The Flutter project is split into a number of repositories, and we sometimes get asked why we don't use a monorepo.

Separate products

There are several reasons. The core reason is that each repo is a separate product.

The engine repository is where we build the library that executes Dart code. It defines an API used to embed Flutter in applications, and an API for Dart code to interact with the library. This is a standalone product that can be used on its own, in principle (though doing so would require creating a whole framework in Dart).

The flutter repository is where we build our framework that wraps the engine's APIs and makes them easy to use. This library is intentionally optional when using Flutter; it could be entirely replaced without affecting the engine.

The plugins and packages repos are where we put the packages we ship to pub, each of which are also separate products.

Clean separation

These different products each have APIs to talk to each other. It's important to keep the APIs clean. Any time you put code inside a repository such that the friction for communicating between them is lowered, you run the very real risk that accidental dependencies will be introduced between the two APIs. Keeping the repositories separate helps prevent that kind of intermingling.

Licensing

We intentionally keep the flutter/flutter repository separate from the others because it is single-licensed, unlike our other repositories.

Repository size

Since flutter/flutter is the main repository downloaded by developers, we want to keep it relatively small. This means, for example, that it cannot contain binaries or third-party packages.

A full checkout of the flutter/engine repository takes >10gb of local space. Additional artifacts for builds of various platforms and release modes can easily take 10s of gigabytes more space.

Approachability

Since flutter/flutter is the main repository for new contributors, we want to keep it approachable, which means minimizing the amount of complexity present in the repo itself.

Rolls

The flutter/flutter repository depends on binaries built from the flutter/engine repo. If the two repositories were merged, we'd either have a confusing situation where part of the repository would depend on binaries built from an older commit of the same repository, or we'd require that developers on the repo build the binaries locally. Both are more confusing than the current setup.