

WoT Status Update

February 2020

W3C Web of Things

Goal: Support IoT Interoperability via Open Standards

- **W3C WoT Interest Group (IG)**

<https://www.w3.org/2016/07/wot-ig-charter.html>

- Started spring 2015
- ~200 participants
- Informal work and outreach
- “PlugFest” validation with running code
- Exploration of new building blocks
- “OpenDays” with external speakers
- Liaisons and collaborations with other organizations and SDOs
- ***Second Workshop on Web of Things held 3-5 June 2019 in Munich***
- ***IG charter renewal October 2019***

- **W3C WoT Working Group (WG)**

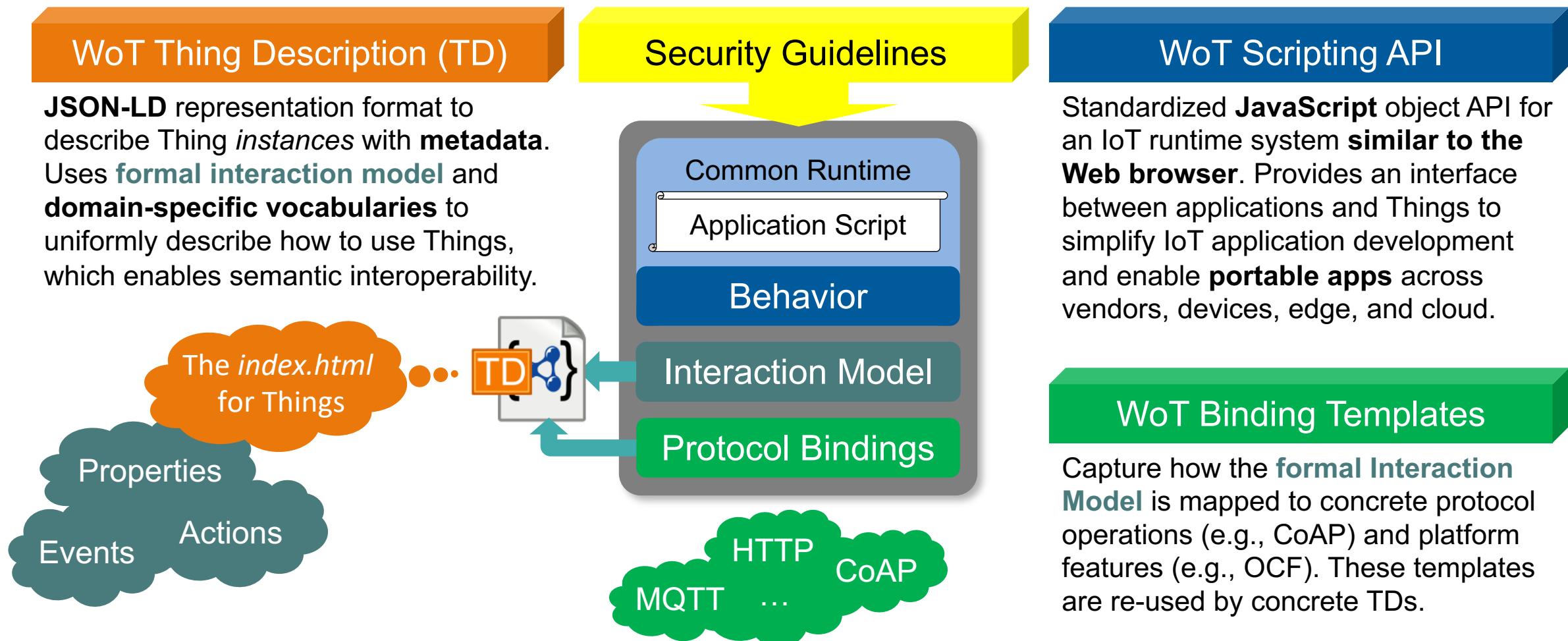
<https://www.w3.org/2016/12/wot-wg-2016.html>

- Started end of 2016 (effectively Feb 2017)
- ~100 participants
- Normative work on specific deliverables
- W3C Patent Policy for royalty-free standards
- Only W3C Members and Invited Experts
- ***Architecture and Thing Description have both transitioned to Proposed Recommendations in late January 2020***
- ***Notes published on Protocol Bindings, Security, and Scripting API***
- ***WG charter renewed January 2020***
- ***Transition to W3C Recommendations expected in March 2020***

W3C Web of Things – Building Blocks

WoT Architecture

Overarching umbrella with architectural constraints and guidance on how to use and combine building blocks.



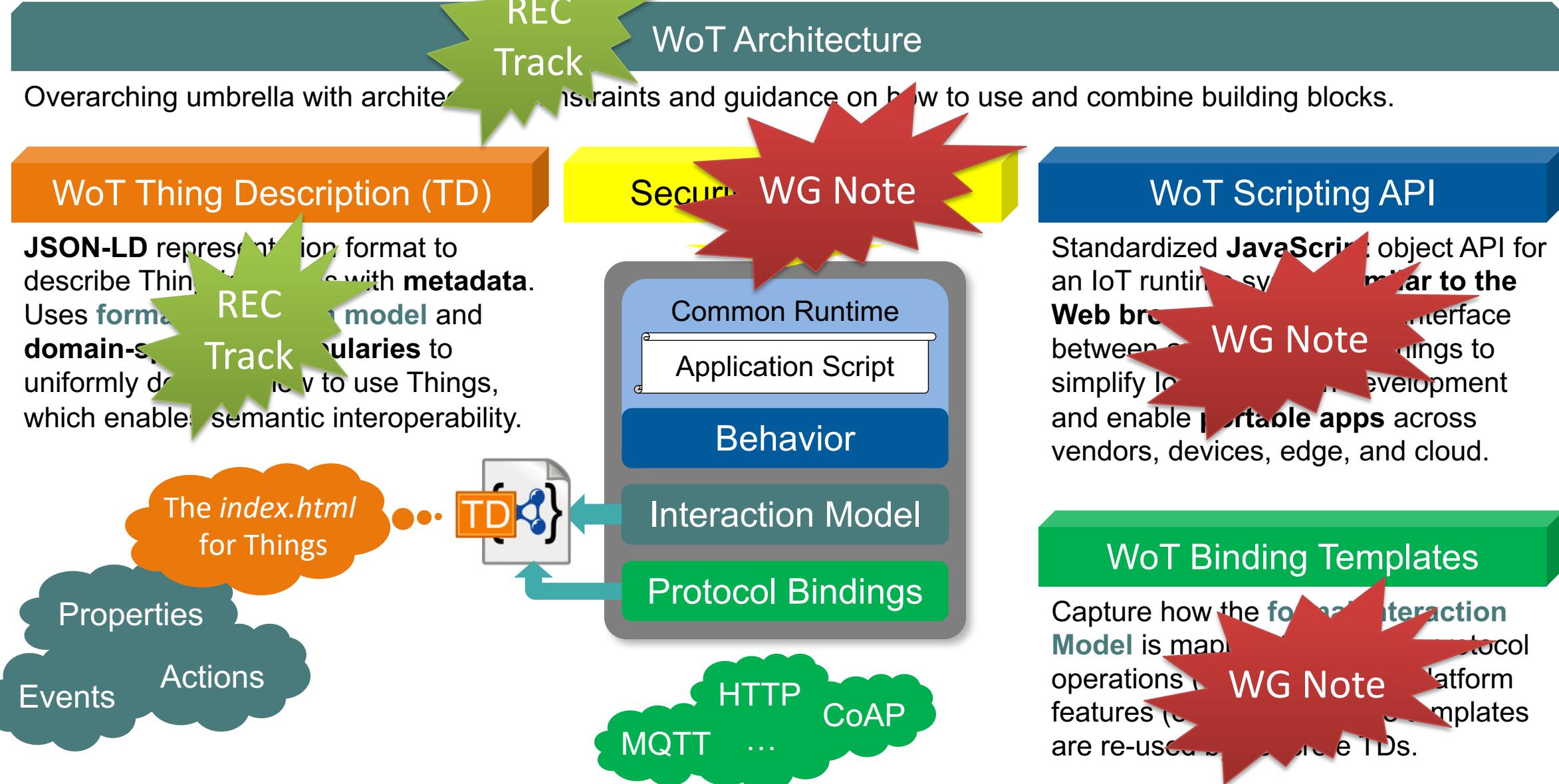
WoT Scripting API

Standardized **JavaScript** object API for an IoT runtime system **similar to the Web browser**. Provides an interface between applications and Things to simplify IoT application development and enable **portable apps** across vendors, devices, edge, and cloud.

WoT Binding Templates

Capture how the **formal Interaction Model** is mapped to concrete protocol operations (e.g., CoAP) and platform features (e.g., OCF). These templates are re-used by concrete TDs.

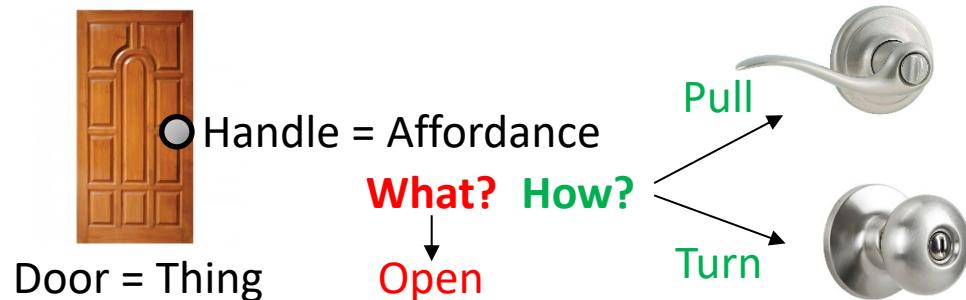
W3C Web of Things – Building Blocks



Published Proposed Recommendations

WoT Architecture

- Constraints
 - Things must have TD (W3C WoT)
 - Must use hypermedia controls (general WoT)
 - URIs
 - Standard set of methods
 - Media Types
- Interaction Affordances
 - Metadata of a Thing that shows and describes the possible choices (**what**) to Consumers, thereby suggesting **how** Consumers may interact with the Thing



WoT Thing Description (TD)

```
{
    "@context": [
        "https://www.w3.org/2019/wot/td/v1",
        { "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": {
            ...
        }
    }
}
```

Published WG Notes

- **WoT Security and Privacy Guidelines**
 - Details beyond the security considerations in each specification for a holistic security and privacy configuration of Things
 - Security testing plan

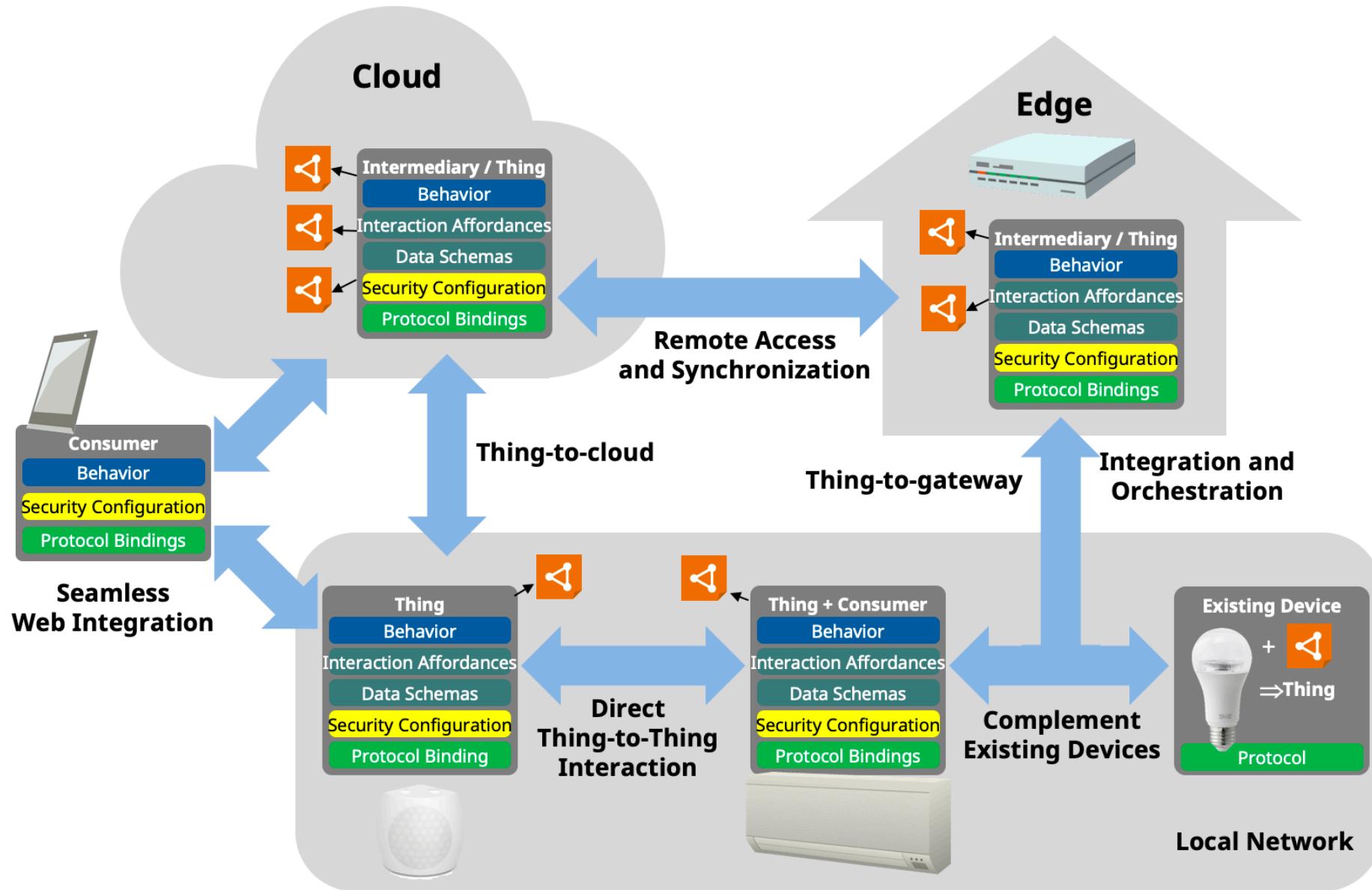
- **WoT Binding Templates**
 - Documentation for how to describe existing IoT ecosystems (e.g., OCF or generic Web) with WoT Thing Description

- **WoT Scripting API**
 - Proposal for a standard API to consume and produce WoT Thing Descriptions
 - Provides interface between applications and network-facing API of IoT devices (cf. Web browser APIs)

Status and Key Design Decisions

- Decision to adopt JSON-LD 1.1 proposed features to allow:
 - Default values
 - Object notation (name: value) instead of arrays
 - Alignment with common JSON practices
- Security metadata
 - Focus on HTTPS (Basic Auth, Digest, Bearer Tokens, PSK, OAuth2)
- Protocol Bindings
 - Focus on HTTP and structured payloads compatible with JSON
 - Support for Events using subprotocols (e.g., long polling in HTTP)
- Extension Points
 - CoAP(S), MQTT(S), and further security schemes
 - Semantic annotations with custom vocabularies (JSON-LD @context and @type)

Use Case Overview



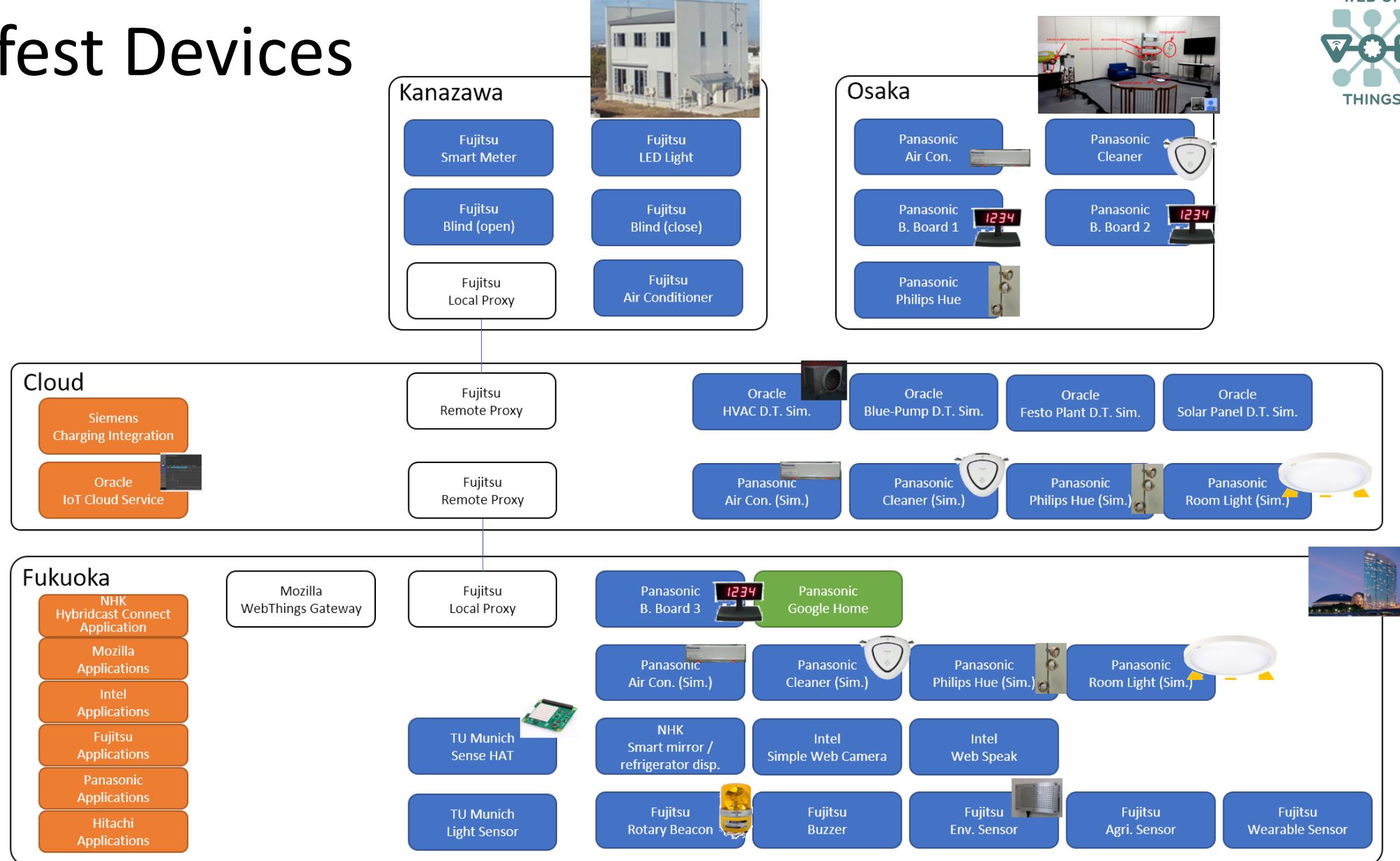
WoT Workshop: Munich 2019



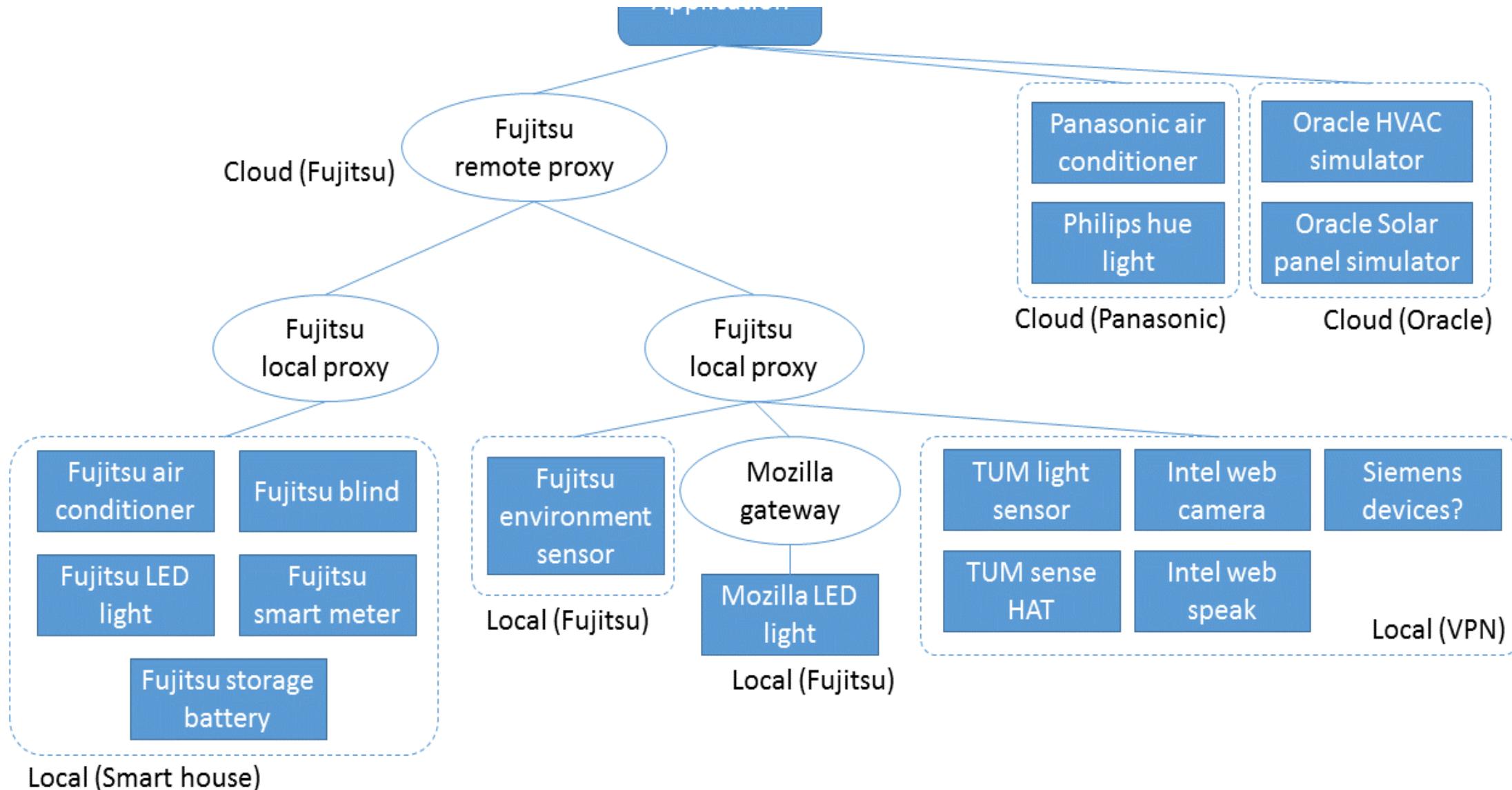
TPAC 2019 Plugfest



Plugfest Devices



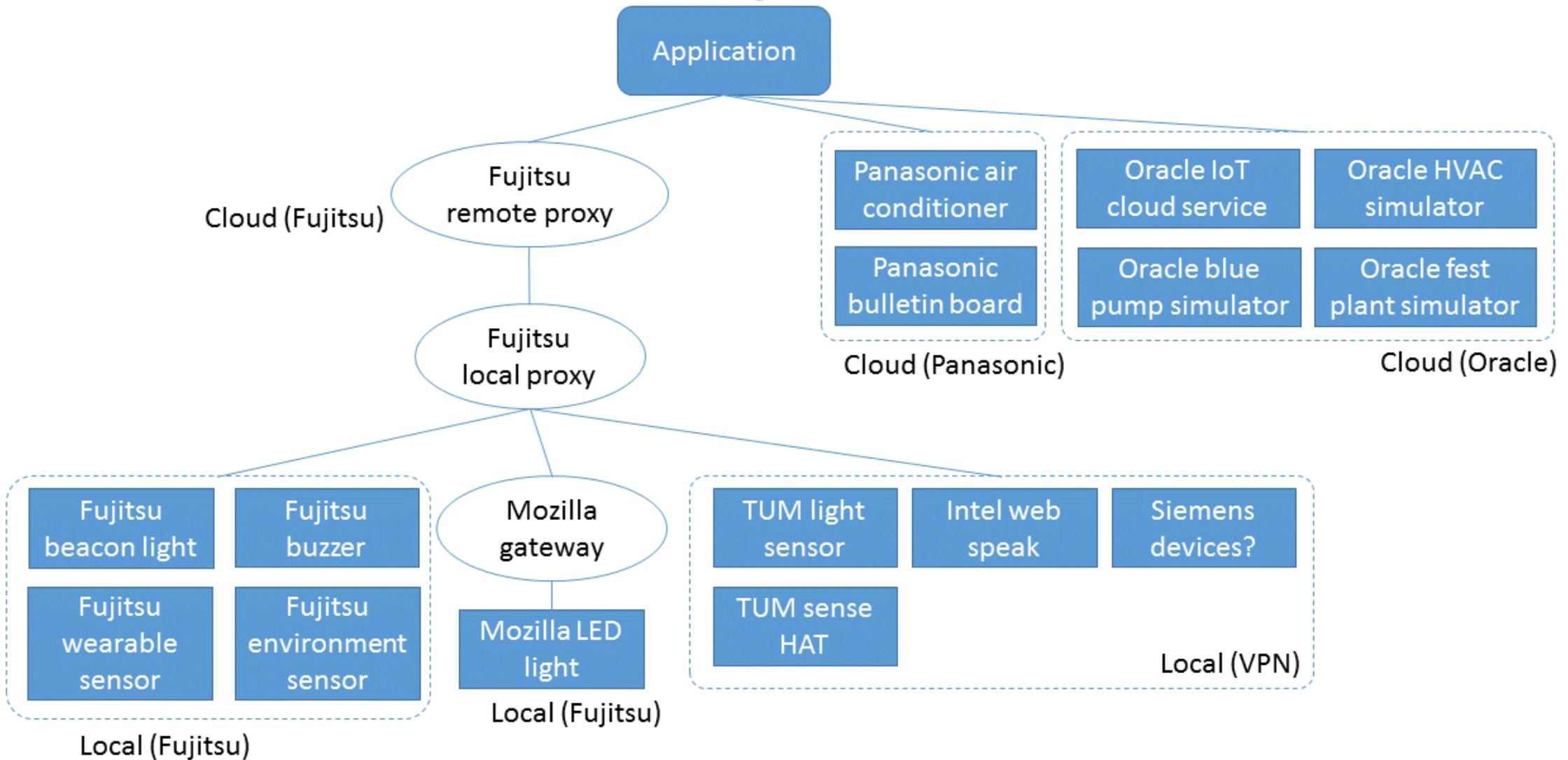
Scenario 1: Home/Building







Scenario 2: Industrial



Chromium Web Browser MAXN En 10:48

Node-RED Dashboard - Chromium

Node-RED Node-RED Dashboard +

localhost:1880/ui/#/2?socketid=ym8xdk8nSxhKgUGIAAAA

Industrial integration

WiFi sensor

Accel

Temperature

Humidity

Air pressure

Sense HAT

Gyro

Accel

Compass

Environ HAT

Heading

LightLevel

x:0.0078125
y:-0.032470703125
z:1.04681396484

RGBValues r:206 g:142 b:138

IoT Digital Twin Simulator - Chromium

IoT Digital Twin Simulator +

https://w3ctest.iadstg.iot.ocs.oraclecloud.com/ds/?root=device&id=54326C2A-FDEC-4E97-9A58-7C93F67...

ORACLE IoT Digital Twin Simulator

CONTROLS

DEVICE STATUS

DATA CHART

EVENTS

- MOTOR FAILURE
- SHORT CIRCUIT
- NOT WORKING

ALERTS

- SEND: Unable to connect to sensors
- SEND: Security breach door opened

DEVICE MONITORING

10:48:45

STATUS

HVAC HVAC_Fujitsu

54326C2A-FDEC-4E97-9A58-7C93F67...

OUTPUT TEMPERATURE 23

FAN VIBRATION 0.56

OIL VISCOSITY 0.01

MOTOR AMPERAGE 49.7

TARGET TEMPERATURE 2

IoT Digital Twin Simulator - Chromium

Settings IoT Digital Twin Simulator +

https://w3ctest.iadstg.iot.ocs.oraclecloud.com/ds/?root=device&id=D142D836-64FF-4A3C-B256-38D10BE...

ORACLE IoT Digital Twin Simulator

EVENTS

- PUMP ON!
- OPEN VALVE!

CUSTOM DATA

- tanklevel
- plantstatus

STATUS

Festo Simulator 1 Festo

D142D836-64FF-4A3C-B256-38D10BE...

PUMPSTATUS

TANK101MAXIMUMLEVELSTA

TANK101MINIMUMLEVELSTA

TANK101OVERFLOWSTATUS

TANK102LEVELVALUE 53

TANK102OVERFLOWSTATUS

VALVESTATUS

Car Charging



* 06:00
*c 20 °C
▲ 4 Okta
● 0 mm/hour

1h

< back

0 kW

15 kW

50 kW



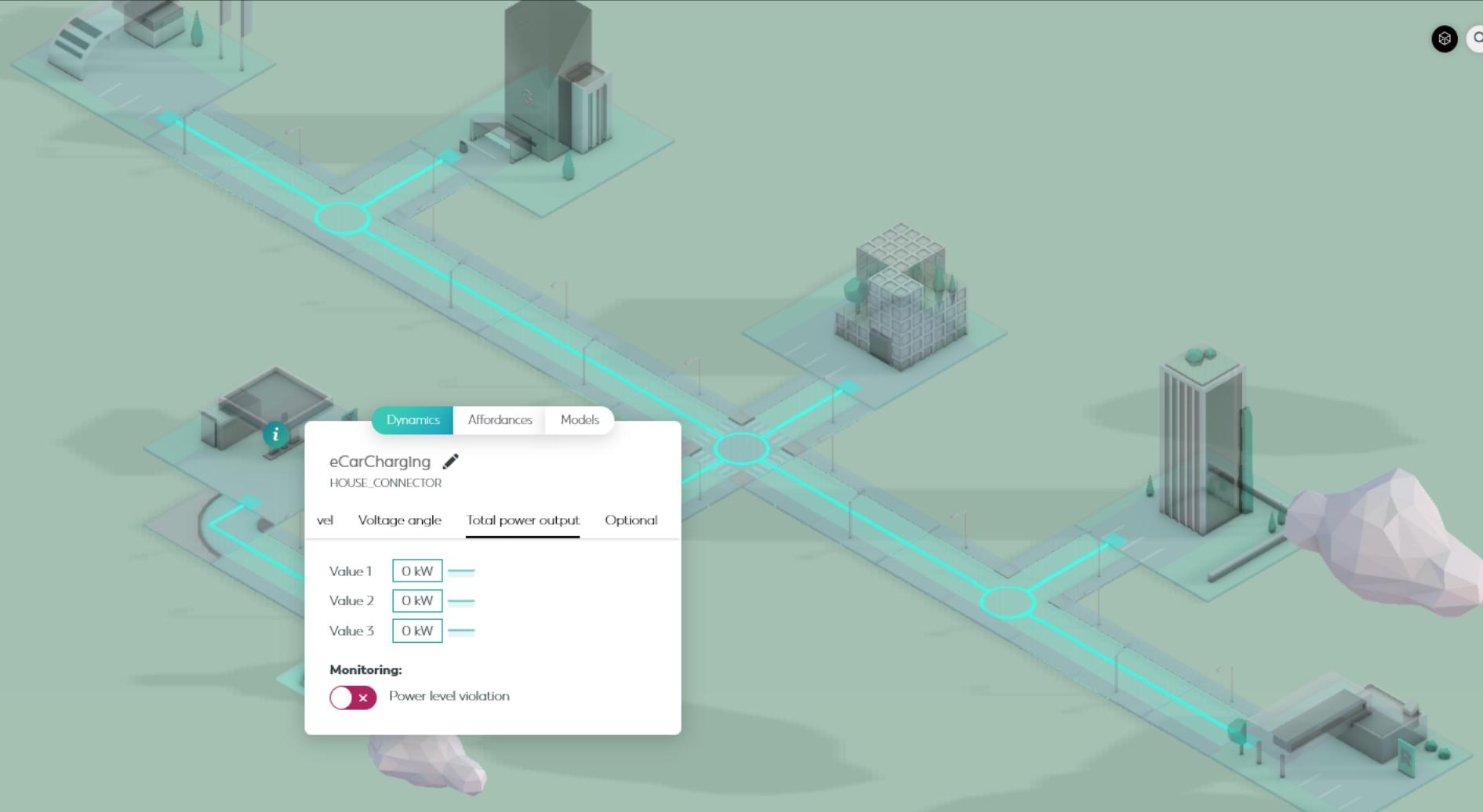
eCarCharging HOUSE_CONNECTOR

Dynamics Affordances Models

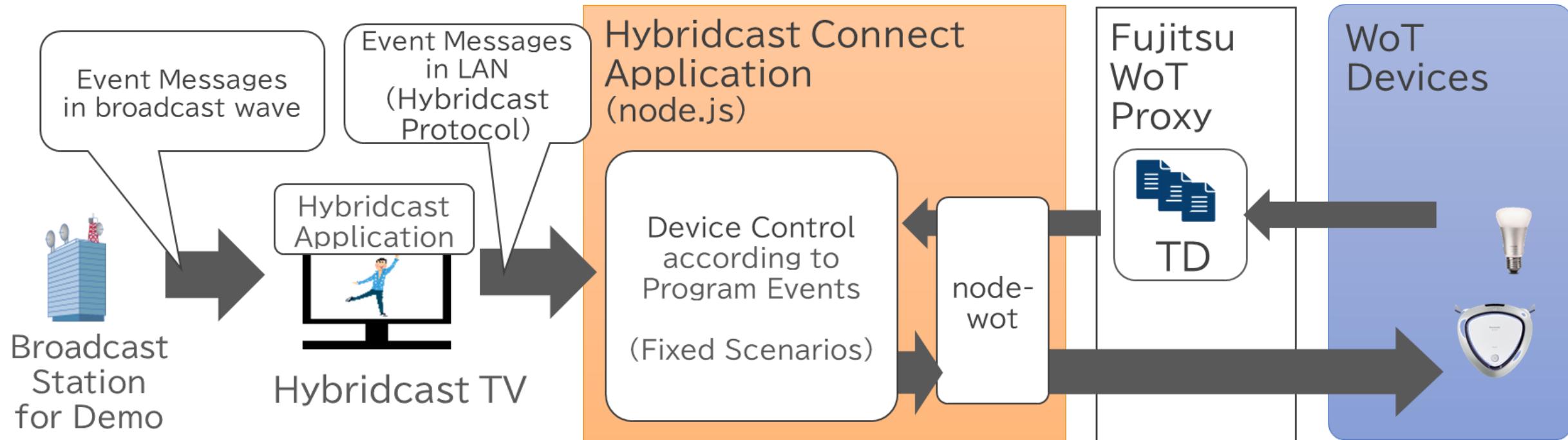
vel Voltage angle Total power output Optional

Value 1: 0 kW Value 2: 0 kW Value 3: 0 kW

Monitoring:
 Power level violation



NHK Hybridcast Integration





T

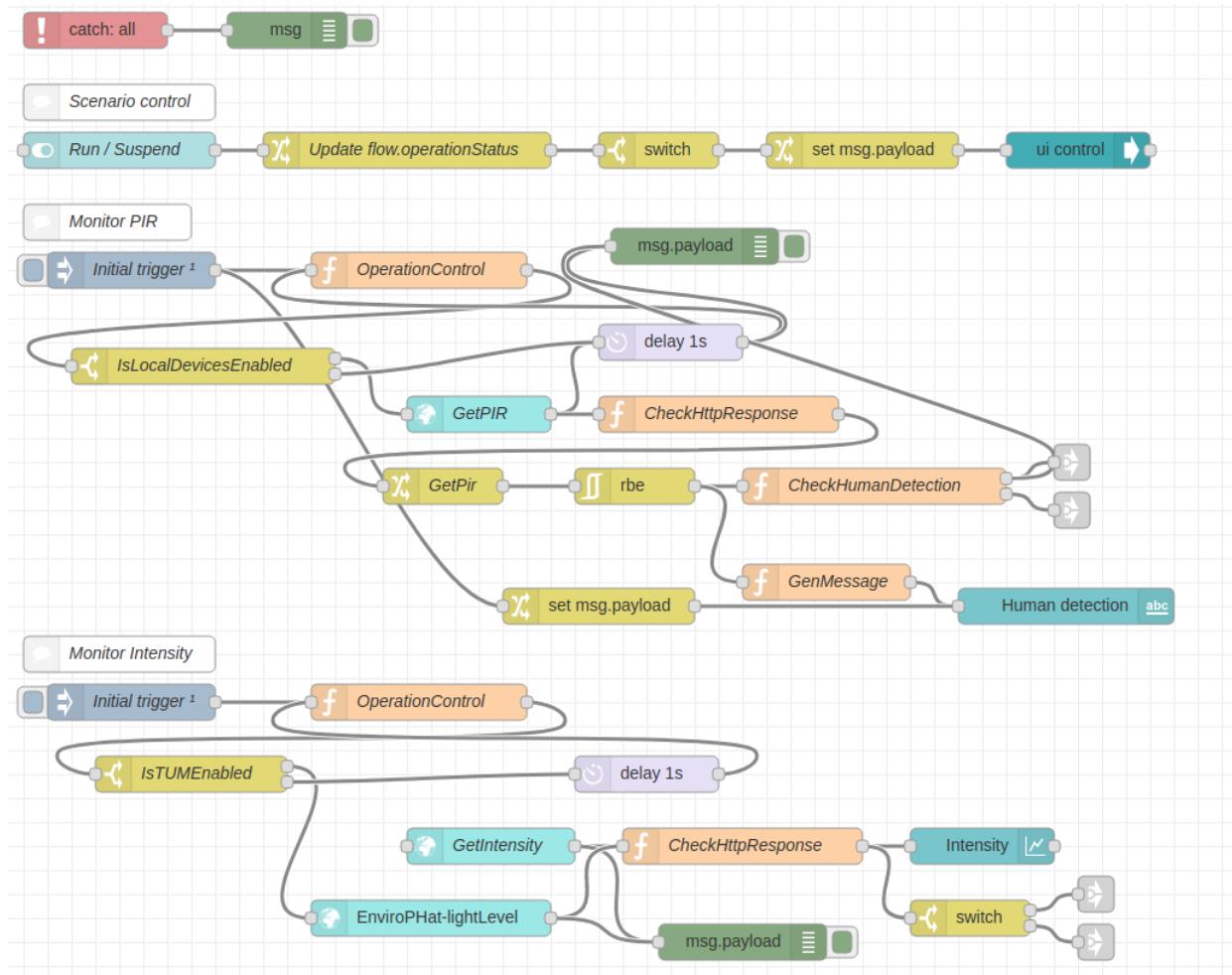
asonic

Panasonic

SANWA
SUPPLY

Orchestration

Node-RED



node-wot

```
WoTHelpers.fetch("coap://localhost:5683/counter").then( async (td) => {
// using await for serial execution (note 'async' in then() of fetch())
try {
```

```
let thing = await WoT.consume(td);
console.info("== TD ===");
console.info(td);
console.info("=====");
```



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

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

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

```
// decrement property
await thing.invokeAction("decrement");
let dec1 = await thing.readProperty("count");
console.info("count value after decrement is", dec1);
```

```
} catch(err) {
  console.error("Script error:", err);
}
```

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

New WoT WG Charter Work Items

Architectural Requirements, Use Cases, and Vocabulary

- Understand and state requirements for new use cases, architectural patterns, and concepts.

Link Relation Types:

- Definition of specific link relation types for specific relationships.

Observe Defaults:

- For protocols such as HTTP where multiple ways to implement "observe" is possible, define a default.

Implementation View Spec:

- More fully define details of implementations.

Interoperability Profiles:

- Support plug-and-play interoperability via a profile mechanism
- Define profiles that allow for finite implementability

Thing Description Templates:

- Define how Thing Descriptions can be defined in a modular way.

Complex Interactions:

- Document how complex interactions can be supported via hypermedia controls.

Discovery:

- Define how Things are discovered in both local and global contexts and Thing Descriptions are distributed.

Identifier Management:

- Mitigate privacy risks by defining how identifiers are managed and updated.

Security Schemes:

- Vocabulary for new security schemes supporting targeted protocols and use cases.

Thing Description Vocabulary:

- **Extensions to Thing** Description vocabulary definitions.

Protocol Vocabulary and Bindings:

- Extensions to protocol vocabulary definitions and protocol bindings.

New WoT WG Charter Work Items

Architectural Requirements, Use Cases, and Vocabulary

- Understand and state requirements for new use cases, architectural patterns, and concepts.

Link Relation Types:

- Definition of specific link relation types for specific relationships.

Observe Defaults:

- For protocols such as HTTP where multiple ways to implement "observe" is possible, define a default.

Implementation View Spec:

- More fully define details of implementations.

Interoperability Profiles:

- Support plug-and-play interoperability via a profile mechanism
- Define profiles that allow for finite implementability

Thing Description Templates:

- Define how Thing Descriptions can be defined in a modular way.

Complex Interactions:

- Document how complex interactions can be supported via hypermedia controls.

Discovery:

- Define how Things are discovered in both local and global contexts and Thing Descriptions are distributed.

Identifier Management:

- Mitigate privacy risks by defining how identifiers are managed and updated.

Security Schemes:

- Vocabulary for new security schemes supporting targeted protocols and use cases.

Thing Description Vocabulary:

- Extensions to Thing Description vocabulary definitions.

Protocol Vocabulary and Bindings:

- Extensions to protocol vocabulary definitions and protocol bindings.

W3C WoT Resources

- W3C WoT Wiki
 - <https://www.w3.org/WoT/IG/wiki>
(IG/WG organizational information)
- W3C WoT Interest Group
 - <https://www.w3.org/2016/07/wot-ig-charter.html>
(old charter)
 - <https://www.w3.org/2019/10/wot-ig-2019.html>
(new charter)
 - <https://lists.w3.org/Archives/Public/public-wot-ig/>
(mailing list)
 - <https://github.com/w3c/wot>
(technical proposals)
- W3C WoT Working Group
 - <https://www.w3.org/2016/12/wot-wg-2016.html>
(old charter)
 - <https://cdn.staticly.io/gh/w3c/wot/master/charters/wot-wg-charter-draft-2019.html?env=dev>
(new charter draft)
 - <https://www.w3.org/WoT/WG/>
(dashboard)
- W3C WoT Candidate Recommendations
 - <https://www.w3.org/TR/wot-architecture/>
 - <https://www.w3.org/TR/wot-thing-description/>
- W3C WoT Working Drafts / Group Notes
 - <https://www.w3.org/TR/wot-binding-templates/>
 - <https://www.w3.org/TR/wot-scripting-api/>
 - <https://www.w3.org/TR/wot-security/>
- W3C WoT Editors' Drafts and Issue Tracker
 - <https://github.com/w3c/wot-architecture/>
 - <https://github.com/w3c/wot-thing-description/>
 - <https://github.com/w3c/wot-binding-templates/>
 - <https://github.com/w3c/wot-scripting-api/>
 - <https://github.com/w3c/wot-security/>
 - <https://github.com/w3c/wot-security-best-practices/>
 - <https://github.com/w3c/wot-profile/>
- Reference Implementations and Tools: node-wot
 - node-wot: <https://github.com/eclipse/thingweb.node-wot>
 - TD playground: <https://github.com/thingweb/thingweb-playground>

Contacts

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

Dr. Michael McCool

Principal Engineer

Intel

Technology Pathfinding

michael.mccool@intel.com

Dr. Sebastian Kaebisch

Research Scientist

Siemens

Corporate Technology

sebastian.kaebisch@siemens.com