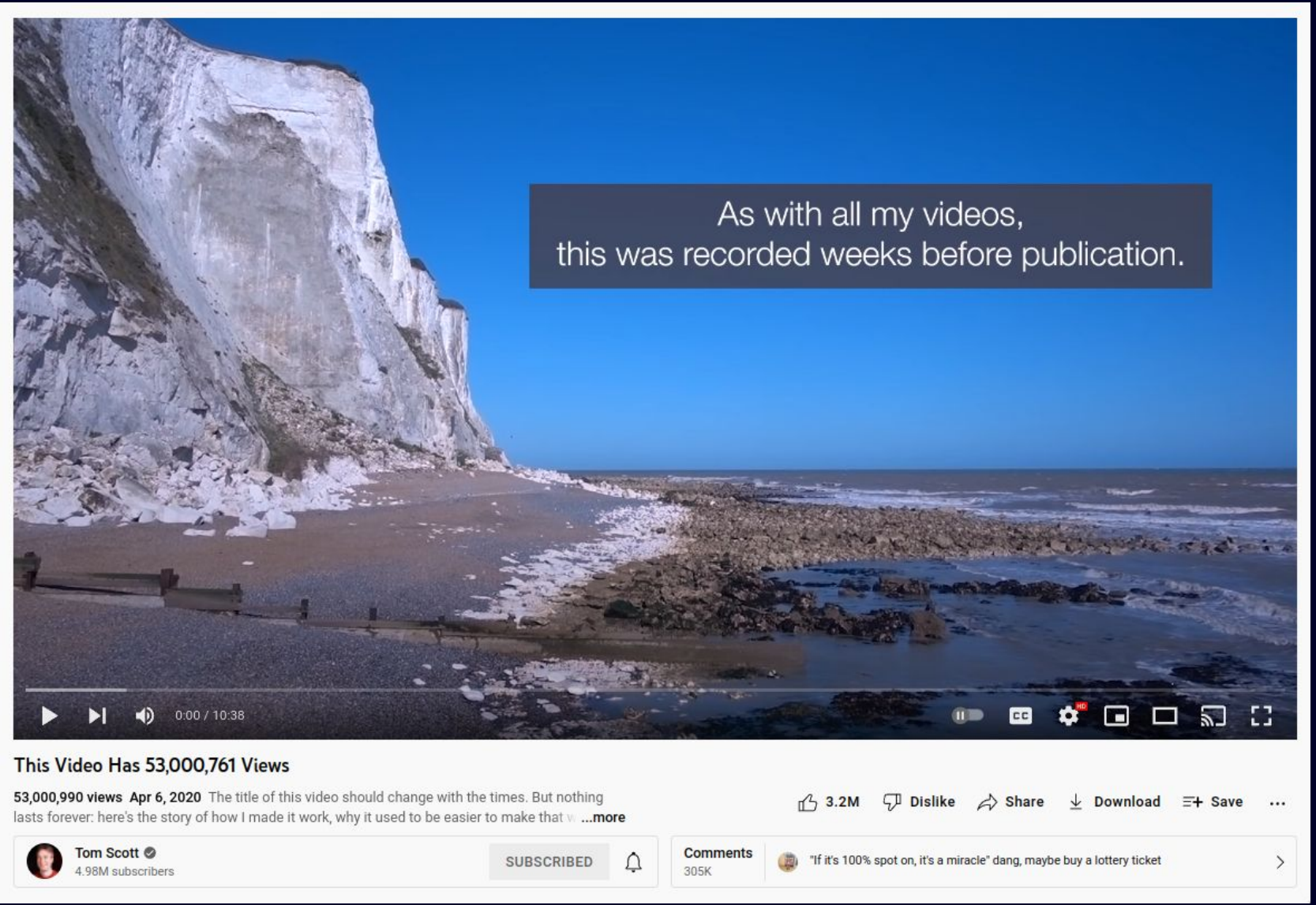
Bringing the Power of the Web to the IoT: Web of Things and node-wot

Ege Korkan - Global Summit for Node.js'22

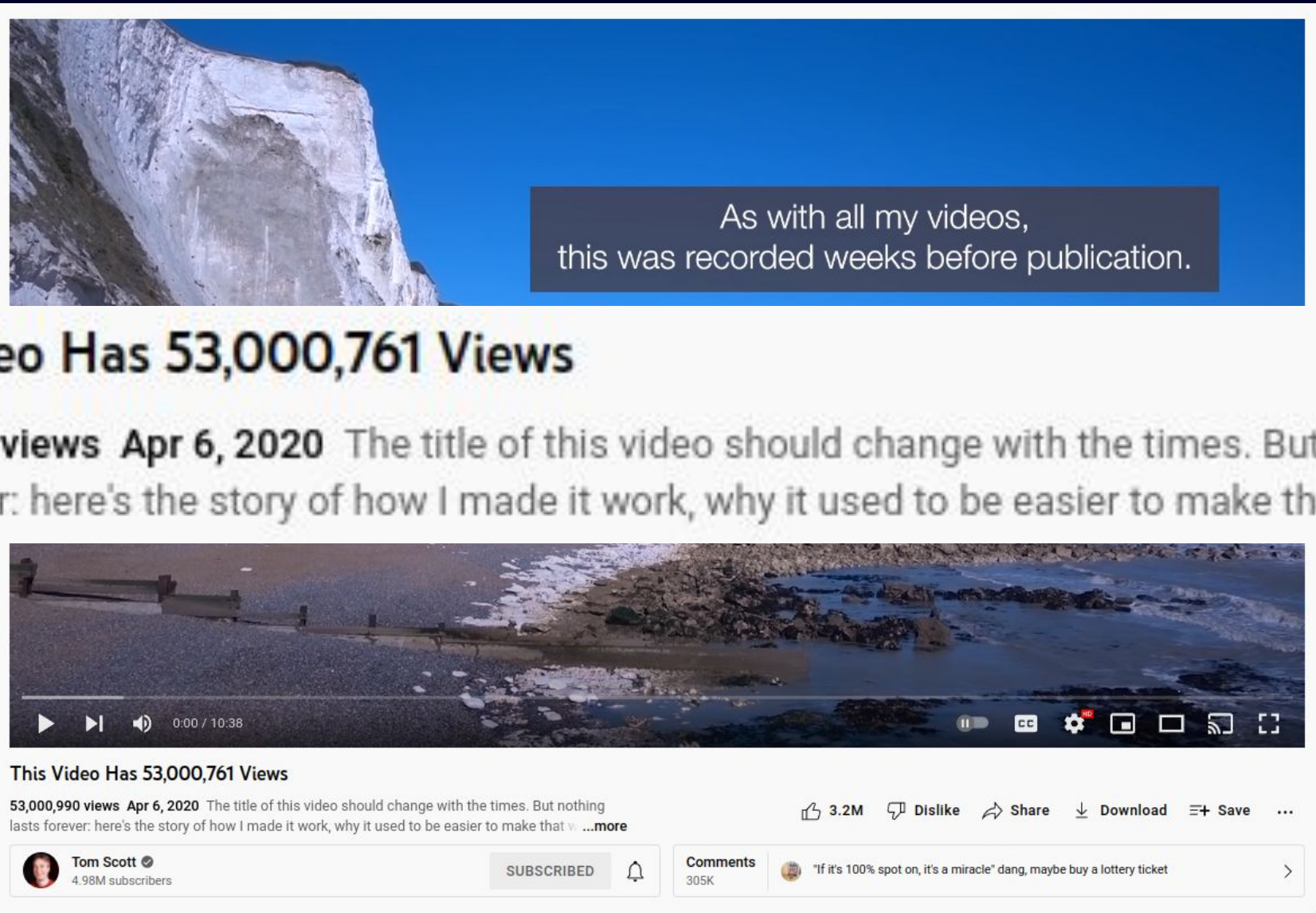
A Peculiar YouTube Video Title



https://www.youtube.com/watch?v=BxV14h0kFs0

SIEMENS

A Peculiar YouTube Video Title



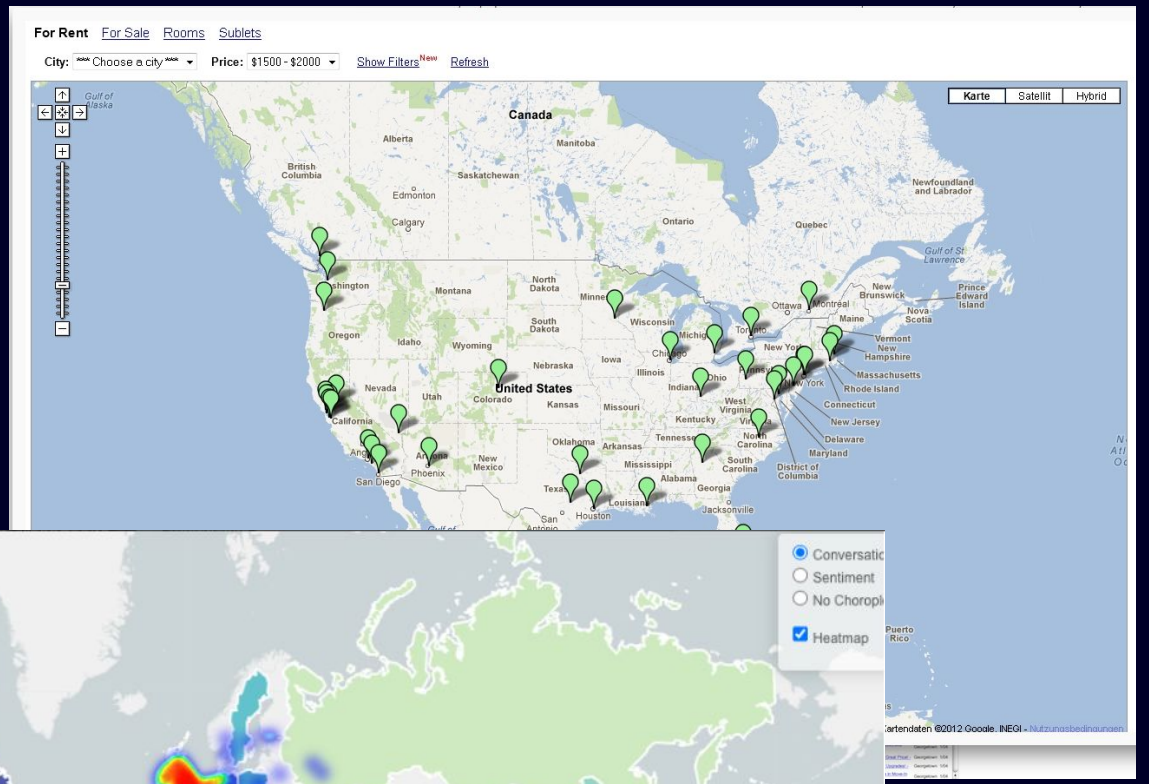
https://www.youtube.com/watch?v=BxV14h0kFs0

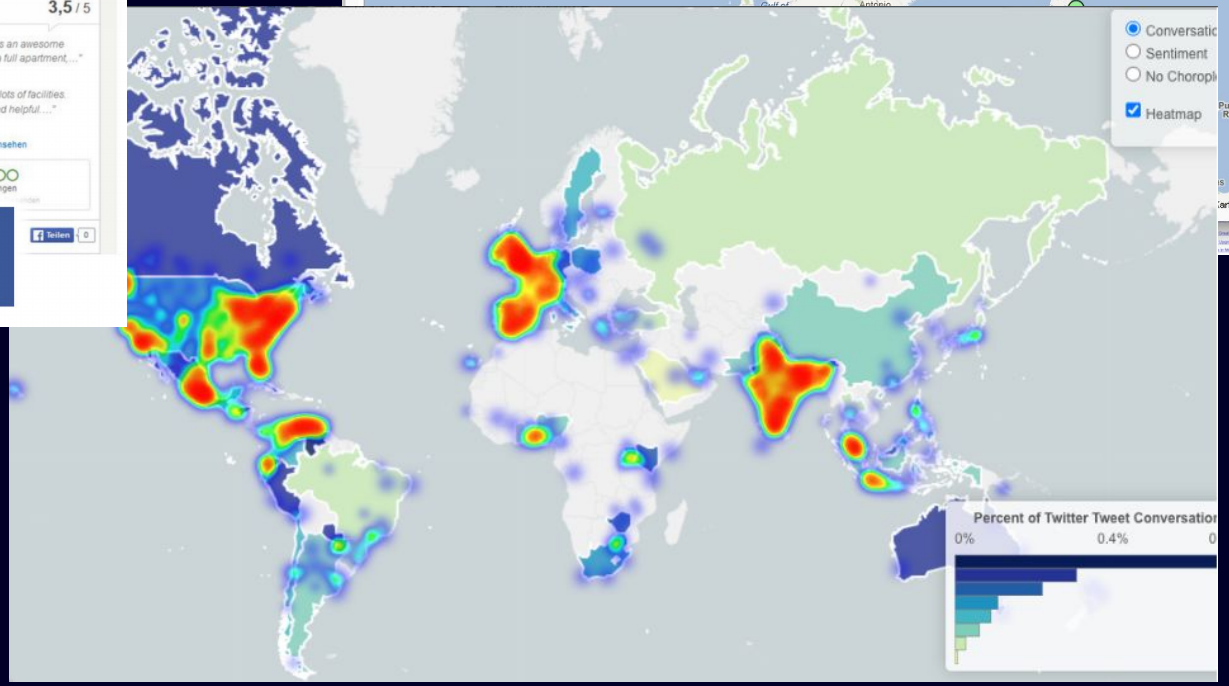
SIEMENS

Web Mashups?

http://www.housingmaps.com/

https://www.expedia.com/





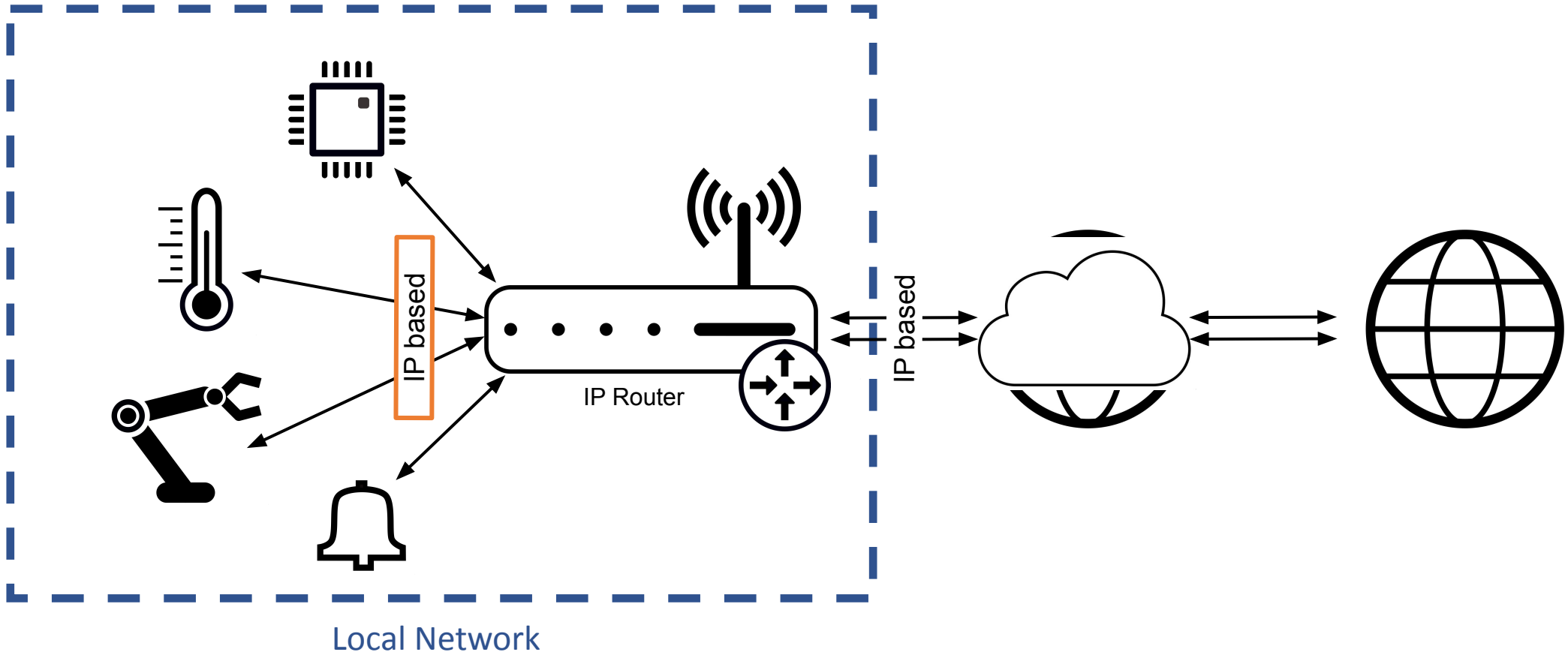
https://www.trendsmap.com/

SIEMENS

What is IoT?

(at least in this talk)

At least one way to look at it



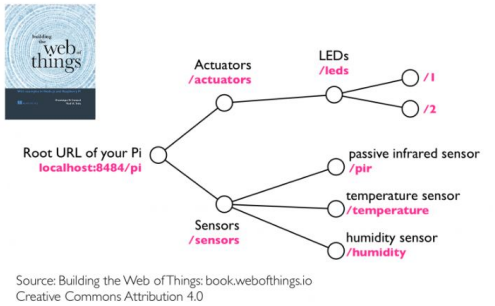
SIEMENS

The Web of Things way to look at it

Source: Building the Web of Things: book.webofthings.io
Creative Commons Attribution 4.0

SIEMENS

The Web of Things way to look at it



Source: Building the Web of Things: book.webofthings.io
Creative Commons Attribution 4.0

Was a proposal on how to build REST APIs for IoT devices

Is about describing *any* kind of API for IoT devices using *any* protocol

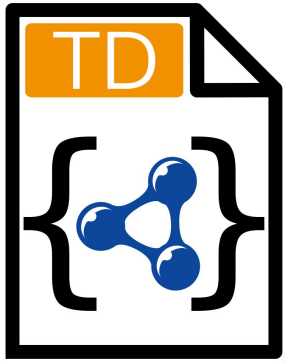
SIEMENS

Now 4 normative deliverables, 93 participants over 35 organizations!

SIEMENS

Web of Things: Thing Description (TD)

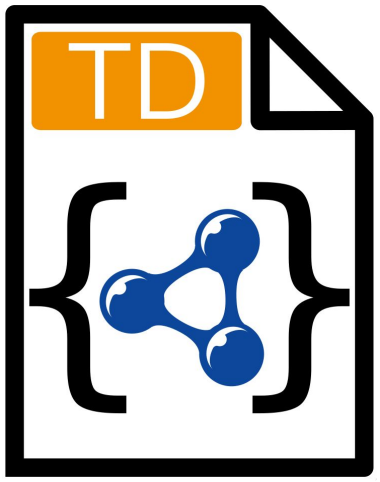
1. Abstract description of network APIs of IoT devices
   – Property, Action, Event
2. Semantic annotation of capabilities by other vocabularies over the Web: schema.org, SSN, ...
3. Concrete protocol description of individual devices



SIEMENS

Why talk about standards in a Node.js summit?

Because it enables



Abstraction

Design Pattern

Protocol Agnostic Programming

Well also, the reference implementation is in Node.js

thingweb.io

Eclipse Thingweb node-wot

sane-city (Java)

This implementation was strongly inspired by node-wot.

dart_wot (Dart)

dart_wot is an implementation of the Web of Things Scripting API modelled after the WoT reference implementation node-wot.

wot-py (Python)

Inspired by the exploratory implementations located in the thingweb GitHub page.

Let’s see some code!

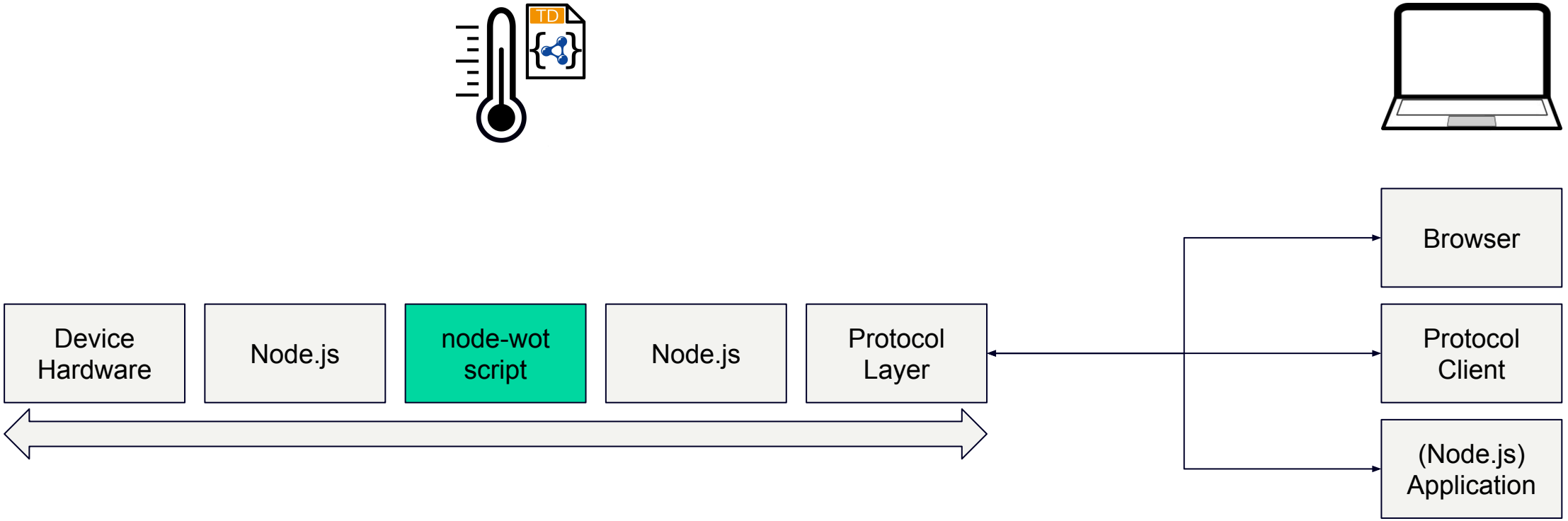
Code of a Thing

```
thing.setPropertyReadHandler("temperature", async () => {
    return mySensor.temperature.read();
});

thing.setActionHandler("increaseTemperature", async () => {
    myHeater.turnOn();
    setTimeout(() => {
        myHeater.turnOff();
    }, "1000")
    return true;
});


```

What happens in the background

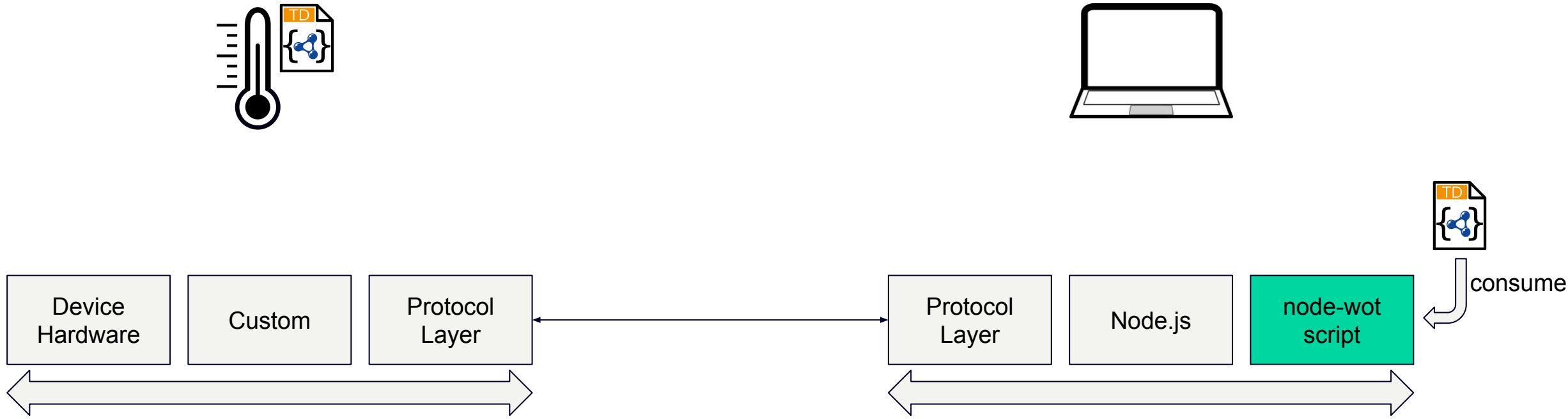


SIEMENS

Code of a Thing Consumer

```
1   const thing = await WoT.consume(td);
2
3   setInterval(async ()=>{
4       const temperatureReading = await thing.readProperty("temperature")
5       const temperatureValue = await temperatureReading.value();
6
7       if (temperatureValue < 20){
8         await thing.invokeAction("increaseTemperature")
9       }
10
11  },10000)
```

What happens in the background



SIEMENS

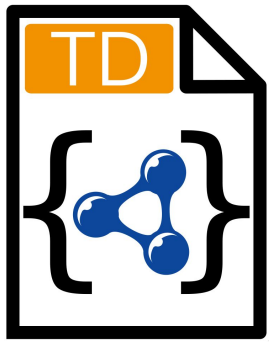
How is this all possible?

First: Thing Descriptions

SIEMENS

Deeper look into the Thing Description

Standardized JSON (-LD) document



W3C Recommendation

TABLE OF CONTENTS

Web of Things (WoT) Thing Description

W3C Recommendation 9 April 2020 (Link errors corrected 23 June 2020)

This version:
https://www.w3.org/TR/2020/REC-wot-thing-description-20200409/

Latest published version:
https://www.w3.org/TR/wot-thing-description/

Latest editor's draft:
https://w3c.github.io/wot-thing-description/

Implementation report:
https://w3c.github.io/wot-thing-description/testing/report.html

Previous version:
https://www.w3.org/TR/2020/PR-wot-thing-description-20200130/

Editors:
Sebastian Kaebisch (Siemens AG)
Takuki Kamiya (Fujitsu Laboratories of America)
Michael McCool (Intel)
Victor Charpenay (Siemens AG)
Matthias Kovatsch (Huawei)

Participate:
GitHub w3c/wot-thing-description

SIEMENS

Deeper look into the Thing Description: **Thing Level**

```
{
    "@context": [
                "https://www.w3.org/2022/wot/td/v1",
                {"iot":"http://iotschema.org"}
    ],
    "@type":"SprinklerStation",
    "id": "urn:dev:ops:Agriculture-7331",
    "title": "SoilStation",
    "description": "A soil management station for use in agriculture",
    "securityDefinitions": { "basic_sc": {"scheme": "basic"} },
    "security": "basic_sc",
    "properties": {
        "temperature": {...}
    },
    "actions": {
        "startSprinkler": {...}
    },
    "events": {
        "tooDry": {...}
    },
    "links":["href":"docs", "rel":"about", "type":"text/html"]
}
```

Thing Metadata

Semantic Annotations

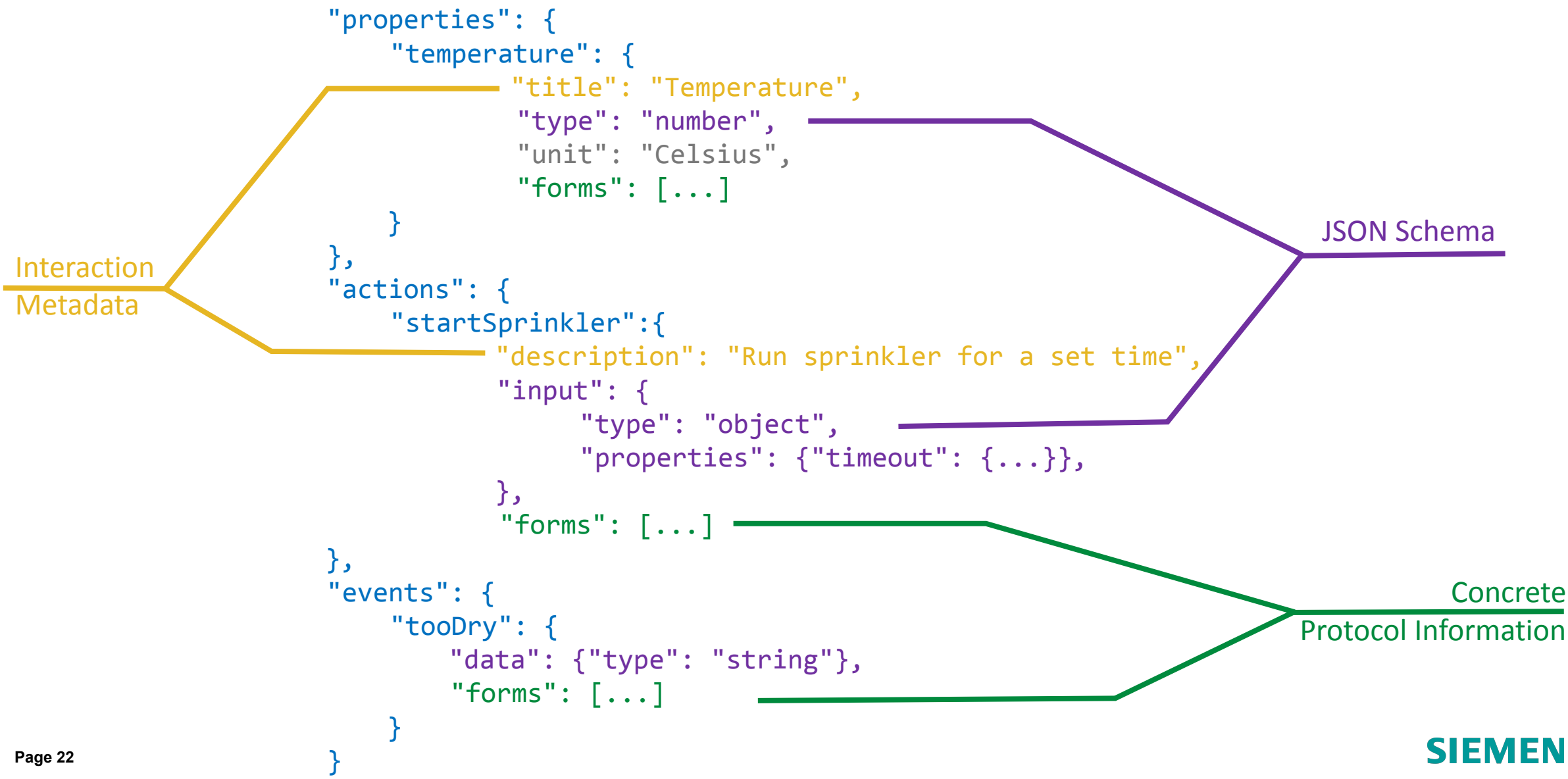
Security Metadata

Definitions of Interaction Affordances

Links to other Documents

SIEMENS

Deeper look into the Thing Description: **Interaction Level**



SIEMENS

Deeper look into the Thing Description: **Protocol Level**

SIEMENS

How is this all possible?

Second: node-wot :)

How to get running

Install the Dependencies

`npm install @node-wot/core` (mandatory core component)

`npm install @node-wot/binding-coap` (optional bindings)

Include them

```
Servient = require("@node-wot/core").Servient;
HttpServer = require("@node-wot/binding-http").HttpServer;
```

Or use the CLI by following our Readme

Different Protocol Options

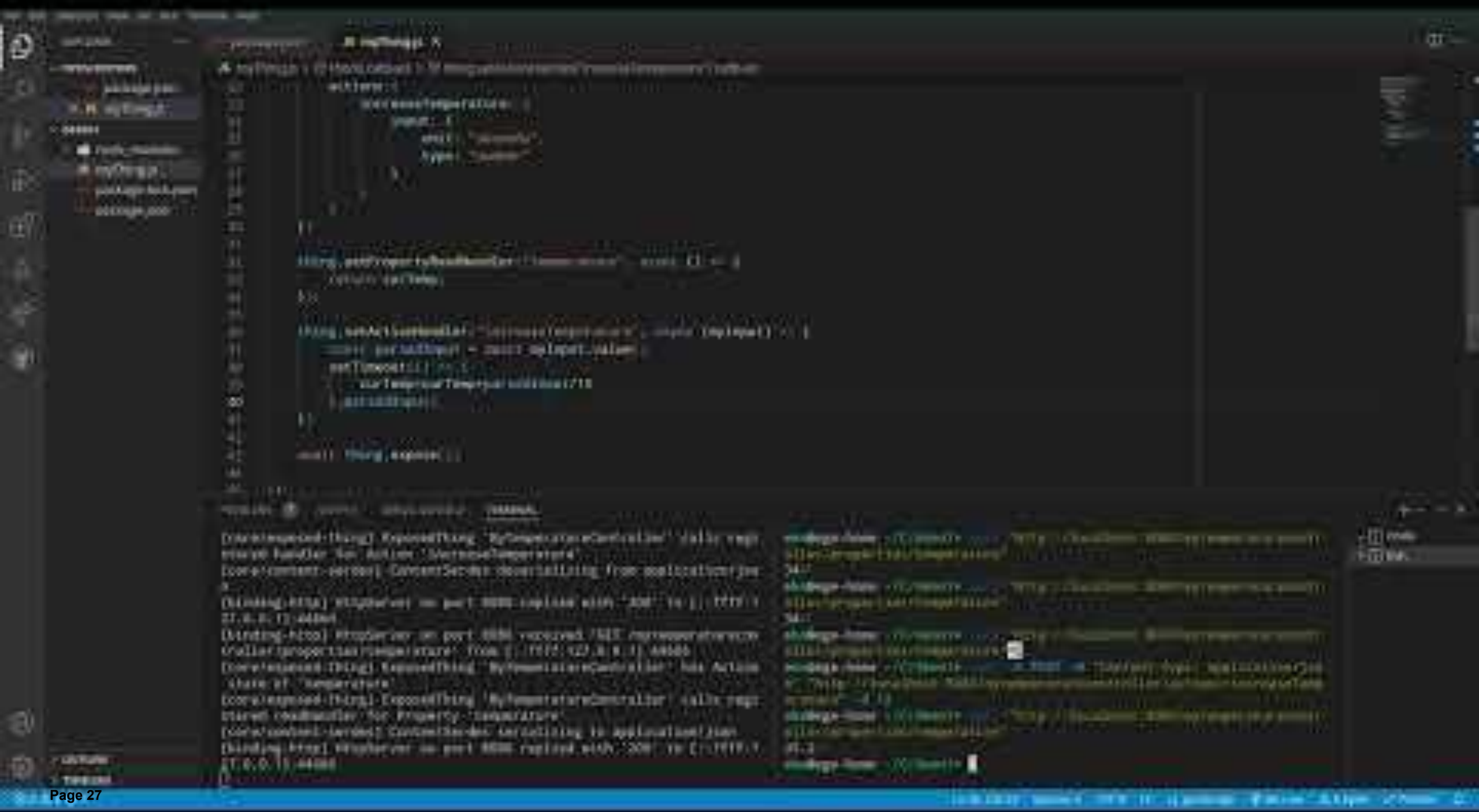
You can use node-wot with Internet protocols:

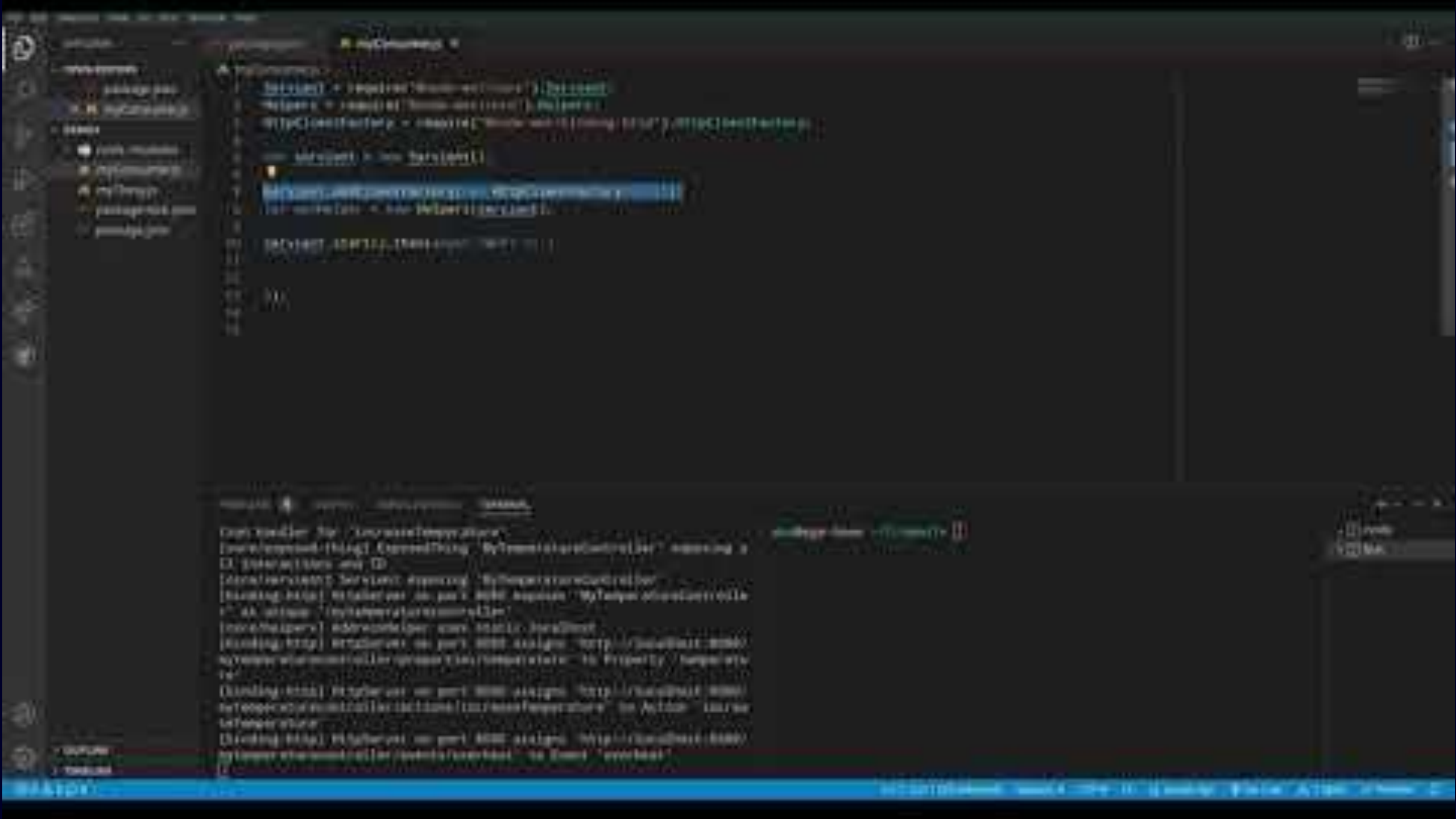
- HTTP and HTTPS
- CoAP and CoAPS
- MQTT
- Google Firestore
- Websocket (Server-side)
- NETCONF (Client-side)

and even with some industrial automation protocols:

- OPC-UA (Client-side)
- Modbus (Client-side)
- M-Bus (Client-side)

and more that you can simply implement over a protocol interface





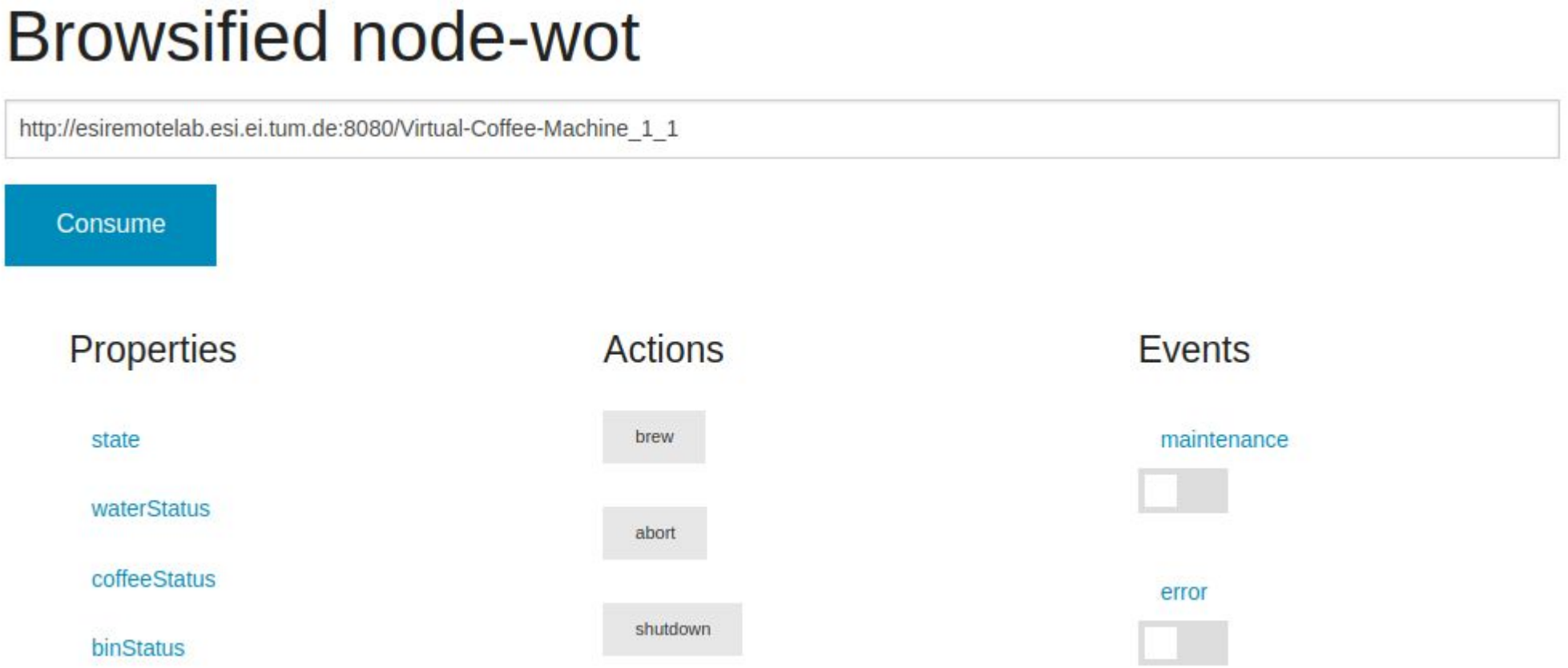
Give node-wot a try!

and/or a star a GitHub to support the development

SIEMENS

Mashups at Standardization Events

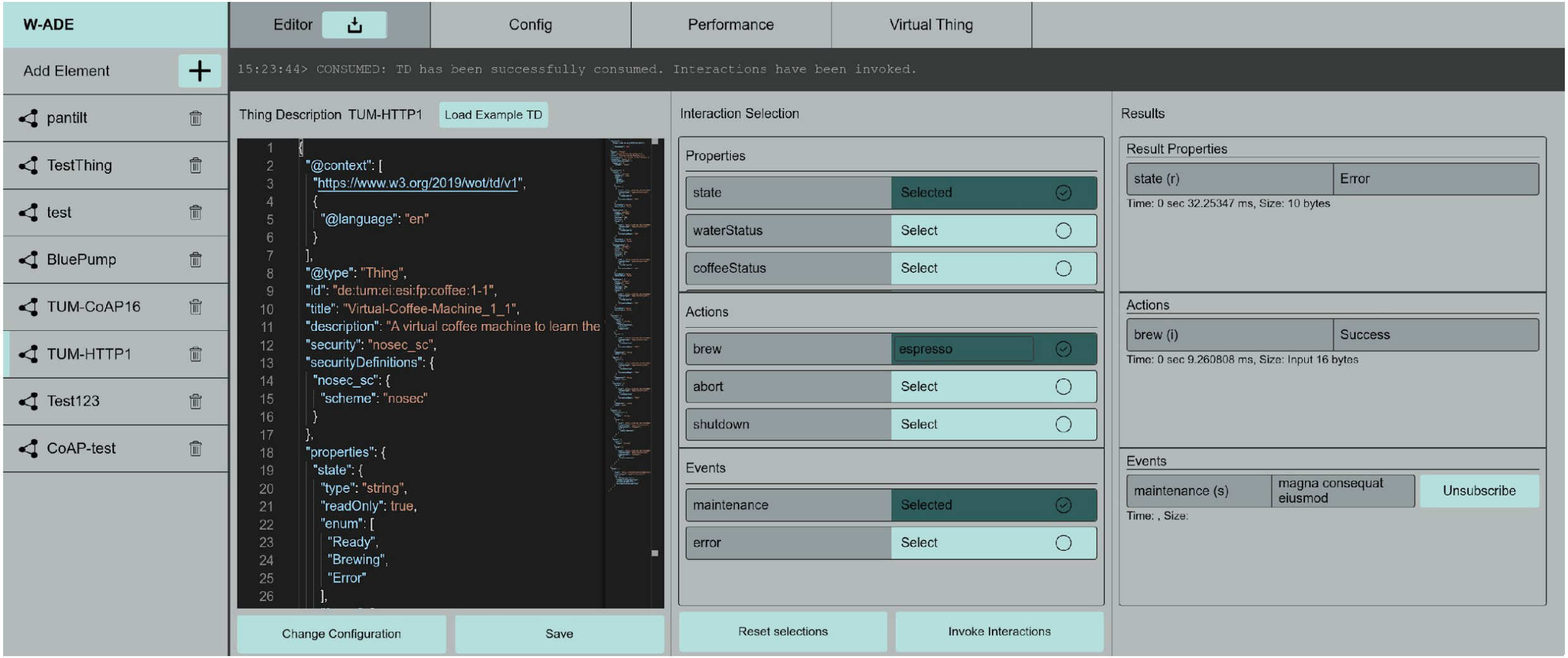
On the Browser



Simple demo hosted at http://plugfest.thingweb.io/webui/

SIEMENS

Desktop Applications

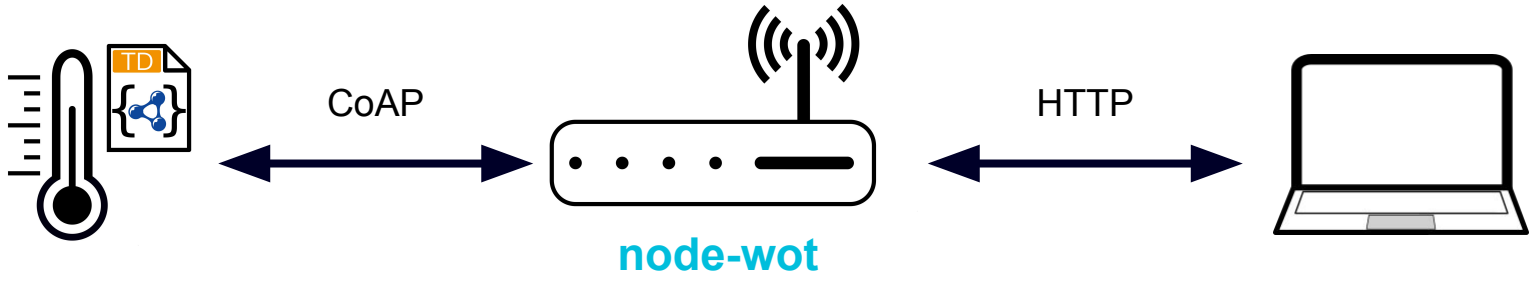


Available at https://github.com/tum-esi/wade

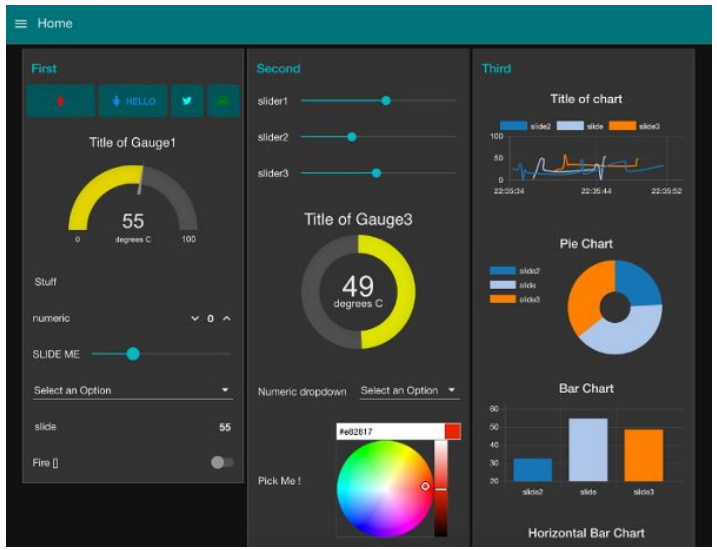
SIEMENS

And more!

Proxies

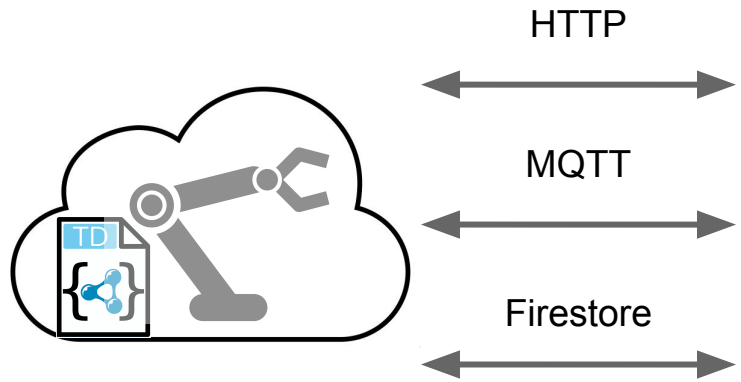


Dashboards



https://flows.nodered.org/node/node-red-dashboard

Simulators



SIEMENS

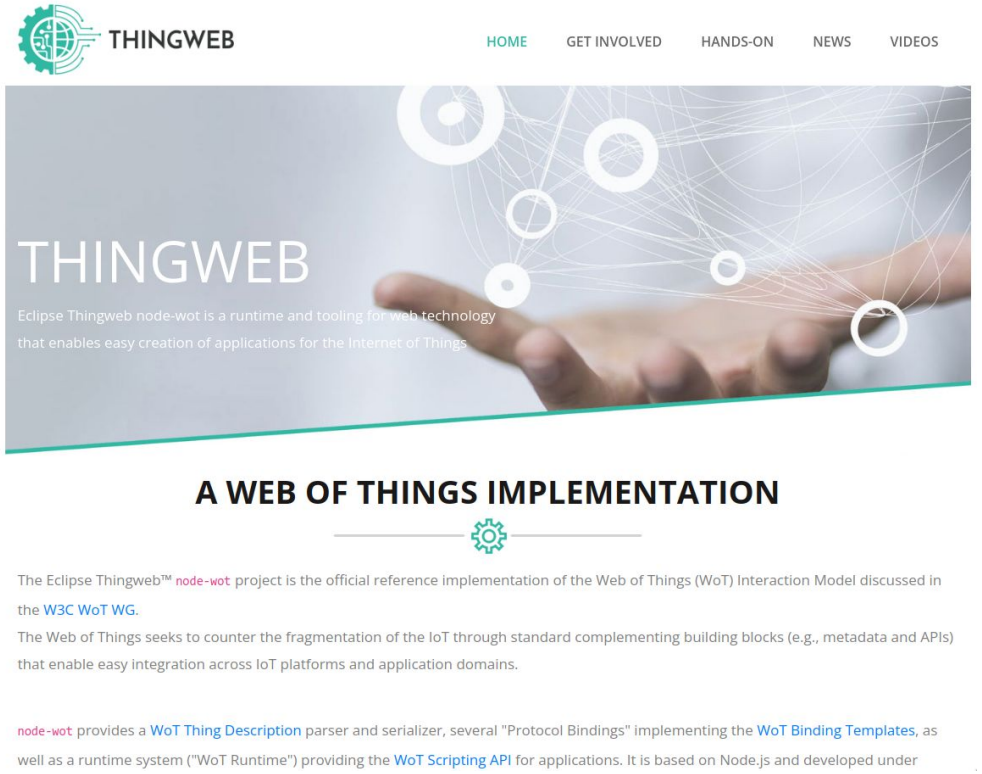
For more on Web of Things and node-wot

W3C Web of Things

https://www.w3.org/WoT/

@W3C_WoT

Eclipse Thingweb node-wot



https://www.thingweb.io/

SIEMENS

Contact

Published by Siemens

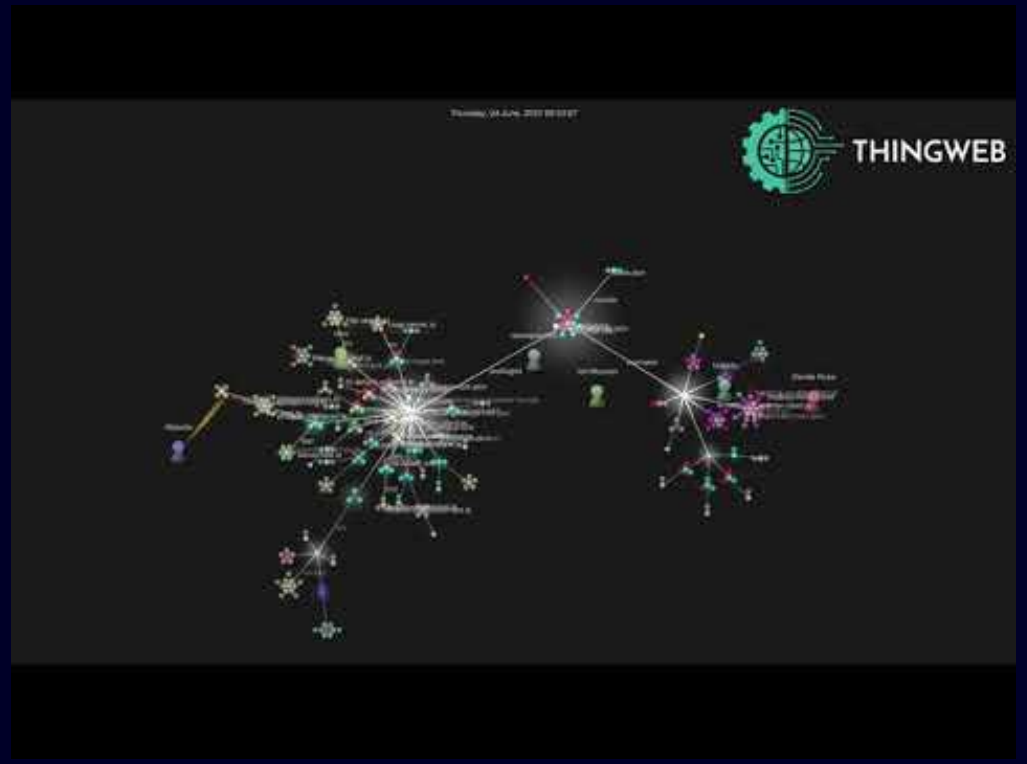
Ege Korkan
Research Scientist
T CED EWT

Phone +49 173 9401758

E-mail ege.korkan@siemens.com

Twitter: @egekorkan

GitHub: @egekorkan



SIEMENS