

Managing Device Descriptions with the Thing Model Catalog

Ege Korkan

Siemens AG

Smart Infrastructure, CTO Office



Onboarding

Commissioning

Integration

Provisioning

Data
Onboarding

Adding a new Device

State of the Art, Focusing on the Modbus TCP Protocol
Home automation forums

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" ] //   
- Heizen Konstanttemperatur" [ readStart="41110", readValueType="int16" ] //   
- Heizen Konstanttemp Absenk" [ readStart="41111", readValueType="int16" ] //   
- Kühlen Konstanttemperatur" [ readStart="41112", readValueType="int16" ] // ]
```

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

```
- 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
```

[HomeAssistant Forum](#)

At least Siemens does
better 😊

Or does not...



vaor Valtor

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

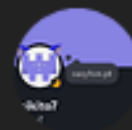
Invalid config for [modbus]: [scale] is an invalid option for [modbus]. Check: modbus->modbus->0->binary_sensors->0->scale. (See /config/configuration.yaml, line 17).

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
```

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: !secret siemens_logo_ip
  port: 502
  sensors:
    - name: "MyLOG01"
      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.

What are all these people doing?

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

Communication protocol *

Modbus TCP Connector

Name *

ModbusServer

IPv4 address *

10.14.6.99

IP Port Number *

502

Slave Address *

1

Layout *

CDAB

Acquisition Mode *

CyclicOnChange

Response timeout (ms)

1000

☐ Zero based addressing

☐ Change bit order

☐ Use single write

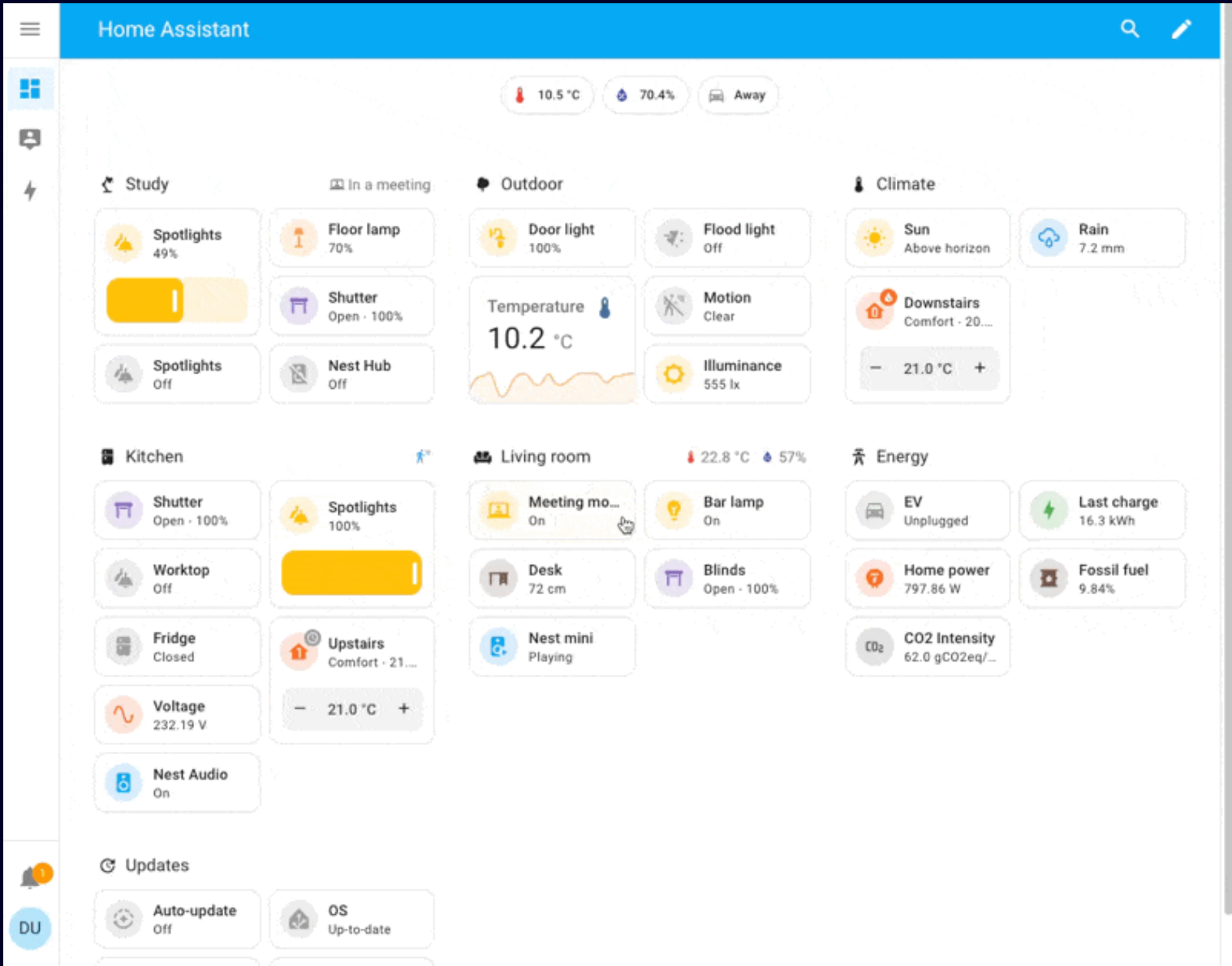
What are all these people doing?



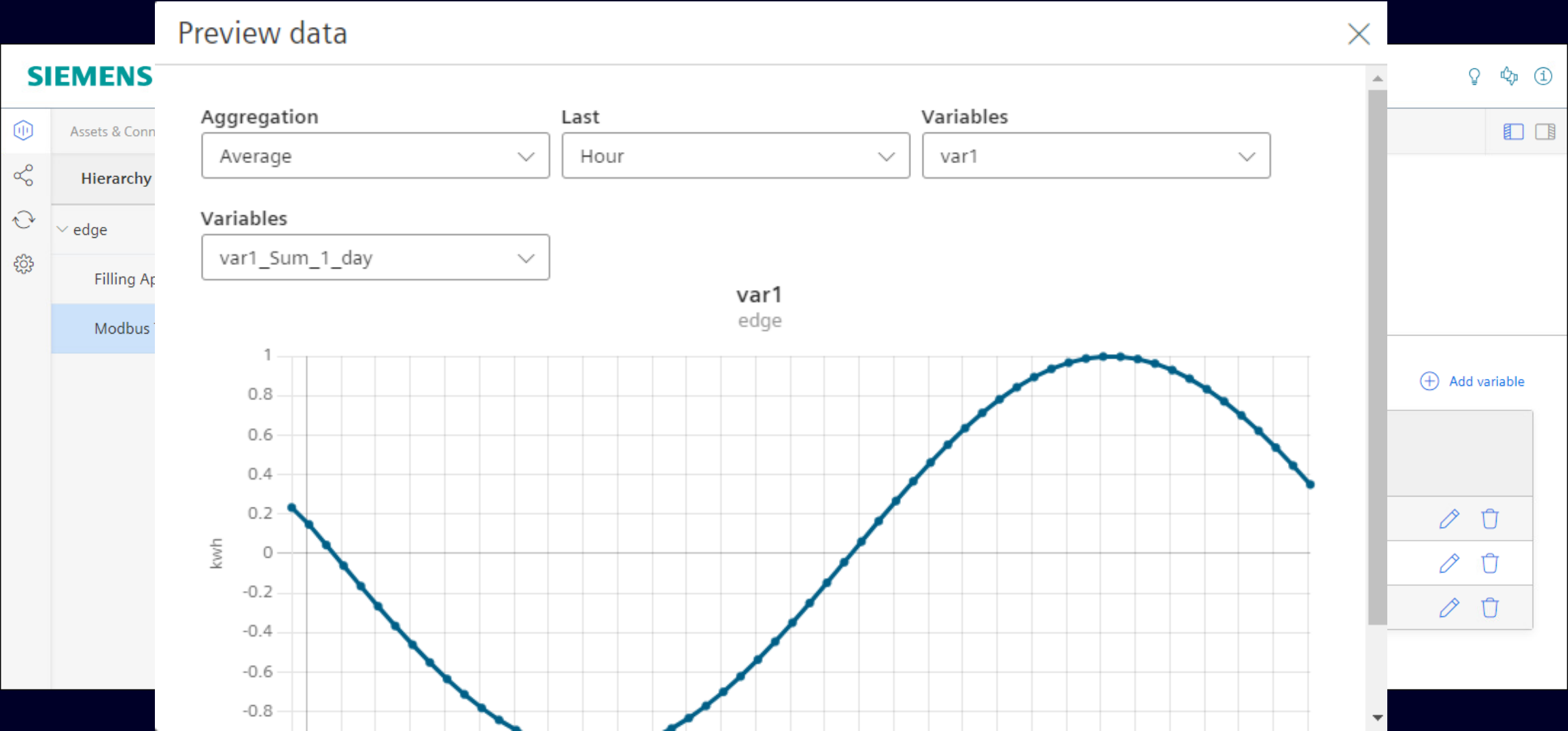
```
<tag>
  <tagDefinition>
    <unitId>0</unitId>
    <startIdx>11</startIdx>
    <readType>HOLDING_REGISTERS</readType>
    <dataType>INT_64</dataType>
  </tagDefinition>
  <tagName>myTag</tagName>
</tag>
```

The image shows the "Edit Modbus-Read node" configuration window in Node-RED. The window has a "Delete" button on the left and a "Cancel" button on the right. Below the buttons is a "Properties" section with a gear icon and a "Settings" tab. The "Settings" tab is active, showing several fields: "Name" (Read Holding Registers), "Topic" (holdingRegisters), "Unit-Id" (empty), "FC" (FC 3: Read Holding Registers), "Address" (0), and "Quantity" (4). Red arrows point to each of these fields. The "Optional" tab is also visible but not selected. The Node-RED logo is visible in the top right corner of the interface.

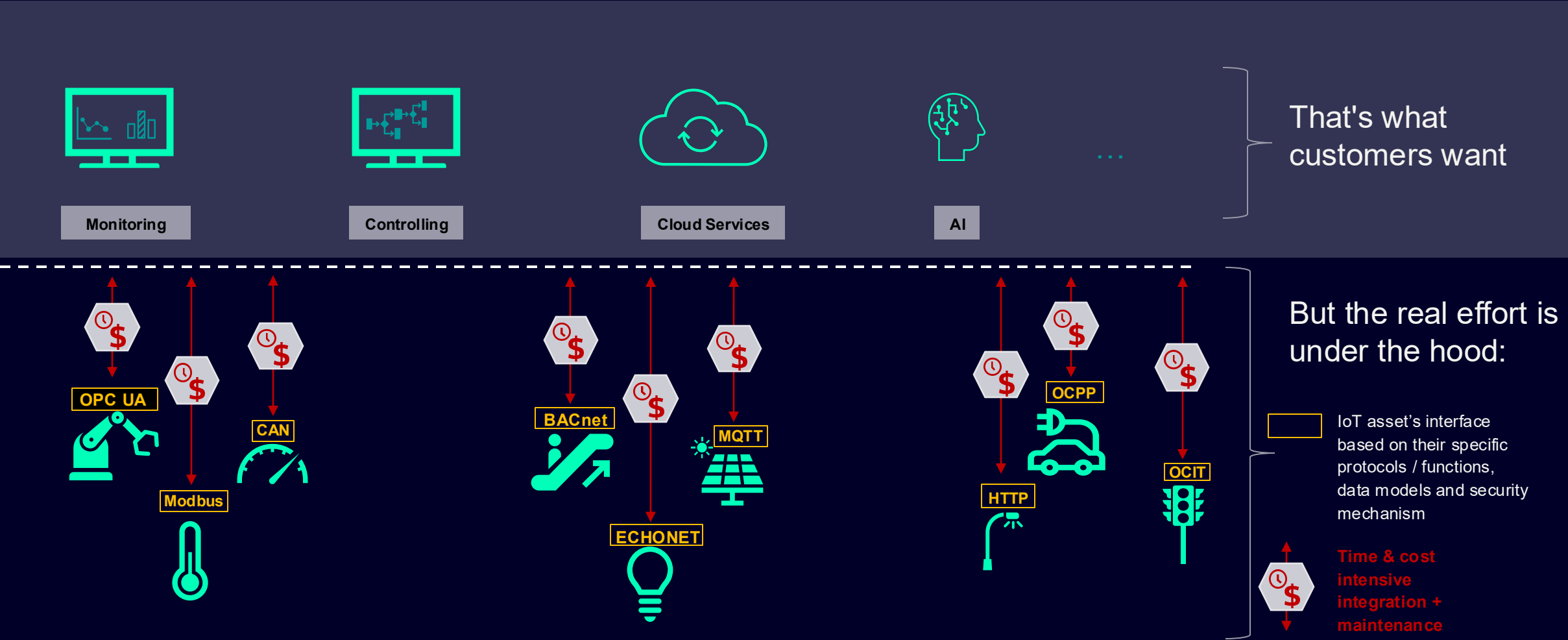
End Goal



End Goal



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



Quantifying the Problem

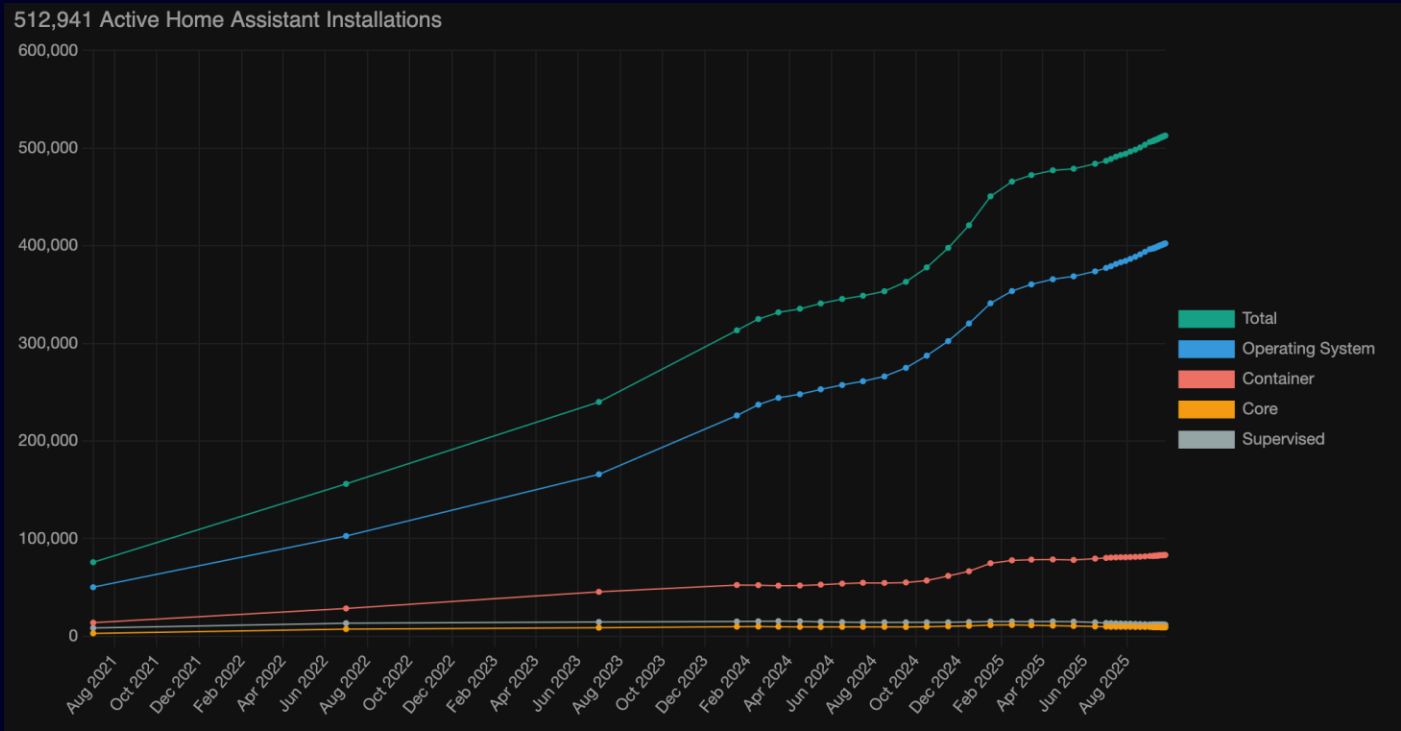


The Modbus integration was introduced in Home Assistant pre 0.7, and it's used by 2.6% of the active installations.

📖 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

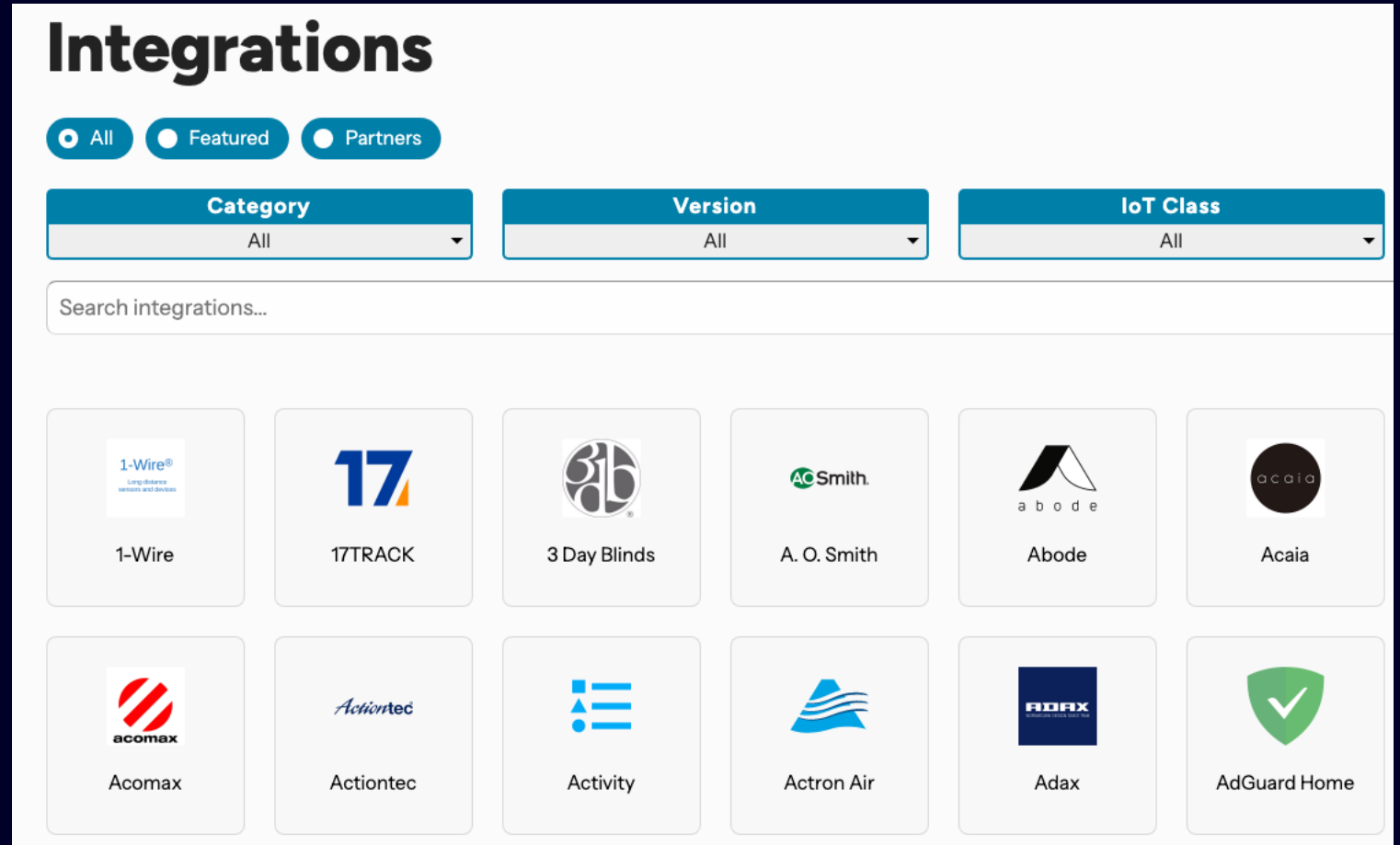
Why not build integrations for these platforms?

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/>

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

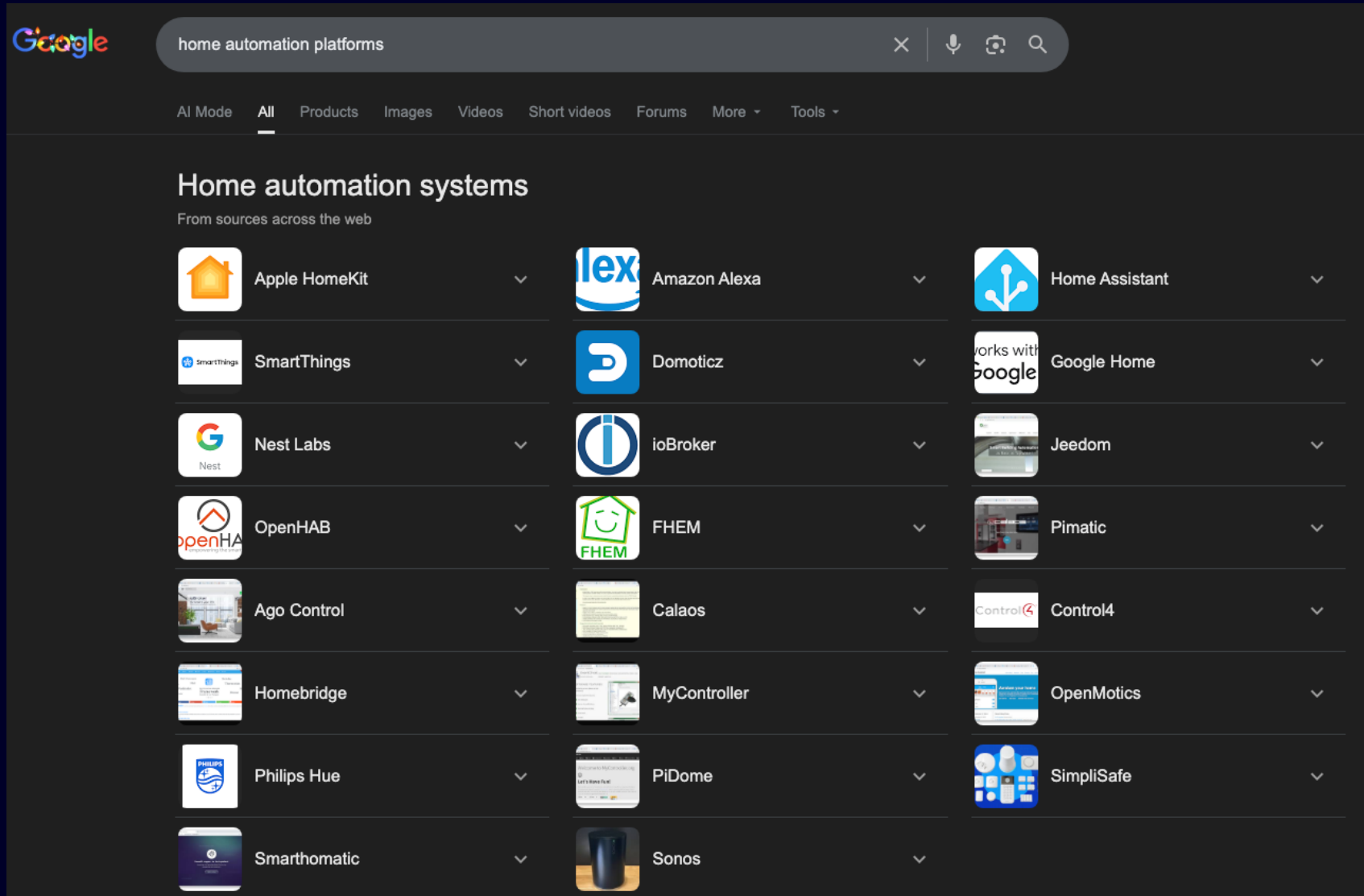


Each is a family with many subtypes



Servo drives

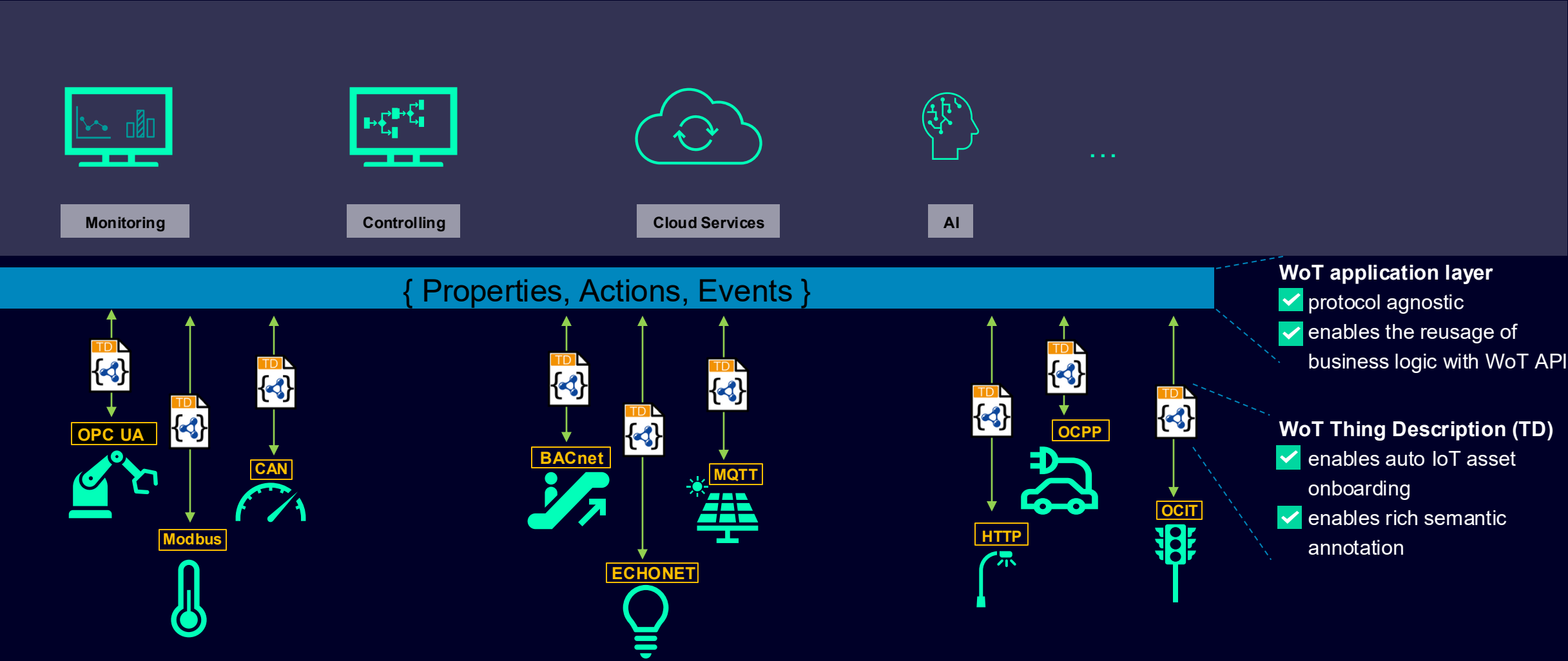
2. Which platform?



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

Web of Things to the rescue

Tackling the onboarding problem



Example TD for the Node-RED example from before

Edit Modbus-Read node

Delete

Cancel

Done

Properties

Settings

Optionals

Name

Read Holding Registers

Topic

holdingRegisters

Unit-Id

FC

FC 3: Read Holding Registers

Address

0

Quantity

4

Poll Rate

1

second(s)

Delay to activate input



TD Extract



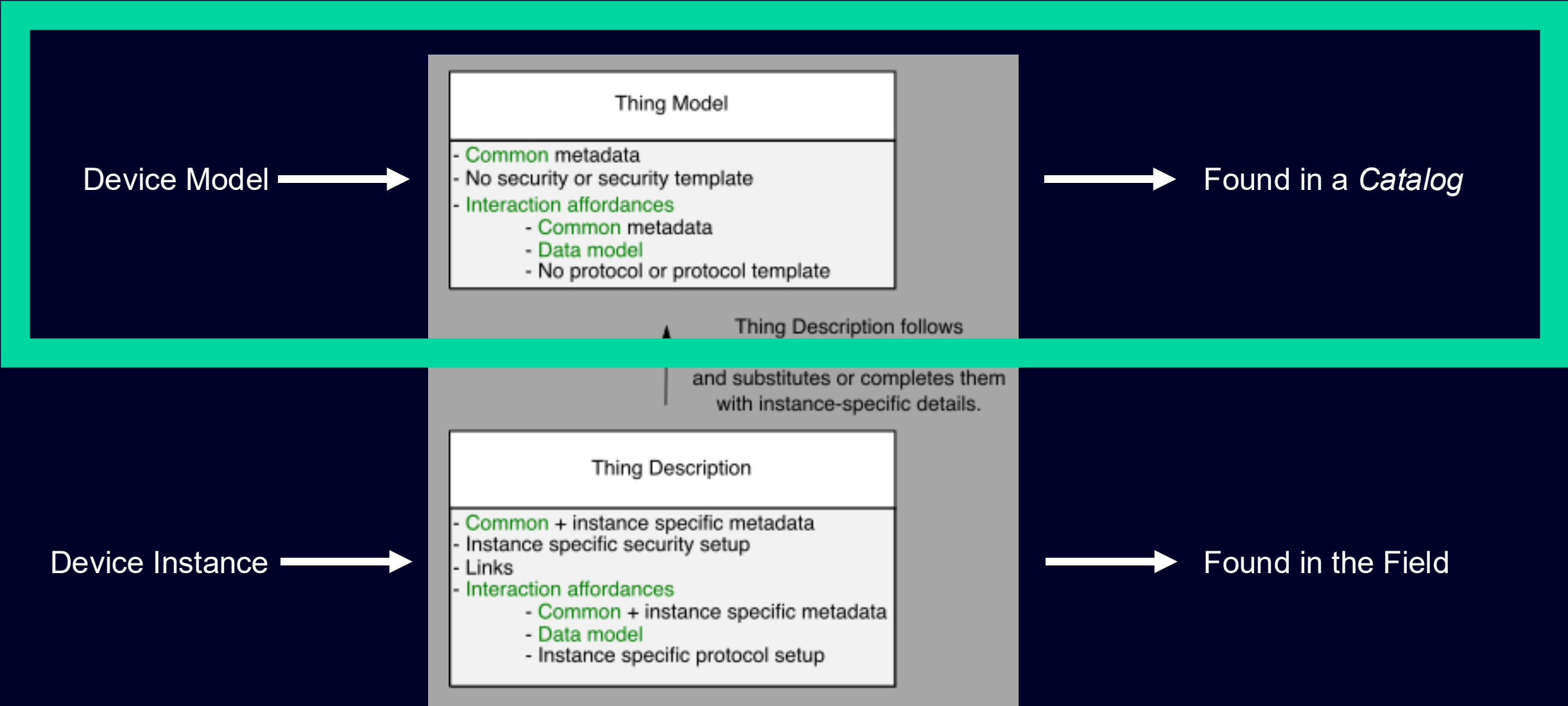
does not repeat

```
"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



Concentrating on the Thing Model



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](#)



Thing Model catalog

Tooling, home and services

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

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.

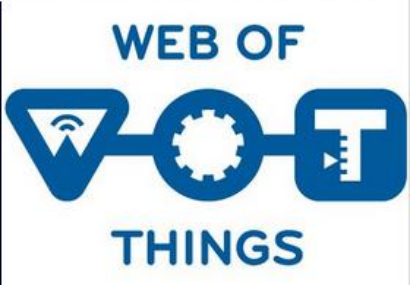

Explore the models on github:
<https://lnkd.in/eZxT7M-R>

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

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](#)



Thing Model catalog

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 internal index
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

Getting Started Documentation

Type to search

Thing Model Catalog - Documentation

Installation

Getting Started

Configure Autocompletion (Optio...

Browse the Example Catalog

Configure the Example Reposi...

List the Contents of the Exam...

List Versions

Fetch a Thing Model

Host Your Own Catalog

Concepts

Workflows

Commands

Architecture

Frequently Asked Questions

☰

A

🔄

🐦

f

🔗

Browse the Example Catalog

We provide an [example repository](#) for you to get acquainted with `tmc`. The following commands assume that you use the example repository. If your organization hosts a TM catalog for use as a default, you will need to change the commands accordingly.

Configure the Example Repository

```
tmc repo add -t http example https://raw.githubusercontent.com/wot-oss/example-catalog/main/thing-models/
```

List the Contents of the Example Repository

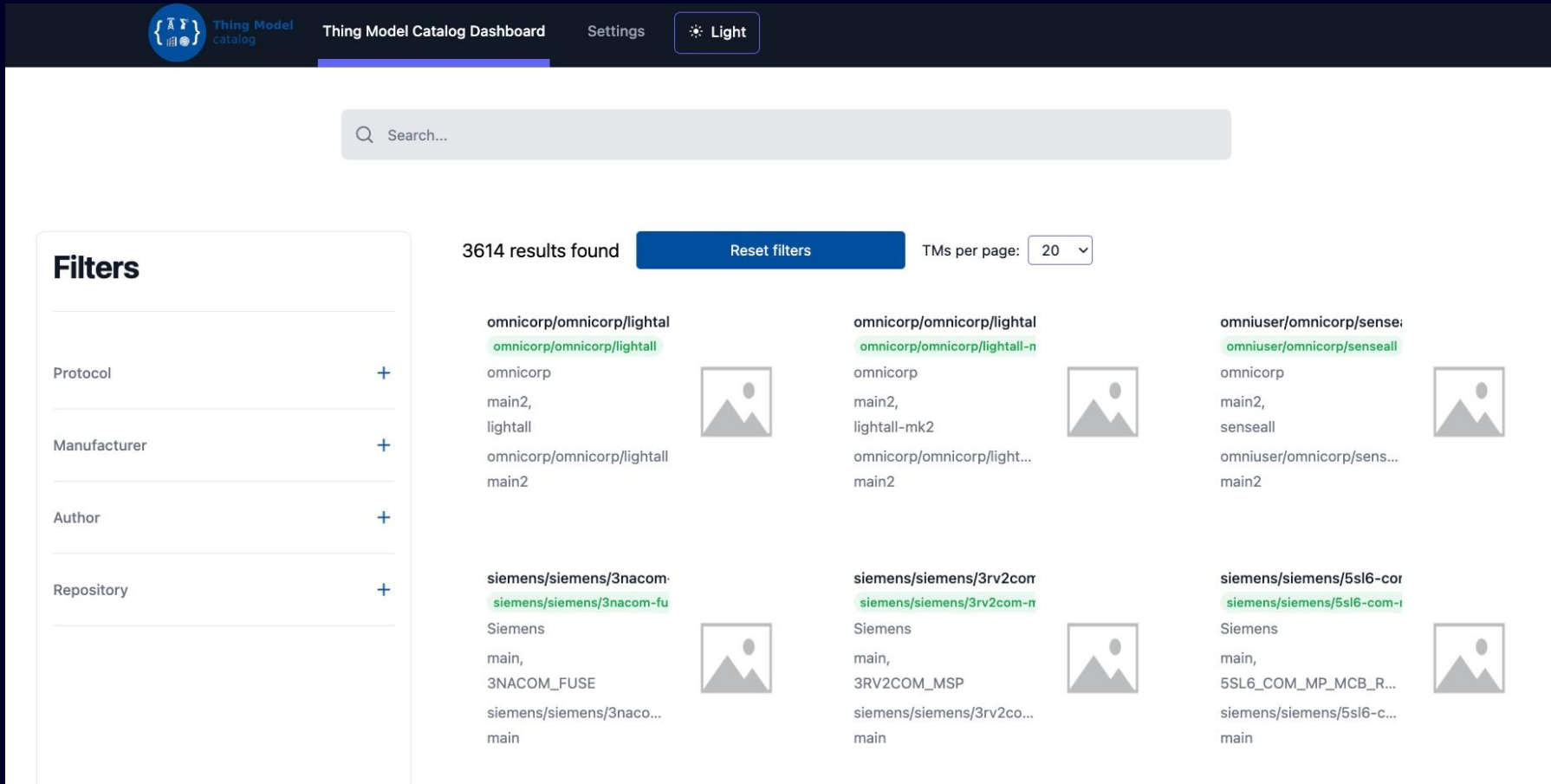
```
tmc list
```

The listed names are formatted as follows

```
<author>/<manufacturer>/<model>[/<optional-path>]
```

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

Web UI (Quite new, not perfect yet)



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

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!

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?

Authoring Tools: Easier Creation TMs

The screenshot displays the editDor IDE interface. On the left, a sidebar shows the 'HotelRoom PowerMeter' model with fields like `id`, `@context`, `securityDefinitions`, `security`, `@type`, `name`, and `base`. Below this is a 'Properties' section with a 'List' tab and a 'Table' tab. The 'Table' tab shows a table with columns for 'Unit ID' and 'Address Of'. The main editor area shows a JSON schema for a PowerMeter, with a large blue overlay in the center containing a white upload icon and the text 'Contribute to Catalog'. The top toolbar includes buttons for 'Send TD', 'Contribute to Catalog', 'Share', 'Open', 'Create', 'To TD', 'Download', and 'Settings'. The bottom status bar shows 'Properties: 9 | Actions: 0 | Events: 0 | Size: 3.363 KiB | Northbound State: ...' and a notification 'You have unsaved changes - v0.8.0'.

ediDor

JSON Validation ✓
JSON Schema Validation ✓

HotelRoom PowerMeter

id urn:PowerMeter0

@context >

securityDefinitions >

security >

@type >

name PowerMeter

base modbus://{(IP)}:{(PORT)}

> Forms

> Links

Properties List Table

Group Controls

Unit ID 1 + - Address Of

> device_id

> serial_number

> creation_date

> battery_failure

> voltage_constant

> light_bulb_power_in_ampere

> battery_charge_in_percent

> grid_voltage

> statistic_value

Contribute to Catalog

Properties: 9 | Actions: 0 | Events: 0 | Size: 3.363 KiB | Northbound State: ...

You have unsaved changes - v0.8.0

TMC Contributions further simplified with EdiTDor

Contribute your TM to a TM Catalog - Metadata

Follow the steps below to contribute your TM to a Catalog specified in the last step

1

2

3

The following fields will be added in the background to your TM for cataloging purposes to ensure quality and discoverability of Thing Models.

Model*

powermeter3000

Author*

EgeKorkan

Manufacturer*

ege inc.

License

URL of the license, e.g., https://www.apache.org/licenses/LICENSE-2.0.txt

Copyright Year

e.g. 2024...

Copyright Holder

Organization holding the copyright of the TM...

Validate

✓ TM is valid

Click to copy the full Thing Model

Close

Next

TMC Contributions further simplified with EdiTDor

Contribute your TM to a TM Catalog - Interaction

Follow the steps below to contribute your TM to a Catalog specified in the last step

1

2

3

If you want to verify the correctness of your Thing Model, you can interact with a device instance here. To do so, please configure the proxy (northbound, southbound, valuepath) and provide instance-specific information such as IP address.

> 2.1 Instance

> 2.2 Gateway

< 2.3 Value Verification

> 2.4 Saving results

Read property values from device instance

Property Name		Title	Preview Value
device_id	ⓘ	device_id	"7KG7750" ✓
serial_number	ⓘ	serial_number	"BF0703100052" ✓
creation_date	ⓘ	creation_date	"05082020" ✓
battery_failure	ⓘ	battery_failure	false ✓
voltage_constant	ⓘ	voltage_constant	false ✓
light_bulb_power_in_ampere	ⓘ	light_bulb_power_in_ampere	-0.20791169081775987 ✓
battery_charge_in_percent	ⓘ	battery_charge_in_percent	100 ✓
grid_voltage	ⓘ	grid_voltage	-0.20791169081775987 ✓
statistic_value	ⓘ	statistic_value	1.2534438669959154e-263 ✓

Test All Properties

Previous

Close

Next

TMC Contributions further simplified with EdiTDor

Contribute your TM to a TM Catalog - Submission

Follow the steps below to contribute your TM to a Catalog specified in the last step

1

2

3

Add the TM Catalog Endpoint and Repository URL

TM Catalog Endpoint

http://localhost:8082

Name of the Repository

example

Submit

✓ TM submitted successfully!

Copy TM id

egekorkan/ege-inc/powermeter3000/v0.0.0-20251204121622-f126a0719eeb.tm.json

Open in new tab

http://localhost:8082/thing-models/egekorkan/ege-inc/powermeter3000/v0.0.0-20251204121622-f126a0719eeb.tm.json

Previous

Close

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

Smart Edge Connector runs on an Edge Device

SIEMENS

Smart Edge Connector

?

S

«

Things

Data Planes

Things

All things provisioned to the Smart EDGE Connector.

Refresh

Import

+ Add

Search

<input type="checkbox"/>	Name ↑	Types	Protocol	Status	
<input type="checkbox"/>	POC1000	Thing	Modbus+TCP	Online	<div><div>👁</div><div>🗑</div></div>
<input type="checkbox"/>	Siemens SENTRON PAC4200	tm:ThingDescription	Modbus	Online	<div><div>👁</div><div>🗑</div></div>

We search through a Catalog

SIEMENS

Smart Edge Connector

?

S

Things

Data Planes

Cancel

Model Selection

Model Configuration

Next

Manufacturer

Series

Author

Manufacturer	Series	Author
Siemens	5SL6COM	nexus-x
Siemens	GH180	nexus-x
Siemens	P5320R	nexus-x
Siemens	PAC2200	nexus-x
Siemens	PAC4200	nexus-x
siemens	3NACOM-FUSE	siemens
siemens	5ST3COM-ASFC	siemens
siemens	5SV6COM-AFDD	siemens
siemens	POC1000	siemens
siemens	7KM2200-2EA30-1DA1	systemx

We instantiate a TD from TM

The screenshot displays the Siemens Smart Edge Connector web interface. On the left, a sidebar contains a menu with 'Things' and 'Data Planes'. The main area shows a configuration form for 'Model Selection'. At the top of the main area, there are navigation buttons: 'Cancel' (with a close icon), 'Back' (with a left arrow icon), and a progress indicator showing 'Model Selection' as the current step (with a checkmark) and 'Model Configuration' as the next step (with a circle). The form includes three input fields: 'IP*' with the value '192.168.4.136', 'PORT*' with the value '502', and 'ID*' with the value 'wotmeetup-poc1000'. A 'Finish' button is located below the ID field. The top of the interface features the Siemens logo and the text 'Smart Edge Connector', along with a help icon and a user icon labeled 'S'.

SIEMENS Smart Edge Connector

Things

Data Planes

Cancel Back Model Selection Model Configuration

IP*
192.168.4.136

PORT*
502

ID*
wotmeetup-poc1000

Finish

Device is onboarded!

SIEMENS

Smart Edge Connector

?

S

<<

D> > Devices > POC1000

POC1000

Export

Things

Data Planes

Details

Properties 122

Northbound 0

+ Properties

Preview

Search...

<input type="checkbox"/>	Name ↑	Title	Northbound	Value Preview
<input type="checkbox"/>	idenT_DEVICE_STATUS_UNIT_ID24	Device Status (24)		<div><div>!</div>0</div> Updated 28/10/2024, 14:30:32 <div><div></div><div></div></div>
<input type="checkbox"/>	idenT_DEVICE_STATUS_UNIT_ID3	Device Status (3)		<div><div>✓</div>3</div> Updated 28/10/2024, 14:30:32 <div><div></div><div></div></div>
<input type="checkbox"/>	idenT_DEVICE_STATUS_UNIT_ID4	Device Status (4)		<div><div>✓</div>3</div> Updated 28/10/2024, 14:30:32 <div><div></div><div></div></div>
<input type="checkbox"/>	idenT_DEVICE_STATUS_UNIT_ID5	Device Status (5)		<div><div>✓</div>3</div> Updated 28/10/2024, 14:30:32 <div><div></div><div></div></div>
<input type="checkbox"/>	idenT_DEVICE_STATUS_UNIT_ID6	Device Status (6)		<div><div>✓</div>3</div> Updated 28/10/2024, 14:30:32 <div><div></div><div></div></div>
<input type="checkbox"/>	idenT_DEVICE_STATUS_UNIT_ID7	Device Status (7)		<div><div>!</div>0</div> Updated 28/10/2024, 14:30:32 <div><div></div><div></div></div>
<input type="checkbox"/>	idenT_DEVICE_STATUS_UNIT_ID8	Device Status (8)		<div><div>!</div>0</div> Updated 28/10/2024, 14:30:32 <div><div></div><div></div></div>
<input type="checkbox"/>	idenT_DEVICE_STATUS_UNIT_ID9	Device Status (9)		<div><div>!</div>0</div> Updated 28/10/2024, 14:30:32 <div><div></div><div></div></div>
<input type="checkbox"/>	idenT_FW_COM	FW Version Communication Contr...		<div><div>✓</div>^Vu0002lu0000lu0002lu000...</div> Updated 28/10/2024, 14:30:32 <div><div></div><div></div></div>

CG Challenge

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



Contact

Ege Korkan

SI CTO SSA

ege.korkan@siemens.com