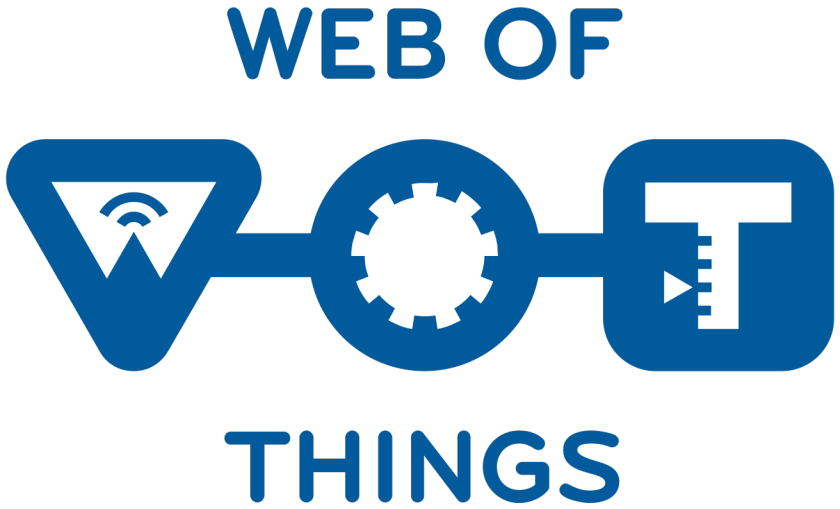


Discovery Architecture Overview

October 20, 2020

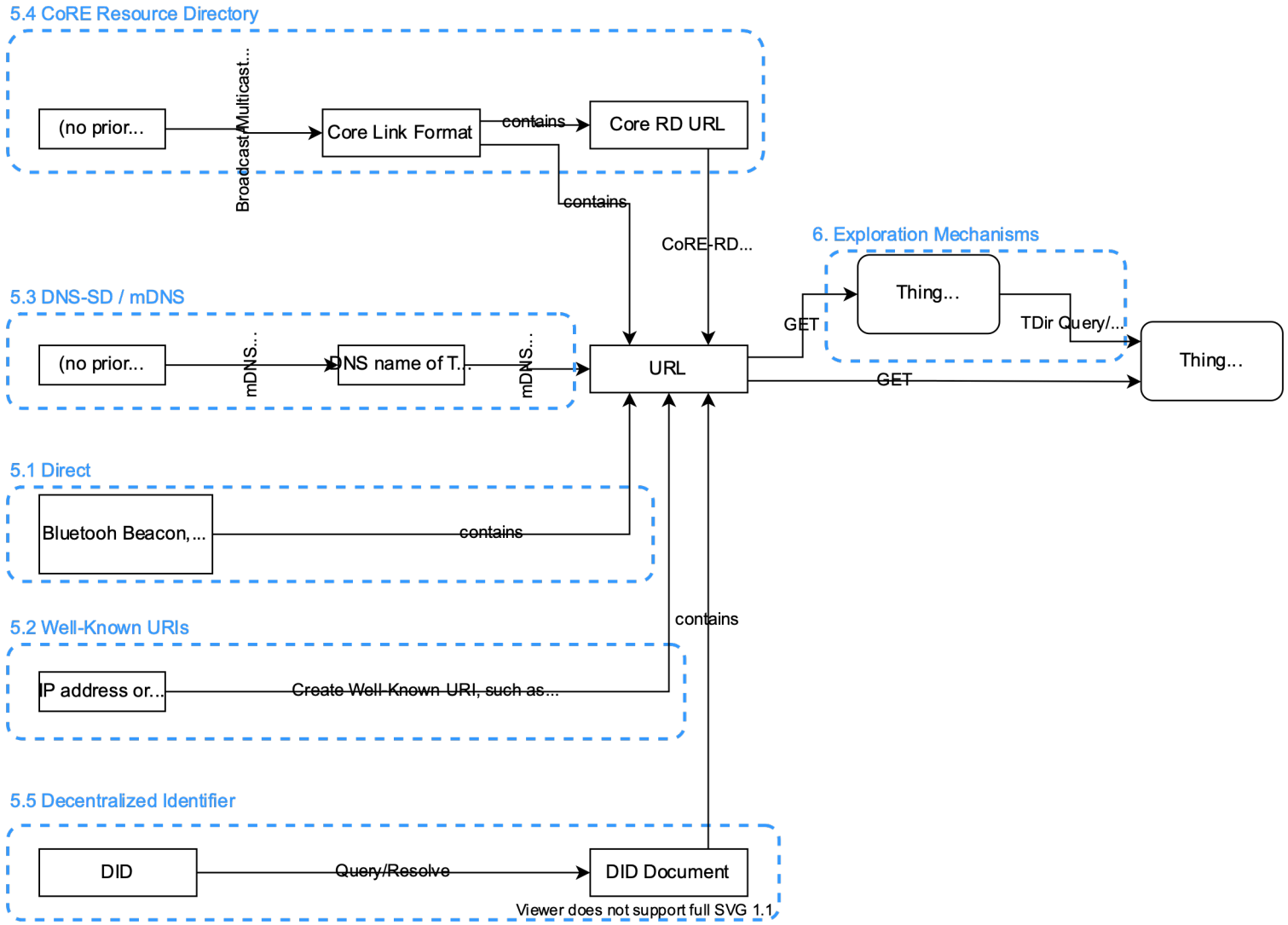
Michael McCool

Discovery - Summary

- Defines (***prescriptively***) a mechanism to distribute TDs
  - Support ad-hoc retrieval
  - Support "late binding"
  - Support spatial and content-based search
- Both local and global contexts
  - Spatial search not limited to local network
  - Both peer-to-peer and directory-based retrieval supported
- Two-phase introduction/exploration
  - Support for a variety of introduction ("First Contact") protocols
  - Actual metadata only released in second, "exploration" phase *after* authentication and authorization
- Emphasis on privacy protection
  - Need to protect queries and exploration services
  - Need to avoid privacy risks both from direct information and inferred information leaks

Overview

- Phase 1: Introduction
  - Multiple mechanisms supported
  - ***No metadata released at this point***
  - Just identifies an exploration mechanism → URL
- Phase 2: Exploration
  - ***Authentication required***
  - Can be peer-to-peer or directory based



Introduction Mechanisms

- Direct
  - Just a URL, by any means (including typed in manually); NO TYPE
  - Also covers a variety of other mechanisms: QR codes, Bluetooth beacons, etc.
- Well-Known URIs
  - Use RFC8615 conventions, so if you know e.g. the IP address of device, you find the TD:
  - `{{base}}/.well-known/wot-thing-description`
- DNS-SD
  - Service Name: `_wot` with subtype `_directory._sub._wot`
  - Multicast DNS also supported (in local networks)
  - Additional information in TXT records, including alt path and type
- CoRE Link Format and CoRE RD
  - Set of typed links → define types to identify and differentiate Things and Directories
- DID Documents
  - Set of typed links → define types to identify and differentiate Things and Directories

Differentiation of Exploration Services

- Two types of exploration
  - Direct link to TD, e.g. hosted by device
  - Link to directory service where TDs can be queried
- Introduction gives just a URL
  - Some Introduction services can also provide a type, some can't
  - ***Have to assume there is no type available when the link is provided...***
- Exploration URL Resolution:
  - ALL URLs provided by introduction services resolve to TDs
  - For a device, that is the device's own TD
  - For a directory, that is the directory's TD, marked with "Directory" @type
  - For introduction services that can do so, there is also a sub-type defined for directories

Directory Query Mechanisms

Syntactic

JSONPath

- Mandatory (MUST)
- Popular
- Not a standard (or even that well documented...)

XPath

- Recommended (SHOULD)
- Same functionality as JSONPath
- An actual W3C standard

Semantic

SPARQL Endpoint

- Optional (MAY)
  - Not all directories need to support it
  - Relatively expensive to implement

Discussion: Is SPARQL *too* powerful?

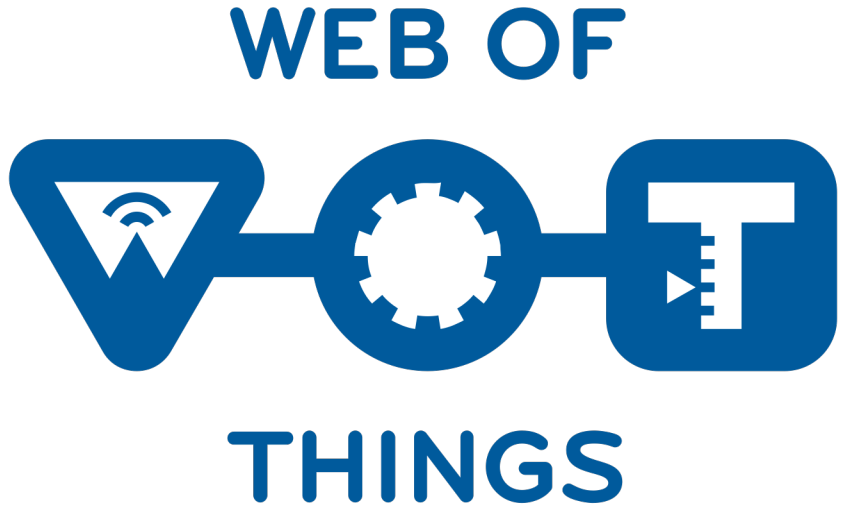
- Easy to form pathological queries that take excessive compute time to complete

Major Open Issues

- Geospatial Query: https://github.com/w3c/wot-discovery/issues/79
  - Need to define a geospatial query filter for directory searches, e.g. point and radius/elevation interval
  - Open 1: Definition of an Introduction mechanism with geospatial filters (a DNS extension is being considered by IETF, however)
  - Open 2: Where does the geospatial information for TDs come from?
  - Open 3: Protecting privacy
- JSONPath
  - Not an actual standard... being discussed by IETF, however
- Federated/Chained/Merged Queries
  - Can directories query other directories? (Right now: No...)
- Registration of TDs (and System Lifecycle in WoT Architecture)
  - How do directories get TDs in the first place?

Resources

- Repo
  - https://github.com/w3c/wot-discovery
- Editor's Draft (and FPWD Candidate):
  - https://w3c.github.io/wot-discovery/
- Requirements:
  - https://github.com/w3c/wot-discovery/blob/master/requirements.md
  - Note: being migrated to WoT Architecture document


WEB OF
THINGS