University of Sulaimani

College of Science – Computer Dep.

2025-2026

By: Mohammed Q. Kheder

2nd Lecture

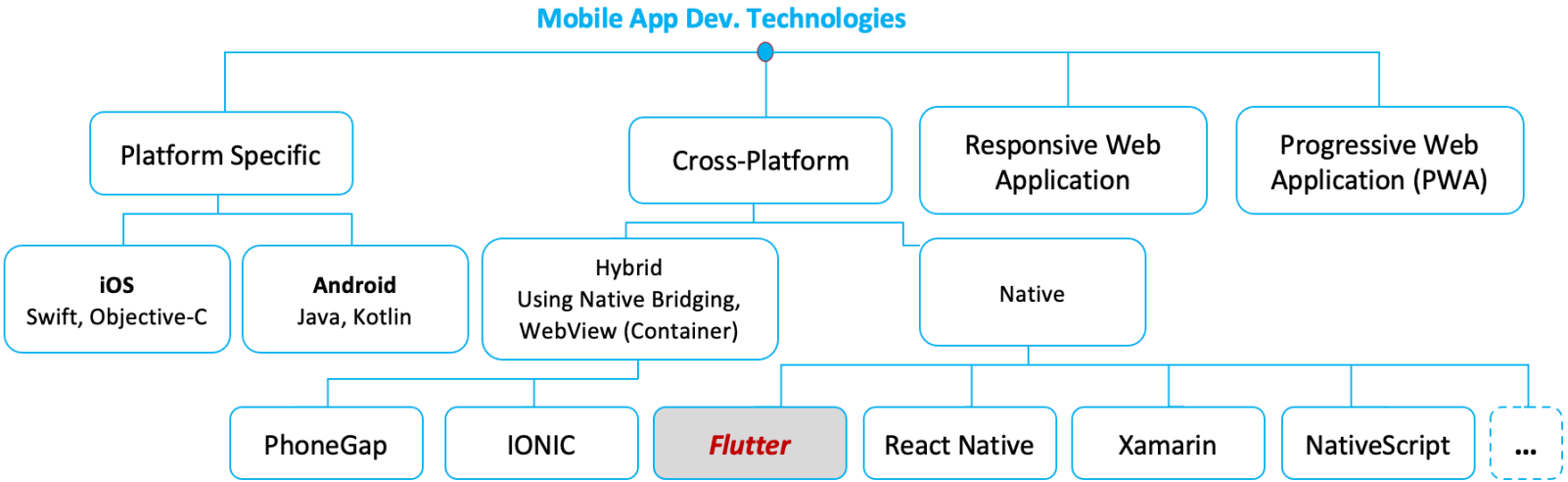
Mobile Apps using Flutter Framework

4 October, 2025

Outline

- Mobile App Dev. Technologies
- What is Flutter?
- Why Flutter?
- Flutter Architecture
- Dart Programing Language
- Why Dart?
- Flutter vs. Dart
- Flutter IDE

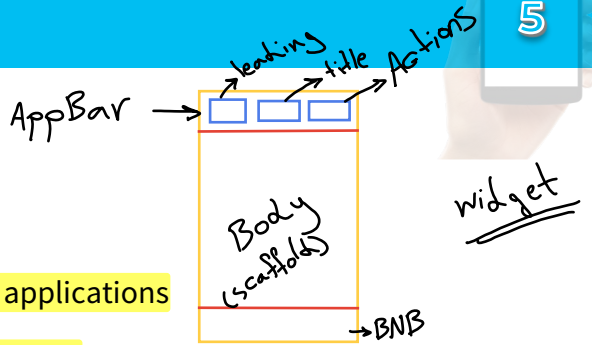
Mobile App Dev. Technologies



What is Flutter?

- **Flutter** is a free and open-source mobile UI framework created by Google, and released in May 2017
  - latest version is 3.35.5, released on 29 September, 2025
- It allows developers to build high-performance, natively compiled applications for mobile, web, and desktop from a single codebase
- Cross-platform: Android, iOS, Desktop (Windows, MacOS, Linux), and Web
- Flutter uses the **Dart** programming language, which is also developed by Google in October 2011
  - newest version is 3.9.2, which was released on 4 September, 2025

What is Flutter? (2)



Flutter consists of **two** important parts:

1. **SDK** *(Software Development Kit)*
   - a collection of tools that are going to help you develop your applications
   - this includes tools to compile your code into native machine code (code for iOS and Android)
2. **Framework** *(UI Library based on Widgets)*
   - a collection of reusable UI elements (buttons, text inputs, sliders, and so on) that you can personalize for your own needs

كيف يفرق بين اثنين Button ؟
كلهم نفس نوع العنصر راح يكون لهم نفس ID
لكن راح نفرق بينهم بواسطة ال API

شلون فلاتر يفرق مثلاً بين Button و TextField ؟
نفرق بواسطة ال API

Why Flutter?

1. Fast Development Cycle

- Flutter's "hot reload" features allows developers to make changes to the code and see the results instantly, without having to wait for the app to recompile. This makes development faster and more efficient

2. Single Codebase for Multiple Platform

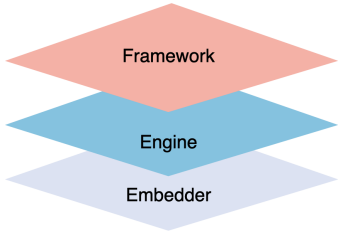
- Flutter allows developers to build apps for multiple platforms, including iOS, Android, and web, from a single codebase

Why Flutter? (2)

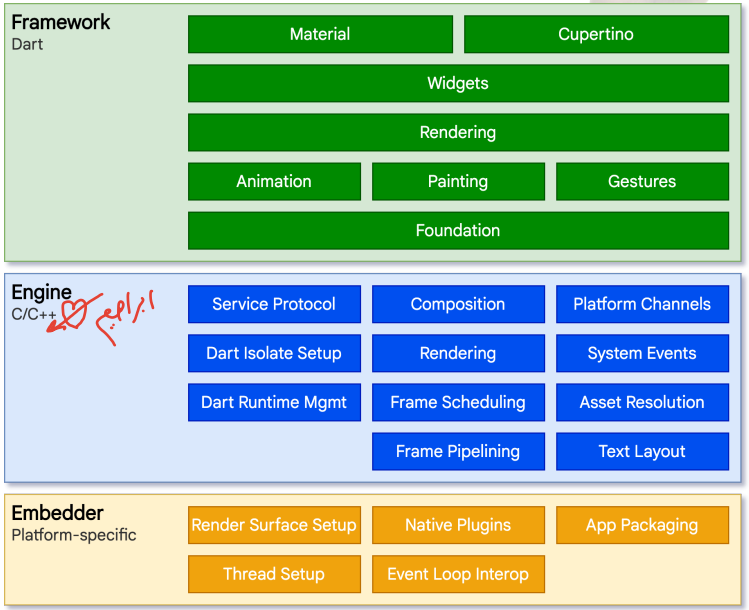
3. **Customizable Widgets**
   - Flutter provides a wide range of customizable widgets and an easy-to-use UI toolkit that allows developers to create unique and engaging designs
4. **High Performance**
   - Flutter apps are compiled to native machine code, which makes them fast and efficient

Flutter Architecture

- The flutter framework is a layered system with each layer dependent on the layer below
- A single layer has several independent libraries
- On a high level, the flutter framework's architecture has three main parts or layers:

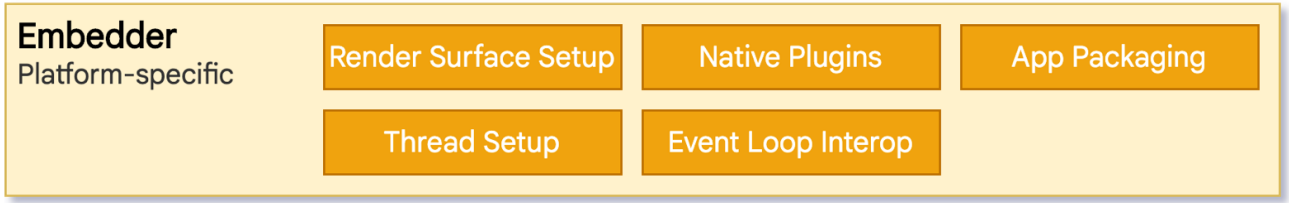


- Framework
- Engine
- Embedder



Flutter Architecture (2)

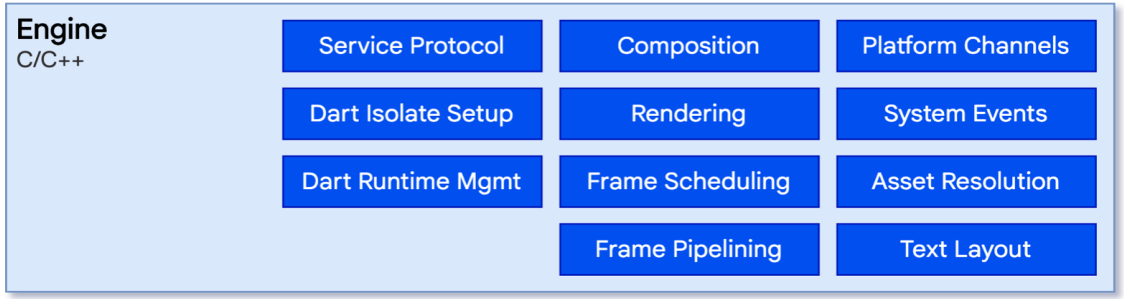
Embedder Layer



- The embedder layer has platform-specific embedders, many of which provide an entry point and coordinate with their respective operating system to access services like *rendering*, *storage*, and more
- Java and C++ are used in embedders for Android, objective-C/ C++ for IOS and macOS, and C++ for Linux and windows
- *Note:* As flutter is open-source, **we can check the code and change it according to our needs**

Flutter Architecture (3)

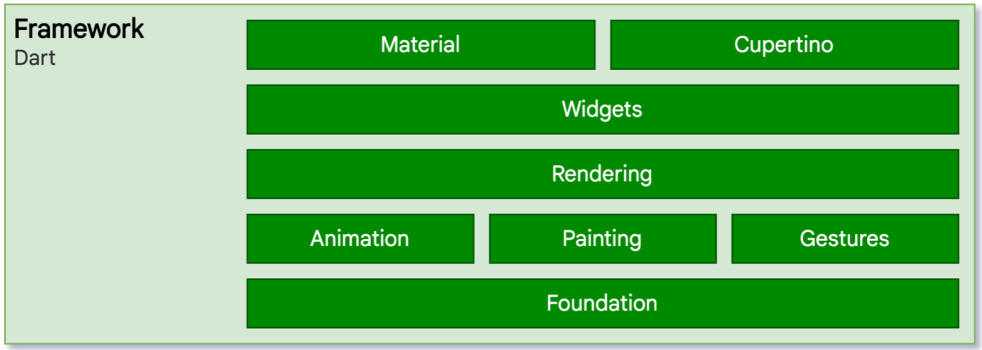
Engine Layer



- The engine layer is the core of flutter written in C/C++
- It is responsible for taking care of input, output, and rasterizing composited scenes, as flutter is a UI toolkit. It uses the ***skia*** library for rendering graphics
- It is also responsible for service and network protocols, such as network input and output, file management, and the core API of flutter

Flutter Architecture (4)

Framework Layer



- In the framework layer, the developer interacts with and writes flutter applications
- It is written in dart language and has predefined libraries, layouts, and more
- The framework layer has three main layer components, which are the following:
  1. *Foundation layer*
  2. *Rendering layer*
  3. *Widget layer*

Dart Programing Language

- **Dart** is an ***open-source***, ***scalable programming*** language for building web, and mobile apps
  - the purpose of **Dart** programming is to create a ***frontend*** user interfaces for the web and mobile apps
- It is **purely OOP**, dynamic language with C-style syntax
- It supports **optional static** typing and type checks
- Adopts single **inheritance** with mixins supports
- Influenced by strongly type languages like *Java, C++ and C#*, and loosely type dynamic language like *JavaScript*

Dart Programing Language (2)

- Dart is widely used to develop the mobile app, modern web-applications, desktop application, and the Internet of Thing (IoT) using by Flutter framework
- It is a compiled language and supports two types of compilation techniques
  - **AOT** (**Ahead of Time**)- it converts the *Dart code* in the optimized *JavaScript code* with the help of the ***dar2js*** compiler and runs on all modern web-browser. It compiles the code at build time
  - **JIT** (**Just-In-Time**)- it converts the *byte code* in the machine code (native code), but only code that is necessary

Example of Dart Program

- To learn more about Dart programming language use the bellow tutorials:
  - https://www.tutorialspoint.com/dart_programming/index.htm
  - https://www.javatpoint.com/flutter-dart-programming

```
void main() {
    var num = 12;
    if (num % 2 = = 0) {
        print("Number is Even.");
    } else {
        print("Number is Odd.");
    }
}
```

Why Dart?

- Dart is a **platform-independent language** and supports all operating systems such as Windows, Mac, Linux, etc.
- It is an **open-source language**, which means it available free for everyone. It comes with a BSD license and recognized by the ECMA standard
- It is an **object-oriented programming language** and supports all features of oops such as inheritance, interfaces, and optional type features
- Dart is very useful in building **real-time applications** because of its stability
- Dart comes with the ***dar2js*** compiler which transmits the Dart code into JavaScript code that runs on all modern web browser
- The **stand-alone Dart VM** permits Dart code to run in a command-line interface environment

Flutter vs. Dart

framework

language

	Flutter	Dart
Purpose	It is a framework for building mobile apps	It is a general purpose programming language
Syntax	It uses a reactive-style programming model called widgets	It has a syntax similar to other object-oriented languages such as Java or C++
Architecture	It uses a widget-based architecture, where everything is a widget	It is designed to support a variety of programming paradigms including object-oriented and functional programming
Platform	It is cross-platform, meaning that it can run on both Android and iOS	It can be used for a variety of apps including web development and server-side development

Flutter IDE

- IDE (editors) are software programs that allow the user to create and edit text files
- We can create Flutter apps using any text editor that can easily combine with our command-line tools
- Flutter supports several IDE to build the apps, including:
  - Android Studio
  - **Visual Studio Code**
  - IntelliJ Idea
  - Emac
  - Codemagic

Flutter IDE (2)

- Reliable IDEs or text editors are important tools for app development
- IDEs offer several benefits for app development, namely, *code completion*, *debugging*, *collaboration*, and *customization*
- However, with so many options available, it can be challenging to choose one
- Considerations when choosing the best IDE for Flutter development

1. **Features**
2. **Ease of use**
3. **Platform compatibility**
4. **Community support**
5. **Cost**
6. **Integration with other tools**
7. **Performance**

Flutter IDE (3)

Visual Studio Code

- It is a lightweight and versatile text editor that is widely used for Flutter development
- It provides a range of features and extensions that can help streamline the development process, including syntax highlighting, code completion, debugging, and Git integration
- The key features that make VSC a great option as a Flutter IDE:
  - lightweight and customizable
  - Flutter-specific extensions and plugins
  - code completion and debugging features
  - hot reload for faster iteration and development
  - integration with Git and other development tools

Assignment#1

Explain the differences between **Flutter** and **React Native** in terms of the following aspects:

- ✓ Programming Language
- ✓ User Interface
- ✓ Performance
- ✓ Security
- ✓ Development Environment

THE END