Flutter Basics

Prashanth B S[1]
[1]Department of Information Science & Engineering, Nitte Meenakshi Institute of Technology, Yelahanka - 560064, Bengaluru

1 Flutter

Flutter is a powerful language packed with a powerful mobile framework that can be used in both iOS and Android applications. Flutter is often used with DART, which is an object-oriented programming language by Google[1]. Flutter was introduced in the year 2017 in the dart summit targetting app development for the lower or mid-range devices. Flutter is an open source framework by Google for building beautiful, natively compiled, multi-platform applications from a single codebase[2]. Some of the features of flutter are,

1. Flutter code compiles to ARM or Intel machine code as well as JavaScript, for **fast performance** on any device
2. Deploy to multiple devices from a **single codebase**: mobile, web, desktop, and embedded devices.
3. Flutter is supported and used **by Google**, trusted by well-known brands around the world, and maintained by a community of global developers.
4. Flutter uses its **own SDK and set of widgets** and is not dependent on the native components
5. Flutter supports hot reload function, typical of web
6. Flutter is an UI-Toolkit/Library which are used with the help of dart programming language

Some of the advantages of using flutter are,

- Apps can *run on both iOS and Android platform*, no additional configuration is needed
- The *development and NRE cost is very less* for product development
- Apps can be built with *less time* since dart has less learning curve
- *Code written* are comparatively less compared to Native counterpart

The downside of using flutter are,

- Lack of native feel
- Bit sluggish on older devices and the performance depends upon the CPU used
- Having its own components has its downside, if needed a new feature one has to develop it on its own or wait till the developers make it available as add-on packages
- Large APK size compared to Java and Kotlin code

Some of the apps build by flutter are, Google Ads, Google Assistant, Baidu, My Leaf, TopLine

[1]https://www.geeksforgeeks.org/what-is-flutter/
[2]https://flutter.dev/

2 Environment Setup

The flutter can be setup in windows, macOS, Linux and Chrome OS as well. Read the documentation for the required installation steps[3],

- Windows: https://docs.flutter.dev/get-started/install/windows
- Linux: https://docs.flutter.dev/get-started/install/linux
- MacOS: https://docs.flutter.dev/get-started/install/macos
- ChromeOS: https://docs.flutter.dev/get-started/install/chromeos

3 Setting up an IDE

To code applications using Flutter, make sure the relevant steps are followed to install the packages in section 2. The next step is to use an IDE for coding. The possible choice of IDEs available are,

1. Android Studio with Flutter plugin
2. IntelliJ IDE with flutter plugin
3. VSCode with flutter add-on

4 Flutter Architecture

The following figure 1[4] shows the architecture of the flutter and the components of Flutter SDK as well. Flutter comes with full fledged SDK for building apps. That includes tools to create UI,

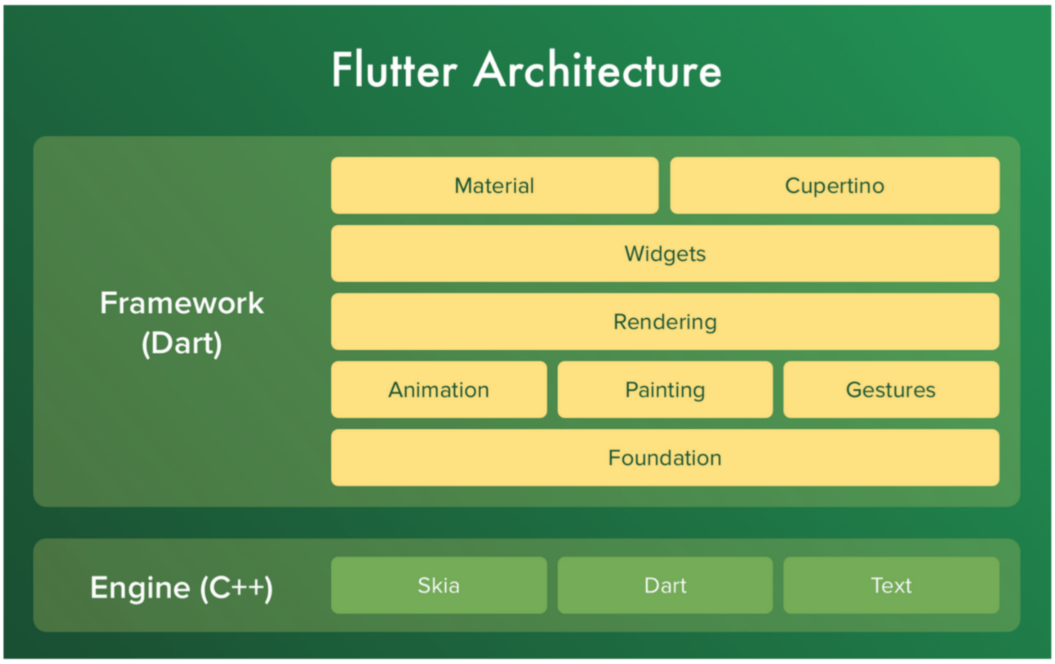


Figure 1: Flutter Architecture

widgets for both Android devices as well as iOS devices. The android apps uses *Material Design* based themes and Widgets, where as iOS apps uses *Cupertino* based themes and widgets. Flutter

[3] https://docs.flutter.dev/get-started/install
[4] https://www.cleveroad.com/blog/why-use-flutter

uses Dart for both server side and client side development and its completely open-source and object-oriented programming language. With the flexibility of dart language combined with C++ flutter core makes the app behave very close to the native ones in terms of performance.Flutter does not use native components in any form, so developers don't have to make bridges for communicating with them.Flutter draws the interface on its own. Buttons, text, media elements, background – all this is drawn inside the Skia graphics engine in Flutter[5].

5 Exercise

Create a sample Flutter application with a simple text at the center of the screen and apply styling to it.
The steps followed in Android Studio IDE are as follows,

1. Create a new flutter project and let the gradle build gets completed
2. Run the Emulator in the backend using AVD manager
3. Under the Project folder->lib->open the main.dart file which consist of the relevant template code which is by default a counter app which shows how many times the "+" button is pressed(Press the play button to run the app)
4. Delete everything in the main.dart file except the following lines,

```
import 'package:flutter/material.dart';

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

5. Change the app name inside the runApp as Materialapp as shown below ,

```
import 'package:flutter/material.dart';
void main() {
  runApp(MaterialApp());
}
```

6. Navigate to test folder under project directory, open the widget_test.dart and change the name of the app as shown below,

```
// Build our app and trigger a frame.
await tester.pumpWidget(const MyApp());  //by default

// change this line to
await tester.pumpWidget(MaterialApp()); // pointing to the app created
```

7. Draw and visualize a widget tree consisting of components such as,
   (a) A Homepage widget which houses appBar & body Widget
   (b) Scaffold widget acts as a container that houses appBar, body
   (c) appBar widget houses title widget
   (d) body widget houses a text widget

   Everything is treated as widget in flutter. The overall widgets forms the app UI and are visualized using Widget Tree prior implementation as shown in figure 2

[5] https://www.cleveroad.com/blog/why-use-flutter

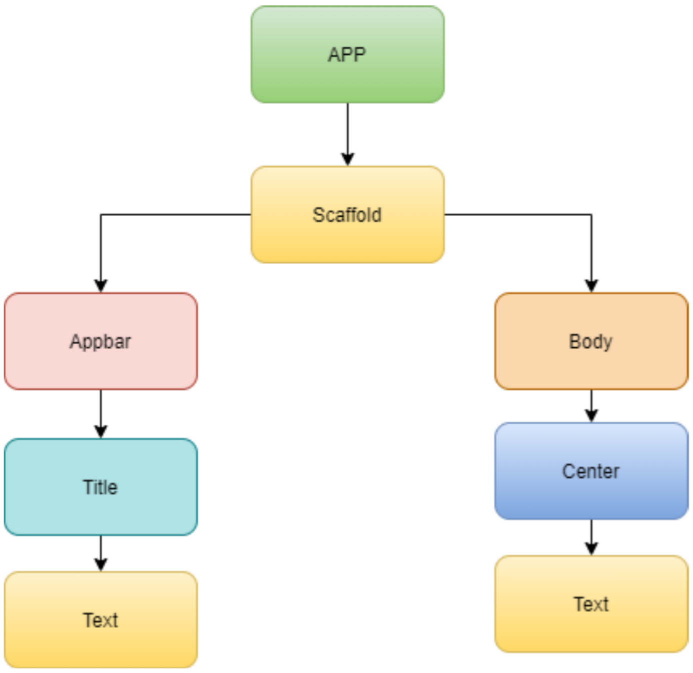


Figure 2: Widget Tree

8. Seeing the Widget Tree customize the MaterialApp with the following code snippet

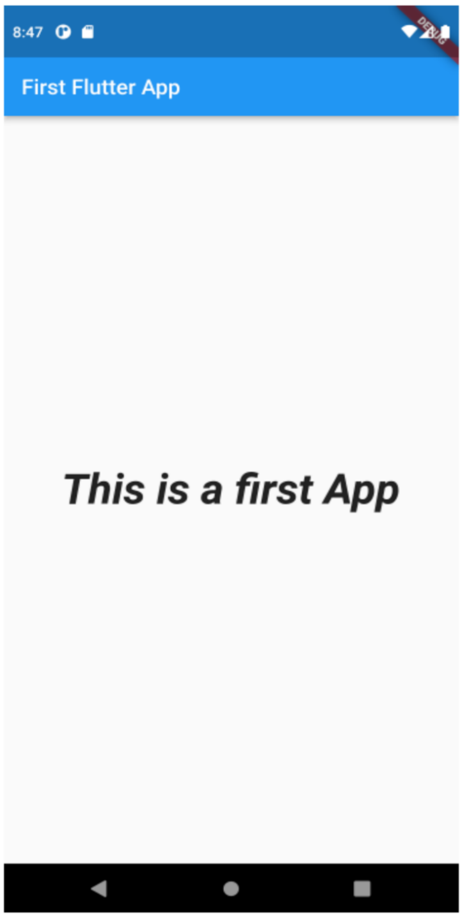
```
import 'package:flutter/material.dart';

void main() {
  runApp(
    MaterialApp(
      home:Scaffold(
        appBar: AppBar(
          title: Text("First Flutter App"),
        ),
        body: Center(
          child: Text(
            'This is a first App',
            style: TextStyle(
              fontSize: 40.0,
              fontStyle: FontStyle.italic,
              fontWeight: FontWeight.bold,
            ),
          ),
        ),
      ),
    ),
  );
}
```

9. Run the App and test out the design

6 Results

The output of the app is as shown below figure 3



This is a first App

Figure 3: Output of the App