WEB OF THINGS

Discovery: Introduction Mechanisms

Kunihiko Toumura

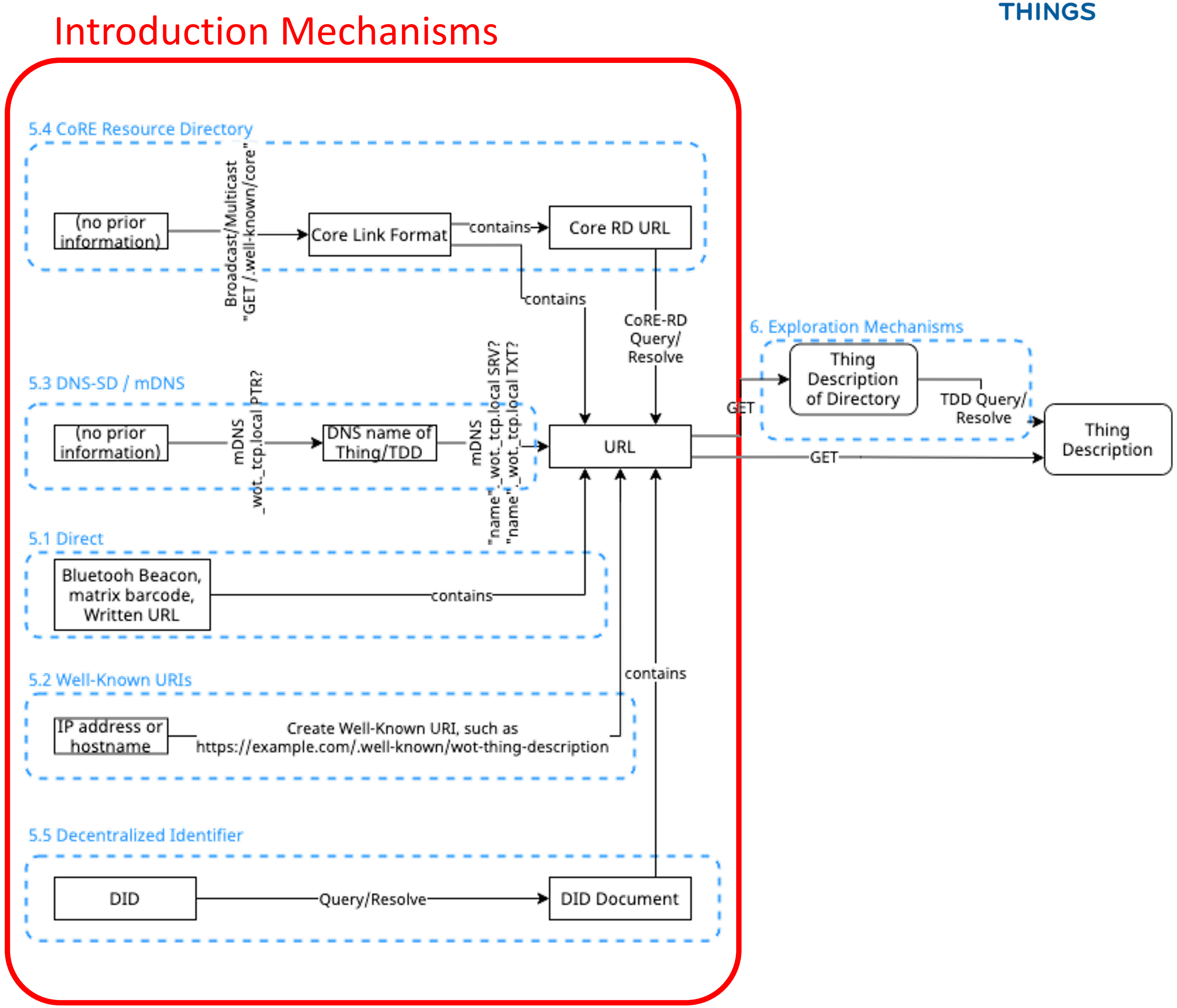
October 20, 2020

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 which points to Thing Description of Thing or Thing Directory.
  - Consumer may issue HTTP GET request to the URL to retrieve a TD.
    - Accessing to TD SHOULD be authenticated
  - Utilize existing discovery mechanisms. Avoid inventing mechanisms.
  - TD’s Content-type MUST be: `application/td+json`
  - ”@type” of Directory TD MUST be “`Directory`”
    - Type for Thing TD is ”`Thing`”, but not mandatory.



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.

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

Well-known URI

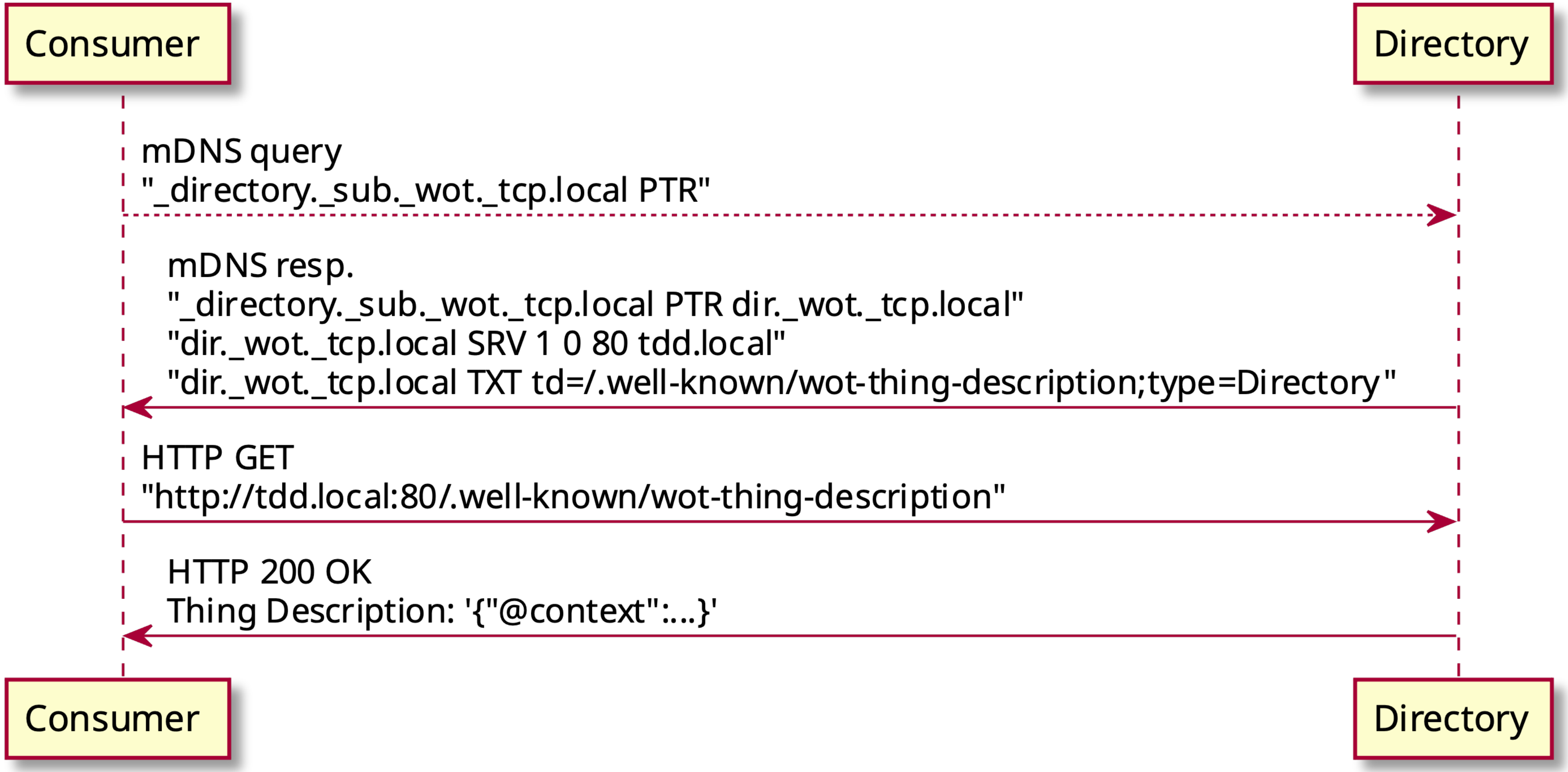
- RFC8615: Well-Known Uniform Resource Identifiers (URIs)
- Thing or Directory Service can host their Thing Description as a site-wide metadata
- “`wot-thing-description`” (tentative) for URL suffix
  - Example 1: a Consumer heuristically get a FQDN of some site: tdd.example.com, then issue HTTP request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

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



Consumer

Directory

mDNS query
"_directory._sub._wot._tcp.local PTR"

mDNS resp.
"_directory._sub._wot._tcp.local PTR dir._wot._tcp.local"
"dir._wot._tcp.local SRV 1 0 80 tdd.local"
"dir._wot._tcp.local TXT td=/.well-known/wot-thing-description;type=Directory"

HTTP GET
"http://tdd.local:80/.well-known/wot-thing-description"

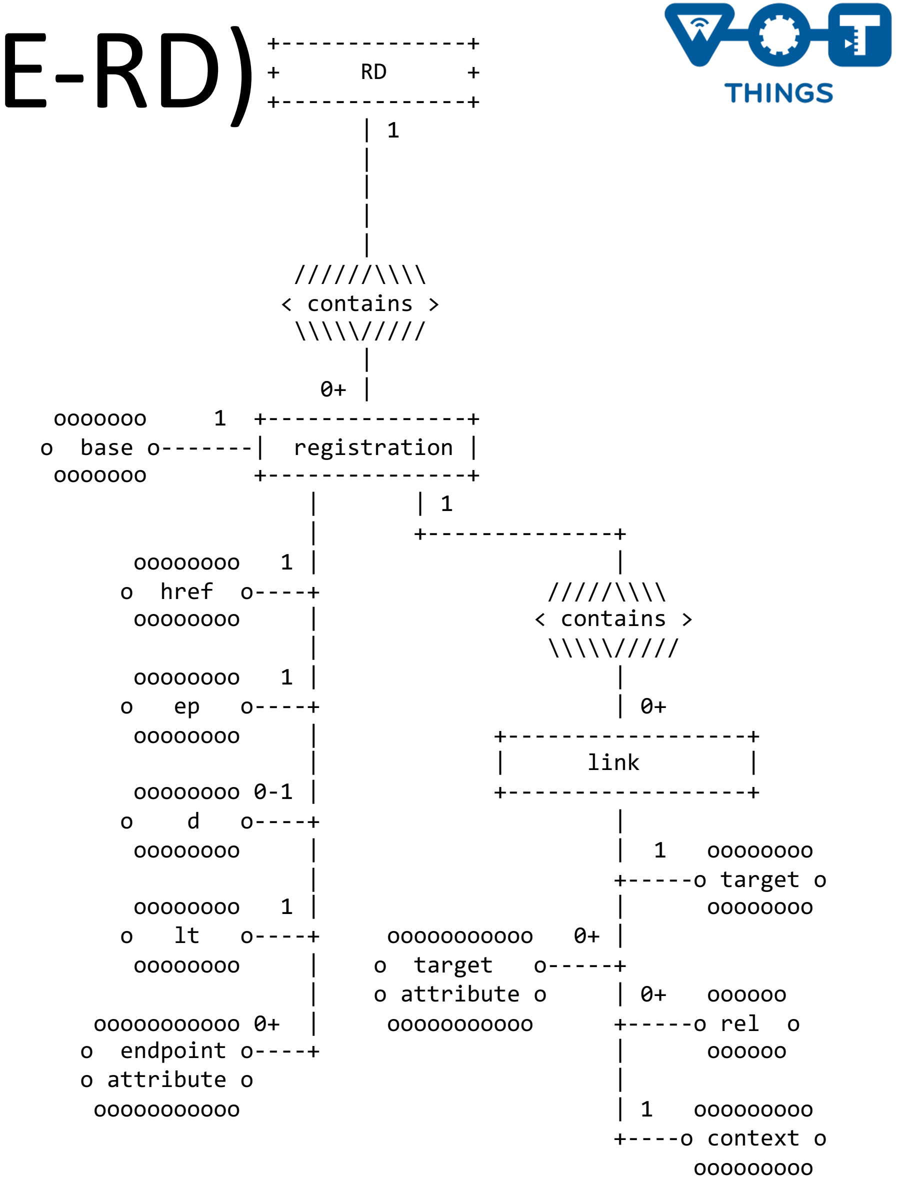
HTTP 200 OK
Thing Description: '{"@context":...}'

Consumer

Directory

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(`et`):
  - TD for Thing: `wot.thing`
  - TD for Directory Service: `wot.directory`

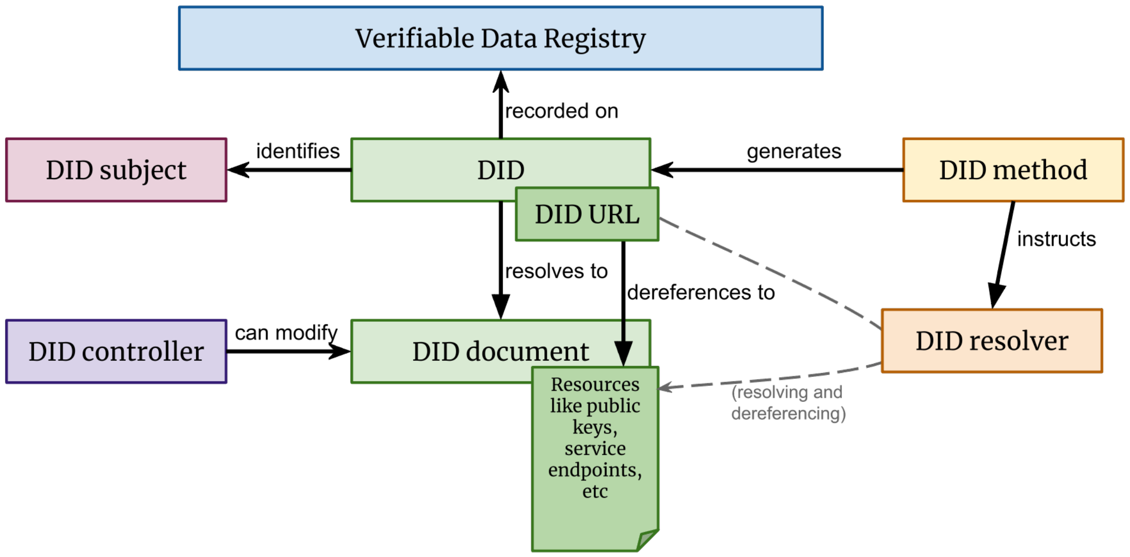


```
                 +---------------+
                 +      RD       +
                 +---------------+
                        | 1
                        |
                        |
                        |
                        |
                   //////\\\\
                  < contains >
                   \\\\\/////
                        |
                     0+ |
 ooooooo     1  +---------------+
o  base o-------|  registration |
 ooooooo        +---------------+
                    |       | 1
                    |       +--------------+
       oooooooo   1 |                      |
      o  href  o----+                 //////\\\\
       oooooooo     |                < contains >
                    |                 \\\\\/////
       oooooooo   1 |                      |
      o   ep   o----+                      | 0+
       oooooooo     |              +------------------+
                    |              |      link        |
       oooooooo 0-1 |              +------------------+
      o    d   o----+                      |
       oooooooo     |                      |  1   oooooooo
                    |                      +-----o target o
       oooooooo   1 |                      |      oooooooo
      o   lt   o----+     ooooooooooo   0+ |
       oooooooo     |    o  target   o-----+
                    |    o attribute o     | 0+   oooooo
    ooooooooooo 0+  |     ooooooooooo      +-----o rel  o
   o  endpoint o----+                      |      oooooo
   o attribute o                           |
    ooooooooooo                            | 1   ooooooooo
                                           +----o context o
                                                 ooooooooo
```

Figure 3: ER Model of the content of the RD

Decentralized Identifier (DID)

- DID can be used for pointing a Thing Description of 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"
  }]
}
```

Example Service Endpoint description in DID document

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.
    - We should use them in a private network which is protected by authentication (WPA, 802.1x, VPN, etc.), or in a space that is protected by physical security.
  - … 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?