Generic and Extensible Web of Things Manager Using JSON Schema & AI

Andreas Eberhart

dashjoin.com

Ege Korkan

siemens.com

Internet of Things

- Estimated 18 Billion devices connected
- 390 Billion market, expected to double by 2030

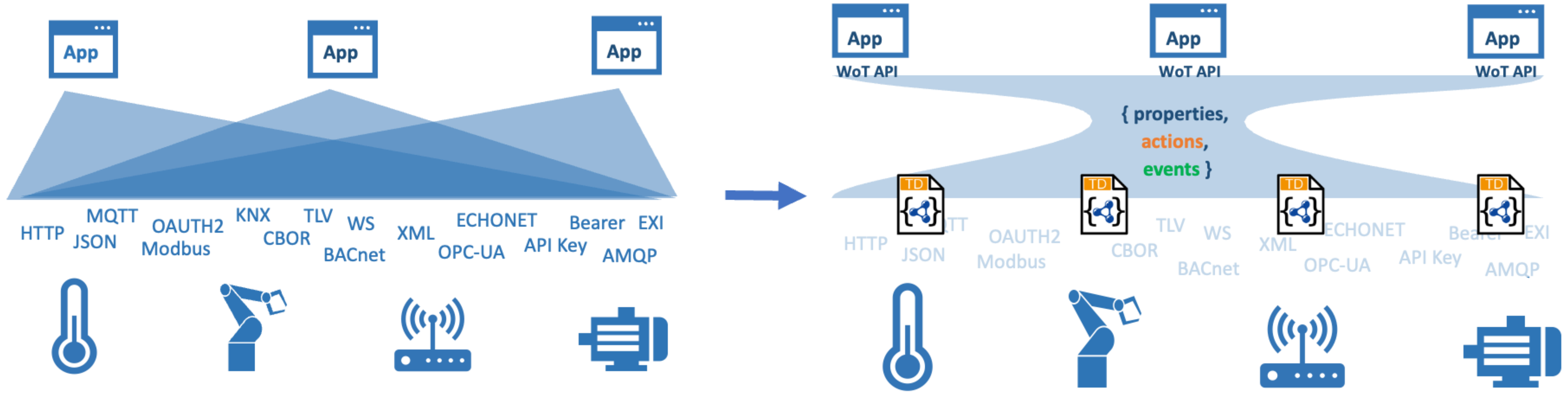
- (statista.com)

IoT
Ecosystem
GEO

https://websites.fraunhofer.de/Lab-IoT/?p=676

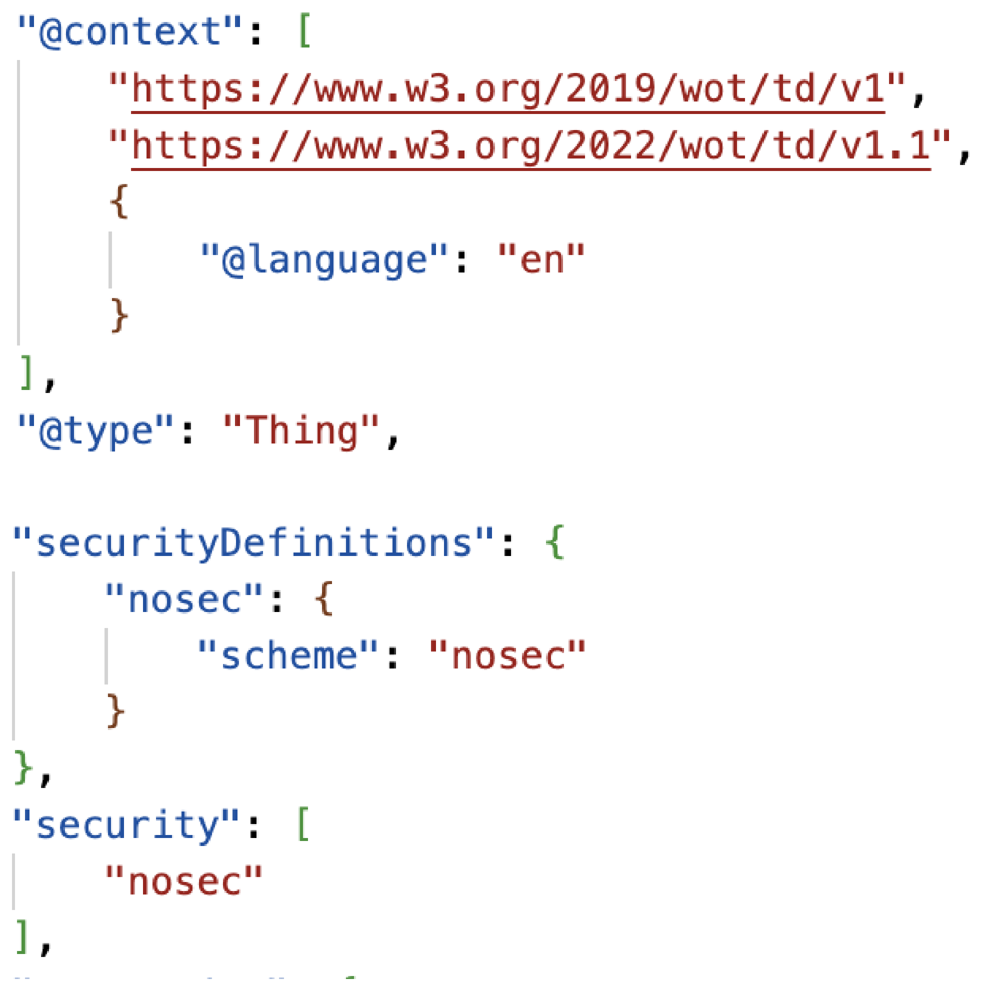
W3C Web of Things (WoT)

- Counter the fragmentation of IoT
  - Use standardized Web technologies
  - Provide device metadata



Thing Description

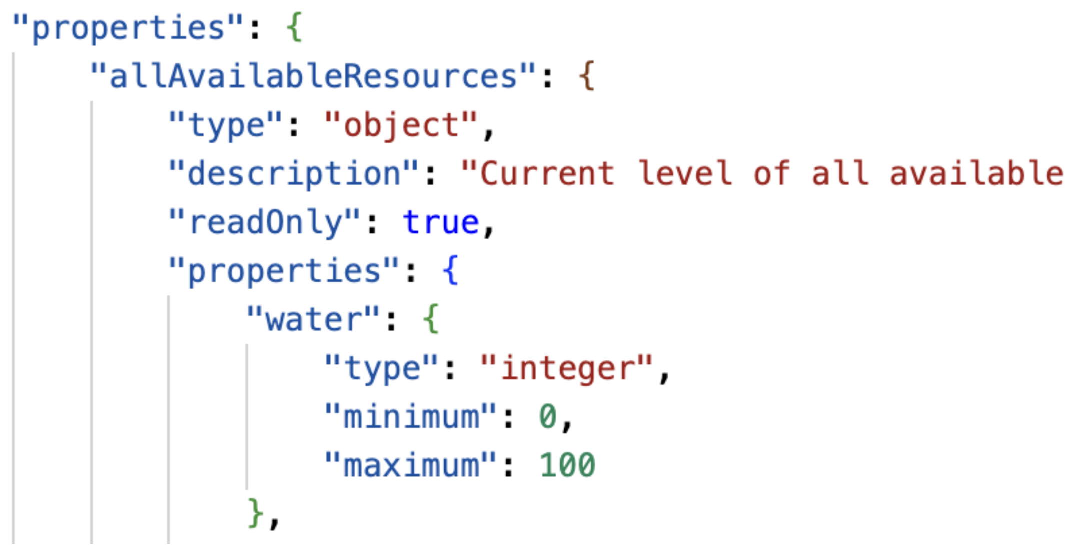
- Example: Coffee Machine
  - o http://plugfest.thingweb.io/http-advanced-coffee-machine
- JSON-LD Context
  - o Describes the device using agreed-upon terminology
- Security Metadata
  - OpenAPI & more
  - Basic auth, OpenID, …



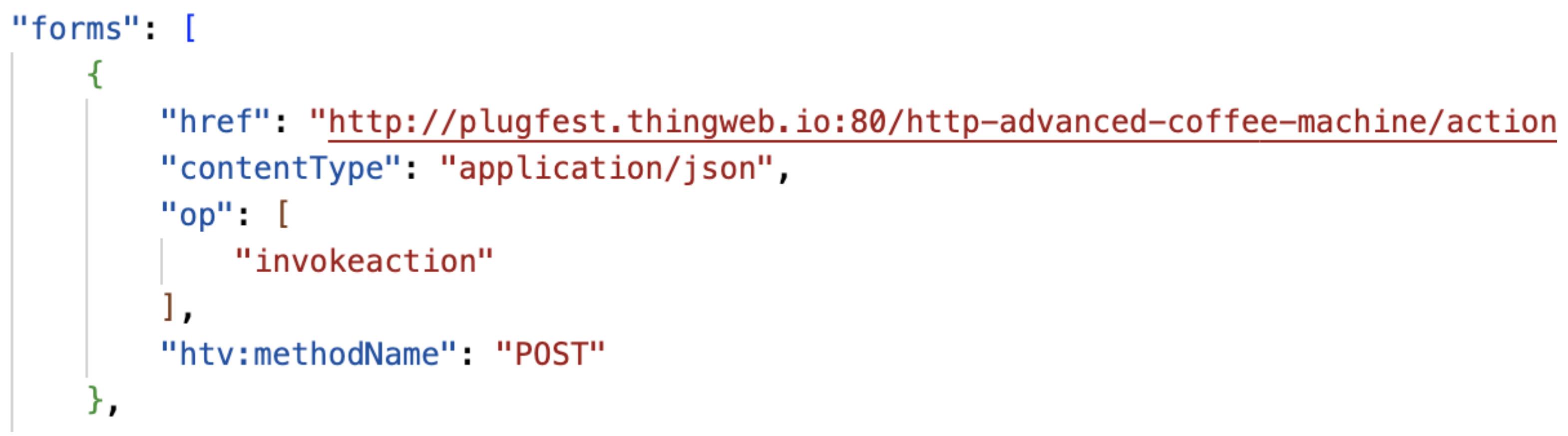
```
"@context": [
    "https://www.w3.org/2019/wot/td/v1",
    "https://www.w3.org/2022/wot/td/v1.1",
    {
        "@language": "en"
    }
],
"@type": "Thing",

"securityDefinitions": {
    "nosec": {
        "scheme": "nosec"
    }
},
"security": [
    "nosec"
],
```

Thing Description

- Data of devices
  - Represented with JSON Schema
- Properties
  - Device configuration & sensors
- Actions
  - Device operations to call
- Events
  - Async device events



```
"properties": {
    "allAvailableResources": {
        "type": "object",
        "description": "Current level of all available
        "readOnly": true,
        "properties": {
            "water": {
                "type": "integer",
                "minimum": 0,
                "maximum": 100
            },
```

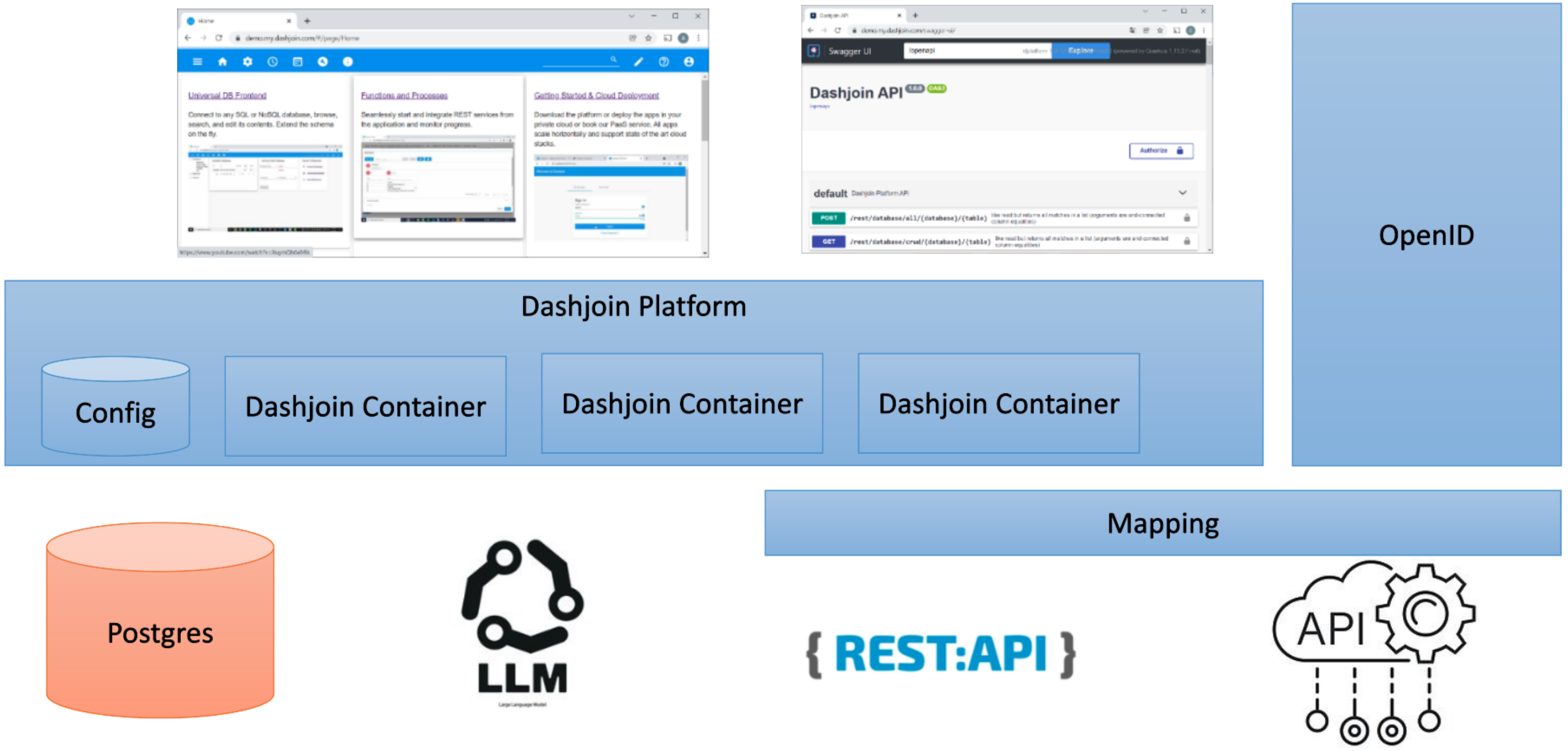


```
"forms": [
    {
        "href": "http://plugfest.thingweb.io:80/http-advanced-coffee-machine/action
        "contentType": "application/json",
        "op": [
            "invokeaction"
        ],
        "htv:methodName": "POST"
    },
```

WoT Manager

- Design Goals
  - Generic: Manage any device
  - Extensible: Allow apps for specific use cases
- Value Add
  - Manage: securely connect devices
  - Control: call device actions
  - Automate: react to device events
  - Analyze: provide overview & dashboards

Architecture



Home

demo.my.dashjoin.com/#/page/Home

Universal DB Frontend

Connect to any SQL or NoSQL database, browse, search, and edit its contents. Extend the schema on the fly.

Functions and Processes

Seamlessly start and integrate REST services from the application and monitor progress.

Getting Started & Cloud Deployment

Download the platform or deploy the apps in your private cloud or book our PaaS service. All apps scale horizontally and support state of the art cloud stacks.

Dashjoin API

demo.my.dashjoin.com/swagger-ui/

Swagger UI /openapi Explore

Dashjoin API 1.0.0 OAS3

Authorize

default Dashjoin Platform API

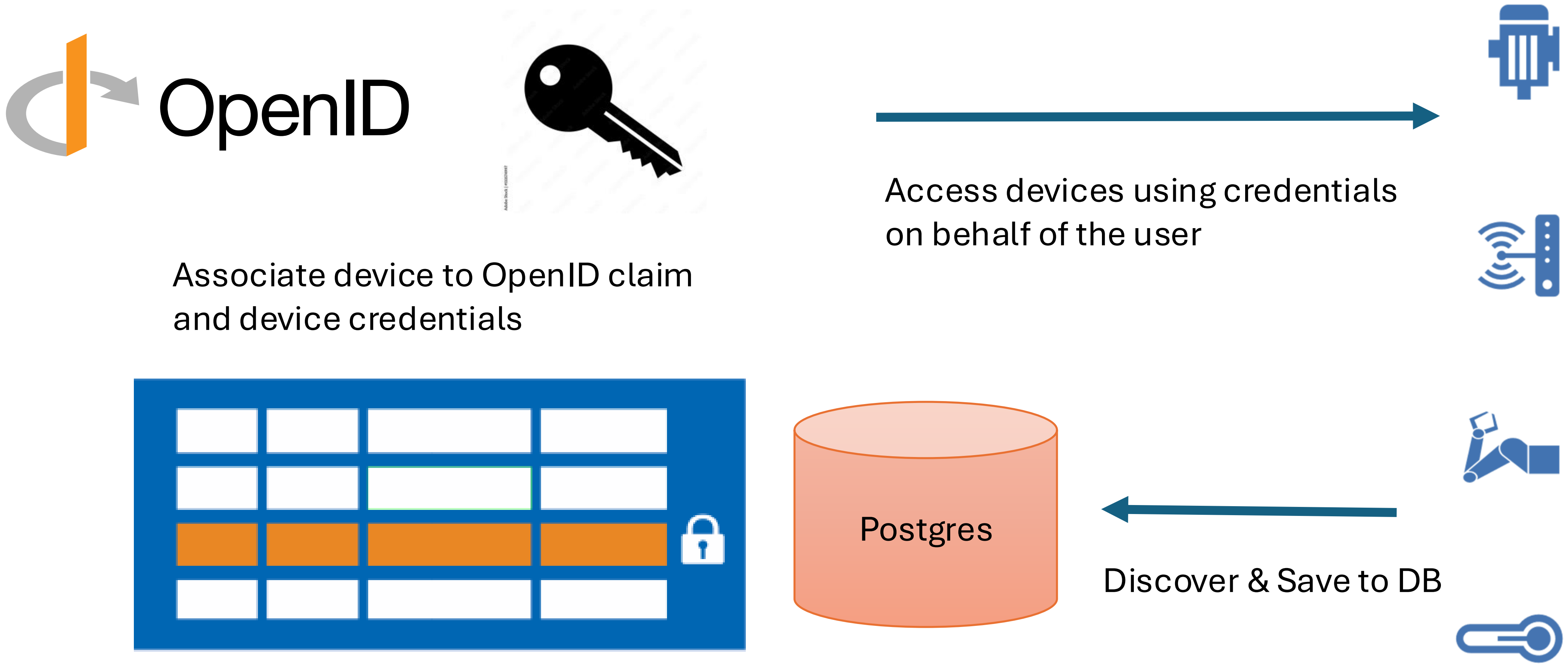
POST /rest/database/all/{database}/{table}

GET /rest/database/crud/{database}/{table}

Discovery & Role Based Access Control

Identity Management

Device credentials



Access devices using credentials on behalf of the user

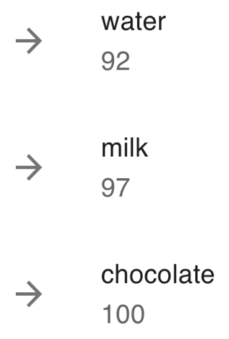
Associate device to OpenID claim and device credentials

Discover & Save to DB

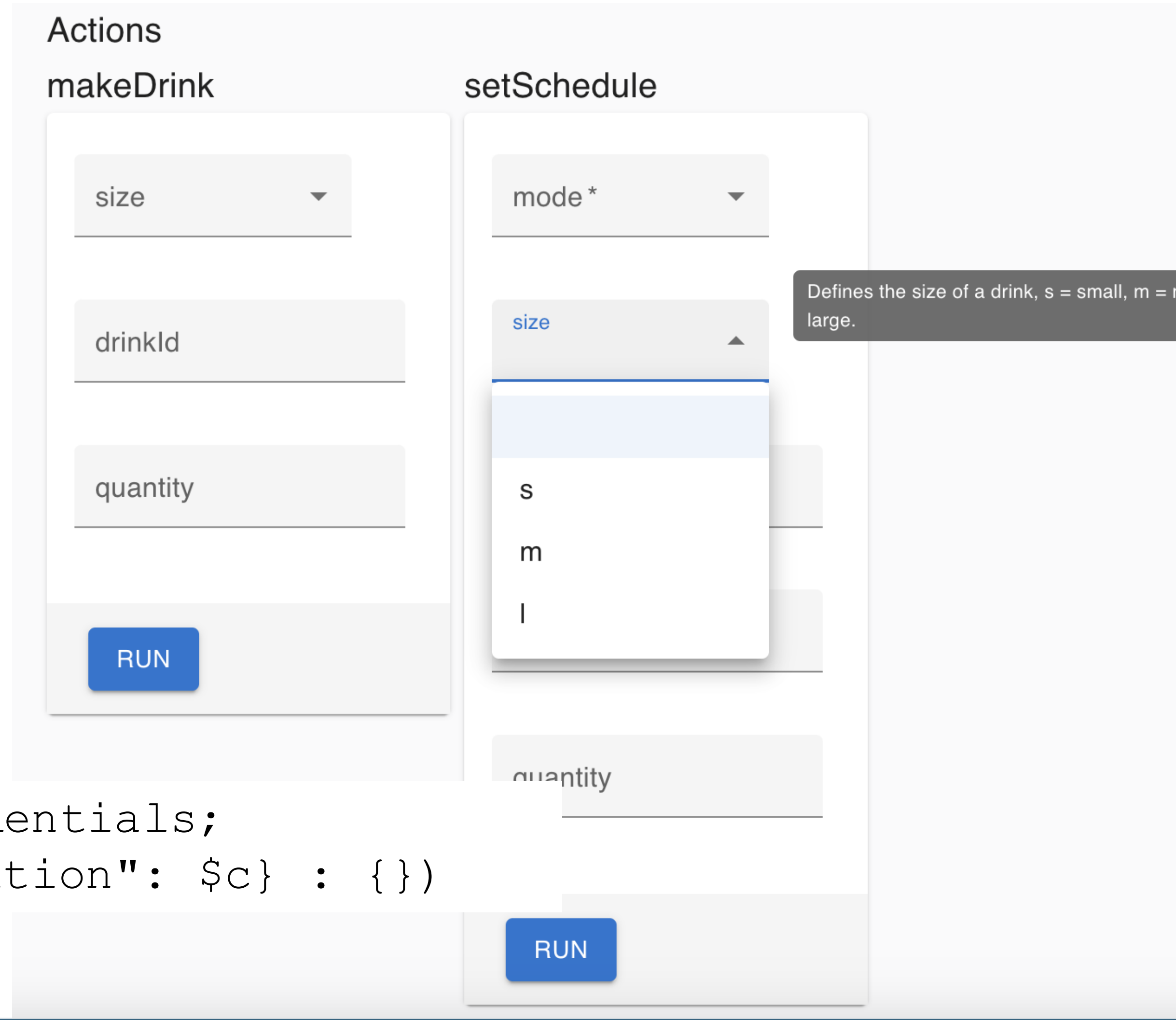
Row-Level Security

Generic Properties & Actions

- On every device page
  - Gather the sensors
  - Display a JSON Schema driven form



water	92
milk	97
chocolate	100



Actions

makeDrink

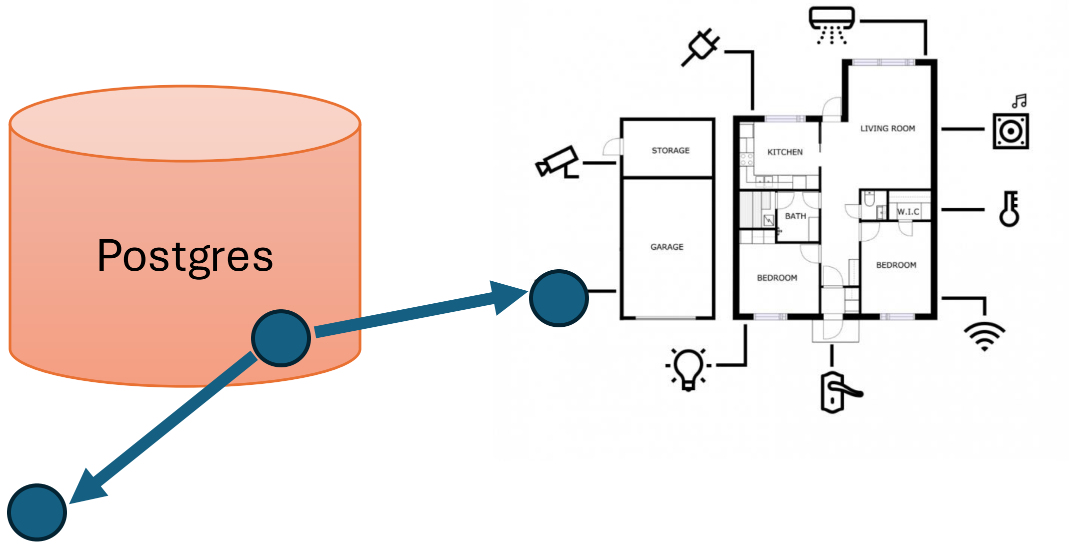
setSchedule

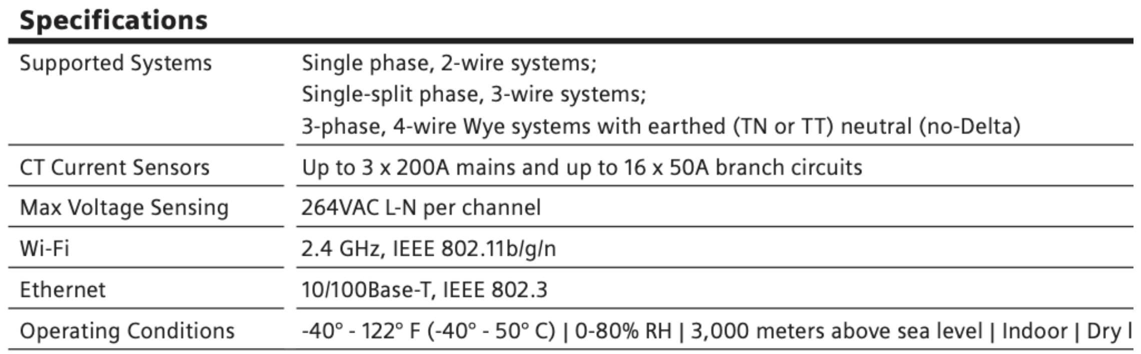
Defines the size of a drink, s = small, m = r
large.

```
$c := $read("wot", "thing", value.thing).credentials;
$curl("GET", value.href, {}, $c ? {"Authorization": $c} : {})
```

Background Knowledge

- Integrate additional information to be able to answer more questions
- Asset DB
  - Where is the device installed?
- Datasheets
  - Additional information about the device





Specifications

Supported Systems	Single phase, 2-wire systems;
	Single-split phase, 3-wire systems;
	3-phase, 4-wire Wye systems with earthed (TN or TT) neutral (no-Delta)

CT Current Sensors	Up to 3 x 200A mains and up to 16 x 50A branch circuits
Max Voltage Sensing	264VAC L-N per channel
Wi-Fi	2.4 GHz, IEEE 802.11b/g/n
Ethernet	10/100Base-T, IEEE 802.3
Operating Conditions	-40° - 122° F (-40° - 50° C) | 0-80% RH | 3,000 meters above sea level | Indoor | Dry |

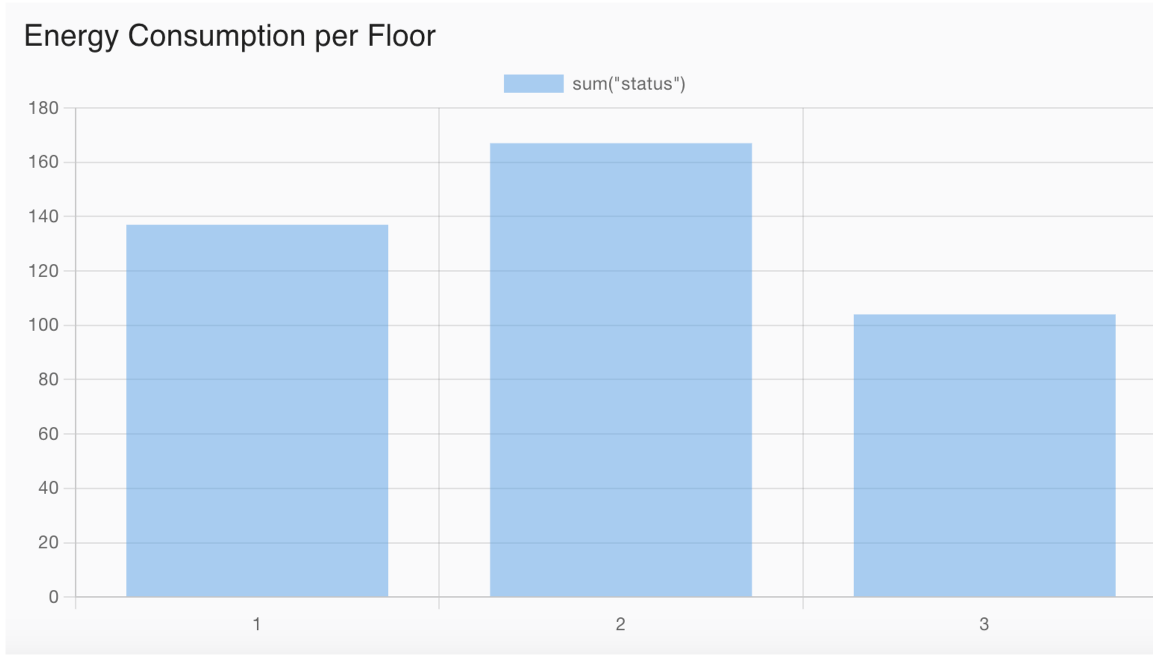
Semantic Data Harmonization

- JSON LD allows grouping similar devices
- Not all things within the group might report data in the same format
- Leverage JSONata to translate into a common format
- Allows dashboarding via SQL

```
{ "watt": 45 }
```

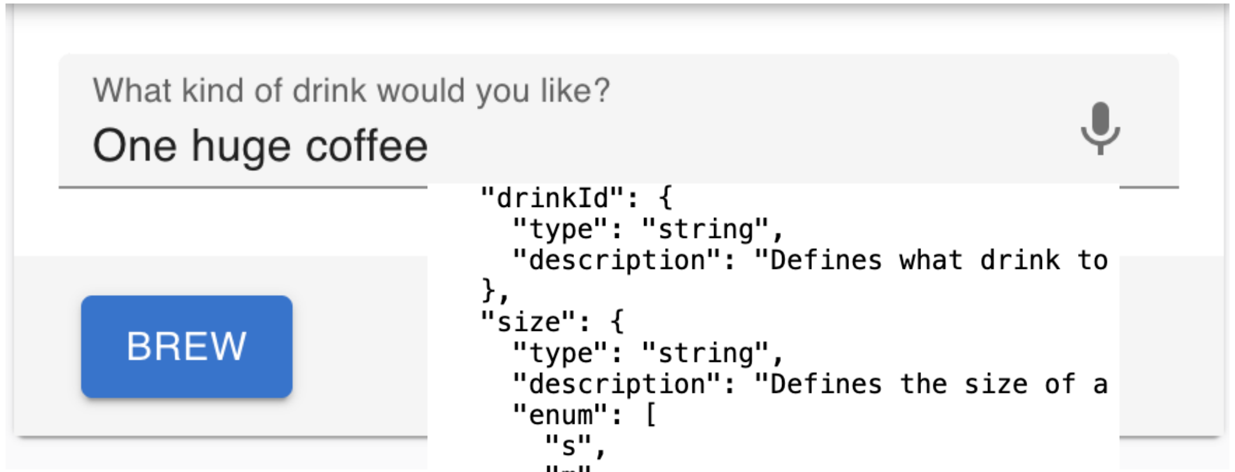
```
"power": {
  "unit": "W",
  "amount": 45
```

```
watt ? watt : power.amount *
(power.unit = 'kW' ? 1000 : 1)
```



Natural Language Commands

- Information extraction from text using LLMs and JSON Schema

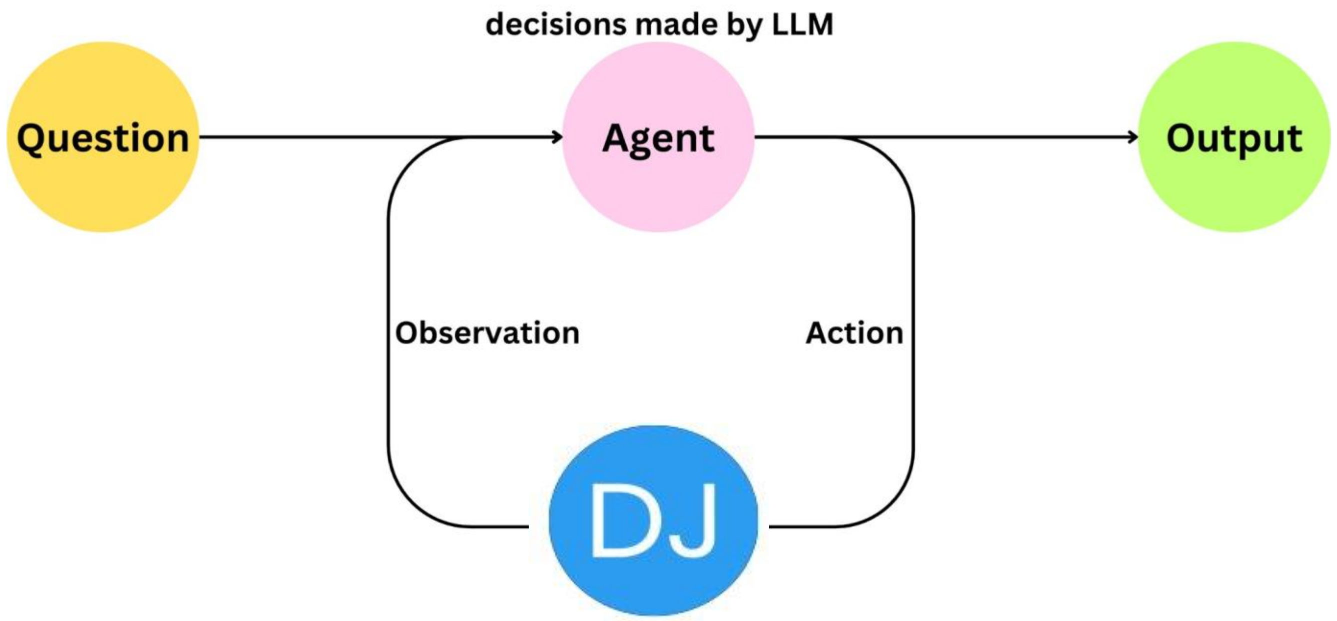


```
  "drinkId": {
    "type": "string",
    "description": "Defines what drink to
  },
  "size": {
    "type": "string",
    "description": "Defines the size of a
    "enum": [
      "s",
      "m",
      "l"
    ]
  },
  "quantity": {
    "type": "integer",
    "description": "Defines how many drink
    "minimum": 1,
    "maximum": 5
  }
},
```

```
{
  "drinkId": "coffee",
  "size": "l",
  "quantity": 1
}
```

WoT Agents

- Expose device actions to the LLM via tools
- User can formulate a higher-level question or goal
- LLM tries to solve the request by leveraging its background knowledge and the provided WoT tools



Example: Turn off all lights in unused office spaces

Wot Agent Log

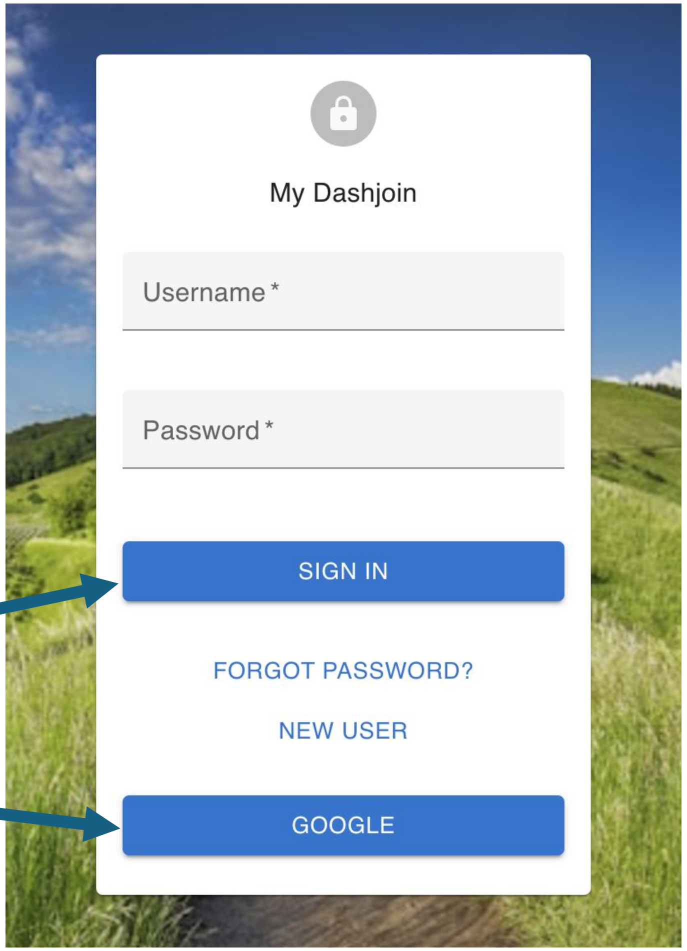
```
Locking system: get a list of persons in the building
Directory service: get LDAP information about persons
Asset DB: get devices by room
Light action: turn lights off/on
```

Here is a sample trace:

```
> Running step 336b4b45-06c4-499f-8bf8-fbf59d943708. Step input: turn off lights in offices where people have gone h
Thought: The current language of the user is English. I need to use a tool to help me identify which offices have pe
Action: lockingSystem
Action Input: {}
Observation: ['joe@emaple.org', 'mike@example.org']
> Running step 637db939-be6e-491a-9a7e-0b80a1498dcf. Step input: None
Thought: I have the list of people currently in the office building. I need to get directory information to determin
Action: ldap
Action Input: {}
Observation: [{'email': 'joe@example.org', 'office': '12b'}, {'email': 'mike@example.org', 'office': '15'}, {'email'
> Running step 8608270f-5f1c-4b84-a74c-5a4946ee739d. Step input: None
Thought: I have the office information for each person. Joe is in office 12b, Mike is in office 15, and Sarah is in
Action: assets
Action Input: {'office': '4a'}
Observation: [{'id': '35313-2426', 'type': 'light-switch'}, {'id': 'af425-f352', 'type': 'energy sensor'}]
> Running step f95c73ed-50a8-42e7-a5a5-aa2303b21cd2. Step input: None
Thought: I have identified the light switch asset in Sarah's office (4a) with the ID '35313-2426'. I will now procee
Action: light
Action Input: {'id': '35313-2426', 'state': False}
Observation: None
> Running step c8d309f5-0a2b-4598-b153-a5dc6d81bfbc. Step input: None
Thought: I have successfully turned off the lights in Sarah's office (4a) as she has gone home. I can now provide th
Answer: The lights in office 4a have been turned off as the occupant has gone home.
```

Demo

- Code
    - https://github.com/dashjoin/djapp-wot
- Live Demo
    - https://wot.run.dashjoin.com/
- Login
    - Create Account using any email
    - Google User



Summary

- Standardization is important
  - WoT thing description (TD) represents any device in a uniform way
  - JSON Schema + TD describes how to interact with devices
  - JSON-LD allows categorizing devices
- Allows developing generic software
  - Works with any kind of device
- Lightweight Low Code Approach
  - Minimal code required
  - Easy to extend (custom actions, dashboards)