Solid Cockpit: Data manager, privacy editor, and query facilitator for Solid pods

Elias Crum[1,2][0009−0005−3991−754X], Bart Buelens[1,2][0000−0001−7734−3747], Gokhan Ertaylan[2][0000−0001−5602−6435], and Ruben Taelman[1][0000−0001−5118−256X]

[1] IDLab, Department of Electronics and Information Systems, Ghent University – imec, Belgium
[2] Flemish institute for Technological Research (VITO), Mol, Belgium
elias.crum@ugent.be

Abstract. Solid, a decentralized Web-based personal data storage protocol, enables users to store, manage, and share data in a flexible and user-controlled manner, especially for sensitive data. Despite this promise, practical interaction with Solid Pods remains difficult because existing applications only cover fragments of the overall workflow, while common tasks such as permission management and cross-resource querying still require substantial technical knowledge. We present *Solid Cockpit*, a browser-based interface for managing and querying Solid Pods with a particular focus on privacy-transparent sharing workflows. Beyond basic Pod data management, Solid Cockpit contributes two draft technical specifications for currently under-supported tasks in the ecosystem: auditable permission-change notifications and Pod-native storage of query results. The application combines (i) data management, (ii) a visual privacy editor with integrated permission-change notifications, and (iii) an embedded SPARQL querying component with example queries, query history, and result materialization. We position the system against related Solid user interfaces and architectures and illustrate how notification support fits into a high-level sharing workflow. The demonstrator is deployed online at `https://knowledgeonwebscale.github.io/solid-cockpit/`.

Keywords: Solid · Privacy Controls · Notifications · SPARQL Querying · Data Management

1 Introduction

Solid Pods [10,5] provide a flexible foundation for decentralized data storage, fine-grained access control, and interoperable data exchange on the Web. By separating data from applications, Solid enables individuals to retain sovereignty over their personal data while allowing third-party applications to access it under explicitly defined permissions. Despite this strong conceptual foundation, practical interaction with Pod data remains challenging and does not yet fully leverage the capabilities of Solid.

Table 1. Positioning of Solid Cockpit with respect to related Solid systems.

System	Privacy UI	Notification support	Query support
solid-auth-ui [3]	Yes	No	No
solid-web-push [4]	No	Browser Push	No
Comunica [11]	No	No	Yes
Solid Gateway [8]	Consent Enforcement	No	Partial
Pod Pro[3]	No	No	No
Penny[4]	.acl editing	No	No
Solid Cockpit	**Yes**	**In-App**	**Yes**

Existing Solid applications focus on file-system-like navigation or basic Pod dashboards, such as Pod Pro[3] or Penny[4]. While these tools provide essential storage and navigation functionality, they offer limited support for integrated privacy management, auditable sharing workflows, or semantic querying. Other systems address narrower aspects of the ecosystem, such as authorization interfaces [3], notification delivery [4], or query processing over decentralized Solid environments [11], but do not provide a single user-facing environment for holistic Pod interaction.

Table 1 situates Solid Cockpit with respect to closely related Solid interfaces. solid-auth-ui [3] focuses on authorization requests and usage control, with an explicit user study. solid-web-push [4] demonstrates Web Push notifications from Solid Pods, but their emphasis is on push delivery for progressive web applications rather than auditable permission-change workflows. The Comunica Link-Traversal demonstrator [11] focuses on query execution over decentralized Solid environments rather than end-user privacy management or a general pod management interface. Solid Gateway [8] presents consent-driven access from privacy-preserving analysis environments to Solid Pods as well as a user-facing Request Portal, but is mostly focused on back-end consent mediation and analytics.

Solid Cockpit is complementary to the efforts represented in Table 1 and aims to fill three key Solid usage gaps. First, defining granular data-sharing policies and understanding when permissions have changed requires a level of technical expertise that exceeds that of many end users. Second, even advanced users have limited support for tracking who shared resources with them or how permissions evolved over time. Third, while SPARQL querying can be a powerful abstraction over RDF data, most Solid interfaces do not support such functionalities.

To address these gaps, we present *Solid Cockpit*, a web-based application that integrates Solid Pod data management, visual privacy controls, permission-change notifications, and semantic querying. Our contribution is not a new Solid

[3] `https://podpro.dev/`

[4] `https://penny.vincenttunru.com/`

architecture, but a user-facing cockpit that operationalizes two currently under-supported workflows in the Solid ecosystem: *auditable permission-change management* and *Pod-native storage of query results*. In particular, this paper contributes:

1. a browser-based cockpit for common Solid Pod operations;
2. a visual privacy editor with contextual, auditable permission-change notifications based on our Linked Data privacy notifications technical specification (LDPN) [6];
3. support for query history and materialized query results based on our SPARQL view materialization containers technical specification (QVMC) [7].

2 System Design

Solid Cockpit is implemented as a client-side web application. All interactions with user data occur directly between the browser and the user's Solid Pod through standardized Solid protocols [5], avoiding intermediary application servers for authentication, storage, or querying. The implementation relies on the Inrupt Solid JavaScript SDK [2] for authentication, resource access, and access-control updates, and on Comunica [1] for in-browser query execution. Open-source code is available at `https://github.com/KNowledgeOnWebScale/solid-cockpit`.

2.1 Core Interaction Layers

Authentication and data management. Users authenticate with their Solid identity provider and then interact with Pod resources through common CRUD (Create, Read, Update, Delete) workflows. The data-management interface intentionally resembles familiar file-management interactions such as browsing resource trees and drag-and-drop uploads, so that non-expert users can perform common Pod tasks without manipulating RDF directly.

Privacy editing and notifications. Solid Cockpit provides a visual policy editor that exposes permissions in user-facing terms rather than raw RDF triples. Users can specify which agents may read, write, append, or control a resource via Web Access Control and Access Control List updates through the Solid client SDK. At a high level, the intended workflow is that a user selects or uploads a resource, changes its sharing settings, and later sees the resulting grant or revocation as a contextual notification. To make these changes more transparent, the application includes a notification subsystem following our draft Linked Data privacy notifications (LDPN) technical specification [6], which models grants and revocations as append-only linked-data records. The goal is not merely to notify that "something changed", but to show *what* permission changed and *for whom*, thereby improving transparency and accountability in sharing workflows.

SPARQL querying and result materialization. To support semantic data discovery, Solid Cockpit integrates an embedded SPARQL query interface [9]. Queries can target one or more RDF sources hosted in Solid Pods or exposed

as SPARQL endpoints, and are executed fully in the browser using Comunica [1]. Because raw SPARQL remains an advanced feature, the interface includes example queries of various types and query history to lower the entry barrier. In addition, Solid Cockpit supports storing query results back into a Pod as reusable linked-data artifacts according to our draft SPARQL view materialization containers (QVMC) technical specification [7]. This creates a lightweight bridge between interactive querying and persistent reuse of derived views.

3 Demonstration

We produced a short video walkthrough of key features of Solid Cockpit at `https://youtu.be/eKKpJv-1U4s`. The demonstrated scenario covers Solid Pod authentication, data upload, selective sharing with another user, notification generation after the sharing action, query execution, and result materialization for a query against a SPARQL endpoint. The live demonstrator is available at `https://knowledgeonwebscale.github.io/solid-cockpit/`.

4 Limitations and Discussion

Solid Cockpit has several limitations. First, while the interface reduces the friction of common Pod management tasks, it does not eliminate the need for RDF or SPARQL knowledge in all cases; raw querying is still an expert-oriented capability. Second, interoperability across Solid providers may vary depending on the maturity of their support for different access-control and notification mechanisms, so the application should be understood as operating within the current practical boundaries of the Solid ecosystem rather than as a guarantee of uniform behavior across all providers. Third, both LDPN and QVMC are draft technical specifications rather than mature standards, so they should be interpreted as exploratory extensions that demonstrate useful workflows, not as finalized ecosystem specifications. Finally, a broader comparative usability study against existing Solid interfaces remains future work.

5 Conclusion

Solid Cockpit addresses a practical gap in the Solid ecosystem by combining everyday Pod management with privacy-aware sharing workflows and integrated semantic querying. Its main contribution is not merely bundling existing capabilities, but making permission changes visible and auditable for users while connecting query execution to reusable Pod-native result materialization. At the same time, non-expert support for querying remains an open challenge, because direct SPARQL authoring still requires RDF knowledge. Future work will focus on broader usability evaluation, stronger interoperability testing across providers, and guided query interfaces that further reduce the need for direct SPARQL authoring.

Acknowledgments. Project funding was provided from the Research Foundation – Flanders (FWO) and from the CHIST-ERA grant CHIST-ERA-22-ORD-09. Ruben Taelman is a postdoctoral fellow of the Research Foundation – Flanders (FWO) (1202124N).

References

1. Comunica – a knowledge graph querying framework (2025), https://comunica.dev/
2. Javascript SDK | inrupt documentation (Sep 2025), https://docs.inrupt.com/sdk/javascript-sdk
3. Bailly, H., Kirrane, S., Polleres, A.: Prototyping an end-user authorization interface for the solid application interoperability specification. In: The Semantic Web: ESWC 2023 Satellite Events. Lecture Notes in Computer Science, Springer (2023). https://doi.org/10.1007/978-3-031-33455-9_33
4. Braun, C.H.J., K"afer, T.: Web push notifications from solid pods. In: The Semantic Web: ESWC 2022 Satellite Events. Lecture Notes in Computer Science, Springer (2022). https://doi.org/10.1007/978-3-031-09917-5_41
5. Capadisli, S., Berners-Lee, T., Verborgh, R., Kjernsmo, K.: Solid protocol (Dec 2020), https://solidproject.org/TR/protocol
6. Crum, E.: Linked data privacy notifications specification (2026), https://ecrum19.github.io/ldp-permissions-notifications-specification/
7. Crum, E.: SPARQL view materialization containers (2026), https://ecrum19.github.io/sparql-view-materialization-containers/
8. Hajszan, M., Buelens, B., Ertaylan, G., Lewis, D., Polleres, A., Kirrane, S.: You shall not pass (without consent): Enforcing data sovereignty with Solid in privacy-preserving semantic data platforms. Proceedings of the ACM on Management of Data (2024). https://doi.org/10.1145/3771554
9. Harris, S., Seaborne, A., Prud'hommeaux, E.: SPARQL 1.1 query language (2013), https://www.w3.org/TR/sparql11-query/
10. Solid Project: Solid project (2023), https://solidproject.org/
11. Taelman, R., Verborgh, R.: Demonstration of link traversal SPARQL query processing over the decentralized solid environment. In: Proceedings of the 27th International Conference on Extending Database Technology (EDBT) (2024)