Tackling the Write-to-Read Web of Data with Trustflows

Ben De Meester, Julian Andrés Rojas, Femke Ongenae, Pieter Colpaert and Ruben Verborgh

IDLab, Ghent University – imec, Technologiepark-Zwijnaarde 122, 9052 Ghent, Belgium

Abstract

The need for the read–write Linked Data Web is currently implemented as a Create, Read, Update or Delete (CRUD) strategy on resources, which implies that agents will read exactly what has once been written. In practice, however, deploying writeable Linked Data nodes in an ecosystem reveals the following problems. Data reuse is hindered as later data usage requirements diverge from how the data was originally written, as well as by evolving data models. Moreover, the absence of an authoritative source to guarantee auditability forces individual data point verification at the read side. We present key characteristics of writeable Linked Data nodes (cope with longevity for written data, combine read data using different models, and provide an explicit trust context) as arguments justifying the need for a more complex operational model. We apply a Command Query Responsibility Segregation pattern to decouple write and read interfaces, introduce semantic mappings within the writeable Linked Data node to support evolving data read requirements, and apply Event Sourcing aligned with PROV-O's Actor-Entity-Activity model to provide an explicit trust context. We term this operational model "Trustflows". Applied to the PACSOI use case, Trustflows showcases the additional affordances of handling data coming in different granularities, from different sources, adhering to different reading requirements.

Keywords

read–write Linked Data Web, CRUD, CQRS, Event Sourcing

1. Introduction

The need for the read–write Linked Data Web [1] is evidenced by the recommendation of W3C's Linked Data Platform [2], and its uptake in, e.g., the Solid protocol [3] and the Fedora project [4], currently being standardized in the W3C Linked Web Storage Working Group[1]. In the context of this paper, we define the concept of a *writeable Linked Data node*: an online storage service that publishes Linked Data, where that Linked Data can be created and edited by multiple (authorized) actors in an ecosystem.

However, these current initiatives apply writeable Linked Data nodes through (authenticated and authorized) PUT, POST, and DELETE requests on the same RDF-based HTTP resources available for GET today. This follows the Create-Read-Update-Delete (CRUD) operational model, deemed simple and effective for the majority of applications [5]. In practice, such **a symmetric read-write approach conflicts with the reality of a writeable Linked Data node**.

Whilst applying writeable Linked Data nodes in various R&D projects in domains such as Circular Economy (in the Horizon Europe Onto-DESIDE project), HR (in the imec.icon SHARCS project), and health (in the imec.icon PACSOI project)[2], we encountered the following exemplary problems.

- In the Onto-DESIDE project, a product's material composition is **shared in a different granularity than how it is stored**: only whether certain chemical substances exceed defined thresholds must be shared, not the exact quantitative concentrations[3]. This currently requires a parallel process to continuously keep the derived data summary up to date with the product details.

ISWC 2025 Companion Volume, November 2–6, 2025, Nara, Japan

ben.demeester@ugent.be (B. De Meester); julianandres.rojasmelendez@ugent.be (J. A. Rojas); femke.ongenae@ugent.be (F. Ongenae); pieter.colpaert@ugent.be (P. Colpaert); ruben.verborgh@ugent.be (R. Verborgh)
https://ben.de-meester.org/#me (B. De Meester)
0000-0003-0248-0987 (B. De Meester); 0000-0002-6645-1264 (J. A. Rojas); 0000-0003-2529-5477 (F. Ongenae); 0000-0001-6917-2167 (P. Colpaert); https://orcid.org/0000-0002-8596-222X (R. Verborgh)

© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

[1] https://www.w3.org/groups/wg/lws/

[2] Onto-DESIDE: https://ontodeside.eu/, SHARCS: https://www.imec-int.com/en/research-portfolio/sharcs, PACSOI: https://www.imec-int.com/en/research-portfolio/pacsoi.

[3] As per the Registration, Evaluation, Authorisation and Restriction of Chemicals (REACH) regulation: https://echa.europa.eu/regulations/reach/understanding-reach.

CEUR-WS.org/Vol-4085/paper37.pdf

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

- In the SHARCS project, reading diploma credentials written **using the European Learning Model (ELM) v3.2 requires a substantial processing step by every new client application that uses ELM v3.3**[4], which severely limited the reusability of the published Linked Data, even though Semantic standards were meticulously followed.
- In the PACSOI project, **the source of a data write** such as a weight measurement – whether obtained from a calibrated sensor or entered manually – **influences its suitability** when being read for diagnostic purposes.

In this paper, we examine how certain characteristics of writeable Linked Data nodes give rise to specific requirements (Section 2), which conflict with the symmetric read-write approach. We further propose an alternative operational model (Section 3) named *Trustflows*, by applying Command Query Responsibility Segregation (CQRS) and Event Sourcing to Linked Data, thus enabling a **write-to-read Web of Data**: a transparent pipeline from data writes with an explicit trust context mapped to evolvable data reads. We validate this approach by applying it to the PACSOI project's use cases.

2. Requirements

R1: Linked Data publishing stems from the need for standardizing data interpretation, which implies the requirements of coping with longevity and decentralization. The assumption is that adding the complexity of Linked Data publishing (compared to regular data publishing, e.g., via JSON:API [6]) is justified because the published data is deemed of a sufficient value (i) to a larger ecosystem of actors, (ii) for a longer period of time, or (iii) both. However, as the ecosystem evolves, data reuse is hindered: the data models needed to read data diverge from the data model used to originally write the data. Applications, that might not have been envisioned when the data was originally published, require different data models, e.g., at different granularities (cfr. Onto-DESIDE's examplar). A writable Linked Data node thus adds the challenge that during a data write, it is impossible to know how the data will be used by all the possible actors and across time and domains, meanwhile, CRUD's read-write symmetry implies that how the data is written directly affects how the data can be read.

R2: Linked Data allows to more easily integrate data from different domains, which implies the requirement of handling data using different models. Data reuse is hindered when data models evolve. Applications' data interpretation requirements evolve, which affects the needed data models (cfr. SHARCS's exemplar). CRUD requires each client to individually (re-)interpret (old or differently modelled) data into its own context, which introduces overhead, redundant processing, and diverging interpretations of the same data. Alternatively, a more flexible writeable Linked Data node could perform commonly needed data processing tasks, increasing efficiency and providing common alternative data interpretations. If we take the SHARCS exemplar, a mapping from ELM v3.2 to ELM v3.3 could be provided by the writeable Linked Data node, reusable by all new client applications without impeding the functionalities of the existing client applications.

R3: Using writable Linked Data nodes invalidates the authoritative source assumption, which implies the requirement of an explicit trust context. Typical Linked Data nodes act as an *authoritative source*, i.e., the actor governing the node is assumed to reassure the published data is accurate. Once multiple actors are allowed to write data – even with security measures such as a trusted registry in place – the implicit trust context of a single authority is no longer valid. Each read's individual data point must be verified separately, which introduces a large processing overhead. Alternatively, an explicit trust context allows clients to determine which data came from which writing actor. This allows moving the verification from individual data points to writers, which improves efficiency. CRUD provides no guidance for providing such an explicit trust context. If we take the

[4]Changes, among others, included the change of the ontology base uri: https://op.europa.eu/en/web/eu-vocabularies/dataset/-/resource?uri=http://publications.europa.eu/resource/dataset/snb-model.

PACSOI exemplar, not only the data values (i.e., the weight measurements) are important, but also the type of event they originated from (e.g., were they measured via a scale, manually entered, or calculated), and from which actor they originated (e.g., from a calibrated scale, from a healthcare provider, or via self-assessment).

3. Approach

Foremost, we argue that fit-for-purpose authentication and authorization must be put in place: a complementary task that is required for any data exchange use case.

To cover **R1** and **R2**, we **decouple the write interface from the read interface**, at the cost of increased complexity. A mature operational model is Command Query Responsibility Segregation (CQRS) [7]: write operations, or *commands* to update data, are separated from read operations, or *queries* to retrieve data. Within the context of Linked Data publishing, we can take advantage of mature semantic mapping methods to make the mappings between written data and read data explicit. On the one hand, alternative mappings and queries can cater to different data granularities which increases efficiency and privacy, e.g., by providing both all product details and REACH compliance statements in the Onto-DESIDE use case. On the other hand, alternative mappings and queries can cater to different data models and different data model versions which increases the longevity of the once written data, e.g., by providing an alternative ELM v3.3 mapping in parallel in the SHARCS use case.

To cover **R3**, we make **the trust context explicit for every data write**. For this, we can apply the Event Sourcing design pattern [8]. In typical Event Sourcing Systems (ESS), one of the main challenges is that *implicit event schema complicates handling event schema evolution* [9]. This challenge is partially tackled in our Linked Data publishing context; on the one hand this means that the schema is always explicit (which aligns with the advice to create a versioned event schema [9]), on the other hand this means that extending an existing schema is supported by design (which aligns with the advice to create a weak event schema [9]). Mature models such as PROV-O's Actor-Entity-Activity data model [10] can be used to cover the data values, originating event type, and writing actor.

4. Use Case: PACSOI

The imec.icon PACSOI project (2024–2026)[5] aims to leverage Solid-based decentralized "pod" (i.e., an implementation of a writeable Linked Data node) and Linked Data technology to give patients control over their health data while enabling scalable, privacy-preserving federated analytics, initially demonstrated in a proof-of-concept (PoC) focused on follow-up of patients that underwent bariatric surgery (Figure 1). The following use case is investigated.

> As a patient, I write all my weight measurements (done by healthcare providers, through a ByteFlies smart scale, and via a questionnaire in the MoveUP application) to my personal data storage, governed by the FAQIR Foundation. My healthcare providers can use these measurements to check my progress, and simultaneously, my measurements are aggregated into cohorts to be used in research, e.g., by pharmaceutical players.

Our PoC for this use case uses data from 12 sources (i.e., smart scales, questionnaires, and physiological measurements), provided by medical sensors, for 200 patients.

The authentication and authorization information during a write operation captures the Actor part of the Actor-Entity-Activity model. Mature authentication specifications (e.g., OpenID Connect [11]) and mature authorization specifications (e.g. OAuth 2.0 [12]) provide some initial metadata as part of the write operation, however, specific metadata requirements still need to be further analysed.

The write operation's body data – adhering to the Event Sourcing design pattern – captures the Entity-Activity part of the model. We explicitly store the trust context linked with the data events (i.e.,

[5]https://www.imec-int.com/en/research-portfolio/pacsoi

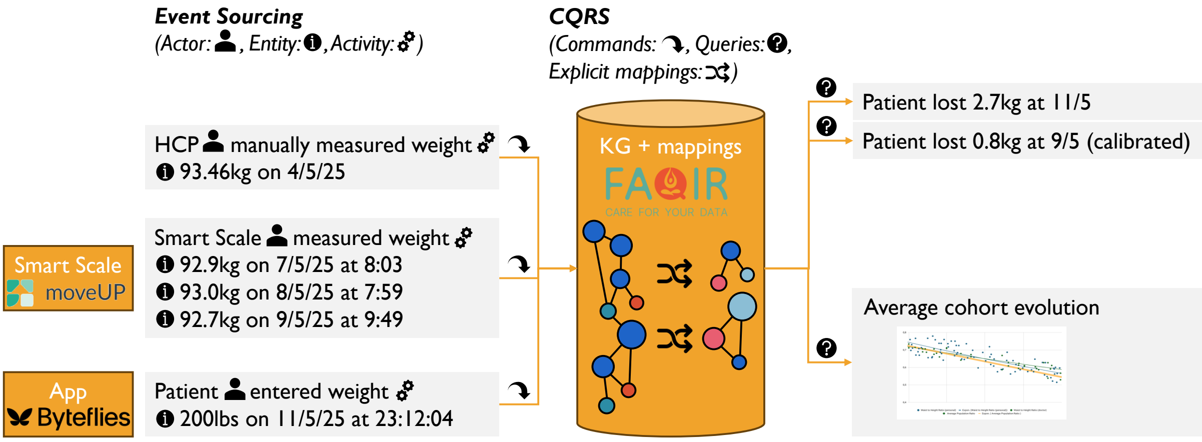

Figure 1: Example application of the Trustflows operational model to the PACSOI use case: data coming in different granularities (**R1**), from different sources (**R3**), are stored using the event sourcing model, and applying explicit mappings and queries (adhering to CQRS) makes sure different reading requirements (**R2**) are met.

the entire Actor-Entity-Activity model) in an event log. For storing such enriched data events, we can apply approaches such as Linked Data Event Streams [13, 14]. Managing these enrichments in an event log thus adheres to the interpretation of a pod as a "hybrid, contextualized knowledge graph" [15].

Mappings (e.g., through SPARQL CONSTRUCT queries, potentially enhanced with ontology alignment mappings, entailment reasoning, or other established Linked Data mapping techniques) allow customizing the reads based on performance requirements (e.g., provide a "last thirty days" overview per patient), data model requirements (e.g., provide a long-term trend across patients), and trust requirements (e.g., only include measurements taken by a calibrated smart scale).

5. Conclusion

This paper argues that the demands of real-world read-write data processes have different underlying assumptions and requirements than those currently provided by read-write Linked Data systems. From an analysis of concrete demands from use cases, we propose a more fitting operational model that leverages CQRS with explicit semantic mappings and Event Sourcing. We performed an initial validation of our Trustflows operational model within the PACSOI project, showing how the Trustflows operational model can handle data coming in different granularities, from different sources, and adhering to different reading requirements. This work is an initial presentation, and will benefit from discussion and feedback from the broader community. By further investigating common pitfalls of deploying writeable Linked Data nodes, we will research a more concrete specification and accompanying implementation that aligns with the current state of the art, brings support for our aforementioned requirements, and avoids the broader set of pitfalls.

Acknowledgments

The described research activities were supported by SolidLab Vlaanderen (Flemish Government, EWI and RRF project VV023/10), and by the imec.icon project PACSOI (HBC.2023.0752), which is co-financed by imec and VLAIO and brings together the following partners: FAQIR Foundation, FAQIR Institute, MoveUP, Byteflies, AContrario, and Ghent University – IDLab. We thank all partners in the following projects, as their contributions helped to shape the requirements: imec.icon SHARCS and Horizon Europe Onto-DESIDE.

Declaration on Generative AI

During the preparation of this work, the author(s) used GPT-4o to: Grammar and spelling check. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] T. Berners-Lee, K. O'Hara, The read–write Linked Data Web, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 371 (2013) 20120513. doi:10.1098/rsta.2012.0513.

[2] 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/.

[3] S. Capadisli, T. Berners-Lee, K. Kjernsmo, Solid Protocol, Draft Community Group Report, Solid Community Group, 2024. URL: https://solidproject.org/TR/protocol.

[4] B. Armintor, E. Cowles, D. Lamb, S. Warner, W. Andrew, Fedora API Specification 1.0, Technical Report, 2021.

[5] M. Fowler, Patterns of Enterprise Application Architecture, The Addison-Wesley Signature Series, 1st ed., Addison-Wesley, Boston, 2013.

[6] Y. Katz, D. Gebhardt, G. Sullice, J. Hanschke, JSON:API v1.1, Technical Report, 2022. URL: https://jsonapi.org/format/.

[7] G. Young, CQRS Documents, Technical Report, 2010. URL: https://cqrs.wordpress.com/wp-content/uploads/2010/11/cqrs_documents.pdf.

[8] M. Fowler, Event Sourcing, Technical Report, 2005. URL: https://martinfowler.com/eaaDev/EventSourcing.html.

[9] M. Overeem, M. Spoor, S. Jansen, S. Brinkkemper, An empirical characterization of event sourced systems and their schema evolution — lessons from industry, Journal of Systems and Software 178 (2021) 110970. doi:10.1016/j.jss.2021.110970.

[10] P. Missier, L. Moreau, PROV-DM: The PROV Data Model, W3C Recommendation, Provenance Working Group, 2013. URL: https://www.w3.org/TR/prov-dm/.

[11] N. Sakimura, et al., OpenID Connect Core 1.0 incorporating errata set 1, Technical Report, The OpenID Foundation, 2014. URL: https://openid.net/specs/openid-connect-core-1_0.html.

[12] D. Hardt, The OAuth 2.0 Authorization Framework, Technical Report, Internet Engineering Task Force (IETF), 2012. URL: https://datatracker.ietf.org/doc/rfc6749/.

[13] W. Slabbinck, R. Dedecker, S. Vasireddy, R. Verborgh, P. Colpaert, Linked Data Event Streams in Solid LDP containers, in: Proceedings of the 8th Workshop on Managing the Evolution and Preservation of the Data Web (MEPDaW) co-located with the 21st International Semantic Web Conference (ISWC 2022), volume 3339, 2022, pp. 28–35.

[14] D. Van Lancker, P. Colpaert, H. Delva, B. Van de Vyvere, J. Rojas Meléndez, R. Dedecker, P. Michiels, R. Buyle, A. De Craene, R. Verborgh, Publishing Base Registries as Linked Data Event Streams, in: Proceedings of the 21th International Conference on Web Engineering, volume 12840 of *Lecture Notes in Computer Science*, Springer, Cham, 2021, pp. 28–36. doi:10.1007/978-3-030-74296-6_3.

[15] R. Dedecker, W. Slabbinck, J. Wright, P. Hochstenbach, P. Colpaert, R. Verborgh, What's in a Pod? A Knowledge Graph Interpretation For The Solid Ecosystem, in: QuWeDa 2022: 6th Workshop on Storing, Querying and Benchmarking Knowledge Graphs, volume 3279, 2022, pp. 81–96.