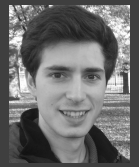
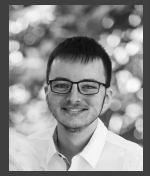
EOSC-Performance: find most suitable EOSC site for your task

EGI Conference 2022

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**

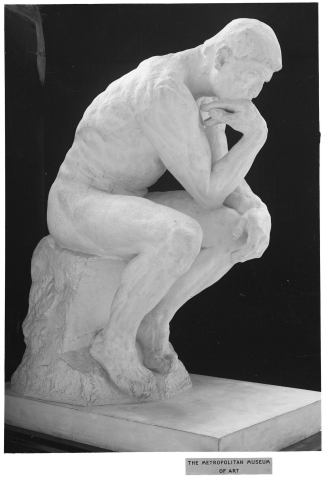
Spain, Portugal, UK, Czech Republic, Germany, Slovakia, Poland, Netherlands

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

*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:



US1

As: **a domain scientist**
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**

US2

As: **an advanced user / a resource provider**
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

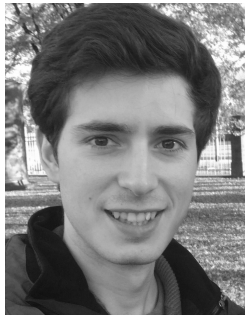
Key Developers

Christophe Laures

FRONTEND DEVELOPER

Marcus Hardt

PROJECT MANAGER



Borja Esteban Sanchis

BACKEND DEVELOPER

Valentin Kozlov

TECHNICAL LEAD

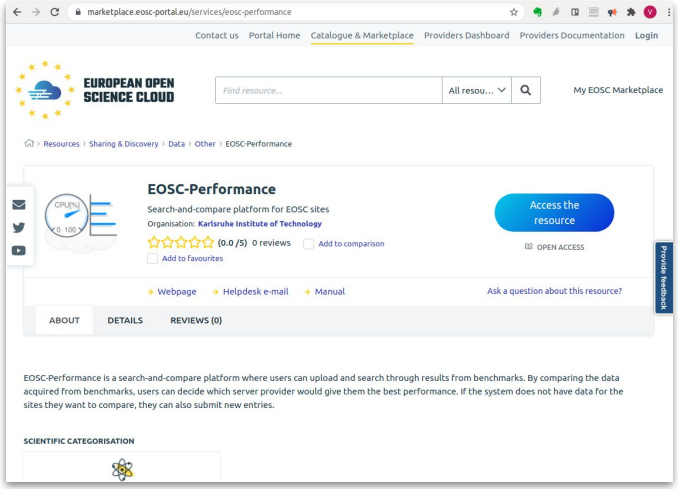
EOSC-Perf: in brief

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

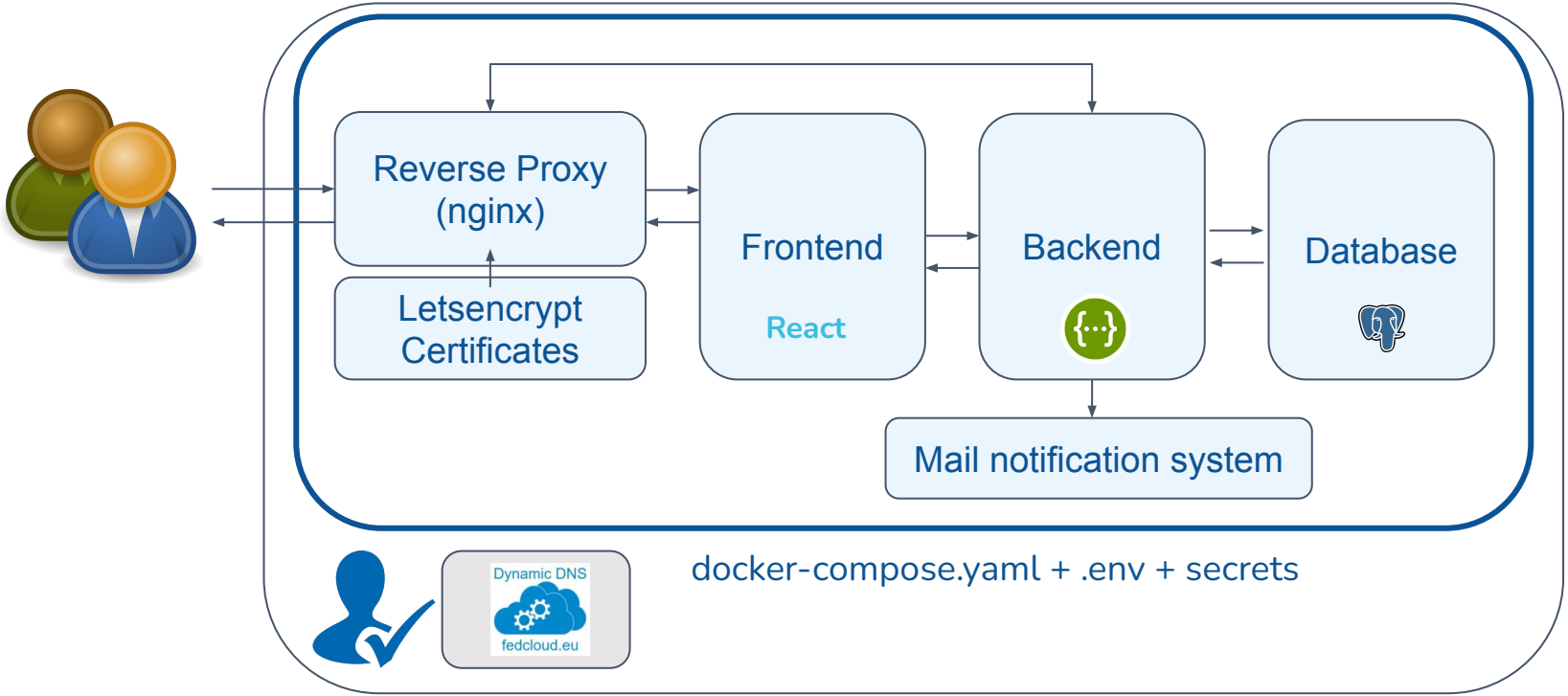
Benchmark: is a code packed in a Docker container, available on the Docker Hub, and produces JSON output.

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

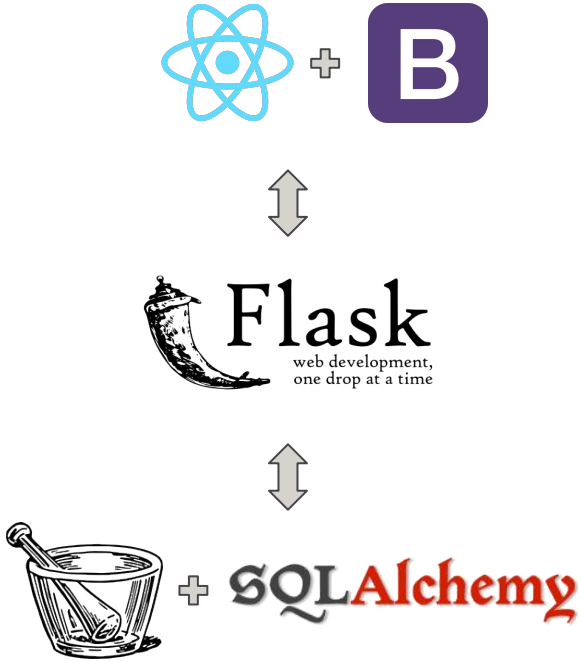
It is available in the EOSC-Marketplace:



EOSC-Perf: structure under the hood



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 Flaat, OIDC and EGI-Check-In

Database/Model:

- Powered by SQLAlchemy
- PostgreSQL

Web Frontend

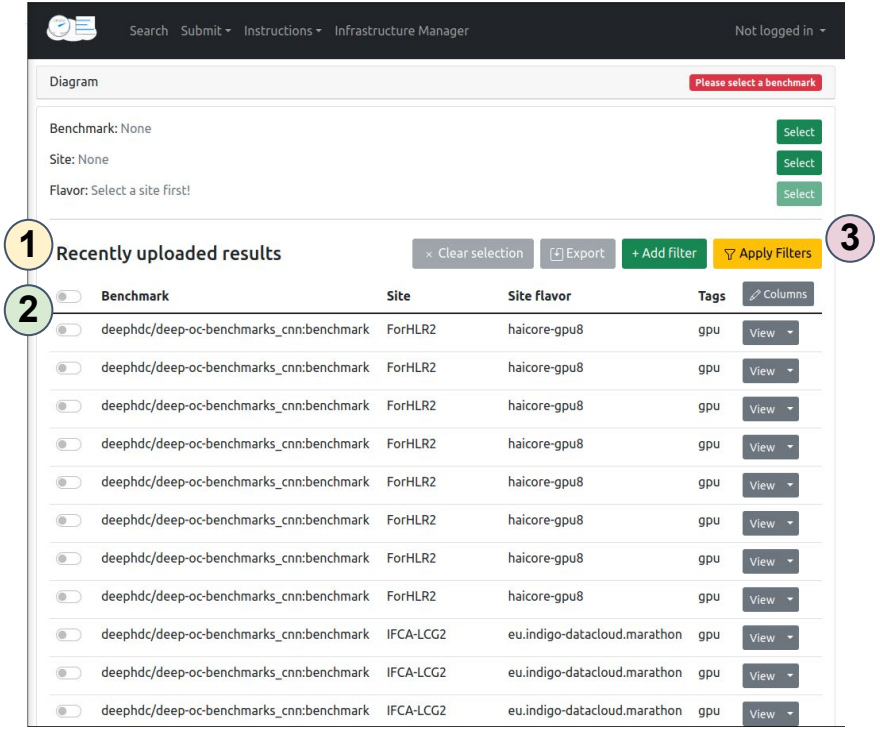
Note: This demo will use a non production endpoint for the demonstration: https://perf-stage.test.fedcloud.eu

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



Diagram

Please select a benchmark

Benchmark: None

Site: None

Flavor: Select a site first!

Recently uploaded results

Benchmark	Site	Site flavor	Tags
deephdc/deep-oc-benchmarks_cnn:benchmark	ForHLR2	haicore-gpu8	gpu
deephdc/deep-oc-benchmarks_cnn:benchmark	ForHLR2	haicore-gpu8	gpu
deephdc/deep-oc-benchmarks_cnn:benchmark	ForHLR2	haicore-gpu8	gpu
deephdc/deep-oc-benchmarks_cnn:benchmark	ForHLR2	haicore-gpu8	gpu
deephdc/deep-oc-benchmarks_cnn:benchmark	ForHLR2	haicore-gpu8	gpu
deephdc/deep-oc-benchmarks_cnn:benchmark	ForHLR2	haicore-gpu8	gpu
deephdc/deep-oc-benchmarks_cnn:benchmark	ForHLR2	haicore-gpu8	gpu
deephdc/deep-oc-benchmarks_cnn:benchmark	ForHLR2	haicore-gpu8	gpu
deephdc/deep-oc-benchmarks_cnn:benchmark	IFCA-LCG2	eu.indigo-datacloud.marathon	gpu
deephdc/deep-oc-benchmarks_cnn:benchmark	IFCA-LCG2	eu.indigo-datacloud.marathon	gpu
deephdc/deep-oc-benchmarks_cnn:benchmark	IFCA-LCG2	eu.indigo-datacloud.marathon	gpu

EOSC-Perf: Supported Flows

US1: well-structured and searchable information

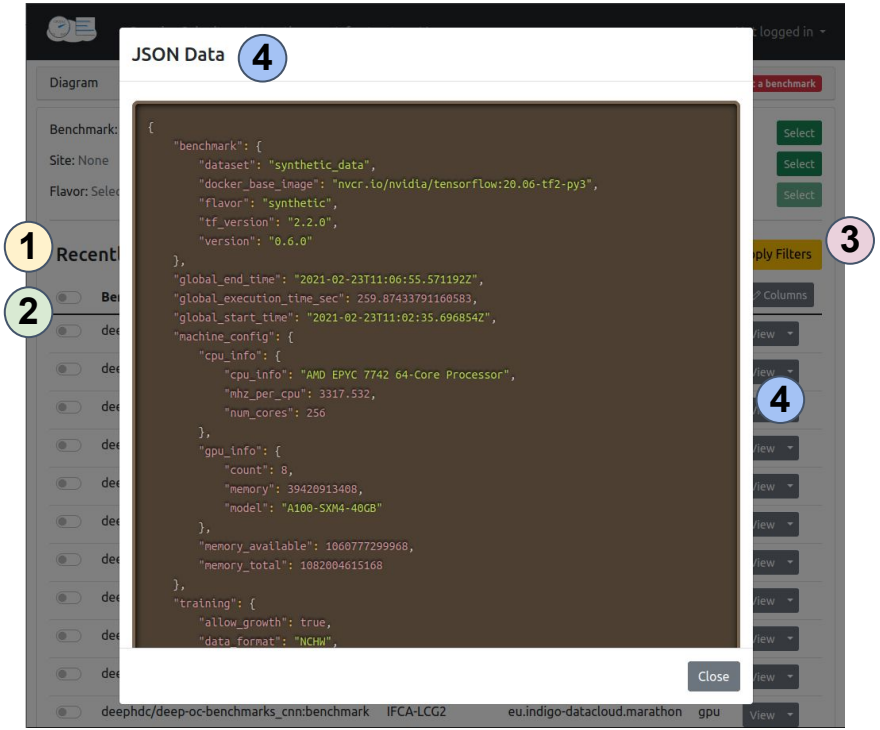
Any user can:

Browse the existing results (1)

Compare (table, plotting) (2)

Filter them (3)

View full JSON output (4)



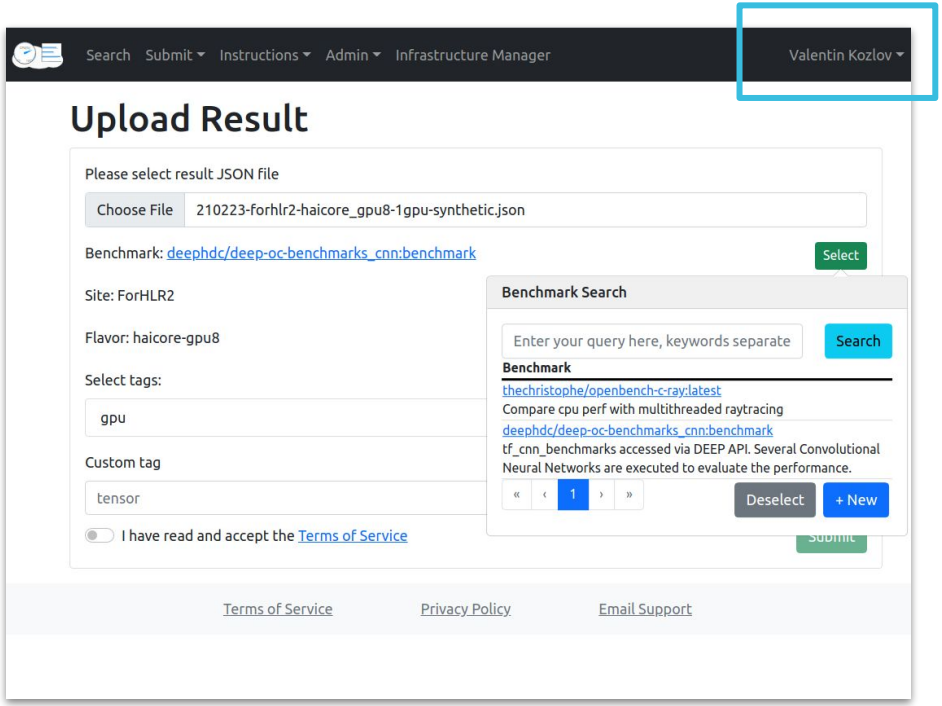
EOSC-Perf: Supported Flows

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:

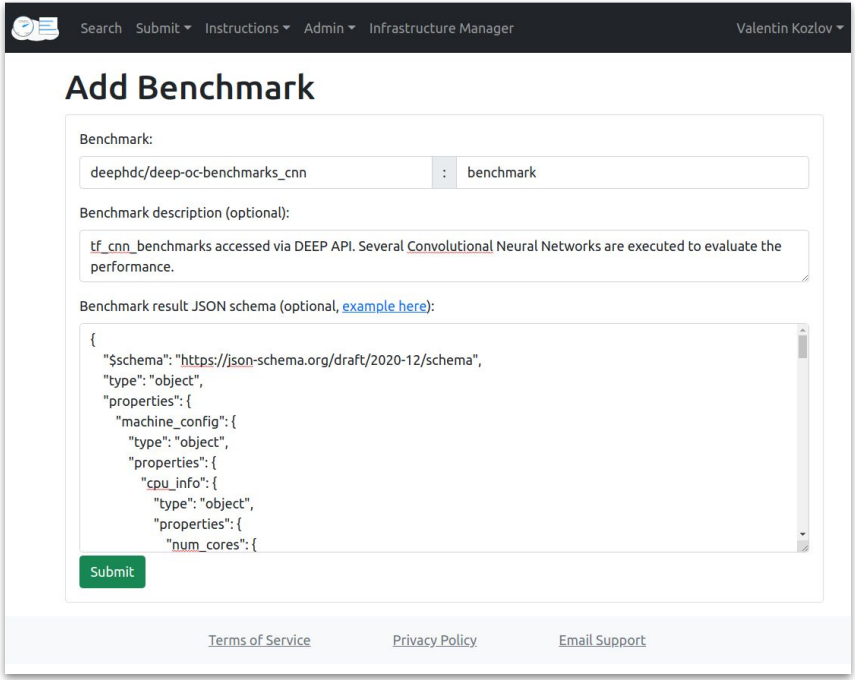
- Benchmark:

  Dockerhub image + docker tag

- Benchmark description

- JSON schema

  (may include "suggestToUser" key)



EOSC-Perf: Supported Flows

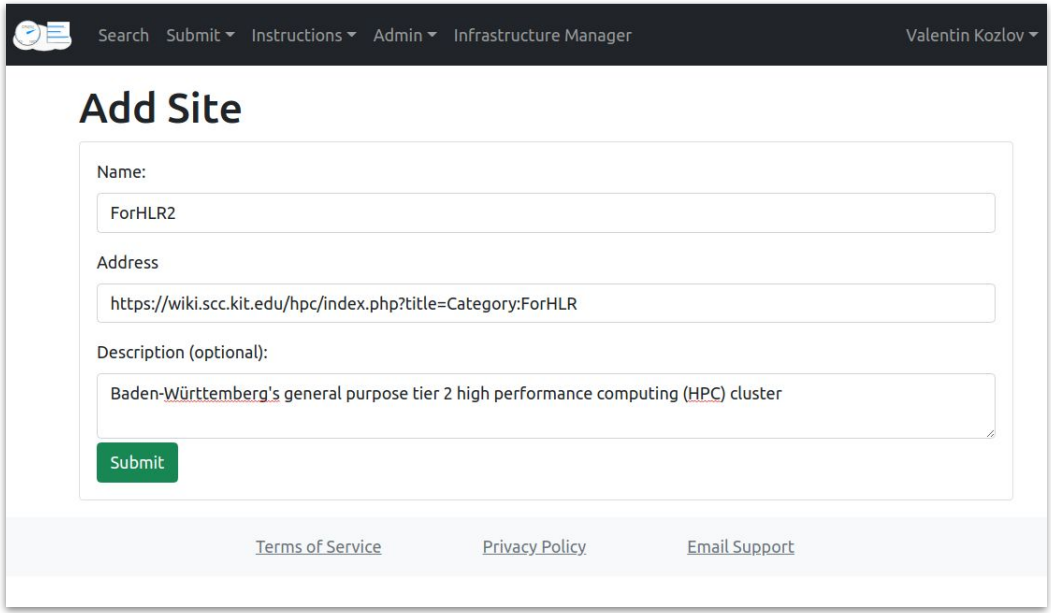
US2: store results of
any benchmark of interest

Contributors (⇨ authentication):

Add a new resource:

- Name
- Address
- Description

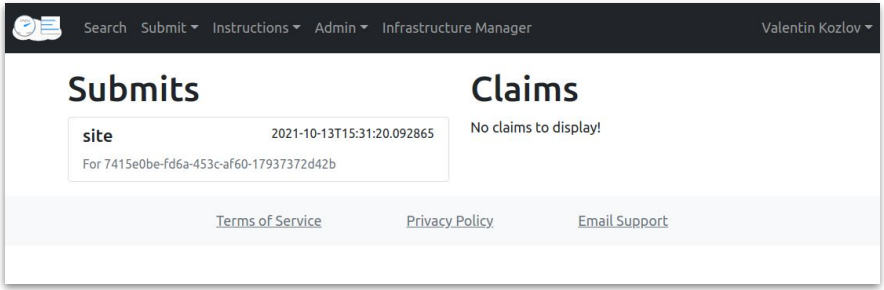
N.B.: Adding Site / Flavor is also possible during “Upload Result”



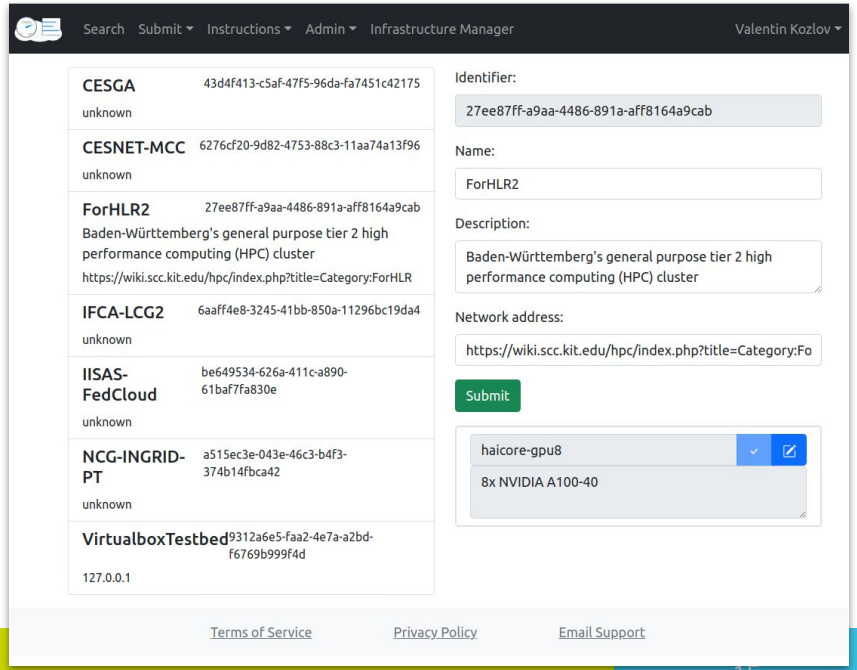
EOSC-Perf: Supported Flows

Admin functions:

Review Submissions and Reports



Edit sites with “Site Editor”



EOSC-Perf: more features

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

API

Note: This demo will use a non production endpoint for the demonstration: https://perf-stage.test.fedcloud.eu

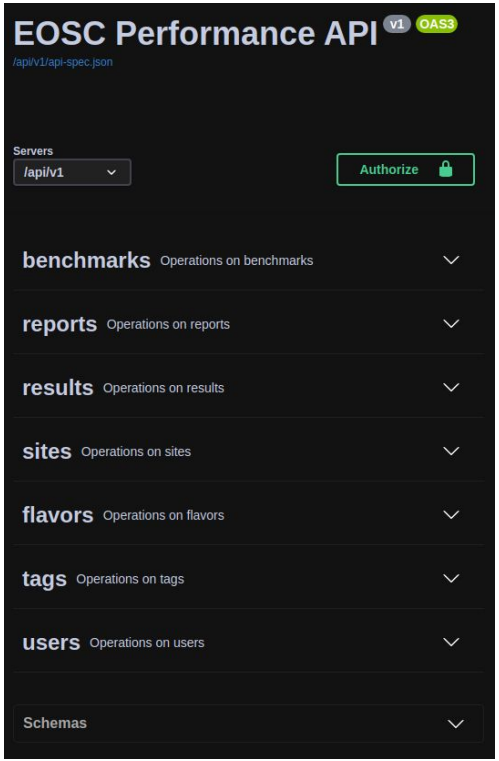
API - Powered by Swagger and OpenAPI

Based on OpenAPI v3

- 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.

Set of demo notebooks:
https://github.com/EOSC-synergy/eosc-perf-backend/blob/main/examples/

API - To contribute, register

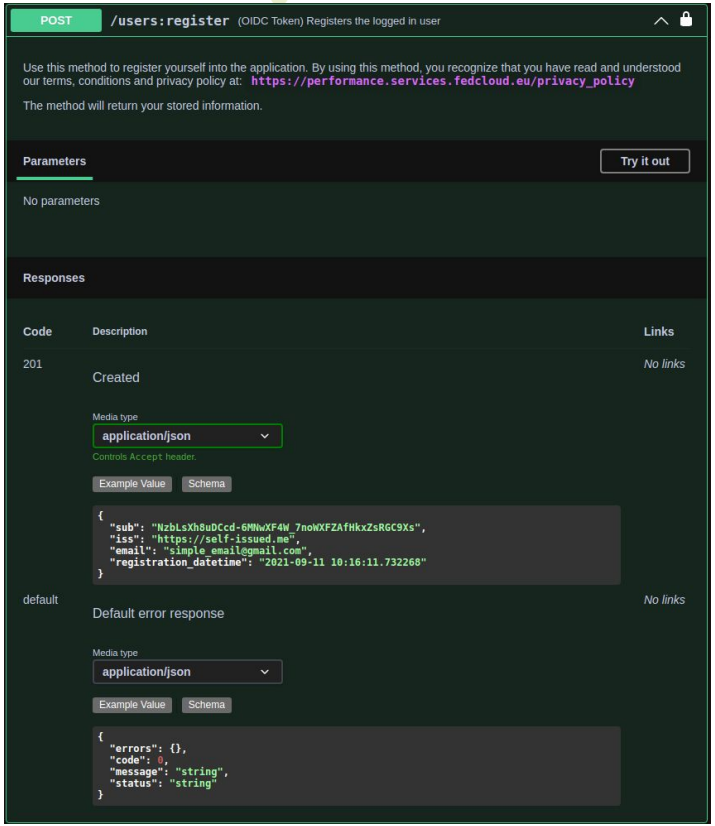
Please, read our privacy policy at:
https://performance.services.fedcloud.eu/privacy_policy

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



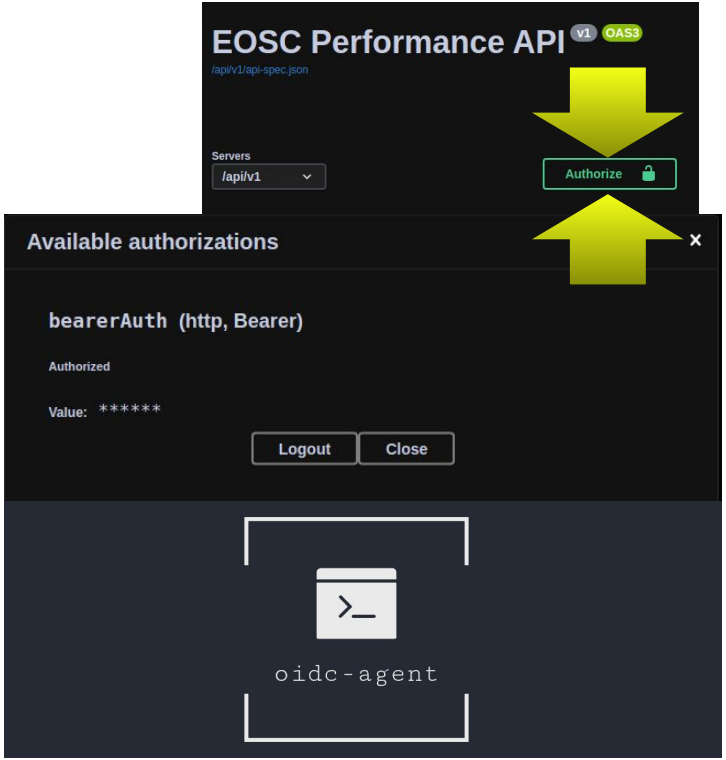
API - How to provide identification

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:

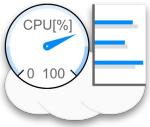
https://indigo-dc.gitbooks.io/oidc-agent/



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

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'22 Abstract