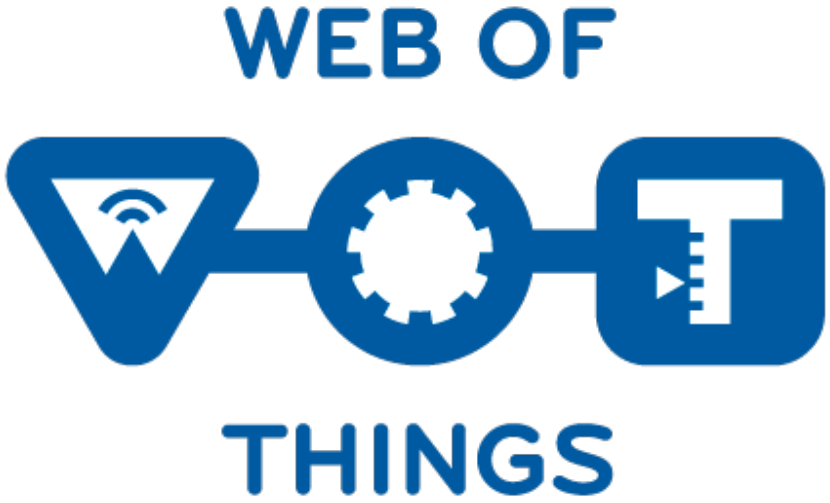

Discovery Intro

Michael McCool, Intel

17 March 2021

Discovery Goals

Capabilities

- Support both local and global/remote discovery (unconstrained by network domain)
- Support "localizable" discovery (constrainable by location)
- Support both "syntactic query" (keywords) and "semantic query" (linked data)
- Support a directory service for searching large repositories of Things
- Support peer-to-peer (self-identifying "smart object") discovery
- **Stretch goal**: Support filtering of result by geolocation

Privacy-Preserving Architecture

- Respect device and information Lifecycle
- Distribute TDs only to authenticated and authorized users
- Don’t leak private data to unauthorized users
- Don’t leak data that can be used to INFER private information to unauthorized users

Alignment with Existing and Evolving Standards

- IETF CoRE Resource Directories, CoRE Link Format, DID, OGC, WGS84, XPath, …
- Compatible with WoT Scripting API

Two-Phase Architecture

Phase 1: Introduction

- “First Contact” Protocol
  - Answers the question: How to initiate discovery from zero knowledge?
- Open
  - Can be accessed with no or limited access controls
  - Based on existing standards, and can be extended to new standards
- Lightweight
  - Does not use significant resources on responder
  - Resistant to Denial of Service attacks
- Provides intentionally limited information
  - Avoid leaking any metadata that can be used to infer private data
  - This includes types of devices, device ids, owners, timestamps, etc.

Phase 2: Exploration

- Authentication and authorization required
- Supports more complex query and filtering capabilities
- Provides access to rich metadata (TDs)
- Access controls can limit data returned

Introduction

- “First Contact” protocol
  - Output: Address of exploration service, for example, a directory service
  - Need not use broadcast mechanism or even a network protocol
  - MAY use well-known network discovery mechanism (eg DNS-SD, DHCP, etc)
- Address should not leak any other metadata, e.g. type of devices
- Can have multiple mechanisms for introduction
  - Local: QR code, mDNS, DNS-SD, DHCP, Bluetooth beacons (Eddystone), etc.
  - Global: Search engine, global repositories, company repositories, city portal, etc.
  - Self: Well-known addresses, e.g. “.well-known/td”
- Existing mechanisms that have lists of typed links can also be used here:
  - CoRE RD, DID Documents, DNS-SD, etc.
  - Use these to find entry point rather than to distribute metadata directly
- Introduction MAY in some cases point directly at a Thing Description
  - Self-describing objects are similar to a “directory” has only one TD
  - Still requires authentication in principle to access content

Exploration

- Authentication required, and **ONLY THEN** can metadata be accessed

Two general forms:

1. **Directory service: Queryable database**
   - Syntactic query: for keywords, by title, by id; JSON Path or XPath
   - Semantic query (optional): by semantic LD terms; SPARQL
   - Typically running on a "hub" (edge computer, gateway, cloud server, etc.)
   - Repository for a potentially large number of TDs
   - Registration: devices can self-register or be registered by another agent)
2. **Self: Direct retrieval of TD from Thing**
   - To be discussed: limited query support to enable filtering on client side?

Privacy Considerations

Avoid Distribution of Direct and Inferencable Private Data

- Two-phase approach ***not sufficient*** to preserve privacy in all contexts
- Privacy preservation also depends on the design of the API
- API needs to hide data that can be used to infer private information, such as the location of device doing the discovery
- The query itself also needs to be hidden, not just the results

Third-party Code Risks (e.g. browser):

- If discovery API follows two-phase structure, where Introduction returns list of directories, then even without authenticating the list of directories visible can possibly be used to infer location
- This is especially true if the introduction can be constrained to specific mechanisms and repeated, as different results with different mechanisms might be used to create a location fingerprint

Security Considerations

Man-in-the-Middle Attacks

- Modification of URLs in TDs might be used to redirect consumer through an unauthorized intermediary
- However, this might be the desired behaviour for proxies or protocol translators
- Mitigation: chainable TD signing

Denial of Service Attacks

- Queries can be expensive and can return a lot of data
- If abused individual queries can consume an unreasonable fraction of the hub's capability or bandwidth
- Mitigation: Limiting query execution times and result size

Code Injection

- Query languages (e.g. JSONPath, XPath, SPARQL) need to be constrained to disallow execution of arbitrary code on the server, access to filesystem, etc.

Resources

- Repository: https://github.com/w3c/wot-discovery
- Published FPWD: https://www.w3.org/TR/wot-discovery/
- Current Editor's Draft: https://w3c.github.io/wot-discovery/
- Proposals: https://github.com/w3c/wot-discovery/tree/master/proposals
  - Geolocation: https://github.com/w3c/wot-discovery/blob/main/proposals/geolocation.md