Google Home, But Better
Smart Home Display with Flutter

01 **About**

02 Hardware

03 Software: Embedded Flutter

04 Software: Implementation


Snapp X
Experts
Flutter & Dart
Moritz Theis
@MoritzTheis
@GoogleDevExpert for Flutter & Dart
Co-Founder @SnappX_io
Co-Founder @SnappEmbedded
Co-Organiser @FlutterMunich
Flutter Dev

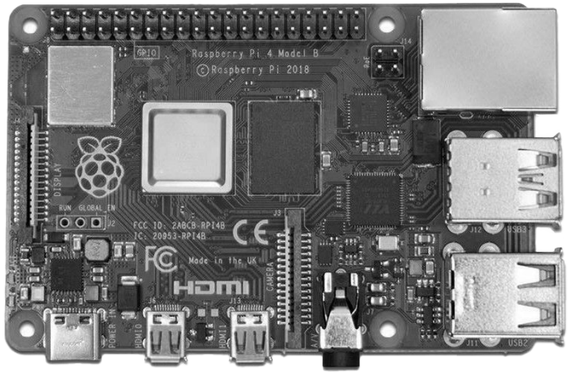

WEATHER
18.2°
Moderate rain
London
TODAY 12° 23°
THU 13° 24°
FRI 12° 25°
SAT 11° 22°
01:20 AM
Thu, Oct 9 2023
LIGHT
POWER CONSUMPTION KWH
250
200
150
100
50
0
MON TUE WED THU FRI SAT SUN
AIR QUALITY
POOR
TEMPERATURE
18.2°
HUMIDITY
43.3%
LIVING ROOM
TV
LIVING ROOM
STEREO
LIVING ROOM
THERMOSTAT
LIVING ROOM
FAN

01 About

02 **Hardware**

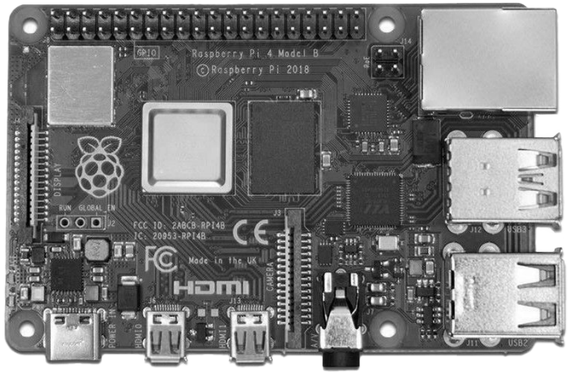
03 Software: Embedded Flutter

04 Software: Implementation


Raspberry Pi 4 Model B
Raspberry Pi 2018
HDMI


3-5V
SDA
SCL
GND
CO2
pimoroni.com


Shelly BULB
RGBW



Development Board

Raspberry Pi

Model 4B

4Gb Ram

32Gb SD Card

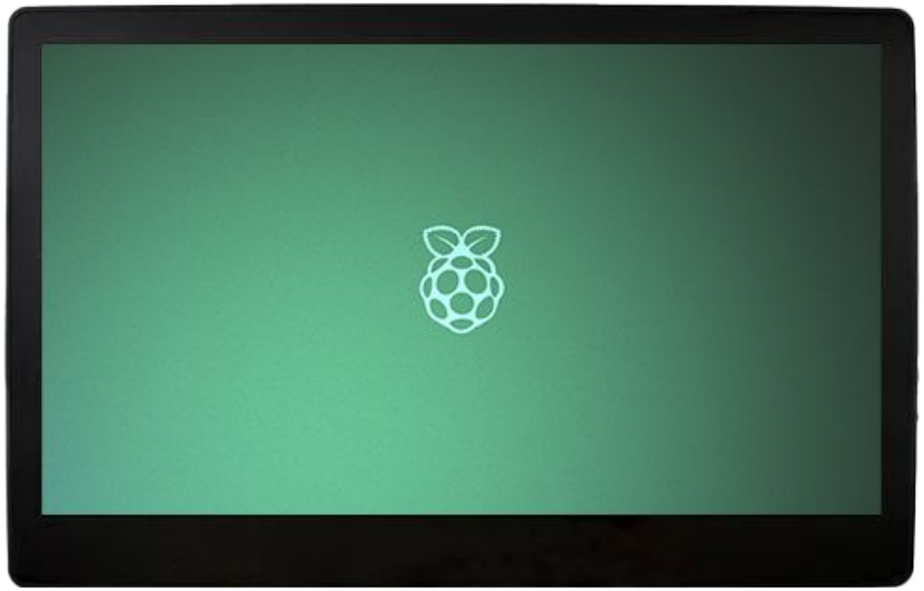
Air Sensor

Pimoroni SCD41 (Sensiron)

CO2, Temperature and Humidity

I2C Connection

Raspberry Pi Python Library



Display

11.6inch IPS panel

Capacitive Touch via. USB

Connection via. HDMI

Tested with Raspberry Pi

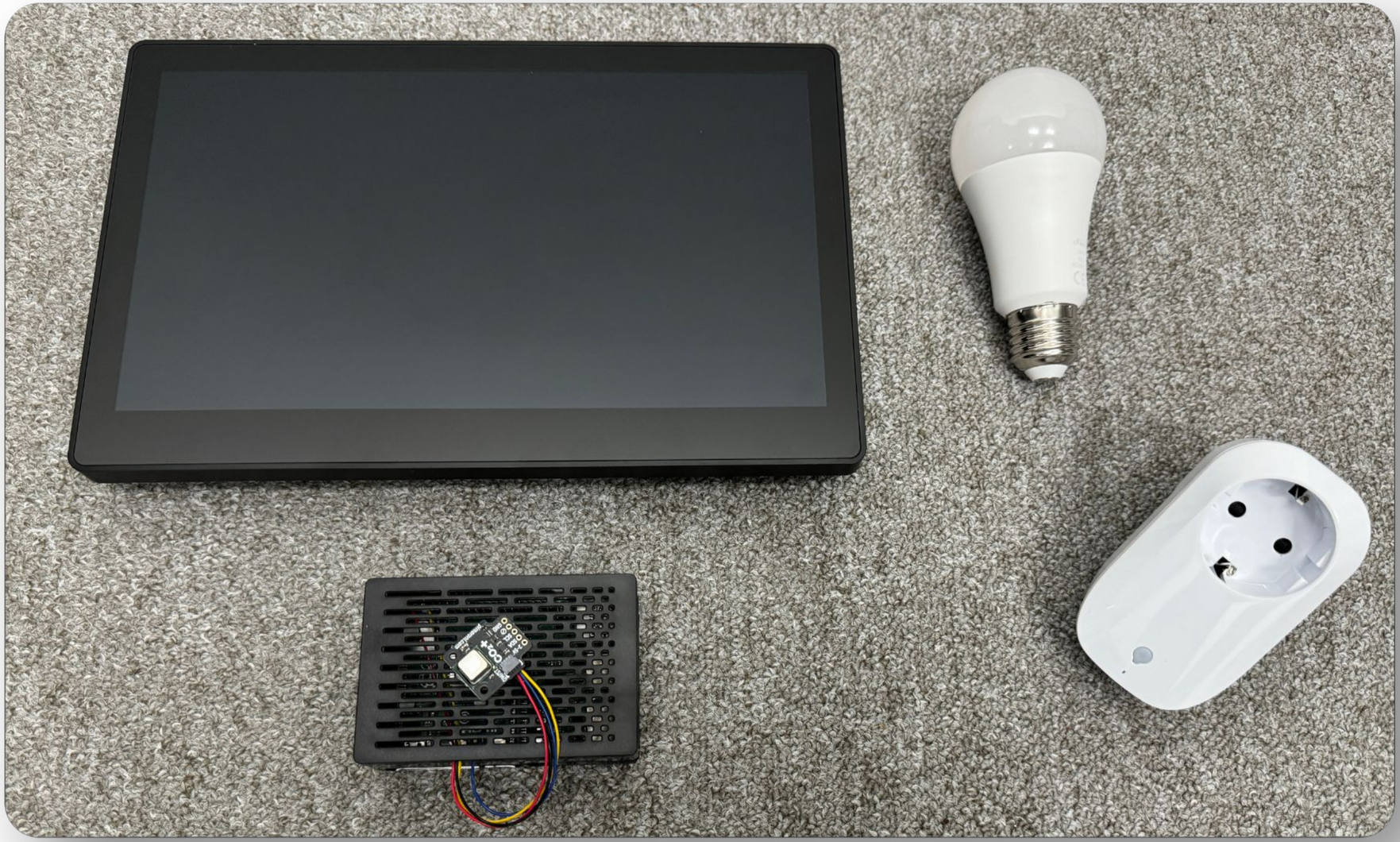
Smart Bulbs and Plugs

Shelly DUO RGBW & Shelly Plug

Wifi and Bluetooth

Built-in web server

Controllable via. REST-API

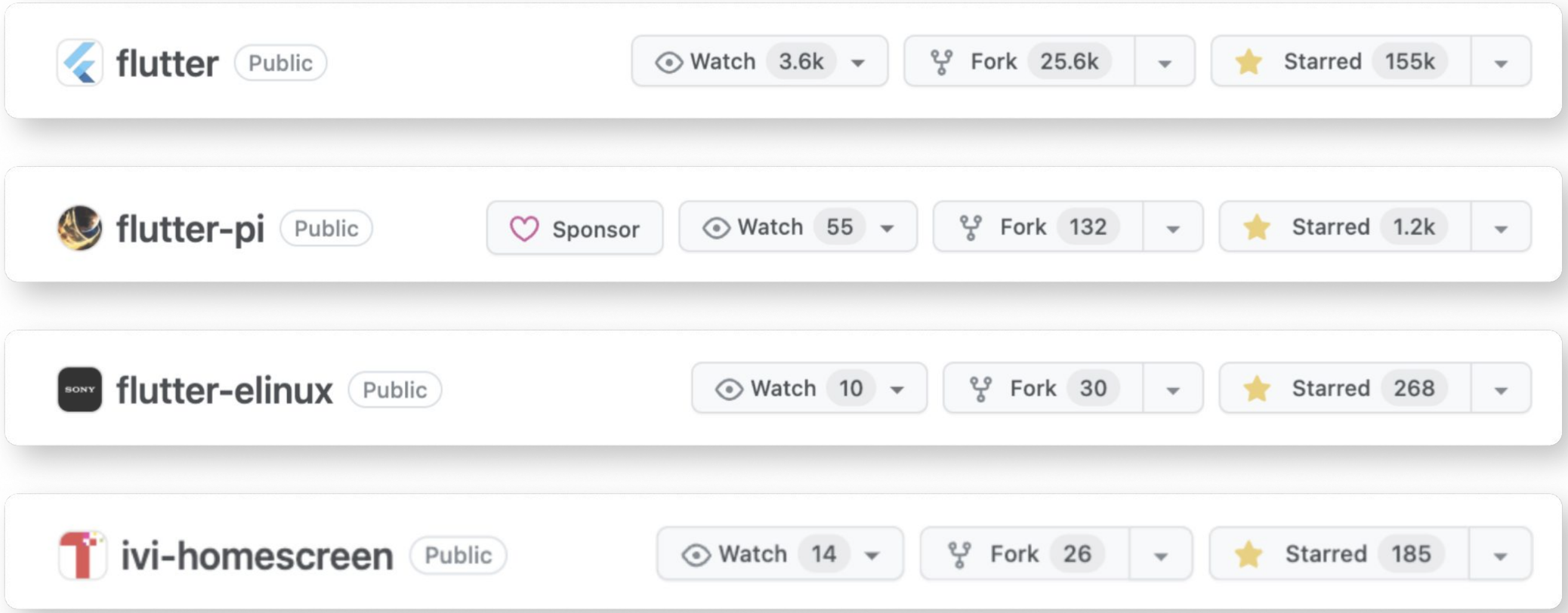


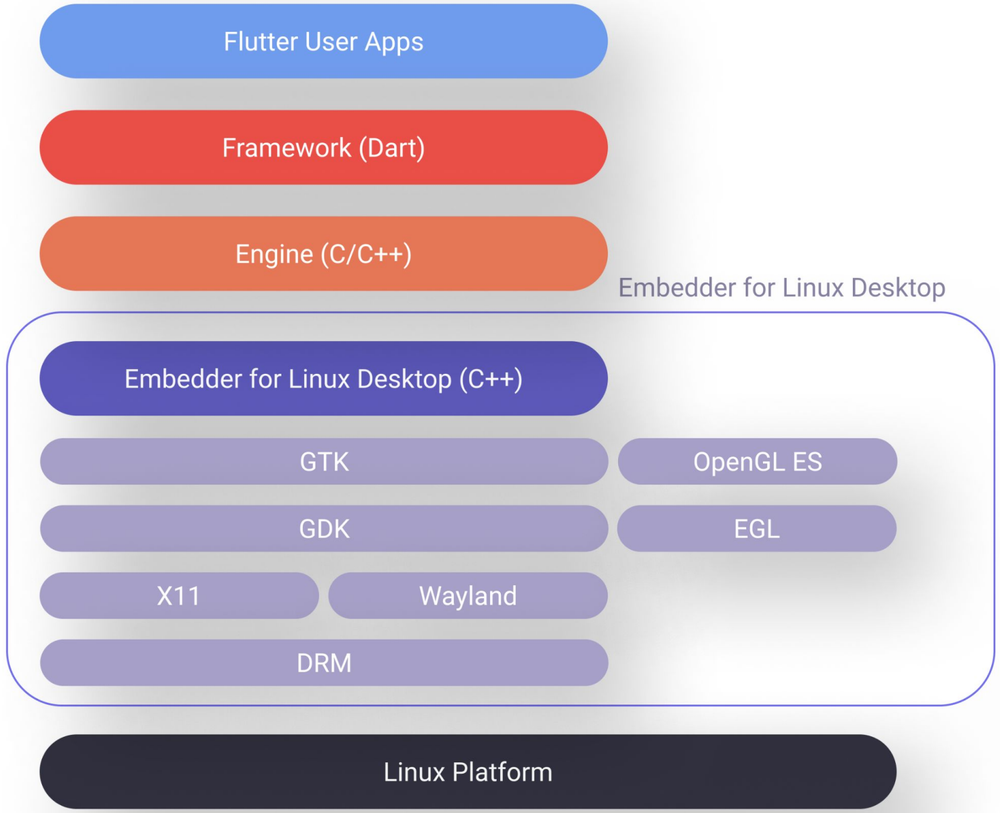
01 About

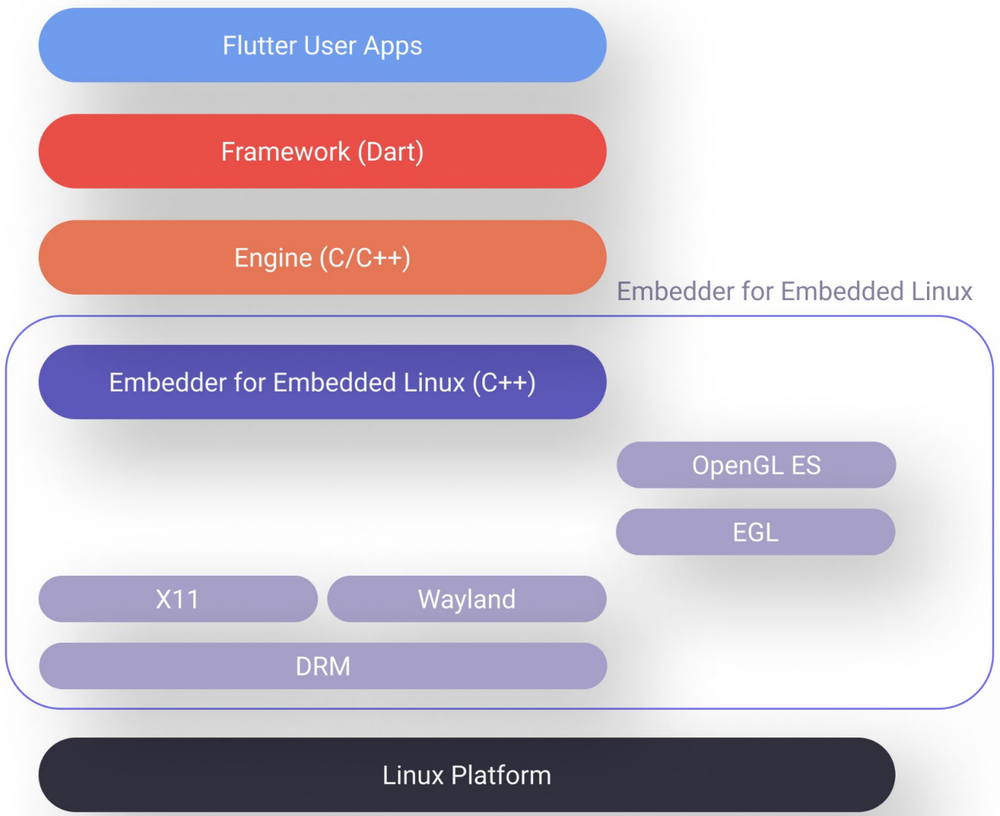
02 Hardware

03 **Software: Embedded Flutter**

04 Software: Implementation


flutter
Public
Watch 3.6k
Fork 25.6k
Starred 155k
flutter-pi
Public
Sponsor
Watch 55
Fork 132
Starred 1.2k
flutter-elinux
Public
Watch 10
Fork 30
Starred 268
ivi-homescreen
Public
Watch 14
Fork 26
Starred 185


Flutter User Apps
Framework (Dart)
Engine (C/C++)
Embedder for Linux Desktop
Embedder for Linux Desktop (C++)
GTK
OpenGL ES
GDK
EGL
X11
Wayland
DRM
Linux Platform


Flutter User Apps
Framework (Dart)
Engine (C/C++)
Embedder for Embedded Linux
Embedder for Embedded Linux (C++)
OpenGL ES
EGL
X11
Wayland
DRM
Linux Platform

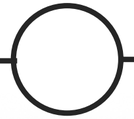
Status of Flutter on embedded Devices

Phase 1

Enterprises

- Custom embedders for Wayland & DRM are available
- OS & tooling for eLinux are custom builds

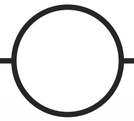
Phase 2



Community Projects

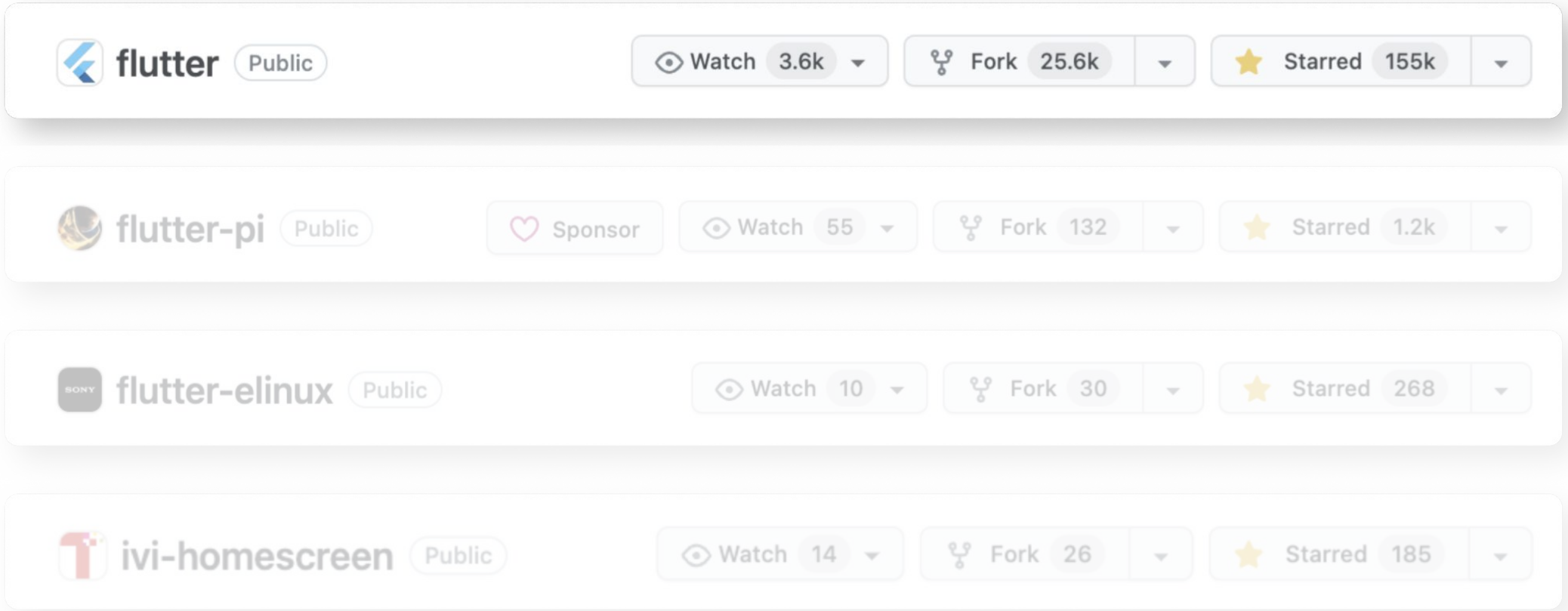
- Basic “plug & pray” operating systems for Flutter apps are available
- Tutorials & how-to’s are available

Phase 3



SMEs & Startups

- Flutter tailored OS and device management available as a service


flutter
Public
Watch 3.6k
Fork 25.6k
Starred 155k
flutter-pi
Public
Sponsor
Watch 55
Fork 132
Starred 1.2k
flutter-elinux
Public
Watch 10
Fork 30
Starred 268
ivi-homescreen
Public
Watch 14
Fork 26
Starred 185

01 About

02 Hardware

03 Software: Embedded Flutter

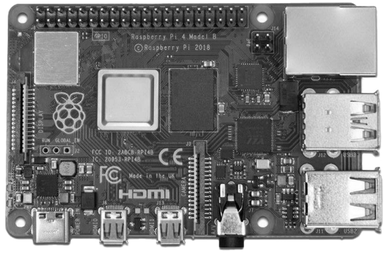
04 **Software: Implementation**

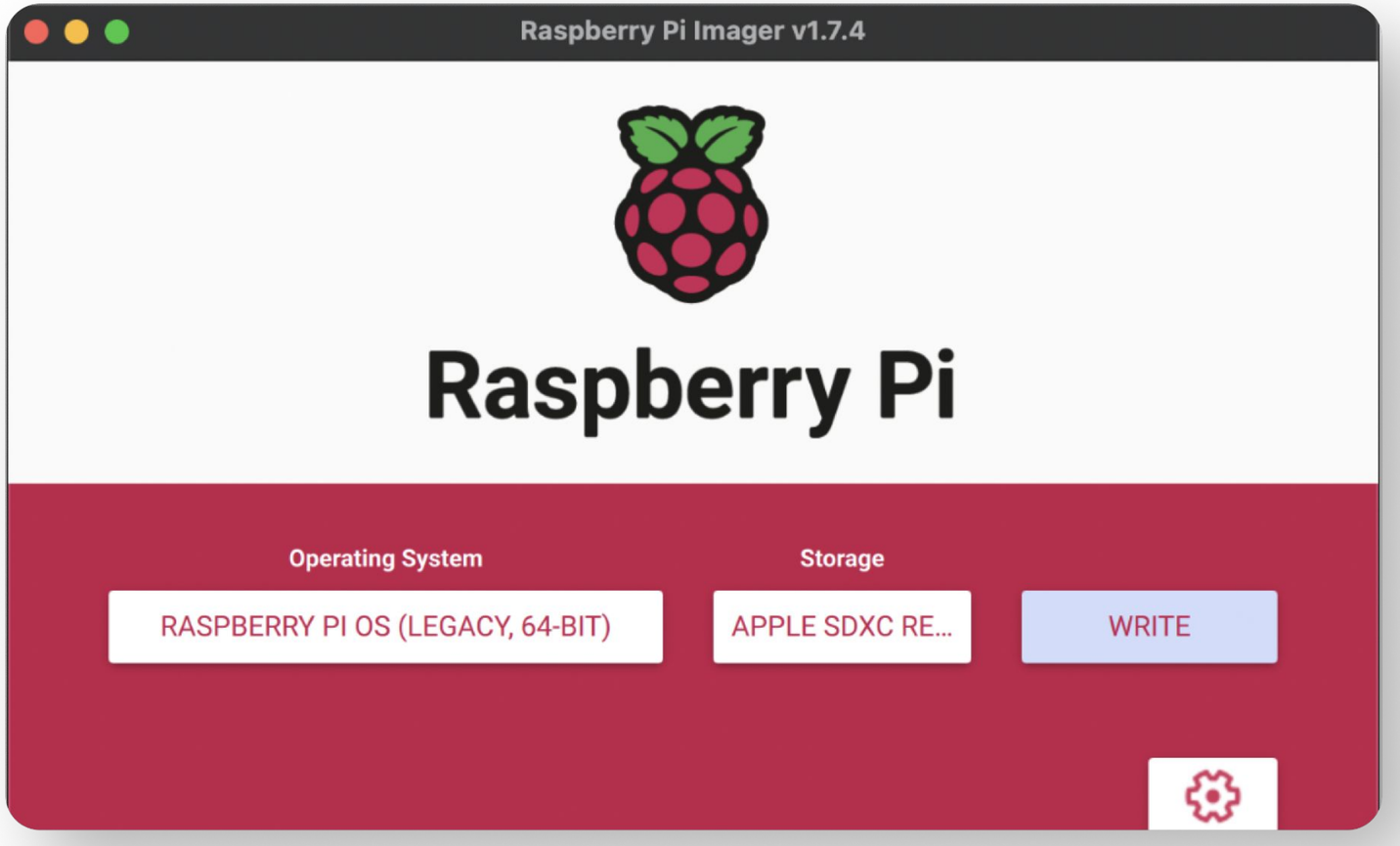
Display & Touch

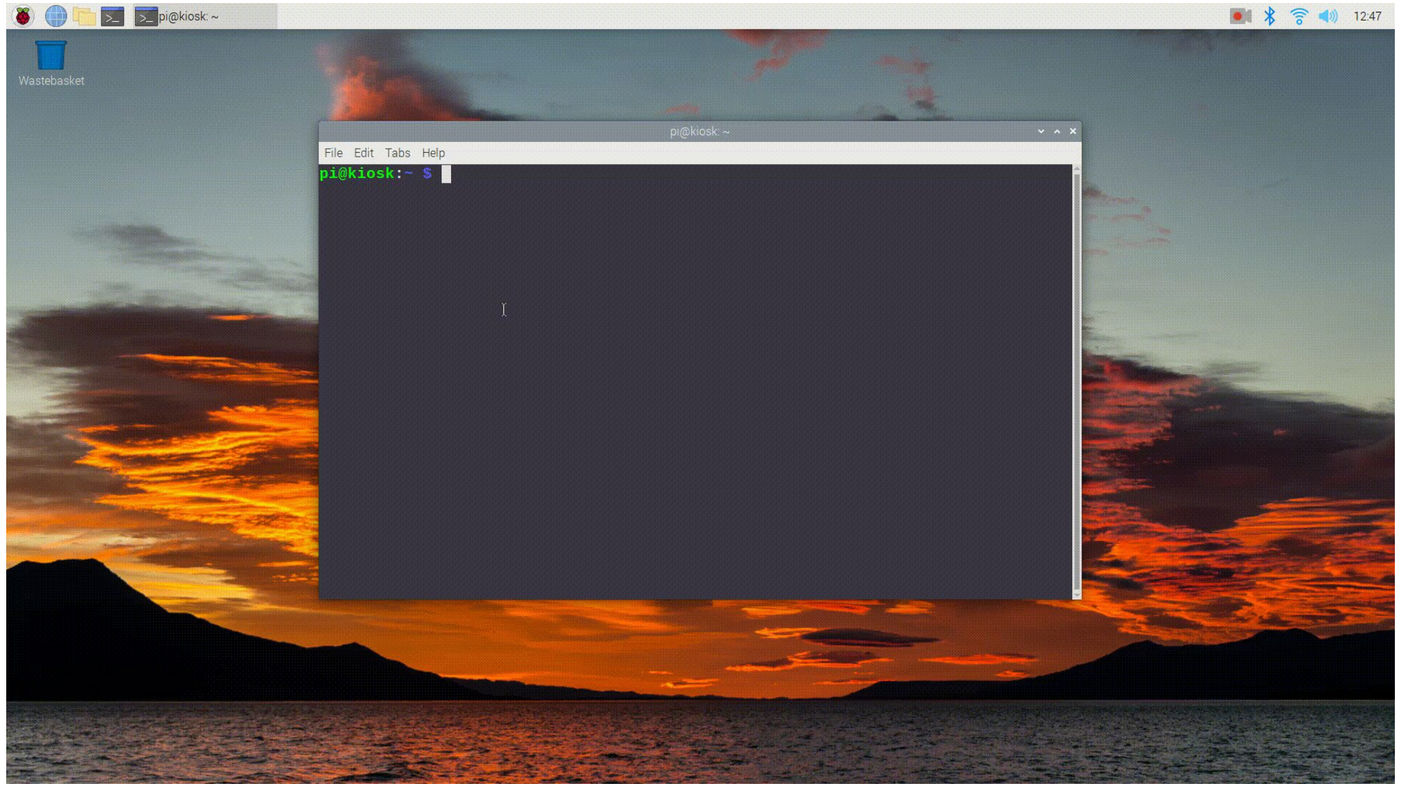
HDMI

USB

Raspberry Pi
Model 4B 4GB




Raspberry Pi Imager v1.7.4
Raspberry Pi
Operating System
Storage
RASPBERRY PI OS (LEGACY, 64-BIT)
APPLE SDXC RE...
WRITE

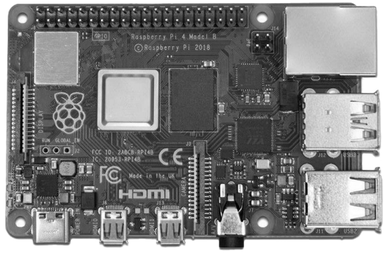

pi@kiosk: ~
12:47
Wastebasket
pi@kiosk: ~
File Edit Tabs Help
pi@kiosk:~ $

Display & Touch

HDMI

USB

Raspberry Pi
Model 4B 4GB



WIFI

Plug & Bulb

```dart
Future<BulbStatusEither> getBulbStatus() async {
  try {
    final response = await httpService.get('$bulbUrl/status');

    final bulbStatus = BulbStatusResponse.fromJson(
      response,
    ).toBulbStatus();
    return right(
      bulbStatus,
    );
  } on DioException catch (error) {
    return left(error);
  }
}
```

```dart
Future<LightEither> setBulb(bool on) async {
  final turn = on ? 'on' : 'off';
  try {
    final response = await httpService.get('$bulbUrl/color/0?turn=$turn');

    final light = LightResponse.fromJson(
      response,
    ).toLight();
    return right(
      light,
    );
  } on DioException catch (error) {
    return left(error);
  }
}
```

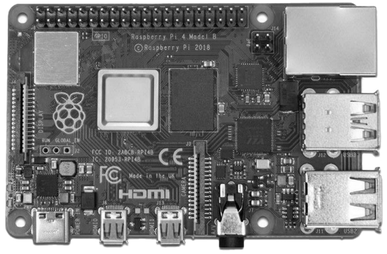
```dart
Future<LightEither> setBulbColor(Color color) async {
  try {
    final response = await httpService.get(
        '$bulbUrl/color/0?turn=on&red=${color.red}&green=${color.green}&
          blue=${color.blue}&white=0');

    final light = LightResponse.fromJson(
      response,
    ).toLight();
    return right(
      light,
    );
  } on DioException catch (error) {
    return left(error);
  }
}
```

Display & Touch

HDMI

USB

Raspberry Pi
Model 4B 4GB

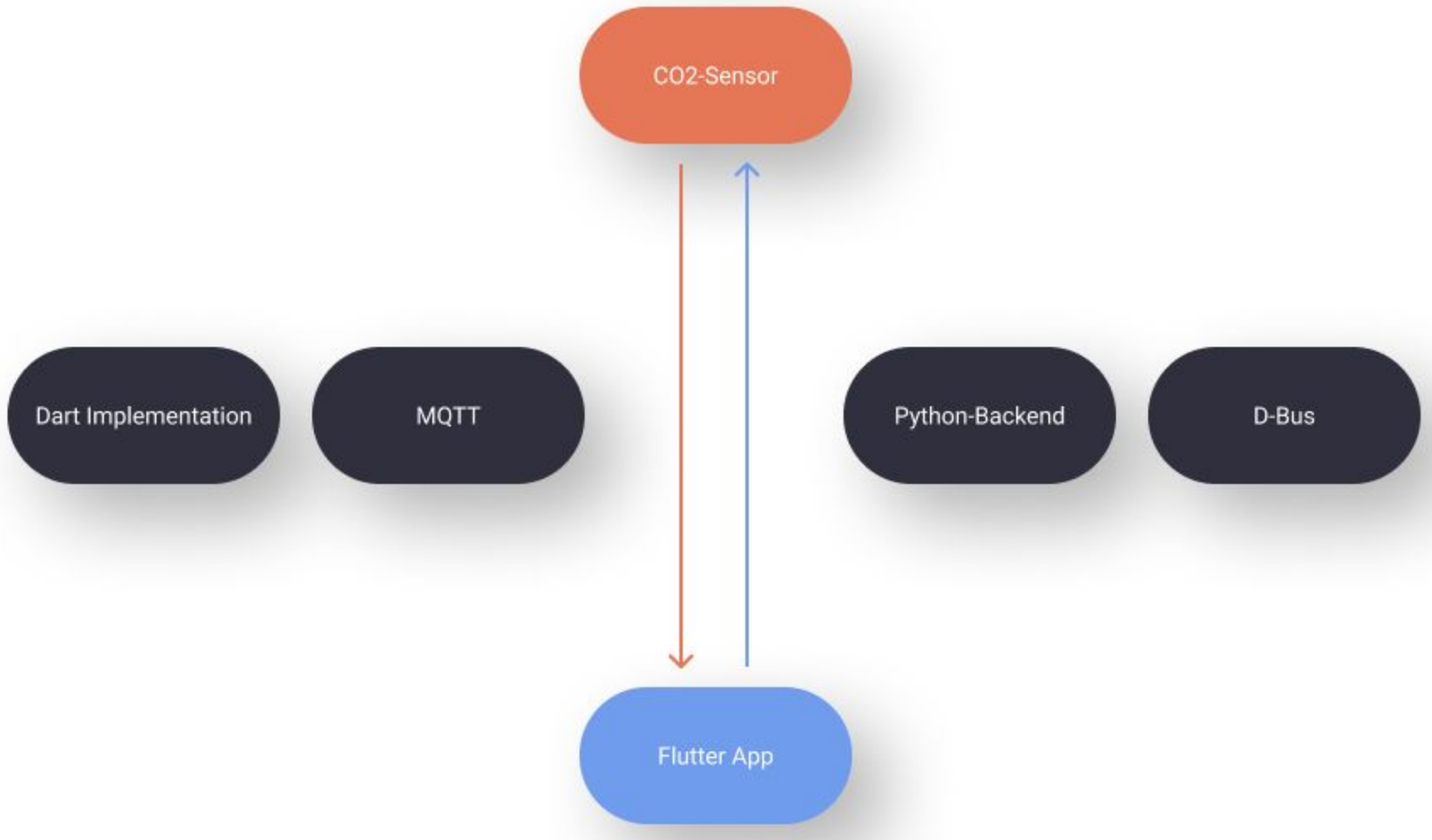


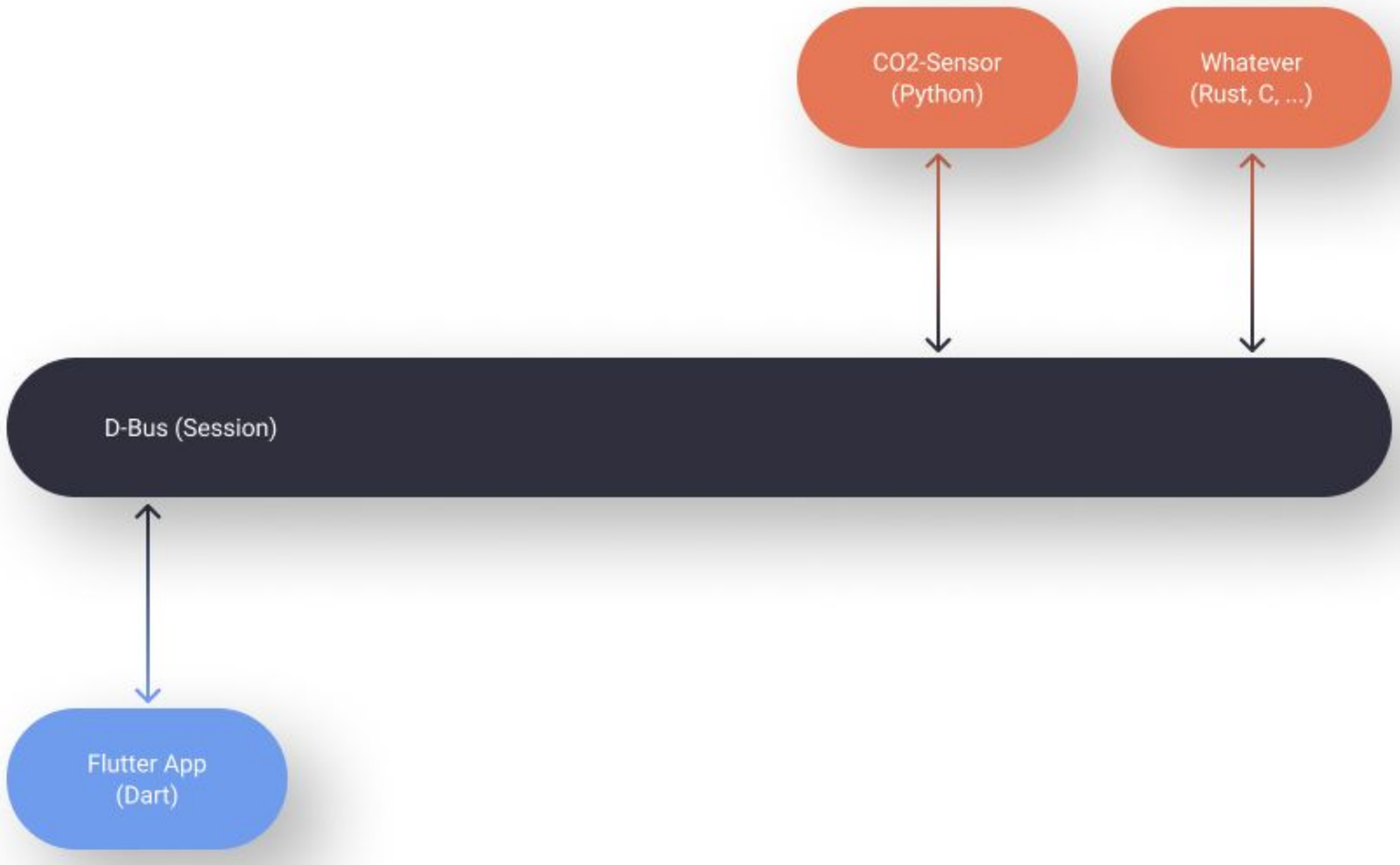
I2C

WIFI

CO2

Plug & Bulb


CO2-Sensor
Dart Implementation
MQTT
Python-Backend
D-Bus
Flutter App


CO2-Sensor
(Python)
Whatever
(Rust, C, ...)
D-Bus (Session)
Flutter App
(Dart)

```python
#!/usr/bin/env python3

import dbus
from scd4x import SCD4X
from gi.repository import GLib
import dbus.mainloop.glib

if __name__ == '__main__':
    dbus.mainloop.glib.DBusGMainLoop(set_as_default=True)

    device = SCD4X(quiet=False)
    device.start_periodic_measurement()

    session_bus = dbus.SessionBus()
    name = dbus.service.BusName("de.snapp.CoSensorService", session_bus)
    object = Sensor(session_bus, '/Sensor')

    mainloop = GLib.MainLoop()
    mainloop.run()

    ...
```

```
...

class Sensor(dbus.service.Object):

    @dbus.service.method("de.snapp.SensorInterface",
                         in_signature='', out_signature='as')
    def GetSensorValue(self):
        # print the name of the bus name we are connected to
        print("GetSensorValue request:", session_bus.get_unique_name())
        # call the scd4x sensor and return the values
        co2, temperature, relative_humidity, timestamp = device.measure()
        return [f"Temperature: {temperature:.4f}°C", f"Humidity: {relative_humidity:.2f}%",
            f"CO2: {co2:.2f}PPM",
                session_bus.get_unique_name()]

    @dbus.service.method("de.snapp.SensorInterface",
                         in_signature='', out_signature='')
    def Exit(self):
        mainloop.quit()
```

```dart
import 'package:dbus/dbus.dart';

Future<void> main() async {
  final client = DBusClient.session();
  final object = DBusRemoteObject(client,
      name: 'de.snapp.CoSensorService', path: DBusObjectPath('/Sensor'));

  final response = await object.callMethod(
      'de.snapp.SensorInterface', 'GetSensorValue', [],
      replySignature: DBusSignature('as'));

  /// convert DBusArray to List
  final result = List.from(response.values[0].asStringArray());

  print('temp: ${result[0]}');
  print('hum: ${result[1]}');
  print('co2: ${result[2]}');

  await client.close();
}
```


WEATHER
18.2°
Moderate rain
London
TODAY 12° 23°
THU 13° 24°
FRI 12° 25°
SAT 11° 22°
01:20 AM
Thu, Oct 9 2023
LIGHT
POWER CONSUMPTION KWH
250
200
150
100
50
0
MON TUE WED THU FRI SAT SUN
AIR QUALITY
POOR
TEMPERATURE
18.2°
HUMIDITY
43.3%
LIVING ROOM
TV
LIVING ROOM
STEREO
LIVING ROOM
THERMOSTAT
LIVING ROOM
FAN

Resources

Github-Repo

X (Twitter)