



# Thing Description

Sebastian Kaebisch

7 October 2021

# Outline

- TD Editor's Draft Review
  - What's New
- TD 1.1 vs TD 2.0
  - impact on the TD versioning
  - check the labeled "defer 2.0" issues
- New feature „Collections for TMs“
- Reducing verbosity in TDs (Cris)
- Latest about Binding Templates? (Ege)

# What's New

## Editor's Draft Review

- Many minor editorial changes & bug fixes, e.g.,
  - [fix: maxItems,minItems are xsd:nonNegativeInteger](#)
  - [Fix definition of multipleOf in TD schema](#)
  - [fix example 37 and its canonical form](#)
  - [Security reorder](#)
  - ...
- New features
  - [Add queryaction and cancelaction operations - closes #302](#)
  - [Add queryallactions operation](#)
- TD Signature will be not included in the TD spec
  - [WIP: TD Signatures](#)
  - Instead, a new W3C Note is being drafted as an alternative solution until JSON-LD Signature is available
  - Latest draft: <https://w3c.github.io/wot-ejs/>

# TD 1.1 vs TD 2.0 (I/III)

- The intention of TD 1.1 was to be backward compatible with TD 1.0.  
→ TD processor based on TD 1.0 implementation shall also work for TD 1.1 versions
- Many of the new features in the latest TD 1.1 draft could fulfill the promise so far
- However, some very useful features, however, would break backward compatibility, e.g.
  - [define forms optional and introducing global endpoint definitions for optimization](#)

# TD 1.1 vs TD 2.0 (II/III)

- Example:
  - WebSockets
    - typically, WS are announced once which IP and Port address is used
    - subprotocol would give the context about a specific protocol behavior for the interaction affordances (e.g., webthing, coap)
    - To be compliant with TD 1.0, forms in the interaction affordances would be always have identical, [redundant information](#)
    - each form entry would signalize, that I have open the connection again and again  
→ Solution: Global definition of an “endpoint” that announce WS usage + address information + contentType + subprotocol
  - Other protocols would also benefit of a global “endpoint” definition to point what is the “initialization” of the connection to the Thing which is then used to interact with Thing’s affordances e.g., to MQTT Broker, Modbus device, ...) and to provide basic communication paradigm (contentType, byteSeq (e.g., big or little endian))
  -

# TD 1.1 vs TD 2.0 (III/III)

- Questions:
  - Shall we stay with TD 1.1, however, keep backwards compatibility as much as possible with exceptions? E.g.,
    - Provide developer note that says like “If you want it to be backward compatible, you should provide forms with the mandatory content at the interaction level.”
  - Or should we be clear and go for TD 2.0?
  - Or should have both, TD 1.1 and TD 2.0 in same charter?
  - If so, we should evaluate all [“defer to TD 2.0” labeled issues](#)
    - [Adding term to indicate a stream of data](#)
    - [JSON Schema breaking changes in Drafts 2019 and newer](#)
    - [Undefined behavior for writeOnly and readOnly](#)
    - [Single 'base' field but multiple 'forms' elements](#)
    - [Reconsider security & securityDefinitions being mandatory](#)
    - [Is the use of "\\$ref" and "definitions" allowed in TD ?](#)
    - [Add overridable global default value\(s\) for mediaType](#)
    - ..

# Thing Model Update

- New update is provided in [WIP: Updates for TM Chapter](#)
- New feature: TM Composition with two TD generation strategies
  - New relation typ in links “tm:submodel”
  - provide “instanceName” for distinguishing same TM compositions
- Evaluated in [PlugFest](#)

```
{
  "@context": "http://www.w3.org/ns/td",
  "@type": "tm:ThingModel",
  "title": "Smart Ventilator Thing Model",
  "version": { "model" : "1.0.0" },
  "links": [
    {
      "rel": "tm:submodel",
      "href": "./Ventilation.tm.jsonld",
      "type": "application/tm+json",
      "instanceName": "ventilation"
    },
    {
      "rel": "tm:submodel",
      "href": "./LED.tm.jsonld",
      "type": "application/tm+json",
      "instanceName": "led"
    }
  ],
  "properties": {
    "status": { "type": "string", "enum": ["On", "Off", "Error"]}
  }
}
```

