WoT, JSON-LD, RCH Joint Session

W3C TPAC 2023

Before we start!

- This sessions is part of W3C TPAC 2023, which means that a registration is necessary. Please leave if you have not registered for the session
- All observers, i.e. not members of the WGs, are allowed to participate in the WoT CG meetings
- These meetings follow the *Positive Work Environment at W3C: Code of Ethics and Professional Conduct* to ensure a healthy environment. Read more here.
- We follow COVID regulations set by the organizers. More in the next slide or here.
- Automatic Captions should be enabled

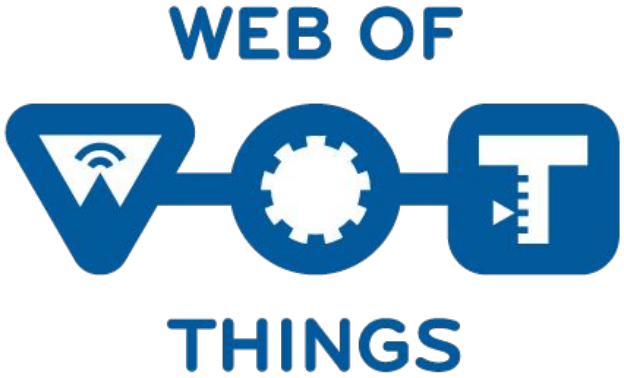
Health rules reminder

- Take a covid test each day before attending any in-person event
- Do not come to the meeting if your test is positive
- Masks must be worn at all times in all common spaces and meeting rooms
- Masks must cover the nose and mouth
- Masks can be removed only as necessary to consume food and beverages (Food forbidden in meeting rooms and only allowed in the dedicated space)
- Masks must be absolutely worn while speaking

Find all the rules at: www.w3.org/2023/09/TPAC/health

Agenda

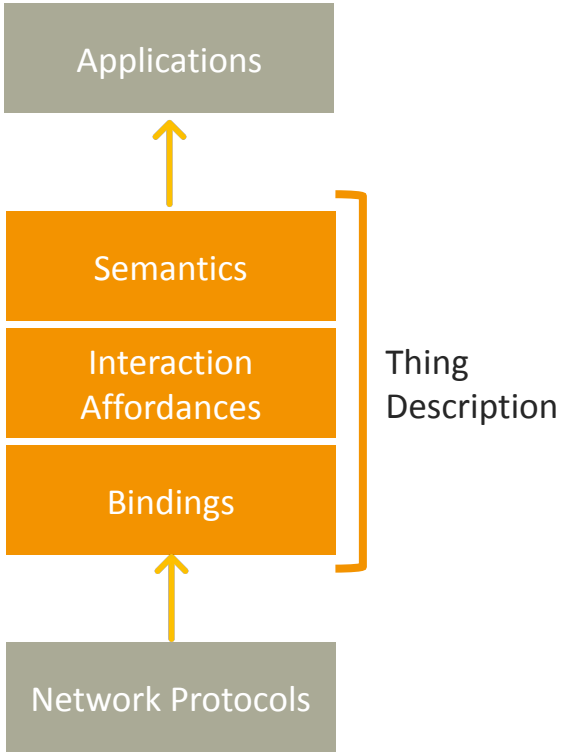
- Introduction of Participants (10 mins)
- Short Introduction of WoT (10 mins)
- Signing and Canonicalization (30 mins)
- Degraded Consumption (20 mins)
- Additional Serialization (15 mins)
- Linting (15 mins)

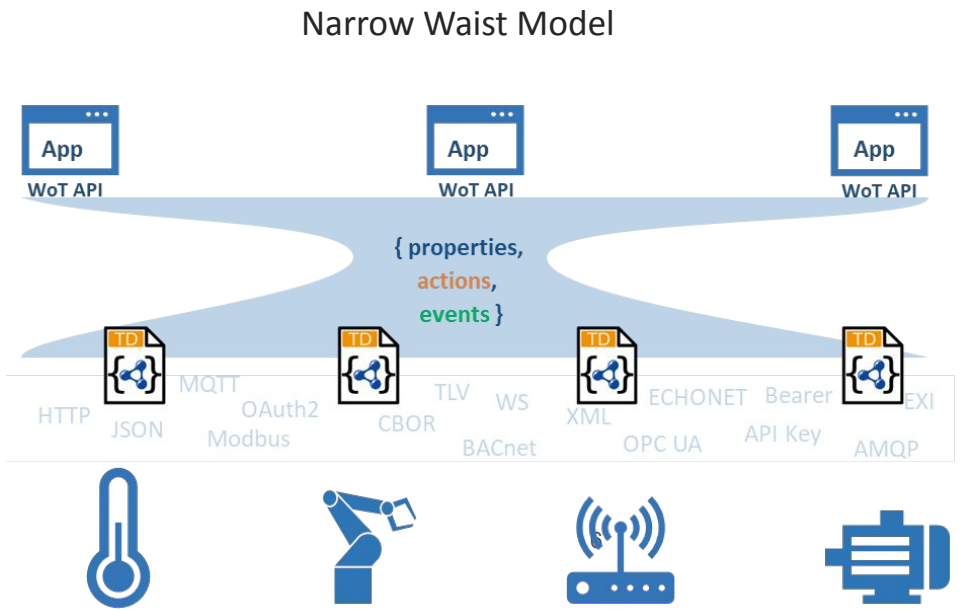


Short Introduction to WoT

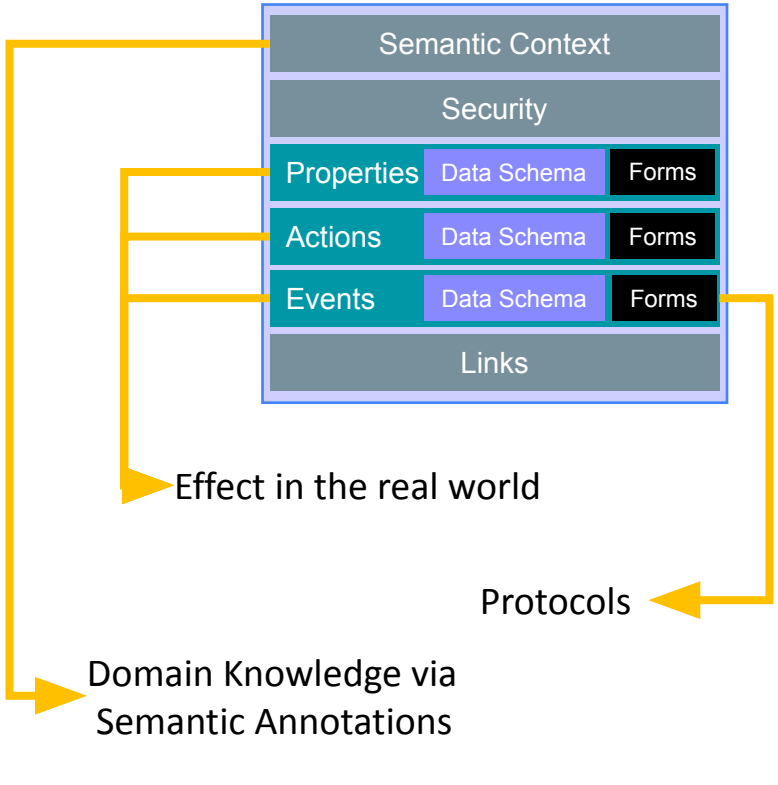
A longer tutorial is offered on Thursday

Web of Things, not a Protocol





Core Specification: Thing Description (TD)




```
{
  "@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

Other Specifications

REC

Architecture

Common Reference, Deployment Patterns, Relationship between specifications

REC

Discovery

How to find TDs in a network, TD management via a REST API

Binding Templates

Extending TDs for different protocols, media types, ecosystems, etc.

Scripting API

Programming API for building Consumer and Thing applications

Discovery


Brand new specification!
§ 6. Introduction Mechanisms
§ 7. Exploration Mechanisms
A link to a TD or Directory
Searching for TDs in a Directory

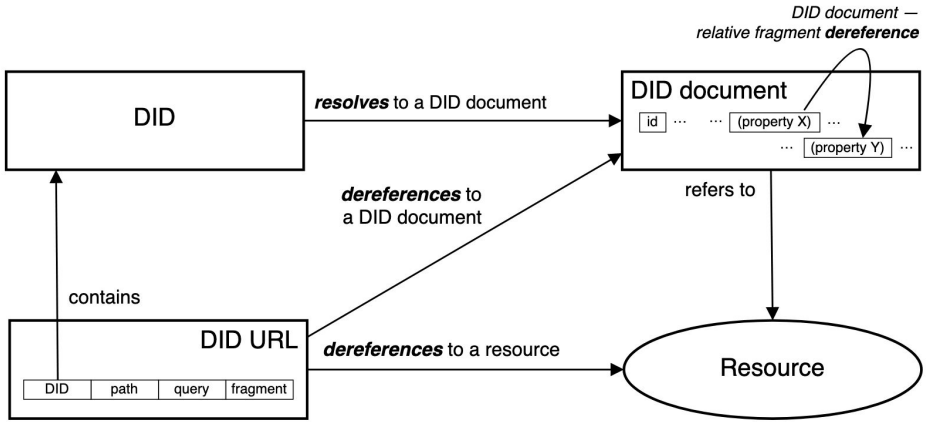
Discovery: Exploration

A Thing Description Directory is an HTTP API specification for managing TDs

Main features:

- Creation, Retrieval, Update, Deletion, Listing (CRUDL) of TDs
- Searching TDs
  - JSONPath
  - XPath
  - SPARQL
- Eventing (CUD of TDs) via Server Sent Events

Discovery: Introduction via DID



Decentralized Identifiers is a *new* W3C REC that allows verifiable and decentralized digital identities

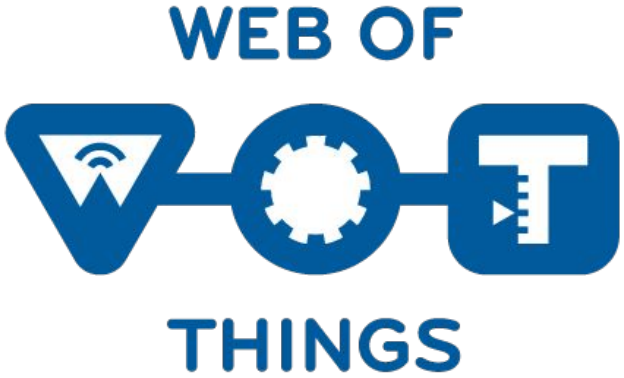
EXAMPLE 1: A Example Service Endpoint in a DID Document - WotThing

```
{
    "service": [{
        "id": "did:example:wotdiscoveryexample#td",
        "type": "WotThing",
        "serviceEndpoint":
            "https://wot.example.com/.well-known/wot"
    }]
}
```

EXAMPLE 2: A Example Service Endpoint in a DID Document - WotDirectory

```
{
    "service": [{
        "id": "did:example:wotdiscoveryexample#tdd",
        "type": "WotDirectory",
        "serviceEndpoint":
            "https://wot.example.com/tdd"
    }]
}
```

Discussions



Signing and Canonicalization

(Mahda Noura)

Canonicalization and Signing: Context

- **Goal:** Enhance the current WoT TD standard by designing the technical means for enabling trust and integrity of TD documents as JSON-LD
- Existing work in WoT WG:
  - Old version containing canonicalization: https://www.w3.org/TR/2021/WD-wot-thing-description11-20210607/#canonicalization-serialization-json
  - Removal of content: https://github.com/w3c/wot-thing-description/pull/1304
  - Use of DIDs in Discovery: https://w3c.github.io/wot-discovery/#introduction-did-sec

TD Canonicalization & signing

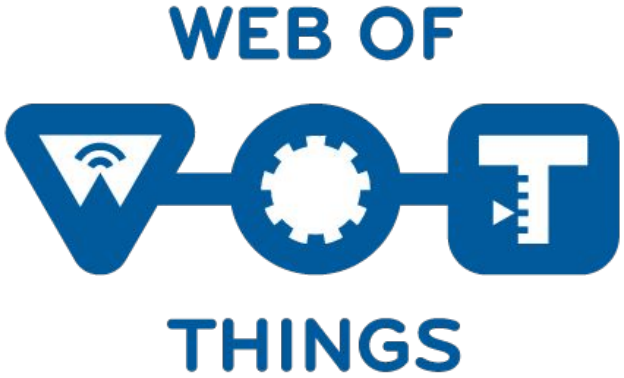
- To verify a digital signature the entire document needs to be shared. How to validate a **selective disclosure** of information rather than an entire credential
- How to ensure the **integrity** of **linked data** that lies outside of the protection of the proof, e.g. LD @context content and links section in TDs
  - HashLinks (https://github.com/w3c-ccg/hashlink) - IETF draft specification
  - IPFS (InterPlanetary File System) links - decentralized file
- Which **signature schemes** are appropriate for TD?
  - Linked Data Signatures spec for JSON-LD: JS library of *json-ld signature by Digital Bazaar (https://github.com/digitalbazaar/jsonld-signatures)*
  - *jsonld-signatures-bbs: JS library that extends json-ld signatures library, for using BBS+ signature suite*
- Default Values
  - Canon. before or after injecting default values
  - Attack model needs to be understood to decide if injecting is necessary

TD Canonicalization & signing

- How to approve the generated Verifiable Credentials?
  - How does a verifier know that the received VC have been issued by an **entitled entity** or an **authorized** for that type of credential, e.g., a diploma can only be issued by a university and what about TMs?
  - A TM can be issued by the manufacturer but TD is only after instantiation. Who issues the TD? Thing itself, directory? -> (cf. point 5 roles)
  - Prove the credential was really issued not later than the given time point in the **“Issued at”** field of credential

TD Canonicalization & signing

- Mapping of the **actors** in the VC to the WoT Ecosystems: Subject, Issuer, Holder, & Verifier
  - Who is the **trustable** entity in this ecosystem?
  - Verifiable Data Registry is required to implement the trust model. In spec examples are: t*rusted DBs, decentralized DBs*, *distributed ledgers*, what is suitable for WoT?
- Canonicalization implications for pure JSON processors in WoT Consumers (cf. slide 19)



Degraded Consumption

(Luca Barbato)

Degraded consumption of TDs

Right now the TD spec is not very explicit on what is the expectation on the Consumer

- Failing to parse any element of the TD **invalidates** the whole?
- If a Form or an Affordance contains vocabulary elements **unknown** to the Consumer should only that element be **excluded**?
- Unknown vocabulary elements shall be **ignored** and the, potentially incomplete, Forms shall be used nonetheless?

Degraded consumption of TDs

Which consumer needs what part of TDs?

- Being unaware of a protocol-binding **prevents** the consumer from using a Form at all potentially
- Being **unaware** of risk-labels or artwork elements does **not** prevent the consumer from using an Affordance
- Being **unaware** of custom elements in a data schema may or may not prevent the consumer from exchanging messages or validating them, but it could always forwarding them verbatim.

Degraded consumption of TDs

We can split vocabulary terms in two groups:

- **Structural** needed to consume or correctly consume a Thing
  - Some are always necessary:
    - **Protocol bindings, security schemes, …**
- **Auxiliary** providing information can be ignored
  - Depending **on the task** at hand some terms can be structural
    - **Geolocation**, **Risk/Hazard annotations**, **Currency**

We should make sure that **structural** vocabulary terms can be consumed by the widest kinds of consumer.

Degraded consumption of TDs

A Thing Description is a JSON-LD document, but Consumers may use plain JSON processors.

- A plain JSON parser **cannot parse** the @context
  - But even a plain JSON consumer can recognise a @context with **canonical** prefixes and well-known uris.
- A plain JSON parser **cannot** apply any transformation (e.g. term expansion)
  - A consumer using a plain JSON parser can consume descriptions only by hardwiring vocabulary terms with canonical prefixes.
- A plain JSON parser **cannot** do any JSON-LD specific validation
  - But may validate using different means nonetheless (e.g. JSON Schema)

Degraded consumption of TDs

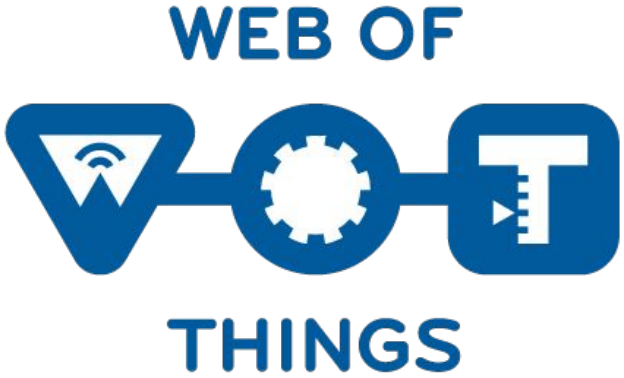
Non JSON-LD Consumer consuming a TD may be able to fully consume the information as long it:

- Comes with a **canonical** context with **fixed** prefixes it is aware of
  - pretty much as it does with URI schemas.
  - no parsing at all of the actual JSON-LD context is required: the vocabularies are pre-compiled in.
- Is a compact document
  - No JSON-LD types used in ways the parser does not expect
  - No renaming/reprefixing of vocabulary terms

Existing Mechanisms in JSON-LD and VC

Known prior arts

- JSON-only processing https://w3c.github.io/vc-data-model/#json-processing
- Known URLs: https://w3c.github.io/vc-data-model/#vocabularies



Other Serialization Formats

(Mahda Noura, Ege Korkan)

Motivation

- TD Information Model is not limited to JSON, thus not to JSON-LD.
- Having more formats makes it better for adoption (some people don’t like JSON)
- There are some built-in advantages of other formats:
  - YAML-LD allows comments
  - CBOR-LD saves space

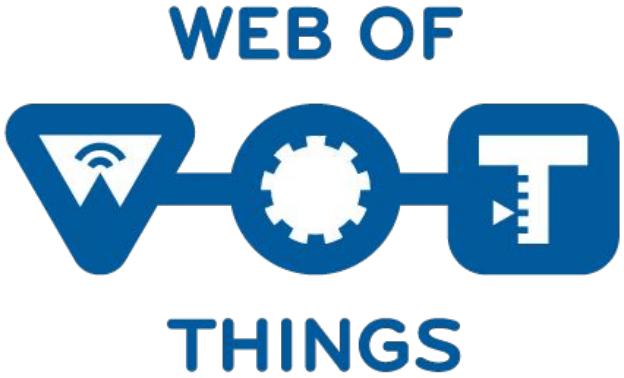
Existing Work

There are some work on YAML-LD and CBOR-LD:

- CBOR-LD: https://digitalbazaar.github.io/cbor-ld-spec/ and slides
- YAML-LD: https://json-ld.github.io/yaml-ld/spec/
  - We are also testing YAML-LD (http://plugfest.thingweb.io/playground/)

Questions

- What is the experience with other formats as a whole? What is the adoption?
- YAML-LD & CBOR-LD canonicalization for cryptographic signing and VC's
  - YAML provides canonicalization for scalar only (https://yaml.org/spec/1.2.2/#canonical-form)
  - Is there canonicalization algorithms available for YAML?



Linting

(Ege)

Motivation

API Description Linting is a recent topic of relevance (see https://stoplight.io/open-source/spectral) and since TDs are API descriptions of IoT devices, this topic will become important for us.

Examples

- Each affordance should have a title
- TD should contain an MQTT form for each affordance
- All affordances should contain an annotation to schema.org ontology

Questions

- Are there any mechanisms within JSON-LD?
  - E.g. tool: Structural Data Linter, schema.org site has
  - Safe mode in JSON-LD processors to reject undefined terms
  - VC has a credentialSchema keyword.
- Experience with linting rules for JSON-LD documents?
- Any interest from the community?
  - Playground has some warnings, safe mode toggle

- Overall JSON Schema, SHACL

Meeting Minutes

https://www.w3.org/2023/09/11-wot-minutes.html