Flutter Tutorial 2: Flutter Architecture

Introduction to Flutter Architecture

Flutter has a unique architecture that allows it to build **high-performance, cross-platform apps**. Unlike other frameworks, Flutter does not rely on platform-specific components for rendering. Instead, it uses its own **widgets and rendering engine** to create the user interface.

Flutter architecture can be understood in three main layers:

1. Framework Layer (Dart Code Layer)

- This is the top layer where developers write their app code using **Dart** language.
- It includes **widgets, rendering, and animation libraries**.
- Everything in Flutter is a **widget**, including layout, buttons, text, and images.
- This layer provides **easy-to-use APIs** for UI design, gestures, and navigation.

2. Engine Layer (C++ Layer)

- The middle layer is the **Flutter engine**, written in **C++**.
- Provides low-level **rendering support, text layout, graphics, and network requests**.
- Uses **Skia graphics library** to draw UI on the screen.
- Handles **Dart runtime and garbage collection**, ensuring smooth app performance.

3. Embedder Layer (Platform-Specific Layer)

- The bottom layer connects Flutter to the underlying platform (**Android, iOS, web, desktop**).
- Provides **platform-specific functionalities** like accessing camera, sensors, storage, or device hardware.
- Allows Flutter apps to **run on multiple platforms** without changing the Dart code.

Key Features of Flutter Architecture

- **Widget-Based**: Everything is a widget, making UI development modular and reusable.
- **Reactive Framework**: UI automatically updates when the app state changes.
- **High Performance**: Uses its own rendering engine, so apps are fast and smooth.
- **Cross-Platform**: Single codebase works for multiple platforms.
- **Skia Rendering Engine**: Draws every pixel on the screen, ensuring consistent UI across platforms.

Conclusion

In simple words, Flutter's architecture allows **fast development, smooth animations, and consistent UI across different devices** by controlling every layer from widgets to rendering.

End of Flutter Tutorial 2