Horizon 2020 Programme
Digital Excellence & Science infrastructure
eInfrastructure Science Cloud

Grant Agreement Number: **731049**
Project Acronym: **eInfraCentral**
Project Full Title: **European E-Infrastructure Services Gateway**
Type of Action: **Coordination and Support Action (CSA)**
Call: **H2020-INFRASUPP-2016-2017**

Deliverable D4.1
Portal functional and architectural specifications

eInfra Central

Type (distribution level):	**Public**
Deliverable Leader:	UOA
Authors:	George Papastefanatos, Akrivi Katifori, Natalia Manola, Stefania Martziou, Antonis Lempesis, Panagiotis Labropoulos (UoA)
Due Date:	31/08/2017
Actual Submission Date:	27/09/2017
Version:	1.0

Abstract: This deliverable specifies the functional and architectural specifications of the eInfraCentral Portal, including the platform architecture and detailed specification of each components, the list of functional specifications corresponding to the user requirements collected, as well as the design of the UI of the platform.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731049

Document Revision History

Date	Version	Author/Editor/Contributor	Summary of main changes/Status
01/07/2017	0.1	George Papastefanatos	ToC prepared
04/08/2017	0.2	George Papastefanatos, Akrivi Katifori, Natalia Manola, Stefania Martziou, Antonis Lempesis, Panagiotis Labropoulos (UoA)	Draft sent for review
18/09/2017	0.3	Diego Scardaci, Yannick Legre (EGI), Annabel Grant, Karl Meyer, Afroditi Sevasti (GEANT), Jorge Sanchez (JNP)	First batch of comments integrated
21/09/2017	0.9	George Papastefanatos	Final document for approval and review
27/09/2017	1.0	Alasdair Reid (EFIS)	Final document approved and submitted

Executive Summary

The ***eInfraCentral platform*** is the major output of the eInfraCentral project and it aims at offering end-users a single gateway to a **harmonized service catalogue** of European e-infrastructures, including key performance indicators, such as availability, quality of service and end-user satisfaction.

eInfraCentral aims at offering a **collaborative web platform to act as the main entry point to European e-Infrastructure services**, providing uniform access to the existing service catalogues offered disparately by each e-Infrastructure. For that, the eInfraCentral platform will collect (through registration and harvesting of services and their KPIs) and aggregate content about service offerings and create a uniform service catalogue through standardized guidelines and Application Programming Interfaces (APIs), to continuously offer up-to-date information.

D4.1 describes the functional specifications of the eInfraCentral platform. In brief, it provides:

- An overview of the data model designed for representing the main entities managed within the platform namely, services, users, service providers, KPIs, etc.
- The architecture design of the platform, structured in four different logical layers, namely the repository, business, user interface (UI) and interoperability layers.
- The components of each layer with details on their functionality, data exchange and dependencies as well as the technology used for their implementation.
- The functional specifications that will guide the development of the platform and their mapping to user requirements, as collected during the requirements elicitation phase of the project.
- The UI mock-ups which will be used for the development of the front-end of the platform.
- The different users of the platform and the offered functionalities.

The functional specifications aim at fulfilling the requirements of different types of users (e.g. researchers, service providers), which are described along with the requirement elicitation process. Each functional specification was compiled by a member of the WP4 team of eInfraCentral (developers and/or architects) and corresponds to a set of functionalities that cover one or more user requirements. The derived functionalities are organized in larger modules, the services of the eInfraCentral platform.

This deliverable does not intend to be the final functional specifications document, but rather a first approach to the design of the platform, defining the basic architectural concepts and technologies to be used in the first phase of the implementation of the platform. It is expected that the user assessment activities following each release of the platform will introduce new requirements and changes that will enrich the list of specifications presented in this deliverable.

The current platform release as well as any following releases will be available in beta.eInfraCentral.eu. Following the eInfraCentral public launch/release in M24, the platform will be the main entry point for the www.eInfraCentral.eu.

Table of Contents

List of Figures

List of Tables

Acronyms

Table 1: Acronyms

Abbreviation	Meaning
API	Application Programming Interface
CA	Consortium Agreement
DoA	Description of Action
DoW	Description of Work
EC	European Commission
e-IRG	e-Infrastructure Reflection Group
ERA	European Research Area
ESFRI	European Strategic Framework for Research Infrastructure
IT	Information Technology
ITSM	IT Service Management
KPI	Key Performance Indicator
LB	Lead Beneficiary
MC	Management Committee
MS	Milestone
PID	Persistent ID
PC	Project Coordinator
PM	Project Manager
SDT	Service Description Template
SLA	Service Level Agreement
SMS	Service Management System
TRL	Technology Readiness Level
WP	Work Package
WPL	Work Package Leader

Short Names of Partner Organisations

Table 2: Partner organization and their short names

Short name	Name of Organization
EFIS	European Future Innovation System Centre
JNP	JNP CO
UoA	National and Kapodistrian University of Athens
LUH	Gottfried Wilhelm Leibniz Universität Hannover
GEANT	GEANT Limited
PRACE	Partnership for Advanced Computing in Europe AISBL
EGI	Stichting EGI
UEDIN	The University of Edinburgh
CNR	Consiglio Nazionale delle Ricerche

1 Introduction

The mission of *eInfraCentral* is to ensure that, by 2020, a broader and more varied set of users (including industry) discovers and accesses the existing and developing e-infrastructure capacity. A common approach to defining and monitoring e-infrastructures services will increase their uptake and enhance understanding of where improvements can be made in delivering e-infrastructure services.

A major output of the project is to develop a single-entry point for end-users to browse a **harmonized** service catalogue, and enhance the **monitoring** of key performance indicators that focus on availability and quality of services and user satisfaction across multiple service offerings and providers.

eInfraCentral will design, develop and deploy a **collaborative web platform to act as the main entry point to European e-Infrastructure services**. This will provide access to the existing service catalogues and related KPIs. Based on the requirement elicitation, the portal will provide different views/facets of the catalogue for different user perspectives and will act as the *"matchmaker"* between e-Infrastructure services and service requests by end-users.

The eInfraCentral platform will collect (through registration of services and harvesting of their KPIs) and aggregate content about a service offering and create a uniform service catalogue through standardized guidelines and APIs (Application programming interface), to continuously offer up-to-date information.

This deliverable describes the functional specifications of the eInfraCentral platform. It provides an overview of the data model designed for representing the main entities managed within the platform namely, services, users, organization, and KPIs. Furthermore, it presents the architecture of the platform, structured in different logical layers and describes in details the main components of each layer, their functionality, their interactions and dependencies and the technology used for their implementation. Next, the functional specifications for the development of the business logic of platform are presented and mapped to the user requirements, which were collected during the requirements elicitation phase of the project. Then, it presents the UI mock-ups which will be used for the development of the front end of the platform.

The specifications that are presented in this document are the ones that have to be met and implemented in the eInfraCentral platform in order to fulfil the requirements of different types of users (e.g. researchers, service providers). These user requirements, as well as how they were collected, are described in Deliverable D3.1 eInfraCentral Service Catalogue Requirements. Each functional specification was compiled by a member of the WP4 team of eInfraCentral (developers and/or architects) and corresponds to a set of functionalities that cover one or more user requirements. The derived functionalities are organized in larger modules, the services of the eInfraCentral platform; see more information in Section 3.

The eInfraCentral project employs an agile software development approach, in which requirements and delivered functionality evolve through the collaborative effort of the development and the end user teams. It is based on adaptive planning, evolutionary development, frequent delivery of releases, and continuous improvement, and it encourages rapid and flexible response to change[1]. As

[1] "What is Agile Software Development?". Agile Alliance. 8 June 2013.

such, this deliverable does not intend to be a final and complete functional specifications document, but rather a first approach to the design of the platform, defining the basic architectural concepts and technologies to be used in the first phase of the implementation of the platform. It is expected that the user assessment activities following each release of the platform will introduce new requirements and changes that will enrich the list of specifications presented in this deliverable.

The eInfraCentral platform aims to accommodate a wide range of stakeholders, including users of e-services, from research and industry, as well as e-infrastructure providers and other stakeholder groups like policy makers and funding bodies. In brief, platform functionality will be offered to a) public users who have access to functionality related to the browsing, search, comparison and visualization of the service catalogue, and b) authenticated privileged users who have access to personalized content and functionality of the platform, c) the service providers who have access to the service monitoring and service registration modules, and d) platform administrators who perform administrative tasks through the administration panel and the user management module. These are also described in details in Section 6.

Finally, the current platform release as well as any following releases will be available in beta.eInfraCentral.eu. Following the eInfraCentral public launch/release in M24, the platform will be the main entry point for the www.eInfraCentral.eu.

The rest of the deliverable is organized as follows. Section 2 of the describes the platform's underlying data model, denoting the main entities and their interrelationships, Section 3 presents the architecture of the eInfraCentral platform describing the different layers, modules and frontend services, Section 4 presents in detail the functional specifications of the platform based on the analysis of the user requirements of D3.1, whereas Section 5 presents the design of the platform Front End, detailing the user actors and the mock-ups. Section 6 provides an overview of the different user groups of the platform, their authentication levels and offered functionalities. Finally, Section 0 concludes the deliverable.

2 eInfraCentral Data Model

The primary goal of the eInfraCentral platform is to offer users the ability to access and browse a unified catalogue of services offered by e-Infrastructures, i.e., service providers. To satisfy the required functionality, the following data model in Figure 1 has been created, which describes the main entities and resources managed within the platform along with their interrelationships.

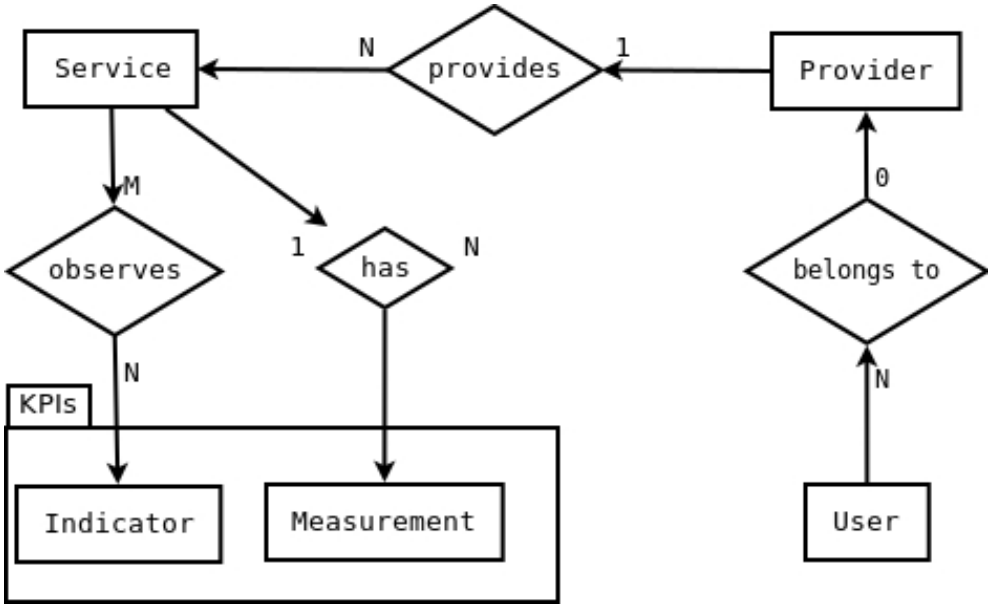


Figure 1: Data model overview of the eInfraCentral Platform

The basic entity is the Service resource. The Service resource captures the notion of a service offering and is described by the service schema as presented in D3.1 Section 1.7. It contains all attributes, mostly free text types and collections thereof. The Provider resource represents the organization offering services within the platform, and the User resource represents the users of the platform who belong to an organization. Furthermore, a service is related to one or more KPI resources; a service may observe an indicator, and in that case it has a series of measurements for it.

Apart from the core entities, there exists an auxiliary Vocabulary class, satisfying the need for relatively static enumerations of the service entity's attributes' possible values, such as a service deployment phase, its target countries, regions and languages, as well as its category and subtype.

In Annex I, the schema of the main entity of the platform, the Service resource in the form of XSD, is shown, followed by the appropriate field descriptions. The finalization of the schema is work-in-progress incorporating input from the consultations that the eInfraCentral project is performing with all the e-Infrastructure stakeholders. Following the progress of the project, the next releases of the platform will accommodate refinements and extensions to the schema as well as any additional information provided by the e-Infrastructures and other stakeholders.

3 Platform Architecture

The overall architecture of the eInfraCentral Platform is shown in Figure 3. It comprises of four main layers:

- the **Repository Layer**, responsible for all underlying content management of the platform

- the **Business Layer** that implements and offers the core functionality of the platform
- the **User Interface (UI) Layer** offers the front-end to the platform users
- the **Interoperability Layer** enables the exchange of information between the platform and external systems, such as the service providers' interfaces, or 3rd party systems that consume information regarding the platform's service catalogue.

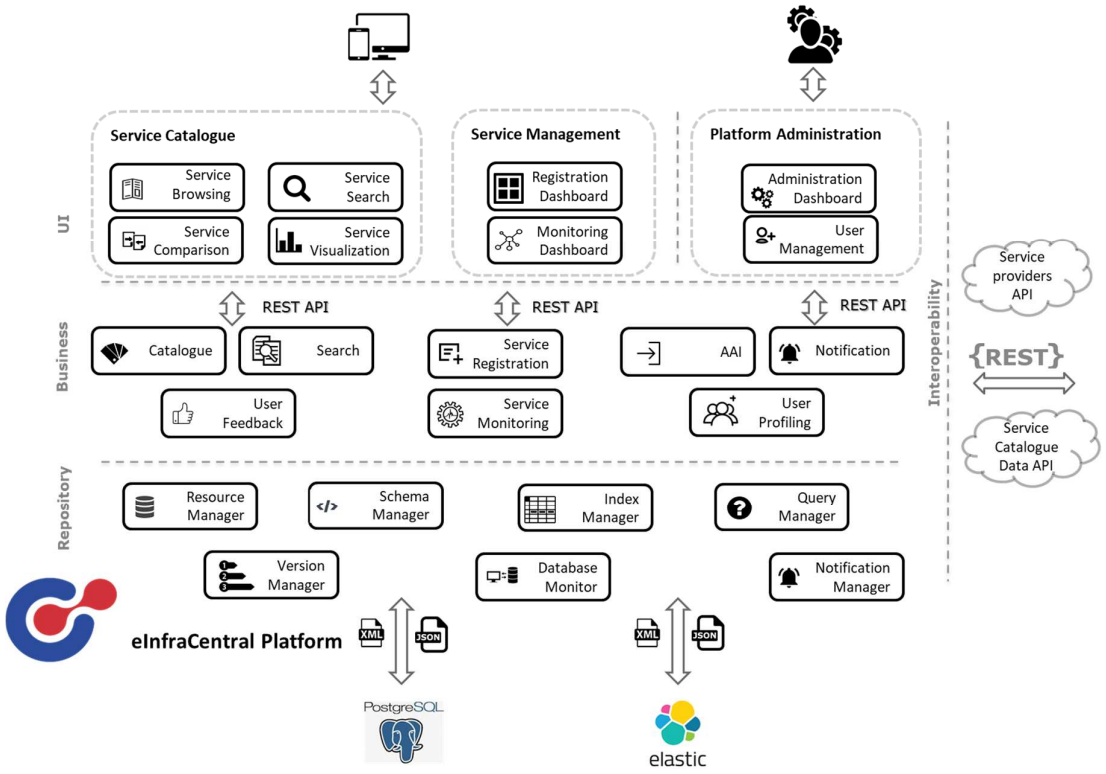


Figure 2: Architecture Overview of the eInfraCentral Platform[2]

The four layers are presented in detail in the next sections.

3.1 Repository Layer

eInfraCentral manages a large and diverse set of resources: services, KPIs, users, etc. These resources must be easily discoverable and in order to correctly use them a set of metadata must be adopted to describe them. The Repository layer contains a set of general-purpose modules responsible of storing and managing the metadata of the various resources.

The diversity of the managed resources requires that the Repository must be resource type agnostic: administrators of the Repository must be able to add/modify and delete resource types while the service is running without the need to modify the source code of the registry or the underlying database schema. Another reason for this versatility is that the eInfraCentral platform is under continuous development, with new features, services and resource types being added regularly and the Repository should be able to easily cope with these changes.

[2] Icons made by Freepik from www.flaticon.com

Metadata are traditionally described using XML but in the recent years JSON is gaining popularity among developers. The Repository is able to store metadata in either XML or JSON format providing the same functionality. A key feature of the Repository is supporting constructs from the relational databases: primary keys, unique constraints, and even referential integrity between the values of different resource types.

Finally, the Repository is able to notify interested users about changes in the contents by creating and posting events for every insertion/modification or deletion of a resource. Following, we provide a more detailed view of the architecture and the types of objects the repository manages.

3.1.1 Resource Type

In order to start storing and managing resources of a certain type (e.g. services or KPIs), an object called **Resource Type** is created and submitted to the Repository with the following information:

- **Name**: the name of the resource type, which must be unique among all the resource types, e.g., a service resource type, a user resource type, etc.
- **Payload type**: the format of the resource (XML or JSON). While the Repository supports both formats, the metadata of each resource must be expressed in one format, XML or JSON.
- **Schema (or schema URL)**: The schema of the resources (XSD for XML, JSON Schema for JSON) or the URL of the schema, if it can be found online. The schema is used when a resource is added or modified for validation and to ensure that all stored resources are valid.
- **Indexed fields (or IndexMapper)**: The underlying storage or the source code makes no assumptions on the schema and contents of the stored resources; thus, it is necessary for the Repository services extracting information from each resource type to allow the execution of keyword search queries. An example of Indexed fields are the email and name attribute of a “Person” resource, which is described by an XSD schema. While a “Person” resource may contain a lot of information about the user, the users need to search persons by their name or email. This is accomplished by specifying an IndexedFields, 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 (XPath for XML resources or JSON Path for JSON resources). When new resources are added or existing are modified, the repository uses this list in the IndexedFields to extract the values and store them in an index to allow for efficient query answering.

3.1.2 Resource

The contents of the Repository are defined in an object called **Resource**. Each Resource contains the following information:

- **Id**: a unique identifier, which is assigned by the registry to each new resource, e.g. a service Id or a user ID
- **Resource type**: the type of the resource. Before attempting to store a resource in the registry, its resource type must already be registered.
- **Payload**: the actual contents of the resource. It is either an XML or a JSON file, depending on the payload type of the respective resource type. When storing a resource, instead of the actual payload, a URL of the payload can be provided.
- **Creation and modification date**: the dates when this resource was inserted in the registry and when it was last modified.

- **Version**: whenever a resource is modified its version number is incremented and the previous versions are retained. Only the latest version of a resource is available through the search APIs and the users can access the previous versions only by using the version API.

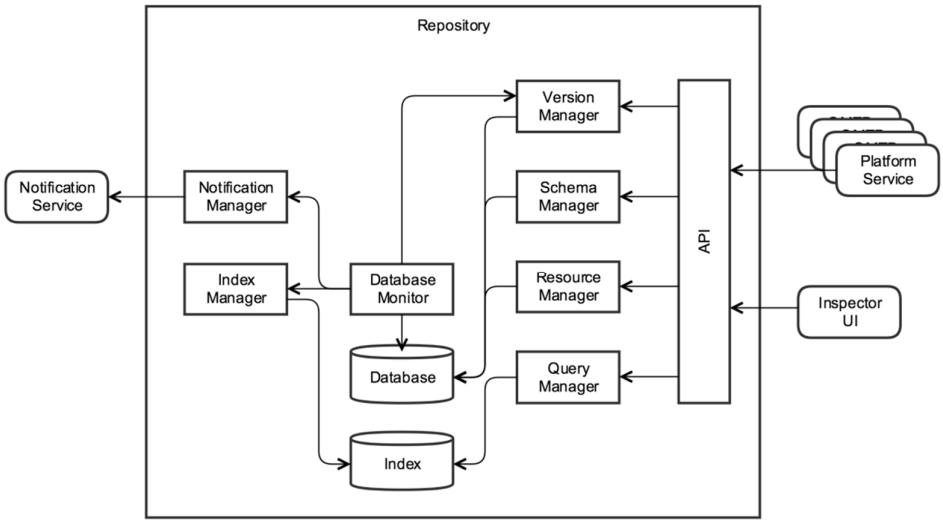


Figure 3: The Repository architecture

3.1.3 Repository components

The Repository Layer is composed from a number of different modules, each one responsible for a different task:

- **Schema Manager**: is responsible for registering resource types, validating the schema provided, and preparing the database to accept new resources of the new types.
- **Resource Manager**: manages resources: validating new resources against the schema and the constraints (referential, primary keys and unique values), and registering in the database. The resource manager is also used to update and delete existing resources.
- **Query Manager**: performs queries to the index (based on the index fields defined in the resource type) and returns the results. While the resource manager is using the database to register the resources, the queries are performed at the index, which allows for far more efficient queries.
- **Database Monitor**: monitors the database for any changes in registered resource types or resources and notifies other modules that are interested in these changes. For example, the Version Manager updates the version of resources 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 features of the repository (i.e. CRUD operations on resource types and resources) since the more computationally heavy operations (indexing, versioning, etc.) occur asynchronously.
- **Version Manager**: manages the different versions of the resources. Whenever a resource is updated, the version manager stores the previous version and assigns a new version number to the modified resource. Using the Version Manager, users can trace the evolution of a resource from the moment it was created up to its latest version.

- **Index Manager**: is responsible for updating the full text index of resources. The full text index allows the execution of queries on the resources providing results much faster than the relational database that is used to store the resource types and resources. By responding to notifications from the Database Monitor, the Index Manager keeps the underlying index synchronized with the contents of the database.
- **Notification Manager**: provides the connection between the Repository and the Notification service in the Business layer of the platform. Whenever a new resource 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 resources of type "service" respectively) and when resources are created, updated or deleted the Notification manager sends new messages in the respective topic. The notification manager allows for any other service in the platform to be asynchronously notified about the changes in the Repository without explicitly interacting with the Repository. Again, as is the case with the previous two modules, the Notification manager is notified for the changes in the resources by the database monitor.

The Repository Layer is developed using Java 8, the underlying relational database is PostgreSQL[3] and the full text index used is ElasticSearch[4].

3.2 Business Layer

The Business Layer implements the main functionality of the eInfraCentral Platform. It comprises of a set of modules that communicate with the UI Layer and the Interoperability Layer via REST APIs. These modules are the following:

- **Catalogue Module**: stores and manages all service-related resources.
- **Search Module**: offers all keyword search and faceted browsing functionality. Facets are created automatically on top of specific attributes of the service catalogue, such as the service categories and the service providers and allow for easy filtering of platform content.
- **User Feedback Module**: is responsible for collecting, storing, and aggregating all user-feedback and exposing it to the platform in the form of user statistics.
- **Service Registration Module**: manages all the registration process for 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.
- **AAI Module**: is responsible for managing user registration, logins and authentication.
- **Notification Module**: manages message propagation and notification within the platform and finally;
- **User profiling Module**: offers to authenticated users of the platform personalized features, such as favourite services.

The implementation of the modules in the Business Layer of the eInfraCentral portal aims at offering a loosely coupled and modular architecture that can be adjustable to adding new features and services and in the same time be scalable to large number of end users.

[3] www.postgresql.org

[4] www.elastic.co

All modules are implemented with the Spring Framework5 in Java1.8, exposing and exchanging data via REST APIs with the components of the other layers. The APIs are exposed also as external data services for 3rd party systems to connect, query and retrieve information from the platform.

They are deployed and executed in the Apache Tomcat Application Server as dockerised software modules. Docker6 is a software technology providing containers, an additional layer of abstraction and automation of operating-system-level virtualization on Windows and Linux.

The following subsections describe the functionality of each module in details and the interactions with the other layers of the platform.

3.2.1 Catalogue

The Catalogue manages all service-related resources of the eInfraCentral Platform offering the functionality for the retrieval of services-related resources and for catalogue viewing, browsing and service comparison. It allows users to retrieve the list of services contained in the platform's catalogue, retrieve the metadata of a specific service as well as KPIs and statistics aggregated for it or finally retrieve specific characteristics of one or more services, which will be used for comparison. The Catalogue is also used to manage auxiliary resources such as lookup fields and enumerated values for service categories, languages, etc., as well as information about persons, organizations, service providers, etc.

While the underlying Resource Manager of the repository layer handles the bulk of the metadata management operations (storing, retrieving and searching of metadata in a resource type agnostic way, the Catalogue builds on top of this functionality and it provides a REST API that is tailored to the needs of the UI and the functionality of the platform.

3.2.2 Search

The Search module offers the two main functionalities: keyword search over the catalogue of services and faceted filters generation, with which users can easily navigate to services of interest within the platform.

Regarding the keyword search functionality, the Search module takes as input free text, which is tokenized in words, and performs queries on the content service metadata of the platform. 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 applicable for optimizing the list of results and making the service 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, 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.

Regarding the facets generation, the Search module allows the generation of facets on top of specific attributes of the service catalogue, such as the service categories, the service providers, etc. Facets classify each service along multiple explicit dimensions, called facets, enabling the classifications to be accessed and ordered in multiple ways rather than in a single, pre-determined,

[5] https://projects.spring.io/spring-framework/

[6] https://www.docker.com/

taxonomic order. Facets are defined on specific attributes of the service catalogue and derived by the distinct values in these attributes. 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 service that corresponds to a value.

The Search module, builds on top of the Indexing and Query Manager functionality of the underlying Repository layer, providing a REST API to the UI and offering search, service comparison and service browsing functionality.

3.2.3 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, access to a service, navigation to the service in the remote service provider's portal, etc. User actions are collected by the UI 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. It communicates with the Resource Manager of the Repository layer to store this information.

Furthermore, user feedback calculates aggregated statistics (counts and average) about the user interaction. It enriches the metadata of a service and makes this information available to the UI in the service profile. The calculation of statistics and updates in the service profiles are performed in a synchronous way, each time a user performs an action.

3.2.4 Service Registration

The Service Registration module is responsible for processing the creation of a new service in the eInfraCentral platform as well as for the any subsequent updates in of the service metadata by the service providers. Service registration offers the creation, update and removal of a service and is accessible by the users of the service providers, through the registration dashboards of the UI. It takes as input a service description in the form of an XML, validates the compliance to the service schema (required fields, correct data types, etc.), enriches the service with platform-generated metadata (e.g., id for enumerated values, etc.) and process the content to the underlying Data layer for storage and indexing. In case of any errors, it produces validation messages for the UI.

The service registration communicates also with the service monitoring, 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 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.

Detail specifications on the service registration and processes will be described in D4.3.

3.2.5 Service Monitoring

The Service Monitoring is responsible for managing the monitoring and update of service metadata, such as the KPIs for a service, i.e., the synchronization of the eInfraCentral content with external

service providers' systems. In essence, service monitoring allows users of the service providers 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 an XML 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 eInfraCentral platform.

Similarly to the 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.

Details on the service monitoring, its processes and interactions will be described in D4.2.

3.2.6 AAI

The AAI (identity, authentication and authorization) module is responsible for the identity and access management of the platform, namely for providing the registration, authentication and authorization functionality to its users.

AAI will provide two ways of user authentication; a remote one in which users are authenticated by external identity providers and a local authentication mechanism in which users credentials are provided by the eInfraCentral platform. The latter case applies to users who do not own an account to a remote identity provider or do not wish to delegate access to the eInfraCentral portal to external identity providers.

For the remote authentication, users will be able to access and authenticate themselves using the credentials they have been provided by their Home Organizations or other identity providers and the AAI module will process and verify the login information provided by the identity providers.

For that, AAI will connect to an 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 will require users to register themselves in the platform and fill in a set of minimum required attributes, such as name, email, etc., in his\her profile. In case of remote authentication, AAI will 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.

Finally, access to the various services of the platform will be granted based on the eInfraCentral roles assigned to the user during registration.

AAI will support the OAuth 2.0 protocol and interact with the following tentative list of external identity providers:

- eduGAIN
- ORCID
- Google
- LinkedIn

Connection with other identity providers such as Social Id and Facebook will be also considered for implementation during the project.

The AAI module will interact with the login page in the UI layer, with the interoperability layer for connecting to external identity providers, as well with the underlying repository layer for storing user information.

3.2.7 Notification

The Notification module is responsible for message handling in the eInfraCentral 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 generate appropriate user notifications. It communicates with the notification manager of the underlying repository layer, receives events in the data and transforms them to user notifications. The Notification module will be implemented as a publish-subscribe service with the Apache ActiveMQ framework[7].

3.2.8 User profiling

The User profiling module is responsible for managing all user related data in the platform, processing information about the users profile, the various roles and privileges, the organization they belong to, etc. The main interaction is with the user 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 his\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.

3.3 User Interface Layer

The UI layer contains the front-end UI modules, through which users interact with the platform. Based on the role of each user, different group of UI modules are accessible, namely public and authenticated users have access to the modules related to the browsing, search, comparison and visualization of the service catalogue, users of the service providers have access to the service monitoring and service registration UI modules, whereas platform administrators can perform administrative tasks through the administration panel and the user management module. These are also described in details in Section 6.

The portal is the top layer (UI Layer) of the eInfraCentral platform. It is implemented in HTML5, CSS and JavaScript using Angular2[8], a development toolkit for building and optimizing complex JavaScript browser-based applications in typescript. Angular2 was selected because it is a modern framework, widely adopted and supported by the open source community which allows us to easily build software that will be maintained and supported for the foreseeable future. Also, it uses UIKit 2[9] for the design and development of the web interface. UIKit is a lightweight and modular front-end framework for developing fast and powerful web interfaces, offering a broad collection of HTML, CSS, and JS components, which are simple to use, easy to customize and extendable. For the visualization, Highcharts and d3js will be used for the visualization of the charts and Leaflet for the maps visualization.

[7] http://activemq.apache.org/
[8] http://angular.io
[9] https://getuikit.com/v2/

Below we briefly present these UI modules, as offered to the different stakeholder groups, focusing on their high level functionality and their interaction with the other layers; a most detailed description of their functionality and their UI design is presented in Sections 4 and 5, respectively.

3.3.1 Service Catalogue UI modules

The service catalogue UI modules offer functionality to the end users for accessing and browsing the full eInfraCentral catalogue. Namely, they allow users to:

- Access and view the catalogue of services through different views, e.g., view services organized in categories or according to the different types of targeting users.
- View details on the characteristics of a service offering
- Search on the service metadata and have an overview of the results
- Navigate and discover services in the catalogue via faceted filters
- Compare service based on one or more characteristics
- Navigate to the remote service provider and access the service
- View rich visualizations of the service catalogue and service metadata (e.g. KPIs)
- View aggregated statistics about services within the platform, such as user ratings and service usage
- View FAQs regarding the usage of eInfraCentral platform
- Register as a user in the platform

In addition, authenticated users can

- Rate a service, i.e., a user can rate a service only once.
- Subscribe to and receive notifications regarding a service or a category update
- Create collections of favourite services in their profile.

3.3.2 Service Management UI modules

The Service Management UI modules offer functionality to the users of the service providers to register and manage their offerings in the platform. Namely, they allow users to:

- Register themselves as delegates of a service provider, such that they are authorized to manage services from this organization
- Register a new service in the platform in various ways, by either filling in the service details in a web form, uploading of an XML file with the service description or denoting a remote API endpoint, where service description can be retrieved.
- View the list of services corresponding to their organization and edit/update details or make active/inactive certain services.
- Define new KPIs and associate them with services
- Configure the monitoring mechanism (frequency of updates, endpoint of each service, etc) for updating of service metadata and KPIs
- View and monitor usage statistics regarding their service offering in the platform
- Subscribe to notifications related to user feedback and events on their services in the platform.

3.3.3 Platform administration UI modules

The Platform Administration UI modules enables administrators of the eInfraCentral platform 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 as well as the management of users, organization and roles.

3.4 Interoperability Layer

The Interoperability Layer enables the exchange of data from the eInfraCentral platform with external systems. It offers the platform APIs for integrating service information from external systems of the service providers and providing the content of the platform (service catalogue and metadata) to 3rd parties.

3.4.1 Service Catalogue Data API

Access to the service catalogue and service metadata will be provided through dedicated data APIs in the eInfraCentral portal (www.einfracentral.eu). The service catalogue itself and each service entity are described by a clear set of metadata (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. A tentative list of data APIs, which will be offered through the platform, include

- Get the description of a service offering based on a service ID
- Get the KPIs of a service offering based on a service ID
- Get the full list of services in the eInfraCentral platform
- Get the list of services offered by a service provider
- Get a list of services based on a free keyword search
- Get a list of services corresponding to a category

This list will be enriched with more services in the project progress. The APIs will expose metadata related to the service catalogue via HTTP GET or POST methods. The data will be provided in XML and JSON formats. For metadata interoperability, service schema will reuse terms from widely known vocabularies and ontologies such as the Dublin Core terms and SKOS. Taxonomies and classifications for enumerated attributes of a service (e.g., service categories, service providers, languages, countries, etc.) will be modelled as SKOS concept schemes

Also, specific APIs will be provided on request to serve interoperability with custom e-infrastructures needs and requirements from relevant projects that act as service aggregators.

Finally, the interoperability layer will offer the functionality for the service catalogue to be uploaded, on a regular basis, in Zenodo, an open access repository provided within OpenAire, as well as in EU Open Data portal, thus ensuring the long-term access and reusability of the data.

3.4.2 Service metadata harvesting from service providers APIs

eInfraCentral 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 monitoring dashboard.

3.5 eInfraCentral platform licensing

The eInfraCentral platform will be powered by the UoA's and ATHENA RC's software modules already used in OpenAIRE and the OpenMinted service catalogue, distributed under the GPL/A license.

All extensions to the software will follow the same licensing pattern. As such, the eInfraCentral software will be available as Open Source and made accessible through the GitHub repository (https://github.com/eInfraCentral).

4 Platform Functional Specifications

In this chapter, we provide the functionality that will be offered by the eInfraCentral Platform in the form of functional specifications.

4.1 Platform Specifications

Each specification is described by the following information fields:

Table 3: Template for the presentation of a Specification

Specification ID	is the unique ID of the specification. All functional specifications IDs are prefixed with "FS" which stands for functional specification[10], followed by an acronym of the corresponding service (e.g. "CAT" for Catalogue) and by an incremental number (1,2, etc)
Name	the name of the specification
Purpose	the description of the functional specification in the form of operations that the user can perform
Input	the input of the function each specification performs
Operation	The operations performed by the function
Output	describes the operations to be performed within this function, including validity checks and types of processing required
Corresponds to Requirement	The correspondence of this function to the user requirements as documented in D3.1 Section3. Most functional specifications cover one or more user requirements, as listed in D3.1. In these cases, it lists the requirements this functional specification covers, using the user requirement ID and name

In the next sections, we provide the functional specifications for the main parts of the platform as recorded in the User Requirements section of the Deliverable D3.1.

For the service registration and service monitoring modules, we only provide an overview, as more detailed specifications will be presented in D4.2 and D4.3 regarding the Monitoring services specifications and eInfraCentral public launch/release, respectively as well as in MS13 and MS15 regarding the Monitoring services prototype release and Registration services prototype release, respectively.

4.1.1 Service Catalogue Browsing

In the following tables the list of functional specifications for the service catalogue browsing functionality is presented. This functionality is offered to the public users of the platform, i.e., anonymous users.

[10] "PL" prefix is used for the user requirements.

Table 4: FS/CAT/01 - List the categories/subcategories of a list of services

Specification ID	FS/CAT/01
Title	List the categories/subcategories of a list of services
Purpose	The user shall be able to view all categories and subcategories of a list of services in the eInfraCentral platform.
Input	List of services IDs
Operation	FS/CAT/01 gets the categories and subcategories corresponding to the input list of services from the eInfraCentral repository.
Output	List of all categories with the containing subcategories
Corresponds to Requirement	PL/CAT/01 - View categories of services PL/CAT/02 - View subcategories of services

Table 5: FS/CAT/02 - List the services for a specific category/subcategory

Specification ID	FS/CAT/02
Title	List the services for a specific category/subcategory
Purpose	The user shall be able to select a specific category or subcategory and view its services
Input	A category or a subcategory
Operation	FS/CAT/02 retrieves all services in the eInfraCentral repository corresponding to the input category/subcategory.
Output	List of service IDs and titles
Corresponds to Requirement	PL/CAT/03 - View a list of services for a specific category/subcategory

Table 6: FS/CAT/03 - View overviews of a list of services

Specification ID	FS/CAT/03
Title	View overviews of a list of services
Purpose	The user shall be able to select a list of services and view an overview of them
Input	List of service IDs
Operation	FS/CAT/03 retrieves from the registry the following metadata of each service in the list: id, title, description, tagline and average user rating.
Output	List of service overviews

Corresponds to Requirement	PL/CAT/04 - View an overview of services

Table 7: FS/CAT/04 - View details for a service

Specification ID	FS/CAT/04
Title	View details for a service
Purpose	The user shall be able to select a service and view its detailed characteristics
Input	A service ID
Operation	FS/CAT/04 retrieves from the registry all metadata of each service.
Output	A service object containing all metadata and additional material (images and/or videos) about a service.
Corresponds to Requirement	PL/CAT/05 - Select and view detailed characteristics of a specific service PL/CAT/07 - View additional material on the services

Table 8: FS/CAT/05 - Navigate to the service at the remote service provider

Specification ID	FS/CAT/05
Title	Navigate to the service at the remote service provider
Purpose	The user shall be able to select a service URL and navigate to the service access point at the remote service provider's site.
Input	A service URL
Operation	FS/CAT/05 redirects the user request to the remote location of the service provider for accessing and using the service. FS/CAT/05 records the user navigation for providing service usage statistics within the platform
Output	Updated usage statistics for the specified service
Corresponds to Requirement	PL/CAT/06 - Navigate to the service at the remote service provider

Table 9: FS/CAT/06 - Search for a service through keywords

Specification ID	FS/CAT/06
Title	Search for a service through keywords
Purpose	The user is able to search for specific services by providing keywords
Input	A list of keywords

Operation	FS/CAT/06 performs a keyword query and retrieves from the registry a ranked set of services. The ranking is configurable and determined based on several parameters, e.g., a higher rank is assigned to a service when more keywords are matched in a service's metadata, or when a keyword is matched in the title of a service, etc.
Output	A list of service IDs
Corresponds to Requirement	PL/CAT/08 - Search the catalogue of services through keywords

Table 10: FS/CAT/07 - View faceted filters from a list of services

Specification ID	FS/CAT/07
Title	View faceted filters from a list of services
Purpose	The user is able to view a set of faceted filters.
Input	A list of services
Operation	FS/CAT/07 retrieves from the service catalogue the distinct values of metadata which are used for producing faceted filters, such as service language, service categories and sub-categories, providers, brand name, etc. FS/CAT/07 selects those values in the filters that correspond to the input list of services.
Output	A list of faceted filters with selected values
Corresponds to Requirement	PL/CAT/09 - Faceted Search of the catalogue of services

Table 11: FS/CAT/08 - Filter a list of services through faceted filters

Specification ID	FS/CAT/08
Title	Filter a list of services through faceted filters
Purpose	The user is able to filter out a list of services through faceted filters.
Input	A list of services and a set of filter values selected by the user
Operation	FS/CAT/08 filters the list of services that match the values of each filter selected by the user.
Output	A filtered list of services
Corresponds to Requirement	PL/CAT/09 - Faceted Search of the catalogue of services

Table 12: FS/CAT/09 - Compare a list of services

Specification ID	FS/CAT/09
Title	Compare a list of services
Purpose	The user is able to select two or more services and compare their metadata.
Input	A list of services selected for comparison
Operation	FS/CAT/09 retrieves the metadata of the services selected for comparison and produces a comparison table. The user can add or remove metadata from the table or apply filters to the compared services.
Output	A table with the services selected for comparison
Corresponds to Requirement	PL/CAT/10 - Compare Services characteristics

Table 13: FS/CAT/10 - List the services for a specific user group

Specification ID	FS/CAT/10
Title	List the services for a specific user group
Purpose	The user shall be able to view a set of services that corresponds to specific user group, as defined by the administrator of the platform
Input	A user group
Operation	FS/CAT/10 retrieves all services and their predefined metadata in the eInfraCentral repository corresponding to the input user group. Mappings of services and service metadata to specific user groups are defined via the administration module of the platform.
Output	List of service overviews
Corresponds to Requirement	PL/CAT/11 - Select different views of the services according to the user group

The following list of functional specifications offers features only to registered users of the platform. Namely, users can register themselves in the platform and access features that are available only for authenticated users.

Table 14: FS/CAT/11 - Register as an end user via local authentication

Specification ID	FS/CAT/11
Title	Register as an end user via local authentication
Purpose	The user shall be able to manually register himself/herself in the platform and fill in information about his/her profile

Input	A user profile containing information regarding the name, email, organization, role, etc. of the user.
Operation	FS/CAT/11 validates the information provided by the user and upon success sends a confirmation email for activating the account in the platform. The user is then requested to enter a password to be used for being authenticated in the platform. FS/CAT/11 creates the user and assigns a unique ID.
Output	A user ID
Corresponds to Requirement	PL/UM/01 - Register in the platform as an authenticated user

Table 15: FS/CAT/12 - Register as an end user via remote authentication

Specification ID	FS/CAT/12
Title	Register as an end user via remote authentication
Purpose	The user shall be able to register himself/herself 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.
Operation	FS/CAT/12 requests from remote identity provider the authentication of the user and access to information of the user profile (name, email, etc.). It validates whether the provided information covers the required fields for the user profile and if not, it authenticates the user and redirects him\her to fill in any missing information. FS/CAT/12 creates the user and assigns a unique ID.
Output	A user ID
Corresponds to Requirement	PL/UM/01 - Register in the platform as an authenticated user

Table 16: FS/CAT/13 - Register as a service provider user

Specification ID	FS/CAT/13
Title	Register as a service provider user
Purpose	The user shall be able to register himself/herself in the platform as a service provider, who will manage service offerings of his\her affiliated organization in the platform.
Input	A user profile
Operation	FS/CAT/13 operates similarly to FS/CAT/11 or FS/CAT/12, requiring from the user to explicitly select the role of a service manager, and in that case to define his\her affiliation. The user is assigned the role of a service manager after he\she is authorized by the platform administrator or an existing service

	manager of his\her organization. FS/CAT/13 creates the user and assigns a unique ID.
Output	A user ID
Corresponds to Requirement	PL/UM/02 - Register in the platform as an eInfra (service provider) user

Table 17: FS/CAT/14 - Login of a user

Specification ID	FS/CAT/14
Title	Login of a user
Purpose	The user shall be able to login to the platform either using the local credentials or the credential by a remote identity provider.
Input	User credentials
Operation	FS/CAT/14 authenticates the user either in the local repository or via the remote identity provider and enables user access to the platform.
Output	A user session
Corresponds to Requirement	PL/UM/04 - Login in the platform

Table 18: FS/CAT/15 - Edit the profile of a user

Specification ID	FS/CAT/15
Title	Edit the profile of a user
Purpose	The user shall be able to edit his\her profile in the platform.
Input	A user ID and the updated profile information
Operation	FS/CAT/15 validates and stores updated profile information in the repository.
Output	Upon success, it returns a success message; otherwise it returns any validation errors.
Corresponds to Requirement	PL/UM/05 - Edit user profile

Table 19: FS/CAT/16 - Add/Remove a service from the list of favourites of a user

Specification ID	FS/CAT/16
Title	Add/Remove a service from the list of favourites of a user
Purpose	The user shall be able to add/remove a service from the favourite list of

	services in his\her profile.
Input	A user ID and a service ID
Operation	FS/CAT/16 adds (if not already added) or removes the service from the favourite list of services of the user.
Output	List of favourite services of the user.
Corresponds to Requirement	PL/UM/06 - Add favourites PL/UM/07 - Remove from favourites

Table 20: FS/CAT/17 - View the list of favourite services for a user

Specification ID	FS/CAT/17
Title	View the list of favourite services for a user
Purpose	The user shall be able to view the list of favourite services in his\her profile.
Input	A user ID
Operation	FS/CAT/17 retrieves the list of favourite services corresponding to the user.
Output	List of favourite services of the user.
Corresponds to Requirement	PL/UM/06 - Add favourites PL/UM/07 - Remove from favourites

Table 21: FS/CAT/18 - Subscribe/Unsubscribe to platform alerts

Specification ID	FS/CAT/18
Title	Subscribe/Unsubscribe to platform alerts.
Purpose	The user shall be able to subscribe to and be notified for custom alerts regarding the services of the platform, such addition of a new service, update of a service, etc.
Input	A user ID and an alert ID
Operation	FS/CAT/19 adds or removes the input alert from the user profile.
Output	List of alerts, which the user is subscribed to.
Corresponds to Requirement	PL/UM/08 - Subscribe to updates and alerts

Table 22: FS/CAT/19 - View the list of alerts for a user

Specification ID	FS/CAT/19
Title	View the list of alerts for a user.
Purpose	The user shall be able to view the list of alerts, which the user is subscribed to, in his\her profile.
Input	A user ID
Operation	FS/CAT/19 retrieves the list of active alerts for the user.
Output	List of alerts, which the user is subscribed to.
Corresponds to Requirement	PL/UM/08 - Subscribe to updates and alerts

Table 23: FS/CAT/20 - Rate a service

Specification ID	FS/CAT/20
Title	Rate a service
Purpose	The user shall be able to rate a service (on a five star scale).
Input	A user ID, a service ID and a rating
Operation	FS/CAT/20 stores the user rating, and updates the average rating of the service.
Output	-
Corresponds to Requirement	PL/MON/01 - Add personal rating to a service

4.1.2 Service Registration

The goal of the service registration services of the eInfraCentral Platform is to provide to authenticated users of the service providers/e-Infrastructures, the functionality to register their service offerings in the platform through a user friendly and intuitive way. Service registration will be performed via a web form as well as via the provisioning of a URL for the XML file that describes the service.

In the following tables, the basic set of the functional specifications of these services is provided. Following the implementation phase of these services, a refined and more detailed presentation will be included in D4.3.

Table 24: FS/REG/01 - Register a new service

Specification ID	FS/REG/01
Title	Register a new service
Purpose	The user shall be able to register a new service in the platform, by filling in the service information fields, required and optional. The user can also provide the

	location of the XML file that describes the new service.
Input	A service description in the form of an XML file
Operation	FS/REG/01 accepts the file with service description validates the input and stores the service in the platform's repository. It also records the details of the action, such as the user who added the service, the date, etc.
Output	Upon success, a new service ID is assigned to the new service, otherwise it returns any validation errors.
Corresponds to Requirement	PL/REG/01 - Create a new service

Table 25: FS/REG/02 - View a list of services provided by a service provider

Specification ID	FS/REG/02
Title	View a list of services provided by a service provider
Purpose	The user shall be able to view the list of services offered by his\her service provider.
Input	User id
Operation	FS/REG/02 retrieves the list of service provided by the service provider, in which the user belongs to. For each service, information about the user who initially registered and lastly updated the service is also retrieved.
Output	A list of service overviews with additional information on the user who registered and lastly updated each service.
Corresponds to Requirement	PL/REG/02 - View list of services associated with the service provider

Table 26: FS/REG/03 - Update an existing service

Specification ID	FS/REG/03
Title	Update an existing service
Purpose	The user shall be able to update the metadata of a service created by all eInfra users of his\her service provider. The user can also provide the location of the XML file that describes the updated description of a service.
Input	A service description in the form of an XML file
Operation	FS/REG/03 accepts the file with the service description, validates the input and updates the service in the platform's repository, inactivating and keeping the older version. It also records the details of the action, such as the user who updated the service, the date, etc.

Output	Upon success, it returns the service ID which was updated; otherwise it returns any validation errors.
Corresponds to Requirement	PL/REG/03 - Update a service

Table 27: FS/REG/04 - Deactivate/activate a service

Specification ID	FS/REG/04
Title	Deactivate/activate a service
Purpose	The user shall be able to deactivate or activate a service. A deactivated service is not accessible via the eInfraCentral platform.
Input	A service ID and the action for activate or deactivate
Operation	FS/REG/04 performs validations on whether the service can be deactivated (e.g., if it is already deactivated) and makes it inactive, thus inaccessible for the end users of the platform. The service is still visible and marked as inactive in the registration dashboard. FS/REG/04 also records the action
Output	Upon success, it returns the service ID which was deactivated\activated; otherwise it returns any validation errors.
Corresponds to Requirement	PL/REG/04 - Deactivate/activate a service

Table 28: FS/REG/05 - Delete a service

Specification ID	FS/REG/05
Title	Delete a service
Purpose	The user shall be able to delete a deactivated a service and remove it from the platform. This function is provided only for wrongly created services. Services that are not accessible anymore via eInfra platform should be flagged as inactive.
Input	A service ID
Operation	FS/REG/05 performs validations on whether the service can be deleted and deletes it from the repository. FS/REG/05 also records the action
Output	Upon success, it returns a success message; otherwise it returns any validation errors.
Corresponds to Requirement	PL/REG/05 – Delete a service

Table 29: FS/REG/06 - List all actions applied on a service

Specification ID	FS/REG/06
Title	List all actions applied on a service
Purpose	The user shall be able to view all actions applied on service
Input	A service ID
Operation	FS/REG/06 retrieves from the repository details (e.g., creator, date of creation, date of last update, etc.) about all actions applied on a service.
Output	A list of actions applied on the service
Corresponds to Requirement	PL/REG/06 – View all actions applied on a service

4.1.3 Service Monitoring & Platform Statistics

The goal of the service monitoring services of the eInfraCentral Platform is to provide to registered users of the service providers/e-Infrastructures, the functionality to monitor their service offerings in the platform through rich statistics and visualizations.

Service monitoring will enable the automatic harvesting and update of service information from the remote e-Infrastructures systems and the enrichment of the service catalogue. Along with statistics gathered within the platform, this information will be available to the users of the service providers for monitoring the overall usage and performance of their services within the platform.

In the following tables, a first sketch of the functional specifications of these services is provided. Following the implementation phase of these services, a refined and more detailed presentation will be included in D4.2.-Monitoring services specifications.

Table 30: FS/MON/01 - Add/remove a KPI type

Specification ID	FS/MON/01
Title	Add/remove a KPI type
Purpose	The service provider user shall be able to define a new or delete an existing type of KPI. A KPI definition comprises at least a name and a description, a set of attributes – called dimensions - that characterize the reporting of indicator (by default time dimension is selected, but other attributes, such as geographical regions can be selected) and a unit of measure that provides the measurement criteria of the numeric value (e.g., value in €, no Of Users, % service up time, etc.). Other metadata can be also defined, such as the type of visualization to be applied, the level of aggregation, etc.
Input	A KPI definition
Operation	FS/MON/01 validates the KPI definition and stores the new KPI in the repository. For deletions, FS/MON/01 validates that the KPI is not associated with a service.

Output	A KPI Id
Corresponds to Requirement	PL/MON/08 - Define a new KPI

Table 31: FS/MON/02 - Associate/Remove a KPI type from a service

Specification ID	FS/MON/02
Title	Associate/Remove a KPI type from a service
Purpose	The service provider user shall be able to associate a KPI with a specific service or remove it from the service.
Input	A KPI definition, a service ID
Operation	FS/MON/02 associates/removes the KPI with the service and stores this information in the repository.
Output	-
Corresponds to Requirement	PL/MON/09 - Associate a KPI with a service

Table 32: FS/MON/03 - View KPIs associated with a service

Specification ID	FS/MON/03
Title	View KPIs associated with a service
Purpose	The service provider user shall be able to view the list of KPIs associated with a service.
Input	A service ID
Operation	FS/MON/03 retrieves from the repository the list of KPIs types which have been associated with the service.
Output	List of KPI types
Corresponds to Requirement	PL/MON/05 - View service KPIs

Table 33: FS/MON/04 - Append/Update values of a KPI for a service

Specification ID	FS/MON/04
Title	Append/Update values of a KPI for a service
Purpose	The service provider user shall be able to insert or update the values of a KPI associated with a service. The values of the KPI can be updated manually via a

	web form, via an XML or JSON file, or through a web service.
Input	A KPI ID, a service ID, list of values
Operation	FS/MON/04 validates the input values, appends or updates the values of the KPI associated with the service and stores this information in the repository.
Output	Service object with the updated values of the KPI
Corresponds to Requirement	PL/MON/10 - Manual update of a KPI for a service through data entry PL/MON/11 - Manual update of a KPI for a service through a data file PL/MON/12 - Automatic update of a KPI for a service through a web service

Table 34: FS/MON/05 - Set the time interval for automatic updates of a KPI

Specification ID	FS/MON/05
Title	Set the time interval for automatic updates of a KPI
Purpose	The service provider user shall be able to select a KPI and set the time interval for automatic updates of a KPI associated to a service (on demand or in fixed intervals, for example once a month), when updated through a web service.
Input	A KPI ID, s service id
Operation	FS/MON/05 sets and stores the update frequency of a KPI associated with the service. This value is used by the FS/MON/04 during the automatic update of the KPI through a web service.
Output	-
Corresponds to Requirement	PL/MON/13 - Set Interval of updates

Table 35: FS/MON/06 - Visualize values of a KPI for a list of services

Specification ID	FS/MON/06
Title	Visualize values of a KPI for a list of services
Purpose	The service provider user shall be able to select a KPI and visualize its values for a list of services, which this KPI is associated with. The chart used for the visualization depends on the definition of the KPI, as defined with FS/MON/01.
Input	A KPI ID, a list of services
Operation	FS/MON/06 retrieves the KPI values and processes them (aggregation of values, filtering for a specific reporting period, etc) for the visualization.

Output	List of services with the values of the KPI to be visualized
Corresponds to Requirement	PL/MON/14 - Visualize KPIs and platform statistics for a service in dashboard

Table 36: FS/MON/07 - Get average rating for a service

Specification ID	FS/MON/07
Title	Get average rating for a service
Purpose	The user shall be able to view the average rating provided by registered users for a service.
Input	A service ID
Operation	FS/MON/07 calculates the average rating for the service
Output	A service object enriched with the average rating.
Corresponds to Requirement	PL/MON/06 - View service rating

Table 37: FS/MON/08 - Get usage statistics for a service

Specification ID	FS/MON/08
Title	Get usage statistics for a service
Purpose	The user shall be able to view aggregated usage statistics for a service, such as how many times a service detail page has been accessed, or how many times users navigate to the service provider's remote page to use the service, how many times a service is added to favourites, etc.
Input	A service ID
Operation	FS/MON/08 calculates the usage statistics for the input service
Output	A service object enriched with the usage statistics.
Corresponds to Requirement	PL/MON/07 - View usage statistics about a service

4.1.4 Platform Administration

Platform administration offers the functionality to the user administrators of the platform to configure various parameters of the UI layer as well to manage the users, roles and privileges.

The first set of functions refers to the user management functionality of the platform.

Table 38: FS/UM/01 - View the list of users

Specification ID	FS/UM/01
Title	View the list of users
Purpose	The admin user shall be able to view the list of all users in the platform
Input	-
Operation	FS/UM/01 retrieves the list of users from the repository.
Output	List of users
Corresponds to Requirement	-

Table 39: FS/UM/02 - Add/Remove a user

Specification ID	FS/UM/02
Title	Add/Remove a user
Purpose	The admin user shall be able to manually add a new user or remove an existing one from the platform
Input	User ID
Operation	FS/UM/01 creates a new user with local credentials or removes an existing one from the platform.
Output	User ID
Corresponds to Requirement	-

Table 40: FS/UM/03 - Activate/Deactivate a user

Specification ID	FS/UM/03
Title	Activate/Deactivate a user
Purpose	The admin user shall be able to manually activate or deactivate the access of a user in the platform
Input	User ID
Operation	FS/UM/01 activates or deactivates the access of a user in the platform.
Output	User ID
Corresponds to Requirement	-

Table 41: FS/UM/04 - Authorize a user as a service provider user

Specification ID	FS/UM/04
Title	Authorize a user as a service provider user
Purpose	The admin user shall be able to authorize a user registered in the platform as a service provider user and grant him\her the privilege to manage service offerings of his\her affiliated organization in the platform.
Input	A user ID
Operation	FS/UM/01 grants the role of service provider to a user.
Output	-
Corresponds to Requirement	PL/UM/03 - Authorize an eInfra user

Table 42: FS/UM/05 - Remove a user from the role of a service provider user

Specification ID	FS/UM/05
Title	Remove a user from the role of a service provider user
Purpose	The admin user shall be able to revoke the privilege to a service provider user to manage service offerings of his\her affiliated organization in the platform. The user, along with all profile information, is not deleted, but he\she becomes a simple end user.
Input	A user ID
Operation	FS/UM/02 revokes the role of service provider to a user.
Output	-
Corresponds to Requirement	-

In the following table, the function offers the ability to edit enumerated values of service metadata in the platform, such as creating/editing/removing a category, etc.

Table 43: FS/UM/06 - Edit metadata enumerated values

Specification ID	FS/UM/06
Title	Edit metadata enumerated values
Purpose	The admin user shall be able to select the attribute of the service metadata and edit the list of enumerated values.
Input	A list of values for an attribute of the service metadata

Operation	FS/UM/05 retrieves the existing list of values of the attribute, the user updates the list and FS/UM/05 stores the updated information in the repository.
Output	A list of values for an attribute of the service metadata
Corresponds to Requirement	-

4.2 Functional specifications overview and Requirements Coverage

The Functional specifications listed in the above section cover the user requirements as they have been identified and recorded in Deliverable D3.1 - Section3. To transform the user expectations and requirements into concrete specifications for the first release of the platform, several factors have been taken into account, including the user perceived requirement priority.

As a result, almost all user requirements recorded have been covered by corresponding functional specifications, except those related to the authenticated user possibility to comment on specific services, which then a user with moderator rights would be able to approve or reject (Table 44). The commentary process has not been included in the functional specifications for the first release as the considerable effort needed for moderation of the comments has been weighted against the low priority of the particular requirement.

Table 44: User requirements not covered by the functional specifications

Requirement ID	Title	Description
PL/MON/02	Add commentary on the service	The user can add comments on the service.
PL/MON/03	View commentary on the service	The user can see comments other users have made on the service.
PL/MON/04	Approve comments on a service	The administrator is responsible to approve or reject comments before they are published

Finally, specifications which do not correspond to explicit user requirements have been added, however they have been considered important for the smooth management and operation of the platform, including FS/UM/01, FS/UM/02, FS/UM/05 and FS/UM/06.

5 Platform Front-End Design

The eInfraCentral platform front-end will offer the interface for the interaction of the end-users with the platform. Its functionality is implemented in HTML5, CSS and JavaScript using Angular2 and UIKit. The UI follows an HTML responsive web design for enabling seamless browsing via web browsers and mobile devices. In the following sections, the main groups of users expected to access the platform are presented followed by the mock-ups of the platform web pages.

According to the development phases of the platform, the mock-ups have been produced based on the collected user requirements of D3.1 Section 3. Primarily they concern the web pages of the first release of the platform regarding the browsing, search and comparison of the service catalogue.

Finally, these mock-ups will be used for the initial user assessment, which in turn will provide feedback, changes and extensions for inclusion in next releases. Mock-ups for the service monitoring, service registration and platform administration will be designed and developed in next releases and presented in D4.2, D4.3, respectively.

5.1 eInfraCentral Landing Page

The eInfraCentral landing page provides the entry point to the platform, offering intuitive ways to the users to access and start browsing the service catalogue. First a free text area is available for the users to quickly start browsing the catalogue via a free keyword search. The landing page also contains the categories of services in the catalogue, such that the users can easily focus and search for services belonging to a specific category of their interest. Finally, the main page gives users the ability to directly select their role (e.g., service provider or service end user) and navigate to the pages available for each role. These are shown in the next Figures.

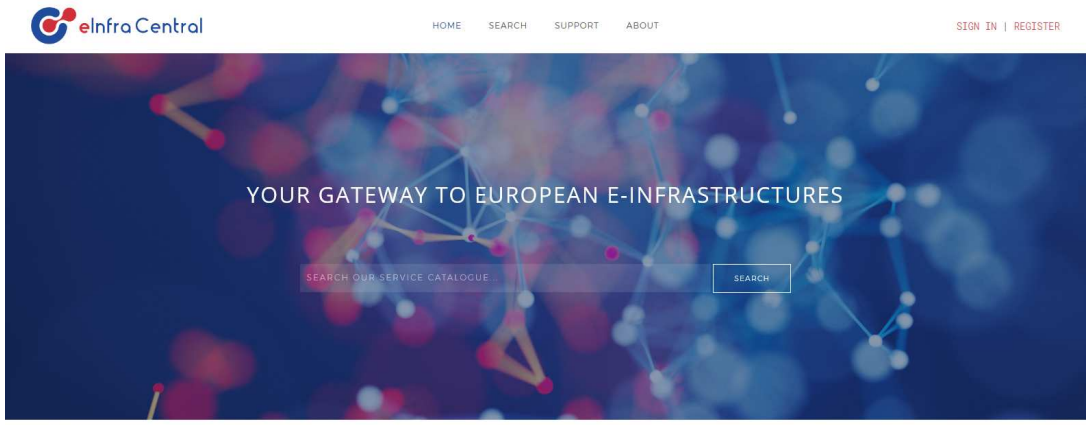


Figure 4: eInfraCentral Home Page – Quick Search for a service

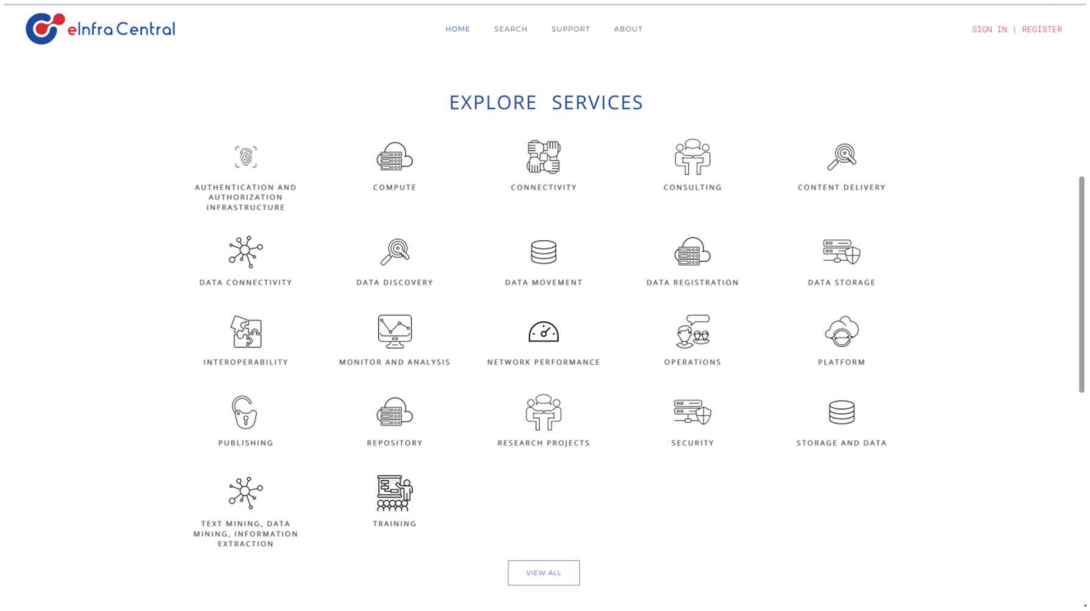


Figure 5: eInfraCentral Home Page – View Service Categories

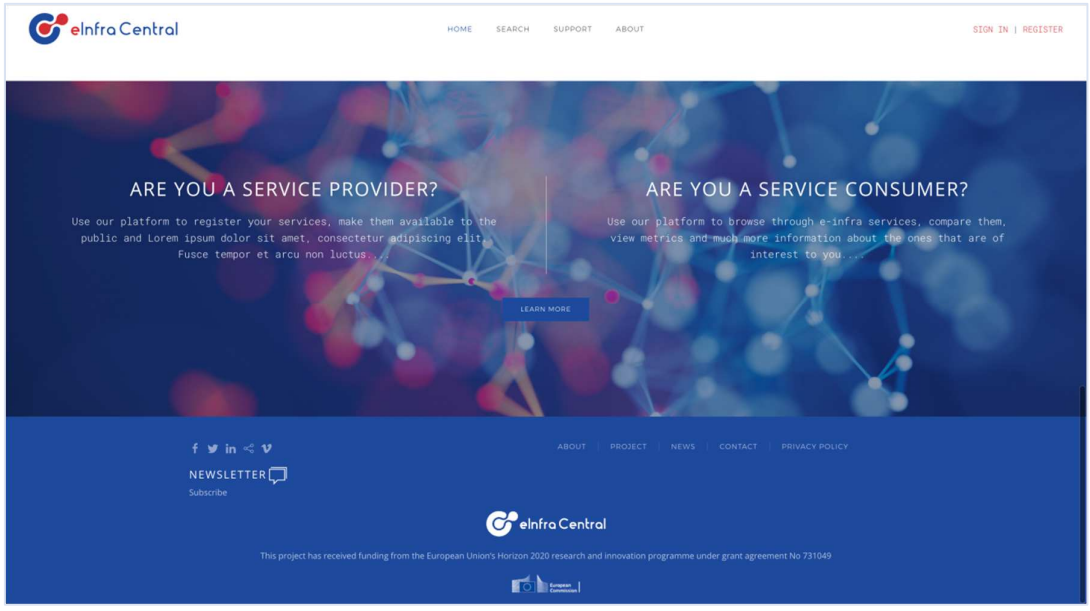


Figure 6: eInfraCentral Home Page – Choose a User Role for browsing

5.2 eInfraCentral Service Browsing Page

The eInfraCentral browsing page provides a second entry point to the catalogue, for the users to start browsing the full list of services, grouped in categories.

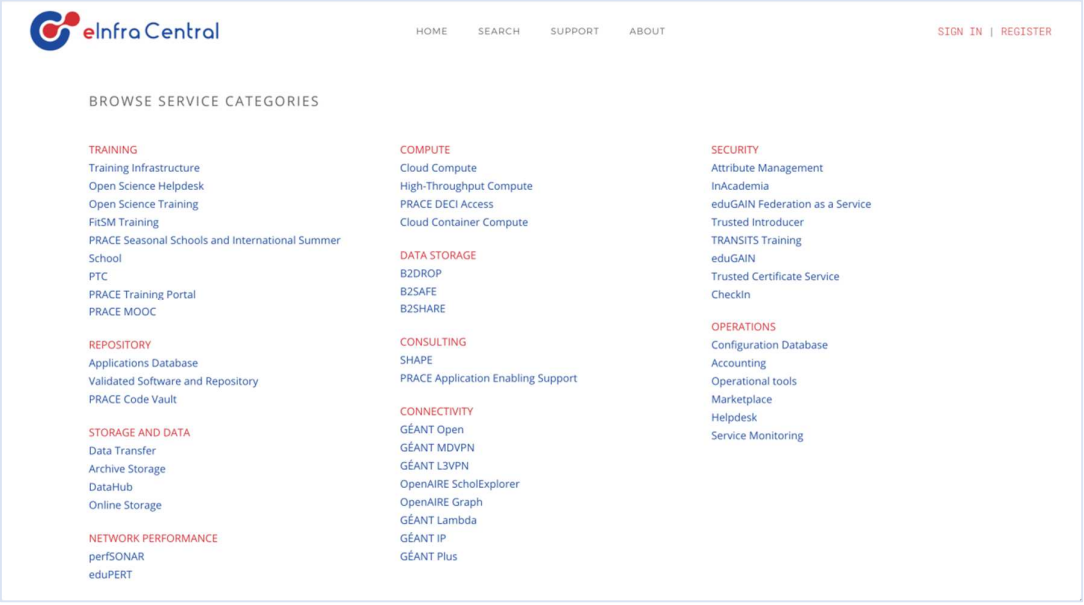


Figure 7: Service Browsing Page

5.3 eInfraCentral Catalogue Search and Faceted Filtering

The eInfraCentral search page offers the main interface for service filtering and search. It comprises a search area, where the users can enter search criteria in the form of free text. The search is performed on all indexed fields of the service resource in the underlying repository. The results are presented in the main results panel, in the form of service overviews. Finally, it offers a filter panel with facets, where the users can apply filters and limit the list of results.

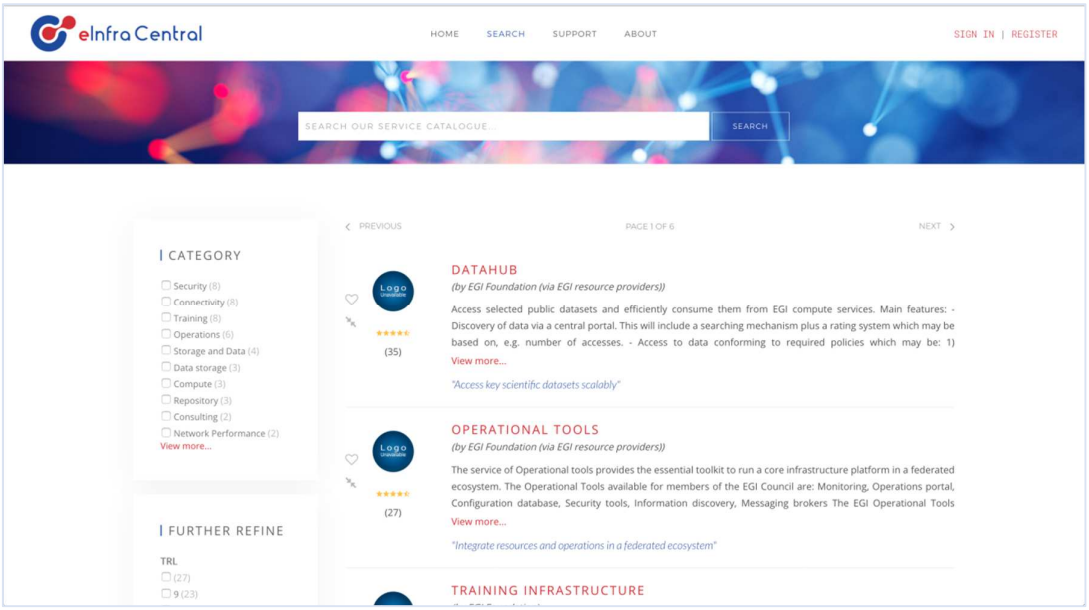


Figure 8: eInfraCentral Search Page – Overview of Results

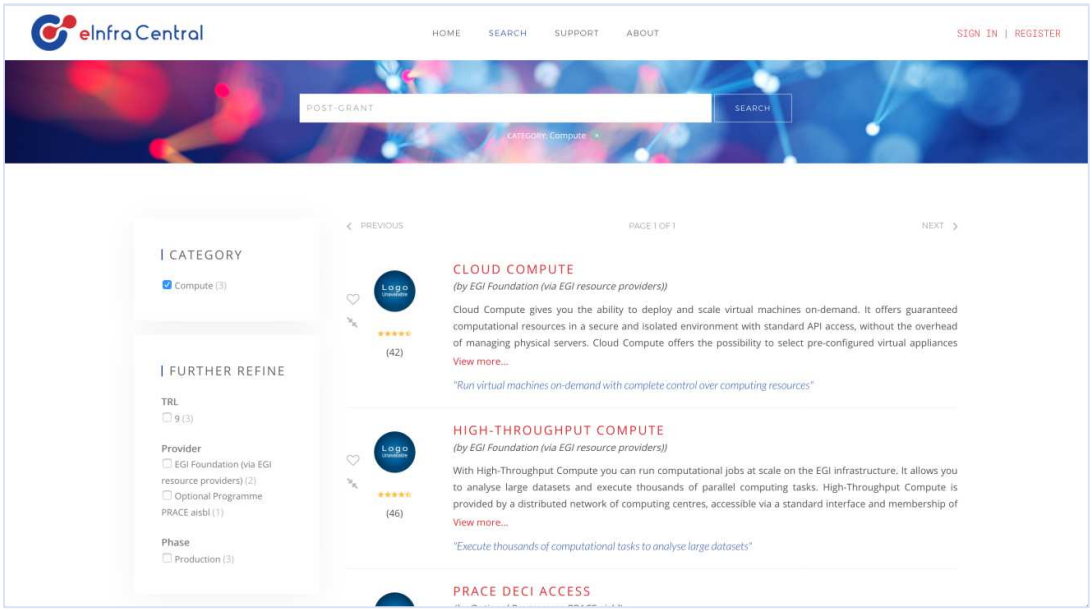


Figure 9: eInfraCentral Search Page – Filter By a Service Category

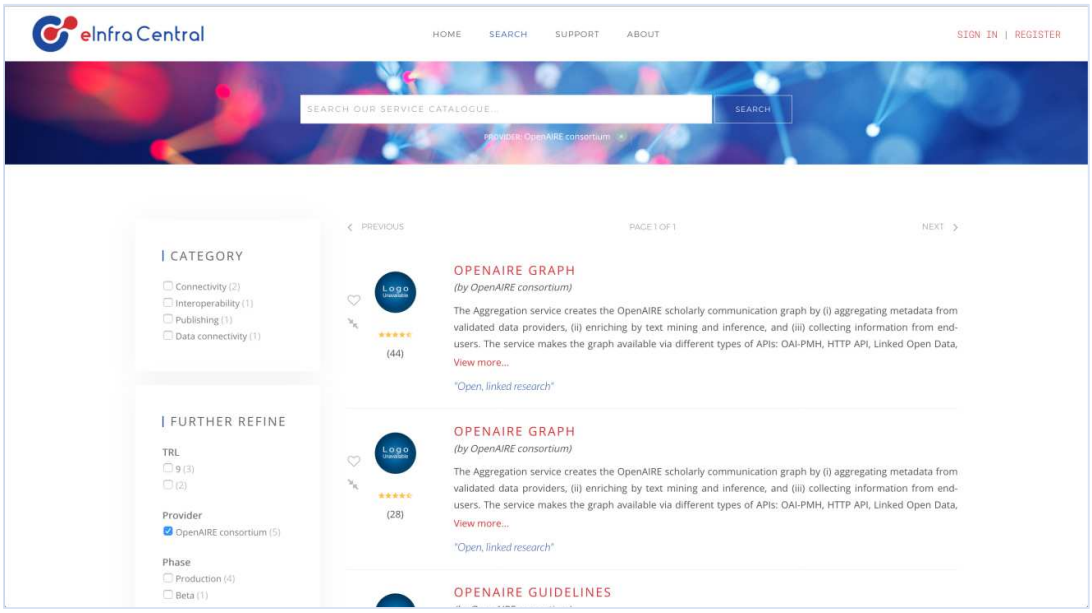


Figure 10: eInfraCentral Search Page – Filter By a Service Provider

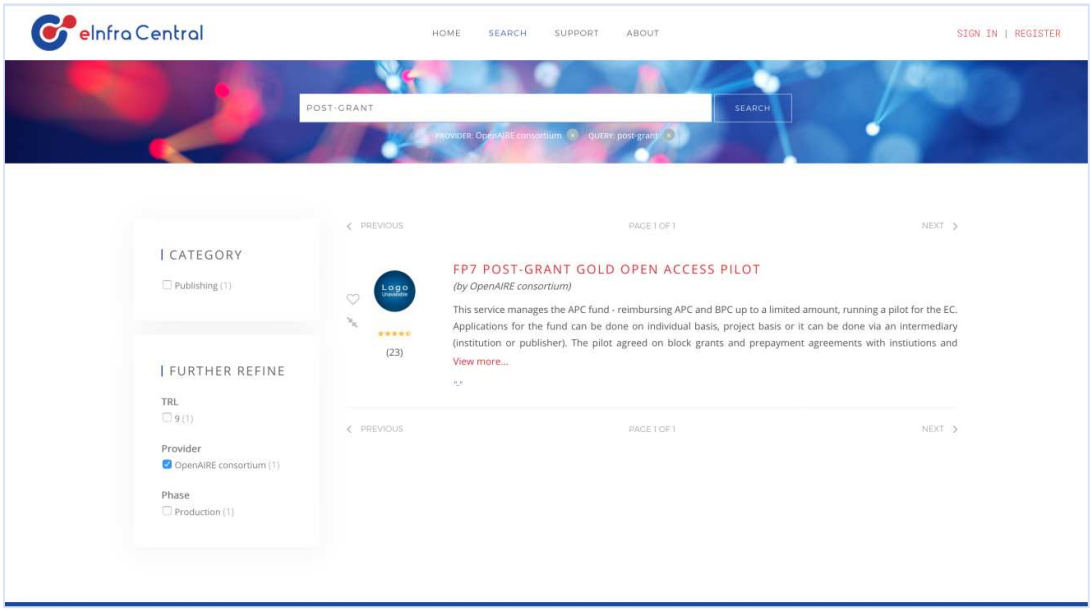


Figure 11: eInfraCentral Search Page – Filter By a Service Provider and Keywords

5.4 eInfraCentral Service Details

By selecting a service overview, the users can navigate to the details page, where all information about a service is presented. The service details page uses tags and labels for presenting the enumerated values of a service, text areas for the description and other text fields of a service, placeholders for the logo and multimedia related to the service, as well as panels for the service statistics and the links to the remote sites of the service providers. Finally it organizes further details about a service in tabs, such that the user can easily find the type of information he\she is interested in.

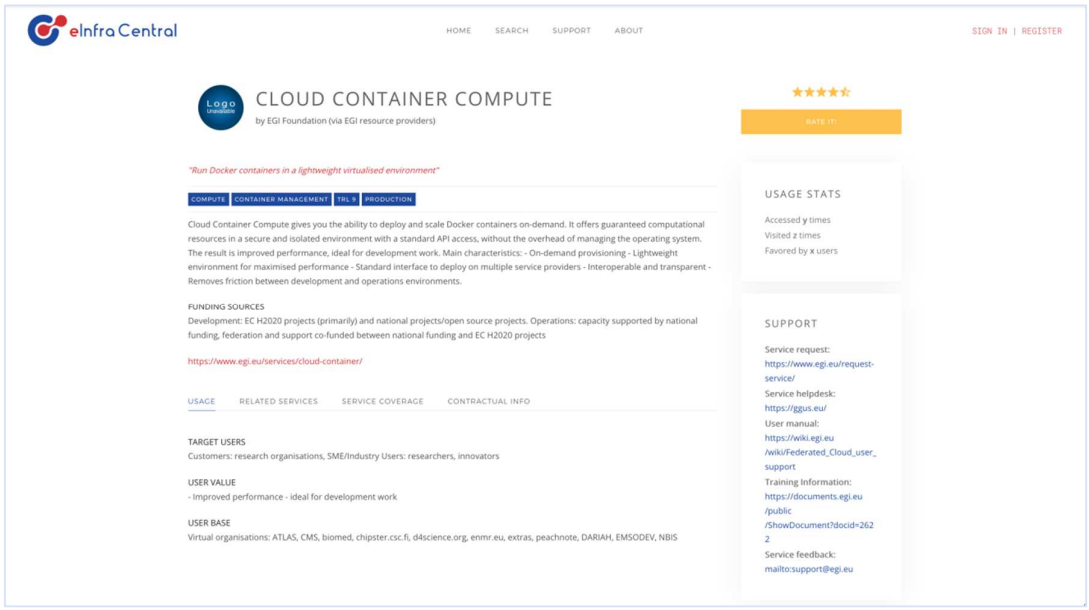


Figure 12: Service Details Page – General Information and Usage tab

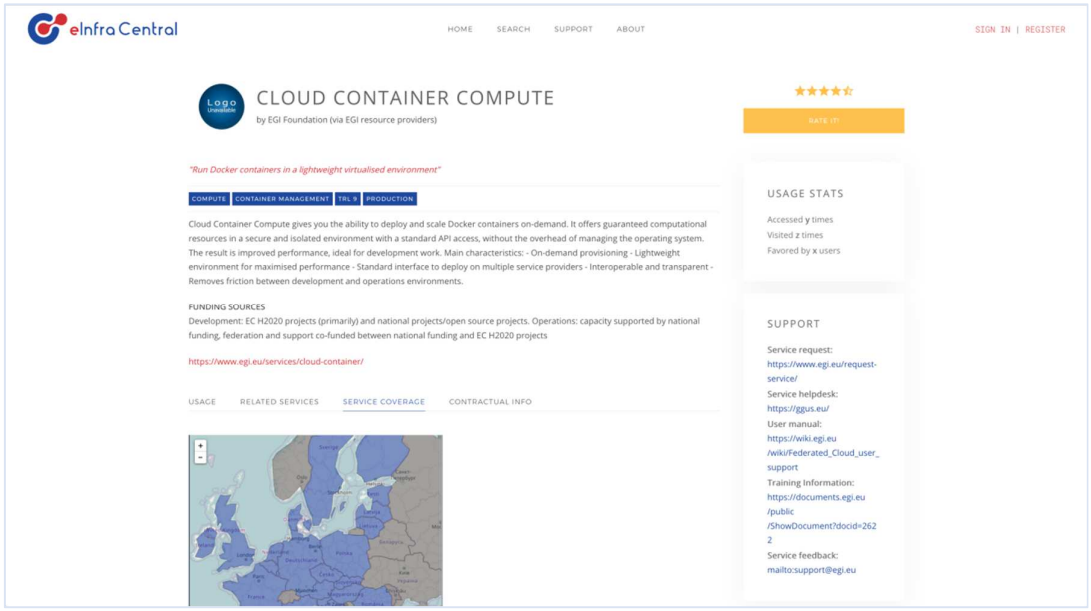


Figure 13: Service Details Page –Service Regional Coverage Tab

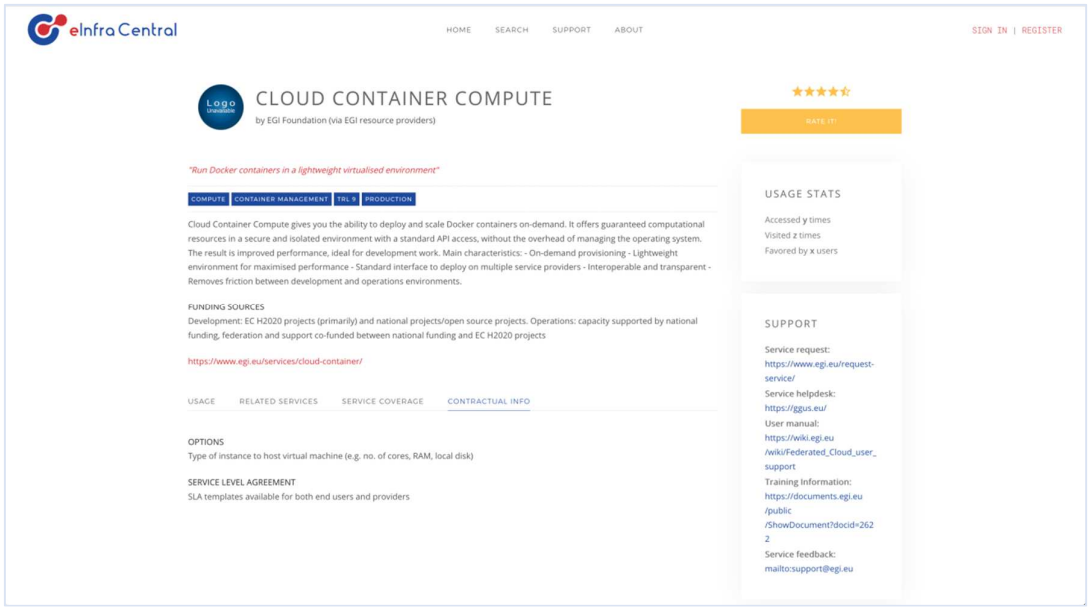


Figure 14: Service Details Page –Contractual Information Tab

5.5 eInfraCentral Service Comparison

The users can select one or more services and compare their characteristics. The service comparison page presents selected services in a table, which places each comparable attribute in a row. The user can scroll down to see more details on the comparable attributes.

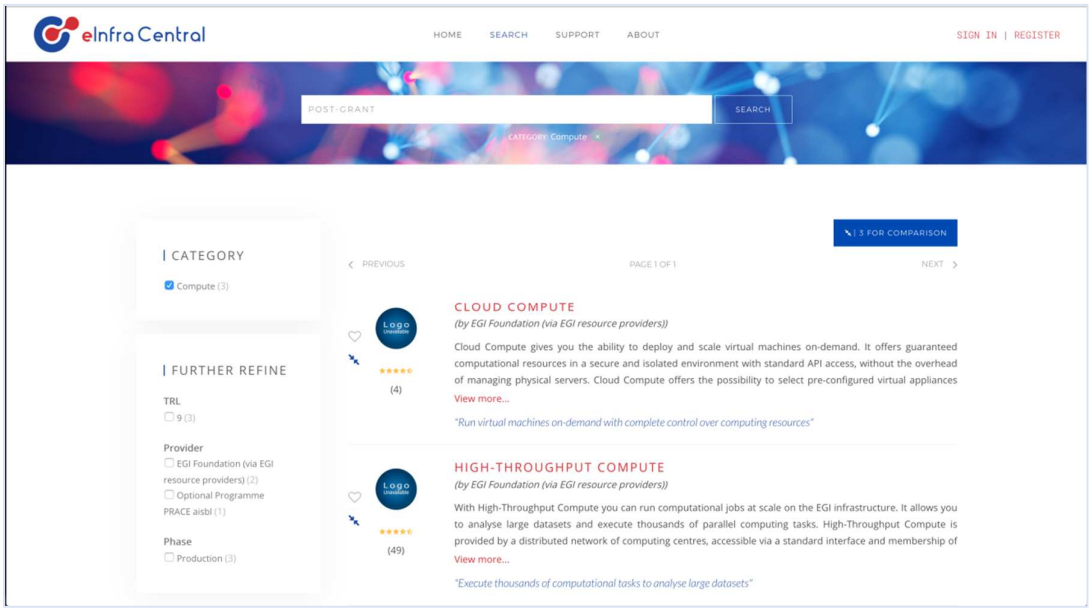


Figure 15: eInfraCentral Service Comparison – Select services for comparison

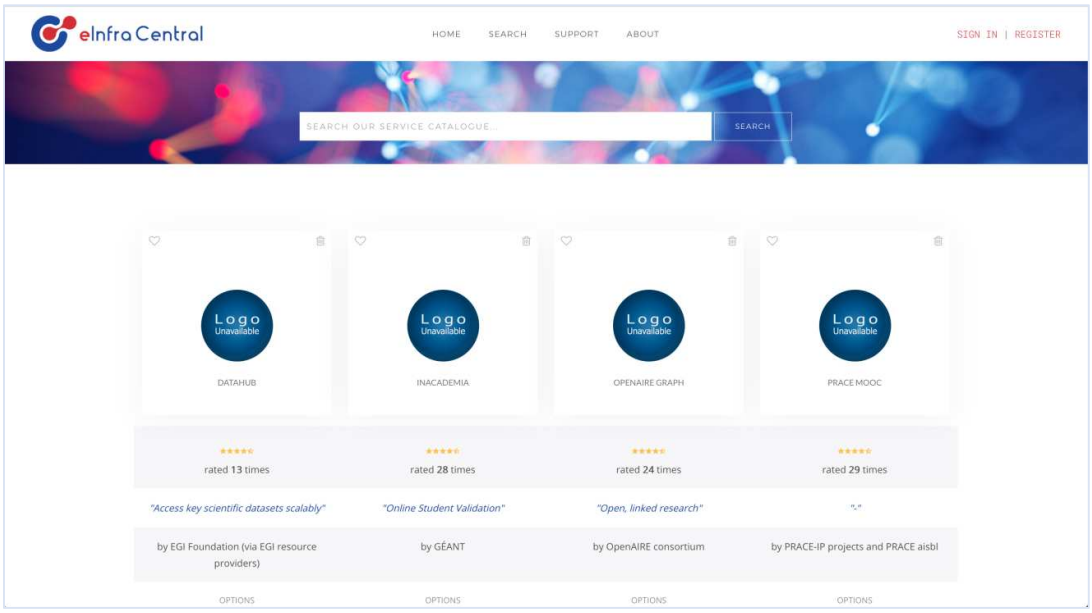


Figure 16: eInfraCentral Service Comparaison – Compare Services

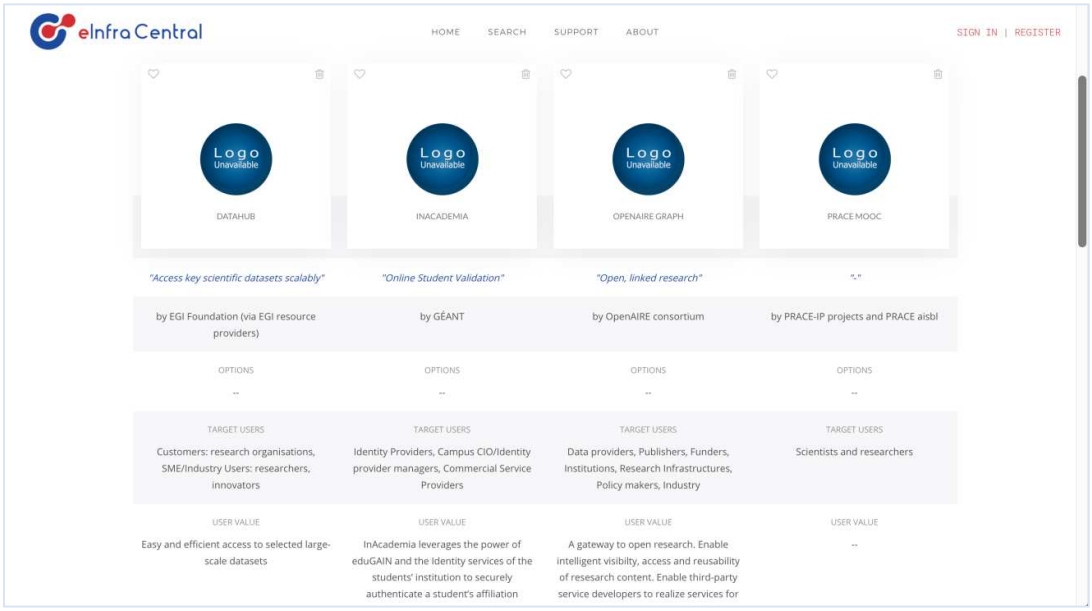


Figure 17: eInfraCentral Service Comparison – View Details in Service Comparison

5.6 eInfraCentral User Favourites

The users can select a service and add it to their favourite list by clicking the "heart" icon. The users can remove a service from the list of favourites by clicking on a favourite service.

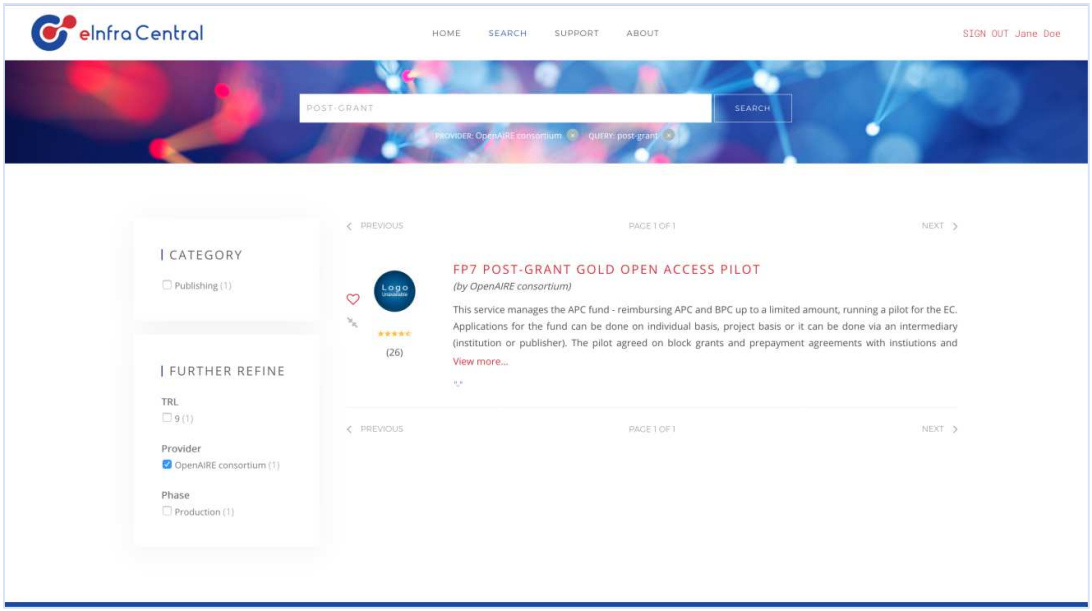


Figure 18: eInfraCentral User Favourites - Add a service to favourites

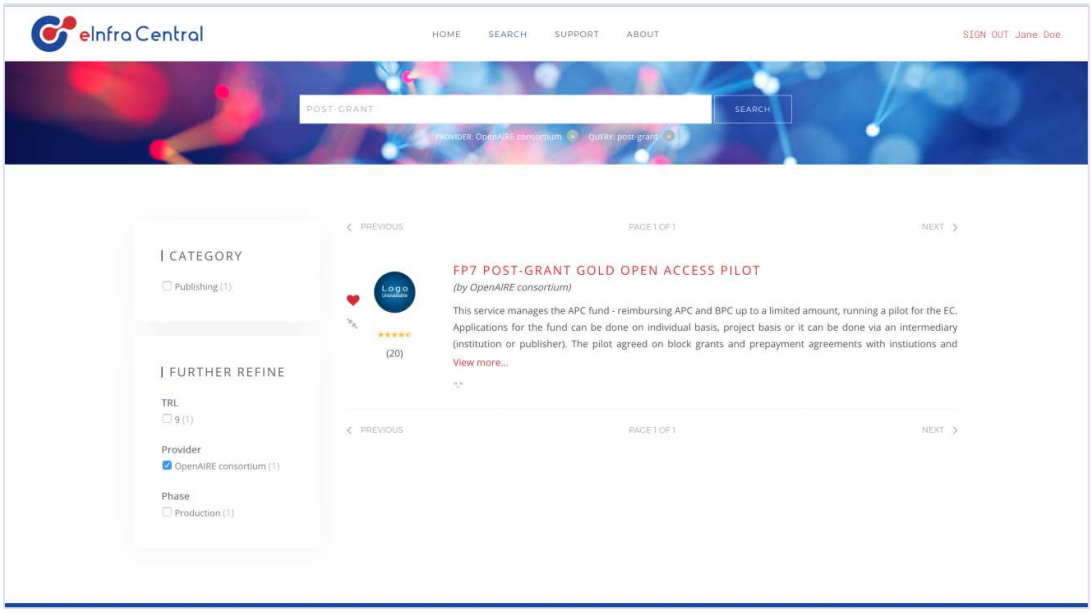


Figure 19: eInfraCentral User Favourites - Service added to favourites

5.7 eInfraCentral Signin/Signup

The sign in page allows users to login in the platform either by using one of the available social or academic logins or by filling in their credentials provided by the platform. For the latter case, the user must first sign up and fill in information about his\her profile. If the user is a service provider user, he\she must provide additional information about his\her organization.

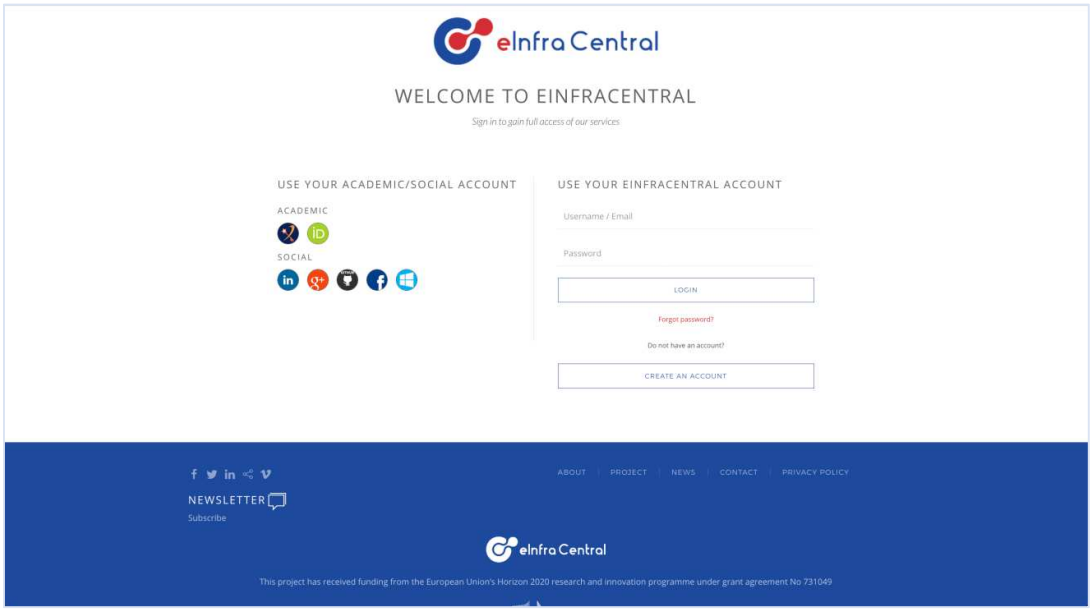


Figure 20: eInfraCentral Login and Signup Page

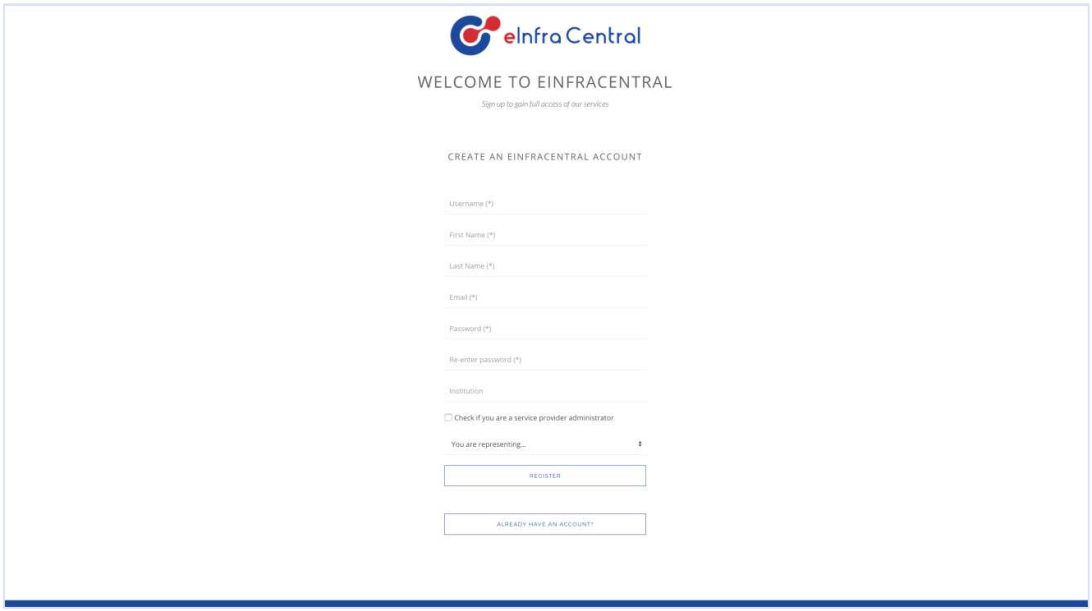


Figure 21: eInfraCentral User Registration Page – Register as a simple user

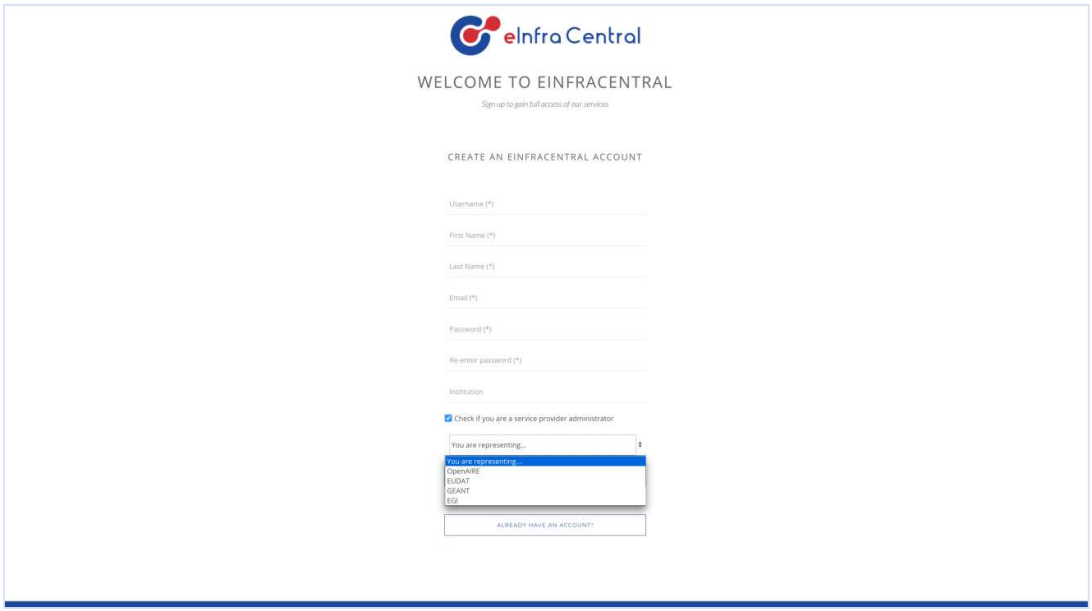


Figure 22: eInfraCentral User Registration Page – Register as a service provider

6 Users of the platform

The eInfraCentral platform aims to accommodate a range of stakeholders, including users of e-services, from research and industry, as well as e-infrastructure providers and other stakeholder groups like policy makers and funding bodies. As it is also described in Deliverable D3.1, we foresee four user groups with different authentication levels that allow access to different functionalities in the platform.

6.1 Non-authenticated end-users

Non-authenticated users are the main target group of the platform as they include a number of individual stakeholders with different interests that need to have access to e-services for their work or research, as well as to evaluate the current status of the e-infrastructure domain and make specific decisions on policy and funding. 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 view the average rating of a service as provided by other users and access the service itself at its source. They can also compare services with similar ones of the same category.

6.2 Authenticated end-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.

6.3 Service providers

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

6.4 eInfraCentral administrators

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

The following Table presents in details the mapping of user roles to the functional specifications.

Table 45: Offered functionality to each user role

Functional Specification	Public Users	Authenticated End Users	Service providers	Admin Users
FS/CAT/01 - List the categories/subcategories of a list of services	√	√	√	√
FS/CAT/02 - List the services for a specific category/subcategory	√	√	√	√
FS/CAT/03 - View overviews of a list of services	√	√	√	√
FS/CAT/04 - View details for a service	√	√	√	√
FS/CAT/05 - Navigate to the service at the remote service provider	√	√	√	√
FS/CAT/06 - Search for a service through keywords	√	√	√	√
FS/CAT/07 - View faceted filters from a list of services	√	√	√	√
FS/CAT/08 - Filter a list of services through faceted filters	√	√	√	√
FS/CAT/09 - Compare a list of services	√	√	√	√
FS/CAT/10 - List the services for a specific user group	√	√	√	√
FS/CAT/11 - Register as an end user via local authentication	√	√	√	√
FS/CAT/12 - Register as an end user via remote authentication	√	√	√	√
FS/CAT/13 - Register as a service provider user			√	
FS/CAT/14 - Login of a user		√	√	√
FS/CAT/15 - Edit the profile of a user		√	√	√
FS/CAT/16 - Add/Remove a service from the list of favourites of a user		√	√	√
FS/CAT/17 - View the list of favourite services for a user		√	√	
FS/CAT/18 - Subscribe/Unsubscribe to platform alerts		√	√	√
FS/CAT/19 - View the list of alerts for a user		√	√	√
FS/CAT/20 - Rate a service		√	√	
FS/REG/01 - Register a new service			√	√
FS/REG/02 - View a list of services provided by a service provider			√	√
FS/REG/03 - Update an existing service			√	√
FS/REG/04 - Deactivate/activate a service			√	√
FS/REG/05 - Delete a service			√	√
FS/REG/06 - List all actions applied on a service			√	√
FS/MON/01 - Add/remove a KPI type				√
FS/MON/02 - Associate/Remove a KPI type from a service			√	√
FS/MON/03 - View KPIs associated with a service			√	√
FS/MON/04 - Append/Update values of a KPI for a service			√	√
FS/MON/05 - Set the time interval for automatic updates of a KPI			√	√
FS/MON/06 - Visualize values of a KPI for a list of services	√	√	√	√
FS/MON/07 - Get average rating for a service	√	√	√	√
FS/MON/08 - Get usage statistics for a service	√	√	√	√
FS/UM/01 - View the list of users				√
FS/UM/02 - Add/Remove a user				√
FS/UM/03 - Activate/Deactivate a user				√
FS/UM/04 - Authorize a user as a service provider user				√
FS/UM/05 - Remove a user from the role of a service provider user				√
FS/UM/06 - Edit metadata enumerated values				√

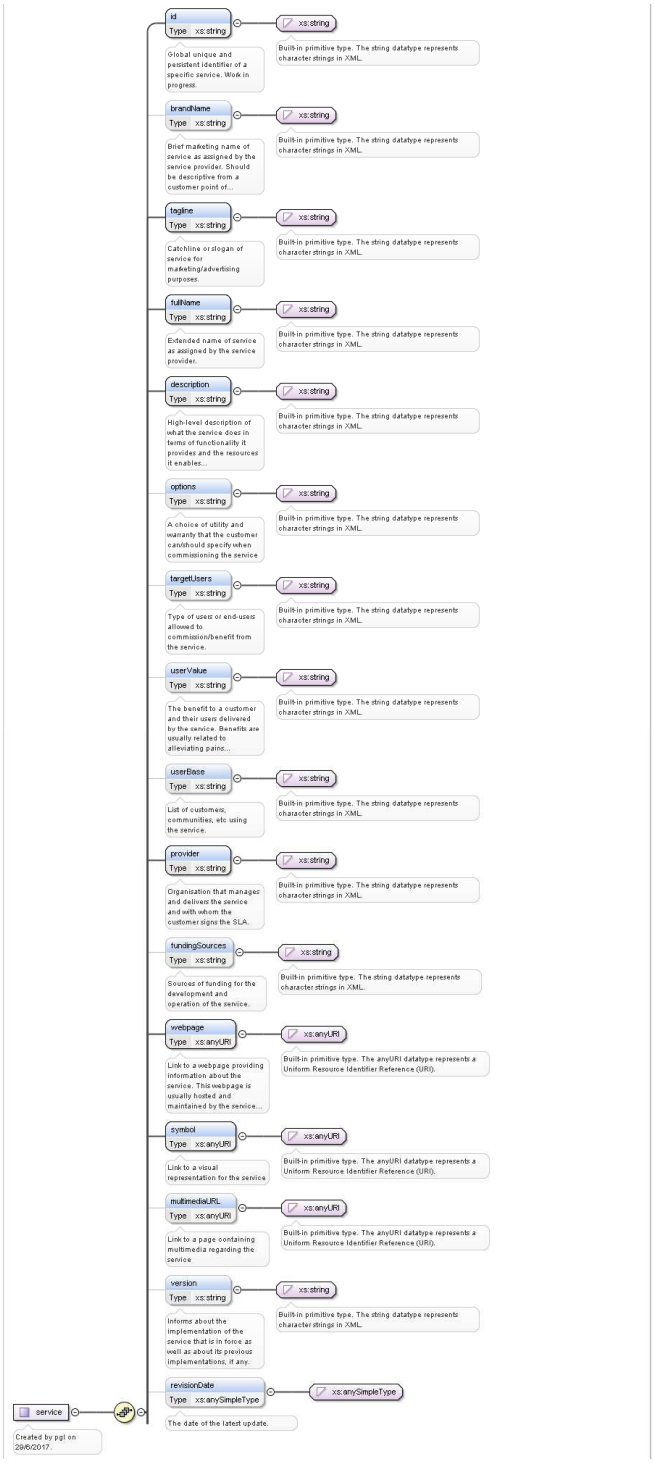
7 Conclusions

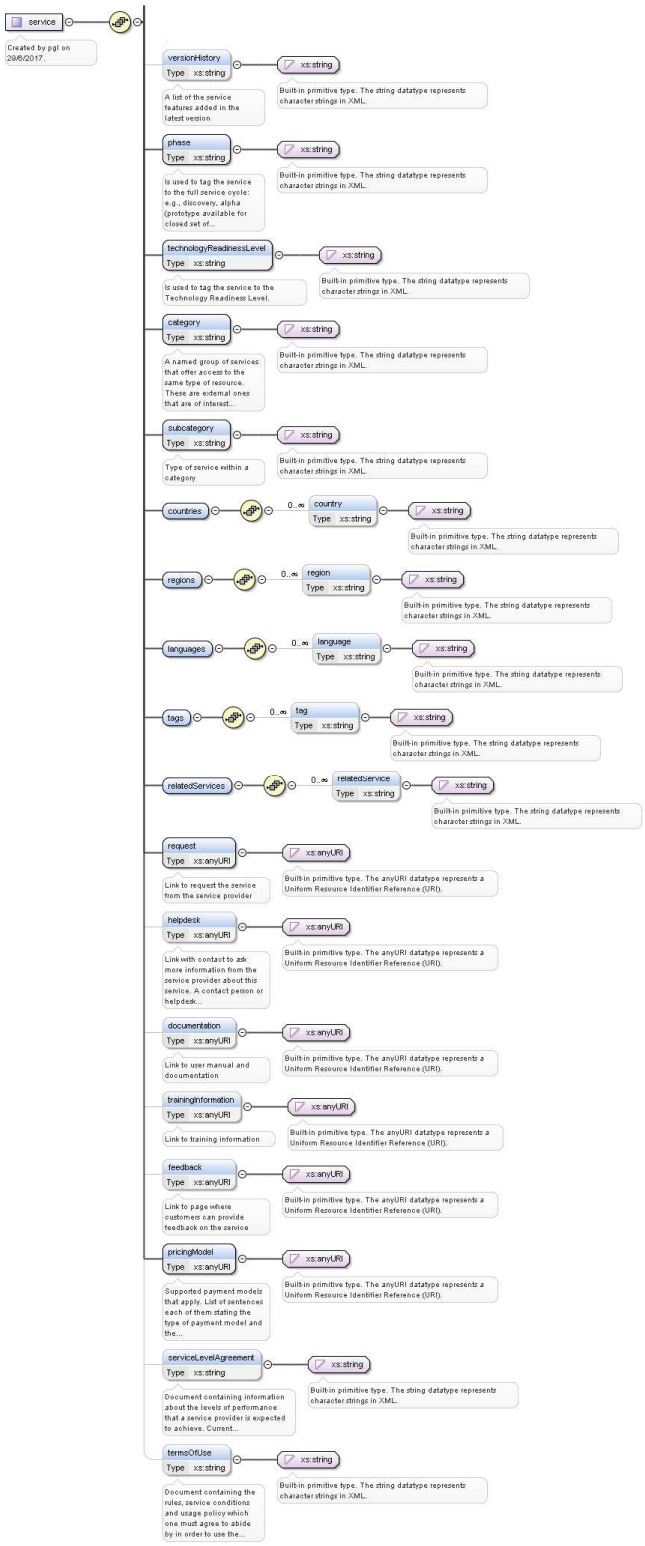
D4.1. provides the functional specifications of the ***eInfraCentral platform***, the main outcome of the eInfraCentral project, focusing on the data model designed for representing the main entities managed within the platform, the architecture of the platform, structured in different logical layers, the offered functionality of the main components and the technology used for their implementation. It also provides the list of functional specifications that are used for the development of the business logic of platform and the UI mock-ups which present the front end of the platform.

As aforementioned, the development of the eInfraCentral platform follows an agile software development approach, in which requirements and delivered functionality evolve through the collaborative effort of the development and the end user teams.

It is based on adaptive planning, evolutionary development, frequent delivery of releases, and continuous improvement, and it encourages rapid and flexible response to change. As such, this deliverable is intended to be a living document, and in its current version a first approach to the design of the platform, defining the basic architectural concepts and technologies to be used in the first phase of the implementation of the platform. It is expected that the user assessment activities following each release of the platform will introduce new requirements and changes that will enrich the list of specifications presented in this deliverable.

Annex I: Overview of the Service Schema




service
Created by pgl on 29/6/2017.
versionHistory
Type xs:string
A list of the service features added in the latest version
xs:string
Built-in primitive type. The string datatype represents character strings in XML.
phase
Type xs:string
Is used to tag the service to the full service cycle: e.g., discovery, alpha (prototype available for closed set of...
xs:string
Built-in primitive type. The string datatype represents character strings in XML.
technologyReadinessLevel
Type xs:string
Is used to tag the service to the Technology Readiness Level.
xs:string
Built-in primitive type. The string datatype represents character strings in XML.
category
Type xs:string
A named group of services that offer access to the same type of resource. These are external ones that are of interest...
xs:string
Built-in primitive type. The string datatype represents character strings in XML.
subcategory
Type xs:string
Type of service within a category.
xs:string
Built-in primitive type. The string datatype represents character strings in XML.
countries
0..∞
country
Type xs:string
xs:string
Built-in primitive type. The string datatype represents character strings in XML.
regions
0..∞
region
Type xs:string
xs:string
Built-in primitive type. The string datatype represents character strings in XML.
languages
0..∞
language
Type xs:string
xs:string
Built-in primitive type. The string datatype represents character strings in XML.
tags
0..∞
tag
Type xs:string
xs:string
Built-in primitive type. The string datatype represents character strings in XML.
relatedServices
0..∞
relatedService
Type xs:string
xs:string
Built-in primitive type. The string datatype represents character strings in XML.
request
Type xs:anyURI
Link to request the service from the service provider
xs:anyURI
Built-in primitive type. The anyURI datatype represents a Uniform Resource Identifier Reference (URI).
helpdesk
Type xs:anyURI
Link with contact to ask more information from the service provider about this service. A contact person or helpdesk...
xs:anyURI
Built-in primitive type. The anyURI datatype represents a Uniform Resource Identifier Reference (URI).
documentation
Type xs:anyURI
Link to user manual and documentation
xs:anyURI
Built-in primitive type. The anyURI datatype represents a Uniform Resource Identifier Reference (URI).
trainingInformation
Type xs:anyURI
Link to training information
xs:anyURI
Built-in primitive type. The anyURI datatype represents a Uniform Resource Identifier Reference (URI).
feedback
Type xs:anyURI
Link to page where customers can provide feedback on the service
xs:anyURI
Built-in primitive type. The anyURI datatype represents a Uniform Resource Identifier Reference (URI).
pricingModel
Type xs:anyURI
Supported payment models that apply. List of sentences each of them stating the type of payment model and the...
xs:anyURI
Built-in primitive type. The anyURI datatype represents a Uniform Resource Identifier Reference (URI).
serviceLevelAgreement
Type xs:string
Document containing information about the levels of performance that a service provider is expected to achieve. Current...
xs:string
Built-in primitive type. The string datatype represents character strings in XML.
termsOfUse
Type xs:string
Document containing the rules, service conditions and usage policy which one must agree to abide by in order to use the...
xs:string
Built-in primitive type. The string datatype represents character strings in XML.