LOAMA: Low-code ODRL Access Management Application

Wout Slabbinck[1,*], Lennert De Rouck[1], Joachim Van Herwegen[1], Wouter Termont[1], Beatriz Esteves[1] and Ruben Verborgh[1]

[1]IDLab, Department of Electronics and Information Systems, Ghent University - imec, Belgium

Abstract

State-of-the-art authorization mechanisms have so far focused on dealing with data management, while leaving policy management and enforcement as an afterthought. Considering that the latter are of the utmost importance to deal not only with low-level technical requirements, but also crucial to deal with legal or economic requirements, we introduce LOAMA, the Low-code ODRL Access Management Application, which can be used to manage ODRL policies in decentralized settings through an Authorization Server. In this paper, beyond a demonstration of the LOAMA User Interface, we provide an overview of the LOAMA architecture, which is based on the User-Managed Access (UMA) specifications. LOAMA abstracts the complexity of managing policies, by providing a tool that people not familiar with policy languages can use to manage their preferences regarding access management. Future works includes the expansion of the LOAMA UI to support further ODRL constraints, e.g., to express purpose-based or temporal usage control policies.

Keywords

access and usage control, policy management, User-Managed Access, ODRL

1. Introduction

Discussions on authorization mechanisms for decentralized (data) spaces exhibit a certain tension between high-level conceptual (legal or economic) requirements and low-level technical mechanisms, precluding the emergence of a broadly accepted integration between the two. New authorization frameworks are typically developed in the context of emerging data (exchange) technologies, which focus predominantly on the resource level (i.e., the data plane), rather than on interoperability in policy management and enforcement (i.e., the control plane). New conceptual frameworks, formulated in a corporate or political context using non-technical terminology, often fail to gain a foothold in technical implementation. Together, this results

ISWC 2025 Companion Volume, November 2–6, 2025, Nara, Japan

*Corresponding author.

wout.slabbinck@ugent.be (W. Slabbinck); lennert.derouck@ugent.be (L. D. Rouck); joachim.vanherwegen@ugent.be (J. V. Herwegen); wouter.termont@ugent.be (W. Termont); beatriz.esteves@ugent.be (B. Esteves); ruben.verborgh@ugent.be (R. Verborgh)

https://woutslabbinck.com/ (W. Slabbinck); https://w3id.org/people/besteves (B. Esteves); https://ruben.verborgh.org/ (R. Verborgh)

0000-0002-3287-7312 (W. Slabbinck); 0000-0002-2968-1394 (W. Termont); 0000-0003-0259-7560 (B. Esteves); 0000-0002-8596-222X (R. Verborgh)

in a plethora of non-interoperable systems and proposals, between which development and interaction are a costly affair.

This tension is particularly visible in recent attempts at streamlining private data exchange on the Web, including *i)* several parallel efforts shaping Data Spaces (e.g., by IDSA, DSSC, Gaia-X, and Eclipse [1, 2, 3, 4]), which struggle to sufficiently align their conceptual visions into concrete interoperable technical specifications[1]; *ii)* numerous new regulations that impose data processing and governance requirements without much of a technical stack ready to facilitate them (e.g., in European Union law [6, 7, 8, 9]); and *iii)* multiple independent technical frameworks for (personal) data exchange, including the Solid and Fedora projects [10, 11][2], which fail to provide a foundation for many of the use cases formulated by the other endeavours.

In [18], the authors highlight the lack of orthogonality between the data plane and the control plane in existing technical solutions[3], and their hierarchical, document-centric view of data, since it forms an inflexible dependency between internal and external interfaces for data and access management, and thus precludes the organic formation of an interoperable and scalable ecosystem in which Digital Trust flows from a variety of mutually beneficial relationships.

In this paper, we introduce LOAMA, a user interface designed to manage policies in an UMA Authorization Server (AS). LOAMA simplifies the complexity of policy management, offering an intuitive experience for users, while the underlying API supports the exchange of complete and detailed policies. To demonstrate the functionality of our implementation, we also provide a screencast.

2. Related Work

These issues are partially addressed by the Solid Application Interoperability (SAI) specification [20, 21], [4] which proposes to combine a registration-based policy model with more fine-grained, declarative resource description languages (e.g., SHACL [22], SHEX [23]), and User-Managed Access (UMA) [24, 25] – an OAuth 2.0 extension enabling asynchronous and delegated multi-user access control, dynamic grant negotiation, and federation over multiple protection domains.

Still, SAI's policy language lacks any legal or corporate semantics and expressivity, forming only a technical interoperability layer *on top of* WAC or ACP. Several authors highlight this shortcoming (with demo implementations in [26, 27]), stating that since "[WAC and ACP] cannot represent more complex rules nor invoke regulation-specific concepts," [28], Solid still lacks "the necessary vocabulary and processes for ensuring transparency and accountability [...] to deal with the obligations and requirements required by [them]," [29], as well as "the granularity and contextual awareness needed to enforce these regulatory requirements" [27]. Each of these papers emphasizes the importance of usage control over mere access control;[5] and

[1] E.g., Eclipse's Dataspace Protocol [5] merely mentions that "requests [...] SHOULD use the Authorization header to include an authorization token [the semantics of which] are not part of this specification."

[2] Their core texts [12, 13], based on the Linked Data Platform (LDP), Web Access Control (WAC), and Access Control Policy (ACP) specifications [14, 15, 16], now inform W3C's Linked Web Storage (LWS) Working Group [17].

[3] This separation of concerns between resource servers (data management) and authorization servers (access management) is ubiquitous in modern access control mechanisms, such as OAuth 2.0 [19].

[4] Surprisingly, the SAI specification has not been taken up as input by the W3C LWS WG (cf. 2).

suggests an integration with the Open Digital Rights Language (ODRL) and the Data Privacy Vocabulary (DPV) [30, 31, 32].

A key piece often overlooked in these architectural designs is how policies should be concretely managed by the resource owner [33]. As [21] succinctly put: "Solid's use of [WAC and ACP] is tedious and prone to errors"; "little attention has been given to how users [...] would actually exert their control." Although the paper discusses a prototype implementation of a user interface based on SAI and DPV, the authors identify several limitations and open issues. Other similar applications [34, 35] restrict themselves to SAI and WAC or ACP, ignoring the important insights from previous work around ODRL and DPV.

The whitepaper *From Resource Control to Digital Trust with User-Managed Access* [18] builds on this earlier work around Solid and ODRL, and identifies several requirements that a technical solution for access and usage control should fulfill in order to form a strong connection to legal and corporate conceptual frameworks. The authors suggest a number of modifications to UMA, which they integrate in the UMA-extension Authorization for Data Spaces (A4DS) [36]. In line with the attempts of [21, 35], this demo paper will discuss and showcase the foundations of a third authorization application, which communicates through ODRL messages with an authorization server following the A4DS UMA extension.

3. Access Management Application

3.1. Architecture

The design of LOAMA builds upon the architecture of UMA, which defines five key roles [24]. Before elaborating on our design and implementation choices, we briefly introduce these five roles: *i)* the **Resource Owner** (RO), who manages access policies for protected resources; *ii)* the **Requesting Party** (RP), who seeks access to a protected resource; *iii)* the **Client**, which acts on behalf of the RP and interacts with both the Resource Server and Authorization Server while adhering to the UMA flow; *iv)* the **Resource Server** (RS), which hosts the protected resources on behalf of the RO; *v)* the **Authorization Server** (AS), which enforces access control policies and issues tokens to the Client on behalf of the RO. UMA does not specify how an RO should configure policies at the AS [6], but leaves this up to the implementer. Illustrated in Figure 1, our architecture therefore incorporates a sixth role, the **Access Management Application** (AMA), which acts as an intermediary between the RO and the AS. To enable this interaction, the AS must expose a dedicated interface through which the AMA can communicate policy updates on behalf of the RO.

Authorization Server

Since UMA leaves policy management open to the implementers, the first step for making policy management possible is devising an API endpoint. A quick study of well-known access

[5]While access control is merely concerned with which parties can access what resources, usage control includes the conditions and obligations associated with this authorization.

[6]The UMA 2.0 Grant specification [24] explicitly states that the AS-RO interface is out of scope in §6.1. Similar remarks are made in the Federated Authorization for UMA specification [25], in §1.2 and §8.

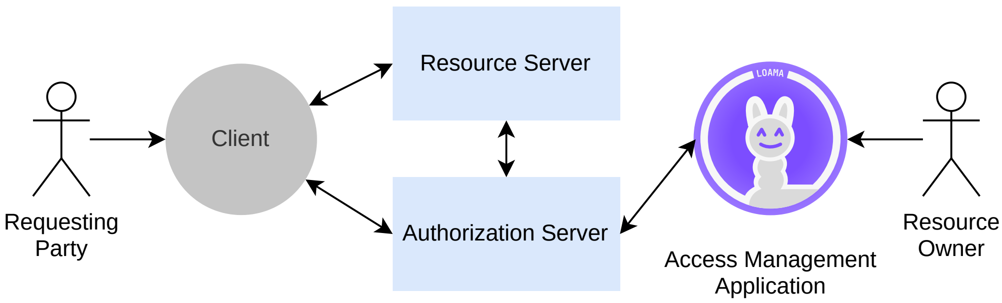

Figure 1: Overview of the UMA architecture extended with a sixth role, the Access Management Application (AMA), which facilitates policy management on behalf of the Resource Owner.

control solutions' endpoints, including the Policy Administration Point (PAP) of XACML [37] and the Open Policy Agent API [38], showcases that best practices use REST APIs [39]. As such, we developed a REST API for Creating, Reading, Updating and Deleting (CRUD) ODRL policies, that builds upon the UMA Authorization Server introduced by [40], which enforces usage control using ODRL policies [41].

In addition to the API design, there are some further design considerations implemented to ensure the correct handling of incoming requests, which are grouped into two main categories: *i)* syntactic payload validation and *ii)* security mechanisms. Syntactic validation targets the structural correctness of policies modeled using the ODRL Information Model [30], which is defined as an RDF ontology [31]. The first step involves verifying that the request payload is of an RDF-compatible content type and can be successfully parsed. Subsequently, the parsed payload is checked for conformance with the ODRL Information Model. The security mechanisms employed ensure that no unauthorized policy management can be executed. First, the presence of an Authorization header in incoming requests is verified. The identifier of the RO is then derived from the information in this header. Requests to read or modify policies are permitted only if the RO's identifier matches the `odrl:assigner` property specified in the corresponding ODRL rule. More information about the REST API endpoint and implementation details can be found in the documentation of the repository[7].

User Interface

The LOAMA User Interface (UI) abstracts away the complexity of manual ODRL policy creation and executes API calls to the aforementioned UMA AS. It consists of three main components: an **Authentication page**, an **Overview page**, and a **Policy Editor page**.

The landing page of the UI is the Authentication page, where ROs authenticate using Solid-OIDC [42]. Upon successful authentication, the UI obtains the RO's credentials, enabling it to issue authorized requests. After authentication, the user is redirected to the Overview page (Figure 2), which displays all resources under the control of the authenticated RO. Selecting

[7]UMA Policy Management documentation and implementation details: https://github.com/SolidLabResearch/user-managed-access/blob/feat/policy-endpoint/docs/policy-management.md

a resource brings the user to the Policy Editor page. In the Policy Editor, the RO can define and modify access control rules for each resource. Specifically, the RO may assign one or more `odrl:assignees` and specify the corresponding permitted `odrl:actions`. Any changes made through the interface are automatically propagated to the Authorization Server via the appropriate API calls. The source code for the UI is publicly available in its GitHub repository[8].

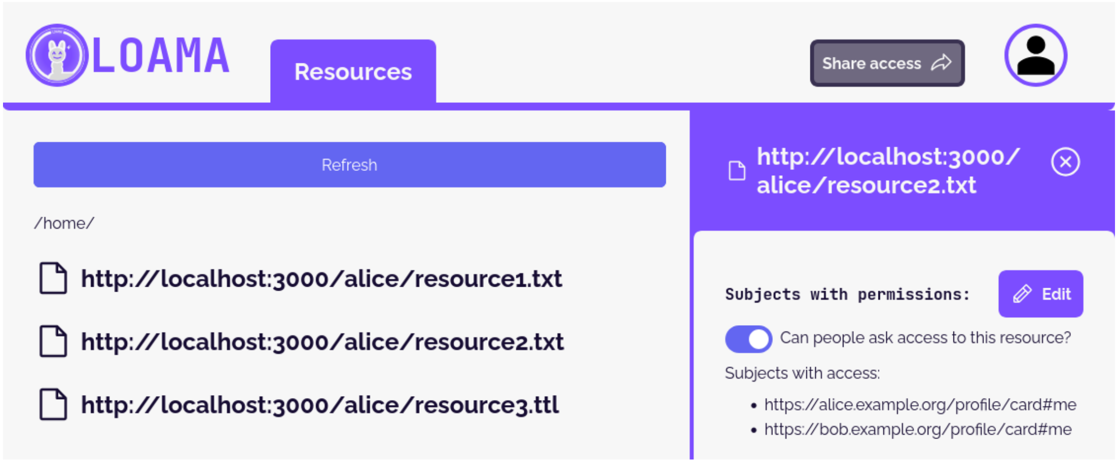

Figure 2: Screenshot of the LOAMA Overview page. On the left, all resources controlled by Alice are shown. On the right, the selected resource is displayed, including all subjects with access. Clicking the *Edit* button opens the Policy Editor page.

3.2. Demonstration

We illustrate the use of the policy management API and LOAMA UI through a scenario in which Alice, the Resource Owner, grants access to Bob, the Requesting Party. Initially unauthorized, Bob is unable to read or update the resource. After Alice updates the policy using the LOAMA interface, Bob successfully performs the intended actions. A complete video demonstration of this scenario, including further details, is available on Zenodo[9].

4. Discussion

The choice for a RESTful API for ODRL policy management for an UMA AS empowers ROs with fine-grained, low-level control over how usage of their data is enforced. At the same time, the LOAMA interface demonstrates how this complexity can be abstracted away, enabling ROs to understand and manage ODRL policies without requiring detailed knowledge of either ODRL or the API interactions. The implementation of the policy management API led to insights on open issues. First, there is a security consideration, namely the need for a mechanism to prevent unauthorized applications from altering policies. For instance, when a Resource Owner

[8]LOAMA with UMA: https://github.com/SolidLabResearch/loama/tree/feat/odrl
[9]LOAMA demonstration with the UMA AS policy management API: https://zenodo.org/records/16640205

authenticates through a generic application to access data, the resulting authentication token could be exploited by malicious code to modify policies. To address this risk, it is advisable to restrict policy changes to certified applications only. This concern has also been noted in research on decentralized data sharing solutions [35]. Another open issue is the need for semantic validation in the AS to prevent contradicting ODRL rules, such as a permission and a prohibition for the same asset, action, and party. If such validation cannot be enforced, an additional safeguard is required: effective conflict resolution strategies must be in place to support consistent decision-making within the UMA AS policy engine. Finally, there is a need for a mechanism to request specific permissions from a RO, which is currently missing in the implementation. However, the existing AS can serve as a foundation for further research; perhaps it can be used to compare existing decentralized negotiation strategies, including the Dataspace Negotiation Protocol[10] [43], SAI [20], or Authapp [35].

5. Conclusion

In this paper, we introduce an ODRL policy management API for a UMA Authorization Server along with LOAMA, a web-based user interface that enables Resource Owners to manage their policies more easily. Future research includes conflict resolution in policy management and decentralized access negotiation mechanisms.

Acknowledgments

This research was funded by SolidLab Vlaanderen (Flemish Government, EWI and RRF project VV023/10), and the European Union’s Horizon Europe research and innovation program under grant agreement no. 101058682 (Onto-DESIDE).

Declaration on Generative AI

During the preparation of this work, the author(s) used Grammarly in order to: Grammar and spelling check. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication’s content.

References

[1] International Data Spaces Association, International Data Spaces, https://internationaldataspaces.org, 2025. Accessed: 2025-07-31.
[2] Data Spaces Support Centre, https://dssc.eu, 2025. Accessed: 2025-07-31.
[3] Gaia-X European Association for Data and Cloud, Gaia-X, https://gaia-x.eu, 2025. Accessed: 2025-07-31.

[10] Data Space Negotiation Protocol: https://github.com/International-Data-Spaces-Association/ids-specification/blob/main/negotiation/contract.negotiation.protocol.md

[4] Eclipse Foundation, Eclipse Dataspace Working Group, https://dataspace.eclipse.org, 2025. Accessed: 2025-07-31.

[5] P. Koen, M. Kollenstart, J. Marino, J. Pampus, A. Turkmayali, S. Steinbuss, A. Weiß, Dataspaces Protocol 2025-1-RC4, Technical Report, Eclipse Foundation, 2025.

[6] Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation), 2016. URL: ttp://data.europa.eu/eli/reg/2016/679/oj.

[7] Regulation (EU) 2022/868 of the European Parliament and of the Council of 30 May 2022 on European data governance and amending Regulation (EU) 2018/1724 (Data Governance Act), 2022. URL: http://data.europa.eu/eli/reg/2022/868/oj/eng.

[8] Regulation (EU) 2022/2065 of the European Parliament and of the Council of 19 October 2022 on a Single Market For Digital Services and amending Directive 2000/31/EC (Digital Services Act), 2022. URL: http://data.europa.eu/eli/reg/2022/2065/oj/eng.

[9] Regulation (EU) 2023/2854 of the European Parliament and of the Council of 13 December 2023 on harmonised rules on fair access to and use of data and amending Regulation (EU) 2017/2394 and Directive (EU) 2020/1828 (Data Act), 2023. URL: http://data.europa.eu/eli/reg/2023/2854/oj.

[10] Open Data Initiative, Solid Project, https://solidproject.org/, 2025. Accessed: 2025-07-31.

[11] Fedora Governance Group, Fedora Repository, https://fedorarepository.org/, 2025. Accessed: 2025-07-31.

[12] W3C Solid Community Group, Solid Protocol, Editor's Draft, W3C, 2025.

[13] Fedora Governance Group, Fedora API, Candidate Recommendation, Fedora, 2018.

[14] W3C Linked Data Platform Working Group, Linked Data Platform 1.0, Recommendation, W3C, 2015.

[15] W3C Solid Community Group, Web Access Control, Draft Community Group Report, W3C, 2025.

[16] W3C Solid Community Group, Access Control Policy, Editor's Draft, W3C, 2022.

[17] A. Coburn, L. Debackere, E. Prud'hommeaux, Linked Web Storage Working Group Charter, Working Group Charter, W3C, 2024. URL: https://www.w3.org/2024/09/linked-web-storage-wg-charter.html.

[18] W. Termont, R. Dedecker, W. Slabbinck, B. Esteves, B. De Meester, R. Verborgh, From Resource Control to Digital Trust with User-Managed Access, Technical Report, SolidLab (IDLab, Ghent University – imec), 2024.

[19] D. Hardt, The OAuth 2.0 Authorization Framework, Technical Report 6749, IETF, 2012. URL: https://www.rfc-editor.org/info/rfc6749. doi:10.17487/RFC6749.

[20] W3C Solid Community Group, Solid Application Interoperability, Draft Community Group Report, W3C, 2025.

[21] H. Bailly, A. Papanna, R. Brennan, Prototyping an End-User User Interface for the Solid Application Interoperability Specification Under GDPR, in: C. Pesquita, E. Jimenez-Ruiz, J. McCusker, D. Faria, M. Dragoni, A. Dimou, R. Troncy, S. Hertling (Eds.), The Semantic Web: 20th International Conference, ESWC 2023, May 28–June 1, Proceedings, volume 13870 of *Lecture Notes in Computer Science*, Springer, Cham, Switzerland, 2023, pp. 557–573. doi:10.1007/978-3-031-33455-9_33.

[22] W3C RDF Data Shapes Working Group, Shapes Constraint Language (SHACL), Recommendation, W3C, 2017. URL: https://www.w3.org/TR/shacl/.

[23] W3C Shape Expressions Community Group, Shape Expressions Language 2.1, Final Community Group Report, W3C, 2019. URL: https://shex.io/shex-semantics/.

[24] M. Machulak, J. Richer, User-Managed Access (UMA) 2.0 Grant for OAuth 2.0 Authorization, Recommendation, Kantara Initiative, 2018. URL: https://docs.kantarainitiative.org/uma/wg/rec-oauth-uma-grant-2.0.html.

[25] M. Machulak, J. Richer, Federated Authorization for User-Managed Access (UMA) 2.0, Recommendation, Kantara Initiative, 2018. URL: https://docs.kantarainitiative.org/uma/wg/rec-oauth-uma-federated-authz-2.0.html.

[26] W. Slabbinck, J. A. Rojas, B. Esteves, R. Verborgh, P. Colpaert, Enforcing Usage Control Policies in Solid using Rule-Based Web Agents, in: Proceedings of the Posters and Privacy Session of the Solid Symposium 2024, 2024, pp. 109–117. URL: https://ceur-ws.org/Vol-3947/short15.pdf.

[27] L. Debackere, P. Colpaert, R. Taelman, R. Verborgh, A Policy-Oriented Architecture for Enforcing Consent in Solid, in: F. Laforest, R. Troncy, L. Médini, I. Herman (Eds.), Companion Proceedings of the Web Conference 2022 (WWW '22), Association for Computing Machinery, New York, United States, 2022, pp. 516–524. doi:10.1145/3487553.3524630.

[28] B. Esteves, H. J. Pandit, V. Rodríguez-Doncel, ODRL Profile for Expressing Consent through Granular Access Control Policies in Solid, in: R. Mutharaju, A. Ławrynowicz, P. Bhattacharyya, E. Blomqvist, L. Asprino, G. Singh (Eds.), 2021 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), IEEE, Vienna, Austria, 2021, pp. 298–306. doi:10.1109/EuroSPW54576.2021.00038.

[29] B. Esteves, H. J. Pandit, Using Patterns to Manage Governance of Solid Apps, in: R. Mutharaju, A. Ławrynowicz, P. Bhattacharyya, E. Blomqvist, L. Asprino, G. Singh (Eds.), Proceedings of the 14th Workshop on Ontology Design and Patterns (WOP 2023) co-located with the 22nd International Semantic Web Conference (ISWC 2023), volume 3636 of *CEUR Workshop Proceedings*, 2023, pp. 43–55.

[30] W3C Permissions & Obligations Expression Working Group, Open Digital Rights Language (ODRL) Information Model 2.2, Recommendation, W3C, 2018.

[31] W3C Permissions & Obligations Expression Working Group, Open Digital Rights Language (ODRL) Vocabulary & Expression 2.2, Recommendation, W3C, 2018.

[32] H. J. Pandit, B. Esteves, G. P. Krog, P. Ryan, D. Golpayegani, J. Flake, Data Privacy Vocabulary (DPV) – Version 2.0, in: G. Demartini, K. Hose, M. Acosta, M. Palmonari, G. Cheng, H. Skaf-Molli, N. Ferranti, D. Hernández, A. Hogan (Eds.), The Semantic Web – ISWC 2024, Springer Nature Switzerland, Cham, 2024, pp. 171–193. doi:10.1007/978-3-031-77847-6_10.

[33] W. Slabbinck, The need for Usage Control in Decentralized and Federated Ecosystems, in: K. Taylor, A. Zimmermann (Eds.), Proceedings of the Doctoral Consortium at ISWC 2024, volume 3884 of *CEUR Workshop Proceedings*, CEUR, Baltimore, USA, 2024. URL: https://ceur-ws.org/Vol-3884/paper2, iSSN: 1613-0073.

[34] ActivityPods, https://activitypods.org/, 2025. Accessed: 2025-07-31.

[35] A. Both, T. Kastner, D. Yeboah, C. Braun, D. Schraudner, S. Schmid, T. Käfer, H. Andreas, AuthApp – Portable, Reusable Solid App for GDPR-compliant Access Granting,

in: K. Stefanidis, K. Systä, M. Matera, S. Heil, H. Kondylakis, E. Quintarelli (Eds.), Web Engineering: 24th International Conference (ICWE 2024) Proceedings, Lecture Notes in Computer Science, Springer, Cham, Switzerland, 2024, pp. 199–214. doi:10.1007/978-3-031-62362-2_14.

[36] W. Termont, Authorization for Data Spaces, Technical Report, KNoWS (IDLab, Ghent University – imec), 2025.

[37] B. Parducci, H. Lockhart, E. Rissanen, eXtensible Access Control Markup Language (XACML) Version 3.0 – OASIS Standard, 2013. URL: http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-en.html.

[38] Open Policy Agent - Policy API, 2025. URL: https://www.openpolicyagent.org/docs/rest-api#policy-api.

[39] R. T. Fielding, R. N. Taylor, Principled design of the modern Web architecture, ACM Trans. Internet Technol. 2 (2002) 115–150. URL: https://dl.acm.org/doi/10.1145/514183.514185. doi:10.1145/514183.514185.

[40] W. Slabbinck, R. Dedecker, W. Termont, B. Esteves, P. Colpaert, R. Verborgh, From Access Control to Usage Control with User-Managed Access, Solid Symposium 2025, Leiden, The Netherlands, 2025. Forthcoming.

[41] W. Slabbinck, J. Rojas Meléndez, B. Esteves, P. Colpaert, R. Verborgh, Interoperable Interpretation and Evaluation of ODRL Policies, in: E. Curry, M. Acosta, M. Poveda-Villalón, M. van Erp, A. Ojo, K. Hose, C. Shimizu, P. Lisena (Eds.), The Semantic Web, Springer Nature Switzerland, Cham, 2025, pp. 192–209. doi:10.1007/978-3-031-94578-6_11.

[42] W3C Solid Community Group, Solid-OIDC, Draft Community Group Report, W3C, 2022.

[43] S. Yumusak, S. Gheisari, J. O. Salas, L.-D. Ibáñez, G. Konstantinidis, Data Sharing Negotiation and Contracting (2024). URL: https://ceur-ws.org/Vol-3828/paper39.pdf.