Rewriting SPARQL Queries using an Authorisation Ontology

Dr. Ir. Hans Schevers [a], Ir. Alexandra Rowland (EngD) [a], Dr. Ir. Erwin Folmer[a b], Sven Mol [a c], Ir. Janneke Michielsen [a], Ing. Marc van Andel [a]

[a] *Kadaster, Laan van Westenenk 701, 7334DP, Apeldoorn, The Netherlands*
[b] *HAN University of Applied Sciences, Ruitenberglaan 31, 6826 CC Arnhem, The Netherlands*
[c] *University of Twente, Drienerlolaan 5, 7522 NB Enschede, The Netherlands*

Abstract

Key registers in the Netherlands containing information such as property ownership, persons and commercial registries can be linked and, in doing so, increase the value of this information. However, not all the data contained in these registries is public data, so data access needs to be appropriately managed. In a linked data context, SPARQL endpoints can be used to retrieve data and must implement access rights. While no standardised mechanism exists for the secure handling of linked data information, any defined mechanism should support the free querying inherent to SPARQL while also managing user access to data. This paper describes an experiment to model granular authorisation rules in an ontology and describes a demonstrator that rewrites SPARQL queries in such a way that access rights are included based on this ontology.

Keywords

Authorisation Ontology, SPARQL, SPARQL rewrite, Federative querying, Linked Data

1. Introduction

Kadaster, the Dutch national cadastre and mapping agency, is the governmental agency responsible for the maintenance and publication of information on property rights and ownership in the Netherlands. In recent years, Kadaster has championed the publication of linked data [1]. At present, all key registers maintained by Kadaster containing open data are available as linked data. These open registers have also been integrated and made available as the Kadaster Knowledge Graph (KKG) [2], itself also accessible via a public SPARQL endpoint. The integration of the open key registers greatly improves the ability of a range of users to analyse data across registers in a simple and accessible way. With the improved analytical possibilities and accessibility, there is an increased interest in the integration of these open registers with closed information [3], both maintained by Kadaster and maintained by other government organisations. At present, there are no standardised mechanisms for securing and applying authorisation on SPARQL endpoints. In order to investigate the possibilities of doing so, the Lock-Unlock project was started as a follow-up to the publication of the KKG and focused on investigating and implementing solutions for securely handling closed linked data.

In contrast to the integration of key registers into a single dataset, as done in the KKG, a key requirement for the Lock-Unlock project was the need to secure SPARQL endpoints made available by a range of data providers. This network of registers more closely represents the reality of data provision in the context of the Dutch government than the architecture of the KKG. To support the securing of the SPARQL endpoints, an authorisation ontology was developed; enabling the modelling of granular access rules associated with each register in the network. A prototype demonstrator implements this ontology as part of a SPARQL Rewrite mechanism which primarily controls access to the data available at each endpoint. This paper briefly outlines the project context and then demonstrates the authorisation

14th International Conference on Formal Ontology in Information Systems, July, 2024, Twente, The Netherlands
EMAIL: Hans.Schevers@kadaster.nl (A. 1); Lexi.Rowland@kadaster.nl (A. 2); Erwin.Folmer@kadaster.nl (A. 3); Sven.Mol@kadaster.nl (A. 4); Janneke.Michielsen@kadaster.nl (A. 5); Marc.vanAndel@kadaster.nl(A. 6)
ORCID: 0009-0000-1017-8097 (A. 1); 0000-0002-2339-6357 (A. 2); 0000-0002-7845-1763 (A. 3); 0009-0000-7557-8067 (A. 4); 0009-0006-2946-808X (A. 5);

ontology and its use in the prototype implementation. The conclusion of this paper will include remarks about the feasibility of the SPARQL Rewrite mechanism.

2. Network of Key Registers

In support of the research and development of mechanisms to secure linked data, two assumptions were made at the beginning of the Lock-Unlock project. Firstly, all data made used in the context of this project would be linked data and, therefore, all access to data would be done via (a) SPARQL endpoint(s). Secondly, based on the current working architecture of the system of key registers in the Netherlands, all key registers used in this project would be independent based on the relative independence of the data providers themselves. The following figure (Figure 1) illustrates the system of key registers created as a test environment for this project. Each of the registers, apart from the open Kadaster dataset, are simplified, synthetic versions of the key registers which shares its name. As such, four simplified synthetic datasets were created, namely; the Key Register of Persons (abbreviated in Dutch: BRP) containing a set of fictious persons owning real estate in the municipalities of Almere and Zeewolde; the Key Register Cadastre (abbreviated in Dutch: BRK) containing real estate ownership information, the Commercial Register (abbreviated in Dutch: NHR) containing fictious information about business owners who own real estate as well as the Register of Foundations (abbreviated in Dutch: ANBI) containing fictious information about charities and foundations. The first three registers contain sensitive information and are, therefore, closed datasets which need to be adequately secured. The latter and the final Kadaster dataset contain open information which can be integrated with the closed data for various use cases.

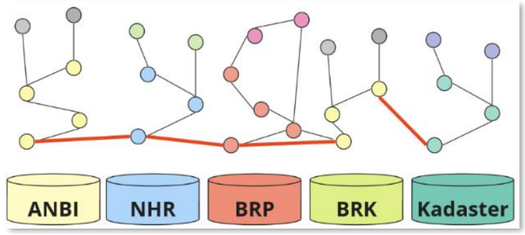


Figure 1: Test network of simplified registries in linked data where synthetic data is linked between registries including ontologies to put the data into a context.

In the figure above (Figure 1), independent linked data graphs containing the synthetic data are connected to the ontologies describing the structure and semantics of each of the registers as well as to each other based on various logical connections between the data. For example, the real estate information made available in the BRK would refer to an owner of that real estate in the BRP. While these connections exist, the data itself is distributed over multiple databases and made available through multiple endpoints, each of which need to be secured based on business rules and requirements defined by the data providers.

3. Authorisation in Linked Data

The key registers made available in the Netherlands can be comprised of entirely open data, closed data or a mixture of both. Access to the latter two categories of data is regulated based on authorisation rules. The most basic authorisation level, which could be implemented, is binary security protocol where access to the entire dataset is provided to authorised users (i.e. dataset-level access control) [4]. In many use cases, more granular access protocols would be preferable. For example, each property owner should have access to information about their own property but not to information about all the properties.

Authorisation of access to the data should be implemented at a subset level (i.e. subset-level access control) based on the verification of personally identifiable information (PII)[1].

Use cases can also be defined which require authorisation protocols to be implemented across the network of registers. For example, a municipality may wish to identify the average age of people who have purchased a house in their municipality over the last five years. This requires access to property ownership information to identify all transactions which occurred in the last five years and access to the birth dates of the owners. The former set of information is available in the BRK maintained as closed data by the Kadaster and the latter is available in the BRP maintained as closed data by Civil Service for Identity Information (Dutch: Rijksdienst voor Identiteitsgegevens). In this use case, authorisation should be granted to two datasets, but the access should be limited to only those persons who purchased property in a given municipality. Binary access control is therefore, not suitable and subset access control is limited only to the silo in which it is implemented. Authorisation to closed data across a distributed network of related registers, particularly government registers heavily influenced and formalised by Dutch law, presents a more detailed set of authorisation requirements.

3.1. Authorisation Requirements

Following the identification of a number of use cases, three high-level requirements were derived for the project. For their derivation, the concepts of horizontal and vertical partitioning[2] were used.

Table 1 Required mechanisms to filter data or to close access to data

No.	Requirement	Description
1	Vertical Data Restrictions	Within a given dataset, or across distributed datasets, user access to properties(predicates) of classes should be controlled.
2	Horizontal Data Restrictions	Within a given dataset, or across distributed datasets, users access to instances(resources) should be controlled.
3	Directional Filtering	Within a given dataset, or across distributed datasets, the direction in which users are able to traverse the graph(s) should be controlled.

Using access to building information as an illustrative example, the first requirement seeks to control user access to the closed properties associated with a given building. Such properties would include the latest purchase price of the building and the name of the current owner(s) of the building. Fulfilling only this requirement would mean that authorised users are able to retrieve purchase prices for every building in the Netherlands. The second requirement seeks to further limit access to only instances to which a user is authorised. Fulfilling this requirement in addition to the first would mean that authorised users are able to only access information about buildings which they own or to which they have judiciary right over such as in the case of a municipality. The final requirement further extends this by ensuring that the linked data graph(s) containing this information cannot be traversed in undesirable manners. Extending the example, it is legal in the Netherlands for authorised users, having paid a small fee, to access ownership information about a given building. It is not, however, legal to retrieve all the buildings owned by a given person without first knowing the address of each building. Having found the owner of a given building, the ability of a user to traverse the relationship in the opposite direction to retrieve all owned buildings should be restricted.

[1] https://www.ibm.com/topics/pii

[2] https://towardsdatascience.com/database-terminologies-partitioning-f91683901716

3.2. Authorisation Ontology

Authentication of users, as deemed to be the first step in providing user access to restricted information, is not specifically investigated within the scope of this project beyond modelling a (logged-in) ‘User’ and its possible relationship with a given ‘Role’ which belongs to a ‘Security Group’ which has a set of ‘Abstract Access Rules’ as part of a basic authentication model. Using this model, rules can be organised according to security groups and these groups can then be used to define the scope that a given role has in the context of data access. In addition to this authentication model, a broader and more detailed authorisation ontology is developed using RDF[3] and OWL[4]. This ontology captures the rules necessary to enable or restrict user access to information. Upon writing this paper, the only known ontology-based access control is presented by Brewster et al. [5]

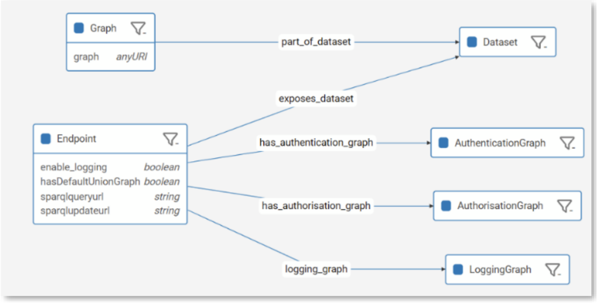
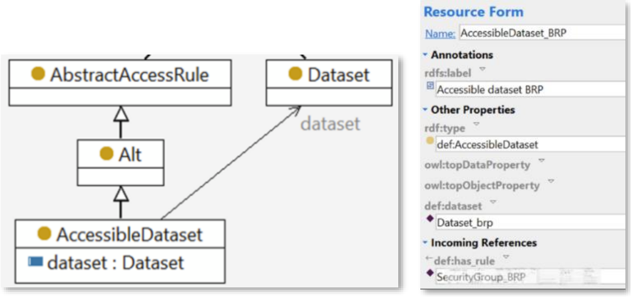


Figure 2: Datamodel of endpoints and datasets

The ‘Endpoint’ class in the above figure (Figure 2) has relationships with a ‘LoggingGraph’ class which can be used to store logging information associated with the endpoint, an ‘AuthenticationGraph’ class and an ‘AuthenticationGraph’ class, each containing the authentication and authorisation data defined for an endpoint respectively. The ‘Endpoint’ exposes the dataset defined using the ‘Dataset’ class where a ‘Graph’ class is a part of this dataset. Using this model as the basis, it is now possible to relate the authentication and authorisation ontologies together. Here, the ‘AbstractAccessRule' class introduced by the authentication ontology can be instantiated via a subclass ‘AccessibleDataset’ (Figure 3) which refers to a given dataset or key register via the ‘dataset’ property.



Resource Form

Name: AccessibleDataset_BRP

Annotations

rdfs:label
Accessible dataset BRP

Other Properties

rdf:type
def:AccessibleDataset

owl:topDataProperty

owl:topObjectProperty

def:dataset
Dataset_brp

Incoming References

def:has_rule
SecurityGroup_BRP

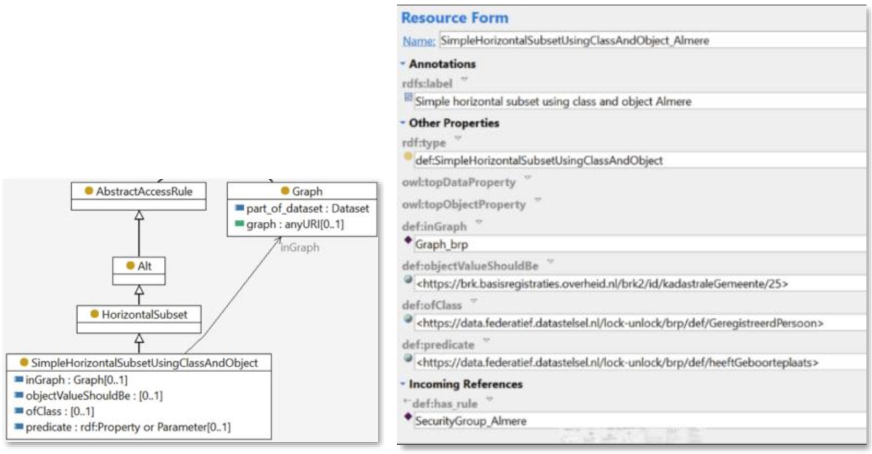
Figure 3: Data model of accessible datasets rule

[3] RDF Primer, https://www.w3.org/TR/rdf-primer/
[4] OWL, https://www.w3.org/TR/owl2-overview/

Because graphs are related to datasets as illustrated in the above figure (Figure 3), it is clear to which graphs such a rule provides access to. In reference to the previously defined requirements, the implementation of this rule allows or restricts access to an entire dataset using binary rules.

In fulfilment of the first requirement, the ‘AbstractAccessRule’ class can also be subclassed with a ‘VerticalRule’ to restrict access to certain predicates (i.e. vertical subsets). For example, predicates for purchase price can be modelled as closed predicates meaning that the values of these predicates are not accessible.

In the figure below (Figure 4), the ‘AbstractAccessRule’ class can again be further subclassed introducing the ‘SimpleHorizontalSubsetUsingClassAndObject’ class, addressing the second requirement in Table 1. The specification of this rule restricts access to data based on a horizontal subset using the predicate ‘objectValueShouldBe’ where the object defined as part of this triple would be the boundary of the horizontal subset in question. For example, access to a dataset could be restricted horizontally in the case of a municipality wishing to access information about persons. The horizontal restrictions would be that said municipality is only allowed to access information about members of their municipality. In this case, the object value would be the municipality in question as denoted in the figure below. Optionally, another restriction could also be defined which ensures that this object or resource is also of a predefined class (i.e. ‘ofClass’).


AbstractAccessRule
Graph
part_of_dataset : Dataset
graph : anyURI[0..1]
inGraph
Alt
HorizontalSubset
SimpleHorizontalSubsetUsingClassAndObject
inGraph : Graph[0..1]
objectValueShouldBe : [0..1]
ofClass : [0..1]
predicate : rdf:Property or Parameter[0..1]

Resource Form

Name: SimpleHorizontalSubsetUsingClassAndObject_Almere

Annotations

rdfs:label

Simple horizontal subset using class and object Almere

Other Properties

rdf:type

def:SimpleHorizontalSubsetUsingClassAndObject

owl:topDataProperty

owl:topObjectProperty

def:inGraph

Graph_brp

def:objectValueShouldBe

<https://brk.basisregistraties.overheid.nl/brk2/id/kadastraleGemeente/25>

def:ofClass

<https://data.federatief.datastelsel.nl/lock-unlock/brp/def/GeregistreerdPersoon>

def:predicate

<https://data.federatief.datastelsel.nl/lock-unlock/brp/def/heeftGeboorteplaats>

Incoming References

def:has_rule

SecurityGroup_Almere

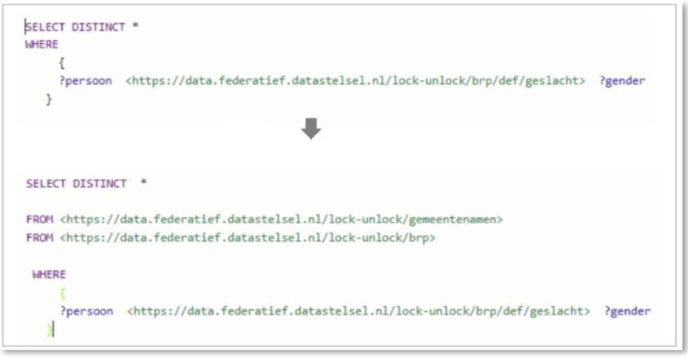
Figure 4: Datamodel and example configuration for horizontal filtering: all resources of the class (../GeregistreedPersoon) must have a ‘../heeftGeboorteplaats’ of ‘../kadastraleGemeente/25’

While the previous modelling approaches support access control to both horizontal and vertical subsets, a new Rule for directional filtering is necessary. Within this project no attempt has been made to implement this.

4. SPARQL-Rewrite Implementation Experiments

For experimental purposes, an implementation has been made of the authorisation ontology. This implementation focuses on the rewriting of SPARQL queries to include or ‘inject’ the constraints necessary to restrict user access to the data being queries. The starting principle behind this implementation is that users of a SPARQL endpoint should be able to freely query the endpoint in line with the principles of linked data and that the user query should be ‘rewritten’ to constrain access to the data available at the endpoint based on the role and security groups to which a user belongs. The rewriting of SPARQL queries was done based on different categories of constraints.

The first category, and arguably the easiest way to constrain user access to data using an injection into a SPARQL query, is based on graph access. In the figure below (Figure 5), a user first writes a SPARQL query which queries for information about persons and their gender. The SPARQL query is then rewritten by adding various 'FROM' statements which add a series of named graphs. An instantiated authorisation ontology can be referenced to check whether a user does indeed have access to a set of graphs based on their role and security group. This implementation supports the fulfilment of the first requirement given that graphs contain subsets of information available in a given dataset.

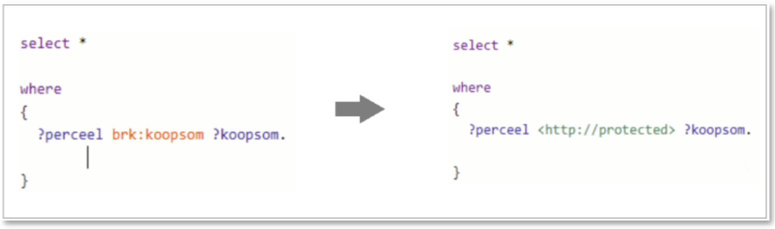


```
SELECT DISTINCT *
WHERE
     {
     ?persoon  <https://data.federatief.datastelsel.nl/lock-unlock/brp/def/geslacht>  ?gender
   }
```

```
SELECT DISTINCT  *

FROM <https://data.federatief.datastelsel.nl/lock-unlock/gemeentenamen>
FROM <https://data.federatief.datastelsel.nl/lock-unlock/brp>

 WHERE
     {
     ?persoon  <https://data.federatief.datastelsel.nl/lock-unlock/brp/def/geslacht>  ?gender
   }
```

Figure 5: Rewritten SPARQL query limiting graph access

A simple approach for implementing vertical filtering is to rewrite the SPARQL query by replacing the predicate that is defined as protected in the authorisation ontology. In the figure below (Figure 6), the user writes a SPARQL query requesting information about the purchase value ('koopsom') of a given parcel. Given that this predicate is modelled as being protected using the authorisation ontology, a replacement predicate is injected into the users SPARQL query ('protected'). This query will now not return any results because no data will be present using this pattern because the 'protected' predicate is not present in the data.



```
select *

where
{
  ?perceel brk:koopsom ?koopsom.

}
```

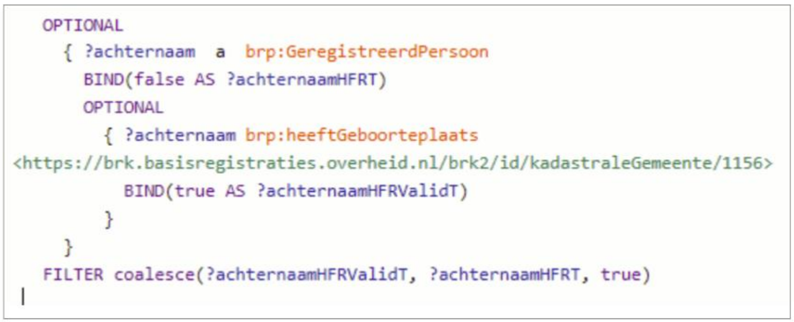
```
select *

where
{
  ?perceel <http://protected> ?koopsom.

}
```

Figure 6: Rewrite of a SPARQL query containing a protected predicate

When a parameter is used as a predicate then an extra filter is necessary to make sure that this parameter never contains the restricted predicate. For more complex cases like for example the usage of the protected predicate within a FILTER NOT EXIST this approach might be oversimplistic.

For horizontal filters, each subject and object in a SPARQL query (resource or parameter) needs to be checked to see whether they meet the horizontal requirements. First, a type-check determines if horizontal filtering is necessary. The type-check uses the 'ofClass' restriction modelled in the authorisation ontology. If so, then a mandatory relation implements the horizontal filter. The following SPARQL snippet shows the injection necessary for a simple horizontal filter for the parameter

‘?achternaam’ (see Figure 7 below). Similarly, additions are made for each subject and object in a SPARQL query. This approach could work for simple situations or if all resources have an easy way of identifying if they are restricted or not. Adding extra data so that all resources are annotated with extra security information makes horizontal filtering more feasible by using the SPARQL rewrite method.



```
    OPTIONAL
      { ?achternaam  a  brp:GeregistreerdPersoon
        BIND(false AS ?achternaamHFRT)
        OPTIONAL
          { ?achternaam brp:heeftGeboorteplaats
<https://brk.basisregistraties.overheid.nl/brk2/id/kadastraleGemeente/1156>
            BIND(true AS ?achternaamHFRValidT)
          }
      }
    FILTER coalesce(?achternaamHFRValidT, ?achternaamHFRT, true)
  |
```

Figure 7: SPARQL addition snippet for horizontal filtering

For most rules further investigation needs to be done to ensure the robustness of this solution.

5. Demonstrator

To demonstrate the implementation of the authorisation ontology for a SPARQL rewrite mechanism, a demonstrator application and test environment was created. For the three registers where synthetic data was required, this was generated and placed in an individual triplestore. The two open registers were also made available at individual SPARQL endpoints. Jena/Fuseki triplestore[5] were used in all cases and the instantiated authorisation implementation was made available as a graph in each endpoint.

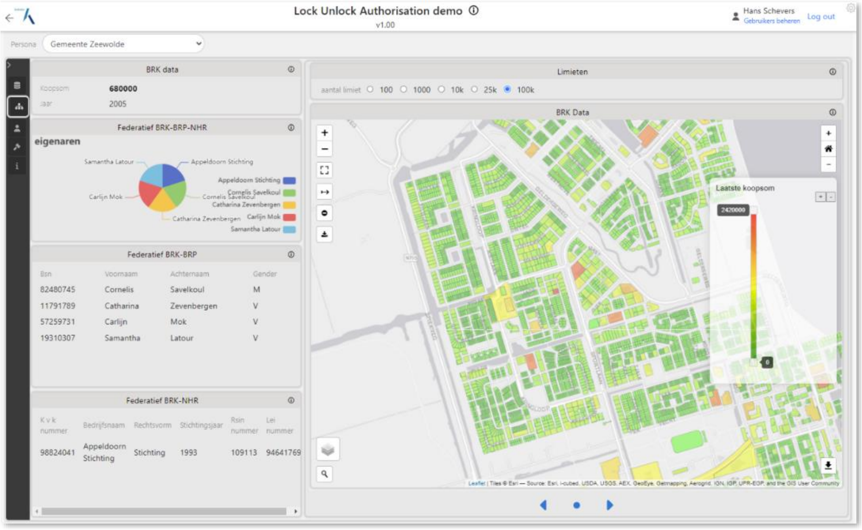
While authentication on these endpoints was not implemented, an extra URL parameter is supplied to mimic the login of certain users and to be able to switch between users and their roles and security restrictions for demo purposes.

In the screenshot below (Figure 8), a dashboard is shown querying and visualizing data from four different triplestores in the network of registers. For the user, in this case the Municipality of Zeewolde, four different authorisation rules have been developed giving access to parts of the data residing in different triplestores. In this case, the user has full access to property information but horizontally limited to only the municipality in question. Consequently, a very general SPARQL query retrieving all parcels is limited to only the parcels in the Municipality of Zeewolde. (i.e. horizontal subset restriction). Selecting an individual parcel retrieves data like the last purchase price and its owners by name and BSN number. This is possible as this user has access to a horizontally limited BRP dataset and a horizontally limited BRK dataset.

[5] Apache Jena, https://jena.apache.org/

heeft opmaak toegepast: Spaans (Spanje)

heeft opmaak toegepast: Spaans (Spanje)



Lock Unlock Authorisation demo

v1.00

Hans Schevers

Persona: Gemeente Zeewolde

BRK data

Koopsom: **680000**

Jaar: 2005

Federatief BRK-BRP-NHR

eigenaren

Federatief BRK-BRP

Bsn	Voornaam	Achternaam	Gender
82480745	Cornelis	Savelkoul	M
11791789	Catharina	Zevenbergen	V
57259731	Carlijn	Mok	V
19310307	Samantha	Latour	V

Federatief BRK-NHR

Kvk nummer	Bedrijfsnaam	Rechtsvorm	Stichtingsjaar	Rsin nummer	Lei nummer
98824041	Appeldoorn Stichting	Stichting	1993	109113	94641769

Limieten

aantal limiet ○ 100 ○ 1000 ○ 10k ○ 25k ● 100k

BRK Data

Gebruikers beheren
Log out
Laatste koopsom
2420000
0
Leaflet | Tiles © Esri — Source: Esri, i-cubed, USDA, USGS, AEX, GeoEye, Getmapping, Aerogrid, IGN, IGP, UPR-EGP, and the GIS User Community

Figure 8: Screenshot demonstrator showing data from multiple triplestores that are secured using the SPARQL-rewrite implementation.

6. Closing remarks

As part of the Lock-Unlock project, a continuation of the Kadaster Knowledge Graph focusing on securing closed linked data, an experimental demonstrator has been implemented based on the development of an authorisation ontology. It comprises of triplestores containing synthetic linked data mimicking simplified closed registers available in the Netherlands and connections between these datasets where available. Each triplestore has its own authorisation graph, an instantiation of the authorisation ontology, securing access to the data by rewriting incoming SPARQL queries which add the necessary constraints. A demonstrator application with the ability to quickly switch between users showed that the application can work giving each user a different view on the data.

Some SPARQL rewrite patterns are clear and are easy to implement. More advanced rules like horizontal and arguably vertical restrictions seem feasible but definitely need more formalization and arguably also more fundamental research. Additionally, several issues are out of scope of this project including, authentication, testing the robustness of the application in all edge cases, the impact of reasoning on this security approach, potential ways in which to circumvent the security approach and more.

The research and the implementation of the demonstrator of this project has been exploratory. This means that more research and development is necessary to further this research and that no conclusive results can be drawn from this research. However, the authors of this paper clearly see the explained approach as a very potential for securing federated SPARQL queries. It is hope that it is possible to further engage the scientific community on this topic.

7. Acknowledgements

We gratefully acknowledge the support of RealisatieIBDS[6] for funding this research.

8. References

[1] Interoperability and Integration: An Updated Approach to Linked Data Publication at the Dutch Land Registry. / Rowland, Alexandra; Folmer, Erwin; Beek, Wouter et al.
In: ISPRS international journal of geo-information, Vol. 11, No. 1, 51, 10.01.2022.

[2] Ronzhin, S.; Folmer, E.; Maria, P.; Brattinga, M.; Beek, W.; Lemmens, R.; van't Veer, R. Kadaster Knowledge Graph: Beyond the Fifth Star of Open Data. Information 2019, 10, 310. https://doi.org/10.3390/info10100310

[3] A. Landi et al., "The 'A' of fair – as open as possible, as closed as necessary," Data Intelligence, vol. 2, no. 1–2, pp. 47–55, Jan. 2020. doi:10.1162/dint_a_00027

[4] L. Costabello, S. Villata, O. Rodriguez Rocha, and F. Gandon, "Access control for HTTP operations on Linked Data," The Semantic Web: Semantics and Big Data, pp. 185–199, 2013. doi:10.1007/978-3-642-38288-8_13

[5] Christopher Brewster, Barry Nouwt, Stephan Raaijmakers, Jack Verhoosel; Ontology-based Access Control for FAIR Data. Data Intelligence 2020; 2 (1-2): 66–77. doi: https://doi.org/10.1162/dint_a_00029

[6] https://realisatieibds.nl/