WEB OF

THINGS

W3C Web of Things and Geolocation Discovery

Michael McCool

21 July 2022

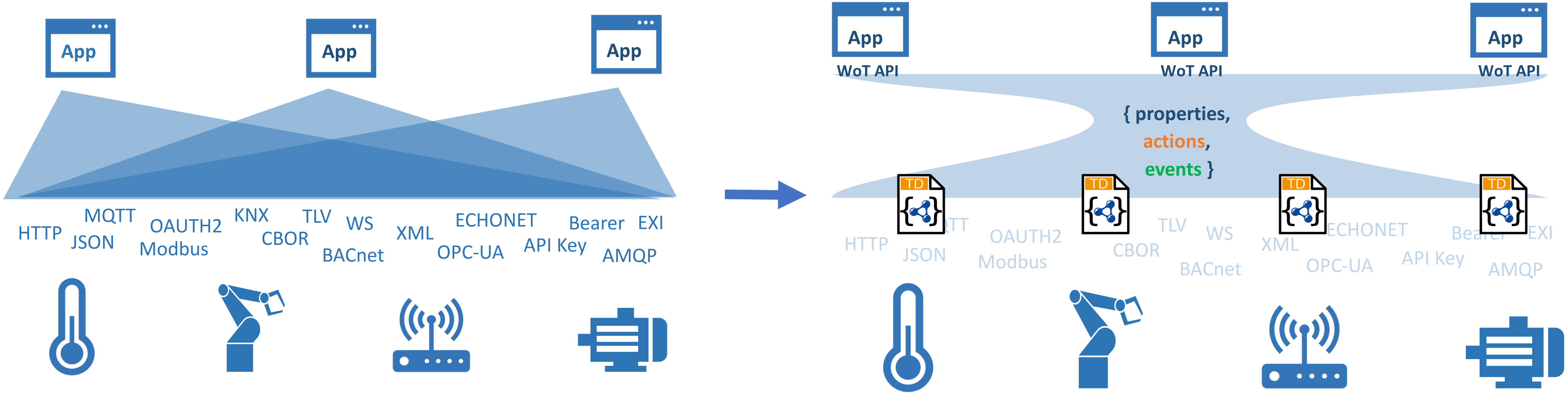
Presented to W3C SDW IG/WG

Outline

- What is WoT?
  - Applying and extending web standards for IoT
  - Descriptive interoperability: Thing Descriptions
  - Recent Activity
  - Finding Thing Descriptions: Discovery
- Discussion
  - Gaps and Future Work
  - Geolocation Discovery
    - Data: Dynamic and Static
    - Geolocation queries for Discovery
    - Interval queries on actual data?

W3C Web of Things (WoT)

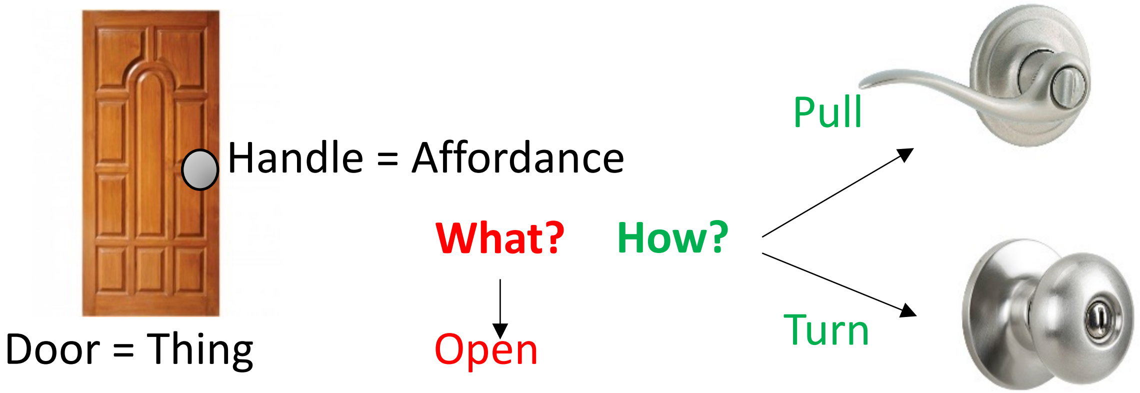
- W3C Working Group goal: Adapting web technologies to IoT
- Already published: Thing Description (TD) metadata format
  - TD describes the available interactions (network API) of a Thing
- New deliverables in progress, including Discovery
  - How does a potential user obtain the TD for a Thing?



Descriptive Interoperability: TDs

WoT Architecture

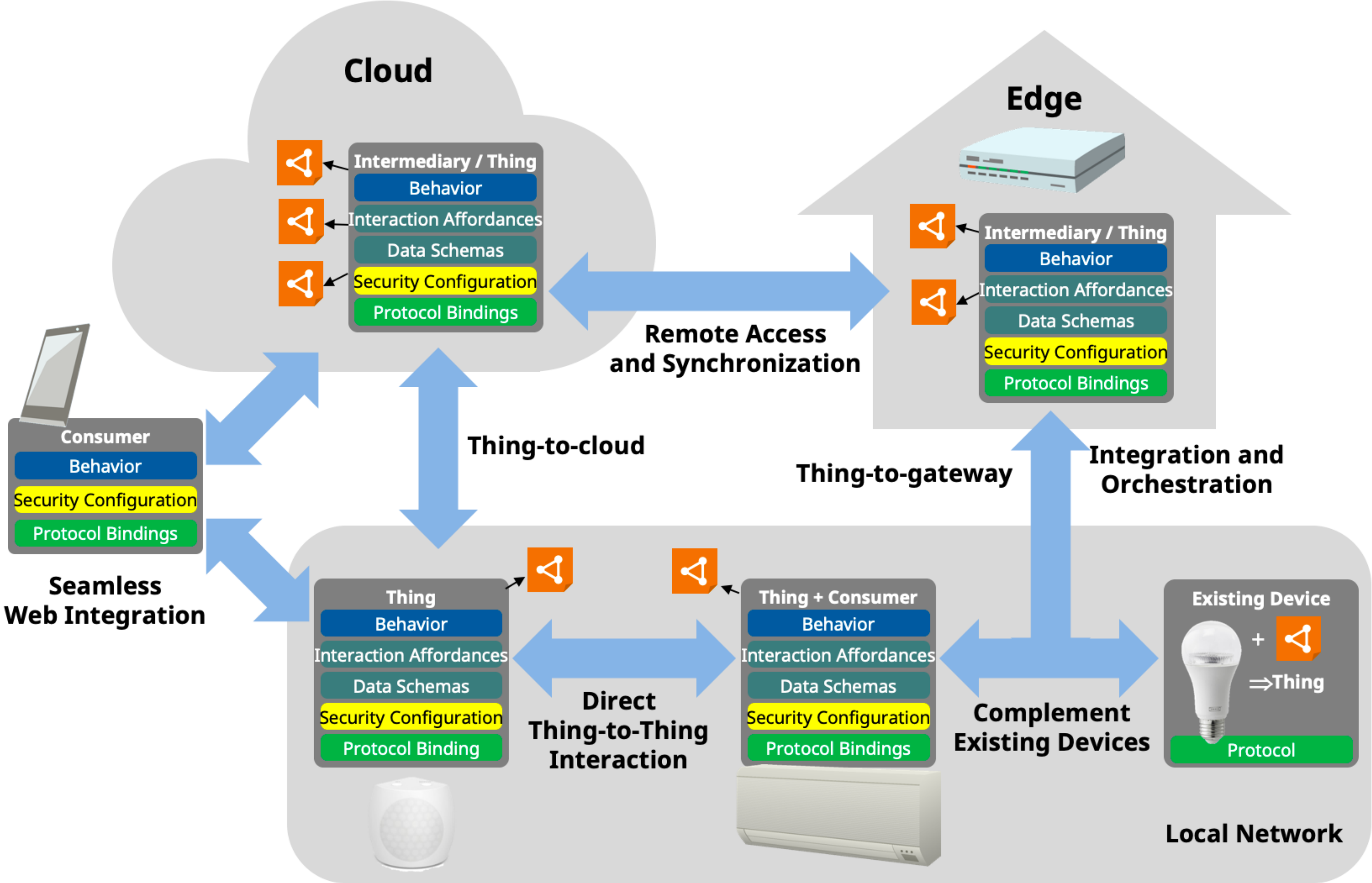
- Constraints
  - "Things" must have a TD
  - Must use URIs, IANA media types, etc.
- Thing Description Affordances
  - Describes WHAT the possible choices are
  - Describes HOW to interact with the Thing



WoT Thing Description (TD)

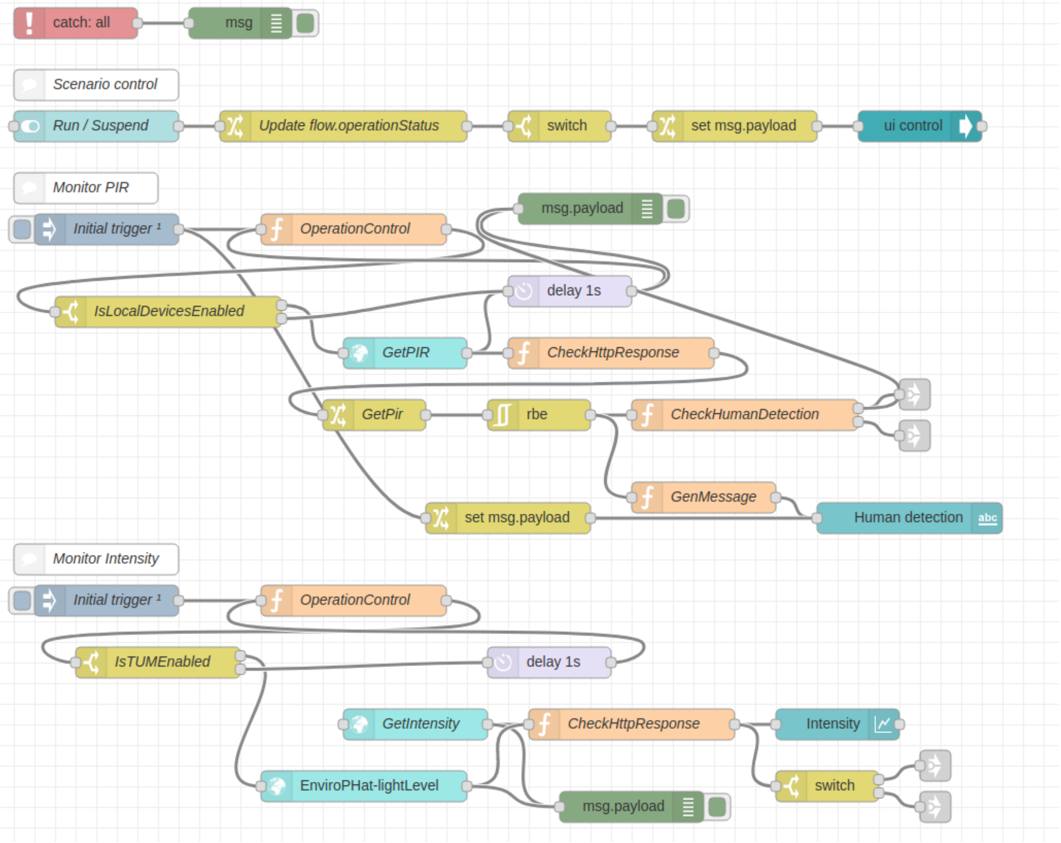
```
{
  "@context": [
    "https://www.w3.org/2022/wot/td/v1.1",
    { "iot": "http://iotschema.org/" }
  ],
  "id": "urn:dev:org:32473:1234567890",
  "title": "MyLEDThing",
  "description": "RGB LED torchiere",
  "@type": ["Thing", "iot:Light"],
  "securityDefinitions": {
    "default": {"scheme": "bearer"}
  },
  "security": ["default"],
  "properties": {
    "brightness": {
      "@type": ["iot:Brightness"],
      "type": "integer",
      "minimum": 0,
      "maximum": 100,
      "forms": [ ... ]
    }
  },
  "actions": {
    "fadeIn": {
      ...
```

Usage Patterns Overview



Orchestration

Node-RED/node-gen



node-wot/Scripting API

```
WoTHelpers.fetch( "coap://localhost:5683/counter" ).then( async (td) => {
 // using await for serial execution (note 'async' in then() of fetch())
 try {
  const thing = await WoT.consume(td);
  console.info( "=== TD ===" );
  console.info(td);
  console.info( "==========" );

  // read property #1
  const read1 = await thing.readProperty( "count" );
  console.info( "count value is" , await read1.value());

  // increment property #1 (without step)
  await thing.invokeAction( "increment" );
  const inc1 = await thing.readProperty( "count" );
  console.info( "count value after increment #1 is" , await inc1.value());

  // increment property #2 (with step)
  await thing.invokeAction( "increment" , {'step' : 3});
  const inc2 = await thing.readProperty( "count" );
  console.info( "count value after increment #2 (w/ step 3) is" , await inc2.value());

  // decrement property
  await thing.invokeAction( "decrement", undefined, {
    formIndex: getFormIndexForDecrementWithCoAP(thing);
  } );
  const dec1 = await thing.readProperty( "count" );
  console.info( "count value after decrement is" , await dec1.value());

 } catch(err) {
  console.error( "Script error:", err);
 }

}).catch( (err) => { console.error( "Fetch error:", err); });
```

Deliverables

New/Updated Normative Documents:

- Architecture 1.1: https://github.com/w3c/wot-architecture
- Thing Description 1.1: https://github.com/w3c/wot-thing-description
- Discovery: https://github.com/w3c/wot-discovery
- Profiles: https://github.com/w3c/wot-profile

New/Updated Informative Documents:

- Binding Templates: https://github.com/w3c/wot-binding-templates
- Scripting API: https://github.com/w3c/wot-scripting-api
- Use Cases and Requirements: https://github.com/w3c/wot-usecases

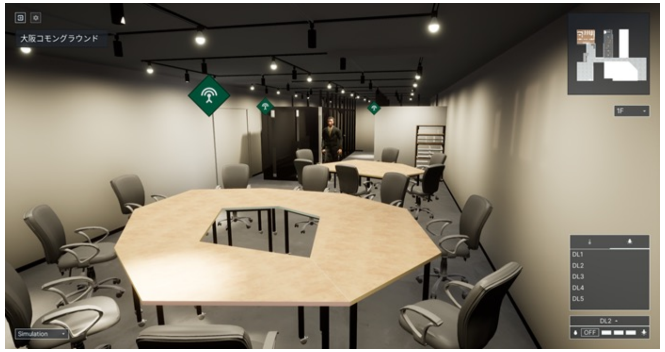
Community Resources:

- Web Site: https://www.w3.org/WoT/

Recent Activity

- Commercial Usages related to Geolocation
  - Takenaka Construction – Smart Building Information Management systems
  - Netzo – IoT dashboards and device management
- Directory Implementations
  - WoT Hive, LogiLab (SPARQL based), Fraunhofer LinkSmart
- IETF Relationships: JSON Path, CoreRD, COSE/JOSE, ASDF
- Under Discussion:
  - Geospatial data, Embedded JSON Signatures
  - New Charters/New Deliverables

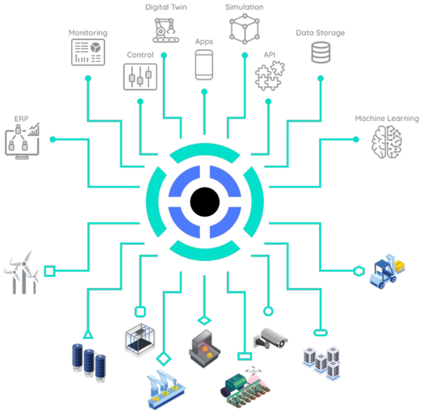
Applications



https://www.takenaka.co.jp/news/2021/05/02/

Takenaka Corporation

- CGLL Platform - BIM



https://netzo.io/

Netzo

- IoT Data Hub
- Dashboards

https://new.siemens.com/global/en/products/buildings/automation/desigo.html

https://www.evosoft.com/en/digitalization-offering/saywot/

Siemens

- Desigo CC – BIM
- Say WoT!

ditto

https://www.eclipse.org/ditto/2022-03-03-wot-integration.html

Bosch

- Eclipse Ditto - Digital twin

Gaps and Future Work

- GIS Integration
  - Geospatial data and discovery
- Data Management
  - Digital Twins and shadows
  - Event notifications
  - Historical data
- Security
  - Key provisioning and onboarding
  - Secure LAN access
  - Proxy services
  - Access control and ad-hoc sharing
  - MUDS
- Accessibility
  - Sensory modality mapping
  - Textual/descriptive interfaces
  - Service location
  - Mobility services
- Advanced Use Cases
  - Transportation
  - Logistics
  - Distributed energy management
  - AR visualization
  - Analytics integration e.g. for health and safety monitoring

Discovery

Goal:

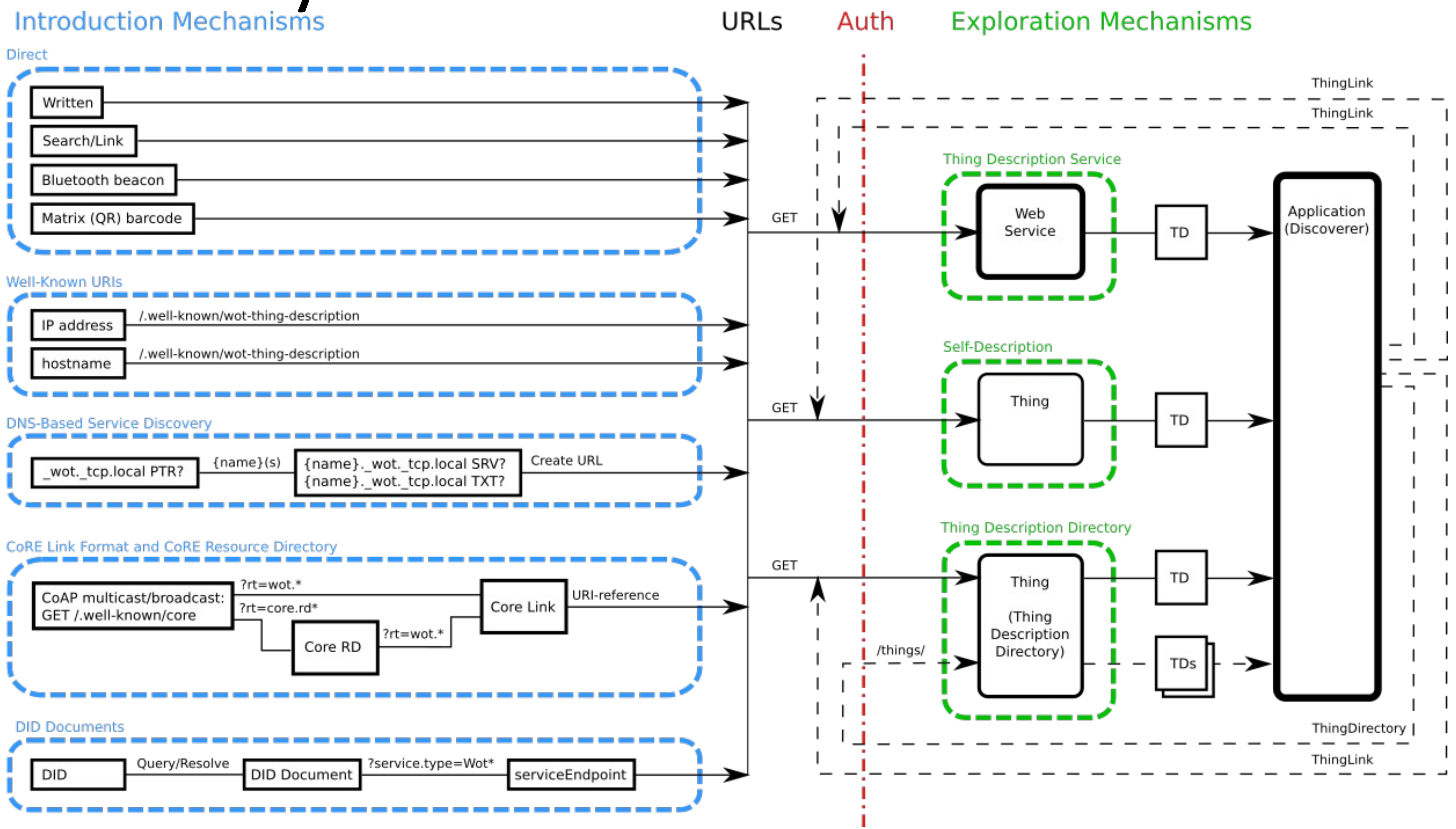
Obtain TD of interest

- Not limited to local network
- Scalable to many TDs
- Need to preserve privacy

Architecture:

- Phased access:
  1. Introduction: open
  2. Exploration: controlled access
- Introduction mechanisms
  - Direct, WK, DNS-SD, CoRE-RD, DID
- Exploration mechanisms
  - TD Server, Self, Directory
- Directories searchable via
  - JSON Path (IETF RFC pending)
  - XPath 3.1
  - SPARQL

Discovery



Introduction Mechanisms
URLs
Auth
Exploration Mechanisms
Direct
Written
Search/Link
Bluetooth beacon
Matrix (QR) barcode
Well-Known URIs
IP address
/.well-known/wot-thing-description
hostname
/.well-known/wot-thing-description
DNS-Based Service Discovery
_wot._tcp.local PTR?
{name}(s)
{name}._wot._tcp.local SRV?
{name}._wot._tcp.local TXT?
Create URL
CoRE Link Format and CoRE Resource Directory
CoAP multicast/broadcast:
GET /.well-known/core
?rt=wot.*
?rt=core.rd*
Core Link
URI-reference
Core RD
?rt=wot.*
DID Documents
DID
Query/Resolve
DID Document
?service.type=Wot*
serviceEndpoint
GET
GET
GET
/things/
ThingLink
ThingLink
Thing Description Service
Web Service
TD
Application (Discoverer)
Self-Description
Thing
TD
Thing Description Directory
Thing
(Thing Description Directory)
TD
TDs
ThingDirectory
ThingLink

Discovery

- Possible future work:
  - Encode static geospatial data in Thing Descriptions
  - Provide ontologies for dynamic geospatial data
  - Index Things by location in TD Directories
  - Provide geolocation filters in Introduction mechanisms
    - e.g. geolocation filters in DNS-SD
  - Discover Things using geospatial queries
    - "find all Things within 100m of lat 49.2803175 long -123.1286623"
    - "find all HVAC sensors in Room 101 of Sheraton Wall Vancouver"
  - Both "quantitative" and "qualitative" locations important
  - Also store sensor data and search based on interval queries
- See:
  - Geolocation proposal and examples
  - Discussion of next WoT charter

Example TDs using Geolocation

- Simple static installation, using schema.org
- Static installation, using modular (proposed) geolocation ontology
- Dynamic installation, using internal geolocation resource
- Dynamic installation, using separate geolocation service
- Dynamic installation, geolocation service

Discovery extensions:

- Spatially-aware introduction mechanisms (e.g. ext. DNS-SD or DID)
- Query mechanisms for Directories (GeoSPARQL and/or JSONPath ext.)

Resources and Contacts

https://www.w3.org/WoT

Dr. Michael McCool

Principal Engineer

Intel

Technology Pathfinding

michael.mccool@intel.com

Dr. Sebastian Kaebisch

Senior Key Expert

Siemens

Technology

sebastian.kaebisch@siemens.com