Creating Compound Components

Brendan Wanlass
ANDROID ARCHITECT

@brendanwanlass

Summary

Intro to compound components

Combining views into a compound component

Communicating between view and activity

Custom attributes

Intro to Compound Components

Basic Components

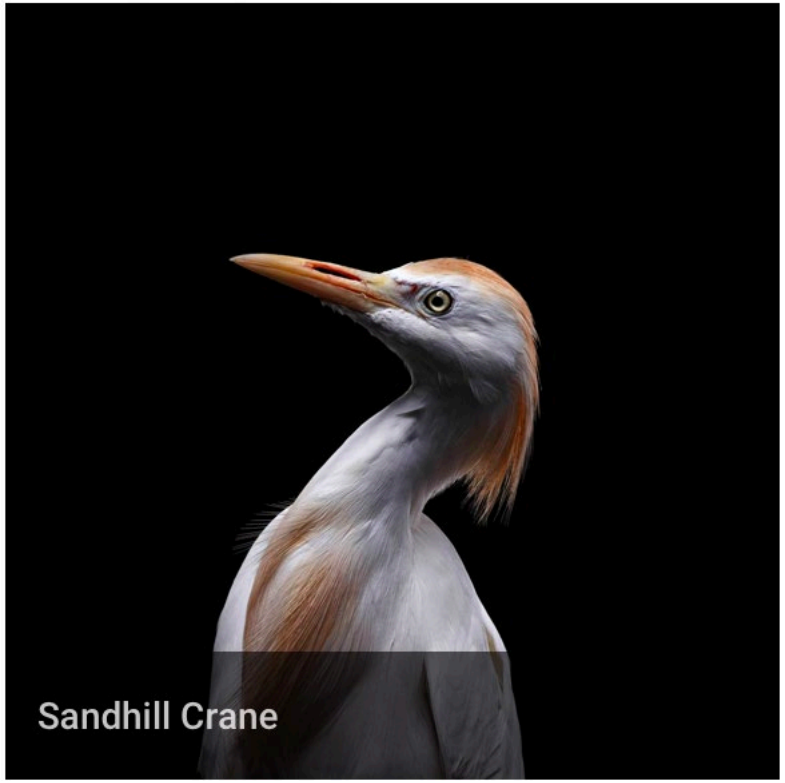
Label

SHOW

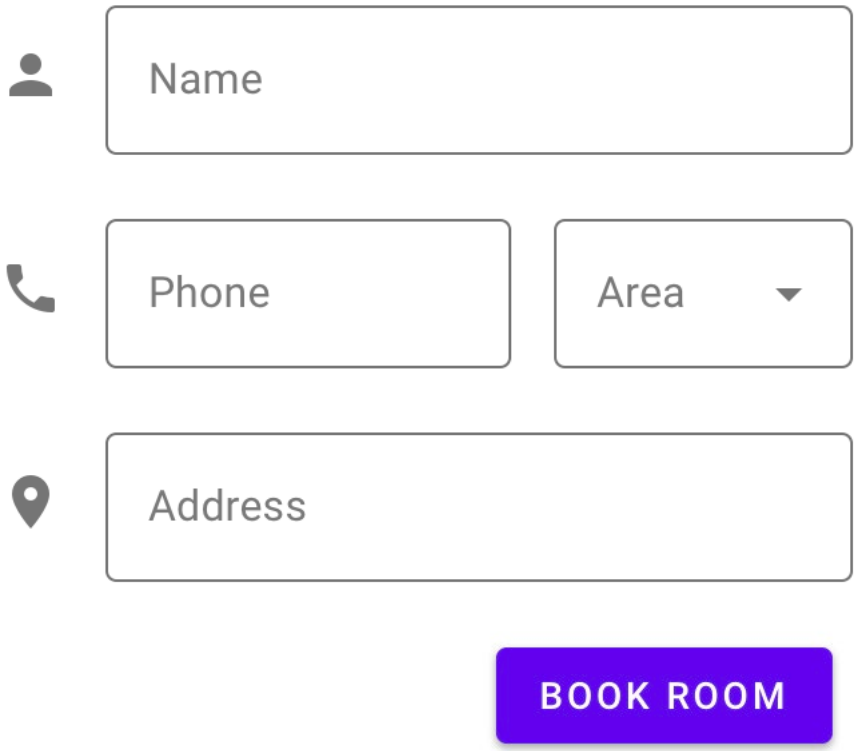
Label

Input text

255



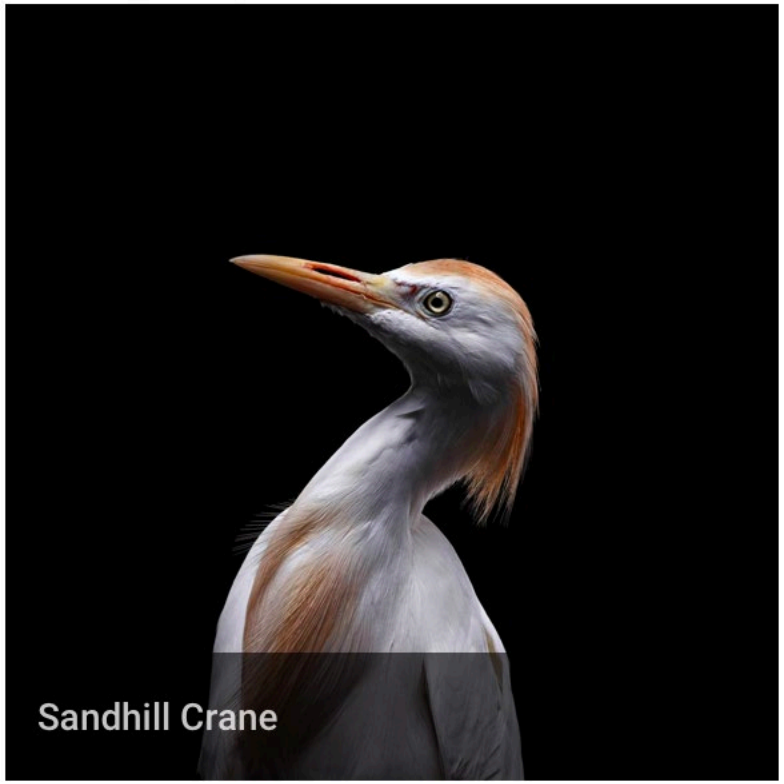
Compound Components



Compound Components

Enabled	Disabled
Promotions	Attractions
Radio	Dining
Food	Education
Sports	Health
Travel	Family
	Office

Compound Components



HIDE SHOW

Compound Components

Reu sability

Separation of concerns

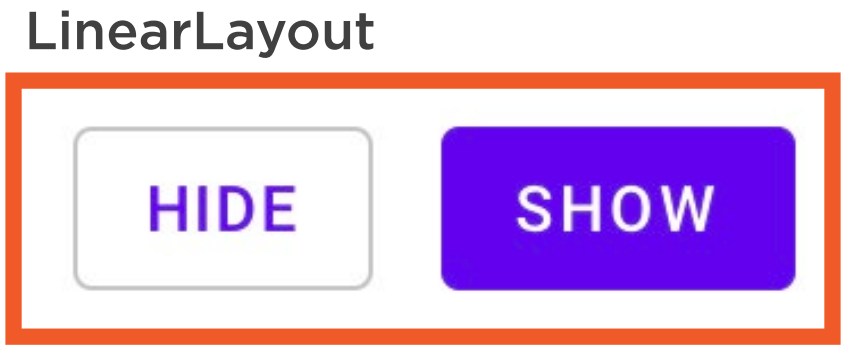
Combining Views into a Compound Component

Compound Components

HIDE

SHOW

Compound Components

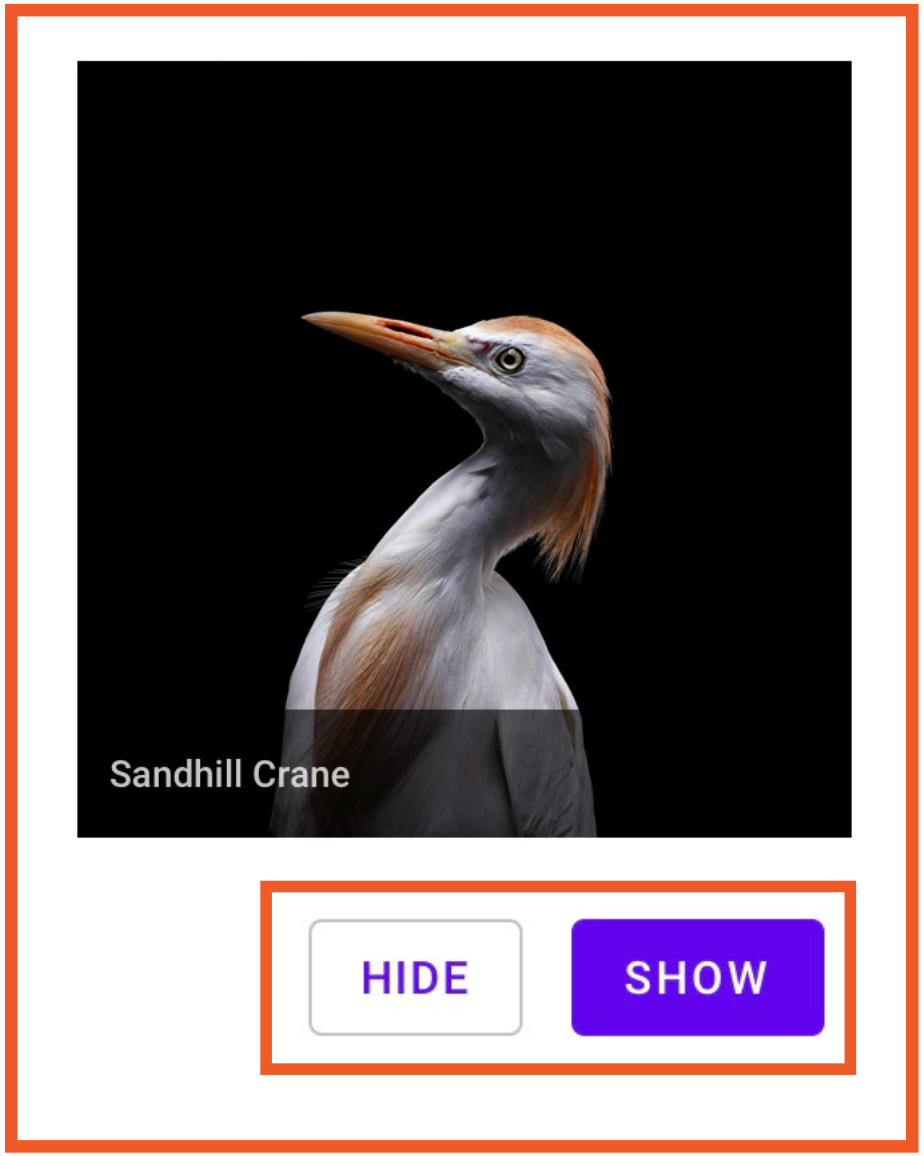


Compound Components

MyImageToggleView

Compound Components

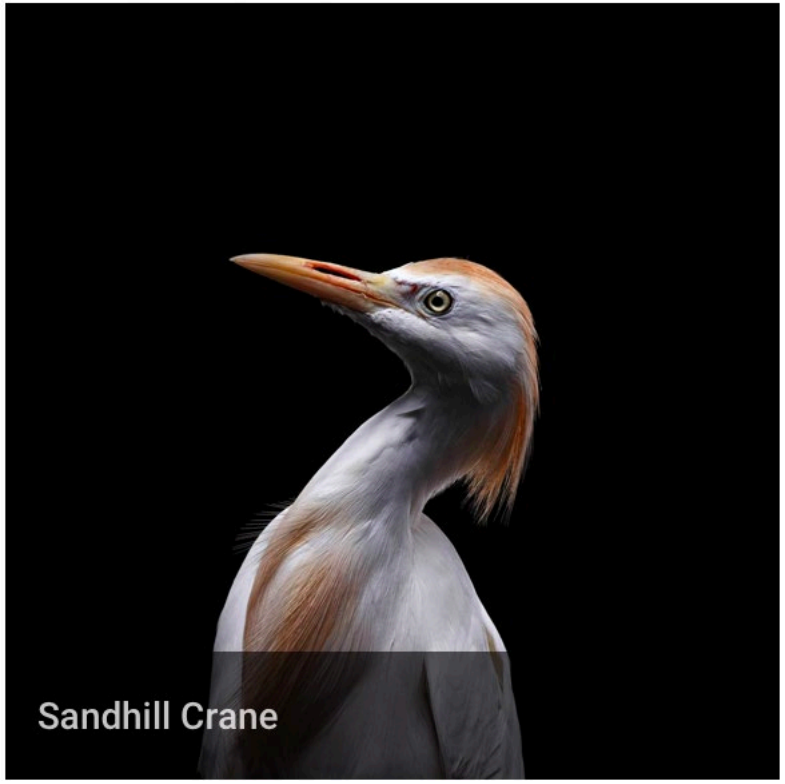
ConstraintLayout



<Live Coding>

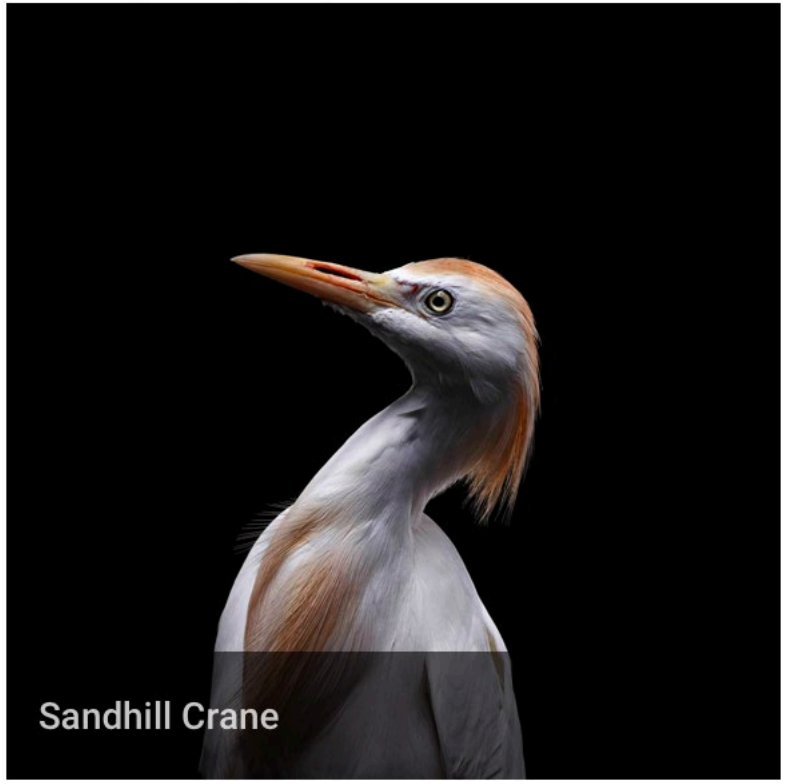
Communicating between View and Activity

Compound Components



HIDE SHOW

Compound Components



HIDE SHOW

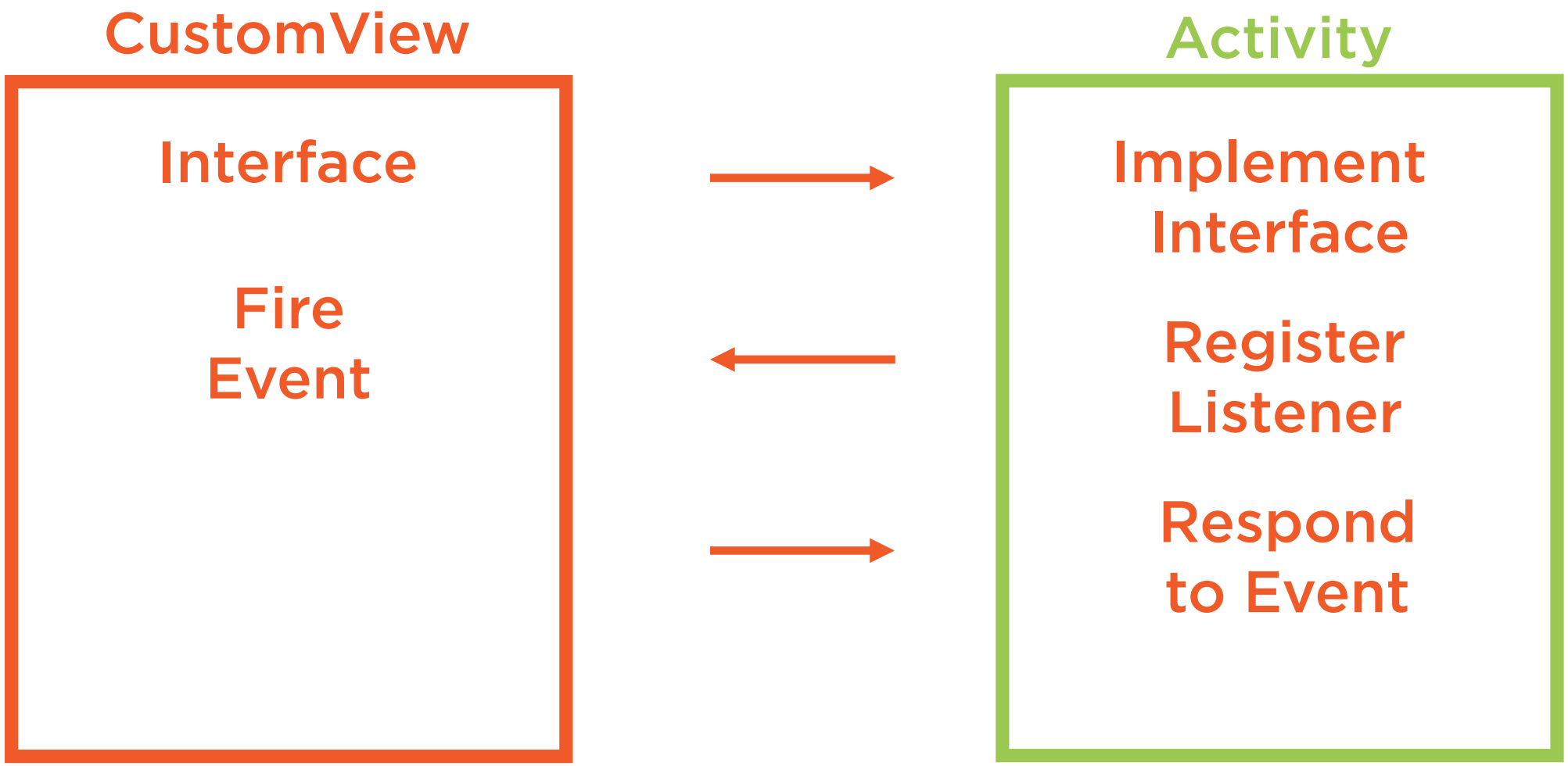
<Activity>

Communicate via interface

Interface

```
interface ColorSelectListener {
    fun onColorSelected(color: Int)
}
```

Communicate via Interface

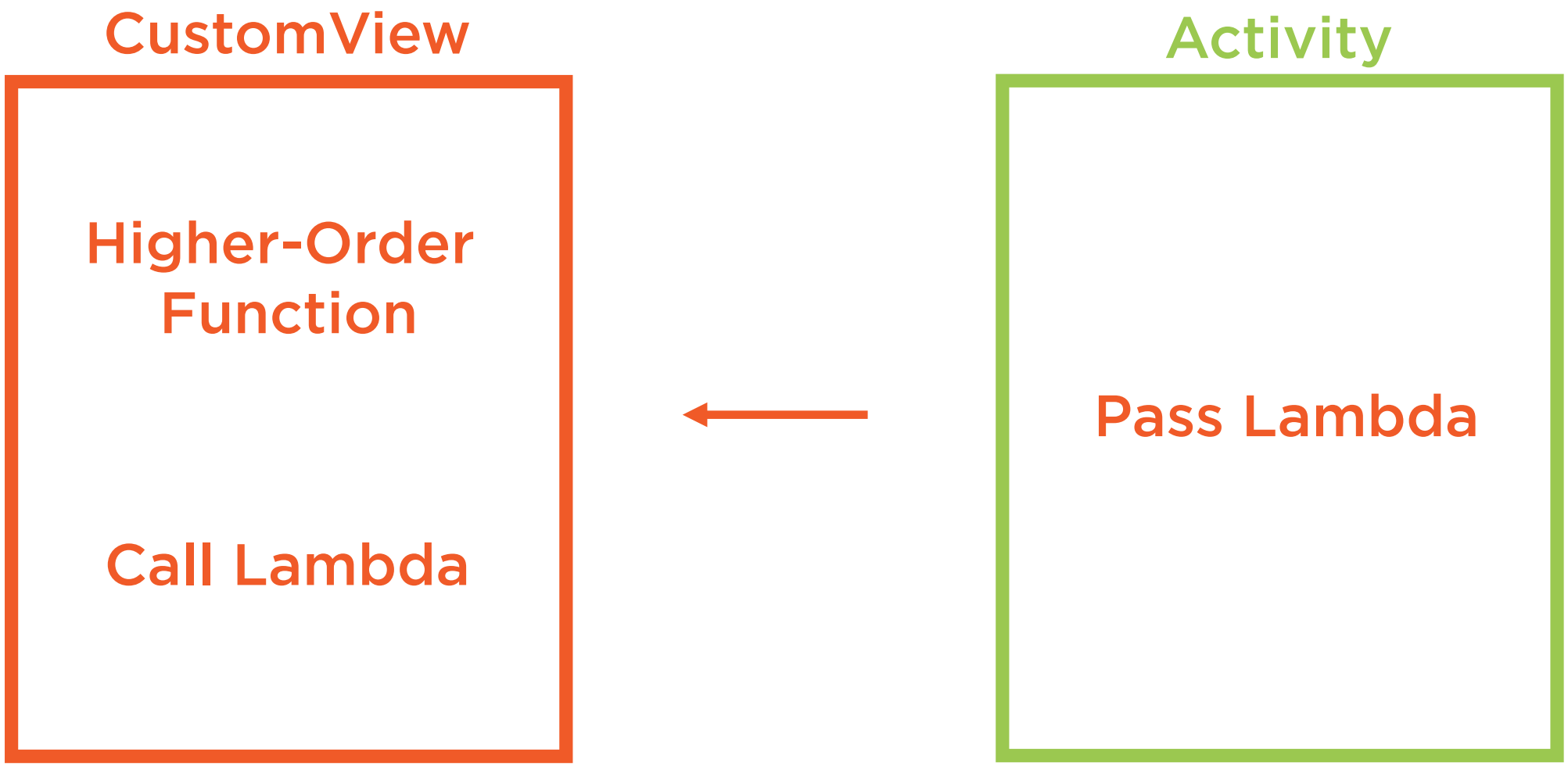


<Live Coding>

Communicate via interface

Communicate via Higher-Order Functions and Lambdas

Communication in Kotlin



Higher-Order Function

```
fun setOnClickListener(function: (View) -> Unit) {

}
```

Higher-Order Function

```
fun setOnClickListener(function: (View) -> Unit) {

}
```

Lambda Expression

```
myView.setOnClickListener { view ->
    view.visibility = View.GONE
}
```

<Live Coding>

Custom Attributes

Custom Attributes

```
class ColorSelector @JvmOverloads
constructor(context: Context, attributeSet: AttributeSet? =
null,defStyle: Int = 0, defRes: Int = 0)
    : LinearLayout(context, attributeSet, defStyle, defRes)
```

Custom Attributes

```
attributeSet: AttributeSet
```

Custom Attributes

Define custom attributes for your view in a <declare-styleable> resource element

Specify values for the attributes in your XML layout

Retrieve attribute values at runtime

Apply the retrieved attribute values to your view

Custom Attributes

```
<resources>
    <declare-styleable name="ColorSelector">
        <attr name="colors" format="reference" />
    </declare-styleable>
</resources>
```

Custom Attributes

```
format="reference"
       "integer"
       "string"
       "dimension"
```

Custom Attributes

```
<resources>
    <declare-styleable name="ColorSelector">
        <attr name="colors" format="reference" />
    </declare-styleable>
</resources>
```

<Live Coding>

Summary

Intro to compound components

Combining views into a compound component

Communicating between view and activity

Custom attributes

Next Up: Extending Views