INTERACTION DESIGN

MQTT

Physical Computing HS22

MQTT

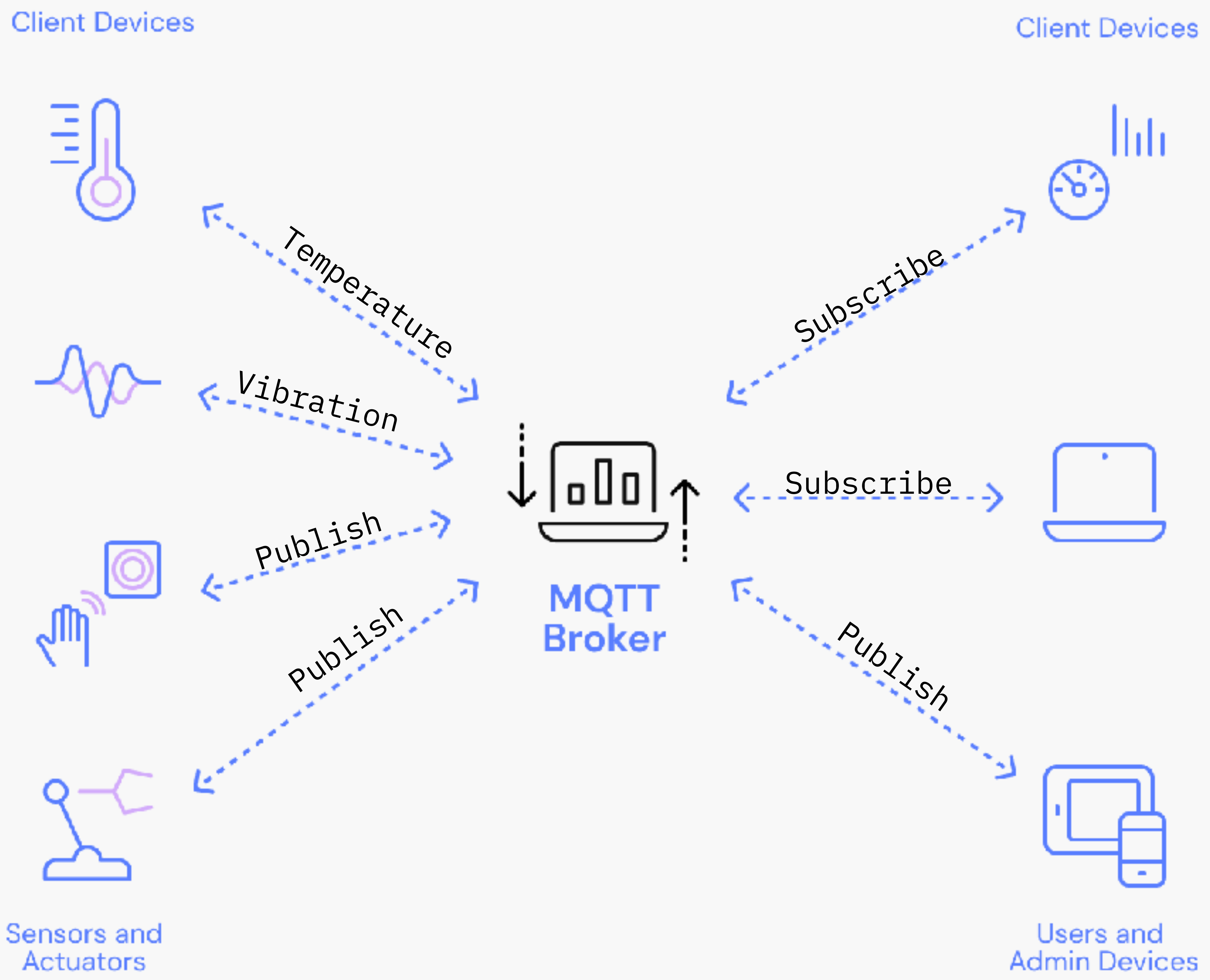
MQTT is standard messaging protocol for the **Internet of Things (IoT)**. It is designed as an extremely lightweight publish/subscribe messaging transport that is ideal for connecting remote devices with a small code footprint and minimal network bandwidth.

MQTT

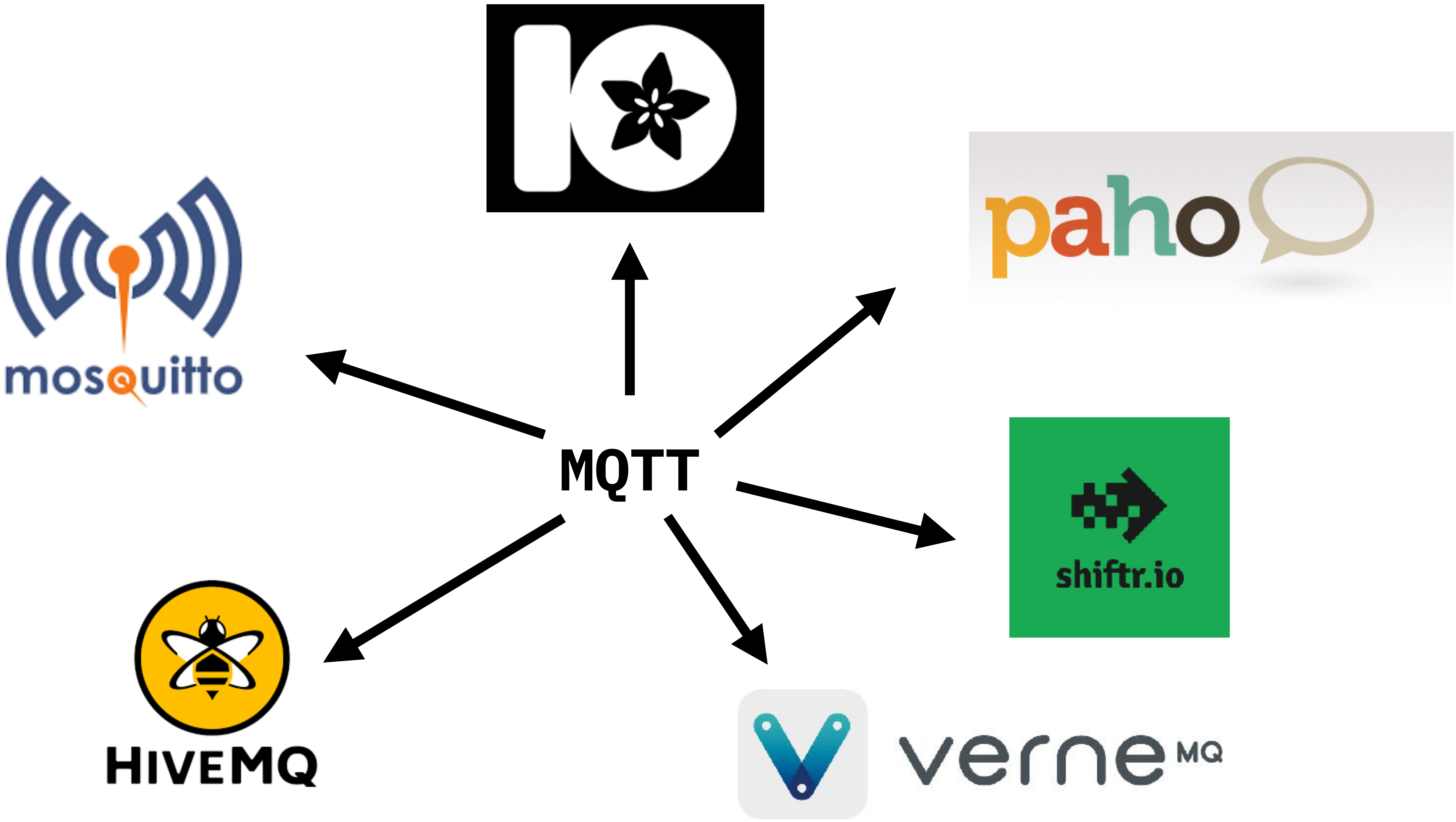
- Connect and communicate between different devices
- Designed for resource-constrained devices
- Used across platforms while consuming minimal bandwidth
- Easy integration of new devices
- Getting data from Arduino via WIFI!

PUBLISH / SUBSCRIBE

MQTT messages are published and subscribed to as a **payload** i.e the essential data that is being carried within a packet. A payload is a string formatted piece of information that can be e.x. value from a sensor, user interaction etc.



MQTT BROKERS



MQTT

SHIFTR.IO

- Developed at ZHdK in 2015 by Joël Gähwiler as MA project.
- Open-source and free to use (limit of messages)
- Cloud or desktop-based (more bandwidth, and less latency)
- **The payload size of publish messages is limited to 64 KB**.
- The active subscriptions per connection are limited to 100.

INTERACTION DESIGN

SHIFTR.IO + ARDUINO

Physical Computing HS22

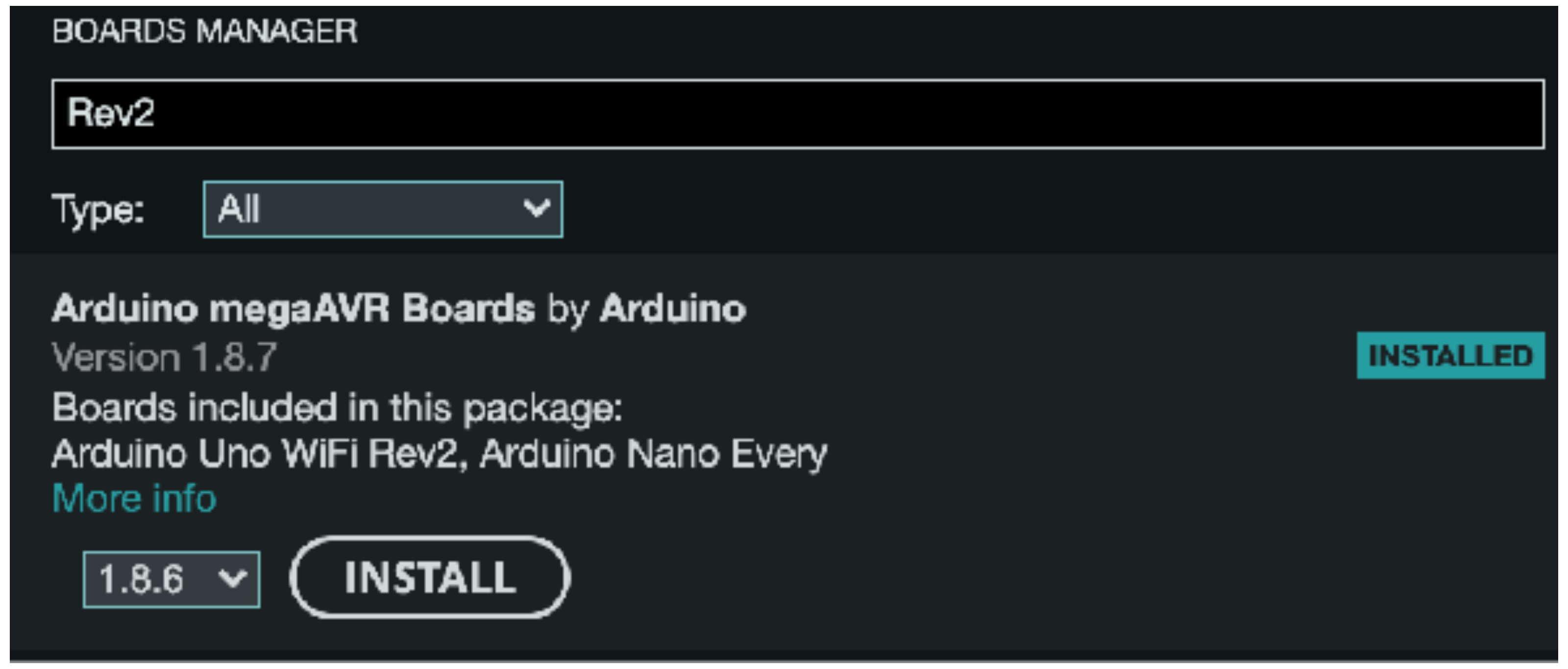
shiftr.io

Connect to our own WiFi, ZHDK does not allow to send data over network, as the connections are not secure.

SSID : iad_zhdk
Password: i@d_4ever

1.After plugging in Arduino Uno WiFi install the board in:
Arduino->Tools->Board->Boards Manager

2. Install **Arduino megaAVR Boards** library:



BOARDS MANAGER

Rev2

Type: All

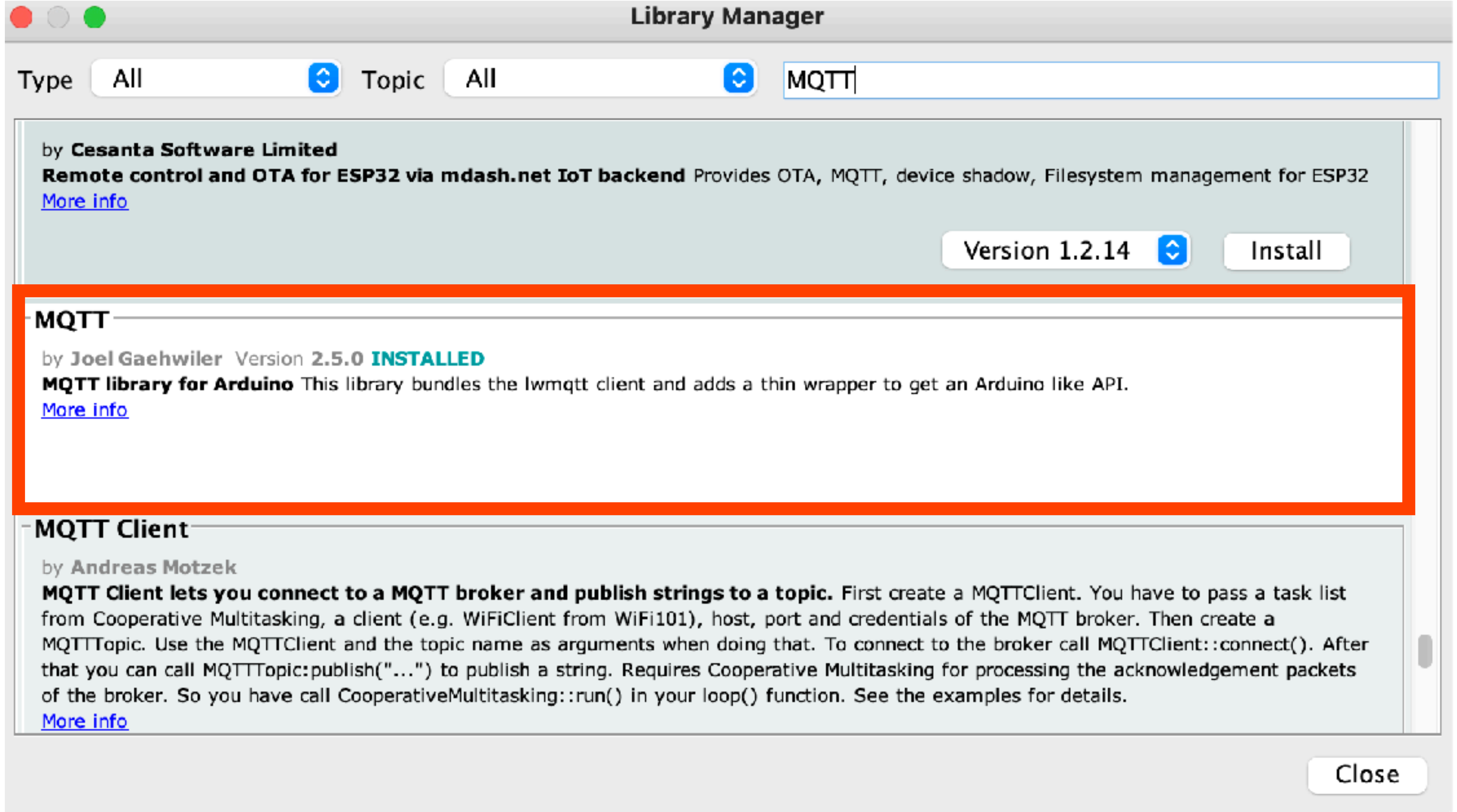
Arduino megaAVR Boards by **Arduino**
Version 1.8.7
Boards included in this package:
Arduino Uno WiFi Rev2, Arduino Nano Every

3.Select the board in :
Arduino->Tools->Board->Arduino megaAVR Boards->Arduino Uno WiFi Rev2

4.Go to:
Tools -> WiFi/WiFiNINA Firmware Updater

3.Select your board and click “Install”. If your board is not visible make sure you selected it in:
Arduino->Tools->Port

Install **MQTT** by **Joel Gaehwiler** in Library Manager:



Type All Topic All MQTT

by **Cesanta Software Limited**
Remote control and OTA for ESP32 via mdash.net IoT backend Provides OTA, MQTT, device shadow, Filesystem management for ESP32
More info

MQTT

by **Joel Gaehwiler** Version **2.5.0** **INSTALLED**
MQTT library for Arduino This library bundles the lwmqtt client and adds a thin wrapper to get an Arduino like API.
More info

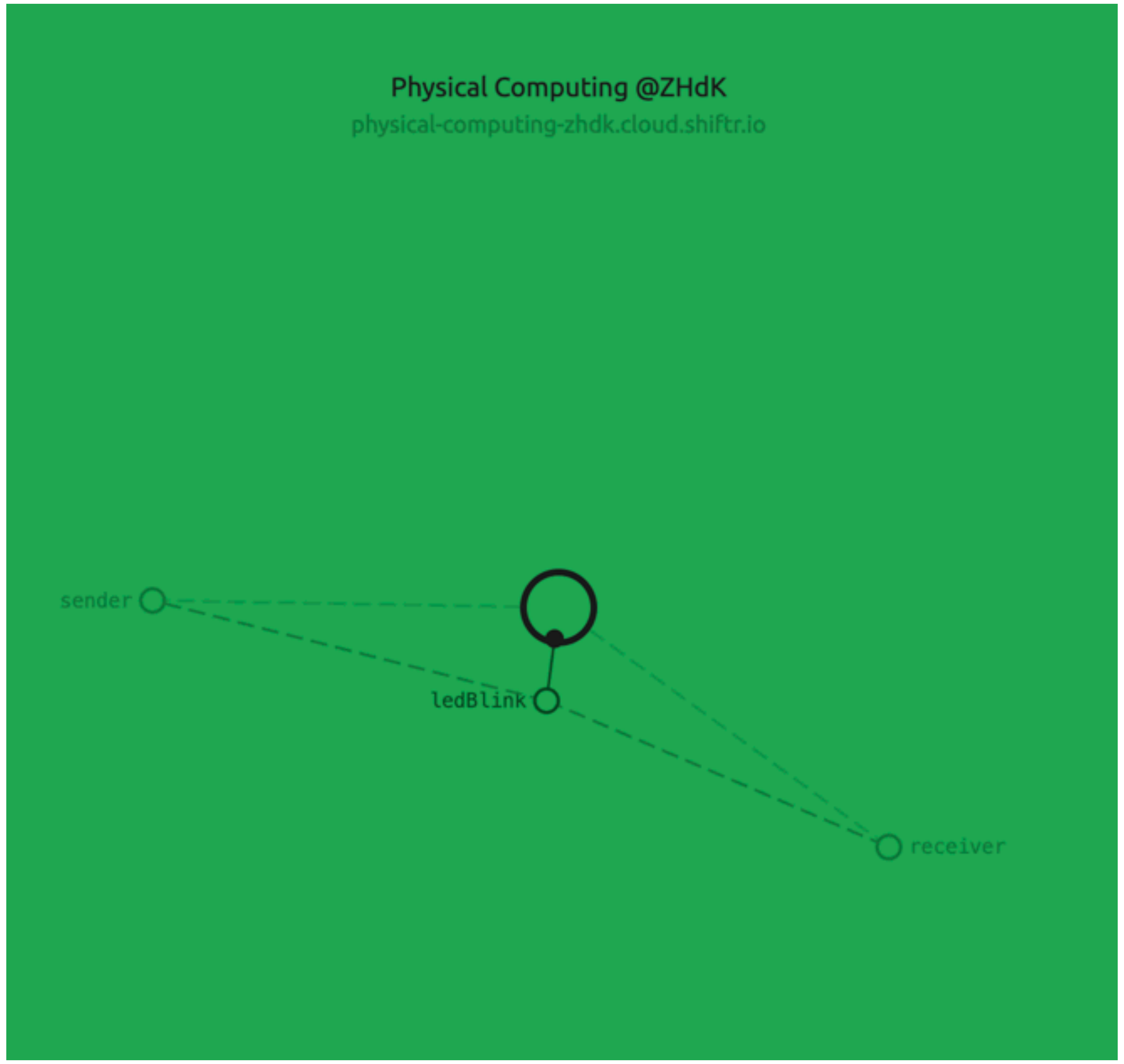
MQTT Client

by **Andreas Motzek**
MQTT Client lets you connect to a MQTT broker and publish strings to a topic. First create a MQTTClient. You have to pass a task list from Cooperative Multitasking, a client (e.g. WiFiClient from WiFi101), host, port and credentials of the MQTT broker. Then create a MQTTTopic. Use the MQTTClient and the topic name as arguments when doing that. To connect to the broker call MQTTClient::connect(). After that you can call MQTTTopic:publish("...") to publish a string. Requires Cooperative Multitasking for processing the acknowledgement packets of the broker. So you have call CooperativeMultitasking::run() in your loop() function. See the examples for details.
More info

Example 1

Send data from Arduino to Arduino using shiftr.io

In groups of two decide who is going to be the sender and who the receiver.



Physical Computing @ZHdK

physical-computing-zhdk.cloud.shiftr.io

shiftr.io

Connect to client

```
client.connect("sender", "physical-computing-zhdk", "QO1d1kxcIhqD2pi2")
```

”sender”: client ID displayed as the connection name

”physical-computing-zhdk”: name of the instance you send data to

“O1d1kxcIhqD2pi2”: secret token configured in the settings panel

Start Instance

```
client.begin("physical-computing-zhdk.cloud.shiftr.io", net);
```

"physical-computing-zhdk.cloud.shiftr.io": instance domain

net: depends on the chosen network client. Use net!

shiftr.io

Receiving Messages

```
client.subscribe(“ledBlink");
```

“ledBlink” : the name of the topic to subscribe.

Sending Messages

```
client.publish("/ledBlink", String(ledBlink));
```

“ledBlink" : the topic to publish the message to.

String(ledBlink) : the payload of the message. IT HAS TO BE A STRING!

Exercise 1

Sign up to shiftr.io and deploy your own shiftr.io instance and adapt the values in the Arduino code from Example 1 to match your own credentials.

Exercise 2

In groups of two decide who is going to be **the sender** and who is going to be **the receiver.**

The receiver: Connect a LED to Arduino (values from 0 to 255)
The sender: Connect a potentiometer to Arduino (values from 0 to 1023)

Adapt the code so that **the sender** fades in/out the LED of **the receiver** by sending the values received from potentiometer via shiftr.io

For this exercise decide whose shiftr.io instance you are going to use!

INTERACTION DESIGN

SHIFTR.IO + P5.JS

Physical Computing HS22

Example 2

Send data from Arduino to p5.js via shiftr.io

shiftr.io

```
let broker = {
    hostname: 'physical-computing-zhdk.cloud.shiftr.io',
    port: 443
};

let creds = {
      clientID: ‘p5',
      userName: ‘physical-computing-zhdk',
      password: ‘QO1d1kxcIhqD2pi2'
}
```

For shiftr.io Cloud instances the interface is available over the insecure port 1883 (TCP), secure port 8883 (TLS) and secure **WebSocket port 443 (WSS/HTTPS).**

With shiftr.io Desktop the interface is only available over the insecure port 1883 (TCP) and WebSocket port 1884 (WS/HTTP) due to the lack of a certificate. Other ports are selected if one of the ports is already in use by another application.

Exercise 3

Create a new topic called “diameter”

Control the diameter of the circle from Example 2 using the values from a potentiometer plugged into Arduino.

Make sure you don’t go above the diameter of 300.

Exercise 4

Connect LED to your Arduino and control its brightness (analogWrite()) with a p5.js slider.

Use createSlider() function and slider.value(). Make sure the range of your slider is between 0 and 255.

Example 3

Using the p5.js color picker change the colors of a NeoPixel attached to Arduino on pin 6.

Make sure you connect NeoPixel DIN pin (not DOUT)!