Mobile Application Development with Flutter

Lecture 3: Flutter Architecture

Flutter Architecture

- Flutter has a **layered architecture**, which helps it achieve **high performance**
- Flutter consists of three main components:

1. Framework
2. Engine
3. Embedder

1. Flutter Framework

- The **Framework** is the top layer where **developers write code**
- It is written in **Dart** and includes:

- Widgets
- UI components
- Animation libraries
- Material & Cupertino widgets

What are Widgets in Flutter?

- In Flutter, **everything is a widget** — buttons, text, layouts, screens, even the app itself
- Flutter provides **two main widget design systems**:

- Material Widgets
- Cupertino Widgets

What are Material Widgets?

- **Material widgets** follow **Google's Material Design** guidelines
- They are mainly used for **Android-style UI**, but they also work on iOS and web

- When to use Material Widgets?
  - Android apps
  - Cross-platform apps with a single UI
  - When you want **ready-made UI components**
  - Faster development

What are Cupertino Widgets?

- **Cupertino widgets** follow **Apple's iOS Human Interface Guidelines**
- They give your app a **native iOS look and feel**
- When to use Cupertino Widgets?
  - iOS-only apps
  - When you want **native iOS experience**
- Can we use BOTH together?
  - Flutter allows mixing **Material and Cupertino widgets** in the same app

Responsibilities of Framework

- Builds UI using widgets
- Handles app logic
- Manages state
- Provides ready-to-use components

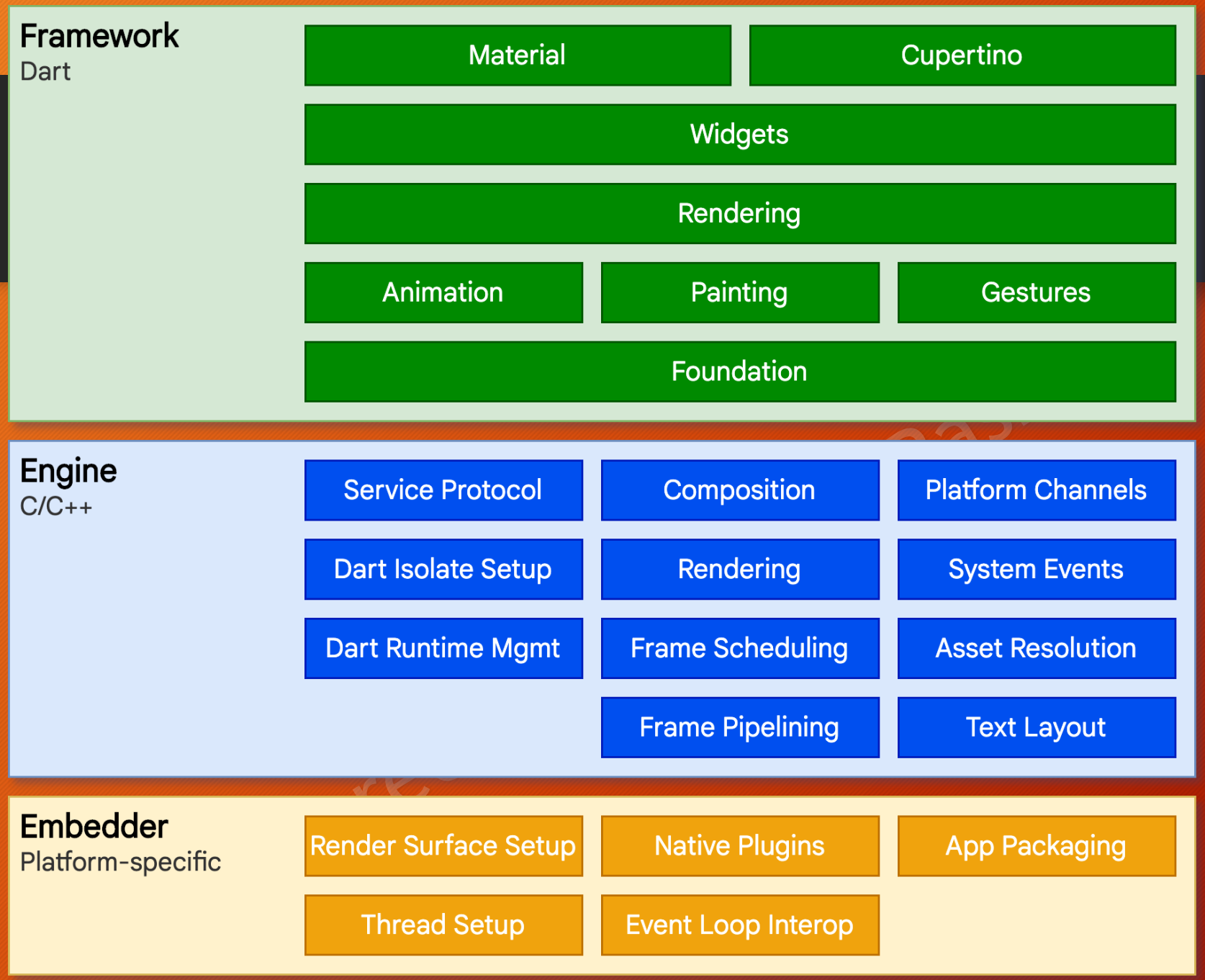
- Everything you see on screen in Flutter is a widget

2. Flutter Engine

- The **Engine** is the core runtime of Flutter
- It is written in C++
- Responsibilities of Engine
  - Rendering UI
  - Handling animations
  - Managing text, images, and graphics
  - Communicating with native APIs
- Engine is responsible for performance

3. Flutter Embedder

- The **Embedder** connects Flutter with the **underlying operating system**
- It is platform-specific:
  - Android Embedder
  - iOS Embedder
- Responsibilities of Embedder
  - App startup
  - Window management
  - Input handling (touch, keyboard)
  - Communication with OS
- Embedder acts as a bridge between Flutter and the OS


Framework
Dart
Material
Cupertino
Widgets
Rendering
Animation
Painting
Gestures
Foundation
Engine
C/C++
Service Protocol
Composition
Platform Channels
Dart Isolate Setup
Rendering
System Events
Dart Runtime Mgmt
Frame Scheduling
Asset Resolution
Frame Pipelining
Text Layout
Embedder
Platform-specific
Render Surface Setup
Native Plugins
App Packaging
Thread Setup
Event Loop Interop

Flutter vs React Native (High-Level Comparison)

- Flutter
  - Developed by **Google**
  - Language: **Dart**
  - UI rendered using Flutter's own engine
  - Consistent UI across platforms
  - Near-native performance
- React Native
  - Developed by **Meta (Facebook)**
  - Language: **JavaScript**
  - Uses native UI components
  - Depends on JavaScript bridge
  - Performance slightly lower than Flutter

Comparison Table (Flutter vs Reactive Native)

Feature	Flutter	React Native
Language	Dart	JavaScript
UI Rendering	Own engine	Native components
Performance	High	Moderate
Hot Reload	Yes	Yes
UI Consistency	High	Depends on platform
Learning Curve	Moderate	Easy (for JS devs)

Flutter provides better UI consistency and performance