

W3C Web of Things Status and Applications

Michael McCool

11 January 2023

IEEE CCNC 2023 - IIWoT Workshop

Outline

- W3C Activity
 - Groups: IG, WG, CG, CG-JP
 - Goal: Applying and extending web standards for IoT
 - Deliverables
 - Descriptive interoperability: Thing Descriptions
 - Finding Thing Descriptions: Discovery
 - Enhancing Interoperability: Profiles
 - Use Cases and Requirements
 - Recent Activity
- WoT Adoption
 - Commercial Applications
 - Open Source Projects
 - Other Engagements
- Discussion
 - Gaps and Future Work

W3C Activity

The W3C supports community-driven development of royalty-free standards.

Groups

W3C Web of Things Interest Group (WoT IG)

- Informative - Exploratory and Experimental work

W3C Web of Things Working Group (WoT WG)

- Normative - Specification Development

W3C Web of Things Community Group (WoT CG)

- English-language community development

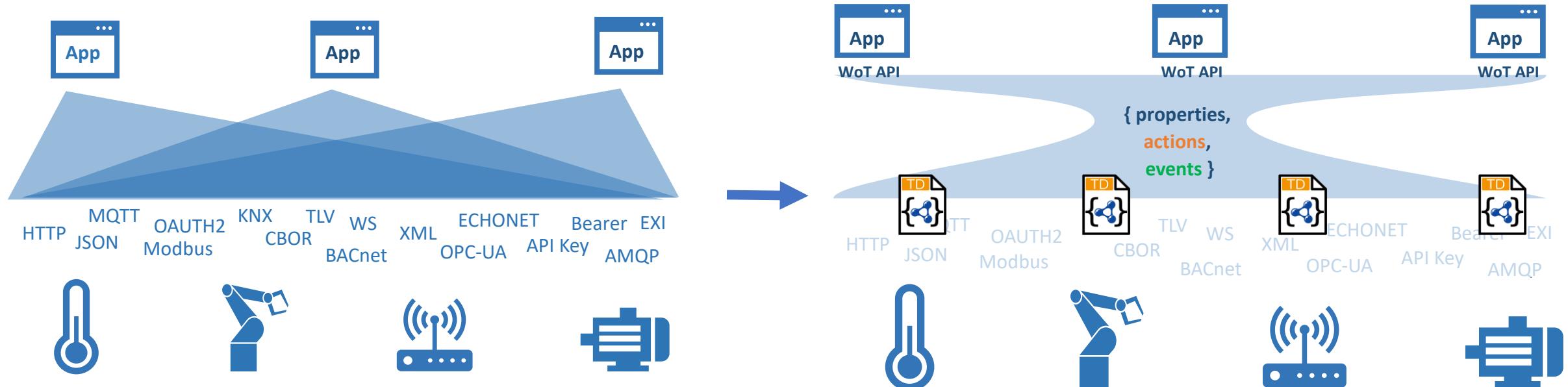
W3C Web of Things Japan Community Group (WoT-JP CG)

- Japanese-language community development

W3C Web of Things (WoT)

W3C WoT 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 and Profiles
 - How does a potential consumer obtain the TD for a Thing?
 - What constraints on TDs are appropriate for best practices and interoperability?



WoT WG and IG Deliverables

New/Updated Normative Documents:

- Architecture 1.1: <https://github.com/w3c/wot-architecture>
- Thing Description 1.1: <https://github.com/w3c/wot-thing-description>
- Discovery: <https://github.com/w3c/wot-discovery>
- Profiles: <https://github.com/w3c/wot-profile>

New/Updated Informative Documents:

- Binding Templates: <https://github.com/w3c/wot-binding-templates>
- Scripting API: <https://github.com/w3c/wot-scripting-api>
- Use Cases and Requirements: <https://github.com/w3c/wot-usecases>

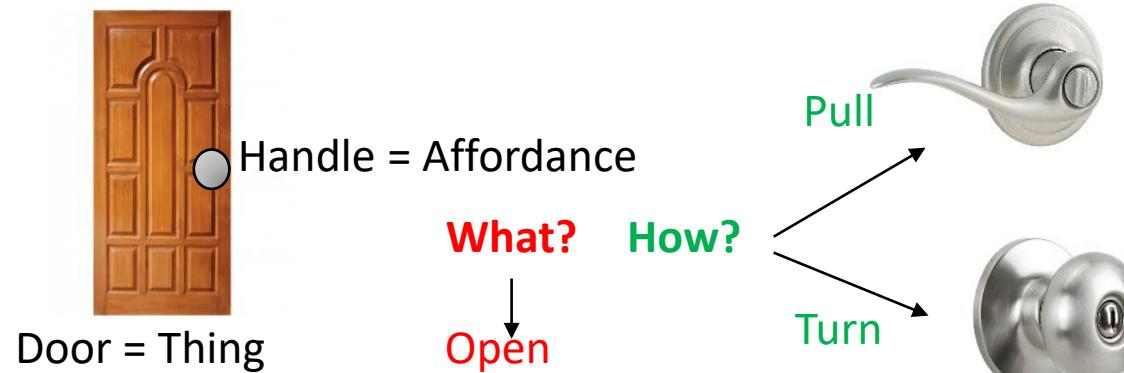
Community Resources:

- Web Site: <https://www.w3.org/WoT/>

Descriptive Interoperability: TDs

WoT Architecture

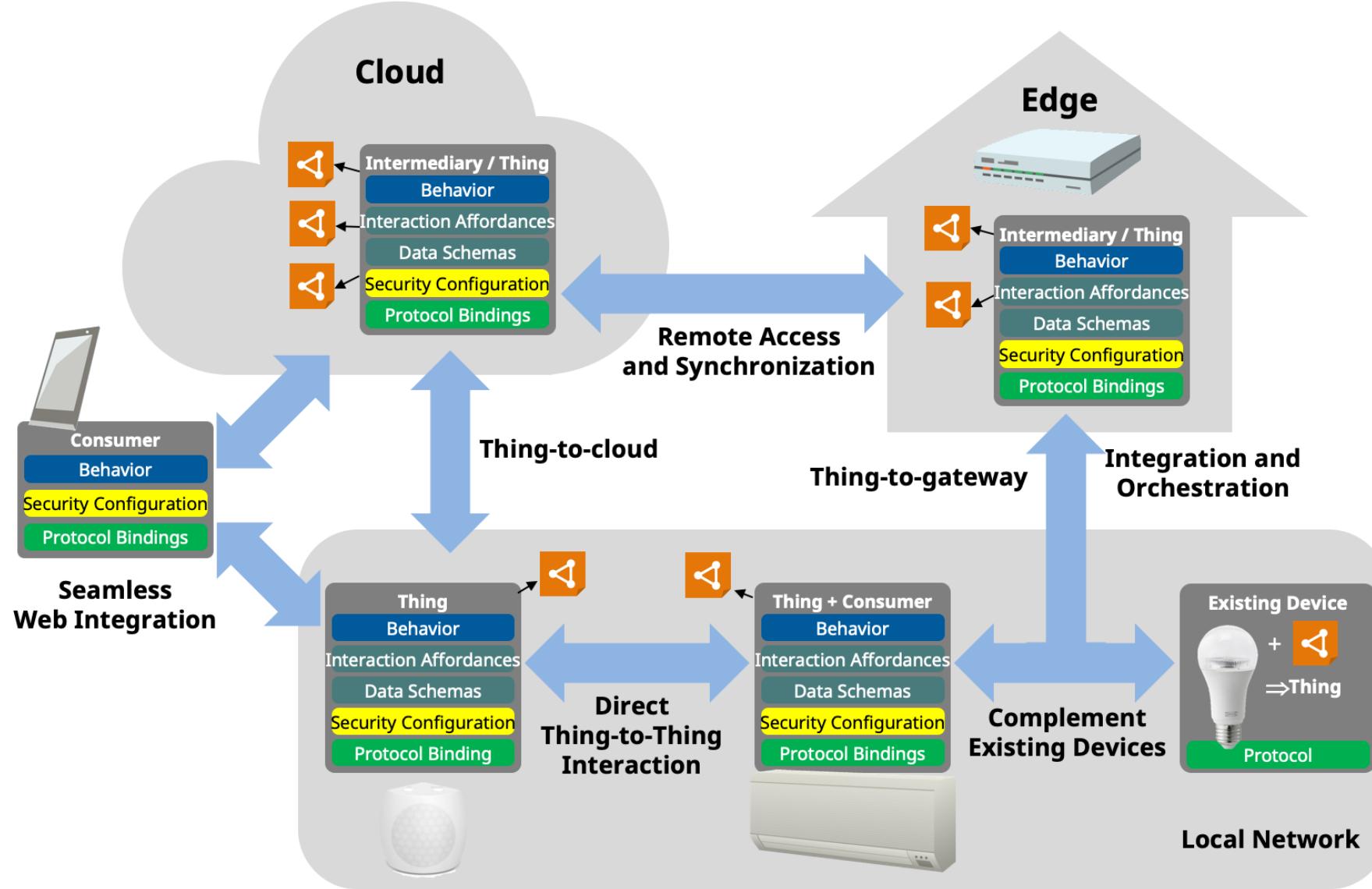
- Constraints
 - "Things" must have a TD
 - Must use URIs, IANA media types, etc.
- Thing Description Affordances
 - Describes **WHAT** the possible choices are
 - Describes **HOW** to interact with the Thing



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": {
      ...
    }
  }
}
```

Usage Patterns Overview



Use Cases and Requirements

Informative Deliverable: <https://github.com/w3c/wot-usecases>

Purpose and Process:

- Identify specific use cases
- Identify application domains
 - Collect use cases from other W3C groups
 - Collect use cases from other stakeholders and SDOs
- Identify usage patterns
 - For example, hubs, proxies, automation, etc.
- Identify relevant technologies
 - For example, edge computing, digital twins, etc.

→ Extract common requirements to drive current and future work

Profiles

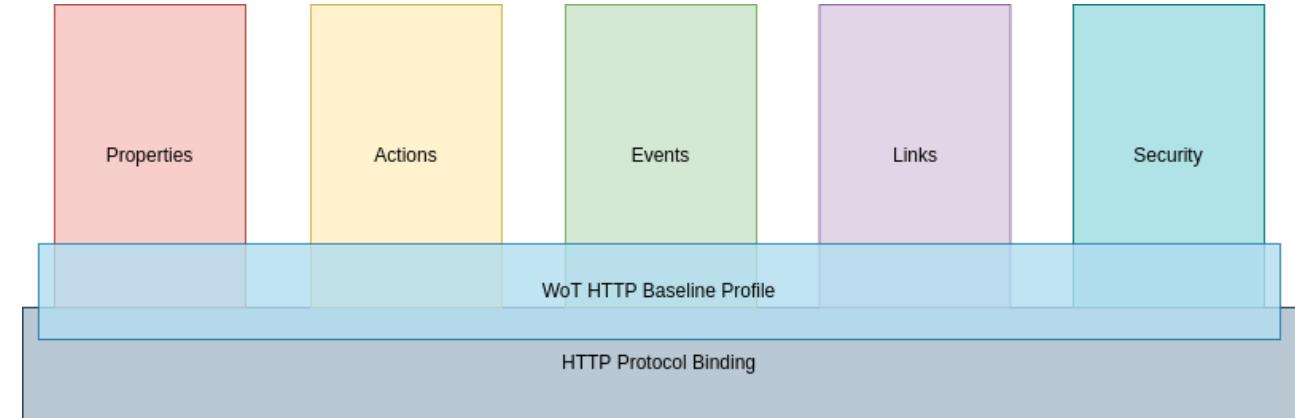
Set of constraints to:

- Improve interoperability
- Define best practices

- ***Prescriptive***

- Why:

- TDs are descriptive
 - Some devices that need to be described do not follow best practices, e.g. for security
- TDs are extensible
 - To interoperate device manufacturers need to know in advance what to expect

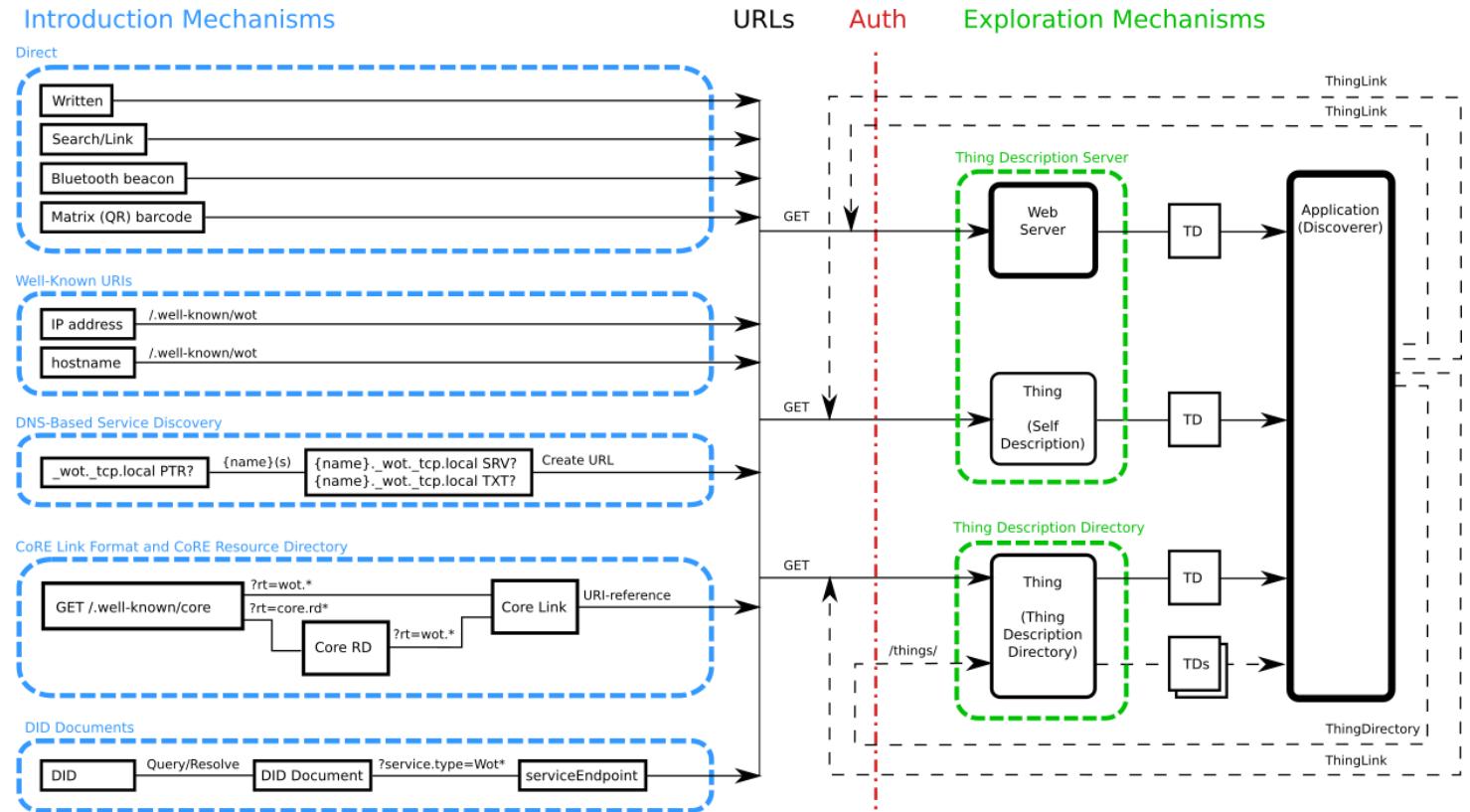


- Currently focused on HTTP Baseline Profile and two sub-profiles for SSE and WebHook Event mechanisms
- TDs satisfying profile set "profile" member and must follow constraints of profile
- Things and Consumers that implement profile also need to satisfy a set of other assertions

Discovery

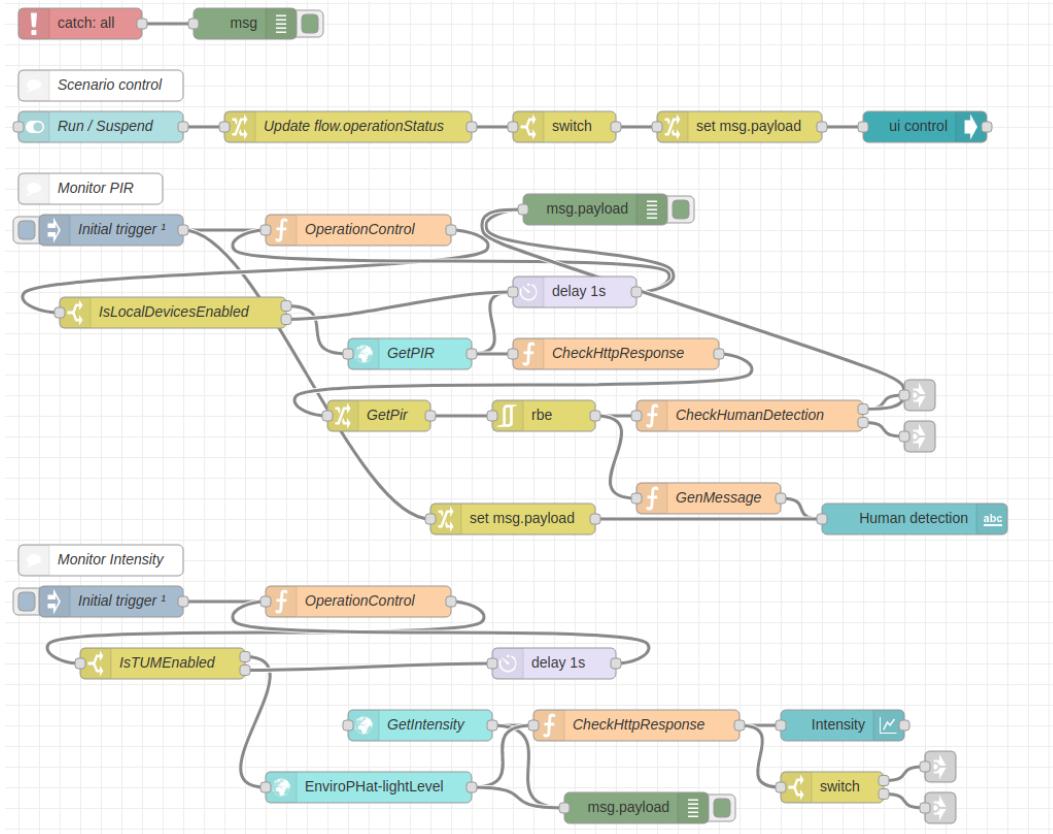
Goal: Obtain TD of interest

- Not limited to local network
- Scalable to many TDs
- Need to preserve privacy
- Phased access:
 1. Introduction: open
 2. Exploration: controlled
- Searchable via JSON Path, XPath, or SPARQL
- Future work:
 - Find "nearby" Things using geospatial data



Orchestration

Node-RED/node-gen



node-wot/Scripting API

```
WoTHelpers.fetch( "coap://localhost:5683/counter" ).then( async (td) => {  
  // using await for serial execution (note 'async' in then() of fetch())  
  try {  
    const thing = await WoT.consume(td);  
    console.info( "==== TD ===" );  
    console.info(td);  
    console.info( "=====*" );  
  } catch (err) {  
    console.error( "Script error:", err );  
  }  
}).catch( (err) => { console.error( "Fetch error:", err );});
```



```
// read property #1  
const read1 = await thing.readProperty( "count" );  
console.info( "count value is" , await read1.value());  
  
// increment property #1 (without step)  
await thing.invokeAction( "increment" );  
const inc1 = await thing.readProperty( "count" );  
console.info( "count value after increment #1 is" , await inc1.value());  
  
// increment property #2 (with step)  
await thing.invokeAction( "increment" , {step: 3});  
const inc2 = await thing.readProperty( "count" );  
console.info( "count value after increment #2 (w/ step 3) is" , await inc2.value());  
  
// decrement property  
await thing.invokeAction( "decrement" , undefined, {  
  formIndex: getFormIndexForDecrementWithCoAP(thing);  
});  
const dec1 = await thing.readProperty( "count" );  
console.info( "count value after decrement is" , await dec1.value());  
  
} catch(err) {  
  console.error( "Script error:", err );  
}  
}).catch( (err) => { console.error( "Fetch error:", err );});
```

WoT Community Group

Goal: Build a community around the Web of Things specifications.

Activities:

- Increase awareness of the WoT specifications
- Collect use cases from the wider community
- Collect implementation experience and references to technologies from the wider community
- Facilitate the implementation of the specifications by providing guidelines and tutorials
- Organize community Meetups

WoT-JP Community Group

1. Outreach
 - To promote WoT with SDOs, Companies, Communities, etc
2. Deployment
 - To Create Libraries, Tools, Documentations for development
3. Use Cases
 - To Discover industrial Use Cases
4. Translation
 - To translate documentations from English to Japanese and from Japanese to English

Recent Activity

- Testing
 - <https://github.com/w3c/wot-testing/tree/main/events>
- Applications and Systems
 - Takenaka, Netzo, Siemens, Deutsch Telekom, Ditto, ...
- Discovery
 - WoT Hive, LogiLab, Fraunhofer LinkSmart, TinyIoT, Zion, Node-RED, ...
- Relationships
 - JSON Path, CoreRD, DID, COSE/JOSE, ASDF/OneDM, OPC UA, Microsoft DTDL, OGC, IEEE, ...
- Under Discussion:
 - New Charter: New Deliverables, Updates to existing specifications, Liaisons
 - Geolocation, Onboarding, Historical data: [proposals/deliverable-proposals](#)

Adoption

The W3C develops and publishes documents, not implementations.

The following implementations were performed by external parties, not all of whom are necessarily W3C members.

Commercial Applications



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

Takenaka Corporation

- [CGLL Platform](#)
- IPA DADC



<https://netzo.io/>

Netzo

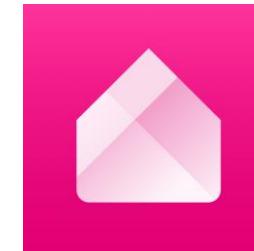
- IoT Data Hub
- Dashboards



<https://www.nhk.or.jp/hybridcast/online/>

NHK

- Hybridcast Integration



<https://www.smarthome.de/magentazuhause-app>

Deutsche Telekom AG

- Smart Home
- App/Hub/Cloud

SCHAEFFLER

<https://www.schaeffler.com/>

Schaeffler Technologies AG

- Manufacturing and Industry

krelian

<https://krelian.com/>

Krelian

- Smart Building Management
- Digital Twins



<https://new.siemens.com/global/en/products/buildings/automation/designo.html>

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

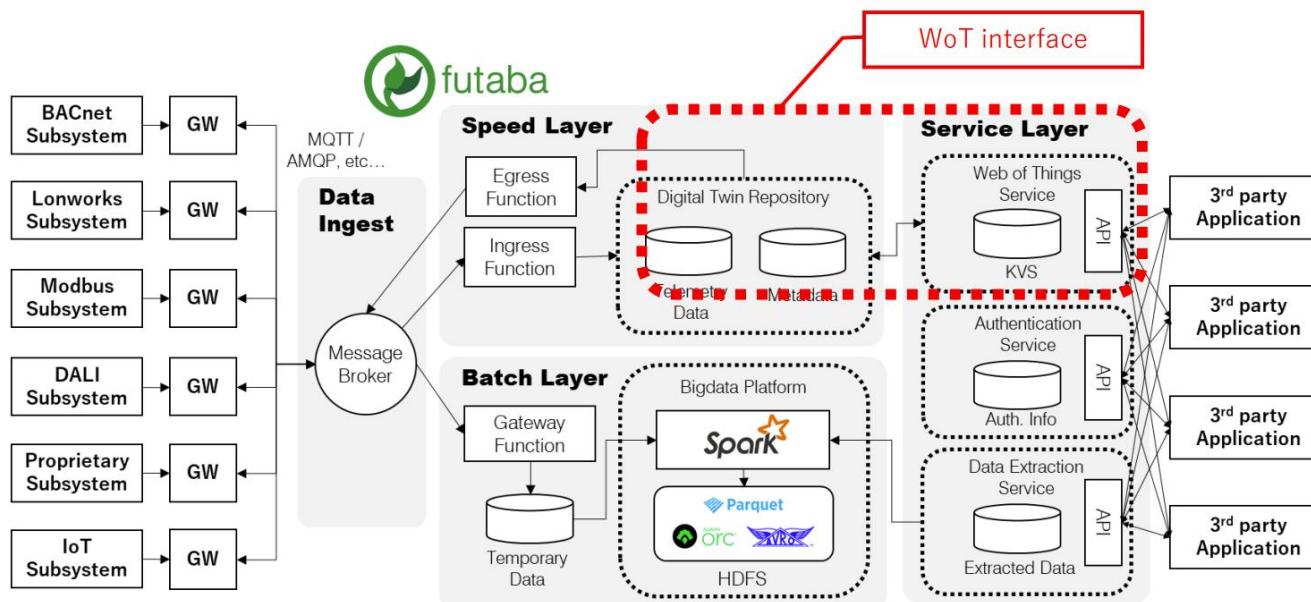
<https://www.evosoft.com/en/application-of-the-w3c-web-of-things-standard-in-the-wunsiedel-hydrogen-generation-plant/>

Siemens AG

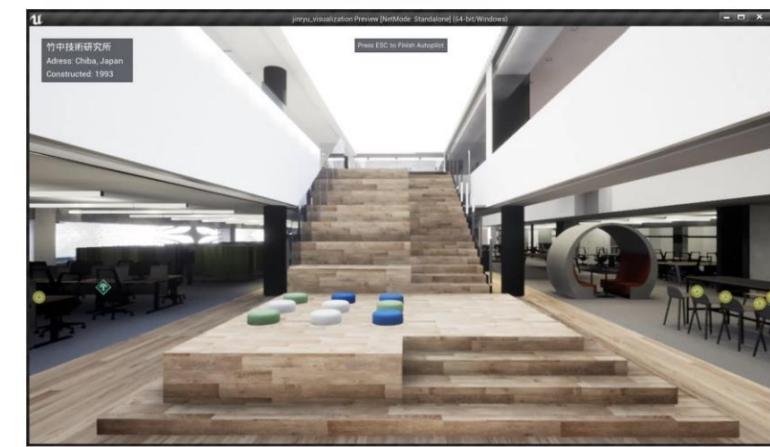
- Designo CC – BIM
- Asset Performance Suite
- Say WoT! (evosoft)
- Wunsiedel (H_2 Generation plant)

Takenaka Corporation

1. Takenaka Corporation has developed a data platform for smart buildings using WoT.
2. Based on the lambda architecture which is a best practice for real-time data processing.
3. WoT is used to implement data models and APIs.
4. It is the foundation for many services such as energy management, AI control, personal control and digital twin.



EQ House : Remote control by reinforcement learning
https://www.takenaka.co.jp/eq_house/

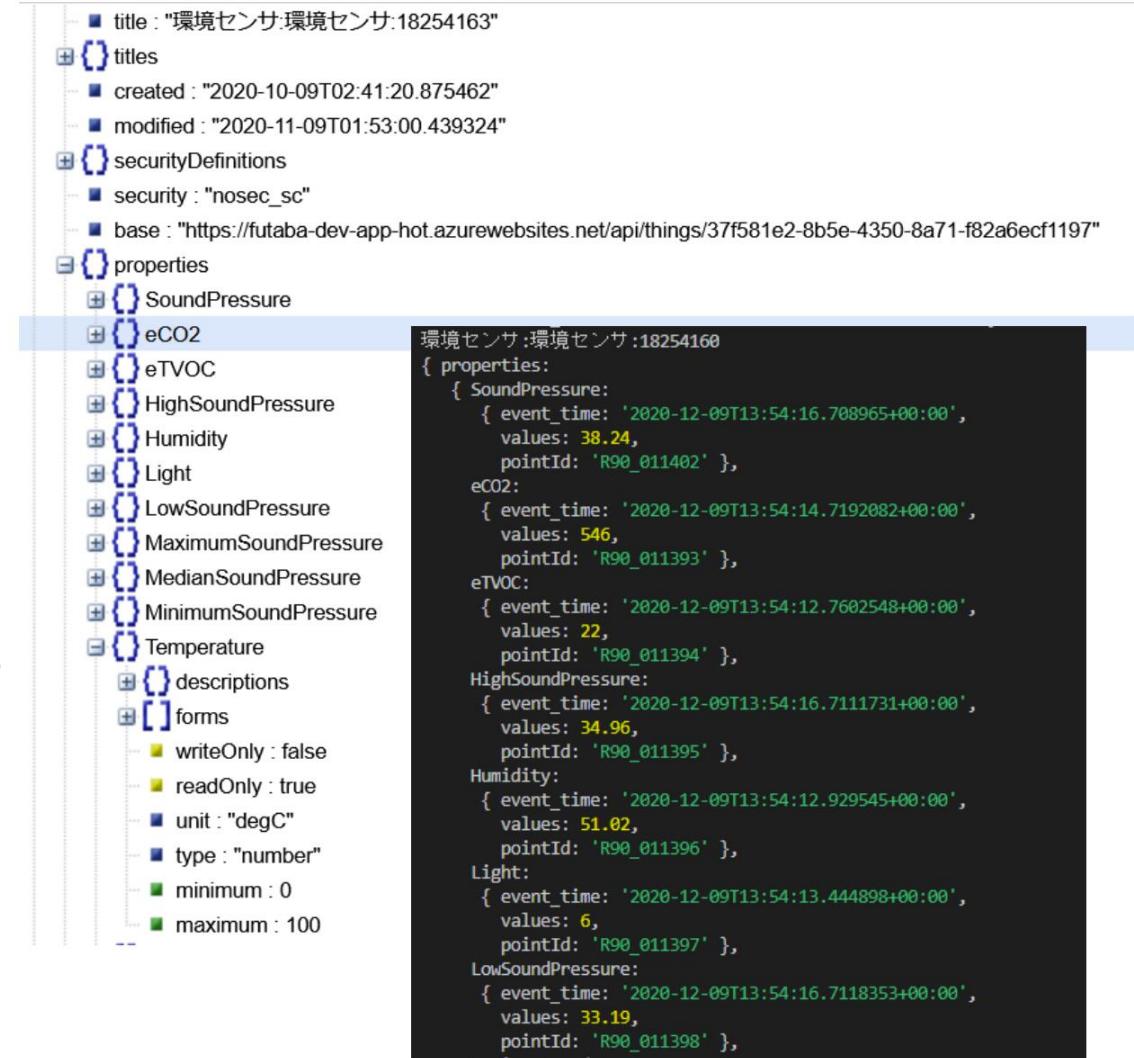
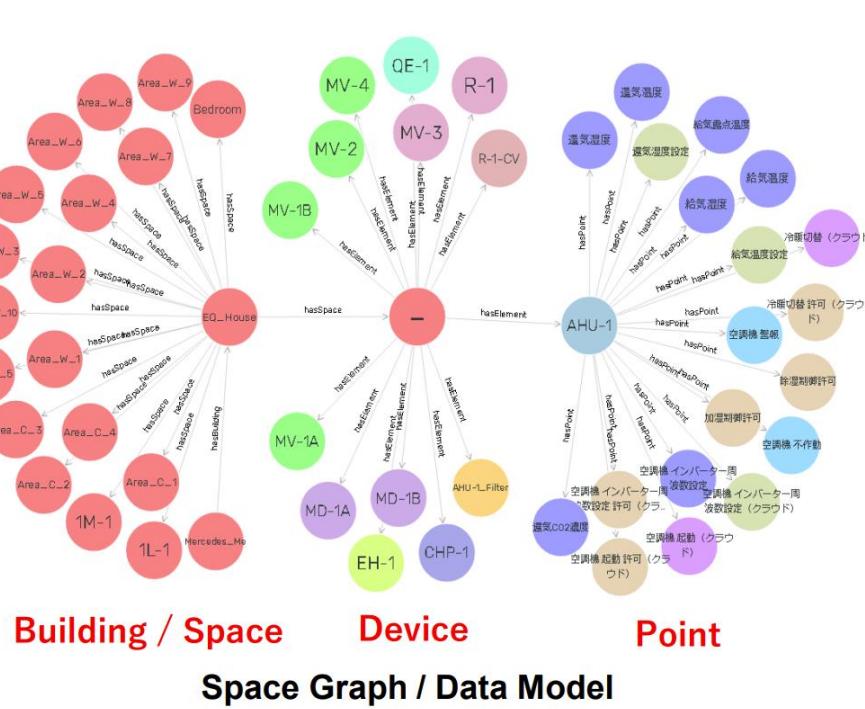


Digital Twin Application

Takenaka Corporation

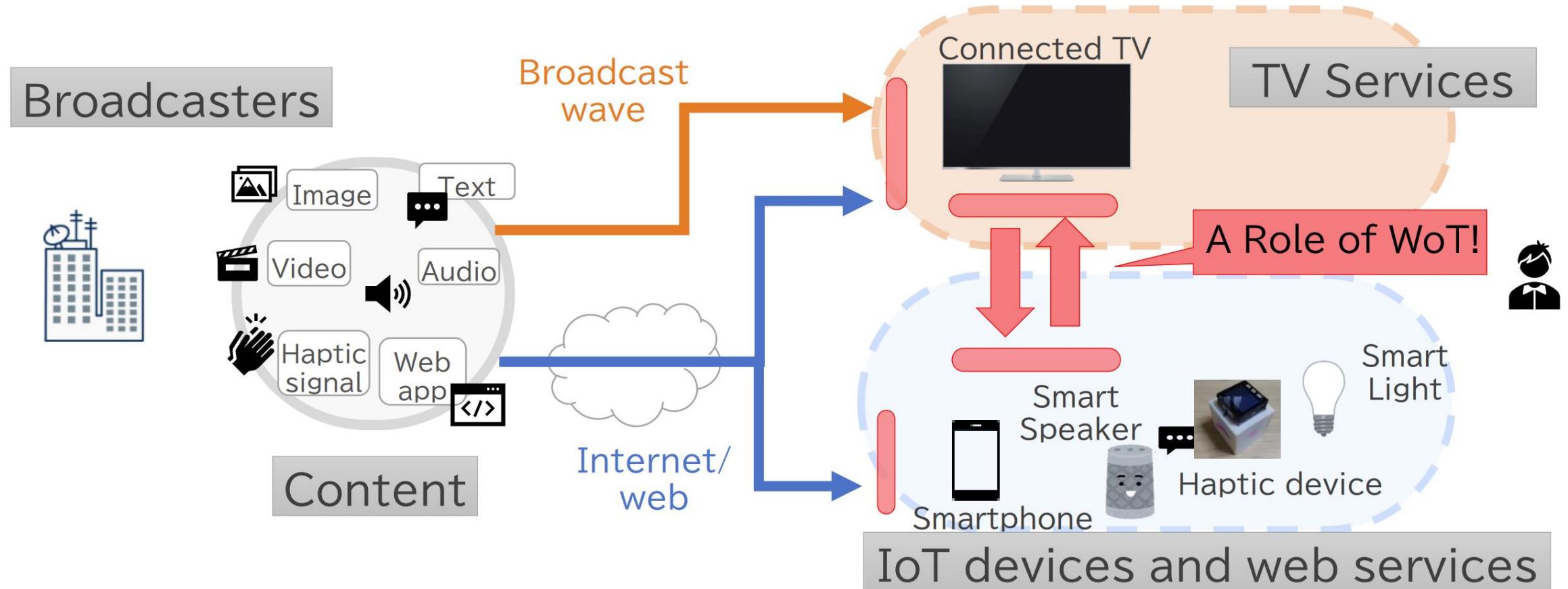
1. Must manage many devices and tons of associated measurement and control points.
2. It is impossible to write a TD for each device manually.
3. We must extend the original WoT API for batch property acquisition.

Visual graph ⓘ



NHK Hybridcast

- Broadcasters will deliver content with IoT devices.
- We expect WoT to connect things across TV and Web worlds.



NHK Hybridcast



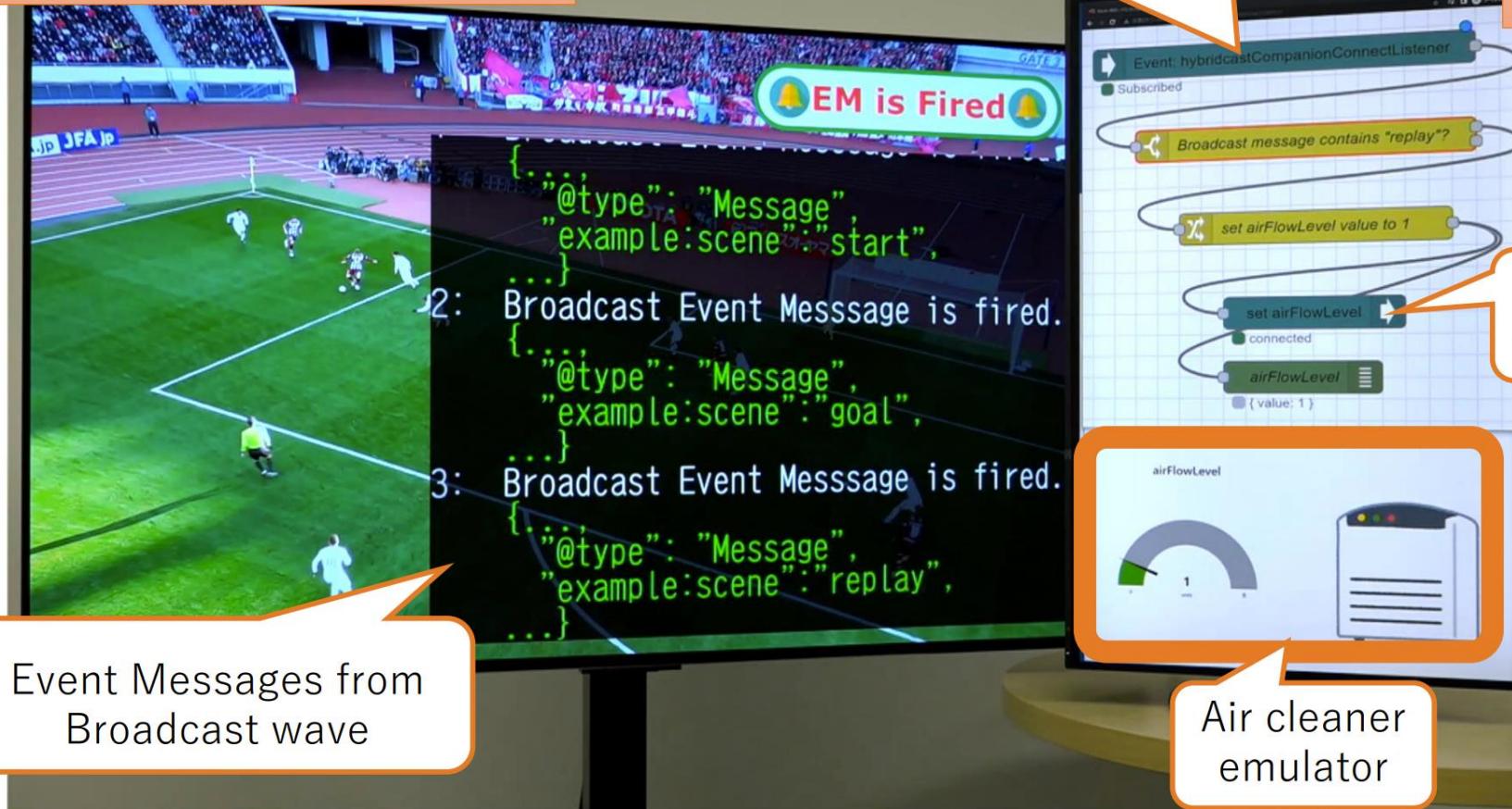
- A TV plays a soccer program.
- An air cleaner is noisy in turbo mode.
- Users want quietness in exciting scenes.
- TV can receive Event Messages related to a program via broadcasting wave.

NHK Hybridcast

Hybridcast Connect TV
(LG OLED42C2PJA)

WoT node to listen
Hybridcast Connect Event

Node-RED
App



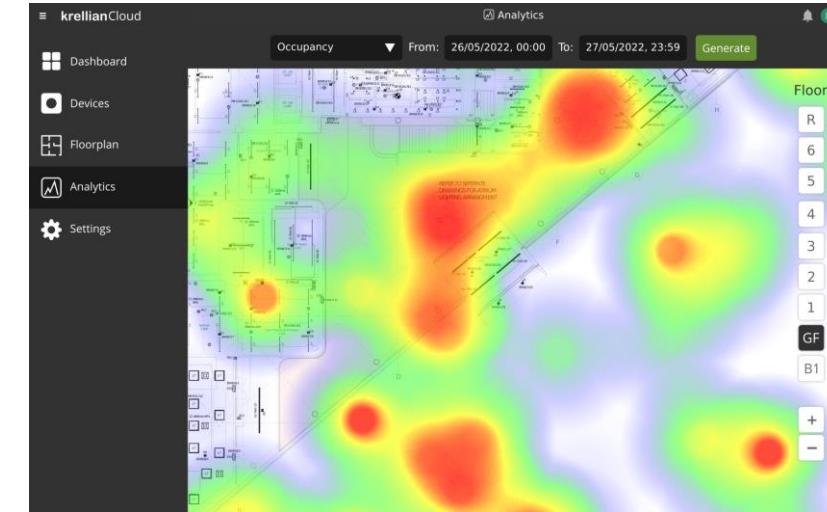
Krellian

Krellian provides a commercial smart building platform, built on the web of Things.



krellianHub

Consolidates multi-vendor building management systems into a single standardised interface.



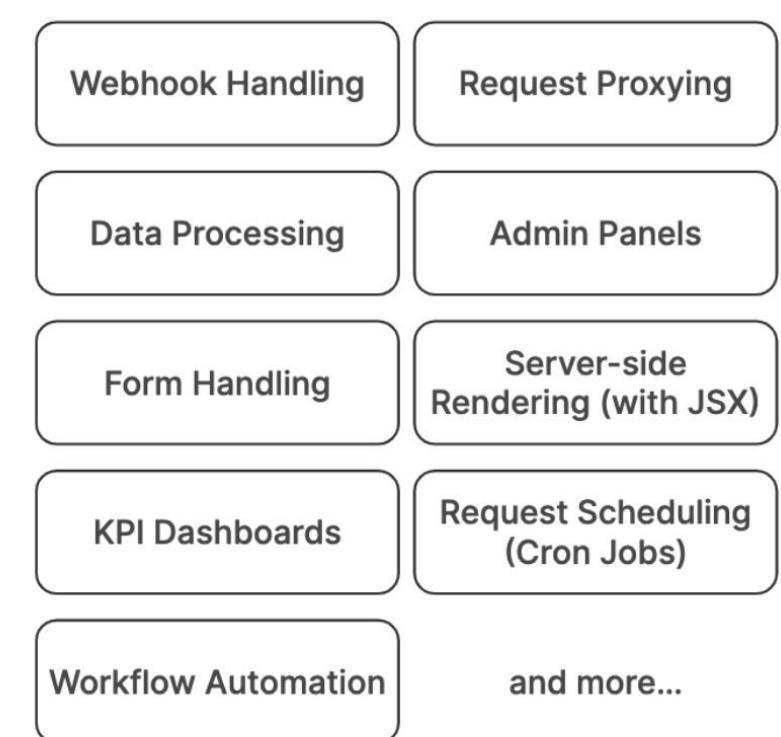
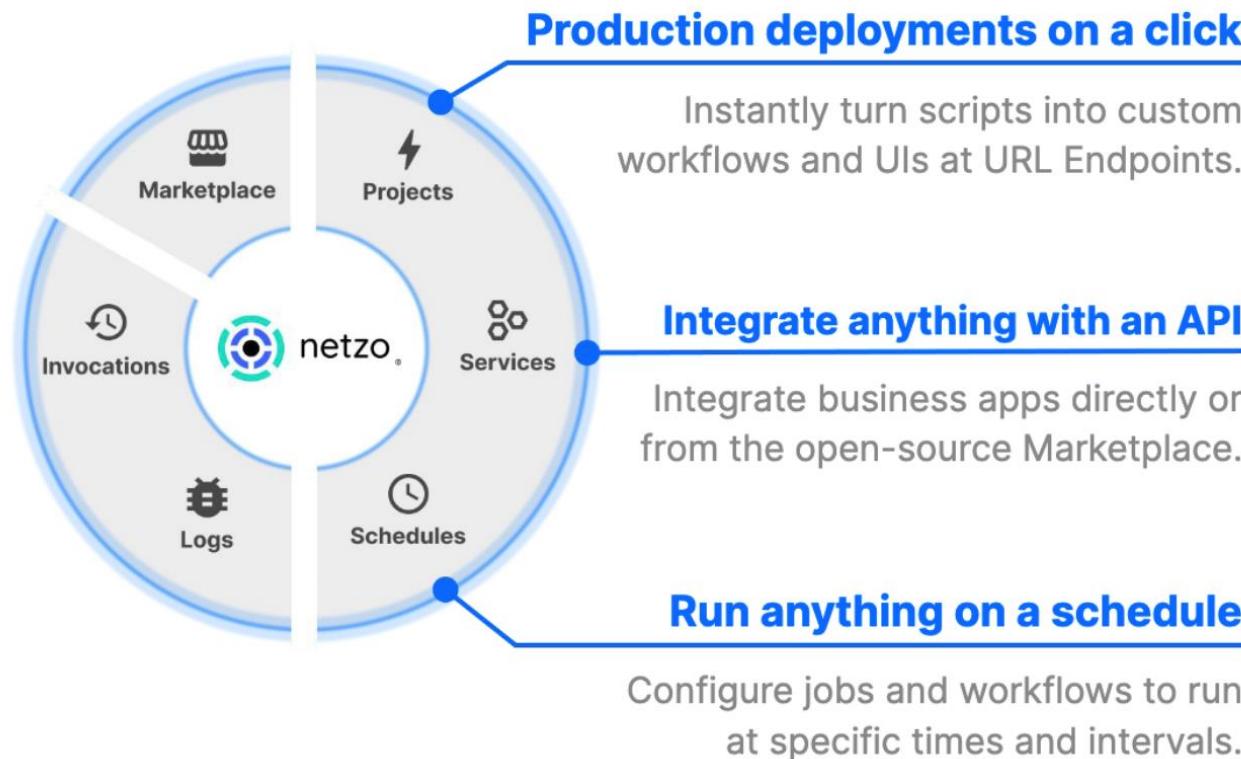
krellianCloud

Creates a digital twin of your building to model how it's being used and identify potential optimisations.

krellian.com

Netzo

Deploy JavaScript and TypeScript to URL endpoints instantly. Connect essential APIs, automate processes and build tools faster, without managing infrastructure.



Netzo

Marketplace

Marketplace items are ready-made solutions for specific use-cases. Anything you find here is public and you are free to fork into any or your Workspaces. When you fork an items, a copy is created so you can go ahead and make changes. Forked items will count normally for your Workspace usage.

TYPE

- All
- Services
- Projects

STATUS

- All
- Stable
- Beta
- Alpha
- Requested
- Deprecated

CATEGORY

- All
- Core
- Community
- Enterprise

Filter by name

Aa Request Contribute

 ActiveCampaign <small>service</small> Service for the ActiveCampaign... <small>crm marketing</small>	 Bar Chart with Billboar... <small>project</small> An HTTP server that serves a ... <small>example chart</small>	 Bigin <small>service</small> Service for the Bigin API <small>crm</small>	 Clarifai <small>service</small> Service for the Clarifai API <small>artificial-intelligence machine-learning</small>	 Cloudflare <small>service</small> Service for the Cloudflare API <small>infrastructure</small>	 Discord <small>service</small> Service for the Discord API <small>social communications</small>
 Enode <small>service</small> Service for the Enode API <small>iot energy solar</small>	 Enphase <small>service</small> Service for the Enphase API <small>iot energy solar</small>	 Ergast F1 <small>service</small> Service for the Ergast F1 API <small>mock</small>	 Facturama <small>service</small> Service for the Facturama API <small>billing</small>	 Fathom Analytics <small>service</small> Service for the Fathom Analyti... <small>analytics</small>	 FaunaDB <small>service</small> Service for the FaunaDB Graph... <small>database</small>
 Get Client IP Address <small>project</small> An HTTP server that responds ... <small>example json</small>	 GitHub <small>service</small> Service for the GitHub API <small>infrastructure</small>	 Google AppSheet <small>service</small> Service for the Google AppShe... <small>productivity</small>	 Google Sheets <small>service</small> Service for the Google Sheets ... <small>productivity google</small>	 Handling Form Submis... <small>project</small> An HTTP server that serves a ... <small>example form json</small>	 Hashnode <small>service</small> Service for the Hashnode API <small>social blog</small>

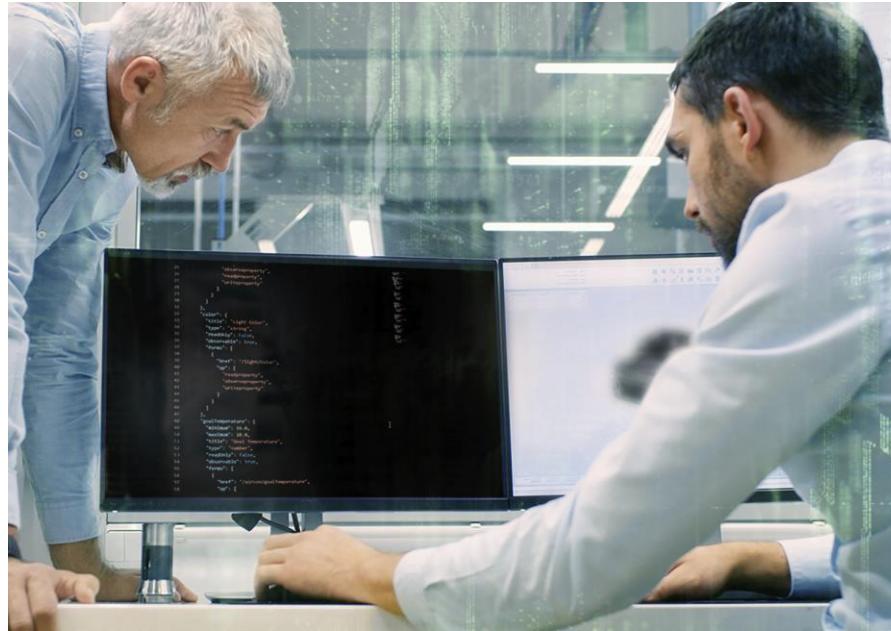
Siemens – Desigo CC

Building Information Management (BIM)

- View status in context of 3D model
- Security
- Energy Efficiency
- Maintenance management



Siemens/evosoft – SayWoT!



SayWoT!

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

Thing Models, Thing Descriptions, Protocol adapters...



Wunsiedel H₂ Generation Plant

<https://www.evosoft.com/en/application-of-the-w3c-web-of-things-standard-in-the-wunsiedel-hydrogen-generation-plant/>

Application of SayWoT to cloud integration

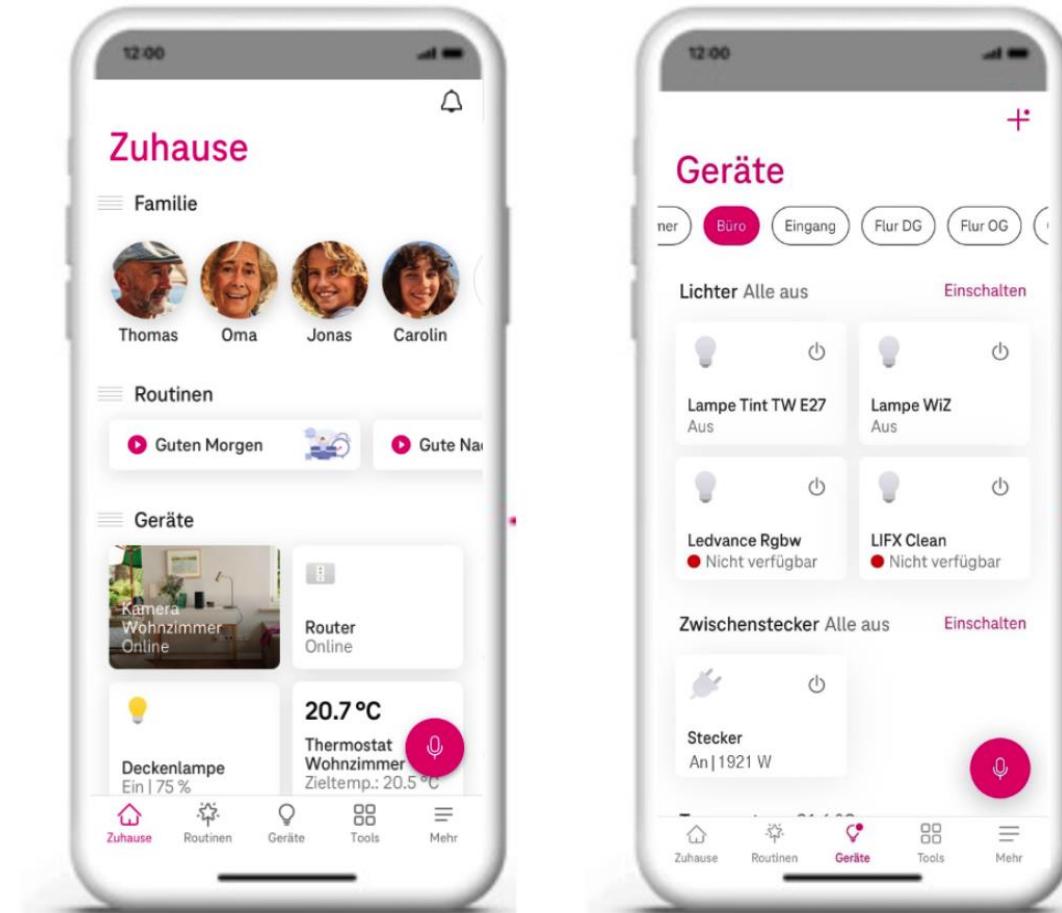
Deutsch Telekom - MagentaZuhause



Good & safe family life
Take care of your loved ones and manage everyday family coordination with ease.



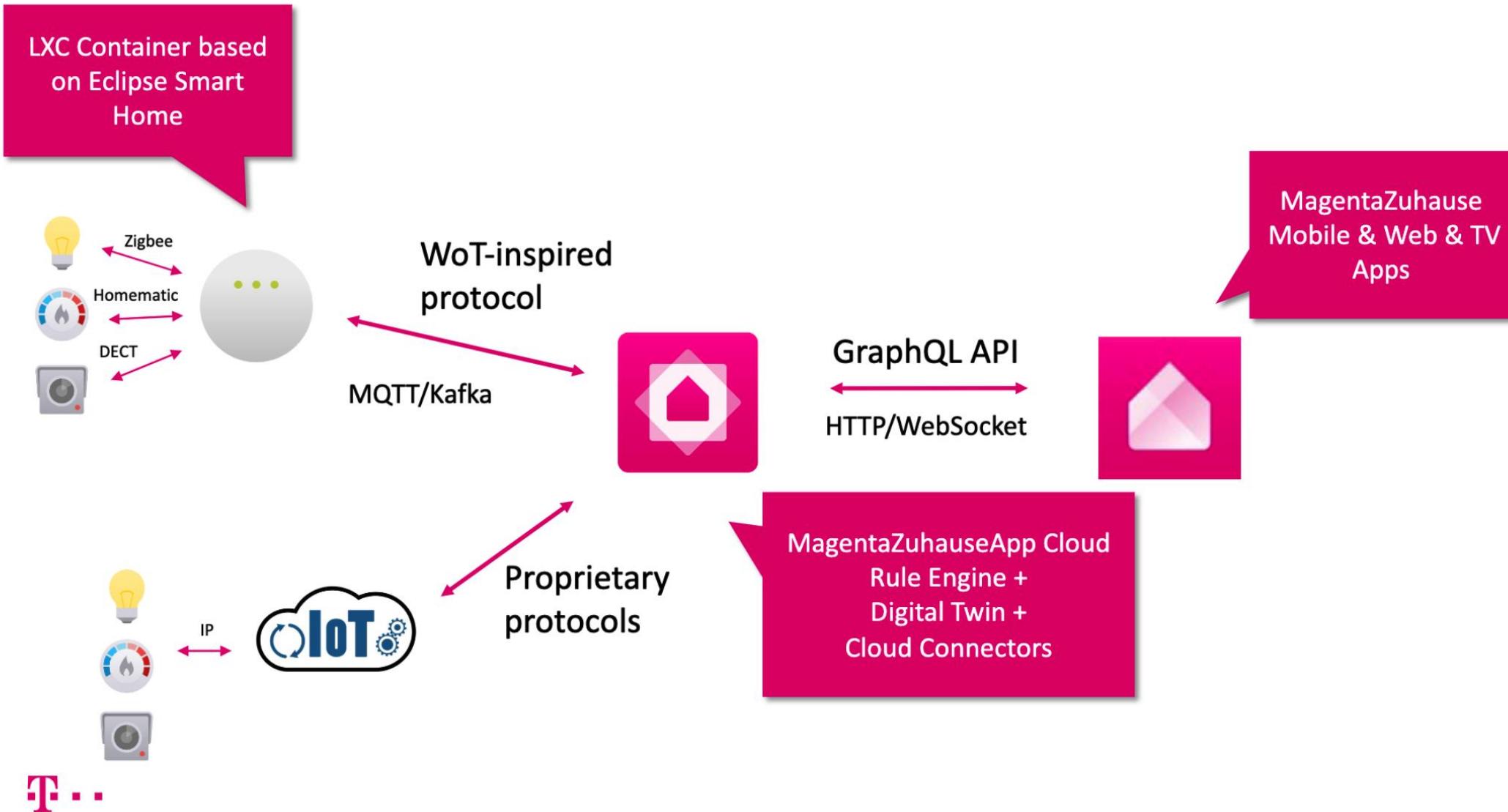
Smarter home
Gain security and comfort for your home.

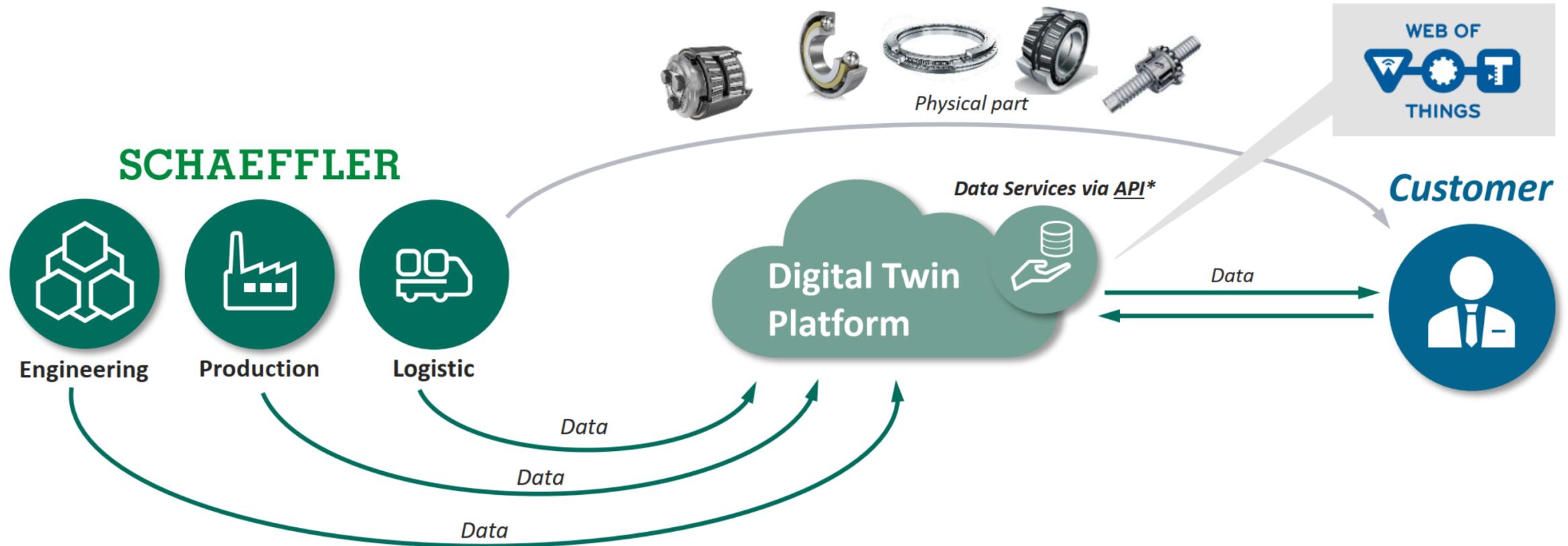


4

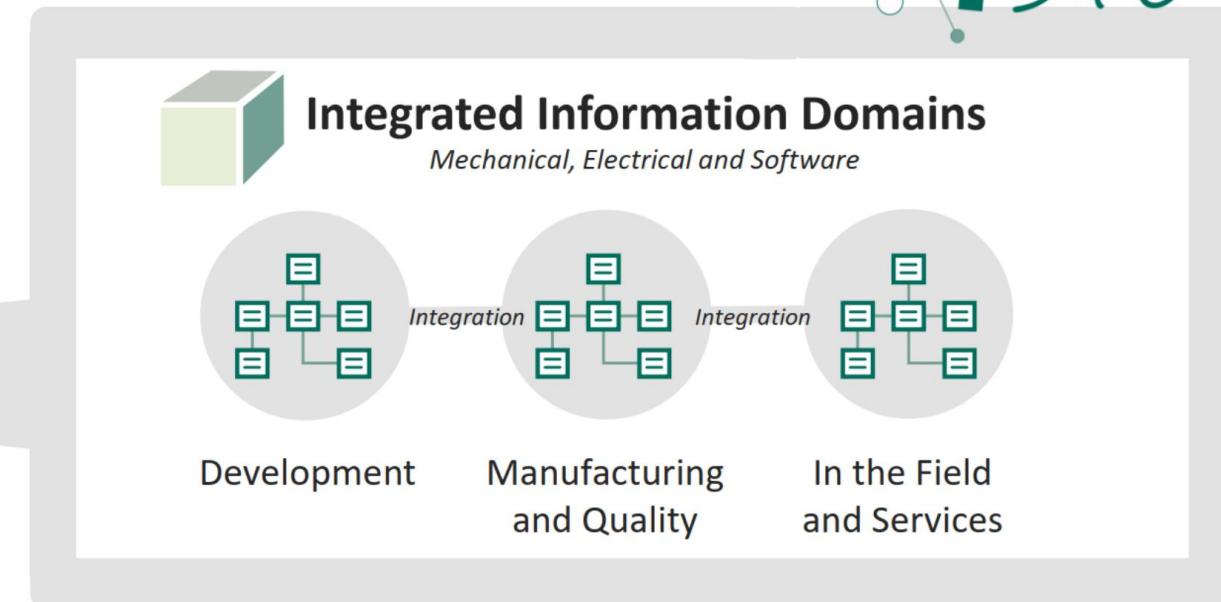
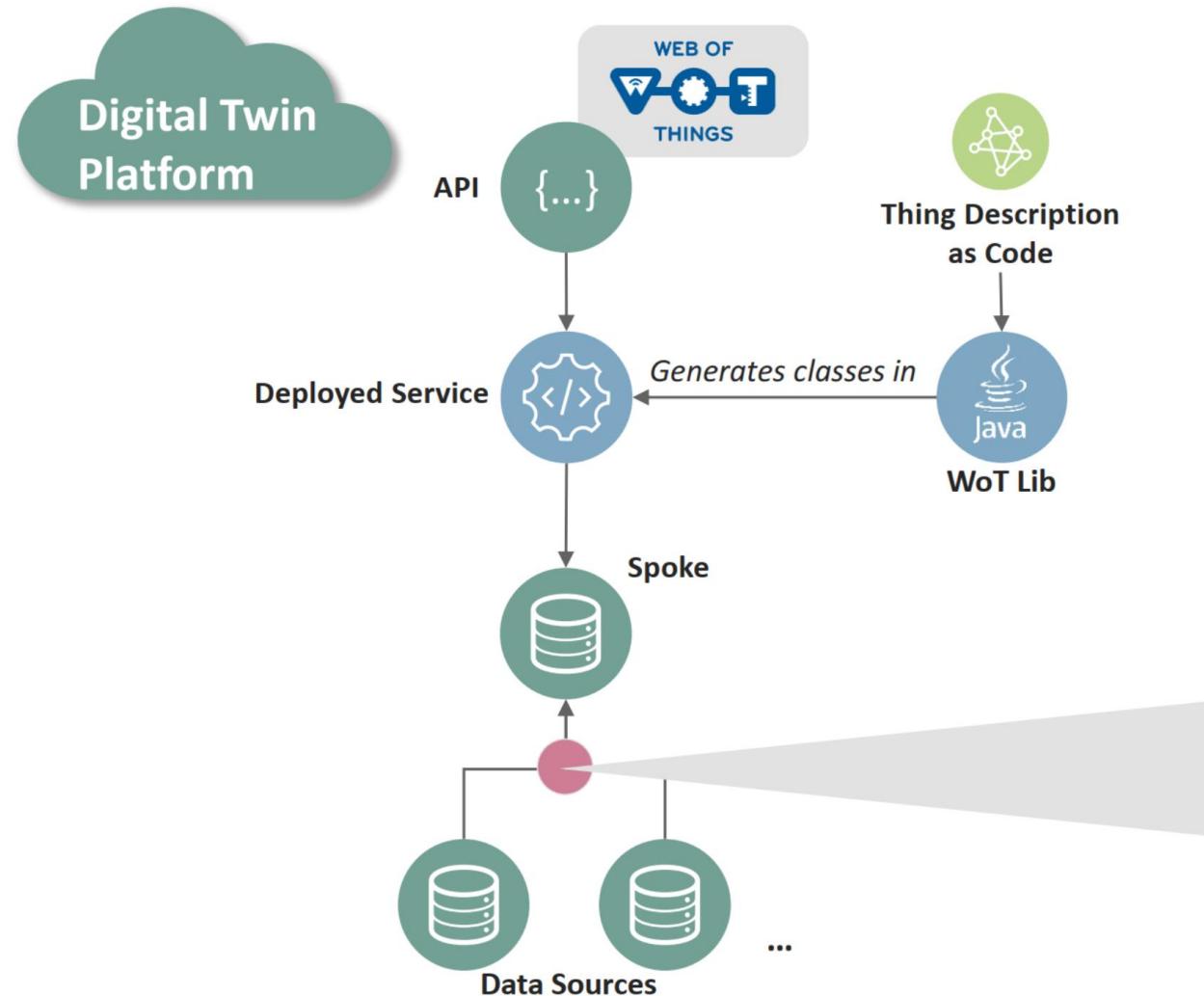
32

Deutsch Telekom - MagentaZuhause





Schaeffler



Open Source Projects



WebThings

<https://webthings.io/>

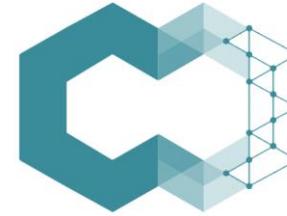
- Smart Home Hub
- Reusable Software Components



THINGWEB

<https://www.thingweb.io/>

- node-wot Scripting API
- W3C WoT Reference Implementation
- Validation tools



ditto

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

- Eclipse (Bosch, Siemens, ...)
- Digital twin

edi{TD}or

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

- TD Authoring



<https://www.sifis-home.eu/>

- Horizon 2020 Project
- Trustworthy Connected Home

See also:

<https://www.w3.org/WoT/developers/>

ThingWeb/node-wot

<https://www.thingweb.io/>

- Official reference implementation for W3C WoT interaction model
- Follows the WoT Scripting API specification
- Thing Description parser and serializer, protocol bindings, runtime system for scripting applications, based on Node.js/TypeScript
- Validation tools (TD Playground)

```

WoTHelpers.fetch( "coap://localhost:5683/counter" ).then( async (td) => {
  // using await for serial execution (note 'async' in then() of fetch())
  try {
    const thing = await WoT.consume(td);
    console.info( "==== TD ===" );
    console.info(td);
    console.info( "===========" );

    // read property #1
    const read1 = await thing.readProperty( "count" );
    console.info( "count value is" , await read1.value());

    // increment property #1 (without step)
    await thing.invokeAction( "increment" );
    const inc1 = await thing.readProperty( "count" );
    console.info( "count value after increment #1 is" , await inc1.value());

    // increment property #2 (with step)
    await thing.invokeAction( "increment" , {step: 3});
    const inc2 = await thing.readProperty( "count" );
    console.info( "count value after increment #2 (w/ step 3) is" , await inc2.value());

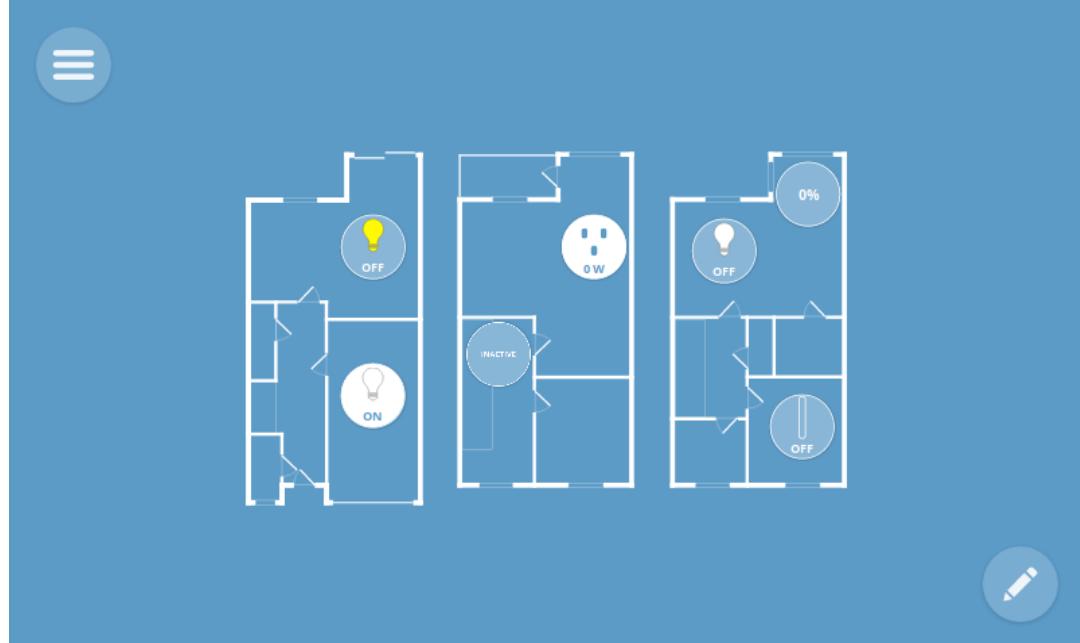
    // decrement property
    await thing.invokeAction( "decrement" , undefined, {
      formIndex: getFormIndexForDecrementWithCoAP(thing);
    });
    const dec1 = await thing.readProperty( "count" );
    console.info( "count value after decrement is" , await dec1.value());

  } catch(err) {
    console.error( "Script error:", err);
  }
}).catch( (err) => { console.error( "Fetch error:", err); });

```

WebThings

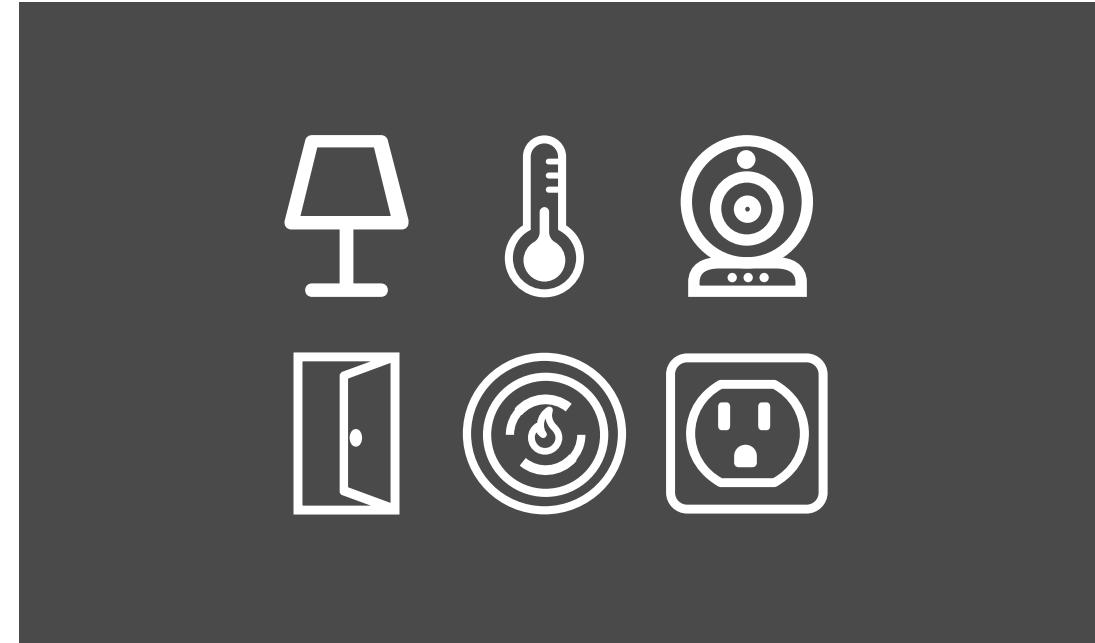
WebThings is an open platform for monitoring and controlling devices over the web.



WebThings Gateway

A software distribution for smart home hubs,
focused on privacy, security and interoperability.

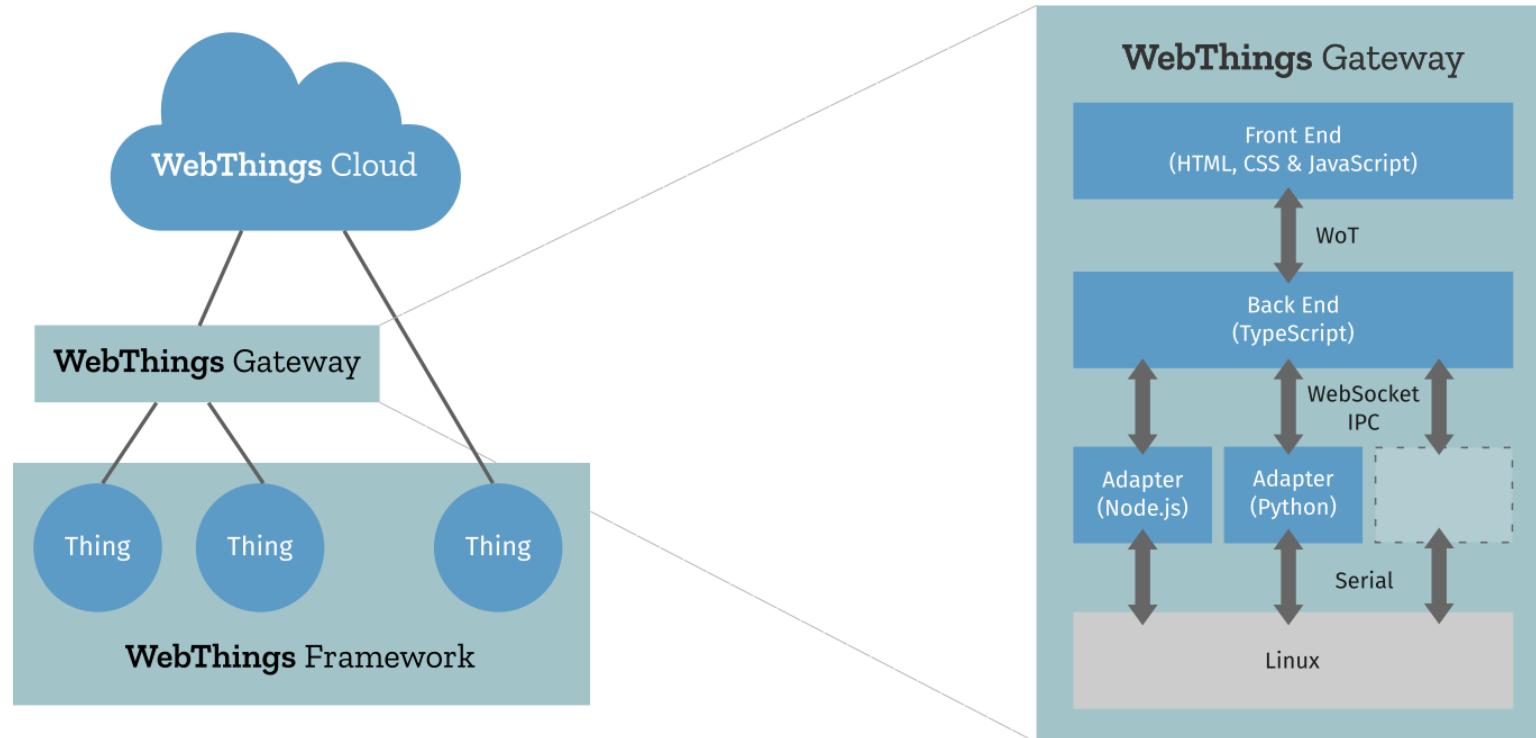
webthings.io



WebThings Framework

A collection of re-usable software components to help
developers build their own web things

WebThings



WoT Thing Description 1.1

✓ IMPLEMENTED

WoT Discovery 1.0

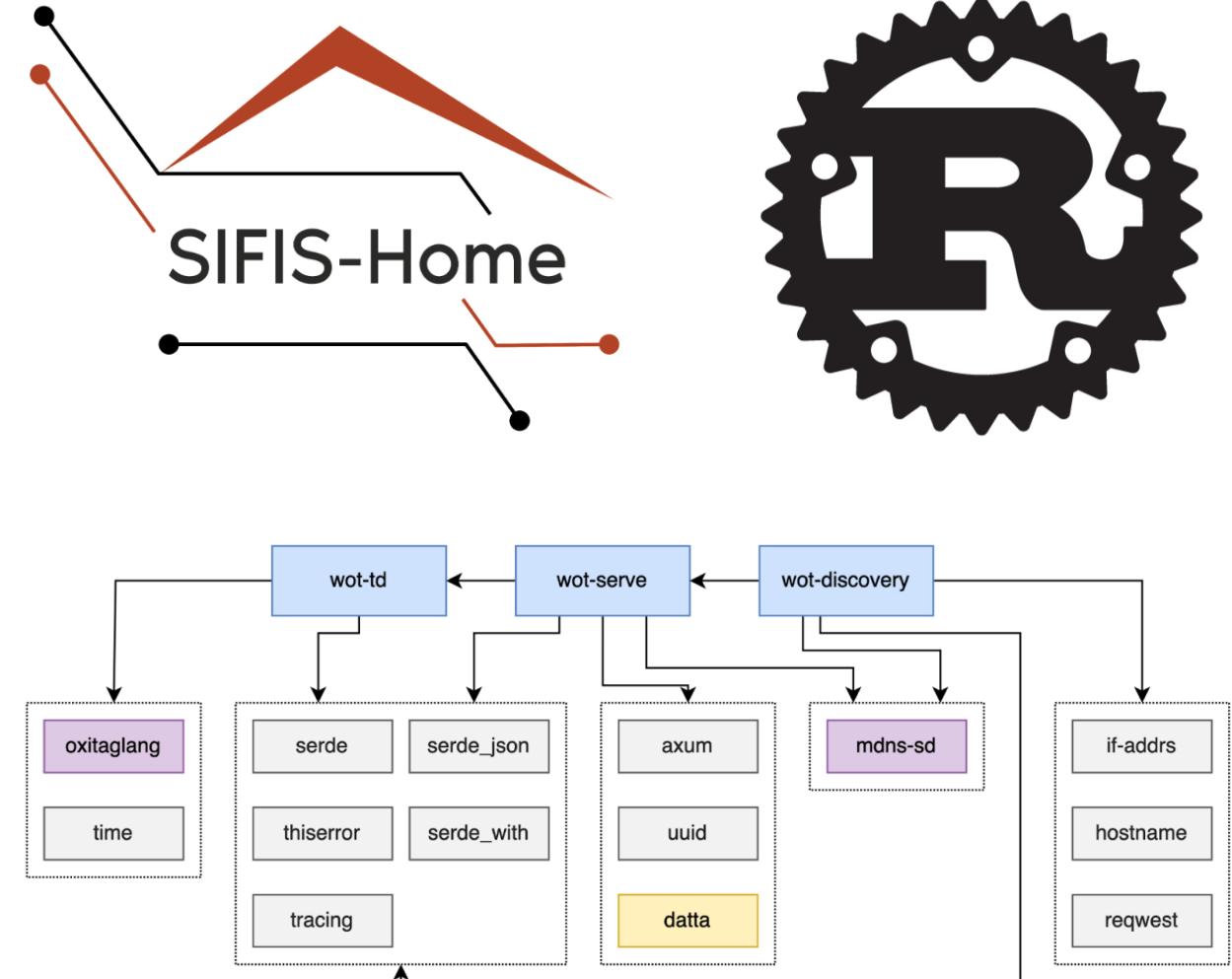
○ IN PROGRESS

WoT Profile 1.0

○ IN PROGRESS

SIFIS Home

- Horizon 2020 Project
 - <https://www.sifis-home.eu/>
- Secure and trustworthy full-stack internet of things for Smart Home
- Open Standards
- Using Rust to avoid memory bugs
- wot-rust implementation



SIFIS Home

```
.property("on", |b| {
    b.finish_extend_data_schema()
    .attype("OnOffProperty")
    .title("On/Off")
    .description("Whether the lamp is turned on")
    .form(|b| {
        b.href("/properties/on")
        .http_get(get_on_property)
        .http_put(put_on_property)
        .op(wot_td::thing::FormOperation::ReadProperty)
        .op(wot_td::thing::FormOperation::WriteProperty)
    })
    .bool()
})
.property("brightness", |b| {
    b.finish_extend_data_schema()
    .attype("BrightnessProperty")
    .title("Brightness")
    .description("The level of light from 0-100")
    .form(|b| {
        b.href("/properties/brightness")
        .http_get(get_brightness_property)
        .http_put(put_brightness_property)
        .op(wot_td::thing::FormOperation::ReadProperty)
        .op(wot_td::thing::FormOperation::WriteProperty)
    })
    .integer()
    .minimum(0)
})
```

```
"properties": {
    "brightness": {
        "@type": "BrightnessProperty",
        "description": "The level of light from 0-100",
        "forms": [
            {
                "href": "/properties/brightness",
                "op": [
                    "readproperty",
                    "writeproperty"
                ]
            }
        ],
        "maximum": 100,
        "minimum": 0,
        "readOnly": false,
        "title": "Brightness",
        "type": "integer",
        "unit": "percent",
        "writeOnly": false
    },
    "on": {
        "@type": "OnOffProperty",
        "description": "Whether the lamp is turned on",
        "forms": [
            {
                "href": "/properties/on",
                "op": [
                    "readproperty",
                    "writeproperty"
                ]
            }
        ]
    }
},
```

```
Running `target/debug/lamp`
2022-09-19T16:39:22.278330Z DEBUG mdns_sd::service_daemon: new socket bind to 0.0.0.0:5353
2022-09-19T16:39:22.278714Z DEBUG lamp: listening on 0.0.0.0:3000
2022-09-19T16:39:22.299215Z DEBUG mdns_sd::service_daemon: register service ServiceInfo { ty_domain: "_wot._tcp.local.", sub_domain: None, fullname: "mybf2fafef3139540509dbecfb1207666c._wot._tcp.local.", server: "enyo.lan.local", addresses: [192.168.1.212], port: 3000, host_ttl: 120, other_ttl: 4500, priority: 0, weight: 0, properties: {"td": "/.well-known/wot", "type": "Thing"}, last_update: 1663605562278 }
2022-09-19T16:39:22.299344Z DEBUG mdns_sd::service_daemon: broadcast service mybf2fafef3139540509dbecfb1207666c._wot._tcp.local.
```

Timestamp	A/R	Flags	if	Domain	Service Type	Instance Name
18:40:34.086	Add	3	6	local.	_wot..tcp.	mybd729b6943214caeb527e15626162967
18:40:34.086	Add	2	6	local.	_wot..tcp.	mybf2fafef3139540509dbecfb1207666c

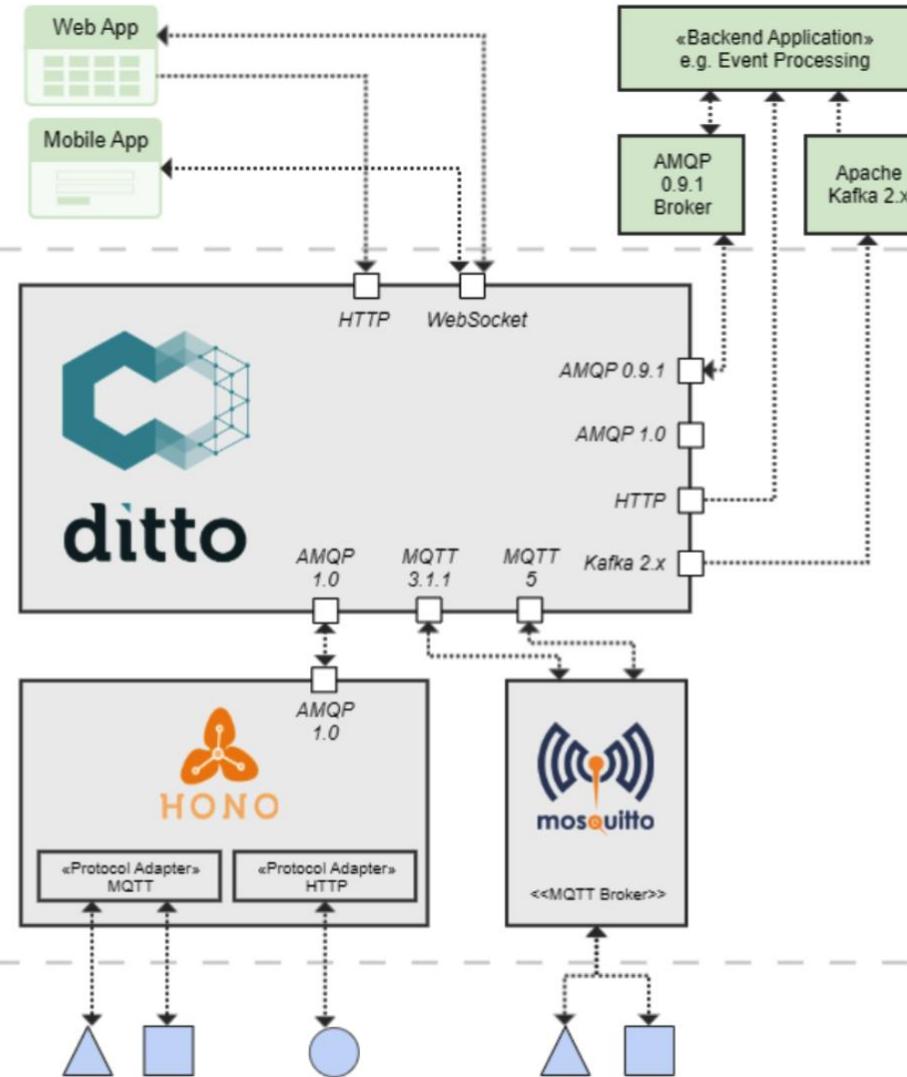
Ditto

IOT
SOLUTION

DEVICE
TWINS

DEVICE
CONNECTIVITY

IOT
DEVICES



Ditto as
Digital Twin
"middleware"

Ditto

- Ditto downloads and caches referenced TMs during runtime
- resolving extensions via `tm:extends` and imports via `tm:ref`, Thing level compositions via `tm:submodel` and TM placeholders
- Ditto generates TDs, injecting forms with HTTP API endpoints

```
curl -u ditto:ditto \
  'https://ditto.eclipseprojects.io/api/2/things/io.eclipseprojects.ditto:floor-lamp-0815' \
  -H 'Accept: application/td+json'

{
  "@context": [
    "https://www.w3.org/2022/wot/td/v1.1",
    ...
  ],
  "title": "Floor Lamp",
  "@type": "Thing",
  "id": "urn:io.eclipseprojects.ditto:floor-lamp-0815",
  "base": "https://ditto.eclipseprojects.io/api/2/things/io.eclipseprojects.ditto:floor-lamp-0815",
  "version": {
    "model": "1.0.0",
    "instance": "1.0.0"
  },
  "links": [
    {
      "rel": "type",
      "href": "https://eclipse.github.io/ditto-examples/wot/models/floor-lamp-1.0.0.tm.jsonld",
      "type": "application/tm+json"
    },
    {
      "rel": "item",
      "type": "application/td+json",
      "href": "/features/Spot1"
    },
    ...
  ],
  "security": "basic_sc",
  "securityDefinitions": {
    "basic_sc": {
      "in": "header",
      "scheme": "basic"
    }
  },
  "forms": [
    {
      "op": "readAllProperties",
      "href": "/attributes{?channel,timeout}",
      "htv:methodName": "GET",
      "contentType": "application/json"
    }
  ]
}
```

EdiTDoor

The screenshot shows the EdiTDoor interface for creating a Thing Description (TD). The top navigation bar includes the logo "WEB OF THINGS", the title "editTDor", and buttons for "New TD", "Open TD", "Save", and "Save As".

simulated hotel room
[id: urn:HotelRoom, base: "coap://localhost:3000"]
This is a simulated hotel room with CoAP devices.

Properties

Light brightness
[id: brightness, type: integer, min: 0, max: 100]
Brightness is a value between 0 and 100 that shows the dim level of the hotelrooms lamp.

Read /light/Brightness
Write /light/Brightness
Observe /light/Brightness

Smoke
[id: smoke, type: boolean]
Smoke returns a boolean value. True if smoke was detected, false if not.

Read /smoke
Write /smoke
Observe /smoke

Actions

Turn off light
[id: turn-off-light]
Invoking this action turns off the light in the simulated hotel room.

Invoke /path/to/endpoint

The left sidebar displays the full JSON-LD code for the TD:

```
1 {
2   "@context": "https://www.w3.org/2019/wo/t/td/v1",
3   "id": "urn:HotelRoomOK",
4   "securityDefinitions": {
5     "nosec_sc": {
6       "scheme": "nosec"
7     }
8   },
9   "description": "This is a simulated hotel room with CoAP devices.",
10  "security": "nosec_sc",
11  "@type": "Thing",
12  "base": "coap://localhost:3000",
13  "title": "simulated hotel room",
14  "properties": {
15    "brightness": {
16      "minimum": 0,
17      "maximum": 100,
18      "type": "Integer",
19      "title": "Light Brightness",
20      "description": "Brightness is a value between 0 and 100 that ...",
21      "readOnly": false,
22      "observable": true,
23      "forms": [
24        {
25          "href": "/light/Brightness",
26          "ep": [
27            "observeproperty",
28            "readproperty",
29            "writeproperty"
30          ]
31        }
32      ],
33    },
34  },
35 }
36
37
38
39
40
41
42
.
.
.
```

TD Authoring Tool

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

Other Engagements



<https://echonet.jp/english/>

- Smart Home / Smart City
- Echonet Lite / WoT TD Integration



<https://www.conexxus.org/>

- Convenience Stores
- Data Models and Services

Protocols:

- MQTT
- CoAP
- OCF
- OPC UA
- MODBUS

Groups

- IETF/ITRF T2TRG
- Open Geospatial Consortium
- IEC CDD/ECLASS
- OneDM
- IEEE
- ITU-T
- Microsoft DTDL

Discussion: Open Problems/Next Steps

- GIS Integration
 - Geospatial data and discovery
- Data Management
 - Digital Twins and shadows
 - Event and action modelling
 - Data management
- Security
 - Key provisioning and onboarding
 - Secure LAN access
 - Proxy services
 - Access control and ad-hoc sharing
 - MUDS
- Accessibility
 - Sensory modality mapping
 - Textual/descriptive interfaces
 - Service location
 - Mobility services
- Advanced Use Cases
 - Transportation
 - Logistics
 - Distributed energy management
 - AR visualization
 - Analytics integration e.g. for health and safety monitoring

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