

ff. next



Vogel Csongor

Lead Android and Flutter Developer



 [gerfalcon](#)

 [GerfalconVogel](#)

Say Hi to

**Flutter 3.0**

# Supported **STABLE** platforms



iOS



Android



Web



Windows



macOS

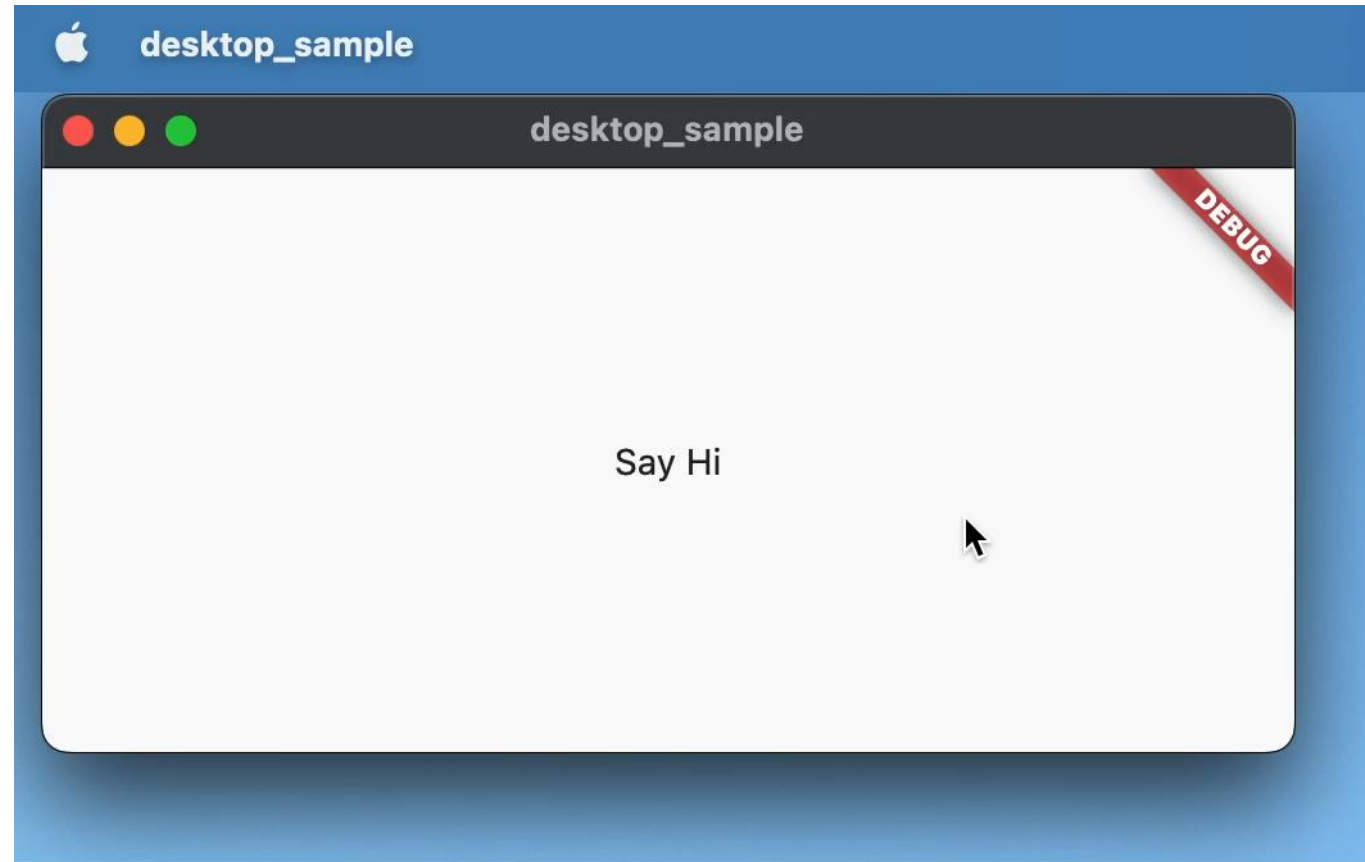


Linux

# Desktop support

macOS – menu bars

mac  
OS



# Desktop support

Linux



CANONICAL™

 pub.dev

GSettings  
database

Power  
management

D-Bus  
message bus

Network  
connections

Firmware  
updates

Geolocation

Desktop  
notifications

PackageKit

Bluetooth

UDisk

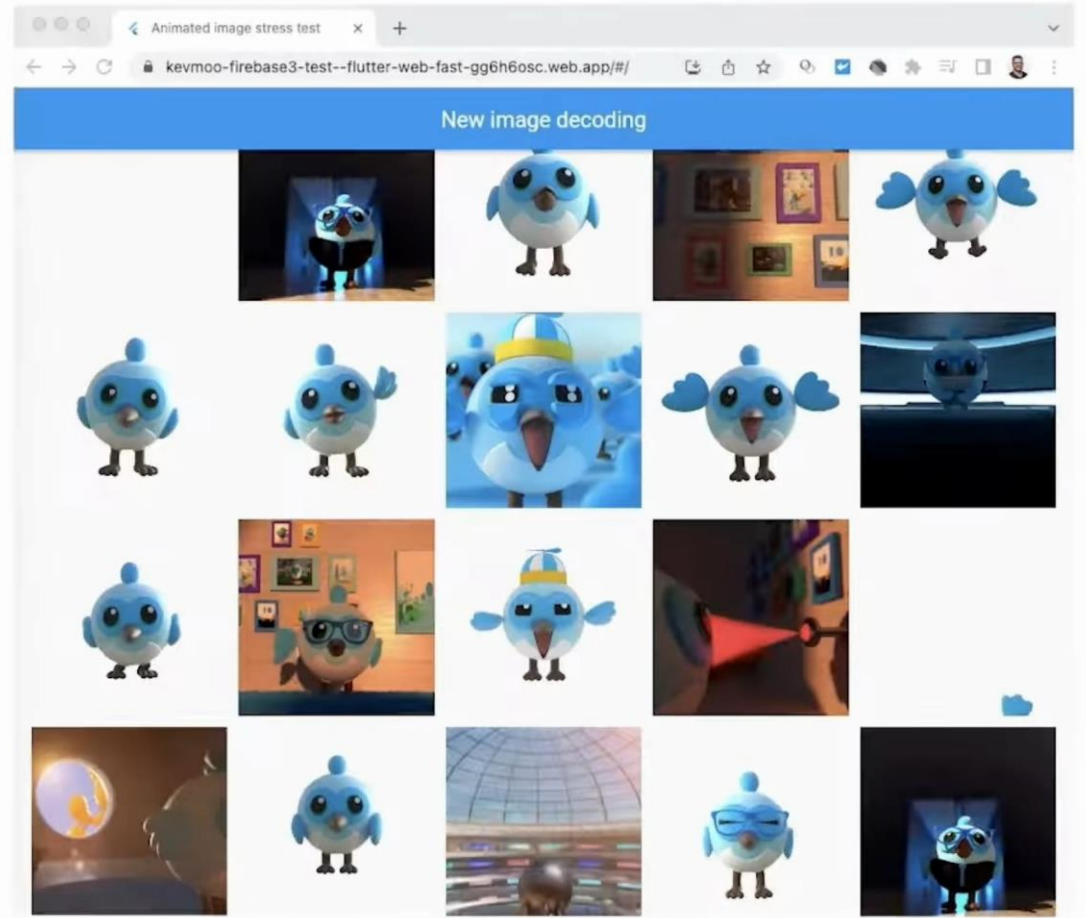
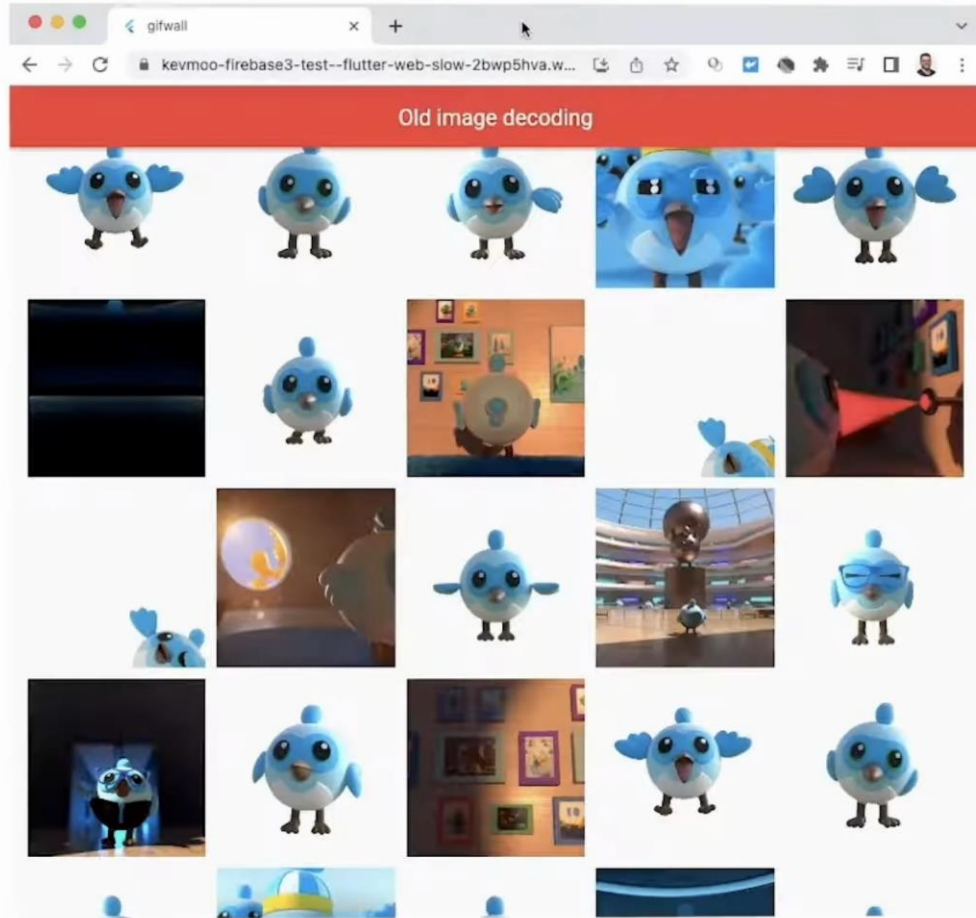
ff. next

# Flutter Web

Performance improvements

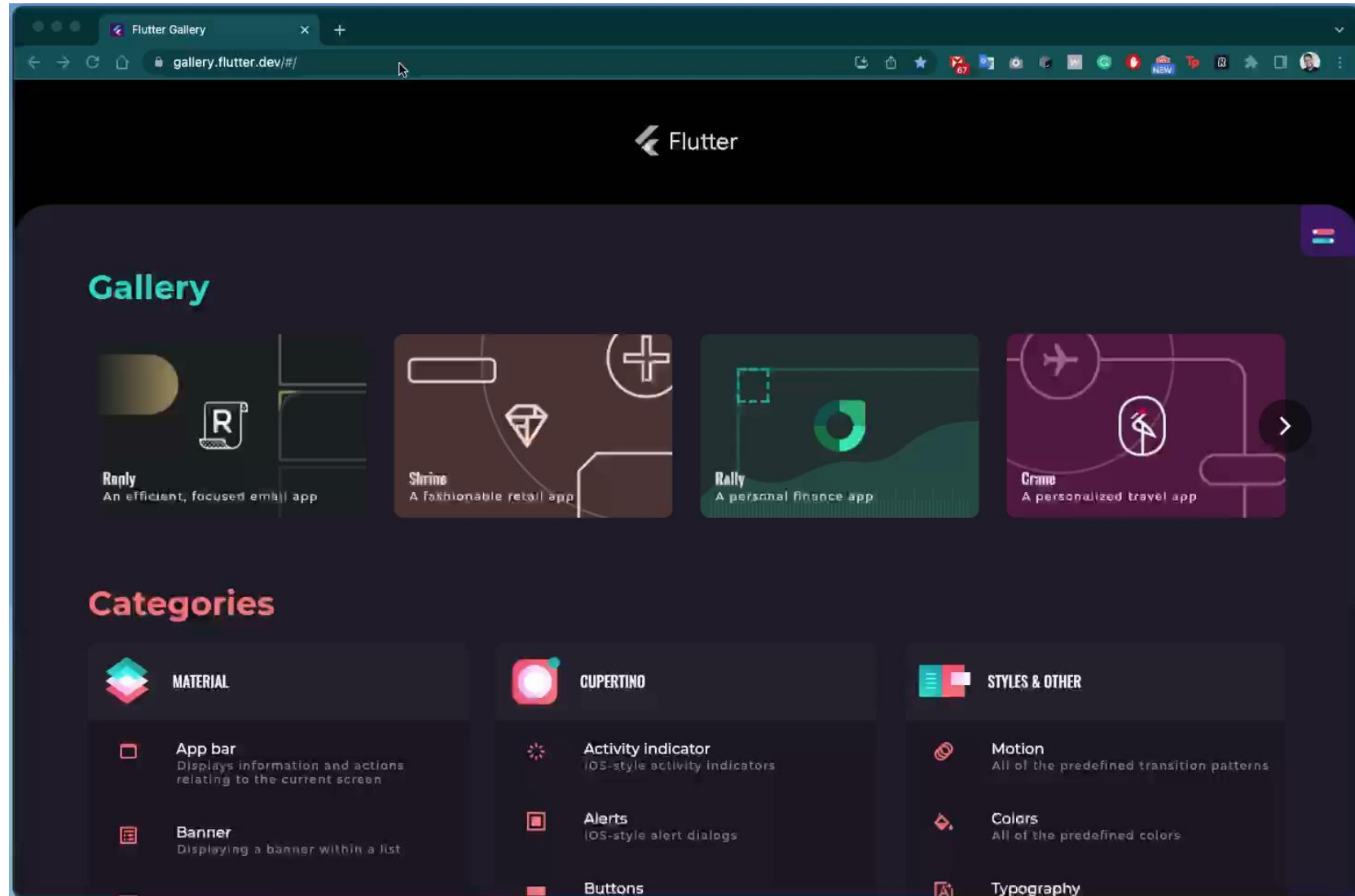


# Image decoding



# App lifecycle API

*gallery.flutter.dev*



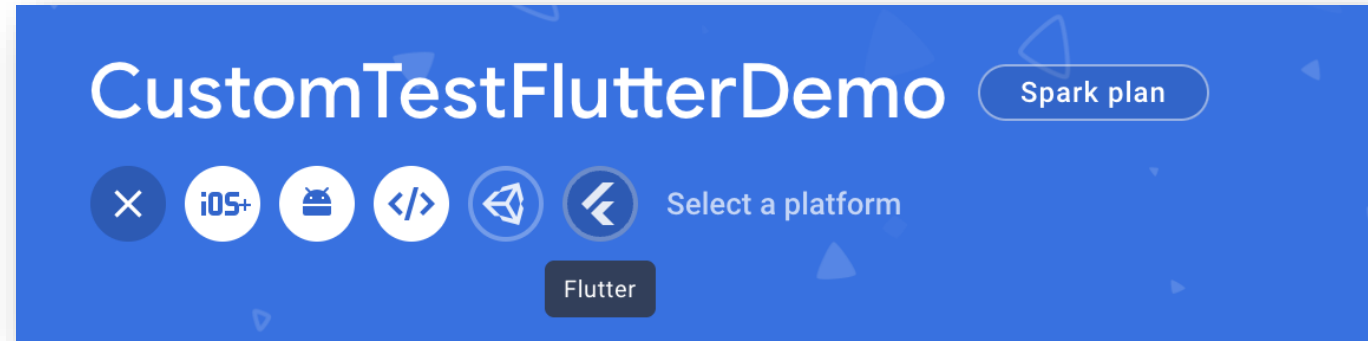
ff. next

# Firebase

improvements




The **integration** never was easier





# Add **Firestore** to your **Flutter** app

 Prepare your workspace

**2** Install and run the FlutterFire CLI

From any directory, run this command:

```
$ dart pub global activate flutterfire_cli
```



Then, at the root of your Flutter project directory, run this command:

```
$ flutterfire configure --project=customtestflutterdemo
```



This automatically registers your per-platform apps with Firebase and adds a `lib/firebase_options.dart` configuration file to your Flutter project.

[Previous](#)

[Next](#)

**3** Initialize Firebase and add plugins



# Add **Firestore** to your **Flutter** app

- 1 Prepare your workspace
- 2 Install and run the FlutterFire CLI
- 3 Initialize Firestore and add plugins

To initialize Firestore, call `Firestore.initializeApp` from the `firebase_core` package with the configuration from your new `firebase_options.dart` file:

```
import 'package:firebase_core/firebase_core.dart';
import 'firebase_options.dart';

// ...

await Firestore.initializeApp(
  options: DefaultFirebaseOptions.currentPlatform,
);
```



Then, add and begin using the [Flutter plugins](#) for the Firestore products you'd like to use.

Note: If you're using Analytics or Performance Monitoring, you may need to follow a few additional setup steps.

[Previous](#)

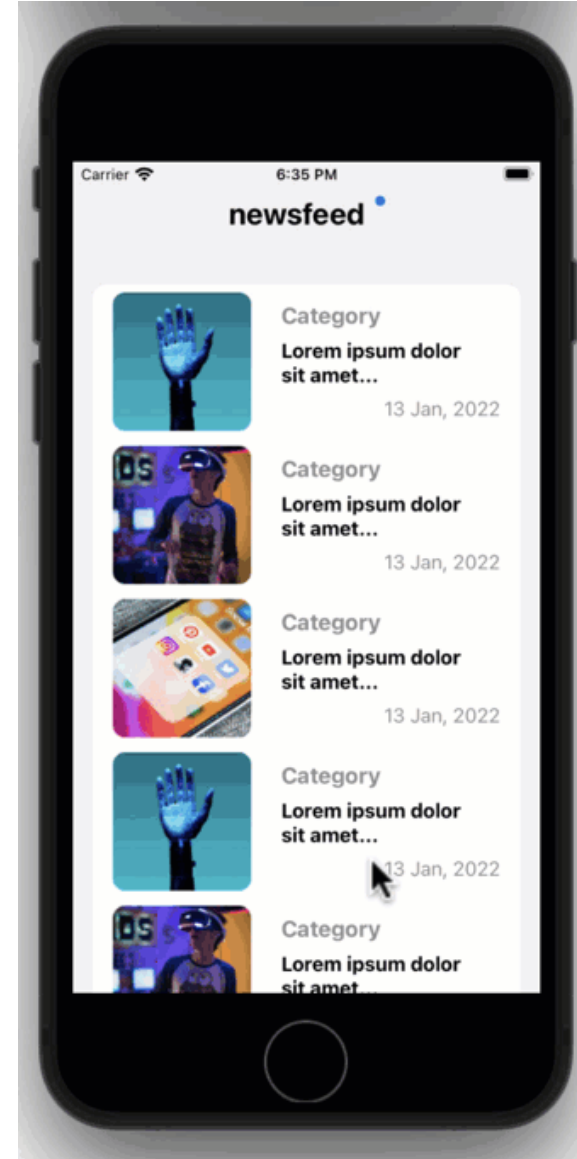
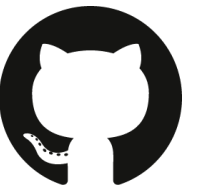
[Continue to console](#)



# App to app

samples

## Put Flutter to Work 🏠

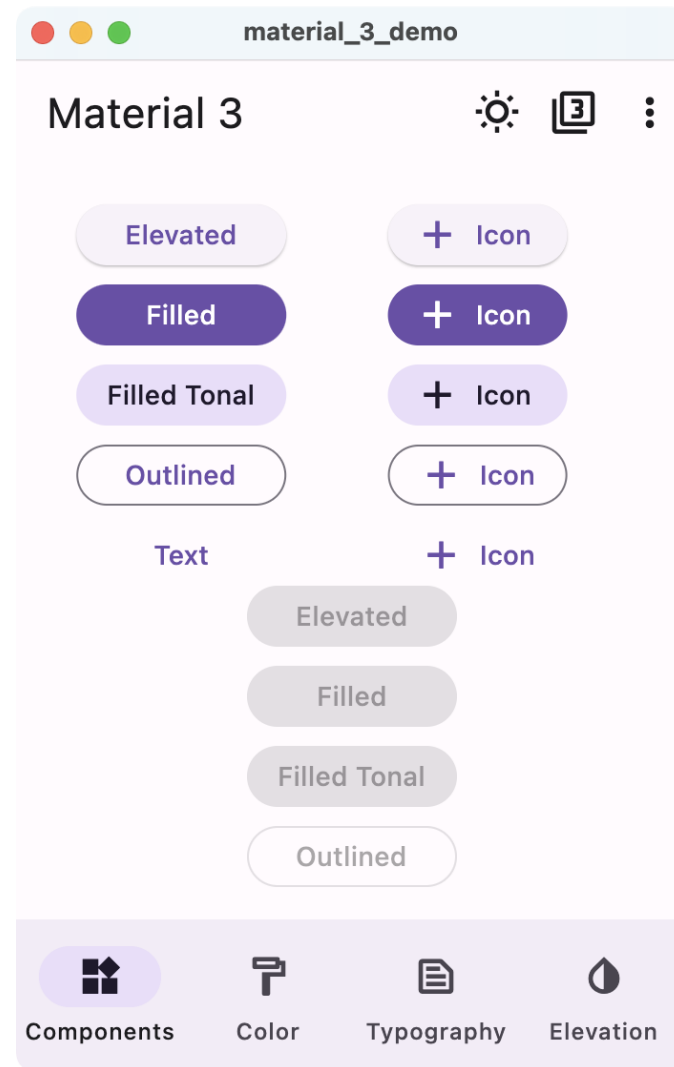


ff. next

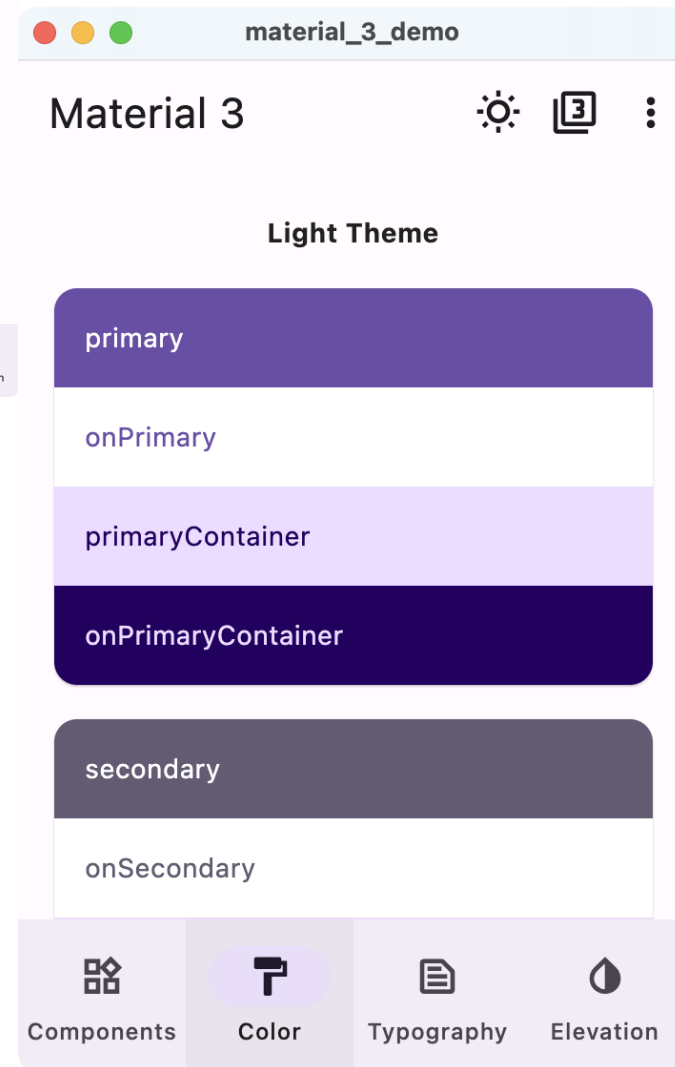
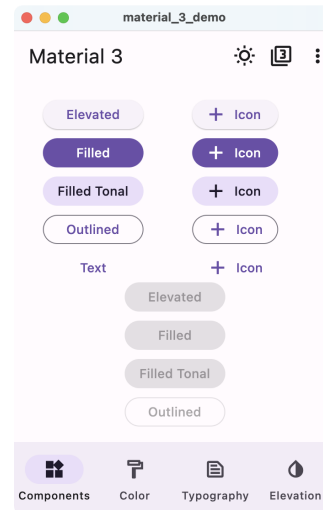


# Material 3 stable

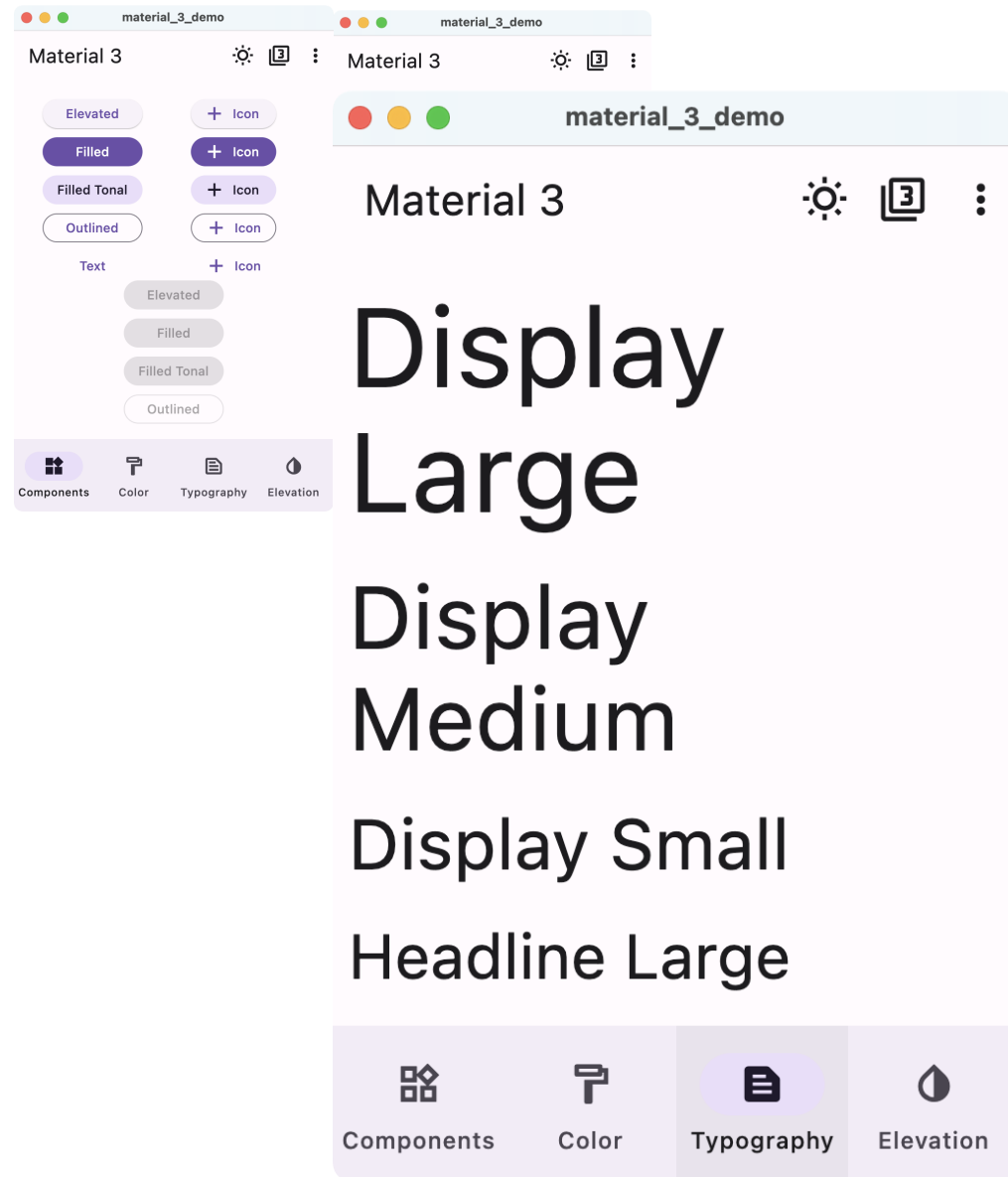
# Material 3 stable



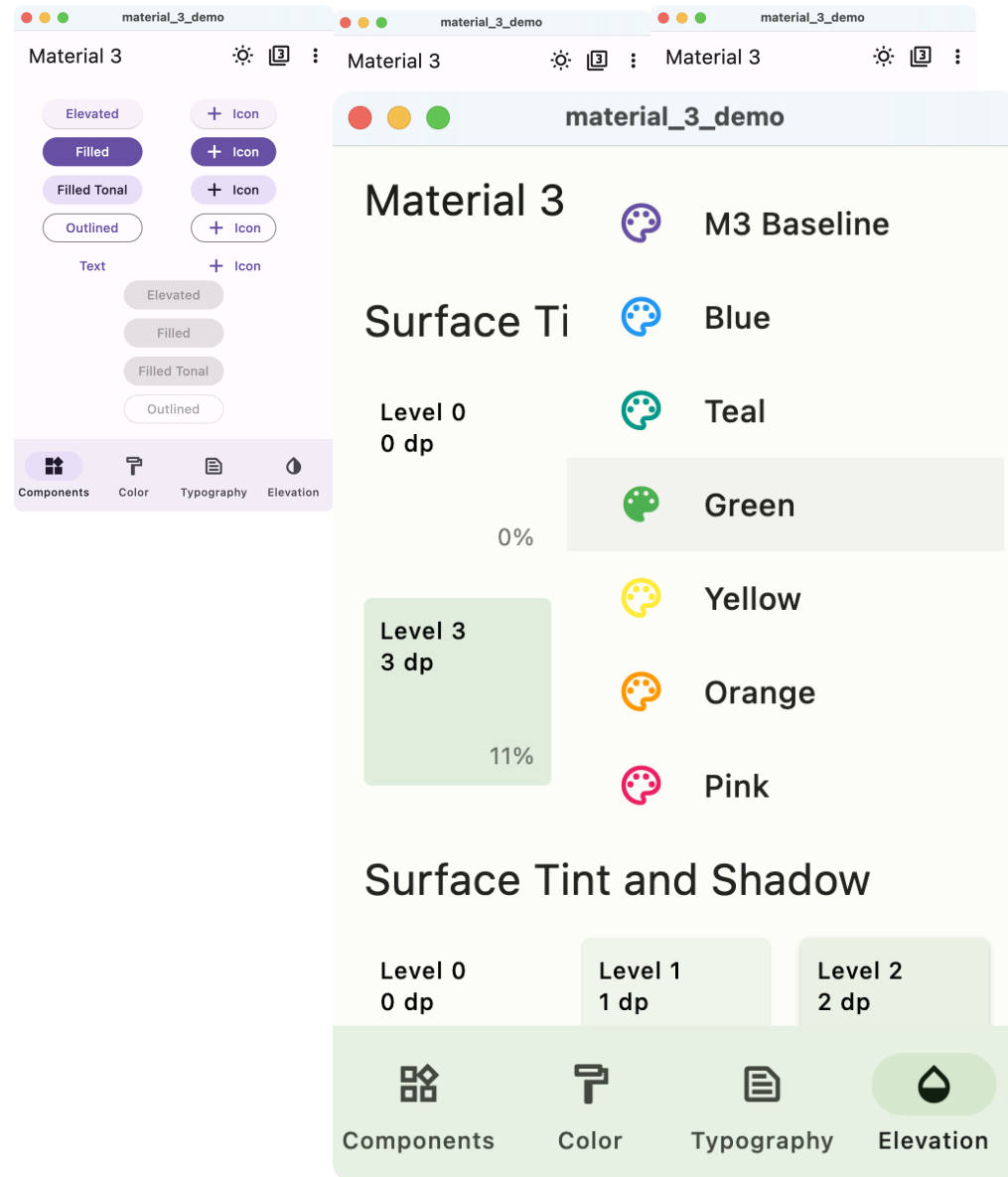
# Material 3 stable



# Material 3 stable



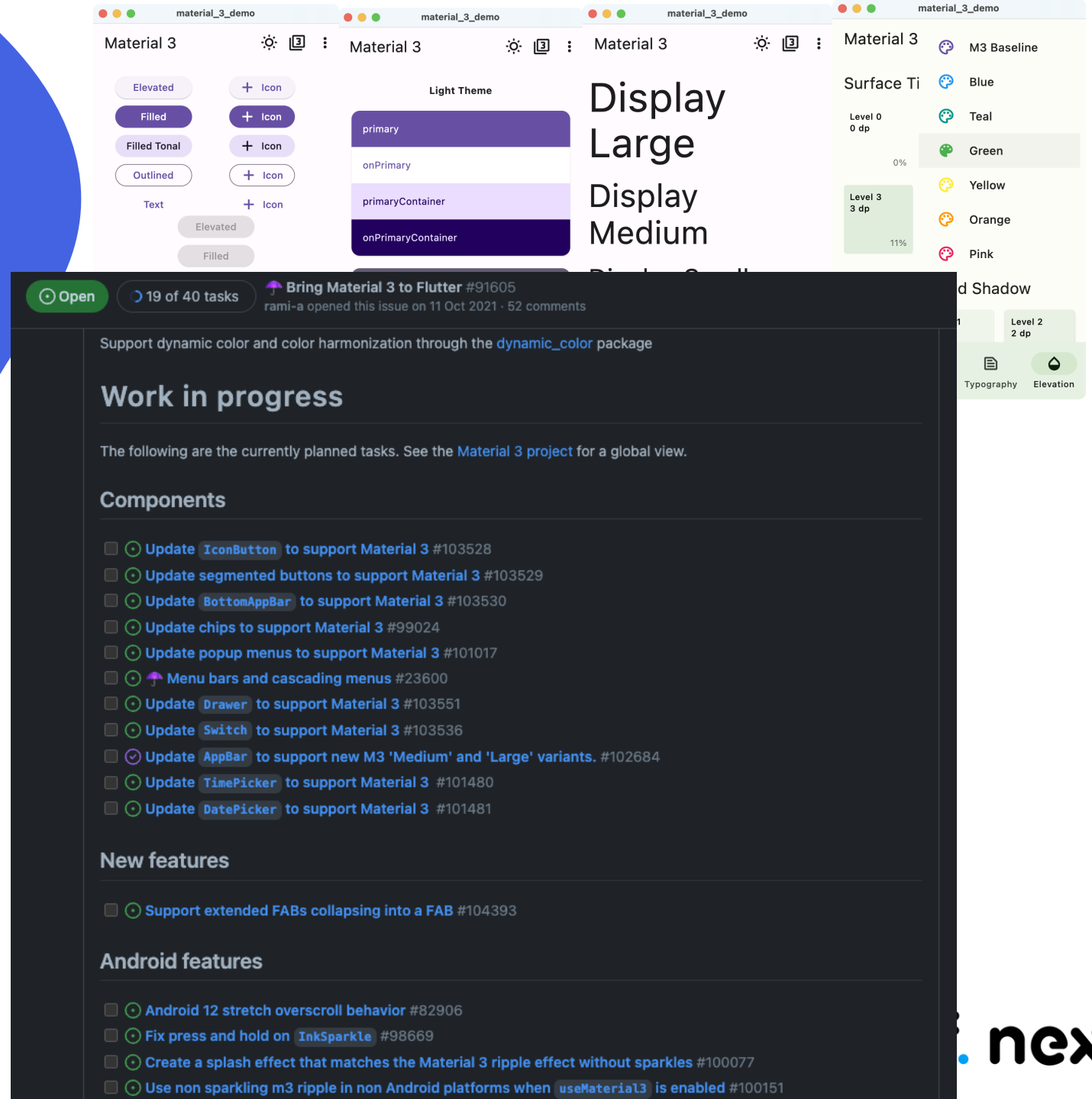
# Material 3 stable





# Material 3 stable

(WIP)



The image displays the Material 3 design system components and a GitHub project page. The top section shows four browser windows of the Material 3 design tool, each displaying a different component: 'Elevated', 'Filled', 'Filled Tonal', and 'Outlined' buttons; 'Text' and 'Icon' components; 'Light Theme' color palette; and 'Surface Tint' color palette. The bottom section shows a GitHub project page for 'Bring Material 3 to Flutter #91605' by rami-a, opened on 11 Oct 2021 with 52 comments. The page is titled 'Work in progress' and lists the following currently planned tasks: Support dynamic color and color harmonization through the `dynamic_color` package. The page is divided into three sections: Components, New features, and Android features.

## Work in progress

The following are the currently planned tasks. See the [Material 3 project](#) for a global view.

### Components

- ☐ ☒ Update `IconButton` to support Material 3 #103528
- ☐ ☒ Update segmented buttons to support Material 3 #103529
- ☐ ☒ Update `BottomAppBar` to support Material 3 #103530
- ☐ ☒ Update chips to support Material 3 #99024
- ☐ ☒ Update popup menus to support Material 3 #101017
- ☐ ☒ Menu bars and cascading menus #23600
- ☐ ☒ Update `Drawer` to support Material 3 #103551
- ☐ ☒ Update `Switch` to support Material 3 #103536
- ☐ ☒ Update `AppBar` to support new M3 'Medium' and 'Large' variants. #102684
- ☐ ☒ Update `TimePicker` to support Material 3 #101480
- ☐ ☒ Update `DatePicker` to support Material 3 #101481

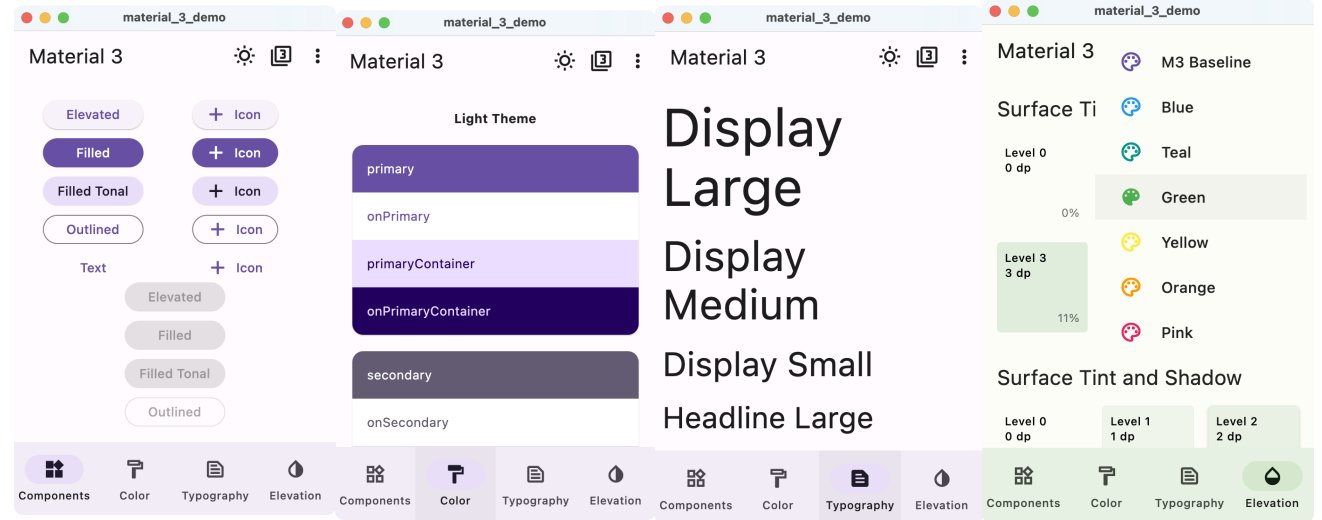
### New features

- ☐ ☒ Support extended FABs collapsing into a FAB #104393

### Android features

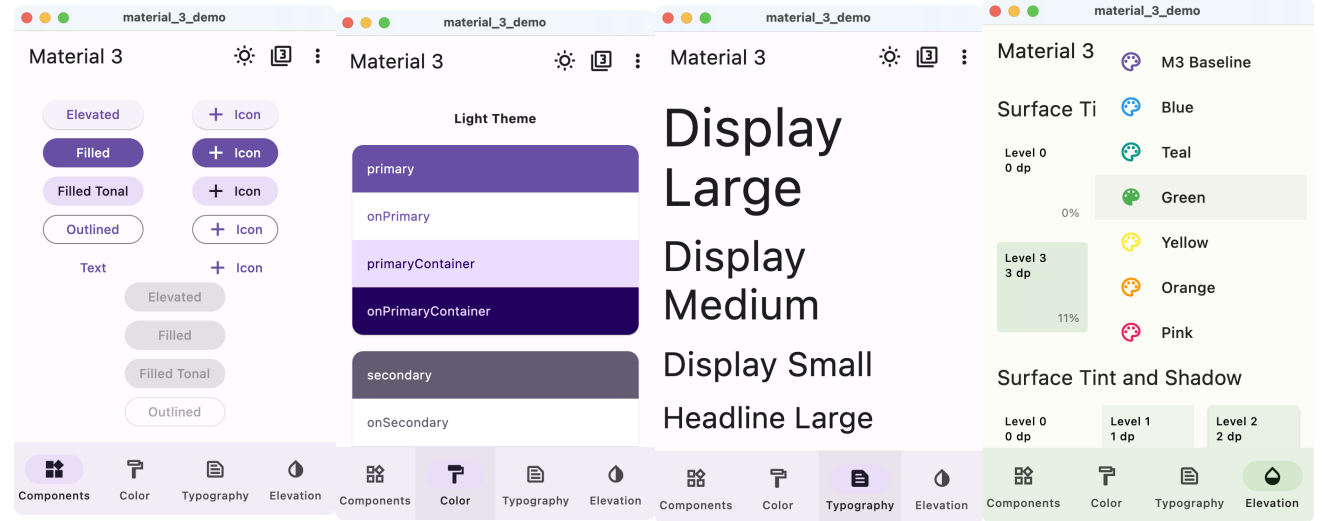
- ☐ ☒ Android 12 stretch overscroll behavior #82906
- ☐ ☒ Fix press and hold on `InkSparkle` #98669
- ☐ ☒ Create a splash effect that matches the Material 3 ripple effect without sparkles #100077
- ☐ ☒ Use non sparkling m3 ripple in non Android platforms when `useMaterial3` is enabled #100151

# Material 3 stable (WIP)



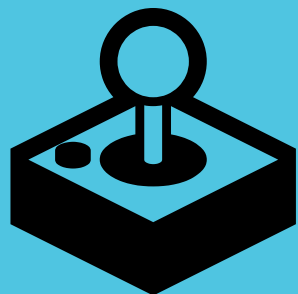
```
MaterialApp(  
  theme: ThemeData(),  
  
  /// (...)  
);
```

# Material 3 stable (WIP)



```
MaterialApp(  
  theme: ThemeData(useMaterial3: true)  
  /// (...)  
);
```

# Game Toolkit



# Dart 2.17

New announcements



OLD

## Named args anywhere

```
Text(  
  'Title',  
  style: TextStyle(fontWeight: FontWeight.bold),  
  
  /// (...)  
)
```



NEW

## Named args anywhere

```
Text(  
  style: TextStyle(fontWeight: FontWeight.bold),  
  'Title',  
  
  /// (...)  
)
```



OLD

enums

```
enum City {  
    madrid,  
    dubai,  
    dunabogdany  
}
```



OLD

```
enum City {  
    madrid,  
    dubai,  
    dunabogdany  
}
```

enums

OLD

## enums

```
enum City {  
    madrid,  
    dubai,  
    dunabogdany  
}
```

```
extension CityExtensions on City {  
    String getCurrency(City city) {  
        switch (city) {  
            case City.madrid:  
                return 'EUR';  
            case City.dubai:  
                return 'AED';  
            case City.dunabogdany:  
                return 'HUF';  
            default:  
                return ':';  
        }  
    }  
}
```

```
String convertToString() =>  
    'The currency code of $name is ${getCurrency(this)}';  
}
```

```
void main() {  
    print(City.madrid.convertToString());  
}
```

NEW

## enums

```
enum City {  
    madrid('EUR'),  
    dubai(currencyOfDubai),  
    dunabogdany('HUF');  
  
    final String currency;  
    const City(this.currency);  
    static const currencyOfDubai = 'AED';  
  
    String get getCurrencySign {  
        switch (this) {  
            case madrid:  
                return '€';  
            case dubai:  
                return 'د.ا';  
            case dunabogdany:  
                return 'Ft';  
            default:  
                return ':';(  
        }  
    }  
}  
  
@override  
String toString() ⇒ name.toUpperCase();  
}
```

```
void main() {  
    print(City.madrid);  
}
```

OLD

## Super constructors

```
class _SemanticsDiagnosticableNode extends  
DiagnosticableNode<SemanticsNode> {  
  
  _SemanticsDiagnosticableNode({  
    String? name,  
    required SemanticsNode value,  
    required DiagnosticsTreeStyle? style,  
    required this.childOrder,  
  }) : super(  
    name: name,  
    value: value,  
    style: style,  
  );  
  
  /// ...  
}
```

NEW

## Super constructors

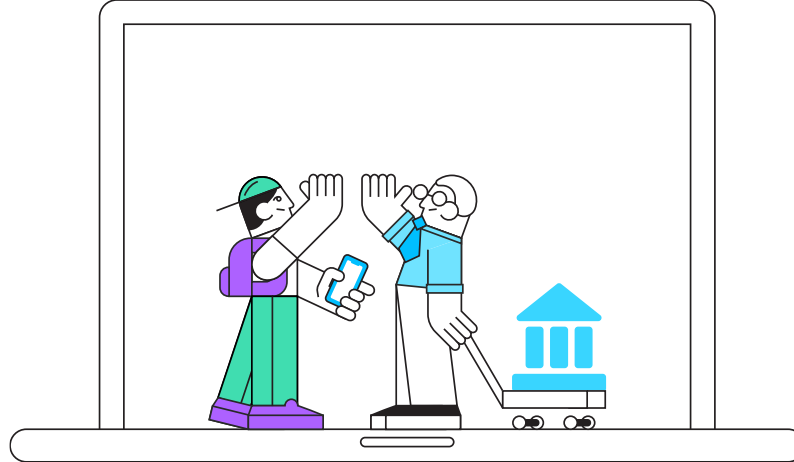
```
class _SemanticsDiagnosticableNode extends  
DiagnosticableNode<SemanticsNode> {  
  
  _SemanticsDiagnosticableNode({  
    super.name,  
    required super.value,  
    required super.style,  
    required this.childOrder,  
  });  
  
  /// ...  
}
```

Minus ~2k lines of code

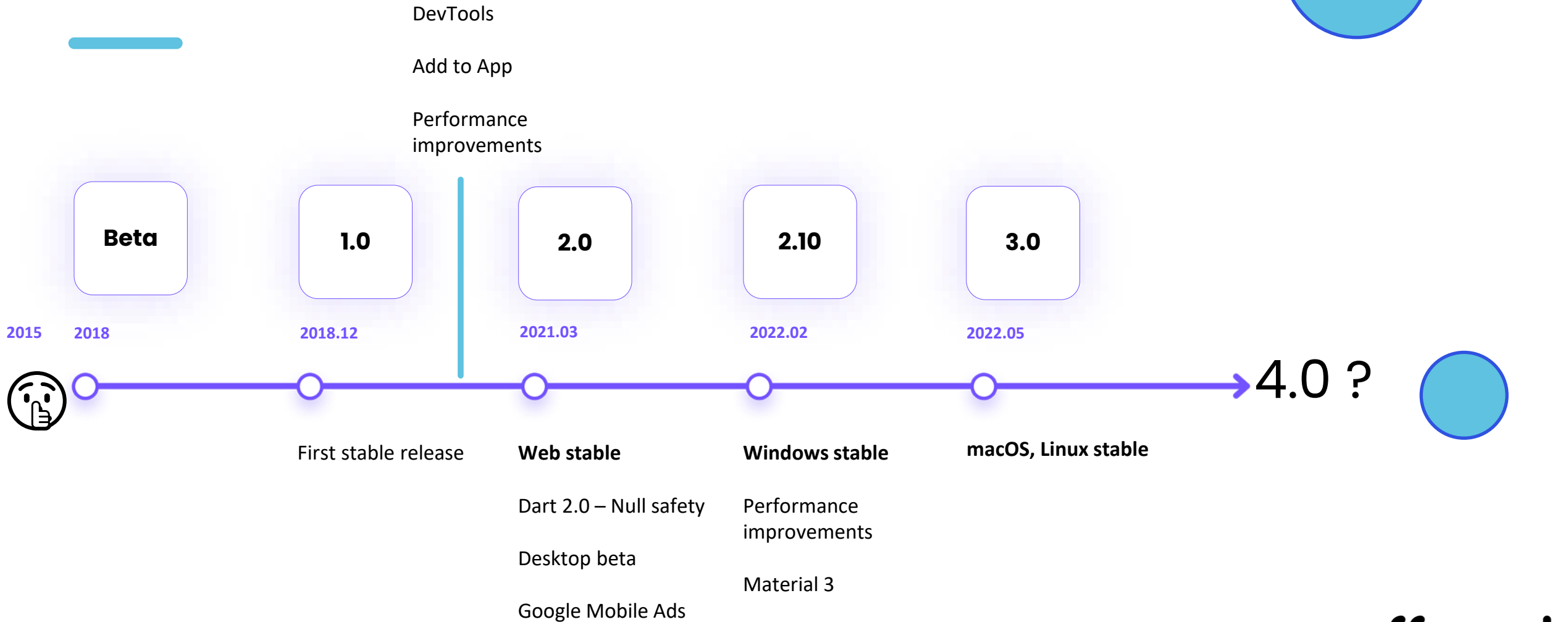
**dart fix --apply**

**ff. next**

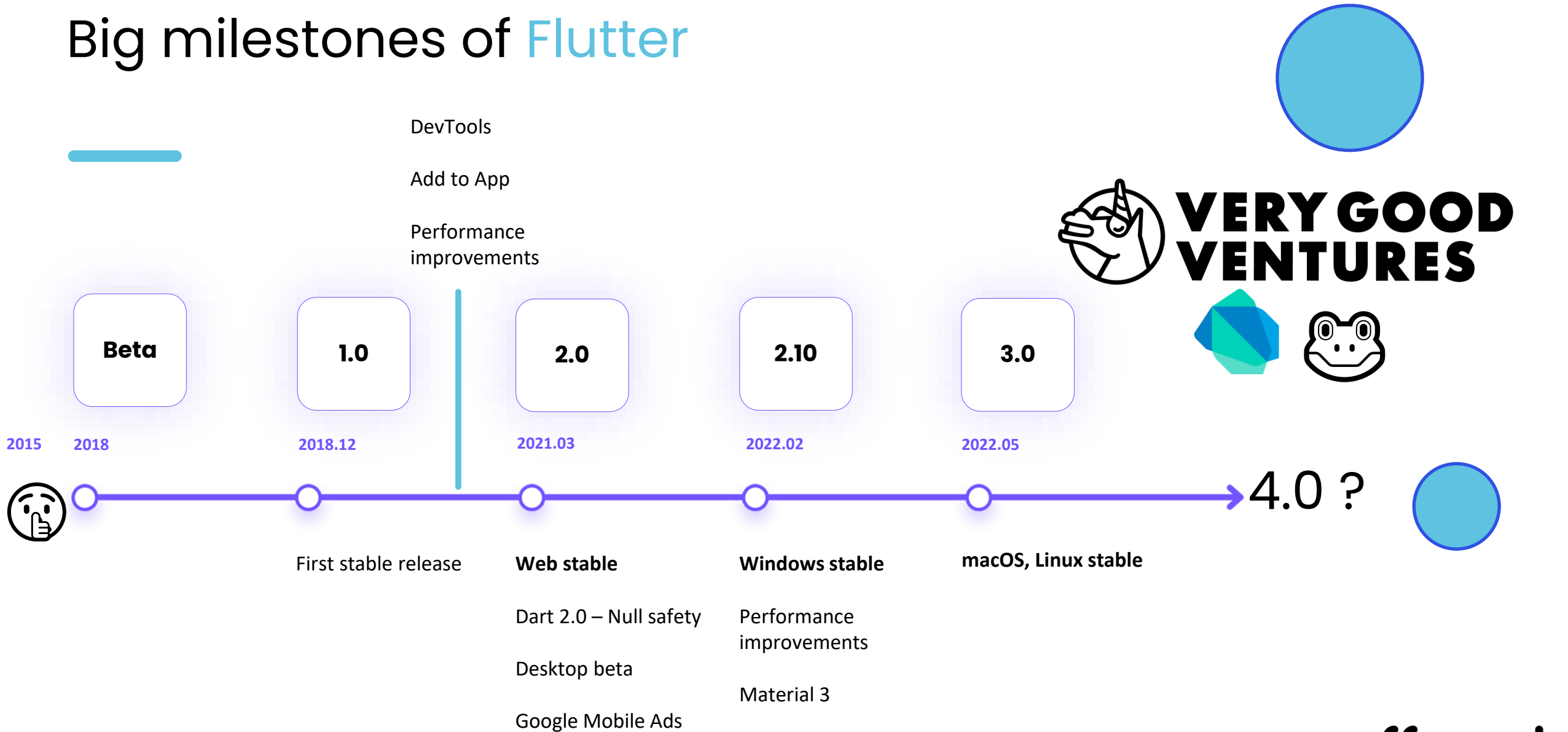
“ From many angles, this is **Flutter**’s biggest release yet”



# Big milestones of Flutter



# Big milestones of Flutter





## Companies using Flutter



TOYOTA



TIZEN™

H★MILTON



# Resources

- Introducing Flutter 3 – <https://medium.com/flutter/introducing-flutter-3-5eb69151622f>
- What's new in Flutter 3 – <https://medium.com/flutter/whats-new-in-flutter-3-8c74a5bc32d0>
- Flutter 3.0.0 release notes – <https://docs.flutter.dev/development/tools/sdk/release-notes/release-notes-3.0.0>
- Bring Material 3 to Flutter – <https://github.com/flutter/flutter/issues/91605>
- Flutter Casual Games Toolkit – <https://flutter.dev/games>



# Questions?



gerfalcon



GerfalconVogel

csongor.vogel@ffnext.io

