERN

```
public class MyFragment extends Fragment{

    public static MyFragment newInstance(String name) {
        Bundle args = new Bundle();
        args.putString("name", name);
        MyFragment fragment = new MyFragment();
        fragment.setArguments(args);
        return fragment;
    }
}
```

```
MyFragment fragment = MyFragment.newInstance("Alan Turing");
```

FRAGMENT CREATION PATTERN

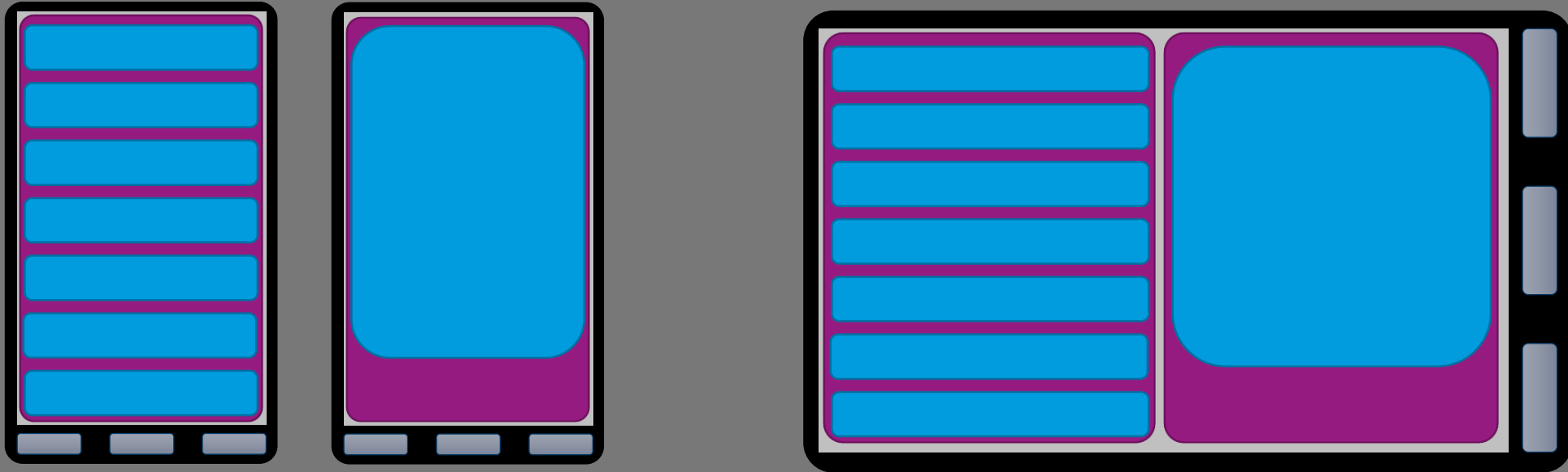
```
public class MyFragment extends Fragment{  
    private static final String NAME = "name";  
    public static MyFragment newInstance(String name) {  
        Bundle args = new Bundle();  
        args.putString(NAME, name);  
        MyFragment fragment = new MyFragment();  
        fragment.setArguments(args);  
        return fragment;  
    }  
}
```

```
MyFragment fragment = MyFragment.newInstance("Alan Turing");
```

FRAGMENT COMMUNICATION

Let the fragment notify the activity when something happens in the fragment.

- E.g. click on button in fragment → Call method in activity.



FRAGMENT COMMUNICATION

```
public class MyFragment extends Fragment{  
    // Listen for click on a button in the fragment.  
    public void onClick(View view){  
        // Call a method on the activity.  
        ((Interface) getActivity()).theButtonWasClicked();  
    }  
  
    public interface Interface{  
        void theButtonWasClicked();  
    }  
}
```

```
public class MyActivity extends Activity implements MyFragment.Interface{  
    public void theButtonWasClicked(){ /* Handle it! */ }  
}
```

SURVIVING CONFIG. CHANGES

Fragments can survive configuration changes.

- Called model fragments.
- Does usually not contain any GUI, just data.

```
public class MyModelFragment extends Fragment{  
    public int counter = 0;  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setRetainInstance(true);  
    }  
}
```


SURVIVING CONFIG. CHANGES

```
public class MyActivity extends Activity{
    private MyModelFragment myModelFragment;
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        FragmentManager fm = getSupportFragmentManager();
        myModelFragment = fm.findFragmentByTag("myModelFragment");
        if(myModelFragment == null){
            myModelFragment = new MyModelFragment();
            FragmentTransaction ft = fm.beginTransaction();
            ft.add(myModelFragment, "myModelFragment");
            ft.commit();
        }
    }
}
```