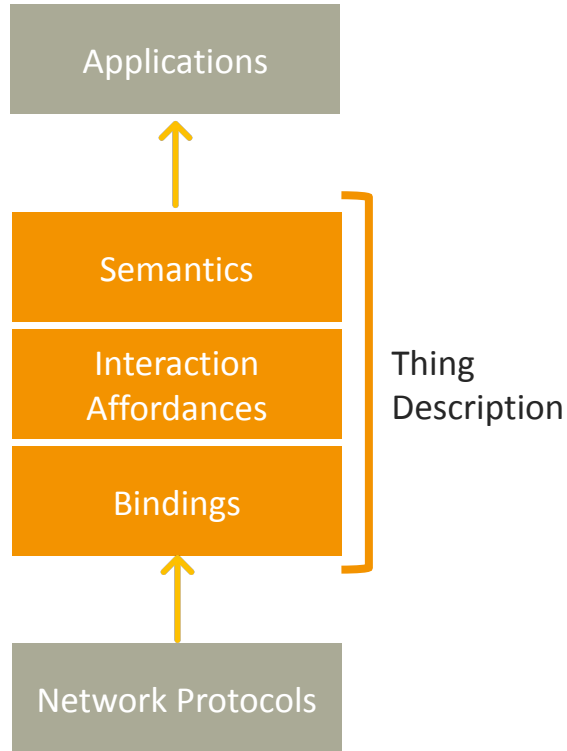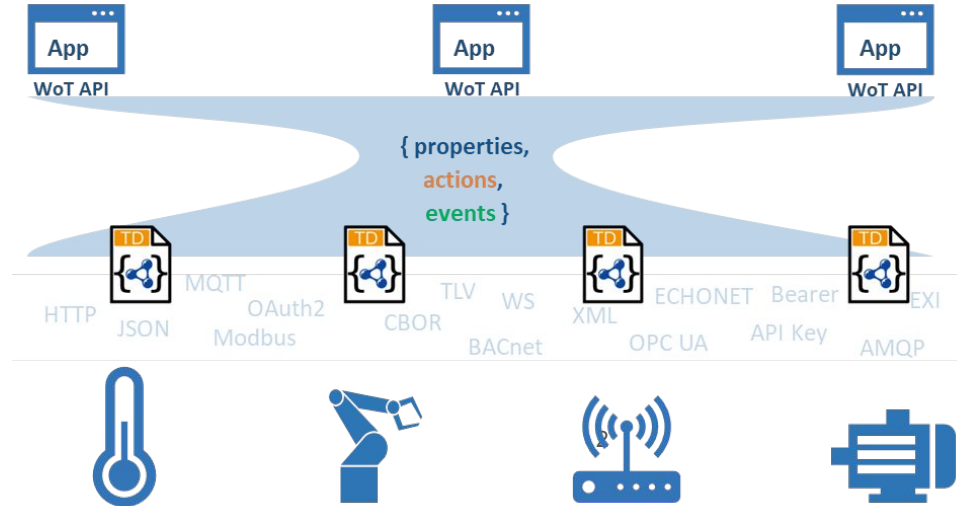# WoT Thing Description Single Source of Truth

Ege Korkan
W3C TPAC Breakouts

Web of Things, not a Protocol
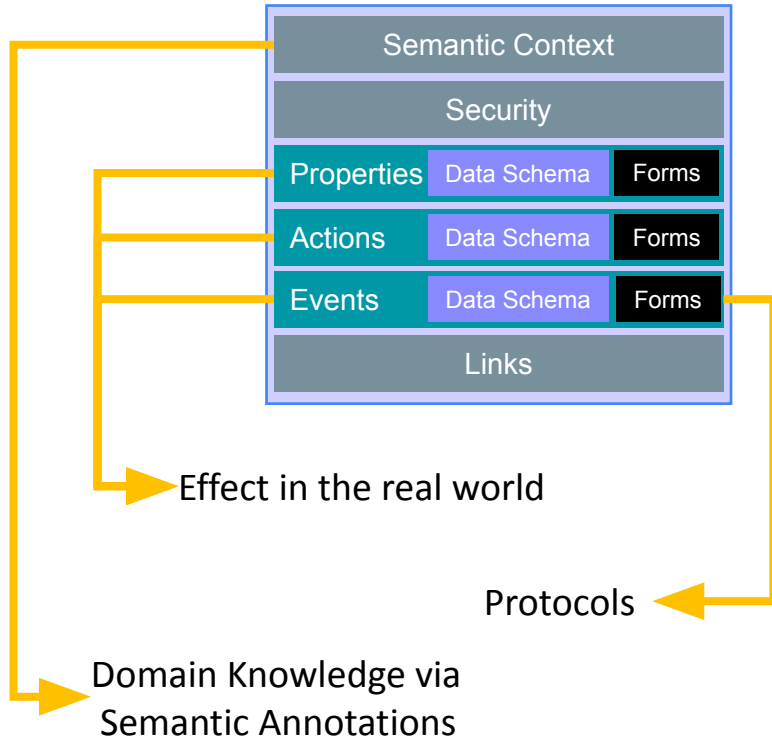
Core Specification: Thing Description (TD)

Semantic Context

Security

Properties | Data Schema | Forms

Actions | Data Schema | Forms

Events | Data Schema | Forms

Links

Effect in the real world

Protocols

Domain Knowledge via Semantic Annotations

```
{
  "@context": "https://www.w3.org/2019/wot/td/v1",
  "id": "urn:HotelRoom",
  "@type": "Thing",
  "base": "coap://localhost:3000",
  "title": "simulated Hotel Room",
  "properties": {
    "brightness": {
      "type": "integer",
      "title": "Light Brightness",
      "forms": [
        {
          "href": "/light/Brightness",
          "contentType": "application/cbor",
          "op": [
            "observeproperty",
            "readproperty",
            "writeproperty"
          ]
        }
      ]
    },
```

JSON-LD Serialization

Discovery and Usage by Clients

Why is the session relevant for WoT

Example of a T

```json
{
    "@context": "https://www.w3.org/2022/wot/td/v1.1",
    "id": "urn:uuid:0804d572-cce8-422a-bb7c-4412fcd56f06",
    "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"
            }]
        }
    }
}
```

Example of a TD Vocabulary Term

```
"security": "basic_sc",
```

It is not just a string

EXAMPLE 15

```
{
    // ...
    "securityDefinitions": {
        "proxy_sc": {
            "scheme": "digest",
            "proxy": "https://portal.example.com/"
        },
        "bearer_sc": {
            "scheme": "bearer",
            "in": "header",
            "format": "jwt",
            "alg": "ES256",
            "authorization": "https://servient.example.com:8443/"
        }
    },
    "security": ["proxy_sc", "bearer_sc"],
    // ...
}
```

Where are those terms defined?

In many places…

security	Set of security definition names, chosen from those defined in `securityDefinitions`. These must all be satisfied for access to resources. *index.html / Human Readable*	mandatory	string or Array of string But wait! This is not the definition!

```
sh:property [
    sh:path td:hasSecurityConfiguration ;
    skos:definition """Set of security definition names, chosen from
            those defined in <code>securityDefinitions</code>.
            These must all be satisfied for access to resources."""^^rdf:HTML ;
    sh:datatype xsd:string ;
    sh:minCount 1 ;
    sh:order 15 ;
] ;
```

This is used to generate the index.html

https://github.com/w3c/wot-thing-description/blob/main/validation/td-validation.ttl#L144

Not done

Official JSON
Schema

```
29          "security": {
30            "oneOf": [
31              {
32                "type": "array",
33                "items": {
34                  "type": "string"
35                },
36                "minItems":1
37              },
38              {
39                "type": "string"
40              }
41            ]
42          },
```

Test Case for Implementation Report Generation

Based on the official JSON Schema

```
 79        "form_element": {
 80            "type": "object",
 81            "if": {
 82                "properties": {
 83                    "security": {
 84                        "oneOf": [
 85                            {
 86                                "type": "array",
 87                                "items": {
 88                                    "type": "string"
 89                                }
 90                            },
 91                            {
 92                                "type": "string"
 93                            }
 94                        ]
 95                    }
 96                },
 97                "required": ["security"],
 98                "additionalProperties": true
 99            },
100            "then": {
101                "const": "td-vocab-security--Form=pass"
102            }
103
```

https://github.com/eclipse-thingweb/playground/blob/master/packages/assertions/assertions-td/td-vocab-security--Form.json

Going to published versions

TD Context

```
"security": {
  "@id": "td:hasSecurityConfiguration",
  "@type": "xsd:string",
  "@context": {
    "td": "https://www.w3.org/2019/wot/td#",
    "jsonschema": "https://www.w3.org/2019/wot/json-schema#",
    "wotsec": "https://www.w3.org/2019/wot/security#",
    "hctl": "https://www.w3.org/2019/wot/hypermedia#",
    "dct": "http://purl.org/dc/terms/",
    "rdf": "http://www.w3.org/1999/02/22-rdf-syntax-ns#",
    "@vocab": "https://www.w3.org/2019/wot/security#",
    "in": {
      "@id": "wotsec:in"
    },
    "name": {
      "@id": "wotsec:name"
```

Going to published versions

TD Ontology in TTL
and HTML

```
140
141     :hasSecurityConfiguration rdf:type owl:ObjectProperty ;
142         rdfs:label "hasSecurityConfiguration" ;
143         rdfs:comment "A security configuration is a a security scheme applied to a (set of) affordance(s)."@en ;
144         schema:domainIncludes :Thing, hctl:Form ;
145         rdfs:isDefinedBy <https://www.w3.org/2019/wot/td> .
146
```

https://github.com/w3c/wot-thing-description/blob/main/ontology/td.ttl#L141

§ 3.2.14 hasSecurityConfiguration

IRI: https://www.w3.org/2019/wot/td#hasSecurityConfiguration

A security configuration is a a security scheme applied to a (set of) affordance(s).

https://www.w3.org/2019/wot/td#hasSecurityConfiguration

| Domain includes | hctl:Form |
| | td:Thing |

Final Thoughts

- This of course applies to all terms defined in TD
- We wish to change only one line of "code" if we ever change "security" to "sec"