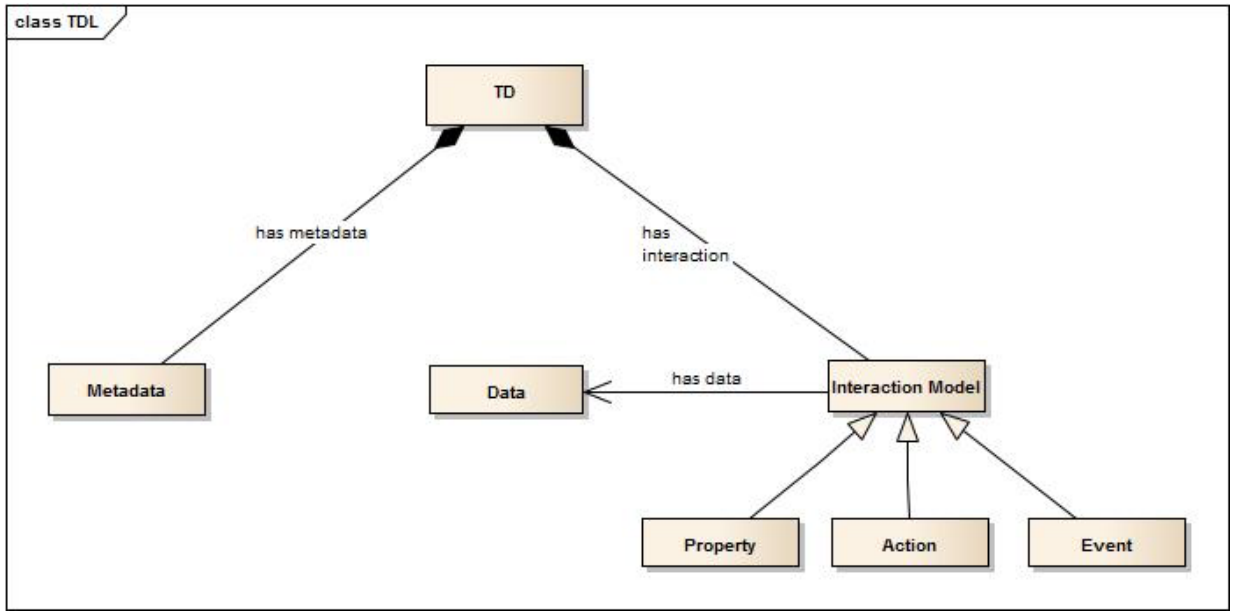
W3C WoT WG
TD Session

VF2F 2020

Sebastian Kaebisch

Remember?




```
{
  "@context": "http://w3c.github.io/wot/w3c-wot-td-context.jsonld",
  "metadata": {
    "name": "Traffic Light Thing",
    "protocols" : {
      "CoAP" : {
        "uri" : "coap://192.168.1.242:5683",
        "priority" : 1
      }
    },
    "encodings": [
      "JSON"
    ]
  },
  "interactions": [
    {
      "@type": "Property",
      "name": "lightStatus",
      "outputData": "xsd:string",
      "writable": false
    } ,{
      "@type": "Action",
      "name": "turnGreen",
      "inputData": "",
      "outputData": ""
    }, {
      "@type": "Action",
      "name": "turnRed",
      "inputData": "",
      "outputData": ""
    }
  ]
}
```

W3C Thing Description is (Almost) a Recommendation!

Web of Things (WoT) Thing Description

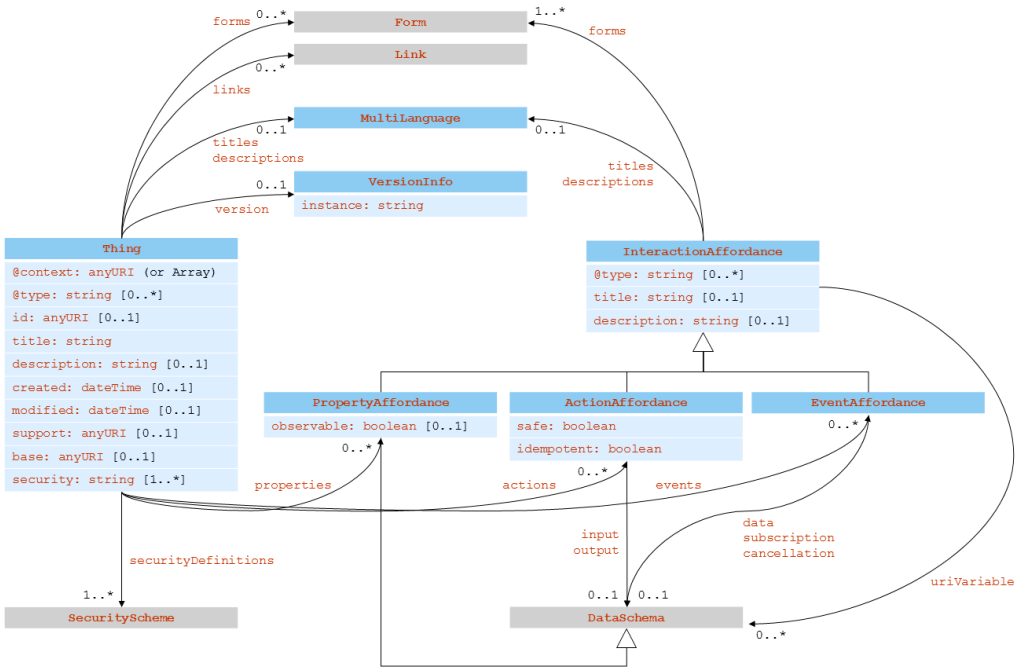
W3C Proposed Recommendation 30 January 2020

This version:
https://www.w3.org/TR/2020/PR-wot-thing-description-20200130/

Latest published version:
https://www.w3.org/TR/wot-thing-description/

Latest editor's draft:
https://w3c.github.io/wot-thing-description/

Implementation report:




```
{
    "@context": "https://www.w3.org/2019/wot/td/v1",
    "id": "urn:dev:ops:32473-WoTLamp-1234",
    "title": "MyLampThing",
    "securityDefinitions": {
        "basic_sc": {"scheme": "basic", "in":"header"}
    },
    "security": ["basic_sc"],
    "properties": {
        "status" : {
            "type": "string",
            "forms": [{"href": "https://mylamp.example.com/status"}]
        }
    },
    "actions": {
        "toggle" : {
            "forms": [{"href": "https://mylamp.example.com/toggle"}]
        }
    },
    "events":{
        "overheating":{
            "data": {"type": "string"},
            "forms": [{
                "href": "https://mylamp.example.com/oh",
                "subprotocol": "longpoll"
            }]
        }
    }
}
```

Well Done Everybody!!!

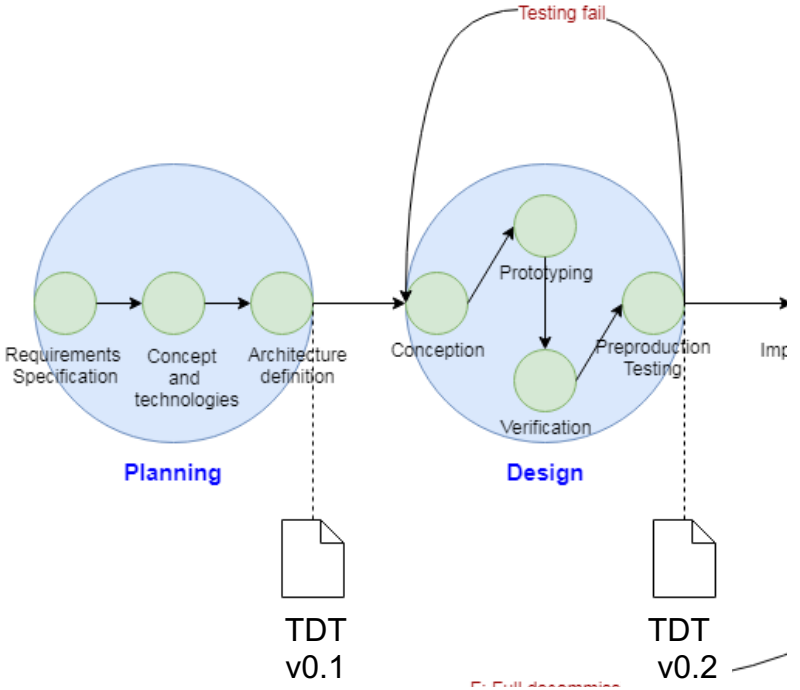
TD Session Agenda

- Thing Description Templates
  - Related issues https://github.com/w3c/wot-thing-description/labels/TD%20Template
- Describing initial connection
  - Related issue https://github.com/w3c/wot-thing-description/issues/878
- Efficient formats for TD (Taki & Daniel)
  - Related issues https://github.com/w3c/wot-thing-description/labels/Optimization
- Full OAuth2 support in TD 1.1
  - Related issues
- Backlog
  - Requirements from Architecture TF
  - Eventing

Thing Description Templates

Where and why Thing Description Template (TDT) are needed?

- TDTs can be included at very early stages of product development, without the need specific safety and communication details. TDTs may crow over design time (e.g., additional interaction affordances are added) and may end-up to valid TD
- TDTs can be used for simulation (just take the data model defined in the interaction affordances)
- Describe classes of devices
  → define the metadata and capacities of a device once, apply to all device instances
- TDTs already (unknowingly) used for WoT.produce() in Scripting APIs
- …



What is the Formal Definition of a TDT?

- $\mathrm{TDT} \subset TD$
- What is the minimum requirement to be a TDT? E.g.,

```
{
}
```

or

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

or

```
{
"@context": "https://www.w3.org/2019/wot/td/v1",
"title": ”TDT example"
}
```

or

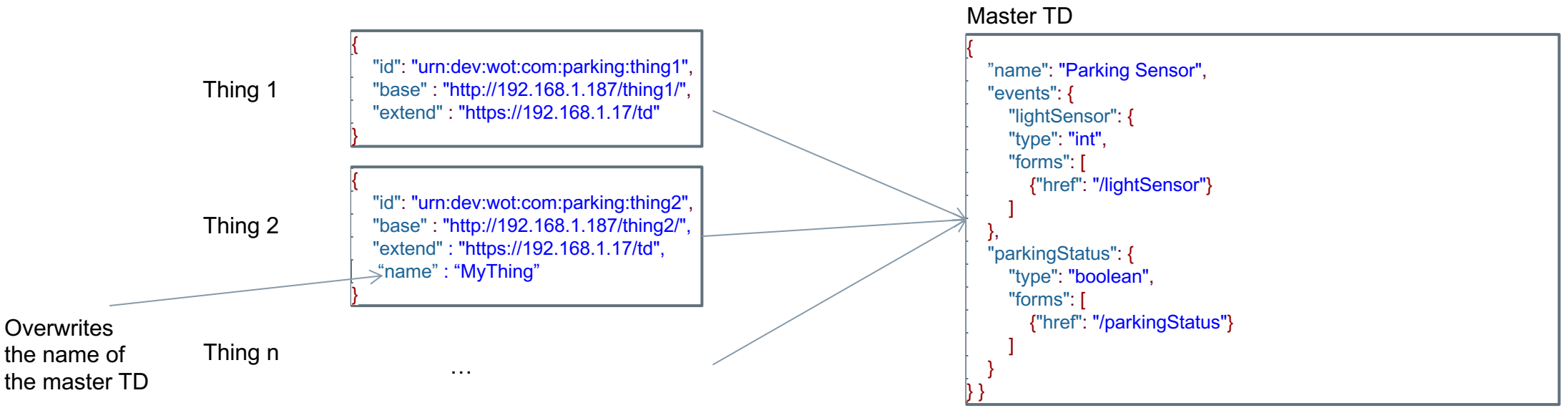
```
{
"@context": "https://www.w3.org/2019/wot/td/v1",
”@type” : “TDT”
}
```

Self-contained TDTs only vs references to other TDTs possible?
Note: Slide from Bundang f2f 2018: Extend TD Template Definition

Also see: https://github.com/w3c/wot-thing-description/issues/168

Lets assume there is a master TD template providing all basic interactions and a huge set of (identical) Things implements these interactions. It would be beneficial to maintain only the master TD and all the Things would inherit the interactions. Thing's TD only extends its individual information (e.g., id, base URI, semantics like locations, etc) or overwrite master TD information.

Use case: Parking slot sensors, room lights, mass production of Things, …



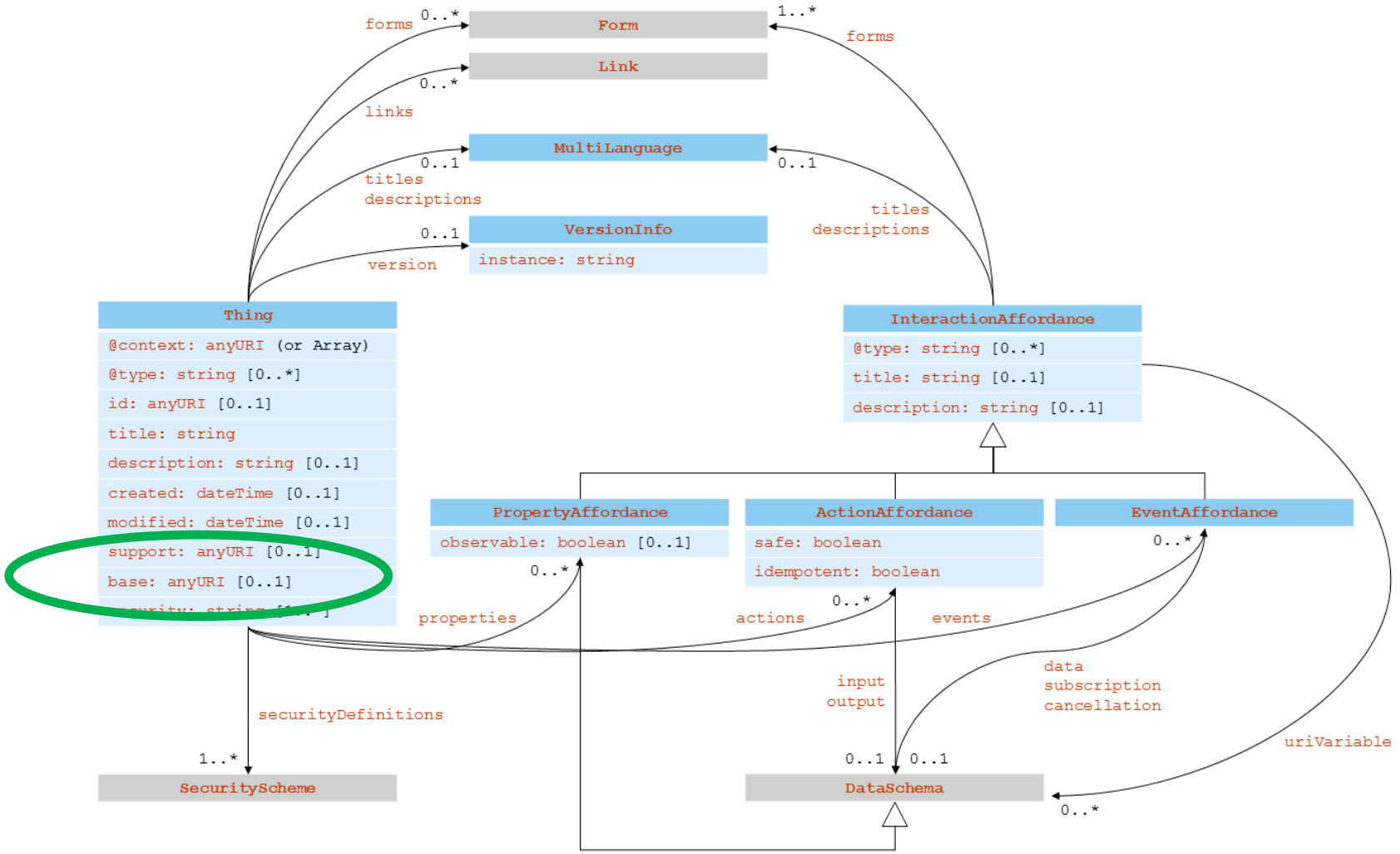
Proposed Next Steps

- Extend TD model with a TDT class
  - Addresses issue https://github.com/w3c/wot-thing-description/issues/416
- Provide a formal definition what is understood by TDT
- Evaluate if we need operations for TDTs like collections, subsetOf or are TDTs always self-contained submitted to the WoT consumers?
  - We should collect concrete use cases?
  - Related issues
    - https://github.com/w3c/wot-thing-description/issues/168
    - https://github.com/w3c/wot-thing-description/issues/490

Describing initial connection

Current Base Term Definition

base is mainly used for address optimization: **hrefs** in interaction affordances can be relative to the base URI → no further 'base' communication information can be provided as we know from **forms** (e.g., subprotocol)



Motivation to have more than a ‘base’ with URI information

- Provide communication information that is valid for all interactions affordances, e.g.,
  - used subprotocol
  - contentType
  - <protocol specific metadata>

  - see issue https://github.com/w3c/wot-thing-description/issues/878

- Improves the design of TDTs
  - Configure protocol settings in a single place instead of redundant definitions in the interaction affordances

  - See discussion https://github.com/w3c/wot-thing-description/issues/803

Examples

WebSocket connections can be announced with a particular subprotocol (e.g., Mozilla's WebThing)

```
"base": {
    "rel": "alternate",
     "href": "wss://mywebthingserver.com/things/lamp",
     "subprotocol": "webthing"
}
```

MQTT connections uses the retain flag and qos=1 for all interactions

```
"base": {
    "href": "mqtt://192.168.1.187:1883",
    ”mqv:options":
      [{
        “mqv:optionName”: “"retain”

      },
      {
        “mqv:optionName”: “”qos”,
        ”mqv:              optionValue”:”1”

      }]}
```

Shall we offer ‘base’ as string and as an array?

- See issue https://github.com/w3c/wot-thing-description/issues/803
- We allow multiple **forms** entries at the interaction level, however, only single **base** term is used
- How about if there are multiple endpoints possible or there are multiple protocols supported by a WoT producer?
- We can introduce the same feature as we have for @context:
  - keep **base** definition as is → take a single URI
  - In addition, allow **base** to be an array of JSON Objects
- Proposal: The JSON Object should be a class of **Form**
- Improves also the TDTs design

Form
op: string [1..*]
href: anyURI
contentType: string
contentCoding: string [0..1]
subprotocol: string [0..1]
security: string [0..*]
scopes: string [0..*]

Proposed Next Steps

- Extend **base** to be array (keep string type to be backwards compatible)
- JSON Object in base array shall be class of **Forms**
- Related issues can be solved:
  - https://github.com/w3c/wot-thing-description/issues/803
  - https://github.com/w3c/wot-thing-description/issues/878

- Efficient formats for TD (Taki & Daniel)

Full OAuth2 Support in TD

We should add again the full OAuth2 support

§ 5.3.3.11 `OAuth2SecurityScheme`

OAuth2 authentication security configuration for systems conformant with [RFC6749] and [RFC8252], identified by the Vocabulary Term `oauth2` (i.e., `"scheme": "oauth2"`). For the `implicit` flow `authorization` *MUST* be included. For the `password` and `client` flows `token` *MUST* be included. For the `code` flow both `authorization` and `token` *MUST* be included. If no `scopes` are defined in the `SecurityScheme` then they are considered to be empty.

Vocabulary term	*Description*	*Assignment*	*Type*
`authorization`	URI of the authorization server.	optional	`anyURI`
`token`	URI of the token server.	optional	`anyURI`
`refresh`	URI of the refresh server.	optional	`anyURI`
`scopes`	Set of authorization scope identifiers provided as an array. These are provided in tokens returned by an authorization server and associated with forms in order to identify what resources a client may access and how. The values associated with a form should be chosen from those defined in an `OAuth2SecurityScheme` active on that form.	optional	`string` or Array of `string`
`flow`	Authorization flow.	mandatory	`string` (one of `implicit`, `password`, `client`, or `code`)

Sources: https://www.w3.org/TR/2019/CR-wot-thing-description-20191106/#oauth2securityscheme