Thing Description, Profiles and Bindings

TPAC 2024 - WoT WG

Ege Korkan, Michael Koster, Luca Barbato

These slides are public. Link.

Agenda

- Overall Progress (10 mins + 5 mins Q&A)
- Use Cases Extraction from Issues (25 mins)
- Relationship between Bindings and Profiles (25 mins)
- Reports from Breakouts and JSON-LD Meeting (10 mins)

Overall Progress

Overall Progress

- Less features and more “getting ready”
- Under the hood
  - Resource Management (wot-resources)
  - Errata Handling
  - Toolchain and LinkML
- Refactoring
  - Binding Templates Core -> TD
- Management and Organization
  - Project Management
  - Resource Versioning Discussions
  - Registry Discussions

- Feature Discussion:
  - **Reusable Connection Container**
  - Data Mapping
  - Degraded Consumption / Gradual Enhancement
- Bindings:
  - Improved CoAP and Modbus Bindings
  - New BACnet Binding
- WoT Week Targets:
  - Reusable Connection Container
  - Bindings + OPC UA Binding First Version
  - Untested features of TD like additionalResponses etc.

Use Case Extraction from Issues

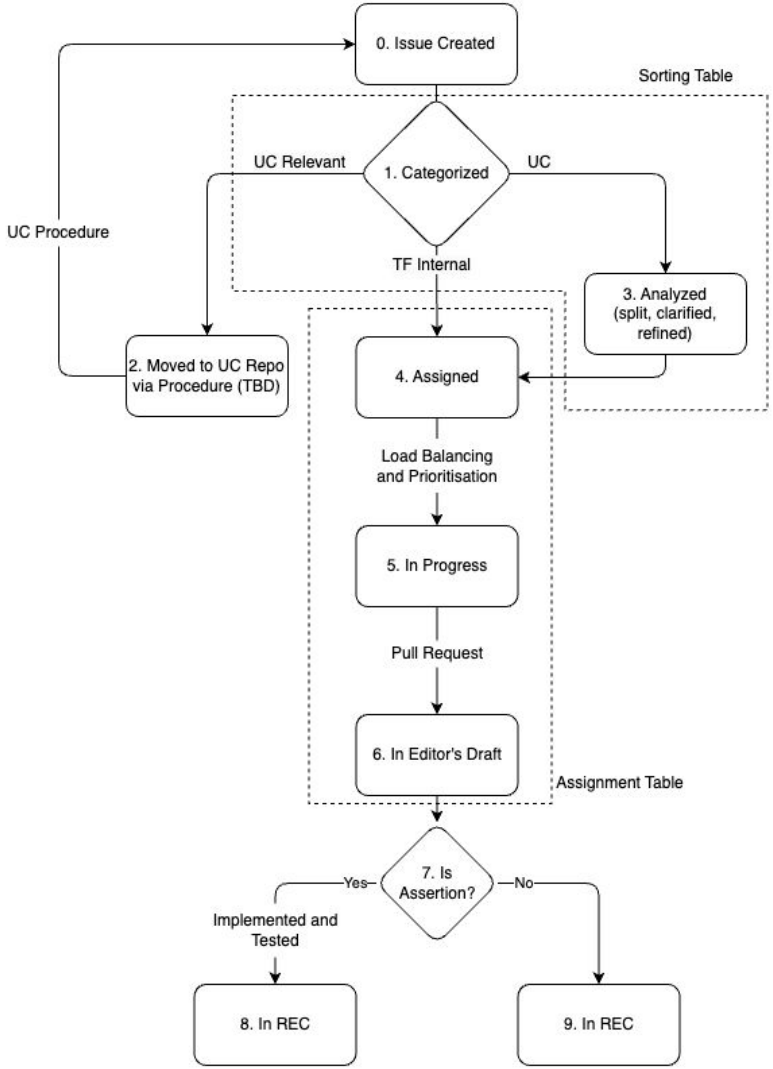
Use Case Extraction from Issues

Idea: We have received many issues to TD and Binding Repositories, which are significant enough to be treated as use cases. So let’s do that.

Already Done: Sorting almost 300 issues to find the “worthy” ones as whole TF

- Link to worthy TD Issues
- Link to worthy Binding Issues

Handling future issues



Three Use Case Extractions Trials

Goal: Test the new use case template as the TD TF to formulate use cases from issues in the TD repository

- Daniel Peintner:
  - Writing to a Property Request returns a response
  - Link
- Jan Romann:
  - Simplifying security descriptions
  - Link
- Luca Barbato:
  - Profiles for converging interoperability
  - Link

danielpeintner commented on Aug 8 • edited Member ···

> - Should it be possible to indicate whether writing a property returns set value? #875 @danielpeintner

Find below a first try to fill out the *new* use case template.

What I *missed* while filling out the template:

- Name/title of the concern/matter
- Possible disadvantages / concerns (we just list advantages)

- Submitter Contact Information: -
- Domain or Industry: -

Introduction

At the moment the TD does not provide a way to indicate whether writing a property returns a value.

There are many possible reasons why writing a property may return a value, e.g.,:

- the set value might be (slightly) different (e.g., rounding a number 1.2345 to 1.23 and notify that the provided value is not supported w.r.t. to digits)
- avoid reading the value again (after writing). Note: re-reading is *different* anyway since someone else might have written another value in the meantime.
- setting a value for Philips HUE lights returns a completely different payload (1)
- Pretty common to return a body in HTTP-based APIs (2)

Note: There are also scenarios (maybe most of them) were "no" value is returned when writing. Hence, we should not forget about that *common* use-case.

WoT Usage

Thing Descriptions are used.

Technical Environment

HTTP, CoAP, Philips HUE, ..

Problem

At the moment the TD specification misses to describe what is "practice" in many environments.

Expectation

The TD should allow do describe what is used/done in practice.

Solution Proposal

Add `response` term to properties similar to `response` in forms. The response might be fully missing or defining a completely different payload as sketched in the example below.

```
"properties": {
        "value": {
                "type": "number",
                "forms": [{"href": "https://example.com/value"}],
                "response": {
                        "contentType": "application/json",
                        "type": "boolean"
                }
        }
}
```

JKRhb commented on Aug 16 · edited Member

Below you can find a potential submission of the form for the use cases described in #757 and #300.

Besides the points mentioned by **@danielpeintner** above, I was also wondering if we should add a form field for implementation experience, e.g., if a submitter has already implemented a proposed feature because they needed it to cover the use case they are describing.

- Submitter Contact Information: –
- Domain or Industry: No specific domain/applicable to all domains

Introduction

At the moment, TDs require both a `security` and a `securityDefinitions` field to be defined, even when there are no security mechanisms in place.
In that case, a `NoSecurityScheme` or a reference to it has to be added to both fields, which mainly serves as an incentive to have some kind of security mechanism in place instead of "no security".
However, as there are scenarios where, for example, a plain HTTPS connection without authorization is the intended way of communicating with a Thing (e.g., in the context of public places or when using a Proxy), the way the feature is currently specified only introduces boilerplate and does not contribute to an substantial increase of security.

The mandatory split into `security` and `securityDefinitions` also creates some boilerplate that could be avoided by allowing for the "inlining" of security definitions, i.e., by defining them directly within the `security` field and setting them as the default security mechanisms.

Both changes together can make TDs less verbose and more compact, which not only makes them easier to read but also more efficient when it comes to their size on the wire.

WoT Usage

TDs are being used.

Technical Environment

This use case applies to most protocols, devices, data types, and standards, if not all.

Problem

- Expressing that no security mechanisms are being used without unnecessary boilerplate
- Reducing boilerplate for cases, where security definitions are actually being used

Expectation

- `security` and `securityDefinitions` should not be mandatory anymore
- `security` should allow for an inline definition of security schemes, instead of only containing references to the separately defined `securityDefinitions`

Solution Proposal

The potential solution for this problem is two-fold.

Make both fields optional in the WoT TD specification

That would allow for the definition of this minimal TD:

```
{
  "@context": "https://www.w3.org/wot/td/next",
  "title": "Minimalistic Thing"
}
```

Introduce a way to define `securityDefinitions` within `security` WoT TD specification

A minimalistic solution to the second part of the problem could look like this:

```
{
  "@context": "https://www.w3.org/wot/td/next",
  "title": "Minimalistic Thing",
  "security": {
    "nosec_sc": {
      "scheme": "nosec"
    }
  }
}
```

Here, the `security` member defines both a `NoSecurityScheme` and sets it as the default security mechanism with the key `nosec_sc`.
The example above is equivalent to following TD:

```
{
  "@context": "https://www.w3.org/wot/td/next",
  "title": "Less minimalistic Thing",
  "securityDefinitions": {
    "nosec_sc": {
      "scheme": "nosec"
    }
  },
  "security": "nosec_sc"
}
```

Note that in the case of inlined security definitions, there could also be a separate `securityDefinitions` field defined in addition to an inlined `security` member, like so:

```
{
  "@context": "https://www.w3.org/wot/td/next",
  "title": "Minimalistic Thing with additional securityDefinitions",
  "securityDefinitions": {
    "basic_sc": {
      "scheme": "basic"
    }
  },
  "security": {
    "nosec_sc": {
      "scheme": "nosec"
    }
  }
}
```

The TD specification should put assertions in place that disallow the redefinition of keys in cases like this one or define which security definition should have the higher priority (probably the ones defined in `security`).

Note that inline security definitions could also be created within the `security` field of form instances (e.g., within the form of a property), in a similar way as outlined for the top-level `security` field above.

lu-zero commented 5 days ago • edited Member

Submitter Contact Information

@lu-zero

Domain or Industry

Profile TF experiment

Introduction

Profiles focus on giving `out-of-box` interoperability: a Consumer supporting a profile can use every and all the capabilities a Thing supporting the same profile exposes.
A Thing using a binding protocol cannot signal that a Consumer supporting the same binding protocol can use all the capabilities available through it.

This Use Case focuses on using profiles to have vendors converge on them, reduce the overall iot fragmentation and reduce the effort required to write generic consumers.

WoT Usage

Profiles restrict what is described in a TD by defining the scope of the extension points (e.g. only a set of bindings can be used and it MUST cover all the Things affordances)

Technical Environment

Profile-dependent

Problem

```
Thing Description main aim is to be able to faithfully describe pre-existing devices and foster interoper i

This is a leap compared to the alternative of having to build clients accoding to documentation that might not

Profiles can solve this problem.
```

Expectation

The profiles registry would contain greenfield profiles that should provide a blueprint to use in an efficient way a set of protocols described by their bindings and restrict them so it is easy to write consumer and things supporting it.

Ideally at most a profile per (sub)protocol should exist

Solution Proposal

The current Profiles 1.0 are close to the philosophy described above

Comments

This is an experiment to test the UC template and brainstorm regarding the scope of Profile.

Overall Experience with the Process

We need to be able to extract the scenario, user story. Something more high-level to motivate the work on it. We should hold back the technical solution in the first level or step.

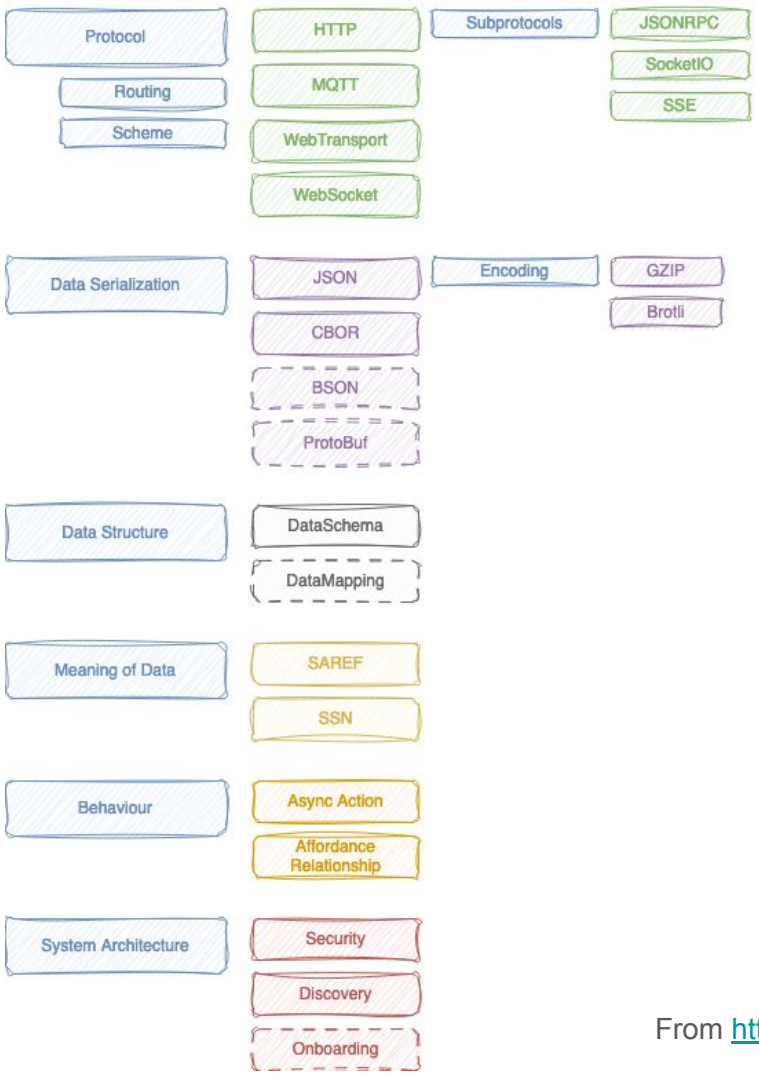
Slides from McCool

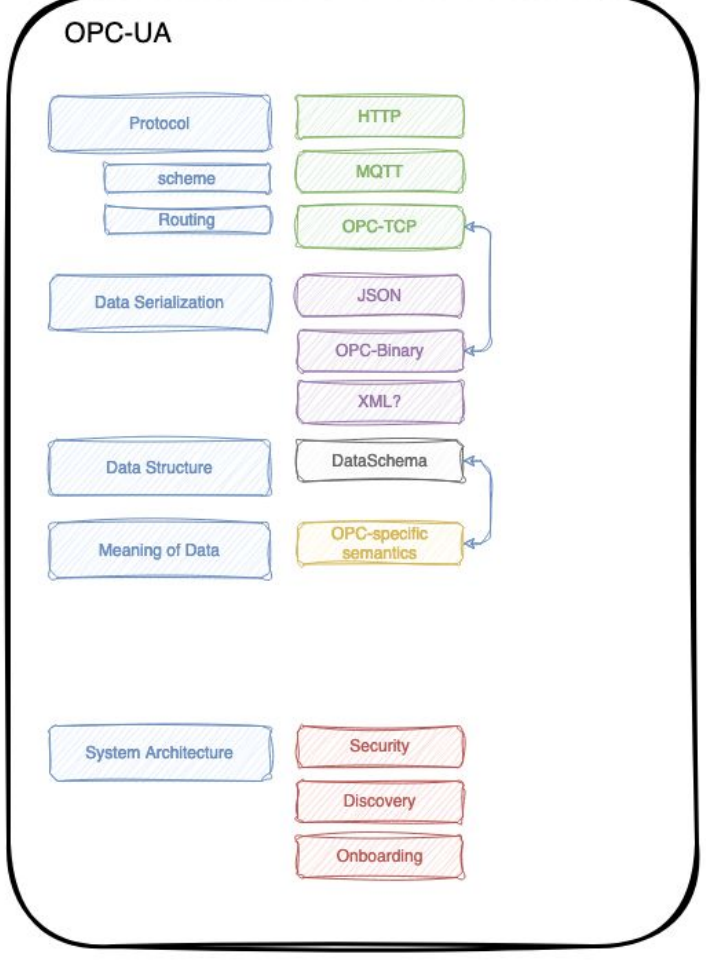
Relationship between Bindings and Profiles

Profiles outside of WoT: Bluetooth

- In order to use Bluetooth, a device must be compatible with the subset of Bluetooth profiles (often called services or functions) necessary to use the desired services. A Bluetooth profile is a specification regarding an aspect of Bluetooth-based wireless communication between devices.
- Examples:
  - Headset
  - Human Interface Device: mouse, keyboard, etc.
  - Health Device Profile
  - Generic Attribute Profile: discovery related

How much can we constrain?





From https://github.com/w3c/wot-profile/issues/416

Relationship between Bindings and Profiles

- Profiles provides "out of box" interoperability warranties
  - It restricts what a compliant TD can use regarding Bindings (and other extension points)
  - Even it can restrict to subsets of a Binding (Meeting: Need to define what means a restriction and what is a subset and provide examples. How much extra information or ontology is allowed?)
- Bindings extend the TD, Profiles restrict the TD
  - Not the core vocabulary
- Open questions
  - Semantic interoperability included?
    - E.g. Temperature is a number? With which Unit?
  - Profiles as way to "onboard" external ecosystems? (when just a protocol is not enough)
    - E.g. Matter Profile for WoT, OPC UA Profile etc.
  - Profiles as a way to enforce some way to converge on a subset of protocols?
    - E.g. Greenfield profiles targeting newcomers

Reports from Other Meetings

Report from LinkML Breakout

- https://www.w3.org/events/meetings/e5cb0ff4-1696-4367-b9fe-9e74c5034dd3
- Not very good attendance
- Overall good feedback and appreciation of work
- Other SDO Groups, such as Electrical CIM/CGMES, are transitioning to LinkML
- Further meetings with LinkML Community is being planned

Report from Registry Breakout

- https://www.w3.org/events/meetings/bf75e0c9-c534-4ccf-988e-0e720f5280ff/
- Not very good attendance
- Florian Rivoal, co-editor of the process document, was there. Recommended opening issue at the Process CG to find a place for the best practices and tag him, fantasai (other editor) and PLH.
  - https://github.com/w3c/Guide/issues/247
- Our feedback that the process is too open is by design
- No changes are planned for the registry track

Report from JSON-LD Meeting

- https://www.w3.org/2024/09/26-wot-minutes.html#t11
- JSON-LD update to include RDF-Star
- CBOR-LD will use a registry of contextes to compress
- YAML-LD will be done but it is a superset of JSON anyways. No conversion of comments
  - Cris: There is commentable JSON used in places like VS Code.
  - Luca: https://json5.org/ is an alternative but not a standard.
- Canonicalization discussion: Any decision etc? -> None
- @type issue needs a follow-up with Mahda
- JSON Schema discussion: Any decision etc?
  - From W3C PoV: If the standard is stable enough, it can be normatively referenced. What is stable enough?
  - VC JSON Schema is in CR phase.