

# EOSC-IF

## Order Management System: Interoperability Guideline

Version 1.0  
09 2023

# EOSC-IF / Order Management System

Lead by **ACC Cyfronet AGH**

Authored by Jakub Sawicki (ACC Cyfronet AGH), Roksana Wilk (ACC Cyfronet AGH),  
Agnieszka Pułapa (ACC Cyfronet AGH), Paweł Gorczyca (ACC Cyfronet AGH), Katarzyna  
Lechowska-Winiarz (ACC Cyfronet AGH)

Reviewed by John Shepherdson (CESSDA)

## Dissemination Level of the Document

Public

## Abstract

The document describes the current architecture of the Marketplace Order Management System and provides guidelines for integration options for the providers and communities.

## Version History

Version	Date	Authors/Contributors	Description
V0.1	08/09/2023	Jakub Sawicki (ACC Cyfronet AGH), Roksana Wilk (ACC Cyfronet AGH), Agnieszka Pułapa (ACC Cyfronet AGH), Paweł Gorczyca (ACC Cyfronet AGH)	Initial version, technical information
V0.2	11/09/2023	Agnieszka Pułapa (ACC Cyfronet AGH)	Review
V0.3	12/09/2023	Agnieszka Pułapa (ACC Cyfronet AGH), Katarzyna Lechowska-Winiarz (ACC Cyfronet AGH)	Definitions added, Minor changes
V0.4	13/09/2023	Agnieszka Pułapa (ACC Cyfronet AGH), Paweł Gorczyca (ACC Cyfronet AGH)	Responded to review comments
V1.0		Agnieszka Pułapa (ACC Cyfronet AGH), Paweł Gorczyca (ACC Cyfronet AGH), Katarzyna Lechowska-Winiarz (ACC Cyfronet AGH), Ron Dekker (TGB), Mike Chatzopoulos (ATHENA)	Final Version submitted to EC

## Copyright Notice



This work by Parties of the EOSC Future Consortium is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/). The EOSC Future project is co-funded by the European Union Horizon Programme call INFRAEOSC-03-2020, Grant Agreement number 101017536.

## Table of Contents

<b>Table of Tables</b>	<b>2</b>
<b>Tabel of Figures</b>	<b>2</b>
<b>Glossary</b>	<b>3</b>
<b>List of Abbreviations</b>	<b>3</b>
<b>1 Intended Audience</b>	<b>4</b>
<b>2 Description and Main Features</b>	<b>4</b>
Offering	4
Ordering	4
Access model	4
Data model	4
Trigger	6
Request and response format	6
Authorization	6
<b>3 Response to Community Need</b>	<b>7</b>
<b>4 High-level Service Architecture</b>	<b>8</b>
<b>5 Definitions</b>	<b>8</b>
<b>6 Licensing Information</b>	<b>9</b>
<b>7 Related Standards</b>	<b>9</b>
<b>8 Integration Options</b>	<b>9</b>
<b>9 Interoperability Guidelines</b>	<b>9</b>
Integration steps	9
1. OMS Adapter (General)	10
2. Reference Adapter (JIRA)	10
Access model	10
1. OMS roles	10
• Read access	10
• Write access	11
Creating and Managing Offers	11
Managing Orders	16
<b>Table of Contents</b>	<b>2</b>

## Table of Tables

Table 8-1: Related Standards	8
Table 10-1: Permissions for specific roles in the message flow	10

## Tabel of Figures

Figure 5-1: Data model relations	5
Figure 5-2: Communication flow in the EOSC Marketplace	5
Figure 7-1: Architecture of the ordering process	7
Figure 8-1: Integration Options	8
Figure 11-1: Access to the offers panel	11
Figure 11-2: Adding new offer	11

Figure 12-1::Creating new offer	12
Figure 13-1:Defining technical parameters	13
Figure 14-1: Editing offer parameters	14
Figure 15-1:Save changes, delete, and save offer as draft.	15

## Glossary

EOSC Future project Glossary is incorporated by reference: <https://wiki.eoscfuture.eu/x/JQCK>

## List of Abbreviations

Acronym	Definition
OMS	Order Management System
API	Application Programming Interface
EOSC	European Open Science Cloud
JSON	JavaScript Object Notation
HTTP	Hypertext Transfer Protocol
SOMBO	(EOSC) Service Order Management Back-Office

## 1 Intended Audience

The intended audience for the Order Management Interoperability Guideline are providers and communities who would like to integrate with the Marketplace Order Management on different levels.

## 2 Description and Main Features

The API of the EOSC Marketplace enables integration in two areas: offering and ordering.

### Offering

Exposes functions that are necessary to manage offerings, including their technical parameters and ordering configuration. Its [OpenAPI Specification is available here](#).

### Ordering

This is an API that enables integration with the ordering process. External Order Management Systems (OMSes) can use it to keep order processing on the provider or community side, while still providing users with a consistent order workflow and support.

The Offering API is accessible for data administrators regardless of the existence of the OMS. The Ordering API is available after registering an OMS in the marketplace.

### Access model

OMS has relations with data administrators, which designate who can manage it. Data administrators can create a provider's group or resource dedicated OMSes. Data administrators can choose an OMS for an offer and the available options include: global OMSes, provider's OMS and a service dedicated OMS.

An OMS can have two roles in the context of project:

- mediator OMS: the default OMS
- provider OMS: every OMS, which is a primary OMS of a project's project item

An OMS can have two roles in a context of the project item:

- primary OMS: either OMS selected by a data administrator, or the default OMS
- secondary OMS: the default OMS

Every OMS with a role in project or project item has read access to all messages (*user\_direct?* with obfuscated message).

### Data model

In the EOSC Marketplace, every user can create Projects to organize their actions and get guidance from the operations team. Within a Project, the user will create service orders and add open-access resources. Both are represented as a single resource type in the API: the Project Item.

*Example: We have two resources, and one of them has the status "Order required," while the other is "Open Access." For "Order required" resources, users need to add them to the Marketplace project to access them. "Open Access" resources, on the other hand, do not require this, but they can still be added to the Marketplace project if desired. However, if a user decides to add such a resource to the project, the data model behind it is the same as for "Ordered Required" resources.*

A Project Item's status and user details are mutable by an owning OMS. In the scope of both a Project and a Project Item, a conversation between parties is held as Messages associated with either a Project or a Project Item.

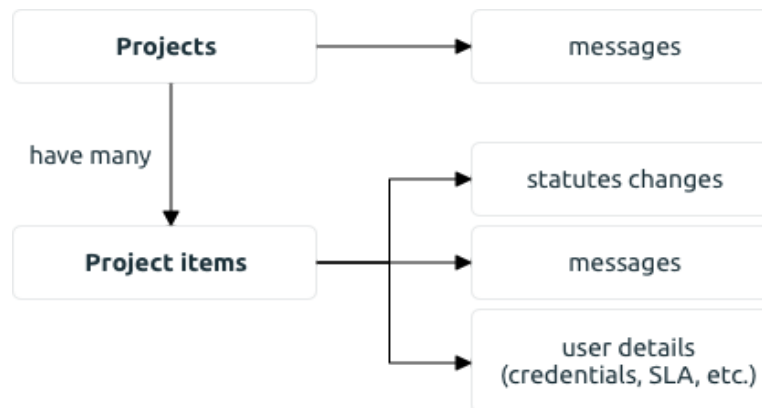


Figure 5-1: Data model relations

Below, we present an overview of the EOSC Marketplace APIs in the scope of interoperability patterns regarding order management. Users are presented with a consistent EOSC Marketplace user interface, while communities, providers, and operations teams can leverage the APIs to achieve a high interoperability level and provide value to the users in a way they see fit.

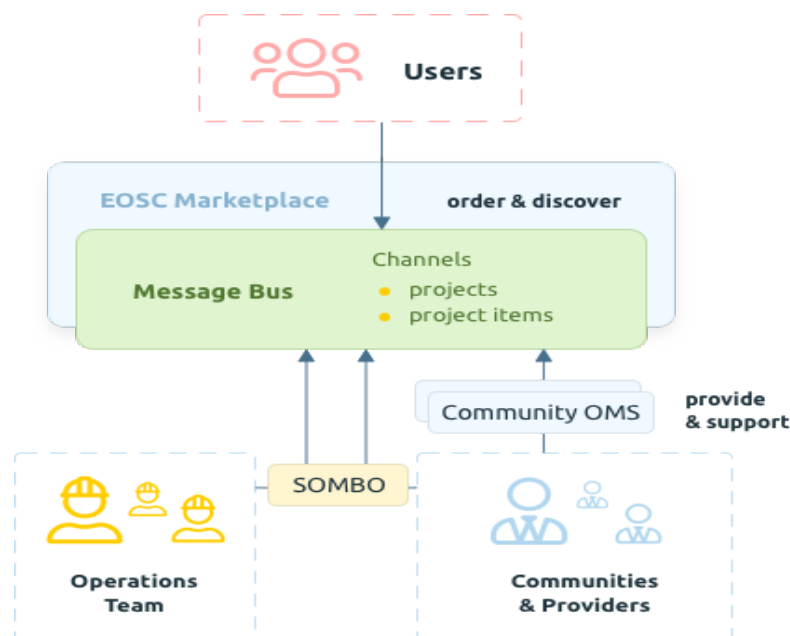


Figure 5-2: Communication flow in the EOSC Marketplace

## Trigger

In order to receive asynchronous notifications about new and changed resources, an OMS needs to expose a HTTP trigger endpoint. HTTP (GET, POST or PUT) requests will be issued on this endpoint each time the OMS needs to be notified of any changes.

A trigger call is issued with each change to resources (Projects, Project Items, Messages) that the OMS has access to.

After handling a trigger call, the adapter should send a request to the GET /events endpoint for updates.

*Example: curl -X 'GET' 'api/v1/oms/2/events?from\_timestamp=2021-04-07T15%3A31%3A46.123456Z' \  
-H 'accept: application/json' \  
-H 'X-User-Token: my\_generated\_token'*

## Request and response format

The API expects requests in JSON format and returns JSON too. Its content-type is application/json. Full API documentation is available [here](#)

## Authorization

Authorization tokens are used to authorize both APIs. Such a token can be retrieved from the [EOSC Marketplace API documentation](#) and will be similar to: *ios\_Bg6L1hsvDyvFYK\_C*. In the case of the Ordering API, such a token will be issued upon OMS registration. To provide integration support for other software systems the token is long-lived, but can be revoked and regenerated. To do this for the OMS token, one needs to contact support (using : cyfronet-support@mailman.eosc-portal.eu). To authenticate your HTTP requests, set the X-User-Token request header.

*For example: curl -H "X-User-Token: ios\_Bg6L1hsvDyvFYK\_C" [...]*

## 3 Response to Community Need

Integration becomes necessary when users must seek permission or make payments for resource usage, or when resources require configuration to grant access. Resources that necessitate ordering can be linked with the EOSC Order Management System.

Orders can be administered in two ways:

- Through the EOSC Order Management System.
- Through providers' own channels.

Consequently, providers with their order management systems aren't obligated to integrate with the EOSC Order Management System. Nevertheless, it is advisable to do so, as it offers greater convenience to users who can then place orders directly from the EOSC Marketplace.

By fully integrating with the EOSC Order Management System, one can avoid the need to develop an independent order management service (if you don't already have one). The resource's maturity will be enhanced. The end-user experience will be enriched:

1. Users will receive notifications regarding the status of their orders via the EOSC Portal.
2. Managing all orders will become more straightforward for end users, as they can do so using a consistent interface without leaving the EOSC Portal.



## 4 High-level Service Architecture

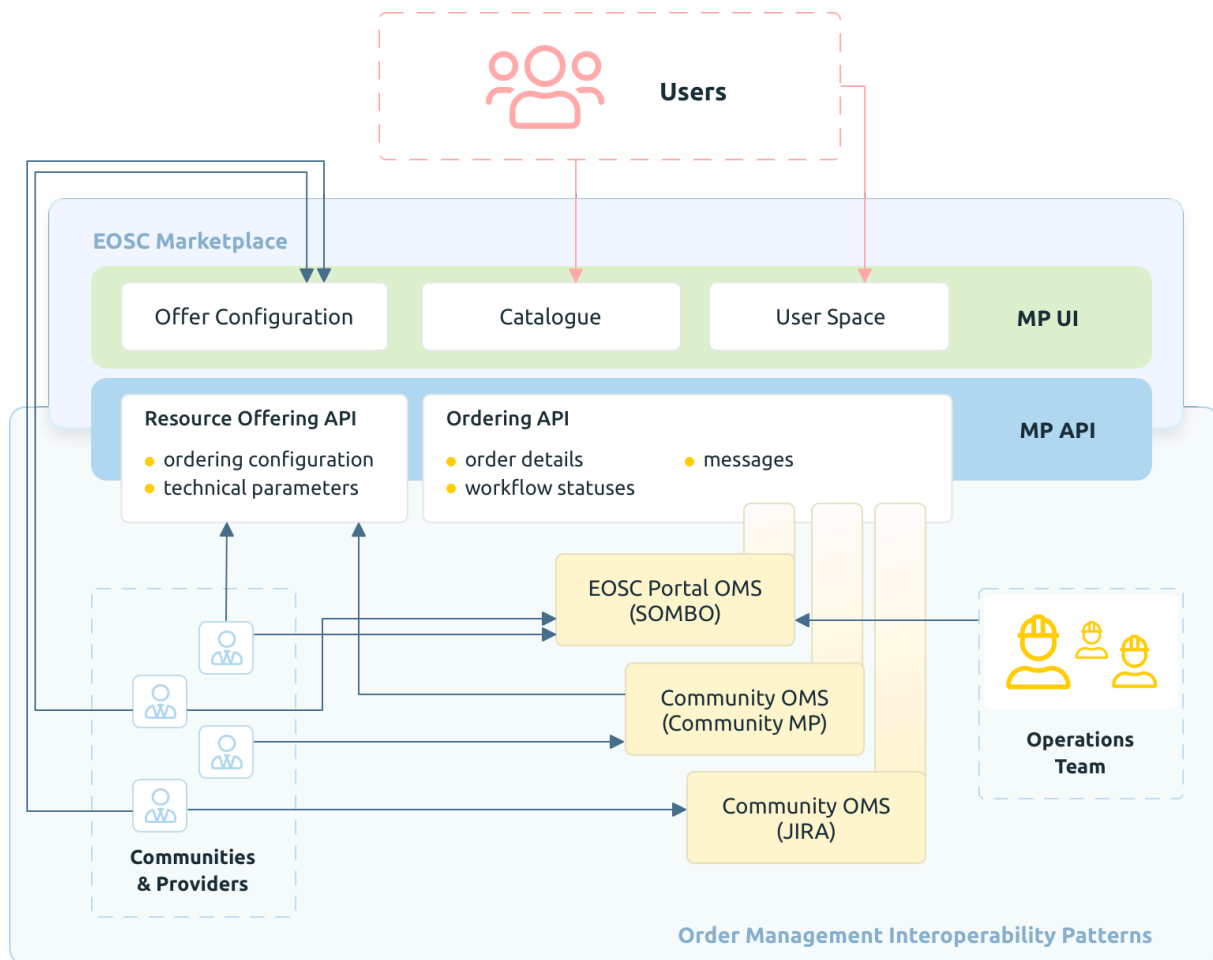


Figure 7-1: Architecture of the ordering process

Integrators (providers and communities) can choose from a range of integration levels, adding flexibility. In a minimal setup, a provider can use the Marketplace UI to configure their offerings and use SOMBO for order management, avoiding any implementation and integration costs. On the other hand, one can choose to use an in-house system that fully integrates with EOSC Marketplace APIs both for offering and order management. Existing community systems can be extended this way, exposing the communities to EOSC users while maintaining the same set of tools for operations. Users can check project items statuses and exchange messages with the operations team and service providers through the Marketplace UI. Communities and providers should provide an OMS endpoint to push a trigger for updating data. This allows project item information to be kept up to date. The Offering API enables the management of offers, which are objects of customers' orders. The Ordering API keeps all the information about orders regarding OMS connected to the provider or a community.

## 5 Definitions

EOSC Order Management is a set of services, processes and guidelines to enable providers to integrate ordering and procurement of their EOSC Resources. Users, on the other hand, can then enjoy a more integrated user journey from resource discovery to procurement, within a coherent EOSC Portal. It provides users who order EOSC Resources with a more uniform way through the ordering process. When a user aggregates resource orders (project items) within a project, it also enables inter-provider communications, to enhance composability of resources and increase outreach. Both users and providers also gain support from the EOSC Portal Operations Team that is in charge of reviewing the orders, clarify the orders with the user if

needed, assign the orders to the corresponding provider, keep up to date the status of the orders. The team is organised in shifters with rotation every two weeks.

Offering -implies the act of giving, providing, or making something available to others. The nature of offerings can vary widely depending on the context in which they are used. When it comes to technology, software, and digital services, one should think of them as purpose-driven tools—software applications, platforms, or features thoughtfully crafted to meet users' needs and solve specific challenges.

Ordering - refers to process of organizing resources in a project. The goal of ordering is to establish a systematic and organized structure, making it easier to locate, categorize, or understand the items being ordered.

## 6 Licensing Information

The Offering and Ordering APIs are compatible with OpenAPI under the Apache-2.0 license.

## 7 Related Standards

Table 8-1: Related Standards

Title	Short Description	relatedStandardIdentifier
OpenAPI v3.0.1	Both the Offering and Ordering APIs are compatible with the OpenAPI standard	<a href="https://github.com/OAI/OpenAPI-Specification/blob/3.0.1/versions/3.0.1.md">https://github.com/OAI/OpenAPI-Specification/blob/3.0.1/versions/3.0.1.md</a>

## 8 Integration Options

The Ordering API is meant to support integration of external OMSes.

The interoperability layer is based on the Ordering API with asynchronous triggers that can be accessed by interested parties. Integration Options are presented on the picture.

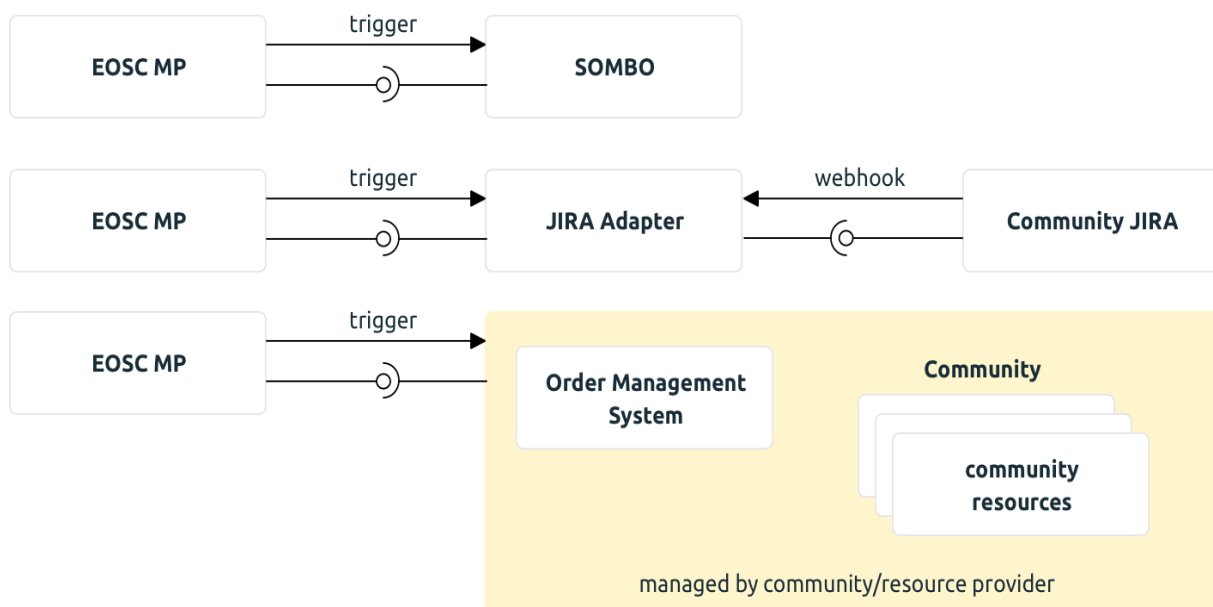


Figure 8-1: Integration Options

## 9 Interoperability Guidelines

### Integration steps

In order to integrate your ordering process with the EOSC Marketplace, you have to follow a few steps:

- Obtain an OMS registration in the test infrastructure,
- Connect your instance of an OMS adapter (you can use the reference Jira implementation for this).
- Switch to the production environment.

#### 1. OMS Adapter (General)

The OMS adapter functions as a way to generalize communication with ticket management systems. The reference implementation supports integration with JIRA. But any integrator can develop its own compatibility layer with their current ticketing system, see [API documentation](#) for more details.

The EOSC Marketplace has a very simple way of communicating with the Adapter (trigger). Everything else is done proactively by the OMS adapter itself. Usually the ticket management system with which integration occurs informs the OMS adapter about some changes, then the OMS adapter calls the Marketplace API. When any change occurs in the Marketplace (new Project / Project Item is created, Message is posted) the Marketplace triggers the OMS adapter (every registered OMS adapter has a corresponding trigger configuration) and the adapter then makes any necessary calls to the ticketing system to enact that change.

It is in the scope of the OMS adapter to store mappings between Marketplace's project item IDs and ticketing system IDs.

#### 2. Reference Adapter (JIRA)

The provided reference implementation contains a Marketplace API client which is responsible for communication with the Ordering API. It also contains a thin translation layer which translates Project Item parameters to fields in JIRA. The core philosophy enshrined in the system is that it should not, under any circumstances, lose information about ordering changes. To fulfill this purpose all actions are scheduled to the Distributed Task queue, this way they can be retried if network errors occur, or in case they fail they can be reviewed manually by the operator.

### Access model

The following applies to the Ordering API.

#### 1. OMS roles

An OMS can have two roles in the context of a Project Item:

- primary: if selected by the EOSC Resource provider to handle that offering orders,
- secondary: the default OMS.

In case the primary OMS wasn't selected, then the default OMS is considered both as primary and secondary OMS.

#### 2. An OMS can have two roles in the context of a Project:

- mediator: the default OMS,
- provider: every OMS, which is a Primary OMS of any Project Item of the Project.

- Read access

If an OMS has any role in a Project or Project Item, it can read it. The Project Item's field *user\_secrets* will have obfuscated values (keys will be visible).

Every OMS with a role in a Project or Project Item has read access to all associated messages. However, the content of messages with *user\_direct* scope will be obfuscated. Messages with internal scope will not be disclosed to users.

- Write access

A Project cannot be changed via the API. A Project Item can be changed by its primary OMS.

Messages have fields *scope* and *role*, which determine their write access. The scope can have values *public*, *internal* or *user\_direct*. The role can have the following values: *user*, *provider* or *mediator*.


The write permissions for different combinations of scope and role are as follows. They also differ slightly for Project and Project Item messages.

*Table 10-1: Permissions for specific roles in the message flow*

scope \ role	user	provider	mediator
<b>public</b>	false	is provider OMS?	is mediator OMS?
<b>internal</b>	false	is provider OMS?	is mediator OMS?
<b>user_direct</b>	false	false	is mediator OMS?

## Creating and Managing Offers

1. Accessing the ordering configuration page
  - To initiate the process, the service provider should access the EOSC Marketplace website and log in to their provider account.
  - Navigate to the service page by using the search engine to find their service in the Marketplace catalog.
2. Accessing service management options
  - On the service page, the Provider should locate and select the "Manage the service" button. This action will reveal a dropdown menu with three distinct options; services dashboard, edit service details and set parameters and offers.



### Demonstration Service Platform

DSP

DS Platform

Organisation: Cyfronet\_Test\_L3

☆☆☆☆☆ (0.0 / 5) 0 reviews ☐ Add to comparison ☐ Add to favourites

[Access the service](#)

[Manage the service](#)

- Services dashboard
- Edit service details
- Set parameters and offers

[Webpage](#)
[Helpdesk e-mail](#)

[ABOUT](#)
[DETAILS](#)
[GUIDELINES](#)
[REVIEWS \(0\)](#)

The Demonstration Service Platform is a user-friendly digital tool that helps businesses process and analyze large datasets from diverse sources. It simplifies data management, automates tasks, and provides valuable insights. Users benefit from streamlined operations, improved decision-making, and cost savings. It serves a wide range of organizations and data professionals across various industries.

#### SCIENTIFIC CATEGORISATION




- Engineering & Technology
  - Electrical, Electronic & Information Engineering

#### CATEGORISATION

- Compute
  - Virtual Machine Management

Figure 11-1: Access to the offers panel

- From the dropdown menu, the Provider should choose the "Set parameters and offers" option. This action will unveil a panel that facilitates the creation and management of the Provider's offers.



### Demonstration Service Platform

DS Platform

Organisation: Cyfronet\_Test\_L3

☆☆☆☆☆ (0.0 / 5) 0 reviews

[Back to the service](#)
[Set parameters and offers](#)

ORDER REQUIRED

[Webpage](#)
[Helpdesk e-mail](#)

[ABOUT](#)
[DETAILS](#)
[GUIDELINES](#)
[REVIEWS \(0\)](#)

The Demonstration Service Platform is a user-friendly digital tool that helps businesses process and analyze large datasets from diverse sources. It simplifies data management, automates tasks, and provides valuable insights. Users benefit from streamlined operations, improved decision-making, and cost savings. It serves a wide range of organizations and data professionals across various industries.

#### SCIENTIFIC CATEGORISATION



- Engineering & Technology
  - Electrical, Electronic & Information Engineering

#### CATEGORISATION

- Compute
  - Virtual Machine Management

#### STATISTICS

- Service catalog entry views: **GA not initialized**
- Service visits from the Service catalog: **GA not initialized**
- Total numbers of appearances

#### Parameters and offers

**PUBLISHED**

**Offer**

Demonstration Service Platform Offer

The technical parameters configuration is empty

Edit parameters

ORDER REQUIRED

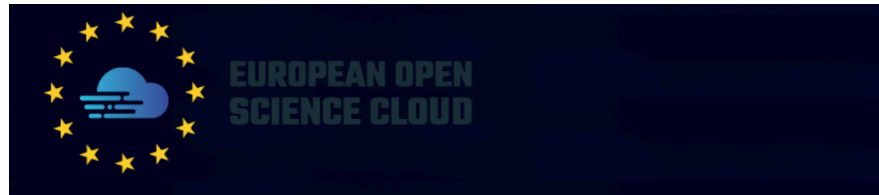
You can add a dedicated offer for your service

Add new offer

Figure 11-2: Adding new offer

### 3. Creating a new offer and completing offer details

- Once the "Parameters and offers" panel is selected, to initiate the creation of a new offer, the Provider should click the "Add new offer" button.
- The provider will then be presented with an offer form that must be filled out comprehensively. Mandatory fields for the offer include: name, description and order type.



## Create new offer

Name \*

Description \*

Tag list

☐ Bundle exclusive

Order type \*

☐ Use EOSC Portal as the order management platform

Order / access URL

Url should start with http:// or https:// prefix [e.g. https://example.com].

Given URL will be used in the offer order/access mechanism - on the service access information step this URL will be used under 'Go to the order website' / 'Go to the service' button (depending on the offer order type). If it isn't specified it will fall back to the service's webpage URL.

*Figure 12-1::Creating new offer*

- In the second paragraph of the form, the Provider should define the technical parameters of the offer. This may encompass details such as, amount of RAM per CPU core, number of CPU cores, number of days, requested resource capacity, max size dataset and other technical details specific to the service.

## OFFER PARAMETERS

Constant

Input

Select

Multiselect

Date

Range

Quantity price

Select parameter
⤴ ⤵ ⌕ 🗑

Name \*

Number of CPU Cores ✓

Hint

Select number of cores you want ✓

Values \*

1 X

2 X

4 X

8 X

+ start typing to add

Value type \*

integer ✓✓

Mode \*

buttons ✓✓

Unit

Select parameter
⤴ ⤵ ⌕ 🗑

Name \*

Amount of RAM per CPU core ✓

Hint

Select amount of RAM per core ✓

Values \*

1 X

2 X

4 X

+ start typing to add

Value type \*

integer ✓✓

Mode \*


buttons ✓✓

Unit

GB ✓

Figure 13-1: Defining technical parameters

- After duly completing all the mandatory fields and specifying the technical parameters, the Provider should proceed by clicking the "Create offer" button.
4. Editing offer details
- To initiate modifications to a specific offer, the provider should start by clicking the "Edit Parameters" button associated with the respective offer. This action opens the offer's editing form, where the Provider can make the necessary adjustments to the offer.



### Demonstration Service Platform

DS Platform  
Organisation: Cyfronet\_Test\_3  
☆☆☆☆☆ (0.0 / 5) 0 reviews

[Back to the service](#)  
[Set parameters and offers](#)  
ORDER REQUIRED

[Webpage](#) [Helpdesk e-mail](#)

[ABOUT](#) [DETAILS](#) [GUIDELINES](#) [REVIEWS \(0\)](#)

The Demonstration Service Platform is a user-friendly digital tool that helps businesses process and analyze large datasets from diverse sources. It simplifies data management, automates tasks, and provides valuable insights. Users benefit from streamlined operations, improved decision-making, and cost savings. It serves a wide range of organizations and data professionals across various industries.


## Parameters and offers

PUBLISHED

ORDER REQUIRED


### Offer

Demonstration Service Platform Offer



The technical parameters configuration is empty

[Edit parameters](#)



You can add a dedicated offer for your service

[Add new offer](#)

#### SCIENTIFIC CATEGORISATION



Engineering & Technology

- Engineering & Technology
  - Electrical, Electronic & Information Engineering

#### CATEGORISATION

- Compute
  - Virtual Machine Management

#### STATISTICS

- Service catalog entry views: **GA not initialized**
- Service visits from the Service catalog: **GA not initialized**
- Total numbers of appearances in the project: **0**

*Figure 14-1: Editing offer parameters*

- After making the desired modifications, the Provider should ensure to save the changes by clicking the "Update offer" button. This step is crucial to apply the modifications to the existing offer.
- Additionally, within the offer editing form, the Provider has the option to choose:
  - "Delete offer" button if there are multiple existing offers from the provider, and they wish to remove this particular offer.
  - "Draft offer" button if they want to save the changes as a draft without immediately updating the live offer.



Constant  
Input  
Select  
Multiselect  
Date  
Range  
Quantity price

Min \*  
 ✓  
☐ Exclusive min

Max \*  
 ✓  
☐ Exclusive max

Range parameter ^ v x trash

Name \*  
 ✓

Hint  
 ✓

Unit  
 ✓

Min \*  
 ✓  
☐ Exclusive min

Max \*  
 ✓  
☐ Exclusive max

Update Offer  
Delete Offer  
Draft offer

CANCEL AND BACK TO PREVIOUS PAGE

Figure 15-1: Save changes, delete, and save offer as draft.

## Managing Orders

Providers have two primary methods for managing orders on the EOSC Marketplace, depending on their order management capabilities:

1. **Provider's Dedicated Order Management System (OMS):** If a provider has their own OMS in place, it can independently manage orders received via EOSC Marketplace. The OMS can seamlessly integrate with the EOSC Marketplace through the provided OMS Adapter, enabling the Provider to handle order processing, tracking, and fulfilment within their existing infrastructure.
2. **Utilising the JIRA Integration:** For providers who do not maintain a dedicated Order Management System, orders directed to them are managed through the EOSC Marketplace's integration with JIRA. The EOSC Marketplace's Reference Adapter serves as a bridge between the marketplace and JIRA. It communicates with the Ordering API and ensures that all order-related information is accurately captured and processed.

This dual approach provides flexibility for providers, allowing them to choose the order management method that best aligns with their existing systems and preferences, while ensuring efficient order processing and communication.