Privacy-Preserving Family Media Profiles in Decentralized Data Ecosystems

Els de Vleeschauwer[1], Maarten Vandenbrande[1], Karim Dahdah[2], Pieter Bonte[3], Femke Ongenae[1], Ben De Meester[1] and Pieter Colpaert[1]

[1]IDLab – Ghent University – imec, Technologiepark-Zwijnaarde 122, 9052 Ghent, Belgium
[2]VRT, Auguste Reyerslaan 52, 1043 Brussels, Belgium
[3]Department of Computer Science, KU Leuven Campus Kulak, Etienne Sabbelaan 53, 8500 Kortrijk, Belgium

Abstract

Decentralized data ecosystems (e.g., Solid, Linked Web Storage) allow users to retain control over their personal data while enabling data reuse across applications. However, additional mechanisms are required to create and share derived group-level data without granting consumers access to the underlying resources. This challenge arises in the Solid4Media family media profile use case, where family members want to benefit from a shared profile without exposing their individual media profiles to one another. In this paper, we present a privacy-preserving, transparent, interoperable, and reusable approach for aggregating decentralized data. We incorporate a third-party component on top of Solid pods: a provisioned aggregator. Following an open and implementation-independent aggregator specification, the aggregator is assigned its own identity and enables multiple data contributors to govern downstream access to an aggregated data resource without exposing the contributing source data. We validate this approach on the Solid4Media family media profile use case. Our demonstrator, available at http://w3id.org/imec/knows/solid4media, shows that a provisioned aggregator can generate and expose a shared family media profile while preserving access restrictions on the individual media profiles. This realization of the family pod use case demonstrates the feasibility of the provisioned aggregator, thus complementing the other aggregator types and the ongoing standardization efforts of the Linked Web Storage Working Group.

Keywords

Solid, Aggregator, Family Media Profile, Decentralized Data Ecosystem

CEUR-WS.org/Vol-4322/paper_11.pdf

1. Introduction

Today, media profiles (e.g., a person's preferences, likes, and dislikes) are distributed across global media platforms, such as Netflix and Spotify, and local platforms, such as the Belgian VRT Max, VTM GO and GoPlay. Each platform captures only a partial and isolated view of the user. Combining information from multiple profiles can support richer services, including cross-platform personalization and recommendation [1].

To explore this potential, the **Solid4Media**[1] living lab was launched. The consortium brings together a public broadcaster (VRT), data technology partners (Datavillage and Athumi), and Ghent University research groups in socio-technical, semantic web, and techno-economic domains. In close collaboration with stakeholders and end users, Solid4Media investigates how decentralized media infrastructures can give users more control over their personal data while enabling its reuse across media platforms and in adjacent domains, such as culture. Linked Data principles provide an important foundation for integrating data from distributed sources and therefore serve as guiding principles within the project.

DMKG'26: 2nd International Workshop on Data Management for Knowledge Graphs, October 2026, Bari, Italy
els.devleeschauwer@ugent.be (E. de Vleeschauwer); maarten.vandenbrande@ugent.be (M. Vandenbrande); karim.dahdah@vrt.be (K. Dahdah); pieter.bonte@kuleuven.be (P. Bonte); femke.ongenae@ugent.be (F. Ongenae); ben.demeester@ugent.be (B. De Meester); pieter.colpaert@ugent.be (P. Colpaert)
0000-0002-8630-3947 (E. de Vleeschauwer); 0009-0003-6470-9123 (M. Vandenbrande); 0009-0004-8466-5348 (K. Dahdah); 0000-0002-8931-8343 (P. Bonte); 0000-0003-2529-5477 (F. Ongenae); 0000-0003-0248-0987 (B. De Meester); 0000-0001-6917-2167 (P. Colpaert)
© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

[1]Launched June 2023 as a part of the Flemish recovery programme "Digital Transformation and Innovation in Flemish Media", https://www.solid4media.be/.

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073
published 2026-11-09

One of the use cases identified in Solid4Media[2] is the *family pod*: family members want to receive household-level recommendations, based on their combined interests and preferences, without disclosing their individual media profiles to external data consumers or to one another. Implementing this use case revealed a specific technical challenge: while current decentralized data ecosystems (e.g., Solid, Linked Web Storage) support user-controlled access to individual resources, creating and sharing derived group-level data requires additional mechanisms to ensure that access to the aggregated results does not imply access to the underlying resources. A straightforward client-side approach requires direct access to the individual profiles, which undermines the intended separation between access to the group-level results and access to its contributing resources. Privacy-preserving distributed computation approaches, e.g., federated learning or secure multiparty computation, may avoid centralizing raw data, but are typically designed for specific computational settings and do not necessarily provide a reusable mechanism for transparently materializing, governing, and sharing derived resources in a decentralized data ecosystem.

In this paper, we propose a privacy-preserving, transparent, interoperable, and reusable approach to share aggregated data, applied to a media use case. We apply the Trustflows principles[3] [2], the Solid Protocols[4] [3, 4] (which decouple personal data storage from applications using open web standards) and the open, implementation-independent Aggregator specification [5]. The resulting architecture combines Solid pods for decentralized data storage, an authorization server, and a third-party provisioned aggregator that provides a family media profile and allows multiple data contributors to govern downstream access to the aggregated resource without exposing their contributing upstream data sources. We validate this approach on the Solid4Media use case and derive architectural lessons regarding the governance and maintenance of aggregated resources in decentralized ecosystems. The demonstrator, available at http://w3id.org/imec/knows/solid4media, shows how a provisioned aggregator can generate and expose an aggregated family media profile while preserving the access restrictions applied to the contributing individual profiles.

2. Related Work

There have been multiple efforts to share aggregated data derived from decentralized data sources. Data Mesh promotes decentralization of data within the infrastructure of a single organization and allows exposing and consuming data through *data products*, which encapsulate data sources, transformations, and access interfaces [6]. Decentralized social protocols, such as ActivityPub[5] (used by Mastodon) and Bluesky's AT Protocol [7], precompute and materialize *derived views* in decentralized environments. Within data spaces, *services* that operate on data rather than exposing raw data directly, were proposed [8], implemented in practice by frameworks, such as Eclipse Data Space Components (EDC). Van Herwegen et al. [9] propose *derived data resources*: Solid Protocol compliant resources enabled through server-side selection, transformation, and exposure of data from multiple resources within a single pod. Libertas is a modular system architecture that integrates *Secure Multi-Party Computation* (MPC) with Solid pods or similar decentralized personal data stores [10]. It enables collective data computation without revealing individual inputs, eliminating the need to centralize data or trust a single entity.

None of these existing efforts addresses how access control policies from these multiple independent data providers can be preserved and enforced on the resulting derived data. Moreover, they either (i) cannot combine data from decentralized sources [9], (ii) are not designed for reuse (i.e., only work within the organizational boundaries) [6], or (iii) are application-specific [7, 8]).

Vandenbrande et al. [11] introduce an *aggregator*: a third-party component that materializes SPARQL query results over distributed RDF sources to mitigate performance challenges in decentralized ecosystems, such as link traversal and complex query evaluation [5], applied to a watch-party demonstrator.

[2] http://w3id.org/imec/knows/solid4media/usecases
[3] https://trustflows.eu/charter/
[4] https://solidproject.org/
[5] https://www.w3.org/TR/activitypub/

The owners of the resources contributing to the query results fully govern the access to those results.

3. Requirements Analysis

Solid4Media aims to give users control over their media data while enabling the reuse of that data across different applications and domains. This is translated to the following functional requirements:

Decentralized Data Storage [R1] Users choose where they store their data.

Access Control [R2] Users decide who can access their data.

Technical Interoperability [R3] Data is reusable by multiple applications.

Semantic Interoperability [R4] Data is reusable across multiple domains.

The project adopts the Solid protocol to satisfy these requirements. Solid allows users to store their data in resources on pods hosted wherever they prefer (R1) and to restrict access to specific resources to selected actors (R2). Solid secures that data stored in pods can be accessed via standard operations (R3) and promotes RDF as data format, providing a shared framework for data representation and alignment of data models (R4).

The family pod use case seeks a technical solution that satisfies these requirements for aggregated data, thus impacting R1 and R2. Specifically, the family should be able to choose where the aggregated data is hosted (R1) and control who can access their data, even when their data is part of an aggregated derived data set. (R2). Importantly, access to aggregated data should *not imply access to individual source data*, allowing each family member to preserve the privacy of their personal media profile.

4. Approach

In the family pod use case, the family media profile aggregates data from multiple resources distributed across different Solid pods, each containing a media profile of a family member or a part of such media profile, represented in RDF. A federated SPARQL query over these resources can compose an aggregated family media profile. However, executing this query, which could happen by any family member's application, currently requires that family members expose their individual media profiles to each other if they all want access to the family media profile.

To avoid such exposure of upstream media profiles, we introduce an aggregator complying with the Aggregator Protocol [5] (Figure 1). Compliance to this open specification ensures transparency and keeps technical interoperability (R3). The aggregator is hosted on an aggregator server: an HTTP-based interface that lets a client create and manage aggregators and configure aggregator services that execute data transformations. Similar to a Solid server, an aggregator server can be hosted by any user or organization at any location, securing decentralized data storage (R1).

The Aggregator Protocol enforces authentication and authorization using open specifications. A user authenticates himself with a registered identity at an identity provider that complies with OpenID Connect (OIDC)[6]. For authorization, the Aggregator Protocol relies on User Managed Access (UMA)[7], through the Authorization for Data Spaces (A4DS) profile[8]. A UMA-based authorization server allows access decisions to be governed by policies defined outside the resource server itself, and is as such applied to the aggregator and all upstream resource hosting Solid pods[9]. We deployed an authorization server that employs an ODRL-based policy model[10]. ODRL is capable of expressing broader usage

[6] https://openid.net/specs/openid-connect-core-1_0.html
[7] https://docs.kantarainitiative.org/uma/rec-uma-core.html
[8] https://spec.knows.idlab.ugent.be/A4DS/L1/latest/
[9] This is an extension to the original Solid protocol, in line with the extensibility as foreseen in the Linked Web Storage First Public Working Drafts: https://w3c.github.io/lws-protocol/.
[10] https://w3c.github.io/odrl/

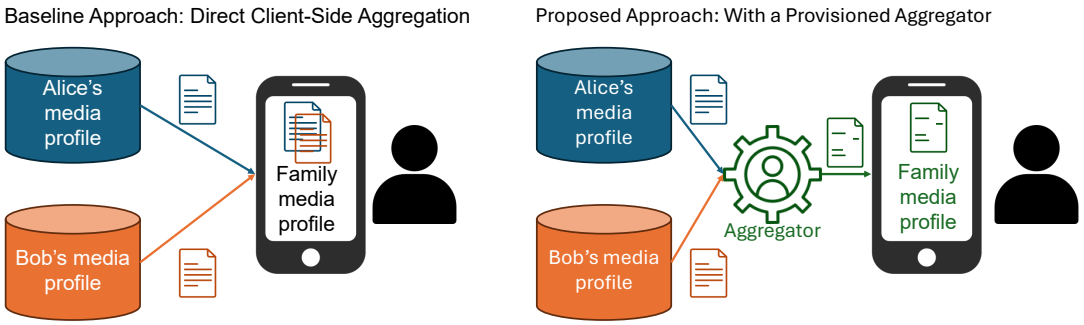


Figure 1: In direct client-side aggregation, consumers need access to all underlying resources to construct the family media profile. With a provisioned aggregator, consumers access the aggregated resource, while only the aggregator requires access to the underlying data.

policies. This opens the possibility of governing not only who may access data, but also how that data may be processed by data consumers.

We specifically instantiate a *provisioned* aggregator[11]: a provisioned aggregator gets its own identity at the aggregator server level, along with a dedicated authorization server, without exposing the aggregator's credentials to any of the users. In our use case, a trusted intermediary hosts the aggregator server and provides it with its identity. Each family member grants access to the aggregator for reading their media profile. The resulting family media profile is an aggregated resource. If a family member revokes the aggregator's access to their media profile, that family member's data is excluded from the subsequent aggregation. Access rules of that aggregated resource are managed through access delegation[12], ensuring that the owners of the original data remain in control of the access rules for the aggregated resource: access to the aggregated resource must be approved by *all* data contributors. In our use case, family members grant the other family members read access to the derived family media profile. This satisfies the access control requirement (R2).

The Aggregator Protocol foresees a declarative description of transformation services executed under the aggregator's authority, offering transparency to the data contributors and users. For our use case, the transformation evaluates a federated SPARQL CONSTRUCT query over the upstream media profiles maintained by individual family members, to compose the family media profile. The output of this transformation is RDF data, supporting semantic interoperability (R4).

The following functionalities were added to the transformation to increase maintainability and privacy protection when using an aggregator. On the one hand, the transformation *dynamically discovers and retrieves the query sources* from a resource stored and maintained outside the aggregator. This design enables the addition and removal of sources without requiring changes to the aggregator service itself. On the other hand, the transformation *only produces output when data is contributed by at least a predefined minimum number of sources.* This threshold-based approach helps ensure that individual contributors cannot be identified from the resulting profile. This approach is internally enforced by our exemplary transformation service and exposed as part of the declarative transformation description, allowing contributors to inspect the condition under which aggregated results are produced.

The presented transformation should be regarded as an example. The Aggregator Protocol does not prescribe specific privacy-preserving or maintenance mechanisms. However, such mechanisms can be encoded in the transformation logic. Depending on the use case, transformations may enforce different constraints, aggregation rules, or disclosure thresholds before exposing derived results.

[11] https://w3id.org/aggregator#creation-provision
[12] https://w3id.org/aggregator#access-delegation

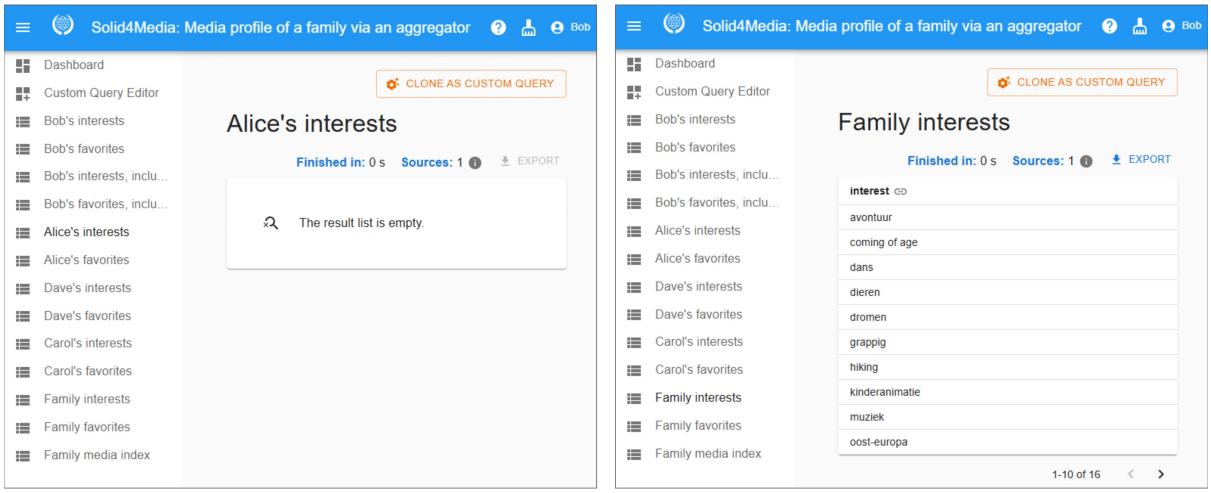


Figure 2: When Bob logs in, he can view the family's aggregated interests, but not Alice's individual interests.

5. Validation

Our demonstrator, online available (including screencasts) at http://w3id.org/imec/knows/solid4media. shows how the requirements of the family pod use case are satisfied by the introduction of the provisioned aggregator. Its source code is available on GitHub[13] under the permissive MIT license.

Our demonstrator consists of four services: a resource server service, an identity provider service, an authorization server service, and an aggregator server service. Additionally, we set up a linked data viewer using Miravi [12] to simulate an application, such as a recommender system accessing the decentralized data, and an access management application using LOAMA [13] to demonstrate how users manage access to their own resources and to the aggregated resource. The resource server service and identity provider service are provided by a Community Solid Server with a UMA-compliant configuration. On this server, we set up family member pods for mother Alice, father Bob, daughter Carol, and son Dave. The aggregator's identity is also provided. The media profiles of the family members are stored as resources on their own pod. The authorization for accessing resources on the Solid pods is handled by the UMA-compliant authorization server service. The aggregator server hosts an aggregator that exposes the family media profile. Bob has created the aggregator server, but Bob cannot access the resources the aggregator server has access to.

The linked data viewer is a UMA-compliant version of Miravi that demonstrates the enforcement of restricted data access. When a family member logs in, he gets access to the interests and favourites from himself and the family, but not to those of the other family members (Figure 2).

Through the access management application, an UMA- and aggregator-compliant version of LOAMA, users can adapt the access policies to their resources and accept access requests. A family member can revoke the aggregator's access to his media profile, thus removing his data from the family media profile. An external party, e.g., a researcher, may request access to the family media profile; access is granted only if all family members accept the request.

The online demonstrator is compared to a baseline scenario to illustrate the limitations of a Solid approach without aggregator: the family media profile is then achieved by a federated query over a family index listing the individual media profiles of each family member, and then performing link traversal. To execute this query, the logged-in family member needs read access to the individual media profiles of all other family members.

[13] https://github.com/KNowledgeOnWebScale/solid4media-architecture/releases/tag/v1.1.0

6. Conclusion

The Aggregator Protocol provision flow shows how an aggregator with its own identity enables the creation and controlled sharing of aggregated data without exposing individual-level information. Through the Solid4Media family media profile use case, we demonstrate that provisioned aggregators can serve as a reusable architectural building block for sharing derived resources in decentralized data ecosystems while maintaining a separation between access to aggregated results and access to contributing source data.

The implementation revealed practical requirements for operating such aggregators in real-world scenarios. In particular, dynamic source discovery and threshold-based output generation proved useful to support maintainability and reduce disclosure risks. These mechanisms illustrate how application-specific requirements can be realized through aggregator transformations and are not limited to the family media profile use case. The aggregator acts as an automated and transparent processing pipeline whose transformation logic is explicitly defined and can be inspected by contributors. This provides visibility into how data is accessed and transformed to produce aggregated resources. However, as with other infrastructure components in decentralized ecosystems, users ultimately rely on the correct implementation of the aggregator, authorization framework, and transformation services.

The proposed approach satisfies the requirements of the Solid4Media living lab, namely decentralized data storage, fine-grained access control, and both technical and semantic interoperability. More specifically, it enables family members to benefit from a shared media profile without requiring direct access to one another's individual media profiles. The demonstrator confirms the feasibility of the approach and highlights its advantages over alternative solutions, such as direct link traversal. Beyond feasibility, our work contributes to ongoing standardization efforts by providing a concrete implementation and validation of the provisioned aggregator.

Where this paper presents the proof of validity, several open challenges remain. Performance impact with respect to link traversal during data discovery and complex queries containing expensive operators has received initial attention [11], but more extensive benchmarking is needed. The privacy-preserving aspects of the approach have been demonstrated, but not (formally) verified.

Together, these efforts will contribute to making privacy-preserving aggregation a practical and reusable building block for decentralized data ecosystems.

Acknowledgments

The described research activities were supported by the living lab Solid4Media (Agentschap Innoveren en Ondernemen project nr. HBC.2022.0770) with partners VRT, Datavillage, Athumi, mict-UGent, IDLab-KNoWS, and IDLab-TE.

Declaration on Generative AI

During the preparation of this work, the authors used Copilot for grammar and spelling checks and coding. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] X. Meng, Cross-domain information fusion and personalized recommendation in artificial intelligence recommendation system based on mathematical matrix decomposition, Sci. Rep. 14 (2024) 7816.

[2] B. De Meester, J. Rojas Meléndez, F. Ongenae, P. Colpaert, R. Verborgh, Tackling the write-to-read Web of data with Trustflows, in: I. Celino, O. Hassanzadeh, A. Bernstein, N. Noy, G. Cheng, S. Wang, S. Ferrada, T. Soulard, K. Kozaki, H. Takeda, A. L. Gentile (Eds.), Proceedings of the

Industry, Doctoral Consortium, Posters and Demos at ISWC 2025, volume 4085 of *CEUR Workshop Proceedings*, CEUR, 2025, pp. 233–237.

[3] A. V. Sambra, et al., Solid: a platform for decentralized social applications based on linked data, MIT CSAIL & Qatar Computing Research Institute, Tech. Rep. (2016).

[4] R. Verborgh, Re-decentralizing the Web, for good this time, in: O. Seneviratne, J. Hendler (Eds.), Linking the World's Information: Essays on Tim Berners-Lee's Invention of the World Wide Web, ACM, 2023, pp. 215–230. doi:10.1145/3591366.3591385.

[5] M. Vandenbrande, P. Nachtergaele, E. de Vleeschauwer, B. De Meester, P. Bonte, F. Ongenae, Aggregator Specification, Technical Report, 2026. URL: https://w3id.org/aggregator/.

[6] A. Goedegebuure, I. Kumara, S. Driessen, W.-J. Van Den Heuvel, G. Monsieur, D. A. Tamburri, D. D. Nucci, Data mesh: A systematic gray literature review, ACM Comput. Surv. 57 (2024) 11:1–11:36. doi:10.1145/3687301.

[7] M. Kleppmann, et al., Bluesky and the AT protocol: Usable decentralized social media, in: Proceedings of the ACM Conext-2024 Workshop on the Decentralization of the Internet, ACM, 2024, pp. 1–7. doi:10.1145/3694809.3700740.

[8] B. T. Arnold, C. Lange, C. Gillmann, S. Decker, A Service Architecture for Dataspaces, 2025. doi:10.48550/arXiv.2507.07979. arXiv:2507.07979, preprint.

[9] J. Van Herwegen, R. Verborgh, Granular Access to Policy-Governed Linked Data via Partial Server-Side Query, in: The Semantic Web: ESWC 2024 Satellite Events, volume 15344 of *Lecture Notes in Computer Science*, Springer Nature Switzerland, 2025, pp. 331–335. doi:10.1007/978-3-031-78952-6_52.

[10] R. Zhao, et al., Libertas: Privacy-preserving collaborative computation for decentralized personal data stores, volume 9, ACM, 2025. doi:10.1145/3757490.

[11] M. Vandenbrande, E. Nijs, P. Bonte, F. Ongenae, Privacy gets social: Scalable watch-parties across personal data vaults, in: Proceedings of the 23rd European Semantic Web Conference (ESWC 2026), Posters and Demos Track, Lecture Notes in Computer Science (LNCS), Springer, 2026. To appear.

[12] E. de Vleeschauwer, M. Vanbrabant, B. De Meester, P. Colpaert, Miravi : a linked data viewer, in: SEMANTICS-PDWT 2025, volume 4064, CEUR, 2025, p. 6.

[13] W. Slabbinck, et al., LOAMA : low-code ODRL access management application, in: ISWC-C 2025 : Industry, Doctoral Consortium, Posters and Demos at ISWC 2025, volume 4085, CEUR, 2025, pp. 502–510.