



# Reduce TD redundant information

Can we describe more with less?

# Problem

Thing Descriptions contains vital information to understand how to interact with a web thing.

Web things may be **complex** software agents with a variety of affordances and APIs. Example: Thing Description Directory.

Therefore, a Thing Description can easily become **hard to read and maintain** for a human being.

Machines would prefer also shorter TDs because the shorter they are the shorter is the loading time.

# Reasons why Thing Descriptions get longer



- Security definitions
- Different protocol supported
- Different flavors of the same API endpoint
- Intrinsic protocol redundancy (i.e., identical protocol information is duplicated in every form)
- Complex data schemas

# Security Definitions

The main complain that we received about security definition is the inability to express a security scheme inline.

Useful one the TD has just one schema and it used across all the Affordances.

- [w3c/wot-thing-description#945](https://w3c/wot-thing-description#945)
- has implications also in signing

```
"securityDefinitions": {
  "no_sec": {
    "scheme": "nosec"
  }
},
"security": "no_sec",
```

```
"title": "Test",
"security": { "scheme" : "nosec" },
```

# Different protocol supported

If a webthing support different protocol this usually mean a linear increment of its forms.

If the API can be described by two forms (e.g. one for read and one for write) if the thing supports other 3 protocols it means that each affordance may contain  $3 \times 2 = 6$  forms.

```
"forms": [
  {
    "href": "http://example.com/test"
  },
  {
    "href": "mqtt://example.com/test"
  },
  {
    "href": "coap://example.com/test"
  },
  {
    "href": "modbus://example.com"
  }
]
```

# Different flavors of the same API endpoint

Node-wot generates a group of form for each network interface.

The affordances and their data schemas are usually hidden behind a long list of forms.

Counter example Thing Description contains **798** lines.

```
writeonly: true,
- forms: [
  - {
    href: "http://172.23.112.1:8080/counter/properties/count",
    contentType: "application/json",
    op: [
      "readproperty"
    ],
    htv:methodName: "GET"
  },
  - {
    href: "http://172.23.112.1:8080/counter/properties/count/observable",
    contentType: "application/json",
    op: [
      "observeproperty",
      "unobserveproperty"
    ],
    subprotocol: "longpoll"
  },
  - {
    href: "http://169.254.214.40:8080/counter/properties/count",
    contentType: "application/json",
    op: [
      "readproperty"
    ],
    htv:methodName: "GET"
  },
  - {
    href: "http://169.254.214.40:8080/counter/properties/count/observable",
    contentType: "application/json",
    op: [
      "observeproperty",
      "unobserveproperty"
    ],
    subprotocol: "longpoll"
  },
  - {
    href: "http://192.168.178.110:8080/counter/properties/count",
    contentType: "application/json",
    op: [
      "readproperty"
    ],
    htv:methodName: "GET"
  },
  - {
    href: "http://192.168.178.110:8080/counter/properties/count/observable",
    contentType: "application/json",
    op: [
      "observeproperty",
      "unobserveproperty"
    ],
    subprotocol: "longpoll"
  },
  - {
    href: "http://172.22.48.1:8080/counter/properties/count",
    contentType: "application/json",
    op: [
```

# Intrinsic protocol redundancy

Some protocols requires the same configuration in each forms. Examples:

- Modbus
  - Content-type is usually always **application/octet-stream** (defaults may help here)
  - Endianess must be provided in the content-type, but usally is the same for all forms (e.g. **application/octet-stream;byteSeq=BIG\_ENDIAN**)
- Websockets
  - Most websocket protocols needs only an endpoint to describe the interaction.
- MQTT
  - Broker connection parameters are shared among all the forms

[wot-thing-description#878](#)

# Complex data schemas

- Consider for example to correctly describe the schema output of an action that returns another TD. (Real world example: TDDs returns a Thing Description for the **retrieveTD** action).

Thing description schema: 1280 lines

- Data schema used in multiple places.

In the current spec we tried to mitigate this issue thanks to the introduction of: **schemaDefinitions**