Enabling **Automatic Discovery** and **Querying** of **Web APIs** at **Web Scale** using **LD Standards**

F. Michel, C. Faron-Zucker, O. Corby, F. Gandon

Wimmics* joint research team (Univ. Côte d'Azur, Inria, CNRS, I3S, France)

Linked Data on the Web and its Relationship with Distributed Ledgers (LDOW/LDDL)

13 May, 2019 - San Francisco, USA

* AI in bridging social semantics and formal semantics on the Web
Whom to ask to discover datasets?
No perfect discovery solution

**Restricted scope (topic, format, API...)**
- Need to query multiple resources, accommodate disparate interfaces, mash up results

**Manual dataset/service registration**
- Outdated metadata
- Deprecated services

**Limited relevance of results**
- Keyword-based: many irrelevant results.
- Metadata-based: just a first step in the selection process.

**No detailed insight into the data**
- What resources?
- What properties?
- What relationships?
3 principles to achieve **automatic discovery and consumption** of datasets at **Web scale**

1. **Leverage Web search engines**
   Harvest data portals, exploit structured markup

2. **Rich description of dataset/query services**
   Machine-readable, beyond simple metadata

3. **Rely on well-adopted (simple) standards**
   Need for consensus on technologies and practices
The SPARQL Micro-Service Architecture

Lightweight method to **query a Web API with SPARQL**, and assign dereferenceable URIs to Web API resources.
A SPARQL μ-service is a **CONFIGURABLE** SPARQL endpoint whose **ARGUMENTS** delineate the graph being queried.

**Endpoint:** http://hostname/flickr/getPhotosByTag?tag=bridge

```
SELECT * WHERE {
  ?photo a schema:Photograph;
  schema:name ?title;
  schema:contentUrl ?img.
}
```
Example: search photos by tag using Flickr’s Web API

```
SELECT * WHERE {
  ?photo a schema:Photograph;
  schema:keywords "bridge";
  schema:name ?title;
  schema:contentUrl ?img.
}
```

http://example.org/flickr/getPhotosByTag/
Machine-readable description of a dataset and its SPARQL micro-service
SPARQL Service Description as the framework

SPARQL SD document
http://example.org/flickr/getPhotosByTag/
Metadata about the dataset and micro-service

SPARQL SD document
http://example.org/flickr/getPhotosByTag/
Specification of the graphs produced by the μ-service

**Goal:** give insight into the data, so applications can decide whether the service is relevant for their goal

SPARQL SD document
http://example.org/flickr/getPhotosByTag/
Identification of the Service

SPARQL SD document

http://example.org/flickr/getPhotosByTag/

Goal: give insight into the data, so applications can decide whether the service is relevant for their goal
Description of the **data source** and μ-service **arguments**

SPARQL SD document

http://example.org/flickr/getPhotosByTag/

```json
<> dct:source {
  a schema:WebAPI;
  schema:name "Flickr API";
  schema:url <https://www.flickr.com/services/api/>;
  schema:potentialAction [ 
    a schema:SearchAction, hydra:IriTemplate;
    hydra:template "https://api.flickr.com/?...&tags={tags}".
    hydra:mapping [ 
      hydra:variable "tags";
      hydra:required "true"^^xsd:boolean;
      hydra:property schema:keywords;
    ]
  ]
}
```
Description of the **data source** and μ-service **arguments**

SPARQL SD document
http://example.org/flickr/getPhotosByTag/

```
SELECT * WHERE {
  ?photo a schema:Photograph;
  schema:keywords "bridge";
  schema:name ?title;
  schema:contentUrl ?img.
}
```
Description of the **data source** and **µ-service arguments**

SPARQL SD document
http://example.org/flickr/getPhotosByTag/

```
SELECT * WHERE {
  ?photo a schema:Photograph;
  schema:keywords "bridge";
  schema:name ?title;
  schema:contentUrl ?img.
}
```
A single SPARQL query reasons upon
- the Service Description graph,
- the Shapes graph,
- the input query,

to extract the µ-service arguments
Discover a SPARQL micro-service using web search engines
Discovery using search engines requires a web page
Discovery using search engines requires a web page

Franck MICHEL - Université Côte d'Azur, CNRS, Inria, I3S, France
Web-scale discovery of SPARQL μ-services

*STTL: SPARQL Template Transformation Language
http://ns.inria.fr/sparql-template/
< WRAP-UP >
The global picture

SPARQL SD graph

Shapes graph

SPARQL engine

SPARQL micro-service

HTML

JDON-LD

(1)
The global picture

SPARQL SD graph
Shapes graph
SPARQL engine

SPARQL micro-service

HTML JSON-LD

(1)
The global picture

SPARQL micro-service

SPARQL SD graph
Shapes graph
SPARQL engine

HTML
JSON-LD

(1) search

LD-based application

(2)

Yahoo!
Google
Bing
The global picture

SPARQL micro-service

SPARQL engine

Shapes graph

SPARQL SD graph

HTML

JSON-LD

(1)

(2) search

(3) fetch

LD-based application

(1) JSON-LD

(2) search

(3) fetch
The global picture

SPARQL micro-service

(1) HTML

(2) search

(3) fetch

(4) SPARQL query

(5) Web API

SPARQL engine

Shapes graph

SPARQL SD graph

Franck MICHEL - Université Côte d’Azur, CNRS, Inria, I3S, France
3 principles to achieve **automatic discovery** and **consumption** of SPARQL micro-services at **Web scale**

1. **Leverage Web search engines**
   Machine-readable description to web page + Schema.org/DCAT markup data

2. **Rich description of dataset/query services**
   Metadata, SHACL, Schema.org, Hydra

3. **Rely on well-adopted (simple) standards**
   SPARQL SD, SHACL, Schema.org/DCAT
Perspectives
Perspectives

Demonstrate the whole chain

An application seeks to answer a query / achieve a goal:

- Discovers candidate services using search engines
- Selects relevant services based on SD/Shapes graphs
- Computes & enacts valid compositions

Extend SPARQL federated query engines

- Reason on SPARQL $\mu$-services descriptions
- Query plan respecting services’ inputs
Perspectives

Schema.org unpractical to denote dataset interfaces (API, endpoint…)
- WebAPI type extensions (EntryPoint)
- Convergence with DCAT 1.2 DataService

Google Dataset Search more effective than generic web search engines: more dataset search engines in the future?

Three principles,
many potential architectural/modelling choices in different contexts
Thank-you

Citation:

https://github.com/frmichel/sparql-micro-service
https://hub.docker.com/u/frmichel