Enhancing Interoperability of SPARQL Endpoints: RESTful and OAI-PMH API for the DH-ATLAS Project

Giorgia Rubin[1,2,4], Alessia Bardi[1,3,*], Riccardo Del Gratta[2] and Angelo Mario Del Grosso[2]

[1]*Istituto di Scienza e Tecnologie dell'Informazione "Faedo", Consiglio Nazionale delle Ricerche*
[2]*Istituto di Linguistica Computazionale "Zampolli", Consiglio Nazionale delle Ricerche*
[3]*OpenAIRE AMKE*
[4]*Università di Pisa*

Abstract

Digital repositories leveraging RDF and Linked Data for Cultural Heritage metadata face a critical challenge: their native SPARQL endpoints often create silos because major aggregators rely on the OAI-PMH protocol for harvesting, and developers prefer RESTful APIs. This divergence undermines the visibility and FAIRness (Findable, Accessible, Interoperable, Reusable) of scholarly resources. This paper details the unified access strategy implemented for the DH-ATLAS project, which generated an Ontology and Knowledge Graph focused on Italian Digital Cultural Heritage research. To ensure broad dissemination and resource reuse without duplicating data, DH-ATLAS developed two configurable and modular software components to implement compatibility with the OpenAIRE guidelines and REST clients. Together, these components establish a cohesive and reusable solution for integrating semantic repositories with diverse metadata consumers, promoting the long-term sustainability and broader accessibility of the DH-ATLAS Knowledge Graph and serving as a model for other RDF infrastructures.

Keywords

Interoperability, Knowledge Graph, API, OpenAIRE, OAI-PMH, Digital Humanities and Cultural Heritage

1. Introduction

The adoption of the Resource Description Framework (RDF)[1] and SPARQL[2] in digital libraries and cultural heritage institutions has transformed how metadata is structured, queried, and linked. RDF enables expressive modeling of entities, relationships, multilingual, and multimodal content, while SPARQL provides a standardized query language for accessing this data.

However, developers of metadata aggregators and discovery platforms prefer OAI-PMH and RESTful APIs rather than SPARQL endpoints.[1, 2] While SPARQL endpoints are powerful, they are not always accessible or intuitive for developers building web applications, data portals, or client-side tools.

RESTful APIs remain the dominant approach for data access in modern software ecosystems, offering simplicity, scalability, and compatibility with standard HTTP tooling. Without a RESTful interface, RDF repositories risk being siloed from broader application development and integration efforts.

OAI-PMH remains a widely adopted protocol for metadata harvesting, particularly among aggregators such as Europeana[3], ARIADNE[4], the OpenAIRE Graph[5], the CLARIN VLO [3], and national digital library networks. OAI-PMH is designed to facilitate interoperability across heterogeneous repositories. Despite its limitations, such as lack of semantic richness, it remains a cornerstone of metadata aggregation workflows. However, RDF repositories typically lack native support for OAI-PMH, making it difficult

IRCDL2026
*Corresponding author.
giorgiarubin@cnr.it (G. Rubin); alessia.bardi@isti.cnr.it (A. Bardi); riccardo.delgratta@cnr.it (R. D. Gratta); angelomario.delgrosso@ilc.cnr.it (A. M. D. Grosso)
0009-0008-0584-7290 (G. Rubin); 0000-0002-1112-1292 (A. Bardi); 0000-0003-1867-8445 (R. D. Gratta); 0000-0002-4867-6304 (A. M. D. Grosso)

[1]https://www.w3.org/RDF/
[2]https://www.w3.org/TR/sparql11-query/
[3]https://europeana.eu
[4]https://ariadne-infrastructure.eu/
[5]https://graph.openaire.eu

for aggregators to harvest metadata without custom integrations or data duplication. Institutions are often forced to maintain separate OAI-PMH-compliant repositories or transform RDF data into XML formats manually. This approach is inefficient and undermines the benefits of semantic modeling on which RDF is based.

This divergence creates challenges for institutions that have invested in RDF and Linked Open Data infrastructures: how to expose their metadata via protocols compatible with external systems without duplicating data or maintaining parallel repositories. Being indexed by third-party aggregators like Europeana, ARIADNE, the OpenAIRE Graph, and the CLARIN VLO is fundamental for the visibility of digital humanities institutions and for increasing the level of FAIRness (Findable, Accessible, Interoperable, Reusable)[4] of their resources.

The DH-ATLAS project[5], funded by the Next Generation program of the European Commission from 2023 to 2025, delivered an ontology and a knowledge graph about research projects and resources on Italian Digital Cultural Heritage. The DH-ATLAS Ontology [6] establishes a semantic framework for the detailed representation of the diverse forms of DH resources (such as digital editions, software, text collections, ontologies, and Linked Open Data) and their relationships with other entities of the research ecosystem (e.g. authors, organisations, projects). The DH-ATLAS Knowledge Graph (KG)[7, 8] has been populated by project members in collaboration with the Italian Digital Heritage (DH) community, which was involved via a number of workshops and seminars. Such events involved master and doctoral students as well as junior and senior researchers in the field of DH. Data was also enriched via semi-automatic enrichment processes leveraging Wikidata, ORCID and other trusted sources.[9]

For a broader dissemination of the DH-ATLAS KG, the project collaborated with OpenAIRE to: (i) include the data of the DH-ATLAS KG in the OpenAIRE Graph, and (ii) identify research outputs in the OpenAIRE Graph that are related to the resources described in the DH-ATLAS KG. Moreover, inspired by the FAIR recommendations and experiences of other initiatives working with RDF (see for example OpenCitations[6], and ARCO[10]), we wanted to ease access and reuse to the resources of the DH-ATLAS KG by exposing a RESTful API.

To address this, we developed an interoperability suite composed of two open-source tools to expose RDF resources in a format compliant with the OpenAIRE guidelines and to offer RESTful API over a SPARQL endpoint.

Each tool targets a distinct interoperability need. Together they form a cohesive strategy for making the DH-ATLAS KG accessible to a wide range of metadata consumers. Thanks to the modularity and configurability of the software, the solution can be adapted to other RDF repositories with a SPARQL endpoint.

Outline. Section 2 presents related work highlighting the gaps in existing literature and practice. Section 3 discusses the methodology adopted to design and develop an interoperability suite for the DH-ATLAS project and customisable for other user cases. The architecture of the interoperability suite and details on design and implementation choices are presented in section 4. The final section summarises the achievements and limitations of the proposed solution and describe the actions that have been carried out to ensure its long-term sustainability.

2. Related Works

Our research of literature and practices to offer an OAI-PMH publisher over a SPARQL endpoint has not produced any relevant results. On the other hand, we found several works concerning SQL databases, repository and journal platforms.

OAIzer [11] is a tool for the configuration and automatic deploy of OAI-PMH interfaces on digital libraries built on top of relational databases. A more recent software was produced in the context of the EOSC Beyond EC project [12][7]. The software implements a SpringBoot application that exports metadata records stored in a PostgreSQL database via the OAI-PMH protocol.

[6]OpenCitations REST API built with RAMOSE: https://opencitations.net/querying/
[7]On the EOSC Beyond Discovery Hub at: https://doi.org/21.15133/puycna

Platforms for digital repositories and journals have also been extended with OAI-PMH support to increase the interoperability within the scholarly communication ecosystem.

DSpace, on top of which are built some of the trusted repositories of the CLARIN research infrastructure and the Italian university repositories[8], has a built-in functionality for OAI-PMH[9] and compatibility with the OpenAIRE guidelines[10]. Other platforms supporting OAI-PMH by default are Dataverse[11], and Figshare[12].

ePrints also supports OAI-PMH[13]. Compatibility with the OpenAIRE guidelines is possible with a plugin[14].

The Fedora platform used to support a non-configurable OAI-PMH endpoint.[13] The support to OAI-PMH was dropped, and some institutions had to implement their own home-made solutions to not stop being harvested by aggregators.[15]

The Open Journal System (OJS), a platform for Open Access journals, has also built-in support for OAI-PMH and a dedicated plugin to ensure compatibility with the OpenAIRE guidelines[14].

Several software tools for the deployment of REST API over SPARQL endpoints are available. In addition to RAMOSE[15], which we selected due to its configurability and the expertise of the project members with it, we mention BASIL [16], grlc [17], Ontology-Based API (OBA)[18] and CRAFTS[2].[16]

3. Methodology

The methodological framework adopted for this work was designed to tackle interoperability challenges between RDF-based infrastructures, traditional metadata aggregation protocols (namely OAI-PMH), and current practices in web development, while ensuring full compliance with FAIR principles and Open Science practices. The proposed approach combines semantic analysis, software design and development, and community-driven validation.

3.1. Requirements

The first step consisted in defining functional and non-functional requirements for a minimum viable interoperability solution for the DH-ATLAS KG, adaptable to other small-mid scale DH projects:

- An OAI-PMH Publisher that OpenAIRE can use to harvest metadata records hosted in a SPARQL endpoint;
- A RESTful endpoint to access RDF data, enabling developers to retrieve RDF resources without SPARQL expertise.

To be harvestable by OpenAIRE, the OAI-PMH Publisher must:

- Support the OAI-PMH verb ListRecords
- Expose records in the format of the OpenAIRE Guidelines v4 [17]

Full compliance with the OAI-PMH protocol is considered desirable but not required for the minimum viable version of the DH-ATLAS project. In addition, the solution must maximise semantic fidelity and

[8]Most Italian University repositories use IRIS, a version of DSpace customised by CINECA
[9]OAI-PMH functionality for DSpace https://wiki.lyrasis.org/display/DSDOC9x/OAI
[10]DSpace compatibility with the OpenAIRE guidelines https://wiki.lyrasis.org/display/DSPACECRIS/OpenAIRE+Integration
[11]OAI-PMH functionality on Dataverse https://guides.dataverse.org/en/latest/admin/harvestserver.html
[12]OAI-PMH in Figshare: https://info.figshare.com/user-guide/how-to-use-figshares-oai-pmh-service/
[13]OAI-PMH functionality for ePrints: https://wiki.eprints.org/w/OAI
[14]ePrints plugin for the OpenAIRE guidelines https://github.com/eprintsug/EPrintsOpenAIRE
[15]Thread on Fedora Google group: https://groups.google.com/g/fedora-tech/c/MMXtfe-J-uo?pli=1; implementation by SAW Leipzig (CLARIN centre that contributes to the CLARIN Virtual Language Observatory (VLO)) for Fedora v6: https://github.com/saw-leipzig/foaipmh
[16]A detailed analysis of the tools goes beyond the scope of this work: comparison tables are available in the respective articles.
[17]OpenAIRE guidelines v4: https://openaire-guidelines-for-literature-repository-managers.readthedocs.io/en/v4.0.0/

preservation of the DH-ATLAS Ontology semantics during transformation into OpenAIRE-compliant records.

Both the RESTful endpoint and the OAI-PMH Publisher software should be configurable and therefore re-usable in settings different than the DH-ATLAS project.

Non-functional requirements focused on sustainability and usability:

- Modularity and extensibility: the architecture should allow future support for additional metadata formats (e.g., Dublin Core) and vocabularies.
- Performance optimization: avoid heavy SPARQL queries that could overload small RDF stores typical of DH projects.
- Open Science compliance: release under an open-source license, with public documentation and reproducible workflows.
- Deployment readiness: compatibility with appropriate infrastructure for long-term hosting and monitoring.

3.2. Analysis of data models and mappings

A key methodological challenge was achieving semantic alignment between the DH-ATLAS Ontology v.2 and the OpenAIRE Guidelines v.4. The DH-ATLAS Ontology provides a detailed representation of Italian Digital Humanities resources, treating them as datasets—objects that can be described and catalogued through structured metadata. Many of the metadata elements selected for describing research products correspond to basic, transversal properties common across most existing models, including the OpenAIRE Application Profile. These include mandatory properties that ensure core identification and essential usage information. Beyond these, DH-ATLAS introduces a wide set of optional properties that capture features specific to cultural heritage research products, often providing information that is difficult to find or entirely absent in standard documentation. This allows DH-ATLAS to reach a high level of granularity, offering type-specific metadata that extend significantly beyond the cross-cutting properties shared by different resource types [6].

In contrast, OpenAIRE adopts a more generic schema based on Datacite[18] and Dublin Core[19], with vocabularies defined by the Coalition of Open Access Repositories (COAR)[20], prioritizing research outputs and their relationships with organizations, projects, and funding information. It also provides greater detail in structural elements—such as creator subproperties, affiliations, and identifier specifications—whereas DH-ATLAS typically groups these under broader Schema.org[21] properties. Given the intention of our OAI-PMH publisher to expose DH-ATLAS resources to the OpenAIRE Graph, this divergence in scope required simplifying the fine-grained, cultural-heritage-specific properties of DH-ATLAS. Type-dependent metadata were therefore omitted, retaining only the transversal core elements needed for interoperability, while structural properties were expanded to match the mandatory subproperties required by the OpenAIRE model.

The mapping in Figure 1 [19] illustrates the alignment of shared metadata properties between the DH-ATLAS and OpenAIRE models, detailing the semantics and structural features of each mapped element. This required a multi-step alignment process:

- Normalization of property cardinality, ensuring compatibility with OpenAIRE constraints.
- Identification of equivalences between properties in the two models.
- Vocabulary alignment, integrating COAR Resource Types to guarantee compliance with OpenAIRE controlled list:
  - `atlas:DigitalScholarlyEdition → coar:encodedData`;
  - `atlas:Ontology → coar:KnowledgeOrganizationSystem`;

[18] Datacite schema: https://schema.datacite.org/
[19] Dublin Core: https://www.dublincore.org/schemas/
[20] COAR Vocabularies: https://vocabularies.coar-repositories.org/
[21] Schema.org: https://schema.org/

ID	OpenAIRE Application Profile	Cardinality	Type	Range Restriction	ATLAS Ontology	Cardinality	Type	Range Restriction
	Research Product				schema:Dataset			
1	**datacite:identifier**	1	URI		**URI**	1	URI	
1.1	datacite:identifierType	1	xsd:String	ARK; DOI; Handle; PURL; URN; URL	(url)			
2	**datacite:title**	1-n	xsd:String		**schema:name**	1-n	xsd:String	
3	**dc:description**	0-n	xsd:String		**schema:description**	1-n	xsd:String	
4	**datacite:creator**	1-n			**schema:creator**	1-n	URI	foaf:Agent[ORCID;catalogue]
4.1	datacite:creatorName	1-n	xsd:String		schema:name			
4.2	datacite:affiliation	0-n	xsd:String		schema:affiliation			
4.3	datacite:nameIdentifier	0-n			schema:identifier			
4.3.1	nameIdentifierScheme	1	xsd:String		(schema prefix)			orcid, wikidata
4.3.2	schemeURI	1	URI		(schema uri)			
5	**datacite:contributor**	0-n			**schema:contributor**	0-n	URI	foaf:Agent[ORCID;catalogue]
5.1	datacite:contributorName	1-n	xsd:String		schema:name			
5.2	datacite:affiliation	0-n	xsd:String		schema:affiliation			
5.3	datacite:nameIdentifier	0-n			schema:identifier			
5.3.1	nameIdentifierScheme	1	xsd:String		(schema prefix)			orcid, wikidata
5.3.2	schemeURI	1	URI		(schema uri)			
6	**dc:publisher**	0-n	xsd:String		**schema:publisher**	1-n	URI	foaf:Agent[ORCID;catalogue]
7	**dc:language**	0-n	xsd:String	Language code preferrably in ISO 639-3	**schema:inLanguage**	0-n	URI	EU Vocabularies - Language
8	**datacite:subject**	0-n	xsd:string		**schema:educationalUse**	1-n	URI	TaDiRAH
9	**oaire:file**	0-n	URI		**schema:distribution**	1-n	URL	
10	**datacite:alternateIdentifier**	0-n	URI		**schema:url**	0-1	URL	
10.1	datacite:alternativeidentifierType	1	xsd:string	Controlled list of identifiers types	(url)			
11	**oaire:fundingReference**	0-n			**schema:producer**			
11.1	oaire:funderName	1	xsd:String		schema:name	0-1	URI	schema:ResearchProject
12	**oaire:resourceType**	1			**schema:additionalType**	1	URI	atlas:ScholarlyDigitalEdition ; atlas:TextualArchive ; atlas:LinkedOpenData ; atlas:Ontology ; atlas:Software; atlas:LanguageModel ; atlas:3DDigitalTwin
12.1	text value	1	xsd:String	COAR Resource Type Vocabulary (label)	(text value)			
12.2	uri	1	URI	COAR Resource Type Vocabulary (URI)	(uri)			
12.3	resourceTypeGeneral	1	xsd:String	literature, dataset, software or other research product	schema:additionalType			
13	**datacite:date**	1			**schema:datePublished**	1	xsd:Date	ISO 8601 [W3CDTF]
13.1	text value	1	xsd:Date	ISO 8601 [W3CDTF]	(text value)			
13.2	datacite:dateType	1	xsd:String	Controlled vocabulary	("issued")			
14	**datacite:rights**	1			**dct:accessRights**	1	URI	COAR
14.1	text value	1	xsd:string	COAR Access Right Vocabulary (label)	(text value)			
14.2	datacite:rightsURI	1	URI	COAR Access Right Vocabulary (URI)	(uri)			
15	**oaire:licenseCondition**	0-1			**schema:license**	1	URI	schema.gov - Licenze | EU Vocabularies - Licenses
15.1	text value	1	xsd:string		(text value)			

Figure 1: DH-ATLAS and OpenAIRE models mapping. It establishes a coherent alignment between the two models.

- `atlas:TextCollection` → `coar:collection`;
- `atlas:LinkedOpenData` → `coar:dataset`;
- `atlas:Software` → `coar:software`;
- `atlas:3DDigitalTwin` → `coar:dataset`;
- `atlas:LanguageModel` → `coar:software`.

3.3. Implementation

The implementation phase started with a desk research on existing frameworks and technologies. Following the best practices of Open Science and software development, we did not want to reinvent the wheel, but rather re-use existing solutions and adapt them to our use case.

As detailed in Section 2 a set of options were available to set up a RESTful API over a SPARQL enpoint. Among those, we selected RAMOSE [15] due to its configurability and the expertise of the project members with it.

Instead, we could not find any existing framework for setting up an OAI-PMH endpoint. We decided therefore to implement a software that can be used to serve use cases like the one we had in DH-ATLAS.

Details on the architecture of the two software components are given in Section 4.

3.4. Test and validation

Testing activities combined functional validation with compliance checks, ensuring both technical correctness and adherence to standards:

- OAI-PMH compliance: responses were validated against the OpenAIRE Guidelines v4 with the OpenAIRE Validator Service[22] to guarantee proper XML structure, inclusion of mandatory elements, and correct usage of controlled vocabularies.
- RESTful API functionality: endpoints were tested using tools such as Postman and cURL, verifying parameter handling, response consistency, and JSON serialization.
- Community feedback: preliminary versions were shared with DH-ATLAS contributors and OpenAIRE technical staff.

Validation also addressed semantic integrity, confirming that transformed records preserved essential relationships and provenance information from the DH-ATLAS Knowledge Graph.

3.5. Deployment

Deployment targeted long-term sustainability and Open Science compliance. Both services were containerized and integrated into the D4Science infrastructure[20], hosted by CNR-ISTI. This environment offers:

- Monitoring and accounting: automatic tracking of service usage.
- Authentication and authorization: via the D4Science Authorization and Authentication Infrastructure (AAI) based on OpenID Connect (OIDC).
- Virtual Research Environment (VRE): facilitating collaborative access for DH researchers.

The source code for both tools has been released under an open-source license on GitHub, accompanied by documentation and configuration examples. [21, 22]. This ensures reproducibility and fosters reuse by other institutions seeking to expose RDF metadata via OAI-PMH or RESTful APIs.

4. Architecture

4.1. The DH-ATLAS Project

The DH-ATLAS project[5] built a knowledge graph of resources about digital humanities scholarly resources and projects on Italian Cultural Heritage. The DH-ATLAS Ontology[6, 23], establishes a semantic framework for the detailed representation of the diverse forms of DH resources and their relationships with other entities of the research ecosystem. The ontology models:

- research products that can be: text collections, digital scholarly editions, software, linked open data, ontologies, language models, and 3D models;
- research projects
- data services
- websites

Entities are semantically linked to each others (e.g. a research project produced a digital scholarly edition), to persons (e.g. creators, authors), to organisations (e.g. participants to research projects, affiliations of authors).

The DH-ATLAS Knowledge Graph (KG)[7, 8] has been populated with the DH-ATLAS App, an instance of the CLEF software (Crowdsourcing Linked Entities via web Form) [24], a LOD-native crowd sourcing platform for collaborative data collection and curation. For the project, CLEF was configured

[22] OpenAIRE Provide: https://provide.openaire.eu

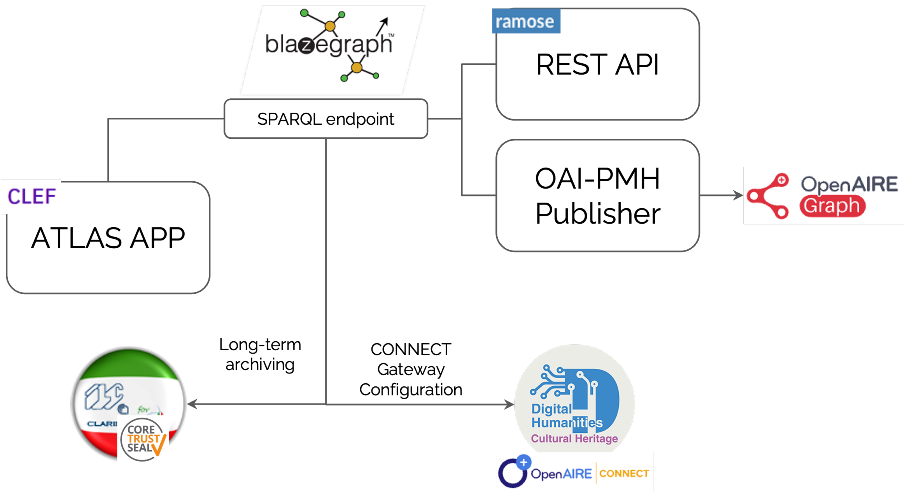

Figure 2: Overview of the architecture of the DH-ATLAS project

to use the DH-ATLAS Ontology and extended to support data entry and curation with semi-automatic enrichment processes leveraging Wikidata, ORCID and other trusted sources.[9]

The backend of CLEF - and therefore of the DH-ATLAS App - is a triple store: an instance of BlazeGraph[23]. The DH-ATLAS App communicates with BlazeGraph via the SPARQL protocol. The same SPARQL endpoint is used by the other components of the DH-ATLAS suite:

- The OAI-PMH Publisher (see Section 4.2)
- The RESTful API, built on top of RAMOSE (see Section 4.3)

Data was extracted from the SPARQL endpoint also to:

- Export the triples and deposit them in the repository of CLARIN Italy, a trusted, persistent repository certified with the TrustCore Seal
- Extract information about the resources of the KG and use it to configure the Italian subcommunity of the DH-CH Gateway of OpenAIRE[24]. Specifically, the extracted information has been used to specify the criteria by which a research product in the OpenAIRE Graph should be considered about Italian CH.

An overview of the architecture of the DH-ATLAS suite is depicted in Figure 2.

4.2. OAI-PMH Publisher

The OAI-PMH Publisher implemented for the DH-ATLAS project is a minimum viable product for the project requirements.[25] As such, it currently supports only the delivery of metadata records compliant to the OpenAIRE Guidelines,[26] although it can be easily extended to deliver other metadata formats (such as Dublin Core).

The OAI-PMH Publisher provides the following characteristics:

- Modular SPARQL integration: Queries are configurable per verb and metadata format.
- Metadata transformation: RDF triples are provided following the OpenAIRE Guidelines v4 specification.
- Extensibility: The software is Designed to support additional export metadata formats and ontologies.

[23]BlazeGraph web site: https://blazegraph.com/
[24]OpenAIRE Gateway for Digital Humanities and Cultural Heritage: https://dh-ch.openaire.eu
[25]The current implementation of the Publisher along with all the required configuration files is available at https://github.com/dh-atlas/oai-publisher.git.
[26]The current implementation returns metadata complied with the `oai-datacite` profile

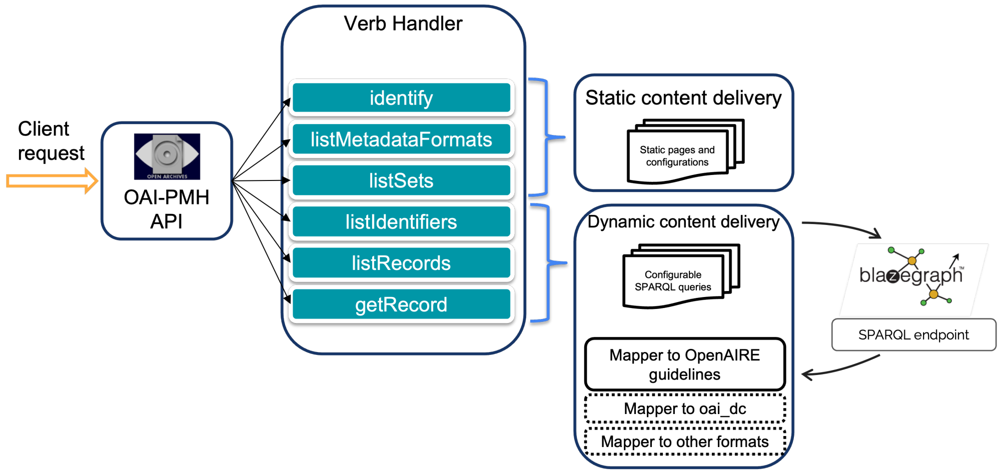

Figure 3: Design of the OAI-PMH Publisher

The service is implemented as a Flask application with Unicorn having three main modules and two configuration files. The OAI-PMH API is the interface used by clients to perform requests. The component dispatches the specific call (i.e. one of the OAI-PMH verbs) and relative parameters to the Verb Handler component. As Figure 3 shows, depending on the request, the Verb Handler delivers a customizable static XML response (for *Identify*, *ListMetadataFormats*) or a dynamic page of results (for *GetRecord*, *ListIdentifiers*, and *ListRecords*). Results are transformed into the format of the OpenAIRE guidelines by a dedicated mapper function.
Two configuration files guide the delivery of the dynamic page of results:

- A suite of SPARQL queries to get the desired data from the SPARQL endpoint. For query optimization, two strategies were evaluated:
  - One single `CONSTRUCT` query to retrieve all the properties required to build the response.
  - Multiple `SELECT` queries: one SPARQL query to get the main properties of a resource and the URLs of related agents (contributors, creators, funders, publishers). Another `SELECT` query is dedicated to the retrieval of details about the agents.

  Depending on the model and the resources of the KG, a single `CONSTRUCT` query could be heavy to be executed, especially in case of small-sized RDF stores, a typical case for small-medium DH projects. The multiple `SELECT` approach, while slower (as multiple queries are needed to build the final metadata record), proved more robust for the DH-ATLAS Knowledge Graph, as decomposing retrieval into smaller queries reduced the risk of overloading the SPARQL endpoint.
- A set of dictionaries that support the mapping process in the selection of the proper field names according to the OpenAIRE format and for semantic alignment of controlled vocabularies, specifically for resource types and access rights.

Figure 4 shows the flowchart for a `GetRecord` OAI-PMH call. Given the identifier of a resource in the DH-ATLAS KG, the first `SELECT` query is executed. For each agent URI in the response `ResSet_temp{agent_ids}`, another query is sent to retrieve the details of the agent. The final response (`Result_def_agent`) is then constructed, mapped to the OpenAIRE guidelines and returned to the caller.

4.3. DH-ATLAS RESTful Application Programming Interface

The DH-ATLAS Knowledge Graph (KG) is programmatically accessible via RESTful APIs[27] implemented with the RAMOSE framework (Restful API Manager Over SPARQL Endpoints). [15, 25]

[27] The APIs and the relative documentation is available at https://services.d4science.org/group/atlas/dh-atlas-rest-api. The codebase is available at https://github.com/dh-atlas/api.

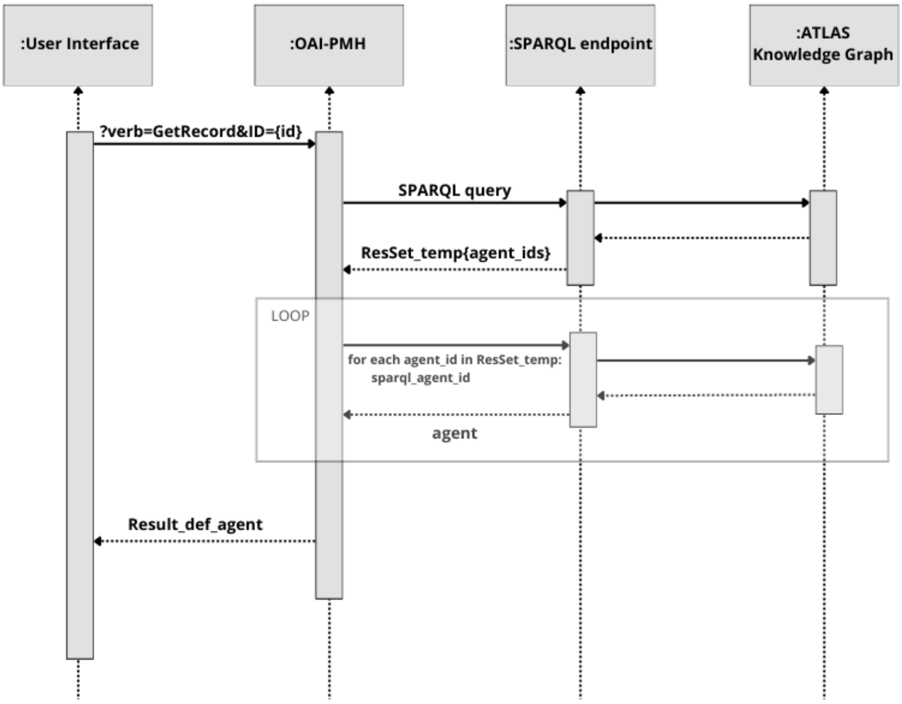

Figure 4: Flowchart of an OAI-PMH call `GetRecord` showing how the metadata retrieval is decomposed into multiple SPARQL SELECT queries

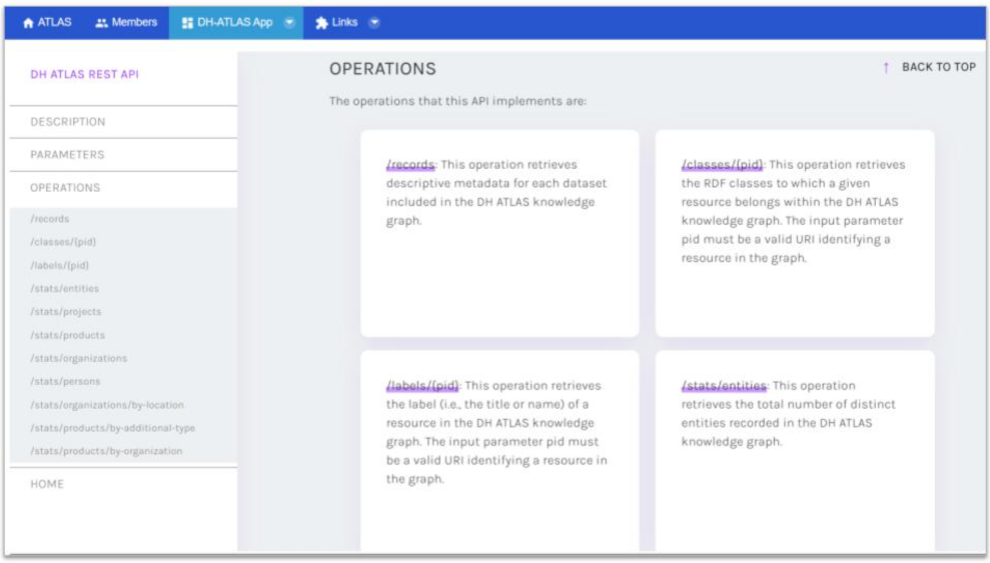

Figure 5: Figure displaying an excerpt of the RESTful API documentation made available through RAMOSE.

RAMOSE enabled us to configure the RESTful API with a declarative language, without requiring a full knowledge of both the RDF data model and the complexity of SPARQL. This API layer is intended to support machine-to-machine integration scenarios, enabling the development of dashboards with statistics, digital library portals, and data analysis tools. The system inherently offers several key benefits, including:

- RESTful access to RDF data with parameterized endpoints.
- JSON and CSV serialization of RDF results for easy consumption.
- Modular configuration for multiple endpoints.

- Compatibility with web applications and data portals.

Clients can interact with the DH-ATLAS API via simple HTTP GET requests, specifying any parameters in the path, and obtaining JSON structures containing both the data of interest and technical metadata (HTTP headers, status codes, etc.).

As noted previously, in the RAMOSE configuration file (`.hf` *hashformat* format),[28] each operation is defined declaratively; in terms of maintenance and sustainability, this means that we have great flexibility in the evolution of the API support. In detail, the configuration strategy needs to specify at least the following parameters per operation:

- The URL path for the end-point with parameters
- The type of operation and the allowed HTTP method
- The structure and format of the parameter
- The type of the SPARQL variables
- The associated SPARQL query
- A description of the operation

The current version of the API is documented within a Web application available through DH ATLAS Virtual Research Environment powered by the *D4Science* infrastructure.[29] It offers a set of operations covering three main dimensions: metadata retrieval, inspection of resources, and KG statistics.

As far as metadata retrieval is concerned, we implemented the `GET /records` service path. It returns metadata for the datasets in the DH-ATLAS KG. Clients can request the output in JSON or CSV format.

Two API paths are available for resource inspection: `GET /labels/{pid}` and `GET /classes/{pid}`. These methods are particularly useful for applications that need to navigate the DH-ATLAS graph without having a detailed knowledge of the underlying ontology. The first service path (`GET /labels/{pid}`) returns the human-readable label associated with the resource in the graph given its persistent identifier (*pid*), such as the title of a project or the name of an institution. For example, the path:
`labels/https://w3id.org/dh-atlas/1743001460-3508337`
returns:
`"label": "The Betrothed (PhiloEditor)"`.

The second service path (`GET /classes/{pid}`) returns the list of RDF classes the resource belongs to, allowing one to immediately understand its conceptual context (research project, dataset, digital edition, person, organization, etc.). For example, the path:
`/classes/https://w3id.org/dh-atlas/1743001460-3508337` returns:
`"classes": "https://schema.org/Dataset; https://w3id.org/dh-atlas/DigitalScholarlyEdition"`.

A set of operations are devoted to statistics and aggregate indicators over the DH-ATLAS KG:

- `GET /stats/entities`: returns the total number of modeled entities (projects, products, persons, organizations, etc.).
- `GET /stats/projects`: number of projects recorded in the graph.
- `GET /stats/products`: total number of censused research products.
- `GET /stats/persons`: number of persons recorded.
- `GET /stats/organizations/by-location`: distribution of organizations by geographic location, useful for territorial analyzes.
- `GET /stats/products/by-additional-type`: distribution of products by type (digital editions, corpora, software, LOD datasets, etc.), in line with the DH-ATLAS Ontology.
- `GET /stats/products/by-organization`: distribution of products by producing institution.

They support the realization of visualization and monitoring scenarios for the Italian DH ecosystem. Examples are interactive dashboards showing the productivity of individual institutions or the coverage of specific types of resources. Some operations also expose distributions along specific analytical dimensions (by location, by additional type, by organization).

[28]https://github.com/dh-atlas/api/tree/7a732914524ae7dba9ca8d102e4379eade11330b/v1/config
[29]DH-ATLAS API's end-point is available at: https://atlas-prin2022-api.cloud.d4science.org/api/v1

5. Conclusion

The coexistence of legacy protocols and modern semantic technologies in the digital library and scholarly communication ecosystem demands flexible and interoperable solutions. The tools described in this paper have enabled the DH-ATLAS project to make its knowledge graph compliant with diverse access requirements of external platforms, without duplicating data, while preserving its authoritative information in an RDF store.

Existing solutions such as RAMOSE, which enables the deployment of RESTful APIs over SPARQL endpoints, are valuable for integrating RDF data into modern web applications. They lower the barrier to accessing the DH-ATLAS knowledge graph by exposing a stable set of endpoints for metadata retrieval, semantic inspection, and basic statistics.

Our experience with RAMOSE in the DH-ATLAS project was positive: we could set up a working RESTful API with ease, simply by configuring the framework in terms of endpoints and SPARQL queries. Future developments can focus on incrementally enriching programmatic access to the KG with new services, while keeping the underlying infrastructure decoupled yet integrated. In particular, future API enhancements could support clients with operations designed to filter research products by different properties at once, such as type, period, institution, methodological approach. Additionally, they could facilitate retrieving products in relation to other products (e.g. editions derived from the same corpus or datasets that share a common ontology). In this way, we plan to support faceted navigation patterns that mirror typical scholarly workflows.[26]

However, the same approach could not be applied to the OAI-PMH protocol. We found no ready-to-use, configurable solution to activate an OAI-PMH publisher on top of a SPARQL endpoint. Consequently, we developed a minimum viable architecture and implementation tailored to the requirements of the DH-ATLAS project. Our current proposal does not yet fully address this gap. The OAI-PMH publisher we developed only partially implements the protocol in order to fulfill the operational requirement for being harvested by the OpenAIRE Graph. Support for OAI sets, Dublin Core, resumption tokens, and deleted records is missing. In particular, the latter two will likely require a significant revision of the design and architecture.

For the long-term sustainability of the DH-ATLAS suite, all services are being deployed on the infrastructure D4Science.org hosted by CNR-ISTI. A Virtual Research Environment that hosts the ATLAS App based on CLEF and the RESTful API is already available at https://services.d4science.org/group/atlas/home. Thanks to deployment on the D4Science infrastructure, the project's services and application can benefit from automatic monitoring, accounting, and compliance with latest authentication and authorization like OpenID Connect (OIDC).

Finally, the Knowledge Graph was deposited on CLARIN to ensure long-term persistence[27] and registered on the registry fairsharing.org[30]. These activities collectively ensured the visibility, scalability, interoperability, and long-term sustainability of the ATLAS digital ecosystem.

Acknowledgments

This work is partially funded by the European Union - Next Generation EU Mission 4 Component 1 CUP B53D23022220006.

CRediT author statement

AB: Conceptualization, Methodology, Writing – original draft, review & editing. RDG: Conceptualization, Methodology, Software, Writing – review & editing, AMDG: Conceptualization, Methodology, Software,

[30]FAIRsharing.org: DH-ATLAS; The Atlas of Digital Humanities Research on the Italian Cultural Heritage, FAIRsharing ID: https://fairsharing.org/7181, Last Edited: Friday, December 12th 2025, 20:35, Last Editor:alessia.bardi,Last Accessed: Friday, December 12th 2025, 20:36,Last Reviewed: Thursday, November 27th 2025, 12:23

Writing – original draft, review & editing. GR: Methodology, Formal analysis, Software, Writing – original draft, review & editing.

Declaration on Generative AI

During the preparation of this work, the author(s) used Copilot in order to: Paraphrase and reword, improve writing style. 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] K. Angele, M. Meitinger, M. Bußjäger, S. Föhl, A. Fensel, Graphsparql: a graphql interface for linked data, in: Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing, SAC '22, Association for Computing Machinery, New York, NY, USA, 2022, p. 778–785. URL: https://doi.org/10.1145/3477314.3507655. doi:10.1145/3477314.3507655.

[2] G. Vega-Gorgojo, Crafts: Configurable rest apis for triple stores, IEEE Access 10 (2022) 32426–32441. doi:10.1109/ACCESS.2022.3160610.

[3] M. Eskevich, F. de Jong, A. König, D. Fišer, D. V. Uytvanck, T. Aalto, L. Borin, O. Gerassimenko, J. Hajič, H. van den Heuvel, N. Kahusk, K. Liin, M. Matthiesen, S. Piperidis, K. Vider, Clarin: Distributed language resources and technology in a european infrastructure, in: Proceedings of LT4HALA 2020 - 1st Workshop on Language Technologies for Historical and Ancient Languages, European Language Resources Association (ELRA), Marseille, France, 2020, pp. 28–34.

[4] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne, J. Bouwman, A. J. Brookes, T. Clark, M. Crosas, I. Dillo, O. Dumon, S. Edmunds, C. T. Evelo, R. Finkers, A. Gonzalez-Beltran, A. J. Gray, P. Groth, C. Goble, J. S. Grethe, J. Heringa, P. A. 't Hoen, R. Hooft, T. Kuhn, R. Kok, J. Kok, S. J. Lusher, M. E. Martone, A. Mons, A. L. Packer, B. Persson, P. Rocca-Serra, M. Roos, R. van Schaik, S.-A. Sansone, E. Schultes, T. Sengstag, T. Slater, G. Strawn, M. A. Swertz, M. Thompson, J. van der Lei, E. van Mulligen, J. Velterop, A. Waagmeester, P. Wittenburg, K. Wolstencroft, J. Zhao, B. Mons, The fair guiding principles for scientific data management and stewardship, Scientific Data 3 (2016). URL: http://dx.doi.org/10.1038/sdata.2016.18. doi:10.1038/sdata.2016.18.

[5] A. Bardi, M. Buzzoni, M. Daquino, R. Del Gratta, A. M. Del Grosso, F. Fischer, S. Giacomini, C. Martignano, R. Rosselli Del Turco, G. Rubin, F. Tomasi, Fair digital humanities scholarly metadata. the atlas project, in: WOOC 2025 proceedings, Zenodo, 2025. URL: https://zenodo.org/records/16366693. doi:10.5281/zenodo.16366692.

[6] C. Martignano, G. Rubin, S. Giacomini, A. Bardi, M. Buzzoni, M. Daquino, R. Del Gratta, A. M. Del Grosso, F. Fischer, R. Rosselli Del Turco, F. Tomasi, Atlas: a data model for describing fair digital humanities research outcomes, in: Diversità, Equità e Inclusione: Sfide e Opportunità per l'Informatica Umanistica nell'Era dell'Intelligenza Artificiale, AlmaDL, University of Bologna Digital Library, Bologna, 2025, pp. 440–447. URL: https://amsacta.unibo.it/id/eprint/8380/. doi:10.6092/unibo/amsacta/8380.

[7] M. Daquino, A. Bardi, M. Buzzoni, R. Del Gratta, A. M. Del Grosso, F. Fischer, F. Tomasi, R. Rosselli Del Turco, The atlas: a knowledge graph of digital scholarly research on italian cultural heritage, in: Me.Te. Digitali. Mediterraneo in rete tra testi e contesti. Atti del XIII Convegno Annuale AIUCD2024, 2024, pp. 588–592. URL: https://amsacta.unibo.it/id/eprint/7927/.

[8] S. Giacomini, A. Bardi, M. Buzzoni, M. Daquino, R. Del Gratta, A. M. Del Grosso, F. Fischer, C. Martignano, R. Rosselli Del Turco, G. Rubin, F. Tomasi, Atlas: Towards a knowledge graph of international scholarly research on the italian digital cultural heritage, in: IRCDL 2025 proceedings, volume 3937, CEUR-WP, 2025. URL: https://ceur-ws.org/Vol-3937/paper4.pdf.

[9] S. Giacomini, M. Daquino, F. Tomasi, L. A. Fintoni, Clef 2.0. solutions for native linked data

cataloguing of italian digital cultural heritage, JLIS.it 16 (2025) 108–126. URL: https://www.jlis.it/index.php/jlis/article/view/611. doi:10.36253/jlis.it-611.

[10] V. A. Carriero, A. Gangemi, M. L. Mancinelli, L. Marinucci, A. G. Nuzzolese, V. Presutti, C. Veninata, ArCo: The Italian Cultural Heritage Knowledge Graph, in: The Semantic Web – ISWC 2019, Springer International Publishing, 2019, p. 36–52. URL: http://dx.doi.org/10.1007/978-3-030-30796-7_3. doi:10.1007/978-3-030-30796-7_3.

[11] S. La Bruzzo, P. Manghi, A. Bardi, Oaizer: Configurable oai exports over relational databases, in: E. Garoufallou, J. Greenberg (Eds.), Metadata and Semantics Research, Springer International Publishing, Cham, 2013, pp. 35–47.

[12] M. Artini, OAI-PMH Server v1.0.0, 2024. URL: https://doi.org/10.5281/zenodo.13938870. doi:10.5281/zenodo.13938870.

[13] A. Bardi, S. La Bruzzo, P. Manghi, Customized OAI-ORE and OAI-PMH exports of compound objects for the fedora repository, The Grey journal (Print) 9 (2013) 40–47.

[14] R. Gomola, M. Dvořáková, M. Růžička, Craft-oa deliverable 6.1_ojs connector for openaire research graph, 2024. URL: https://doi.org/10.5281/zenodo.12633203. doi:10.5281/zenodo.12633203.

[15] M. Daquino, I. Heibi, S. Peroni, D. Shotton, Creating RESTful apis over SPARQL endpoints using RAMOSE, Semantic Web 13 (2022) 195–213. URL: https://journals.sagepub.com/doi/abs/10.3233/SW-210439. doi:10.3233/SW-210439. arXiv:https://journals.sagepub.com/doi/pdf/10.3233/SW-210439.

[16] E. Daga, L. Panziera, C. Pedrinaci, Basil: A cloud platform for sharing and reusing sparql queries as web apis, in: CEUR Workshop Proceedings, volume 1486, 2015.

[17] A. Merono Penuela, R. Hoekstra, grlc Makes GitHub Taste Like Linked Data APIs, volume 9989 LNCS of *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer/Verlag, 2016, pp. 342–353. doi:10.1007/978-3-319-47602-5_48.

[18] D. Garijo, M. Osorio, Oba: An ontology-based framework for creating rest apis for knowledge graphs, in: International semantic web conference, Springer, 2020, pp. 48–64.

[19] G. Rubin, A. Bardi, DH ATLAS - OpenAIRE model mapping, 2025. URL: https://doi.org/10.5281/zenodo.17917277. doi:10.5281/zenodo.17917277.

[20] M. Assante, L. Candela, D. Castelli, R. Cirillo, G. Coro, L. Frosini, L. Lelii, F. Mangiacrapa, P. Pagano, G. Panichi, F. Sinibaldi, Enacting open science by d4science, Future Generation Computer Systems 101 (2019) 555–563. URL: http://dx.doi.org/10.1016/j.future.2019.05.063. doi:10.1016/j.future.2019.05.063.

[21] G. Rubin, A. M. DEL GROSSO, R. DEL GRATTA, A. Bardi, Dh-atlas: Rest api, 2025. URL: https://doi.org/10.5281/zenodo.17361169. doi:10.5281/zenodo.17361169.

[22] G. Rubin, R. DEL GRATTA, A. Bardi, A. M. DEL GROSSO, Dh-atlas: Oai-pmh publisher, 2025. URL: https://doi.org/10.5281/zenodo.17360920. doi:10.5281/zenodo.17360920.

[23] C. Martignano, A. Bardi, M. Buzzoni, M. Daquino, R. Del Gratta, A. M. Del Grosso, F. Fischer, S. Giacomini, R. Rosselli Del Turco, G. Rubin, F. Tomasi, DH ATLAS: White book v1.2, 2025. URL: https://doi.org/10.5281/zenodo.14925266. doi:10.5281/zenodo.14925266.

[24] M. Daquino, M. Wigham, E. Daga, L. Giagnolini, F. Tomasi, Clef. a linked open data native system for crowdsourcing, Journal on Computing and Cultural Heritage 16 (2023) 1–17. URL: http://dx.doi.org/10.1145/3594721. doi:10.1145/3594721.

[25] S. Peroni, M. Daquino, opencitations/ramose: Release 1.0.2, 2021. URL: https://doi.org/10.5281/zenodo.4585536. doi:10.5281/zenodo.4585536.

[26] M. Stocker, A. Oelen, M. Y. Jaradeh, M. Haris, O. A. Oghli, G. Heidari, H. Hussein, A.-L. Lorenz, S. Kabenamualu, K. E. Farfar, M. Prinz, O. Karras, J. D'Souza, L. Vogt, S. Auer, Fair scientific information with the open research knowledge graph, FAIR Connect 1 (2023) 19–21. URL: https://journals.sagepub.com/doi/abs/10.3233/FC-221513. doi:10.3233/FC-221513. arXiv:https://journals.sagepub.com/doi/pdf/10.3233/FC-221513.

[27] S. Giacomini, C. Martignano, G. Rubin, A. Bardi, M. Buzzoni, M. Daquino, R. Del Gratta, A. M. Del Grosso, F. Fischer, R. Rosselli Del Turco, F. Tomasi, DH ATLAS: Knowledge graph, 2025.

URL: http://hdl.handle.net/20.500.11752/ILC-1040, ILC-CNR for CLARIN-IT repository hosted at Institute for Computational Linguistics ”A. Zampolli”.