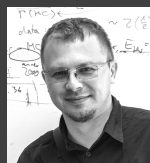
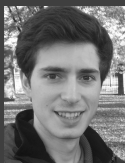


EOSC-Performance: find most suitable EOSC site for your task

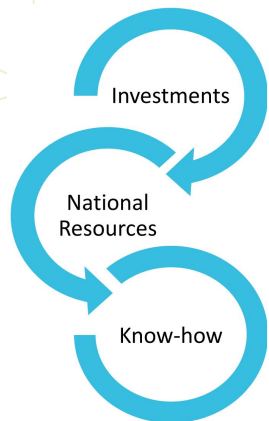
EGI Conference 2021, “virtual Amsterdam”

Karlsruhe Institute of Technology (KIT-SCC):

Borja Esteban Sanchis, Marcus Hardt, Valentin Kozlov, Christophe Laures



EOSC Synergy in a nutshell



To expand **EOSC** by leveraging

- **Investments**
- **Resources** of **national** digital infrastructures
- Existing **experiences & know-how**

Foster EOSC with **Software and Service Quality**

Thematic services in **Astrophysics, Biomedicine, Earth Observation, Environment**

Increase in the number of **resources, services**, and data **repositories** offered to **researchers** through EOSC

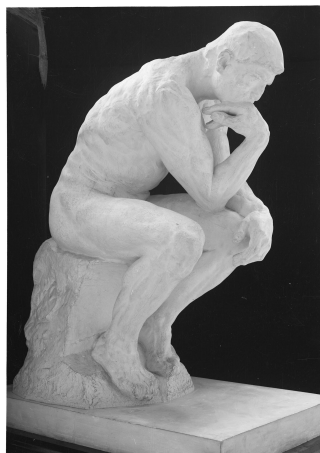


Spain, Portugal, UK,
Czech Republic, Germany,
Slovakia, Poland, Netherlands

*It is in the **general interest** of **users** and **service providers** to **compare** the available **computing resources***

Introduction (aka Motivation)

Example *user stories* for comparing computing resources:



THE THINKER - AUGUSTE RODIN
OF ART

As: a domain scientist

US1

I want: to **compare** various computing resources available in EOSC

So that: I can choose **most suitable** resources to run my own software

Ask for: **well-structured and searchable information**

As: an advanced user / a resource provider

US2

I want: to **store results** of any benchmarks of interest

So that: I can **compare** resources for the metrics of interest

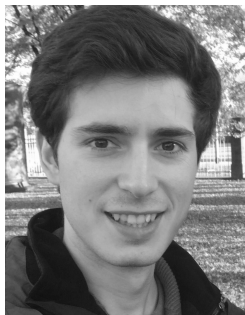
Ask for: a platform to **store results of any benchmark of interest**

EOSC-Performance

Key Developers

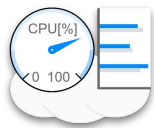


Christophe Lares
Studying Informatics at KIT
Previously worked with C++ and Python



Borja Esteban Sanchis
Works at KIT
Loves Erlang, works with Python

EOSC-Perf: in brief

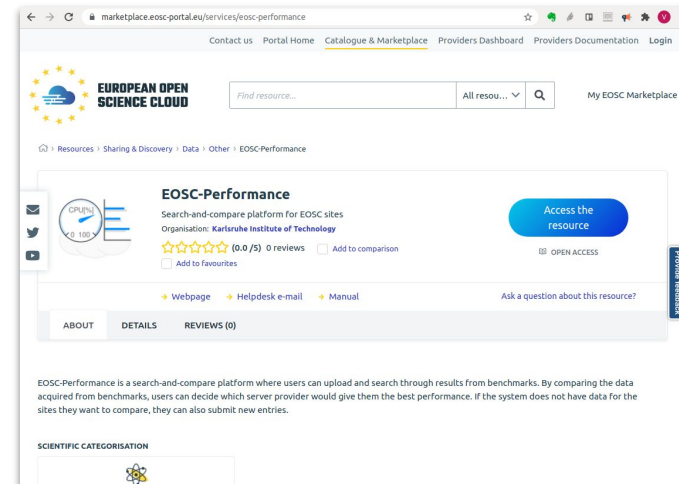


EOSC-Performance is a **web application** to **search, compare** and **submit** results from **benchmarks** run on a variety of different machines (single, cloud, HPC).

Benchmark: is a code packed in a Docker container, available on the Docker Hub, and produces JSON output.

Main endpoint: <https://performance.services.fedcloud.eu/>

It is available in the [EOSC-Marketplace](#):



Notice for this talk

In **this** presentation a **new** version, just deployed **yesterday**, is shown

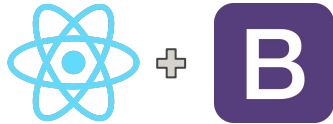
Main enhancements concern:

- Add full-featured **API**
- Corresponding changes in the **Frontend** (re-written in React)
- Internal service re-arrangement (e.g. SQLite -> Postgres)

Test endpoint for demos: <https://perf.test.fedcloud.eu>






EOSC-Perf: structure under the hood



Frontend, User interface:

- Responsive website developed using  React and  react-bootstrap

Backend, API:

- Written entirely in  Flask
- Based on  OpenAPI spec v3
- Swagger interface
- Communication between Database and Interface
- Handles authentication using OIDC and EGI-Check-In 

Database/Model:

- Powered by  SQLAlchemy
- PostgreSQL 

Web Frontend

EOSC-Perf: Supported Flows

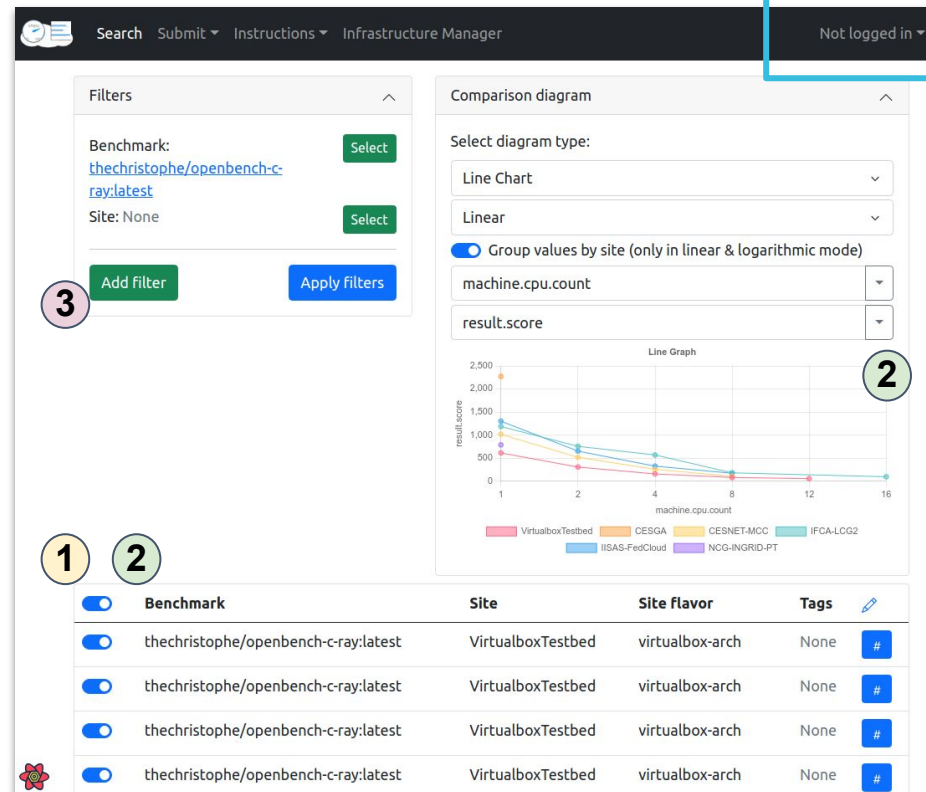
US1: well-structured and searchable information

Any user can:

Browse the existing results 1

Compare (table, plotting) 2

Filter them 3



EOSC-Perf: Supported Flows

US1: well-structured and searchable information

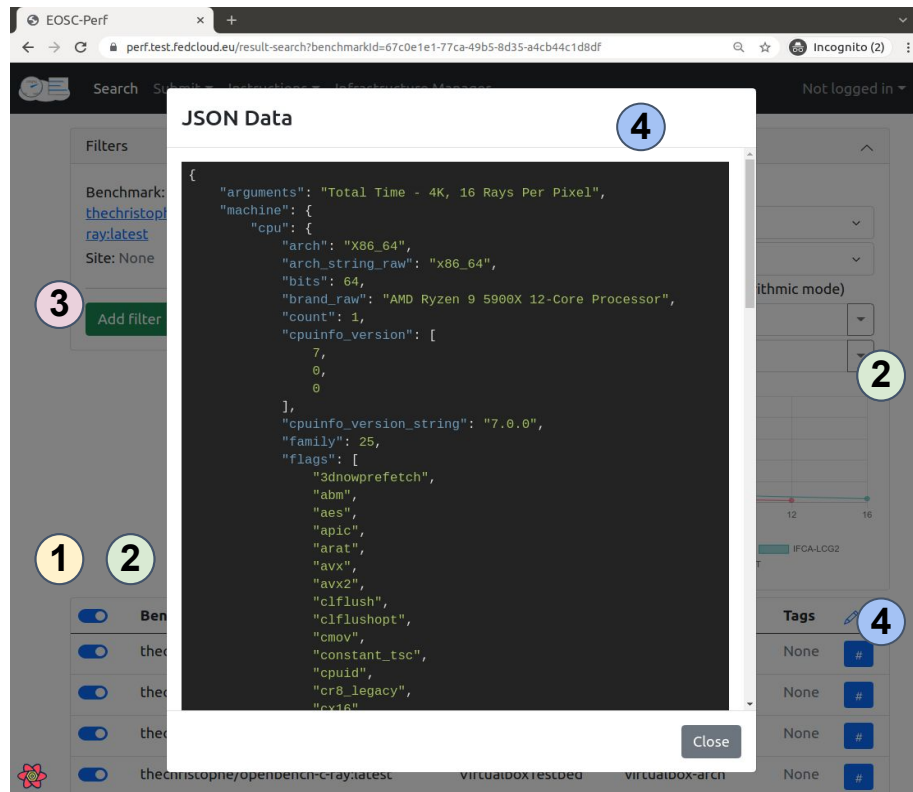
Any user can:

Browse the existing results 1

Compare (table, plotting) 2

Filter them 3

View full JSON output 4



The screenshot displays the EOSC-Perf web application. On the left, a 'Filters' sidebar allows users to refine search results by benchmarks (e.g., 'thechristophe', 'ray/latest'), site, and other criteria. The main area shows a list of results with columns for benchmark, site, and tags. A modal window titled 'JSON Data' is open, displaying the full JSON output for a selected result. The JSON data includes fields such as 'arguments', 'machine', 'cpu', 'arch', 'bits', 'brand_raw', 'count', 'cpuinfo_version', 'cpuinfo_version_string', 'family', 'flags', and '3dnowprefetch'. The interface also features a search bar, a 'Not logged in' status, and a 'Tags' section on the right.

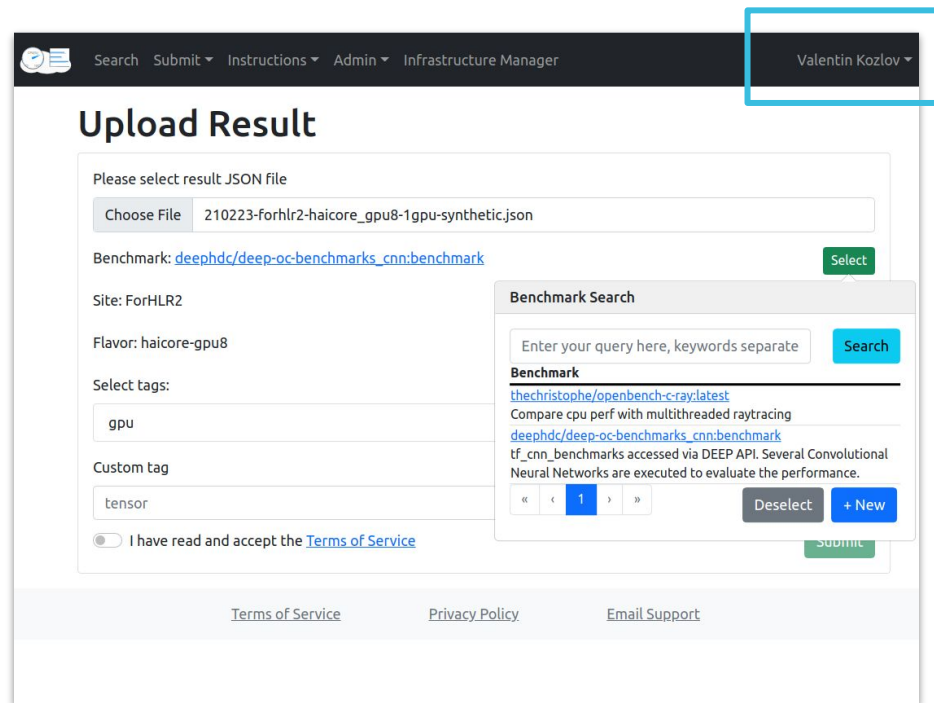
EOSC-Perf: Supported Flows

US2: store results of
any benchmark of interest

Contributors (⇒ authentication):

Upload a new result:

- JSON file
- Benchmark
- Site
- Site Flavor
- Tags



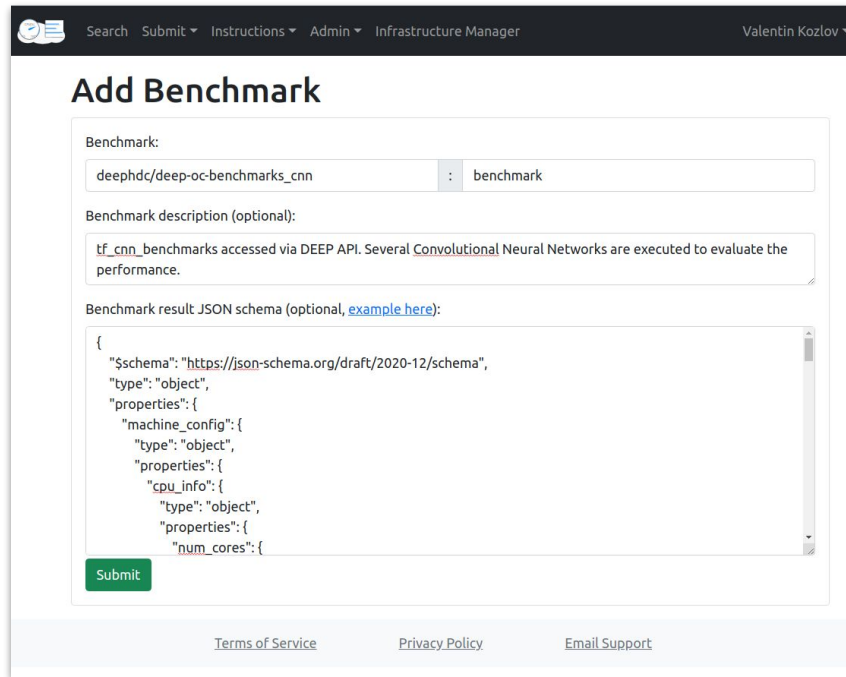
EOSC-Perf: Supported Flows

US2: store results of
any benchmark of interest

Contributors (⇒ authentication):

Add a new benchmark:

- Benchmark:
Dockerhub image + docker tag
- Benchmark description
- JSON schema
(may include
"suggestToUser" key)



The screenshot shows the 'Add Benchmark' form in the EOSC-Perf web application. The form is titled 'Add Benchmark' and is located under the 'Infrastructure Manager' tab. It contains three main sections: 'Benchmark', 'Benchmark description (optional)', and 'Benchmark result: JSON schema (optional, [example here](#))'. The 'Benchmark' section has a text input field with 'deepfdc/deep-oc-benchmarks_cnn' and a dropdown menu with 'benchmark'. The 'Benchmark description (optional)' section has a text area with the text 'tf_cnn_benchmarks accessed via DEEP API. Several Convolutional Neural Networks are executed to evaluate the performance.' The 'Benchmark result: JSON schema (optional, [example here](#))' section has a text area with a JSON schema snippet. At the bottom of the form is a green 'Submit' button. The footer of the page contains links for 'Terms of Service', 'Privacy Policy', and 'Email Support'.

Add Benchmark

Benchmark:

deepfdc/deep-oc-benchmarks_cnn : benchmark

Benchmark description (optional):

tf_cnn_benchmarks accessed via DEEP API. Several Convolutional Neural Networks are executed to evaluate the performance.

Benchmark result: JSON schema (optional, [example here](#)):

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "type": "object",
  "properties": {
    "machine_config": {
      "type": "object",
      "properties": {
        "cpu_info": {
          "type": "object",
          "properties": {
            "num_cores": {
```

Submit

[Terms of Service](#) [Privacy Policy](#) [Email Support](#)

EOSC-Perf: Supported Flows

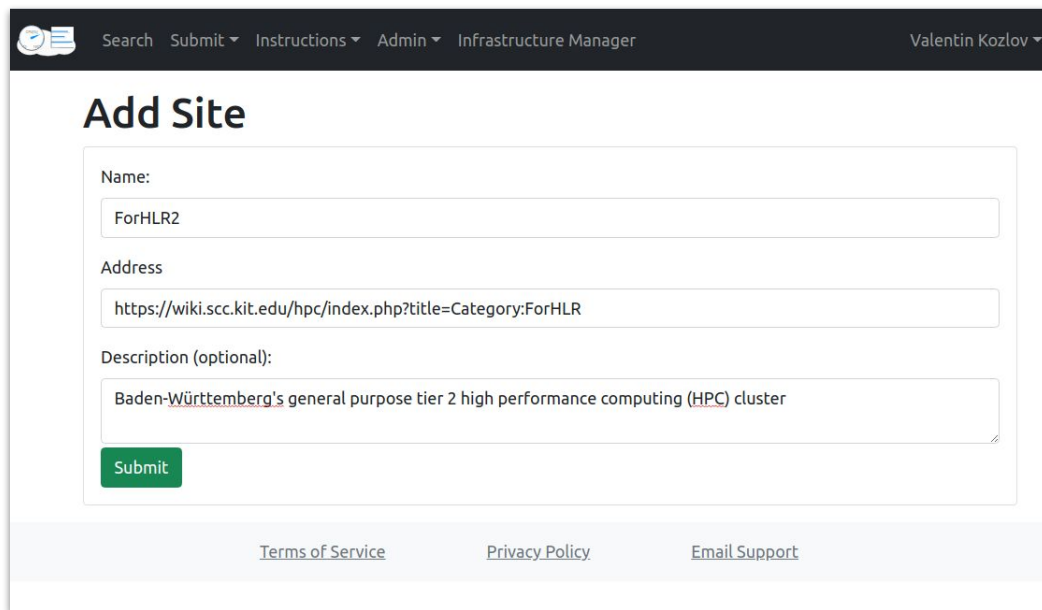
US2: store results of
any benchmark of interest

Contributors (⇒ authentication):

Add a new resource:

- Name
- Address
- Description

N.B.: Adding Site / Flavor is also
possible during “Upload Result”

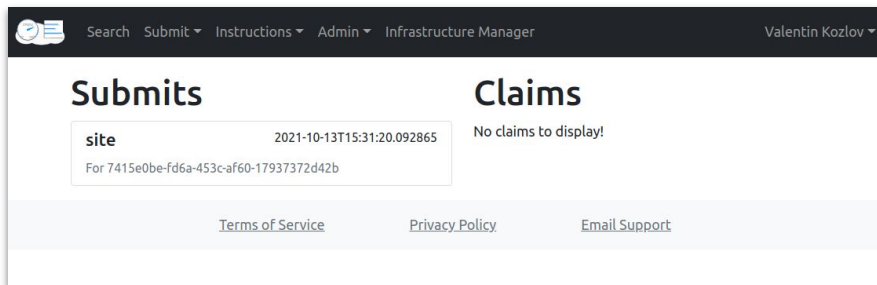


The screenshot shows the 'Add Site' form in the EOSC-Perf interface. The form has a dark header bar with navigation links: Search, Submit, Instructions, Admin, and Infrastructure Manager. The user's name, Valentin Kozlov, is displayed in the top right corner. The form itself is titled 'Add Site' and contains three input fields: 'Name' (with the value 'ForHLR2'), 'Address' (with the value 'https://wiki.scc.kit.edu/hpc/index.php?title=Category:ForHLR'), and 'Description (optional):' (with the value 'Baden-Württemberg's general purpose tier 2 high performance computing (HPC) cluster'). A green 'Submit' button is located at the bottom of the form. Below the form, there are three links: 'Terms of Service', 'Privacy Policy', and 'Email Support'.

EOSC-Perf: Supported Flows

Admin functions:

Review Submissions and Reports



Search Submit Instructions Admin Infrastructure Manager Valentin Kozlov

Submits

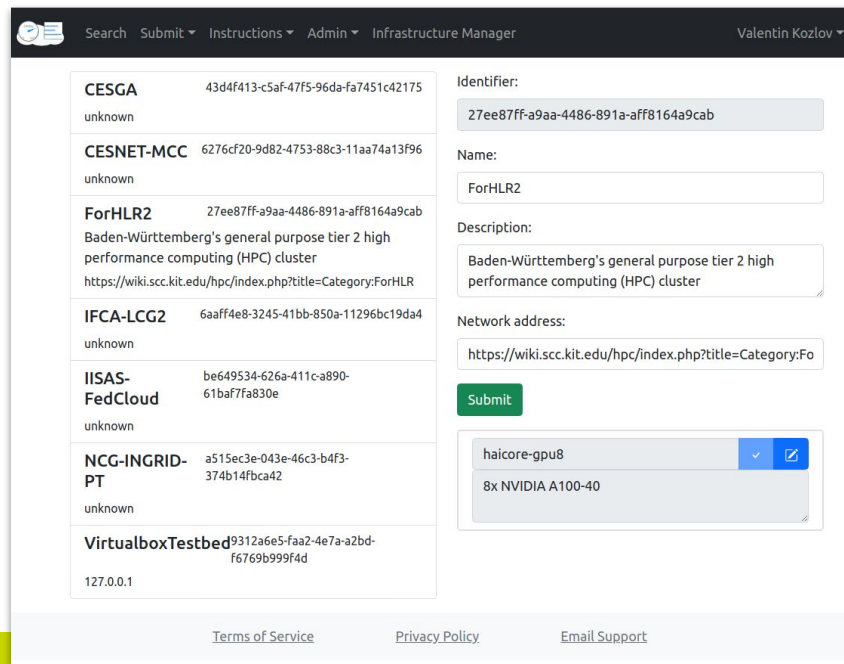
site	2021-10-13T15:31:20.092865
For 7415e0be-fd6a-453c-af60-17937372d42b	

Claims

No claims to display!

[Terms of Service](#)
[Privacy Policy](#)
[Email Support](#)

Edit sites with “Site Editor”



Search Submit Instructions Admin Infrastructure Manager Valentin Kozlov

CESGA	43d4f413-c5af-47f5-96da-fa7451c42175	Identifier:	27ee87ff-a9aa-4486-891a-aff8164a9cab
unknown		Name:	ForHLR2
CESNET-MCC	6276cf20-9d82-4753-88c3-11aa74a13f96	Description:	Baden-Württemberg's general purpose tier 2 high performance computing (HPC) cluster
unknown		Network address:	https://wiki.scc.kit.edu/hpc/index.php?title=Category:ForHLR2
ForHLR2	27ee87ff-a9aa-4486-891a-aff8164a9cab	Submit	haicore-gpu8
Baden-Württemberg's general purpose tier 2 high performance computing (HPC) cluster			8x NVIDIA A100-40
https://wiki.scc.kit.edu/hpc/index.php?title=Category:ForHLR2			
IFCA-LCG2	6aaff4e8-3245-41bb-850a-11296bc19da4		
unknown			
IISAS-FedCloud	be649534-626a-411c-a890-61ba77fa830e		
unknown			
NCG-INGRID-PT	a515ec3e-043e-46c3-b4f3-374b14fbca42		
unknown			
VirtualboxTestbed	9312a6e5-faa2-4e7a-a2bd-f6769b999f4d		
127.0.0.1			

[Terms of Service](#)
[Privacy Policy](#)
[Email Support](#)

EOSC-Perf: more features

- Custom filters
- Expand table with benchmark-specific JSON fields
- Report results (authenticated users)
- Custom tags
- ...

DEMO: adding a new result via web-interface

API

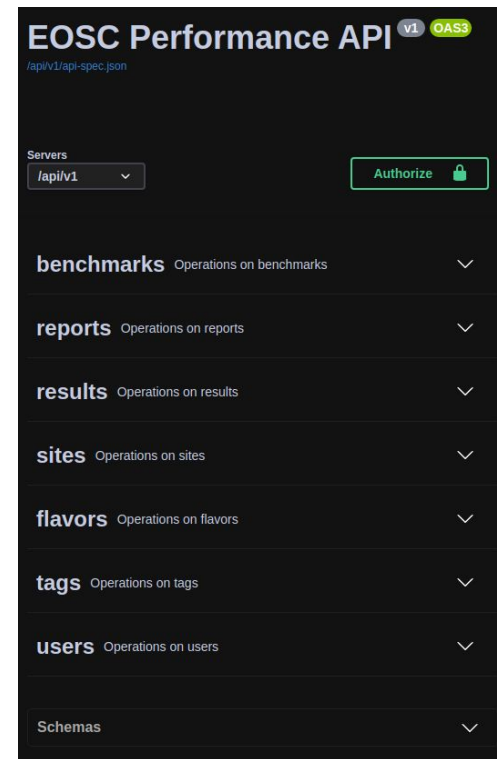
API - Powered by Swagger and OpenAPI

Based on OAS3

- Possibility to provide multiple hosts.
- Supports Bearer/JWT authentication natively.

Friendly web user interface by Swagger accessible at:

<https://performance.services.fedcloud.eu/api/v1/>



API - Designed to follow REST architecture

benchmarks Operations on benchmarks			
GET	/benchmarks	(Public) Filters and list benchmarks	✓ 🔒
POST	/benchmarks	(Users) Uploads a new benchmark	✓ 🔒
GET	/benchmarks:search	(Public) Filters and list benchmarks	✓ 🔒
GET	/benchmarks/{id}	(Public) Retrieves benchmark details	✓ 🔒
PUT	/benchmarks/{id}	(Admins) Implements JSON Put for benchmarks	✓ 🔒
DELETE	/benchmarks/{id}	(Admins) Deletes an existing benchmark	✓ 🔒
POST	/benchmarks/{id}:approve	(Admins) Approves a benchmark to include it on default list methods	✓ 🔒
POST	/benchmarks/{id}:reject	(Admins) Rejects a benchmark to safe delete it.	✓ 🔒

← List items
← Create items

← Get item Admins
← Edit item
← Delete item

API - Automate results and more

DEMO

Powered by bash and Jupyterlab.

How to automatically submit a new result.



https://github.com/EOSC-synergy/eosc-perf/blob/backend-dev/service_backend/examples/how_to_submit_results.ipynb

API - To contribute, register

Please, read our privacy policy at:

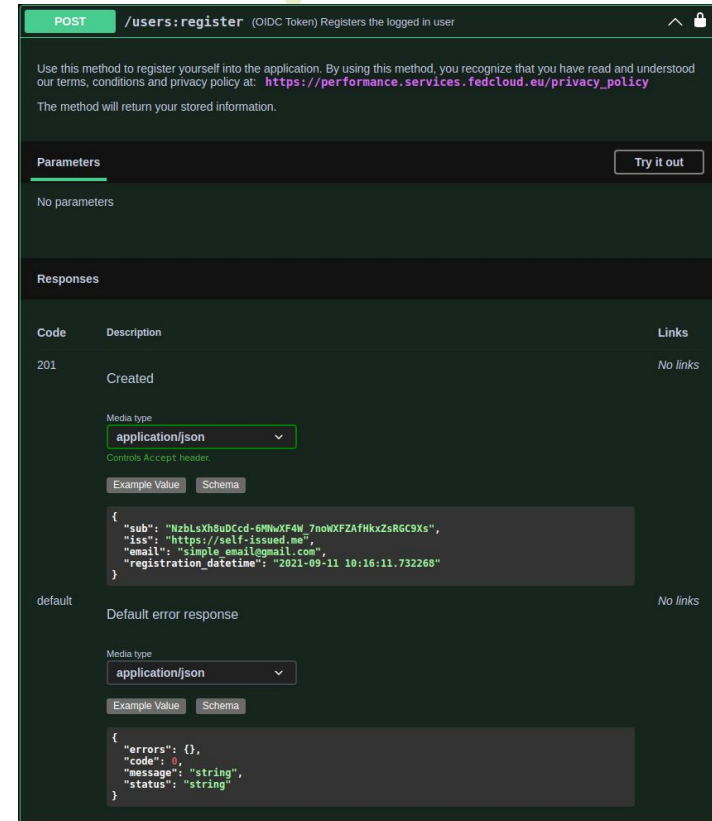
https://performance.services.fedcloud.eu/privacy_policy

Registration is very simple:

```
$ curl -X 'POST' \
  'https://performance.services.fedcloud.eu/api/v1/users:register' \
  -H 'accept: application/json' \
  -H "Authorization: Bearer $access_token"
```

Get access to:

- **Create;** Benchmarks, Results, Sites, Flavors, Tags
- **Report;** Results



POST /users:register (OIDC Token) Registers the logged in user

Use this method to register yourself into the application. By using this method, you recognize that you have read and understood our terms, conditions and privacy policy at: https://performance.services.fedcloud.eu/privacy_policy

The method will return your stored information.

Parameters Try it out

No parameters

Responses

Code	Description	Links
201	Created	No links

Media type: **application/json** Controls Accept header.

Example Value Schema

```
{
  "sub": "NzblxVh8uDCdd:6MhwVFAN_7noWKFZAfhkxZsRGc9Xs",
  "iss": "https://self-issued.me",
  "email": "simple_email@gmail.com",
  "registration_datetime": "2021-09-11 10:16:11.732268"
}
```

default No links

Default error response

Media type: **application/json**

Example Value Schema

```
{
  "errors": {},
  "code": 0,
  "message": "string",
  "status": "string"
}
```

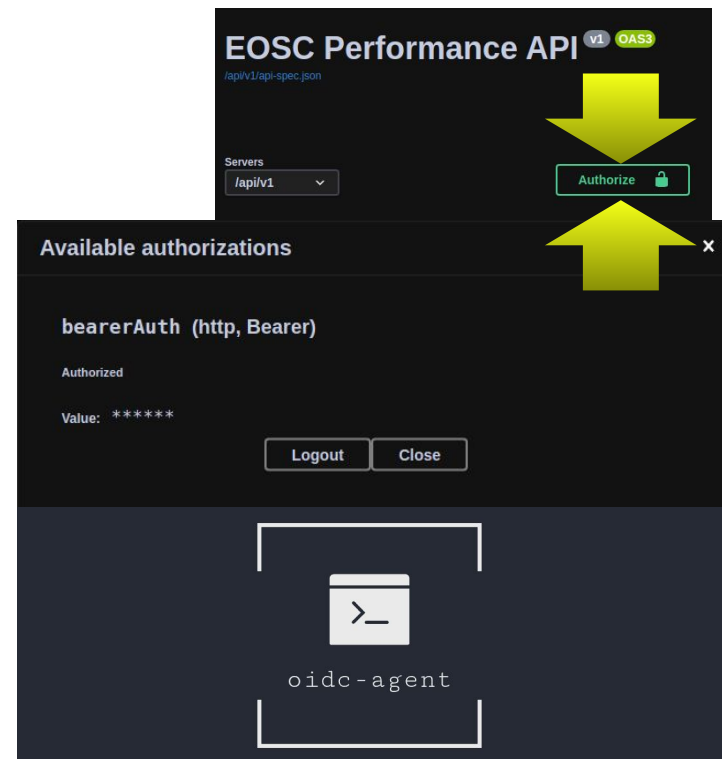
API - How to provide identification

To get your access token is just as simple as:

```
$ access_token=$(oidc-token egi-prod)
```

Full documentation about 'oidc-agent' can be found at:

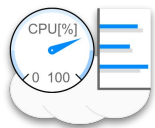
<https://indigo-dc.gitbooks.io/oidc-agent/>



Summary and Outlook

EOSC-Performance is the place to **compare** the **performance** of various EOSC sites

Core features available include:



- **Search** for benchmarks and results
- Result **filtering, comparison, and visualisation**
- **Adding** new results, benchmarks, sites
- **Admin** features
- **Full-featured API** to communicate with the platform

The new version is (just) [deployed](#) and available in the [EOSC-Marketplace](#)

Any **suggestions**? Looking forward to [your feedback](#)!

EOSC-Perf public links:

Detailed documentation: perf.readthedocs.io

Git organisation/repos: github.com/EOSC-Synergy/eosc-perf

EOSC-Marketplace: <https://marketplace.eosc-portal.eu/services/eosc-performance>

Or directly: <https://performance.services.fedcloud.eu/>

Contact perf-support@lists.kit.edu

Stay tuned!



EGI'21 Abstract