Simulation Data Lake

Training Session

CINECA

9 September 2025

Geo-INQUIRE *is funded by the European Union. Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the European Research Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.*

Agenda

1. **Origins**: Where We Started
2. **Key** characteristics of CINECA's SDL
3. **How** to Use the SDL:
   - Web Portal
   - Command Line Interface (CLI)
   - SDK
   - API
4. **Practical Use Cases**
5. **Q&A** Session and Discussion

1. Origins: Where We Started

At the beginning of the Geo-INQUIRE project

Cineca was asked to provide a data lake

that could meet the following partners **requirements**:

- Store and make accessible the **simulation data generated throughout the project**.
- Support the storage of **experiments** consisting **of hundreds of thousands of files**, with total volumes reaching **terabytes (TB)**.
- Include the information needed to re-run simulations.
- Promote data discoverability and reuse in alignment with the **FAIR** principles.
- Coordination with **DT-GEO** and **EPOS**.

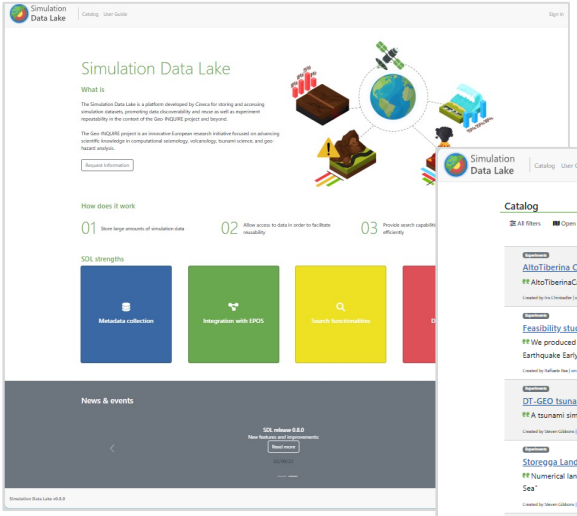
1. Origins: Where We Started

Cineca conducted a **brief review** of existing platforms but found that none fully met the key requirements, due to the following limitations:

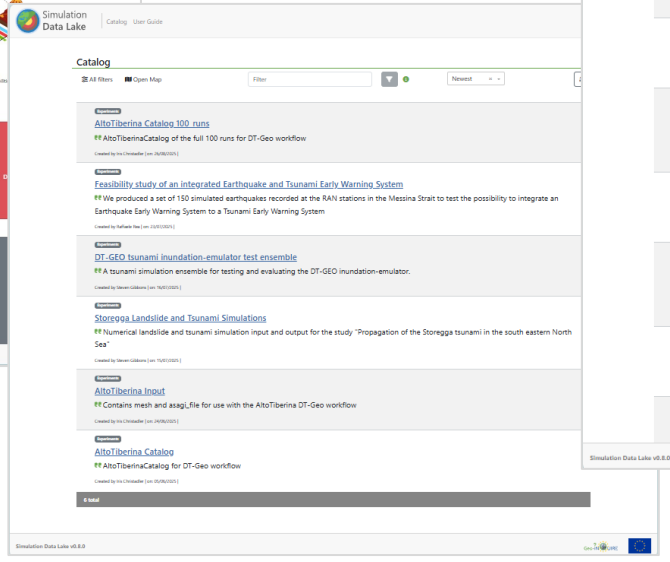
- Max **experiment size** insufficient (20GB/50GB).
- Max individual **file size** insufficient (10GB/50GB).
- Inadequate support for **folder and subfolder structures**.
- Limited **API capabilities for large file** upload and download.

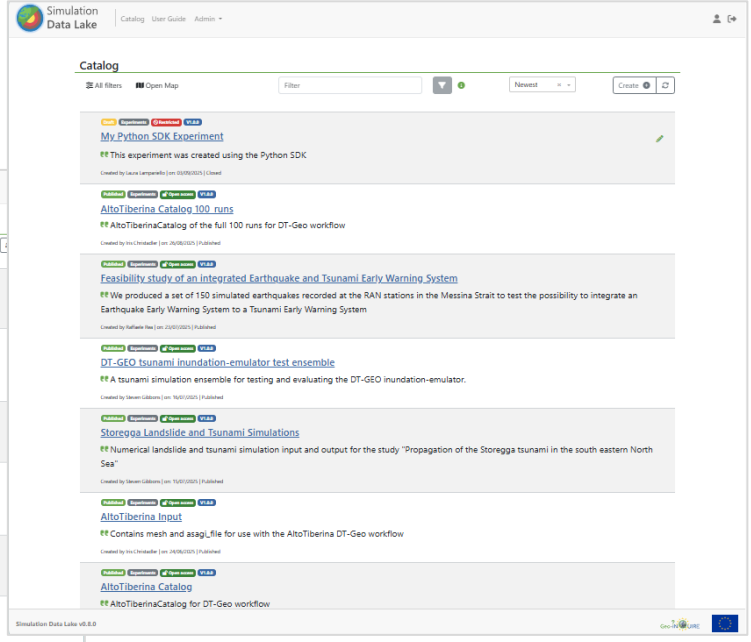
As a result, Cineca decided to **implement a new platform** on its own infrastructure, specifically designed to meet the **partners' requirements** and support the **storage of simulations generated within the Geo-INQUIRE project**.

This led to the implementation of the Simulation Data Lake



SDL v0.8.0





2. Key characteristics of the SDL

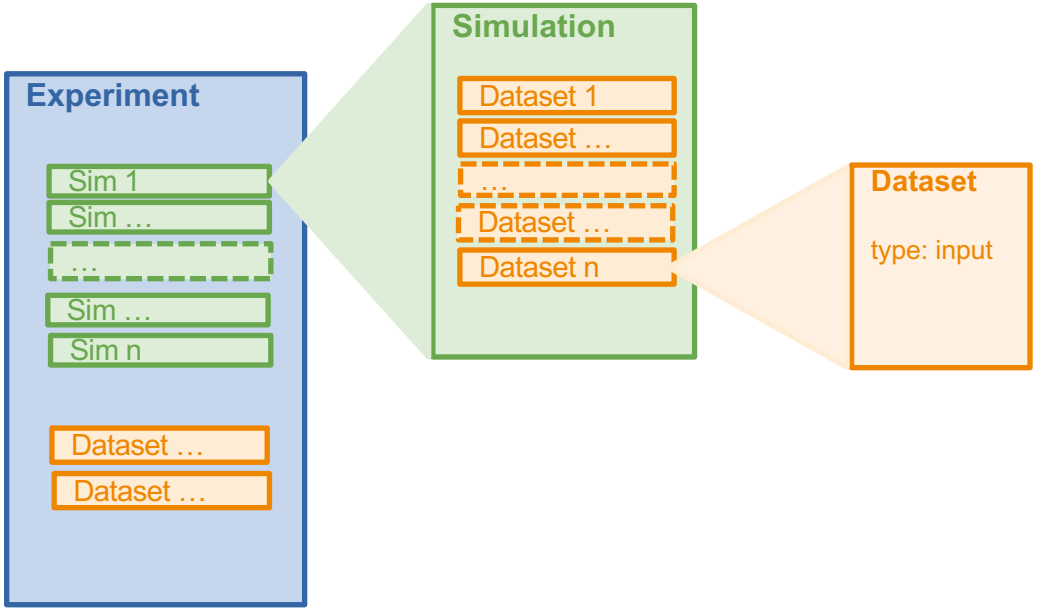
- Store large amounts of simulation data
- Uploading and downloading of huge experiments both data and metadata
- Advanced search functionalities including search by spatial and temporal coverage
- Experiment publication with Access Policy control
- DOI minting and management
- Integration with EPOS Data Portal
- Workflow management (CWL files and diagram visualisation)
- Dashboard of KPIs: automated FAIR assessment with F-UJI tool
- Data proximity and integration with CINECA HPC systems
- OGC Services

Data Model

- Main Entities:
  - Experiment
  - Simulation
  - Workflow
  - Dataset

- A Dataset can be of the following types:
  - Input
  - Output
  - Data product
  - Input parameter

- A Simulation is the execution of a Workflow and contains input data and the result of that specific workflow run



Mapping the data/metadata to the SDL data model and v.v.

Input metadata in **json** format with **schema** described at the following link:

https://sdl.hpc.cineca.it/api/bulk-import/schema?ajvStrict=true

Or using a RO-Crate metadata file

The metadata file is output in:

- JSON format
- RO-Crate format to improve fairness aspects especially interoperability

FAIR

Findable Accessible Interoperable Reusable

- Saves time & resources (avoid duplication)
- Boosts collaboration & innovation across fields
- Strengthens reproducibility & trust
- Extends long-term impact of research investments
- Promotes equity by making knowledge globally accessible

SDL FAIR details

Findable:

- Searchable metadata catalog: Users can easily discover datasets using keywords, text search, or other filters.
- SDL catalog explorable using:
  - Web application
  - CLI
  - Python module
  - REST API
  - EPOS Data Platform (coming soon)
- DOI assignment: Enables citation and persistent access to datasets.

SDL FAIR details

Accessible:

- Clear access policies
- Standardized resource download mechanism: makes it easy to retrieve data programmatically or manually

SDL FAIR details

Interoperable:

- Standard/well known formats: users can work with data using familiar tools
- RO-Crate packaging: ensures metadata and data are bundled in a machine-readable, structured way

SDL FAIR details

Reusable:

- Rich metadata: users understand the context, origin, and structure of the data
- Clear licensing: users know how they can use the data

FAIRNESS

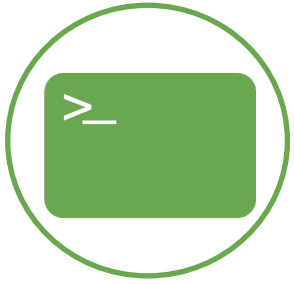
- Automated periodic FAIR assessment
  - Ensures continuous monitoring and improvement
  - F-UJI tool used to assess experiments FAIRness

- Exposition of the metadata in RO-Crate
  - Lightweight, machine-readable metadata format
  - Facilitates interoperability and reusability
  - Widely adopted in research data management
  - Allows for CWL workflows representations using profiles such as Workflow RO-Crate

3. How to Use the SDL

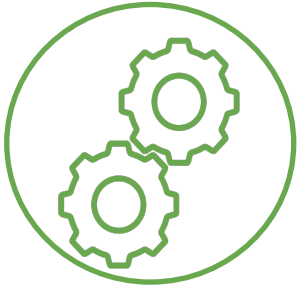
WEB PORTAL

- User friendly interface
- Simple access to data
- For non-technical users



CLI + Python SDK

- Interact programmatically
- For large quantities of experiments, simulations and datasets
- For technical users



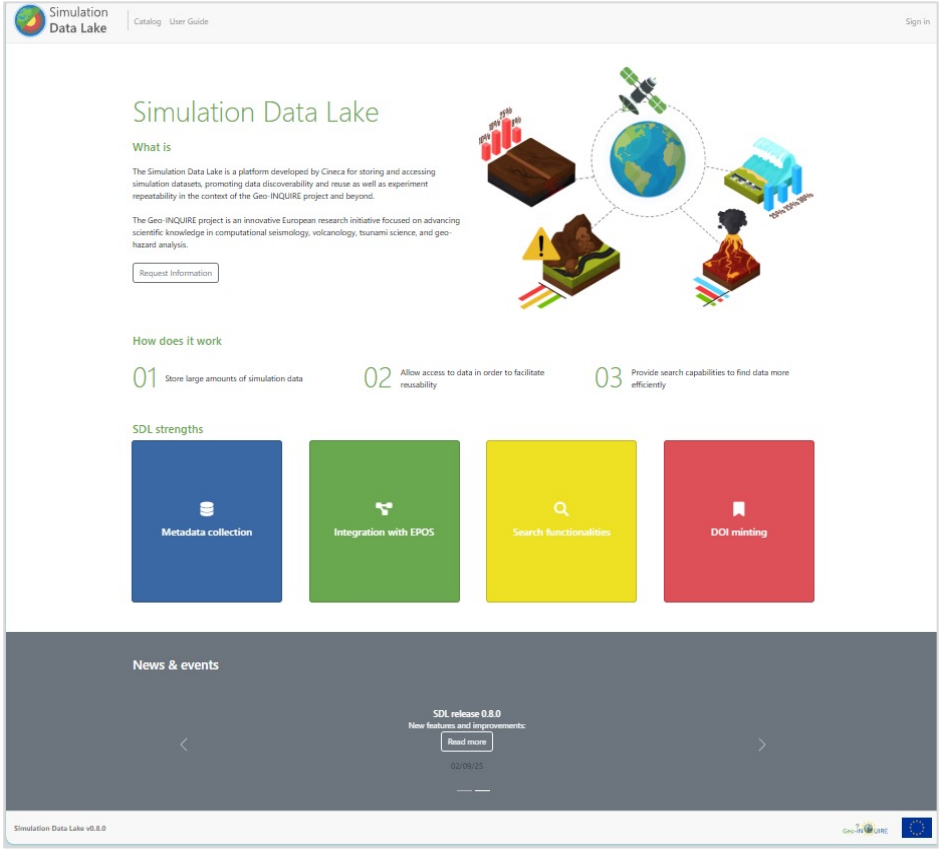
API

- Interact programmatically
- Integration with other services
- For technical users

1. Using the SDL: Web Portal

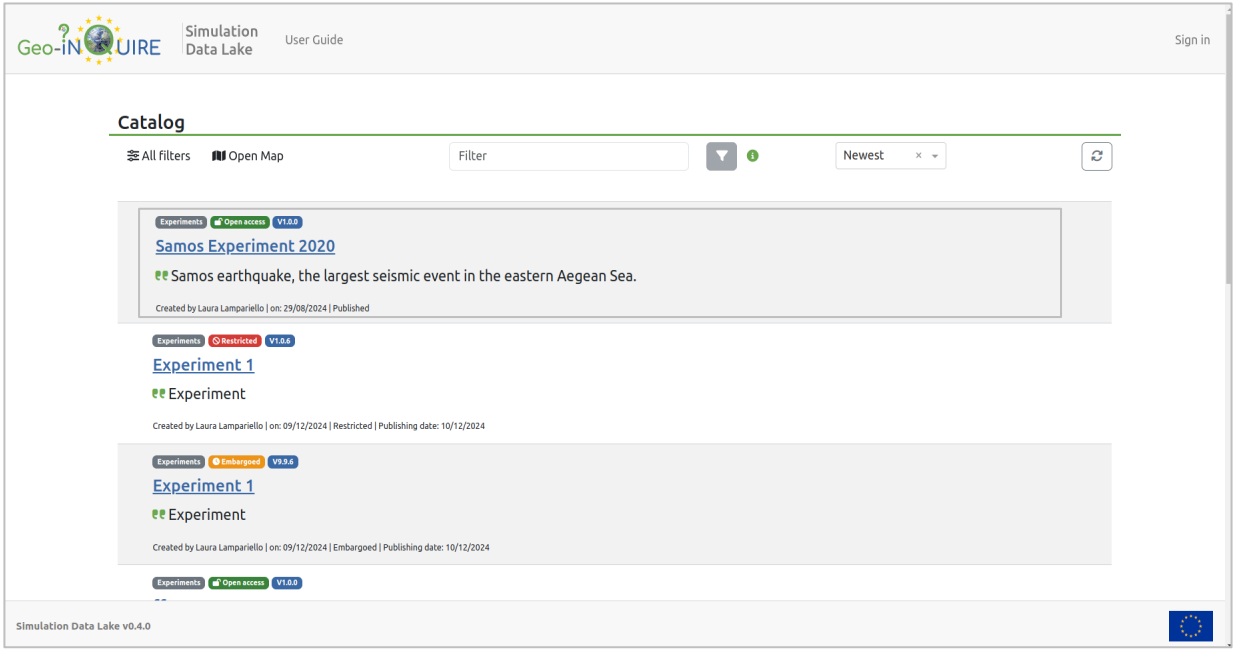
SDL Home Page

- Useful links
- Latest news
- Documentation



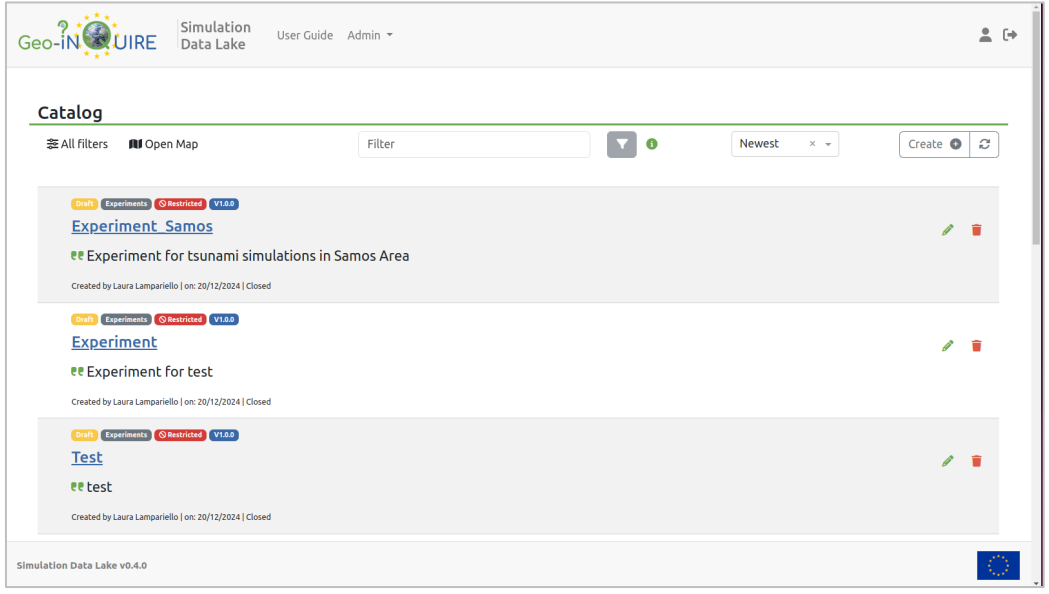
Web portal: Public Catalog

- Everyone can access
- It is possible to see the list of:
  - Experiments
  - Simulations
  - Datasets
- Only published experiments are visible
- The experiment can be published with some policies:
  - Restricted access
  - Embargoed
  - Public access
- Available only limited features



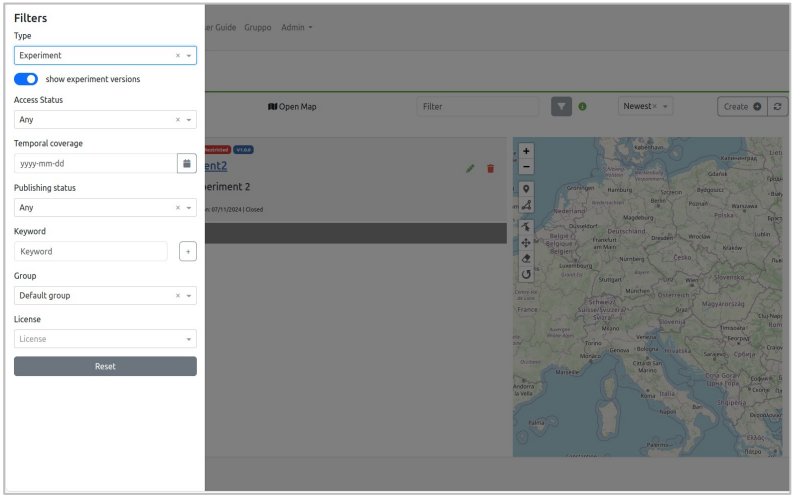
Web portal: Catalog

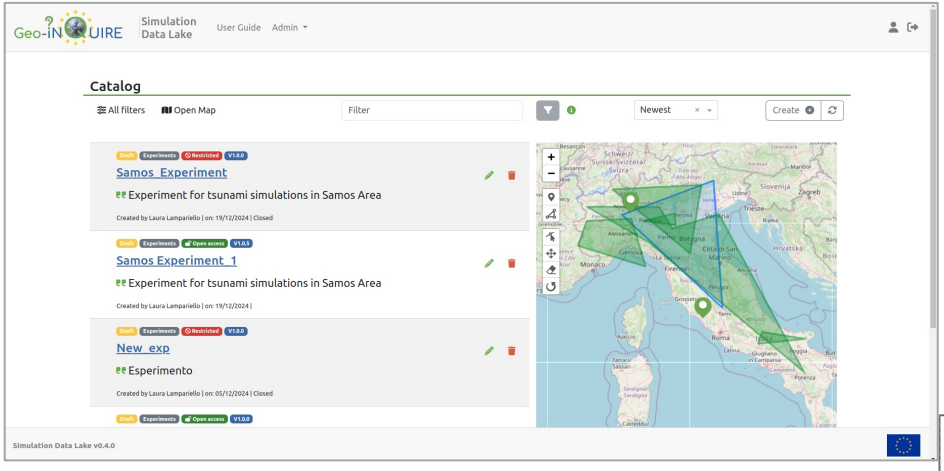
- Logged users can see also non published experiments
- Unpublished experiments can be seen by:
  - the creator
  - users who are part of the same group as the creator
  - the collaborators
- The catalog is paginated, the list of elements is divided over several pages



Web portal: Catalog Filters

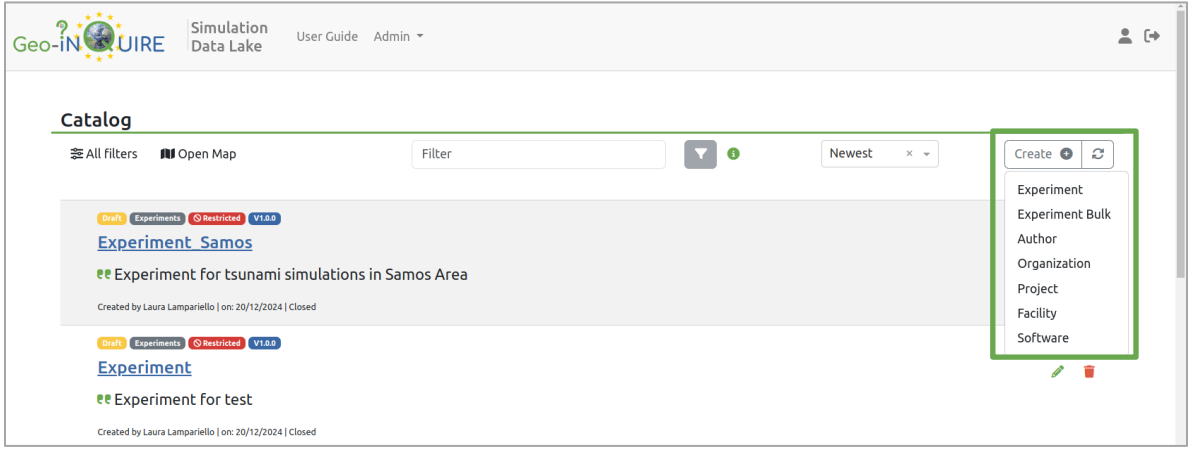
- Filtering by **data type**: Experiment, Simulation, Dataset
- Show/hide **versions of experiments**
- Filtering by **Access Status** (Open Access, Restricted, Embargoed, Any), **Temporal coverage**, **Keywords** and **License** type
- Only logged users can filter by:
  - **Publishing status** (Published, Draft, Any): whether an experiment has been published or not
  - **Group**: group with which the entity is associated
- Using the map, it is possible to select an area or a point of interest



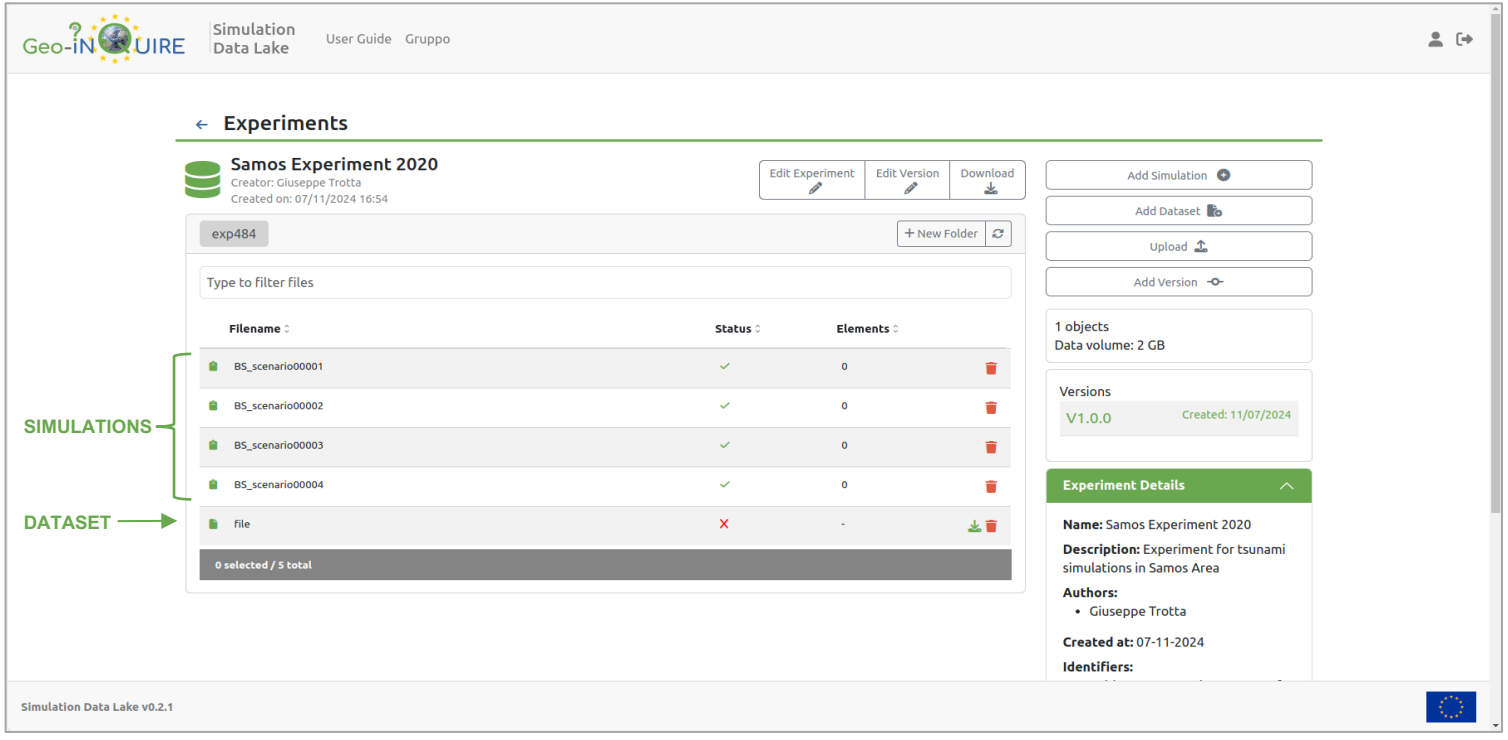


Web portal: Create other entities

- Only for logged users
- There is the possibility to create:
  - Experiment
  - Experiment through a bulk upload
  - Author
  - Organization
  - Project
  - Facility
  - Software

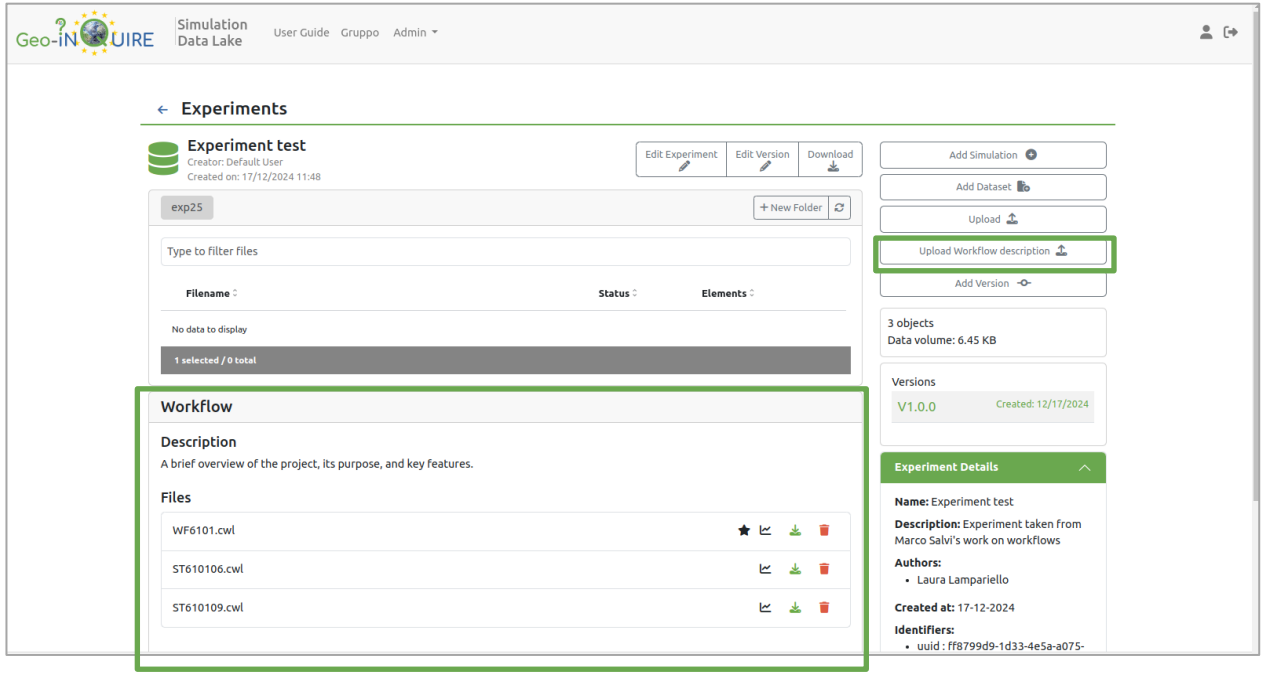


Web portal: Experiment view



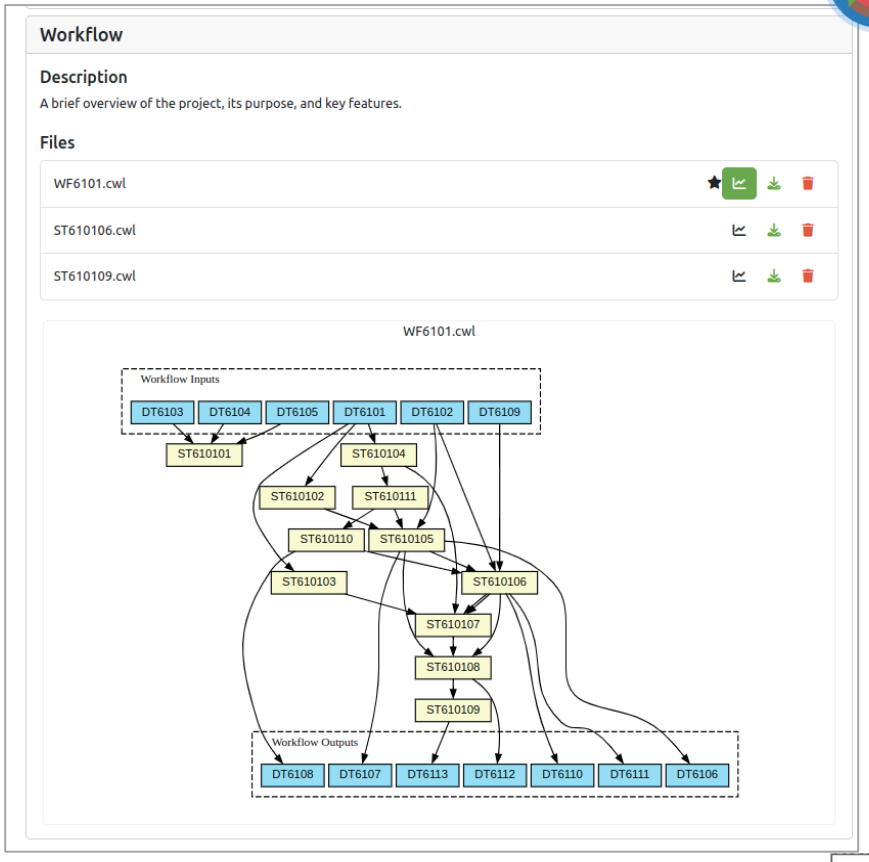
Web portal: Workflows

- The current workflow implementation allows users to create multiple workflows. Each can have multiple workflow description files associated with them (**CWL** or other formats)
- Workflows can be associated with multiple Simulations (workflow executions)
- We shared the information of workflow in the same format (cwl) of DTGEO and EPOS.



Web portal: Workflows

- The workflow files are uploaded using the appropriate form
- It is possible to set a **primary** workflow file via the star logo
- Graphical representation of the **CWL** files using the API of Common Workflow Language standard
- Download of the single workflow file
- Deletion of the single workflow file



2. Using the SDL: Command Line Interface

Command Line Interface (CLI): Getting started

- SDLCTL (Simulation Data Lake ConTroL)
- You can install it using two methods:
  - a remote script
  - in a virtual enviroment
- In addition to the command-line interface, SDL Control provides a Python SDK (`pysdl`) that allows you to interact with the Simulation Data Lake programmatically in your Python applications
- You can find the details of the steps to follow for installation at the following documentation link: https://sdlctl.readthedocs.io/latest/getting_started/installation/

Command Line Interface (CLI): Commands

Look for experiments

```
(.venv) gtrotta@NGTROTTA0205932:~/Projects/geo-inquire/sdlctl$ sdlctl experiment list
                                    Experiments
```

ID	Name	Description	Created At
44	Cinzia test	Cinzia experiment for metadata ONLY: pls do NOT load any content	2024-05-03 14:16:45
30	Tsuanmi GC12	An example experiment.	2024-04-11 10:23:32
26	Lucia	Tsunami silumations	2024-04-05 14:52:31
23	Davide's Experiment	Testing upload service	2024-03-04 13:47:09
21	New CLI Experiment	This is another stupid test.	2024-02-29 18:04:03
12	Another Experiment	A test experiment after some refactoring.	2024-02-28 09:48:47
11	My second CLI experiment	This is another experiment created with the CLI in a more fancy way.	2024-02-27 18:10:41
9	My Experiment	Just a simple test.	2024-02-27 13:19:41
7	test	My first test experiment	2024-02-01 22:14:37

```
                                    Press 'q' di quit.
```

Delete confirmation

```
 % sdlctl file upload                                          ~ NMPUCCINI205808
Experiment ID: 12
Filename: sample50M
  Calculating MD5... ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 100% 0:00:00

Starting new upload of file: sample50M

  Calculating MD5... ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 100% 0:00:00
  Uploading...       ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 100% 0:00:00

Upload completed successfully!

 % sdlctl file delete                                          ~ NMPUCCINI205808
Experiment ID: 12
Filename: sample50M
Are you sure you want to delete the file? [y/N]:
```

Upload resume

```
 % sdlctl file upload -id 12 -f sample50M                      ~ NMPUCCINI205808
  Calculating MD5... ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 100% 0:00:00

Starting new upload of file: sample50M

  Calculating MD5... ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 100% 0:00:00
  Uploading...       ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━  30% 0:00:05^C
Upload interrupted by user.
  Calculating MD5... ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 100% 0:00:00
  Uploading...       ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━  40% 0:00:04
 % sdlctl file upload -id 12 -f sample50M                      ~ NMPUCCINI205808
  Calculating MD5... ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 100% 0:00:00

Resuming upload of file: sample50M

  Calculating MD5... ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 100% 0:00:00
  Uploading...       ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 100% 0:00:00

Upload completed successfully!

 %                                                             ~ NMPUCCINI205808
```

See the following documentation for the complete list of commands: https://sdlctl.readthedocs.io/latest/reference/commands/

Command Line Interface (CLI): basic commands

Get the current user status	**sdlctl user status**
Login to the SDL	**sdlctl user login**
List of experiments	**sdlctl experiment list**
Creating an experiment	**sdlctl experiment create** –n <exp_name> -des <exp_description> -a <id_author>
Get the detailed information of that experiment	**sdlctl experiment get** –id <exp_id> -v <exp_version>

Command Line Interface (CLI): basic commands

- Creating a new version of an experiment
- List of datasets of an experiment version
- Creating a dataset

sdlctl experiment version add –id <exp_id> -v <newexp_version>

sdlctl experiment dataset list –id <exp_id> -v <exp_version>

sdlctl experiment dataset add –id <exp_id> -v <exp_version> -n <ds_name> -d <ds_descr> -t <ds_type> -l <ds_licence>

- See the following documentation for the complete list of commands: https://sdlctl.readthedocs.io/latest/reference/commands/

3. Using the SDL: API

API specs

OpenAPI Specification
https://sdl.hpc.cineca.it/api/docs

upload

- POST `/api/experiments/{experiment_id}/init-upload/{filename}` init upload of a file
- POST `/api/experiments/{experiment_id}/upload` upload file
- POST `/api/experiments/{experiment_id}/complete-upload/{upload_id}/{key}` complete upload

download

- POST `/api/experiments/{experiment_id}/download/{filename}` download a file
- GET `/api/experiments/{experiment_id}/init-download/{filename}` Initialize download of a file

experiment

- POST `/api/experiments` Create a new experiment
- GET `/api/experiments` Get experiments
- GET `/api/experiments/{experiment_id}` Get an experiment by id
- DELETE `/api/experiments/{experiment_id}` Delete an experiment
- PATCH `/api/experiments/{experiment_id}` Update experiment data
- DELETE `/api/experiments/{experiment_id}/files` Delete files whose name begin with a certain prefix in a given experiment (if path ends with '/' it will look for a directory to delete, otherwise, a file)
- GET `/api/experiments/{experiment_id}/files` Get list of files for an experiment
- POST `/api/experiments/{experiment_id}/collaborators` Add a collaborator to the experiment

simulation

- GET `/api/experiments/{experiment_id}/simulations` Get all simulations of an experiment
- POST `/api/experiments/{experiment_id}/simulations` Create a new simulation
- PATCH `/api/experiments/{experiment_id}/simulations` Modify a new simulation

4. Practical use cases

How to create a new experiment?

- First possibility
  1. I upload metadata
  2. Then I upload data (files,…)

- Second possibility
  1. I upload data (files,…)
  2. Then I upload metadata

First possibility

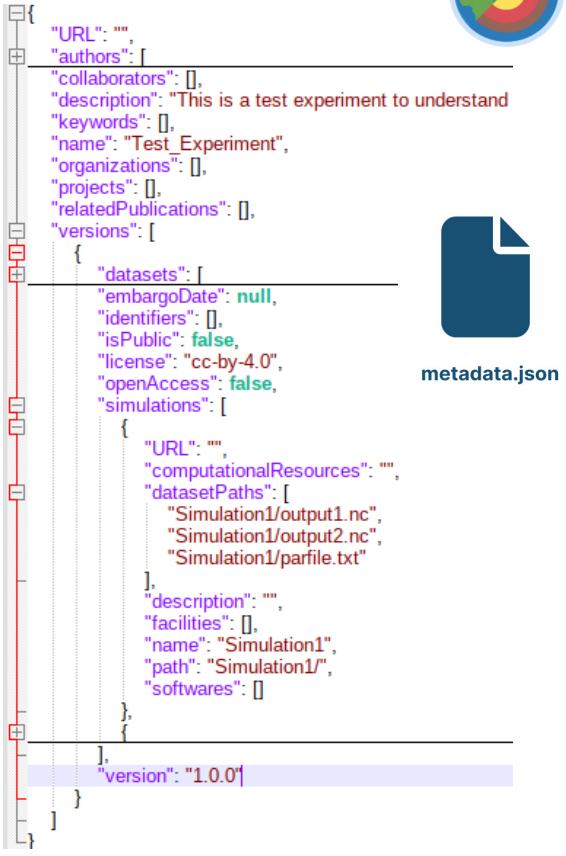
1. I upload metadata
   - Use Web App (create entities by hand or using Experiment bulk)
   - Use CLI or SDK (for more experts users)

   (For help creating the metadata file, use the Wizard)

2. Then I upload data (files,…)
   - Drag and drop from the Web App
   - Upload function using CLI or SDK

Metadata file generation

- Contains all metadata of the experiment
- In a JSON format whose schema is described at the following link: https://sdl-dev.hpc.cineca.it/api/bulk-import/schema?ajvStrict=true
- Complete the JSON fields for the small-scale experiment by hand
- It is possible to obtain support in compiling files with the help of a json editor Documentation of the steps: https://sdl-userguide.readthedocs.io/tutorials/ExperimentBulk/
- Each researcher can build his own custom script for generating the metadata file for his own experiment
- Wizard tool for very large experiments



Wizard

- Help researchers generate the metadata file
- Through a guided procedure, it automatically generate an experiment metadata JSON file by analysing the structure of an input folder.
- It maps folder hierarchy to metadata, detects patterns in files and folders to minimize repetitive input, and guides users through the classification process.
- It is intended to be used with high item count for better results.
- Documentation:
  - https://sdlctl.readthedocs.io/latest/getting_started/wizard/
  - https://sdlctl.readthedocs.io/latest/tutorials/5-wizard/

Complete process of ingestion using Wizard

I

metadata.json

sdlctl experiment bulk -f metadata.json

- CREATION OF THE EXPERIMENT
- METADATA UPLOAD

II

FOLDER

sdlctl file bulk -id <experiment-id> -lp <path of the experiment root folder>

- FILES AND FOLDERS UPLOAD

Second possibility

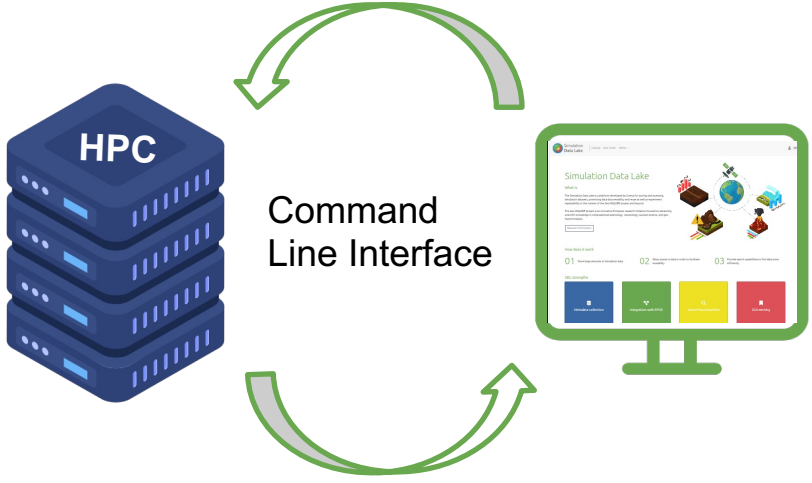
1. I upload data (files,…)
   - Drag and drop from the Web App
   - Upload function using CLI or SDK

2. Then I upload metadata
   - Use Web App (add metadata by hand but for a few entities)
   - Use Cli or SDK (for technical users)

Not recommended for huge experiments

Load data from HPC to SDL

- To upload data to SDL that resides on HPC machines, you need to use the CLI
- Install the latest version of the CLI on the HPC machine where the data is located
- Invoke the data transfer command to upload the files
- Documentation: https://sdlctl.readthedocs.io/latest/getting_started/installation/#install-on-cineca-hpc-systems



How to modify an existing experiment?

To modify entities of an existing experiment, you have the following options:

- **Manually using the graphical interface** – recommended for its intuitiveness and useful when there are only a few entities.

- **Using the Command Line Interface (CLI) or Python SDK** – with the SDK, you can create a Python script to loop through commands to modify multiple entities.
  - Documentation: https://sdlctl.readthedocs.io/latest/sdk/overview/

- **Using API experiment bulk put** giving in input the modified metadata file

Useful information

- Dev: https://sdl-dev.hpc.cineca.it
  - Development environment with 500GB of storage
  - It is the most updated enviroment with features in development
- Prod: https://sdl.hpc.cineca.it
  - Production environment with 500TB of storage
- Contacts: sdl@cineca.it
- Request an SDL account using the following link: https://sdl-userguide.readthedocs.io/tutorials/Introduction/
- Documentation
  - Web Portal: https://sdl.hpc.cineca.it/docs
  - Command Line Interface and SDK: https://sdl.hpc.cineca.it/cli/docs

5. Question & Answer session and discussion

Thank you for your attention

Geo-INQUIRE is a joint effort of 51 institutions

Geo-INQUIRE is funded by the European Commission under project number 101058518 within the HORIZON-INFRA-2021-SERV-01 call.