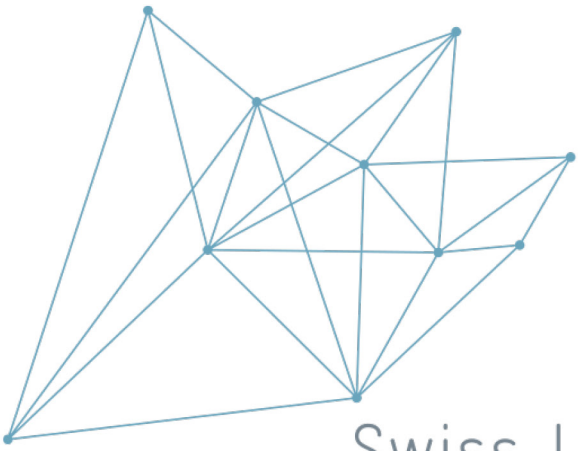


Swiss Learning Health System

Swiss Health Metadata Repository: implementation recommendations

Requirements, recommended architecture and prototype implementation

Alessio De Santo, Eliane Maalouf, Paul Cotofrei, Kilian Stoffel

SLHS Policy Brief No. 1

Table of Contents

List of Abbreviations 5
Key Messages 6
Executive Summary 9
Background and Context 14
Stakeholders' requirements 16
Users' personas 17
Tasks 18
Environments 20
Technical Analysis 21
Architecture 21
Potential solutions 26
Interactive prototype implementation 29
Technological choices 29
Proof-of-Concept and Usability tests 33
Discussions and outlook 36
From prototype to production 36
Major risks and potential mitigation 38
References 39
Appendix I 41
Detailed functionalities list 41
Appendix II A 44
First iteration prototype mockups 44
Appendix II B 47
Proof-of-concept prototype 47
Appendix III A 49
Additional wished functionalities and known bugs 49
Appendix III B 52
Augmented set of metadata attributes 52
Appendix IV 54
Improved prototype mockups 54

Figure 1. The Steps of Design Science Research Methodology [7] 14
Figure 2. Schematic of a proposed system architecture 22
Figure 3. Model-View-Controller architectural design pattern 24
Figure 4. Entity Relation diagram for users' database 29
Figure 5. The links connecting the entities of the metadata repository 30
Figure 6. Main window of SH Metadata Repository 44
Figure 7. Text view of search results 44
Figure 8. Graph view of search results 45
Figure 9. User's credential view 45
Figure 10. Metadata template window 46
Figure 11. Current main window of SH Metadata Repository 47
Figure 12. Textual and graphical view of search results 47
Figure 13. Current metadata template window 48
Figure 14. Improved main window of SH Metadata Repository 54
Figure 15. Improved textual and graphical view of search results 55
Figure 16. Improved metadata template window 55

List of Abbreviations

HIS	Health Information System
MVC	Model-View-Controller
SHIS	Swiss Health Information System
SHS	Swiss Health System
URI	Universal Resource Identifier
URL	Universal Resource Location
WS	Web Services
WSDL	Web Services Description Languages
W3C	World Wide Web Consortium

Key Messages

The Challenge

In the context of a decentralized approach for health data resources, a central metadata repository becomes essential for the identification/description of the necessary resources. A Swiss Health Metadata Repository is expected to enable multiple advantages, namely: to provide a single entry point for searching/retrieving health-related resources; to help identify possible semantic linking between data sources; to provide a consistent data catalog ensured by the use of standardized vocabularies/ontologies; to enable potential exchange of experience and know-how between health data related projects in Switzerland; and to increase the capacity of research groups to share/access/analyse health related data.

This policy brief focus on the implementation aspects of a proof-of-concept prototype for Swiss Health Metadata Repository.

Options to address the challenge

A user-centered design process is initiated to identify the users' requirements as a list of specifications that will orient the technical analysis and the system architecture. Due to the heterogeneity of the users interacting with the metadata repository, the design cycle is focusing only on a particular target group (mostly concerned in searching for and accessing data resources), the health researchers' user group.

The design process identified four *personas*, representing specific user profiles playing particular roles when interacting with the system: (1) *health researcher, (2) metadata creator, (3) collection curator*, and (4) *repository administrator*.

Three specific user's *scenario*, representing a minimal set of tasks enabling a specific personas to achieve a particular objective, were identified: (1) *creating and editing metadata resources*; (2) *browsing and curating metadata repository*; and (3) *managing the repository and the metadata templates*.

Finally, the design process considers two categories of environments where the task reffered by users' scenarios are to be perfomed: (1) *digital environments*, (e.g. desktop or mobile interfaces); and (2) *physical environments* (e.g. offices).

The architecture of SH Metadata Repository is designed as a flexible structure satisfying several objectives: searching for specific data resources, adding new information and editing existing ones, as well as establishing relations between the various resources on the basis of similarities in their metadata.

The proposed system architecture is a 3-tiers model (each tier representing a layer in the system that can be managed and secured separately), divided into:

- *Data tier* - designed to store and retrieve data from different categories of databases with particular storing mechanism, information type and management policy. Four databased are identified as essential for the system's architecture: (1) *users* (relational database), (2) *metadata descriptions* (graph database), (3) *metadata templates* (relational database), and (4) *controlled vocabularies* (graph database).

- *Logic tier* – structured as a web application using Model-View-Control design pattern, manages the core of the application and contains the code of the functions satisfying the functional requirements.
- *Presentation tier* - provides user interface components that allow users to interact with the application, in the context of the three specific users' scenarios.

Implementation Considerations

The implementation of the proposed architecture is analysed from two complementary approaches: adapting an existing, open-source repository plateform (CKAN, Zenodo, CEDAR) or using a tailor-made solution. The advantages of a tailor-made solution, but integrating relevant components and tools from existing systems, are manifolds:

- the metadata search can be tailored to the researchers needs and expected ways of looking and processing information;
- tools to enforce annotating metadata with standard vocabularies and ontologies can be put in place;
- tools for a richer exploration of the available data through richer metadata could be implemented and tested.

Based on a first design cycle iteration, a proof-of-concept prototype of SH Metadata Repository was developed and implemented. This functional, first minimum viable solution is accessible throught Internet at the address http://imi-slhs.unine.ch/. The metadata repository, which is the core of the implemented prototype, is managed through a graph database, using a dynamic schema based on four entities: *Dataset, Organisation, Person*, and *Tag*. The core set of the Dublin Core metadata schema has been adapted to the context of SHIS and implemented as metadata descriptions stored in *Dataset* entities. The links relating the four entities allow the navigation, the discovery and sorting of *Datasets*.

The Data layer implements the graph database management Neo4j. The Logic layer uses Python language coupled with Flask framework to implement the application code. Finally, the Presentation layer is based on the dynamic, browser-based visualization library vis.js, a Javascript library built on the top of the Web application built with Python and Flask.

Opprtunities and barriers

To evaluate the utility and the potential gaps and shortages of this proof-of-concept prototype, usability tests, under the form of semi-structured interviews, have been conducted with a target group of SLHS researchers. Based on the conclusions rised from these interviews, an improved prototype mockups was created, tackling three main key aspects: (1) *the onboarding experience;* (2) *the visualization of the search result* and (3) *the dataset metadata creation*.

However, several limitations of the current prototype were identified, restricting the full potential of the system:

1) *Available data collection* - the current content of the metadata repository is too limited and impact the usefulness of the application; *potential solution* – the implementation of (semi-)automated tools to integrate existing metadata from curated data providers and repositories;
2) *Metadata attributes* - the current set of metadata attributes was deemed insufficient for practical use, particularly due to a less intuitive meanings; *potential solution* – the extention of the minimal set of attributes and the improvement of the user interface.

3) *Data standards* - an efficient integration of taxonomies and controlled vocabularies is not yet done; *potential solution* – integration of annotations and autocomplete functionality from an ontology portal like Bioportal.
4) *Functionalities and technical ecosystem* - not all the proof-of-concepts's technical aspects have been implemented at their full potential; *potential solution* – integration of CKAN as content management backend with CEDAR metadata authoring tools in the frontend.

The process of transforming a proof-of-concept prototype into a complete system in production rises a number of relevant risks for the SH Metadata Repository project. This list includes:

1) *Requirements shifts* - the risk that new requirements may arise during the project development; *potential measure* - an iterative/incremental approach to the project development and frequent contacts with users and stakeholders.
2) *Technical difficulties* – the risk of lack of knowledge of IT staff for the selected technologies; *potential measure* – an in-depth evaluation and comparison of existing tools and their ease of integration, and a continuous training.
3) *Poor management and budgeting* – the risk of poor project management practices and lack of long term vision; *potential measure* – apply agile project management methodologies.

These risks, and potentially others, need to be addressed by a risk management plan at the beginning of a project dedicated to put the SH Metadata Repository in production.

Executive Summary

Bakground and Context

In the context of a decentralized approach for health data resources, a metadata management infrastructure becomes crucial in order to guarantee that the necessary resources are retrieved and accessed from the right databases and respecting each dataset's specificities. Hence, a central metadata repository, designed as the main element of the metadata management infrastructure, becomes essential for the identification/description of health data sources. Metadata repositories were identified to be central building blocks in multiple initiatives aiming to facilitate health data access, linking and analysis.

A Swiss Health Metadata Repository is expected to enable multiple advantages, namely: to provide a single entry point for searching/retrieving health-related resources; to help identify possible semantic linking between data sources; to provide a consistent data catalog ensured by the use of standardized vocabularies/ontologies; to enable potential exchange of experience and know-how between health data related projects in Switzerland; and to increase the capacity of research groups to share/access/analyse health related data.

Stakeholders' requirements

Following the design science research methodology, this policy brief discuss the implementation aspects of the system (denoted SH Metadata Repository). Concretizing the first step of the methodology, the requirements definition process exploits two complementary approaches to extracts functional requirement as a list of specifications that will orient the technical analysis and the system architecture. The first approach is based on analysing a design process documentation of an existing system (CEDAR) with similar objectives. The second approach complements the first one with observations extracted from semistructured interviews conducted with potential users from a target group.

In a preliminary analysis, the user centered design process distinguished two (overlapping, but not identical) groups of persons with an interest in creating and sustaining a SH Metadata Repository: *stakeholders* and *users*. The design cycle concentrates on the users, i.e. organizations or individuals that need to interact with the metadata repository in order to access specific resources. As the users, globally, represent a heterogenous group, it was decided to instanciate the design cycle by focusing only on the health researchers' user group. This target group is mostly concerned in searching for and accessing data resources; moreover, these particular users generate a large amount of data resources that require documentation in terms of metadata descriptions.

The design process identified four *personas*, representing specific user profiles playing particular roles when interacting with the system: *health researcher, metadata creator, collection curator* and *repository administrator*. Three specific user's scenario, representing a minimal set of tasks enabling a specific personas to achieve a particular objective, were identified: (1) *creating and editing metadata resources*, (2) *browsing and curating metadata repository*, and (3) *managing the repository and the metadata templates*. Finally, the design process considers two categories of environments where the task reffered by users' scenarios are to be perfomed: *digital environments*, concerning the digital plateforms (e.g. desktop or mobile interfaces) accessed by the users to perform their interactions with the system; *physical environments*, concerning the surrounding physical conditions manifested where the users intend to interact with the system (e.g. offices).

Technical Analysis

The architecture of SH Metadata Repository should have a flexible structure satisfying several objectives: searching for specific data resources, adding new information and editing existing ones, as well as establishing relations between the various resources on the basis of similarities in their metadata.

The proposed architecture is a 3-tiers model (each tier representing a layer in the system that can be managed and secured separately) divided into (1) *Presentation tier*, (2) *Logic tier*, and (3) *Data tier*.

The Data layer is designed to store and retrieve data from databases in order to be processed by the Logic layer. The layer allows to call different categories of databases, with particular storing mechanism, information type and management mechanism. The relational databases use static data schema to minimize data redundancy and prevent data inconsistencies. The non-relational databases use a dynamic schema for unstructured data and apply different storing mechanism: column-oriented, document-oriented, graph-based or KeyValue. In particular, graph-databases use a graph structure composed of nodes linked by edges to integrate and query the data. It is particularly useful for interconnected data items. In graph databases, edges representing relationships between the nodes are explicitly identified and can be queried and manipulated.

The design process identified four databases in the system's architecture: the templates database (relational), containing the predefined metadata templates; the users database (relational), containing the users profiles, their associated resources, and their authorizations; the metadata repository (graph), containing the actual metadata content; and the controlled vocabularies (graph), containing the predefined sets of values to be used for autocompletion and for semantic annotation.

The Logic tier contains the code that implements the functionalities of the application; in particular, exchanging data with the Data tier. To guarantee an easy and wide ranging access to the system, web applications and web browsers represent an optimal choice. Moreover, a web application for the metadata repository is also coherent with the digital environments that the target users group is very familiar with. The Model-View-Controller (MVC) programming design pattern is a popular pattern for designing web applications using n-tiers architectures. MVC allows to structure the logic tier such that all the heavy lifting logic and content rendering is done on the server side to allow for simpler client-side applications that only need to request the rendered content and display it to the user. The proposed architecture logic tier manages the core of the application and implements the functions satisfying the functional requirements identified during the users' requirements process.

The Presentation tier provides user interface components that allow users to interact with the application, depending on their role. This tier does not directly process data from databases nor handle business rules. It mainly consists of tools to render the relevant content and allow its display in a compelling and useful way to the user. In the proposed architecture the presentation tier would directly allow the user to act on the system in the context of the three specific users scenarios.

The analysis of several relevant existing platforms (CKAN, Zenodo, CEDAR) concerning their potential usefulness in implementing the proposed architecture is discussed in comparison to a tailor-made solution. Each such plateform presents components or tools satisfying particular requirements for SH Metadata Repository, but no one fulfills all the specifications. On the other hand, the advantages of a tailor-made solution are manifolds: the metadata search can be tailored to the researchers needs and expected ways of looking and processing information; tools to enforce annotating metadata with standard vocabularies and ontologies can be put in place;

tools for a richer exploration of the available data through richer metadata could be implemented and tested. In order to reduce the burden of a development from the scratch, a feasible option would be to actually identify and choose relevant components and tools from existing plateforms and combine them in a useful way. In that sense the tailor-made solution is not an effort to develop a solution from scratch but an effort to integrate multiple tools and functionalities instead with additional development time for non existing functionalities.

Interactive prototype implementation

Based on the conclusions of the first Policy Brief and of the corresponding Stakeholder Dialog, the design of a prototype mockup for SH Metadata Repository, under the form of a clickable model with pre-defined behavior, was proposed to SLHS users. The result of this first design cycle iteration concluded with a first consolidated list of functionalities. Based on this list, a proof-of-concept prototype was developed and implemented. In addition to the purely visual aspect and potential desired functionalities that the prototype made it possible to evaluate, the proof-of-concept is a functional, first minimum viable solution of a SH Metadata Repository. This prototype is accessible throught Internet at the address http://imi-slhs.unine.ch/.

The technological choices applied during the implementation phase were made in such a way that the prototype could be easily understood, taken up and maintained. Moreover, the selected technologies guarantee the evolution and adaptation to the business requirements of future iterations of the prototype.

The metadata repository, which is the core of the implemented prototype, is managed through a graph database, using a dynamic schema based on four entities: *Dataset, Organisation, Person*, and *Tag*. *Dataset,* which is the central entity of the SH Metadata Repository, *is Published* by an *Organization*, *is Created* by metadata repository users, represented by *Person,* and *is Categorized* by *Tags*, which are uniquely identified keywords associated with the dataset. The links relating the four entities allow the navigation, the discovery and sorting of *Datasets* through uniquely and unequivocal *Person*, *Organizations* and *Tags*.

The core set of the Dublin Core metadata schema has been adapted to the context of SHIS and implemented as metadata descriptions stored in *Dataset* entities. Yet, it was deemed insufficient for practical use and its attributes significance not necessarily intuitive.

To guarantee that every future instance of *Person, Organization* and *Tag* created in the metadata repository is uniquely identified, the prototype was designed to generate and link URIs with existing ones in the Linked Open Data. Thus, these instances can be linked to Datasets instances generated by a metadata creator, but also to external, uniquely identified online resources already existing in the Linked Open Data.

In the actual version of the proof-of-concept, the system automatically creates links between similar tags in order to allow the discovery of resources categorized by the same tag. However, this process could be largery improved by linking generated Tags to existing taxonomies and ontologies, thus allowing capturing the exact semantic meaning of each Tag instance.

The Data layer implements the graph database management Neo4j (Network Exploration and Optimization 4 Java). The Logic layer uses Python language coupled with Flask framework to implement the application code. Flask is a highly customizable development framework, allowing to structure the code according to the MVC pattern and facilitating the interfacing with databases of different kinds. Finally, the Presentation layer is based on the dynamic, browser-based visualization library vis.js, a Javascript library built on the top of the Web application built with Python and Flask.

As the system has to grow and evolve, successive design iterations would allow to refine and polish this first proposition that proves the feasibility and utility of the metadata repository concept. Concretely, to evaluate the utility and the potential gaps and shortages of this proof-of-concept prototype, usability tests have been conducted with stakeholders. The conclusions and recommendations raised after conducting semi-structured interviews with a target group of SLHS researchers, together with the complete protocol of the dialog, are presented in a second Stakeholder Dialog. Based on these recommendation (including a list of additional functionalities, counterintuitive interfaces and bugs that should be considered for future iterations), an improved prototype mockups was created, tackling three main key aspects: the onboarding experience, the visualization of the search result and the dataset metadata creation.

Discussions and outlook

The implementations aspects of the proof-of-concept prototype for Swiss Health Metadata Repository, based on a 3-tiers architecture for a metadata repository that satisfies health researchers' requirements, is the main focus of this policy brief. Several possible solutions showcasing new forms of interaction with the metadata were designed on the base of the recommendations concluding a series of usability tests under the form of semi-structured interviews with a target group of users.

However, several limitations of the current prototype were identified, restricting the full potential of the system.

- Firstly, the current content of the metadata repository is too limited and impact the usefulness of the application. Based on the feedback from the health researchers' community, a minimal set most relevant data sources for SLHS users must be considered. A realistic solution is the implementation of (semi-)automated tools to integrate existing metadata from curated data providers and repositories like SPHN, Federal Office of Statistics, FORS or others.
- Secondly, the current set of metadata attributes adapted from Dublin Core metadata schema was deemed insufficient for practical use, particularly due to a less intuitive meanings. Given the heterogeneity of the users, one solution will never fit all expectations. But a common denominator, as a minimal set of metadata attributes, could be identified and verified at future development iterations. Furthermore, improvements of the interface may help users whose needs cannot directly be met by an explicit attribute.
- Thirdly, an efficient integration of taxonomies and controlled vocabularies is not yet done, mainly due to their limited usability to annotate the current set of metadata attributes. These semantic resources are most commonly used at the level of the dataset items/variables coding, which requires an extension of the current set of attributes. To improve the metadata attributes' consistency, a possible solution is to integrate annotations and autocomplete functionality from an ontology portal like Bioportal.
- Lastly, due to time constraint, not all the proof-of-concepts's technical aspects have been implemented at their full potential. A tailor-made solution can be accelerated by opting to integrate parts of existing solutions satisfying the requirements and to develop from scratch only those functionalities that are not met by existing tools. Specifically, integrating CKAN as content management backend with CEDAR metadata authoring tools in the frontend, along with the graph based visualization, would make for a modular system that can easily be adapt to users' needs evolution.

The process of transforming a proof-of-concept prototype into a complete system in production rises a number of relevant risks for the SH Metadata Repository project. This list includes the risk of requirements shift, expressing that new requirements may arise during the project development. An iterative/incremental approach to the project development following the user-centered design cycle [1], and frequent contacts with users and stakeholders, can largely take those developments into account. A second risk is represented by technical difficulties that can stem from lack of knowledge of IT staff for the selected technologies or from a suboptimal choice of technological tools. To avoid such issues, the system designers need to spend time on evaluating and comparing existing tools and their ease of integration. Finally, the risk of poor management and budgeting, both in terms of time and finances, can negatively impact all aspects of the project. Even though agile project management methodologies reduce the risks related to uncertainty arising in the lifetime of a project, careful planning is still needed. This brings the importance of having a strong governing body in support of the development of the metadata repository that can provide the necessary resources, guarantee a long term vision and viability for the product and overall maintain an oversight and accountability of all involved parties.

These risks, and potentially others, need to be addressed by a risk management plan [23] at the beginning of a project dedicated to put the proposed metadata repository in production.

Background and Context

A Health Information System (HIS) represents the "set of components and procedures organized with the objective of generating information that will improve health management decisions at all levels of the health system" [2].To support health related operational and strategic decision making, a HIS must ensure that reliable and timely health information is available.

Our prior work in "Design principles of a central metadata repository as a key element of an integrated health information system" [3] established the need to design and implement a hybrid infrastructure for health data management in order to serve the particular specificities of the Swiss Health System. We distinguished three access modes in this infrastructure: (1) bridge access mode, for data stored in a persistent data-warehouse, (2) ferryboat access mode, for data stored in persistent local repositories, and (3) crane access mode, for data stored in temporal, local databases. We further established that a metadata management infrastructure becomes crucial in order to guarantee that the necessary data are retrieved and accessed from the right databases and respecting each dataset's specificities. Hence, a central metadata repository, designed as the main element of the metadata management infrastructure, becomes essential for the identification/description of health data sources. Metadata repositories were identified to be central building blocks in multiple initiatives aiming to facilitate health data access, linking and analysis ([4] [5] [6]).

In the continuity of this previous work, we focus in this policy brief on the implementation aspects of a Swiss Health Metadata Repository. In developing the prototype and writing this brief we followed the design science research methodology (DSRM) [7] [8]. DSRM proposes six steps depicted in Figure 1.

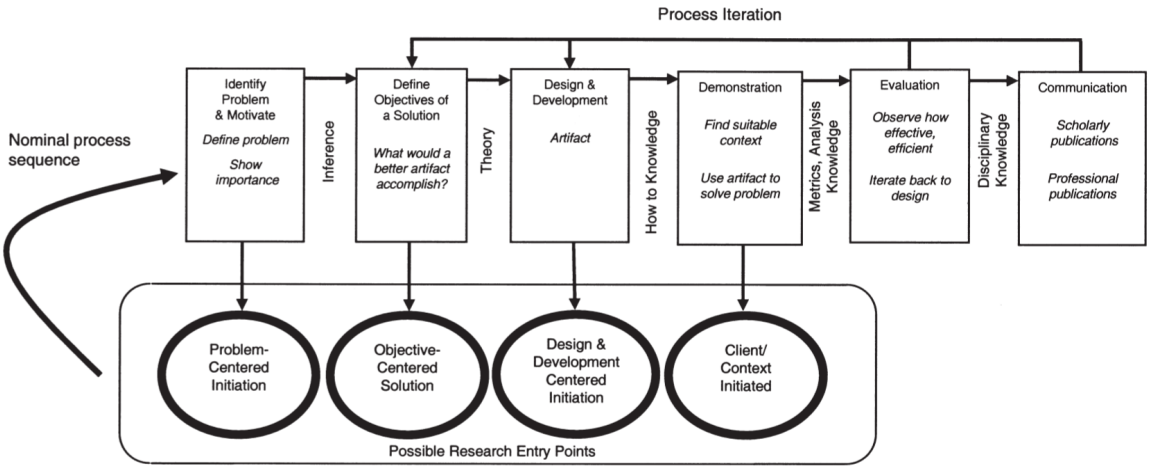


Figure 1. The Steps of Design Science Research Methodology [8]

The first step, treated in the following chapter "Stakeholders' requirements", is to identify the problem and the motivation. The second step, addressed in the chapter "Technical Analysis", is to define the objectives of a solution. The third and fourth steps correspond to the design and implementation of an artifact, deeply discussed in the chapter "Interactive prototype implementation". The fifth step is to evaluate the solution, as done through the semi-structured interviews,

described in [9] . Finally the sixth step is to communicate about the designed artifact, which is the main objective of this policy brief.

The system we envision by performing the analysis presented in this brief aims to enable multiple advantages, namely: to provide a single entry point for searching/retrieving health-related resources; to help identify possible semantic linking between data sources; to provide a consistent data catalog ensured by the use of standardized vocabularies/ontologies; to enable potential exchange of experience and know-how between health data related projects in Switzerland; and to increase the capacity of research groups to share/access/analyse health related data.

Stakeholders' requirements

In initiating the user-centered design process, three elements are of essential concern (1.) users, (2.) tasks and (3.) environments (see [3]). The main goal of the requirements definition process is to specify each element through a list of activities (enumerated in Box 1) generating specific outputs.

In order to generate these outputs, we followed two, complementary, approaches. The first is based on analysing a design process documentation of an existing system, with similar objectives. The second approach complements the first one with observations extracted from semistructured interviews conducted with potential users from the target group.

For the first approach we identified CEDAR[1], Center for Expanded Data Annotation and Retrieval, as an existing system close to ours in terms of objectives. It aims to facilitate creation and publishing of metadata describing scientific experiments. In the following we reuse some of CEDAR proposed personas and interaction scenarios and we complement them with observations relevant to our target users.

1. Identify *personas,* a set of typical users' profiles (i.e. archetypes).
2. Distill typical *scenarios,* a set of common interaction and usage scenarios with the system that help the user achieve a specific goal.
3. List *modalities*, through which the users interact with a system according to the environment where the interaction takes place.
4. Extract the *functional requirements* from the previous outputs, as a list of specifications that will orient the technical analysis and the system architecture.

Box 1. Activities oft he requirements definition process

Users

SHIS comprises multiple stakeholders that have an interest in creating and sustaining a health information central metadata repository. Our initial contact with a group of stakeholders in the context of a first focus group [10] distilled the need to make a distinction between stakeholders of the metadata repository and the actual users of the metadata repository. These two groups are generally not the same even though they could overlap.

In essence, stakeholders would include organizations that have a vested interest in the existence of a metadata repository or are furthering the interests of other groups. Stakeholders might interact directly with the data and the metadata resources, but they also might not interact at all. They include, but are not limited to: policy makers (decision makers, administrations and federal/cantonal offices); data and service providers (hospitals, practitioners, associations and NGOs), academic and research institutions.

The users, on the other hand, would be organizations or individuals having a need to actually access a data or a metadata resource. They include, but are not limited to: researchers, public

[1] https://metadatacenter.org/

health practitioners, hospitals, NGOs, patient organizations, standard development organizations (e.g. WHO, Elixir), media, healthcare industries, citizens.

With this distinction in mind, we will focus on users that need to interact with the metadata repository in order to access specific resources. When the term "stakeholders" is used interchangeably in this document it would refer to those users only. Additionally, the focus group discussion also distilled the need to select a specific group of users in order to clearly delimit the scope of the metadata repository. Indeed, the users groups enumerated are quite diverse and have different needs. A design cycle needs to be instantiated for each one of them and their outputs need to be combined. However, the focus group discussion also challenged the necessity of having a unique central metadata repository for all categories of users and stakeholders which was deemed infeasible.

Under these constraints, we elected to focus on the health researchers' user group for the design cycle that we instantiate in this policy brief. This choice can be argued based on the following observations: health researchers are one of the users groups that is mostly concerned in searching for and accessing data resources; they generate a large amount of data resources that require documentation in terms of metadata descriptions. Moreover, by focusing on this group, the Swiss National Science Foundation emerges as the main funding body that is encouraging and enforcing FAIR[2] data practices in its funded projects, which creates an important incentive to share the data resources created during a research project as well as their metadata.

In summary, the **health researchers users group** emerges as the most in need of such a metadata repository for health research data and this focus holds the most promising perspective for sustaining this repository.

Users' personas

User personas = primary roles that users play when interacting with the system being developed

In the context of the metadata repository we need to answer the needs of this minimal set of typical users.

Health researcher

This user role is the most frequent in our system. Such a user needs to gather information about datasets that are useful for the research being performed.

The typical interaction with the system would allow to:

- Search for and locate datasets with a given set of characteristics;
- Be able to filter datasets based on specific criteria;
- See how many datasets exist using different selection facets;
- Access metadata for any dataset meeting the search criteria, via a user interface or an Application Program Interface (API).

Metadata creator

This user role can be played by the health researcher even if, at the same time, it does entail a different set of goals. Basically, this user needs to describe one or more datasets using metadata that is conformant with the repository data model.

[2] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4792175/pdf/sdata201618.pdf

The typical interaction with the system would allow to:

- Use a data entry tool to provide the appropriate metadata;
- Fill out the metadata correctly, consistently, completely, and quickly;
- Follow the appropriate community-specific guidelines and standards for completing the metadata.

Collection curator

Metadata creator could also act as collection curator. This user works with existing metadata to improve their quality, including consistency, completeness, and conformance to community guidelines and standards.

The typical interaction with the system would allow to:

- Learn what metadata on datasets are available for curation;
- Be able to find and examine metadata on specific datasets, or collections of datasets;
- Be able to obtain summary statistics on the collections of interest, including analyzing the contents of specific metadata fields;
- Correct existing metadata;
- Review the provenance of metadata for the existing datasets;
- Review the status of datasets that they curate.

Repository administrator

This user is an administrative user that creates or selects specifications for other users to follow when submitting metadata based on existing community standards or developing new ones. The user is also responsible for developing the technical capabilities of the system and for integrating new ones in the future.

The typical interaction with the system would allow to:

- Implement an existing specification as a template;
- Learn what templates already exist, for a particular purpose or specification;
- Edit existing templates;
- Constrain how users fill out the templates;
- See what templates they have created and how they were used;
- Provide support to users on filling the predefined templates;
- Track and follow-up on system errors in the metadata creation and re-use by other users;
- Defines and provides user access, creation and editing rights.

Tasks

Specific actions and tasks allow the users to fulfill their objectives by interacting with the system. We organize these actions in groups to form usage scenarios (see Box 2). These scenarios normally comprise a minimal set of tasks to be enabled by the envisioned system. Each scenario enables the concerned user to

- Creating and editing metadata resources
- Browsing and curating matadata repository
- Metadata repository and template management

Box 2. List of usage scenarios

achieve a specific objective.

Creating and editing metadata resources

This scenario allows metadata creators and curators to add new metadata resources to the repository and to edit existing ones.

The typical actions in this scenario allow to:

- Receive suggestions for metadata values to fill in
- Save metadata
- Edit metadata
- View metadata editing history

Browsing and curating metadata repository

This scenario allows all users, but mainly health researchers and metadata curators, to search the metadata repository based on a specific search query or through browsing and links traversals.

The typical actions in this scenario allow to:

- Search metadata resources using semantic queries
- Browse the metadata repository
- Filter metadata resources based on specified concepts (e.g, from ontologies)
- Display a metadata record in the user-requested format
- Present a collection of metadata records for external analysis
- Export a metadata record
- Access metadata services and resources via APIs
- Generate usage statistics of metadata resources
- Use feedback and metadata quality information to update original metadata content

Managing the repository and metadata templates

This scenario allows the repository administrators to prepare the metadata templates to be filled by the metadata creators.This scenario ensures that the necessary standards are enforced in the template, both in terms of metadata schemas and in terms of controlled vocabularies.

The typical actions in this scenario allow to:

- Define the structure of the template (i.e. the schema) and its specific metadata elements' attributes.
- Define value sets for the metadata elements either by selecting terms from existing controlled vocabularies (e.g. ontologies or value sets in BioPortal) or from tailormade vocabularies
- Save metadata templates and their versions to a template database
- Edit existing template from the template database
- Search and view metadata templates and elements, and their histories
- Store, search, and revise value sets
- Create and edit users roles and permissions

Environments

Given the previously presented usage scenarios, identifying the environments where the tasks are to be performed dictates the level of flexibility required for the chosen technical system. Two environment categories are to be considered: physical and digital.

Digital environments

Digital environments are concerned with identifying the digital platforms that the users need to access in order to perform their interactions with the system, and what subsets of tasks are performed on these platforms. Our main users group, health researchers, would mainly interact with the digital system through a computer or a mobile based interface. Browsing the repository can involve both types of interfaces, while creating and editing metadata would most likely be executed through the computer based interface. With the advancement of mobile application development, we believe that this distinction is not crucial for the envisioned system as long as transferring the user experience from one interface to the other is seamless and transparent to the user.

Physical environments

Physical environments are concerned with identifying the surrounding physical conditions manifested where the users intend to interact with the system. They dictate the types of digital environments to enable. Our main users group, health researchers, largely operate from office facilities in a sedentary manner. Their requirements to upload data and metadata to online repositories are often fulfilled in the office environment and very rarely on the go or while performing other interventions in critical environments (e.g., hospital). This observation confirms the two digital environments identified previously.

Technical Analysis

The previous section detailed the users' requirements. In this section, we propose a high level architecture of the system that fulfills those requirements. As an intermediate step, we translated the users' requirements into specific functionalities. To keep the flow of the text, we relegate the details of those functionalities to Appendix I.

Architecture

The SH Metadata Repository provides a structured entry point to existing data resources, allowing its users to quickly locate and gather data of interest in order to perform analysis or reporting. The metadata repository needs to have a flexible structure able to handle a wide variety of unknown in advance and changing data. This structure should enable searching for specific data resources, adding new information and editing existing ones, as well as establishing relations between the various resources on the basis of similarities in their metadata. The proposed architecture answers those objectives.

Most user-oriented applications follow a *n-tier* or *multi-tier* architecture. Each tier represents a layer in the system and aims to separate the processing, data management and presentation functions. Usually, these different functions are hosted on distinct machines, ensuring that services are provided with enough resources. Hardware specifications will not be addressed in this report, nevertheless, they are a crucial element of the sustainability of the system in production. The main benefits of such n-tier architectures are: security, ease of management, flexibility and scalability. Each tier can be managed and secured separately without affecting the other tiers.

The proposed architecture, represented in Figure 2, is a 3-tier model divided into:

1. *Presentation tier*
2. *Logic tier*
3. *Data tier*

The following subsections present a detailed view of each tier and their subcomponents.

Data tier

This layer stores and retrieves data from databases or file systems. The data is then passed to the logic tier for processing. Two main categories of databases could be called for in this architecture: relational databases and non-relational databases. The differences concern their construction, the type of information they store, and the storing mechanism.

A relational database consists of a collection of tables arranged in columns and rows, manipulated by relational operations [11]. Each row in a table represents a record, and each column defines an attribute coding a specific type of information for the record, like a name, address, and phone number. The assignment of attributes to tables is encoded by the database schema. In a relational database, the schema is defined before any information is added. Data stored in relational databases has to be structured according to the schema. A well-designed schema minimizes data redundancy and prevents data inconsistencies.

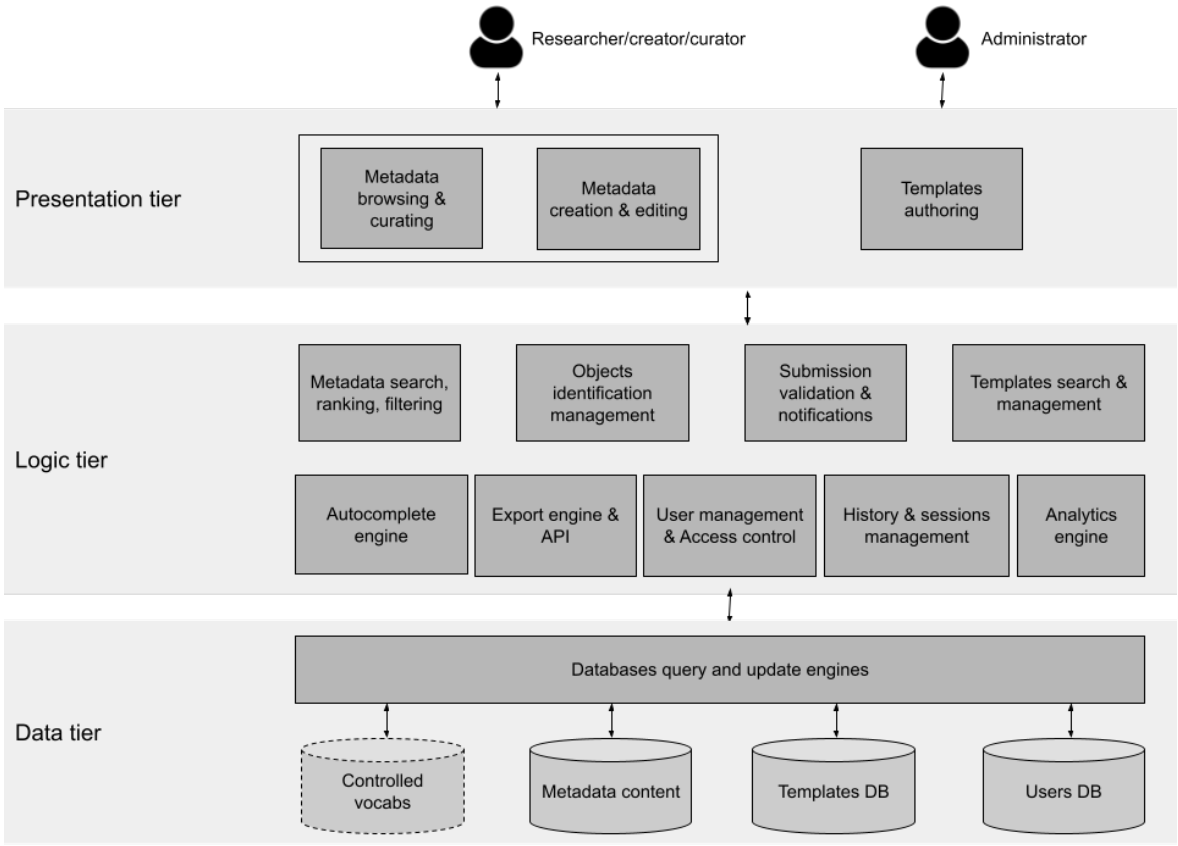


Figure 2. Schematic of a proposed system architecture

A non-relational database has a dynamic schema for unstructured data which could be stored in different ways: column-oriented [12], document-oriented [13], graph-based [14] or as a KeyValue store [15]. This flexibility means that data can be added without having to structure it according to a predefined schema. Each data type maintains its own structure. This increased flexibility requires however a greater processing effort and more storage than with relational databases. Non-relational databases are therefore suited to situations with unknown in advance data requirements or dealing with large amounts of unstructured data.

In particular, graph-databases use a graph structure composed of nodes linked by edges to integrate and query the data [16]. It is particularly useful for interconnected data items. In graph databases, edges representing relationships between the nodes are explicitly identified and can themselves be queried and manipulated. The stored graph structure reflects the conceptual view of the data. Under the assumption of unlimited storage, new data can be added as long as it can be formatted as nodes and edges and their associated properties. We refer the reader to [3] for more specific details on semantic graphs formed by subject-predicate-object triples as a specific instance of a graph model.

A minimum of four databases are needed in our architecture: the templates database (relational), containing the predefined metadata templates; the users database (relational), containing the users profiles, their associated resources, and their authorizations; the metadata repository (graph), containing the actual metadata content; and the controlled vocabularies (graph), containing the predefined sets of values to be used for autocompletion and for semantic annotation.

Logic tier

The logic tier contains the code that implements the functionalities of the application; in particular, writing and reading data into/from the data tier. Many programming languages and corresponding frameworks may be considered to implement this tier. Choosing the best technology needs to take into account some of the following questions[3]:

1. Does the language have a proper ecosystem support for the long haul? Is there technical support provided for the language?
2. In which environment the project will run (e.g., web, mobile)?
3. What are infrastructure and deployments considerations for using the programming language, (e.g., new hardware required)?
4. What are the industry standards in terms of libraries, features, and tools for this type of project?
5. What are the non-negotiable project constraints (e.g., time, budget, resources)?
6. What are the benchmarks and performance measures that the project is required to reach and how the programming language would accommodate those benchmarks?
7. Are there any legacy codebase considerations for the project?
8. What types of interfaces with upstream and downstream systems or third-party systems will need to be implemented?
9. What are the security considerations that need to be guaranteed?
10. What are the human resources needed and/or readily available for such an implementation and what are their competencies? What are the costs to acquire the missing skills?

Given that the metadata repository is an entry point for users from different organizations, allowing easy access to its platform in a frictionless manner would improve its usability. Web applications through the use of the Internet and web browsers are the go-to choice to guarantee such easy and wide ranging access, both in terms of user location and computing devices (e.g., computers, smartphones, and tablets). A web application for the metadata repository is also coherent with the digital environments that our selected users group is very familiar with.

Therefore, the chosen programming language should allow the development of Web applications, allow an n-tier deployment and be able to read and write data from relational and/or non-relational databases. To meet those constraints, the Model-View-Controller (MVC) programming design pattern, an important design pattern in computer science [17], can be used to architect software systems. Its key idea is to separate every element of the application into parts with strictly dedicated roles [18], as depicted in Figure 3:

- Model - manipulates data and provides the business logic element
- View - uses data from Model to create the representations provided to the end-user
- Controller - accepts users interactions and converts them to commands to Model and View

[3] https://medium.com/better-programming/how-to-choose-a-programming-language-for-a-project-7c7a3e5a4de6

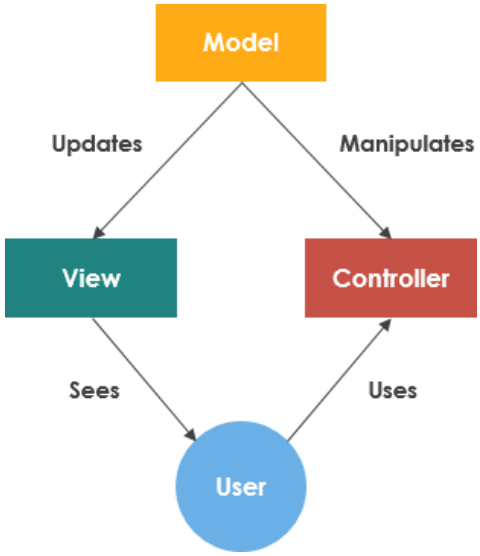


Figure 3. Model-View-Controller architectural design pattern[4]

This design pattern has become popular for designing web applications [19]. Popular programming languages like JavaScript, Python, Ruby, PHP, Java, C#, and Swift have MVC frameworks that are used for web or mobile application development.

MVC allows to structure the logic tier of the application such that all the heavy lifting logic and content rendering is done on the server side to allow for simpler client-side applications that only need to request the rendered content and display it to the user.

With these considerations in mind, the proposed architecture logic tier manages the core of the application and implements the following functions (along the detailed functionalities they serve from the list in Appendix I):

- *Metadata search, ranking and filtering*:
  - takes the users' keywords and filters and formats them into queries called on the metadata repository database.
  - receives the answers from the database, applies the ranking algorithm, and formats the answer in a relevant way for the user.
  - serves functionalities: 1, 2, 6, 7, 8, 9, 10.
- *Submission validation and notifications*:
  - manages the inputs on the metadata fields when a metadata template is being filled and validates the content before committing it to the metadata database.
  - prepares the feedback to the user to notify about successful submission or eventual errors.
  - serves functionalities: 3, 4, 13, 14, 15, 16, 17, 18, 19, 20, 21, 24, 25, 26, 32.
- *Object identification management*:
  - guarantees that each entity in the system can be uniquely identified by a Unique Resource Identifier (URI).

[4] https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-model-view-control-mvc/

  - manages looking up and reusing publicly available identifiers (e.g., for organizations) or maps the public identifiers to the local ones.
  - serves functionalities: 2, 7, 33.
- *Templates search and management*:
  - provides the tools necessary to create new metadata templates by the administrator, to search for existing templates, to visualize or edit and to open up the necessary connections with the templates databases to retrieve and update this information.
  - serves functionalities: 27, 28, 29, 30, 31.
- *Autocomplete engine*:
  - manages the connection between the metadata templates filling functionalities and the predefined vocabularies databases.
  - supports the users in filling the metadata values according to predefined vocabularies or preset values by suggesting adequate inputs based on typed letters.
  - serves functionalities: 2, 8, 22.
- *Export engine and API*:
  - provides functionalities for bulk export of metadata as files or through an Application Programming Interface (API) that allows to connect the metadata repository to external applications.
  - serves functionalities: 5, 11.
- *Users management and access control*:
  - manages the creation of new user profiles in the users' database.
  - manages the assignment of metadata resources to those users and their rights for accessing and editing existing resources.
  - serves functionalities: 15, 16, 19, 23, 27, 28, 30, 3, 35, 36, 37, 38, 39.
- *History and session management*:
  - manages the user sessions and allows to save the navigation history and all interactions with the system.
  - serves functionalities: 34, 35, 38, 40.
- *Analytics engine*:
  - allows to follow up on the usage level of the different resources and frequent queries.
  - analyses the historical information to provide a better tailored experience in the future by updating the ranking algorithms or by identifying potential recommendations of new resources to show to the user
  - optimizes the most frequently used functionalities.
  - serves functionalities: 12, 40.

Presentation tier

The presentation tier provides user interface components that allow users to interact with the application, depending on their role. This tier does not directly process data from databases nor handle business rules. It mainly consists of tools to render the relevant content and allow its display in a compelling and useful way to the user.

In the proposed architecture the presentation tier would directly allow the user to act on the system in the context of the scenarios presented earlier, namely, to browse the metadata repository and curate its content (served by functionalities 1 to 12), to create or edit existing metadata in the repository (served by functionalities 13 to 26) and to manage the repository and its metadata templates (served by functionalities 27 to 40). These scenarios are accessible to the user depending on her role and her authorization level. The user roles are also dependent on the personas proposed earlier. We note four types of roles to configure:

- *researcher* is the standard user role that allows to browse the metadata repository content;
- *creator* is a user role that is authorized to input and edit metadata content through the predefined templates;
- *curator* is a user role that verifies and improves the quality of the existing metadata in the collections under her responsibility;
- *administrator* makes sure that the predefined vocabularies and data schemas are respected by creating the necessary templates for the metadata, connecting necessary vocabularies, and assigning access roles to users.

If the repository is open to the public, the *researcher* role would be the default role for non-logged in users. Those roles are also organized in a hierarchy where higher roles in the hierarchy inherit the rights from lower roles and receive additional rights. The roles in this architecture could be organized, from lower to upper, as follows: *researcher, creator, curator* and *administrator*.

Potential solutions

Here we present some of the relevant existing platforms and discuss them in comparison to a tailor-made solution. The goal here is to give some relevant platforms, without being exhaustive, and their potential usefulness in implementing the proposed architecture.

CKAN[5]

Comprehensive Knowledge Archive Network (CKAN) is a data management system that allows to store and share data on the web through a dedicated portal. It is a widely used platform to publish open government data (e.g., data.gov, opendata.swiss, data.europa.eu and others). CKAN uses PostgreSQL[6] for storing data and SOLR[7] for keyword search. By default it supports the following metadata[8]: title, description, tags, license, organization, link to the data (or a possibility to upload), dataset file format, visibility, author, author email, maintainer, maintainer email, and other custom fields as key-value pairs.

CKAN is a quite mature ecosystem for generic open data publishing. It is open source and has a rich documentation. It could be a good point to start to implement the most basic functionalities of the proposed architecture. However, it is important to note that it does not provide by default the possibility to link the metadata that is provided with controlled vocabularies and ontologies.

[5] https://ckan.org/

[6] https://www.postgresql.org/

[7] https://solr.apache.org/

[8] https://docs.ckan.org/en/2.9/user-guide.html?highlight=metadata#datasets-and-resources

Furthermore, it shows query results in lists, which indeed is useful, but lacks a visually expressive browsing tool benefiting the user from the potential links between the datasets (e.g., in terms of variables, similarities, organizations, etc.).

Zenodo[9]

Zenodo is a data repository hosted by CERN. It is one example of a generic data repository[10] used by researchers in Switzerland to upload their datasets in accordance with their projects' FAIR data management plan. It is based on PostgreSQL, Elasticsearch[11] and Redis[12] and functions on top of Invenio[13]. Zenodo is open source and can be extended by developers. Compared to CKAN, Zenodo adds some smarter search engines on top of the database that can also match some of the content in the data. Despite the powerful search technologies at its core, the browsing modality, very similar to CKAN, returns a list of datasets given the query and the selected filters. As a generic repository, the contained data comes from various sources: datasets, publications, code repositories and others. This data variability is not supported by a well-defined metadata structure and it's the responsibility of researchers to explain/annotate the content of the data and its formatting in a few attributes like the title and the description. Furthermore, the metadata is not controlled by any domain vocabulary. Yet, when the dataset format allows for it, the content of the data can be previewed on the dataset page. In addition to datasets indexing, two relevant functionalities of Zenodo are to be pointed out: the providing of a Digital Object Identifier (DOI) for each asset in the repository which is common to all versions of the same dataset and the connection with a github repository to refer to the used codes in manipulating the dataset.

CEDAR[14]

The Center for Expanded Data Annotation and Retrieval (CEDAR) is a content manager facilitating the submission of metadata about biomedical scientific data. Metadata is saved using JSON-LD method in a MongoDB database. The focus of CEDAR is to provide tools to create metadata attributes, to combine them in templates and to fill these templates for given datasets. By default, all metadata can be linked to taxonomies, vocabularies and ontologies available on Bioportal[15]. Unlike CKAN and Zenodo, which focuses on providing access to the data, CEDAR's main focus is on authoring quality metadata. It is open source and its workbench can be integrated to provide some of the proposed functionalities. Yet, the initial provided documentation was not clear enough on how to do such an integration. Recently, an initiative was launched to integrate CEDAR tools in a generic metadata authoring workbench to facilitate FAIR data sharing practices (FAIR Data collective[16]).

---

[9] https://zenodo.org/

[10] http://www.snf.ch/fr/leFNS/points-de-vue-politique-de-recherche/open_research_data/Pages/depots-de-donnees.aspx

[11] https://www.elastic.co/elasticsearch/

[12] https://redis.io/

[13] https://inveniosoftware.org/

[14] https://metadatacenter.org/

[15] https://bioportal.bioontology.org/

[16] https://github.com/fair-data-collective

CEDAR targets scientific data publication and reproducibility by allowing for richer descriptions of the content of the datasets and by controlling the content of those descriptions by domain relevant ontologies and vocabularies. CEDAR does not provide a repository for the data itself but allows users to connect to existing repositories and to submit the created metadata to those repositories along with a given scientific dataset.

In an early stage of our analysis, while recognizing CEDAR to be conceptually close to our project, we dismissed its adaptation due to the lack of clear documentation regarding integration. However, with the release of the codes on the FAIR Data Collective github, it might be worth reconsidering some of the components during the implementation, namely the autocomplete functionality supported by controlled vocabularies from Bioportal or any other ontology portal.

Tailor-made solution

From our analysis of existing data repository portals, (particularly CEDAR) and from the interviews with the target group of users (particularly, the *researchers*), it appears that, at this moment, there is no available entry point to search for datasets that are relevant to the health community. Health researchers, as proved by the SLHS consortium, are related to multiple scientific domains, from biomedical, economics, legal, social and others. Their needs could be served by multiple generic or specialized repositories, but the necessity of a central search point for those scattered resources is obvious. A tailor-made solution for health researchers, and SLHS in particular, is a very logical and relevant proposition. The advantages of such a choice are manifolds: the search entry point based on metadata search can be tailored to the researchers needs and expected ways of looking and processing information; tools to enforce annotating metadata with standard vocabularies and ontologies can be put in place; tools for a richer exploration of the available data through richer metadata could be implemented and tested. An obvious drawback of such a choice is the development effort that is needed to put in place and validate such a solution. However, all the previously explored solutions are open source, meaning that their code is available with more or less clear documentation for their integration. A feasible option for a tailor-made solution would be to actually identify and choose relevant components from the different tools and combine them in a useful way. In that sense the tailor-made solution is not an effort to develop a solution from scratch but an effort to integrate multiple tools and functionalities instead with additional development time for non existing functionalities. For example, the workbench of CEDAR could be used to create and fill templates with controlled vocabularies from local and remote ontology portals, DOI management tools can be adapted from Zenodo. Richer browsing functionalities would need to be developed on top of a graph structured database in order to benefit from the linked metadata capabilities.

In the following chapter we present the first tile in a functional prototype that aims to mainly test a graph based browsing functionality on top of a graph database. The CEDAR workbench was not integrated at the time of development due to lack of the documentation, therefore we provided instead a fixed template based on the fields defined in our first analysis [3].

Interactive prototype implementation

Technological choices

In line with the above technical analysis, each tiers of the prototype was the result of a series of technological choices. The choices were made in such a way that the prototype could be easily understood, taken up and maintained. In addition, the technologies chosen are intended to be flexible and documented to guarantee the evolution and adaptation to the business requirements of future iterations of this prototype. In the following sections choices are depicted tiers by tiers.

Data layer

In the data tier, as the implemented solution deals with four databases responding to very specific needs, different choices have been made for each one. The users' database implements a well-known issue across multiple information systems. As such, the implementation of the users' database relies on an approved and predictable structure presented in the following entity-relation diagram (Figure 4). Each *User* can endorse zero to many *Roles*, each *Role* being associated zero to many *Permissions*

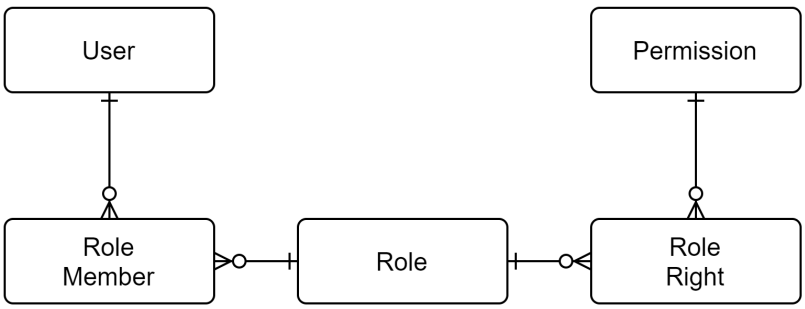


Figure 4. Entity Relation diagram for users' database

Regarding the controlled vocabularies, they are set by pre-existing taxonomies and ontologies on which the implemented prototype can rely. The metadata repository, which is the core of the implemented prototype, is managed through a graph database as suggested by [3]. Table 1 describes the four main entities of this database schema.

Entity	**Description**
Dataset	Datasets are the main resources of the SH metadata repository. They describe a database, a file, a document or any other piece of information users are willing to make visible in the repository. This entity does not contain the dataset itself, but the description of it. The dataset entity aims in fact to contain the descriptive metadata elements allowing the identification and retrieval of datasets / data resource.

Organization	Organization designates the entity owning the dataset. This entity potentially allows the discovery of Datasets published by the same Organization.
Person	Person represents the creator of the Dataset description recorded in the metadata repository, and can be part of an Organization. This entity allows to facilitate the metadata management for metadata creators, but also to navigate and discover Datasets related to the same Person or Organization.
Tag	A tag represents a keyword that one or more datasets can share. Tags help to group Datasets around similar topics.Therefore, when a user performs research on a specific tag, he will be able to find all the datasets using that tag.

Table 1. Entities designed in the graph database

Dataset is the central entity of the SH Metadata Repository. A dataset is usually published by an *Organization*. Datasets descriptions are manually managed by metadata repository users, represented by the *Person* entity. Finally, to ease the discovery of datasets, users can make use of *Tags*, which are uniquely identified keywords associated with the dataset in order to facilitate its discovery.

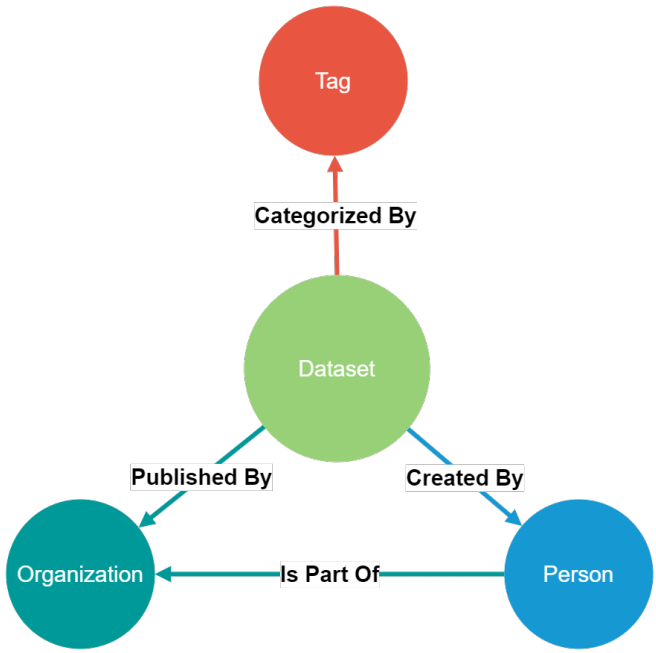


Figure 5. The links connecting the entities of the metadata repository

In the context of SHIS, the recommendations established in Policy Brief 1 [3] suggest adopting Dublin Core metadata schema [20]. Consequently, the core set of the Dublin Core has been adapted to the context (as detailed in Table 2) and implemented as metadata descriptions stored in *Dataset* entities. (*Note*: "resource" represents the data source referred by a *Dataset* instance)

Element	**Definition**
Identifier	Auto-generated URI

Title	A name given to the Dataset
Description	An account of the resource content
Languages	Languages of the resource content as URIs references
Rights	Information about rights related to the resource as URI reference
Access	Information on how to access the resource as URL reference
Spatial Coverage	Geographic coverage of the resource content as URI reference
Temporal Coverage	Calendar coverage of the resource content as URI reference
Creation Date	Date of creation of the resource content
Modification Date	Last modification date of the resource content
Created At	Auto-generated date of creation of the Dataset instance in the SH Metadata Repository
Updated At	Auto-generated date related to the last modification of the Dataset instance in the SH Metadata Repository

Table 2. Fields of entity Database (based on Dublin Core schema)

Dataset entities are also linked to *Person*, *Organization* and *Tag* entities through the links presented in Figure 5. Such links allow the navigation, the discovery and sorting of Datasets through uniquely and unequivocal Person, Organizations and Tags.

Relation	**Direction**	**Definition**
Created By	*Dataset → Person*	Person who created the resource description in the SH Metadata Repository
Published By	*Dataset → Organization*	Owning organization of the content of the resource as URI reference
Categorized By	*Dataset → Tag*	Keywords categorizing the content of the resource as URIs references

The design of the entities *Person*, *Organization* and *Tag* facilitates the addition of extra descriptive information of interest when needed in future iterations of the SHI Metadata Repository. In the current version of the proof-of-concept, descriptive information of the various entities are limited to the fields described in Table 3 for the *Person* entity, Table 4 for *Organization* entity and Table 5 for the *Tag* entity. The relation between the entities *Person* and *Organization*, guaranteeing the navigability between these two entities, is described in Table 6.

Element	Definition
Identifier	Auto-generated URI
Firstname	Firstname of the Person
Lastname	Lastname of the Person
Email	Email of the Person

Table 3. Fields of entity Person

Element	Definition
Identifier	Auto-generated URI
Label	Name of the Organization

Table 4. Fields of entity Organization

Element	Definition
Identifier	Auto-generated URI
Label	Label of the Tag

Table 5. Fields of entity Tag

Relation	Direction	Definition
Is Part Of	*Person → Organization*	Belonging Organization of the Person

Table 6. Relation between the entities Person adn Organization

In order to guarantee that every future instance of *Person, Organization* and *Tag* created in the metadata repository is uniquely identified, this prototype is designed to permit the generation of URIs and the linkage of them with existing ones in the Linked Open Data. The Linked Open Data represents the Web of Data, where all data sources could potentially be uniquely identified. Thus, once created, the instances of *Person, Organization* and *Tag* can be linked to Datasets instances generated by a metadata creator, but also to external (to the SH Metadata Repository) entities, uniquely identified online resources whether this resource already exists in the Linked Open Data.

In the actual version of the proof-of-concept, the system automatically creates links between similar tags in order to allow the discovery of resources categorized by the same tag. However,

this process could be larger improved by linking generated Tags to existing taxonomies and ontologies, thus allowing capturing the exact semantic meaning of each Tag instance.

For the sake of prototypisation the graph database is based on Neo4j (Network Exploration and Optimization 4 Java), a graph database management system developed by Neo4j, Inc[17]. But any other graph database such as MarkLogic[18] or DataStax[19] would fit.

Logic layer

Regarding the application code, there are many programming languages and development frameworks allowing to implement the logic tier. Main considerations in selecting the development platform were related to the user needs, specifically the need to access the metadata repository from everywhere, anytime and possibly from mobile devices. Another consideration was the non-relational databases interfacing since this is one of the main features of the metadata repository. The popularity of the coding technologies, and therefore the ease of maintenance and evolution of the platform was also included in the decision process. Considering these selection criterias, the Python programming language was ranked as the first choice. Other potential alternatives such as Java or Ruby programming language were discarded from our final choice, mainly for functionality and community size issues, especially considering the interfacing with the graph database. Python language coupled with Flask framework[20] allows the development of web digital solutions interfacing with flexible data structure and are among the most popular platforms. Flask is a highly customizable development framework, allowing to structure the code according to the MVC pattern and facilitating the interfacing with databases of different kinds but also facilitating integration of functionalities such as the authentication, through pre-built libraries of code.

Presentation layer

Python, Flask and Neo4j are offering several visualization libraries allowing users to interact with entities stored in the graph database. A potential choice is the dynamic, browser-based visualization library vis.js[21]. The library is designed to be easy to use, to handle large amounts of dynamic data, and to enable manipulation of and interaction with the data. This kind of Javascript library is built on the top of the Web application built with Python and Flask.

In the end, the proposed prototype relies on an architecture highly flexible and scalable. Each technological choice can be, in further iteration of the implementation, replaced or modified without major impact on the rest of the architecture. The n-tiers architecture voluntarily led us to weakly coupled technologies that could be replaced layer by layer with a minimum of effort.

Proof-of-Concept and Usability tests

As we needed the early involvement of stakeholders to build such an important and innovative infrastructure, the SH Metadata Repository proof-of-concept was implemented following the

[17] https://neo4j.com/

[18] https://www.marklogic.com/

[19] https://www.datastax.com/products/datastax-enterprise

[20] https://flask.palletsprojects.com/en/2.0.x/

[21] https://visjs.org/

design science research methodology (DSRM), allowing us to avoid failures due to needs misinterpretation.

Based on the conclusions of the Policy Brief [3] and of the corresponding Stakeholder Dialog [10], we could come up with a clear idea of the design of a first prototype mockup for SH Metadata Repository (see Appendix II A). This prototype, in the form of a clickable model with pre-defined behaviour, was intended as a first demonstrator to validate our understanding of the SLHS stakeholders' needs. The mockups were presented during the SLHS Annual Meeting in November 2019. An interactive presentation of the clickable interface was performed in the plenum assembly, while the printed version was then commented in a discussion group. The result of this first iteration allowed us to establish a first consolidated list of functionalities (see Appendix I).

Starting from this list of functionalities we developed a proof-of-concept prototype. In addition to the purely visual aspect (presentation tier) and potential desired functionalities that the prototype made it possible to evaluate, the proof-of-concept is a functional, first minimum viable solution of a SH Metadata Repository. This prototype is accessible throught Internet at the address http://imi-slhs.unine.ch/.

The minimum viable solution is not a sub-part or a component of the SH metadata repository, but an already functional solution in a lighter form. The proof-of-concept takes the form of a search engine, where keywords can be inserted in a search field in order to retrieve related datasets. Various Datasets instances have been included for the sake of the usability tests. These datasets were retrieved among SLHS stakeholders and by retrieving publicly available datasets, such as the ones of the Federal Statistical Office. The result of a search request is presented in a text format as well as a graph format allowing the users to visualize datasets, organizations and persons corresponding to their request (see Appendix II B).

As the system has to grow and evolve with user generated content, many other iterations would allow to refine and polish this first proposition that proves the feasibility and utility of the concept. Concretely, to evaluate the utility and the potential gaps and shortages of this proof-of-concept, usability tests have been conducted with stakeholders. Seven researchers (two professors and five PhD students) recruited from the SLHS consortium were invited to test the usability of the implemented prototype through semi-directed interviews. Each interview lasted for one hour and was held online. It was decomposed in two parts: a first part, to better learn about the interviewee's current use of data and metadata resources and a second part, to evaluate the implemented prototype showcasing the functionalities of the envisioned metadata repository. The first part took the form of a discussion, while during the second part, the interviewees interacted with the functional prototype through a web browser. Their interactions were observed by the interviewers through screen sharing and "think aloud" protocol. All interviews were recorded and the outputs were used to come up with a list of additional functionalities, counterintuitive interfaces and bugs that should be considered for future iterations (see Appendix III A). A detailed analysis of the recorded interviews and the complete list of the recommended implementations raised by the usability tests are presented in [9].

On the top of this list of additional features and bugs we created a new prototype illustrating what would resemble the next iteration of the SH Metadata Repository (see Appendix IV). The improved prototype mockups are tackling three main key aspects: the onboarding experience, the visualization of the search result and the dataset metadata creation.

The onboarding experience is improved by providing some clarification on the start page of the SH Metadata Repository. Predefined query search examples are provided, to start exploring the repository and provide a better intuition of the platform utility. Some textual clarification has also been added to further clarify the aim of the metadata repository. A getting started video (still to be produced) also appears on the main page of the platform.

The result visualization has been much more oriented around the graph visualization, which is now at the center of the interactions with the user. Clicking on the nodes allows to expand the visualization and further discover attached resources. Textual information is also provided in function of the selected node in the visualization. The search result can also be sorted and filtered to make the visualization automatically adapt to the refined query.

Finally the metadata creation is also facilitated by further clarifying what is expected for each field of the form. Contextual help and examples are also displayed when the mouse of the user is over a specific field. This prototype mockup is a first step in the concretisation of a second iteration over the design of the SH Metadata Repository. The functional proof-of-concept should be adapted to align itself with these latter mockups.

Discussions and outlook

As concluded in the Policy Brief [3], in the context of a decentralized approach for health data resources, a metadata management infrastructure becomes crucial in order to guarantee that the necessary resources are retrieved and accessed from the right databases and respecting each dataset's specificities. Hence, a central metadata repository, designed as the main element of the metadata management infrastructure, becomes essential for the identification/description of health data sources. Metadata repositories were identified to be central building blocks in multiple initiatives aiming to facilitate health data access, linking and analysis ([4] [5] [6]).

A Swiss Health Metadata Repository is expected to enable multiple advantages, namely: to provide a single entry point for searching/retrieving health-related resources; to help identify possible semantic linking between data sources; to provide a consistent data catalog ensured by the use of standardized vocabularies/ontologies; to enable potential exchange of experience and know-how between health data related projects in Switzerland; and to increase the capacity of research groups to share/access/analyse health related data.

From prototype to production

In the previous chapters we presented a possible architecture for a metadata repository that satisfies users (i.e., health researchers) requirements. We discussed possible solutions to bring it to reality and presented a proof-of-concept implementation mainly showcasing new forms of interaction with the metadata. However, the current in production proof-of-concept is still restricted to actually serve users in their daily tasks, due to several limitations that we discuss hereafter.

(1) **Available data collection**: the current content of the metadata repository is too limited to be useful for researchers needs. However, this is a very important element to tackle moving forward since it will impact the usefulness of the repository. Following users' interviews [9], we propose to identify a minimal set most relevant to the SLHS community by considering the feedback from the health researchers' community at large when proposing new resources. The current content actually inlcudes some elements we identified from this minimal set. In fact, curating and filling the content of the repository might become a limiting factor if the effort is not compensated by the usefulness gained. In order to circumvent this limitation we propose to plan for (semi-)automated tools to integrate existing metadata that can be imported from curated data providers and repositories like SPHN, Federal Office of Statistics, FORS or others. Additionally, we propose to develop tools to facilitate the integration of new metadata in parallel to new dataset submissions to repositories (e.g., when required by data management plans).

(2) **Metadata attributes**: we identified Dublin Core as a suitable starting point to structure the metadata schema. Yet, it was deemed insufficient for practical use and its attributes significance not necessarily intuitive. We proposed in this document an extension of the minimal set described in [3], as well as improvements in the interface to help the user identify the significance of each attribute. However, given the diversity of the users and how they look for and analyse data in their respective domains, one solution will never fit all expectations. What we identified (see Appendix III B) is a common denominator to the

users we were able to discuss with (yet, this denominator needs to be verified and updated at future development iterations). Furthermore, users whose needs cannot directly be met by an explicit content in the interface should have an easy way to look for this information further (e.g., if details on a survey cannot be added to the interface, at least a file to the survey's detailed description or codebook should be made easily accessible through the same interface).

(3) **Data standards**: taxonomies and controlled vocabularies remained in our work hard to efficiently integrate in the proof-of-concept, mainly due to their limited usability to annotate Dublin Core high-level metadata. Instead, these semantic resources are most commonly used, even though not consistently, at the level of the dataset items/variables coding, which is normally strongly dependent on the domain. Following our initial evaluation of the proof-of-concept, we identified the need to expand the metadata attributes to include survey/study variables. For this extension and to improve the other metadata fields' consistency (e.g., common significance of keywords) we propose to integrate annotations and autocomplete from an ontology portal like Bioportal. For example, parts of the CEDAR workbench could be used to implement this integration and enrich the metadata editing tools available.

(4) **Functionalities and technical ecosystem**: the proof-of-concept aimed to showcase the richer browsing capabilities available with a graph-based structure for the metadata repository. Due to time limitations and its explorative nature, all the proof-of-concepts's technical aspects have a large potential of improvement. As we discussed earlier, a tailor-made solution can be accelerated by opting to integrate parts of existing solutions that would answer the requirements and to develop from scratch only those functionalities that are not met by existing tools. Specifically, integrating CKAN as content management backend with CEDAR metadata authoring tools in the frontend, along with the graph based visualization, would make for a modular system that can easily be adapt to users' needs evolution. Furthermore, enforcing access control and controlling users' authorizations in creating/editing the content need to be done by an appointed administrator. It is also neccesary to implement measures to secure the system's databases from tempering and breach, as well as to make the system fault tolerant in terms of physical and network infrastructures to guarantee all time accessibility.

From the previous exposition, it becomes clear that going from this proof-of-concept work into a system in production would require resources, human and material, for:

- maintaining contact with users and further gather their requirements;
- performing the necessary integrations and developments;
- maintaining the system infrastructure;
- planning future improvements, and
- keeping up to date with relevant technological developments as well as other needs.

This brings up the questions of governance and resources that were identified in [3]. Without repeating here the cited arguments, we complement the previous discussions by identifying a non-exhaustive list of risks [21] that could be relevant for the SH Metadata Repository project.

Major risks and potential mitigation

- **Requirements shifts**: it is very common in the lifetime of an IT project development life cycle that users and stakeholders' needs and requirements evolve, especially for long projects. Even if initial requirements do not drastically change, it is very common that new requirements arise during the project development. An iterative/incremental approach to the project development, following the user-centered design cycle [1], can largely take those developments into account. In practice, user-centered design cycles are integrated with agile project management approaches [22] to discover users' needs, divide the project into iterations with minimal outputs at each iteration, and evaluate each output together with the users / stakeholders for preparing the upcoming cycles. This frequent contact with users and stakeholders allows the technical team to show frequent advancements and to integrate more efficiently changing and newly arising needs.
- **Technical difficulties**: these difficulties can stem from lack of knowledge in the available human resources for the selected technologies, a suboptimal choice of technological tools or lack or suboptimal sizing of physical resources on which the IT project will run. To avoid such issues the system designers need to spend time on evaluating and comparing existing tools and their ease of integration (i.e., available and clear documentation). The IT staff must ensure that technical support is available either by the tools providers or by open source communities and evaluate the potential load on the system by sizing the physical resources accordingly. Acquiring the competent human resources is crucial for success - this acquisition can be supported through new recruitments or by liberating time to train existing resources on new technologies as needed.
- **Poor management and budgeting**: poor budgeting both in terms of time and finances can be detrimental to the project. In general, poor project management practices and lack of long term vision negatively impact all aspects of a project. Even though agile project management methodologies reduce the risks related to uncertainty arising in the lifetime of a project, careful planning is still needed especially in terms of finances and the potential need for extra resources in the near future. This brings the importance of having a strong governing body in support of the development of the metadata repository that can provide the necessary resources, guarantee a long term vision and viability for the product and overall maintain an oversight and accountability of all involved parties.

These risks, and potentially others, need to be addressed by a risk management plan [23] at the beginning of a project dedicated to put the proposed metadata repository in production.

References

[1] D. Norman et S. W. Draper, User Centered System Design - New Perspectives on Human-Computer Interaction Edited by, 1986.

[2] T. Lippeveld, R. Sauerborn and C. Bodart, Design and Implementation of Health Information Systems, WHO, 2000.

[3] A. De Santo, E. Maalouf, P. Cotofrei and K. Stoffel, "Design principles of a central metadata repository as a key element of an integrated health information system," SLHS, 2019.

[4] K. Crameri, A. Lawrence and L. Selter, "SPHN short report," Swiss Personalized Health Network, 2018.

[5] FORS, "Accessing and linking data for research in Switzerland," November 2020. [Online]. Available: https://linkhub.ch/fr/rapport/.

[6] A. L. Egyedi, M. J. O'Connor, M. Martínez-Romero, D. Willrett, J. Hardi, J. Graybeal et M. A. Musen, «Using semantic technologies to enhance metadata submissions to public repositories in biomedicine,» 2018.

[7] A. R. Hevner, S. T. March, J. Park et S. Ram, «Design science in information systems research,» *MIS Quarterly: Management Information Systems,* vol. 28, n° %11, 2004.

[8] K. Peffers, T. Tuunanen, M. A. Rothenberger et S. Chatterjee, «A design science research methodology for information systems research,» *Journal of Management Information Systems,* vol. 24, n° %13, 2007.

[9] E. Maalouf and A. De Santo, "Swiss Health Metadata Repository: Prototype Evaluation," SLHS, 2021.

[10] E. Maalouf and A. De Santo, "Summary of the Stakeholder-Dialog on the topic: Design principles of a central metadata repository," SLHS, 2020.

[11] P. J. Pratt, Concepts of Database Management, 2008.

[12] D. J. Abadi, P. B. Boncz et S. Harizopoulos, «Column-oriented database systems,» *Proceedings of the VLDB Endowment,* vol. 2, n° %12, 2009.

[13] K. K. Y. Lee, W. C. Tang et K. S. Choi, «Alternatives to relational database: Comparison of

NoSQL and XML approaches for clinical data storage,» *Computer Methods and Programs in Biomedicine,* vol. 110, n° %11, 2013.

[14] R. Angles et C. Gutierrez, «Survey of graph database models,» *ACM Computing Surveys,* vol. 40, n° %11, 2008.

[15] W. Puangsaijai et S. Puntheeranurak, «A comparative study of relational database and key-value database for big data applications,» 2017.

[16] Z. J. Zhang, «Graph Databases for Knowledge Management,» *IT Professional,* vol. 19, n° %16, 2017.

[17] J. Bucanek, «Model-View-Controller Pattern,» 2009.

[18] B. Padraic, The M in MVC: Why Models are Misunderstood and Unappreciated, Astrum Futura, 2008.

[19] I. Davis, "What Are The Benefits of MVC?," *Internet Alchemy,* 9 12 2008.

[20] S. Weibel, J. Kunze, C. Lagoze et M. Wolf, «RFC2413: Dublin Core Metadata for Resource Discovery,» *Internet RFCs,* 1998.

[21] H. Barki, S. Rivard et J. Talbot, «Toward an Assessment of Software Development Risk,» *Journal of Management Information Systems,* vol. 10, n° %12, pp. 203-225, 1993.

[22] K. Conboy, «Agility from first principles: Reconstructing the concept of agility in information systems development,» *Information Systems Research,* vol. 20, n° %13, 2009.

[23] E. Doval, «Risk Management Process in Projects.,» *Review of General Management,* vol. 29, n° %12, 2019.

[24] C. Guest, W. Ricciardi, I. Kawahi and I. Lang, Oxford Handbook of Public Health Practice, Oxford University Press, 2013.

Appendix I

Detailed functionalities list

ID	Name	Short description	Related scenarios
1	*Resource display*	Interactive metadata resources graph display	Browse
2	*Linked resources*	Metadata resources links to other metadata resources in the graph	Browse
3	*Report error*	Functionality to report errors in metadata resources or request additions	Browse
4	*Edit button*	Edit current resource	Browse
5	*Export button*	Export current resource and linked resources (to a certain depth level) in a selected format (RDF, JSON-LD, XML)	Browse
6	*Search bar*	Search bar field as a starting point to browse the repository	Browse
7	*Search results*	Listing of search results	Browse
8	*Search query*	Translate keywords into a semantic search query (support for: lemmatization and misspelling; standard search strings e.g., AND, OR etc.)	Browse
9	*Results ranking*	Search results ranking algorithm given the users search queries (by relevance, by dates of creation/last update)	Browse
10	*Filtering*	Filtering of search results based on metadata fields (eg. by category, by publisher, by type of study, etc.)	Browse
11	*API*	Services to programmatically query and export contents of the repository	Browse
12	*Recently/mostly visited and newly added*	A list under the search bar of the recently visited, the most visited resources (requires cookies) and the newly added resources	Browse
13	*Metadata form*	Metadata fields form to fill (requires login)	Create/Edit

14	*Save & Close*	Button to finalise metadata filling	Create/Edit
15	*Create new*	Create a new resource button (requires login)	Create/Edit
16	*Edit existing*	Edit existing resources (requires login and ownership or right to edit the resource)	Create/Edit
17	*Cancel*	Button to cancel metadata filling and discard changes	Create/Edit
18	*Mandatory fields*	Specify mandatory fields with a red asterisks and control their filling at save	Create/Edit
19	*Open metadata form to edit*	Open an existing metadata form with its content available for editing	Create/Edit
20	*Save Edits*	Button to finalise form editing (save to database)	Create/Edit
21	*Revert Edits*	Button to discard edits and revert to original content	Create/Edit
22	*Autocomplete*	Query controlled vocabularies for autocomplete (available for search box and metadata fields controlled by specific value sets)	Create/Edit
23	*Verify user role on create and edit*	Verify authorization to create or edit	Create/Edit
24	*Auto save*	Periodically save user inputs while filling the metadata form (available both on newly created forms or during edits of existing forms)	Create/Edit
25	*Notify on save*	Notify the user that modifications were saved	Create/Edit
26	*Edits versioning*	keep versions of metadata edits along the user that made the edits	Create/Edit
27	*Edit template*	Edit existing template structure	Repo. & Template management
28	*Create new template*	Compose a new template from predefined elements	Repo. & Template management
29	*Template elements*	Define template elements types: text, date, list, etc. Define authorized values, and value sets	Repo. & Template management
30	*Save new/edited templates*	Save modifications to templates	Repo. & Template management
31	*Tooltips*	Tooltips to provide descriptions on the fields	Repo. & Template

		when mouse pointer hovers over them	management
32	*Special characters*	Control special characters inputs in all text fields	Repo. & Template management
33	*Objects identifiers management*	Generate and guarantee uniqueness of object identifiers (applies to metadata resources and concepts in vocabularies and controlled sets)	Repo. & Template management
34	*Login page*	Login to the repository	Repo. & Template management
35	*Sign in*	Sign in functionality for a new user (creates a user profile)	Repo. & Template management
36	*Hash passwords*	Save hashed password in database	Repo. & Template management
37	*Edit user profile*	Edit profile, password change	Repo. & Template management
38	*Logout*	Logout button (available from all pages after login)	Repo. & Template management
39	*Access user profile*	Access user profile (available from all pages after login)	Repo. & Template management
40	*Save users interactions*	Save users interaction history in cookie to track used keywords and clicked on metadata resources	Repo. & Template management

Appendix II A

First iteration prototype mockups

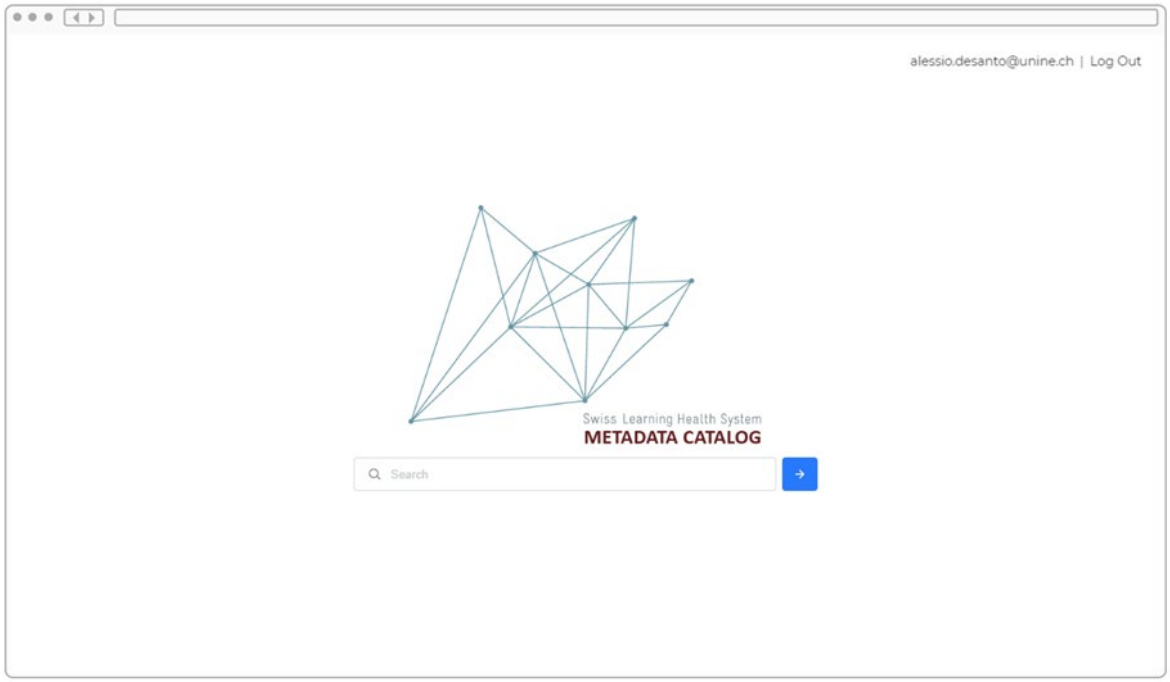


Figure 6. Main window of SH Metadata Repository

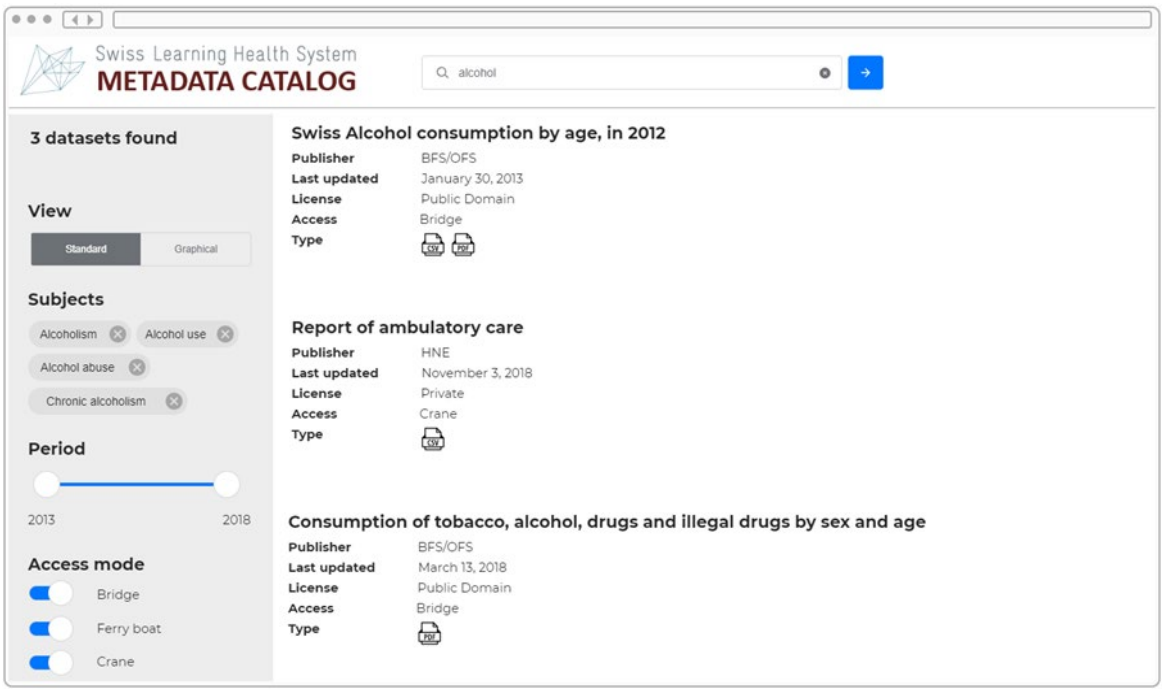


Figure 7. Text view of search results

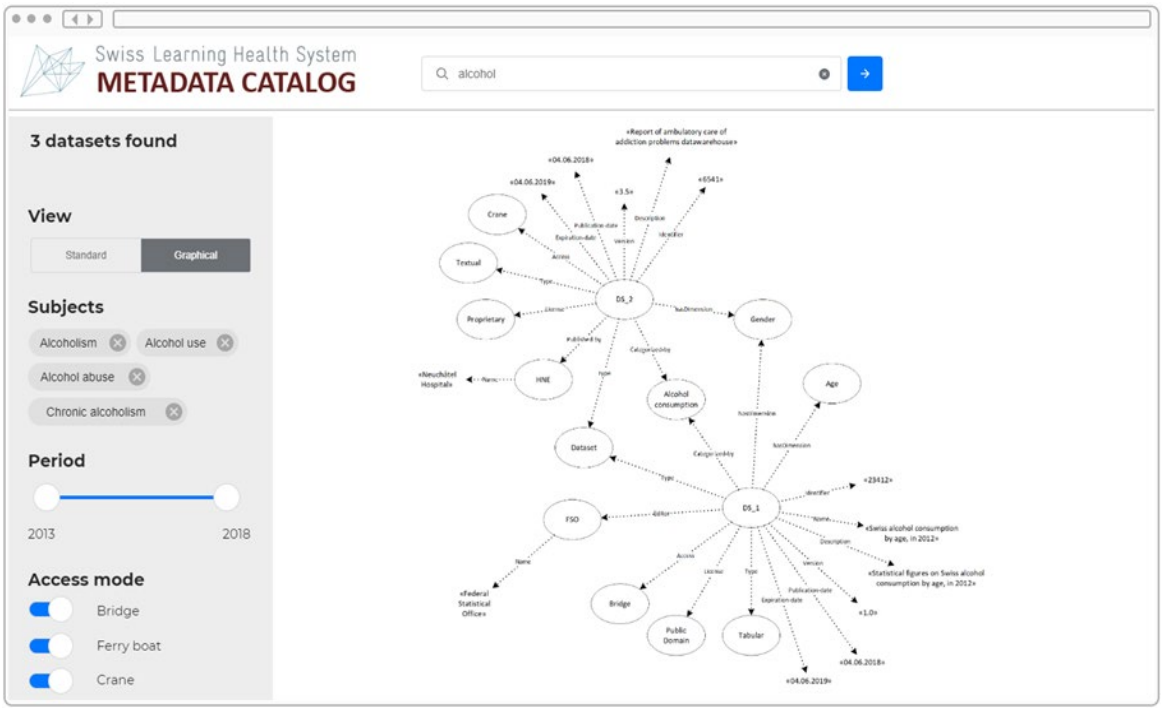


Figure 8. Graph view of search results

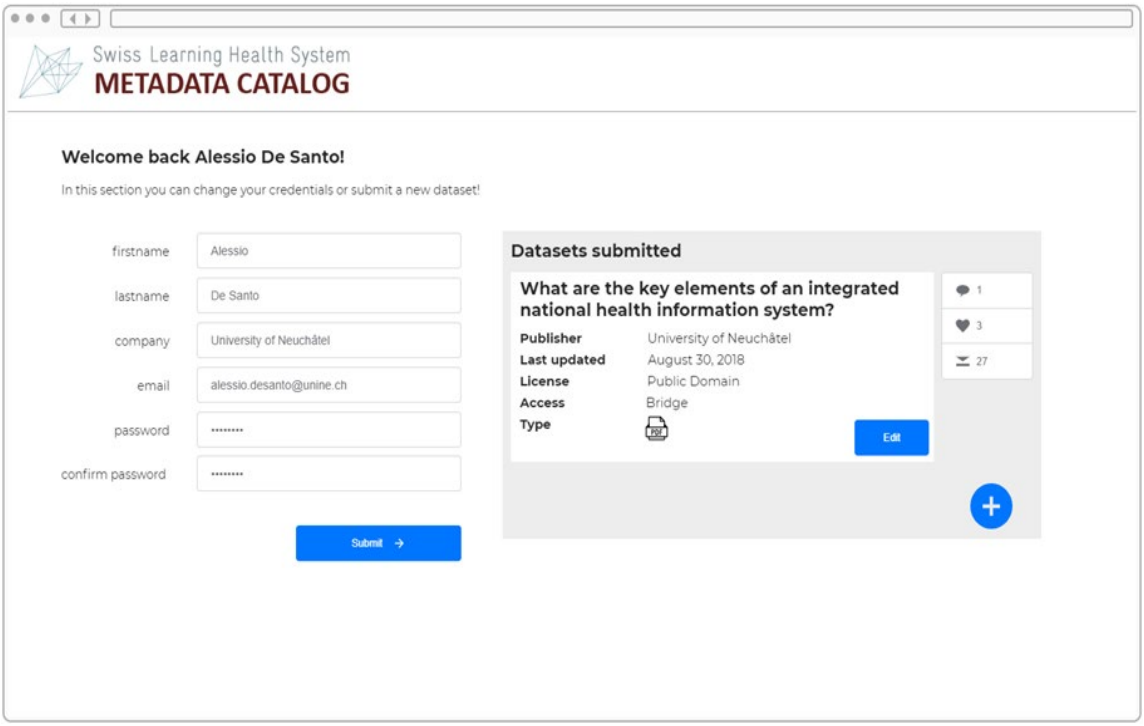


Figure 9. User's credential view

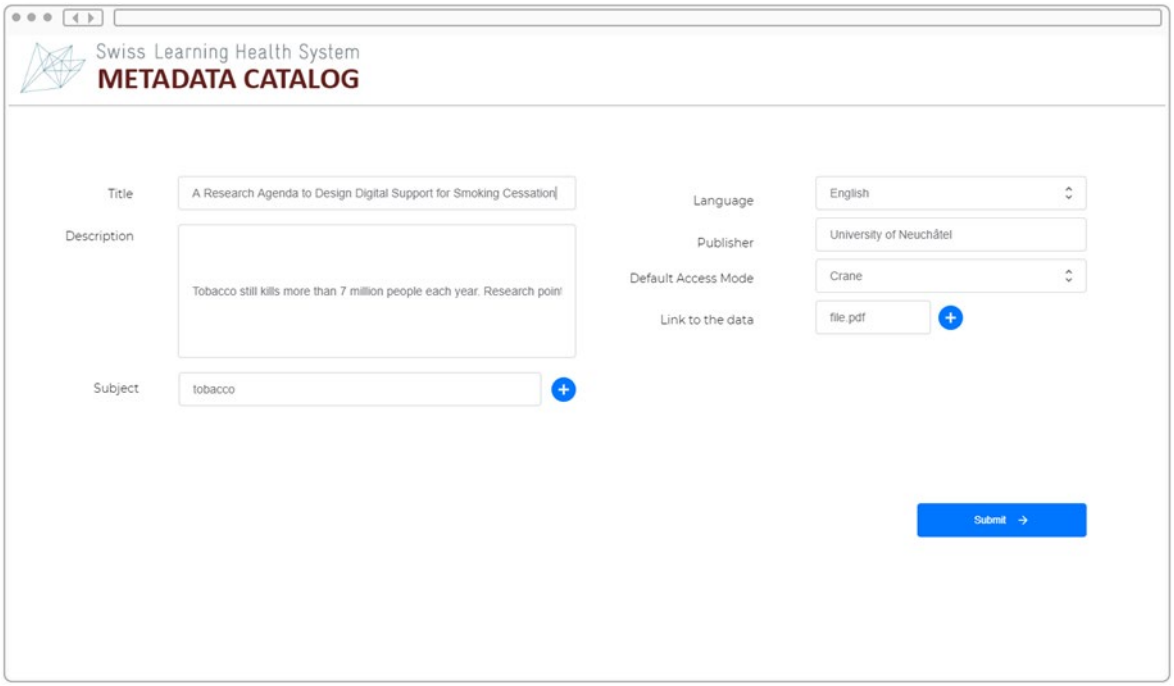


Figure 10. Metadata template window

Appendix II B

Proof-of-concept prototype

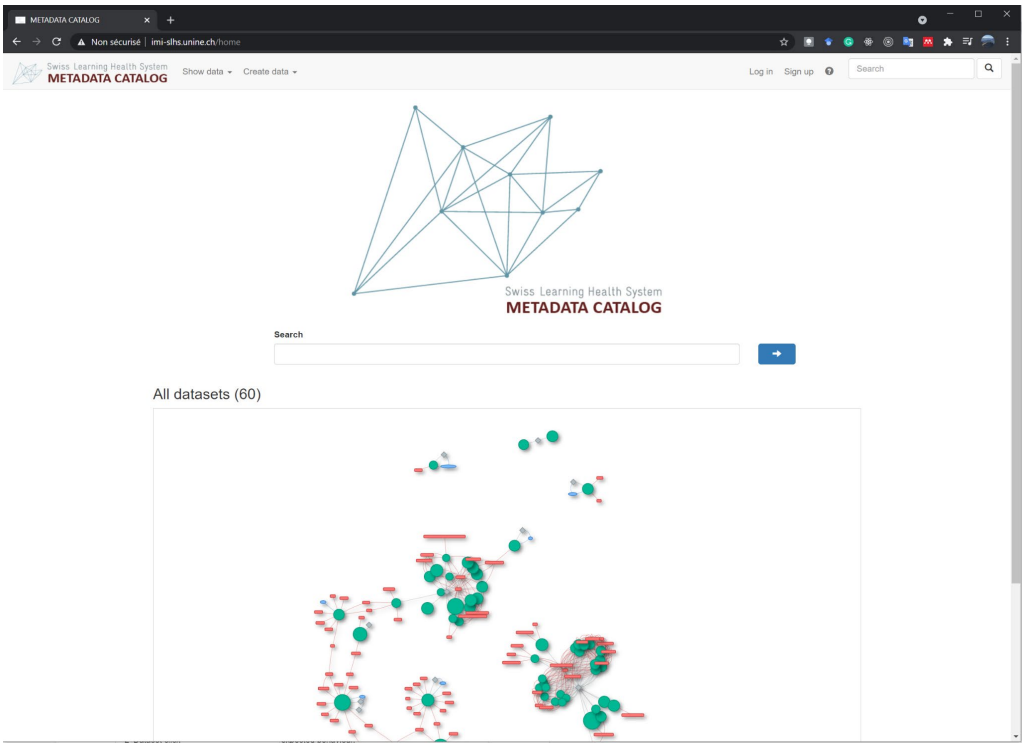


Figure 11. Current main window of SH Metadata Repository

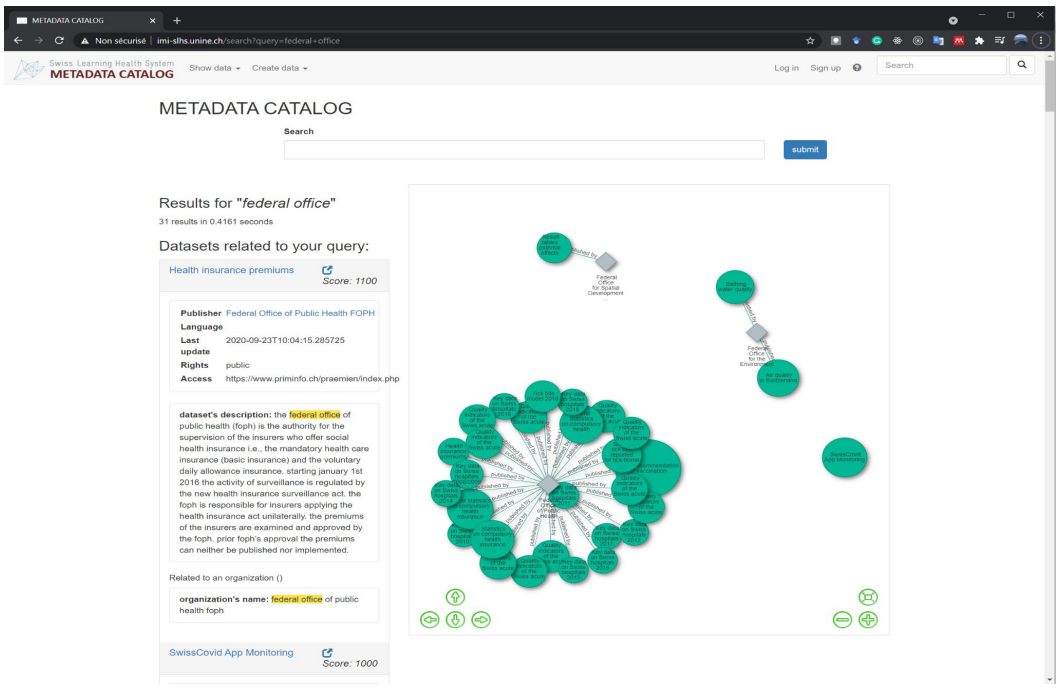


Figure 12. Textual and graphical view of search results

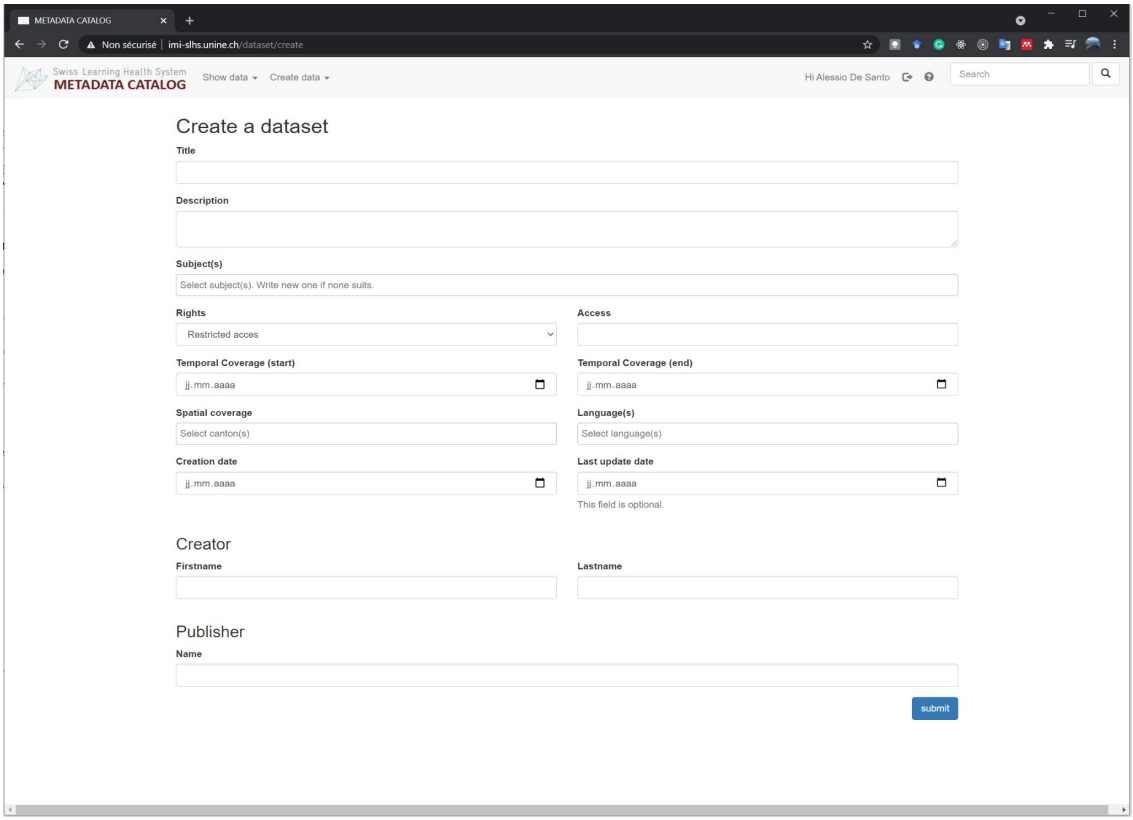


Figure 13. Current metadata template window

Appendix III A

Additional wished functionalities and known bugs

ID	Functionalities and bugs	Description and example
1	*Search lemmatization*	When searching "hospitalization" you should have at least "hospital" results.
2	*Dataset click*	Click on the dataset title closes the box, but it is not an expected behaviour.
3	*Hover popup*	Add contextual help and examples on the various fields displayed.
4	*Add dataset items*	It would be useful to know what variables are stored in the "dataset".
5	*Graph viz should be clickable*	It would be interesting to display datasets by clicking on a tag.
6	*Search keys implementation (AND, OR, ...)*	
7	*Hide score*	
8	*Hide identifier*	
9	*Make results sortable*	For instance by relevance or by date
10	*Welcome on board*	Explain metadata and its usefulness.
11	*Getting started video / About*	To clarify jargon, terminologies but also metadata catalog aim
12	*Legend on the graph visualization*	It would be useful to know the color or size signification of the elements displayed.
13	*Make "access" link clickable*	Actually only the link is provided but it is not clickable
14	*Add file size information*	It would be useful to know if it is just a file or big data
15	*Improve how date are displayed*	Only dd.mm.yyyy would be enough most of the time
16	*Clean redundant tags*	Introduce vocabularies.
17	*Introduce "fil rouge"/filtered search*	Clicks on tags or other elements should always relate to your initial search.

18	*Improve readability of graph visualization*	For instance text on green nodes is not readable enough.
19	*Descriptive information of datasets is expected earlier*	For example magnitude, purpose of the research, …
20	*All hover popups should look the same*	In the create/edit dataset form.
21	*Use same terminology*	E.g. tag vs subject.
22	*Add "Switzerland" in spatial coverage*	In the create/edit dataset form. Maybe "outside Switzerland" also?
23	*Second level of granularity PDF generation*	Provide PDF describing content of dataset.
24	*Have a description for each field*	For instance in the creation of a dataset description, the temporal date attribute is not well understood.
25	*Higher level of granularity of datasets*	Instead of having Hospital Statistics 2015, Hospital Statistics 2016, Hospital Statistics 2017, etc., it should only be Hopital Statistics with the years available related to it.
26	*Add type of dataset field*	For example cross-sectional, longitudinal, etc.
27	*"Did you mean this" Google feature on misspelled search*	
28	*Advanced search by topic, keywords or publishers*	
29	*"Subject" field on the creation of dataset description is misleading*	Replace with keyword or topic.
30	*On landing page we could have "hot" datasets or "last used" one*	
31	*On landing page we could have categories*	
32	*Link datasets to cohort ID*	It could be useful to know if datasets are relative to the same cohort of patients.
33	*Macro/micro-level tags*	
34	*Rights: restricted access is enough?*	What are the restrictions?

35	*Score of the dataset provided by the other researchers*	
36	*Improve search by using MeSH terms*	
37	*Manage versions of datasets*	Health Stats 2012, Health Stats 2017 represent the "same" dataset, but with various versions.
38	*Mark mandatory fields*	
39	*On landing page we could have "hot" datasets or "last used" one*	

Appendix III B

Augmented set of metadata attributes

During the initial evaluation of the prototype using semi-structured interviews, users from researchers and curators personas identified missing information in the metadata that was presented and requested by the prototype. After considering their remarks we propose an updated and extended list of metadata fields. (*Note*: "resource" represents the data source referred/described by a Dataset instance)

Metadata attribute	**Attribute definition**
Title	A concise name that reflects the data resource.
Description	A short description of the resource, summarizing its main characteristics and the characteristics of the project that lead to its collection, if any.
Keywords	A set of short keywords that best describe the content of the resource. These keywords would be proposed by autocomplete, from standard vocabularies and taxonomies, as the user starts to type.
Language	One or more languages that reflect languages used inside the data resource.
Temporal coverage	Explicits the time frame (i.e., start and end) that the content of the data resource concerns.
Spatial coverage	Explicits the geographical zone that the content of the data resource concerns.
Items	Lists the main variables inside the data resource.
Category	Provides a high level category to inform about what the content of resource. Some relevant categories in health research are (partially adapted from [24]): demography/population, risk factors, health needs, mortality/morbidity, health services use, health economics, epidemiology, patient level, medical, legal.
Type	Indicates if the data resource is collected as a routine data collection or special data collection effort (e.g., for a specific study). If routine data is being described, a link needs to be made to a previously existing data resource (if already available). This field could be complemented by a second part indicating the specific type of data collection. Some relevant types are (adapted from [24]): [Quantitative-Observational] cross-sectional, longitudinal, cohort, case-control, descriptive, ecological; [Quantitative-Experimental] randomized controlled trial, field trial; [Qualitative] ethnography, case-study, action-research, grounded-theory.
Access	Provides an indication on how to access the resource. It could be a link to where the dataset could be found and requested or directly downloaded.

Rights	Indicates if the resource is publicly available or is restricted (e.g., upon request).
Publisher	Indicates the organization or the person responsible for making the resource available.
Detailed information	Allows uploading a file (e.g., PDF) containing more in depth information about the data collection process, links to codes for data processing, the coding of variables and others. Those documents could be opened by the researchers during browsing to assess the usefulness of a resource before requesting access to it.
Creator	Indicates the organization or the person that created the dataset. This entity is not necessarily the same as the publisher.
Size	Indicates the approximate size in mega/gigabytes of the dataset file.

The presented list remains minimal and presents a possible common denominator between the different needs of health researchers with different backgrounds and scientific expertise.

Appendix IV

Improved prototype mockups

The interactions of users with the prototype have shown inefficiencies in the presentation and disposition of the elements in the user interface. We take their remarks into account and we propose here updated mockups for the prototype interfaces.

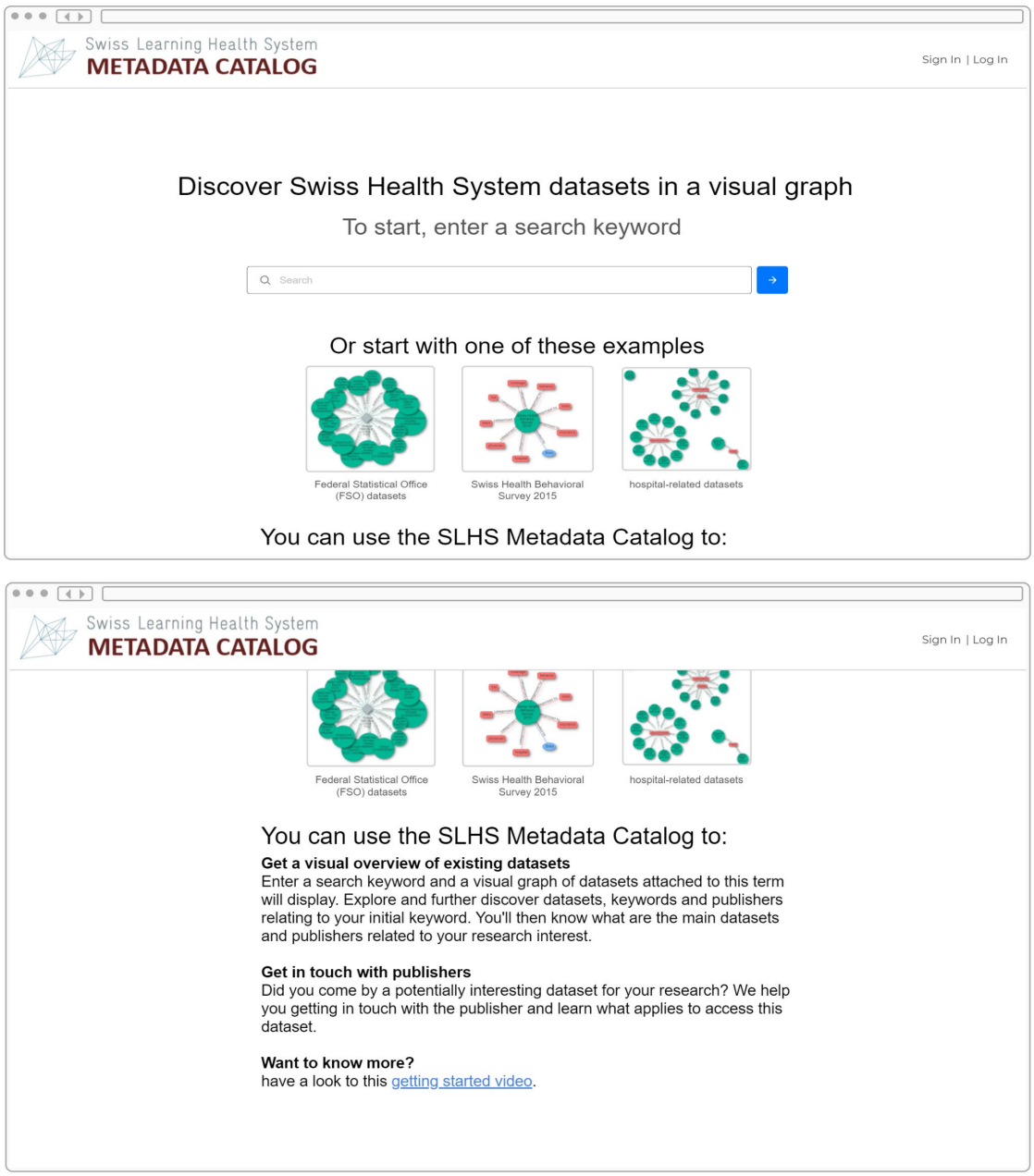


Figure 14. Improved main window of SH Metadata Repository

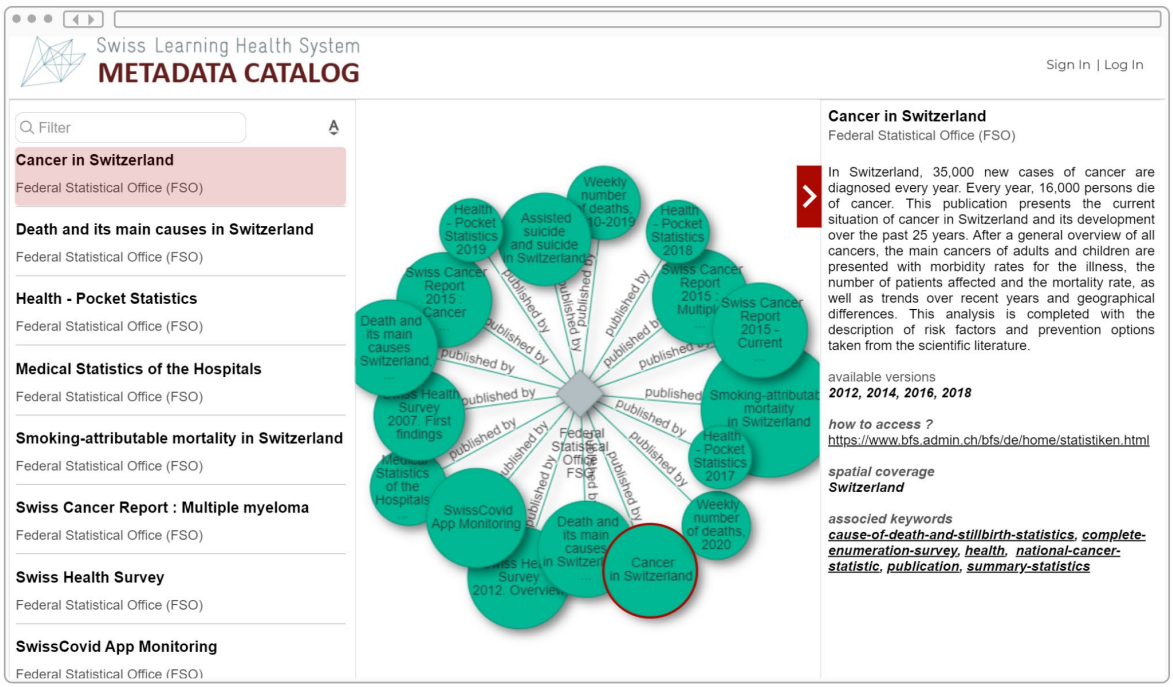


Figure 15. Improved textual and graphical view of search results

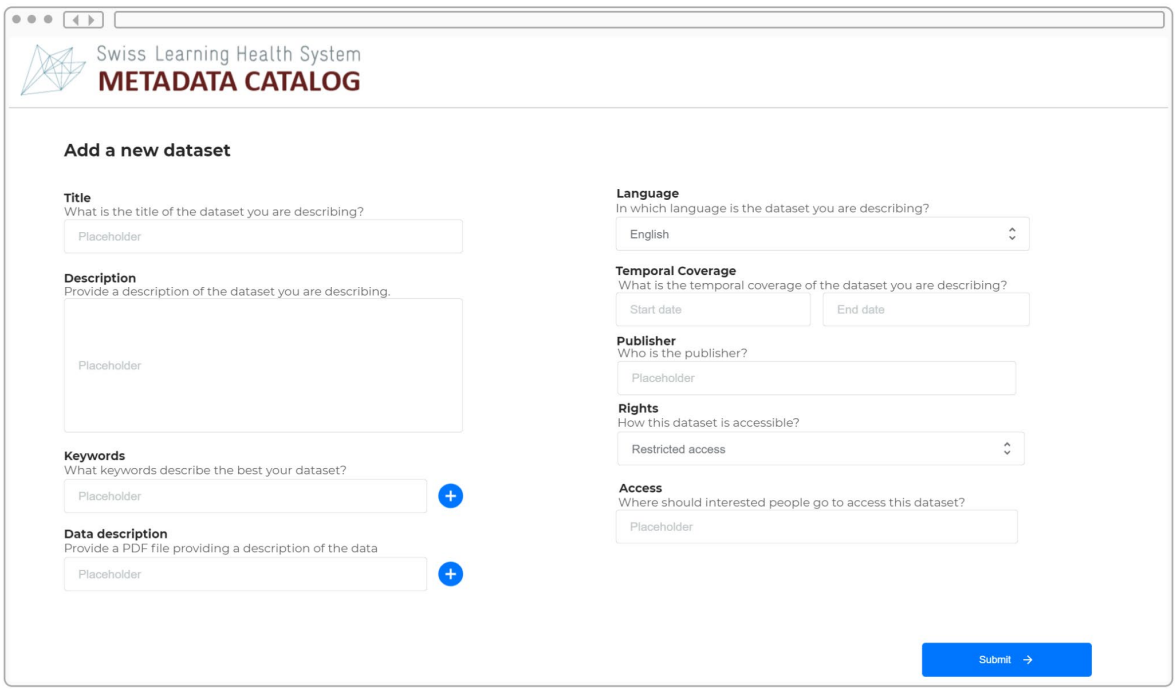
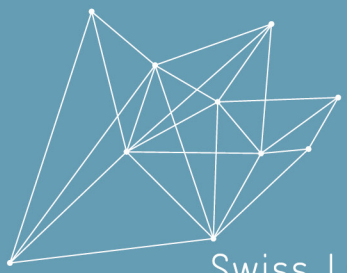


Figure 16. Improved metadata template window



Swiss Learning
Health System