From fragmentation to federation: bridging heterogeneous Social Sciences and Humanities knowledge graphs through query mediation in GRAPHIA

Julien Homo[1,*], Kévin Darty[1], Luca De Santis[2], Silvio Peroni[3], Matteo Romanello[4], Tomasz Hoffmann[5], Arkadiusz Margraf[5], Sy Holsinger[6], Maxim Kupreyev[6], Ursula Rabar[6], Florian Lemoine[7], Yann Le Franc[7], Jan Wiebelitz[8] and Benjamin Zapilko[9]

[1]*Foxcub, France*
[2]*Net7 Srl, Italy*
[3]*University of Bologna, Italy*
[4]*Odoma Sàrl, Switzerland*
[5]*Poznańskie Centrum Superkomputerowo-Sieciowe (PCSS), Poland*
[6]*OPERAS, Belgium*
[7]*e-Science Data Factory, France*
[8]*Leibniz Universität Hannover, Germany*
[9]*GESIS – Leibniz Institute for the Social Sciences, Germany*

Abstract

Federating heterogeneous knowledge graphs that are independently governed and structurally diverse remains a core data management challenge. We report on GRAPHIA, a Horizon Europe project involving 20 partners, which is building a federation of Social Sciences and Humanities (SSH) knowledge graphs, including GoTriple, OpenCitations, the GESIS Knowledge Graph, and the Open Research Knowledge Graph (ORKG), using the Scientific Knowledge Graphs Interoperability Framework (SKG-IF) as its interoperability layer. Applying SKG-IF in practice revealed that vocabulary alignment alone is insufficient for query federation: structural differences between models based on the Functional Requirements for Bibliographic Records (FRBR) and flat data models make direct SPARQL federation across heterogeneous endpoints fragile. This led us to a "fetch then query locally" architecture that leverages the SKG-IF REST API as a structurally neutral access layer, combining REST-based data fetching orchestrated through the Model Context Protocol (MCP) with natural language to SPARQL translation over a QLever-backed consolidated graph. We present a systematic evaluation of RDF backends (Virtuoso, QLever, GraphDB), including a benchmark on a 155-million-triple dataset that led to the adoption of QLever, and analyse cross-graph query patterns derived from 70 competency questions collected with SSH domain experts, mapping which parts of the federation problem SKG-IF solves today and which require principled extensions.

Keywords

federated knowledge graphs, SKG-IF, federated querying, SPARQL, interoperability, QLever, Model Context Protocol, scholarly data

1. Introduction

A historian studying the circulation of ideas across European archives needs to find which publications cite a specific archival collection, who authored them, and what topics they cover. A social scientist wants to identify SSH datasets that have been reused across disciplines and trace their citation patterns. None of these questions can be answered today from a single platform. The SSH research landscape is supported by a growing ecosystem of research information sources, each covering a different facet: GoTriple [1] indexes over 20 million documents; OpenCitations [2] maintains nearly 2 billion citation links; the GESIS Knowledge Graph [3] covers social science datasets and publications; the ORKG [4] structures research contributions; the European Holocaust Research Infrastructure (EHRI) [5] integrates Holocaust-related archival descriptions; E-RIHS DIGILAB provides infrastructure for Heritage

DMKG'26: 2nd International Workshop on Data Management for Knowledge Graphs, October 2026, Bari, Italy
*Corresponding author.
EMAIL: julien.homo@foxcub.fr (J. Homo)
© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

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

CEUR-WS.org/Vol-4322/paper_20.pdf

Science data. These systems are developed independently, with different data models, identifier conventions, and levels of technical maturity.

The Semantic Web community has long argued that shared ontologies and linked data standards can bridge such fragmentation. The adoption of these technologies has also been reinforced by recommendations such as the European Open Science Cloud (EOSC) Interoperability Framework [6]. But what happens when you actually try to federate real, heterogeneous, independently governed knowledge graphs, each with its own data model, identifier conventions, and infrastructure? This paper reports on that experience and explains how the fragmentation mentioned above is overcome through the use of the Scientific Knowledge Graph - Interoperability Framework (SKG-IF) [7].

GRAPHIA (Knowledge Graphs, AI Services, and Next Generation Instrumentation for R&D in Social Sciences and Humanities) is a Horizon Europe project building a federated SSH knowledge graph infrastructure. The project involves 20 partners and adopts a federation model in which participating infrastructures remain autonomous and are interconnected through SKG-IF (Section 3.2).

Our experience over the first 18 months has yielded both encouraging results and practical lessons. Applying SKG-IF in practice revealed that direct SPARQL federation across structurally heterogeneous endpoints requires adaptation when the federated graphs use fundamentally different property path structures. This led us to develop an architecture that leverages the SKG-IF REST API as the primary data access path and resolves queries locally against a consolidated graph. The current implementation is a deliberate two-source proof of concept (GoTriple and OpenCitations, the two sources with operational SKG-IF APIs), validating the pattern end-to-end on a controlled case before scaling to the other aligned sources.

Our contributions are fourfold: (a) we show how SKG-IF enables federation across real SSH knowledge graphs and report on the practical adaptations its implementation required; (b) we describe the Federated Gateway architecture and how it evolved from SPARQL-first federation to a "fetch then query locally" pattern using the Model Context Protocol (MCP); (c) we present a systematic evaluation of RDF backends that led to the adoption of QLever [8] and the design of a Graph-as-a-Service deployment model; (d) we analyse 70 competency questions collected with SSH domain experts and show how their cross-graph query patterns validated the architecture.

2. Related works

2.1. Federated querying over knowledge graphs

Federated SPARQL query processing has been studied extensively. Systems like FedX [9], Semagrow [10], and SPLENDID [11] decompose SPARQL queries and distribute sub-queries to multiple endpoints based on source selection heuristics. The SPARQL 1.1 Federated Query extension [12] provides a W3C-standard mechanism through the SERVICE keyword, though it requires clients to know which endpoints hold relevant data.

In life sciences, Bio2RDF [13] normalised heterogeneous datasets into a common RDF representation behind a unified SPARQL endpoint. While effective for read-only workloads, this requires centralised ingestion with attendant governance and freshness challenges. Wikidata's Query Federation Service [14] combines a massive centralized graph with federated links to external endpoints. Its scale has exposed practical limits of SPARQL federation: timeout sensitivity, variable endpoint reliability, and schema mismatch difficulties.

In the scholarly domain, the OpenAIRE Research Graph [15] aggregates metadata into a centralised graph of over 300 million objects. In SSH, however, data models range from the Functional Requirements for Bibliographic Records (FRBR) [16] to Records in Contexts (RiC) [17] and the CIDOC Conceptual Reference Model (CIDOC-CRM) [18], reflecting substantial diversity in structures, semantics, and a fragmented landscape of partially overlapping standards. This requires a more flexible integration strategy. More recently, Emonet et al. [19] presented LLM-based SPARQL query generation over federated knowledge graphs at ISWC 2024. This directly informs the SPARQL Translator component of our Gateway (Section 4).

2.2. The SSH research information landscape

The GRAPHIA federation includes sources that vary considerably in coverage and data models: GoTriple [1] (20M+ documents, flat TRIPLE ontology [20]), OpenCitations [2] (2.4B+ citation links, based on the SPAR Ontologies [21]), the GESIS KG [3] (social science metadata, Schema.org [22] and DDI [23]), and the ORKG [4] (structured research contributions). EHRI [5] and E-RIHS DIGILAB manage primary resources using RiC [17] and CIDOC-CRM [18] respectively, extending beyond SKG-IF's current publication-centric model; their integration is a perspective discussed in Section 6.

3. Users and requirements identification

3.1. Users involved

The design is grounded in a formal user research process [24]: thirty interviews with SSH researchers, data providers, and commercial data consumers produced twelve personas and confirmed the need for cross-graph querying without SPARQL expertise. Around 90% of surveyed respondents favoured a natural language interface [25]. These needs were operationalised into seventy competency questions (Section 3.3), which drive both the requirements analysis and the evaluation.

3.2. Semantic interoperability and federation via SKG-IF

The Scientific Knowledge Graphs Interoperability Framework (SKG-IF) [7], initiated by the Research Data Alliance, addresses a fundamental challenge in scholarly communication: enabling discovery and analysis across sources of research information developed independently adopting different data models and technologies. It provides several artefacts: a data model covering six core entity types (Research Product, Agent, Grant, Topic, Venue, Data Source) implemented in the SKG-IF Ontology (SKG-O) based on existing semantic artefacts such as the Semantic Publishing And Referencing (SPAR) Ontologies [21], the Provenance Ontology (PROV-O) [26], Dublin Core [27], the Data Catalog Vocabulary [28], and Schema.org; the specification of a JSON-LD format compliant with the SKG-IF data model, aligned via an appropriate JSON-LD context document; a SHACL document to perform semantic validation; and an Open REST API specification exposing endpoints for the core entity types (i.e. /products, /persons, /organisations, /grants, /topics, /venues, /datasources), where the correspondence is not one-to-one: Agent is served by both /persons and /organisations.

SKG-O follows the FRBR [16] modelling pattern: research products are described according to layers depicting their conceptual essence (FRBR Work), content (FRBR Expression), and format (FRBR Manifestation), each connected to their descriptive properties (topics, access conditions, dates, etc.). In addition, since SKG-IF does not cover all possible aspects of the scholarly domain, it provides a formal mechanism to extend its coverage, in both the data model and the API, to address specific needs not explicitly defined in the core.

The key design decision in SKG-IF is that the sources providing research information, which can be expressed as knowledge graphs, stored in relational databases, etc., do not need to change their internal data model and technologies, but rather implement an SKG-IF Open REST API endpoint that maps the internal data model into SKG-IF. This preserves local autonomy and avoids costly redesign of established systems. This non-invasive approach is essential for adoption in a context where research information sources are maintained by independent organisations with their own governance. The specification deliberately does not prescribe how a source produces this representation: a source backed by a relational database may use a mapping language such as R2RML, a query-rewriting layer, or a hand-written export. The framework constrains the shape of the response, not the means of producing it, and GRAPHIA implements none of these on behalf of its members.

For GRAPHIA, SKG-IF was selected as the primary interoperability layer because of this non-invasive design: each participating graph implements the SKG-IF API and describes its capabilities in a machine-readable configuration, without restructuring its data. This reduces what would otherwise be quadratic pairwise alignment to a single alignment per graph, and was decisive for adoption

Table 1
Mappings between GoTriple data model and SKG-O (documented using SSSOM [29]).

GoTriple entity	Mapping	SKG-O entity	Comment
`triple:Document`	`skos:exactMatch`	`fabio:ScholarlyWork`	GoTriple documents contain research literature (articles, books, book parts, conference proceedings, etc.)
`triple:Dataset`	`skos:exactMatch`	`fabio:Dataset`	Labeled Research Data in SKG-O.
`triple:SemanticArtefact`	`skos:closeMatch`	`fabio:Work`	No exact correspondence in SKG-O. Semantic Artefacts are mapped as generic Research Products.
`triple:Multimedia`	`skos:closeMatch`	`fabio:Work`	Multimedia items are mapped as generic Research Products.
`triple:Profile`	`skos:exactMatch`	`foaf:Agent`	`foaf:Agent` is the superclass of both `foaf:Person` and `foaf:Organisation`. GoTriple does not distinguish between the two.
`triple:Project`	`skos:closeMatch`	`frapo:Grant`	GoTriple also contains projects not associated with grants (e.g. citizen science initiatives), so the correspondence cannot be exact.

in a consortium of 20 partners with diverse technical contexts. Several scholarly graphs have already adopted or are adopting SKG-IF, including GoTriple, OpenCitations, and OpenAIRE, but GRAPHIA is among the first projects to use it as the foundation of a multi-graph federation rather than as a secondary export format. SKG-O provides the ontological layer for semantic interoperability, while the SKG-IF API provides the access contract for technical interoperability.

To understand which SKG-IF entities are currently handled by the four sources included in GRAPHIA (GoTriple, OpenCitations, ORKG, and GESIS KG), we created dedicated alignment tables between the sources' data models and SKG-O, using the Simple Standard for Sharing Ontology Mappings (SSSOM) [29]. An example of alignment (for GoTriple) is summarised in Table 1.

The choice of SKOS predicates over `owl:sameAs` is deliberate. Several of these correspondences are knowingly inexact: `triple:Project` is mapped as `skos:closeMatch` to `frapo:Grant` precisely because GoTriple also records unfunded activities such as citizen science initiatives. Asserting `owl:sameAs` would license inferences that are false for part of the population. The mappings are consequently applied at transformation time, by each source when it produces SKG-IF-compliant JSON-LD, rather than resolved by a reasoner at query time. Reasoning appears in Table 2 as a general triplestore criterion, not as a requirement of this architecture: QLever was adopted despite scoring lowest on it.

The GoTriple alignment maps 3/6 SKG-IF entity types (Research Product, Agent, Grant) and it is currently the only source in the federation with a live SKG-IF API implementation, exposing /products and /grants endpoints. OpenCitations covers 3/6 entity types at the ontological level (Research Product, Agent, Venue) and is about to publish a first partial implementation of its SKG-IF API, while the full API will be launched at the end of 2026. GESIS KG and ORKG map primarily to Research Product through multiple entity types, currently exposed via SPARQL endpoints, with SKG-IF APIs planned for the end of 2026.

Joining the federation thus requires only implementing the SKG-IF Open REST API and providing a YAML configuration describing the supported endpoints and capabilities. For sources lacking the infrastructure to deploy their own endpoint, GRAPHIA is designed to offer a hosted onboarding model: the source provides a data dump, and the GRAPHIA team deploys an SKG-IF API endpoint on its behalf.

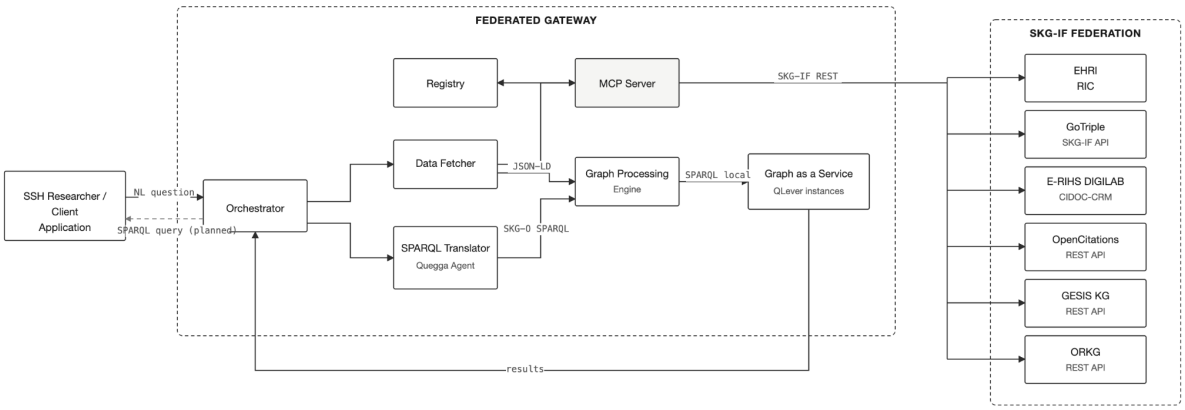


Figure 1: SKG-IF federated architecture: end-to-end flow from a researcher's natural-language question to results, mediated by an orchestrator, a federated gateway, and the SKG-IF endpoints of participating federated Knowledge Graphs.

3.3. Requirements identification through competency questions

Rather than developing a brand new ontology, we adopted SKG-IF and SKG-O as the foundation. The LOT methodology [30] was used to collect competency questions (CQs) that informed what extensions to SKG-O were needed. Seventy CQs were collected from domain experts through workshops, including a session at the Innovation Prototyping Lab in Zagreb in 2025 [31]. Each CQ was designed to leverage content from at least 2 of the 4 initial KGs. The analysis assessed whether these questions could be expressed as SPARQL queries compliant with SKG-O, identifying gaps requiring extensions to the GRAPHIA Ontology [32] or the SKG-IF API.

Of the 70 CQs collected, 10 were identified as top priorities. A detailed analysis revealed a clear gradient of implementability, presented in Section 5.1. This categorisation served a dual purpose: it defined the scope of the initial pilot (starting with what is already aligned and manageable using SKG-O off-the-shelf), and it identified the concrete harmonisation work required for the remaining CQs, directly informing the architectural decisions described in Section 4.

4. Technical architecture for a federated infrastructure

We designed the Federated Gateway to operationalise the federation (Figure 1). The architecture relies on SKG-IF's REST API as the primary data access path, separating data retrieval from query resolution. This also enables sources that are not RDF-native to participate, as long as they implement the SKG-IF API specification.

The Federated Gateway is composed of five main services:

1 Orchestrator. The Orchestrator is the single entry point. It receives a question from the user, expressed in natural language to lower the barrier of entry for SSH researchers who are not familiar with SPARQL, and coordinates the workflow with the other services composing the gateway, i.e. first the SPARQL Translator and then the Data Fetcher.

2 SPARQL Translator. The SPARQL Translator translates the natural language question into an SKG-O-compliant SPARQL query targeting the locally consolidated graph. The architecture also allows expert users to provide an SKG-O compliant SPARQL query directly. The Translator builds on Quagga [33], an agentic system that uses open-weight LLMs and retrieval-augmented generation to produce queries grounded in the target knowledge graph's schema. Quagga is at version 0.1.0 (alpha), tested against GESIS KG, OpenCitations, and EHRI. A community-driven benchmarking initiative is underway, targeting 200 question-query pairs.

3 Data Fetcher. The Data Fetcher retrieves data from the federated sources through their SKG-IF REST API endpoints via a Model Context Protocol (MCP) server, translating, by using an LLM, a query in natural language into a series of invocations of SKG-IF API endpoints. The Data Fetcher selects the appropriate sources by querying the Federation Registry (see below) and leverages the output of the SPARQL Translator to automatically generate the appropriate SKG-IF REST API calls. This explicit, configuration-based endpoint selection contrasts with the heuristic-based source selection used by SPARQL federation engines such as FedX [9]. Notably, source selection and invocation are driven by an LLM that interprets the user's information need and decides at runtime which repositories to query: each repository is exposed as a self-describing MCP tool the agent discovers and invokes through a uniform interface. Via MCP, a source need not be a knowledge graph, it can join the federation provided it produces SKG-O-compliant JSON-LD.

4 Graph Processing Engine and Graph as a Service. The Graph Processing Engine loads the JSON-LD from the Data Fetcher into a local RDF graph store, deduplicates entities by reconciling persistent identifiers rather than labels: records from different sources are merged when they share a DOI or another PID exposed through `datacite:hasIdentifier`. Records without a shared PID are deliberately kept distinct and annotated with their source, so that no identity is asserted where none can be established, at the cost of residual duplication when sources describe the same object under different identifier schemes. The engine then executes the SPARQL query from the Translator against this harmonised graph. Each result is annotated with the provenance of the source it originated from, and results from multiple sources are consolidated into a single response. These graphs are specific to each user query and are temporary, constructed on-the-fly. The use of QLever and its lightweight containerised model enables the Graph as a Service (GaaS) capability necessary to support the operationalisation of the federated queries.

5 Federation Registry. The Federation Registry maintains machine-actionable descriptions of the available sources: supported SKG-IF endpoints, data types, filtering options, access mode, and rate limits. The Data Fetcher consumes these descriptions to dynamically select which sources to query for a given request.

5. Evaluation of the approach

In this section, we assess two dimensions that are critical to the viability of the approach. First, whether SKG-O is expressive enough to capture the competency questions collected from domain experts: if the queries cannot be written, the approach fails regardless of infrastructure (Section 5.1). Second, whether the "fetch then query locally" pattern is operationally feasible at scale, since the approach depends on the ability to ingest data from multiple sources, build temporary consolidated graphs, and resolve complex cross-graph joins efficiently (Section 5.2).

5.1. Competency questions

Of the 70 CQs collected (Section 3.3), we focus on three top priority CQs that illustrate the gradient of implementability.

The CQ "Given a publication PID, find all articles that cite it plus those cited in the publication itself" focuses on returning a list of PIDs of articles, and it can be answered today using the current SKG-O specification, as shown in Listing 1.

```
PREFIX cito: <http://purl.org/spar/cito/>
PREFIX datacite: <http://purl.org/spar/datacite/>
PREFIX fabio: <http://purl.org/spar/fabio/>
PREFIX literal: <http://purl.org/spar/literal/>

SELECT DISTINCT ?pid_value
WHERE {
```

```
    VALUES ?article_id_value {"<ARTICLE_ID_VALUE>"}
    ?article datacite:hasIdentifier [
       literal:hasLiteralValue ?article_id_value .
    ] .
    {
       ?article cito:cites [
          a fabio:ScholarlyWork ;
          datacite:hasIdentifier [
             literal:hasLiteralValue ?pid_value
          ]
       ] .
    }
    UNION
    {
       ?article_citing a fabio:ScholarlyWork ;
          cito:cites ?article ;
          datacite:hasIdentifier [
             literal:hasLiteralValue ?pid_value
          ] .
    }
}
```

Listing 1: CQ1: Find citing and cited articles for a given publication PID.

The CQ "Give me the list of identifiers of all the available datasets related to topic X and keywords Y", instead, shows a case where we were able to obtain a fully working SKG-O-compliant SPARQL query but we had to flatten the concept keyword within that of topic. The query is shown in Listing 2.

```
PREFIX bido: <http://purl.org/spar/bido/>
PREFIX datacite: <http://purl.org/spar/datacite/>
PREFIX fabio: <http://purl.org/spar/fabio/>
PREFIX literal: <http://purl.org/spar/literal/>
PREFIX prism: <http://prismstandard.org/namespaces/basic/2.0/>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>

SELECT DISTINCT ?pid_value
WHERE {
   VALUES ?topic_label {"<TOPIC_LABEL>"}
   ?dataset a fabio:Dataset ;
      bido:holdsBibliometricDataInTime [
         bido:withBibliometricData [
            skos:prefLabel ?topic_label
         ]
      ] ;
      datacite:hasIdentifier [
         datacite:usesIdentifierSchema datacite:doi ;
         literal:hasLiteralValue ?pid_value
      ] .
}
```

Listing 2: CQ2: Find dataset identifiers by topic (keyword flattened to topic).

Finally, the CQ "I have to do research in [domain] using [specific methodology]. List highest citations matching these combinations, prioritising newest contributions" cannot be translated appropriately because, at the moment, SKG-O would allow modelling the domain as a topic, e.g. by flattening it as shown in the previous query, but the notion of methodology is not present in the current SKG-IF model, and thus this CQ cannot be answered.

In total, of the 10 high-priority CQs, 4 have direct SKG-O SPARQL translations, 3 are translatable under assumptions (flattening "keyword" and "scientific domain" into the SKG-IF Topic entity), 1 requires an SKG-IF extension for a missing concept (methodology), and 2 are out of scope as they required cross-source entity identification, now handled by the deduplication step of the Graph Processing Engine. This gradient identifies exactly which parts of the SSH cross-graph querying problem SKG-O solves today and which require principled extensions.

Table 2
Comparative evaluation of RDF backends across 12 criteria (3 = best, 1 = weakest). QLever scores reflect the updated assessment (April 2026) after resolution of SPARQL 1.1 compliance gaps. GraphDB was excluded due to licensing constraints despite strong technical scores.

Category	Criterion	GraphDB	Virtuoso	QLever
Standards & Ecosystem	RDF and SPARQL compliance	3	2	3
	Reasoning capabilities	3	2	1
	Federated querying	3	3	3
Technical Feasibility	Query performance on large datasets	2	2	3
	Data ingestion & indexing scalability	2	2	3+
	Clustering and distribution	2	2	1
Operations & Deployment	Ease of deployment	2	1	3
	Maturity and ecosystem support	3	3	2
	Monitoring and admin tools	2	3	2
Strategic fit	Open-source compliance and licensing	1	2	3
	Long-term sustainability	3	3	2
	Interoperability and FAIR alignment	2	3	3
TOTAL		28	28	28+

A "+" marks a criterion on which measured behaviour exceeded what the top of the scale was defined to capture: QLever indexed the 155M-triple dataset in 3 min 41 s against 15 min for Virtuoso. Totals carry the annotation forward.

5.2. Operational feasibility at scale: triplestore evaluation

The RDF backend selection followed a structured evaluation documented in the project's technical architecture [34]. Three engines were compared: Virtuoso [35], QLever [8], and GraphDB [36]. The twelve criteria are grouped into four pillars defined in that document and scored on a three-point scale: 1 = low alignment (major concerns or poor fit), 2 = moderate alignment (partially meets expectations), 3 = high alignment (fully meets the criterion). The framework was designed to expose trade-offs rather than to select a winner mechanically. The scores reported in Table 2 are the April 2026 reassessment: the SPARQL 1.1 compliance gaps that had constrained QLever in the initial evaluation were resolved in the interim, and the benchmark described below was carried out.

These scores were validated through a systematic benchmark on a 155-million triple synthetic dataset, documented in a dedicated evaluation report [37], reflecting the GoTriple KG structure (8 million documents, 10 million named graphs).[1] The benchmark covered eight areas including ingestion performance, SPARQL 1.1 compliance, named graph management, federation, concurrency, and update persistence. QLever outperformed Virtuoso on 80% of tested query patterns. Notable results include a 5x ingestion speed advantage (155M triples indexed in 3 minutes 41 seconds vs 15 minutes), efficient deep pagination (OFFSET 10,000 in 0.008 seconds vs Virtuoso's hard rejection at that threshold), and successful completion of cross-graph joins on 10 million named graphs where Virtuoso timed out. The main operational trade-off is that QLever follows a consolidation model rather than a transactional one: updates are applied to a delta layer and become durable after a scheduled rebuild-index operation, measured at 23 seconds for 155M triples. The dataset is synthetic: it reproduces the structural profile of the GoTriple KG (8 million documents, 1 million datasets, 500k profiles, 10.1 million named graphs) but not its real-world value distributions, and its citation layer covers 150k of the 8 million documents. The evaluation report documents this scope explicitly. These results should therefore be read as indicative of relative engine behaviour under GoTriple-like structures rather than as absolute throughput figures transferable to arbitrary workloads.

Consolidating on QLever enabled the Graph as a Service model: instances can be spun up in minutes from an RDF dump, making it feasible to offer pre-configured endpoints to federation members lacking their own infrastructure.

[1]The benchmark environment, including the dataset generator, orchestration scripts and evaluation notebook, is openly available at https://doi.org/10.5281/zenodo.22683758.

6. Discussion and limitations

6.1. Why mediate via the SKG-IF API

Implementing SKG-IF as the federation backbone across real SSH sources of research information was an instructive process. The framework proved effective as an interoperability contract, but its application in a federation context required adaptations that informed the architectural design described in Section 4. We report on four areas where our implementation experience led to specific design choices.

Adapting to structural diversity across data models. SKG-O follows the FRBR pattern, connecting research products to properties through intermediate entities. This modelling choice reflects sound ontological principles for bibliographic data. However, several established SSH sources use flatter structures. An example of a simple query: "find all open access publications related to topic X with keyword Y". With the TRIPLE Ontology (GoTriple), this is 5 triple patterns (Listing 3). Against SKG-O, the same question requires 9 triple patterns and 4 intermediate nodes (Listing 4).

```
SELECT DISTINCT ?doc WHERE {
  ?doc a triple:Document ;
    triple:hasAccessCondition acc:open_access ;
    sioc:topic ?topic ;
    schema:keywords ?keyword .
  VALUES ?topic { <X_URI> }
  VALUES ?keyword { <Y_URI> }
}
```

Listing 3: Query against GoTriple's flat TRIPLE Ontology (5 triple patterns).

```
SELECT DISTINCT ?researchProduct WHERE {
  ?researchProduct a fabio:Work ;
    frbr:realization ?expression .
  ?expression pso:holdsStatusInTime ?statusInTime .
  ?statusInTime pso:withStatus pso:open-access .
  ?researchProduct bido:holdsBibliometricDataInTime ?topicRelX .
  ?topicRelX bido:withBibliometricData ?topicX .
  ?researchProduct bido:holdsBibliometricDataInTime ?topicRelY .
  ?topicRelY bido:withBibliometricData ?topicY .
  VALUES ?topicX { <X_URI> }
  VALUES ?topicY { <Y_URI> }
}
```

Listing 4: Same query against SKG-O's FRBR-based model (9 triple patterns, 4 intermediate nodes).

The same structural difference affects authors, publishers, contributors, dates, access conditions, licenses, and identifiers. This is not a limitation of SKG-IF, the REST API handles this transparently by returning JSON-LD, but it means that direct SPARQL federation across heterogeneous endpoints, in the case the original sources already expose their data using Semantic Web technologies, requires either data transformation or query adaptation. This trade-off has been studied systematically by Arenas-Guerrero et al. [38], who find data transformation generally more efficient over heterogeneous sources, particularly for complex analytical queries. Our "fetch then query locally" design is an instance of that pattern: we reached it empirically, from the structural mismatch described above, and their analysis provides its theoretical grounding. This observation was the primary motivation for the architecture of Section 4.

Extending filter capabilities. The SKG-IF data model defines properties (topics, publication dates, access rights) valuable for filtering. The current SKG-IF API specification exposes a subset of these as query parameters. Expanding the available filters, particularly for topic, date, and access rights on the /products endpoint, would reduce the amount of data transferred in federated queries. This has been communicated to the SKG-IF working group. The strategy in GRAPHIA will be to propose a formal SKG-IF extension to address the required modifications to the API specifications.

Table 3
The three federation approaches considered, and the trade-offs that led to retaining the third.

Approach	Enables	Cost	Blocking factor
1. Materialise SKG-O triples in each source	Direct SPARQL federation	Every member adds and maintains FRBR-compatible triples alongside its native model	Alignment burden on all members; adoption risk across 20 partners
2. Materialise in an intermediate graph	Centralised, controlled alignment	Continuous ingestion and freshness management	Reintroduces aggregation, contradicting the federated model
3. Mediate via the SKG-IF REST API	Structural neutrality; non-RDF sources can join	Depends on API filter coverage; suited to entity-level retrieval	Retained

Broadening the SKG-O Grant concept. SKG-IF's Grant entity type covers funded research. SSH research frequently also takes the form of institutional initiatives, citizen science, or community collaborations. GoTriple's Project maps as `skos:closeMatch` to `frapo:Grant` rather than `skos:exactMatch`. Broadening the model to encompass unfunded research activities would improve coverage for the SSH domain.

Towards cultural heritage support. EHRI (RiC), E-RIHS DIGILAB (CIDOC-CRM), and RESILIENCE manage objects central to SSH that are not yet covered by SKG-IF's publication-centric model. Most of their output would be generically marked as Research Products of type "other". The extension mechanism could accommodate this. Defining a cultural heritage extension profile is a perspective that GRAPHIA is well positioned to explore.

The implementation experience described above directly shaped the Gateway architecture (Section 4). Three approaches were considered, summarised in Table 3.

These options were weighed on adoption cost, architectural fit and dependency risk, not measured experimentally: implementing and benchmarking all three was beyond the scope of this stage, and we report the comparison as a design rationale rather than as an evaluation. The SKG-IF REST API returns JSON-LD regardless of internal graph structure, so it already solves the structural diversity problem at the API level. Rather than attempting SPARQL federation across heterogeneous endpoints, we use the REST API to retrieve data and resolve SPARQL queries locally against a harmonised graph.

Approach 3 is what the current architecture implements. The insight is that SKG-IF's REST API, designed for interoperability, is also the right tool for federation: it provides the structural neutrality that SPARQL federation across heterogeneous endpoints cannot guarantee. The use of MCP as an orchestration layer for dynamically querying SKG-IF endpoints, and an LLM as the actual interoperability engine, is, to our knowledge, novel in this context. Its tool description mechanism allows dynamic endpoint discovery beyond static registry lookups. Whether this approach scales to complex analytical queries remains open, it is better suited to entity-level retrieval. We are also contributing our implementation experience upstream to the SKG-IF working group.

6.2. Limitations

The current proof of concept connects two graphs, GoTriple and OpenCitations, which are currently providing an initial implementation of the SKG-IF API. Federating across five or more sources remains to be explored. The MCP/REST path is being prototyped. The GaaS model has not been deployed operationally. The pipeline currently requires a natural language input as entry point; the architecture is designed to also accept direct SKG-O compliant SPARQL queries, but this mode is not yet implemented.

The competency questions validated so far are entity-centric retrievals. None requires transferring intermediate results between endpoints or computing aggregations across sources, both standard in

federated query processing and neither supported by the current pipeline: the Graph Processing Engine evaluates the query over a single consolidated graph, so aggregation is bounded by what has already been fetched. Moving from question-driven to cost-aware, query-driven retrieval is the main open problem this evaluation identifies.

On the evaluation side, no formal quantitative assessment of the natural language to SPARQL translation quality is available at this stage. A community benchmark of 200 question-query pairs is being assembled for Quagga, which remains at version 0.1.0 (alpha); collection closes in October 2026 and the evaluation will be reported separately. LLM-based SPARQL generation introduces hallucination risks, currently mitigated through SHACL [39] validation provided by SKG-IF and testing against the competency question suite.

The paper presents an architecture in active development rather than a deployed system. Federation governance, including data quality responsibility when the GRAPHIA team hosts endpoints on behalf of sources, remains an open challenge. We believe the architectural evolution, the implementation experience with SKG-IF, and the "fetch then query locally" pattern are valuable contributions for projects facing similar federation challenges.

7. Conclusion and future work

We have presented the experience of applying Semantic Web technologies to build a federated infrastructure for SSH knowledge graphs within GRAPHIA, a 20-partner Horizon Europe project. SKG-IF proved to be a viable interoperability contract for the federation. Its non-invasive design allowed four structurally diverse sources to participate without restructuring their data. At the same time, our implementation revealed that vocabulary alignment alone is not sufficient for query federation: the structural differences between FRBR-based and flat data models required us to develop a "fetch then query locally" pattern that leverages SKG-IF's REST API as a structurally neutral access layer rather than attempting direct SPARQL federation.

Three results are transferable beyond the SSH domain: the systematic benchmark leading to QLever adoption as a lightweight, container-based RDF backend (validated on 155 million triples); the hosted onboarding model that lowers the barrier for new sources to join the federation; and the use of MCP as an orchestration layer for dynamically querying SKG-IF endpoints. Our implementation experience with SKG-IF, including specific suggestions for extending the framework to cultural heritage and broadening its filter capabilities, is offered as a contribution to the framework's ongoing evolution.

Future work includes completing the Gateway implementation with a direct SPARQL entry point for expert users, evaluating the Quagga agent's SPARQL generation accuracy against a crowdsourced benchmark (target: 200 question-query pairs), operationalising the Graph-as-a-Service model in production, and proposing a cultural heritage extension profile for SKG-IF.

Acknowledgments

This work has been funded by the European Union's Horizon Europe programme under grant agreement No. 101188018 (GRAPHIA).

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] S. Dumouchel, E. Blotière, G. Breitfuss, Y. Chen, F. Di Donato, M. Eskevich, P. Forbes, H. Georgiadis, A. Gingold, et al., GOTRIPLE: A user-centric process to develop a discovery platform, Information 11 (2020) 563. doi:10.3390/info11120563.

[2] S. Peroni, D. Shotton, OpenCitations, an infrastructure organization for open scholarship, Quantitative Science Studies 1 (2020) 428–444. doi:10.1162/qss_a_00023.

[3] D. Biswas, E. Gupta, R. Yu, B. Zapilko, GESIS knowledge graph (GESIS KG) (version 2.0.0), Data set, GESIS, Cologne, 2025. doi:10.7802/2969.

[4] M. Y. Jaradeh, et al., Open research knowledge graph, in: K-CAP 2019, ACM, 2019, pp. 243–246. doi:10.1145/3360901.3364435.

[5] T. Blanke, M. Bryant, M. Frankl, C. Kristel, R. Speck, V. Vanden Daelen, R. Van Horik, The european holocaust research infrastructure portal, ACM Journal on Computing and Cultural Heritage 10 (2017) 1–18. doi:10.1145/3004457.

[6] European Commission, Directorate-General for Research and Innovation, EOSC Interoperability Framework, Technical Report, Publications Office of the European Union, Luxembourg, 2021. doi:10.2777/620649.

[7] A. Mannocci, S. Peroni, Scientific Knowledge Graphs Interoperability Framework (SKG-IF) WG, Scientific Knowledge Graphs Interoperability Framework (SKG-IF): Enabling Data Exchange for SKGs, Working Group Recommendation 1.0.0, Research Data Alliance, 2025. doi:10.15497/rda00150.

[8] H. Bast, B. Buchhold, QLever: A query engine for efficient SPARQL+Text search, in: CIKM 2017, ACM, 2017, pp. 647–656. doi:10.1145/3132847.3132921.

[9] A. Schwarte, et al., FedX: Optimization techniques for federated query processing on linked data, in: ISWC 2011, volume 7031 of *LNCS*, Springer, 2011, pp. 601–616. doi:10.1007/978-3-642-25073-6_38.

[10] A. Charalambidis, et al., SemaGrow: Optimizing federated SPARQL queries, in: SEMANTICS 2015, ACM, 2015, pp. 121–128. doi:10.1145/2814864.2814886.

[11] O. Görlitz, S. Staab, SPLENDID: SPARQL endpoint federation exploiting VOID descriptions, in: COLD Workshop, ISWC 2011, 2011. https://ceur-ws.org/Vol-782/GoerlitzAndStaab_COLD2011.pdf (last accessed 23 July 2026).

[12] W3C, SPARQL 1.1 federated query, W3C Recommendation, https://www.w3.org/TR/sparql11-federated-query/ (last accessed 23 July 2026), 2013.

[13] F. Belleau, et al., Bio2RDF: Towards a mashup to build bioinformatics knowledge systems, Journal of Biomedical Informatics 41 (2008) 706–716. doi:10.1016/j.jbi.2008.03.004.

[14] D. Vrandečić, M. Krötzsch, Wikidata: A free collaborative knowledgebase, Communications of the ACM 57 (2014) 78–85. doi:10.1145/2629489.

[15] P. Manghi, Challenges in building scholarly knowledge graphs for research assessment in open science, Quantitative Science Studies 5 (2024) 991–1021. doi:10.1162/qss_a_00322.

[16] IFLA, Functional Requirements for Bibliographic Records, Technical Report, International Federation of Library Associations and Institutions, 1998. https://www.ifla.org/wp-content/uploads/2019/05/assets/cataloguing/frbr/frbr_2008.pdf (last accessed 23 July 2026).

[17] ICA Expert Group on Archival Description, Records in Contexts: Conceptual Model, Version 1.0, Technical Report, International Council on Archives, 2023. https://www.ica.org/app/uploads/2024/01/ric-cm-1.0_0.pdf (last accessed 23 July 2026).

[18] M. Doerr, The CIDOC conceptual reference module: An ontological approach to semantic interoperability of metadata, AI Magazine 24 (2003) 75–92. doi:10.1609/aimag.v24i3.1720.

[19] V. Emonet, J. Bolleman, S. Duvaud, T. Mendes de Farias, A. C. Sima, LLM-based SPARQL query generation from natural language over federated knowledge graphs, in: Proceedings of the ISWC 2024 Special Session on Harmonising Generative AI and Semantic Web Technologies, volume 3953 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2024. https://ceur-ws.org/Vol-3953/355.pdf (last accessed 23 July 2026).

[20] GoTriple, TRIPLE ontology, https://gotriple.eu/ontology/triple, 2020.
[21] S. Peroni, D. Shotton, The SPAR ontologies, in: ISWC 2018, volume 11137 of *LNCS*, Springer, 2018, pp. 119–136. doi:10.1007/978-3-030-00668-6_8.
[22] R. V. Guha, et al., Schema.org: Evolution of structured data on the web, Communications of the ACM 59 (2016) 44–51. doi:10.1145/2844544.
[23] International Organization for Standardization, Information and Documentation: Technical Interoperability: Data Documentation Initiative (DDI), Technical Report, ISO/PAS 25955:2026, 2026. https://www.iso.org/standard/92127.html.
[24] S. De Paoli, C. Silver, A. Fawzi, D5.1 User Personas and Scenarios, Technical Report, GRAPHIA Deliverable, Zenodo, 2025. doi:10.5281/zenodo.15690656.
[25] G. Rubidge, C. Silver, Researchers share, GRAPHIA listens: Insights and actions from our latest SSH survey, GRAPHIA Blog, https://graphia-ssh.eu/news-blog/researchers-share-graphia-listens-insights-and-actions-from-our-latest-ssh-survey/ (last accessed 23 July 2026), 2026.
[26] T. Lebo, S. Sahoo, D. McGuinness, PROV-O: The PROV ontology, W3C Recommendation, https://www.w3.org/TR/prov-o/ (last accessed 23 July 2026), 2013.
[27] DCMI Usage Board, DCMI metadata terms (version 2020-01-20), DCMI Recommendation, http://dublincore.org/specifications/dublin-core/dcmi-terms/2020-01-20/ (last accessed 23 July 2026), 2020.
[28] R. Albertoni, D. Browning, S. J. D. Cox, A. Gonzalez Beltran, A. Perego, P. Winstanley, Data catalog vocabulary (DCAT): Version 3, W3C Recommendation, https://www.w3.org/TR/vocab-dcat-3/ (last accessed 23 July 2026), 2024.
[29] N. Matentzoglu, et al., A simple standard for sharing ontological mappings (SSSOM), Database (2022). doi:10.1093/database/baac035.
[30] M. Poveda-Villalón, et al., LOT: An industrial oriented ontology engineering framework, Engineering Applications of Artificial Intelligence 111 (2022) 104755. doi:10.1016/j.engappai.2022.104755.
[31] A. Dubber, M. Magas, GRAPHIA & LUMEN Innovation Prototyping Lab 2025 Report, Technical Report, Zenodo, 2026. doi:10.5281/zenodo.18798396.
[32] e-SDF, Foxcub, Net7, UNIBO, PSNC, TIB, CYI, Genius Loci, IBL-PAN, KU Leuven, LUH, D2.1 GRAPHIA Ontology (1.0), Technical Report, Zenodo, 2026. doi:10.5281/zenodo.17951755.
[33] M. Romanello, Beyond SPARQL: Towards Conversation-based Access to Knowledge Graphs in GRAPHIA, Technical Report, Zenodo, 2026. doi:10.5281/zenodo.19691435.
[34] L. De Santis, J. Homo, K. Darty, A. Margraf, D2.2 Technical Architecture, Technical Report, Zenodo, 2026. doi:10.5281/zenodo.17951781.
[35] O. Erling, I. Mikhailov, RDF support in the Virtuoso DBMS, in: Networked Knowledge: Networked Media, volume 221 of *SCI*, Springer, 2009, pp. 7–24. doi:10.1007/978-3-642-02184-8_2.
[36] B. Bishop, A. Kiryakov, D. Ognyanoff, I. Peikov, Z. Tashev, R. Velkov, OWLIM: A family of scalable semantic repositories, Semantic Web 2 (2011) 33–42. doi:10.3233/SW-2011-0026.
[37] J. Homo, K. Darty, QLever Evaluation Report, Technical Report, Zenodo, 2026. doi:10.5281/zenodo.20019957.
[38] J. Arenas-Guerrero, O. Corcho, M. S. Pérez, Intermediate triple table: A general architecture for virtual knowledge graphs, Knowledge-Based Systems 314 (2025) 113179. doi:10.1016/j.knosys.2025.113179.
[39] W3C, Shapes constraint language (SHACL), W3C Recommendation, https://www.w3.org/TR/shacl/ (last accessed 23 July 2026), 2017.