Toward a Simplified WoT Thing Model

W3C Web of Things IG/WG F2F OpenDay

Prague, Czech Republic, March 2018

Thing Description Sapporo 2015

```
{
  "@context": "http://w3c.github.io/wot/w3c-wot-td-context.jsonld",
  "metadata": {
    "name": "MyDoor",
    "protocols": {
      "WS": {
        "uri": "ws://www.example.com:80/door",
        "priority" : 1
      },
      "HTTP": {
        "uri": "http://www.example.com:80/door",
        "priority": 2
      }
    },
    "encodings": [ "JSON" ]
  },
  "interactions": [{
    "@type": "Property",
    "name": "stateOpen",
    "outputData": "xsd:boolean",
    "writable": false
  },{
    "@type": "Event",
    "name": "stateChanged",
    "outputData": "xsd:boolean"
  }]
}
```

Invalid JSON-LD container for metadata

Addressing via *uri* + *name*

XML Schema Types

Thing Description Montreal 2016

```
{
  "@context": ["http://w3c.github.io/wot/w3c-wot-td-context.jsonld",
               { "actuator": "http://example.org/actuator#" } ],
  "@type": "Thing",
  "name": "MyLEDThing",
  "uris": [ "coap://www.myled.com:5683/", "http://www.myled.com:8080/myled/" ],
  "encodings": [ "JSON", "EXI4JSON" ],
  "security": {
    "cat": "token:jwt",
    "alg": "HS256",
    "as": "https://authority-issuing.org"
  },
  "properties": [{
    "@type": "actuator:onOffStatus",
    "name": "status",
    "valueType": "xsd:string",
    "writable": true,
    "hrefs": [ "status", "myled/status" ]
  }],
  "actions": [{
    "@type": "actuator:fadeIn",
    "name": "fadeIn",
    "inputData": { "valueType": "xsd:short" },
    "hrefs": [ "in", "myled/in" ]
  }],
  "events": [{
    "name": "criticalCondition",
    "valueType": "xsd:string",
    "hrefs": [ "ev", "myled/event" ]
  }]
}
```

Domain-specific vocabulary

Security metadata

Prop. / Action / Event on top level

Array arithmetic

Thing Description Now (1/3)

Valid JSON-LD 1.0

Protocol Bindings

Semantic Data Schema with JSON Schema terms

```
{
  "@context": [
    "https://w3c.github.io/wot/w3c-wot-td-context.jsonld",
    { "iot": "http://iotschema.org/" }
  ],
  "@type": ["Thing"],
  "name": "MyLEDThing",
  "security": {
    "authentication": "Bearer",
    "cat": "token:cwt",
    "as": "https://authority-issuing.example.org"
  },
  "interaction": [
    {
      "@type": ["Property", "iot:SwitchStatus"],
      "name": "status",
      "writeable": true,
      "observable": true,
      "schema": {
        "dataType": "boolean",
        "@type": ["iot:BinarySwitch"]
      },
      "form": [ ... ]
    }, {
```

Thing Description Now (2/3)

Basics to build the request

Deviation from defaults

```
{
  "@type": ["Property", "iot:SwitchStatus"],
  "name": "status",
  ...
  "form": [
    {                    /// Default: GET to read, PUT to write
      "href": "https://myled.example.com:8080/pwr",
      "mediaType": "application/json"
    }
  ]
}, {
  "@type": ["Action", "iot:TurnOn"],
  "name": "fadeIn",
  ...
  "form": [
    {
      "href": "coaps://myled.example.com:5684/pwr",
      "mediaType": "application/ocf+cbor",
      "coap:methodCode": 3,    /// PUT instead of POST to invoke
      "coap:options": [ {
        "coap:optionNumber": 2053, /// OCF-Content-Format-Version
        "coap:optionValue": "1.1.0"
      } ]
```

Thing Description Now (3/3)

```
{
  "@context": [
    "https://w3c.github.io/wot/w3c-wot-td-context.jsonld",
    { "iot": "http://iotschema.org/" }
  ],
  "@type": ["Thing"],
  "name": "MyLEDThing",
  "security": {
    "authentication": "Bearer",
    "cat": "token:cwt",
    "as": "https://authority-issuing.example.org"
  },
  "interaction": [
    ...
  ],
  "links": [{
    "href": "https://servient.example.com/motion-detector/td",
    "rel": "controlledBy",
    "mediaType": "application/ld+json"
  }]
}
```

Web Linking

JSON-LD 1.1

- Proposed charter
  - https://www.w3.org/2018/03/jsonld-wg-charter.html
  - AC Review until 29 April 2018
- Proposed Working Group
  - Start date 1 June 2018
  - End date 31 December 2019
- Mandate
  - “Make JSON-LD even more like idiomatic JSON”
  - **Keys not Arrays**
  - Framing!

@context #103

"I agree on using this for optional semantic annotations

Could we have the option for a context to be a single string (URI) rather than an array? You could still use an array if multiple contexts are needed."

- Semantic annotations fully optional
- Media-type based preprocessing `application/td+json`
- No restrictive conventions e.g., always arrays where possible

```
{
  "@context": {
    "@version": 1.1,
    /* @id generation relative to Thing */
    "@base": "@id",
    /* object notation for Interactions */
    "properties": {
      "@container": "@id"
    },
    "actions": {
      "@container": "@id"
    },
    "events": {
      "@container": "@id"
    },
    /* possibly schema type definitions */
    "definitions": {
      "@container": "@id"
    },
    /* TD term definitions as in current
       w3c-wot-td-context.jsonld */
```

Objects vs Arrays #111

"As I've mentioned before, we'd prefer the object representation. You can easily iterate through keys of an object in JavaScript. Having an ID in an array element is doable, but thing.properties[0] is a lot more clunky than thing.properties.on."

- JSON-LD 1.1 containers
- Key can become @id
- Interactions become referencable

```
"properties": {
  "on": {
    "type": "boolean",
    ...
  }
},
"actions": {
  "fade": {
    "input": {
      "type": "object",
      ...
```

base #106

“Similarly, why is a base URL needed? Can the Thing Description's URL be used as the base URL?”

- Allow origin URI as base?
- Note TD sharing implication
- Cannot serve as ID

```
{
  "name": "lamp",
  "base": "https://...",
  ...
      "href": "prop/a",
  ...
}
```

@id/id #105

"Why is this necessary? Why is the URL of the Thing Description not a suitable URI to identify a Thing? Why is an additional urn needed?"

- Make Thing @id mandatory?
- Mask with plain "id" term?
- Important to match security metadata in runtime

```
{
  "name": "lamp",
  "@id": "urn:dev:org:1234…",
  "base": "http://...",
  ...
}
```

schema #107

"I'm interested in the idea of directly using JSON Schema syntax like this to consistently define data schemas for properties, actions and events."

- Recursive Properties?
- Property == DataSchema?

- Repeated keys → repeated @id!
- $ref vs LD references!

```
"properties": {
  "status": {
    "type": "object",
    "properties": {
      "battery": {
        "type": "number",
        "minimum": 0.0,
        "maximum": 100.0,
      },
      "rssi": {
        "type": "number",
        "minimum": 0.0,
        "maximum": 1.0
      }
    },
    ...
```

@type #104

"Could the three levels of @type (thing type, capability type and data type) be reduced down to two? (thing type and capability type)"

- Reminder
  - optional
  - Valuable addition over syntax only
- Clear concept for iot.schema.org
- Note Property == DataSchema
  - Thing types vs Capability lists
  - Semantic Interaction types
- No semantic data type level
  - Interaction types at sub-properties

```
{
  "name": "MyPIR",
  "@id": "urn:dev:org:1234…",
  "@type": ["Thing","iot:Presence"],
  "properties": {
    "presence": {
      "type": "boolean",
      "@type": "iot:PresenceStatus"
      "links" [{
        "href": "...",
        "mediaType": "..."
      }]
    }
  }
}
```

links vs forms #88, #108

“As discussed in the meeting, I'm still not convinced by this use of "forms" as a generic mechanism for declaratively defining protocol bindings to any protocol inside a Thing Description.”

- Simplify and restrict for more convergence?
  - Still allow “expensive modelling”
  - Unusual Prop. write through Action
- Properties have links?
- Actions have forms?
- Events have …?

```
"properties": {
  "on": {
    "type": "boolean",
    "links" [{
      "href": "...",
      "mediaType": "..."
    }]
  }
},
"actions": {
    "fade": {
      "input": { ... },
      "forms": [{
        "href": "...",
        "encType": "...", // submissionTy
        "mediaType": "...“,
        ...
```

Scripting API

TD

```
{
  "name": "MyThing",
  "@id": "urn:dev:org:1234…",
  "properties": {
    "on": {
      "type": "boolean",
      "@type": "iot:SwitchStatus",
      "writable": true,
      "links" [{ ... }]
    }
  },
  "actions": {
    "fade": {
      "input": { ... },
      "forms": [{ ... }]
    }
  }
}
```

```
let thing = WoT.consume(td);
// metadata
console.log(thing.name);
let schema = thing.properties.on.type;
let semant = thing.properties.on["@type"];
// interact
let status = await thing.properties.on.get();
if (thing.properties.on.writable) {
  thing.properties.on.set(false);
}
// auto-fill from client state
// based on @type annotations
let params =
  myUtils.semanticFill(thing.actions.fade.input);
thing.actions.fade.invoke(params);
// use TD naturally (highly experimental)
let proxy = WoT.produce({ name: "MyProxy" });
thing2.properties.on = { ... };
thing2.properties.on.get = () => { ... };
...
thing2.actions.fade.invoke = (params) => { ... };
```

Simplified Thing Description with JSON-LD 1.1

- Proposal (work in progress)
  - https://w3c.github.io/wot-thing-description/proposals/simplified-td/

- Tuesday program on TD
  - 11:30-12:15 -- Interaction Pattern Definitions
  - 14:00-14:45 -- TD Data Schema Now and with JSON-LD 1.1
  - 14:45-16:00 -- Simplified TD with JSON-LD 1.1
  - 16:30-17:30 -- Security Metadata for Thing Description
  - 17:30-18:00 -- New TD Vocabulary (description, etc.)