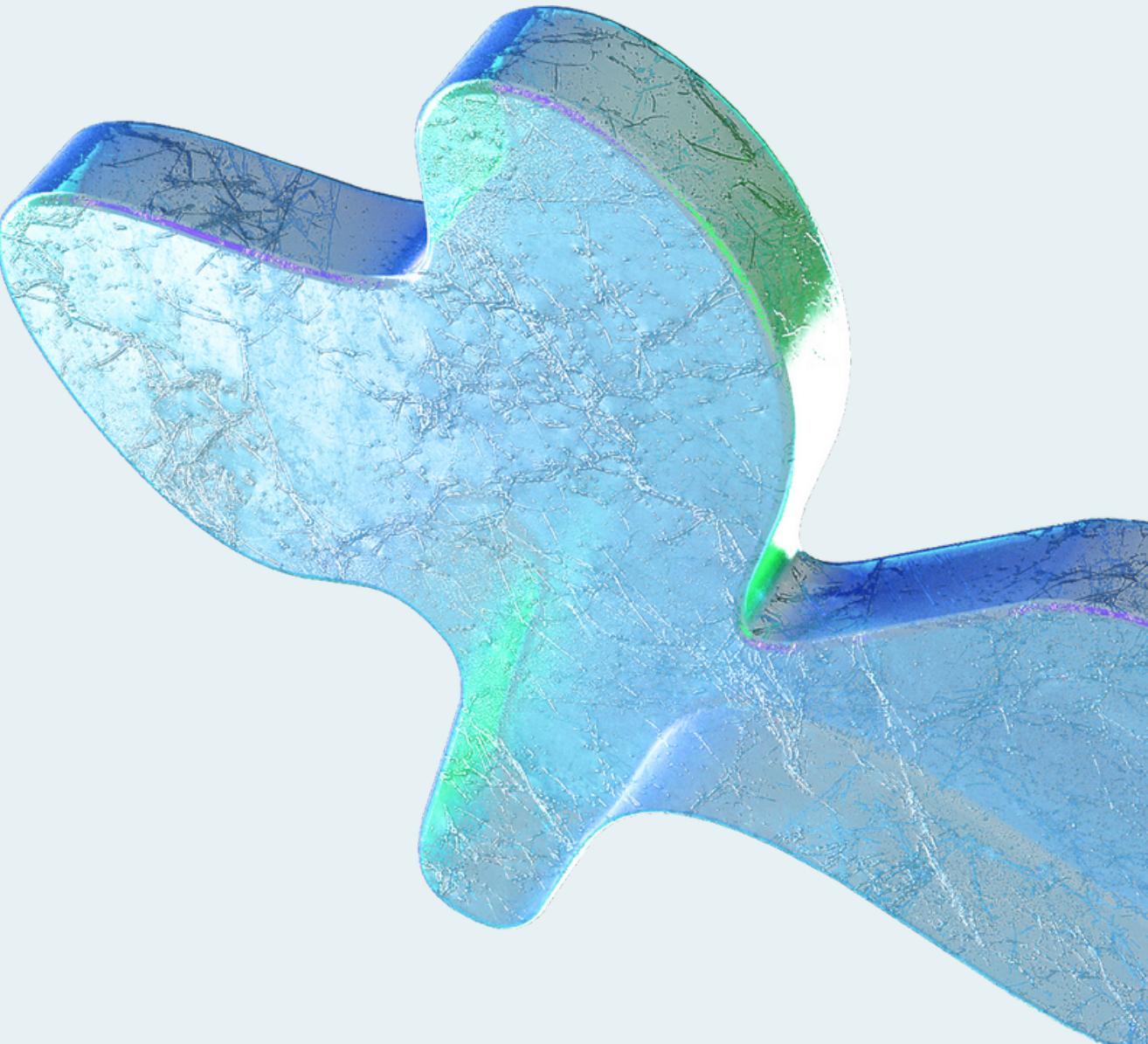


IoT Bridge Platform



Mariana Ramos, University of Bologna, Internet of Things

Professors: Luciano Bononi, Marco Di Felice

June 15, 2022

Overview

Project Scope

Protocols

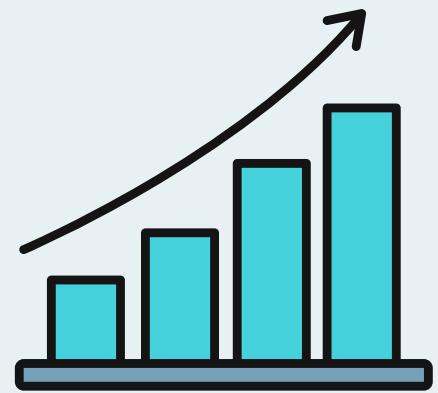
Visualization, Aggregation,
Storage and Bridging

Results

Implementation Details

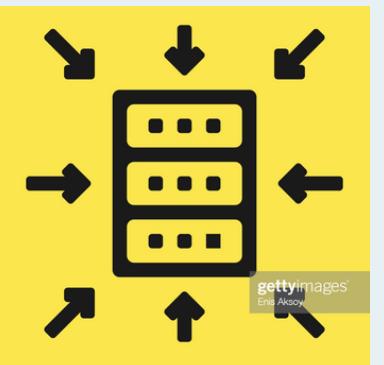
Demo

Project Scope



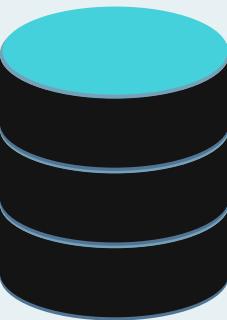
Visualization

View data from 3 different protocols. JSON document, with fields sensor type and value where the latter is the measured value produced by the sensor.



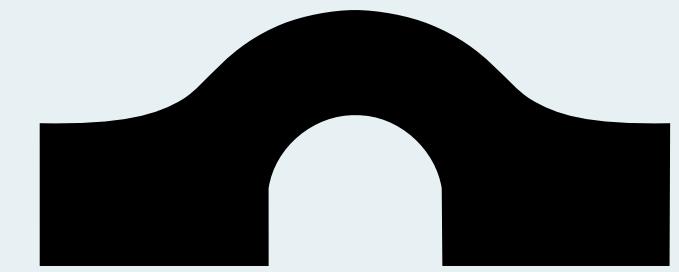
Aggregation

Print the statistical features of the sensory data computed every n observations. Maximum, minimum, average, standard deviation



Storage

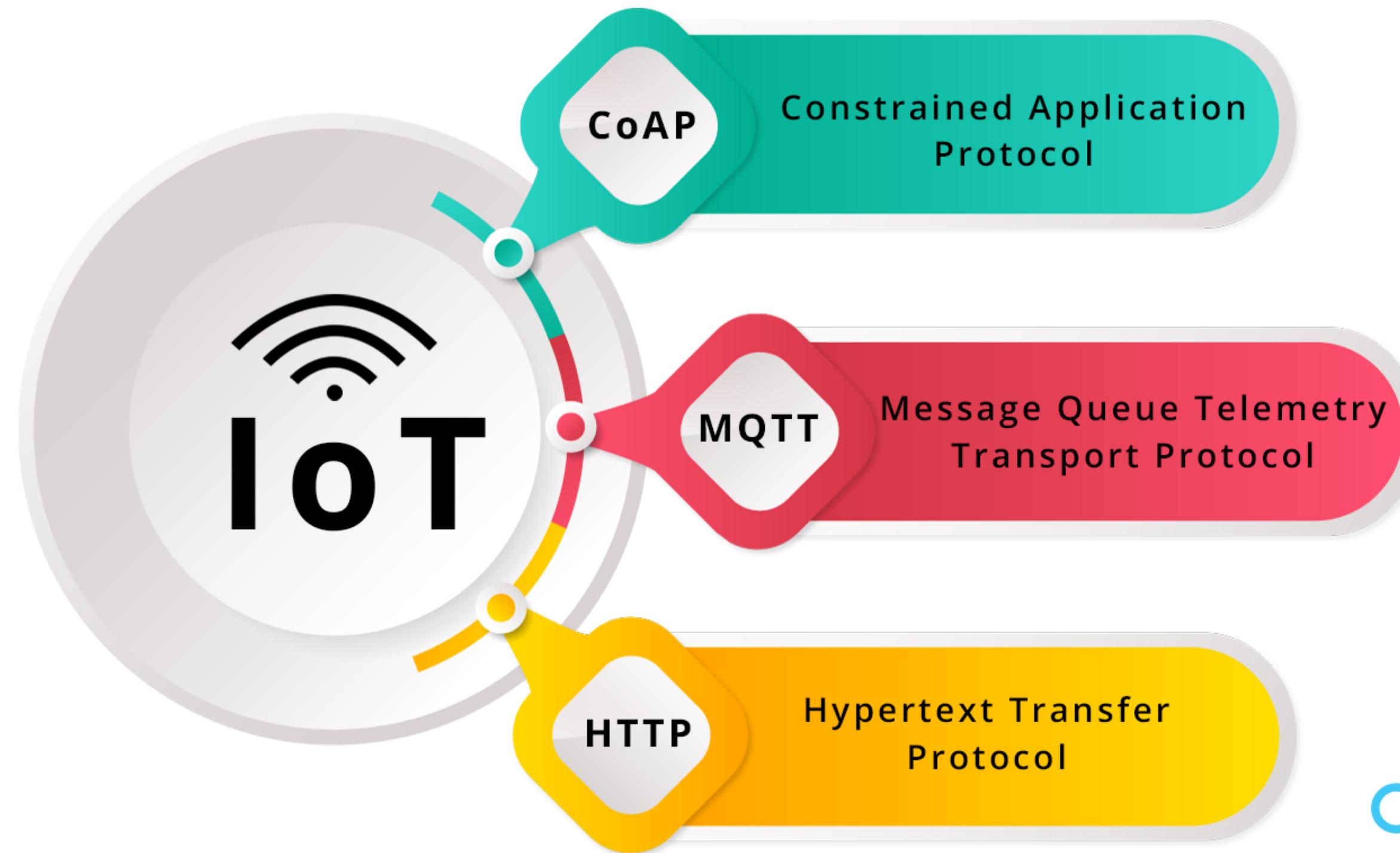
Save data from 3 different protocols in Influx 2.0. Query in Grafana.



Bridging

Convert an input data stream in one protocol to an output data stream other than the capture one. External clients can connect to the output stream regardless of the tool.

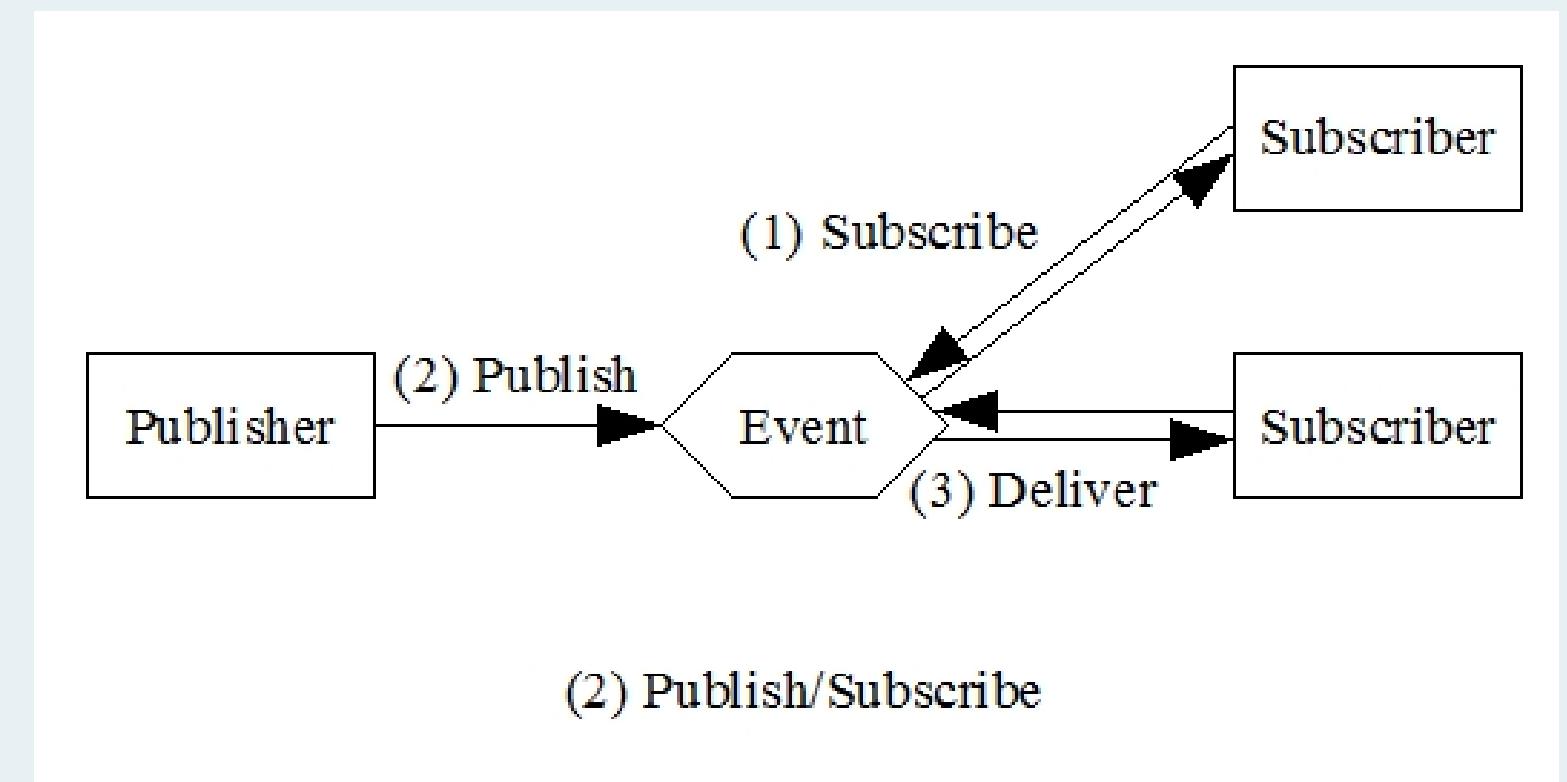
IoT Protocols



Visualization and Aggregation

MQTT

- Publish-Subscribe protocol
- On a temporal level, the publishing/subscribing phases are separate.
- Subscribe to a topic.
- Visualize the JSON responses corresponding to that topic.
- The values are displayed on the screen each time they are received, in real-time
"topic=temperature"

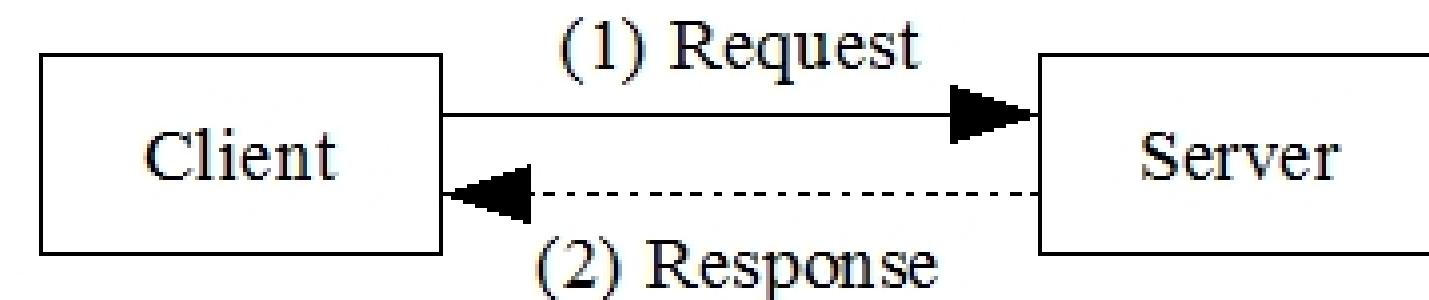


```
let readableStream = new Stream.Readable({
  objectMode: true
});
```

Visualization and Aggregation

COAP and HTTP

- Request-Reply protocol
- Roles are divided into Server/Client. The server satisfies client requests individually
- Request value using GET
- Wait for the response
- Wait for a fixed number of seconds and perform another GET.



(1) Request/Reply

```
let eventEmitterCoap = new EventEmitter()
```

"GET /temperature"

Storage

influxdb-client-javascript



Requirements

For any protocol it is necessary to specify:

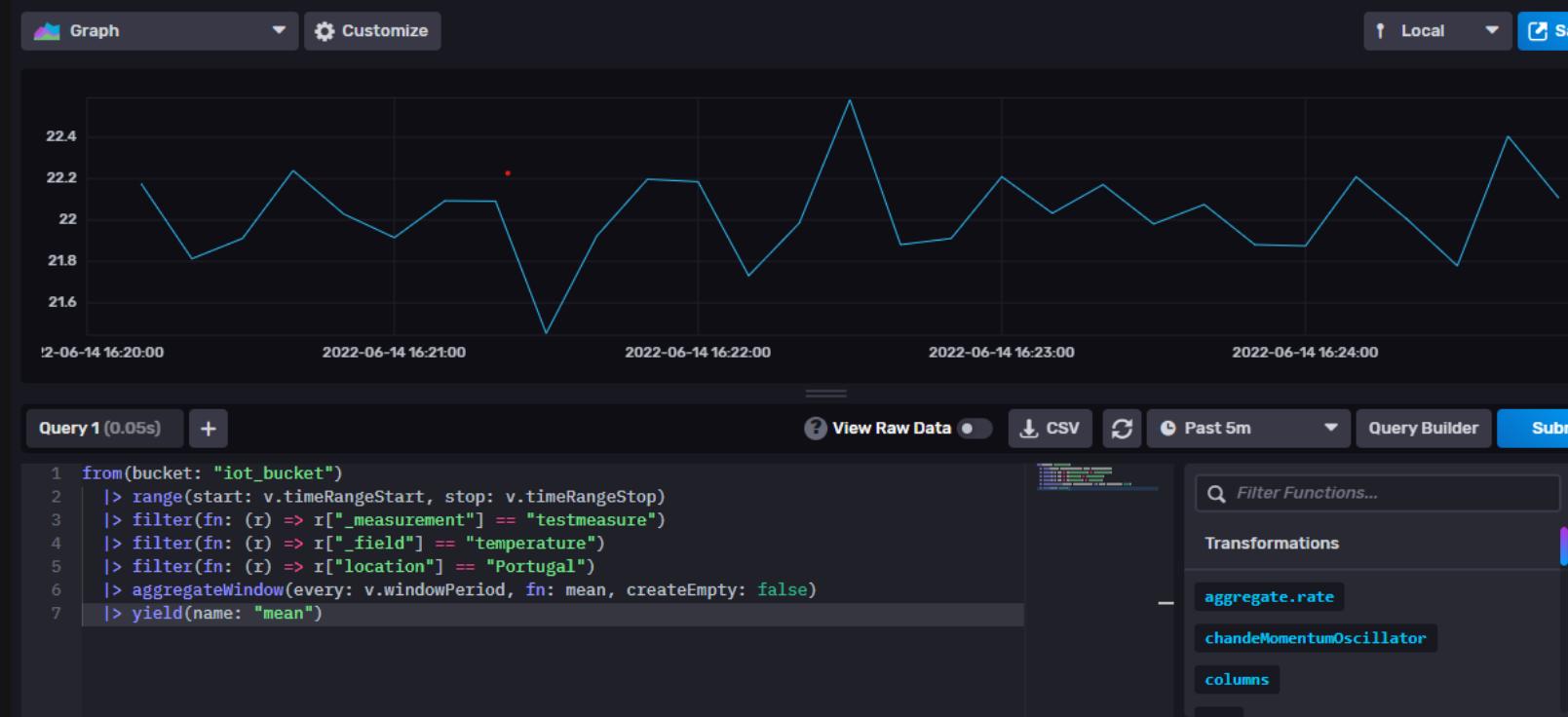
- Token
- Organization
- Name of the bucket
- URL where the influx 2.0 db is running
- Measurement Tag
- Field
- Additional tags

```
const client = new InfluxDB({ url: this.url, token: this.token })
const writeApi = client.getWriteApi(this.org, this.bucket)
```

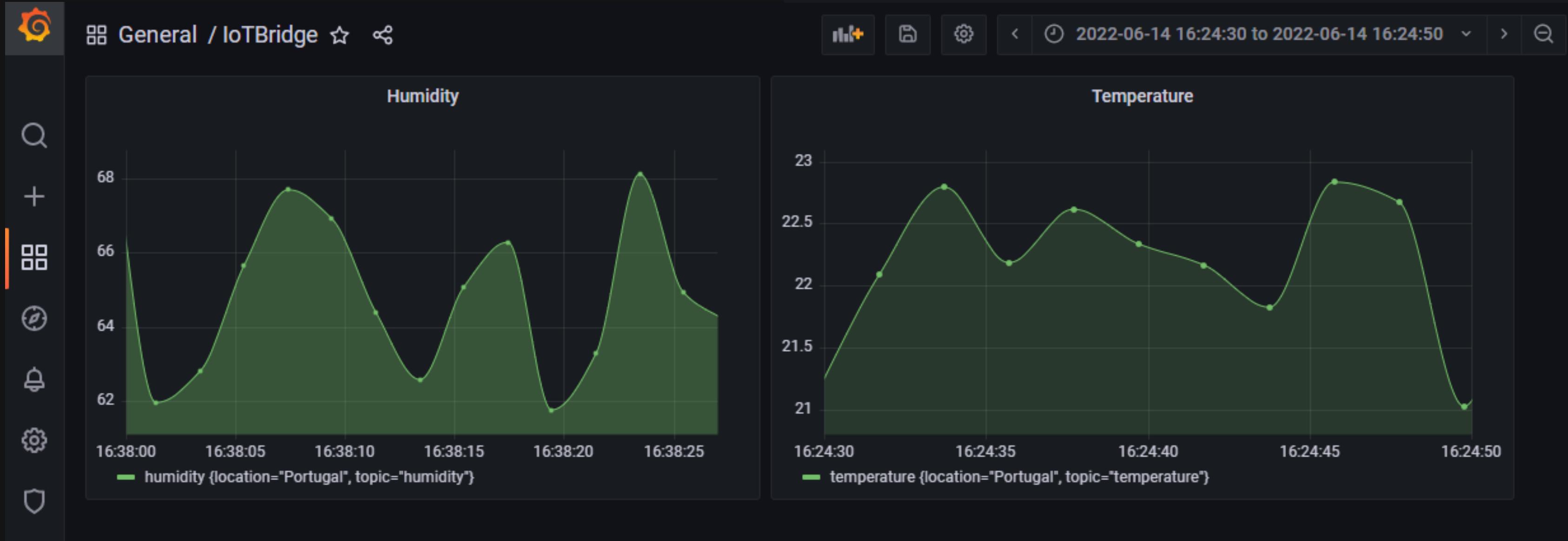
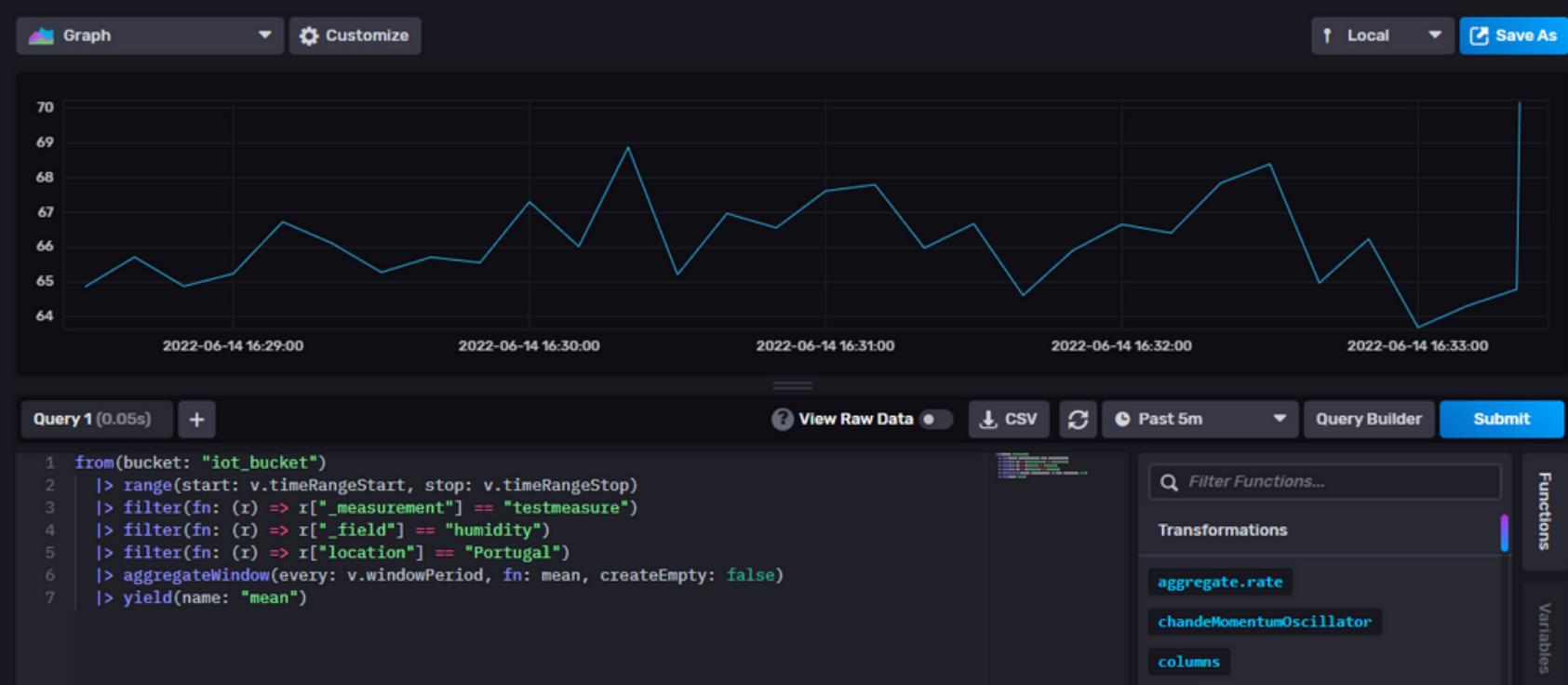
```
let point = new Point(measurement).floatField(field, parseFloat(value))

writeApi.writePoint(point)
```

Data Explorer



Data Explorer



Data Explorer

Graph ▾ Customize

Local ▾

Save As



Query 1 (0.05s)



View Raw Data

CSV



Past 5m

QueryBuilder

Submit

```
1 from(bucket: "iot_bucket")
2   |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
3   |> filter(fn: (r) => r["_measurement"] == "testmeasure")
4   |> filter(fn: (r) => r["_field"] == "temperature")
5   |> filter(fn: (r) => r["location"] == "Portugal")
6   |> aggregateWindow(every: v.windowPeriod, fn: mean, createEmpty: false)
7   |> yield(name: "mean")
```



Filter Functions...

Transformations

aggregate.rate

changeMomentumOscillator

columns

Functions

Variables

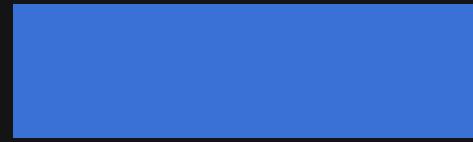
Bridging

1



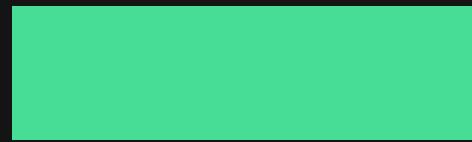
HTTP to COAP
bridging

2



COAP to HTTP
bridging

3



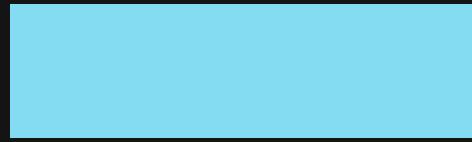
COAP to MQTT
bridging

4



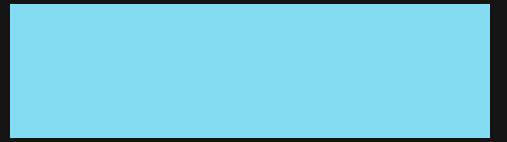
HTTP to MQTT
bridging

5



MQTT to COAP
bridging

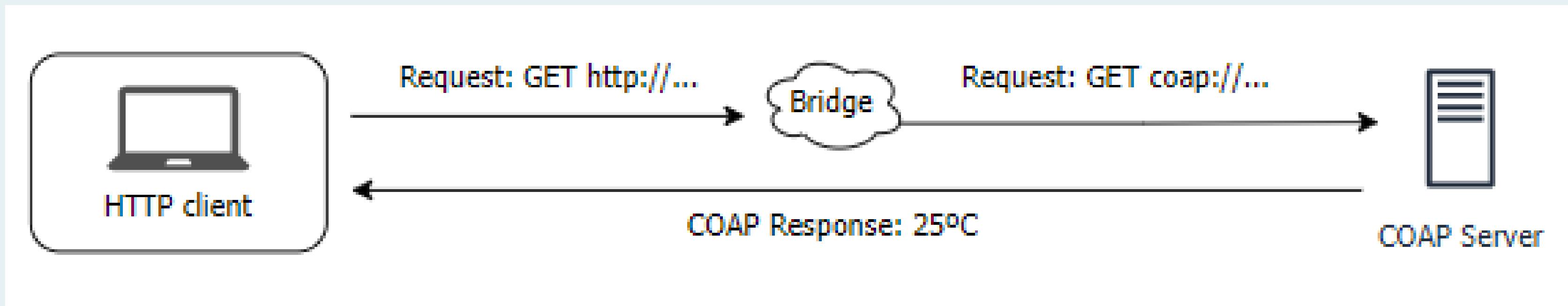
6



MQTT to HTTP
bridging

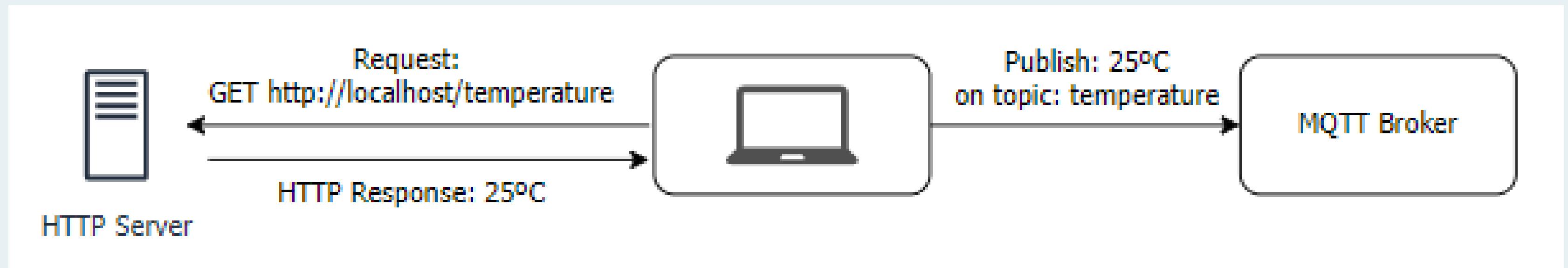
Bridging - Between COAP and HTTP

Translation from COAP to HTTP. Each request to an HTTP server is forwarded to a COAP server.



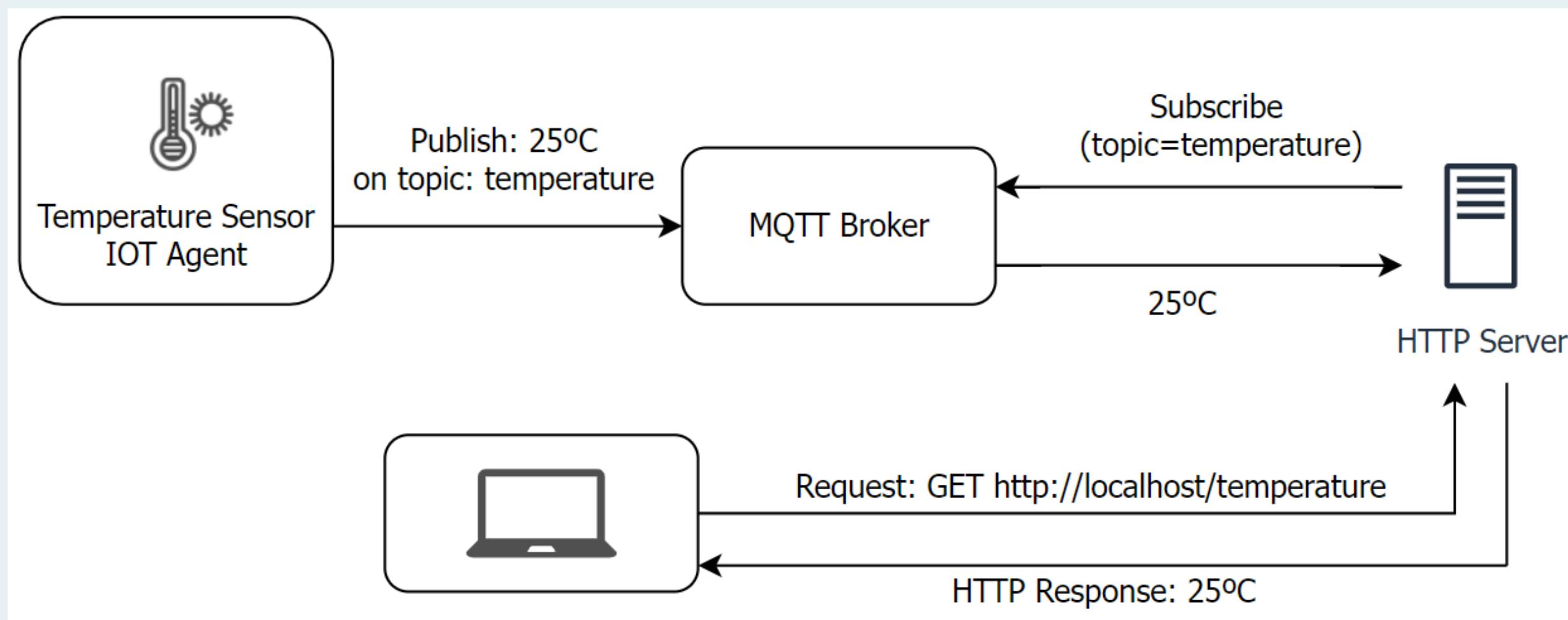
Bridging - To MQTT

Translation from HTTP to MQTT. The response obtained by HTTP response is published to a MQTT broker having the topic equal to the path of the request.



Bridging - From MQTT

Example: an ESP 32 board publishes the humidity and temperature values as topic 'temperature' and 'humidity'. If the conversion from MQTT to HTTP is used and you listen on topic 'temperature', the following URL will be created to which you shall connect: <http://localhost:3001/temperature>



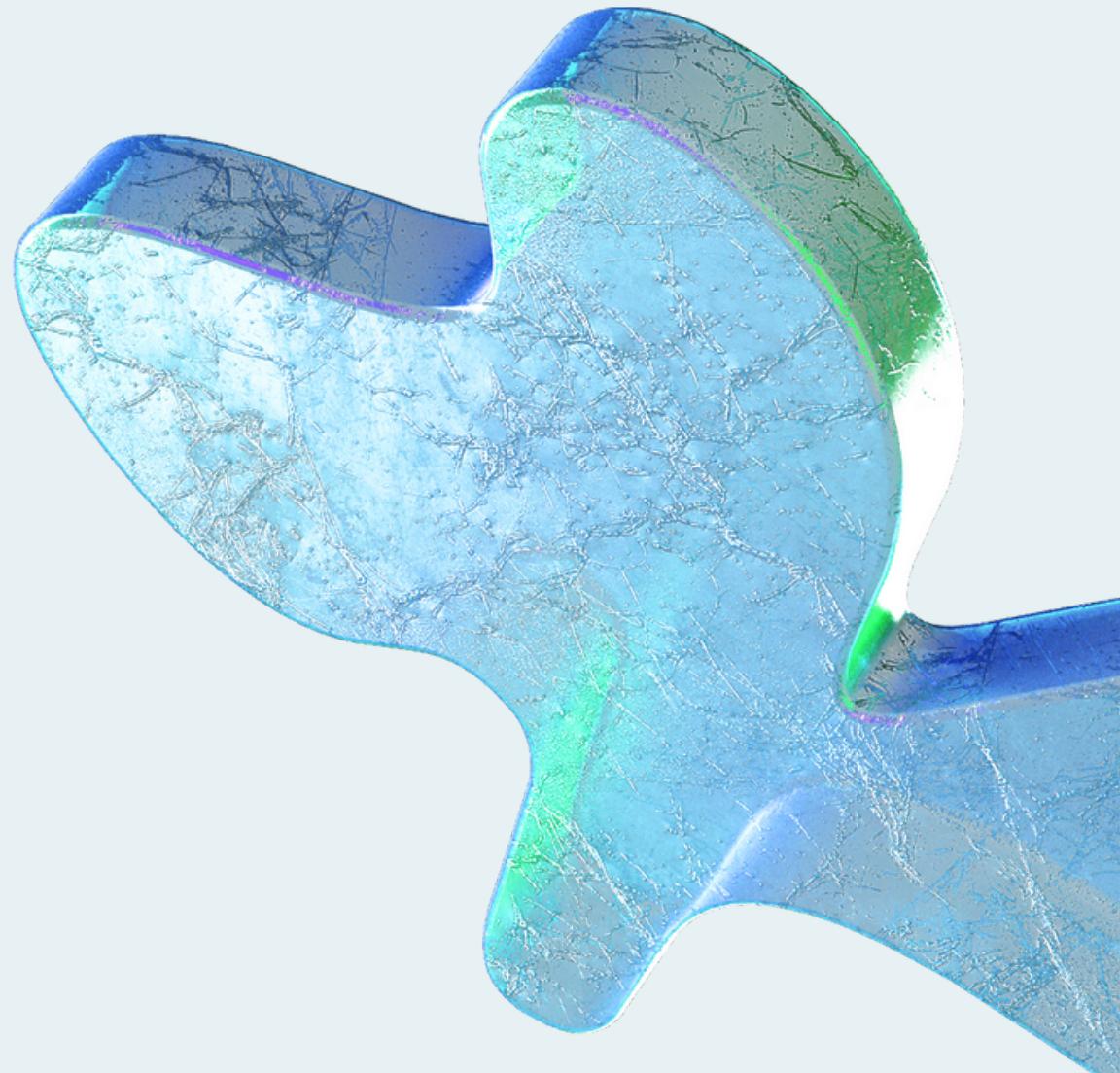
Implementation Details

Programming language Javascript with a node.js environment

To interact with it, the user can enable its functionalities through a CLI (Command Line Interface).

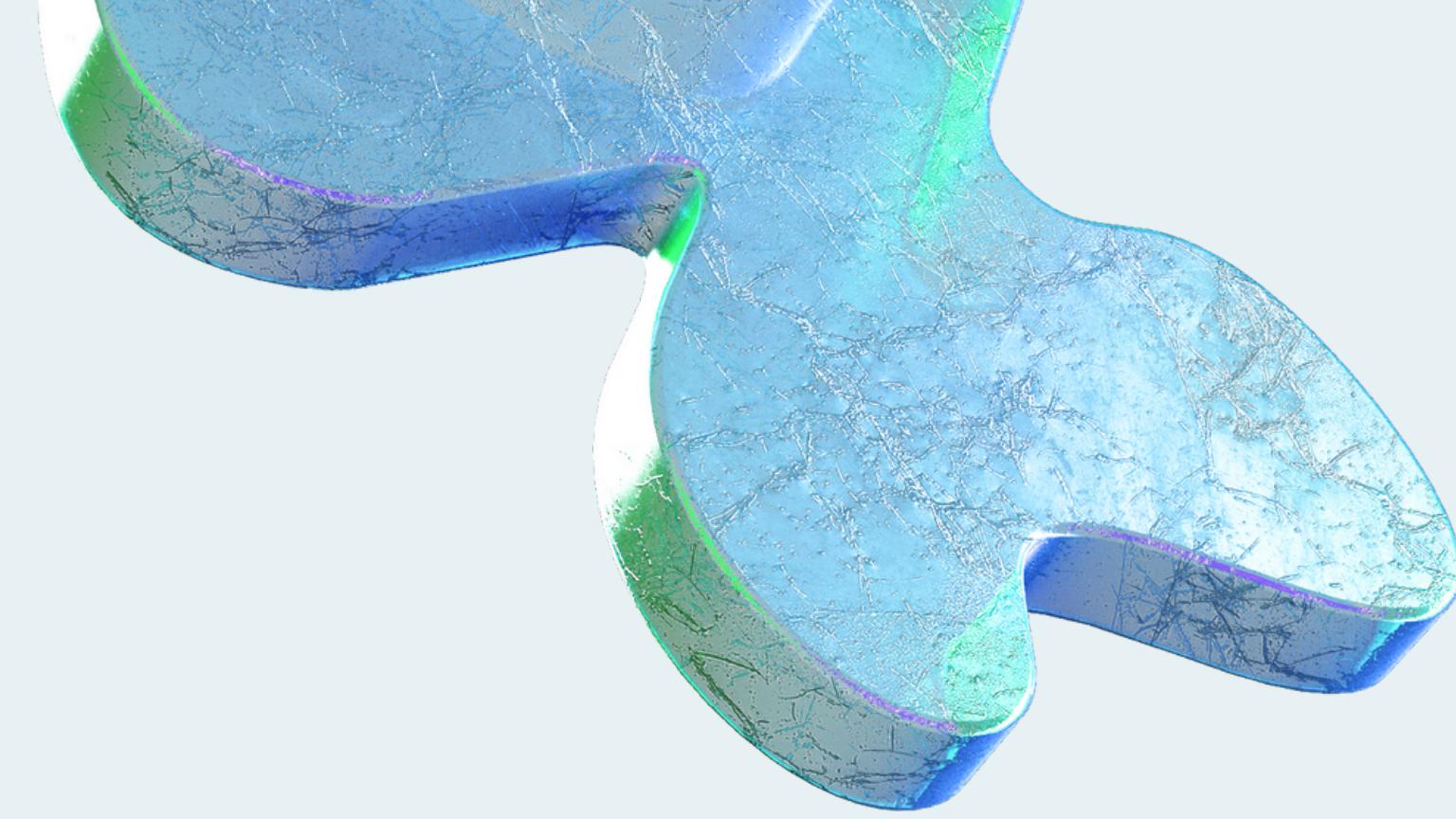
Libraries used:

- coap
- mqtt
- axios
- express
- telegraf
- yargs



A photograph of a person's hands typing on a laptop keyboard. The hands are positioned on the right side of the keyboard, with fingers pressing keys. To the left of the laptop, there is a clear glass filled with water and ice cubes, and above it, a white paper cup with a lid. The background is slightly blurred, showing a light-colored wall and some furniture.

DEMO



Thank you

Mariana Ramos, University of Bologna, Internet of Things

Professors: Luciano Bononi, Marco Di Felice

June 08, 2022