Enforcing Usage Control Policies in Solid Using A Rule-Based Software Agent

Wout Slabbinck[1,*], Julián Andrés Rojas[1] and Ruben Verborgh[1]

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

Abstract

A core concept of the Solid ecosystem is the sharing of resources with other agents using access control policies. However, you might not want to share data indefinitely. The Solid authorization specifications do not allow expressing and thus neither enforcing temporal usage policies. A policy language that does have the expressivity to declare permission rules with fine-grained conditions is the *open digital rights language* (ODRL) standard. To enforce ODRL policies over Solid resources, we configured an open-source demonstrator Web agent that (i) decomposes an ODRL policy to actionable tasks (such as giving and retracting to resources) using condition-action rules and (ii) executes these tasks. Usage control within Solid can be delegated to agents such that neither applications nor users within the Solid ecosystem need to take care of ensuring that the permissions over their resources are up to date. We show that the limitations of the current specifications can be overcome by adopting another policy standard, ODRL, and a long-running agent with the task of enforcement. As currently, all data-sharing actions within the Solid ecosystem are executed by the owner of the resource, future work might include negotiation with Web agents over the policies of other actors.

Keywords

Solid, Access Control, Usage Control, Enforcement, Policy, Intelligent software web agents

1. Introduction

The Solid protocol[1] is a Personal Data Store Web technology that enables individuals to store and govern data on their data space, in Solid terminology also referred to as *pod*. A Solid server must support at least one of the two authorization specifications, namely the Web Access Control (WAC)[2] specification and/or the Access Control Policy (ACP) Language[3], such that a user can declare access control rules over resources in their pod. To this date, however, these specifications do not support fine-grained access rules with temporal conditions. As a result, users must either for revocation re-configure the access control resources again after access was granted, or use a Solid application that can perform the granting and revoking access. However, neither option is infallible. In the first option, users must adjust rules to revoke access, yet human forgetfulness may allow unintended authorizations to persist. For the second option, due to the Web-based nature of the Solid protocol, the application must remain open and active

2nd Solid Symposium, 2-3 May 2024, Leuven

[*] Corresponding author.

wout.slabbinck@ugent.be (W. Slabbinck); julianandres.rojasmelendez@ugent.be (J. A. Rojas); ruben.verborgh@ugent.be (R. Verborgh)

[1] https://solidproject.org/TR/protocol

[2] https://solidproject.org/TR/wac

[3] https://solidproject.org/TR/acp

for revocation to occur. An approach to overcome the limitations of current specifications is to use another standard which supports fine-grained conditions to declare and enforce policies such as the open digital rights language (ODRL)[4] W3C standard. Since Solid only supports the aforementioned specifications, ODRL can not directly be enforced. However, it is possible to materialize ODRL to existing access rules that the Solid protocol can enforce.

In this paper, we extend an open-source, rule-based Web agent [1] to (i) perceive policies from an ODRL policy Knowledge Graph (KG), managed by the resource owner, (ii) interpret those ODRL policies and transform them to Access Control List (ACL) rules as defined by WAC, and (iii) perform long-running tasks to enforce temporal usage control policies.

2. Demo

To demonstrate the enforcement of a temporal ODRL policy, we elaborate the use case of a Solid pod owner, Alice, who wants to share resource X for a limited amount of time with another actor Bob. It is assumed she has a policy Knowledge Graph (KG) to which her Web Agent is subscribed and that the Web Agent has `acl:Control` permission to the resources in Alice her pod.

To start sharing resource X, Alice adds the following data usage ODRL policy to the KG: **Alice** gives **read** access to **Bob** for the duration of 30 seconds. Next, the Agent is notified that the policy has been added to the Policy KG and the Agent fetches this policy. The duration policy is passed to a reasoner, which results in an action plan with two concrete actions to be performed: (i) The authorization resource of X must be updated such that Bob has read access and (ii) In 30 seconds, the authorization resource of X must be updated such that Bob has no longer access. The Agent performs (i) and starts a CronJob for 30 seconds to remove Bob's access to X. Finally, after 30 seconds have passed, the Agent sends another request to the authorization resource of X such that Bob has no more access.

A screencast in the open-source Solid Agent GitHub repository demonstrates this enforcement flow[5]. This repository also contains a duration ODRL policy, which is used in this example.

3. Conclusion

In this paper, we demonstrate how temporal usage control policies can be enforced in the Solid ecosystem by employing ODRL and a Web Agent. Future work may include negotiation over usage control policies with Web Agents such that other actors can initiate asking permission.

References

[1] W. Slabbinck, R. Dedecker, J. A. Rojas Meléndez, R. Verborgh, A rule-based software agent on top of personal data stores, in: Proceedings of the 22nd International Semantic Web Conference: Posters, Demos, and Industry Tracks, 2023.

[4] https://www.w3.org/TR/odrl-model/

[5] https://github.com/SolidLabResearch/Solid-Agent/tree/feat/sosy/documentation/sosy