EXTRACT

A distributed data-mining software platform for extreme data across the compute continuum

TASKA

Transient Astrophysics with a Square Kilometre Array pathfinder

Julien N. Girard, **Baptiste Cecconi** and the EXTRACT collaboration

See deliverables & demos on extract-project.eu

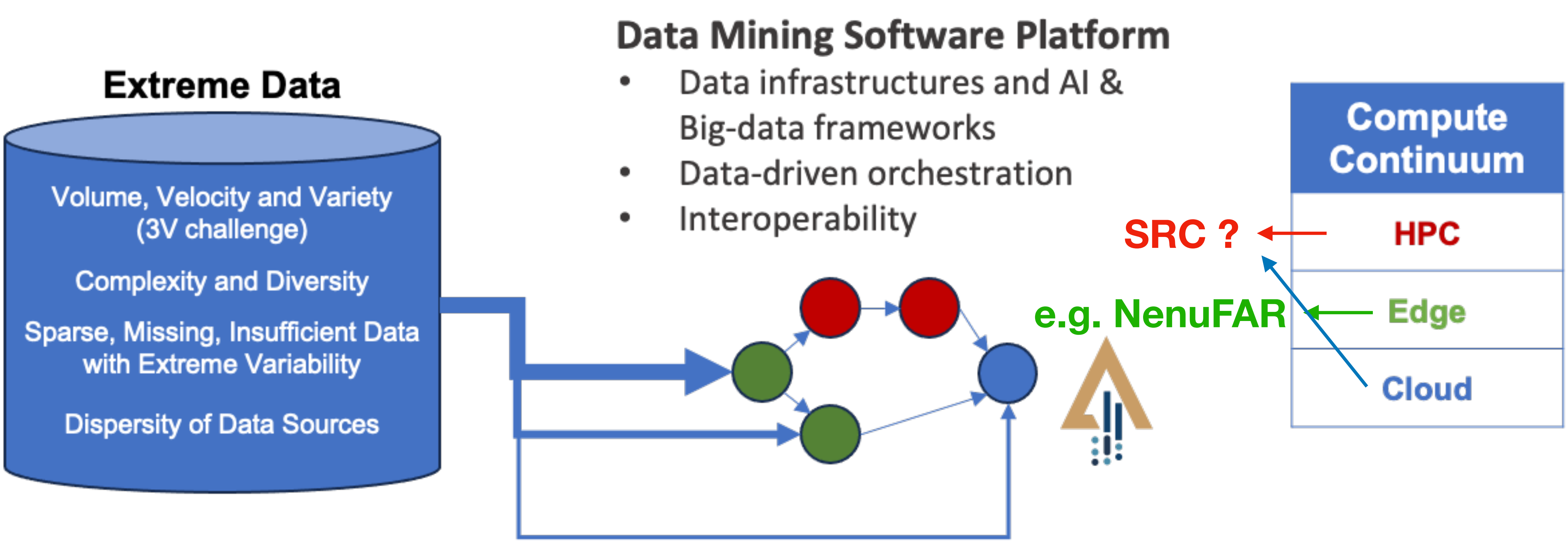
The EXTRACT Project has received funding from the European Union's Horizon Europe programme under grant agreement number 101093110

EU project

https://extract-project.eu/

A distributed data-mining software platform for extreme data across the compute continuum

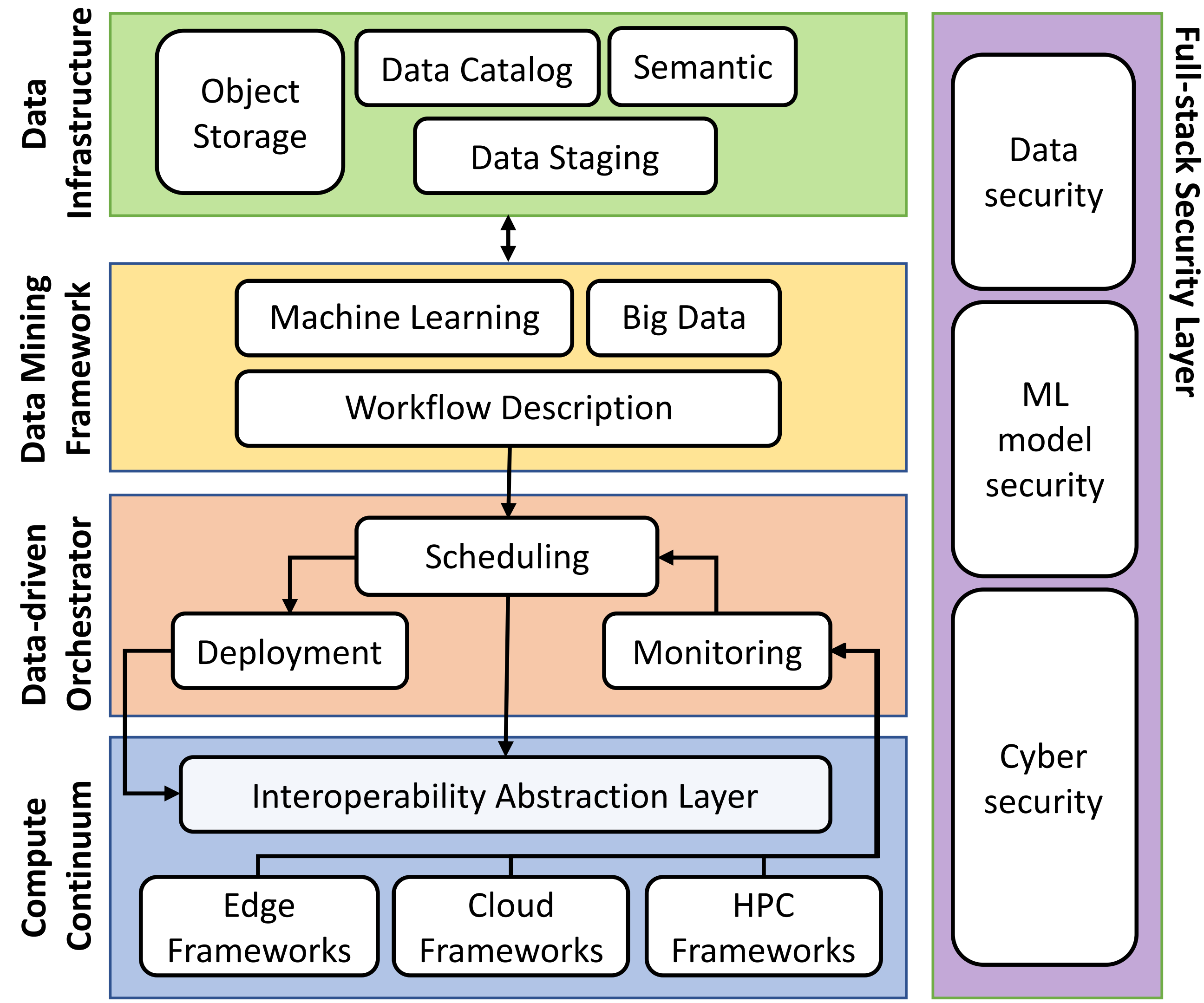
EXTRACT aims to create a data-mining **software platform** for **extreme data** across the **compute continuum**



EXTRACT

A distributed data-mining software platform for extreme data across the compute continuum

- Handle the complete lifecycle and value chain of extreme data
  - **Data collection** across highly distributed and heterogeneous sources
  - **Data mining** of meaningful, accurate, reliable and useful knowledge
  - **Secure and trustworthy used of knowledge** by applications and end users
- Everything looks local



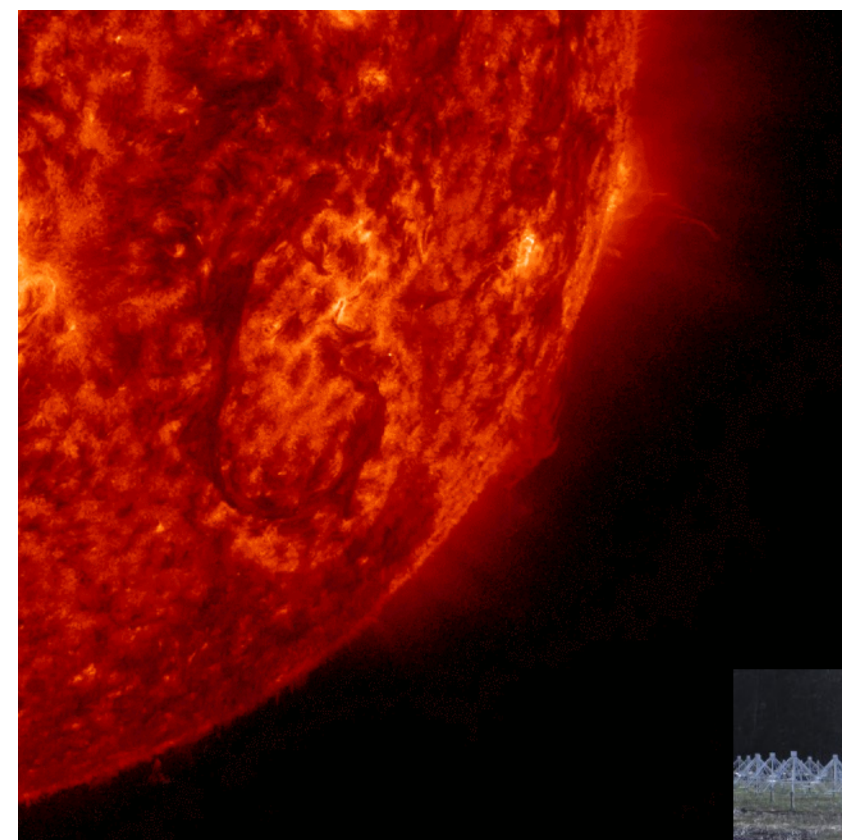
EXTRACT Use-Cases, sharing the same platform

PER

Personalised Evacuation Route (PER) in the City of Venice based on an Urban Digital Twin and an AI engine

TASKA

Transient Astrophysics with the Square Kilometre Array pathfinder (TASKA) NenuFAR generating high-volume and high-velocity data



NenuFAR *New extension in Nançay Upgrading loFAR*

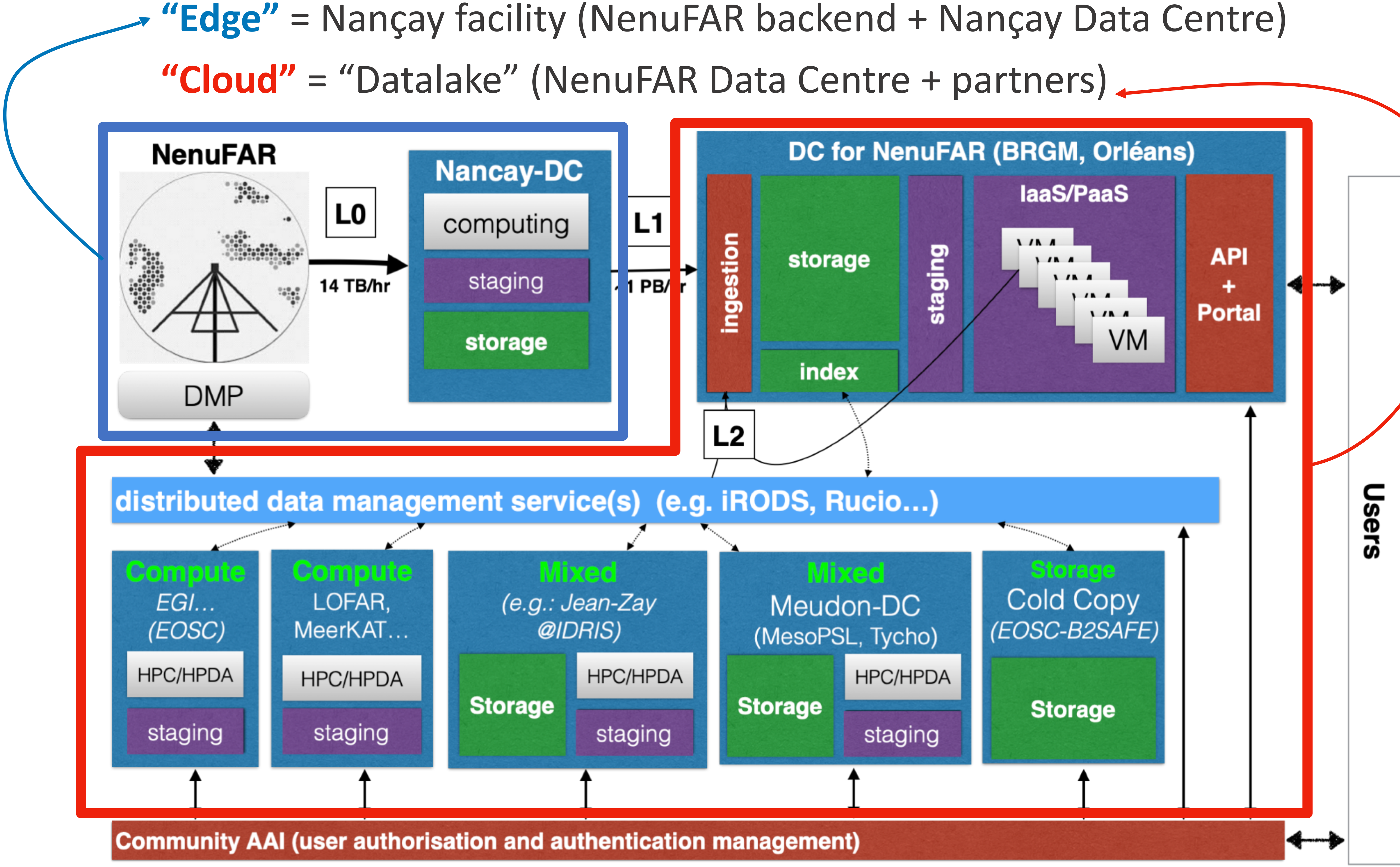
Pathfinder de SKA (LOW) , Infrastructure de recherche **F = 20-80 MHz**

N_A~2000 antennes **Fonctionnement en mode réseau phasé et interféromètre**

NenuFAR digital infrastructure

"Edge" = Nançay facility (NenuFAR backend + Nançay Data Centre)

"Cloud" = "Datalake" (NenuFAR Data Centre + partners)

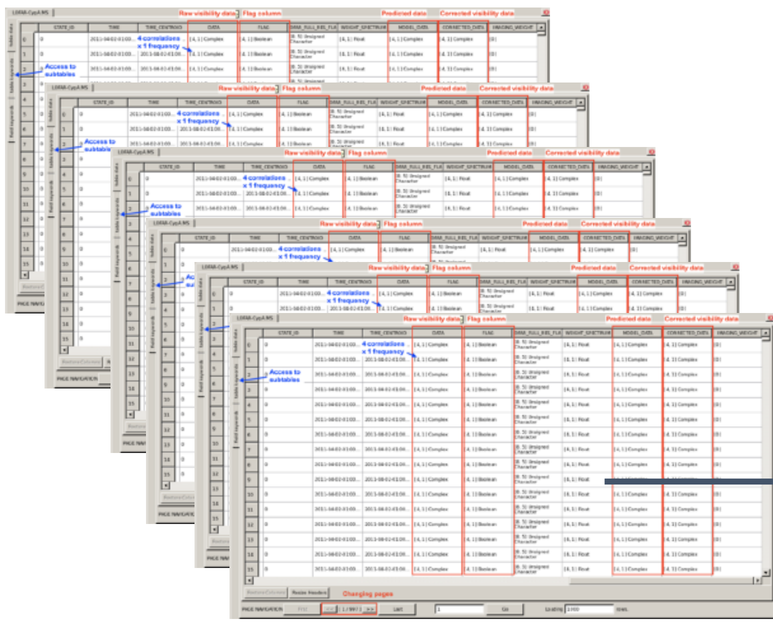


TASKA Use Case Overview

- **Use Case A:** Early detection and selective resolution data recording (space optimality)
- **Use Case C:** Workflow orchestration of interferometric data processing with a focus on improving the processing speed, accuracy and automation on large datasets
- **Use Case D:** Prototype development for “dynamic” imaging of the variable Universe

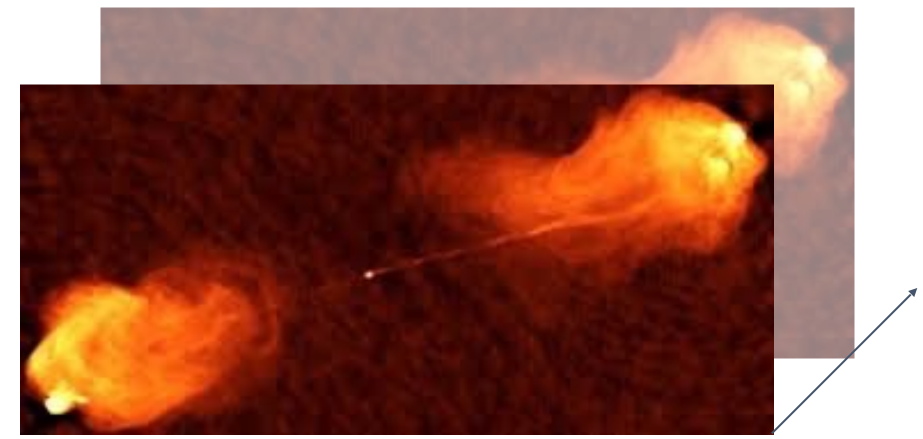
  (DL transient imaging)
- **Use Case E:** Advanced data reduction workflows for multi-dimensional real-time analysis and inference (joining A and C together)

Use-Case C: Workflow orchestration for radiointerferometry



Data transfer
data processing

(Flagging, Rebining,
Calibration, Imaging)



Final product:
Image cubes

time/freq

Starting dataset: Visibilities
(Measurement Sets (MS) Format)

Optimal dataset distribution ?
(Multiple sites)

Flexible reduction?

(Multiple tools)
from combination of
known analytic “bricks”

Analytics
Multiple tools
(scientific quality, fidelity)

What is required ?

- Data **volumes** & data **multi-site storage** (object storage)

- Data **transfer capabilities** between storage sites

- Processing **ressource allocation** & **optimization** (partitionning, //)

- Data **provenance** (reproducibility + metadata)

- Accounting for the explosion in the **number of pipelines** (multiple science cases)

generality

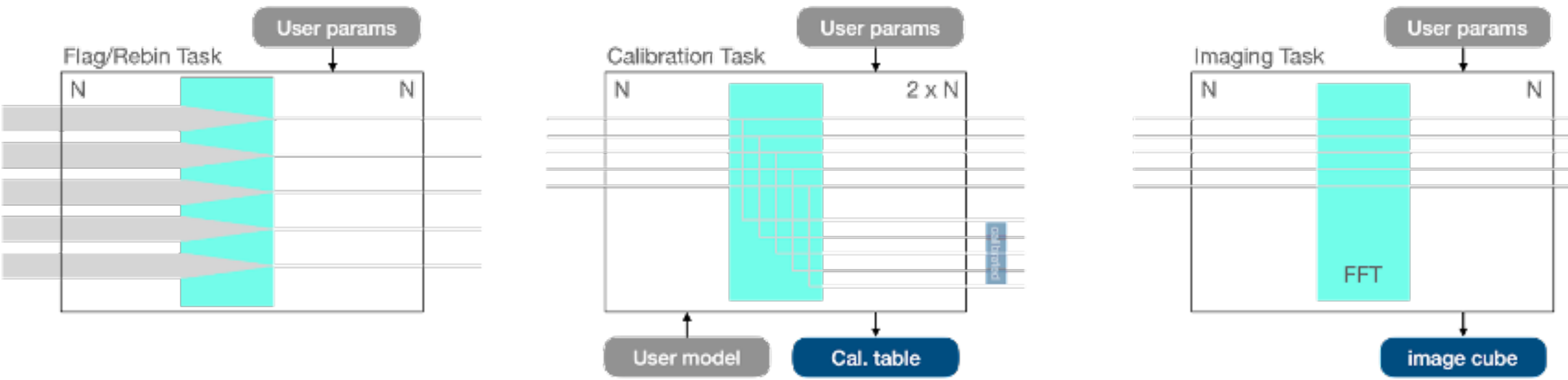
- Lack of **common platforms** for orchestration (**heterogeneity**)

Data <-> HPC <-> Cloud <-> Clients

- Transparency / thresholds / **Agency** for users (i.e. scientists)
    - Design & test workflows from building blocks ("interactive" mode)
    - Deploy workflow for production ("automatic" mode)

Use-Case C: Designing a workflow orchestration framework

- **Processing Modules:** Handles data cleaning, calibration, and transformation tasks, converting raw inputs into usable scientific formats. Can host different tools (now DP3, future: kMS, DDFacet)



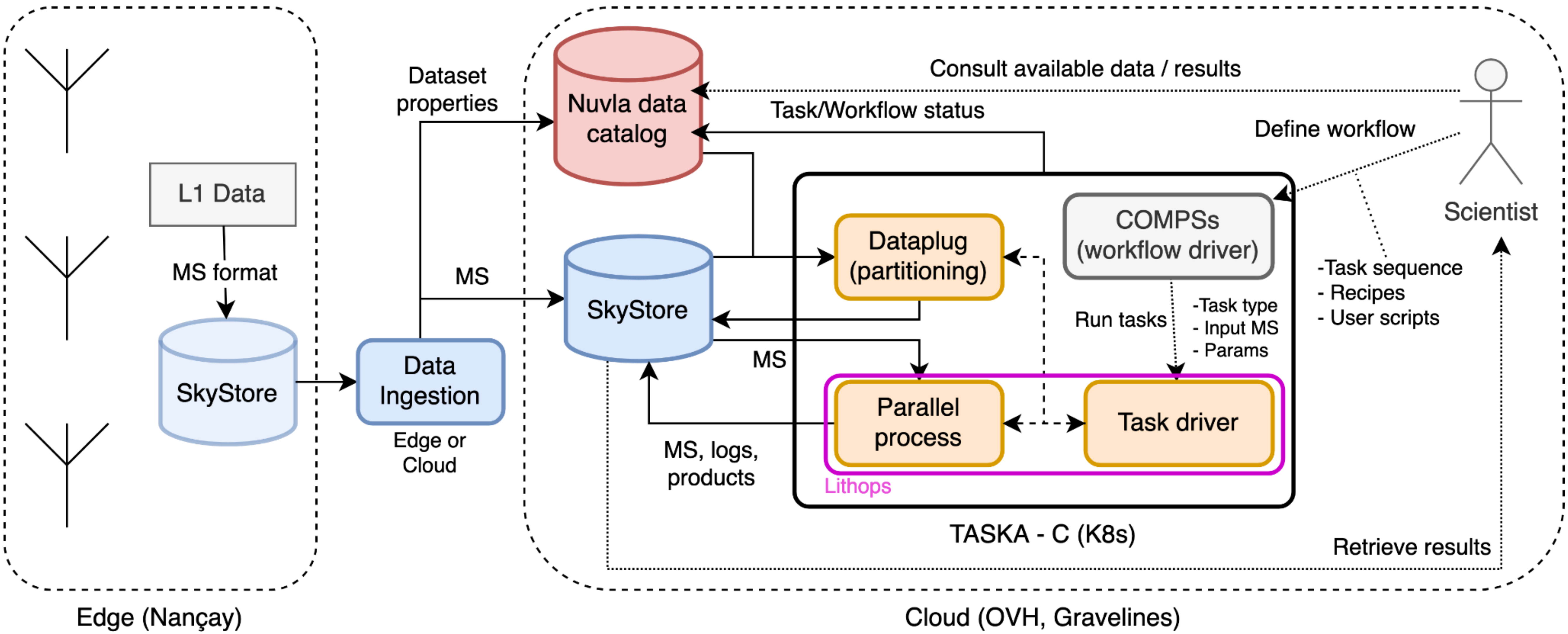
- **Orchestration Layer:** Manages the flow of data between each processing tasks, optimizing resource in the background (*without the scientist knowing*) and ensuring timely data handling.

- **Visualization Tools:** Provides real-time access to processed data, at all stages, allowing scientists to validate intermediate outputs and make decisions on the workflow recipes.

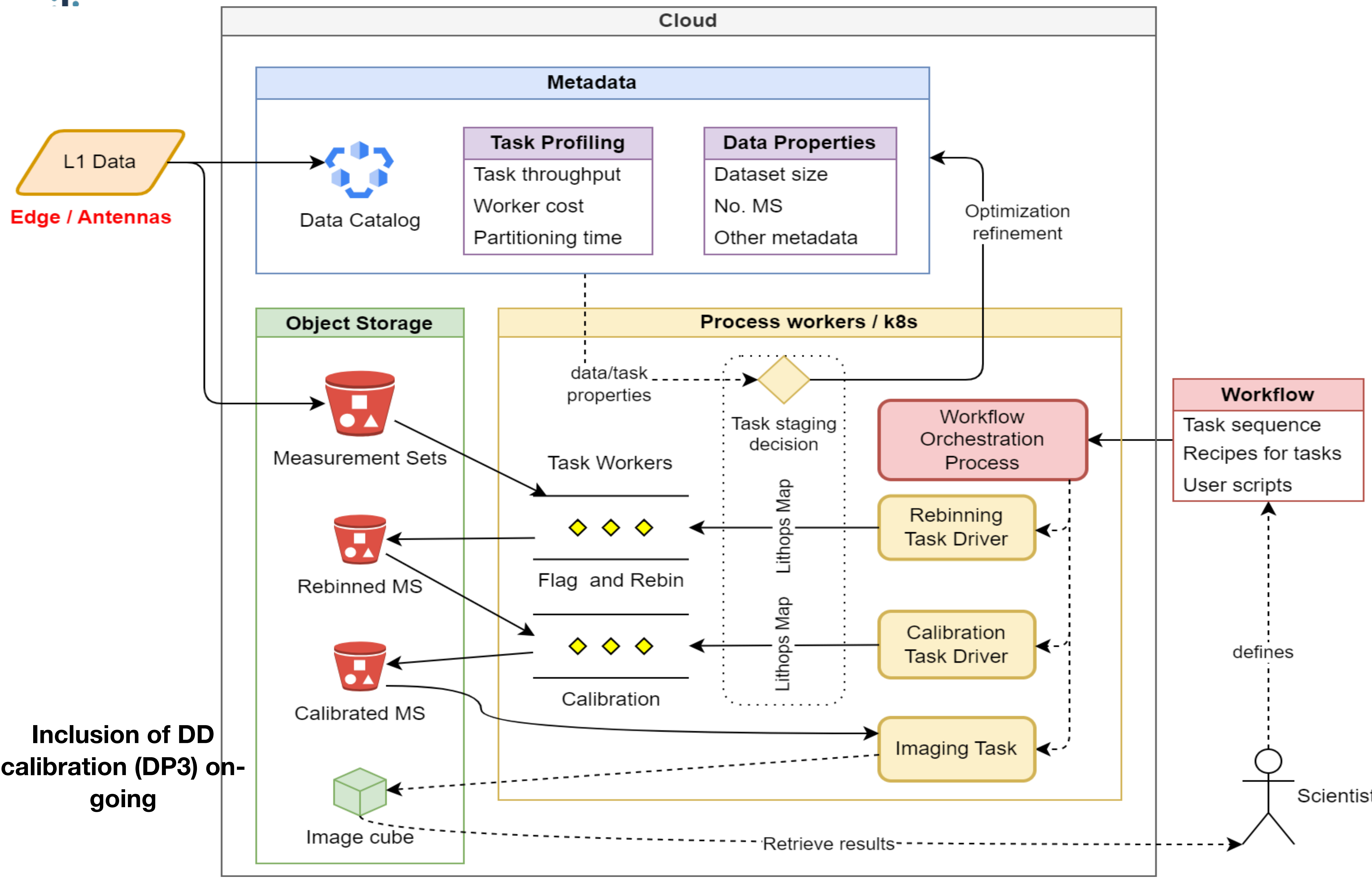
Integration and Interoperability:

- The architecture supports seamless integration with existing astrophysical data platforms and tools, enhancing the usability and impact of the MVP.

Use-Case C: Architecture view



Use-Case C: User view

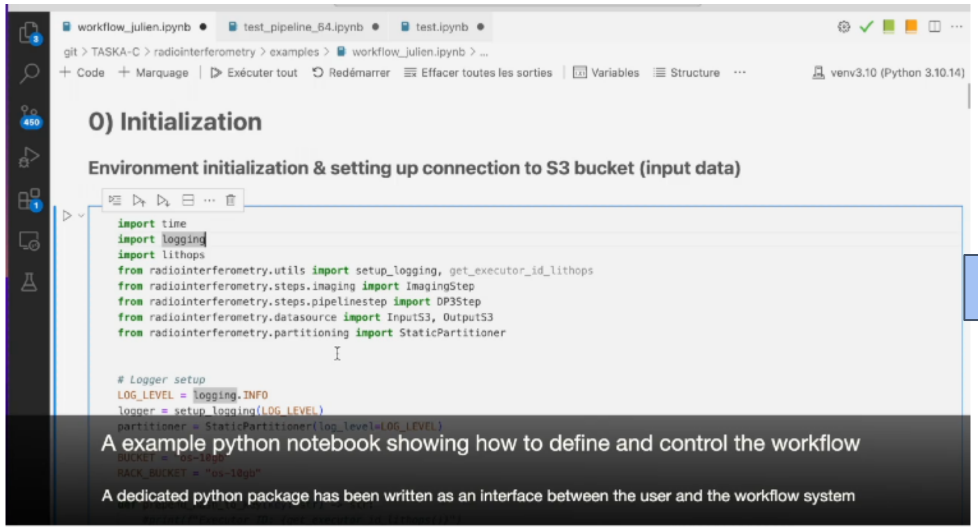


Inclusion of DD calibration (DP3) on-going

TASKA - MVP “Interactive” Workflow

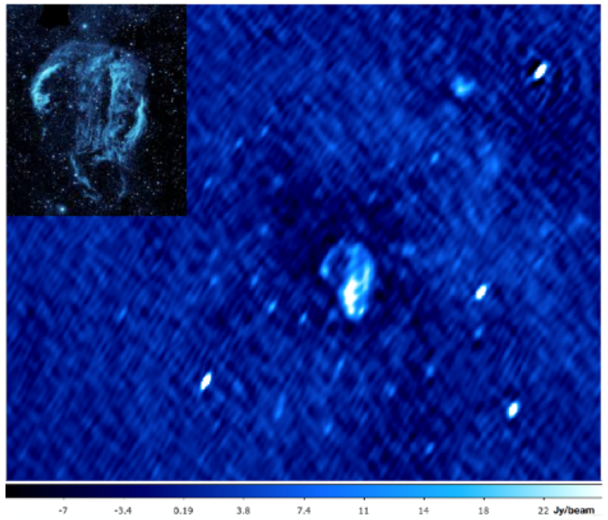
- Built as a “wrapper” that interacts with the astronomy community tools
  High potential impact because of the platform deployment in other communities (security, medical, resource management, etc.)
- Easy to invoke, easy to code, easy to customize, easy to “chain”: *natively made for workflows*
- Each task has a “definition” block and a “run” block: *separating the workflow building from its running*
- Run as a python script or in a python notebook (cf. DEMO video)

```python
import time
import logging
import lithops
from radiointerferometry.utils import setup_logging, get_executor_id_lithops
from radiointerferometry.steps.imaging import ImagingStep
from radiointerferometry.steps.pipelinestep import DP3Step
from radiointerferometry.datasource import InputS3, OutputS3
from radiointerferometry.partitioning import StaticPartitioner
```



Controlled through a python notebook (S3, data partitioning, worker management, …)

The final products are then retrieved on the scientist computer



…as if the process and data were **local**

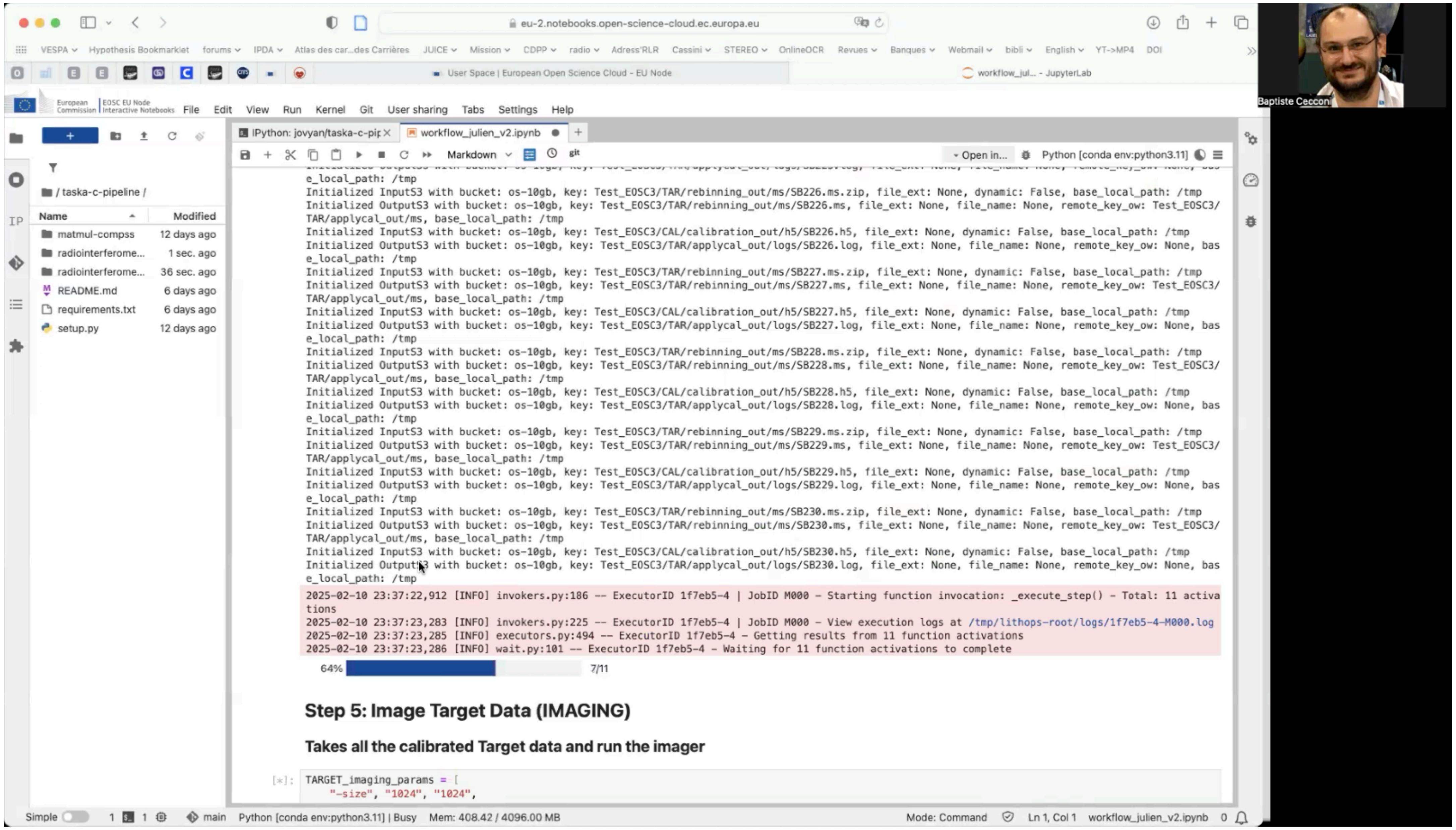
TASKA - Demo

- Demo context:

  running the TASKA-C MVP on OVH K8S cluster

  Using a Jupyter notebook “somewhere” else: let’s run it on the EOSC-EU Node !

  (https://open-science-cloud.ec.europa.eu/)



Current MVP Status

Running on K8S managed by OVH

- Three steps implemented: Flag-rebin step; Calibration step ; Imaging step
- Interactive mode (Jupyter notebook)
- Data available on local S3 storage
- Fully remote processing

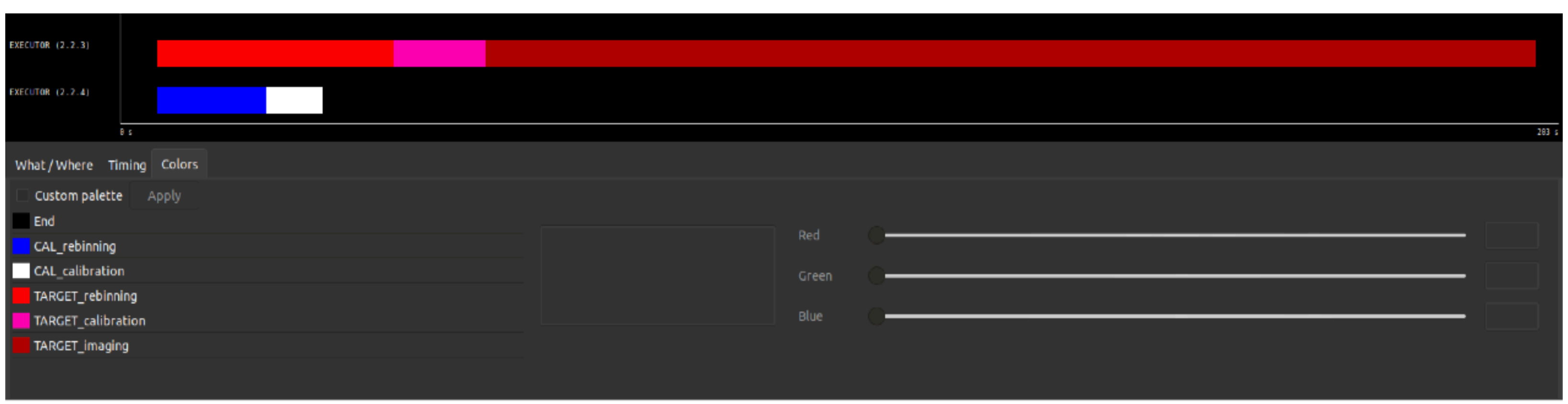
Ongoing developments:

- Testing deployment on other K8S:
  - EGI K8S [Rancher]: testing
  - EOSC EU Node K8S [OKD]: preparing tests
  - ObsParis K8S [OKD] and BRGM K8S [Rancher]: next
- Automated workflow mode
- Integrate data catalogue for automated workflow
- Integrate provenance management for efficient reprocessing
- Integrate new step with HPC - slurm
- Implement data staging and data clean-up steps

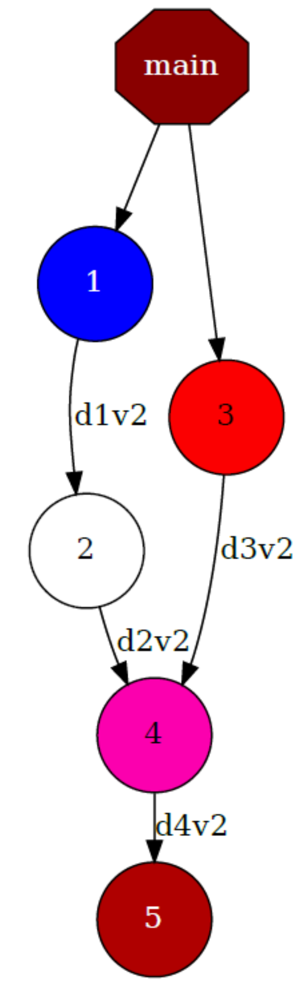
TASKA - MVP “Automated” Workflow

Implementing the chaining of tasks and decision making capabilities

- Less interactivity, but larger autonomy for production
  Each tasks are chained with dependencies and running conditions (splitting, merging, intermediate user scripts)
- Built from the “interactive” workflow developed by the scientist
  Once the processing recipe is validated, it is forged into a production workflow
- Ability to rerun from a given point
  If a task fails or gives unsatisfactory results, it can be rerun with different set of parameters.
- Processing traceability by intermediate logging and meta-data recording

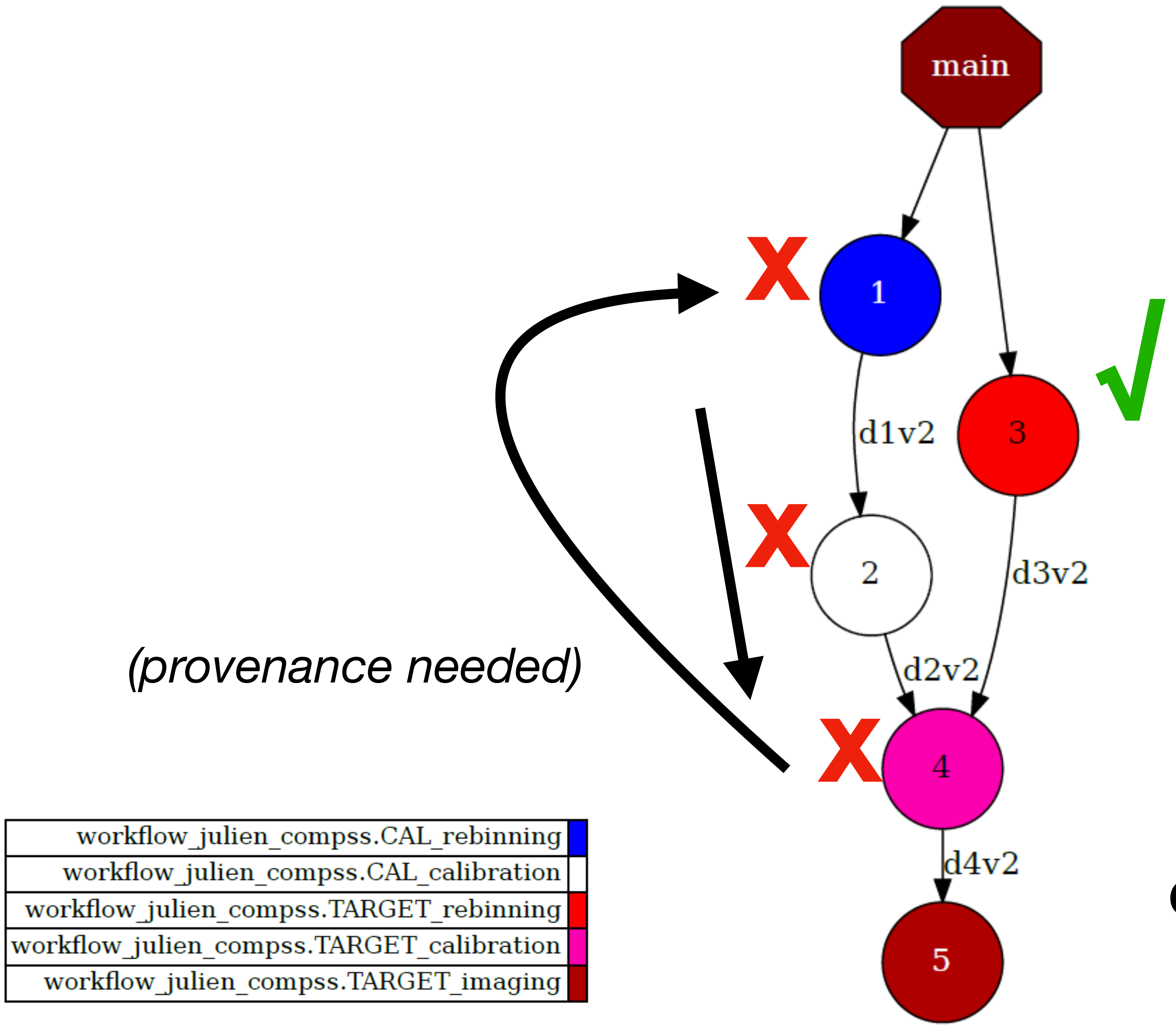


workflow_julien_compss.CAL_rebinning
workflow_julien_compss.CAL_calibration
workflow_julien_compss.TARGET_rebinning
workflow_julien_compss.TARGET_calibration
workflow_julien_compss.TARGET_imaging



TASKA - MVP "Automated" Workflow

Implementing the chaining of tasks and decision making & replay capabilities



On-going development on EOSC, EGI/CESNET, OVH, (Soon Nançay/Obs)

EXTRACT - TASKA - Summary

TASKA-C

- We have developed a **framework for distributed data computing** on cloud clusters
- Currently validating
  - unsupervised/automated workflow
  - running a step on an HPC resource
  - running on a multi-cluster scale (data distributed in several data centers)
- Application on NenuFAR (SKA pathfinder)
- Clear huge potential for SRCNet

Provenance in data catalogue => VOProv
Catalogue of datasets => ObsTAP + radio extension

Job management in HPC => UWS?

Other work not addressed in this presentation:

TASKA-A

- Real time detection (possibly with AI) on high resolution data stream (dynamic spectra): implemented on NenuFAR beamformer backend

Event Detection => VOEvent
Temporal-Spectral features => TFCat (not IVOA)

TASKA-D

- On going work on new imager for dynamical sources with AI-based video reconstruction

Components

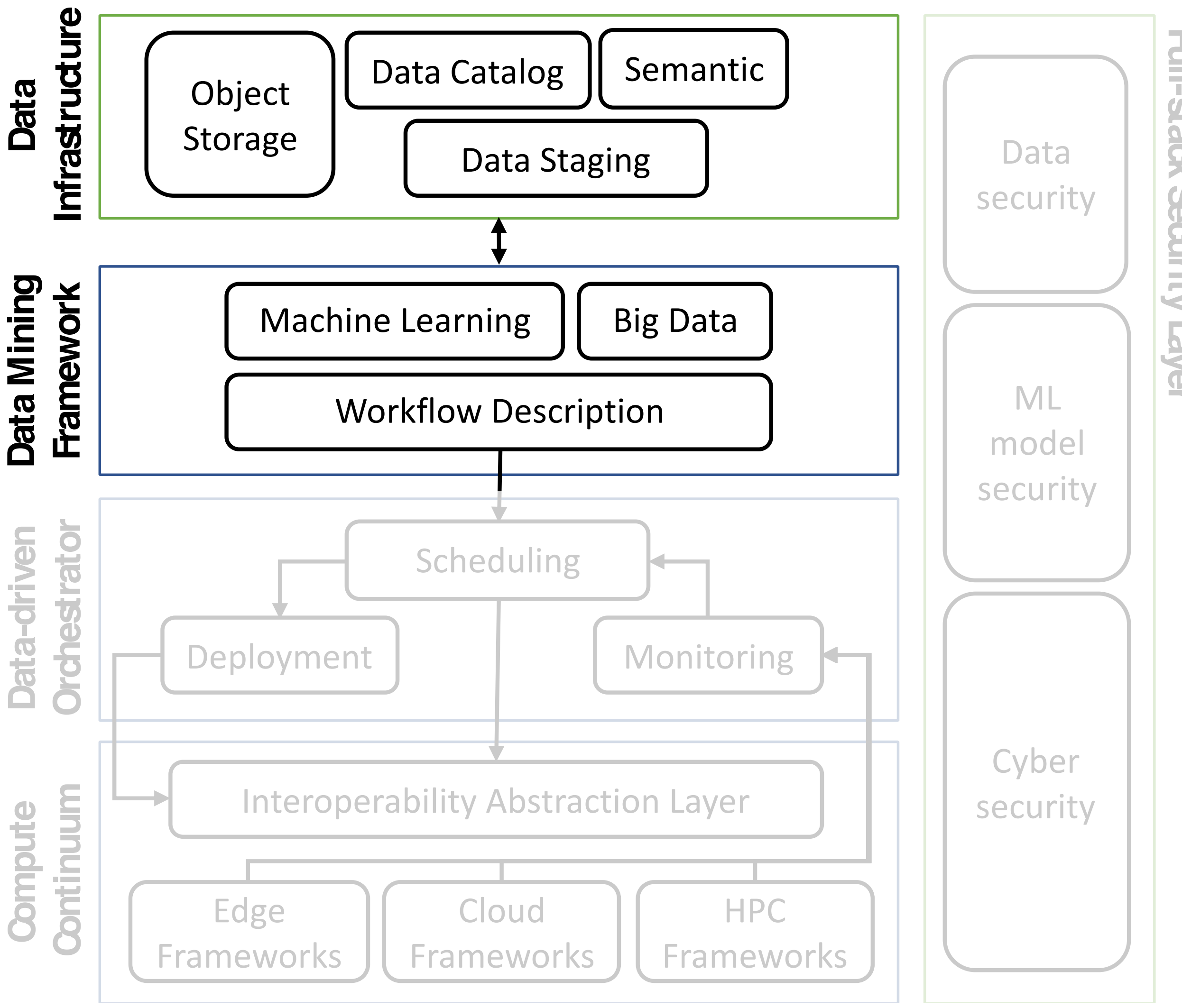
Data Infrastructure

- Object Storage
- Data Staging Engine
- Data Catalog
- Semantic engine

Data Mining Framework

- Complex workflows description
- Serverless approach
- Support to task- and data-based parallelism

Components
S3/InfluxDB
Dataplug
Nuvla
Virtuoso
COMPSs
Lithops
Pytorch
Kserve



Components

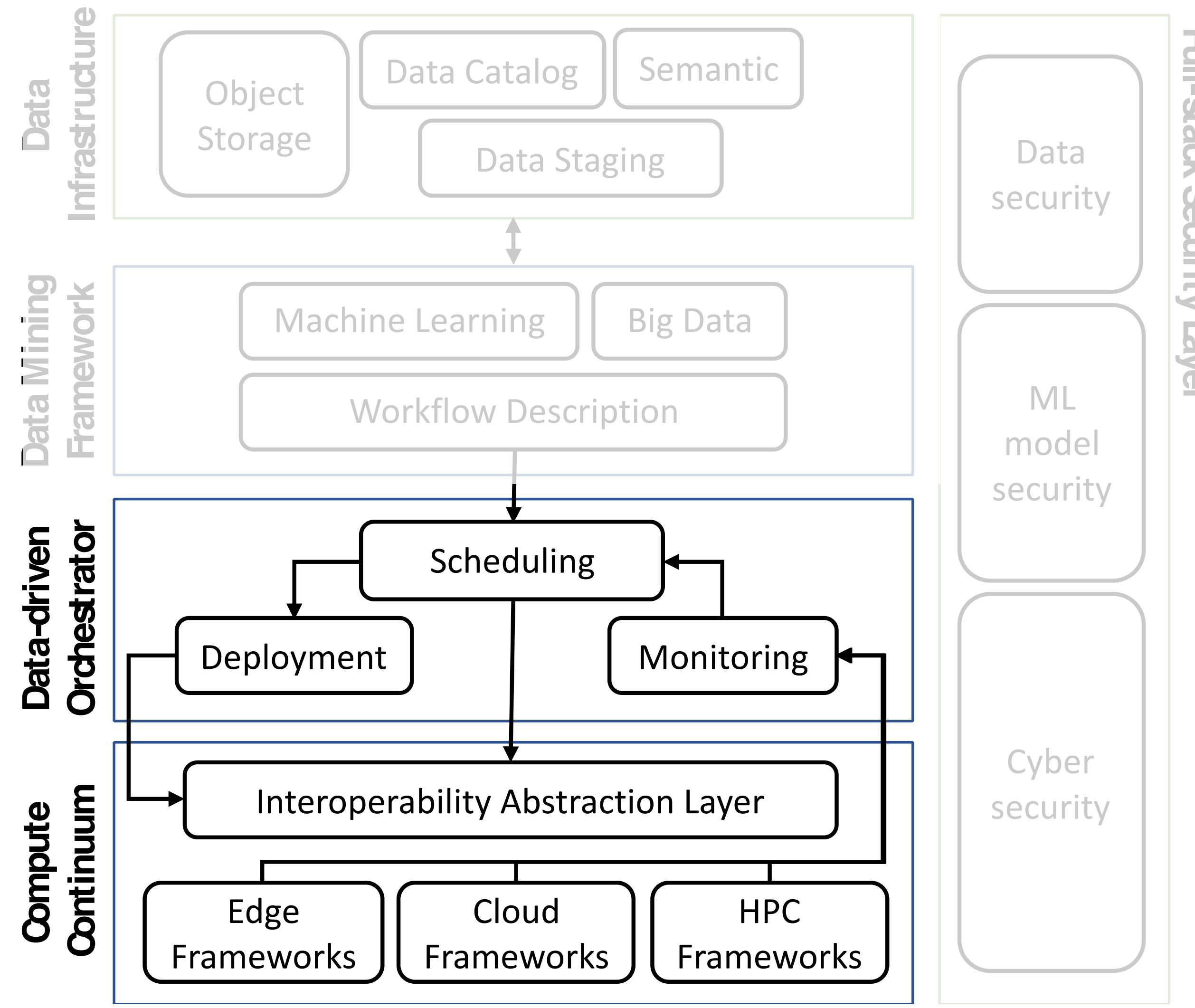
Data-driven Orchestrator

- Select computing resources for workflows based on monitoring

Compute Continuum

- Unified computing abstraction layer based on containers
- Programming paradigms optimized for edge, cloud and HPC

Components
COMPSs
Prometheus
Kubernetes
CUDA
SkyStore

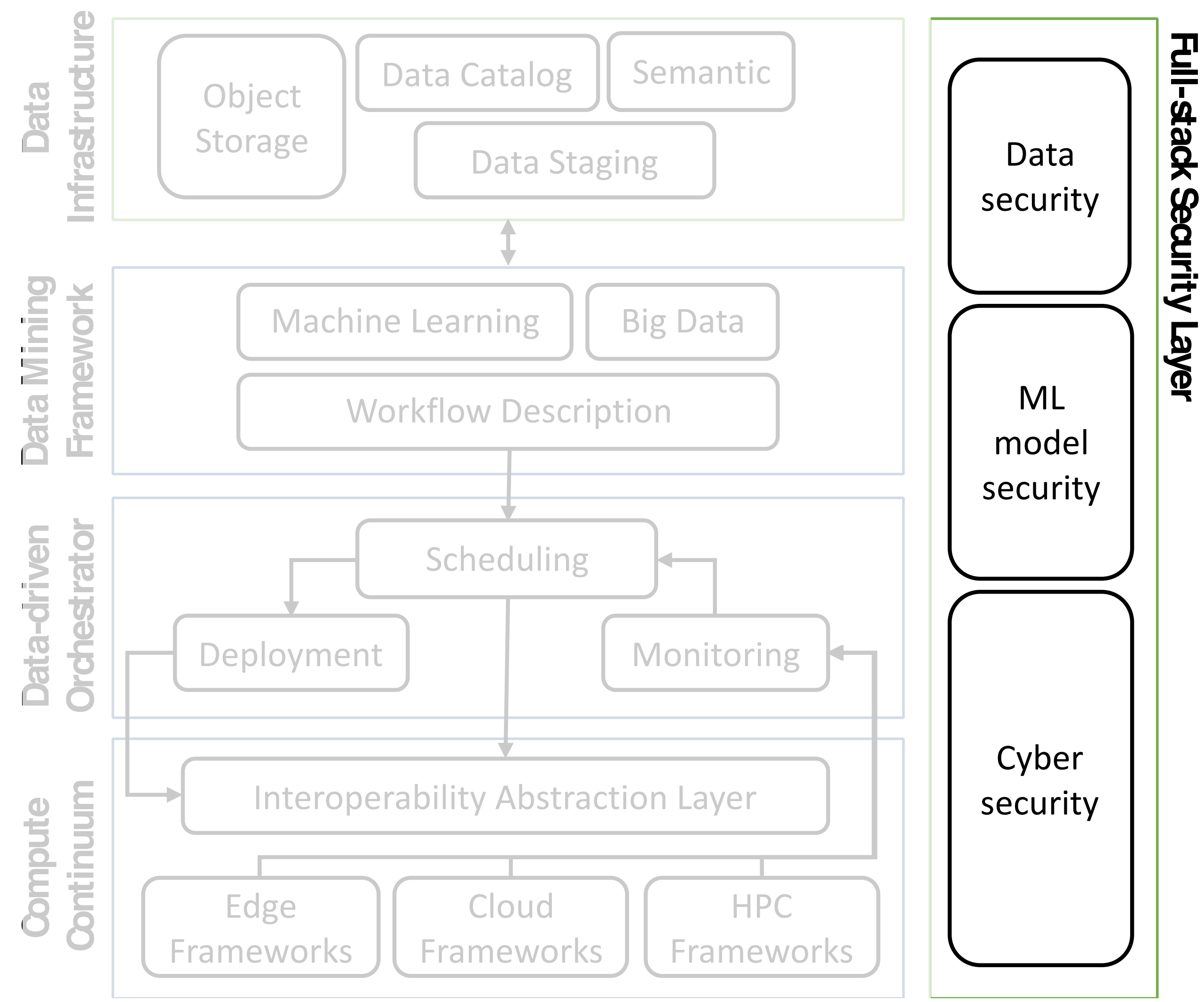


Components

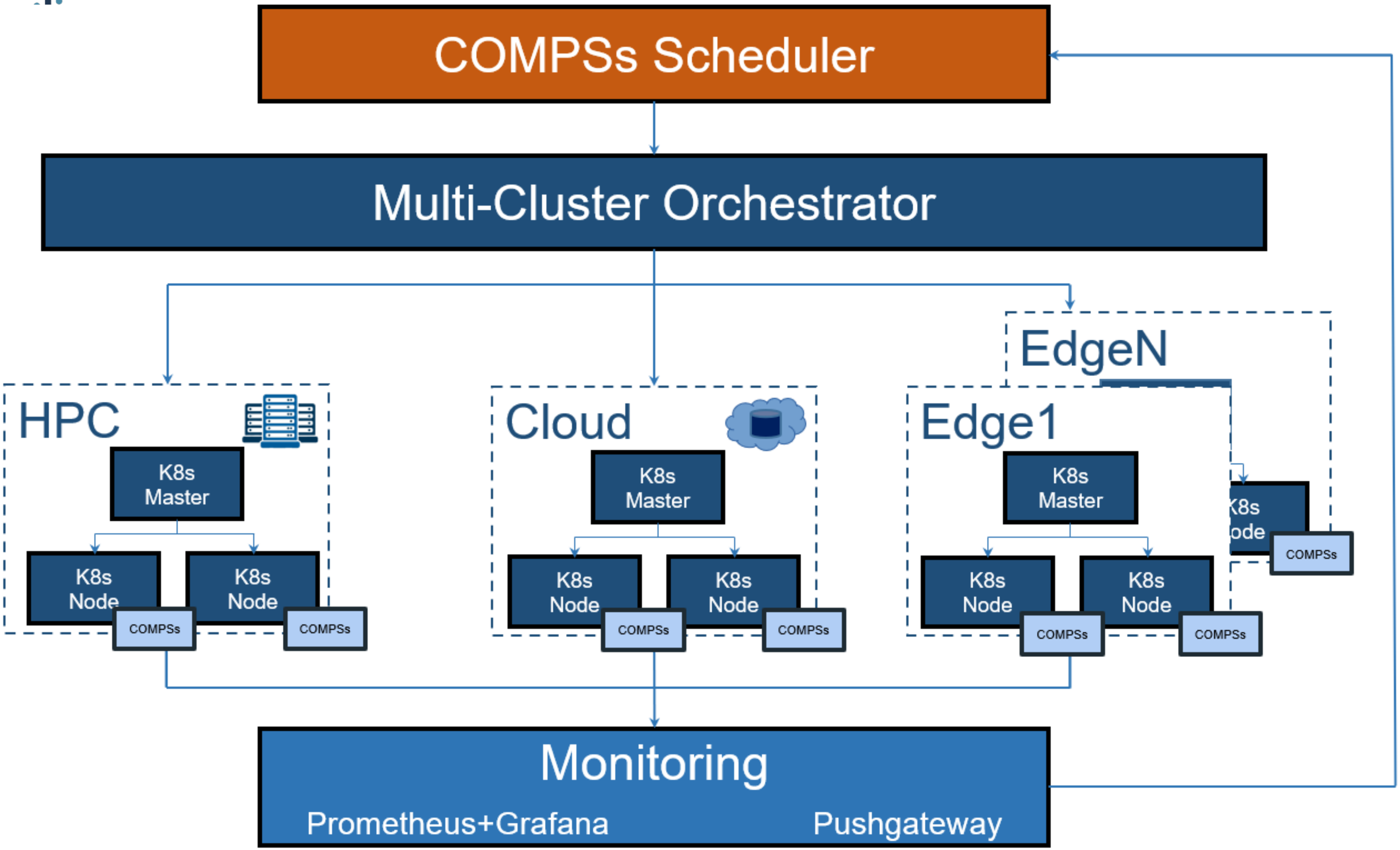
Cybersecurity Capabilities

- Data protection, privacy and confidentiality
- AI models protection
- Authenticity and security for computing nodes
- Trustworthiness and verifiability of routines and libraries

Components
Trivy
MultiParty Computing
Homomorphic Encryption



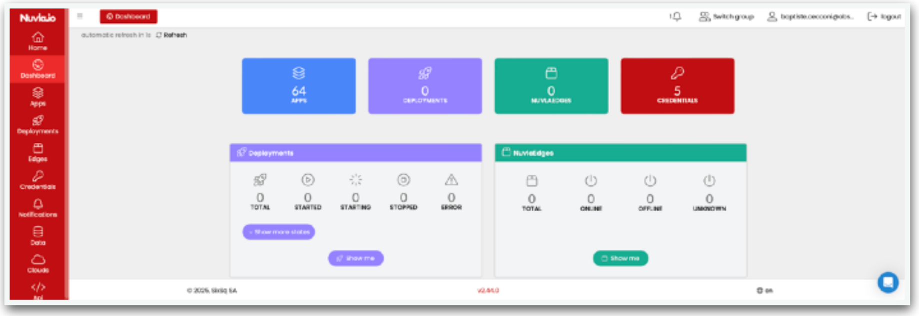
Multi-Cluster orchestrator



Data Catalog: Nuvla data management (DM)

Nuvla.io

https://nuvla.io/ui/



Elements:

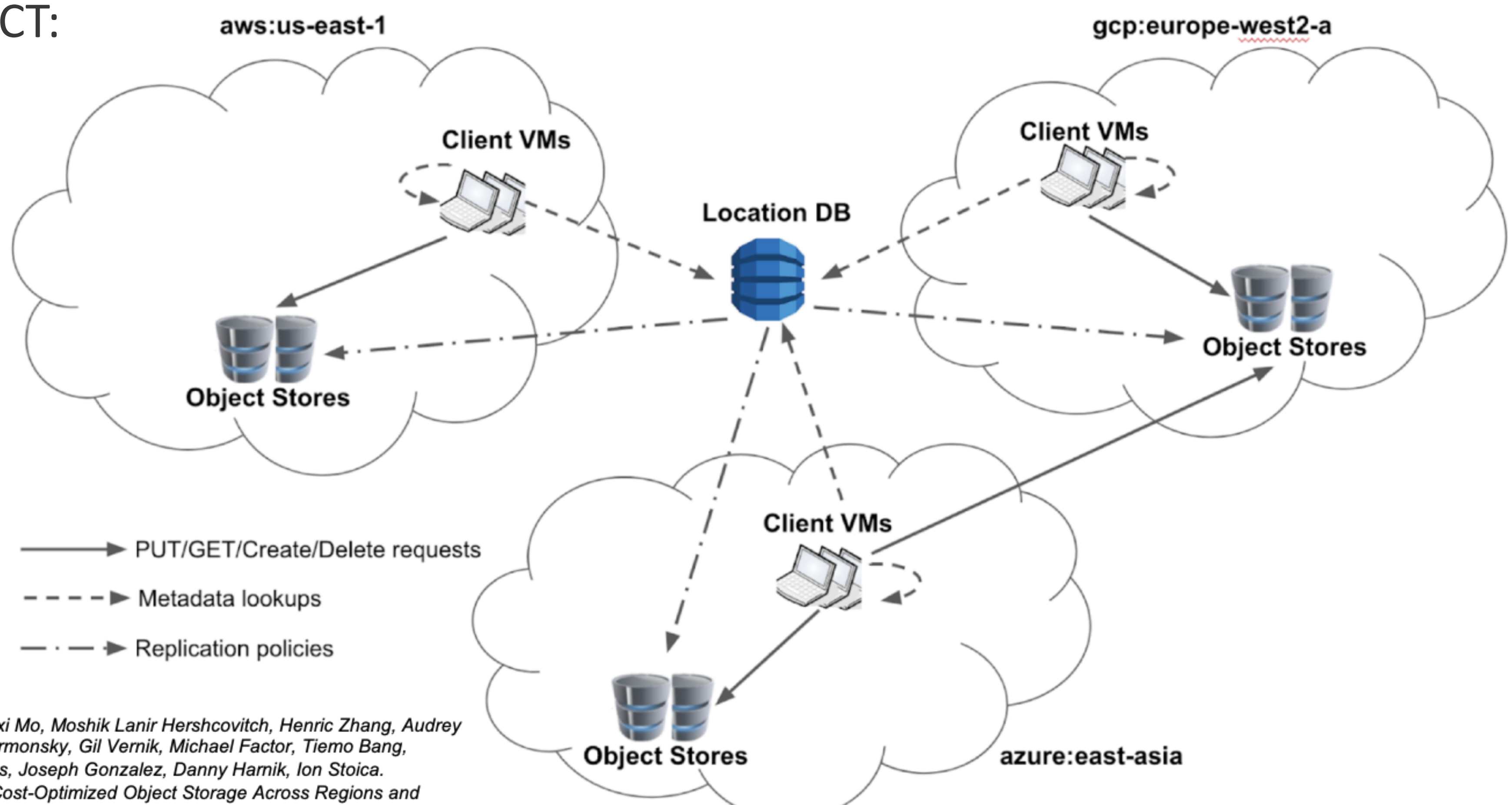
- *S3 infrastructure service*
  - endpoint and credentials
- *data object*
  - reference to S3 object on *S3 infrastructure service*
  - operations: create, obtain S3 pre-signed URL for upload, download, and delete
- *data record*
  - metadata about the *data object*
  - contains reference to the *data object* it describes
- *data set*
  - query against *data records*

Third-party app integration with DM workflow:

- Nuvla API (JSON over HTTPS) for data management
- Nuvla API python-library with data management examples
- Notifications to MQTT on data objects creation

SkyStore

- Joint project UC Berkeley - IBM
- Provides a **virtual global object store** namespace across clouds / premises
- Each client connects to local **S3-proxy**, which is connected to a nearby **S3 cluster** and to a central **location DB server**
- Data replication and consistency controlled via policies
- Specifically, remote objects can be automatically cached using closer/local object storage
- SkyStore work in Phase 2 of EXTRACT:
  - Matured base prototype
  - Joint paper (re-submission)

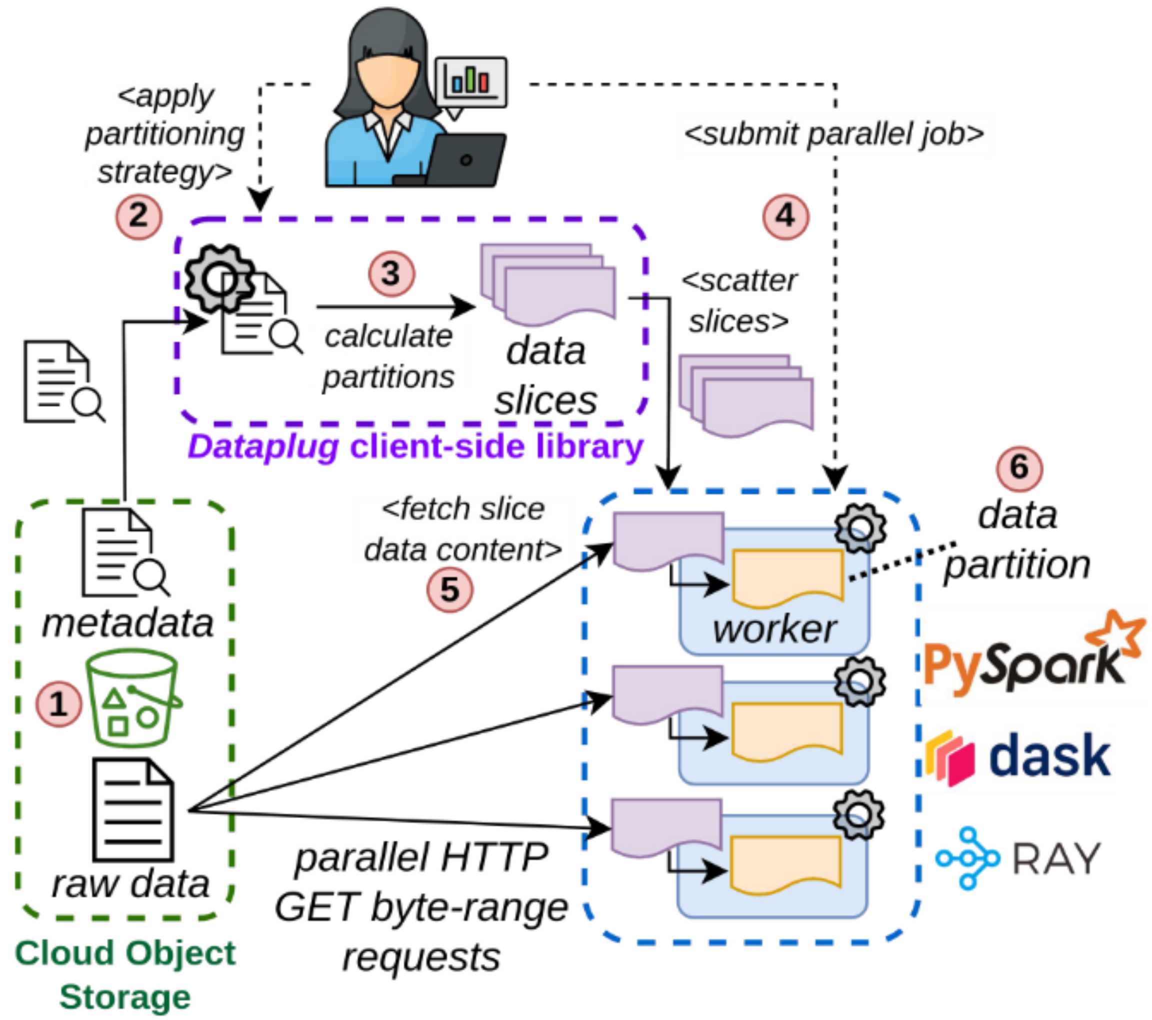


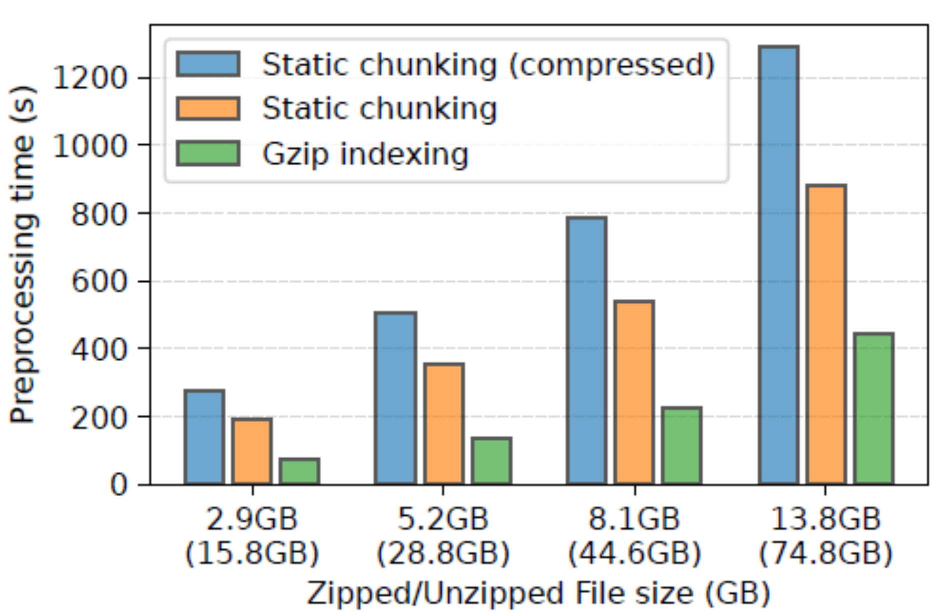
Shu Liu, Xiangxi Mo, Moshik Lanir Hershcovitch, Henric Zhang, Audrey Cheng, Guy Girmonsky, Gil Vernik, Michael Factor, Tiemo Bang, Natacha Crooks, Joseph Gonzalez, Danny Harnik, Ion Stoica. "GlobalStore: Cost-Optimized Object Storage Across Regions and Clouds". Re-submitted

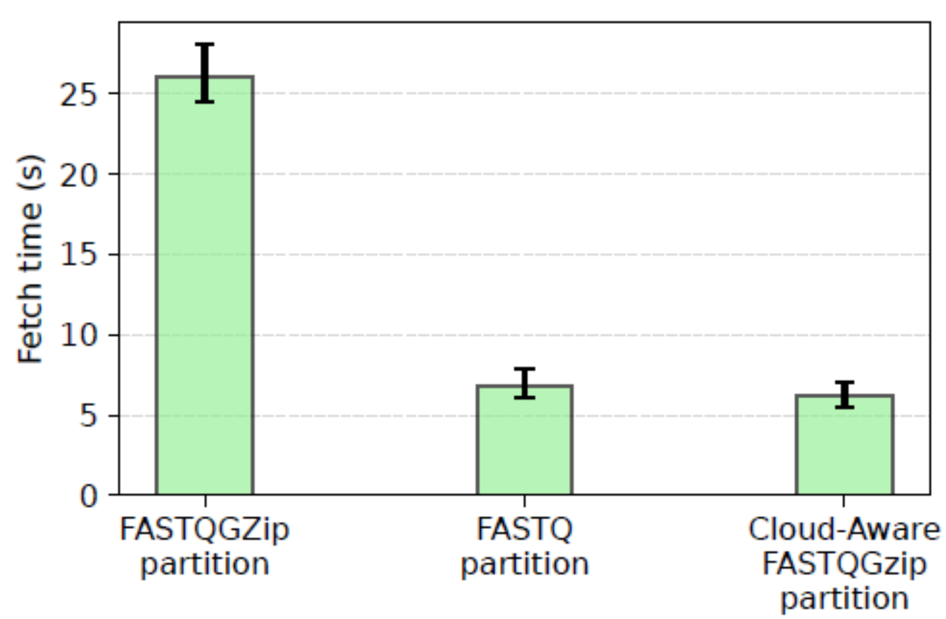
(Slide from Erez Hadad, IBM)

Dataplug dynamic data staging

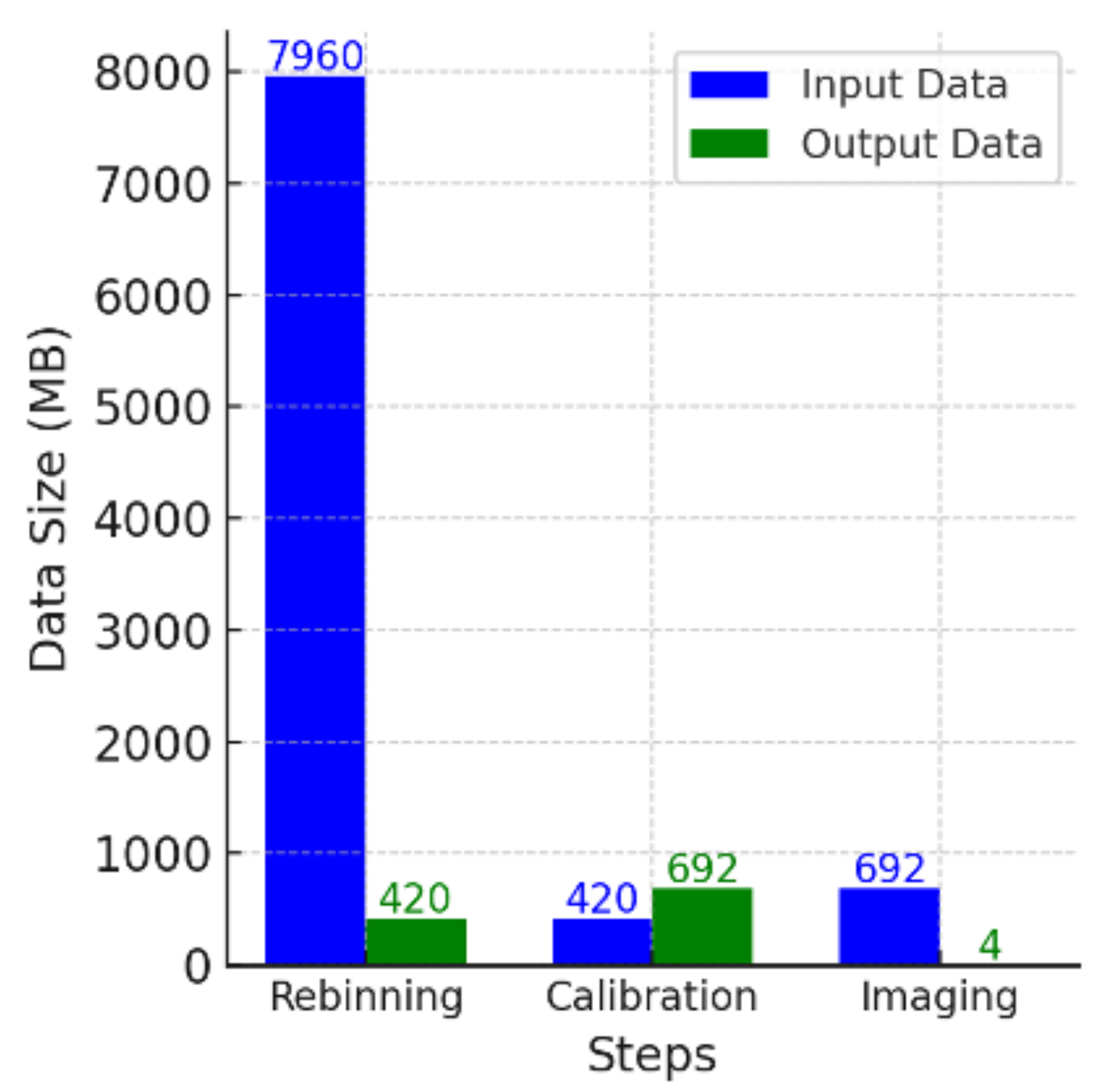
- **Dataplug:** extensible framework that implements on-the-fly data partitioning
- Hide complexities of pre-processing and partitioning unstructured scientific data
- Data-driven and dynamic, efficient parallel access to data
- Generate arbitrary data partitions without modifying existing data
- Extensible to multiple data formats
- **KPI 1.1**: faster partitioning (up to 65.6% less pre-processing time, and 3.7x in fetching partitions) and an important reduction of data transfers in staging



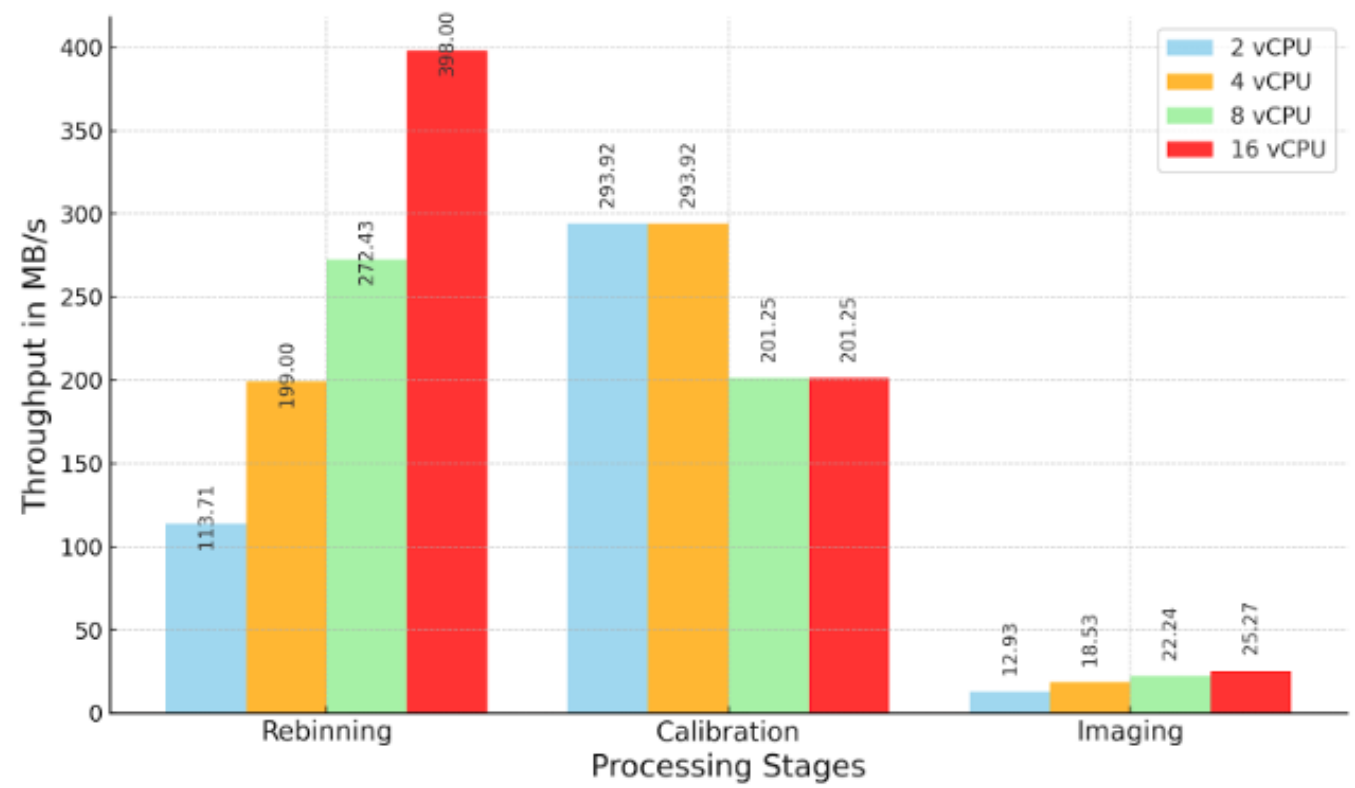




Data throughput analysis



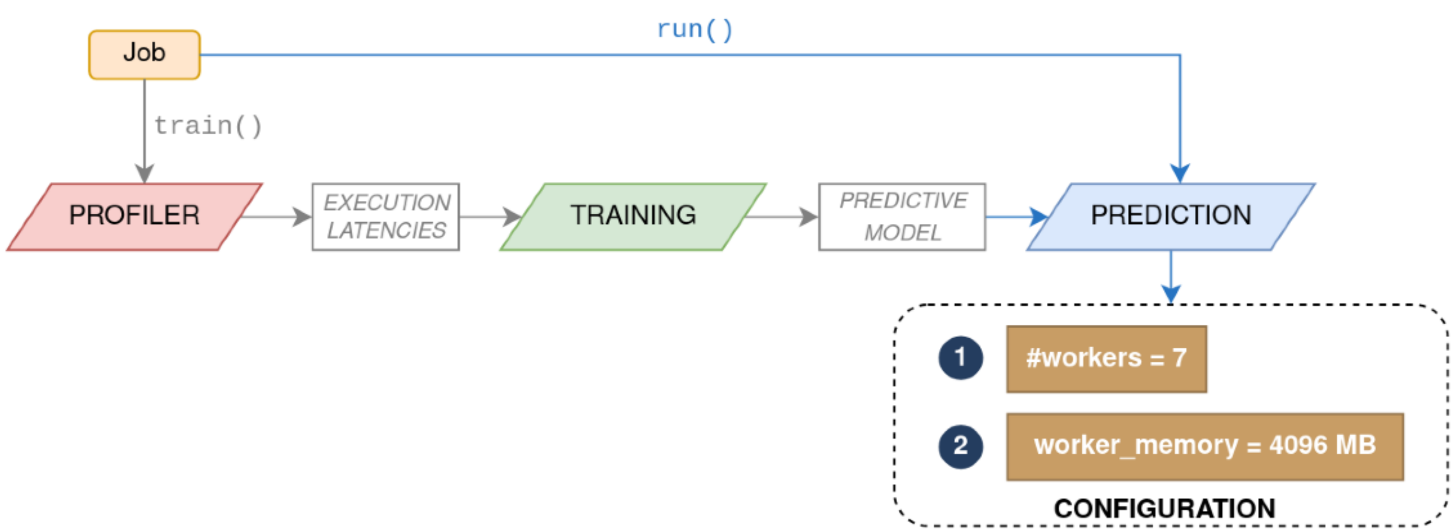
Data volumes per step (input-output

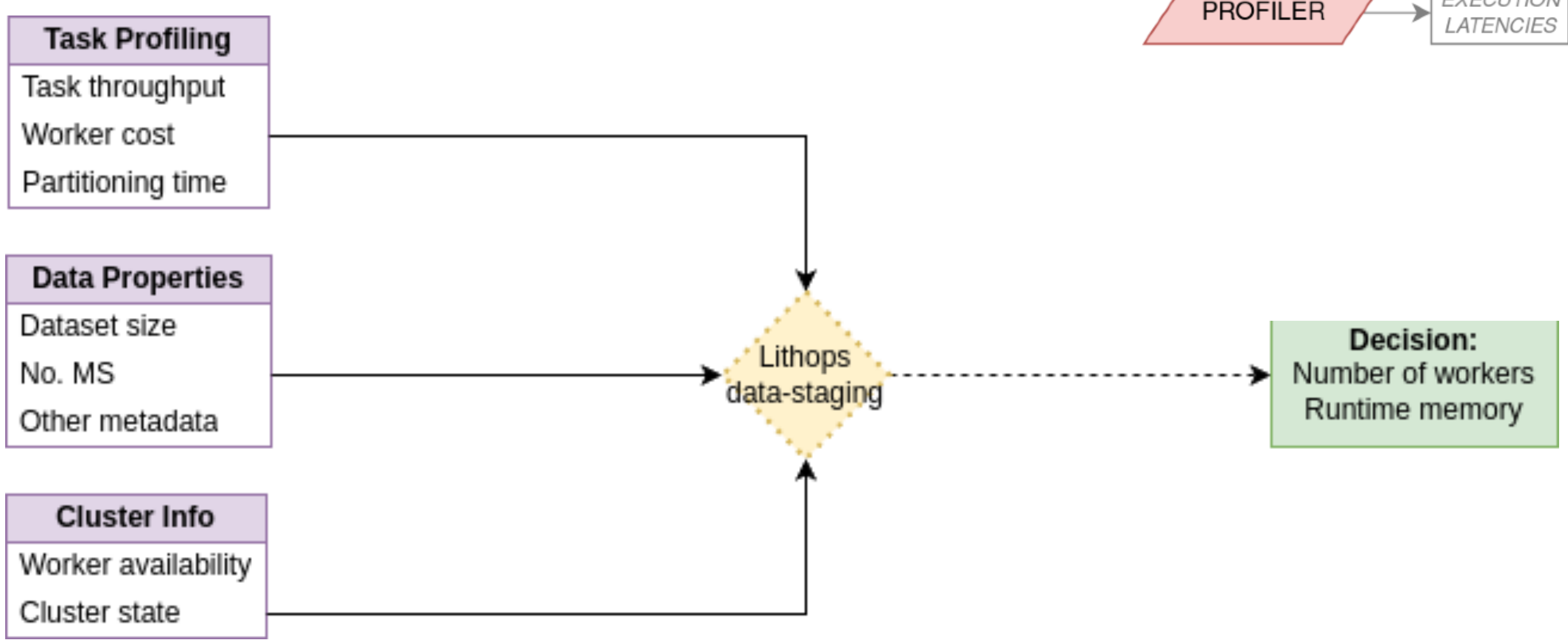


Throughput per step with different VCPU configurations.

Lithops Smart Provisioning Tool

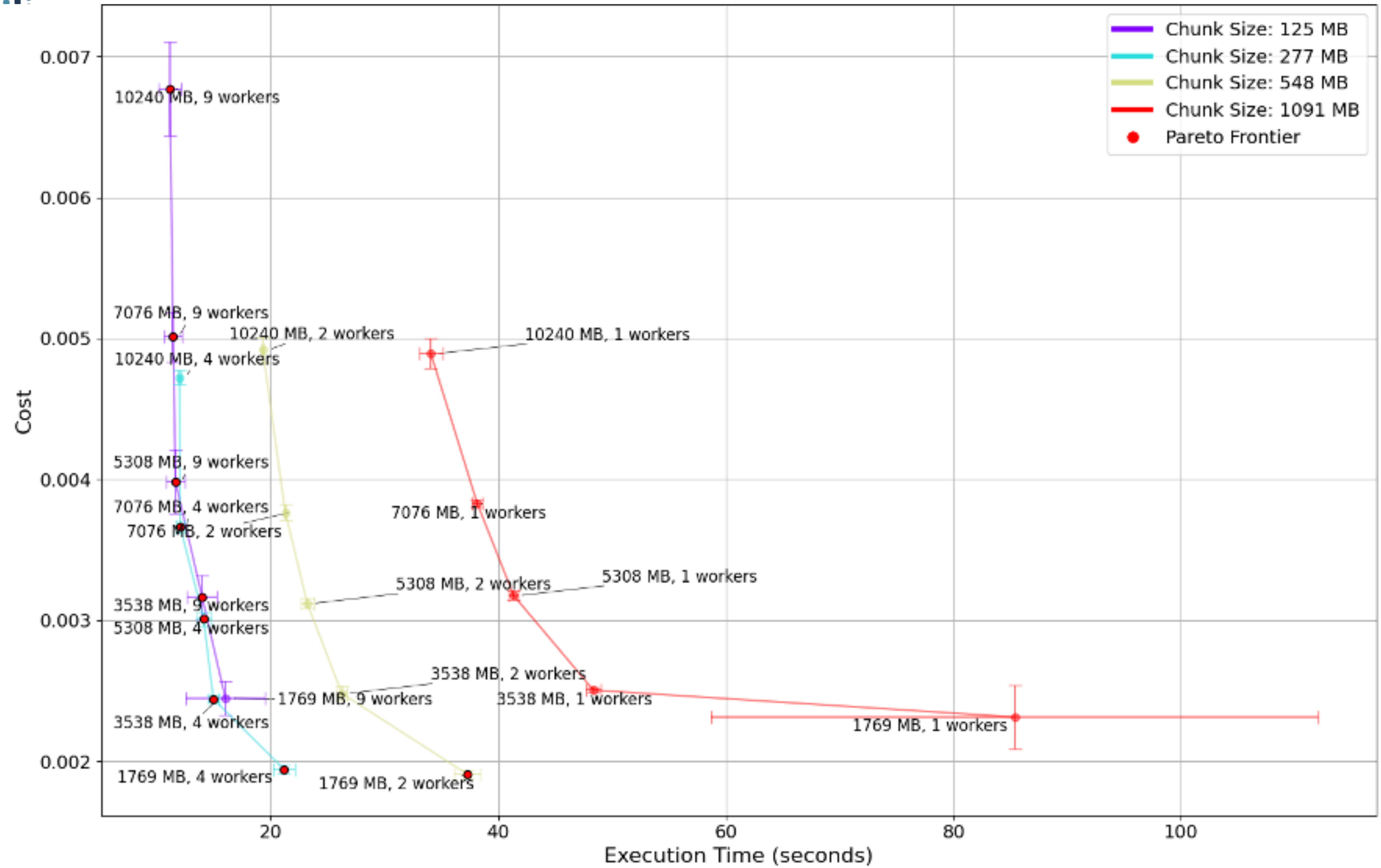
- **Smart provisioning:** new tool for Lithops to calculate the optimal number and size of workers for data staging tasks (WIP)
- Decisions to optimize job completion time and cost based on application performance, data size and transfer speed, and the cost of partitioning
- Working directly with TASKA use case
- **KPI 1.1 and 1.2:** enhancing user experience in developing extreme data processing workflows by abstracting compute resource configuration while optimising performance





Optimal resource allocation studies

1 (small) MS = 1090 MB



The Pareto frontier shows that the most optimal configurations are those that split the input data in small chunks (125 MB or 277 MB) and use multiple (small) workers to run the process