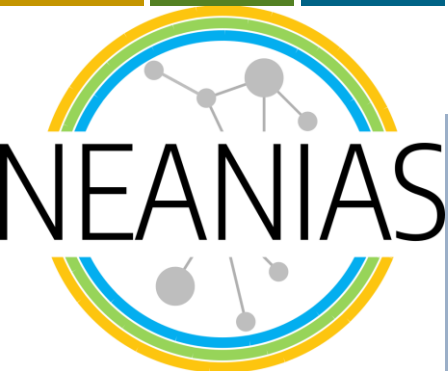


# NEANIAS WP06 – Core Services

Attila Farkas, SZTAKI  
[attila.farkas@sztaki.hu](mailto:attila.farkas@sztaki.hu)



Novel EOSC Services for  
Emerging Atmosphere,  
Underwater & Space  
Challenges

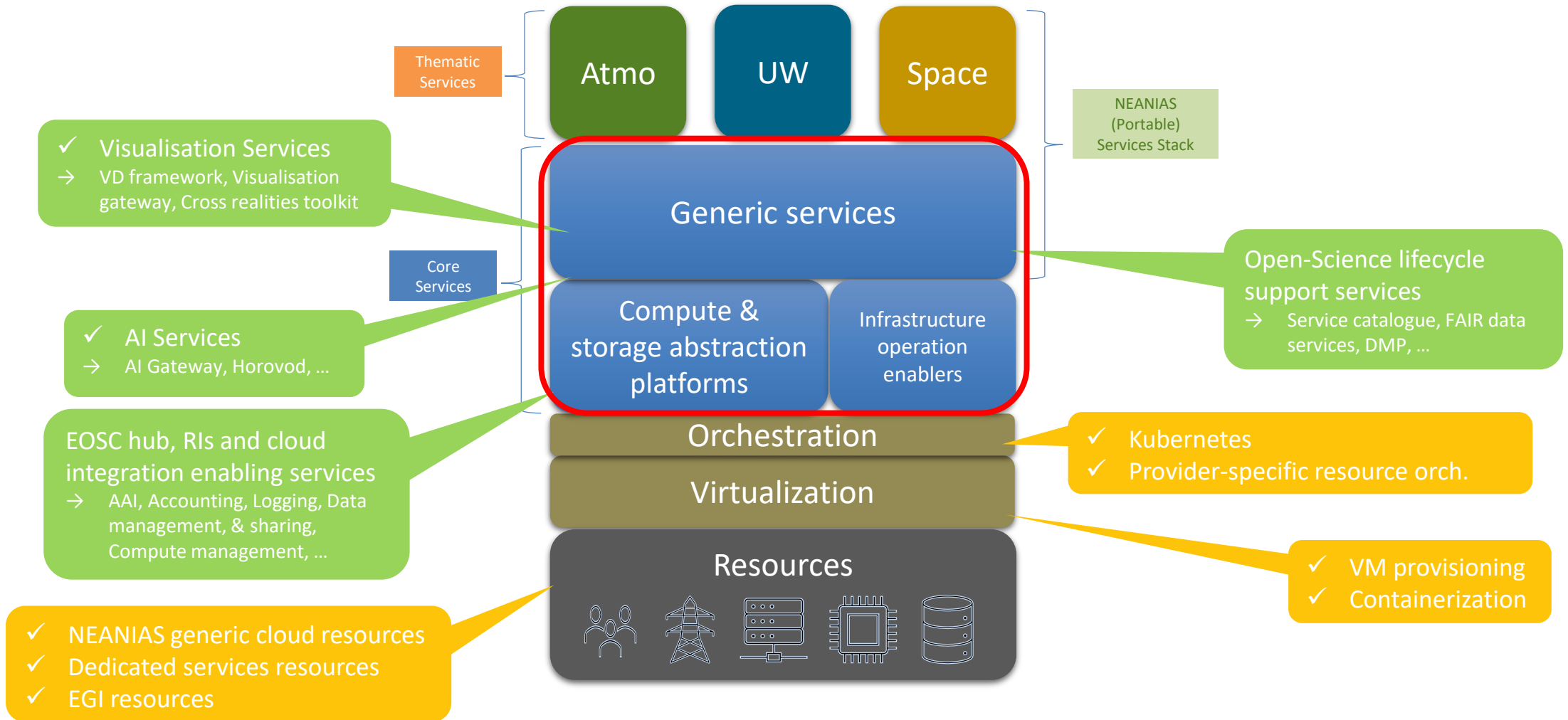
[www.NEANIAS.eu](http://www.NEANIAS.eu)

NEANIAS receives funding from  
European Union under Horizon 2020  
Research and Innovation  
Programme under grant agreement  
No. 863448





# The NEANIAS stack



# WP06 Core service groups

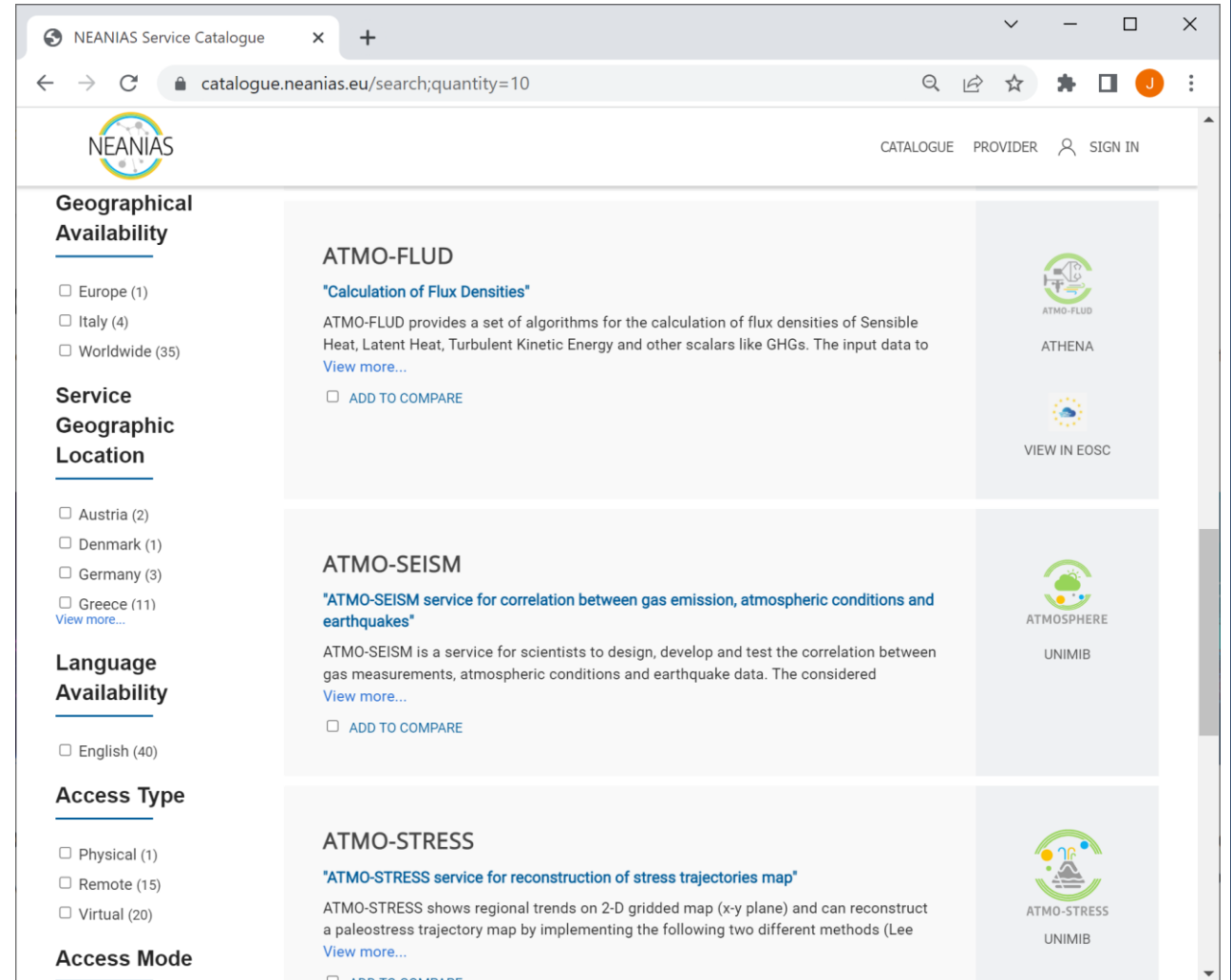
- C1 - Open-Science lifecycle support services (ATHENA)
  - Integration of the NEANIAS services into the EOSC ecosystem
- C2 - EOSC hub, RIs and cloud integration enabling services (CITE)
  - Lower-level services in the NEANIAS ecosystem for resource access
- C3 - AI services (SZTAKI)
  - Upper-level core services providing a typical machine learning workflow lifecycle
- C4 - Visualisation services (INAF)
  - Upper-level core services providing a typical visualization workflow lifecycle

## T6.3 – C1 services

- › C1.1 NEANIAS Catalogue Portal
- › C1.2 NEANIAS Catalogue Service
- › C1.3 NEANIAS Research Product Catalogue
- › C1.4 NEANIAS Data Validation Service
- › C1.5 NEANIAS web toolkit
- › C1.6 OpenDMP/ARGOS
- › C1.7 Data Publishing Service
- › C1.8 PID service for publications and research products

# C1.1 NEANIAS Catalogue Portal and Service (ATHENA)

- › Main entry point for all NEANIAS service offerings
- › Functionalities for
  - thematic service providers to manage their service portfolio
  - end users to search, compare, browse all service characteristics
- › <http://catalogue.neanias.eu/>
- › External REST APIs for the NEANIAS registry for:
  - register and update the service metadata in the NEANIAS catalogue
  - monitor the usage of their services.
  - 3rd party systems including EOOSC portal to synchronize the service metadata with the service catalogue of NEANIAS.
  - <http://catalogue.neanias.eu/developers>



The screenshot shows the NEANIAS Service Catalogue website. The browser address bar displays 'catalogue.neanias.eu/search;quantity=10'. The page features a search results grid with three service cards:

- ATMO-FLUD**: "Calculation of Flux Densities". Description: "ATMO-FLUD provides a set of algorithms for the calculation of flux densities of Sensible Heat, Latent Heat, Turbulent Kinetic Energy and other scalars like GHGs. The input data to View more...". Includes an "ADD TO COMPARE" button.
- ATMO-SEISM**: "ATMO-SEISM service for correlation between gas emission, atmospheric conditions and earthquakes". Description: "ATMO-SEISM is a service for scientists to design, develop and test the correlation between gas measurements, atmospheric conditions and earthquake data. The considered View more...". Includes an "ADD TO COMPARE" button.
- ATMO-STRESS**: "ATMO-STRESS service for reconstruction of stress trajectories map". Description: "ATMO-STRESS shows regional trends on 2-D gridded map (x-y plane) and can reconstruct a paleostress trajectory map by implementing the following two different methods (Lee View more...". Includes an "ADD TO COMPARE" button.

On the left side of the page, there are filter sections:

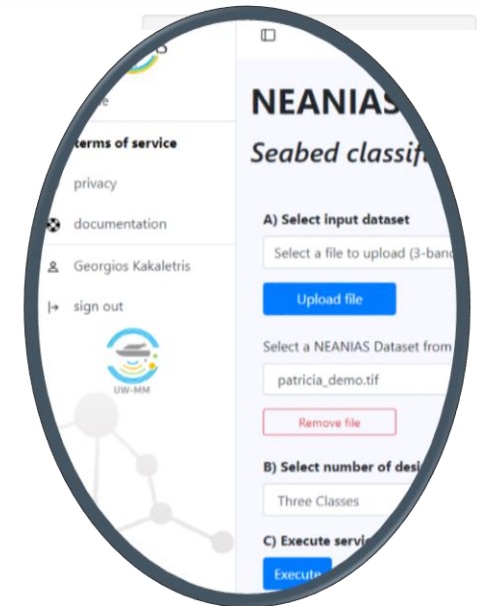
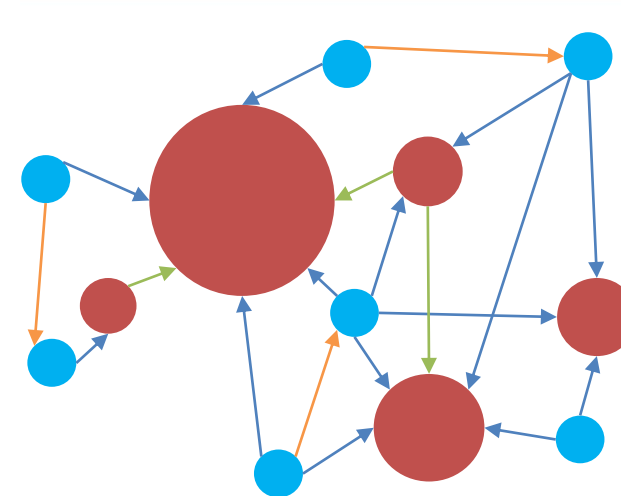
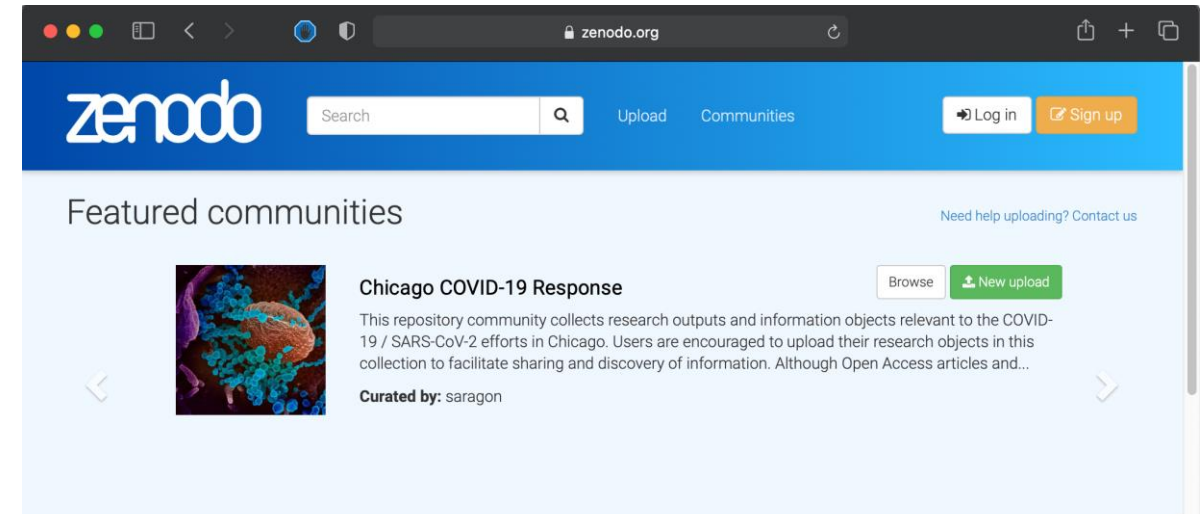
- Geographical Availability**: Europe (1), Italy (4), Worldwide (35)
- Service Geographic Location**: Austria (2), Denmark (1), Germany (3), Greece (11) [View more...]
- Language Availability**: English (40)
- Access Type**: Physical (1), Remote (15), Virtual (20)
- Access Mode**: (filters are partially visible)

On the right side, there are provider logos and links: ATMO-FLUD (ATHENA), ATMO-SEISM (ATMOSPHERE UNIMIB), and ATMO-STRESS (UNIMIB). Navigation links at the top include CATALOGUE, PROVIDER, and SIGN IN.

# NEANIAS C1 Services

- › C1.3 NEANIAS Research Product Catalogue (ARC/ATHENA)
  - Based on Zenodo
  - <https://zenodo.org>
- › C1.4 NEANIAS Data Validation Service (CITE)
  - Validate data from a user perspective from multiple standpoints:
    - › Fairness, compliance, validity, completeness etc
    - › Human driven with hooks for future automations.
- › C1.5 NEANIAS web toolkit (CITE/INAF)
  - Web templates and materials

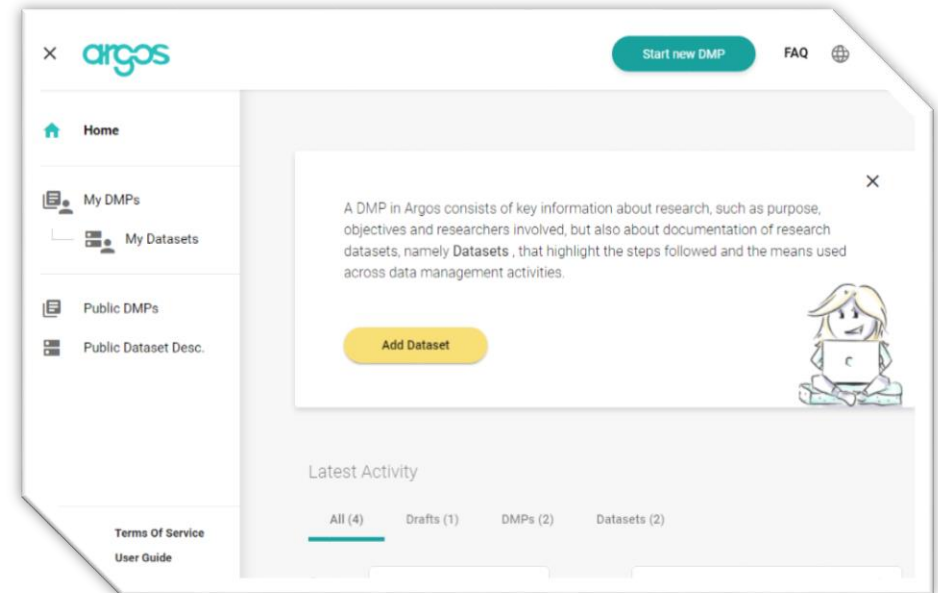
2022-09-23



NEANIAS OPEN EVENT – BARCELONA

# NEANIAS C1 services #2

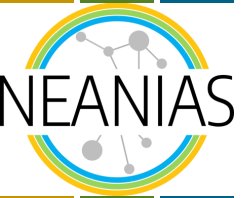
- › C1.6 OpenDMP/ARGOS (CITE)
  - Based on Argos
  - Data Management Planning and validation
  - <https://argos.openaire.eu/splash/>
- › Zenodo Bridge (CITE)
  - Simplified access to Zenodo
- › C1.7 Data Publishing Service (NKUA)
  - Based on Zenodo
- › C1.8 PID service for publications and research products (NKUA)
  - Based on Zenodo



## T6.4 – C2 services

- › C2.1 NEANIAS AAI
- › C2.2 NEANIAS Configuration Service
- › C2.3 NEANIAS Service Registry
- › C2.4 NEANIAS Logging
- › C2.5 NEANIAS Accounting
- › C2.6 NEANIAS Notification
- › C2.7 NEANIAS Object Storage
- › C2.8 NEANIAS Data Sharing Service
- › C2.9 Data Exploration Service
- › C2.10 SMTP email Service
- › C2.11 GARR Cloud Platform
- › C2.12 GARR Container Platform
- › C2.13 GARR DaaS






# C2.1 NEANIAS AAI (CITE)

- › Based on Keycloak
- › Horizontal AuthN solution for all NEANIAS services
- › Role based AuthZ provided
- › Environments:
  - Prod / Dev / Staging
- › Logging & Accounting integrations
- › Custom UI theme to facilitate login process across realms and providers
- › AuthZ & registration support
- › Access Granting process
  - User group / (composite) role restructuring
  - Assignment process
  - Align with SMS
- › <https://sso.neanias.eu/>

2022-09-23

Sign in with

 NEANIAS SSO

Remember me



NEANIAS

English v

Welcome to NEANIAS

In order to login you can use one of the supported Identity Providers from the list on the right

[Privacy Statement](#)

[Terms of Use](#)

Login with EGI Check-in

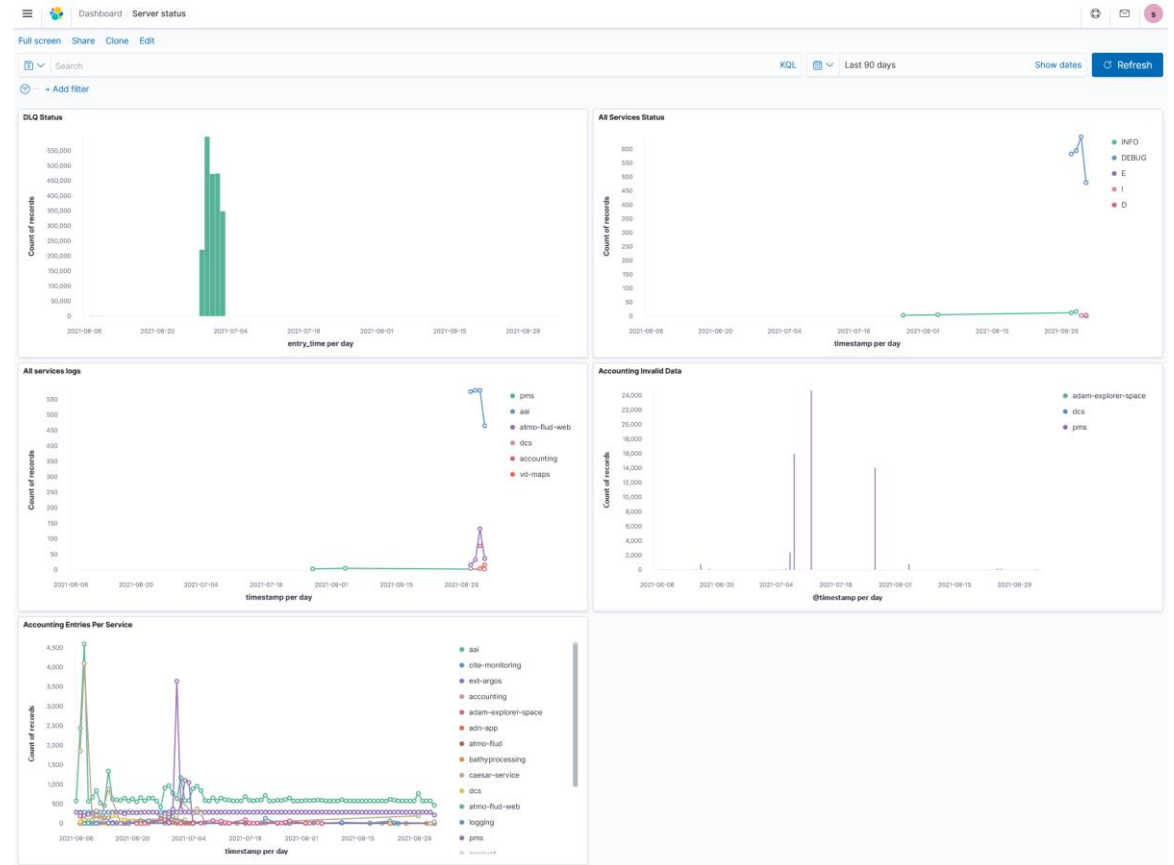
Login with NEANIAS

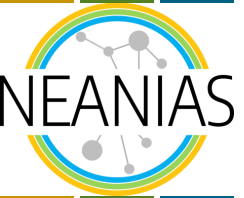
NEANIAS OPEN EVENT – BARCELONA



# C2.4 NEANIAS Logging (CITE)

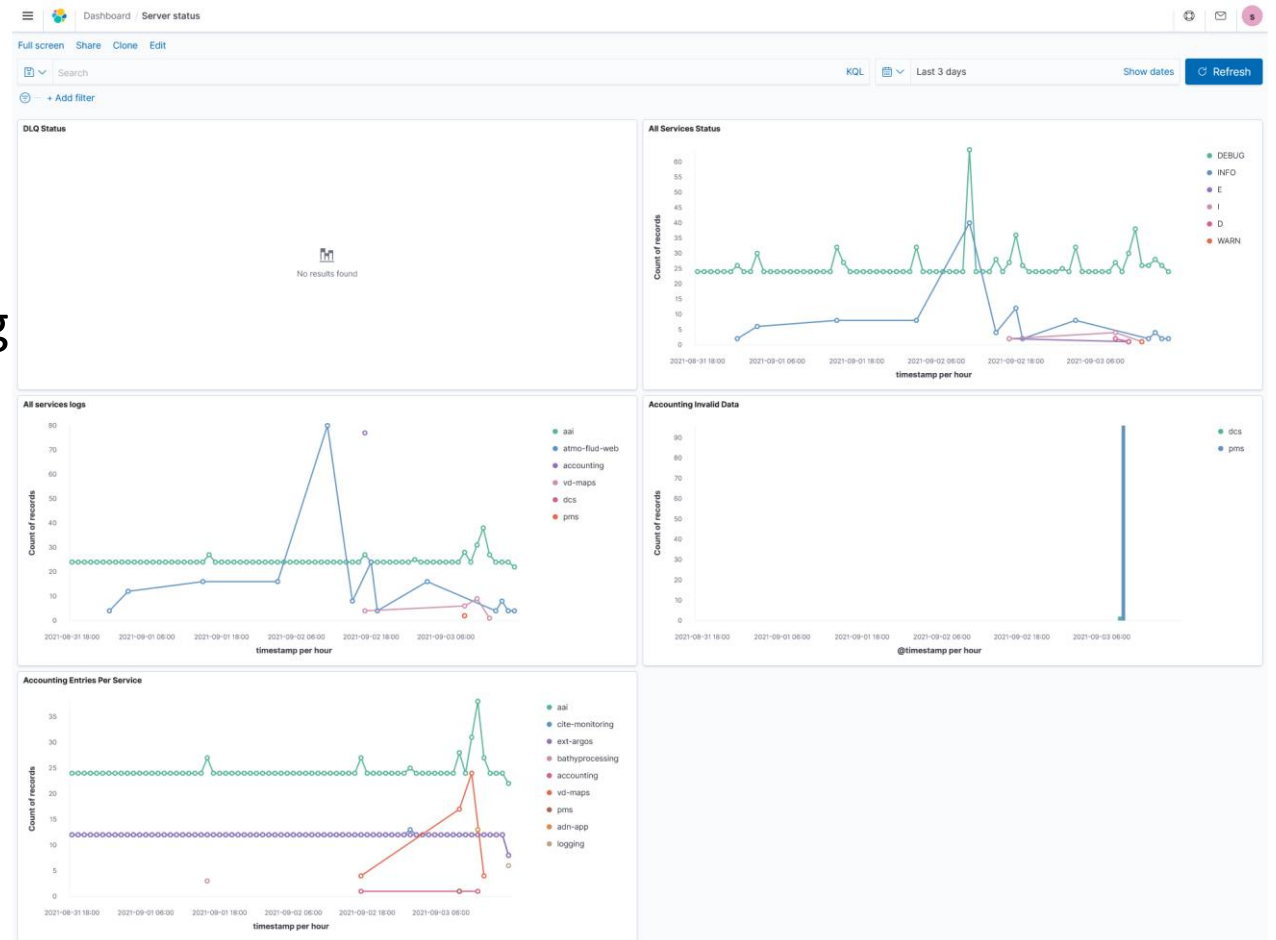
- › Based on ELK Stack
  - Elasticsearch
  - Logstash
  - Kibana
  - Beats
- › Horizontal Log Aggregator
  - Distributed, heterogeneous systems
  - Central Monitoring of Troubleshooting & Higher-level information
- › Versatile approach to allow configuration-based extensibility
  - Log Templates
  - Permissive Log Model
- › AuthN & AuthZ
  - NEANIAS AAI + Client level Identification
- › <https://logging.neanias.eu>





# C2.5 NEANIAS Accounting (CITE)

- › Centrally register accounting information
- › Gradually accumulate through service usage
- › Configuration and log-processing based integration
- › Authorized access and separation of data
- › Distributed and scalable across all layers
  - Data storage
  - Processing
  - Visualization
- › <https://accouting.neanias.eu>

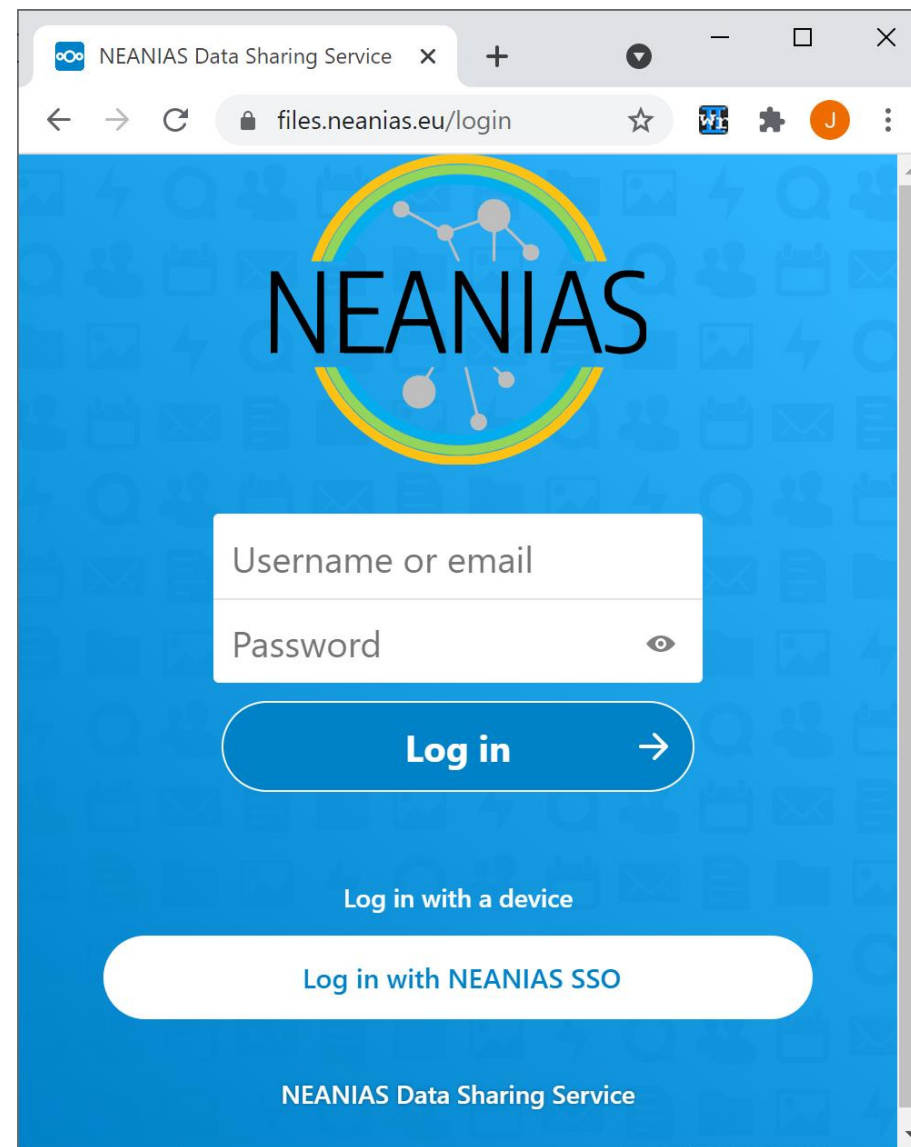


# NEANIAS C2 services

- › C2.2 NEANIAS Configuration Service (CITE)
  - Based on Zookeeper
  - Key Value store for storing NEANIAS service configurations
  - <https://configuration.neanias.eu>
- › C2.3 NEANIAS Service Registry (CITE)
  - Based on Zookeeper
  - Provides service discovery and health status
  - <https://registry.neanias.eu>
- › C2.6 NEANIAS Notification (CITE)
  - HTTP API
  - Notification channel: email
  - Template and Ad-hoc based notifications
  - <https://notification.neanias.eu>

# C2.8 NEANIAS Data Sharing Service (NKUA)

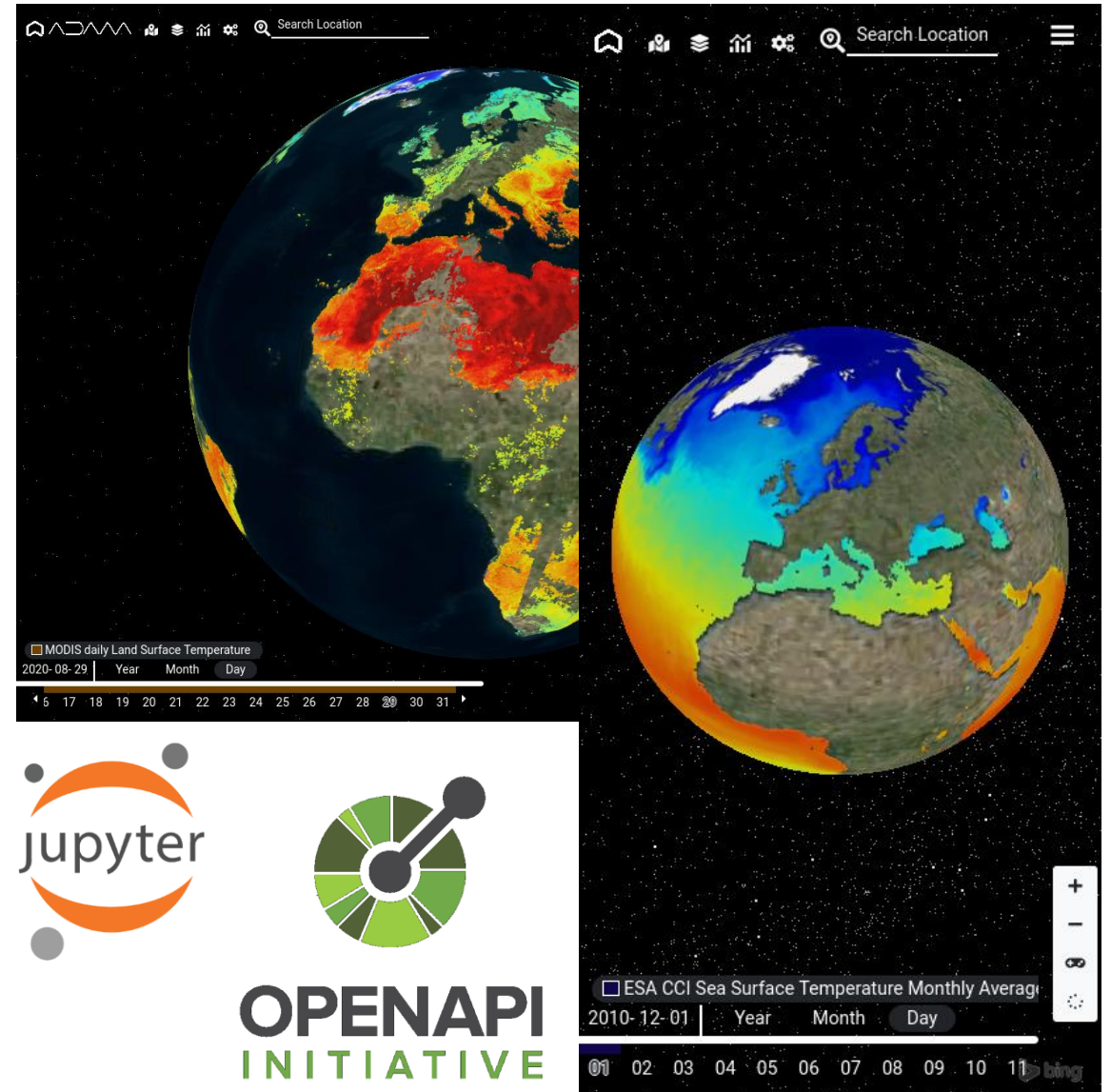
- › Based on Nextcloud
- › Allows end-users to share files with the NEANIAS services
- › Allows inter and intra service communication via file sharing.
- › Quota management
- › Utilized by Thematic and Core services
- › <https://files.neanias.eu>



# C2.9 Data Exploration Service (MEE0)

- › Based on adamapi (2.0.8)
- › Supports the visual browsing and searching of Planetary Data
- › Continuous registration of new datasets and products
- › Integration with C4.4
- › Onboarded on EOSC
- › <https://explorer.adamplatform.eu>

2022-09-23



NEANIAS OPEN EVENT – BARCELONA

# GARR Cloud and Container platform (GARR)

## › C2.11 GARR Cloud platform

- Based on OpenStack
- In production on three geographical regions



## › C2.12 GARR Container platform

- Based on Kubernetes
- Several instances deployed: NEANIAS production cluster, NEANIAS staging cluster



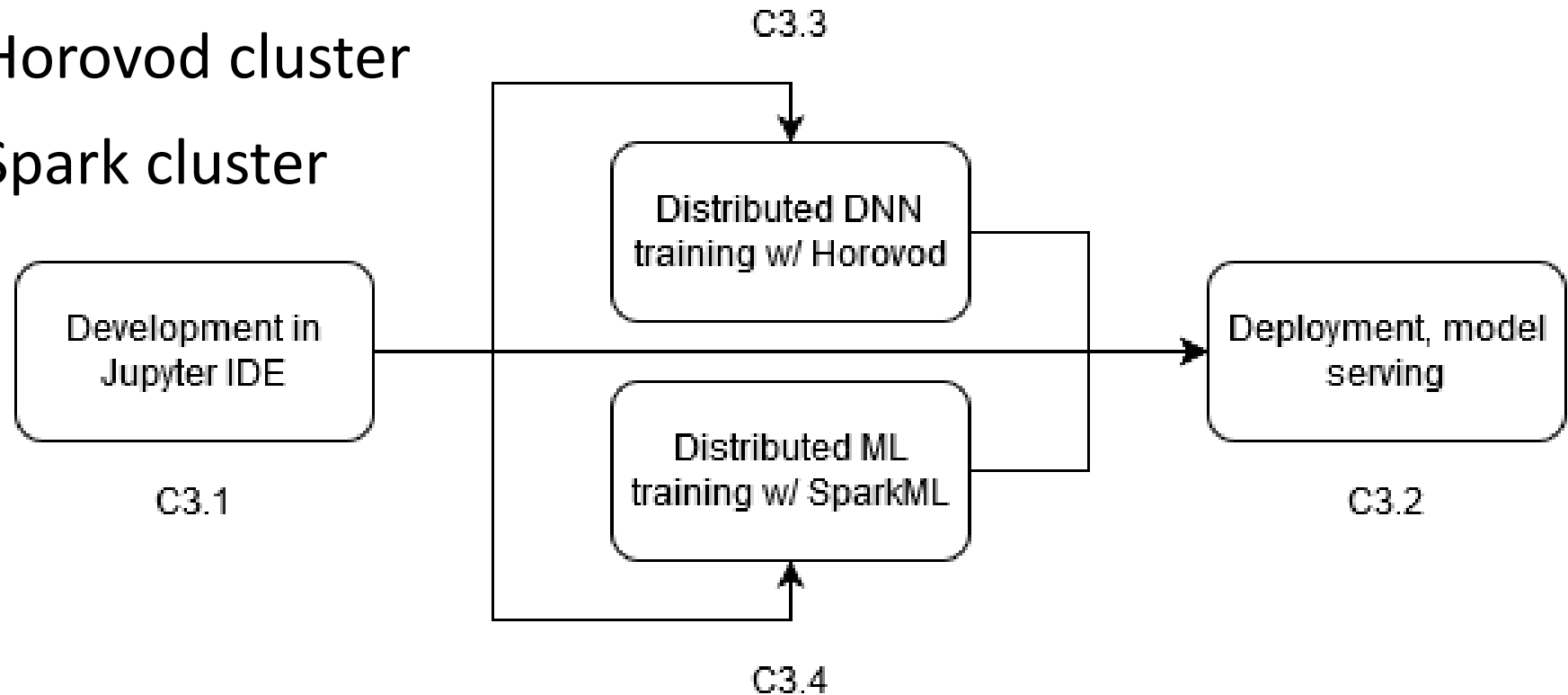
## › C2.13 GARR DaaS

- Based on Juju
- In production in three regions
- Used to deploy and manage GARR Container Platform (Kubernetes) instances



# T6.5 – C3 services

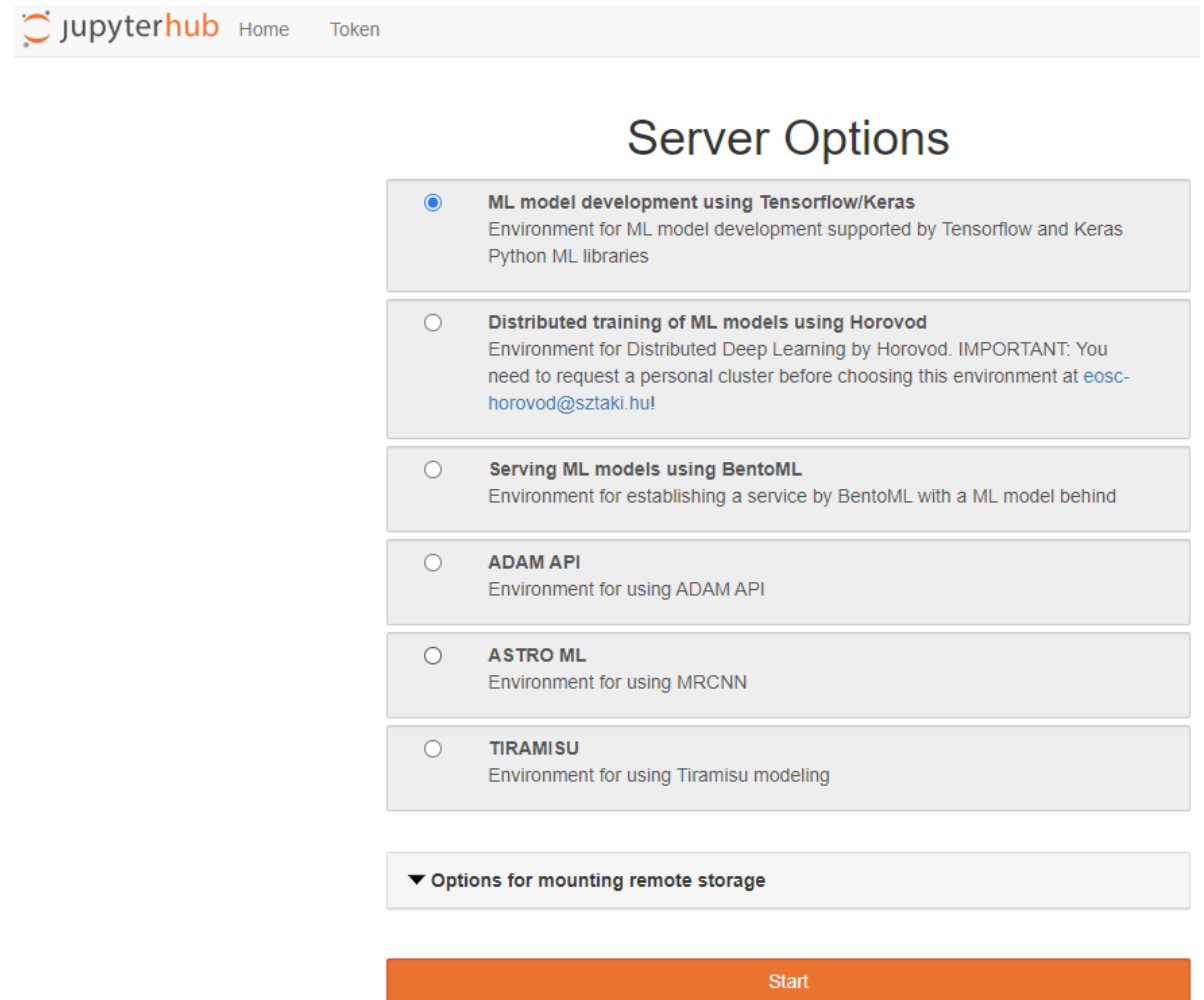
- › C3.1 AI Gateway
- › C3.2 Model serving
- › C3.3 Horovod cluster
- › C3.4 Spark cluster





# C3.1 AI Gateway (SZTAKI)

- › Based on JupyterHUB
- › Supports machine learning related environments
- › Production environments:
  - Tensorflow/Keras
  - Distributed Training with Horovod
  - Model serving with BentoML
  - Adam API
  - ASTROML
  - TIRAMISU
- › Integrated
  - AAI
  - Logging and Accounting
  - Data Sharing Service
- › Deployed on the NEANIAS Kubernetes cluster
- › <https://ai-gateway.neanias.eu>



jupyterhub Home Token

## Server Options

- ML model development using Tensorflow/Keras**  
Environment for ML model development supported by Tensorflow and Keras Python ML libraries
- Distributed training of ML models using Horovod**  
Environment for Distributed Deep Learning by Horovod. IMPORTANT: You need to request a personal cluster before choosing this environment at [eosc-horovod@sztaki.hu](mailto:eosc-horovod@sztaki.hu)
- Serving ML models using BentoML**  
Environment for establishing a service by BentoML with a ML model behind
- ADAM API**  
Environment for using ADAM API
- ASTRO ML**  
Environment for using MRCNN
- TIRAMISU**  
Environment for using Tiramisu modeling

▼ Options for mounting remote storage

Start

# C3.2 Model serving

- › Based on BentoML, supports model serving
- › Support exposing the trained ML models as services
- › Simplifies machine-learning model deployment
- › Runs high-performance model serving at scale;
- › Deployed on NEANIAS Kubernetes cluster
- › Integrated with C3.1 AI-Gateway

### Creating a Prediction Service with BentoML

A minimal prediction service in BentoML looks something like this:

```
In [91]: %writefile iris_classifier.py
from bentoml import env, artifacts, api, BentoService
from bentoml.adapters import DataframeInput
from bentoml.artifact import SklearnModelArtifact

@env(auto_pip_dependencies=True)
@artifacts([SklearnModelArtifact('model')])
class IrisClassifier(BentoService):

    @api(input=DataframeInput(), batch=True)
    def predict(self, df):
        # Optional pre-processing, post-processing code goes here
        return self.artifacts.model.predict(df)
```

Overwriting iris\_classifier.py

This code defines a prediction service that bundles a scikit-learn model and provides an API that expects input data in the form of pandas.DataFrame. The user-defined API function predict defines how the input dataframe data will be processed and used for inference with the bundled scikit-learn model. BentoML also supports other API input types such as ImageInput, JsonInput and more.

The following code trains a scikit-learn model and packages the trained model with the IrisClassifier class defined above. It then saves the IrisClassifier instance to disk in the BentoML SavedBundle format:

```
In [92]: from sklearn import svm
from sklearn import datasets

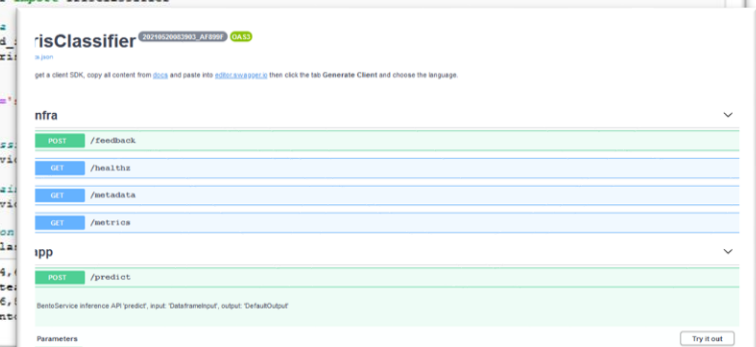
# import the custom BentoService defined above
from iris_classifier import IrisClassifier

# Load training data
iris = datasets.load_iris()
X, y = iris.data, iris.target

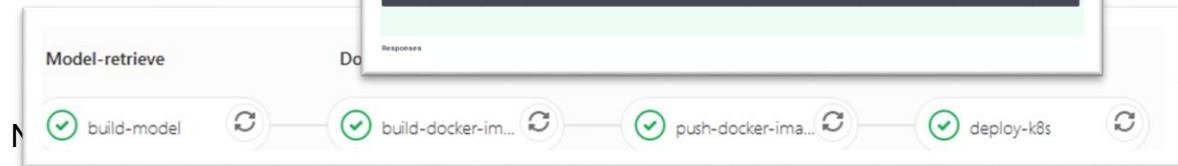
# Model Training
clf = svm.SVC(gamma=0.001)
clf.fit(X, y)

# Create a iris classifier BentoService
iris_classifier_service = IrisClassifier(
    artifacts=[SklearnModelArtifact('model', clf)]
)

# Pack the newly trained model into a BentoML SavedBundle
saved_path = iris_classifier_service.save_to_disk('iris_classifier')
```

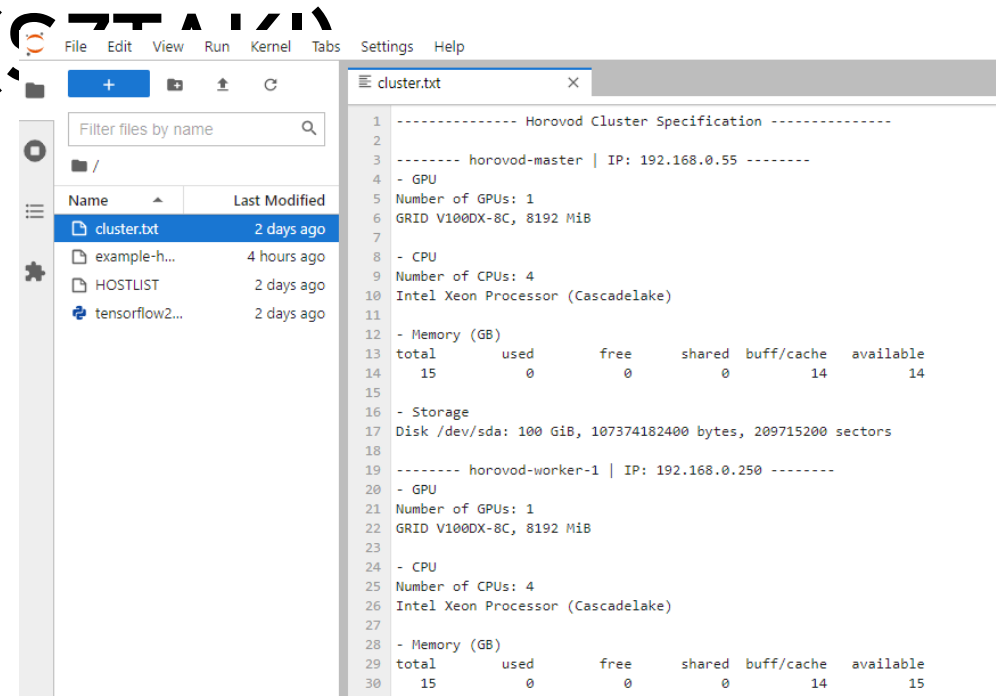


The screenshot shows the BentoML API interface for the 'IrisClassifier' service. It displays a list of endpoints: POST /feedback, GET /healthz, GET /metadata, GET /metrics, and POST /predict. The /predict endpoint is selected, showing its parameters (none), request body (application/json), and an example request body: {}.



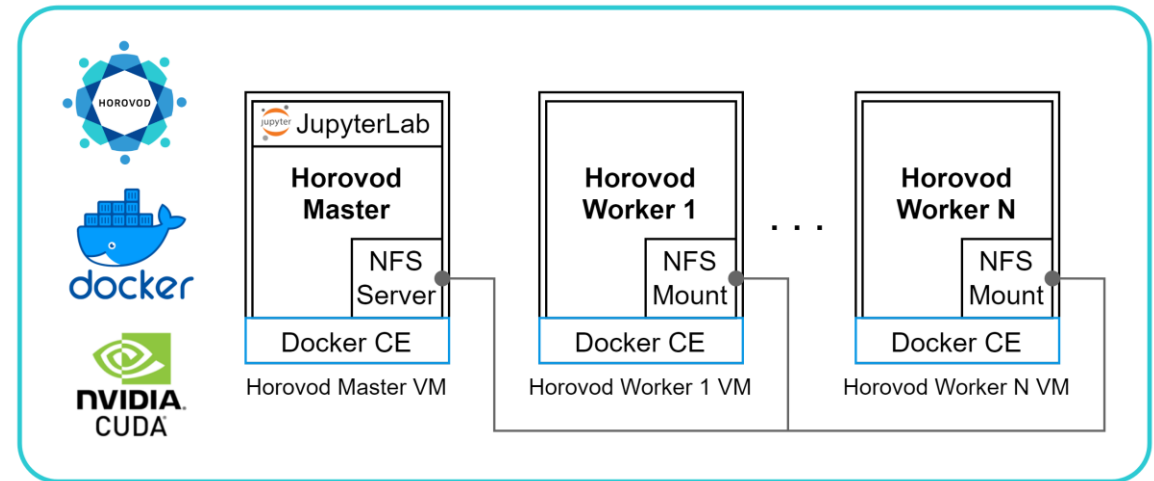
# C3.3 Horovod cluster

- › Distributed Deep Learning based on Horovod
- › Accessible through JupyterLab
- › Shared storage inside the cluster
- › Demo examples for distributed training
- › Automatic deployment method based on Terraform and Horovod for EGI-ACE
- › CI/CD based automatic testing
- › Internal monitoring system based on Prometheus and Grafana
- › Integrated with C3.1 AI-Gateway
- › Onboarded on EOSC



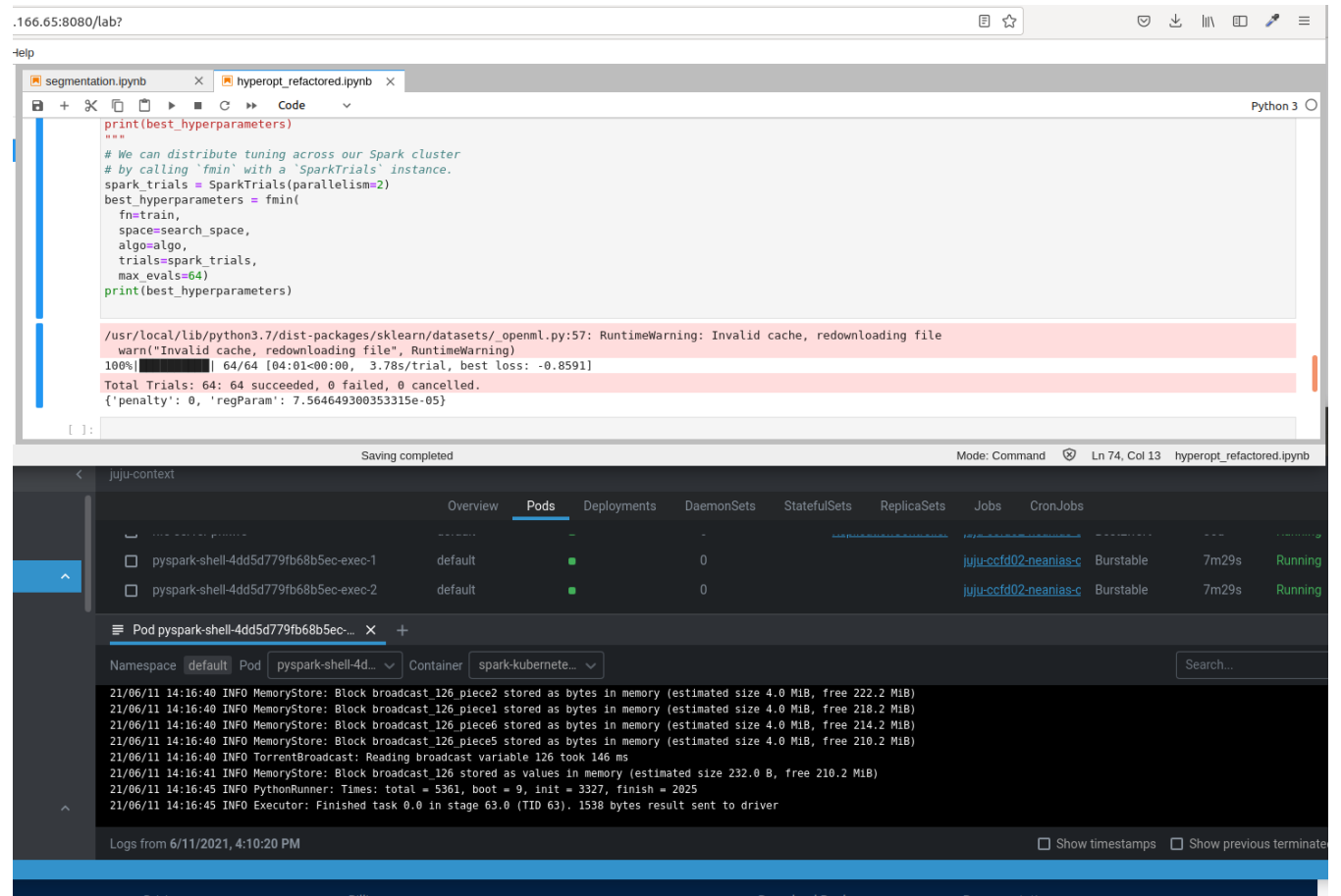
```

1 ----- Horovod Cluster Specification -----
2
3 ----- horovod-master | IP: 192.168.0.55 -----
4 - GPU
5 Number of GPUs: 1
6 GRID V100DX-8C, 8192 MiB
7
8 - CPU
9 Number of CPUs: 4
10 Intel Xeon Processor (Cascadelake)
11
12 - Memory (GB)
13 total    used      free      shared  buff/cache  available
14 15         0         0         0        14         14
15
16 - Storage
17 Disk /dev/sda: 100 GiB, 107374182400 bytes, 209715200 sectors
18
19 ----- horovod-worker-1 | IP: 192.168.0.250 -----
20 - GPU
21 Number of GPUs: 1
22 GRID V100DX-8C, 8192 MiB
23
24 - CPU
25 Number of CPUs: 4
26 Intel Xeon Processor (Cascadelake)
27
28 - Memory (GB)
29 total    used      free      shared  buff/cache  available
30 15         0         0         0        14         15
  
```



# C3.4 Spark cluster (UNIMIB)

- › Distributed ML on Kubernetes
- › Example use cases implemented
- › Split Spark cluster in multiple namespaces to manage resources
- › Jobs could be spawned remotely with right k8s config file
- › Tested successfully on EGI ACE resources



The screenshot displays a JupyterLab environment. The top part shows a code editor with a Python script for hyperparameter tuning using SparkTrials. The script includes comments and code for setting up SparkTrials with parallelism=2 and using fmin for optimization. The output shows a successful run with 64 trials, a best loss of -0.8591, and a penalty of 0.

```

print(best_hyperparameters)
"""
# We can distribute tuning across our Spark cluster
# by calling 'fmin' with a 'SparkTrials' instance.
spark_trials = SparkTrials(parallelism=2)
best_hyperparameters = fmin(
    fn=train,
    space=search_space,
    algo=algo,
    trials=spark_trials,
    max_evals=64)
print(best_hyperparameters)

/usr/local/lib/python3.7/dist-packages/sklearn/datasets/_openml.py:57: RuntimeWarning: Invalid cache, redownloading file
warn("Invalid cache, redownloading file", RuntimeWarning)
100%|██████████| 64/64 [04:01<00:00, 3.78s/trial, best loss: -0.8591]
Total Trials: 64: 64 succeeded, 0 failed, 0 cancelled.
{'penalty': 0, 'regParam': 7.564649300353315e-05}

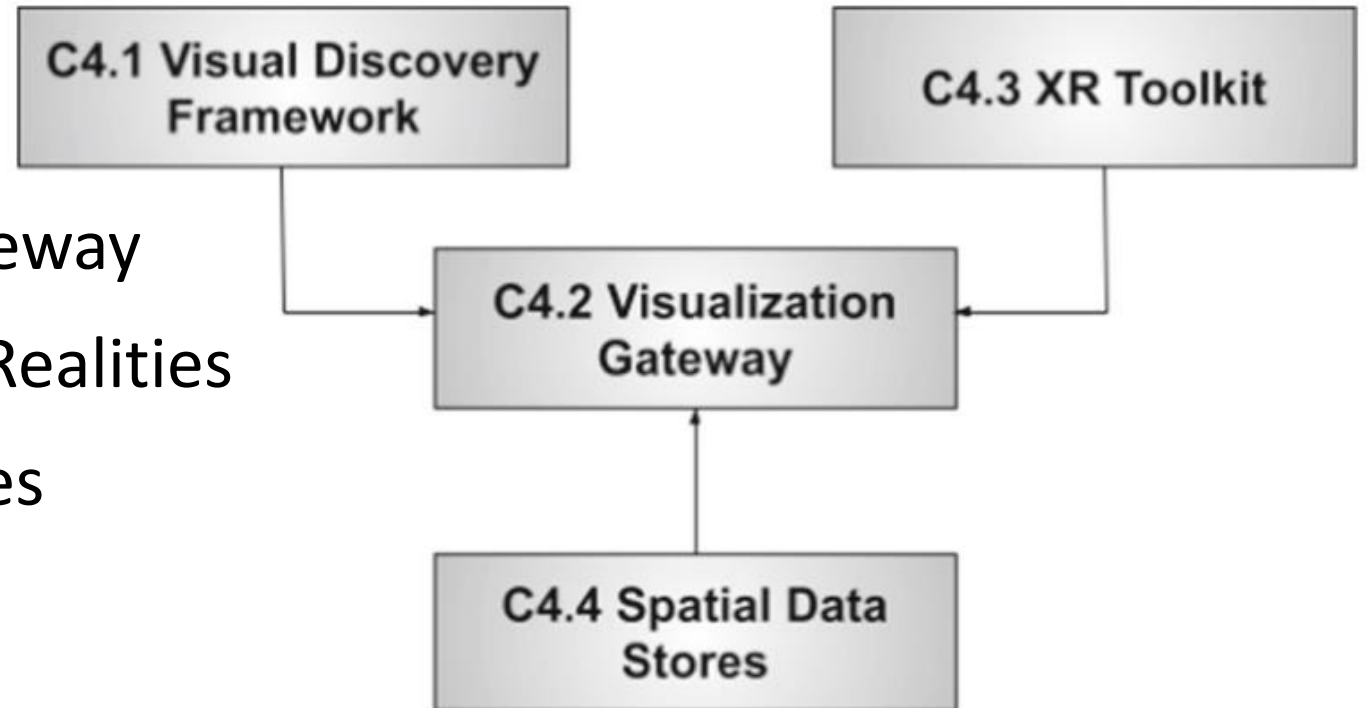
```

The bottom part of the screenshot shows the Kubernetes dashboard for the 'juju-context' namespace. It displays a table of pods, with two pods running: 'pyspark-shell-4dd5d779fb68b5ec-exec-1' and 'pyspark-shell-4dd5d779fb68b5ec-exec-2'. The logs for the selected pod show the Spark execution progress, including memory store operations and the completion of a task.

Pod Name	Namespace	Status	Age	Ready
pyspark-shell-4dd5d779fb68b5ec-exec-1	default	Running	7m29s	1/1
pyspark-shell-4dd5d779fb68b5ec-exec-2	default	Running	7m29s	1/1

# T6.6 – C4 services

- › C4.1 Visual Discovery Framework
  - C4.1.1 VD-Visivo
  - C4.1.2 VD-Splotch
  - C4.1.3 VD-Maps
- › C4.2 Visualisation Gateway
- › C4.3 Toolkit for Cross Realities
- › C4.4 Spatial Data Stores



# C4.1 Visual Discovery Framework (INAF+UoP+CORONIS)

## › VD-VisIVO

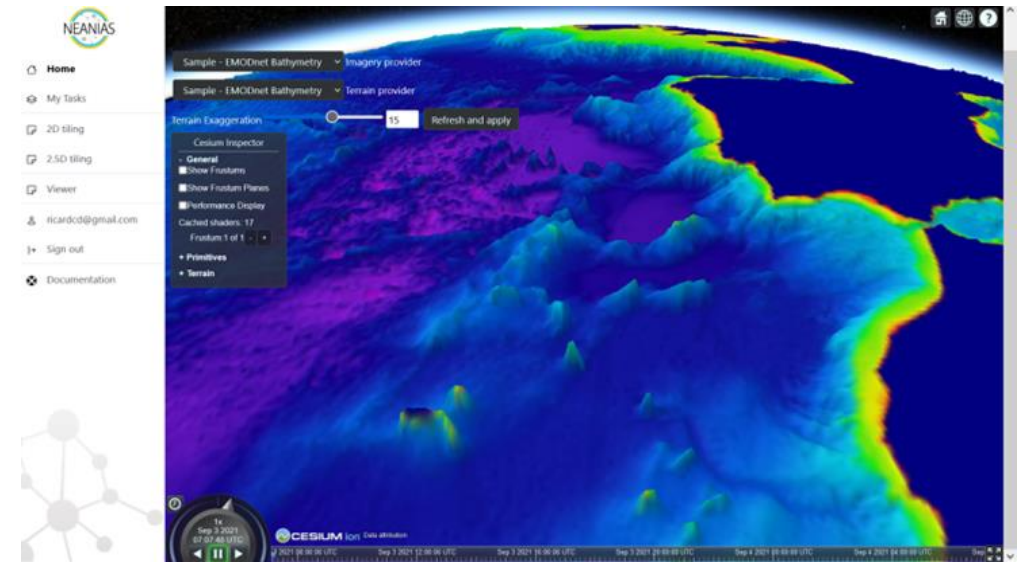
- framework for data intensive visual discovery for experiments and data analysis
- data processing and visual discovery the suite of tools provided by VisIVO
- available using the Visualization Gateway

## › VD-Splotch

- data processing and visual discovery the suite of tools provided by Splotch
- available using the Visualization Gateway

## › VD-MAPS

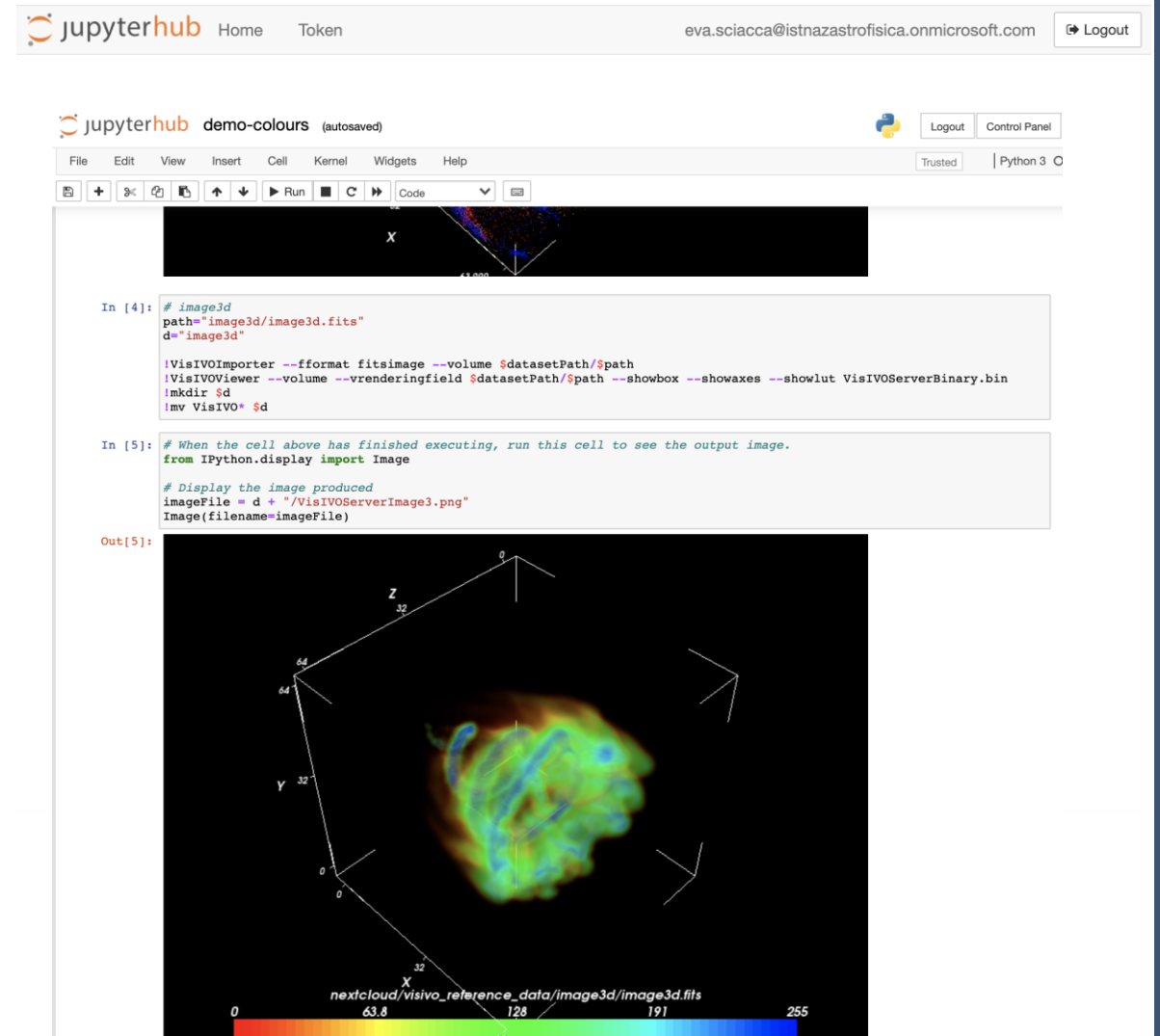
- create, serve and visualize imagery (2D images) or terrain-based (elevation and/or bathymetric maps) hierarchical tiled maps
- Onboarded on EOSC



VD-Maps

# C4.2 Visualization Gateway (INAF+UoP)

- › Fully based on JupyterHub framework with
  - C4.1 VD-VisIVO
  - VD-Splotch
- › Running on the NEANIAS Kubernetes Cluster
- › HPC Testing on Services on the HPE HPC
- › Integrated
  - AAI
  - Logging and Accounting
  - Data Sharing Service
- › <https://vis-gateway.NEANIAS.eu>



The screenshot shows a JupyterHub interface with a notebook titled "demo-colours (autosaved)". The notebook contains two code cells:

```
In [4]: # image3d
path="image3d/image3d.fits"
d="image3d"

!VisIVOImporter --format fitsimage --volume $datasetPath/$path
!VisIVOViewer --volume --vrenderingfield $datasetPath/$path --showbox --showaxes --showlut VisIVOServerBinary.bin
!mkdir $d
!mv VisIVO* $d
```

```
In [5]: # When the cell above has finished executing, run this cell to see the output image.
from IPython.display import Image

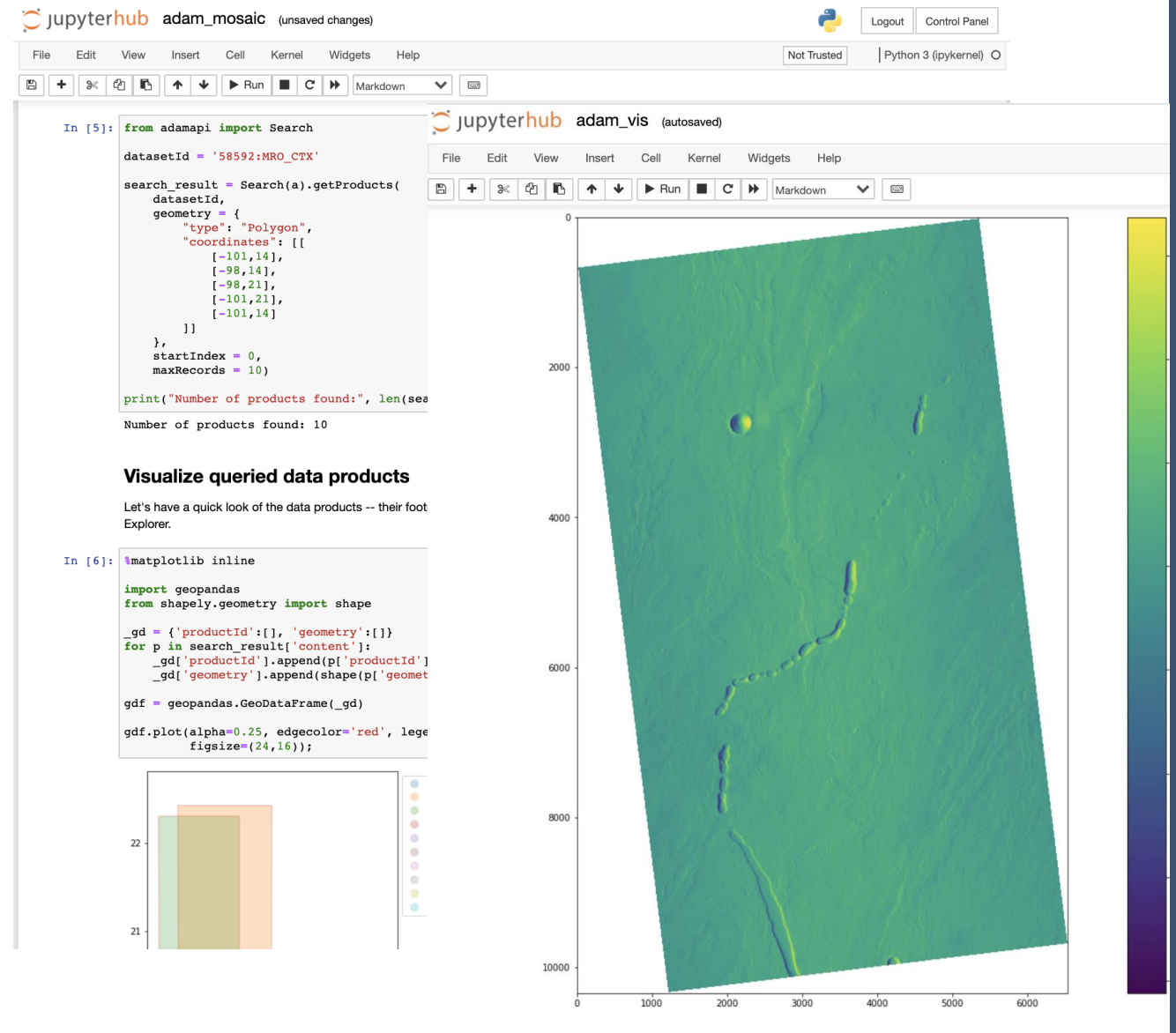
# Display the image produced
imageFile = d + "/VisIVOServerImage3.png"
Image(filename=imageFile)
```

The output of the second cell shows a 3D visualization of a data volume, rendered as a colorful, textured object within a 3D coordinate system. The axes are labeled X, Y, and Z. A color bar at the bottom indicates the scale of the data, ranging from 0 to 255.



# NEANIAS C4 services

- › C4.3 Toolkit for Cross Realities (ALTEC)
  - Environment complex visualisation solutions exposed to end-users via Cross Realities (XR) mechanisms like VR, AR
  
- › C4.4 Spatial Data Stores (JACOBS+MEEEO)
  - Spatial data stores allow data to be referenced, query and retrieved, based on a given location
  - Accessible through ADAM interface



The screenshot displays a JupyterLab environment with two notebooks. The top notebook, titled 'adam\_mosaic', contains Python code that uses the 'adamapi' library to search for products based on a dataset ID. The code defines a polygon geometry and queries the system, returning 10 products. The bottom notebook, titled 'adam\_vis', uses 'matplotlib' and 'geopandas' to visualize the queried data products. It plots the products as semi-transparent polygons on a 3D topographic map. A color scale on the right indicates elevation, ranging from 0 to 10000. Below the main map, a smaller plot shows the bounding boxes of the queried products.

```

In [5]: from adamapi import Search

datasetId = '58592:MRO_CTX'

search_result = Search(a).getProducts(
    datasetId,
    geometry = {
        "type": "Polygon",
        "coordinates": [[
            [-101,14],
            [-98,14],
            [-98,21],
            [-101,21],
            [-101,14]
        ]]
    },
    startIndex = 0,
    maxRecords = 10)

print("Number of products found:", len(search_result))
Number of products found: 10

Visualize queried data products
Let's have a quick look of the data products -- their foot
Explorer.

In [6]: %matplotlib inline

import geopandas
from shapely.geometry import shape

_gd = {'productId': [], 'geometry': []}
for p in search_result['content']:
    _gd['productId'].append(p['productId'])
    _gd['geometry'].append(shape(p['geometry']))

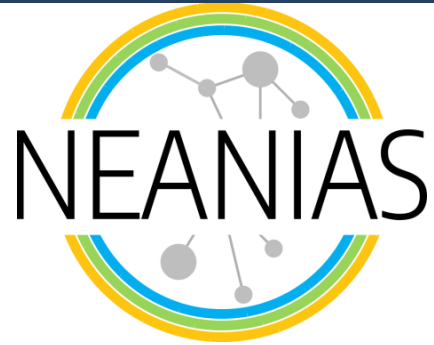
gdf = geopandas.GeoDataFrame(_gd)

gdf.plot(alpha=0.25, edgecolor='red', legend=True,
         figsize=(24,16));
  
```



# Summary

- › Almost **30 core services** are being developed and maintained by **12 partners**
- › **3<sup>rd</sup> release** has been issued in Feb 2022
- › Focus shifted towards operation and maintenance (WP7)
- › Next presentations & demos:
  - NEANIAS Accounting - Giorgos Papanikos
  - Horovod with AI-Gateway - Krisztián Póra
  - VD-MAPS - Ricard Campos



## Novel EO SC Services for Emerging Atmosphere, Underwater & Space Challenges



NEANIAS receives funding from European Union under Horizon 2020 Research and Innovation Programme under grant agreement No. 863448

## Thank you for the attention!

Attila Farkas (SZTAKI) – [attila.farkas@sztaki.hu](mailto:attila.farkas@sztaki.hu)

- Jozsef Kovacs WP6 leader (SZTAKI)
- George Papastefanatos, T6.1, T6.3 leader (ATHENA)
- Giorgos Papanikos, T6.2, T6.4 leader (CITE)
- Gabor Kertesz, T6.5 leader (SZTAKI)
- Eva Sciacca, T6.6 leader (INAF)
- WP6 partners: NKUA, ATHENA, INAF, SZTAKI, CORONIS, CITE, UOP, UNIMIB, JACOBSUNI, MEEO, GARR, ALTEC

### Follow us:

<http://www.NEANIAS.eu>

[https://twitter.com/NEANIAS\\_eu](https://twitter.com/NEANIAS_eu)

<https://www.facebook.com/NEANIAS.eu/>

<https://www.linkedin.com/groups/13786081/>

2022-09-23

NEANIAS OPEN EVENT – BARCELONA