

# Unidirectional Data Flow Rocks

Introducing Redux, Cubits and Blocs.



# Heads up

This is a dense talk.

Slides and code are available online.

Full-time employee at  
**bloom** technologies



urce  
5+ years

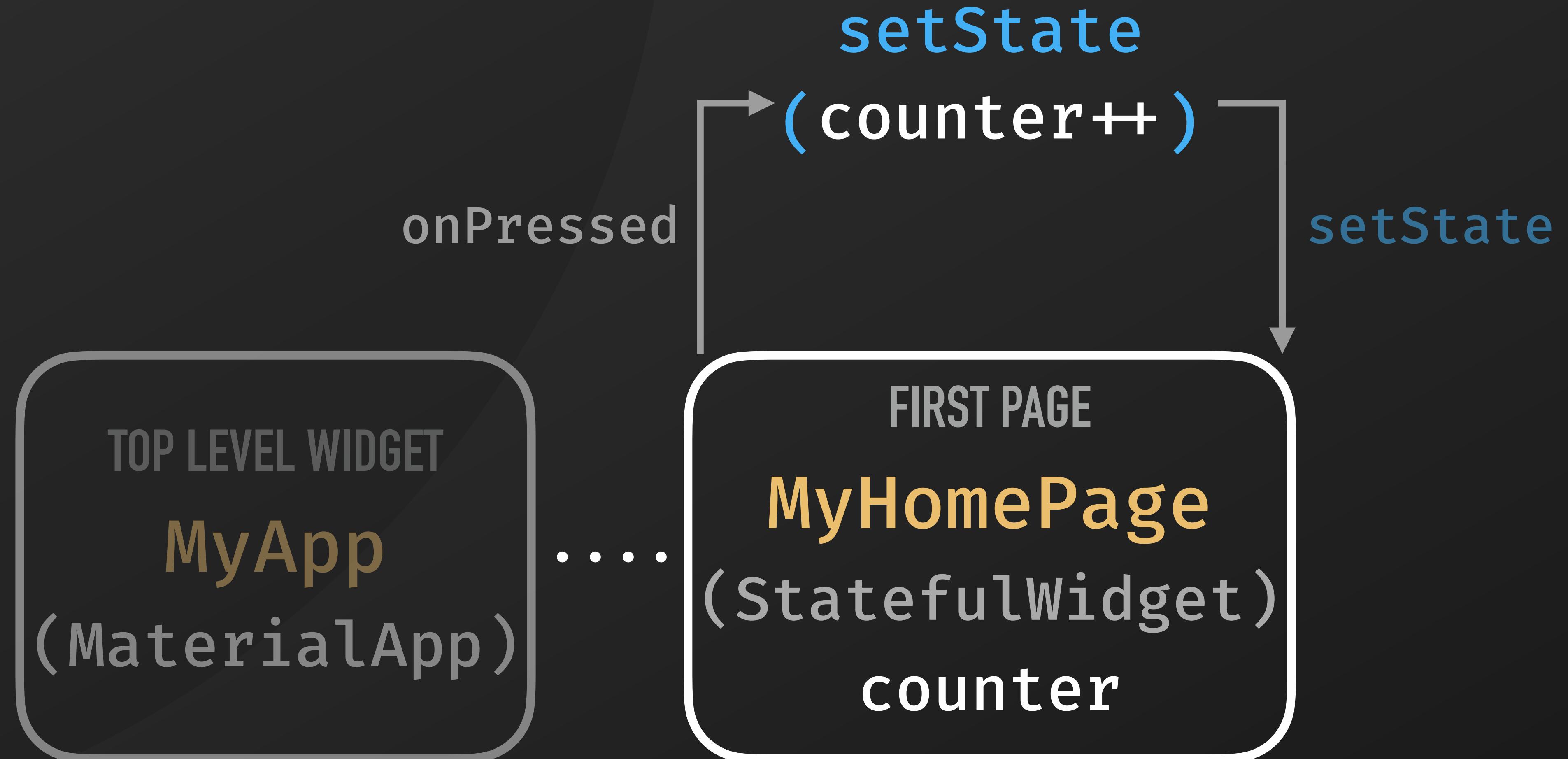
Spoken at  
**8 conferences**  
and smaller meetups

# State Management

# Why

- Requirements and complexity grow
- Data that flows through your app must be managed properly
- “State management” describes the process and approaches for data in your application

# Example



```
class MyApp extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return new MaterialApp(  
      title: 'Flutter Demo',  
      theme: new ThemeData(  
        primaryColor: Colors.blue,  
      ),  
      home: new MyHomePage(title: 'Flutter Demo Home Page'),  
    );  
  }  
}
```

```
class MyHomePage extends StatefulWidget {  
    MyHomePage({Key key, this.title}) : super(key: key);  
  
    final String title;  
  
    @override  
    _MyHomePageState createState() => new _MyHomePageState();  
}
```

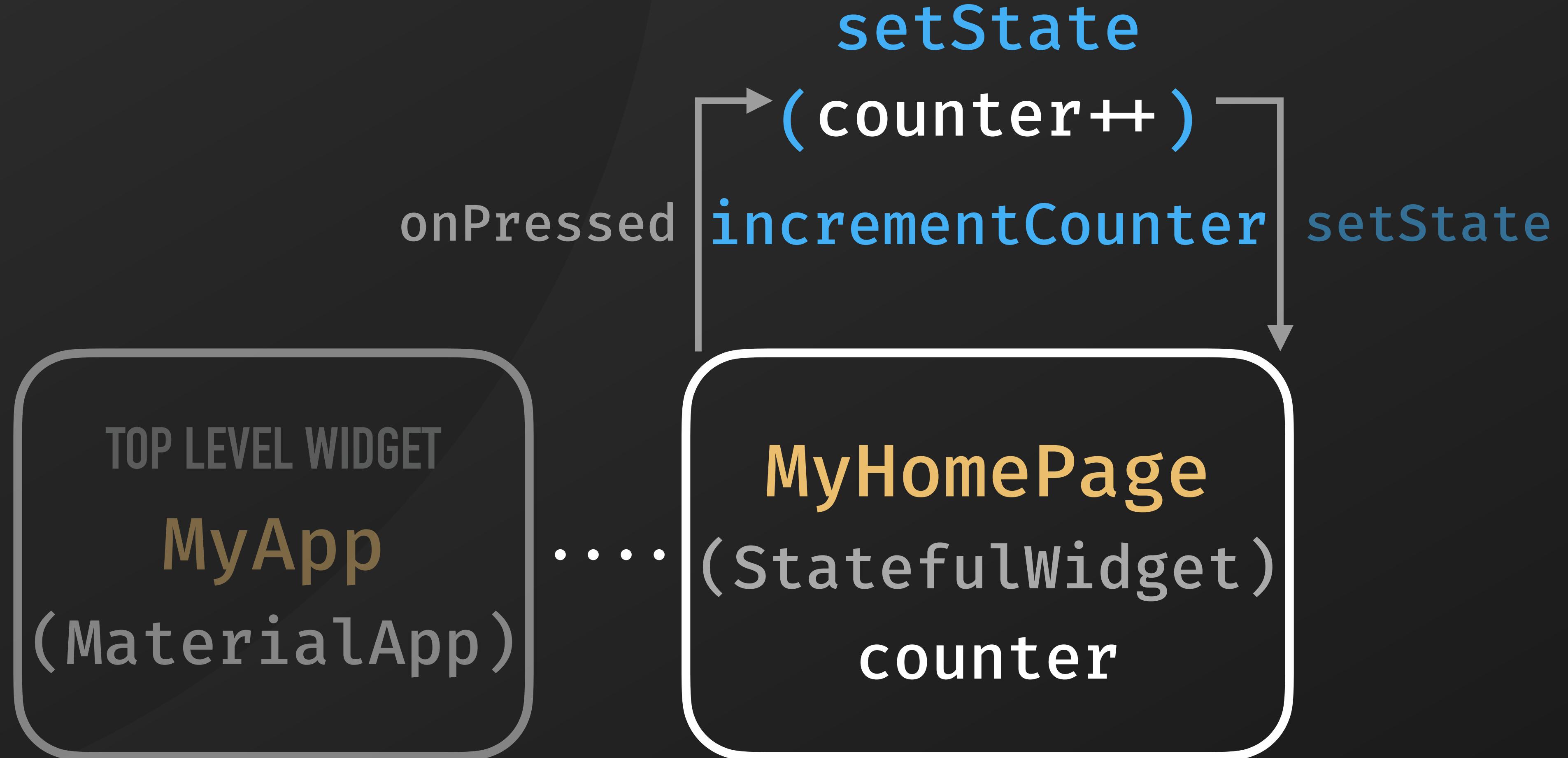
```
class _MyHomePageState extends State<MyHomePage> {  
    int _counter = 0;
```

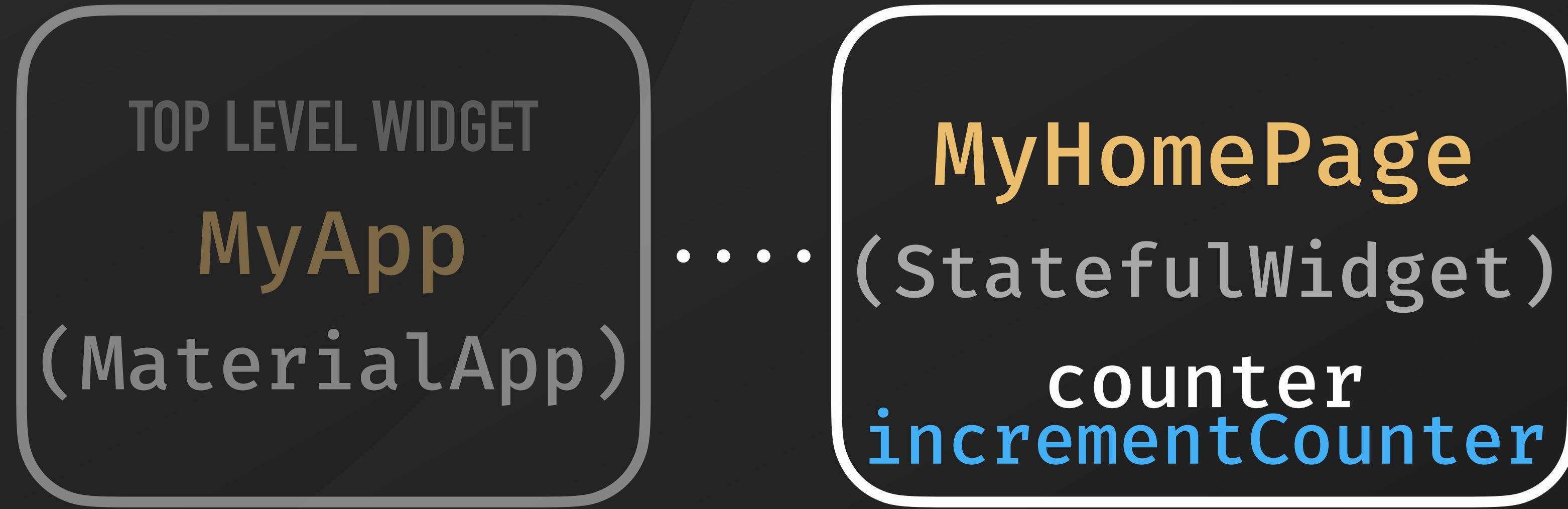
```
    void _incrementCounter() {  
        setState(() {  
            _counter++;  
        });  
    }  
}
```

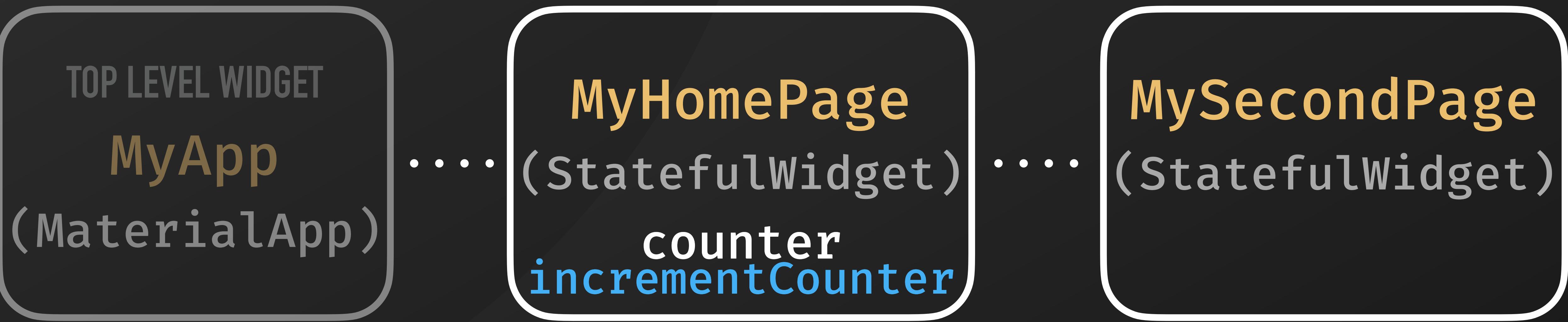
...

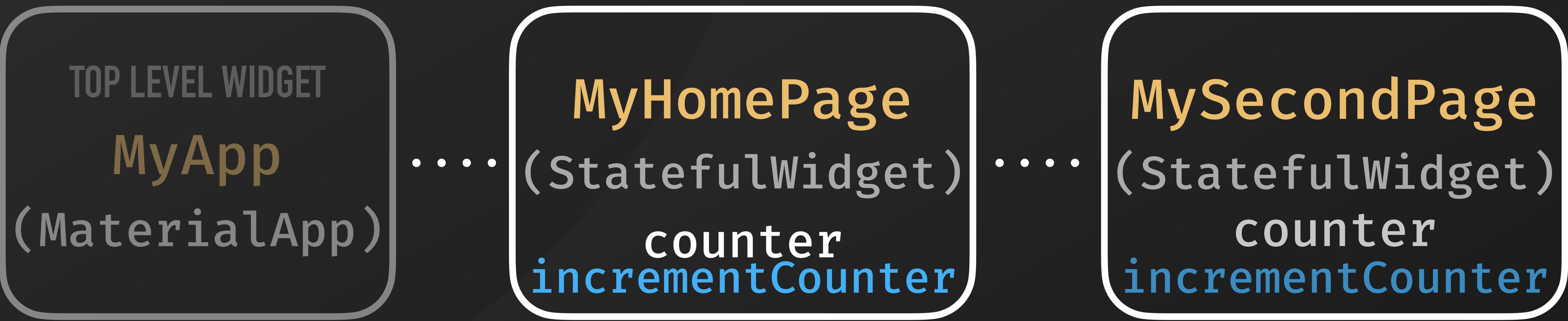
```
floatingActionButton: new FloatingActionButton(  
    onPressed: _incrementCounter,  
    tooltip: 'Increment',  
)
```

...









TOP LEVEL WIDGET  
**MyApp**  
(**MaterialApp**)

....

**MyHomePage**  
(**StatefulWidget**)  
counter  
**incrementCounter**

:

```
new MySecondPage(  
    counter: _counter,  
    incrementCounter: _incrementCounter,  
)
```

:

:

TOP LEVEL WIDGET  
MyApp  
(MaterialApp)

....

MyHomePage  
( StatefulWidget )  
counter  
incrementCounter

:

```
new MySecondPage(  
    counter: _counter,  
    incrementCounter: _incrementCounter,  
)
```

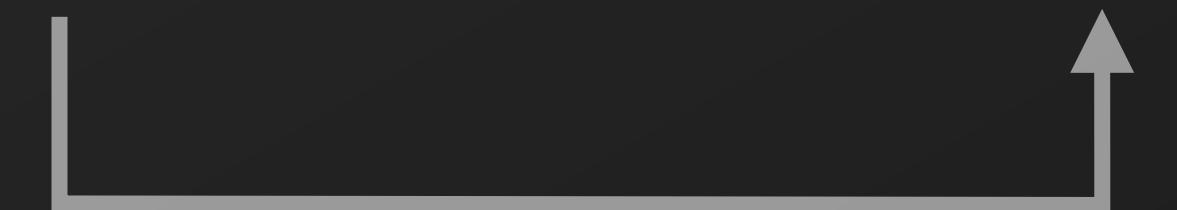
:

:

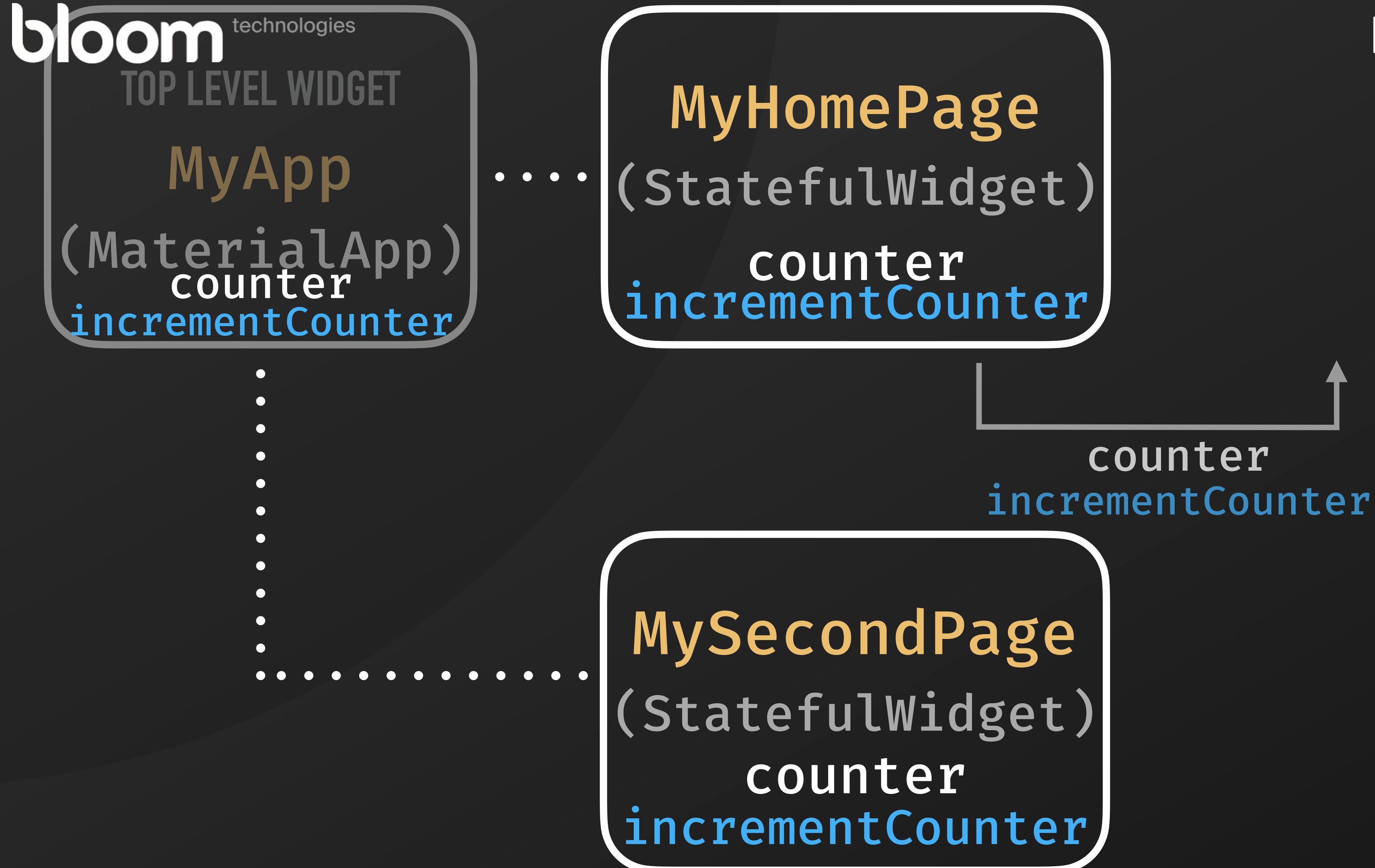
TOP LEVEL WIDGET  
MyApp  
(MaterialApp)

....

MyHomePage  
( StatefulWidget )  
counter  
**incrementCounter**



counter  
**incrementCounter**



# Example

**bloom** technologies  
TOP LEVEL WIDGET  
**MyApp**  
(**MaterialApp**)  
counter  
**incrementCounter**



**MyHomePage**  
(**StatefulWidget**)  
counter  
**incrementCounter**

This is

AnotherPage

AnotherPage

**MySecondPage**  
(**StatefulWidget**)  
counter  
**incrementCounter**

Example  
AnotherPage

AnotherPage

Really?

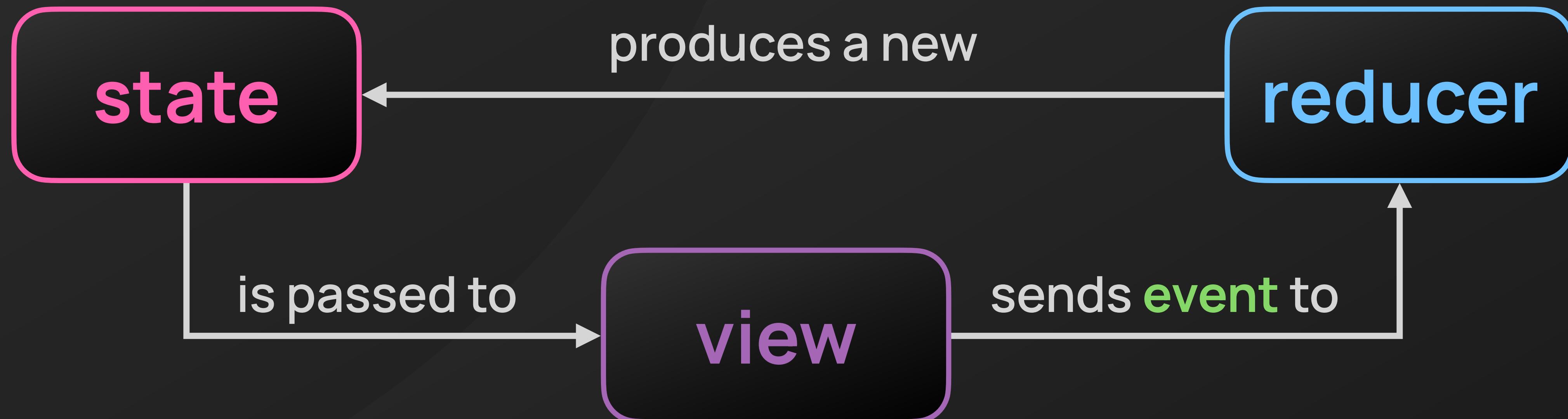
AnotherPage

# Unidirectional Data Flow

# Ingredients

- An immutable **state** that represents (part of) your app
- A collection of **events** that represent actions
- A **reducer/pure function** that takes a state and an event and produces a new state

# Unidirectional Data Flow



Redux

Bloc/Cubit

# About Bloc

- “Business logic component”
- A class containing one **state** and **reducer** functions
- Consumes **events** (usually as classes) and produces new **states**
- Uses the Stream API

```
import 'package:bloc/bloc.dart';

class CounterBloc extends Bloc<CounterEvent, CounterState> { ... }

abstract class CounterEvent { ... }

class CounterState {
  final int count;
}
```

MaterialApp

BlocProvider  
CounterBloc()

BlocBuilder  
<CounterBloc>

# About Cubit

- A smaller, simpler form of Bloc
- Same concept, different execution
- Uses public methods instead of classes to convey **events**

```
import 'package:bloc/bloc.dart';
```

```
class CounterCubit extends Cubit<CounterState> { ... }
```

```
class CounterState {  
  final int count;  
}
```

# Cubit in action



My contact details

# Thank you!

Have any questions or just wanna chat?  
Come talk to me after the session or  
reach out online anytime.



Slides & code