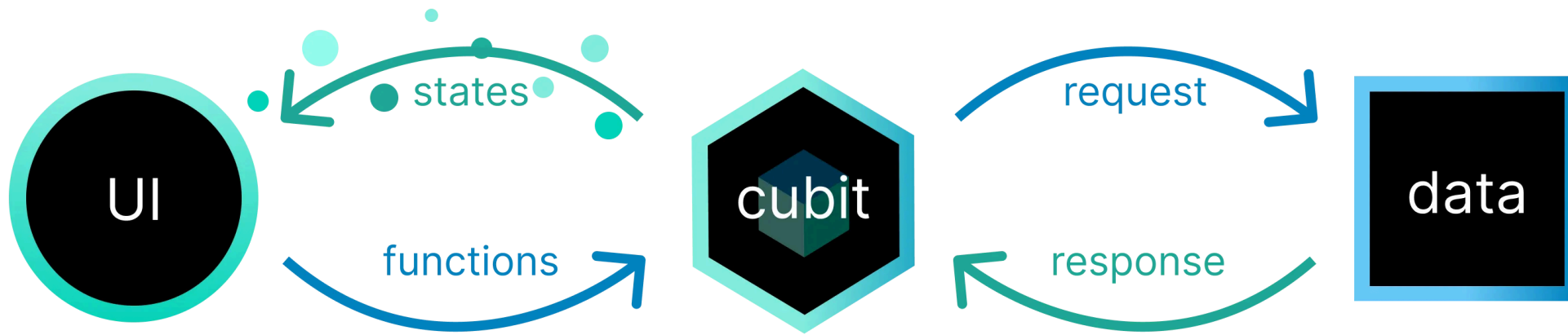


**bloc**

**Bloc (**`flutter_bloc`**)**

Presented by Paul Köhler

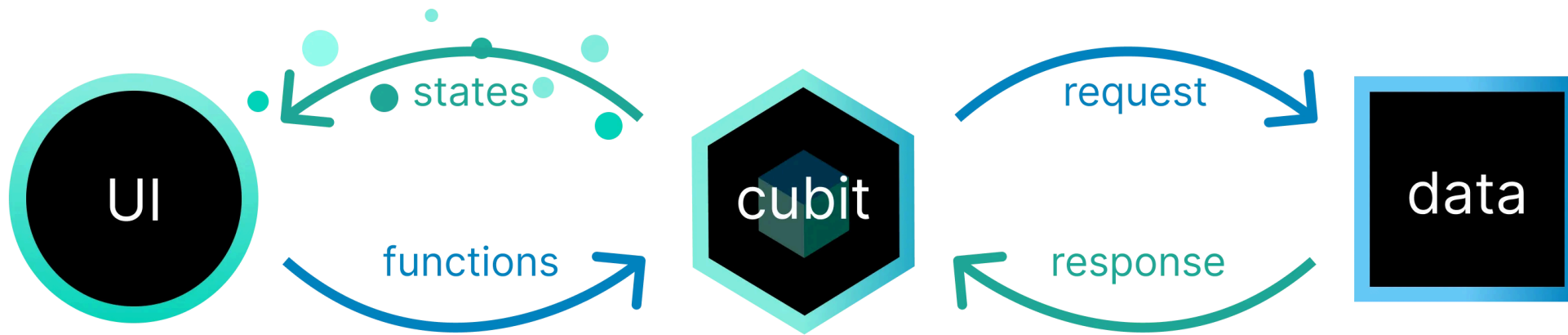
# What is **bloc**



# Why use `bloc`

- State management is hard.
- `bloc` is simple.

# But how?



- What is this `cubit` and how do I create one?

# How to define a **cubit**

```
class CounterCubit extends Cubit<int> {  
  CounterCubit() : super(0);  
  
  void increment() => emit(state + 1);  
  void decrement() => emit(state - 1);  
}
```

# Using a **cubit**: Initialization

```
class App extends StatelessWidget {  
  const App({super.key});  
  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      home: BlocProvider(  
        create: (context) => CounterCubit(),  
        child: const IndexPage()  
      ),  
      // ...  
    );  
  }  
}
```

# Using a **cubit**: Getting the value

```
// ...
class Counter extends StatelessWidget {
  const Counter({super.key});

  @override
  Widget build(BuildContext context) {
    return BlocBuilder<CounterCubit, int>(
      builder: (context, state) => Text('Current counter: $state'),
    );
  }
}
// ...
```

# Using a **cubit**: Setting the value

```
// ...  
void _increment(BuildContext context) => context.read<CounterCubit>().increment();  
// ...
```



# Summary

- `bloc` is very useful and simple for state management.

# Live demo

# Questions?

# References

The full source code is available on Github:

[https://github.com/paulkoehlerdev/bloc\\_example](https://github.com/paulkoehlerdev/bloc_example)



The slides were created with [Marp](#)