

Context for MQTT with the Web of Things

Ege Korkan

Siemens Munich

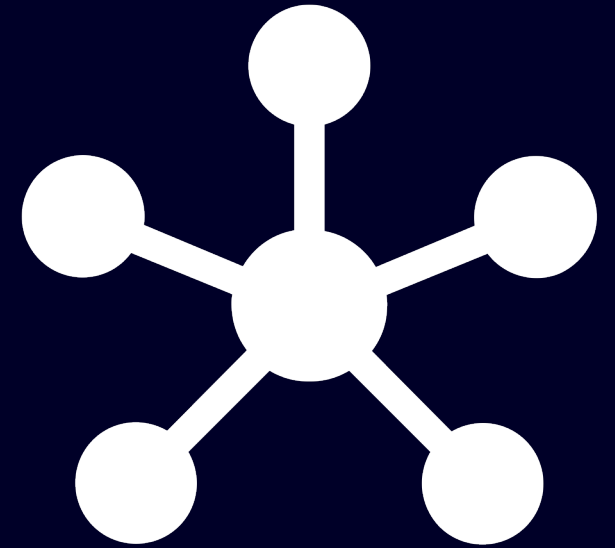
About me

- Doing R&D at Siemens for the last 2 years
- PhD at TU Munich beforehand
- Supporter of everything open source and community-driven
- Active in W3C via WoT Working Group and Community Group
- Eclipse IoT WG participant and Thingweb Project Co-Lead
- Also contributing to Asset Administration Shell
- Find me on [LinkedIn](#), [Mastodon](#), [Twitter](#), HiveMQ Slack



Building up a MQTT Architecture

Step by step



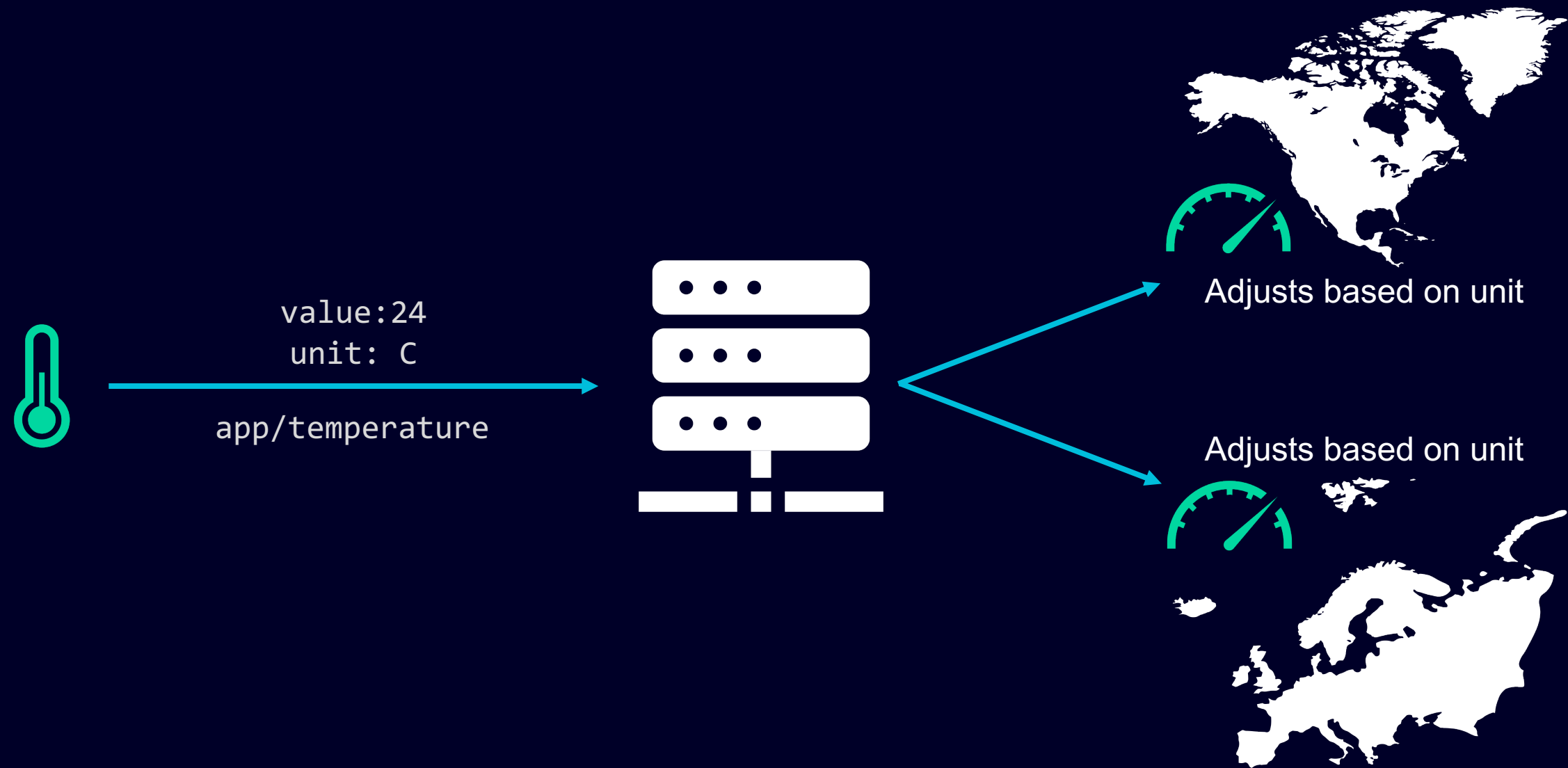
Exchange of Data Between Two or More Clients



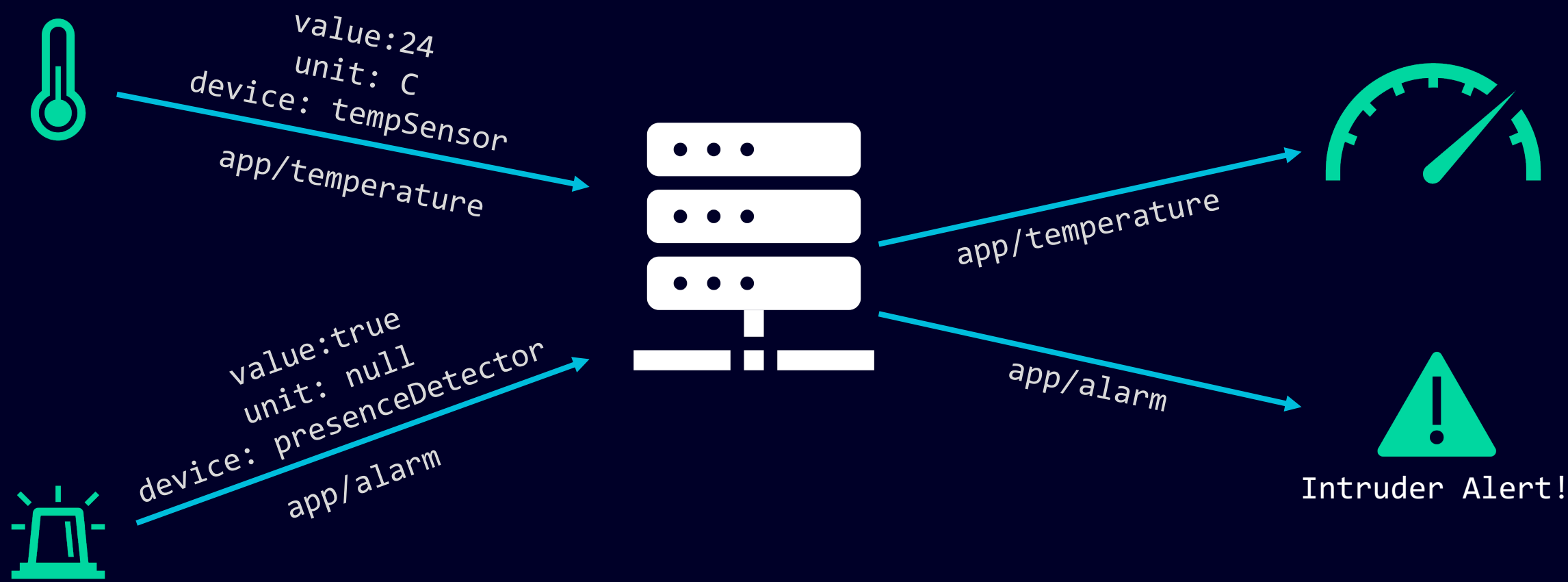
Simple Temperature Payload



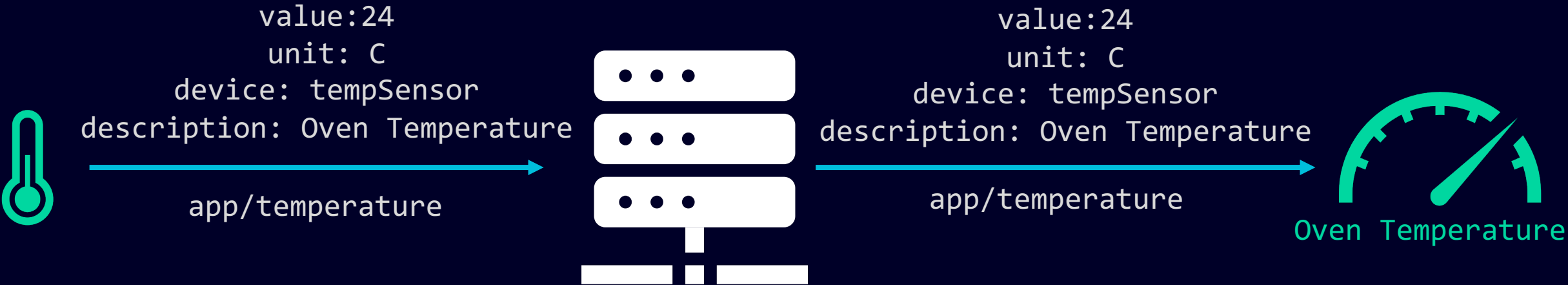
Simple Temperature Payload with Unit



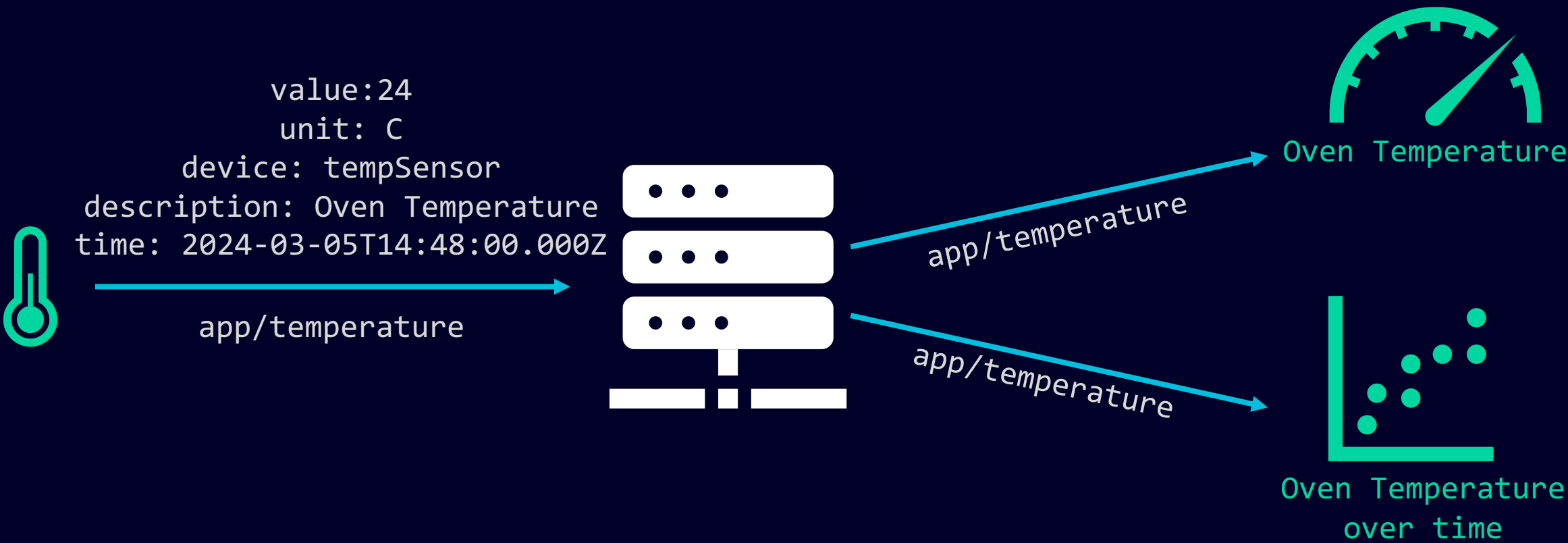
Simple Temperature Payload with Unit and Device Type



Simple Temperature Payload with Unit and Device Type and Description



Simple Temperature Payload with Unit and Device Type and Description and last measurement Date



We could add more but our template's title space is too short

So, let's change our template!

Better not

You would wish I was just creating a theoretical example

[Philips/Signify Hue API](#)

Single Light Information

```
{
  "state": {
    "on": false,
    "bri": 1,
    "hue": 33761,
    "sat": 254,
    "effect": "none",
    "xy": [
      0.3171,
      0.3366
    ],
    "ct": 159,
    "alert": "none",
    "colormode": "xy",
    "mode": "homeautomation",
    "reachable": true
  },
  "swupdate": {
    "state": "noupdates",
    "lastinstall": "2018-01-02T1
  },
  "type": "Extended color light",
  "name": "Hue color lamp 7",
  "modelid": "LCT007",
  "manufacturername": "Philips",
```

```
"productname": "Hue color lamp",
"capabilities": {
  "certified": true,
  "control": {
    "mindimlevel": 5000,
    "maxlumen": 600,
    "colorgamuttype": "B",
    "colorgamut": [
      [
        0.675,
        0.322
      ],
      [
        0.409,
        0.518
      ],
      [
        0.167,
        0.04
      ]
    ],
    "ct": {
      "min": 153,
      "max": 500
    }
  },
  "streaming": {
    "renderer": true,
    "proxy": false
  },
  "config": {
    "archetype": "sultanbulb",
    "function": "mixed",
    "direction": "omnidirectional"
  },
  "uniqueid": "00:17:88:01:00:bd:c7:b9-0b",
  "swversion": "5.105.0.21169"
```

Single Light's State Value

```
{
  "state": {
    "hue": 50000,
    "on": true,
    "effect": "none",
    "alert": "none",
    "bri": 200,
    "sat": 200,
    "ct": 500,
    "xy": [0.5, 0.5],
    "reachable": true,
    "colormode": "hs"
  },
  "type": "Living Colors",
  "name": "LC 1",
  "modelid": "LC0015",
  "swversion": "1.0.3"
}
```

What did just happen?

- Payload got bigger to accommodate everyone's needs
- **Different** needs
- More data
- Also, more **metadata**...
- Data and metadata mixing up

But what is Metadata?

What is the "context"
mentioned in the title of the
presentation?

What is changing all the time, what is not

- Definitely Static
 - Device type
 - Description of measurement
- Infrequently Changing
 - Unit
 - SW Version
- Definitely Changing
 - Sensor measurement
 - Measurement time

Metadata

Data

Let's decouple how we manage the two

One side of the *Context* with Schema Languages

The previous example's data was still **not** enough! How do we know:

- Min-max of the temperature to build the gauge (is 24 a low temperature or not?)
- What device types are possible? What more does the device type imply?
- What is the date format? We were lucky to have ISO date but doesn't have to be
- Can someone send Celsius instead of C?

These kind of questions can be answered with something like [JSON Schema](#)

JSON Schema for our Previous Examples

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "type": "object",
  "properties": {
    "value": {
      "type": "number",
      "minimum": 20,
      "maximum": 300
    },
    "unit": {
      "type": "string",
      "enum": [ "C", "F" ]
    },
    "device": {
      "type": "string"
    },
    "description": {
      "type": "string",
      "maxLength": 30
    },
    "time": {
      "type": "string",
      "format": "date-time"
    }
  }
}
```



```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "$id": "http://example.com/oven-temperature-sensor.json",
  "description": "Oven Temperature",
  "type": "object",
  "properties": {
    "value": {
      "type": "number",
      "minimum": 20,
      "maximum": 300
    },
    "unit": {
      "type": "string",
      "enum": [ "C", "F" ]
    },
    "time": {
      "type": "string",
      "format": "date-time"
    }
  }
}
```

```
{
  "value": 24,
  "unit": "C",
  "time": "2024-03-05T14:48:00.000Z"
}
```


Taking a bit further with information about the device

- What if the device has multiple topics to publish
 - What if it also subscribes to some?
- Device type
- Multiple languages of descriptions?

Can I simply get more information about the device?

Maybe yes thanks to Web of Things and its MQTT Binding

Web of Things (WoT) MQTT Binding Template

W3C Editor's Draft 14 March 2024

Available at <https://w3c.github.io/wot-binding-templates/bindings/protocols/mqtt/index.html>

EXAMPLE 7: A complex thing description using MQTT binding

```
{
  "@context": "https://www.w3.org/2019/wot/td/v1",
  "title": "Gas Sensor",
  "id": "urn:dev:test",
  "description": "Gas sensor is a sensor that can measure combustible gas",
  "securityDefinitions": {
    "nosec_sc": {
      "scheme": "nosec"
    }
  },
  "security": "nosec_sc",
  "properties": {
    "status": {
      "title": "Sensor Status",
      "observable": true,
      "enum": [
        "unknown",
        "warmup",
        "normal",
        "fault"
      ],
      "type": "string",
      "forms": [
        {
          "href": "mqtt://broker.com/",
          "mqv:filter": "application/deviceid/sensor/operation",
          "op": "observeproperty"
```

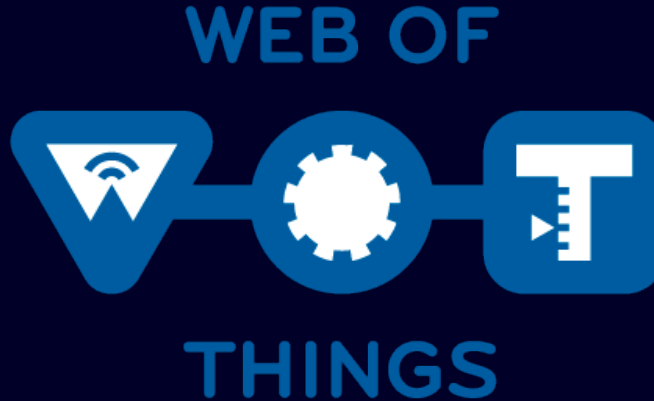
Enough teasing

Let's see what is WoT is all about.

**Family of W3C
Standards**

**(Re)Usable API
Descriptions for every
*Thing***

Royalty Free and Open

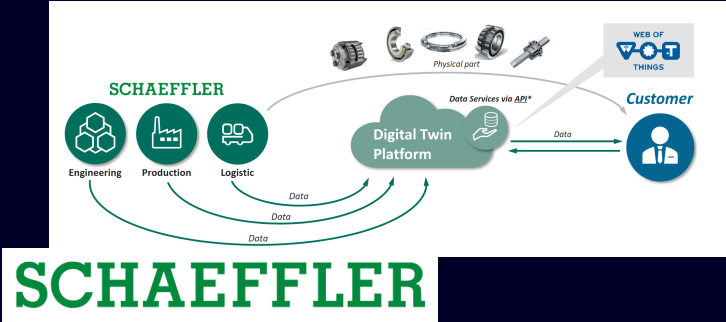


Open Source

Developer Friendly

Market Adoption

Adoption



ditto Blog Documentation HTTP API Sandbox SDKs

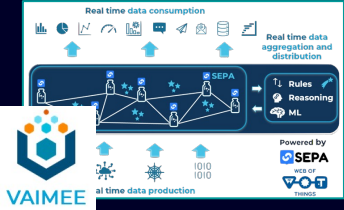
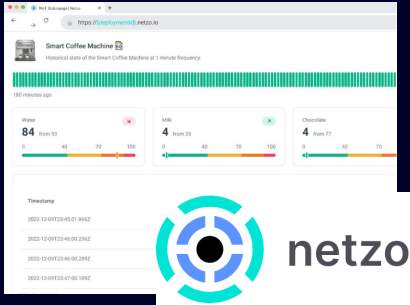
W3C WoT (Web of Things) integration

Published by Thomas Jackle on Mar 3, 2022 - Tags: [blog](#), [wot](#), [http](#)

Table of Contents

- [WoT integration in Ditto](#)
- [Example](#)
- [Ditto WoT Java model](#)
- [Feedback?](#)

The upcoming Eclipse Ditto **version 2.4.0** will add support for [W3C WoT \(Web of Things\)](#) integration, allowing twins describing the Things' capabilities.



See it for yourself: WoT Community Group Meetups

WEB OF THINGS

Web of Things Community Group

@WoTCG · 44 subscribers · 13 videos

Videos about Web of Things from the W3C Web of Things Community Group >

discord.gg/RJNYJsEgmb and 2 more links

Search

Home

Videos

Live

Playlists

Community

Latest

Popular

Oldest

Creating a Thing Description – an Example

32:14

Meetup 15: Beyond Local Connectivity: Bridging Web of Things and Matter ...

22 views · 10 days ago

WebThings

27:04

Meetup 14 - Smart Building Solutions built on WebThings

74 views · 3 weeks ago

#OPENFLEXURECON

36:15

Meetup 13 - Smart microscopy for everyone with OpenFlexure and LabThings

65 views · 1 month ago

Interactable Digital Twins for Robots via WoT

35:36

Meetup 12 - Interactable Digital Twins for Robots via WoT

83 views · 2 months ago

ISO based model WoT Standardized Information Handover

28:06

Meetup 11 - ISO Industrial Data Ontology and WoT

76 views · 3 months ago

Web of Things, not a Protocol

21:57

Introduction to W3C Web of Things - TPAC 2023

67 views · 4 months ago

The W3C WoT and Seam have the same mission – just stated differently

29:42

Meetup 10 - Designing an API for the Physical World - Seam

150 views · 5 months ago

Advantages and Challenges of Web of Things for Digital Twins

20:19

Meetup 8 - Advantages and Challenges of Web of Things for Digital Twins

57 views · 10 months ago

Web Scale Agents with WoT and No-Code Applications in IoT

27:58

Meetup 7 - Web Scale Agents with WoT and No-Code Applications in IoT

54 views · 10 months ago

Let's Integrate a Device

32:28

Meetup 6 - Automated Industrial Asset Onboarding Using Open Standards -...

182 views · 11 months ago

About our Project (NAMIB)

31:02

Meetup 4 - Delegated Authentication and Authorization in NAMIB Project

30 views · 11 months ago

Building Blocks

19:04

Meetup 5 - Empowering Industrial Automation with WoT - SICK

110 views · 11 months ago

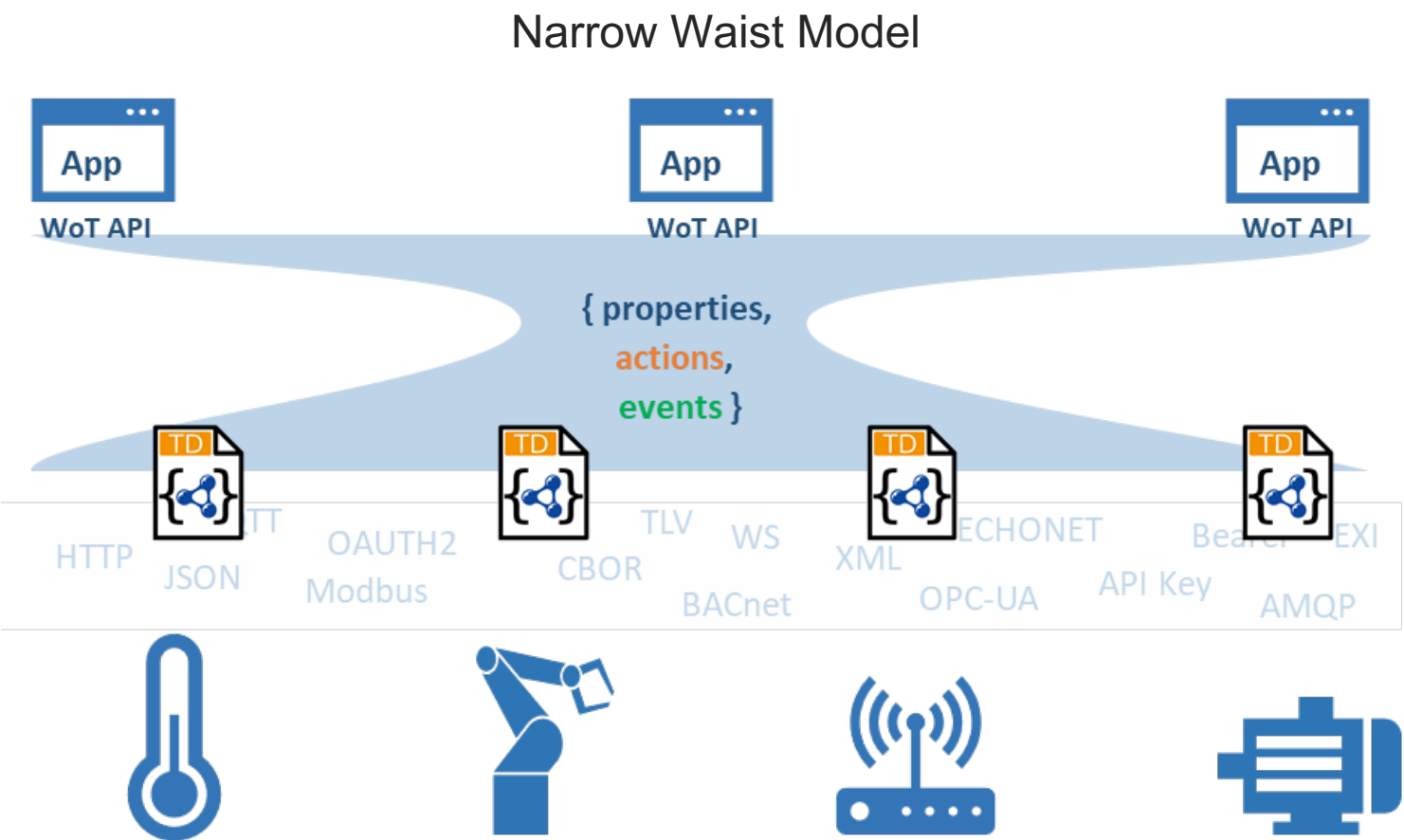
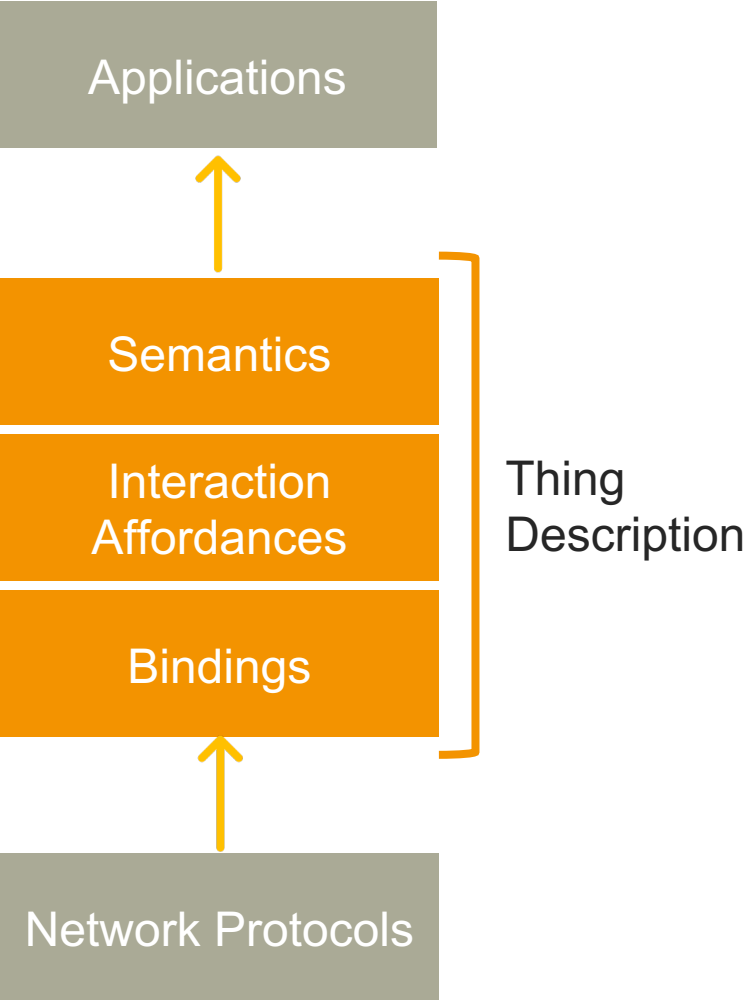
<https://www.youtube.com/@WoTCG>

<https://www.w3.org/community/wot/>

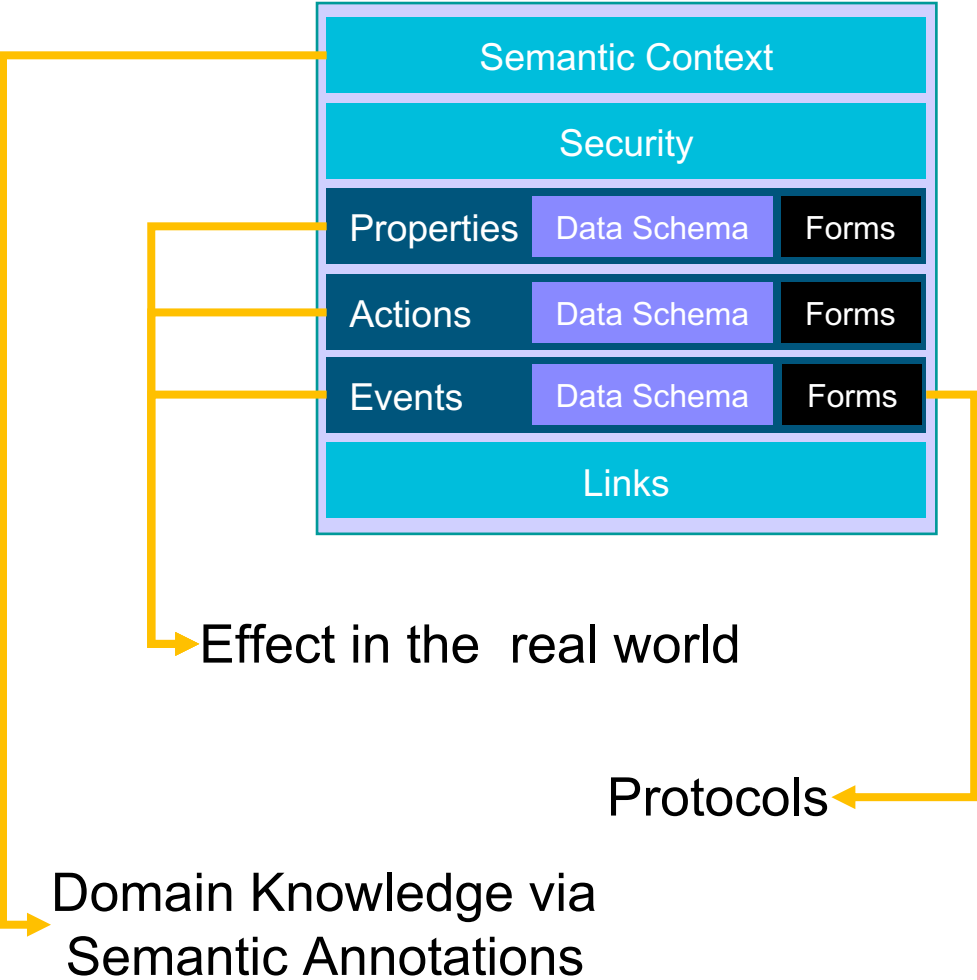
Free to join, present, exchange and connect 😊



Web of Things, not a Protocol



Core Specification: Thing Description (TD)



```
{
  "@context": "https://www.w3.org/2019/wot/td/v1",
  "id": "urn:HotelRoom",
  "@type": "Thing",
  "base": "coap://localhost:3000",
  "title": "simulated Hotel Room",
  "properties": {
    "brightness": {
      "type": "integer",
      "title": "Light Brightness",
      "forms": [
        {
          "href": "/light/Brightness",
          "contentType": "application/cbor",
          "op": [
            "observeproperty",
            "readproperty",
            "writeproperty"
          ]
        }
      ]
    }
  }
},
```

JSON-LD Serialization



Discovery and Usage by Clients

Deeper look into the Thing Description

W3C Recommendation

| TABLE OF CONTENTS | |
|-------------------|------------------------------------|
| | Abstract |
| | Status of This Document |
| 1. | Introduction |
| 1.1 | Thing Description |
| 1.2 | Thing Model |
| 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 |

Web of Things (WoT) Thing Description 1.1



W3C Recommendation 05 December 2023

▼ More details about this document

This version:

<https://www.w3.org/TR/2023/REC-wot-thing-description11-20231205/>

Latest published version:

<https://www.w3.org/TR/wot-thing-description11/>

Latest editor's draft:

<https://w3c.github.io/wot-thing-description/>

History:

<https://www.w3.org/standards/history/wot-thing-description11/>

[Commit history](#)

Implementation report:

<https://w3c.github.io/wot-thing-description/testing/report11.html>

Editors:

Sebastian Kaebisch ([Siemens AG](#))

Michael McCool ([Intel Corp.](#))

Ege Korkan ([Siemens AG](#))

Former editors:

Takuki Kamiya ([Fujitsu Research of America](#))

Victor Charpenay (when at Siemens AG)

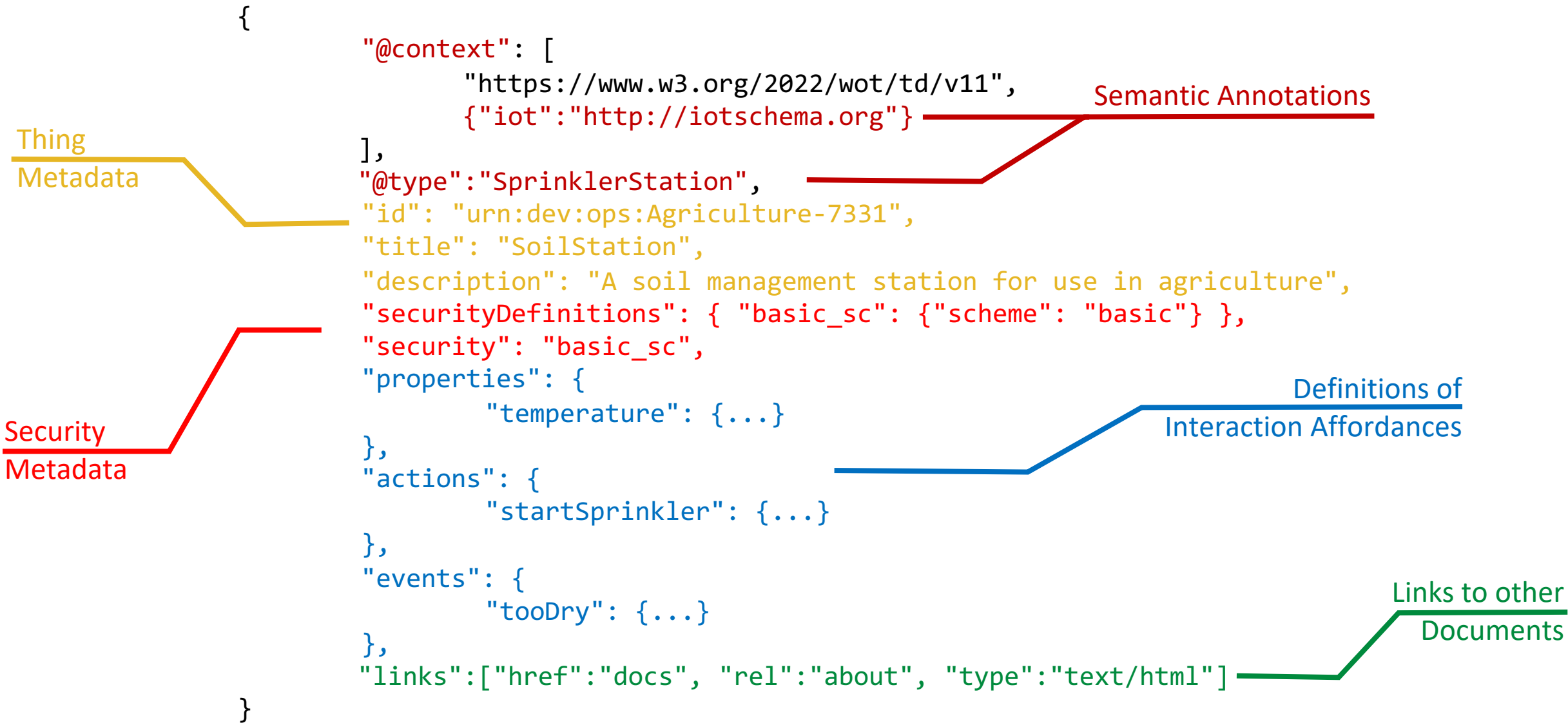
Matthias Kovatsch (when at Huawei)

Feedback:

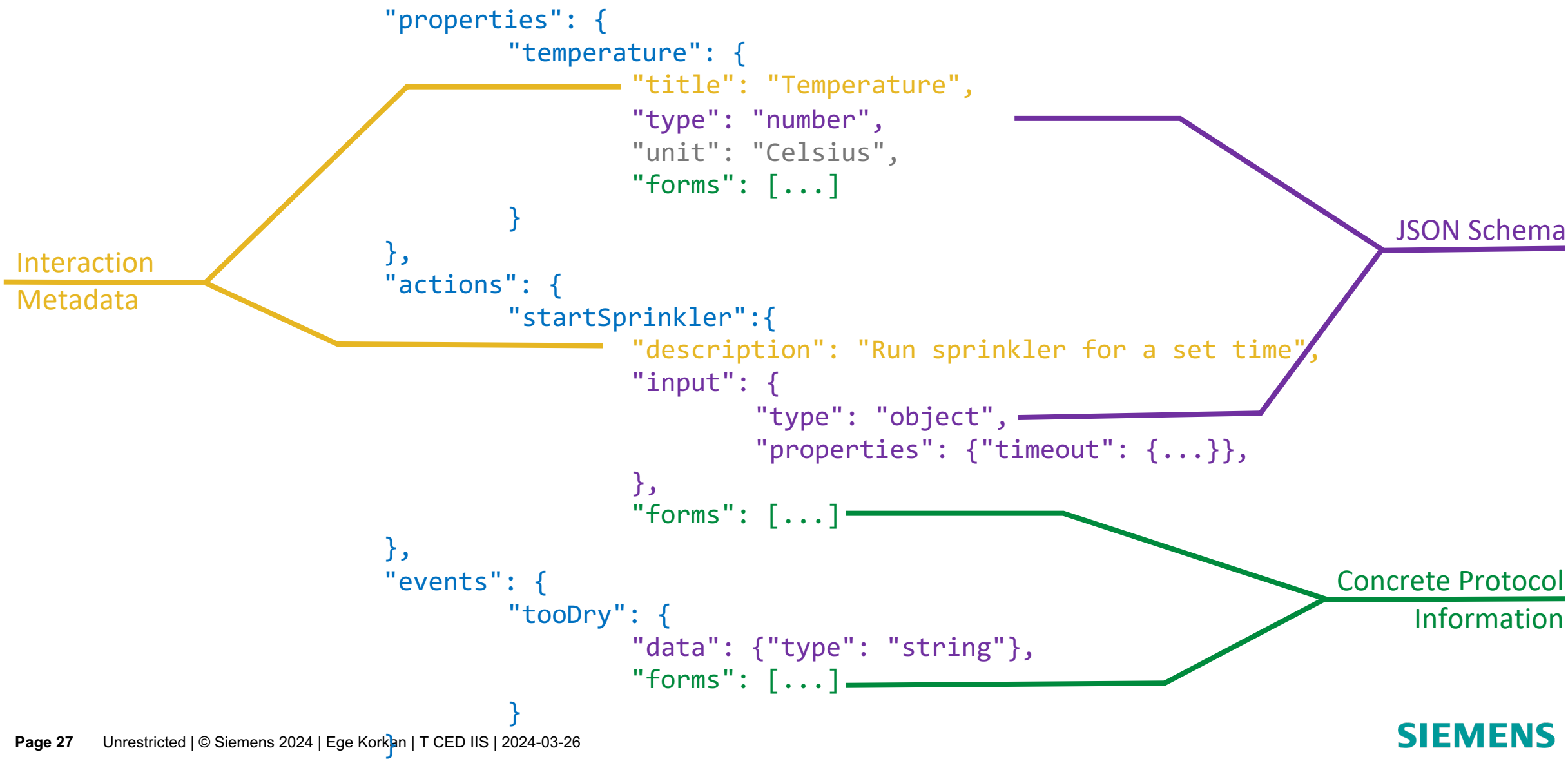
<https://www.w3.org/TR/wot-thing-description11/>



Deeper look into the Thing Description: Thing Level



Deeper look into the Thing Description: Interaction Level



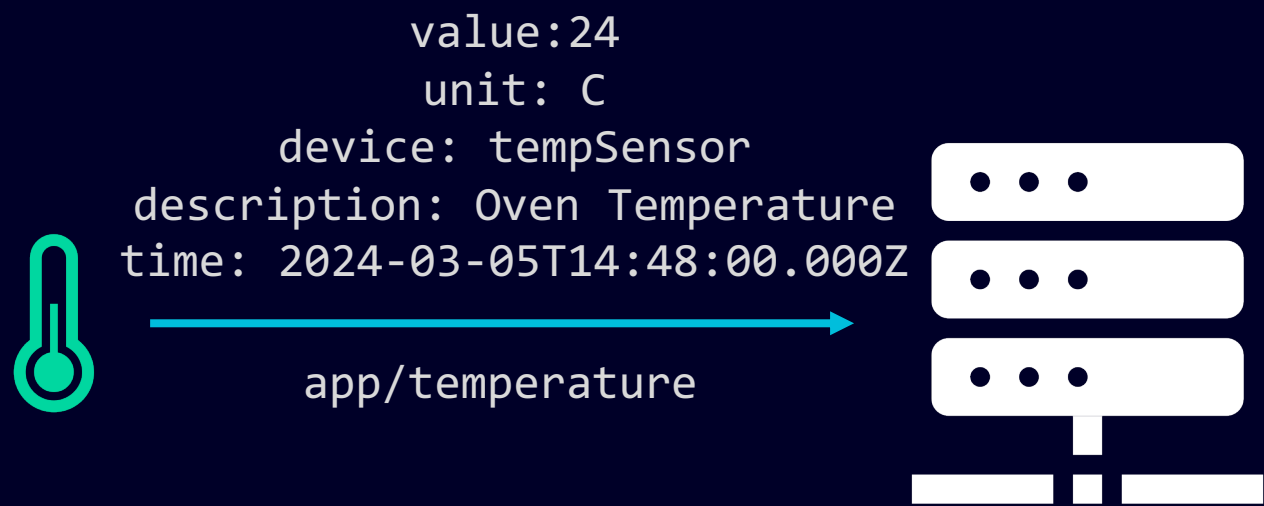
Deeper look into the Thing Description: Protocol Level



After all that talking, did the first example get any better?

It did! Let's see

Last time we left, our temperature sensor was like this



What can we change:

- Description, unit can become part of TD
- Device Type will be understood by the TD
- Data and topic will stay the same

But we can do more ;)

On to the Thing Description

It can be seen in [Thingweb Playground](#) or [EdiTDor](#)

We will go there now; the screenshot is for those who only have slides

```
TD temperaturesensor
1 {
2   "@context": ["https://www.w3.org/2022/wot/td/v1.1", {
3     "om": "http://www.ontology-of-units-of-measure.org/resource/om-2/",
4     "schema": "https://schema.org"
5   }],
6   "id": "urn:uuid:0804d572-cce8-422a-bb7c-4412fcd56f06",
7   "@type": "Thing",
8   "title": "temperaturesensor",
9   "titles": {
10     "en": "temperaturesensor",
11     "de": "Temperatursensor"
12   },
13   "description": "Temperature sensor in an industrial oven used for smelting iron",
14   "descriptions": {
15     "en": "Temperature sensor in an industrial oven used for smelting iron",
16     "de": "Temperatursensor in einem Industrieofen zum Schmelzen von Eisen"
17   },
18   "schema:softwareVersion": "1.1.0",
19   "securityDefinitions": { ...
23 },
24   "security": "nosec_sc",
25   "properties": {
26     "temperatureSimple": { ...
39 },
40     "temperature": {
41       "title": "Temperature",
42       "description": "Oven temperature measured every second",
43       "type": "object",
44       "properties": {
45         "value": {
46           "unit": "om:CelsiusTemperatureUnit",
47           "type": "number",
48           "maximum": 300,
49           "minimum": 20
50         },
51         "time": {
52           "type": "string",
53           "format": "date-time"
54         }
55       },
56       "observable": true,
57       "readOnly": true,
58       "forms": [
59         {
60           "href": "mqtt://broker.hivemq.com",
61           "mqv:filter": "temperaturesensor/properties/temperature",
62           "mqv:controlPacket": "subscribe",
63           "mqv:retain": true,
64           "op": [
65             "observeproperty",
66             "readproperty"

```

Let's go live with MQTT

Get your MQTT clients ready

- You can take any client. For CLI usage, mosquitto is enough
- I have the Thing of the previous TD programmed and connected to broker.hivemq.com .
- Let's build a very simple Consumer application by **looking** at the TD!

Note: Eclipse Thingweb has online Things for you to try. See Readme [here](#). There are also WoT-specific libraries, but it is a not requirement to have them

Takeaways

1. You do not need to change protocols, topic structure, payload structure etc. to use WoT
2. To remove the human-in-the-loop, you only need to use or implement TD parsers
3. Existing MQTT libraries and clients can be just used.
4. Moving metadata to TDs simplifies data flow and unlocks more knowledge about the devices around us

Feel free to join the WG to drive the standardization as well!

Going further with WoT

WoT else can we do?

“Additionally, you will get to see how we can onboard non-MQTT devices into MQTT systems without losing any meaning of the interactions of the device.”

Given that the protocol information is kept in the forms level, you can describe the real device (southbound), consume it with your (edge) platform, reexpose it with MQTT.

TDs can be provided by the device manufacturers or written/generated by the community

Onboarding via WoT

Similar approaches exist in different solutions:

- HiveMQ Edge
- PTC ThingWorx
- Azure DTDL
- OPC UA Onboarding

| | |
|-----------------------|---|
| ⬅ | |
| Name | Web-of-Things Connectivity |
| Short Name | WoTConnectivity |
| Organization | OPC Foundation |
| Collaboration Website | |
| Marketing Website | https://opcfoundation.org/markets-collaboration/ |
| Classification | Cloud Computing |
| Markets | |
| Status | Active |
| Description | <div><p>Define how to configure the OPC UA Information Model that exposes the underlying devices described by WoT Thing Descriptions via an OPC UA Server that acts as a protocol mapping and data model mapping service from non-OPC UA asset interfaces to OPC UA.</p><p>This service is often run on Industrial Edge gateways. It accesses non-OPC UA assets (like Modbus assets or assets with a proprietary interface) and maps the asset's data model into an OPC UA server address space.</p><p>The schema for the mapping to OPC UA will be provided by W3C Web of Things standard in Thing Description JSON-LD format, which will potentially be extended to cover all required protocol bindings (see https://www.w3.org/TR/wot-binding-templates/#binding-overview).</p></div> |

Where to Learn More?

Official Web Page of W3C WoT



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

<https://w3c.social/@wot>

Try one of the many [open-source projects](#)

Tutorial in works [here](#)

Read documentation and watch some videos on the official page

Join one of the office hours of the CG or attend [a meetup!](#)

Thank you for listening!

Contact

Published by Siemens 2024

Ege Korkan

Research Scientist

T CED IIS-DE / Germany

E-mail ege.korkan@siemens.com