technologies

Article

Adopting MOD-API in a Modern Dataset Catalog Platform: Opportunities, Challenges and Limitations

Manuel Fiorelli *, Paolo Bocciarelli and Armando Stellato

Department of Enterprise Engineering, Tor Vergata University of Rome, Via del Politecnico 1, 00133 Roma, Italy; paolo.bocciarelli@uniroma2.it (P.B.); stellato@uniroma2.it (A.S.)
* Correspondence: manuel.fiorelli@uniroma2.it

Abstract

As data exploitation continues to demonstrate its value, ontologies, thesauri, and other semantic datasets are increasingly recognized for enabling semantically meaningful data integration across disparate domains. With the proliferation of dataset catalogs, the MOD ontology (Metadata for Ontology Description and publication) was adopted, and an associated API was developed to support the future European Open Science Cloud (EOSC). Their aim is to harmonize catalogs of semantic datasets with respect to metadata vocabularies and access mechanisms, thereby ensuring compliance with the FAIR principles. Within an implementation action involving developers of prominent dataset catalogs, we were selected to integrate the MOD-API into ShowVoc, our platform for publishing and consuming ontologies, thesauri, lexicons, and other Semantic Web datasets. However, ShowVoc already relied on an expressive metadata model, the MDR (acronym for "Metadata Registry"), named after the component responsible for managing the platform's internal catalog. Due to precise dissemination requirements, the MDR provides multiple abstraction levels and detailed specifications concerning the distributions and formats in which a dataset may be made available. In this article, we report on the challenges that we faced and the trade-offs that we made while reconciling these metadata models, highlighting limitations in the current MOD standard that may inform future enhancements.

Keywords: MOD; MOD-API; ShowVoc; DCAT; semantic artefact catalog; metadata

1. Introduction

Data collection and analysis have long been two fundamental components of the scientific process, supporting both the formulation and the validation of hypotheses. The advent of computers has significantly enhanced analytical capabilities, increasing both the resolution and the speed of data processing, while computer-based simulations have progressively complemented traditional empirical methods. In the early 2000s, however, the emergence of Big Data triggered a paradigm shift toward exploratory, data-intensive science, characterized by the use of large-scale statistical techniques and data mining approaches [1]. The full implications of this shift are still not completely understood and continue to be debated within the scientific community [2,3].

Lowering the barriers to data reuse across seemingly disparate domains is critical to modern data-driven science and is one of the key motivations behind the formulation of the FAIR principles [4,5], which offer guidance to data stewards to ensure that relevant data can be found (F), accessed (A), interoperated (I) with and, ultimately, reused (R). A common means to comply with these principles is to publish data, tools, and services on established

Check for updates

Academic Editor: Sikha Bagui

Received: 27 January 2026
Revised: 14 March 2026
Accepted: 18 March 2026
Published: 23 March 2026

catalogs, which enable prospective users to discover relevant resources more effectively. To achieve this, catalogs are often designed to structure the metadata that describes their items, relying on ontologies, thesauri, and other semantic datasets for this purpose. Since these semantic datasets are also required to be FAIR, the adoption of semantic dataset catalogs has increased, with metadata-related features and access mechanisms emerging as two prominent dimensions for their assessment [6].

The harmonization, if not full federation, of semantic dataset catalogs is necessary to overcome the proliferation of competing catalogs and ensure semantic interoperability across adjacent scientific communities. Adapting the categorization proposed in [7], we illustrate the main approaches to achieve this in Figure 1. These approaches differ in whether the burden of achieving, and indeed committing to, interoperability falls entirely on a third party wishing to expose content from multiple catalogs, on the catalogs themselves, or somewhere in between.

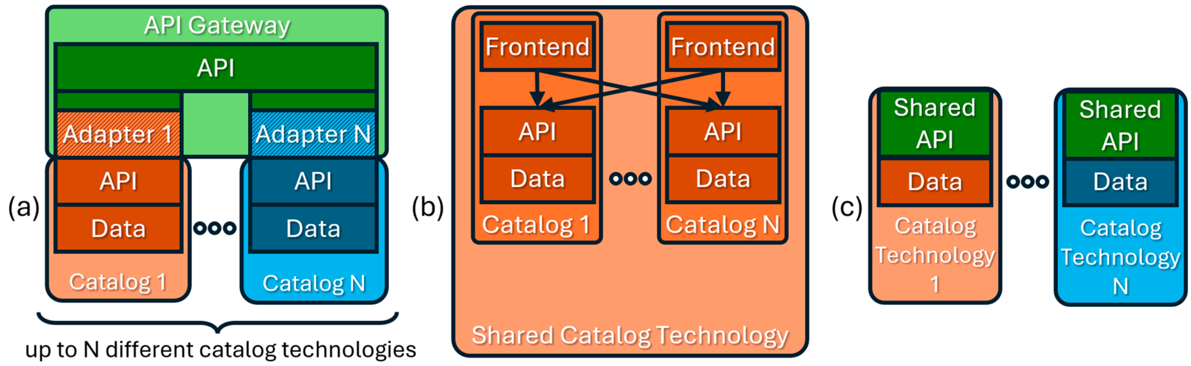

Figure 1. Approaches for semantic dataset catalog interoperability: (**a**) API gateway; (**b**) frontend federation of semantic dataset catalogs sharing the same backend API; (**c**) implementation of a shared API.

The API gateway approach (a) places the entire burden on the gateway owner, who must implement adapters for each catalog technology. This strategy is adopted, for example, by the TS4NFDI's API Gateway [8], which retrieves metadata from different catalogs using dedicated adapters that harmonize the retrieved metadata. The aggregated catalogs may not even be aware of this effort, as it is performed entirely by the gateway, which manages every aspect of the process. Therefore, this strategy imposes a high cost on the party operating the gateway as the number of different technologies increases, and it may be susceptible to failure whenever non-backward-compatible changes are introduced, potentially breaking the adapters. The frontend federation approach (b), exemplified by the OntoPortal Federation [7], requires each catalog frontend to interact not only with its own backend but also with the backends of multiple federated catalogs that rely on the same technology and thus expose the same API. This strategy is indeed effective, but its scope is inherently limited because it cannot support the federation of catalogs built on heterogeneous technologies. Finally, the shared-API approach (c) requires each catalog to implement the same agreed-upon API. This strategy clearly shifts the burden of interoperability to the catalogs themselves, which must commit to the shared API and take responsibility for its implementation. Without requiring any additional harmonization or federation infrastructure, this approach allows serendipitous access to any compliant catalog by clients that happen to encounter it. Nonetheless, it is also compatible with deploying a gateway over a predefined set of catalogs, with sustainability and robustness issues mitigated by the commitment of catalogs to a shared, stable API. In fact, the latter may also enable a minimal degree of frontend federation, regardless of the underlying technology stack of the catalogs.

These interoperability concerns clearly need to be addressed in the ongoing construction of the European Open Science Cloud (EOSC) [9], which aims to "provide researchers and innovators in Europe with an open and trusted multi-disciplinary environment where they can publish, find and reuse data, tools and services for research and innovation". Frontend federation clearly falls short in this context, as the adoption of a single catalog technology is not attainable in such an open and diverse environment. Similarly, a pure API gateway approach would be overly brittle and costly to maintain. By contrast, the shared-API approach addresses these challenges at their root, as it requires different catalogs to commit to a shared metadata model and shared access mechanisms.

The FAIR-IMPACT [10] project is helping make the envisioned EOSC concrete through the development of the MOD-API [11,12], which builds on the MOD (Metadata for Ontology Description and publication) [13] ontology. The MOD ontology has undergone several iterations, informed both by the practices adopted by actual catalogs and by the consensus reached among stakeholders who participated in several workshops. Widespread adoption of the MOD-API by the community is now crucial to realizing its value proposition. FAIR-IMPACT supported progress in this direction through an open call to engage catalog developers interested in implementing the API [14]. As developers of ShowVoc [15], a platform for publishing ontologies, thesauri, lexicons, and Semantic Web datasets in general, and which can be considered a generic catalog technology, we applied to the open call with a proposal to implement MOD-API within ShowVoc, which was eventually selected.

ShowVoc already relied on a sophisticated metadata model called MDR [16,17] (Metadata Registry), named after the component that manages the platform's internal dataset catalog. The model relies on DCAT and extends it by providing a ladder of abstractions that captures a semantic dataset per se, independently of its multiple realizations, each of which may be expressed through several distributions and delivered via different access channels. Because the model captures dimensions not covered by the MOD ontology (and even less by the subset of MOD exposed through the MOD-API), replacing it entirely with the MOD ontology or even realizing a full one-to-one mapping was not viable. We embraced the challenge by examining how the two models could be aligned, identifying semantic mismatches and potential limitations, assessing whether these issues might affect the goal of implementing the MOD-API, and, where necessary, determining how they could be mitigated. While we value the specificity of the MDR, we are not advocating it over the MOD ontology and its API. It is also not our aim to motivate the design of the latter. We take both as given and, acknowledging our role in supporting these emerging standards, we adopted them to provide a complementary access layer to the catalog managed by ShowVoc. In doing so, we help ensure that ShowVoc can continue to thrive within the future EOSC ecosystem, should the MOD-API ultimately succeed in delivering the intended interoperability across semantic dataset catalogs.

This work is structured as follows: Section 2 illustrates related work; Section 3 provides background on the MOD ontology and the MDR metadata model that already powers ShowVoc; Section 4 describes the MOD-API; Section 5 contrasts the two metadata models (in Section 5.1), illustrates the approach to integrate the MOD-API within ShowVoc (in Section 5.2), and explores how the two models can be reconciliated (in Section 5.3); Section 6 provides more implementation details; Section 7 addresses the verification and validation of the proposed solution; Section 8 offers a discussion; and Section 9 concludes the work.

2. Related Work

The Semantic Web is an extension of the document-oriented Web in which "information is given well-defined meaning, better enabling computers and people to work in cooperation" [18]. The Linked Open Data paradigm (https://www.w3.org/DesignIssues/

LinkedData.html, accessed on 6 March 2026) clarified the roots of the Semantic Web by grounding it in the same architectural principles that made the original Web successful, evolving it into a global data space [19] that supports the publication, discovery, and meaningful integration of datasets originating from unfamiliar and distant domains [20,21]. In this regard, the primary mechanism for information retrieval is the follow-your-nose strategy, which consists of traversing links between resources, possibly jumping from one dataset to another, each published independently.

Nevertheless, the need emerged for a coarse-grained perspective of this *giant global graph*, one that acknowledges the boundaries and identities of datasets, their general attributes (including licensing, provenance, and theme), their structure, metrics, and the range of access mechanisms (beyond HTTP dereferencing), as well as their interrelations. This need motivated the development of the VoID [22,23] (Vocabulary of Interlinked Datasets), which enables the description of linked datasets. VoID also standardized the metadata discovery process, including a scenario in which each resource within a dataset is linked to the VoID description of that dataset, which can be found by anyone who comes across it by "following their nose". A concrete example of VoID in practice is the VoID description [24] of AGROVOC [25], the multilingual thesaurus developed by the Food and Agriculture Organization (FAO) of the United Nations, comprising more than 41,000 concepts across 42 languages and covering domains such as food, nutrition, agriculture, fisheries, forestry and environment.

VoID has also been significantly extended by the community. As previously noted, VoID supports dataset metrics, including the number of triples, subjects, and similar indicators. It also allows describing partitions of a dataset defined according to specific criteria, such as identifying the subset of resources belonging to a given class. The void-ext [26] extension broadens the scope of statistical descriptions by introducing additional metrics (e.g., the average subject IRI length) together with new partitioning criteria (e.g., subsets defined by resources appearing in predicate or object position). VOAF [27] is another extension focused on ontologies and other vocabularies used in the LOD cloud. Finally, LIME [28] provides a vocabulary for describing language resources published as LOD [29], as well as linguistic information available within ontologies, thesauri, and datasets.

Despite the distributed publication model at the heart of the Web, it is undeniable that most interactions with it now begin through Google and similar search engines, which have effectively become gatekeepers to the Web. The situation worsened with the rise of conversational agents such as ChatGPT and Microsoft Copilot, which retrieve, analyze, and re-elaborate information on behalf of users, motivating new concerns about anticompetitive behavior [30] and contributing to the rise in zero-click searches that threaten the open web [31]. Without search engines with comparably dominant positions, the Semantic Web, possibly fueled by the widespread adoption of the FAIR principles, has seen a surge in the number and relevance of semantic dataset catalogs. These range from the LOD Cloud [32] to Linked Open Vocabularies [33,34] (LOV), along with a multitude of domain-specific catalogs, such as those built on the OntoPortal [35,36] software stack (e.g., AgroPortal [37,38], EcoPortal [39,40], BiodivPortal [41,42], EarthPortal [43,44], MedPortal [45], TechnoPortal [46]), originally developed for BioPortal [47,48].

The reach of the Semantic Web in data catalogs is broader, as "more research on combining data catalogs with data models of high semantic expressiveness is needed" [49].

DCAT [50] (Data Catalog Vocabulary) enhances interoperability between dataset catalogs through a shared RDF vocabulary that describes datasets (i.e., dcat:Dataset), their accessible forms (i.e., dcat:Distribution), such as downloadable files or data services (i.e., dcat:DataService), as well as the catalog itself (via dcat:Catalog) and the dataset records (via dcat:CatalogRecord). Figure 2 provides a high-level overview of the DCAT model as presented in its third revision. Indeed, data services were added only in DCAT 2 [51], when its scope was broadened beyond cataloging datasets exclusively.

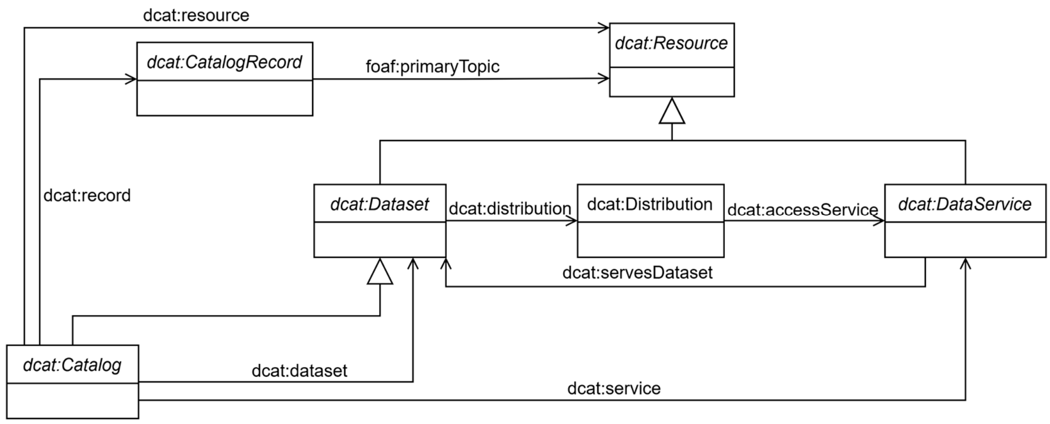

Figure 2. An excerpt of the DCAT 3 model. This figure omits most metadata properties, as well as several additional classes (e.g., dcat:DatasetSeries) and relations.

Unlike VoID, which specializes in the description of RDF datasets, DCAT can describe catalogs comprising resources of any kind, thus serving as a foundation for interoperability between catalogs in general. Application profiles of DCAT have been proposed to capture community practices and to address requirements not covered by the core model available at any given time. For example, ADMS [52] and the HCLS Community Profile [53] addressed versioning, which has since been incorporated into DCAT 3.

Figure 3 illustrates a portion of the ADMS model, showing how the main concepts (asset repository, asset, and asset distribution) correspond to elements in DCAT, and demonstrating the simple yet effective approach used to model different versions of an asset as a linked list of adms:Asset resources, one for each version.

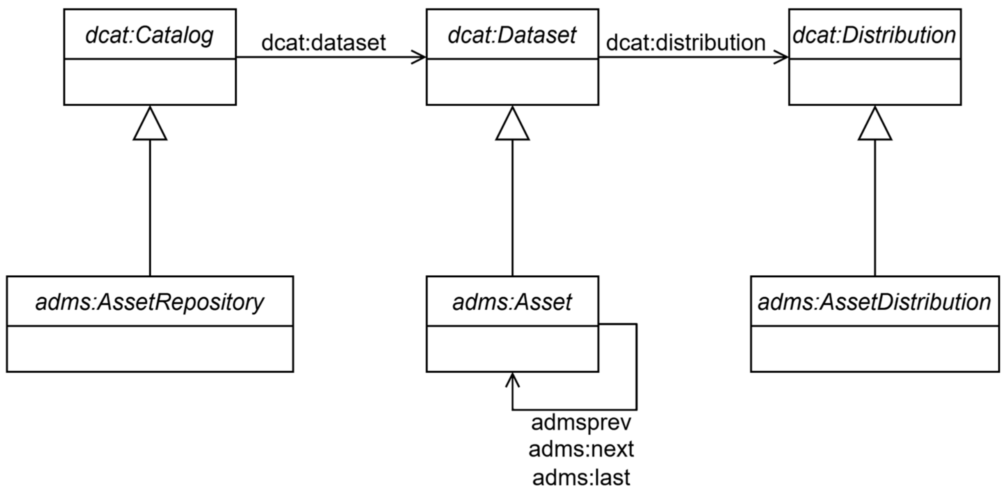

Figure 3. An excerpt of the ADMS model. This figure illustrates the reliance on DCAT in the original ADMS specifications managed by the W3C, as well as the proposed versioning properties used to establish a chain between asset versions.

The figure refers to the now-retired W3C specifications; ADMS has since transitioned under SEMIC [54] and has been decoupled from DCAT, which continues to play a significant role in ADMS use cases (https://semiceu.github.io/ADMS/releases/2.00/#interpretation-of-adms-in-the-context-of-dcat, accessed on 6 March 2026).

As shown in Figure 4, HCLS introduces a summary-level description that characterizes a dataset independently of its evolution over time. Each version is explicitly linked both to the summary-level description (following a hierarchical approach) and to the preceding version (following a chaining approach similar to ADMS).

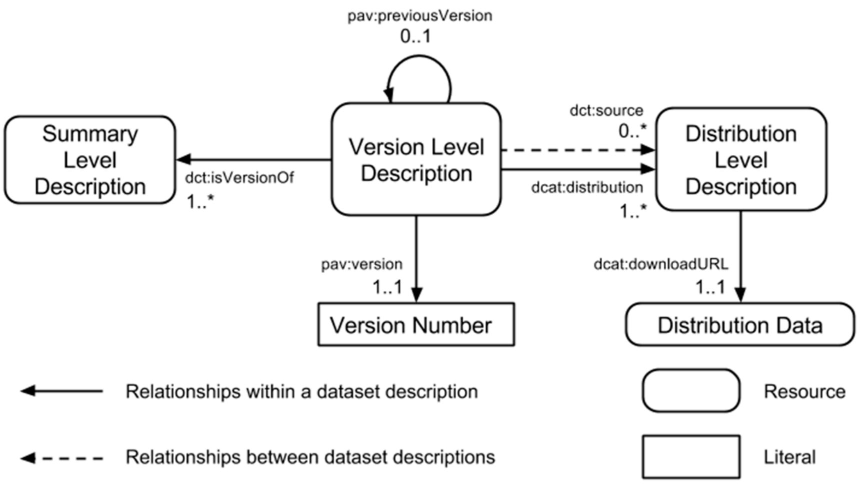

Figure 4. HCLS profile. The figure adopts the usual conventions for expressing the cardinality of relations, where 0..1 means zero or one, 0..* means zero or more, 1..1 means exactly one, and 1..* means one or more. Source: https://www.w3.org/TR/hcls-dataset/. Copyright © (https://www.w3.org/policies/#Copyright) 2015 W3C® (https://www.w3.org/) (MIT (https://www.csail.mit.edu/), ERCIM (https://www.ercim.eu/), Keio (https://www.keio.ac.jp/en/), Beihang (https://ev.buaa.edu.cn/)). W3C liability (https://www.w3.org/policies/#Legal_Disclaimer), trademark (https://www.w3.org/policies/#W3C_Trademarks) and document use (https://www.w3.org/copyright/document-license-2023/) rules apply. The source document was published as an Interest Group Note. All URLs accessed on 6 March 2026.

Heavily influenced by these works, DCAT 3 supports both hierarchical and chaining representations of versions—see Figure 5. Notably, DCAT 3 introduces properties in its own namespace for the relations isVersionOf and hasVersion, since the seemingly corresponding properties in Dublin Core Metadata Terms have a much broader scope: they can describe any kind of versioning relationship, including, for example, considering an animated film adaptation as a version of the original book.

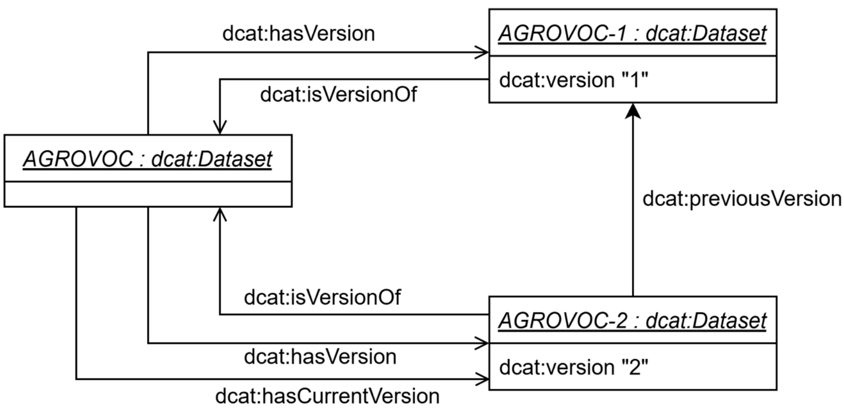

Figure 5. Versioning in DCAT 3.

In 2018, Google launched its Dataset Search service [55], which harvests datasets available on the Web by exploiting associated metadata represented using a variety of formats, including DCAT, schema.org [56], and CSVW (CSV on the Web) [57].

DCAT is, in general, non-prescriptive, accommodating different and occasionally contrasting detailed modeling approaches, while delegating to interested communities the task of defining shared cataloging practices encoded as application profiles. One notable example is DCAT-AP [58], which has been defined to increase interoperability among data catalogs in the European Union. One particularly contentious matter is how DCAT should be combined with VoID when structuring catalogs of RDF datasets, a question that has led to divergent models, such as those adopted in the aforementioned HCLS profile, as well as in DataID [59] and our own MDR.

3. Metadata Models

In this section, we first describe the MOD (Metadata for Ontology Description and publication) ontology, including a brief overview of its evolution. We then illustrate the MDR, discussing its approach to combining DCAT with VoID, as well as highlighting its reliance on additional vocabularies such as LIME.

3.1. MOD (Metadata for Ontology Description and Publication)

First proposed [60] in 2015, the MOD ontology has since been maintained within the RDA's (Research Data Alliance) [61] Vocabulary Services Interest Group [62] and subsequently adopted by the FAIRsFAIR [63] and FAIR-IMPACT European projects, which have positioned it as a building block of the future EOSC. The MOD ontology has undergone several revisions, reflecting the inherent complexity of defining a concise yet effective metadata model capable of meeting the needs of diverse stakeholders. We summarize the evolution of MOD to provide insight into its underlying methodology, main design decisions, and rationale.

The development of MOD 1.2 [64] was informed by an analysis of metadata vocabulary usage across 13 "ontology libraries". The inclusion of descriptive properties in the model was guided by several principles: brevity, clarity, simplicity, authority, standardization, extensibility, usability, and interoperability. In 2018, MOD 1.4 had 128 properties, each representing a cluster of properties drawn from 15 metadata vocabularies that were mapped to one another. MOD became a "maximal metadata model" that satisfied the need for comprehensive coverage, yet lacked guidance on how to prioritize metadata properties. MOD 2.0 [65] greatly enhanced interoperability by defining its main concepts on top of the foundation provided by DCAT. The lack of prioritization among the properties was addressed by indicating which FAIR principles they support, the relevant MIRO [66] guidelines, and, above all, the requirement level in the FAIRsFAIR minimum metadata profile. The latter was defined through the consensus of 30 participants representing about 20 communities who convened at a workshop held in June 2021. Finally, FAIR-IMPACT developed MOD 3 as an application profile of DCAT 2.

Figure 6 provides an overview of the MOD ontology, focusing on its main entities: the semantic artefacts and their distributions, as well as the semantic artefact catalog itself and the records therein. As illustrated, these entities derive from their corresponding DCAT classes, except for the semantic artefact, which rather extends dcat:Resource to emphasize a distinction from datasets conceived as factual data. Nonetheless, since semantic artefacts are typically linked to one or more distributions, they can still be regarded as instances of the class dcat:Dataset. This was a compromise between different propositions that emerged during the development of the third revision of MOD.

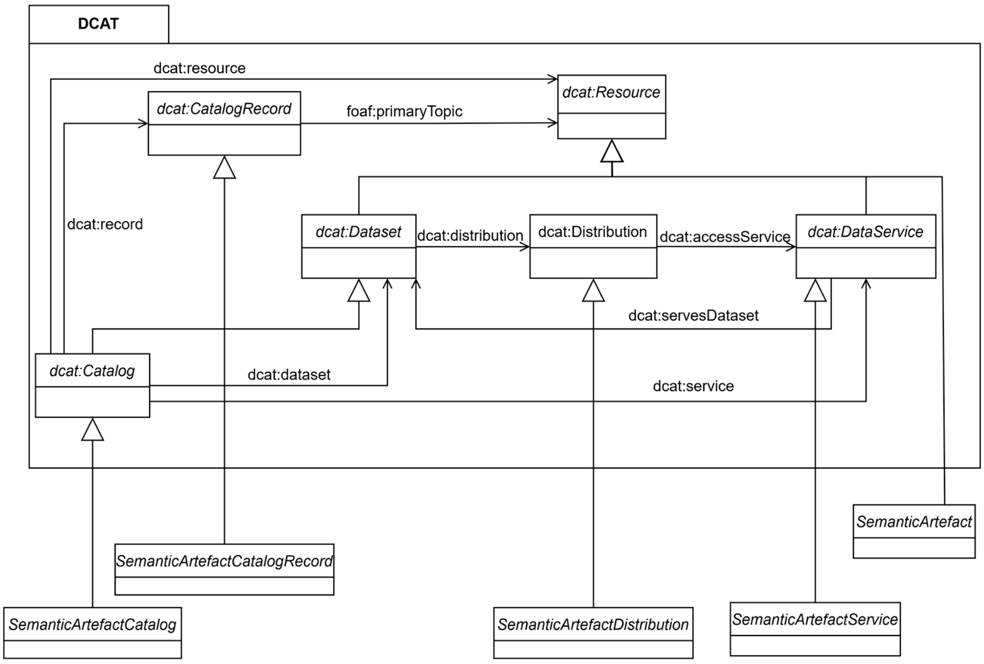

Figure 6. Structure of the MOD ontology (shown without the mod: prefix). This figure omits the metadata properties in the MOD ontology and focuses on a subset of the classes and properties in the DCAT vocabulary required for illustration.

MOD enriches DCAT by adding numerous metadata descriptors drawn from existing vocabularies whenever appropriate. Table 1 shows that MOD adopts 149 properties for the description of the main entities discussed earlier.

Table 1. Distribution of the properties adopted in MOD 3.2 across metadata vocabularies .

	Vocabulary	Count
	https://w3id.org/mod	40
	http://purl.org/dc/terms/	33
	http://omv.ontoware.org/2005/05/ontology	15
	http://www.w3.org/TR/vocab-dcat/	13
	http://schema.org/	8
	http://vocab.deri.ie/void	7
	http://www.w3.org/2002/07/owl#	5
	http://xmlns.com/foaf/0.1/	5
	http://purl.org/vocab/vann/	4
	http://usefulinc.com/ns/doap#	3
	http://www.w3.org/ns/prov-o#	3
	http://purl.org/pav/	3
	http://purl.org/vocommons/voaf	2
	http://creativecommons.org/ns#	2
	http://www.w3.org/2000/01/rdf-schema#	1
	http://www.w3.org/2004/02/skos/core	1
	http://www.w3.org/ns/odrl/2/	1
	http://www.w3.org/TR/vocab-adms	1
	http://www.w3.org/ns/sparql-service-description	1
	https://w3id.org/mod/2.0	1
total	20	149

The properties are drawn from 20 vocabularies: 18 of them are widely used, standard vocabularies, as required by the FAIR principles. The remaining two correspond to MOD itself, which defines 41 properties (28%). These are not necessarily new properties; in some cases, they originate from vocabularies that are no longer maintained, which requires MOD to redefine them locally. Unsurprisingly, the DCMI Metadata Terms [67] is the second most adopted vocabulary, followed, with a noticeable gap, by OMV [68] (Ontology Metadata Vocabulary) and DCAT, each contributing roughly half as much. The breadth of available properties is illustrated in Figure 7, which shows that only 41 of the 149 properties are part of the FAIRsFAIR minimum metadata profile. Moreover, only 19 properties are mandatory, representing 13% of the total and 46% of the FAIRsFAIR profile, thereby making the metadata curation more manageable. Additionally, 13 properties (representing 9% of the total and 32% of the FAIRsFAIR profile) are still recommended for inclusion.

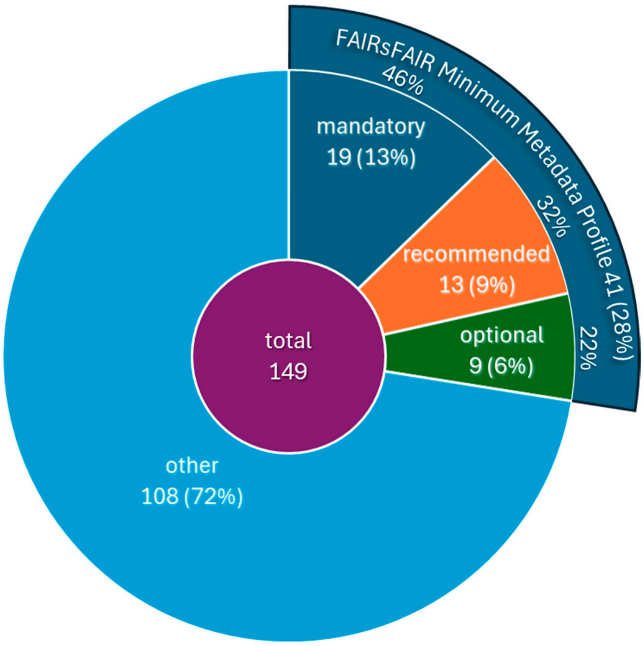

Figure 7. Properties included in the MOD 3.2 ontology.

Figure 8 breaks down these numbers by showing how many properties in the minimum metadata profile fall into each of the three requirement levels, across the main entities defined by the MOD ontology.

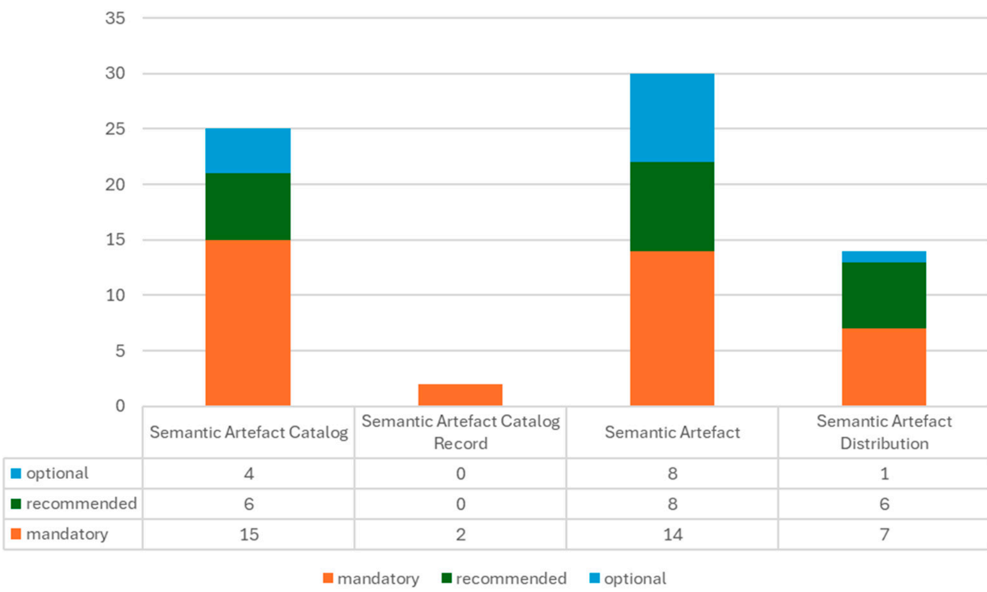

Figure 8. Distribution of the properties in the minimum metadata profile across the main entities defined by the MOD ontology.

3.2. MDR (Metadata Registry)

The MDR (Metadata Registry) ontology (Figure 9) extends DCAT in a more fine-grained manner than MOD, introducing additional semantic distinctions and integrating the VoID vocabulary to describe linked RDF datasets. Its development was primarily motivated by the need to enhance *machine actionability* across various Semantic Web application scenarios. Among these, ontology matching is a prominent example. In this context, we initially developed a metadata-driven platform [69] to support the semi-automatic configuration of ontology matching workflows, potentially leveraging existing language resources. Subsequent improvements [70,71] regarding the standardization of the LIME vocabulary and better integration within our ecosystem led the MDR to depend on the LIME extension of VoID.

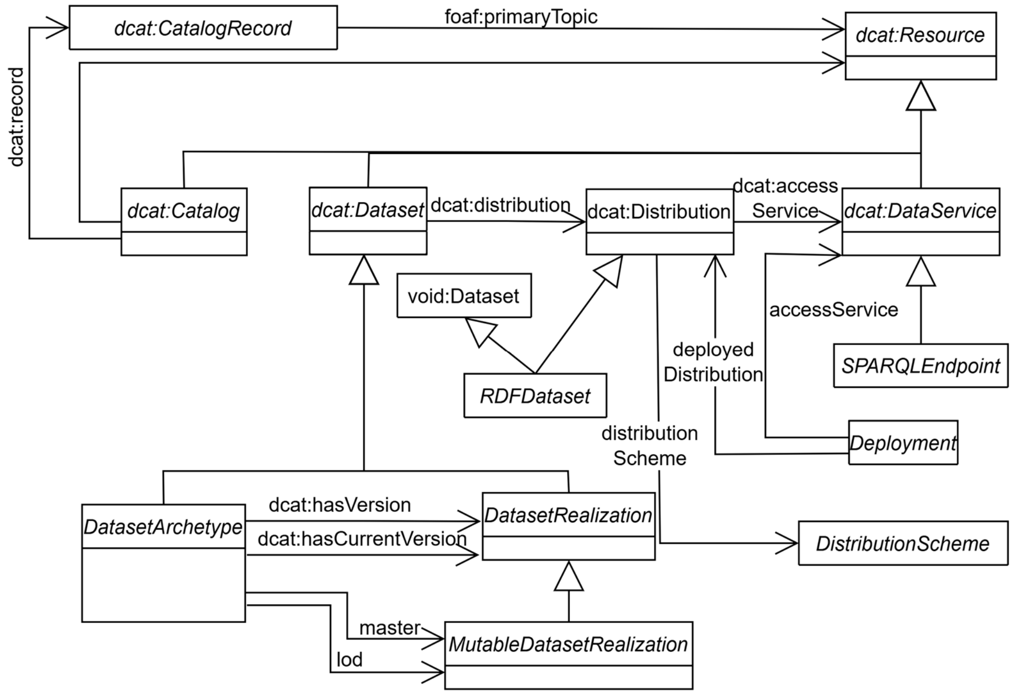

Figure 9. Structure of the MDR ontology (shown without the mdr: prefix). This figure omits additional classes brought in by importing LIME: mainly subclasses of void:Dataset to represent subsets (linked by the property void:subset) of an RDFDataset related to linguistic content.

The class dcat:Dataset is specialized into two subclasses: DatasetArchectype and DatasetRealization. The former captures a dataset's persistent identity as it evolves over time, while the latter represents a concrete instantiation of the dataset, without yet detailing its specific distributions. A dataset realization can correspond to a particular revision of the dataset at a given moment, linked through the properties dcat:hasVersion and dcat:hasCurrentVersion, and abstracts over multiple possible distributions. They may vary not only in their serialization format (e.g., RDF/XML vs. Turtle), expressed through the property dcat:mediaType (potentially combined with dcat:compressFormat to indicate a compressed container), but also in how triples are structured (for example, whether SKOS-XL reified labels are accompanied by plain SKOS counterparts). This variability is handled by linking each distribution to the appropriate instance of the class DistributionScheme via the property distributionScheme.

A MutableDatasetRealization can instead represent either the development copy of a dataset (usually deployed in a knowledge-engineering environment) or its LOD publication made accessible through dereferenceable URIs. These are linked to the underlying archetype via the master and lod properties, respectively.

Both a development environment and a triple store are then modeled as instances of dcat:DataService, and they typically serve a special MDR-defined distribution type called RDFDataset. This class, which extends both void:Dataset and dcat:Distribution, captures the abstract notion of a set of triples, independent of its serialization. It is therefore the place where most VoID metrics should reside and the source from which other file-based distributions can be generated, with PROV-O metadata documenting the derivation process. The Deployment class was also introduced to further detail how a data service is linked to a distribution, for instance, by indicating, through the locator property, where a specific distribution can be found within a service that hosts multiple ones. The MDR also provides a subclass of dcat:DataService, named SPARQLEndpoint, to represent data services that are compliant with the SPARQL protocol.

Figure 10 illustrates the extended dataset hierarchy of the MDR, which also introduces two additional subclasses of dcat:Dataset to represent linksets (between datasets) and changesets (typically between two RDFDataset instances associated with different versions of the same dataset archetype). These classes are further specialized into artefact-level and realization-level subclasses, following the general logic described earlier. Finally, each of them also includes a dedicated subclass of RDFDataset.

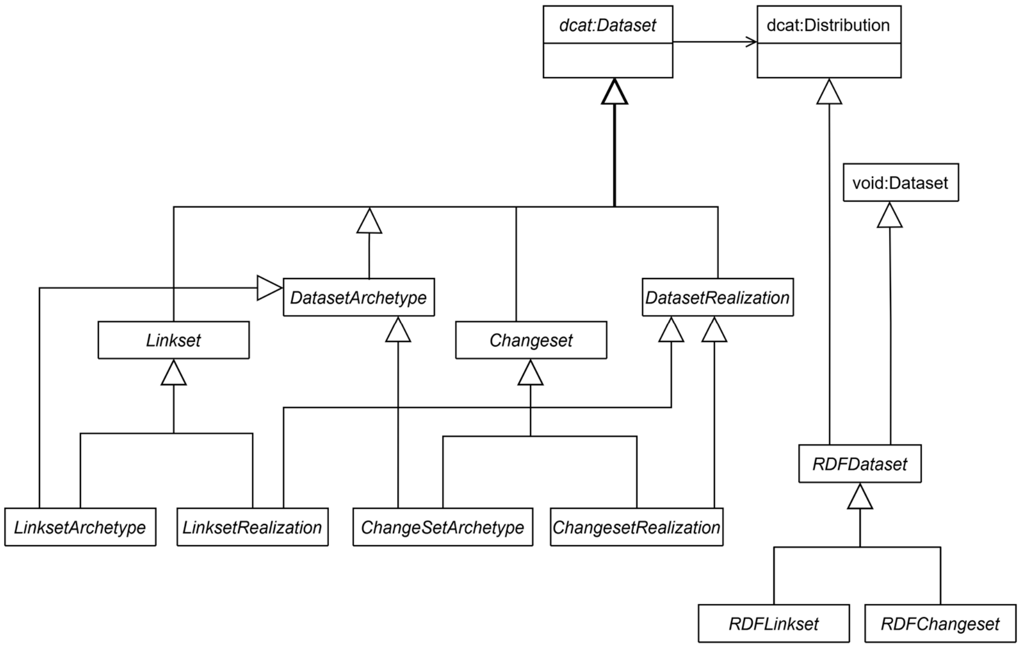

Figure 10. Extended dataset hierarchy in the MDR.

Since an RDFDataset is a void:Dataset, the full range of VoID features can be used to describe it. These include various metrics, as well as the ability to specify subsets through the void:subset property). Moreover, the MDR relies on LIME, which extends VoID with additional vocabulary for modeling specialized subsets related to linguistic content.

Figure 11 illustrates some of the possibilities enabled by combining VoID with the LIME extension, using the EuroVoc [72,73] thesaurus as an example. The complete description is available online in the void.ttl file [74]. We first observe the use of VoID to report basic metrics, such as the number of triples (4,211,719), entities (7615), distinct subjects (472,321), and distinct objects (956,233). VoID class partitions are also used to indicate the number of concept schemes (129), concepts (7486), and collections (0). Beyond a simple rdfs:label, the description includes a dcterms:title and uses dcterms:conformsTo to specify the semantic model, in this case SKOS.

```
<http://eurovoc.europa.eu/void.ttl#EuroVoc_4.23> a dcat:Distribution,
mdr:RDFDataset;
  void:triples 4211719;
  void:distinctSubjects 472321;
  void:distinctObjects 956233;
  dcterms:conformsTo <http://www.w3.org/2004/02/skos/core>;
  mdr:distributionScheme<https://op.europa.eu/dis-
trScheme/SKOS_distScheme> ;
  void:entities 7615;
  void:classPartition [
    void:class <http://www.w3.org/2004/02/skos/core#Concept>;
    void:entities 7486
  ],[
    void:class <http://www.w3.org/2004/02/skos/core#ConceptScheme>;
    void:entities 129
  ],[
    void:class <http://www.w3.org/2004/02/skos/core#Collection>;
    void:entities 0];
  void:subset <http://eurovoc.europa.eu/void.ttl#EuroVoc_de_lexicaliza-
tion_set>, [...] ;
  dcterms:title "EuroVoc";
  rdfs:label "EuroVoc";
  [...] .

<https://op.europa.eu/distrScheme/SKOS_distScheme>  a mdr:Distribu-
tionScheme ;
 rdfs:comment "The base RDF version of the dataset with the core set of
elements. This is the version of the dataset that it should be used in
most cases." .


<http://eurovoc.europa.eu/void.ttl#EuroVoc_de_lexicalization_set> a
lime:LexicalizationSet;
  lime:lexicalizationModel <http://www.w3.org/2008/05/skos-xl>;
  lime:referenceDataset <http://eurovoc.europa.eu/void.ttl#EuroVoc>;
  lime:lexicalizations 18368;
  lime:references 7613;
  lime:avgNumOfLexicalizations 2.413;
  lime:percentage 1.0;
  lime:language "de";
  dcterms:language <http://lexvo.org/id/iso639-3/deu>,
<http://id.loc.gov/vocabulary/iso639-1/de> .

[...]
```

Figure 11. Example description of an RDFDataset.

The RDF dataset is linked to multiple subsets via the void:subset property. Among these, we focus on one type, lime:LexicalizationSet, a subclass of void:Dataset designed to describe the lexicalization of a reference dataset in a given natural language, using a lexicalization model and, when applicable, a separate lexicon resource. In this example, all resources in EuroVoc have German values for the rdfs:label property, with an average of 2.413 labels per resource.

4. MOD-API

The MOD-API is structured as a resource-oriented API, where each resource type (and its collections), corresponding to specific classes in the MOD ontology, is assigned a dedicated URL. These URLs can be accessed via standard GET requests using the HTTP protocol to obtain the representation of the associated resource.

Figure 12 shows the resources related to artefacts. The first represents the complete collection of artefacts, while the second corresponds to a specific artefact, expressed as a URI template parameterized by the artefact identifier in the catalog. The subpaths introduced under/distributions follow the same logic, identifying the distributions associated with the artefact referenced in the preceding part of the URL. The/distribution/latest path provides a permanent address for retrieving the latest distribution of an artefact. This distribution is special, as it is the one assumed to be accessed through the/resources subpaths, enabling the browsing of classes, properties, and other types of entities depending on the nature of the artefact.

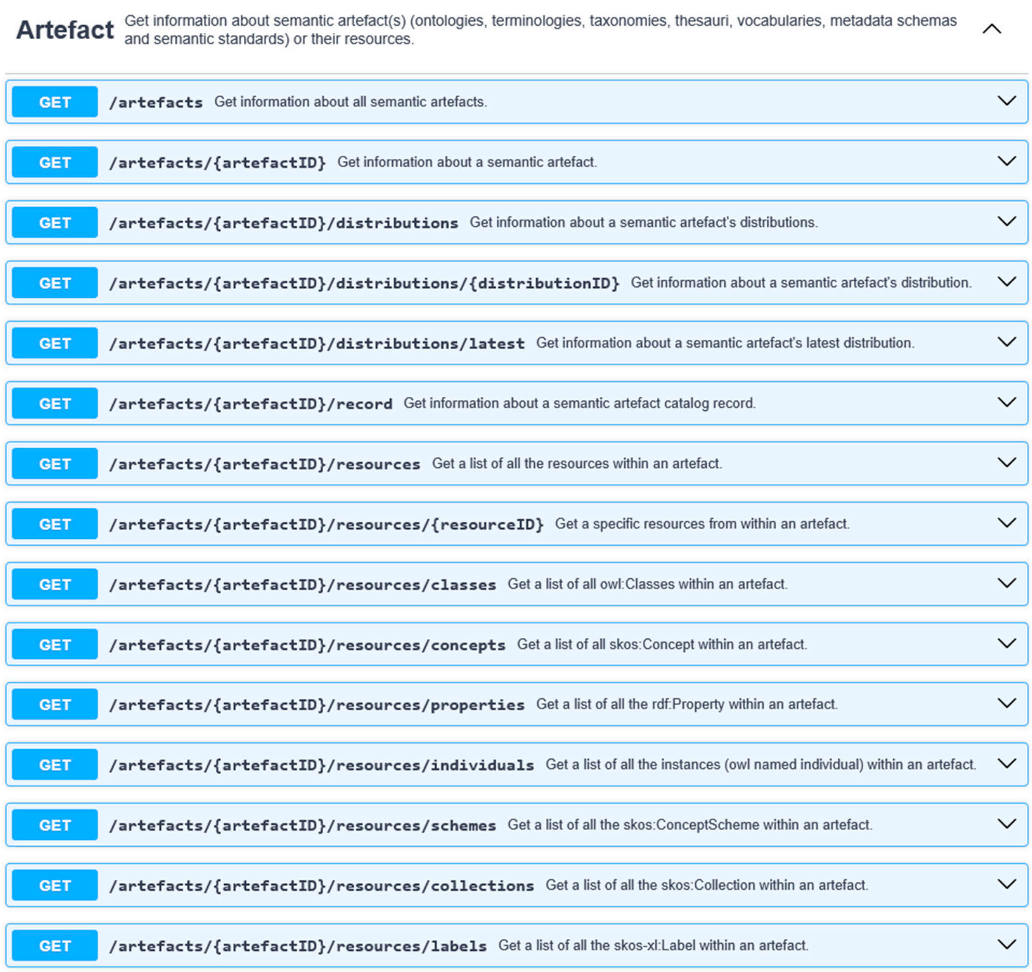

Figure 12. Artefact-related resources in the MOD-API.

Figure 13 illustrates similar endpoints for catalog records. The figure also shows that the API base URL (i.e., the root "/" path) returns a description of the catalog itself. Other relevant endpoints include the search-related ones, which represent a class of purpose-built resources specifically designed to model operations, in this case, search functionalities. Finally, the/doc/api endpoint can be used to access interactive documentation for the API, which also allows users to invoke and experiment with the API directly from the web browser without requiring additional tools.

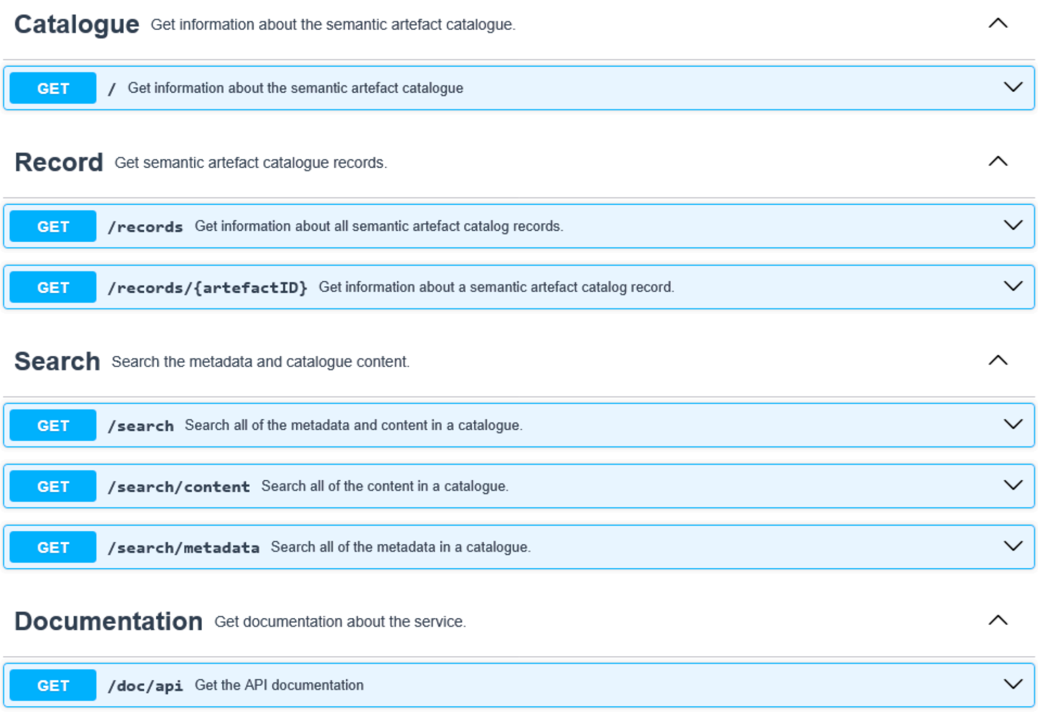

Figure 13. Resources related to the catalog and the records in the MOD-API, as well as search operations.

The default representation format is JSON-LD [75], which is RDF disguised as a plain JSON object, thanks to a JSON-LD context providing proper mapping from JSON properties to RDF properties (along with other mapping information).

Both metadata-related and data-oriented endpoints may return a large number of resources. This can be challenging for users who are unable to manage so many results or are simply not interested in all of them. In fact, they may prefer retrieving the results in chunks, which can be processed or displayed as they arrive, thereby reducing the perceived latency.

The MOD-API addresses these problems by adopting pagination of the resource collections and search results. It achieves this using the Hydra Core Vocabulary [76].

Figure 14 illustrates the approach using a fictional/artefacts resource. The overall resource is a hydra:Collection, whose members are introduced by the member property in the Hydra vocabulary. This representation is only partial, as indicated in the view property. As users navigate through the different pages, the overall resource's @id remains the same, while the view's @id reflects the current page. In particular, the page number appears as a query-string parameter (e.g., ?page=3). The view object also provides links to the first, previous, next, and last pages, enabling clients to traverse the collection without manually composing URLs with the query parameter. The API additionally supports the pagesize parameter to control the number of items included on each page.

```
{
  "@context": {
    ...
    "totalItems": "hydra:totalItems",
    "itemsPerPage": "hydra:itemsPerPage",
    "member": "hydra:member",
    "view": "hydra:view",
    "Collection": "hydra:Collection",
    "PartialCollectionView": "hydra:PartialCollectionView",
    "firstPage": "hydra:first",
    "previousPage": "hydra:previous",
    "nextPage": "hydra:next",
    "lastPage": "hydra:last",
    "view": "hydra:view"
  },
  "@id": "https://example.org/artefacts",
  "@type": "Collection",
  "totalItems": {
    "@type": "nonNegativeInteger",
    "@value": 5436
  },
  "itemsPerPage": {
    "@type": "nonNegativeInteger",
    "@value": 50
  },
  "member": [
    {
      "@id": "https://example.org/artefacts/151",
      "@type": "SemanticArtefact",
    },
    ...
  ],
  "view": {
    "@id": "https://example.org/artefacts?page=3",
    "@type": "PartialCollectionView",
    "firstPage": "https://example.org/artefacts?page=1",
    "previousPage": "https://example.org/iartefacts?page=2",
    "nextPage": "https://example.org/artefacts?page=4",
    "lastPage": "https://example.org/artefacts?page=42"
  }
}
```


Figure 14. Example of pagination in the MOD-API.

5. Adopting the MOD-API in ShowVoc

Relying on the background provided by Sections 3 and 4, the following subsections detail the steps required to integrate the MOD-API within ShowVoc.

5.1. Comparison Between the MOD Ontology and the MDR Metadata Model

We first examined the extent of semantic discrepancies between the MOD ontology and the MDR metadata model used in ShowVoc, without attempting a full or formal alignment between the two. Instead, Table 2 presents the results of a conceptual comparison, showing that there are indeed some discrepancies, an unsurprising outcome given the distinct contexts in which they were developed.

Table 2. Conceptual correspondence between (classes defined in) the MOD ontology and the MDR metadata model.

MOD	MDR
mod:SemanticArtefactCatalog	dcat:Catalog
mod:SemanticArtefactCatalogRecord	dcat:CatalogRecord
mod:SemanticArtefact └ mod:Terminology, mod:Taxonomy, mod:Thesaurus	mdr:DatasetArchetype, mdr:DatasetRealization
	mdr:MutableDatasetRealization
mod:SemanticArtefactDistribution	dcat:Distribution
	mdr:RDFDataset
mod:SemanticArtefactDataService	dcat:DataService, mdr:SPARQLEndpoint
	mdr:Deployment
	mdr:DistributionScheme
	mdr:Linkset, mdr:LinksetArchetype, mdr:LinksetRealization, mdr:RDFLinkset
	mdr:Changeset, mdr:ChangesetArchetype, mdr:ChangesetRealization, mdr:RDFChangeset
mod:Analytics	
mod:EngineeringMethodology	
mod:Evaluation	
mod:Group	
mod:KnowledgeRepresentationParadigm	
mod:SemanticArtefactTask	

As a starting point, we note a straightforward correspondence between mod:Semantic ArtefactCatalog, mod:SemanticArtefactCatalogRecord, and mod:SemanticArtefactDistribution and their counterparts in the MDR, which, in fact, does not subclass DCAT at all. Conversely, the correspondence for mod:SemanticArtefact is less straightforward, as the MDR introduces a finer distinction between mdr:DatasetArchetype and mdr:DatasetRealization. The reduced level of detail in MOD is likely a consequence of its foundation on DCAT 2, which, as previously noted, provides limited guidance on versioning. Moreover, the MDR includes several classes that introduce concepts not represented in the MOD ontology, such as those enabling a detailed characterization of linksets, changesets, deployments, or RDF datasets. SPARQL endpoints could, in principle, be modeled as instances of mod:SemanticArtefactDataService augmented with properties specifying that it uses the SPARQL protocol. The MOD ontology also defines additional (sub)classes that have no counterpart in the MDR. However, since these do not represent core concepts in the MOD-

API or the MDR, their absence within the MDR can be accommodated by incorporating them as is when required for the description of these types of resources.

5.2. Integration Approach

The analysis conducted in the previous section indicates that a one-to-one mapping between the MDR metadata model and the MOD ontology is not feasible. Nevertheless, this does not hinder the use of the MOD-API as an additional access layer for the semantic dataset catalog managed by the MDR component within ShowVoc (see Figure 15).

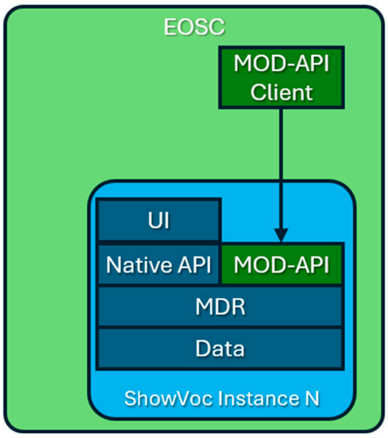

Figure 15. Implementation of the MOD-API as an additional access layer on top of the MDR component.

Recalling that the MOD-API associates its various endpoints with specific classes of the MOD ontology, implementing this strategy merely requires the ability to retrieve the corresponding instances by issuing suitable queries against the MDR. Since the MOD ontology is intended to serve as the *global schema* supporting interoperability within the future EOSC, this integration approach aligns with the Global-as-View (GAV) paradigm for data integration [77]. Conversely, the Local-as-View (LAV) paradigm would have required expressing the MDR entities in terms of the MOD ontology. However, as noted earlier, this would not have allowed a complete instantiation of the MDR model, given that the MOD ontology provides a less detailed representation. Such an approach would have been necessary only if we had chosen to adopt MOD as the foundational model and then re-implement, atop this new foundation, the MDR-based native API to preserve backward compatibility with existing use cases.

5.3. Mapping the MOD Ontology onto the MDR Metadata Model

The key idea underpinning the mapping of the MOD ontology onto the MDR metadata model is to associate each mdr:DatasetArchetype with a corresponding mod:SemanticArtefact. Since the MOD-API targets Semantic Artefact Catalogs (SACs) that expose actual data rather than merely providing metadata, the mapping considers only those archetypes for which ShowVoc manages at least one locally hosted copy that is accessible to the current user (including non-logged-in users).

Figure 16 illustrates a SPARQL query that performs this task, returning one solution for each dataset archetype eligible for the mapping. It should be noted that the MOD ontology specifies certain properties for a semantic artefact, such as owl:versionIRI in the minimum metadata profile and owl:versionInfo, which the MDR represents on the current version of the dataset archetype, potentially using different properties (e.g., dcat:version instead of owl:versionInfo). For this reason, the query retrieves this information together with the dataset archetype in each solution.

```
PREFIX dcat: <http://www.w3.org/ns/dcat#>
PREFIX mdr: <https://w3id.org/mdr#>
PREFIX mod: <https://w3id.org/mod#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX stmdr: <http://semanticturkey.uniroma2.it/ns/stmdr#>
PREFIX st: <http://semanticturkey.uniroma2.it/ns/rdf4j#>

SELECT DISTINCT ?datasetArchetype ?shortName ?currentVersion ?versionId
WHERE {
  ?datasetArchetype
    rdf:type mdr:DatasetArchetype;
    .
  FILTER EXISTS {
    ?datasetArchetype
     (dcat:hasCurrentVersion|dcat:hasVersion)
       /dcat:distribution/mdr:deployment ?deployment;
   .
    ?deployment
      rdf:type mdr:Deployment, stmdr:ProjectDataShard;
      mdr:locator ?locator;
      dcat:accessService ?colocateST;
   .
   FILTER(st:isAccessible(?locator))
  }
  ?datasetArchetype mdr:shortName ?shortName .
  OPTIONAL {
    ?datasetArchetype dcat:hasCurrentVersion ?currentVersion

    OPTIONAL {
      ?currentVersion dcat:version ?versionId
    }
  }

}
ORDER BY ?datasetArchetype
```

Figure 16. SPARQL query that selects dataset archetypes and their current versions for the generation of MOD semantic artefacts. The variable ?colocatedST is expected to be bound to the IRI of the local data service.

To reconcile the two models, the solution is to retrieve this additional metadata from the realization linked to the archetype via the dcat:hasCurrentVersion property. In this regard, it is noteworthy that each pair consisting of an archetype and its current version—which together represent the semantic artefact—is unique at any given time. Updating the reference to a different dataset realization, cataloged as the new current version, corresponds, in MOD, to a semantic artefact updating its versioning information to reflect the latest version.

Figure 17 provides a running example used throughout this section to illustrate the mapping. As shown in the upper part of the figure, an mdr:DatasetArchetype and an mdr:DatasetRealization represent AGROVOC itself and its latest (known) version (which is unique). These entities, highlighted in green, are then combined to generate a single mod:SemanticArtefact (also highlighted in green), which synthesizes information drawn from both.

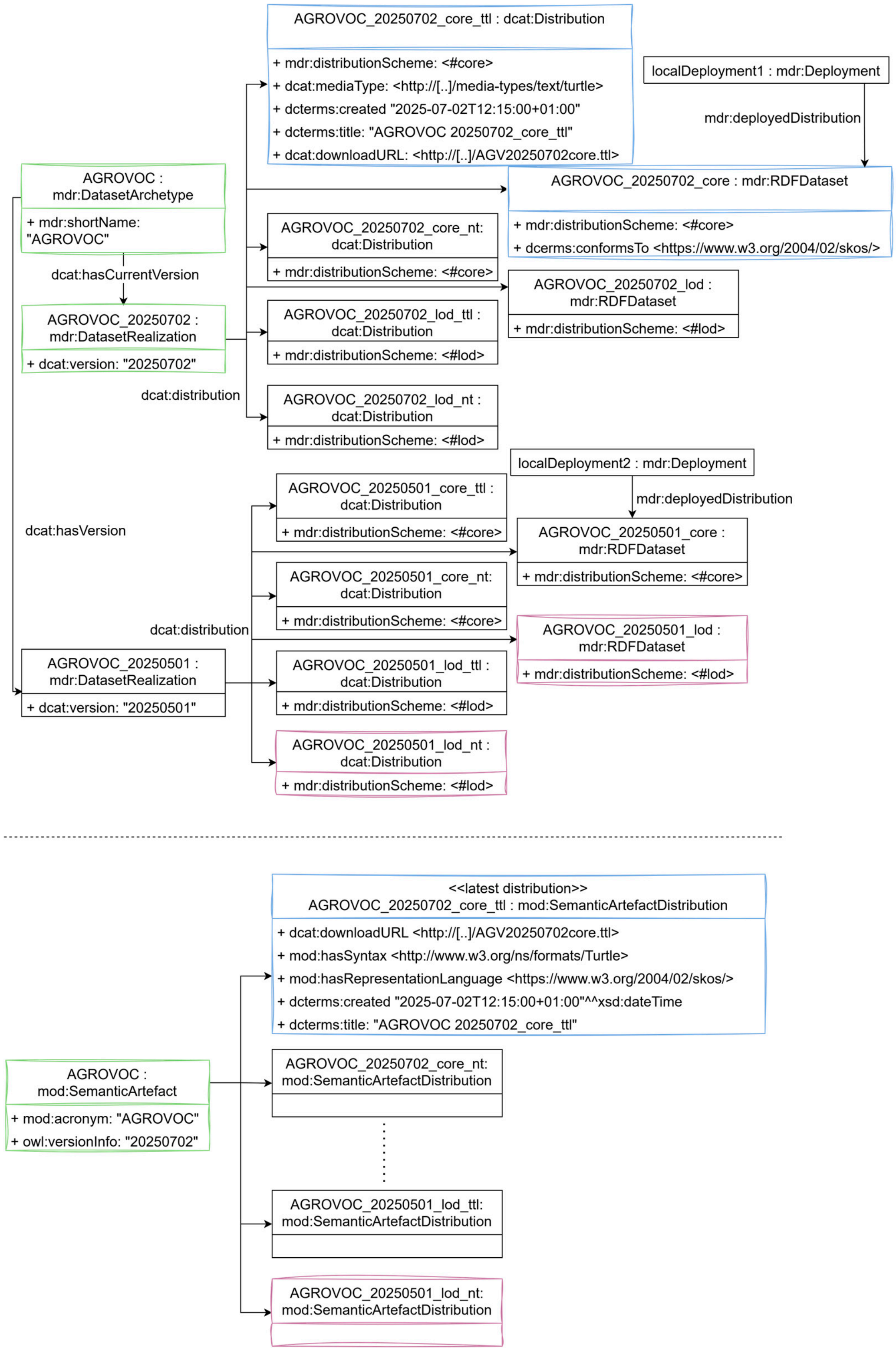

Figure 17. Mapping of semantic artefacts and their distributions.

Concerning the metadata properties to be used on a semantic artefact, Table 3 reveals a very small overlap between the mandatory properties in the FAIRsFAIR minimum metadata profile and the MDR.

Table 3. Metadata about a mod:SemanticArtefact: properties with requirement "mandatory" in the FAIRsFAIR minimum metadata profile (column "MOD") and correspondence in the MDR (column "MDR"). Concerning the latter, we use the following convention: = indicates that the same property is used, ← indicates a different property used in the MDR, while 👤 means that the MDR does not have a corresponding property, but the same property can be used in the user-supplied metadata .

MOD	MDR
http://purl.org/dc/terms/accessRights	👤
https://w3id.org/mod#acronym	← mdr:shortName
http://www.w3.org/ns/dcat#contactPoint	👤
http://purl.org/dc/terms/creator	👤
http://purl.org/dc/terms/description	=
http://purl.org/dc/terms/identifier	👤
http://www.w3.org/ns/dcat#keyword	👤
http://www.w3.org/ns/dcat#landingPage	👤
http://purl.org/dc/terms/license	👤
http://purl.org/dc/terms/rightsHolder	👤
http://purl.org/dc/terms/subject	👤
http://purl.org/dc/terms/title	=
http://purl.org/dc/terms/type	👤
http://www.w3.org/2002/07/owl#versionIRI	👤

The properties dcterms:title and dcterms:description are used as they are, while the property mod:acronym, which provides the identifier of a semantic artefact within a catalog, can be mapped to mdr:shortName. The remaining properties do not have a direct counterpart in the MDR; however, the MDR can accommodate them, provided that the user has supplied values for them in the "main" graph associated with the dataset archetype in the catalog (see Section 6 for more details).

Once the correspondence between MOD and the MDR has been established for semantic artefacts, the corresponding mod:SemanticArtefactCatalogRecord is generated from the archetype's record in the MDR. As shown in Table 4, the only mandatory properties in this case are dcterms:created and dcterms:modified, both of which are already managed by the MDR.

Table 4. Metadata about a mod:SemanticArtefactCatalogRecord: properties with requirement "mandatory" in the FAIRsFAIR minimum metadata profile (column "MOD") and correspondence in the MDR (column "MDR"). Concerning the latter, '=' indicates that the same property is used.

MOD	MDR
http://purl.org/dc/terms/created	=
http://purl.org/dc/terms/modified	=

While the mapping of distributions appears straightforward, given that we identified a 1-to-1 correspondence between mod:SemanticArtefactDistribution and dcat:Distribution used in the MDR, a subtle discrepancy between the two models must be noted. In MOD, the same class is used to manage both multiple distribution schemes (e.g., a "core" distribution including only SKOS-XL labels and a "lod" distribution containing plain SKOS labels and other materialized redundancies) and different versions, all of which are linked to the same semantic artefact resource. By contrast, the MDR would represent the latter (i.e., different versions) as distributions belonging to distinct dataset realizations associated with the same semantic artefact as different versions. Therefore, we generated the distributions of a mod:SemanticArtefact originating from a given mdr:DatasetArchetype in the MDR by considering all archived distributions (i.e., downloadable files, represented as plain dcat:Distribution instances) reachable from the dataset archetype via the property path (dcat:hasCurrentVersion | dcat:hasVersion)/dcat:distribution.

As discussed earlier concerning semantic artefacts, some metadata describing a distribution are stored in the corresponding mdr:RDFDataset, which represents the abstract triple set and can be identified by matching the property mdr:distributionScheme. This abstract triple set, which is a type of void:Dataset, is used to represent metrics and other characteristics of a dataset's informational content (including LIME metadata) that are independent of its specific file-level serialization. Returning to our running example in Figure 17, we note the dcat:Distribution and the corresponding mdr:RDFDataset in the MDR (highlighted in blue) that were used to generate the SKOS core distribution of version "20250702" of AGROVOC in TURTLE format. The figure also illustrates the property mapping, which follows the correspondences summarized in Table 5: most metadata properties are shared between the two models, with mappings required only for those describing the concrete syntax (e.g., TURTLE) and the representation language (e.g., SKOS). We also note that the minimum metadata profile requires the use of dcat:accessURL; however, this property is not appropriate for distributions that are directly downloadable, for which both the MDR and MOD rely on the dcat:downloadURL property.

Table 5. Metadata about a mod:SemanticArtefactDistribution: properties with requirement "mandatory" in the FAIRsFAIR minimum metadata profile (column "MOD") and correspondence in the MDR (column "MDR"). Concerning the latter, we use the following convention: = indicates that the same property is used, ← indicates a different property used in the MDR.

MOD	MDR
http://www.w3.org/ns/dcat#accessURL	=
http://purl.org/dc/terms/created	=
http://purl.org/dc/terms/description	=
https://w3id.org/mod#hasRepresentationLanguage	← dcterms:conformsTo
https://w3id.org/mod#hasSyntax	← dcat:mediaType
http://purl.org/dc/terms/modified	=
http://purl.org/dc/terms/title	=

As can be seen, the mandatory properties of dataset distributions (and the additional ones introduced by MOD) do not record version information or distribution schemes, resulting in the loss of important metadata. For example, MOD provides no explicit indication of which distribution(s) correspond to which version. Moreover, it is not explicitly stated which distribution corresponds to the latest version indicated on the MOD semantic artefact resource. From an operational perspective, this could be approximated by selecting the most recent distribution. A more reliable solution, however, is to use the endpoint/artefacts/{artefactID}/distributions/latest, which returns the single latest distribution.

A potential issue arises from the fact that a unique latest distribution does not accommodate multiple distributions of the same version conforming to different distribution schemes. Fortunately, this is not currently problematic, as we consider only locally deployed distributions of a dataset archetype's current version when computing the latest distribution, and ShowVoc presently enforces the constraint of deploying only one distribution per version. This behavior is also consistent with the MOD-API, whose data-oriented endpoints are expected to serve the content of a uniquely determined latest distribution.

6. Implementation Details

ShowVoc adopts a conventional three-layer architecture: the presentation layer is implemented as a single-page application (SPA) running in the browser, whereas the business and persistence layers operate server-side within Semantic Turkey [78]. Semantic Turkey is a modular, service-oriented platform for Semantic Web applications. Beyond ShowVoc, a prominent system built on this platform is VocBench 3 [79], a collaborative platform for the development and maintenance of ontologies, thesauri, lexicons, and datasets in general.

Semantic Turkey is not restricted to semantic artefacts as defined by MOD; it can manage multiple versions of general RDF datasets, relying on different knowledge and lexicalization models. The platform includes a Metadata Registry (MDR) component, based on the corresponding metadata ontology introduced earlier. The MDR functions as the central access point from which other system components obtain metadata about both locally hosted and remotely available datasets. In line with the objectives described in [16], this metadata enables the system to effectively adapt to, present, and exploit the available information.

Semantic Turkey already supported the API-gateway approach to catalog interoperability through connectors originally devised for use within VocBench [80]. This form of process-level integration was validated in a scenario where VocBench acted as an intermediary between ShowVoc and OntoPortal [81]. Transitioning to the shared-API approach, we enhanced Semantic Turkey with a new service implementing the MOD-API. The service stubs and their domain model were generated using a code generator [82] configured with the MOD-API's OpenAPI [83] specifications [84], and Swagger UI [85] was employed to provide interactive documentation.

Unlike MOD, which requires the use of specific descriptive metadata, the MDR is more in the spirit of DCAT, allowing any metadata vocabulary consistent with the MDR. Consequently, the availability of particular metadata properties depends on the metadata profile configured for the Semantic Turkey instance. We also introduced configuration options to support metadata adaptation. Then, for each dataset, the MDR component manages multiple graphs, whose names are generated from the dataset IRI in the catalog by appending the corresponding suffix in the following list:

- -sys: a graph containing system-managed metadata.
- -main: a graph containing metadata added to the dataset description, potentially using vocabularies and properties beyond those originally foreseen by the MDR.
- -stats: a graph containing statistics computed automatically.

7. Evaluation

The MOD-API defines five compliance levels for each API endpoint, ranging from 0 to 4, with higher levels indicating better compliance:

- Level 0: The endpoint is not implemented.
- Level 1: The endpoint returns a response of any kind.
- Level 2: The endpoint returns an object with the correct type according to the MOD ontology.

- Level 3: As in Level 2, but the object returned by the endpoint includes as many relevant MOD properties as possible.
- Level 4: As in Level 3, and the returned object additionally satisfies a minimum metadata profile, which is ultimately based on the FAIRsFAIR one.

Through code generation, our implementation can easily satisfy at least Level 1, since a stub for each endpoint is generated. The generated domain model instead facilitates achieving Level 2, as the model objects returned by an endpoint are correctly typed according to the MOD classes. This approach further facilitates reaching Level 3, because it enables the inclusion of as many MOD-defined properties as possible. For catalog records, our system consistently attains Level 4, given that the required metadata properties correspond to predefined descriptors in the MDR. As noted in Section 6, the level of compliance achievable for other endpoints depends on the specific metadata profile adopted in the Semantic Turkey instance.

Table 6 summarizes the compliance levels achievable for each endpoint. Cells marked as "≥Level 3" indicate that the endpoint is guaranteed to reach at least Level 3, since it returns all relevant properties currently available, with the possibility of reaching Level 4 when the full set of metadata required by the minimum profile is present in the resource description managed by the MDR. Some endpoints inherently reach Level 4, as the minimum metadata profile only includes properties that the MDR always provides. Endpoints under/artefacts/{artefactID}/resources can reach at most Level 3 because MOD does not specify a minimum metadata profile for resource-level entities such as classes or properties within a semantic artefact returned by these operations. The same reasoning applies to/search and/search/content, which may also return resources defined within semantic artefacts and thus fall outside the scope of the minimum metadata profile. Finally,/doc/api only achieves Level 1, since the MOD ontology does not define a dedicated type for this resource.

Toward the end of the FAIR-IMPACT implementation action, we assessed our work using an API compliance and coverage validator, originally developed for the MOD-API implementation in OntoPortal and recommended within the initiative. The validation computed an overall score based on the five compliance levels across all service endpoints. ShowVoc achieved a score of 71%, which at the time exceeded those of other systems still under development, whose scores were 46%, 24%, 21%, and 6%.

We updated the validator to use a revised OpenAPI specification document, mainly to remove a few data-oriented endpoints that have been removed from the specifications, and we then evaluated the current ShowVoc implementation (specifying display=all& as an additional parameter). Across the 22 evaluated endpoints, the system achieved an overall score of 1670/1760 (95%). Table 7 details the results, reporting the score obtained for each individual endpoint, assuming a maximum possible score of 80 per endpoint. With respect to our initial assessment reported in Table 6, the only discrepancy concerns the endpoint/artefacts/{X}/resources/{Y}. Although this endpoint could theoretically achieve level 3, the dynamic testing performed by the validator revealed a limitation in the underlying REST framework: the platform is unable to correctly extract the resource URI appended to the endpoint path. This issue is in fact a known limitation of the specification, which had suggested the introduction of an alternative syntax:/artefacts/{X}/resources?resource={Y}. However, this proposal did not make its way into the specification before its finalization.

Table 6. Compliance levels with the MOD-API according to manual code analysis.

Endpoint	Compliance Level
/artefacts	≥Level 3
/artefacts/{artefactID}	≥Level 3
/artefacts/{artefactID}/distributions	≥Level 3
/artefacts/{artefactID}/distributions/{distributionID}	≥Level 3
/artefacts/{artefactID}/distributions/latest/resources	≥Level 3
/artefacts/{artefactID}/record	Level 4
/artefacts/{artefactID}/resources	Level 3
/artefacts/{artefactID}/resources/{resourceID}	Level 3
/artefacts/{artefactID}/resources/classes	Level 3
/artefacts/{artefactID}/resources/concepts	Level 3
/artefacts/{artefactID}/resources/properties	Level 3
/artefacts/{artefactID}/resources/individuals	Level 3
/artefacts/{artefactID}/resources/schemes	Level 3
/artefacts/{artefactID}/resources/collection	Level 3
/artefacts/{artefactID}/resources/labels	Level 3
/	≥Level 3
/records	Level 4
/records/{artefactID}	Level 4
/search	Level 3
/search/content	Level 3
/search/metadata	≥Level 3
/doc/api	Level 1

It is worth noting that the validator is based on the MOD-API's OpenAPI specification file, yet it adopts a more permissive interpretation of it by accepting both plain property names (e.g., "title") and prefixed ones (e.g., "dcterms:title"). Although the prefixed form was ultimately agreed upon during the finalization of the MOD-API, the plain form persisted in several implementations, including ours and OntoPortal's. This discrepancy does not pose practical issues, as both forms are fully equivalent from the perspective of a JSON-LD client that is provided with the appropriate context within the response.

Moreover, the validator does not award additional points to systems, such as ShowVoc, that support content negotiation. This is not particularly problematic, since most systems participating in the implementation call provided only the JSON-LD representation. The other representations—possibly with the exception of the HTML one—merely correspond to alternative serializations of the same RDF triples, which can always be derived from JSON-LD and generally do not offer additional practical benefits, given that most RDF libraries are capable of handling all these serializations.

Table 7. Detailed validation report returned by the MOD-API validator.

Endpoint	Exist (Level 1)	Good @ Type (Level2)	Properties (Level 3–4)	Valid JSON-LD	Paginated	Score
/	✓	✓	✓	✓		80
/records	✓	✓	N/A	✓	✓	80
/doc/api	✓	N/A	N/A	✗		70
/artefacts	✓	✓	✓	✓	✓	80
/search	✓	N/A	N/A	✓		80
/search/content	✓	N/A	N/A	✓		80
/search/metadata	✓	✓	✓	✓	✓	80
/records/{X}	✓	✓	N/A	✓		80
/artefacts/{X}	✓	✓	✓	✓		80
/artefacts/{X}/distributions	✓	✓	✓	✓	✓	80
/artefacts/{X}/distributions/latest	✓	✓	✓	✓		80
/artefacts/{X}/record	✓	✓	N/A	✓		80
/artefacts/{X}/resources	✓	N/A	N/A	✓	✓	80
/artefacts/{X}/resources/classes	✓	N/A	N/A	✓	✓	80
/artefacts/{X}/resources/concepts	✓	N/A	N/A	✓	✓	80
/artefacts/{X}/resources/properties	✓	N/A	N/A	✓	✓	80
/artefacts/{X}/resources/individuals	✓	N/A	N/A	✓		80
/artefacts/{X}/resources/schemes	✓	N/A	N/A	✓	✓	80
/artefacts/{X}/resources/collections	✓	N/A	N/A	✓	✓	80
/artefacts/{X}/resources/labels	✓	N/A	N/A	✓	✓	80
/artefacts/{X}/distributions/{Y}	✓	✓	✓	✓		80
/artefacts/{X}/resources/{Y}	✗	N/A	N/A	✗		0

8. Discussion

As previously noted in Section 5.2, the MOD-API cannot replace the API built on the Metadata Registry that ShowVoc relies on; instead, the two will continue to operate in parallel. Nevertheless, adding this additional access layer, aligned with an emerging standard for the EOSC, enhances the interoperability of ShowVoc's platform within this strategic infrastructure.

While Sections 5.2 and 5.3 have already examined the conceptual discrepancies between the MOD ontology and the MDR metadata model, as well as the ways in which these can be reconciled, Table 8 provides a high-level summary based on the requirements outlined in [16] with some revisions to improve their clarity and reflect recent evolutions. Although these requirements originate from the needs the MDR must satisfy to support complex dissemination use cases targeted by ShowVoc, this discussion also offers insights into the applicability of the MOD ontology and its API in broader contexts and may help inform future directions for their development.

Table 8. Functional comparison between the MDR and MOD.

Requirement	MDR	MOD
represent a catalog	✓	✓
versioning	version -> dataset realization	version -> distribution
SPARQL endpoint	mdr:SPARQLEndpoint as a DCAT data service exposing a given distribution	mod:SemanticArtefactDataService as a DCAT data service
SPARQL limitations	✓	✗
express dereferenciation	✓	✗
Linguistic information: lang./model/statistics	✓/✓/✓	✓/~/✗
Structure metadata: knowledge model/statistics	✓/✓	✓/✓
cover lexical resources	✓	✗
linksets	✓	✗
changesets	✓	✗

Both MDR and MOD can *represent a catalog* based on the fundamental distinction, introduced by DCAT, between datasets and their distributions. However, MDR extends this dichotomy by making explicit the concept of a dataset realization, an intermediate construct used to represent different *versions* of the same dataset, modeled as a dataset archetype. This approach leverages the hierarchical versioning model introduced in DCAT 3, whose specifications do, in fact, refer to abstract datasets without assigning them a distinct class. In contrast, MOD manages both different versions and different distribution schemes simply as multiple distributions of the same semantic artefact. As previously noted, this results in a lossy representation: it is not possible to associate each distribution with the correct version, as version metadata is only expressed on the semantic artefact. Moreover, MDR treats all versions uniformly, and ShowVoc can host multiple browsable versions of a dataset. By contrast, the MOD-API currently allows browsing only the latest distribution of a semantic artefact.

Both MDR and MOD support *SPARQL endpoints* through specializations of the dcat:DataService class. However, the MOD-API does not treat SPARQL endpoints as first-class citizens, nor does it include them in its minimum metadata profile. MDR instead supports the representation of SPARQL endpoint limitations through the property mdr:sparqlEndpointLimitation, which links an endpoint to an instance of mdr:SPARQLEndpointLimitation, an extensible mechanism for modeling application-specific endpoint limitations.

MDR also includes a dedicated property, mdr:dereferenciationSystem, to *express the dereferenciation system* supported by a dataset. This property points to an instance of mdr:DereferenciationSystem, which currently includes two predefined values: mdr:standardDereferenciation (identifiers can be dereferenced via standard HTTP lookup) and mdr:noDereferenciation (identifiers cannot be dereferenced).

As previously discussed, MDR leverages LIME to represent details about available *linguistic information* by describing subsets typed as lime:LexicalizationSet. The description of these subsets includes natural *language*, *lexicalization model*, and various *statistics*. The MOD ontology allows a semantic artefact to be associated with its supported natural languages via dcterms:language, and despite lacking the notion of a lexicalization model, it defines properties on artefact distributions (mod:prefLabelProperty, mod:synonymProperty, mod:definitionProperty) to specify how to retrieve linguistic information encoded using an implicit model. MOD's support for linguistic metrics is limited to a few properties, mainly mod:numberOfLabels (the number of labels for any resource within a semantic artefact), mod:classesWithNoLabel (the number of classes with no label associated with them), and mod:classesWithNoDefinition (the number of classes with no definition associated with them). MDR's reliance on LIME also enables the representation of lexical resources encoded in RDF.

Both MDR and MOD represent the *knowledge model* through dcterms:conformsTo (on the mdr:RDFDataset) and mod:hasRepresentationLanguage (on the mod:SemanticArtefact Distribution), respectively, accompanied by various *statistics*. MOD additionally defines a few highly specific properties, such as mod:classesWithMoreThan25Children, which may reflect design choices inherited from the catalogs that informed its development. Conversely, MOD introduces properties such as mod:hasFormalityLevel and dedicated subclasses of mod:SemanticArtefact, which can indeed help distinguish between different types of knowledge representation systems (KOSes) created using the same underlying model—for instance, SKOS.

Finally, we observe that MDR also supports the representation of *linksets* and *changesets* through dedicated specializations of dcat:Dataset, mdr:DatasetArchetype, and mdr:DatasetRealization.

To conclude this section, we clarify that several of the limitations discussed above are, in practice, easy to overcome. Both MDR and MOD can be readily combined with additional metadata vocabularies, provided that no logical inconsistencies are introduced. For example, nothing prevents the use of selected MOD-defined properties together with MDR, and vice versa. In fact, extending MOD to bring it closer to MDR would likely require an opinionated interpretation of the model, since MOD is inherently less detailed with respect to versioning and is not explicitly aligned with the VoID vocabulary, unlike MDR.

9. Conclusions

The widespread adoption of the FAIR principles has led to the proliferation of numerous data catalogs, ranging from general-domain catalogs to highly specialized ones tailored to specific domains, communities, or even individual organizations. However, despite the availability of generic platforms for building such catalogs, significant differences persist between them, particularly in the APIs they expose and in their metadata models. These discrepancies continue to limit interoperability.

To mitigate these issues, the MOD ontology was adopted and an accompanying API was developed as foundational components of the European Open Science Cloud. Their purpose is to provide a unified framework for structuring and accessing catalogs of so-called semantic artefacts, which include ontologies, thesauri, and related resources, thereby enabling meaningful integration of data annotated with them.

Motivated by the potential benefits of these emerging standards, we participated in an open call aimed at fostering their adoption across relevant cataloging platforms. As with any standard, its real value lies in the breadth of its implementation within the community. To this end, we contributed an implementation of the MOD-API within our cataloging platform ShowVoc.

ShowVoc already relied on a sophisticated catalog managed by its MDR component based on the equally named metadata model. The MOD-API, however, could not replace ShowVoc's MDR, which offers a richer set of features not yet supported in the MOD ontology under the lens of its API. This can be seen as a consequence of a deliberate design choice: the MOD-API focuses on a minimal common core that captures the functionalities typically shared across most catalogs, allowing systems to achieve progressively higher levels of compliance depending on their metadata profiles.

Using the current MOD-API specification as our reference, we implemented the API in ShowVoc as an additional access layer that operates alongside the existing native API while relying on the same MDR component. Our approach follows the Global-As-View (GAV) paradigm for data integration, which assumes that the MOD ontology can be mapped onto a (possibly more detailed) catalog-specific metadata model. This work is grounded in a careful interpretation of the MOD ontology, its limitations, and the extent to which it can be aligned with our MDR metadata model. The resulting approach, together with the lessons learned through the integration effort, may serve as a useful reference for other catalog providers and could also inform future developments of the MOD-API.

Finally, we provide anecdotal evidence of the relevance and impact of this work on improving interoperability among catalogs, drawing on the ITINERIS project, in which we were involved during the same period. As part of that project, we extended ShowVoc to harvest semantic datasets from 23 different catalogs, based on several heterogeneous technological platforms and APIs. This required the development of multiple dedicated connectors, following the approach described in [81]. As a proof of concept, we implemented a MOD-API connector that allowed us to retrieve semantic datasets from another instance of ShowVoc, as well as from TIB [86], NERC [87], and AgroPortal. The latter is particularly significant, as its implementation of the MOD-API is being integrated into the core OntoPortal code base, which will make it available to all OntoPortal-based catalogs, including the 11 members of the OntoPortal Alliance. This experiment highlights the potential impact of the MOD-API: with a single connector, ShowVoc would be able to interoperate not only with TIB and NERC but also with seven additional OntoPortal-based catalogs (TechnoPortal, MedPortal, EcoPortal, EarthPortal, BioPortal, BiodivPortal, and AgroPortal), reaching 9 out of the 23 catalogs considered (approximately 39%). Once the MOD-API implementation in these catalogs has stabilized and our connector has been finalized with full support for update detection, the latter could serve as a unified solution, substantially reducing the overall maintenance burden.

Author Contributions: Conceptualization, M.F., P.B. and A.S.; methodology, M.F. and A.S.; software, M.F. and A.S.; validation, M.F., P.B. and A.S.; formal analysis, M.F. and A.S.; investigation, M.F., P.B. and A.S.; resources, M.F. and A.S.; data curation, M.F. and A.S.; writing—original draft preparation, M.F., P.B. and A.S.; writing—review and editing, M.F., P.B. and A.S.; visualization, M.F., P.B. and A.S.; supervision, M.F. and A.S.; project administration, A.S.; funding acquisition, M.F. and A.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was co-funded through tenders awarded to us by the Publications Office of the European Union in the context of the Digital Europe Programme (https://digital-strategy.ec.europa.eu/en/activities/digital-programme, accessed on 6 March 2026) and by the National Research Council of Italy, in the context of the ITINERIS project: Italian Integrated Environmental Research Infrastructures System, a project funded by the European Union—Next Generation EU PNRR- Mission 4 "Education and Research"—Component 2: "From research to business"—Investment 3.1: "Fund for the realisation of an integrated system of research and innovation infrastructures". Furthermore, the implementation of the MOD-API in ShowVoc has been financially supported through the initiative "Implementation of a shared API for semantic catalogues support action" (https://fair-impact.eu/implementing-shared-api-for-semantic-catalogues, accessed on 6 March 2026), organized by the FAIR-IMPACT project, which is funded under the Horizon Europe program (Agreement ID 101057344).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Dataset available on request from the authors.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Hey, T.; Tansley, S.; Tolle, K.M. (Eds.) *The Fourth Paradigm: Data-Intensive Scientific Discovery*; Microsoft Research: Redmond, WA, USA, 2009.
2. Li, X.; Guo, Y. Paradigm shifts from data-intensive science to robot scientists. *Sci. Bull.* **2025**, *70*, 14–18. [CrossRef] [PubMed]
3. Kitchin, R. Big Data, new epistemologies and paradigm shifts. *Big Data Soc.* **2014**, *1*. [CrossRef]
4. FAIR Principles. Available online: https://www.go-fair.org/fair-principles/ (accessed on 6 March 2026).
5. Wilkinson, M.D.; Dumontier, M.; Aalbersberg, I.J.; Appleton, G.; Axton, M.; Baak, A.; Blomberg, N.; Boiten, J.-W.; da Silva Santos, L.B.; Bourne, P.E.; et al. The FAIR Guiding Principles for scientific data management and stewardship. *Sci. Data* **2016**, *3*, 160018. [CrossRef] [PubMed]
6. Corcho, O.; Ekaputra, F.J.; Heibi, I.; Jonquet, C.; Micsik, A.; Peroni, S.; Storti, E. A maturity model for catalogues of semantic artefacts. *Sci. Data* **2024**, *11*, 479. [CrossRef] [PubMed]
7. Jonquet, C.; Bouazzouni, S.; Alviset, G.; Fiore, N.; Karam, N.; Kihal, B.; Pierkot, C.; Pulieri, M.; Rosati, I. Federated FAIR Semantic Artefacts Discovery and Search with OntoPortal Federation. In *The Semantic Web—ISWC 2025*; Garijo, D., Kirrane, S., Salatino, A., Shimizu, C., Acosta, M., Nuzzolese, A.G., Ferrada, S., Soulard, T., Kozaki, K., Takeda, H., et al., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2026; Volume 16141, pp. 434–450. [CrossRef]
8. TSNFDI API Gateway. Available online: https://ts4nfdi.github.io/api-gateway (accessed on 6 March 2026).
9. European Open Science Cloud—EOSC. Available online: https://digital-strategy.ec.europa.eu/en/policies/open-science-cloud (accessed on 6 March 2026).
10. FAIR-IMPACT. Available online: https://fair-impact.eu/ (accessed on 6 March 2026).
11. Wilson, A.; Jonquet, C. *D4.3—Specification of Shared Metadata Description of Semantic Artefacts and Their Catalogues Including Common Reference API*; Project Deliverable; V1.0—DRAFT NOT YET APPROVED BY THE EUROPEAN COMMISSION; Zenodo: Geneva, Switzerland, 2024. [CrossRef]
12. GitHub—FAIR-IMPACT/MOD-API: API Specification for Metadata for Ontology Description (MOD). Available online: https://github.com/FAIR-IMPACT/MOD-API (accessed on 6 March 2026).
13. GitHub—FAIR-IMPACT/MOD: Metadata for Ontology Description and Publication Ontology. Available online: https://github.com/FAIR-IMPACT/MOD (accessed on 6 March 2026).
14. #4: Implementation of a Shared API for Semantic Catalogues | FAIR-IMPACT. Available online: https://fair-impact.eu/implementing-shared-api-for-semantic-catalogues (accessed on 6 March 2026).
15. ShowVoc. Available online: https://showvoc.uniroma2.it/ (accessed on 6 March 2026).
16. Fiorelli, M.; Stellato, A.; Lorenzetti, T.; Turbati, A.; van Gemert, W.; Dechandon, D.; Gerencsér, A.; Francesconi, E. MDR: An Ontology Vocabulary and Registry Service for Dataset Catalogs. In *Advances and Trends in Artificial Intelligence. Theory and Applications*; Fujita, H., Watanobe, Y., Ali, M., Wang, Y., Eds.; Lecture Notes in Computer Science; Springer Nature: Singapore; Kitakyushu, Japan, 2025; Volume 15706, pp. 354–366. [CrossRef]
17. The Metadata Registry Ontology. Available online: https://w3id.org/mdr (accessed on 6 March 2026).
18. Berners-Lee, T.; Hendler, J.; Lassila, O. The Semantic Web. *Sci. Am. Mag.* **2001**, *284*, 34–43. Available online: https://www.scientificamerican.com/article/the-semantic-web/ (accessed on 6 March 2026). [CrossRef]
19. Heath, T.; Bizer, C. *Linked Data—Evolving the Web into a Global Data Space*; Springer: Cham, Switzerland, 2011. [CrossRef]
20. Janowicz, K.; van Harmele, F.; Hendler, J.A.; Hitzler, P. Why the Data Train Needs Semantic Rails. *AI Mag.* **2015**, *36*, 5–14. [CrossRef]
21. Hitzler, P.; Janowicz, K. Linked Data, Big Data, and the 4th Paradigm. *Semant. Web* **2013**, *4*, 233–235. [CrossRef]
22. Alexander, K.; Cyganiak, R.; Hausenblas, M.; Zhao, J. Describing Linked Datasets with the VoID Vocabulary (W3C Interest Group Note), World Wide Web Consortium (W3C). 2011. Available online: http://www.w3.org/TR/void/ (accessed on 16 May 2012).
23. Alexander, K.; Cyganiak, R.; Hausenblas, M.; Zhao, J. Describing Linked Datasets. In Proceedings of the Linked Data on the Web Workshop (LDOW 09) Colocated with WWW 09, Madrid, Spain, 20 April 2009. Available online: https://ceur-ws.org/Vol-538/ldow2009_paper20.pdf (accessed on 6 March 2026).
24. AGROVOC VoID Dataset Description. Available online: https://aims.fao.org/aos/agrovoc/void.ttl#Agrovoc (accessed on 6 March 2026).
25. Caracciolo, C.; Stellato, A.; Morshed, A.; Johannsen, G.; Rajbhandari, S.; Jaques, Y.; Keizer, J. The AGROVOC Linked Dataset. *Semant. Web J.* **2013**, *4*, 341–348. [CrossRef]

26. Mountantonakis, M.; Allocca, C.; Fafalios, P.; Minadakis, N.; Marketakis, Y.; Lantzaki, C.; Tzitzika, Y. Extending VoID for Expressing Connectivity Metrics of a Semantic Warehouse. In Proceedings of the 1st International Workshop on Dataset PROFIling & fEderated Search for Linked Data, Anissaras, Crete, Greece, 26 May 2014. Available online: https://ceur-ws.org/Vol-1151/paper10.pdf (accessed on 6 March 2026).
27. Vocabulary of a Friend (VOAF). Available online: http://purl.org/vocommons/voaf (accessed on 6 March 2026).
28. Fiorelli, M.; Stellato, A.; Mccrae, J.P.; Cimiano, P.; Pazienza, M.T. LIME: The Metadata Module for OntoLex. In *The Semantic Web*; Latest Advances and New Domains; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2015; Volume 9088, pp. 321–336. [CrossRef]
29. Cimiano, P.; Chiarcos, C.; McCrae, J.P.; Gracia, J. *Linguistic Linked Data*; Springer: Berlin/Heidelberg, Germany, 2020. [CrossRef]
30. Commission Opens Investigation into Possible Anticompetitive Conduct by Google in the Use of Online Content for AI Purposes. Available online: https://ec.europa.eu/commission/presscorner/detail/en/ip_25_2964 (accessed on 6 March 2026).
31. Search Engines in Times of Artificial Intelligence. Available online: https://www.europarl.europa.eu/RegData/etudes/ATAG/2025/779238/EPRS_ATA(2025)779238_EN.pdf (accessed on 6 March 2026).
32. The Linked Open Data Cloud. Available online: https://lod-cloud.net/ (accessed on 6 March 2026).
33. Vandenbussche, P.-Y.; Atemezing, G.A.; Poveda-Villalón, M.; Vatant, B. Linked Open Vocabularies (LOV): A gateway to reusable semantic vocabularies on the Web. *Semant. Web* **2016**, *8*, 437–452. [CrossRef]
34. Linked Open Vocabularies. Available online: https://lov.linkeddata.es/ (accessed on 6 March 2026).
35. Jonquet, C.; Graybeal, J.; Bouazzouni, S.; Dorf, M.; Fiore, N.; Kechagioglou, X.; Redmond, T.; Rosati, I.; Skrenchuk, A.; Vendetti, J.L.; et al. Ontology Repositories and Semantic Artefact Catalogues with the OntoPortal Technology. In *The Semantic Web—ISWC 2023*; Payne, T.R., Presutti, V., Qi, G., Poveda-Villalón, M., Stoilos, G., Hollink, L., Kaoudi, Z., Cheng, G., Li, J., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2023; Volume 14266, pp. 38–58. [CrossRef]
36. OntoPortal | OntoPortal Website Presenting the OntoPortal Technology Product Developped by the OntoPortal Alliance. Available online: https://ontoportal.org/ (accessed on 6 March 2026).
37. Jonquet, C.; Toulet, A.; Arnaud, E.; Aubin, S.; Dzalé Yeumo, E.; Emonet, V.; Graybeal, J.; Laporte, M.-A.; Musen, M.A.; Pesce, V.; et al. AgroPortal: A vocabulary and ontology repository for agronomy. *Comput. Electron. Agric.* **2018**, *144*, 126–143. [CrossRef]
38. AgroPortal. Available online: https://agroportal.lirmm.fr/ (accessed on 6 March 2026).
39. Tarallo, A.; Pulieri, M.; Ramezani, P.; Rosati, I. Advancements in EcoPortal: Enhancing functionalities for the ecological domain semantic artefacts repository. *FAIR Connect. Empower. Data Steward.* **2024**, *2*, 1–7. [CrossRef]
40. EcoPortal. Available online: https://ecoportal.lifewatch.eu/ (accessed on 6 March 2026).
41. Karam, N.; Fillies, J.; Jonquet, C.; Bouazzouni, S.; Löffler, F.; Zander, F.; König-Ries, B.; Güntsch, A.; Diepenbroek, M.; Paschke, A. BiodivPortal: Enabling Semantic Services for Biodiversity within the German National Research Data Infrastructure. *Datenbank-Spektrum* **2024**, *24*, 129–137. [CrossRef]
42. BiodivPortal. Available online: https://biodivportal.gfbio.org/ (accessed on 6 March 2026).
43. Pierkot, C.; Alviset, G.; Vernet, M. The EarthPortal, towards an ontology repository for the Earth System semantic artefacts. In Proceedings of the 3rd Workshop on Ontologies for FAIR and FAIR Ontologies (Onto4FAIR), Co-Located with the SEMANTiCS Conference, Leipzig, Germany, 20–22 September 2023. Available online: https://hal.science/hal-04908666/ (accessed on 6 March 2026).
44. EarthPortal. Available online: https://earthportal.eu/ (accessed on 6 March 2026).
45. MedPortal. Available online: https://medportal.bmicc.cn/ (accessed on 6 March 2026).
46. TechnoPortal. Available online: https://technoportal.hevs.ch/ (accessed on 6 March 2026).
47. Salvadores, M.; Alexander, P.R.; Musen, M.A.; Noy, N.F. BioPortal as a dataset of linked biomedical ontologies and terminologies in RDF. *Semant. Web* **2013**, *4*, 277–284. [CrossRef] [PubMed]
48. BioPortal. Available online: https://bioportal.bioontology.org/ (accessed on 6 March 2026).
49. Kropshofer, J.; Schrott, J.; Wöß, W.; Ehrlinger, L. A Survey on the Functionalities of Data Catalog Tools. *IEEE Access* **2025**, *13*, 83297–83319. [CrossRef]
50. World Wide Web Consortium (W3C). Data Catalog Vocabulary (DCAT). World Wide Web Consortium (W3C). Available online: http://www.w3.org/TR/vocab-dcat/ (accessed on 6 March 2026).
51. Albertoni, R.; Browning, D.; Cox, S.; Gonzalez-Beltran, A.N.; Perego, A.; Winstanley, P. The W3C Data Catalog Vocabulary, Version 2: Rationale, Design Principles, and Uptake. *Data Intell.* **2024**, *6*, 457–487. [CrossRef]
52. World Wide Web Consortium (W3C). Asset Description Metadata Schema (ADMS), World Wide Web Consortium (W3C). 2013. Available online: https://www.w3.org/TR/vocab-adms/ (accessed on 6 March 2026).
53. Gray, A.J.G.; Baran, J.; Marshall, M.S.; Dumontier, M. Dataset Descriptions: HCLS Community Profile (W3C Interest Group), World Wide Web Consortium (W3C). 2015. Available online: https://www.w3.org/TR/hcls-dataset/ (accessed on 3 December 2019).
54. Van Nuffelen, B. SEMIC. Available online: https://semiceu.github.io/ADMS/releases/2.00/ (accessed on 6 March 2026).

55. Dataset Search. Google Research. Available online: https://datasetsearch.research.google.com/ (accessed on 6 March 2026).
56. Schema.org. Available online: https://schema.org/ (accessed on 6 March 2026).
57. Tennison, J. CSV on the Web: A Primer. W3C Working Group Note. 2016. Available online: https://www.w3.org/TR/tabular-data-primer/ (accessed on 6 March 2026).
58. DCAT Application Profile for Data Portals in EUROPE (DCAT-AP). EU Vocabularies. Available online: https://op.europa.eu/it/web/eu-vocabularies/dcat-ap (accessed on 6 March 2026).
59. Brümmer, M.; Baron, C.; Ermilov, I.; Freudenberg, M.; Kontokostas, D.; Hellmann, S. DataID: Towards Semantically Rich Metadata for Complex Datasets. In Proceedings of the 10th International Conference on Semantic Systems, Leipzig, Germany, 4–5 September 2014; pp. 84–91. [CrossRef]
60. Dutta, B.; Nandini, D.; Shahi, G.K. MOD: Metadata for ontology description and publication. In Proceedings of the 2015 International Conference on Dublin Core and Metadata Applications (DCMI'15), São Paulo, Brazil, 1–4 September 2015; pp. 1–9. Available online: https://dl.acm.org/doi/abs/10.5555/2907896.2907897 (accessed on 6 March 2026).
61. Research Data Alliance (RDA). Available online: http://rd-alliance.org/ (accessed on 6 March 2026).
62. Vocabulary Service Interest Group. Research Data Alliance. Available online: https://www.rd-alliance.org/groups/vocabulary-services-interest-group/ (accessed on 6 March 2026).
63. FAIRsFAIR. Available online: https://www.fairsfair.eu/ (accessed on 6 March 2026).
64. Dutta, B.; Toulet, A.; Emonet, V.; Jonquet, C. New Generation Metadata Vocabulary for Ontology Description and Publication. In *Metadata and Semantic Research*; Garoufallou, E., Virkus, S., Siatri, R., Koutsomiha, D., Eds.; Communications in Computer and Information Science; Springer: Cham, Switzerland, 2017; Volume 755, pp. 173–185. [CrossRef]
65. Jonquet, C.; Dutta, B.; Bonino da Silva Santos, L.O.; Pergl, R.; Le Franc, Y. Common Minimum Metadata for FAIR Semantic Artefacts. In Proceedings of the 2nd Workshop on Ontologies for FAIR and FAIR Ontologies (Onto4FAIR), Co-Located with the 13th International Conference on Formal Ontology in Informations Systems (FOIS 2023), Sherbrooke, QC, Canada, 17–20 July 2023. Available online: https://hal.science/hal-04106533v2 (accessed on 6 March 2026).
66. Matentzoglu, N.; Malone, J.; Mungall, C.; Stevens, R. MIRO: Guidelines for minimum information for the reporting of an ontology. *J. Biomed. Semant.* **2018**, *9*, 6. [CrossRef] [PubMed]
67. DCMI Usage Board. DCMI Metadata Terms. 2020. Available online: https://www.dublincore.org/specifications/dublin-core/dcmi-terms/ (accessed on 6 March 2026).
68. Hartmann, J.; Palma, R.; Sure, Y.; Suárez-Figueroa, M.C.; Haase, P.; Gómez-Pérez, A.; Studer, R. Ontology Metadata Vocabulary and Applications. In *On the Move to Meaningful Internet Systems 2005*; Meersman, R., Tari, Z., Herrero, P., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2005; Volume 3762, pp. 906–915. [CrossRef]
69. Fiorelli, M.; Pazienza, M.T.; Stellato, A. A meta-data driven platform for semi-automatic configuration of ontology mediators. In Proceedings of the 9th International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, UK, 26–31 May 2014; pp. 4178–4183. Available online: https://aclanthology.org/L14-1041/ (accessed on 6 March 2026).
70. Fiorelli, M.; Stellato, A.; Lorenzetti, T.; Schmitz, P.; Francesconi, E.; Hajlaoui, N.; Batouche, B. Metadata-driven Semantic Coordination. In *Metadata and Semantic Research*; Garoufallou, E., Fallucchi, F., William De Luca, E., Eds.; Communications in Computer and Information Science; Springer: Cham, Switzerland, 2019; Volume 1057. [CrossRef]
71. Fiorelli, M.; Stellato, A.; Lorenzetti, T. MAPLE: Metadata-driven orchestration of ontology matching. *Int. J. Metadata Semant. Ontol.* **2025**, *17*, 18–39. [CrossRef]
72. EuroVoc. EU Vocabularies. Available online: https://op.europa.eu/it/web/eu-vocabularies/dataset/-/resource?uri=http://publications.europa.eu/resource/dataset/eurovoc (accessed on 6 March 2026).
73. Walhain, L.; Albouze, S.; Gerencsér, A.; Paunescu, M.; Tzouvaras, V.; Palma, C. The EuroVoc Thesaurus: Management, Applications, and Future Directions. In Proceedings of the 5th Conference on Language, Data and Knowledge, Naples, Italy, 9–11 September 2025; pp. 340–350. Available online: https://aclanthology.org/2025.ldk-1.34/ (accessed on 6 March 2026).
74. EuroVoc VoID Dataset Description. Available online: https://eurovoc.europa.eu/void.ttl (accessed on 6 March 2026).
75. JSON-LD—JSON for Linking Data. Available online: https://json-ld.org/ (accessed on 6 March 2026).
76. Lanthaler, M. Hydra Core Vocabulary, Hydra W3C Community Group. 2021. Available online: https://www.hydra-cg.com/spec/latest/core/ (accessed on 6 March 2026).
77. Lenzerini, M. Data integration: A theoretical perspective. In *Proceedings of the 21th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS '02)*; Association for Computing Machinery: New York, NY, USA, 2002; pp. 233–246. [CrossRef]
78. Pazienza, M.T.; Scarpato, N.; Stellato, A.; Turbati, A. Semantic Turkey: A Browser-Integrated Environment for Knowledge Acquisition and Management. *Semant. Web* **2012**, *3*, 279–292. [CrossRef]
79. Stellato, A.; Fiorelli, M.; Turbati, A.; Lorenzetti, T.; van Gemert, W.; Dechandon, D.; Laaboudi-Spoiden, C.; Gerencsér, A.; Waniart, A.; Costetchi, E. VocBench 3: A Collaborative Semantic Web Editor for Ontologies, Thesauri and Lexicons. *Semant. Web* **2020**, *11*, 855–881. [CrossRef]

80. VocBench. Available online: https://vocbench.uniroma2.it/ (accessed on 6 March 2026).
81. Fiorelli, M.; Stellato, A.; Rosati, I.; Fiore, N. Process-Level Integration for Linked Open Data Development Workflows: A Case Study. In *Metadata and Semantic Research*; Garoufallou, E., Vlachidis, A., Eds.; Communications in Computer and Information Science; Springer: Cham, Switzerland, 2023; Volume 1789, pp. 148–159. [CrossRef]
82. OpenAPI Generator. Available online: https://openapi-generator.tech/ (accessed on 6 March 2026).
83. OpenAPI Initiative. Available online: https://www.openapis.org/ (accessed on 6 March 2026).
84. Metadata for Ontology Description and Publication Ontology Application Program Interface. GitHub—FAIR-IMPACT. Available online: https://fair-impact.github.io/MOD-API/ (accessed on 6 March 2026).
85. REST API Documentation Tool. Swagger UI. Available online: https://swagger.io/tools/swagger-ui/ (accessed on 6 March 2026).
86. TIB. Available online: https://www.tib.eu/ (accessed on 6 March 2026).
87. NERC. Available online: https://vocab.nerc.ac.uk/ (accessed on 6 March 2026).

https://doi.org/10.3390/technologies14030193