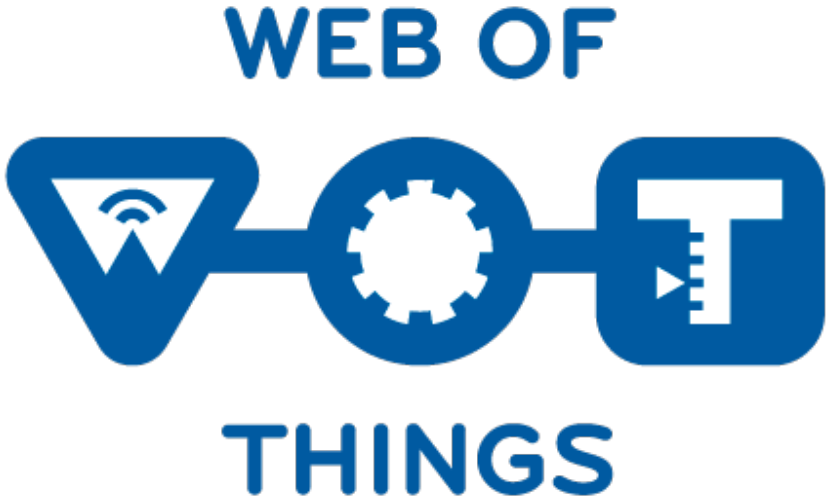

Discovery Issues

Michael McCool, Intel

17 March 2021

Summary

- Framing
  - Reserialization of TD and TD elements/fragments
  - Needed for SPARQL queries and RDF round-tripping
- Pagination
- Signing and canonicalization
  - JSON-LD stability, RDF round-tripping, metadata in "enhanced TDs"
- Validation
- Security and Privacy Considerations
- Geolocation
- JSON Path
- Security Bootstrapping

Framing

- Need JSON-LD 1.1 Framing Document
  - Necessary for any kind of RDF processing that produces TDs compliant with the specification
  - **In particular:** necessary for TD Directories to support SPARQL queries
- Discussion:
  - Is this an implementation issue, or a spec issue?
  - Should we be publishing an "official" framing document?

Pagination

- Much discussion recently about how to break up long responses with multiple TDs, or long TDs
- Various options
- Mostly we have focused on paging through TDs as opposed to breaking up long TDs
  - But then long TDs might break buffer limits, etc.
- Problem is easier if we adopt standard but HTTP-specific pagination
  - Then pagination controls are handled in HTTP headers and do not complicate body

Signing and Canonicalization

- Security has been discussing adding signing to preserve TD integrity
- This requires a canonical form of TDs
  - Needs to be specified in TD spec
  - Foundation is JSON canonicalization, but TD-specific elaborations needed, for example, the handling of default values, ordering to simplify processing, etc.
- Various other operations might break signing:
  - Insertion of metadata by directories in TDs ("enhanced TDs")
  - Protocol translation
- Modification of TDs can be handled by chaining
- Also need to consider whether outputs of SPARQL queries need to be canonicalized, signed

Validation

- A formal definition of "TD Validation" is needed
- This is because directories should only store "valid" TDs
- Of course a valid TD is one that "satisfies the TD specification" but not everything in the spec can be validated just by looking at the TD
- Some things such as validating semantic extensions are too expensive to justify
- **Proposal:** Define a subset of assertions that can be validated just by using JSON Schema (we already know this subset).

Security and Privacy Considerations

- Contexts: Institutional, Personal, various combinations
- Security and Privacy Considerations
  - Mitigating denial of service attacks (security)
  - Protection against location tracking (privacy)
- Other issues
  - What authentication and authorization are suitable for directories, in what circumstances?
  - Protection against code-injection attacks (e.g. JSON Path)

Geolocation

1. Information Model
   - How geolocation data is to be encoded in TDs
   - Needs to be flexible enough to handle both static and dynamic situations
2. Query Model
   - How geolocation data can be used during discovery to filter results
   - In dynamic situations, don't necessarily want to have to update directories constantly
   - Don't want directories to have to contact Things themselves during queries

Proposal:

- https://github.com/w3c/wot-discovery/blob/main/proposals/geolocation.md
- So far, information model only; query model WIP
- Needs to be aligned with existing geolocation standards

JSON Path

- Currently support is *required*
- Popular, nicer syntax for JSON content like TDs
- To use in Directory API, need formal specification
  - Ideally an external specification we can simply cite
- Need certain issues like JavaScript code injection addressed
- IETF proposal is a good start:
  - https://ietf-wg-jsonpath.github.io/draft-ietf-jsonpath-jsonpath/
  - However we will have to discuss timing of when and if this will become an actual standard
- Fallback would be to use XPath, which is (mostly) equivalent

Security Bootstrapping

- Exploration requires secure authentication and authorization before a TD is provided
- How does a client know what security scheme is needed to fetch the TD without access to the TD??? (see wot-discovery issue 135)
- Mostly a problem for self-description (directories can use nosec for the TD as it has no private information)
- Options:
  - Specified default
  - Protocol-specific negotiation (e.g. HTTP headers)
  - Two-phase (proposed in issue above)
    - Error response or other mechanism provides security scheme to fetch TD
    - But where does this scheme come from? It is not given in the TD itself (meta-metadata)

Other

- Addressing WebThings feedback
  - Use of mDNS for local discovery
- Addressing CoreRD feedback
  - We want to limit how much metadata is distributed/leaked in "introductions"
- Directory Federation
  - Need link relation type to allow directories to point at other directories
  - Not clear how to do semantic summaries of the contents of directories
  - Consumer needs to follow links and send queries (if directory did it, it would be nicer for the consumer, but unfortunately this leads to amplification and can be exploited in a DoS attack)
- Directory Semantic Extensions
  - Directory has a context, should be given an official URL