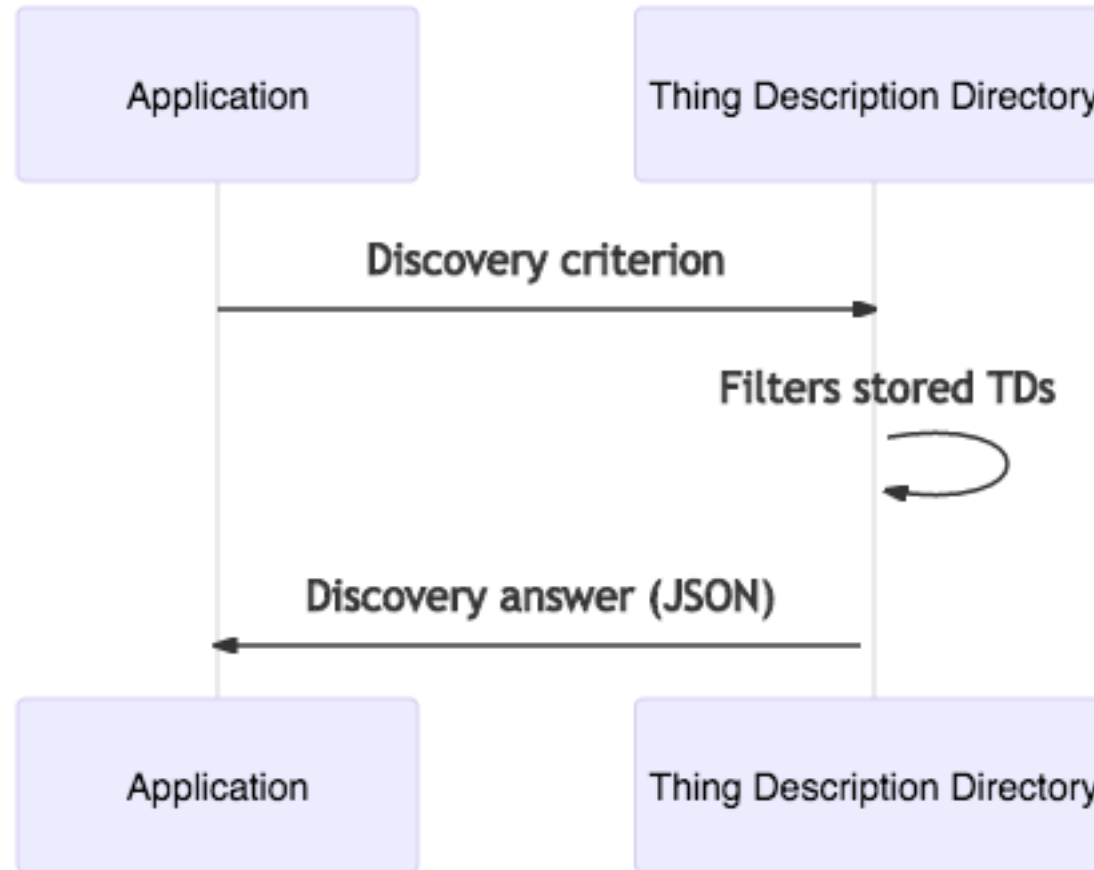# Thing Description Directory Syntactic Discovery Search

Andrea Cimmino

Oct 20, 2020

Discovery in a nutshell

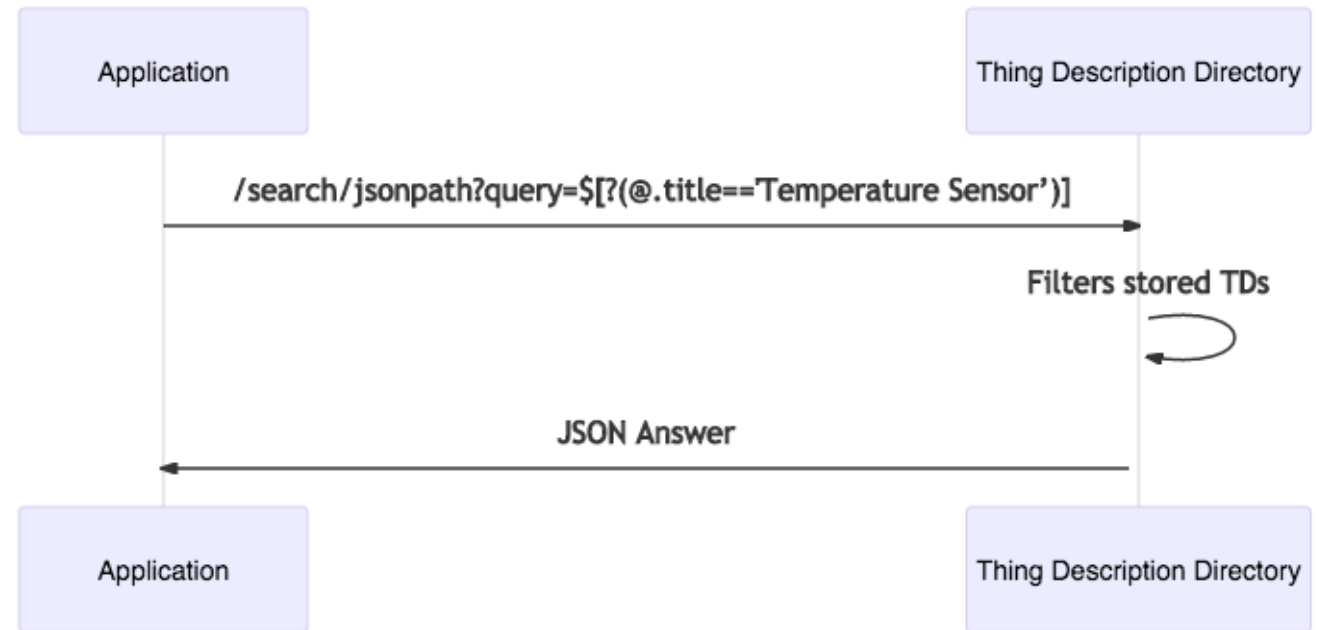# Thing Description Directory (TDD) and discovery

- TDD syntactic discovery criterion:
  - (Syntactic discovery) MUST support JSONPath as discovery criterion
  - (Syntactic discovery) SHOULD support XPath as discovery criterion

- TDD semantic discovery criterion:
  - (Semantic discovery) May support SPARQL as discovery criterion

- TDD discovery answer:
  - A set (array) of Thing Descriptions (TD) meeting the criterion are found
  - A set (array) of Thing Descriptions (TD) fragments fulfilling the criterion are found
  - Discovery results could be paginated (depending on the discovery criterion)

JSONPath syntactic discovery API

- The discovery criterion is JSONPath
  - https://tools.ietf.org/id/draft-goessner-dispatch-jsonpath-00.html
  - Not standard, widely used

- Response code:
  - 200 (Ok) with application/json Content-Type header

- Error codes:
  - 400 (Bad Request): JSONPath expression not provided or contains syntax errors.
  - 401 (Unauthorized): No authentication.
  - 403 (Forbidden): Insufficient rights to the resource.

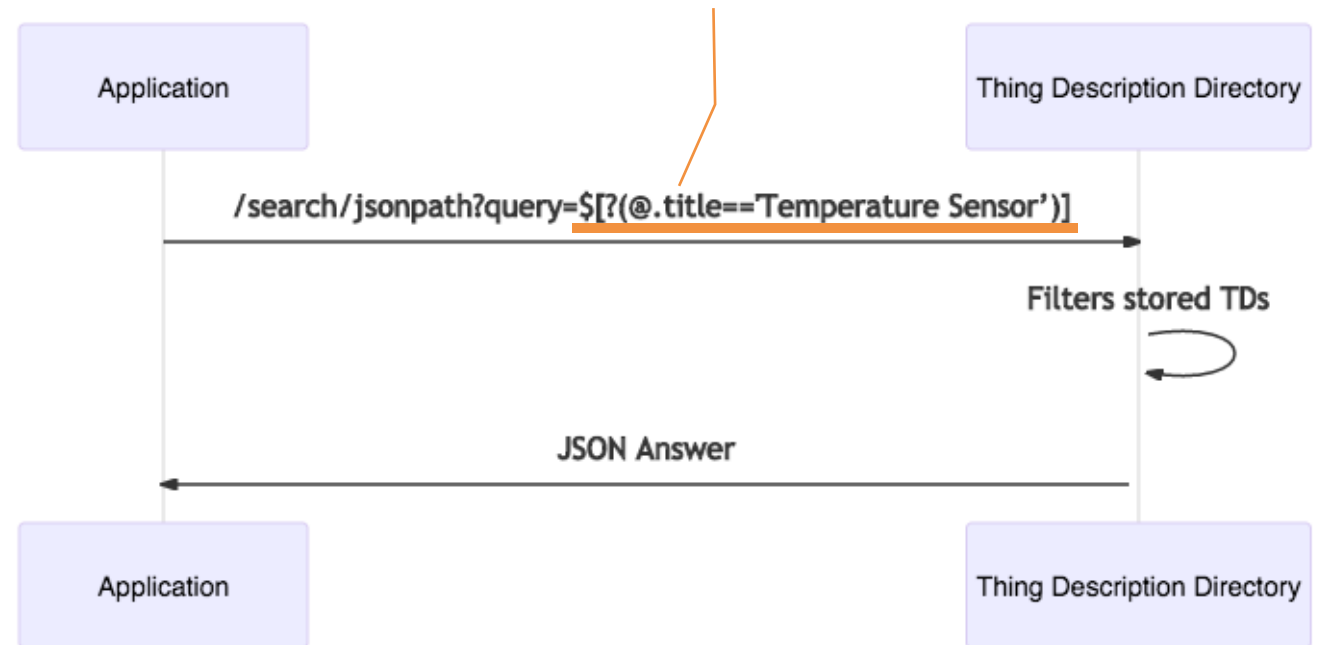# JSONPath syntactic discovery API example

```
...
 "searchJSONPath": {
     "description": "Json Path syntactic search",
     "uriVariables": {
      "query": {
       "title": "A valid Json Path expression",
       "type": "string",
       "format": "iri-reference"
      }
     },
     "forms": [
      {
       "href": "/search/jsonpath?query={query}",
       "htv:methodName": "GET",
       "response": {
        "description": "Success response",
        "htv:statusCodeValue": 200
       },
       "scopes": "search"
      }
     ]
    },
   ...
```

Application

Thing Description Directory

/search/jsonpath?query=$[?(@.title=='Temperature Sensor')]

Filters stored TDs

JSON Answer

Application

Thing Description Directory

JSONPath syntactic discovery API example

```
...
  "searchJSONPath": {
      "description": "Json Path syntactic search",
      "uriVariables": {
        "query": {
          "title": "A valid Json Path expression",
          "type": "string",
          "format": "iri-reference"
        }
      },
      "forms": [
        {
        "href": "/search/jsonpath?query={query}",
        "htv:methodName": "GET",
        "response": {
          "description": "Success response",
          "htv:statusCodeValue": 200
        },
        "scopes": "search"
        }
      ]
    },
    ...
```

Find TDs with title Temperature Sensor

Application

Thing Description Directory

/search/jsonpath?query=$[?(@.title=='Temperature Sensor')]

Filters stored TDs

JSON Answer

Application

Thing Description Directory

JSONPath syntactic discovery API example

WEB OF THINGS

```
...
  "searchJSONPath": {
    "description": "Json Path syntactic search",
    "uriVariables": {
      "query": {
        "title": "A valid Json Path expression",
        "type": "string",
        "format": "iri-reference"
```

Find TDs with title Temperature Sensor

Application

Thing Description Directory

/search/jsonpath?query=$[?(@.title=='Temperature Sensor')]

Filters stored TDs

JSON Answer

Thing Description Directory

Discovery answer is an array of TDs

```
[
  {
    "@context":"https://www.w3.org/2019/wot/td/v1",
    "id":"urn:example:1234",
    "title": "Temperature Sensor"
    ...
  }, {
    "@context":"https://www.w3.org/2019/wot/td/v1",
    "id":"urn:example:9856",
    "title": "Temperature Sensor"
    ...
  }, ....
],
```

Syntactic discovery with XPath

- The discovery criterion is XPath
  - https://www.w3.org/TR/xpath-31/
  - W3C standard

- Response code:
  - 200 (Ok) with application/json Content-Type header

- Error codes:
  - 400 (Bad Request): XPath expression not provided or contains syntax errors.
  - 401 (Unauthorized): No authentication.
  - 403 (Forbidden): Insufficient rights to the resource.
  - 501 (Not Implemented): XPath API not supported.
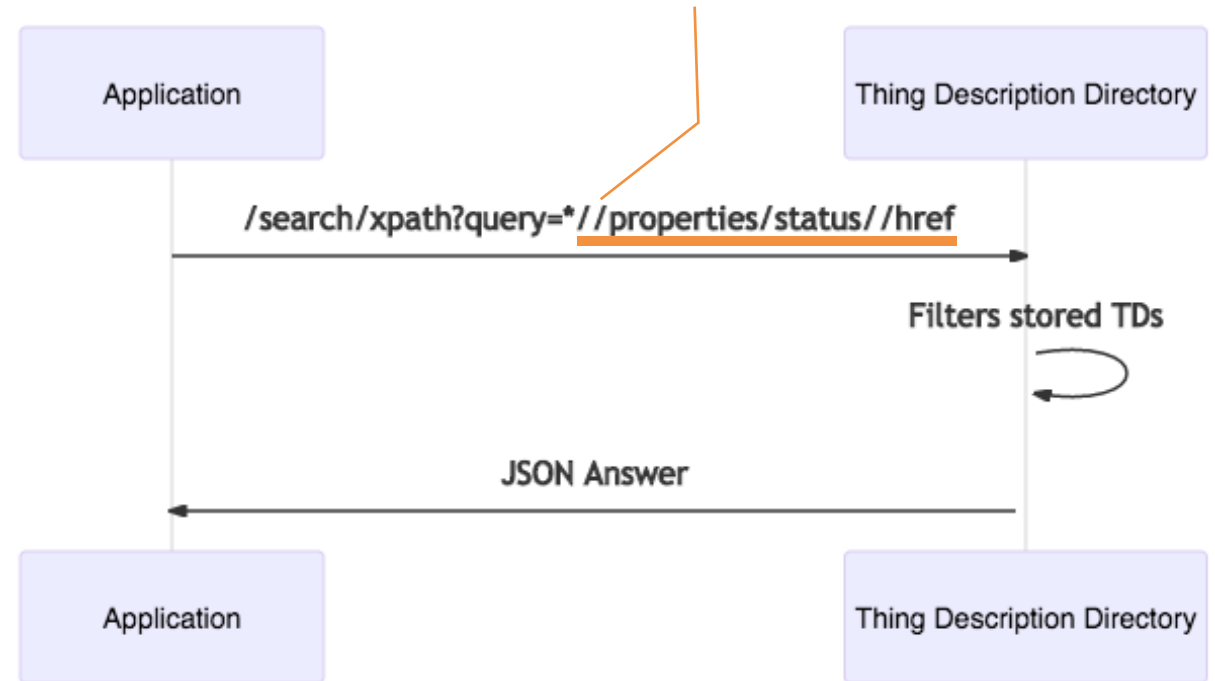
XPath syntactic discovery API example

```
…
"searchXPath": {
      "description": "XPath syntactic search",
      "uriVariables": {
       "query": {
        "title": "A valid XPath expression",
        "type": "string",
        "format": "iri-reference"
       }
      },
      "forms": [
       {
        "href": "/search/xpath?query={query}",
        "htv:methodName": "GET",
        "response": {
         "description": "Success response",
         "htv:statusCodeValue": 200
        },
         "scopes": "search"
       }
      ]
     },
…
```
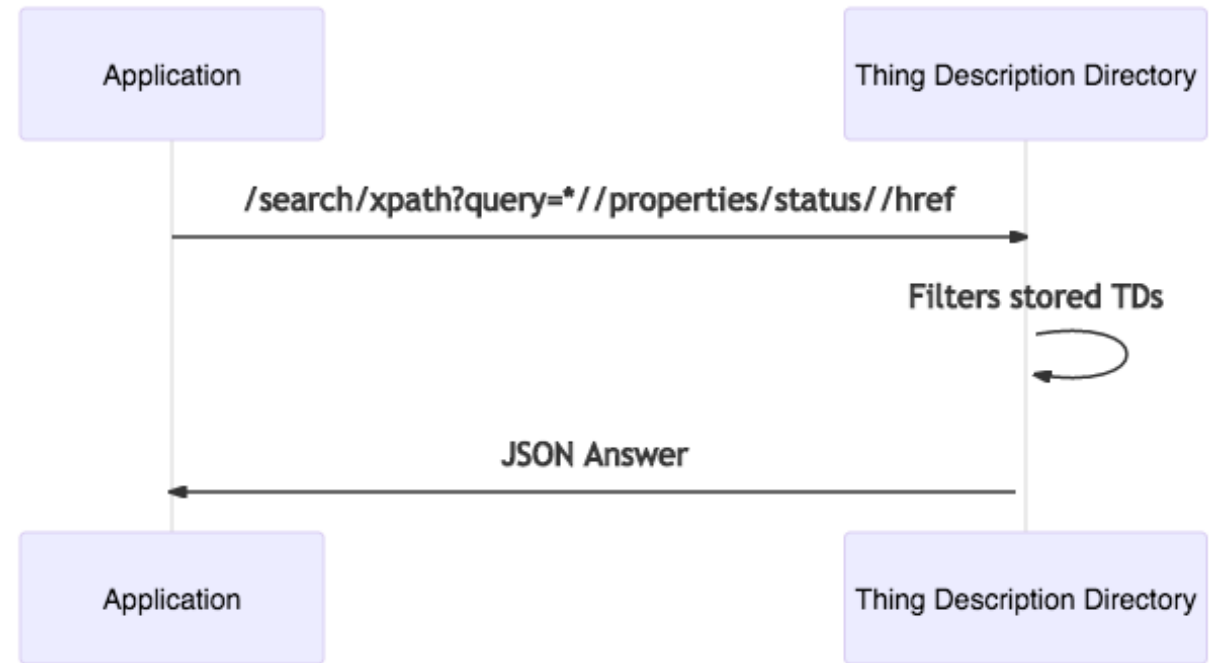
XPath syntactic discovery API example

```
…
"searchXPath": {
    "description": "XPath syntactic search",
    "uriVariables": {
      "query": {
        "title": "A valid XPath expression",
        "type": "string",
        "format": "iri-reference"
      }
    },
    "forms": [
      {
        "href": "/search/xpath?query={query}",
        "htv:methodName": "GET",
        "response": {
          "description": "Success response",
          "htv:statusCodeValue": 200
        },
        "scopes": "search"
      }
    ]
  },
…
```
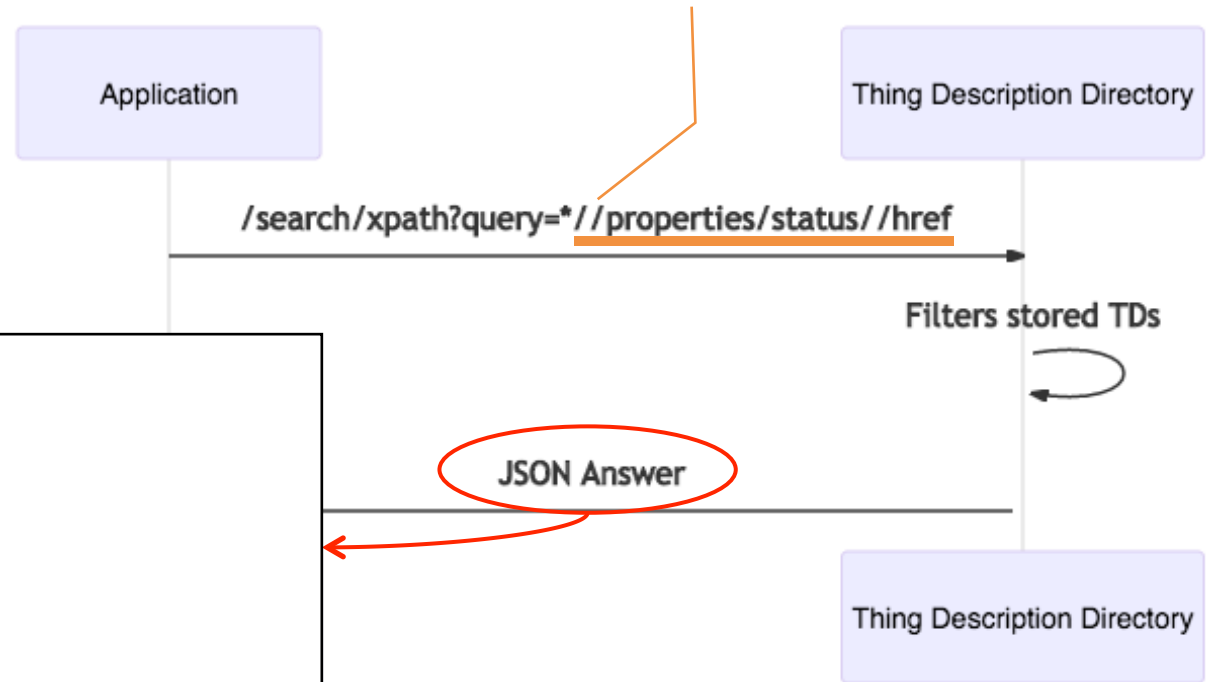
Find values of href in property named status from all TDs

Application → Thing Description Directory

/search/xpath?query=*//properties/status//href

Filters stored TDs

JSON Answer

Application   Thing Description Directory

XPath syntactic discovery API example

```
...
"searchXPath": {
    "description": "XPath syntactic search",
    "uriVariables": {
      "query": {
        "title": "A valid XPath expression",
        "type": "string",
        "format": "iri-reference"
      }
    },
    "forms": [
```

Find values of href in property named status from all TDs

Application

Thing Description Directory

/search/xpath?query=*//properties/status//href

Filters stored TDs

JSON Answer

Thing Description Directory

```
[
    "https://example-1.com/sensor1/status",
    "https://example-2.com/sensor1/status",
    "https://example-3.com/sensor1/status",
    "https://example-4.com/sensor1/status",
    "https://example-5.com/sensor1/status",
]
```

Discovery answer is an array of TD fragments

Example syntactic discovery queries

Description	JSONPath query	XPath query
TDs with title Terrace Temperature Sensor	$[?(@.title=='Terrace Temperature Sensor')]	*[title='Terrace Temperature Sensor']
TDs with title ending with Temperature Sensor	$[?(@.title=~/.*Temperature Sensor/)	*[ends-with(title, 'Temperature Sensor')]
TDs with title ending with Temperature Sensor and created in March 2020	$[?(@.title=~/.*Temperature Sensor/ && @.created=~/2020-03-10/)]	*[ends-with(title, 'Temperature Sensor') and starts-with(created, '2020-03-10')]
TDs with form href values starting with http	not supported by the library	*[*/*/forms/*[starts-with(href, 'http')]]
TDs with version.v:hardware (namespace: v) equal to "1.0"	$[?(@.version.'v:hardware'=='1.0')	??
Second to fourth TD	$[2:4]	*[position()>=2 and position()<4]

More examples: https://github.com/linksmart/thing-directory/wiki/Query-Language

Pros and Cons of JSONPath and XPath

- Pros
  - Short and expressive (XPath and JSONPath)
  - Passed as URL query parameters (XPath and JSONPath)
  - Value/object selection saves both client and server resources
    - (XPath and JSONPath)
  - XPath is a standard

- Cons
  - TD fragments responses are not JSON-LD compliant (XPath and JSONPath)
  - Complex queries can get verbose and messy (XPath and JSONPath)
  - JSONPath not a standard
  - JSONPath may lead to security breaches on some queries

Conclusion

- Syntactic discovery allows to filter registered TDs using:
    - (MUST) JSONPath
    - (SHOULD) XPath

- Discovery answers are always a JSON:
    - Array of TDs
    - Array of TD fragments

- Privacy & Security
    - Discovery is binded to security and privacy policies defined in a TDD