Uniwersytet Wrocławski

Rafał Lewandków

# PROGRAMOWANIE URZĄDZEŃ MOBILNYCH

## WYKŁAD 4

Jetpack Compose

- Fundamenty
- Komponenty
- @Composable
- MutableState

Uniwersytet Wrocławski

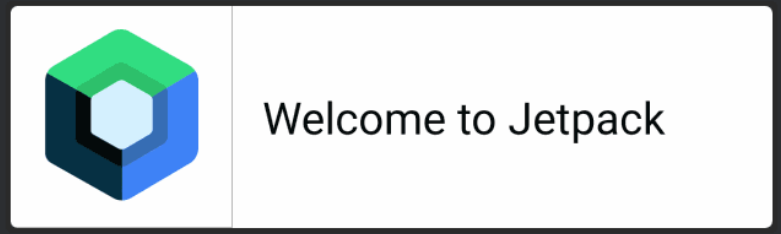
# Jetpack Compose

MainActivity.kt

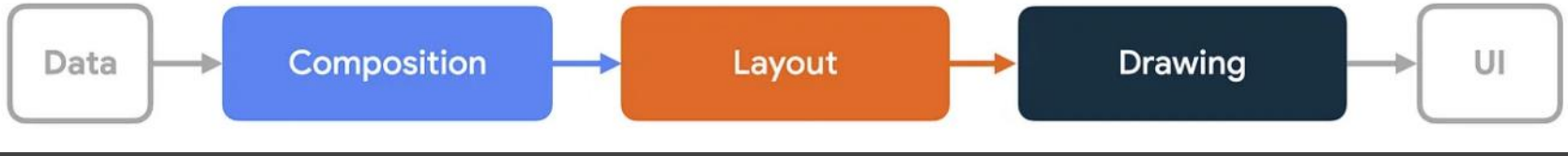
Code Split Design

```
@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()
    }
}
```

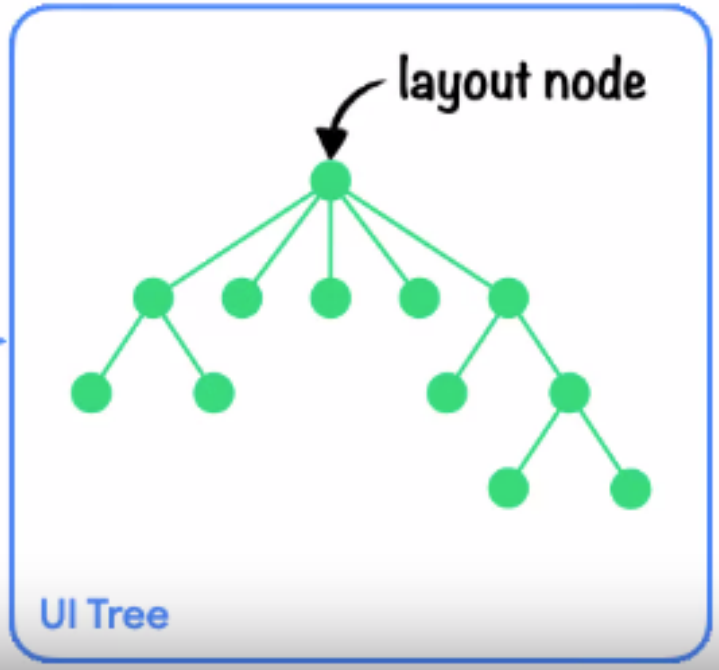
DefaultPreview



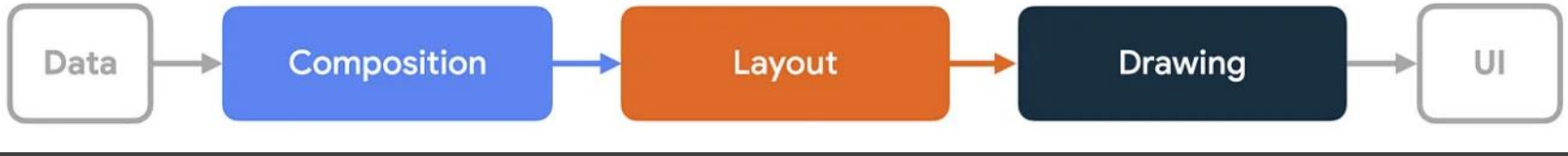
# Jetpack Compose - Fazy



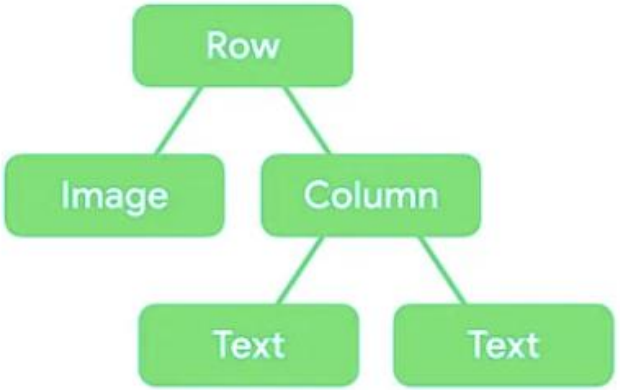

```
Column {
    Row {
        Icon(..)
        Title(..)
    }
    HeaderImage(..)
    Title(..)
    Subtitle(..)
    Row {
        Image(..)
        Column {
            Text(..)
            Text(..)
        }
    }
}
```



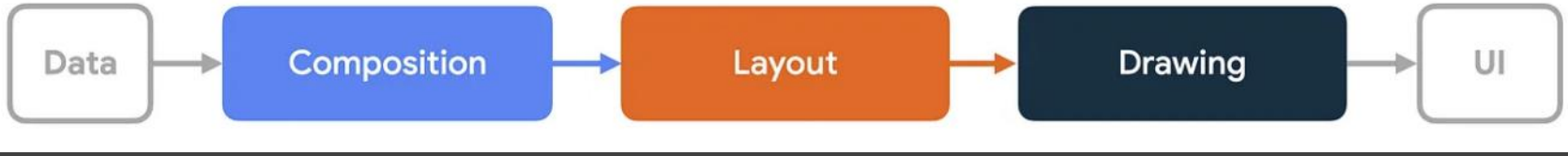
# Jetpack Compose - Fazy

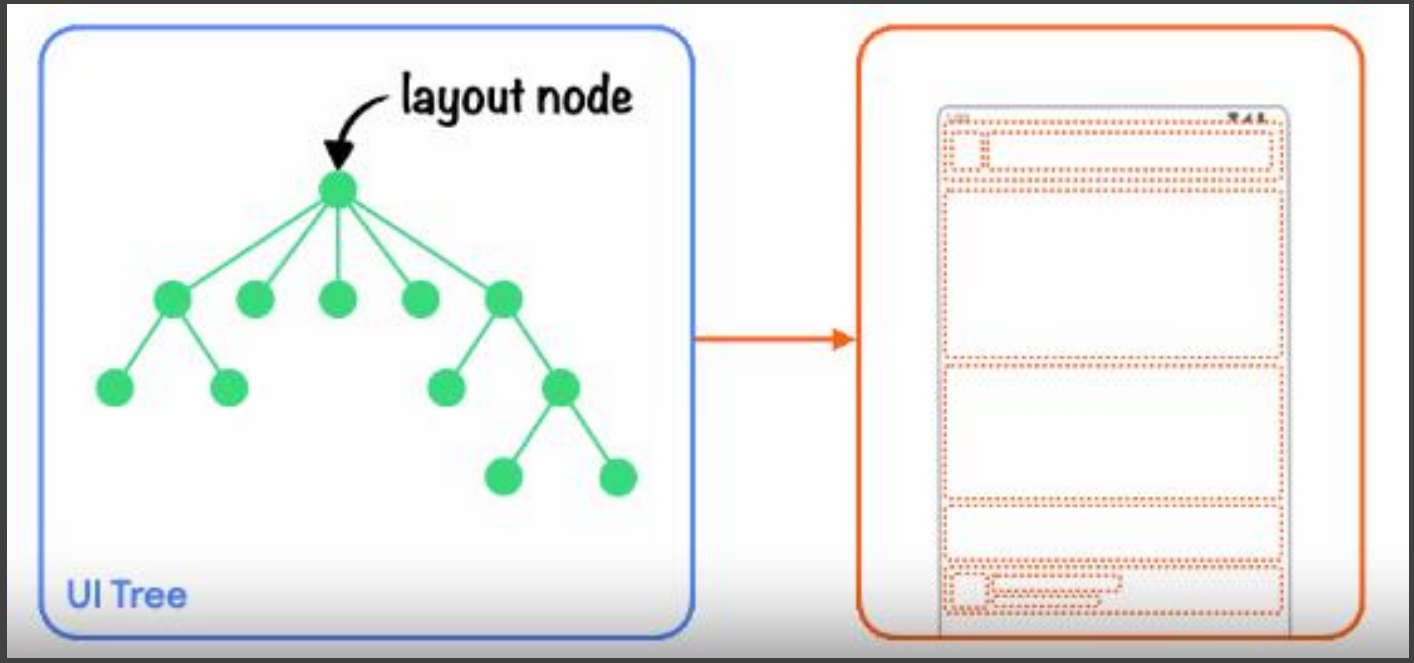



```
Row {
    Image(..)
    Column {
        Text(..)
        Text(..)
    }
}
```

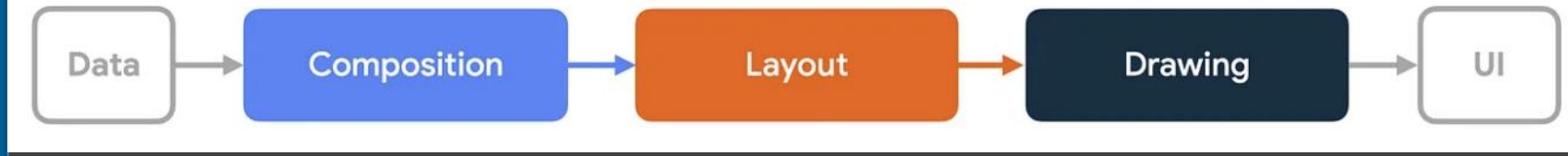


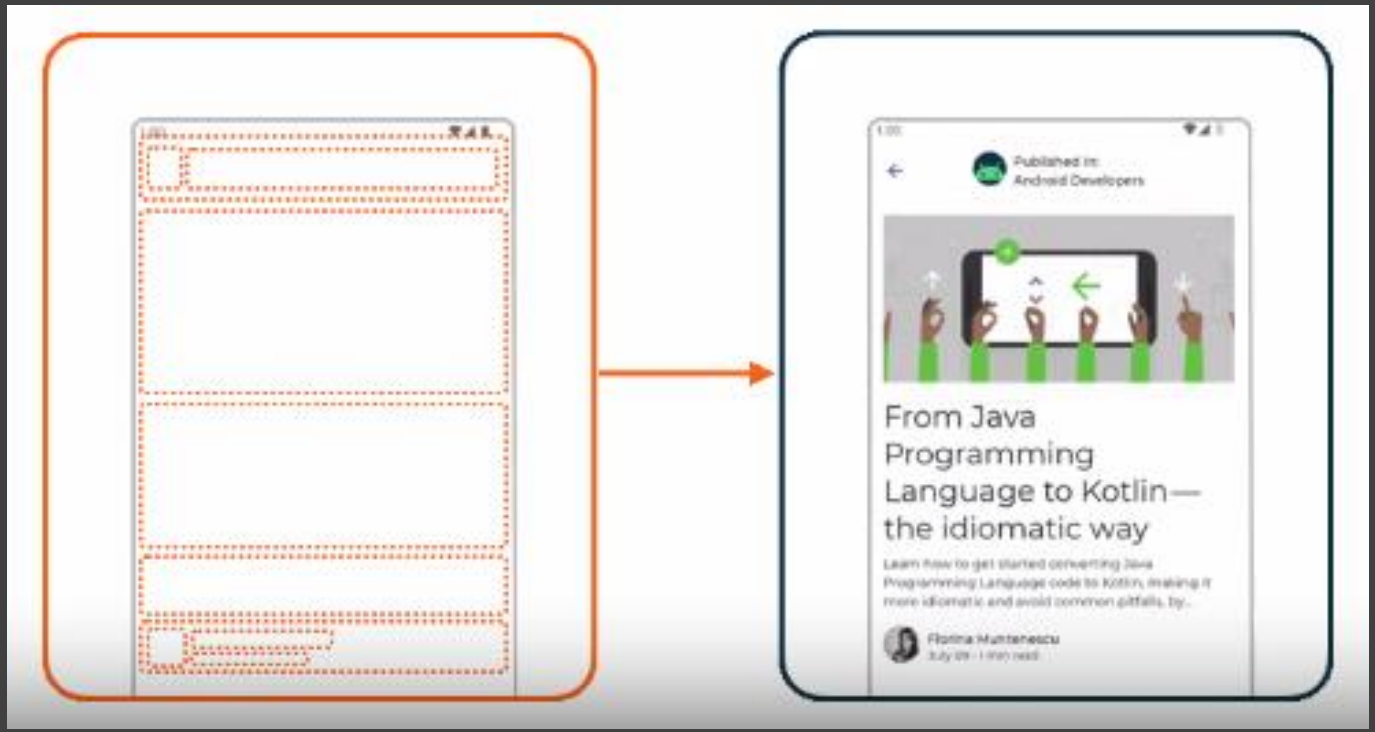
# Jetpack Compose - Fazy



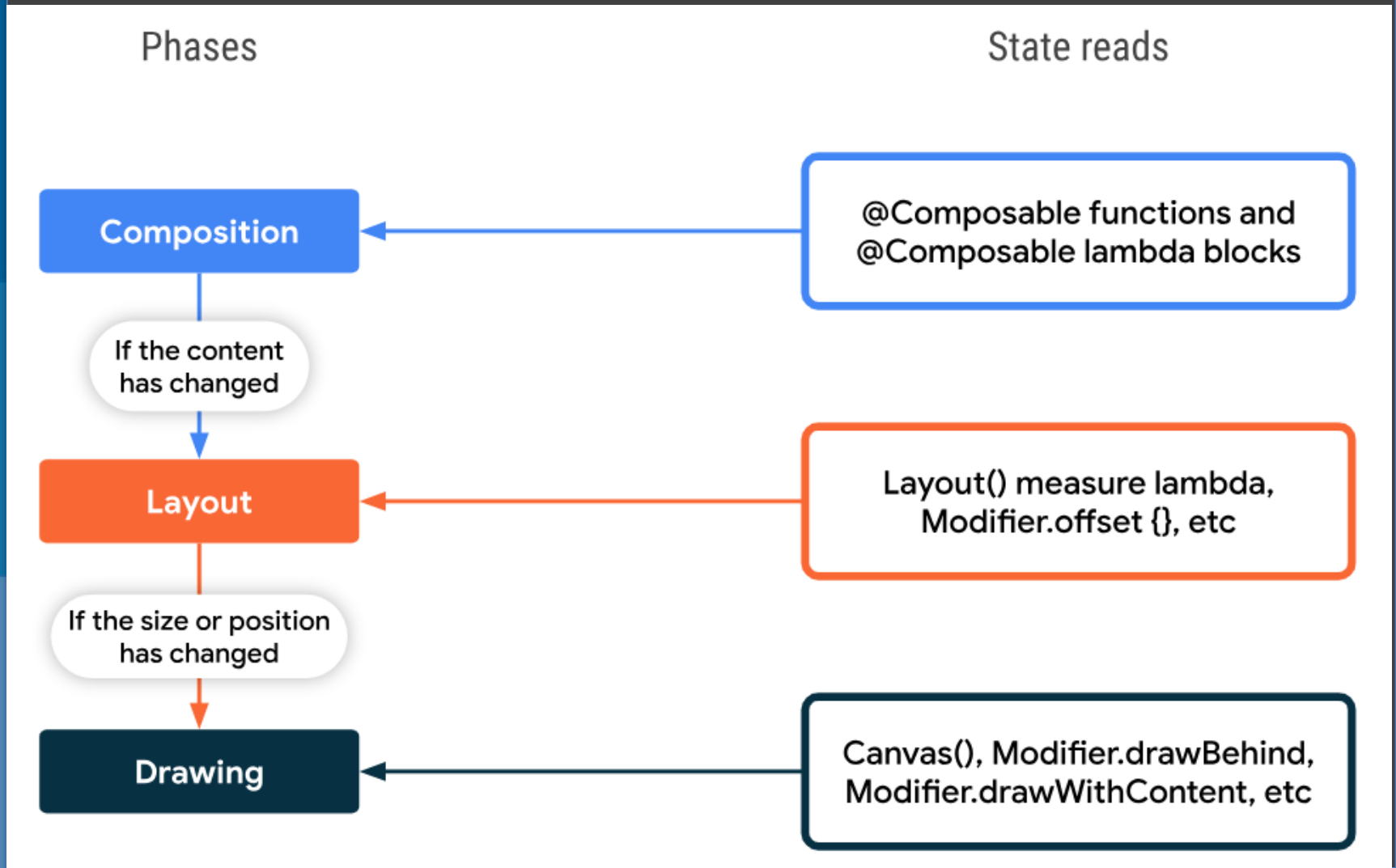



# Jetpack Compose - Fazy

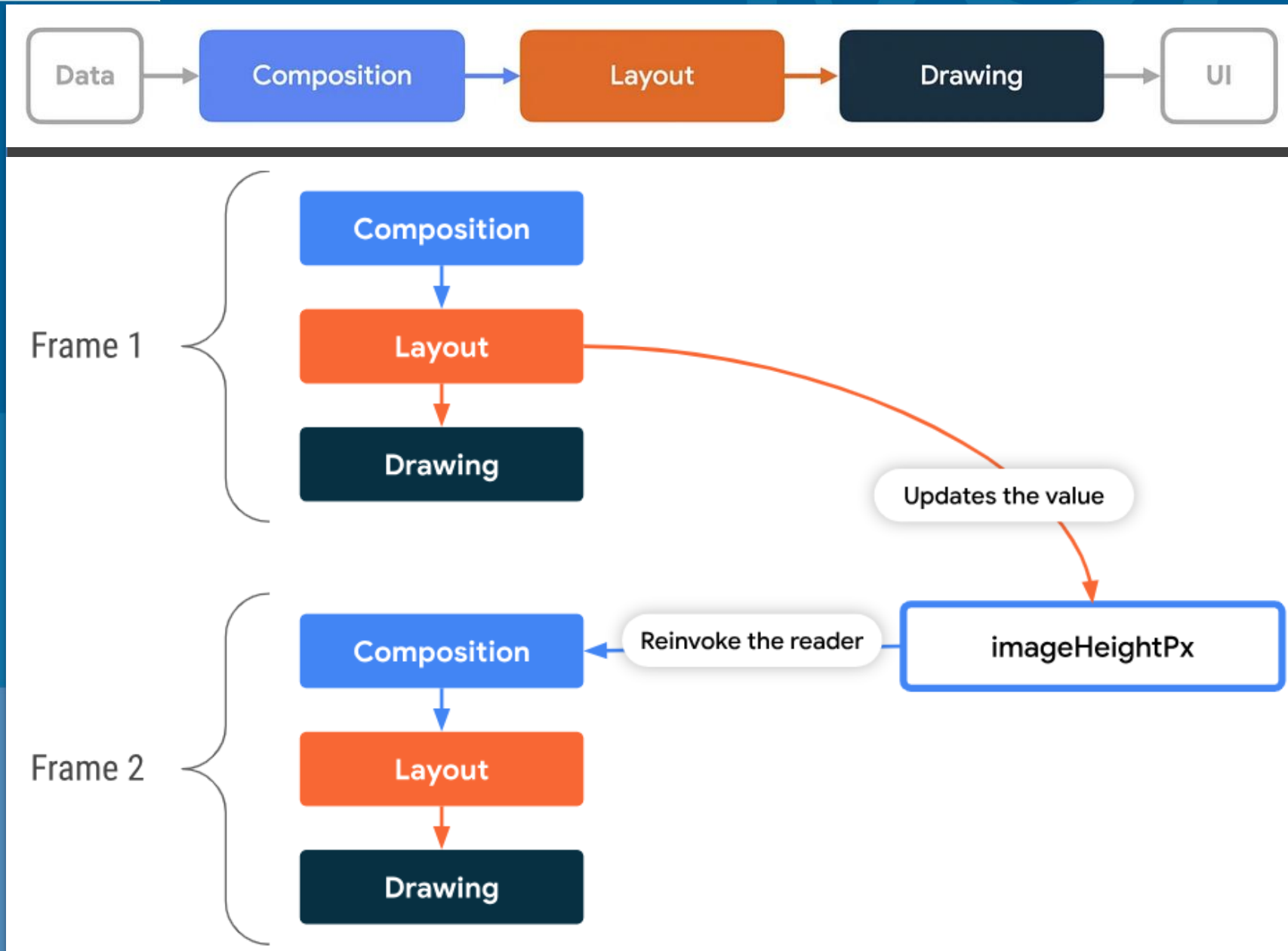


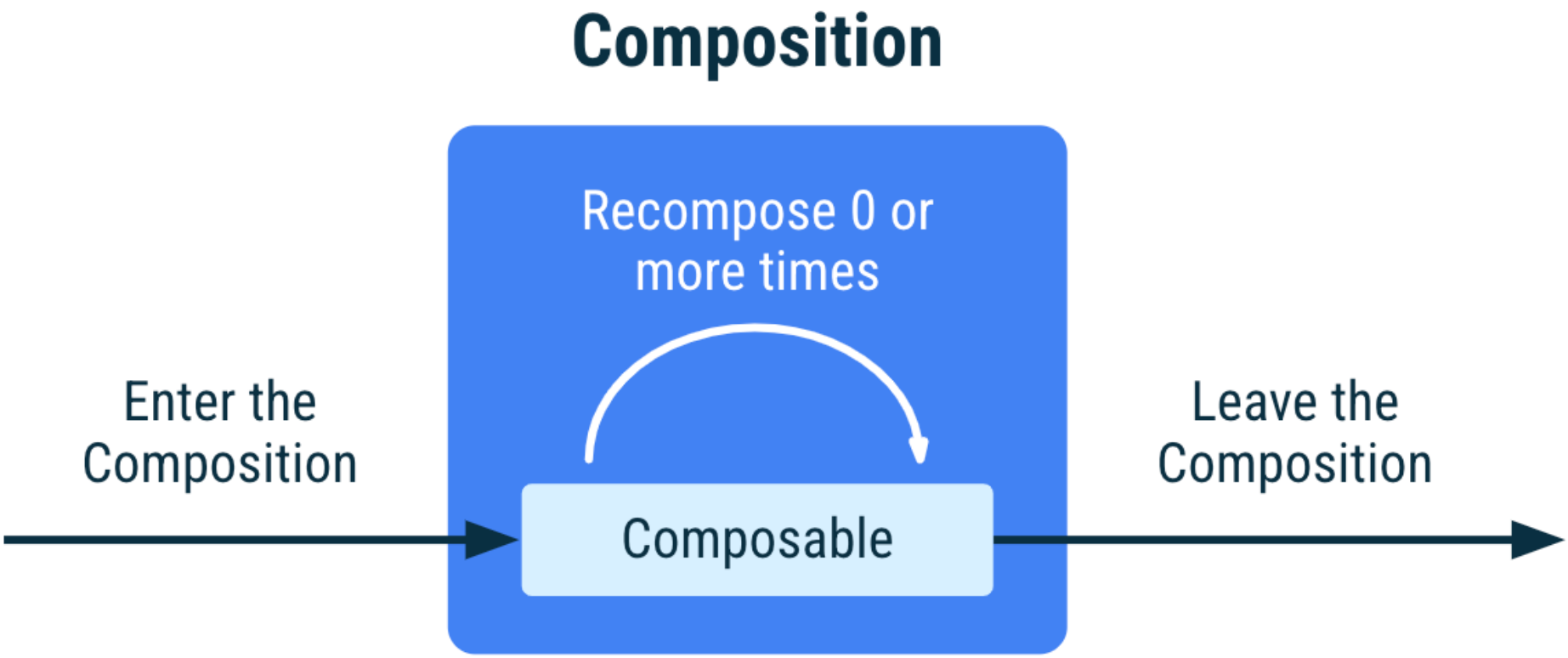
# Jetpack Compose - Fazy



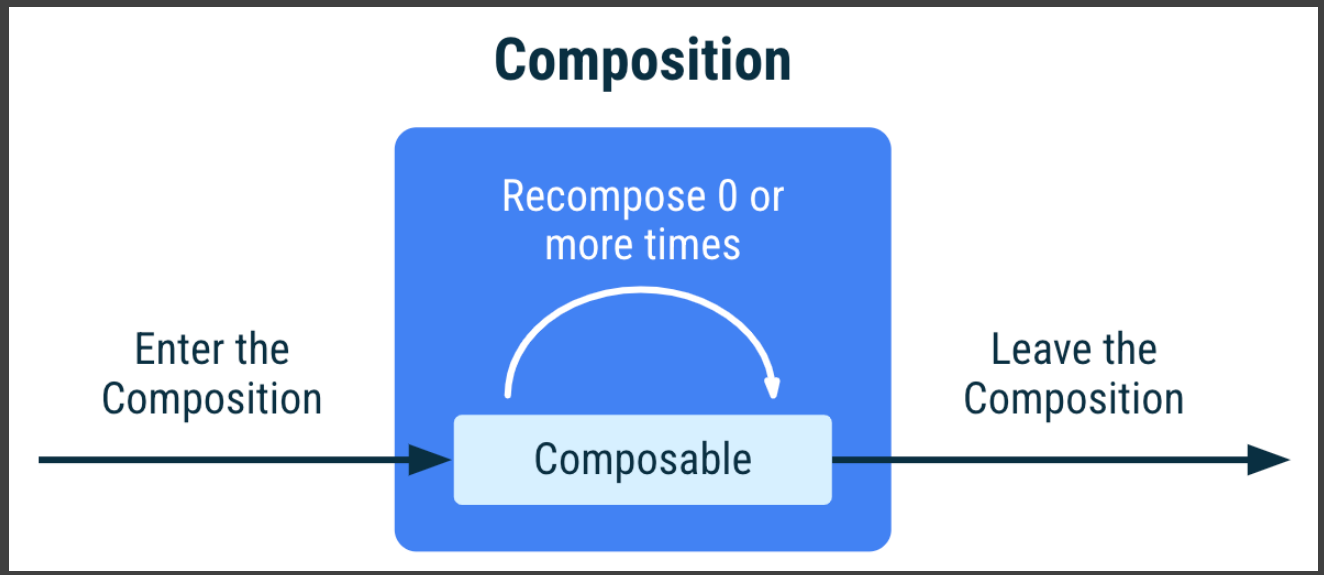
# Jetpack Compose - Fazy



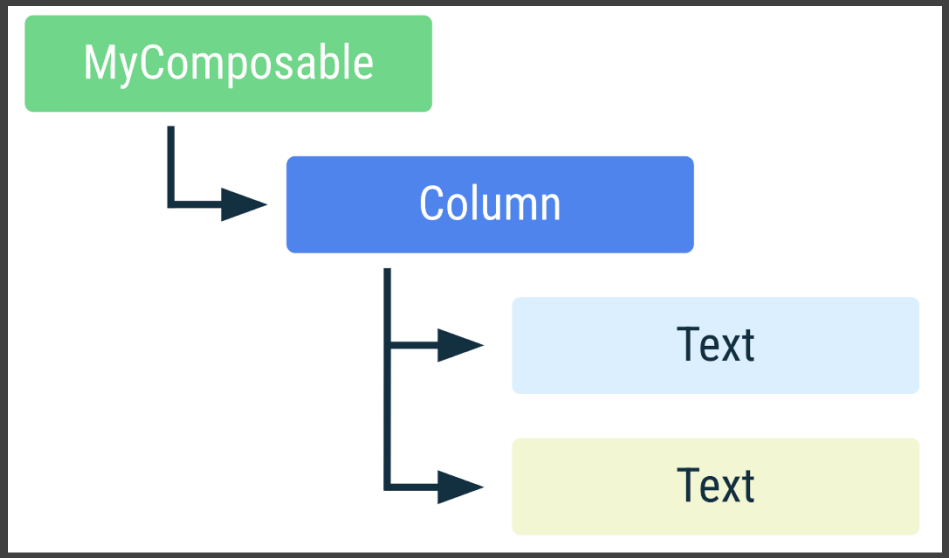
# State


Composition
Recompose 0 or more times
Enter the Composition
Composable
Leave the Composition

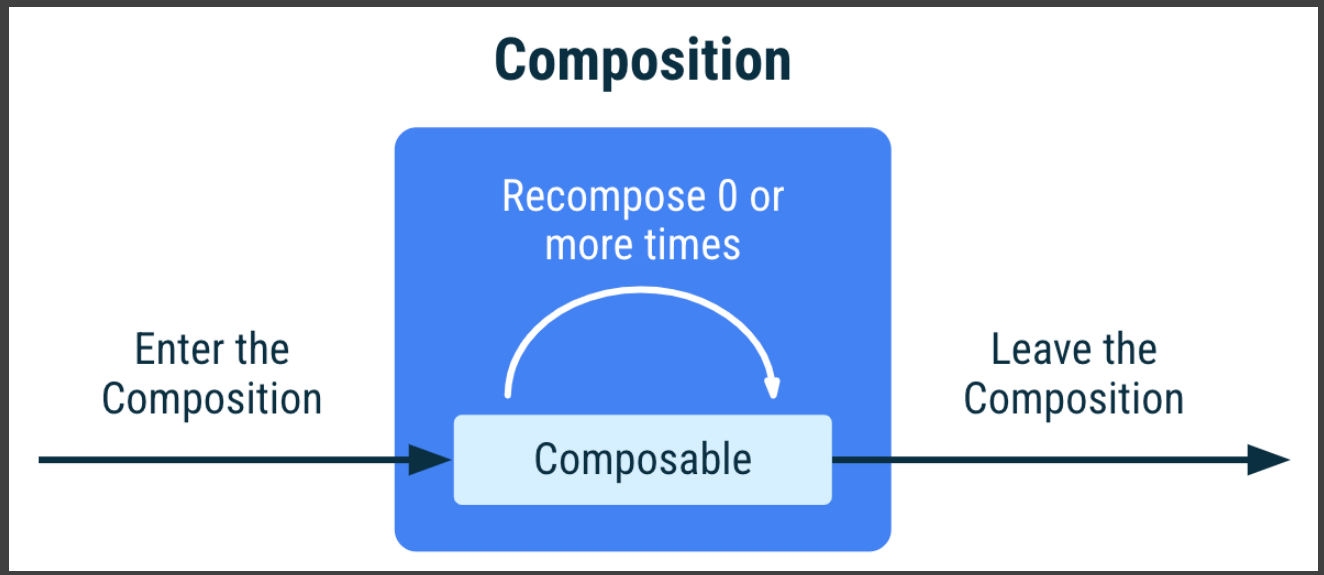
# State

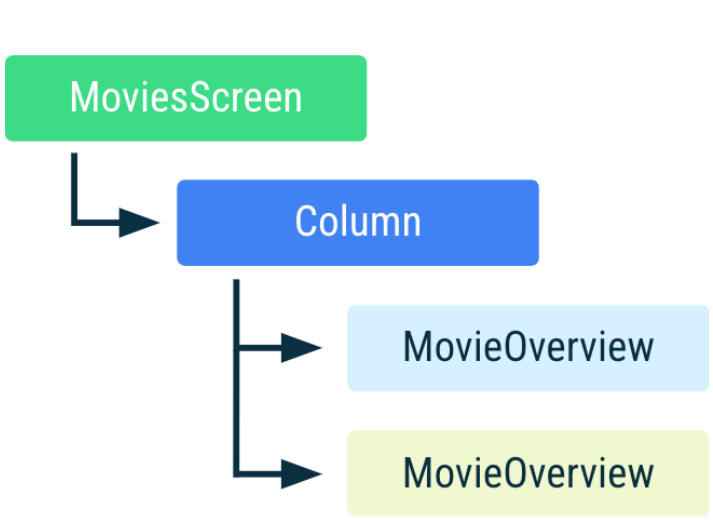



```
@Composable
fun MyComposable() {
    Column {
        Text("Hello")
        Text("World")
    }
}
```

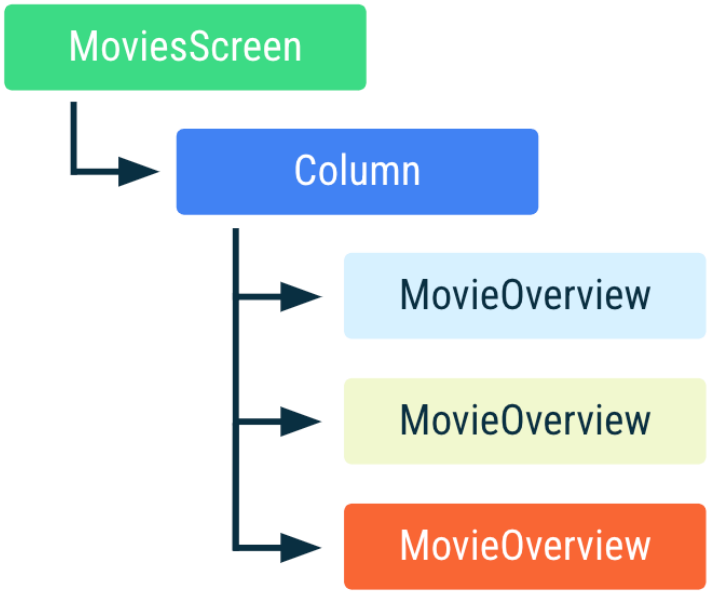


# State



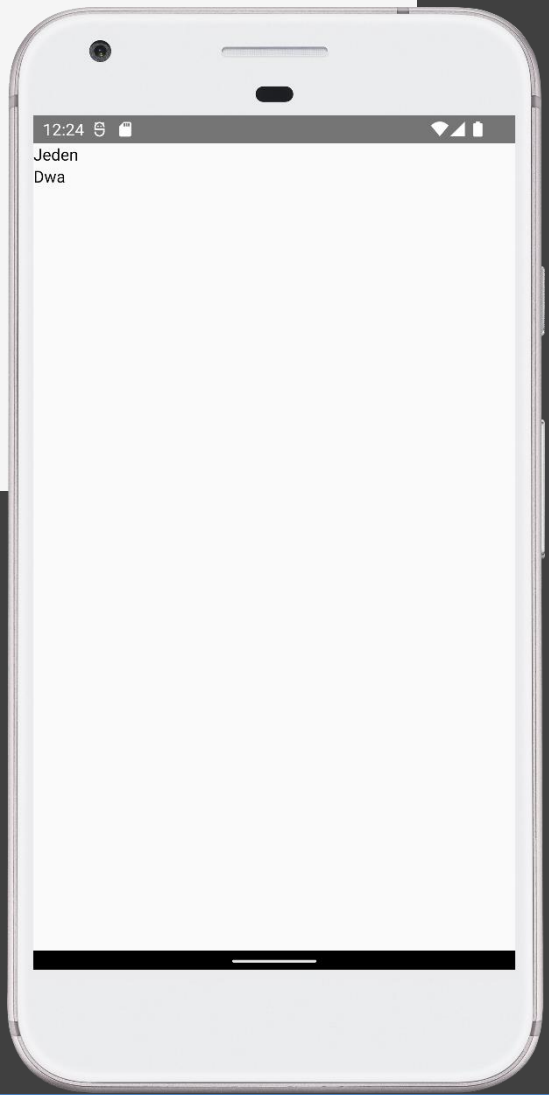

Recomposition (new element added to the bottom of the list)



Uniwersytet Wrocławski

# Jetpack Compose

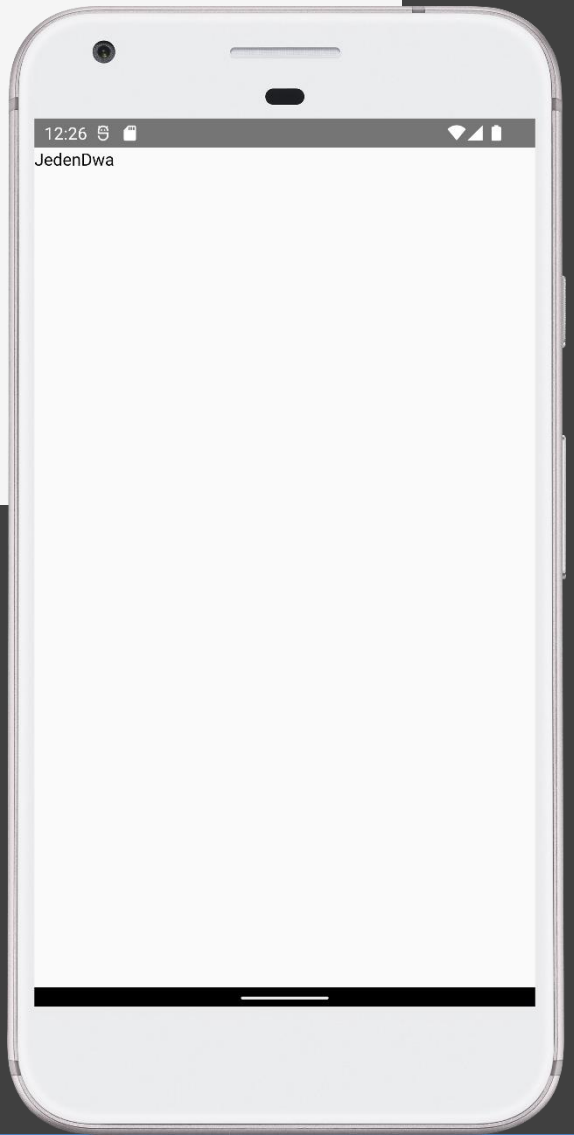
```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContent {
        Column {
            Text(text = "Jeden")
            Text(text = "Dwa")
        }
    }
}
```



Uniwersytet Wrocławski

# Jetpack Compose

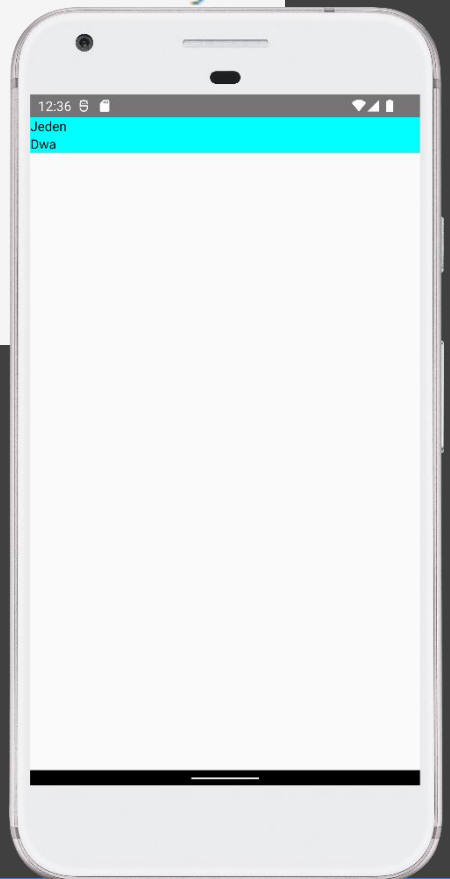
```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContent {
        Row {
            Text(text = "Jeden")
            Text(text = "Dwa")
        }
    }
}
```



Uniwersytet Wrocławski

# 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")
        }
    }
}
```



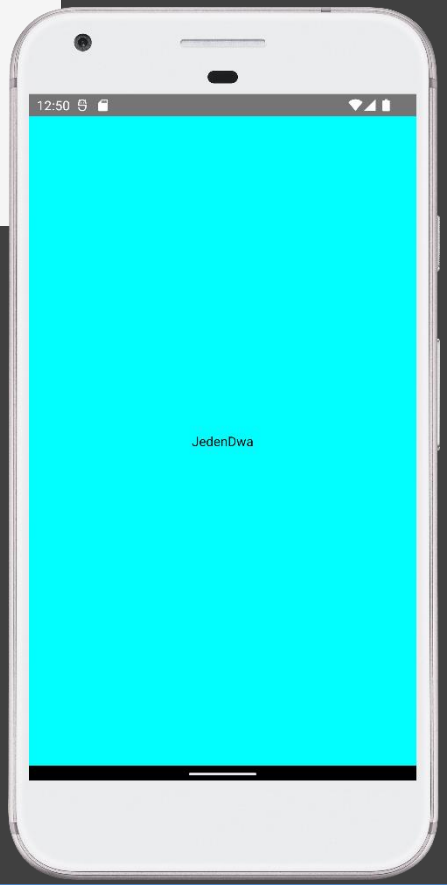
Uniwersytet Wrocławski

# 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")
        }
    }
}
```

# Jetpack Compose

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

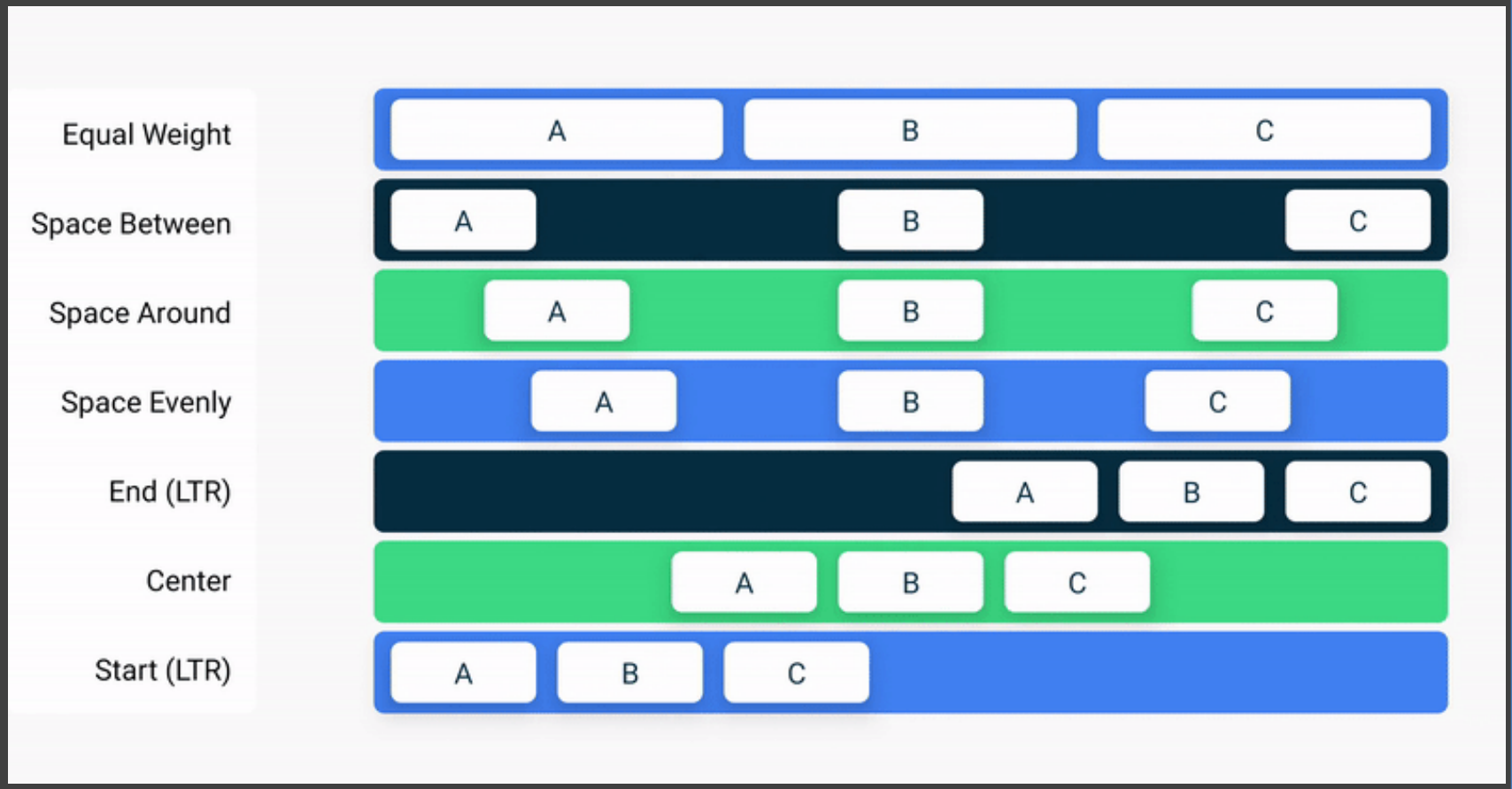


# Jetpack Compose

```
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

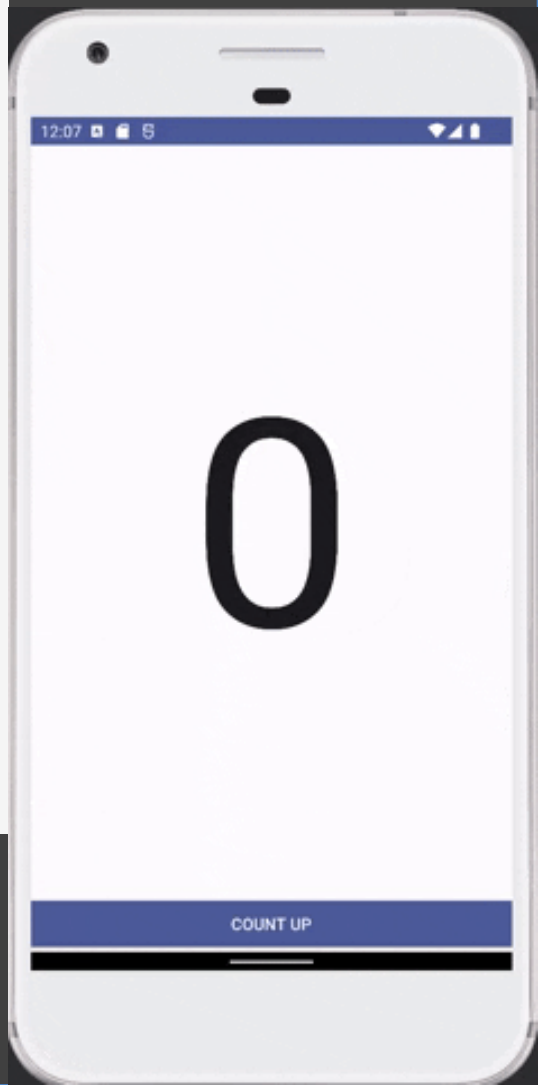
# Jetpack Compose



Uniwersytet Wrocławski

# State

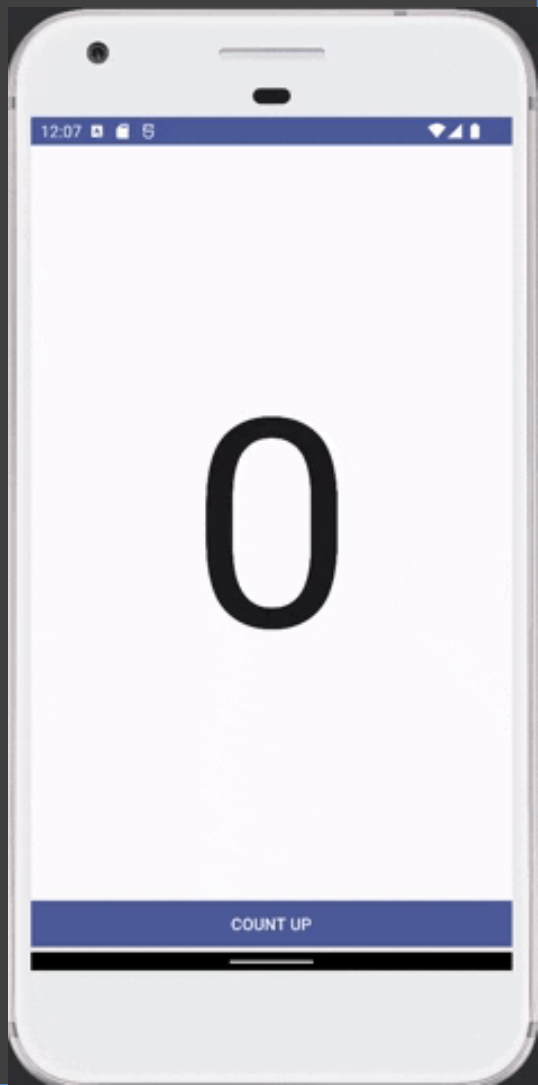
```
@Composable
fun CounterExample() {
    Column(
        verticalArrangement = Arrangement.Center,
        horizontalAlignment = Alignment.CenterHorizontally,
        modifier = Modifier.fillMaxSize()
    ) {
        Spacer(modifier = Modifier.weight(0.3f))
        Text(
            text = "test",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



Uniwersytet Wrocławski

# State

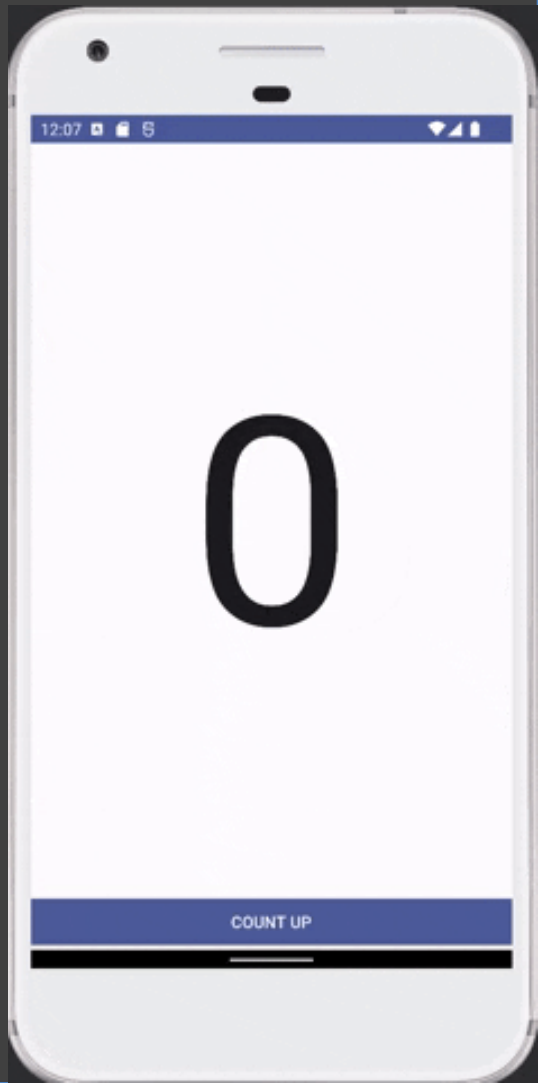
```
@Composable
fun CounterExample() {
    Column(
        verticalArrangement = Arrangement.Center,
        horizontalAlignment = Alignment.CenterHorizontally,
        modifier = Modifier.fillMaxSize()
    ) {

        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")
        }
    }
}
```

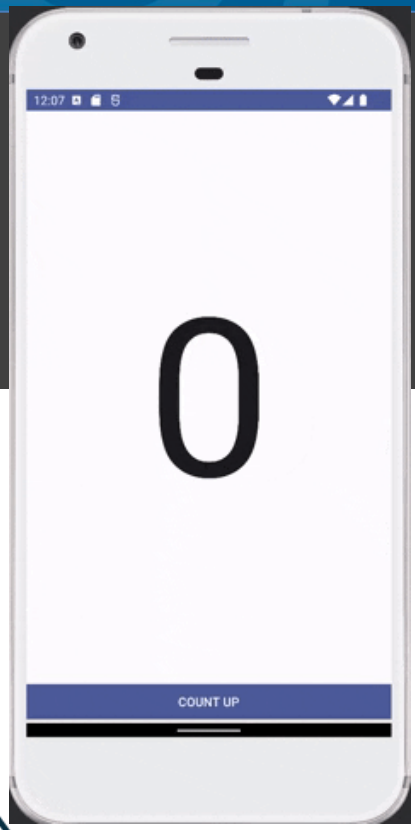
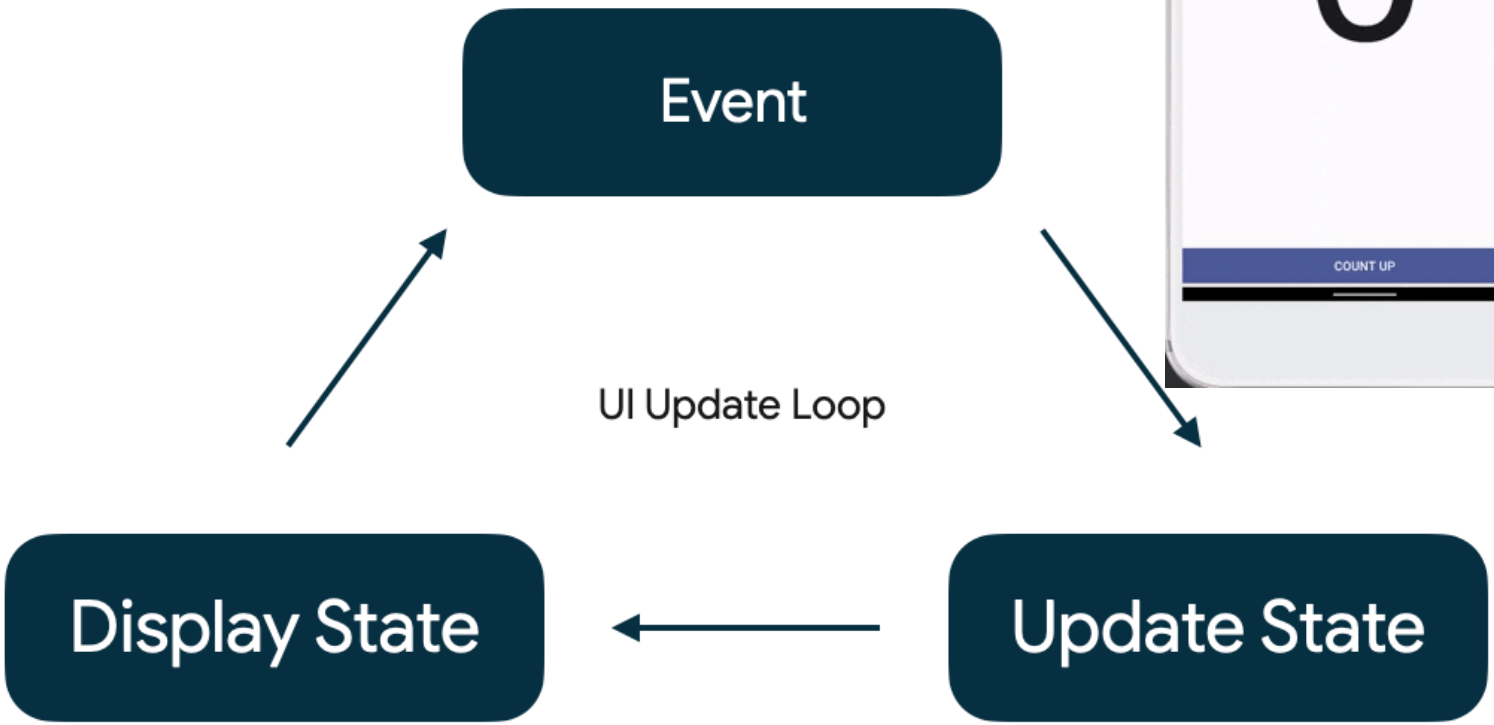


Uniwersytet Wrocławski

# 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)
}
```