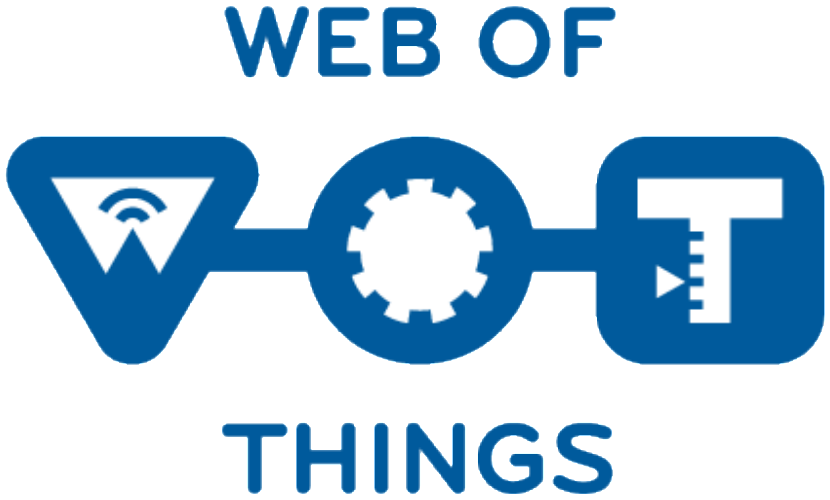


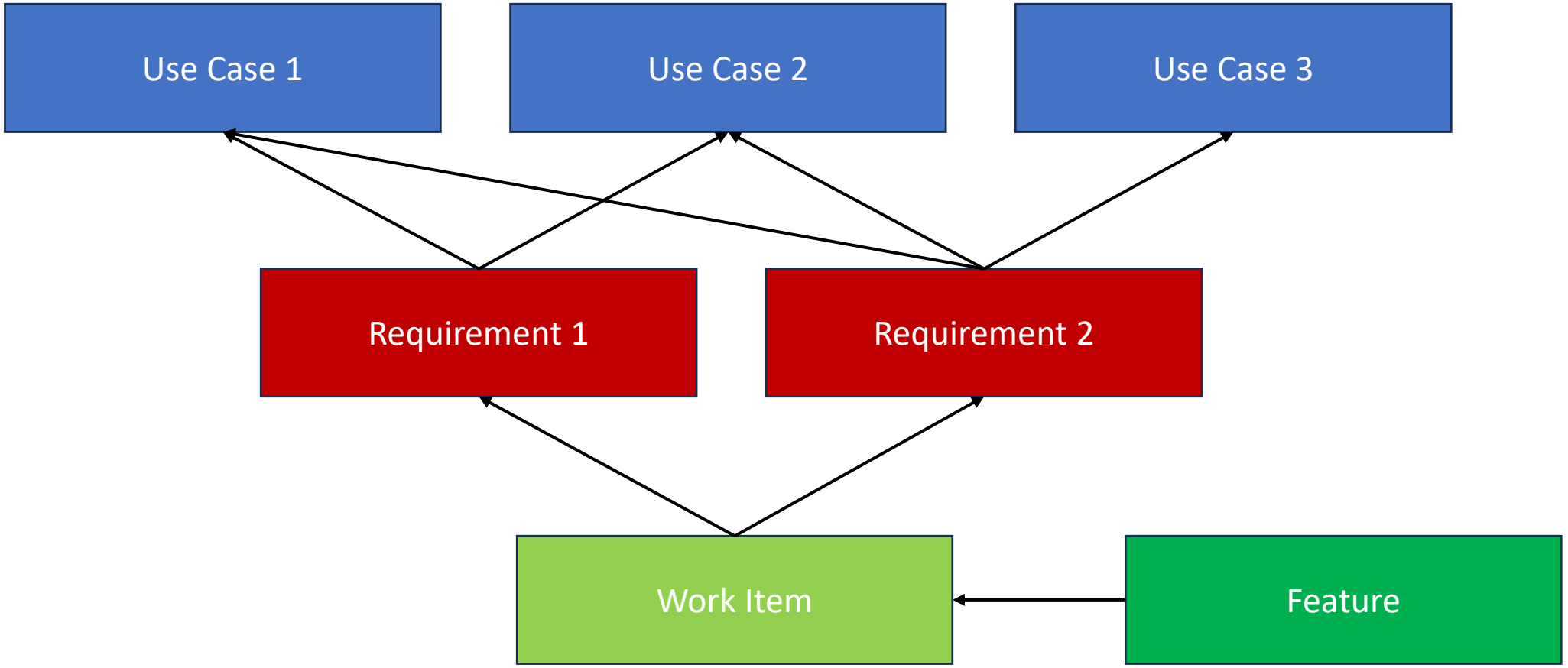

# Use Case/Requirements Examples

Michael McCool

27 September 2024

TPAC 2024

# UC → Req → WI → Feature



# Requirements: Format

- User Story Template:

  As a **PERSONA**, I want **CAPABILITY** so that **PURPOSE**.

**PERSONA**: Who

- Identifies primary user or beneficiary.
- Secondary stakeholders may also be identified in **PURPOSE**.

**CAPABILITY**: What

- May or may not map onto a single specific work item or feature (discuss).
- Should be specific enough that it is clear when it is satisfied.
- Should be satisfiable with finite effort.

**PURPOSE**: Why

- Larger context of goal, other stakeholders (e.g. support another SDO)
- May not be finitely satisfiable.

# Use Cases:

- Reconsider security & securityDefinitions being mandatory · Issue #757 · w3c/wot-thing-description (github.com)

**Use case 1:** TD Access and Things all have same security scheme and access control

- In Mozilla's WebThings Gateway implementation …
- **All the Thing Descriptions use the same security scheme** and
  in order to download a Thing Description you need to already be authenticated
  (this is intentional, the Thing Descriptions are not public for privacy reasons).
- … including security metadata in the Thing Description [is] entirely redundant because in order to read the metadata a client needs to already have that information.

**Use case 2:** No authentication on local network (security added via proxy)

- Web things implemented using Mozilla's WebThings Framework may
  choose to be unauthenticated and discoverable on a local network,
  with the WebThings Gateway used to safely proxy them to the Internet,
  with the gateway handling the authentication scheme separately from the device itself.

**Use case 3:** Public services intentionally not requiring authentication

- Web things in public places (e.g. vending machines, parking meters, interactive exhibits in museums) may intentionally be unauthenticated (though may still use an encrypted connection) because they are designed for public access.

**Comments:**

- security/securityDefinitions are too verbose in common cases, making TDs hard to use and understand.
- It is, however, useful to have the flexibility to model more complicated cases when necessary.
- TDs may or may not be available with the same security as the Things they describe.

# Use Case Example: Original

**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.

# Use Case Example: Rewrite

## Introduction

- Providing security metadata in advance in a TD is meant to allow consumers to know what to expect when connecting to the Thing.
- However, if this metadata is too complicated or verbose to specify then it increases the potential for error, and increases overhead for communicating TDs.
- Common cases should be simple, while more complex cases should be possible.
- Here are some common cases that should be simple to specify:
  - No security.  There are situations where this is appropriate, for example a Thing inside a trusted and protected network domain.
    ***It does not add security to make this case intentionally "irritating".***
  - Security schemes using default parameter values.
  - Thing descriptions where all affordances use the same security scheme.
  - Cases where Things and Thing Descriptions are accessed using the same security.

# Requirements: Examples

Based on https://github.com/w3c/wot-thing-description/issues/2039

- As a ***producer of WoT TDs***,
  I want ***to be able to specify simple security schemes inline***
  so that ***TDs are less verbose and easier to write in simple cases.***
- As a ***producer of WoT TDs***,
  I want ***to be able to depend on security scheme parameter defaults***
  so that ***TDs are less verbose and easier to write in simple cases.***
- As a ***producer of WoT TDs***,
  I want ***the "no security" scheme to be the default***
  so that ***TDs are less verbose and easier to write in simple cases.***