

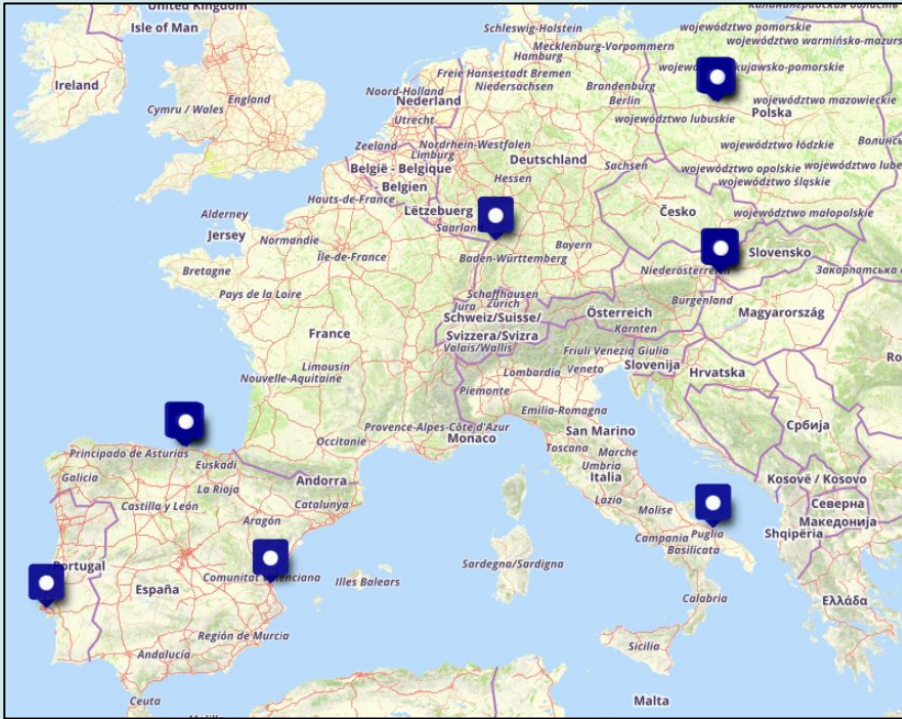
AI4 |  eOSC

Empowering #EOSC scientists with AI services and tools

Álvaro López García (aloga@ifca.unican.es)



Co-funded by
the European Union



AI4



Co-funded by the European Union

AI4EOSC

Artificial Intelligence for the #EOSC

- Evolution of the DEEP Hybrid DataCloud platform
- HORIZON-INFRA-2021-EOSC-01-04 call
- Runs September 1st 2022 – August 2025 (36 months)
- 7 academic + 2 SME + 1 non-profit organization

Advanced features for distributed, federated, composite learning, metadata provenance, MLOps, event-driven data processing, and provision of AI/ML/DL services



Objectives

Objective 1

Feature rich services and platform to build and deploy custom AI applications in the EOOSC

Objective 2

Support for building AI systems on distributed datasets, with a particular focus on federated learning

Objective 3

Services to compose AI tools, enabling the development of complex data-driven applications

Objective 4

AI exchange/hub in the context of the EOOSC, enhancing and increasing the application offer currently available

Objective 5

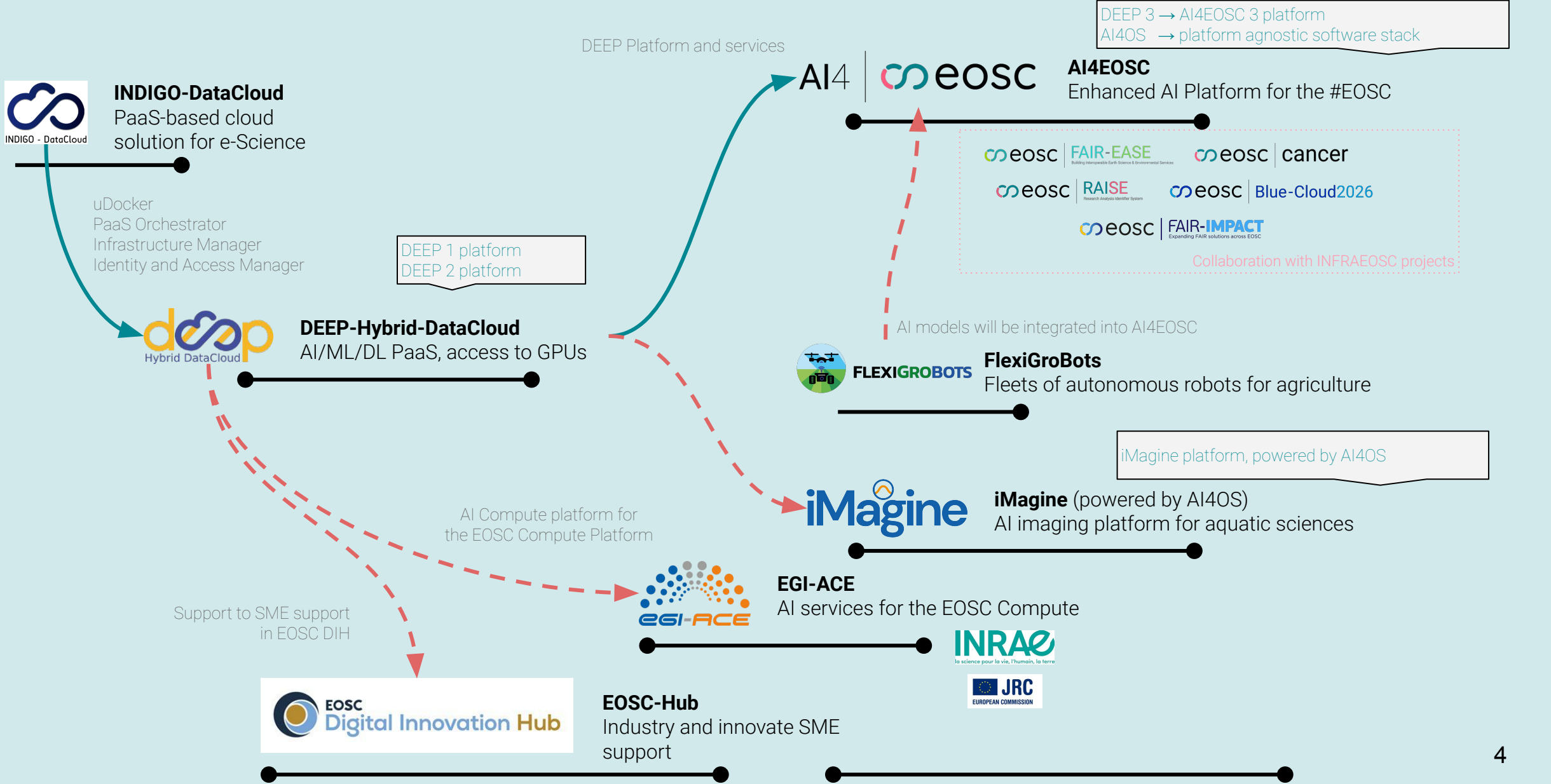
Extend the service offer and the capabilities being offered through the EOOSC portal, with focus on AI

Goal

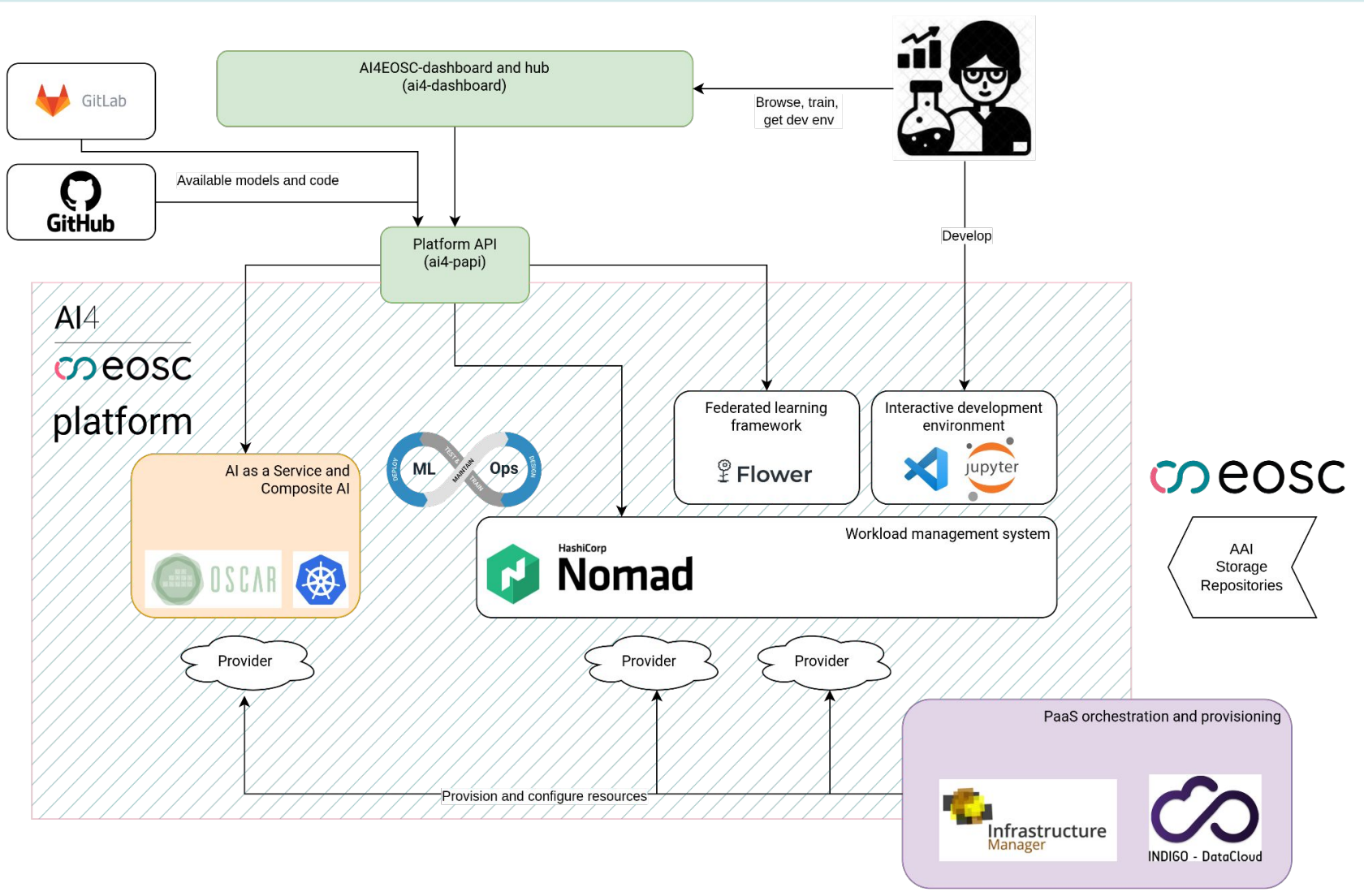
Foster an AI exchange in the EOOSC context transforming the development of AI applications in the EOOSC

Background, ecosystem, collaborations

2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	...
------	------	------	------	------	------	------	------	------	------	------	-----



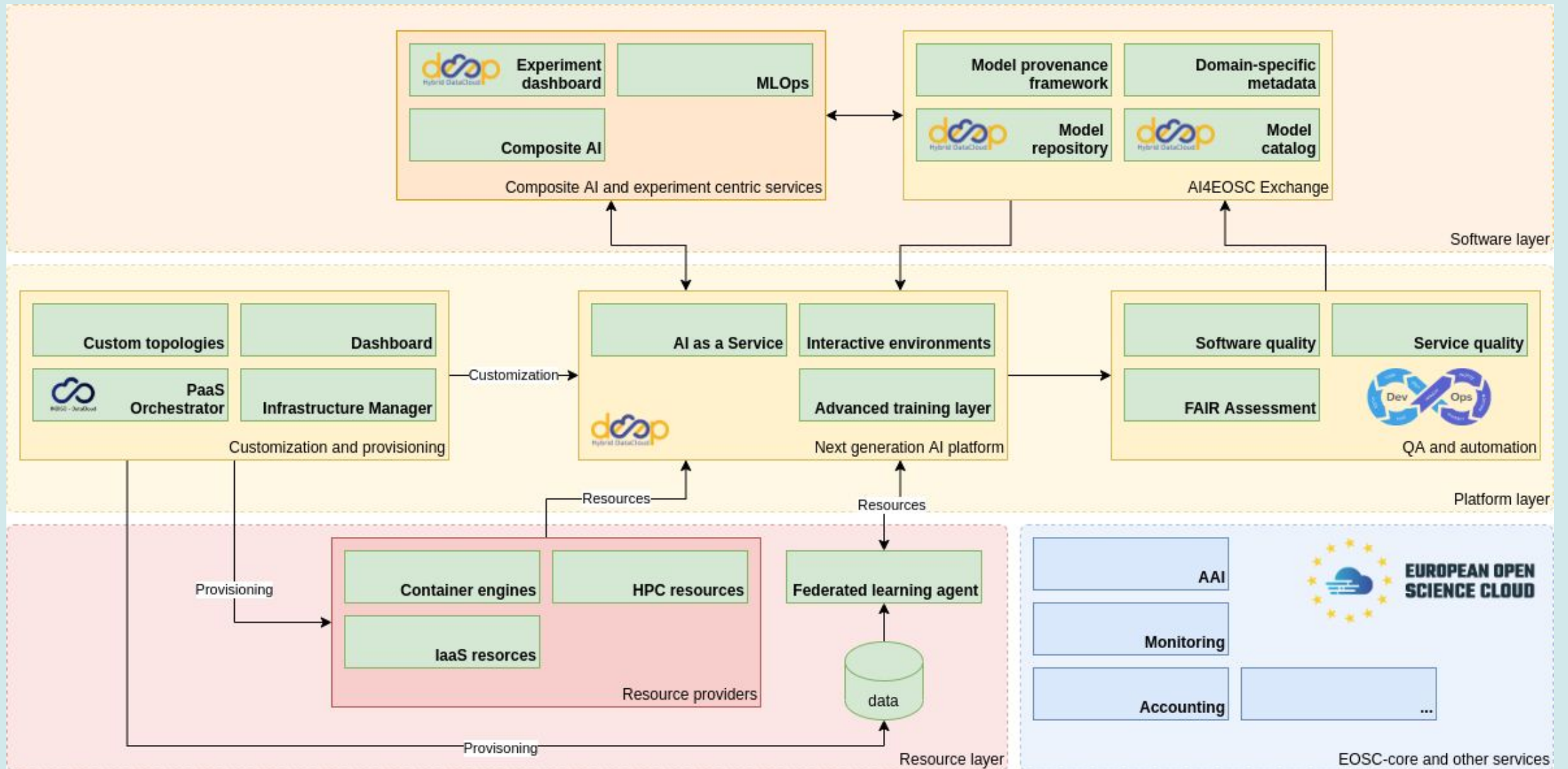
AI4EOSC high level architecture



Detailed C4 architecture can be found here:

- Workspace
<https://structurizr.com/share/73873/2f769b91-f208-41b0-b79f-5e196435bdb1>
- Diagrams:
<https://structurizr.com/share/73873/2f769b91-f208-41b0-b79f-5e196435bdb1/images>

AI4EOSC conceptual diagram



DEEP evolves into... AI4EOSC

Training on single site,
centralized dataset expected

Single AI application, self
deployed or on serverless
computing

Central management of
onboarded sites, complex
on-premises deployment

Federated learning, split
learning, gossip learning,
making possible training on
decentralized datasets

Composite AI for complex AI
tools and applications through
function composition and
serverless computing

Enhanced onboarding of
resources, easier deployment
on-premises

(some) New features

Integration with privacy tools
(differential privacy, anonymity
checks)

ML pipeline composition and
workflows

MLOps tools to monitor
deployed models (drift
detection, concept drift,
accuracy and performance)

Community standards for
models API (Kserve) following
OpenAPI specifications

Enhanced web user interface
for applications

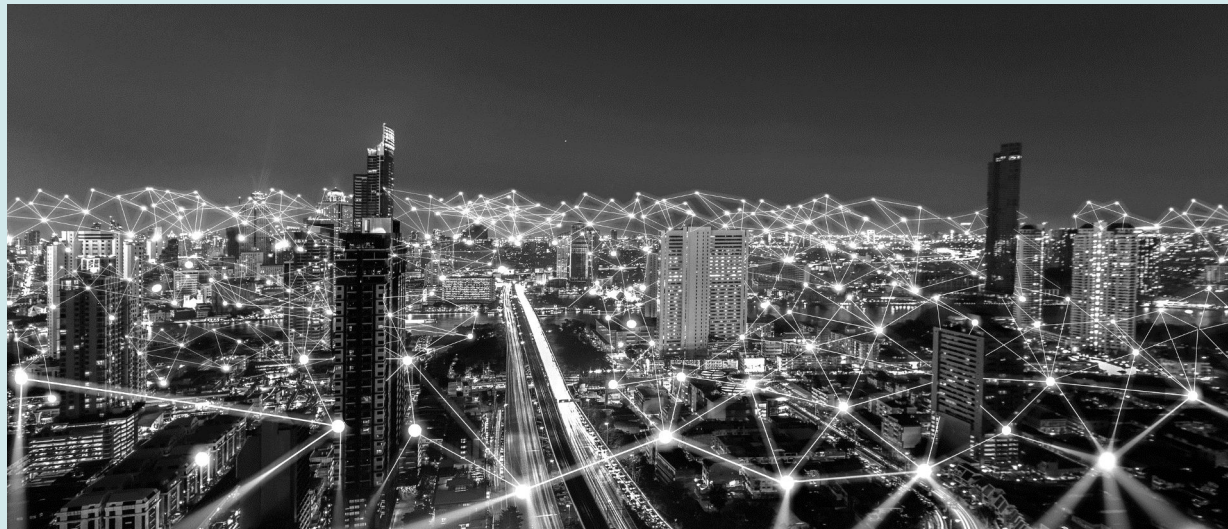
Improved development
environment (VS Code,
JupyterLab)

DEEP-HDC, AI4EOSC, AI4OS...



- **DEEP-1, DEEP-2:** Platform releases
- Platform and software tightly coupled and interlinked, difficult to self-deploy and customize
- **AI4EOSC platform** → Platform “powered by AIOS”
 - DEEP-3 → AI4EOSC-3
- **AI4OS** → software distribution
 - Possible to build custom platforms, partially integrated with AI4EOSC platform (i.e. reusing services) or not

AI4EOSC Use Cases



Agrometeorology

Integrated plant protection

Automated thermography

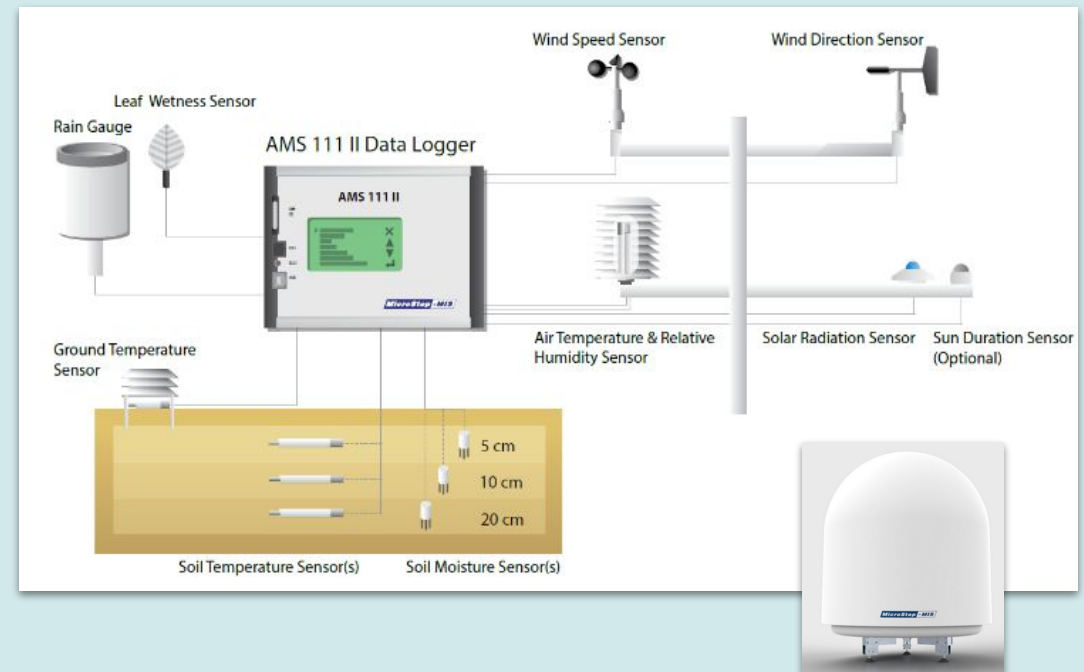
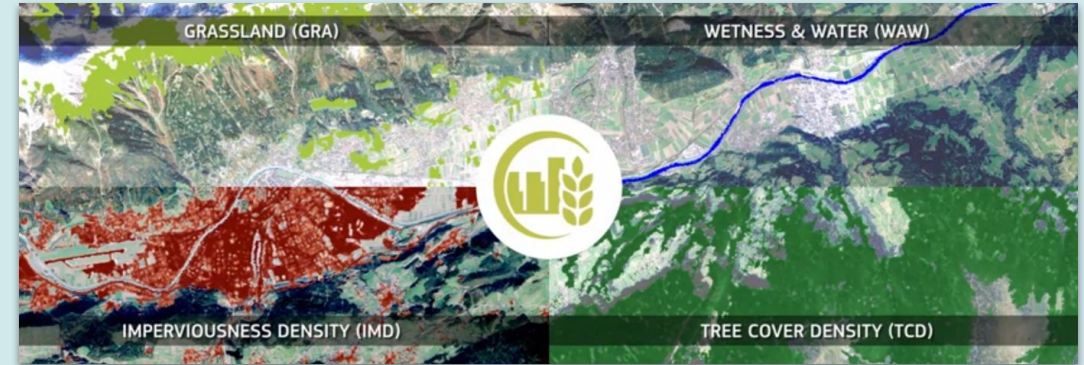
Agrometeorology

Aim: Usage of satellite imagery, in-site measurements, and weather forecasts to generate added-value products for improving farmers activity: e.g. prediction of phenological or pest development stages.

Currently: Measurement system - TRL9, prediction system - TRL3

Within AI4EOSC: Enhancement of the prediction subsystem following a Composite AI approach to combine the different machine learning models used for the different data sources

Partners: Microstep, IISAS, Predictia



Integrated plant protection



Aim: To determine the risk of disease and pests in agricultural crops and determine the phases of plant growth and the condition of crops. The developed AI models are going to be integrated into existing national advisory platforms, operated by WODR and PSNC.

Currently: WODR and PSNC operate a national advisory platform for farmers (eDWIN), which includes a network of meteorological ground stations, the Farm Management System, and ground observations of the occurrence of diseases and pests. The current solutions are based on predictive mathematical models.

Within AI4EOSC: The plan is to add to the current mathematical prediction models the ML/DL-based models used for recognition of the plant diseases and add new sources of the data. Initial focus on wheat and sugar beets and detection of the fungal diseases.

Partners: WODR, PSNC



Automated Thermography

Aim: To identify heat losses and thermal bridges in buildings and infrastructures using drone-based images and ML/DL approach in order to provide a corresponding automated AI-based service.

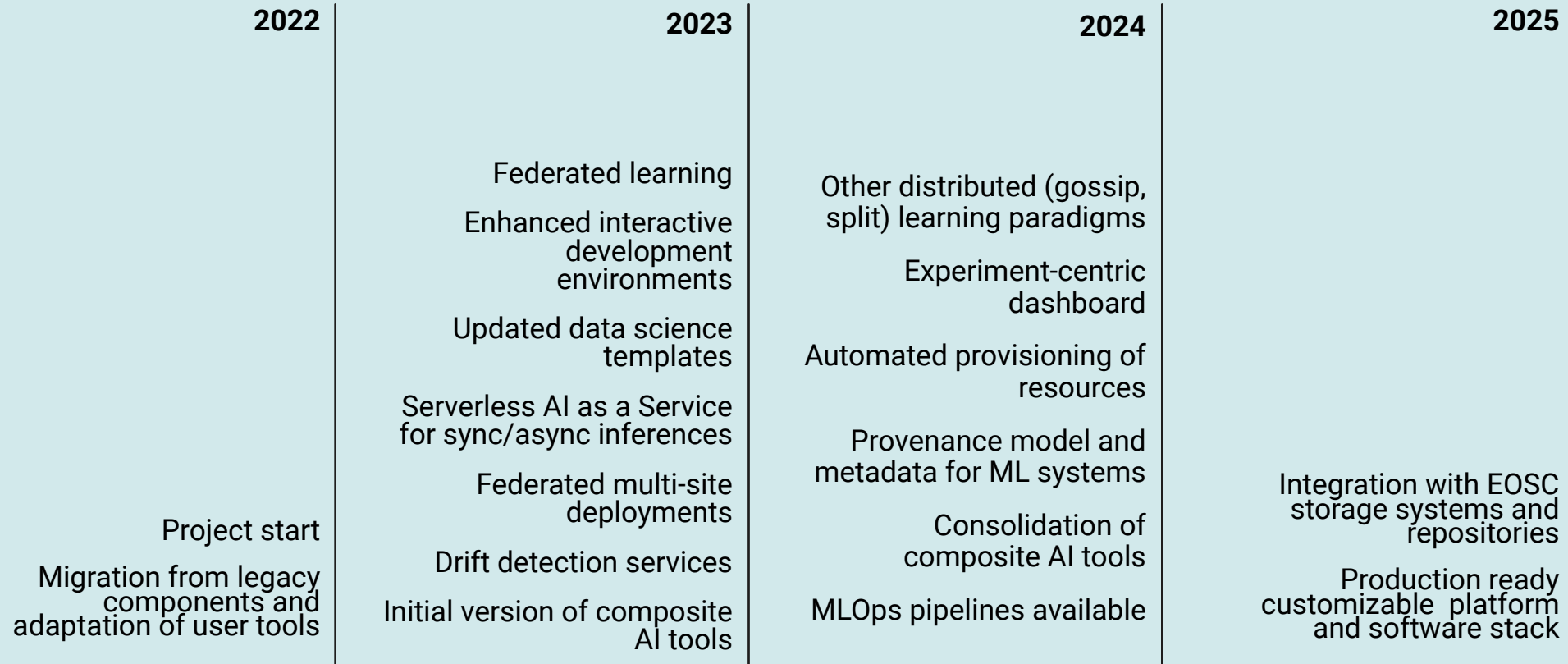
Currently: The group owns a dataset of drone-based images on urban districts and drone-based thermal images on a campus district (ca. 0.8TB). The identification of thermal bridges on roofs is already possible using DL (TRL 4). The identification of leakages in district heating networks is possible too (TRL 5/6).

Within AI4EOSC: Targets enlargement of the training dataset, AI model improvement, optimisation of the workflows, and creation of a cloud-based automated service

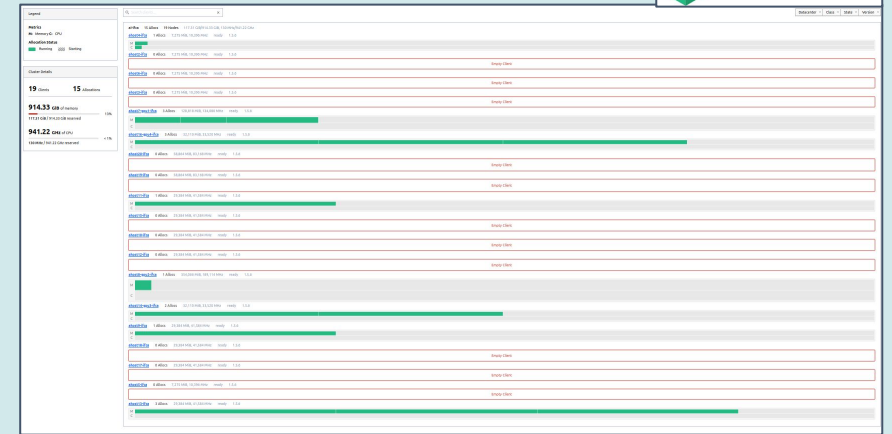
Partners: KIT (IIP, SCC)



Feature-wise timeline



Migration from legacy components






- PaaS layer
 - Moving from Apache Mesos to Hashicorp Nomad + Consul + Traefik
 - Transparent, federated and multi-site platform deployments
 - Update of deprecated internal components at PaaS orchestrator level
- Improved accounting, security and monitoring
- Development of AI4-PAPI, new dashboard and CLI tools
 - Easier to migrate from one COE to another one (just in case)
 - Specific routes for AI and ML (e.g. Federated learning)
 - Removal of TOSCA dependency for user apps
 - Inclusion of additional sidecar tasks (i.e. storage, accounting, monitoring) at API level (i.e. not user-managed)
- Integration with EGI-Check-IN and other OpenID Connect providers

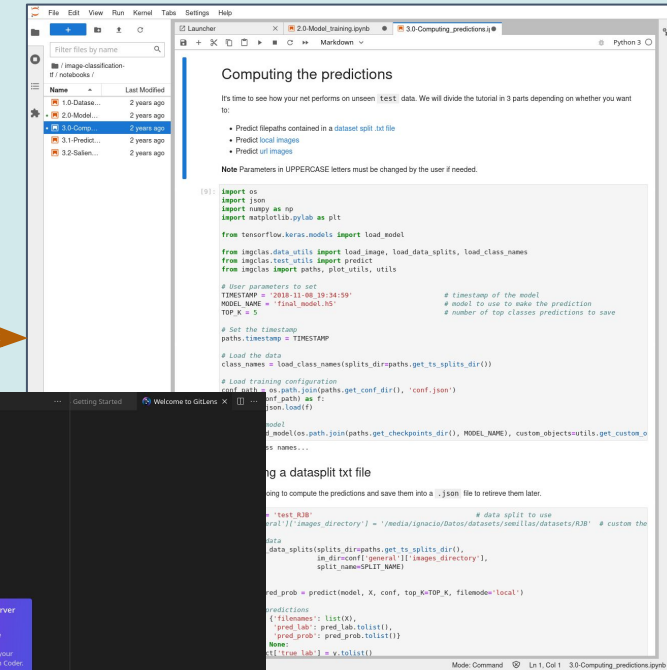
The screenshot shows the AI4 eosc Marketplace dashboard. It features a grid of service cards, each representing a different AI/ML application. Each card includes a title, a brief description, and buttons for 'Trainable', 'Inference', and 'Pre-trained'.

Service Name	Description	Trainable	Inference	Pre-trained
Dogs breed detector	Identify a dogs breed on the image (133 known breeds)	Yes	Yes	Yes
DEEP OC Massive Online Data Streams	Deep learning for proactive network monitoring and security protection.	Yes	Yes	No
DEEP OC Retinopathy Test	A Tensorflow model to classify Retinopathy.	Yes	Yes	Yes
Train an image classifier	Train your own image classifier with your custom dataset. It comes also pretrained on the 1K ImageNet classes.	Yes	Yes	Yes
Plants species classifier	Classify plant images among 10K species from the Naturalist dataset.	Yes	Yes	Yes
Conus species classifier	Classify conus images among 70 species.	Yes	Yes	Yes
Phytoplankton species classifier	Classify phytoplankton images among 60 classes.	Yes	Yes	Yes
Seed species classifier	Classify seeds images among 700K species.	Yes	Yes	Yes
Upscale multispectral satellites images	Upscale (superresolve) low resolution bands to high resolution in multispectral satellite imagery.	Yes	Yes	Yes
Speech keywords classifier	Train a speech classifier to classify audio files between different keywords.	Yes	Yes	No
Body pose detection	Detect body poses in images.	Yes	Yes	Yes
Train an audio classifier	Train your own audio classifier with your custom dataset. It comes also pretrained on the 527 AudioSet classes.	Yes	Yes	Yes
2D semantic segmentation	2D semantic segmentation trained on the Valkyrie dataset	Yes	Yes	Yes
A4OS Development Environment	This is a Docker image for developing new modules	No	Yes	No
TF Benchmarks	tf_cnn_benchmarks accessed via DEEPaaS API	Yes	Yes	No
Object Detection and Classification with Pytorch	A trained Region Convolutional Neural Network (Faster RCNN) for object detection and classification.	Yes	Yes	Yes
Bird sound classifier	Classify audio files among bird species from the Xenocanto dataset.	Yes	Yes	Yes
Artistic style transfer	A module to apply artistic style transfer using pytorch.	Yes	Yes	No
Chest x-ray image classifier	Classify chest x ray images in pathological and non pathological with this x ray classifier.	Yes	Yes	Yes
demo_app	A toy application for demo and testing purposes. We just implement dummy inference, ie we return the same inputs we are fed.	Yes	Yes	Yes

Enhanced interactive environments



Name	Status	Container name	GPUs	Creation Time	Actions
test-vscode	Running	deephd/deep-oc-generic-dev:latest	0	2023-06-20 09:14:29	  
test-jupyter	Running	deephd/deep-oc-plants-classification-0:latest	0	2023-06-20 09:10:18	  

```

import os
import json
import numpy as np
import matplotlib.pyplot as plt

from tensorflow.keras.models import load_model
from imgcls_data_utils import load_image, load_data_splits, load_class_names
from imgcls_test_utils import predict
from imgcls import paths, plot_utils, utils

# User parameters to set
TIMESTAMP = '2018-11-08_19:34:59' # timestamp of the model
MODEL_NAME = "final_model.h5" # model to use to make the prediction
TOP_K = 5 # number of top classes predictions to save

# Set the timestamp
paths.timestamp = TIMESTAMP

# Load the data
class_names = load_class_names(paths.get_ts_splits_dir())

# Load training configuration
conf_path = os.path.join(paths.get_conf_dir(), 'conf.json')
conf = json.load(f=open(conf_path, 'r'))

model = load_model(os.path.join(paths.get_checkpoints_dir(), MODEL_NAME), custom_objects=utils.get_custom_objects())

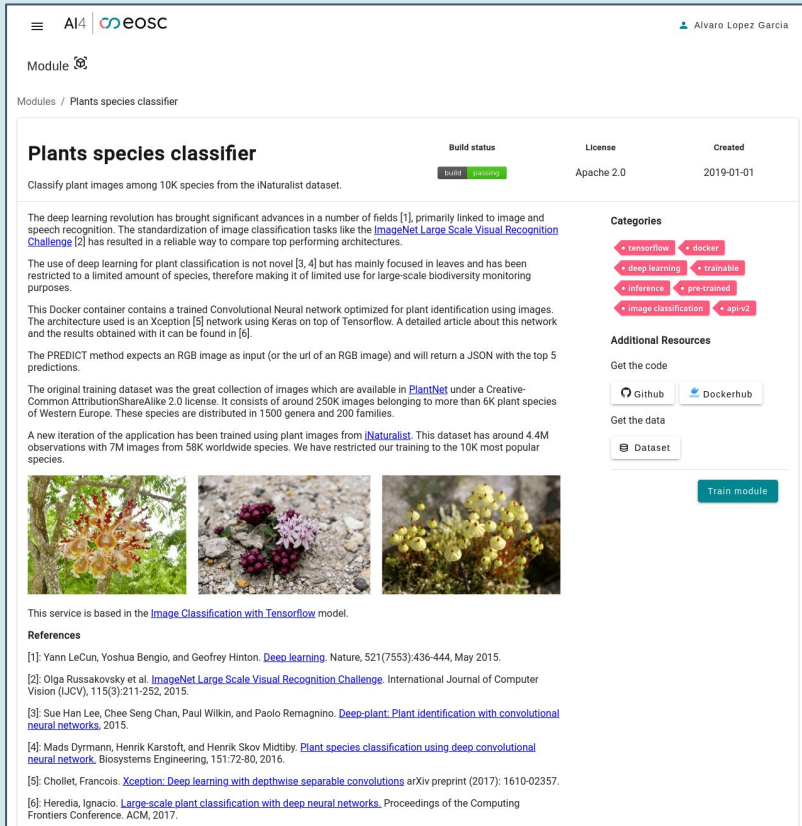
# Compute the predictions and save them into a .json file to retrieve them later.
def predict_dir(directory):
    # data split to use
    split = 'train' if 'train' in directory else 'test'

    data_splits = paths.get_splits_dir(split)
    data_splits = paths.get_splits_dir(split)

    # Predict local images
    pred_prob = predict(model, X, conf, top_k=TOP_K, filename='local')

    # Save the predictions
    pred_prob = predict(model, X, conf, top_k=TOP_K, filename='local')

    # Save the predictions
    pred_prob = predict(model, X, conf, top_k=TOP_K, filename='local')
    
```



Plants species classifier

Classify plant images among 10K species from the iNaturalist dataset.

Build status: Running License: Apache 2.0 Created: 2019-01-01

The deep learning revolution has brought significant advances in a number of fields [1], primarily linked to image and speech recognition. The standardization of image classification tasks like the [ImageNet Large Scale Visual Recognition Challenge](#) [2] has resulted in a reliable way to compare top performing architectures.

The use of deep learning for plant classification is not novel [3, 4] but has mainly focused in leaves and has been restricted to a limited amount of species, therefore making it of limited use for large-scale biodiversity monitoring purposes.

This Docker container contains a trained Convolutional Neural network optimized for plant identification using images. The architecture used is an Xception [5] network using Keras on top of Tensorflow. A detailed article about this network and the results obtained with it can be found in [6].

The PREDICT method expects an RGB image as input (or the url of an RGB image) and will return a JSON with the top 5 predictions.

The original training dataset was the great collection of images which are available in [PlantNet](#) under a Creative-Common Attribution-ShareAlike 2.0 license. It consists of around 250K images belonging to more than 6K plant species of Western Europe. These species are distributed in 1500 genera and 200 families.

A new iteration of the application has been trained using plant images from [iNaturalist](#). This dataset has around 4.4M observations with 7M images from 58K worldwide species. We have restricted our training to the 10K most popular species.

Categories

- tensorflow
- docker
- deep learning
- trainable
- inference
- pre-trained
- image classification
- api-v2

Additional Resources

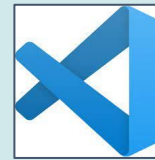
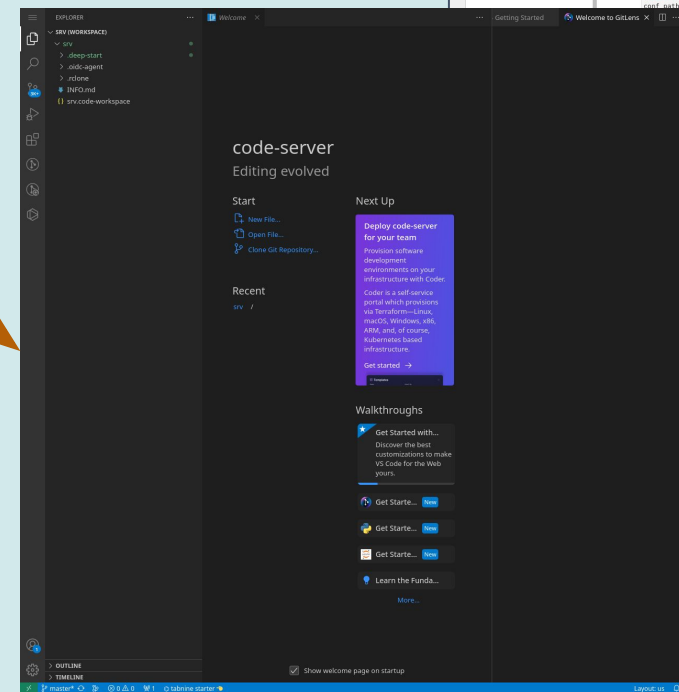
Get the code: [GitHub](#) [Dockerhub](#)

Get the data: [Dataset](#)

[Train module](#)

References

- [1]: Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. [Deep learning](#). Nature, 521(7553):436-444, May 2015.
- [2]: Olga Russakovsky et al. [ImageNet Large Scale Visual Recognition Challenge](#). International Journal of Computer Vision (IJCV), 115(3):211-252, 2015.
- [3]: Sue Han Lee, Chee Seng Chan, Paul Wilkin, and Paolo Remagnino. [Deep-plant: Plant identification with convolutional neural networks](#), 2015.
- [4]: Mads Dyrmmann, Henrik Karstoft, and Henrik Skov Midtby. [Plant species classification using deep convolutional neural network](#). Biosystems Engineering, 151:72-80, 2016.
- [5]: Chollet, Francois. [Xception: Deep learning with depthwise separable convolutions](#) arXiv preprint (2017): 1610-02357.
- [6]: Heredia, Ignacio. [Large-scale plant classification with deep neural networks](#). Proceedings of the Computing Frontiers Conference. ACM, 2017.

code-server
Editing evolved

Next Up

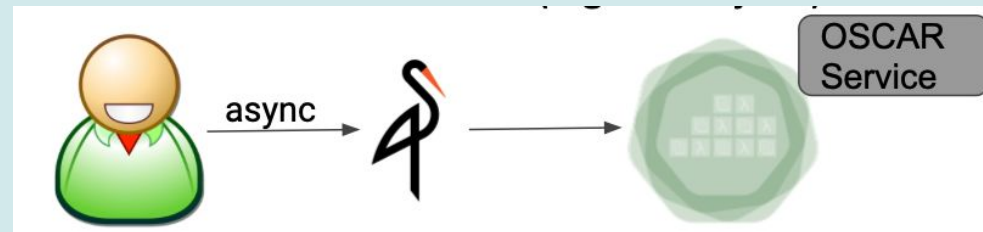
- Deploy code-server for your team
- Provision software development environments on your infrastructure with Code-Server
- Code as a self-service: partial self-hosted via Terraform+Linux, macOS, Windows, etc. ARM, and of course Kubernetes based infrastructure

Walkthroughs

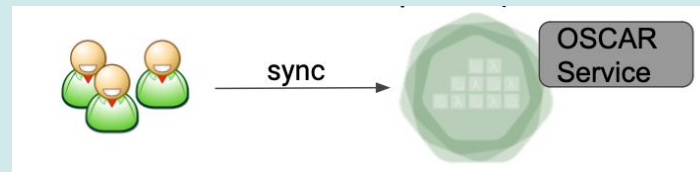
- Get Started with...
- Get Started with...
- Get Started with...
- Get Started with...

Serverless AI as a service

- OSCAR (<https://oscar.grycap.net>) will be used to run the AI models for inference (AI as a Service)
 - Serverless event-driven execution
 - Asynchronous Mode: Files uploaded to the object-store trigger the invocation of a data-processing script that is run inside a container (out of user-defined Docker image) within a scalable Kubernetes cluster (e.g. batch jobs)



- Synchronous mode: Scalable HTTP-based endpoints (based on KNative)



- <https://inference.cloud.ai4eosc.eu/ui/#/login>

Drift detection libraries and tools

- Monitoring of models in production is not enough
 - Model learns from data, data is not stationary
 - Concept learnt by them model may change over time
- Data and concept drift detection → essential to build more robust models
- Frouros: state-of-the-art library for drift detection in ML problems
 - <https://github.com/IFCA/frouros>
- Ongoing work towards online services for drift detection



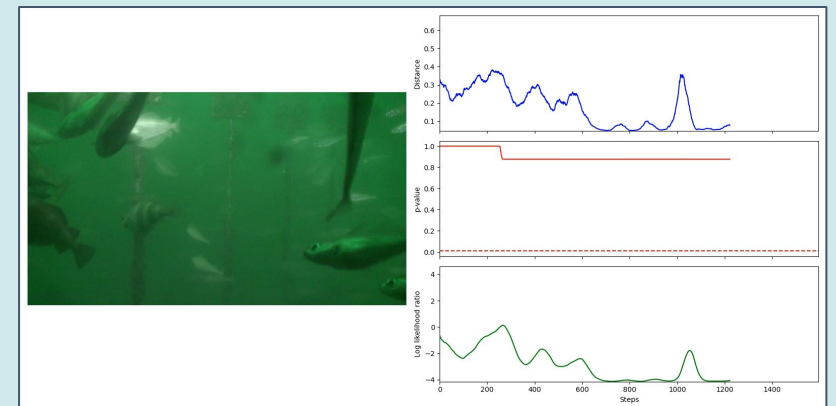
FROUROS

CI passing | codecov 91% | docs passing | downloads 34k | release v0.3.2 | python 3.8 | 3.9 | 3.10 | 3.11 | License BSD 3-Clause

Frouros is a Python library for drift detection in machine learning systems that provides a combination of classical and more recent algorithms for both concept and data drift detection.

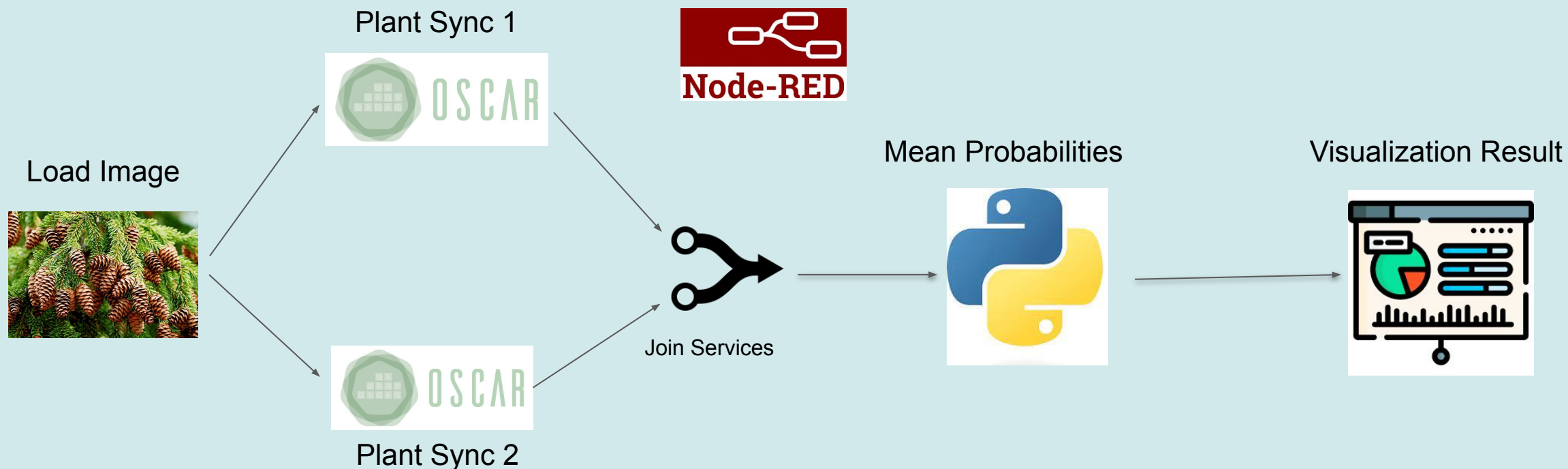
"Everything changes and nothing stands still"
"You could not step twice into the same river"

Heraclitus of Ephesus (535-475 BCE.)



Example: data drift detection in underwater video

Composite AI models, initial version



- Use case: multiple AI models can be triggered for inference and later aggregate the results for enhanced accuracy
- Reuse functions (subflow)
- Visual support (drag & drop + customization)
- Minimize orchestration costs

AI4EOSC challenges

Integration of disparate resources from different providers across EU e-Infras

Data access and privacy-preserving model training on sensitive data

Correct handling of metadata and quality aspects of AI/ML/DL assets

Exploitation of automation, infrastructure as code and automated orchestration of resources

Federated learning, gossip and split learning, together with national security measures (i.e. homomorphic encryption)

Application of RDA recommendations and participation in RDA FAIR for ML WG

AI4EOSC challenges

Data access, remote access,
provider disparity

Access to accelerators and
resources

AI-related developments are
exploding (AI bandwagon)

Sidecar containers providing
tailored and transparent access
for storage services

Users do not need to deal with
low-level details, just interact
with preferred IDE

Focus on scientists and the
EOSC ecosystem

Key impacts and outcomes

- Increased number of services in the EOSC Exchange
 - AI/ML generic (AI4EOSC platform) or community specific (e.g. iMagine, see Gergely's talk https://whova.com/portal/webapp/egi_202305/Agenda/3088789)
- Transforming development of AI models for science in the EOSC
 - FAIR, reproducibility of pipelines, model provenance
 - Best practices for development AI and ML models
- Improvement of robustness of AI systems -> Trustworthy AI
 - **MLOps** infrastructure and services for AI scientists
 - **Drift detection** tools and services to assess data/model/inference validity
 - **Provenance** of models (reproducibility), model metadata
 - FAIR-ification of ML assets
- (Main, relevant) Outcomes
 - AI platform for the EOSC (AI4EOSC platform)
 - Integrated popular IDE environments
 - Advanced ML features: federated learning, homomorphic encryption, privacy tools
 - Composite AI, cross discipline pipelines, serverless platforms
 - Software stack (AI4OS) to build customized AI platforms
 - Best-practices documents for the AI community of practice in the EOSC

Collaboration: INFRAEOSC and beyond

- Starting and ongoing collaborations with EOSC (i.e. INFRAEOSC) projects to provide AI-based tools and services
 - FAIR-EASE, RAISE, EOSC4Cancer, BlueCloud
 - Exploring collaborations on FAIR-ification of AI assets (RDA, FAIR-IMPACT)
- Outside the EOSC realm
 - iMagine → AI based imaging data and services for aquatic science
 - AioD → Connectors for AI4EOSC assets
- Open for additional collaborations
 - Reach out at deep-po@listas.csic.es
 - Collaboration with industry through EOSC-DIH (<https://eosc-dih.eu/>)



AI4

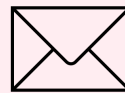
 eOSC



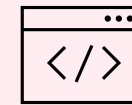
Co-funded by
the European Union



AI4EOSC



ai4eosc-po@listas.csic.es



ai4eosc.eu

Reach us!

Thank you for your attention

Project Coordinator: Álvaro López García - aloga@ifca.unican.es