Western Norway
University of
Applied Sciences

Modeling for Smart Home Interoperability

Volker Stolz
Software Engineering research group

I-MDE-A Workshop 2023-05-16

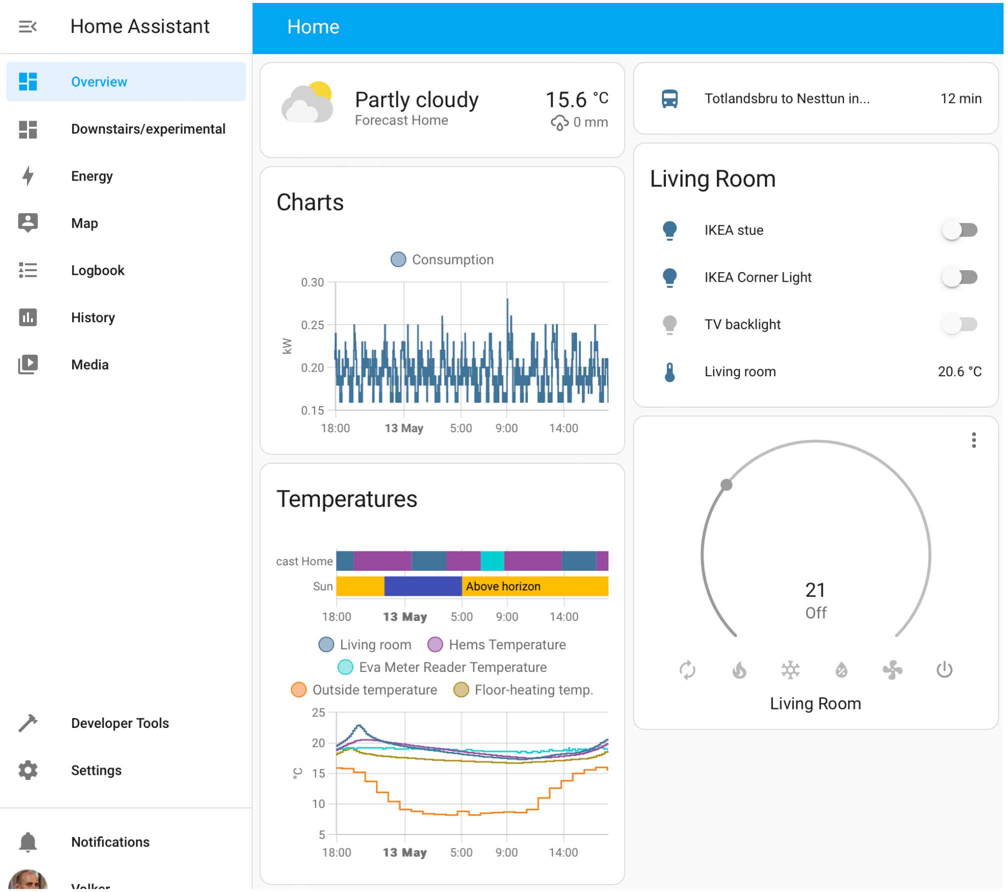
About Myself

- Prof. in Software Engineering @ HVL
  (adjunct position at Univ. of Oslo)

mastodon: @fm_volker

- MC member of COST Action CA20111 EuroProofNet
- Formal Methods
  - Temporal logics, runtime verification
  - Coloured Petri Nets CPN, TeSSLa
  - Modeling & Model Transformations (UML, declarative QVT)
- Programming Languages
  - concurrency, types, semantics, refactoring, coverage analysis

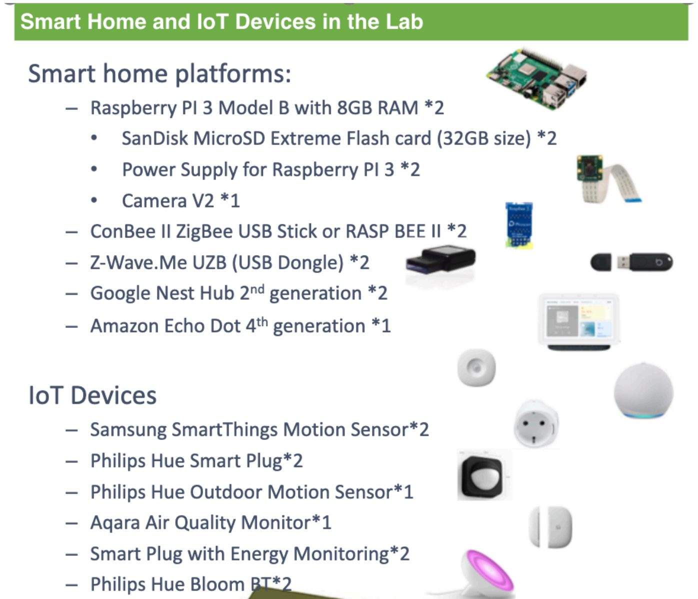
- The Home Assistant open source smart home solution
- Smart homes — nice to haves
- Home Assistant — state of the art
- Current solutions + “tooling”



“Smart Software Systems” IoT lab

- Budget for equipment: ca. 1500 EUR
- Raspberries
- Mostly Zigbee
  (2.4GHz based mesh protocol)
- sensors & actuators

- no real lab-space :-/
- physical scenarios difficult
  to play through



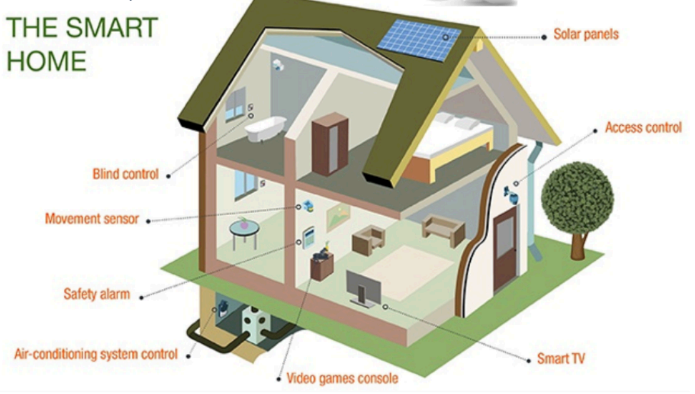
Smart Home and IoT Devices in the Lab

Smart home platforms:

- Raspberry PI 3 Model B with 8GB RAM *2
  - SanDisk MicroSD Extreme Flash card (32GB size) *2
  - Power Supply for Raspberry PI 3 *2
  - Camera V2 *1
- ConBee II ZigBee USB Stick or RASP BEE II *2
- Z-Wave.Me UZB (USB Dongle) *2
- Google Nest Hub 2nd generation *2
- Amazon Echo Dot 4th generation *1

IoT Devices

- Samsung SmartThings Motion Sensor*2
- Philips Hue Smart Plug*2
- Philips Hue Outdoor Motion Sensor*1
- Aqara Air Quality Monitor*1
- Smart Plug with Energy Monitoring*2
- Philips Hue Bloom BT*2



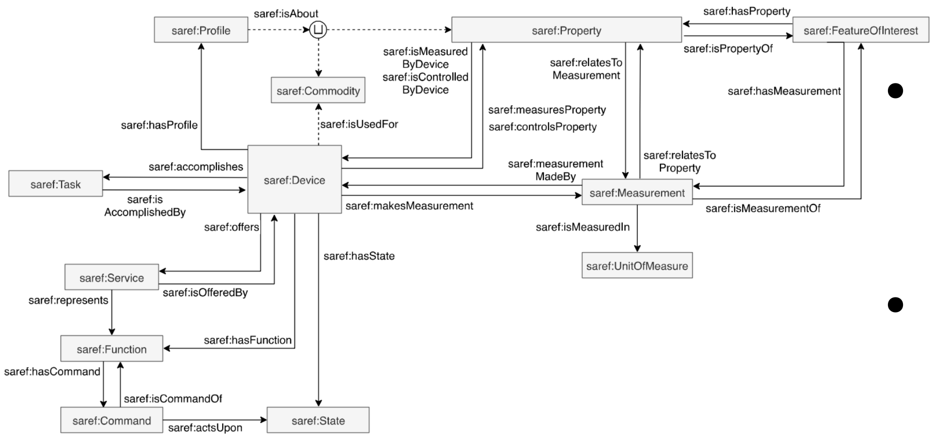
Research Questions

- Can smart homes be liberated from vendor-lock in?
- Improve standards & best practices?

- New features & services that we eventually want to achieve:
  - interoperability/migration
  - consistency checking
  - discoverability (here: blueprints/automations)

Smart Homes

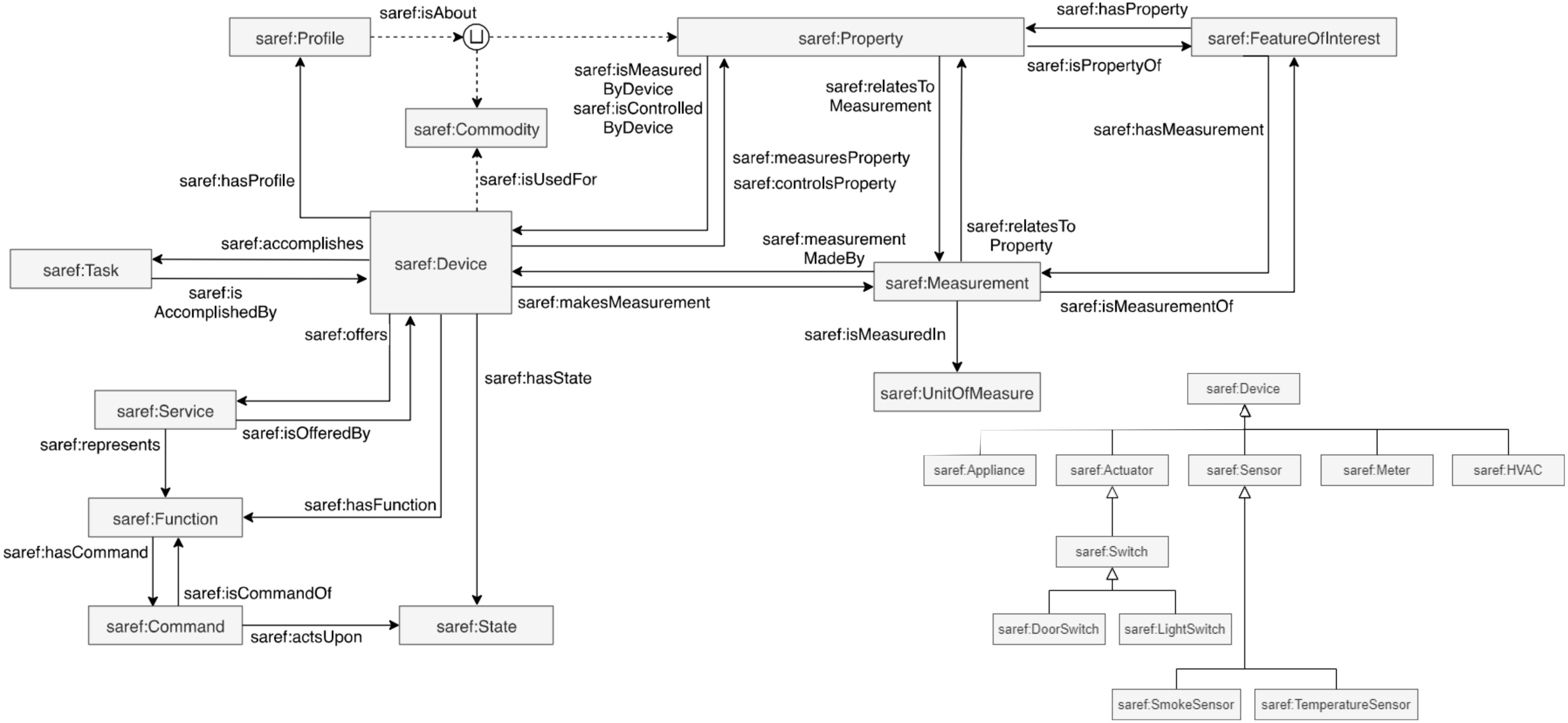
- Overview of structure
- Index of externally available tooling against YOUR smart home…
- …without having to implement it FOR your platform.
- ETSI standard: SAREF (“Smart Applications REFerence Ontology”)



- Alternative: Brick Schema (https://brickschema.org/)

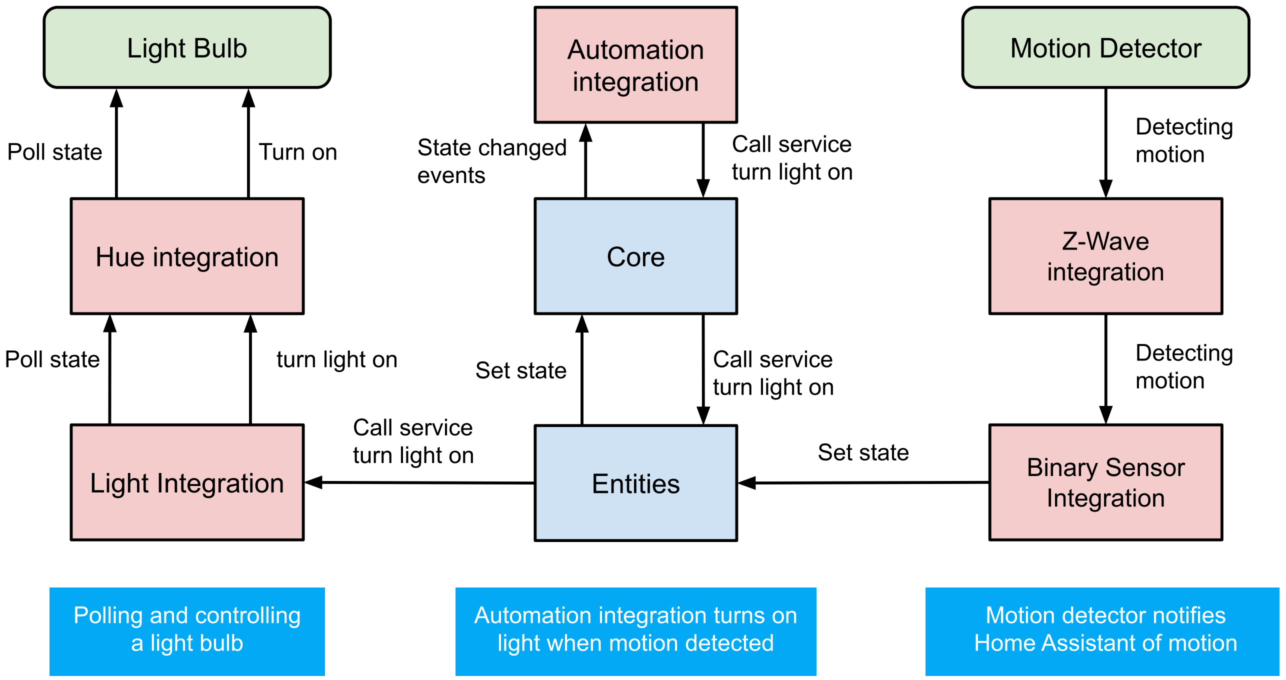
- Configuration Flexibility
  - Home Assistant: web-based (to check: REST)

SAREF Overview



Home Assistant

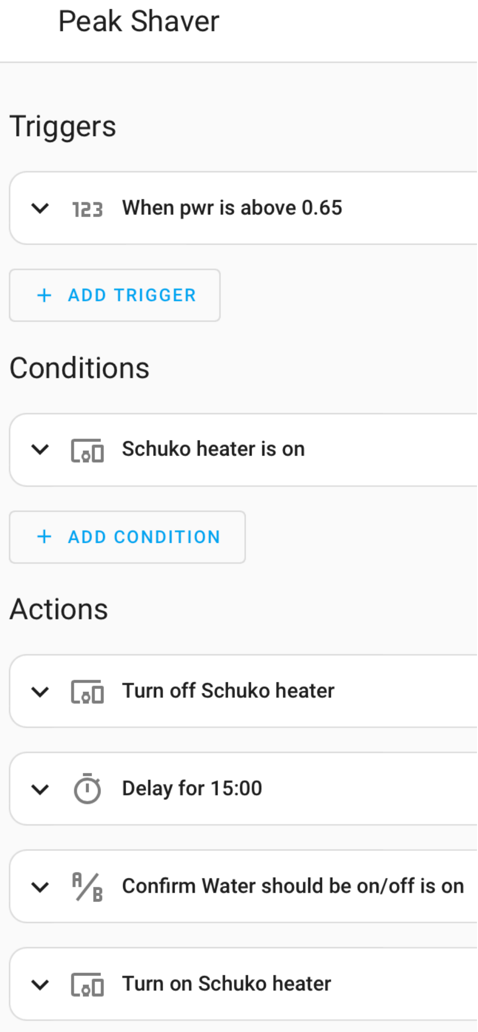
- Devices
- Entities
- UI/Frontend/API
- Automations



Integration Architecture example (developers.home-assistant.io)

Home Assistant - Implementation

- Python (object-oriented, annotations)
  - JSON via REST-API
  - SQL (SQLite by default) for data
  - Frontend in Typescript
  - Plugin mechanism (1000+ within the Home Assistant core repo alone!)
- “Blueprints”: trigger/condition/action templates for automations
  with place-holders for devices



Peak Shaver

Triggers

When pwr is above 0.65

Conditions

Schuko heater is on

Actions

Turn off Schuko heater

Delay for 15:00

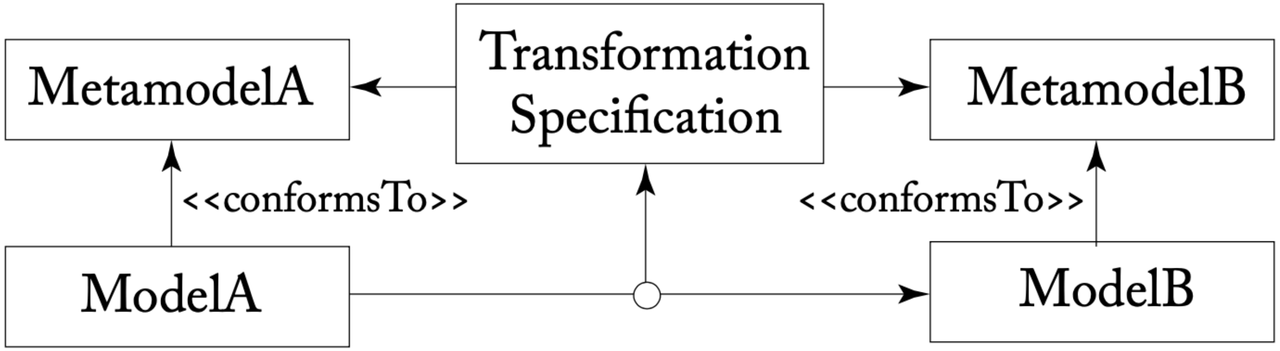
Confirm Water should be on/off is on

Turn on Schuko heater

Model Extraction: Current Approach (1)

- Goals:
  - static metamodel(s)/schema for Home Assistant
  - model = instance configuration
- Modelling framework: UML, RDF, …

- Textbook solution:



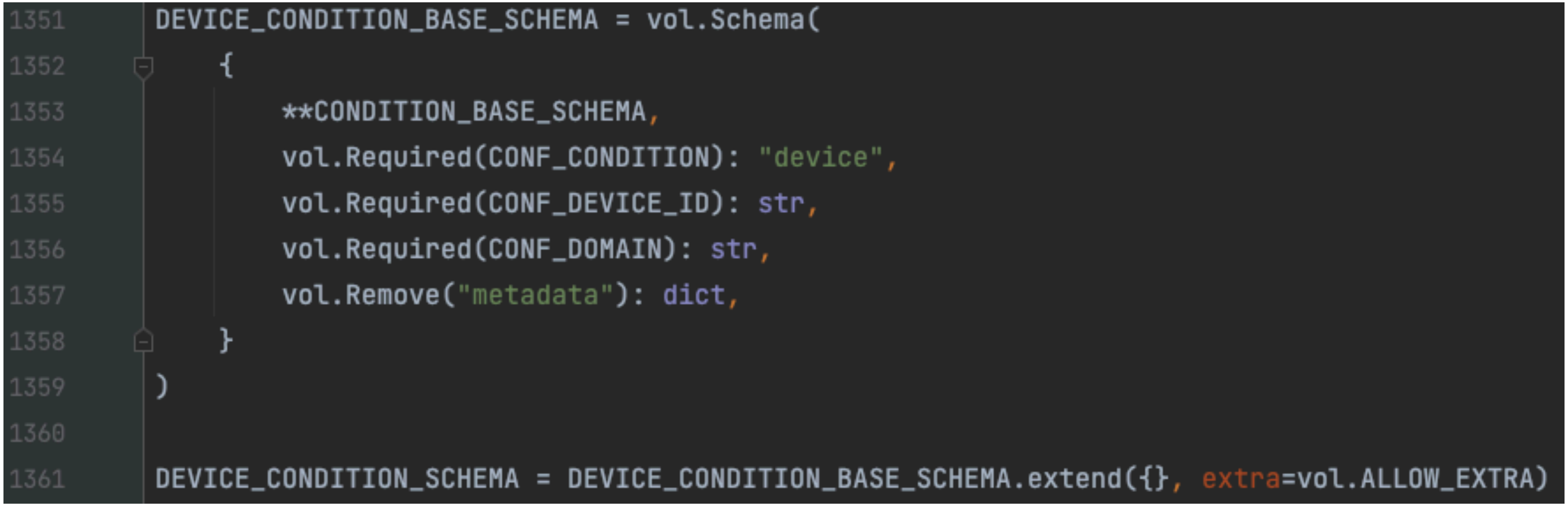
Textbook solution for M2M transformation
[Model Driven Software Engineering in Practice, 2nd ed., 2017]

Model Extraction: Current Approach (2)

- Goals: static metamodel(s)/schema, model = instance configuration
- Static configuration: *should* be from code (it’s static, after all…)
- Instance: extract via REST
  - Reasons:
    i) provide web-service via tokens for central data collection
    ii) no need for potential users to install anything apart from generating a token (and revoking it again afterwards)
- “Blueprints”: scraped from forums/git
- Knowledge represented in RDF/OWL, queries via SPARQL

The Problem: Python…

- Model-extraction from Python?
  not: “of Python”, but of structures modeled in Python…
- 1st challenge: where/which schemas? (Currently: manually identified.)

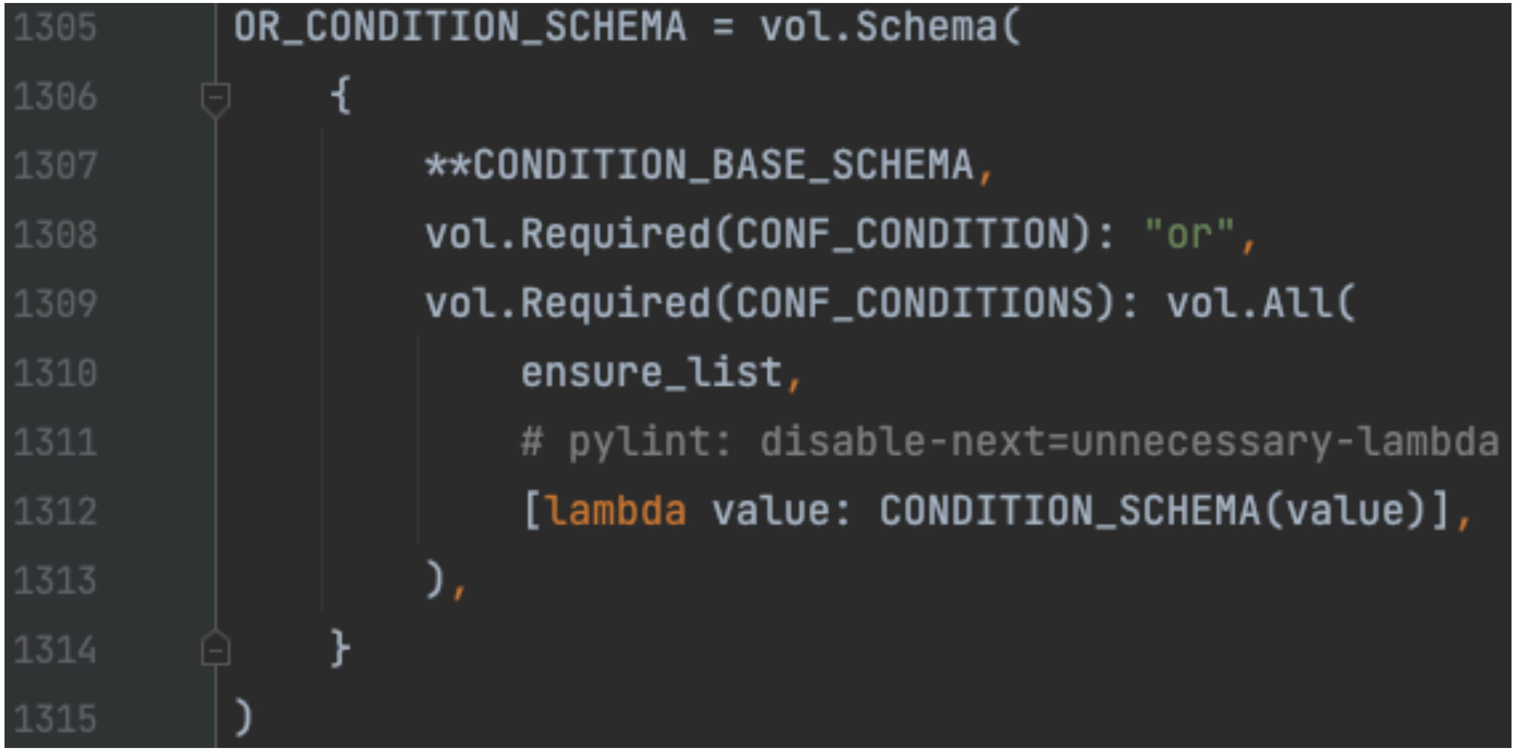


```
DEVICE_CONDITION_BASE_SCHEMA = vol.Schema(
    {
        **CONDITION_BASE_SCHEMA,
        vol.Required(CONF_CONDITION): "device",
        vol.Required(CONF_DEVICE_ID): str,
        vol.Required(CONF_DOMAIN): str,
        vol.Remove("metadata"): dict,
    }
)

DEVICE_CONDITION_SCHEMA = DEVICE_CONDITION_BASE_SCHEMA.extend({}, extra=vol.ALLOW_EXTRA)
```

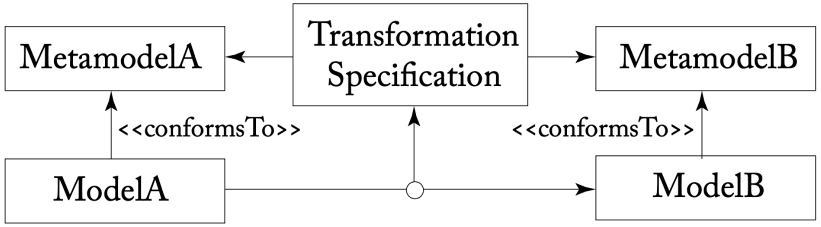
The Problem: Python…

- Model-extraction from Python?
  not: “of Python”, but of structures modeled in Python…
- 2nd challenge: good luck looking into this λ…



```
OR_CONDITION_SCHEMA = vol.Schema(
    {
        **CONDITION_BASE_SCHEMA,
        vol.Required(CONF_CONDITION): "or",
        vol.Required(CONF_CONDITIONS): vol.All(
            ensure_list,
            # pylint: disable-next=unnecessary-lambda
            [lambda value: CONDITION_SCHEMA(value)],
        ),
    }
)
```

Model Extraction



MetamodelA

Transformation Specification

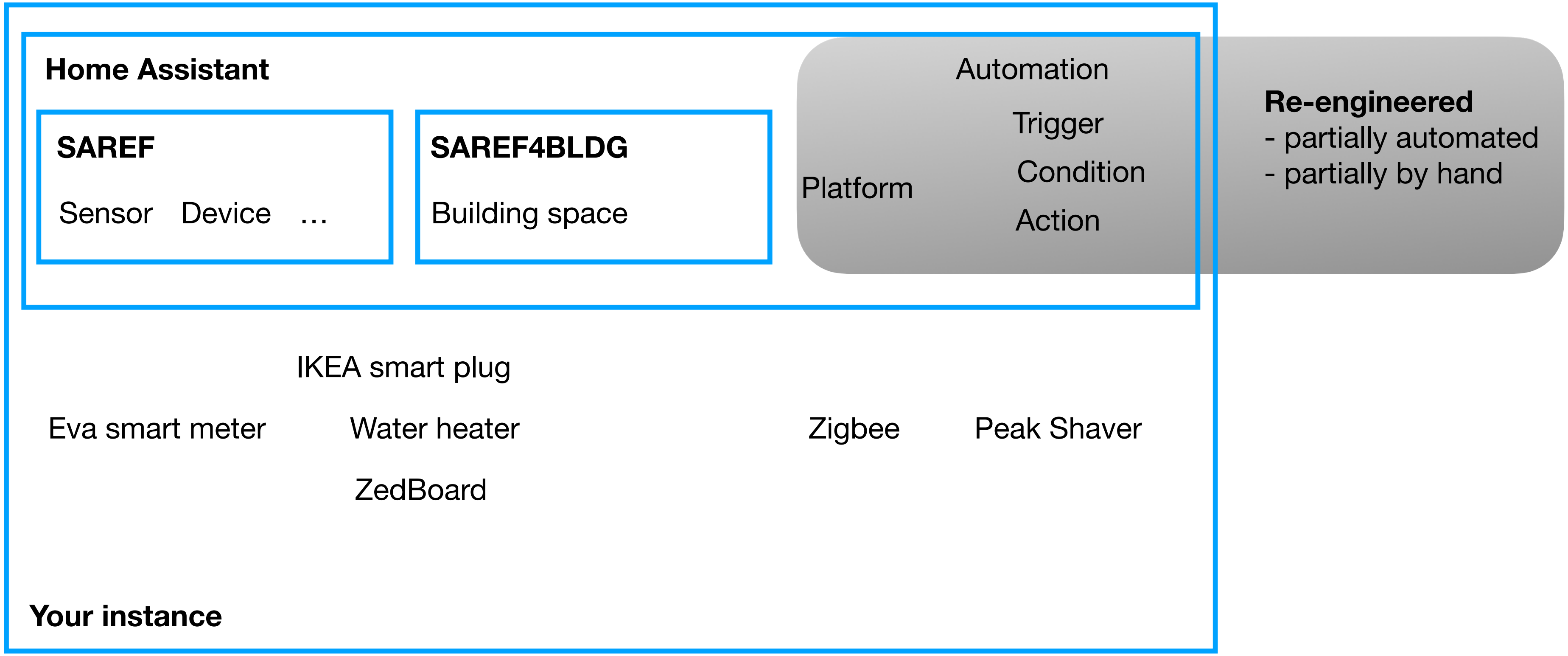
MetamodelB

<<conformsTo>>

<<conformsTo>>

ModelA

ModelB



Home Assistant

SAREF

Sensor Device …

SAREF4BLDG

Building space

Automation

Trigger

Condition

Action

Platform

Re-engineered
- partially automated
- partially by hand

IKEA smart plug

Eva smart meter

Water heater

ZedBoard

Zigbee

Peak Shaver

Your instance

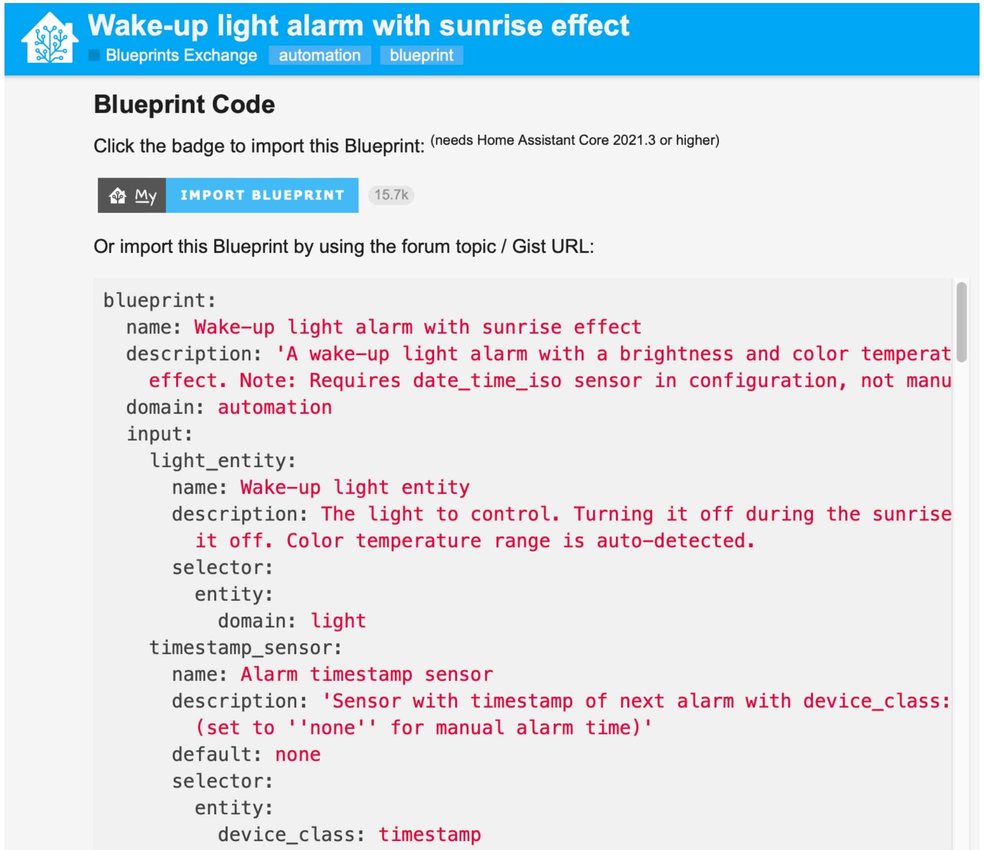
Application: Discovery

- Scrape Blueprints
  (forums, Github, Discord)

- Which blueprints does my config support/which devices should I still get?

- Apply NLP to blueprint description

- Interchange? GAFA / IFTTT



Wake-up light alarm with sunrise effect

Blueprint Code

Click the badge to import this Blueprint: (needs Home Assistant Core 2021.3 or higher)

Or import this Blueprint by using the forum topic / Gist URL:

```
blueprint:
  name: Wake-up light alarm with sunrise effect
  description: 'A wake-up light alarm with a brightness and color temperat
    effect. Note: Requires date_time_iso sensor in configuration, not manu
  domain: automation
  input:
    light_entity:
      name: Wake-up light entity
      description: The light to control. Turning it off during the sunrise
        it off. Color temperature range is auto-detected.
      selector:
        entity:
          domain: light
    timestamp_sensor:
      name: Alarm timestamp sensor
      description: 'Sensor with timestamp of next alarm with device_class:
        (set to ''none'' for manual alarm time)'
      default: none
      selector:
        entity:
          device_class: timestamp
```

Applications: Consistency Checking

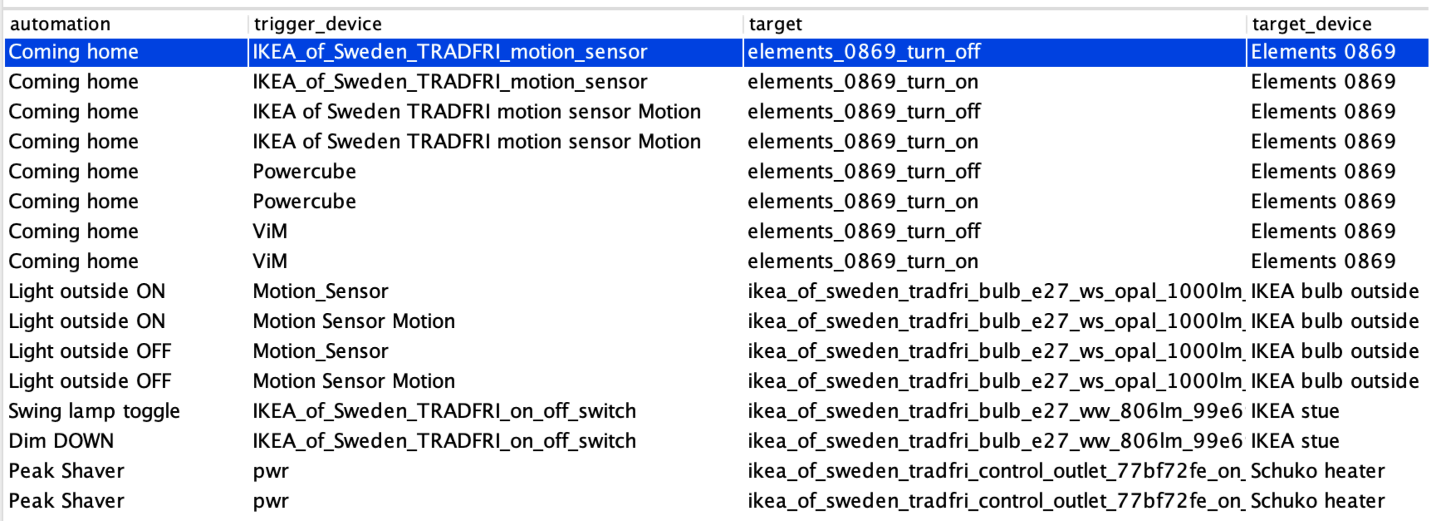
- Is this device controlled by…
  - …multiple automations?
  - …multiple devices?
- Is device X in location A controlled by device Y in location B?
  (needs domain-knowledge: may or may not be meaningful)
- Supplement knowledge (device X consumes 2 KW/h when on)
  (currently needs to be configured within platform)
- future work: check *which action* automation actually does
  (“algebra” for switching on/off devices, adjusting values, …)

Querying the Model: SPARQL

- Semantic web technology standard,
  semantic query language for databases

- Example: *“which device is controlled by which automation?”*

```
SELECT ?automation ?trigger_device ?target ?target_device
   WHERE {
      ?automation rdf:type ha:Automa
      ?automation ha:consistsOf ?acti
      ?automation ha:hasTrigger ?trigg
      { ?trigger ha:device ?trigger_dev
      ?action ha:target ?target .
      ?target_device saref:offers ?targ
   } ORDER BY ?target_device
```



automation	trigger_device	target	target_device
Coming home	IKEA_of_Sweden_TRADFRI_motion_sensor	elements_0869_turn_off	Elements 0869
Coming home	IKEA_of_Sweden_TRADFRI_motion_sensor	elements_0869_turn_on	Elements 0869
Coming home	IKEA of Sweden TRADFRI motion sensor Motion	elements_0869_turn_off	Elements 0869
Coming home	IKEA of Sweden TRADFRI motion sensor Motion	elements_0869_turn_on	Elements 0869
Coming home	Powercube	elements_0869_turn_off	Elements 0869
Coming home	Powercube	elements_0869_turn_on	Elements 0869
Coming home	ViM	elements_0869_turn_off	Elements 0869
Coming home	ViM	elements_0869_turn_on	Elements 0869
Light outside ON	Motion_Sensor	ikea_of_sweden_tradfri_bulb_e27_ws_opal_1000lm_	IKEA bulb outside
Light outside ON	Motion Sensor Motion	ikea_of_sweden_tradfri_bulb_e27_ws_opal_1000lm_	IKEA bulb outside
Light outside OFF	Motion_Sensor	ikea_of_sweden_tradfri_bulb_e27_ws_opal_1000lm_	IKEA bulb outside
Light outside OFF	Motion Sensor Motion	ikea_of_sweden_tradfri_bulb_e27_ws_opal_1000lm_	IKEA bulb outside
Swing lamp toggle	IKEA_of_Sweden_TRADFRI_on_off_switch	ikea_of_sweden_tradfri_bulb_e27_ww_806lm_99e6	IKEA stue
Dim DOWN	IKEA_of_Sweden_TRADFRI_on_off_switch	ikea_of_sweden_tradfri_bulb_e27_ww_806lm_99e6	IKEA stue
Peak Shaver	pwr	ikea_of_sweden_tradfri_control_outlet_77bf72fe_on_	Schuko heater
Peak Shaver	pwr	ikea_of_sweden_tradfri_control_outlet_77bf72fe_on_	Schuko heater

Demo

Technical Challenges

- Home Assistant entities need mapping into SAREF
- Only looked at Home Assistant so far.
  What about OpenHAB, Google Nest, Amazon Alexa, Apple Home?
  Cross-platform / no-code IFTTT
- No machine-readable meta-model for Home Assistant:
  - ad-hoc plugins
  - some object-orientation
  - schemata partially inspectable at runtime; not: statically!
- No model transformations yet (ad-hoc Python from JSON)
  RML.io: Java

Current Results

- Static core Home Assistant meta-model (semi-manual, incomplete) https://github.com/VolkerStolz/HASS-to-OWL-exporter
- Export of devices + entities + automations
    - Privacy filter!
- Import of blueprints
- Future:
    - Data collection on web-server (crowd-sourcing configurations)
    - Automated scraping of blueprints URL/corpus
    - Analysis of natural language blueprint description

Other good ideas?

Starting point for other Bachelor/MSc projects applying formal methods to IoT

Integrating with TeSSLa, STL

Thank You!