LinkedDataHub

A Low-Code Application Platform for RDF Knowledge Graphs

Martynas Jusevičius

AtomGraph

martynas@atomgraph.com

1. Motivation

Knowledge Graphs have become a powerful abstraction for representing structured, linked data across domains including enterprise integration, scientific research, and government transparency. However, turning RDF datasets into usable applications typically requires a custom development effort — resulting in duplicated work, long lead times, and inaccessible tooling for non-experts.

LinkedDataHub was built to address this gap: it is a general-purpose, low-code platform for interacting with RDF data through a web interface. Instead of building a custom stack from scratch, users can install LinkedDataHub and immediately browse, query, and manage linked data — or extend it to power their own Knowledge Graph applications.

2. Architecture

LinkedDataHub's architecture is grounded in this first principle:

Webpage = Transformation(Projection(Dataset))

LinkedDataHub realises this by using SPARQL to project RDF and XSLT 3.0 (Kay, 2017) to transform the result set into XHTML+RDFa (Adida et al., 2008):

Webpage = XSLT(SPARQL(Triplestore))

As a result, LinkedDataHub acts as a thin façade over any standards-compliant SPARQL Graph Store (Ogbuji, 2013) — adding minimal overhead while exposing a powerful frontend UI with authoring capabilities.

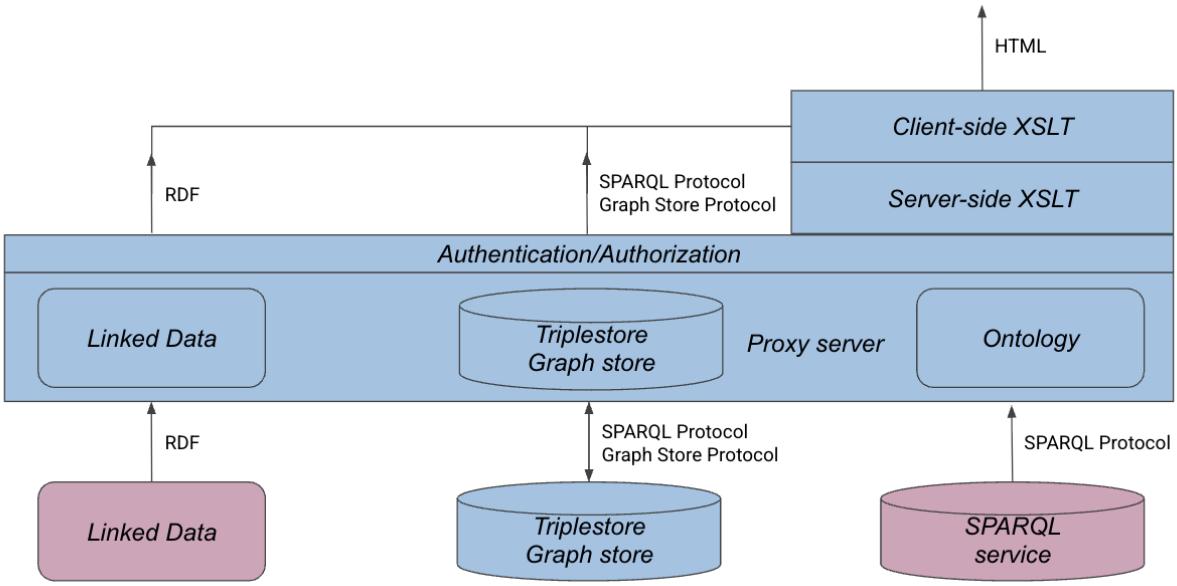

Figure 1 – Runtime stack showing proxy layer, triplestores, XSLT rendering tier, and external data sources. Arrows indicate RDF & SPARQL protocol flows.

2.1 Implementation & Deployment

LinkedDataHub is packaged as a multi-module Maven project (requires Java 17). A single `docker-compose up --build` command launches a production-ready stack consisting of:

Services	Purpose
nginx	SSL termination & reverse proxy
LinkedDataHub	Core application (JAX-RS/Jersey/Tomcat)

Fuseki	Separate SPARQL stores for admin and public datasets, accessed via SPARQL and Graph Store Protocol endpoints
Varnish	HTTP proxy caching layers

Its runtime adds only modest overhead, so system throughput and latency are effectively those of the underlying RDF store.

Backend

An authentication filter validates the caller's identity against WebID (*WebID 1.0*, 2014) profile data, while an authorization filter executes SPARQL queries over W3C Web Access Control (*WAC*, n.d.) authorization data to decide if a request is permitted.

Content negotiation as well as caching headers are supported out of the box. Each document can be serialized as RDF (in any format supported by Apache Jena) or as XHTML+RDFa. This dual representation supports both machine access and human interaction.

Dataset-level versioning and formal provenance tracking are not yet implemented (see Future Work).

LinkedDataHub deliberately does not implement the Linked Data Platform (Speicher et al., 2015) specification. Unlike the SPARQL standards suite, LDP lacks formal semantics and has seen limited real-world adoption.

Frontend

The choice of XSLT as the UI rendering engine is unorthodox, but it is the most robust and well-specified data transformation language, which fits our declarative architecture perfectly and has outlived all JavaScript framework trends.

XSLT 3.0 is used to transform the document's RDF/XML serialization to XHTML; the server- and client-side templates share common templates. The client-side processor in use is SaxonJS

(*SaxonJS*, n.d.); its Interactive XSL extension is also used to handle user events in the web browser, removing the need for any JavaScript in the UI implementation.

Form generation and data validation are driven by SPIN constructors and constraints as well as SHACL shapes. The HTML forms encode RDF in a declarative fashion by effectively using RDF/POST (Egorov & Jusevičius, 2022) as an URL-encoded RDF syntax.

2.2 Dataspace

A *dataspace* in LinkedDataHub is a self-contained unit of content and configuration that can be exported, copied, or hosted independently. Each dataspace bundles four core components:

Component	Purpose
Base URI	Base URI for all document URIs within the dataspace.
SPARQL service	Read-write service (GSP + SPARQL Protocol) for querying and updating the named graphs in the dataspace.
Ontology	Domain vocabulary that defines user-defined and system classes, properties, SPIN/SHACL constructors/constraints/shapes etc.
XSLT stylesheet	Declarative UI layer that transforms SPARQL projections into XHTML+RDFa pages.

Multiple dataspaces can coexist in a single LinkedDataHub instance — each with its own base URI — or be isolated per tenant in the forthcoming cloud offering.

2.3 Data Model

Each LinkedDataHub *document* corresponds to a named graph in the backend triplestore and is part of the document hierarchy. The hierarchy is structured using two document types: *containers* and *items*, modeled using the SIOC vocabulary (*SIOC*, 2018). Containers are akin to folders (can have children documents), and items to files (cannot have children documents) in a filesystem.

Content *blocks* are sub-document level RDF resources of two main types: XHTML and objects. The former are XHTML literals, the latter embed any RDF resource by its URI. Blocks attach to documents using the `rdf:_1`, `rdf:_2`, … sequence properties.

The rest of the data in the documents are treated as generic RDF resources. They can be either of user-defined types or of LinkedDataHub's built-in types from system ontologies, such as query, chart, SPARQL service, import etc.

3. Features

LinkedDataHub includes a broad set of built-in capabilities:

- A query editor for SPARQL queries
- Visualizations of result sets as tables, charts, timelines, maps, and networks
- Graph-native navigation, including faceted search, backlinks and parallax navigation[1]
- Browsing Linked Data resources and cloning them into a local document
- Structured RDF resource editing with autocomplete support
- Vocabulary and ontology authoring
- Drag-and-drop RDF file upload
- CSV-to-RDF mapping driven by user-defined SPARQL queries
- User authentication and access control

4. End-User Experience

LinkedDataHub's default interface lets non-technical users explore and curate RDF data without writing a single line of code. Key UX affordances include:

- **Content Mode** – renders a document as an ordered sequence of blocks (text, images, queries, charts, maps). Block order can be swapped using drag-and-drop.

[1] Parallax is a unique navigation concept enabled by the graph: it allows "jumping" from a SPARQL result set to another result set over a user-selected property. Inspired by Freebase Parallax.

- **Alternate Modes** – Properties, List, Table, Grid, Map, and Graph views expose all RDF resources contained in a named graph, each tuned for a different exploration task.
- **In-place editing** – edit triples directly from the UI, with autocomplete for resources.
- **SPARQL in the browser** – run ad-hoc queries and immediately render results as visual blocks.

Together these modes enable users to browse, query, and assemble knowledge without custom development[2].

5. Developer Experience and Customization

While LinkedDataHub is usable as a standalone data portal, it was also built as a developer platform. Developers can:

- Extend or override default page templates using standard XSLT 3.0 (no proprietary templating).
- Deploy applications via docker-dompose and configure environments declaratively.
- Use the HTTP-based Linked Data API to integrate with other systems.
- Access all management features — uploading graphs, clearing datasets, editing metadata — via a command-line interface (CLI).
- Authenticate via WebID-TLS client certificates or optional OAuth2 (Google) integration. Custom authentication or authorization can be plugged in via standard JAX-RS request filters.

This architecture enables rapid customization without sacrificing transparency or standards compliance.

[2] Watch a screencast demonstrating the UI: LinkedDataHub v5 teaser: structured content (extended version)

6. Use Cases

LinkedDataHub has been successfully deployed in research, enterprise, and public-sector contexts:

- Enterprise data hubs for internal knowledge integration and analytics, e.g., NXP Semiconductors — the first proto-LinkedDataHub deployment and winner of the 2015 European Linked Data Award[3]
- Public open data portals, including the NOI Open Data Hub in collaboration with Ontopic[4], where LinkedDataHub serves as the frontend to a Virtual Knowledge Graph provided via SPARQL endpoint
- Cultural heritage and national library systems, e.g., The Royal Library of Denmark
- Data storytelling using XHTML content, curated views and charts

Hosted multi-tenant cloud deployments are also underway, with a focus on FAIR data and collaborative publishing.

7. Comparison with Related Platforms

	Metaphactory	**LinkedDataHub**	**Wikibase**
Store and publish Linked Data	✔	✔	Using extensions
Edit RDF data	✔	✔	✘
Query with SPARQL	✔	✔	Using extensions

[3] https://2015.semantics.cc/eldc-awards-given
[4] https://ontopic.ai/

Create and import ontologies	✔	✔	✘
Create data-driven content	Using proprietary template language	✔	Using extensions
Customize UI	Using proprietary template language	Using standard W3C stylesheets	Using product-specific code
License	Commercial	Open-source (Apache 2.0)	Open-source (GNU)

LinkedDataHub offers a middle ground: a flexible yet standards-based platform that supports both quick-start and deep customization. It also has the open-source advantage – LinkedDataHub is released under the permissive Apache 2.0 License, giving users full transparency and freedom to modify or self-host the platform.

8. Future Work

8.1 Hosted Multi-Tenant Dataspaces

A forthcoming LinkedDataHub Cloud offering will allow users to spin up personal, read-write dataspaces at their own dedicated URLs, making them ideal for collaborative FAIR data management and consumption. It will lower the barrier for researchers and small organizations that lack DevOps capacity but still need a FAIR-compliant Knowledge Graph portal.

8.2 Dataset Versioning & Provenance

Planned support for snapshotting named graphs and attaching PROV metadata will enable graph-level version control.

8.3 LLM Integration via Model Context Protocol

We are preparing an integration with the Model Context Protocol (MCP) (*Model Context Protocol*, n.d.), which will allow LLM-based agents to control the LinkedDataHub interface programmatically: executing queries, navigating and creating the documents, adding data to them etc..

This will turn LinkedDataHub into a general-purpose rendering engine for LLM assistants working over RDF knowledge bases.

9. Conclusion

LinkedDataHub helps bridge the gap between RDF datasets and real-world applications. By embracing declarative technologies, Linked Data principles, and user-friendly defaults, it empowers both data publishers and consumers to interact meaningfully with semantic data.

Whether you're deploying a research data portal or building a complex knowledge application, LinkedDataHub provides a low-code, standards-based foundation.

Website: https://atomgraph.github.io/LinkedDataHub/

Source code: https://github.com/AtomGraph/LinkedDataHub

References

Adida, B., Birbeck, M., McCarron, S., & Pemberton, S. (2008, October 14). *RDFa in XHTML: Syntax and Processing*. W3C. https://www.w3.org/TR/rdfa-syntax/

Egorov, S., & Jusevičius, M. (2022, August 29). *RDF/POST Encoding for RDF*. https://atomgraph.github.io/RDF-POST/

Kay, M. (2017, June 8). *XSL Transformations (XSLT) Version 3.0*. W3C. https://www.w3.org/TR/xslt-30/

metaphactory. (n.d.). metaphacts. https://metaphacts.com/metaphactory

Model Context Protocol. (n.d.). Model Context Protocol. https://modelcontextprotocol.io/

Ogbuji, C. (2013, March 21). *SPARQL 1.1 Graph Store HTTP Protocol*. W3C. https://www.w3.org/TR/sparql11-http-rdf-update/

SaxonJS. (n.d.). Saxonica. https://www.saxonica.com/saxonjs/

SIOC Core Ontology Specification. (2018, February 28). http://rdfs.org/sioc/spec/

Speicher, S., Arwe, J., & Malhotra, A. (2015, February 26). *Linked Data Platform 1.0*. W3C. https://www.w3.org/TR/ldp/

WebAccessControl. (n.d.). W3C. https://www.w3.org/wiki/WebAccessControl

WebID 1.0. (2014, March 05). https://www.w3.org/2005/Incubator/webid/spec/identity/

Wikibase. (n.d.). Wikimedia Deutschland. https://wikiba.se/