INTERACTION DESIGN

# ARDUINO + P5.JS + MQTT

Physical Computing HS21

# 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.

Source: https://mqtt.org/

# 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

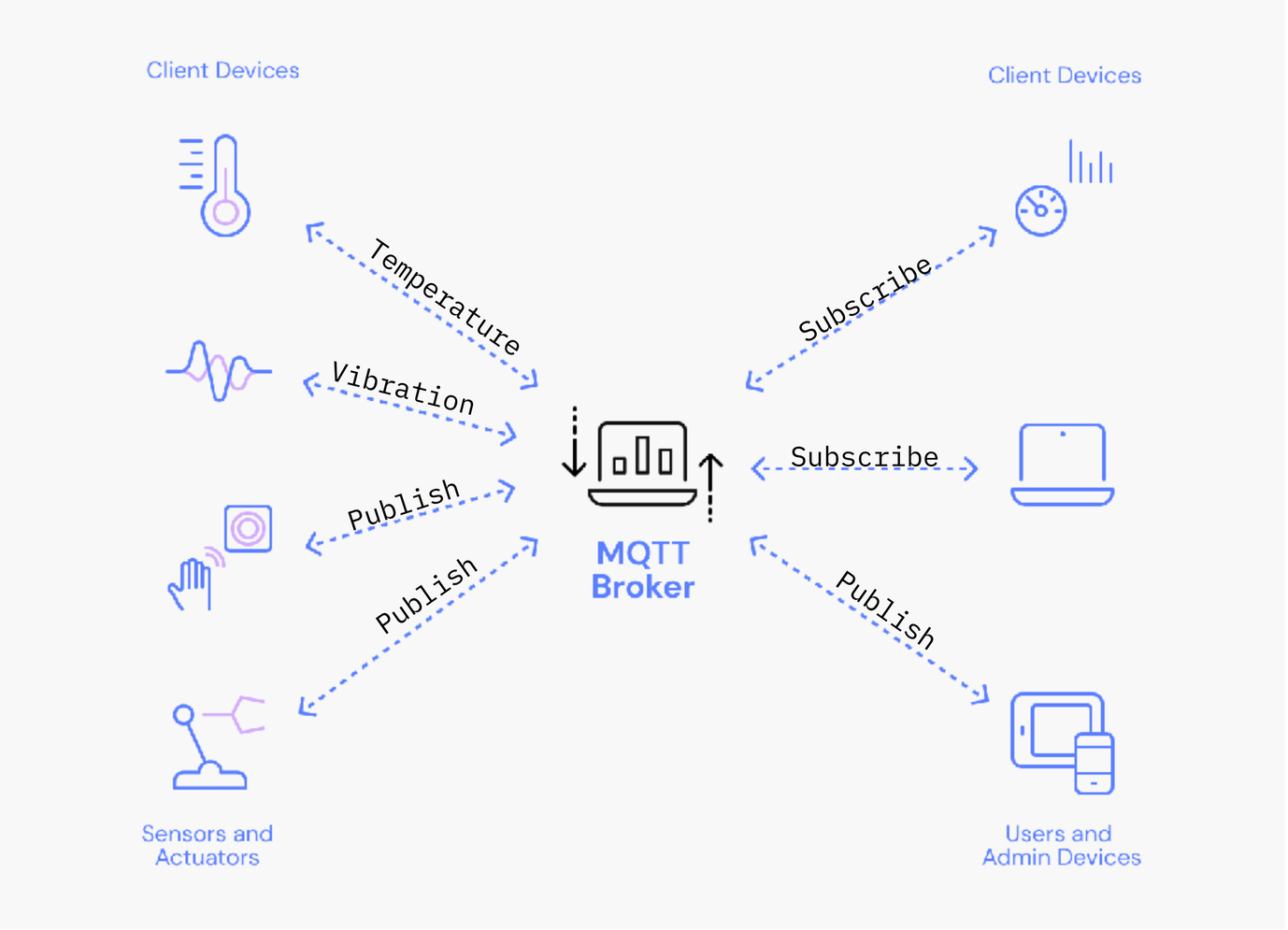
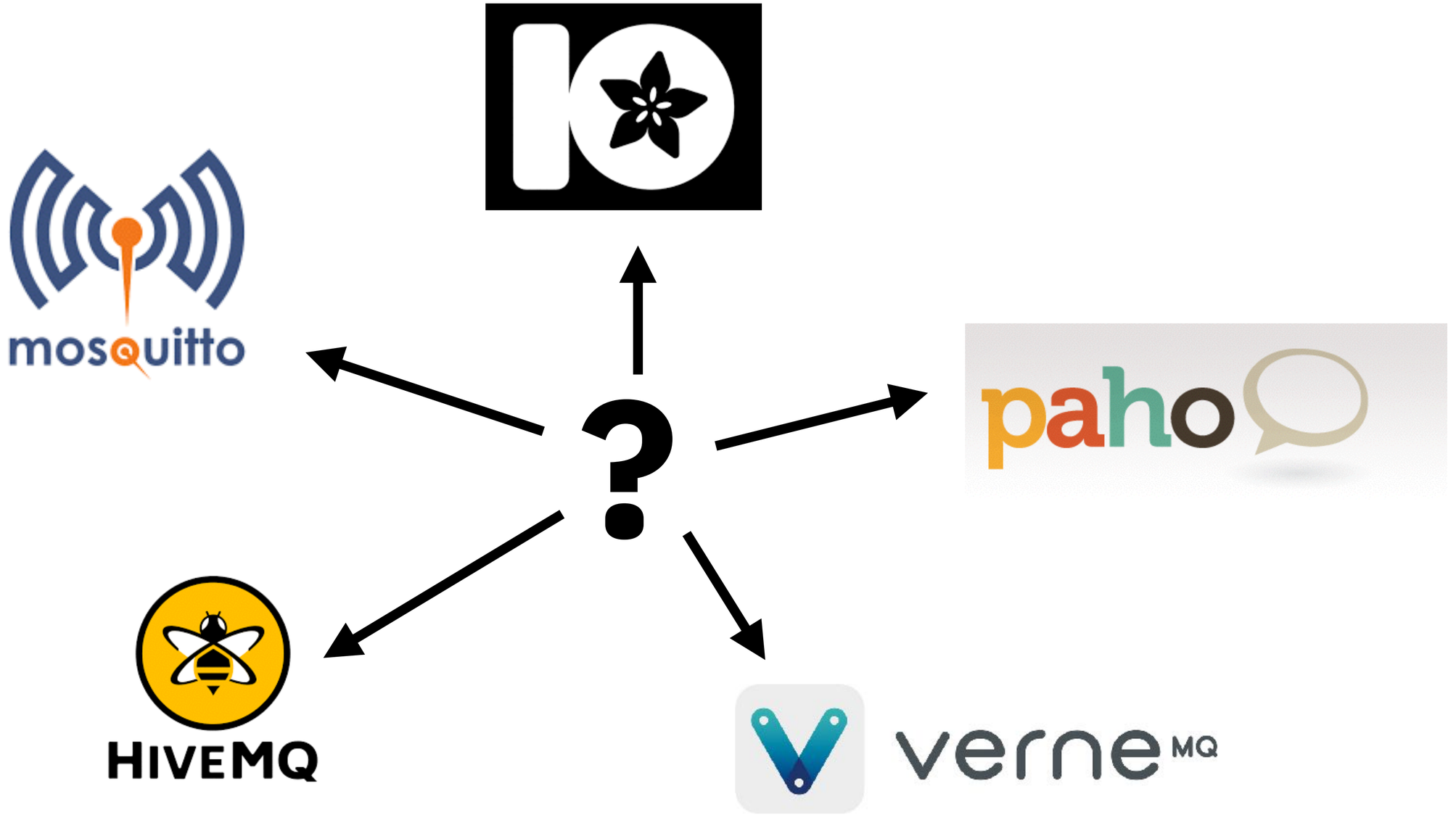



Image Source

# MQTT BROKERS



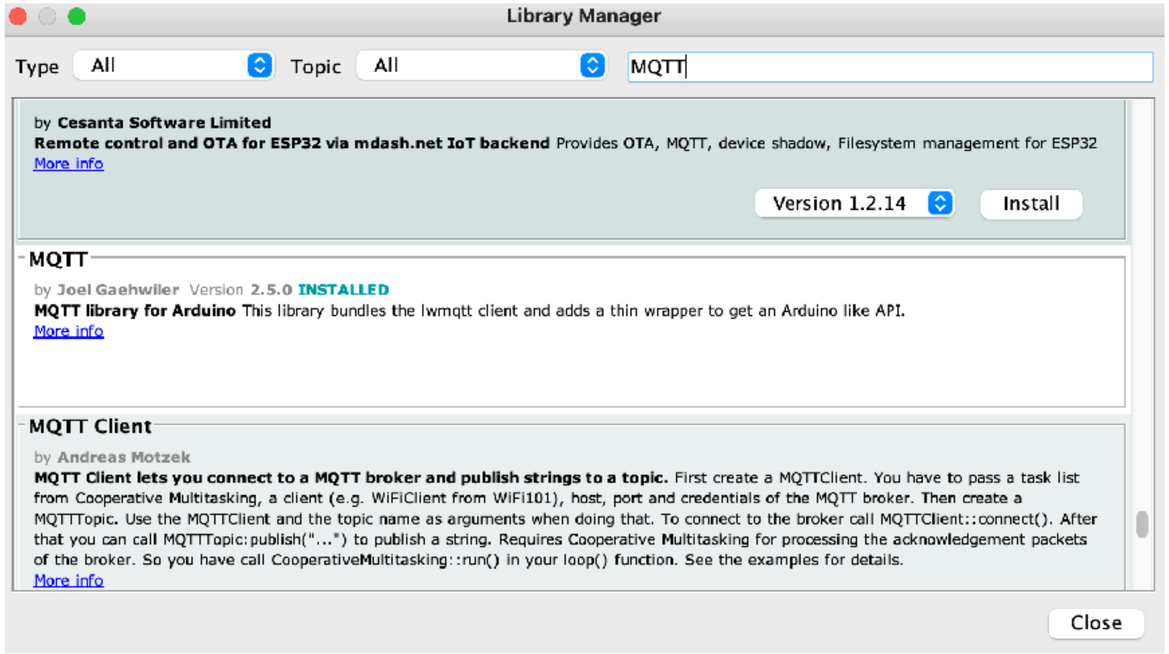
# SHIFTR.IO


- Developed at ZHdK in 2015 by Joël Gähwiler as his MA project.
- Open-source and free to use (except for heavy users)
- Cloud or desktop-based (more bandwidth, and less latency)

Include .zip Library:

**https://github.com/256dpi/arduino-mqtt/releases**

Or search for MQTT by Joel Gäwihler in Library Manager:



# SHIFTR.IO

**client.connect("arduino", "p5toArduino", “xxxxxxxxx")**

1.The client ID,displayed as the connections name
in the real-time graph.
2.The name of your instance.
3.The token secret as configured in the settings panel.

**client.begin("p5toArduino.cloud.shiftr.io", net);**

1.Your instance domain.
2.Depends on the chosen network client. Use net.

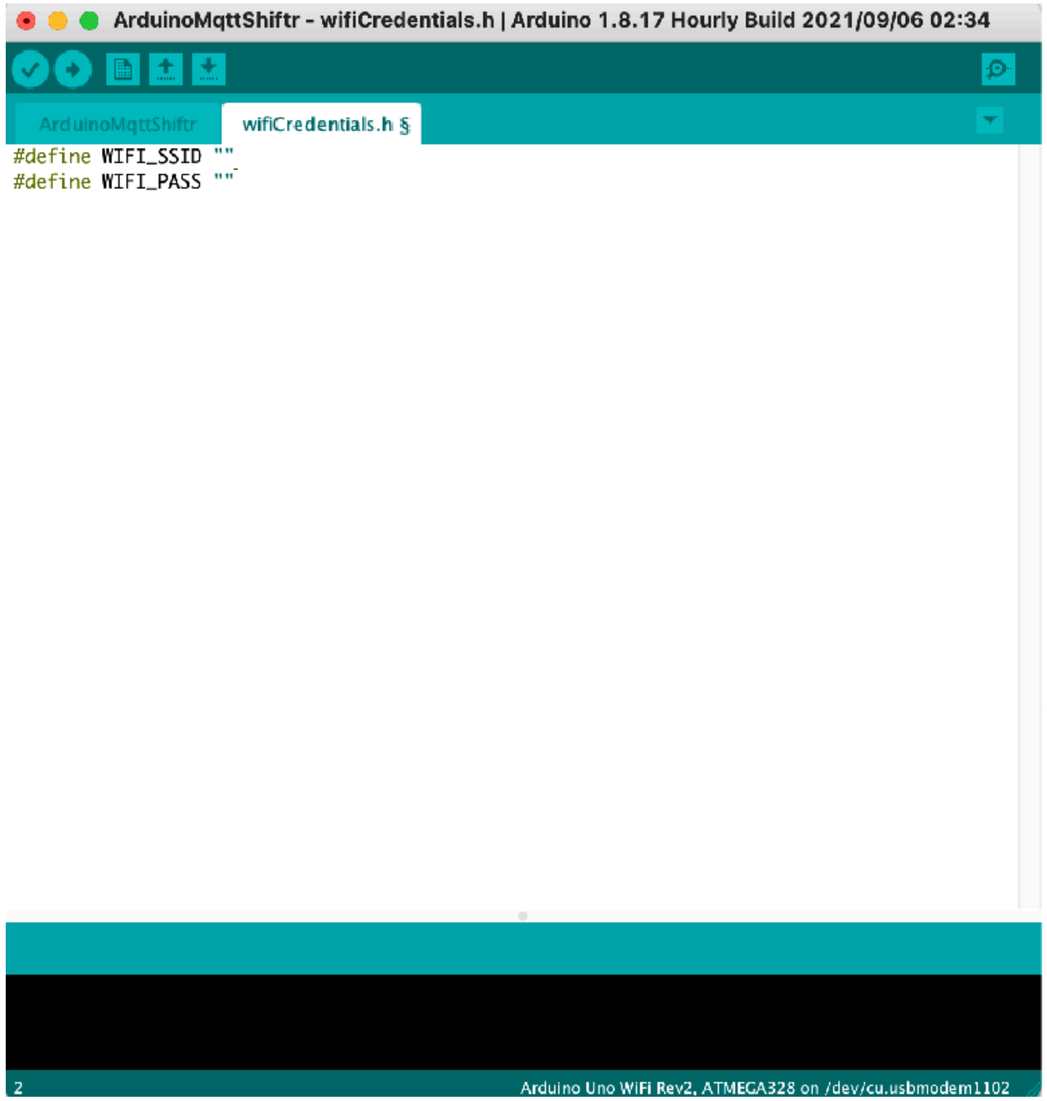
```
client.subscribe(“xValue");
```

1.The name of the topic to subscribe.

```
client.publish("yValue", “0.01”);
```

1.The topic to publish the message to.
2.The payload of the message.

# SHIFTR.IO




WIFI_SSID “zhdkIAD”
WIFI_PASS “i@dZHDK2021”

# SHIFTR.IO

```
let creds = {
    clientID: 'p5',
    userName: 'p5ToArduino',
    password: ‘xxxxxx'
}

let broker = {
    hostname: 'arduinozhdk.cloud.shiftr.io',
    port: 443
};
```

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.

```
client.subscribe(“yValue”);

client.send(“2.3”);
```

MQTT messages are send/received as **payload**, that is the essential data that is being carried within a packet or other transmission unit.

```
{
    "state":"delivered",
    "data":
        {
            "message":"Hello, world!"
        }
}
```

# EXERCISE

Move the circle from p5Mqtt example file using the values from a sensor in your project box(either gyroscope or distance)