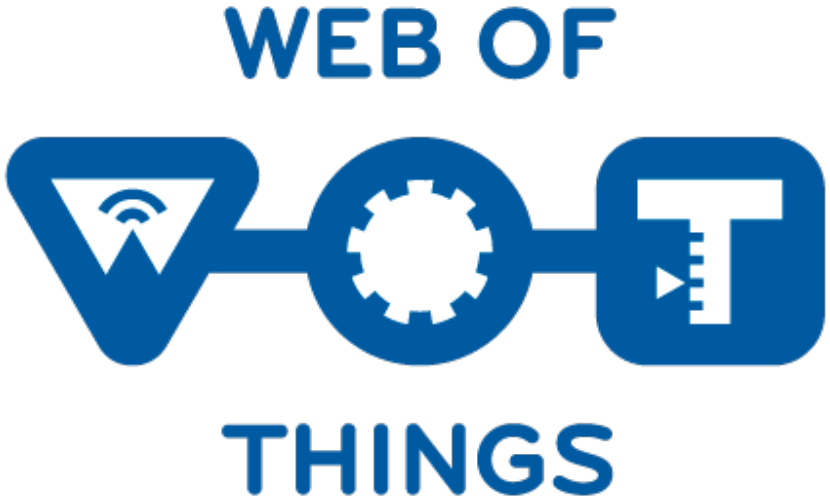


Node-RED – WoT-Discovery Integration and Thing Orchestration

Kunihiko Toumura

2021.10.11

Node-RED – WoT Discovery Integration

We updated a query interface on Node-RED WoT-Discovery plugin so that it can query TDD using SPARQL.

```
PREFIX td: <https://www.w3.org/2019/wot/td#>
PREFIX dc: <http://purl.org/dc/terms/>
SELECT DISTINCT ?id ?title ?desc
WHERE {
    ?id a td:Thing;
        dc:title ?title;
        dc:description ?desc.
        dc:description|dc:title ?desctitle
    FILTER(contains(lcase(str(?desctitle)), lcase("${req.body.query}”))).
}
LIMIT 100
```

Things

Sync script Collects TDs and convert to RDF triple

Thing Description Directory

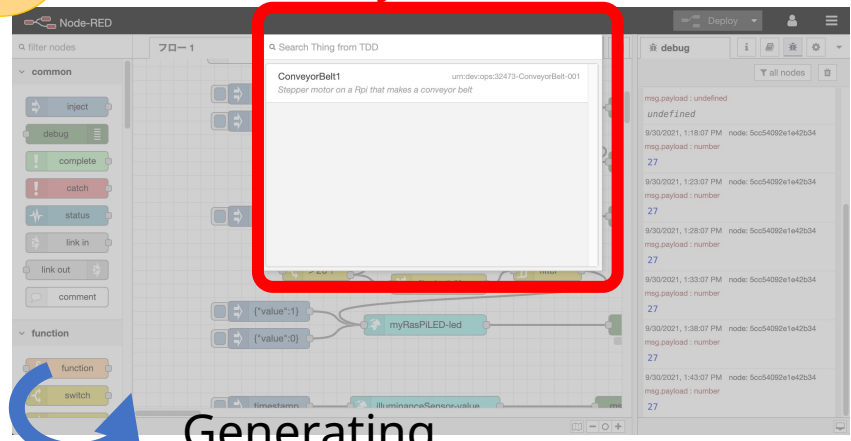
RDF

Siemens/Logilab

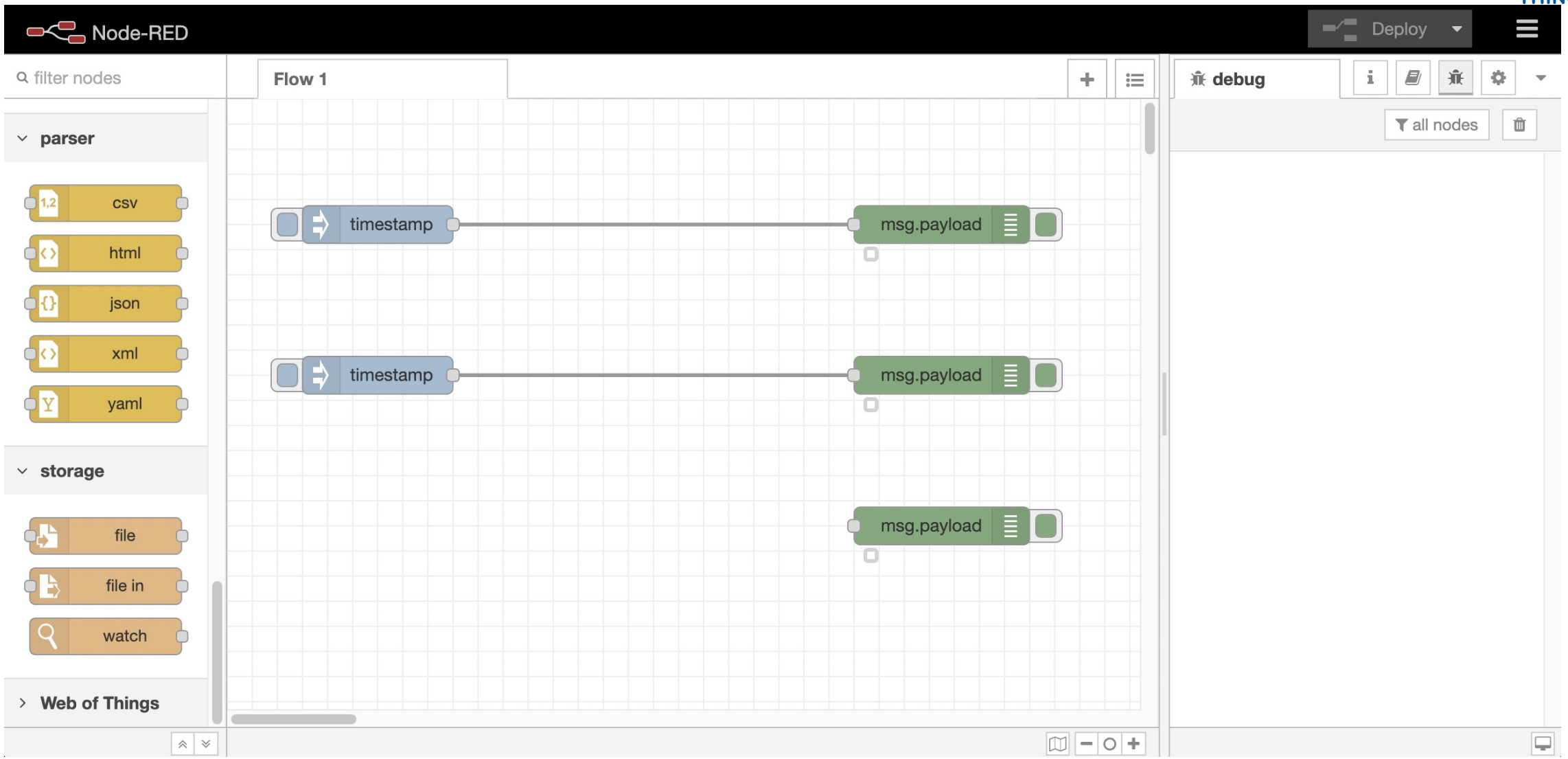
Retrieve TDs using **SPARQL** Query

Node-RED

Integrated Query interface



Generating Node from TD (Node Generator)


Node-RED
Deploy
filter nodes
Flow 1
debug
all nodes
parser
csv
html
json
xml
yaml
storage
file
file in
watch
Web of Things
timestamp
msg.payload
timestamp
msg.payload
msg.payload

Issues / Discussion / Lessons Learned

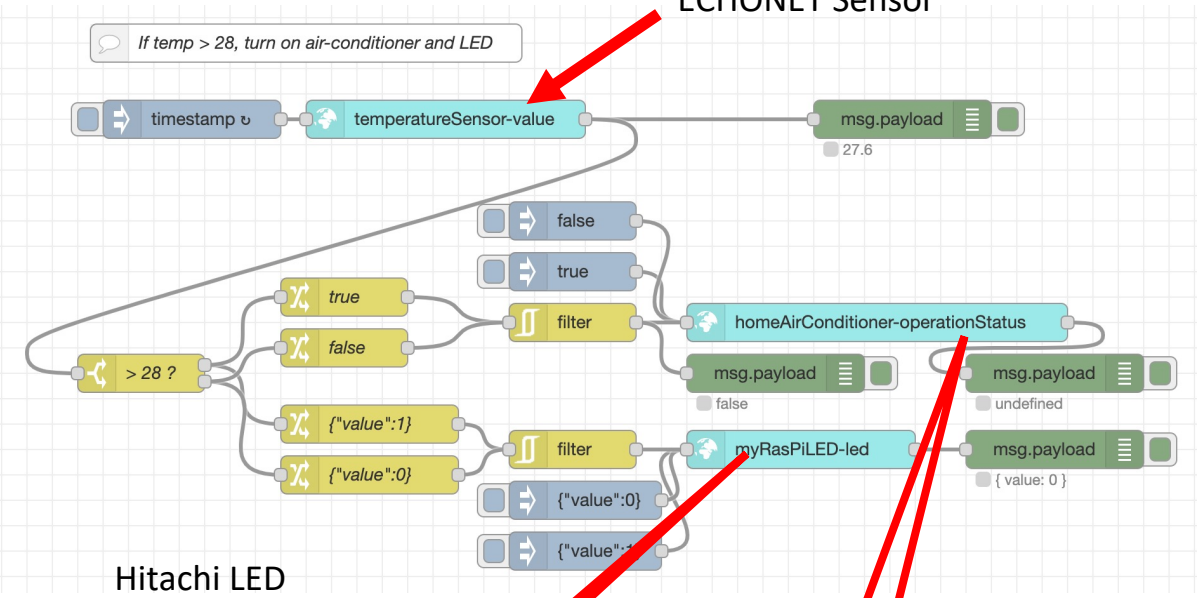
- Round-tripping issues
  - Invalid `securityDefinitions` (w3c/wot-thing-description#1193)
  - `title` and `description` are converted to `titles` and `descriptions`
    - Node generator assumes that all TDs have a `title`.
  - `href`s in `form`s are renamed to `hasTarget`.

- Correct `@context` and `@type` are needed to register a TD to TDD.

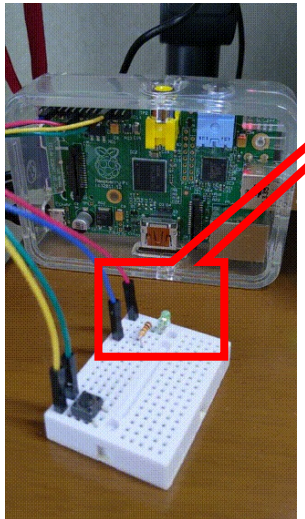
- Need a better search UI that leverages the power of SPARQL
  - The current implementation simply does a keyword search of `description` in TD.

Orchestrating Things using Node-RED

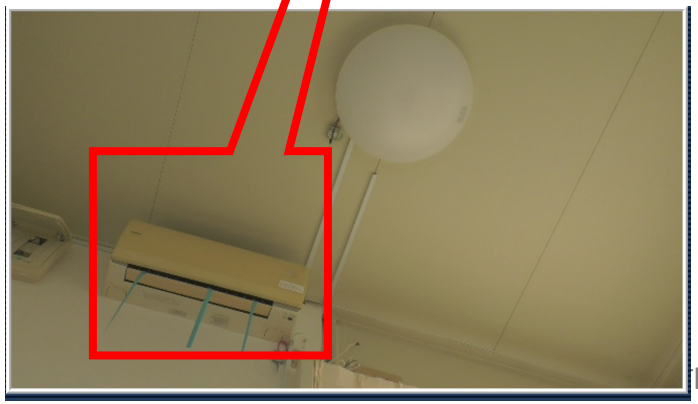
Thermostat



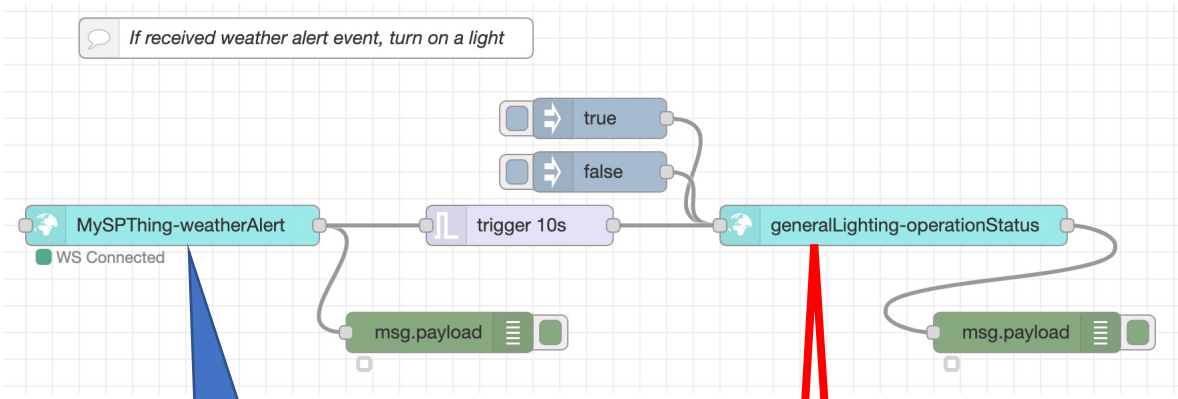
Hitachi LED



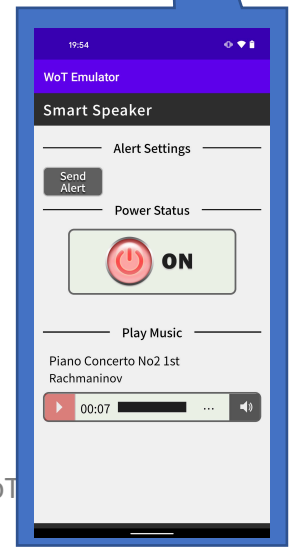
ECHONET Aircon



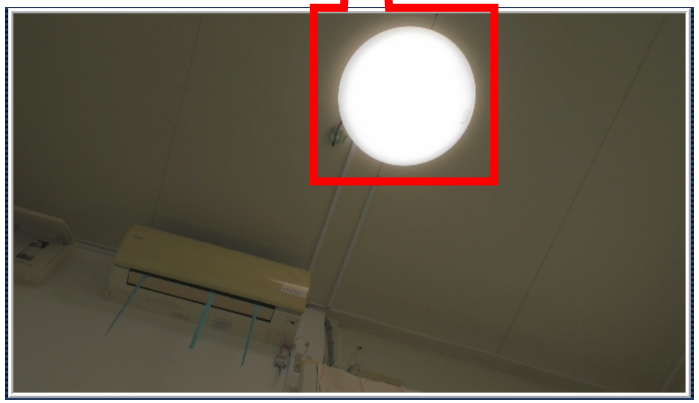
Visual alert



NHK Smart Speaker



ECHONET Light



Issues / Discussion / Lessons Learned

- Direct access to ECHONET Lite web API
  - In this plugfest, we can access ECHONET device via gateway (WoT-ECHONET Lite Web API Intermediary).
    - Gateway removes an encapsulating object from a response payload.
  - If we describe ECHONET Lite Web API using TD, we can also directly access to ECHONET Lite Web API.
  - Between DD and TD, the scope specified by each schema is different. By adding the encapsulating object schema when converting DD to TD, it is possible to access ECHONET Lite Web API directly … ?

■ リクエスト (request)

```
GET /elapi/v1/devices/<device id>/properties/rgb HTTP/1.1
```

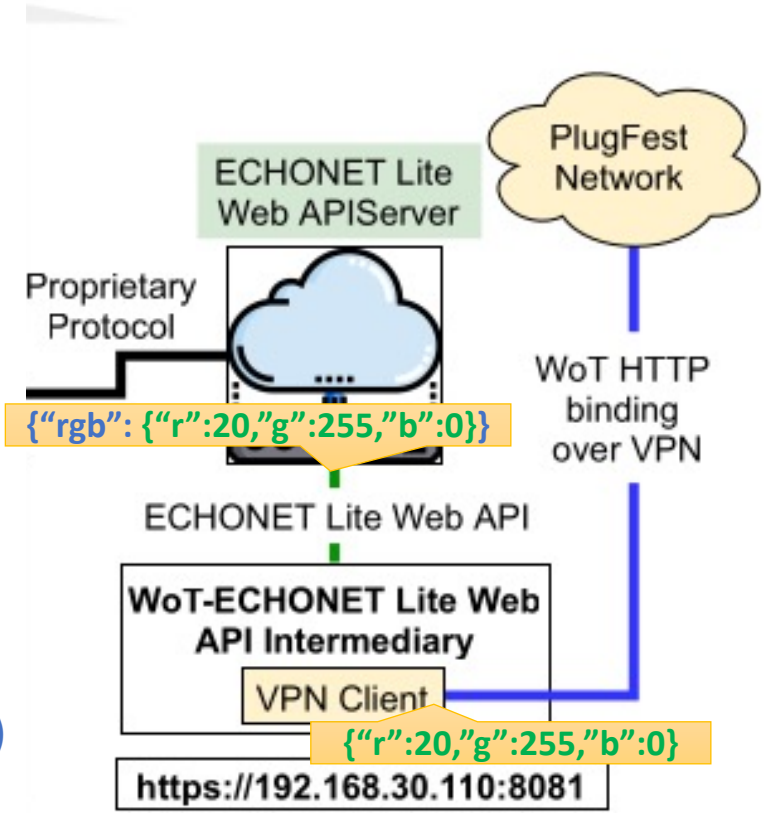
■ レスポンスが object の場合 (When the response is object)

```
{
    "rgb": {
        "r": 20,
        "g": 255,
        "b": 0
    }
}
```

ECHONET Lite Web API Device Description (DD) describes the schema of this part

WoT Thing Description (TD) describes the schema of this part

https://echonet.jp/wp/wp-content/uploads/pdf/General/Download/web_API/ECHONET_Lite_Web_API_Specs_v1.1.3.pdf



https://github.com/w3c/wot-testing/blob/main/events/2021.09.Online/Projects/ECHONET/20210917_Plugfest_ECHONETControllerUpdated.png