This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 870811

SPICE

Social cohesion, Participation, and Inclusion through Cultural Engagement

D4.7 Linked Data server technology: Final release and open-source distribution

Deliverable information	
WP	WP4
Document dissemination level	PU Public
Deliverable type	R Document, report
Lead beneficiary	OU
Contributors	UNIBO, UNITO, IMMA, H-FARM, UH, UCM, DMH, MAIZE
Date	28/04/2023
Document status	Final
Document version	V1.0

INTENTIONALLY BLANK PAGE

Project information

Project start date: 1st of May 2020

Project Duration: 36 months

Project website: https://spice-h2020.eu

Project contacts

Project Coordinator

Silvio Peroni

ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA

Department of Classical Philology and Italian Studies – FICLIT

E-mail: silvio.peroni@unibo.it

Project Scientific coordinator

Aldo Gangemi

Institute for Cognitive Sciences and Technologies of the Italian National Research Council

E-mail: aldo.gangemi@cnr.it

Project Manager

Adriana Dascultu

ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA

Executive Support Services

E-mail: adriana.dascultu@unibo.it

SPICE consortium

No.	Short name	Institution name	Country
1	UNIBO	ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA	Italy
2	AALTO	AALTO KORKEAKOULUSAATIO SR	Finland
3	DMH	DESIGNMUSEON SAATIO - STIFTELSEN FOR DESIGNMUSEET SR	Finland
4	AAU	AALBORG UNIVERSITET	Denmark
5	OU	THE OPEN UNIVERSITY	United Kingdom
6	IMMA	IRISH MUSEUM OF MODERN ART COMPANY	Ireland
7	GVAM	GVAM GUIAS INTERACTIVAS SL	Spain
8	PG	PADAONE GAMES SL	Spain
9	UCM	UNIVERSIDAD COMPLUTENSE DE MADRID	Spain
10	UNITO	UNIVERSITA DEGLI STUDI DI TORINO	Italy
11	FTM	FONDAZIONE TORINO MUSEI	Italy
12	MAIZE	MAIZE SRL (Previously CELI SRL)	Italy
13	UH	UNIVERSITY OF HAIFA	Israel
14	CNR	CONSIGLIO NAZIONALE DELLE RICERCHE	Italy

Executive summary

SPICE is an EU H-2020 project dedicated to research on novel methods for citizen curation of cultural heritage through an ecosystem of tools co-designed by an interdisciplinary team of researchers, technologists, and museum curators and engagement experts, and user communities. This technical report D4.7 presents the deliverables of Work Package 4 of the SPICE project, focusing on presenting the final release of the SPICE Linked Data Hub. Alongside presenting novel functionalities and core components of the Linked Data Hub, we present details on the packaging of the final version of this linked data platform with instructions on how it may be deployed for reuse. We also collect feedback received from pilot application developers about the Linked Data server technologies and their integration with the data platform. Furthermore, we conclude the report by discussing potential future work in continuing to address the developing needs of systems implementing the citizen curation methodology, based on the input of the SPICE pilots.

Document History

Version	Release date	Summary of changes	Author(s) -Institution
V0.1	20/03/2023	First draft	Jason Carvalho (OU)
V0.2	29/03/2023	Contributions from pilot study authors.	All partners
V0.3	30/03/2023	Task status table. Final conclusions added and integrated feedback comments. Image captions and labelling.	Jason Carvalho (OU)
V0.4	05/04/2023	Internal review version	Guillermo Jimenez Diaz (UCM) Pablo Gutierrez Sanchez (PG)
V0.5	20/04/2023	Internal review comments processed and actioned.	Jason Carvalho (OU), Enrico Daga (OU)
V0.6	21/04/2023	Software deployment and code resources information added	Jason Carvalho (OU)
V0.7	26/04/2023	Final editing	Jason Carvalho (OU), Enrico Daga (OU)
V1.0		Final version submitted to REA	UNIBO

Table of Contents

1. Introduction

SPICE is an EU H-2020 project dedicated to citizen curation of cultural heritage. To support citizen curation, the project research upon and develops an ecosystem of methods and tools co-designed by an interdisciplinary team of researchers, technologists, domain experts, and user communities. Work Package 4 develops a Distributed Linked Data social media layer, including the technical architecture, the specification of the communication protocols, and the back-end support to the pilot studies developed within the project. Objective of the WP is to research on the application of Linked Data principles to connect cultural objects, collections, and citizen contributions, into an infrastructure for interoperability and knowledge exchange within citizen curation activities. While the WP aims at providing the infrastructure for interoperability within the project, by doing that, its goal is researching on a social media platform that can support museums and technologists with:

(1) privacy-aware content sharing methods, so that museums can expose their catalogue and digital assets in a safe and controlled data environment;
(2) methods for expressing and reasoning over fine-grained policies and constraints associated to digital assets;
(3) linking assets and metadata to support search and discovery capabilities (on top of a secure and controlled data environment); and
(4) content provenance, usage tracing, and monitoring in order to support large scale analyses of user-generated, (anonymised) content.

Background on the state of affairs with data infrastructures for citizen curation is discussed in [Daga et al, 2021], published by the SPICE WP4 team in the initial phase of the project. The main requirements of the Linked Data Infrastructure, its core functionalities, and relation to the state of the art, are discussed in D4.1 [SPICE D4.1]. In D4.1, we provided insights on the Citizen Curation paradigm and performed an analysis of the state of art technologies for Linked Data content management and provided insights on the status of data management in SPICE museums. In addition, in D4.1 we devised requirements for a data infrastructure based on the performed analyses, illustrating the design of a Linked Data Layer, a collection of components and protocols for data communication and exchange across the SPICE network. These include the SPICE Linked Data Hub, a data management and publishing infrastructure, and the SPARQL Anything tool, to support knowledge engineers in transforming heterogenous resources into Linked Data.

Moving on from the initial requirements of the SPICE Linked Data Hub, D4.2 [SPICE D4.2] described the development during the second year of the project and the new features and functionality that were added to the LDH during this period. D4.2 also contained a thorough summary of the SPICE pilot applications that made use of the LDH, including technical details of their integration and useful feedback from pilot application developers on which aspects of the LDH integration had gone well and where there might be areas of development for consideration to be included into the final year or work on the LDH. This feedback formed an important part of the direction that LDH development took in the final year of the project, providing the opportunity to maximise the effectiveness of the LDH for application developers whilst also working towards meeting the original WP4 objects set out in D4.1.

Much of the development work carried out in the third and final year of the project is detailed in separate subject-specific deliverables. D4.3 [SPICE D4.3] describes the work carried out on the distributed privacy and policy later of the LDH, as well as features that provide content monitoring for personally identifiable information (PII) and hate speech detection. D4.5 [SPICE D4.5] and D4.6 [SPICE D4.6] describe the approach taken to provenance and process analysis; looking at data lifecycles with corresponding use cases.

In this deliverable, we look at some of the final LDH development work but predominantly focus on presenting the LDH in its final form. We present the core components of the LDH along with parallel modules, that together form the SPICE Linked Data Hub in its entirety. We summarise SPICE pilot applications that make use of the LDH along with some final feedback from application developers on their experience of integrating their apps with the LDH platform. We follow this with conclusions drawn from the feedback received, highlighting potential areas for further development in building a platform that supports the emerging needs of citizen curation technologies and methodologies. Finally, we present a packaged version of the software with instructions on how to deploy LDH instances elsewhere.

Since this is a summary of all LDH development over the course of the project, it should be noted that some of the content in this report overlaps with content presented in previous WP4 deliverables. Specifically, section 2.2 describes the core components of the LDH which are all covered in more detail elsewhere within the work package. Additionally, section 5 gives an overview of SPICE applications that are making use of the LDH; these summaries are also given in more detail in previous deliverables.

2. The SPICE Linked Data Hub

2.1. Overview

This SPICE Linked Data Hub (LDH) was developed as a data infrastructure to support the acquisition and management of dynamic data from a variety of sources including museum collection metadata and digital assets, social media events and user activities, systems' activities (e.g., recommendations, reasoning outputs), ontologies and linked data produced by pilot case studies. A layout of the system and its key components is provided in Figure 2.1.1.

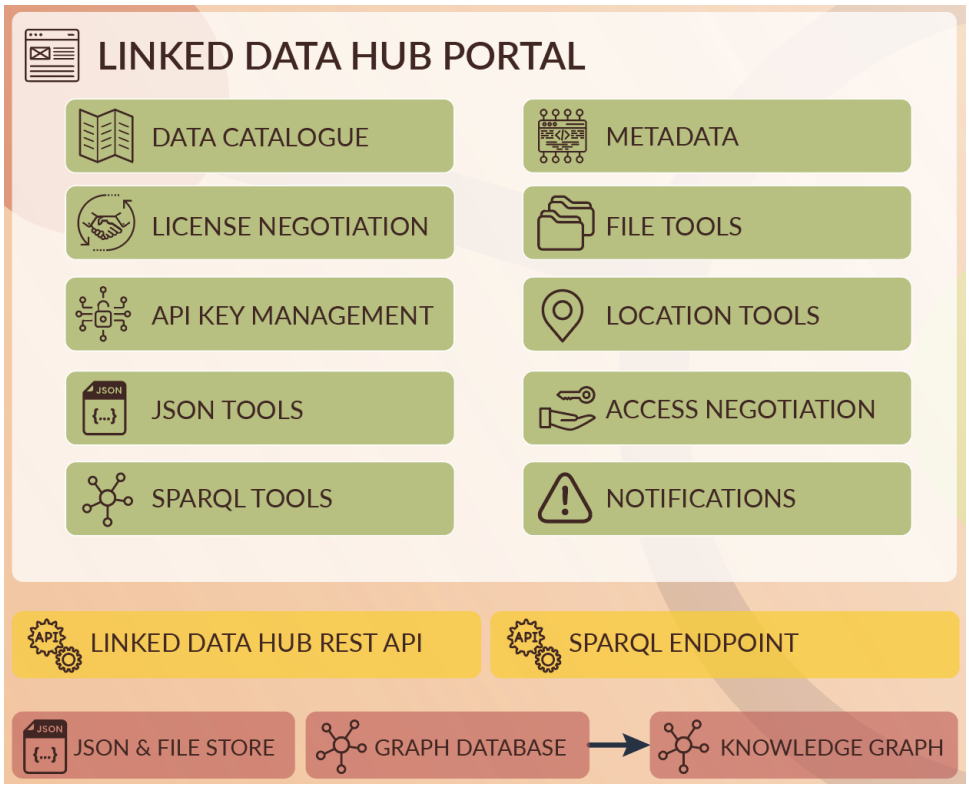

Figure 2.1.1. SPICE Linked Data Hub layout

The SPICE Linked Data Hub is made up of a number of components, the most visible being the front-end web portal. The portal provides a facility for data providers to create and catalogue datasets, manage data and API access as well as manage their privacy, licences and provenance. The web portal enables users to browse catalogues and registries of datasets and their corresponding metadata and data models. Sitting behind the web portal and driving most of its functionality is the LDH API that exposes a range of REST-based API functionality to support the production, management and consumption of data. This API is directly available to all, enabling developers to side-step the web portal and integrate SPICE LDH read/write operations with existing automated systems. The API also offers a range of extended functionality over and above that offered by the web portal such as:

- Full read/write (CRUD) access to datasets for users with appropriate permissions
- Enhanced browsing capabilities
- Advanced querying and data filters (using both JSON-style queries and SPARQL)
- A read-only SPARQL endpoint

Datasets take the form predominantly of JSON documents or static files, so anything that can be encoded as a JSON string can be stored and accessed using the SPICE LDH's full range of features. All JSON documents pushed into the SPICE LDH via the API are also replicated as RDF to a graph database for read-only query access via SPARQL.

2.2. LDH core components

The SPICE Linked Data Hub Portal comprises a series of core components that generally make themselves available to users via a series of tabs on the left-hand side of the portal interface, for each dataset. Some of these components can be made available to all users, such as data metadata and API access details, and other components such as access control management and data notifications are only available to dataset owners and managers.

The development of the SPICE Linked Data Hub has continued throughout the project. New features and functionality are regularly released to address both the initial work package requirements and specific needs or requirements that arise through real-world use and case study and application pilot development. Included in the summary of components below are a number of a new components and developments that have been released since the publication of previous work package deliverables. These include:

- *JSON Tools*. Feedback given in D4.2 (section 5.5.4) stated that the SPICE LDH "would greatly benefit from some sort of web interface within the LDH page itself to be able to do manual browsing and editing, deleting, or uploading of objects in the collections". Previous iterations of the LDH platform required interaction with one's own data entirely through REST API methods provided. This would need to be done either programmatically through scripts or via HTTP-based tools such as CURL or GUI-based API tools such as Postman.

  The JSON Tools section of the LDH portal, described in section 2.2.7, has significantly increased the ease of use of the LDH for dataset managers and application developers. Moreover, it has made it far easier for LDH developers to assist users with data related queries and application debugging, by giving quick and easy access to simple data query, sorting and field selection functions.

- *Content monitoring*. Work has continued since the release of D4.6 to improve the content scanning and notifications features of the LDH. Initially a standalone module for PII detection, the content scanning add-on has been further developed with parent scanning classes that can be inherited and extended by developers. This has made it possible to also drop in an additional scanner to test possible hate speech detection routines. Similarly, the notifications tab of the LDH portal has been

extended to support the rendering of multiple notification types with their own specific sets of notification attributes. Further details are included in section 2.3.1.

2.2.1. Data cataloguing

One of the primary purposes of the LDH portal is to present collections of SPICE datasets in a single place, enabling a range of user types to browse available datasets for use in application development, research and analysis, data re-use and other similar project activities. Dataset owners are given facilities for controlling which of their datasets are listed for specific users and also the option to create theme-specific dataset collections to group together datasets based around particular project pilots or areas of interest. These features are illustrated in Figure 2.2.1.1 and Figure 2.2.1.2.

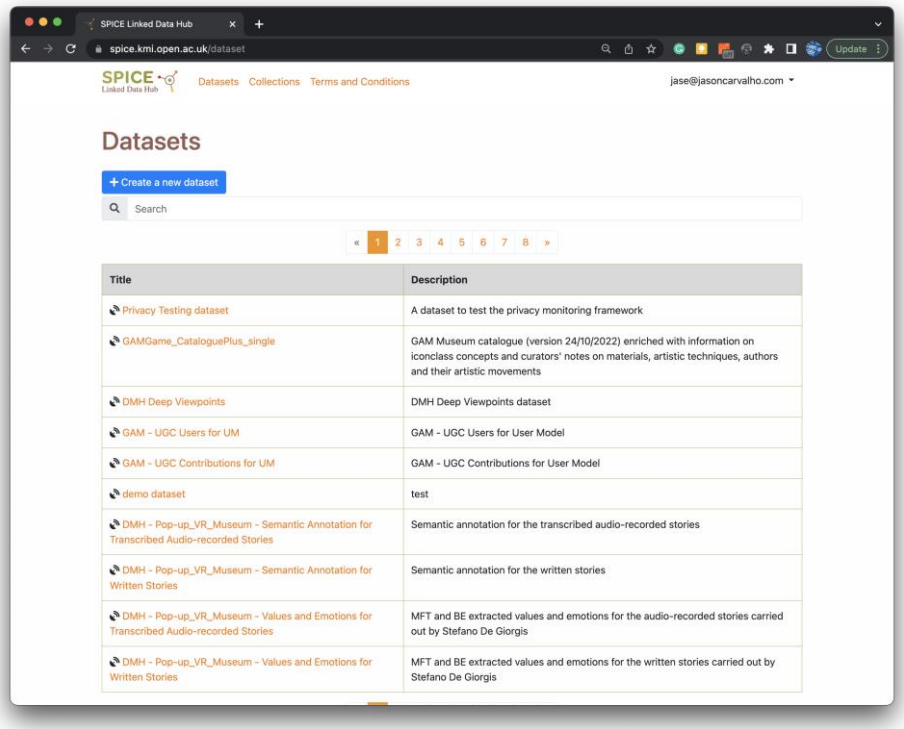

Figure 2.2.1.1 LDH Dataset catalogue

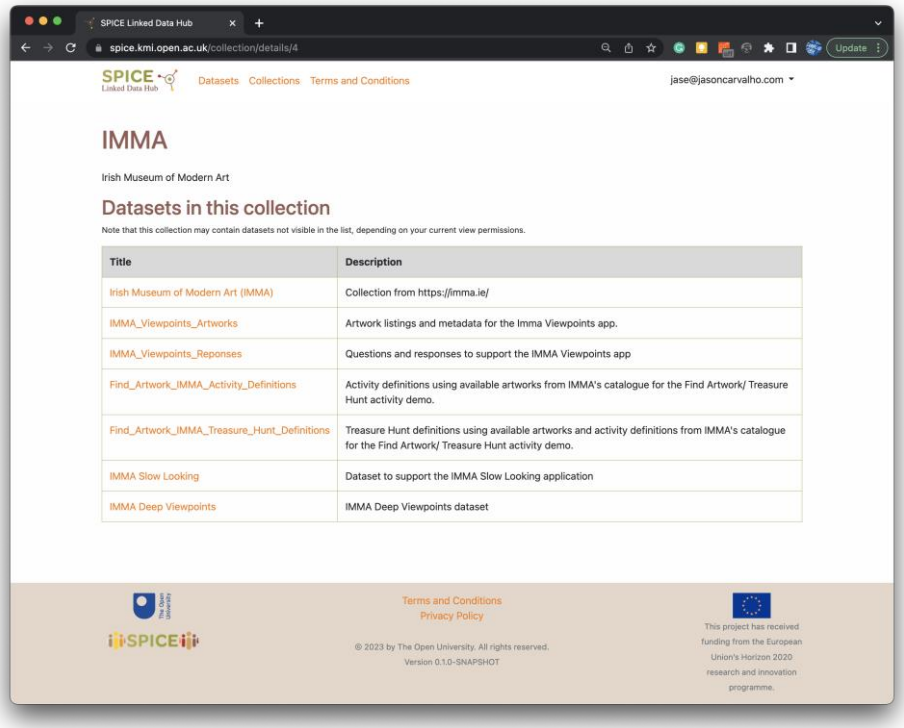

Figure 2.2.1.2 LDH Dataset collections

2.2.2. Access control and negotiation

The SPICE LDH provides dataset owners and managers fine-grained control over exactly who can access particular aspects of their datasets (Figure 2.2.2.1). For each dataset, the following levels of access can be granted or denied:

- View – Dataset appears visible in the LDH catalogue and metadata can be viewed and explored.
- Read – An API access key with read permissions on the content of the dataset can be registered.
- Write – An API access key with write permissions on the content of the dataset can be registered. This includes not only creating new items in the dataset but editing existing items and also deletion of items.
- Manage – The user is granted the same permissions as the dataset owner and can themselves manage the dataset permissions as well as perform other management functions such as managing dataset licensing, RDF graph construction and rebuilding and control of dataset metadata.

Some or all of these access levels can be granted to the following types of users (Figure 2.2.2.2):

- Anonymous – This refers to LDH portal users that are not currently logged in. The only level of access that can be granted for these users is *View.*
- Logged in – This refers to all users that have logged into the LDH portal and serves as a baseline access level which can then be overridden by more specific user access levels stated below. In addition to *View,* both *Read* and *Write* can be granted for this user level.
- Named user – Users can be specifically listed by their email address for their own set of access rights that will override any set in the levels mentioned above. Named users can be granted any of the *View, Read, Write* or *Manage* access levels.
- Dataset owner – This refers to the original owner (creator) of the dataset. All access levels are granted to this type of user and cannot be disabled.

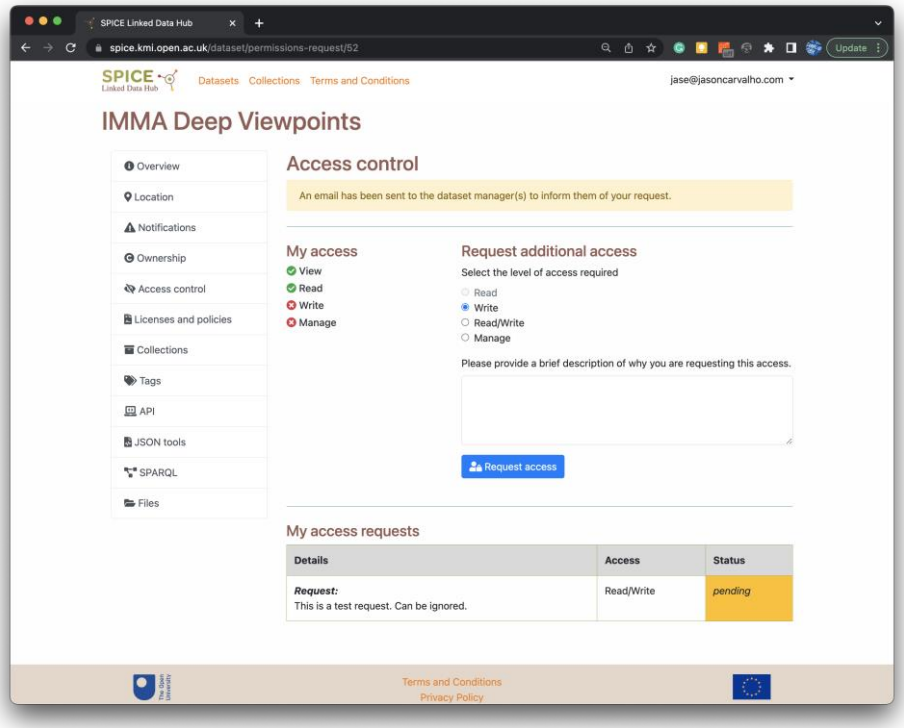

Figure 2.2.2.1 Access control - users

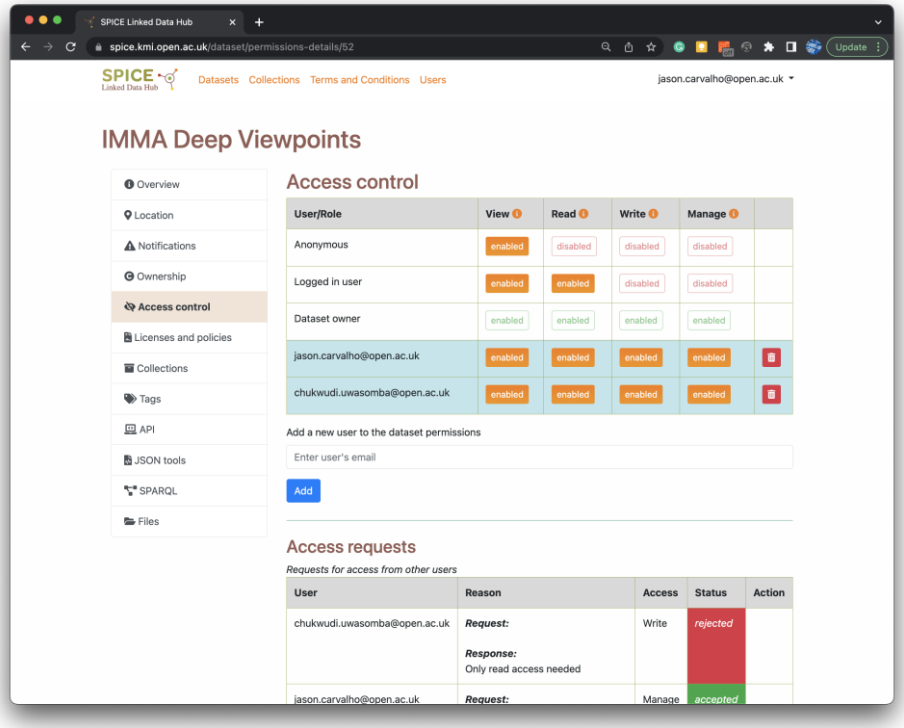

Figure 2.2.2.2 Access control - managers

A development to the original LDH access control mechanisms that arose from feedback from dataset users and SPICE pilot application developers was the facility for access control negotiation. Users can make requests to dataset owners for additional levels of access, with accompanying justifications. These requests are managed in an access request queue and pass through a workflow with appropriate email notifications that allow dataset owners to approve or reject these requests with their own corresponding justifications.

2.2.3. File tools

In D4.2 we presented a suite of new LDH tools for storing and managing files within datasets (Figure 2.2.3.1). All LDH datasets support the storage of files alongside JSON documents via a series of API functions dedicated to *create, read, update, delete* (CRUD) operations on files. Since JSON documents and files can sit alongside each other within a single dataset, all dataset controls and management operations such as access control and policy assignment can be applied across resources regardless of resource type. The LDH also offers user interface tools for accessing the file API CRUS operations via the LDH Portal's web interface.

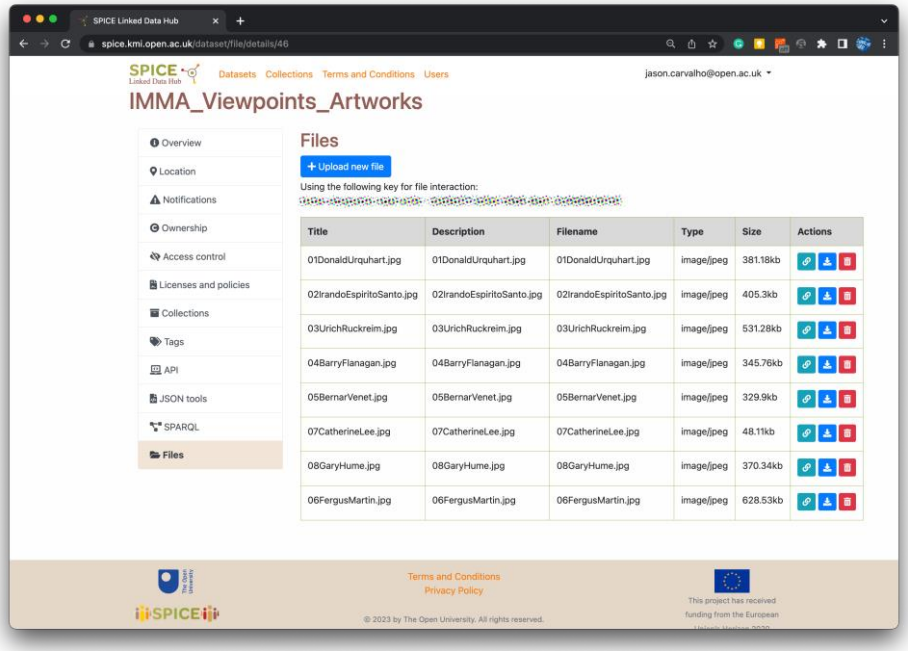

Figure 2.2.3.1 File tools

2.2.4. Location tools

As part of the dataset metadata, the coordinates of the dataset's location can be stored if applicable. Whilst letting LDH users see the dataset's location shown visibly on a map (Figure 2.2.4.1), this also opens options for future development of the SPICE LDH for spatial reasoning and spatial dataset browsing methods such as browsing all datasets within a given radius of a specified city, for example.

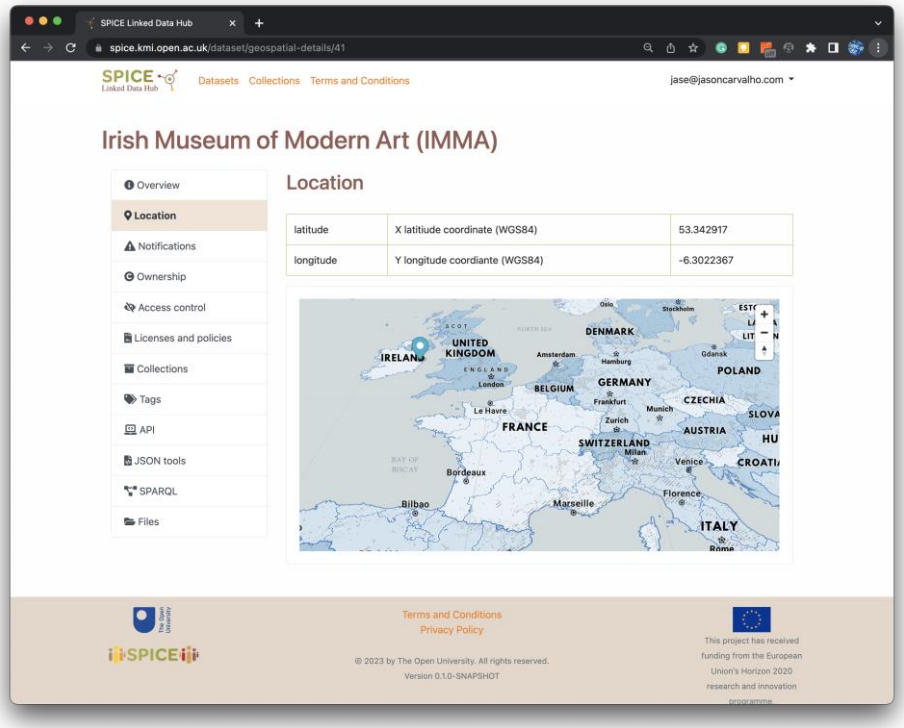

Figure 2.2.4.1 Location tools

2.2.5. API key management

The key management section of the LDH interfaces provides two slightly different interfaces to users, depending on whether they have *Manage* access rights on the dataset. Regular dataset users are presented with a list of keys they currently have registered against a dataset and also the option to register additional keys if permissions allow. From here they can remove key access and, since the recent deployment of the Licensing and Policies layer, view and manage the licenses associated with each of their dataset access keys (Figure 2.2.5.1). Dataset owners and managers are presented with a similar interface, but also have the ability to disable any user's key access to their dataset, should the need arise.

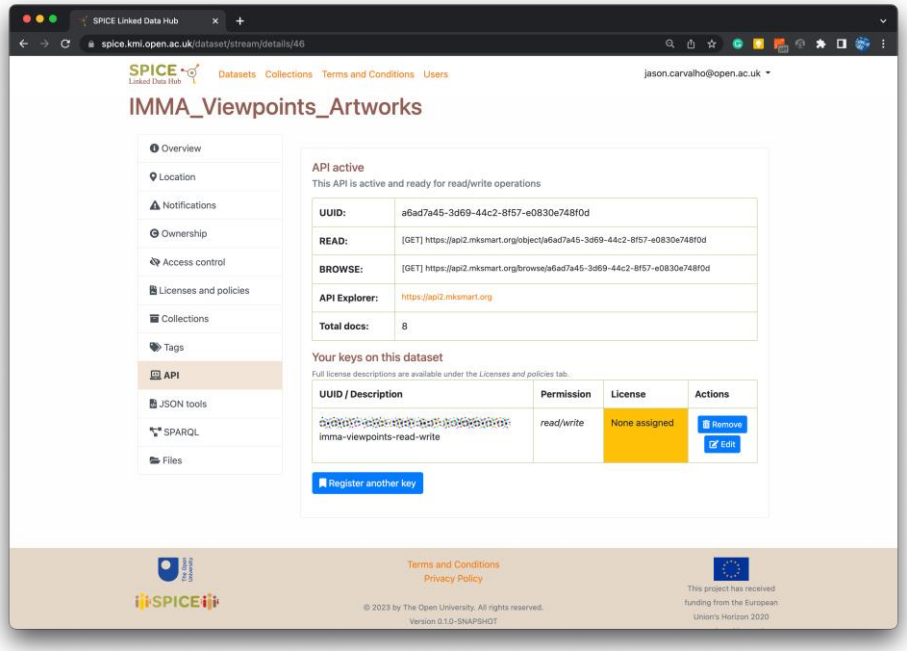

Figure 2.2.5.1 API key management

2.2.6. License and policy negotiation

The policy management layer of the Linked Data Hub addresses a large section of the original requirements for the Linked Data Hub, providing the policy data structures, workflows, licensing rules and user interface components to enable data owners and managers to effectively control the terms of use of their resources.

The policy management layer applies policies at a number of different levels and to various resources through the use of license resource and assignee scope. By offering this level of granularity, combined with appropriate user interface tools and the ability to build application-specific custom licenses, the policy management addresses many of the terms of use requirements initially set out for the LDH. Figure 2.2.6.1 illustrates the licence editing features, while Figure 2.2.6.2 shows how users can associate a security key to a specific licence.

A significant part of the development of this layer of the LDH is based around license and policy negotiation. Workflows and processes have been devised that show how we have addressed this issue, offering dataset owners and dataset users the chance to enter into an automated negotiation dialogue. This dialogue aims to produce a custom set of agreed terms of use for specific data resources. The component also offers tools that data owners and managers might use to manage license allocation and auditing (Figure 2.2.6.3). The licensing and policy management components are described in detail in deliverable D4.3 [SPICE D4.3].

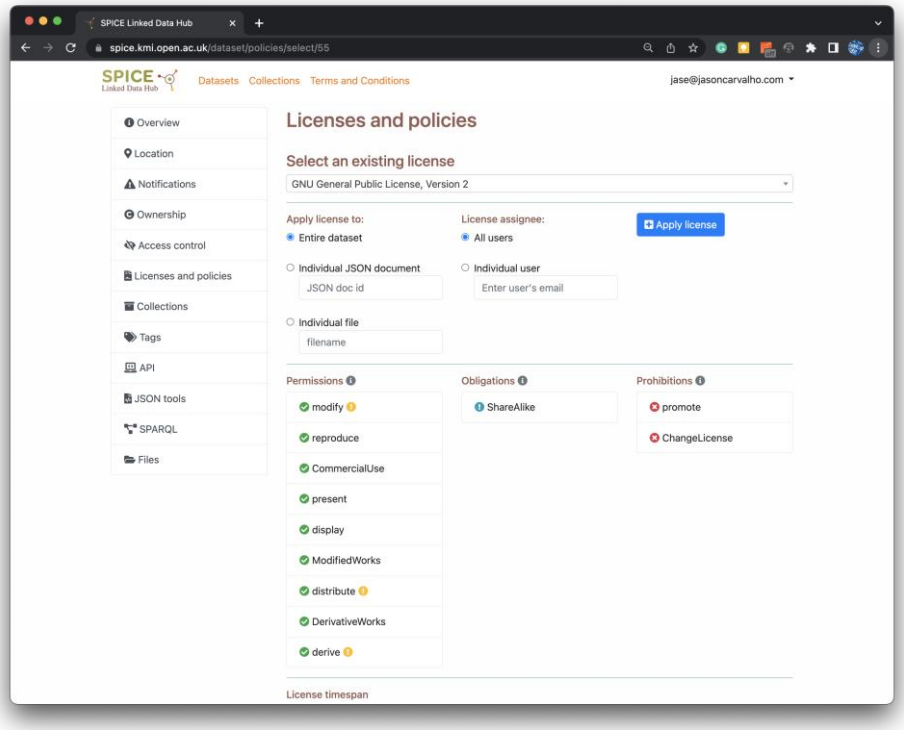

Figure 2.2.6.1 License selection

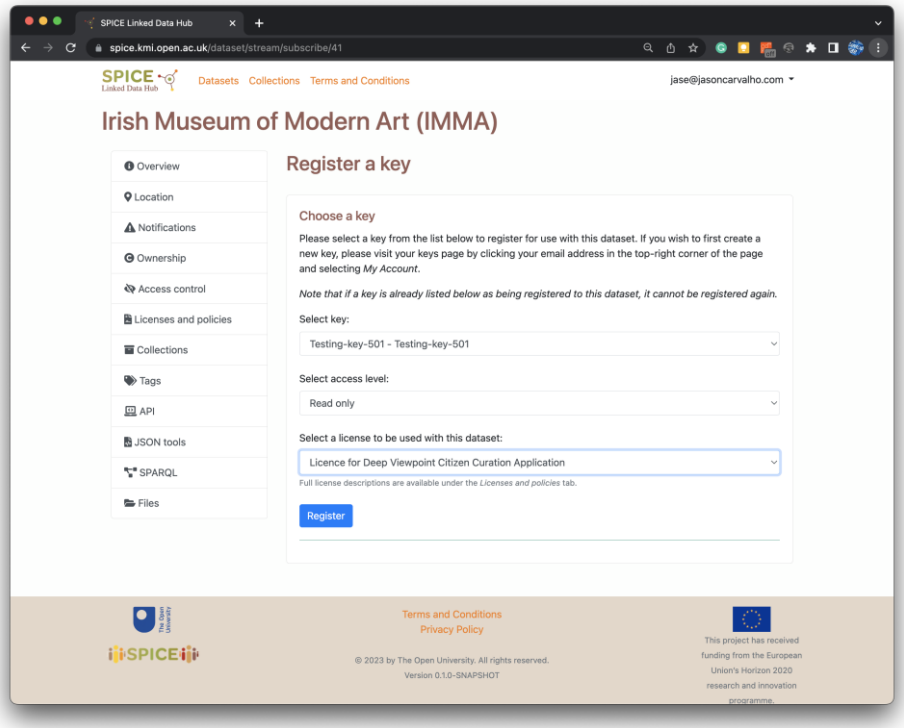

Figure 2.2.6.2 Key license allocation

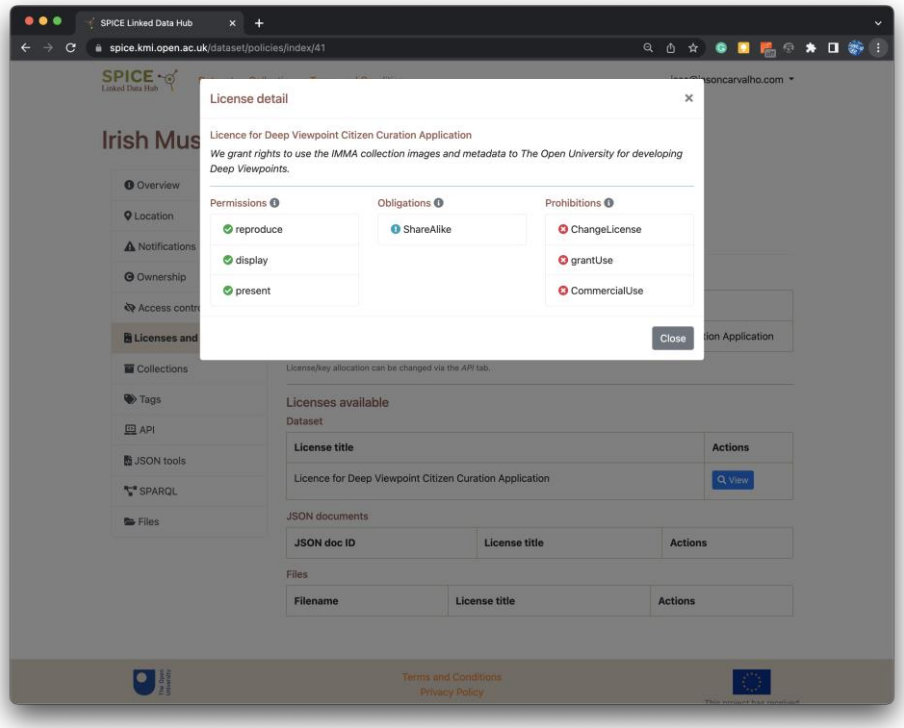

Figure 2.2.6.3 License detail and policy view

2.2.7. JSON tools

The SPICE LDH provides a suite of tools for directly interacting with the JSON documents within a dataset without the need for HTTP calls to the REST API (Figure 2.2.7.1). Whilst the full functionality of the LDH API is not available via the web interface, it does provide features for parameterised browsing of documents as well as creation, editing and deletion of documents (where API permissions permit). Typical web-based browsing features include access to the LDH API's query engine, sorting and paging of documents and control over specific field retrieval.

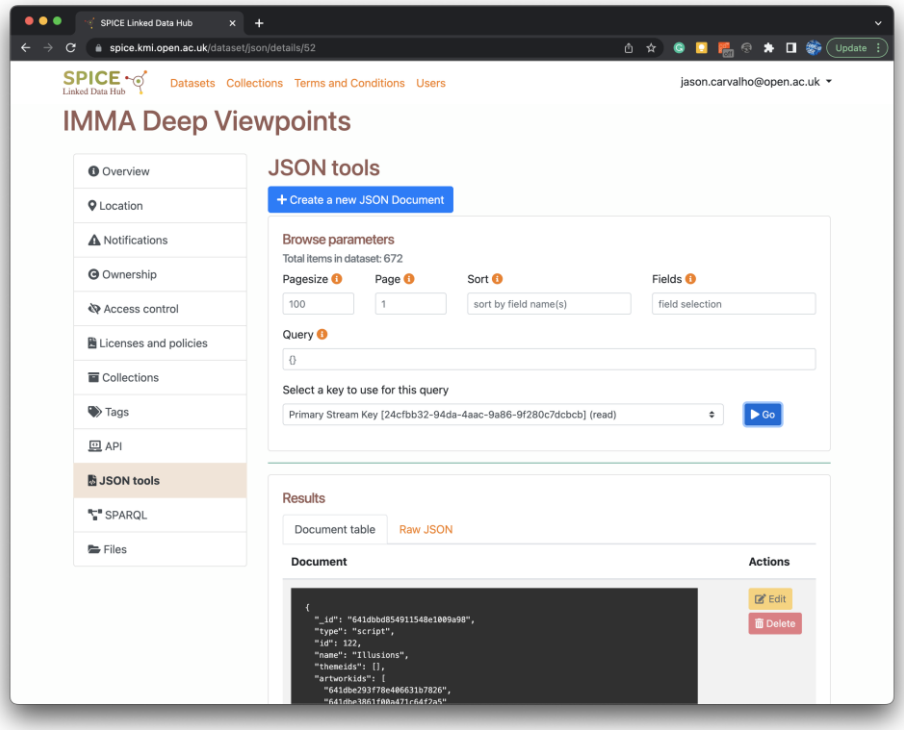

Figure 2.2.7.1 JSON tools

2.2.8. SPARQL, RDF replication and graph management tools

Using the same activity logging features as the content monitoring tools (2.3.1), we run an *RDF Uploader* component alongside the LDH that is responsible for the RDF replication of LDH data that drives the SPARQL querying API endpoint.

Specifically, this component continuously monitors the activity log and reacts to the logged events of creating, deleting, or updating JSON documents and static files. Whenever an event of this kind is detected, the RDF Uploader retrieves from the activity log's entry for the dataset and the document targeted by the logged activity and updates the triple store accordingly. This is described in detail in D4.2.

As well as the LDH API providing a SPARQL endpoint at which SPARQL queries can be directed, the LDH Portal also offers a series of user interface components to assist developers and dataset users with building SPARQL queries and managing RDF data. The UI components provide a web-based, context-sensitive SPARQL query editor and results viewer, shown in Figure 2.2.8.1.

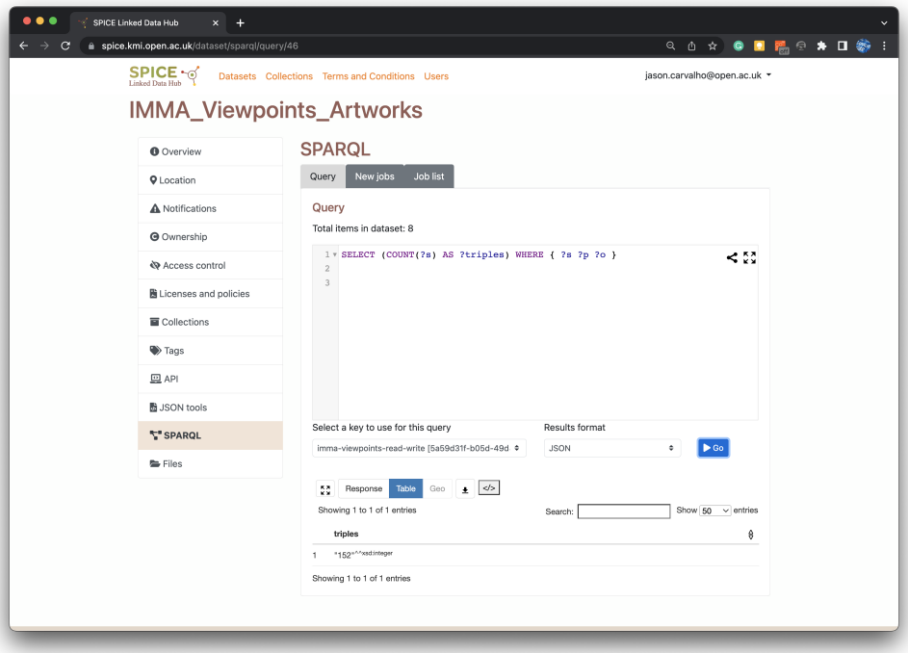

Figure 2.2.8.1 SPARQL query tools

Further development of this component offers users the ability to construct their own custom RDF graphs, materialising views on their data using SPARQL CONSTRUCT statements. These constructed graphs are stored within the same namespace as their parent dataset and are then available to be queried via the API's SPARQL endpoint.

These tools are made available to LDH users via the SPARQL tab for each dataset, offering user interface components to make SPARQL queries against their dataset, issue instructions for CONSTRUCT jobs that will create new graphs, trigger the rebuilding of document graphs and also entire dataset RDF namespaces (Figure 2.2.8.2). Full details on these features are given in deliverable D4.2 [SPICE D4.2].

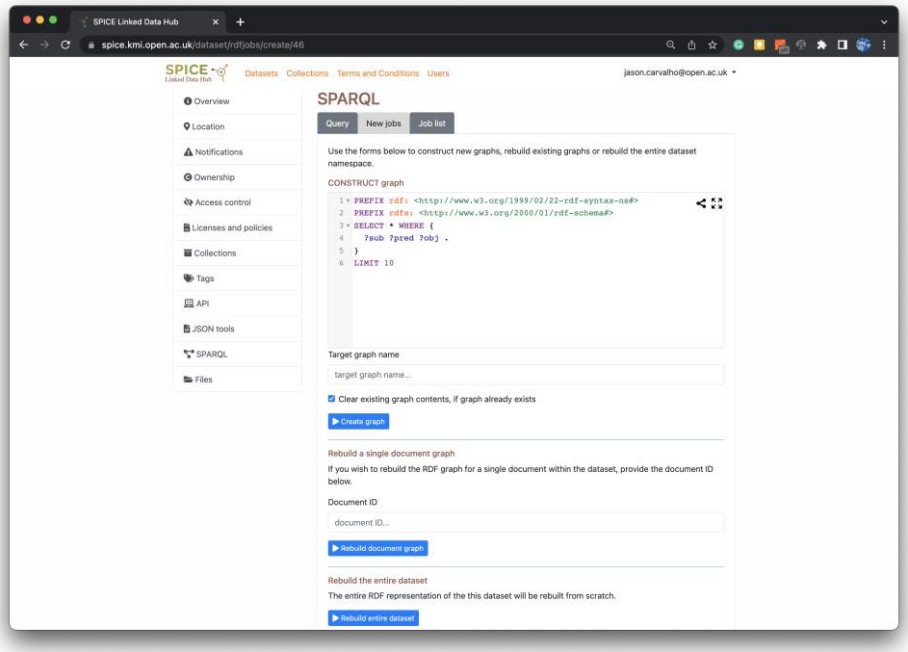

Figure 2.2.8.2 RDF graph construction and management tools

2.2.9. Notifications

Content monitoring features, described in section 2.3.1, have created a need for a notification system to be incorporated into the SPICE LDH, so that dataset owners can be informed and keep track of important notices related to their datasets. Notifications of instances where user generated content may have flagged potential issues in data such as personally identifiable information (PII) or hate speech are now kept and managed in dedicated area, shown in Figure 2.2.9.1. The process flow of issuing notifications also includes optional email notices to dataset owners and provides a framework for notification actions to be implemented for certain future notification types.

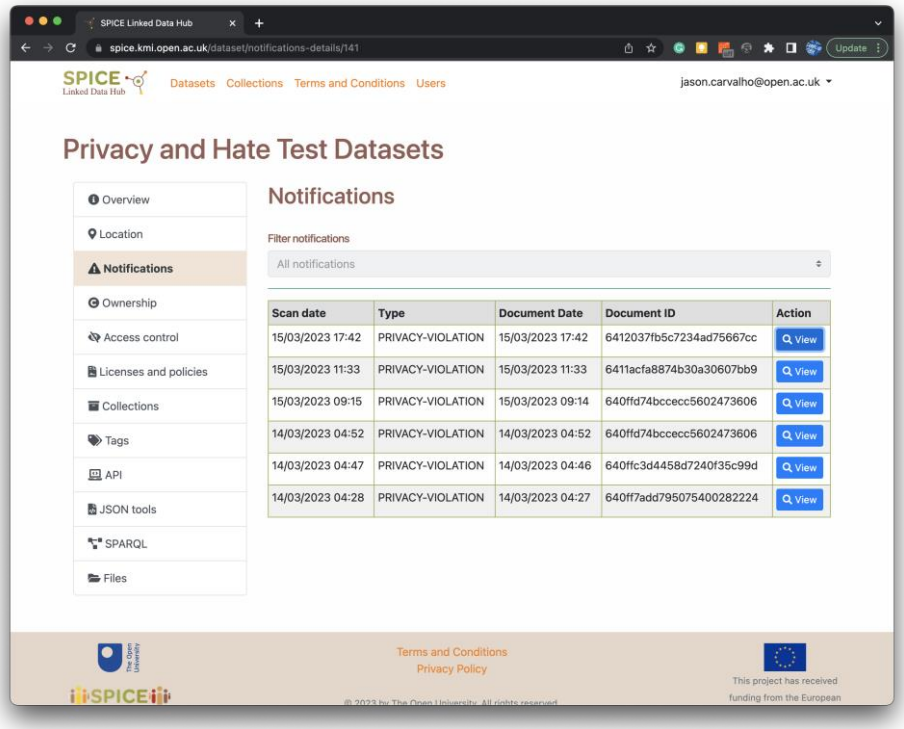

Figure 2.2.9.1 Dataset notifications

2.3. LDH parallel components

In addition to the functionality offered by the core components that make up the SPICE Linked Data Hub, several parallel components have been developed to augment the LDH services, assist LDH users with common tasks and address requirements that fall outside of the core features of the LDH. Generally, these components do not form part of the main LDH codebase, but sit as parallel components that run alongside the LDH and have specific lateral access to selected areas of the LDH's underlying database.

2.3.1. Content monitoring

Described in more detail in deliverable D4.3, content monitoring tools are now in place for regular scanning of new dataset content that violates rules laid out in specific scanning modules. Currently modules exist for the detection of personally identifiable information (PII) and hate/toxic speech. The content monitoring add-on makes use of the LDH's activity log; a logging dataset that records all API activity on every dataset and therefore all additions and updates made to the content of datasets. Activity log entries are made available through the standard LDH API using appropriate credentials. Through this mechanism, the content monitoring software can make regular HTTP API calls to check for new or updated documents.

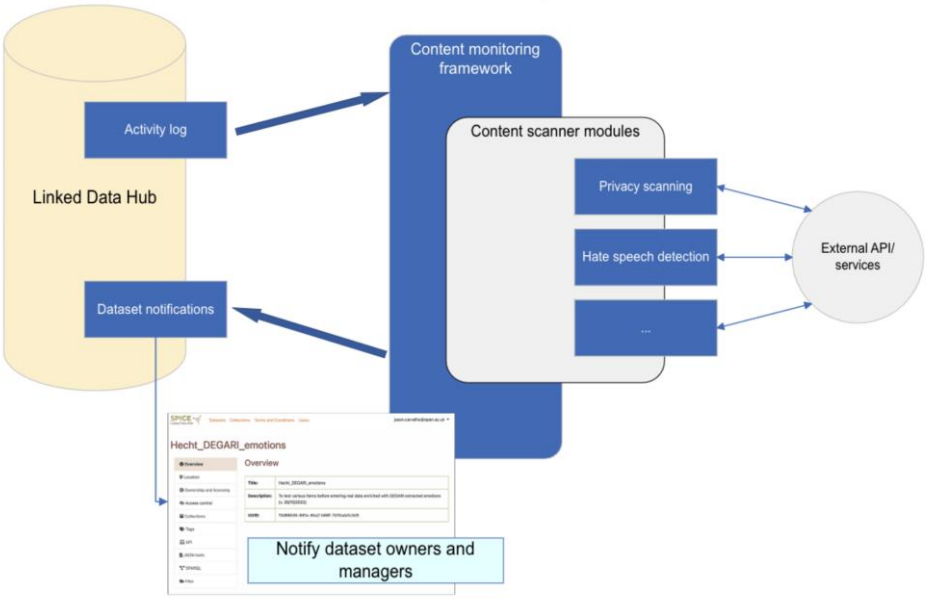

Figure 2.3.1.1 Content monitoring framework

2.3.2. RDF replication and graph management tools

The SPARQL UI components presented in section 2.2.8 are made possible by the RDF Uploader component. D4.2 contains detailed information on how this component that resides parallel the core LDH platform makes use of the LDH's activity log to provide an up-to-date linked data mirror of all LDH data in RDF form. In addition to being able to query LDH resources as linked data using SPARQL, the RDF Uploader also makes it possible for users to construct their own custom RDF graphs, materialising views on the data for use in further queries. The graph management tools used to facilitate this are described in detail in deliverable D4.2 [SPICE D4.2].

2.3.3. SPARQL Anything

SPARQL Anything was already introduced in D4.1 [SPICE D4.1]. The system is designed following our research on Knowledge Graph construction, which proposes the design of a façade, a notion borrowed from object-oriented software engineering [Asprino et al, 2023]. This idea is applied to SPARQL Anything, a system that allows querying heterogeneous resources *as-if* they were in RDF, in plain SPARQL 1.1, by overloading the SERVICE clause. SPARQL Anything supports a wide variety of file formats, from popular ones (CSV, JSON, XML, Spreadsheets) to others that are not supported by alternative solutions (Markdown, YAML, DOCx, Bibtex). Features include querying Web APIs with high flexibility, parametrised queries, and chaining multiple transformations into complex pipelines. The tool was applied in SPICE to support the ingestion of museums' data from legacy content management systems. SPARQL Anything is an open-source project[1].

3. Evaluation of LDH in relation to the requirements

Here we present the requirements for the work package, as set out at the beginning of the project and listed in Table 3.1. We are pleased to confirm that many of the outstanding requirements from previously submitted WP4 deliverables are now marked as complete. Notes on these requirements that have recently changed status follow below:

[1] SPARQL Anything http://sparql-anything.cc

- (2) *Backup Content*. We rely on robust and resilient IT infrastructure and system backups as a primary form of defence against data loss in the case of unexpected system failure. Further to this, the LDH maintains a detailed log of all activity that takes place via the LDH API. This activity log provides all the information required to establish the state of a given dataset resource at any point in time and can be used to retrieve lost data that may have been erroneously overwritten or removed.
- (4) *Browse marketplace*. We mark this requirement at 75%. Currently, we offer a marketplace of datasets and curated dataset collections for browsing and use. Expanding this to individual digital assets (JSON document entries, digital images, and their associated metadata) is a natural evolution from here. The whole life cycle of individual digital assets, from onboarding and curation through to policies, access negotiation, and final use, will follow the same patterns that have been employed at a dataset level. This granular marketplace offering was not required by SPICE pilot applications, however some applications such as IMMA Deep Viewpoint implemented similar features of their own, using the LDH as an underlying platform. In this case, IMMA Deep Viewpoints can be considered a prototype for future LDH features that support more granular offerings of digital assets and we plan to investigate how some of this functionality can be integrated into the LDH core.
- (16) *Filter sensitive content*. This is now marked as complete, following the work done on content scanning for personally identifiable information (PII) and hate speech. Sensitive content can now be flagged for moderation by dataset owners and managers, with a framework in place for adding additional scanning modules should the need arise.

Over the course of the project, and specifically through the support of the development of SPICE pilot applications, s, requirements emerged that required either new, unplanned features or for other LDH requirements to be refined and modified slightly. It also became apparent that some of the original requirements were not relevant, once the work package team began to gain a better understanding of the needs of the pilot applications and the data systems to support them. A short discussion of the requirements that were not fully completed or that have been considered as potential for future development follows below:

- (9) *Express fees*. Fees for the use of digital resources are currently not expressed within the LDH framework. This has not emerged as a requirement by any of the LDH users thus far, although we see no issue with incorporating this into future development should the need arise.
- (13) *Express quality features*. Again, this has not emerged as a requirement for any of the LDH users or SPICE pilot application developers and we suspect it may not be needed in general. The decision was therefore made put this requirement aside.
- (32) *Recognised author*. Further consideration of this requirement led to the conclusion that this is a domain-related problem. The unstructured nature of data storage within LDH datasets gives dataset owners and application developers the freedom to manage and structure their data and metadata in whichever way best suits their particular application. Where appropriate, artwork collection managers are maintaining recognised author details themselves, within their datasets and applications. Managing this at LDH-level would narrow the domain scope of the system unnecessarily.
- (33) *Register sources*. This requirement was considered to be unnecessary, since the focus of the LDH platform was on onboarding and integrating collections of data. The data brought in was largely not in linked data form, which is the reason RDF replication and RDF graph construction tools were developed to allow users to perform linked data operations on locally hosted datasets.

Table 3.1. WP4 requirements

	Nickname	Role	Action	Target	D4.7 Status
1	[AnalyseUsage]	custodian/owner	analyse	access and usage of my data	100%
2	[BackupContent]	data manager	backup/restore	my data to support recovery in the case of a loss event.	100%
3	[BrowseIndex]	builder	browse	an index of the resources I have access to	100%
4	[BrowseMarketplace]	custodian/owner/builder	browse	a marketplace of offers of digital assets	75%
5	[ControlMetadata]	owner/custodian	control	the metadata production in the ingestion process	0%
6	[DetectPII]	custodian/builder	detect	personally identifiable information (PII) included in user-generated content	100%
7	[ExpressCopyright]	custodian/owner	express	the copyright associated with digital assets in my collection	100%
8	[ExpressExemptions]	custodian/owner	express	exemptions and characterize them	100%
9	[ExpressFees]	owner	express	fees as duties associated to the permissions granted	0%
10	[ExpressOffers]	owner	express	offers with relation to the assets I own.	100%
11	[ExpressPermissions]	owner	express	permissions, prohibitions, constraints and duties	100%
12	[ExpressPolicies]	custodian/owner	express	usage policies in relation to my data	100%
13	[ExpressQualityFeatures]	custodian/builder	express	the quality of the asset and their features	0%
14	[ExpressTimeConstraint]	owner/custodian	express	time limitations to permissions I grant	100%
15	[ExternalAccessData]	builder	access	data from an external application	100%
16	[FilterSensitiveContent]	custodian/builder	filter	sensitive content for specific target groups	100%
17	[GrantCheck]	builder/custodian/owner	verify	lawful access to a collection metadata or digital asset	100%
18	[GrantRecovery]	owner/custodian/builder	view	terms of use granted	100%
19	[InappropriateContent]	custodian/builder	identify/filter	user-generated content that can be inappropriate	100%
20	[InspectIngestionProcess]	owner/custodian	inspect	the metadata production in the ingestion process	100%
21	[ManageAccess]	data manager/custodian/owner	manage	access control to the data	100%
22	[ManageVisibility]	data manager	manage	visibility of my registered data sources	100%
23	[MonitorAccess]	data manager/custodian/owner	monitor	access to my data	100%
24	[MultipleRightsAspects]	custodian	express	that multiple subjects hold copyrights on different aspects of the digital asset	100%
25	[NegotiateRights]	custodian	negotiate	rights on behalf of the owner	100%

26	[NominateDelegate]	custodian	nominate	an external entity to negotiate rights on behalf of a copyright owner	100%
27	[ObtainCredentials]	builder	obtain	credential details (e.g., API Keys) to data	100%
28	[OrganiseCollections]	custodian/builder	organise	resources I have access to into customized collections	100%
29	[ProduceLD]	data manager	produce	linked data from existing non-LD resources	100%
30	[PublishLD]	data manager	publish	linked data with alternative Linked Data vocabularies (Viewpoints)	100%
31	[ReadData]	builder	read	data from a dataset –e.g., via a (secured) Web API	100%
32	[RecognisedAuthor]	owner	be_recognised	as author of the picture of the artwork	0% N/A
33	[RegisterSources]	data manager	register	existing Linked Data sources	0% N/A
34	[RequestAccess]	builder	request	access to data	100%
35	[RequestPermission]	builder	request	permission to use a digital asset under specific terms	100%
36	[RevokeRights]	owner/custodian	revoke	usage permissions I granted in the past	100%
37	[SecureStack]	data manager	secure	the content against malicious attacks	100%
38	[SetupRepository]	data manager	setup	a data repository	100%
39	[ShareCollections]	custodian/builder	share	my customized collections as linked data	100%
40	[UploadDataset]	data manager/owner/custodian	upload	data to my dataset	100%
41	[UsagePolicyGrant]	owner/custodian	grant	permission to use a digital asset under requested terms	100%
42	[WriteData]	data manager/builder	write	data to a dataset –e.g., via a (secured) Web API	100%

4. SPICE pilots: applications and feedback

This section of the report provides a summary of the SPICE pilot applications that have been developed that make use of the linked data server technology described in this deliverable. For each pilot application, we present feedback from the pilot leaders and developers on their use of the SPICE Linked Data Hub and the process of its integration with their applications.

Further details on each pilot, including technical requirements on LDH integration and application workflow, are described in deliverable D4.2 [SPICE D4.2].

4.1. IMMA Deep Viewpoints

4.1.1. Background

The IMMA case study is concerned with developing tools to support Citizen Curation in the form of two distinct processes:

i. *mediation*, in which museum visitors select one or more artworks, associate them with a theme and develop a script comprising a sequence of stages of contextual information and questions that guide visitor interpretation

ii. *interpretation*, in which museum visitors select a script associated with a theme and develop and share their own interpretation by following the stages of the script

Two applications working with the LDH have been developed: IMMA Viewpoints and IMMA Deep Viewpoints.

IMMA Viewpoints was the first application developed and focussed on the process of interpretation. Built as a website intended for mobile use, IMMA Viewpoints could be loaded with a set of artworks and a set of questions stored on the LDH. A visitor could view information about the sculpture and respond to a randomly selected question or reload a new question. The responses were queued for moderation by IMMA. The moderated responses were shown on an "Other People" page where the visitor could find out how others had responded to the artwork.

IMMA Deep Viewpoints (later referred to as *IMMA Slow Looking*) was subsequently developed to support both the mediation and interpretation processes. The mediator (who could be a member of IMMA staff or citizen community group) could create a script that guided the visitor through a set of stages containing statements, questions and IMMA artworks. The interpreter could develop and share their interpretation by proceeding through the stages of the script. Script owners or IMMA staff could moderate the responses to any script.

4.1.2. Feedback

The SPICE LDH has been responsive and reliable throughout both the Viewpoints trial (July to October 2021) and the Deep Viewpoints trial (since November 2021).

The artwork image files used within Deep Viewpoints are held in the IMMA online collection rather than in LDH. This means broken links can occasionally occur when artwork locations change in the IMMA online collection. This is resolved by refreshing the artwork links in the LDH. Caching local versions of the artwork images on the LDH would resolve this problem, although this has raised a number of potential copyright issues which would need to be resolved before pursuing this solution.

Deep Viewpoints (and also Viewpoints) are Angular apps that communicate asynchronously with the LDH. This enables a seamless, responsive interaction with the content. However, it requires the use of different design patterns, particularly when the app is being used collaboratively in real-time by multiple clients. This may not necessarily indicate a missing LDH feature but rather the need to adopt appropriate design patterns when consistency is required across multiple clients.

4.2. Hecht Museum

4.2.1. Background

School students before, during and after a museum school trip at the Hecht Museum learn about their country's history and at the same time learn about the diversity of opinions regarding historical and national issues. Students learn to interpret museum artifacts according to their own personal views, reflect upon other students' opinions, connect their opinions with tangible artifacts at the museum, and perform citizen curation activities. Teachers, educators and museum curators have been involved in the design of the activities and application, with teachers providing information and feedback on the educational goals of the system. We have designed and built a web application that supports students in the classroom, during the museum visit (on handheld devices), and after the visit. In addition, we have an application which supports the teachers, curators and researchers to evaluate an and analyse the data. The data consists of content generated by the users (photos, descriptions and tags), user answers to surveys, users' responses to questions, user model and history of user interaction with the system.

4.2.2. Feedback

The case study makes use of the LDH directly through the User Model and storing User Generated Content. The LDH covers all necessary functionality. The case study was originally developed with Java Spring Boot, making use of JPA (Java Persistence API) and a MYSQL database. The LDH covers the essential functionalities of this case study (e.g., files repository), and migration and integration were successful. In addition, the Social Recommender was developed on top of data contained within the LDH.

The browse interface, provided by the JSON Tools component, was very useful as it allowed for both queries and trimming the amount of data sent across the network by using the fields parameter. The response time is reasonable. In addition, both the website API interface and the JSON Tools were extremely useful and saved a great deal of time. Given that the LDH is a RESTful Web API, we were able to implement parts both in JAVA and Python.

4.3. Helsinki Pop-up VR museum

4.3.1. Background

The focus of the Design Museum Helsinki (DMH) Case Study is on developing the citizen curation methods by first gathering interpretations of DMH collection objects in workshops with selected end-user communities, namely senior citizens, remote dwellers, and asylum seekers (D7.3 [SPICE D7.3], pg33). An application known as the *Pop-up VR Museum* has been designed and made accessible to audiences via portable VR headsets. Its users can access, interact, and engage with Design Museum Helsinki's collections. The target audience of the pilot include three different end-user communities, namely senior citizens, rural communities, and asylum seekers. In this process, the mediators play an important role in aiding the application's development and guiding its users; these mediators are often comprised of curators, researchers, and members of institutions such as senior care centres.

4.3.2. Feedback

The feedback from the DMH team on the integration with the LDH has been largely positive, although there are areas that could be developed further to make the operation between the two systems more seamless. The development team have been making good use of a selection of LDH features, but the Pop-up VR Museum application is not fully integrated with the LDH platform.

The application has been developed in a hybrid style and doesn't make use of the LDH for all of its data storage requirements. Specifically, it does not make use of real-time fetching of data from the LDH for several reasons:

1. The application does not need to make use of constantly changing information.
2. There is not always a reliable internet connection at every location we take the VR equipment to.
3. Data is coming through in different languages and curators need to collect, transcribe, and translate before feeding back.

The application does, however, make use of the LDH API to update the dataset of user interactions in the application such as the gameplay time, artefacts interacted with, and so on. This has worked well.

The JSON Tools component has been instrumental in the development and testing of this application. It has also been useful in collaborating with other work package streams (such as semantic annotation and sensemaking activities), enabling others to easily query and work with our datasets.

It was noted in D4.2 that it would be useful for the LDH to enable curators, mediators, and researchers to add datasets about artefacts and narratives. Specifically, the API features supporting binary file storage and retrieval would support multimedia resources such as 3D models and audio descriptions of artefacts to be exchanged alongside traditional textual information using the existing JSON API. We still feel further integration between binary file storage and JSON documents would be useful, moving forwards.

4.4. GAM Game

4.4.1. Background

The case study of the Gallery of Modern Art (GAM) in Turin, which addresses the inclusion of deaf people as target community, revolves around the notion of storytelling. Through the web application users can create short stories by collecting and sequencing the artworks from the museum collection, and add a personal response to each of the artworks in the story. Stories can be created at any time before, during and after the museum visit.

Stories can be saved in the user's personal space associated to her/his account and later deleted (but not edited). In order to identify them, and also as part of the creative process, users are prompted to give a title to each story as a precondition to saving it.

After creating each story, the user receives a set of recommendations of artworks based on the emotions assigned by the DEGARI (see deliverable D6.3) reasoning service to the artworks in the story (in turn, these associations are inferred by DEGARI from the text content which accompanies the artwork: artwork description from the collection record, user comments and so on). Recommendations are of two types: they concern artworks associated with the same or similar emotions, and artworks associated with emotions of opposite polarity. If accepted, a recommended artwork will be attached to the story.

Once shared, stories are associated with the artworks in the museum collections, and can be browsed by the other users, who can express their appreciation to them through likes, in the style of social media.

4.4.2. Feedback

The feedback given from the GAM Game application team in D4.2 [SPICE D4.2] concerns the difficulties of providing near real-time use of sense-making services provided in WP6. The LDH provides facilities for requesting updates on the changes that have occurred in datasets, but these requests must be instigated by the client application and be made frequently. This leads to both redundant and unnecessary API calls and also periods where artwork recommendations may be out of date.

This feedback is still relevant now. A push-based architecture, where updates flow in the direction from the LDH to the ontological reasoning service, and not the other way around, would be better suited to this scenario and would improve the integration potential of applications making use of the LDH, enabling them to have a more direct connection with their underlying data.

It has also been noted during the development of the GAM Game application that the SPICE LDH would benefit from some form of version on control on its resources. This would enable developers and users to monitor the lifecycle of data and easily track and audit the history of JSON documents and file resources within the LDH.

4.5. Madrid Treasure Hunt

4.5.1. Background

The case study of the Natural Musem of Natural Sciences of Madrid (MNCN) revolves around treasure hunts in the museum. A treasure hunt consists of a series of searches guided by clues describing the object in the collection to be found. Once the object is found, the game provides relevant information related to it and may pose related questions.

The activity is designed for a group of schoolchildren led by their teacher. The teacher has the possibility to design a specific treasure hunt for her students, and thus choose the theme of the game, the selection of objects in the collection, the information to be shown and the questions to be asked.

The game itself is an application developed in Unity 3D that runs on a tablet. The definition of the content of the treasure hunt is done through a web application where the teacher can create the content and where she also has access to other treasure hunts created previously.

The answers to the questions posed during the quest are collected and provided to the teacher so that she can analyse the results of the activity.

The pilot application is used for both the building and creation of treasure hunt activities and also the execution of the treasure hunts and collection of user responses and interaction.

4.5.2. Feedback

I was noted in D4.2 [SPICE D4.2] that the LDH lacked some basic data browsing functions outside of the main REST API provided to developers. There was also a requirement to manually add data items to the dataset from time to time, a slow process when having to make API calls to do this without a web interface.

The addition of JSON Tools component in the LDH has been a great help in creating and manipulating items in the different use cases (as an example, it is now much easier to fix bugs or make repetitive formatting changes to JSON documents that describe activities or user responses). This is a *quality of life* improvement for simple but tedious modifications such as replacing the formatting of images in a collection of works to change their prefix, which can now be done without having to write a single line of code or HTTP request. It has also helped considerably in debugging applications that make use of LDH thanks to the option to search using JSON/MongoDB style queries. While the latter could previously be done via SPARQL or HTTP queries, the ability to do this directly from the web tool is invaluable and again, greatly speeds up the development process.

It was also noted in D4.2 that, while the LDH provides some functionality for the storage of multimedia files, this is handled as a separate component and not linked tightly with the storage of JSON documents. When using JSON documents with rich metadata and accompanying multimedia files, the link between both must be handled manually and integration of them is not automatic.

Within the scope of this use case, the application was simply switched to hosting the images and other multimedia resources needed in separate databases and servers, referenced via links in LDH documents.

5. Moving forwards

Following on from the feedback given from pilot applications in D4.2 [SPICE D4.2], significant development work was undertaken in the last year to address many of the needs of these pilots whilst also working toward completion of the original WP4 requirements. It is encouraging to hear in this latest round of feedback that many of the LDH developments that were made have been well received and are helping to streamline the process of application development and integration.

In terms of areas for improvement, there are some consistent themes that arise across the feedback from several pilot applications, both as reported in this deliverable and as shared anecdotally over the last period of development.

Whilst the JSON Tools component has been well received and has helped users more quickly and effectively get applications up and running, it has given inspiration to some other similar tools that may also help speed up the process of application development and deployment. A future expansion of this component would allow for a suite of bulk processing functions. Specifically, the ability to bulk upload multiple JSON documents (either located in a larger JSON container document or uploaded as a zipfile) and also do bulk deletes and edits has been mentioned by several developers.

It has also been noted that the basic file handling of the LDH and its API could be extended to be more integrated with the existing JSON store. Storage of associated file metadata is currently handled manually by developers, leaving them with the responsibility of creating associated JSON documents and maintaining links between the two. A potential solution to this problem would be automatic JSON document entries for

all uploaded files. The entries could be extended by users to incorporate their own custom metadata, facilitate version control and provide a means of simple JSON<->JSON linking between LDH entities for file referencing from within LDH applications.

With the content scanning features mentioned in section 2.4.2 and described in detail in D4.6 [SPICE D4.6], we have begun to move the LDH from being a simple content store into the realm of being a content-aware system. These really are the first steps into a notification-only set of features, but the potential for further development in this area is significant. Work on the pilot applications, and the IMMA apps in particular, has shown the importance of content monitoring and also content moderation in citizen curation activities. IMMA has trialled the use of PII and hate speech detection modules with Deep Viewpoints. However, currently, it is not integrated into the Deep Viewpoints moderation workflow. Moving forwards, it would be useful to explore integration and whether this could support the semi-automated moderation of content contributed to Deep Viewpoints.

We also see scope here for the incorporation of content moderation facilities to be built directly into the LDH. These features can then be optionally used by application developers and integrated into applications with appropriate API functions. The scope for actioning scanned content also extends beyond simply the approval of user generated content (UGC). Additional operations on UGC may include options to delete documents, automatically remove/amend selected parts of documents or trigger other actions such as the creation of new content or sending of custom email notifications.

6. Packaging and deployment

In this section we present the final packaged version of the SPICE Linked Data Hub, links to the resources required for obtaining a copy of the LDH platform and instructions for deploying new LDH instances.

6.1. Docker installation

For ease of deployment, the LDH and its constituent components are made available here as a series of Docker[2] containers. Docker is a popular tool used to create, distribute, and deploy software applications. In this documentation, we will discuss the steps involved in downloading and running the Docker containers and how to access the resulting LDH services once they are up and running.

Installing and running the containers

i. Install Docker. If Docker is not already installed on your system, follow the instructions on the Docker website[3] to install it.

ii. Download the latest version of the SPICE LDH docker environment from its GitHub repository, at https://github.com/spice-h2020/spice-ldh-docker, using the command:
```
> git clone https://github.com/spice-h2020/spice-ldh-docker.git
```

iii. Within the repository directory, build the containers:
```
> cd spice-ldh-docker
> docker compose build
```

iv. Run the Docker containers:
```
> docker compose up
```

Accessing the LDH services

Once the containers are running, the LDH services are available locally at the following URLs:

[2] https://www.docker.com/
[3] https://www.docker.com/get-started/

- http://localhost:8091/ - ***LDH Portal***. This is the main LDH web interface. On first install, an admin user is automatically created with the following credentials:

  username: *admin@example.com*
  password: *password*

  Once logged in, additional users can be created. You should start by creating your own administrator user account. Once logged in using this account, you can begin creating datasets and access keys.

- http://localhost:8090/ - ***LDH API***. Use of this web interface isn't strictly required, although it is a useful tool for listing all API functions in a single place and providing an interface for testing API calls used in application development. Full documentation on the LDH and its web interface is covered in deliverable D6.8.

- http://localhost:8099/ - ***LDH graph database (Blazegraph)***. This is not required for normal LDH use, but the graph database is visible at this address for debugging, inspection and SPARQL query testing purposes.

Stopping and starting the services

Once running, the Docker containers will be shown in the Docker desktop application as green, as shown in Figure 6.1.1.

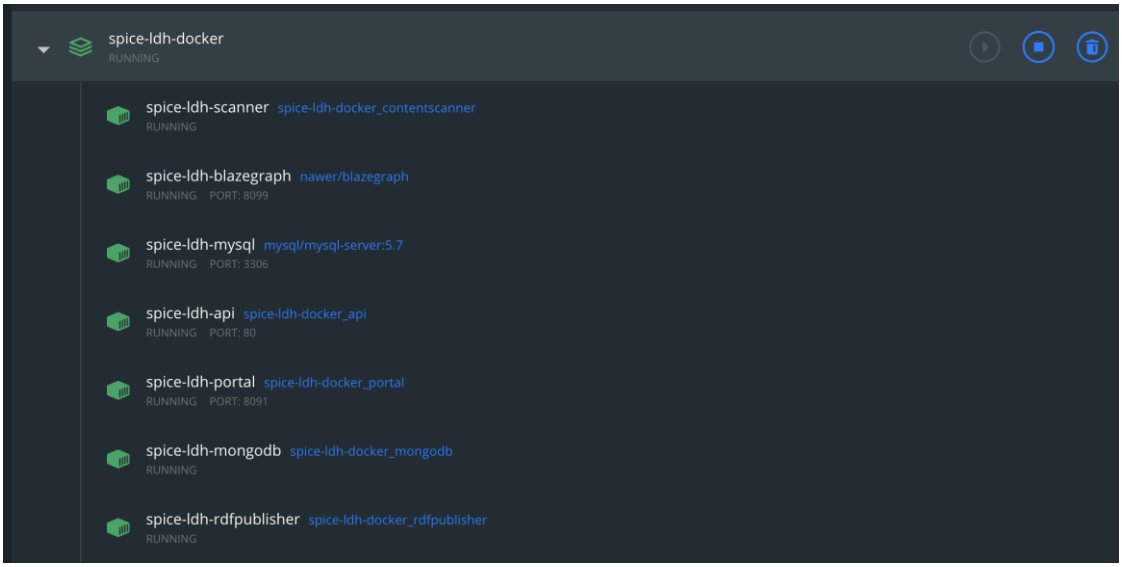

Figure 6.1.1 Docker desktop application

The entire SPICE LDH Docker environment can be started and stopped using the start and stop buttons shown in the top-right of Figure 6.1.1. On the first creation of the containers, blank datasets will be initialised for all the required LDH storage. After this, stopping and starting the containers will persist this storage so data is not lost. Note that deleting and rebuilding the containers will lose the existing data storage and initialise the LDH environment with a new fresh and empty data storage component once again.

6.2. Software components

The SPICE Linked Data Hub is made up of a series of software components. These are all made available publicly at the following locations

- LDH API - https://github.com/mkdf/api-factory
- LDH API SPARQL module - https://github.com/mkdf/api-factory-sparql
- LDH portal - https://github.com/spice-h2020/linked-data-hub
- mkdf-core - https://github.com/mkdf/mkdf-core
- mkdf-datasets - https://github.com/mkdf/mkdf-datasets
- mkdf-topics - https://github.com/mkdf/mkdf-topics
- mkdf-stream - https://github.com/mkdf/mkdf-stream
- mkdf-keys - https://github.com/mkdf/mkdf-keys
- mkdf-sparql - https://github.com/mkdf/mkdf-sparql
- mkdf-policies - https://github.com/mkdf/mkdf-policies
- LDH content scanner - https://github.com/spice-h2020/ldh-scanning-framework
- RDF uploader - https://github.com/spice-h2020/rdf.uploader
- SPICE LDH Docker environment - https://github.com/spice-h2020/spice-ldh-docker

7. Conclusions

This report presented the deliverable of WP4 of the SPICE project, focusing on the development of the SPICE Linked Data Hub and associated technical components. It contains a full breakdown of the core components of the LDH platform along with details of the packing of these. Instructions are given on how to create and deploy blank instances of the SPICE LDH, using a series of prepared Docker images. It also provided reports from SPICE pilot applications that make use of the Linked Data Hub, offering feedback on the integration process and areas that could be focussed on in any future LDH development.

8. References

[Asprino et al, 2023] Asprino, Luigi, Enrico Daga, Aldo Gangemi, and Paul Mulholland. "Knowledge Graph Construction with a façade: a unified method to access heterogeneous data sources on the Web." ACM Transactions on Internet Technology 23, no. 1 (2023): 1-31.

[Daga et al, 2021] Daga, Enrico; Asprino, Luigi; Damiano, Rossana; Daquino, Marilena; Agudo, Belen Diaz; Gangemi, Aldo; Kuflik, Tsvi; Lieto, Antonio; Maguire, Mark; Marras, Anna Maria; Pandiani, Delfina Martinez; Mulholland, Paul; Peroni, Silvio; Pescarin, Sofia and Wecker, Alan (2022). Integrating Citizen Experiences in Cultural Heritage Archives: Requirements, State of the Art, and Challenges. Journal on Computing and Cultural Heritage, 15(1) pp. 1–35. DOI: https://doi.org/10.1145/3477599

[SPICE D4.1] "Linked data server technology: requirements and initial prototype," SPICE Project, European Union's Horizon 2020 grant agreement No 870811, vol. Deliverable 4.1, 2021.

[SPICE D4.2] Linked data server technology: integrating feedback from use case requirements," SPICE Project, European Union's Horizon 2020 grant agreement No 870811, vol. Deliverable 4.2, 2022

[SPICE D4.3] Distributed Privacy and Policy Layer, SPICE Project, European Union's Horizon 2020 grant agreement No 870811, vol. Deliverable 4.3, 2022

[SPICE D4.5] Provenance and process analysis layer: requirements analysis, SPICE Project, European Union's Horizon 2020 grant agreement No 870811, vol. Deliverable 4.5, 2022

[SPICE D4.6] Provenance and process analysis layer: supporting use cases, SPICE Project, European Union's Horizon 2020 grant agreement No 870811, vol. Deliverable 4.6, 2023

[SPICE D7.3] Case studies: progress and plans, SPICE Project, European Union's Horizon 2020 grant agreement No 870811, vol. Deliverable 7.3, 2021