SHERPA

Rural Science-Society-Policy Interfaces

D4.2 SHERPA Online Repository Design and Technical Specifications

31 March 2020

SHERPA receives funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 862448.

SHERPA Online Repository Design and Technical Specifications

Project name	SHERPA: Sustainable Hub to Engage into Rural Policies with Actors
Project ID	862448
H2020 Type of funding scheme	CSA Coordination and Support Action
H2020 Call ID & Topic	RUR-01-2018-2019–D / Rural society-science-policy hub
Website	www.rural-interfaces.eu
Document Type	Deliverable
File Name	D4.2 SHERPA online repository design and technical specifications
Status	Final
Dissemination level	Internal
Authors	Hercules Panoutsopoulos (Agricultural University of Athens), Borja Espejo Garcia (Agricultural University of Athens), Olivier Chartier (Ecorys), Spyros Fountas (Agricultural University of Athens)
Work Package Leader	AUA
Project Coordinator	ECORYS

Disclaimer: The content of this document does not reflect the official opinion of the European Union. Responsibility for the information and views expressed therein lies entirely with the author(s).

Executive summary

The SHERPA online repository system is a core part of the technological infrastructure that will be developed and delivered in SHERPA. The aim of the system is to provide storage capacities for the data that is going to be produced and made available in the context of the SHERPA methodology for identification, selection and evaluation of past and ongoing research projects, as well as facilitate access to this data by stakeholders or any interested party. To this end, there is need for a rigorous system design that will lead to the development of a robust system able to meet end-user needs efficiently. It should also be stressed that there needs to be a smooth 'communication' between the SHERPA repository and the web scraping and text mining tools that are going to be deployed for the extraction and retrieval of content from various sources. The description of the system's design takes place by:

- providing details of the data that needs to be stored and made available to end-users;
- defining end-user profiles and needs;
- providing details about the employed data model and system's architecture; and
- illustrating how the targeted end-user types, and their needs, are aligned to system functionalities.

Data that will be stored in the SHERPA online repository is made available by each phase of the methodology that is used for taking stock of past and ongoing research projects. To this end, a detailed description of the data produced takes place together with references about the information that is needed for an efficient data annotation. This information forms the basis on which a list of required metadata is proposed and presented in the end of Section 2.

End-user types that can benefit and, thus should be interested in making use of the SHERPA repository, are defined and described in Section 3. After an introduction to the end-user categories that are considered and the associated with them types of end-users, the definitions of end-user type profiles are provided. Then, a description of the needs of users is made available together with references to the user categories that have each need. As a result, an alignment of end-user categories and types with system functionalities is provided in Section 5.

As far as the technical aspects of the system's design are concerned, a detailed description of the architecture of the SHERPA online repository takes place in Section 5 after having introduced the technologies considered in Section 4. The system's structure and modules are presented with the help of a tier-based architecture. A core part of the system's design and technical specifications is that of the employed data model. The SHERPA data model helps to understand what the core data types are, as well as the relationships between them. In addition to that, details are provided with regard to the way in which the SHERPA repository system is going to handle and execute queries for information retrieval. Finally, the system's functionalities are made explicit and an alignment of end-user categories and types with the functionalities provided takes place.

A critical aspect of the system's design description is that of data security mechanisms that will be deployed. The data security mechanisms do not only relate to mechanisms for detecting malicious network activity, but also to the allocation of specific access rights to the users.

Finally, issues to be taken into consideration for the needs of developing a sustainability plan for the SHERPA repository system are highlighted in the last section of the document. These issues are very significant when trying to identify potential solutions for keeping the technological infrastructure running and the surrounding community active after the project's completion. Some concluding remarks and thoughts are provided in the Discussion and conclusions section of the document.

Table of contents

List of figures

List of tables

1. Introduction

Making the outcomes of research conducted in the context of SHERPA available to stakeholders involved in the process and members of the wider community is of increased importance for the advancement of project-related activities and the realisation of the work undertaken. Outcomes of SHERPA-related research are going to be made available via the SHERPA online repository, which is a core part of the technological infrastructure that will be developed and delivered in the project. However, the SHERPA repository is more than a collection of digital objects. Provision of quality content and information in an efficient way requires the existence of a robust system developed with the help of a rigorous design. Such a design should offer descriptions of data structure, definitions of user type profiles and needs, as well as present how user needs align with the system functionalities. Relational Database Management Systems (RDBMSs) have been used for many years as tools that facilitate access to well-structured data by allowing for specific access rights. Recently, the generation and availability of large volumes of data at an increasing pace has led to a shift in the needs for data storage and management. As a result, the NoSQL technology paradigm has emerged. NoSQL data store systems have gained momentum in the last years due to their capacity to provide flexibility and increased performance.

All the above shape the context for the design of the SHERPA online repository, which is the aim of Deliverable 4.2. The dimensions on which the design of the SHERPA online repository system is presented are:

- description of data stored into the SHERPA online repository;
- definition of profiles of targeted end-user types and needs;
- presentation of the architecture of the SHERPA online repository system; and
- description of system functionalities and alignment with end-user types and needs.

In addition to the above, details about data security mechanisms, a critical aspect of the SHERPA repository system deployment, are provided. Finally, the description of the design concludes with the presentation and analysis of a number of issues relating to the sustainability of the system after the project ends.

2. Data stored into the SHERPA online repository

The SHERPA methodology for taking stock of past and ongoing research in rural topics, documented in Deliverable 4.1 ('**Framework for identification, selection and evaluation of past and ongoing research projects**'), consists of **four phases** each of which provides the data needed for storage in the SHERPA online repository. These four phases relate to:

- **rural topics**

Research in SHERPA focuses on topics related to rural areas. Compilation of the list of rural topics needed for driving research in the context of SHERPA is the outcome of the first phase of the methodology (namely, the '**Identification of topics**' phase).

- **projects**

Each rural topic has been addressed by research projects implemented in the context of Project Frameworks (e.g. FP6, FP7, H2020, LIFE, etc.). Outputs of these projects contain results relevant to the investigated rural topics. Identification of projects that relate to the investigated rural topics and extraction of information about them take place in the second phase of the SHERPA methodology (i.e. '**Search-Retrieve-Pool**').

- **summaries of topic-related results per project**

Summaries of the results of research conducted in the context of each project (from those identified in Phase 2 of the methodology), with regard to a topic, need to be produced. These summaries help to identify actions taken for each rural topic by each project, as well as produced results. Summaries of topic-related results per project are the output of the third phase of the SHERPA methodology (i.e. '**Extraction of research content and synthesis of outcomes**'). **Summaries of topic-related results** (from all relevant projects) integrate the results of research that has taken place in all projects related to a specific rural topic. They are also produced in the third phase of SHERPA methodology.

- **SHERPA Papers**

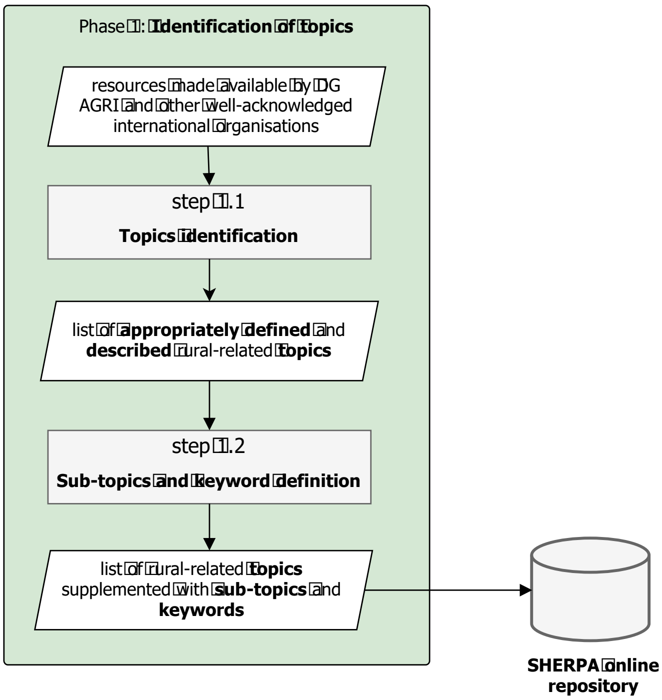
SHERPA Papers are documents reporting: (i) the input that is required for initiating interactions in the context of Multi-Actor Platforms (MAPs); and (ii) outputs of MAP-based interactions. The SHERPA Discussion Paper, which constitutes a specific case of SHERPA Papers produced in the fourth phase of the SHERPA methodology (i.e. '**Reporting research-related results**'), will provide input, relating to a specific rural topic, to the local and EU MAPs and is going to be created with the help of the summary of topic-related results.

The aim of this section is to provide details about the data outputs available by each methodology phase and descriptions of the information needed for annotating these data outputs (i.e. for metadata definition). Based on the above, the following subsections offer descriptions of the data produced by each phase of the SHERPA methodology, as well as of the necessary information for data annotation.

2.1. Rural topics – Phase 1

Phase 1 (i.e. '**Identification of topics**') is about the compilation of the list of rural topics that drive the project activities of SHERPA. The outputs of the 'Identification of topics' phase are:

- a list of rural-related topics;
- lists of sub-topics into which rural topics are subdivided;
- lists of keywords associated with each topic.



Phase 1: Identification of topics
resources made available by DG AGRI and other well-acknowledged international organisations
step 1.1
Topics identification
list of appropriately defined and described rural-related topics
step 1.2
Sub-topics and keyword definition
list of rural-related topics supplemented with sub-topics and keywords
SHERPA online repository

Figure 1: Steps of the 'Identification of topics' phase and data outputs stored in the SHERPA repository

Documents, reports and policy papers issued by DG-AGRI, and those from recognised organisations, such as the OECD, World Bank, FAO and UNESCO, are examples of resources which will be used for the identification and definition of rural topics. As shown in Figure 1 above, the list of rural topics, made available from the 'Identification of topics' phase, is stored into the SHERPA repository.

The information needed for describing each topic accurately includes:

- topic title;
- short topic description;
- date of topic definition;
- list of sub-topics into which the topic is subdivided;
- list of topic-related keywords; and
- title/DOI/URL of resources that have been utilised for the needs of topic definition.

2.2. Projects – Phase 2

The second phase of the methodology is the '**Search**-**Retrieve**-**Pool**' for the identification, selection and evaluation of past and ongoing research projects. This goal of this phase is to identify research projects (either completed or in-progress) that relate to specified rural topics and retrieve project-related information, as shown in Figure 2 below.

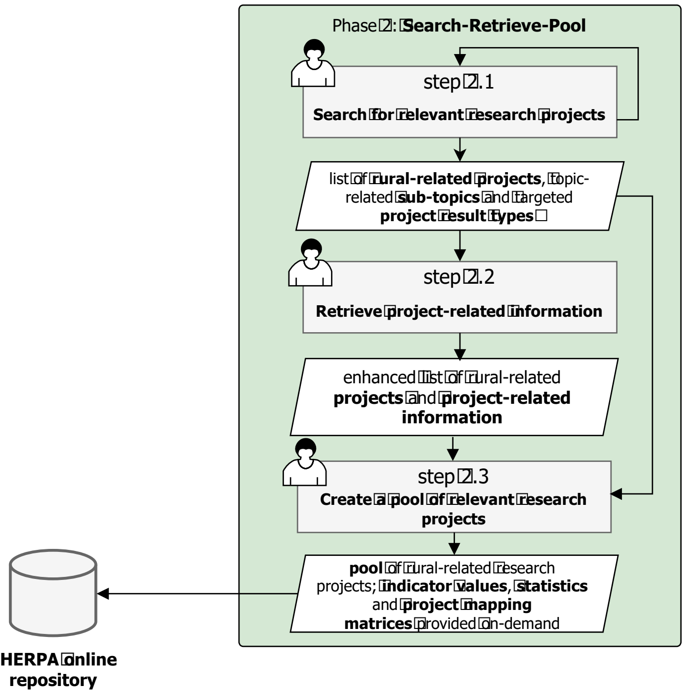

Phase 2: Search-Retrieve-Pool
step 2.1
Search for relevant research projects
list of rural-related projects, topic-related sub-topics and targeted project result types
step 2.2
Retrieve project-related information
enhanced list of rural-related projects and project-related information
step 2.3
Create a pool of relevant research projects
pool of rural-related research projects; indicator values, statistics and project mapping matrices provided on-demand
SHERPA online repository

Figure 2: Steps of the 'Search-Retrieve-Pool' phase and data outputs for storage in the SHERPA repository

The output of the '**Search**-**Retrieve**-**Pool**' phase is a pool of research projects, which have been concerned with the identified rural topics, and information about these projects. The project framework databases that are listed below have been used as key sources for finding and retrieving project-related information:

- CORDIS (https://cordis.europa.eu/projects/en);
- ESPON (https://www.espon.eu/);

- LIFE (https://ec.europa.eu/environment/life/project/Projects/index.cfm);
- ERA-NET (https://ec.europa.eu/research/fp7/index_en.cfm?pg=eranet-projects);
- INTERREG (https://www.interregeurope.eu/library/).

Each database offers different items of information per project. In some cases, similar information is provided but the fields are named differently in one database compared to another. Project-related information (in other words, project metadata), provided by these databases were compared to facilitate the identification of similarities. Such comparisons, taking the form of metadata schema mappings, will help identify similarities in project-related information provided by project framework databases. Table 1 summarises project metadata (i.e. project-related information) made available by the considered project framework databases. The lists of metadata presented in this table have been retrieved as the outcome of thorough investigation of the respective databases and the information provided by them.

Table 1: Project metadata provided by each research project framework database

Database Name	Project-related Metadata				

CORDIS	**FP6**	Project title;
		Project acronym;
		Grant agreement;
		Project website URL;
		Start/End date;
		Funded under programme;
			Total project budget;
			EU contribution to project funding;
			Project objective;
			Related programme;
			Related topic(s);
			Call for proposal;
				Funding scheme;
				Project coordinator;
				Coordinator details;
				Partner organisation details;
				Project results;	

	FP7	Project title;
		Project acronym;
		Grant agreement;
		Project website URL;
		Start/End date;
		Funded under programme;
			Total project budget;
			EU contribution to project funding;
			Project objective;
			Field of Science;
			Related programme;
			Related topic(s);
				Call for proposal;
				Funding scheme;
				Project coordinator;
				Coordinator details;
				Partner organisation details;
				Project results;
				Project-related keywords;	

	H2020	Project title;
		Project acronym;
		Grant agreement;
		Project website URL;
		Start/End date;
		Funded under programme;
			Total project budget;
			EU contribution to project funding;
			Project objective;
			Field of Science;
			Related programme;
			Related topic(s);
				Call for proposal;
				Funding scheme;
				Project coordinator;
				Coordinator details;
				Partner organisation details;
				Project results;
				Project-related keywords;	

LIFE	**Project description** Background Objectives Results	**Environmental issues** Themes + Keywords Target habitat types Natura 2000 sites	**Beneficiaries** Project coordinator – Type of organisation – Description Partners	**Administrative data** Project reference Duration Total budget + EU contribution Project location	
ESPON	Project acronym Project full name Project theme/Thematic scope Lead partner	Project budget Project lifetime Delivery of reports Publishing	Contact information Main research areas Main results envisaged		
INTERREG	Project acronym Project full name	Project summary Overall budget	Start/end date Partnership		

A review of Table 1 illustrates the significant differences in the amount and type of project-related information provided by the different databases. The Cordis database provides a broad range of information compared to the databases of LIFE, ESPON and INTERREG projects. Some cases are specialized in nature (e.g. LIFE projects deal specifically with environment-related topics), thus some items of information are provided which are not relevant to other project programmes (for example, the LIFE projects database provides information about '**Target habitat types**' and '**Natura 2000 sites**').

Mapping the set of project-related information available from one database to the information set of another enables the identification of similarities in the information being provided. For example, the type of information available in the Cordis database as '**Project objective**' is similar to that in the '**Background**' and '**Objective**' fields of the LIFE database, and the '**Project theme**/**Thematic scope**' and '**Project summary**' fields of the ESPON and INTERREG project databases.

There are also differences in the information provided by a single database. This is the case with the CORDIS database and the information offered for FP6, FP7 and H2020 projects. For example, the '**Field of Science**' and '**Project-related keywords**' properties are only available for FP7 and H2020 projects and not projects that have been implemented in the context of the FP6 framework. In addition to that, it needs to be stressed that more coordinator- and partner-related details are being provided for FP7 and H2020 projects than in the case of FP6 projects.

The above analysis lays the foundations for making decisions about project-related information needed to be stored and definition of metadata. Appropriately defined metadata are important for efficient project search and project-related information retrieval. By taking account of both the differences and similarities in provided by existing databases information, Table 2 below presents a **baseline set of project-related information** that needs to be stored with regard to research projects.

Table 2: Project-related information and brief explanation, to be stored into the SHERPA online repository

Project-related Information	Explanation and Details
Project acronym	The acronym of the project. This is usually the way in which a project is publicly known. When information about a project is required, its acronym is used in order to execute a search in a project programme database and retrieve the information required.
Project full name	The project's full name. For example, the full name of the project having the acronym DESIRA is 'Digitisation: Economic and Social Impacts in Rural Areas'.
Research project framework	Research in SHERPA focuses on (completed or ongoing) projects implemented in the context of FP6, FP7, H2020, LIFE, ESPON, ERA-NET and INTERREG programmes. This is the information about the programme through which the project was funded.
Grant agreement	Contract number of the agreement between the EC and the project coordinator.
Project website URL	The URL of the project website.
Project framework page URL	The URL of the relevant project programme page with information about the project.
Project description	Short description of the project scope and objectives.
Start year	The year in which the project has been launched.
End year	The year in which the project ended or will end.
Project duration	The total duration of the project in years.
Total budget	The total budget of the project (in euros).
Project coordinator	The name of the organisation that leads the project. The type of the coordinating organisation and the organisation's website URL is also provided.

Project-related Information	Explanation and Details
Project partners	The names of organisations that participate in the project and the type of organisation.
Project output types	Output types are provided in the case that project-related outputs are made available. Outputs types are: deliverables, reports, publications and grey literature.
Types of results produced	Results produced are not always outputs or documents. They can be: policies and policy recommendations, data and indicators, interventions, etc.
Date of information retrieval	The date when project-related information was found and retrieved.

Phase 2 of the SHERPA methodology for the identification, selection and evaluation of past and ongoing research projects also involves the automated, on-demand generation of statistics, indicators and project categorisations (namely the categorisations of projects with regard to topics and sub-topics, as well as result types). This type of information is produced after the execution of user queries and made available in the form of reports for downloading and local storage on a user's computer or device. This type of information will not be stored in the SHERPA online repository. The input needed for the generation of these reports is made up of information stored in the repository.

2.3. Summaries of topic-related results per project – Phase 3

Outputs of Phase 3 of the SHERPA methodology '**Extraction of research content and synthesis of outcomes**') is a summary of topic-related results produced by as single project, or by all projects relating to the rural topic.

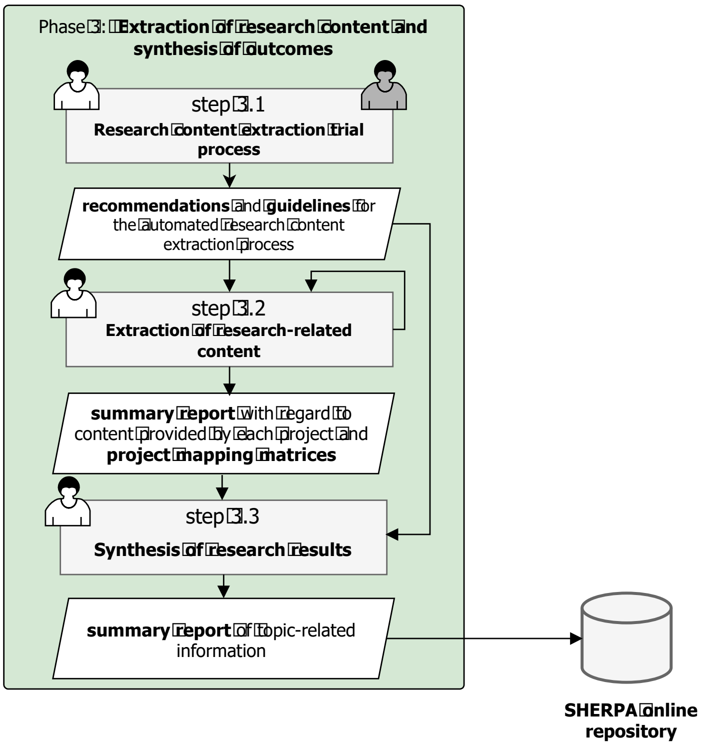

Phase 3: **Extraction of research content and synthesis of outcomes**
step 3.1
Research content extraction trial process
recommendations and **guidelines** for the automated research content extraction process
step 3.2
Extraction of research-related content
summary report with regard to content provided by each project and **project mapping matrices**
step 3.3
Synthesis of research results
summary report of topic-related information
SHERPA online repository

Figure 3: Steps in the 'Extraction of research content and synthesis of outcomes' phase and data outputs stored in the SHERPA repository

Summaries' content (i.e. results related to a rural topic) is extracted from project outputs through use of state-of-the-art text mining tools. Generation of result summaries takes place with the significant contribution of the Scientific Editor, who is responsible for refining the outcomes of the automated content extraction and fine tuning the automatically generated summaries.

The information to be stored with topic-related summaries of relevant project is summarised below:

- the methodology that has been employed in the context of the project;
- available result types (e.g. policy recommendations, data and indicators, interventions, etc.);
- description of results;
- the title(s) of source(s) of results (i.e. title of the output from which results have been drawn);
- author(s) of the source(s) of results (i.e. name(s) of author(s) of the outputs that constitute sources of topic-related results);
- author(s) contact details (i.e. email account of the author or authors of the project outputs that have been utilised as sources of topic-related results);
- date of result(s) retrieval.

Project outputs used as sources of topic-related results (i.e. deliverables, reports, publications, grey literature) will not be stored in the SHERPA repository. The URL of project website and the respective project programme page, from where project outputs have been retrieved, will instead be made available.

2.4. Summaries of topic-related results (from all relevant projects) – Phase 4

The final output of the SHERPA methodology for the identification, selection and evaluation of past and ongoing research projects is the **SHERPA Discussion Paper**

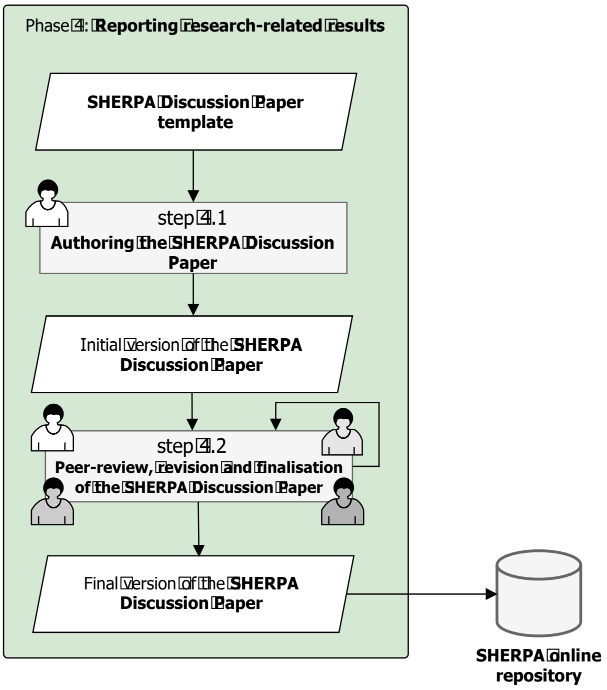

Phase 4: Reporting research-related results
SHERPA Discussion Paper template
step 4.1
Authoring the SHERPA Discussion Paper
Initial version of the SHERPA Discussion Paper
step 4.2
Peer-review, revision and finalisation of the SHERPA Discussion Paper
Final version of the SHERPA Discussion Paper
SHERPA online repository

Figure 4: Steps in the 'Reporting research-related results' phase and data outputs stored in the SHERPA repository

The SHERPA Discussion Paper is a document that informs discussions of the local Multi-Actor Platforms (MAPs). As shown in Figure 4 above, the SHERPA Discussion Paper is the output of the last phase of the SHERPA methodology (i.e. '**Reporting research-related results**').

Development of the SHERPA Discussion Paper is based on summaries of topic-related results from all relevant projects. These summaries are generated, in an automated way, through the aggregation of summaries of topic-related results per project and reviewed, for purposes of fine-tuning, by the Scientific Editor. The summaries of topic-related results (from all relevant projects) are stored into the SHERPA online repository, providing information about:

- descriptions of summaries of topic-related results;
- associated result types; and
- references to policy recommendations that may have been provided as project results.

2.5. SHERPA Papers

The **SHERPA Discussion Paper** is one of the papers produced in SHERPA either for providing input to MAP-based interactions or as reports of activities in the context of MAPs[1]. Other SHERPA Papers are:

- the **MAP Discussion Paper** (developed with input from the SHERPA Discussion Paper and used for guiding interactions in local MAPs);
- the **MAP Position Paper** (report of the outcomes of interactions in local MAPs); and
- the **EU MAP/SHERPA Position Paper** (report of the outcomes of interactions in the EU MAP).

All SHERPA Papers are stored in the SHERPA repository. The information (i.e. metadata) required for the storage of this type of textual data is summarised in Table 3. This information is properties of the **SHERPA Papers** that make them searchable and findable. These properties are proposed and defined by the Dublin Core Metadata Initiative[2].

Table 3: Metadata properties for SHERPA Papers and their definitions

Metadata Property Name	Metadata Property Definition
Abstract	A short textual description of the paper in the form of summary.
Audience	Group(s) of people for whom the paper is designed.
Creator name	The person responsible for the development of the SHERPA Paper.
Creator contact details	Email address of the person responsible for the development of the SHERPA paper.
Date available	The date on which the paper has become publicly available through the SHERPA online repository.
Bibliographic citation	Description of how the paper should be referenced in other documents.
Date created	The date on which the currently available version of the paper has been created.
Format	The format of the digital file in which the Paper is made available.
Identifier	A unique identification code of the paper.

[1] More details about SHERPA Papers are provided in Deliverable 1.2 ('Working principles of the Multi-Actor Platforms').
[2] https://www.dublincore.org/

Metadata Property Name	Metadata Property Definition
Keywords	A list of keywords that are associated with the paper.
Language	The language(s) in which the paper is available.
License	License description that makes explicit the ways in which the paper may be used.
Title	The title of the paper.
Type	Type of the paper. It can take any value from the set {SHERPA Discussion Paper, MAP Discussion Paper, MAP Position Paper, EU MAP/SHERPA Position Paper}.
Version	Current version of the SHERPA paper.

2.6. Metadata properties for data stored in the SHERPA online repository

Table 4 summarises **all of the types of data** that need to be stored into the SHERPA online repository, their source and the metadata needed for their description.

Table 4: Data stored in the SHERPA online repository, their source and metadata required for their description

Stored Data	Data Source	Metadata Properties
Rural topics	Output of the 'Identification of topics' phase of SHERPA methodology for identification, selection and evaluation of past and ongoing research projects.	• topicID • topicTitle • topicDescription • dateOfTopicDefinition • subTopics • keywords • associatedResources (title, DOI, URL)
Projects	Output of the 'Search-Retrieve-Pool' phase of SHERPA methodology for identification, selection and evaluation of past and ongoing research projects.	• projectID • projectAcronym • projectFullName • researchProjectFramework • grantAgreement • projectWebsiteURL • projectFrameworkPageURL • projectDescription • startYear • endYear • duration • totalBudget • projectCoordinator (name, type, website URL) • projectPartners (name, type, website URL) • projectOutputTypes • resultTypes • dateOfInformationRetrieval
Summary of topic-related results per project	Output of the 'Extraction of research content and synthesis of outcomes' phase of the SHERPA methodology for the identification, selection and evaluation of past and ongoing research projects.	• summaryID • employedMethodology • resultTypes • resultDescription • resultSource (title, author name and author contact details) • dateOfResultRetrieval
Summary of topic-related results (from all relevant projects)	Output of the 'Extraction of research content and synthesis of outcomes' phase of the SHERPA methodology for the identification, selection and evaluation of	• topicSummaryID • summaryOfTopicRelatedResults • policyRecommendations

Stored Data	Data Source	Metadata Properties
	past and ongoing research projects.	
SHERPA Papers	**SHERPA Discussion Paper** Output of the 'Reporting research-related results' phase of the SHERPA methodology for the identification, selection and evaluation of past and ongoing research projects. **MAP Discussion Paper** Paper produced by the Facilitators[3] of local MAPs for providing input to MAP-based discussions. **MAP Position Paper** Paper produced by Facilitators of local MAPs for documentation of outcomes of discussions in local MAPs. **EU MAP/SHERPA Position Paper** Paper produced by the Facilitator of the EU MAP for documentation of outcomes of discussions in its context.	• paperID • title • type • version • abstract • keywords • language • audience • creator (name and contact details) • dateCreated • dateAvailable • bibliographicCitation • format • license

It needs to be stressed that from the information seeking point of view, the response provided to a query is useful to the user when the appropriate amount and type of information is made available. For instance, when retrieving a SHERPA Paper, apart from Paper-related information, it is important to know what is the rural topic that the Paper relates to. So, topic-related information should be provided as well. Similarly, when searching for projects that are associated with a rural topic, both project and topic-related information should be returned. Joins of data-related information, such as those mentioned above, can be achieved as a result of the relationships between the various types of data stored in the SHERPA repository. These relationships are presented in Sub-section 5.2 (i.e. '**Data persistence layer**').

3. SHERPA online repository system end-user types and needs

An integral part of the description of the SHERPA online repository design is the definition of the profiles of targeted end-user types together with the description of their needs. To this end, Section 3 starts with the presentation of the profiles of the SHERPA online repository end-user types and then continues with an overview of their needs.

[3] Description of the role of MAP Facilitators is provided in Deliverable 1.2 ('Working principles of the Multi-Actor Platforms').

3.1. Profile definition of targeted end-user types

The types of end-users that the SHERPA online repository targets at belong to three major categories. These categories are: (i) **SHERPA consortium member**, (ii) **SHERPA network member**, and (iii) **member of the wider community**. More specifically:

- The SHERPA consortium member category comprises the following end-user types: Scientific Editor, Review Editor, Communication Editor, Support Staff, MAP Facilitator and MAP Monitor, SHERPA data repository Administrator, and other SHERPA consortium member.
- The SHERPA network member category comprises the 'Project coordinator' and 'Project consortium member' end-user types.
- The 'Member of the wider community' category comprises the 'SHERPA MAP member' and 'Society member' end-user types.

Detailed profile definitions of each end-user category and respective type are provided in Table 5 below.

Table 5: Categories and types of the SHERPA end-users and definitions of their profiles

End-user Category	Associated End-user Type	Profile Definition of End-user Type
SHERPA consortium member	**Scientific Editor**	Assesses the appropriateness and quality of results from automated project search and filtering. Contributes to the creation and updating of sub-topic lists and lists of types of project results to be reviewed for the relevance of their content. Reviews and edits automatically produced project summaries. Supervises the process of extraction of research content from project documents, reviews and fine tunes outcomes. Authors the SHERPA Discussion Paper and proceeds to revisions based on feedback received.
	Review Editor	Provides comments and feedback about the content of the SHERPA Discussion Paper. Feedback provided is used for revisions to the SHERPA Discussion Paper and development of the final version.
	Communication Editor	Provides editorial suggestions and comments related to content of the SHERPA Discussion Paper and issues of layout. Feedback provided is used for revisions to the SHERPA Discussion Paper and development of the final version of the document.
	Support Staff	Supports the Scientific Editor in: (i) execution of (further) manual search for rural-related projects, and (ii) integrating changes in the SHERPA Discussion Paper (after review and comments).
	MAP Facilitator	Undertakes activities related to the invitation and enrolment of MAP members. Executes actions necessary for the operation of the local/EU MAP. Prepares the SHERPA Discussion and Position Papers. Organises dissemination activities and attends training events.
	MAP Monitor	Provides Support to the MAP Facilitator by: (i) inviting actors to the local MAP; and (ii) observing the dynamics of the MAP. Assists the MAP Facilitator by adding reflexivity to MAP activities and encouraging a positive approach during the MAP meetings.

End-user Category	Associated End-user Type	Profile Definition of End-user Type
	SHERPA data repository Administrator	Member of the Agricultural University of Athens (i.e. the partner responsible for the data repository development) who has administrative tasks related to the operation of the SHERPA online repository.
	Other consortium member	SHERPA consortium members that do not fall into any of the above categories.
SHERPA network member	**Project consortium member**	Users of the SHERPA online repository system who belong to organisations related to the wider SHERPA network. These organisations are coordinators or consortium members in ongoing research projects of interest to SHERPA. End-user types falling into the specific category are interested in accessing content and information available by the SHERPA repository, but they are also asked to proceed to creation of new data records regarding information and results about the research project they represent. Update/deletion of data records, related to projects that the specific end-user types represent, should also be enabled.
	Project coordinator	
Member of the wider community	**SHERPA MAP member**	This end-user type comprises scientists/researchers, policy representatives/makers and citizens. Members of SHERPA MAPs are not members of the project consortium, but are actively involved in SHERPA-related activities. Through their participation in MAP-based interactions they contribute to the development of recommendations for future policies and research concerning rural areas in the EU.
	Society member	Members of society who are not in any of the end-user types and categories above. This type of end-users may include representatives of civil society organisations, policy makers, members of the scientific community, or citizens who may exhibit an interest in SHERPA-related research and activities and, thus, would be interested in having access to the SHERPA online repository in order to make use of it.

3.2. End-user needs

Having described the profile of each of the end-user types targeted by the SHERPA system, the following step is to provide accounts of specific user needs. Table 6 provides definitions of needs, and brief explanations, and presents associations between these needs and user type categories.

Table 6: Description of needs and association with end-user types

End-user Need	Brief Description of User Need	Associated End-user Category
Access to information related to rural topics	Information related to one or more rural topics needs to be accessed. This information may relate to the title of the topic(s), or the associated sub-topics and keyword lists.	SHERPA consortium member SHERPA network member Member of the wider community
Access to information related to projects addressing a given topic	Information related to projects addressing a specific rural topic needs to be accessed. This information may relate to project title, partner and coordinator details, budget, project duration and the types of results available. Information about the given topic is also necessary.	SHERPA consortium member SHERPA network member Member of the wider community
Access to individual project results on a given topic	Information or content related to research results offered by a research project needs to	SHERPA consortium member SHERPA network member

End-user Need	Brief Description of User Need	Associated End-user Category
	be accessed. This information should be about type(s) of results and brief description of them. Project outputs from which results have been made available and author(s) details may also be required.	Member of the wider community
Access to summary information, on a given topic, from the stock of science evidence from past and ongoing research projects	Information or content related to results from all topic-related projects needs to be accessed.	SHERPA consortium member SHERPA network member Member of the wider community
Access to SHERPA Papers	Any SHERPA Paper needs to be accessed.	SHERPA consortium member SHERPA network member Member of the wider community
Access to statistics and indicators related to research activity in a given topic or the full set of rural topics	Statistics and indicators need to be accessed. They enable insights to the research activity relating to one or more rural topics.	SHERPA consortium member SHERPA network member Member of the wider community
Access to information about project classification	Projects relating to a rural topic can be classified according to sub-topics and types of topic-related results. Outcomes of classification are provided in the form of reports generated on-demand.	SHERPA consortium member SHERPA network member Member of the wider community
Access to the most recent results relevant to rural policy	Access needed to recent project results relevant to rural policy. The user determines the time period for which results would be relevant.	SHERPA consortium member SHERPA network member Member of the wider community
Provision of project results relevant to rural policy	Project results relevant to rural policy. These results come from projects in progress for which results are not yet publicly available. This need is a subset of that for ‘Creation of new data records with regard to rural topics, projects, individual results on a topic or summary information on a given topic’.	SHERPA consortium member SHERPA network member
Creation of new data records with regard to rural topics, projects, individual results on a topic or summary information on a given topic	New data records need to be created. These data records may be about rural topics, topic-related projects, individual results on a topic, or summary information on a given topic. This need is a superset of the that for ‘Provision of project results relevant to rural policy‘.	SHERPA consortium member SHERPA network member
Update of information related to rural topics, projects, results from a single project on a topic or summary of information on a given topic	Data records need to be updated. These data records may relate to rural topics, topic-related projects, individual results on a topic, or summary information on a given topic. This need is a superset of that for ‘Creation of new data records with regard to rural topics, projects, individual results on a topic or summary information on a given topic’.	SHERPA consortium member SHERPA network member
Creation and update of data records related to SHERPA Papers	Data records related to SHERPA Papers need to be created or updated. A new SHERPA Paper may need to be stored or a data record about an existing SHERPA Paper may need updating.	SHERPA consortium member

End-user Need	Brief Description of User Need	Associated End-user Category
Deletion of non-relevant content and information	Information or content related to rural topics, projects or results deemed to be irrelevant to the topic or the SHERPA Papers can be deleted.	SHERPA consortium member SHERPA network member

The above user need descriptions serve as the basis for alignment between end-user categories, respective types of end-users, and system functionalities. This alignment is provided in Section 5 and more specifically paragraph 5.4.3 (namely, '**System functionalities provided to each end-user type**').

4. Technology framework

Section 4 starts with an overview of NoSQL data store systems and MongoDB, i.e. the particular representative of NoSQL systems adopted by SHERPA. That is followed by the descriptions of the fundamentals of search engine technologies and frameworks used for the development of the back- and front-end.

4.1. Data storage

4.1.1. Introduction to NoSQL data store systems

Relational Database Management Systems (RDBMSs) comprise core data storage and management software solutions, which have been used for decades. They are highly consistent; however, this comes at the expense of not being able to be scaled horizontally. NoSQL systems, on the other hand, enable horizontal scalability but, at the cost of consistency. NoSQL data store systems adopt different principles for storing data compared to RDBMSs. These differences are highlighted by Tiwari (2011) according to whom, NoSQL is "*an umbrella term for all databases and data stores that don't follow the popular and well-established RDBMS principles and often relate to large data sets accessed and manipulated on a Web scale*" (p. 4). Trade-offs between efficient data storage and compliance with the ACID principles (namely, Atomicity, Consistency, Isolation and Durability) are stressed by Vaish (2013) who notes that the term NoSQL is used to refer to "*databases that attempt to solve the problems of scalability and availability against that of atomicity or consistency*" (p. 9). NoSQL data store systems support all CRUD operations (namely, Create, Read, Update and Delete)[4] despite the fact that there are some differences in implementation between the various NoSQL system types and products (Tiwari, 2011).

Horizontal scaling refers to the use of clusters of commodity hardware to store data, with each piece of hardware being responsible for the execution of processes, such as look-ups and read/write operations, on the data stored (Lake and Crowther, 2013). This specific capacity of NoSQL data stores is identified in the definition of Cattell (2010) according to whom NoSQL systems are "*designed to provide good horizontal scalability for simple read/write database operations distributed over many servers*" (p. 12).

However, advantages of NoSQL systems go beyond scalability-related issues. According to Vaish (2013), NoSQL data stores enable the representation of schemaless data, which means that application developers are able to dynamically integrate changes into their design without needing to predefine a fixed data structure. Schemaless data representation reduces development time given that data access is handled by application code rather than complex SQL queries. The absence of a strict schema makes querying more flexible with queries being addressed to the entire database (Vaish, 2013). Over recent years, a substantial

[4] An overview of CRUD operations is provided in paragraph 5.4.1 ('Overview of CRUD operations').

number of widely-adopted open source products have emerged (CouchDB, MongoDB, etc.) allowing access through custom-made APIs in addition to their built-in shell-based environments (Lake and Crowther, 2013). As a result, there is potential to develop applications able to efficiently respond to various workloads and deliver results very quickly.

4.1.2. The case of MongoDB

The solution adopted for the SHERPA data repository is MongoDB[5] specifically, the free to use MongoDB Community Edition. MongoDB belongs to the document-oriented family of NoSQL systems. MongoDB is "*a document store that can persist arbitrary collections of data as long as it can be represented using a JSON-like object hierarchy*" (Tiwari, 2011, p. 47). It supports standard JSON data types (i.e. integer, string, Boolean, double, null, array, and object) and additional types, such as date, binary data and regular expression. As a specific type of NoSQL systems, document-oriented databases allow for the adoption of a dynamic or changeable schema, or no schema at all. This feature makes them ideal for storage of content that changes over time (Vaish, 2013).

MongoDB is an open-source database management system providing high read and write throughput[6], as well as the ability for horizontal scaling and automatic failure recovery (Banker, 2012). It has become popular because of its capacity to efficiently represent and retrieve hierarchically structured information without the need for execution of resource-intensive table joins (Banker, 2012). Moreover, according to Banker (2012), MongoDB supports ad hoc queries and data indexing. It also provides automatic data replication, which means distribution of data across the nodes of a cluster in order to eradicate data loss due to hardware or network failure. The distribution of data across nodes is internally handled by a mechanism called auto-sharding. As a result, developers do not need to become involved in development of code for this purpose, but they can rather leverage the seamless way in which MongoDB scales out. Write operations are executed, by default, in the so-called 'fire-and-forget' mode. This means that no receipt of acknowledgement is required upon submission of the write operation. However, it may also be implemented in 'safe mode', which ensures that the operation has been successfully executed (Banker, 2012).

MongoDB is the data store system that **has been selected for** the deployment of the **SHERPA repository**, because of:

- its potential, as a NoSQL system, to efficiently handle various workloads through horizontal scaling;
- its capacity to cater for changes in data representation on the fly by supporting dynamic schemas or no schema at all;
- the fact that query execution is handled through application code by avoiding complex table joins;
- the standard data exchange format (i.e. JSON) used for data storage and search results delivery and the potential it offers for development of any application on top of the data store; and
- the large MongoDB community that can easily be reached and advised for issues relating to the data repository development and maintenance.

4.1.3. MongoDB's document model

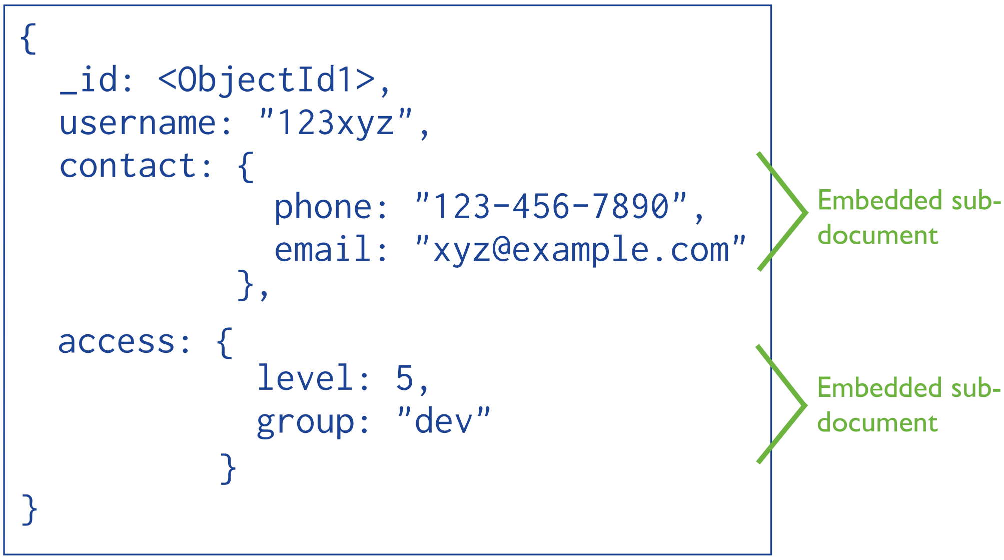
Document-oriented data stores (or document stores) employ a data model based upon the XML, JSON, BSON, or YAML data exchange formats (Vaish, 2013). The term document is used to denote "*loosely structured sets of key/value pairs*" in files of supported formats rather than documents in the sense of text available in some

[5] https://www.mongodb.com/

[6] According to Berkeley DB's Performance Metrics & Benchmarks White Paper, 'throughput' is defined as "*records read or written in a fixed time interval*" and is the most common measure of database performance.

digital format (Tiwari, 2011, p. 18). In essence, any digital object can be regarded as a document and stored in a document store. To draw analogy with RDBMSs, a document is the table row counterpart though semi-structured in nature (Vaish, 2013). The document model, based on the JSON (namely, JavaScript Object Notation) data exchange format, is the database model employed by MongoDB for data storage. MongoDB makes use of a binary representation of JSON called BSON (Lake and Crowther, 2013). BSON documents can have a maximum size of 16 MBs. Larger documents can also be stored, however, with the help of the GridFS API.

A document is made up of property names and values. Values can be of any BSON compatible data type (e.g. string, number, Boolean, date, etc.), an array, a document (called embedded document) and an array of documents. Thus, MongoDB offers flexibility for modelling complex data structures (Banker, 2012). A 'collection' is a group of documents and can be considered the equivalent of a table of a relational database. By being schemaless, MongoDB enables to documents with different fields in the same collection. Figure 5 provides an example of a MongoDB document.



```
{
  _id: <ObjectId1>,
  username: "123xyz",
  contact: {
              phone: "123-456-7890",
              email: "xyz@example.com"
            },
  access: {
             level: 5,
             group: "dev"
           }
}
```

Figure 5: Overview of MongoDB's document model (source: https://docs.mongodb.com/manual/core/data-modeling-introduction/)

4.2. Information retrieval and search engine technology fundamentals

The meaning of Information Retrieval (IR) can be very broad. It is an umbrella term used to denote any retrieval mechanism that is implemented on top of unstructured data (i.e. data which does not have a semantic computer-oriented structure). In reality, almost no data are completely unstructured which is because of the latent linguistic structure of human's natural language. However, there are significant differences with the canonical relational database paradigm in which data is structured. Ideally, the goal of information retrieval is to respond to users' queries rather than just give documents that might be related to that query. Due to a problem of information overload, information retrieval is quickly becoming the dominant form of information access overtaking traditional database style searching. As an academic field of study, information retrieval could be defined as a set of different techniques for finding resources of an unstructured nature, satisfying an information need, from within large collections usually stored on computers (Manning *et al.*, 2008). Given that, it can be concluded that information retrieval encompasses the acquisition, organization, storage, retrieval, and distribution of information. Information retrieval also encompasses related topics, such as information detection, topic extraction, summarization and document filtering.

An information need relates to something about which the user wants to know more and, thus, uses a query (probably consisting of a number of keywords) to communicate that need. If the user perceives a retrieved document as containing valuable information, then the document is considered to be relevant.

There are two main metrics that impact the performance of an information retrieval system and are used to assess the quality of returned results, namely: (i) precision, i.e. the fraction of retrieved documents which are relevant to the information need, and (ii) recall, i.e. the fraction of relevant documents in the collection retrieved by the system. An important factor that characterises information retrieval systems is the scale at which they operate with the World Wide Web being the most challenging one. In web searches, the system has to search billions of documents stored on millions of computers. It is important to recognize that information retrieval is broader than the search engine concept. That is only one of the many different kinds of information retrieval systems working on content distributed across web documents. At the enormous scale of the World Wide Web, distinctive issues need to be addressed to build systems that work efficiently.

From a technical perspective, there are two main concepts: (i) inverted index, and (ii) document model. Inverted index is central to the first major concept in information retrieval. Indexing is the method of storing text data in index files within a format that helps searching in a fast and efficient way. To achieve this goal, statistical methods (for example, machine learning techniques) can help to map extracted terms in an ontology containing the relevant vocabulary representing a domain. Moreover, the weighting schemes given to the index represent the main differences between the different search engines. Another key technical part is the model or representation of documents and queries, which will impact the final relevance of retrieved documents. Some of these are the 'Boolean model', the 'Vector space model' and the 'Statistical language model'.

4.3. Back- and front-end development frameworks

In the recent years, many applications have been deployed within the web infrastructure. In the web industry, a distinction has emerged between two main parts of application development, namely: front-end and back-end (Lindley, 2019). Front-end refers to the client-side or what the user will see on the application and how the user will interact with the application through the browser. It comprises several aesthetic issues, such as the design of the user experience, images selection, fonts and colours, formatting, and laying out the content according to the device display and browser's JavaScript engine. The principal technologies in the front-end are a combination of HTML, CSS and JavaScript although, in recent years, various frameworks have emerged which improve the developments. These frameworks are Angular[7], Bootstrap[8], jQuery[9], Sass[10], etc.

Back-end refers to the server-side of the application or the place where business logic is run. While front-end code usually runs on the browser, the back-end code runs on a web server. Consequently, the parts and characteristics implemented by back-end developers are indirectly accessed by users through the front-end side. In other words, when a user clicks on a feature in the application the process will run in the backend behind the scenes and a result will be returned. In the case of search engines, after pressing the 'Enter' key inside the search bar (front-end), the back-end will compute and retrieve a list of relevant (or not) documents. Since the back-end is responsible for data storage, several databases are used in order to cover a range of needs. Such databases are MySQL[11], MongoDB[12], RDF4J[13], etc. As opposed to languages used on the client

[7] https://angular.io/
[8] https://getbootstrap.com/
[9] https://jquery.com/
[10] https://sass-lang.com/
[11] https://www.mysql.com/
[12] https://www.mongodb.com/es
[13] https://rdf4j.org/

side, back-end is built using languages, such as Java, Python and PHP. JavaScript beside Node.js[14] have also arisen as a feasible technological option in the back-end.

5. SHERPA Online repository system architecture

This section focuses on a detailed description of the architecture of the SHERPA online repository. After an introduction to the system's architecture, in Sub-section 5.1, details about each architecture layer are provided. Sub-sections 5.2, 5.3 and 5.4 offer the necessary documentation for specific design choices, which will be 'appropriately translated' in the development phase. In addition, Sub-section 5.5 is concerned with descriptions of the services intended to be exploited for the needs of implementing the necessary 'communication' between the various modules of the system.

5.1. The SHERPA online repository system architecture: overview

Complex information systems, such as one for information retrieval, often employ multiple components with different requirements. Given that, the multi-tier architecture paradigm has emerged in the software engineering domain. This architecture paradigm is a type of client–server architecture in which presentation, application and data functionalities are logically and (probably) physically separated. The concepts of layer and tier are often used interchangeably. However, the convention is to use 'layer' as a logical component, and 'tier' as the physical components of the infrastructure in which the application runs. This architecture provides a model by which developers can create flexible and reusable applications, improving on the monolithic approach in which all layers are tightly connected on the same machine.

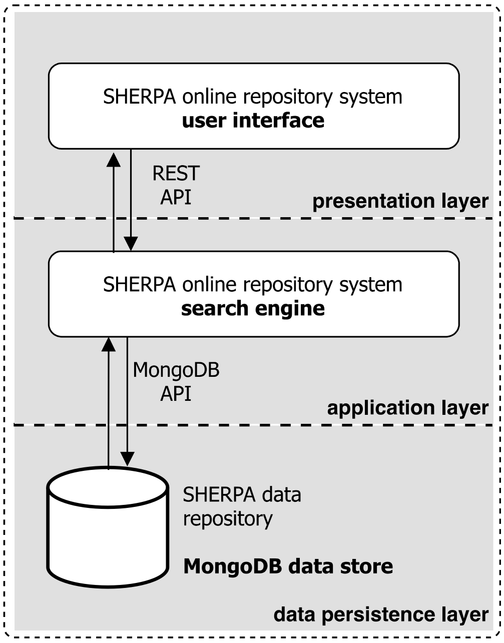


Figure 6: SHERPA online repository system three-layer architecture

In web development, the most widely adopted multi-tier approach is the three-tier architecture. The three-tier architecture is a client-server architectural design in which the user interface, business logic and data storage are developed and maintained as independent modules, most often on separate platforms. The three-tier architecture allows the upgrading of any of the tiers independently in response to changes in user

[14] https://nodejs.org/es/

requirements or technology. For example, a change to the database where the Information Retrieval system is storing the documents would not affect the presentation layer and, consequently, the user would see no change. Three-tier architecture is a significant improvement on the more traditional two-tier design, in which the presentation layer is usually considered part of the business layer placing substantial loads on the network.

The system design of the SHERPA online repository is based upon a three-layer architecture (i.e. the data persistence layer, the application layer and the presentation layer) as shown in Figure 6 below. System architecture layers are presented in the following paragraphs. Finally, Sub-section 5.5 focuses on description of the Application Programming Interfaces (APIs)[15] used for the communication of the system's modules.

5.2. Data persistence layer

5.2.1. SHERPA data model

The data persistence layer is responsible for data storage and is the system's layer where the data repository lies. Consistent data storage requires a rigorous data model. Therefore, this paragraph is concerned with the description of the data model employed in SHERPA. The SHERPA data model makes explicit the way in which data is structured by illustrating model-related entities, associated with the types of data stored into the data repository (described in Section 2), and relationships between them (see Figure 7 below).

- **Entities**

The entities of the SHERPA data model are: **Rural Topic** (relates to data that need to be stored with regard to the rural topics on which SHERPA research is going to focus), **Project** (data needed to be stored in relation to projects that are associated with the investigated rural topics), **Summary of topic-related results per project** (data relating to results produced by a single project with regard to a specific rural topic), **Summary of topic-related results** (data relating to results available from all projects that are associated with a specific rural topic), and **SHERPA Papers** (data related to the produced SHERPA Papers).

- **Relationships between entities**

Relationships between entities have been derived from the below presented modelling-related assumptions:

A Rural Topic may be addressed by one or more Projects and a Project may focus on one or more Rural Topics. Therefore, there is a 'many-to-many' relationship between the entities **Rural Topic** and **Project**.

A SHERPA Paper addresses one Rural Topic and each Rural Topic is addressed by one SHERPA Paper. Therefore, there is a 'one-to-one' relationship between the entities **Rural Topic** and **SHERPA Paper**.

A Project may focus on one or more Rural Topics and, thus, relate to one or more SHERPA Papers. Each SHERPA Paper addresses one Rural Topic and, thus, may be related with one or more Projects (given the potential focus of a Project on one or more Rural Topics). Therefore, there is a 'many-to-many' relationship between the entities **SHERPA Paper** and **Project**.

A Rural Topic can relate to one or more Summaries of topic-related results per project (given that a Rural Topic may be addressed by one or more Projects). Therefore, there is a 'one-to-many' relationship between the entities **Rural Topic** and **Summary of topic-related results per project** with the 'one' part of the relationship being on the side of the Rural Topic entity and the 'many' part on side of the Summary of topic-related results per project entity.

[15] According to Webopedia (https://www.webopedia.com/TERM/A/API.html), an application programming interface (API) is a "*set of routines, protocols, and tools for building software applications*", which specify "*how software components should interact.*"

A Project can relate to one or more Summaries of topic-related results per project (given that a Project may focus on one or more Rural Topics). Therefore, there is a 'one-to-many' relationship between the entities **Project** and **Summary of topic-related results per project** with the 'one' part of the relationship being on the side of the Project entity and the 'many' part on the Summary of topic-related results per project side.

There is a 'one-to-many' relationship between the entities **Summary of topic-related results** and **Summary of topic-related results per project** with the 'one' part of the relationship being on the side of the Summary of topic-related results entity and the 'many' part on the Summary of topic-related results per project side.

A Project relates to one or more Summaries of topic-related results, given that each Project focuses on one or more Rural Topics. Each Summary of topic-related results per project relates to one or more Projects, given that a Rural Topic may be addressed by one or more Projects. Therefore, there is a 'many-to-many' relationship between the entities **Project** and **Summary of topic-related results**.

There is a 'one-to-one' relationship between the entities **SHERPA Paper** and **Summary of topic-related results**, given that each SHERPA Paper is created with input provided in one Summary of topic-related results and each Summary of topic-related results is used for creating one SHERPA Paper.

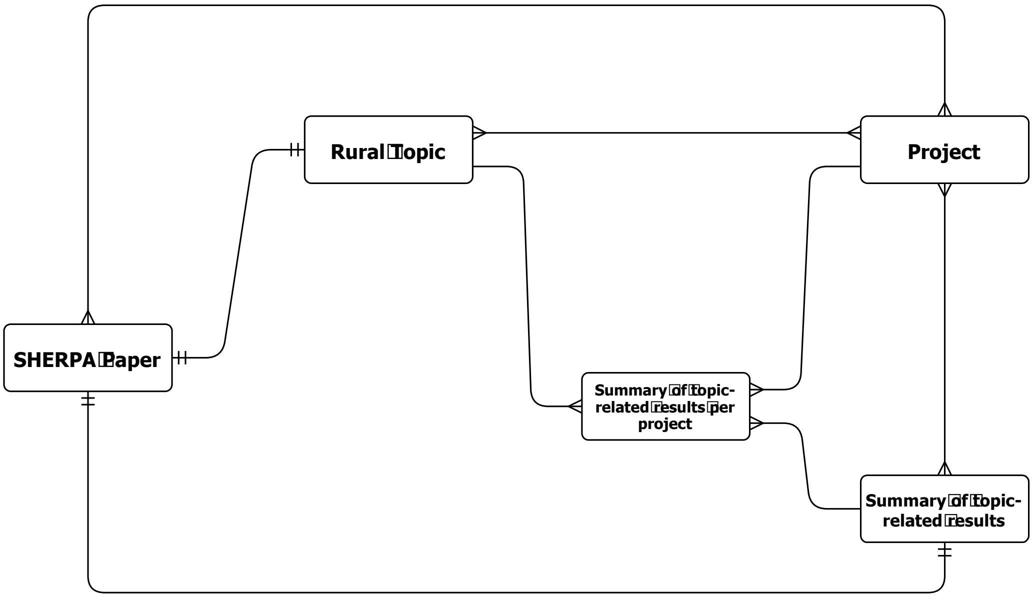

Rural Topic
Project
SHERPA Paper
Summary of topic-related results per project
Summary of topic-related results

Figure 7: SHERPA data model

The SHERPA data model is made more explicit by being 'translated' into a respective database model. The database model illustrates: (i) how data is stored into the SHERPA repository with the help of JSON documents (i.e. the data exchange format used in MongoDB), and (ii) references from one JSON document to another representing the relationships between entities. The SHERPA database model is provided in the Annex section of the document.

5.2.2. Compliance with the FAIR principles

Since every publisher can have their assumptions, guidelines and models, information may be published by making use of data formats that are processable by specific systems. Consequently, resulting data ecosystems may become less integrated day-by-day with serious effects on the discovery and reusability of resources by a data-intensive society. Suitable data management is key to knowledge discovery, its integration and subsequent innovation. To achieve this goal, a set of high-level, technology-agnostic principles, known as the FAIR Data Principles (Wilkinson *et al.*, 2016), have been developed and proposed. FAIR principles provide distinct considerations for data publishing to make digital resources **Findable**, **Accessible**, **Interoperable**, and **Reusable**. Moreover, due to the information overload experienced by human agents, the FAIR principles emphasise upon the capacity of computational systems to find, access and reuse data with minimum human intervention. It is important to note that implementation of these high-level principles is enabled by a range of technologies able to facilitate long-term maintainability of valuable digital assets.

The SHERPA online repository will undertake the FAIR principles at different levels. More specifically:

- Design is based on standard metadata schemas (namely, Dublin Core), where applicable, in order to make data findable.
- Accessibility will be maximised by making data accessible through an internet connection by the open HTTP protocol.
- Data will be encoded with JSON-LD given the employed API-based architecture.
- Interoperability will be achieved by using vocabularies following the FAIR principles and linking data with other FAIR datasets.
- Data will be made reusable by using extensive metadata and trying to integrate it with community-controlled vocabularies.

5.3. Application layer

A multi-tier architecture increases the level of decoupling of software components by creating an application layer, distinct from the presentation layer, which controls the main functionality of the application and executes demanding, process-related tasks separately. Examples of such tasks are functionalities dealing with information processing, execution of batch commands and coordination of different software services. The application layer implements the business logic of the system; in other words, it is the part of the program that encodes the real-world business rules that determine how data should be created, stored, and changed. This layer is different from the rest of the software which is concerned with lower-level details of managing the Operating System calls, displaying the user interface or connecting various components of the program.

The SHERPA online repository system application layer is mainly concerned with execution of user queries related to information retrieval. The execution of such queries requires several issues to be taken into account like the number of indexes and their types. For instance, a query for finding and retrieving geospatial data is not the same with a query for text retrieval. The latter usually involves natural language processing operations such as query extension, where the query is enriched with synonyms or equivalent words depending on the domain. This way, the number of relevant documents retrieved can be increased; however, this expansion can also lead to information overload with high recall but low precision. For that reason, a constant analysis of the search engine effectiveness must be performed. For this analytical purpose, the application layer must have a logging functionality too. This service will also be useful for defining the best caching policies in the application layer. The right decisions will definitely have an impact on user perceived system performance.

Human language contains concepts described by multiple terms and, thus, the application layer should have the capacity to recognise and process them efficiently. For example, using the term 'stone walls' is different than making use of the terms 'stone' and 'walls'. If the terms are divided, any kind of document containing the term 'walls' would be retrieved without taking into account the material, i.e. 'stone walls', which is the information actually needed. Related to this, this layer also implements linguistic operations such as stemming or lemmatization. These operations can improve recall by reducing inflected or derived words to their root (e.g. the word 'policies' can be reduced to 'policy').

Another important consideration is how query results are finally formatted and sent to the presentation layer in order to be delivered to the user. The format needs to be supported by different browsers or applications. JSON format is the data exchange format in MongoDB, which is used for the deployment of the SHERPA data repository, but others (e.g. XML) could also be created, in the application layer, depending on HTTP headers.

5.4. Presentation layer

5.4.1. Overview of CRUD operations

The acronym CRUD stands for **Create**, **Retrieve**, **Update** and **Delete**; a list of terms describing the kinds of operations that can be performed when interacting with database systems. More specifically, the **Creation** operation relates to the creation of data records that will be stored in the database and can be retrieved upon user request. Creation of data records does not take place in an arbitrary way, but rather follows some rules. These rules are imposed by the underlying data model (namely, a blueprint illustrating, in a formal way, what pieces of data are stored in the database and relationships between them). Data that is stored in the database can be accessed and **retrieved** by users upon execution of queries. A well-known query language, used for accessing and retrieving data from Relational Database systems, is SQL. In the case of NoSQL data repository systems, which have gained momentum in the last years, access to data is facilitated through execution of application code. Finally, **Update** and **Delete** are database operations that result to changes in stored data. Therefore, they should be implemented with caution so as to avoid unexpected and harmful effects. Request for user confirmation prior to the execution of these operations can help towards avoiding unintentional data loss. All contemporary database management systems have built-in mechanisms for 'secure' implementation of these operations.

5.4.2. SHERPA online repository functionalities

Taking account of the overview of database-related CRUD operations, the SHERPA online repository system is going to deliver the following list of functionalities:

- **Creation of new user account**

SHERPA consortium members, SHERPA network members and SHERPA MAP members will be able to create user accounts and, therefore, gain access to the whole spectrum of functionalities provided to them. Creation of a new user account will involve, among others, the definition of username and password that are going to be used as credentials for authentication. When registering to the SHERPA online repository system, users will also declare the category (i.e. end-user type[16]) to which they belong. By this way, they will be able to access the entire set of functionalities made available to the respective end-user type. Other user types who belong to the 'members of the wider community', apart from SHERPA MAP members, will be able to access information without having to create an account.

- **User authentication**

Users (i.e. SHERPA consortium members, SHERPA network members and SHERPA MAP members) can log-in to the SHERPA online repository by providing their unique credentials through a fit-for-purpose interface.

- **Change of authentication credentials**

Users (i.e. SHERPA consortium members, SHERPA network members and SHERPA MAP members) can change their username or password though an appropriately designed interface.

- **Search for information**

All types of end-users will be able to search for information through use of appropriately designed interfaces. Search may take place either by typing queries in a search bar or through use of search zones (i.e. predefined search options). Search for information may relate to rural topics, projects, summaries of topic-related results per project, summaries of topic-related results (from a bulk of relevant projects), or SHERPA Papers.

[16] Definition and description of the end-user types considered in SHERPA are made available in Section 3 and, more specifically, Sub-section 3.1.

- **Request for generation of statistics-/indicators-/project classification – related results**

Users of all types will be able to submit requests for automatic generation of: (i) statistics and/or indicators about research activity in one or more rural topics, and (ii) tables presenting project classifications with regard to sub-topics and produced, by research projects, types of results.

- **Configuration of display of retrieved search results**

After execution of a search query, users are being delivered with the results of their search. In this context, users will be provided with the option to use filters and configure the type and amount of information to be displayed as part of query results delivery. This functionality will also be available for display of results of user queries relating to generation of statistics-/indicators-/project classification – related information.

- **Creation of new data records**

Users will be able to create new data records through use of appropriately designed interfaces. New records may relate to rural topics, projects, summaries of topic-related results per project, summaries of topic-related results (from a bulk of relevant projects), or SHERPA Papers.

- **Update of data records**

Users will be able to update data records through use of the appropriate interfaces. Updates to data records can relate to rural topics, projects, summaries of topic-related results per project, summaries of topic-related results (from a bulk of relevant projects), or SHERPA Papers.

- **Deletion of data records**

Users will be able to delete existing data records through use of the appropriate interfaces. The deletion of data records can relate to rural topics, projects, summaries of topic-related results per project, summaries of topic-related results (from a bulk of relevant projects), or SHERPA Papers.

5.4.3. System functionalities provided to each end-user type

Based on the system functionalities defined above and user need descriptions provided in Sub-section 3.2, an alignment between the targeted end-user categories, and types, and the functionalities of the SHERPA online repository system that are being provided to them Table 7 below provides.

Table 7: Categories and types of SHERPA system end-users, needs and provided system functionalities

Category of End-user Type	End-user Type	Provided System Functionalities
SHERPA consortium member	**Scientific Editor** **Review Editor** **Communication Editor** **Support Staff** **MAP Facilitator** **MAP Monitor** **SHERPA data repository Administrator** **Other consortium member**	Creation of new user account User authentication Change of authentication credentials Search for information Request for generation of statistics-/indicators-/project classification – related results Configuration of display of retrieved search results Creation of new data records Update of data records Deletion of data records

Category of End-user Type	End-user Type	Provided System Functionalities
SHERPA network member	**Project consortium member** **Project coordinator**	Creation of new user account User authentication Change of authentication credentials Search for information Request for generation of statistics-/indicators-/project classification – related results Configuration of display of retrieved search results Creation of new data records Update of data records Deletion of data records
Member of the wider community	**SHERPA MAP member**	Creation of new user account User authentication Change of log-in credentials Search for information Request for generation of statistics-/indicators-/project classification – related results Configuration of display of retrieved search results
	Society member	Search for information Request for generation of statistics-/indicators-/project classification – related results Configuration of display of retrieved search results

5.5. Communication between layers

In multitier architectures, most of the communication between different layers is made through APIs. APIs have a number of features that make them valuable and useful as they reduce the complexity of connector semantics. For example, they comply with standards such as HTTP, which are easily accessible, through the Internet, and developer-friendly. In the case of the SHERPA online repository system, the Representational state transfer (REST) API style (Murphy et al., 2018) will be used. REST is a software architectural style that defines a set of constraints for creating APIs within the web context. RESTful APIs provide interoperability between computer systems on the Internet allowing the access, creation and modification of digital resources by using a uniform and predefined set of stateless operations that increase the scalability of the final application.

6. Data security mechanisms

Deployment of the SHERPA online repository requires ensuring the integrity and security of data stored. Banker *et al*. (2016) describe the deployment of database systems in secure environments, encryption of network traffic, authentication and authorisation as core aspects of data security. Secure environments can be all but guaranteed by security mechanisms of all contemporary operating systems, with a firewall being one of the most important mechanisms. In addition to having secure environments for running database systems, the encryption of network traffic is also critical to avoiding malicious network activity. Encryption mechanisms prevent the decryption of messages exchanged between the client and the server by agents that may attempt to monitor database-related traffic. MongoDB comes with the TLS/SSL (Transport Layer Security/Secure Sockets Layer[17]) built-in library that handles the encryption of network traffic.

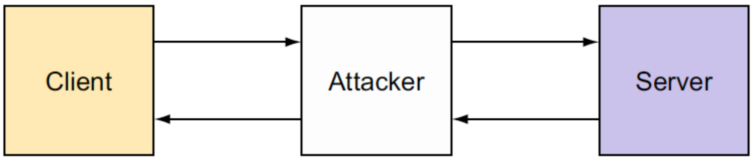


Figure 8: The 'man-in-the-middle-attack' malicious activity (source: Banker *et al.*, 2016)

Service authentication adds another security layer to that offered by network traffic encryption. Specifically, service authentication is a verification means for applications that try to establish a connection to the database. Such a verification mechanism has the ability to prevent types of malicious activity such as the so called 'man-in-the-middle attack' (Banker *et al.*, 2016). In such a case, a malicious application connects to both the client and the server, and by establishing connection is able to decrypt and encrypt messages exchanged between the client and the server, as well as send arbitrary messages to both sides. Figure 8 above provides a graphic depiction of this type of malicious activity. Service authentication can be provided through the issue of authentication certificates by third parties, called certificate authorities[18], which are able to verify that the application attempting to establish connection can be trusted. As MongoDB states on its official website, it "*supports a number of authentication mechanisms that clients can use to verify their identity*". SCRAM[19] is the authentication mechanism used, by default, by MongoDB and, thus, by the SHERPA repository.

Apart from provision of authentication mechanisms on the service level, MongoDB also enables the authentication of users. User authentication is associated with the assignment of roles to users and, thus, permission of specific operation sets per role. MongoDB provides a number of built-in roles and supports the configuration of fit-for-purpose roles. It enables access to data and database-related operations through role-based authorisation mechanisms and, for this purpose, supports a range of roles associated with different levels of access to the database system. In the context of the MongoDB ecosystem, a 'role' is conceptualised as a set of 'privileges' with the term 'privilege' being described as consisting of "*a specified resource*[20] *and the actions permitted on the resource.*"[21] MongoDB enables the following list of resource-related actions[22]:

- Query and Write Actions
- Database Management Actions
- Deployment Management Actions
- Change Stream Actions
- Replication Actions
- Sharding Actions
- Server Administration Actions
- Session Actions
- Free Monitoring Actions
- Diagnostic Actions
- Internal Actions

[17] https://docs.mongodb.com/manual/core/security-transport-encryption/

[18] A certificate authority can be either an organization or automated mechanisms/tools.

[19] https://docs.mongodb.com/manual/core/security-scram/#authentication-scram

[20] In MongoDB, a resource can be a collection, a database or a cluster. Further documentation is provided by MongoDB's official website (https://docs.mongodb.com/manual/reference/resource-document/#resource-document).

[21] https://docs.mongodb.com/manual/core/authorization/

[22] https://docs.mongodb.com/manual/reference/privilege-actions/#security-user-actions

Descriptions of roles built-in to MongoDB which are considered as relevant for the needs of the SHERPA repository design are summarised in Table 8. These roles are across the 'Database User Roles' and 'Database Administration Roles' categories. Detailed documentation on MongoDB roles and their characteristics is provided on MongoDB's official website[23].

Table 8: Description of roles built into MongoDB relevant to the design of the SHERPA online repository.

MongoDB role category	MongoDB built-role	role description
Database User Roles	**read**	Role granted permission to access and read data from non-system collections[24].
	readWrite	Role granted permission to read and modify (i.e. update and delete) data on all non-system collections.
Database Administration Roles	**dbAdmin**	Role granted permission to perform administrative tasks, such as schema-related tasks, indexing and collection of statistics. It has no privileges for user/role management.
	dbOwner	Role granted permission to perform any administrative action on the database. Privileges provided to this role are a combination of those of the readWrite, dbAdmin and userAdmin roles.
	userAdmin	Role granted superuser access to the database system. It has the permission to create new and modify existing users in the database system.

Alignment of the roles defined in the context of the SHERPA online repository ecosystem and MongoDB built-in roles (and role categories) is presented in Table 9.

Table 9: Roles for operation and use of the SHERPA online repository, and alignment with roles and role categories built-in to MongoDB.

Category of SHERPA end-user types	SHERPA end-user type	MongoDB built-in role	MongoDB built-in role category
SHERPA consortium member	Scientific Editor	readWrite	Database User Role
	Review Editor	readWrite	Database User Role
	Communication Editor	readWrite	Database User Role
	Support Staff	readWrite	Database User Role
	MAP Facilitator	readWrite	Database User Role
	MAP Monitor	readWrite	Database User Role
	SHERPA data repository Administrator	dbOwner	Database Administration Role
	Other consortium member	readWrite	Database User Role

[23] https://docs.mongodb.com/manual/reference/built-in-roles/

[24] System collections are collections stored into the admin database. The purpose of the admin database is to store system-related information, as well as user authentication and authorisation data (e.g. admin and user usernames, passwords and roles).

Category of SHERPA end-user types	SHERPA end-user type	MongoDB built-in role	MongoDB built-in role category
SHERPA network member	Project consortium member	readWrite	Database User Role
	Project coordinator	readWrite	Database User Role
Member of the wider community	SHERPA MAP member	read	Database User Role
	Society member	read	Database User Role

Apart from technical details provided by the technology employed, the topic of data security is also of relevance to management and safety of sensitive personal data of registered users. Personal data, such as first and last names, valid email accounts, usernames and passwords, need to be provided as part of processes of user registration and authentication/authorisation. The collection and storage or management of this type of data will be handled in full compliance with GDPR regulations. From a technical perspective, personal data, related to users, is stored in different collections, termed as system collections in the MongoDB ecosystem, accessed only by the SHERPA data repository Administrator. Details about the Data Management Plan of the SHERPA project are provided in Deliverable 1.3 ('**Data Management Plan**').

7. Towards a sustainability plan for the SHERPA online repository

Developing a sustainable future for the SHERPA online repository system and the community built around it means to enabling it to continue beyond the end of the project's lifecycle. According to the Cambridge online Dictionary[25], 'sustainability' can be generically defined as "*the quality of being able to continue over a period of time*". However, in the software engineering context and, more specifically, from the perspective of software application deployment and use, 'sustainability' is more closely related to the capacity to "*maintain and evolve a software system with minimized environmental impact, a sufficient economic balance, and social responsibility*." (Penzenstadler, 2013). Therefore, when it comes to defining sustainability for software products, there are a number of parameters which need to be considered (e.g. environmental impact, economic balance and social responsibility).

Making the SHERPA online repository system sustainable necessitates the development of a clear, well-documented plan. In this context, the aim of this section is to identify issues relating to the development of the SHERPA online repository sustainability plan, and to present potential approaches. Sustainability needs to be considered in terms of the resources needed to keep the system running and the surrounding community served. These resources relate to technological infrastructure and to human and financial capital required.

Development of the SHERPA online repository will be based on an open source technology stack and a neat, tier-based design able to facilitate maintenance after the project ends. The exploitation of open source technologies together with the provision of a well-documented design constitute the baseline for a long-term and cost-efficient system maintenance. However, apart from framework- and tool-related solutions adopted, maintenance, and thus sustainability, is also related to the infrastructure that will be used for hosting the SHERPA repository system after completion of the project.

[25] https://dictionary.cambridge.org/

In the case of system hosting services there are two potential options:

- outsourcing hosting to a cloud-service provider;
- in-house hosting.

As far as the first option is concerned, a number of cloud-based service providers can be considered with regard to offered solutions and costs. In this case, a significant is that all hardware maintenance and upgrade costs are on the side of the provider. If the in-house system hosting solution is selected, all software and hardware maintenance or upgrade related tasks and costs need to be covered by the organisation responsible. Balancing options can help towards making informed decisions which lead to solutions with minimised environmental impact (e.g. through the use of modern hardware systems leaving a minimal energy footprint), as well as balanced economic performance and social responsibility (via easy to use resource-efficient services).

Assuming stakeholders are interested in continuing the provision of the SHERPA services and outputs then issues arising of the human and financial resources required are also significant when considering alternative solutions. An initial solution that could be explored is to make the SHERPA outputs available through open repositories like Zenodo[26]. Zenodo is a service provided for free use of enabling access to information consistent with the FAIR principles outlined in Deliverable 1.3 ('**Data Management Plan**'). Similarly, synergies will be sought with other projects interested in making the outputs of SHERPA available through their digital collections, could be sought. Such a project is EUREKA[27], which aims to develop a repository of digital knowledge objects created by Multi-Actor projects. Towards the end of the SHERPA project, new initiatives could be launched which are suitable for the storage of outputs created by SHERPA, and/or utilising its technological infrastructure. Such an initiative could be in the form of a new project funded under the Horizon Europe Programme (2021 - 2027). In such a case, SHERPA's legacy infrastructure and contents could be transferred to the interested stakeholder, while protecting the legacy Intellectual Property Rights of the SHERPA project and the repository development teams.

Within the SHERPA project, costs for service provision and maintenance are covered by the SHERPA project budget of the relevant partner (AUA). Those would not be available for a legacy arrangement. So, if there was interest from SHERPA partner organisations, or organisations within their networks to take on, post-SHERPA suitable resourcing would be required. An organisation planning to provide SHERPA-related services would be required to develop a clear plan for the resourcing of a legacy facility. Resource estimates would require to cover the needs of software and hardware, running, maintenance and upgrading facilities, community support, and the mechanisms for the execution of processes for the extraction and provision of new results. Such a plan would require close collaboration with the SHERPA Coordinator and relevant partners to ensure the protection of Intellectual Property Rights, and appropriate protection of SHERPA rights (e.g. project and partner reputations).

Opportunities for all possible arrangements for legacy operation of the SHERPA repository will remain under review throughout the life of the SHERPA project.

8. Discussion and conclusions

The aim of this document has been to provide a detailed description of the SHERPA online repository system's architecture and technical specifications. In an attempt to conceptualise the system's design in a holistic way, design-related aspects and considerations have been provided with regard to the data that will be stored into the repository, end-user type profiles and needs, the system's architecture and embedded modules, the data model employed, as well as functionalities of the system and alignment of those with the targeted end-user types. The critical issue of data security has also been highlighted by providing descriptions of the mechanisms

[26] https://zenodo.org/
[27] https://www.h2020eureka.eu/

that are built to MongoDB, for the detection of malicious network activity, and will be deployed as part of the SHERPA online repository system's function. In addition to that, details about authentication and authorisation of users together with permission and access rights to the system have been made available. Finally, aspects related to a plan for making the SHERPA online repository system sustainable have been stressed. A potential interest in having the system's services provided after the project's completion should be investigated on the basis of a well-structured and documented plan. Such a plan should focus on a balanced allocation of required resources, as well as on strict mechanisms for the protection of Intellectual Property Rights.

The design that has been presented and documented in this deliverable provides the blueprints needed for the SHERPA online repository system's development. However, it needs to be stressed that the development of the system is going to take place in the context of an agile approach. This means that as the work related to the system's development will evolve, new needs, not anticipated from the beginning, may emerge and, as a result, further design-related requirements may come into play. Therefore, any changes in design-related requirements will feed any necessary iterations and serve as input to any revisions proposed. Design updates will be appropriately documented and a log of versions of the design will be thoroughly kept so as to efficiently track changes. Changes, if any, will mostly relate to the application and presentation layers and will not affect the data persistence layer of the system. They will take the form of slight design adaptations considered with the aim to adequately meet any emerging requirements. Such adaptations will not have any significant impact on the basic design principles that have been presented in this document.

9. References

Banker, K. (2012). *MongoDB in action*. Manning Publications Co., New York, USA.

Banker, K. et al. (2016). *MongoDB in action (2nd edition)*. Manning Publications Co., New York, USA.

Chartier, O., Salle, E., Miller, D. and Martino, G. (2020). Deliverable 1.2 Working Principles of the Multi-Actor Platforms, SHERPA Project, Report to the European Commission. pp. 19.

Chartier, C., Salle, E., Miller, D. and Panoutsopoulos, H. 2020. Data Management Plan, D1.3. Sustainable Hub to Engage into Rural Policies with Actors (SHERPA), Report to the European Union, pp. 23.

Cattell, R. (2010). Scalable SQL and NoSQL data stores. *Acm Sigmod Record 39* (4), 12-27.

Lake, P., & Crowther, P. (2013). *Concise guide to databases*. Springer, London, UK.

Lindley, C. (2019). *Front-end Developer Handbook*.

Manning, C., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, UK.

Murphy L., Kery M. B., O. Alliyu, Macvean A. & Myers B. A. (2018). API Designers in the Field: Design Practices and Challenges for Creating Usable APIs in *Proceedings of 2018 IEEE Symposium on Visual Languages and Human-Centric Computing* (VL/HCC), Lisbon, 249-258.

Penzenstadler, B. (2013). Towards a definition of sustainability in and for software engineering. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing* (pp. 1183-1185).

Tiwari, S. (2011). *Professional NoSQL*. John Wiley & Sons, Indianapolis IN, USA.

Vaish, G. (2013). *Getting started with NoSQL*. Packt Publishing Ltd, Birmingham, UK.

Wilkinson, M. D., *et al*. (2016). The FAIR Guiding Principles for scientific data management and stewardship. *Scientific data*, *3*.

Annex

Figure 9 shows the SHERPA database model, which illustrates how data is stored into the SHERPA repository with the help of JSON documents, and references from one JSON document to another.

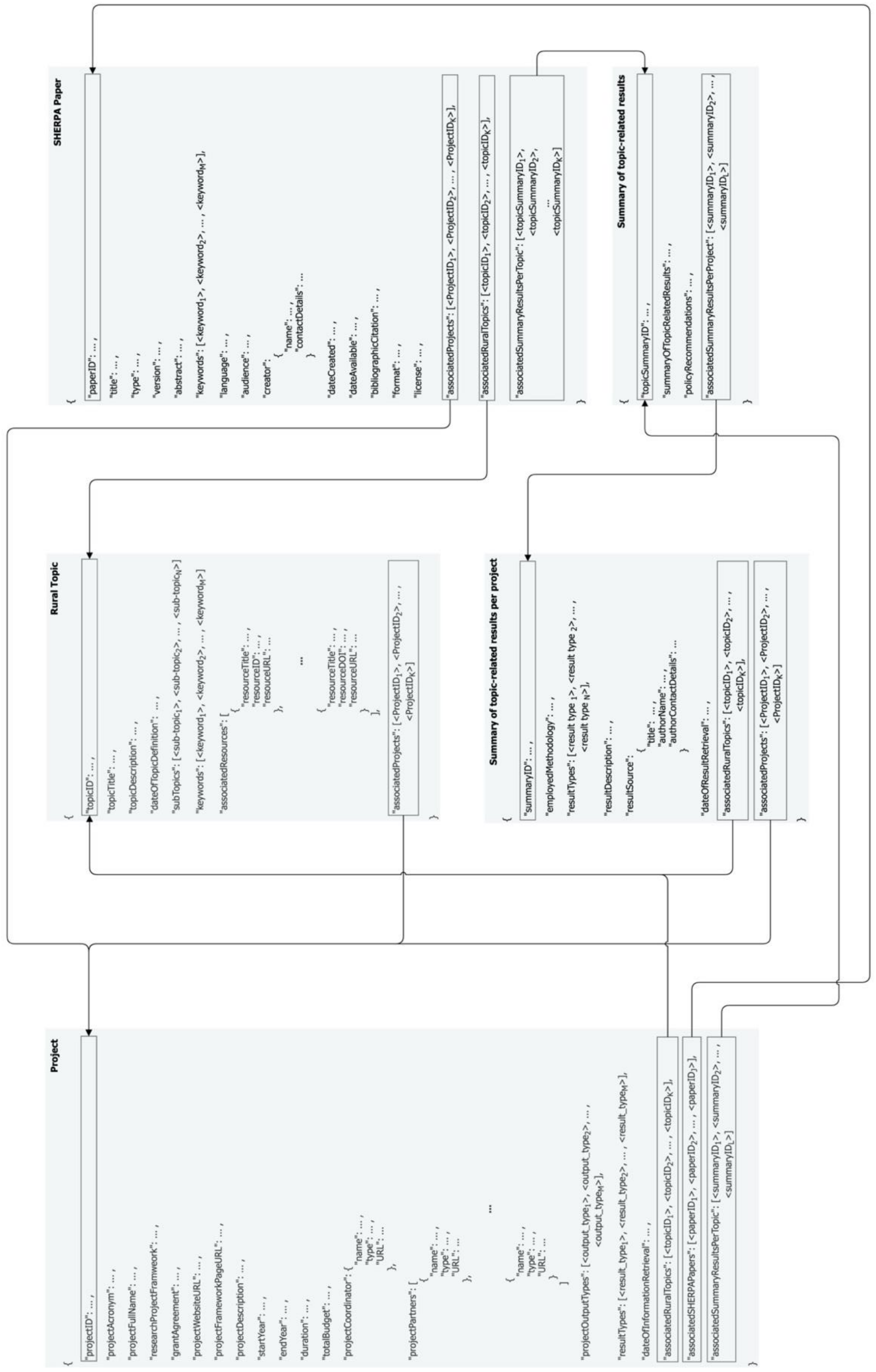

SHERPA Paper
{
"paperID": … ,
"title": … ,
"type": … ,
"version": … ,
"abstract": … ,
"keywords": [<keyword$_1$>, <keyword$_2$>, … , <keyword$_M$>],
"language": … ,
"audience": … ,
"creator":
{
"name": … ,
"contactDetails": …
}
"dateCreated": … ,
"dateAvailable": … ,
"bibliographicCitation": … ,
"format": … ,
"license": … ,
"associatedProjects": [<ProjectID$_1$>, <ProjectID$_2$>, … , <ProjectID$_K$>],
"associatedRuralTopics": [<topicID$_1$>, <topicID$_2$>, … , <topicID$_K$>],
"associatedSummaryResultsPerTopic": [<topicSummaryID$_1$>,
<topicSummaryID$_2$>,
…
<topicSummaryID$_K$>]
}
Summary of topic-related results
{
"topicSummaryID": … ,
"summaryOfTopicRelatedResults": … ,
"policyRecommendations": … ,
"associatedSummaryResultsPerProject": [<summaryID$_1$>, <summaryID$_2$>, … ,
<summaryID$_L$>]
}
Rural Topic
{
"topicID": … ,
"topicTitle": … ,
"topicDescription": … ,
"dateOfTopicDefinition": … ,
"subTopics": [<sub-topic$_1$>, <sub-topic$_2$>, … , <sub-topic$_N$>]
"keywords": [<keyword$_1$>, <keyword$_2$>, … , <keyword$_M$>]
"associatedResources": [
{
"resourceTitle": … ,
"resourceID": … ,
"resouceURL": …
},
…
{
"resourceTitle": … ,
"resourceDOI": … ,
"resourceURL": …
}
],
"associatedProjects": [<ProjectID$_1$>, <ProjectID$_2$>, … ,
<ProjectID$_K$>]
}
Summary of topic-related results per project
{
"summaryID": … ,
"employedMethodology": … ,
"resultTypes": [<result type $_1$>, <result type $_2$>, … ,
<result type $_N$>],
"resultDescription": … ,
"resultSource":
{
"title": … ,
"authorName": … ,
"authorContactDetails": …
}
"dateOfResultRetrieval": … ,
"associatedRuralTopics": [<topicID$_1$>, <topicID$_2$>, … ,
<topicID$_K$>],
"associatedProjects": [<ProjectID$_1$>, <ProjectID$_2$>, … ,
<ProjectID$_K$>]
}
Project
{
"projectID": … ,
"projectAcronym": … ,
"projectFullName": … ,
"researchProjectFramweork": … ,
"grantAgreement": … ,
"projectWebsiteURL": … ,
"projectFrameworkPageURL": … ,
"projectDescription": … ,
"startYear": … ,
"endYear": … ,
"duration": … ,
"totalBudget": … ,
"projectCoordinator": {
"name": … ,
"type": … ,
"URL": …
},
"projectPartners": [
{
"name": … ,
"type": … ,
"URL": …
},
…
{
"name": … ,
"type": … ,
"URL": …
}
]
"projectOutputTypes": [<output_type$_1$>, <output_type$_2$>, … ,
<output_type$_M$>],
"resultTypes": [<result_type$_1$>, <result_type$_2$>, … , <result_type$_M$>],
"dateOfInformationRetrieval": … ,
"associatedRuralTopics": [<topicID$_1$>, <topicID$_2$>, … , <topicID$_K$>],
"associatedSHERPAPapers": [<paperID$_1$>, <paperID$_2$>, … , <paperID$_J$>],
"associatedSummaryResultsPerTopic": [<summaryID$_1$>, <summaryID$_2$>, … ,
<summaryID$_L$>]
}

Figure 9: SHERPA database model

SHERPA

Rural Science-Society-Policy
Interfaces