



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



Social cohesion, Participation, and Inclusion
through Cultural Engagement

D4.1 Linked Data server technology: requirements and initial prototype

Deliverable information	
WP	4
Document dissemination level	PU Public
Deliverable type	R Document, report
Lead beneficiary	OU
Contributors	UNIBO, UNITO, IMMA, CELI, UH, UCM
Date	01/05/2021
Document status	Final
Document version	v1.0

Disclaimer: The communication reflects only the author's view and the Research Executive Agency is not responsible for any use that may be made of the information it contains

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	CELI	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 D4.1 presents the interim deliverables of Work Package 4 of the SPICE project, focusing on the Linked Data server technologies. After describing the content and scope of the document in the introduction, the report presents insights on the Citizen Curation paradigm it performs an analysis of the state of art technologies for Linked Data content management. Furthermore, it provides insights on the status of data management in SPICE museums. After devising requirements for a data infrastructure based on the performed analyses, the report describes 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. Specifically, the SPICE Linked Data Hub supports a privacy-aware data sharing platform and innovative technologies for publishing Linked Data from legacy resources. Finally, the report shows preliminary work on applying the infrastructure to develop a dashboard for data-driven decision making, targeting museum curators as primary users, and social media content as first-class citizen.

Document History

Version	Release date	Summary of changes	Author(s) -Institution
V0.1	15/02/2021	Deliverable structure and initial plan for content	Enrico Daga (OU)
V0.2	06/04/2021	Completed draft of sections 1-6	Enrico Daga, Paul Mulholland, Jason Carvalho (OU); Marilena Daquino (UNIBO); Rossana Damiano, Antonio Lieto (UNITO); Tsvi Kuflik (UH); Alessio Bosca (CELI); Belen Díaz (UCM)
V0.3	09/04/2021	Complete draft	Enrico Daga (OU)
V0.4	18/04/2021	Internal review	Guillermo Jimenez (UCM), Alan Wecker (UH)
V0.5	23/04/2021	Version released to all partners for final integrations	Enrico Daga (OU)
V1.0	30/04/2021	Final version submitted to REA	UNIBO

Table of Contents

Project information	3
Project contacts.....	3
SPICE consortium.....	3
Executive summary	4
Document History.....	5
1 Introduction.....	8
1.1 Description of the work package and objectives	8
1.2 Description of the work done so far in relation to the objectives	8
2 Outline	8
3 Supporting Citizen Curation	9
4 State of affairs	11
4.1 Related work on Data Sharing Infrastructures for CH	11
4.1.1 Data management for the cultural heritage.....	11
4.1.2 Web technologies for metadata publishing and exchange	14
4.2 Survey of content management practices of museums in the SPICE consortium.....	16
4.3 Analysis.....	20
5 Requirements	22
6 Architecture layout.....	26
7 Linked Data Transformations with SPARQL Anything	28
7.1 Requirements of LD transformers.....	29
7.2 Approach and solution design	30
7.3 Description of the system and features.....	35
7.3.1 Facade-X.....	35
7.3.2 Querying anything	35
7.3.3 Supported formats.....	37
7.3.4 Functionalities and Command Line Interface	38
7.4 Example LD Transformer: the IMMA catalogue.....	39
7.5 Evaluation.....	49
7.5.1 Cognitive Complexity Comparison.....	49
7.5.2 Requirements' satisfaction and discussion	50
8 The SPICE Linked Data Hub v1.0.....	52
8.1 Concept	52
8.2 Software components.....	53
8.3 Web Portal	55
8.4 Web API	61
8.5 Data	64
9 Proof of concept: Social Media and citizen curation activities dashboard	65

10	Conclusions and future work.....	68
11	Research outputs.....	69
Annexes		70
A. SPICE Linked Data Hub User Guide		70
References		74

1 Introduction

1.1 Description of the work package and objectives

SPICE is an EU H-2020 project dedicated to citizen curation of cultural heritage through an ecosystem of methods and tools co-designed by an interdisciplinary team of researchers, technologists, and museum curators and engagement experts, and user communities. Work Package 4 is dedicated to the development of 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 develop an infrastructure, based on the Linked Data principles, to connect cultural objects, collections, and citizen contributions, to be the *backbone* for interoperability and knowledge exchange within SPICE 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.

1.2 Description of the work done so far in relation to the objectives

Based on these objectives, this document describes the first year of activity on such a Linked Data Ecosystem, mainly referring to Tasks 4.1 “Linked Data server technology” (M1--M36) and T4.2 “Distributed privacy and policy layer” (M1--M24), focussed on the development of a Linked Data Hub for ingesting, connecting, and distributing data through a SPICE network of systems. However, some of the work done already relates to the two remaining tasks starting at M12: Task 4.3 “Linking and discovering digital assets” (M13--M24) and Task 4.4 “Provenance and process analysis”. As such, this deliverable is mainly reporting on Task 4.1, but also include initial work towards the fulfilment of the other objectives, in particular in relation to data acquisition, linking, and analysis (Tasks 4.2, 4.3 and 4.4). We will refer to those objectives in the following outline description.

2 Outline

The deliverable is structured as follows. The next Section is dedicated to **Supporting Citizen Curation** from the socio-technical stand-point, giving the motivation behind the design of a Linked *non-Open* Data ecosystem.

The **State of Affairs** section reports on a two-fold survey. The first, is a recognition on related work on data infrastructures for cultural heritage, with the objective of eliciting existing tools and methods that we could build upon, and making emerge the gaps in relation to Citizen Curation requirements. The second, is an internal survey of content management practices of museums members of the SPICE consortium, with the objective of identifying constraints that are specific to the SPICE project, and leverage those in the design of our solution. With this two-way survey, we aim at getting sufficient insight in order to maximise the relevance of our approach towards the target community but also minimise the impact of our solution in the workflows of museums involved in the project. A key requirement of cultural heritage organisations is related to the protection of copyright-protected digital assets. We will see how this assumption is the corner-stone of our data management approach (our objective 1).

Subsequently, we introduce the result of a collaborative analysis of **Technical Requirements** for an infrastructure supporting citizen curation, at the current state of development (these will be the object of continuous curation during the life-cycle of the project).

The **Architecture Layout** section describes the shape of the technical ecosystem of the SPICE project, and highlights the various types of components involved. These include the key technical developments of WP4: The Linked Data Transformers developed with a new tool named “SPARQL Anything” and the first version of the “SPICE Linked Data Hub”. In addition, the section describes the role of the Linked Data Intelligence layer, where other technical developments of the project play a key role, specifically, the User Model management component (WP3), the community model API (WP3), the Semantic Annotator (WP3), and the Ontology Reasoner (WP6).

In the remaining sections we give a close look at the three main outputs of the work package.

Linked Data Transformations with SPARQL Anything introduces a new, open-source tool for building knowledge graphs from heterogeneous data sources. The approach is based on the notion of *facade*, borrowed from software engineering, and makes it easier to the RDF-aware user to onboard data in a variety of non-RDF formats by using the same query language of the Semantic Web (objectives 1 and 3).

The SPICE Linked Data Hub is designed for supporting the management of data streams coming from diverse sources with fine-grained access control (objective 1 and 2): museum collections metadata and digital assets, social media events and user activities, systems’ activities (e.g., recommendations, reasoning outputs), ontologies and linked data produced by the pilot case studies. Particularly, the infrastructure supports interoperability between different systems, enabling reuse of data and knowledge produced across the pilot studies (objectives 3 and 4).

Finally, we present the initial version of the **SPICEboard**, an end-user application which leverages the plethora of interlinked Citizen Curation outputs to provide data-driven sense-making to museum professionals (objective 4).

3 Supporting Citizen Curation

Traditionally, museums could be thought of as providing an authoritative account of their collection, informing and educating citizens as to the meaning, importance and relevance of their artefacts. The role of the citizen was to appreciate the artefacts and acquire the knowledge and stories associated with them. The role and purpose of museums now tends to be viewed quite differently. The Faro convention on the value of cultural heritage for society (Conseil de l’Europe, 2006) argues for the need to “involve everyone in society in the ongoing process of defining and managing cultural heritage”, that “every person has a right to engage with the cultural heritage of their choice” and that “all cultural heritages [should be treated] equitably and so promote dialogue among cultures and religions”. The current ICOM definition (ICOM, 2020) describes the museum as an institution which “acquires, conserves, researches, communicates and exhibits the tangible and intangible heritage of humanity and its environment for the purposes of education, study and enjoyment”. This definition can be perceived as consistent with a more traditional view of the museum as having a responsibility to communicate an understanding of heritage to the public. A proposed revision to the ICOM definition (ICOM, 2020) describes museums as “democratising, inclusive and polyphonic spaces for critical dialogue about the pasts and the futures”. It goes on to state that museums are “participatory and transparent, and work in active partnership with and for diverse communities to collect, preserve, research, interpret, exhibit, and enhance understandings of the world”. The Faro convention and the revision to the ICOM definition both extend the traditional conceptualisation of the museum in two ways. First, they both highlight that there is not necessarily a single interpretation of heritage. There may be multiple interpretations. Second, they emphasise that the role of the citizen is not confined to acquiring what is presented to them. Citizens can be actively engaged in sharing their voices, participating in dialogue and creating understandings. This trend toward multiple voices and active participation can be seen in recent initiatives to decolonise the museum, challenge the dominant narrative and introduce new perspectives.

Within the SPICE project we are developing tools and methods to support a process we term Citizen Curation. We define Citizen Curation as *citizens applying curatorial methods to archival materials available in heritage and memory institutions as well as to items depicted in exhibitions in order to develop their own interpretations, share their own perspective and appreciate the perspectives of others*. Crucially, our definition of Citizen Curation covers both citizens sharing their own perspectives and also engaging positively with the interpretations of others. The aim is not to just provide multiple interpretations so that the citizen can select the one that fits with their World view, but rather to promote dialogue across perspectives as anticipated by the Faro convention.

Our definition of Citizen Curation has deliberate parallels to the concept of empathy. Zaki (2019) characterises empathy as encompassing a number of ways in which people respond to each other: identifying what the other person feels (cognitive empathy), sharing the emotion of the other person (emotional empathy) and wanting to improve the experiences of the other person (empathic concern). Empathy can often come easily toward people similar to oneself; people are tribal by nature (Bazalgette, 2017). However, empathy can also be cultivated toward perceived out-groups. Two processes that can help to build empathy toward out-groups are perspective giving (sharing one's point of view) and perspective taking (seeing the World from someone else's perspective) (Bazalgette, 2017). Citizen Curation, in incorporating both the sharing and appreciation of perspectives, recognises the role that museums can potentially play in building empathy and cohesion across as well as within communities.

Our definition is informed by previous initiatives that have engaged citizens in the curatorial process. Mauer (2017) and Hill et al (2018) characterise Citizen Curation as a process in which citizens with little or no background in museum curation are provided with training and guidance to create their own physical and virtual exhibitions. Our approach builds on this work but aims to extend Citizen Curation to larger scale participation without additional training. Some previous work has used online tools to widen participation in Citizen Curation. Moqtaderi (2019) uses the term citizen curator to describe members of the public voting for an artwork to be included in an exhibition curated by the museum. The citizen curator initiative developed by Ride (2013) involved citizens sharing contributions via Twitter, later used in a video installation developed by the museum. We take a related approach but emphasise the importance of engaging citizens in perspective taking (appreciating the viewpoints of others) as well as perspective giving. Further discussion of our definition of Citizen Curation and its relationship to other uses of the term in the literature can be found in SPICE deliverable D2.1.

The following guide scenario illustrates a typical citizen curation pipeline that could be implemented with the aid of systems such as digital archives, Web sites, and social media:

Cath is a museum curator. She decides to run an online activity which supports citizens in sharing and reading personal stories inspired by artworks. She selects a set of artworks to be used in the activity. This involves checking that the museum has appropriate permissions to use images of the artworks in the activity and, where necessary, securing appropriate permission from the rights holder. Once the activity has been prepared it is launched on a website developed by a company of the cultural industry sector. Citizens can choose to take part in the activity anonymously, create an account on the system or login via a third-party, mobile application. Citizens can select one of the artworks, tell a personal story related to the artwork and send it to a friend. The friend can send a response to the person who wrote the story. The citizen can also choose to share the story with the curator. Even when a citizen has decided to share their story with the curator, they retain the option to withdraw the story at any time. Cath is able to monitor stories contributed by the citizens. Any story that may contain inappropriate content is automatically flagged and she can choose to remove it. Once the activity had been running for two weeks, Cath creates an online exhibition featuring a selection of the contributions shared with her. In the presentation, she draws attention to the different ways in which artworks have been interpreted. She decides to close the activity to further contributions and relaunch it with a new set of artworks later in the year. Finally, she curates the contributed content and includes it in the museum's digital archive for preservation.

The above scenario hints at some of requirements that the social media layer is required to support. These include:

- The museum professional must be able to set up and launch Citizen Curation activities, such as storytelling, that virtual and physical museum visitors can take part in.
- The museum professional must be able to determine whether appropriate permissions are held for any digital resources provided by the museum for use in Citizen Curation activity.
- The citizen must have appropriate control over how their identity is handled, for example, creating an account, participating anonymously, or connecting via a third-party application.
- The citizen must have appropriate control over how their content is shared, for example, with friends or with the museum.
- The citizen should maintain control over their contributions in the longer-term, rather than it being a one-off decision. For example, the citizen should be able to withdraw content they were previously willing to share.
- The museum professional must be able to easily moderate content shared in the platform, and also curate citizen contributions shared with the museum for access by other visitors.
- The citizen should be supported in identifying alternative perspectives among the shared contributions and sharing these within and across citizen groups.

The following sections review the current state of museum technical infrastructure from the perspective of Citizen Curation. This is followed by a detailed outline of the requirements and current work on the development of an infrastructure to meet these requirements.

4 State of affairs

4.1 Related work on Data Sharing Infrastructures for CH

In this section we survey technologies that currently contribute to data management of cultural heritage institutions and to publishing and exchanging data to support the development of third-party applications. We identified two areas of interest, namely: Data management for cultural heritage; and Web technologies for metadata publishing and exchange. We focus on three types of technologies for data management: (a) end-user tools, (b) services, and (c) components meant to be used within a distributed technical infrastructure. We do not survey related work on approaches to co-design, engagement, metadata management, vocabularies, datasets, or end-user applications that are not targeted to data acquisition and management, that do not have an element of distribution, or that are not maintained anymore. In addition, we postpone the review of areas of interest for Tasks 4.2, 4.3, and 4.4, for example, rights data management, social media and crowdsourcing applications, and distributed online social networks. Those will be covered in D4.2.

While we do not claim that the survey is exhaustive, we expect it to significantly represents current state-of-the-art solutions for data sharing of digital heritage.

4.1.1 Data management for the cultural heritage

In the last years, collection management platforms have become an essential part of the dissemination plans of cultural institutions. Situated midway between Content Management Systems and professional cataloguing tools, these hybrid systems combine the traditional functions of museum software with the need to make collections available online. Solutions range from proprietary high-end products such as the TMS Suite by Gallery Systems¹ to open-source platforms such as Omeka² or Collective Access³.

¹ <https://www.gallerysystems.com/solutions/collections-management/>

² <https://omeka.org/>

³ <https://www.collectiveaccess.org/>

Based on previous surveys and comparisons (Hardesty, 2014 and Wu, 2016), in the following we review Omeka S, DSpace, Fedora (with its spin-offs Islandora and Samvera), ResearchSpace, as well as two data aggregators in the cultural heritage domain: Google Arts and Culture and Europeana.

Omeka S⁴ has established itself as a solution for small to medium projects, due to it being open source and easy to use (Maron and Feinberg, 2018); its modular architecture facilitates the integration with external services such as the ones implementing the International Image Interoperability Framework (IIIF) and indexing services (Solr). While Omeka Classic relies on the DCMES, its linked data version, Omeka S, leaves the user free to create her/his own metadata schema from any RDF vocabulary (Li, 2020), allowing the same set of semantically described items to be shared among different sites on the same installation and across different installations through APIs. From this perspective, Omeka partly satisfies the requirements concerning interoperability and open data. Omeka's modular structure lends itself to data source linking but it lacks predefined policies and protocols to regulate them. Omeka S can support the acquisition of user generated content as part of its workflows, through dedicated modules for commenting and crowdsourcing contributions. Although Omeka S can support the inclusion of user generated content, it does not include a workflow for ingesting this type of content from external applications. Finally, Omeka S can expose collections as linked data through its APIs, but the latter cannot meet the needs of a diverse set of applications such as the ones under development in SPICE

DSpace (Smith et al, 2003) is an open-source platform for collecting, managing, indexing, and distributing digital assets including text, images, videos and data sets. DSpace allows to search and retrieve items, to upload digital items and to decide user roles. It complies with several protocols for access, ingest, and export data (such as: OAI-PMH, WebDAV, RSS, ATOM) and supports common authentication methods, including: LDAP (and hierarchical LDAP), Shibboleth, X.509, IP-based. DSpace covers various requirements for museums, spanning from content management to publishing and curation of copyright information and it has a modular and extensible architecture. However, its installation and management require significant technical skills, so it can be out of reach for small institutions; on the contrary, it scales well to large repositories.

Fedora (Payette and Lagoze, 1998) is an open-source repository for long-term digital preservation and reliable access to any type of digital objects. Fedora is designed to fulfil the requirements of interoperability and extensibility. As far as interoperability is concerned, Fedora supports a number of widely adopted standards. Specifically, it provides a robust RESTful API layer, it allows to serve data as RDF, thus meeting several requirements of our ecosystem. Fedora is designed to integrate with other applications and services to provide additional functionalities (such as dissemination using OAI-PMH, deposit with SWORD etc.). In addition, Fedora can control access to content via a pluggable framework compliant with the SOLID specification. In addition, Fedora implements the memento protocol for enabling the versioning of the resources. Finally, concerning security issues, Fedora is able to integrate with existing authentication systems such as LDAP and Shibboleth. Being primarily a repository server Fedora is often embedded in larger architectures to implement museum solutions, as described below.

Islandora and Samvera

The open source Islandora and Samvera projects share the use of Fedora as repository system, to which they add web publication and user management functions. So, they both inherit the advantages and limitations of Fedora, while their support to the requirements not met by Fedora depend on the specific additional components of each system (and, in the case of Samvera, of the specific configuration of the installed modules). **Islandora**⁵ software framework plugs a well-known CMS, Drupal, into Fedora, thus augmenting it with the content publishing and user management functions provided by Drupal, of which it inherits the limitations in collection and user management. Open to different content types, from scientific to cultural data, Islandora is characterised by a focus on access, through advanced search functions, thanks to the integration with Solr; similarly, to Omeka S, it can be integrated with external servers such as IIIF and OAI-PMH servers. The integration of Fedora with content management functions extends the framework's

⁴ <https://omeka.org/s/>

⁵ <https://islandora.ca/>

capability to comply with citizen curation processes: in particular, it enables the creation of collections on top of the items in the catalogue; thanks to the user management system, it allows tailoring the access and permissions over the items and collections to specific role types. For the end user, the availability of search tools results in a more flexible, personal exploration of the repository. In addition, acquisition of user content can be accomplished through Drupal Form API⁶. In the same spirit, **Samvera**⁷, an evolution of the former Hydra system, is an open-source suite of repository software tools aimed at creating flexible solutions for specific knowledge domains and tasks. Also built on top of Fedora, this framework does not offer canned solutions for citizen curation, but sets of specific modules that can be configured and programmed to satisfy the citizen curation requirements. This openness, although paving the way to a creating a citizen curation system from this framework, is an obstacle to the easy deployment of installations, especially by smaller institutions. The collection management function in Samvera is provided by the Blacklight⁸ search and discovery platform, which meets some end user requirements such as the exploration of content and the curation and publishing of customised collections. Samvera's identity management system supports the restriction of access to the repository items, but without providing a proper DRM management system.

ResearchSpace (Oldman and Tanase, 2018) is an open-source platform for enabling researchers to create, link, share and search data by using Semantic Web languages and technologies. It provides an interesting “assertion and argumentation” model for tracing multiple perspectives on historical facts; enables a multilevel visual representation of resources; allows creating data and narratives in form of knowledge graphs; captures provenance of data; expresses researchers' views as graphs connecting narrative, data, processes, and arguments. ResearchSpace builds on the knowledge graph platform enabling customisation and extensibility of the interaction with the graph database through the use of Semantic Web standards, expressive ontologies for schema modelling based on CIDOC-CRM. The platform also integrates external tools including OntoDia, MIRADOR Image Viewer with an IIIF Image Server. Due to its ability of capturing multiple viewpoints with Semantic Web standards, ResearchSpace addresses several requirements of Citizen Curation. Moreover, it also allows custodians to manage and curate metadata and it enables end users to explore digital assets and to read copyright terms.

A leading platform that contributes to making available digital cultural content to citizens is **Google Arts and Culture (GAC)**. It is especially focused on virtual exhibitions and digital collections, and to date it allows the exploration of the collections of more than 2000 cultural institutions from around the world (Lee et al., 2019). GAC covers various requirements of citizens as end users, including exploration, sharing, and enhancing cultural heritage digital objects. Citizens can explore millions of images of artworks, and compare them in multiple ways, such as confronting them by colour or historical period; create personal virtual galleries by choosing artworks and commenting on them; and share artworks or galleries with others. GAC does not, however, allow users to claim ownership over their personal galleries and interpretations. Additionally, according to (Lee et al, 2019), GAC does not manage copyright issues thoroughly, as GAC blurs images of artworks under copyright restriction in ways that hinder transparency to the user. Additionally, because GAC captures some artworks in extremely high-resolution images, there is a potential to print and misuse them for personal or commercial need (Wahyuningtyas, 2017). This has led to GAC failing to win the trust of some artists to give permission of bringing their art into Google's digital museum, principally because it does not allow copyright owners to know how the digital asset is used –nor museums to manage access control to the digital assets and metadata, and to monitor and analyse the usage of the assets by third parties.

The flagship project of the European Union, **Europeana**, is another leading service for data management of cultural heritage. Officially released in 2009, Europeana's web portal contains over 58 million digitised cultural heritage records of more than 3,600 institutions across Europe (Ioannides and Davies, 2018). It offers a single access point to Europe's digital cultural and scientific heritage aggregated from libraries, archives, museums and audio-visual archives. Europeana offers various services for different stakeholders such as

⁶ <https://www.drupal.org/docs/7/api/form-api>

⁷ <https://samvera.org/>

⁸ <http://projectblacklight.org/>

citizens, cultural heritage professionals and the creative industry, publishing its aggregated content as Linked Data. It thoroughly covers end users' requirement of exploring digital heritage, allowing users to explore the digital archive by a plethora of features, including collection topics, type of media, copyright specifications, providing country, language, institution, colour, even image orientation and size⁹, and as such allowing them to compare and confront digital material from multiple sources. Furthermore, for citizens, Europeana is especially successful at allowing end users to enhance digital collections. For example, in the Europeana Migration campaign (Purday, 2012) over 3000 citizens contributed to a migration thematic collection by sharing their personal migration stories and accompanying pictures, diaries, videos and letters, and Europeana recently announced the 2020 "Gif It Up" competition for most creative reuse of digitized cultural heritage material¹⁰.

4.1.2 Web technologies for metadata publishing and exchange

We survey Web technologies and their application for metadata aggregation in cultural heritage, reusing the survey conducted by (Freire et al, 2017) and complement it with recent advances, namely the Linked.Art project, the W3C ActivityPub, and The Solid Project.

First, we review specifications and protocols and discuss them from the perspective of their use in the reference sector.

The most established technology for publishing and aggregating digital archives is the **Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH)**¹¹. Based on XML technologies and the basic notions of producer and consumer, its adoption has been widespread before the advent of the REST approach for developing Web APIs (Freire et al, 2017). Basically, this protocol allows a consumer to contact providers which expose their catalogue metadata in a common format, whose default is Dublin Core; thanks to this schema, complex federated catalogues can be created, of which the most well-known is Europeana.

Sitemaps enable webmasters to provide a structured layout of the website content to Web engines, providing a list of URLs associated with an extensible set of metadata in XML. This technology builds on a file specification (Sitemap) and a protocol¹². Digital libraries on the Web can use Sitemaps to provide a structured catalogue of digital objects within the library (Freire et al, 2017).

ResourceSync is a standard developed by NISO for synchronizing repositories supporting versioning and notifications of changes (Haslhofer et al, 2013). It is based on Sitemaps protocol, extended with Websub for supporting notifications. However, its application in the domain of cultural heritage is rather limited (Freire et al, 2017), although the approach could be used to publish and consume metadata, particularly supporting timeless synchronisation of remote repositories with live updates.

The Open Publication Distributions System (OPDS) is based originally on the Atom XML syndication format mainly directed to support eBook reading systems, publishers, and providers¹³. The currently developed 2.0 draft adopts JSON-LD and Schema.org and include support for catalogues of images¹⁴. Overall, the specification supports a subset of the requirements already covered by IIIF.

Webmention¹⁵ tackles the problem of receiving notifications when other content is published which refer to a document owned by the receiver. When an agent publishes or updates a document, it can send a notification to the target URL mentioned in the document, to notify the target. This is often a blog post but it can be any type of content. The protocol builds on very basic HTTP techniques, such as the Link header or a HTTP form post. No mechanism is provided to assess trust; therefore, the specification suggests to validate

⁹ <https://www.europeana.eu/en/search>

¹⁰ <https://pro.europeana.eu/page/gif-it-up>

¹¹ <https://www.openarchives.org/pmh/>

¹² <https://www.sitemaps.org/protocol.html>

¹³ The OpenPub Community. OPDS Catalog 1.1 specification (2011): \url{https://specs.opds.io/

¹⁴ <https://drafts.opds.io/opds-2.0>

¹⁵ <https://www.w3.org/TR/2017/REC-webmention-20170112/>

the mention by downloading the source. Notifications do not include data transmission, so the technology needs to be complemented by other means for performing content update.

WebSub provides an option for developing connections across distributed systems based on HTTP web hooks¹⁶. The protocol assumes *hubs* managing the discovery and exchange of content, without requiring any commitment on the format of the document. Content publishers and subscribers interact by the means of *topics*. Agents can discover the hub (or more hubs) managing content of a topic URL, and perform a HTTP POST to subscribe. Publishers of the *topic* notify the hub of changes, which in turn broadcast the message to all subscribers. WebSub covers requirements related to monitoring and tracing content changes, with particular focus on feeds such as Atom or RSS.

The **ActivityPub** protocol¹⁷ is a specification for supporting decentralised social networking on the Web, building on top of the ActivityStreams 2.0 data format¹⁸. ActivityPub provides two APIs, one for client to server communication (for creating, updating and deleting content), and the other for server-to-server communication, to tackle content distribution. Actors have one inbox and one outbox, and an associated collection of followers, to which servers broadcast new messages. ActivityPub is a promising technology to enable communication of social applications on the Web.

Next, we review initiatives and projects aiming at supporting development infrastructures.

In the context of cultural heritage, a key initiative towards a truly distributed infrastructure is the **International Image Interoperability Framework (IIIF)** focused on supporting the publishing and exchange of high-quality images and media on the Web, supporting authentication, access, and presentation of digital images and media through dedicated Web APIs. Requirements covered by IIIF include the publishing of digital objects and associated metadata in a standard way, and the ability to feed live end-user applications from a distributed set of repositories. Also, the standard explicitly supports end-user applications developing browsers and explorators of digital content, an important requirement of citizen curation applications.

Schema.org (Guha et al, 2016) is the only technology mentioned by (Freire et al, 2017) that deals with modelling aspects of metadata aggregation for cultural heritage. In our analysis, its relevance relates to the fact that it represents a set of techniques used for publishing structured metadata. Data is incorporated into HTML pages using various techniques (RDFa, Microformats). In our analysis, we leave out considerations related to knowledge representation but focus on its capacity to develop consensus from the bottom up within communities of practitioners of the Web of data. Cultural heritage institutions can use Schema.org to publish structured metadata on Web pages, which in turn can be scraped by agents to extract structured information.

The Linked.Art project follows the steps of IIIF with the aim of providing a set of APIs for the exchange of detailed metadata information in cultural heritage, focusing primarily on the art domain¹⁹. Building on top of Web APIs and the Linked Data format JSON-LD, its vision is to make Linked Data *usable* by relieving the developers from dealing with RDF and heterogenous vocabularies. Linked.Art can be considered as the development of an ad-hoc API for the exchange of catalogue metadata. Museums can use Linked.Art for publishing content on the Web of Data in a standardized way.

The Solid Project builds on the RDF technology stack and the REST approach to Web APIs to provide a set of techniques for accessing and manipulating resources as Linked Data. The Solid family of specifications, which are built on top of LDP, includes techniques for distributed identity management (WebID), specification of access control (WebACL), and others. The Solid project aims to develop an ecosystem of specifications and technologies for building decentralised applications, with particular accent on improving privacy and data ownership on the Web. Solid takes the challenge of developing a fully distributed infrastructure on top of the

¹⁶ <https://www.w3.org/TR/2018/REC-websub-20180123/>

¹⁷ <https://www.w3.org/TR/2018/REC-activitypub-20180123/>

¹⁸ <http://www.w3.org/TR/2017/REC-activitystreams-core-20170523/>

¹⁹ <https://linked.art/>

Web of Data. The goals of the project are ambitious and so far, the core specifications cover resource management (Linked Data Platform²⁰), content update notification (Linked Data Notifications²¹), identity, and distributed access control (Web Access Control²²). This technology stack is very promising for citizen curation as it satisfies some key requirements related to copyright, access, distributed identity, and extendibility. However, the Solid Project is under development²³ and, therefore, specifications have different levels of maturity.

4.2 Survey of content management practices of museums in the SPICE consortium

In addition to reviewing the literature on data management and Web technologies for cultural heritage, we also conducted a survey, internal to the SPICE consortium, about the actual content management practices of the partner museums.

One of the curator competences is to select artifacts for exhibitions. In order to survey the curators' activities in the selection and preparation of the artworks, we conducted two types of analysis: the first one concerned the software used by museums to manage the collections; the second one was aimed at learning more about the context of use in the partner museums. We found 21 different software systems, of which only 3 are open source and available from public repositories such as GitHub (Omeka, Collective Access and Xdams). Most of the software systems for collection management are also cataloguing software, and only 3 are also Real-time exhibition design tools (CatalogIT, Ortelia curator, Sebina). Museums use different curatorial software platforms: some are more oriented to cataloguing; some are more oriented to the creation of collections; some platforms integrate tools for designing the physical or virtual spaces of temporary collections. Table 4.1 contains the list of the systems, with the main functionality (column 3) and the additional functions (Collection Management, Cataloguing, Real-Time exhibition management); the last two columns refer to the license type and the availability on GitHub).

Table 4.1: List of software systems used by the museums

²⁰ <https://www.w3.org/TR/2015/REC-ldp-20150226/>

²¹ <https://www.w3.org/TR/2017/REC-ldn-20170502/>

²² <https://github.com/solid/web-access-control-spec>

²³ The development is open and community-oriented: <https://github.com/solid>

Name	Functionalities	Collection management	Cataloguing	Real-time exhibition design	Open Source	GitHub
Gallery system- TMS	Collection management	x	x			
Axiell	Collection management	x	x			
Second Canvas	Collection management	x	x			
Artwork archive	Collection management	x	x			
Art Gallery	Collection management	x	x			
ArtSystem	Collection management	x	x			
Artlogic	Collection management	x	x			
ComWork	Collection management	x	x			
CatologIT	Collection management	x	x			
Ortelia curator	Real-time exhibition design software	x	x	x		
Cura3D	Real-time exhibition 3D design software	x	x	x		
Museum space	Collection management	x	x			
Past perfect	Collection management	x	x			
Artbase	Collection management	x	x			
Memoron	Collection management	x	x			
Actimuseo	Collection management	x	x			
Xdams	Collection management	x	x		x	x
Collective access	Collection management	x	x		x	x
Comwork	Collection management	x	x			
Sebina	Collection management	x	x	x		
Omeka	Collection management	x	x		x	x
Memora	Collection management	x	x			
Archiui	Collection management	x	x			

Concerning the second issue (use of museum software by partner museums), we conducted a survey on the use of collection management software by the partner museums. The aim of this survey was to gain a better understanding of the internal dynamics of software use and the role of curators in the exhibition design, creation and management. To do so, we distributed a questionnaire which contained the following questions:

- Do you use or have you used software programs for temporary exhibition design and management?
- Which collection management program or tool do you use?
- Who of the staff is involved in the exhibition design besides the curator? (educators, the director, architects, designers, etc. ...)
- Do you have an accessibility manager?

Application Name	Type of application	Functionalities	Institution	Used by
AutoCAD	Computer-aided design (CAD) and drafting software application	exhibition design	IMMA	external exhibition designers
AutoCAD	Computer-aided design (CAD) and drafting software application	exhibition design	GAM	external exhibition designers
AutoCAD	Computer-aided design (CAD) and drafting software application	exhibition design	HECHT	external exhibition designers
AutoCAD	Computer-aided design (CAD) and drafting software application	exhibition design	DMH	external exhibition designers
SketchUp	3D modeling computer program	exhibition design	IMMA	external exhibition designers
Slack	Communication platform	workflow/ management	IMMA	Internal and external staff
Memoron	Collection management	organize, management the collections, catalogue	DMH	Internal staff
Museumplus	Collection management	organize, management the collections,	IMMA, DMH	Internal staff
GAM Multimedia	Collection management	organize, management the collections	GAM	Internal staff
MANA	Collection management	organize, management the collections	HECHT	Internal staff
Excell		Budget	HECHT	Internal staff

Figure 4.2 Summary of the survey

The survey showed that the situation is different from museum to museum, both at the software level and at the level of staff organization. Regarding the exhibitions, museums don't use any particular digital support, and the design of exhibitions is often entrusted to external professionals as architects. DMH and IMMA use the same collection management software, Museumplus, one of the most used; GAM relies on a special program specifically developed for the museum and called GAM Multimedia; HECHT uses MANA, a software developed by the Israeli Ministry of Cultural Heritage. All museums lack a disability manager. In museums, the Equality and Diversity Manager should “advocate for access, diversity and equality across the Museum, helping to ensure that the Museum is complying with the latest legislation for visitors and employees. Provide advice and direction to exhibition planning in relation to intellectual, sensory and physical access, and will ensure that access, diversity and equality issues are fully considered in all major projects”.²⁴

The data that emerged from the survey confirm the heterogeneity of museum staff and the importance of different skills for the realization of an exhibition, sometimes outsourced –and so out of the direct control of the museum professional roles. For SPICE, this aspect is very important, because it allows us to better define the context where citizen curation activities will take place. In particular, we envisage as a primary prerequisite of the project to underline the importance of the role of museum educator or mediator, who relates directly to the public and acts as a cultural mediator to make the heritage accessible and understandable to all. Also, it would be desirable to have Disability Managers in partner museums, since project pilots refer to groups at risk of exclusion. A visualization of the staff chart of partner museums can be found in Figure 4.3.

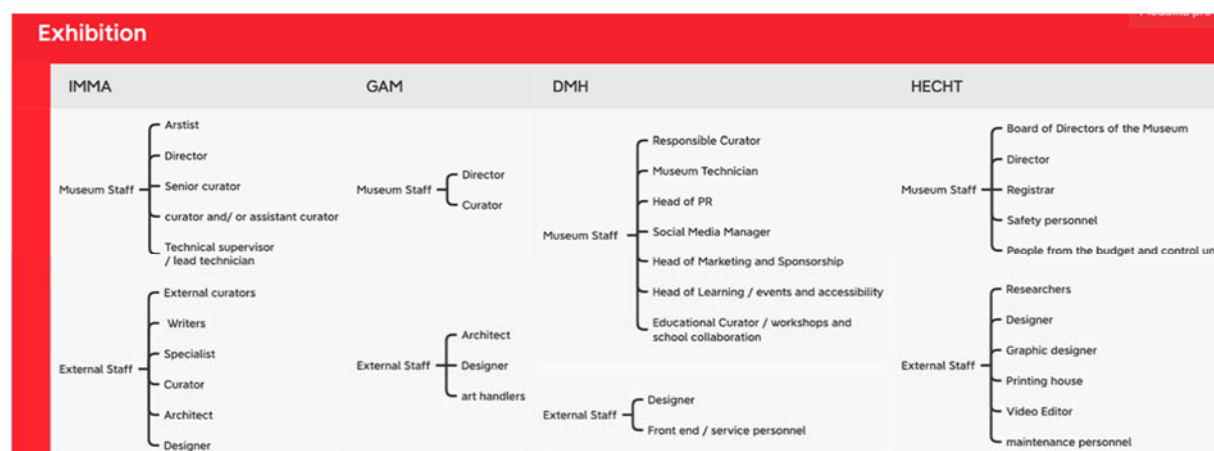


Figure 4.3: Analysis of roles in cultural heritage organisations

Digital engagement and Covid-19. In addition to the surveys illustrated above, the analysis of museum software and technologies cannot abstract from the changes brought by the Covid-19. The pandemic, in fact, propelled the use of digital technologies by museums, but at the same time it revealed some underlying gaps in their practices.

2020 was an important turning point for digital technologies and museums. In particular, online communication was, for most of the year, the only opportunity to involve the public. According the We are

²⁴ <https://equalentry.com/qa-with-ruth-starr-accessibility-manager-cooper-hewitt/>

social report, there hasn't been a significant difference between 2019 and 2020 in terms of the number of internet users; however, there has been a relevant rise in terms of smartphone and mobile internet connections and average time spent on the internet by single users. At the same time, mobile phones and smartphones, as well as game consoles and other devices, were preferred to laptops, desktop computers and tablets to access the internet^{25,26}. The online presence of museums may take several forms, ranging from traditional online catalogues and digital exhibitions that include narratives and audio-visual content to virtual representations of the museum's physical environment (Vayanou et al.2020)

To investigate this point, several studies have been carried out by important organizations such as ICOM, NEMO and UNESCO:

- NEMO “**Survey on the impact of the COVID-19 situation on museums in Europe**”²⁷ and “**Survey on the impact of the COVID-19 situation on museums in Europe-follow up**”²⁸ to which about 600 institutions replied, report that 70% of the museums participating in the survey have increased or created new digital services to reach their audience; this figure in the second follow-up saw a strong increase, registering 93%. More than 75% of the institutions have focused on increasing their social media activities, with the creation of many video contents, recording an excellent response from the public 60%, among the most popular initiatives are videos (42%) and virtual tours (28%). However, nearly 40% of museums responded that they have not tracked or are unaware of the development in online visitor numbers, indicating a lack of digital measurement frameworks and methods.
- **UNESCO Museums Around the World in the Face of COVID-19**²⁹ highlights an increase in the digital activities of museums and states that museum social activities have increased by 15%. Furthermore, many institutions have created, starting from the already digitized collections, various offers: online collections, 360° virtual tours, virtual museums, online publications, digital exhibitions.
- **ICOM International Museums, museum professionals and COVID-19**³⁰ is based on 1,600 responses from museums and museum professionals from 107 countries collected between 7 April and 7 May 2020. The analysis touched five aspects: the closure and situation of the staff, the estimated economic impact, digital activities and communication, safety and conservation, an in-depth study of professionals and museum consultants. With regard to aspects related to digital, 47.49% said they had increased communication activity on social media and 17.97% the online collection. The follow-up survey³¹ carried out between September-October 2020 confirms some data that emerged in the previous one: in general, digital activities have increased or started after the first periods of closure for at least 15% of the participants. In particular, social media activity increased for 50% of respondents and the percentage of museums that opened a new channel also increased compared to April.

In all surveys, the increase in the digital offer of museums is confirmed, mainly aimed at an intensification of activities on social media and the creation of new content such as virtual tours, videos, podcasts. However, this digital production is not often accompanied by the presence of a full-fledged digital strategy. The increase in the use of social media activities is highlighted also by the Art Newspaper annual survey³². According to (Agostino et al 2020), in Italian museums “social media platforms, especially Facebook, Twitter and Instagram, have become the museums’ preferred means to spread culture during the COVID-19 lockdown”. Along with the increase in the online presence of museums reported by the surveys mentioned above, data

²⁵ Report We are social, 2020, <https://wearesocial.com/it/digital-2020/italia>

²⁶ <https://wearesocial.com/>

²⁷ <https://www.nemo.org/news/article/nemo/nemo-report-on-the-impact-of-covid-19-on-museums-in-europe.html>

²⁸ https://www.nemo.org/fileadmin/Dateien/public/NEMO_documents/NEMO_COVID19_FollowUpReport_11.1.2021.pdf

²⁹ <https://unesdoc.unesco.org/ark:/48223/pf0000373530>

³⁰ <https://icom.museum/wp-content/uploads/2020/05/Report-Museums-and-COVID-19.pdf>

³¹ https://icom.museum/wp-content/uploads/2020/11/FINAL-EN_Follow-up-survey.pdf

³² <https://www.theartnewspaper.com/analysis/visitor-figures-2020-top-100-art-museums>

sharing practices are tightly coupled with data management infrastructures, which are typically isolated from consumption/engagement media (for example, social media).

To summarize, the surveys on the initiatives run by museums worldwide during the Covid-19 emergency confirm the prevailing lack of a digital strategy by museums, which can be partly attributed to the disconnectedness of workstreams of social presence and collection management. The publication of items and collections on social media and the management of collections are dealt with by using distinct tools which are not interoperable, thus obtruding the ingestion of the responses collected from social media into collection management platforms. At the same time, the museum professionals who design and implement the activities to engage the users in different spaces than the traditional ones, including virtual spaces, cannot rely on collection management software. In most cases, since collections are not published in the linked data cloud, the virtual spaces created by museums to cultivate social presence and foster citizen engagement overlap with the boundaries between the single institution, instead of creating new larger spaces.

4.3 Analysis

The needs of collection managers and citizen engagement professionals seem to be somehow orthogonal, which is not entirely surprising considering that the latter essentially act as mediators between users and custodians. However, they seem to receive limited support, resulting in a cultural industry which does not have a structured, resilient approach to data sharing and reuse –the foundational infrastructure for citizen curation. In addition, citizen engagement professionals do not have means for discovering protected content, for example, to request access and negotiating terms of usage. Instead, existing tools are occasional in nature and typically result in isolated, stand-alone applications. Another neglected area relates to the monitoring of content in digital archives of custodians. Although we did not explore how content monitoring is performed on mass social media –for which we refer the reader to (Batrınca and Treleaven, 2015), it is evident how cultural heritage archives do not have means for dealing with the heterogeneity of issues related to user generated content. However, Web technologies already offer basic components to allow external agents to react to the production and update of content in third-party repositories (for example, ActivityPub). Intelligent agents could monitor user generated content and raise alerts when potential issues are identified, for example, with relation to privacy or copyright violations. These shortcomings are also reflected in relation to the socio-technical ecosystem. Although persistent identifiers are naturally supported by any system based on Linked Data principles, issues such as distribution of identity have limited support, and data sharing platforms and aggregators are essentially fully centralised systems. In particular, data management systems may have the capacity of delegating the authentication to an external system, for example, with OAuth 2.0 (Jones and Hardt, 2012), but does not prevent the disclosure of the identity of the end-user (the mail address).

However, citizen curation expects end-users to interact with system provided by third-parties, which act as *mediators* between the user and the organisation (the cultural heritage institution). These mediators will need to handle user identity and deal with commitments related to privacy law (GDPR), avoiding to propagate personal identifiable information to remote parties. Although the identity may be unknown to the organisation collecting and curating the content, users may still want to be able to have control over the final content. The problem cannot be addressed until identity management and control over content shared and reused will get attention by the research community on data infrastructures for cultural heritage.

However, a crucial question is: what types of connections need to be established?

As discussed in the previous sections, both data and identities need to be available on the Web but access must be controlled by content owners or delegated to a trusted entity. Identities should be linked to digital assets through metadata annotations, and these metadata need also to be accessed with appropriate restrictions. These prerequisites make the context fundamentally different from the one of (Linked) Open Data, where personal identifiable information can be removed through anonymisation techniques and one

can expect to expose the same content to any user without distinctions. However, it is an open question how an infrastructure for supporting **Linked Non-Open Data** would look like.

A specific issue for the creation of citizen curation platforms from existing systems and frameworks concerns the lack of dedicated workflows for the type of user-generated content produced by citizen curation processes. Thanks to their architectural modularity and the extendibility of their representation schemas, most systems, especially in the data management area, have in principle the capability to represent user-generated content, but they don't acknowledge this type of content as part of their workflows. In the current situation, any attempt to use these systems to include user generated content would fall short of creating the appropriate paths for handling them according to the requirements. Current workflows, in fact, mostly rely on a unidirectional path from ingestion to fruition, with user responses not being reintroduced in the system as first-class citizens. The flexibility of access tools provided by most systems, which rely on effective indexing modules, is generally intended for the end user, and it is not available to create curation paths depending on content types and features (in other terms, they are not ready to enable the *scripting* of activities for generating and managing user-generated contents). Also, extending the current solutions to create citizen curation systems may not be feasible for small organisations, that cannot afford a similar effort (in terms of know-how, costs, staff, infrastructures) needed to make significant development work on top of their current solutions. We can conclude from the analysis that the role of mediators will be crucial in supporting the heterogeneity and diversity of organisations involved in citizen curation projects, providing specific services (e.g., licence clearance or monitoring inappropriate content) and flexible, cost-effective platforms to design and curate interactions, processes, and data.

Our discussion so far focused on the limitations of current approaches with relation to the challenges raised by citizen curation, while we would like to conclude this section by tackling the question: what approach should we take in order to fill the gaps?

Some of those challenges above described were inherently part of Semantic Web research in the last twenty years. Cultural heritage is a field where semantic technologies have already been introduced and employed in the past two decades (in particular for what concerns the development of models for the representation of cultural items that, however, are not in the scope of this report). Semantic web technologies, in particular, could support many of the requirements reviewed in this article. For what concerns the semantic data management, however, it emerges that little advancements have been made from the agenda settled in 2012 (Vavliakis et al, 2012). Specifically, still insufficient efforts have been made for what concerns the availability of integrated knowledge systems, the validation of the extended data models provided by each cultural institution, the capability of handling uncertain reasoning, the multilingualism of the exposed cultural repositories etc. In general, a crucial element to improve would concern the integration of the existing technologies with the standard workflow operation of Cultural Experts and Information Professionals (IPs) in Libraries, Archives and Museums (LAMs) (McKenna et al, 2018). Finally, and consistently with the requirements provided by the citizens curation ecosystem, the crucial issues concerning the ownership, permission of use, trust, and copyrights issues have only been recently sketched.

The mentioned Solid Project represents the front-runner initiative in the field to answer to such requirements. However, the project is very ambitious and the technologies involved have different levels of maturity. Specifically, the community is still far from tackling important issues such as the one of information discovery and diffusion –issues which a technological infrastructure aiming at supporting citizen curation should necessarily be able to cope with. However, we have seen how research on Distributed Online Social Network provided solution to securely broadcast protected content. The Mastodon project is based on Web technologies (ActivityPub) and at the same time offers a social media platform based on content distribution. However, the type of content and interactions supported seems limited.

In short, we can devise two main directions for research currently pursued by the SPICE project:

- Identify new ways to support museums in sharing collection metadata and digital assets, preserving their control on the content;

- Support businesses in developing innovative applications for citizen engagement, relying on a distributed, resilient infrastructure, federating content from multiple sources;

The SPICE project is developing new methods for publishing linked data from collection metadata and digital assets, into a Linked non-Open Data network of resources. The main novelty of the SPICE approach is to go beyond aggregators and research on an infrastructure that could mediate between cultural heritage institutions, citizen engagement organisations, and citizens, keeping ownership and control in the hands of the data providers.

5 Requirements

In this section, we present the results of the requirements' analysis performed in WP4. These were collected during the first year of the project in a collaborative fashion by researchers and museum professionals of the SPICE consortium, participating in the activities of all technical work packages, and follow an analysis of the scenarios and pilot case studies developed in WP7. It is important to highlight how these requirements are not intended to be complete and that they will be iteratively verified, corrected, and enhanced during the course of the project. From our perspective, they constitute a *living resource*, which fuels the research and development work of the SPICE data infrastructure. In what follows, we introduce the current state of our requirement's analysis and collection, giving prominence to the requirements covered by the current version of the Linked Data layer, the main subject of this deliverable.

Table 5.2, provides an overview of requirements collected during the first year of the project and related to the development of the *Linked Data layer*. We organised the requirements around four main roles:

Custodians: this role represents the Museums' professionals and Cultural Heritage institutions. Custodians have the duty to preserve and valorise cultural heritage. Crucially, they may or may not be the proprietary of the digital assets. Certainly, they have control on the content and metadata which needs to be used in Citizen curation.

Builders: this role refers to users of companies involved in reusing cultural heritage to develop innovative citizen engagement technologies. Typically, builders realize front-facing applications reusing content derived by custodians. More importantly, interaction with end-users create new content, which needs to be managed, and is of interest to the custodians.

Owners: this role refers to individuals or organisations holding the ownership of cultural heritage content. This may be artists or copyright management organisations, but also social media users producing content reused by the museum.

Data managers: this role refers to entities which are in charge of technical support activities on behalf of individual and organisations. For example, the museum may delegate a company in developing the required infrastructure for connecting the collection database to third-party systems.

Of course, these roles are not meant to be disjoint, and there are many museums which also own their assets, and develop their own front-facing applications. Notice also that these functional roles must not be confused with the actual roles found in museum organization charts, as surveyed in Section 4.2, which pertain to a wider set of objectives, ranging (but not limited to) from the preservation, study and dissemination of cultural assets to the universal accessibility of collections and spaces. However, there is a value in making this distinction, which lies in characterising the gap that current infrastructures for cultural heritage have, as illustrated in Section 4. In addition, it should be noted how we are not considering requirements for target users of the systems at this stage, which are specifically covered in Deliverable D5.1. Particularly, here we focus on roles which impact on data management activities. From this perspectives, social media users are considered *owners* of generated content.

The complete list of requirements is available in Table 5.1. The Table provides a description of the requirement and associated roles/actions, including a development status, in relation to this deliverable. In the following sections, we will present the outputs of the deliverable which result from the needs identified.

Discussion with connections to the rest of the deliverable

Table 5.1. List of requirements under development in WP4

	Nickname	Role	Action	Target	Status
1	[AnalyseUsage]	custodian/owner	analyse	access and usage of my data	50%
2	[BackupContent]	data manager	backup/restore	my data to support recovery in the case of a loss event.	30%
3	[BrowseIndex]	builder	browse	an index of the resources I have access to	30%
4	[BrowseMarketplace]	custodian/owner/builder	browse	a marketplace of offers of digital assets	0%
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	0%
7	[ExpressCopyright]	custodian/owner	express	the copyright associated with digital assets in my collection	0%
8	[ExpressExemptions]	custodian/owner	express	exemptions and characterize them	0%
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.	50%
11	[ExpressPermissions]	owner	express	permissions, prohibitions, constraints and duties	0%
12	[ExpressPolicies]	custodian/owner	express	usage policies in relation to my data	0%
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	0%
15	[ExternalAccessData]	builder	access	data from an external application	100%
16	[FilterSensitiveContent]	custodian/builder	filter	sensitive content for specific target groups	0%
17	[GrantCheck]	builder/custodian/owner	verify	lawful access to a collection metadata or digital asset	30%
18	[GrantRecovery]	owner/custodian/builder	view	terms of use granted	0%
19	[InappropriateContent]	custodian/builder	identify/filter	user-generated content that can be inappropriate	0%
20	[InspectIngestionProcess]	owner/custodian	inspect	the metadata production in the ingestion process	30%
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	30%
24	[MultipleRightsAspects]	custodian	express	that multiple subjects hold copyrights on different aspects of the digital asset	0%
25	[NegotiateRights]	custodian	negotiate	rights on behalf of the owner	0%
26	[NominateDelegate]	custodian	nominate	an external entity to negotiate rights on behalf of a copyright owner	0%

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	10%
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)	0%
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%
33	[RegisterSources]	data manager	register	existing Linked Data sources	0%
34	[RequestAccess]	builder	request	access to data	0%
35	[RequestPermission]	builder	request	permission to use a digital asset under specific terms	0%
36	[RevokeRights]	owner/custodian	revoke	usage permissions I granted in the past	30%
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	0%
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	0%
42	[WriteData]	data manager/builder	write	data to a dataset –e.g., via a (secured) Web API	100%

6 Architecture layout

In order to better understand the role of the various components of the SPICE ecosystem, we designed an architecture layout. This design methodology allowed us, on the one hand, to orchestrate the competences of the various work packages from the technical standpoint, and on the other hand, to analyse the connections of the various components and sketch the basis of the technical cooperation. Figure 6.1 provides an illustration of the Architecture layout at the current stage of the project (v1.0). As it can be seen, the architecture is composed of a number of layers, whose foundation is on the transport layer, which is essentially the Web protocol HTTP. The central area of the diagram refers to what we call the Linked Data Layer (LDL), which essentially combines a number of Web APIs, relying on W3C standards (SPARQL endpoints) or designing new ones relying on agreed metadata schemas and operations in order to manage and exchange data. By supporting a shared system for authentication and authorization, we can orchestrate three types of components: a Linked Data Hub, which comprises a registry and data aggregator, a stack of transformers, dedicated to ingesting legacy data and publish them as Linked Data within the secured infrastructure, and a set of intelligent agents (LDI) which inspect the content within the LDL and generate annotations automatically. In the next sections we will explore in detail the key technical developments of WP4: The Linked Data Transformers (LDT) developed with a new tool named “SPARQL Anything” (Section 7) and the first version of the “SPICE Linked Data Hub” (Section 8).

The architecture layout is flexible and we expect new services to emerge during the project, having the roles of either managing content (LD transformers) or reasoning over produced content (LD intelligence). The higher levels pertain the development of applications within the SPICE projects, particularly in the context of the pilot studies (WP7). We expect systems to support four types of users: citizens, builder, custodians, and owners. These roles will be object of further analysis in the next period as their needs provide important insights on the nature of the technical infrastructure required to support them (this has been partly analysed in the requirements section). For the time being, we started to look into monitoring and analysing user-generated content by museum officers and develop a dashboard for citizen curation and social media activities as part of the deliverable (Section 9).

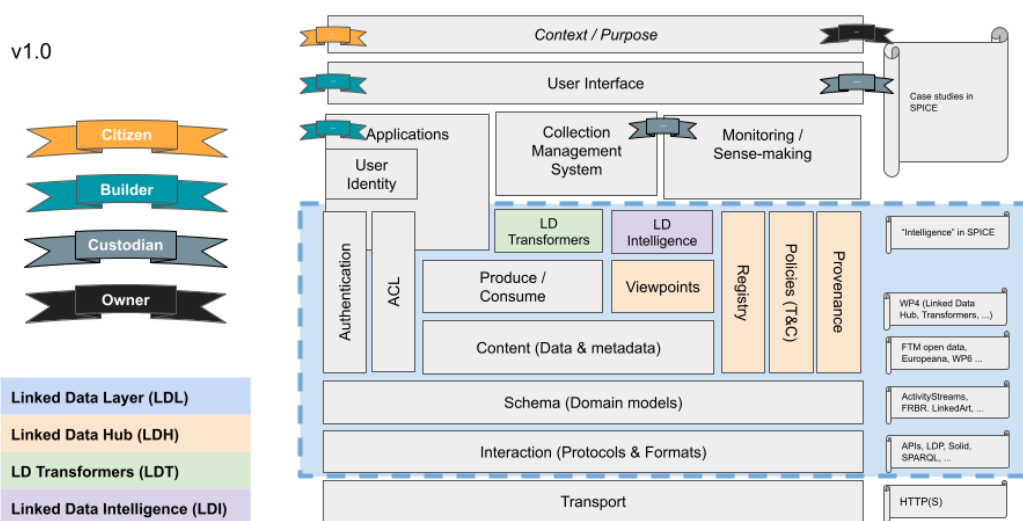


Figure 6.1: Architecture layout, v1.0

In what follows, we describe the role of the other technical developments of the project which play a key role in the ecosystem, specifically, the User Model Management component (WP3), the Community Model API (WP3), the Semantic Annotator (WP3), and the Ontology Reasoner (WP6). These sub-systems are supposed to interact through the stack of APIs provided by the Linked Data Hub (LDH).

The **User Model management** component is a data structure that contains information about individual users. It contains personal information that is explicitly provided by either the user or a system administrator and it contains user preferences in the form of attribute- value pairs, as inferred from user generated content and user interactions with the system. The user model is stored in the LDH and used by the User Modeler (for updating its content) and by the recommender system (for reasoning on the content to be presented next to the user). A detailed description of the service can be found in the Deliverable document D3.1 ("Prototype User and Community Model ").

The **Semantic Annotator** is an annotation service for the semantic enrichment of textual contents, targeting user generated contents as well as descriptions of museum artifacts. The service is multilingual and supports English, Finnish, Hebrew, Italian and Spanish. It consists of a natural language processing pipeline that performs sentiment analysis, emotion detection and entities linking; the service annotates the textual contents with respect to the ontological model developed in WP6 and stores the generated RDF graph in the linked data hub developed by WP4. The service is exposed through standard RESTful APIs and the output is provided as a JSON-LD document.

The **User Modeler** is a reasoning component that monitors the user's actions and the user's generated content (by means of content/emotions analysis – see D3.2) for modelling the user's interests in the current objects/concepts discussed, to be used later on by the community modeller and the recommender. The information is stored in the user model and whenever new information about the user becomes available (at the initiation of the user model – explicitly from a system administrator/user input and during regular operation from reasoning on user generated content) the user modeller updates the user model as needed.

The **Community Model (CM)** is responsible for the detection of implicit communities using user data and activities and the computation of similarities among these communities. The CM component uses the SPICE Linked Data Hub to store information about the community model and to obtain information from the part of the content model that can be derived from the user activities, specifically, to compute the similarity metrics needed in the community detection module and to generate and enrich the explanations generated by the explanation module. The system allows to produce communities which are persistent or virtual. Persistent communities are those that are stable in time and interesting for a certain use case (museum) (like the personas identified in the use cases) and virtual communities are not persistent, as they have a temporary and dynamic character and arise with new users, new opinions, stories and/or reflections. They can become persistent if required. The **Community Model API v1.0 (CM API)**³³ allows to query information about the community model (CM). A detailed description of the service can be found in the Deliverable document D3.1 ("Prototype User and Community Model ").

The **Ontology Reasoning** component is one of the elements composing the Intelligence block in the SPICE Linked Data Hub. It is built as a collection of API services to use for the enrichment of both museum data and user generated data with the high-level semantic categories of the network of ontologies developed in WP6 (task 6.3). This reasoning layer serves three main functions and will be provided with different modules (each of them usable via APIs by other services). The first function (already in place and based on the architecture described in the Deliverable 6.1) concerns the possibility of grouping and querying with SPARQL museum items by leveraging the narrative and emotional descriptions associated to them via standard Description Logic reasoners (e.g., Hermit). For example: the narrative ontology developed in task 6.3 can enable the possibility of grouping cultural objects according to the "events" or the "actions" they share (e.g., "killing", "kissing" ...) by showing unexplored and hidden connections among museum items and, therefore, fostering the interpretation and reflections loops activities designed in WP2. The second one, under development,

³³ Available at <https://app.swaggerhub.com/apis-docs/gjimenezUCM/SPICE-CommunityModelAPI/v1>

concerns of a thematic reasoner able to extract, from museum data about a given collection, the main Thematic Subjects to which such collection refers to. Such thematic extraction is done by founding the common Wikipedia ancestors (if any) among the objects of a collection. Finally, the third function, under development, is based on a novel reasoner called DEGARI relying on a process of dynamic knowledge-based expansion obtained via the non-monotonic description logics TCL. The process of knowledge expansion can be used for the creation of novel classes in the ontologies that can better fit the museum data and that can be used for reclassifying in a novel fashion the museum objects. Such reclassifications could represent an element taken into account by the recommender system. The preliminary version of this reasoner is available at <http://www.di.unito.it/~lieto/DEGARI/>.

Finally, the **Recommender** is a reasoning component that considers the user model, the user's current activity (provided by the User Modeler), the group model (provided by the Community Model) and using this data about the user and users' communities' modeller, following the activity script, selects the most appropriate content to be presented to the user. A detailed description of the service will be found in the Deliverable document D3.6 ("Community Recommender").

7 Linked Data Transformations with SPARQL Anything

As we have seen in Section 5, the majority of cultural heritage data is not published as Linked Data. Crucially, the totality of cultural heritage institutions involved in the SPICE project do not have a linked data publishing infrastructure in-house. Therefore, the LD Transformer class of components is a crucial element of the Linked Data Layer. However, this issue is not new to cultural heritage. **Knowledge graphs** have nowadays a key role in domains such as enterprise data integration, where domain applications typically deal with heterogeneous data objects. In those contexts, Semantic Web experts develop knowledge graph construction pipelines that include the transformation of different types of content into RDF. This is achieved by using refactoring tools that act as mediators between the data sources and the needed format and data model (Haslhofer et al, 2011). Alternatively, dedicated software components implement ad-hoc transformations from custom formats to a multiplicity of ontologies relevant to the domain (Daga et al, 2016).

The SPICE project aims at developing a linked data infrastructure for integrating and leveraging museum collections using multiple ontologies covering sophisticated aspects of citizen engagement initiatives. Museum collections come in a variety of data objects, spanning from public websites to open data sets. These include metadata summaries as CSVs, record details as JSON files, and binary objects (e.g., artwork images), among others. The motivation for researching on novel ways to transform non-RDF resources into RDF comes from the scenarios under development. In particular, pilot case studies are relying on a linked data network of resources from museums, social media, and businesses active in the cultural industry. However, the majority of resources involved are not exposed as Linked Data but are released, for example, as CSV, XML, JSON files, or combinations of these formats. It is clear how the effort required for transforming resources could constitute a significant cost to the project. In addition, the research activity aims at the design of task-oriented ontologies. This implies that there will be multiple semantic **viewpoints** on the resources and their metadata. In the absence of a strategy to cope with this diversity of resources and ontologies, content transformation may result in duplication of effort and become a serious bottleneck.

The semantic lifting of such a variety of resources can be a serious bottleneck for the project activities. Several languages have been developed to either engineer content transformation (e.g., RML) or extending the SPARQL query language to access non-RDF resources (e.g., SPARQL Generate). A long history of mapping languages for transforming heterogeneous files into RDF can be considered superseded by RML (Dimou et al, 2014), also specialised to support data cleaning operations (Slepicka et al, 2015) and specific forms of data, for example, relational (Rodriguez-Muro et al, 2015) or geospatial data (Kyzirakos et al, 2014). RML is also representative for general data integration approaches such as OBDA (Xiao et al, 2018). This family of solutions are based on a set of declarative rules that Semantic Web practitioners are expected to develop by analysing the input data sources. Overall, existing solutions are based on the idea that developers are willing to learn and use a dedicated mapping language and that they are happy to encode the remodelling process accordingly for each new resource. Indeed, existing solutions require Semantic Web practitioners to use an

ad-hoc language, or even combine multiple languages, for example requiring to use XPath for XML transformations and JSONPath for JSON documents. In addition, these require Semantic Web practitioners to know the details of the original format (e.g., XML) as well as the target domain ontology before implementing any transformation.

In the SPICE project we are researching on a novel approach to RDF lifting, which doesn't require to develop and use new language. Instead, we aim at reducing the effort of Semantic Web practitioners in dealing with heterogeneous data sources by providing a generic, domain-independent meta-model that can be queried with SPARQL. To achieve that, we approach the problem by decoupling the (syntactic) reengineering task from the (semantic) remodelling one, and focus exclusively on the first. To solve the re-engineering problem, we propose to use a *facade* to wrap the original resource and to make it query-able *as-if* it was RDF. Specifically, we contribute a general-purpose meta-model and associated algorithm for converting non-RDF resources into RDF: **Facade-X**. Our approach can be implemented easily by overriding the SERVICE operator and does not require to extend the SPARQL syntax. We compare our approach with the state of art methods RML and SPARQL Generate, and show how our solution has lower learning demands and cognitive complexity, and it is cheaper to implement and maintain, while having comparable extensibility and efficiency (in our naive implementation).

7.1 Requirements of LD transformers

In this section we define the requirements for generating RDF from heterogeneous sources. We elaborate on the requirements in detail, integrating and complementing the requirements introduced in (Lefrançois et al, 2017). Table 7.1 provides a summary of the requirements.

Table 7.1. Requirements of LD transformers

Requirement	Description
Transform	Transform several sources having heterogeneous formats
Query	Query resources having heterogeneous formats
Binary	Support the transformation of binary formats
Embed	Support the embedding of content in RDF
Metadata	Support the extraction of metadata embedded in files
Domain Independent	Make RDF in a domain-independent way, in the absence of a domain ontology
Low learning demands	Minimise the tools and languages that need to be learned
Low complexity	Minimise complexity of the queries
Meaningful abstraction	Enable focus on data structures rather than implementation details
Explorability	Enable data exploration without premature commitment to a mapping
Workflow	Integrate with a typical Semantic Web engineering workflow
Adaptable	Be generic but flexible and adaptable
Sustainable	Inform into a software that is easy to implement, maintain, and does not have evident efficiency drawbacks
Extendable	Support the addition of an open set of formats

First, we look into functional requirements. The main requirement is the ability to support users in transforming existing non-RDF resources having heterogeneous formats (**Transform**). In addition, the solution should be able to support cases in which practitioners only need to interrogate the content *as-if* it

was RDF (**Query**). A valid approach should be able to cope with binary resources as well as textual formats (**Binary**). In the cultural heritage domain, metadata files are typically associated to repositories of binary content such as images in various formats. Applications may need to transfer data and metadata in a single operation, embedding the binary content in a data value (**Embed**) and extracting metadata (**Metadata**) from the file (from EXIF annotations). Finally, resources should be transformed into RDF *before* decisions on domain modelling are made (**Domain Independent**). Second, we consider requirements related to usability and adoption. The approach should ideally limit the number of new languages and tools that need to be learned in order to transform and use non-RDF resources (**Low learning demands**). This can be expected to both encourage adoption and reduce the learning curve for new users. The code that the user is required to develop in order to access the resources should be as simple as possible (**Low complexity**). The approach should provide the user with a meaningful level of abstraction, enabling them to focus on the structure of the data (e.g., data rows and hierarchies) rather than the details of how the structure has been implemented (**Meaningful abstraction**). The approach should support an exploratory way of working in which the user does not have to prematurely commit to a particular ontology before they come to understand the data representation that they require (**Explorability**). The resulting technology should be easily combined with typical Semantic Web engineering workflow (**Workflow**). This requirement, already mentioned in (Lefrançois et al, 2017), is interpreted considering that the solution should rely as much as possible on already existing technologies typically used by our domain users. The approach should allow for a technical solution that is generic but easily **Adaptable** to user tasks, for example, supporting symbol manipulation, variable assignments, and data type manipulation. Finally, we look into requirements of software engineering. The approach should be **Sustainable** and inform a software that is easy to implement on top of existing Semantic Web technologies, easy to maintain, and does not have efficiency drawbacks compared to alternative state of the art solutions. Ultimately, the system should be easy to extend (**Extendable**) to support an open-ended set of formats.

7.2 Approach and solution design

We introduce a novel approach to interrogate non-RDF resources with SPARQL. We consider the task of transforming resources into RDF should be decoupled in two very different operations: (a) re-engineering, and (b) remodelling. We define *re-engineering* as the transformation of formats, at the *symbol* level. Instead, remodelling is the transformation at the *knowledge* level (Newell, 1982), where the original domain model is re-framed into a new one. From this perspective, we propose to solve the re-engineering problem automatically and delegating the remodelling to the RDF-aware user. As anticipated, we don't propose a new language. Instead, we choose to confront with the following research question: *How to use RDF to represent heterogeneous source formats?* To answer this question, instead of focusing into the domain model, we aim at transforming the source *meta-model* into RDF.

Indeed, formats such as CSV, JSON, or XML have different meta-models that can be mapped to RDF in many different ways. We rely on the notion of *facade* as "*an object that serves as a front-facing interface masking more complex underlying or structural code*"³⁴. Applied to our problem, a facade acts as a generic meta-model allowing (a) to inform the development of transformers from an open-ended set of formats, and (b) to generate RDF content in a consistent and predictable way.

To support the reader, we introduce a guide scenario reusing the data of the Tate Gallery collection, published on GitHub³⁵. The repository contains CSV tables with metadata of artworks and artists and a set of JSON files with details about each catalogue record, for example, with the hierarchy of archive subjects. Both types of resources include references to Web URLs pointing to digital images of the artworks. The file `artwork_data.csv` includes metadata of the artworks in the collection and references several external resources such as a JSON file with the artwork subject headings and a link to a JPG thumbnail image. Similarly, the file `artists_data.csv` includes the list of artists, linking to a collection of JSON documents for each one of them.

³⁴ The Facade Design Pattern: https://en.wikipedia.org/wiki/Facade_pattern (accessed 15/12/2020).

³⁵ <https://github.com/tategallery/collection>

Here, we describe the process of designing an RDF-based *facade* applicable to heterogeneous file formats. From the methodological standpoint, we rely on *design science* as a guiding principle. Pragmatically, we place into the *problem space* a collection of formats and associated meta-models, and on the *solution space* the components of the RDF(S) specifications, as described in the W3C documents³⁶.

The design process is as follows:

1. Our problem space includes the following formats: CSV, JSON, XML, HTML, Plan text, Binary.
2. The solution space includes the components from the RDF+RDFS specifications.
3. Initially, our facade specification is empty.
4. We select a format from the problem space, and observe its meta-model.
5. The meta-model is partitioned topologically, and its parts and relations mapped to RDF components, first by selecting the ones already collected and, in case something is missing, we pick a new component from the RDF specification.
6. We move to the next format, until the problem space is empty.

CSV. A CSV file is a resource, identifiable by a URI, which contains a dataset, composed of an ordered sequence of rows, which in turn contain an ordered sequence of data fields. Besides, we can already observe how all the formats in the problem space implement the basic notion of containment: they are all files including data, sometimes structured in parts, segmented in different ways according to the specific syntax. Therefore, we identify **containment** as a primary requirement of our facade. Now, we look at the solutions space for a component of RDF to use. The simplest way to express containment in RDF is with an RDF **property** linking the container with the contained item. This is our first component. However, what type of property should link the container to the contained elements in the case of a CSV? Rows are ordered; therefore, this case of containment can be represented as an **ordered sequence** (our second component). Relying on a recent survey on sequential linked data (Daga et al, 2021), we learn that there are several ways of representing sequences in RDF, and that some representations are more efficient to deal with in SPARQL than others. In the light of that analysis, we select container membership properties from the solution space (rdf:_1, rdf:_2, rdf:_n) instead of using a rdf:List.

What about data values? We observe how CSV data may have an optional “header”, where the first line is the list of field names. When this happens, we can use the property component and generate an RDF property reusing the field name, and minting an IRI with a conventional namespace. Otherwise, we can consider the values on each row as another sequence, and fallback to the ordered sequence component. This is an example from the Tate Gallery open data:

```
id,accession_number,artist,artistRole,artistId,title,dateText,medium,creditLine,year,acquisitionYear,dimensions,width,height,depth,units,inscription,thumbnailCopyright,thumbnailUrl,url
1035,A00001,"Blake, Robert",artist,38,A Figure Bowing before a Seated Old Man with his Arm Outstretched in Benediction. Verso: Indecipherable Sketch,date not known,"Watercolour, ink, chalk and graphite on paper. Verso: graphite on paper",Presented by Mrs John Richmond 1922,,1922,support: 394 x 419 mm,394,419,,mm,,http://www.tate.org.uk/art/images/work/A/A00/A00001_8.jpg,http://www.tate.org.uk/art/artworks/blake-a-figure-bowing-before-a-seated-old-man-with-his-arm-outstretched-in-benediction-a00001
1036,A00002,"Blake, Robert",artist,38,"Two Drawings of Frightened Figures, Probably for 'The Approach of Doom'",date not known,Graphite on paper,Presented by Mrs John Richmond 1922,,1922,support: 311 x 213
...
```

Applying the facade, the above CSV can be represented in RDF as follows:

```
@prefix fx: <http://sparql.xyz/facade-x/ns/>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@base <http://sparql.xyz/facade-x/data/>.
[
```

³⁶ RDF: <http://www.w3.org/TR/2014/NOTE-rdf11-primer-20140624/>
RDFS: <http://www.w3.org/TR/2014/REC-rdf-schema-20140525/>

```

rdf:_1 [:id "1034"; :artist "Blake Robert"; :artistId "38"; ...
rdf:_2 [:id "16216"; :artist "Williams Terrick" :artistId "2149"; ...
rdf:_3 [:id "12086"; :artist "Pissarro Lucien" :artistId "1777"; ... ]
...

```

Our facade currently includes the following RDF components: `rdf:Property` and `rdf:Seq` as two ways of representing containment. Let's move to our next format.

JSON. The JavaScript Object Notation is specified by ECMA. Interestingly, its specification is focused only to define the syntax of valid JSON texts and does not provide any semantics or interpretation of content conforming to that syntax³⁷. Here, we focus on the syntax, which specifies three types of elements: objects, expressed as a collection of key-value pairs, where keys are unique; arrays, which specifies sequences of values, and literals, which are either strings, numbers, boolean, or the primitive value null. Values can be literals, arrays, or objects. Considering our current facade, we can reuse `rdf:Property` to link object (containers) to values. Arrays can be represented by the ordered sequence component `rdf:Seq`. Literal values can be represented by selecting relevant XSD datatypes from the RDFS specification: `xsd:string`, `xsd:boolean`, `xsd:int`, etc ...

The following example shows a JSON document with metadata of an artist in the Tate Gallery Collection:

```

{
  "activePlaceCount": 2,
  "activePlaces": [
    {
      "name": "Ukrayina",
      "placeName": "Ukrayina",
      "placeType": "nation"
    },
    {
      "name": "Moskva, Rossiya",
      "placeName": "Moskva",
      "placeType": "inhabited_place"
    }
  ],
  "birth": {
    "place": {
      "name": "Kiyev, Ukrayina",
      "placeName": "Kiyev",
      "placeType": "inhabited_place"
    },
    "time": {
      "startYear": 1879
    }
  },
  "birthYear": 1879,
  "date": "1879\u20131935",
  "death": {
    "place": {
      "name": "Sankt-Peterburg, Rossiya",
      "placeName": "Sankt-Peterburg",
      "placeType": "inhabited_place"
    },
    "time": {
      "startYear": 1935
    }
  },
  "fc": "Kazimir Malevich",
  "gender": "Male",
  "id": 1561,
  "mda": "Malevich, Kazimir",
  "movements": [
    {

```

³⁷ <https://www.ecma-international.org/publications-and-standards/standards/ecma-404/>


```

    "era": {
      "id": 8,
      "name": "20th century 1900-1945"
    },
    "id": 318,
    "name": "Supremetism"
  }
],
"startLetter": "M",
"totalWorks": 2,
"url": "http://www.tate.org.uk/art/artists/kazimir-malevich-1561"
}

```

The JSON above will be represented as follows in RDF (in Turtle syntax):

```

@prefix fx: <http://sparql.xyz/facade-x/ns/>.
@prefix xyz: <http://sparql.xyz/facade-x/data/>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.

[ a fx:root ;
  xyz:activePlaceCount
    "2"^^xsd:int ;
  xyz:activePlaces
    [ rdf:_1
      [ xyz:name "Ukrayina" ;
        xyz:placeName "Ukrayina" ;
        xyz:placeType "nation"
      ] ;
      rdf:_2
      [ xyz:name "Moskva, Rossiya" ;
        xyz:placeName "Moskva" ;
        xyz:placeType "inhabited_place"
      ]
    ] ;
  xyz:birth
    [ xyz:place
      [ xyz:name "Kiyev, Ukrayina" ;
        xyz:placeName "Kiyev" ;
        xyz:placeType "inhabited_place"
      ] ;
      xyz:time
      [ xyz:startYear "1879"^^xsd:int ]
    ] ;
  xyz:birthYear "1879"^^xsd:int ;
  xyz:date "1879-1935" ;
  xyz:death
    [ xyz:place
      [ xyz:name
        "Sankt-Peterburg, Rossiya" ;
        xyz:placeName
        "Sankt-Peterburg" ;
        xyz:placeType
        "inhabited_place"
      ] ;
      xyz:time
      [ xyz:startYear "1935"^^xsd:int ]
    ] ;
  xyz:fc "Kazimir Malevich" ;
  xyz:gender "Male" ;
  xyz:id "1561"^^xsd:int ;
  xyz:mda "Malevich, Kazimir" ;
  xyz:movements
    [ rdf:_1
      [ xyz:era
        [ xyz:id "8"^^xsd:int ;
          xyz:name "20th century 1900-1945"
        ] ;

```

```

        xyz:id "318"^^xsd:int ;
        xyz:name "Supremetism"
    ]
] ;
xyz:startLetter "M" ;
xyz:totalWorks "2"^^xsd:int ;
xyz:url "http://www.tate.org.uk/art/artists/kazimir-malevich-1561"
] .

```

So far, we were able to express CSV and JSON data by using a limited set of RDF components. The JSON format required us to add a number of datatype formats, while CSV values could only be represented with the default datatype (string).

HTML and XML. We will tackle these two formats together, as their main difference is on syntactic aspects, while their meta-model is the same. In fact, both formats can be captured by the Document Object Model (DOM) specification³⁸, which we will refer to in the following description. However, it needs to be clarified how our methodology focuses on the elements of the syntax and does not aim at reproducing the DOM API in RDF. HTML/XML elements (also known as tags) can be definitely considered containers, so we can reuse both the `rdf:Property` component for specifying tag attributes, and container membership properties for specifying relations to child elements in the DOM tree. These may include text, which can be expressed as RDF literals of type `xsd:string`. What about element types (tag names)? Our current facade does not provide a solution of unary attributes. We can go back to our solution space and select the `rdf:type` relation. The range of the property will therefore be a `rdf:Resource` whose URI can be minted by using the tag name as local name. An interesting feature of XML and HTML is that these formats incorporate the notion of namespace. Therefore, we can also reuse namespaces which are used within the original document to name properties and types. Examples with HTML content will be presented later in this section.

So far, we collected the following components: `rdf:Property`, `rdf:ContainerMembershipProperty`, XSD datatypes, `rdf:type`. We complete our analysis with two more cases.

Text. Textual data is an interesting case where we can use containment to refer to different elements of the text. The whole content can be just including one single plain literal. In the alternative, the text can be tokenized and the resulting sequence represented in RDF. For the sake of our analysis, text can be considered a single container including a sequence of literals.

Binary. Binary content such as images can be also supported, by embedding the content in a single literal value, of datatype `xsd:binary64encoding`. This solution does not require to add components to the facade but still allows to *bring in* the content as linked data.

In both these cases, we can reuse the components already selected.

The output of our design activity is an RDF *facade*, which we named *Facade-X*. Next, we design a method to inject facades into SPARQL engines.

Our objective is to serve this content to the Semantic Web practitioners for exploration and reuse. To this end, we *overload* the SPARQL SERVICE operator by defining a custom URI-schema, associated with a set of parameters. The implementation of Facade-X will act as a *virtual* endpoint that can be queried exactly as a remote SPARQL endpoint. In order to instruct the query processor to delegate the execution to facade-x, we introduce a specific protocol for building an IRI to be used within SERVICE clauses: `x-sparql-anything:`. The related URI-schema supports an open-ended set of parameters specified by the facade implementations available. A minimal example only includes the resource locator, and guesses the data source type from the file extension. Options are embedded as key-value pairs, separated by comma.

Facade implementations are expected to either derive the source type from the resource locator or to obtain an indication of the type from the URI schema, for example, with an option "media-type":

```
x-sparql-anything:media-type=application/json; charset=UTF-8,location=...
```

³⁸ <https://dom.spec.whatwg.org/>

We now show how users can use Facade-X. Following our example scenario, users can select metadata from the CSV file and embed the content of the remote JPG thumbnails in RDF. Additional SERVICE clauses may integrate data from other files, for example, the JSON file with details about artwork subjects.

We leave the content of the CONSTRUCT section to be filled by the end-user:

```
PREFIX fx: <http://sparql.xyz/facade-x/ns/>
PREFIX xyz: <http://sparql.xyz/facade-x/data/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
CONSTRUCT {
  [...] # Amazing ontology mappings here
} WHERE {
  BIND (IRI(CONCAT(STR(tate:), "artwork-", ?id )) AS ?artwork) .
  BIND (IRI(CONCAT(STR(tate:), "artist-", ?artistId )) AS ?artist) .
  SERVICE <x-sparql-anything:csv.headers=true,location=file:./artwork_data.csv> {
    [] xyz:id ?id ;                xyz:artist ?artistLabel ;
      xyz:accessionId ?accId ;      xyz:artistId ?artistId ;
      xyz:title ?title;            xyz:medium ?medium ;
      xyz:year ?year ;             xyz:thumbnailUrl ?thumbnail .
  }
  # JPEG Thumbnail from the Web
  BIND (IRI(CONCAT("x-sparql-anything:location=", ?thumbnail )) AS ?embedJPG ) .
  SERVICE ?embedJPG { [] rdf:_1 ?imageInBase64 } .
  # JSON File with subjects
  BIND (IRI(CONCAT("x-sparql-anything:file:./artworks/", ?accId )) AS ?subJSON ) .
  SERVICE ?subJSON { [ xyz:id ?subjectId ; xyz:name ?subjectName ] } .
}
```

7.3 Description of the system and features

SPARQL Anything is distributed as a command line interface tool. The tool is open source and distributed under the commercial-friendly Apache Licence 2.0. The project is managed on GitHub at this address: <https://github.com/SPARQL-Anything/sparql.anything>.

7.3.1 Facade-X

SPARQL Anything uses a single generic abstraction for all data source formats called Facade-X. Facade-X is a simplistic meta-model used by SPARQL Anything transformers to generate RDF data from diverse data sources. Intuitively, Facade-X uses a subset of RDF as a general approach to represent the source content as-it-is but in RDF. The model combines two types of elements: containers and literals. Facade-X has always a single root container. Container members are a combination of key-value pairs, where keys are either RDF properties or container membership properties. Instead, values can be either RDF literals or other containers. This is a generic example of a Facade-X data object (more examples below):

```
@prefix fx: <http://sparql.xyz/facade-x/ns/> .
@prefix xyz: <http://sparql.xyz/facade-x/data/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
[] a fx:Root ; rdf:_1 [
  xyz:someKey "some value" ;
  rdf:_1 "another value with unspecified key" ;
  rdf:_2 [
    rdf:type xyz:MyType ;
    rdf:_1 "another value"
  ]
] .
```

7.3.2 Querying anything

The software extends the Apache Jena ARQ processors by overloading the SERVICE operator, as in the following example. Suppose having the following JSON file³⁹ as input:

³⁹ Available at <https://raw.githubusercontent.com/SPARQL-Anything/sparql.anything/main/examples/example1.json>

```
[
  {
    "name": "Friends",
    "genres": [
      "Comedy",
      "Romance"
    ],
    "language": "English",
    "status": "Ended",
    "premiered": "1994-09-22",
    "summary": "Follows the personal and professional lives of six twenty to thirty-
something-year-old friends living in Manhattan.",
    "stars": [
      "Jennifer Aniston",
      "Courteney Cox",
      "Lisa Kudrow",
      "Matt LeBlanc",
      "Matthew Perry",
      "David Schwimmer"
    ]
  },
  {
    "name": "Cougar Town",
    "genres": [
      "Comedy",
      "Romance"
    ],
    "language": "English",
    "status": "Ended",
    "premiered": "2009-09-23",
    "summary": "Jules is a recently divorced mother who has to face the unkind realities
of dating in a world obsessed with beauty and youth. As she becomes older, she starts
discovering herself.",
    "stars": [
      "Courteney Cox",
      "David Arquette",
      "Bill Lawrence",
      "Linda Videtti Figueiredo",
      "Blake McCormick"
    ]
  }
]
```

Users can select the TV series starring "Courteney Cox" with the SPARQL query:

```
PREFIX xyz: <http://sparql.xyz/facade-x/data/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?seriesName
WHERE {
  SERVICE <x-sparql-anything:https://raw.githubusercontent.com/SPARQL-
Anything/sparql.anything/main/examples/example1.json> {
    ?tvSeries xyz:name ?seriesName .
    ?tvSeries xyz:stars ?star .
    ?star ?li "Courteney Cox" .
  }
}
```

and get this result without caring of transforming JSON to RDF:

```
seriesName
"Cougar Town"
"Friends"
```

The SPARQL Anything query engine will act as a *virtual endpoint* that can be queried exactly as a remote SPARQL endpoint. In order to instruct the query processor to delegate the execution to our system, users must use the following URI-schema within SERVICE clauses.

`x-sparql-anything ':' ([option] ('=' [value]))? ',')+`

A minimal URI that uses only the resource locator is also possible.

`x-sparql-anything ':' URL`

In this case SPARQL Anything guesses the data source type from the file extension.

Table 7.2 General purpose options

Option name	Description	Valid Values	Default Value
location	The URL of the data source.	Any valid URL.	Mandatory
root	The IRI of generated root resource.	Any valid IRI.	location + '#'
media-type	The media-type of the data source.	Any valid Media-Type . Supported media-types: application/xml, image/png, text/html, application/octet-stream, application/json, image/jpeg, image/tiff, image/bmp, text/csv, image/vnd.microsoft.icon, text/plain	No value (the media-type will be guessed from the file extension)
namespace	The namespace prefix for the properties that will be generated.	Any valid namespace prefix.	http://sparql.xyz/facade-x/data/
blank-nodes	It tells SPARQL Anything to generate blank nodes or not.	true/false	true
triplier	It forces SPARQL Anything to use a specific triplifier for transforming the data source	A canonical name of a Java class	No value
charset	The charset of the data source.	Any charset.	UTF-8
metadata	It tells SPARQL Anything to extract metadata from the data source and to store it in the named graph with URI http://sparql.xyz/facade-x/data/metadata	true/false	false

7.3.3 Supported formats

Currently, SPARQL Anything supports the following formats: "json", "html", "xml", "csv", "bin", "png", "jpeg", "jpg", "bmp", "tiff", "tif", "ico", "txt". For example, JSON, HTML, and Binary content are transformed as in Tables 7.2-7.3-7.4. More examples can be found on the GitHub project page.

Table 7.2. JSON


Input	Triplification
<code>{ "stringArg": "stringValue",</code>	<code>@prefix xyz: <http://sparql.xyz/facade-x/data/> . @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .</code>

<pre>"intArg":1, "booleanArg":true, >nullArg": null, "arr":[0,1] }</pre>	<pre>@prefix xsd: <http://www.w3.org/2001/XMLSchema#> . [xyz:arr [rdf:_0 "0"^^xsd:int ; rdf:_1 "1"^^xsd:int] ; xyz:booleanArg true ; xyz:intArg "1"^^xsd:int ; xyz:stringArg "stringValue"] .</pre>
---	---

Table 7.3 HTML

Input	Triplification
<pre><html> <head> <title>Hello world!</title> </head> <body> <p class="paragraph">Hello world</p> </body> </html></pre>	<pre>@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> . @prefix xhtml: <http://www.w3.org/1999/xhtml#> . [a xhtml:html ; rdf:_1 [a xhtml:head ; rdf:_1 [a xhtml:title ; rdf:_1 "Hello world!"] ; rdf:_2 [a xhtml:body ; rdf:_1 [a xhtml:p ; rdf:_1 "Hello world" ; xhtml:class "paragraph"]]] .</pre>

Table 7.4. Binary content

Input	Triplification
	<pre>@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> [rdf:_1 " /9j/4AAQSkZJRgABAQEASABIAAD/4QmsRXhpZGAAASUkqAAgAAAALAA8BAgAGAAAAkgAAABABAgOAAAaMAAAABIBAwABAAAAQAAA BoBBQABAAAApGAAABsBBQABAAAAAgAAACgBAwABAAAAAgAAAEBAgALAAAAAgAAADIBAgAUAAAAAgAAABMCwABAAAAAgAAAGmHBAAB AAAA1gAAACWIBAAAAA0gMAAOQDAABDYW5vbG9DYW5vb2JmZT1mGNDBEAEgAAAAABAAAAAASAAAAAEAAABHSU1QID1uNC41AAyMDA40jA 30jMxIDFwOjM4OjExAB4AmoIFAAEAAABEAgAAyYIFAAEAAABMAGAAIoGDAEAAABAAAAAJ4gDAEAAABKAAAAAJAHAAQAAAAWmJIXA5 ACABQAAABUAgAAABJACABQAAABBoGAAAEZEHAAQAAAAABAgMAAZIKAAEAAAB8AgAAApIFAAEAAACEAgAAABJIKAAEAAACMAgAAAB5IDAAEAA AAFAAAACZIDAAEAAAJAAACpIFAAEAAACUAgAAhpiHAAGBAACcAgAAkIICAAMAAAAwMAAAkZICAAMAAAAwMAAAkpICAAMAAAAwMAAA AKAHAAQAAAAwMTAwAAADAAEAAABAAAAAgAAEAAABKAAAAA6AAEAAEAAABEAAABAAEAAEAAAC0AAwAADqIFAAEAAACkAAwAAD6IFAAE AAACsAAwAAEKIDAAEAAACAAAAAQDAEAAEAAAAAAAgQDAEAAABAAAAA6QDAEAAEAAAAABgQDAEAAEAAAAAAAEAAACgAA AARwAAAAoAAAAyMDA40jA10jMwIDE1OjU20jAxADIwMDg6MDU6MzAgMTU6NTY6MDEAAAGAHAAAAQAAoAUAAAAABA[...]"^^<http: //www.w3.org/2001/XMLSchema#base64Binary>] .</pre>

7.3.4 Functionalities and Command Line Interface

SPARQL Anything can be used to interrogate one (or more) non-RDF resources with plain SPARQL 1.1. Therefore, the system behaves essentially as a SPARQL query engine, receiving as input a query and returning either a SPARQL Result Set (for SELECT/ASK queries) or an RDF stream (for CONSTRUCT/DESCRIBE query types). However, the SPARQL Anything command line interface provides a number of features:

- **Query.** The path to the file storing the query to execute or the query itself.
- **Output format.** Users can specify the format of the output (parameter `-f` or `--format`). Supported values are JSON, XML, CSV, TEXT, TTL, NT, NQ.
- **Parametrized queries.** The system supports parametrized queries using the BASIL syntax convention (Daga et al, 2015). Users can specify a SPARQL Result Set file (option `-i` or `--input`) to provide variable parameter values. When present, the query is pre-processed by substituting variable names with values from the bindings provided. The system will repeat the query for each of the provided bindings. Variables will be replaced with input parameters provided by the bindings. These variables

need to follow a set of conventions. The syntax is based on the underscore character: '_', and can be easily learned by examples:

- `?_name` (single underscore): The variable specifies a mandatory parameter. The value is incorporated in the query as plain literal.
- `?__name` (double underscore): The parameter name is optional.
- `?_name_iri`: The variable is substituted with the parameter value as an IRI.
- `?_name_en`: The parameter value is replaced as a literal with the language annotation 'en'.
- `?_name_integer`: The parameter value is a literal with the XSD datatype 'integer'.
- `?_name_prefix_datatype`: The parameter value is considered as literal and the datatype 'prefix:datatype' is added during substitution. The prefix must be included in query header.
- **Load RDF data.** With this feature, it is possible to reuse content from a previously performed transformation and execute the query against an existing (set of) RDF files (option `-l` or `--load`). The option requires the path to one RDF file or a folder including a set of files to be loaded. When present, the data is loaded in memory and the query executed against it.
- **Output filename.** Users can specify the name of the output file (option `-o` or `--output`)
- **Output filename pattern.** Users can specify an output filename pattern, reusing parameter values passed in combination with the option `--input`. Variables should start with '?' And refer to bindings from the input file. This option can only be used in combination with 'input' and is ignored otherwise. This option overrides 'output'.

An executable JAR can be obtained from the releases page on GitHub⁴⁰. The jar can be executed as follows:

```
usage: java -jar sparql.anything-<version> -q query [-f format] [-i
        filepath] [-l path] [-o filepath]
```

Logging can be configured adding the following option (using SLF4J):

```
-Dorg.slf4j.simpleLogger.defaultLogLevel=trace
```

7.4 Example LD Transformer: the IMMA catalogue

In this section we describe how we used SPARQL Anything to construct a knowledge graph of artworks and artists from the Website of the Irish Museum of Modern Art (IMMA), one of the key partners of the SPICE project (Figure 7.1). The Website includes web pages for each one of the artists and artworks in the catalogue, including images of the artworks and essential metadata. For example, Figure 7.2 shows the web page of the artist Marina Abramovic, including useful information such as birth date and biography. Crucially, the web page includes the list of works included in the IMMA catalogue. Similarly, Figure 7.3 shows the webpage of one of Abramovic's works. The web page includes information from the museums' collection metadata, including a description, the type of medium used in the work, credit and copyright information, the catalogue item number, and the official caption of the image report of the work (see Figure 7.4). It needs to be noted how here we do not focus on the metadata schema itself, meaning we do not focus on the vocabulary to be used for representing the content as Linked Data. Instead, we look into the content itself and how it can be captured in order to produce one possible RDF representation, among the many possible. Details about the vocabulary and ontology used can be found in deliverable "D6.2 Initial Ontology Network Specification".

However, our job is to extract this information from the web content. We explore the website and find a web page listing all the artists (Figure 7.5). We plan to create a JSON-LD file for each one of the artists and artworks included in the catalogue.

⁴⁰ <https://github.com/SPARQL-Anything/sparql.anything/releases>

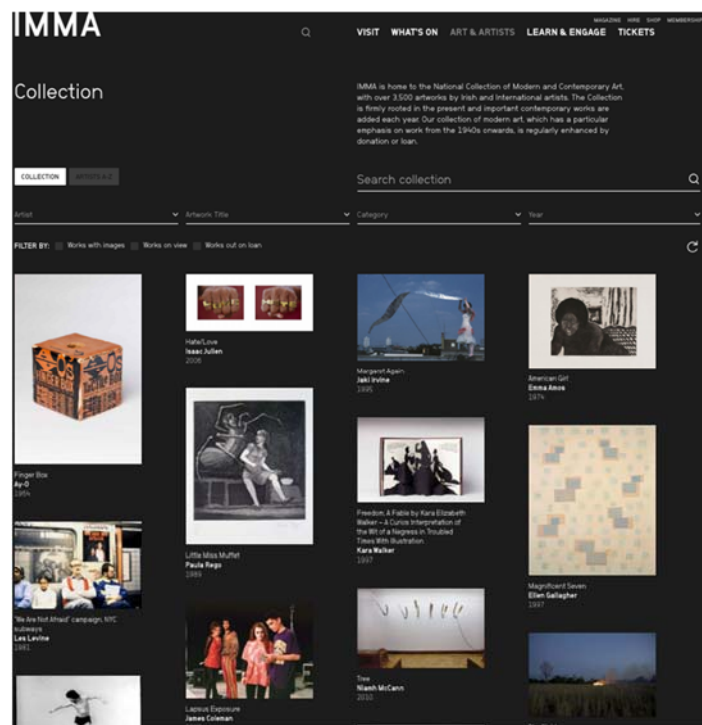


Figure 7.1. The IMMA Catalogue

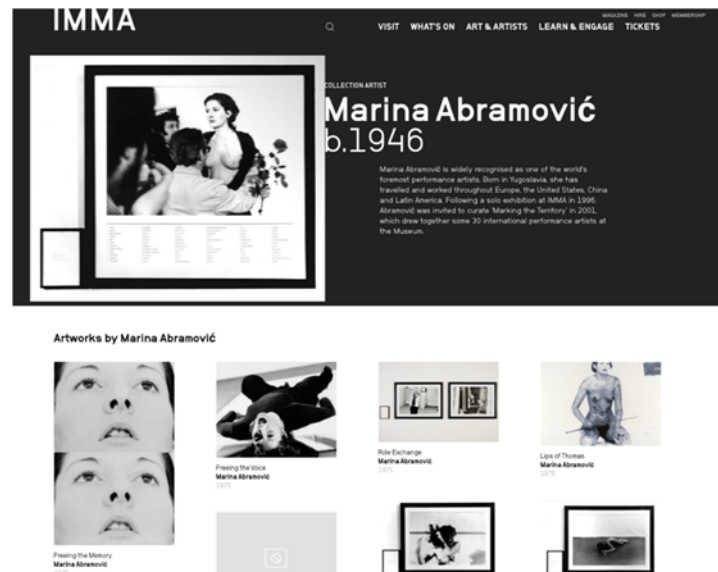


Figure 7.2 The artist page of Marina Abramovic⁴¹

⁴¹ <https://imma.ie/artists/marina-abramovic/>

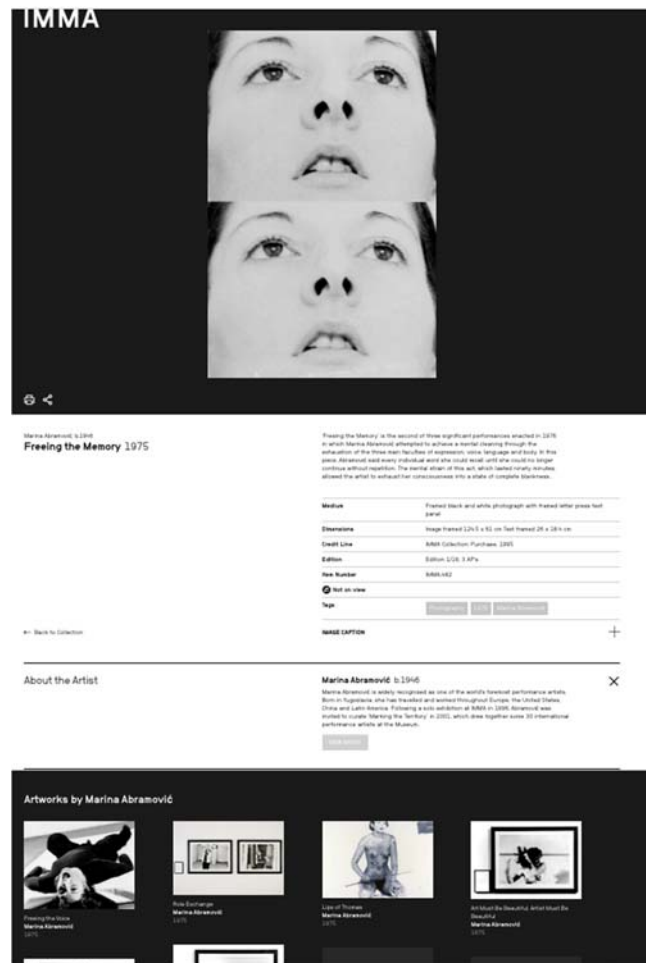


Figure 7.3 The artwork page for Feeling the Memory (1975) by Marina Abramovic⁴²

⁴² <https://imma.ie/collection/freeing-the-memory/>

'Freeing the Memory' is the second of three significant performances enacted in 1976 in which Marina Abramović attempted to achieve a mental cleaning through the exhaustion of the three main faculties of expression, voice, language and body. In this piece, Abramović said every individual word she could recall until she could no longer continue without repetition. The mental strain of this act, which lasted ninety minutes, allowed the artist to exhaust her consciousness into a state of complete blankness.

Medium	Framed black and white photograph with framed letter press text panel
Dimensions	Image framed 124.5 x 61 cm Text framed 26 x 18.4 cm
Credit Line	IMMA Collection: Purchase, 1995
Edition	Edition 1/16, 3 AP's
Item Number	IMMA.482
Not on view	
Tags	Photography 1975 Marina Abramović

IMAGE CAPTION

Marina Abramović, *Freeing the Memory*, 1975. Framed black and white photograph with framed letter press text panel. Image framed 124.5 x 61 cm Text framed 26 x 18.4 cm, Collection Irish Museum of Modern Art, Purchase, 1995

Figure 7.4. Detail of the web page for *Feeling the Memory* (1975) by Marina Abramović⁴³

IMMA					
VISIT WHAT'S ON ART & ARTISTS LEARN & ENGAGE TICKETS					
Artists A-Z					
Search artists					
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z					
A	Abramović, Marina	Abramović, Marina	Abramović, Marina	Abramović, Marina	Abramović, Marina
B	Bach, Johann Sebastian	Bach, Johann Sebastian	Bach, Johann Sebastian	Bach, Johann Sebastian	Bach, Johann Sebastian
C	Cadogan, John	Cadogan, John	Cadogan, John	Cadogan, John	Cadogan, John
D	Dalí, Salvador	Dalí, Salvador	Dalí, Salvador	Dalí, Salvador	Dalí, Salvador
E	Edwards, George	Edwards, George	Edwards, George	Edwards, George	Edwards, George
F	Fleming, John	Fleming, John	Fleming, John	Fleming, John	Fleming, John
G	Gauguin, Paul	Gauguin, Paul	Gauguin, Paul	Gauguin, Paul	Gauguin, Paul
H	Habsburg, Charles	Habsburg, Charles	Habsburg, Charles	Habsburg, Charles	Habsburg, Charles
I	Irish, John	Irish, John	Irish, John	Irish, John	Irish, John
J	James, John	James, John	James, John	James, John	James, John
K	Kelly, John	Kelly, John	Kelly, John	Kelly, John	Kelly, John
L	Larkin, John	Larkin, John	Larkin, John	Larkin, John	Larkin, John
M	Marshall, John	Marshall, John	Marshall, John	Marshall, John	Marshall, John
N	Nelson, John	Nelson, John	Nelson, John	Nelson, John	Nelson, John
O	O'Connell, John	O'Connell, John	O'Connell, John	O'Connell, John	O'Connell, John
P	Pearce, John	Pearce, John	Pearce, John	Pearce, John	Pearce, John
Q	Quinn, John	Quinn, John	Quinn, John	Quinn, John	Quinn, John
R	Reilly, John	Reilly, John	Reilly, John	Reilly, John	Reilly, John
S	Shannon, John	Shannon, John	Shannon, John	Shannon, John	Shannon, John
T	Taylor, John	Taylor, John	Taylor, John	Taylor, John	Taylor, John
U	Upton, John	Upton, John	Upton, John	Upton, John	Upton, John
V	Vincent, John	Vincent, John	Vincent, John	Vincent, John	Vincent, John
W	Ward, John	Ward, John	Ward, John	Ward, John	Ward, John
X	Xavier, John	Xavier, John	Xavier, John	Xavier, John	Xavier, John
Y	Yates, John	Yates, John	Yates, John	Yates, John	Yates, John
Z	Zimmerman, John	Zimmerman, John	Zimmerman, John	Zimmerman, John	Zimmerman, John

Figure 7.5 List of artists in the IMMA catalogue⁴⁴

⁴³ <https://imma.ie/collection/freeing-the-memory/>

⁴⁴ <https://imma.ie/artists/>

```
java -jar sparql-anything-0.0.4-SNAPSHOT.jar
```

```

▼<div id="az-group">
  ▼<div class="letter-group" id="letter-group-A">
    <h4>A</h4>
    ▼<ul>
      ▶<li class="artist" data-image="https://imma.ie/wp-content/uploads/2018/11/48-676x1024.jpg" data-filter="collection">_</li>
      ▶<li class="artist" data-image="/wp-content/themes/imma/css/img/no-img-dark.png" data-filter="collection usa">_</li>
      ▼<li class="artist" data-image="https://imma.ie/wp-content/uploads/2018/11/845.jpg" data-filter="collection">
        ▼<a href="https://imma.ie/artists/marina-abramovic/">= $0
          "Abramović, Marina "
          <span style="display: inline-block; width: 100px; height: 100px; background-color: #ccc; vertical-align: middle;">
```

```
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix xhtml: <http://www.w3.org/1999/xhtml#>

select distinct ?artistUrl ?artistNickname
WHERE {
    SERVICE <x-sparql anything:media-type=text/html,html.selector=#az-group,location=https://imma.ie/artists/> {
        [] xhtml:data-image [] ;
        rdf:_1 [
            xhtml:href ?artistUrl ;
            ?i [ a xhtml:span ; rdf:_1 ?artistNickname ]
        ].
    } .
    BIND ( IRI( CONCAT("https://w3id.org/spice/imma/agent/", ?artistNickname) ) as ?artistEntity ) .
}
```

```
fx -q imma-artists.sparql -o imma-artists.xml -f xml
```

```
<?xml version="1.0"?>
<sparql xmlns="http://www.w3.org/2005/sparql-results#">
  <head>
    <variable name="artistUrl"/>
    <variable name="artistNickname"/>
  </head>
  <results>
    <result>
      <binding name="artistUrl">
        <literal>https://imma.ie/artists/william-leech</literal>
      </binding>
      <binding name="artistNickname">
        <literal>leech-william</literal>
      </binding>
    </result>
    <result>
      <binding name="artistUrl">
        <literal>https://imma.ie/artists/marie-foley</literal>
      </binding>
      <binding name="artistNickname">
        <literal>foley-marie</literal>
      </binding>
    </result>
  </results>
</sparql>
```

44

```
<div class="row imma_eventsbanner cat-exhibition" style>
  <!-- Carousel -->
  <div class="owl-carousel owl-theme owl-loaded hide"></div>
  <script></script>
  <div class="clearer"></div>
  <!-- Heading Block -->
  <div class="col_6 flush banner_heading">
    <label></label>
    <h1 class="low_lrg">
      <strong>Marina Abramović</strong>
      <span class="dob">b.1946</span>
    </h1>
  </div>
  <div class="clearer"></div>
  <!-- Text Block -->
  <div class="col_5 banner_detail">
    <p class="header_para"> == $0
      "Marina Abramović is widely recognised as one of the world's foremost performance artists. Born in Yugoslavia, she has
      travelled and worked throughout Europe, the United States, China and Latin America. Following a solo exhibition at IMMA in
      1996, Abramović was invited to curate 'Marking the Territory' in 2001, which drew together some 30 international performance
      artists at the Museum."
    </p>
  </div>
```

Instead, the following excerpt shows the section with the list of artworks, each one included into a div tag with classes "col_3 flush grid-item collection-grid-item". From this section, we are interested in the artwork web page, year, title, and image URL:

```
<div class="col_12"> == $0
  <h2>Artworks by Marina Abramović</h2>
  <div class="isotope" style="position: relative; height: 1440.89px;">
    <div class="col_3 flush grid-item collection-grid-item" data-obj="obj_23818" data-paged style="p
    px; top: 0px;">
      <a href="https://imma.ie/collection/freeding-the-memory/">
        <p>
          
        </p>
      </a>
      <h4>
        <a href="https://imma.ie/collection/freeding-the-memory/">
          <span class="artwork-name">Freeing the Memory</span>
        </a>
        <span class="artist-name">
          <a href="/artists/marina-abramovic">Marina Abramović</a>
        </span>
        <span class="artwork-year">1975</span>
      </h4>
    </div>
    <div class="col_3 flush grid-item collection-grid-item" data-obj="obj_23819" data-paged style="p
    39.9920x; top: 0px;"></div>
```

We can query the artist page following the same approach used for obtaining the list of artists. However, this time we want to generate a Linked Data object about the artist, defining mappings to one or more of the SPICE ontologies. This is shown in figures 7.7 and 7.8 (in the following page). In addition, our query includes two parameters: the artist nickname and artist web page, since we want to run this against all the artists and generate one file for each one of them.

With the following command, we extract data from the artists' Web page and build one JSON-LD file each, using the previously extracted list of artists as input:

```
fx -q imma-artist.sparql -i imma-artists.xml -p "artists/?artistNickname.jsonld" -f json
```

We now have a collection of JSON-LD files ready to be published into the Linked Data Layer. However, we want to follow a similar approach to produce a linked data version of the artworks in the IMMA catalogue. This time we don't have a web page listing all the artworks. Fortunately, SPARQL Anything allows to load a collection of RDF files and run a query against it. Exploiting this feature, we generate a SPARQL Result Set file listing artwork webpages and nicknames mentioned in Artists' JSON-LD file generated before:

```
fx -q imma-artworks.sparql -l artists/ -o imma-artworks.xml -f xml
```

The `-l` option (`--load`) instructs the tool to load the files from the given folder in an in-memory RDF dataset. The query used is as simple as the following:

```
PREFIX arco: <https://w3id.org/arco/ontology/arco/>
PREFIX schema: <http://schema.org/>

SELECT DISTINCT ?artworkUrl ?artworkNickname
WHERE {
  GRAPH ?G {
    [] a arco:CulturalProperty ;
      schema:url ?artworkUrl
      .
    BIND ( REPLACE(REPLACE(?artworkUrl, "https://imma.ie/collection/", ""), "/", "") AS ?artworkNickname )
  }
}
```

We can reuse the list of bindings in `imma-artworks.xml` to run another query, specifically designed to extract content from the artwork web page, for example, the content of the table illustrated in Figure 7.5. The following command extracts data from the artworks' Web pages and build one JSON-LD file each.

```
fx -q imma-artwork.sparql -i imma-artworks.xml -p "artworks/?artworkNickname.jsonld" -f json
```

With this command, our workflow is completed and we were able to build a metadata catalogue as Linked Data with the sole use of SPARQL Anything. Figures 7.9 and 7.10 report an extract of the JSON-LD of the artist and artwork mentioned in the example. The collection of JSON-LD files is ready to be loaded in the SPICE Linked Data Hub. This tutorial can be reproduced following the instructions at <https://github.com/sparql-anything/showcase-imma>.

```
prefix fx: <http://sparql.xyz/facade-x/ns/>
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix xhtml: <http://www.w3.org/1999/xhtml#>
prefix dom: <https://html.spec.whatwg.org/#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX spice: <https://w3id.org/spice/imma/>
PREFIX arco: <https://w3id.org/arco/ontology/arco/>
PREFIX arco-cd: <https://w3id.org/arco/ontology/context-description/>
PREFIX arco-core: <https://w3id.org/arco/ontology/core/>
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX schema: <http://schema.org/>

CONSTRUCT {
  ?artistEntity a foaf:Person ;
    rdfs:label ?name ;
    dc:source ?_artistUrl ;
    schema:url ?_artistUrl ;
    schema:birthDate ?dob ;
    dc:description ?artistDescription .
  ?artefactEntity a arco:CulturalProperty, schema:CreativeWork ;
    dc:creator ?artistEntity ;
    dc:source ?artistUrl ;
    schema:url ?artefactUrl ;
    arco:hasRelatedAgency <https://w3id.org/spice/institute/imma> ; # keeper
    arco-cd:hasDocumentation ?artefactDocumentation ;
    arco-cd:hasTitle ?artefactTitle ;
    schema:author ?artistEntity ;
    schema:dateCreated ?artefactYear ;
    arco-cd:hasAuthorshipAttribution ?artefactAttribution .
  ?artefactAttribution a arco:AuthorshipAttribution ;
    arco-cd:hasAttributedAuthor ?artistEntity .
  ?artefactDocumentation
    arco-cd:hasDocumentationType <https://w3id.org/spice/doc_type/catalogue_photograph> ;
    arco-cd:url ?artefactImageUrl
} WHERE {
```

Figure 7.7 SPARQL Anything query for IMMA artist page: mapping Facade-X RDF to SPICE ontologies.

```

} WHERE {
  # To be implemented by BASIL variables
  BIND ( IRI( CONCAT("https://w3id.org/spice/imma/agent/", ?_artistNickname) ) as ?artistEntity ) .
  BIND ( IRI( CONCAT("x-sparql-anything:media-type=text/html,location=", ?_artistUrl) ) as ?artistFx ) .
  SERVICE ?artistFx {
    {
      [] xhtml:class "low_lrg" ;
      ?l145 [
        rdf:type xhtml:strong ;
        rdf:_1 ?name ]
    }
    UNION {
      [] xhtml:class "low_lrg" ;
      ?l112 [
        rdf:type xhtml:span ;
        rdf:_1 ?dob ]
    }
    UNION {
      [] xhtml:class "header_para" ;
      rdf:_1 ?artistDescription
    }
    UNION {
      ?n rdf:type xhtml:div ; #;
      xhtml:class "col_3 flush grid-item collection-grid-item " ;
      xhtml:data-obj ?objectId ;
      rdf:_1 [
        rdf:type xhtml:a ;
        xhtml:href ?artefactUrl ]
      ;
      rdf:_2 [
        rdf:_1/rdf:_1/rdf:_1 ?artefactTitle ;
        rdf:_3/rdf:_1 ?artefactYear
      ]
    }
    OPTIONAL {
      # we may not have an image
      ?n rdf:_1/rdf:_1/rdf:_1 [
        rdf:type xhtml:img ;
        xhtml:src ?artefactImageUrl ] .
      # We have a documentation only if we have an image
      BIND ( IRI( CONCAT("https://w3id.org/spice/imma/documentation/", ?artefactNickname ) ) as ?artefactDocumentation ) .
    }
    UNION {
      [] rdf:_1 [ rdf:type xhtml:h2 ] ; rdf:_2 [ rdf:type xhtml:p ; rdf:_1 ?artistDescription ] .
    }
  }
  BIND ( REPLACE(REPLACE( ?artefactUrl , 'https://imma.ie/collection/', '' ),"/","", "" ) AS ?artefactNickname ) .
  BIND ( IRI( CONCAT( "https://w3id.org/spice/imma/artefact/", ?artefactNickname ) ) as ?artefactEntity ) .
  BIND ( IRI( CONCAT( "https://w3id.org/spice/imma/attribution/", ?artefactNickname ) ) as ?artefactAttribution ) .
}

```

Figure 7.8 SPARQL Anything query for IMMA artist page: querying the Web page and minting entity IRIs.


```

{
  "@graph": [ {
    "@id": "spice:agent/abramovic-marina",
    "@type": "foaf:Person",
    "dc:description": "Marina Abramović is widely recognised as one of the world's foremost performanc",
    "dc:source": "https://imma.ie/artists/marina-abramovic/",
    "schema:birthDate": "b.1946",
    "schema:url": "https://imma.ie/artists/marina-abramovic/",
    "rdfs:label": "Marina Abramović"
  }, {
    "@id": "spice:artefact/art-must-be-beautiful-artist-must-be-beautiful",
    "@type": [ "schema:CreativeWork", "arco:CulturalProperty" ],
    "creator": "spice:agent/abramovic-marina",
    "author": "spice:agent/abramovic-marina",
    "schema:dateCreated": "1975",
    "schema:url": "https://imma.ie/collection/art-must-be-beautiful-artist-must-be-beautiful/",
    "hasRelatedAgency": "https://w3id.org/spice/institute/imma",
    "hasAuthorshipAttribution": "spice:attribution/art-must-be-beautiful-artist-must-be-beautiful",
    "arco-cd:hasTitle": "Art Must Be Beautiful, Artist Must Be Beautiful"
  }, {
    "@id": "spice:artefact/freeing-the-body",
    "@type": [ "schema:CreativeWork", "arco:CulturalProperty" ],
    "creator": "spice:agent/abramovic-marina",
    "author": "spice:agent/abramovic-marina",
    "schema:dateCreated": "1975",
    "schema:url": "https://imma.ie/collection/freeing-the-body/",
    "hasRelatedAgency": "https://w3id.org/spice/institute/imma",
    "hasAuthorshipAttribution": "spice:attribution/freeing-the-body",
    "arco-cd:hasTitle": "Freeing the Body"
  }, {

```

Figure 7.9. Excerpt from the JSON-LD file for the IMMA artist Marina Abramovic.

```

{
  "@graph": [ {
    "@id": "spice:artefact/freeing-the-memory",
    "@type": "schema:CreativeWork",
    "dc:description": "'Freeing the Memory' is the second of three significant performances enacted in 1976 in which",
    "dc:source": "https://imma.ie/collection/freeing-the-memory/",
    "schema:creditText": "IMMA Collection: Purchase, 1995",
    "image": "spice:documentation/freeing-the-memory",
    "maintainer": "https://w3id.org/spice/institute/imma",
    "schema:material": "Framed black and white photograph with framed letter press text panel",
    "schema:size": "Image framed 124.5 x 61 cm Text framed 26 x 18.4 cm",
    "schema:version": "Edition 1/16, 3 AP's",
    "arco-cd:hasInventory": "IMMA.482"
  }, {
    "@id": "spice:documentation/freeing-the-memory",
    "@type": "schema:ImageObject",
    "schema:caption": "Marina Abramović, Freeing the Memory, 1975, Framed black and white photograph with framed lett",
    "schema:copyrightNotice": "For copyright information, please contact the IMMA Collections team.",
    "maintainer": "https://w3id.org/spice/institute/imma",
    "schema:url": "https://imma.ie/wp-content/uploads/2018/11/850.jpg"
  } ],
  "@context": {
    "copyrightNotice": {
      "@id": "http://schema.org/copyrightNotice",

```

Figure 7.10. Excerpt from the JSON-LD file for the IMMA artwork Freeing the Memory.

7.5 Evaluation

We conduct a comparative evaluation of SPARQL Anything with respect to the state of art methods RML and SPARQL Generate. First, we analyse in a quantitative way the cognitive complexity of the frameworks. Second, we conduct a performance analysis of the reference implementations. Finally, we discuss the approaches in relation to the requirements elicited in Section 7.1, in a more qualitative way. Competency questions, queries, experimental data, and code used for the experiment are available on the GitHub repository of the SPARQL Anything project⁴⁵.

7.5.1 Cognitive Complexity Comparison

We present a quantitative analysis on the cognitive complexity of SPARQL Anything, SPARQL Generate and RML frameworks. One effective measure of complexity is the number of distinct items or variables that need to be combined within a query or expression [Halford and Andrews 2004]. Such a measure of complexity has previously been used to explain difficulties in the comprehensibility of Description Logic statements [Warren et al. 2015]. Specifically, we counted the number of tokens needed for expressing a set of competency questions. We selected four JSON files from the case studies of the SPICE project where each file contains the metadata of artworks of a collection. Each file is organised as a JSON array containing a list of JSON objects (one for each artwork). This simple data structure avoids favouring one approach over the others. Then, an analysis of the schema of the selected resources allowed us to define a set of 12 competency questions (CQs) that were then specified as SPARQL queries or mapping rules according to the language of each framework, in particular: (i) 8 CQs (named q1-q8), aimed at retrieving data from the sources, were specified as SELECT queries (according to SPARQL Anything and SPARQL Generate); (ii) 4 CQs (named q9-q12), meant for transforming the source data to RDF, were expressed as CONSTRUCT queries (according to SPARQL Anything and SPARQL Generate) or as mapping rules complying with RML. These queries/rules intend to generate a blank node for each artwork and to attach the artwork's metadata as data properties of the node. Finally, we tokenized the queries (by using "({},;\n\t\r" as token delimiters) and we computed the total number of tokens and the number of distinct tokens needed for each query. By observing the average number of tokens per query we can conclude that RML is very verbose (109.75 tokens) with respect to SPARQL Anything (26.25 tokens) and SPARQL Generate (30.75 tokens) whose verbosity is similar (they differ of the ~6.5%). However, the average number of distinct tokens per query shows that SPARQL Anything requires less cognitive load than other frameworks. In fact, while SPARQL Anything required 18.25 distinct tokens, SPARQL Generate needed 25.5 distinct tokens (~39.72% more) and RML 45.25 distinct tokens (~150% more).

Performance Comparison. We assessed the performance of three frameworks in generating RDF data. All of the tests described below were run three times and the average time among the three executions is reported. The tests were executed on a MacBook Pro 2020 (CPU: i7 2.3 GHz, RAM: 32GB).

Figure 7.1 shows the time needed for evaluating the SELECT queries q1-q8 and for generating the RDF triples according to the CONSTRUCT queries/mapping rules q9-q12. The three frameworks have comparable performance.

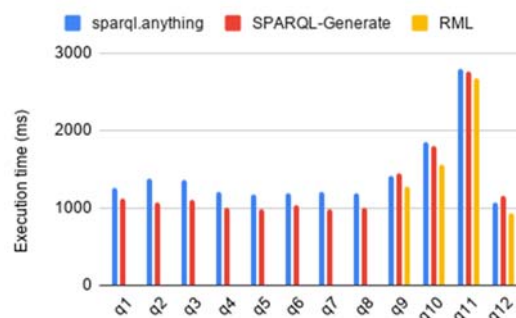


Figure 7.1 Execution time per query

⁴⁵ <https://github.com/SPARQL-Anything/sparql.anything/tree/main/experiment>

We also measured the performance in transforming input of increasing size. To do so, we repeatedly concatenated the data sources in order to obtain a JSON array containing 1M JSON objects and we cut this array at length 10, 100, 1K, 10K and 100K. We ran the query/mapping q12 on these files and we measured the execution time shown in Figure 7.2. We observe that for inputs with size smaller than 100K the three frameworks have equivalent performance. With larger inputs, SPARQL Anything is slightly slower than the others. The reason is that, in our naive implementation, the data source is completely transformed and loaded into an RDF dataset in memory, before the query is evaluated. A more advanced implementation should instead stream the triples during query execution, achieving better performance on large volumes of input data. However, experiments show how our prototype is usable also on larger data files.

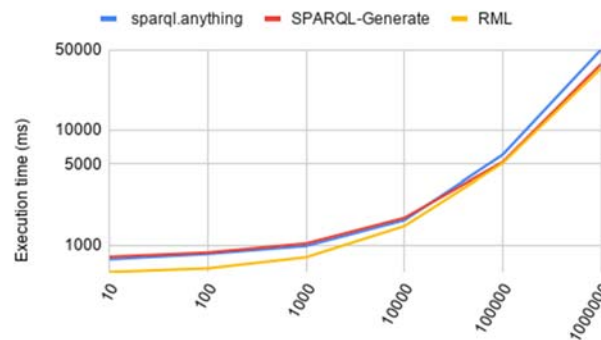


Figure 7.2 Execution time with increasing input size

7.5.2 Requirements' satisfaction and discussion

First, we look into functional requirements.

Transform, Binary, Embed, and Metadata. All the frameworks support users in transforming heterogeneous formats with few differences (a comparison is provided in Table 7.3).

Table 7.3. Comparison of the formats supported

	JSON	CSV	HTML	Bin.	XML	Text	Embed	Meta.	RDB	Spread.
RML	X	X	X		X	X			X	
SPARQL-Generate	X	X	X	X	X	X			X	
SPARQL Anything	X	X	X	X	X	X	X	X		X

Currently, SPARQL Anything and SPARQL-Generate covers the largest set of input formats. SPARQL-Generate but does not support embedding content (Embed) and extracting metadata from files (Metadata). Both features are not supported by RML, which doesn't support plain text as well. SPARQL Anything allows users to query spreadsheets, but it is not able to handle relational databases yet. However, relational tables can be transformed using an approach equivalent to CSV and spreadsheets tables. A dedicated triplifier is currently being developed. SPARQL Anything is the only tool supporting the extraction of metadata and the embedding of binary content.

Query. In terms of query support, while RML requires data to be transformed first and then uploaded to a SPARQL triple store, SPARQL Anything and SPARQL-Generate enable users to query resources directly.

Domain Independent. In RML and SPARQL Generate the re-engineering and re-modelling tasks are tightly coupled, since in both frameworks any information can be accessed only if the tool has been explicitly configured by the end-user, for example, with an XPath expression. In fact, the mapping task requires the user reflect on the subjects, predicates, and objects that will be generated from the non-RDF resource. Indeed, a SPARQL-aware user cannot interrogate the source formats without engaging in a re-modelling task.

Thanks to Facade-X, SPARQL Anything can access non-RDF resources in a domain independent way, allowing the user to focus only on the re-modelling task.

We now discuss requirements related to learning curve, usability, and potential for adoption.

Low learning demands. SPARQL Generate uses an extension to SPARQL 1.1 to transform source formats into RDF. RML provides an extension to the R2RML vocabulary in order to map source formats into RDF. Therefore, either a SPARQL extension or a new mapping language has to be learned to perform the translation. In the case of Facade-X, no new language has to be learned as data can be queried using existing SPARQL 1.1 constructs.

Low complexity. Complexity can be measured as the number of distinct items or variables that need to be combined with the query. In experiments, Facade-X is found to perform favorably in comparison to SPARQL Generate and RML.

Meaningful abstraction. Differently from RML and SPARQL-Generate, which require users to be knowledgeable of the source formats and their query languages (e.g., XPath, JSONPath etc.), Facade-X users can access a resource as if it was an RDF dataset, hence the complexity of the non-RDF languages is completely hidden to them. The cost for this solution is limited to the users which are required to explore the facade that is generated and tweak the configuration via the Facade-X IRI schema.

Explorability. With SPARQL Generate and RML, the user needs to commit to a particular mapping or transformation of the source data into RDF. However, the data representation required to carry out a knowledge intensive task often emerges from working with data and cannot be wholly specified in advance (this is a crucial requirement of our project SPICE). By distinguishing the processes of re-engineering and re-modelling, Facade-X enables the user to avoid prematurely committing to a mapping and rather focus on querying the data within SPARQL.

Workflow. All the technologies considered can in principle be integrated with a typical Semantic Web engineering workflow. However, while we cannot assume that Semantic Web experts have knowledge of RML, XPath, and SPARQL Generate, we can definitely expect knowledge of SPARQL.

Adaptable. All technologies provide a flexible set of methods for data manipulation, sparql.anything relying on plain SPARQL. We make the assumption that SPARQL itself is enough for manipulating variables, content types, and RDF structures. It is an interesting, open research question to investigate content manipulation patterns in the various languages and compare their ability to meet user requirements.

Finally, we discuss requirements of software engineering.

Extendable and Sustainable. We practically demonstrated how our approach can be implemented with existing SPARQL query processors with minimal development effort. Extending SPARQL Anything requires to write a component that transforms the data source into a Facade-X RDF. Facade-X is a metamodel which does not need to be encoded in the software but serves as a guidance for triplifying an open-ended set of formats. In contrast, extending SPARQL Generate and RML requires developing software components to handle the specific element of the source format, including exposing to users' specific functions for querying, filtering, traversing, and so on. In addition, our approach leads to a more sustainable codebase. To give evidence of this statement, we use the tool cloc⁴⁶ to count the lines of Java code required to implement the core module of SPARQL Generate in Apache Jena (without considering format-specific extensions⁴⁷) and the RML implementation in Java⁴⁸. SPARQL-Generate and RML require developing and maintaining 12280 and 7951 lines of Java code, respectively. We developed the prototype implementation of SPARQL Anything with 3842 lines of Java code, including all the currently supported transformers.

⁴⁶ cloc: <https://github.com/AlDanial/cloc> (accessed 15/12/2020)

⁴⁷ For SPARQL Generate, we only considered the code included in the submodule sparql-generate-jena.

⁴⁸ RMLMapper: <https://github.com/RMLio/rmlmapper-java>.

8 The SPICE Linked Data Hub v1.0

8.1 Concept

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 is provided in Figure 8.1.

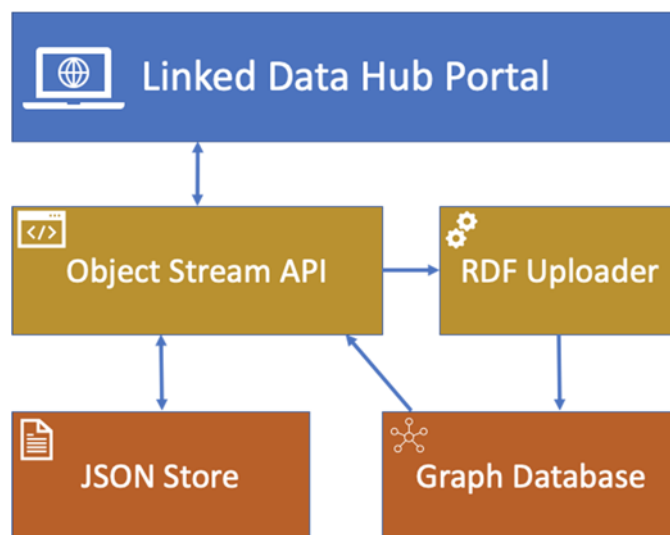


Figure 8.1: SPICE Linked Data Hub layout

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 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. These datasets are largely made up of social media activities within the SPICE network. Sitting behind the web portal and driving most of its functionality is an 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)
- JSON Schema management
- 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. JSON Schema support also enables dataset owners to enforce adherence to any number of data models. These can be custom data models created and stored locally with the SPICE LDH or referenced externally. 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.

8.2 Software components

The SPICE Linked Data Hub (LDH) is comprised of and based on the following components:

- **Linked Data Hub Portal**
<https://github.com/spice-h2020/linked-data-hub>
Description of LDH Portal
 Much of the LDH Portal's functionality resides in the following sub-modules:
 - **mkdf-core**
<https://github.com/mkdf/mkdf-core>
 Core functionality for the operation and layout of the Linked Data Hub.
 - **mkdf-datasets**
<https://github.com/mkdf/mkdf-datasets>
 Management of the Linked Data Hub dataset catalogue and dataset access permissions.
 - **mkdf-keys**
<https://github.com/mkdf/mkdf-keys>
 Creation and management of the dataset access keys that are required for use of the main data API.
 - **mkdf-topics**
<https://github.com/mkdf/mkdf-topics>
 Management of curated dataset collections within the Linked Data Hub.
 - **mkdf-stream**
<https://github.com/mkdf/mkdf-stream>
 The mkdf-stream module handles the Linked Data Hub's communications with the API Factory software (below).
 - **mkdf-sparql**
<https://github.com/mkdf/mkdf-sparql>
 This add-on module provides a web interface, within the Linked Data Hub Portal, for running SPARQL queries to explore datasets without the need for direct API calls.
 - **mkdf-file**
<https://github.com/mkdf/mkdf-file>
 A web interface for managing static files within a dataset, directly from the Linked Data Hub Portal without the need for API calls.
- **API** **Factory**
(v0.8.1)
<https://github.com/mkdf/api-factory>
 The API Factory software sits behind the Linked Data Hub Portal and provides the main REST-based web API for read, write, browse and management operations on SPICE datasets. Once datasets have been created and configured using the Linked Data Hub Portal, developers can then interact with them outside of the portal by accessing this data API directly. The API Factory is predominantly based around datasets made of unstructured JSON documents, although JSON schemas can optionally be applied and enforced on datasets. File handling features also allow static files of any format to be stored alongside the main JSON data store.

Swagger

interface

<https://api2.mksmart.org/>

The web API has been developed with a self-documenting web interface, describing all API functions and parameters and providing a quick interface for testing and development without the need for writing code. This is shown in Figure 8.2.

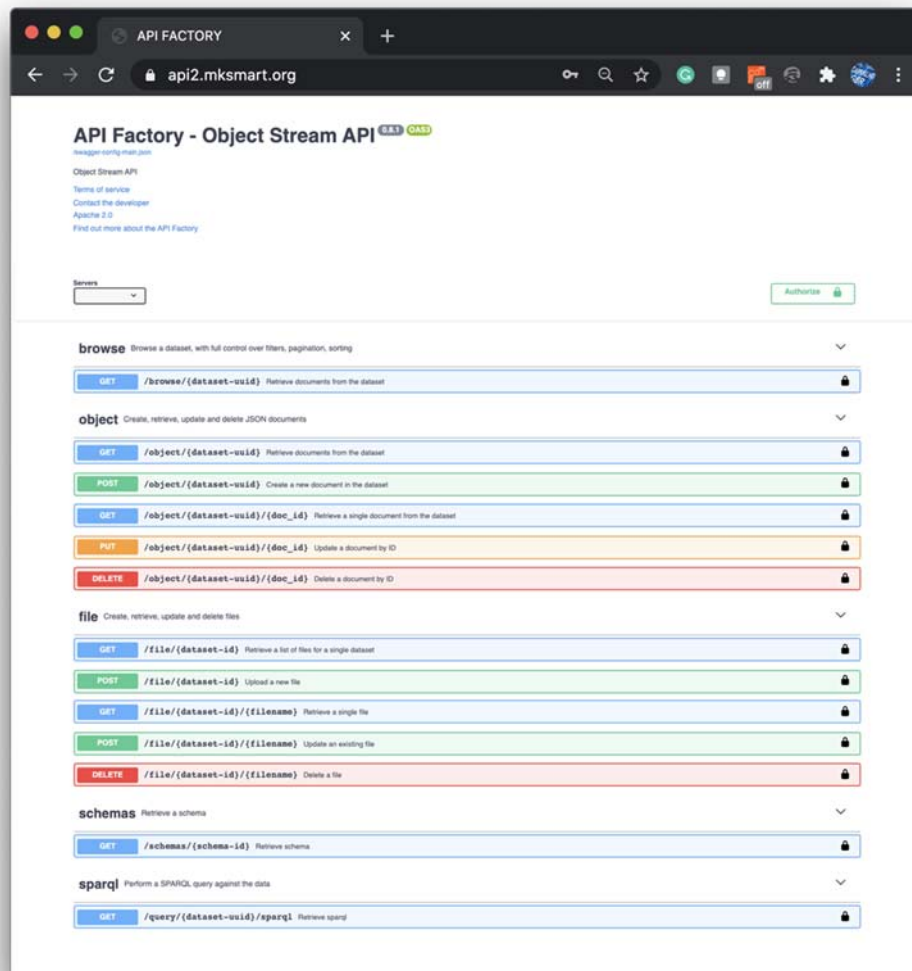


Figure 8.2: Web API – Swagger interface

The API Factory also makes of the following sub-modules:

- **api-factory-sparql**
<https://github.com/mkdf/api-factory-sparql>

The SPARQL add-on module provides the API with a read-only SPARQL endpoint for interacting with datasets that have been enabled via the *RDF Uploader* package (below) for RDF replication to a graph database.

• RDF

Uploader

<https://github.com/spice-h2020/rdf.uploader>

The RDF Uploader software component runs alongside the web API, regularly monitoring its activity log. All new JSON data is also converted to RDF for replication in a graph database. This replicated

RDF data is made available for SPARQL queries via both the web API's SPARQL endpoint and also the Linked Data Hub Portal's web-based SPARQL query tool.

- **MongoDB and Blazegraph – third party database storage**

In addition to software developed within the SPICE project, the SPICE LDH makes use of MongoDB and Blazegraph database software.

- MongoDB is document-based database, built to store JSON-like documents. Unlike a traditional database based on rows and columns of data, MongoDB offers maximum flexibility for storing rich data in a variety of formats, whilst also providing support for data models and schema validation when required.
- Blazegraph is an RDF-based graph database. For each JSON document that is submitted to MongoDB via the SPICE LDH's API, a graph is created within Blazegraph with the original JSON key:value pairs represented as RDF triples. This serves as the back-end storage mechanism that drives the SPICE LDH's SPARQL querying functions.

8.3 Web Portal

The SPICE LDH's web portal⁴⁹ provides a web-based catalogue and management tools for the datasets held within the SPICE Linked Data Hub (Figures 8.3-8.5). The portal homepage highlights a selection of featured datasets. These are manually selected by the site administrator in order to promote specific datasets.

⁴⁹ <https://spice.kmi.open.ac.uk/>

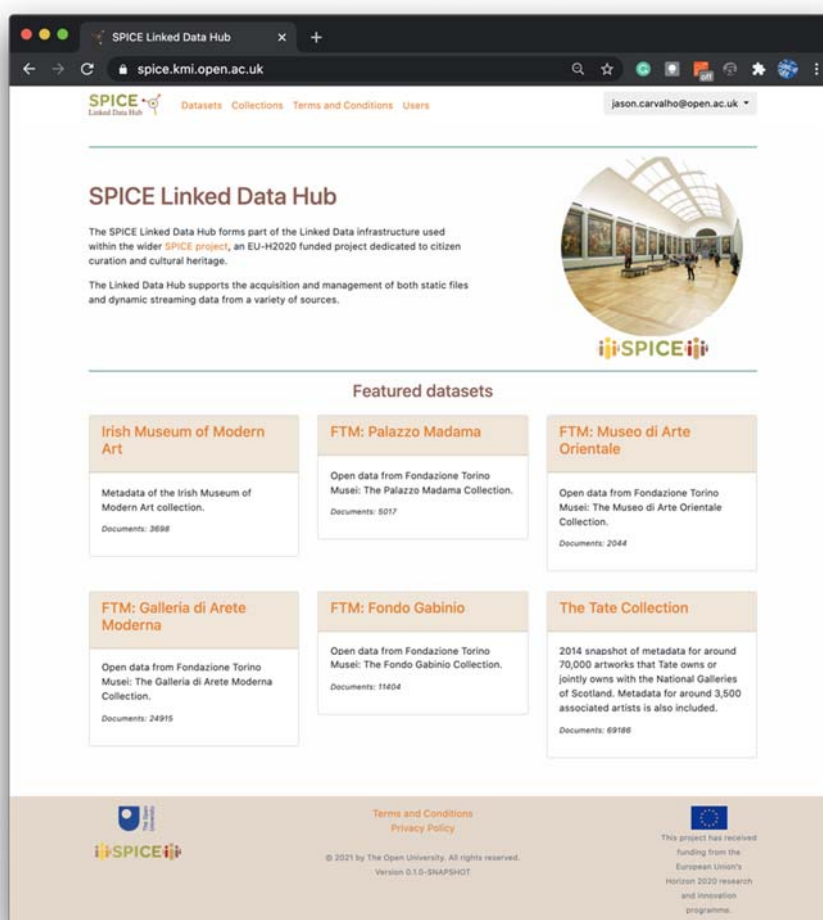


Figure 8.3: SPICE LDH - Homepage

The main method of dataset exploration with the web portal is via the dataset catalogue. All datasets are listed here, with a simple search filter provided. Datasets for which the current user is the owner are highlighted accordingly.

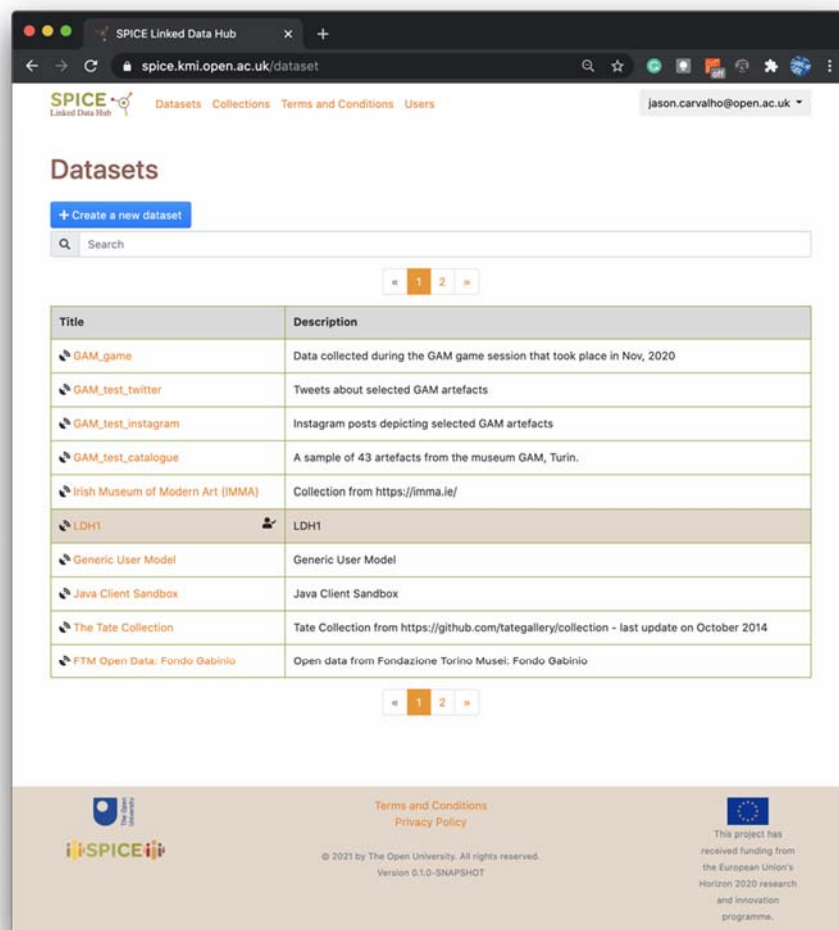


Figure 8.4: SPICE LDH - Dataset catalogue

The web portal also offers the ability to create custom collections of datasets. All users can create collections of datasets, including datasets that they do not own. This offers users an alternative method of browsing datasets in logically grouped clusters such as belonging to a particular project, event or region, being of a particular type (e.g., social media feeds or museum collection metadata).

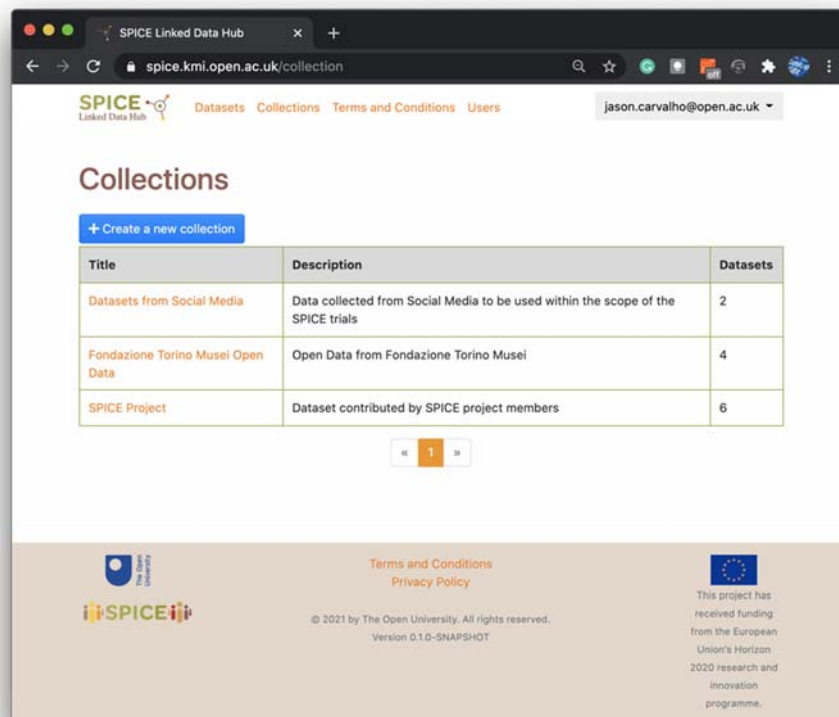


Figure 8.5: SPICE LDH – Dataset collections

Selecting a dataset within the LDH reveals an overview of the dataset, along with a series of tabs on the left-hand side for viewing and managing aspects of the dataset's metadata, depending on the user's access permissions (Figures 8.6-8.8). These include:

- *Overview*
Dataset title, description and unique ID.
- *Location*
Each dataset has an optional geotag which can be assigned. This facilitates the building of future interfaces that would enable browsing datasets by location on a map, for example.
- *Ownership and licensing*
Dataset provenance is set here, along with the ability to select any number of dataset licenses from a predefined list.
- *Permissions*
The SPICE LDH provides fine-grained access controls for each dataset. Dataset owners have control over who can view (visibility within the dataset list), read from (use the API to retrieve data from the dataset), write to and manage their datasets. Access for these operations can be granted for anonymous users, logged-in users and specific named users.
- *Collections*
Describes which collections this dataset is a member of.
- *Tags*
Tags can be assigned to datasets to assist with searching and browsing. When using the search

feature on the dataset catalogue, search strings are compared against the dataset title, description and any tags assigned to the dataset.

- **API**
This tab provides details for accessing the data via the web API. API URLs are provided here as well as the option to subscribe to datasets using an access key, if permissions allow.
- **SPARQL**
The web portal provides a GUI tool for running read-only SPARQL queries on datasets. A read access key is required on the dataset in order to use this feature. The query tool uses interface components from the YASGUI platforms; specifically, the Yasqe component for SPARQL query editing and the Yasr component for visualising results.

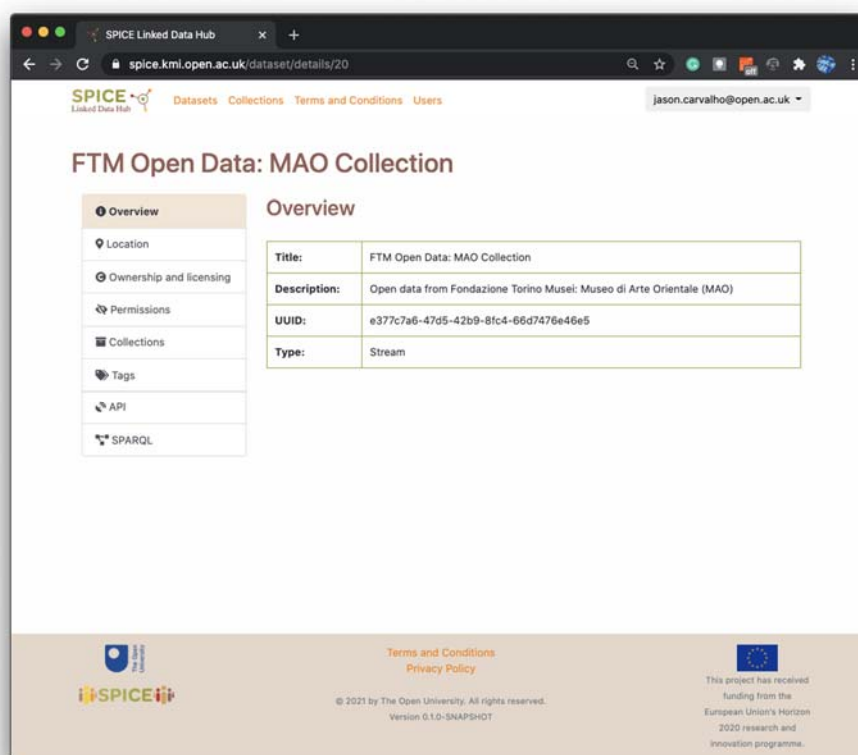
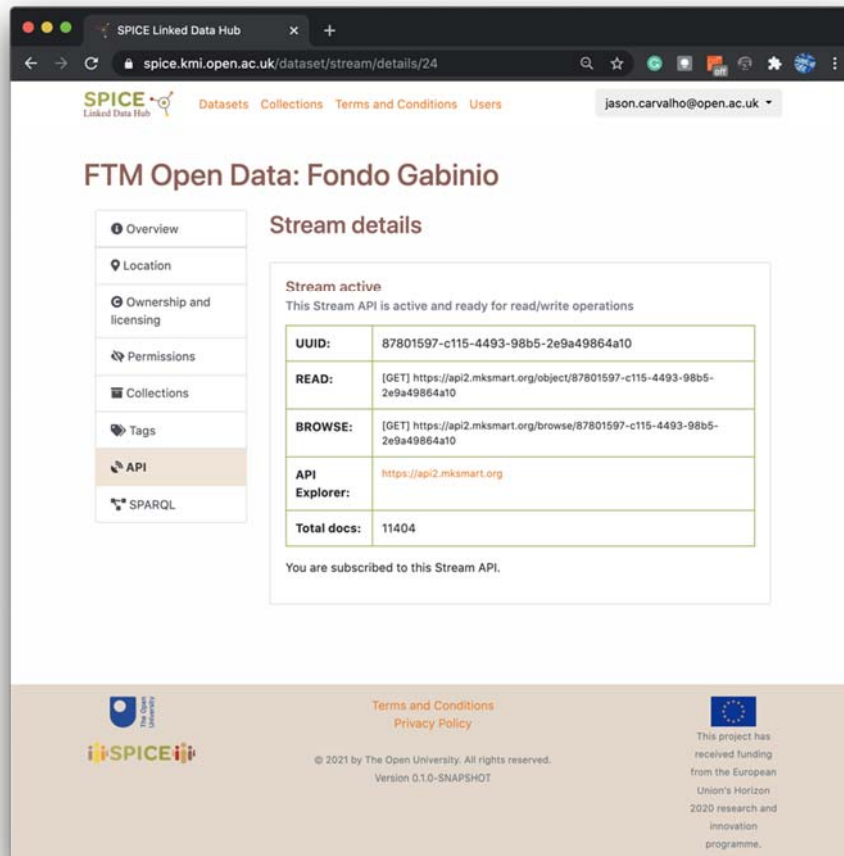


Figure 8.6: SPICE LDH – Dataset overview



The screenshot shows a web browser window with the URL `spice.kmi.open.ac.uk/dataset/stream/details/24`. The page title is "FTM Open Data: Fondo Gabinio". On the left, there is a sidebar with navigation links: Overview, Location, Ownership and licensing, Permissions, Collections, Tags, API (highlighted), and SPARQL. The main content area is titled "Stream details" and contains a section "Stream active" with the text "This Stream API is active and ready for read/write operations". Below this is a table with the following data:

UUID:	87801597-c115-4493-98b5-2e9a49864a10
READ:	[GET] https://api2.mksmart.org/object/87801597-c115-4493-98b5-2e9a49864a10
BROWSE:	[GET] https://api2.mksmart.org/browse/87801597-c115-4493-98b5-2e9a49864a10
API Explorer:	https://api2.mksmart.org
Total docs:	11404

Below the table, it states "You are subscribed to this Stream API." The footer of the page includes the SPICE logo, copyright information "© 2021 by The Open University. All rights reserved. Version 0.1.0-SNAPSHOT", and a European Union logo with text "This project has received funding from the European Union's Horizon 2020 research and innovation programme."

Figure 8.7: SPICE LDH – Dataset API access details

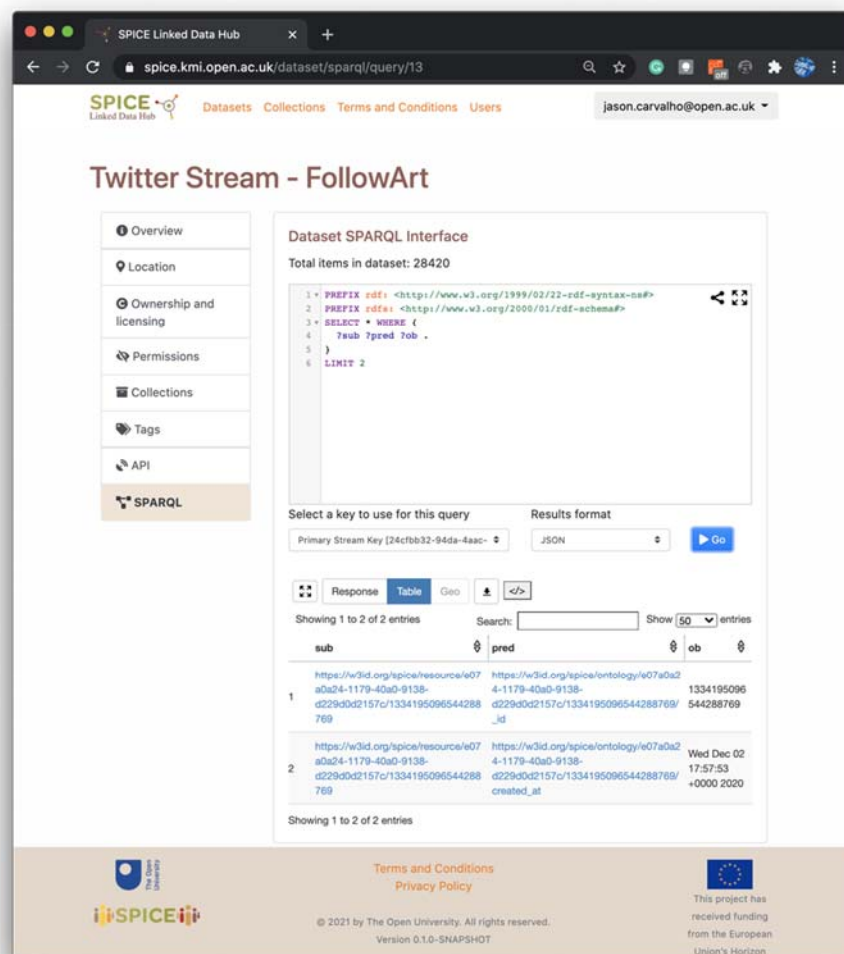


Figure 8.8: SPICE LDH – SPARQL query interface

A user guide detailing basic operations of both the web portal and web API is available in Annex A: *Spice Linked Data Hub User Guide*.

8.4 Web API

The web API is an instance of the API Factory software, developed as part of the SPICE project. The API exposes a selection of user operations for creating, managing and consuming data as well as management functions which are used by the web portal for creating datasets and managing permissions.

User operations - <https://api2.mksmart.org/>

/browse

A read-only API endpoint for retrieving data. The endpoint provides options for paging, sorting, filtering, field selection and complex database queries using MongoDB-style JSON queries.

GET /browse/{dataset-uuid} Retrieve documents from the dataset

Parameters

Name	Description
dataset-uuid * required string (path)	dataset uuid identifier
query string (query)	The filter query
sort string (query)	Optionally specify fields on which to sort the data. Sort fields should be specified as a comma separated list. Data will be sorted in ascending order. To specify a field to sort in descending order precede that field with a minus ("-")
fields string (query)	Optionally specify which fields to return. Fields should be specified as a comma separated list. Fields preceded with a minus ("-") will be excluded from the results. The "_id" field is always returned, unless explicitly excluded.
pagesize integer (query)	Optionally specify page size (defaults to a page size of 100)
page integer (query)	Optionally specify the page number of results to return (defaults to page 1)

Responses

Code	Description	Links
200	Success	No links
400	Bad request, malformed JSON	No links
500	Fatal error creating object	No links

Figure 8.9: Web API - **browse** endpoint

/object

The object endpoint is used for standard CRUD-style database operations; reading, writing, updating and deleting. The HTTP method used (GET/POST/UPDATE/DELETE) defines which function is called.

GET /object/{dataset-uuid} Retrieve documents from the dataset

POST /object/{dataset-uuid} Create a new document in the dataset

GET /object/{dataset-uuid}/{doc_id} Retrieve a single document from the dataset

PUT /object/{dataset-uuid}/{doc_id} Update a document by ID

DELETE /object/{dataset-uuid}/{doc_id} Delete a document by ID

Figure 8.10: Web API - **/object** endpoint

/schemas

This endpoint is used to retrieve a JSON Schema that has been stored within the SPICE LDH. Datasets that have been marked for schema validation will have new data checked against the corresponding schemas automatically, this API endpoint is provided merely for reference.

GET `/schemas/{schema-id}` Retrieve schema

Parameters

Name	Description
schema-id * required string (path)	Schema name

Try it out

Responses

Code	Description	Links
200	Success	No links
404	Schema not found	No links
500	Fatal error retrieving schema	No links

Figure 8.11: Web API - */schemas* endpoint

/sparql

The API's SPARQL endpoint. Read-only SPARQL queries can be run here against JSON data that has been replicated to the SPICE LDH's RDF graph database.

GET `/query/{dataset-uuid}/sparql` Retrieve sparql

Parameters

Name	Description
dataset-uuid * required string (path)	Dataset name
query * required string (query)	The SPARQL query

Try it out

Responses

Code	Description	Links
200	Query results	No links
400	Bad request or malformed SPARQL	No links
500	Fatal error	No links

Media type: application/sparql-results+json

Controls: Accept header

Example Value | Schema

```
{}
```

Figure 8.12: Web API - */sparql* endpoint

Management operations - <https://api2.mksmart.org/management>

The API Factory software exposes a set of management operations, used for creating and managing datasets, dataset permissions and dataset schemas. In normal use it is not necessary to use this directly. These management operations provide an interface between the front-end (web portal) and back-end (API Factory) parts of the SPICE LDH and provide the opportunity to develop multiple front-end interfaces that make use of the same data store. Administrators with the required access permissions can use these management operations directly if they wish, for specific admin, maintenance and development tasks.

datasets <small>Manage datasets</small>		Find out more: http://datahub.mksmart.org
GET	/datasets Retrieve all datasets	
POST	/datasets Create a new dataset	
GET	/datasets/{dataset-id} Retrieve single dataset details	
POST	/datasets/{dataset-id}/schemas/{schema-id} Create schema/dataset association	
DELETE	/datasets/{dataset-id}/schemas/{schema-id} Delete schema/dataset association	
permissions <small>Manage permissions</small>		
GET	/permissions Get all permissions	
GET	/permissions/{key} Get permissions for single key	
POST	/permissions/{key} Set/update permissions	
schemas <small>Manage schemas</small>		
GET	/schemas Retrieve all schemas	
POST	/schemas Create a new schema	
GET	/schemas/{schema-id} Retrieve full details for a single schema	
PUT	/schemas/{schema-id} Update an existing schema	
POST	/datasets/{dataset-id}/schemas/{schema-id} Create schema/dataset association	
DELETE	/datasets/{dataset-id}/schemas/{schema-id} Delete schema/dataset association	

Figure 8.13: Web API – Management operations

8.5 Data

JSON-based datasets within the Linked Data Hub include metadata collections of artworks from a number of museums and galleries, data models and ontologies and social media activities related to the SPICE network. Table

Table 8.14 List of datasets currently managed by the Linked Data Hub

Dataset	Description	Number of entries	Policies
---------	-------------	-------------------	----------

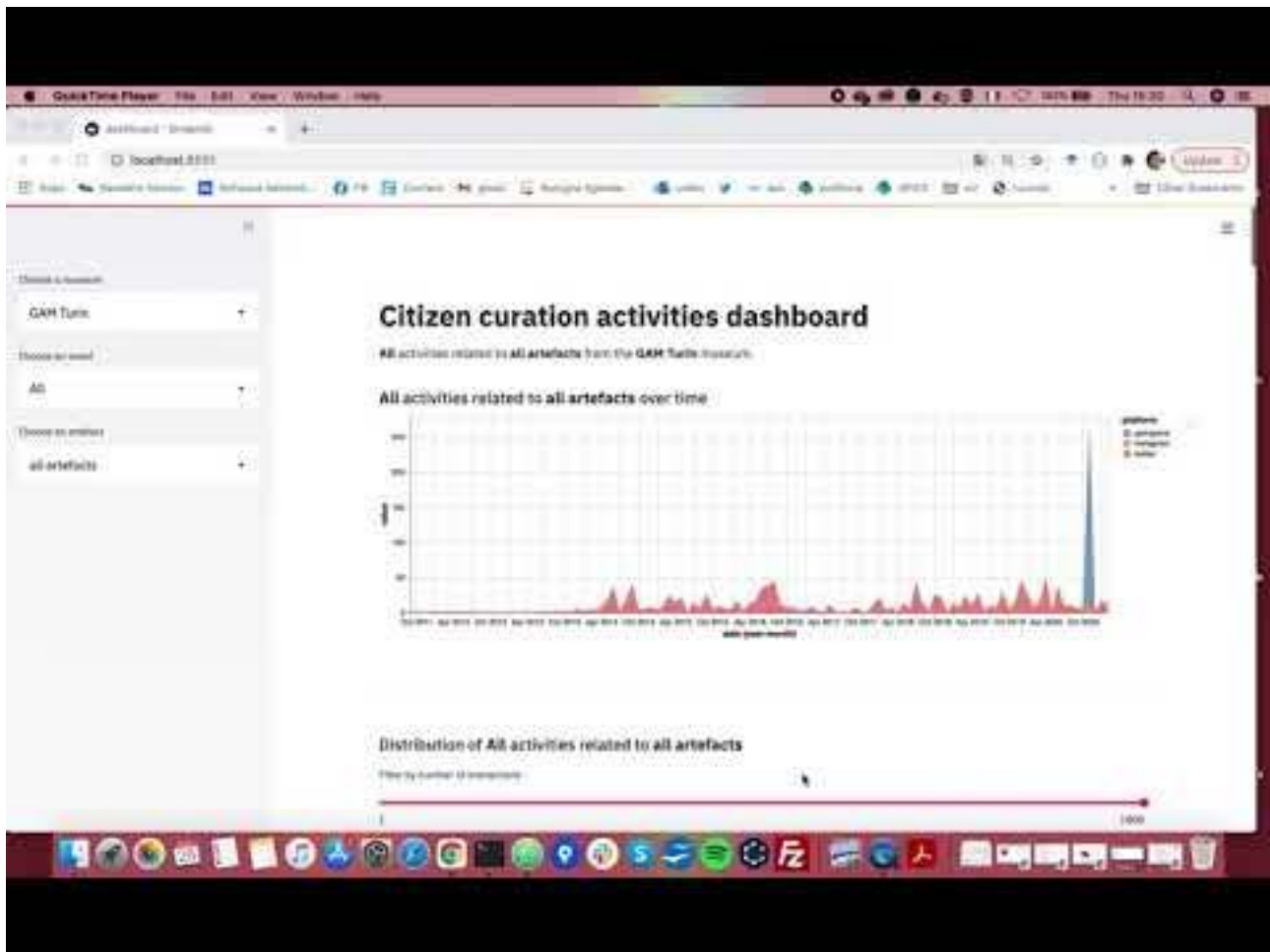
The Tate Collection	2014 snapshot of metadata for around 70,000 artworks that Tate owns or jointly owns with the National Galleries of Scotland. Metadata for around 3,500 associated artists is also included	69,186	CC0 Licence, Public domain
FTM Open Data: GAM Collection	Open data from Fondazione Torino Musei: The Galleria di Arete Moderna Collection	24,915	Open Data
FTM Open Data: MAO Collection	Open data from Fondazione Torino Musei: The Museo di Arte Orientale Collection	2,044	Open Data
FTM Open Data: Palazzo Madama Collection	Open data from Fondazione Torino Musei: The Palazzo Madama Collection	5,017	Open Data
FTM Open Data: Fondo Gabinio	Open data from Fondazione Torino Musei: The Fondo Gabinio Collection	11,404	Open Data
Twitter Stream - GAM Torino	Twitter search on 'GAM Torino'	223	Private to SPICE consortium
Twitter Stream - FollowArt	Twitter search on 'FollowArt'	28,410	Private to SPICE consortium
Irish Museum of Modern Art (IMMA)	Metadata of the Irish Museum of Modern Art collection	3,698	Private to SPICE consortium
GAM_game	Anonymised data collected during a GAMgame session	~300	Open Data
GAM_test_twitter	Twitter post on selected artefacts of the GAM gallery	~1000	Open Data
GAM_test_instagram	Instagram post on selected artefacts of the GAM gallery	80	Open Data
GAM_test_catalogue	Cataloguing data of selected artefacts of the GAM gallery	43	Open Data

9 Proof of concept: Social Media and citizen curation activities dashboard

In order to demonstrate feasibility, usefulness, and efficient reuse of data served by the Linked Data Layer, we defined the following use case.

In the scenario, a developer of the SPICE consortium wants to publish data collected from a museum catalogue, users' posts on social media platforms, and users' stories elicited during an engagement activity performed by the museum. The developer wants to create a proof-of-concept dashboard for social media and citizen curation activities analysis. However, the data must be accessible only to the partners of the consortium.

The use case also works as a harmonizer of SPICE partners' technical contributions with the aim to facilitate a future evaluation of the soundness of SPICE ecosystem⁵⁰. A video of the demo is available at: [SPICE citizen curation activities dashboard](#)⁵¹.



Data collection. The dashboard provides statistics on user-generated data relevant to selected artifacts from the *Galleria d'Arte Moderna (GAM)* collection. We created four datasets, namely: the GAM catalogue dataset, including descriptions and curators' stories on 43 artefacts; the GAM Twitter dataset and the GAM Instagram dataset, including posts addressing the artefacts; the GAMgame dataset, including users' stories and emotional responses with respect to selected GAM artefacts. Twitter and Instagram data were collected via Twitter Academic API⁵² and Instagram Basic Display API⁵³ respectively, and posts were associated to artefacts by means of Pastec⁵⁴, a popular image matching software. GAMgame data were collected by the GAM museum and the University of Turin in November 2020 via a dedicated web application.

⁵⁰ The source code of both GAM data pipeline and the dashboard is available at <https://github.com/spice-h2020/linked-data-hub-notebooks/tree/master/demonstrator>.

⁵¹ <https://youtu.be/GIDD7VkZyA8>

⁵² <https://developer.twitter.com/en/solutions/academic-research>

⁵³ <https://developers.facebook.com/docs/instagram-basic-display-api/>

⁵⁴ <https://github.com/magwyz/pastec>

Data enrichment and annotation. Both curators and users' data are enriched with annotations on emotional responses, thanks to an annotation service provided by CELI (WP3). An auxiliary dataset with the description of the script used to perform the GAMgame activity is also provided. It addresses requirements highlighted by the Scripting working group (including members of WP3, WP4, WP6). All datasets are served as JSON-LD documents according to ArCO⁵⁵ (Cataloguing data), Schema.org (social media posts), and the SON ontologies⁵⁶ (users' stories, emotions, scripting), as defined in D6.2 "Initial Ontology Network Specification".

Data publication. Datasets were uploaded on the Linked Data Hub by following HOW-TO guidelines (see previous section). A developer of the SPICE consortium uploaded data on behalf of the GAM museum, who also associated to the datasets the proper attribution license (CC-BY), claimed GAM ownership, and set the permissions for reuse only to SPICE partners only. A Web API has been created for each dataset, to which the dashboard web application can send requests.

The dashboard. The dashboard provides curators and mediators with analytics on user engagement with museum exhibits as retrieved from Social Media platforms and past citizen curation activities. Potentially, it can leverage data belonging to different museums and pilots in SPICE. The objective of the application is to give museums a feedback on how their assets are perceived on the web and how their data are used in engagement activities. Currently, the development of the dashboard is guided by the following questions:

- How is an artefact perceived across Social Media Platforms and citizen curation activities? What are the emotional reactions and how do they differ across different types of activities? In which activity there is more emotional response?
- What is the gap between curators/artists' view and users' view? Do elicited emotions and represented contents conform to the museum point of view?

In particular, the dashboard supports curators in the following activities:

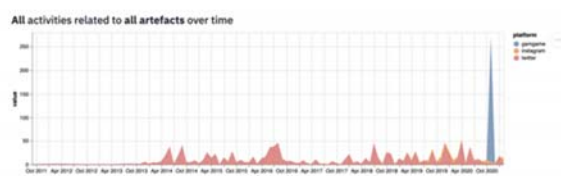
1. **Monitor** activities involving museum assets on the web and in dedicated citizen curation activities.
2. **Review** user-generated contents, displayed in aggregated views and individually for close reading.
3. **Survey and select** citizen curated contents to be included in their catalogues.
4. **Evaluate** past activities, for instance by measuring the emotional response with respect to an artifact, or by comparing curators' emotional responses to users' responses (again both in aggregated views or by close reading users' stories).
5. **Design** future activities based on prior activities results, users' emotional responses, level of engagement on different platforms and with different activities scripts.

In Figure 9.1 we show some of the current functionalities of the dashboard built on the GAM datasets.

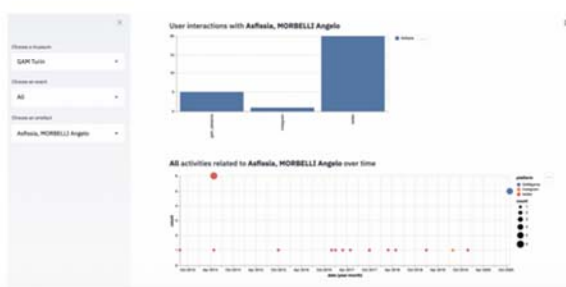
⁵⁵ <http://wit.istc.cnr.it/arco>

⁵⁶ <http://github.com/spice-h2020/son>

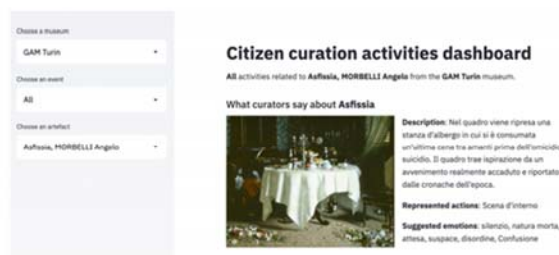
Monitor activities going on with respect to a museum asset



Monitor activities going on with respect to a museum exhibit



Compare users' opinions with curators' descriptions



Review users' interactions related to artefacts

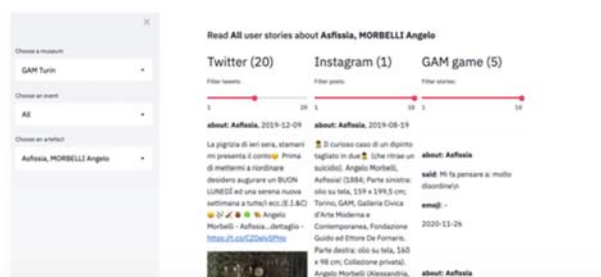


Figure 9.1. Screenshots of the SPICE dashboard and main functionalities

In future works we would like to refine the tool and to address new research questions, such as:

- Are there any relevant relations between artefacts? Which qualities of artefacts generally convey a certain message? E.g., women sculpture generally obtain positive emotional response.
- What is the best way to engage with users if a new artefact is introduced? Given reactions to similar artefacts, can we predict or recommend a best way to propose the artefact? E.g., what are the candidate scripts to be used?

Crucially, we aim at involving curators of the SPICE consortium in contributing the design of the system, by following the principles developed in WP2. From the technical standpoint, we plan to integrate LD intelligence technologies such as the ones mentioned in Section 6 and developed in WP3 for recommendation purposes, so as to highlight different opinions on artefacts and support the detection of communities based on different interpretations. Finally, the dashboard is also the playground for defining some of the future challenges of the work package, particularly, in relation to tasks 4.3 “Linking and discovering digital assets” (M13–M24) and 4.4 “Provenance and process analysis”.

10 Conclusions and future work

This report presented the deliverable of WP4 of the SPICE project, focusing on the SPICE Linked Data server technology. After a journey on the state of art technologies for Linked Data content management and insights on the status of data management in SPICE museums, we reported on the design of the Linked Data Layer, a collection of components and protocols for data communication and exchange across the SPICE network, supporting a privacy-aware data sharing platform and innovative technologies supporting the generation of Linked Data from legacy resources. Crucially, we are already applying the infrastructure to develop a dashboard for data-drive decision making, targeting museum curators as primary users, and social media content as first-class citizen.

The survey on the use of software systems for cataloguing and managing collections by the partner museums (Section 2) reveals a number of tools and roles, and exposes the use of platforms which tend to be monolithic, i.e., scarcely interoperable and often lacking linked data functions. In parallel with this situation, the analysis of museum organization charts shows that roles differ significantly from museum to museum; moreover, diversity is still underrepresented, with specialized roles for inclusion still lacking in most museums. This

overall picture poses two main problems for data management in SPICE: on the one hand, the data generated by the citizen curation activities at a given museum are not interoperable with the specific software in use at the museum, and the activities themselves cannot be planned and managed by using the standard museum management tools. The plurality of platforms in use, often embedding proprietary solutions for data management and sharing, adds a further element of complexity to these two types of integration. The next steps will focus on developing an application layer on top of the infrastructure to support the development of front-facing components of the SPICE pilots in WP7 and provide the backbone for interoperating with museums' own management systems. This will include developing a Scripting API, to support the design of reusable SPICE interfaces (WP5 and T6.5). In addition, we will expand the capabilities of the Linked Data Hub to support a privacy and policy layer (T4.2), targeted to the management of copyright and policy information associated to the reuse of museums' assets, considering issues such as supporting workflows for copyright management and brokering of terms and conditions. The definition of roles involved in data management proposed is a step towards integration, since it provides a clear and unified way to map data management roles to the functions of the software platforms adopted by museums, and to assign them to specific professional roles in the citizen curation workflows.

11 Research outputs

Submitted or published research relevant to the deliverable:

- Enrico Daga, Luigi Asprino, Rossana Damiano, Marilena Daquino, Belen Diaz Agudo, Aldo Gangemi, Tsvi Kuflik, Antonio Lieto, Mark Maguire, Anna Maria Marras, Delfina Martinez Pandiani, Paul Mulholland, Silvio Peroni, Sofia Pescarin, and Alan Wecker. **Integrating citizen experiences in cultural heritage archives: requirements, state of the art, and challenges** ACM Journal on Computing and Cultural Heritage, Special Issue on Computational Archival Science. Submitted December 2020, Accepted in April 2021 with minor revisions.
- Enrico Daga, Luigi Asprino, Paul Mulholland, and Aldo Gangemi. **Facade-X: an opinionated approach to SPARQL Anything**. Submitted on April 2021 to SEM-EU conference, currently under review.
- Luigi Asprino, Enrico Daga, Paul Mulholland, and Aldo Gangemi. **Minimalistic re-engineering for streamlining knowledge graph construction**. Submitted to the International Semantic Web Conference on April 2021, currently under review.
- Enrico Daga, Luigi Asprino, Paul Mulholland, and Aldo Gangemi. **Knowledge Graph Construction with SPARQL Anything** Submitted to the International Semantic Web Conference (Resources Track) on April 2021, currently under review

Software:

SPICE Linked Data Hub Enrico Daga, Jason Carvalho, Luigi Asprino. (2021, April 28). spice-h2020/linked-data-hub: v0.1.0 (Version v0.1.0). Zenodo. <http://doi.org/10.5281/zenodo.4724833>

SPARQL Anything Luigi Asprino and Enrico Daga. (2021, April 28). SPARQL-Anything/sparql.anything: (Version v0.1.1). Zenodo. <http://doi.org/10.5281/zenodo.4724587>

RDF Uploader Luigi Asprino and Enrico Daga. (2021, April 28). spice-h2020/rdf.uploader: v0.0.4 (Version 0.0.4). Zenodo. <http://doi.org/10.5281/zenodo.4724839>

Annexes

A. SPICE Linked Data Hub User Guide

Here follows a brief overview of getting started with the SPICE Linked Data Hub.

Request an account and login

Request an account on the LDH via email to enrico.daga@open.ac.uk or jason.carvalho@open.ac.uk

Access the LDH at <https://spice.kmi.open.ac.uk/> and login.

Click your email on the top right corner and select **My account**.

My account: Overview

You'll be redirected to your profile page. In the left sidebar, click on **My Keys**.

Create a key to be used by your applications

My account: Keys

Click on **Create a new key**

My account: New key

Assign a **name** and a **description** to the key (mandatory). You may wish to create a selection of keys for reading and/or writing to a variety of datasets, so it's important to create meaningful descriptions for these keys to help you manage them. Note that one key can be used to access several datasets, if you wish. Similarly, one dataset can be accessed by a variety of keys.

Create a dataset

My account: Datasets

In the left sidebar, click on **My datasets**. In the new page, click on **Create a new dataset**.

Add dataset

Insert the **name** and the **description** of the dataset.

Be sure the button '**stream**' is checked (this will not be needed in future releases as stream and file dataset types are to be merged).

Set dataset permissions

*You can choose how you want your dataset to be visible and accessible to other users. From the dataset page, click on **Permissions**, review the content of this section, and customise as needed.*

Activate the dataset API

From the dataset page, click on **Stream API**.

Select the key (previously created) that you want to use for writing/reading the dataset.

Be aware that you can use the same API key to access several datasets.

Take a note of the UUID identifying your dataset, and the URL of the Web API call to read your dataset.

Use the API to write/read content

SPICE Linked Data Hub - Data Stream API 0.2.0 0.1.0
[https://api2.mksmart.org/](#)
 Data Stream API
 Terms of service
 Contact the developer
 Apache 2.0
 Find out more about the API Factory

browse	Browse a dataset, with full control over filters, pagination, sorting	>
object	Retrieves, push, update and delete documents	>
schemas	Retrieve a schema	>
sparql	Perform a SPARQL query against the data	>

The web API is reachable at
<https://api2.mksmart.org/>

The interface requires authentication with an appropriate key (supplied as both the username and password) and shows a list of API actions, including:

/browse actions are for browsing content

/object action for creating, updating, and deleting JSON documents

/sparql action is for querying the dataset with SPARQL.

Upload your data

To upload your JSON document you need:

1. The API URL
2. An identifier to be associated to a new document (docId) to be created in your dataset (defined by you the first time you upload a doc)
3. The UUID of the dataset (dataset)
4. A valid JSON or JSON-LD (payload)
5. The API key associated to the dataset (key)

The following is an example Python script for uploading new data:

```
import json, requests

def upload(api, docId, dataset, payload, key):
    payload['_id'] = docId
    try:
        r = requests.put(api+'/'+'object/'+' '+ dataset+'/'+' '+docId,
                        json=payload,
                        auth=(key,key))
        if r.status_code == 200:
            print(r.status_code)
        else:
            print(r.status_code, r.reason, r.content)
    except Exception as e:
        print(e)
```

```
API = "https://api2.mksmart.org"
my_dataset_ID = "abcdefg"
authKey = "123456"
```

```
with open("my_data.json") as jfile:
    data = json.load(jfile)
    upload(api=API,
```

```
docId="my_catalogue",  
dataset=my_dataset_ID,  
key=authKey, payload=data)
```

References

- [Agostino et al 2020] Agostino D., Arnaboldi M., Lampis A., Italian state museums during the COVID-19 crisis: from onsite closure to online openness, *Museum Management and Curatorship*, pp.1-11
- [Batrinca and Treleaven, 2015] Batrinca, B. and Treleaven, P.C., 2015. Social media analytics: a survey of techniques, tools and platforms. *Ai & Society*, 30(1), pp.89-116.
- [Daga et al, 2015] Daga, Enrico, Luca Panziera, and Carlos Pedrinaci. "A BASILar approach for building web APIs on top of SPARQL endpoints." In *CEUR Workshop Proceedings*, vol. 1359, pp. 22-32. 2015.
- [Daga et al, 2016] Daga, Enrico, Mathieu d'Aquin, Alessandro Adamou, and Stuart Brown. "The open university linked data—data. open. ac. uk." *Semantic Web* 7, no. 2 (2016): 183-191.
- [Daga et al, 2021] Daga, Enrico and Albert Meroño-Peñuela and Enrico Motta. "Sequential Linked Data: the State of Affairs". *Semantic Web Journal*, IOS Press, 2021
- [Dimou et al, 2014] Dimou, Anastasia, Miel Vander Sande, Pieter Colpaert, Ruben Verborgh, Erik Mannens, and Rik Van de Walle. "RML: a generic language for integrated RDF mappings of heterogeneous data." In *Ldow*. 2014.
- [Freire et al, 2017] Freire, Nuno, Glen Robson, Antoine Isaac, Hugo Manguinhas, and John B. Howard. "Web Technologies: A Survey of Their Applicability to Metadata Aggregation in Cultural Heritage." In *ELPUB*, pp. 235-244. 2017.
- [Guha et al, 2016] Guha, Ramanathan V., Dan Brickley, and Steve Macbeth. "Schema. org: evolution of structured data on the web." *Communications of the ACM* 59, no. 2 (2016): 44-51.
- [Halford and Andrews 2004] Halford, G.S., Andrews, G.:The development of deductive reasoning: How important is complexity? *Thinking & Reasoning* 10(2), 123–145 (2004)
- [Warren et al. 2015] Warren, P., Mulholland, P., Collins, T., Motta, E.: Making sense of description logics. In: *Proceedings of the 11th International Conference on Semantic Systems*. pp. 49–56 (2015)
- [Hardesty, 2014] Hardesty, Juliet L. "Exhibiting library collections online: Omeka in context." *New Library World* (2014).
- [Haslhofer et al, 2011] Haslhofer, Bernhard, and Antoine Isaac. "data. europeana. eu: The europeana linked open data pilot." In *International Conference on Dublin Core and Metadata Applications*, pp. 94-104. 2011.
- [Haslhofer et al, 2013] Haslhofer, Bernhard, Simeon Warner, Carl Lagoze, Martin Klein, Robert Sanderson, Michael L. Nelson, and Herbert Van de Sompel. "ResourceSync: leveraging sitemaps for resource synchronization." In *Proceedings of the 22nd International Conference on World Wide Web*, pp. 11-14. 2013.
- [Ioannides and Davies, 2018] Ioannides, Marinos, and Robert Davies. "ViMM-Virtual Multimodal Museum: a manifesto and roadmap for Europe's digital cultural heritage." In *2018 International Conference on Intelligent Systems (IS)*, pp. 343-350. IEEE, 2018.
- [Jones and Hardt, 2012] Jones, Michael, and Dick Hardt. The oauth 2.0 authorization framework: Bearer token usage. RFC 6750, October, 2012.
- [Kyzirakos et al, 2014] Kyzirakos, Kostis, Ioannis Vlachopoulos, Dimitrianos Savva, Stefan Manegold, and Manolis Koubarakis. "GeoTriples: a Tool for Publishing Geospatial Data as RDF Graphs Using R2RML Mappings." In *TC/SSN@ ISWC*, pp. 33-44. 2014.
- [Lee et al, 2019] Lee, Jin Woo, Yikyung Kim, and Soo Hee Lee. "Digital Museum and User Experience: The Case of Google Art & Culture." In *International Symposium on Electronic Art. International Symposium on Electronic Art*. 2019.
- [Lefrançois et al, 2017] Lefrançois, Maxime, Antoine Zimmermann, and Noorani Bakerally. "A SPARQL extension for generating RDF from heterogeneous formats." In *European Semantic Web Conference*, pp. 35-50. Springer, Cham, 2017.

- [Li, 2020] Li, Jasmine. "Omeka Classic vs. Omeka. net." *Emerging Library & Information Perspectives* 3, no. 1 (2020): 232-236.
- [Markusen and Gadwa, 2010] Markusen, Ann, and Anne Gadwa. "Arts and culture in urban or regional planning: A review and research agenda." *Journal of planning education and research* 29, no. 3 (2010): 379-391.
- [Maron and Feinberg, 2018] Maron, Deborah, and Melanie Feinberg. "What does it mean to adopt a metadata standard? A case study of Omeka and the Dublin Core." *Journal of Documentation* (2018).
- [McKenna et al, 2018] McKenna, Lucy, Christophe Debruyne, and Declan O'Sullivan. "Understanding the position of information professionals with regards to linked data: a survey of libraries, archives and museums." In *Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries*, pp. 7-16. 2018.
- [Newell, 1982] Newell, Allen. "The knowledge level". *Artificial intelligence* 18, no. 1 (1982): 87-127.
- [Oldman and Tanase, 2018] Oldman, Dominic, and Diana Tanase. "Reshaping the Knowledge Graph by connecting researchers, data and practices in ResearchSpace." In *International Semantic Web Conference*, pp. 325-340. Springer, Cham, 2018.
- [Payette and Lagoze, 1998] Payette, Sandra, and Carl Lagoze. "Flexible and extensible digital object and repository architecture (FEDORA)." In *International Conference on Theory and Practice of Digital Libraries*, pp. 41-59. Springer, Berlin, Heidelberg, 1998.
- [Purday, 2012] Purday, Jonathan. "Europeana: Digital access to Europe's cultural heritage." *Alexandria* 23, no. 2 (2012): 1-13.
- [Rodriguez-Muro et al, 2015] Rodriguez-Muro, Mariano, and Martin Rezk. "Efficient SPARQL-to-SQL with R2RML mappings." *Journal of Web Semantics* 33 (2015): 141-169.
- [Slepicka et al, 2015] Slepicka, Jason, Chengye Yin, Pedro A. Szekely, and Craig A. Knoblock. "KR2RML: An Alternative Interpretation of R2RML for Heterogenous Sources." In *Cold*. 2015.
- [Smith et al, 2003] Smith, MacKenzie, Mary Barton, Mick Bass, Margret Branschofsky, Greg McClellan, Dave Stuve, Robert Tansley, and Julie Harford Walker. "DSpace: An open source dynamic digital repository." (2003).
- [Vayanou et.al, 2020] Vayanou, Katifori, Chrysanthi, Antoniou. Cultural Heritage and Social Experiences in the Times of COVID 19, *AVI 2CH* 2020, September 29, Island of Ischia, Italy
- [Vavliakis et al, 2012] Vavliakis, Konstantinos N., Georgios Th Karagiannis, and Pericles A. Mitkas. "Semantic Web in cultural heritage after 2020." In *Proceedings of the 11th International Semantic Web Conference (ISWC)*, Boston, MA, USA, pp. 11-15. 2012.
- [Wahyuningtyas, 2017] Wahyuningtyas, Ratri. "ELIMINATING BOUNDARIES IN LEARNING CULTURE THROUGH TECHNOLOGY: A REVIEW OF GOOGLE ARTS AND CULTURE." In *The 10th International Conference*, p. 179. 2017.
- [Wu, 2016] Wu, Annie, Santi Thompson, Rachel Vacek, Sean Watkins, and Andrew Weidner. "Hitting the road towards a greater digital destination: Evaluating and testing DAMS at the University of Houston Libraries." (2016)
- [Xiao et al, 2018] Xiao, Guohui, Diego Calvanese, Roman Kontchakov, Domenico Lembo, Antonella Poggi, Riccardo Rosati, and Michael Zakharyashev. "Ontology-based data access: A survey." *International Joint Conferences on Artificial Intelligence*, 2018.

Reports:

[ICOM International, 2020] *Museums, museum professionals and COVID-19*

<https://icom.museum/wp-content/uploads/2020/05/Report-Museums-and-COVID-19.pdf>

[ICOM International, 2020] *Museums, museum professionals and COVID-19: follow-up survey*

https://icom.museum/wp-content/uploads/2020/11/FINAL-EN_Follow-up-survey.pdf

[NEMO, 2020] *Survey on the impact of the COVID-19 situation on museums in Europe*

https://www.nemo.org/fileadmin/Dateien/public/NEMO_documents/NEMO_Corona_Survey_Results_6_4_20.pdf

[NEMO, 2021] *Follow-up Survey on the impact of the COVID-19 situation on museums in Europe*

https://www.nemo.org/fileadmin/Dateien/public/NEMO_documents/NEMO_COVID19_FollowUpReport_11.1.2021.pdf

[UNESCO, 2020] *Museums around the world in the face of COVID-19*
<https://unesdoc.unesco.org/ark:/48223/pf0000373530>

[We are social, 2020] *Digital 2020*

<https://wearesocial.com/it/digital-2020-italia>