Querying Federations of SPARQL Endpoints and JSON-based Web APIs with HeFQUIN

Olaf Hartig [a] and Jonas Westman [b]

(a) Dept. of Computer and Information Science (IDA), Linköping University, Linköping, Sweden
(b) TRATON, Sweden

Abstract

This demonstration illustrates the capability of the HeFQUIN federation engine to integrate JSON-based Web APIs as federation members. Specifically, the engine can run SPARQL queries that join data from such APIs with RDF data from SPARQL endpoints, where the JSON data from the APIs is mapped into RDF at runtime. Instead of imposing a fixed mapping for this process, HeFQUIN supports arbitrary user-defined mappings, which can be described declaratively using the mapping language RML.

Example Query

A SPARQL query that retrieves the current temperature in every city in Germany for which the DBpedia knowledge graph contains geographic coordinates. The query joins data from the DBpedia SPARQL endpoint with live weather data from the Web API of the Open-Meteo service. The PARAMS clause used in this query is a HeFQUIN-specific extension to SPARQL.

```
SELECT * WHERE {
    SERVICE <http://dbpedia.org/sparql> {
      ?city rdf:type dbo:City ;
            dbo:country dbp:Germany ;
            geo:lat ?lat ;
            geo:long ?long .
    }
    SERVICE <http://example.org/OpenMeteoWrapper> PARAMS(?lat AS "latitude", ?long AS "longitude") {
      ?x ex:temperature ?temp .
    }
}
```

Intermediate Solution Mapping

```
?city → dbp:Berlin
?lat  → "52.52"^^xsd:float
?long → "13.41"^^xsd:float
```

Intermediate Solution Mapping

```
?x    → _:b
?temp → "13.1"^^xsd:double
```

Resulting Solution Mapping

```
?city → dbp:Berlin
?lat  → "52.52"^^xsd:float
?long → "13.41"^^xsd:float
?x    → _:b
?temp → "13.1"^^xsd:double
```

Federation Description

A snippet of an RDF-based federation description that introduces the Web API of the Open-Meteo service as a federation member to be considered by HeFQUIN. The description defines the corresponding URI to be used in SERVICE clauses, the URI template for creating the request URIs, and the RML mapping.

```
ex:openMeteo a fd:WrappedFederationMember ;
             fd:serviceURI "http://example.org/OpenMeteoWrapper"^^xsd:anyURI ;
             fd:interface  _:openMeteoRestEndpoint ;
             fd:wrapper    _:openMeteoWrapper .

_:openMeteoRestEndpoint a fd:TemplateBasedInterface ;
    fd:supportedProtocol fd:GenericWebAPIProtocol ;
    fd:uriTemplate [
        hydra:template "https://api.open-meteo.com/v1/forecast?current=temperature_2m{&latitude,longitude}" ;
        hydra:mapping [ hydra:variable "latitude" ; hydra:required true ; fd:paramType xsd:float ] ;
        hydra:mapping [ hydra:variable "longitude" ; hydra:required true ; fd:paramType xsd:float ] ] .

_:openMeteoWrapper fd:rmlTriplesMaps ( _:openMeteoTM1 ) .

_:openMeteoTM1 a rml:TriplesMap ;
    rml:logicalSource [ rml:source _:openMeteoRestEndpoint ;
                        rml:referenceFormulation rml:JSONPath ;
                        rml:iterator "$.current" ] ;
    rml:subjectMap [ rml:termType rml:BlankNode ] ;
    rml:predicateObjectMap [
                rml:predicateMap [ rml:constant ex:temperature ] ;
                rml:objectMap [ rml:reference "temperature_2m" ;
                                rml:datatype xsd:double ] ] .
```

Request URI for the Web API

```
https://api.open-meteo.com/v1/forecast?current=temperature_2m&latitude=52.52&longitude=13.41
```

Response from the API (in JSON format)

```
{
  "latitude" : 52.52,
     ...
  "elevation" : 38.0,
  "current_units" : { "time" : "iso8601",
                      "interval" : "seconds",
                      "temperature_2m" : "°C" },
  "current" : { "time" : "2026-02-28T21:00",
                "interval" : 900,
                "temperature_2m" : 13.1 }
}
```

RDF Representation of the Response (generated at runtime)

```
_:b  ex:temperature  "13.1"^^xsd:double .
```

HeFQUIN Website

HeFQUIN Repo

Acknowledgements
This work has been funded by the Knut and Alice Wallenberg Foundation (KAW 2023.0111) and by Vetenskapsrådet (project 2025-06246).

LINKÖPING UNIVERSITY
DEPARTMENT OF COMPUTER AND INFORMATION SCIENCE