


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




This Video Has 53,000,761 Views

53,000,990 views Apr 6, 2020 The title of this video should change with the times. But nothing lasts forever: here's the story of how I made it work, why it used to be easier to make that w...more

**Tom Scott** 
4.98M subscribers

SUBSCRIBED 

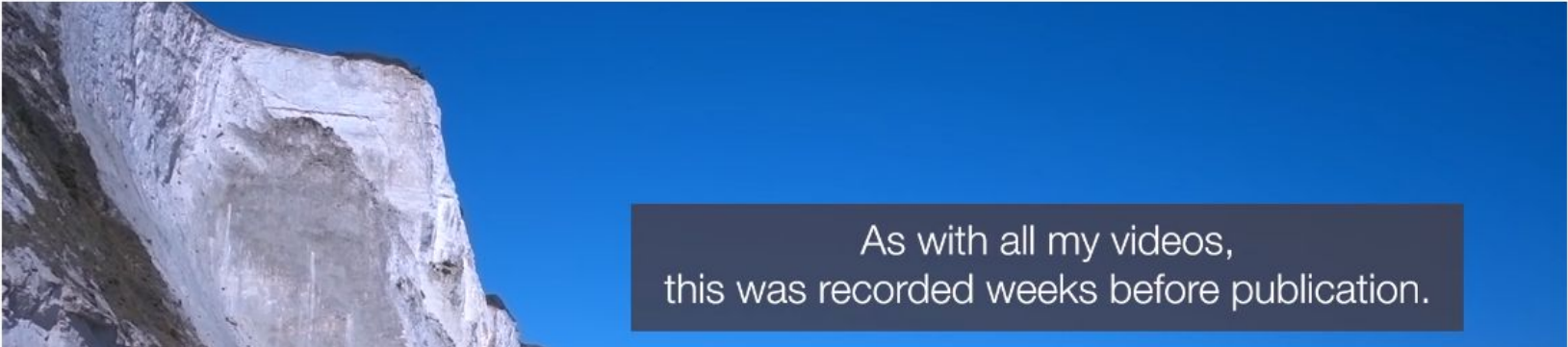
Comments
305K

 "If it's 100% spot on, it's a miracle" dang, maybe buy a lottery ticket

>

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


A Peculiar YouTube Video Title



As with all my videos,
this was recorded weeks before publication.

This Video Has 53,000,761 Views


53,000,990 views Apr 6, 2020 The title of this video should change with the times. But nothing lasts forever: here's the story of how I made it work, why it used to be easier to make that w ...more



0:00 / 10:38

This Video Has 53,000,761 Views

53,000,990 views Apr 6, 2020 The title of this video should change with the times. But nothing lasts forever: here's the story of how I made it work, why it used to be easier to make that w ...more



Tom Scott
4.98M subscribers

SUBSCRIBED

Comments
305K

"If it's 100% spot on, it's a miracle" dang, maybe buy a lottery ticket

3.2M Dislike Share Download Save ...

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

Web Mashups?

<http://www.housingmaps.com/>

<https://www.expedia.com/>

Nemea Appart'hotel - Biot

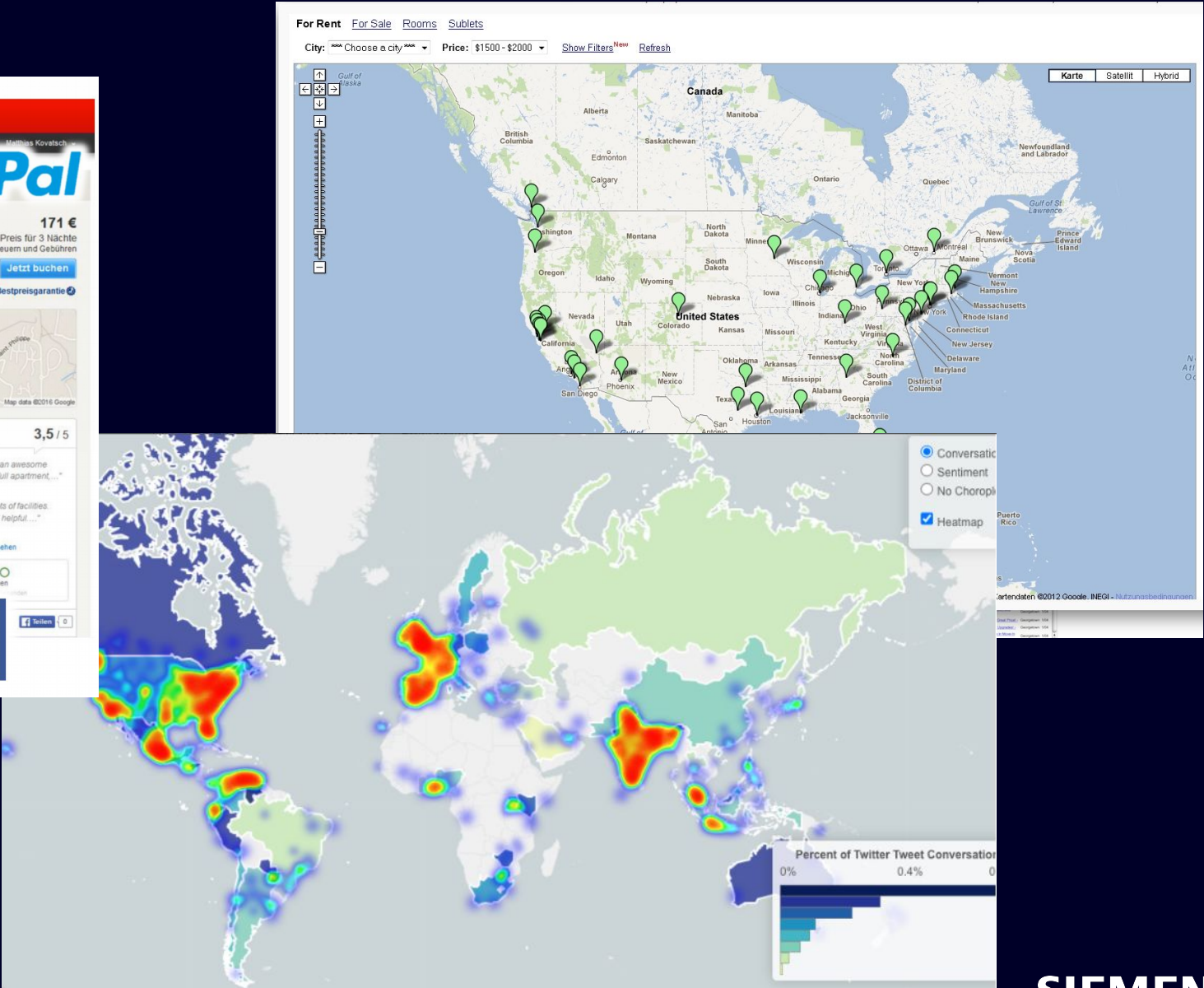
45 Rue Henri Poincaré - Accès 17, (Avenue Roumanille), Biot, Alpes-Maritimes, 06410, Frankreich, 069-945 192 395

Mo, 25. Januar 2016 - Do, 28. Januar 2016, 3 Nächte, 1 Zimmer, 1 Erwachsener

171 €
Preis für 3 Nächte
inkl. Steuern und Gebühren

[Jetzt buchen](#)

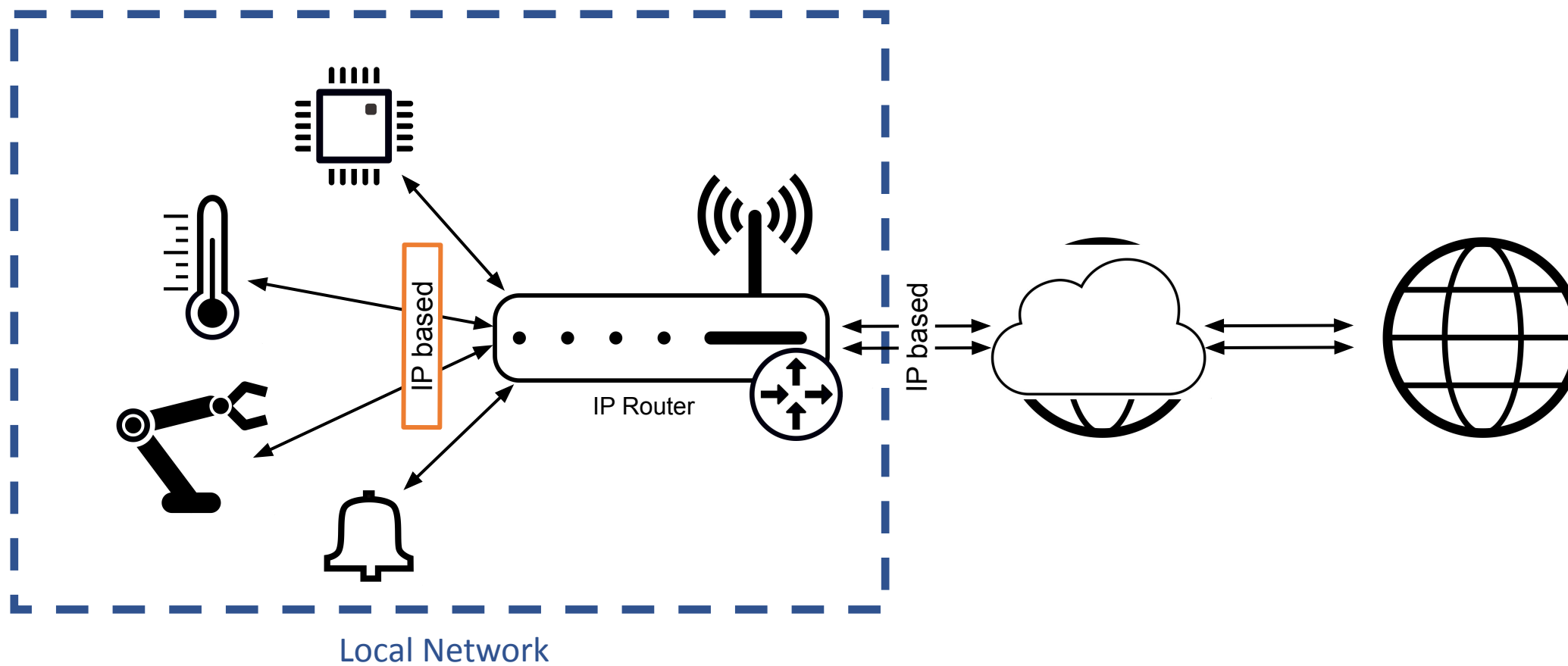
Bestpreisgarantie



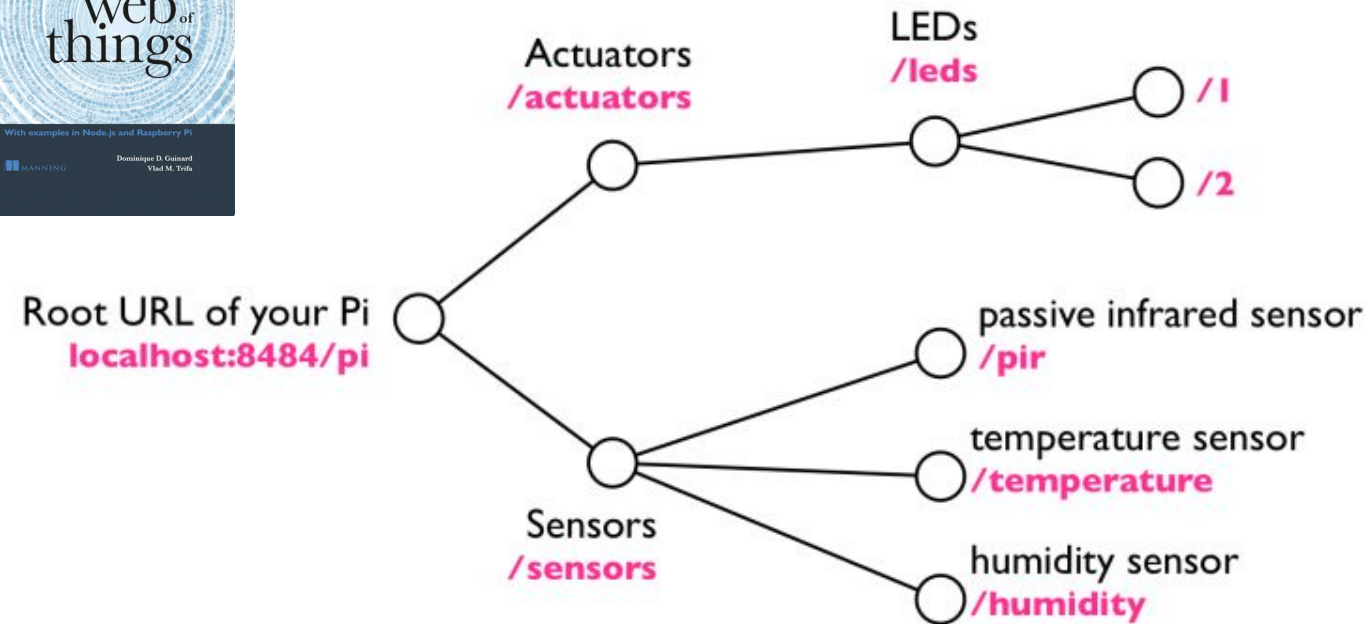
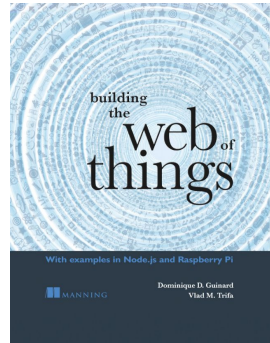
What is IoT?

(at least in this talk)

At least one way to look at it

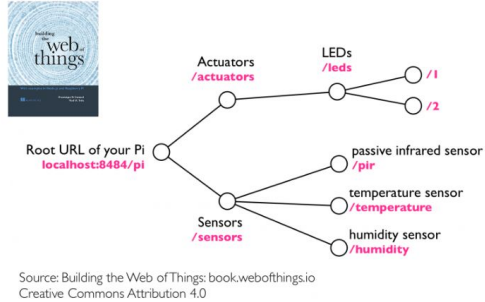


The Web of Things way to look at it



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

The Web of Things way to look at it



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

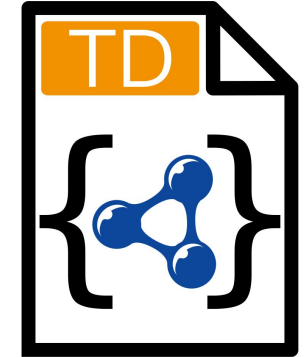


Is about describing *any* kind of API for IoT devices using *any* protocol

**Now 4 normative
deliverables, 93
participants over 35
organizations!**

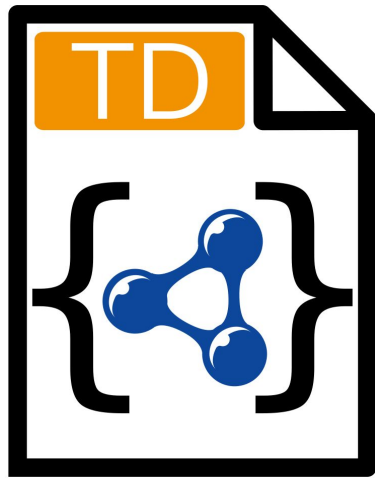
Web of Things: Thing Description (TD)

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



Why talk about standards in a Node.js summit?

Because it enables



Abstraction

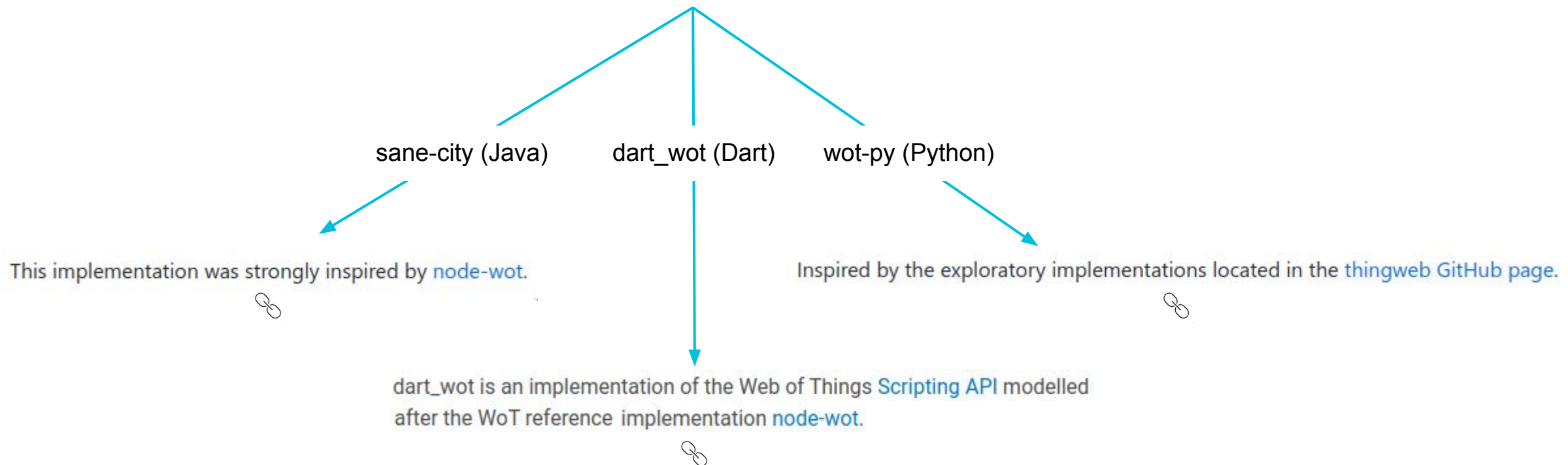
Design Pattern

Protocol Agnostic Programming

Well also, the reference implementation is in Node.js



Eclipse Thingweb node-wot

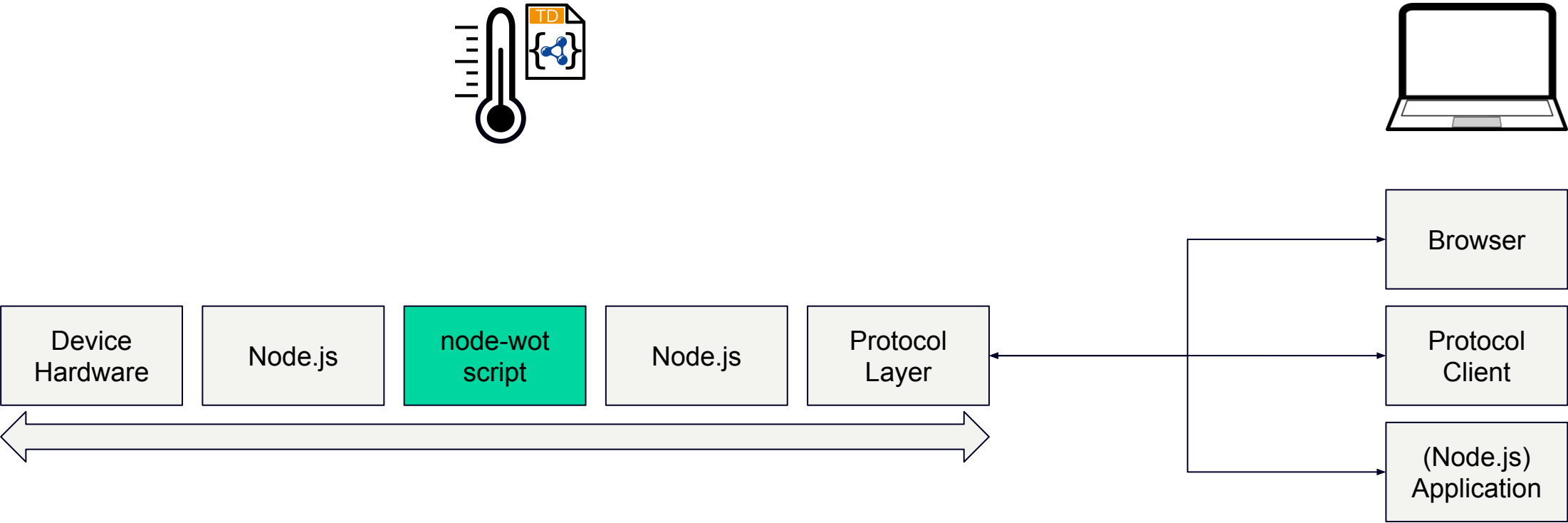


Let's see some code!

Code of a Thing

```
1 thing.setPropertyReadHandler("temperature", async () => {
2     return mySensor.temperature.read();
3 });
4
5 thing.setActionHandler("increaseTemperature", async () => {
6     myHeater.turnOn();
7     setTimeout(() => {
8         myHeater.turnOff();
9     }, "1000")
10    return true;
11 });
12
13
```

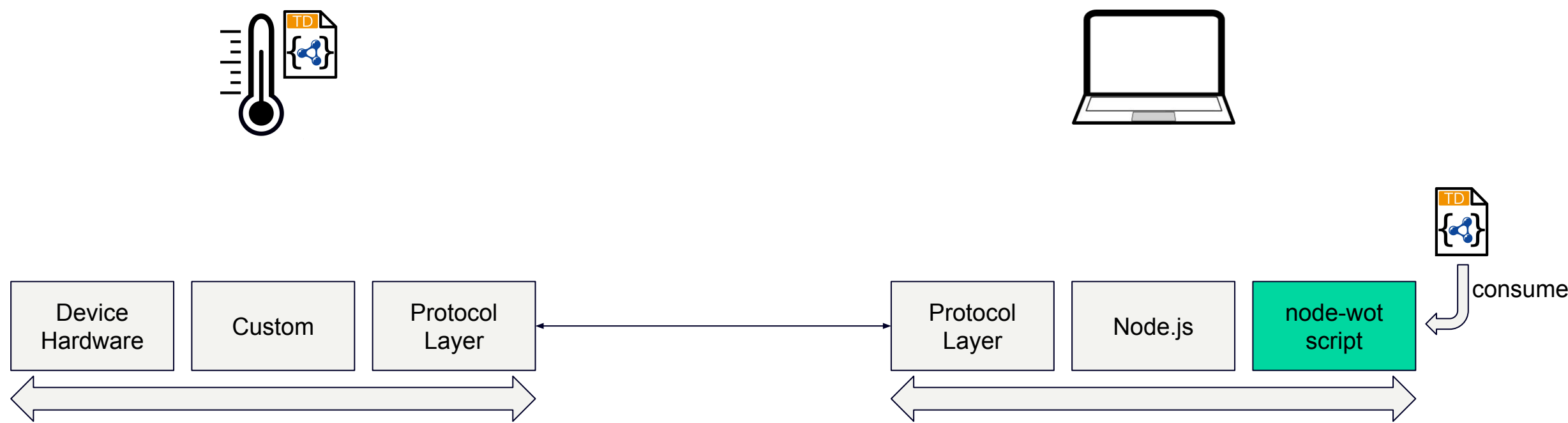
What happens in the background



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

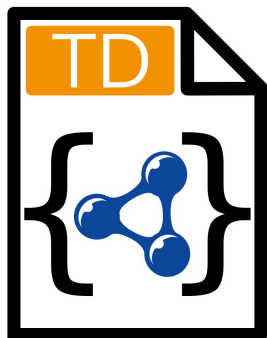


How is this all possible?

First: Thing Descriptions

Deeper look into the Thing Description

Standardized JSON (-LD) document



W3C Recommendation		TABLE OF CONTENTS
1.		Introduction
2.		Conformance
3.		Terminology
4.		Namespaces
5.		TD Information Model
5.1		Overview
5.2		Preliminaries
5.3		Class Definitions
5.3.1		Core Vocabulary Definitions
5.3.1.1		Thing
5.3.1.2		InteractionAffordance
5.3.1.3		PropertyAffordance
5.3.1.4		ActionAffordance
5.3.1.5		EventAffordance
5.3.1.6		VersionInfo
5.3.1.7		MultiLanguage
5.3.2		Data Schema Vocabulary Definitions
5.3.2.1		DataSchema
5.3.2.2		ArraySchema
5.3.2.3		BooleanSchema

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](#))

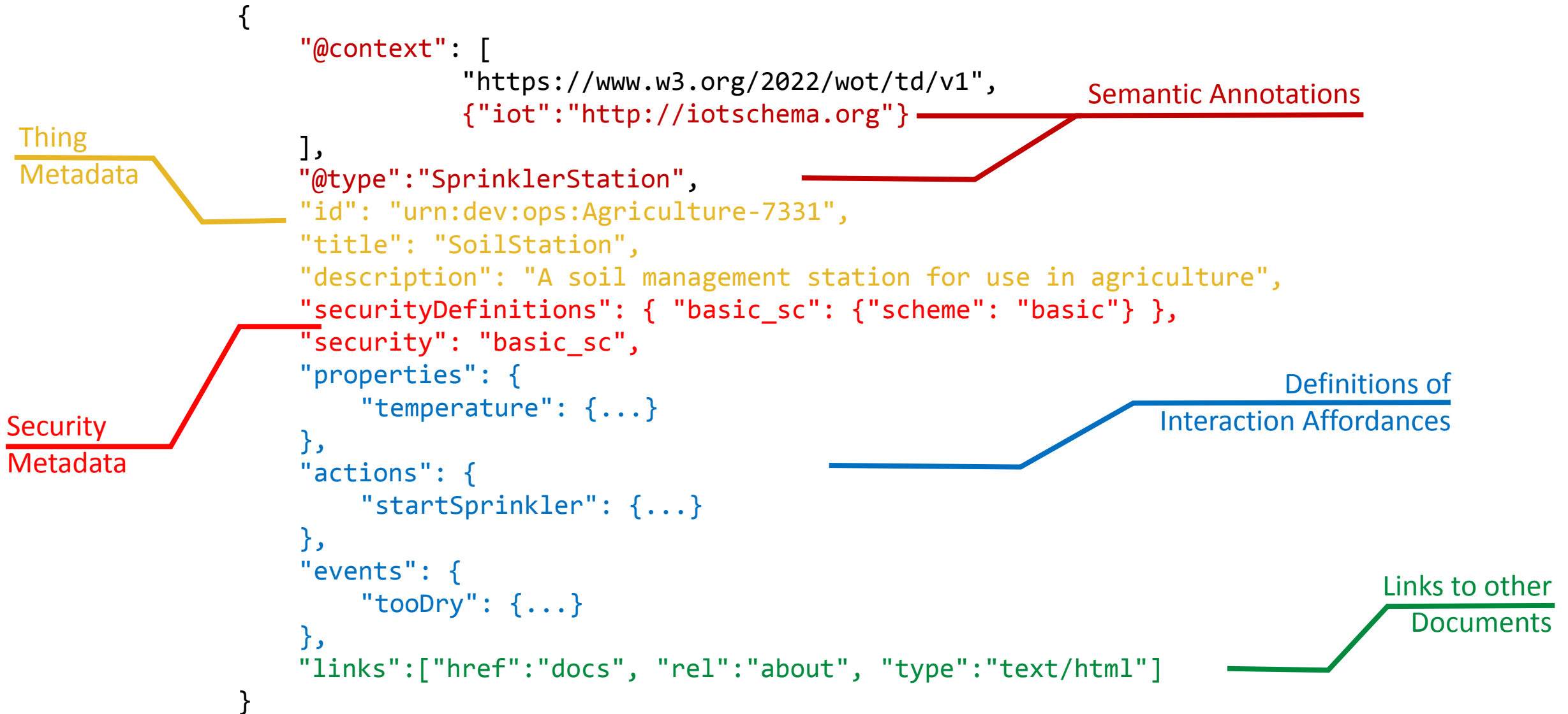
Victor Charpenay ([Siemens AG](#))

Matthias Kovatsch ([Huawei](#))

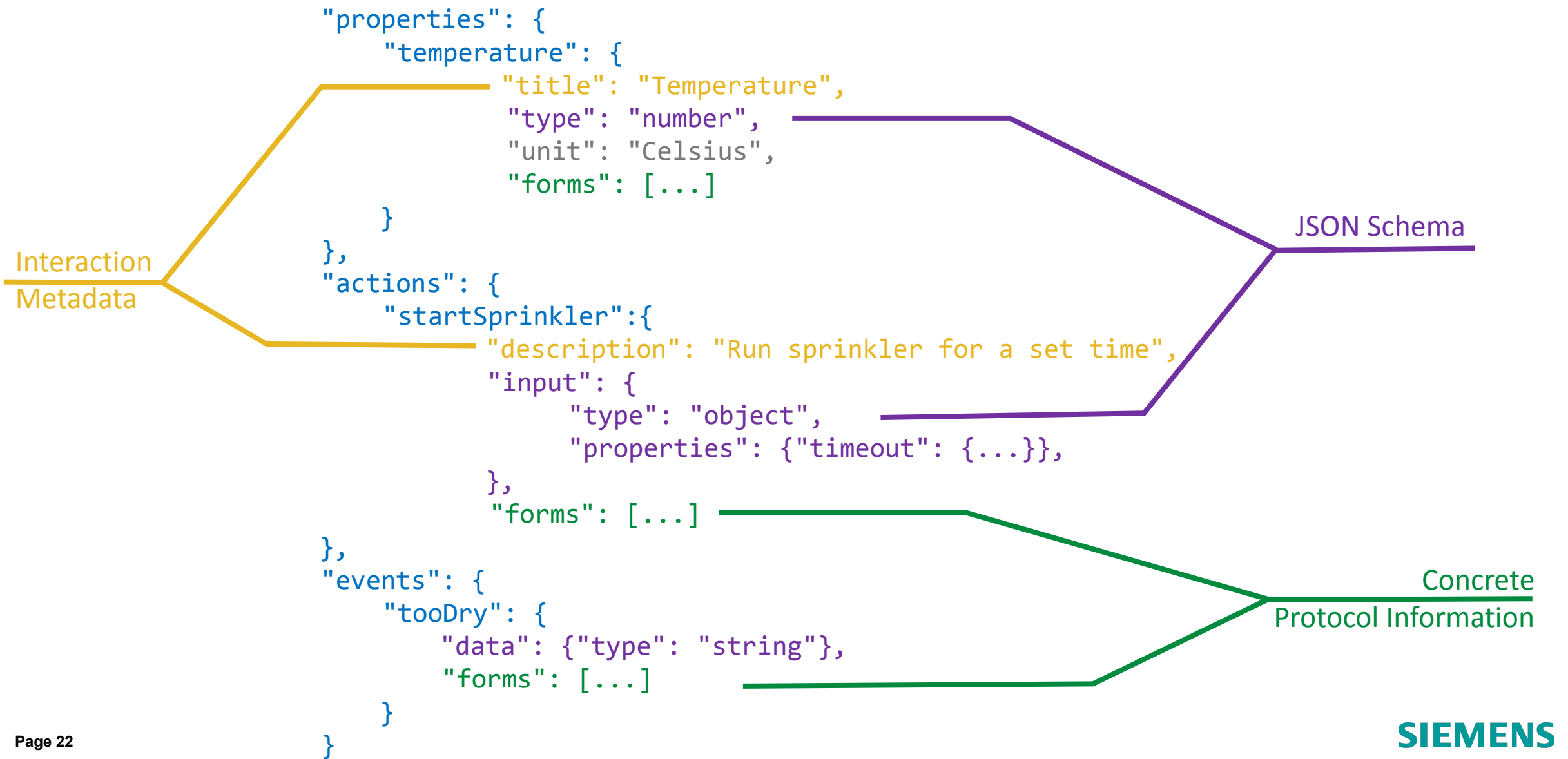
Participate:

[GitHub w3c/wot-thing-description](#)

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



Deeper look into the Thing Description: Interaction Level



Deeper look into the Thing Description: Protocol Level



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)

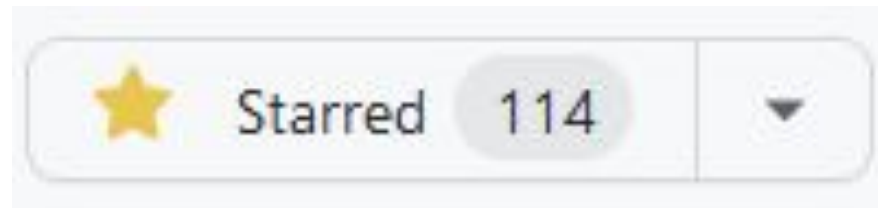
and even with some industrial automation protocols:

- 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



[illegible]

Browsified node-wot

http://esiremotelab.esi.ei.tum.de:8080/Virtual-Coffee-Machine_1_1

Consume

Properties

- state
- waterStatus
- coffeeStatus
- binStatus

Actions

- brew
- abort
- shutdown

Events

- maintenance

☐
- error

☐

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

Desktop Applications

W-ADE

Add Element

pantilt

TestThing

test

BluePump

TUM-CoAP16

TUM-HTTP1

Test123

CoAP-test

Editor

Config

Performance

Virtual Thing

15:23:44> CONSUMED: TD has been successfully consumed. Interactions have been invoked.

Thing Description TUM-HTTP1

Load Example TD

1 {
2 "@context": [
3 "https://www.w3.org/2019/wot/td/v1",
4 {
5 "@language": "en"
6 }
7],
8 "@type": "Thing",
9 "id": "de.tum.esi:fp:coffee:1-1",
10 "title": "Virtual-Coffee-Machine_1_1",
11 "description": "A virtual coffee machine to learn the",
12 "security": "nosec_sc",
13 "securityDefinitions": {
14 "nosec_sc": {
15 "scheme": "nosec"
16 }
17 },
18 "properties": {
19 "state": {
20 "type": "string",
21 "readOnly": true,
22 "enum": [
23 "Ready",
24 "Brewing",
25 "Error"
26],
27 }
28 }
29 }
30 }

Interaction Selection

Properties

state

Selected

waterStatus

Select

coffeeStatus

Select

Actions

brew

espresso

abort

Select

shutdown

Select

Events

maintenance

Selected

error

Select

Change Configuration

Save

Reset selections

Invoke Interactions

Results

Result Properties

state (r)

Error

Time: 0 sec 32.25347 ms, Size: 10 bytes

Actions

brew (i)

Success

Time: 0 sec 9.260808 ms, Size: Input 16 bytes

Events

maintenance (s)

magna consequat eiusmod

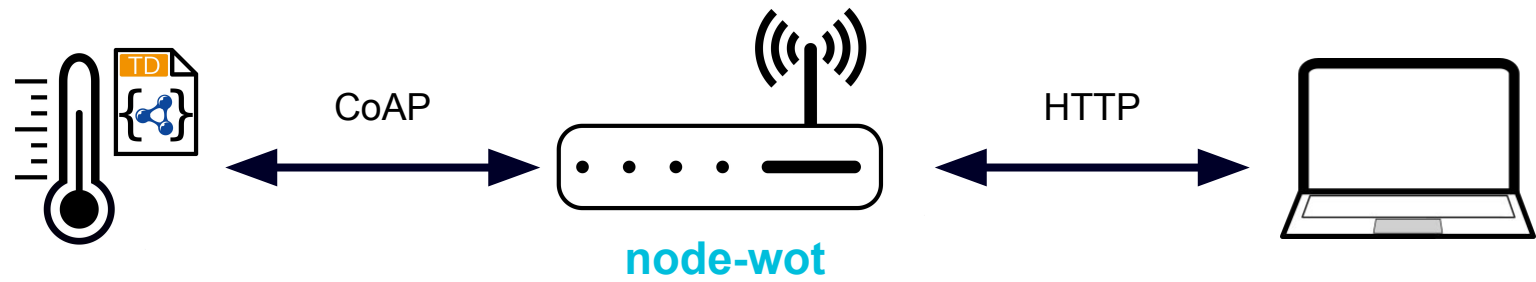
Unsubscribe

Time: , Size:

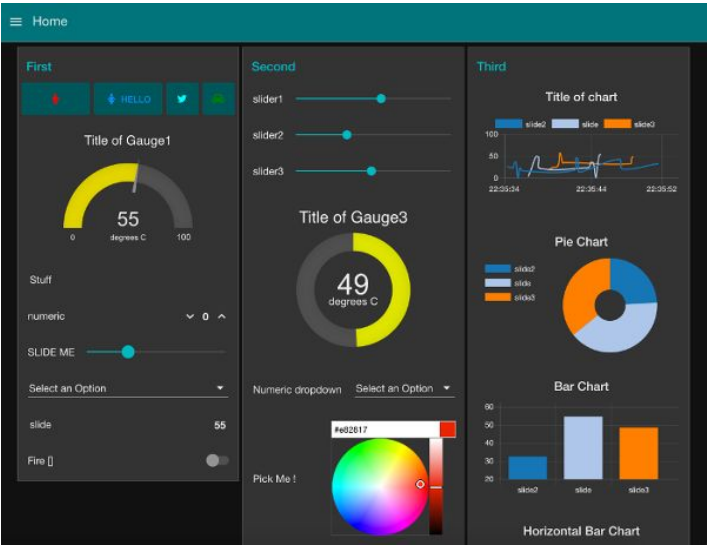
Available at <https://github.com/tum-esi/wade>

And more!

* Proxies

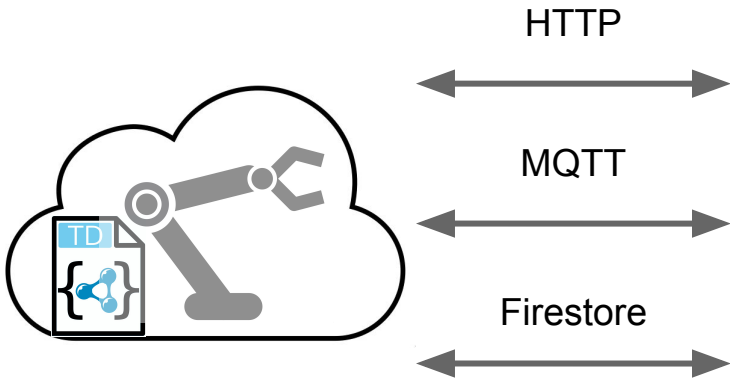


* Dashboards



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

* Simulators



For more on Web of Things and node-wot

W3C Web of Things

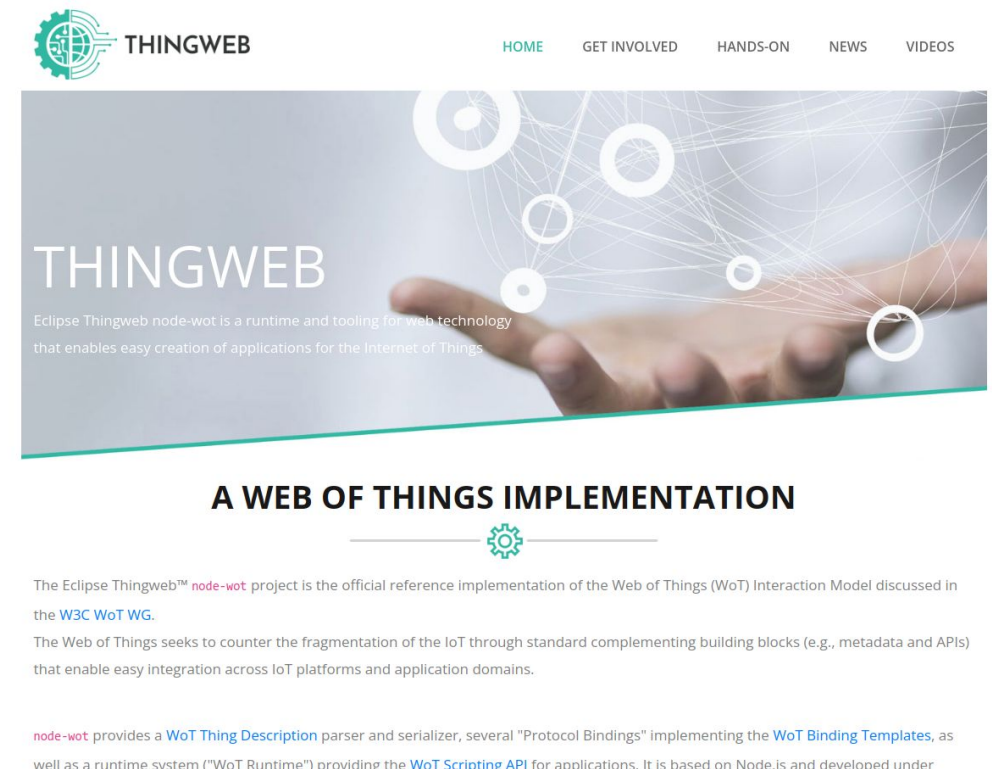


The screenshot shows the W3C Web of Things website. At the top, there is a navigation bar with the W3C logo and links for Standards, Participate, Membership, and W3C. Below this is a secondary navigation bar with links for Groups, Activities, Developers, Documentation, Videos, and Contact. The main content area features a section titled "W3C Web of Things" with a description of the project's goal to counter IoT fragmentation. To the right of this section is a "WoT Videos" link and a video player showing a "WoT" logo. Below the description are six icons representing different aspects of the project: Working Group, Interest Group, Community Groups, Task Forces, Developers, and Documentation. On the right side of the main content area, there is a "Tweets by @W3C_WoT" section showing a tweet from the W3C Web of Things account dated March 31, 2022, discussing new features like the W3C WoT Scripting API and support for Thing Models.

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

[@W3C_WoT](https://twitter.com/W3C_WoT)

Eclipse Thingweb node-wot



The screenshot shows the Eclipse Thingweb node-wot website. At the top, there is a navigation bar with the Thingweb logo and links for HOME, GET INVOLVED, HANDS-ON, NEWS, and VIDEOS. Below this is a large hero section with a background image of a hand interacting with a network of nodes. The text "THINGWEB" is prominently displayed, followed by a subtitle: "Eclipse Thingweb node-wot is a runtime and tooling for web technology that enables easy creation of applications for the Internet of Things". Below the hero section is a section titled "A WEB OF THINGS IMPLEMENTATION" with a gear icon. The text describes the Eclipse Thingweb™ node-wot project as the official reference implementation of the Web of Things (WoT) Interaction Model. It mentions that the project provides a WoT Thing Description parser and serializer, several "Protocol Bindings" implementing the WoT Binding Templates, and a runtime system ("WoT Runtime") providing the WoT Scripting API for applications. The website is based on Node.js and developed under the Eclipse Foundation.

<https://www.thingweb.io/>

| Contact

Published by Siemens

Ege Korkan

Research Scientist

T CED EWT

Phone +49 173 9401758

E-mail ege.korkan@siemens.com

Twitter: [@egekorkan](https://twitter.com/egekorkan)

GitHub: [@egekorkan](https://github.com/egekorkan)

