



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**

Deliverable D3.3

Guidelines for schema representation and APIs



eInfraCentral

Type (Distribution Level):	Public
Deliverable Leader:	UOA
Authors:	George Papastefanatos, Akrivi Katifori, Natalia Manola, Stefania Martziou, Antonis Lempesis, Panagiotis Labropoulos (UoA)
Due Date:	30.04.2018
Actual Submission Date:	21.05.2018
Version:	1.0

Abstract: This document specifies the guidelines for the service-related representation of eInfraCentral information, and the REST API specifications 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
28.03.2018	0.1	George Papastefanatos	ToC distributed
02.04.2018	0.1	J. Sanchez, N. Vogiatzis (JNP), Jelena Angelis (EFIS),	Comments on ToC received and integrated
27.04.2018	0.1	George Papastefanatos	1 st Draft prepared
27.04.2018	0.1	George Papastefanatos, Akrivi Katifori, Natalia Manola, Stefania Martziou, Antonis Lempesis, Panagiotis Labropoulos (UoA)	Final Draft sent for review
06.05.2018	0.1	J. Sanchez, N. Vogiatzis (JNP)	QAT leaders send final comments to the deliverable editor
18.05.2018	0.1	Sergio Andreozzi (EGI), Shaun Cairns (GEANT), Rob Baxter (EUDAT)	Feedback collected by the reviewers
21.05.2018	0.2	George Papastefanatos	Deliverable editor integrates comments
21.05.2018	0.2	J. Angelis	Final review of the deliverable
21.05.2018	1.0	Alasdair Reid (EFIS)	Final document approved and submitted

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 harvesting 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 deliverable (D3.3 Guidelines for schema representation and APIs) specifies the guidelines for the service-related representation of eInfraCentral information and the REST API specifications 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, 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 (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 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 **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 a living document, generated using frameworks compliant with OpenAPI Specification 2.0. **API v1.0** is available at the beta site¹. After the public release launch this will be available at <http://www.einfracentral.eu/developers>.

¹ <http://beta.einfracentral.eu/developers>

Table of Contents

Document Revision History	2
Executive Summary	3
Acronyms	7
Short Names of Partner Organisations	8
1 Introduction	9
2 Use Cases for the use of eInfraCentral APIs	12
2.1 Provisioning of eInfraCentral Catalogue information	12
2.1.1 Use Cases for retrieving service-related information	13
2.1.2 Use Cases for retrieving provider-related information	15
2.1.3 Use Cases for retrieving indicators - and analytics - related information	16
2.1.4 Use Cases for retrieving vocabulary-related information	17
2.2 Collection of eInfraCentral service information	18
3 Representation of eInfraCentral Resources	21
3.1 eInfraCentral Resource Model	21
3.2 eInfraCentral Resource Vocabularies	26
4 eInfraCentral API specifications	29
4.1 eInfraCentral API technology stack	29
4.2 eInfraCentral API overview	29
4.2.1 Service Controller	32
4.2.2 Service Provider Controller	38
4.2.3 Indicator Controller	40
4.2.4 Measurement Controller	41
4.2.5 Statistics Controller	43
4.2.6 User Controller	45
4.2.7 Vocabulary Controller	46
5 Conclusions	48
Appendix 1: Detailed Description of the eInfraCentral resource model	49
Appendix 2: Guidelines for the use of the API by a Service Provider	63

List of Figures

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

List of Tables

Table 1: The service resource represented in JSON	22
Table 2: The service provider resource represented in JSON	24
Table 3: The service manager resource represented in JSON	24
Table 4: The Performance Indicator resource represented in JSON	24
Table 5: The Indicator Measurement resource represented in JSON	25
Table 6: The User resource represented in JSON	25
Table 7: The statistics resource represented in JSON	26
Table 8: Overview of the eInfraCentral API methods	31
Table 9: POST /Service -Add a new service.....	32
Table 10: POST /Service/validate -Validates the service without actually changing the catalogue.....	32
Table 11: PUT /Service -Update a service.....	33
Table 12: GET /service/{id} - Get a service based on the service ID	33
Table 13: GET /service/all - Filter a list of services based on a set of filters or get a list of all services in the eInfraCentral Catalogue.....	34
Table 14: GET /service/by/{field}- Get all services in the catalogue organized by a field	34
Table 15: GET /service/byID/{ids} -Get a list of services based on a set of IDs	35
Table 16: GET /service/versions - Gets versions of a specific service.....	36
Table 17: PUT /provider -Update a service provider	38
Table 18: GET/provider/{id} - Get a specific provider.....	38
Table 19: GET/provider/all - Get all providers in the catalogue	38
Table 20: GET/provider/{id}/services - Get all services of a provider.....	39
Table 21: POST /indicator -Add a new indicator.....	40
Table 22: PUT /indicator -Updates an existing indicator	40
Table 23: GET /indicator -Gets an existing indicator	41
Table 24: POST /measurement -Adds a new measurement.....	41
Table 25: PUT /measurement -Updates an existing measurement.....	42
Table 26: GET /measurement -Gets an existing measurement.....	42
Table 27: GET /stats/provider/favourites/{id} -Gets total favorites for all services of a provider	43
Table 28: GET /stats/provider/orders/{id} -Gets total orders for all services of a provider.....	43
Table 29: GET /stats/provider/ratings/{id} -Gets average ratings for all services of a provider	44
Table 30: GET /stats/provider/visits/{id} -Gets total visits for all services of a provider	44
Table 31: GET /stats/provider/visitation/{id} -Gets % of visits for all services of a provider	45
Table 32: API/login -Login to the eInfraCentral API.....	45
Table 33: GET /vocabulary/all -Gets vocabulary terms	46
Table 34: GET /vocabulary/getEU -Gets the list of country codes belonging to EU code.	47
Table 35: Service resource	49
Table 36: Indicator resource	58
Table 37: Indicator Measurement Resource	59
Table 38: Service Provider Resource.....	59
Table 39: Service Manager Resource.....	60
Table 40: Authenticated User Resource	60
Table 41: User Events.....	61
Table 42: Statistics collected by eInfraCentral.....	62

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.

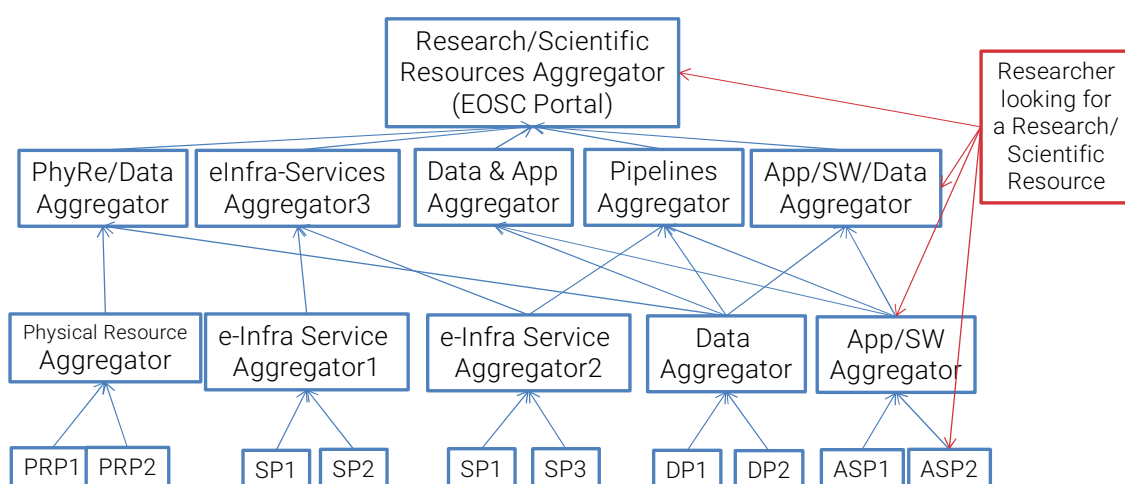


Figure 1: A model for the development of the Catalogue of Research/Scientific Resources

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 (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.

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.

This deliverable provides the specifications of the APIs (v1.0) of the eInfraCentral Gateway. These specifications describe the interoperability interfaces for the automatic provisioning and synchronisation of information regarding a service from e-Infrastructures to eInfraCentral Gateway as well as the provisioning of catalogue information towards third party systems to enable the development of added value services on top of this catalogue. This is depicted in Figure 2.

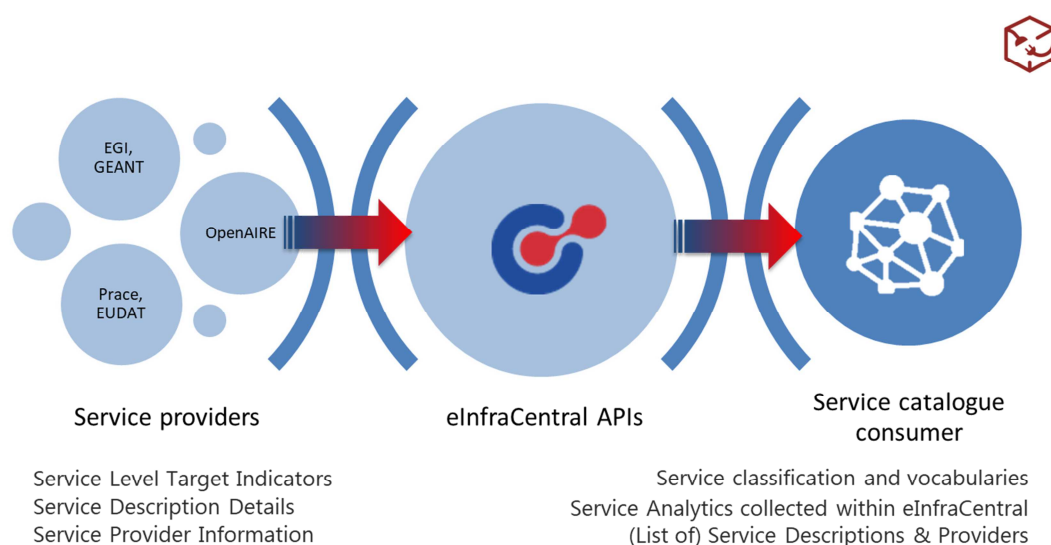


Figure 2: An overview of the eInfraCentral APIs

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.

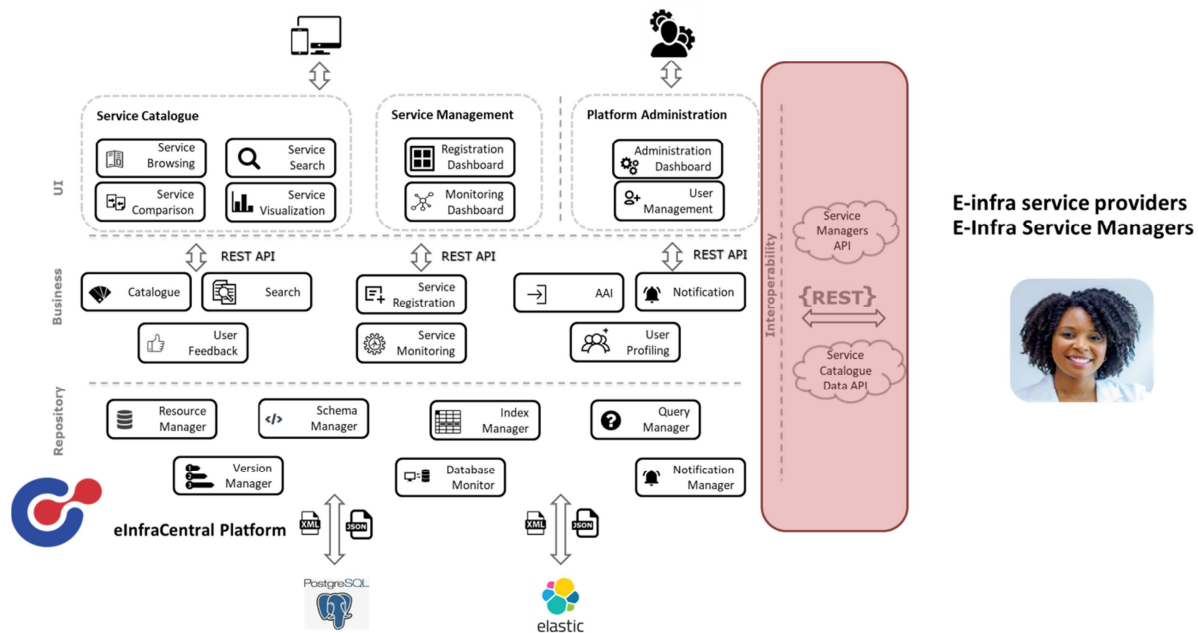


Figure 3: An architecture of the eInfraCentral Gateway with the interoperability layer highlighted

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.

The current eInfraCentral Gateway release as well as any following releases are available at the beta site². Following a public launch, the eInfraCentral Gateway will be the main entry point for www.eInfraCentral.eu.

The complete [eInfraCentral API documentation](http://www.eInfraCentral.eu) is a living document, generated using frameworks compliant with OpenAPI Specification 2.0. API v1.0 and is also available at the beta site; similarly after the public launch will be available at <http://www.eInfraCentral.eu/developers>.

The rest of the deliverable 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 D3.2); 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.

² beta.eInfraCentral.eu

2 Use Cases for the use of 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 **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 eInfraCentral 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 eInfraCentral 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 eInfraCentral catalogue based on its ID. Each service in the eInfraCentral 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.

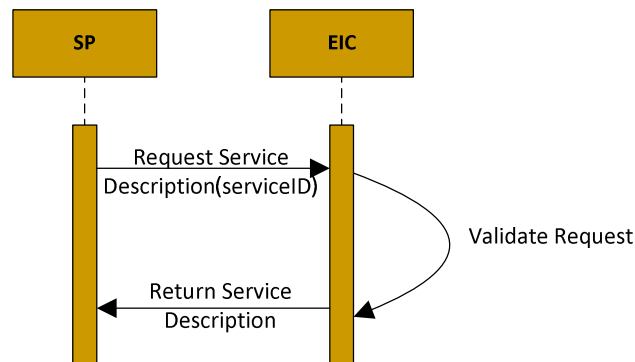


Figure 4: UC1.1 – Get a description of a service

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.

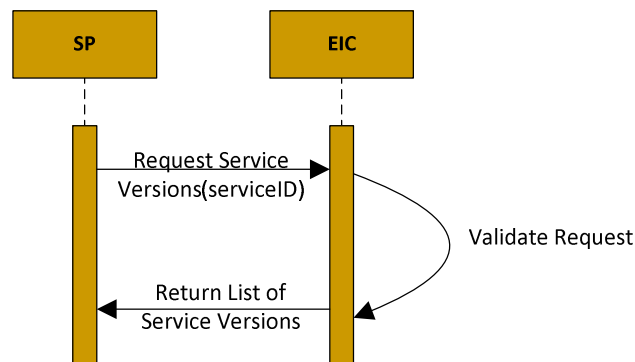


Figure 5: UC1.2 – Get all versions of a service

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.

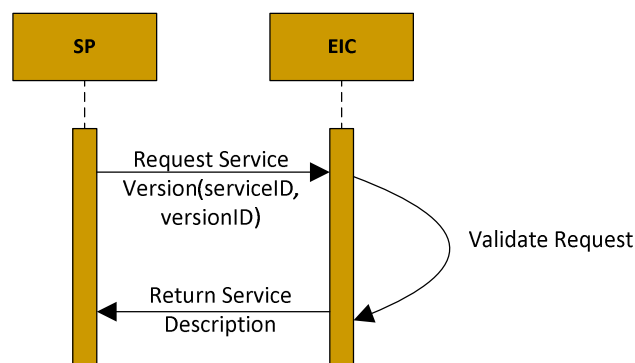


Figure 6: UC1.3 – Get a past version of a service

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.

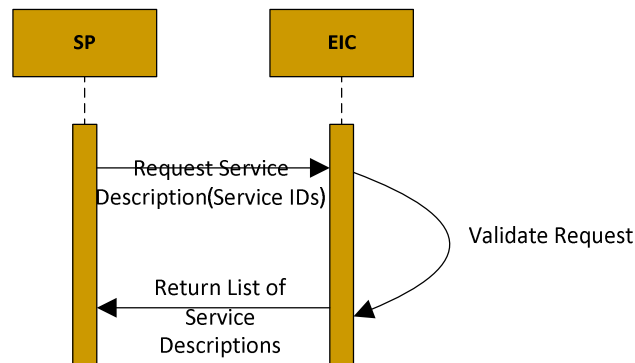


Figure 7: UC1.4 – Get multiple services based on a set of Service IDs

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.

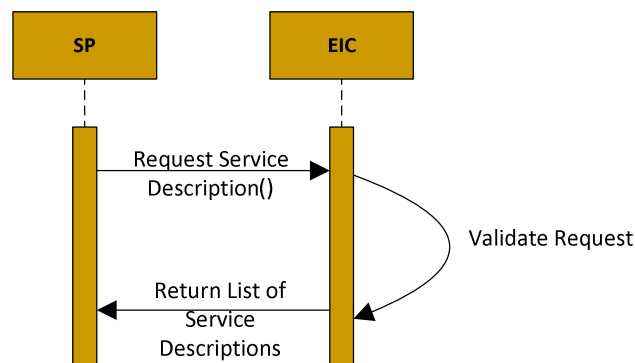


Figure 8: UC1.5 – Get all services in the eInfraCentral Catalogue

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.

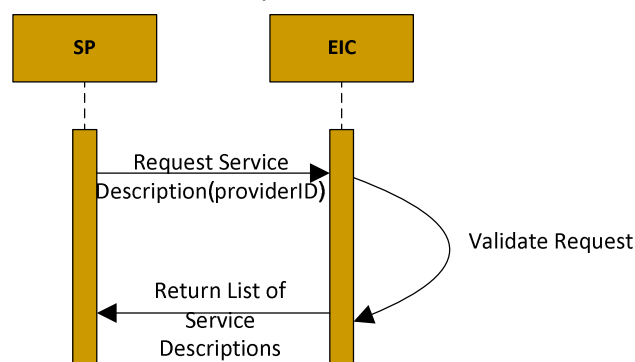


Figure 9: UC1.6 – Get a list of services provided by a service provider

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.

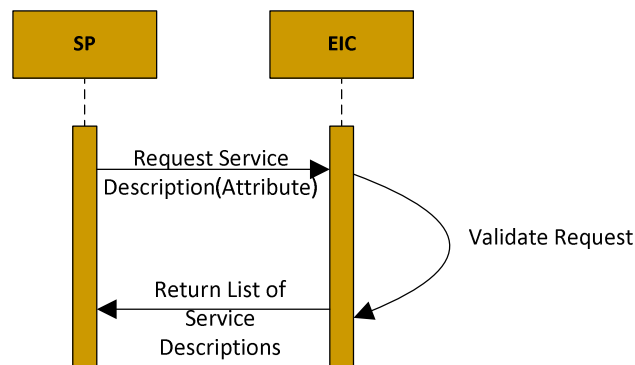


Figure 10: UC1.7 – Get all services in the catalogue organised by a service attribute

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.

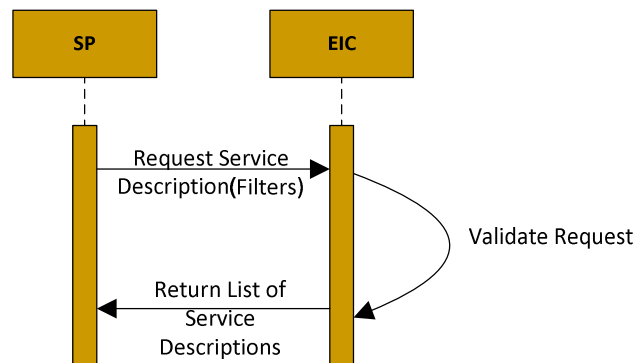


Figure 11: UC1.7 – Get a list of services based on filters

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.

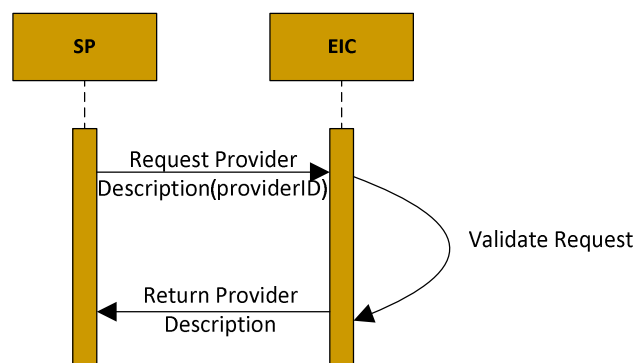


Figure 12: UC2.1 – Get information about a specific service provider

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.

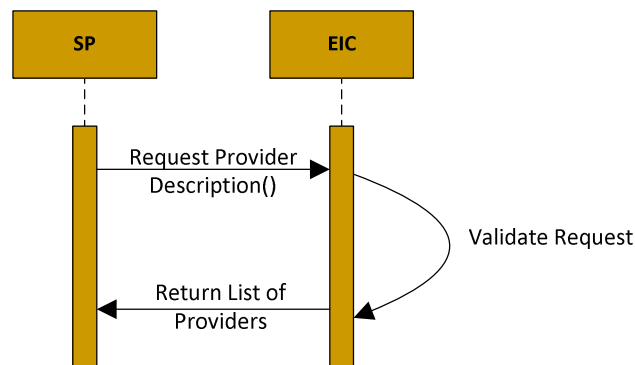


Figure 13: UC2.2 – Get the list of service providers

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 analytics 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).

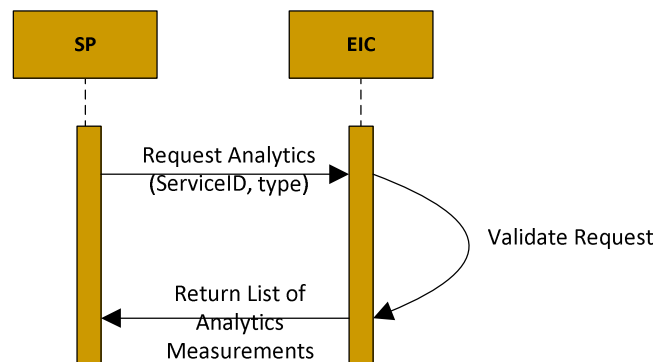


Figure 14: UC3.1 – Get analytics for a service

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.

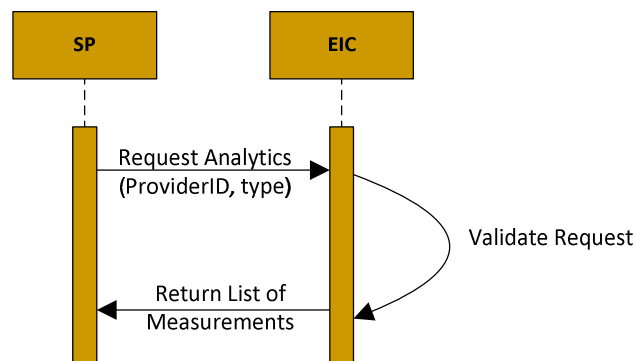


Figure 15: UC3.2 – Get analytics for a service provider

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.

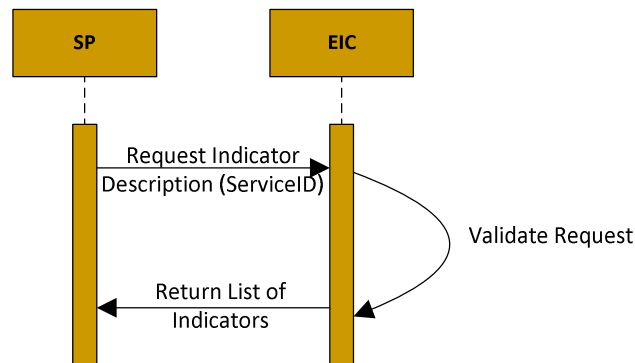


Figure 16: UC3.3 – Get a list of all indicators associated with a service

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.

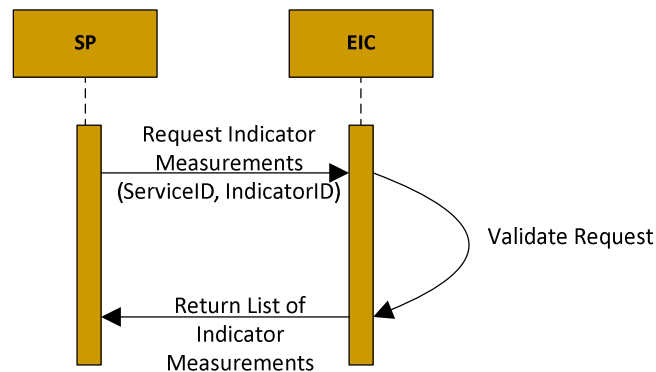


Figure 17: UC3.4 – Get the values of an indicator of a service

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.

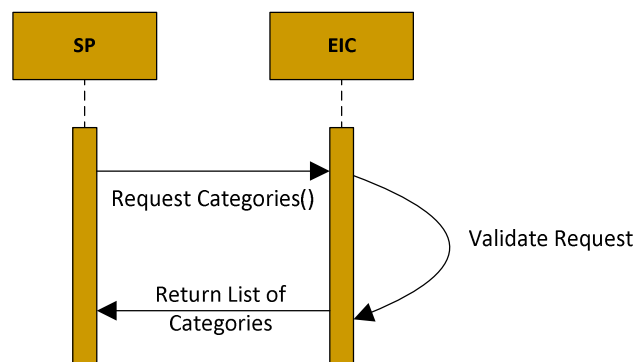


Figure 18: UC10 – Get a list of service categories

2.2 Collection of eInfraCentral service information

D4.2 initially introduced a mechanism and the API methods for the automatic collection of information in the eInfraCentral Gateway. It addresses the need of service providers to add or update service related information through a REST API.

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 makes all updates available at a remote endpoint from where the eInfraCentral monitoring mechanism 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.

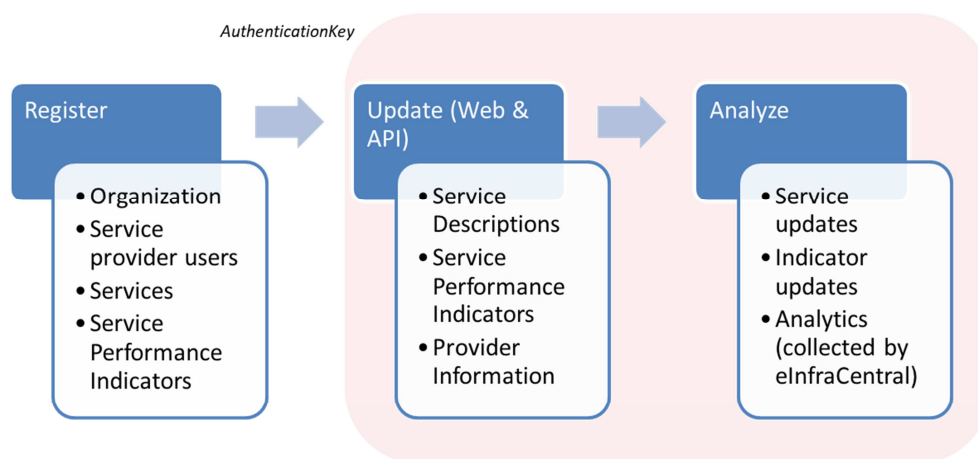


Figure 19: A workflow for the update of eInfraCentral information by a service provider

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 19. For completeness reasons, in this section, we present and refine these API calls according to the latest API version.

UC5.1. 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.

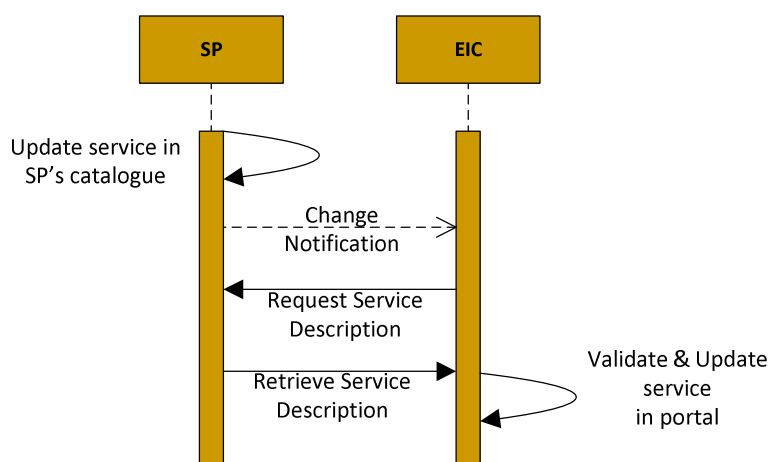


Figure 20: UC5.1 – Update of information of an existing service via a PULL method

UC5.2. 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.

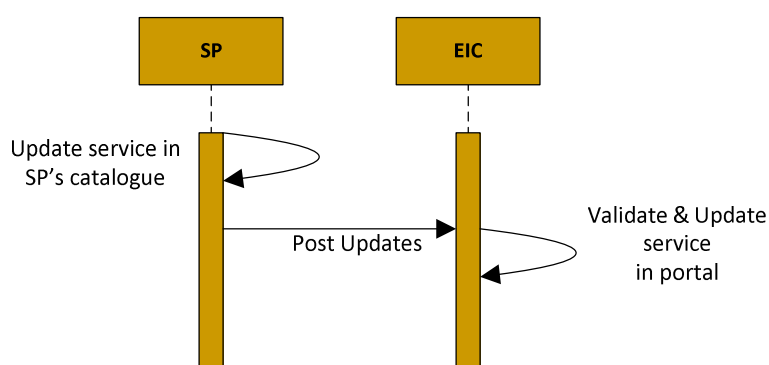


Figure 21: UC5.2 – Update of information of an existing service via a PUSH method

UC5.3. 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. eInfraCentral returns a success or a message with the error of the validation.

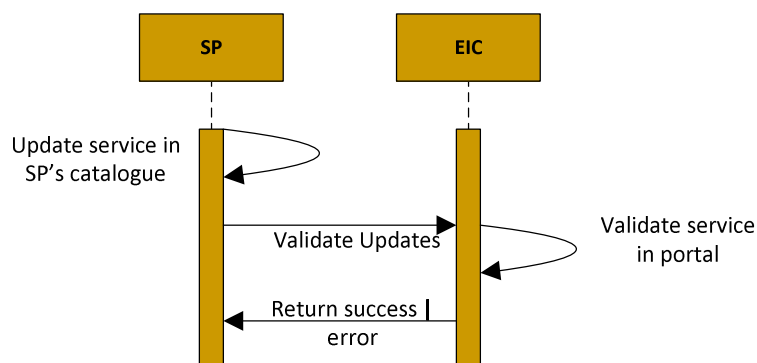


Figure 22: UC5.3 – Validate an update of a service via a push method

UC5.4. 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 eInfraCentral, which in turn validates and updates the description in the catalogue.

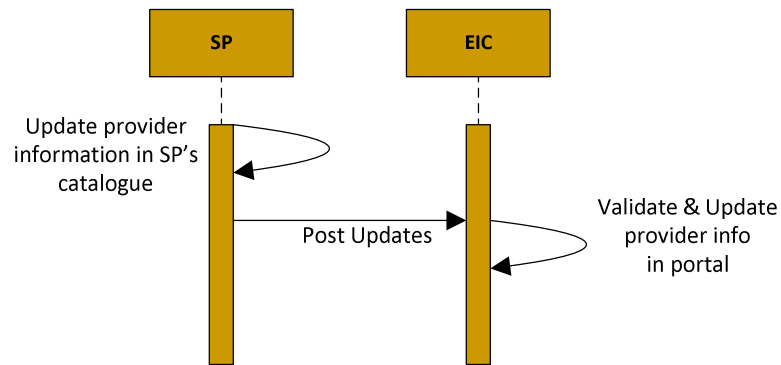


Figure 23: UC5.4 – Update of information of a service provider via a push method

UC5.5. 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 eInfraCentral 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.

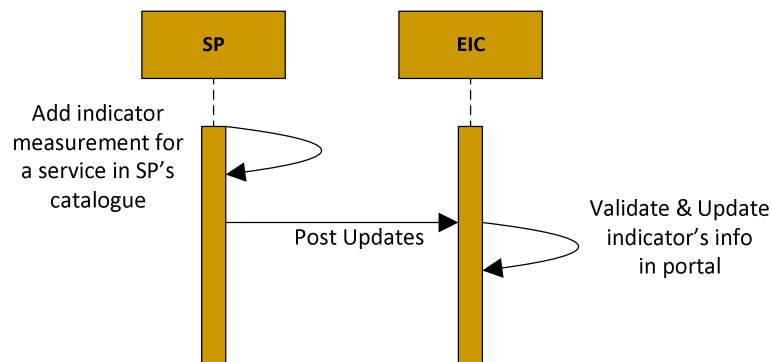


Figure 24: UC5.5 – Addition of a new measurement for a performance indicator via a push method

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 25.

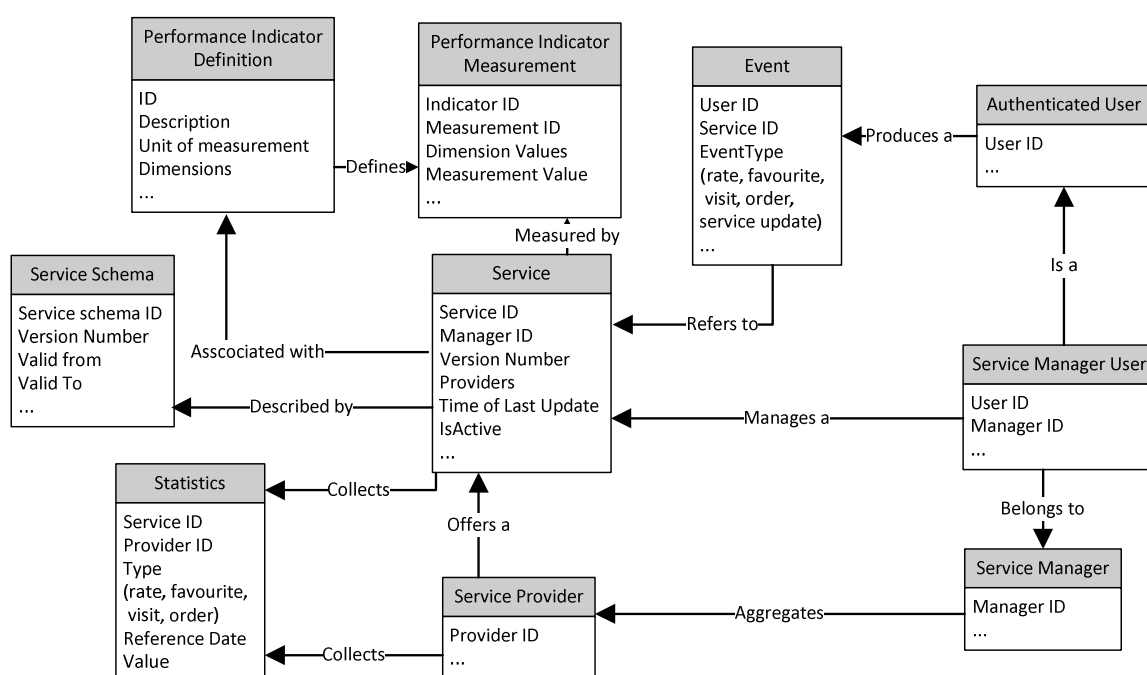


Figure 25: An overview of the eInfraCentral Gateway resource model

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 service description template (latest approved version v1.10 as described in D3.2). A service resource should comply with a service schema.

A service resource is associated with one or more indicators, which are used for defining indicator measurements. An indicator measurement holds the measurement value for the reference dimension values, e.g. a time period 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 manager is an organisation which is responsible to manage (add, update) services on behalf of multiple service providers.

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 manager meaning they would be authorised to manage and monitor the services of that service manager.

Finally, a service and a provider resource are characterised by a set of usage statistics collected by the eInfraCentral Gateway, such as the number of visits on a service page, number of orders on the service provider page, number of favourites, average ratings, etc.

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 which is described in D3.2 and is also available in <https://www.gitbook.com/@jnp>.

Sample representations of the resources used by the eInfraCentral Gateway for the exchange of information through the APIs are presented as JSON objects in the following tables. The full schema of the eInfraCentral model is also available as XSD in <https://github.com/eInfraCentral/eic-data/blob/master/dropbox/schema1.xsd>.

Table 1: The service resource represented in JSON

Service Resource

```
service: {
  "id": "string",
  "url": "string",
  "name": "string",
  "tagline": "string",
  "description": "string",
  "options": "string",
  "targetUsers": "string",
  "userValue": "string",
  "userBase": "string",
  "symbol": "string",
  "multimediaURL": "string",
  "providers": [
    "string"
  ],
  "version": "string",
  "lastUpdate": "timestamp",
  "changeLog": "string",
  "validFor": "timestamp",
  "lifeCycleStatus": "string",
  "trl": "string",
  "category": "string",
  "subcategory": "string",
  "places": [
    "string",
  ],
  "languages": [
    "string"
  ],
  "tags": [
    "string",
  ],
  "requiredServices": [
    "string"
  ],
  "relatedServices": [
    "string"
  ],
  "order": "string",
```

```

"helpdesk": "string",
"userManual": "string",
"trainingInformation": "string",
"feedback": "string",
"price": "string",
"serviceLevelAgreement": "string",
"termsOfUse": [
  "string"
],
"funding": "string"
}

```

Service Resource Example for Service EGI High-Throughput Compute

```

{
  "id": "1.03",
  "url": "https://www.egi.eu/services/high-throughput-compute",
  "name": "EGI High-Throughput Compute",
  "tagline": "Execute thousands of computational tasks to analyse large datasets",
  "description": "With High-Throughput Compute you can run computational jobs at scale on the EGI infrastructure. It allows you to analyse large datasets and execute thousands of parallel computing tasks. High-Throughput Compute is provided by a distributed network of computing centres, accessible via a standard interface and membership of a virtual organisation. EGI offers more than 650,000 cores of installed capacity, supporting about 1.6 million computing jobs per day. This service supports research and innovation at all scales: from individuals to large collaborations. Main characteristics of the service: access to high-quality computing resources, integrated monitoring and accounting tools to provide information about the availability and resource consumption, workload and data management tools to manage all computational tasks, large amounts of processing capacity over long periods of time, faster results for your research.- Shared resources among users, enabling collaborative research",
  "options": null,
  "targetUsers": "Research organisations, SME/Industry, Researchers, Innovators",
  "userValue": "Enable collaborative research",
  "userBase": null,
  "symbol": "https://www.egi.eu/wp-content/uploads/2016/08/icon-HTC-Compute.jpg",
  "multimediaURL": null,
  "providers": [
    "egi"
  ],
  "version": "0",
  "lastUpdate": null,
  "changeLog": null,
  "validFor": null,
  "lifeCycleStatus": "Production",
  "trl": "9",
  "category": "Compute",
  "subcategory": "job execution",
  "places": [
    "EU"
  ],
  "languages": [
    "en"
  ],
  "tags": [
    "HTC"
  ],
  "requiredServices": null,

```

```

"relatedServices":null,
"order":"https://marketplace.egi.eu/32-high-throughput-compute",
"helpdesk":"http://helpdesk.egi.eu/",
"userManual":null,
"trainingInformation":"https://www.egi.eu/egi-trainings/",
"feedback":"https://www.egi.eu/contact/",
"price":"https://www.egi.eu/access-policy/",
"serviceLevelAgreement":"https://documents.egi.eu/document/2733",
"termsOfUse":[
  "https://documents.egi.eu/document/2623"
],
"funding":"Development: EC H2020 projects (primarily) and national
projects/open source projects. Operations: capacity supported by national
funding, federation and support co-funded between national funding and EC
H2020 projects"
}

```

Table 2: The service provider resource represented in JSON

Service Provider Resource
<pre> provider: { "id": "string", "name": "string", "contactInformation": "string" } </pre>
Service Provider Resource Example for EGI
<pre> provider: { "id": "egi", "name": "EGI", "contactInformation": "contact@egi.eu" } </pre>

Table 3: The service manager resource represented in JSON

Service Manager Resource
<pre> manager: { "id": "string", "name": "string", "contactInformation": "string", "providers": ["string"] } </pre>
Service Manager Resource Example for EOSC-hub
<pre> manager: { "id": "eoschub", "name": "EOSC-hub", "contactInformation": "contact@eoschub.eu", "providers": ["egi", "eudat"] } </pre>

Table 4: The Performance Indicator resource represented in JSON

Indicator Resource
<pre> indicator: { "id": "string", "description": "string", "unit": "string", "dimensions": ["string"] } </pre>
Indicator Resource Example for Service Availability Indicator

<pre> indicator: "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": "PCT", "dimensions": ["TIME PERIOD"] </pre>	{
Where Unit of measurement takes a value from PCT NUM BOOL (Percentage, Numeric, Boolean) and Dimensions can be TIME PERIOD and/or LOCATION	

Table 5: The Indicator Measurement resource represented in JSON

Indicator Measurement Resource	
<pre> measurement: "id": "string", "serviceId": "string", "IndicatorId": "string", "timePeriod": "string", "locations": ["string"], "Value": "string" </pre>	{
Measurement Resource Example for Service Availability Indicator	
<pre> measurement: "measurementId": "123456", "serviceId": "openaire", "IndicatorId": "availability", "timePeriod": "01/01/2018-31/01/2018" "locations": ["WW"] "Value": "99.99" </pre>	{

Table 6: The User resource represented in JSON

User Resource	
<pre> user: "id": "string", "name": "string", "surname": "string", "email": "string", "password": "string", "joinDate": "string", "organization": "string", "manager": "string" </pre>	{
User Resource Example	
<pre> user: "id": "alice@eic", "name": "Alice", "surname": "Robertson", "email": "alice@robertson", "password": "changeMe", "joinDate": "01/02/2018", "organization": "My organization", "manager": "egi" </pre>	{

Table 7: The statistics resource represented in JSON

Statistics Resource	
<pre>stat: "service id": "string", "provider id": "string", "type": "string", "stats": [{"date": "string", "value": "string"}]</pre>	}
Statistics Resource Example for the number of visits for a service	
<pre>stat: "service id": "1.03", "provider id": "egi", "type": "visits", "stats": { "2018-03-29": 85, "2018-03-28": 69, "2018-04-18": 70, "2018-04-19": 59, "2018-04-12": 66, "2018-04-13": 79, "2018-04-16": 65, "2018-04-17": 66, "2018-04-14": 79, "2018-03-27": 71, "2018-04-15": 83, "2018-04-20": 85]</pre>	}

3.2 eInfraCentral Resource Vocabularies

The eInfraCentral 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:

- **Category – Sub Category.** They follow a parent-child relationship. As such a value for the subcategory is valid if the corresponding top-level category is selected.
 1. Networking
 - Direct Connect
 - Virtual Network
 - Load Balancer
 - Application Gateway
 - VPN Gateway
 - Content Delivery Network
 - Traffic Manager
 - API Gateway
 - Other
 2. Compute
 - Virtual Machine Management
 - Container Management
 - Batch Processing
 - Serverless Applications Repository
 - Load Balancing
 - Other
 3. Storage

- Blob
 - File
 - Queue
 - Disk
 - Archive
 - Backup
 - Synchronised
 - Replicated
 - Recovery
 - Other
4. Data
- Mining
 - Access
 - Management
 - Transfer Management
 - Registration
 - Persistent Identifier
 - Interlinking
 - Publishing
 - Discovery
 - Anonymisation
 - Preservation
 - Brokering
 - Annotation
 - Other
5. Software
- Platform
 - Tool
 - Component
 - Algorithm
 - Developer
 - Other
6. Application
- Porting
 - Scaling
 - Education
 - Productivity
 - Business
 - Developer
 - Other
7. Security
- Authentication and Authorisation
 - Attacks protection
 - Coordination
 - Tools
 - Other
8. Analytics
- Business Analytics

- Web Analytics
- Learning Analytics
- Predictive Analytics
- Machine Learning
- Other
- 9. Operations
 - Accounting
 - Helpdesk
 - Monitoring
 - Analysis
 - Configuration Management
 - Tools
 - Coordination
 - Order Management
 - Other
- 10. Training
 - Online Courses
 - Open Registration Courses
 - In-house Courses
 - Tools
 - Platform
 - Other
- 11. Consulting
 - Audit and Assessment of IT Service Management
 - Audit and Assessment of Information Security
 - Other
- 12. Other
- **Lifecycle Status**
 - alpha
 - beta
 - production
- **Technology Readiness Levels**
 - TRL7
 - TRL8
 - TRL9
- **Places:** Places refer to countries worldwide. eInfraCentral follows the ISO 639-1³ 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 standard⁴ for the 2-letter codes for representing the languages.

³ <https://www.iso.org/iso-639-language-codes.html>

⁴ <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://beta.einfracentral.eu/developers) is a living document, generated using frameworks compliant with OpenAPI Specification 2.0. **API v1.0** is available in <http://beta.einfracentral.eu/developers>. After the public release launch it will be available at <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 3. 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 completely update a resource identified by its resource URI
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 An overview of the eInfraCentral API

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 all versions of a 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,
- **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,
- **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,
- **Statistics Controller:** It offers the functionality for retrieving statistics collected in eInfraCentral Gateway, such as:
 - retrieving service orders per day for a service,
 - 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 service orders 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 token⁵ (JWT) for authenticating further requests in the eInfraCentral Gateway.
- **Vocabulary Controller:** It offers the functionality to retrieve the vocabularies used in the eInfraCentral Gateway, such as:
 - Categories and subcategories,
 - TRL values,
 - Lifecycle Status values,
 - Country Codes,
 - Language Codes.

⁵ <https://jwt.io/>

An overview of the methods offered by each controller is shown in Table 8. Note that the service, provider, indicator, measurement and user controllers offer functionality for adding (POST) and updating (PUT) information in the eInfraCentral Gateway. These methods have been first introduced in D4.2 as part of the API of the eInfraCentral monitoring services and are now integrated in the complete API offered by eInfraCentral. For reasons of completeness, we include their descriptions and functionality in this section, as well.

Table 8: An overview of the eInfraCentral API methods

Controller	Type	Method Name	Description
Service	POST	/service	Adds, i.e. registers, a new service resource in the eInfraCentral Gateway
	PUT	/service	Updates an existing service resource description
	POST	/service/validate	Validates the registration or update of a service resource without it actually being added/modified in the registry
	GET	/service/{id}	Gets the most current version of a specific service providing the service ID
	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
	GET	/service/by/{field}	Gets all services in the catalogue organised by a field, e.g. get service organized in categories
	GET	/service/byID/{ids}	Gets a list of services based on a set of IDs
	GET	/service/versions	Gets a past version of a specific service providing the service ID and a version identifier or get a list of all versions of a service
Indicator	POST	/indicator	Adds the given indicator
	PUT	/indicator	Updates an existing indicator with the given idID
	GET	/indicator/{id}	Gets the indicator given its ID
Measurement	POST	/measurement	Adds the given indicator measurement
	PUT	/measurement	Updates an existing measurement with the given ID
	GET	/measurement/{id}	Gets the measurement given its id
Provider	PUT	/provider	Updates provider info
	GET	/provider/all	Gets a list of all service providers in the catalogue
	GET	/provider/{id}	Gets provider's data given the provider id
	GET	/provider/{id}/services	Gets a list of services offered by a provider
Statistics	GET	/stats/provider/favourites/{id}	Gets aggregate favourites per day for all services offered by a provider
	GET	/stats/provider/orders/{id}	Gets visits in the provider's site for all of the provider's services
	GET	/stats/provider/ratings/{id}	Gets average ratings per day for all services offered by a provider
	GET	/stats/provider/visitation/{id}	Gets percentage of visits for all services offered by a provider

	GET	/stats/provider/visits/{id}	Gets aggregate visits per day for all services offered by a provider
	GET	/stats/service/favourites/{id}	Gets favourites per day for a service
	GET	/stats/service/orders/{id}	Gets visits in the provider's site of the service
	GET	/stats/service/ratings/{id}	Gets average ratings per day for a service
	GET	/stats/service/visits/{id}	Gets visits per day for a service
User	POST	/user/login	Issues a new token given the username and the password
Vocabulary	GET	/vocabulary/all	Gets vocabulary terms used in eInfraCentral, such as categories \ sub categories, etc.
	GET	/vocabulary/getEU	Gets the list of EU countries as modelled in eInfraCentral.

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 9: POST /Service – Adds a new service

Request	
POST: 8080/eic-registry/service Accept: application/json Cookie: {auth-string}	
Parameters (The full service description of the service to be added)	
<pre>service: { "id": "string", "url": "string", "name": "string", "tagline": "string", "description": "string", ... }</pre>	
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 10: POST /Service/validate – Validates the service without actually changing the catalogue

Request	
POST: 8080/eic-registry/service/validate Accept: application/json	

Cookie: {auth-string}	
Parameters (The full service description of the service to be validated)	
<pre> service: { "id": "string", "url": "string", "name": "string", "tagline": "string", "description": "string", ... }</pre>	
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 11: PUT /Service – Updates a service

Request	
PUT: 8080/eic-registry/service	
Accept: application/json	
Cookie: {auth-string}	
Parameters (The full service description of the service to be updated)	
<pre> service: { "id": "string", "url": "string", "name": "string", "tagline": "string", "description": "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 12: GET /service/{id} – Gets a service based on the service ID

Request	
GET: 8080/eic-registry/service/{id}	
Parameters	
id: the service id e.g., 8080/eic-registry/service/3.06	
Response	
Status: 200	
Content-Type: application/json	
Response Body (A service description)	

```
{
  "id": "3.06",
  "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.",
  ...
}
```

Response Messages

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

Table 13: 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

Request

GET: 8080/eic-registry/service/all

Parameters

Query: An expression with filters

From: Used for paging the results. It indicates the nth result.

Quantity: Number of results to fetch

E.g.,

- Get all services in the catalogue → 8080/eic-registry/service/all/
- Get the first 5 services of the compute category → 8080/eic-registry/service/all/query=compute&from=0&quantity=5

Response

Status: 200

Content-Type: application/json

Response Body (A list of service descriptions)

```
{  "total": 9,  "from": 0,  "to": 5,  "results": [    {      "id": "1.02",      "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 14: GET /service/by/{field} – Gets all services in the catalogue organized by a field

Request
GET: 8080/eic-registry/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 → 8080/eic-registry/service/by/category

Response

Status:

200

Content-Type:

application/json

Response Body (A list of service descriptions)

{

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

{

"id": "4.06",

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

Request
GET: 8080/eic-registry/service/byID/{ids}
Parameters
<p>ids: Comma separated service ids</p> <p>E.g.,</p> <ul style="list-style-type: none"> Get services 1.03 and 4.13 → 8080/eic-registry/service/byID/1.03,4.13
Response
<p>Status: 200</p> <p>Content-Type: application/json</p>
Response Body (A list of service descriptions)
<pre>[{ "id": "1.03",</pre>

```

"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": "4.13",
"url": "http://scholexplorer.openaire.eu",
"name": "OpenAIRE ScholExplorer",
>tagline": "The data and literature interlinking service",
...
}
]

```

Response Messages

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

Table 16: GET /service/versions – Gets versions of a specific service

Request

GET: 8080/eic-registry/service/versions

Parameters

id: A service id

version: A version identifier, gets all versions if empty

E.g.,

- Get version v2 for the service 4.02 → 8080/eic-registry/service/versions/4.02/v2
- Get all versions of service 4.02 → 8080/eic-registry/service/versions/4.02/

Response

Status: 200

Content-Type: application/json

Response Body (A list of service descriptions)

```
[
{
  "id": "4.02",
  "url": "http://api.openaire.eu",
  "name": "OpenAIRE Graph",
  "tagline": "Open, linked research ",
  ...
  "version": "v2",
  ...
}
]
```

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 17: PUT /provider – Updates a service provider

Request	
PUT: 8080/eic-registry/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 18: GET/provider/{id} – Gets a specific provider

Request	
GET: 8080/eic-registry/provider/{id}	
Parameters	
id: The provider's ID e.g.,: 8080/eic-registry/provider/geant	
Response	
Status: 200	
Content-Type: application/json	
Response Body (A service 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 19: GET/provider/all – Gets all providers in the catalogue

Request	
GET: 8080/eic-registry/provider/all	
Parameters	
None	
Response	

Status: 200
Content-Type: application/json
Response Body (A list of provider descriptions)

```
{
  "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
    }
  ]
}
```

Response Messages

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

Table 20: GET/provider/{id}/services – Gets all services of a provider

Request
GET: 8080/eic-registry/provider/{id}/services
Parameters
Id: the provider's id, e.g., <ul style="list-style-type: none"> Get all services from Eudat →: 8080/eic-registry/provider/eudat/services
Response
Status: 200 Content-Type: application/json

Response Body (A list of service descriptions)

```
[
  {
    "id": "2.04",
    "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 Indicator Controller

The following API methods manage information about Indicators.

Table 21: POST /indicator – Adds a new indicator

Request

POST: 8080/eic-registry/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 22: PUT /indicator – Updates an existing indicator

Request
PUT: 8080/eic-registry/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	Updated
401	Unauthorized
403	Forbidden
404	Not Found
500	Other

Table 23: GET /indicator – Gets an existing indicator

Request

GET: 8080/eic-registry/indicator/{id}

Parameters

id: The indicator's ID

e.g.,: 8080/eic-registry/indicator/availability

Response

Status: 200

Content-Type: application/json

Response Body (A service description)

```

{
  "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": "PCT",
  "dimensions": [
    "TIME PERIOD"
  ]
}

```

Response Messages

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

4.2.4 Measurement Controller

The following API methods manage information about measurements of Indicators.

Table 24: POST /measurement – Adds a new measurement

Request

POST: 8080/eic-registry/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 25: PUT /measurement – Updates an existing measurement**Request**

PUT: 8080/eic-registry/measurement

Accept: application/json

Cookie: {auth-string}

Parameters (The description of the indicator)

```

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	Updated
401	Unauthorized
403	Forbidden
404	Not Found
500	Other

Table 26: GET /measurement – Gets an existing measurement**Request**

GET: 8080/eic-registry/measurement/{id}

Parameters

id: The measurement's ID
 e.g.,: 8080/eic-registry/measurement/123456

Response

Status: 200

Content-Type: application/json

Response Body (A service description)

```

{

```

```

"measurementId": "123456",
"serviceId": "openaire",
"IndicatorId": "availability",
"timePeriod": "01/01/2018-31/01/2018"
"locations": [
  "WW"
],
"Value": "99.99"
}

```

Response Messages

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

4.2.5 Statistics Controller

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

Table 27: GET /stats/provider/favourites/{id} – Gets total favourites for all services of a provider

Request	
GET: 8080/eic-registry/stats/provider/favourites/{id}	
Parameters	
id: The provider's ID e.g., : 8080/eic-registry/stats/provider/favourites/eudat	
Response	
Status: 200 Content-Type: application/json	
Response Body (A list of statistics)	
<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>	
Response Messages	
HTTP Status Code	Reason
400	No Content
401	Unauthorized
403	Forbidden
404	Not Found
500	Other

Table 28: GET /stats/provider/orders/{id} – Gets total orders for all services of a provider

Request
GET: 8080/eic-registry/stats/provider/orders/{id}
Parameters
id: The provider's ID e.g.,: 8080/eic-registry/stats/orders/favourites/geant
Response
Status: 200

Content-Type: application/json	
Response Body (A list of statistics)	
<pre>{ "2018-03-29": 57, "2018-03-28": 122, "2018-04-18": 156, "2018-04-19": 244, ... }</pre>	
Response Messages	
HTTP Status Code	Reason
400	No Content
401	Unauthorized
403	Forbidden
404	Not Found
500	Other

Table 29: GET /stats/provider/ratings/{id} – Gets average ratings for all services of a provider

Request	
GET: 8080/eic-registry/stats/provider/ratings/{id}	
Parameters	
id: The provider's ID e.g.,: 8080/eic-registry/stats/provider/ratings/prace	
Response	
Status: 200	
Content-Type: application/json	
Response Body (A list of statistics)	
<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>	
Response Messages	
HTTP Status Code	Reason
400	No Content
401	Unauthorized
403	Forbidden
404	Not Found
500	Other

Table 30: GET /stats/provider/visits/{id} – Gets total visits for all services of a provider

Request	
GET: 8080/eic-registry/stats/provider/visits/{id}	
Parameters	
id: The provider's ID e.g.,: 8080/eic-registry/stats/provider/visits/openaire	
Response	
Status: 200	
Content-Type: application/json	
Response Body (A list of statistics)	
<pre>{ "2018-03-29": 1250, "2018-03-28": 2501, "2018-04-18": 3456, ... }</pre>	

```
"2018-04-19": 7890,
...
}
```

Response Messages

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

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

Table 31: GET /stats/provider/visitation/{id} – Gets % of visits for all services of a provider

Request	
GET: 8080/eic-registry/stats/provider/visitation/{id}	
Parameters	
<i>id</i> : The provider's ID e.g., : 8080/eic-registry/stats/provider/visitation/openaire	
Response	
Status: 200 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

4.2.6 User Controller

The following method implements the login and authentication of a service provider user to the eInfraCentral API. The method returns an authentication key, which is required for posting service information in the API.

Table 32: API/login – Logins to the eInfraCentral API

Request
POST: 8080/eic-registry/login Accept: application/json

Response
Status: 200 Content-Type: application/json Cookie: {auth-string}
Request Body (The user's e-mail and password)
{ "email": "{email}", "password": "{password}" }

Authentication to the services is implemented either via a JWT cookie exchange given on successful login (see aforementioned API/login), or via an access token generated by the server. The authenticated user can request the generation, deletion, and listing of API tokens via the eInfraCentral Gateway, which acts as a mediating management interface.

4.2.7 Vocabulary Controller

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

Table 33: GET /vocabulary/all – Gets vocabulary terms

Request	
GET: 8080/eic-registry/vocabulary/all	
Parameters	
<i>Query:</i> The vocabulary type (i.e., name of the attribute) for which we want to retrieve the terms.	
<i>From:</i> Used for paging the results. It indicates the nth result.	
<i>Quantity:</i> Number of results to fetch	
E.g.,	
<ul style="list-style-type: none">Get all allowed TRL values in the catalogue → 8080/eic-registry/vocabulary/all?query=trlGet the first 5 categories →8080/eic-registry/vocabulary/all?query=category&from=1&quantity=5	
Response	
Status: 200	
Content-Type: application/json	
Response Body (A list of vocabulary terms)	
<pre>{ "total": 27, "from": 1, "to": 6, "results": [{ "id": "Interoperability", "name": "Interoperability", "type": "Category" }, { "id": "Operations", "name": "Operations", "type": "Category", }, ...] }</pre>	
Response Messages	
HTTP Status Code	Reason
400	No Content
401	Unauthorized

403	Forbidden
404	Not Found
500	Other

Table 34: GET /vocabulary/getEU – Gets the list of country codes belonging to EU code

Request	
GET: 8080/eic-registry/vocabulary/getEU	
Parameters	
NONE	
Response	
Status: 200	
Content-Type: application/json	
Response Body (A list of vocabulary terms)	
<pre>["AX", "AT", "BE", "BG", "HR", "CY", "CZ", "DK", "EE", "FO", "FI", "FR", "GF", "DE", "GI", "GR", "HU", "IE", "IM", "IT", "LV", "LT", "LU", "MT", "NL", "PL", "PT", "RO", "SK", "SI", "ES", "SE", "GB"]</pre>	
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.

5 Conclusions

D3.3. provided the specifications of the APIs (API v1.0) 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 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://beta.einfracentral.eu/developers) is a living document, generated using frameworks compliant with OpenAPI Specification 2.0. **API v1.0** and is available at <http://beta.einfracentral.eu/developers> and after the public release launch this will be available at <http://www.einfracentral.eu/developers>.

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

The REST API specifications provided in this deliverable will be communicated and implemented by the five flagship e-Infrastructures (EGI, EUDAT, GENAT, OpenAIRE, PRACE), in order to automate the exchange of service-related information in the eInfraCentral Gateway.

As such, this deliverable is intended to be a living document, and any further refinements and extensions will be published online in the developers' page of the eInfraCentral Gateway.

Finally it is expected that the user assessment activities following this release of the eInfraCentral Gateway will introduce new requirements and changes that will enrich the list of specifications presented in this deliverable.

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 35: Service resource

Service Attribute	Attribute name in API	Description	Type	Example Value	Recommendation	<u>Mandatory \</u> <u>Optional</u>
Basic Service information						
Service ID	id	Global unique and persistent identifier of the service.	Formatted Text (xx.yyy)	01.03	In the above example, the first part denotes the Service Provider and the second part the unique identifier of the service within the Service Provider.	M
Service URL	url	The Uniform Resource Locator (web address) to the entry web page of the service usually hosted and maintained by the service provider.	URL	http://www.egi.eu/cloud-compute or http://cloud-compute.egi.eu	Create a unique URL for each service; provide the shortest possible alias	M
Service Provider Name	Providers[]	The organisation that manages and delivers the service and with whom the customer signs the SLA.	Multi value Free text	e.g. GEANT, PRACE, EGI, EUDAT, OpenAIRE, etc.	N/A	M

Service Name	name	Brief and descriptive name of a service as assigned by the service provider.	Free text (max 80)	B2DROP	Should be descriptive from a customer point of view and should be quite simple, such that someone non-technical is able to understand what the service is about. This field will be used in the search function to prioritise results.	M
Service Tagline	tagline	Short text, catch line or slogan which serves mainly marketing and advertising purposes.	Free text (max 100)	Store, share and access your files and their metadata on a global scale	Marketing specialists are encouraged to be engaged in order to come up with a catch phrase that can aid service dissemination and visibility.	O
Service Description	description	High-level description in fairly non-technical terms of what the service does, functionality it provides and resources it enables access to.	Free text (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 and guarantees their long-term persistence.	Description in one paragraph without bullets or rich HTML. This field is used in the search function to prioritise results.	M
Service Options	options	A high-level description of the various options or forms in which the service can be instantiated.	Free text (max 1000)	"Standard or enhanced", "Bronze, silver and gold", "10, 20, 30", "Elementary Level, Advanced Level, Expert Level"	Non-exhaustive description in one paragraph without bullets or rich HTML.	O

Target Users	targetUsers	Type of users/customers allowed to commission/benefit from the service.	Free text(max 1000)	Research Organisations, Industry, SMEs, Researchers, Scientists, Funders, Policy Makers, Service Providers, Data Providers, etc.	Target users should be defined as accurately and as inclusively as possible at the same time.	O
User Value	userValue	Description of the benefit delivered to a customer/user by the service.	Free text (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.	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).	O
User Base	userBase	List of customers, communities, users, etc. using the service.	Free text (max 1000)	900+ data providers in Europe with different compatibility levels. Adoption in Latin America (LaReferencia), Mexico and Japan.	Description should be as quantified as possible.	O
Service Symbol	symbol	The Uniform Resource Locator (web address) to the logo/visual identity of the service.	URL	http://www.egi.eu/cloud-compute/Symbol.jpg	Provide the shortest possible alias.	M
Service Multimedia	multimediaUrl	The Uniform Resource Locator (web address) to the multimedia material of the service (screenshots or videos).	URL	http://www.egi.eu/cloud-compute/Multimedia/	Provide the shortest possible alias.	O

Service Classification Information						
Service Version	version	Inform about the service version that is in force.	Free text (max 10)	3.1	Only stable releases should be referenced.	M
Service Last Update	lastUpdate	The date of the latest update of the service.	Date (dd/mm/yyyy)	28/02/2018	The date should refer to when the updated version became available to users, not when it was developed or released internally in the SP.	M
Service Change Log	changeLog	A log of the service features added in the last and previous versions.	Free text (max 1000)	Upgrade of user interface. Correction of minor bugs.	Short sentences for each feature should be used.	O
Service Valid for	validFor	The date up to which the service description is valid.	Date (dd/mm/yyyy)	03/12/2018	The expiring date should imply validity period until 23:59:59. In case it is earlier, the previous day should be used in the date.	O
Service Life Cycle Status	lifeCycleStatus	Used to tag the service to the full service cycle.	List of values ⁶	Service providers may use this attribute for other internal service management purposes.	N/A	M
Service TRL	trl	Used to tag the service to the Technology Readiness Level, a method of estimating technology maturity of critical technology elements. TRL are	List of values ⁷	For the eInfraCentral catalogue, allowed values are: TRL7, TRL8, TRL9.	https://ec.europa.eu/research/participants/data/ref/h2020/wp/2014_2015/annexes/h2020-wp1415-annex-g-trl_en.pdf	M

⁶alpha, beta, production.

⁷TRL7, TRL8, TRL9

		based on a scale from 1 to 9 with 9 being the most mature technology.				
Service Category	category	A named group of services that offer access to the same type of resource that is of interest to a customer/user.	List of values ⁸	Networking	N/A	M
Service Subcategory	subcategory	Type/Subcategory of service within a category	List of values ⁹	Direct Connect	N/A	M
Service Place	places[]	Regions/Countries Availability	Multi value ¹⁰	Global	N/A	M
Service Language	languages[]	Languages of the User interface	Multi value ¹¹	English	N/A	M
Service Tags	tags[]	Attribute to facilitate searching based on keywords.	Free text \ Multiple values	AIA, Security	This field will be used in the search function to prioritise results.	O
Required Services	requiredService[]	Other services that are required with this service.	Service IDs	List of Service IDs	N/A	O

⁸ 1. Networking, 2. Compute, 3. Storage, 4. Data, 5. Software, 6. Security, 7. Analytics, 8. Operations, 9. Training, 10. Consulting, 11. Other

⁹ Direct Connect, Virtual Network, Load Balancer, Application Gateway, VPN Gateway, Content Delivery Network, Traffic Manager, API Gateway, Virtual Machine Container Management, Batch, Serverless Applications Repository, Load Balancing, Blob, File, Queue, Disk, Archive, Backup, Synchronised, Replicated, Recovery, Mining, Access, Management, Transfer, Registration, Persistent Identifier, Interlinking, Publishing, Discovery, Anonymisation, Preservation, Brokering, Annotation, Platform, Tool, Component, Developer, Porting, Scaling, Education, Productivity, Business, Developer, Authentication and Authorisation, Attacks protection, Business Analytics, Web Analytics, Learning Analytics, Predictive Analytics, Machine Learning, Accounting, Helpdesk, Monitoring, Analysis, Con-figuration, Online, Face-to-face, Platform, Other

¹⁰ Values from ISO 639-1 list of prioritized countries, eInfraCentral-extension adds EU for Europe and WW for world

¹¹ Values from ISO 3166-1 alpha-2

Related Services	relatedServices[]	Other services that are commonly used with this service.	Service IDs	List of Service IDs	N/A	O
Service Support Information						
Service Order	order	The Uniform Resource Locator (web address) to the webpage to request the service from the service provider.	URL	http://www.egi.eu/cloud-compute/Order	Visible to Customers	M
Service Helpdesk	helpdesk	The Uniform Resource Locator (web address) to a webpage with the contact person or helpdesk to ask more information from the service provider about this service.	URL	http://www.egi.eu/cloud-compute/Helpdesk http://helpdesk.egi.eu	N/A	O
Service User Manual	userManual	The Uniform Resource Locator (web address) to the service user manual and documentation.	URL	http://www.egi.eu/cloud-compute/UserManual	N/A	O
Service Training Information	trainingInformation	The Uniform Resource Locator (web address) to training information on the service.	URL	http://www.egi.eu/cloud-compute/Training http://training.egi.eu	N/A	O
Service Feedback	feedback	The Uniform Resource Locator (web address) to the page where customers can provide feedback on the	URL	http://www.egi.eu/cloud-compute/Feedback http://servicefeedback.egi.eu	N/A	O

		service.				
Service Contractual Information						
Service Price	price	The Uniform Resource Locator (web address) to the information about the payment models that apply, the cost and any related information.	URL	http://www.egi.eu/cloud-compute/Price	N/A	O
Service Level Agreement	serviceLevel Agreement	The Uniform Resource Locator (web address) to the information about the levels of performance that a service provider is expected to achieve.	URL	http://www.egi.eu/cloud-compute/SLA	Downloadable in several document formats.	M
Service Terms Of Use	termsOfUse	The Uniform Resource Locator (web address) to the webpage describing the rules, service conditions and usage policy which one must agree to abide by in order to use the service.	URL	http://www.egi.eu/cloud-compute/TermsOfUse	Downloadable in several document formats.	O
Service Funding	funding	Sources of funding for the development and/or operation of the service.	Free text (max 1000)	EC (Horizon) and National (NREN) Development: EC (H2020), National (NREN). Operations: National (capacity and federation), EC (federation).	Logos should be used as well as text references. Publicity guidelines of each respective funding sources should be adhered to.	O

Service Level Targets and Performance Information						
Service Availability	Indicator{} ¹²	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.	Number	99,99%	As many decimal points as needed should be used.	O
Service Reliability	Indicator{}	Reliability, i.e., the probability that an item will function without failure under stated conditions for a specified amount of time. “Stated conditions” indicates 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	Number	98.1%	As many decimal points as needed should be used.	O

¹² See resource description in next table

		cooling systems would not be considered a failure of the supercomputer.				
Service Serviceability/Durability	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.	Number	99.99%	As many decimal points as needed should be used.	0
Service Performance Indicator Name	Indicator{}	Other Service Level Target or Performance Indicator	Free text	Users	N/A	0
Service Performance Indicator Value	Indicator{}	Indicator Value Measurement of Other Indicator	Number	1	N/A	0

Table 36: Indicator resource

Indicator Attribute	Attribute name in API	Description	Type	Example Value	Recommendation	<u>Mandatory \</u> <u>Optional</u>
Indicator ID	id	Global unique and persistent identifier of the indicator	Free Text	Availability	ID is assigned by eInfraCentral following the registration of the indicator.	M
Indicator Description	description	Brief and descriptive name of indicator	Free Text	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	N/A	M
Indicator Dimensions	dimensions[]	The dimension characterising the measurements of the indicator	List of text values used as dimensions of the indicator ¹³	TIME Period, LOCATION	N/A	M
Indicator Unit of Measurement	unit	The unit of measurement	List of values used as unit of measurement ¹⁴	Percentage, Numeric, Boolean	N/A	M

¹³ Values are {TIME PERIOD, LOCATION}¹⁴ Values are {PCT, NUM, BOOL} (Percentage, Numeric, Boolean)

Table 37: Indicator Measurement Resource

Measurement Attribute	Attribute name in API	Description	Type	Example Value	Recommendation	Mandatory \ Optional
Measurement ID	id	Global unique and persistent identifier of the measurement	Number	N/A	ID is assigned by eInfraCentral following the addition of the measurement.	M
Indicator Id	Indicator id	The indicator ID	Indicator ID	availability	N/A	M
Service ID	Service id	The service associated with the measurement	Service ID	1.03	N/A	M
Measurement Dimensions Values	TimePeriod, Locations[]	The values for the dimensions characterising the measurement	Dimension values identifying the measurement	TIME PERIOD= "1/1/2018-31/01/2018", Locations [UK, GR, DE]	N/A	M
Measurement Value	Value	The actual value of the measurement	Number	99.9%	N/A	M

Table 38: Service Provider Resource

Provider Attribute	Attribute name in API	Description	Type	Example Value	Recommendation	Mandatory \ Optional
Provider ID	id	Global unique and persistent identifier of the provider	Free Text	openaire	ID is assigned by eInfraCentral following the registration of the provider.	M
Provider Name	Name	Brief and descriptive name of provider as given by the service provider	Free Text	OpenAIRE		M
Provider Contact	contactInformation	The contact information in the	Free Text	contact@openaire.eu	N/A	O

Information		form of an email				
-------------	--	------------------	--	--	--	--

Table 39: Service Manager Resource

Manager Attribute	Attribute name in API	Description	Type	Example Value	Recommendation	Mandatory \ Optional
Manager ID	id	Global unique and persistent identifier of the manager	Free Text	eoschub	ID is assigned by eInfraCentral following the registration of the manager.	M
Manager Name	Name	Brief and descriptive name of provider as given by the service manager	Free Text	EOSCHub		M
Manager Contact Information	contactInformation	The contact information in the form of an email	Free Text	contact@eoschub.eu	N/A	O
Providers	Providers []	List of providers managed by the service manger	List of Provider IDs	{egi, eudat}		O

Table 40: Authenticated User Resource

User Attribute	Attribute name in API	Description	Type	Example Value	Recommendation	Mandatory \ Optional
User ID	id	Global unique and persistent identifier of the user, serving as username	Free Text	alice@eic	ID is provided during registration. It may be the same with the user email.	M
User Name	name	The name of the user	Free Text	Alice	N/A	M
User Surname	Surname	The surname of the	Free Text	Robertson	N/A	M

		user				
User Email	email	The email of the user	Free Text	alice@robertson.com	N/A	M
User Password	password	The password of the user	Free Text	changeMe	N/A	M
User Join Date	joinDate	The date the user registered in eInfraCentral	Date	05/06/2018	N/A	M
Organization	Organization	The organisation the user is affiliated with	Free Text	My organization	N/A	M
Service Manager	Manager	If the user belongs to a service manager, the service manager ID	Manager ID	egi	N/A	O

Table 41: User Events

Event Attribute	Attribute name in API	Description	Type	Example Value	Recommendation	Mandatory \ Optional
Event ID	id	Global unique identifier of the generated event	Number	12345	ID is assigned by eInfraCentral on event generation.	M
User Id	user	The ID of the user generating the event	User ID	egi@eic	N/A	M
Service Id	service id	The surname of the user	Service ID	1.13	N/A	M
Event type	Type	The type of the event	List values ¹⁵ of	Rate	N/A	M

¹⁵ Sample Values for the event type are rate, favorite, modifyService.

Event date	date	The date of the event	Date	05/06/2018	N/A	M
Event Parameter	Value	Optional parameters passed by an event	Multi value		N/A	O

Table 42: Statistics collected by eInfraCentral

Statistics Attribute	Attribute name in API	Description	Type	Example Value	Recommendation	<u>Mandatory \</u> <u>Optional</u>
Service Id	serviceId	The id of the service	Service ID	1.13	N/A	M
Provider Id	Provider id	The id of the provider	Provider Id	Egi	N/A	M
Statistic type	type	The type of the statistic	List of values ¹⁶	Rate	N/A	M
Reference Date	date	The reference date of the value reported	Date	05/06/2018	N/A	M
Value	Value	The actual value reported	Number	25	N/A	M

¹⁶ Sample Values for the type are rate, favorite, visit, order.

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://beta.einfracentral.eu/developers> and <http://beta.einfracentral.eu/openapi>
- **Step 2.** Register your organisation in eInfraCentral
 - Signup as a new Service provider¹⁷ and receive login credentials.
- **Step 3.** Export one or more services to JSON according to Service Description Template.
 - See a reference json in Section 3.1.
- **Step 4.** Use the POST\user method (see section 4.2.6) and retrieve a JWT
 - This method passes a JSON body containing the username and password and issues a JWT cookie. 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.

¹⁷ In the beta version of the eInfraCentral Gateway, new providers must send an email to contact@einfracentral.eu or dev-support@einfracentral.eu for the prior approval of their registration.