SONY

Flutter for Embedded Systems

05/27/2021

Hidenori Matsubayashi

Hidenori.Matsubayashi@sony.com

HidenoriMatsubayashi

R&D Center, Sony Group Corporation

Announcing Flutter at Google I/O 2021

Google I/O 2021 keynote: https://youtu.be/a553D0s7HeE?t=1778

Flutter for embedded use - Sony

More than a Google project

While Google continues to be the primary contributor to the Flutter project, we're delighted to see the growth of the broader ecosystem around Flutter.

TOYOTA CANONICAL SONY

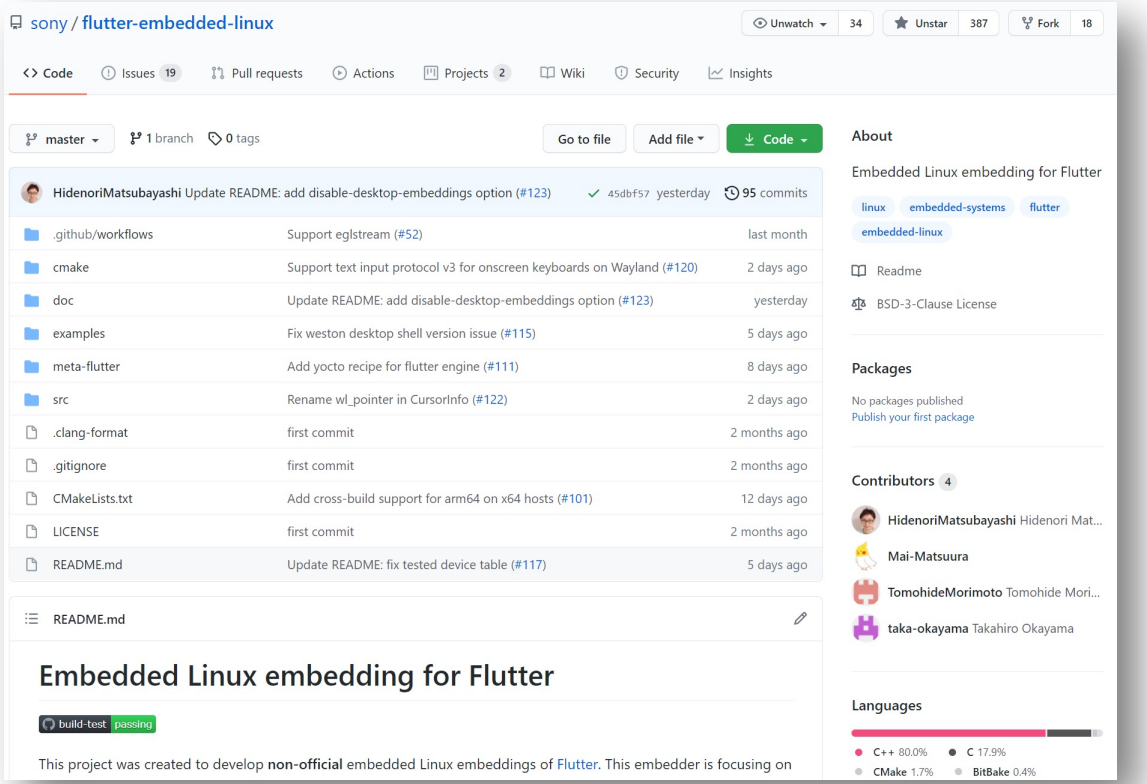
SAMSUNG Microsoft Surface

One area of particular growth over recent months has been the broadening of Flutter to an ever growing number of platforms and operating systems. At Flutter Engage, we announced that Toyota is bringing Flutter to their next generation vehicle infotainment systems. And last month, Canonical shipped their first release of Ubuntu with integrated support for Flutter, with Snap integration and support for Wayland.

Two new partners demonstrate this ever-growing ecosystem. Samsung is porting Flutter to Tizen, with an open source repository that others can also contribute to. And Sony is leading the effort to deliver a solution for embedded Linux.

[Announcing Flutter 2.2 at Google I/O 2021 | by Tim Sneath | Flutter | May, 2021 | Medium]

Flutter for Embedded Linux :
https://github.com/sony/flutter-embedded-linux



Flutter for embedded use - Samsung

More than a Google project

While Google continues to be the primary contributor to the Flutter project, we're delighted to see the growth of the broader ecosystem around Flutter.

TOYOTA CANONICAL SONY

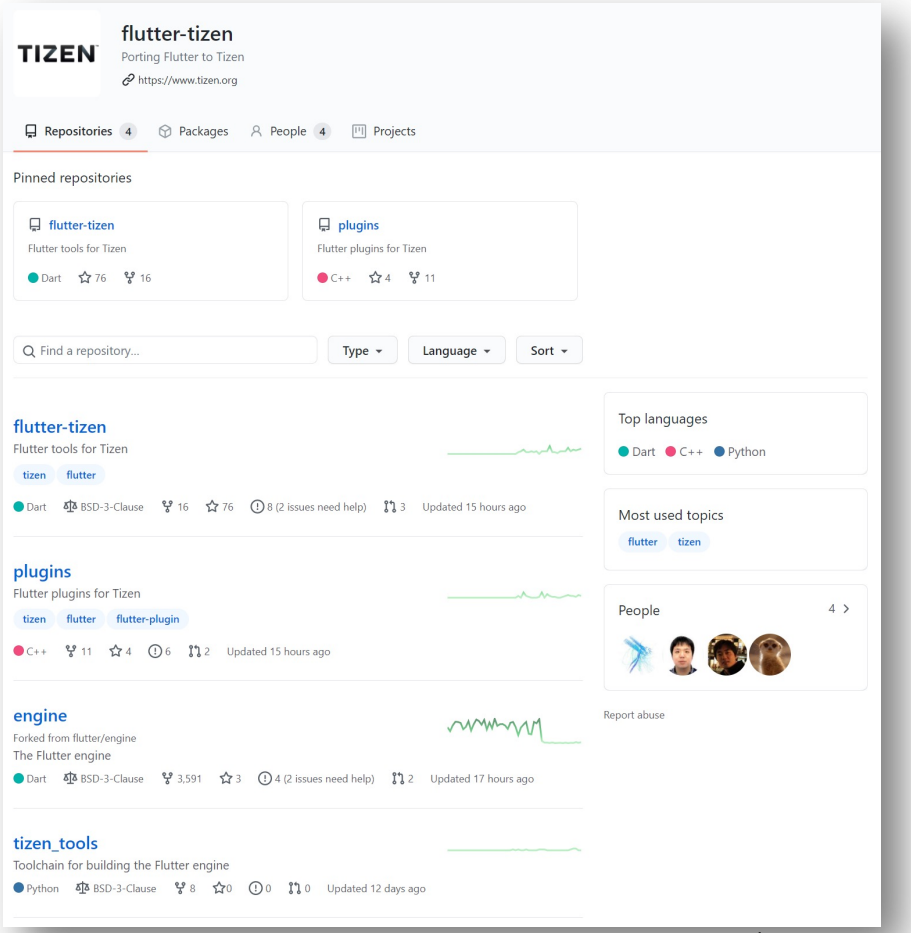
SAMSUNG Microsoft Surface

One area of particular growth over recent months has been the broadening of Flutter to an ever growing number of platforms and operating systems. At Flutter Engage, we announced that Toyota is bringing Flutter to their next generation vehicle infotainment systems. And last month, Canonical shipped their first release of Ubuntu with integrated support for Flutter, with Snap integration and support for Wayland.

Two new partners demonstrate this ever-growing ecosystem. Samsung is porting Flutter to Tizen, with an open source repository that others can also contribute to. And Sony is leading the effort to deliver a solution for embedded Linux.

[Announcing Flutter 2.2 at Google I/O 2021 | by Tim Sneath | Flutter | May, 2021 | Medium]

Tizen for Flutter :
https://github.com/flutter-tizen



SONY

Flutter for embedded use - Toyota

More than a Google project

While Google continues to be the primary contributor to the Flutter project, we're delighted to see the growth of the broader ecosystem around Flutter.

TOYOTA CANONICAL SONY

SAMSUNG Microsoft Surface

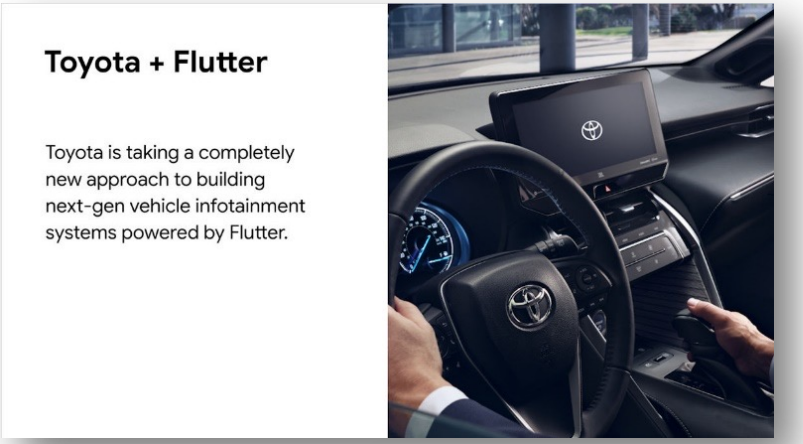
One area of particular growth over recent months has been the broadening of Flutter to an ever growing number of platforms and operating systems. At Flutter Engage, we announced that Toyota is bringing Flutter to their next generation vehicle infotainment systems. And last month, Canonical shipped their first release of Ubuntu with integrated support for Flutter, with Snap integration and support for Wayland.

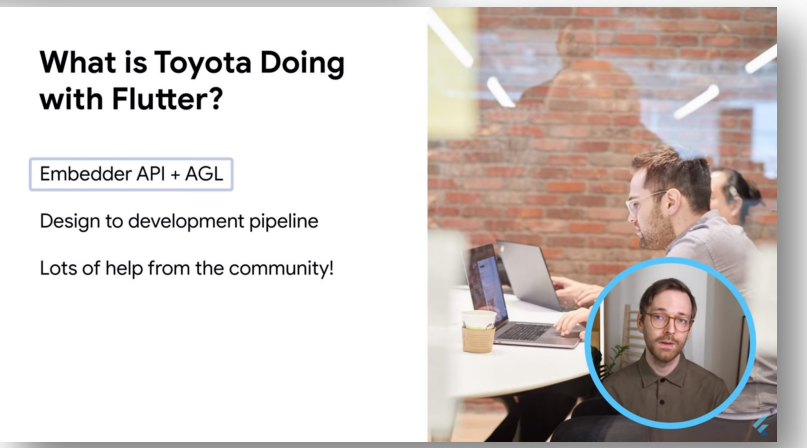
Two new partners demonstrate this ever-growing ecosystem. Samsung is porting Flutter to Tizen, with an open source repository that others can also contribute to. And Sony is leading the effort to deliver a solution for embedded Linux.

Announcing Flutter 2.2 at Google I/O 2021 | by Tim Sneath | Flutter | May, 2021 | Medium

AGL (Toyota's in-vehicle use) :

Keynote (Flutter Engage) – YouTube





Agenda

1. **Overview of modern Linux graphics**
   - Window Manager: X11, Wayland
   - Graphics API: Open GL ES, EGL
   - Framebuffer API: GBM, EGLStream
   - Kernel module: DRM, KMS

2. **Flutter and its architectural overview**
   - Flutter overview
   - Flutter internal architecture
   - Arm64 for Flutter Linux support
   - Bindings to native code

   etc.

3. **Introduction to “Flutter for embedded Linux”**
   - Overview & Objective
   - Features
   - Future works

   etc.

Agenda

1. **Overview of modern Linux graphics**
   - Window Manager: X11, Wayland
   - Graphics API: Open GL ES, EGL
   - Framebuffer API: GBM, EGLStream
   - Kernel module: DRM, KMS

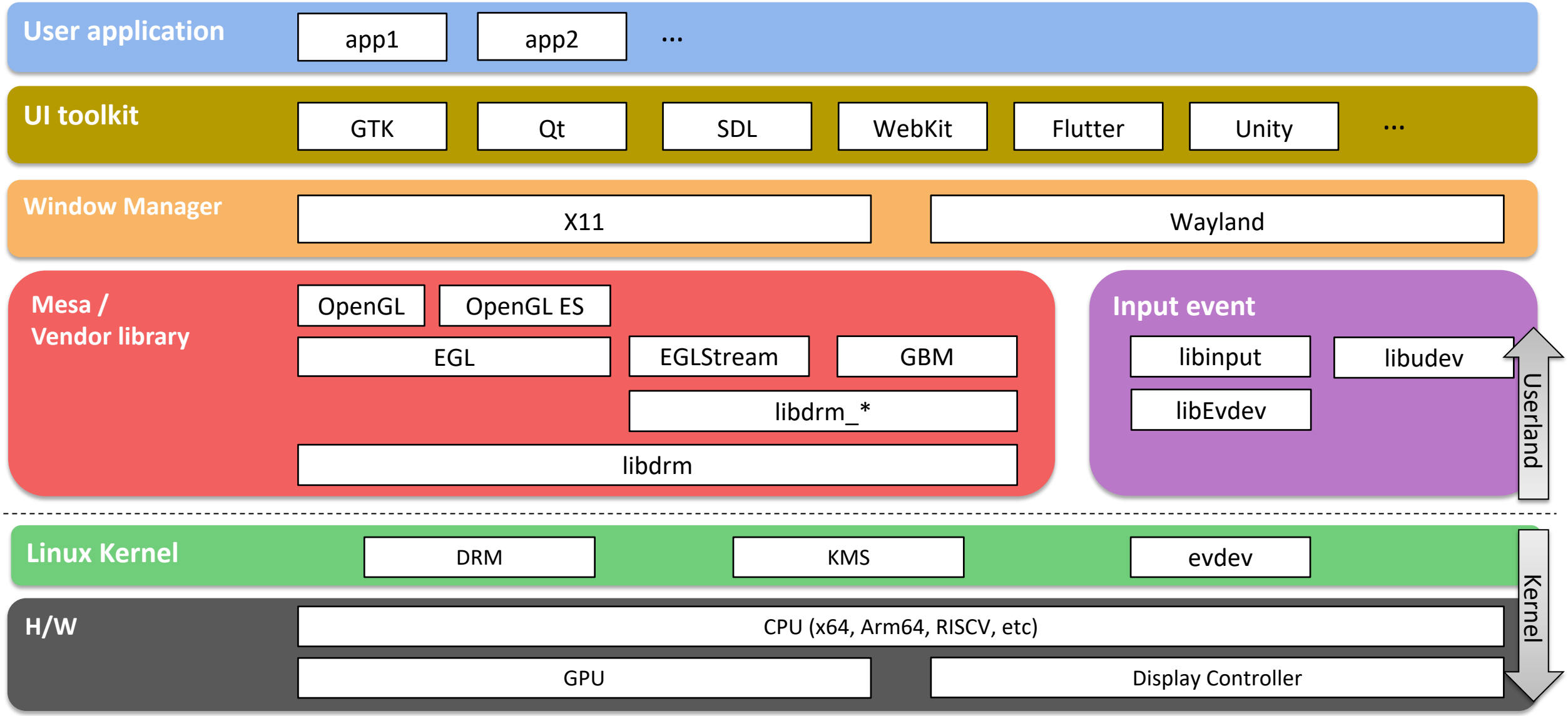
2. **Flutter and its architectural overview**
   - Flutter overview
   - Flutter internal architecture
   - Arm64 for Flutter Linux support
   - Bindings to native code

   etc.

3. **Introduction to “Flutter for embedded Linux”**
   - Overview & Objective
   - Features
   - Future works

   etc.

Overview of modern Linux graphics stack

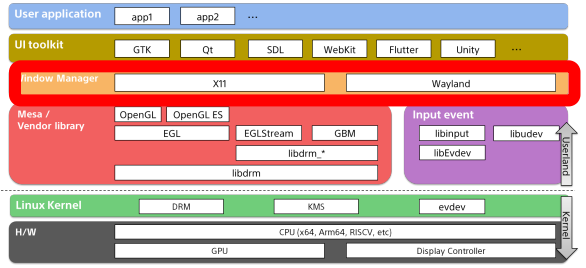


SONY

UI toolkit examples (Except for toolkits that only focused on Android, iOS, or desktops)

Category	Name	Software License	Main Maintainer
Web-based	Electron	MIT License	GitHub
	NW.js	MIT License	Intel
	Chromium	BSD 3-Clause	Google
	WebKit	LGPL, BSD	Apple
	Gecko	Mozilla Public License 2.0	Mozilla
Desktop-based	GTK	LGPL v2.1+	GNOME
	Qt	Commercial License (or GPL/LGPL v3.0)	Qt Company
	Mono	MIT, BSD, GPL etc.	Microsoft (Xamarin)
	SDL	zlib License	- (OSS Community)
	Kivy	MIT License	- (OSS Community)
	wxWidgets	wxWindows License	- (OSS Community)
	openFrameworks	MIT License	- (OSS Community)
Mobile-based	Flutter	BSD 3-Clause	Google
	Kotlin Multiplatform	Apache License 2.0	Jet Brains
Game-based	Unreal Engine	Commercial License (depends on sales)	Epic Games
	Unity	Commercial License (depends on sales)	Unity

Window Manager: X11, Wayland

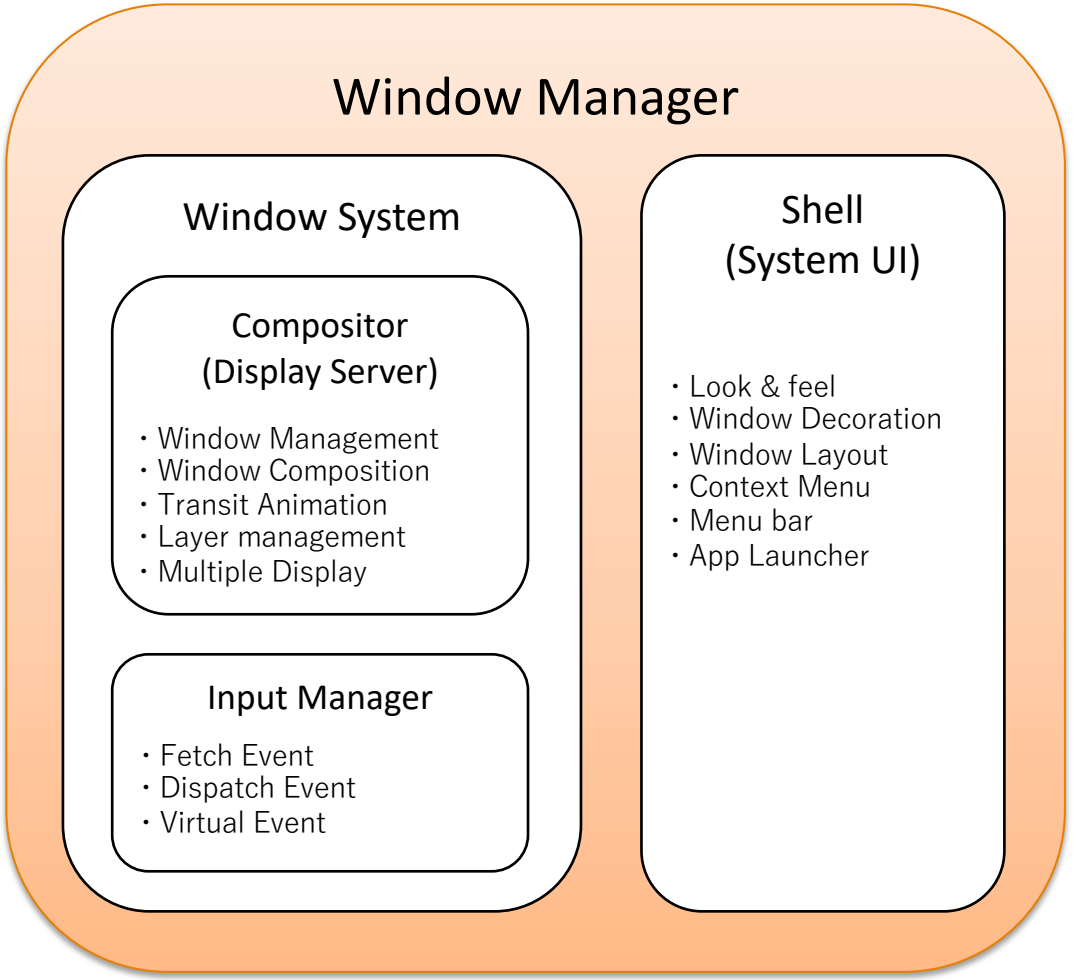


- **Window Manager**

  Window manager is system software that controls the placement and appearance of windows within a windowing system

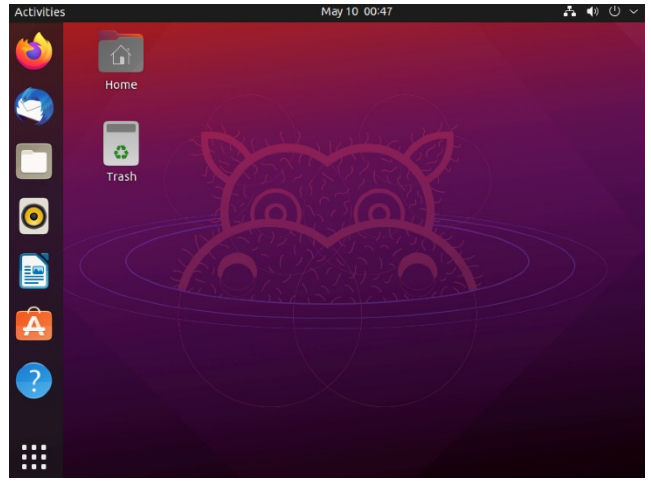
- **X11 (X Window System)**
  - https://www.x.org/wiki/
  - X Window System (X11, or simply X) is a windowing system for bitmap displays, common on Unix-like operating systems

- **Wayland**
  - https://wayland.freedesktop.org/
  - Next-generation window manager that will replace X11
  - Wayland is a communication protocol that specifies the communication between a display server and its clients



Wayland

- Ubuntu 21 enabled Wayland graphics by default
- Wayland is a lightweight than X11
  - Wayland just requires far fewer libraries than X11
- Board Support Package (BSP)
  - Many SoC vendors support Wayland (Weston)



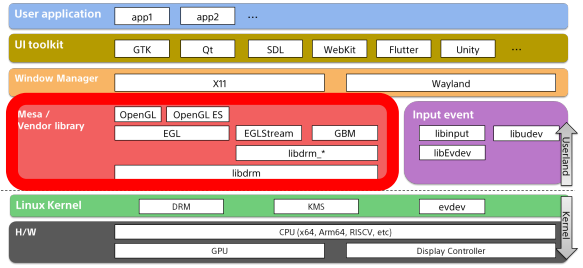
Ubuntu 21.04 is here | Ubuntu

Vendor	SoC/Board/ Platform	Window Manager		Notes
		X11	Wayland	
NXP	i.MX 8M	×	○	X11 isn't supported after i.MX 6
Xilinx	Zynq	○	○	Peta Linux supports Wayland from 2019.2
NVIDIA	Jetson	○	○	-
Qualcomm	RB5	○	○	SoC: SD865
Raspberry Pi4	Broadcom BCM2711	○	△	-

○ ・・・ Official Support
△ ・・・ Unofficial (3rd party)
× ・・・ Not supported

SONY

Graphics library



- **OpenGL, OpenGL ES**

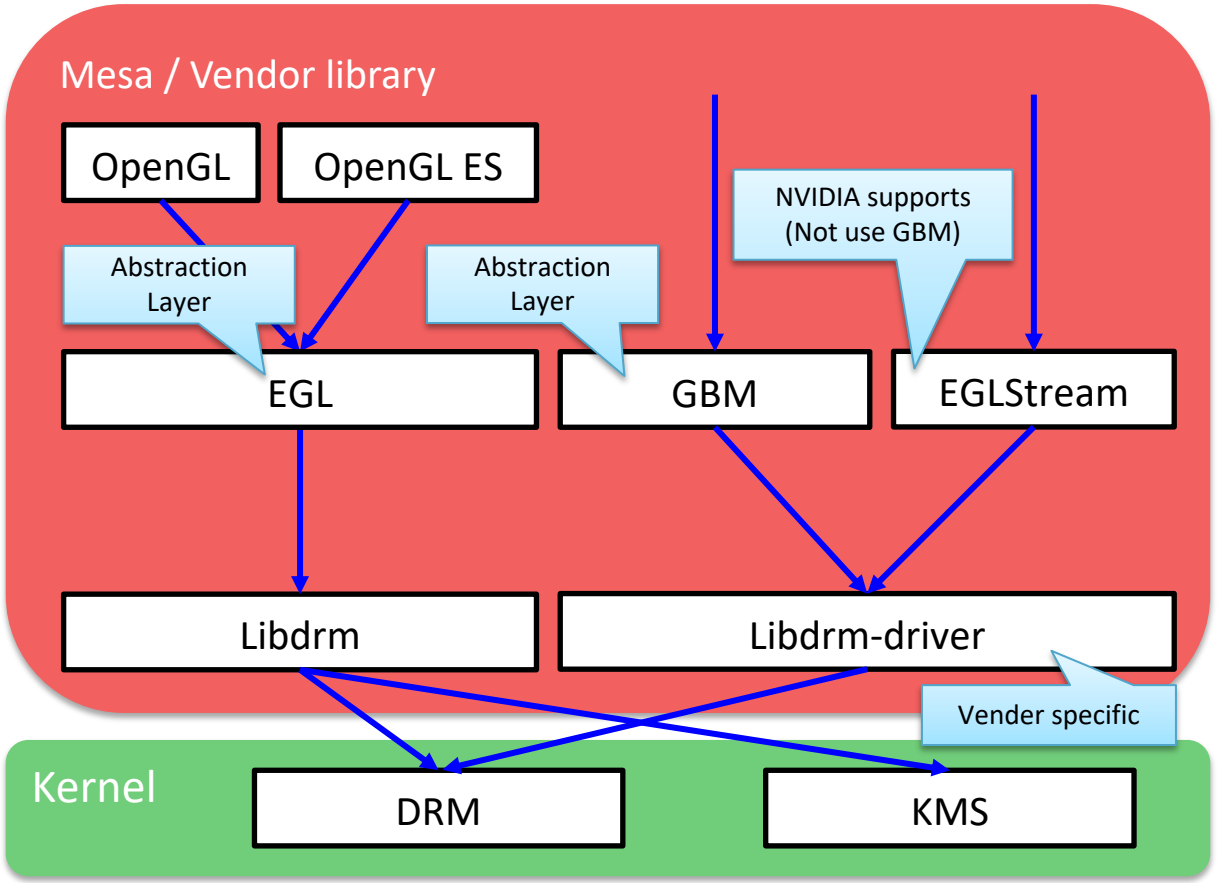
  OpenGL (Open Graphics Library) is a cross-language, cross-platform application API for rendering 2D and 3D vector graphics

- **EGL**

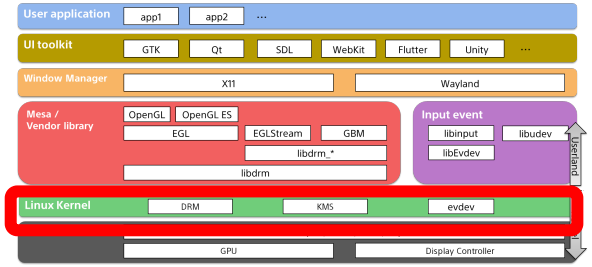
  An interface between Khronos rendering APIs (such as OpenGL, OpenGL ES or OpenVG) and the underlying native platform windowing system

- **Generic Buffer Management (GBM)**
  - An abstraction of the graphics driver specific buffer management APIs
  - Allocating buffers for graphics rendering

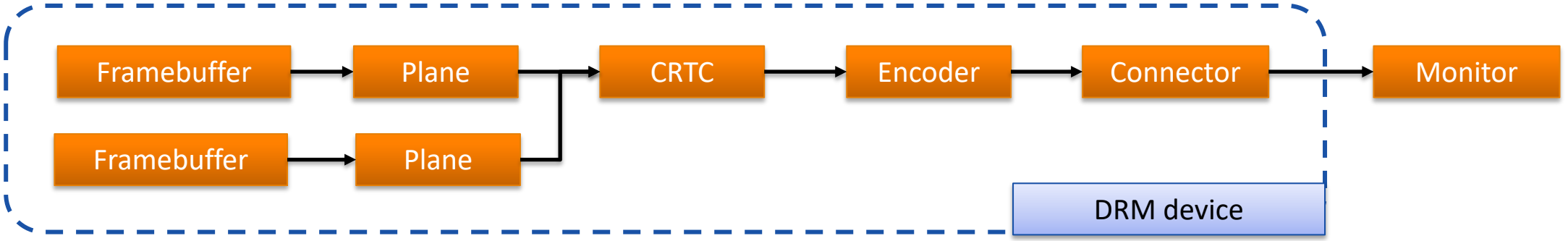
- **EGLStream**
  - EGLStream is a mechanism that efficiently transfers a sequence of image frames from one API to another
  - Nvidia only supports EGLStream



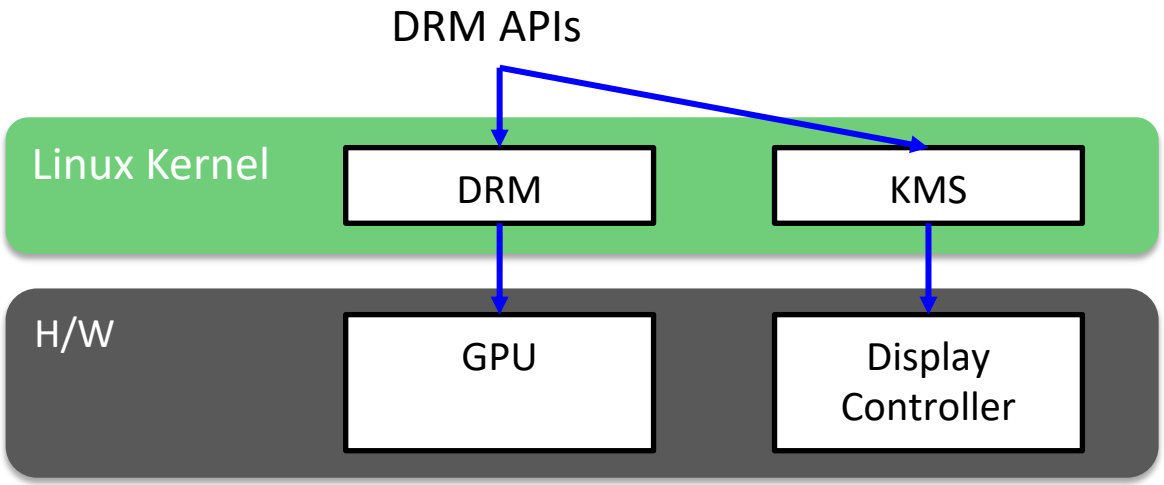
Kernel module: DRM / KMS



See: https://dri.freedesktop.org/docs/drm/gpu/index.html



- **Direct Rendering Manager (DRM)**
  - One of Linux kernel module (/dev/dri/card0, etc.)
  - Draw graphics to frame buffer directly
  - Authentication
- **Kernel Mode Setting (KMS)**
  - Display settings such as resolution and color depth



Agenda

1. Overview of modern Linux graphics
   - Window Manager: X11, Wayland
   - Graphics API: Open GL ES, EGL
   - Framebuffer API: GBM, EGLStream
   - Kernel module: DRM, KMS

2. **Flutter and its architectural overview**
   - Flutter overview
   - Flutter internal architecture
   - Arm64 for Flutter Linux support
   - Bindings to native code

   etc.

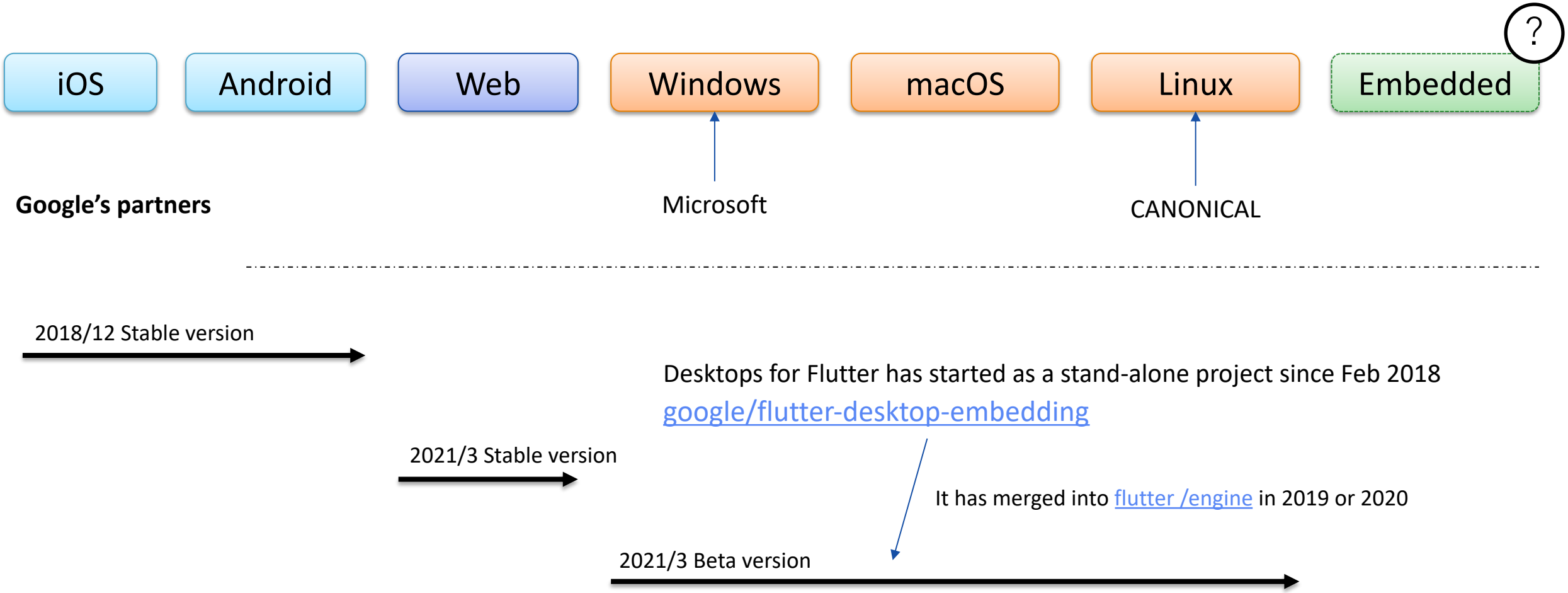
3. Introduction to “Flutter for embedded Linux”
   - Overview & Objective
   - Features
   - Future works

   etc.

What is Flutter?

- https://flutter.dev/
- Flutter is Google's UI toolkit for building beautiful, natively compiled applications
- Released in 2017
- Original rendering engine using Skia
- Programing language: Dart
- Supported platforms from a single source code
  - Mobile (Android/iOS)
  - Web
  - Desktops (Linux/Windows/macOS): β version
    - ✓ Official version will come within 2021?

Flutter Roadmap (History)

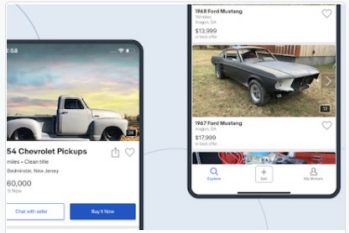


SONY

Pros & Cons of Flutter

- **Pros**
  - Easy to create beautiful UI
  - Create a prototype quickly using Adobe XD, etc.
  - A lot of 3rd libraries (plugins), documents, and information
  - Native compiled applications, strongly debug function like hot-reload
  - Cross-platform
  - Package management system and test tools (Dart SDK)
  - Support of major IDEs like VS Code
  - Active community
  - Embedded use cases
    - ✓ Custom embedder API-layer for specific platforms
    - ✓ Flutter Engine requires just fewer library dependencies
    - ✓ Easy access to other native codes and hardware resources
    - ✓ BSD 3-Clause licensed software
- **Cons**
  - Need to study Dart
  - Platform-specific features like media players need to be implemented natively
  - Lighter than WebView, but might be heavier than native apps (We need to benchmark)

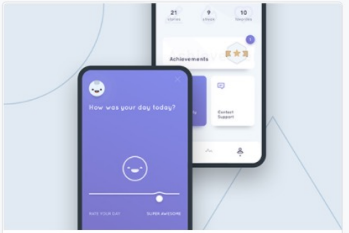
Apps made with Flutter



ebay

The eBay Motors app is a powerful tool for browsing, buying, and selling vehicles directly from consumers' phones.

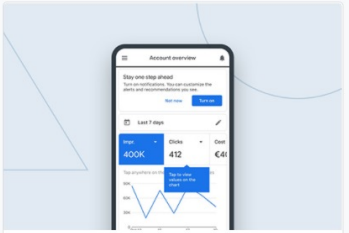
Learn more Download



Reflectly

An award-winning mindfulness app built with Flutter. It was featured on the Apple App Store as 'app of the day'.

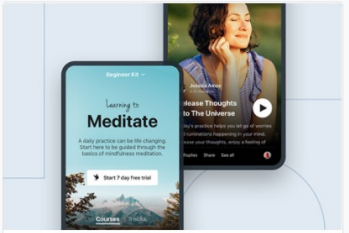
Learn more Download



Google Ads

The Google Ads app helps users manage, optimize and track ad campaigns all from the convenience of an android device.

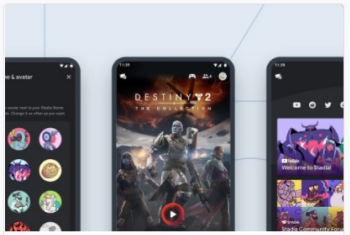
Learn more Download



Insight Timer

The Insight timer app used by 9M+ people provides access to the largest free library of guided meditations on earth.

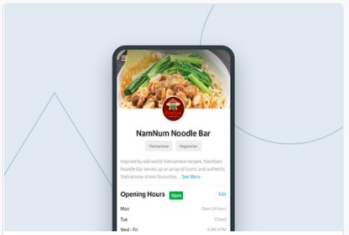
Learn more Download



STADIA

Google's Stadia app is built using Flutter for both iOS and Android.

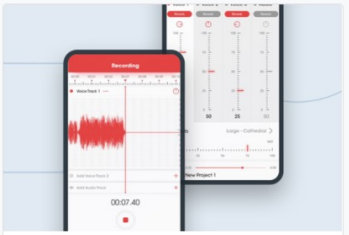
Learn more Download



Grab

Flutter helped Grab build the merchant app for its fast-growing food delivery business.

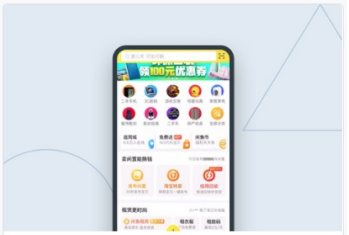
Download



Abbey Road Studios

Abbey Road Studio's Topline app continues the studio's tradition of innovation.

Learn more Download



Alibaba Group

Flutter helped bring a new app for the world's biggest online marketplace to life.

Learn more Download

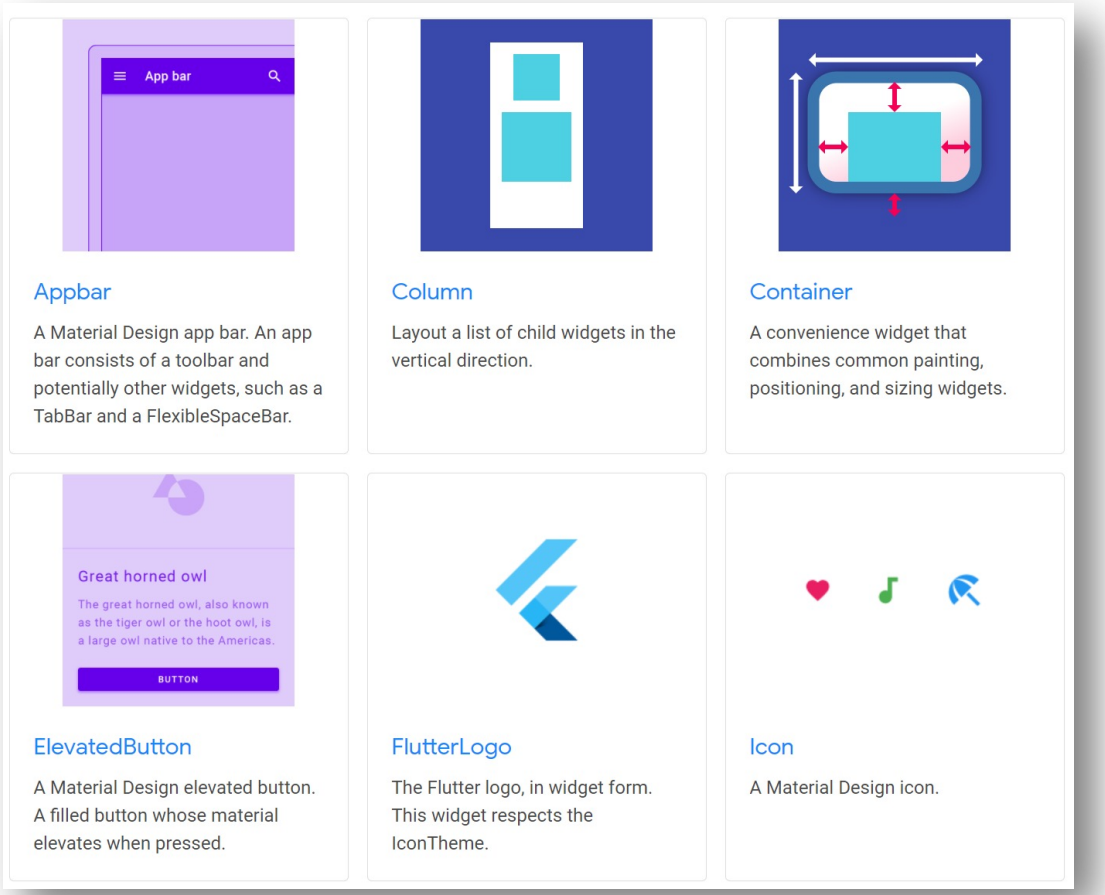
https://flutter.dev/showcase

SONY

Flutter Widget

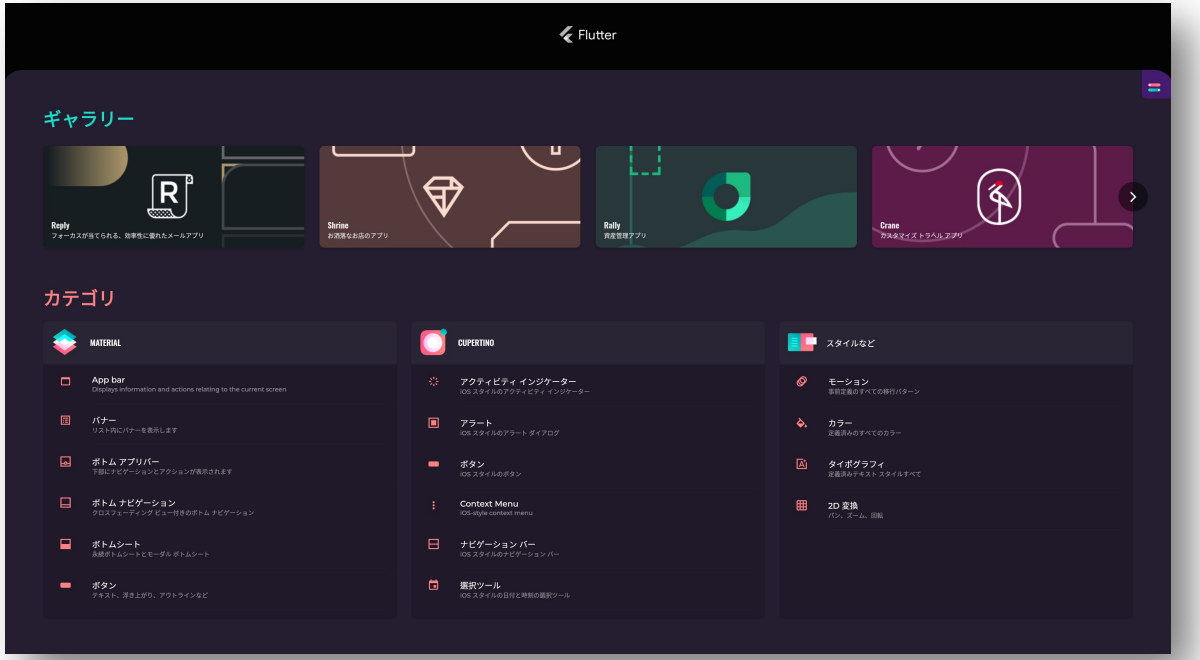
- In Flutter, Widget is the UI component to declare and construct UI

- See: [Widget catalog - Flutter]

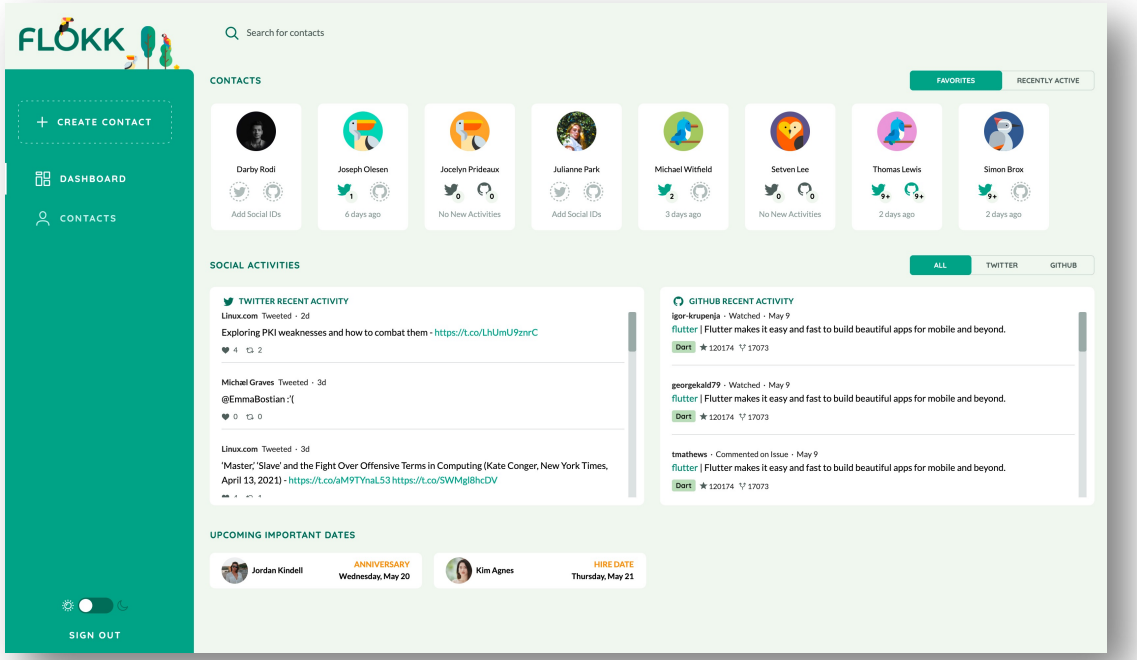


Famous Flutter apps examples

- https://github.com/flutter/gallery
  Flutter Gallery is an official resource and example app to help developers evaluate and use Flutter
- https://github.com/gskinnerTeam/flokk
- https://github.com/gskinnerTeam/flutter-folio



https://gallery.flutter.dev/#/



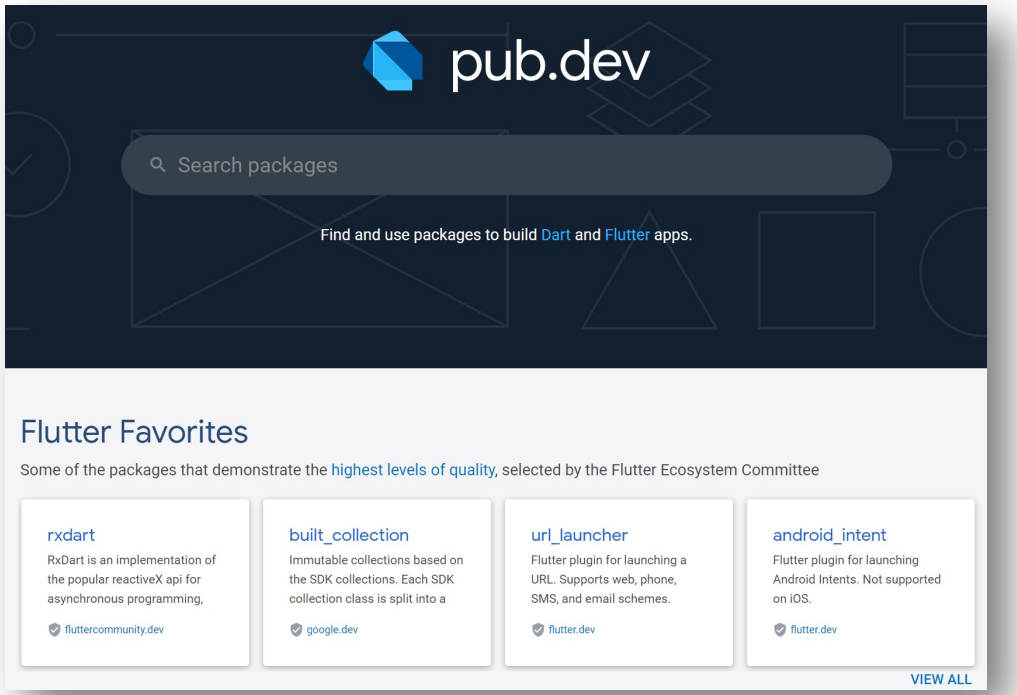
https://flutter.gskinner.com/flokk/

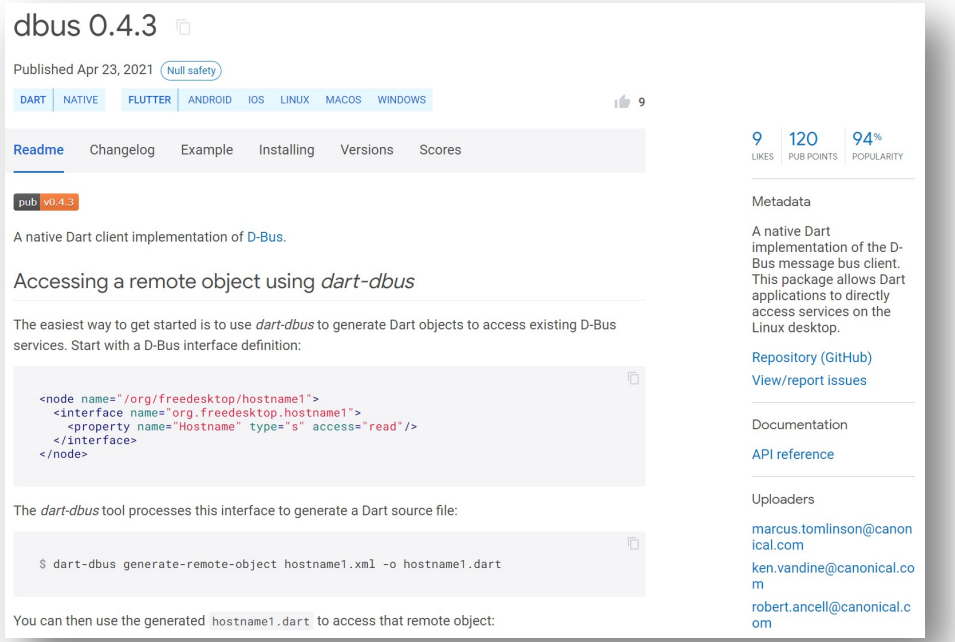
SONY

Flutter plugin

- A lot of official and 3rd party libraries (Flutter plugins) on pub.dev
- Flutter SDK provides a package management function to make developers install Flutter plugins easily
- Plugins

  Flutter is just the UI toolkit. It means not includes platform native features like a media player. You need to implement it yourself or use 3rd party one





Dart programming language

- Dart is been developing by Google from 2011 to be used to build server and desktop applications. Development started with the goal of replacing JavaScript

- Dart 1.0 release: Nov 2013
  - Latest stable version: 2.13.0

- Null Safety support

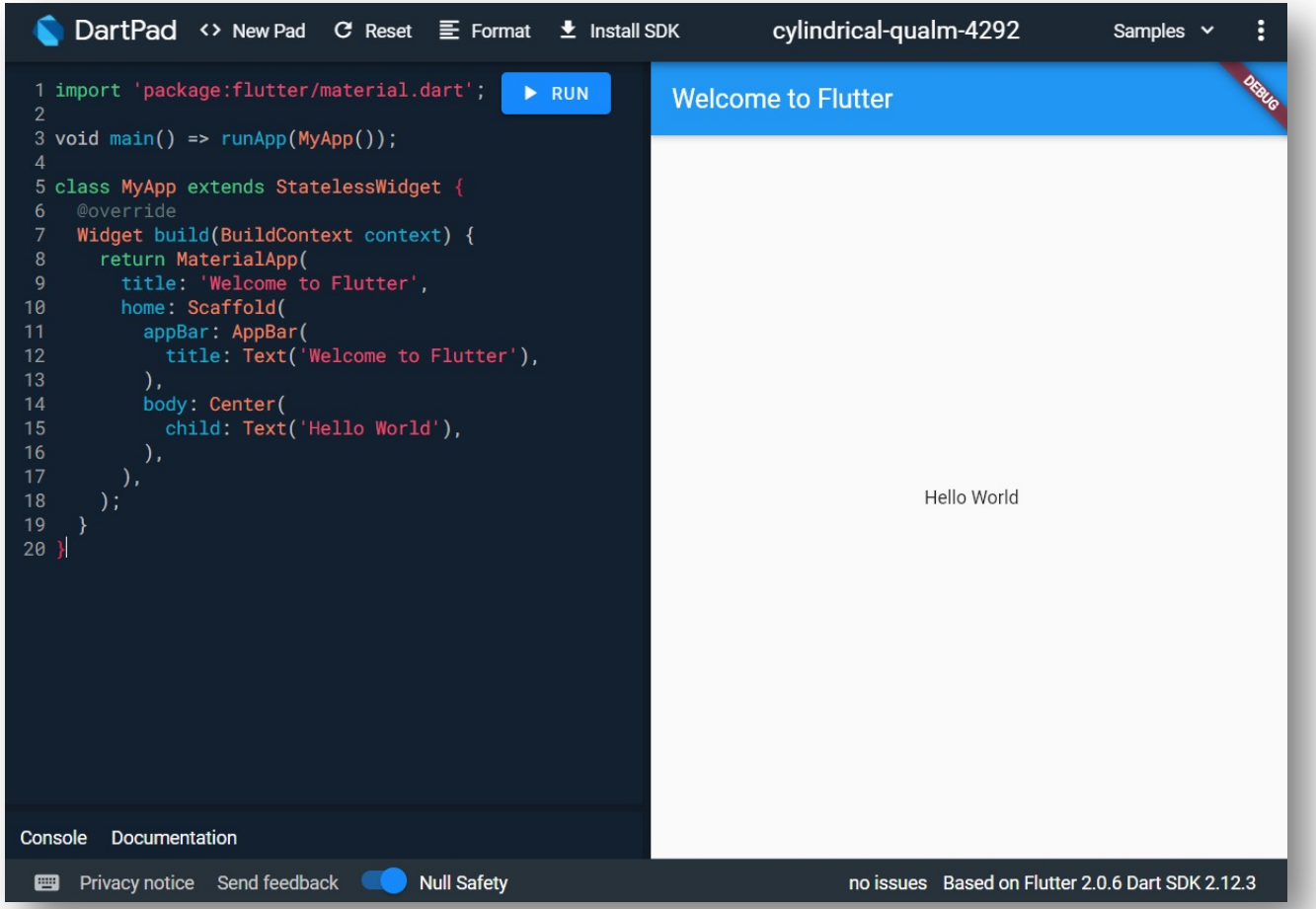
- Dart language specification is quite close to JavaScript, and it is said it is a language with very low learning cost for developers who are familiar with JavaScript, Java, or C++

- Dart compiler
  - Dart to JavaScript
  - Dart to native (machine code JIT and AOT)

- Binding to native code / IPC
  - dart:ffi (FFI stands for foreign function interface)
  - Unix Domain Socket in dart:io
  - 3rd party library: gRPC, Dbus etc.

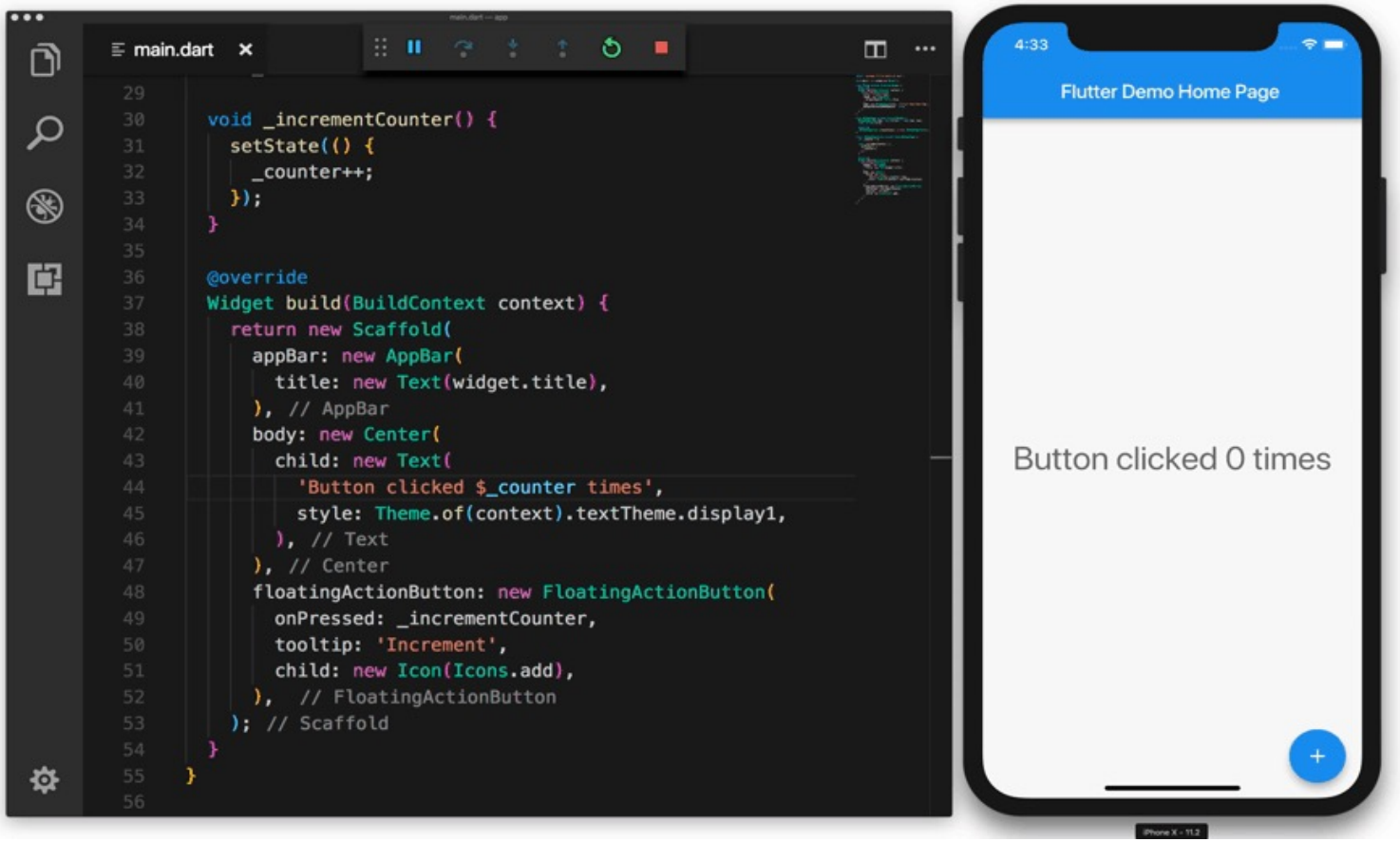
DartPad

- https://dartpad.dev/flutter
- DartPad is a free, open-source online editor to help developers learn about Dart and Flutter



Hot reload

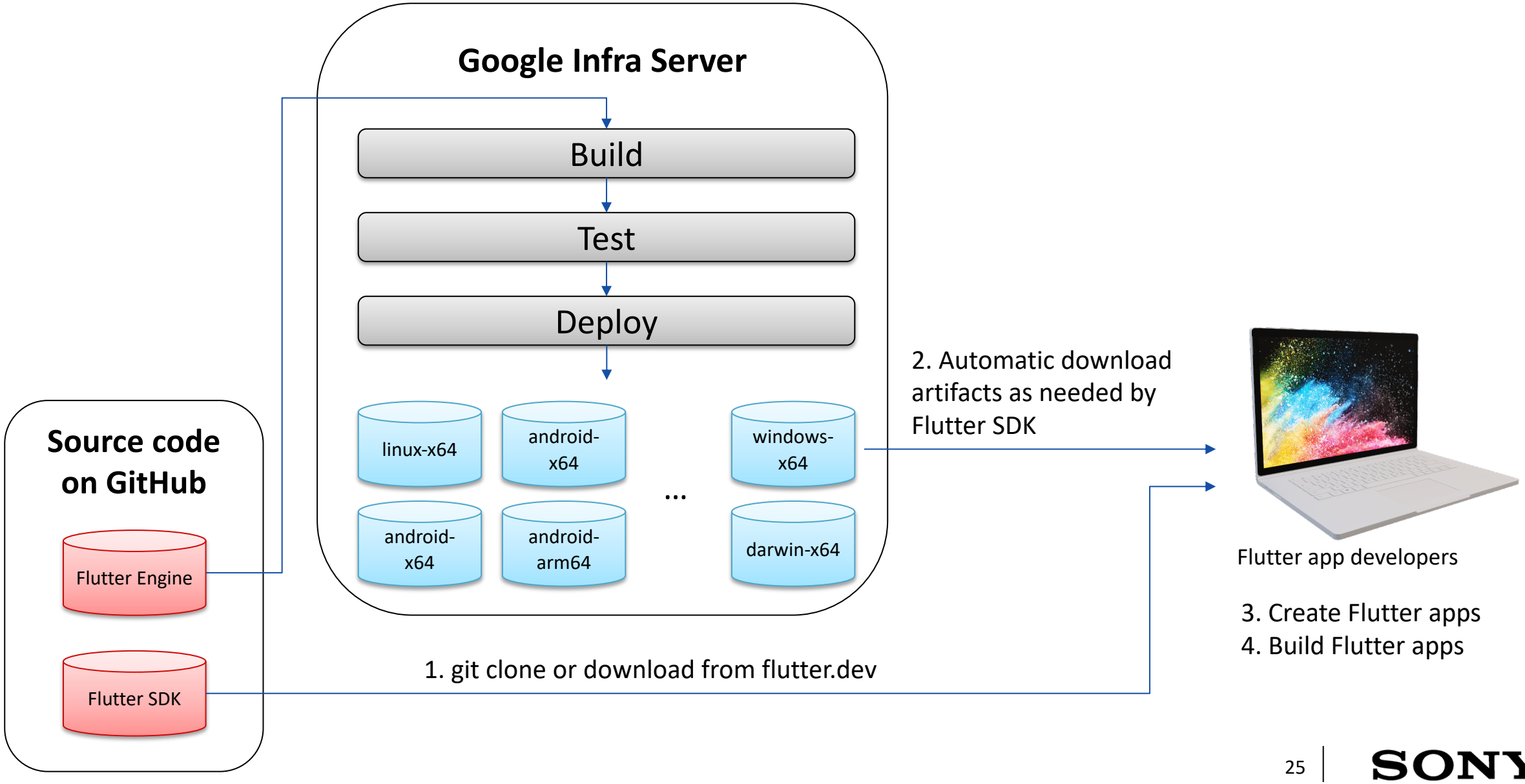
- Hot reload helps developers quickly and easily create UIs, and fix bugs
- Developers can inject updated source code files into the running Flutter apps without relaunch it



Excerpt from https://github.com/flutter/flutter

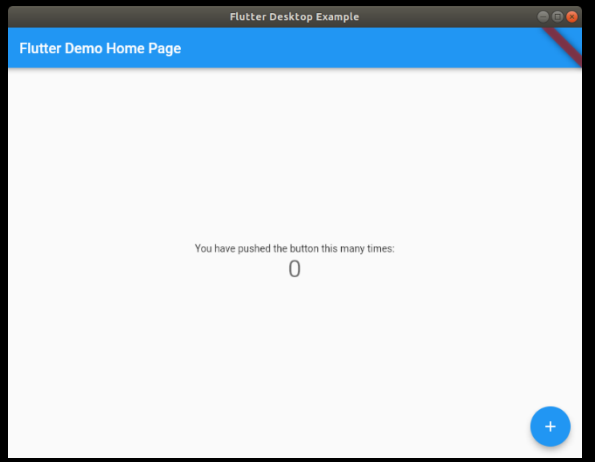
SONY

Flutter repo and development workflow



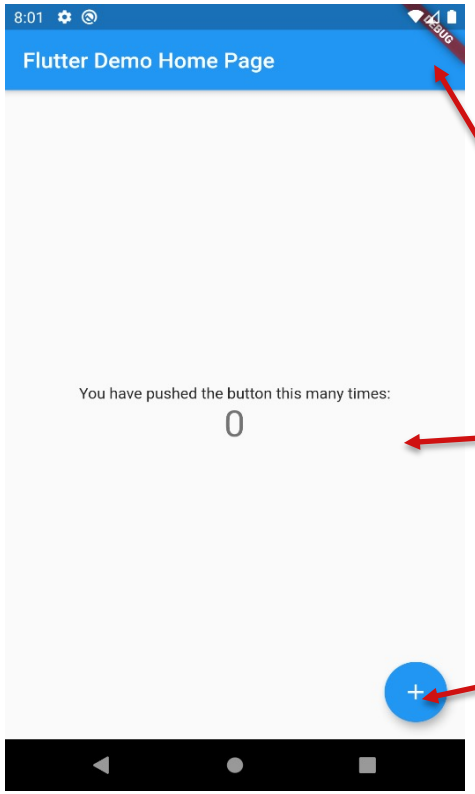
Quick start on Ubuntu Linux Hosts

```
# Install Flutter SDK
$ git clone https://github.com/flutter/flutter
$ sudo mv flutter /opt/
$ export PATH=$PATH:/opt/flutter/bin

# Install dependent packages
$ sudo apt install clang curl pkg-config ninja-build cmake libgtk-3-dev libblkid-dev liblzma-dev unzip

# Enable Flutter desktop for Linux
$ flutter config --enable-linux-desktop

# Run Flutter sample app
$ flutter create sample
$ cd sample
$ flutter run –d linux
```

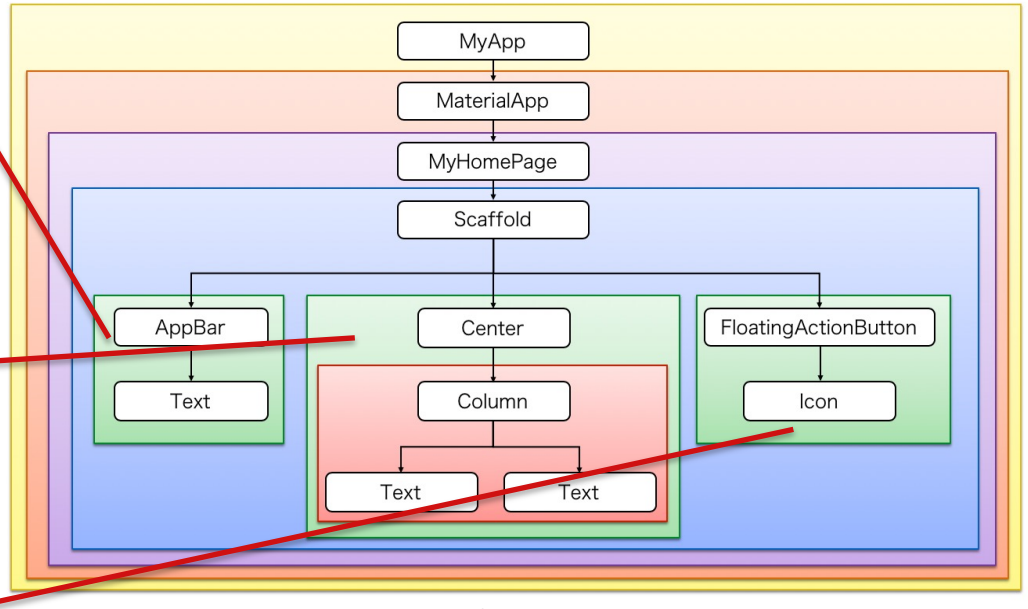


SONY

Flutter sample app & Widget tree



Sample app



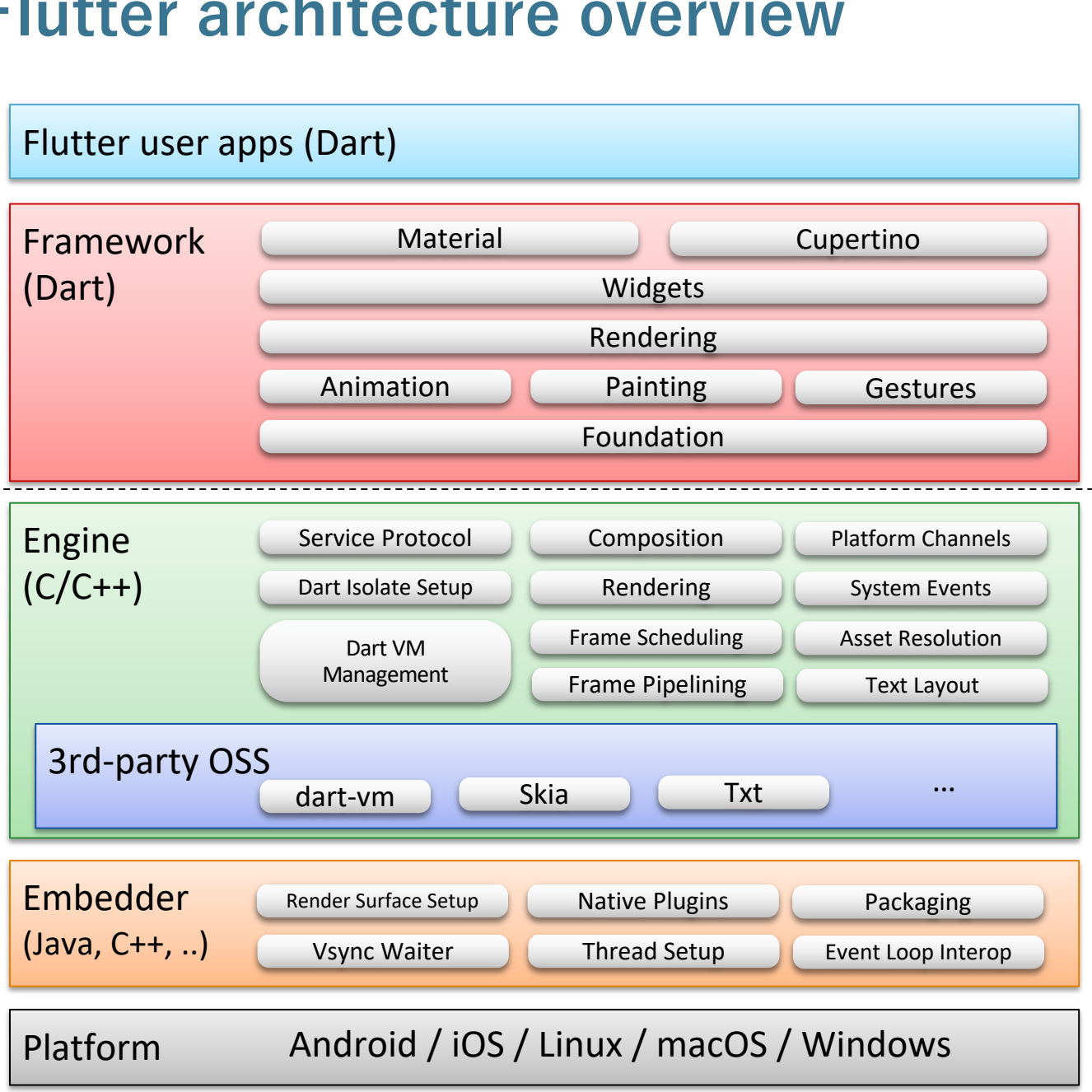
Widget tree

Source code

```dart
class _MyHomePageState extends State<MyHomePage> {
  int _counter = 0;

  void _incrementCounter() {
    setState(() {
      _counter++;
    });
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(widget.title),
      ),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[
            Text(
              'You have pushed the button this many times:',
            ),
            Text(
              '$_counter',
              style: Theme.of(context).textTheme.display1,
            ),
          ],
        ),
      ),
      floatingActionButton: FloatingActionButton(
        onPressed: _incrementCounter,
        tooltip: 'Increment',
        child: Icon(Icons.add),
      ),
    );
  }
}
```


SONY

Flutter architecture overview



- Written in Dart language
- Create UI by using widgets
- Works on Dart-VM

Framework

- Graphics Engine by being written in C/C++
- Provide graphics shell
- Manage Dart-VM
- Drawing by using Skia

etc.

Runtime

Porting layer for specific platforms

SONY

Embedder API

- https://github.com/flutter/flutter/wiki/Custom-Flutter-Engine-Embedders

- Flutter desktops (Linux / macOS / Windows) use this API
  - Android and iOS aren't using it. The API were created for Flutter desktops?

- Header file
  - https://github.com/flutter/engine/blob/master/shell/platform/embedder/embedder.h

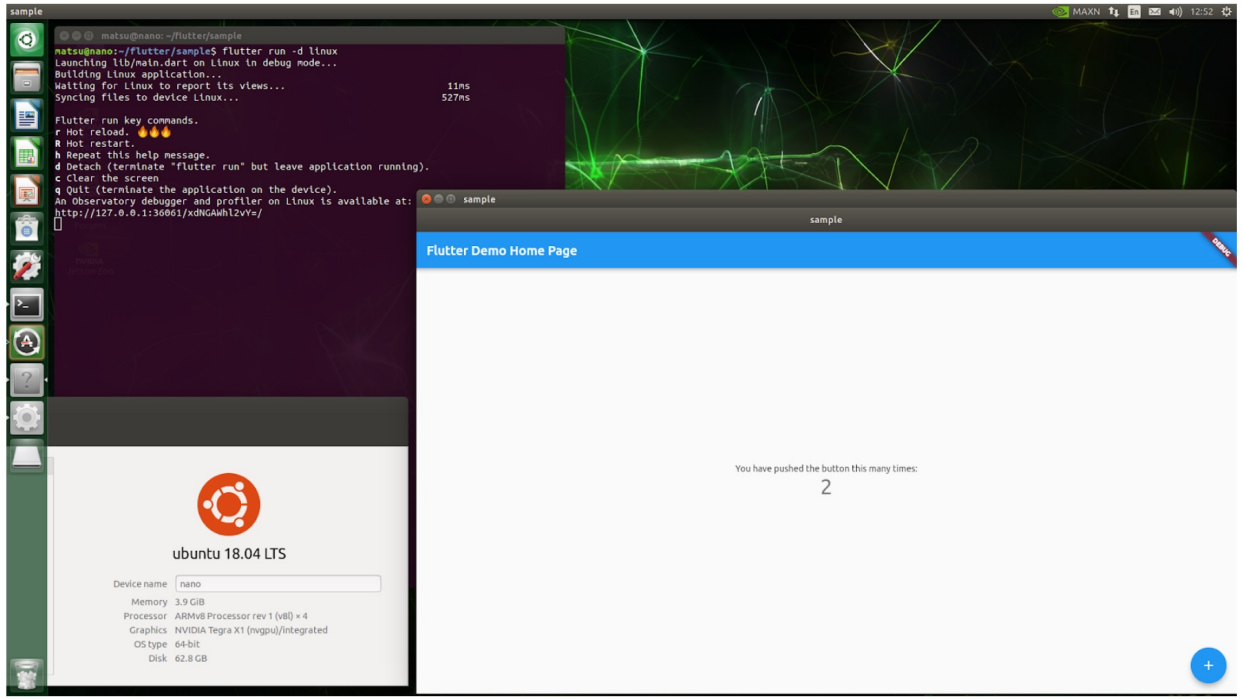
```
FLUTTER_EXPORT
FlutterEngineResult FlutterEngineRun(size_t version,
                                     const FlutterRendererConfig* config,
                                     const FlutterProjectArgs* args,
                                     void* user_data,
                                     FLUTTER_API_SYMBOL(FlutterEngine) *
                                         engine_out);
```

Arm64 Linux support (One of our contributions to Flutter)

[What's new in Flutter 2.2 | Flutter (medium.com)]

ARM64 Linux host support from Sony

Another excellent effort by a Flutter community member-at-large is from HidenoriMatsubayashi, a software engineer at Sony, who has contributed support for targeting ARM64 Linux. This PR enables you to build and run Flutter apps on ARM64 Linux machines.

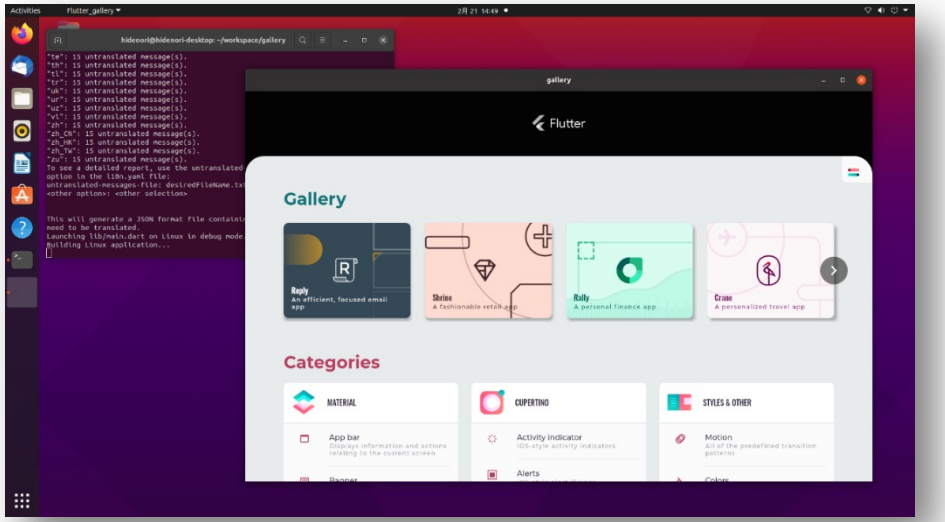


Your favorite Flutter app running on an ARM64 Linux machine

It's exciting to see the Flutter community bringing Flutter to places that the team at Google could never have imagined. Keep up the good work, HidenoriMatsubayashi!

- Currently, Flutter supports desktop for Arm64 Linux hosts. Also, Flutter SDK works on Arm64 Linux hosts

- Flutter works on general Arm64 devices such as Raspberry Pi4 and Jetson Nano



Raspberry Pi4

Jetson Nano

SONY

Binding to native code in Flutter

➢ Binding to native code

- Communication APIs are provided by Flutter
  - ✓ Method Channel
  - ✓ Event Channel
  - ✓ Basic Message Channel
- dart:ffi
  - ✓ Foreign Function Interface for interoperability with C programming language

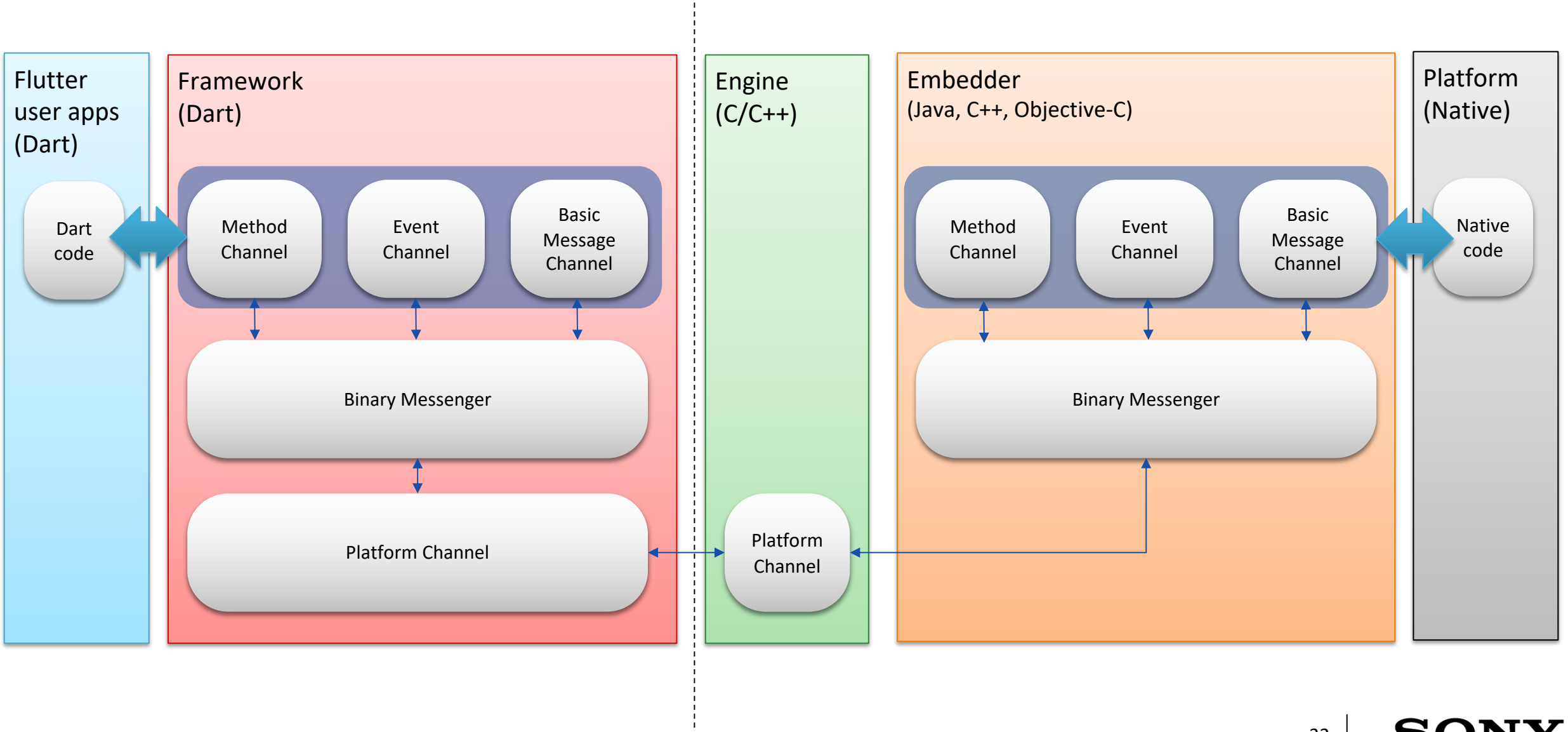
➢ Inter-process communication (IPC)

- Unix domain sockets in dart:io
- 3rd party library
  - ✓ A native Dart client implementation of D-Bus: https://github.com/canonical/dbus.dart
  - ✓ grpc-dart: https://github.com/grpc/grpc-dart

  …

SONY

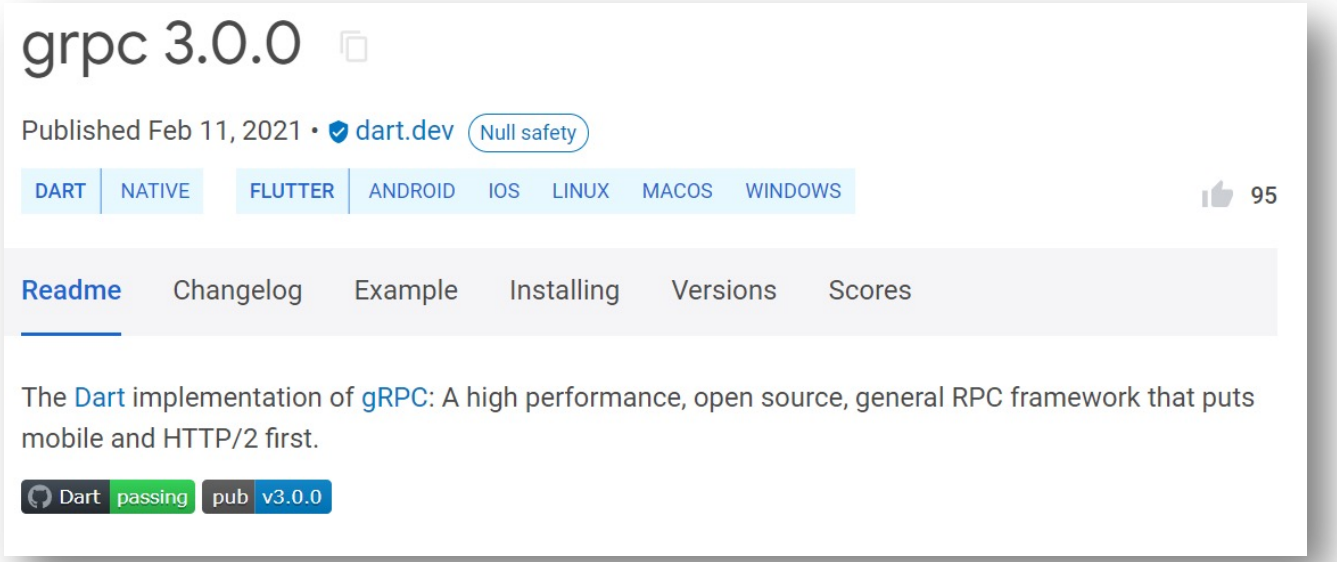
Communication APIs are provided by Flutter



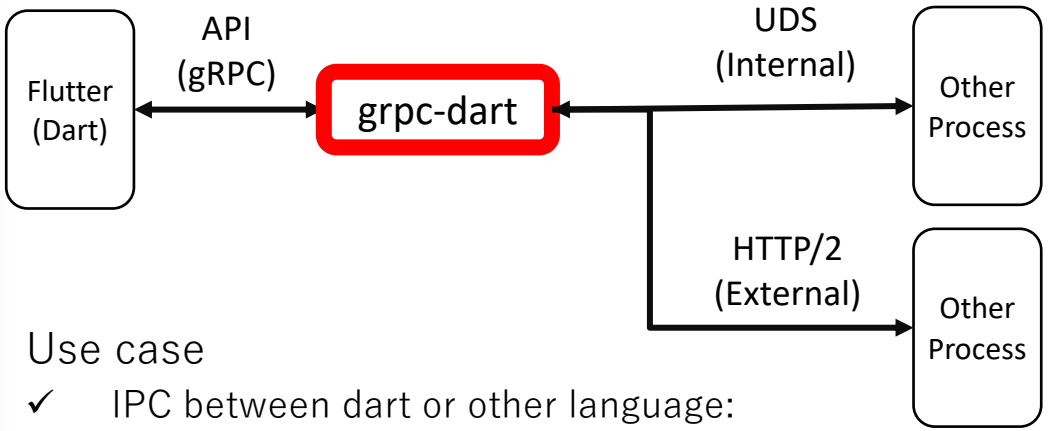
SONY

Flutter plugin: Unix domain socket support in grpc-dart

- grpc-dart is the gRPC library in Dart implementation
- We have contributed to add Unix Domain Socket support to grpc-dart



https://pub.dev/packages/grpc



Use case

- ✓ IPC between dart or other language: Unix domain socket
- ✓ External communication between the device and other devices: HTTP/2

Agenda

1. Overview of modern Linux graphics
   - Window Manager: X11, Wayland
   - Graphics API: Open GL ES, EGL
   - Framebuffer API: GBM, EGLStream
   - Kernel module: DRM, KMS

2. Flutter and its architectural overview
   - Flutter overview
   - Flutter internal architecture
   - Arm64 for Flutter Linux support
   - Bindings to native code

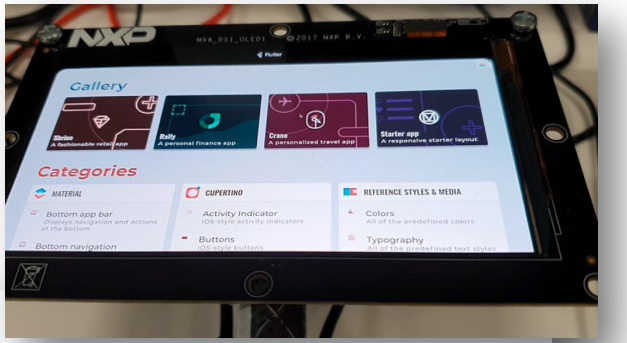
   etc.

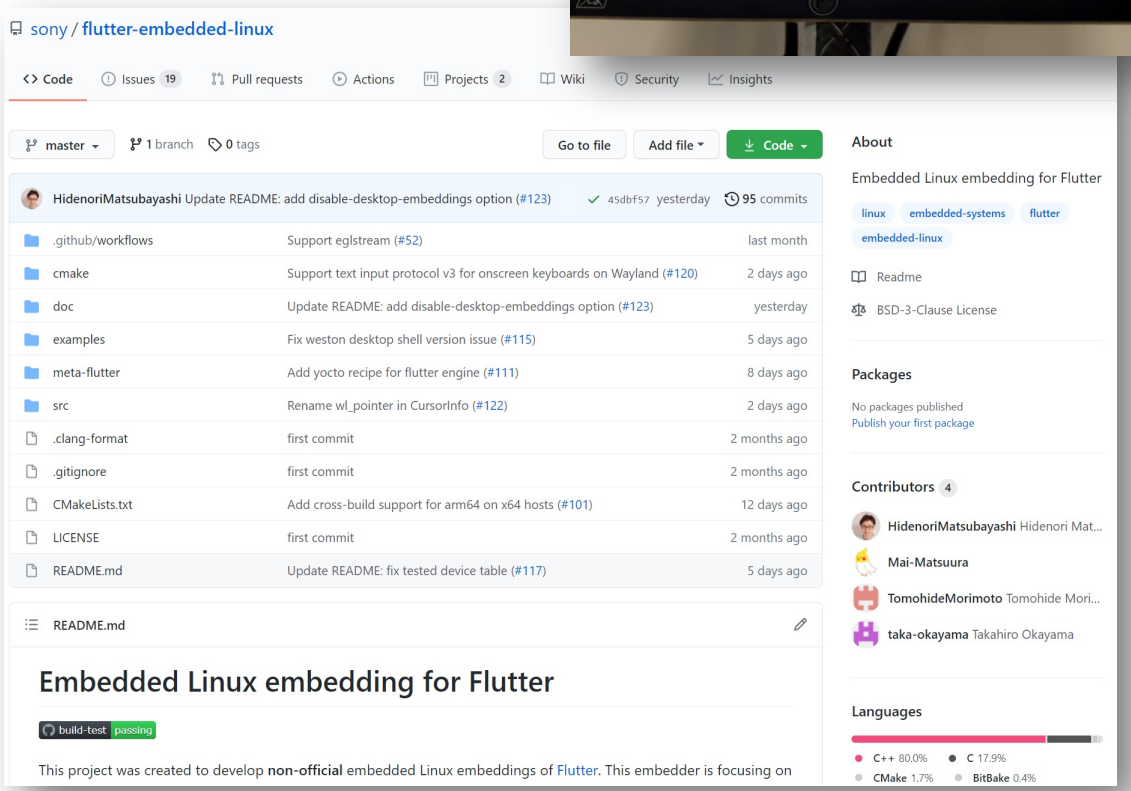
3. **Introduction to “Flutter for embedded Linux”**
   - Overview & Objective
   - Features
   - Future works

   etc.

Flutter for Embedded Linux

Open-source: https://github.com/sony/flutter-embedded-linux





- **Overview**
  - Porting Flutter to embedded Linux systems
- **Objective**
  - Use Flutter in embedded systems
- **Non-goal**
  - Not intended to replace the existing Flutter desktop for Linux in desktop use cases
  - Objective is just for Embedded Systems use cases

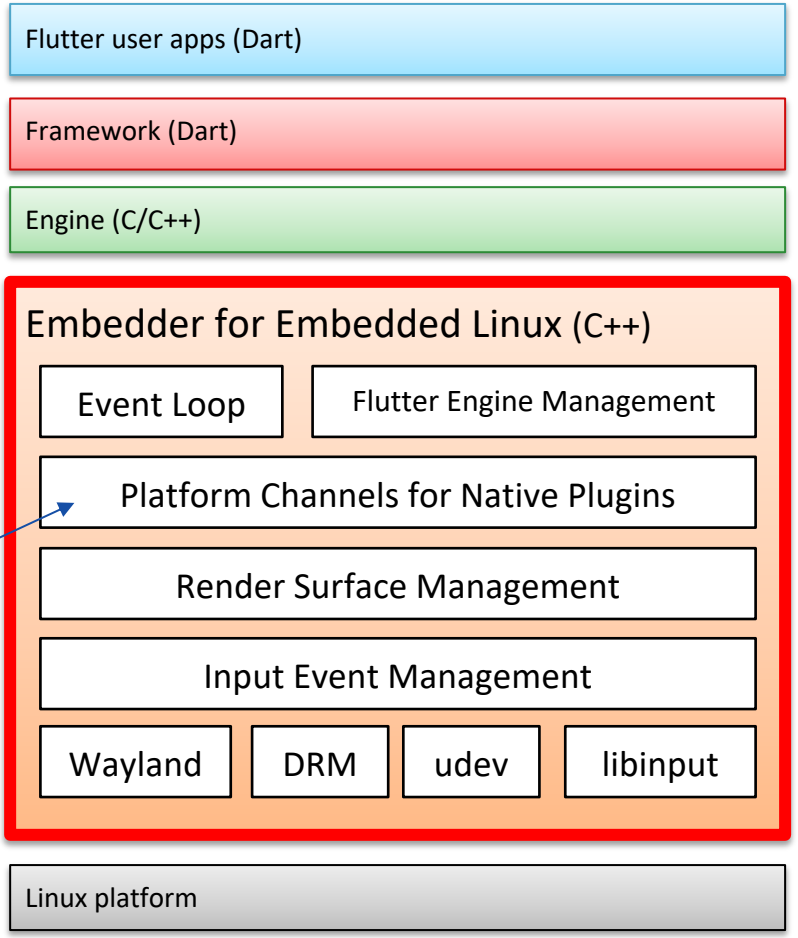
SONY

Features

- Flutter embedder optimized for embedded systems
- Both x64 architecture and Arm64 architecture support
- Lightweight than Flutter desktop for Linux (without X11 and GTK/GDK)
- Wayland, DRM (GBM or EGLStream) backends support
- Single full-screen or flexible-screen
- Keyboard, touch, mouse, clipboard support
- API compatibility with Flutter desktop for Windows
  - External texture plugin (texture composition in Flutter) for media player, etc.
  - Based on Flutter desktop for Windows

    Initially, it was created with full scratch, but we changed from the middle

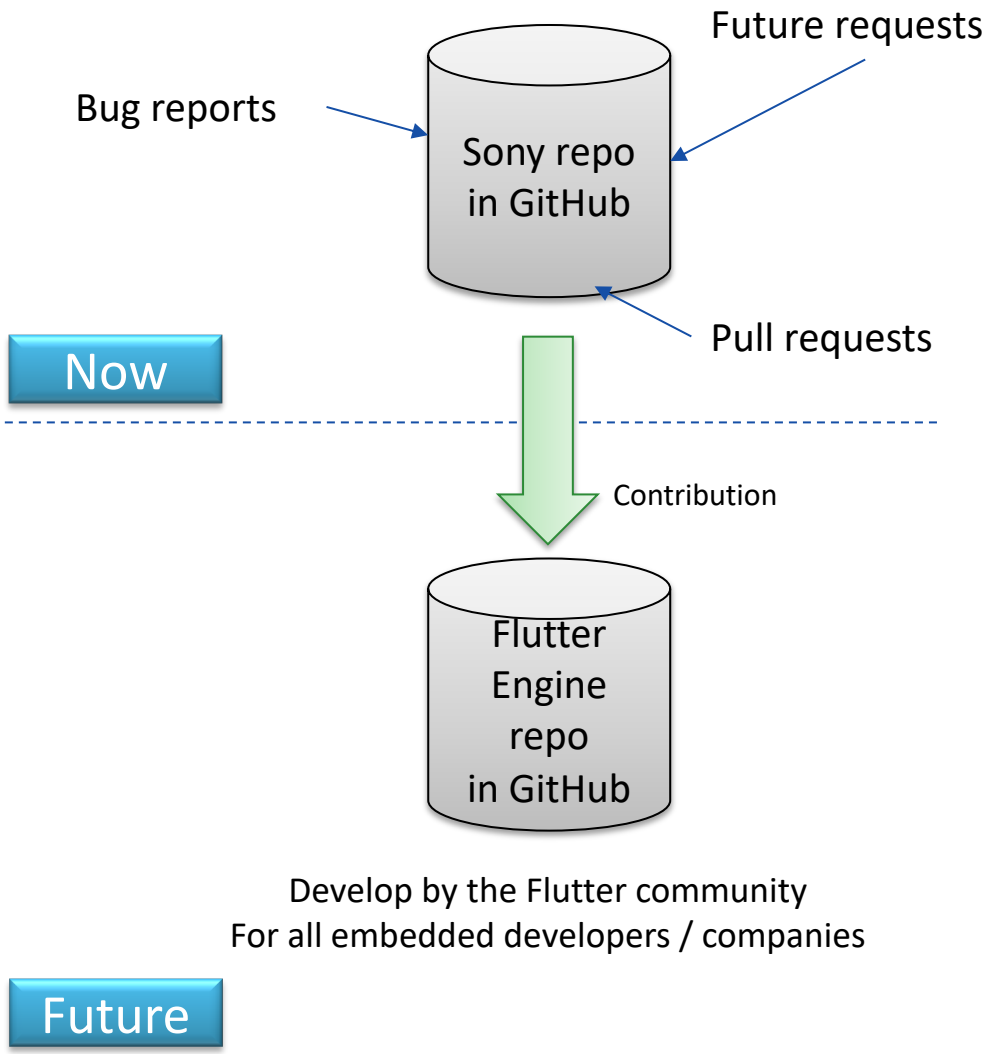
etc.



SONY

Objective of open-source

- We are looking for partners to develop together
  - It's difficult to cover all embedded specifications for Sony alone
  - Google Flutter team is closely us
- Contribution
  - Welcome all your contribution and feedbacks
    - ✓ If you want to send a PR, you need to accept our CLA
    - ✓ CLA is still under construction…
- Target the mainline
  - Propose and contribute this software to the mainline of Flutter Engine repo, which means we would like to add embedded systems support into Flutter for all embedded developers in the future



Why is this embedder necessary instead of “Flutter desktop for Linux”?

Requirements for embedded systems are not equal desktop one.

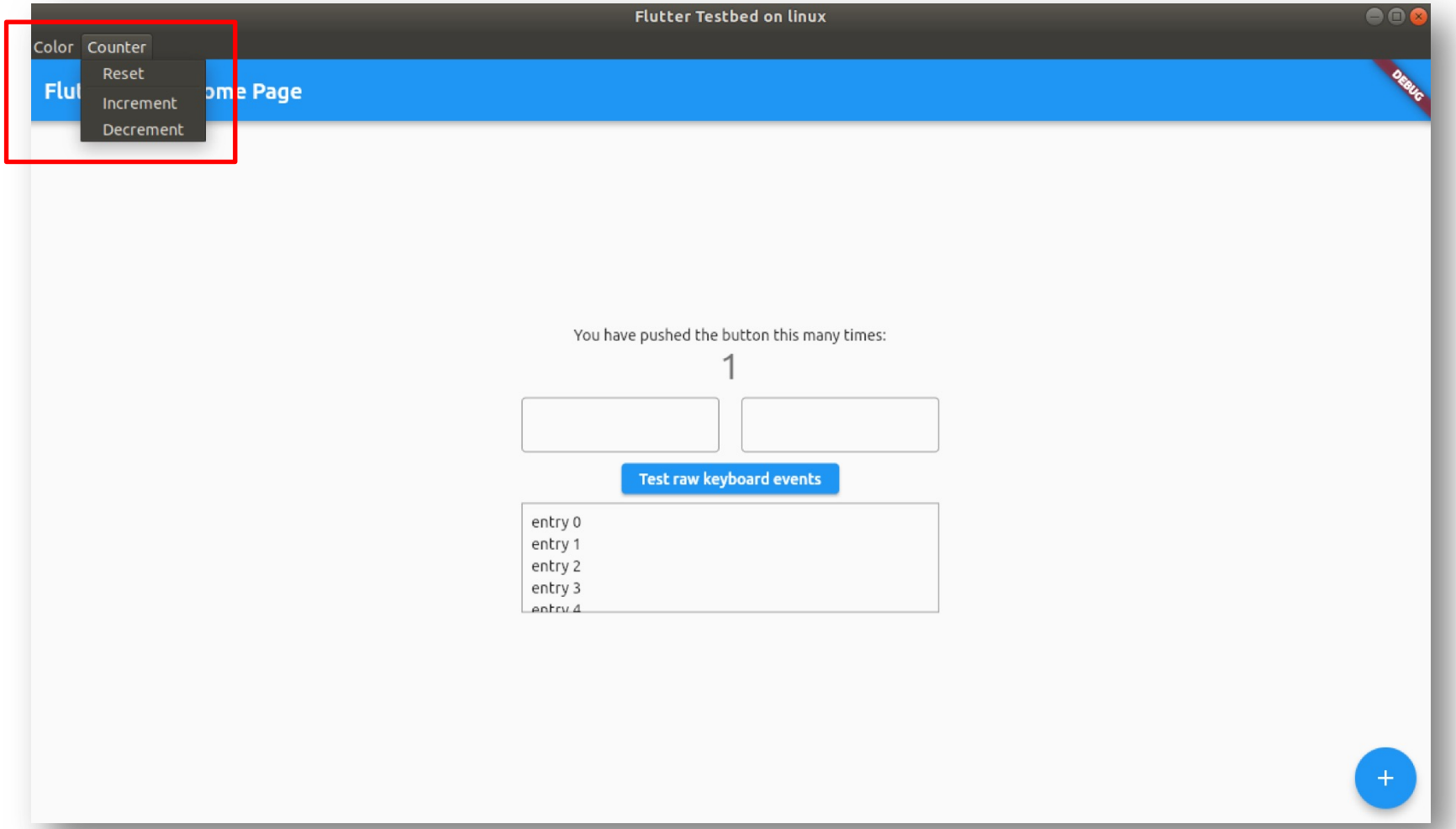
Example:

No	Item	Requirement	
		Desktop	Embedded
1	Widgets (GTK) to create a desktop-like UI	✓	-
2	Window manager: X11	✓	-
3	Window manager: Wayland	✓	✓ (partially)
4	Multi window	✓	✓ (partially)
5	Graphics composition in app	✓	✓
6	Keyboard input	✓	✓ (partially)
7	Touch input	✓ (partially)	✓ (partially)
8	Limited memory / storage capacity	-	✓
9	Limited CPU power	-	✓

SONY

Why is this embedder necessary instead of “Flutter desktop for Linux”?

For example, desktop apps require a widget like menubar (Need GTK)



https://github.com/google/flutter-desktop-embedding

SONY

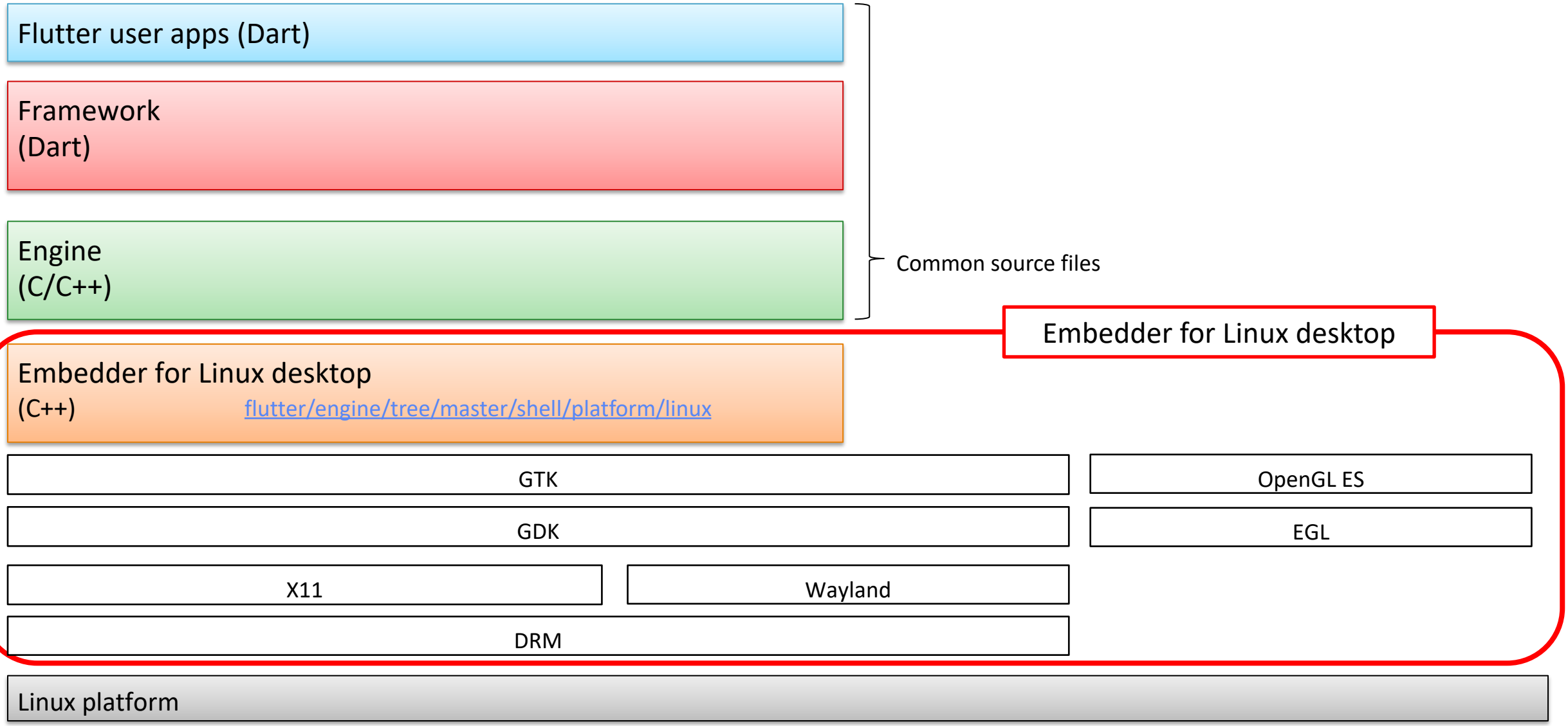
Why is this embedder necessary instead of “Flutter desktop for Linux”?

- Flutter desktop for Linux uses X11 and GTK/GDK

- Source files strongly dependent on GDK
  - We don’t want to install libraries that are not used as functions as much as possible
  - As a side note, we initially thought about sharing the source code with the desktop version but concluded that it was difficult

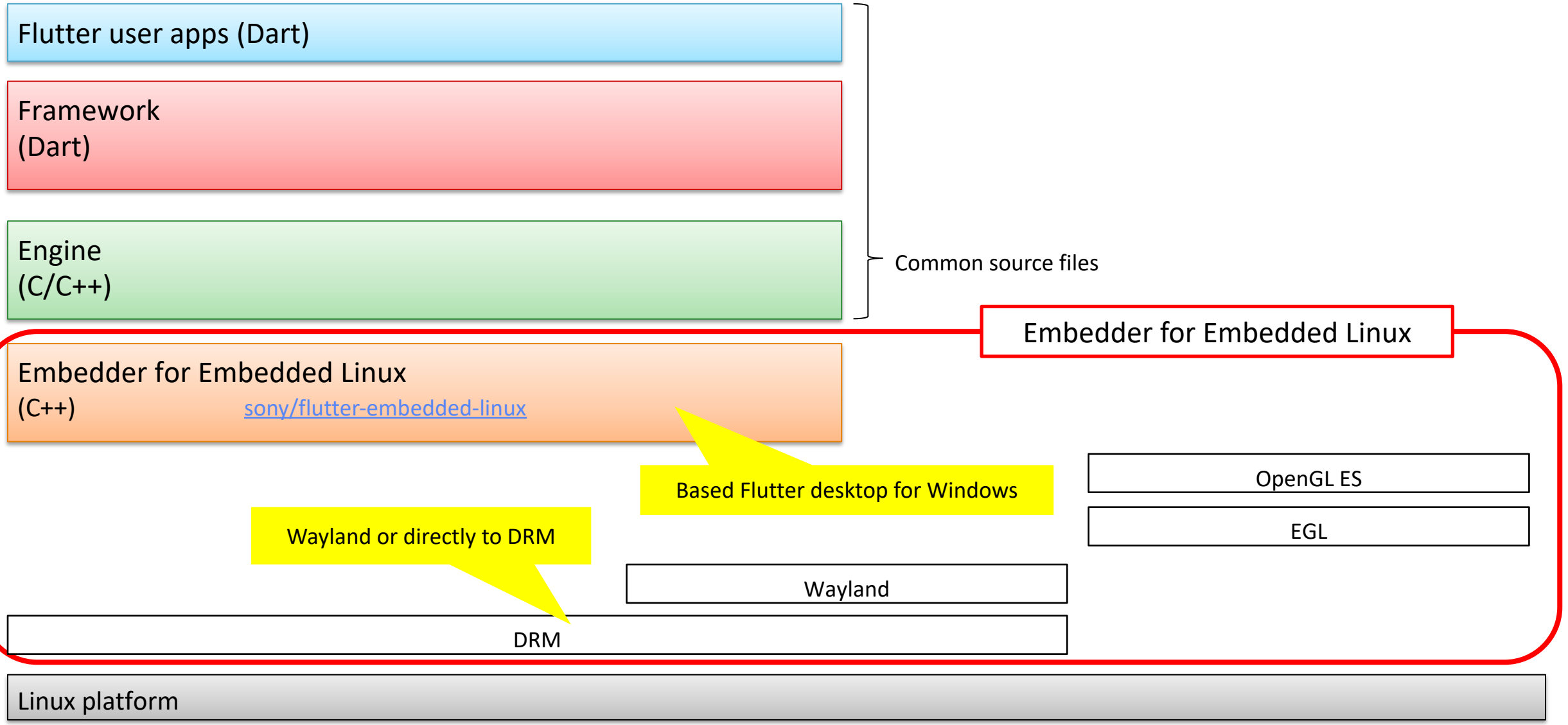
- X11 and GTK require a lot of dependent libraries (includes GPL/LGPL v3 licensed software)
  - e.g. (on Ubuntu 18.04)
    - ✓ Xserver-xorg: https://packages.ubuntu.com/bionic/xserver-xorg
    - ✓ libgtk-3: https://packages.ubuntu.com/bionic/libgtk-3-0

Flutter desktop for Linux

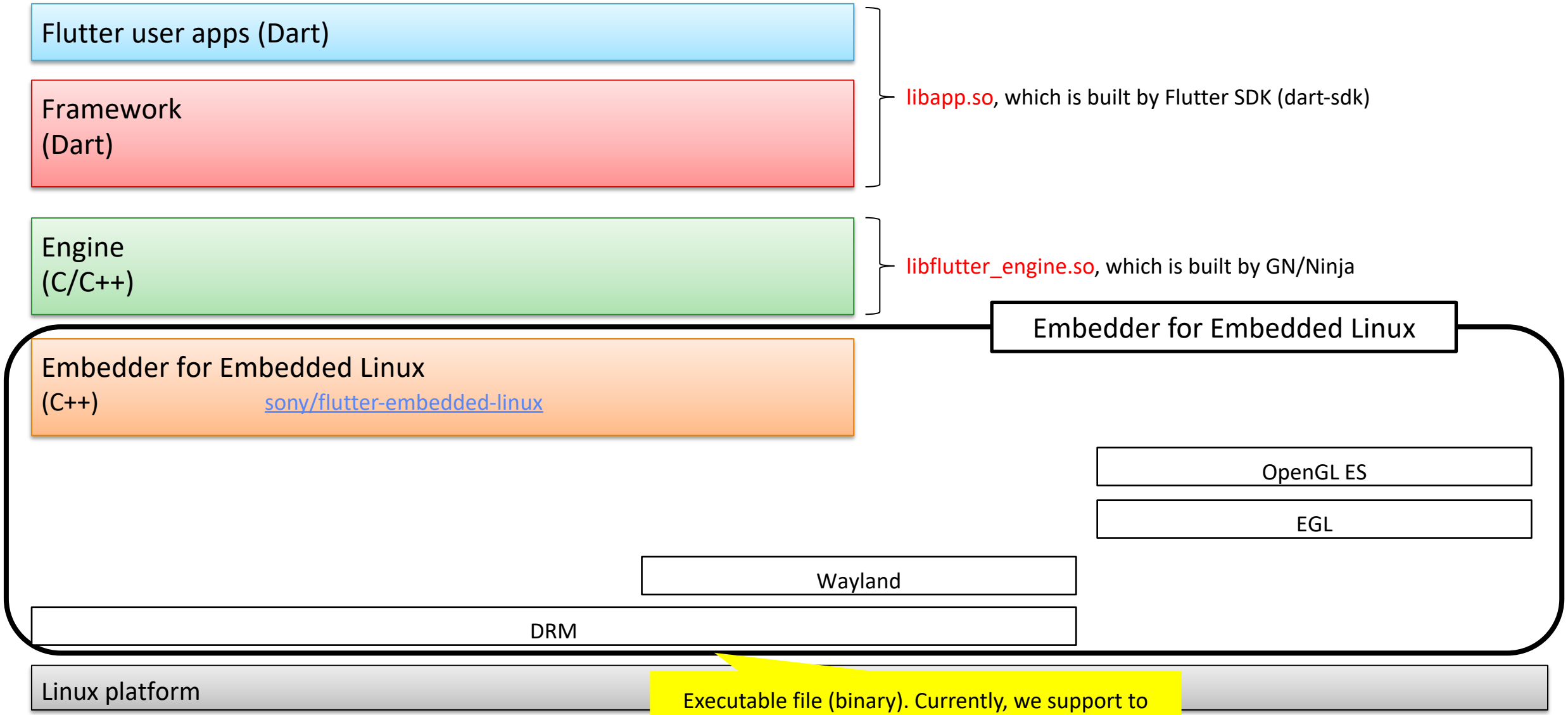


SONY

Flutter for Embedded Linux



Flutter artifacts



SONY

Build Flutter for embedded Linux using Yocto

See: https://github.com/sony/flutter-embedded-linux/tree/master/meta-flutter

```
$ git clone https://github.com/sony/flutter-embedded-linux.git

# Add meta-flutter layer into your conf/bblayers.conf
$ bitbake-layers add-layer ../flutter-embedded-linux/meta-flutter

# Build flutter-embedded-linux with Wayland backend
$ bitbake flutter-wayland-client
```

Build Flutter for embedded Linux using CMake

See: https://github.com/sony/flutter-embedded-linux/tree/master/doc

```
$ git clone https://github.com/sony/flutter-embedded-linux.git

$ mkdir build && cd build

# Self-build on x64 or arm64
$ cmake -DUSER_PROJECT_PATH=examples/flutter-wayland-client ..
$ cmake --build .

# Cross-build on x64 for arm64
$ cmake -DUSER_PROJECT_PATH= examples/flutter-wayland-client ¥
            -DCMAKE_TOOLCHAIN_FILE=<toolchain-template-file> ..
$ cmake --build .
```

Future works

- Embedder tasks
  - Platform Views (Texture composition in Flutter embedder) support
  - Multi / Dual-screen support
  - Vsync support
  - Add compiler switch to disable input function (Keyboard, touch, mouse)

  etc.

- Flutter plugins
  - Audio / Video Player
  - WebView
  - Path provider

  etc.

- Flutter SDK
  - Add / contribute custom-devices support to build and debug using Flutter SDK

Thank you for your time