# Miravi - a Linked Data Viewer

Els de Vleeschauwer[1], Martin Vanbrabant[1], Ben De Meester[1] and Pieter Colpaert[1]

[1]*IDLab, Department of Electronics and Information Systems, Ghent University – imec, Technologiepark-Zwijnaarde 122, 9052 Ghent, Belgium*

**Abstract**

One of the advantages of applying Linked Data practices is ease of decentralized data integration, which, however, is typically an advantage that happens at the backend of applications, and is rarely visible for an end-user. Moreover, we see a rise in applying Linked Data practices for permissioned data sharing, specifically in an industrial setting. To demonstrate the value of Linked Data practices in an industrial setting, we present "Miravi": a Web application to manage SPARQL queries and visualize their results, integrating data over decentralized heterogeneous Linked Data sources, both openly accessible and permissioned. Stemming from (and fulfilling) the requirements identified within Horizon Europe project Onto-DESIDE, Miravi demonstrates how Linked Data practices, even for permissioned data publishing, allow for ease of decentralized data integration, combining mature and well-supported open-source tools such as Inrupt's Solid Client Authentication library, IDLab's Comunica, and Marmelab's React-Admin Web framework. Miravi is currently used as part of demonstrations for projects such as Onto-DESIDE and SecuWeb, and as a dedicated search interface for Digital Flanders' OSLO program (Open Standards for Linked Organizations). Being a usecase-independent client-side application that is fully configurable with a single configuration file, and under the open MIT license, Miravi allows for a lightweight setup, whilst relying on mature software libraries helps to increase its longevity. With Miravi, we will continue to more easily demonstrate the advantages of Linked Data practices.

**Keywords**

Linked Data, Decentralized, Linked Data Consumption Viewer, Solid



## 1. Introduction

One of the advantages of applying Linked Data practices is ease of integrating data across decentralized (and heterogeneous) data sources. This leads to *data federation*: where traditional data applications require extensive data processing to align different sources' data models and data values, the use of Semantic Web standards allows to integrate data on-the-fly using a single (SPARQL) query. However, this advantage is typically noticeable at the backend of a (Linked Data) application, and is rarely visible for an end-user. To prove its value, Linked Data consumption platforms over decentralized data sources (i.e., Linked Data viewers) are needed.

Moreover, we see a rise in applying Linked Data practices for *permissioned* data sharing (i.e., publishing Linked Data behind authentication/authorization). This is specifically true in an industrial setting: the need to privately share data with and integrate data from trusted third parties. This adds requirements to a Linked Data viewer, as data is not only published *decentralized*, but also *heterogeneously*: both openly accessible and permissioned data can be integrated, using different authentication schemes, over different interfaces (e.g., publishing a single resource or via a SPARQL endpoint).

To demonstrate Linked Data's value of easy integration across decentralized, heterogeneous, and permissioned data sources in an industrial setting, we present Miravi (v2.0.0): a Web application to (i) as a Semantic Web developer, manage queries; and (ii) as a non-technical user, visualize query results. Stemming from (and fulfilling) the requirements identified within Horizon Europe project Onto-DESIDE, Miravi demonstrates how Linked Data practices, even for permissioned data publishing, allow for ease of decentralized data integration (specifically for non-technical users), combining mature

*Developers Workshop, co-located with SEMANTiCS'25: International Conference on Semantic Systems, September 3–5, 2025, Vienna, Austria*

els.devleeschauwer@ugent.be (E. de Vleeschauwer); martin.vanbrabant@ugent.be (M. Vanbrabant); ben.demeester@ugent.be (B. De Meester); pieter.colpaert@ugent.be (P. Colpaert)

0000-0002-8630-3947 (E. de Vleeschauwer); 0000-0003-0248-0987 (B. De Meester); 0000-0001-6917-2167 (P. Colpaert)

and well-supported open-source tools. Miravi is published at https://github.com/SolidLabResearch/miravi-a-linked-data-viewer/releases/tag/v2.0.0, under MIT License.

After discussing the identified requirements and matching them with related works (Section 2), we describe the architecture and implementation (Section 3), and the use cases Miravi has been applied to (Section 4). Finally, we conclude (Section 5).

## 2. Motivating use case and requirements

In the Ontology-based Decentralized Sharing of Industry Data in the European Circular Economy project (**Onto-DESIDE**)[1], semantic ontologies and Linked Data are combined with the concept of decentralized data sharing to enable data collaboration in the context of circular economy [1].

The Open Circularity Platform serves as the technical backbone for Onto-DESIDE [2]. It enables the automated sharing of existing data in a way that ensures both *semantic and technical interoperability*, while allowing data owners to keep full control over their data. To this end, the platform makes extensive use of both existing and emerging standards. The remainder of this section will introduce and reference these standards.

Extended use of Linked Data practices from *open* data sharing to *permissioned* data sharing (i.e., requiring authentication and authorization) are exemplified by standardization efforts from, e.g., the Solid project[2] and the Fedora Repository[3], which both extend the W3C recommended Linked Data Platform (LDP) [3], decoupling data storage from applications. Several other Linked Data sharing technologies foresee (customized) solutions for access control, e.g., customized SPARQL endpoints such as GraphDB[4] and Linked Data platforms such as Metaphactory[5] [4].

Within Onto-DESIDE, permissioned data sharing was implemented via the Solid project, which introduces an extension to the OpenID-Connect (OIDC) protocol [5] for authentication, namely, Solid-OIDC [6]. Individual users and organizations can store their data as a Web resource in online data stores, called pods. Users decide who has access to their resources, granting permissions to specific resources on their pod.

To demonstrate the value of the Open Circularity Platform (and thus, of Linked Data for easy data integration) to non-technical users, a user-friendly application was needed. To meet this need, Miravi was developed. The following requirements were identified for the development of Miravi, derived from user stories formulated by Onto-DESIDE industry partners[6]:

**Authenticate [R1]** Initiate a secure session to interact with your and other actor's data, behind a layer of access control.

**Query [R2]** Retrieve specific data from one or more actors in the network. This data is thus by nature decentralized, and can be a combination of permissioned and public data.

**View [R3]** Create custom views on top of existing data sources.

**Calculate [R4]** Calculate derived information from existing data.

**Validate [R5]** Validate that the retrieved data is genuine and has not been tampered with.

**Scalability [R6]** Scale the management with the number and the heterogeneity of sources and queries.

**Fault tolerance [R7]** Have redundancy by design, so that a failing node does not impact the entire ecosystem.

[1] https://ontodeside.eu

[2] https://solidproject.org/, taken up by the W3C Working Group on Linked Web Storage: https://www.w3.org/2024/09/linked-web-storage-wg-charter.html

[3] https://fedorarepository.org/

[4] https://graphdb.ontotext.com/documentation/10.7/access-control.html

[5] https://help.metaphacts.com/resource/Help:Security

[6] https://ontodeside.eu/results/, Deliverable 2.2 "Project requirements specification and research methodology".

**Reproduce [R8]** Provide well-documented (open-source) code and APIs.

Multiple Linked Data viewers exists. Examples are collected at, e.g., https://linkeddata-89b9d.web.app/, and compared in several surveys [7, 8, 9]. Most (mature) Linked Data viewers build on top of SPARQL queries that are directly targeted at a (single) SPARQL endpoint or RDF resource, and use the query result types to create visualizations for the query results (e.g., query results that contain latitude and longitude are visualized on a map). Access control support relies on the functionality of the employed SPARQL endpoint, typically basic HTTP Authentication, i.e., either full access on a (named) graph or no access at all, or vendor-specific authentication schemes.

## 3. Architecture and implementation

In this section we describe the architecture and implementation of Miravi. We link the supported features to the requirements listed in Section 2.

### 3.1. Design

As an architecture covering the requirements, we present a client-side Web application without server-side processing. This simplifies the overall technological setup: as all processing is done client-side, no additional component can fail. This does exclude the possibility to move any heavy processing to a server-side component and could thus impact overall performance of the application.

To implement this Web application, we maximally rely on mature and well-supported open-source tools such as Inrupt's Solid Client Authentication library, IDLab's Comunica, and Marmelab's React-Admin Web framework. We develop the application in a single programming environment, using the Javascript programming language and packaging for browser usage with Vite.

To access permissioned data on Solid Pods, you must *authenticate* **[R1]** as a user who has been granted appropriate access to that data. We applied Solid-OIDC, being the Solid authentication protocol. Miravi reuses Inrupt's Solid Client Authentication library[7] to authenticate the users in a browser, producing a custom fetch function, which you can use to execute authenticated HTTP requests.

To retrieve and manipulate the data, we applied the SPARQL Query Language (SPARQL) [10], the W3C recommended query language for Linked Data. SPARQL supports federated queries to retrieve data from decentralized sources **[R2]**, and includes projection, filter, sort and aggregation operators, as well as transformation functions, to construct custom views **[R3]** with calculated values **[R4]**, without the need for supplementary processing pipelines.

Existing Linked Data viewers primarily rely on a single (SPARQL) endpoint. To support a decentralized ecosystem, we want to support multiple endpoints, scalable to heterogeneous interfaces **[R6]**: we can not assume that each actor will set up a (resource-intensive) SPARQL endpoint to share data, instead, we assume that heterogeneous data sources are made available, of which we primarily focused on RESTful Web APIs, e.g., the Linked Data Platform API as integrated in the Solid protocols.

Several query engines that support federated SPARQL querying exist [11], however, for the development of Miravi, we selected Comunica[8]. Comunica is a flexible and modular knowledge graph querying framework, allowing for client-side query execution of SPARQL queries over decentralized Linked Data [11]. Comunica fits our requirements: (i) it is a SPARQL-compliant query engine **[R2-4]**; (ii) it supports multiple decentralized heterogeneous data sources, so it is functionally scalable **[R6]**; (iii) it can be run client-side, being compatible with building a purely client-side Web application, so it provides sufficient redundancy to recover from calamities **[R7]**; (iv) it is a mature software library with extensive documentation **[R8]**; and (v) its extensibility has been applied to support Solid-OIDC authentication, making it compatible with the Solid authentication protocol **[R1]**.

In Miravi we forward the custom fetch function – produced by the Inrupt library – to Comunica to make it perform *authenticated queries* over Solid pods and other Linked Data sources.

[7] https://docs.inrupt.com/developer-tools/javascript/client-libraries/authentication/
[8] https://comunica.dev/

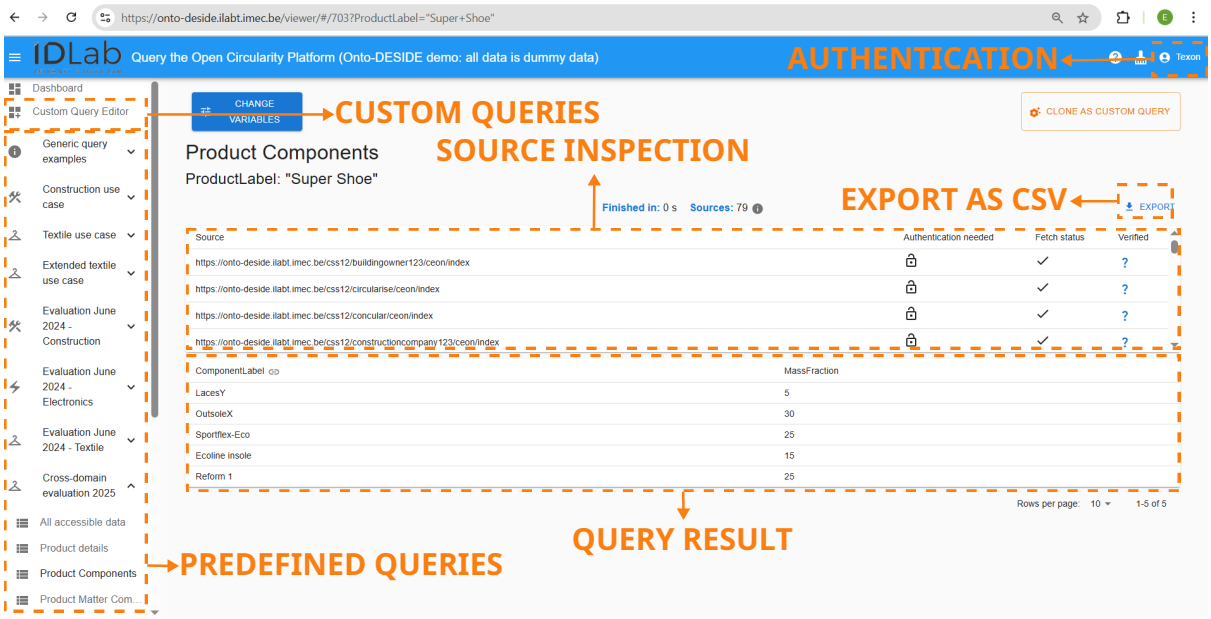


**Figure 1:** With the Miravi user interface a user can authenticate himself, execute predefined or custom queries over open RDF sources as well as protected RDF sources on a Solid pod, inspect the sources of the executed queries, view their result in a tabular format, and export the result as a CSV file.

Comunica has an online user interface[9], targeting users with knowledge of Linked Data and SPARQL. Miravi adds more features (Section 3.2), specifically supporting non-technical users. Its user interface (Figure 1) is built with Marmelab's React-Admin[10], an open-source framework for building Web data dashboards, in which we wrap the authentication and query capabilities of Miravi. The configuration of the interface is fully customizable using a single configuration JSON file. Titles, descriptions and icons can be adapted, but mostly the queries can be predefined, grouped in categories.

To validate that the retrieved data is genuine and has not been tampered with **[R5]**, we applied W3C recommended Verifiable Credentials[11].

## 3.2. Functionalities and Interface

At its base, Miravi is a data dashboard that integrates data from decentralized (dynamically discoverable) data sources, triggered by underlying configured SPARQL queries (or query templates), to visualize the results in a tabular format.

Miravi allows – for each configured query – to set a predefined set of data sources to query. However, to avoid the need to configure each data source individually, Miravi supports *data source discovery*: we allow to start from a single index source, and then apply Comunica's link traversal functionality [12]. Starting from one source, all links from that source are following to recursively process each Linked Data document to get a complete set of data sources.

Miravi allows specifying preconfigured SPARQL queries. However, to avoid the need to configure each query for each instance separately, we can apply *query templates* that contain parameters. Before a SPARQL query is submitted, each template variable is dynamically replaced with the actual value selected by the user. The list of available values for each variable can either be predefined or are the result set of another query, i.e., dynamically constructed from data. This approach allows queries to be customized per data type, even when new data of the same type is added.

By combining data source discovery and query templates, the configuration becomes scalable **[R6]**: queries do not need to be adapted when more data sources and more data are added to the network.

[9] https://query.comunica.dev/
[10] https://marmelab.com/react-admin/
[11] https://www.w3.org/TR/vc-data-model-2.0/

The predefined queries enable *views* and derived *calculations* **[R3-4]** on the data sources (Figure 1, left), understandable for users without any knowledge of SPARQL or Linked Data (Figure 1, bottom center). Alternatively, users can also create and share their own custom queries in the interface, resulting in additional custom views (Figure 1, top left). Additionally, users can export query results as CSV files for further analyses and more complex calculations (Figure 1, top right). The user can also inspect the protection, fetch and verification status of the consulted data sources (Figure 1, top center). The verification status *validates* **[R5]** any Verifiable Credentials wrapping the data to assure the retrieved data is genuine and has not been tampered with.

Mirava is published on GitHub at https://github.com/SolidLabResearch/miravi-a-linked-data-viewer under the open MIT-license, and with extensive documentation, including a screencast demonstrating its main features, and possibilities to register issues, making it *reproducible* and reusable **[R8]**.

## 4. Application

At the moment of writing, customized instances of Miravi have been set up for three projects: the OSLO Knowledge Graph, Onto-DESIDE, and Secuweb.

**Open Standards for Linked Organizations** (OSLO)[12] is a project of Digital Flanders, that has published at the moment of writing 251 vocabularies and application profiles, at https://data.vlaanderen.be. Miravi was used to publish a dedicated search interface over the published vocabularies and application profiles, available at https://w3id.org/imec/knows/oslo-kg-viewer, as an aid for developers of future standards as well as users of existing standards. Within this setup, data is queried from a small[13] set of public raw Turtle data files published directly from Github, with a small set of query templates.

**Onto-DESIDE** is a Horizon Europe project on data sharing for increased circular economy. Access control is an important requirement for data sharing in this context. The development of Miravi was a part of the Onto-DESIDE project, and enabled the setup of an Onto-DESIDE viewer, available at https://w3id.org/imec/knows/onto-deside/viewer. Within this setup, the integration of a medium[14] set of permissioned decentralized Linked Data sources is demonstrated, shared under Solid-compliant access control.

A similar demonstrator was set up for **Secuweb**[15], an Interreg Europe project, aiming to strengthen data security in companies and organizations in Flanders, Wallonia, and France. With Miravi, a Secuweb viewer was instantiated, integrating decentralized Linked Data from the Food Supply Chain domain, with access control, screencast available at https://www.youtube.com/watch?v=CgCVOqOTFFg.

## 5. Conclusion

Being a usecase-independent client-side application that is fully configurable with a single configuration file, Miravi allows for setting up lightweight Linked Data dashboards that support decentralized and heterogeneous data sources, both open and permissioned (currently supporting Solid-compliant access control). Data source indexes and SPARQL query templates are applied to easily scale the number of data sources and query types. The user interface facilitates access to preset queries for non-technical uses and ad hoc configuration of custom queries for administrators. Query results are presented in a tabular format and can be exported to CSV to prevent lock-in.

Miravi is usecase-independent and flexible, as evidenced by the three current setups for the OSLO Knowledge Graph, Onto-DESIDE, and Secuweb.

Miravi is built on mature software libraries with extensive documentation and published on GitHub under the open MIT license, helping to increase its longevity. Future work includes improving Miravi's

[12] https://www.vlaanderen.be/digitaal-vlaanderen/onze-diensten-en-platformen/oslo
[13] Queries integrate on average 2 data sources.
[14] Queries integrate on average 70 data sources.
[15] https://www.interreg-fwvl.eu/nl/projecten/secuweb

current JSON configuration to a semantic (JSON-LD) configuration, to improve internationalization support and make query template reusable outside of the configured project.

With Miravi, we will continue to more easily demonstrate the advantages of Linked Data practices.

## Acknowledgments

The described research activities were supported 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

The author(s) have not employed any Generative AI tools.

## References

[1] Blomqvist, Eva and Lindecrantz, Mikael and Blomsma, Fenna and Lambrix, Patrick and De Meester, Ben, Decentralized digital twins of circular value networks : a position paper, in: García-Castro, Raúl and Davies, John (Ed.), Proceedings of the Third International Workshop on Semantic Digital Twins (SeDiT 2022), volume 3291, CEUR, 2022, p. 8. URL: https://ceur-ws.org/Vol-3291/paper1.pdf.

[2] G. De Mulder, E. de Vleeschauwer, B. De Meester, P. Colpaert, O. Hartig, The Open Circularity Platform: a Decentralized Data Sharing Platform for Circular Value Networks, in: Proceedings of the 2nd International Workshop on Knowledge Graphs for Sustainability (KG4S), Hersonissos, Greece, 2024. URL: https://ceur-ws.org/Vol-3753/paper4.pdf.

[3] S. Speicher, A. Malhotra, J. Arwe, Linked Data Platform 1.0, Recommendation, World Wide Web Consortium (W3C), 2015. URL: http://www.w3.org/TR/ldp/.

[4] P. Haase, D. M. Herzig, A. Kozlov, A. Nikolov, J. Trame, metaphactory: A platform for knowledge graph management, Semantic Web 10 (2019) 1109–1125. doi:10.3233/SW-190360.

[5] OpenID Connect Protocol, Technical Report, Auth0 by Okta, ???? URL: https://auth0.com/docs/authenticate/protocols/openid-connect-protocol.

[6] A. Coburn, e. Pavlik, D. Zagidulin, Solid-OIDC, Editor's Draft, Solid Community Group, 2022. URL: https://solidproject.org/TR/oidc.

[7] J. Klímek, P. Škoda, M. Nečaský, Survey of tools for linked data consumption, Semantic Web (2019). doi:10.3233/sw-180316.

[8] F. Desimoni, L. Po, Empirical evaluation of linked data visualization tools, Future generations computer systems (2020). doi:10.1016/j.future.2020.05.038.

[9] E. Bernasconi, M. Ceriani, D. D. Pierro, S. Ferilli, D. Redavid, Linked data interfaces: a survey, Italian Research Conference on Digital Library Management Systems (2023). doi:10.3390/info14090483.

[10] C. B. Aranda, O. Corby, S. Das, L. Feigenbaum, P. Gearon, B. Glimm, S. Harris, S. Hawke, I. Herman, N. Humfrey, N. Michaelis, C. Ogbuji, M. Perry, A. Passant, A. Polleres, E. Prud'hommeaux, A. Seaborne, G. T. Williams, SPARQL 1.1 Overview, Recommendation, World Wide Web Consortium (W3C), 2013. URL: http://www.w3.org/TR/sparql11-overview/.

[11] Taelman, Ruben and Van Herwegen, Joachim and Vander Sande, Miel and Verborgh, Ruben, Comunica : a modular SPARQL query engine for the web, in: SEMANTIC WEB - ISWC 2018, PT II, volume 11137, 2018, pp. 239–255. URL: http://doi.org/10.1007/978-3-030-00668-6_15.

[12] R. Taelman, R. Verborgh, Link Traversal Query Processing over Decentralized Environments with Structural Assumptions, in: ISWC2023, 2023. URL: https://comunica.github.io/Article-ISWC2023-SolidQuery/.