How does flutter work behind the scenes?

- Flutter SDK and Flutter Framework
- Compilation task is done by Flutter SDK
- Compile dart code to native machine code
- How does Flutter run my code on Android and iOS?
  - The engine’s C and C++ code are compiled with Android’s NDK or (for iOS) LLVM.
  - The Dart code (both the SDK’s and yours) are ahead-of-time (AOT) compiled into native, ARM, and x86 libraries.
  - Those libraries are included in a “runner” Android project or iOS Project, and the whole thing is built into an APK or iPA.
  - This is similar to the way many game engines work.
  - Debug mode builds use a virtual machine (VM) to run Dart code (hence the “debug” banner they show to remind people that they’re slightly slower) in order to enable Stateful Hot Reload. (Just-In-Time)

What kinds of apps can I build with Flutter?

Flutter is optimized for 2D mobile apps.

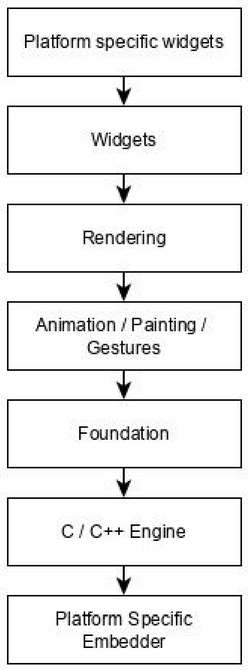
What makes Flutter unique compared to other Cross-platform?

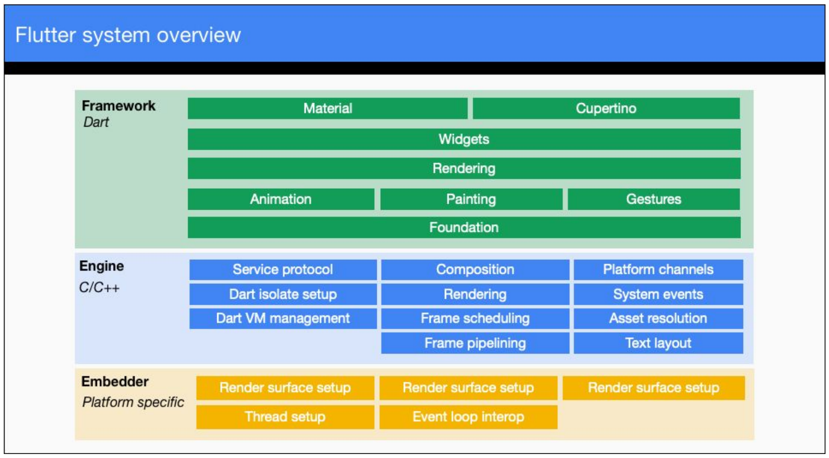
Flutter uses neither WebView nor the OEM widgets that shipped with the device. Instead, Flutter uses its own high-performance rendering engine to draw widgets.

- Mobile-first 2D rendering engine: Skia, to draw widgets.
- Flutter don't reuse the OEM widgets, Means no bridging like React Native

Layers of flutter framework

- The Flutter framework is organized into a series of layers, with each layer building upon the previous layer in terms of complexity and clearly arranged in layers of decreasing complexity.


Platform specific widgets
Widgets
Rendering
Animation / Painting / Gestures
Foundation
C / C++ Engine
Platform Specific Embedder



- The top most layer is a widget specific to Android and iOS.
- The next layer has all flutter native widgets.
- The next layer is the Rendering layer, which is a low level renderer component and renders everything in the flutter app.

Example how to use each level: https://github.com/flutter/flutter/tree/master/examples/layers

What language is Flutter written in?

Dart for the framework and widgets. The underlying graphics framework and the Dart virtual machine are implemented in C/C++.

What devices and OS versions does Flutter run on?

- Mobile operating systems: Android Jelly Bean, v16, 4.1.x or newer, and iOS 8 or newer.
- Mobile hardware: iOS devices (iPhone 4S or newer) and ARM Android devices.

Can I use Flutter inside of my existing native app?

- Yes, one fullscreen Flutter instance can be integrated per app on Android and iOS. See the integration documentation

Can I access platform services and APIs

- Flutter gives developers out-of-the-box access to some platform-specific services and APIs from the operating system
- A number of platform services and APIs have ready-made packages available on **pub.dev**. https://pub.dev/flutter/packages
- You can use Flutter’s asynchronous message passing system to create your own integrations with platform and third-party APIs, See How?
  See Demo App on Use Background service with Flutter Plugins
  Executing Dart in the Background with Flutter Plugins and Geofencing

Pros

- One code base
- Own Rendering engine gives high performance while rendering UI (compare to other cross platform)
- You can easily/quickly (code less to) build beautiful UI. (with **composition** and declarative widget tree)
- Large number of pre build widgets. (In Material and cupertino)
- Hot reload - Fast development

Neutral

- Network request
- Architecture, (Provider, Blogs etc)

Cons

- Flutter requires native developers to build third party Plug-Ins to access native features (Bluetooth, Push notification, background services and many more). To utilise newer OS features need to depend on Flutter communities, But you can also create plugins, if you had experty on Native development.

- Large file size: Project size - Native project 50 mb vs Flutter Project - 500 mb
  App file size: 1mb vs 8 to 10 mb
  Flutter knows this issue, they have given guidelines for this.