Workshop: Auth & Navigation

Applikationsudvikling

NIFR 2024

Reading vs. Writing code

What is the purpose of today?
Asking questions & understanding

Setting up firebase

Android Auth

- Fork / Clone the codebase
  - https://github.com/nicklasdean/compose-firebase-auth
- Follow step 1-4: Add Firebase: using the Firebase console
  - https://firebase.google.com/docs/android/setup
    - Register application on firebase
    - Download google.services.json
    - Try to register a user in the application

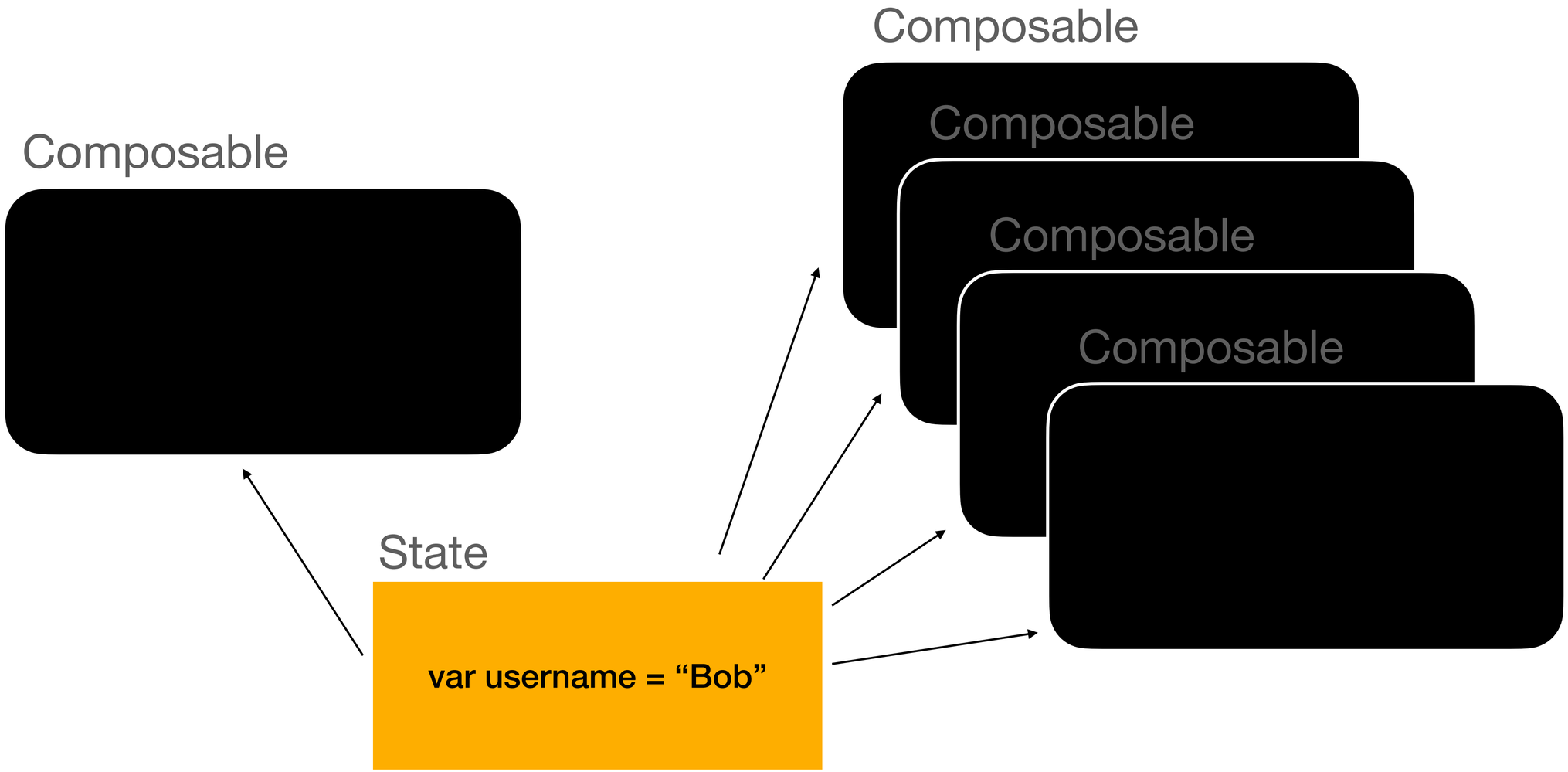
Agenda

Applikationsudvikling

- Setting firebase up for Android
- Recap: State & ViewModel
- Navigation in the Auth codebase
- User Registration in the Auth codebase

Resolve

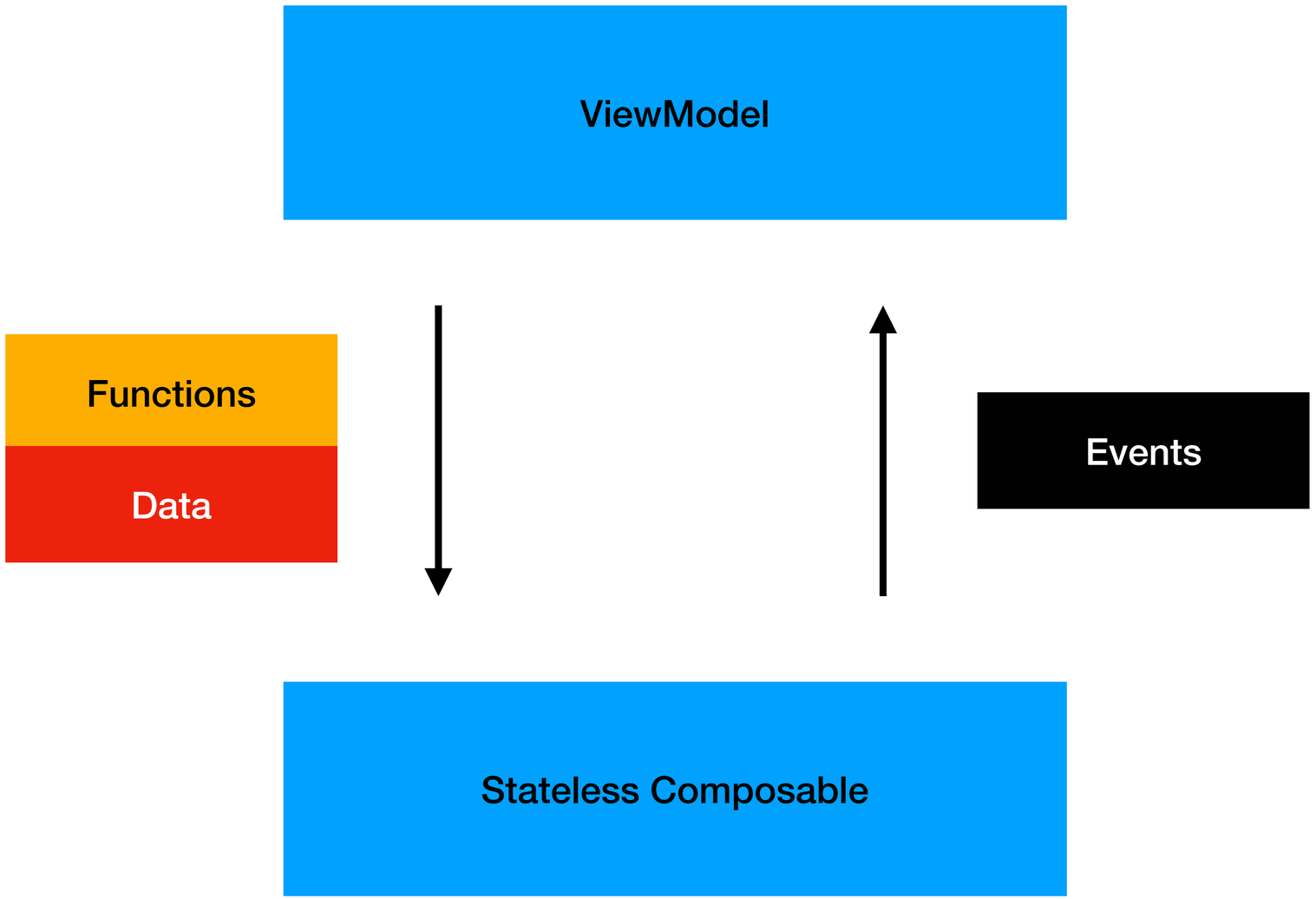
State hoisting

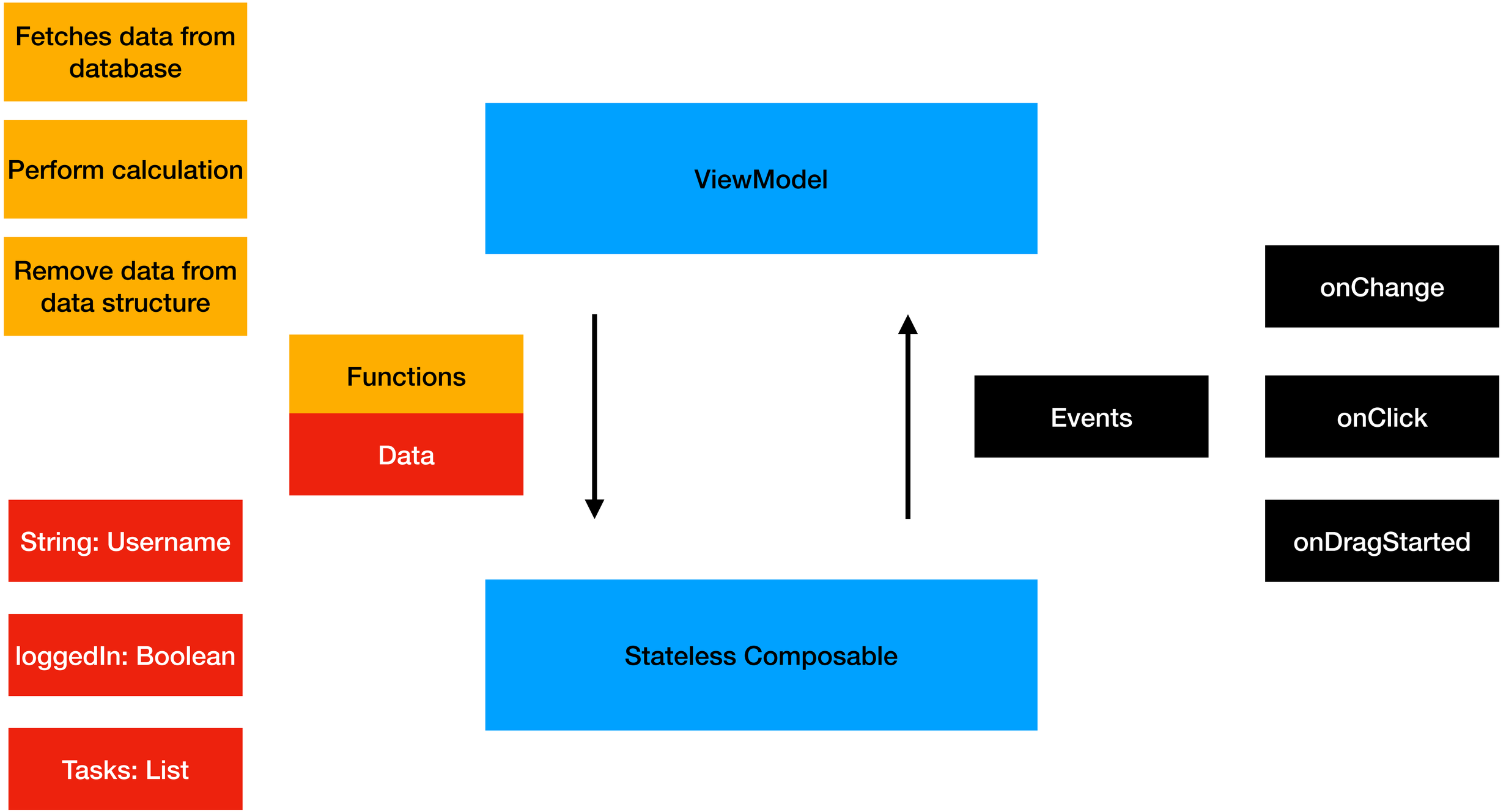


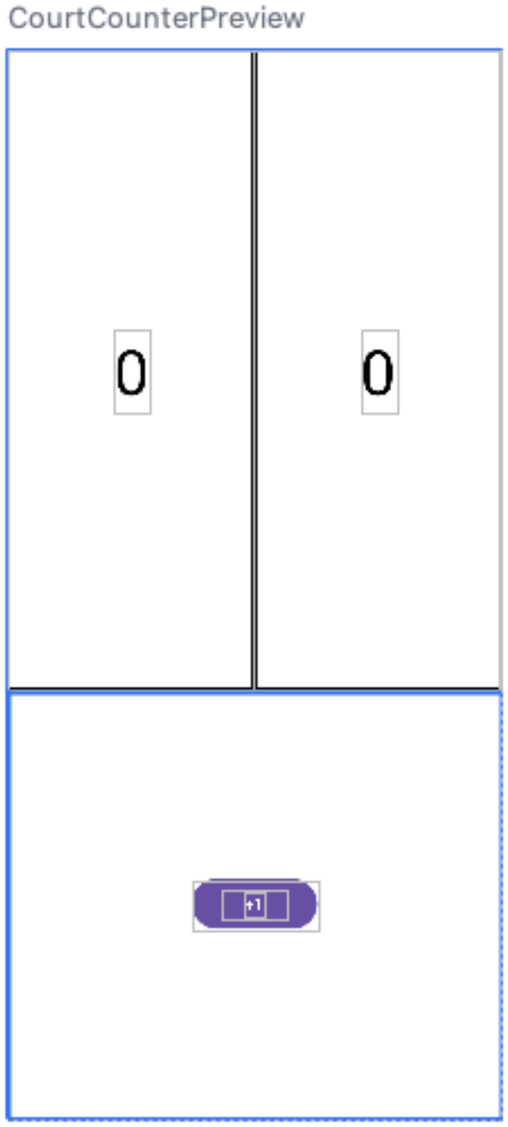
State in the ViewModel

```
class CounterViewModel : ViewModel() {
    var count: Int by mutableIntStateOf(0);

    fun incrementScore(): Unit{
        count ++;
    }
}
```

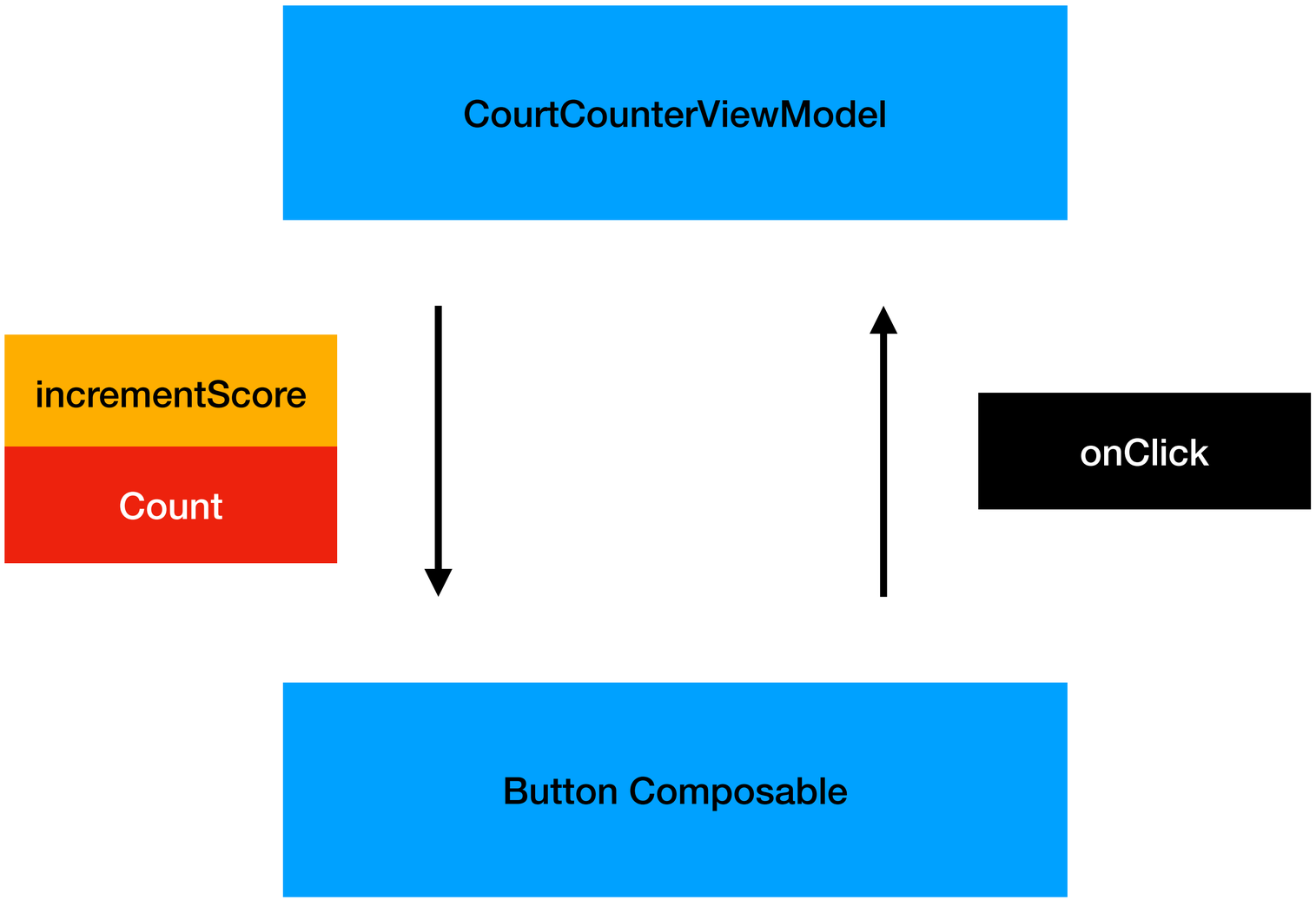
Functionality in the ViewModel


ViewModel
Functions
Data
Events
Stateless Composable


Fetches data from database
Perform calculation
Remove data from data structure
ViewModel
onChange
Functions
Data
Events
onClick
onDragStarted
String: Username
loggedIn: Boolean
Tasks: List
Stateless Composable



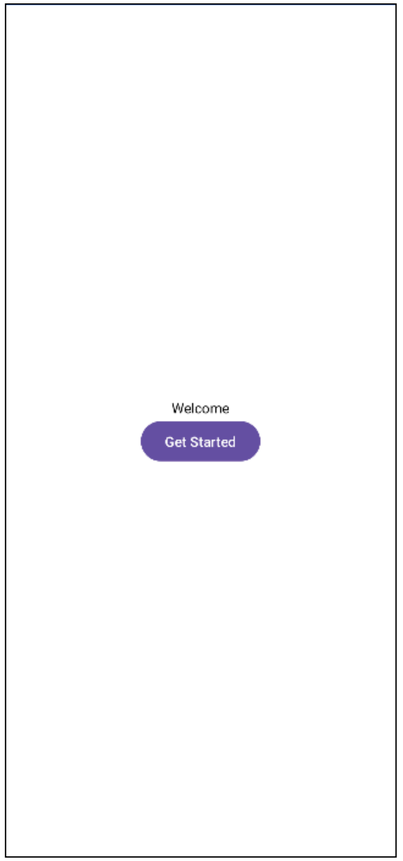

```
Row(
    modifier = Modifier
        .fillMaxWidth()
        .fillMaxHeight(),
    verticalAlignment = Alignment.CenterVertically,
    horizontalArrangement = Arrangement.SpaceEvenly,

    ) { this: RowScope
    Button(modifier = Modifier.width(100.dp),
        onClick = {
            myCounterViewModel.incrementScore()
        }) { this: RowScope
        Text( text: "+1");
    }
}
```



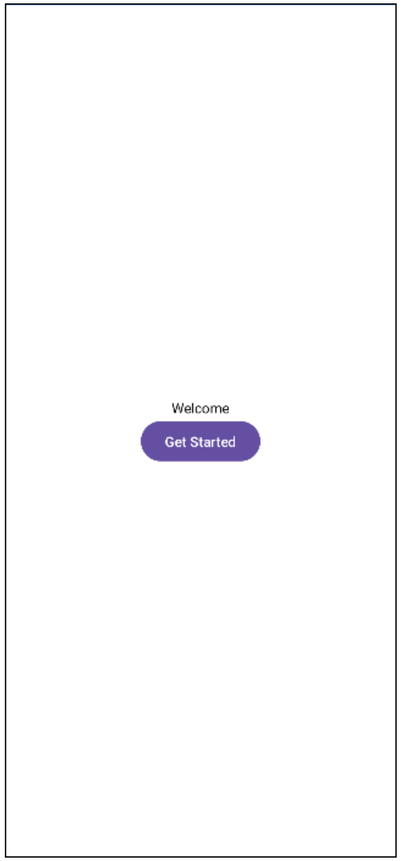

```
class CounterViewModel : ViewModel() {
    var count: Int by mutableIntStateOf(0);

    fun incrementScore(): Unit{
        count ++;
    }
}
```

```
Row(
    modifier = Modifier
        .fillMaxWidth()
        .fillMaxHeight(),
    verticalAlignment = Alignment.CenterVertically,
    horizontalArrangement = Arrangement.SpaceEvenly,

    ) { this: RowScope
    Button(modifier = Modifier.width(100.dp),
        onClick = {
            myCounterViewModel.incrementScore()
        }) { this: RowScope
        Text( text: "+1");
    }
}
```

Navigation in the case application




```
@Composable
fun Welcome(getStarted: () -> Unit) {
    Box(
        Modifier.fillMaxSize(),
        contentAlignment = Alignment.Center
    ) {
        Column(
            horizontalAlignment = Alignment.CenterHorizontally
        ) {
            Text(text = "Welcome")
            Button(onClick = getStarted) {
                Text("Get Started")
            }
        }

    }
}
```




```
@Composable
fun Welcome(getStarted: () -> Unit) {
    Box(
        Modifier.fillMaxSize(),
        contentAlignment = Alignment.Center
    ) {
        Column(
            horizontalAlignment = Alignment.CenterHorizontally
        ) {
            Text(text = "Welcome")
            Button(onClick = getStarted) {
                Text("Get Started")
            }
        }

    }
}
```

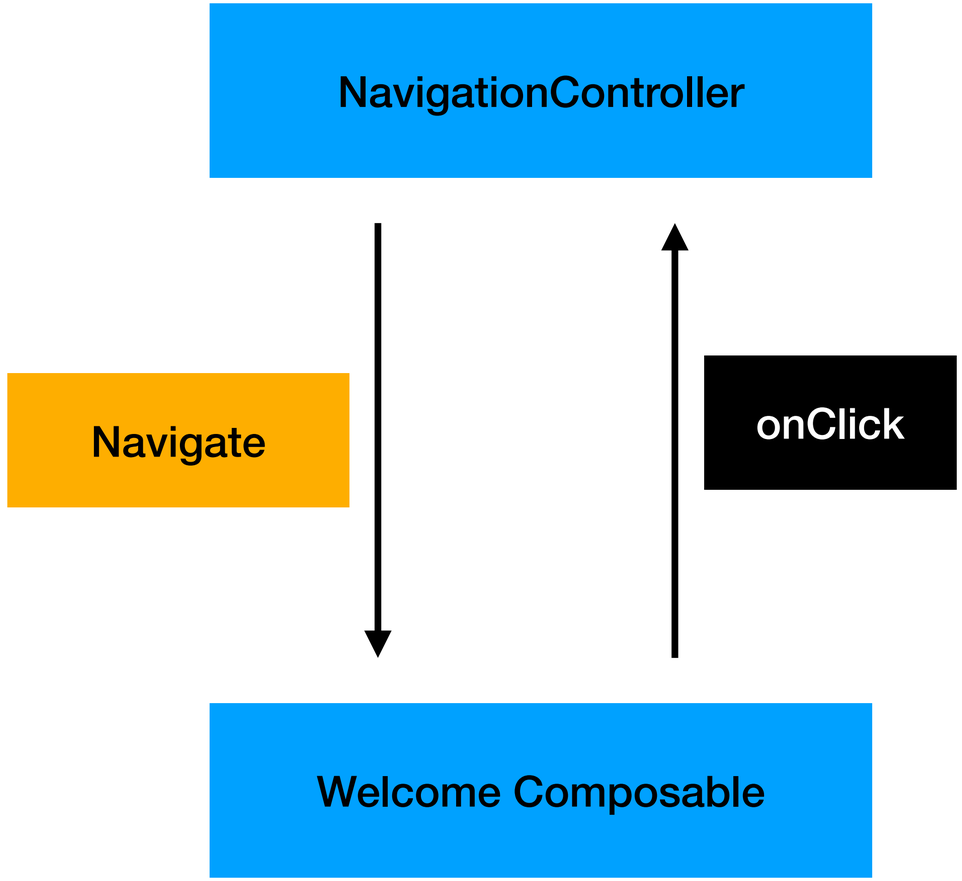
```
@Composable
fun Navigation() {
    val navController = rememberNavController()
    NavHost(navController = navController, startDestination = "welcome") {
        composable("welcome") {
            Welcome(getStarted = { navController.navigate("register") })
        }
        composable("register") {
            Login(onSuccess = { navController.navigate("success") })
        }
        composable("success") {
            Success()
        }
    }
}
```

```
@Composable
fun Navigation() {
    val navController = rememberNavController()
    NavHost(navController = navController, startDestination = "welcome") {
        composable("welcome") {
            Welcome(getStarted = { navController.navigate("register") })
        }
        composable("register") {
            Login(onSuccess = { navController.navigate("success") })
        }
        composable("success") {
            Success()
        }
    }
}
```

Navigation in the demo app

Jetpack Compose

- All navigation comes from the Navigation Composable
- If navigation needs to happen - we need to push the function down the stack
- We push the function down the stack by passing the navigate function to the Composable as a parameter



Example (Welcome):
User Registration in the sample app