

Putting the Athlete in Control of their Performance Data through Privacy-Preserving Aggregators across Data Vaults^{*}

Maarten Vandenbrande¹, Pieter Bonte² and Femke Ongenae¹

¹IDLab, Ghent University - imec, Belgium

²Departement of Computer Science, KU Leuven Kulak, Belgium

Abstract

Athlete monitoring requires heterogeneous data from coaches, medical staff, psychologists, and activity-tracking devices amongst others. However, current systems are often centralized, limiting athlete data control, fine-grained access management, and data portability. This demo presents a semantic decentralized architecture in which athlete data remains in personal data vaults under athlete control. An Aggregator component is introduced that allows to efficiently compute reusable & interoperable performance metrics from raw activity data and athlete settings. Using a modified version of the performance tracking application Elevate, we show how this set-up can support privacy-preserving, application-independent computation for athlete-centric performance monitoring.

Keywords

Athlete performance monitoring, Personal Data Vault, Aggregator, Privacy-preserving

1. Introduction

Athlete monitoring is essential for supporting athlete health, training adaptation, and peak performance [1]. Such monitoring requires data from multiple domains, including performance, sleep, health, and wellness [1]. It also involves a wide range of practitioners, such as medical staff, physiologists, coaches, strength and conditioning coaches, performance analysts, nutritionists, and psychologists [1]. Currently, athlete monitoring is commonly performed using centralized athlete-monitoring systems, where practitioners collect, store, analyse, and disseminate performance, health, and wellness data [1]. However, these centralized approaches provide limited support for athlete-centric data control, fine-grained permissions, and reusable access policies across organisations. As a result, athletes have limited control over who can access specific parts of their data. Moreover, when an athlete changes teams, organisations, or practitioners, their historical data is not easily transferred, while the previous team may no longer be allowed to retain access to the athlete's performance statistics.

A solution to address these limitations is to store athlete data in decentralized athlete-centric personal data vaults like Solid¹. This enables, different parties to add data to the athlete's data vaults, while authorized applications and practitioners can access this data to derive insights. This requires fine-grained access policies that allow athletes to share only the specific data that each practitioner requires or with a particular team. Finally, by representing the data semantically, information from different sources becomes interoperable and easier to link across domains. For example, performance data can be connected to sleep, dietary, wellness, or training-plan data, enabling richer analysis and more contextual interpretation. Such semantic descriptions can also provide structured context for AI systems,

Posters, Demos, Blue Sky, and Tutorials at SEMANTiCS 2026, Sep 2026, Ghent, Belgium

^{*}You can use this document as the template for preparing your publication. We recommend using the latest version of the ceurart style.

^{*}Corresponding author.

✉ maarten.vandenbrande@ugent.be (M. Vandenbrande); pieter.bonte@kuleuven.be (P. Bonte); femke.ongenae@ugent.be (F. Ongenae)

ORCID 0009-0003-6470-9123 (M. Vandenbrande); 0000-0002-8931-8343 (P. Bonte); 0000-0003-2529-5477 (F. Ongenae)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

¹Solid: <https://solidproject.org>

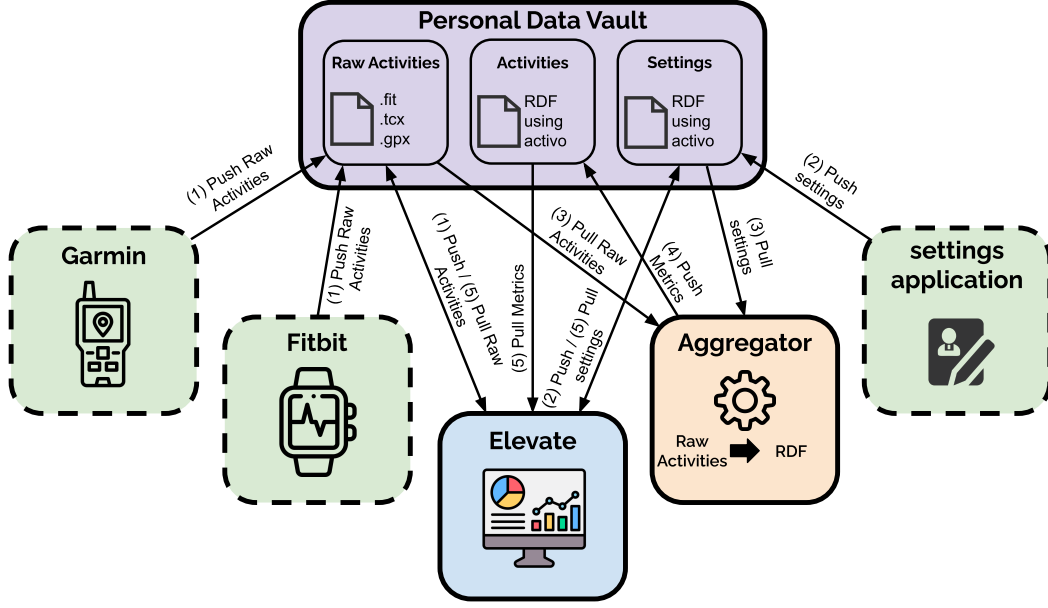


Figure 1: An architectural overview of the system.

supporting knowledge-graph-based retrieval approaches that reduce reliance on latent model knowledge and help mitigate hallucinations [2].

We demonstrate this approach through a decentralized ecosystem that includes a modified version of Elevate, a performance tracking application for athletes (<https://github.com/SolidLabResearch/elevate>). This application extends the open-source Elevate project ² by replacing its data layer with a personal data vault and Linked Data. Importantly, in the decentralized architecture presented here, we include our Aggregator technology. The Aggregator is a separate component in the decentralized architecture that in this demo is set up to derive activity metrics from raw activity files and athlete settings. The Aggregator allows moving reusable metric computation to an external service that can react to newly uploaded data and write the resulting metrics back to the vault. As a result, applications such as Elevate do not need to recompute all metrics themselves on startup, and other authorized applications can reuse the same derived performance data. The demo shows how activities, raw files, derived metrics, and athlete performance parameters can be stored in an athlete-centric data vault and reused by different applications. While this work focuses on performance tracking, the same strategy can be extended to additional applications, for example, health, dietary, or wellness data.

In this work, we present the architecture of the personal data vault backed Elevate application, the data model used to describe activities and performance metrics, and the resulting demo. The main contributions of this paper are: (1) a performant and privacy-preserving decentralized architecture with an aggregator that enables athlete-centric performance monitoring. (2) an ontology (*activo*) for semantically describing sports activities and performance metrics (<https://www.w3id.org/activity-ontology/>).

2. Architecture

This section describes the decentralized architecture with the Aggregator and shows how the different components interact to support the athlete performance monitoring. Figure 1 gives an overview of the system architecture without any authorization (authorization will be addressed later) which consists of four types of components: data producers, personal data vaults, aggregators, and data consumers.

Data originates from the data producers, shown as the green blocks with dotted lines in Figure 1. These producers include activity tracking devices, such as Garmin or Fitbit, as well as a management

²Original Elevate github repo: <https://github.com/thomaschampagne/elevate>

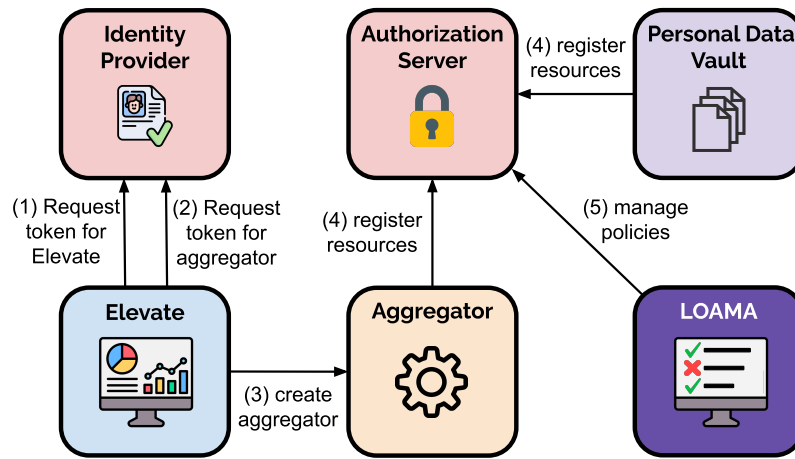


Figure 2: An overview of the authentication and authorization system.

application for athlete performance information. Activity-tracking devices commonly export data using file formats such as FIT, TCX, and GPX. The Elevate application also allows exporting activities or manually creating them. These raw activities are individually stored on the Personal Data Vault (1 in Figure 1). The athlete performance information includes parameters such as average and resting heart rate, functional threshold power, weight, and lactate threshold. This information is created by the Elevate but could also be managed by a separate application who then stores it in a separate resource using the activo ontology in the Personal Data Vault (2 in Figure 1).

The Aggregator is responsible for computing derived performance metrics from the athlete settings and raw activity data. It reuses the original metric-computation logic from Elevate, where this processing was previously performed locally in the application. In this architecture, the computation is moved to a separate Aggregator component. The aggregator can compute the metrics immediately when a new raw activity is uploaded, instead of when opening the application, this greatly improves the user experience. The aggregator also allows different applications to share the same computation. The Aggregator subscribes to changes in the personal data vault, allowing it to detect when new raw activities (FIT, TCX, and GPX files) are added or when athlete settings are updated (3 in Figure 1). After calculating the metrics, the Aggregator writes the resulting statistics back to the personal data vault using the activo ontology (4 in Figure 1). The Aggregator follows the Aggregator specification³, which defines how to create and manage an Aggregator.

The final component is the Elevate application, which acts as a data consumer. It fetches athlete settings, and computed metrics (5 in Figure 1) from the personal data vault, parses them, and indexes them locally to support SPARQL query execution. In the Elevate application, SPARQL queries are generated automatically by a translation layer that maps the internal application model to the semantic data model. For example, the activities screen generates queries to retrieve the activities that match the selected filters, such as sport type, date range, or duration. If a user wants to sort swimming activities from the last year by duration, the application generates a SPARQL query for that selection and executes it locally using Oxigraph [3]. When a specific activity is selected, the application fetches all data related to that activity from the personal data vault, including the original raw activity file. This raw data is then used to render time-series graphs and to display GPS information on a map. The application uses Incremunica [4] to remain synchronized with the personal data vault. Incremunica can discover and decide what notification method to use like polling, webhooks, websockets, or application controlled. When changes occur, Incremunica determines how the local index should be updated, allowing the Oxigraph-backed query layer to remain up to date.

Figure 2 depicts the authentication and authorization architecture that extends the system architecture in Figure 1. In this demo, authentication is handled using OpenID Connect (OIDC), while authorization

³Aggregator spec: <https://www.w3id.org/aggregator/>

is handled using user-managed access (UMA), as defined by the Authorization for Data Spaces (A4DS) specification⁴. A4DS defines the main components involved in this process: the Identity Provider, Resource Server, Authorization Server, Client, and User. The Identity Provider uses OIDC to provide proofs of identity, the Resource Server hosts the protected resources, and the Authorization Server manages access using UMA. A Client, such as the Elevate application, requests access to protected resources, while the User manages the policies that determine which access requests are allowed. During login, Elevate follows an OIDC flow to obtain identity tokens from the identity provider (1 in Figure 2). These tokens can be used as claims when requesting access from the Authorization Server. After login, Elevate initiates a second OIDC-based flow according to the Aggregator specification (2 in Figure 2). This flow allows the Aggregator to obtain user-delegated tokens (3 in Figure 2), enabling it to request resources and write computed metrics on behalf of the user. During this process, Elevate also communicates which Authorization Server should be used. This ensures that the user can define access policies on the Authorization Server for both the resources stored in the Personal Data Vault and the Aggregator itself (4 in Figure 2). Policies are formally expressed in the open digital rights language (ODRL) on the Authorization Server. In this demo, these policies express basic read, write, create, update, and delete permissions. However, ODRL also supports more expressive policy constraints, allowing the access-control model to be extended in future work. The ODRL policies are managed using the LOAMA [5] application (5 in Figure 2). If Elevate does not have access to a required resource, it can send an access request, which is then shown in LOAMA, so the user can decide whether or not to grant access. This makes the system more practical for end users, because they do not have to configure every permission in advance. Instead, access can be requested when it is needed, and the user can approve or reject it directly in LOAMA.

3. Data Model

Semantically describing performance data makes it possible to connect information that would otherwise remain isolated. For example, an athlete’s wellness report can be linked to the activity it refers to, dietary changes or sleep metrics can be connected to corresponding performance outputs such as peak power, and training plans can be linked to the statistics produced during the planned activities. This creates a connected view of the athlete’s performance context, instead of a set of separate, application-specific datasets. Existing prominent sport-related ontologies^{5,6,7} only partly cover our use case. They focus on sports events, results, participation, news coverage, or activity discovery, rather than on detailed athlete-monitoring data and training-performance metrics. Similar limitations are also reported by Li and Lambrix, who note that existing sports ontologies often focus on statistics, event, participation, or news-related concepts rather than richer domain-level modelling [6].

Therefore, we created a dedicated ontology for modelling sports activities and their associated performance data. The ontology focuses on the performance-monitoring aspect of sport, and is defined using OWL and RDFS. Its namespace is <https://solidlabresearch.github.io/activity-ontology/#>, for which we suggest using the prefix `act:ivo`. Its design is based on domain knowledge gathered from the original open-source Elevate application, which has accumulated a wide range of commonly used activity types and performance metrics through several years of community-driven development. The core idea of the ontology is to model an activity not only as a timed event, but as a rich data object connected to laps, athlete context, devices, applications, and computed statistics. The ontology covers common activity-level information such as activity type, start and end time, distance, duration, and flags, while also supporting detailed performance metrics including speed, pace, power, heart rate, cadence, grade, elevation, scores, zones, peaks, and sport-specific dynamics. A central design goal is interoperability with existing semantic web standards. The ontology reuses PROV-O to describe provenance, such

⁴A4DS spec: <https://spec.knows.idlab.ugent.be/A4DS/L1/latest/>

⁵BBC Sport Ontology: <https://iptc.org/thirdparty/bbc-ontologies/sport.html>

⁶IPTC Sport Schema: <https://iptc.org/standards/sport-schema/>

⁷OpenActive: <https://openactive.io/>

as which device or application generated or computed the data, FOAF for athlete/person modelling, and SKOS for controlled activity type concepts. Local activity types can be linked to OpenActive identifiers, allowing datasets to use domain-specific terms while remaining compatible with broader activity vocabularies. This makes the ontology suitable for representing both raw sports activity data and higher-level analytics in a machine-readable, extensible way.

4. Demonstration

The demonstration shows how different applications can contribute data to the athlete's personal data vault, while the athlete retains fine-grained control over access through the LOAMA application. It also demonstrates the creation of an Aggregator through a second login flow in Elevate, which allows the Aggregator to obtain the delegated authorization required to process data on behalf of the athlete. Raw activities are uploaded using a simple web application that generates synthetic FIT files. Once a new activity is added, the Aggregator detects it, computes the corresponding performance metrics, and writes the resulting semantic descriptions back to the vault. Elevate subsequently retrieves the athlete settings, activities, and derived metrics to populate its performance dashboard, without having to recompute these metrics when the application is opened. Together, these steps demonstrate how the proposed architecture combines athlete-controlled data storage, fine-grained authorization, reusable computation, and semantic interoperability. Derived performance data remains under the athlete's control and can be reused by other authorized applications rather than remaining tied to a single centralized platform. A setup guide and a video of the demonstration are available in the Elevate repository (<https://github.com/SolidLabResearch/elevate>).

Acknowledgments

This work was partly funded by the SolidLab Vlaanderen project (Flemish Government, EWI and RRF project VV023/10). Furthermore, this research was supported by the imec.icon project PACSOI (HBC.2023.0752), which was co-financed by imec and VLAIO and brings together the following partners: FAQIR Foundation, FAQIR Institute, MoveUP, Byteflies, AContrario, and Ghent University – IDLab.

Declaration on Generative AI

During the preparation of this work, the authors used chatGPT-5.5 in order to: Grammar and spelling check. After using this tool, the authors reviewed and edited the content as needed and takes full responsibility for the publication's content.

References

- [1] E. Neupert, L. Gupta, T. Holder, S. A. Jobson, Athlete monitoring practices in elite sport in the united kingdom, *Journal of Sports Sciences* 40 (2022) 1450–1457. URL: <https://www.tandfonline.com/doi/full/10.1080/02640414.2022.2085435>. doi:10.1080/02640414.2022.2085435.
- [2] M. Weyns, T. Blyau, B. Steenwinckel, F. De Turck, S. Van Hoecke, F. Ongenae, Explainable knowledge graph embeddings for industrial process monitoring & control, *Information Fusion* 123 (2025) 103242. doi:10.1016/j.inffus.2025.103242.
- [3] T. Pellissier Tanon, Oxigraph, 2022. URL: <https://oxigraph.org>. doi:10.5281/zenodo.7408022.
- [4] M. Vandenbrande, R. Taelman, P. Bonte, F. Ongenae, Incremunica: Web-based incremental view maintenance for sparql, in: *The Semantic Web*, volume 15719 of *Lecture Notes in Computer Science*, Springer Nature Switzerland, 2025, pp. 297–315. doi:10.1007/978-3-031-94578-6_17.

- [5] W. Slabbinck, L. De Rouck, J. Van Herwegen, W. Termont, B. Esteves, R. Verborgh, Loama: Low-code odrl access management application, in: Proceedings of the International Semantic Web Conference (ISWC), 2025. URL: <https://ceur-ws.org/Vol-4085/paper83.pdf>.
- [6] H. Li, P. Lambrix, Badmintonto: A badminton domain ontology, in: Joint Ontology Workshops (JOWO), 2024. URL: <https://www.diva-portal.org/smash/get/diva2:1860840/FULLTEXT02.pdf>.