Towards authorization for purpose-based Agent search in Solid

Ruben Dedecker

Ghent University - imec.

Abstract

AI agents are increasingly used to answer questions over heterogeneous data spaces. Solid provides a Web-native, resource-centric architecture with server-enforced authentication and authorization, making it a natural foundation for agent-accessible personal data storage. However, effective question answering over Solid pods requires performant semantic search, yet current Solid discovery mechanisms are insufficient for the retrieval demands of LLM-based agents. We discuss three indexing and retrieval approaches, in client-side link traversal, resource-specific index artifacts, and Policy-Scoped Indexing, in their respective capabilities for enabling semantic search over Solid pod environments. We argue that policy-scoped indexing on dimensions such as purpose-based authorization, grounded in ODRL and the DPV, enables powerful cross-resource embeddings that serve both the pod owner and authorized third-party agents through verifiable, purpose-scoped semantic search.

1. Introduction

AI agents are rapidly becoming a primary interface to the Web, capable of answering questions over heterogeneous, distributed data sources. Solid [1] provides a Web-native storage architecture in which resources are accessed over HTTP and protected by server-enforced authorization via Web Access Control (WAC) [2] or Access Control Policies (ACP) [3]. It allows agents to retrieve heterogeneous user data from user-controlled storage with scoped, per-resource access grants, rather than having to rely on the capabilities of centralized data silos.

However, current AI agent systems assume fast semantic search to extract relevant information from data spaces. This demands indexing capabilities beyond what the current Solid mechanisms provide. Existing approaches in the Solid ecosystem include Type Indexes as lightweight public and private indexes that link RDF types to resources on the pod [4], the use of a Quad Pattern Fragments (QPF) [1] endpoint that exposes access to the pod's internal RDF graph, but is restricted to users with the control access, reflecting the difficulty of authorizing cross-resource queries to third parties. Finally, the Solid Application Interoperability specification takes a complementary approach, organizing resources by entity shape and using this as the basis for access negotiation.

None of these mechanisms meets the semantic retrieval requirements of RAG-based agents, motivating a dedicated treatment of indexing strategies and their authorization implications.

2. Retrieval And Indexing Approaches

Client-side link traversal makes use of Link Traversal Query Processing (LTQP) [5] to discover and retrieve resources from a Solid pod by following RDF links from seed URIs, following the native Solid authorization flow. This provides the strongest authorization guarantees and is compatible with any Solid implementation. However, LTQP retrieves based on structural reachability, not semantic relevance, making it poorly suited for RAG. This reliance on link structure may cause agents to miss rele-

vant resources that are not directly linked from the seed URIs within the user authorization space. Additionally, the ability to stream intermediate results [5] does not improve system response times unless context requests can be meaningfully chained by the AI agent.

Resource-specific indexing artifacts go a step beyond link traversal, allowing clients to navigate per-resource maintained indexing artifacts, taking the form of vector embeddings, graph embeddings, or even simple type indexes. External clients can traverse their authorized search space for relevant indexing artifacts, based on which the identified top-k matching resources for the user query can be retrieved by the client. The server may additionally materialize aggregated indexes at the container or pod level to speed up retrieval. Binding index access to the resource's authorization boundary ensures correctness, but limits the discoverability of cross-resource links, missing possibly relevant resources the client could request access to.

Policy-Scoped Indexing moves the query evaluation fully to the Solid server, allowing indexing of resources to approach to generic authorization dimensions, such as identity, role, or purpose. Here, a complete semantic index can be maintained over the pod contents, that supports both vector similarity search and graph traversal. Crucially, embeddings can be computed cross-resource within an authorization dimension, enabling full semantic search over the entire indexed space rather than individual resources in isolation. Because the indexing dimension and per-resource access rights do not necessarily coincide, protocols that support access negotiation such as User-Managed Access (UMA) [6] allow clients to negotiate access for relevant resources.

3. Purpose-Based Authorization

The existing Solid WAC and ACP authorization systems are restricted to identity, application and mode-based authorization. They do not provide a mechanism for cross-resource policies or for conditioning access on the reason data is being processed, that is needed to allow more broad semantic search for agents. Here, additional dimensions such as Purpose-based authorization provide the ability to govern semantic search over cross-resource embeddings.

Here, the Data Privacy Vocabulary (DPV) [7] provides a basis for expressing these requirements, covering dimensions such as dpv:Purpose and dpv:LegalBasis. ODRL [8] complements DPV as a policy expression language for encoding verifiable access conditions over Solid resources.

This enables two distinct access patterns. For the pod owner or broadly authorized agents, purpose-based indexing supports deep semantic search across the full pod space via cross-resource embeddings. For third-party agents, verifiable purpose declarations act as policy-scoped shortcuts into the semantic index, without exposing underlying resources directly.

References

[1] Sambra, A.V., Mansour, E., Hawke, S., Zereba, M., Greco, N., Ghanem, A., Zagidulin, D., Aboulnaga, A., Berners-Lee, T.: Solid: a platform for decentralized social applications based on linked data (2016).
[2] Berners-Lee, T., Story, H., Capadisli, S.: Web Access Control. W3C Solid Community Group, https://solidproject.org/TR/wac (2024).
[3] Bosquet, M. ed: Access Control Policy (ACP). https://solid.github.io/authorization-panel/acp-specification/ (2022).
[4] Turdean, T. ed: Type Indexes. https://solid.github.io/type-indexes/ (2023).
[5] Taelman, R., Verborgh, R.: Link Traversal Query Processing over Decentralized Environments with Structural Assumptions. In: Proceedings of the 22nd International Semantic Web Conference (2023).

[6] Machulak, M., HSBC, Richer, J., Engineering, B.: User-Managed Access (UMA) 2.0 Grant for OAuth 2.0 Authorization. Kantara Initiative, https://docs.kantarainitiative.org/uma/wg/rec-oauth-uma-grant-2.0.html (2019).
[7] Pandit, H.J. ed: Data Privacy Vocabulary (DPV). W3C Data Privacy Vocabularies and Controls Community Group (DPVCG), https://w3c.github.io/dpv/2.2/dpv/ (2025).
[8] Iannella, R., Villata, S. eds: ODRL Information Model 2.2. World Wide Web Consortium (W3C), https://www.w3.org/TR/odrl-model/ (2018).

1. The Enterprise Solid Server exposes a Quad Pattern Fragments query interface for each pod, queryable via https://fragments.inrupt.com/ ↩