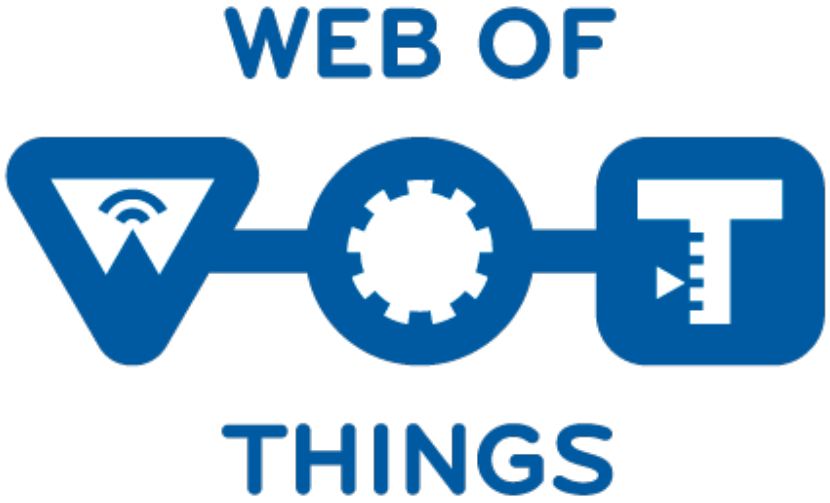


W3C Web of Things Retail Demo Scenario

Michael McCool

3 May 2022

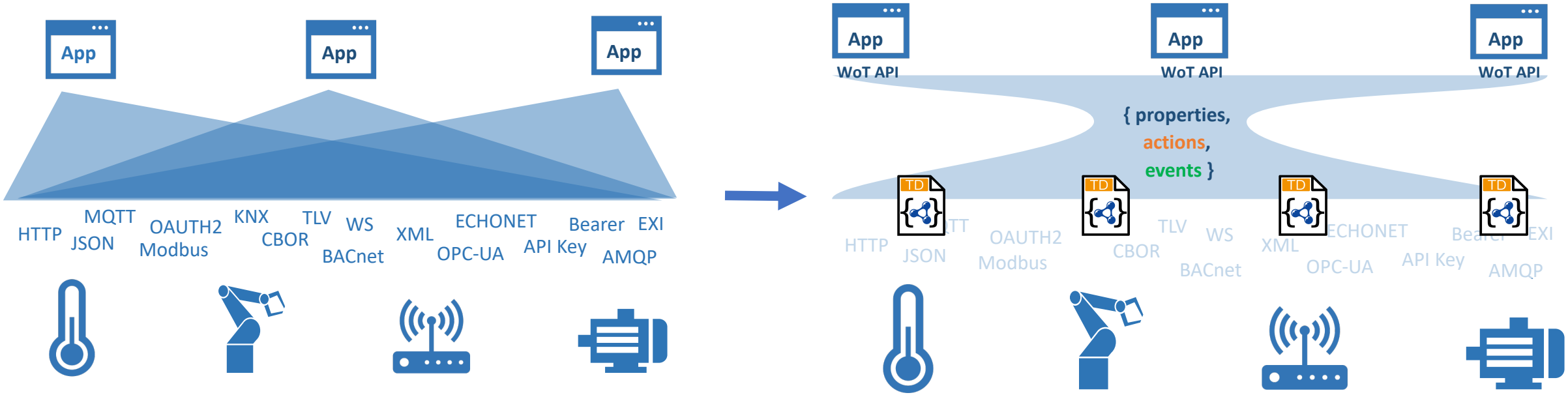
Conexxus Conference

Outline

- What is WoT?
  - Applying and extending web standards for IoT
  - Descriptive interoperability
- Test Scenario: Cooler Monitor
  - Hub architecture
  - Desired behavior and devices
- Implementations
  - Options
  - Home Assistant and Node-RED Prototypes
- Discussion
  - Semantic interoperability

W3C Web of Things (WoT)

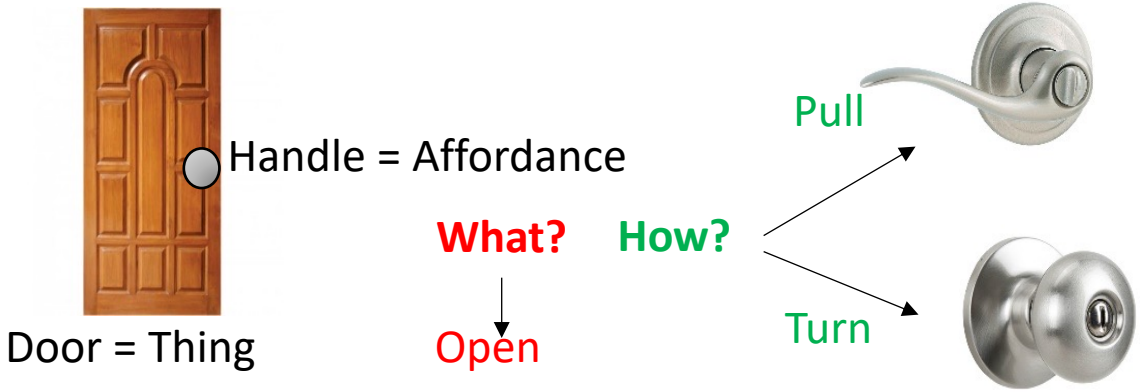
- W3C Working Group goal: Adapting web technologies to IoT
- Already published: Thing Description (TD) metadata format
  - TD describes the available interactions (network API) of a Thing
- New deliverables in progress, including Discovery
  - How does a potential user obtain the TD for a Thing?



Descriptive Interoperability: TDs

WoT Architecture

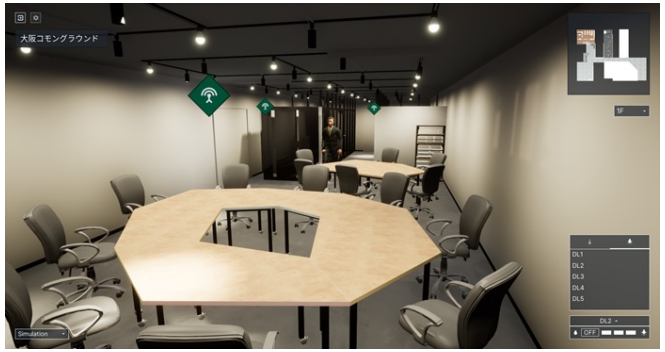
- Constraints
  - "Things" must have a TD
  - Must use URIs, IANA media types, etc.
- Thing Description Affordances
  - Describes WHAT the possible choices are
  - Describes HOW to interact with the Thing



WoT Thing Description (TD)

```
{
  "@context": [
    "https://www.w3.org/2022/wot/td/v1.1",
    { "iot": "http://iotschema.org/" }
  ],
  "id": "urn:dev:org:32473:1234567890",
  "title": "MyLEDThing",
  "description": "RGB LED torchiere",
  "@type": ["Thing", "iot:Light"],
  "securityDefinitions": {
    "default": {"scheme": "bearer"}
  },
  "security": ["default"],
  "properties": {
    "brightness": {
      "@type": ["iot:Brightness"],
      "type": "integer",
      "minimum": 0,
      "maximum": 100,
      "forms": [ ... ]
    }
  },
  "actions": {
    "fadeIn": {
      ...
```

Applications



https://www.takenaka.co.jp/news/2021/05/02/

Takenaka Corporation

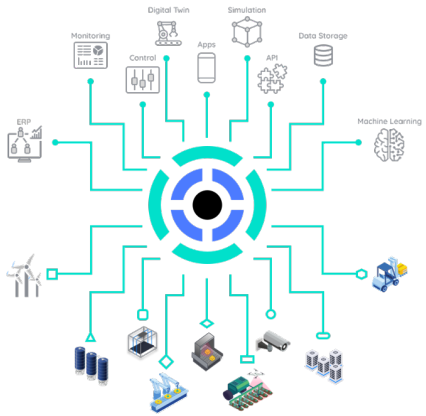
- CGLL Platform - BIM

https://new.siemens.com/global/en/products/buildings/automation/desigo.html

https://www.evosoft.com/en/digitalization-offering/saywot/

Siemens

- Desigo CC – BIM
- Say WoT!



https://netzo.io/

Netzo

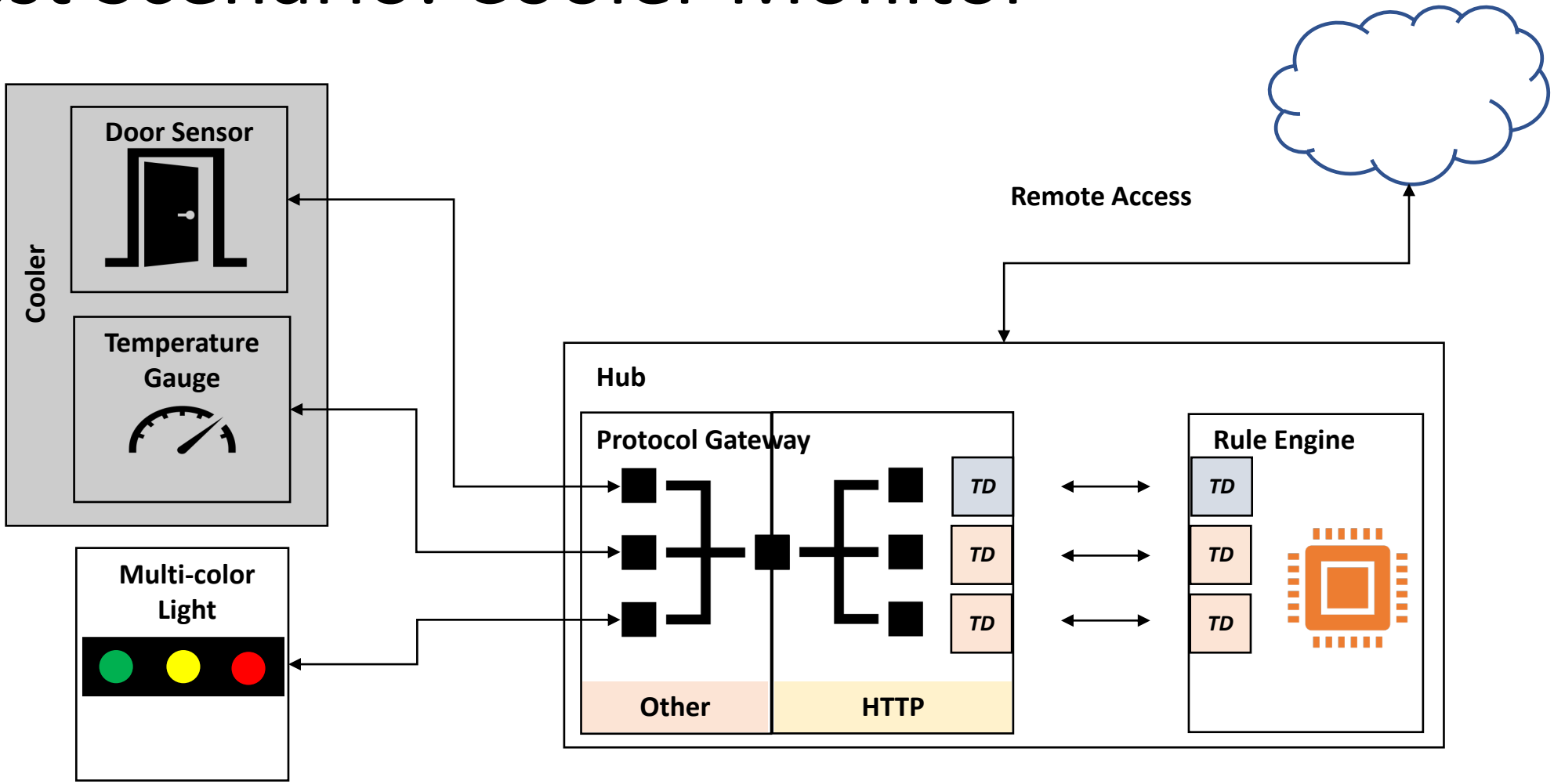
- IoT Data Hub
- Dashboards

https://www.eclipse.org/ditto/2022-03-03-wot-integration.html

Bosch

- Eclipse Ditto - Digital twin

Test Scenario: Cooler Monitor



Why a Hub?

- Protocol translation
- Data translation
- Other services
  - Historical data
  - Dashboards
- Rules engine (automation)
- Shadowing
  - Caching state of sleeping devices
  - Queuing commands to sleeping devices
  - Important for battery-powered devices
- Secure Remote Access

Desired Behavior

Door Open Alert

- If the cooler door has been left open for more than a certain time: **flash light.**
- If the door is closed: **cancel flashing**

Temperature Out of Range

- If internal freezer temp is too low: **turn light BLUE**
- If internal freezer temp is too high: **turn light RED**
- If internal freezer temp is normal: **turn light WHITE**

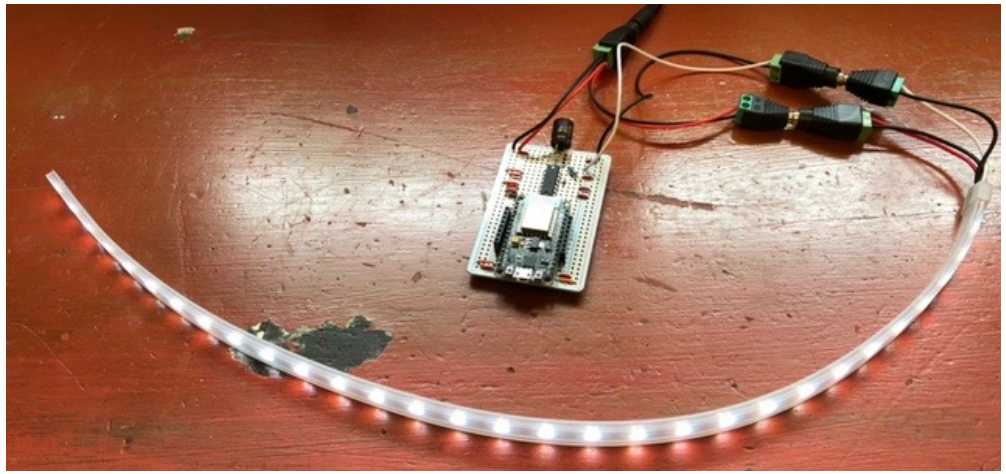
NOTE:

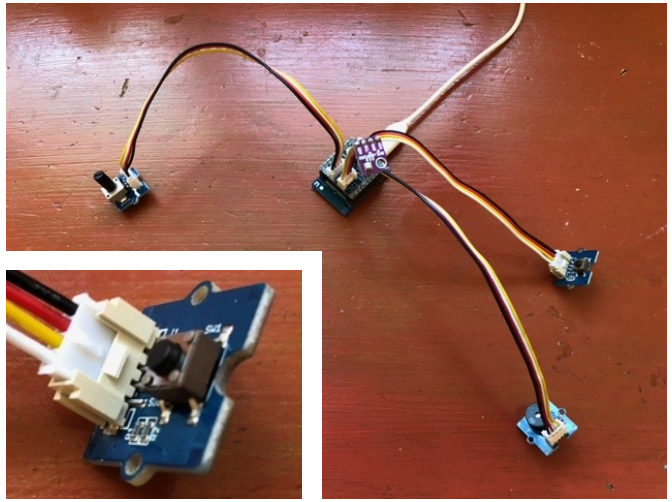
- Rules should co-exist.

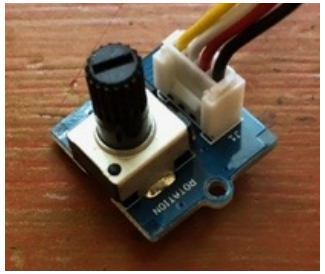
→ If door is open *and* temps are high, light should be **RED** *and* flashing

Devices

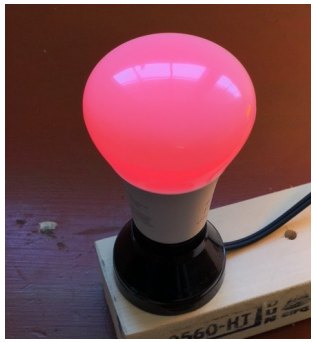
Test Devices



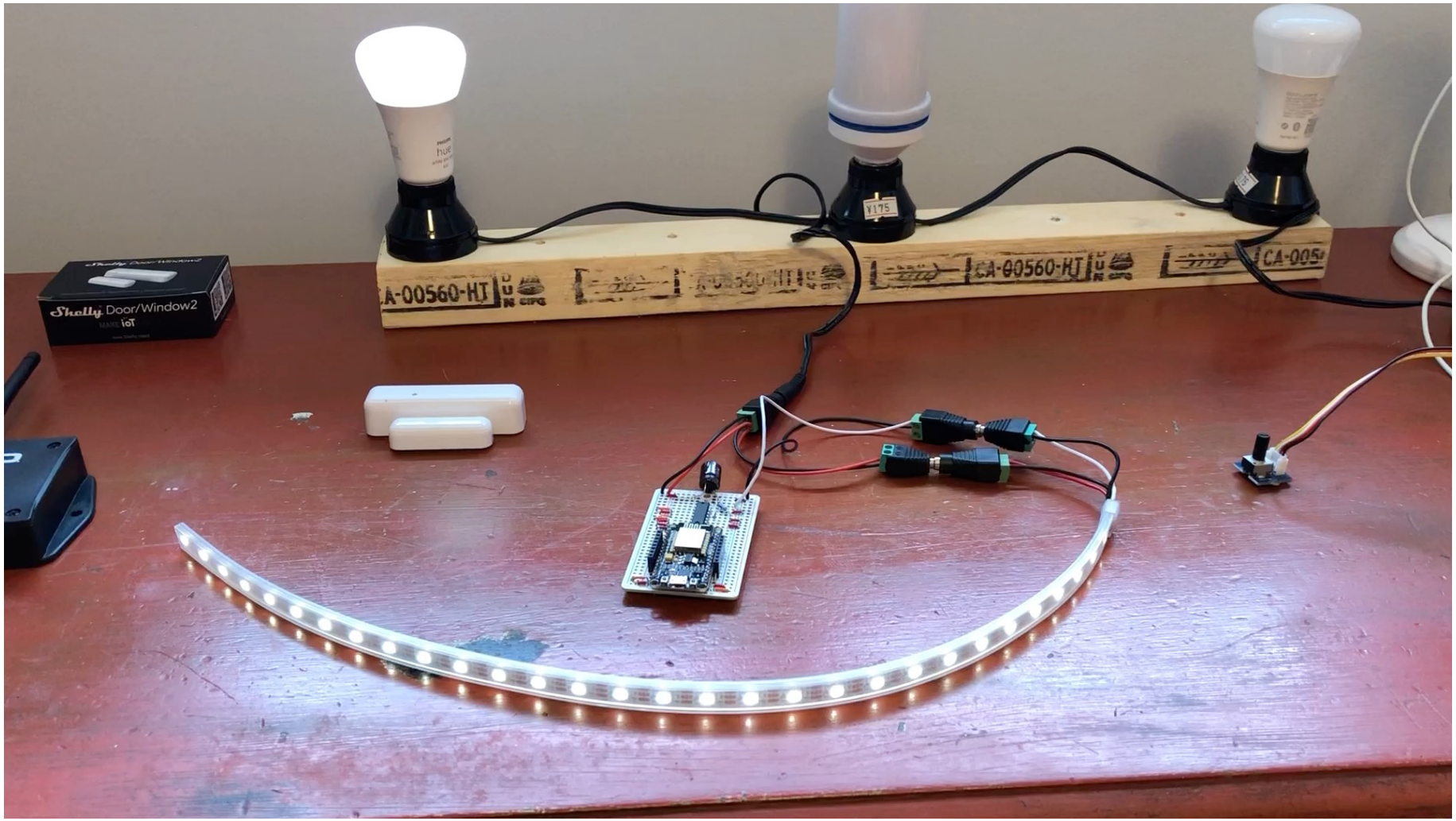




Commercial Devices



Video: System Demo



https://youtu.be/m0F11Vu2wf8

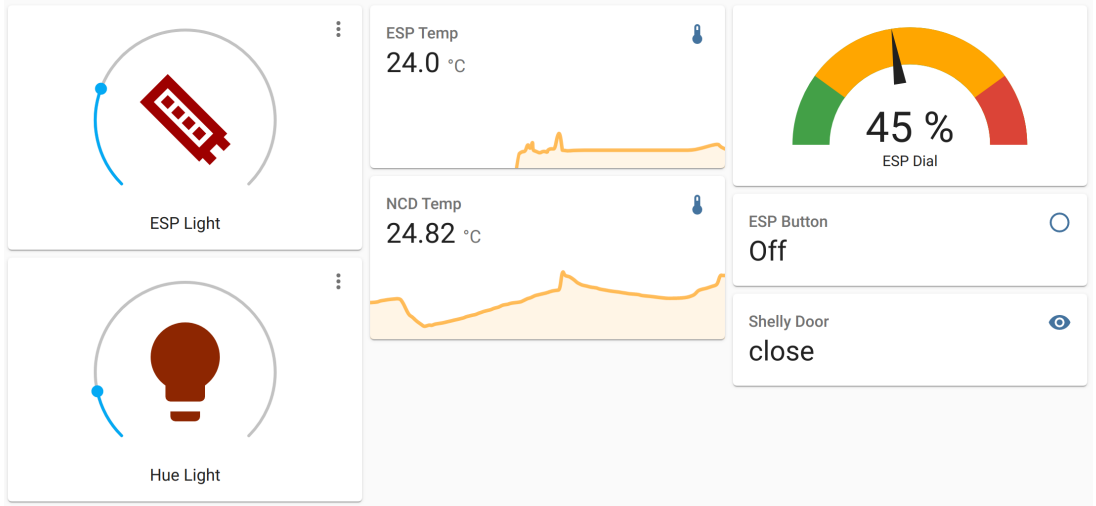
Implementation Approaches

- Proprietary Turnkey System
  - Contract out entire system to a third party
  - Issues: Maintenance, modifications, ownership transfer, business risk
- Proprietary Gateway/Rules Engine (e.g. Smart Things)
  - Integrations limited to those provided by vendor, business risk, portability
- Open-source Gateway/Rules Engine (e.g. Home Assistant, Node-RED)
  - Crowdsourced integrations, variable quality, business risk, portability
- Standards-based Gateway/Rules Engine (e.g. node-wot)
  - Subclass of Proprietary and Open-Sourced Gateways
  - Addresses portability concern, which also helps to address business risk

Two Implementations

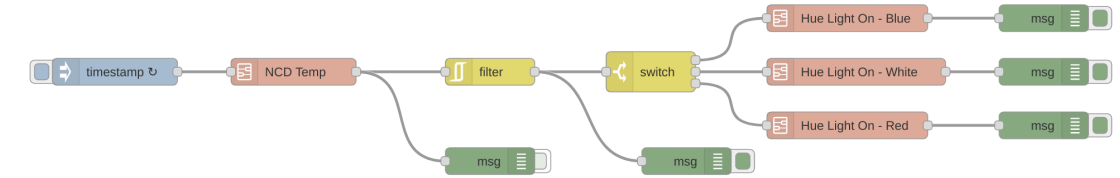
Home Assistant Native Rules

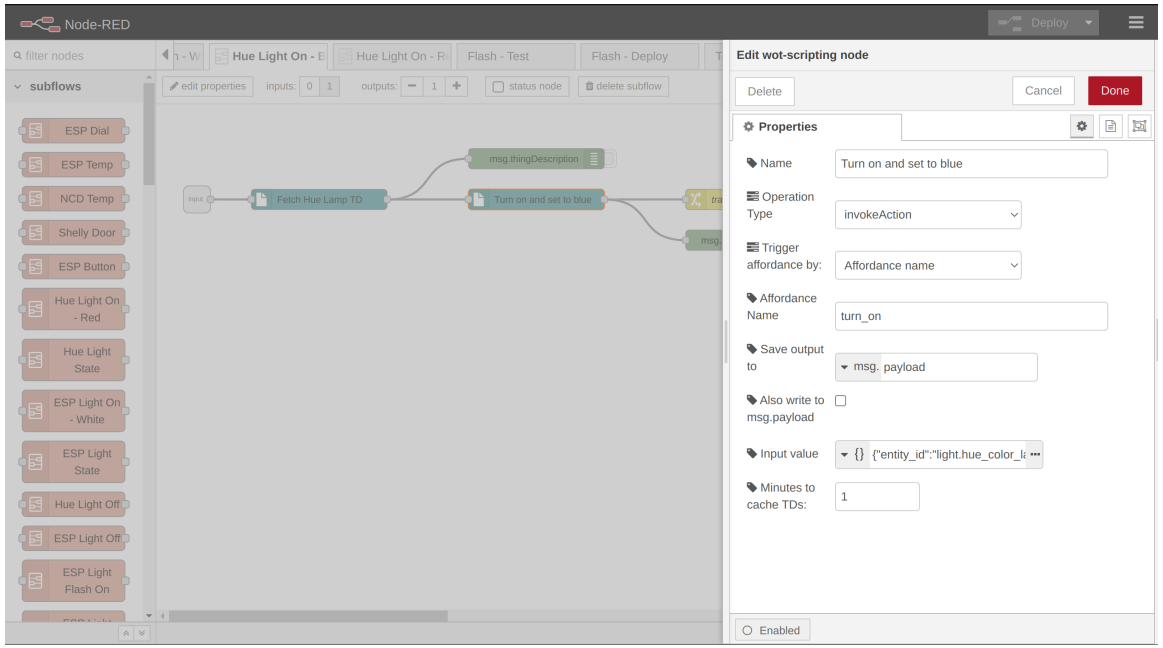
- Use built-in rules engine



Node RED

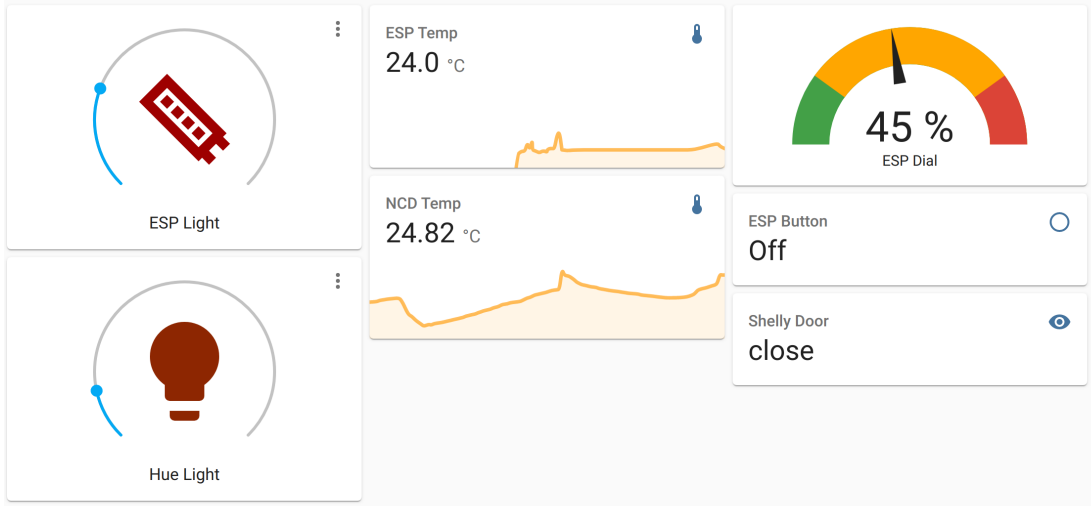
- Rules based on WoT TD/node-wot





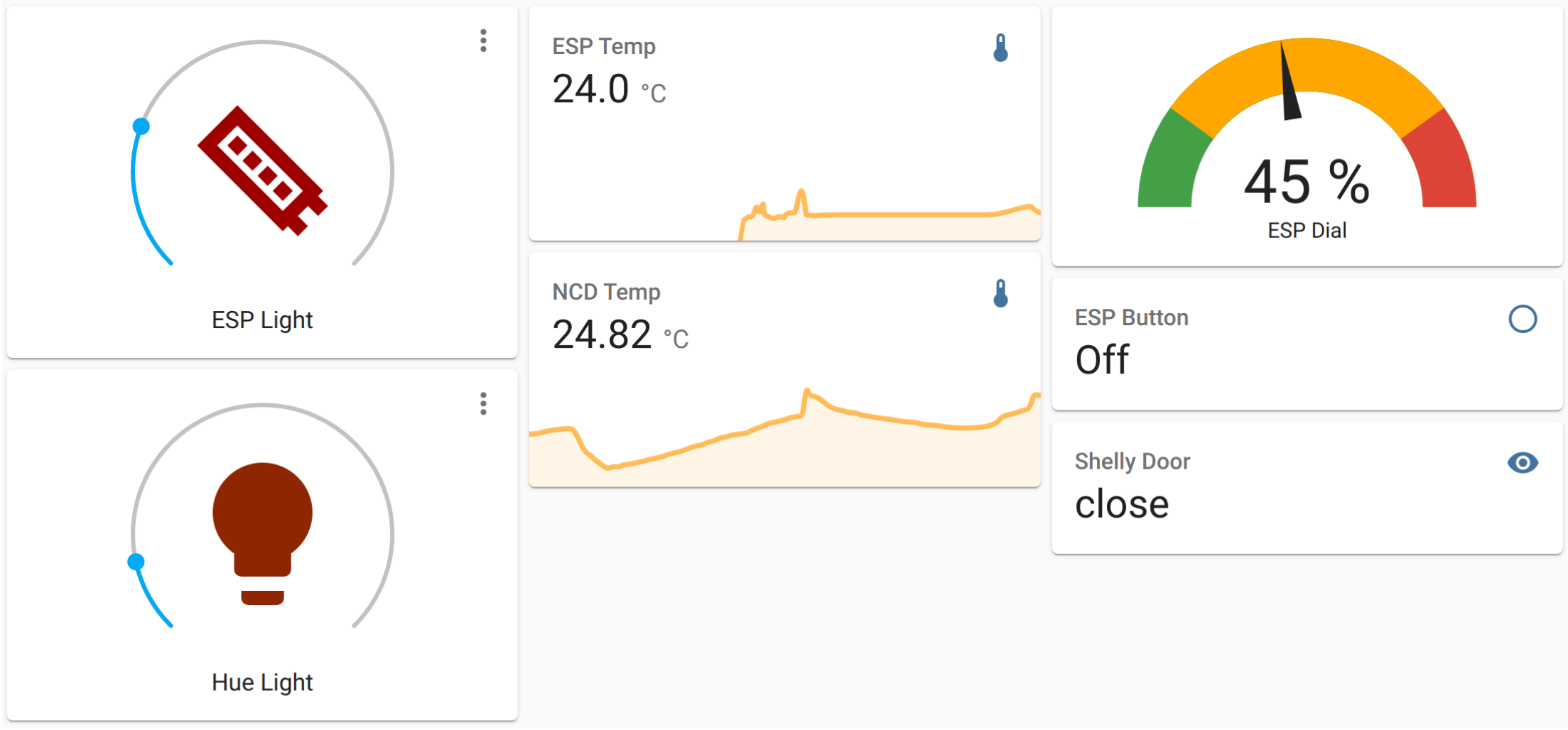
Home Assistant – Native Rule Engine

- HA is used as a "typical" gateway
  - EdgeX Foundry would be more appropriate for retail
- Frontend provides an HTTP API
- Backend supports a variety of device/ecosystem integrations
  - Hue is HTTP (proxy to Zigbee)
  - NCD Temp is MQTT
  - Shelly is MQTT and HTTP
  - Test RGBW strip, dial, button, and temp sensors are MQTT and HTTP, generated via ESPHome

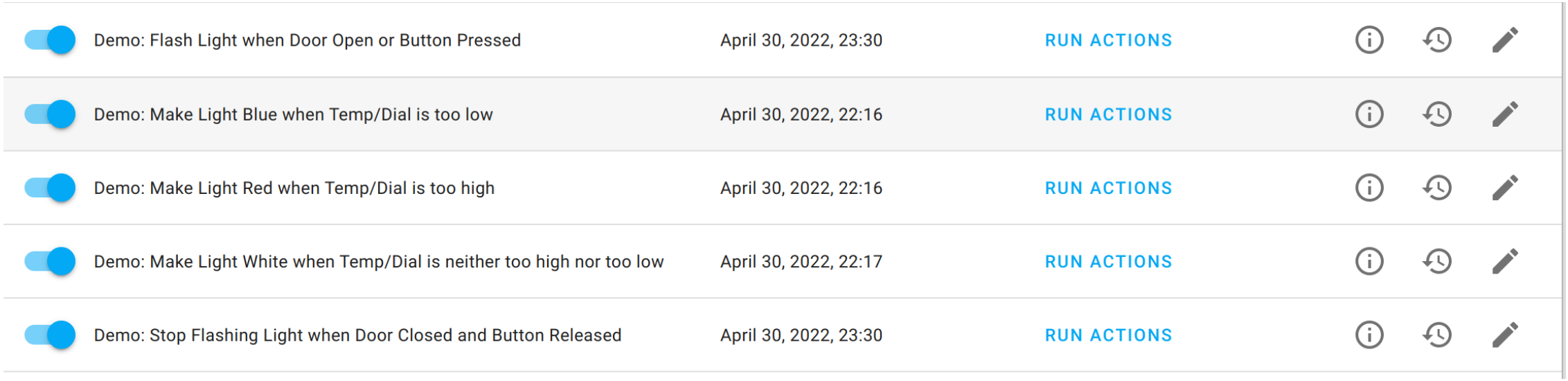


- *Automation rules are based on HA-specific YAML configuration scripts*

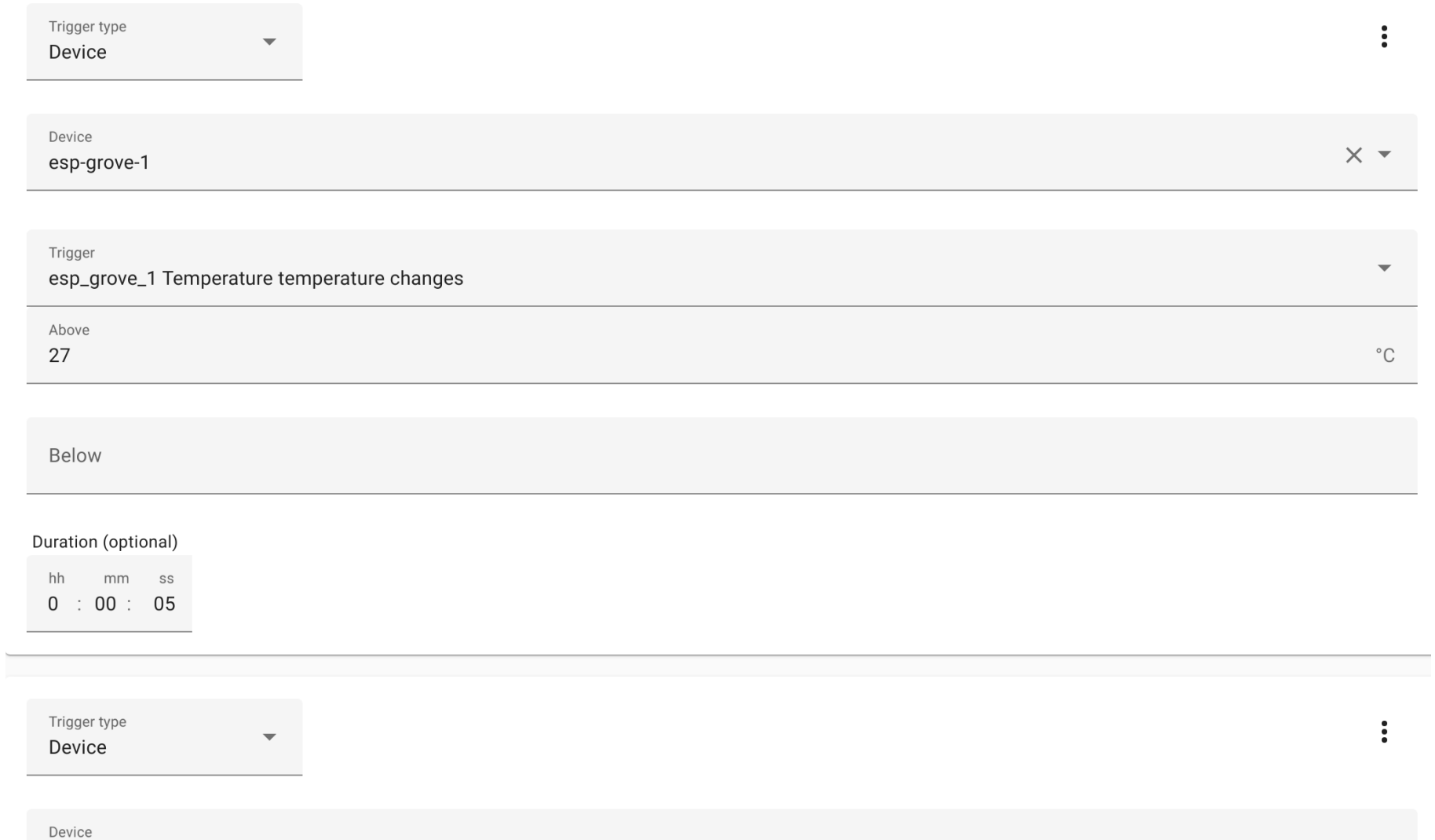
Home Assistant – Dashboard



Home Assistant – Rules



Home Assistant – Temp High Rule

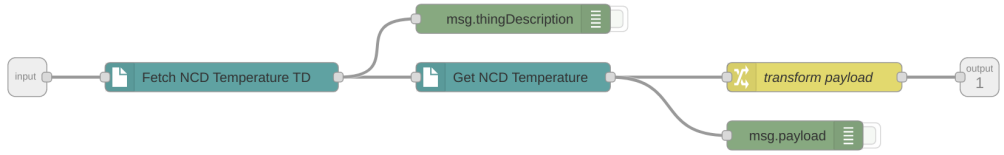


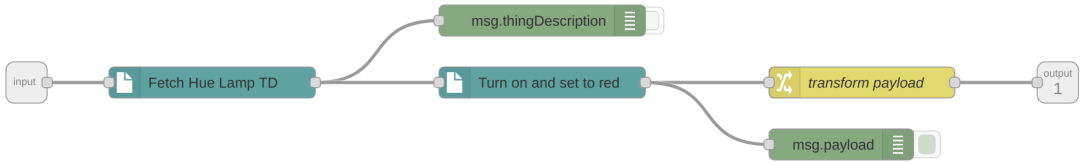
Home Assistant – Temp High Rule

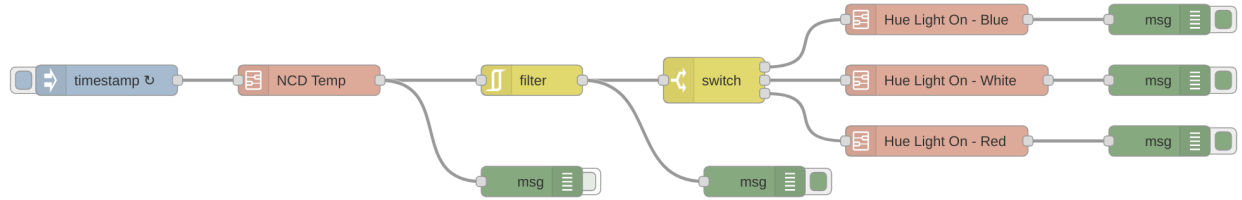
```yaml
alias: 'Demo: Make Light Red when Temp/Dial is too high'
description: ''
trigger:
  - type: temperature
    platform: device
    device_id: 57e2b74dc8beaf0fcb16979d8f1d8dc0
    entity_id: sensor.esp_grove_1_temperature_2
    domain: sensor
    above: 27
    for:
      hours: 0
      minutes: 0
      seconds: 5
  - type: voltage
    platform: device
    device_id: 57e2b74dc8beaf0fcb16979d8f1d8dc0
    entity_id: sensor.esp_grove_1_dial
    domain: sensor
    above: 80
condition: []
action:
  - service: light.turn_on
    data:
      rgb_color:
        - 246
        - 4
        - 4
    target:
      entity_id: light.esp_rgbw_1_rgbw_strip_2
```

Node-RED/WoT Discovery

- Node-RED is an open-source rule engine
- An extension can be used to discover WoT TDs and interface to what they describe
- In the demo we use HA as a gateway
- HOWEVER, we *could* also talk directly to devices via MQTT (for example), given a suitable WoT TD

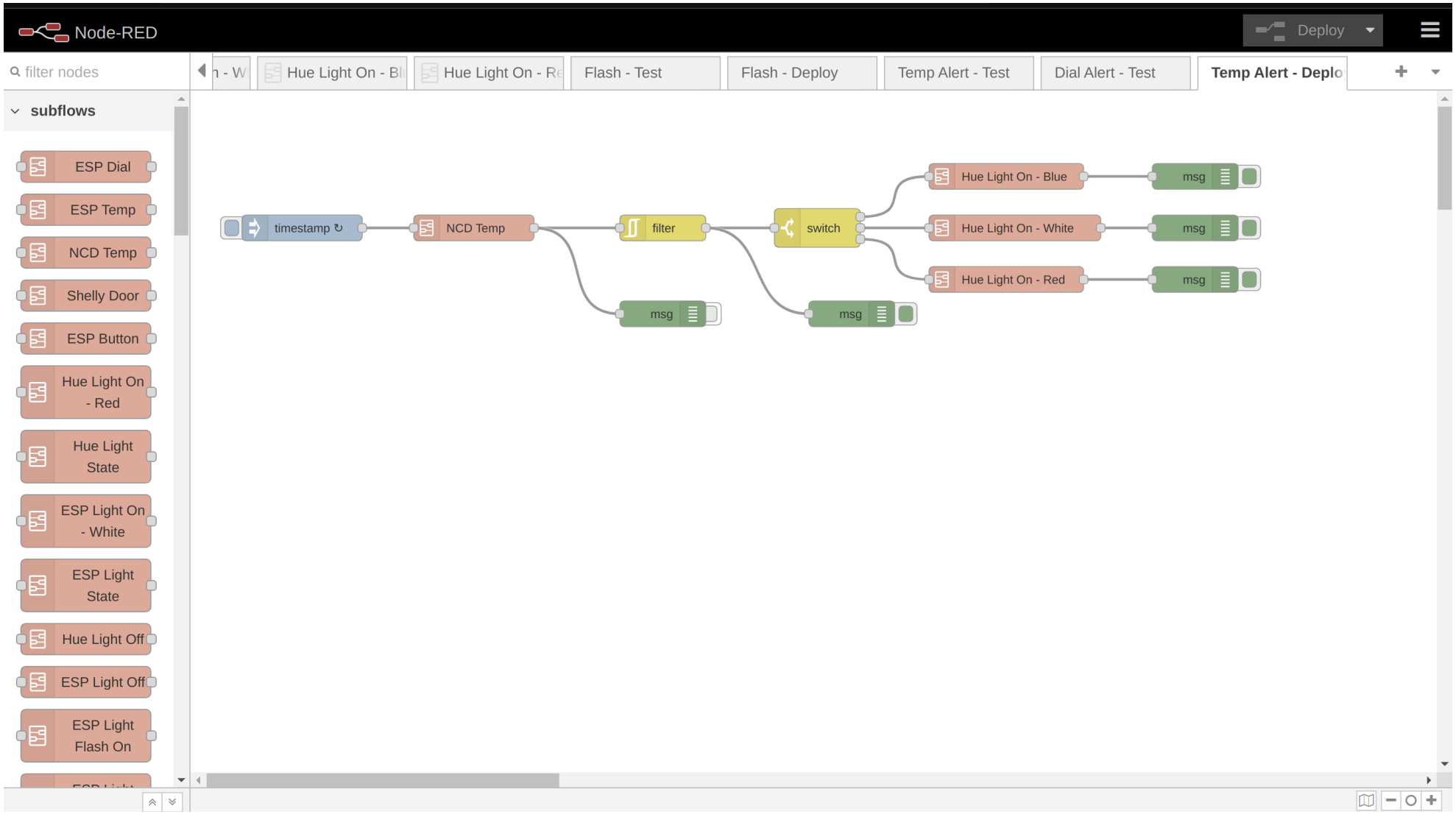




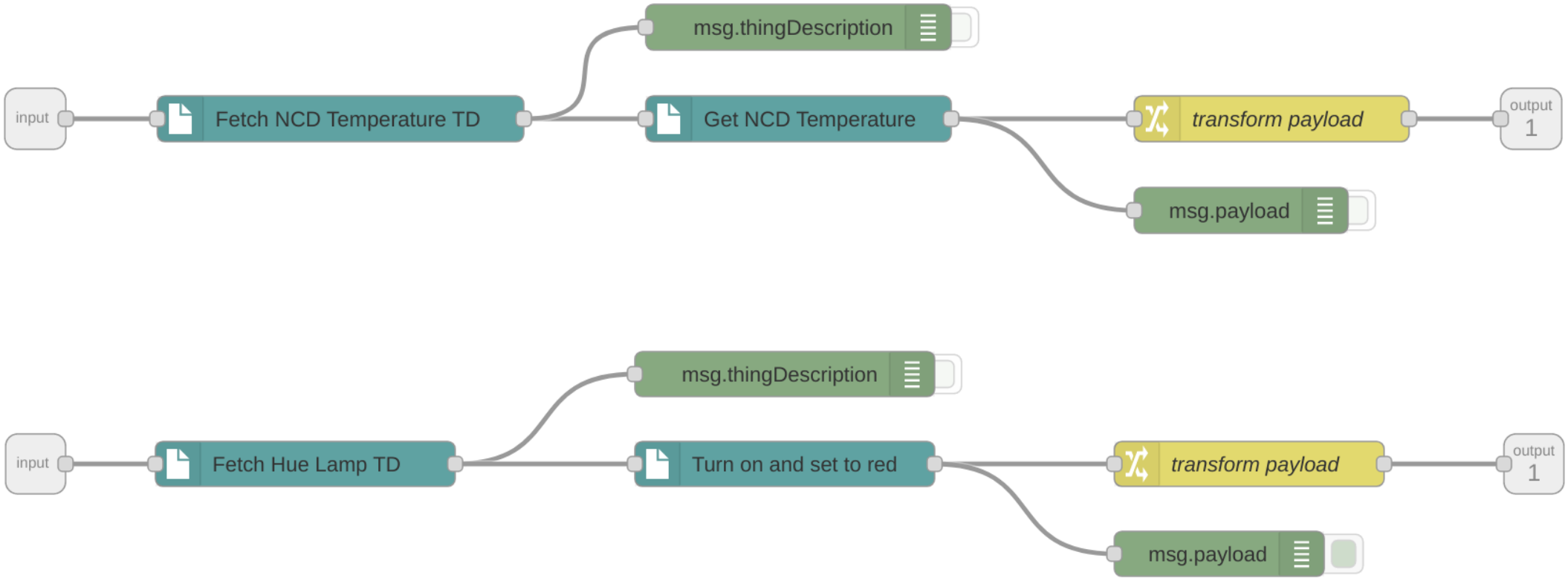


- *Automation rules are based on Node-RED "flows"*

Node-RED – Temp Rule

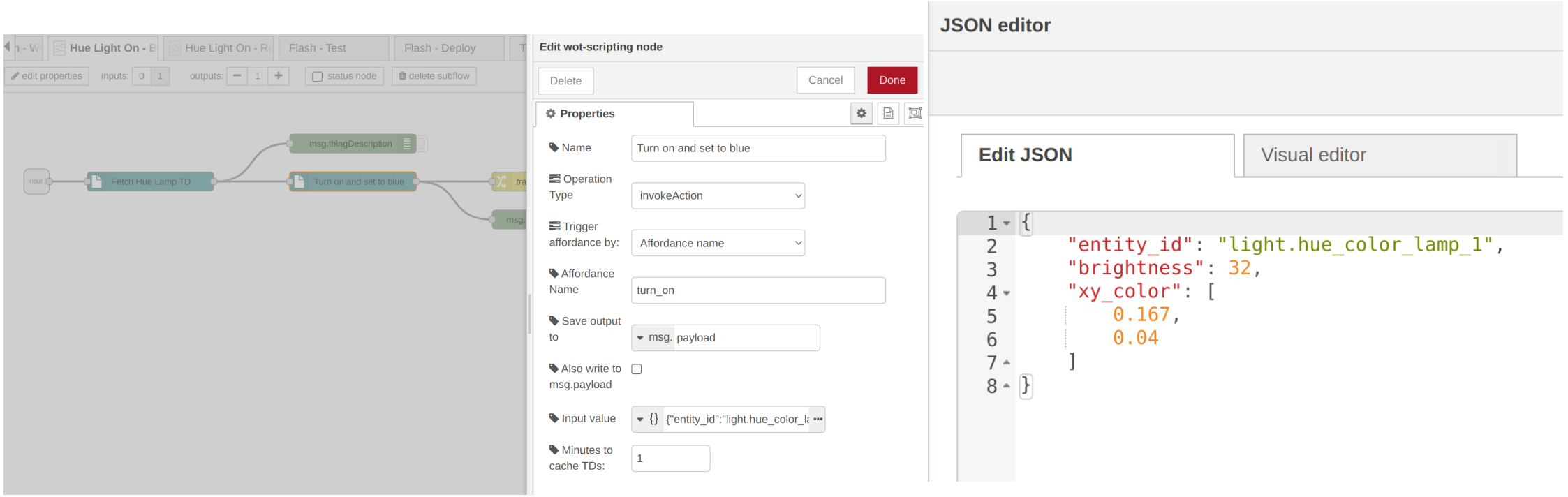


Node-RED Subflows



Interoperability

- WoT TDs describe device interfaces in a standard way
  - Interactions are now is protocol and payload encoding independent
- HOWEVER, devices that do the "same" thing may still have different data models and behavior...



Semantic Interoperability

- WoT Thing Descriptions
  - Currently address "syntactic interoperability"
  - Transform multiple protocols into a common interaction abstraction
  - Transform multiple payload encodings into a common format (JSON)
- Next Steps
  - Annotating Thing Description with semantics
  - Transform data into a standardized data model
  - WoT TDs are encoded using JSON-RD, which provides for *semantic annotation* and inferencing
  - Still need (**Conexxus can contribute here**):
    - Use cases driving required operations and behavior
    - Common semantic vocabularies
    - Standardized data models

Resources and Contacts

https://www.w3.org/WoT

Dr. Michael McCool
Principal Engineer

Intel
Technology Pathfinding

michael.mccool@intel.com

Dr. Sebastian Kaebisch
Senior Key Expert

Siemens
Technology

sebastian.kaebisch@siemens.com