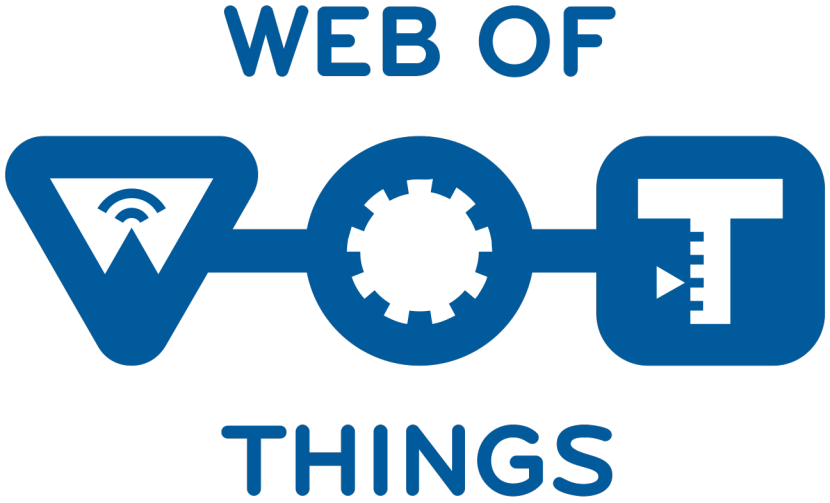


Discovery: Introduction Mechanisms

Kunihiko Toumura

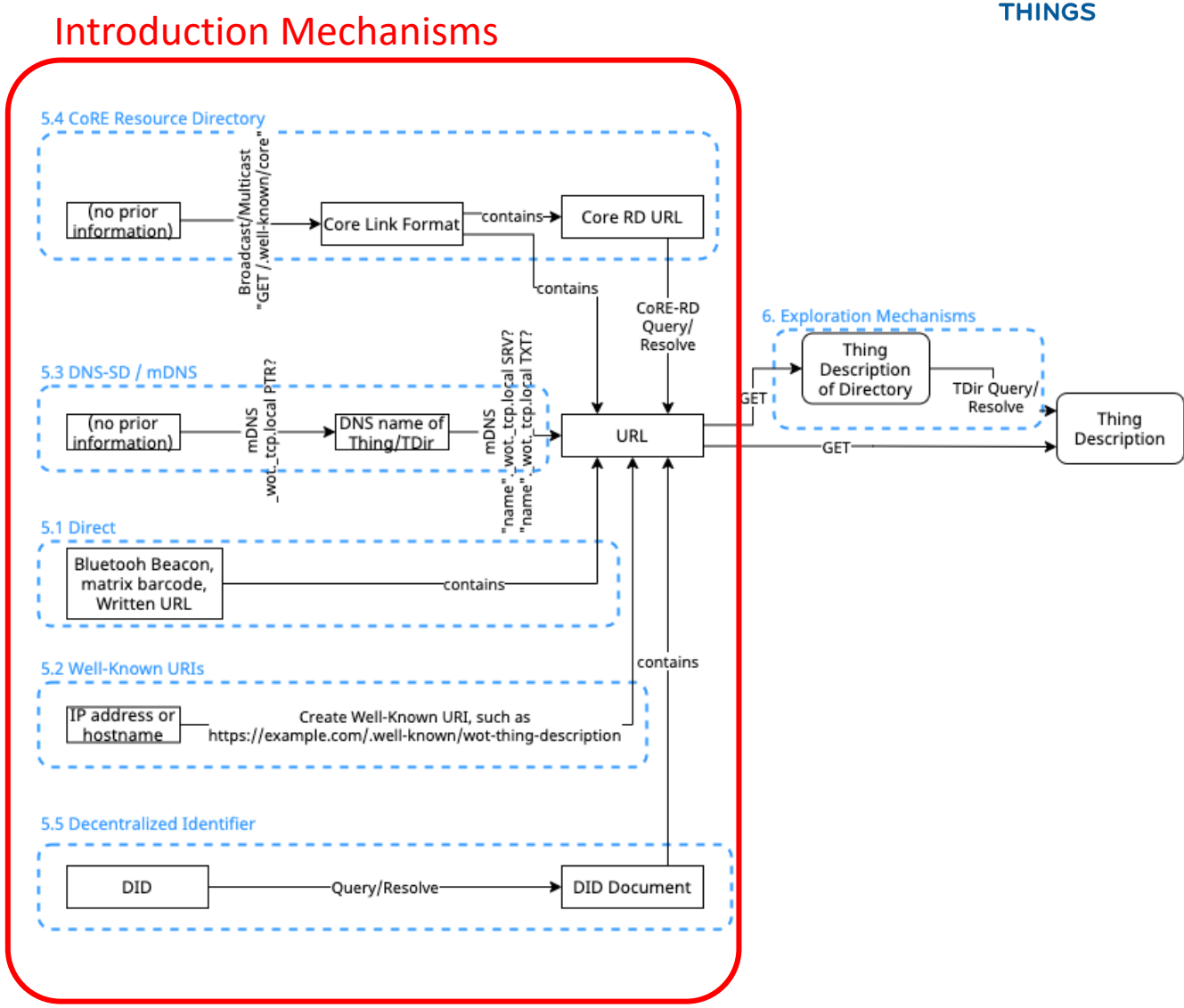
October 20, 2020

Outline

- Overview
- Current described mechanisms:
  - Direct
  - Well-known URI
  - DNS-based service discovery
  - CoRE Link Format and Core Resource Directory
  - Decentralized Identifier (DID) documents
- Discussions
  - Security/Privacy
  - Any other introduction mechanisms?

Overview

- Introduction Mechanism:
  - Find an URL of Thing Description of Thing or Thing Directory.
  - Utilize existing discovery mechanisms. Avoid inventing mechanisms.
  - Consumer may issue HTTP GET request to the URL to retrieve a TD.
  - TD's Content-type MUST be:
    `application/td+json`
  - "@type" of TD MUST be:
    - {"@type": "Thing"} => Thing
    - {"@type": "Directory"} => Directory



Direct

- Any mechanism that result in a single URL.
  - Bluetooth beacons, Matrix barcodes, and written URL.
- A GET on all such URLs MUST result in a TD.

Example 1: a QR code that contains an URL
‘http://ktorpi.local:1880/.well-known/wot-thing description’

Example 2: a QR code that contains entire TD
(for comparison; gzip+base64 encoded)

Well-known URI

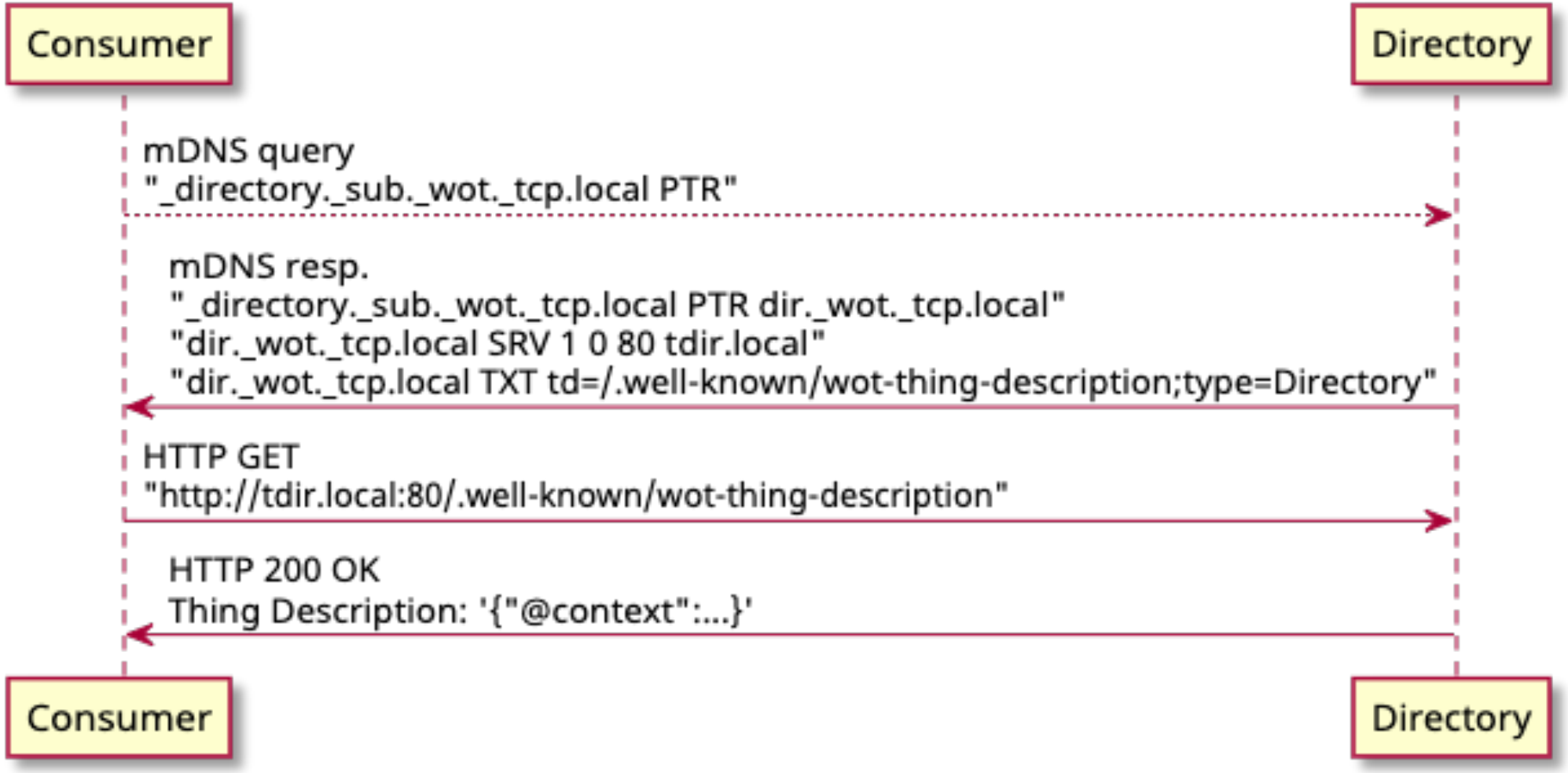
- RFC8615: Well-Known Uniform Resource Identifiers (URIs)
- Thing or Directory can host their Thing Description as a site-wide metadata
- “`/.well-known/wot-thing-description`” (tentative)
  - Example 1: a Consumer heuristically get a FQDN of some site: tdir.example.com, then issue HTTP GET https://tdir.example.com/.well-known/wot-thing-description to retrieve a Thing Description
  - Example 2: Broadcast/multicasting CoAP GET request to /.well-known/wot-thing-description to find Things/Directory Services in a same subnet.

DNS-based service discovery (1/2)

- DNS-based Service Discovery (RFC6763)
- Multicast DNS (RFC6762)
- Use (multicast) DNS query to discover Things or Directory Services
- DNS-SD Service Instance Name:
  - *<Instance>.<Service>.<Domain>*
- <Service> MUST be:
  - Thing: _wot._tcp (HTTP or HTTPS) or _wot._udp (CoAP)
  - Directory Service: _directory._sub._wot._tcp or _directory._sub._wot._udp
- When Consumer resolves above domain name, it receives following TXT records:
  - `td`: Absolute pathname of the Thing Description of the Things or Directory Service
  - `type`: Type of the Things Description, i.e. `Thing` or `Directory`.

DNS-based service discovery (2/2)

- Example sequence of Directory Discovery by mDNS



CoRE Resource Directory (CoRE-RD)

- draft-ietf-core-resource-directory-25
- We can use CoRE-RD as an introduction mechanism of Thing or Directory Service.
- Link for a Thing Description is stored as a CoRE Link (RFC6690).
- Endpoint type(`ep`):
  - TD for Thing: `wot.thing`
  - TD for Directory Service: `wot.directory`

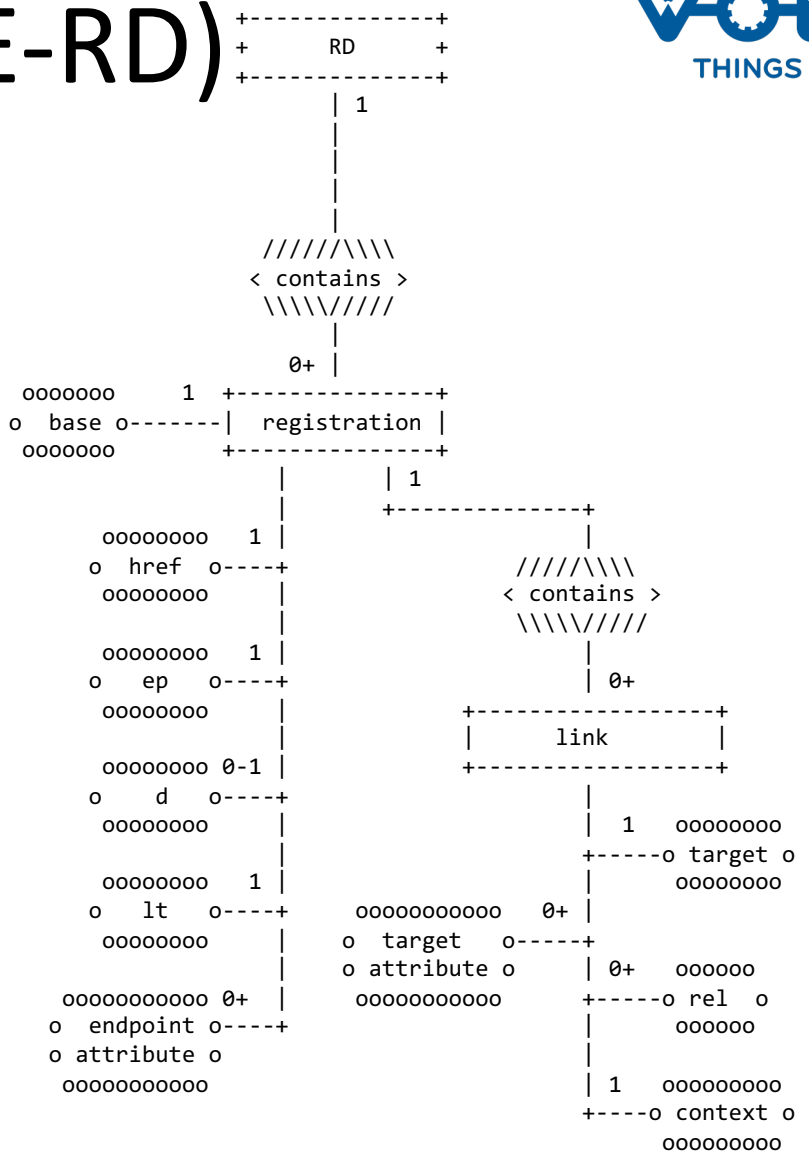
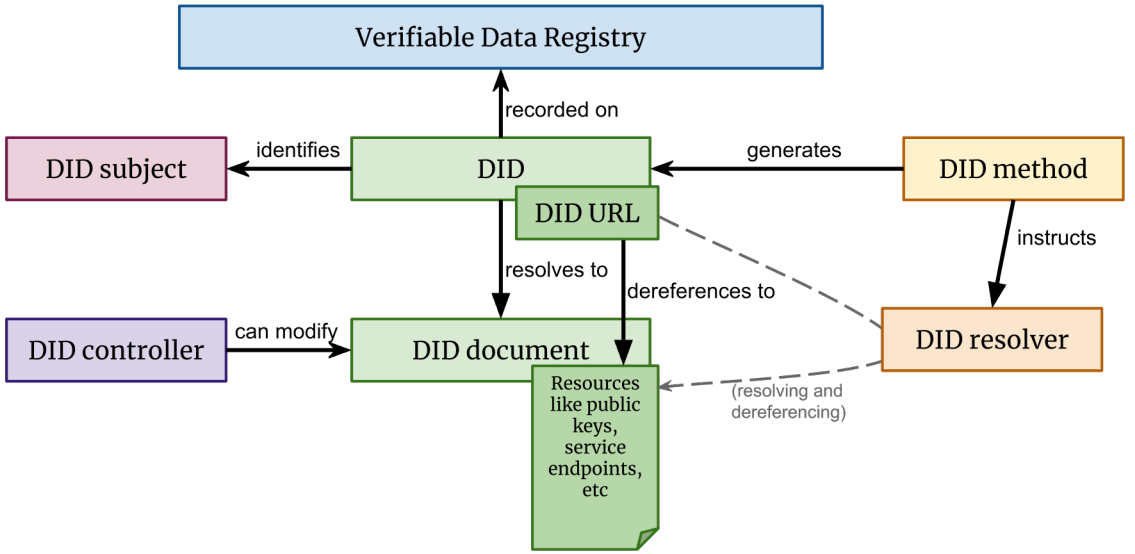


Figure 3: ER Model of the content of the RD

Decentralized Identifier (DID)

- DID can be used for pointing a Thing or Thing Directory.
- DID is resolved to DID documents, by DID resolver.
- DID document can contain a Service Endpoint which point to Thing or Thing Directory




```
{
  …
  "service": [{
      "id": "did:example:wotdiscoveryexample#td",
      "type": "WotThingDescription",
      "serviceEndpoint": "https://wot.example.com/td"
  }]
}
```

Discussions

- Security/Privacy Considerations
  - Some introduction mechanisms should be used in trusted environment…
    - Direct, well-known URL and DNS-SD are not protected from unautholized access.
    - To protect, we should use them in a private network which is protected by authentication (WPA, 802.1x, VPN, etc.)
  - … and/or TD should be protected by authentication
    - HTTP basic/digest auth, OAuth, etc.

- Are there any other mechanisms that should be included in the specification?