Uniwersytet Wrocławski

Rafał Lewandków

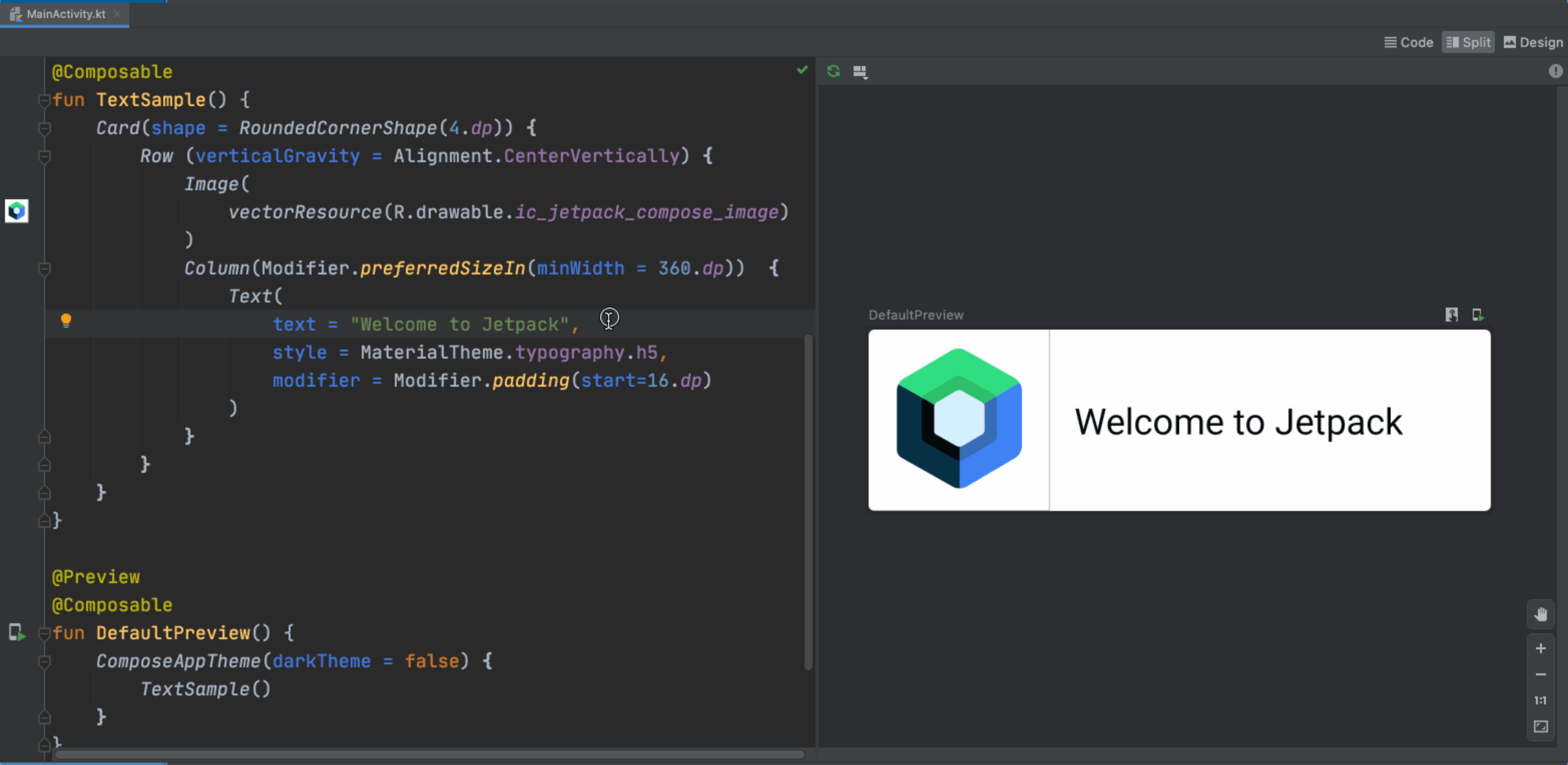
PROGRAMOWANIE URZĄDZEŃ MOBILNYCH

WYKŁAD 5
Jetpack Compose

- Fundamenty
- Komponenty
- @Composable
- MutableState

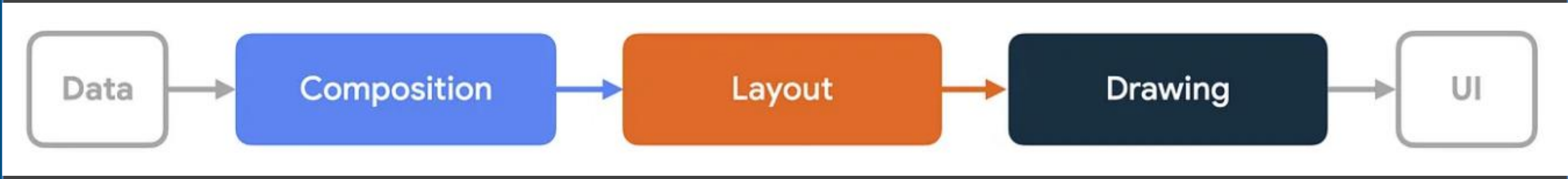
Uniwersytet Wrocławski

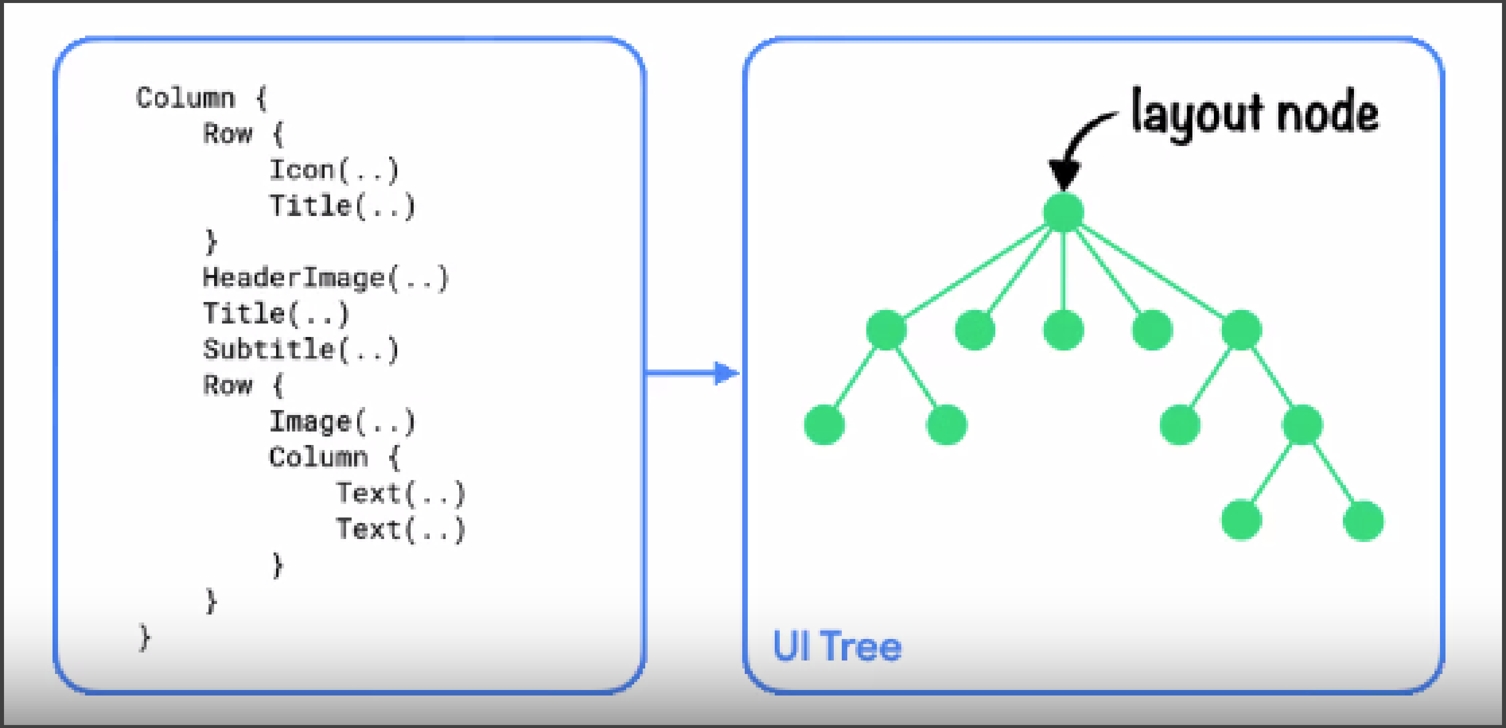
Jetpack Compose



```kotlin
@Composable
fun TextSample() {
    Card(shape = RoundedCornerShape(4.dp)) {
        Row (verticalGravity = Alignment.CenterVertically) {
            Image(
                vectorResource(R.drawable.ic_jetpack_compose_image)
            )
            Column(Modifier.preferredSizeIn(minWidth = 360.dp))  {
                Text(
                    text = "Welcome to Jetpack",
                    style = MaterialTheme.typography.h5,
                    modifier = Modifier.padding(start=16.dp)
                )
            }
        }
    }
}

@Preview
@Composable
fun DefaultPreview() {
    ComposeAppTheme(darkTheme = false) {
        TextSample()
    }
}
```

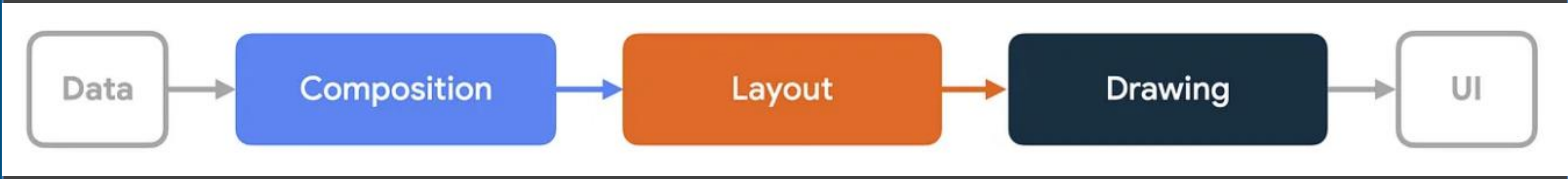
Jetpack Compose - Fazy

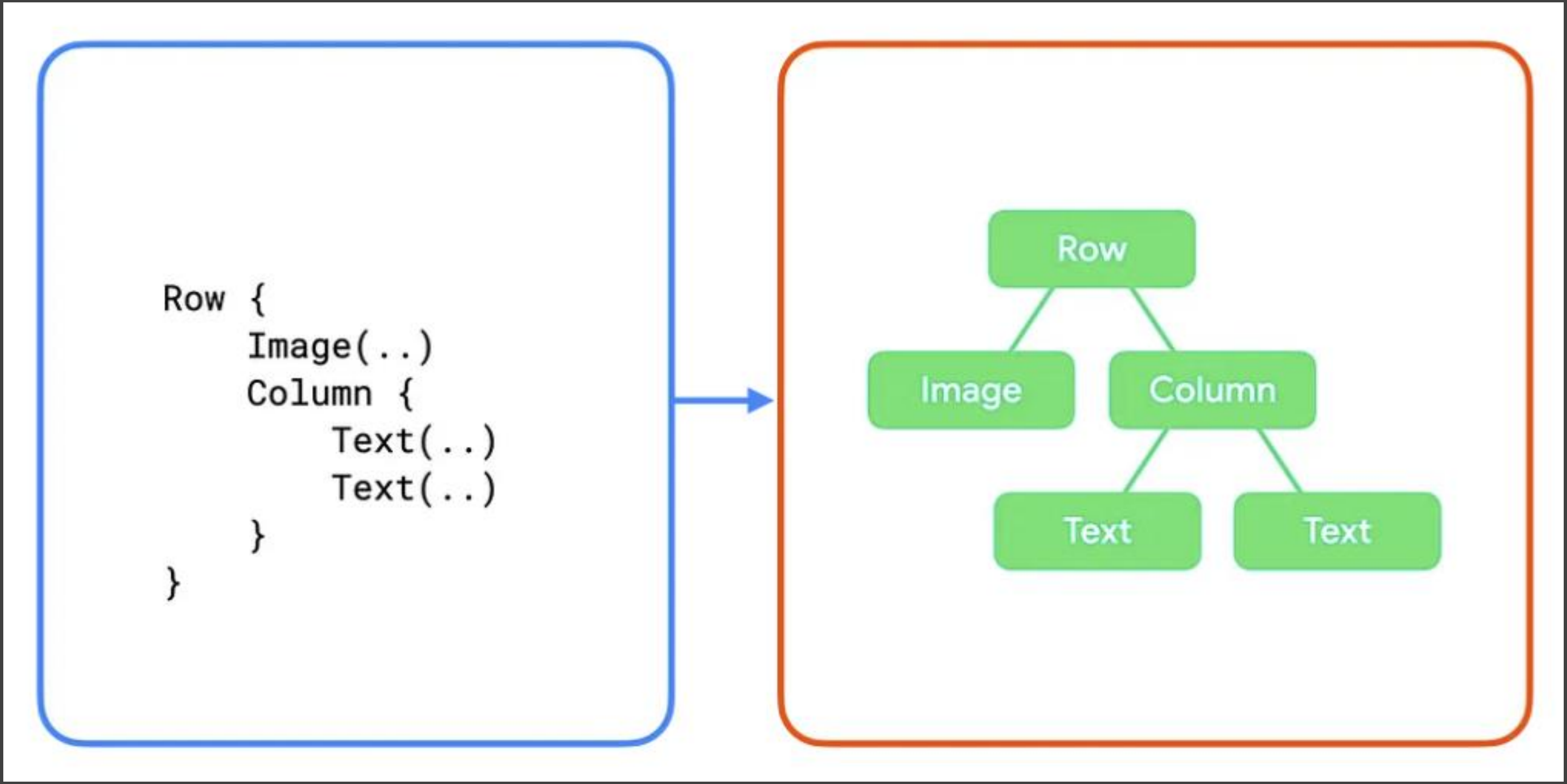




```
Column {
    Row {
        Icon(..)
        Title(..)
    }
    HeaderImage(..)
    Title(..)
    Subtitle(..)
    Row {
        Image(..)
        Column {
            Text(..)
            Text(..)
        }
    }
}
```

Jetpack Compose - Fazy

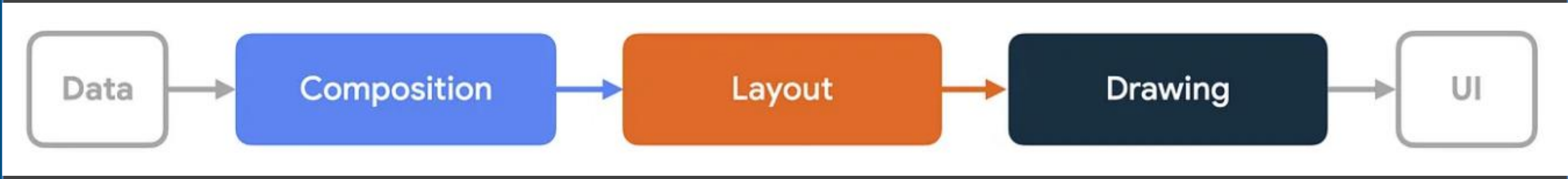


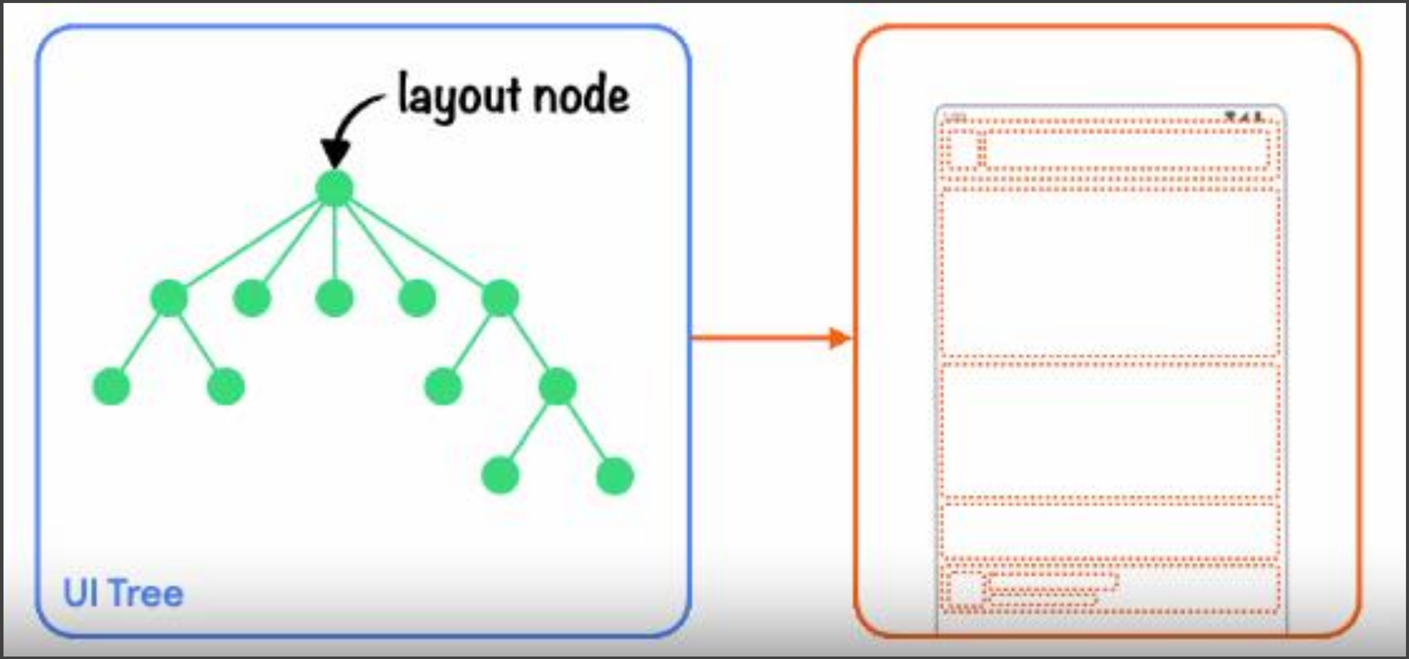


```
Row {
    Image(..)
    Column {
        Text(..)
        Text(..)
    }
}
```

Uniwersytet Wrocławski

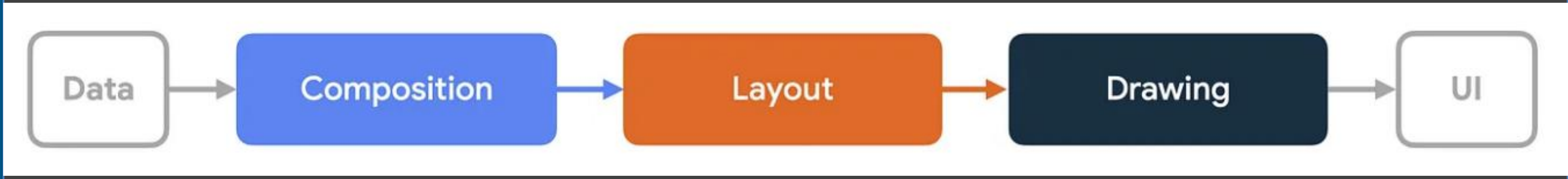
Jetpack Compose - Fazy

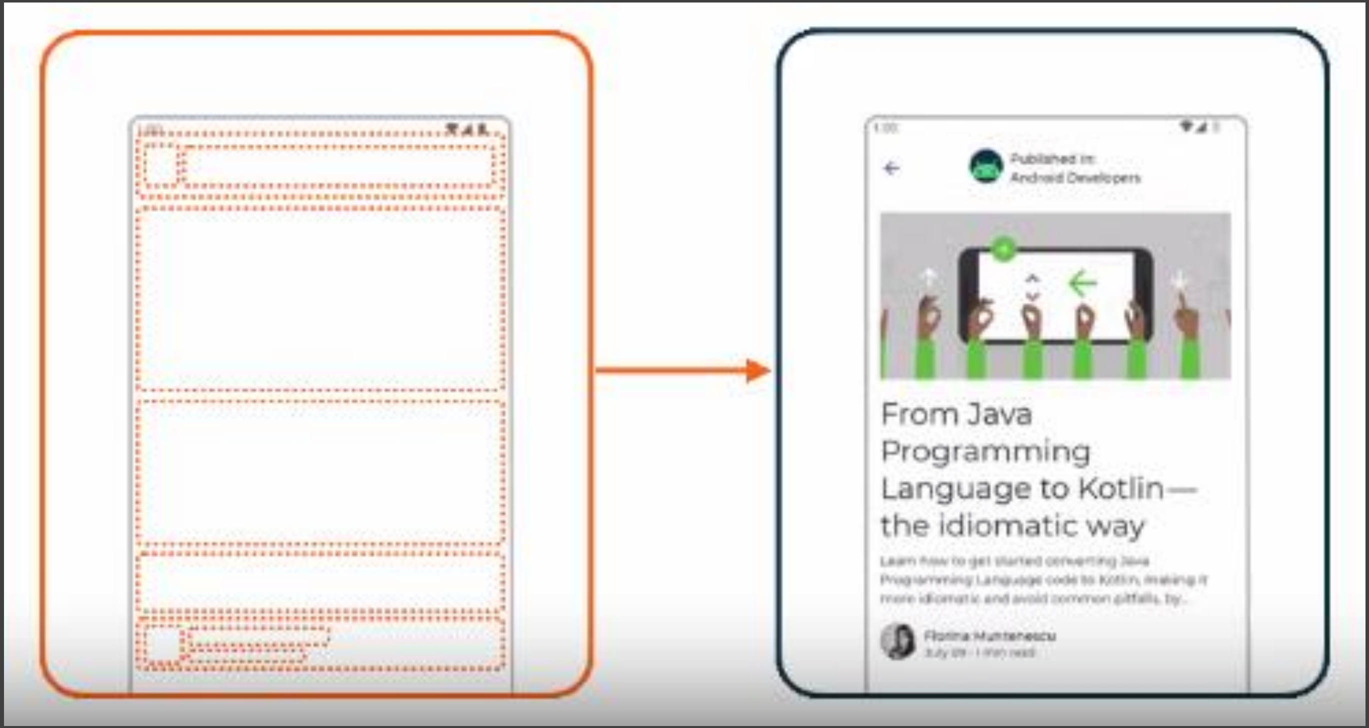




Uniwersytet Wrocławski

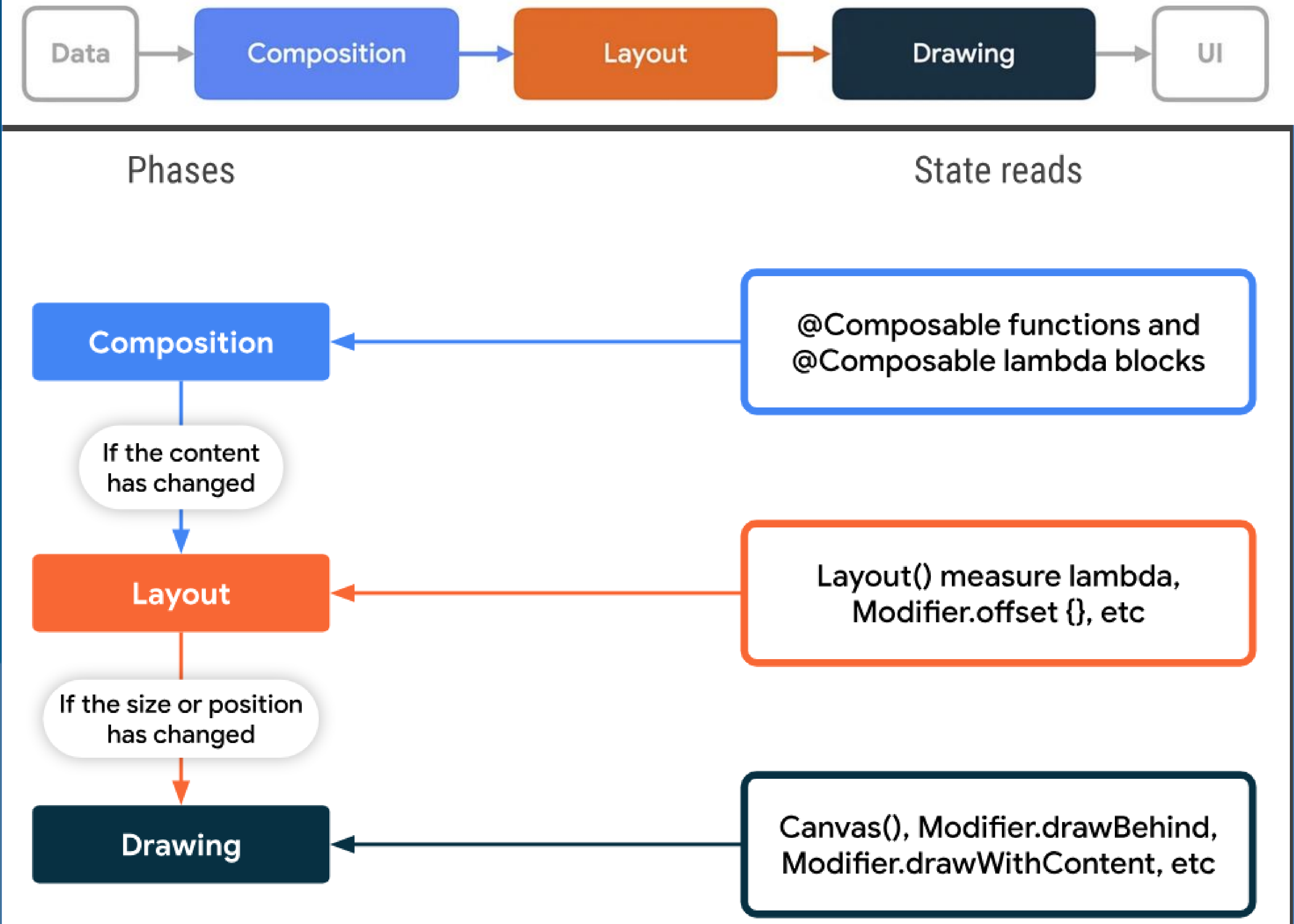
Jetpack Compose - Fazy




Published in:
Android Developers
From Java Programming Language to Kotlin— the idiomatic way
Learn how to get started converting Java Programming Language code to Kotlin, making it more idiomatic and avoid common pitfalls, by...
Florina Muntenescu

Uniwersytet Wrocławski

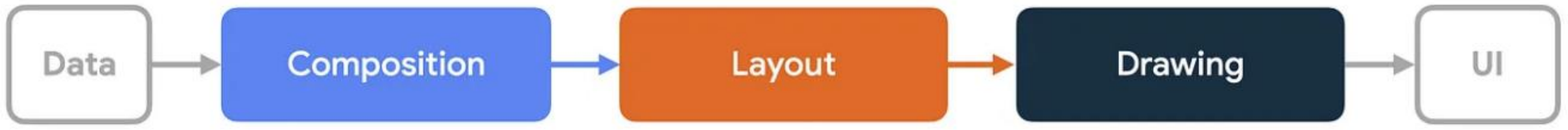
Jetpack Compose - Fazy

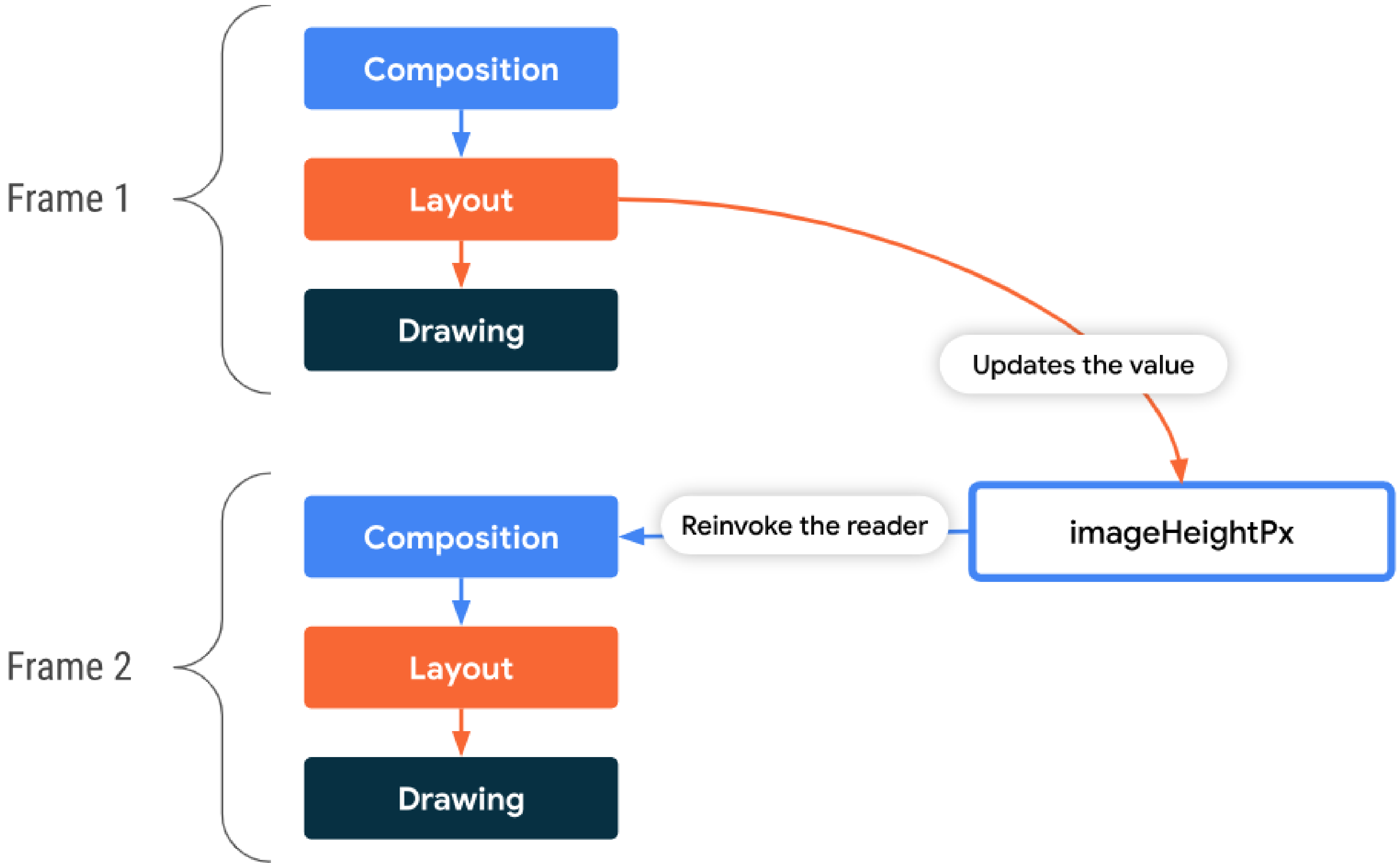


Data
Composition
Layout
Drawing
UI
Phases
State reads
Composition
If the content has changed
Layout
If the size or position has changed
Drawing
@Composable functions and @Composable lambda blocks
Layout() measure lambda, Modifier.offset {}, etc
Canvas(), Modifier.drawBehind, Modifier.drawWithContent, etc

Uniwersytet Wrocławski

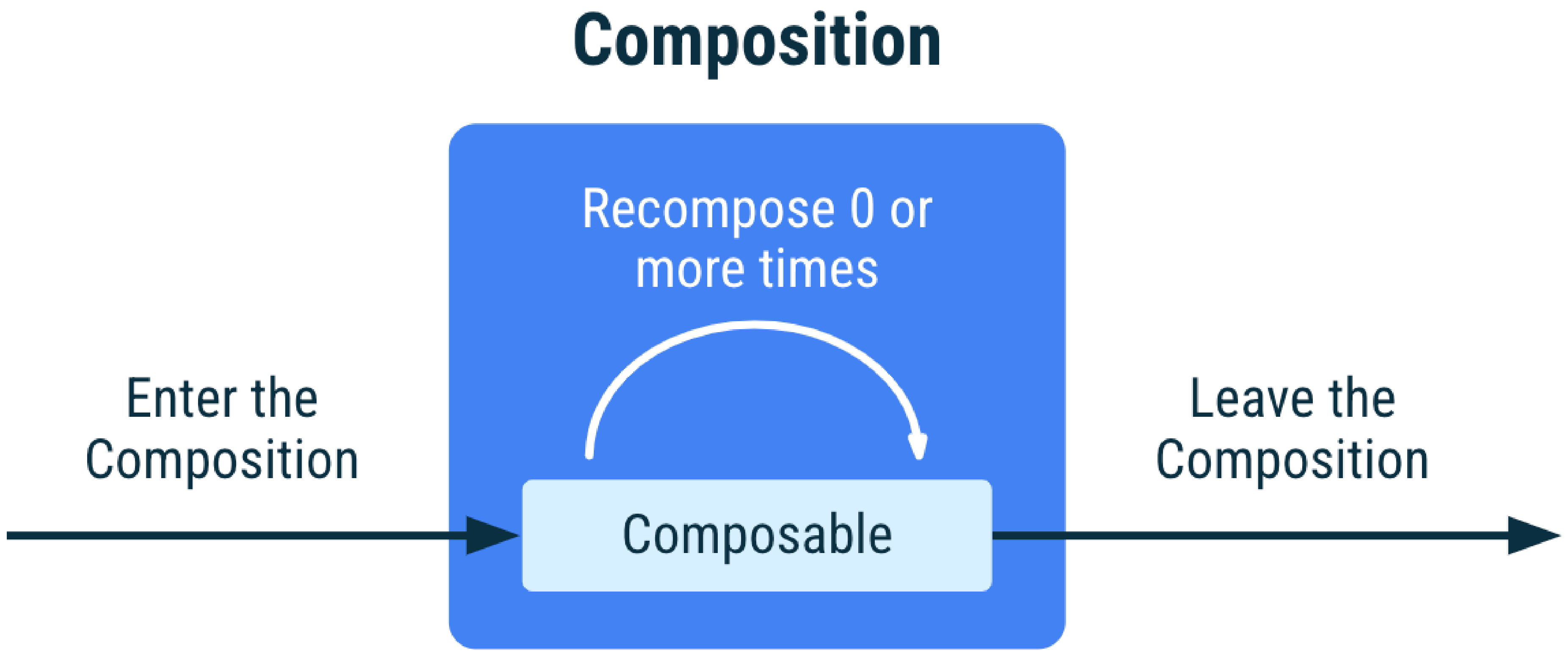
Jetpack Compose - Fazy



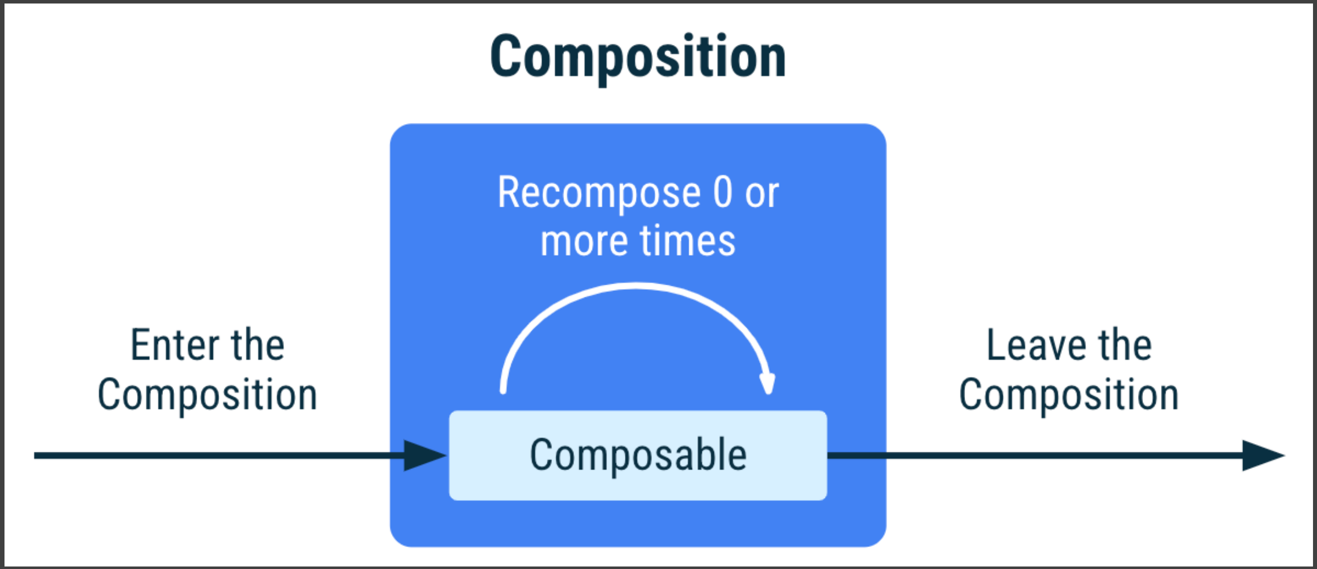


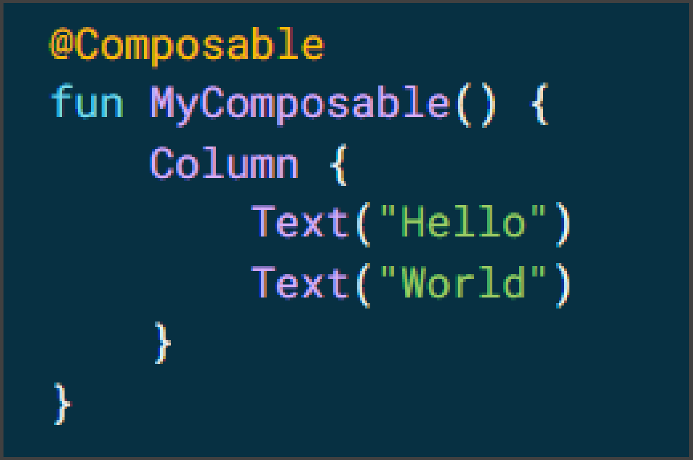
Uniwersytet Wrocławski

State

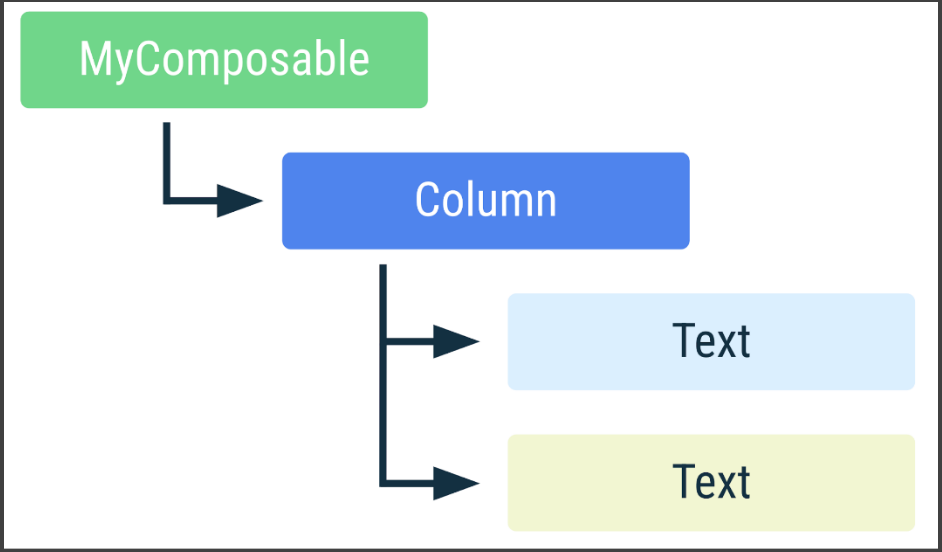


State



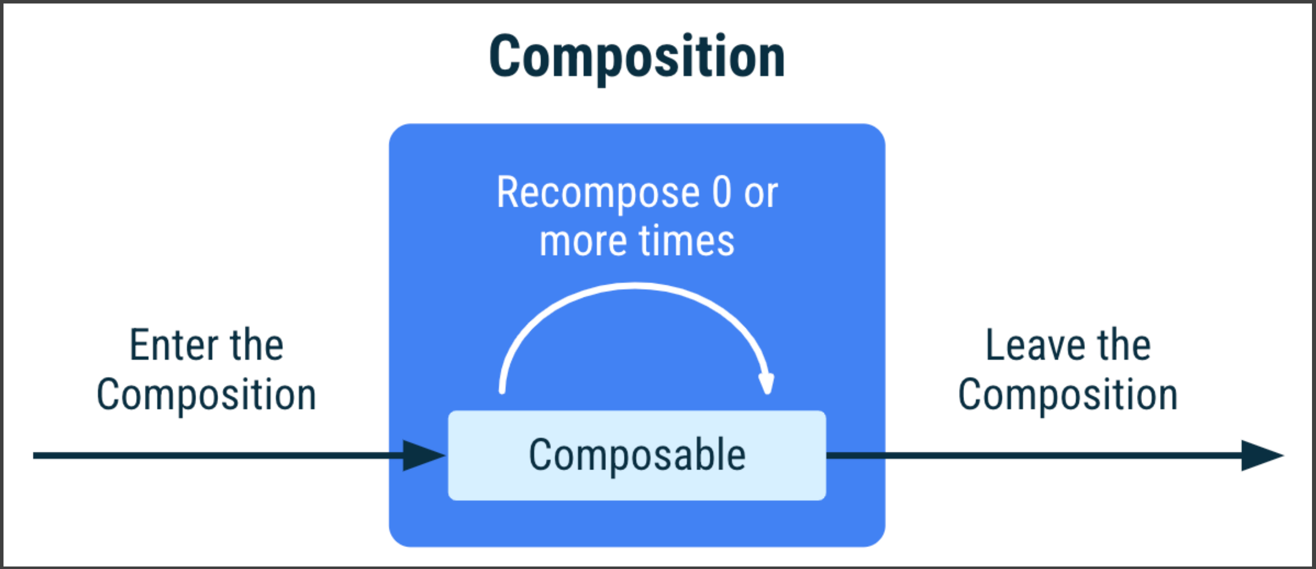


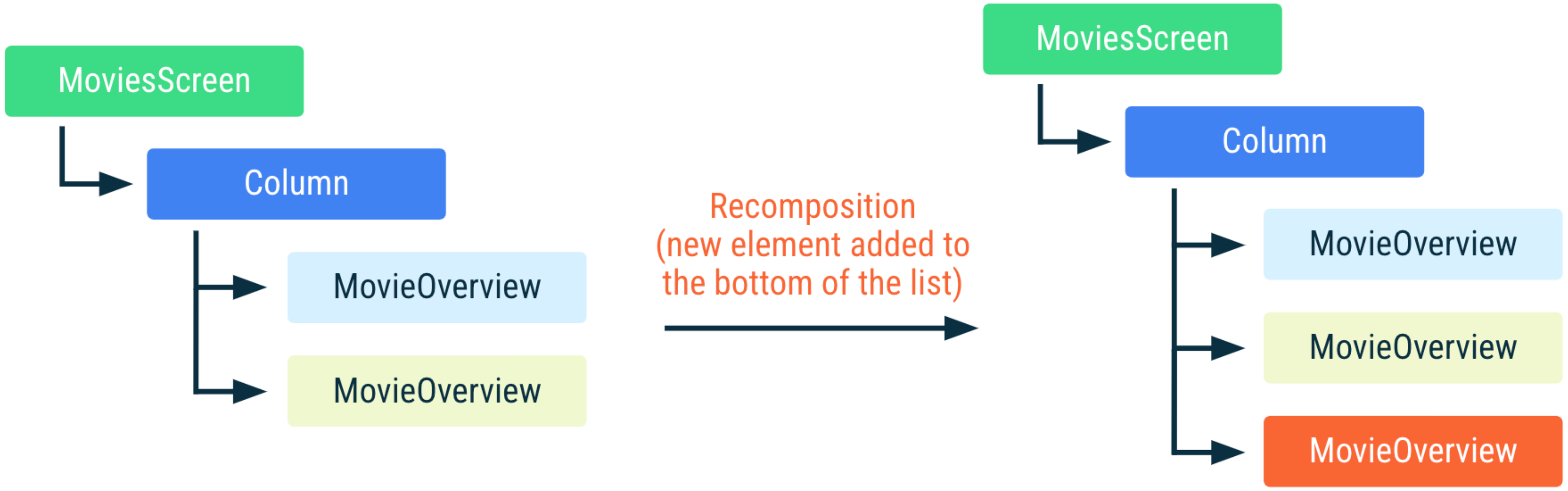
```
@Composable
fun MyComposable() {
    Column {
        Text("Hello")
        Text("World")
    }
}
```



Uniwersytet Wrocławski

State

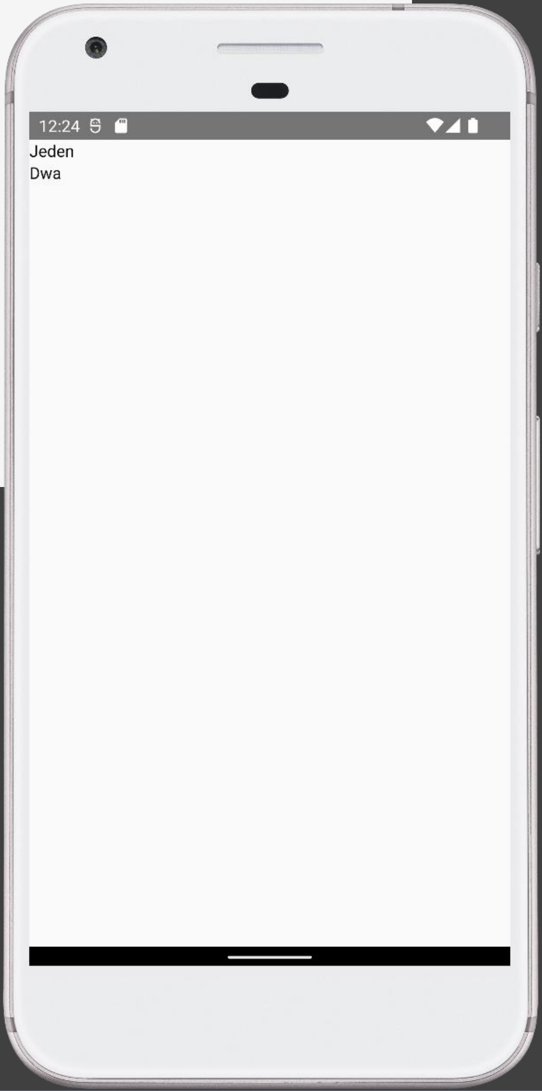




Uniwersytet Wrocławski

Jetpack Compose

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContent {
        Column {
            Text(text = "Jeden")
            Text(text = "Dwa")
        }
    }
}
```

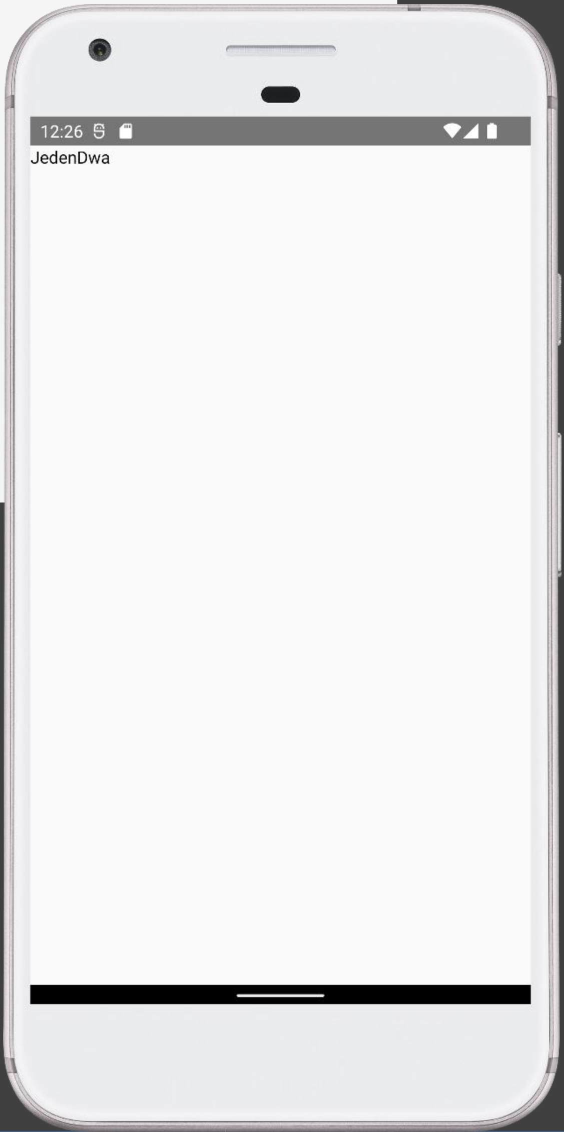

12:24

Jeden
Dwa

Uniwersytet Wrocławski

Jetpack Compose

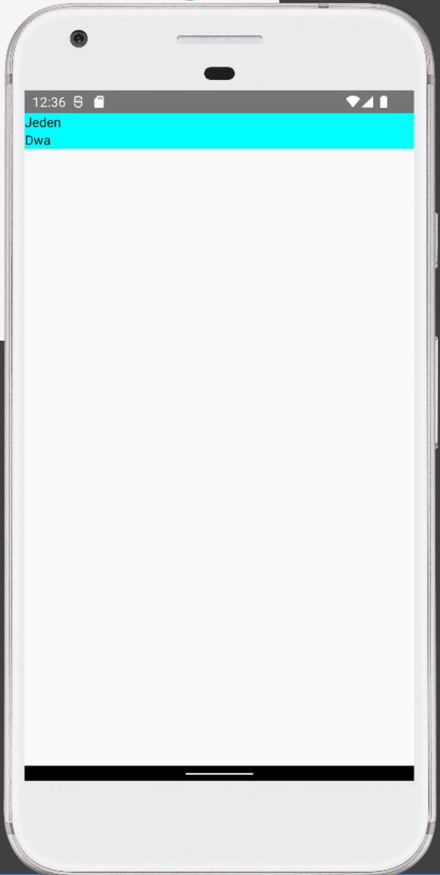
```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContent {
        Row {
            Text(text = "Jeden")
            Text(text = "Dwa")
        }
    }
}
```


12:26

JedenDwa

Jetpack Compose

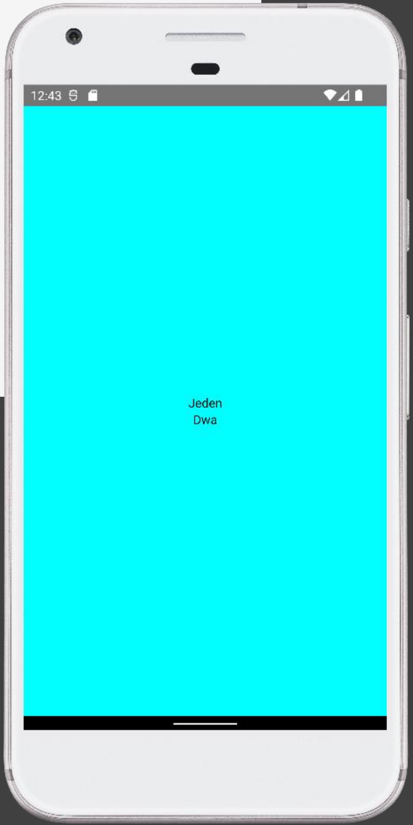
```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContent {
        Column(
            modifier = Modifier
                .fillMaxWidth() // szerokość kolumny
                .background(Color.Cyan) // kolor tła kolumny
        ) {
            Text(text = "Jeden")
            Text(text = "Dwa")
        }
    }
}
```



Uniwersytet Wrocławski

Jetpack Compose

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContent {
        Column(
            modifier = Modifier
                .fillMaxSize()
                .background(Color.Cyan),
            horizontalAlignment = Alignment.CenterHorizontally,
            verticalArrangement = Arrangement.Center
        ) {
            Text(text = "Jeden")
            Text(text = "Dwa")
        }
    }
}
```

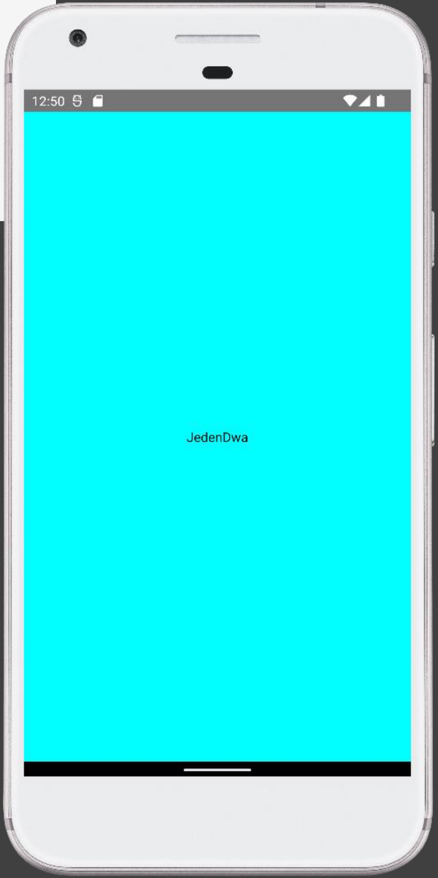


```
Jeden
Dwa
```

Uniwersytet Wrocławski

Jetpack Compose

```
Row(
    modifier = Modifier
        .fillMaxSize()
        .background(Color.Cyan),
    horizontalArrangement = Arrangement.Center,
    verticalAlignment = Alignment.CenterVertically
) {
    Text(text = "Jeden")
    Text(text = "Dwa")
}
```

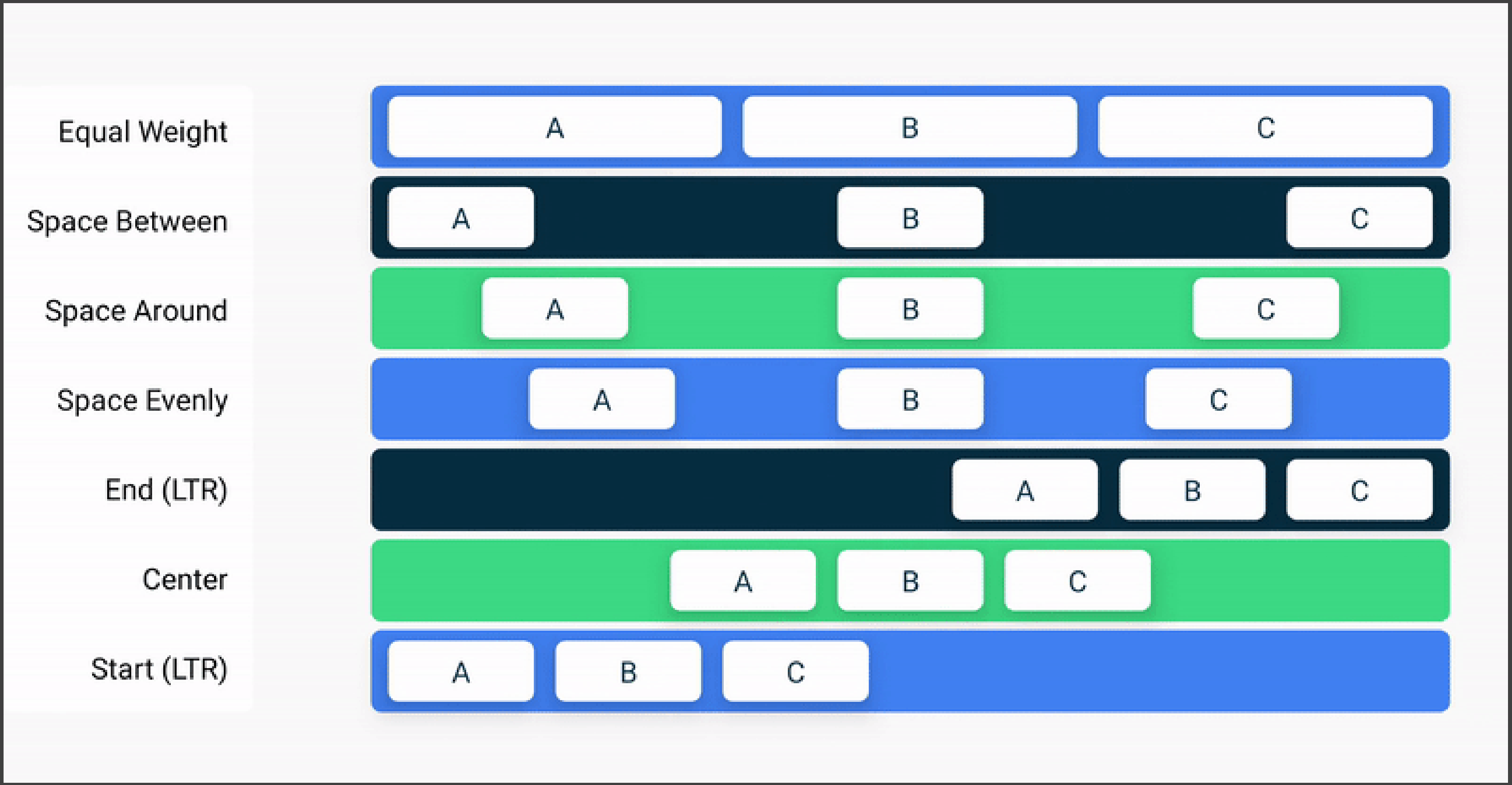


Jetpack Compose

```
Column(
    modifier = Modifier.fillMaxSize(),
    verticalArrangement = Arrangement.spacedBy(8.dp)
) {
    Text(
        text = "Element 1",
        modifier = Modifier.size(50.dp)
    )
    Text(
        text = "Element 2",
        modifier = Modifier.size(80.dp)
    )
    Text(
        text = "Element 3",
        modifier = Modifier.size(100.dp)
    )
}
```

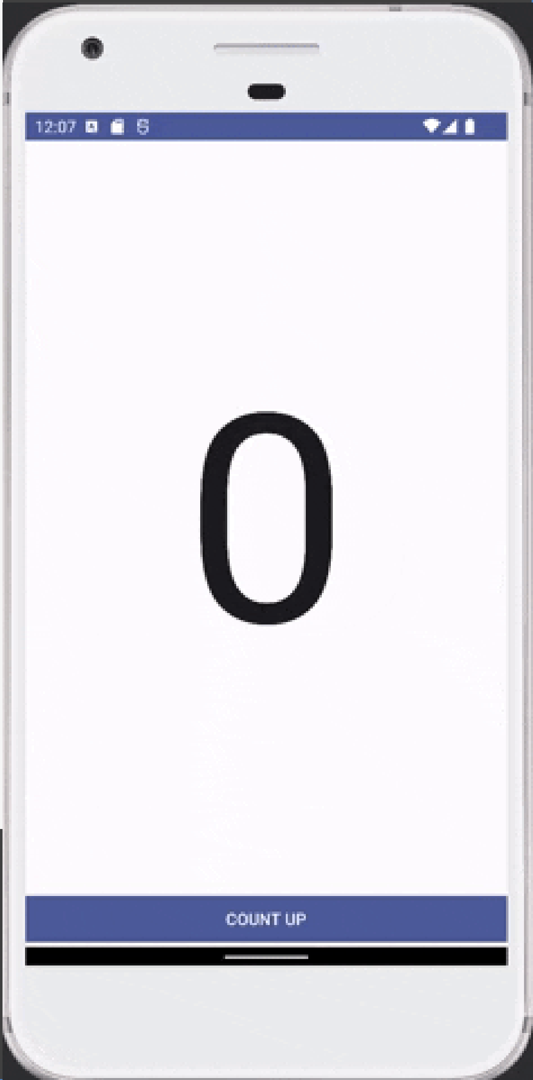
Uniwersytet Wrocławski

Jetpack Compose



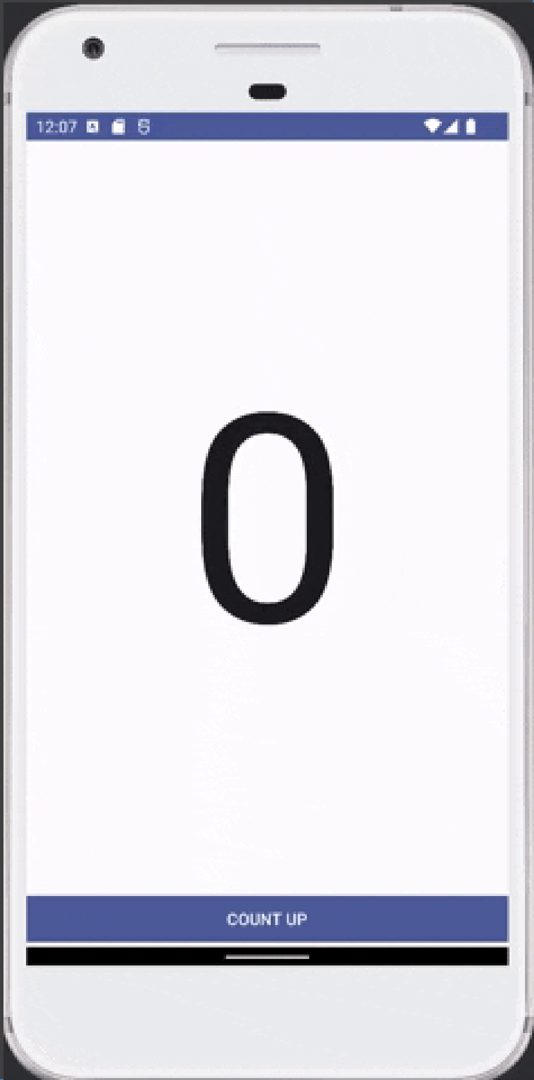
State

```
@Composable
fun CounterExample() {
    Column(
        verticalArrangement = Arrangement.Center,
        horizontalAlignment = Alignment.CenterHorizontally,
        modifier = Modifier.fillMaxSize()
    ) {
        Spacer(modifier = Modifier.weight(0.3f))
        Text(
            text = "test",
            fontSize = 250.sp,
            textAlign = TextAlign.Center,
            modifier = Modifier.weight(1f),
        )
        Button(
            modifier = Modifier.fillMaxWidth(),
            shape = RectangleShape
        ) {
            Text(text = "COUNT UP")
        }
    }
}
```



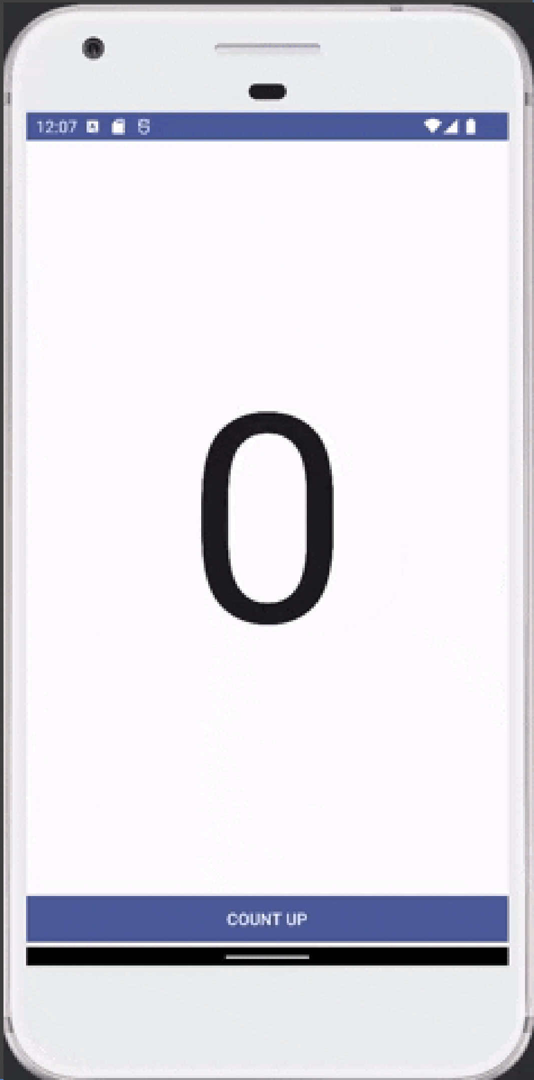
State

```kotlin
@Composable
fun CounterExample() {
    Column(
        verticalArrangement = Arrangement.Center,
        horizontalAlignment = Alignment.CenterHorizontally,
        modifier = Modifier.fillMaxSize()
    ) {

        val count: MutableState<Int> = remember {
            mutableStateOf(0)
        }

        Spacer(modifier = Modifier.weight(0.3f))

        Text(
            text = "${count.value}",
            fontSize = 250.sp,
            textAlign = TextAlign.Center,
            modifier = Modifier.weight(1f),
        )

        Button(
            onClick = { count.value++ }, // onclick
            modifier = Modifier.fillMaxWidth(),
            shape = RectangleShape
        ) {
            Text(text = "COUNT UP")
        }
    }
}
```



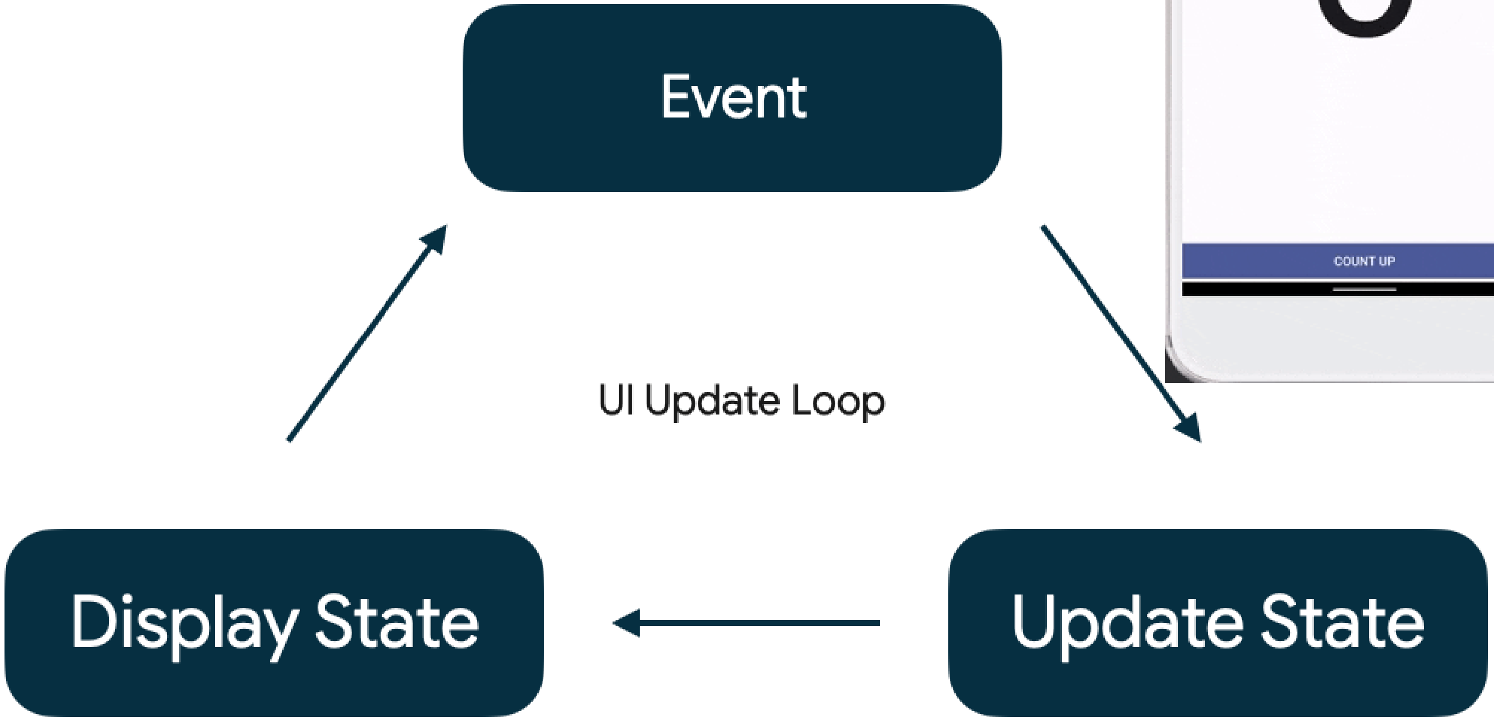
State

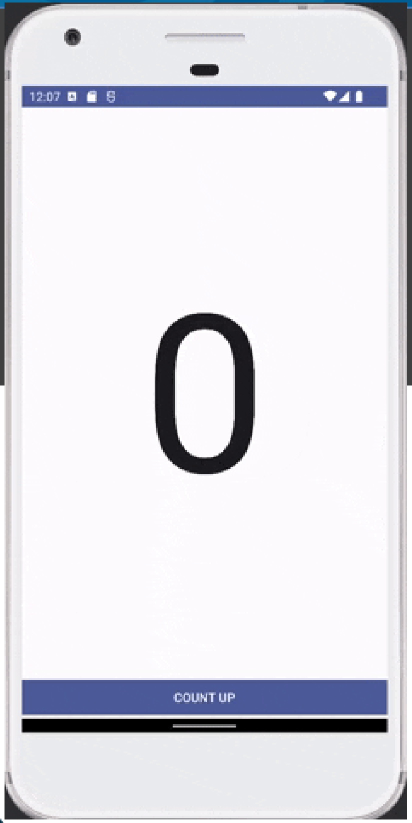
```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContent {
        JetpackComposeStateBasicsTheme {
            Surface(
                modifier = Modifier.fillMaxSize(),
                color = MaterialTheme.colorScheme.background
            ) {
                CounterExample()
            }
        }
    }
}
```



State

```
val count: MutableState<Int> = remember {
    mutableStateOf(0)
}
```


Event
UI Update Loop
Display State
Update State


12:07
0
COUNT UP