

CatRIS

Catalogue of Research Infrastructures Services

Horizon 2020 CSA

WP5

CatRIS gateway and API development

Deliverable 5.1

Gateway functional specifications

Main Author(s):

George Papastefanatos, NKUA

Giorgos Giannopoulos, NKUA

Due date of deliverable: 31.08.2019

Actual submission date: 20.09.2019

Dissemination level: PU



Funded by
the European Union

List of Contributors

Partner	Acronym	Partner name	Name of contact
1	ESF	European Science Foundation	A.Helman, A.Tahvanainen
2	EFIS	European Future Innovation System Centre	V.Cvijanovic
3	JNP	JNP Strategy & Management Consulting	J.Sanchez, P.Spyrou
4	CTLS	Core Technologies for Life Science Association	A.Kamenskaya
5	NKUA	National Kapodistrian University of Athens	G.Papastefanatos, G.Giannopoulos

Document Revision History

Date	Version	Author/Reviewer	Summary of changes
29.07.2019	v0.1	G. Giannopoulos, G. Papastefanatos	ToC and first set of functional specifications
01.08.2019	v0.2	G. Papastefanatos, G. Giannopoulos	Finalized set of functional specification
05.08.2019	v0.3	G. Papastefanatos, G. Giannopoulos	Data model
09.08.2019	v0.4	G. Giannopoulos, G. Papastefanatos	Architecture and functionality description
16.08.2019	v0.5	G. Giannopoulos, G. Papastefanatos	First complete draft of the document
21.08.2019	v0.6	G. Papastefanatos, G. Giannopoulos	Version ready for internal review
05.09.2019	v0.7	A.Helman, A.Tahvanainen, A.Kamenskaya, V.Cvijanovic, J.Sanchez, P.Spyrou	Incorporation of comments/suggestions
11.09.2019	v0.8	G. Papastefanatos, G. Giannopoulos	Final version for submission to EC

Copyright © CatRIS consortium, 2019

This document contains material, which is the copyright of certain CatRIS consortium members and the EC, and may not be reproduced or copied without permission, except as mandated by the European Commission Grant Agreement no 824173 for reviewing and dissemination purposes. The information contained in this document is provided by the copyright holders "as is" and any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall the members of the CatRIS collaboration, including the copyright holders, or the European Commission be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or other-wise) arising in any way out of the use of the information contained in this document, even if advised of the possibility of such damage.

TABLE OF CONTENTS

- Summary 7**
- Introduction..... 8**
- 1 The CatRIS Data Model 9**
- 2 Architecture of the CatRIS gateway..... 11**
 - 2.1 Data Layer11**
 - 2.1.1 Data Schema12
 - 2.1.2 Data Management.....13
 - 2.2 Application Layer15**
 - 2.2.1 Search16
 - 2.2.2 Metadata Retrieval16
 - 2.2.3 Analytics17
 - 2.2.4 User Feedback.....17
 - 2.2.5 Recommendation17
 - 2.2.6 Notification18
 - 2.2.7 Provider Communication18
 - 2.2.8 Service Registration18
 - 2.2.9 Service Monitoring18
 - 2.2.10 User Registration/Authentication19
 - 2.2.11 Provider Registration19
 - 2.2.12 User Profiling19
 - 2.2.13 Resource Management.....20
 - 2.2.14 Validation.....20
 - 2.3 User Interface (UI) Layer20**
 - 2.3.1 Service Catalogue Facilities20
 - 2.3.2 User and Service Management.....21
 - 2.3.3 Platform Administration22
 - 2.4 Interoperability Layer22**
- 3 Functional specifications..... 24**
 - 3.1 CatRIS Gateway specifications24**
 - 3.1.1 Catalogue Browsing and Searching24
 - 3.1.2 End-User Interaction28
 - 3.1.3 Service/Resource Provider Interaction32
 - 3.1.4 Visualisation and Analytics.....36
 - 3.1.5 Catalogue Interoperability37

- 3.1.6 Administrator functionality39
- 3.2 Discussion on user requirements for further consideration40**
- 3.3 User roles and functionality access rights43**
- 3.3.1 Unregistered users43
- 3.3.2 Registered RI users44
- 3.3.3 Service Provider (SP) users44
- 3.3.4 Administrator users44
- Conclusions 47**

List of Tables

Table 1: Template for the description of a functional specification	24
Table 2: FS-CBS-01 - View classification information of services/resources.....	25
Table 3: FS-CBS-02 - Search services/resources using keywords	25
Table 4: FS-CBS-03 - View and search services/resources using filters.....	25
Table 5: FS-CBS-04 - View detailed characteristics of a service/resource.....	26
Table 6: FS-CBS-05 - Navigate to the service/resource at the remote service provider.....	26
Table 7: FS-CBS-06 - Compare a list of services/resources w.r.t. a set of attributes	26
Table 8: FS-CBS-07 - Sort service/resource search results.....	27
Table 9: FS-CBS-08 - View number of service/resource requests	27
Table 10: FS-EUI-01 - Register as an authenticated user.....	28
Table 11: FS-EUI-02 - Login/Logout as authenticated user	28
Table 12: FS-EUI-03 - Edit user profile.....	28
Table 13: FS-EUI-04 - Delete user profile.....	29
Table 14: FS-EUI-05 - Add/remove favourite services.....	29
Table 15: FS-EUI-06 - Add/remove favourite category	29
Table 16: FS-EUI-07 - Subscribe/Unsubscribe to catalogue updates and alerts.....	30
Table 17: FS-EUI-08 - Get recommendations for new services based on similar users	30
Table 18: FS-EUI-09 - Get recommendations for new services/resources based on similar services.....	30
Table 19: FS-EUI-10 - View list of services/resources based on user profile	31
Table 20: FS-EUI-11 - Allow different views of services/resources for different user groups.....	31
Table 21: FS-EUI-12 - Query service providers for details on selected services/resources	32
Table 22: FS-EUI-13 - Book a service for a specific timeslot	32
Table 23: FS-SPI-01 - Register as provider	33
Table 24: FS-SPI-02 - Login/Logout as provider.....	33
Table 25: FS-SPI-03 - Register a service/resource.....	33
Table 26: FS-SPI-04 - Update a service/resource	34
Table 27: FS-SPI-05 - Delete a service/resource.....	34
Table 28: FS-SPI-06 - Publish or unpublish a service/resource	34
Table 29: FS-SPI-07 - View list of services/resources associated with the provider.....	35
Table 30: FS-SPI-08 - View log of actions applied to a service/resource	35
Table 31: FS-SPI-09 - View statistics and visualizations on a service/resource	36
Table 32: FS-SPI-10 - Publish and update service availability calendars.....	36
Table 33: FS-VA-01 - Visualize service/resource distribution.....	36
Table 34: FS-VA-02 - Visualize classifications of services/resources with a colour coding ...	37
Table 35: FS-CI-01 - Programmatically insert a new service/resource.....	37
Table 36: FS-CI-02 - Programmatically update a service/resource	38
Table 37: FS-CI-03 - Expose service/resource data in third parties	38
Table 38: FS-AF-01 - Approve/reject a provider registration	39
Table 39: FS-AF-02 - Add/remove a provider user	39
Table 40: FS-AF-03 - Approve/reject a service/resource registration.....	39
Table 41: FS-AF-04 – Moderate user comments and actions	40
Table 42: FS-AF-05 – Manage vocabulary/classification content.....	40
Table 38: User requirements for further consideration	42
Table 39: Mapping of Functionality to User Roles.....	44

List of Figures

Figure 1: The CatRIS Data Model	9
Figure 2: Architecture of the CatRIS gateway.....	11

Acronyms

Abbreviation	Meaning
AAI	Authentication and Authorisation Infrastructure
API	Application Programming Interface
EOSC	European Open Science Cloud
JMS	Java Message Service
RI	Research Infrastructure
SDT	Service Description Template
SP	Service Provider
UI	User Interface
UR	User Requirement

Summary

CatRIS aims at developing a portal (gateway¹) for registering and discovering all available Research Infrastructure (RI) services across Europe. The portal will allow the harmonized registration and management of RI services and will facilitate their retrieval by end users of different disciplines and expertise levels. It will instantiate a well-defined schema and respective vocabularies that will allow service providers (SP) to describe their services in a harmonized, detailed and informative manner. Further, it will provide a rich set of facilities for searching, comparing, visualizing statistics and getting recommendations on services and contacting their providers. Finally, it will implement proper APIs for programmatically ingesting or updating services, exchanging data with existing related catalogues (eInfraCentral, MERIL) and exposing service information from the CatRIS catalogue to external, third-party applications.

WP5 handles the development of the CatRIS portal, as well as its evaluation through user assessment studies. Specifically, in the frame of WP5: (a) the architecture and functional specifications of the gateway are prescribed based on the results of an extensive requirements elicitation process taken place in WP3 and WP4, (b) the gateway framework is implemented, deployed, populated and tested in a phased manner, producing three releases, through the course of the project, while (c) user assessment studies continuously guide the improvement of the catalogue and its alignment with the requirements of RI users, RI managers/providers and Policy makers.

Deliverable 5.1 presents the functional specifications for the CatRIS portal, the data model for the content stored in the gateway, as well as the architecture of the gateway. Functional specifications prescribe in detail the core functionalities of the gateway. Each specification is mapped to the respective user requirement (UR) it derived from, as recorded in D4.4 "Functional and non-functional requirements". The functional specifications are categorized per major functionalities offered by the catalogue, as well as per user role of the CatRIS portal.

The data model of the CatRIS catalogue instantiates the most current version of the Service Description Template, SDT v2.0.0 (D4.3 "Service Description Template"), which provides a thorough, structured description of the major entities that need to be handled in the catalogue (services, service providers, options, etc.). The catalogue integrates the SDT in a largely schema agnostic manner, ensuring its adaptability to potential evolution of the SDT. Finally, the gateway architecture presents the four major components of the catalogue (Data, Application, User Interface and Interoperability layers), discussing the supported functionality by the components and their modules.

The presented work complies to the workplan prescribed in the Description of Work and follows the implementation timeline of the project without deviations. The content of the deliverable is organized as follows.

Section 1 presents the CatRIS data model, briefly discussing the major resources of the catalogue and their relations.

Section 2 presents the architecture of the CatRIS gateway and discusses in detail the functionality of the individual components that comprise it.

Section 3 presents the functional specifications of the gateway and provides a short discussion on them, as well as a mapping of functional specifications to user roles in the CatRIS portal.

The final section concludes the deliverable, summing up its content and discussing the next steps of work in WP5.

¹ The terms *gateway* and *portal* are used interchangeably.

Introduction

The major goal of CatRIS consists in designing and developing a catalogue that will facilitate the harmonization and discoverability of RI services, aspiring to become the central point for registering, searching and comparing RI services across Europe. To achieve these goals, the CatRIS portal needs to be developed taking into account the needs of a large number of heterogeneous stakeholders, regarding both their disciplines, since RI services span a wide range of scientific domains, and their roles (RI users, service providers/RI managers, policy makers). Additionally, CatRIS needs to exploit background work, technologies and insights gained from previous cataloguing efforts (MERIL, eInfraCentral), as well as from the European Open Science Cloud initiative that promotes, among others, the harmonization, reutilization and interoperability on models and frameworks. At the same time, CatRIS needs to reutilize and augment existing content regarding RI services and providers and provide well-defined and rich APIs to service providers and third parties for populating, updating and obtaining content from the CatRIS platform.

This deliverable presents the work performed towards designing the CatRIS platform and prescribing the core set of supported functionalities. First, the data model that forms the basis for organizing the catalogue's content is presented. The data model instantiates the modelling of CatRIS entities that is presented in D4.3, while it enriches it with additional, auxiliary entities regarding events, statistics and KPIs logged and measures by the platform. Then, the architecture of the CatRIS platform is presented and its major components are discussed. The architecture comprises a four-layer scheme, with three core layers (Data, Application and User Interface) implementing the core functionality of the platform and a horizontal layer (Interoperability) implementing the APIs for the communication between the core layers and between the platform and external, third parties.

Finally, the functional specifications of the platform are formally prescribed and mapped to user roles of the portal. Further, a discussion on how these specifications will instantiate in the platform and will optimally satisfy end user needs is presented. The functional specifications mainly build on the user requirements that were prescribed in D4.4, while they also incorporate insights and findings directly from D3.3 "Report on the results of the survey on requirement elicitation" and D3.4 "Concept note for CatRIS", thus taking account all requirement elicitation processes performed in the frame of CatRIS and the derived documents.

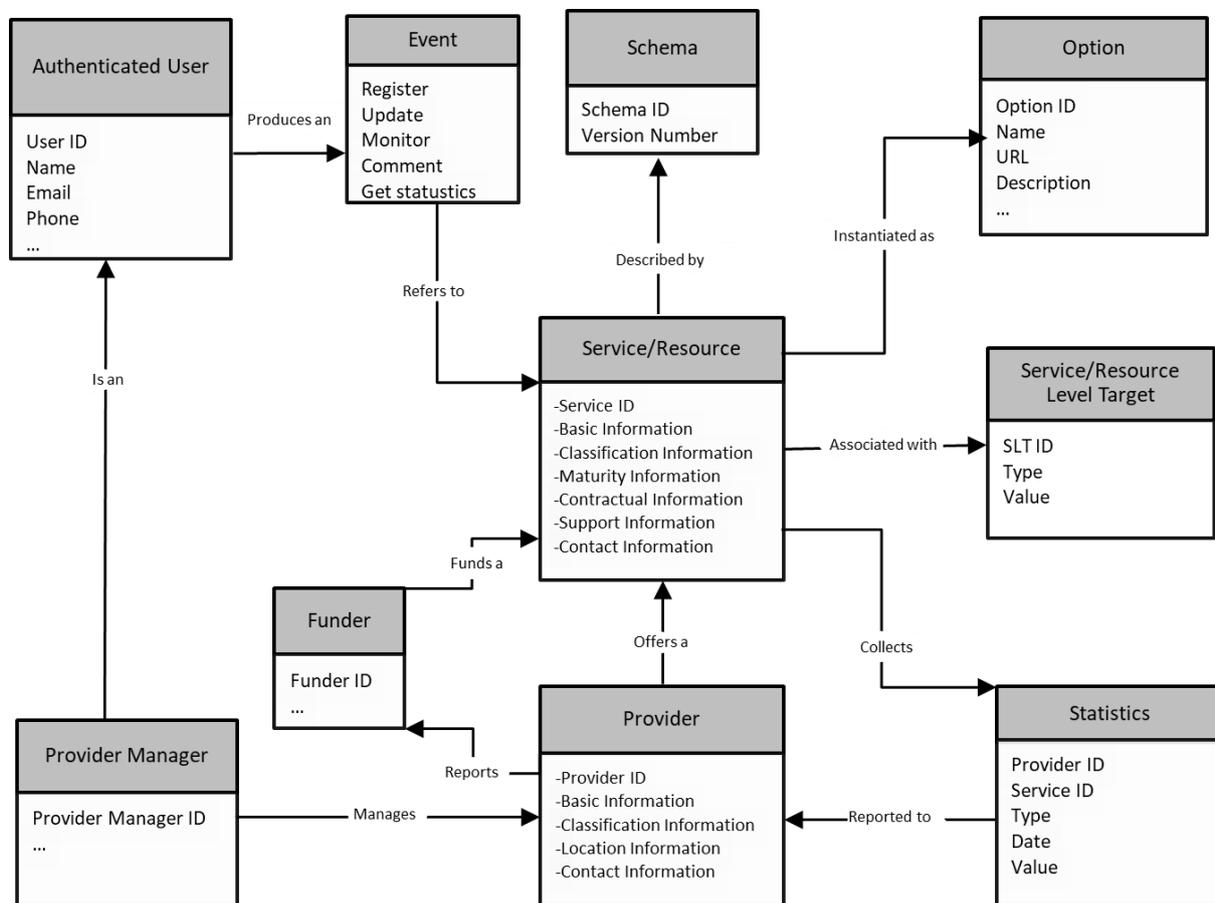
This document presents the architecture and functionality of the CatRIS portal as formed for the first release of the platform, which is documented in D5.2 "1st prototype of the gateway with the service catalogue browsing functionality and the registration of services". However, the development of the portal will employ an agile process where each release will incorporate additions, changes and enhancements that will also take into account stakeholder feedback derived via user assessment processes initiated after the first release of the CatRIS portal. Given that, several architectural decisions and specification might change or be reprioritized following this feedback. These changes will be reported in upcoming deliverables of WP5 and, in particular, in D5.5 "Documentation of the gateway functionality and the APIs".

1 The CatRIS Data Model

Facilitating RI users into discovering useful services and exploring their characteristics/metadata is a major goal of CatRIS. At the same time, providing a gateway for all service providers to register their services in a harmonized way and to be able to manage and monitor them is of crucial importance towards establishing a central endpoint for RI users and service providers, and strengthening the collaboration and communication throughout RI stakeholder communities and disciplines.

Given the above, Service/Resource and Provider comprise two significant entities within the CatRIS model. However, there exist several auxiliary entities that need to be separately defined, in order to formulate a complete and well-defined model that optimally represents the catalogue’s resources, their attributes and their relations. Figure 1 presents these entities and their relations. We note that the full set of attributes for the major entities in CatRIS are thoroughly presented in D4.3.

Figure 1: The CatRIS Data Model



As mentioned above, the main entity in CatRIS is the Service/Resource. A Service/Resource is identified by a persistent unique ID, which is generated by the CatRIS gateway during service registration. Further, it is described by a rich set of attributes, regarding its classification, characteristics, offerings, contact and contractual information, etc. Since CatRIS emphasizes on the harmonization and discoverability of services/resources, these must comply to a certain schema, regarding the mandatory attributes they satisfy, as well as the values they receive for certain attributes. This is ensured by a Schema entity, which describes the Service/Resource and against which a new or updated Service must be validated before it is registered/modified in the catalogue.

To allow for the same Service/Resource to be offered in different options, an Option entity is defined, which is essentially used to describe different instantiations of a Service. Additionally, a Service is associated with one or more Service/Resource Level Targets via measurements/indicators for the specific Service/Resource.

A Service/Resource is offered by a Provider resource, identified by a provider ID. Provider also comprises a first-class citizen in the CatRIS model, which is used for offering to provider users several functionalities (e.g. search, analytics, ratings, etc.). A Service/Resource is also associated with one or more Funders, which have funded the release of the Service/Resource.

An Authenticated (Registered) User is a user who can login in the CatRIS portal and generate events for a service/resource, such as rate a Service/Resource, add a Service/Resource to the favourites, update a Service/Resource, etc. An Authenticated User may belong to a Provider meaning they would be authorised to manage and monitor the Services/Resources of that Provider.

Finally, a set of Statistics are collected for a Service/Resource from the user events. These Statistics are reported to the Provider, who later reports them to the Funder.

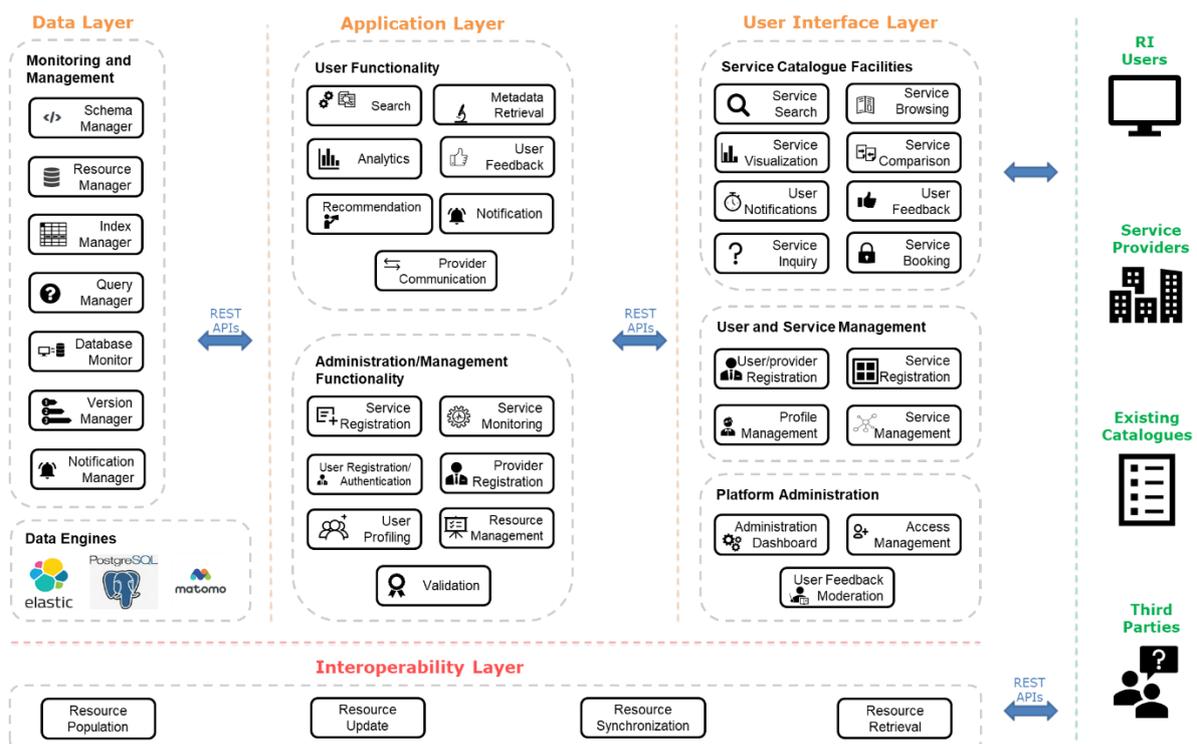
2 Architecture of the CatRIS gateway

The architecture of the CatRIS portal is presented in Figure 2². It comprises three tiered layers, with an additional, horizontal fourth layer, as presented next:

- **Data Layer** manages the storage and indexing of all the content handled by the portal, including the various types of entities, logs and statistics on user interaction with the portal, as well as content produced by computations and analysis performed on the above data.
- **Application Layer** handles the core functionality of the platform, including user and service management, service search, analytics, and alerts and recommendations.
- **User Interface (UI) Layer** exposes the functionality of the CatRIS platform to the end users.
- **Interoperability Layer** implements APIs for populating and updating the CatRIS catalogue, integrating the catalogue with related catalogues and exposing data to third parties.

In the following subsections, each of the four layers is presented in detail.

Figure 2: Architecture of the CatRIS gateway



2.1 Data Layer

Section 1 presented the data model of the CatRIS catalogue, making evident the need to organize and manage a diverse set of entities (e.g. Service/Resource, Provider, Option, etc.).

² We note that, for presentation purposes, “Service/Resource” is represented by the term “Service” in the CatRIS architecture figure, as well as in the following detailed discussion.

Further, most of the entities are described by rich metadata (attributes), several of which need to be searchable by the end users. Consequently, the Data layer needs to be able to manage entities of diverse and potentially evolving schema, as well as to efficiently store and index their metadata, in order to satisfy end user functionality. Next, we discuss how the Data layer: (a) handles the schema of the entities and (b) stores, indexes and monitors the entities and their metadata.

2.1.1 Data Schema

Each of the handled entities in CatRIS is described by a certain schema to which it must conform, so as the integrity and quality of the catalogue content is maintained. To this end, the Data layer needs to be schema agnostic, that is, it needs to support the addition and deletion of entity types, as well as changes in their schema. This is implemented by defining two core objects: *Entity Type* and *Entity*. The former represents the schema of an entity while the latter represents the instantiation of the schema into individual entities of the same type. Next, the attributes of these two objects are provided.

Entity Type:

- **Name:** the name of the entity type, which must be unique among all the entity types, e.g., a service type, a user type, etc.
- **Format type:** the format of the entity (XML or JSON). While the Data layer supports both formats, the metadata of each entity must be expressed in one format, XML or JSON.
- **Schema:** the schema of the entities (represented in JSON Schema). The schema is used when an entity is added or modified for validation and to ensure that all stored entities are valid.
- **Indexed Fields:** the underlying storage makes no assumptions on the schema and contents of the stored entities; thus, it is necessary for the Data layer services extracting information from each entity type to allow the execution of keyword search queries. An example of Indexed Fields are the description and the scientific domain attributes of a Service entity. While a Service entity may contain various metadata, the users need to search services (at least) by their description or scientific domain. This is accomplished by specifying Indexed Fields, which contains information about the searchable fields, such as the name of the field, its type (a string, a number, etc.) and the location of the value in the resource (JSON Path). When new entities are added or existing are modified, the Data layer uses this list in the Indexed Fields to extract the values and store them in an index to allow for efficient query answering.

Entity:

- **ID:** a unique identifier, which is assigned by the registry to each new entity, e.g. a service ID or a user ID.
- **Entity Type:** the type of the entity. Before attempting to store an entity in the registry, its Entity Type must already be registered.
- **Payload:** the actual contents of the entity. It is either an XML or a JSON file, depending on the payload type of the respective entity type. When storing an entity, instead of the actual payload, a URL of the payload can be provided.
- **Creation and modification date:** the dates when this entity was inserted in the registry and when it was last modified.
- **Version:** whenever an entity is modified, its version number is incremented and the previous versions are retained. Only the latest version of an entity is available through the search APIs and the users can access the previous versions only by using the version API.

Having defined the above two objects, the CatRIS catalogue supports the storage and management of entities of arbitrary schema. The process for populating the database with a new type of entity is then straightforward: The schema of the entity, in the form of JSON schema, is inserted as a new Entity Type. Within the Entity Type, several attributes of the schema may be marked for indexing, rendering them searchable. Then, the database can be populated with new records (entities) conforming to the defined schema. In case the schema of an entity type changes, then the respective Entity Type entry in the database can be replaced by the updated schema object. The conformance of the respective entities to the updated schema can then be validated through mechanisms described in the following subsection, and, eventually, the entities can be updated too, so that they conform to the updated schema.

2.1.2 Data Management

Both the Entity Type and the Entity objects are stored as JSON documents within the database. JSON is at this moment a widely adopted, lightweight data representation format that suffices for our data and schema representation purposes in CatRIS, by introducing minimum storage and processing overhead. Further, the adopted DBMS, PostgreSQL³ natively supports storing and indexing JSON objects, further facilitating the adoption of JSON format and JSON Schema.

CatRIS manages a diverse set of entities, which carry a set of metadata describing them, and which must be easily discoverable by end users. The Data layer contains a set of general-purpose modules responsible for storing and managing the metadata of the various resources. Additionally, several of the attributes of the resources need to be individually indexed, so that they can efficiently be retrieved, in order to satisfy the rich searching, browsing, analytics and visualization functionalities offered by CatRIS. Finally, certain notification and cascading mechanisms need to be available to trigger on specific types of changes in the database (e.g. deletion of a Provider, modification of an Entity Type).

Next, we present the core modules of the Data layer that handle the aforementioned functionality.

- **Schema Manager** is responsible for registering entity types, validating the provided JSON schema, and preparing the database to accept new instances of the new types. Further, it communicates with the search engine component (ElasticSearch⁴) and the Index Manager, so that specific attributes declared in the Entity Type are indexed for keyword search. Finally, in case of an Entity Type modification, it communicates with the Resource Manager to request/verify appropriate changes to the respective entities, in order to maintain the validity and integrity of the database.
- **Resource Manager** manages entities by validating new entities against the schema (drawn by the respective Entity Type object) and the constraints of the database (referential, primary keys and unique values), and by registering/storing them in the database. The Resource manager is also used to update and delete existing entities. Further, it is triggered every time an Entity Type is updated, so as to re-validate the respective entities, and alert the platform administrator for potentially required updates on the entities.
- **Index Manager** is responsible for updating the full text index of entities maintained in ElasticSearch, as well as the database indexes on specific fields of the entities, maintained internally in PostgreSQL. The full text index allows the execution of keyword queries on the resources providing results much faster than the relational database that

³ www.postgresql.org

⁴ www.elastic.co

is used to store the entity types and entities, and allowing advanced keywords search facilities. By responding to notifications from the Database Monitor, the Index Manager keeps the underlying index synchronized with the contents of the database. On the other hand, the database indexes are utilized for retrieving frequently queried attribute values of the stored entities, for example frequently used attributes for faceted service search, frequently queried aggregate statistics on the distribution of services and service providers, etc.

- **Query Manager** is responsible for performing queries to the proper storage component and the respective indexes. In case of keyword search queries, it forwards them to the search engine indexes (ElasticSearch), based on the Index Fields defined in the Entity Type for the specific entities, and returns the results. In case of filtering queries or queries calculating aggregate statistics on the entities' metadata, the Query Manager executes the queries on the database component (PostgreSQL), exploiting the database indexes that are built on specific (frequently used) attributes of the entities.
- **Database Monitor** monitors the database for any changes in registered entity types or entities and notifies other modules that are interested in these changes. For example, the Version Manager updates the version of entities when it receives a notification by the Database Monitor. These notifications from the database monitor to other modules allow for faster completion of the basic functionalities of the Data layer (i.e. CRUD operations on entity types and entities), since the more computationally heavy operations (indexing, versioning, etc.) occur asynchronously.
- **Version Manager** manages the different versions of the entities. Whenever an entity is updated, the version manager stores the previous version and assigns a new version number to the modified entity. Using the Version Manager, users can trace the evolution of an entity from the moment it was created up to its latest version.
- **Notification Manager** provides the connection between the Data layer and the Notification service in the Application layer (see Section 2.2) of the platform. Whenever a new entity type is registered, the Notification Manager creates a new JMS topic (e.g. "registry.service.create" or "registry.service.update" for insertions and updates of entities of type Service respectively) and when entities are created, updated or deleted the Notification Manager sends new messages in the respective topic. The Notification Manager allows for any other entity in the platform to be asynchronously notified about the changes in the Data layer without explicitly interacting with it. Again, as is the case with the previous two modules, the Notification Manager is notified for the changes in the entities by the Database Monitor.
- **Statistics Component** is responsible for logging actions regarding specific entities of the database (e.g. services, service providers) and periodically computing statistics on them. These statistics include the numbers views or orders of services, the number of times specific attributes or attribute values were applied for faceted search, the number of visits to service providers' homepages, etc. Such statistics are periodically updated and stored, and are exploited in analytics and visualization, as well as in personalized facilities of the catalogue. The Statistics Component consists in an *Matomo*⁵ instance. It communicates with the Analytics module and makes available all gathered user feedback for further aggregation and processing.
- **Database Component** consists in a *PostgreSQL* database instance, where the proper database schema for storing the CatRIS content is implemented (including the required tables, keys, indexes and triggering mechanisms to support the functionality previously described) and populated with the catalogue's content.

⁵ <https://matomo.org/>

- **Search Engine Component** consists in an *ElasticSearch* engine instance that indexes the proper (defined in the Entity Type) entities' attributes, so as to support advanced keyword search functionality on them.

2.2 Application Layer

The Application layer implements the core functionality of the CatRIS gateway. It serves as the middleware between the Data layer and the UI and Interoperability layers, handling the transfer of content between these layers, as well as the querying, computations and analytics required for exposing the CatRIS catalogue functionality. The Application layer consists of a set of modules that can be categorized in two distinct groups: *User-oriented functionality* and *Administration functionality*. Briefly, the core modules within the two groups are:

User-oriented functionality:

- **Search Module** manages the keyword search functionality that allows users to retrieve services based on keywords that match their metadata descriptions.
- **Metadata Retrieval Module** manages all the remaining retrieval functionality, regarding metadata of all the entities that are queried in the database and satisfying several end user functionalities, such as faceted browsing, service presentation and comparison, etc.).
- **Analytics Module** orchestrates the retrieval of necessary metadata and statistics from the Data layer, as well as the processing and computations for producing aggregate analytics on the catalogue's content.
- **User Feedback Module** is responsible for collecting, storing, and aggregating all user feedback (e.g. views, rates, favourites) and exposing it to the platform in the form of user statistics.
- **Recommendation Module** is responsible for producing service recommendations for RI users, based either on similar services or similar users.
- **Notification Module** manages notification generation and propagation within the platform, including alerts and recommendations to users, validation errors to SP users, updates on newly added services, etc.
- **Provider Communication Module** handles the communication with individual, external providers' contact points, and mainly includes the contact via mail and the integration and updating of (provider managed) service booking calendars on the CatRIS catalogue (see Section 3.1.2 for more detail).

Administration/Management Functionality:

- **Service Registration Module** manages all processes related to the registration of new services and the update of existing services.
- **Service Monitoring Module** manages the update of service metadata via the synchronization with remote service providers' systems.
- **User Registration/Authentication Module** handles the registration of RI users via AAI.
- **Provider Registration Module** handles the registration of provider users that are entitled with registering and managing services in the CatRIS catalogue.
- **User profiling Module** offers to authenticated users of the platform personalized features, such as favourites, ratings, recommendations, personalized views, etc.

- **Resource Management Module** handles the management (addition, modification, deletion) of all entities in the CatRIS catalogue (e.g. services, providers, options), as well as of all classifications and vocabularies used in the descriptions of the entities.
- **Validation Module** is responsible for the validation of input services (and additional entities if required), against their conformance with the respective Entity Types.

The aforementioned modules are designed to be implemented in a modular manner, and their communication is performed via well-defined APIs implemented through the Interoperability layer (Section 2.4). This loosely coupled scheme ensures a high degree of independence and adjustability with respect to changes in the functionality of specific modules or changes in the handled entities in the catalogue.

Next, the modules of the Application layer are discussed in detail.

2.2.1 Search

The Search module supports full keyword search over the descriptions of services stored in the CatRIS catalogue. The Search module receives as input free text and, optionally, a set of service attributes on which the user wishes to focus her keyword search. If the set of attributes is not provided, then the keyword search is performed on all indexed attributes of the service. The Search module exploits the Elasticsearch APIs in order to perform the required pre-processing on the input text (tokenization, special character/stopword removal, stemming) and to produce a matching score for each service-result. Then, it fetches the results and makes them available via a REST API to the platform's UI, in the form of a list of service overviews.

Several configurations are available for optimizing the list of results and making the discoverability of services easy, intuitive and fast, such as the definition of weights on specific attributes for being matched on a different priority with the input keywords or the definition of the ranking order of the results based on a specific attribute (e.g., date of service creation, the inclusion/exclusion of attributes from the search, etc.). These configurations are implemented in the form of parameters passed, in a seamless way, from the UI to the Search module.

The Search module builds on top of the Index and Query Manager functionality of the underlying Data layer, providing a REST API to the UI and offering advanced keyword search functionality.

2.2.2 Metadata Retrieval

The Metadata Retrieval module handles all the functionality related to querying the CatRIS database on metadata of services or any other entity types. This functionality varies from supporting faceted search on several attributes of the services, to retrieving and presenting certain attribute subsets for one or a set of services, to presenting a list of services related to a specific service provider, etc.

Regarding the facets generation, the Metadata Retrieval module allows the definition of facets on top of specific attributes of the service catalogue, such as the service categories and subcategories, scientific domains, the service providers, etc. Facets classify each service along multiple explicit dimensions, enabling the services to be accessed and ordered in multiple ways rather than in a single, pre-determined, taxonomic order. In addition, for each value in a faceted list, the number of services pertaining to this value is calculated and presented to the user. Faceted filtering allows the user to quickly view all available values for an attribute and filter out the services that corresponds to a value. Regarding service presentation, the Metadata Retrieval module supports the formulation of configurable queries to the database for selecting subsets of entities' attributes. These queries are exposed to the UI layer as configurable API functions.

The Metadata Retrieval module builds on top of the Index and Query Manager functionality of the underlying Data layer, providing a REST API to the UI and offering service comparison and service browsing functionality.

2.2.3 Analytics

The Analytics module is responsible for producing and managing statistics of the CatRIS platform, regarding both (a) statistics calculated on the distributions of the handled entities and their metadata and (b) statistics computed on the usage of the CatRIS platform by all types of users.

The former includes statistics calculated on the attributes of service or provider entities, e.g., the distribution of scientific disciplines, of services per category or country, the number of SPs per discipline, the distribution of access type of services generally, or per category. In a fully populated catalogue of services and providers, for most of these statistics, pre-computed results may be periodically updated in the CatRIS database and directly retrieved when requested by a user, with significant performance gains. However, since such queries can combine a large number of dimensions (entity attributes), the number of potential query formulations grows prohibitively large to allow exhaustive pre-computation and storage. Thus, this functionality is supported for a limited set of such query formulations, which might comprise the most frequently required queries for the overlying end user functionality. For the remaining query formulations, the Analytics module handles the on-the-fly query execution and result retrieval.

The latter case, indicatively, includes statistics on service views, favourite services, user visits to the service provider or the service homepage, statistics on user rates and comments on services, etc. The values of most of these types of statistics constantly change, thus maintaining pre-computed statistics on them is not an efficient option. On the other hand, several of these statistics are directly stored in the database, as KPIs on services and providers, and are constantly updated, triggered by changes in the database content caused by user interactions.

The Analytics module communicates with several components of the Data layer, i.e., the Statistics Manager, the Index Manager, and the Query Manager and exposes its functionality to the UI layer via REST APIs tailored to the needs of the end user functionality of the platform.

2.2.4 User Feedback

The User Feedback module is responsible for gathering and processing user feedback regarding the service catalogue by the platform's UI and providing aggregated statistics. Users of the platform, either public or authenticated, provide implicitly or explicitly different feedback regarding a service offering such as rating of a service, service views, visit to the service or the respective service provider homepage. User actions are collected by the UI via the Matomo software and processed to the User Feedback module, which associates each action with the user, the context in which this action is performed and the specific service.

2.2.5 Recommendation

The Recommendation module is responsible for recommending new/unseen services to authenticated users, implementing two well-known approaches: (a) item-based recommendations and (b) collaborative filtering recommendations. The former method takes into account the user's interaction history, including the services she has viewed, visited reaching the service or the offering such as rating of a service, service views, visit to the service or the respective service homepage, included in favourites, rated, etc. Based on the service metadata of these historical interactions, it identifies similar services, emphasizing on some important factors, such as scientific discipline or countries offered, and recommends them to the end user.

The latter method identifies similar users to the end user, taking into account both profile similarity and similarity in the services both users have interacted with. Then, it recommends to the end user services that similar users have positively interacted with (e.g., included in favourites), again emphasizing on important factors-attributes of the recommended services.

2.2.6 Notification

The Notification module is responsible for notification generation and propagation in the CatRIS platform between the different modules and layers. Its main functionality is responding to user actions in the UI of the platform (e.g., a user registers a new service, a user adds rating to a service, etc.) and generating appropriate user notifications. It produces alerts, recommendations and updates to the user, including also validation messages and success or failure messages. It communicates with the Notification Manager of the underlying Data layer, receives events on the data and transforms them to user notifications.

2.2.7 Provider Communication

This module manages the communication with an external service provider, considering two specific functionalities of the CatRIS catalogue: service inquiry and service booking. Regarding the former, the module is responsible for retrieving available contact information for a set of selected services, and automatically forwarding a user-drafted inquiry to all the respective contact points. Regarding the latter, the module is responsible for the management of a service calendar per service, including its publishing and updating by the service provider responsible for the service, and the recording of user interactions (bookings) on the calendar.

2.2.8 Service Registration

The Service Registration module is responsible for processing the creation of a new service in the CatRIS platform, as well as for managing any subsequent updates of the service metadata by the SPs. Service Registration supports the creation, update and removal of a service and is accessible by the users representing service providers (SP users – see Section 3.3), through the registration dashboards of the UI. It takes as input a service description in the form of a JSON object, validates the compliance to the service schema stored in the respective Entity Type object (required fields, correct data types, etc.), enriches the service with platform-generated metadata (e.g., unique IDs) and forwards the content to the underlying Data layer for storage and indexing. The module communicates also with the Service Monitoring module, for retrieving service descriptions from a remote service provider system through an API, in case a user wishes to register a new service via a URL, i.e., the service description is stored in the service provider's system.

Finally, the Service Registration module is responsible for recording all actions related to the creation, update and removal of a service in the platform. It tracks the history of all actions applied on a service (e.g., updates on the metadata) and presents them to the users of the service providers, via the service registration dashboard of the UI layer.

2.2.9 Service Monitoring

The Service Monitoring module is responsible for managing the monitoring and update of service metadata, such as the KPIs for a service. The module allows SP users to define a set of KPIs and associate them with their services. It operates both in a manual update manner, by allowing users to explicitly update this information through the UI (via a web form or uploading a JSON file), as well as in an automatic manner, through which users provide the remote source (in the form of an API) where service updates are published as well as the update frequency and the service collects and integrates this information in the CatRIS platform.

Similarly to Service Registration, Service Monitoring records all actions related to the update of service metadata in the platform and makes them available to the users of the service providers, through the service monitoring dashboard of the UI layer.

2.2.10 User Registration/Authentication

The User Registration module is responsible for the registration/authentication of users of the CatRIS platform. CatRIS has adopted, for its user management services, AAI (Authentication and Authorisation Infrastructure) services. An AAI service enables an organisation to manage access to digital resources among its users, assigning authentication and authorisation to external identity providers.

The module allows the authentication of RI users in order to offer advanced and personalized facilities to them. A user authenticates herself using an external identity provider for remote authentication. Specifically, a user is able to access and authenticate herself using the credentials she has been provided by her Home Organization or other identity providers and the AAI mechanism of the module will process and verify the login information provided by the external identity provider. For that, AAI will connect to the identity provider and receive (the first time a user delegates access) an identifier that uniquely identifies the user. This identifier is persistent across different sessions by the same user, within the scope of the identity provider.

The AAI mechanism will require users to fill in a set of minimum required attributes, such as name, email, etc., comprising their profile. The AAI mechanism will initially attempt to retrieve these attributes from the user's Home Organization or identity provider. If this is not possible, then the user must manually fill in any missing information. Additional profile information may also be filled in by the user (e.g., scientific domain, country, service categories of interest), to facilitate the personalization of functionality offered by the CatRIS catalogue.

CatRIS employs the AAI services implemented for the EOSC Portal which already integrate various AAI services and solutions, such as the EGI CheckIn, EUDAT B2ACCESS, GEANT EduTeams, INDIGO IAM and EduGAIN. It also offers connection with other identity providers such as Google and ORCID.

2.2.11 Provider Registration

The second category of users consists in providers that want to register and manage their services through the CatRIS catalogue. The user registration process in this scenario is performed as follows. First, the SP needs to have been registered as an authenticated user, following the process described above. Then, the user needs to login as a user and successfully register a service related to the organization the user is affiliated with. Upon validation of the registered service, the user is authenticated and authorized as a SP user, obtaining access to all services registered by her affiliated service providing organization.

2.2.12 User Profiling

The User Profiling module is responsible for managing all user related data in the platform, processing information about the users' profiles, the various roles and privileges, the organization they belong to, etc. The main interaction is with the Profile management module of the UI, which pushes and requests information about a user upon registration, and login. Moreover, it processes and enriches the profile of a user with personalized information such as the favourite services a user adds in her profile, services created by a user, etc., which can be later used as quick reference by the user to the content of the catalogue (e.g., view list of favourite services). The user profiling module communicates with the underlying Data layer for storing and retrieving data.

2.2.13 Resource Management

This module handles the management of entities that cannot be registered and managed by RI users or SP users. Such entities include the schemas of the Service/Resource and Provider entities, the Option, Funder and Service Level Target entities, as well as all vocabularies and classifications used by the catalogue entities. The module allows the platform's administrator to add, update and delete such content, following potential evolution of the schema or the vocabularies that characterize the handled entities.

2.2.14 Validation

The Validation module is responsible for assessing the validity of the entities stored in the CatRIS catalogue. This consists in verifying: (a) the validity of the entity schemas imported in the platform and (b) the validity of the imported entities against the respective schemas.

As previously discussed, in order to represent the schema for a specific entity, an Entity Type object is created and populated with a JSON schema document that prescribes the entity's schema. The Validation module is responsible for validating the schema of the imported/update JSON object of the Entity Type, w.r.t. to its conformance to JSON schema. As long as a valid Entity Type exists in the catalogue, the database can be populated with entities of the specific type. Each entity is then validated using the respective JSON schema, e.g., by checking the existence of all required attributes or the alignment with the data structures prescribed in the JSON schema for the entity's attributes. Further, since JSON schema does not support the definition of more elaborate/strict validation rules in the data values, the Validation module implements and applies additional validation rules on the imported entities. This includes, for example, checking whether the "Webpage" attribute of a service is actually a valid URL, or the "Last Update" attribute is an actual date. In case of any errors, the module produces validation messages for the UI.

2.3 User Interface (UI) Layer

The UI layer exposes the CatRIS catalogue functionality to the end users of the platform. Four types of users, with different roles and, thus, with access on different facilities of the platform are recognized: non-registered users, registered RI users, SP users and administrators (see Section 3.3 for a more detailed discussion). Based on the role of each user, different groups of UI modules are accessible. For example, all users have access to the modules related to the browsing, search, comparison and visualization of the service catalogue; SP users have access to the service monitoring and service registration UI modules; platform administrators can perform administrative tasks through the administration panel and the User Management module.

The UI layer comprises several modules which can be categorized in three groups, based on the intended functionality: *Service Catalogue Facilities*, *Service and User Management* and *Platform Administration*. Since the end user functionality is also presented via the functional specifications of Section 3, next, we briefly discuss the functionality of these modules.

2.3.1 Service Catalogue Facilities

This group of UI modules handles all end user functionality related to searching, browsing, comparing, and viewing statistics on services. Also, it includes the interaction between users and the catalogue, via user feedback provision from the users and notification forwarding to the users. In particular, the following modules are included:

- **Service Search.** This UI module supports the service search functionality, which consists in keyword search and faceted search. The UI combines both functionalities in the same page, since the two search modes can be jointly applied by a user. Within each facet, corresponding to a service attribute that can be used for filtering service

results, only a few values are initially visible, while the facet can be expanded by the user, so that all available values can be viewed for selection. The keyword search area lies on the top of the page and comprises a single search box. Upon expansion, the user is able to select specific service attributes on which the keyword search is performed. The user can iteratively change the keyword query or her choice of facet values, invoking a new search and being presented with a new set of retrieved service summaries-results.

- **Service Browsing.** The module allows users to view classifications of services, view services organized in categories, view details on the attributes of a service and, through them, navigate to the remote service provider and directly access services of interest. For the registered users, the module allows personalized views of service descriptions according to the different profiles of the users.
- **Service Visualization.** This module employs tables, charts, maps, as well as cubic visualizations for presenting to the end users aggregate statistics and analytics on the catalogue content and usage. Depending on the user role, different visualizations are available. General statistics, such as distributions of services based on several of their attributes, as well as number of service views are available to all users. Personal catalogue usage statistics are limited per registered user, while statistics revolving around service providers (e.g. total views or additions to favourites of services per service provider), or ratings of services are limited to the respective SP users.
- **Service Comparison.** This module supports the comparison of a set of user-selected services with respect to a set of user-selected attributes. The comparison is performed via a table where the user can view, side by side, the different values the compared services take per attribute.
- **User Notifications.** This module spans over several interfaces of the UI layer, supporting the generation of several types of notifications towards the end user. These include recommendations of relevant, new services, updates on services or categories, validation errors, messages issued by the platform administrator or the SP users, etc.
- **User Feedback.** Similarly, this module spans over several interfaces, including all the offered means for feedback provision by the end user. These include ratings and comments on services and adding services or categories to favourites. However, user actions like navigating to a SP homepage or performing an inquiry on a service are also considered types of user feedback and are captured/logged by the platform.
- **Service Inquiry.** This UI module allows users to select a set of services and forward a small inquiry, in the form of free text, to the respective contact points of the services.
- **Service Booking.** This module supports the publishing and updating, by the SP users, service availability calendars on the CatRIS catalogue, as well as the booking of service time slots via the calendar, by registered RI users.

2.3.2 User and Service Management

The Service and User Management UI modules offer functionality to the RI and SP users concerning the management of their profiles and services. In particular, the following modules are included:

- **User/provider Registration.** This module supports the registration of a user as RI user or as SP user. The latter allows users to register themselves as delegates of a service provider, so that they are authorized to manage services from this organization.
- **Service Registration.** This module supports the registration of a new service in the platform in various ways, by either filling in the service details in a web form, uploading a JSON file with the service description or denoting a remote API endpoint, where the service description can be retrieved.

- **Profile Management.** This module allows users to create their profile (during registration), as well as update or delete it. This functionality includes selecting scientific domain(s) and service categories of interest, subscribing to certain alerts and updates mechanisms and creating lists of favourite services.
- **Service Management.** This module supports the management of services by the respective SP users. This functionality includes viewing the list of services corresponding to their organization and editing/updating details or making active/inactive certain of them; defining new KPIs and associating them with services; configuring the monitoring mechanism (frequency of updates, endpoint of each service, etc.) for updating of service metadata and KPIs; viewing and monitoring usage statistics regarding their service offering in the platform; and subscribing to notifications related to user feedback and events on their services in the platform.

2.3.3 Platform Administration

The Platform Administration UI modules enable administrators of the CatRIS platform to manage users, content and functionality of the platform. In particular, the following modules are included:

- **Administration Dashboard.** This module allows the administrator to configure various parameters and content within the platform, such as the management of enumerated fields (categories, types of services, list with languages and countries, etc.), the construction of FAQs and help pages, etc.
- **Access Management.** This module allows the administrator to grant access to registered users, depending on their role, to different sets of functionality and content of the platform.
- **User Feedback Moderation.** This module supports the moderation of user comments and ratings to services. Additionally, it monitors and moderates the user access to Service Inquiry and Service Booking facilities, to prevent excessive usage.

2.4 Interoperability Layer

The Interoperability layer enables the exchange of data from the CatRIS platform with external systems. It will implement APIs for the automatic population and synchronisation of services from individual RIs as well as APIs to export data to third-party stakeholders. Specifically, the APIs will offer the functionality: a) for an RI service provider to connect to the gateway and programmatically populate the catalogue with new services or update existing services, b) to ingest/exchange data with the eInfraCentral Portal⁶ and the Catalogue of Services of the European Open Science Cloud⁷, c) to ingest/exchange data with the MERIL Portal⁸, d) to expose service information from the CatRIS gateway to external third-party applications.

Access to the service catalogue and service metadata will be provided through dedicated data APIs in the CatRIS gateway. The service catalogue itself and each service resource are described by a clear set of metadata, defined by the SDT and instantiated in JSON Schema. Furthermore, each service will be identified by a persistent ID (PID), which is generated by the portal based on unique characteristics of the service, such as the service provider, the service internal ID and the service version. An indicative list of API calls, which will be supported through the platform, is presented below, noting that the list will be enriched/adapted through the course of the project:

⁶ <https://www.einfracentral.eu/>

⁷ catalogue.eosc-portal.eu

⁸ portal.meril.eu/meril

- Retrieve Information about a service, such as:
 - the most current version of a specific service, providing the service ID;
 - all versions of a service, providing the service ID;
 - a past version of a specific service, providing the service ID and a version identifier;
 - a list of services based on a set of IDs;
 - a list of all services in the catalogue;
 - a list of services offered by a SP;
 - all services in the catalogue organised by an attribute, e.g. by categories;
 - a list of services based on a set of filters.
- Retrieve information about a SP such as:
 - SP's data providing the provider ID;
 - a list of all SPs in the catalogue;
- Retrieve information about a Funder such as:
 - funder's data providing the funder ID;
 - a list of all funders in the catalogue;
 - funder's statistics providing the funder ID;
- Retrieve information about Indicators and CatRIS gateway usage statistics such as:
 - all indicators associated with a service;
 - indicator measurements collected for a service;
 - service orders per day for a service;
 - visits per day for a service;
 - favourites per day for a service;
 - average ratings per day for a service;
 - aggregate visits per day for all services offered by a SP;
 - aggregate service orders per day for all services offered by a SP;
 - aggregate favourites per day for all services offered by a SP;
 - percentage of visits for all services offered by a SP;
- Retrieve auxiliary information about list of values (i.e. vocabularies) used in CatRIS, such as:
 - all categories/subcategories used in CatRIS;
 - the values of enumerated fields in the service description.

The APIs will expose metadata related to the service catalogue via HTTP GET or POST methods, providing data in JSON format. Further, the Interoperability layer will provide a harvesting module for retrieving service metadata and KPIs by service providers' APIs, thus enabling the automatic update of the service catalogue. Service offerings, which are exposed in OAI-PMH by registered service providers, will be harvested in regular time intervals by the platform. The intervals will be customized in the platform for each service provider, through the service management dashboard.

3 Functional specifications

This section presents the functional specifications, which describe the core functionality of the CatRIS platform. The majority of the functional specifications are mapped to one (or more) corresponding User Requirement (UR), as presented in D4.4, formally prescribing the functionality that covers the respective UR. Apart from the corresponding UR, each specification is represented by a unique ID, its name, its purpose, its input and output and its operation. Table 1 presents the template we use for describing the functional specifications. Following the categorization of the URs, and adding an additional category regarding the administration functionality of the platform, the functional specifications are grouped into six (6) categories: *Catalogue Browsing and Searching (CBS)*, *End-User Interaction (EUI)*, *Service/Resource Provider Interaction (SPI)*, *Visualization and Analytics (VA)*, *Catalogue Interoperability (CI)* and *Administration Functionality (AF)*.

Table 1: Template for the description of a functional specification

Field	Short Description
ID	Unique Identifier of the functional specification. Every ID starts with the prefix "FS", followed by the type of specification and an increasing specification index, for the specific type.
Name	Name of the specification
Purpose	Short description of the end-user functionality that is covered by the specification
Input	The input that is provided to the function(s) of the specification
Output	The output that is produced by the function(s) of the specification
Operation	The operations that are performed by the function of the specification
User Requirement	The ID of the corresponding User Requirement from D4.4

Next, we first present the set of functional specifications, following the template of the above table. Then, we briefly discuss the prioritization of the functional requirements, according to the initial feedback from the performed and ongoing requirements elicitation processes of the CatRIS project. Finally, we present the different types of users of the CatRIS gateway and map each functional specification to the user types that have access to the corresponding functionality.

3.1 CatRIS Gateway specifications

3.1.1 Catalogue Browsing and Searching

The following tables provide the functional specifications for the browsing and searching functionality of the catalogue. This set of specifications focuses on the supported functionality for searching, retrieving and comparing information on services.

Table 2: FS-CBS-01 - View classification information of services/resources

Field	Short Description
ID	FS-CBS-01
Name	View classification information of services/resources
Purpose	The user will be able to view information about service classifications (e.g. categories, subcategories, scientific domain)
Input	Selected classification attributes
Output	Classification hierarchies for the selected attributes
Operation	Retrieve from the repository the values for each of the input classification attribute.
User Requirement	UR-CBS-01: View classifications of services/resources

Table 3: FS-CBS-02 - Search services/resources using keywords

Field	Short Description
ID	FS-CBS-02
Name	Search services/resources using keywords
Purpose	The user will be able to search for services using keywords
Input	List of keywords; (Optional: service attributes to be searched)
Output	Ranked list of service IDs accompanied with a small set of service-representing attribute values per service
Operation	<p>Perform a keyword query and retrieve from the repository a ranked set of services, whose descriptions/attribute values have been matched to the keywords.</p> <p>The searching/ranking can be configured so as: (a) matching a keyword to specific attributes of the service provides higher score; (b) the keyword search focuses on specific attributes of the service.</p>
User Requirement	UR-CBS-02: Search the catalogue with keywords

Table 4: FS-CBS-03 - View and search services/resources using filters

Field	Short Description
ID	FS-CBS-03
Name	View and search services/resources using filters
Purpose	The user will be able to perform faceted search for services by filtering the values of several attributes of the service description
Input	Set of key-value pairs (filters) of the form: <attribute_name, value>; (Optional: List of service IDs)
Output	List of filtered service IDs accompanied with a small set of service-representing attribute values per service

Operation	Perform a query on the repository that retrieves the services that satisfy both following criteria: (a) their IDs belong to the input service IDs and (b) Their attributes match the input criteria. If a list of service IDs is not given as input, then only criterion (b) is applied to all services of the repository.
User Requirement	UR-CBS-03: Search the catalogue with filters UR-CBS-04: View list of services/resources for specific classifications/filters

Table 5: FS-CBS-04 - View detailed characteristics of a service/resource

Field	Short Description
ID	FS-CBS-04
Name	View detailed characteristics of a service//resource
Purpose	The user will be able to select a service and view all its attributes
Input	Service ID
Output	A service object containing all attributes and additional material (images and/or videos) about a service
Operation	Query the repository with the specific service ID. Retrieve all attributes for the result service
User Requirement	UR-CBS-05: View detailed characteristics of a service/resource

Table 6: FS-CBS-05 - Navigate to the service/resource at the remote service provider

Field	Short Description
ID	FS-CBS-05
Name	Navigate to the service/resource at the remote service provider
Purpose	The user will be able to select (click) a service URL and navigate to the service access point at the remote service provider's site
Input	Service ID
Output	Updated usage statistics for the service
Operation	Retrieve the service provider URL from the repository. Redirect the user to the specific URL. Update the service usage statistics with the performed action
User Requirement	UR-CBS-06: Navigate/redirect to the service/resource provider site for additional information/functionality

Table 7: FS-CBS-06 - Compare a list of services/resources w.r.t. a set of attributes

Field	Short Description
ID	FS-CBS-06

Name	Compare a list of services/resources w.r.t. a set of attributes
Purpose	The user will be able to select a list of services and compare their attributes, through a comparison table
Input	List of service IDs; (Optional: list of attributes for comparison)
Output	A table with the selected services and attribute values
Operation	Retrieve, for the input service IDs, their values for the selected attributes for comparison. Render the information in a comparison table. If a set of attributes is not provided as input, base the comparison on a predefined set of attributes.
User Requirement	UR-CBS-07: Compare services/resources characteristics

Table 8: FS-CBS-07 - Sort service/resource search results

Field	Short Description
ID	FS-CBS-07
Name	Sort service/resource search results
Purpose	The user will be able to sort a list of result services based on their values for a selected attribute
Input	List of service IDs; Selected attribute for sorting
Output	A sorted list of services
Operation	Retrieve, for the input service IDs, their values for the selected attribute for sorting. Sort the services based on their values for the specific attribute.
User Requirement	UR-CBS-10: View sorted search results

Table 9: FS-CBS-08 - View number of service/resource requests

Field	Short Description
ID	FS-CBS-08
Name	View number of service/resource requests
Purpose	The user will be able view the total number of service requests or the number of service requests per scientific domain, for a service
Input	Service ID
Output	Statistics about the service
Operation	Retrieve, for the input service ID, statistics about their total requests or number of requests per scientific domain
User Requirement	UR-CBS-11: View number of total requests UR-CBS-12: View number of scientific-specific requests

3.1.2 End-User Interaction

The following tables provide the functional specifications for the end-user interaction functionality of the catalogue. Such functionality includes user registration, profiling, providing feedback on the services, receiving recommendations, etc.

Table 10: FS-EUI-01 - Register as an authenticated user

Field	Short Description
ID	FS-EUI-01
Name	Register as an authenticated user
Purpose	The user will be able to register in the platform by allowing authentication via a remote identity provider
Input	A link to a remote identity provider for requesting user authentication and profile information
Output	A user ID
Operation	Request from remote identity provider the authentication of the user and access to information of the user profile (name, email, etc.). Validate whether the provided information covers the required fields for the user profile and if not, authenticate the user and redirect her to fill in any missing information. Create the user and assign a unique ID
User Requirement	UR-EUI-01: Register in the platform as an authenticated user

Table 11: FS-EUI-02 - Login/Logout as authenticated user

Field	Short Description
ID	FS-EUI-02
Name	Login/Logout as authenticated user
Purpose	The user will be able to login and logout from the platform using her authentication credentials
Input	User authentication credentials
Output	A user session
Operation	Authenticate the user via the remote identity provider. Enable user access to the platform
User Requirement	UR-EUI-02: Login/Logout to/from the platform as an authenticated user

Table 12: FS-EUI-03 - Edit user profile

Field	Short Description
ID	FS-EUI-03
Name	Edit user profile
Purpose	The user will be able to edit her profile in the platform

Input	A user ID; A set of updated profile information of the form <profile_attribute, value>
Output	A success message or a validation error
Operation	Validate the updated values. Store updated profile information in the repository
User Requirement	UR-EUI-03: Edit user profile

Table 13: FS-EUI-04 - Delete user profile

Field	Short Description
ID	FS-EUI-04
Name	Delete user profile
Purpose	The user will be able to delete her profile from the platform
Input	A user ID
Output	A success message
Operation	Delete user record and metadata from the repository
User Requirement	UR-EUI-04: Delete user profile

Table 14: FS-EUI-05 - Add/remove favourite services

Field	Short Description
ID	FS-EUI-05
Name	Add/remove favourite services
Purpose	The user will be able to add/remove a service to/from the list of her favourite services
Input	User ID; Service ID
Output	Updated list of favourite services of the user
Operation	Add: Check if service ID already exists in the user's favourites list. If not, add the service Remove: Check if service ID exists in the user's favourites list. If so, remove the service
User Requirement	UR-EUI-05: Add/remove favourite services/resources

Table 15: FS-EUI-06 - Add/remove favourite category

Field	Short Description
ID	FS-EUI-06
Name	Add/remove favourite category
Purpose	The user will be able to add/remove a supercategory/category/subcategory to/from her respective favourites list
Input	User ID; List of categories

Output	Updated list of favourite categories of the user
Operation	Add: Check if category already exists in the user's favourites list. If not, add the category Remove: Check if category exists in the user's favourites list. If so, remove the category
User Requirement	UR-EUI-06: Add/remove favourite categories

Table 16: FS-EUI-07 - Subscribe/Unsubscribe to catalogue updates and alerts

Field	Short Description
ID	FS-EUI-07
Name	Subscribe/Unsubscribe to catalogue updates and alerts
Purpose	The user will be able to subscribe to and be notified for custom updates/alerts regarding the services of the platform, such addition of a new service, update of a service, etc
Input	A user ID; An update/alert ID
Output	Updated list of updates/alerts the user is subscribed to
Operation	Add: Check if update/alert ID already exists in the user's subscription list. If not, add the update/alert Remove: Check if update/alert ID exists in the user's subscription list. If so, remove the update/alert
User Requirement	UR-EUI-07: Subscribe/Unsubscribe to catalogue updates and platform alerts

Table 17: FS-EUI-08 - Get recommendations for new services based on similar users

Field	Short Description
ID	FS-EUI-08
Name	Get recommendations for new services based on similar users
Purpose	The user will periodically receive recommendations for new services based on services that have been viewed/requested/selected as favourites by similar users
Input	A user ID
Output	A list of recommended services with all their attributes
Operation	Retrieve user profile attributes. Execute collaborative filtering recommendation algorithm. Retrieve list of result services
User Requirement	UR-EUI-08: Get recommendations for new services/ re-sources based on similar users

Table 18: FS-EUI-09 - Get recommendations for new services/resources based on similar services

Field	Short Description
ID	FS-EUI-09

Name	Get recommendations for new services/resources based on similar services
Purpose	The user will periodically receive recommendations for new services based on previously viewed/requested/searched services by her
Input	A user ID
Output	A list of recommended services with all their attributes
Operation	Retrieve user profile attributes and user interaction log. Execute item-based recommendation algorithm. Retrieve list of result services
User Requirement	UR-EUI-09: Get recommendations for new services/resources based on services/resources a user already searched or ordered

Table 19: FS-EUI-10 - View list of services/resources based on user profile

Field	Short Description
ID	FS-EUI-10
Name	View list of services/resources based on user profile
Purpose	The user will be able to see a filtered set of services, based on specific preferences declared in her profile (e.g. scientific domain, categories, country, etc.)
Input	A user ID
Output	A list of services with all their attributes
Operation	Retrieve user profile attributes. Filter services based on user profile attributes. Retrieve list of result services
User Requirement	UR-EUI-12: View list of services/resources based on user profile

Table 20: FS-EUI-11 - Allow different views of services/resources for different user groups

Field	Short Description
ID	FS-EUI-11
Name	Allow different views of services/resources for different user groups
Purpose	Different attributes of a service will be shown on top/highlighted, depending on the user role (RI user, SP user, Administrator)
Input	A user ID
Output	Personalized view of service search and presentation pages
Operation	Select a personalized view of the search or presentation page, based on user role
User Requirement	UR-EUI-13: Allow for different views of services/resources based on the user group

Table 21: FS-EUI-12 - Query service providers for details on selected services/resources

Field	Short Description
ID	FS-EUI-12
Name	Query service providers for details on selected services/resources
Purpose	The user will be able to select one or a few services, pose a short inquiry on the services and directly send it to the contact points of the respective service providers
Input	A user ID; An inquiry text; A list of service IDs
Output	An email sent to the responsible service providers
Operation	Retrieve user name and email. Retrieve, for each service ID, the contact information of the responsible service provider. Forward user inquiry, via email, to all retrieved contact points
User Requirement	UR-EUI-14: Perform inquiry on Service providers

Table 22: FS-EUI-13 - Book a service for a specific timeslot

Field	Short Description
ID	FS-EUI-13
Name	Book a service for a specific timeslot ⁹
Purpose	The user will be able to reserve a timeslot for the use of a service on a calendar that keeps up-to-date information on the service's availability
Input	A user ID; A service ID
Output	A confirmation or rejection of the booking request, via mail and/or alert
Operation	Forward a request containing user information, service ID, timeslot information. Receive and present to the user confirmation/rejection
User Requirement	UR-EUI-15: Book a service

3.1.3 Service/Resource Provider Interaction

The following tables provide the functional specifications for the service provider interaction functionality of the catalogue. Such functionality focuses on facilitating service providers in registering and managing their services.

⁹ We consider that this specification is planned for next releases of the CatRIS catalogue and under the condition that service providers are ready/willing to have such functionality handled through CatRIS.

Table 23: FS-SPI-01 - Register as provider

Field	Short Description
ID	FS-SPI-01
Name	Register as a provider
Purpose	The user will be able to register in the platform as a provider, being responsible for managing the services offered by her organization in the catalogue
Input	A user ID; A user profile, including affiliation and role
Output	A provider user ID
Operation	Request user login as user. Request user profile information and affiliation. Validate the information provided by the user. Request the registration of a service (FS-SPI-03) and validate the imported service. Upon success, provide authorization via the platform administrator or an existing service. Send a confirmation email for activating the account in the platform.
User Requirement	UR-SPI-01: Register in the platform as a Provider

Table 24: FS-SPI-02 - Login/Logout as provider

Field	Short Description
ID	FS-SPI-02
Name	Login/Logout as a provider
Purpose	The user will be able to login and logout from the platform, as provider, using her authentication credentials
Input	User authentication credentials
Output	A user session
Operation	Authenticate the user. Enable user access to the platform
User Requirement	UR-SPI-02: Login/Logout to/from the platform as a Provider

Table 25: FS-SPI-03 - Register a service/resource

Field	Short Description
ID	FS-SPI-03
Name	Register a service/resource
Purpose	The user will be able to register a new service in the platform, by filling in the service information fields, required and optional. The user may alternatively provide the location of the JSON file that describes the new service.
Input	A service description
Output	New service ID assigned to the service or a validation error
Operation	Validate the input service description. If valid, store the service in the platform's repository. Record the details-metadata of the

	action (e.g. the user who added the service and the date). If not valid, return validation error
User Requirement	UR-SPI-03: Register a service/resource

Table 26: FS-SPI-04 - Update a service/resource

Field	Short Description
ID	FS-SPI-04
Name	Update a service/resource
Purpose	The user will be able to update the metadata of a service directly via the catalogue UI. Alternatively, the user may provide the location of the JSON file that describes the updated description of the service
Input	A service ID; A service description
Output	An updated service description and success message or a validation error
Operation	Validate the input service description. If valid, update the service in the platform's repository. Record the details-metadata of the action (e.g. the user who added the service, the date). If not valid, return validation error
User Requirement	UR-SPI-04: Update a service/resource

Table 27: FS-SPI-05 - Delete a service/resource

Field	Short Description
ID	FS-SPI-05
Name	Delete a service/resource
Purpose	The user will be able to delete a service from the platform. This function is provided only for wrongly created services. Services that should not be accessible anymore should be flagged as inactive (FS-SPI-06).
Input	A service ID
Output	A confirmation or rejection of the deletion request
Operation	Validate whether the service can be deleted and delete it from the repository. Produce confirmation or rejection message
User Requirement	UR-SPI-05: Delete a service/resource

Table 28: FS-SPI-06 - Publish or unpublish a service/resource

Field	Short Description
ID	FS-SPI-06
Name	Publish or unpublish a service/resource
Purpose	The user will be able to flag a service as inactive or active rendering the service inaccessible or accessible for the users of

	the catalogue. Deactivated services still remain visible to the user that is the provider of the service
Input	A service ID; The selected action
Output	A confirmation or rejection of the request
Operation	Validate on whether the service can be activated/deactivated. Activate/deactivate the service.
User Requirement	UR-SPI-06: Publish/Unpublish a service/resource

Table 29: FS-SPI-07 - View list of services/resources associated with the provider

Field	Short Description
ID	FS-SPI-07
Name	View list of services/resources associated with the provider
Purpose	The user will be able to view the list of services offered by her service provider
Input	User id
Output	A list of service IDs, along with their attributes and metadata on specific actions on them (e.g. registration and last update information)
Operation	Retrieve service provider ID associated with the user ID. Retrieve a list of IDs of services provided by the service provider. Retrieve attributes and metadata of the retrieved services IDs
User Requirement	UR-SPI-07: View list of services/resources associated with the SP

Table 30: FS-SPI-08 - View log of actions applied to a service/resource

Field	Short Description
ID	FS-SPI-08
Name	View log of actions applied to a service/resource
Purpose	The user will be able to view details about the registration or the update of a service (e.g., creator, date of creation, date of last update, etc)
Input	Service ID; User ID
Output	Requested metadata for the service
Operation	Validate whether the service is accessible by the user. If yes, retrieve requested metadata of the service. If not, return a rejection message
User Requirement	UR-SPI-08: View log of actions applied to a service/resource

Table 31: FS-SPI-09 - View statistics and visualizations on a service/resource

Field	Short Description
ID	FS-SPI-09
Name	View statistics and visualizations on a service/resource
Purpose	The user will be able to view aggregated usage statistics for a service (number of service views, number of visits to the service homepage, number of times the service was added to favourites, etc.)
Input	Service ID; User ID
Output	Requested metadata-statistics for the service
Operation	Validate whether the service is accessible by the user. If yes, calculate/retrieve requested metadata of the service. If not, return a rejection message
User Requirement	UR-SPI-09: View usage statistics/analytics and visualizations for services/resources

Table 32: FS-SPI-10 - Publish and update service availability calendars

Field	Short Description
ID	FS-SPI-10
Name	Publish and update service availability calendars
Purpose	The user will be able to publish an availability calendar for a specific service on the CatRIS catalogue and update it based on user requests
Input	Service ID; User ID; Calendar information
Output	Published/updated calendar
Operation	Validate whether the service is accessible by the user. If yes, publish or update the calendar of the service. If not, return a rejection message
User Requirement	UR-SPI-11: Publish/update service availability calendars

3.1.4 Visualisation and Analytics

The following tables provide the functional specifications for the visualization and analytics functionality of the catalogue. Such functionality focuses on the visualization facilities supported in the CatRIS catalogue for viewing aggregate statistics on service metadata.

Table 33: FS-VA-01 - Visualize service/resource distribution

Field	Short Description
ID	FS-VA-01
Name	Visualize service/resource distribution

Purpose	The user will be able to view statistics on services attributes (e.g. scientific domain, TRL, classification, location) on either tables, charts, maps or cubic visualizations
Input	Attribute ID; Selected visualization option; (Optional: List of service IDs)
Output	List of services with the values of the attribute to be visualized
Operation	Retrieve attribute values for the list of service IDs. Calculate aggregate statistics. Return the results according to the selected visualization option
User Requirement	UR- VA-01: Visualize service/resource distribution on tables UR- VA-02: Visualize distribution of services/resources on charts UR- VA-03: Visualize distribution of services/resources on maps UR- VA-06: Visualise aspects of the catalogue with cubic visualisation

Table 34: FS-VA-02 - Visualize classifications of services/resources with a colour coding

Field	Short Description
ID	FS-VA-02
Name	Visualize classifications of services/resources with a colour coding
Purpose	The user will be able view classifications of services via different colouring
Input	Classification attribute; (Optional: List of service IDs)
Output	List of services marked with different colour coding
Operation	Assign colour coding to different values of the classification attribute. Return list of services annotated with their assigned colour
User Requirement	UR- VA-05: Visualise classifications of services/resources with a colour coding

3.1.5 Catalogue Interoperability

The following tables provide the functional specifications for the interoperability. Such functionality mainly regards the APIs for programmatically inserting, updating and exposing information to/from the CatRIS catalogue.

Table 35: FS-CI-01 - Programmatically insert a new service/resource

Field	Short Description
ID	FS-CI-01
Name	Programmatically insert a new service/resource

Purpose	The user will be able to use the CatRIS API in order to insert a new service available in JSON format
Input	A service description
Output	New service ID assigned to the service or a validation error
Operation	Validate the input service description. If valid, store the service in the platform's repository. Record the details-metadata of the action (e.g. the user who added the service, the date). If not valid, return validation error
User Requirement	UR-CI-01: Programmatic insertion of a new service

Table 36: FS-CI-02 - Programmatically update a service/resource

Field	Short Description
ID	FS-CI-02
Name	Programmatically update a service/resource
Purpose	The user will be able to use the CatRIS API in order to update an existing service
Input	A service ID; A service description
Output	An updated service description and success message or a validation error
Operation	Validate the input service description. If valid, update the service in the platform's repository. Record the details-metadata of the action (e.g. the user who added the service, the date). If not valid, return validation error
User Requirement	UR-CI-02: Programmatic update of an existing service

Table 37: FS-CI-03 - Expose service/resource data in third parties

Field	Short Description
ID	FS-CI-03
Name	Expose service/resource data in third parties
Purpose	The user will be able to access, via RESTful APIs, one or more services in structured formats (JSON)
Input	A list of service IDs; A user ID
Output	An JSON output with the requested service data/metadata
Operation	Retrieve descriptions for the requested service IDs. Render the descriptions available via RESTful APIs
User Requirement	UR-CI-04: Exposing service data in third parties

3.1.6 Administrator functionality

The following tables provide the functional specifications for the administrator functionality of the CatRIS platform. Such functionality focuses mainly on approval of registration processes and moderation of catalogue content and usage.

Table 38: FS-AF-01 - Approve/reject a provider registration

Field	Short Description
ID	FS-AF-01
Name	Approve/reject a provider registration
Purpose	The user will be able to approve or reject the pending registration of a new service/resource provider
Input	A new provider description
Output	A new provider is registered in the catalogue or the application is rejected
Operation	Validate the information provided by the user. Validate the imported service (FS-AF-03). Select to approve/reject the provider registration. Notification emails are sent at each action.
User Requirement	-

Table 39: FS-AF-02 - Add/remove a provider user

Field	Short Description
ID	FS-AF-02
Name	Add/remove a provider user
Purpose	The user will be able to assign/revoke the role of a service provider user to a registered user.
Input	A user email and a provider ID
Output	The user is added \ removed from the list of users for the specific provider.
Operation	Add/remove user using her email. Produce confirmation or rejection message. Notification email is sent to all existing managers of the provider, as well as to the newly added member.
User Requirement	-

Table 40: FS-AF-03 - Approve/reject a service/resource registration

Field	Short Description
ID	FS-AF-03
Name	Approve/reject a service/resource registration
Purpose	The user will be able to approve or reject the registration of a service/resource

Input	A service description
Output	New service ID assigned to the service or a rejection message
Operation	Validate the imported service (FS-AF-02). Select to approve/reject the service
User Requirement	-

Table 41: FS-AF-04 – Moderate user comments and actions

Field	Short Description
ID	FS-AF-03
Name	Moderate user comments and actions
Purpose	The user will be able to approve publishing of user comments, as well as moderate (e.g., hide) excessive/abusive use of the catalogue's functionality
Input	A set of comments or actions per user
Output	Approval/rejection selections per comment or action
Operation	Inspect the content of the comment or statistics on the action. Select to approve/reject the comment/action
User Requirement	-

Table 42: FS-AF-05 – Manage vocabulary/classification content

Field	Short Description
ID	FS-AF-04
Name	Manage vocabulary/classification content
Purpose	The user will be able to insert/update lists of classifications and vocabularies utilized in the descriptions of the services/resources and providers
Input	A vocabulary/classification list ID; (Optional: a new vocabulary/classification term)
Output	An updated list for the given ID
Operation	Inspect the content of the list/term. Replace the existing list with new the list or update the existing list with the new term
User Requirement	-

3.2 Discussion on user requirements for further consideration

The previous subsection presented a core set of functional specifications that prescribe the major functionality to be offered by the CatRIS catalogue. The presented functional specifications correspond to and further specify the CatRIS user requirements, accumulating feedback and background from an extended set of instruments: a survey with 207 responses; 29 interviews; a validation workshop and three focus groups engaging different stakeholders (RI users, service providers, policy makers). Further, the requirements elicitation and

specifications derivation processes where facilitated by a thorough desk research (also presented in D4.4), as well as by the strong background of several of the project partners related to previous cataloguing efforts (MERIL and eInfraCentral projects).

Consequently, the prescribed functional specifications are expected to represent real stakeholder needs for all considered user groups mentioned above. Additionally, they incorporate several “lessons learned” from the previous cataloguing efforts. Based on all the aforementioned feedback, we present some major insights and directions, that are utilized in the specification design and guide the development of the catalogue:

- **Diverse levels of user expertise.** A significant insight is the fact that there exist considerable differences in the level of expertise through RI users, with respect to searching and identifying the RIs and their services. For example, a user visiting the catalogue for the first time, might wish to have an overview of the contents and browse over a small description of the services to find which one may fit her needs, whereas a returning or expert user can directly employ the proper keywords or facets to reach the desired services. On the other hand, there exist users that need to examine several details for a service, e.g. view service’s use cases, or compare two or more services along several characteristics, before they decide on whether a service fits their purposes. A service catalogue should be structured and presented in such way, so as to cover the whole range of users, in terms of expertise and experience. To this end, the catalogue’s UIs should present gradually, but in a user friendly and intuitive way, increasing levels of detail of a service or set of services, starting from the most important pieces of information (e.g. categories, access type, country) and expanding to more detail (e.g. use cases, target users).
- **Rating, commenting and recommendations.** A large part of RI stakeholders, either users or service providers, that participated in the organized focus groups consider user rating and commenting a sensitive issue, with ambiguous or even negative expected effects on a catalogue of services. From the RI users’ point of view, ratings on services might unjustifiably bias the users’ selection on services, while this selection should mainly be based on objective criteria-features of the service. From the service providers’ view, unmoderated and unmonitored user feedback might be of ambiguous quality and intentions and could negatively affect the user engagement to otherwise effective and well-provisioned services. On the other hand, recommendation of new, unseen services was found by RI users as an interesting and potentially useful feature, as long as the recommended services are in some way relevant to the user (either similar to previous services she has interacted with or related to similar users). Given the above, recommendation, rating and commenting mechanisms need to be carefully designed and incorporated in the gateway’s interfaces, so as to function in a non-intrusive but auxiliary and helpful way.
- **Personalized views.** This functionality is interrelated to the previous two bullets. Given the diversity of the catalogue’s users with respect to user expertise and disciplines, a useful feature would be the personalized presentation of information to the end users. Personalization would include selecting different pieces of information or differently prioritizing the presentation of information to the end user, in a service search, browsing, presentation or comparison scenario. Effectively incorporating such features to the gateway’s interfaces requires gradual and iterative implementation and feedback from user assessment groups.
- **Interaction between RI users and service providers.** Although the main goal of the CatRIS catalogue is facilitating the harmonization and discoverability of RI services, facilitating the communication and interaction between RI users and service providers

could be of high value, as indicated by related paradigms¹⁰. To this end, incorporating such functionality, from simply facilitating users to directly and easily communicate with service providers, to implementing a first-cut service booking functionality within the CatRIS catalogue could potentially increase end user engagement.

- **Common, controlled vocabularies.** A significant means towards achieving the harmonization of service descriptions in a highly fragmented landscape (Research Infrastructures), is promoting the (re)utilization of common vocabularies for describing services by all stakeholders. Of course, this cannot be an exclusively bottom down approach, since the vocabularies are essentially created by the stakeholders and evolve through time. Thus, it is of high significance to provide, for as many service description attributes as possible, controlled vocabularies to be used by: (a) service providers to register and describe their services and (b) for RI users to search for services. At the same time, it is equally significant to allow users to easily discover the proper vocabulary values through the catalogue's UIs, as well as to provide users with the option to add-suggest additional vocabulary terms, in case the existing ones do not satisfy their needs.
- **Integration with existing RI catalogues.** Within the goals of CatRIS lies the integration with existing catalogues and, in particular, MERIL and eInfraCentral. While the latter is a straightforward process, since the two platforms (CatRIS and eInfraCentral) share similar technologies and APIs, the integration with the former needs further consideration and communication between the two projects (MERIL and CatRIS), which has already started but is not yet finalized until the writing of this deliverable. The finalization of the discussions on the optimal strategy of integration (optimizing the re-use of MERIL content within CatRIS, minimizing the effort of service providers to provide and repopulate catalogue's data and avoiding the implementation of complex technical solutions that will be potentially unexploited and unnecessary), is expected within M9 of the CatRIS project. After that milestone, the respective functional specification will be able to be finalized and implemented within the CatRIS gateway APIs.

Given the above discussion points and based on the functional specifications described in the previous subsection, the following table presents a set of user requirements which require further consideration and will be materialized as functional specifications for the upcoming releases of the CatRIS catalogue.

Table 43: User requirements for further consideration

User Requirement	Title	Discussion
UR-CBS-08	View services/resources rating	This UR requires further consideration and feedback from the RI stakeholders and will be prescribed as functional specification for the 3 rd (final) release of the catalogue (M20)
UR-CBS-09	View services/resources comments/ rankings	This UR requires further consideration and feedback from the RI stakeholders and will be prescribed as functional specification for the 3 rd (final) release of the catalogue (M20)

¹⁰ <https://www.scienceexchange.com/>

UR-EUI-10	Add personal rating to a service/resource	This UR requires further consideration and feedback from the RI stakeholders and will be prescribed as functional specification for the 3rd (final) release of the catalogue (M20)
UR-EUI-11	Give feedback/comment on a service/ resource	This UR requires further consideration and feedback from the RI stakeholders and will be prescribed as functional specification for the 3rd (final) release of the catalogue (M20)
UR-SPI-10	Approve comments on a service/resource	This UR requires further consideration and feedback from the RI stakeholders and will be prescribed as functional specification for the 3rd (final) release of the catalogue (M20)
UR-VA-04	Visualize ratings of services/resources	This UR requires further consideration and feedback from the RI stakeholders and will be prescribed as functional specification for the 3rd (final) release of the catalogue (M20)
UR-CI-03	Data integration from existing catalogues	This UR requires the finalization of the integration plan between MERIL and CatRIS and will be prescribed as functional specification for the 2nd (intermediate) release of the catalogue (M14)

3.3 User roles and functionality access rights

CatRIS recognizes four different user roles, with each role having access to different sets of functionalities offered by the catalogue. *Unregistered users* are able to exploit the core facilities of the platform, which consist in service search, browsing and comparison, as well as a subset of aggregate statistics and visualizations on services. *Registered RI users* obtain access to advanced functionality, including personalized views of the catalogue's content, as well as the ability to provide user feedback and access notifications and recommendations to/from the platform in several forms. *Service Provider (SP) users* are responsible for registering and managing services in the platform, representing essentially the service providing organization; consequently, they are granted access to additional functionality related to their role. Finally, *Administrator users* are assigned the maximum level of access to the platform functionality, being additionally responsible for the management of content and users of the platform.

Next, a more detailed presentation for each user role is provided, followed by a mapping of the platform's functionality (functional specifications) to individual user roles.

3.3.1 Unregistered users

Unregistered (non-authenticated) users comprise a core target group of the platform as they include a number of individual stakeholders with different interests that need to have access to RI services for their work or research. These users have access to the main browsing and searching functionalities of the platform. They can browse and search the catalogue for services related to their interests and needs and they can select specific ones to get information about their characteristics and KPIs. They can also compare services with similar ones of the same category.

3.3.2 Registered RI users

User registration provides to the end users access to additional functionality, mainly profiling, personalization and rating. By filling in information on their profile and interests, they can have the possibility to access personalized contents by selecting services or service categories of interest and get alerts and notifications in relation to them, when there are updates available. They can also have personalized views of the service catalogue, which will provide focused presentation of specific parts of the service catalogue that are relevant to their interests and profile. They will also be able, if they wish, to rate the services they access.

3.3.3 Service Provider (SP) users

The service providers can access the same advanced functionality of the registered RI users and in addition they also have the possibility to register their services in the platform, manually or automatically, and to define harvesting mechanisms for updates in the services and their KPIs. They are also able to access monitoring functionalities for their services and their related KPIs.

3.3.4 Administrator users

The CatRIS platform administrators are super-users who have access to the functionality available to the registered users. Additionally, they have the authorization to manage all aspects of the platform data, including user authentication and service registration.

Table 44: Mapping of Functionality to User Roles

Functional Specification	Unregistered Users	Registered RI Users	SP Users	Administrator Users
FS-CBS-01 - View classification information of services/resources	✓	✓	✓	✓
FS-CBS-02 - Search services/resources using keywords	✓	✓	✓	✓
FS-CBS-03 - View and search services/resources using filters	✓	✓	✓	✓
FS-CBS-04 - View detailed characteristics of a service/resource	✓	✓	✓	✓
FS-CBS-05 - Navigate to the service/resource at the remote service provider	✓	✓	✓	✓
FS-CBS-06 - Compare a list of services /resources w.r.t. a set of attributes	✓	✓	✓	✓
FS-CBS-07 - Sort service/resource search results	✓	✓	✓	✓
FS-CBS-08 - View number of service/resource requests	✓	✓	✓	✓
FS-EUI-01 - Register as an authenticated user	✓	✓	✓	✓
FS-EUI-02 - Login/Logout as authenticated user		✓	✓	✓
FS-EUI-03 - Edit user profile		✓	✓	✓

FS-EUI-04 - Delete user profile		✓	✓	✓
FS-EUI-05 - Add/remove favourite services		✓	✓	✓
FS-EUI-06 - Add/remove favourite category		✓	✓	✓
FS-EUI-07 - Subscribe/Unsubscribe to catalogue updates and alerts		✓	✓	✓
FS-EUI-08 - Get recommendations for new services based on similar users		✓	✓	✓
FS-EUI-09 - Get recommendations for new services based on similar services		✓	✓	✓
FS-EUI-10 - View list of services based on user profile		✓	✓	✓
FS-EUI-11 - Allow different views of services for different user groups		✓	✓	✓
FS-EUI-12 - Query service providers for details on selected services		✓	✓	✓
FS-EUI-13 - Book a service for a specific timeslot		✓	✓	✓
FS-SPI-01 - Register as provider			✓	✓
FS-SPI-02 - Login/Logout as provider			✓	✓
FS-SPI-03 - Register a service/resource			✓	✓
FS-SPI-04 - Update a service/resource			✓	✓
FS-SPI-05 - Delete a service/resource			✓	✓
FS-SPI-06 - Publish or unpublish a service/resource			✓	✓
FS-SPI-07 - View list of services/resources associated with the provider			✓	✓
FS-SPI-08 - View log of actions applied to a service/resource			✓	✓
FS-SPI-09 - View statistics and visualizations on a service/resource	✓	✓	✓	✓
FS-SPI-10 - Publish and update service availability calendars			✓	✓
FS-VA-01 - Visualize service/resource distribution	✓	✓	✓	✓
FS-VA-02 - Visualize classifications of services/resources with a colour coding	✓	✓	✓	✓
FS-CI-01 - Programmatically insert a new service/resource			✓	✓
FS-CI-02 - Programmatically update a service/resource			✓	✓
FS-CI-03 - Expose service/resource data in third parties	✓	✓	✓	✓

FS-AF-01 - Approve/reject a provider registration				✓
FS-AF-02 - Add/remove a provider user			✓	✓
FS-AF-03 - Approve/reject a service/resource registration				✓
FS-AF-04 - Moderate user comments and actions				✓
FS-AF-05 - Manage vocabulary/classification content				✓

Conclusions

D5.1 presented the data model, the architecture and the functional specifications of the CatRIS gateway, which build on the requirement elicitation and modelling processes followed during the first eight months of the project. The deliverable presents information on the content managed in the catalogue, the functionality offered, as well as the user roles supported. D5.1 serves as the basis for the development of the CatRIS gateway, with its first release being documented in D5.2.

The content of the deliverable presented an initial instantiation of the prescribed functionality of the CatRIS gateway. Since the gateway's development will comprise an agile and iterative process, several decisions and prioritizations might be revisited, regarding both the functionality and the design of the platform. Such changes will be mainly guided by the user assessment processes that will take place in the frame of WP5 and will be documented in upcoming WP5 deliverables and, in particular, in D5.5.