SHACL-ACL: Access Control with SHACL

Philipp D. Rohde[1,2,3][0000−0002−9835−4354], Enrique Iglesias[2][0000−0002−8734−3123], and Maria-Esther Vidal[1,2,3][0000−0003−1160−8727]

[1] TIB Leibniz Information Centre for Science and Technology, Hannover, Germany
{philipp.rohde,maria.vidal}@tib.eu
[2] L3S Research Center, Hannover, Germany
iglesias@l3s.de
[3] Leibniz University of Hannover, Hannover, Germany

Abstract. The number of publicly accessible knowledge graphs is increasing and so are their applications. Knowledge graphs may contain private data and need to be protected against unauthorized access. There are different approaches for access control to knowledge graphs, e.g., user-based or policy-based. User-based access control can be hard to maintain in systems with hundreds or even thousands of users. In contrast, policy-based approaches use rules to decide whether the access should be granted or denied. ODRL is designed for licensing but also used for policy-based access control. Hence, the evaluation of access policies is not defined and no external data can be considered during the decision-making process. Policies can be seen as integrity constraints and, hence, it is natural to specify them in SHACL; the semantics of SHACL validation are well-defined. SHACL-ACL demonstrates how SHACL can be utilized in a policy-based access control approach. Furthermore, utilizing RML mappings, SHACL-ACL is capable of considering data from various heterogeneous sources for the policy evaluation, e.g., JSON data from Web APIs. The demo is available as an interactive Jupyter notebook.

Keywords: Access Control · Privacy · SHACL

1 Introduction

Knowledge graphs are used more and more to publish data on the Web [7]. The data of a knowledge graph is commonly expressed in the *Resource Description Framework* (RDF) [11,12]. When it comes to sharing private data over the Web, security comes into play. *Solid Pods* [1] are one possibility to control the access to one's private data. Solid relies on OpenID [15] and usually grants access to a resource on a per-user basis, i.e., for each resource, the access rights of each user need to be set. In contrast, access policies define conditions for the access, e.g., access is granted during the night or on rainy days. The *Open Digital Rights Language* (ODRL) [8] is designed with licensing in mind but also used for access control in some projects [4,5]. Since the access is granted when all access policies are fulfilled, access policies can be seen as integrity constraints for access control.

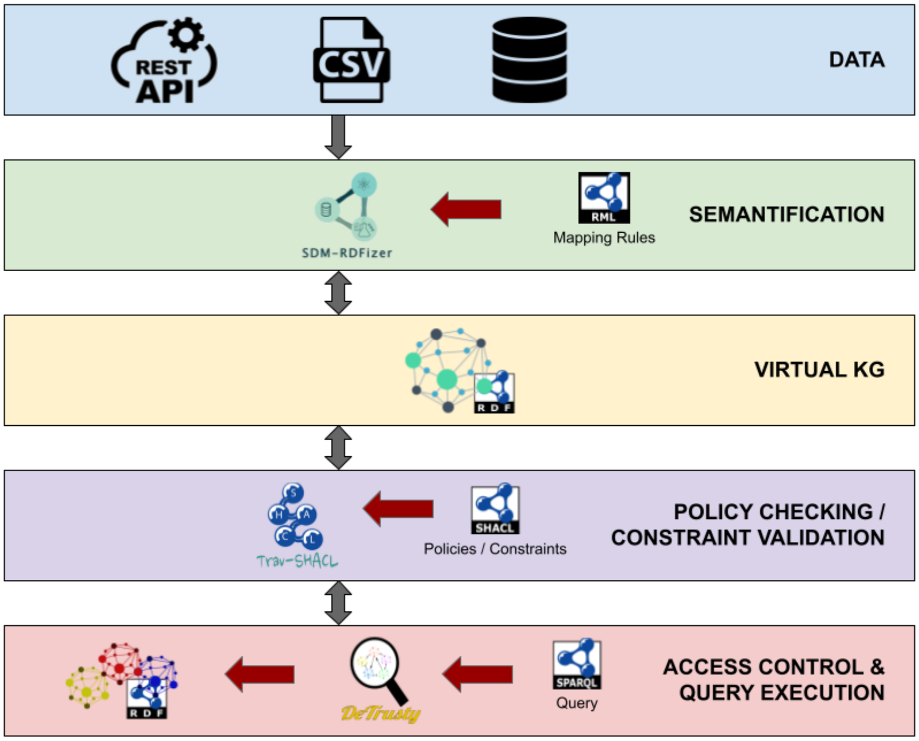

Fig. 1. The SHACL-ACL Architecture. With the use of RML mappings, data from various heterogeneous sources are semantified into a virtual knowledge graph (KG). The virtual knowledge graph is then validated against the access control policies specified in SHACL. If all requirements are met, the SPARQL query is executed. After the execution of the query, the query result is presented to the user.

This paper demonstrates *SHACL Access Control Lists* (SHACL-ACL), a framework able to grant access to RDF knowledge graphs based on access control policies defined in the *Shapes Constraint Language* (SHACL) [10]. While the evaluation of ODRL policies is not explicitly explained in the specification, the semantics of SHACL validation are well-defined [2]. Since ODRL is designed for licensing instead of access control, only data from the policy or known to the evaluation system can be considered during the evaluation of ODRL policies. SHACL-ACL overcomes this limitation by utilizing the *RDF Mapping Language* (RML) [3] to create a virtual RDF knowledge graph on the fly. This allows the consideration of external data, e.g., JSON data from a Web API, during the decision-making process. Hence, decisions can be based on up-to-date data. The virtual knowledge graph is then evaluated by a SHACL validator; following the known semantics for SHACL validation. If all requirements are fulfilled, the access to the resource is granted. Once the access is granted, a SPARQL query posed by the user can be executed and the results presented.

2 Access Control with SHACL

As mentioned in the introduction, access control policies can be seen as integrity constraints for access control. Hence, SHACL can be used for controlling the access to RDF knowledge graphs. The SHACL-ACL architecture for controlling the access to RDF sources for the execution of SPARQL (*SPARQL Protocol And RDF Query Language*) [13] queries using access control policies defined in SHACL is shown in Figure 1. Once the user poses a SPARQL query to the system, the access control policies need to be checked. Since SHACL requires the data to be validated to be in RDF, first, a virtual knowledge graph is created from various heterogeneous sources that contain the data necessary for the policy checking. The creation of the virtual knowledge graph utilizes RML mappings to semantify the data sources on-the-fly. Then a SHACL validator is used to check the virtual knowledge graph against the policies. After checking the policies, a decision is made whether the query can be executed. If all requirements are met, the access is granted and the query engine executes the query. If not, the access is denied and an error message is returned to the user.

The architecture presented in Figure 1 can be implemented using existing tools. The provided implementation[4] relies on the SDM-RDFizer [9] for creating the virtual knowledge graph. The SHACL validator used is Trav-SHACL [6]. For executing SPARQL queries, the federated query engine DeTrusty [14] is utilized. These tools are state-of-the-art and implemented in Python.

3 Demonstration of Use Case

To demonstrate the application of SHACL-ACL[5], an access control policy is defined in SHACL as well as a SPARQL query over the data from the World Bank. The World Bank knowledge graph comprises 250,097,215 RDF triples stating per year and country the value of several indicators, e.g., life expectancy, population, inflation, and age distribution. The knowledge graph contains the data for 1,436 different indicators for 265 countries covering the years 1960 to 2021. The average number of indicators per country per year is 711.225 which implies that not all indicators are recorded for all countries for all years. One reason is that some indicators are not yet available for 2021, e.g., the life expectancy in Germany. The query for the demonstration retrieves the life expectancy in Germany for the last three years available, i.e., 2018 to 2020. The access control policy considers local conditions, i.e., conditions of the machine of the user like CPU usage, available RAM, and local time. Additionally, the weather conditions in Hannover (Germany) are considered. Access is granted if: i) the CPU usage is below 30%, ii) at least 80% of RAM are available, iii) it is night time, i.e., 7 pm to 6 am, iv) the temperature in Hannover (Germany) is below 25°C, and v) the humidity in Hannover (Germany) is at least 75% (see Table 1).

[4] The code is available at https://github.com/SDM-TIB/SHACL-ACL

[5] The live demo is available at https://mybinder.org/v2/gh/SDM-TIB/SHACL-ACL/HEAD?labpath=SHACL-ACL.ipynb

Table 1. Access Control Policy and Simulated Data Overview. Data values for the *current conditions* use case are omitted since they depend on the time of execution.

	Policy	Invalid Conditions	Valid Conditions
Time	19:00:00 to 06:00:00	09:09:09	20:15:36
CPU Usage	$< 30\%$	0.4%	20.5%
RAM Available	$\geq 80\%$	50.50%	86.21%
Temperature	< 25°C	0.6°C	9.1°C
Humidity	$\geq 75\%$	99%	87%

Current Conditions. The first use case collects the current conditions of the machine executing the demonstration and gathers the current weather data of Hannover (Germany) via a Web API. For this, a modified version of the SDM-RDFizer [9] is used. The modifications include data collection from Web APIs as well as returning a virtual knowledge graph instead of a file containing the RDF triples. The virtual knowledge graph is validated against the access control policies mentioned above using Trav-SHACL [6]. This use case demonstrates the capability of gathering live and external data for the policy evaluation. Since the result of the decision-making process cannot be guaranteed, two additional use cases with static data are presented.

Invalid Conditions. This use case uses static data for the policy evaluation that is known to violate the access control policy (see Table 1). More precisely, the time in the data is 9 am and only about 50% of the RAM are available. Hence, the time and memory policy are violated. All other conditions are met. Due to the violations, the access is denied and the query is not executed. An error is returned stating that the access was denied.

Valid Conditions. The static data for this use case ensures that the policy evaluation succeeds without any violations, i.e., all the constraints are fulfilled (see Table 1); it is night time, the machine is under a low load, and it is cold and humid in Hannover (Germany). Since no violations are detected, SHACL-ACL grants access to the World Bank knowledge graph and the SPARQL query is executed. After the execution of the query, the query result is shown to the user presenting the life expectancy in Germany for the last three years (in the data).

4 Conclusion

SHACL-ACL demonstrates the use of SHACL as a language to define access control policies which can be seen as integrity constraints. In contrast to ODRL, SHACL is defined for validating constraints over KGs. Additionally, the use of RML mappings allows to generate a virtual KG from various heterogeneous sources, i.e., local and external. SHACL-ACL relies on widespread concepts that are well-known in the Semantic Web community and is capable of controlling the access to resources on the Web, e.g., a SPARQL endpoint with private data.

Acknowledgements This work has been partially supported by the EU H2020 RIA funded project CLARIFY (grant agreement No 875160) and the Federal Ministry for Economic Affairs and Energy of Germany (BMWK) in the project CoyPu (project number 01MK21007[A-L]).

References

1. Capadisli, S., Berners-Lee, T., Verborgh, R., Kjernsmo, K.: Solid Protocol. W3C Solid Community Group Submission (12 2021), https://solidproject.org/TR/2021/protocol-20211217
2. Corman, J., Reutter, J.L., Savković, O.: Semantics and Validation of Recursive SHACL. In: The Semantic Web – ISWC 2018 (2018). https://doi.org/10.1007/978-3-030-00671-6_19
3. Dimou, A., Vander Sande, M., Colpaert, P., Verborgh, R., Mannens, E., Van de Walle, R.: RML: A generic language for integrated RDF mappings of heterogeneous data. In: Proceedings of the Workshop on Linked Data on the Web co-located with WWW (2014), https://ceur-ws.org/Vol-1184/ldow2014_paper_01.pdf
4. Esteves, B., Pandit, H.J., Rodríguez-Doncel, V.: ODRL Profile for Access Control. Working Draft (12 2022), https://protect.oeg.fi.upm.es/odrl-access-control-profile/oac.html
5. Esteves, B., Rodríguez-Doncel, V., Pandit, H.J., Mondada, N., McBennett, P.: Using the ODRL Profile for Access Control for Solid Pod Resource Governance. In: The Semantic Web: ESWC 2022 Satellite Events. pp. 16–20. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-11609-4_3
6. Figuera, M., Rohde, P.D., Vidal, M.E.: Trav-SHACL: Efficiently Validating Networks of SHACL Constraints. In: The Web Conference (2021). https://doi.org/10.1145/3442381.3449877
7. Hogan, A., Blomqvist, E., Cochez, M., D'amato, C., Melo, G.D., Gutierrez, C., Kirrane, S., Gayo, J.E.L., Navigli, R., Neumaier, S., et al.: Knowledge graphs. ACM Computing Surveys **54**(4), 1–37 (Jul 2021). https://doi.org/10.1145/3447772
8. Ianella, R., Villata, S.: ODRL Information Model 2.2. W3C Recommendation (02 2018), https://www.w3.org/TR/2018/REC-odrl-model-20180215/
9. Iglesias, E., Jozashoori, S., Chaves-Fraga, D., Collarana, D., Vidal, M.E.: SDM-RDFizer: An RML Interpreter for the Efficient Creation of RDF Knowledge Graphs. In: CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020 (2020). https://doi.org/10.1145/3340531.3412881
10. Knublauch, H., Kontokostas, D.: Shapes Constraint Language (SHACL). W3C Recommendation (07 2017), https://www.w3.org/TR/2017/REC-shacl-20170720/
11. Lassila, O., Swick, R.R.: Resource Description Framework (RDF) Model and Syntax Specification. W3C Recommendation (02 1999), https://www.w3.org/TR/1999/REC-rdf-syntax-19990222/
12. Manola, F., Miller, E.: RDF Primer. W3C Recommendation (02 2004), https://www.w3.org/TR/2004/REC-rdf-primer-20040210/
13. Prud'hommeaux, E., Seaborne, A.: Shapes Constraint Language (SHACL). W3C Recommendation (2008), https://www.w3.org/TR/rdf-sparql-query/
14. Rohde, P.D.: DeTrusty v0.11.2 (02 2023). https://doi.org/10.5281/zenodo.7670670
15. Sakimura, N., Bradley, J., Jones, M., de Medeiros, B., Mortimore, C.: OpenID Connect Core 1.0 incorporating errata set 1. OpenID (11 2014), https://openid.net/specs/openid-connect-core-1_0.html