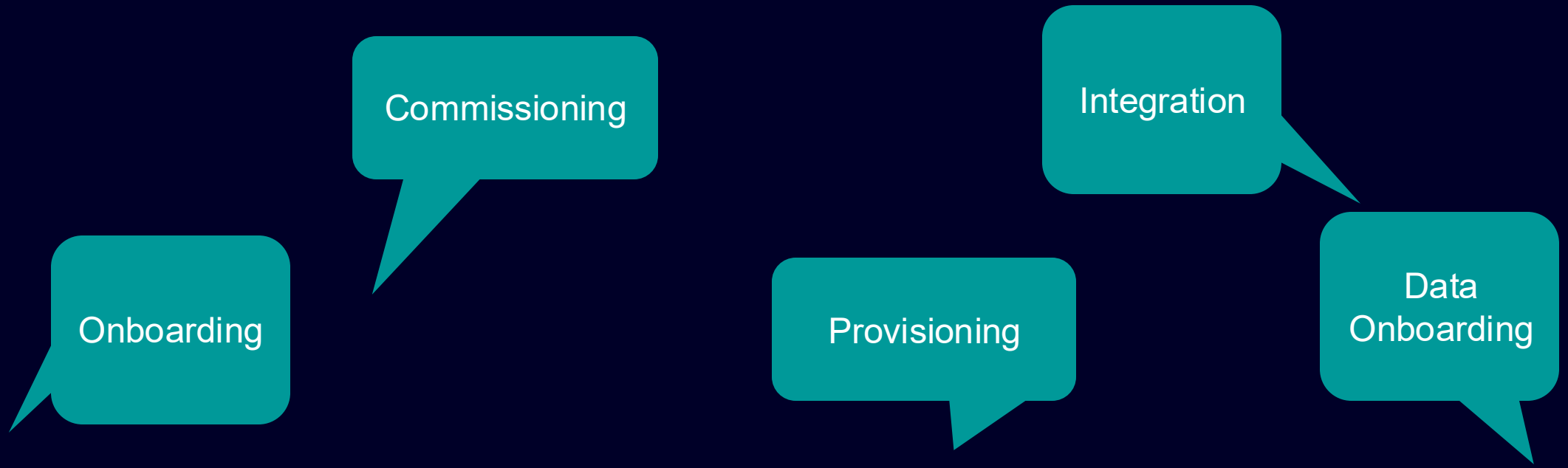
# Managing Device Descriptions with the Thing Model Catalog

Ege Korkan

Siemens AG
Smart Infrastructure, CTO Office



SIEMENS




# Adding a new Device

*State of the Art*, Focusing on the Modbus TCP Protocol

Home automation forums



SIEMENS

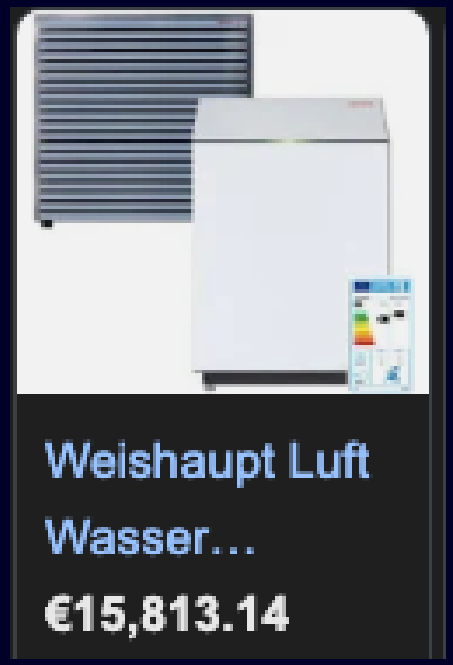
## Weishaupt WBB, WWP LS und WGB Modbus Configuration

Tutorials & Examples  Solutions

**Blizzard** Andreas — 1 — Oct 2024

Hi all,
I wanted to shared by Modbus Configuration for reading (and writing) Weishaupt Heatpumps WBB, WWP LS und WGB (according to the documentation I have). Maybe some other people have a Weishaupt Heatpump, too, and can make use of it. You need to enable Modbus in the Expert Section of the Heatpump. This configuration has been running with my heatpump for quite some time.

openHAB forum




```
"Weishaupt Wärmepumpe" [ start="41101", length="12", refresh="30000", type="hold
  - Konfiguration"                [ readStart="41101", readValueType="int16" ]  // (
  - Anforderung Typ"              [ readStart="41102", readValueType="int16" ]  // (
  - Betriebsart"                  [ readStart="41103", readValueType="int16", write
  - Pause / Party"                [ readStart="41104", readValueType="int16", write
  - Raumsolltemperatur Komfort"   [ readStart="41105", readValueType="int16" ]  // (
  - Raumsolltemperatur Normal"    [ readStart="41106", readValueType="int16" ]  // (
  - Raumsolltemperatur Absenk"    [ readStart="41107", readValueType="int16" ]  // (
  - Heizkennlinie"                [ readStart="41108", readValueType="int16" ]  // (
  - Sommer Winter Umschaltung"    [ readStart="41109", readValueType="int16" ]  // 3
  - Heizen Konstanttemperatur"    [ readStart="41110", readValueType="int16" ]  // 2
  - Heizen Konstanttemp Absenk"   [ readStart="41111", readValueType="int16" ]  // 2
  - Kühlen Konstanttemperatur"    [ readStart="41112", readValueType="int16" ]  // (
```



SIEMENS

# Weishaupt Heatpump integration via modbus

Share your Projects!

**tobiasm** TobiM — 2 — Jul 2022

Dear Weishaupt Users,

today i would like to share with you how i have integrated my Weishaupt Biblock Heat pump via modbus in Homeassistant.
I was in touch with some Weishaupt employees and happy that they have shared a List of Datapoints for modbus connectivity. Other ways of integrating were not 100% stable & i have now tested modbus through the last months & it works really great! It is even possible to set e.g. Operating mode or Target warmwater temperature.

Here are the steps to follow:

1. Configure Modbus with static IP on your Weishaupt Device (not possible in WEMPortal - you need to configure it on the device)
2. Configure modbus connectivity - here is my sample yaml:

▶ Modbus YAML config

[HomeAssistant Forum]

```yaml
- name: wpump
  type: tcp
  host: 192.168.X.X
  port: 502
  climates:
      - name: "WP_Warmwasser"
        address: 42103
        input_type: holding
        count: 1
        data_type: int16
        max_temp: 50
        min_temp: 15
        offset: 0
        precision: 1
        scale: 0.1
        target_temp_register: 42103
        temp_step: 1
        temperature_unit: C
  sensors:
#Warmwasser
    - name: WP_Warmwassersolltemperatur
      slave: 1
      address: 32101
      input_type: input
      unit_of_measurement: °C
      state_class: measurement
      count: 1
      scale: 0.1
```



SIEMENS

# At least Siemens does better ☺



SIEMENS

# Or does not…

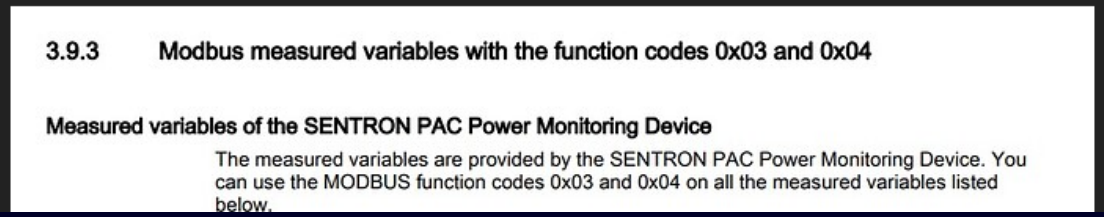
**vaox** Valter — 1 — Nov 2022

Hi, I would like to read from Sentron PAC3200 . I have tried this configuration:
modbus:

- name: pac3200
  type: tcp
  host: 192.168.5.3
  port: 502
  binary_sensors:
  - name: “Voltage L1”
    slave: 0
    address: 1
    input_type: holding
    unit_of_measurement: V (this command generate error)
    state_class: measurement (this command generate error)
    count: 1 (this command generate error)
    offset: 0 (this command generate error)
    scale: 0.1 (this command generate error)
    data_type: uint16 (this command generate error)

Example of error
Invalid config for [modbus]: [scale] is an invalid option for [modbus]. Check: modbus->modbus->0->binary_sensors->0->scale. (See /config/configuration.yaml, line 17).

This is part of manual that shows how to read data from device.



3.9.3 Modbus measured variables with the function codes 0x03 and 0x04

Measured variables of the SENTRON PAC Power Monitoring Device

The measured variables are provided by the SENTRON PAC Power Monitoring Device. You can use the MODBUS function codes 0x03 and 0x04 on all the measured variables listed below.

**nikito7**

Change binary_sensors: to sensors:

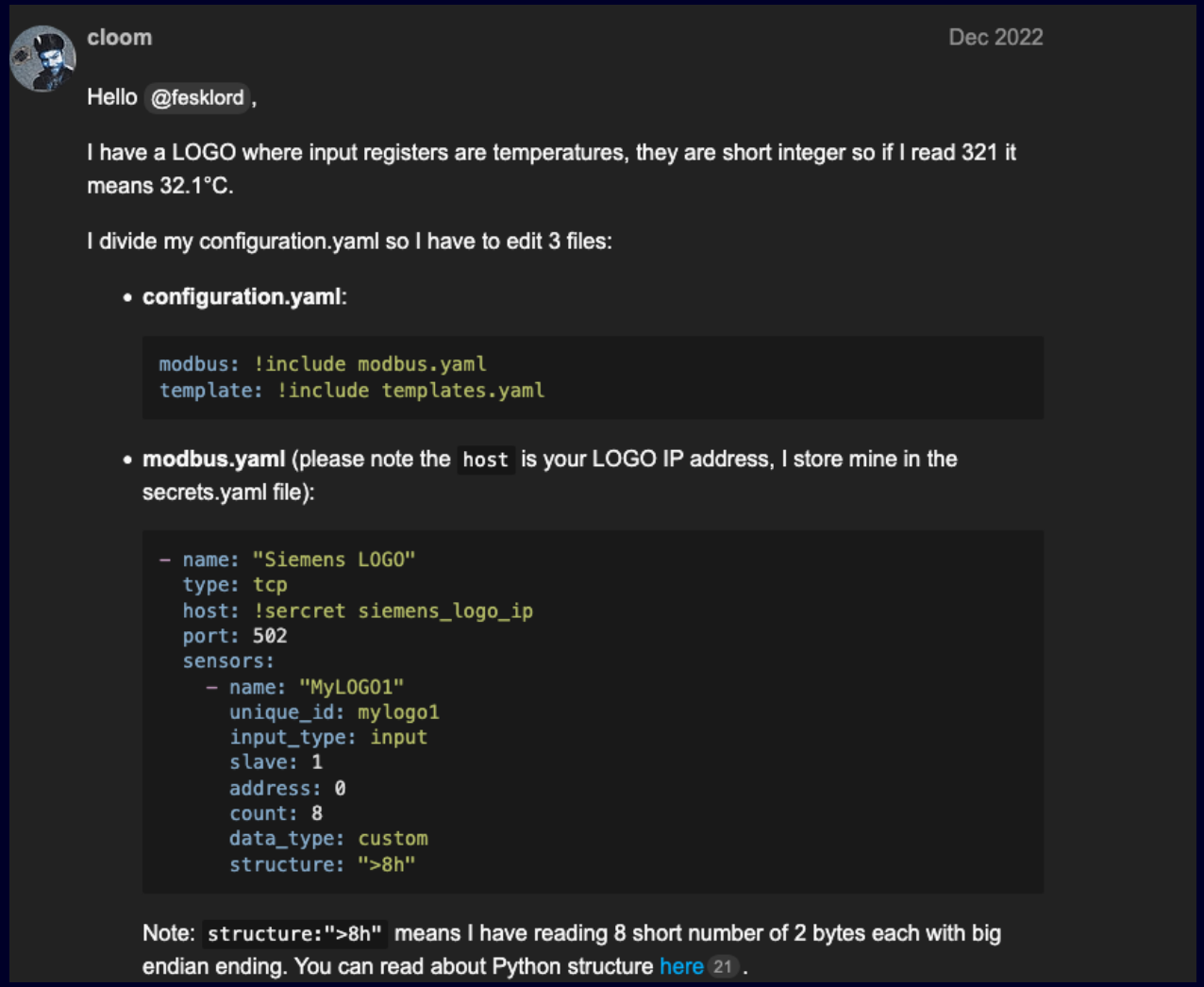
and

```
data_type: float32
count: 2
```



SIEMENS

# Some more Siemens Devices don’t hurt



**cloom** Dec 2022

Hello @fesklord,

I have a LOGO where input registers are temperatures, they are short integer so if I read 321 it means 32.1°C.

I divide my configuration.yaml so I have to edit 3 files:

- **configuration.yaml**:

```
modbus: !include modbus.yaml
template: !include templates.yaml
```

- **modbus.yaml** (please note the `host` is your LOGO IP address, I store mine in the secrets.yaml file):

```
- name: "Siemens LOGO"
  type: tcp
  host: !sercret siemens_logo_ip
  port: 502
  sensors:
    - name: "MyLOGO1"
      unique_id: mylogo1
      input_type: input
      slave: 1
      address: 0
      count: 8
      data_type: custom
      structure: ">8h"
```

Note: `structure:">8h"` means I have reading 8 short number of 2 bytes each with big endian ending. You can read about Python structure here 21.

- As I read all 8 input registers at once and then use a template to separate the values, then I have less modbus requests made to the LOGO. So I divide the values in **templates.yaml**:

```
- name: Heater temperature
  state: "{{ states('sensor.mylogo1').split(',')[0] | multiply(0.1) }}"
  state_class: measurement
  unit_of_measurement: °C
- name: External temperature
  state: "{{ states('sensor.mylogo1').split(',')[1] | multiply(0.1) }}"
  state_class: measurement
  unit_of_measurement: °C
# and so on until states('sensor.mylogo1').split(',')[7]
```

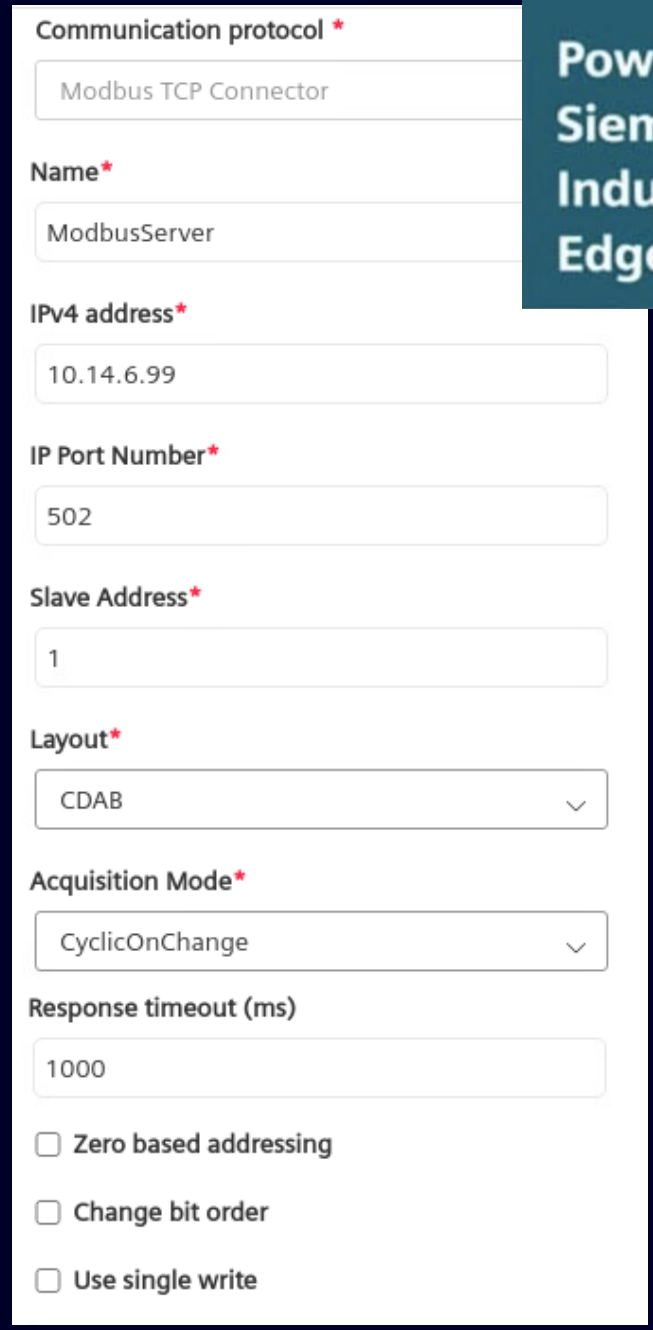
I hope that helps and do not hesitate to reach out if you have any question.



SIEMENS

# What are all these people doing?

```yaml
modbus:
  - name: hub1
    type: tcp
    host: IP_ADDRESS
    port: 502
    binary_sensors:
      - name: my_relay
        address: 100
        device_class: door
        input_type: coil
        scan_interval: 15
```




Powered by Siemens Industrial Edge



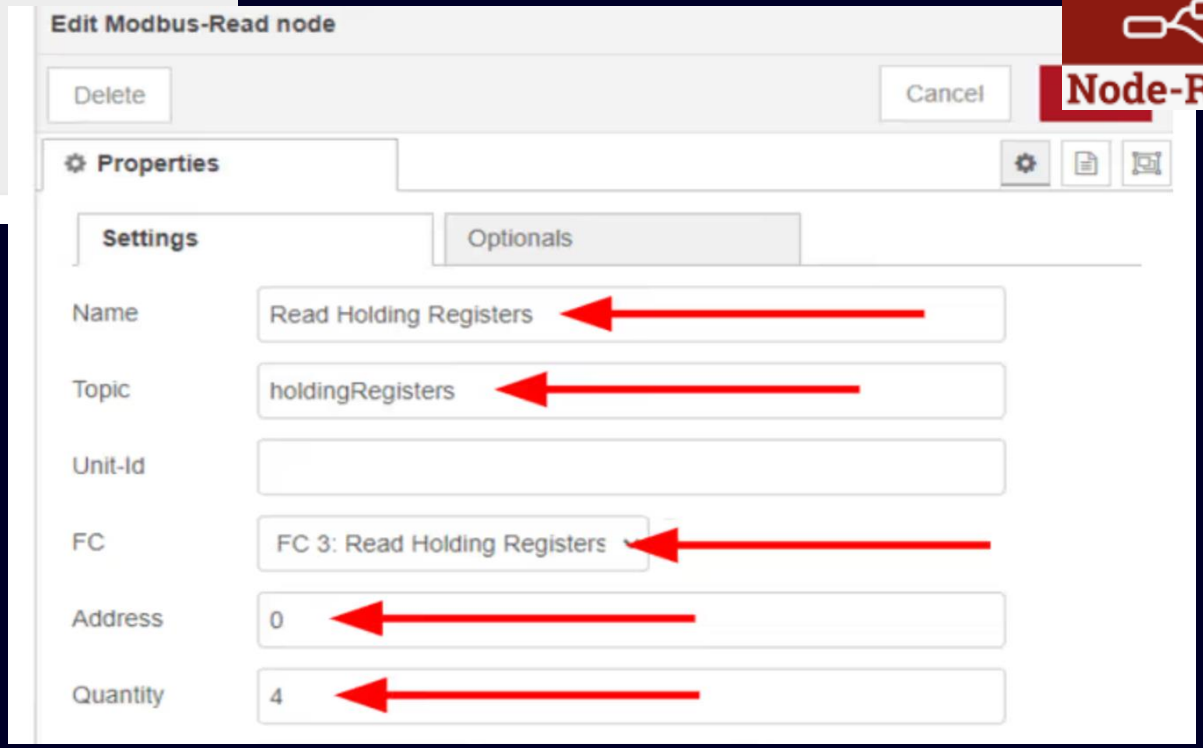
SIEMENS

# What are all these people doing?

HIVEMQ EDGE

```
<tag>
    <tagDefinition>
        <unitId>0</unitId>
        <startIdx>11</startIdx>
        <readType>HOLDING_REGISTERS</readType>
        <dataType>INT_64</dataType>
    </tagDefinition>
    <tagName>myTag</tagName>
</tag>
```

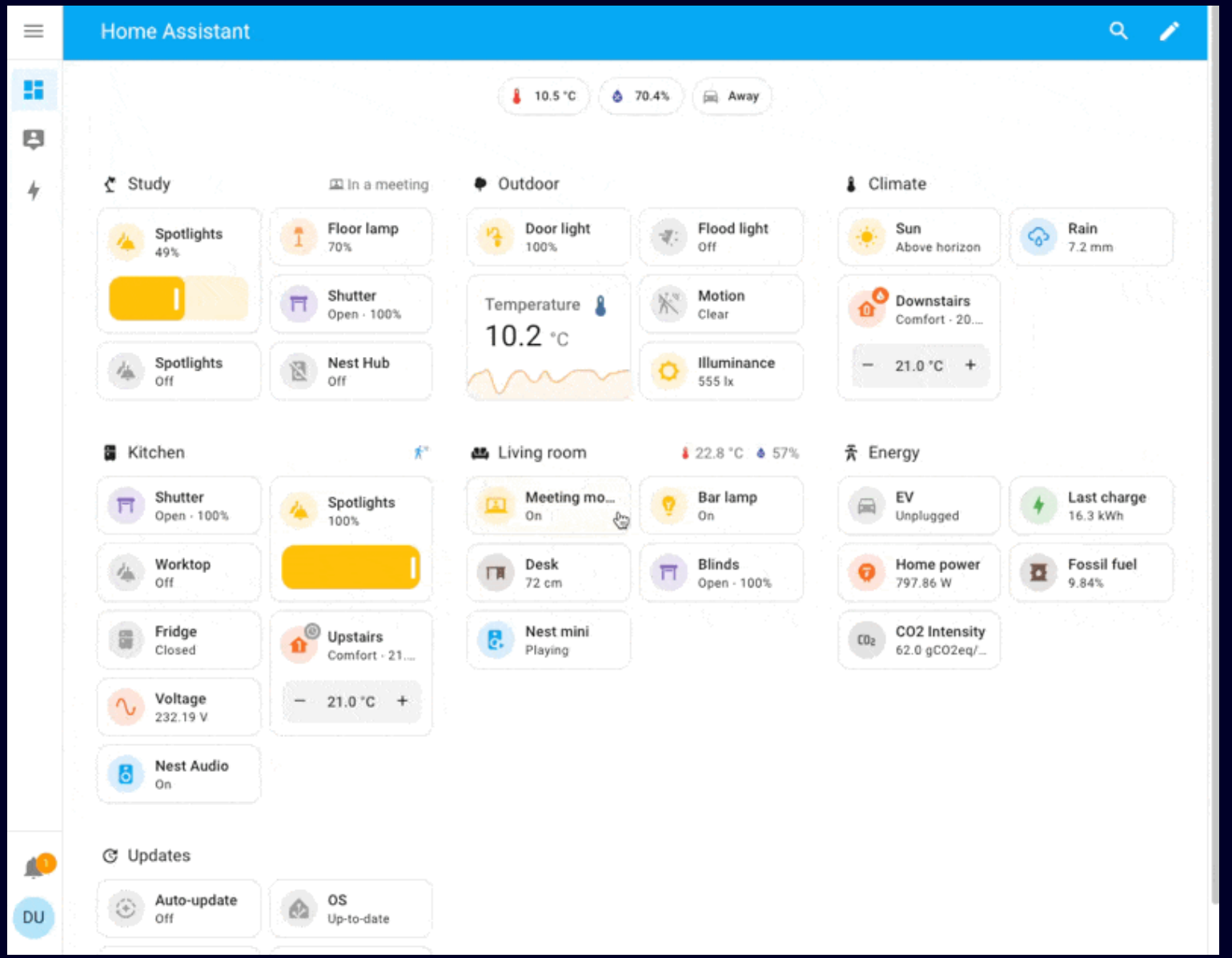
Node-RED






SIEMENS

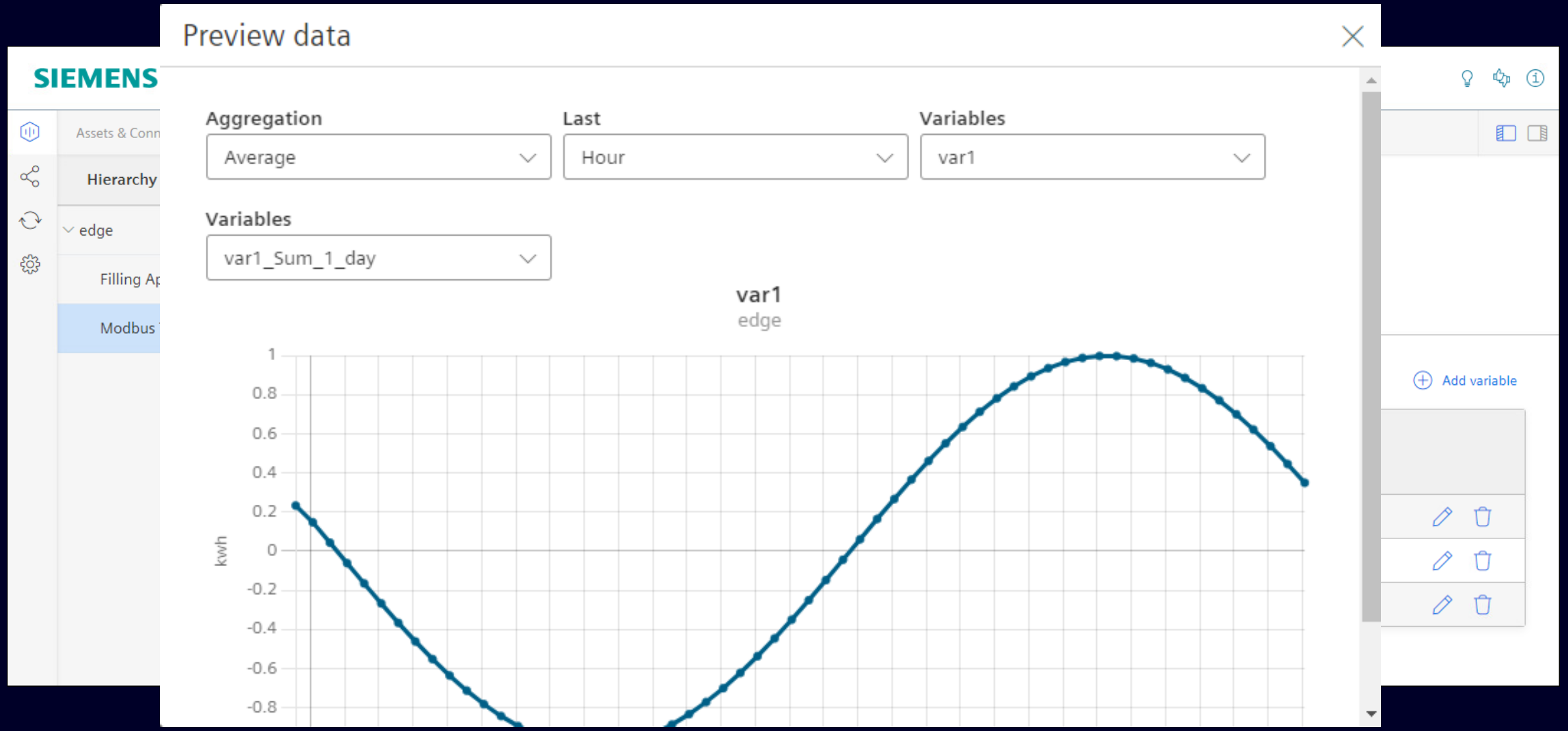
# End Goal






SIEMENS

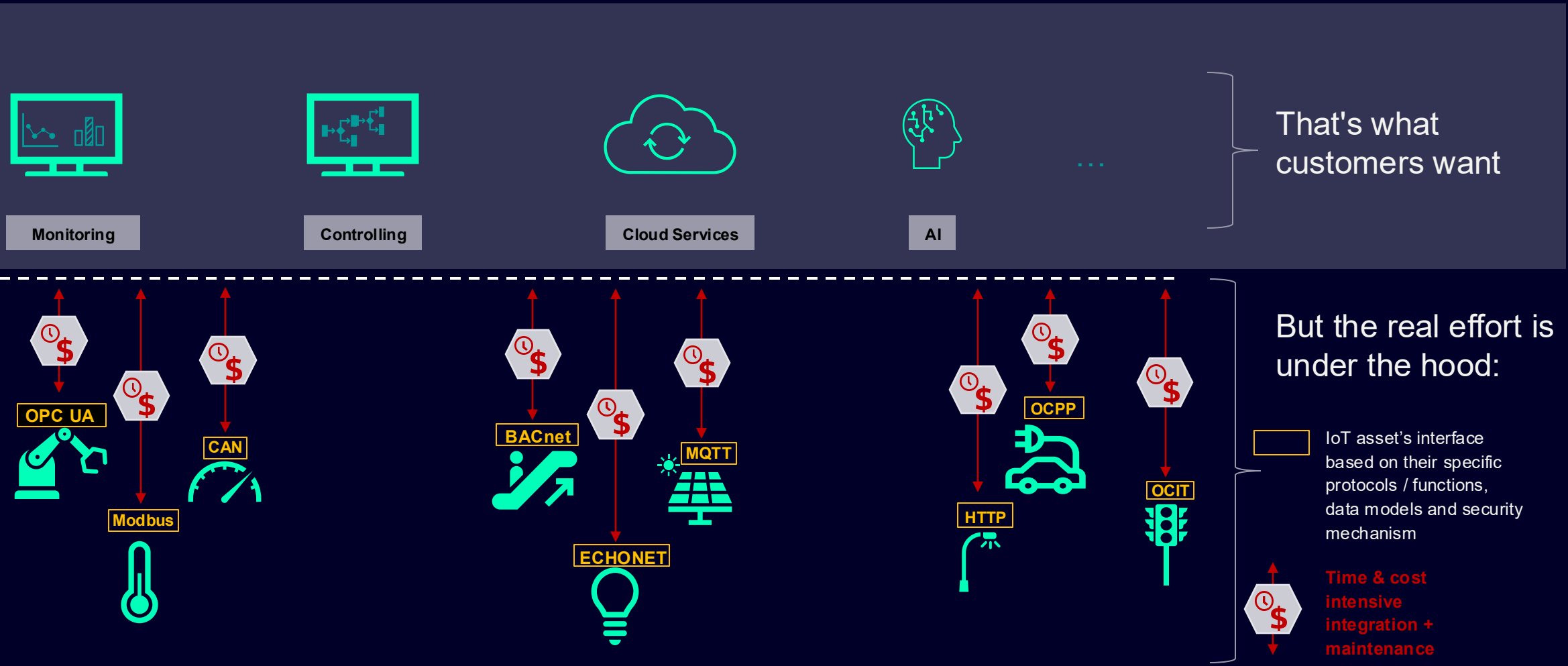
# End Goal






SIEMENS

# There is a common problem if you want to use Things smart: Onboarding






SIEMENS

# Quantifying the Problem

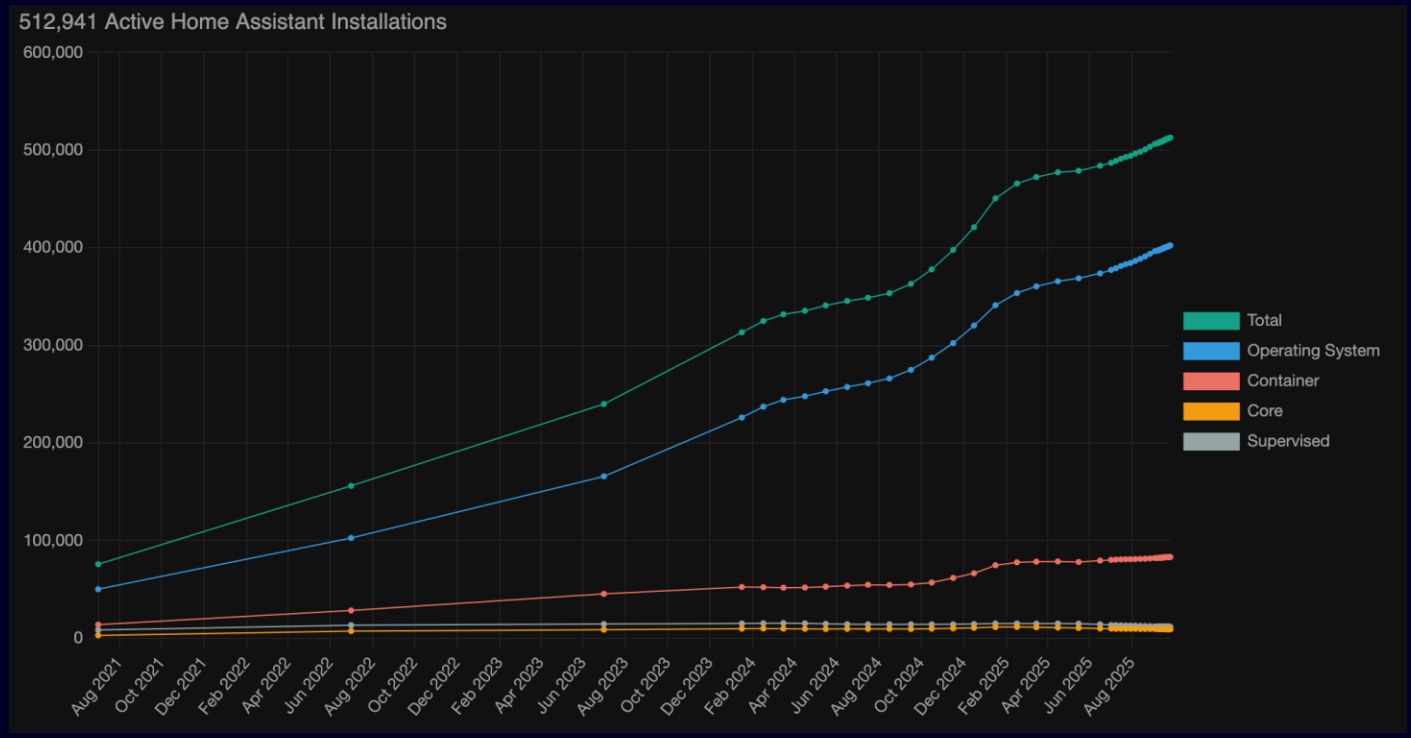
Modbus

The Modbus integration was introduced in Home Assistant pre 0.7, and it's used by 2.6% of the active installations.

Its IoT class is Local Polling.

View source on GitHub

View known issues




13 000 Similar Cases for 1 Protocol in 1 Platform! (2.6% of 500k)
Also: 14.8k in ioBroker

For us, smart home is not the target market,
but it is not that different in more commercial settings



SIEMENS

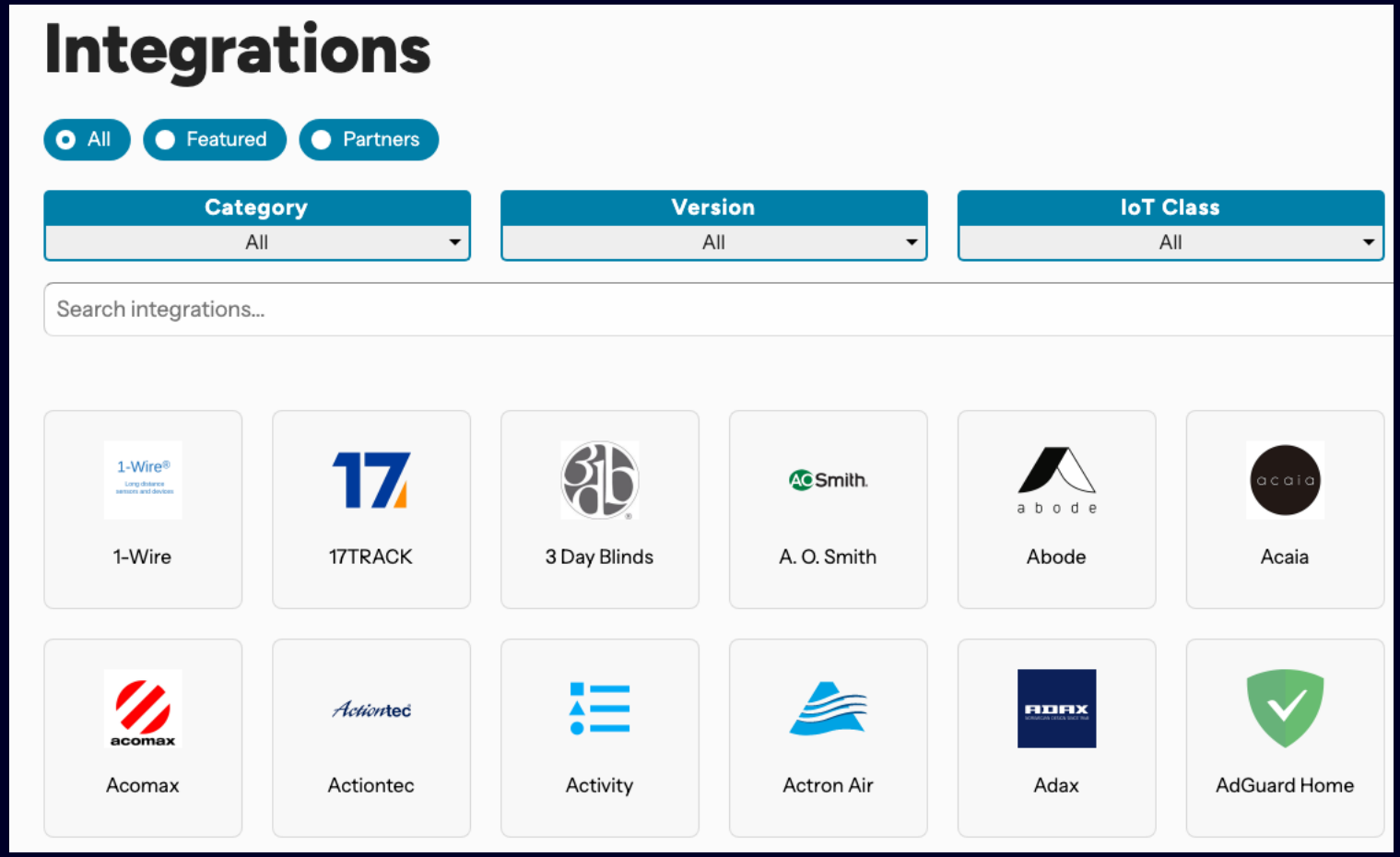
# Why not build integrations for these platforms?

Sure it is *possible*!

Two problems, of course ☺

1. Which device?
2. Which platform?

Also, it doesn't work seamlessly when there is no discovery built-in.



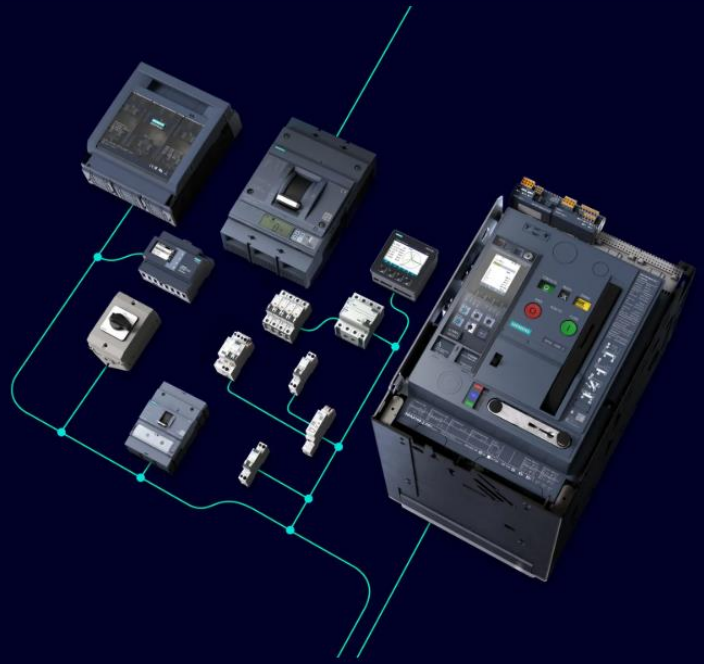

https://www.home-assistant.io/integrations/



SIEMENS

# 1. Which device?

Rather huge device portfolio!



Current protection

- 3RW soft starter
- ET 200SP e-Starter
- ET 200SP motor starter
- 3RM1 motor starter
- M200D motor starter
- ET 200pro motor starter
- Solid-state switching devices
- SIRIUS Starter Kits hybrid switching technology

Motor starters

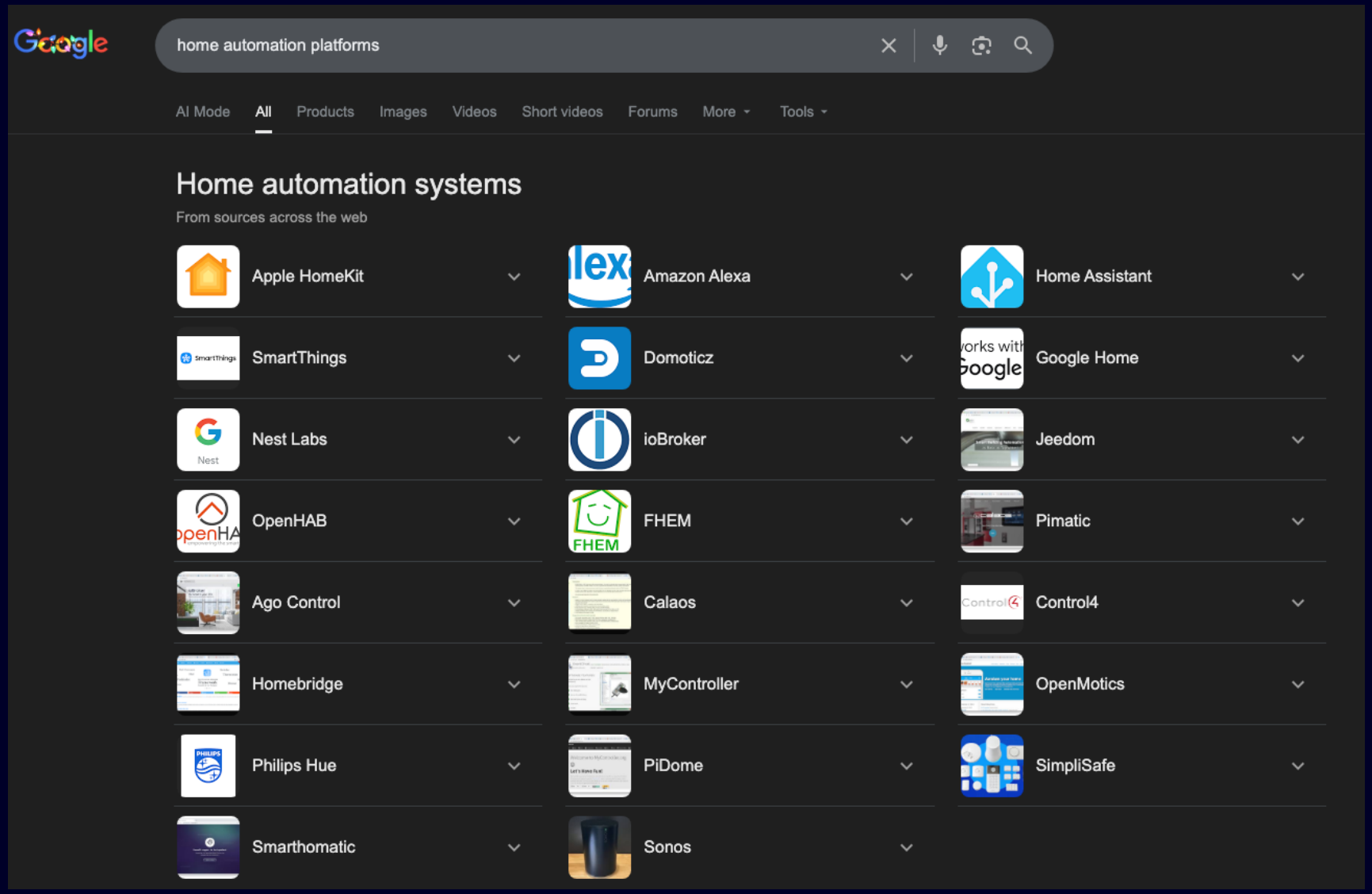
Each is a family with many subtypes

Servo drives



SIEMENS

# 2. Which platform?




- Not all are platforms
- Still too many
- All with their own device modeling language



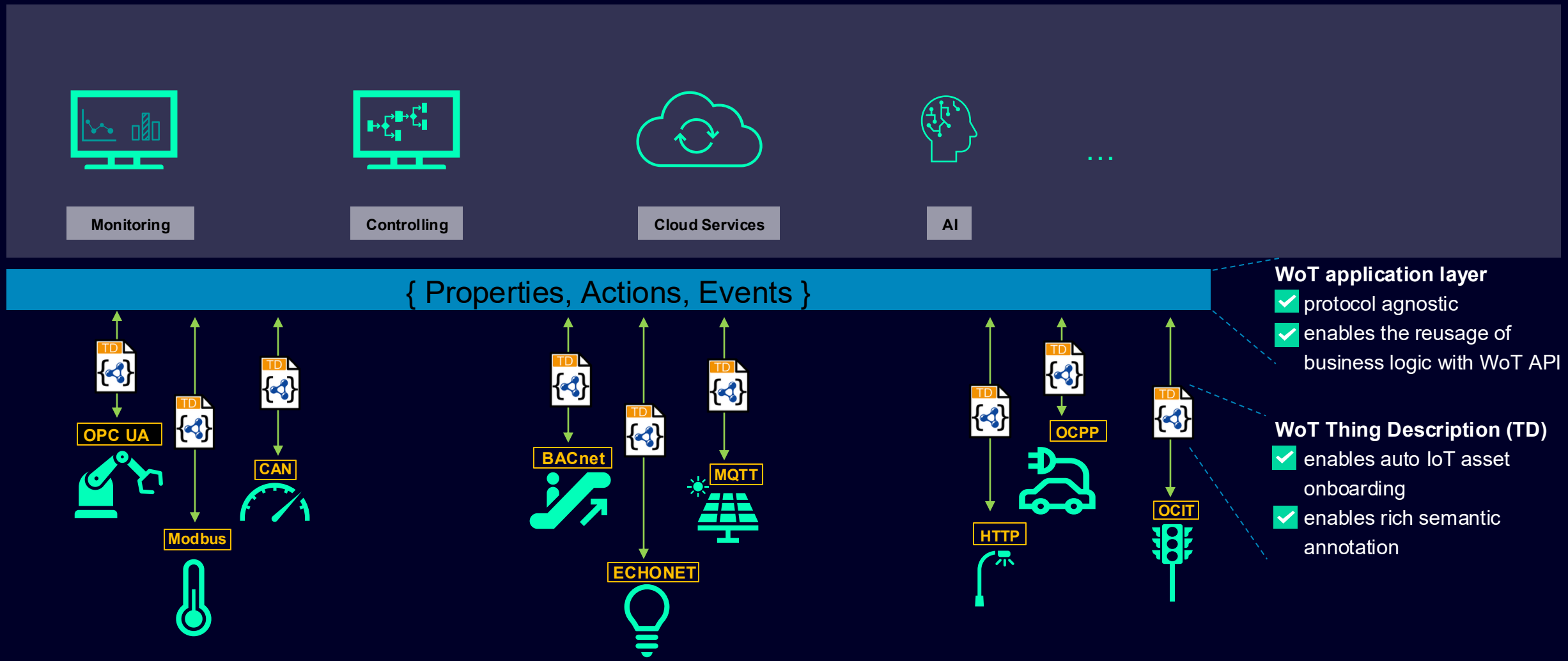
SIEMENS

# Web of Things to the rescue



SIEMENS

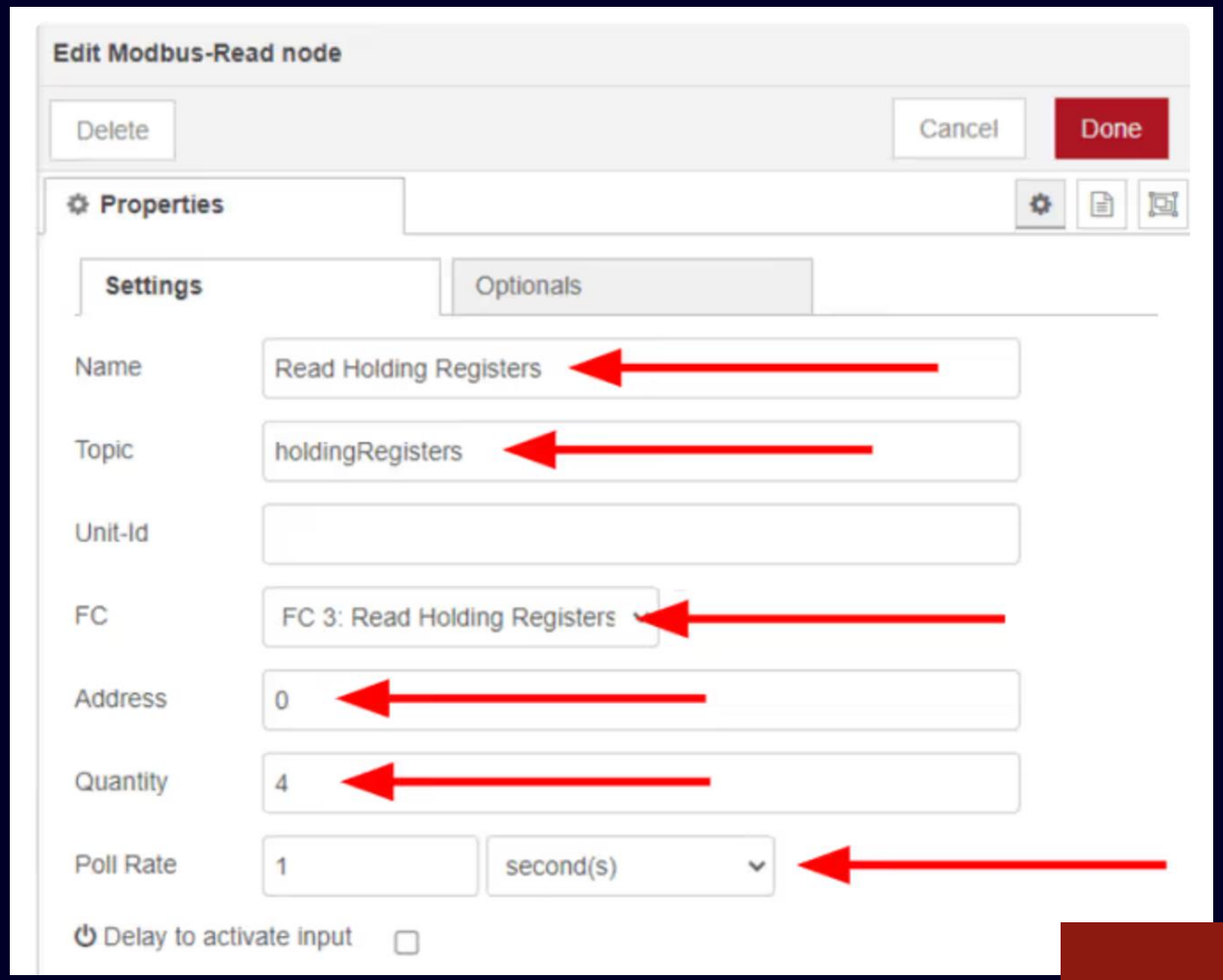
# Tackling the onboarding problem






SIEMENS

# Example TD for the Node-RED example from before




Node-RED

TD Extract

does not repeat

```
"base":"modbus+tcp://192.168.0.10:502",
...
{
  "href": "0?quantity=4",
  "modv:function":"readHoldingRegisters",
  "modv:type":"xsd:float",
  "modv:mostSignificantByte": true,
  "modv:mostSignificantWord": true
}
```

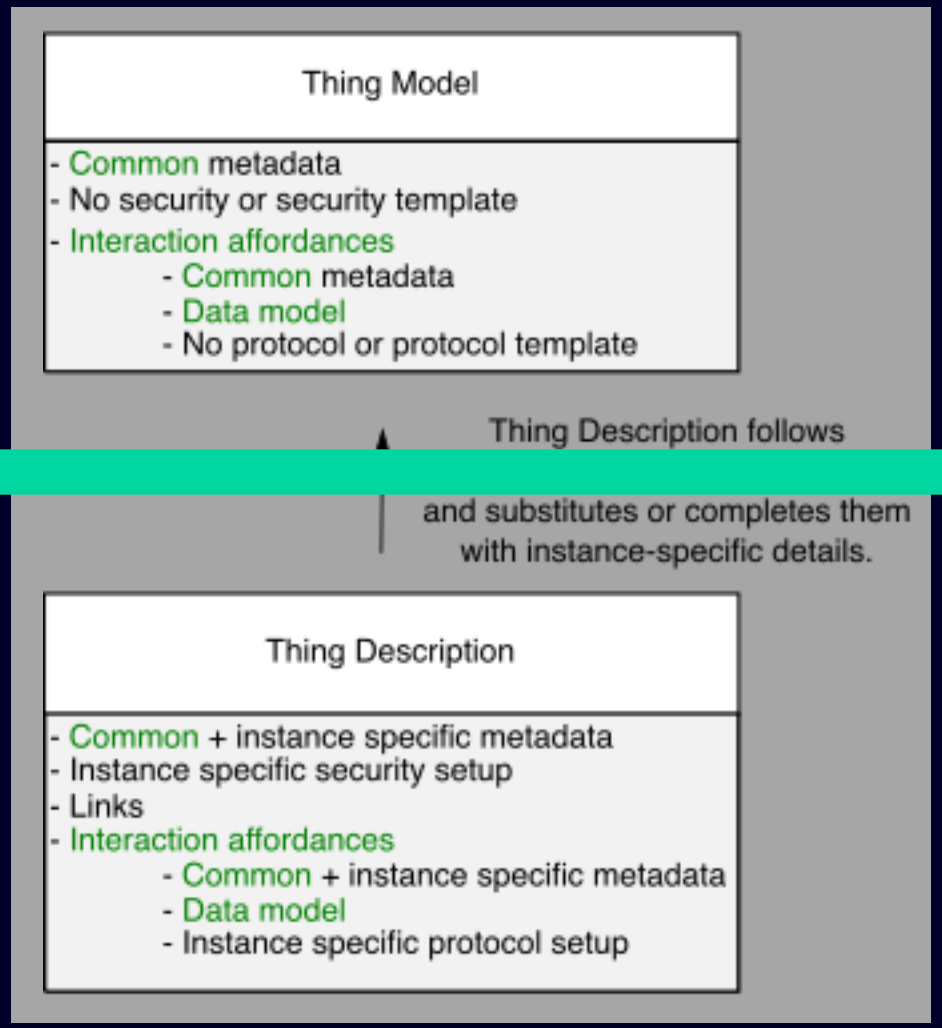
repeats for each instance



SIEMENS

# Concentrating on the Thing Model

Device Model →




→ Found in a *Catalog*

Device Instance →

→ Found in the Field



SIEMENS

# What kind of Thing Model though?

## Generic device/data model

```
{
  "@context": ["https://www.w3.org/2022/wot/td/v1.1"],
  "@type": "tm:ThingModel",
  "title": "Switchable",
  "properties": {
    "on": {
      "description": "Whether the switch is on or off.",
      "type": "boolean"
    }
  },
  "actions": {
    "toggle": {
      "description": "Toggles/inverts the current 'on' state.",
      "output": {
        "title": "New 'on' state",
        "type": "boolean"
      }
    },
    "switch-on-for-duration": {
      "description": "Switches the switchable on for a given duration",
      "input": {
        "title": "Duration in seconds",
        "type": "integer",
        "unit": "time:seconds"
      }
    }
  }
}
```

Example from Eclipse Ditto

Focus of today

## Specific device, just not instantiated

```
{
    "@context": ["https://www.w3.org/2022/wot/td/v1.1"],
    "@type": "tm:ThingModel",
    "title": "3NACOM_FUSE",
    "base": "modbus://{{IP}}:{{PORT}}",
    "securityDefinitions": {"nosec_sc": {"scheme": "nosec"}},
    "security": "nosec_sc",
    "properties": {
        "I_L1_AVERAGE": {
            "title": "Average Current",
            "readOnly": true,
            "type": "number",
            "unit": "om:ampere",
            "forms": [
                {
                    "op": "readproperty",
                    "href": "/",
                    "modbus:unitID": "{{UNITID}}",
                    "modbus:quantity": 2,
                    "modbus:address": 3078,
                    "modbus:type": "number",
                    "modbus:entity": "HoldingRegister",
                    "modbus:zeroBasedAddressing": false,
                    "modbus:pollingTime": 2000
                }
            ]

        }
    }
```

Example from Siemens



SIEMENS

# Thing Model catalog

Tooling, home and services

https://github.com/wot-oss/tmc



SIEMENS

# Some Design Goals

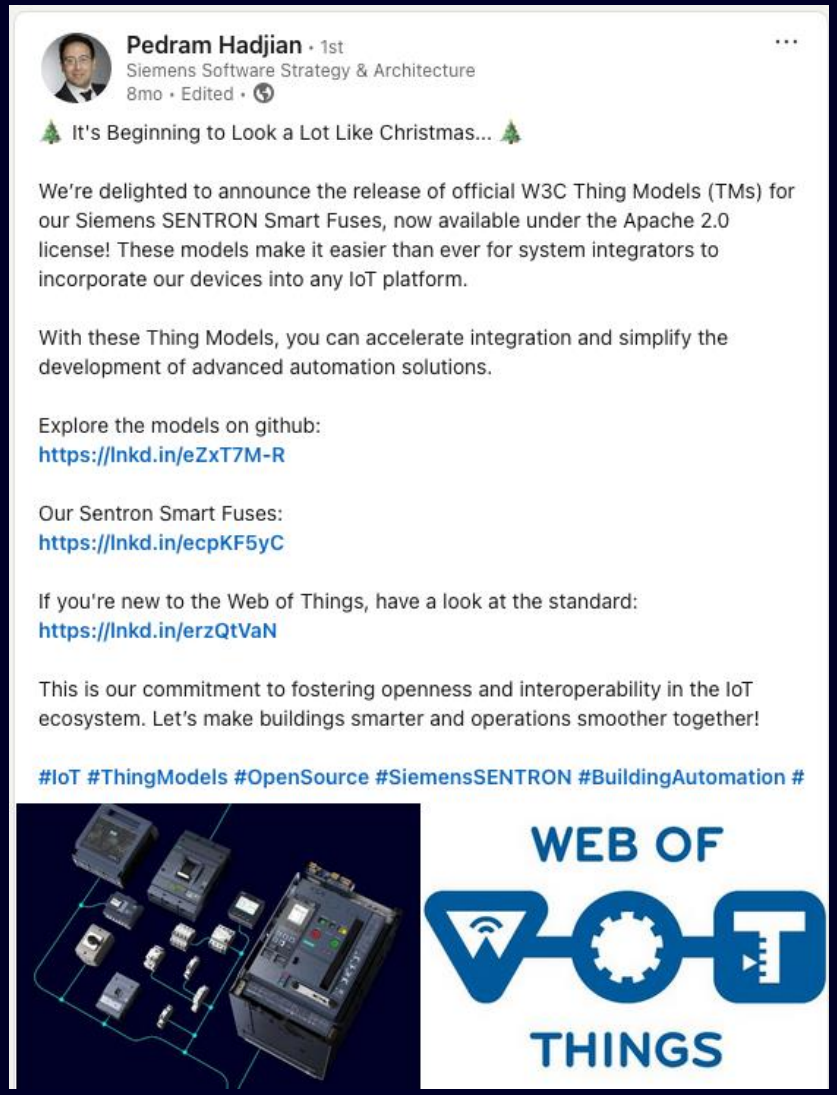
- **Authors** contributing TMs of **device models** of **manufacturers**
- Finding a TM of a device “in front of you”
- Supporting online and offline deployment
- Git and CI/CD friendly
- No domain-specific requirements
- Federated deployment (merging multiple repositories in one API)

Importantly: Not centralizing TMs in one place.

- Provide the tooling to you to manage your own TM Catalog

You can find more at https://github.com/wot-oss/proposal/issues/8

# First sightings in the wild:



Pedram Hadjian · 1st
Siemens Software Strategy & Architecture
8mo · Edited ·

🎄 It's Beginning to Look a Lot Like Christmas... 🎄

We're delighted to announce the release of official W3C Thing Models (TMs) for our Siemens SENTRON Smart Fuses, now available under the Apache 2.0 license! These models make it easier than ever for system integrators to incorporate our devices into any IoT platform.

With these Thing Models, you can accelerate integration and simplify the development of advanced automation solutions.

Explore the models on github:
https://lnkd.in/eZxT7M-R

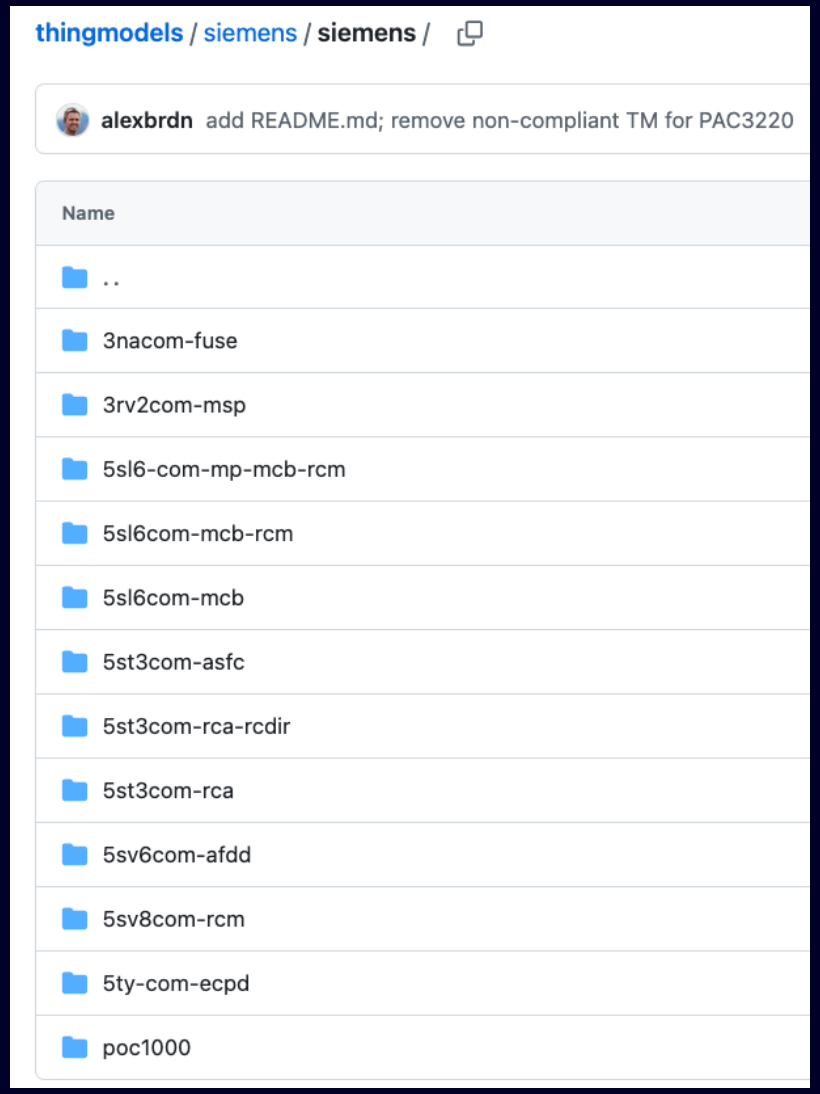
Our Sentron Smart Fuses:
https://lnkd.in/ecpKF5yC

If you're new to the Web of Things, have a look at the standard:
https://lnkd.in/erzQtVaN

This is our commitment to fostering openness and interoperability in the IoT ecosystem. Let's make buildings smarter and operations smoother together!

#IoT #ThingModels #OpenSource #SiemensSENTRON #BuildingAutomation #





thingmodels / siemens / siemens /

alexbrdn add README.md; remove non-compliant TM for PAC3220

Name

..
3nacom-fuse
3rv2com-msp
5sl6-com-mp-mcb-rcm
5sl6com-mcb-rcm
5sl6com-mcb
5st3com-asfc
5st3com-rca-rcdir
5st3com-rca
5sv6com-afdd
5sv8com-rcm
5ty-com-ecpd
poc1000

SENTRON Smart Fuses

Apache License

Manufacturer Publication

https://github.com/wot-oss/thingmodels



SIEMENS

# Thing Model Catalog: Tooling

README Apache-2.0 license

## Thing Model Catalog CLI

go report A+ release v0.1.2 go.dev docs license scan passing


Find, use and contribute device descriptions for industrial IoT devices!

```
Usage:
  tmc [command]

Available Commands:
  attachment  Manage TM attachments
  check       Check the integrity of all or only named resources in repository's inte
  completion  Generate the autocompletion script for the specified shell
  copy        Copy multiple TMs and their attachments from one repository to another
  create-si   Create or update search index
  delete      Delete a TM by id
  export      Export multiple TMs from a catalog and, optionally, their attachments
  fetch       Fetch a TM by name or id
  help        Help about any command
  import      Import a TM or a directory with TMs into a catalog
  index       Refresh the repository's internal index, if it has one
  list        List TMs in catalog
  repo        Manage repositories
  search      Search full text of TMs in catalog using bleve search engine
  serve       Start a REST API server
  validate    Validate a TM before importing
  version     Show tmc version information
  versions    List available versions of the TM with given name
```

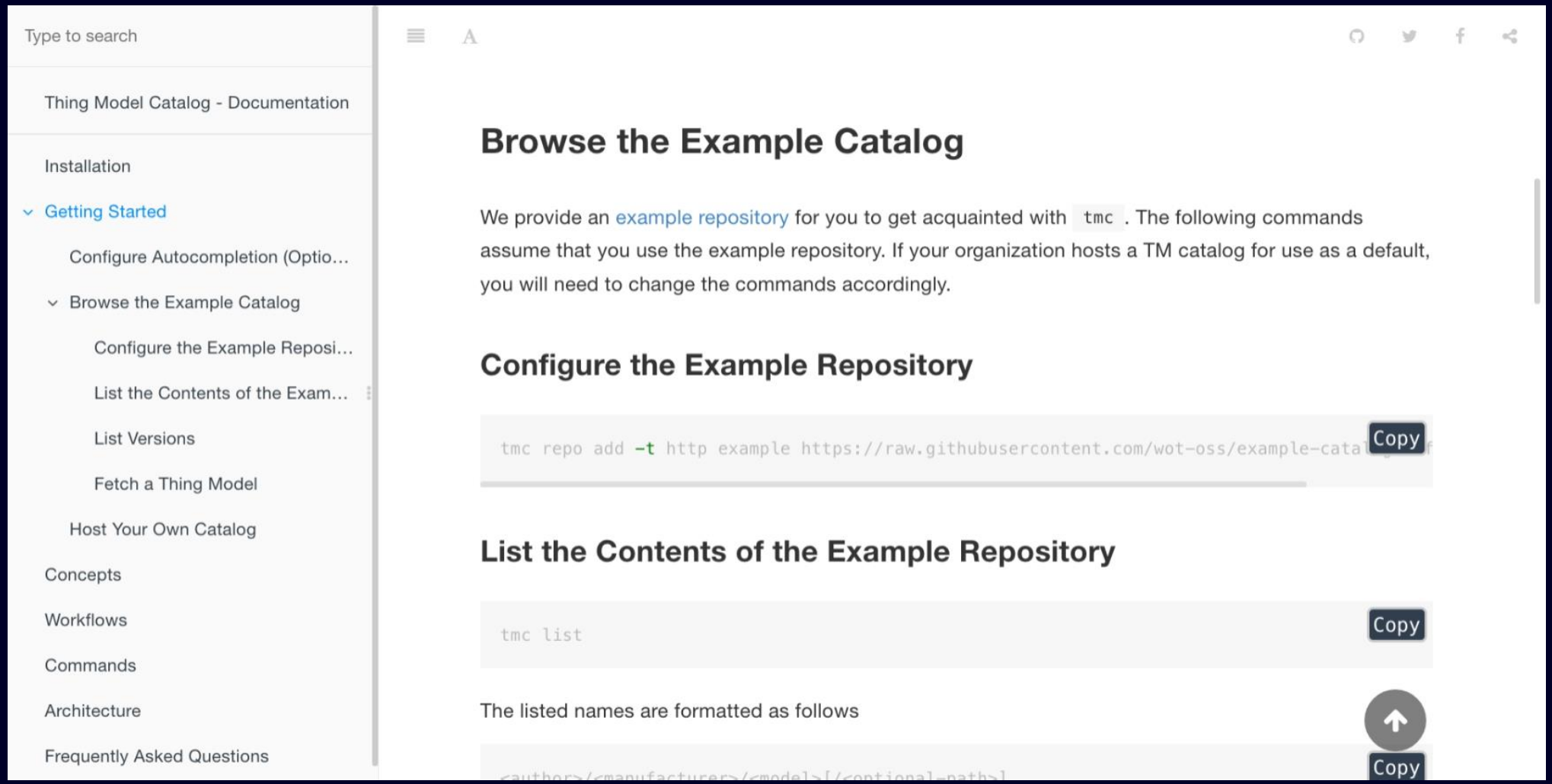
https://www.github.com/wot-oss/tmc

Also available with a REST API




SIEMENS

# Getting Started Documentation

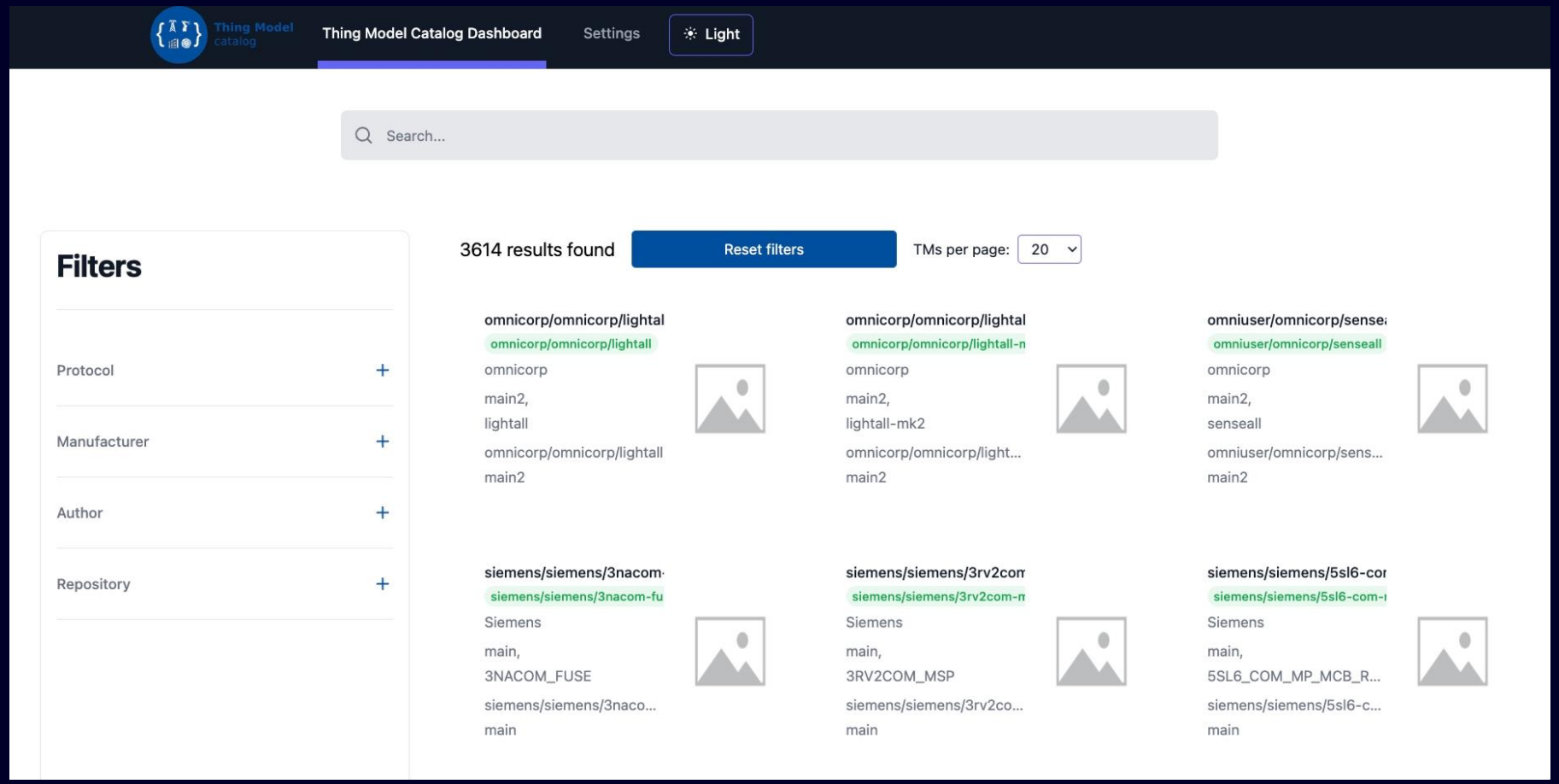



https://wot-oss.github.io/tmc/gettingstarted



SIEMENS

# Web UI (Quite new, not perfect yet)




- Configurable TMC endpoint
- Customizable Theme
- Follow TMC REST API
- Open in EdiTDor, Playground



SIEMENS

# TM Content

- We catalog based on the author (like a namespace), manufacturer and product name (Model Part Number). So schema.org annotations are required

```
"schema:author": {
  "schema:name": "Siemens"
},
"schema:manufacturer": {
  "schema:name": "Siemens"
},
"schema:mpn": "5ST3COM_RCA”
```

- The id you will have in your TM will be overwritten to include a timestamp and a hash based on the content. author/manufacturer/mpn will be also used

```
"id": "siemens/siemens/5st3com-rca/v1.0.0-20240802121832-941071515746.tm.json",
```

- We can sort TMs, guarantee uniqueness and more. See https://github.com/wot-oss/proposal/issues/10

# Let’s see it in action!



SIEMENS

# Demo 1: Simple CLI and REST API, Federated Demo

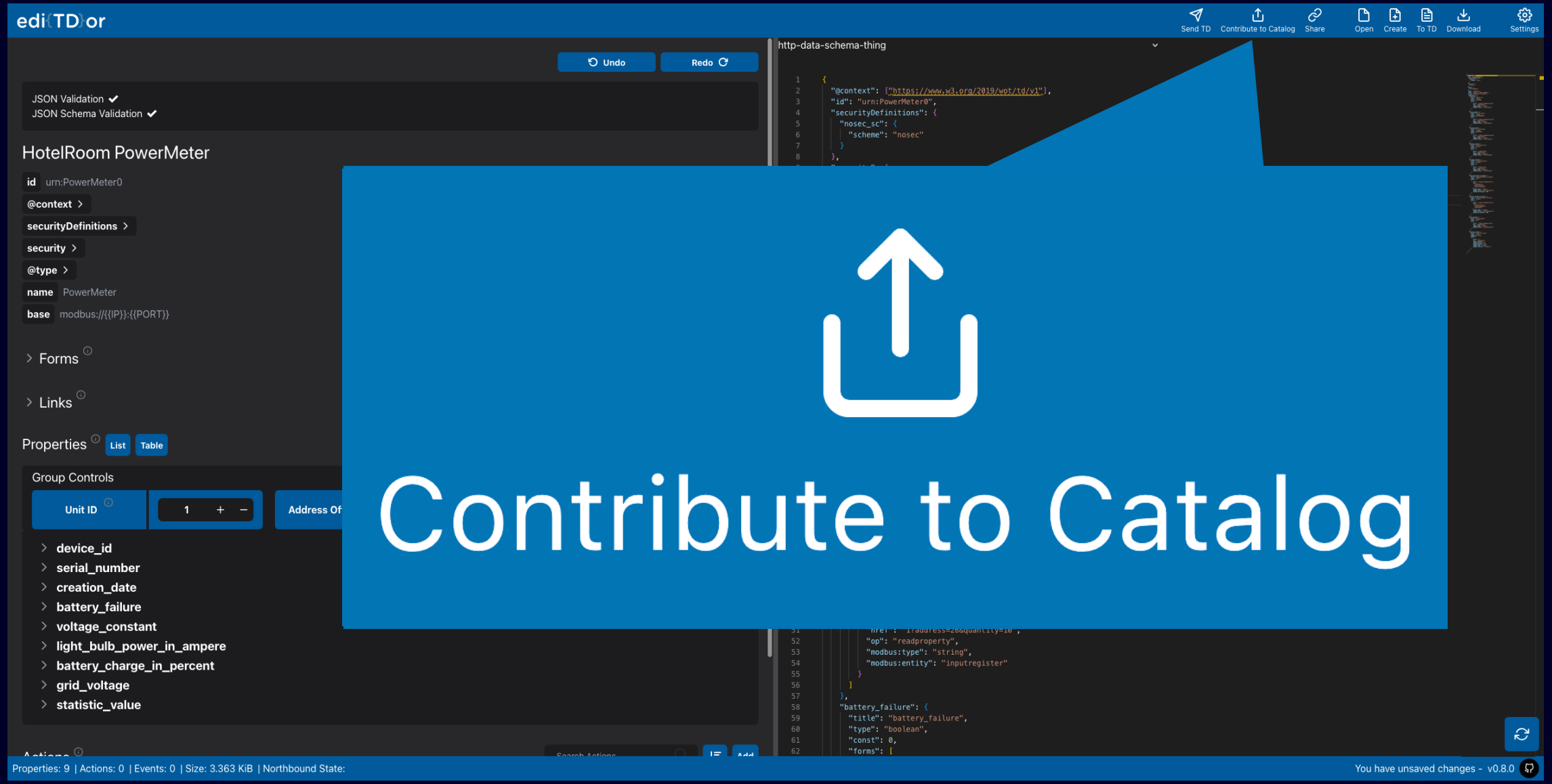
1. Install TMC CLI
2. Add a file repository (local on my computer)
3. Do some operations with CLI
4. Do same operations via REST API
5. Add a git repository (on GitHub.com)
6. Show multiple repositories under one roof

# How to Contribute TMs?



SIEMENS

# Authoring Tools: Easier Creation TMs

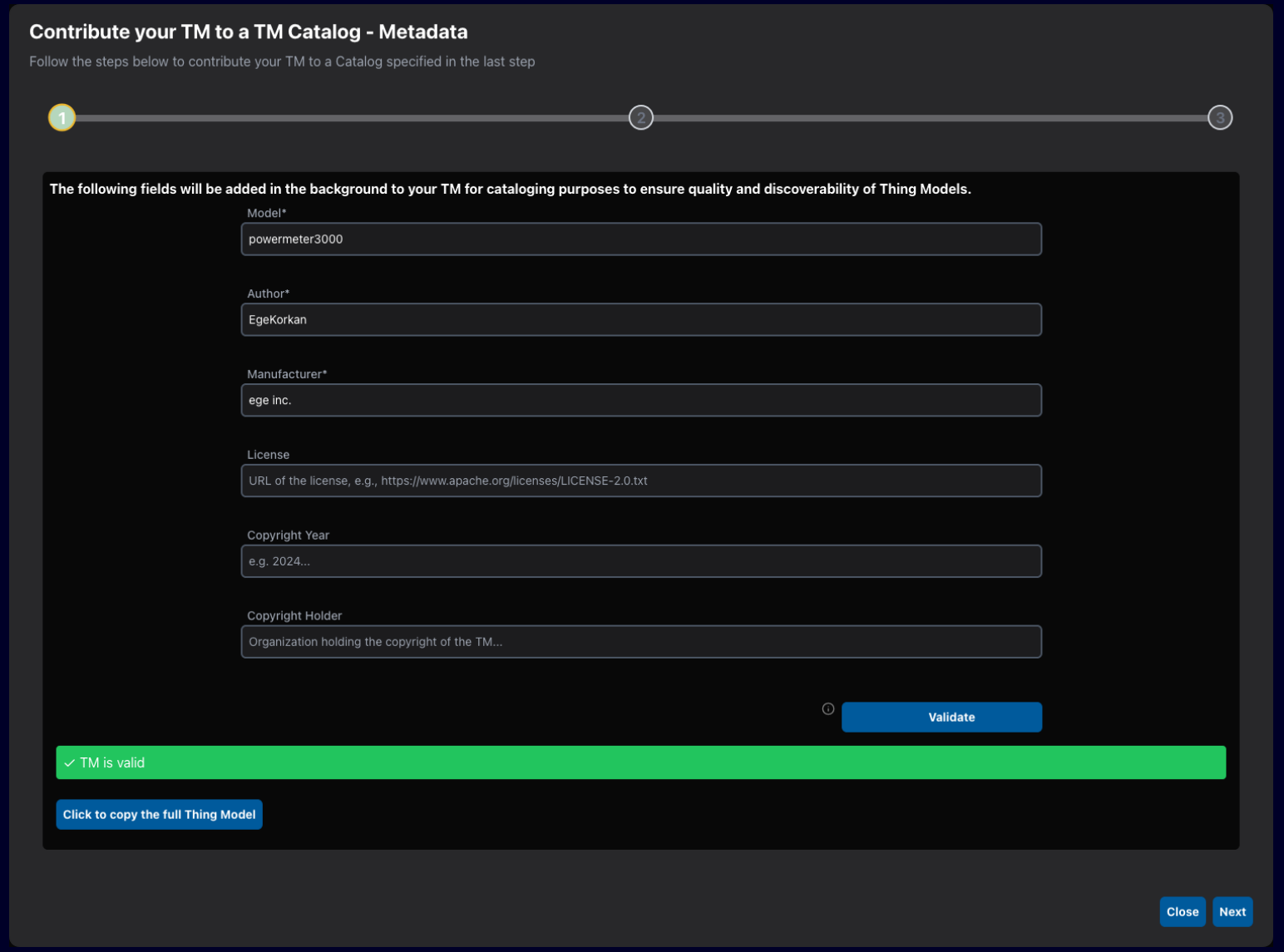



https://eclipse.github.io/editdor /




SIEMENS

# TMC Contributions further simplified with EdiTDor

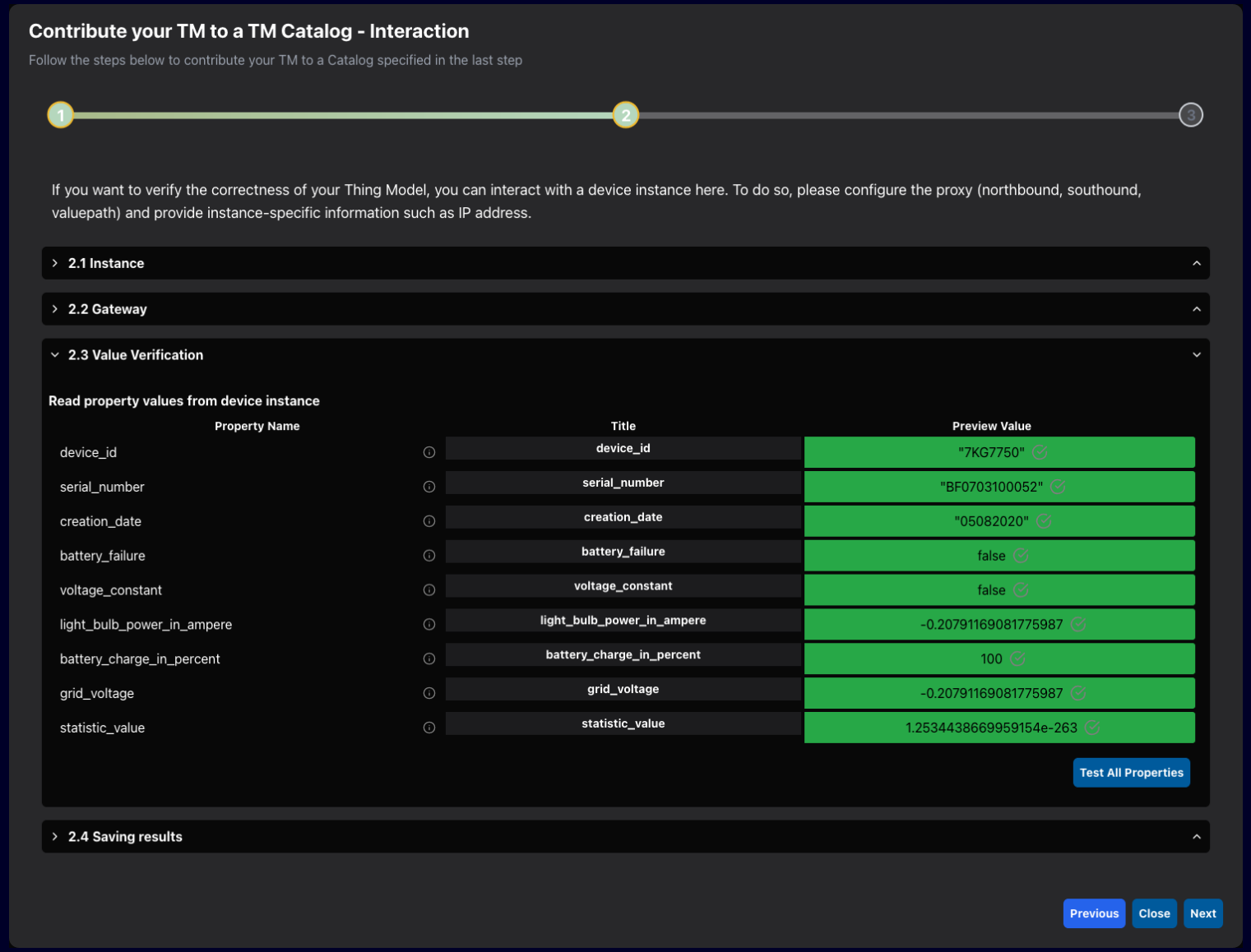






SIEMENS

# TMC Contributions further simplified with EdiTDor

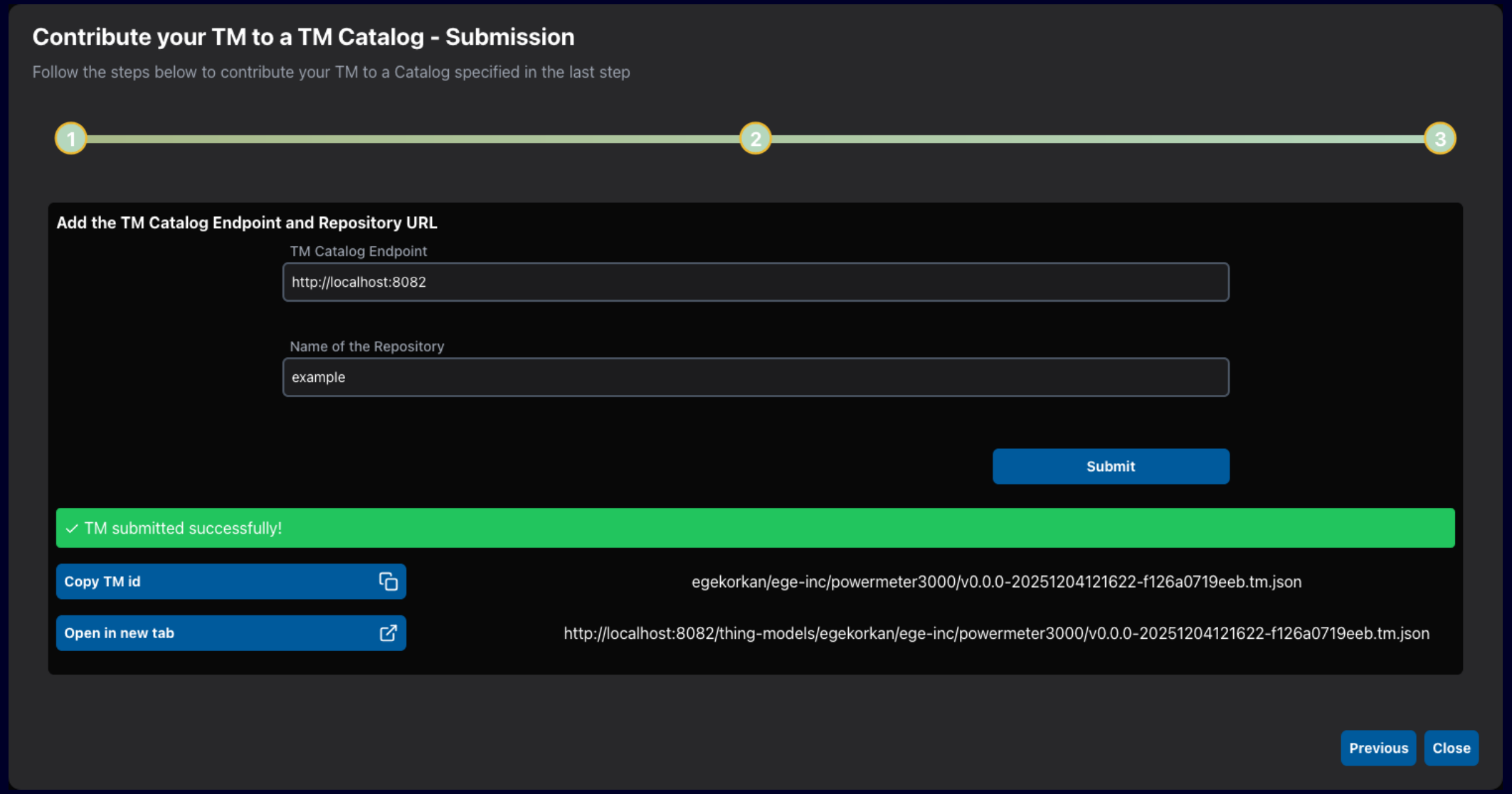






SIEMENS

# TMC Contributions further simplified with EdiTDor







SIEMENS

# Demo 2: Authoring Workflow Supported with Eclipse EdiTDor

1. Start EdiTDor locally
2. Start a Thing (we will use a Modbus Thing for the demo)
    1. Start a gateway in this case
3. Write its TM
4. Start Contribution Workflow
5. Optionally test all the endpoints
6. Contribute to the specified catalog

In general, all parameters typed here can be sent via query parameters or stored in browser local storage.

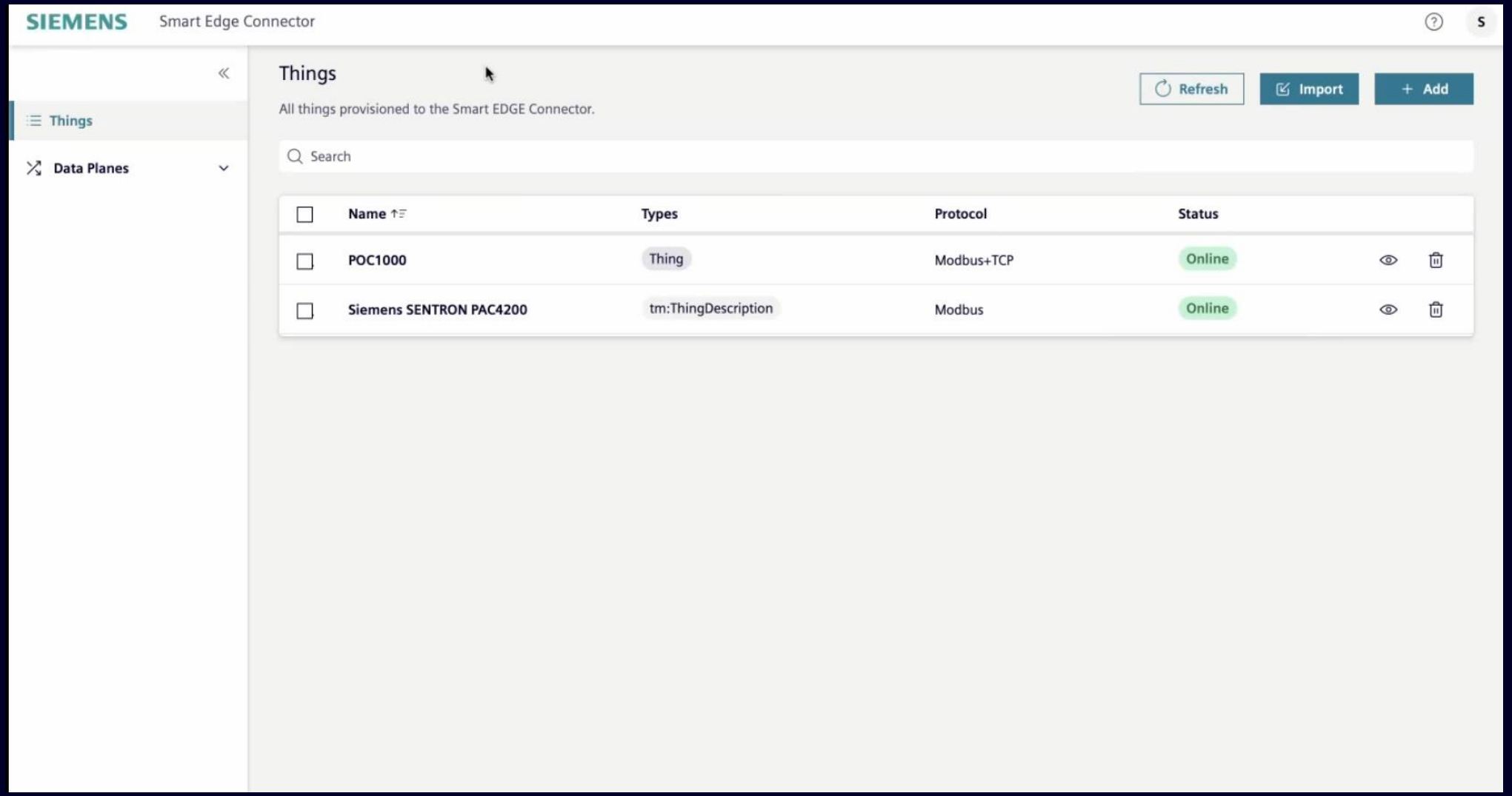
# Using the TMC at Siemens

Extract from Meetup 20




SIEMENS

# Smart Edge Connector runs on an Edge Device

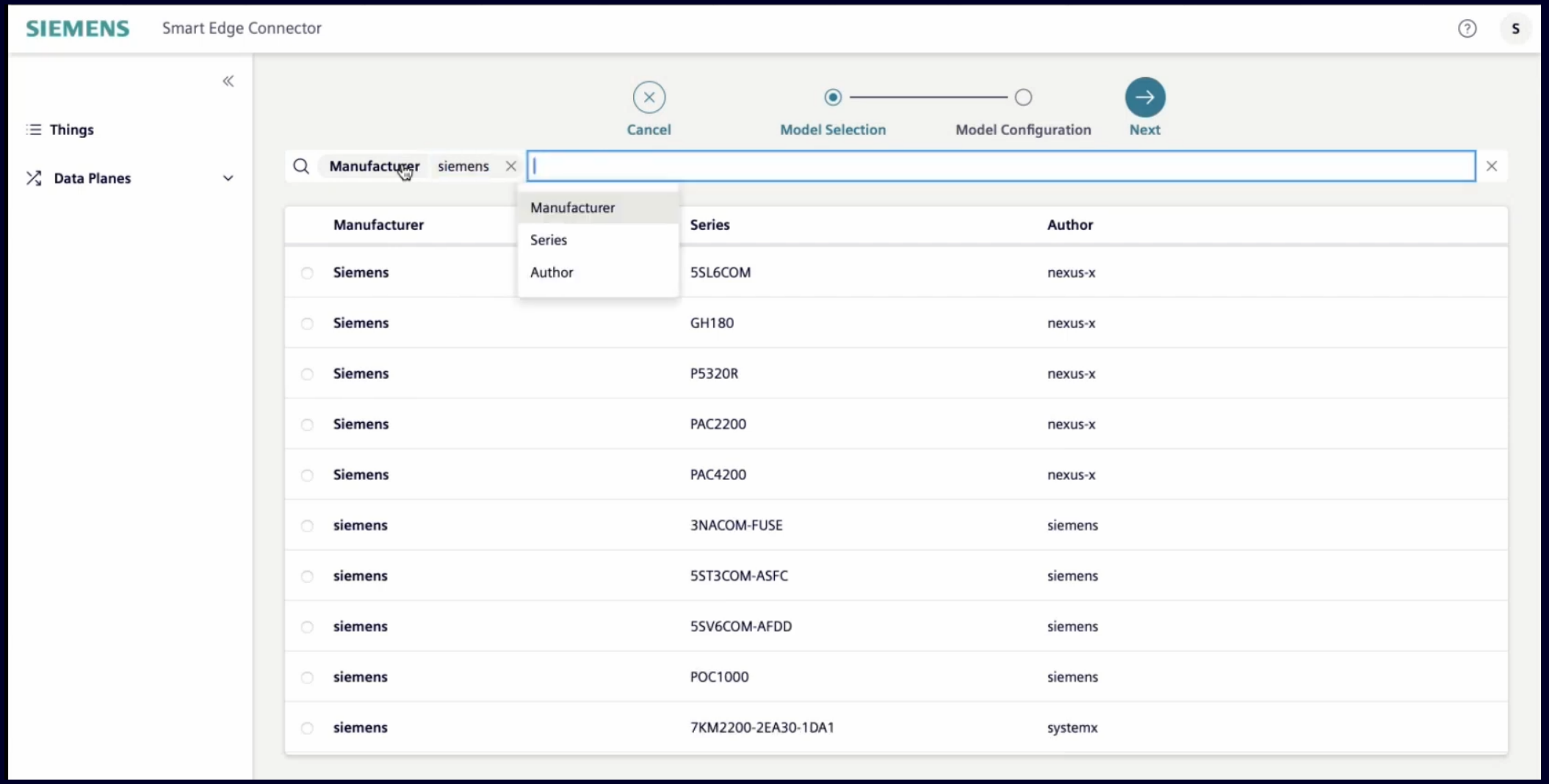



https://www.youtube.com/watch?v=vRdX5ItQHX0




SIEMENS

# We search through a Catalog

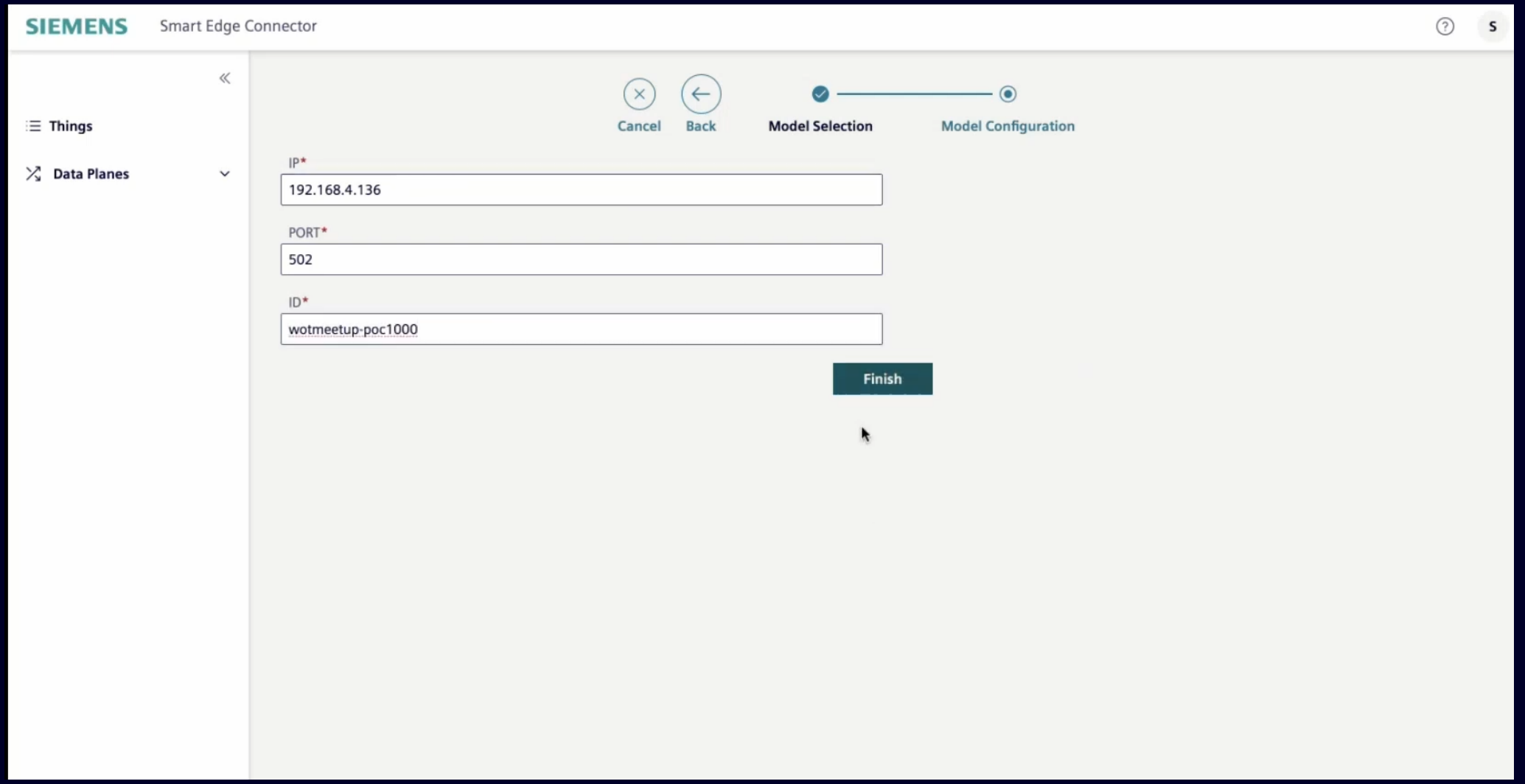



https://www.youtube.com/watch?v=vRdX5ItQHX0




SIEMENS

# We instantiate a TD from TM

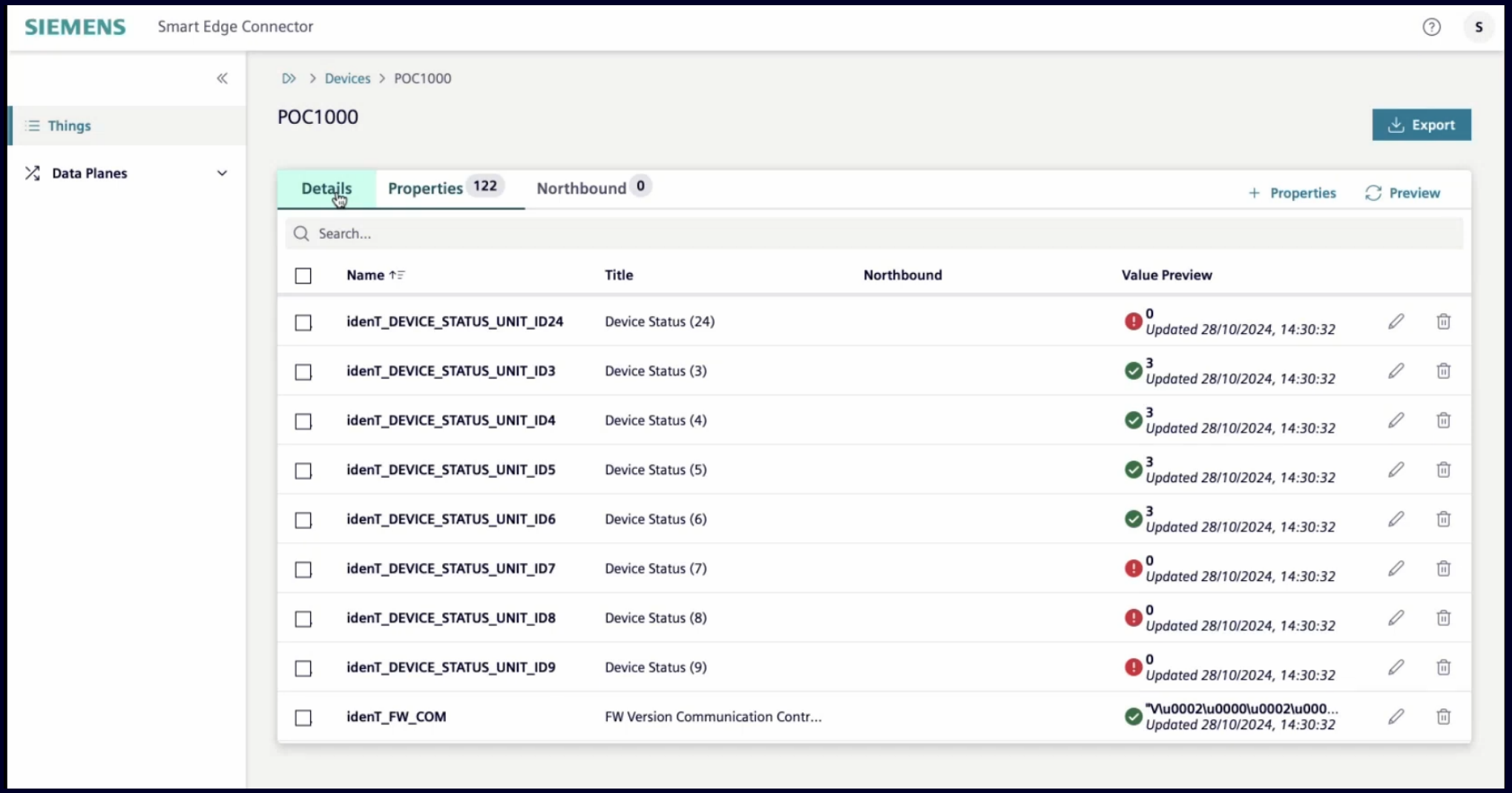



https://www.youtube.com/watch?v=vRdX5ItQHX0




SIEMENS

# Device is onboarded!




https://www.youtube.com/watch?v=vRdX5ItQHX0




SIEMENS

# CG Challenge



SIEMENS

# Hacktoberfest, Advent of Code, now…

We will host a small event for the community to write TMs for the IoT devices around themselves.

As TMC can pull TMs from different repositories, you don't even need to commit them to the CG repository. We will just run a simple bash script to pull your repositories at the end of the challenge.

Instructions, also on our repo:

- Create an empty repository under your own GitHub username
- Install TMC CLI and create a new repository
- Write/generate a TM however you like
- Do `tmc import yourTM.json`
- Your repository will be updated with the TM, table of contents.
- Push the changes
- Create a PR to the CG repository with the `tmc repo add` commands
  - Issues or Discord messages are also fine! We will add a new line with your repo.
- That's it!

My challenge ☺
Tado X Thermostat with Matter




SIEMENS

# Contact

Ege Korkan
SI CTO SSA
ege.korkan@siemens.com



SIEMENS