Flutter

Study JAM

By : Ahmed Abu Eldahab @dahabdev

Ahmed Abu Eldahab

Senior Technical Consultant
Google Developer Expert in Flutter

/Dahabdev

AGENDA

20 MINUTES	**Intro**
40 MINUTES	**Flutter Beyond Hello world! (Part 1)**
10 MINUTES	**Break**
60 MINUTES	**Flutter Beyond Hello world! (Part 2)**
SHARE FEEDBACK	Share photos, insights using the hashtags #StudyJamMENA #GDGMENA #Flutter #FlutterStudyJam #OnlineFlutterStudyJam #FlutterDev @dahabdev

Study

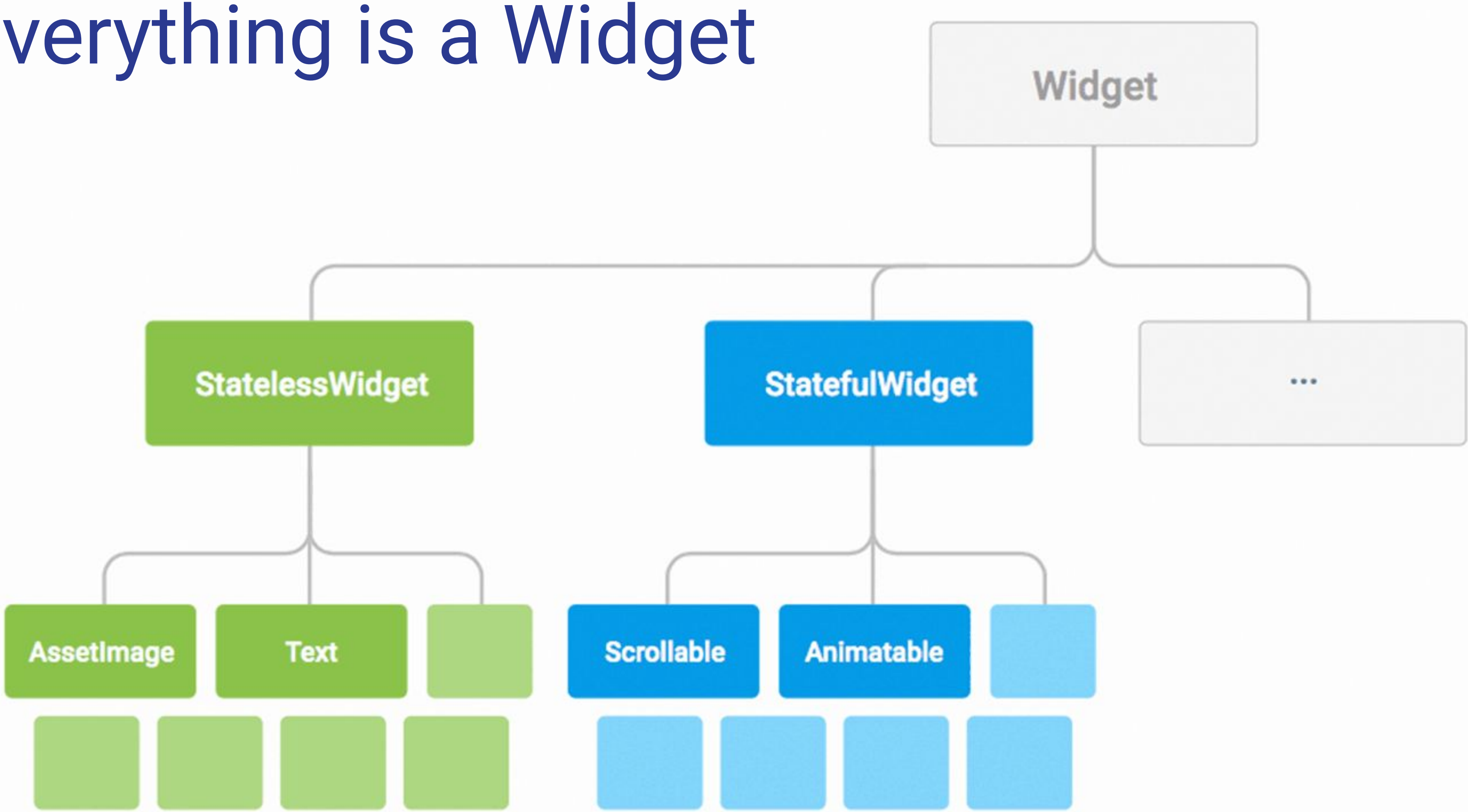
Flutter Beyond Hello world!

By : Ahmed Abu Eldahab

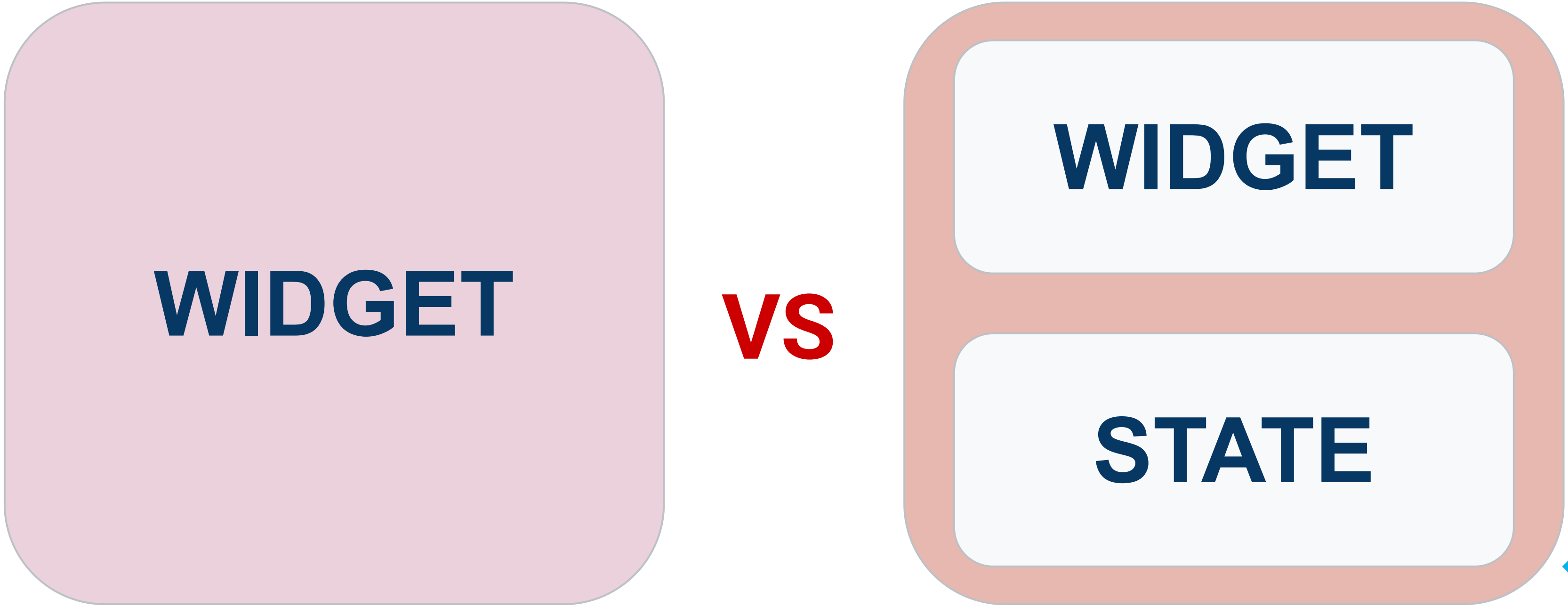
Online StudyJam

@dahabdev

Everything is a Widget



Everything is a Widget



Stateless Widget

WIDGET

VS

Stateful Widget

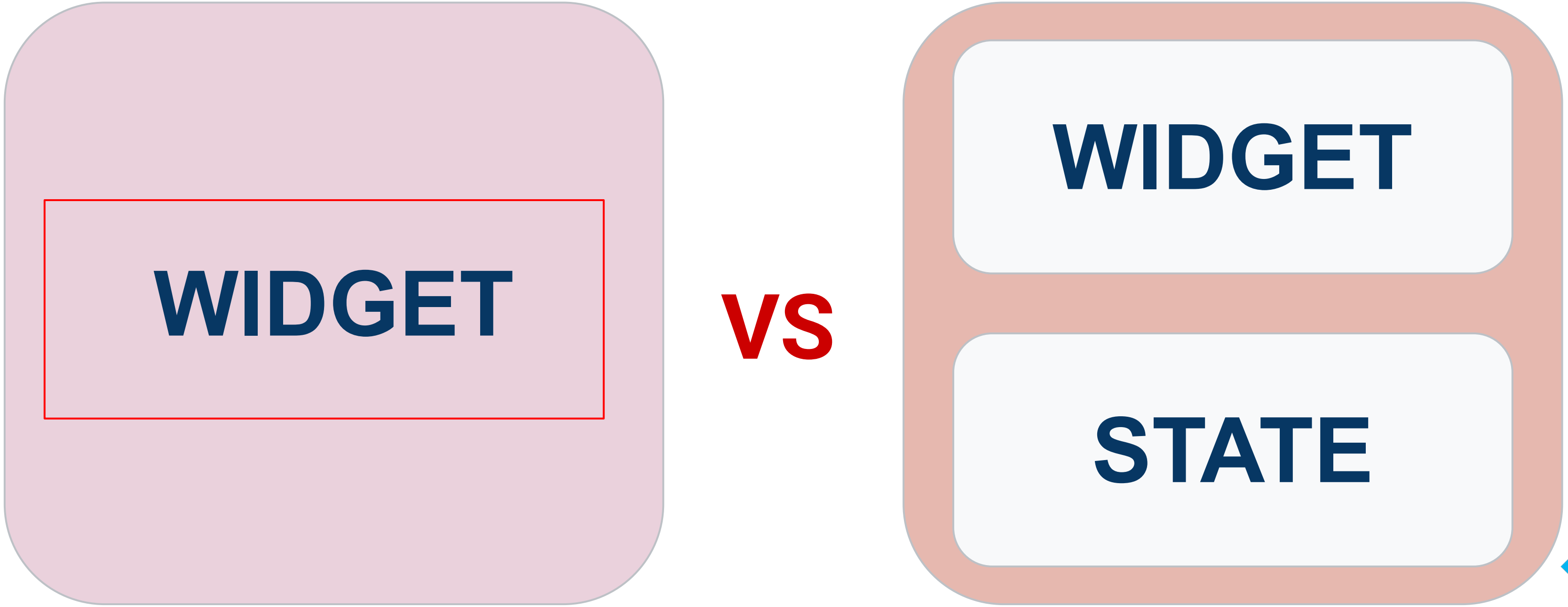
WIDGET

STATE

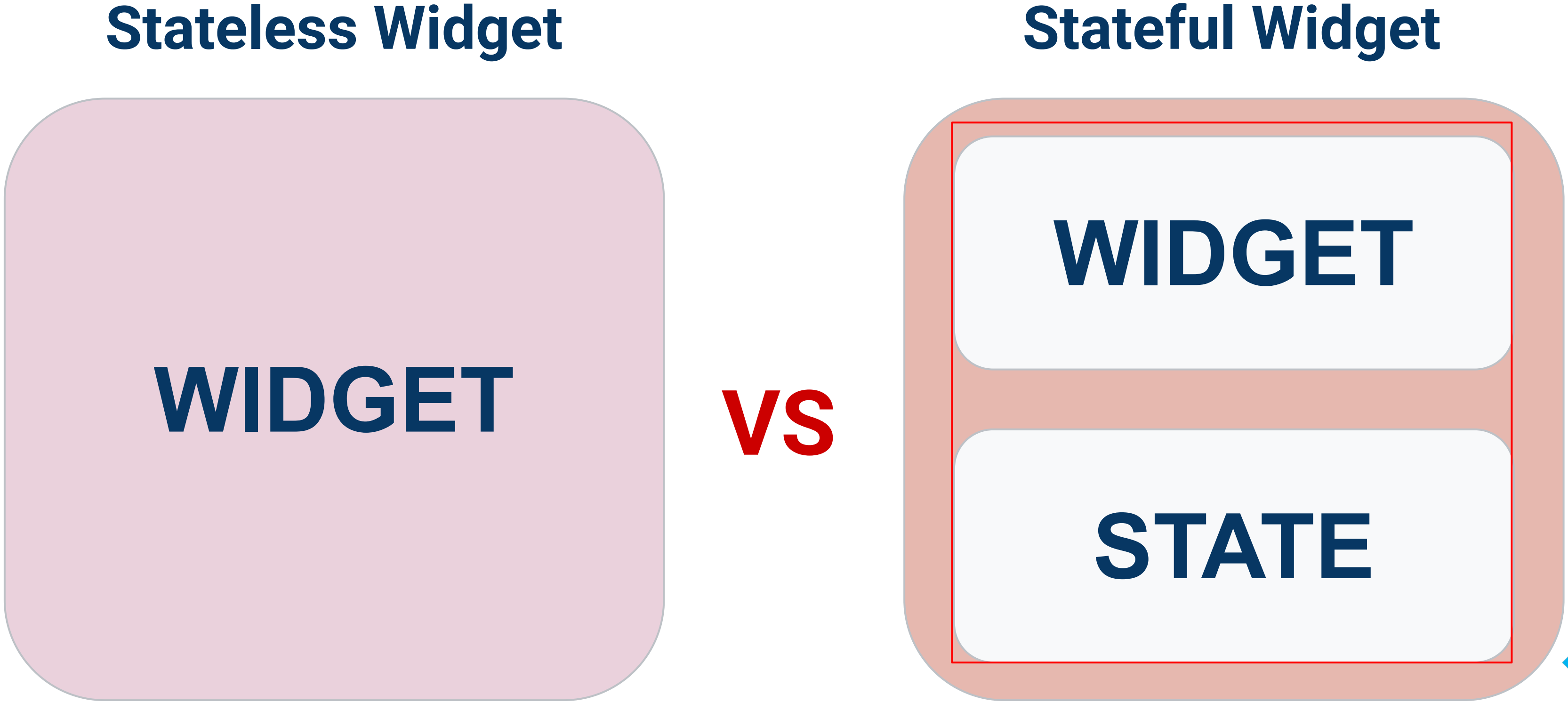
Everything is a Widget

Stateless Widget

Stateful Widget



Everything is a Widget



Stateless Widget

WIDGET

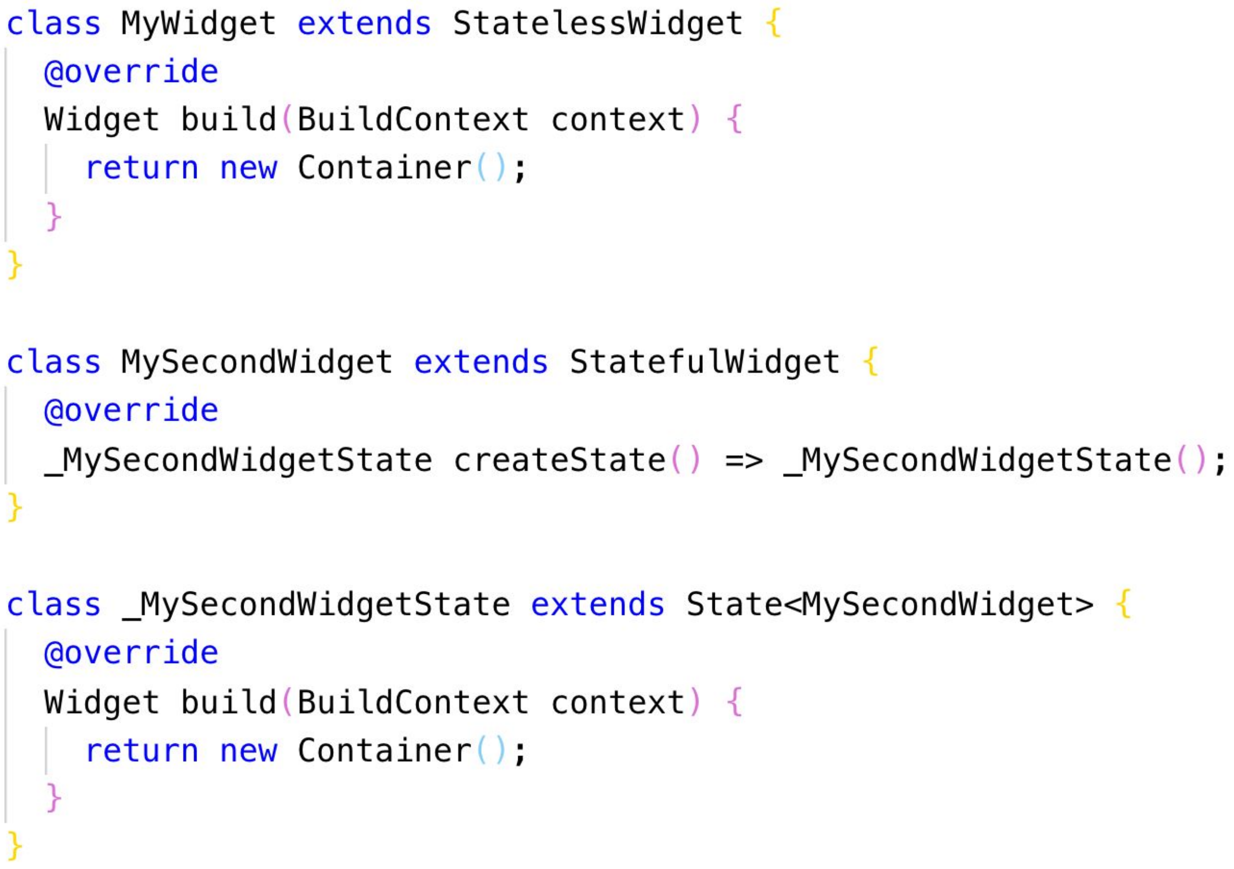
VS

Stateful Widget

WIDGET

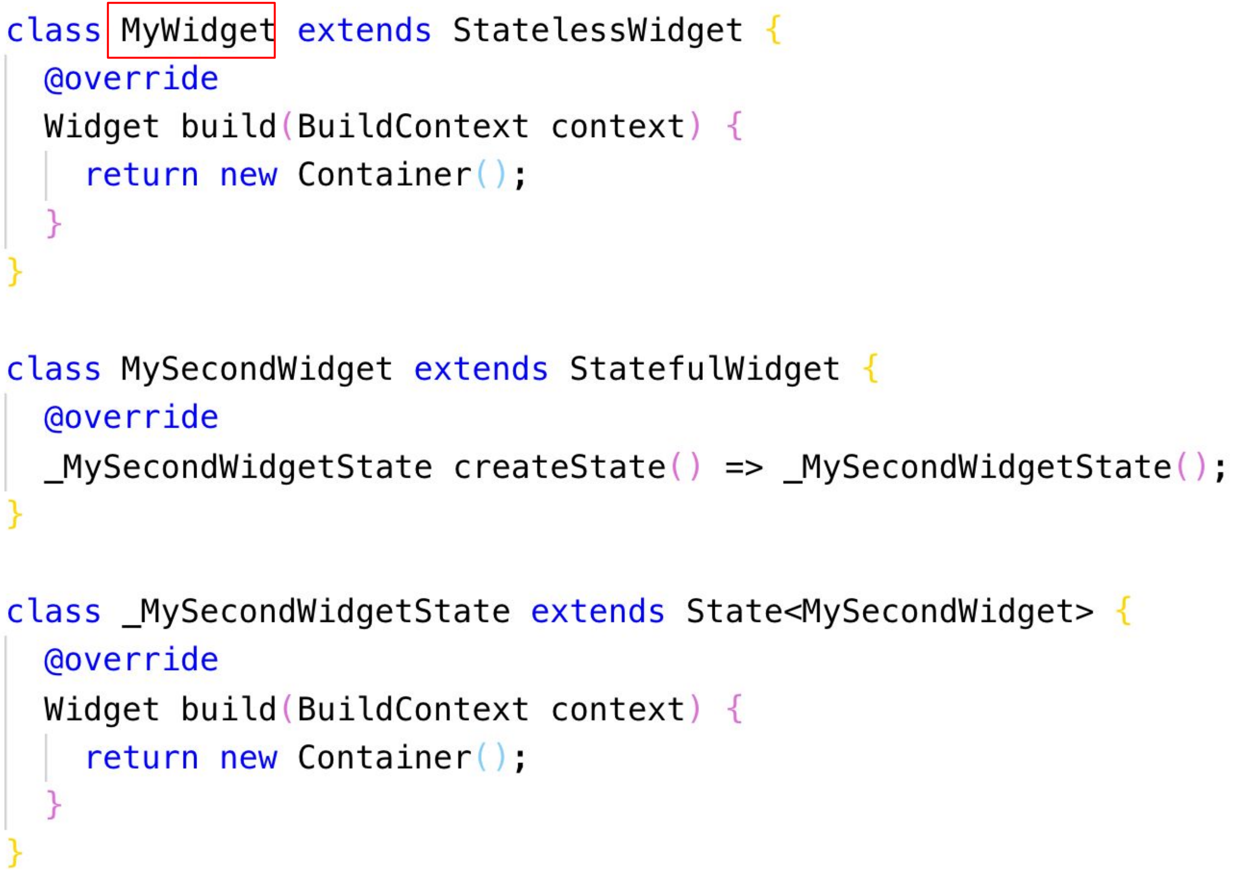
STATE

Flutter Widgets



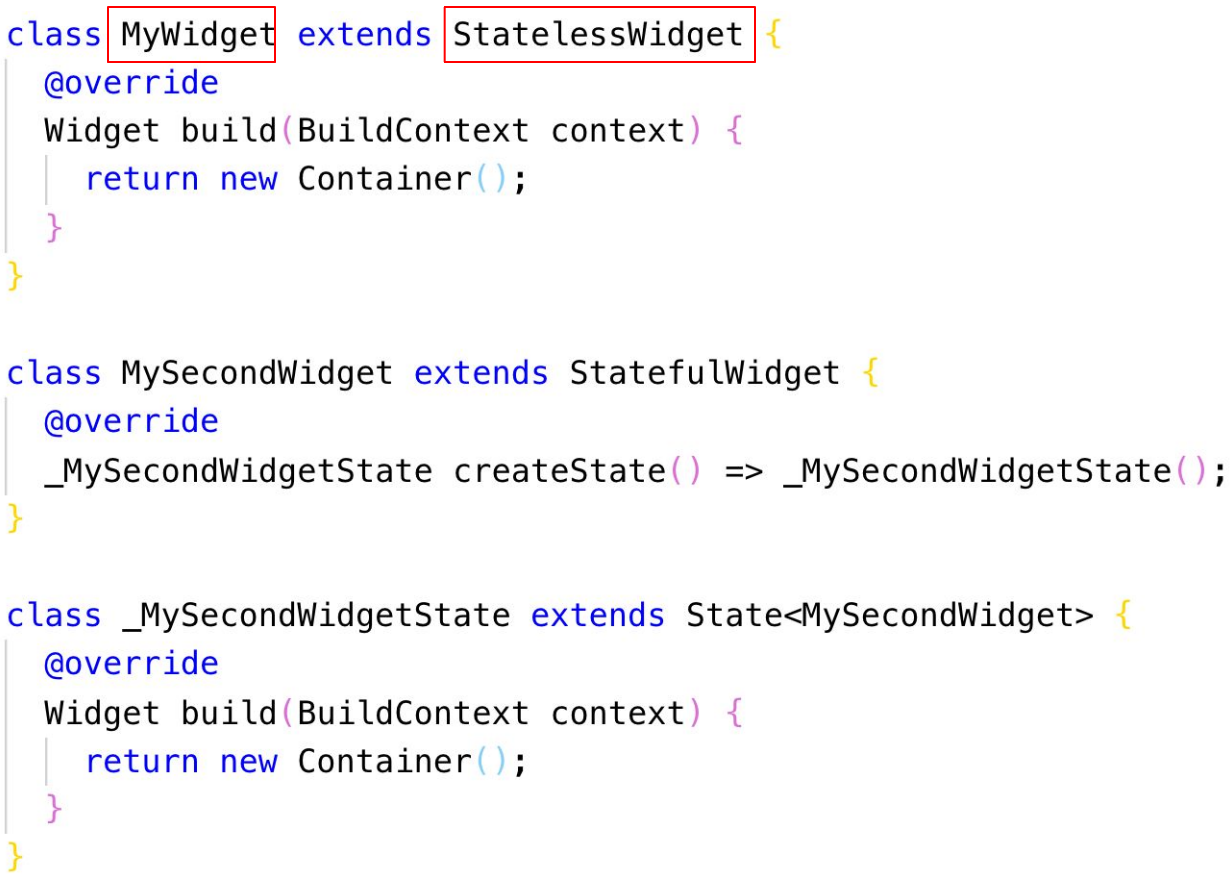
```
class MyWidget extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return new Container();
  }
}

class MySecondWidget extends StatefulWidget {
  @override
  _MySecondWidgetState createState() => _MySecondWidgetState();
}

class _MySecondWidgetState extends State<MySecondWidget> {
  @override
  Widget build(BuildContext context) {
    return new Container();
  }
}
```

Flutter Widgets



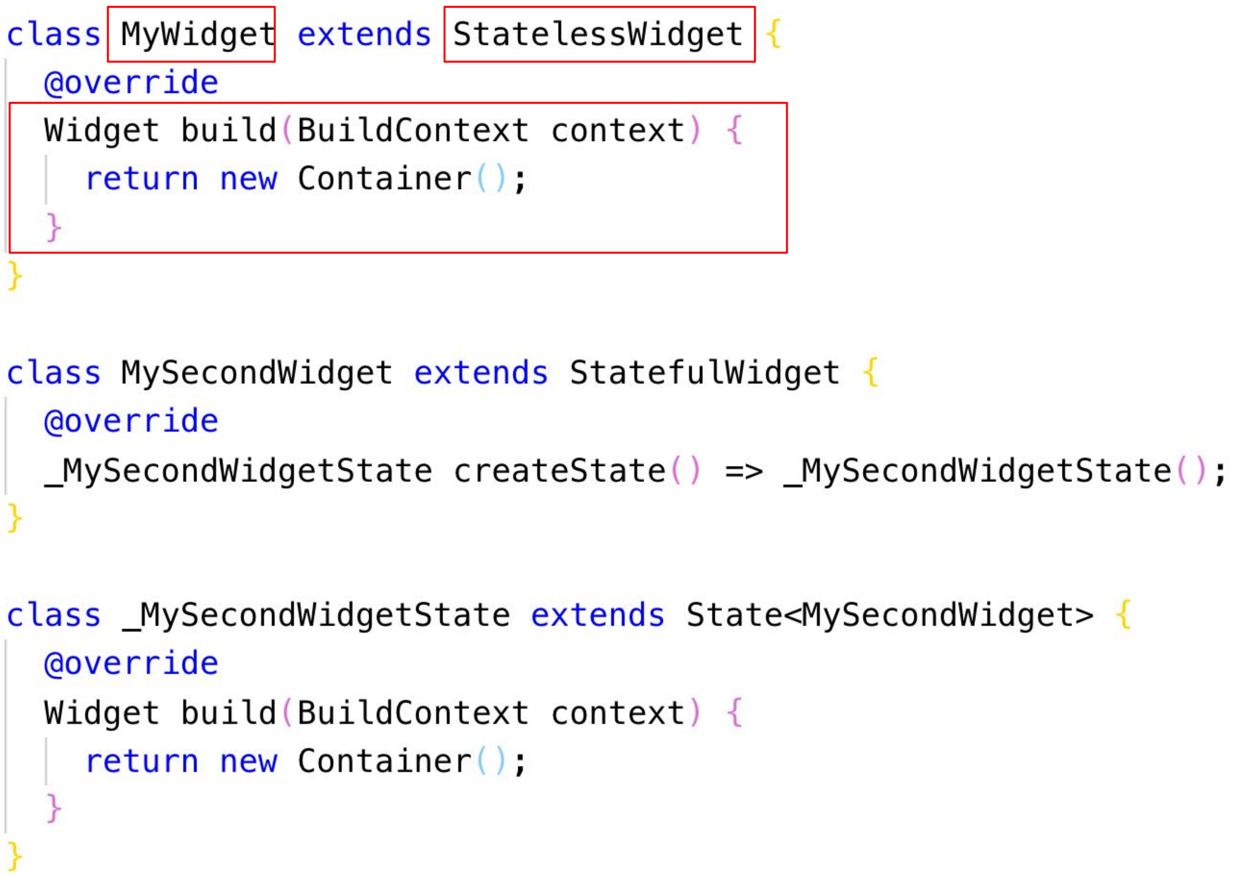
```
class MyWidget extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return new Container();
  }
}

class MySecondWidget extends StatefulWidget {
  @override
  _MySecondWidgetState createState() => _MySecondWidgetState();
}

class _MySecondWidgetState extends State<MySecondWidget> {
  @override
  Widget build(BuildContext context) {
    return new Container();
  }
}
```

Flutter Widgets



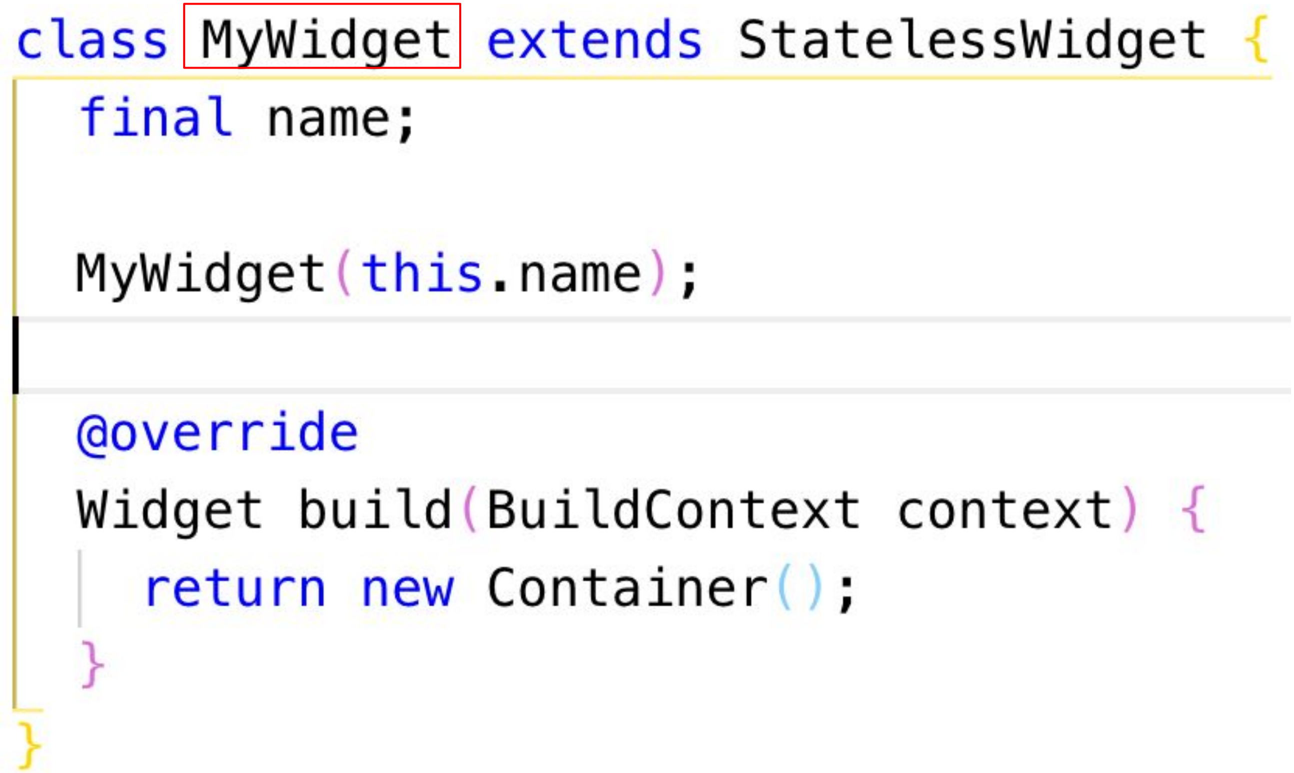
```
class MyWidget extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return new Container();
  }
}

class MySecondWidget extends StatefulWidget {
  @override
  _MySecondWidgetState createState() => _MySecondWidgetState();
}

class _MySecondWidgetState extends State<MySecondWidget> {
  @override
  Widget build(BuildContext context) {
    return new Container();
  }
}
```

Flutter Widgets



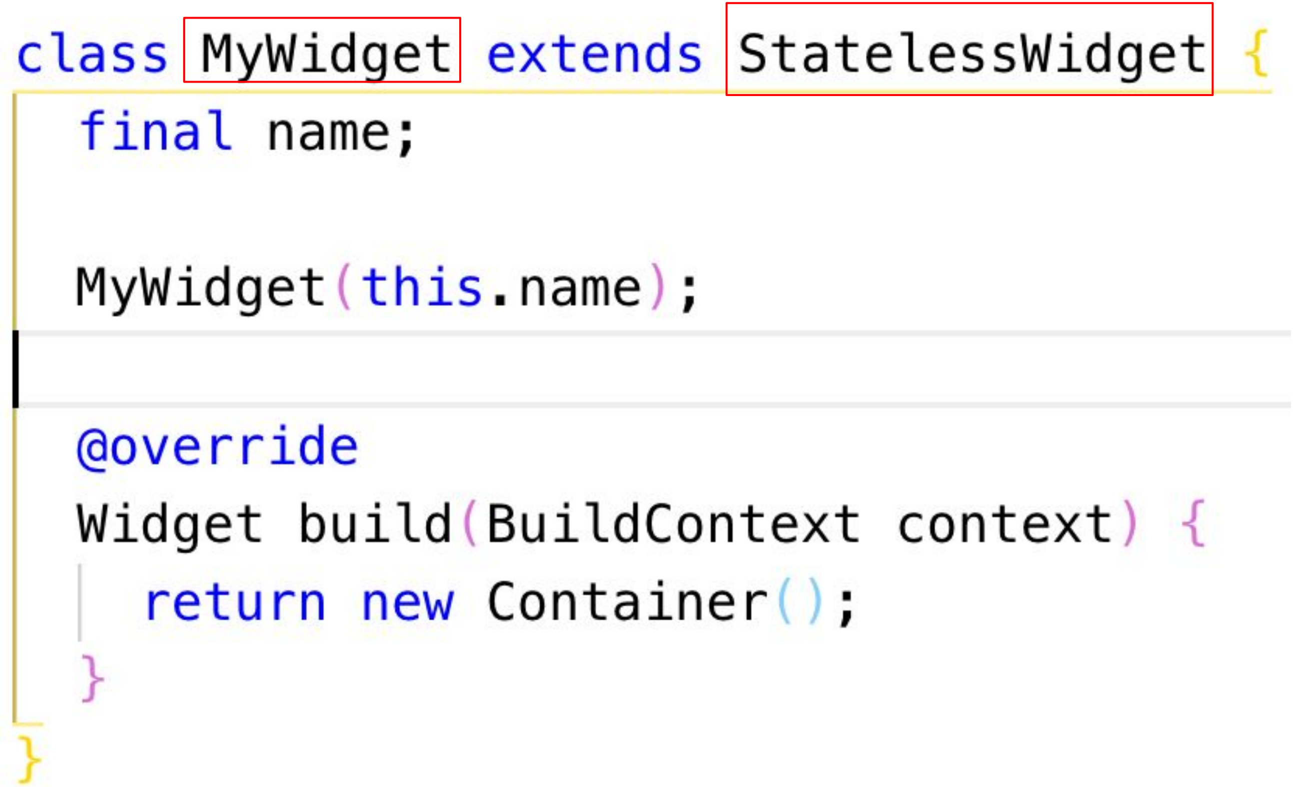
```
class MyWidget extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return new Container();
  }
}

class MySecondWidget extends StatefulWidget {
  @override
  _MySecondWidgetState createState() => _MySecondWidgetState();
}

class _MySecondWidgetState extends State<MySecondWidget> {
  @override
  Widget build(BuildContext context) {
    return new Container();
  }
}
```

Flutter Widgets



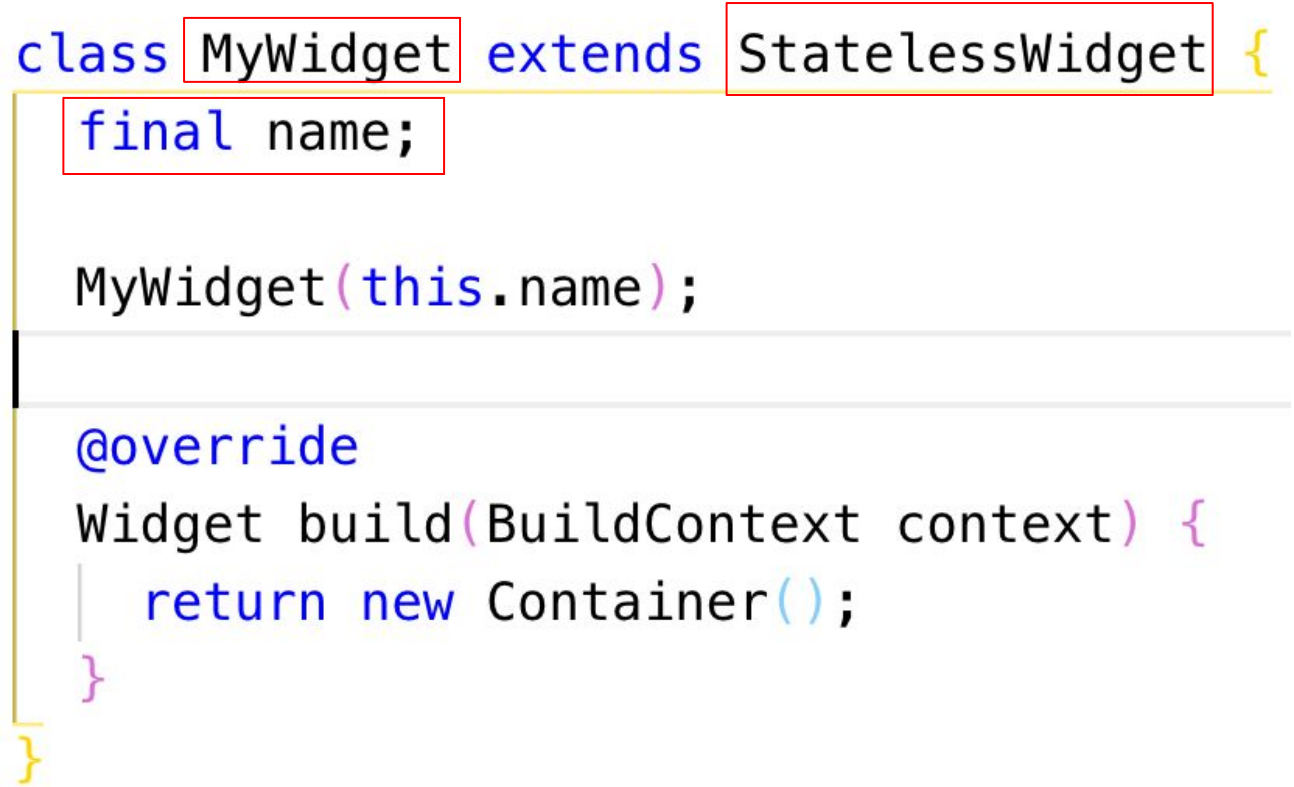
```
class MyWidget extends StatelessWidget {
  final name;

  MyWidget(this.name);

  @override
  Widget build(BuildContext context) {
    return new Container();
  }
}
```

Flutter Widgets



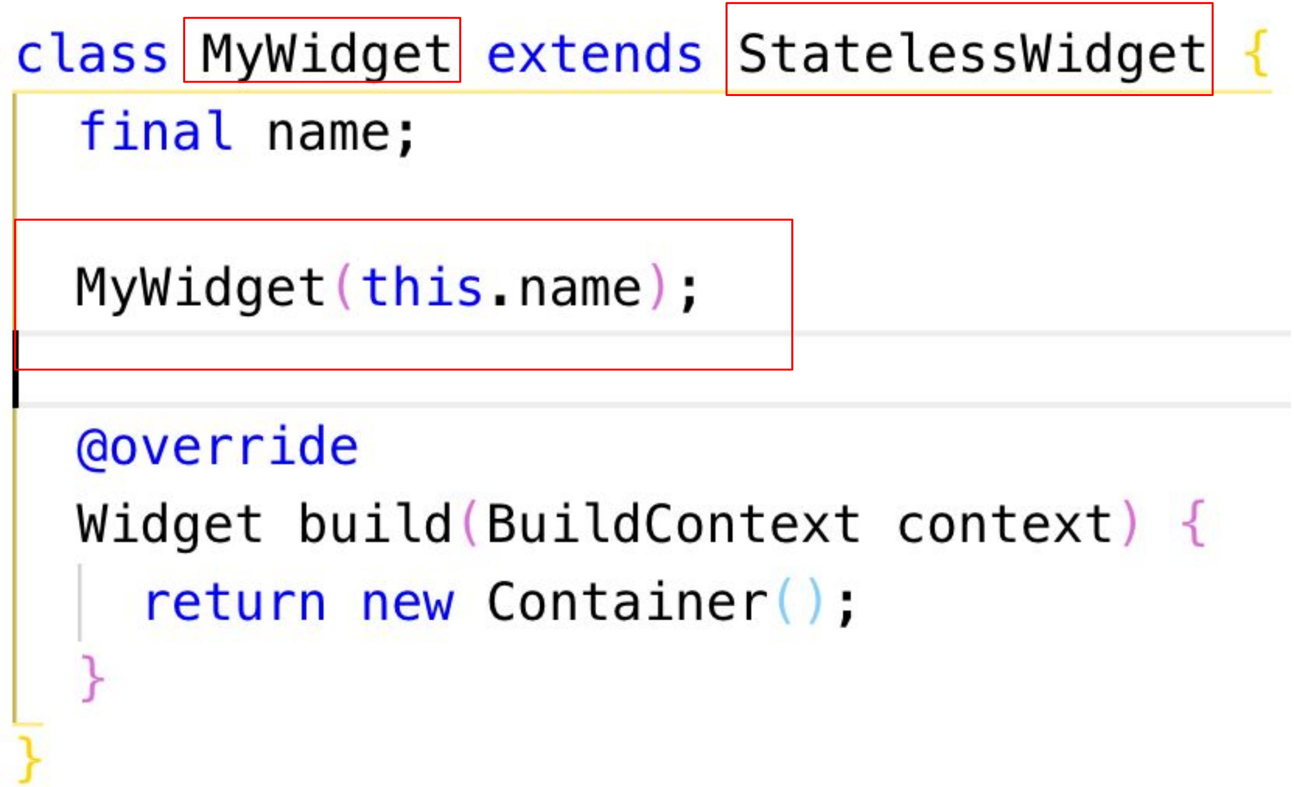
```
class MyWidget extends StatelessWidget {
  final name;

  MyWidget(this.name);

  @override
  Widget build(BuildContext context) {
    return new Container();
  }
}
```

Flutter Widgets



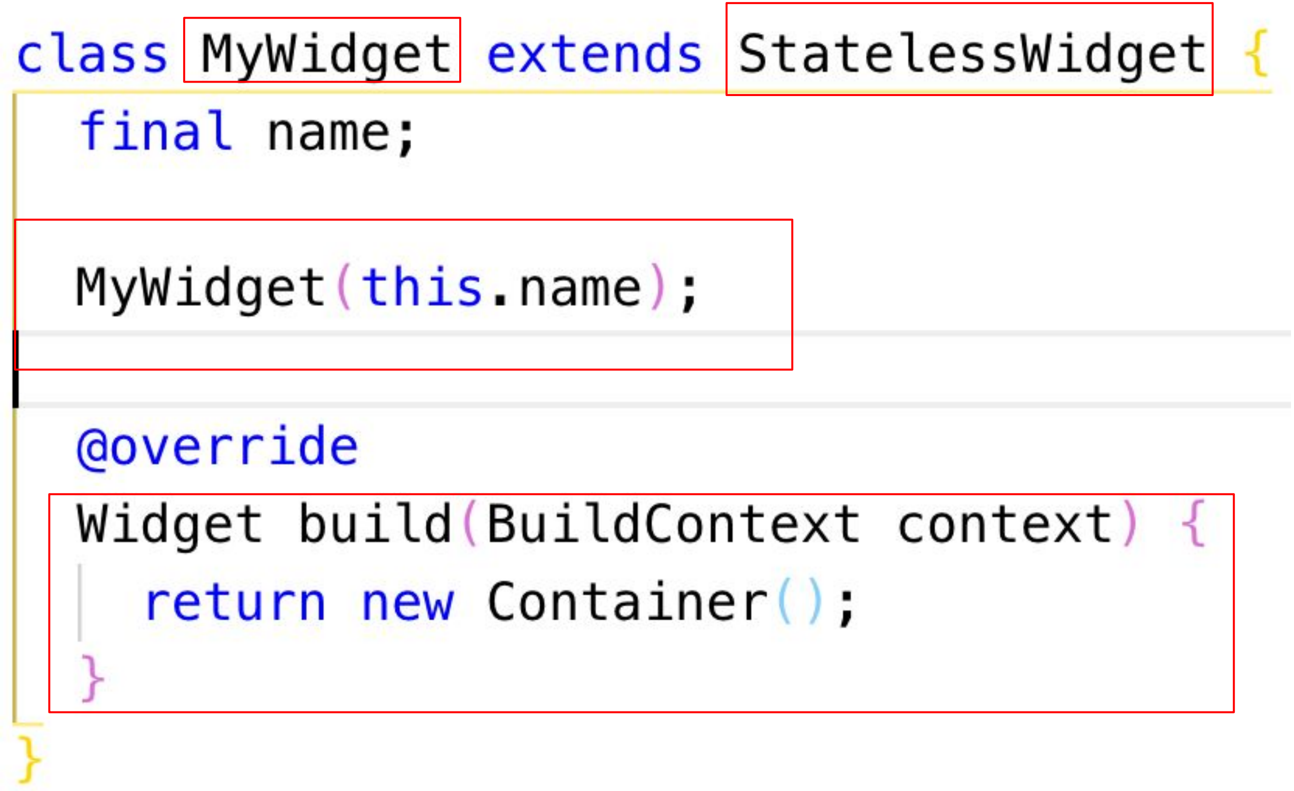
```
class MyWidget extends StatelessWidget {
  final name;

  MyWidget(this.name);

  @override
  Widget build(BuildContext context) {
    return new Container();
  }
}
```

Flutter Widgets



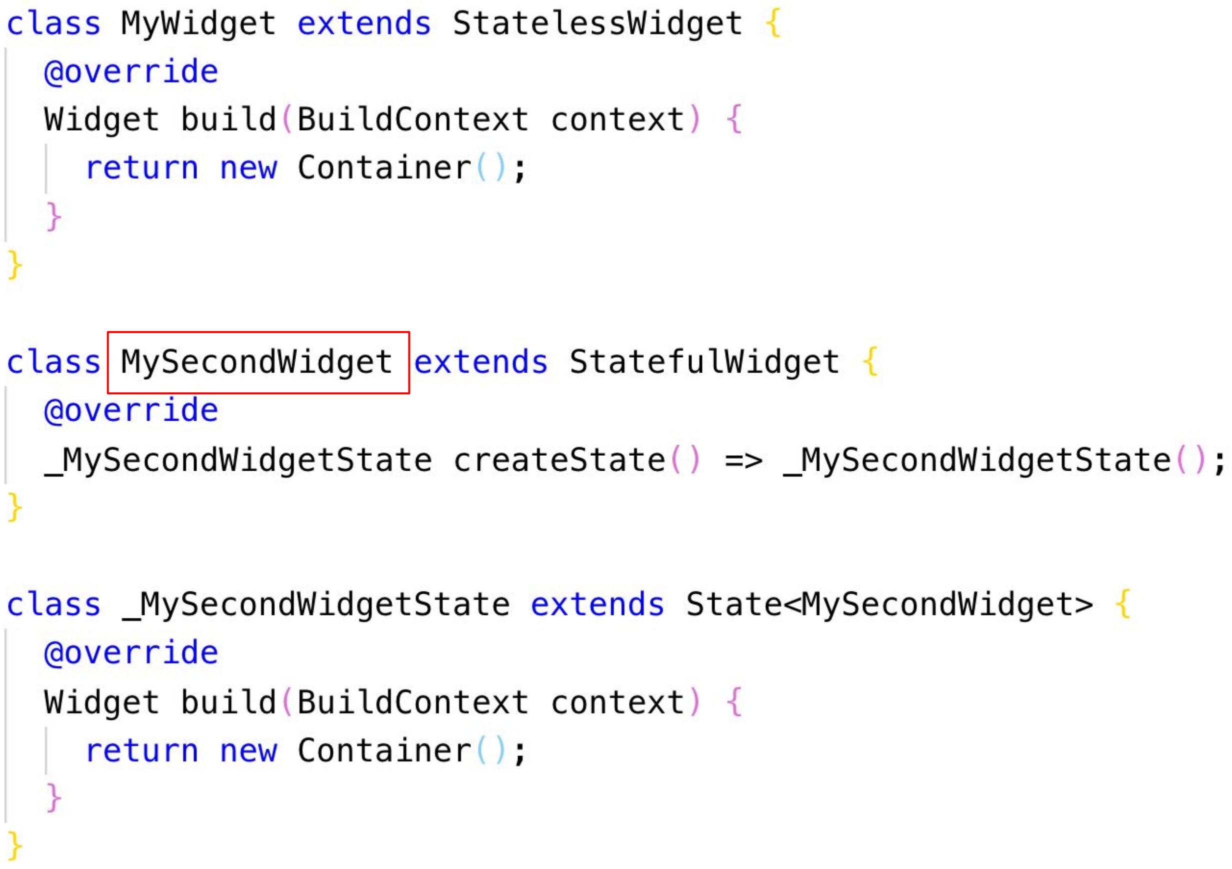
```
class MyWidget extends StatelessWidget {
  final name;

  MyWidget(this.name);


  @override
  Widget build(BuildContext context) {
    return new Container();
  }
}
```

Flutter Widgets



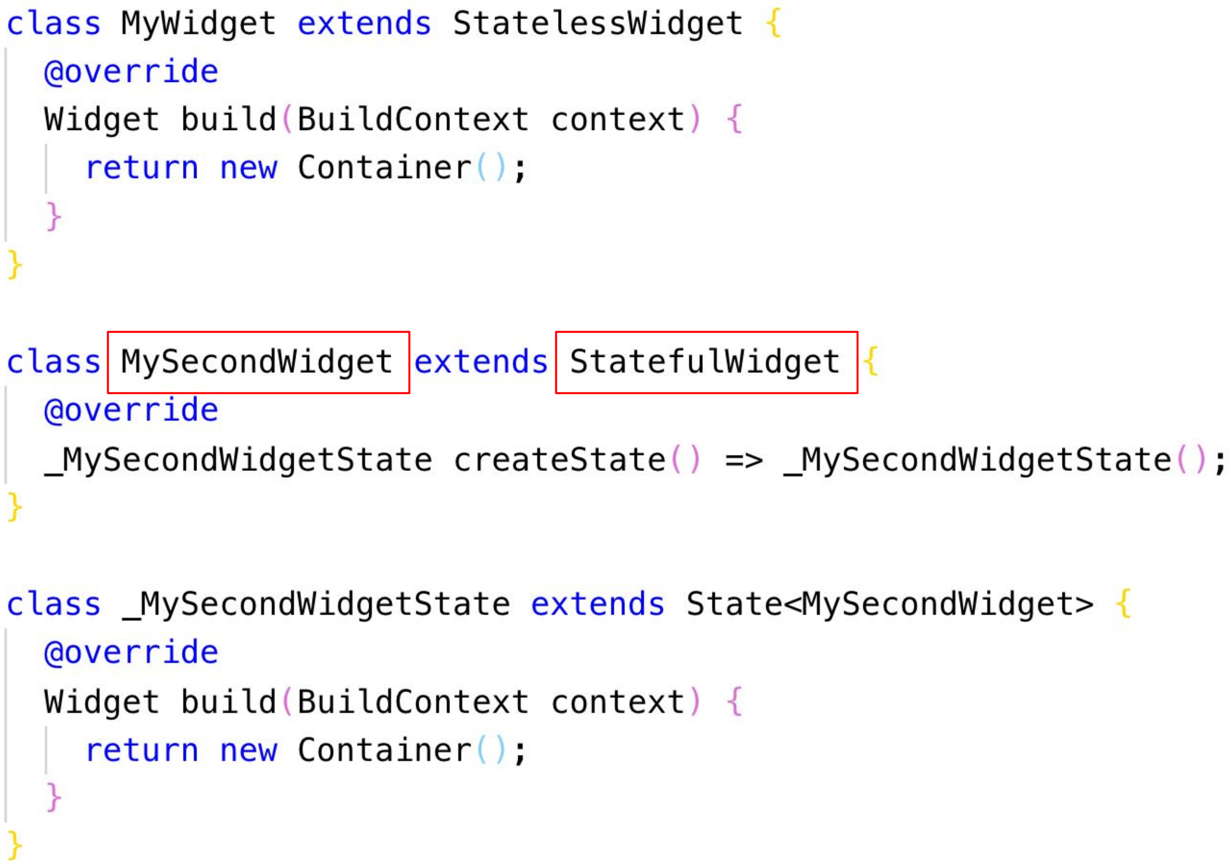
```
class MyWidget extends StatelessWidget {
  final name;

  MyWidget(this.name);

  @override
  Widget build(BuildContext context) {
    return new Container();
  }
}
```

Flutter Widgets



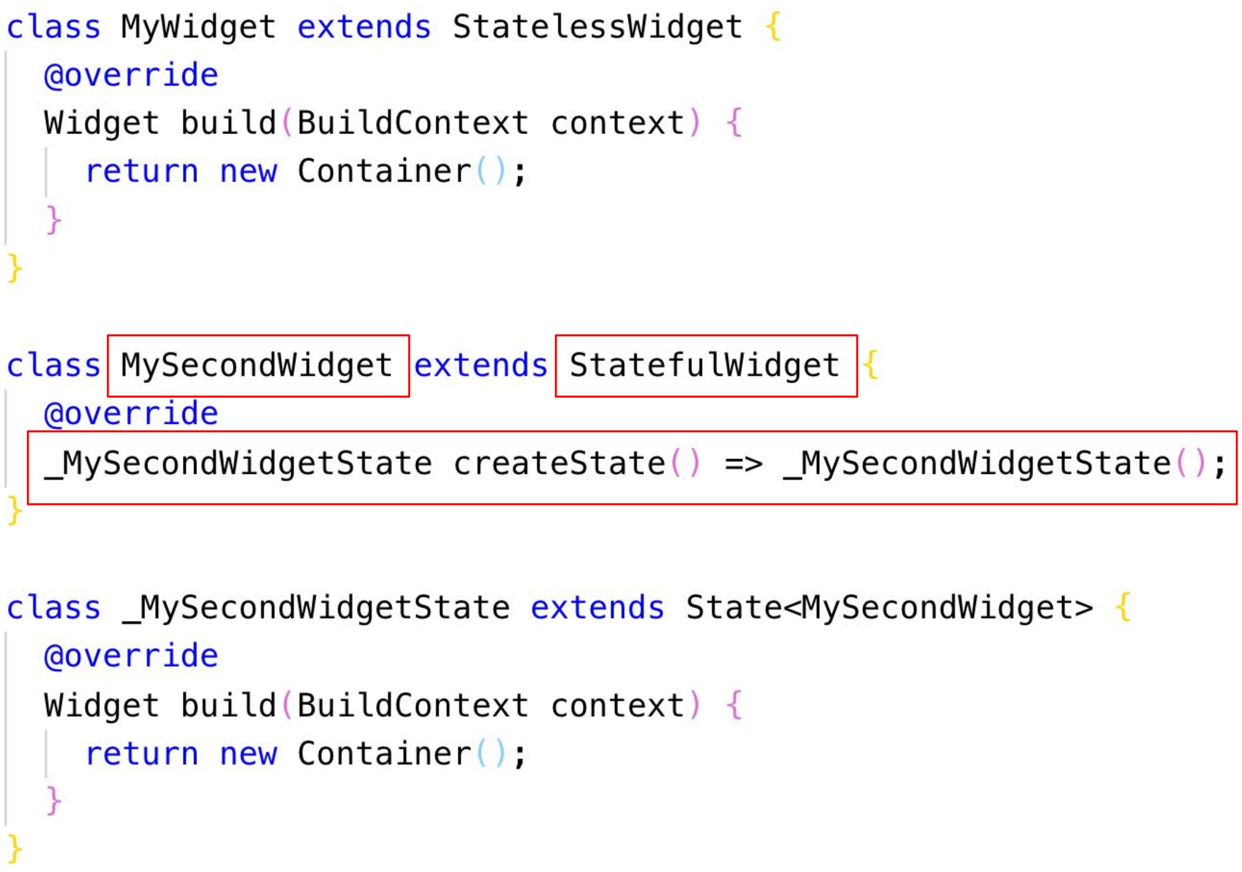
```dart
class MyWidget extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return new Container();
  }
}

class MySecondWidget extends StatefulWidget {
  @override
  _MySecondWidgetState createState() => _MySecondWidgetState();
}

class _MySecondWidgetState extends State<MySecondWidget> {
  @override
  Widget build(BuildContext context) {
    return new Container();
  }
}
```

Flutter Widgets



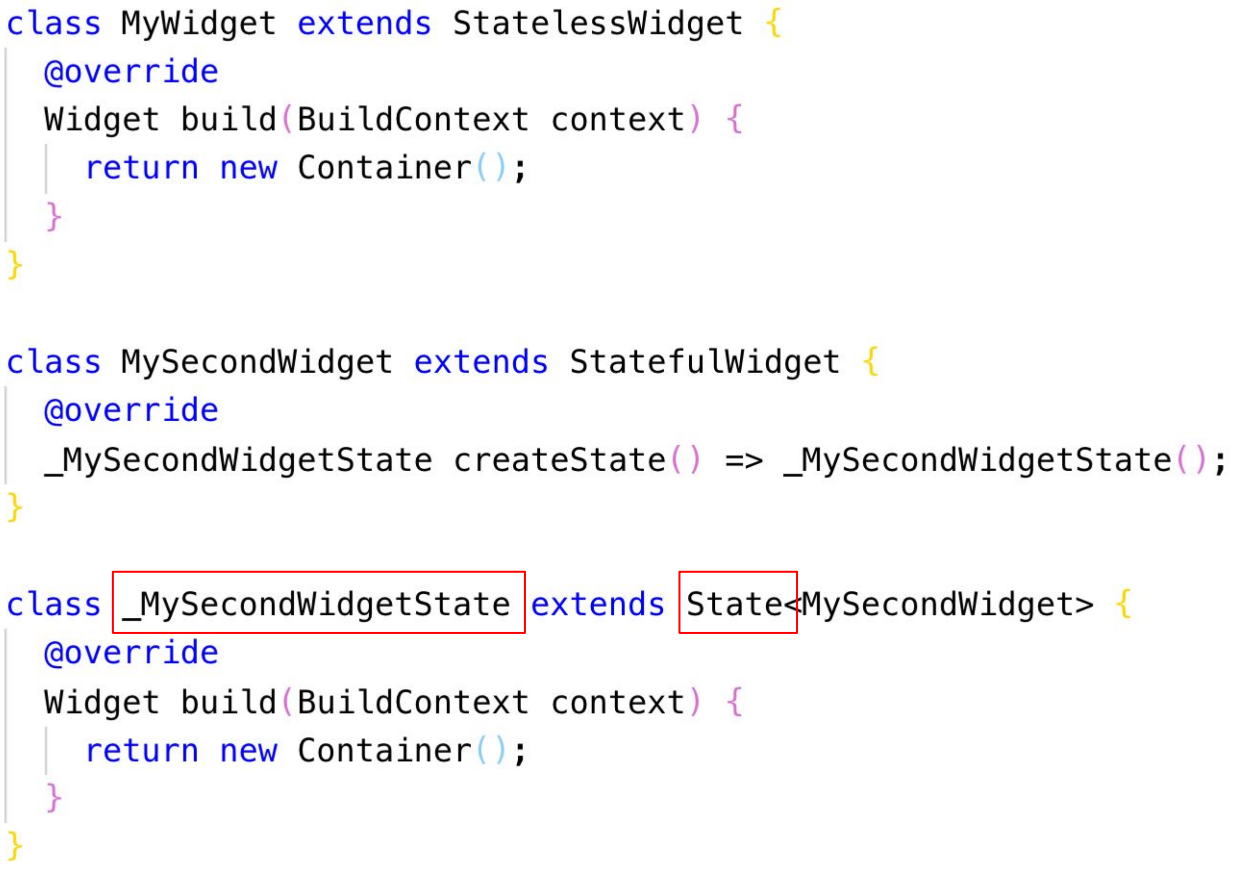
```
class MyWidget extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return new Container();
  }
}

class MySecondWidget extends StatefulWidget {
  @override
  _MySecondWidgetState createState() => _MySecondWidgetState();
}

class _MySecondWidgetState extends State<MySecondWidget> {
  @override
  Widget build(BuildContext context) {
    return new Container();
  }
}
```

Flutter Widgets



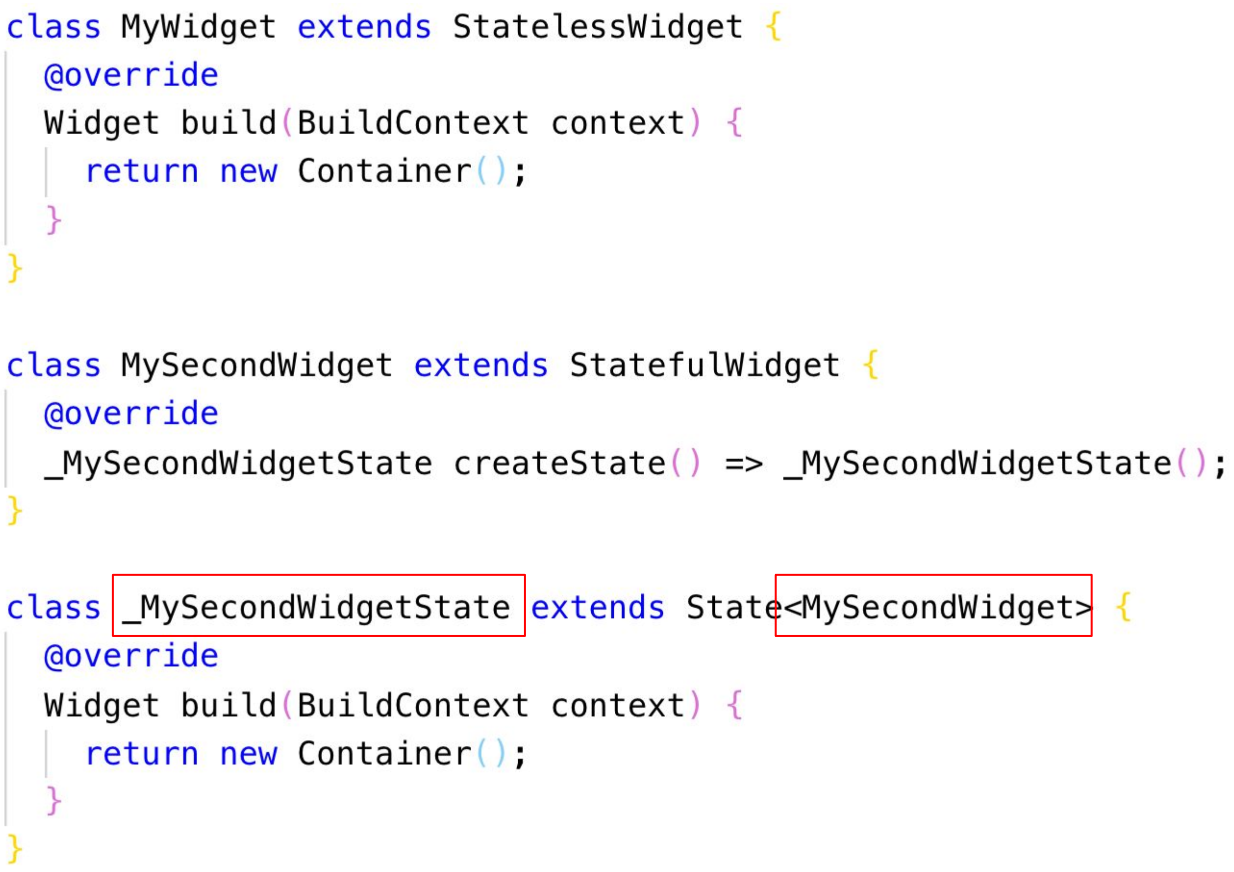
```
class MyWidget extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return new Container();
  }
}

class MySecondWidget extends StatefulWidget {
  @override
  _MySecondWidgetState createState() => _MySecondWidgetState();
}

class _MySecondWidgetState extends State<MySecondWidget> {
  @override
  Widget build(BuildContext context) {
    return new Container();
  }
}
```

Flutter Widgets



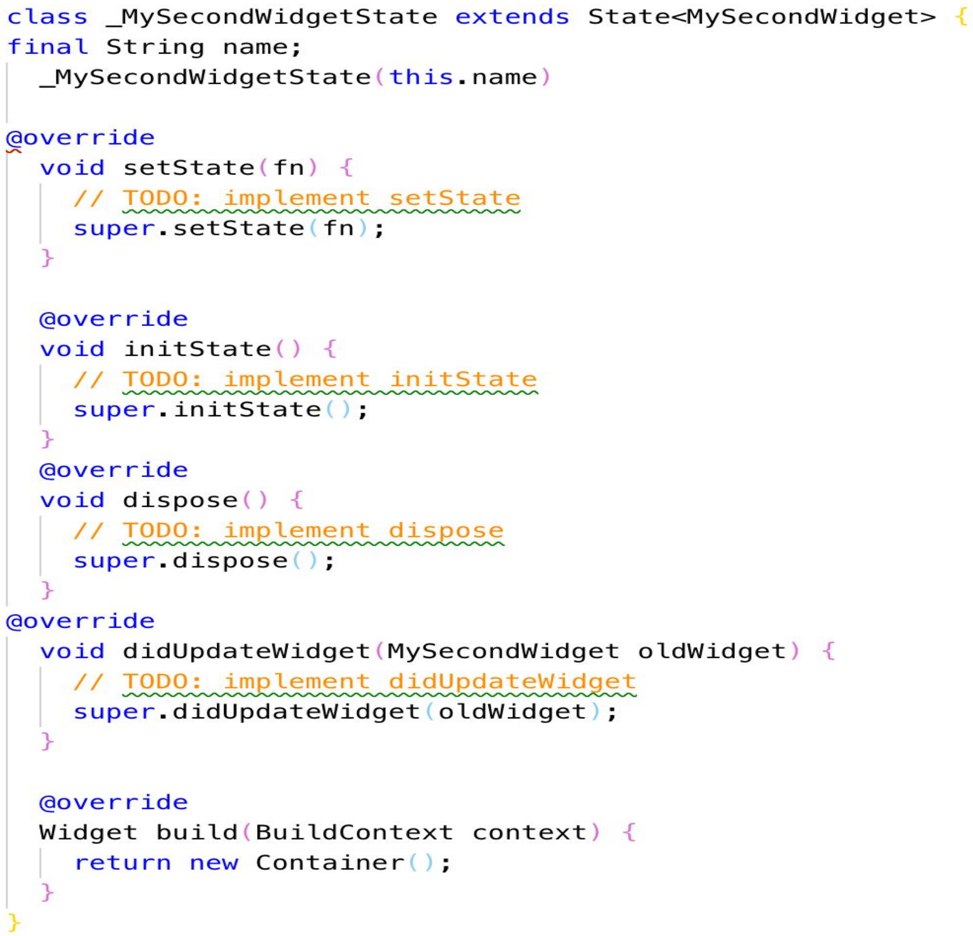
```
class MyWidget extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return new Container();
  }
}

class MySecondWidget extends StatefulWidget {
  @override
  _MySecondWidgetState createState() => _MySecondWidgetState();
}

class _MySecondWidgetState extends State<MySecondWidget> {
  @override
  Widget build(BuildContext context) {
    return new Container();
  }
}
```

Flutter Widgets



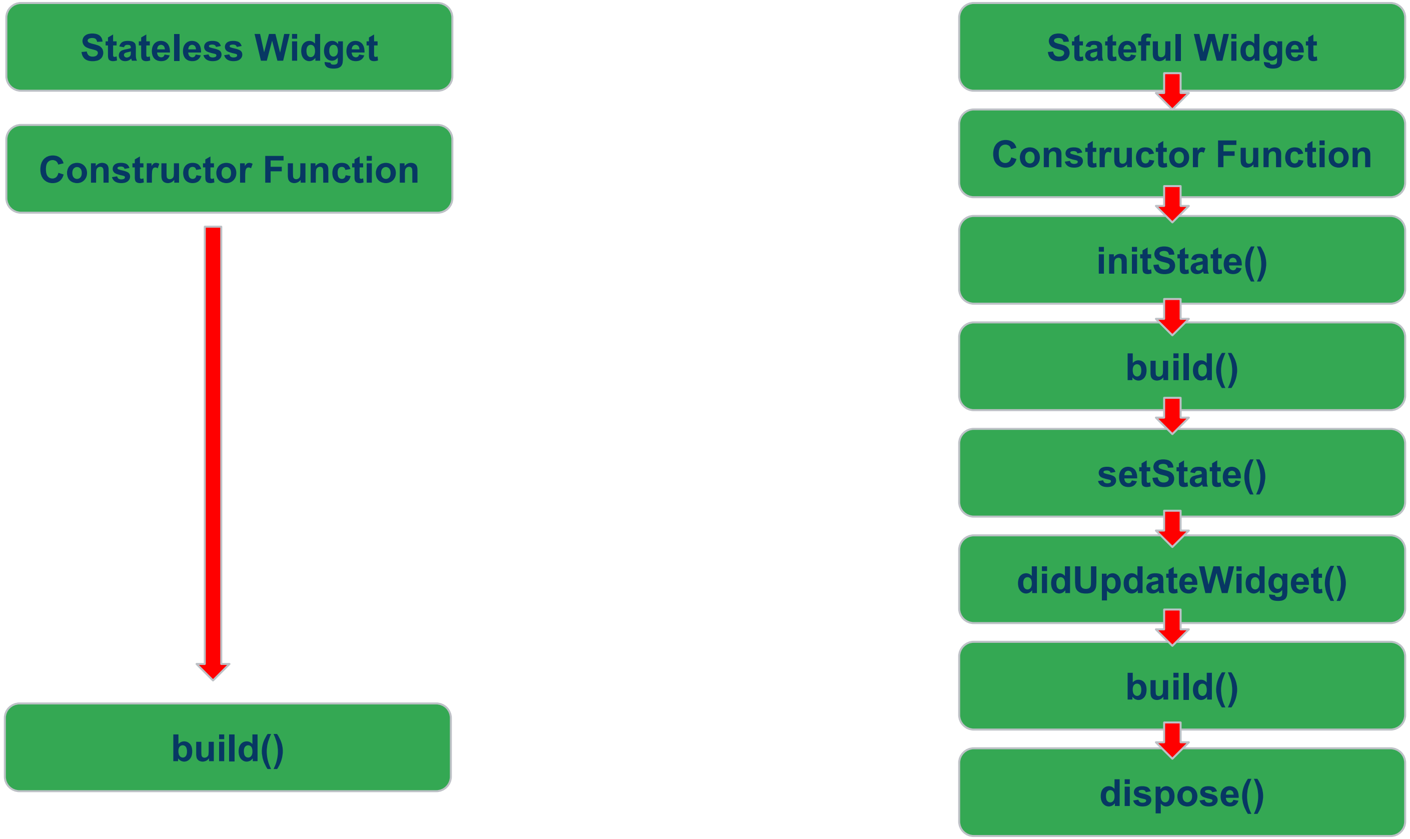
```
class MyWidget extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return new Container();
  }
}

class MySecondWidget extends StatefulWidget {
  @override
  _MySecondWidgetState createState() => _MySecondWidgetState();
}

class _MySecondWidgetState extends State<MySecondWidget> {
  @override
  Widget build(BuildContext context) {
    return new Container();
  }
}
```

Flutter Widgets



```
class _MySecondWidgetState extends State<MySecondWidget> {
final String name;
  _MySecondWidgetState(this.name)

@override
  void setState(fn) {
    // TODO: implement setState
    super.setState(fn);
  }

  @override
  void initState() {
    // TODO: implement initState
    super.initState();
  }
  @override
  void dispose() {
    // TODO: implement dispose
    super.dispose();
  }
@override
  void didUpdateWidget(MySecondWidget oldWidget) {
    // TODO: implement didUpdateWidget
    super.didUpdateWidget(oldWidget);
  }

  @override
  Widget build(BuildContext context) {
    return new Container();
  }
}
```

Flutter Widgets



Stateless Widget

Constructor Function

build()

Stateful Widget

Constructor Function

initState()

build()

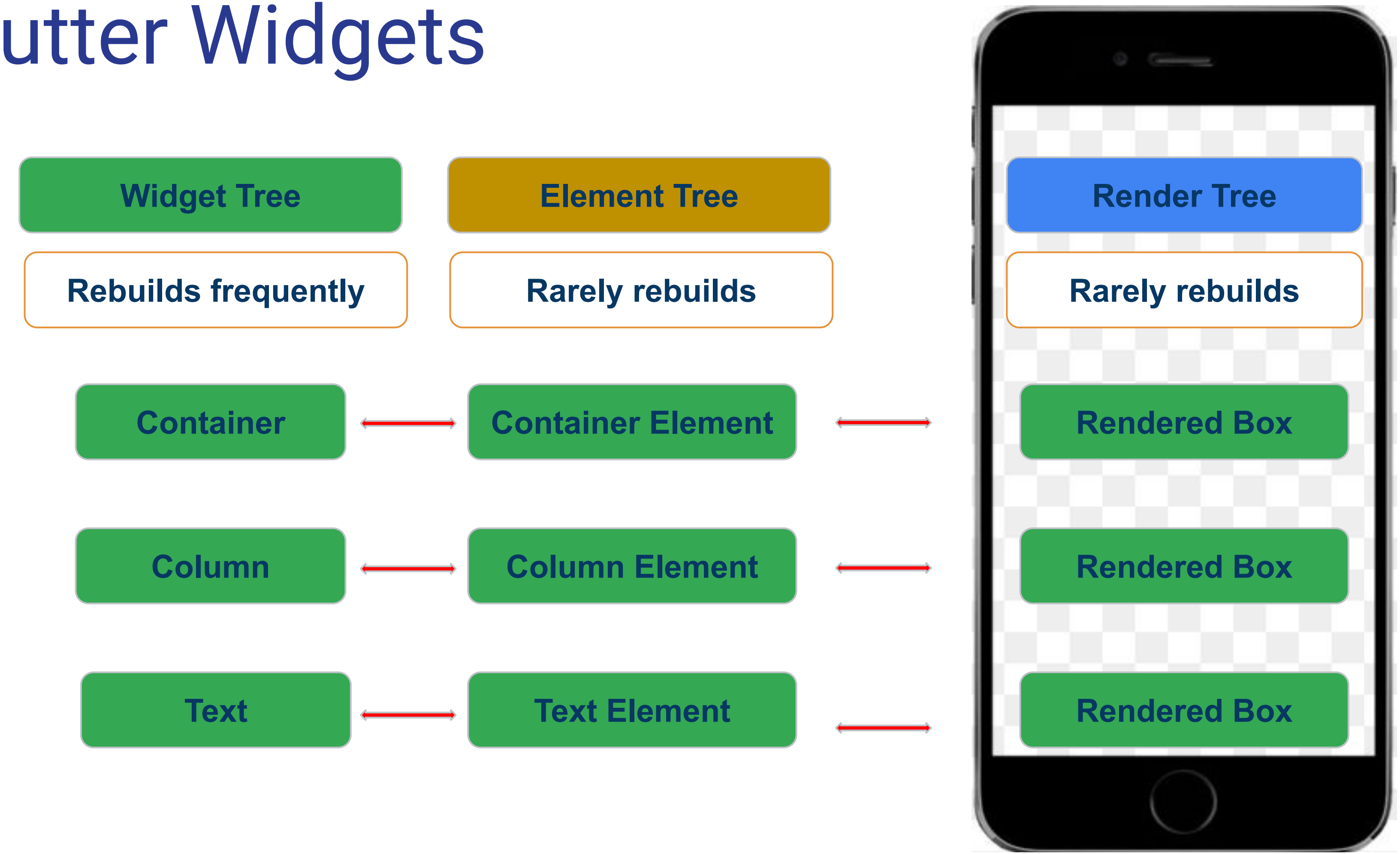
setState()

didUpdateWidget()

build()

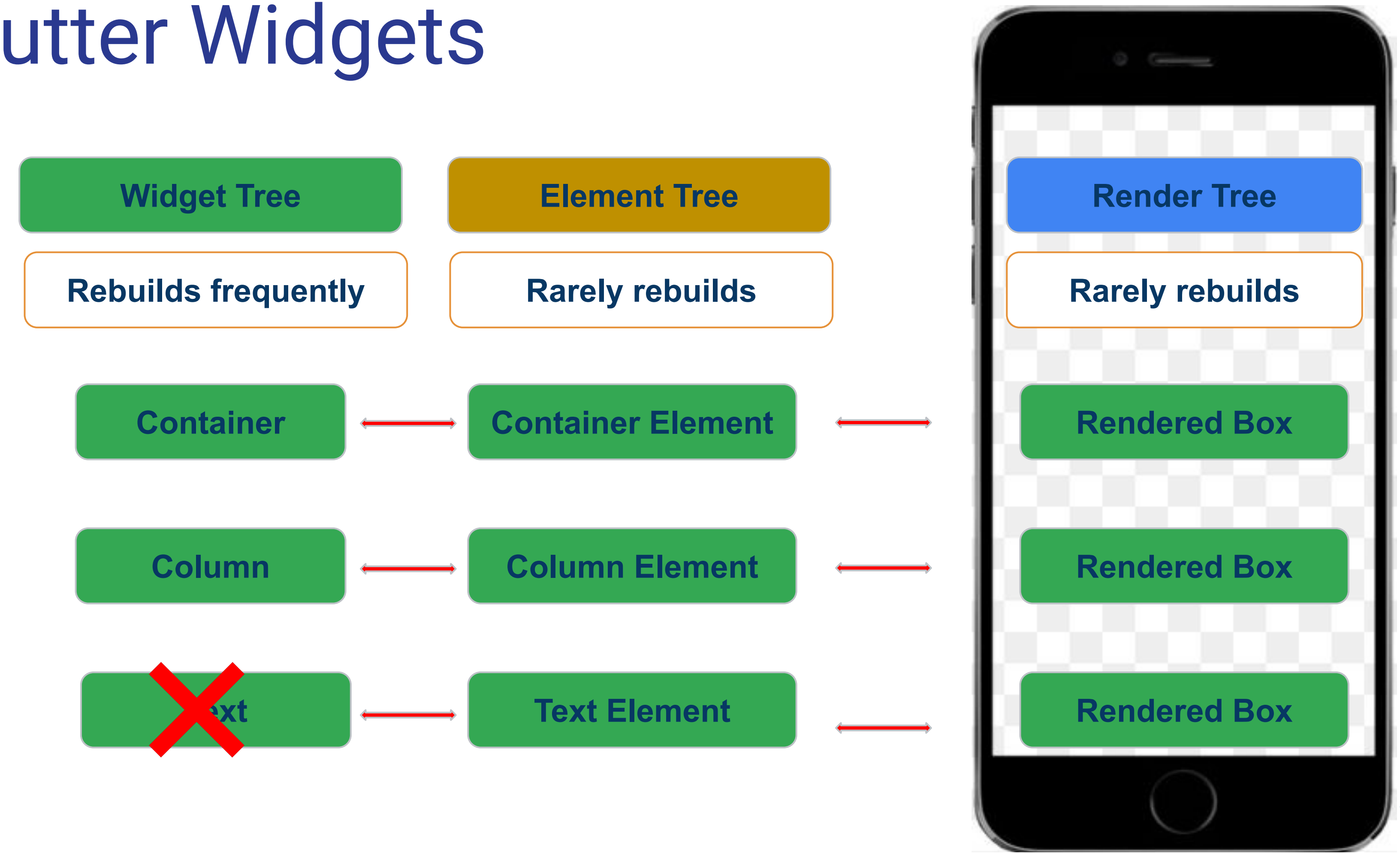
dispose()

Flutter Widgets



Widget Tree	Element Tree	Render Tree
Rebuilds frequently	Rarely rebuilds	Rarely rebuilds
Container	Container Element	Rendered Box
Column	Column Element	Rendered Box
Text	Text Element	Rendered Box

Flutter Widgets



Widget Tree

Rebuilds frequently

Element Tree

Rarely rebuilds

Render Tree

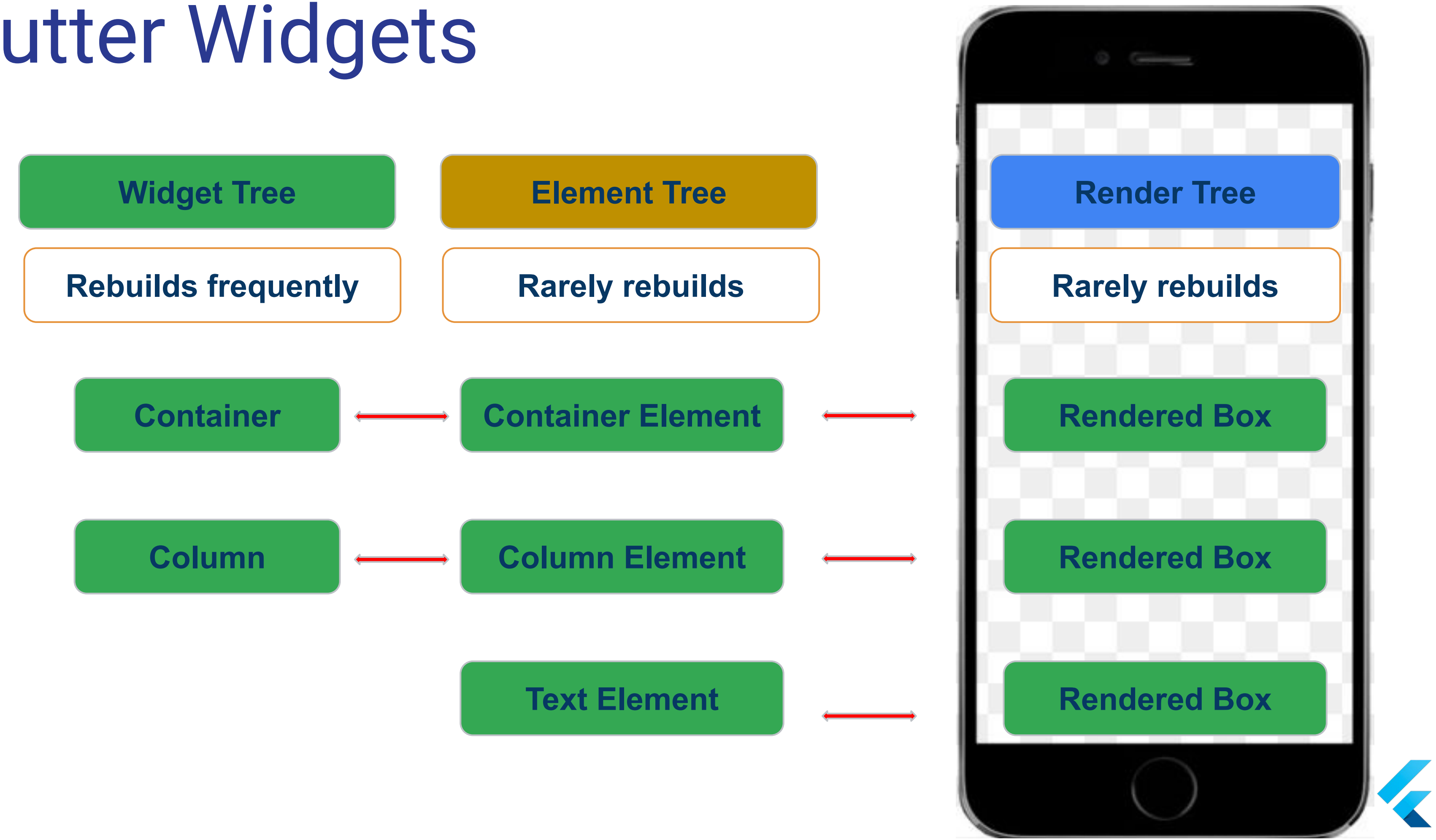
Rarely rebuilds

Container ↔ Container Element ↔ Rendered Box

Column ↔ Column Element ↔ Rendered Box

~~Text~~ ↔ Text Element ↔ Rendered Box

Flutter Widgets



Widget Tree

Rebuilds frequently

Element Tree

Rarely rebuilds

Render Tree

Rarely rebuilds

Container

Container Element

Rendered Box

Column

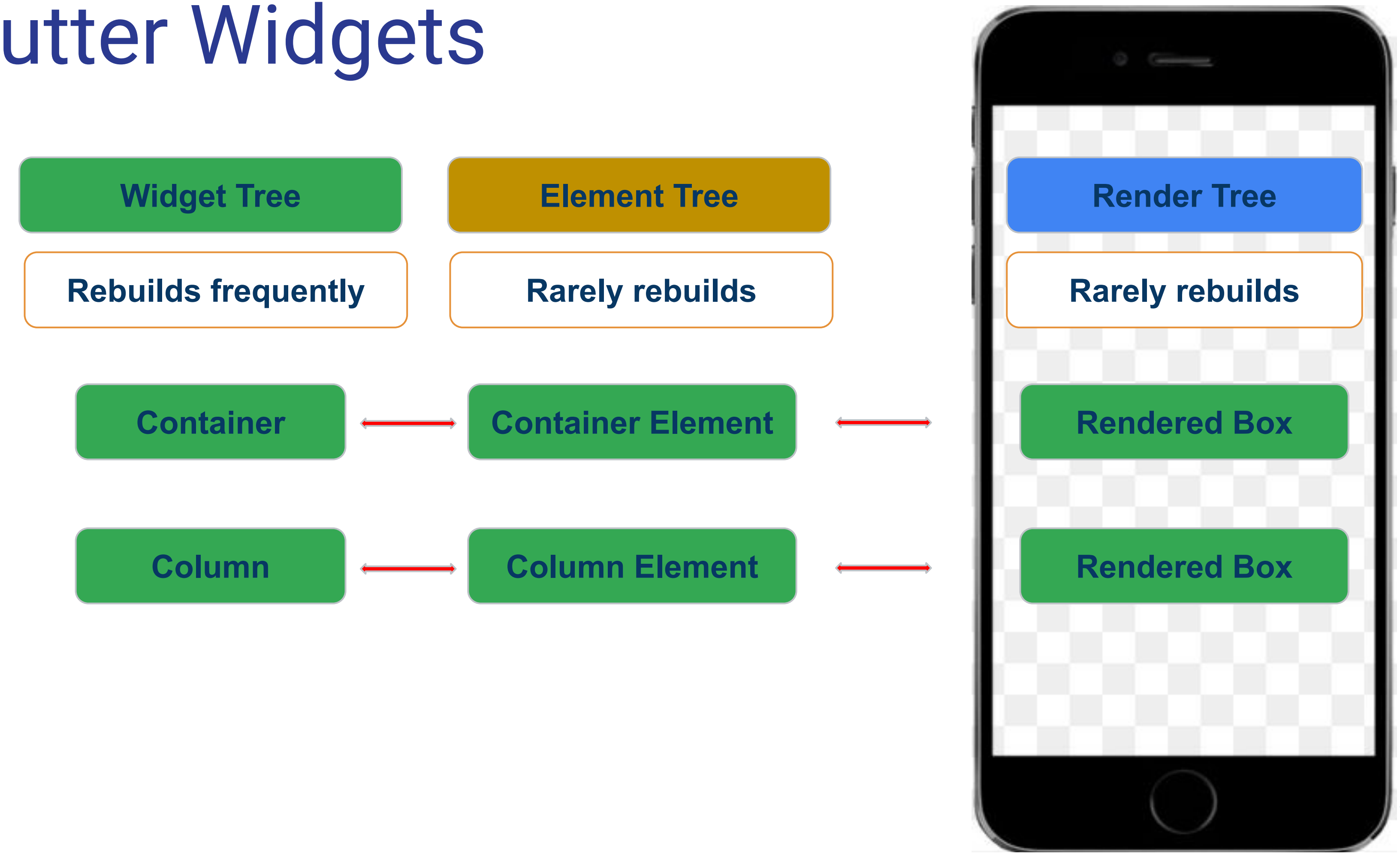
Column Element

Rendered Box

Text Element

Rendered Box

Flutter Widgets



Widget Tree

Rebuilds frequently

Element Tree

Rarely rebuilds

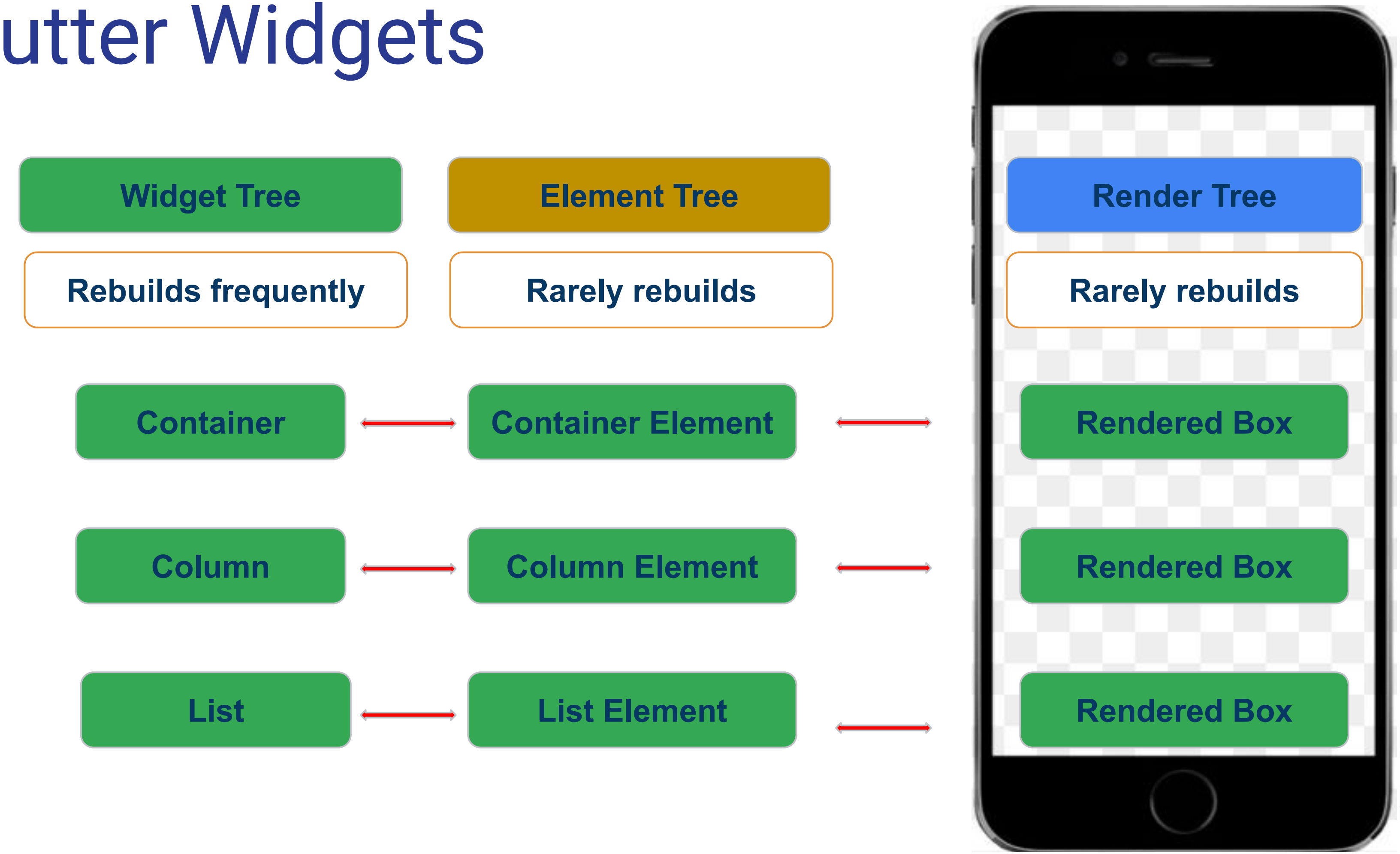
Render Tree

Rarely rebuilds

Container ↔ Container Element ↔ Rendered Box

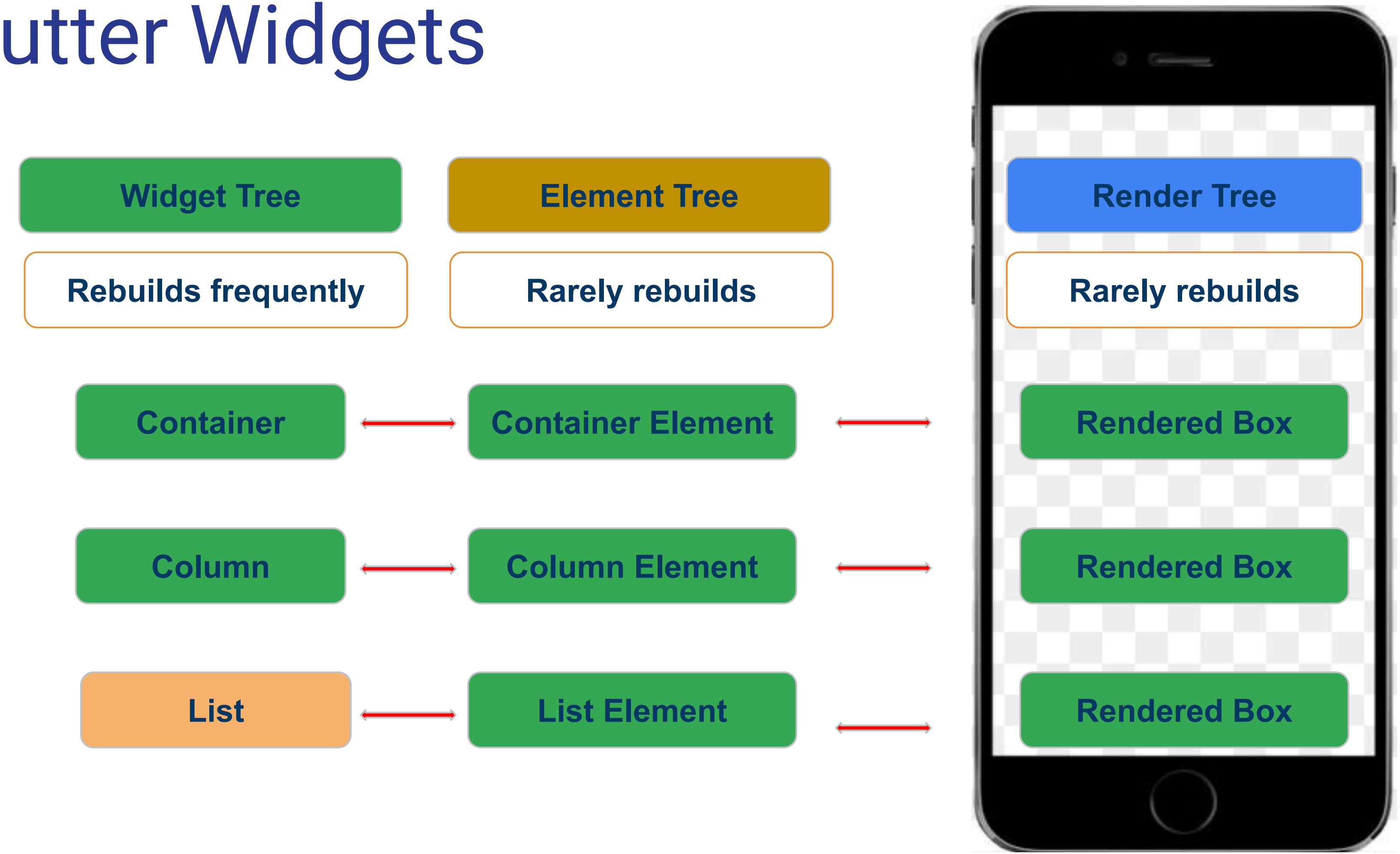
Column ↔ Column Element ↔ Rendered Box

Flutter Widgets



Widget Tree	Element Tree	Render Tree
Rebuilds frequently	Rarely rebuilds	Rarely rebuilds
Container	Container Element	Rendered Box
Column	Column Element	Rendered Box
List	List Element	Rendered Box

Flutter Widgets



Widget Tree

Rebuilds frequently

Element Tree

Rarely rebuilds

Render Tree

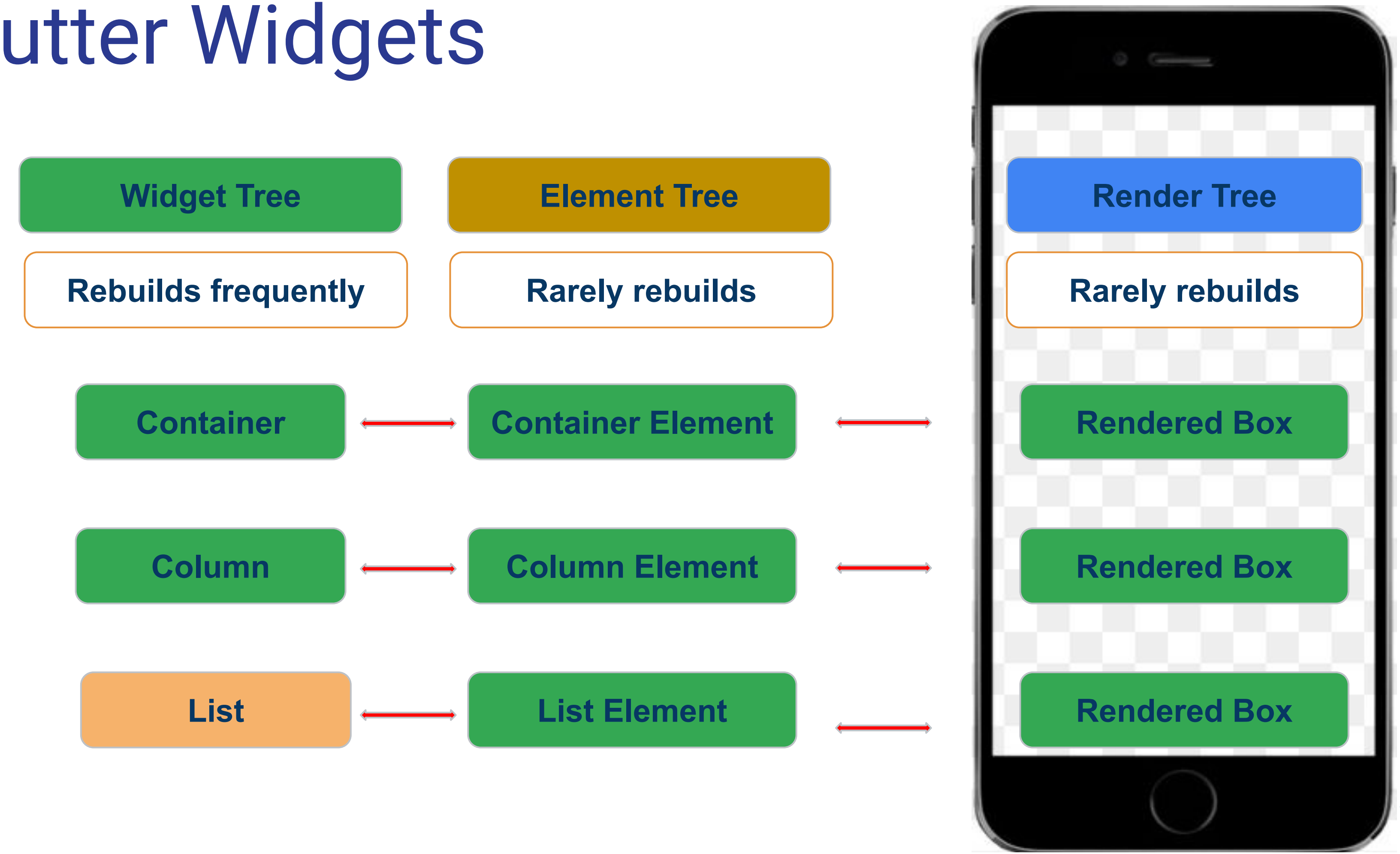
Rarely rebuilds

Container ↔ Container Element ↔ Rendered Box

Column ↔ Column Element ↔ Rendered Box

List ↔ List Element ↔ Rendered Box

Flutter Widgets



Widget Tree

Rebuilds frequently

Element Tree

Rarely rebuilds

Render Tree

Rarely rebuilds

Container

Container Element

Rendered Box

Column

Column Element

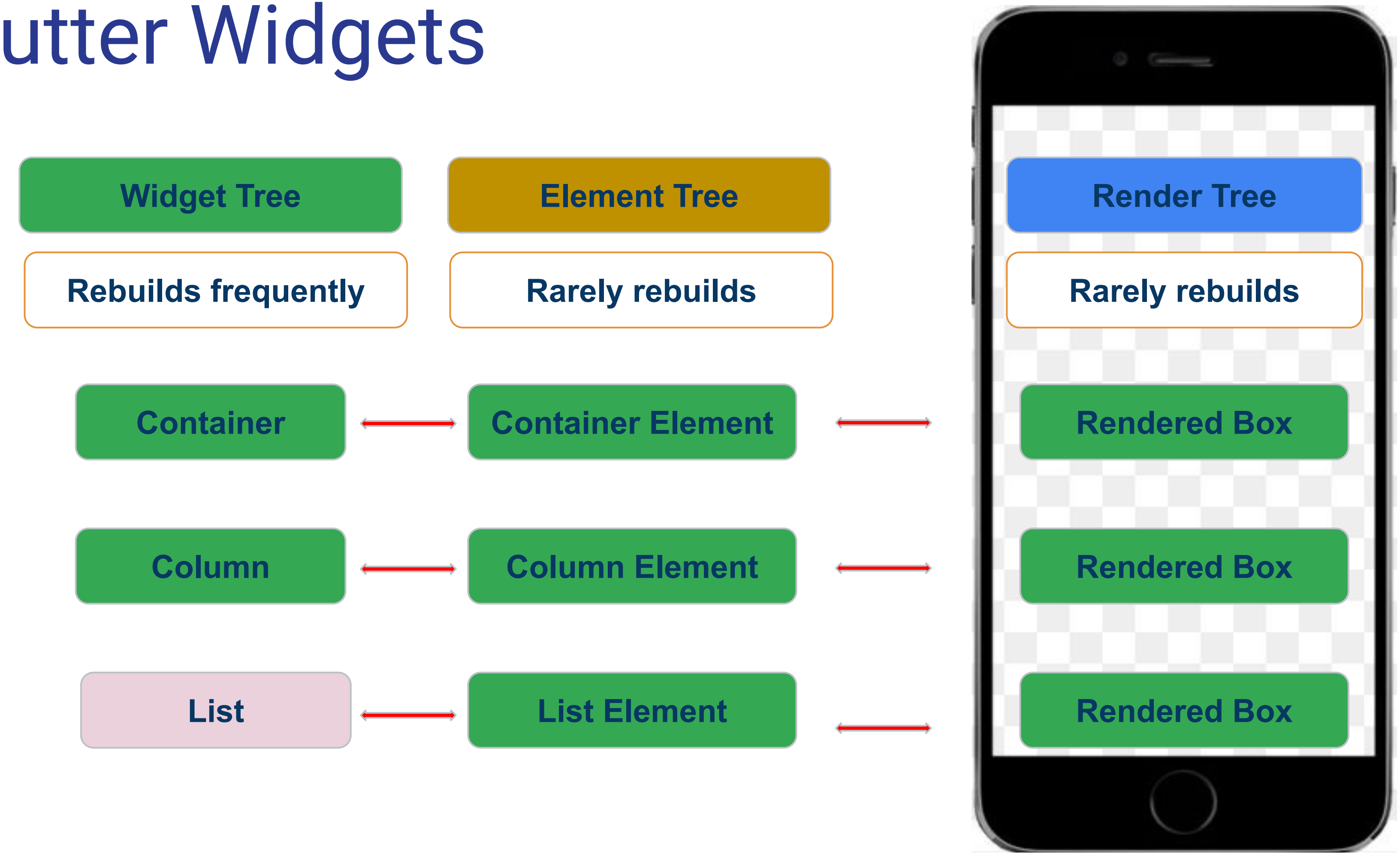
Rendered Box

List

List Element

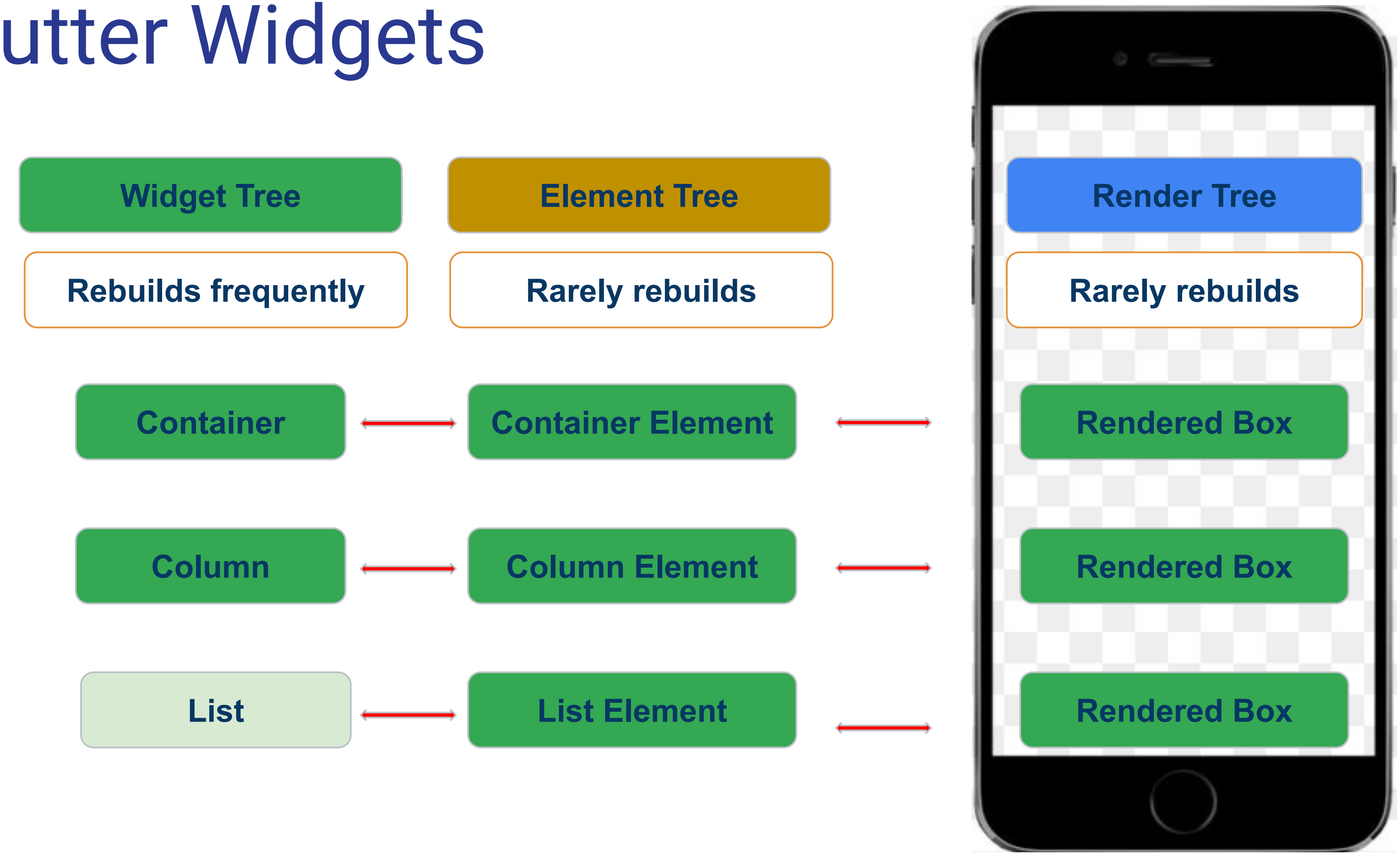
Rendered Box

Flutter Widgets



Widget Tree	Element Tree	Render Tree
Rebuilds frequently	Rarely rebuilds	Rarely rebuilds
Container	Container Element	Rendered Box
Column	Column Element	Rendered Box
List	List Element	Rendered Box

Flutter Widgets



Widget Tree

Rebuilds frequently

Element Tree

Rarely rebuilds

Render Tree

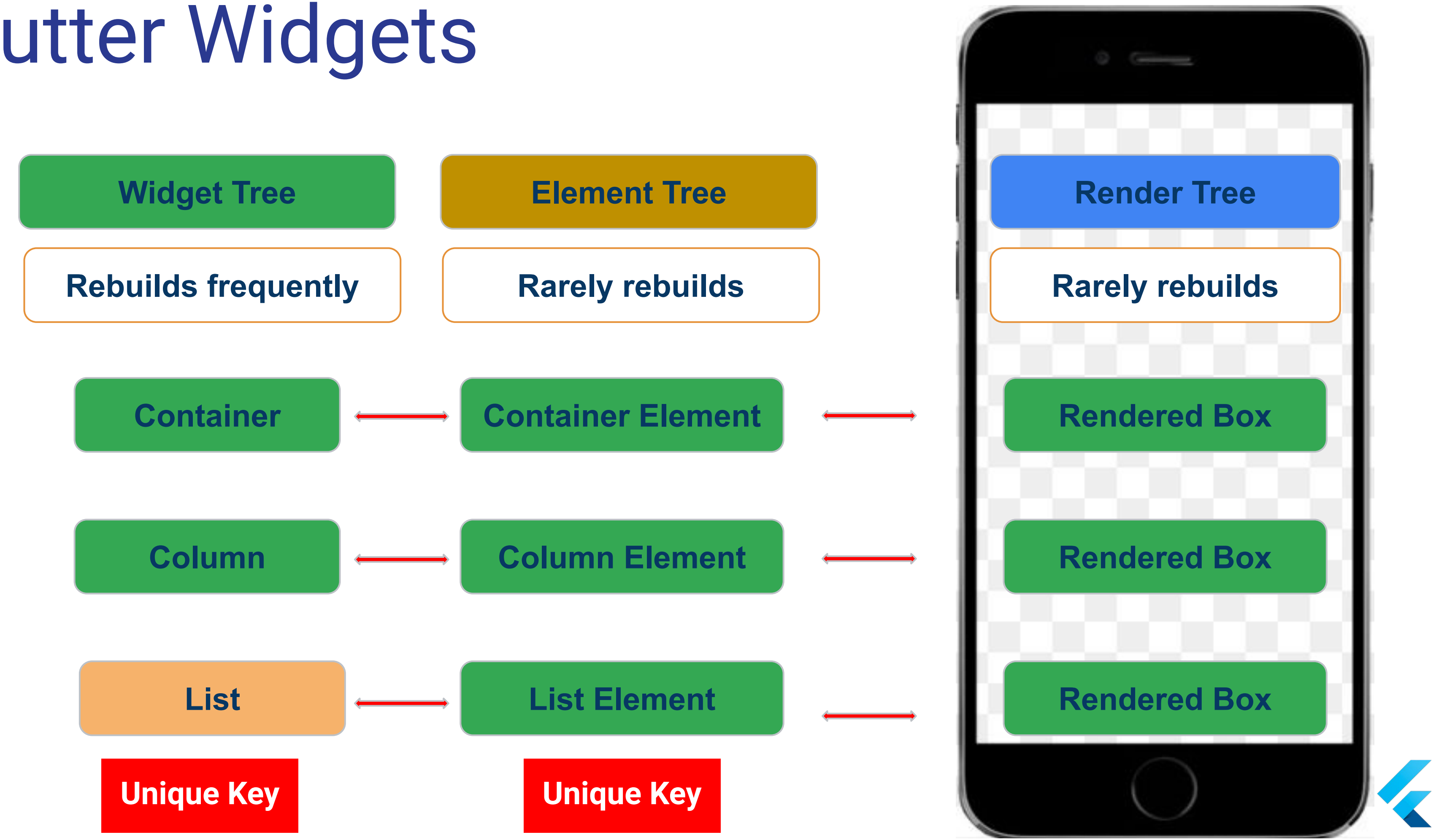
Rarely rebuilds

Container ⟷ Container Element ⟷ Rendered Box

Column ⟷ Column Element ⟷ Rendered Box

List ⟷ List Element ⟷ Rendered Box

Flutter Widgets



Widget Tree

Rebuilds frequently

Element Tree

Rarely rebuilds

Render Tree

Rarely rebuilds

Container ⟷ Container Element ⟷ Rendered Box

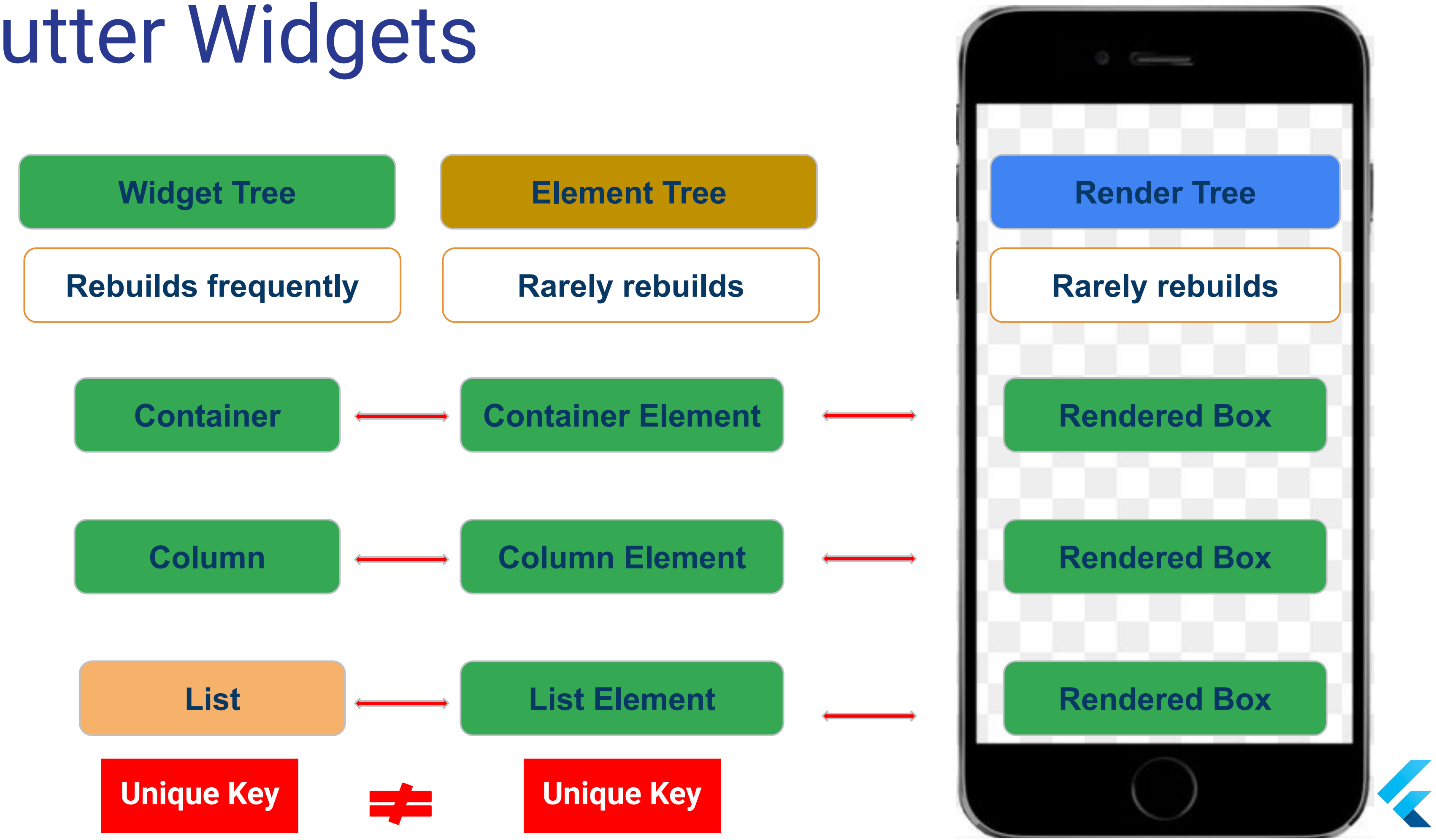
Column ⟷ Column Element ⟷ Rendered Box

List ⟷ List Element ⟷ Rendered Box

Unique Key

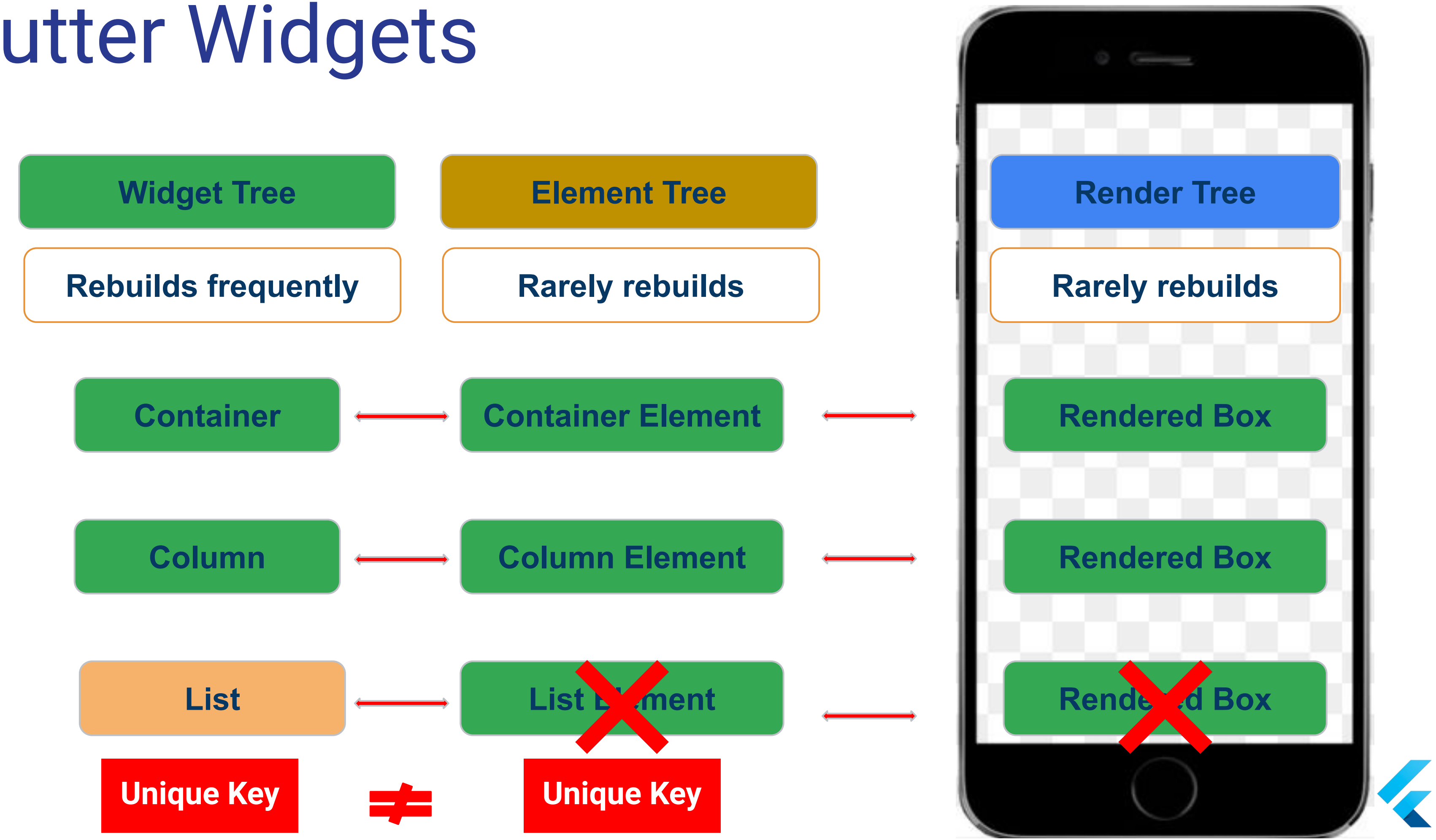
Unique Key

Flutter Widgets



Widget Tree	Element Tree	Render Tree
Rebuilds frequently	Rarely rebuilds	Rarely rebuilds
Container ⟷	Container Element ⟷	Rendered Box
Column ⟷	Column Element ⟷	Rendered Box
List ⟷	List Element ⟷	Rendered Box
Unique Key ≠	Unique Key	

Flutter Widgets



Widget Tree

Element Tree

Render Tree

Rebuilds frequently

Rarely rebuilds

Rarely rebuilds

Container

Container Element

Rendered Box

Column

Column Element

Rendered Box

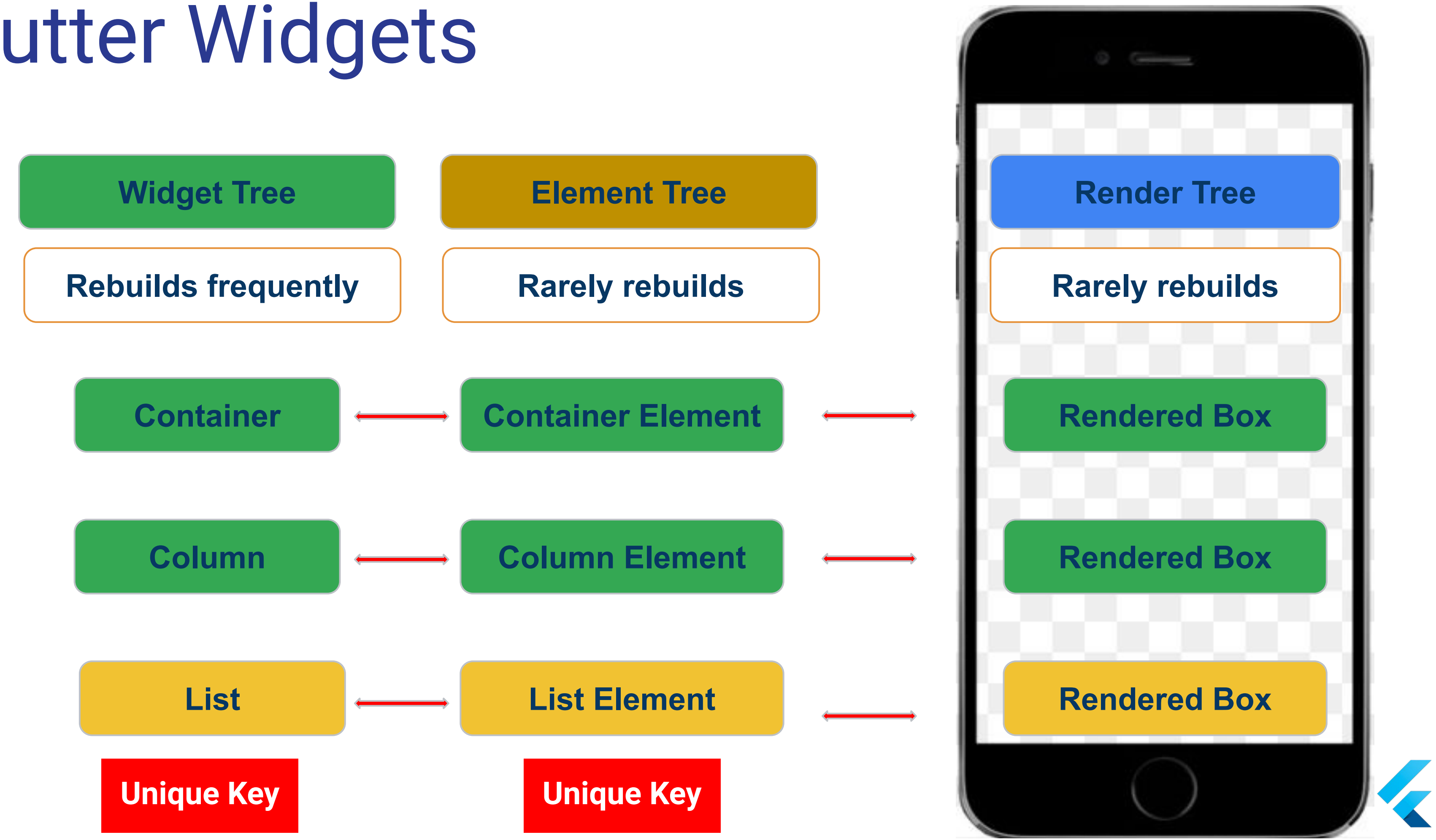
List

List Element

Rendered Box

Unique Key ≠ Unique Key

Flutter Widgets



Widget Tree

Rebuilds frequently

Container

Column

List

Unique Key

Element Tree

Rarely rebuilds

Container Element

Column Element

List Element

Unique Key

Render Tree

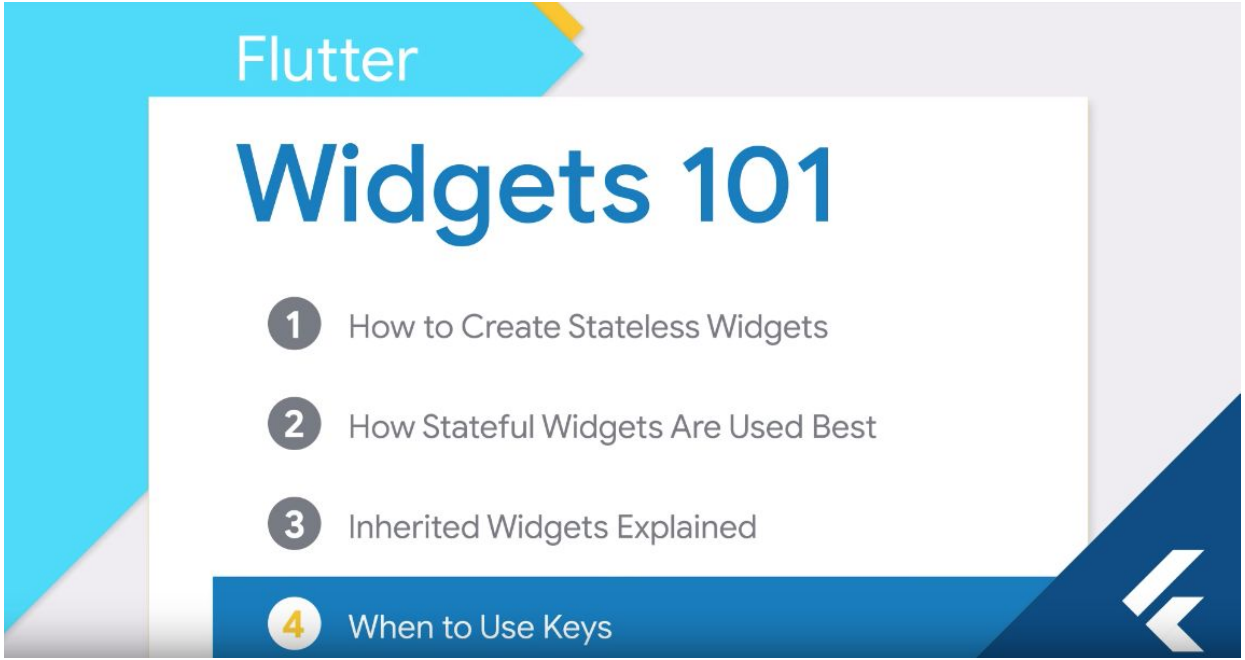
Rarely rebuilds

Rendered Box

Rendered Box

Rendered Box

Flutter Widgets



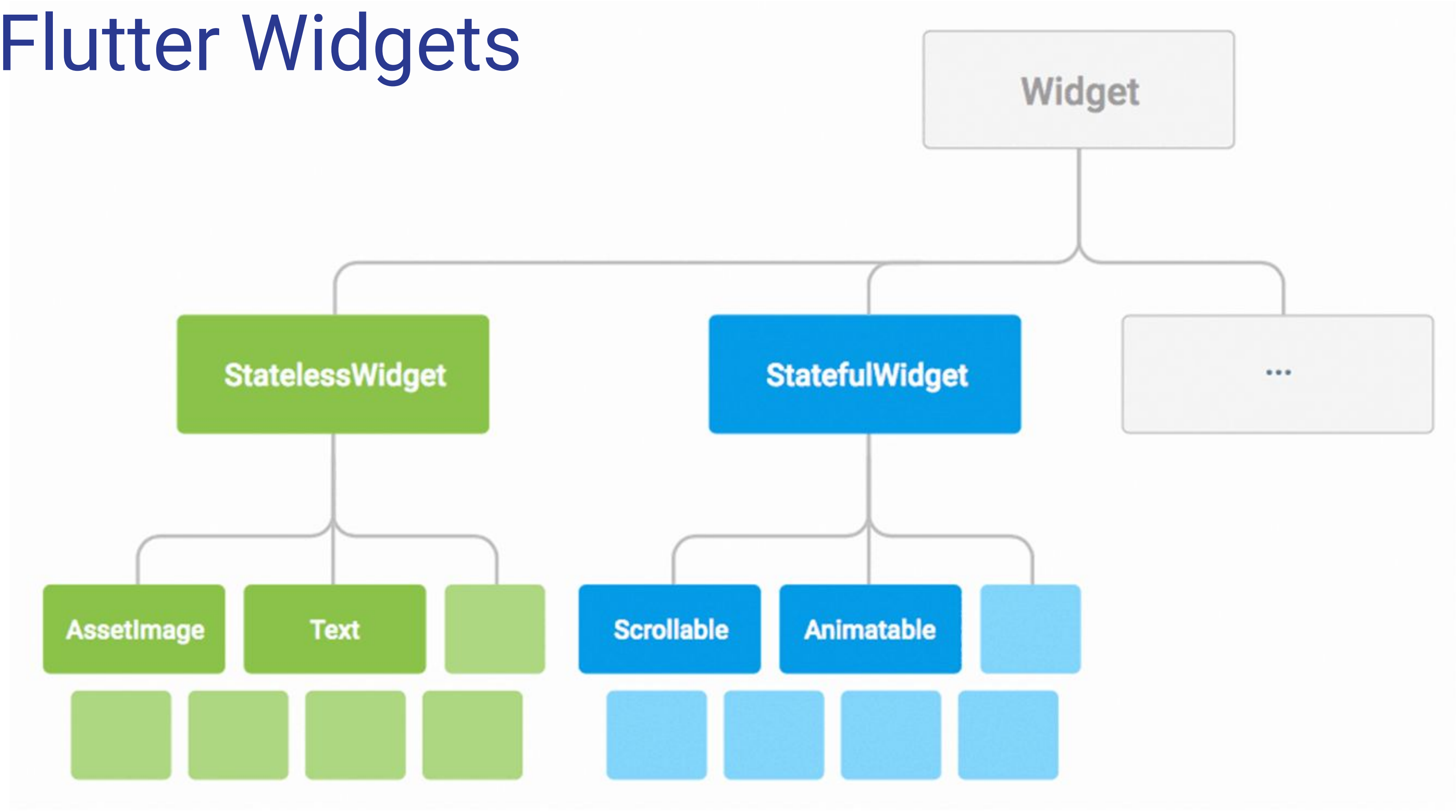
Flutter

Widgets 101

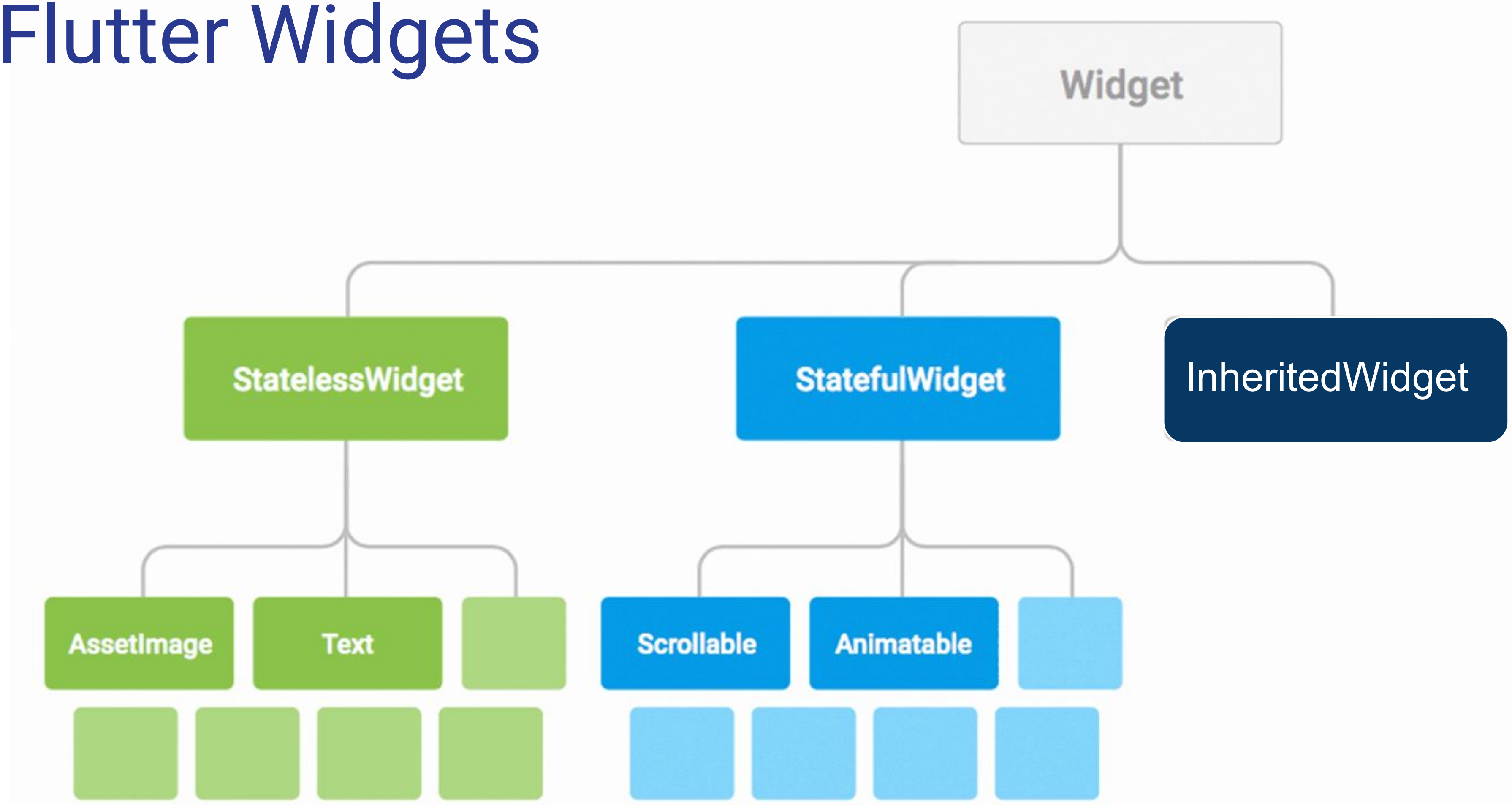
1. How to Create Stateless Widgets
2. How Stateful Widgets Are Used Best
3. Inherited Widgets Explained
4. When to Use Keys

https://www.youtube.com/watch?v=kn0EOS-ZiIc

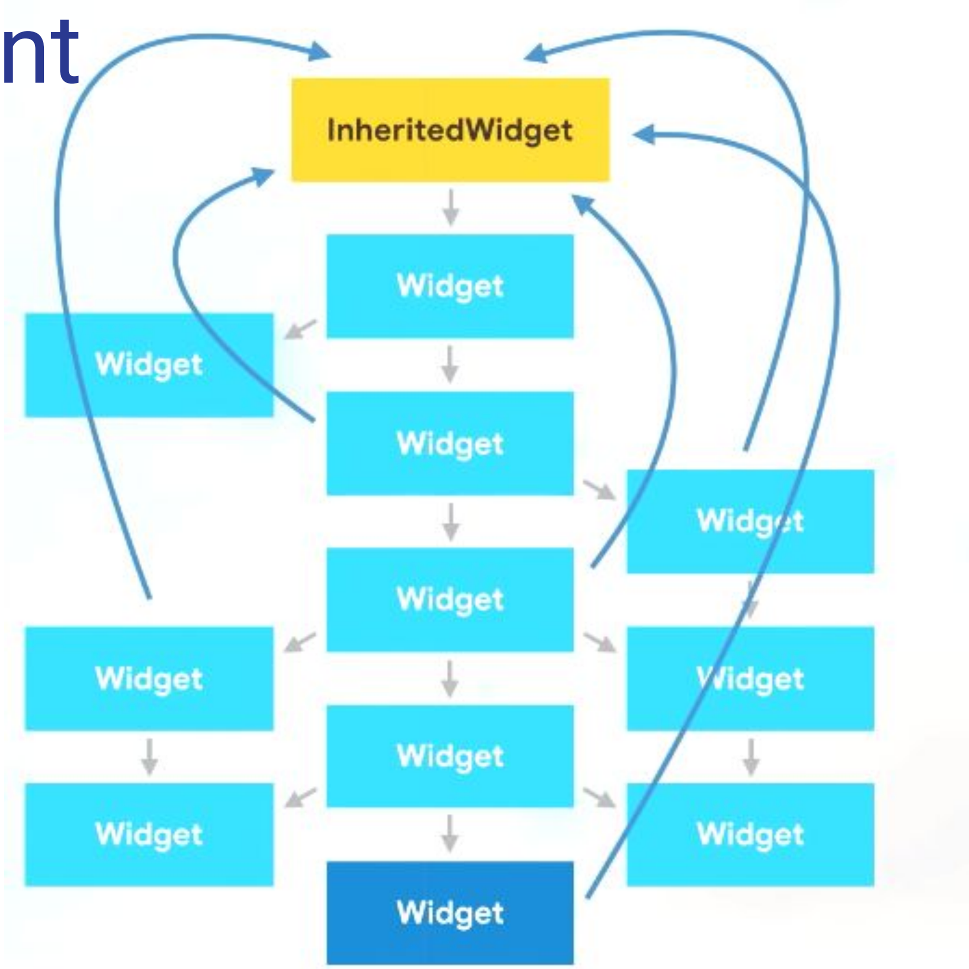
Flutter Widgets



Flutter Widgets



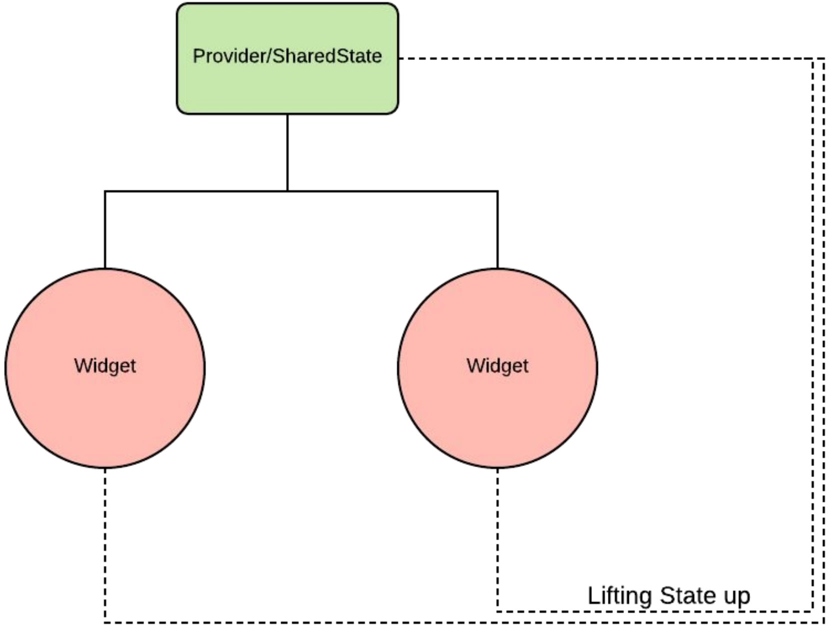
State Management



State Management

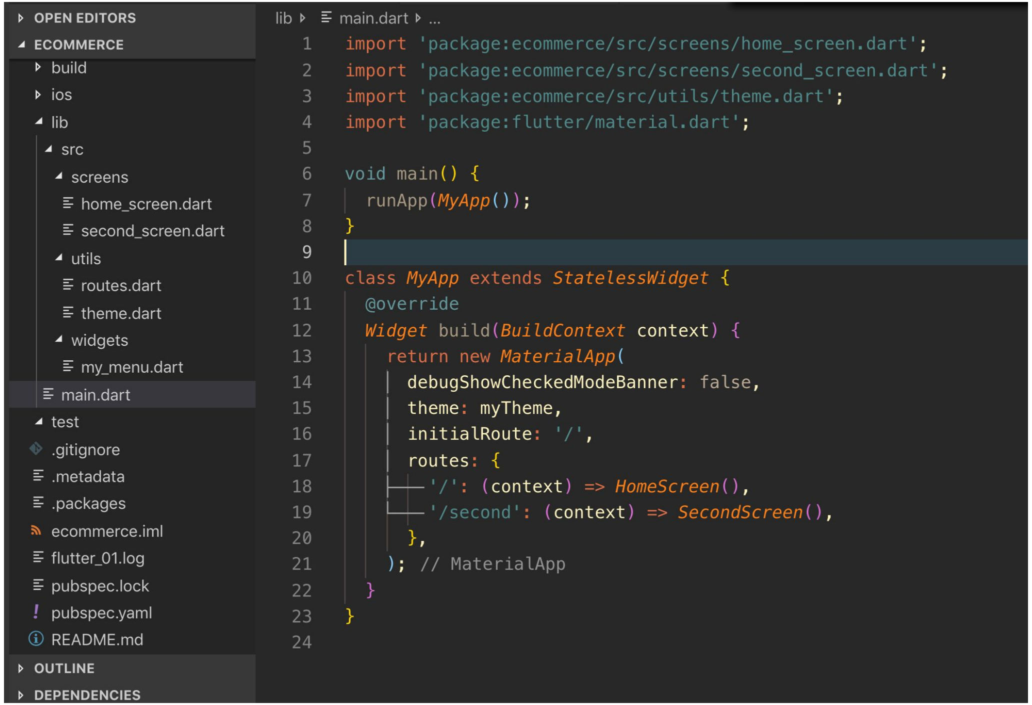
Flutter
Provider

https://pub.dev/packages/provider



BEST PRACTICE

Directory structure



```
OPEN EDITORS
ECOMMERCE
  build
  ios
  lib
    src
      screens
        home_screen.dart
        second_screen.dart
      utils
        routes.dart
        theme.dart
      widgets
        my_menu.dart
    main.dart
  test
  .gitignore
  .metadata
  .packages
  ecommerce.iml
  flutter_01.log
  pubspec.lock
  pubspec.yaml
  README.md
OUTLINE
DEPENDENCIES
```

lib › main.dart › ...

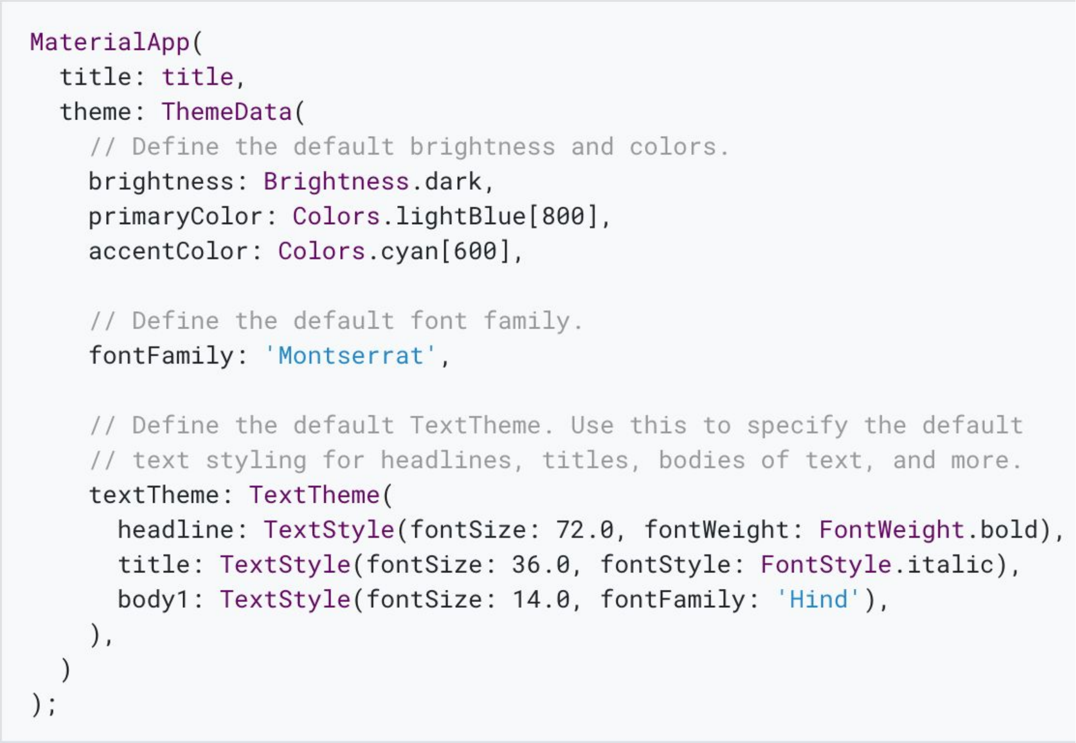
```dart
import 'package:ecommerce/src/screens/home_screen.dart';
import 'package:ecommerce/src/screens/second_screen.dart';
import 'package:ecommerce/src/utils/theme.dart';
import 'package:flutter/material.dart';

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return new MaterialApp(
      debugShowCheckedModeBanner: false,
      theme: myTheme,
      initialRoute: '/',
      routes: {
        '/': (context) => HomeScreen(),
        '/second': (context) => SecondScreen(),
      },
    ); // MaterialApp
  }
}

```

Directory structure

lets.code(it

We Will build simple Application with three screens

App Theme

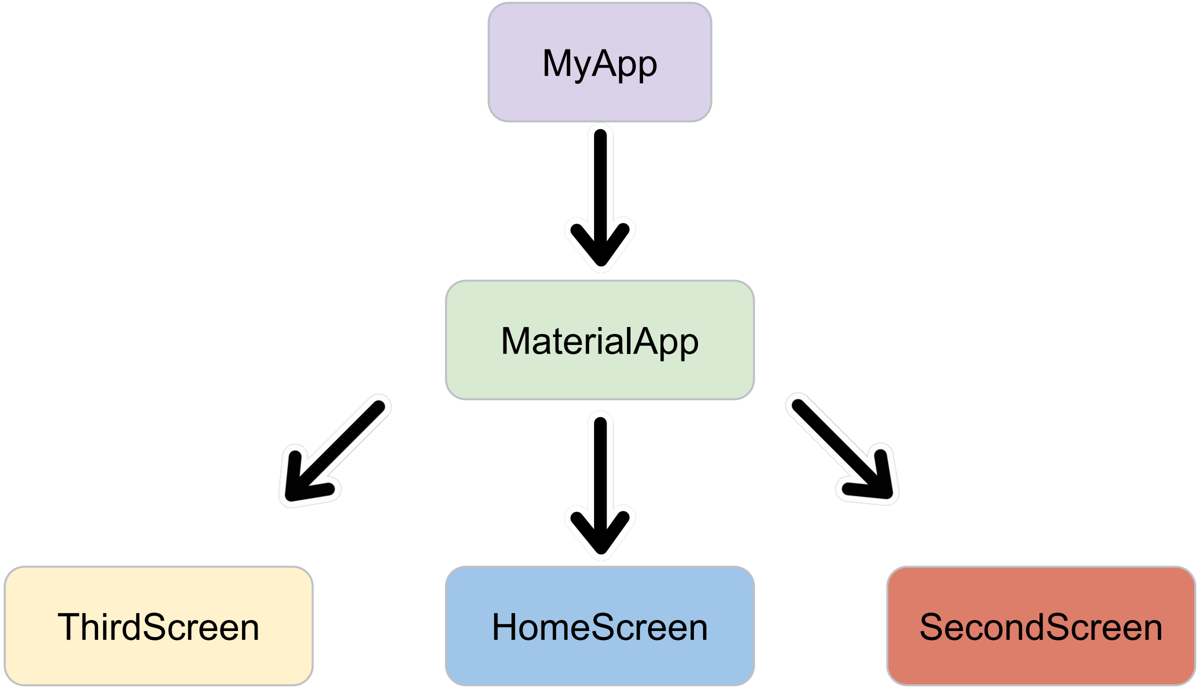


```
MaterialApp(
  title: title,
  theme: ThemeData(
    // Define the default brightness and colors.
    brightness: Brightness.dark,
    primaryColor: Colors.lightBlue[800],
    accentColor: Colors.cyan[600],

    // Define the default font family.
    fontFamily: 'Montserrat',

    // Define the default TextTheme. Use this to specify the default
    // text styling for headlines, titles, bodies of text, and more.
    textTheme: TextTheme(
      headline: TextStyle(fontSize: 72.0, fontWeight: FontWeight.bold),
      title: TextStyle(fontSize: 36.0, fontStyle: FontStyle.italic),
      body1: TextStyle(fontSize: 14.0, fontFamily: 'Hind'),
    ),
  )
);
```

App Theme

lets.code(it

Routing



MyApp

MaterialApp

ThirdScreen

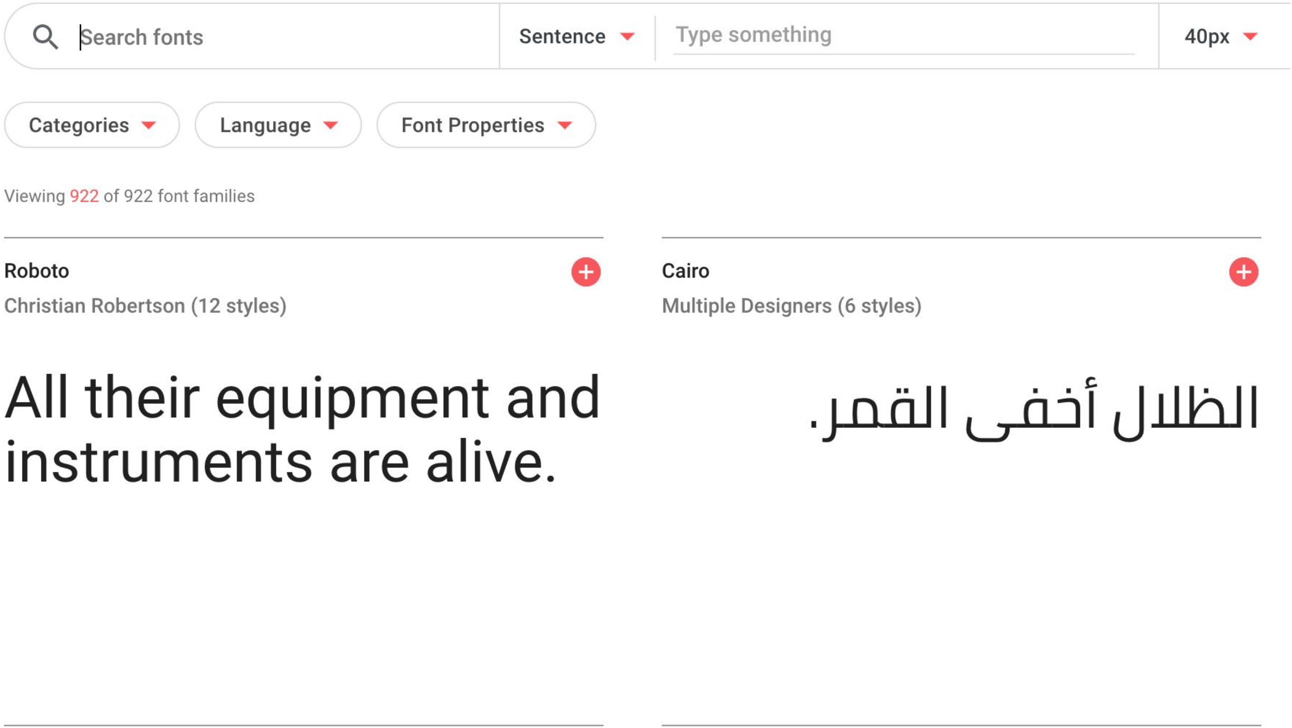
HomeScreen

SecondScreen

Named Routes

lets.code(it

Custom Fonts



Viewing 922 of 922 font families

Roboto

Christian Robertson (12 styles)

All their equipment and instruments are alive.

Cairo

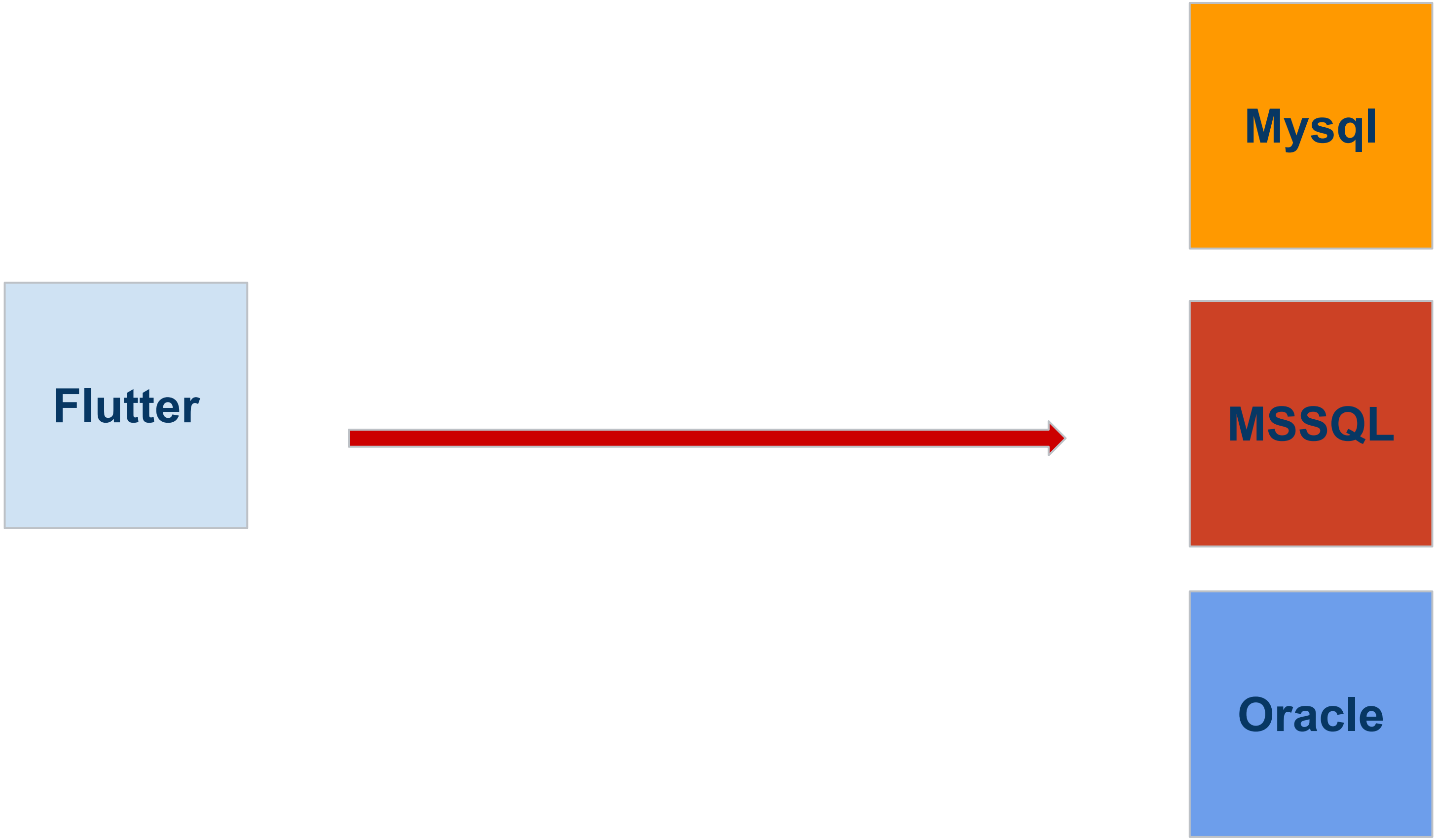
Multiple Designers (6 styles)

الظلال أخفى القمر.

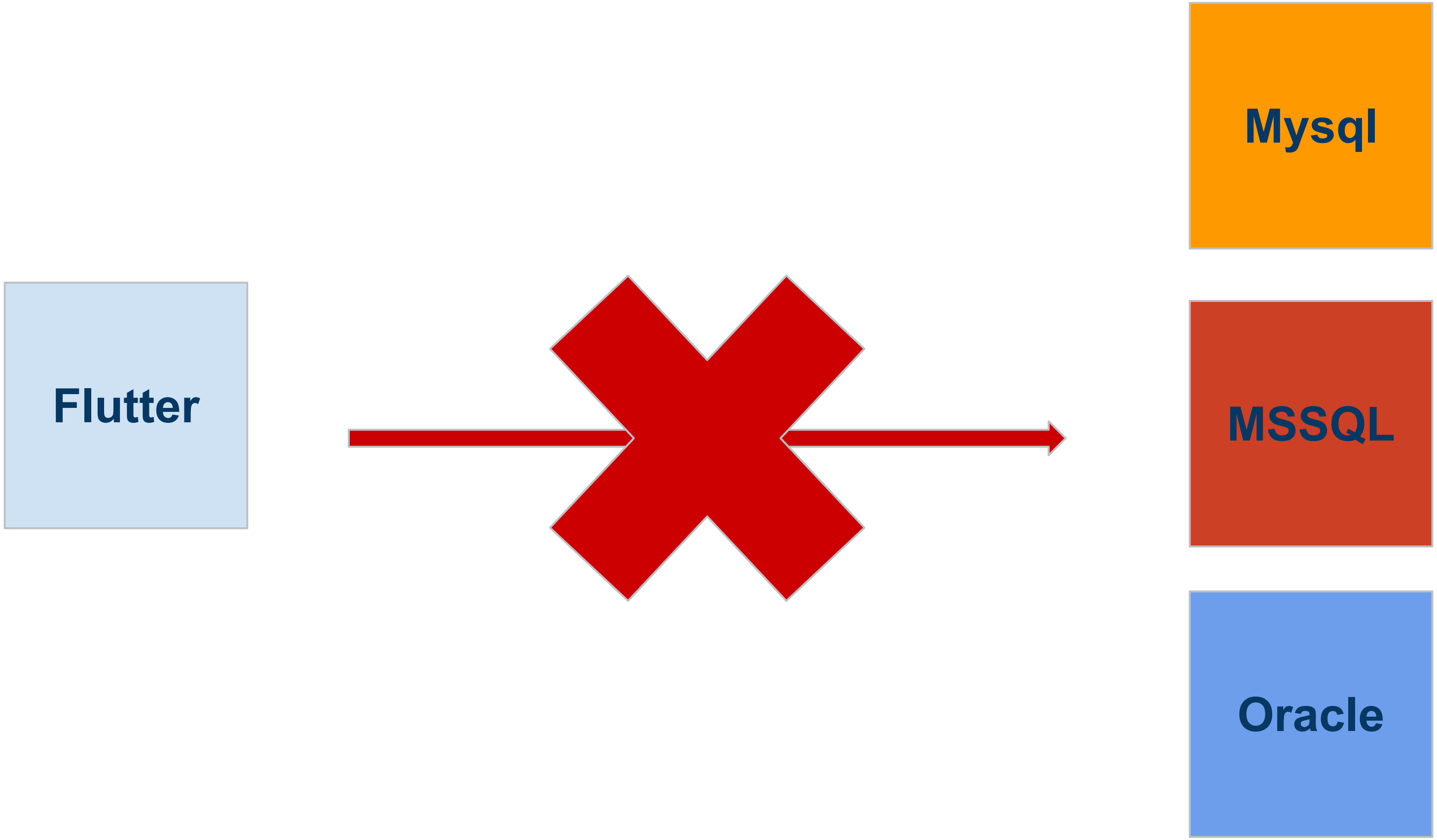
Custom Fonts

lets.code(it

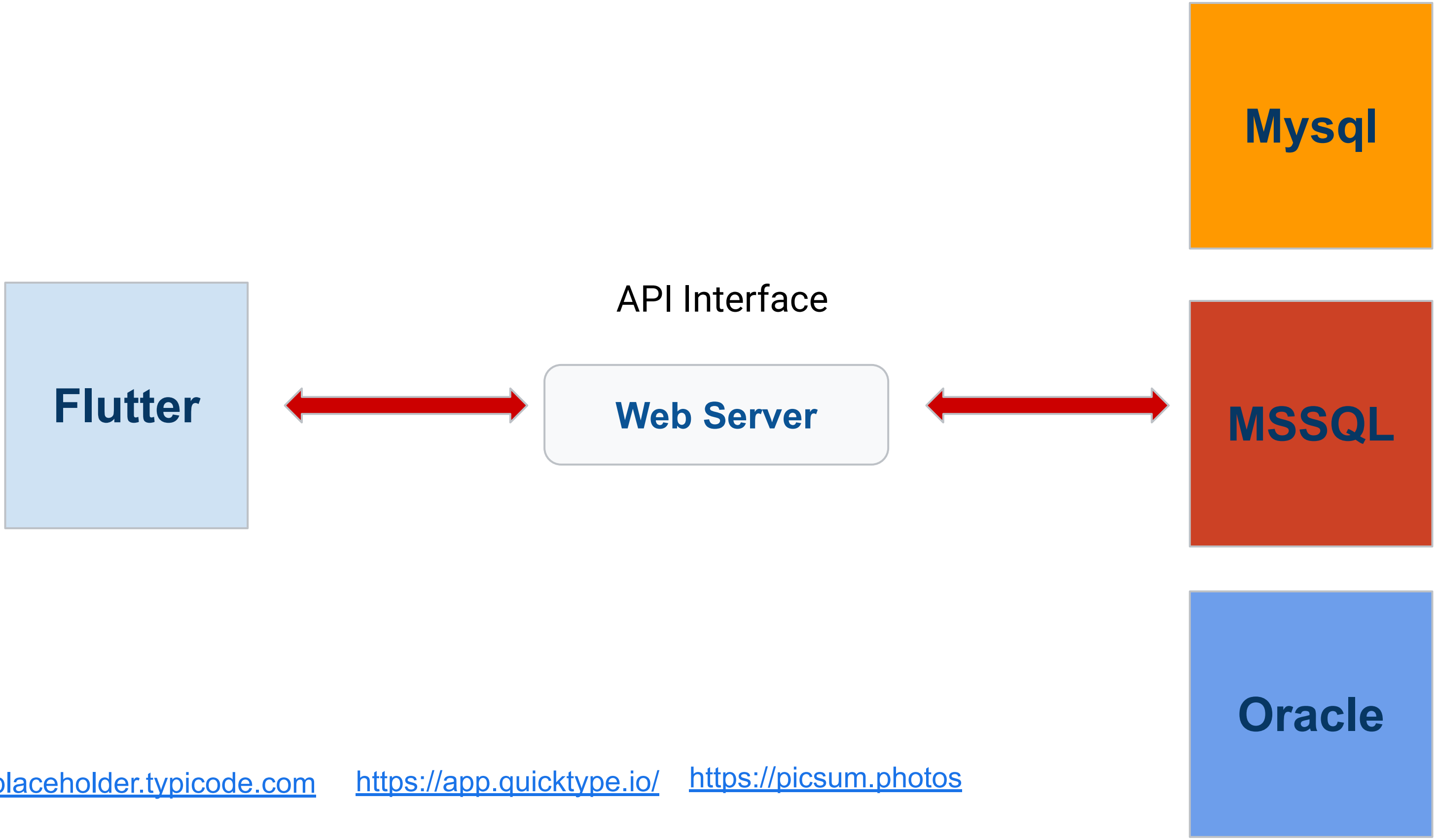
Connect to Database



Connect to Database

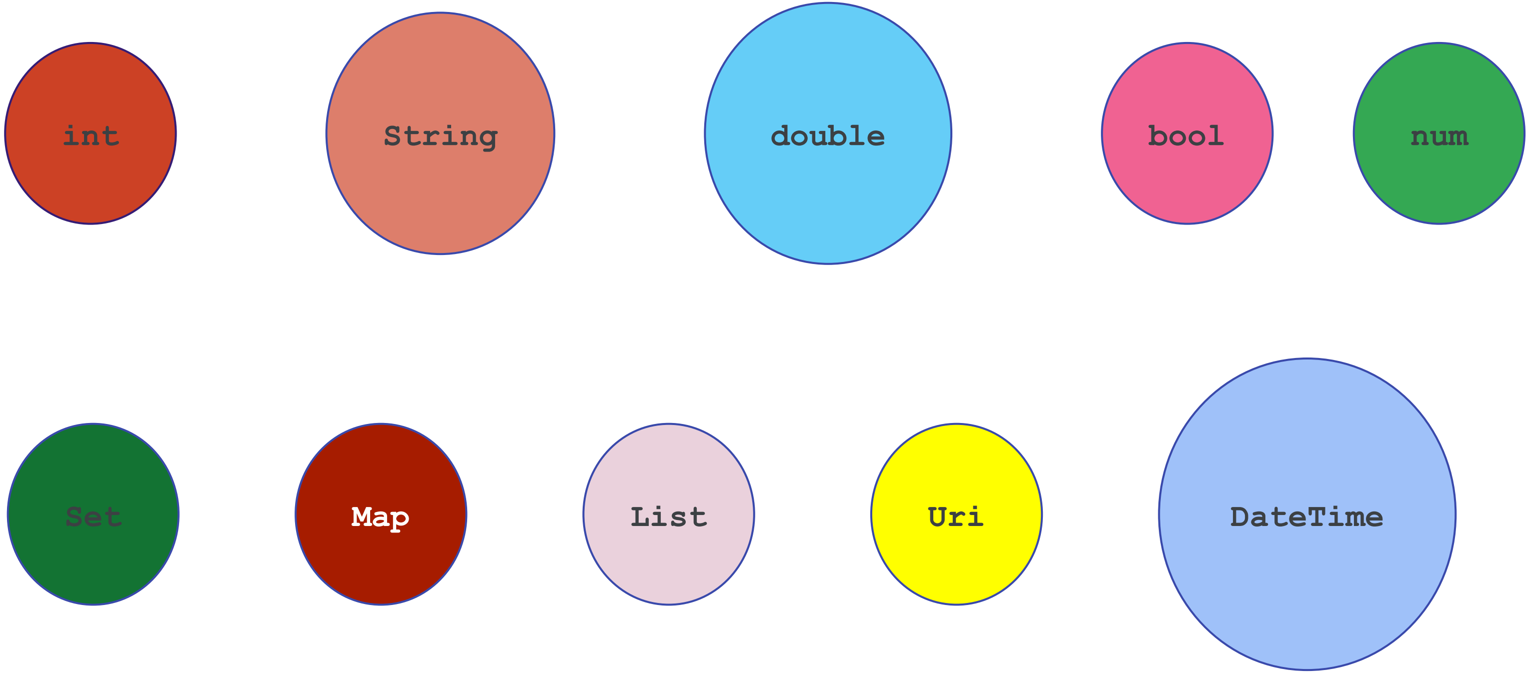


Connect to Database



http://jsonplaceholder.typicode.com https://app.quicktype.io/ https://picsum.photos

Dart Data Types



int

String

double

bool

num

Set

Map

List

Uri

DateTime

https://api.dart.dev/stable/2.4.0/dart-core/dart-core-library.html

Fetching Data from Web

lets.code(it

Async Programming

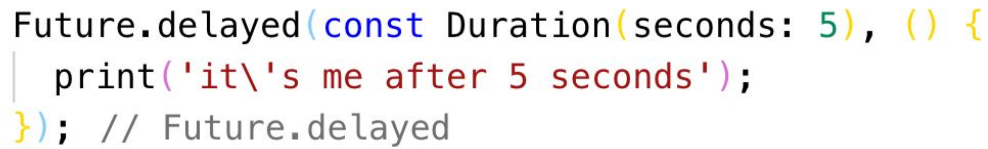
Flutter

Async in Dart

lets.code(it

FutureBuilder

Async = await



```
Future.delayed(const Duration(seconds: 5), () {
  print('it\'s me after 5 seconds');
}); // Future.delayed
```

Thanks

Flutter Study

We will be back in 10 minutes

Ahmed Abu Eldahab

Senior Technical Consultant
Google Developer Expert in Flutter

/Dahabdev