R2S
Resource Software Solution

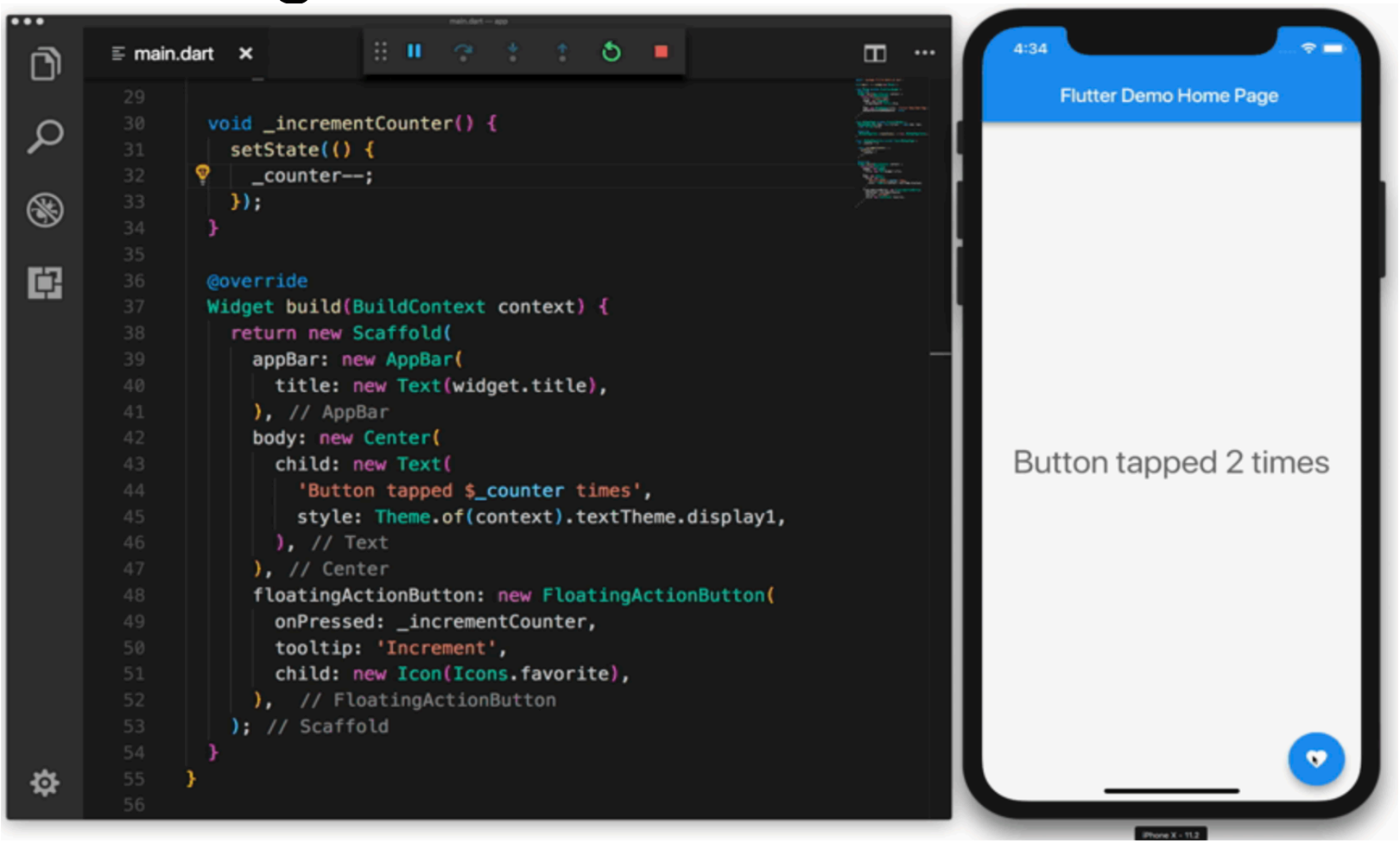
Stateful vs Stateless Widgets

What is State

- The familiar math formula **y = f (x)**. It is a function, when we have the **value of x**, based on a **function f** we get the **value of y**. Whenever x changes, it gives us a new value of y, right.
- Flutter is similar, it uses a formula of:

What is State

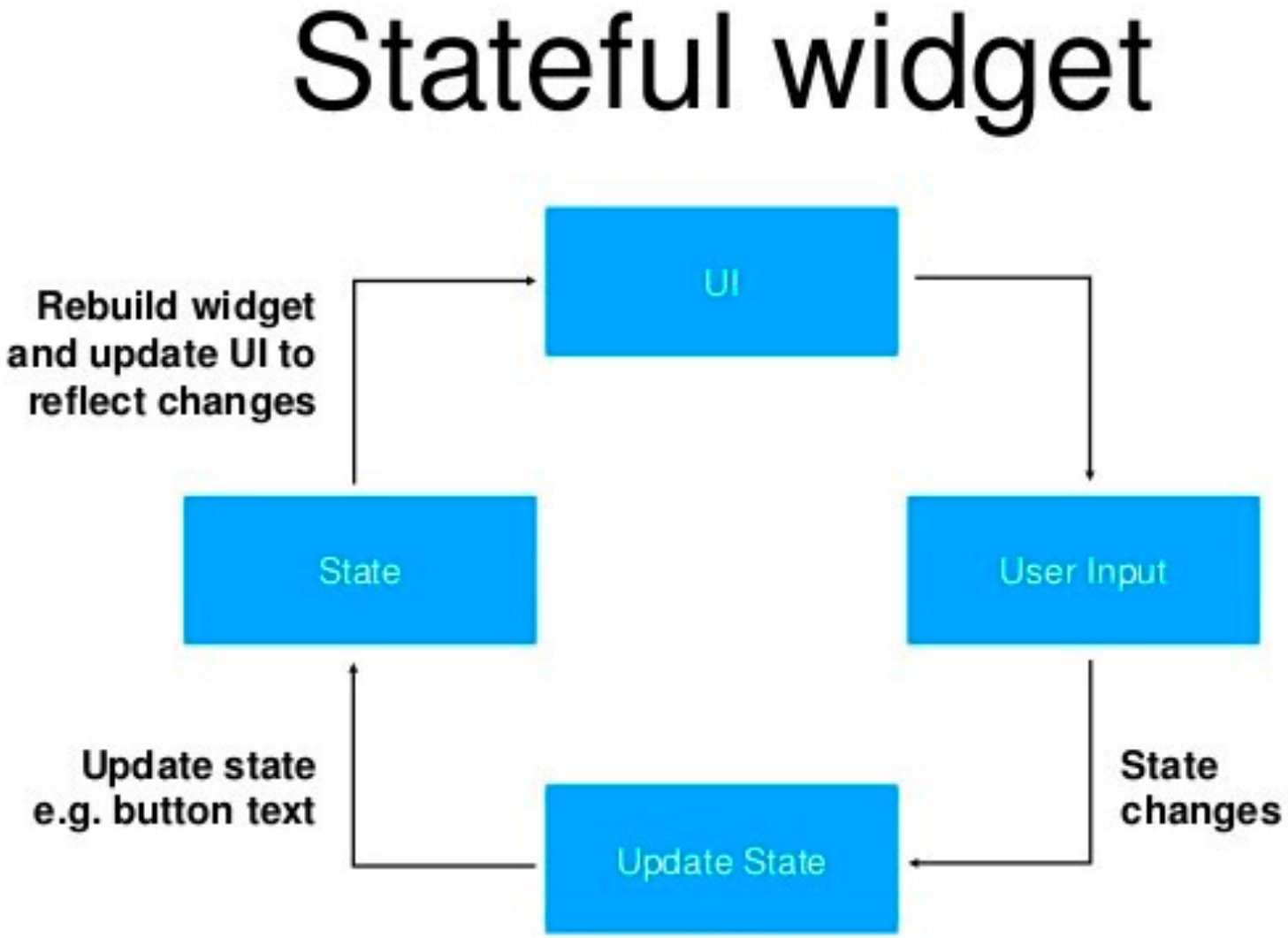
- When the **Widget Data changes**, the **UI** will be **updated** according to the formula f.



UI = f(state)

The layout on the screen — Your build methods — The application state

What is State



What is State

- Suppose now, you create a Widget yourself as a light bulb. What information will the light bulb have:
  1. The **size** of the bulb is of type int . This information **never changes**. For example, the bulb when produced is size 20, 10 years later, it will also be size 20, but it cannot grow or shrink over the years, unless it is smashed (Widget die) =))
  2. The **color** is displaying the Color light, the default bulb color will be yellow, but sometimes it will change to red, sometimes it will be blue. This is **changeable** information. If the light color does not change, it could be damaged (Widget die) =))

What is State

- State is **information/properties** shown on Widget that **can change** during the lifetime of the Widget.
- When **state changes**, the **build** function will be called back to **update the UI** (we call this the rebuild Widget).
- There are two types of widgets (Each with a **build** function, but the way they call the build function to update the UI is different)
  1. **StatefulWidget**
  2. **StatelessWidget**

StatefulWidget

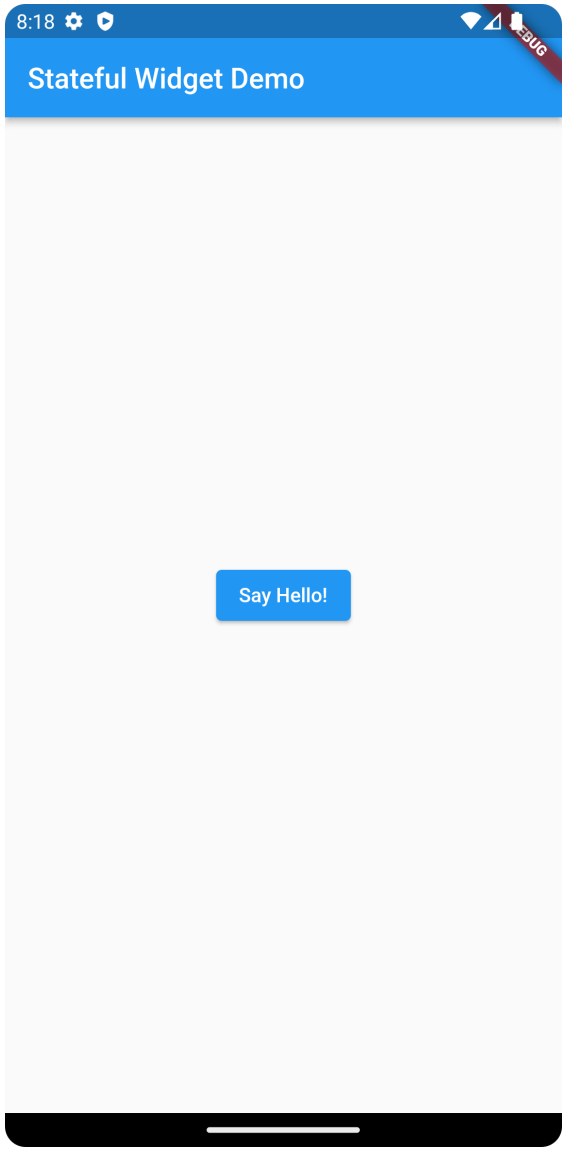
- A widget that has **mutable state**. A stateful widget is dynamic. For example, it can change its appearance in response to events triggered by user interactions or when it receives data.
- Checkbox, Radio, Slider, Form, and TextField are examples of stateful widgets.
- When the **state changes**, it calls the **build function again** to **rebuild the widget**. The **UI changes**.

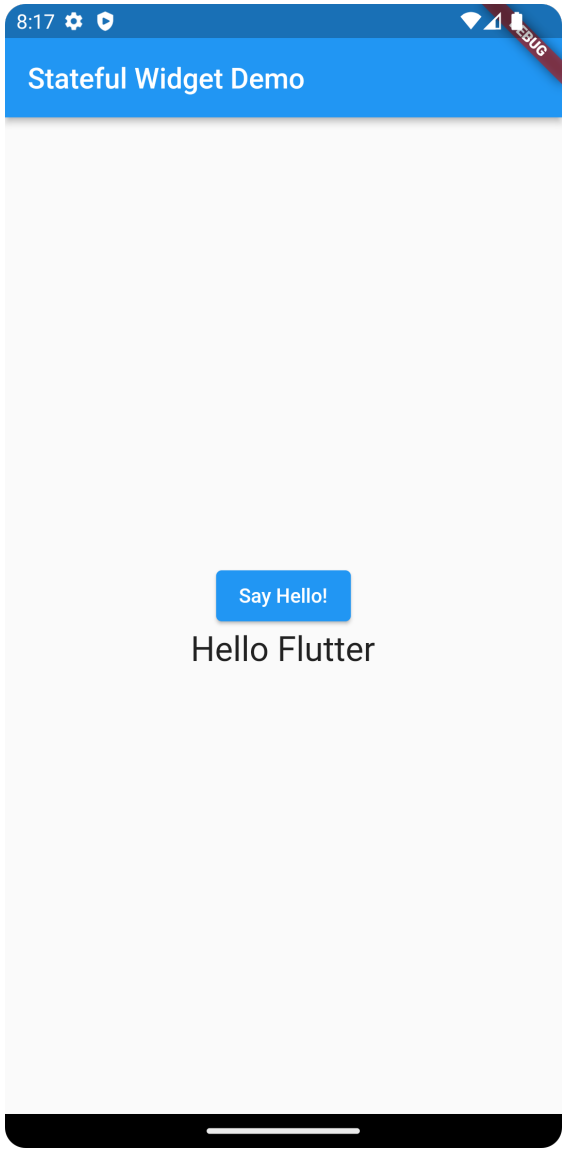
StatelessWidget

- A stateless widget **never changes**. Icon, IconButton, and Text are examples of stateless widgets.
- **Stateless** widgets have **no state**. It does not accept change within it. So, it has nothing to do with **State at** all. It itself does not have a **createState** function.

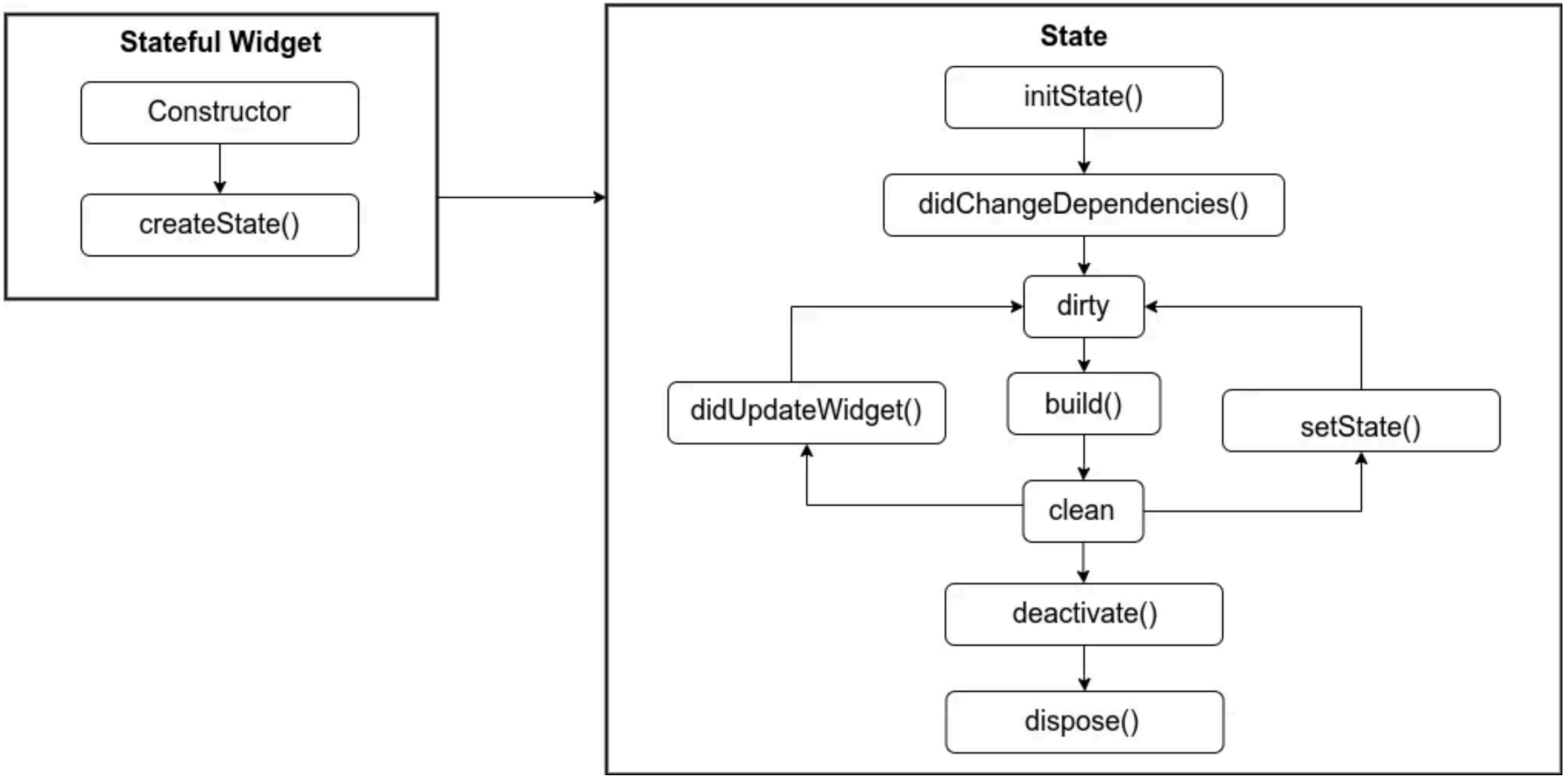
Example: Say Hello!

```dart
class StatefulWidgetDemo extends StatefulWidget {
  const StatefulWidgetDemo({super.key});

  @override
  State<StatefulWidgetDemo> createState() => _StatefulWidgetDemoState();
}

class _StatefulWidgetDemoState extends State<StatefulWidgetDemo> {
  var message = '';

  void sayHello() {
    setState(() {
      message = 'Hello Flutter';
    });
  }

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(title: const Text('Stateful Widget Demo'),),
        body: Container(
          alignment: Alignment.center,
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              ElevatedButton(
                  onPressed: sayHello,
                  child: const Text('Say Hello!')
              ),  // ElevatedButton
              Text(message, style: const TextStyle(fontSize: 24),)
            ],
          ),  // Column
        ),  // Container
      ),  // Scaffold
    );  // MaterialApp
  }
}
```



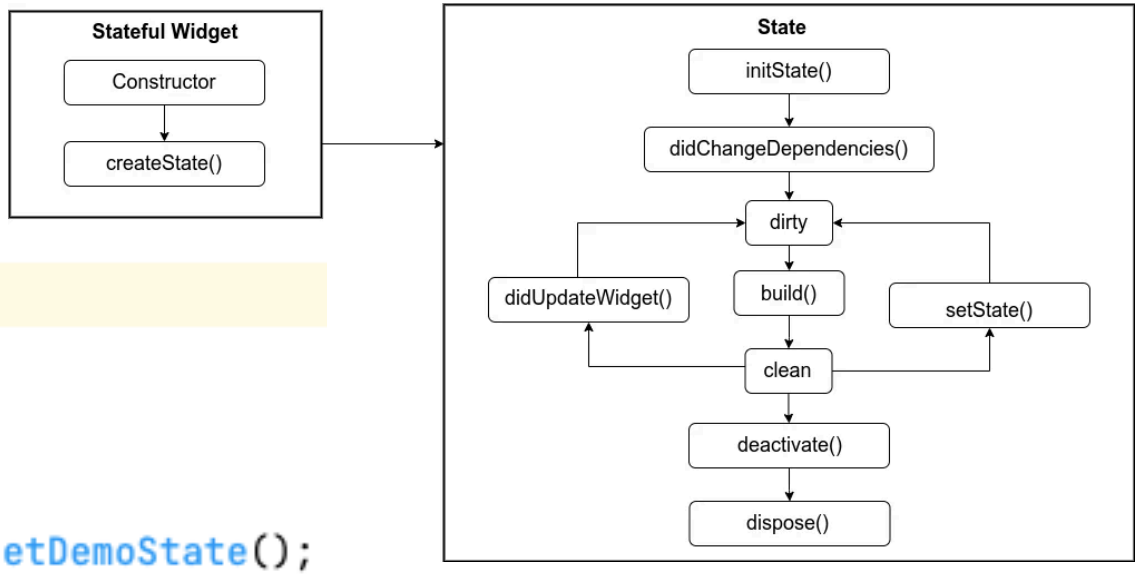


StatefulWidget Lifecycle



StatefulWidget Lifecycle

- **createState()**: The *createState()* method is responsible for creating a **State object**

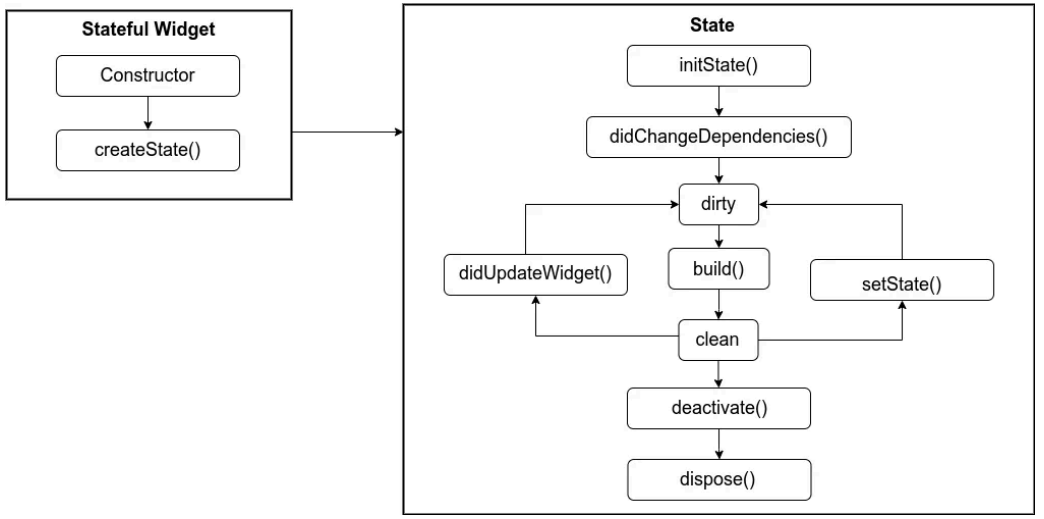



```dart
class StatefulWidgetDemo extends StatefulWidget {
  const StatefulWidgetDemo({super.key});

  @override
  State<StatefulWidgetDemo> createState() => _StatefulWidgetDemoState();
}
```

StatefulWidget Lifecycle

- **initState():**
  1. This method strictly executes **only once**
  2. It also requires to **call the *super.initState()*** method
  3. We can **initialize** variables, data, properties, etc

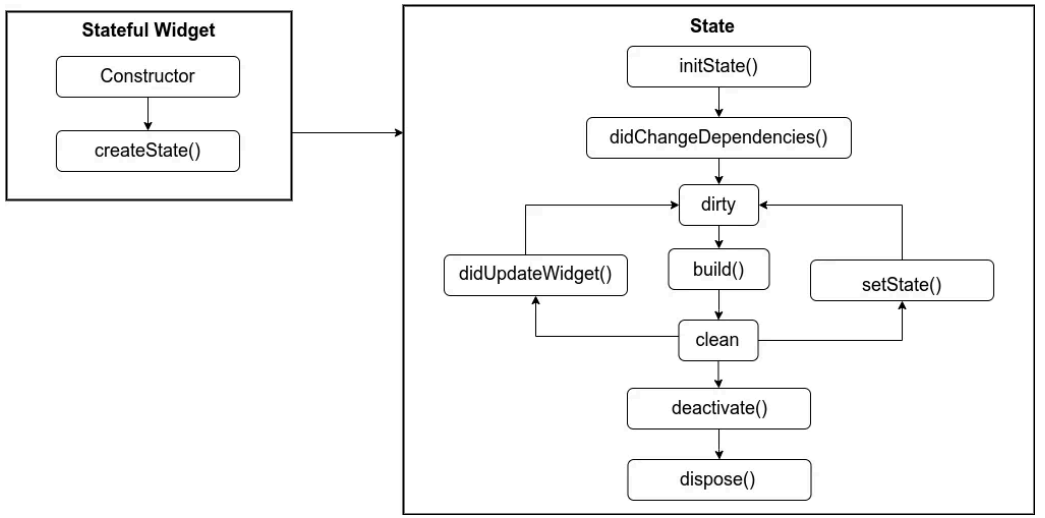
```
@override
void initState() {
  super.initState();

  message = '';
  isPressed = false;
}
```



StatefulWidget Lifecycle

- **build()**: It is the most essential lifecycle method for both a stateless and a stateful widget. It is responsible for **describing** and **rendering** widgets

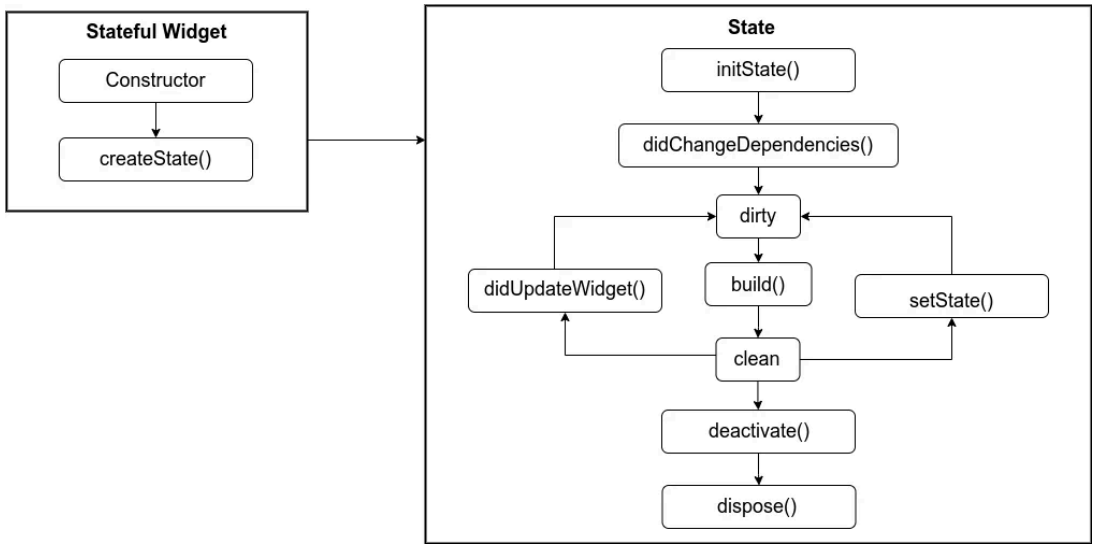
```
@override
Widget build(BuildContext context) {
  return MaterialApp(
    home: Scaffold(
      appBar: AppBar(
        title: const Text('Stateful Widget Demo'),
      ),  // AppBar
      body: Container(
        alignment: Alignment.center,
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            ElevatedButton(
              onPressed: sayHello, child: const Text('Say Hello!')),
            Text(
              message,
              style: const TextStyle(fontSize: 24),
            )  // Text
          ],
        ),  // Column
      ),  // Container
    ),  // Scaffold
  );  // MaterialApp
}
```



StatefulWidget Lifecycle

- **setState():**
  1. **Changes** in the **state object** and the need to **rebuild** the necessary **widget**.
  2. When calling the *setState()*, the **build** function is triggered for that state object which in turn **updates the UI**.

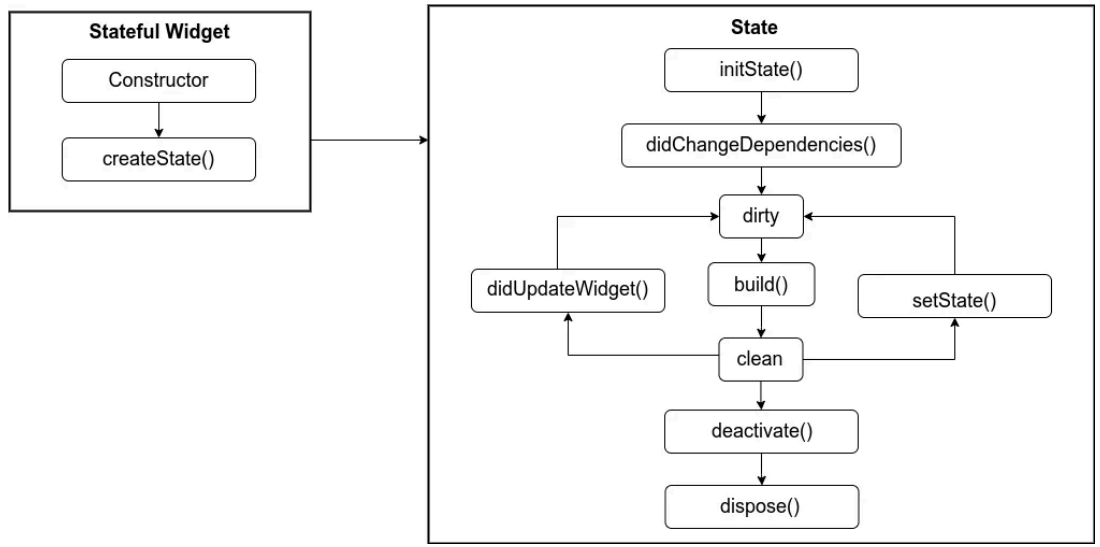
```
void sayHello() {
  if (mounted) {
    setState(() {
      message = 'Hello Flutter';
    });
  }
}
```



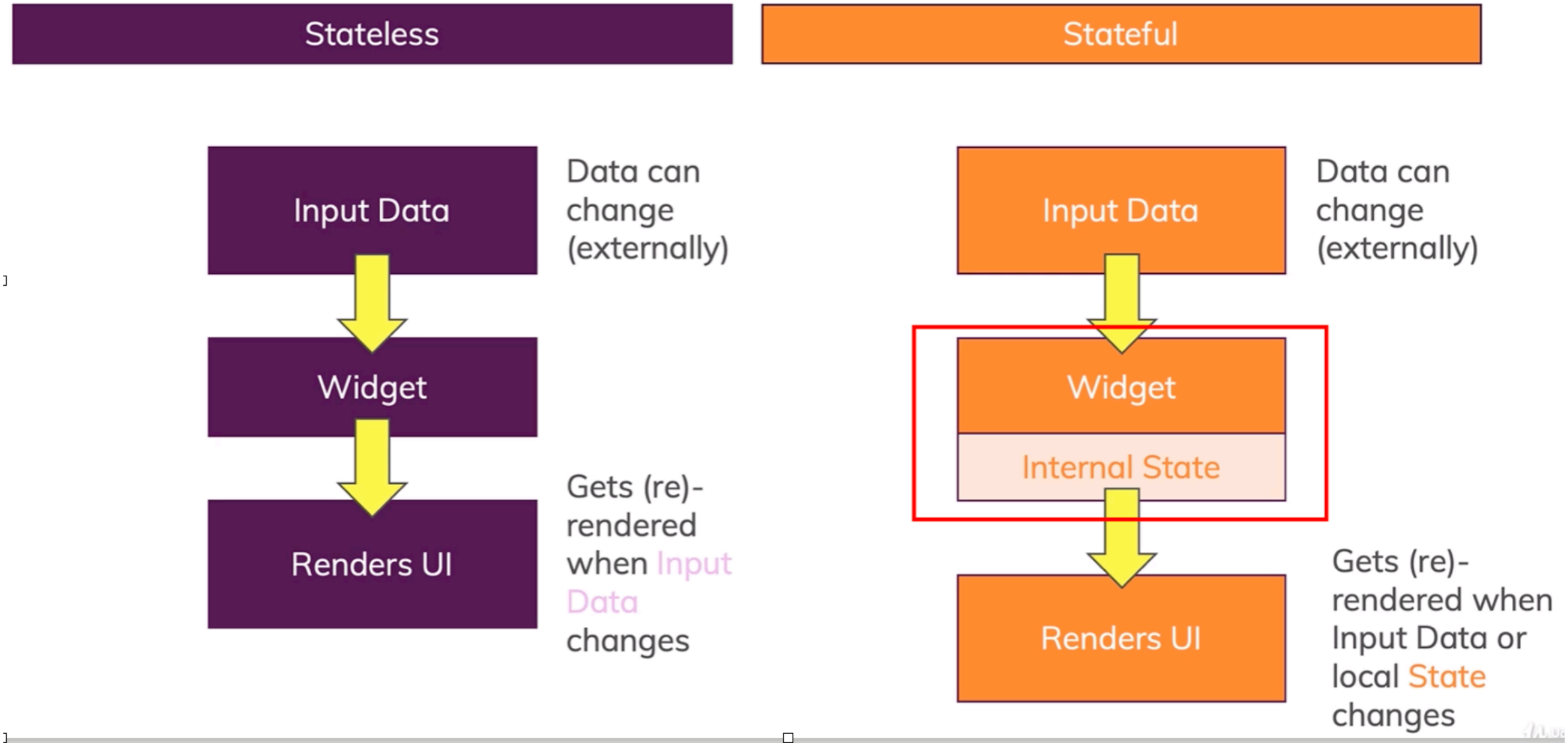
StatefulWidget Lifecycle

- **dispose():**
  1. Within the *dispose()* method we **release resources** held by the corresponding object.
  2. The state object's **mounted property** is set to **false** indicating that it will never be built again.

```
@override
void dispose() {
  super.dispose();
}
```



Stateless vs Stateful Widgets



Demo: Stateful Widgets

```dart
class _TapboxAState extends State<TapboxA> {
  bool _active = false;

  void _handleTap() {
    setState(() {
      _active = !_active;
    });
  }

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(title: const Text('Stateful Widget'),),
        body: GestureDetector(
          onTap: _handleTap,
          child: Center(
            child: Container(
              width: 200, height: 200,
              decoration: BoxDecoration(
                color: _active ? Colors.lightGreen : Colors.grey,
              ),  // BoxDecoration
              child: Center(
                child: Text(
                    _active ? 'Active' : 'Inactive',
                    style: const TextStyle(fontSize: 32, color: Colors.white),
                  ),  // Text
              ),  // Center
              ),  // Container
            )  // Center
          ),  // GestureDetector
        ),  // Scaffold
      );  // MaterialApp
    }
}
```