



REPORT ON FLUTTER.

BY:

1. Akshat Saxena (201550016)
2. Nikhil (201550090)
3. Prakhar Verma (201550100)
4. Punnya Prasoon Gaur (201550108)
5. Vivek Kumar (2115990025)

DATE : 18-JUL-2022

Introduction

Flutter is an open-source mobile SDK developer can use to build native-looking Android and iOS applications from the same code base. Flutter has been around since 2015 when Google introduced it and remained in the beta stage before its official launch in December 2018. Since then, the buzz around Flutter has been growing stronger. Flutter is now the top 11 software repos based on GitHub stars. Moreover, we have already seen thousands of Flutter apps being published on app stores. One of the most notable examples is the Xianyu app created by Alibaba team, used by over 50 million people.



Android provides a native framework based on Java language and iOS provides a native framework based on Objective-C / Swift language. However, to develop an application supporting both the OSs, we need to code in two different languages using two different frameworks. To help overcome this complexity, there exists mobile frameworks supporting both OS. These frameworks range from simple HTML based hybrid mobile application framework (which uses HTML for User Interface and JavaScript for application logic) to complex language specific framework (which do the heavy lifting of converting code to native code). Irrespective of their simplicity or complexity, these frameworks always have many disadvantages, one of the main drawbacks being their slow performance. In this scenario, Flutter – a simple and high-performance framework based on Dart language, provides high performance by rendering the UI directly in the operating system's canvas rather than through native framework. Flutter also offers many ready to use widgets (UI) to create a modern application. These widgets are optimized for mobile environment and designing the application using widgets is as simple as designing HTML. To be specific, Flutter application is itself a widget. Flutter widgets also supports animations and gestures. The application logic is based on reactive programming. Widget may optionally have a state. By changing the state of the widget, Flutter will automatically (reactive programming) compare the widget's state (old and new) and render the widget with only the necessary changes instead of re-rendering the whole widget.

Features of Flutter

Flutter framework offers the following features to developers:

1. Modern and reactive framework.
2. Uses Dart programming language and it is very easy to learn.
3. Fast development.
4. Beautiful and fluid user interfaces
5. Huge widget catalogue.
6. Runs same UI for multiple platforms.
7. High performance application.

What else flutter can do?

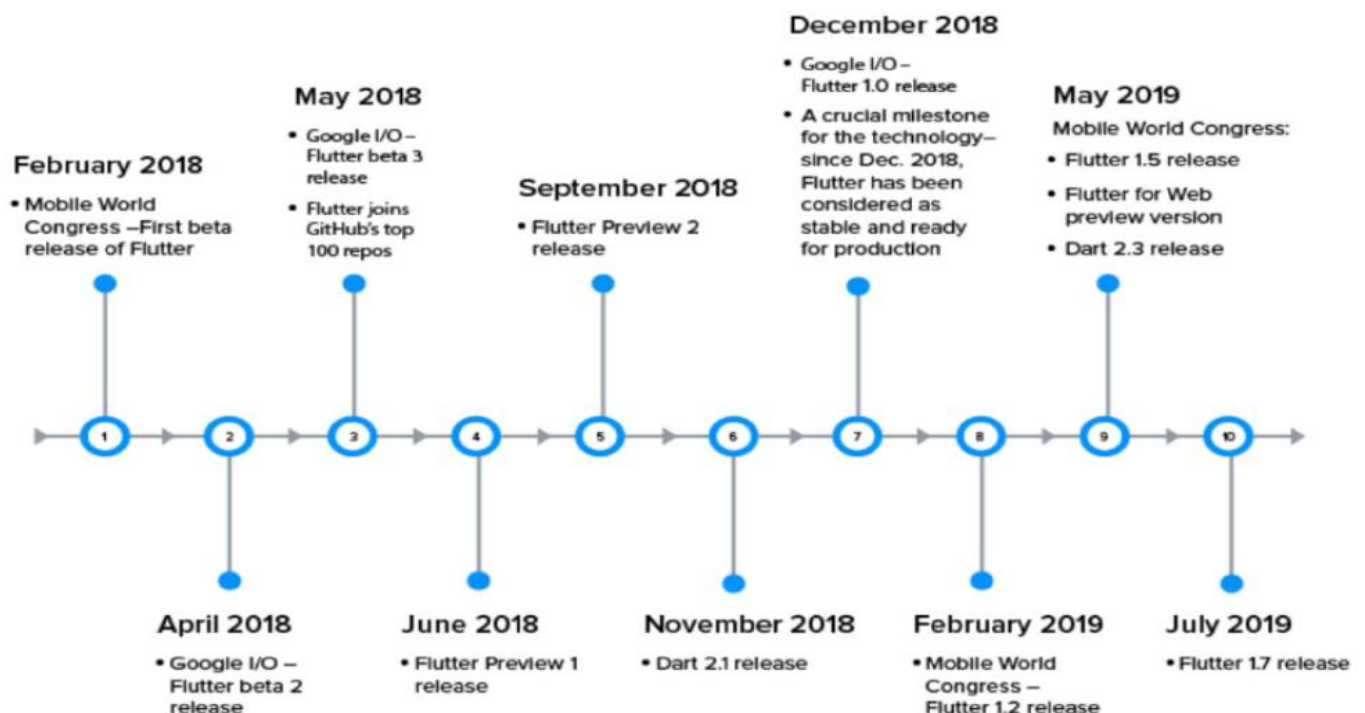
Machine learning (ML) is another trend in the mobile app world, as it enables personalization and accurate sales forecasts. The technology has been gaining more and more popularity in the last decade, and Flutter development allows for integrating ML into your app. Flutter offers a plugin, `firebase_ml_vision` 0.9.7, that gives access to Firebase's ML features and kits. Using this kit, even inexperienced Flutter developers can add ML components to their applications, such as face or speech recognition.

History

Flutter was first announced 4 years ago in 2015 at the Dart Developer Summit. It was initially called “SKY”. I think that would have been cooler. Flutter alpha (V-0.06) was released in May 2017. Later, Google launched the 2nd preview of Flutter back in September 2018. That’s when the developers around the world started to feel the heat. Even before the Flutter 1.0 was announced, many Flutter apps came into the stores. Apps like – Alibaba, Google Ads, Tencent, reflect were proof that Flutter could be so Awesome. Finally, Google launched Flutter 1.0, the stable version of Flutter on 5th of December 2018.

At this launch, Google released ‘The History of Everything’. A timeline of the world from the Big Bang to the age of WWW. An app built with Flutter with a single code base that can run in any Android or IOS device smoothly. This app has smooth animations because Flutter can run apps with a of 120 Frames Per Second. (Others only can achieve a maximum of 60 FPS).

Brief History



Why use Flutter?

1. Fast development.

Flutter engineered for high development velocity. Stateful hot reload allows you to change your code and see it come to life is less than a second without losing the state of the app. Flutter also ships with a rich set of customizable widgets, all built from modern reactive framework.

2. Expressive + Flexible UI.

Flutter moves to a widget, rendering, animation and gestures into this framework to give you to complete control over every pixel on the screen. It means you have the flexibility to build a custom design.

3. Native apps for Android and IOS.

Flutter apps follow platform conventions and interface details such as scrolling, navigation, icons, fonts, etc. That why apps built with Flutter features on both APP STORE and GOOGLE PLAY STORE.

4. Hot Reload.

In flutter, very save on the app and just as you do on the web just hit a refresh and your codes also refresh. Imaging that Facebook SDK it would be so humongous if it would have been designing android and you hit a recompile. So many things to have recompile and it would take probably days. Unmistakably, Hot reloading is tech which is kind of necessary when your applications or product goes like incense crazy like Facebook.

5. High Performance.

Flutter doesn't require a Javascript bridge and the speed is much faster.

6. Using Dart as a Programming Language.

Dart is an object-oriented programming language that which used for writing mobile application code for Flutter and which contributes to the efficiency and effectiveness of app development flow. It is a comfortable language and uses a lot of CSS parts as well.

7. Reduce the Third Parties.

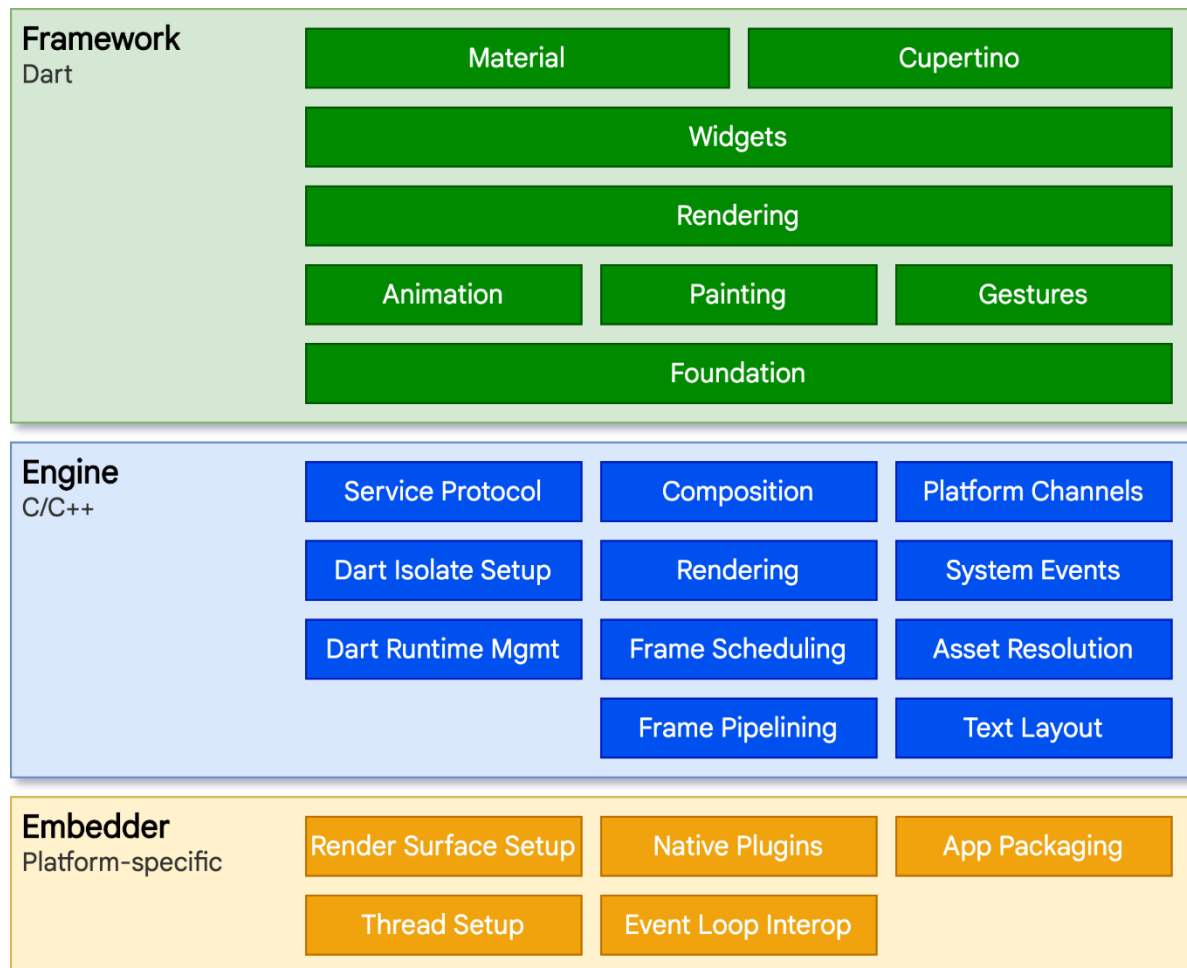
8. API. The Flutter API is very consistent.

9. The customizable kit of Widgets.

The Core Principle of Flutter

Flutter includes a modern react-style framework, a 2D rendering engine, ready-made widgets, and development tools. These components work together to help you design, build, test, and debug apps. Everything is organized around a few core principles.

1. Flutter architecture.



The Flutter architecture uses a layered design, which is divided into three layers from bottom to top, in this order: Embedder, Engine, Framework.

1. Embedder : Operating system adaptation layer to realize rendering Surface settings, thread settings, etc.
2. Engine : Realize functions such as Flutter rendering engine, text layout, event processing, and Dart runtime. Including Skia graphics drawing library, Dart VM, Text, etc., among which Skia and Text provide the ability to call the underlying rendering and typesetting for the upper layer interface.

3. Framework : It is a UI SDK implemented with Dart. From top to bottom, it includes two major style component libraries, basic component libraries, graphics drawing, gesture recognition, animation and other functions.

Who is Flutter for?

Designers converge on a brand-driven experience on Android and iOS

Prototypers enjoy a high-fidelity and fast way to build working prototypes.

Developers benefit from fantastic developer tools, an easy-to-use language, a rich set of widgets and great IDE support. Flutter frees up valuable time for working on features and delightful experiences.

cross-platform applications are a long-standing dream of any business, because separate native iOS and Android apps are more expensive to develop and maintain. Flutter offers a quick way to create visually attractive apps for both operational systems, desktop computers, and the web from a single codebase. It is actively used and promoted by Google, and we expect that it will become universally popular in the near future.

1. Low-cost app development

An ideal cross-platform should meet two requirements: provide a high-quality user experience (smooth animations, native UI elements without slowing down), and be cost-friendly from the development perspective.

The advantage of Flutter is not only in making one application instead of two. With this framework, you will also have fewer problems when adapting your app to each of the platforms. Most frameworks can cope, but that's only if to consider them in the context of simple applications. The real problems begin when an application becomes "overgrown" with functions and inevitably gets complicated (it is about most products during the development stage).

2. It goes beyond mobile

As we mentioned, the technology appeared as Flutter 1.0 in December 2018. Since then, Flutter has demonstrated its ability not only to work as a toolkit for mobile platforms but also for web and desktop browsers. In May 2019, its development team announced a Flutter preview version for the web which was first called Hummingbird. However, it was later renamed Flutter Web.

3. Simple logic implementation

Flutter provides advanced OS features like GPS coordinates, sensor data collection, permission handling, Bluetooth, credentials, and other features in ready-to-use plugins that are supported by Google.

If your app is reliant on an OS-level feature not available as a plugin, Flutter can establish communication between its Dart programming language and the native code using platform channels.

4. Customizable kit of widgets

Widgets are the basic blocks used to build a user interface of a Flutter application. Its layering makes it easy to create applications of any complexity. You can use ready-made widgets from Material-UI or implement specific components based on basic widgets.

Efficient Tooling

\$ flutter doctor

Checks your environment and displays a report to the terminal window

\$ flutter upgrade

Updates both the Flutter SDK and your packages

\$ flutter packages get

Checks your environment and displays a report to the terminal window

\$ flutter packages upgrade

Will retrieve the highest available version of the package

\$ flutter format

Automatically formats your code according to the Flutter-style

\$ flutter analyze

Analyzes your code and help you find possible mistakes\

What is Widget?

- In flutter, Widget is a way to declare and construct UI.
- If you are familiar with the Android or iOS development, then you might make the immediate connection with the views (on Android) or UIViews (on iOS).
- Widget is a lot more than just structural elements like buttons, text, image, list or slider. A widget might display Something, it might help define design, it might help with layout, it may handle user interaction, etc.
- For example, Padding is a widget, Margin is a widget, Center is a widget, Layout rows and columns are also widgets.
- So you can consider that a widget is a blueprint. Flutter uses these blueprints to create views.

Types of Widgets

- Flutter has a rich set of in-built widgets like text, buttons, slider, lists, layouts, gesture detector, animations, etc. Flutter team works really hard to create a set of widgets that helps you in almost every situation. And they are continuously adding more widgets as developers needs.
- But apart from built-in widgets, you can create your own widgets according to your needs.

Flutter divides widgets into two categories:

1. Stateless Widgets

- In simple words, if a widget doesn't do anything its Stateless Widget. They are static in nature.
- Stateless widgets don't store any state. That means they don't store values that might change.

2. Stateful Widgets

- In simple words, if a widget does anything then its Stateful Widget.

- A Stateful widget is dynamic in nature. That means it can keep track of changes and update the UI based on those changes.
- Radio, Slider, Form and Text Field are an example of stateful widgets.

What is Dart?

Google had its first ever release of Flutter 1.0 last December, after having it in beta mode for over 18 months. Dart is the programming language used to code Flutter apps. Dart is another product by Google and released version 2.1, before Flutter, in November. As it is starting out, the Flutter community is not as extensive as React Native, Ionic.

Dart is not only used for mobile app development but is a programming language. Approved as a standard by Ecma (ECMA-408), it's used to build just about anything on the web, servers, desktop and of course, mobile applications (Yes, the same people who standardized our favorites ES5 and ES6.)

Dart, when used in web applications, is transpiled to JavaScript so it runs on all web browsers.

The Dart installation comes with a VM as well to run the .dart files from a command-line interface. The Dart files used in Flutter apps are compiled and packaged into a binary file (.apk or .ipa) and uploaded to app stores.

1. What does coding in Dart look like?

Like most ALGOL languages (like C# or Java):

1. The entry point of a Dart class is the main() method. This method acts as a starting point for Flutter apps as well.
2. The default value of most data types is null.
3. Dart classes only support single inheritance. There can be only one superclass for a particular class but it can have many implementations of Interfaces.
4. The flow control of certain statements, like if conditions, loops (for, while and dowhile), switch-case, break and continue statements are the same.
5. Abstraction works in a similar manner, allowing abstract classes and interfaces.

2. Dart specification

1. Dart has type inference. The data type of a variable need not be explicitly declared, as Dart will “infer ”what it is. In Java, a variable needs to have its type explicitly given during declaration. For example, String something;. But in Dart, the keyword is used instead like so, var something;. The code treats the variable according to whatever it contains, be it a number, string, bool or object.
2. All data types are objects, including numbers. So, if left uninitialized, their default value is not a 0 but is instead null.
3. A return type of a method is not required in the method signature.
4. The type num declares any numeric element, both real and integer.
5. The super() method call is only at the end of a subclass’s constructor.
6. The keyword new used before the constructor for object creation is optional.
7. Method signatures can include a default value to the parameters passed. So, if one is not included in the method call, the method uses the default values instead.
8. It has a new inbuilt data type called Runes, that deal with UTF-32 code points in a string. For a simple example, see emojis and similar icons.

Dart also has inbuilt libraries installed in the Dart SDK, the most commonly used being:

1. dart:core for core functionality; it is imported in all dart files.
2. dart:async for asynchronous programming.
3. dart:math for mathematical functions and constants.
4. dart:convert for converting between different data representation like JSON to UTF-8.

3. Using Dart in Flutter

Flutter has more app-specific libraries, more often on user interface elements like:

1. Widget: common app elements, like the Text or ListView.
2. Material: containing elements following Material design, like FloatingActionButton.
3. Cupertino: containing elements following current iOS designs, like CupertinoButton.

Conclusion

Flutter framework does a great job by providing an excellent framework to build mobile applications in a truly platform independent way. By providing simplicity in the development process, high performance in the resulting mobile application, rich and relevant user interface for both Android and iOS platform, Flutter framework will surely enable a lot of new developers to develop high performance and feature-full mobile application in the near future.

References

Blog: What's Revolutionary about Flutter by Wm Leler: goo.gl/bZcFR9

Video: Flutter's Rendering Pipeline by Adam Barth: youtu.be/UUfXWzp0-DU

Video: The Mahogany Staircase by Ian Hickson: youtu.be/dkyY9WCGMi0