

Compose - Estado

PDM - Programação para Dispositivos Móveis

Paulo Pereira
paulo.pereira@isel.pt



Modelo simplificado



- UI em função dos dados
- Recomposição automática quando os dados mudam



Gestão de Estado em Compose

- É acerca de decidir onde colocamos os dados (qual a autoridade)
 - A que se referem?
 - Qual a sua origem?
 - O que pretendemos fazer-lhes?
- Resumidamente:
 - Estado de apresentação 
 - Estado de aplicação 



Exemplo (1)

Email

Password

Entrar

1

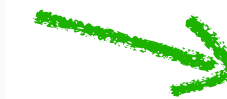


Email

Password


Entrar

2

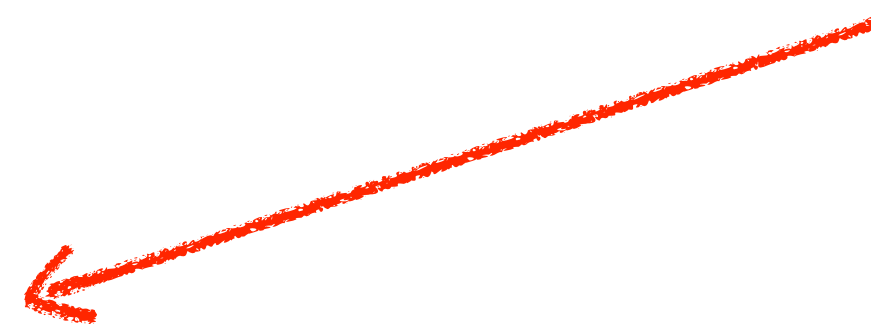


Email

Password

 Entrar

3



Email

Password

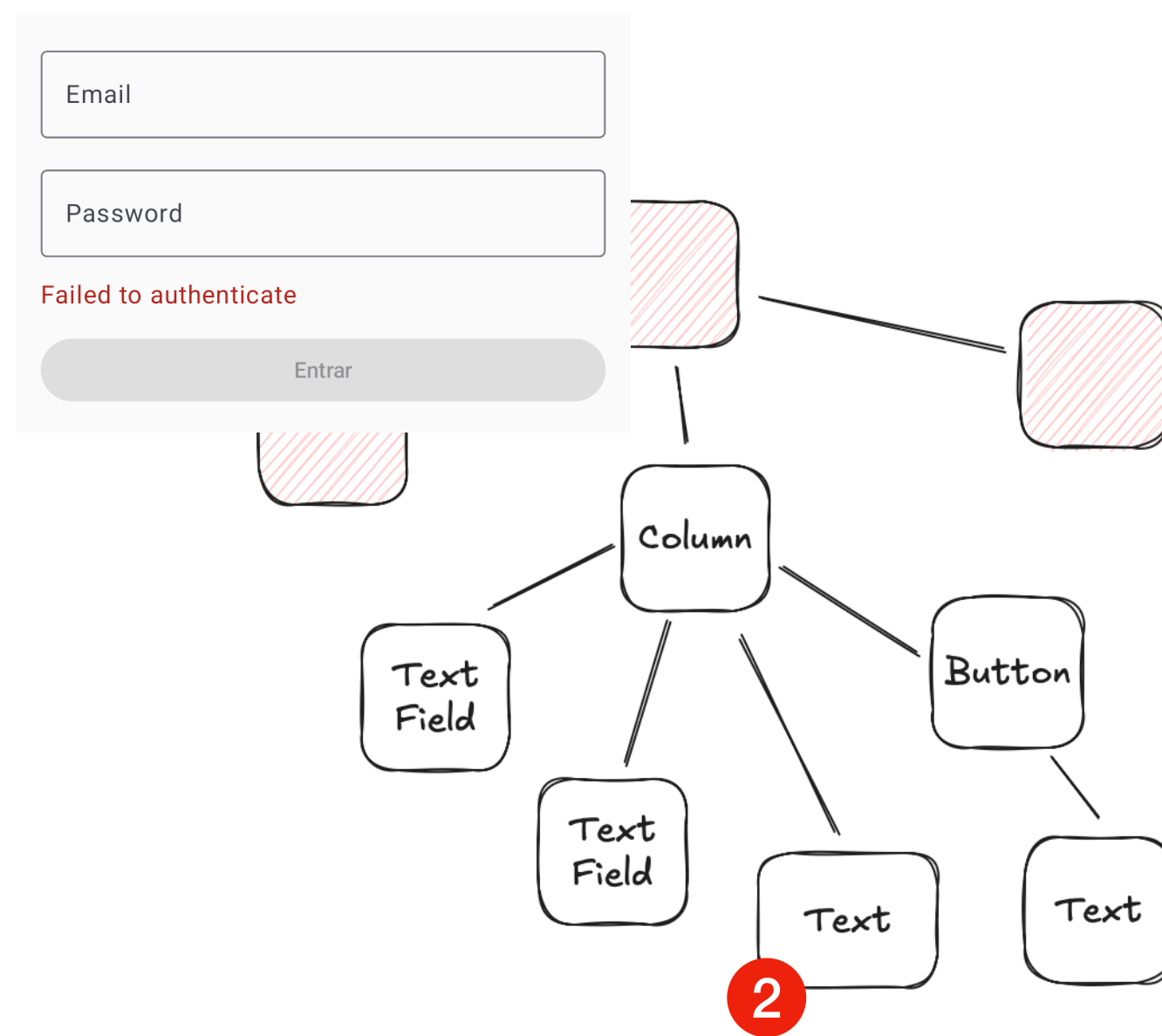
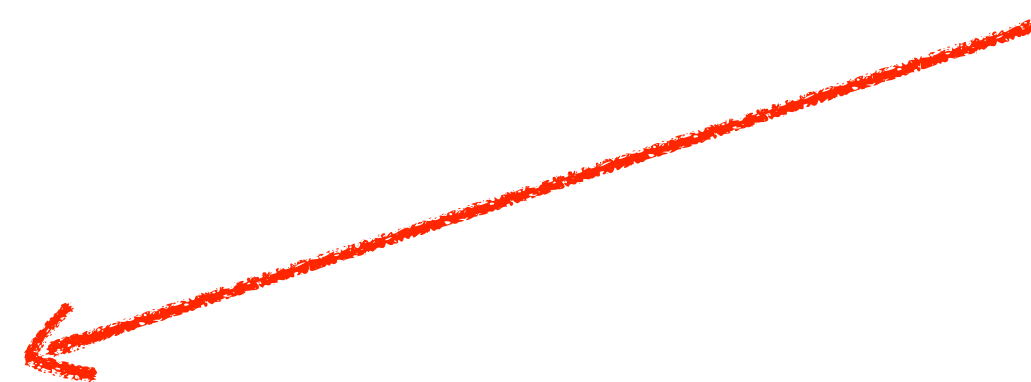
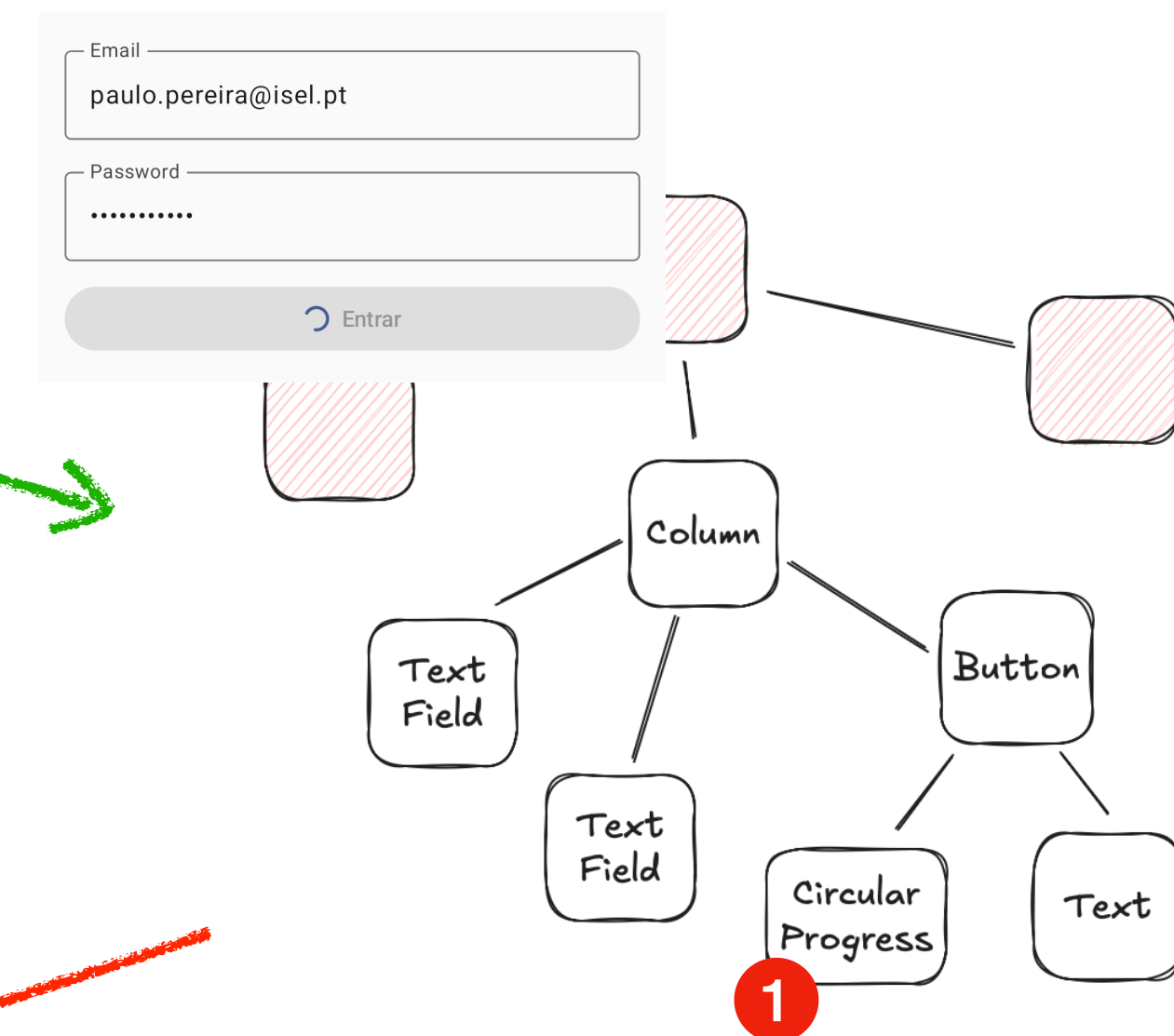
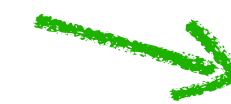
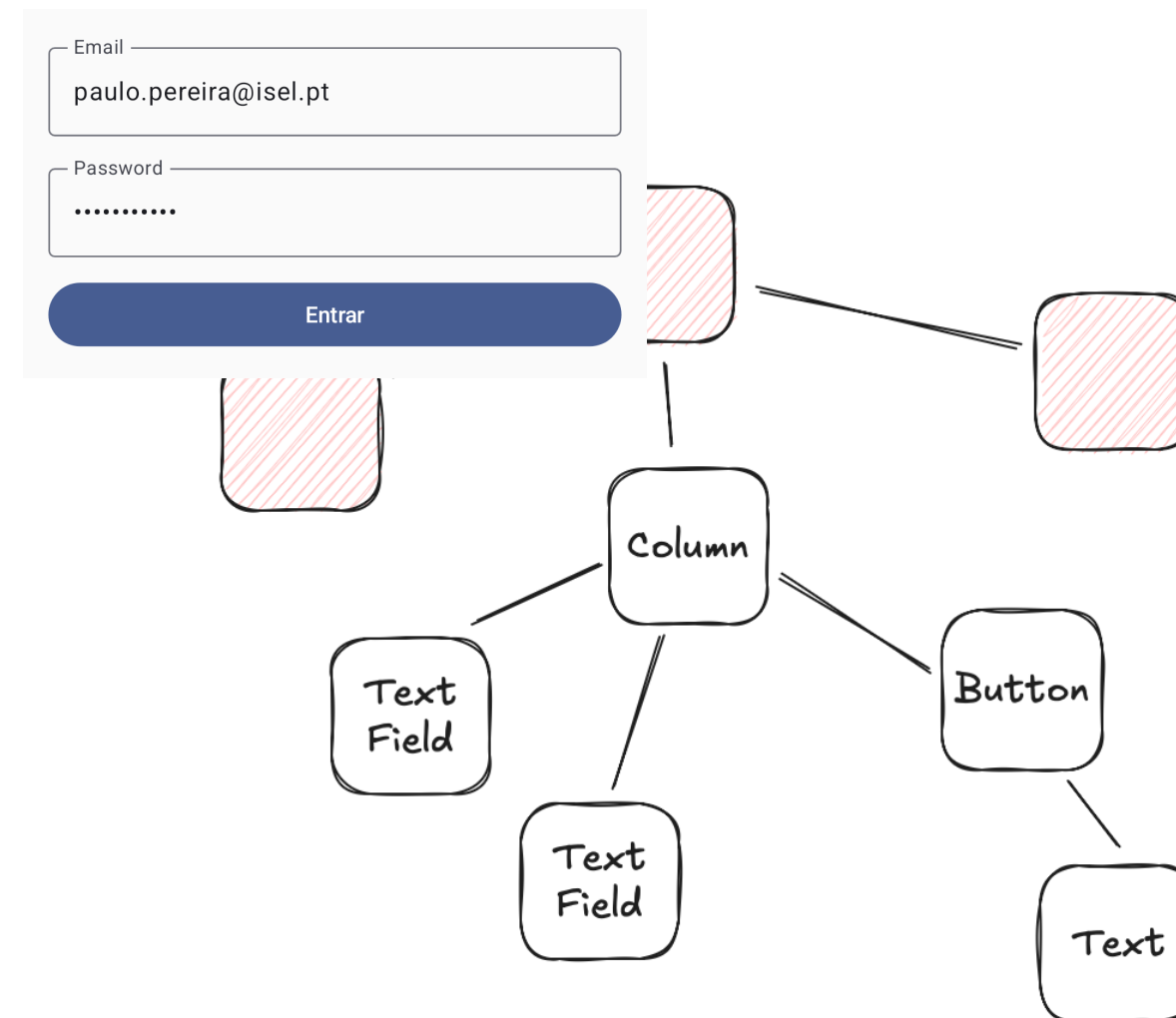
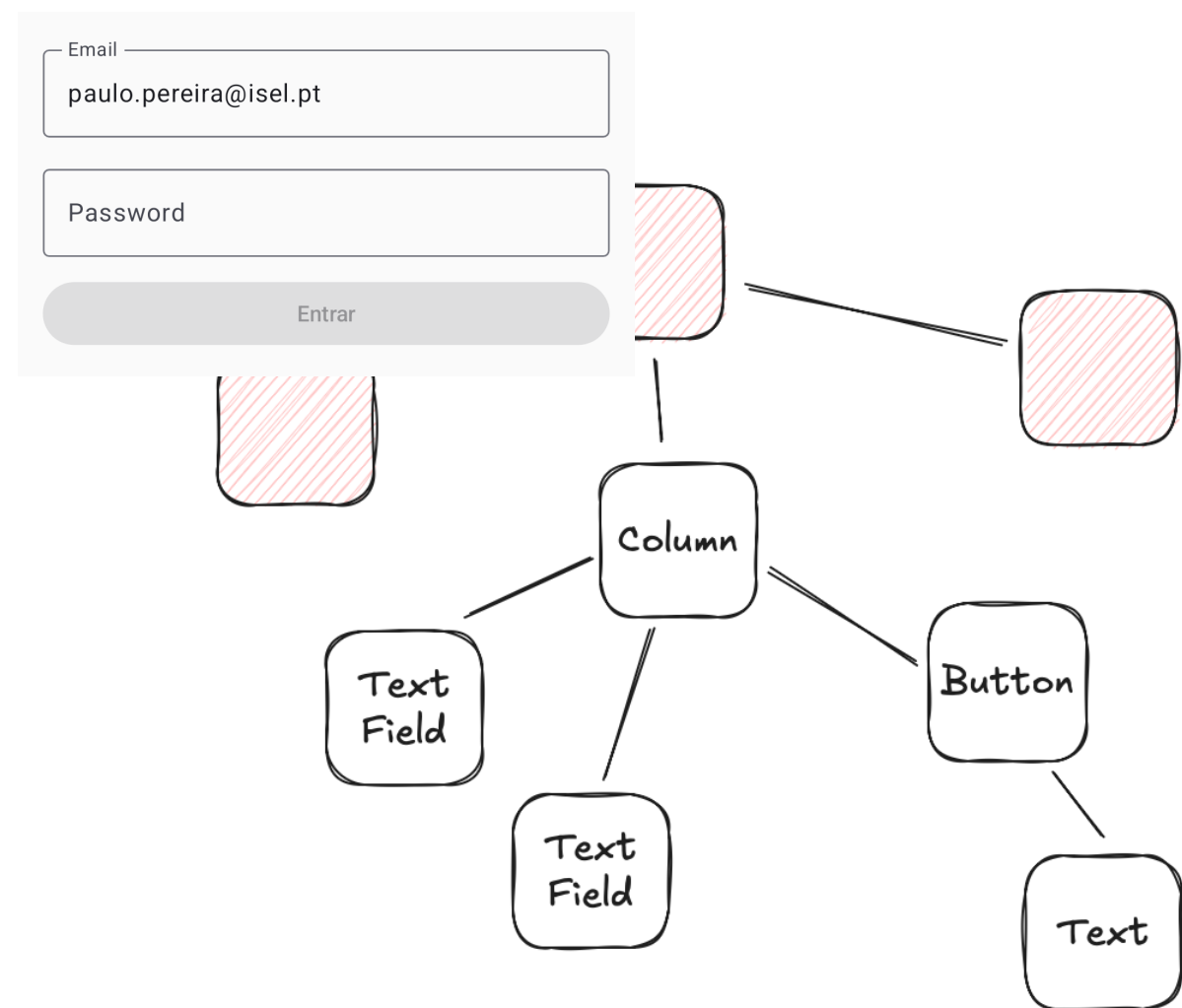
Failed to authenticate

Entrar

4



Exemplo (2)

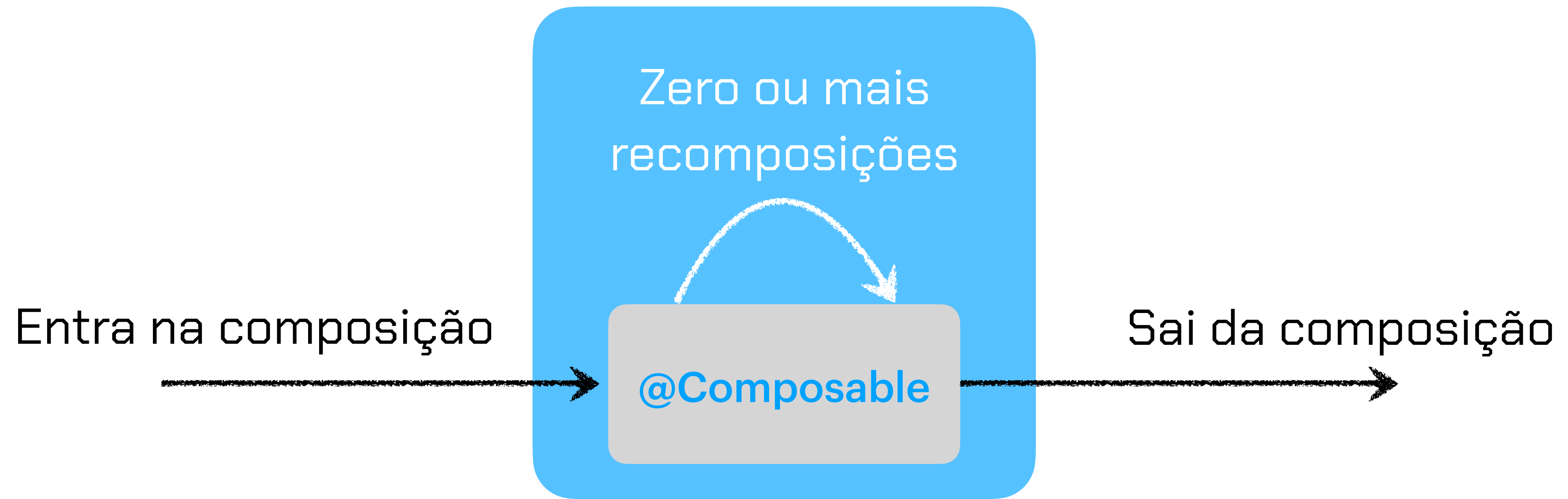


Exemplo (3)

```
@Composable
fun LoginFormStateless(
    loading: Boolean,
    error: String?,
    email: String,
    password: String,
    isValid: Boolean,
    onEmailChange: (String) -> Unit,
    onPasswordChange: (String) -> Unit,
    onLogin: (email: String, password: String) -> Unit
) {
    Column(...) {
        OutlinedTextField(
            value = email,
            onValueChange = onEmailChange,
        )
        OutlinedTextField(
            value = password,
            onValueChange = onPasswordChange,
        )
        if (!error.isNullOrEmpty()) {
            Text(error, color = MaterialTheme.colorScheme.error)
        }
        Button(
            onClick = { onLogin(email, password) },
            enabled = isValid && !loading,
        ) {
            if (loading) {
                CircularProgressIndicator(
                    Modifier.size(16.dp), strokeWidth = 2.dp
                )
            }
            Text("Entrar")
        }
    }
}
```



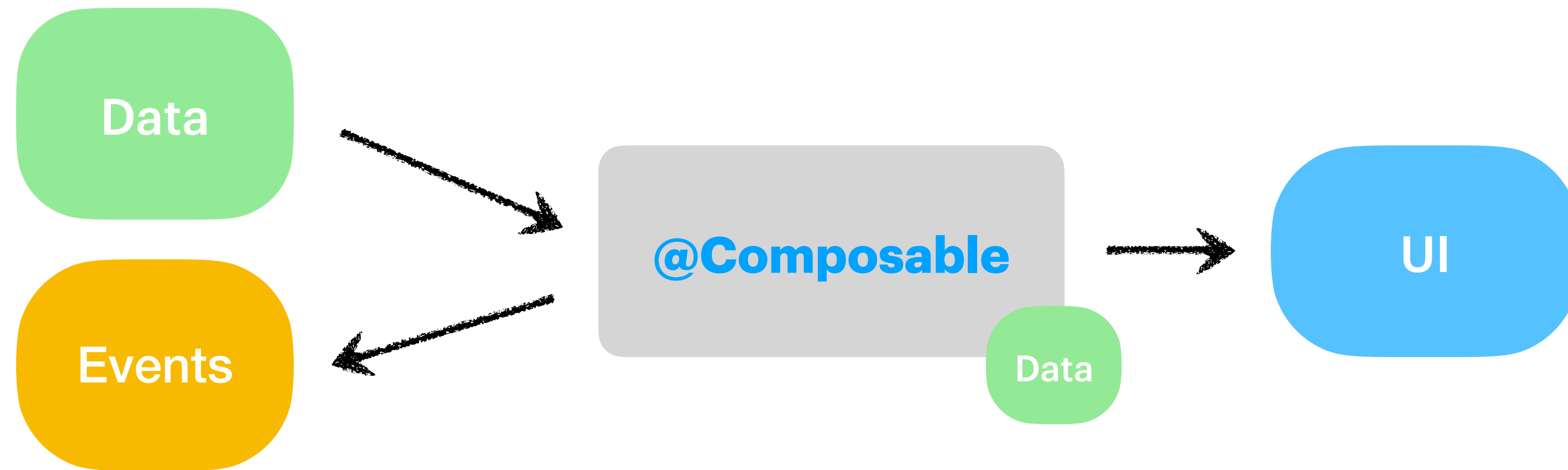
Ciclo de vida



- As recomposições ocorrem quando os dados mudam
- Desencadeadas por alterações em instâncias de `State<T>`



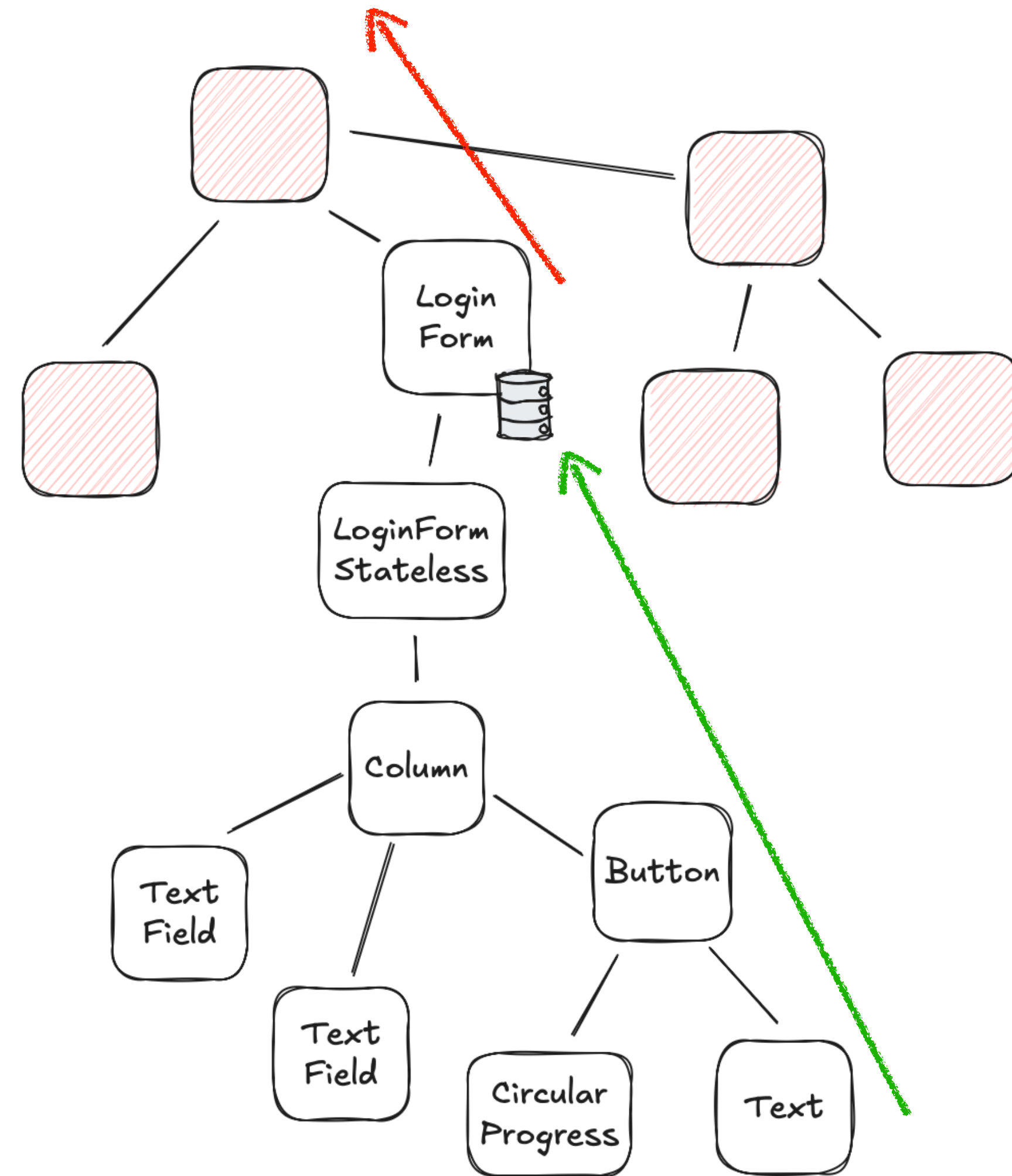
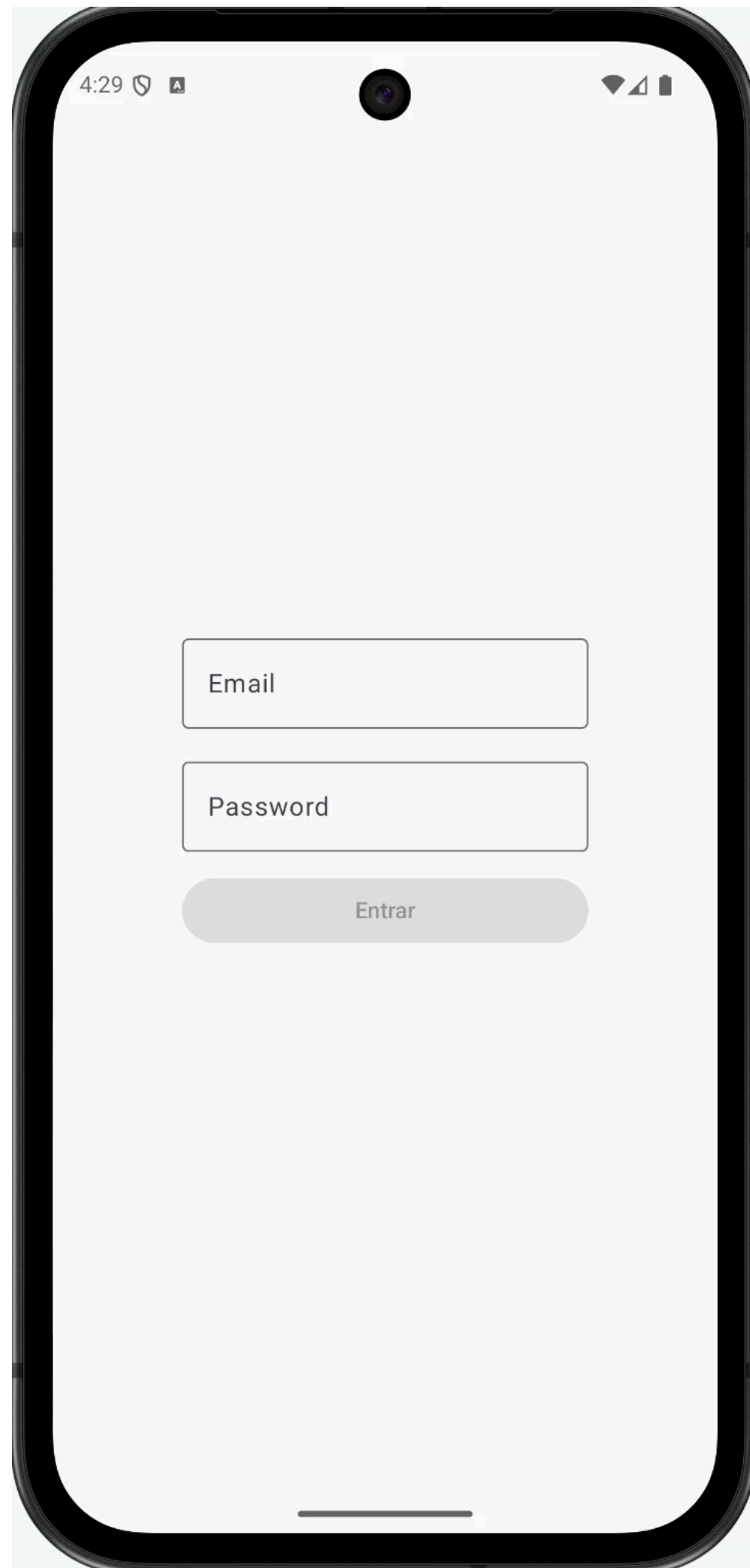
Modelo, revisitado



- UI em função dos dados (externos e internos)
- Recomposição automática quando os dados mudam
- Classificam-se como: **Stateless** e **Stateful**

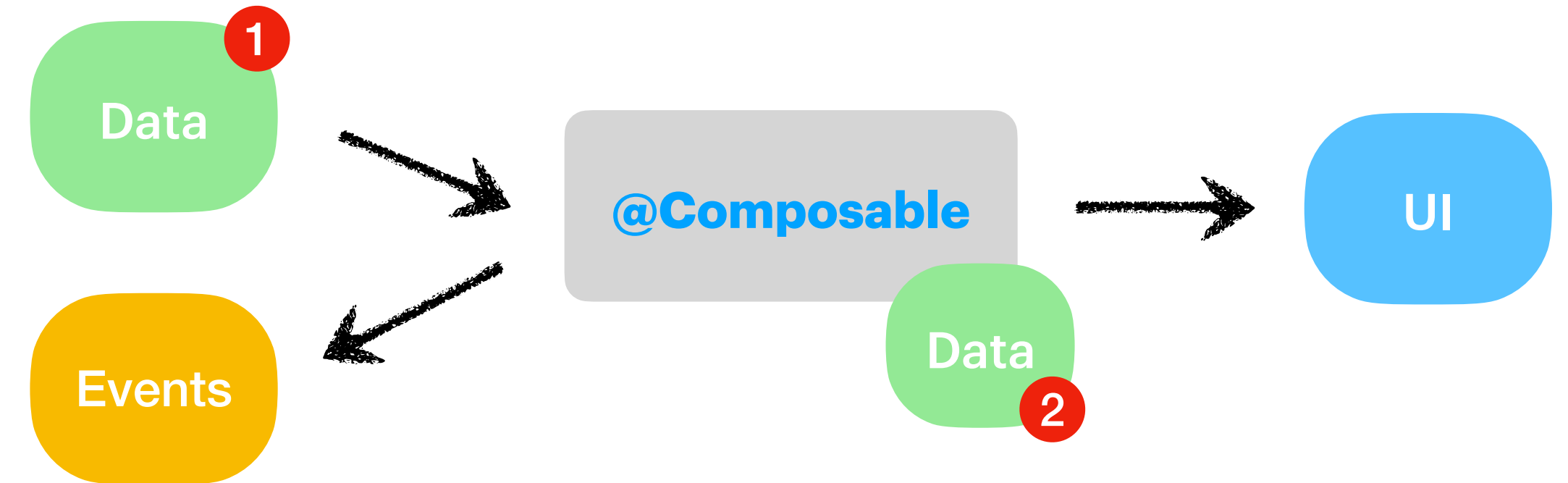


Exemplo (4)



Exemplo (5)

```
@Composable
fun LoginForm(
  1 loading: Boolean,
  error: String?,
  onLogin: (email: String, password: String) -> Unit,
  modifier: Modifier = Modifier
) {
  2 var email by remember { mutableStateOf("") }
  var password by remember { mutableStateOf("") }
  val isValid = email.isValidEmail() && password.isNotBlank()
  LoginFormStateless(
    loading = loading,
    error = error,
    email = email,
    password = password,
    isValid = isValid,
    onEmailChange = { email = it },
    onPasswordChange = { password = it },
    onLogin = onLogin,
    modifier
  )
}
```

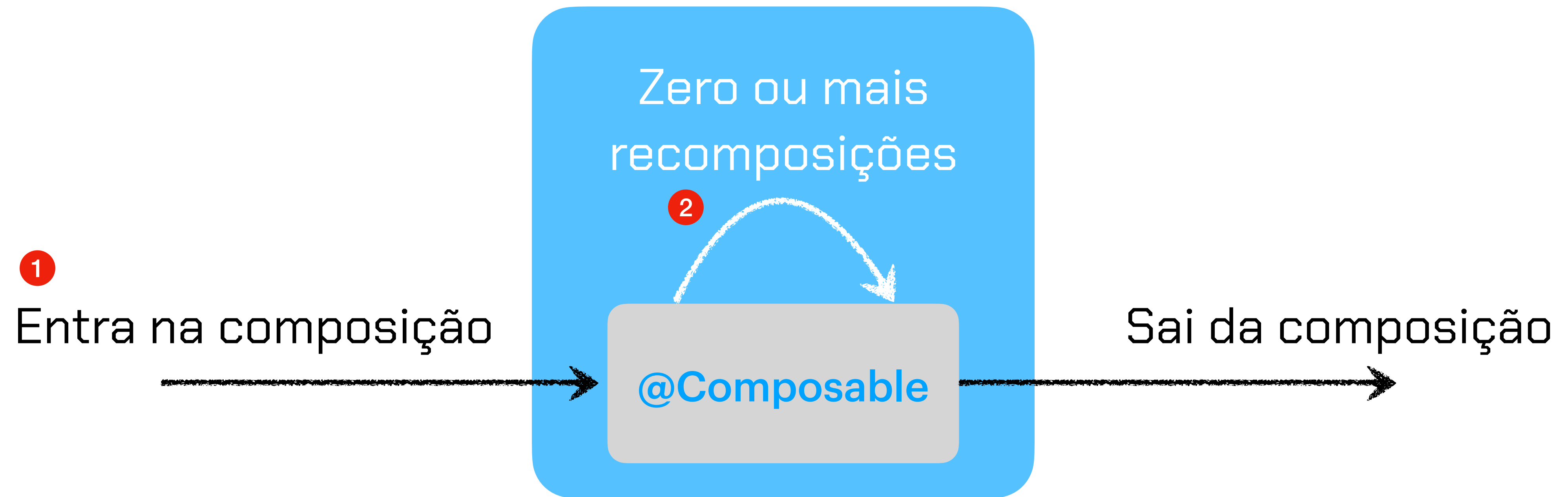


Estado de Apresentação

- Suportado por duas primitivas:
 - `mutableStateOf` (para interceptação das mutações)
 - `remember` (para *memoization*)
- Por conveniência, existem outras variantes construídas à custa destas



Ciclo de vida (2)



```
var email by remember { mutableStateOf("") }
```

Diagram illustrating the lifecycle of a Composable component using the `remember` function:

- 1** (under `remember`): Enters the composition.
- 2** (under `email`): Leaves the composition.

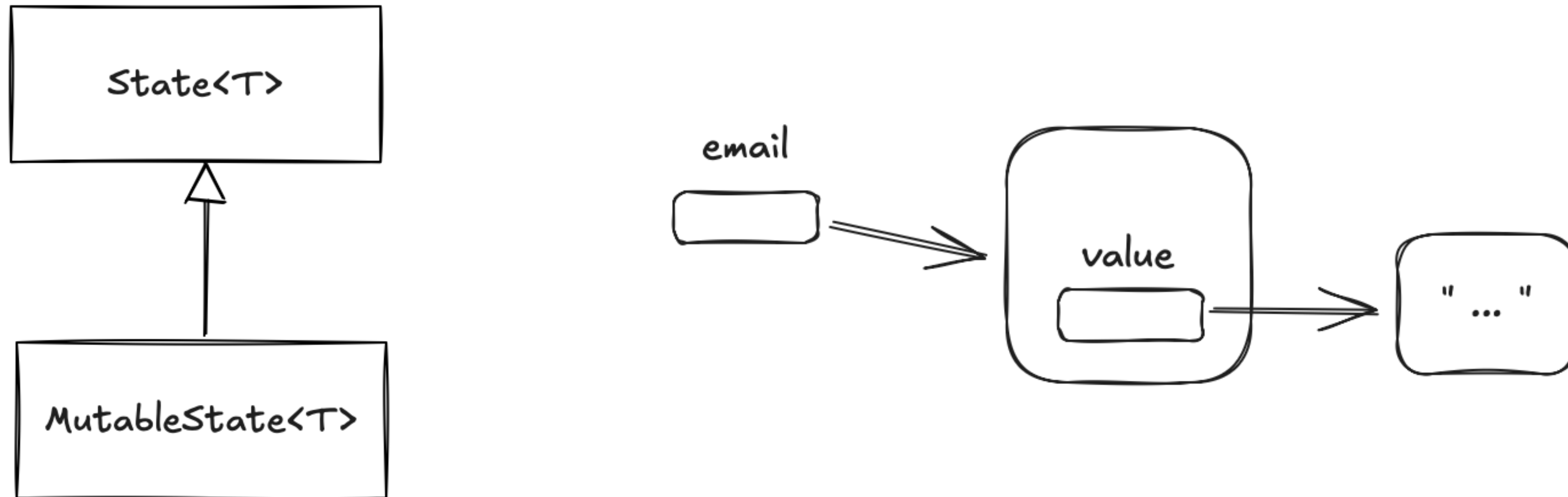


MutableStateOf<T>

```
var email = remember { mutableStateOf("") }
```

2

1



```
var email by remember { mutableStateOf("") }
```

2



Documentação oficial

- Ciclo de vida de Composables
 - <https://developer.android.com/develop/ui/compose/lifecycle>
- Estado de apresentação em Composables
 - <https://developer.android.com/develop/ui/compose/state>
 - <https://developer.android.com/develop/ui/compose/state-hoisting>



Compose - Estado

PDM - Programação para Dispositivos Móveis

Paulo Pereira
paulo.pereira@isel.pt