



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-INFRA supp-2016-2017**

## Milestone MS36

### Updated specifications of eInfraCentral APIs (v.2.0)



**eInfra Central**

|                                   |   |
|-----------------------------------|---|
| <b>Type (Distribution Level):</b> | Public  |
| <b>Deliverable Leader:</b>        | UOA   |
| <b>Authors:</b>                   | George Papastefanatos, Akrivi Katifori, Natalia Manola, Stefania Martziou, Antonis Lempesis, Konstantinos Spyrou, Michael Zouros, Ioannis Balasis (UoA) |
| <b>Due Date:</b>                  | 30.06.2019  |
| <b>Actual Submission Date:</b>    | 30.06.2019  |
| <b>Version:</b>                   | 1.0   |

**Abstract:** This document provides the updated specifications (v2.0) of the REST Application programming interface (API) methods of the eInfraCentral as reported in D3.3 “Guidelines for schema representation and APIs”. The updates refer to the extension and adaptation of the methods to the new Service Description Template (v.2.0), designed to accommodate new features and new classification schemes in the service description. The REST API are used for the exchange of information from service providers to the eInfraCentral Gateway and its provisioning to third party applications. The APIs methods offer the functionality to programmatically access, search and retrieve the contents of the eInfraCentral catalogue, such as services, service providers, service performance indicators and service analytics.



HELLENIC REPUBLIC  
National and Kapodistrian  
University of Athens



Leibniz  
Universität  
Hannover



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                      |
|------------|---------|--|---|
| 30.06.2019 | 1.0     | George Papastefanatos  | Final document                                      |
| 25.06.2019 | 0.1     | Stefania Martziou, Antonis Lempesis,<br>Konstantinos Spyrou, Michael Zouros, Ioannis<br>Balasis , Natalia Manola | Comments received                                   |
| 20.06.2019 | 0.1.    | George Papastefanatos  | 1 <sup>st</sup> Draft of the document<br>circulated |
|            |         |  |   |

### Copyright © eInfraCentral consortium, 2017-2018

This document contains material, which is the copyright of certain eInfraCentral consortium members and the EC, and may not be reproduced or copied without permission, except as mandated by the European Commission Grant Agreement no 731049 for reviewing and dissemination purposes. The information contained in this document is provided by the copyright holders "as is" and any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall the members of the eInfraCentral collaboration, including the copyright holders, or the European Commission be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of the information contained in this document, even if advised of the possibility of such damage.

## Executive Summary

The **eInfraCentral Gateway** is one of the major outputs of the eInfraCentral project. It aims to offer the end-users a **central entry point/gateway to a harmonised and aggregated service catalogue and an additional distribution channel to e-infrastructure services offered disparately by each e-Infrastructure or other aggregators**.

To achieve this, eInfraCentral is collecting service descriptions and related data through registration and synchronization methods, and aggregating those service offerings into a uniform service catalogue. The catalogue is offered via the eInfraCentral Gateway employing formal/standardised guidelines and Application Programming Interfaces (APIs) to offer up-to-date information to end-users.

This report (Milestone MS36 - Update of eInfraCentral APIs (v.2.0)) provides the updated specifications (v2.0) of the REST Application programming interface (API) methods of the eInfraCentral as reported in D3.3 “Guidelines for schema representation and APIs”. The updates refer to the extension and adaptation of the methods to the new Service Description Template (v.2.0), designed to accommodate new features and new classification schemes in the service description. The REST API are used for the exchange of information from service providers to the eInfraCentral Gateway and its provisioning to third party applications. The APIs methods offer the functionality to programmatically access, search and retrieve the contents of the eInfraCentral catalogue, such as services, service providers, service performance indicators and usage statistics collected through the eInfraCentral Gateway, as well as terms and vocabularies used for service classification. In brief, it provides the following contributions:

- An overview of the different use cases offered by the eInfraCentral Gateway for the e-infrastructure service providers to register and synchronise their service descriptions with the eInfraCentral catalogue through an API, as well as for service catalogue consumers to retrieve service-related information.
- The most recent version of the resource model SDT2.0 (i.e. main entities and their interrelationships) that eInfraCentral offers for the exchange of information between third party systems and the eInfraCentral Gateway.
- The detailed specifications of the v2.0 API methods, which are organised according to their functionality in the following controllers:
  - The **Service Controller** offers the functionality for managing a service resource, such as adding, updating, validating a service resource, as well as retrieving service-related information with various different criteria.
  - The **Indicator Controller** offers the functionality for managing an indicator definition, i.e. an indicator resource, used for monitoring service offering performance.
  - The **Measurement Controller** offers the functionality for managing an indicator measurement as reported by service providers.
  - The **Provider Controller** offers the functionality for managing a service provider resource.
  - The **Funder Controller** offers the functionality for managing a funder resource.
  - The **Statistics Controller** offers the functionality for retrieving statistics collected in the eInfraCentral Gateway, such as service orders per day for a service or aggregate visits per day for all services offered by a provider.
  - The **User Controller** offers the functionality to remote systems to login and receive a JWT token for authenticating further requests in the eInfraCentral Gateway.
  - The **Vocabulary Controller** offers the functionality to retrieve the vocabularies and terms used in the eInfraCentral Gateway for service classification.

The complete [eInfraCentral API documentation](#) is available online, generated using frameworks compliant with OpenAPI Specification 2.0. It is also available at the [Github](#) repository of the project.

## Table of Contents

|  |    |
|--|----|
| Document Revision History  | 2  |
| Executive Summary  | 3  |
| Acronyms   | 8  |
| Short Names of Partner Organisations                                 | 9  |
| 1 Introduction   | 10 |
| 2 Use of the eInfraCentral APIs                                      | 13 |
| 2.1 Provisioning of eInfraCentral Catalogue information              | 13 |
| 2.1.1 Use Cases for retrieving service-related information           | 14 |
| 2.1.2 Use Cases for retrieving provider-related information          | 16 |
| 2.1.3 Use Cases for retrieving indicators and analytics              | 17 |
| 2.1.4 Use Cases for retrieving vocabulary-related information        | 18 |
| 2.1.5 Use Cases for retrieving funder-related information            | 19 |
| 2.2 Collection of eInfraCentral service information                  | 20 |
| 3 Representation of eInfraCentral Resources                          | 23 |
| 3.1 The eInfraCentral Resource Model                                 | 23 |
| 3.2 eInfraCentral Resource Vocabularies                              | 37 |
| 4 The eInfraCentral API specifications                               | 45 |
| 4.1 The eInfraCentral API technology stack                           | 45 |
| 4.2 eInfraCentral API methods  | 45 |
| 4.2.1 Service Controller   | 48 |
| 4.2.2 Service Provider Controller                                    | 54 |
| 4.2.3 Funder Controller  | 56 |
| 4.2.4 Indicator Controller   | 58 |
| 4.2.5 Measurement Controller   | 60 |
| 4.2.6 Statistics Controller  | 63 |
| 4.2.7 Vocabulary Controller  | 65 |
| 4.3 User Authentication  | 69 |
| 4.4 Synchronization between a remote and the eInfraCentral catalogue | 69 |
| 5 Conclusions  | 69 |
| Appendix 1: Detailed Description of the eInfraCentral resource model | 70 |
| Appendix 2: Guidelines for the use of the API by a Service Provider  | 85 |

## List of Figures

|  |    |
|--|----|
| Figure 1: A model for the development of the Catalogue of Research/Scientific Resources .....        | 10 |
| Figure 2: An overview of the eInfraCentral APIs.....   | 11 |
| Figure 3: An architecture of the eInfraCentral Gateway with the interoperability layer highlighted.. | 12 |
| Figure 4: UC1.1 – Get a description of a service .....   | 14 |
| Figure 5: UC1.2 – Get all versions of a service .....  | 14 |
| Figure 6: UC1.3 – Get a past version of a service.....   | 15 |
| Figure 7: UC1.4 – Get multiple services based on a set of Service IDs .....                          | 15 |
| Figure 8: UC1.5 – Get all services in the eInfraCentral Catalogue .....                              | 15 |
| Figure 9: UC1.6 – Get a list of services provided by a service provider.....                         | 16 |
| Figure 10: UC1.7 – Get all services in the catalogue organised by a service attribute.....           | 16 |
| Figure 11: UC1.7 – Get a list of services based on filters.....                                      | 16 |
| Figure 12: UC2.1 – Get information about a specific service provider .....                           | 17 |
| Figure 13: UC2.2 – Get the list of service providers .....   | 17 |
| Figure 14: UC3.1 – Get analytics for a service .....   | 18 |
| Figure 15: UC3.2 – Get analytics for a service provider.....   | 18 |
| Figure 16: UC3.3 – Get a list of all indicators associated with a service .....                      | 18 |
| Figure 17: UC3.4 – Get the values of an indicator of a service .....                                 | 18 |
| Figure 18: UC10 – Get a list of service categories.....  | 19 |
| Figure 19: UC5.1 – Get information about a specific funder.....                                      | 19 |
| Figure 20: UC5.2 – Get the list of funders.....  | 19 |
| Figure 21: UC5.2 – Get statistics about services funded by a specific funder .....                   | 20 |
| Figure 22: Workflow for the update of eInfraCentral information by a service provider .....          | 20 |
| Figure 23: UC6.1 – Update of information of an existing service via a PULL method .....              | 21 |
| Figure 24: UC6.2 – Update of information of an existing service via a PUSH method .....              | 21 |
| Figure 25: UC6.3 – Validate an update of a service via a push method .....                           | 22 |
| Figure 26: UC6.4 – Update of information of a service provider via a push method.....                | 22 |
| Figure 27: UC6.5 – Addition of a new measurement for a performance indicator via a push method.....  | 22 |
| Figure 28: An overview of the eInfraCentral Gateway resource model .....                             | 23 |

## List of Tables

|   |    |
|---|----|
| Table 1: The service resource represented in JSON .....   | 24 |
| Table 2: The options resource represented in JSON.....  | 30 |
| Table 3: The service provider resource represented in JSON.....   | 31 |
| Table 4: The funder resource represented in JSON.....   | 32 |
| Table 5: The Service Level Target and Performance Indicator resource represented in JSON .....  | 33 |
| Table 6: The Indicator Measurement resource represented in JSON .....   | 34 |
| Table 7: The statistics resource represented in JSON.....   | 35 |
| Table 8: The vocabulary resource represented in JSON .....  | 36 |
| Table 9: An overview of the eInfraCentral API methods .....   | 47 |
| Table 10: POST /Service – Adds a new service.....   | 48 |
| Table 11: POST /Service/validate – Validates the service without actually changing the catalogue ...  | 49 |
| Table 12: PUT /Service – Updates a service.....   | 49 |
| Table 13: GET /service/{id} – Gets a service based on the service ID.....   | 50 |
| Table 14: GET /service/all – Filters a list of services based on a set of filters or get a list of all services in the eInfraCentral Catalogue..... | 50 |
| Table 15: GET /service/by/{field} – Gets all services in the catalogue organized by a field .....   | 51 |
| Table 16: GET /service/byID/{ids} – Gets a list of services based on a set of IDs .....   | 52 |
| Table 17: GET /service/{id}/{version} – Gets versions of a specific service .....   | 52 |
| Table 18: PUT /provider – Updates a service provider.....   | 54 |
| Table 19: GET/provider/{id} – Gets a specific provider .....  | 54 |
| Table 20: GET/provider/all – Gets all providers in the catalogue.....   | 54 |
| Table 21: GET/provider/{id}/services – Gets all services of a provider .....  | 55 |
| Table 22: GET/funder/{id} – Gets a specific funder .....  | 56 |
| Table 23: GET/funder/all – Gets all funders in the catalogue .....  | 56 |
| Table 24: GET/funder/funderStats/{id} – Gets stats about services funded by a specific funder.....  | 57 |
| Table 25: POST /indicator – Adds a new indicator.....   | 58 |
| Table 26: PUT /indicator – Updates an existing indicator .....  | 58 |
| Table 27: GET /indicator – Gets an existing indicator .....   | 59 |
| Table 28: GET/indicator/all – Gets all indicators in the catalogue .....  | 59 |
| Table 29: POST /measurement – Adds a new measurement.....   | 60 |
| Table 30: PUT /measurement – Updates an existing measurement.....   | 61 |
| Table 31: GET /measurement – Gets an existing measurement .....   | 61 |
| Table 32: GET /measurement/service/{id} – Gets all Measurements for the specific service.....   | 62 |
| Table 33: GET /measurement/latest/service/{id} – Gets all latest Measurements for the specific service.....   | 63 |
| Table 34: GET /stats/provider/favourites/{id} – Gets total favourites for all services of a provider ...  | 63 |
| Table 35: GET /stats/provider/ratings/{id} – Gets average ratings for all services of a provider.....   | 64 |
| Table 36: GET /stats/provider/visits/{id} – Gets total visits for all services of a provider.....   | 64 |
| Table 37: GET /stats/provider/visitation/{id} – Gets % of visits for all services of a provider .....   | 65 |
| Table 38: GET /vocabulary – Gets all terms for a vocabulary .....   | 65 |
| Table 39: GET /vocabulary/all – Gets vocabulary terms based on filters.....   | 66 |
| Table 40: GET /vocabulary/countries/EU (or /WW) – Gets the list of country codes belonging to EU or Worldwide.....                                  | 67 |
| Table 41: Service resource .....  | 70 |
| Table 42: Indicator resource .....  | 80 |

Table 43: Indicator Measurement Resource .....81

Table 44: Options Resource .....81

Table 45: Service Provider Resource.....82

Table 46: Funder Resource .....84

Table 47: Authenticated User Resource .....84

## Acronyms

Table 1: Acronyms

| Abbreviation | Meaning  |
|--------------|--|
| API          | Application Programming Interface                        |
| CA           | Consortium Agreement                                     |
| DoA          | Description of Action                                    |
| DoW          | Description of Work                                      |
| EC           | European Commission                                      |
| EIC          | eInfraCentral  |
| e-IRG        | e-Infrastructure Reflection Group                        |
| ERA          | European Research Area                                   |
| ESFRI        | European Strategic Framework for Research Infrastructure |
| IT           | Information Technology                                   |
| ITSM         | IT Service Management                                    |
| 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                               |
| UI           | User Interface   |
| 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      | GÉANT 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 eInfraCentral project has a key role in the European Open Science Cloud ecosystem. It is the first effort to provide access to a uniform catalogue of services offered by major e-Infrastructures. Figure 1 presents an overview of the eInfraCentral vision in the research/scientific resources market. Service management and offering are performed by different providers, comprising project-specific e-Infrastructures and service providers (SPs), region- and domain-specific providers and aggregators (e.g. data repositories, SW aggregators, technology enablers, etc.), and flagship European Research e-Infrastructures (e.g. EGI, EUDAT, PRACE, GEANT, OpenAIRE), acting as proxies or aggregating service access and ordering for multiple other providers.

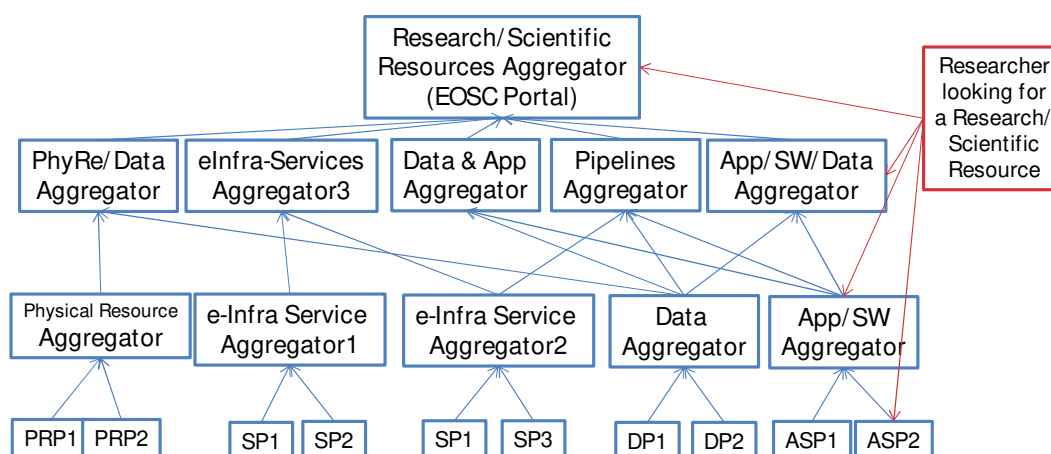


Figure 1: A model for the development of the Catalogue of Research/Scientific Resources

The main goal of eInfraCentral is to cope with the highly fragmented and multilevel landscape and deliver a single-entry point for end-users to browse a **harmonised** service catalogue, thus enhancing **accessibility, discoverability, monitoring and comparability** of e-Science services. It aims at ensuring that a broader and more varied set of users (including industry) discovers and accesses the existing and developing e-infrastructure capacity.

The eInfraCentral Gateway – one of the major outputs of eInfraCentral project – will **act as the main entry point to European e-Infrastructure services**. It collects (through registration of services and automatic updating of service-related information), aggregates content about service offerings for and offers a uniform service catalogue through standardized guidelines and APIs (Application programming interface).

Two major achievements developed so far by the project include: a) a standardised description of research resources (Service Description Template -SDT, as presented in D3.2), namely the description of an electronic service resource offered by service providers, as well as a set of indicators that can be used for monitoring and evaluating the performance and evolution of services; and b) standardised APIs for the exchange of service/resources-related information among service providers and aggregators of this ecosystem (presented in D3.3).

eInfraCentral has shared all its outputs with all key stakeholders in the wider ecosystem to underpin faster convergence towards a framework that allows an automatic exchange of service-related information. It drives interoperability to allow service providers for additional distribution channels and end-users to select their preferred aggregation point and abstraction level.

It is now recognised by the community that a **common approach to a) describing and b) exchanging service-related information is the way forward** to increase accessibility, discoverability, interoperability, aggregation, monitoring and comparability of services. This will increase their uptake by the users and will allow for enhancing the understanding of gaps and improvements that could be delivered by e-infrastructure services. eInfraCentral, therefore, is focusing most of its efforts on this direction.

The initial SDT models (v1.\*) designed and evolved during the initial stages of the project aimed at capturing and representing features from the services offered by the 5 service providers of the project. The expansion of the catalogue with new providers and services has led to the extension of this model to a new updated version SDT v2.0, which towards the finalization of the project, has incorporated new features and refined existing attributes. Moreover, it has provided a great improvement and refinement of the service classifications used in the catalogue in order to support the inclusion in the catalogue of services coming from very diverse service providers (both e-Infrastructures and physical research infrastructures) and scientific disciplines.

Following the new release of the SDT, this report extends the initial specifications of the APIs (v1.0) of the eInfraCentral Gateway as reported in D3.3 “Guidelines for schema representation and APIs”. The updates APIs v2.0 refer to the extension and adaptation of the methods to the new Service Description Template (v.2.0), designed to accommodate new features and new classification schemes in the service description. An overview functionality of the APIs is depicted in Figure 2.

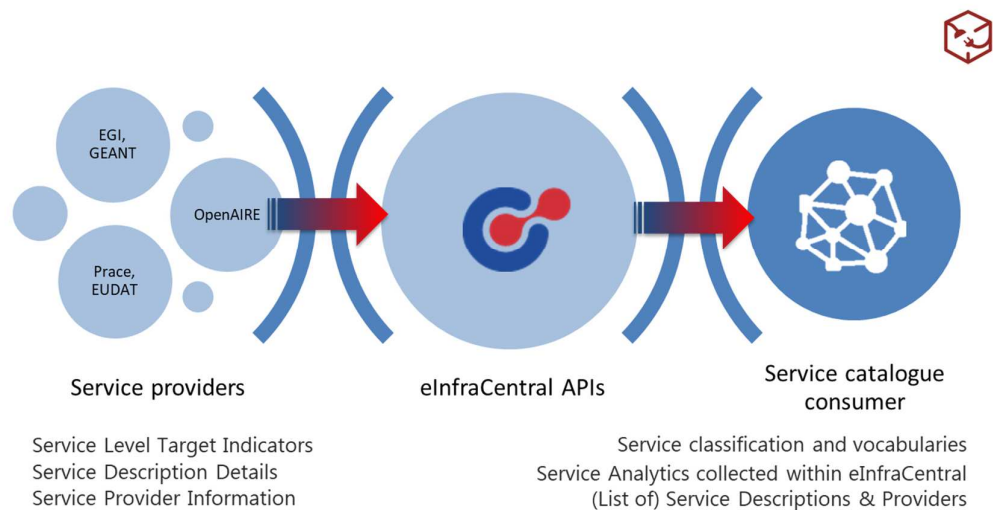


Figure 2: An overview of the eInfraCentral APIs

Service providers can use the APIs to populate service-related information in the eInfraCentral Gateway, whereas catalogue consumers can retrieve service providers’ information as well as information collected within the eInfraCentral Gateway and integrate it in their applications.

The eInfraCentral Gateway overall architecture (as introduced in D4.1) is shown in Figure 3.

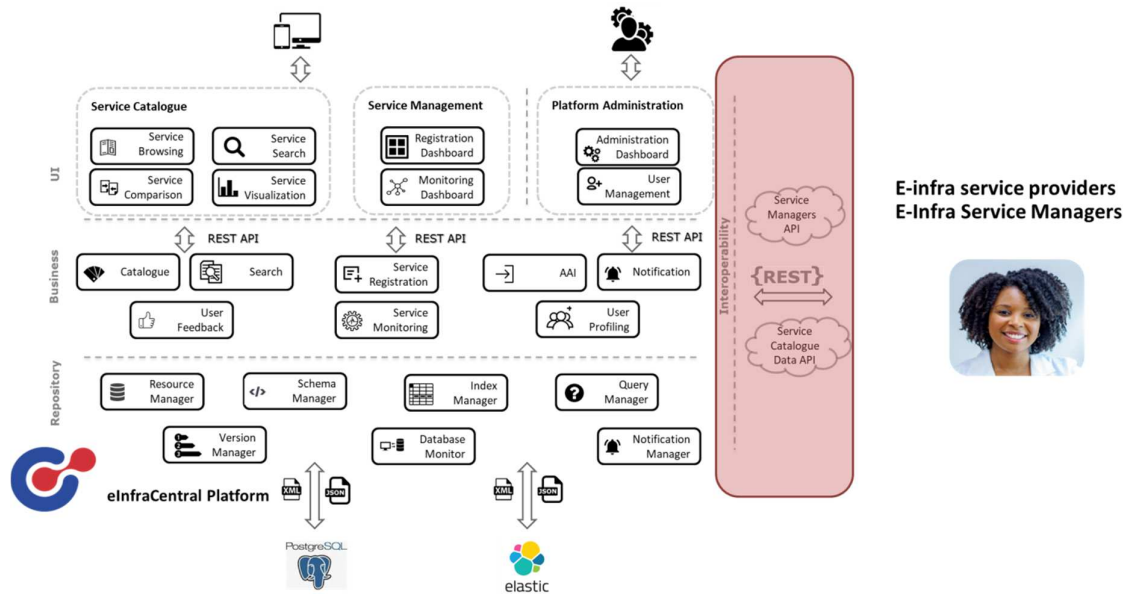


Figure 3: An architecture of the eInfraCentral Gateway with the interoperability layer highlighted

The highlighted part is the Interoperability Layer implementing the APIs for the exchange of data between the eInfraCentral Gateway and external systems. The update of service-related information in the eInfraCentral Gateway has been presented in D4.2. It operates both in a manual 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.

For completeness, this document refines these APIs and provides the full specifications for the provisioning of information towards the consumers of the service catalogue.

Following the public launch of the project, the eInfraCentral Gateway is available at [www.eInfraCentral.eu](http://www.eInfraCentral.eu).

The complete [eInfraCentral API documentation](#) is available online, generated using frameworks compliant with OpenAPI Specification 2.0. It is also available at the [Github](#) repository of the project. Following the structure of the specification of the API v1.0 the rest of this report is organised as follows:

- Chapter 2 provides detailed use cases for the exchange of information through the eInfraCentral Gateway APIs;
- Chapter 3 describes an overview of the eInfraCentral Gateway underlying resource model, denoting the main entities that are accessible through the API, as well as their representation in the API methods;
- Chapter 4 provides the API specifications;
- Chapter 5 concludes the deliverable;
- Appendix 1 provides a detailed eInfraCentral resource model (as it was defined in SDT 2.0); and, finally,
- Appendix 2 presents a brief set of guidelines, i.e. a step-by-step example, for service provider to make use of the APIs to populate the eInfraCentral catalogue.

*In the context of this report, the term API refers to the updated version v2.0 of the API methods.*

## 2 Use of the eInfraCentral APIs

The eInfraCentral APIs allow to manage the entire lifecycle of the catalogue elements, i.e. from data collection from the service providers to data provisioning to third party systems. Below, we provide a set of simple use cases that demonstrate the use of the API by relevant stakeholders, namely e-Infrastructures and in general service providers that wish to manage their service within the eInfraCentral Gateway. The first set describes the use cases for offering catalogue information to third parties through the REST API of the eInfraCentral Gateway. The second set of use cases describes the management of service-related information, i.e. service description, providers, indicators, from the service providers.

### 2.1 Provisioning of eInfraCentral Catalogue information

The eInfraCentral Gateway offers a facility for a remote system to request and get catalogue information via a variety of API methods. Namely it offers the ability to:

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

The above list of use cases is a first set of methods that is available to remote systems. It is not intended to be a closed list; rather it will be extended with more methods according to the type of information collected and maintained in the elnfraCentral Gateway. In the following sections we provide more details about the aforementioned use cases. All use cases are described in the form of UML sequence diagrams, where EIC denotes the elnfraCentral endpoint and SP denotes the remote system endpoint (service provider or a third party).

### 2.1.1 Use Cases for retrieving service-related information

**UC1.1. Get a description of a service:** This use case enables a remote system to request a description of a service in the elnfraCentral catalogue based on its ID. Each service in the elnfraCentral catalogue is uniquely identified by a service ID, generated during service registration. The following request provides as input the unique service ID. It validates the input request and returns a single service object (XML or JSON) with the service description.

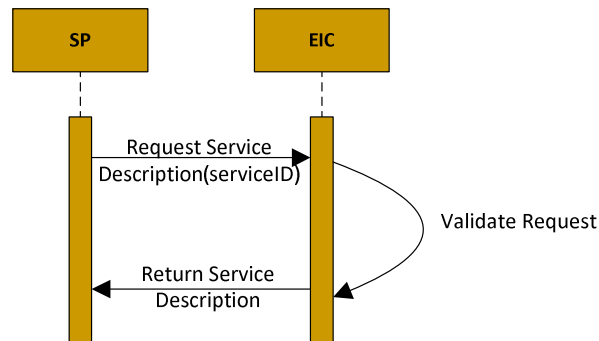


Figure 4: UC1.1 – Get a description of a service

**UC1.2. Get all versions of a service:** This use case enables a remote system to request all versions of a service. Similarly to the above, it provides the service ID.

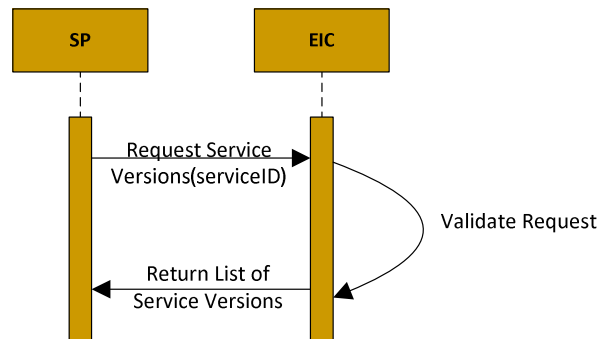


Figure 5: UC1.2 – Get all versions of a service

**UC1.3. Get an earlier version of a service:** This use case enables a remote system to request for a specific past version of a service. Similarly to the above, it provides the service ID, as well as the version identifier (v2.35), which identifies a past version of the service in the catalogue.

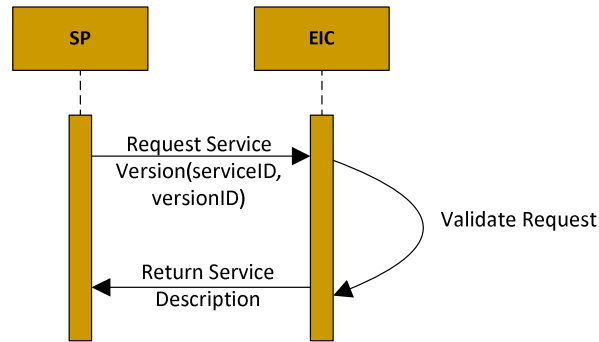


Figure 6: UC1.3 – Get a past version of a service

**UC1.4. Get multiple services based on a set of Service IDs:** This use case enables a remote system to request the description of multiple service descriptions in the eInfraCentral catalogue based on their IDs, e.g. for comparing them across a set of characteristics.

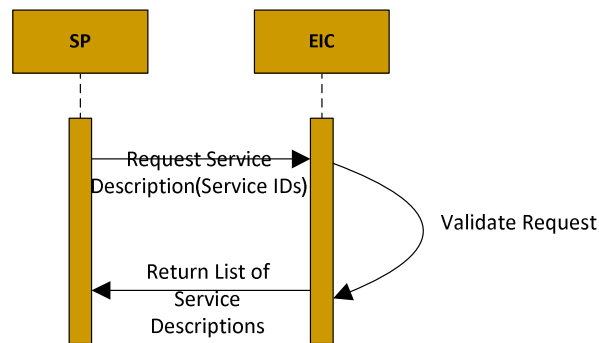


Figure 7: UC1.4 – Get multiple services based on a set of Service IDs

**UC1.5. Get a list of all services in the eInfraCentral Catalogue:** This use case enables a remote system to request all latest versions of services offered by all service providers in the eInfraCentral catalogue. It performs an empty request and returns the list of all service objects in the eInfraCentral catalogue.

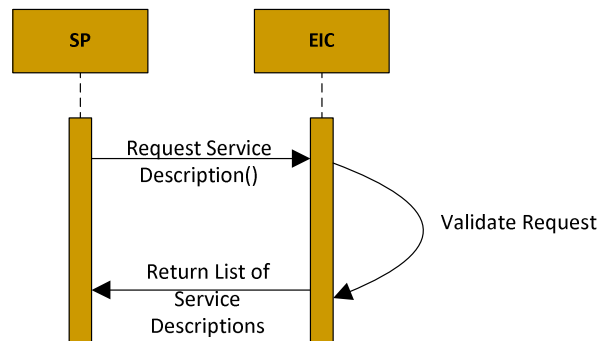


Figure 8: UC1.5 – Get all services in the eInfraCentral Catalogue

**UC1.6. Get a list of services provided by a service provider:** This use case enables a remote system to request for all latest versions of services offered by a service provider. It provides as input only the provider ID and returns a list of service descriptions.

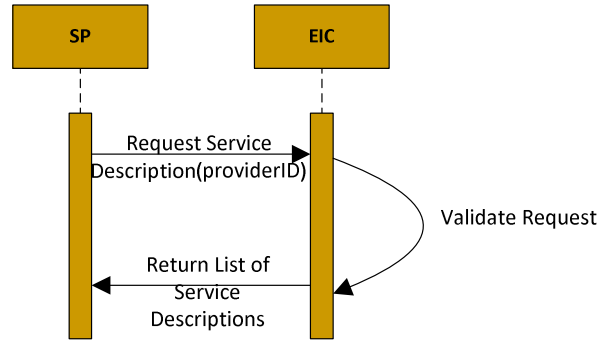


Figure 9: UC1.6 – Get a list of services provided by a service provider

**UC1.7. Get all services in the catalogue organised by a service attribute:** This use case enables a remote system to request for the description of all services in the eInfraCentral catalogue and organise them according to the values of an attribute, e.g. get all services organised in categories.

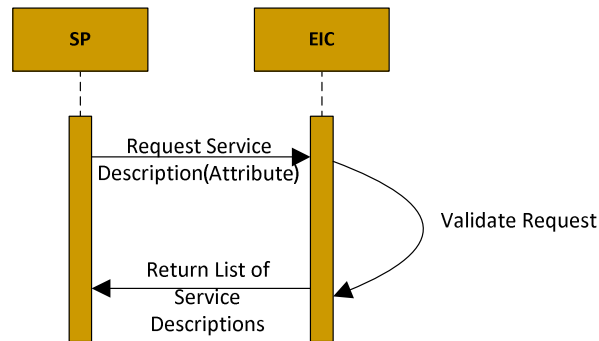


Figure 10: UC1.7 – Get all services in the catalogue organised by a service attribute

**UC1.8. Get a list of services based on filters:** This use case enables a remote system to request all latest versions of services and filter them based on a set of criteria applied on the attributes of the service. The criteria can be either a keyword query and key-value pairs for filtering on the values of specific attributes, such as the category or the provider of the services.

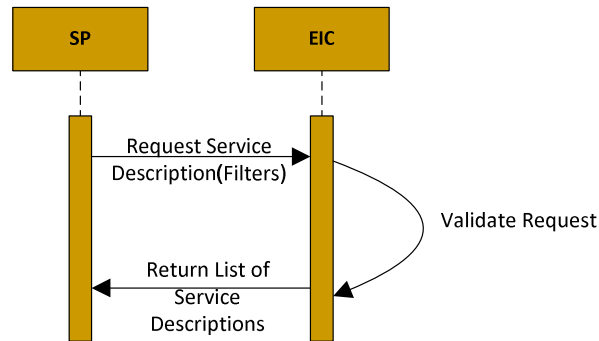


Figure 11: UC1.7 – Get a list of services based on filters

## 2.1.2 Use Cases for retrieving provider-related information

**UC2.1. Get information about a specific service provider:** This use case enables a remote system to request the description of a service provider in the eInfraCentral catalogue based on its ID. It provides as input the ID of the provider, validates the input request and returns the description of the provider.



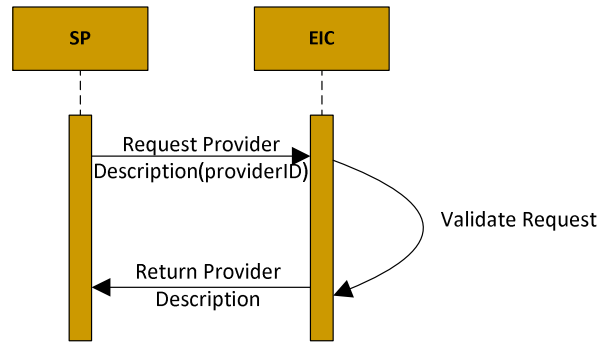


Figure 12: UC2.1 – Get information about a specific service provider

**UC2.2. Get the list of service providers:** This use case enables a remote system to request a list of all service providers in the eInfraCentral catalogue. It performs an empty request and returns the list of all service providers in the eInfraCentral catalogue.

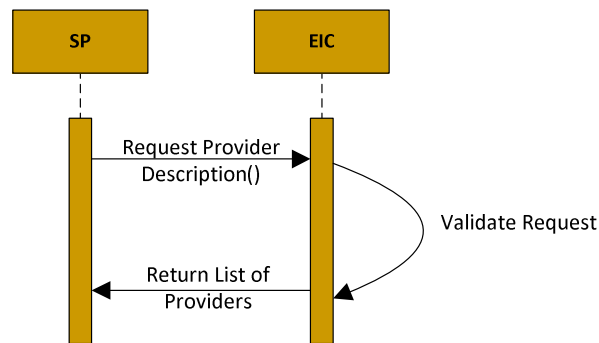


Figure 13: UC2.2 – Get the list of service providers

### 2.1.3 Use Cases for retrieving indicators and analytics

The following list of use cases concerns retrieving usage statistics, i.e. analytics for a service or a service provider collected by the eInfraCentral Gateway. They also concern methods for retrieving information regarding indicators provided by a service provider for their offerings.

**UC3.1 Get usages statistics for a service:** This use case enables a remote system to request and retrieve usage statistics, commonly referred as analytics, for a service. As analytics, the use case considers a number of user visits to the service page in the eInfraCentral Gateway, a number of service orders from the service provider’s page, a number that a service is added to the list of favourites of users and an average rating of a service. For each type, it provides the service ID and retrieves daily results (e.g. visits per day).

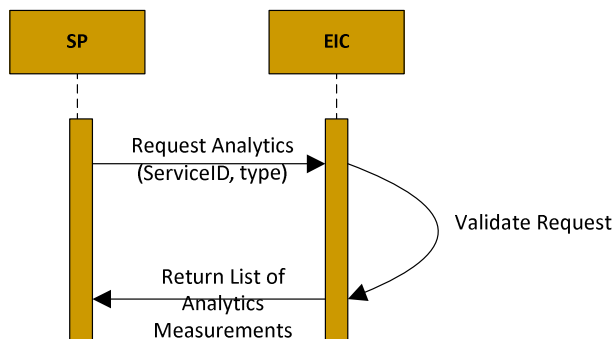


Figure 14: UC3.1 – Get analytics for a service

**UC3.2 Get analytics for a service provider:** Similarly to the above, the request provides the provider ID and retrieves the aforementioned analytics referring to aggregated values for all services in the provider catalogue. This use case retrieves an additional statistic, concerning the percentage of visits for all services offered by the provider.

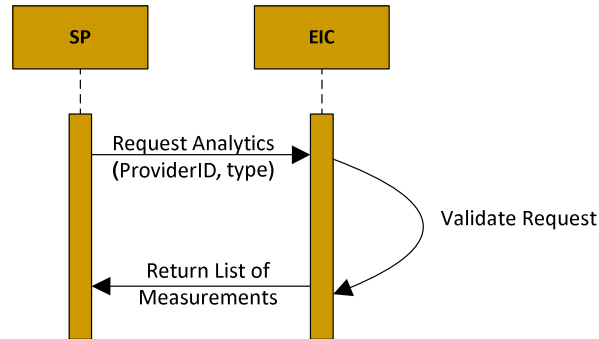


Figure 15: UC3.2 – Get analytics for a service provider

**UC3.3. Get a list of all indicators associated with a service:** This use case enables a remote system to request all indicators associated with a service in the eInfraCentral catalogue. It provides as an input the service ID and returns the list of indicators descriptions.

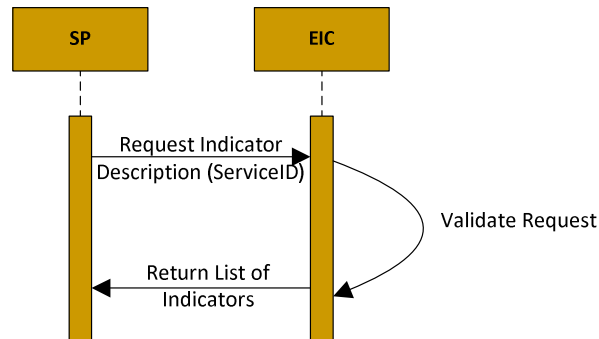


Figure 16: UC3.3 – Get a list of all indicators associated with a service

**UC3.4. Get the values of an indicator of a service:** This use case enables a remote system to request the values of a specific indicator associated with a service in the eInfraCentral catalogue. It provides as an input the service ID and the indicator ID and returns the values of the indicator for this service.

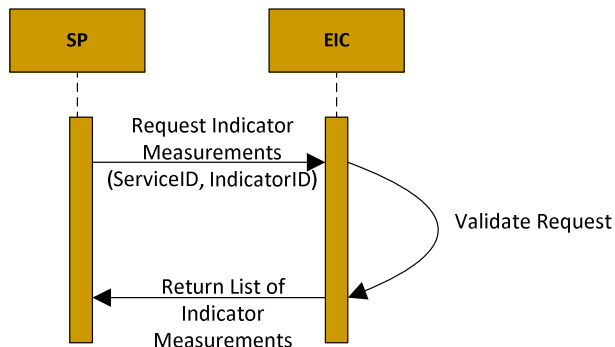


Figure 17: UC3.4 – Get the values of an indicator of a service

## 2.1.4 Use Cases for retrieving vocabulary-related information

**UC4.1. Get a list of service categories:** This use case enables a remote system to request for the list of the enumerated attributes used in the service description. It performs a request providing the name of the attribute (e.g. category, life cycle status, etc.) and returns a list of values allowed by eInfraCentral for this attribute.

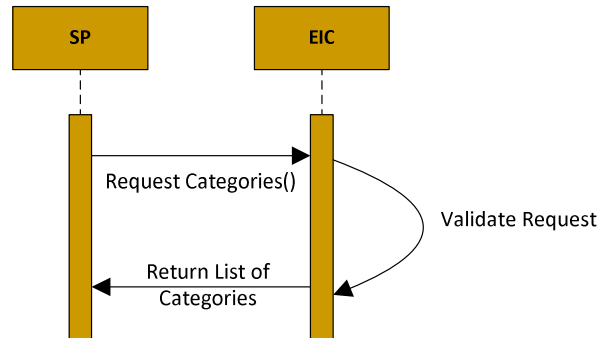


Figure 18: UC10 – Get a list of service categories

## 2.1.5 Use Cases for retrieving funder-related information

**UC5.1. Get information about a specific service funder:** This use case enables a remote system to request the description of a service funder in the eInfraCentral catalogue based on its ID. It provides as input the ID of the funder, validates the input request and returns the description of the funder along with the list of service ids funded by this funder.

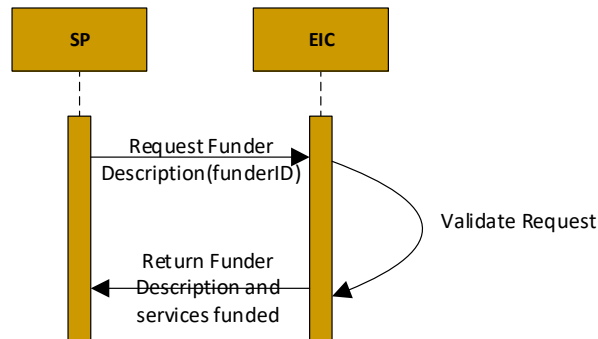


Figure 19: UC5.1 – Get information about a specific funder

**UC5.2. Get the list of service funders:** This use case enables a remote system to request a list of all service funders in the eInfraCentral catalogue. It performs an empty request and returns the list of all service funders in the eInfraCentral catalogue along with the list of service ids funded by each funder.

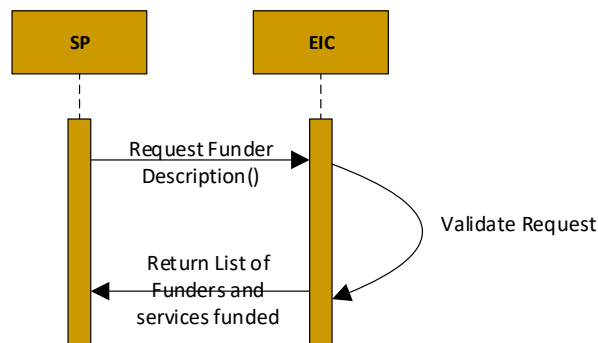


Figure 20: UC5.2 – Get the list of funders

**UC5.3. Get statistics about services funded by a specific funder:** This use case enables a remote system to request statistics related the classification of all services funded by a specific funder in the eInfraCentral catalogue, such as the category \ sub category, TRL, Lifecycle status, providers, languages and places. These statistics are visualized in the [funder dashboard](#) of the eInfraCentral. It provides as input the ID of the funder, validates the input request and returns a list of statistics per classification attribute for the services associated with the funder.

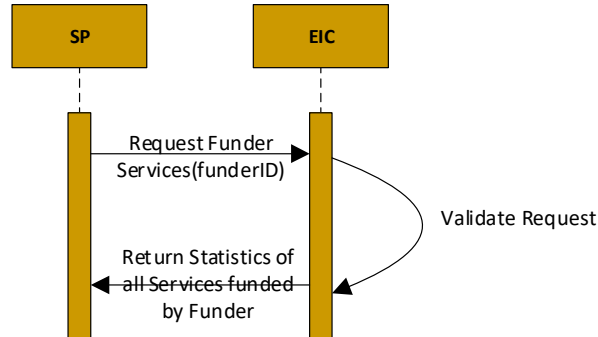


Figure 21: UC5.2 – Get statistics about services funded by a specific funder

## 2.2 Collection of eInfraCentral service information

The automatic update of a service in the eInfraCentral Gateway can be performed either via a PUSH method (where the service manager is responsible for publishing information in the eInfraCentral Gateway) or a PULL method (where the eInfraCentral Gateway harvests new updates from a remote endpoint offered by the service manager).

More specifically, the two options for the update of services are:

- A service provider posts updated information in the eInfraCentral Gateway in an ad-hoc manner through the dedicated update-API of the eInfraCentral Gateway or;
- A service provider updates its own catalogue and automatically synchronizes all changes in the eInfraCentral catalogue. EInfraCentral collects, validates and integrates this information in the eInfraCentral Gateway at a regular basis.

The first option covers arbitrary updates, in which a service provider wishes to explicitly update information in the catalogue (e.g. a new version of a service is available), whereas the second option covers updates usually performed at a regular basis, e.g. updates in the performance indicators of a service, which are published e.g. every day or on other intervals defined by the service provider according to their internal management practices.

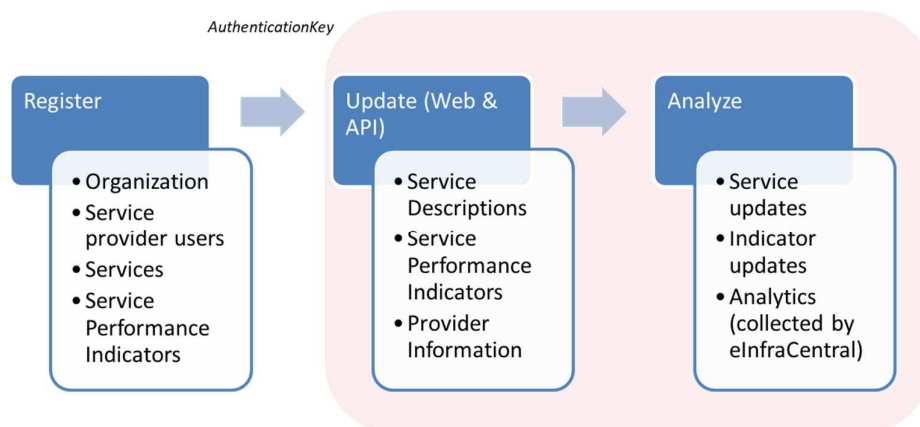


Figure 22: Workflow for the update of eInfraCentral information by a service provider

A prerequisite for a service provider to be able to use the API and update a resource in the eInfraCentral is to register its organisation in the eInfraCentral and receive a unique ID, as well as an authentication token, which can be used in all API calls.

An abstract flow of the actions that should be followed by a service provider to get access to the functionality of the monitoring services is shown in Figure 22. For completeness reasons, in this section, we present and refine these API calls according to the latest API version.

**UC6.1. Update of information of an existing service offering via a pull method:** This use case requires a service provider/manager to publish a list of services along with their descriptions in a public endpoint, accessible via an API. Upon registration, a new service manager registers the remote endpoint, where eInfraCentral has access and retrieves an initial list of services. In case of a service update, the service manager updates a service in its catalogue and makes a new service description (in the form of a service resource object, presented in section 3) available at the registered endpoint (identified by a URL). The update action can be optionally notified to the eInfraCentral. The eInfraCentral pulls (on request or at regular time intervals) all service descriptions and performance indicator measurements from the endpoint based on a timestamp attribute, which denotes the last update time of a service. It determines the new information, validates and updates the catalogue for this provider. All actions are recorded in the update history of each service updated in the eInfraCentral.

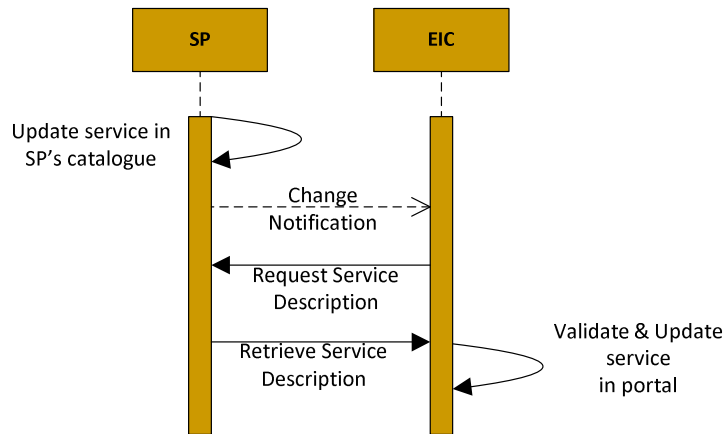


Figure 23: UC6.1 – Update of information of an existing service via a PULL method

**UC6.2. Update of information of an existing service via a push method:** A service provider updates a service in its catalogue. The new service description is posted to the eInfraCentral endpoint, which in turn validates and updates the service description in the catalogue. All actions are recorded in the update history of the service. An update on a service may result in a new version for this service in case the version identifier of that service has been changed in the service description.

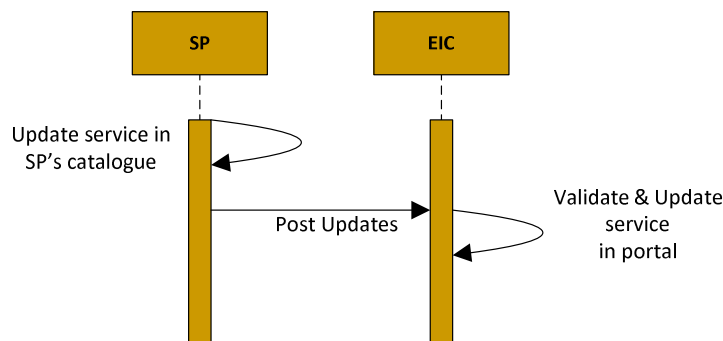


Figure 24: UC6.2 – Update of information of an existing service via a PUSH method

**UC6.3. Validate an update of a service via a push method:** This use case enables a service provider to validate the service description before updating it in the catalogue. elnraCentral returns a success or a message with the error of the validation.

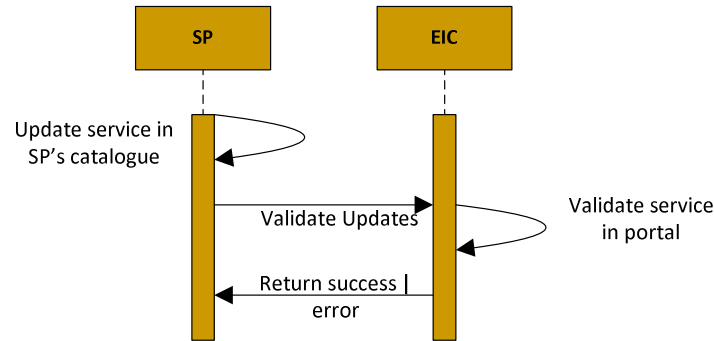


Figure 25: UC6.3 – Validate an update of a service via a push method

**UC6.4. Update of information of a service provider via a push method:** A service provider may update the information of its organisation in the catalogue, e.g. modify the contact email, etc. Again the new provider information is posted to the elnraCentral, which in turn validates and updates the description in the catalogue.

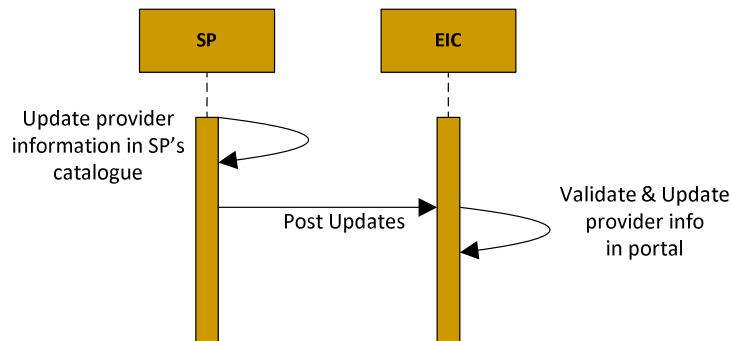


Figure 26: UC6.4 – Update of information of a service provider via a push method

**UC6.5. Addition of a new measurement for a service performance indicator via a push method:** A service provider may add a new measurement concerning a service performance indicator in the catalogue. Again the new measurement is posted to the elnraCentral providing the indicator ID, the service ID and the values (e.g. the reference time period and the reported value), for the indicator, which in turn validates and updates the catalogue.

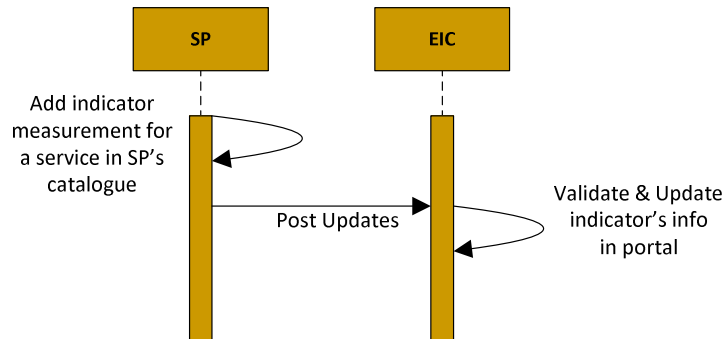


Figure 27: UC6.5 – Addition of a new measurement for a performance indicator via a push method

### 3 Representation of eInfraCentral Resources

#### 3.1 The eInfraCentral Resource Model

The primary goal of the eInfraCentral Gateway is to offer users the ability to access and browse a unified catalogue of services offered by e-Infrastructures, i.e. service providers. D4.1. provided an initial description of the resource model, focusing on the attributes of the main entity offered within the eInfraCentral Gateway, i.e. a service. D4.2 refined this model adding the concept of service versioning, the concept of indicators as well as the concept of the service manager-aggregator who can manage services on behalf of multiple service providers (see also Figure 1). Finally, D3.2 has presented the last version of the service description template with details of the service characteristics. Based on these, this section provides a résumé of these concepts and reports on the latest version of the eInfraCentral Resource model as shown in the form of a UML diagram in Figure 28.

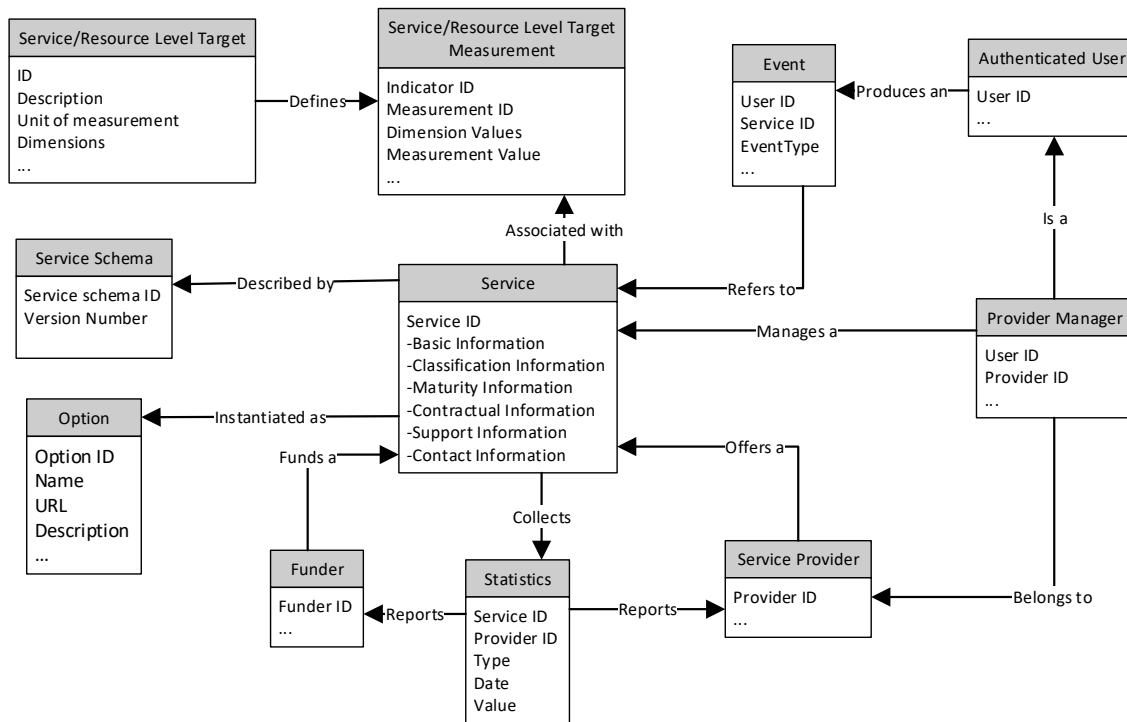


Figure 28: An overview of the eInfraCentral Gateway resource model

In essence, the main resource within eInfraCentral is *the service*. Within the eInfraCentral Gateway a service is identified by a persistent unique ID, which is generated by the eInfraCentral during service registration. Furthermore, a service is described by a set of attributes, according to the latest service description template (v2.0). A service resource should comply with a service schema. Finally, a service is offered in different options.

A service resource is associated with one or more Service Level Targets and Performance indicators, which are used for defining indicator measurements. An indicator measurement holds the measurement value for the reference dimension values, e.g. a time or a reference location, and refers to a specific service.

A service is offered by a service provider, identified by a provider ID. A service is also associated with one or more funders, which have funded the release of the service.

An authenticated user is a user who can login in the eInfraCentral Gateway and generate events for a service, such as rate a service, add a service to the favourites, update a service, etc. An

authenticated user may belong to a service provider meaning they would be authorised to manage and monitor the services of that service provider.

Finally, a set of statistics are collected for a service from the user events. These statistics are reported to the provider or the funder.

In Appendix 1: Detailed Description of the eInfraCentral resource model, we provide the details of the schema of each of the aforementioned resources as it is modelled in the eInfraCentral catalogue and used in the APIs. For the main resource, i.e. the service resource, we follow the Service Description Template v2.0 which is publicly available in <https://github.com/eInfraCentral/docs>.

Sample representations of the resources used by the eInfraCentral Gateway for the exchange of information through the APIs are presented as JSON objects (Json Schema and Example JSON) in the following tables.

Table 1: The service resource represented in JSON

#### Service Resource Schema

```
{
  "title": "Service",
  "properties": {
    "accessModes": {
      "items": {
        "type": "string"
      },
      "type": "array"
    },
    "accessPolicy": {
      "type": "string"
    },
    "accessTypes": {
      "items": {
        "type": "string"
      },
      "type": "array"
    },
    "category": {
      "type": "string"
    },
    "certifications": {
      "items": {
        "type": "string"
      },
      "type": "array"
    },
    "changeLog": {
      "type": "string"
    },
    "description": {
      "type": "string"
    },
    "funders": {
      "items": {
        "type": "string"
      },
      "type": "array"
    },
    "helpdesk": {
      "type": "string"
    },
    "id": {
```



```

        "type": "string"
    },
    "languages": {
        "items": {
            "type": "string"
        },
        "type": "array"
    },
    "lastUpdate": {
        "type": [ "string", "number" ]
    },
    "logo": {
        "type": "string"
    },
    "maintenance": {
        "type": "string"
    },
    "manual": {
        "type": "string"
    },
    "monitoring": {
        "type": "string"
    },
    "multimediaUrls": {
        "items": {
            "type": "string"
        },
        "type": "array"
    },
    "name": {
        "type": "string"
    },
    "options": {
        "items": {
            "properties": {
                "description": {
                    "type": "string"
                },
                "id": {
                    "type": "string"
                },
                "logo": {
                    "type": "string"
                },
                "name": {
                    "type": "string"
                },
                "url": {
                    "type": "string"
                }
            },
            "required": [
                "id",
                "name",
                "description",
                "url"
            ],
            "type": "object"
        },
        "type": "array"
    },
    "order": {

```

```

        "type": "string"
    },
    "orderType": {
        "type": "string"
    },
    "ownerContact": {
        "type": "string"
    },
    "ownerName": {
        "type": "string"
    },
    "paymentModel": {
        "type": "string"
    },
    "phase": {
        "type": "string"
    },
    "places": {
        "items": {
            "type": "string"
        },
        "type": "array"
    },
    "pricing": {
        "type": "string"
    },
    "privacyPolicy": {
        "type": "string"
    },
    "providers": {
        "items": {
            "type": "string"
        },
        "type": "array"
    },
    "relatedServices": {
        "items": {
            "type": "string"
        },
        "type": "array"
    },
    "requiredServices": {
        "items": {
            "type": "string"
        },
        "type": "array"
    },
    "scientificDomains": {
        "items": {
            "type": "string"
        },
        "type": "array"
    },
    "scientificSubdomains": {
        "items": {
            "type": "string"
        },
        "type": "array"
    },
    "securityContact": {
        "type": "string"
    },
    },

```

```
"securityName": {
  "type": "string"
},
"sla": {
  "type": "string"
},
"standards": {
  "items": {
    "type": "string"
  },
  "type": "array"
},
"subcategories": {
  "items": {
    "type": "string"
  },
  "type": "array"
},
"supercategory": {
  "type": "string"
},
"supportContact": {
  "type": "string"
},
"supportName": {
  "type": "string"
},
>tagline": {
  "type": "string"
},
"tags": {
  "items": {
    "type": "string"
  },
  "type": "array"
},
"targetUsers": {
  "items": {
    "type": "string"
  },
  "type": "array"
},
"termsOfUse": {
  "type": "string"
},
"training": {
  "type": "string"
},
"trl": {
  "type": "string"
},
"url": {
  "type": "string"
},
"useCases": {
  "items": {
    "type": "string"
  },
  "type": "array"
},
"userBaseList": {
  "items": {
```

```

        "type": "string"
      },
      "type": "array"
    },
    "userValue": {
      "type": "string"
    },
    "version": {
      "type": "string"
    }
  },
  "required": [
    "id",
    "name",
    "url",
    "description",
    "logo",
    "providers",
    "scientificDomains",
    "scientificSubdomains",
    "category",
    "subcategories",
    "supercategory",
    "targetUsers",
    "languages",
    "places",
    "phase",
    "trl",
    "orderType"
  ]
}

```

#### Service Resource Example

```

{
  "accessModes": [
    "access_mode-other"
  ],
  "accessPolicy": "https://www.example.com/access_policy",
  "accessTypes": [
    "access_type-physical"
  ],
  "category": "category-access_physical_and_eInfrastructures-network",
  "certifications": [
    "Gold Label"
  ],
  "changeLog": "Update of laser",
  "description": "The focus of this core facility is on various light and electron microscopy methods. Research Services include: confocal laser scanning microscopy, immuno fluorescence/morphometry, live cell imaging, analytical fluorescence microscopy, construction and modification of microscopy Systems, patch-clamp, SEM, TEM, analytical TEM (electron energy loss spectroscopy), TEM tomography, cryo-preparation methods for TEM. This range of methods allows a variety of applications in basic and applied research, such as aging research, allergy, tumor research, stress physiology and plant biology.",
  "funders": [
    "funded_by-other"
  ],
  "helpdesk": "https://www.example.com/helpdesk",
  "id": "sampleProvider.sampleService",
  "languages": [
    "german",

```

```

    "english"
  ],
  "lastUpdate": "2019-05-01",
  "logo": "https://www.example.com/logo.gif",
  "maintenance": " https://www.example.com/maintenance",
  "manual": " https://www.example.com/manual",
  "monitoring": " https://www.example.com/monitoring",
  "multimediaUrls": [
    "https://www.youtube.com/watch?time_continue=22&v=Cj3LeWua9-s"
  ],
  "name": "SampleService",
  "options": [
    {
      "description": "50mbps VDSL",
      "id": "option1",
      "name": "Internet Speed",
      "url": "https://www.uni-salzburg.at/biosciences/advanced-imaging-and-
microscopy/option1"
    }
  ],
  "order": "https://www.uni-salzburg.at/biosciences/advanced-imaging-and-
microscopy/order",
  "orderType": "order_type-order_required",
  "ownerContact": "mail@example.org",
  "ownerName": "Prof. Dr. Ursula Lütz-Meindl",
  "paymentModel": " https://www.example.com/payment_model",
  "phase": "phase-production",
  "places": [
    "AT"
  ],
  "pricing": " https://www.example.com/price",
  "privacyPolicy": " https://www.example.com/privacy_policy",
  "providers": [
    "sampleProvider"
  ],
  "relatedServices": [
    "Figsharecom.figshare"
  ],
  "requiredServices": [
    "Figsharecom.figshare"
  ],
  "scientificDomains": [
    "scientific_domain-natural_sciences"
  ],
  "scientificSubdomains": [
    "scientific_subdomain-natural_sciences-biological_sciences"
  ],
  "securityContact": "my.user@example.com",
  "securityName": "Security Name",
  "sla": " https://www.example.com/sla",
  "standards": [
    "ISO 15430"
  ],
  "subcategories": [
    "subcategory-access_physical_and_eInfrastructures-network-traffic_manager"
  ],
  "supercategory": "supercategory-access_physical_and_eInfrastructures",
  "supportContact": "contact@example.com",
  "supportName": "Contact Name",
  "tagline": "Light and Electron Microscopy methods",
  "tags": [
    "cell imaging",

```

```

    "microscopy",
    "spectroscopy",
    "imaging",
    "biology"
  ],
  "targetUsers": [
    "target_users-research_infrastructure_managers",
    "target_users-research_communities",
    "target_users-research_networks"
  ],
  "termsOfUse": "https://www.example.com/terms_of_use",
  "training": " https://www.example.com/training",
  "trl": "trl-9",
  "url": " https://www.example.com/advanced-imaging-and-microscopy_2820",
  "useCases": [
    "All these methods require elaborate sample preparations, for example chemical and/or physical sample fixation (cryo-fixation), dehydration, embedding and sectioning/ultrathin sectioning. Examples include a cryo-sectioning microtome, a sputtering device, a high-pressure freezer (Leica Empact), a cryo-substitution device (Leica AFS), a plunge freezer (KF80, Reichert), a cryo-jet device (Balzers), a freeze-etching device (Balzers), several ultramicrotomes (Reichert Ultracut, Ultracut E, Leica UC7) a cryo-ultramicrotome, trimmers (Leica EM Trim), polymerization ovens and numerous diamond knives."
  ],
  "userBaselist": [
    "More than 150 Researchers since 2014"
  ],
  "userValue": "For contrast enhancement, the samples can be labelled by chemical or immunological methods or also by in situ hybridization. Electron microscopic methods can be used to study biological structures from molecular to cellular and tissue level. In light microscopy, in particular fluorescence and confocal laser scanning microscopy (CLSM), investigation of both, fixed biological samples and living cells is possible. Therefore, in the Core Facility is also equipped with safety cabinets that enable live cell imaging, toxicological evaluations and further bioanalytical investigations. The measuring stations allow complete electrophysiological characterization of cells up to single channel resolution. For the further investigation of living cells and organoids in 3D, bioreactors as well as modern microfluidic analysis systems are available, thus offering the possibility of a flow living cell microscopy. The facility also has the prerequisites for producing high-quality biochips in small quantities in very fast production cycles. In particular, there is a rapid prototyping facility with a) precision 3D printer b) polymer milling machine c) hot embossing facility d) sputtering facility e) plasma cleaner f) spin coater.",
  "version": "v1.0"
}

```

Table 2: The options resource represented in JSON

## Option Resource Schema

```

{
  "title": "Options",
  "properties": {
    "description": {
      "type": "string"
    },
    "id": {
      "type": "string"
    },
    "logo": {
      "type": "string"
    }
  }
}

```

|   |
|---|
| <pre>"name": {   "type": "string" }, "url": {   "type": "string" } }, "required": [   "id",   "name",   "description",   "url" ] }</pre>  |
| Option Resource Example   |
| <pre>{   "description": "50mbps VDSL",   "id": "option1",   "logo": "https://www.uni-salzburg.at/biosciences/advanced-imaging-and- microscopy/optionLogo1.png",   "name": "Internet Speed",   "url": "https://www.uni-salzburg.at/biosciences/advanced-imaging-and- microscopy/option1" }</pre> |

Table 3: The service provider resource represented in JSON

|   |
|---|
| Service Provider Resource Schema  |
| <pre>{   "title": "Provider",   "properties": {     "contactEmail": {       "type": "string"     },     "contactName": {       "type": "string"     },     "contactTel": {       "type": "string"     },     "description": {       "type": "string"     },     "id": {       "type": "string"     },     "logo": {       "type": "string"     },     "name": {       "type": "string"     },     "users": {       "items": {         "properties": {           "email": {             "type": "string"           },           "id": {             "type": "string"           }         }       }     }   } }</pre> |

|   |
|---|
| <pre>        "name": {           "type": "string"         },         "surname": {           "type": "string"         }       },       "required": [         "email"       ],       "type": "object"     },     "type": "array"   },   "website": {     "type": "string"   } }, "required": [   "id",   "name",   "website",   "description",   "logo",   "contactName",   "contactEmail",   "contactTel" ] }</pre>  |
| <b>Service Provider Resource Example</b>  |
| <pre>{   "contactEmail": "email1@gmail.com",   "contactName": "John",   "contactTel": "+302111111094",   "description": "Av example for a new Provider.",   "id": " sampleProvider",   "logo": " https://www.example.com/images?q=tbm:ANd9GcRvdbhOCz-e5F9CrLaEsFW4EySJtkDs15WAavTVXvauQZe-rjB7YQ",   "name": "Example Provider",   "users": [     {       "email": "email1@gmail.com",       "id": "jb",       "name": "John",       "surname": "Balianos"     },     {       "email": "email2@outlook.com",       "id": "mg",       "name": "Mike",       "surname": "Grigoriou"     }   ],   "website": "https://exampleprovider.com" }</pre> |

Table 4: The funder resource represented in JSON

|                                   |
|-----------------------------------|
| <b>Funder Resource Schema</b>     |
| <pre>{   "title": "Funder",</pre> |



|   |
|---|
| <pre>"properties": {   "id": {     "type": "string"   },   "logo": {     "type": "string"   },   "name": {     "type": "string"   },   "services": {     "items": {       "type": "string"     },     "type": "array"   } }, "required": [   "id",   "name",   "logo",   "services" ] }</pre> |
| <b>Funder Resource Example</b>  |
| <pre>{   "id": "ec",   "logo": "https://ec-europe.org/logo.png",   "name": "European Commission",   "services": [     "catris",     "validator",     "eInfraCentral"   ] }</pre>  |

Table 5: The Service Level Target and Performance Indicator resource represented in JSON

|  |
|--|
| <b>Indicator Resource</b>  |
| <pre>{   "title": "Indicator",   "properties": {     "description": {       "type": "string"     },     "dimensions": {       "items": {         "enum": [ "time", "locations" ]       },       "type": "array"     },     "id": {       "type": "string"     },     "name": {       "type": "string"     },     "unit": {       "enum": [ "percentage", "numeric", "boolean" ]     },     "unitName": {</pre> |

|  |
|--|
| <pre>    "type": "string"   } }, "required": [   "id",   "name",   "description",   "dimensions",   "unit",   "unitName" ] }</pre>   |
| <b>Indicator Resource Example for Service Availability Indicator</b>   |
| <pre>{   "description": "A Service Reliability, i.e. the probability that an item will function without failure under stated conditions for a specified amount of time. “Stated conditions” indicate prerequisite conditions external to the item being considered. For example, a stated condition for a supercomputer might be that power and cooling must be available - thus a failure of the power or cooling systems would not be considered a failure of the supercomputer.",   "dimensions": [     "time"   ],   "id": "reliability",   "name": "Reliability",   "unit": "percentage",   "unitName": "%" }</pre> |
| Where Unit of measurement takes a value from (Percentage, Numeric, Boolean) and Dimensions can be TIME and/or LOCATIONS  |

Table 6: The Indicator Measurement resource represented in JSON

|   |
|---|
| <b>Indicator Measurement Resource</b>   |
| <pre>{   "title": "Measurement",   "properties": {     "id": {       "type": "string"     },     "indicatorId": {       "type": "string"     },     "locations": {       "items": {         "type": "string"       },       "type": "array"     },     "rangeValue": {       "properties": {         "fromValue": {           "type": "string"         },         "toValue": {           "type": "string"         }       }     },     "required": [       "fromValue",       "toValue"     ]   } }</pre> |

|   |
|---|
| <pre>    ],     "type": [ "object" ]   },   "serviceId": {     "type": "string"   },   "time": {     "type": [ "string", "number" ]   },   "value": {     "type": [ "string" ]   },   "valueIsRange": {     "type": "boolean"   } }, "if": {   "properties": {     "valueIsRange": {       "const": false     }   } }, "then": {   "required": [ "value" ],   "not": { "required": [ "rangeValue" ] } }, "else": {   "required": [ "rangeValue" ],   "not": { "required": [ "value" ] } }, "required": [   "id",   "indicatorId",   "serviceId",   "valueIsRange" ] ]</pre> |
| Measurement Resource Example for Service Availability Indicator   |
| <pre>{   "id": "measurementId",   "indicatorId": "deliverytime",   "locations": [ "EU" ],   "rangeValue": {     "fromValue": "10",     "toValue": "20"   },   "serviceId": "prace.prace_project_access",   "time": "1992-03-03",   "valueIsRange": true }</pre>   |

Table 7: The statistics resource represented in JSON

|  |
|--|
| Statistics Resource  |
| <pre>{   "title": "StatisticsResponse",   "properties": {     "stats": {       "type": "array",       "items": [</pre> |

|  |
|--|
| <pre>{   "type": "object",   "properties": {     "date": {       "type": "string",       "format": "date-time"     },     "value": {       "type": "number"     }   },   "required": [     "date",     "value"   ] }</pre> |
| Statistics Resource Example for the number of visits for a service   |
| <pre>{   "stats": [     {       "date": "2019-03-01",       "value": 532     },     {       "date": "2019-04-01",       "value": 584     }   ] }</pre>   |

Table 8: The vocabulary resource represented in JSON

|  |
|--|
| Vocabulary Resource  |
| <pre>{   "title": "Vocabulary",   "properties": {     "description": {       "type": "string"     },     "id": {       "type": "string"     },     "name": {       "type": "string"     },     "parentId": {       "type": "string"     },     "type": {       "enum": [         "SUPERCATEGORY",         "CATEGORY",         "SUBCATEGORY",         "LANGUAGE",         "PLACE", </pre> |

```

        "TRL",
        "PHASE",
        "SCIENTIFIC_DOMAIN",
        "SCIENTIFIC_SUBDOMAIN",
        "TARGET_USERS",
        "ACCESS_TYPE",
        "ACCESS_MODE",
        "ORDER_TYPE"
    ]
  },
  "required": [
    "id",
    "name",
    "type"
  ]
}

```

#### Vocabulary Resource Example

```

{
  "description": "Services for monitoring, scaling, creating, tracking and automating operations on infrastructures and services.",
  "id": "category-security_and_operations-operations_and_infrastructure_management_services",
  "name": "Operations & Infrastructure Management Services",
  "parentId": "supercategory-security_and_operations",
  "type": "CATEGORY"
}

```

## 3.2 elnraCentral Resource Vocabularies

The elnraCentral service resource makes use of controlled vocabularies (e.g. list of values) for the following set of attributes. The allowed values used for each attribute are presented below:

- **Supercategory \ Category \ Subcategory**
  - **Access physical & elnrastructures**
    - Instrument & Equipment
      - Spectrometer
      - Radiation
      - Microscopy
      - Laser
      - Geophysical
      - Chromatographer
      - Cytometer
      - Spectrophotometer
      - Digitisation Equipment
      - Monument Maintenance Equipment
      - Other
    - Network
      - Direct Connect
      - Virtual Network
      - Load Balancer
      - VPN Gateway
      - Exchange

- Content Delivery Network
  - Traffic Manager
  - Other
- Compute
  - Virtual Machine Management
  - Container Management
  - Job Execution
  - Workload Management
  - Orchestration
  - Serverless Applications Repository
  - Other
- Data Storage
  - Data
  - File
  - Queue
  - Disk
  - Online
  - Archive
  - Backup
  - Synchronised
  - Replicated
  - Recovery
  - Digital Preservation
  - Other
- Material Storage
  - Warehousing
  - Fulfilment
  - Assembly
  - Sorting
  - Reworking
  - Packaging
  - Quality Inspecting
  - Archiving
  - Disposal
  - Repository
  - Preservation
  - Other
- **Sharing & Discovery**
  - Data
    - Government & Agency Data
    - Statistical Data
    - Scientific Data
    - Online Service Data
    - Clinical Trial Data
    - Epidemiological Data
    - Data Archives

- Other
- Scholarly Communication
  - Preparation
  - Discovery
  - Analysis
  - Writing
  - Publication
  - Outreach
  - Assessment
  - Other
- Software
  - Software Repository
  - Platform
  - Software Package
  - Libraries
  - Other
- Applications
  - Communication
  - Collaboration
  - Productivity
  - Business
  - Education
  - Social
  - Utilities
  - Applications Repository
  - Other
- Development Resources
  - Developer Tools
  - Software Development Kits
  - Software Libraries
  - APIs Repository
  - Other
- Samples
  - Biological Samples
  - Chemical Compounds Library
  - Preparation
  - Characterisation
  - Other
- **Processing & Analysis**
  - Data Management
    - Mining
    - Access
    - Transfer
    - Registration
    - Persistent Identifier
    - Interlinking

- Publishing
  - Discovery
  - Anonymisation
  - Preservation
  - Brokering
  - Annotation
  - Validation
  - Maintenance
  - Embargo
  - Digitisation
  - Other
- Data Analysis
  - Machine Learning
  - Artificial Intelligence
  - Forecast
  - Visualization
  - Data Exploitation
  - Image Analysis
  - Workflows
  - 2D/3D Digitisation
  - Other
- Measurement & Materials Analysis
  - Analysis
  - Maintenance & Modification
  - Production
  - Testing & Validation
  - Characterisation
  - Validation
  - Workflows
  - Other
- **Security & Operations**
  - Security & Identity
    - User Authentication
    - Identity & Access Management
    - Threat Protection
    - Coordination
    - Tools
    - Certification Authority
    - Single Sign-On
    - Firewall
    - Group Management
    - Other
  - Operations & Infrastructure Management Services
    - Accounting
    - Helpdesk
    - Monitoring



- Analysis
    - Configuration
    - Utilities
    - Coordination
    - Billing
    - Order Management
    - Transportation
    - Other
  - **Training & Support**
    - Education & Training
      - Online Courses
      - Open Registration Courses
      - In-House Courses
      - Training Tool
      - Training Platform
      - Other
    - Consultancy & Support
      - Consulting
      - Audit & Assessment
      - Application Porting
      - Application Scaling
      - Application Optimisation
      - Software Development
      - Software Improvement
      - Modeling & Simulation
      - Prototype Development
      - Testing
      - Certification
      - Calibration
      - Benchmarking
      - Technology Transfer
      - Methodology Development
      - Other
  - **Aggregators & Integrators**
    - Aggregators & Integrators
      - Services
      - Data
      - Applications
      - Software
      - Other
  - **Other**
    - Other
      - Other
- **Scientific domain \ scientific subdomain**
  - **Natural Sciences**

- Mathematics
- Computer Sciences
- Information Sciences
- Earth Sciences
- Biological Sciences
- Physical Sciences
- Chemical Sciences
- **Engineering & Technology**
  - Civil Engineering
  - Electrical, Electronic & Information Engineering
  - Mechanical Engineering
  - Aerospace Engineering
  - Chemical Engineering
  - Materials Engineering
  - Bioengineering & Biomedical Engineering
  - Environmental Engineering
  - Environmental Biotechnology
  - Industrial Biotechnology
  - Micro & Nanotechnology
- **Medical & Health Sciences**
  - Basic Medicine
  - Clinical Medicine
  - Health Sciences
  - Medical Biotechnology
- **Agricultural Sciences**
  - Agriculture, Forestry & Fisheries
  - Animal & Dairy Sciences
  - Veterinary Sciences
  - Agricultural Biotechnology
- Social Sciences
  - Psychology
  - Economics, Finance & Business
  - Educational Sciences
  - Sociology
  - Law
  - Political Sciences
  - Social & Economic Geography
  - Media & Communications
- Humanities
  - History & Archaeology
  - Languages & Literature
  - Philosophy, Ethics & Religion
  - Arts
- Interdisciplinary
  - Interdisciplinary
- Other
  - Other

- **Target users**
  - Researchers
  - Research Groups
  - Research Communities
  - Research Projects
  - Research Networks
  - Research Managers
  - Research Organisations
  - Innovators
  - Businesses
  - Service Providers
  - Funders
  - Policy Makers
  - Research Infrastructure Managers
  - Service Provider Managers
  - Service Managers
  - Other
- **Access type**
  - Remote
  - Physical
  - Virtual
  - Mail-In
  - Other
- **Access mode**
  - Excellence Driven
  - Market Driven
  - Policy Based
  - Wide Access
  - Other
- **Order Type**
  - Order Required
  - Open Access
  - Fully Open Access
  - Other
- **Phase**
  - In containment
  - Preparation
  - Discovery
  - Concept development
  - Production
  - Implementation
  - Planned
  - Termination
  - Alpha

- Design
- Operation
- Beta
- Retirement
- Other
- **Technology Readiness Levels**
  - TRL1 - basic principles observed
  - TRL2 - technology concept formulated
  - TRL3 - experimental proof of concept
  - TRL4 - technology validated in lab
  - TRL5 - technology validated in relevant environment
  - TRL6 - technology demonstrated in relevant environment
  - TRL7 - system prototype demonstration in operational environment
  - TRL8 - system complete and qualified
  - TRL9 - actual system proven in operational environment
- **Places:** Places refer to countries worldwide. eInfraCentral follows the ISO 639-1<sup>1</sup> list of codes for the representation of the countries and extends it with two codes: EU for Europe and WW for worldwide.
- **Languages.** eInfraCentral follows the ISO 3166-1 alpha-2 standard<sup>2</sup> for the 2-letter codes for representing the languages.

---

<sup>1</sup> <https://www.iso.org/iso-639-language-codes.html>

<sup>2</sup> <https://www.iso.org/iso-3166-country-codes.html>

## 4 The eInfraCentral API specifications

In this chapter, we provide the functionality corresponding to the Use Cases presented in Chapter 2 that will be offered by the eInfraCentral Gateway in the form of *API specifications*. The eInfraCentral API is a lightweight web service with a REST interface, which provides to remote system easy access to the eInfraCentral Service Catalogue. An API (Application Programming Interface) is a protocol intended to be used as an interface by software components to communicate with each other. The complete [eInfraCentral API documentation](http://www.einfracentral.eu/developers) is a living document, generated using frameworks compliant with OpenAPI Specification 2.0. **API v2.0** is available in <http://www.einfracentral.eu/developers>.

### 4.1 The eInfraCentral API technology stack

The eInfraCentral Gateway employs well-established web technologies, i.e. HTTP REST, for the implementation of its API methods. HTTP defines a set of request methods to indicate the desired action to be performed for a given service resource. Each of these methods (referred to as HTTP verbs) implements a different action. The primary or most-commonly-used HTTP verbs are POST, GET, PUT, PATCH, and DELETE. These correspond to create, read, update, and delete (or CRUD) operations, respectively, as shown in Table 4. The eInfraCentral Gateway API makes use of these methods for offering RESTful Web Services for the monitoring and management of the service catalogue. Representational State Transfer (REST) is an architectural style that specifies constraints, such as the uniform interface, that if applied to a web service induces desirable properties, such as performance, scalability, and modifiability, that enable services to work best on the Web. In the REST architectural style, data and functionality are considered resources and are accessed using Uniform Resource Identifiers (URIs), typically links on the Web. The resources are acted upon by using a set of simple, well-defined operations. The REST architectural style is designed to use a stateless communication protocol, typically HTTP. In the REST architecture style, clients and servers exchange representations of resources by using a standardized interface and protocol.

Table 3: HTTP verbs used in EIC API and their operations

| Operation  | Uniform API Operation | Description  |
|--|-----------------------|--|
| Query Catalogue Resources                            | GET                   | GET must be used to retrieve a representation of a service                       |
| Create of a Catalogue Resource                       | POST                  | POST must be used to create a new resource                                       |
| Update of a Catalogue Resource                       | PUT                   | PUT must be used to make a complete replacement of the resource in the catalogue |
| Apply partial modifications to a catalogue resource. | PATCH                 | PATCH must be used to apply minor modification to a resource object.             |
| Remove a Catalogue Resource                          | DELETE                | DELETE must be used to remove a resource   |

In the following sections, we provide the detailed API methods descriptions.

### 4.2 eInfraCentral API methods

The API methods are organised in controllers according to the type of information offered by each method, i.e. a controller is a set of API methods used for posting or retrieving information from the eInfraCentral Gateway. The API controllers and the functionality it offers are the following:

- **Service Controller.** It offers the functionality for managing a service resource, such as:
  - adding, i.e. registering, a new service resource in the eInfraCentral Gateway,
  - updating an existing service resource description,

- validating the registration or update of a service resource without it actually being added/modified in the registry,
- retrieving the most current version of a specific service providing the service ID,
- retrieving a past version of a specific service providing the service ID and a version identifier,
- retrieving a list of services based on a set of IDs,
- retrieving a list of all services in the eInfraCentral Catalogue,
- retrieving all services in the catalogue organised by an attribute, e.g. get service organised in categories,
- retrieving a list of services based on a set of filters,
- **Provider Controller:** It offers the functionality for managing a service provider resource, such as:
  - updating an existing provider information,
  - retrieving all service providers in the catalogue,
  - retrieving a service provider description providing the provider ID,
  - retrieving a list of services offered by a provider,
- **Funder Controller:** It offers the functionality for managing a funder resource, such as:
  - retrieving all funders in the catalogue, along with the list of services funded by each funder
  - retrieving a funder description providing the funder ID, along with the list of services funded by this funder
  - retrieving aggregated statistics for all vocabularies in the catalogue for services funded by a funder, providing the funder ID
- **Indicator Controller:** It offers the functionality for managing a KPI definition, i.e. indicator resource, such as:
  - adding, i.e. registering, a new indicator resource in the eInfraCentral Gateway,
  - updating an existing indicator resource,
  - retrieving the definition of an indicator providing the indicator ID,
- **Measurement Controller:** It offers the functionality for managing a KPI measurement, i.e. measurement resource, such as:
  - adding a new measurement for a service in the eInfraCentral Gateway,
  - updating an existing measurement,
  - retrieving the measurements of an indicator for a specific service,
- **Statistics Controller:** It offers the functionality for retrieving statistics collected in eInfraCentral Gateway, such as:
  - retrieving visits per day for a service,
  - retrieving favourites per day for a service,
  - retrieving average ratings per day for a service,
  - retrieving aggregate visits per day for all services offered by a provider,
  - retrieving aggregate favourites per day for all services offered by a provider,
  - retrieving average ratings per day for all services offered by a provider,
  - retrieving percentage of visits for all services offered by a provider,
- **User Controller.** It offers the functionality to remote systems to login and receive a JSON Web token<sup>3</sup> (JWT) for authenticating further requests in the eInfraCentral Gateway.
- **Vocabulary Controller:** It offers the functionality to retrieve the vocabularies used in the eInfraCentral Gateway.

---

<sup>3</sup> <https://jwt.io/>

An overview of the methods offered by each controller is shown in Table 9. All methods are available at the following base url [https://www.einfracentral.eu/api/service/geant.geant\\_l3vpn](https://www.einfracentral.eu/api/service/geant.geant_l3vpn)

Table 9: An overview of the eInfraCentral API methods

| Controller       | Type | Method Name              | Description  | Requires Auth Token? |
|------------------|------|--------------------------|--|----------------------|
| <b>Service</b>   | POST | /service                 | Adds, i.e. registers, a new service resource in the eInfraCentral Gateway  | YES                  |
|                  | PUT  | /service                 | Updates an existing service resource description   | YES                  |
|                  | POST | /service/validate        | Validates the registration or update of a service resource without it actually being added/modified in the registry      | YES                  |
|                  | GET  | /service/{id}            | Gets the most current version of a specific service providing the service ID   | NO                   |
|                  | GET  | /service/{id}/{version}  | Gets the specified version of a Service, providing the Service id and version.   | NO                   |
|                  | GET  | /service/all             | Filters a list of services based on a set of filters (if empty gets the list of all services in the Catalogue)           | NO                   |
|                  | GET  | /service/by/{field}      | Gets all services in the catalogue organised by a field, e.g. get service organized in categories                        | NO                   |
|                  | GET  | /service/byID/{ids}      | Gets a list of services based on a set of IDs  | NO                   |
| <b>Provider</b>  | PUT  | /provider                | Updates provider info  | YES                  |
|                  | GET  | /provider/all            | Filters a list of service providers based on a set of filters (if empty gets the list of all providers in the Catalogue) | NO                   |
|                  | GET  | /provider/{id}           | Gets provider's data given the provider id   | NO                   |
|                  | GET  | /provider/services/{id}  | Gets a list of services offered by a provider  | NO                   |
| <b>Funder</b>    | GET  | /funder/all              | Filters a list of Funders based on a set of filters (if empty gets the list of all funders in the Catalogue)             | NO                   |
|                  | GET  | /funder/{id}             | Gets the Funder with the given id.   | NO                   |
|                  | GET  | /funder/funderStats/{id} | Gets various stats about a specific Funder   | NO                   |
| <b>Indicator</b> | POST | /indicator               | Adds the given indicator   | YES                  |
|                  | PUT  | /indicator               | Updates an existing indicator with the given ID  | YES                  |
|                  | GET  | /indicator/{id}          | Gets the indicator given its ID  | NO                   |
|                  | GET  | /indicator/all           | Filters a list of Indicators based on a set of filters (if empty gets the list of all indicators in the                  | NO                   |

|                    |      |                                  |  |     |
|--------------------|------|----------------------------------|--|-----|
|                    |      |                                  | Catalogue)   |     |
| <b>Measurement</b> | POST | /measurement                     | Adds the given indicator measurement   | YES |
|                    | PUT  | /measurement                     | Updates an existing measurement with the given ID  | YES |
|                    | GET  | /measurement/{id}                | Gets the measurement given its id  | NO  |
|                    | GET  | /measurement/latest/service/{id} | Gets the latest Measurements for the specific service.   | NO  |
|                    | GET  | /measurement/service/{id}        | Gets all measurements of the service given the service ID  | NO  |
| <b>Statistics</b>  | GET  | /stats/provider/favourites/{id}  | Gets aggregate favourites per day for all services offered by a provider   | NO  |
|                    | GET  | /stats/provider/ratings/{id}     | Gets average ratings per day for all services offered by a provider  | NO  |
|                    | GET  | /stats/provider/visitation/{id}  | Gets percentage of visits for all services offered by a provider   | NO  |
|                    | GET  | /stats/provider/visits/{id}      | Gets aggregate visits per day for all services offered by a provider   | NO  |
|                    | GET  | /stats/service/favourites/{id}   | Gets favourites per day for a service  | NO  |
|                    | GET  | /stats/service/ratings/{id}      | Gets average ratings per day for a service   | NO  |
|                    | GET  | /stats/service/visits/{id}       | Gets visits per day for a service  | NO  |
| <b>Vocabulary</b>  | GET  | /vocabulary                      | Returns the entries of a given Vocabulary type   | NO  |
|                    | GET  | /vocabulary/all                  | Filters the list of Vocabularies based on a set of filters (if empty gets the list of all vocabularies in the Catalogue) | NO  |
|                    | GET  | /vocabulary/countries/EU         | Gets the list of countries included in the EU list of the catalogue  | NO  |
|                    | GET  | /vocabulary/countries/WW         | Gets the list of countries included in the Worldwide list of the catalogue   | NO  |

In the next sections we present in details each API method in the form of API request and response messages. Each method is presented in a separate table, containing the request string and a sample response object.

#### 4.2.1 Service Controller

The following API methods manage information about a service. Based on the input parameters, they are used to get a specific service or a list of service resources, for a given service provider or for the full catalogue.

Table 10: POST /Service – Adds a new service

|   |
|---|
| <b>Request</b>  |
| POST: /service  |
| Accept: application/json  |
| Cookie: {auth-string}   |
| <b>Parameters (The full service description of the service to be added)</b> |



```

service: {
  "id": "string",
  "url": "string",
  "name": "string",
  "tagline": "string",
  "description": "string",
  ...
}

```

**Response**

Status: 200

Content-Type: application/json

**Response Messages**

| HTTP Status Code | Reason       |
|------------------|--------------|
| 201              | Created      |
| 401              | Unauthorized |
| 403              | Forbidden    |
| 404              | Not Found    |
| 500              | Other        |

**Table 11: POST /Service/validate – Validates the service without actually changing the catalogue**

Request

POST: /service/validate

Accept: application/json

Cookie: {auth-string}

Parameters (The full service description of the service to be validated)

```
service: {  
  "id": "string",  
  "url": "string",  
  "name": "string",  
  "tagline": "string",  
  "description": "string",,  
  ...  
}
```

Response

Status: 200

Content-Type: application/json

Response Messages

| HTTP Status Code | Reason            |
|------------------|-------------------|
| 201              | Valid description |
| 401              | Unauthorized      |
| 403              | Forbidden         |
| 404              | Not Found         |
| 500              | Other             |

**Table 12: PUT /Service – Updates a service**

|  |
|--|
| <b>Request</b>   |
| PUT: /service<br>Accept: application/json<br>Cookie: {auth-string}   |
| <b>Parameters (The full service description of the service to be updated)</b>  |
| <pre> service: {   "id": "string",   "url": "string",   "name": "string",   "tagline": "string",   "description": "string",   ... } </pre> |
| <b>Response</b>  |
| Status: 200  |

| Content-Type: application/json |              |
|--------------------------------|--------------|
| <b>Response Messages</b>       |              |
| HTTP Status Code               | Reason       |
| 201                            | Updated      |
| 401                            | Unauthorized |
| 403                            | Forbidden    |
| 404                            | Not Found    |
| 500                            | Other        |

Table 13: GET /service/{id} – Gets a service based on the service ID

| <b>Request</b>  |              |
|---|--------------|
| GET: /service/{id}  |              |
| <b>Parameters</b>   |              |
| id: the service id<br>e.g., /service/geant.geant_l3vpn  |              |
| <b>Response</b>   |              |
| Status: 200<br>Content-Type: application/json   |              |
| <b>Response Body (A service description)</b>  |              |
| <pre>{   "id": " geant.geant_l3vpn",   "url": "https: //www.geant.org/Services/_ Connectivity_and_network/Pages/VPN_Services.aspx",   "name": "GÉANT L3VPN",   "tagline": "Increased privacy and control - helping to build effective virtual teams across borders.",   ... }</pre> |              |
| <b>Response Messages</b>  |              |
| HTTP Status Code  | Reason       |
| 400   | No Content   |
| 401   | Unauthorized |
| 403   | Forbidden    |
| 404   | Not Found    |
| 500   | Other        |

Table 14: GET /service/all – Filters a list of services based on a set of filters or get a list of all services in the elnraCentral Catalogue

|  |  |
|--|--|
| <b>Request</b>   |  |
| GET: /service/all  |  |
| <b>Parameters</b>  |  |
| Query: An expression with filters<br>From: Used for paging the results. It indicates the nth result.<br>Quantity: Number of results to fetch<br>OrderField: The name of the field for ordering the results<br>Order: Order type (Asc Desc)<br>E.g.,<br>Get all services in the catalogue → /service/all/<br>Get the first 5 services of the compute category ordered by trl ascending →<br>/service/all/query=compute&from=0&quantity=5&orderfield=trl&order=asc |  |
| <b>Response</b>  |  |
| Status: 200<br>Content-Type: application/json  |  |
| <b>Response Body (A list of service descriptions)</b>  |  |
| <pre>{   "total": 9,   "from": 0,   "to": 5,   "results": [</pre>  |  |

```
{
  "id": "egi.egi_cloud_compute",
  "url": "https://www.egi.eu/services/cloud-compute",
  "name": "EGI Cloud Compute",
  "tagline": "Run virtual machines on-demand with complete control over computing resources",
  ...
}
```

**Response Messages**

| HTTP Status Code | Reason       |
|------------------|--------------|
| 400              | No Content   |
| 401              | Unauthorized |
| 403              | Forbidden    |
| 404              | Not Found    |
| 500              | Other        |

**Table 15: GET /service/by/{field} – Gets all services in the catalogue organized by a field**

Request

GET: /service/by/{field}

Parameters

Field: The name of an attribute used to group the results

E.g.,

Get all services in the catalogue organized in categories → /service/by/category

Response

Status: 200

Content-Type: application/json

Response Body (A list of service descriptions)

{

"Text Mining, Data Mining, Information Extraction": [

{

"id": "openaire.openaire\_inference\_",

"url": "http://mining.openaire.eu",

"name": "OpenAIRE Inference ",

>tagline": "Text and data mining for scholarly communication",

"description": "This service performs text mining (entity resolution) on the metadata and the fulltext of publications and extracts information on: links to projects/grants and funders, data citations or links to entities e(.g., links to PDB - Protein Data Bank), software citations, author affiliation, references. Furthermore it classifies publications according to several taxonomies. Results of this service are presented in the OpenAIRE portal.",

"options": null,

"targetUsers": "all scholarly communication stakeholders",

"userValue": "Improved linked open science. Improved research analytics. Improved research monitoring and impact assessment. Customers get structured metadata related to the publications. Funders have access to a list of publications that acknowledge their projects. Research may link their research results. Data providers (Repository managers/ OA publishers) may enrich their content.",

"userBase": "The EC is using the current mechanism for monitoring the open access policy. ",

"symbol": null,

"multimediaURL": null,

...

]

}

Response Messages

| HTTP Status Code | Reason       |
|------------------|--------------|
| 400              | No Content   |
| 401              | Unauthorized |
| 403              | Forbidden    |
| 404              | Not Found    |
| 500              | Other        |

Table 16: GET /service/byID/{ids} – Gets a list of services based on a set of IDs

| Request  |              |
|--|--------------|
| GET: /service/byID/{ids}   |              |
| Parameters   |              |
| ids: Comma separated service ids<br>E.g.,  |              |
| <ul style="list-style-type: none"> <li>Get services egi.egi_high-throughput_compute and openaire.openaire_scholexplorer → /service/byID/egi.egi_high-throughput_compute, openaire.openaire_scholexplorer</li> </ul>  |              |
| Response   |              |
| Status: 200  |              |
| Content-Type: application/json   |              |
| Response Body (A list of service descriptions)   |              |
| <pre>[   {     "id": "egi.egi_high-throughput_compute",     "url": "https://www.egi.eu/services/high-throughput-compute",     "name": "EGI High-Throughput Compute",     "tagline": "Execute thousands of computational tasks to analyse large datasets",     ...   }   {     "id": "openaire.openaire_scholexplorer",     "url": "http://scholexplorer.openaire.eu",     "name": "OpenAIRE ScholeXplorer",     "tagline": "The data and literature interlinking service",     ...   } ]</pre> |              |
| Response Messages  |              |
| HTTP Status Code   | Reason       |
| 400  | No Content   |
| 401  | Unauthorized |
| 403  | Forbidden    |
| 404  | Not Found    |
| 500  | Other        |

Table 17: GET /service/{id}/{version} – Gets versions of a specific service

| Request  |  |
|--|--|
| GET: /service/{id}/{version}   |  |
| Parameters   |  |
| id: A service id   |  |
| version: A version identifier, the term "latest" gets the most recent version, similarly to the GET: /service/{id}   |  |
| E.g.,  |  |
| <ul style="list-style-type: none"> <li>Get version v2 for the service openaire.openaire_graph → /service/versions/openaire.openaire_graph/v2</li> </ul>  |  |
| Response   |  |
| Status: 200  |  |
| Content-Type: application/json   |  |
| Response Body (A list of service descriptions)   |  |
| <pre>[   {     "id": "openaire.openaire_graph",     "url": "http://api.openaire.eu",     "name": "OpenAIRE Graph",     "tagline": "Open, linked research ",     ...     "version": "v2",     ...   } ]</pre> |  |

|                   |              |
|-------------------|--------------|
| }                 |              |
| ]                 |              |
| Response Messages |              |
| HTTP Status Code  | Reason       |
| 400               | No Content   |
| 401               | Unauthorized |
| 403               | Forbidden    |
| 404               | Not Found    |
| 500               | Other        |

## 4.2.2 Service Provider Controller

The following API methods manage information about a provider. Based on the input parameters, they are used to get a single or a list of service provider resources along with their services.

Table 18: PUT /provider – Updates a service provider

| Request   |              |
|---|--------------|
| PUT: /provider  |              |
| Accept: application/json  |              |
| Cookie: {auth-string}   |              |
| Parameters (The full description of the provider to be updated)                                 |              |
| <pre>provider: {   "id": "string",   "name": "string",   "contactInformation": "string" }</pre> |              |
| Response  |              |
| Status: 200   |              |
| Content-Type: application/json  |              |
| Response Messages   |              |
| HTTP Status Code  | Reason       |
| 201   | Updated      |
| 401   | Unauthorized |
| 403   | Forbidden    |
| 404   | Not Found    |
| 500   | Other        |

Table 19: GET/provider/{id} – Gets a specific provider

| Request   |              |
|---|--------------|
| GET: /provider/{id}   |              |
| Parameters  |              |
| id: The provider's ID<br>e.g., : /provider/geant                                |              |
| Response  |              |
| Status: 200   |              |
| Content-Type: application/json  |              |
| Response Body (A provider description)  |              |
| <pre>{   "id": "geant",   "name": "GÉANT",   "contactInformation": null }</pre> |              |
| Response Messages   |              |
| HTTP Status Code  | Reason       |
| 400   | No Content   |
| 401   | Unauthorized |
| 403   | Forbidden    |
| 404   | Not Found    |
| 500   | Other        |

Table 20: GET/provider/all – Gets all providers in the catalogue

| Request   |  |
|---|--|
| GET: /provider/all  |  |
| Parameters  |  |
| Query: An expression with filters                               |  |
| From: Used for paging the results. It indicates the nth result. |  |
| Quantity: Number of results to fetch                            |  |
| OrderField: The name of the field for ordering the results      |  |

| Order: Order type (Asc Desc)   |              |
|--|--------------|
| <b>Response</b>  |              |
| Status: 200  |              |
| Content-Type: application/json   |              |
| <b>Response Body (A list of provider descriptions)</b>   |              |
| <pre>{   "total": 5,   "from": 0,   "to": 5,   "results": [     {       "id": "geant",       "name": "GÉANT",       "contactInformation": null,     },     {       "id": "egi",       "name": "EGI",       "contactInformation": null     },     {       "id": "prace",       "name": "PRACE",       "contactInformation": null,     },     {       "id": "demo",       "name": "Example Provider",       "contactInformation": null,       "users": null,       "services": null     },     {       "id": "eudat",       "name": "EUDAT CDI consortium",       "contactInformation": null,     },     {       "id": "openaire",       "name": "OpenAIRE consortium",       "contactInformation": null     }   ] }</pre> |              |
| <b>Response Messages</b>   |              |
| HTTP Status Code   | Reason       |
| 400  | No Content   |
| 401  | Unauthorized |
| 403  | Forbidden    |
| 404  | Not Found    |
| 500  | Other        |

Table 21: GET/provider/{id}/services – Gets all services of a provider

|  |  |
|--|--|
| <b>Request</b>   |  |
| GET: /provider/{id}/services   |  |
| <b>Parameters</b>  |  |
| Id: the provider's id, e.g.,   |  |
| <ul style="list-style-type: none"> <li>Get all services from Eudat → /provider/eudat/services</li> </ul> |  |
| <b>Response</b>  |  |
| Status: 200  |  |
| Content-Type: application/json   |  |
| <b>Response Body (A list of provider descriptions)</b>   |  |
| [  |  |

```
{
  "id": "eudat.b2find",
  "url": "https://www.eudat.eu/services/b2find",
  "name": "B2FIND",
  "tagline": "Find research data",
  ...
}
```

**Response Messages**

| HTTP Status Code | Reason       |
|------------------|--------------|
| 400              | No Content   |
| 401              | Unauthorized |
| 403              | Forbidden    |
| 404              | Not Found    |
| 500              | Other        |

### 4.2.3 Funder Controller

The following API methods retrieves information about a funder. Based on the input parameters, they are used to get a single or a list of funder resources along with the list services they have funded.

Table 22: GET/funder/{id} – Gets a specific funder

| Request  |              |
|--|--------------|
| GET: /funder/{id}  |              |
| Parameters   |              |
| id: The funder's ID<br>e.g.,: /funder/ec   |              |
| Response   |              |
| Status: 200<br>Content-Type: application/json  |              |
| Response Body (A funder description)   |              |
| <pre>{   "id": "ec",   "name": "European Commission",   "logo":     "https://upload.wikimedia.org/wikipedia/commons/8/84/European_Commission.svg",   "services": [     "SixSq.nuvla_multi-cloud_application_management_platform",     ...] }</pre> |              |
| Response Messages  |              |
| HTTP Status Code   | Reason       |
| 400  | No Content   |
| 401  | Unauthorized |
| 403  | Forbidden    |
| 404  | Not Found    |
| 500  | Other        |

Table 23: GET/funder/all – Gets all funders in the catalogue

|  |
|--|
| <b>Request</b>   |
| GET: /funder/all   |
| <b>Parameters</b>  |
| <i>Query</i> : An expression with filters<br><i>From</i> : Used for paging the results. It indicates the nth result. |



**Quantity:** Number of results to fetch

**OrderField:** The name of the field for ordering the results

**Order:** Order type (Asc|Desc)

#### Response

Status: 200

Content-Type: application/json

#### Response Body (A list of funder descriptions)

```
{
  "total": 11,
  "from": 0,
  "to": 10,
  "results": [
    {
      "id": "ec",
      "name": "European Commission",
      "logo":
        "https://upload.wikimedia.org/wikipedia/commons/8/84/European_Commission.svg",
      "services": [
        "SixSq.nuvla_multi-cloud_application_management_platform",
        ...,
      ]
    },
    {
      "id": "pop",
      "name": "Prace Optional Programme",
      "logo":
        "https://pbs.twimg.com/profile_images/611860024089006081/CerGhzR4_400x400.png",
      "services": [
        "prace.patc",
        "prace.prace_application_enabling_support",
        ...,
      ]
    }
  ]
}
```

#### Response Messages

| HTTP Status Code | Reason       |
|------------------|--------------|
| 400              | No Content   |
| 401              | Unauthorized |
| 403              | Forbidden    |
| 404              | Not Found    |
| 500              | Other        |

**Table 24: GET/funder/funderStats/{id} – Gets stats about services funded by a specific funder**

| Request  |
|--|
| GET: /funder/funderStats/{id}  |
| Parameters   |
| Id: the funder's id, e.g., <ul style="list-style-type: none"> <li>Get stats for European Commission → /funder/funderStats/ec</li> </ul>  |
| Response   |
| Status: 200  |
| Content-Type: application/json   |
| Response Body (A list of funder statistics)  |
| <pre>{   "Categories": {     "Networking": 8,     "Aggregator": 17,     "Data": 27,     "Security": 5,     "Analytics": 2,     "Training": 9,     "Storage": 3,     "Compute": 4,     "Consulting": 2,   } }</pre> |

```

    "Software": 17,
    "Application": 4,
    "Operations": 12,
    "Other": 6
  },
  "Subcategories": {
    "Archive": 1,
    "High Performance Computing": 1,
    "Platform": 3,
    ...
  },
  "TRL": {
    ...
  },
  ...

```

#### Response Messages

| HTTP Status Code | Reason       |
|------------------|--------------|
| 400              | No Content   |
| 401              | Unauthorized |
| 403              | Forbidden    |
| 404              | Not Found    |
| 500              | Other        |

### 4.2.4 Indicator Controller

The following API methods manage information about Indicators.

Table 25: POST /indicator – Adds a new indicator

Request

POST: /indicator  
Accept: application/json  
Cookie: {auth-string}

Parameters (The description of the indicator)

indicator: {  
 "id": "string",  
 "description": "string",  
 "unit": "string",  
 "dimensions": [  
 "string"  
 ]  
}

Response

Status: 200  
Content-Type: application/json

Response Messages

| HTTP Status Code | Reason       |
|------------------|--------------|
| 201              | Created      |
| 401              | Unauthorized |
| 403              | Forbidden    |
| 404              | Not Found    |
| 500              | Other        |

Table 26: PUT /indicator – Updates an existing indicator

| Request  |
|--|
| PUT: /indicator<br>Accept: application/json<br>Cookie: {auth-string} |
| Parameters (The description of the indicator)                        |
| <pre> indicator: { </pre>  |

```

    "id": "string",
    "description": "string",
    "unit": "string",
    "dimensions": [
      "string"
    ]
  }
}

```

Response

Status: 200

Content-Type: application/json

Response Messages

| HTTP Status Code | Reason       |
|------------------|--------------|
| 201              | Updated      |
| 401              | Unauthorized |
| 403              | Forbidden    |
| 404              | Not Found    |
| 500              | Other        |

Table 27: GET /indicator – Gets an existing indicator

| Request  |              |
|--|--------------|
| GET: /indicator/{id}   |              |
| Parameters   |              |
| id: The indicator's ID<br>e.g.,: 8080/eic-registry/indicator/availability  |              |
| Response   |              |
| Status: 200<br>Content-Type: application/json  |              |
| Response Body (A service description)  |              |
| <pre>{   "id": "availability",   "description": "The fraction of a time period that an item is in a condition to perform its intended function upon demand ("available" indicates that an item is in this condition); availability is often expressed as a probability.",   "unit": "percentage",   "dimensions": [     "TIME"   ] }</pre> |              |
| Response Messages  |              |
| HTTP Status Code   | Reason       |
| 400  | No Content   |
| 401  | Unauthorized |
| 403  | Forbidden    |
| 404  | Not Found    |
| 500  | Other        |

Table 28: GET/indicator/all – Gets all indicators in the catalogue

|   |
|---|
| <b>Request</b>  |
| GET: /indicator/all   |
| <b>Parameters</b>   |
| <b>Query:</b> An expression with filters<br><b>From:</b> Used for paging the results. It indicates the nth result.<br><b>Quantity:</b> Number of results to fetch<br><b>OrderField:</b> The name of the field for ordering the results<br><b>Order:</b> Order type (Asc Desc) |
| <b>Response</b>   |
| Status: 200   |
| Content-Type: application/json  |
| <b>Response Body (A list of indicator descriptions)</b>   |
| <pre> { </pre>  |

```

"total": 11,
"from": 0,
"to": 10,
"results": [
  {
    "id": "serviceRequests",
    "name": "Requests",
    "description": "The total number of demands for a specific service, e.g.
service access requests, service information requests, requests for other material
service attributes, etc.",
    "dimensions": [
      "time"
    ],
    "unit": "numeric",
    "unitName": "(requests)"
  },
  {
    "id": "serviceOrderCapacity",
    "name": "Order Capacity",
    "description": "The maximum volume of available service provision while
maintaining standards of quality and performance. e.g. 3,000 service orders per
day",
    "dimensions": [
      "time"
    ],
    "unit": "numeric",
    "unitName": "(orders/day)"
  },
  {
    "id": "serviceUsage",
    "name": "Usage",
    "description": "The level or percentage of actual penetration/utilisation of a
specific service.",
    "dimensions": [
      "time"
    ],
    "unit": "percentage",
    "unitName": "%"
  },
  ...
]
}

```

#### Response Messages

| HTTP Status Code | Reason       |
|------------------|--------------|
| 400              | No Content   |
| 401              | Unauthorized |
| 403              | Forbidden    |
| 404              | Not Found    |
| 500              | Other        |

### 4.2.5 Measurement Controller

The following API methods manage information about measurements of Indicators.

Table 29: POST /measurement – Adds a new measurement

| Request   |
|---|
| POST: /measurement                              |
| Accept: application/json                        |
| Cookie: {auth-string}                           |
| Parameters (The description of the measurement) |
| measurement: {                                  |

```

    "id": "string",
    "serviceId": "string",
    "IndicatorId": "string",
    "timePeriod": "string",
    "locations": [
      "string"
    ],
    "Value": "string"
  }

```

**Response**

Status: 200

Content-Type: application/json

**Response Messages**

| HTTP Status Code | Reason       |
|------------------|--------------|
| 201              | Created      |
| 401              | Unauthorized |
| 403              | Forbidden    |
| 404              | Not Found    |
| 500              | Other        |

**Table 30: PUT /measurement – Updates an existing measurement**

Request

PUT: /measurement  
Accept: application/json  
Cookie: {auth-string}

Parameters (The description of the measurement)

measurement: {  
 "id": "string",  
 "serviceId": "string",s  
 "IndicatorId": "string",  
 "time": "string",  
 ...  
}

Response

Status: 200  
Content-Type: application/json

Response Messages

| HTTP Status Code | Reason       |
|------------------|--------------|
| 201              | Updated      |
| 401              | Unauthorized |
| 403              | Forbidden    |
| 404              | Not Found    |
| 500              | Other        |

**Table 31: GET /measurement – Gets an existing measurement**

| Request  |
|--|
| GET: /measurement/{id}   |
| Parameters   |
| id: The measurement's ID , e.g. a UUID<br>e.g., : /measurement/b0c1d117-fe12-4514-91b5-e8b5980060a7  |
| Response   |
| Status: 200<br>Content-Type: application/json  |
| Response Body (A service description)  |
| <pre> {   "id": "b0c1d117-fe12-4514-91b5-e8b5980060a7",   "indicatorId": "serviceUsers",   "serviceId":   "SeaDataNet.seadatanet_european_directory_of_marine_environmental_data_edmed",   "time": 1546214400000, </pre> |

```

    "locations": null,
    "valueIsRange": false,
    "value": "35000",
    "rangeValue": null
  }

```

**Response Messages**

| HTTP Status Code | Reason       |
|------------------|--------------|
| 400              | No Content   |
| 401              | Unauthorized |
| 403              | Forbidden    |
| 404              | Not Found    |
| 500              | Other        |

**Table 32: GET /measurement/service/{id} – Gets all Measurements for the specific service.**

| <b>Request</b>   |              |
|--|--------------|
| GET: /measurement/service/{id}   |              |
| <b>Parameters</b>  |              |
| id: The service ID ,<br>e.g., : /measurement/service/<br>SeaDataNet.seadatanet_european_directory_of_the_cruise_summary_reports_csr  |              |
| <b>Response</b>  |              |
| Status: 200  |              |
| Content-Type: application/json   |              |
| <b>Response Body (All measurements of a specific service)</b>  |              |
| <pre> {   "total": 2,   "from": 0,   "to": 2,   "results": [     {       "id": "558f9e94-86ee-4574-bc1e-fbbddd7f0b71",       "indicatorId": "serviceUsers",       "serviceId": "SeaDataNet.seadatanet_european_directory_of_the_cruise_summary_reports_csr",       "time": 1540857600000,       "locations": null,       "valueIsRange": false,       "value": "889",       "rangeValue": null     },     {       "id": "8f812da4-268b-44f4-8f31-073d9921cd43",       "indicatorId": "serviceReliability",       "serviceId": "SeaDataNet.seadatanet_european_directory_of_the_cruise_summary_reports_csr",       "time": 1540857600000,       "locations": null,       "valueIsRange": false,       "value": "99.83",       "rangeValue": null     }   ],   "facets": null } </pre> |              |
| <b>Response Messages</b>   |              |
| HTTP Status Code   | Reason       |
| 400  | No Content   |
| 401  | Unauthorized |
| 403  | Forbidden    |
| 404  | Not Found    |
| 500  | Other        |

**Table 33: GET /measurement/latest/service/{id} – Gets all latest Measurements for the specific service.**

| Request  |              |
|--|--------------|
| GET: /measurement/latest/service/{id}  |              |
| Parameters   |              |
| <b>id:</b> The service ID ,<br>e.g., : /measurement/latest/service/<br>SeaDataNet.seadatanet_european_directory_of_the_cruise_summary_reports_csr  |              |
| Response   |              |
| Status: 200<br>Content-Type: application/json  |              |
| Response Body (All latest measurements of a specific service)  |              |
| <pre>{   "total": 2,   "from": 0,   "to": 2,   "results": [     {       "id": "558f9e94-86ee-4574-bc1e-fbbddd7f0b71",       "indicatorId": "serviceUsers",       "serviceId": "SeaDataNet.seadatanet_european_directory_of_the_cruise_summary_reports_csr",       "time": 1540857600000,       "locations": null,       "valueIsRange": false,       "value": "889",       "rangeValue": null     },     {       "id": "8f812da4-268b-44f4-8f31-073d9921cd43",       "indicatorId": "serviceReliability",       "serviceId": "SeaDataNet.seadatanet_european_directory_of_the_cruise_summary_reports_csr",       "time": 1540857600000,       "locations": null,       "valueIsRange": false,       "value": "99.83",       "rangeValue": null     }   ],   "facets": null }</pre> |              |
| Response Messages  |              |
| HTTP Status Code   | Reason       |
| 400  | No Content   |
| 401  | Unauthorized |
| 403  | Forbidden    |
| 404  | Not Found    |
| 500  | Other        |

#### 4.2.6 Statistics Controller

The following API methods retrieve information about usage statistics collected by elnraCentral.

**Table 34: GET /stats/provider/favourites/{id} – Gets total favourites for all services of a provider**

| Request  |  |
|--|--|
| GET: /stats/provider/favourites/{id}                                     |  |
| Parameters   |  |
| <b>id:</b> The provider's ID<br>e.g., : /stats/provider/favourites/eudat |  |
| Response   |  |
| Status: 200  |  |

| Content-Type: application/json   |              |
|--|--------------|
| <b>Response Body (A list of statistics)</b>  |              |
| <pre>{   "2018-03-29": 85,   "2018-03-28": 69,   "2018-04-18": 70,   "2018-04-19": 59,   "2018-04-12": 66,   "2018-04-13": 79,   "2018-04-10": 62,   ... }</pre> |              |
| <b>Response Messages</b>   |              |
| HTTP Status Code   | Reason       |
| 400  | No Content   |
| 401  | Unauthorized |
| 403  | Forbidden    |
| 404  | Not Found    |
| 500  | Other        |

Table 35: GET /stats/provider/ratings/{id} – Gets average ratings for all services of a provider

| <b>Request</b>  |              |
|---|--------------|
| GET: /stats/provider/ratings/{id}   |              |
| <b>Parameters</b>   |              |
| id: The provider's ID<br>e.g.,: /stats/provider/ratings/prace   |              |
| <b>Response</b>   |              |
| Status: 200   |              |
| Content-Type: application/json  |              |
| <b>Response Body (A list of statistics)</b>   |              |
| <pre>{   "2018-03-29": 22.043709,   "2018-03-28": 29.315617,   "2018-04-18": 24.529007,   "2018-04-19": 29.378208,   "2018-04-12": 28.294714,   ... }</pre> |              |
| <b>Response Messages</b>  |              |
| HTTP Status Code  | Reason       |
| 400   | No Content   |
| 401   | Unauthorized |
| 403   | Forbidden    |
| 404   | Not Found    |
| 500   | Other        |

Table 36: GET /stats/provider/visits/{id} – Gets total visits for all services of a provider

|   |  |
|---|--|
| <b>Request</b>  |  |
| GET: /stats/provider/visits/{id}                                |  |
| <b>Parameters</b>   |  |
| id: The provider's ID<br>e.g.,: /stats/provider/visits/openaire |  |
| <b>Response</b>   |  |
| Status: 200   |  |
| Content-Type: application/json                                  |  |
| <b>Response Body (A list of statistics)</b>                     |  |
| <pre>{   "2018-03-29": 1250,</pre>                              |  |



```
"2018-03-28": 2501,
"2018-04-18": 3456,
"2018-04-19": 7890,
...
}
```

**Response Messages**

| HTTP Status Code | Reason       |
|------------------|--------------|
| 400              | No Content   |
| 401              | Unauthorized |
| 403              | Forbidden    |
| 404              | Not Found    |
| 500              | Other        |

The aforementioned methods are also available for retrieving statistics *of a single service* accepting as parameter the service id instead of the provider ID.

**Table 37: GET /stats/provider/visitation/{id} – Gets % of visits for all services of a provider**

| Request   |              |
|---|--------------|
| GET: /stats/provider/visitation/{id}  |              |
| Parameters  |              |
| id: The provider's ID<br>e.g.,: /stats/provider/visitation/egi  |              |
| Response  |              |
| Status: 200<br>Content-Type: application/json   |              |
| Response Body (A list of statistics)  |              |
| <pre>{   "EGI Check-In": 0.09,   "EGI Validated Software and Repository": 0.15,   "EGI Accounting": 0.07,   "EGI Operational tools": 0.25,   "EGI FitSM Training": 0.01,   "EGI Applications on Demand": 0.25,   "EGI Marketplace": 0.04,   "EGI Configuration Database": 0.11,   ... }</pre> |              |
| Response Messages   |              |
| HTTP Status Code  | Reason       |
| 400   | No Content   |
| 401   | Unauthorized |
| 403   | Forbidden    |
| 404   | Not Found    |
| 500   | Other        |

## 4.2.7 Vocabulary Controller

The following API methods retrieve information about enumerated fields used in eInfraCentral catalogue, such as the categories, subcategories, phases, etc.

**Table 38: GET /vocabulary – Gets all terms for a vocabulary**

| Request          |
|------------------|
| GET: /vocabulary |
| Parameters       |

*type*: The type of the vocabulary as listed in Section ; one of the values (SUPERCATEGORY|CATEGORY|SUBCATEGORY|LANGUAGE|PLACE|TRL|PHASE|SCIENTIFIC\_DOMAIN|SCIENTIFIC\_SUBDOMAIN|TARGET\_USERS|ACCESS\_TYPE|ACCESS\_MODE|ORDER\_TYPE)

E.g.,

- Get all allowed TRL values in the catalogue → /vocabulary/TRL

#### Response

Status: 200

Content-Type: application/json

#### Response Body (A list of vocabulary terms)

```
{
  "total": 1,
  "from": 0,
  "to": 1,
  "results": [
    {
      "id": "trl",
      "name": "TRL",
      "entries": {
        "trl-7": {
          "children": null,
          "extras": {},
          "id": "trl-7",
          "name": "7 - system prototype demonstration in operational
environment"
        },
        "trl-8": {
          "children": null,
          "extras": {},
          "id": "trl-8",
          "name": "8 - system complete and qualified"
        },
        ...
      ]
    }
  ]
}
```

#### Response Messages

| HTTP Status Code | Reason       |
|------------------|--------------|
| 400              | No Content   |
| 401              | Unauthorized |
| 403              | Forbidden    |
| 404              | Not Found    |
| 500              | Other        |

Table 39: GET /vocabulary/all – Gets vocabulary terms based on filters

| Request  |
|--|
| GET: /vocabulary/all   |
| Parameters   |
| <i>Query</i> : The vocabulary type (i.e., name of the attribute) for which we want to retrieve the terms.<br><i>From</i> : Used for paging the results. It indicates the nth result.<br><i>Quantity</i> : Number of results to fetch |
| E.g., <ul style="list-style-type: none"> <li>• Get the first 5 categories → /vocabulary/all?query=categories&amp;from=1&amp;quantity=5</li> </ul>  |
| Response   |
| Status: 200  |
| Content-Type: application/json   |

**Response Body (A list of vocabulary terms)**

```
{
  "total": 1,
  "from": 0,
  "to": 1,
  "results": [
    {
      "id": "trl",
      "name": "TRL",
      "entries": {
        "trl-7": {
          "children": null,
          "extras": {},
          "id": "trl-7",
          "name": "7 - system prototype demonstration in operational
environment"
        },
        "trl-8": {
          "children": null,
          "extras": {},
          "id": "trl-8",
          "name": "8 - system complete and qualified"
        },
        ...
      ]
    }
  ]
}
```

**Response Messages**

| HTTP Status Code | Reason       |
|------------------|--------------|
| 400              | No Content   |
| 401              | Unauthorized |
| 403              | Forbidden    |
| 404              | Not Found    |
| 500              | Other        |

**Table 40: GET /vocabulary/countries/EU (or /WW) – Gets the list of country codes belonging to EU or Worldwide**

| Request  |              |
|--|--------------|
| GET: /vocabulary/countries/EU<br>GET: /vocabulary/countries/WW           |              |
| Parameters   |              |
| NONE   |              |
| Response   |              |
| Status: 200<br>Content-Type: application/json                            |              |
| Response Body (A list of vocabulary terms)                               |              |
| [<br>"AX",<br>"AT",<br>"BE",<br>"BG",<br>"HR",<br>"CY",<br>"CZ",<br>...] |              |
| Response Messages  |              |
| HTTP Status Code   | Reason       |
| 400  | No Content   |
| 401  | Unauthorized |
| 403  | Forbidden    |
| 404  | Not Found    |

**500**

Other

Appendix 2 provide some guidelines – in the form of a step-by-step example – for a service provider to start using the API for adding and updating service information in the eInfraCentral Gateway.

### 4.3 User Authentication

Authentication to the services is implemented via an access token generated by the AAI server. eInfraCentral currently uses the AAI mechanism of the European Open Science Cloud (EOSC) Portal, which offers users the ability to authenticate themselves and generate an API token (<https://aai.eosc-portal.eu/eic-api/token.php>). The API token is used in all eInfraCentral API methods requiring authenticated access. The authenticated user can request the generation, deletion, and listing of API tokens via the EOSC Portal AAI, which acts as a mediating management interface.

### 4.4 Synchronization between a remote and the eInfraCentral catalogue

The eInfraCentral software platform allows for a service provider to add a new service or measurement in its own catalogue, and automatically synchronize the new addition in the eInfraCentral Catalogue, via the use of the appropriate API calls. More specifically, when a provider makes a POST, PUT a Service (or a Measurement) in its catalogue, the corresponding API method is called. Each of these methods needs 2 values to work properly:

1. The host, i.e. the eInfraCentral API URL, where the synchronization will take place.
2. The token needed for the provider to authorize the synchronization.

Both of these values are configured through the `registry.properties` file and a `restTemplate` (used to make REST calls) is created with the required parameters (host url, method).

For each call, if all values pass the validation, the call is successful and the Service (or Measurement) is synchronized to the eInfraCentral catalogue.

If at least one value fails the validation (e.g. wrong token), then the call fails and is inserted in a queue. This queue consists of failed calls. Another service, called `ScheduleService`, is assigned with the task to periodically check the queue (for failed Services and Measurements) and retry the corresponding calls (every 10 mins).

Making use of the eInfraCentral Software backend, a 3<sup>rd</sup> party provider can automate the process of maintaining its own catalogue and in the same time propagate changes to the central eInfraCentral catalogue.

## 5 Conclusions

This report provided the specifications of the APIs (API v2.0) of the eInfraCentral Gateway. The API specifications offer the ability to service providers to populate the eInfraCentral Gateway with service related information as well as to service catalogue consumers to have access and programmatically retrieve this information for use in their systems. Namely, it offers the ability to add, update, validate and retrieve information about a *service*, a *service provider*, a *funder*, a set of service-related *indicators* and *usage statistics* collected in the eInfraCentral Gateway as well as auxiliary information concerning *vocabularies* and *terms* used for service classification.

The complete [eInfraCentral API documentation](http://www.einfracentral.eu/developers) is a living document, generated using frameworks compliant with OpenAPI Specification 2.0. **API v2.0** in the form of OpenAPI specifications is available at <http://www.einfracentral.eu/developers>.

## Appendix 1: Detailed Description of the eInfraCentral resource model

The following Appendix presents a detailed schema of each resource used in the eInfraCentral Data catalogue and is accessible via the eInfraCentral APIs.

Table 41: Service resource

| Service Attribute         | Attribute name in API | Description   | Type   | Example Value   | <u>M</u> andatory \ <u>O</u> ptional |
|---------------------------|-----------------------|---|--|---|--------------------------------------|
| Basic Service information |                       |   |  |   |                                      |
| <b>ID</b>                 | id                    | Global unique and persistent identifier of the service/resource.  | Formatted Text (providerID.Service Name)<br>; set during the registration/onboarding process | eudat.b2drop  | M                                    |
| <b>Name</b>               | name                  | Brief and descriptive name of service/resource as assigned by the service/resource provider.  | String (max 80)  | B2DROP  | M                                    |
| <b>URL</b>                | url                   | Webpage with information about the service/resource usually hosted and maintained by the service/resource provider.                                     | URL  | <a href="https://www.eudat.eu/services/b2drop">https://www.eudat.eu/services/b2drop</a>   | M                                    |
| <b>Description</b>        | description           | A high-level description in fairly non-technical terms of what the service/resource does, functionality it provides and resources it enables to access. | String (max 1000)  | B2SHARE is a user-friendly, reliable and trustworthy service that allows to store and share worldwide small-scale research data, results and metadata from diverse contexts | M                                    |

|                   |                    |  |                              |  |   |
|-------------------|--------------------|--|------------------------------|--|---|
|                   |                    |  |                              | and guarantees their long-term persistence.  |   |
| <b>Logo</b>       | logo               | Link to the logo/visual identity of the service.   | URL                          | <a href="http://www.egi.eu/cloud-compute/Symbol.jpg">http://www.egi.eu/cloud-compute/Symbol.jpg</a>  | M |
| <b>Tagline</b>    | tagline            | Short catch-phrase for marketing and advertising purposes. It will be usually displayed close the service name and should refer to the main value or purpose of the service.   | String<br>(max 100)          | Store, share and access your files and their metadata on a global scale  | O |
| <b>User Value</b> | userValue          | The benefit to a user/customer delivered by a service; benefits are usually related to alleviating pains (e.g., eliminate undesired outcomes, obstacles or risks) or producing gains (e.g. increased performance, social gains, positive emotions or cost saving). | String<br>(max 1000)         | For the communities who need to guard against data loss, B2SAFE is a customer facing service that allow data replication and safe storage between geographically distributed centres in the EUDAT CDI. | O |
| <b>User Base</b>  | userBaseList<br>[] | List of customers, communities, users, etc. using the service.   | List of Strings<br>(max 100) | 900+ data providers in Europe with different compatibility levels. Adoption in Latin America (LaReferencia), Mexico and Japan.   | O |
| <b>Use Cases</b>  | Usecases []        | List of use cases supported by this service/resource.  | List of Strings<br>(max 100) |  | O |

|   |                        |   |                                  |   |   |
|---|------------------------|---|----------------------------------|---|---|
| <b>Multimedia</b>                         | multimediaUrls[]       | Link to video, screenshots or slides showing details of the service/resource.                             | List of URLs                     | <a href="http://www.egi.eu/cloud-compute/Multimedia/">http://www.egi.eu/cloud-compute/Multimedia/</a>             | O |
| <b>Options</b>                            | Options[]              | High-level description of the various options or forms in which the service/resource can be instantiated. | List of Options                  | "Standard or enhanced", "Bronze, silver and gold", "10, 20, 30", "Elementary Level, Advanced Level, Expert Level" | O |
| <b>Required Services</b>                  | requiredServices[]     | List of other services/resources required with this service/resource.                                     | Service/Resource ID              | List of Service IDs   | O |
| <b>Related Services</b>                   | relatedServices[]      | List of other services/resources that are commonly used with this service/resource.                       | Service/Resource ID              | List of Service IDs   | O |
| <b>Service Classification Information</b> |                        |   |                                  |   |   |
| <b>Providers</b>                          | Providers[]            | The organisation that manages and delivers the service and with whom the customer signs the SLA.          | List of Provider IDs             | e.g. GEANT, PRACE, EGI, EUDAT, OpenAIRE, etc.   | M |
| <b>Scientific Domain</b>                  | scientificDomains[]    | The branch of science, scientific discipline that is related to the service/resource.                     | List of values (see Section 3.2) | Humanities  | M |
| <b>Scientific Subdomain</b>               | scientificSubdomains[] | The subbranch of science, scientific subdiscipline that is related to the service/resource.               | List of values (see Section 3.2) | History & Archaeology   | M |



|                      |                   |   |                                  |  |   |
|----------------------|-------------------|---|----------------------------------|--|---|
| <b>Category</b>      | categories[]      | A named group of services/resources that offer access to the same type of resource or capabilities.                                     | List of values (see Section 3.2) | Network  | M |
| <b>Subcategory</b>   | subcategories[]   | A named group of services/resources that offer access to the same type of resource or capabilities, within the defined service category | List of values (see Section 3.2) | Direct Connect   | M |
| <b>Supercategory</b> | Supercategories[] | A named group for a predefined list of categories.  | List of values (see Section 3.2) | Access physical & eInfrastructures   | M |
| <b>Target Users</b>  | targetUsers[]     | Type of users/customers allowed to commission/benefit from the service.   | List of values (see Section 3.2) | Researchers, Research Groups, Research Communities, Research Projects, etc | O |
| <b>Language</b>      | languages[]       | Languages of the User interface   | List of values (see Section 3.2) | English  | M |
| <b>Place</b>         | places[]          | Regions/Countries Availability  | List of values (see Section 3.2) | Global   | M |
| <b>Access Type</b>   | accessTypes[]     | The way a user can access the service/resource (Remote, Physical, Virtual, etc.)  | List of values (see Section 3.2) | Remote, Physical   | O |
| <b>Access Mode</b>   | accessModes[]     | The mode a user can access the service/resource (Excellence Driven, Market driven, etc)   | List of values (see Section 3.2) | Policy Based, Wide Access  | O |

|                                     |                  |  |                                  |  |   |
|-------------------------------------|------------------|--|----------------------------------|--|---|
| <b>Funding</b>                      | funders[]        | Sources of funding for the development and/or operation of the service.              | List of Funder IDs               | EC   | O |
| <b>Tags</b>                         | tags[]           | Keywords associated to the service/resource to simplify search by relevant keywords. | Free text \ Multiple values      | AIA, Security  | O |
| <b>Service Maturity Information</b> |                  |  |                                  |  |   |
| <b>Phase</b>                        | phase            | Phase of the service/resource lifecycle.   | List of values (see Section 3.2) | Production, Implementation, Planned, etc             | M |
| <b>Technology Readiness Level</b>   | trl              | The Technology Readiness Level of the Tag of the service/resource.                   | List of values (see Section 3.2) | TRL8 - system complete and qualified                 | M |
| <b>Version</b>                      | version          | Version of the service/resource that is in force.                                    | Free text (max 10)               | 3.1  | M |
| <b>Last Update</b>                  | lastUpdate       | Date of the latest update of the service/resource.                                   | Date (dd/mm/yyyy)                | 28/02/2018   | O |
| <b>Change Log</b>                   | changeLog        | Summary of the service/resource features updated from the previous version.          | Free text (max 1000)             | Upgrade of user interface. Correction of minor bugs. | O |
| <b>Certifications</b>               | certifications[] | List of certifications obtained for the service from independent third parties.      | Free text \ Multiple values      | ISO/IEC 20000-1:2011                                 | O |
| <b>Standards</b>                    | standards[]      | List of standards supported by the service.  | Free text \ Multiple values      | DC-Terms   | O |

| Service Contractual Information |                       |  |                                  |   |   |
|---------------------------------|-----------------------|--|----------------------------------|---|---|
| <b>Order Type</b>               | orderType             | Describes if the service/resource can be accessed with an ordering process.  | List of values (see Section 3.2) | Open Access   | M |
| <b>Order</b>                    | order                 | Webpage to request the service/resource from the service/resource provider.  | URL                              | <a href="http://www.egi.eu/cloud-compute/Order">http://www.egi.eu/cloud-compute/Order</a>                 | O |
| <b>Service Level Agreement</b>  | serviceLevelAgreement | Webpage with the information about the levels of performance that a service/resource provider is expected to deliver.                    | URL                              | <a href="http://www.egi.eu/cloud-compute/SLA">http://www.egi.eu/cloud-compute/SLA</a>                     | O |
| <b>Terms Of Use</b>             | termsOfUse            | Webpage describing the rules, service/resource conditions and usage policy which one must agree to abide by in order to use the service. | URL                              | <a href="http://www.egi.eu/cloud-compute/TermsOfUse">http://www.egi.eu/cloud-compute/TermsOfUse</a>       | O |
| <b>Privacy Policy</b>           | privacyPolicy         | Link to the privacy policy applicable to the service.  | URL                              | <a href="http://www.egi.eu/cloud-compute/PrivacyPolicy">http://www.egi.eu/cloud-compute/PrivacyPolicy</a> | O |
| <b>Access Policy</b>            | accessPolicy          | Webpage to the information about the access policies that apply.   | URL                              | <a href="http://www.egi.eu/cloud-compute/AcessPolicy">http://www.egi.eu/cloud-compute/AcessPolicy</a>     | O |
| <b>Payment Model</b>            | paymentModel          | Webpage with the supported payment models and restrictions that apply to each of them  | URL                              | <a href="http://www.egi.eu/cloud-compute/Payment">http://www.egi.eu/cloud-compute/Payment</a>             | O |
| <b>Pricing</b>                  | pricing               | Webpage with the information on the price scheme for this service in   | URL                              | <a href="http://www.egi.eu/cloud-compute/Pricing">http://www.egi.eu/cloud-compute/Pricing</a>             | O |

|                                    |              |   |                 |  |   |
|------------------------------------|--------------|---|-----------------|--|---|
|                                    |              | case the customer is charged for.   |                 |  |   |
| <b>Service Support Information</b> |              |   |                 |  |   |
| <b>Manual</b>                      | manual       | Link to the service/resource user manual and documentation.   | URL             | <a href="http://www.egi.eu/cloud-compute/UserManual">http://www.egi.eu/cloud-compute/UserManual</a>  | O |
| <b>Training</b>                    | training     | Webpage to training information on the service.   | URL             | <a href="http://www.egi.eu/cloud-compute/Training">http://www.egi.eu/cloud-compute/Training</a><br><a href="http://training.egi.eu">http://training.egi.eu</a> | O |
| <b>Helpdesk</b>                    | helpdesk     | The URL to a webpage with the contact person or helpdesk to ask more information from the service/resource provider about this service. | URL             | <a href="http://www.egi.eu/cloud-compute/Helpdesk">http://www.egi.eu/cloud-compute/Helpdesk</a><br><a href="http://helpdesk.egi.eu">http://helpdesk.egi.eu</a> | O |
| <b>Monitoring</b>                  | monitoring   | Webpage with monitoring information about this service  | URL             | <a href="http://www.egi.eu/cloud-compute/monitoring">http://www.egi.eu/cloud-compute/monitoring</a>  | O |
| <b>Maintenance</b>                 | maintenance  | Webpage with information about planned maintenance windows for this service   | URL             | <a href="http://www.egi.eu/cloud-compute/maintenance">http://www.egi.eu/cloud-compute/maintenance</a>  | O |
| <b>Service Contact Information</b> |              |   |                 |  |   |
| <b>Owner Name</b>                  | ownerName    | Name of the person who has accountability for the whole service/resource from a management point of view                                | String (max 50) |  | O |
| <b>Owner Contact</b>               | ownerContact | E-mail contact of the service/resource owner  | Email           |  | O |

|  |                          |  |                              |      |   |
|--|--------------------------|--|------------------------------|------|---|
| <b>Support Name</b>                                      | supportName              | Name of the person to request technical/ operational support   | String (max 50)              |      | 0 |
| <b>Support Contact</b>                                   | supportContact           | E-mail contact of the person to request technical/ operational support   | Email                        |      | 0 |
| <b>Security Name</b>                                     | securityName             | Name of the person responsible for the security aspects of the service/resource  | String (max 50)              |      | 0 |
| <b>Security Contact</b>                                  | securityContact          | Contact of the person responsible for the security aspects of the service/resource   | email                        |      | 0 |
| <b>Service Level Targets and Performance Information</b> |                          |  |                              |      |   |
| <b>Cost</b>  | Indicator{} <sup>4</sup> | The monetary value that a user is requested to pay in order to utilise a specific service, e.g. free of charge, 100 €, etc.  | Percentage, Numeric, Boolean | 100€ | 0 |
| <b>Requests</b>  | Indicator{}              | The total cumulative number of demands for a specific service since it was first offered publicly, e.g. service access requests, service information requests, requests for other material service | Percentage, Numeric, Boolean | 1000 | 0 |

<sup>4</sup> See resource description in Table 42

|                     |             |   |                              |                             |   |
|---------------------|-------------|---|------------------------------|-----------------------------|---|
|                     |             | attributes, etc.  |                              |                             |   |
| <b>Users</b>        | Indicator{} | The total cumulative number of people who utilise the specific service at the time of reporting.  | Percentage, Numeric, Boolean | 100                         | 0 |
| <b>Usage</b>        | Indicator{} | The level or percentage of actual utilisation of a specific service.  | Percentage, Numeric, Boolean | 99,99%                      | 0 |
| <b>Capacity</b>     | Indicator{} | The maximum volume of available service provision while maintaining standards of quality and performance. e.g. 30,000 concurrent users, 3,000 service orders per day, etc.  | Percentage, Numeric, Boolean | 30,000 concurrent users     | 0 |
| <b>Coverage</b>     | Indicator{} | The range of geographical areas and/or thematic sectors that a service is addressed to, e.g. EU Member States (regional), high energy physics (scientific community) etc.   | Percentage, Numeric, Boolean | EU Member States (regional) | 0 |
| <b>Availability</b> | Indicator{} | The Service Availability, i.e. the fraction of a time period that an item is in a condition to perform its intended function upon demand ("available" indicates that an item is in this condition); availability is often expressed as a probability. | Percentage, Numeric, Boolean | 99.99%                      | 0 |

|                                  |             |   |                              |        |   |
|----------------------------------|-------------|---|------------------------------|--------|---|
| <b>Reliability</b>               | Indicator{} | A Service Reliability, i.e. the probability that an item will function without failure under stated conditions for a specified amount of time. "Stated conditions" indicate perquisite conditions external to the item being considered. For example, a stated condition for a supercomputer might be that power and cooling must be available - thus a failure of the power or cooling systems would not be considered a failure of the supercomputer. | Percentage, Numeric, Boolean | 99.99% | 0 |
| <b>Serviceability/Durability</b> | Indicator{} | Serviceability, i.e. the probability that an item will be retained in, or restored to, a condition to perform its intended function within a specified period of time. Durability, i.e. the ability of a physical product to remain functional, without requiring excessive maintenance or repair, when faced with the challenges of normal operation over its design lifetime.   | Percentage, Numeric, Boolean | 99.99% | 0 |

|   |             |   |                              |  |   |
|---|-------------|---|------------------------------|--|---|
| <b>Other Performance Indicator Name/Value</b> | Indicator{} | Other Service Level Target or Performance Indicator | Percentage, Numeric, Boolean |  | O |
|---|-------------|---|------------------------------|--|---|

Table 42: Indicator resource

| Indicator Attribute                  | Attribute name in API | Description  | Type   | Example Value                  | <u>M</u> andatory \ <u>O</u> ptional |
|--------------------------------------|-----------------------|--|--|--------------------------------|--------------------------------------|
| <b>Indicator ID</b>                  | id                    | Global unique and persistent identifier of the indicator       | Free Text  | projectsawarded                | M                                    |
| <b>Indicator Description</b>         | description           | Brief description of indicator                                 | Free Text  | The number of Projects awarded | M                                    |
| <b>Indicator Name</b>                | Name                  | Brief and descriptive name of indicator                        | Free Text  | Projects awarded               | M                                    |
| <b>Indicator Dimensions</b>          | dimensions []         | The dimension characterising the measurements of the indicator | List of text values used as dimensions of the indicator {TIME, LOCATIONS}      |                                | M                                    |
| <b>Indicator Unit of Measurement</b> | unit                  | The unit of measurement  | A single value from (Percentage, Numeric, Boolean) used as unit of measurement | Numeric                        | M                                    |
| <b>Indicator Unit Name</b>           | unit                  | The unit name of measurement                                   | Name of the unit measuring the indicator                                       | projects                       | M                                    |



Table 43: Indicator Measurement Resource

| Measurement Attribute         | Attribute name in API           | Description  | Type   | Example Value                             | <u>M</u> andatory \ <u>O</u> ptional |
|-------------------------------|---------------------------------|--|--|---|--------------------------------------|
| Measurement ID                | id                              | Global unique and persistent identifier of the measurement   | UUID assigned during insertion               | 558f9e94-86ee-4574-bc1e-fbbddd7f0b71      | M                                    |
| Indicator Id                  | Indicator id                    | The indicator ID   | Indicator ID                                 | availability                              | M                                    |
| Service ID                    | Service id                      | The service associated with the measurement                  | Service ID                                   | egi.egi_cloud_compute                     | M                                    |
| Measurement Dimensions Values | Time, Locations []              | The values for the dimensions characterising the measurement | Dimension values identifying the measurement | TIME = "1/1/2019", Locations [UK, GR, DE] | M                                    |
| Measurement Value             | Value                           | The actual value of the measurement                          | Number                                       | 99.9%                                     | O <sup>5</sup>                       |
| Measurement Range Value       | RangeValues{FromValue, ToValue} | A range characterized by from – to values                    | Numbers                                      | 90 - 99.9%                                | O                                    |

Table 44: Options Resource

| Options Attribute | Attribute name in API | Description | Type | Example Value | <u>M</u> andatory \ <u>O</u> ptional |
|-------------------|-----------------------|-------------|------|---------------|--------------------------------------|
|-------------------|-----------------------|-------------|------|---------------|--------------------------------------|

<sup>5</sup> Either the Value or Range value attribute is mandatory

|                           |             |  |                   |  |   |
|---------------------------|-------------|--|-------------------|--|---|
| <b>Option ID</b>          | id          | Identifier of the service/resource option.                         | UUID              |  | M |
| <b>Option Name</b>        | name        | Name of the service/resource option.                               | String (max 80)   |  | M |
| <b>Service ID</b>         | serviceid   | The service associated with this option                            | Service id        |  | M |
| <b>Option Description</b> | description | The description of the service/resource option.                    | String (max 1000) |  | M |
| <b>Option Logo</b>        | logo        | Link to the logo/visual identity of the service/resource provider. | URL               |  | O |
| <b>Option URL</b>         | url         | Webpage with information about the service/resource option.        | URL               |  | M |

Table 45: Service Provider Resource

| <b>Provider Attribute</b>   | <b>Attribute name in API</b> | <b>Description</b>  | <b>Type</b>       | <b>Example Value</b>                        | <b><u>M</u>andatory \ <u>O</u>ptional</b> |
|-----------------------------|------------------------------|---|-------------------|---|---|
| <b>Provider ID</b>          | id                           | Identifier of the service/resource provider.                  | Free Text         | openaire                                    | M   |
| <b>Provider Name</b>        | name                         | Name of the service/resource provider.                        | String (max 80)   | OpenAIRE                                    | M   |
| <b>Provider URL</b>         | url                          | Webpage with information about the service/resource provider. | URL               | https://www.openaire.eu                     | M   |
| <b>Provider Description</b> | description                  | The description of the service/resource provider.             | String (max 1000) | OpenAIRE's mission is closely linked to the | M   |

|                                   |                       |  |                 |   |   |
|-----------------------------------|-----------------------|--|-----------------|---|---|
|                                   |                       |  |                 | mission of the European Commission: to provide unlimited, barrier free, open access to research outputs financed by public funding in Europe...         |   |
| <b>Provider Logo</b>              | logo                  | Link to the logo/visual identity of the service/resource provider.     | URL             | <a href="https://www.openaire.eu/images/OpenAIRE_branding/Logo_Horizontal.png">https://www.openaire.eu/images/OpenAIRE_branding/Logo_Horizontal.png</a> | M |
| <b>Provider Contact Name</b>      | contactName           | Name of the main contact person of the service/resource provider.      | String (max 20) |   | M |
| <b>Provider Contact Email</b>     | contactEmail          | Email of the main contact person of the service/resource provider.     | Email           |   | M |
| <b>Provider Contact Telephone</b> | contactTel            | Telephone of the main contact person of the service/resource provider. | String (max 20) |   | M |
| <b>Provider Users</b>             | Users{ } <sup>6</sup> | The list of users administering the provider account                   |                 |   | M |

<sup>6</sup> <sup>6</sup> See resource description in Table 47

Table 46: Funder Resource

| Funder Attribute | Attribute name in API | Description   | Type      | Example Value   | <u>M</u> andatory \ <u>O</u> ptional |
|------------------|-----------------------|---|-----------|---|--------------------------------------|
| Funder ID        | id                    | Global unique and persistent identifier of the Funder | Free Text | ec  | M                                    |
| Funder Name      | name                  | Brief and descriptive name of funder                  | Free Text | European Commision  | M                                    |
| Funder Logo      | Logo                  | Link to the logo/visual identity of the funder        | URL       | <a href="https://ec.europa.eu/commission/sites/beta-political/themes/europa/images/svg/logo/logo--en.svg">https://ec.europa.eu/commission/sites/beta-political/themes/europa/images/svg/logo/logo--en.svg</a> | O                                    |

Table 47: Authenticated User Resource

| User Attribute | Attribute name in API | Description  | Type      | Example Value  | <u>M</u> andatory \ <u>O</u> ptional |
|----------------|-----------------------|--|-----------|--|--------------------------------------|
| User ID        | id                    | Global unique and persistent identifier of the user, provided by the AAI service | Free Text | alice@eic  | M                                    |
| User Name      | name                  | The name of the user   | Free Text | Alice  | M                                    |
| User Surname   | Surname               | The surname of the user  | Free Text | Robertson  | M                                    |
| User Email     | email                 | The email of the user  | Free Text | <a href="mailto:alice@robertson.com">alice@robertson.com</a> | M                                    |

## Appendix 2: Guidelines for the use of the API by a Service Provider

This section provides guidelines – in the form of a step-by-step example – for a service provider to start using the API for adding and updating service information in the eInfraCentral Gateway.

### Add a new Service.

- **Step 1.** Start from the documentation, i.e. <http://www.einfracentral.eu/developers> and <http://www.einfracentral.eu/openapi>
- **Step 2.** Register your organisation in eInfraCentral
  - Signup as a new Service provider following the [registration process](#)
- **Step 3.** Export one or more services to JSON according to Service Description Template.
  - See a reference json in Section 3.1.
- **Step 4.** Using the [AAI service](#) of the EOSC Portal, retrieve a new API token
  - Any subsequent request to the API using that cookie, is authenticated.
- **Step 5.** Validate the service description
  - Validate that the service is well formed by calling the **POST/service/validate** method.
- **Step 6.** Make a POST \ Service call to add the new service in the catalogue.
  - Upon success you get a new service ID.
- **Step 7.** The new service is registered and visible in the eInfraCentral Gateway. You may validate the addition by calling the **GET/service/{id}** and providing the service id.

### Update an existing Service

- **Repeat Steps 1-5.**
- **Step 6.** Make a **PUT/Service** call to update the service in the catalogue.
  - upon success you get the service ID
- **Step 7.** The service is updated and visible in the eInfraCentral Gateway. You may validate the update by calling the **GET/service/{id}** and providing the service id.