GOBIERNO
DE ESPAÑA

MINISTERIO
DE CIENCIA
E INNOVACIÓN

Ciemat
Centro de Investigaciones
Energéticas, Medioambientales
y Tecnológicas

Comisión Nacional
de Energía Atómica

LAGO INDICA
EOSC, MEIGA y Muongrafía

Hernán Asorey[1,2] for the LAGO Collaboration

BGA, Colombia, 20/Nov/2024

[1] Departamento de Tecnología, CIEMAT, Madrid, Spain.
[2] Departamento Física Médica, CNEA, San Carlos de Bariloche, Argentina

hernanasorey@cnea.gob.ar

outline

1. Some basis about HPC and cloud
2. Data standardization
3. LAGO EOSC Implementation
4. LAGO WCD Application in Meiga
5. Hodoscope Application in MEIGA

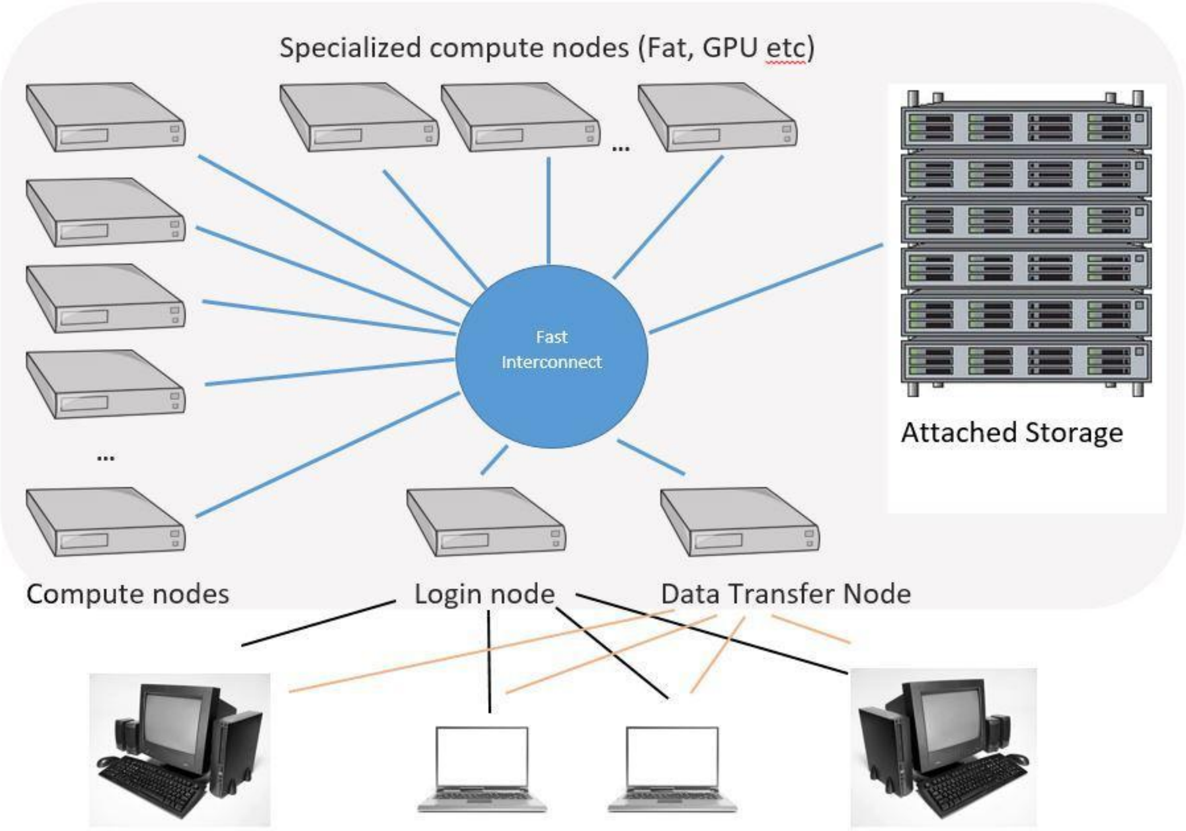
clusters, nodos, cpu, and their v-relatives

Cluster?

Node?

CPU?

Process?



A computer cluster is a collective of individual computers yoked together so they function like a single entity for a variety of tasks

High Performance Computing

core or processor

Processor cores are individual processing units within the central processing unit (CPU) + Core memory cache (kiB). Each core is capable to perform individual tasks

processor or core   <- always is the CPU

Processing power. The more powerful and updated your processor, the faster your computer can complete its tasks.
(multi)-core (c cores) + some fast access memory caches (MiB)

node

*an individual computer with p processors with c cores each one, so (p*c is the total number of cores) + local disk storage (TiB) + local memory (GiB)*

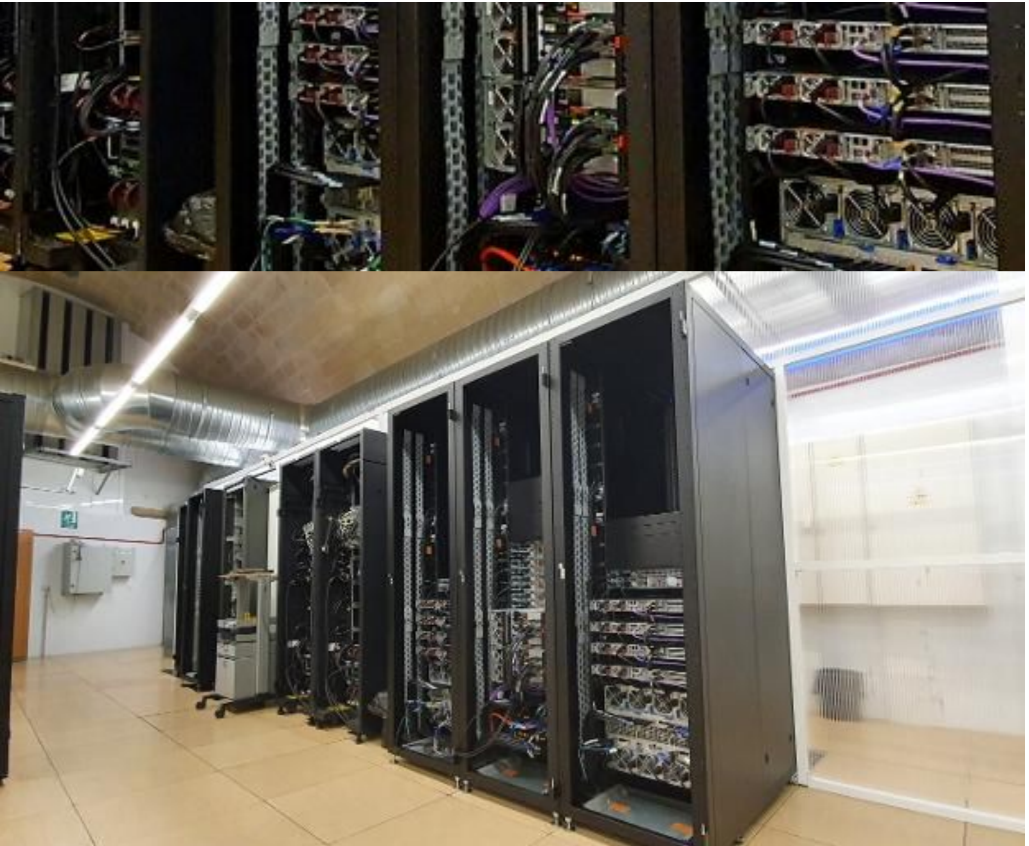
cluster

A computer cluster is a collective of individual nodes (computers) yoked together so they work as a single entity for a variety of tasks. A core could have f front-end nodes + n nodes + distributed memory (TiB) + distributed storage (PiB)

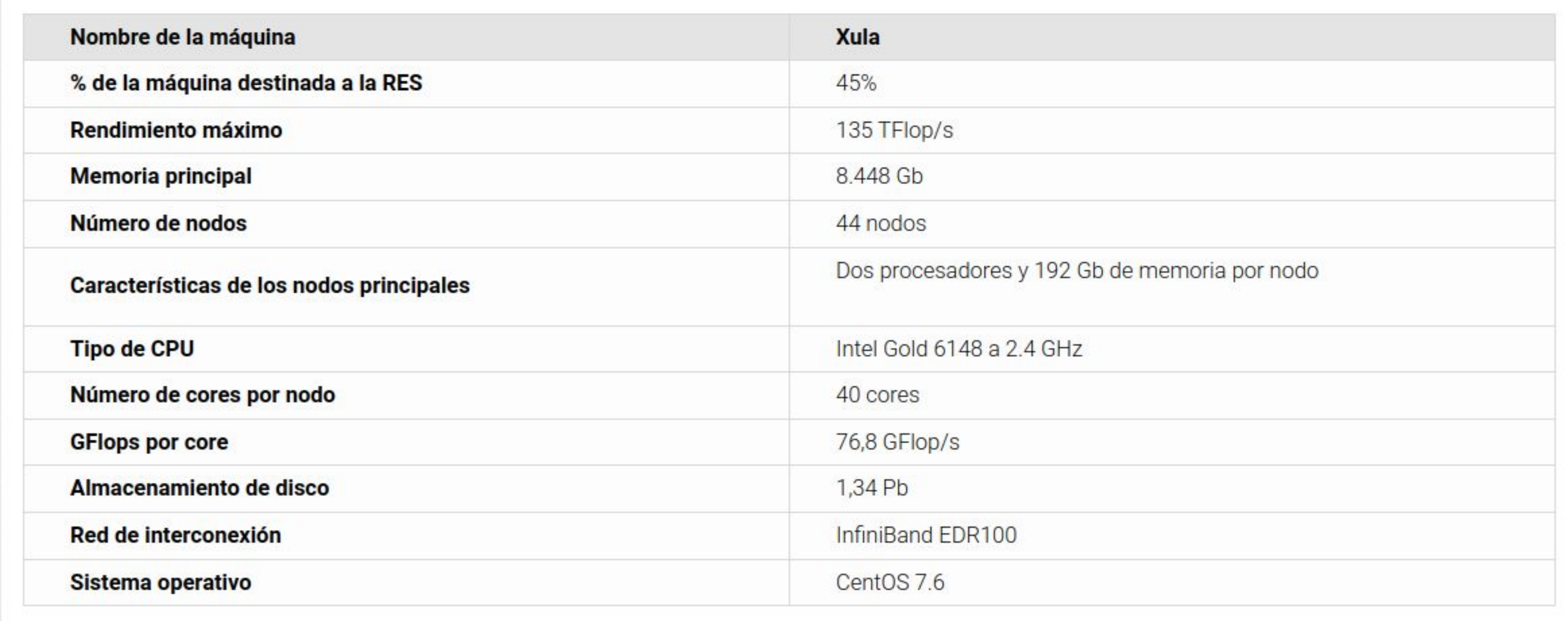
cluster: Marenostrum 4 (BSC)

cluster: Marenostrum 5 (BSC)

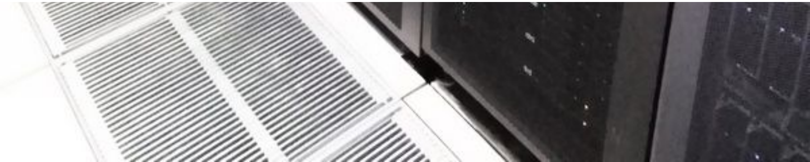
Turgalium, ACME, XULA



Turgalium ACME XULA



Nombre de la máquina	Xula
% de la máquina destinada a la RES	45%
Rendimiento máximo	135 TFlop/s
Memoria principal	8.448 Gb
Número de nodos	44 nodos
Características de los nodos principales	Dos procesadores y 192 Gb de memoria por nodo
Tipo de CPU	Intel Gold 6148 a 2.4 GHz
Número de cores por nodo	40 cores
GFlops por core	76,8 GFlop/s
Almacenamiento de disco	1,34 Pb
Red de interconexión	InfiniBand EDR100
Sistema operativo	CentOS 7.6



Encapsulated pipelines

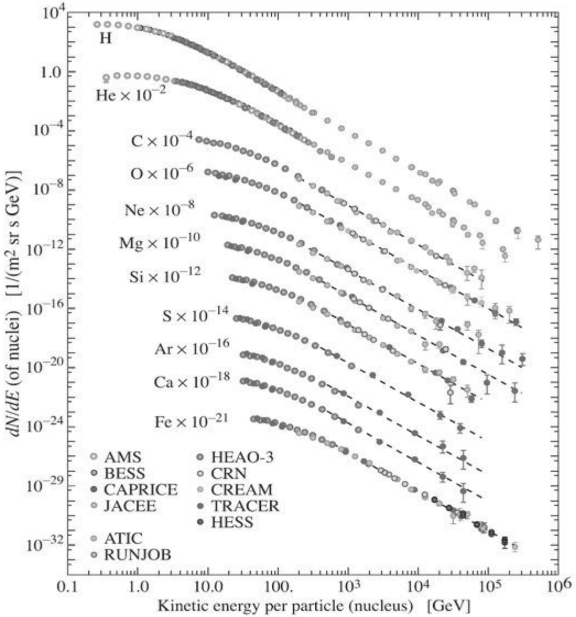
GCR Flux

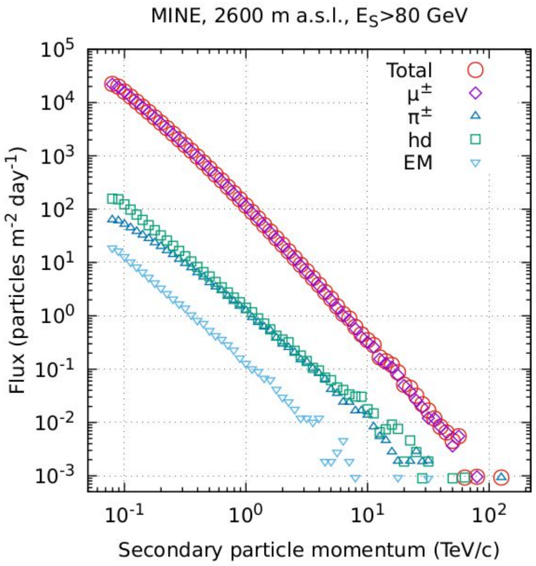
Heliosphere

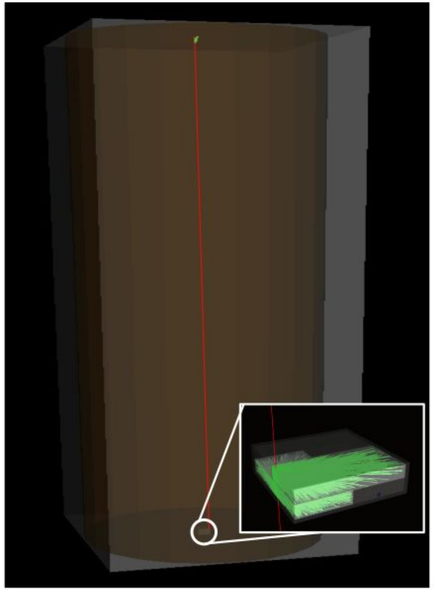
Modulated Flux

Magnetosphere

Primaries (S0) → Atmosphere → **Secondaries (S1)** → Detector → **Signals or Doses (S2)**







AJ Rubio-Montero et al., (ICRC 2021), DOI:10.22323/1.395.0261, H. Asorey et al, COLAGE2021

Encapsulated pipelines

GCR Flux

Heliosphere

Modulated Flux

Magnetosphere

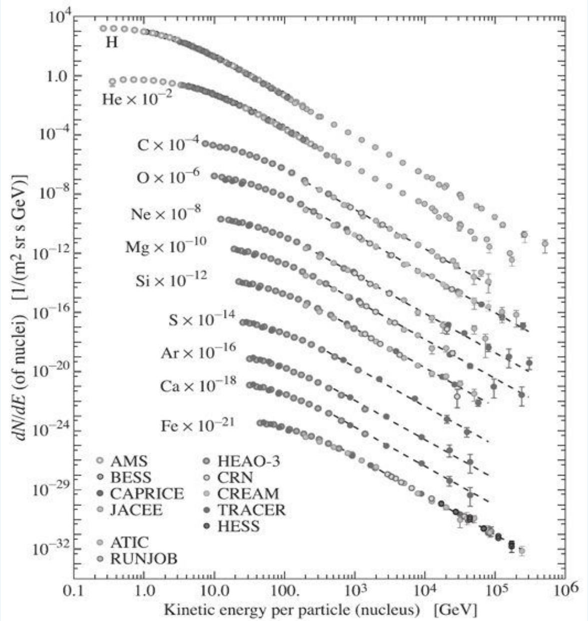
Primaries (S0)

Atmosphere

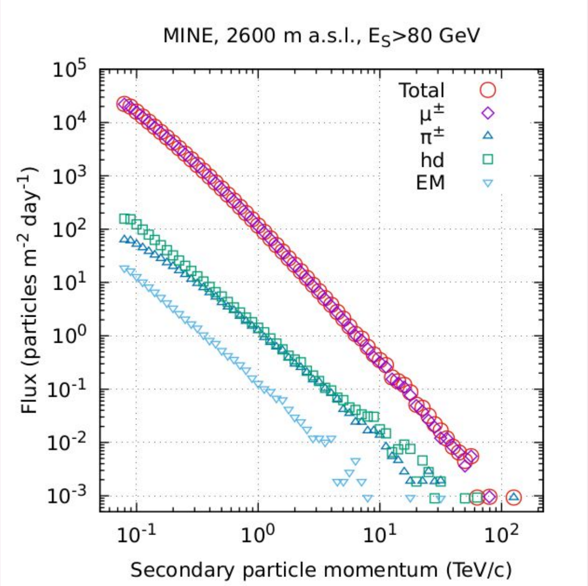
Secondaries (S1)

Detector

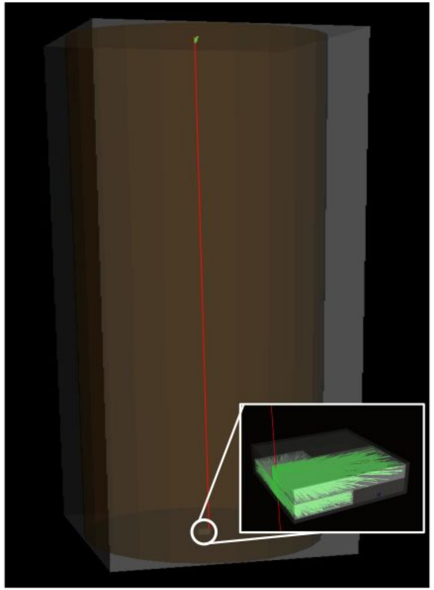
Signals or Doses (S2)



onedataSim-S0



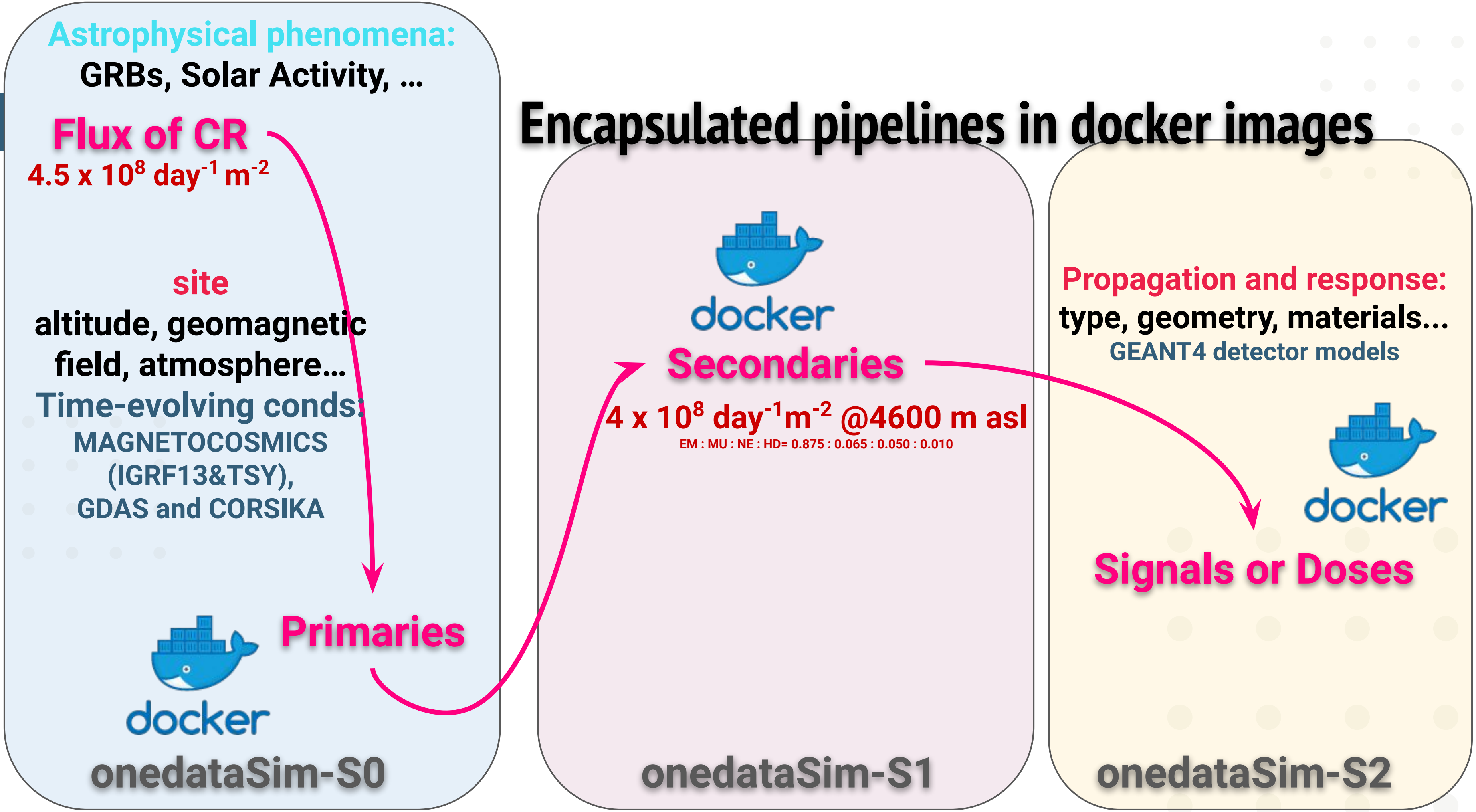
onedataSim-S1



onedataSim-S2

AJ Rubio-Montero et al., (ICRC 2021), DOI:10.22323/1.395.0261, H. Asorey et al, COLAGE2021

Encapsulated pipelines in docker images



Astrophysical phenomena:
GRBs, Solar Activity, …

Flux of CR
4.5 x 10^8 day^{-1} m^{-2}

site
altitude, geomagnetic field, atmosphere…
Time-evolving conds:
MAGNETOCOSMICS (IGRF13&TSY),
GDAS and CORSIKA

Primaries

onedataSim-S0

Secondaries
4 x 10^8 $day^{-1}m^{-2}$ @4600 m asl
EM : MU : NE : HD= 0.875 : 0.065 : 0.050 : 0.010

onedataSim-S1

Propagation and response:
type, geometry, materials...
GEANT4 detector models

Signals or Doses

onedataSim-S2

C. Sarmiento-Cano et al., (ICRC2019), DOI:10.22323/1.358.0412; Hernán Asorey et al., Journal of European Physics G, submitted

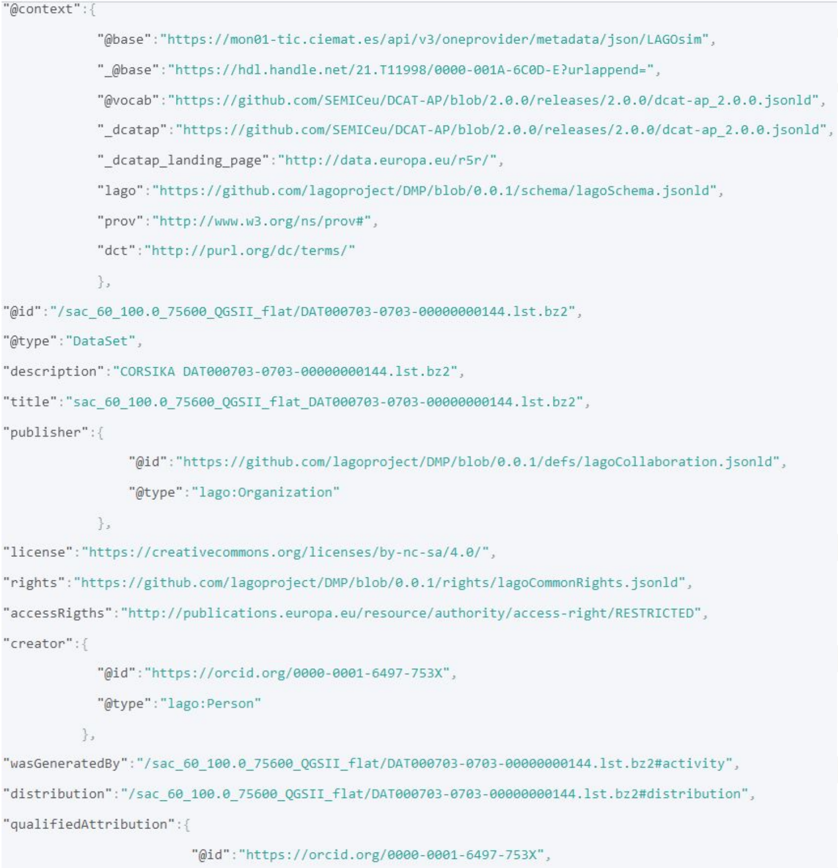
Schemas for data and metadata

Metadata:

- Language syntax: **JSON-LD 1.1** (W3C). Enabling linked-data.
- Main vocabulary: **DCAT-AP2** (EC), a profile of DCAT2 (W3C), for government, research centers & funded projects.
- The **LAGO vocabulary: re-profile of DCAT-AP2**, extending classes & properties needed (geomagnetic locations, expected atmospheres, software...).

Data conventions compatible with DCAT-AP2:

- A super-Catalog comprises whole repository
- A Catalog is any complete simulation (S0, S1 or S2)
- A Dataset is any file belonging one simulation



```
"@context":{
            "@base":"https://mon01-tic.ciemat.es/api/v3/oneprovider/metadata/json/LAGOsim",
            "_@base":"https://hdl.handle.net/21.T11998/0000-001A-6C0D-E?urlappend=",
            "@vocab":"https://github.com/SEMICeu/DCAT-AP/blob/2.0.0/releases/2.0.0/dcat-ap_2.0.0.jsonld",
            "_dcatap":"https://github.com/SEMICeu/DCAT-AP/blob/2.0.0/releases/2.0.0/dcat-ap_2.0.0.jsonld",
            "_dcatap_landing_page":"http://data.europa.eu/r5r/",
            "lago":"https://github.com/lagoproject/DMP/blob/0.0.1/schema/lagoSchema.jsonld",
            "prov":"http://www.w3.org/ns/prov#",
            "dct":"http://purl.org/dc/terms/"
            },
"@id":"/sac_60_100.0_75600_QGSII_flat/DAT000703-0703-00000000144.lst.bz2",
"@type":"DataSet",
"description":"CORSIKA DAT000703-0703-00000000144.lst.bz2",
"title":"sac_60_100.0_75600_QGSII_flat_DAT000703-0703-00000000144.lst.bz2",
"publisher":{
                "@id":"https://github.com/lagoproject/DMP/blob/0.0.1/defs/lagoCollaboration.jsonld",
                "@type":"lago:Organization"
            },
"license":"https://creativecommons.org/licenses/by-nc-sa/4.0/",
"rights":"https://github.com/lagoproject/DMP/blob/0.0.1/rights/lagoCommonRights.jsonld",
"accessRigths":"http://publications.europa.eu/resource/authority/access-right/RESTRICTED",
"creator":{
            "@id":"https://orcid.org/0000-0001-6497-753X",
            "@type":"lago:Person"
          },
"wasGeneratedBy":"/sac_60_100.0_75600_QGSII_flat/DAT000703-0703-00000000144.lst.bz2#activity",
"distribution":"/sac_60_100.0_75600_QGSII_flat/DAT000703-0703-00000000144.lst.bz2#distribution",
"qualifiedAttribution":{
                        "@id":"https://orcid.org/0000-0001-6497-753X",
```

Metadata of a S0 dataset (CORSIKA output)

AJ Rubio-Montero et al. (2021), WSC'21

Schemas for data and metadata

Metadata:

- Language syntax: **JSON-LD 1.1** (W3C). Enabling linked-data.

- Main vocabulary: **DCAT-AP2** (EC), a profile of DCAT2 (W3C), for government, research centers & funded projects.

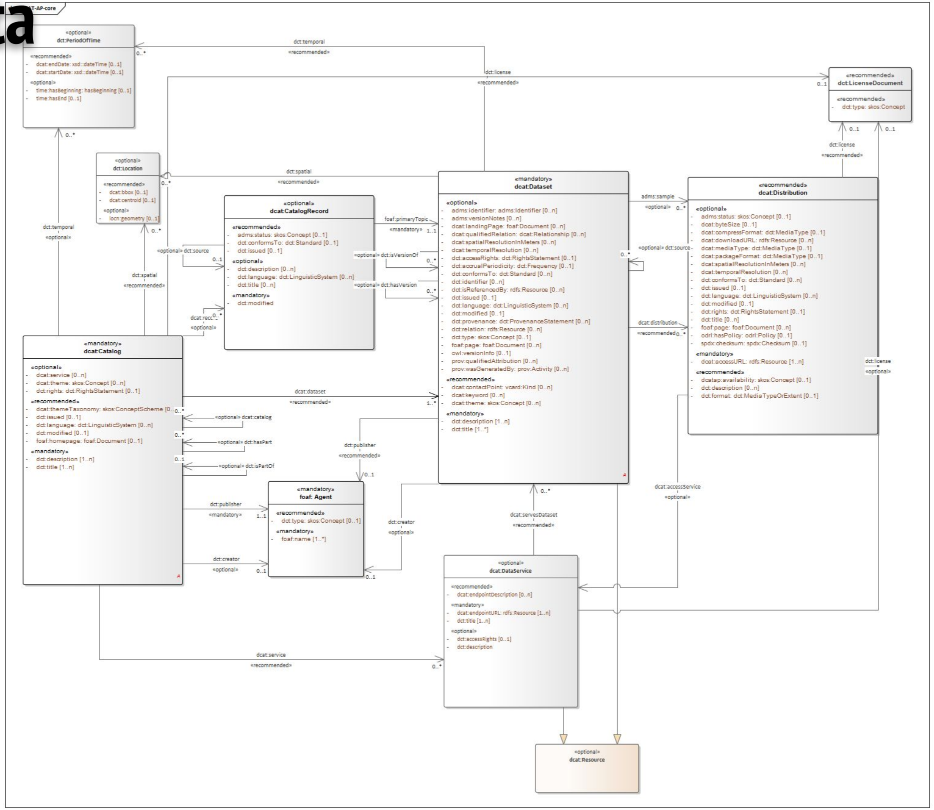
- The **LAGO vocabulary: re-profile of DCAT-AP2**, extending classes & properties needed (geomagnetic locations, expected atmospheres, software...).

Data conventions compatible with DCAT-AP2:

- A super-Catalog comprises whole repository

- A Catalog is any complete simulation (S0, S1 or S2)

- A Dataset is any file belonging one simulation



AJ Rubio-Montero et al. (2021), WSC'21

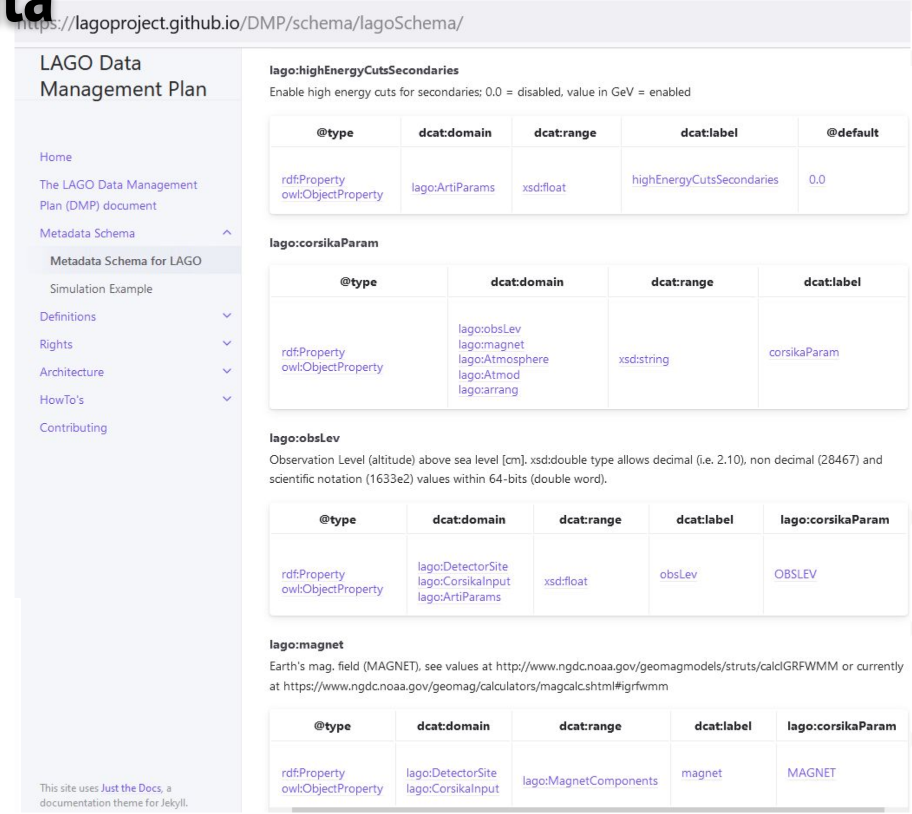
Schemas for data and metadata

Metadata:

- Language syntax: **JSON-LD 1.1** (W3C). Enabling linked-data.
- Main vocabulary: **DCAT-AP2** (EC), a profile of DCAT2 (W3C), for government, research centers & funded projects.
- The **LAGO vocabulary: re-profile of DCAT-AP2**, extending classes & properties needed (geomagnetic locations, expected atmospheres, software...).

Data conventions compatible with DCAT-AP2:

- A super-Catalog comprises whole repository
- A Catalog is any complete simulation (S0, S1 or S2)
- A Dataset is any file belonging one simulation



This site uses Just the Docs, a documentation theme for Jekyll.

lago:highEnergyCutsSecondaries

Enable high energy cuts for secondaries; 0.0 = disabled, value in GeV = enabled

@type	dcat:domain	dcat:range	dcat:label	@default
rdf:Property owl:ObjectProperty	lago:ArtiParams	xsd:float	highEnergyCutsSecondaries	0.0

lago:corsikaParam

@type	dcat:domain	dcat:range	dcat:label
rdf:Property owl:ObjectProperty	lago:obsLev lago:magnet lago:Atmosphere lago:Atmod lago:arrang	xsd:string	corsikaParam

lago:obsLev

Observation Level (altitude) above sea level [cm]. xsd:double type allows decimal (i.e. 2.10), non decimal (28467) and scientific notation (1633e2) values within 64-bits (double word).

@type	dcat:domain	dcat:range	dcat:label	lago:corsikaParam
rdf:Property owl:ObjectProperty	lago:DetectorSite lago:CorsikaInput lago:ArtiParams	xsd:float	obsLev	OBSLEV

lago:magnet

Earth's mag. field (MAGNET), see values at http://www.ngdc.noaa.gov/geomagmodels/struts/calcIGRFWMM or currently at https://www.ngdc.noaa.gov/geomag/calculators/magcalc.shtml#igrfwmm

@type	dcat:domain	dcat:range	dcat:label	lago:corsikaParam
rdf:Property owl:ObjectProperty	lago:DetectorSite lago:CorsikaInput	lago:MagnetComponents	magnet	MAGNET

Definitions in the LAGO DMP

AJ Rubio-Montero et al. (2021), WSC'21

Schemas for data and metadata

Metadata:

- Language syntax: **JSON-LD 1.1** (W3C). Enabling linked-data.
- Main vocabulary: **DCAT-AP2** (EC), a profile of DCAT2 (W3C), for government, research centers & funded projects.
- The **LAGO vocabulary: re-profile of DCAT-AP2**, extending classes & properties needed (geomagnetic locations, expected atmospheres, software...).

Data conventions compatible with DCAT-AP2:

- A super-Catalog comprises whole repository
- A Catalog is any complete simulation (S0, S1 or S2)
- A Dataset is any file belonging one simulation

AJ Rubio-Montero et al. (2021), WSC'21

Every Catalog (simulation) has a Handle.net PiD

View provided by ceta-ciemat-01

LAGOsim

S0_and_5184000_645000.0_77402_QGSII_volu_HEcuts_defaults

S0_and_60_77402_QGSII_flat_defaults

S0_and_7776000_400000.0_77402_QGSII_volu_HEcuts_defaults

S0_and_86400_470000.0_77402_QGSII_volu_defaults

S0_and_86400_520000.0_77402_QGSII_volu_defaults

S0_and_86400_570000.0_77402_QGSII_volu_defaults

S0_and_86400_620000.0_77402_QGSII_volu_defaults

S0_and_86400_77402_QGSII_flat_defaults

Partial content of LAGOsim repo

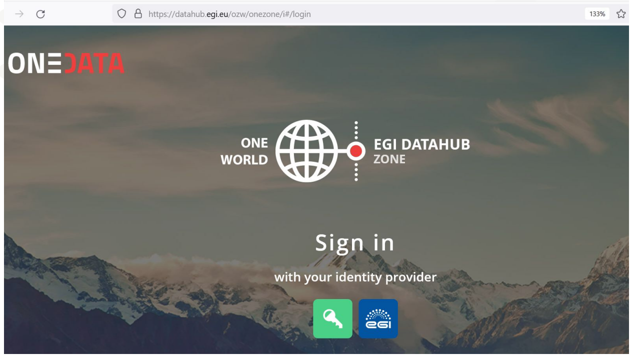
Where are the meta & data stored?

ONEDATA is an object-based distributed FS focused on cloud and HPC:

- offers unified data access and roles across globally distributed environments;
- manages metadata and expose them to harvesters.

EGI/EOSC supports the “DataHub” instance that:

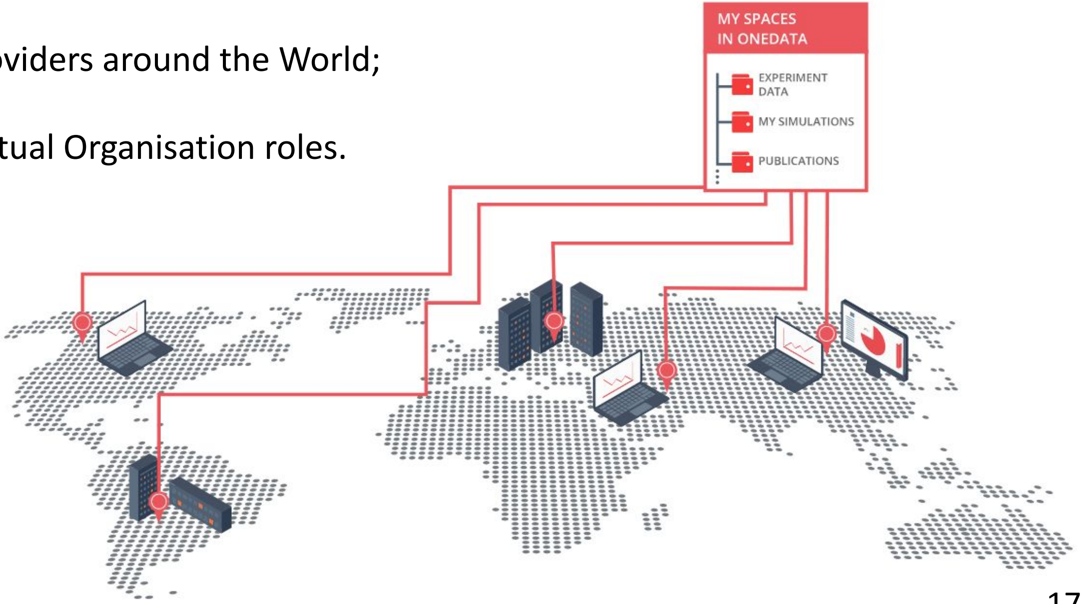
- allows extending the capacity by adding providers around the World;
- asks for (Handle.net) PiDs;
- enable pan-European IdPs and translate Virtual Organisation roles.



Sign in

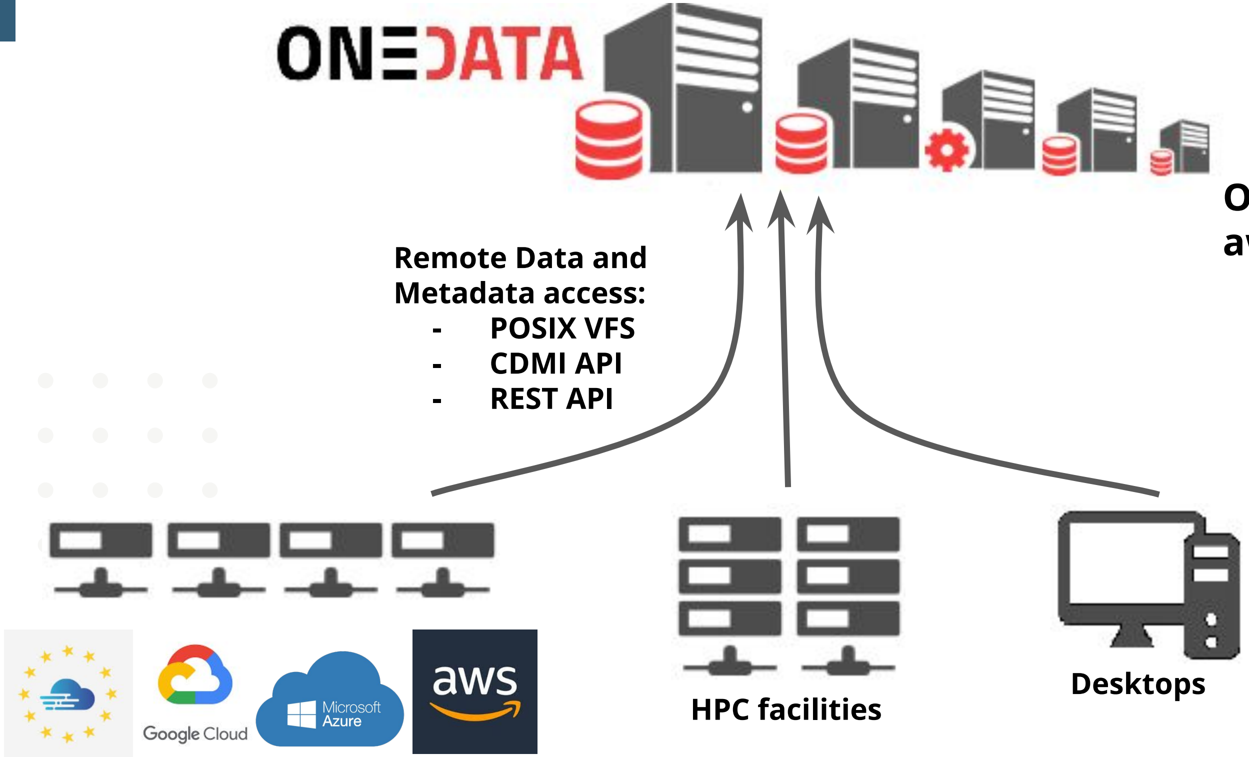
with your identity provider

https://datahub.egi.eu



AJ Rubio-Montero et al. (2021), WSC’21

How are the meta & data stored and retrieved?



Remote Data and Metadata access:
- **POSIX VFS**
- **CDMI API**
- **REST API**

Nodes on public clouds

HPC facilities

Desktops

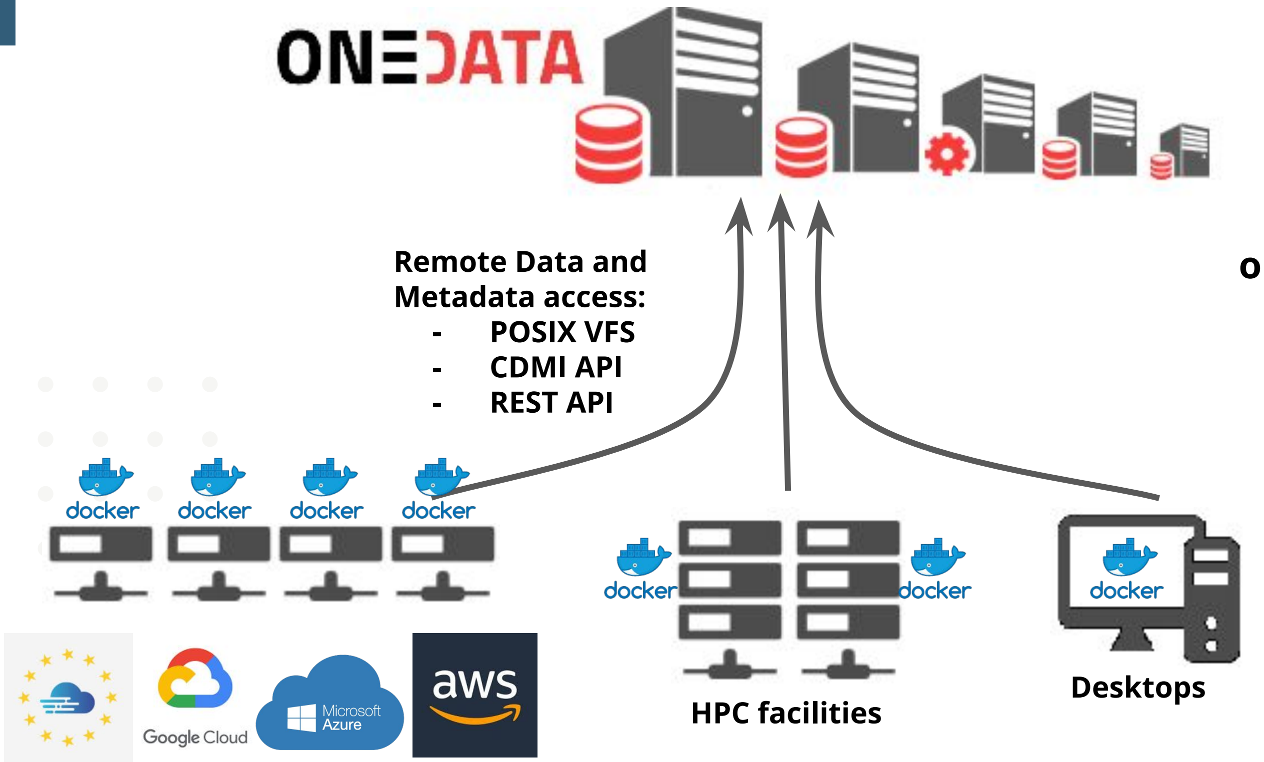
OneData providers can be far away from the computing nodes

Simulations are:
- CPU & disk (scratch) intensives.
- Multi-threaded

We are only interested in
- getting remote inputs
- storing final results
- pipelining the copies
- manage metadata

AJ Rubio-Montero et al. (2021), WSC'21

How are the meta & data stored and retrieved?



Remote Data and Metadata access:
- **POSIX VFS**
- **CDMI API**
- **REST API**

Nodes on public clouds

HPC facilities

Desktops

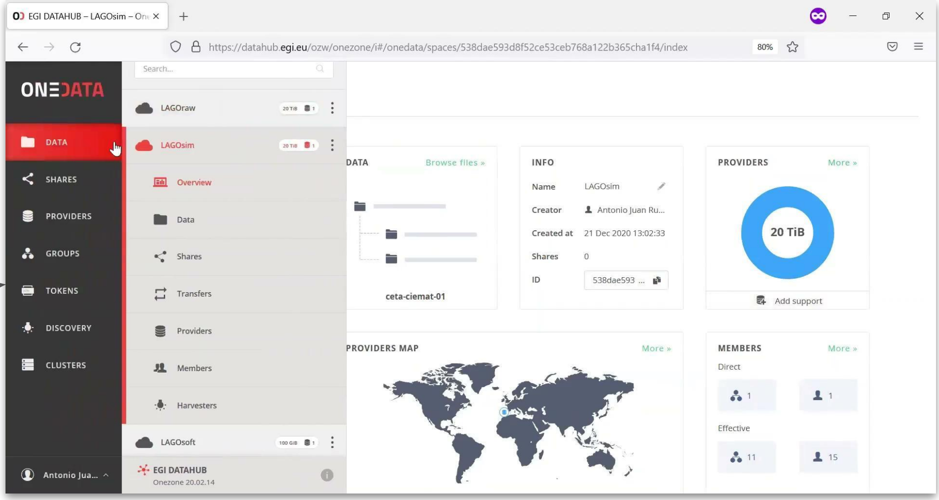
onedataSim images:
- guarantee the compatibility across platforms
- packet the communication with the OneData APIs

AJ Rubio-Montero et al. (2021), WSC'21

How are the meta & data stored and retrieved?

LAGO members can:

- explore whole repository with a Web browser;
- download non-owned data after embargo;
- authenticate through LAGO IdPs;
- maintain their account, roles and permissions stored by the eduTEAMs platform (GEANT).



https://datahub.egi.eu/ozw/onezone/i#/onedata/spaces/538dae593d8f52ce53ceb768a122b365cha1f4/index

Name LAGOsim
Creator Antonio Juan Ru...
Created at 21 Dec 2020 13:02:33
Shares 0
ID 538dae593 ...

OneData Web portal

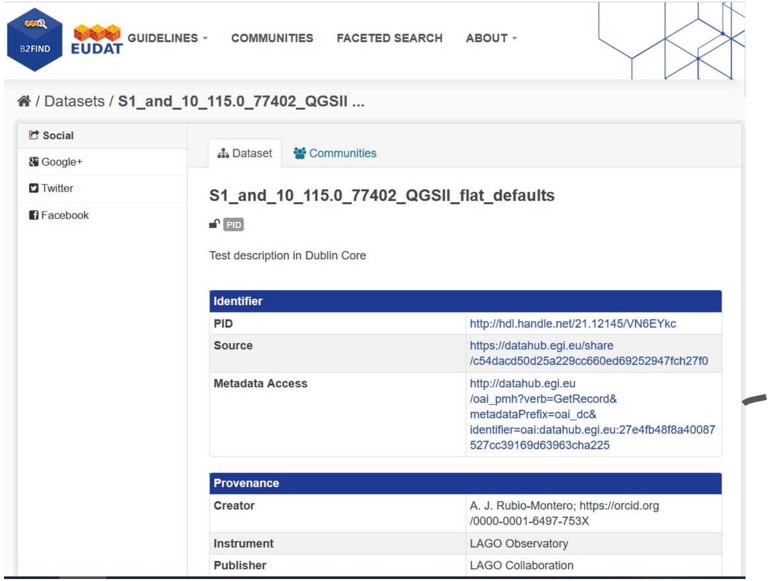
Desktops

Identity Providers for LAGO

How are the meta & data gathered and accessed?

OAI-PMH API

Handle.net PiD to friendly Web page and alternate links through CDMI & REST



/ Datasets / S1_and_10_115.0_77402_QGSII ...

Social
- Google+
- Twitter
- Facebook

Dataset | Communities

S1_and_10_115.0_77402_QGSII_flat_defaults

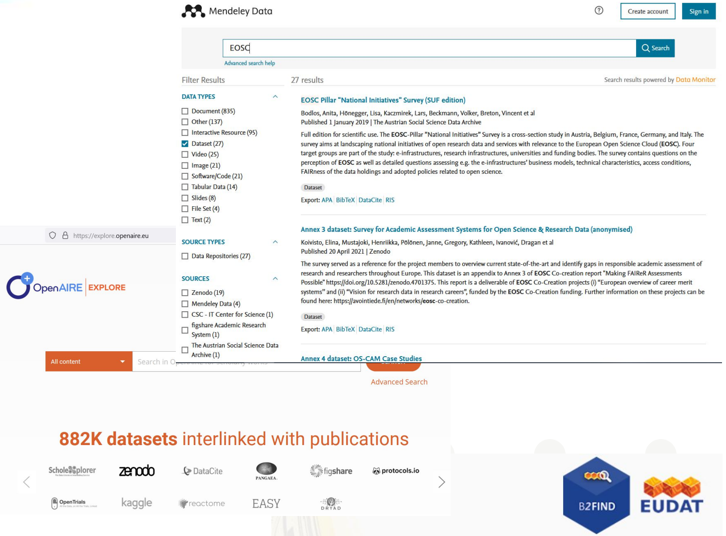
PID

Test description in Dublin Core

Identifier	
PID	http://hdl.handle.net/21.12145/VN6EYkc
Source	https://datahub.egi.eu/share /c54dacd50d25a229cc660ed69252947fch27f0
Metadata Access	http://datahub.egi.eu /oai_pmh?verb=GetRecord& metadataPrefix=oai_dc& identifier=oai:datahub.egi.eu:27e4fb48f8a40087 527cc39169d63963cha225

Provenance	
Creator	A. J. Rubio-Montero; https://orcid.org /0000-0001-6497-753X
Instrument	LAGO Observatory
Publisher	LAGO Collaboration

Metadata gathered for a simulation



882K datasets interlinked with publications

EOSC

Filter Results — 27 results — Search results powered by Data Monitor

DATA TYPES
- ☐ Document (835)
- ☐ Other (137)
- ☐ Interactive Resource (95)
- ☑ Dataset (27)
- ☐ Video (25)
- ☐ Image (21)
- ☐ Software/Code (21)
- ☐ Tabular Data (14)
- ☐ Slides (8)
- ☐ File Set (4)
- ☐ Text (2)

SOURCE TYPES
- ☐ Data Repositories (27)

SOURCES
- ☐ Zenodo (19)
- ☐ Mendeley Data (4)
- ☐ CSC - IT Center for Science (1)
- ☐ figshare Academic Research System (1)
- ☐ The Austrian Social Science Data Archive (1)

EOSC Pillar "National Initiatives" Survey (SUF edition)

Bodlos, Anita, Hönegger, Lisa, Kaczmirek, Lars, Beckmann, Volker, Breton, Vincent et al
Published 1 January 2019 | The Austrian Social Science Data Archive

Full edition for scientific use. The **EOSC**-Pillar "National Initiatives" Survey is a cross-section study in Austria, Belgium, France, Germany, and Italy. The survey aims at landscaping national initiatives of open research data and services with relevance to the European Open Science Cloud (**EOSC**). Four target groups are part of the study: e-infrastructures, research infrastructures, universities and funding bodies. The survey contains questions on the perception of **EOSC** as well as detailed questions assessing e.g. the e-infrastructures' business models, technical characteristics, access conditions, FAIRness of the data holdings and adopted policies related to open science.

Dataset

Export: APA | BibTeX | DataCite | RIS

Annex 3 dataset: Survey for Academic Assessment Systems for Open Science & Research Data (anonymised)

Koivisto, Elina, Mustajoki, Henriikka, Pölönen, Janne, Gregory, Kathleen, Ivanović, Dragan et al
Published 20 April 2021 | Zenodo

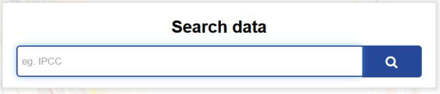
The survey served as a reference for the project members to overview current state-of-the-art and identify gaps in responsible academic assessment of research and researchers throughout Europe. This dataset is an appendix to Annex 3 of **EOSC** Co-creation report "Making FAIReR Assessments Possible" https://doi.org/10.5281/zenodo.4701375. This report is a deliverable of **EOSC** Co-Creation projects (i) "European overview of career merit systems" and (ii) "Vision for research data in research careers", funded by the **EOSC** Co-Creation funding. Further information on these projects can be found here: https://avointiede.fi/en/networks/**eosc**-co-creation.

Dataset

Export: APA | BibTeX | DataCite | RIS

Annex 4 dataset: OS-CAM Case Studies

Search engine



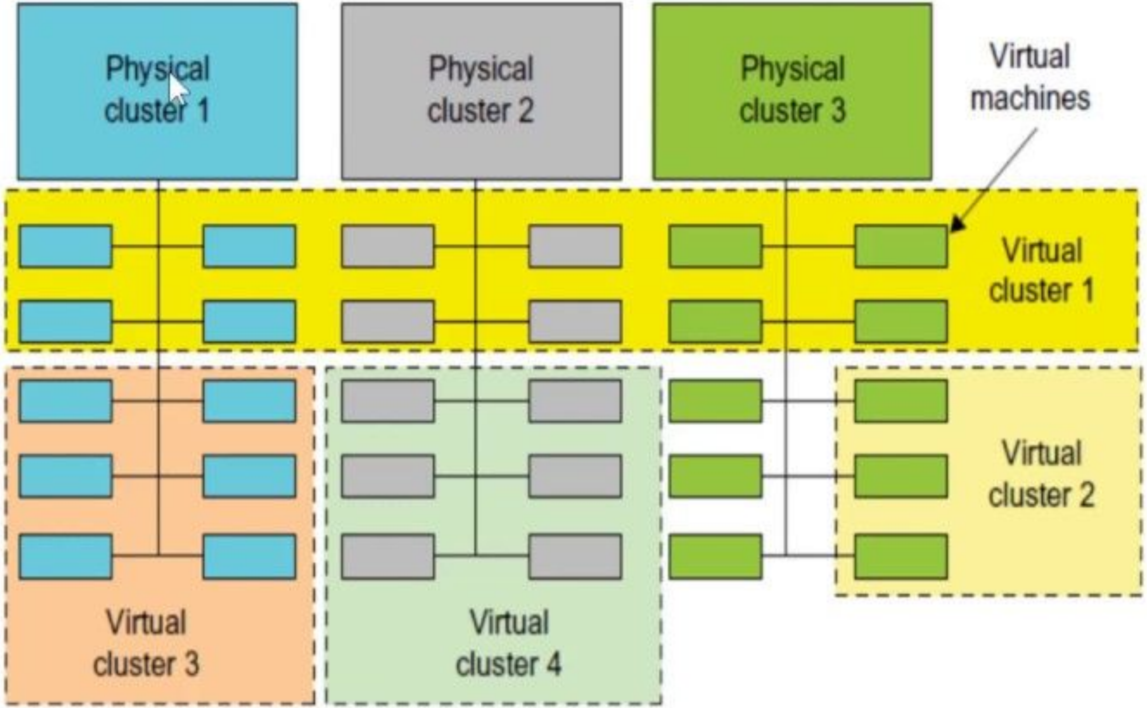
Search data

eg. IPCC

Public harvesters

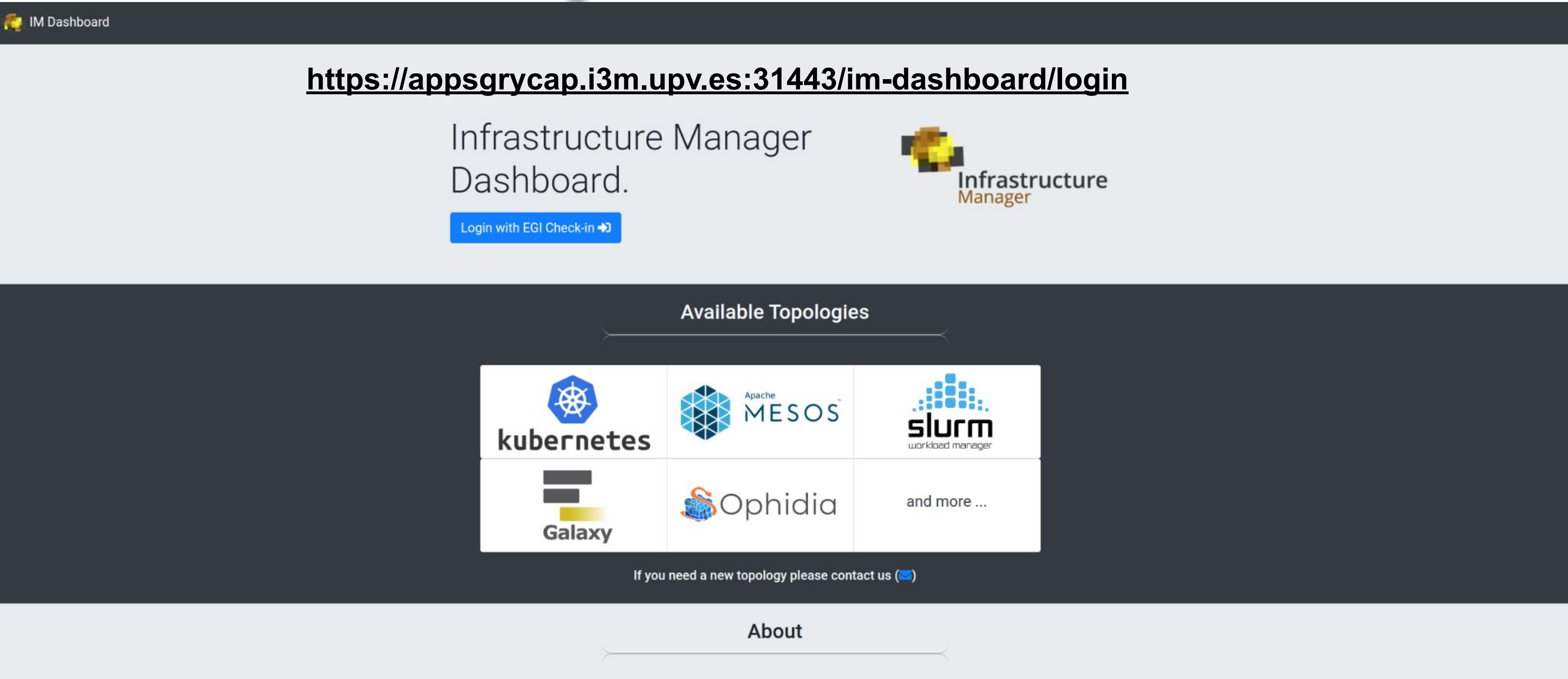
AJ Rubio-Montero et al. (2021), WSC'21 (in press)

cluster virtualization



A cloud platform with four virtual clusters over three physical clusters shaded differently.

Infrastructure manager



IM Dashboard

https://appsgrycap.i3m.upv.es:31443/im-dashboard/login

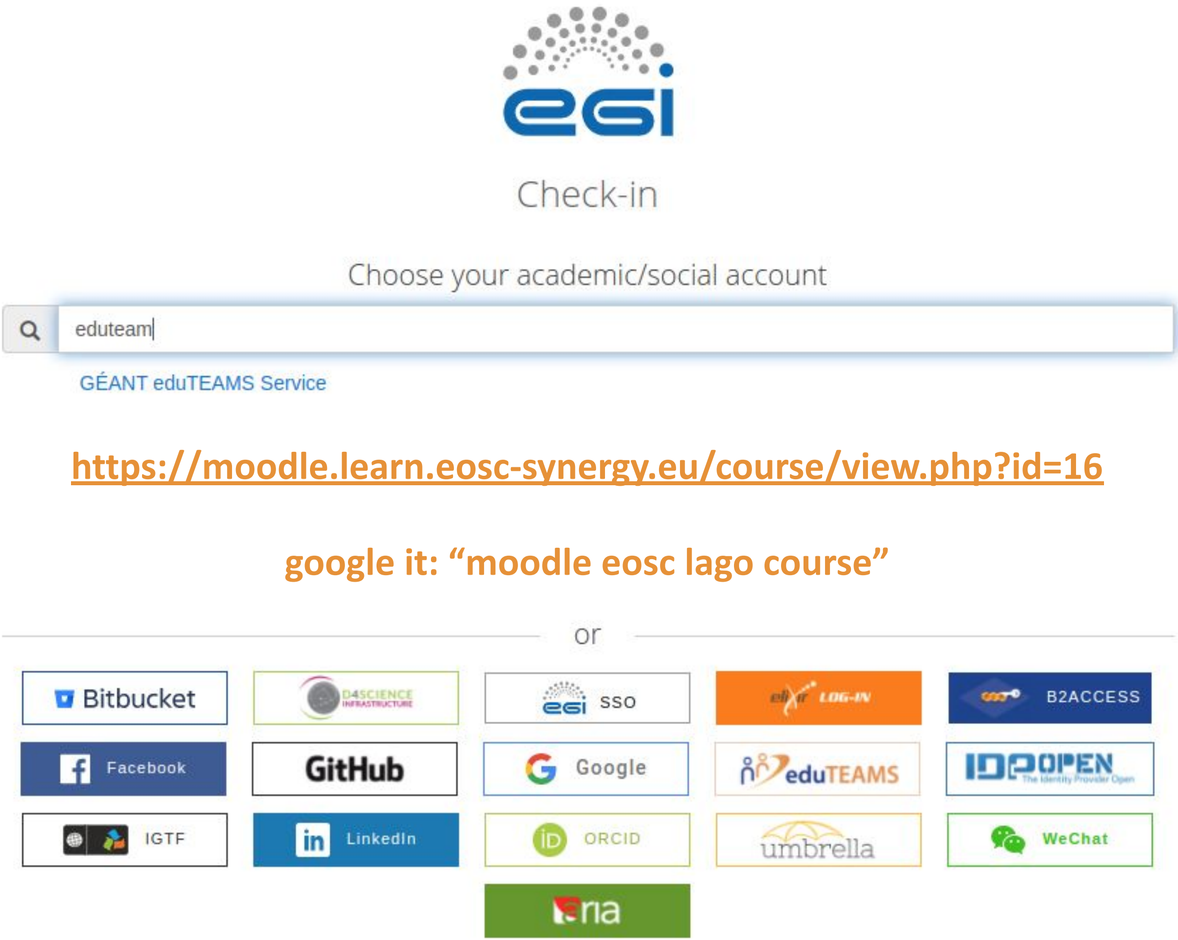
Infrastructure Manager Dashboard.

Login with EGI Check-in

Available Topologies

If you need a new topology please contact us (✉)

About



Check-in

Choose your academic/social account

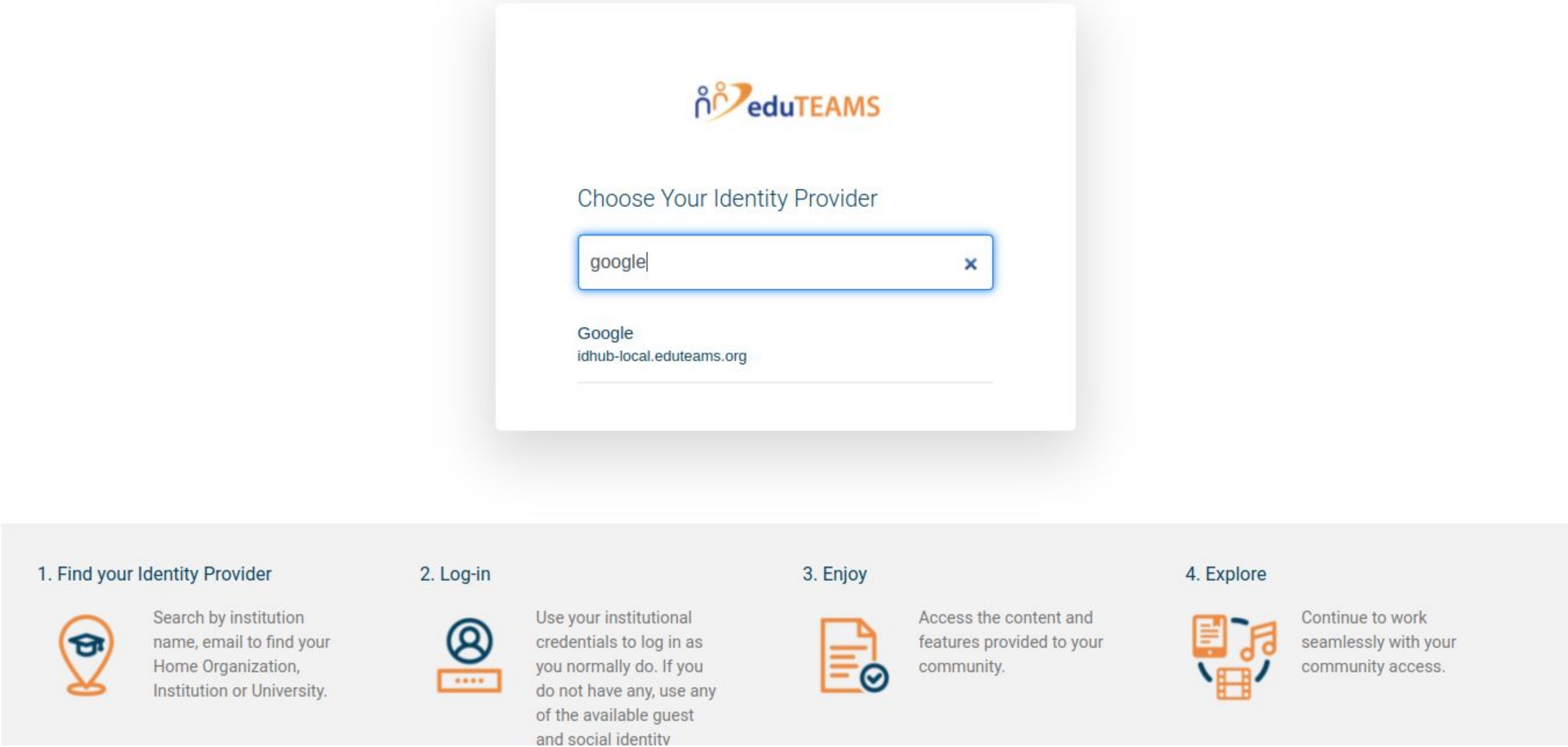
eduteam

GÉANT eduTEAMS Service

or

https://moodle.learn.eosc-synergy.eu/course/view.php?id=16

google it: “moodle eosc lago course”



eduTEAMS

Choose Your Identity Provider

google

Google
idhub-local.eduteams.org

1. Find your Identity Provider

Search by institution name, email to find your Home Organization, Institution or University.

2. Log-in

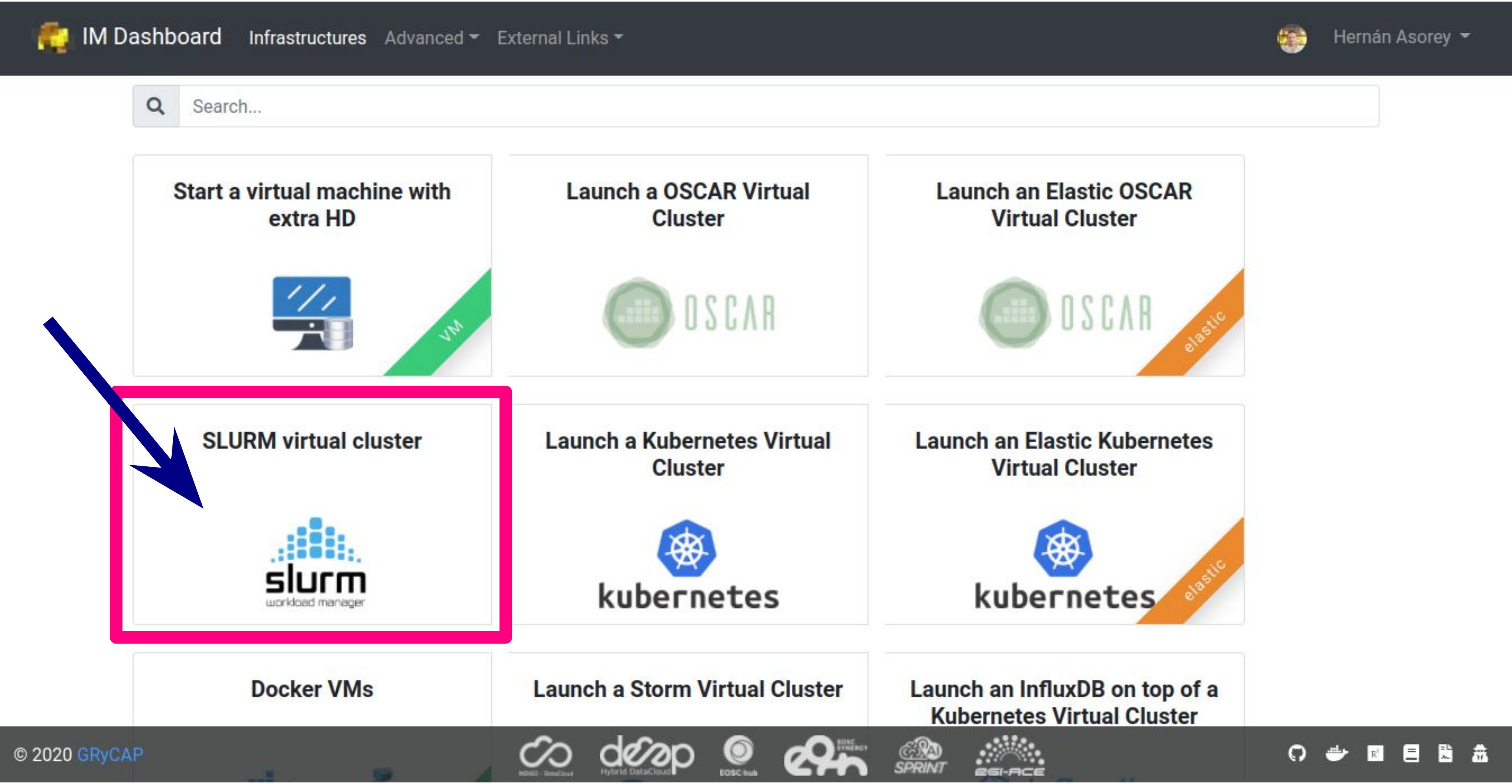
Use your institutional credentials to log in as you normally do. If you do not have any, use any of the available guest and social identity

3. Enjoy

Access the content and features provided to your community.

4. Explore

Continue to work seamlessly with your community access.



Start a virtual machine with extra HD

Launch a OSCAR Virtual Cluster

Launch an Elastic OSCAR Virtual Cluster

SLURM virtual cluster

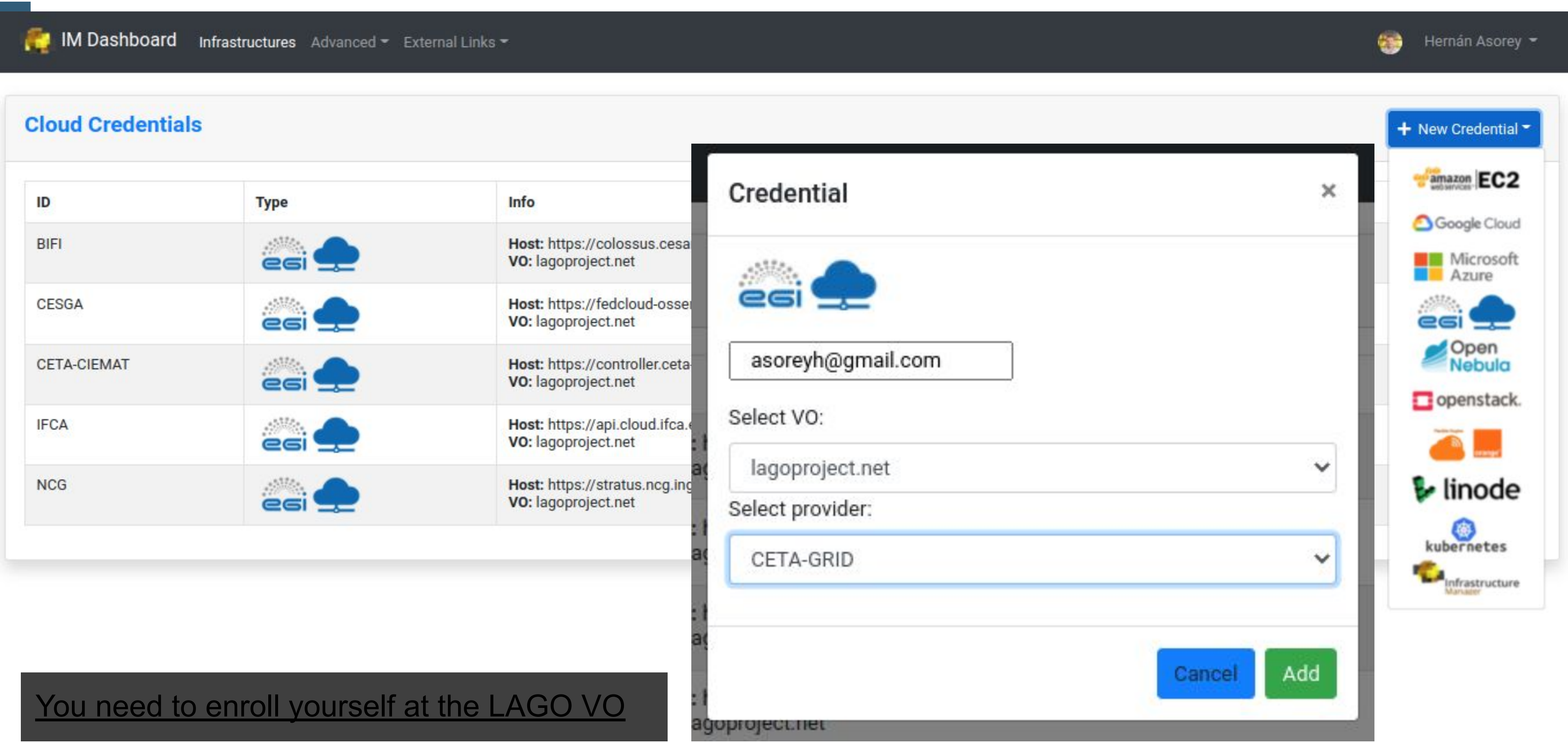
Launch a Kubernetes Virtual Cluster

Launch an Elastic Kubernetes Virtual Cluster

Docker VMs

Launch a Storm Virtual Cluster

Launch an InfluxDB on top of a Kubernetes Virtual Cluster



Cloud Credentials

ID	Type	Info
BIFI		**Host:** https://colossus.cesa **VO:** lagoproject.net
CESGA		**Host:** https://fedcloud-osse **VO:** lagoproject.net
CETA-CIEMAT		**Host:** https://controller.ceta **VO:** lagoproject.net
IFCA		**Host:** https://api.cloud.ifca.e **VO:** lagoproject.net
NCG		**Host:** https://stratus.ncg.ing **VO:** lagoproject.net

Credential

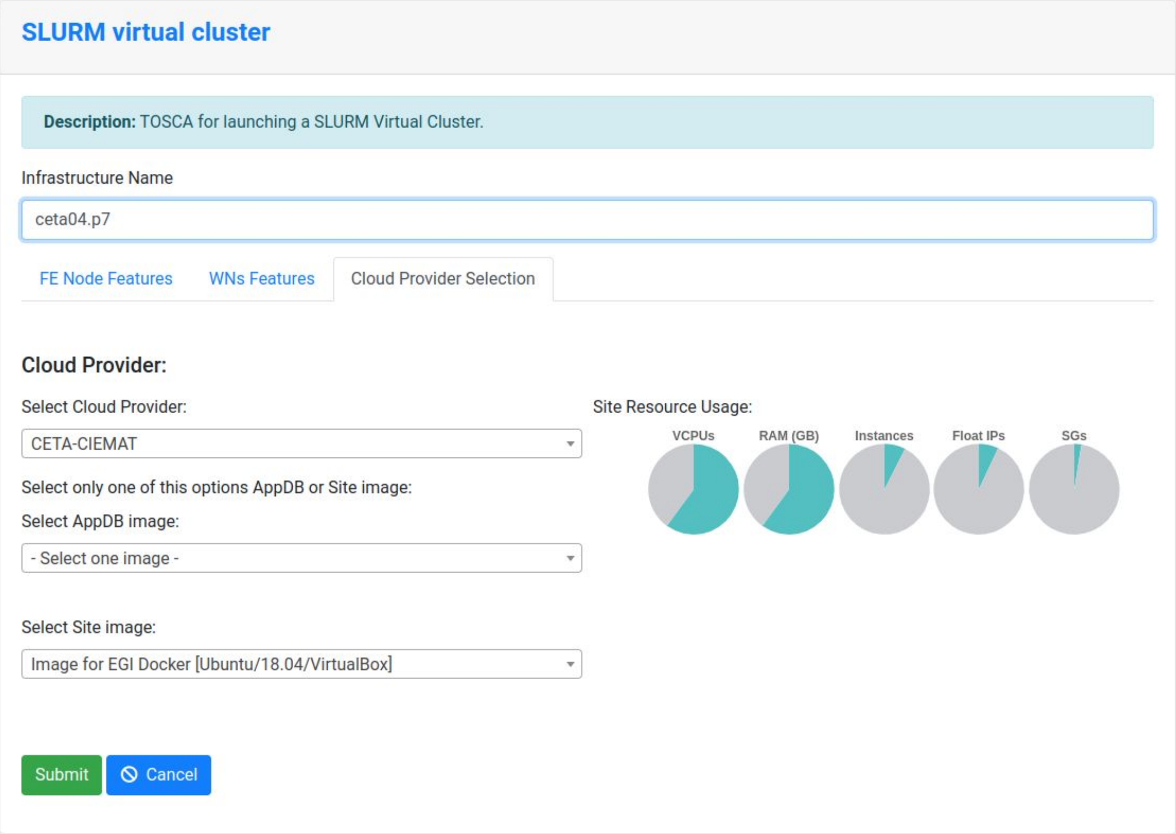
asoreyh@gmail.com

Select VO: lagoproject.net

Select provider: CETA-GRID

You need to enroll yourself at the LAGO VO

Creating a new v-cluster at cloud



SLURM virtual cluster

Description: TOSCA for launching a SLURM Virtual Cluster.

Infrastructure Name: ceta04.p7

Cloud Provider:

Select Cloud Provider: CETA-CIEMAT

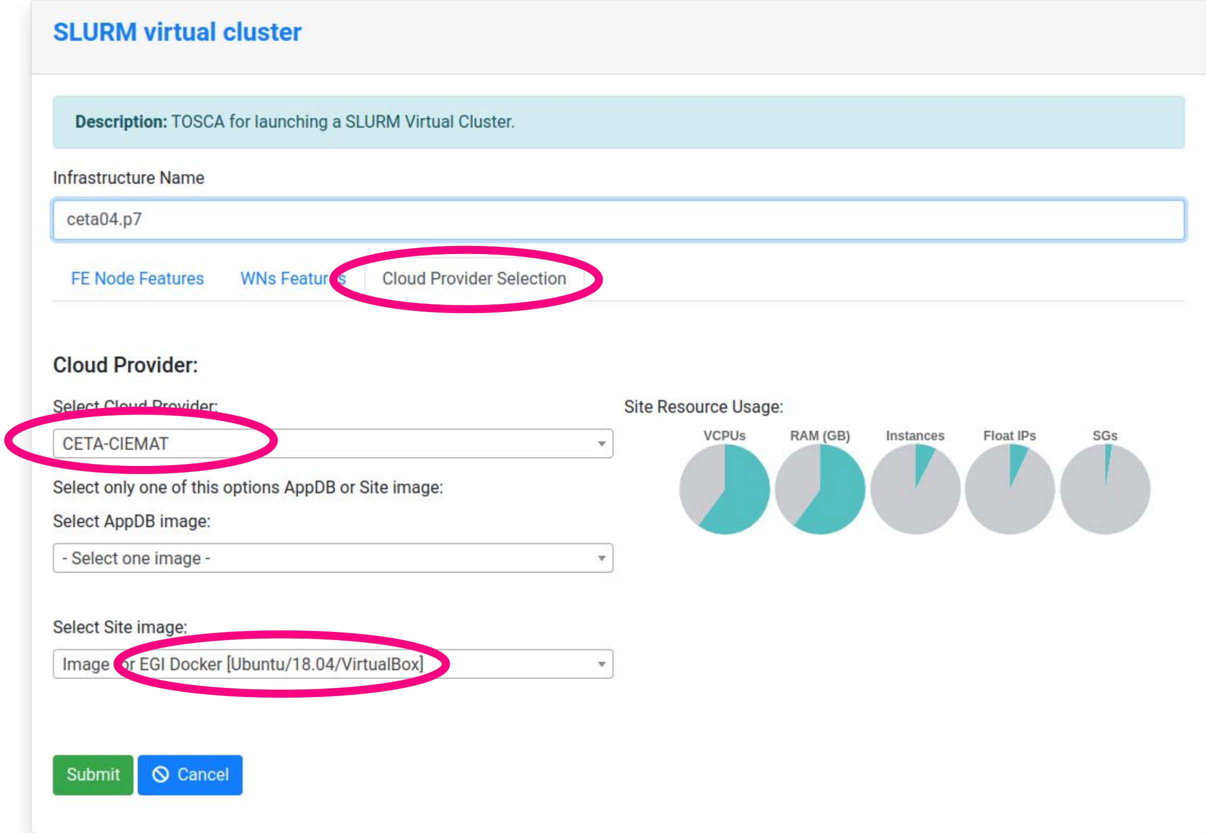
Select only one of this options AppDB or Site image:

Select AppDB image: - Select one image -

Select Site image: Image for EGI Docker [Ubuntu/18.04/VirtualBox]

Site Resource Usage:

Creating a new v-cluster at cloud



SLURM virtual cluster

Description: TOSCA for launching a SLURM Virtual Cluster.

Infrastructure Name: ceta04.p7

Cloud Provider:

Select Cloud Provider: CETA-CIEMAT

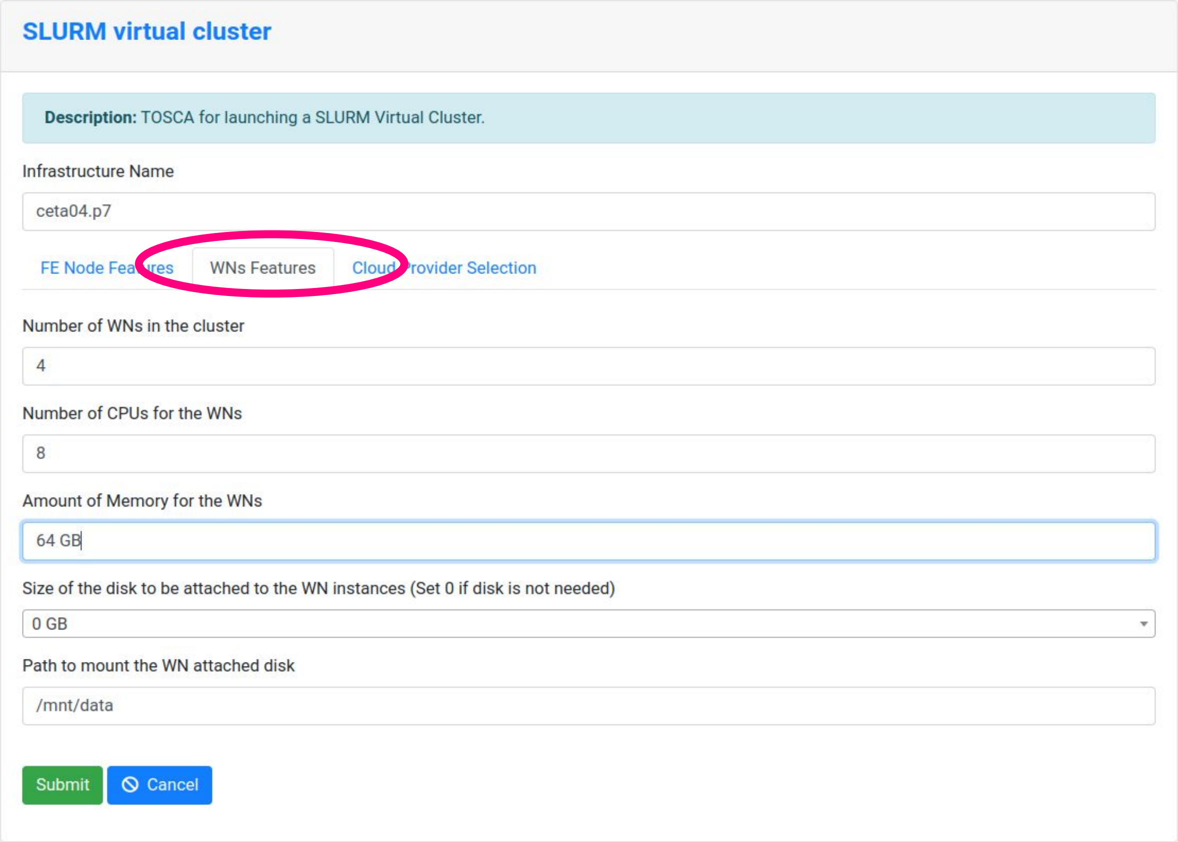
Select only one of this options AppDB or Site image:

Select AppDB image: - Select one image -

Select Site image: Image for EGI Docker [Ubuntu/18.04/VirtualBox]

Site Resource Usage:

Creating a new v-cluster at cloud



Description: TOSCA for launching a SLURM Virtual Cluster.

Infrastructure Name: ceta04.p7

Number of WNs in the cluster: 4

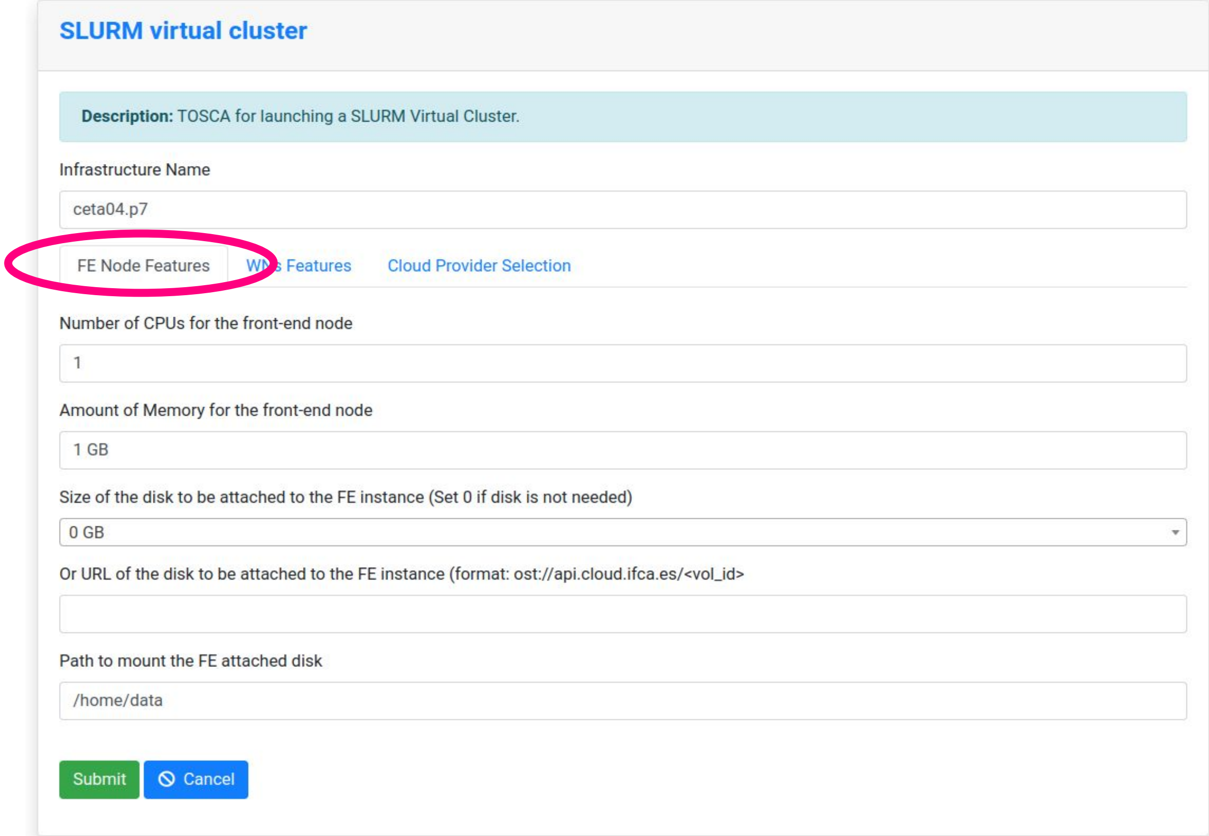
Number of CPUs for the WNs: 8

Amount of Memory for the WNs: 64 GB

Size of the disk to be attached to the WN instances (Set 0 if disk is not needed): 0 GB

Path to mount the WN attached disk: /mnt/data

Creating a new v-cluster at cloud



SLURM virtual cluster

Description: TOSCA for launching a SLURM Virtual Cluster.

Infrastructure Name: ceta04.p7

Number of CPUs for the front-end node: 1

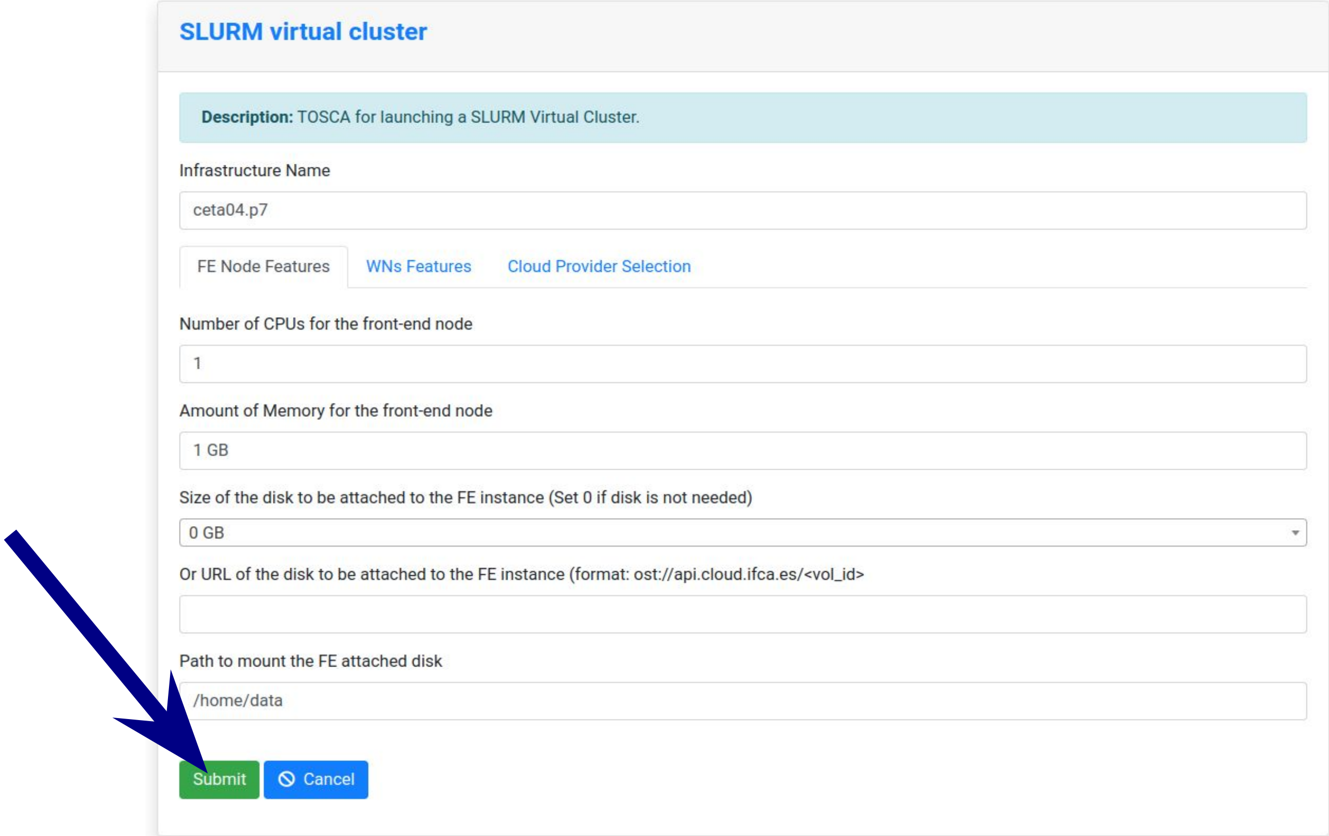
Amount of Memory for the front-end node: 1 GB

Size of the disk to be attached to the FE instance (Set 0 if disk is not needed): 0 GB

Or URL of the disk to be attached to the FE instance (format: ost://api.cloud.ifca.es/<vol_id>

Path to mount the FE attached disk: /home/data

Creating a new v-cluster at cloud



SLURM virtual cluster

Description: TOSCA for launching a SLURM Virtual Cluster.

Infrastructure Name: ceta04.p7

Number of CPUs for the front-end node: 1

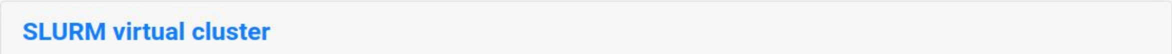
Amount of Memory for the front-end node: 1 GB

Size of the disk to be attached to the FE instance (Set 0 if disk is not needed): 0 GB

Or URL of the disk to be attached to the FE instance (format: ost://api.cloud.ifca.es/<vol_id>

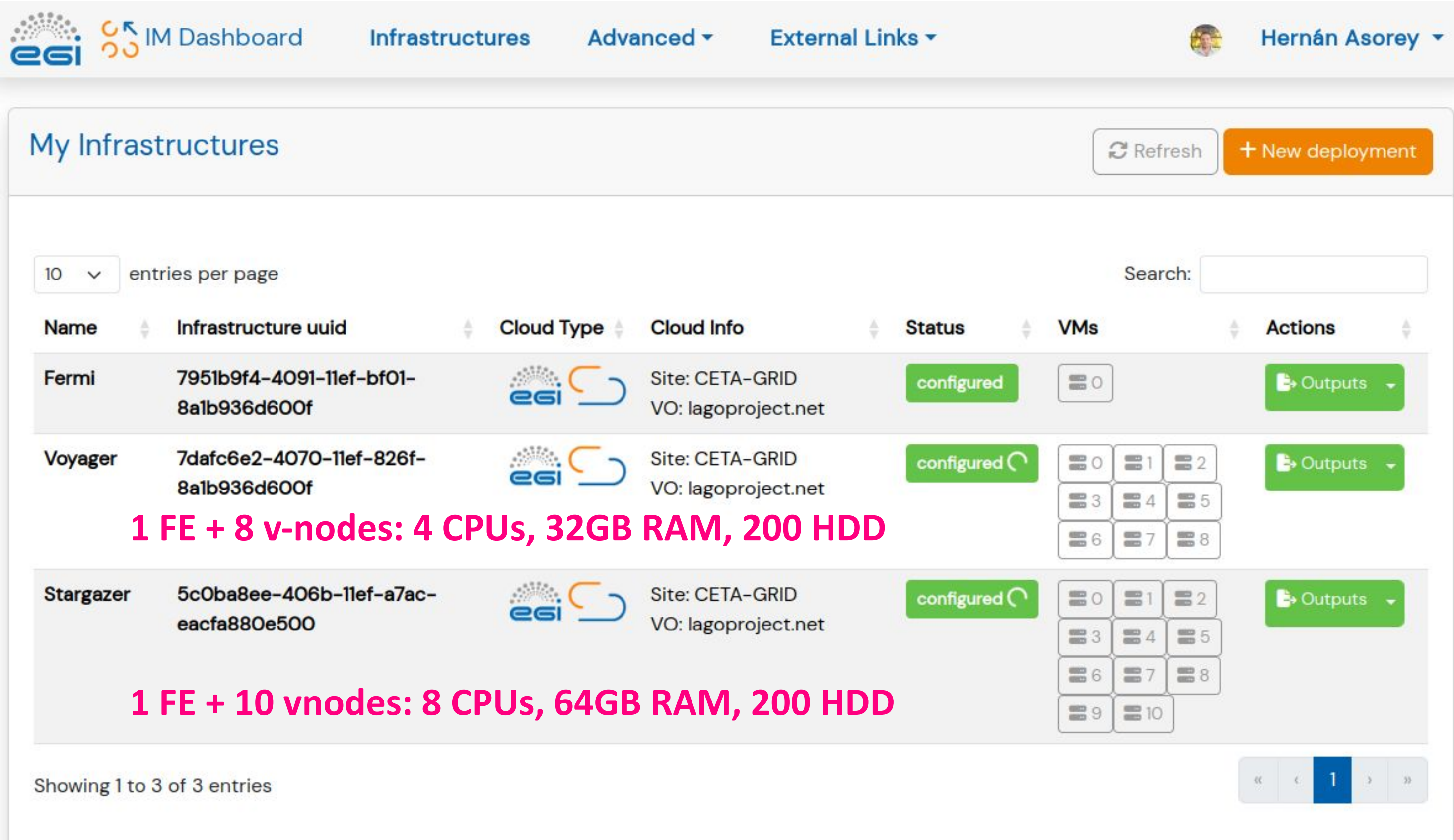
Path to mount the FE attached disk: /home/data

Creating a new v-cluster at cloud



SLURM virtual cluster

Current LAGO services at cloud (v-clusters) (+ RES' HPC)



My Infrastructures

10 entries per page

Search:

Name	Infrastructure uuid	Cloud Type	Cloud Info	Status	VMs	Actions
Fermi	7951b9f4-4091-11ef-bf01-8a1b936d600f		Site: CETA-GRID VO: lagoproject.net	configured	0	Outputs
Voyager	7dafc6e2-4070-11ef-826f-8a1b936d600f		Site: CETA-GRID VO: lagoproject.net	configured	0 1 2 3 4 5 6 7 8	Outputs
Stargazer	5c0ba8ee-406b-11ef-a7ac-eacfa880e500		Site: CETA-GRID VO: lagoproject.net	configured	0 1 2 3 4 5 6 7 8 9 10	Outputs

Showing 1 to 3 of 3 entries

1 FE + 8 v-nodes: 4 CPUs, 32GB RAM, 200 HDD

1 FE + 10 vnodes: 8 CPUs, 64GB RAM, 200 HDD

+ Single virtual machines on-demand

+ 20TB cloud storage via Onedata

How does it work?

HPC provider assigns cloud resources to EOSC:
n Nodes, *r* GB/TB of RAM, *d* TB local storage

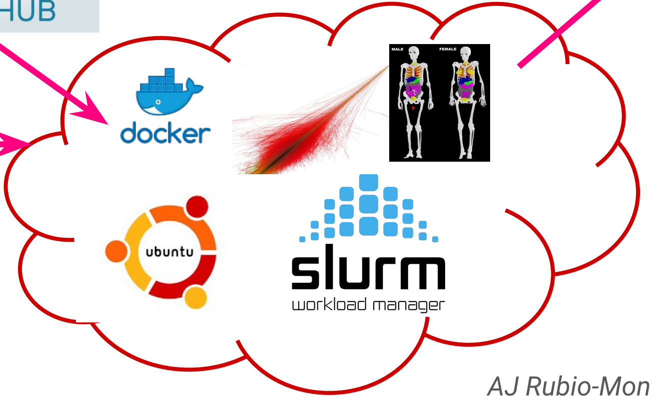
Infrastructure Manager

web-based service with templates for distributing to available resources in virtual clusters. Installs OS, in our case, ubuntu 20.04 + slurm manager

docker HUB

docker containers with our codes are deployed from docker HUB in the virtual cluster

virtual cluster



B2FIND

FAIR: catalogs are findable, accessible, interoperable, and reusable

ONEDATA

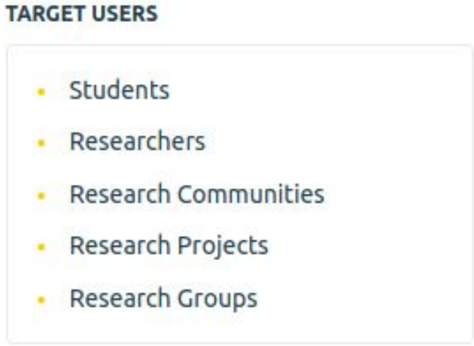
Results are stored at cloud-based storage services. Access trough personal tokens

B2HANDLE

PID (Persistent IDentifiers) are assigned for each data catalog

AJ Rubio-Montero et al., IEEE WSC2021 21706697, 2021

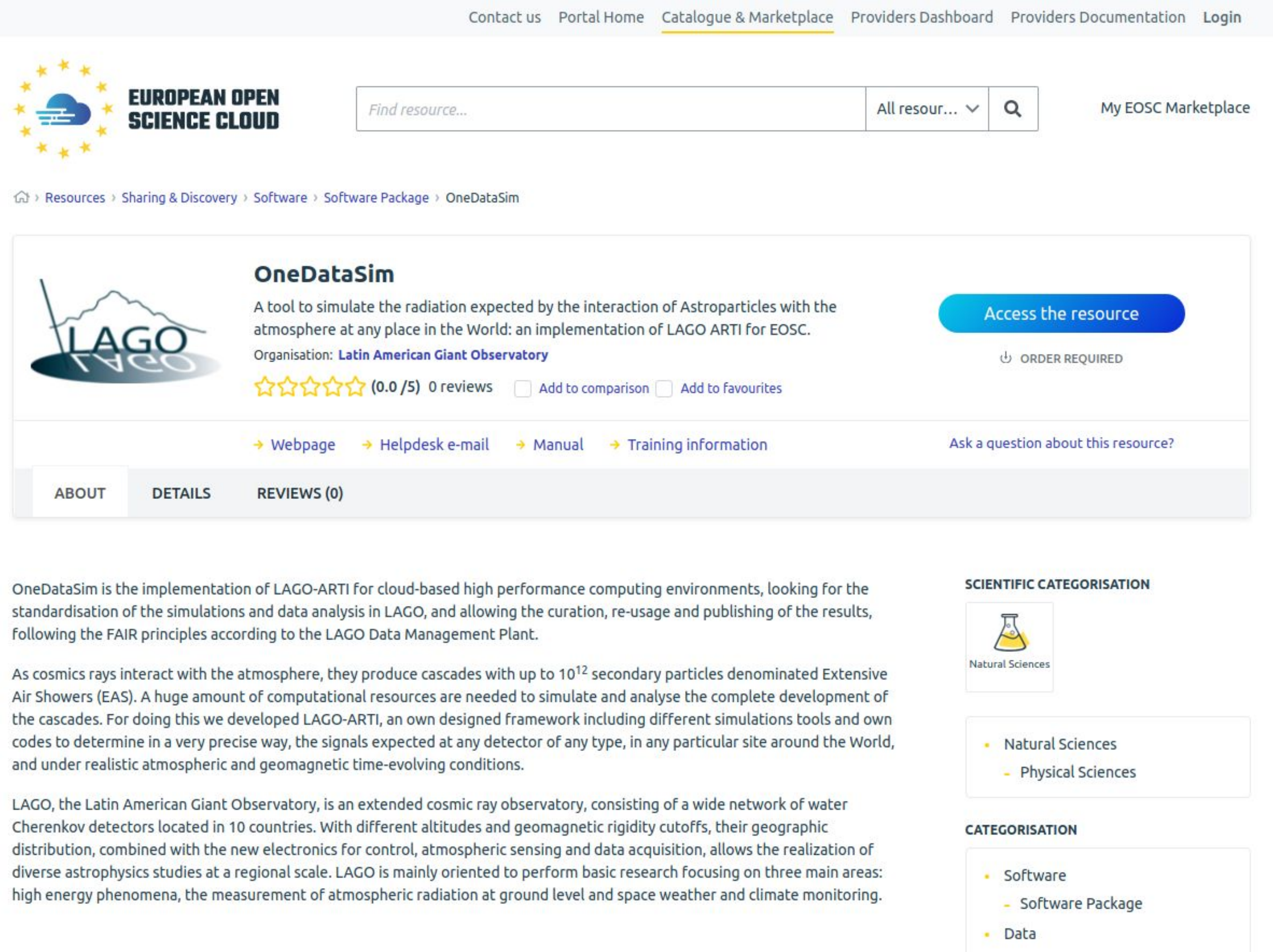
OneDataSim



TARGET USERS

- Students
- Researchers
- Research Communities
- Research Projects
- Research Groups

Free service now publicly available at EOSC Marketplace



Resources › Sharing & Discovery › Software › Software Package › OneDataSim

OneDataSim

A tool to simulate the radiation expected by the interaction of Astroparticles with the atmosphere at any place in the World: an implementation of LAGO ARTI for EOSC.

Organisation: **Latin American Giant Observatory**

☆☆☆☆☆ (0.0 /5) 0 reviews ☐ Add to comparison ☐ Add to favourites

OneDataSim is the implementation of LAGO-ARTI for cloud-based high performance computing environments, looking for the standardisation of the simulations and data analysis in LAGO, and allowing the curation, re-usage and publishing of the results, following the FAIR principles according to the LAGO Data Management Plant.

As cosmics rays interact with the atmosphere, they produce cascades with up to 10^{12} secondary particles denominated Extensive Air Showers (EAS). A huge amount of computational resources are needed to simulate and analyse the complete development of the cascades. For doing this we developed LAGO-ARTI, an own designed framework including different simulations tools and own codes to determine in a very precise way, the signals expected at any detector of any type, in any particular site around the World, and under realistic atmospheric and geomagnetic time-evolving conditions.

LAGO, the Latin American Giant Observatory, is an extended cosmic ray observatory, consisting of a wide network of water Cherenkov detectors located in 10 countries. With different altitudes and geomagnetic rigidity cutoffs, their geographic distribution, combined with the new electronics for control, atmospheric sensing and data acquisition, allows the realization of diverse astrophysics studies at a regional scale. LAGO is mainly oriented to perform basic research focusing on three main areas: high energy phenomena, the measurement of atmospheric radiation at ground level and space weather and climate monitoring.

SCIENTIFIC CATEGORISATION

Natural Sciences

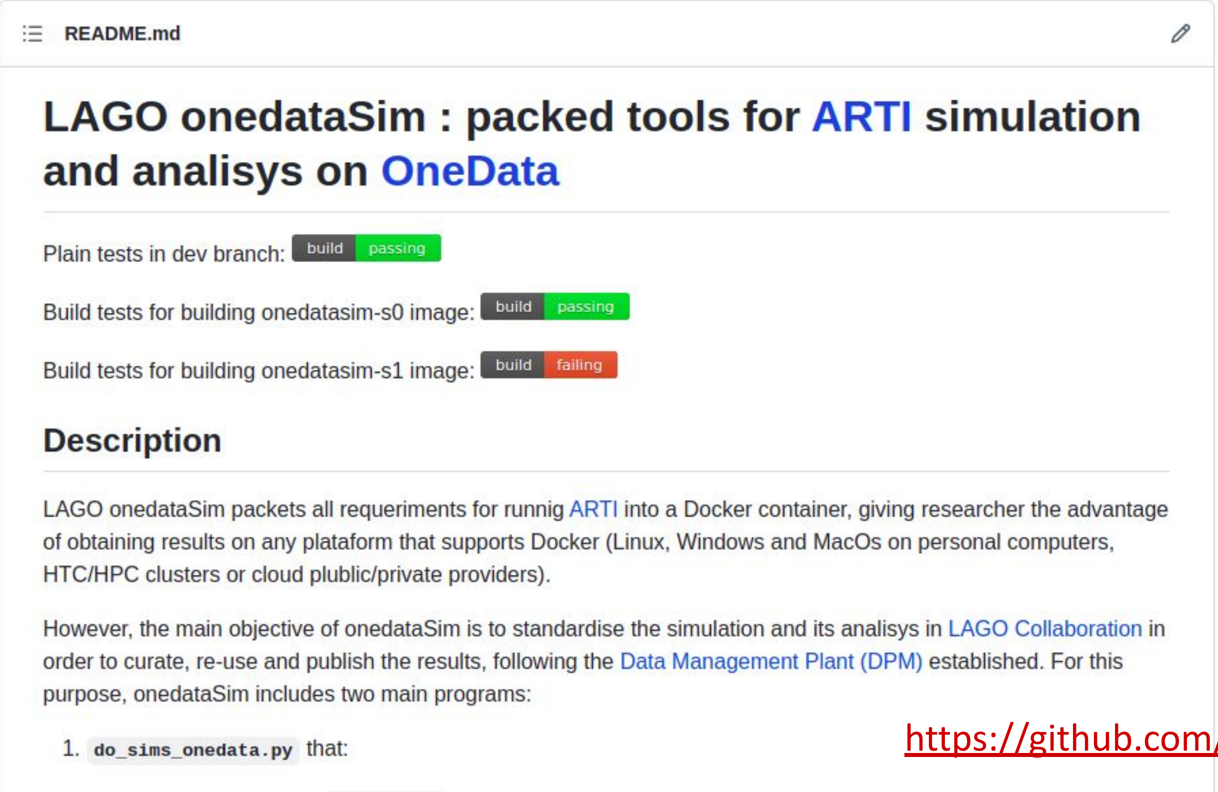
- Natural Sciences
  - Physical Sciences

CATEGORISATION

- Software
  - Software Package
- Data

one $TOKEN to rule them all

LAGO OneDataDim



LAGO onedataSim : packed tools for ARTI simulation and analisys on OneData

Plain tests in dev branch: build passing

Build tests for building onedatasim-s0 image: build passing

Build tests for building onedatasim-s1 image: build failing

Description

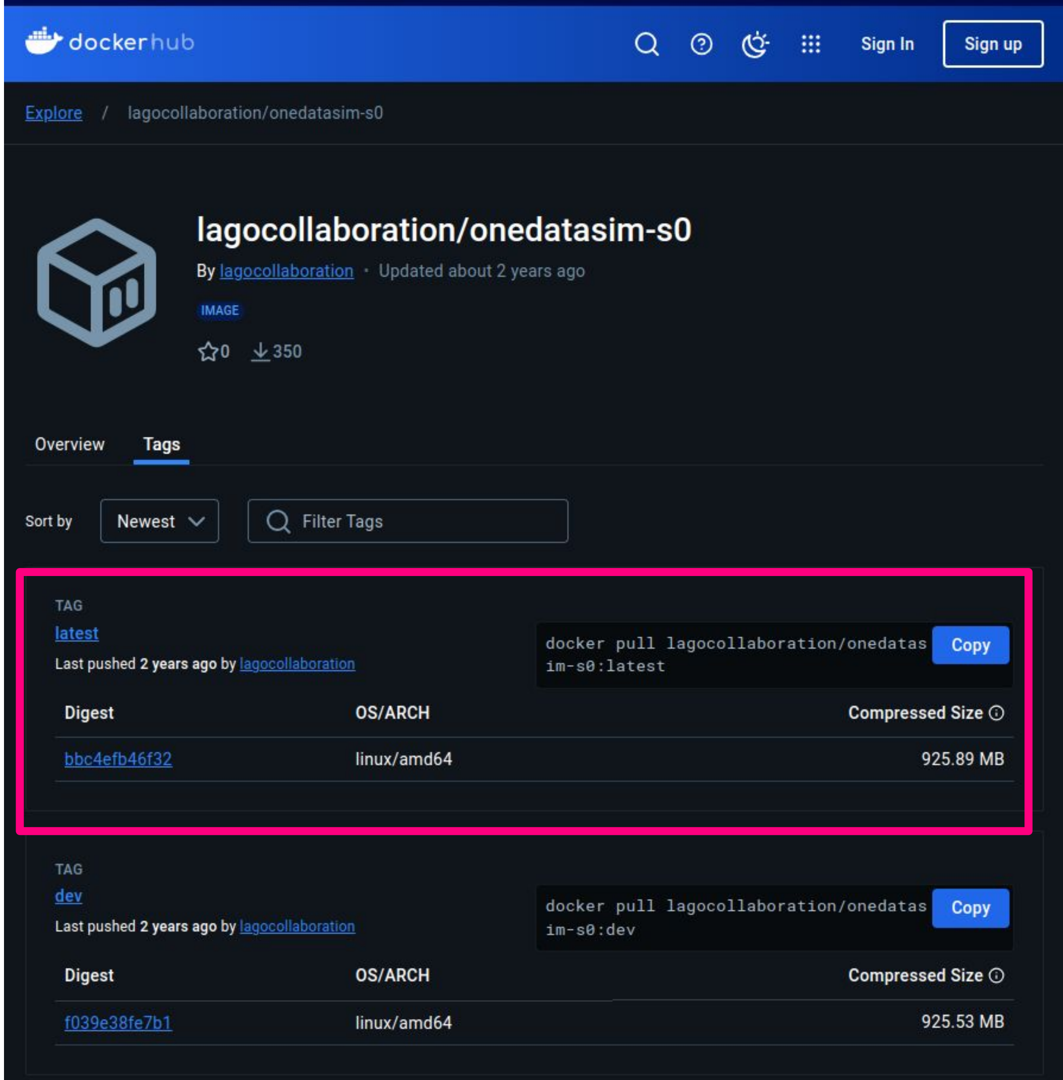
LAGO onedataSim packets all requeriments for runnig ARTI into a Docker container, giving researcher the advantage of obtaining results on any plataform that supports Docker (Linux, Windows and MacOs on personal computers, HTC/HPC clusters or cloud plublic/private providers).

However, the main objective of onedataSim is to standardise the simulation and its analisys in LAGO Collaboration in order to curate, re-use and publish the results, following the Data Management Plant (DPM) established. For this purpose, onedataSim includes two main programs:

1. `do_sims_onedata.py` that:

https://github.com/lagoproject/onedataSim

LAGO docker hub



lagocollaboration/onedatasim-s0

By lagocollaboration · Updated about 2 years ago

IMAGE

TAG

latest

Last pushed **2 years ago** by lagocollaboration

```
docker pull lagocollaboration/onedatasim-s0:latest
```

Digest	OS/ARCH	Compressed Size
bbc4efb46f32	linux/amd64	925.89 MB

TAG

dev

Last pushed **2 years ago** by lagocollaboration

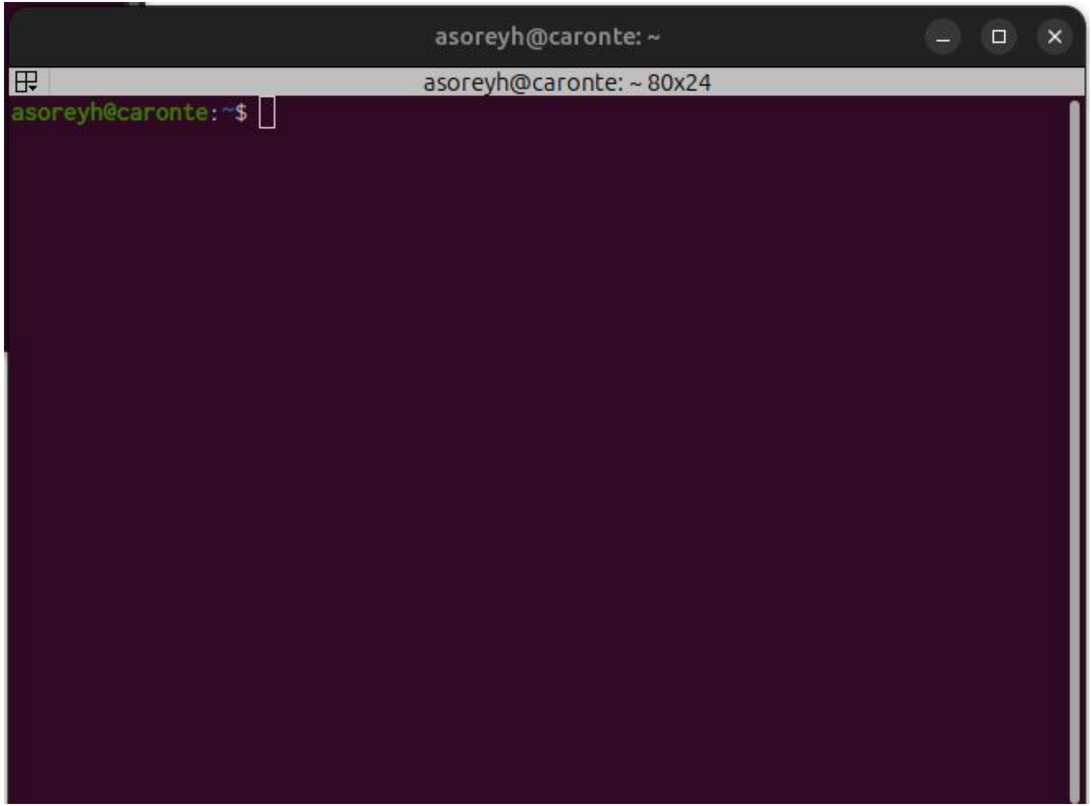
```
docker pull lagocollaboration/onedatasim-s0:dev
```

Digest	OS/ARCH	Compressed Size
f039e38fe7b1	linux/amd64	925.53 MB

lagocollaboration/onedatasim-s0

- CentOS Linux release v 7.8.2003 (Core, rpm based)
- Standard installation of LAGO CORSIKA (CORSIKA v 7.7402 compiled with LAGO standards)
- ARTI v 1.9.0
- onedataSim 1.0 branch: dev commit: dc803020
- FUSE v 2.9
- oneclient v 20.02.17
- coreutils, vim, rsync, screen, python v 3.6.8, …

let’s go to the terminal



```
asoreyh@caronte:~$ 
```

TOKEN & PROVIDER

The access to the shared resources is granted trought oauth2 TOKEN provided at EGI DATAHUB: https://datahub.egi.eu/

```
export TOKEN="MDAxY2xv…b5Ao"
```

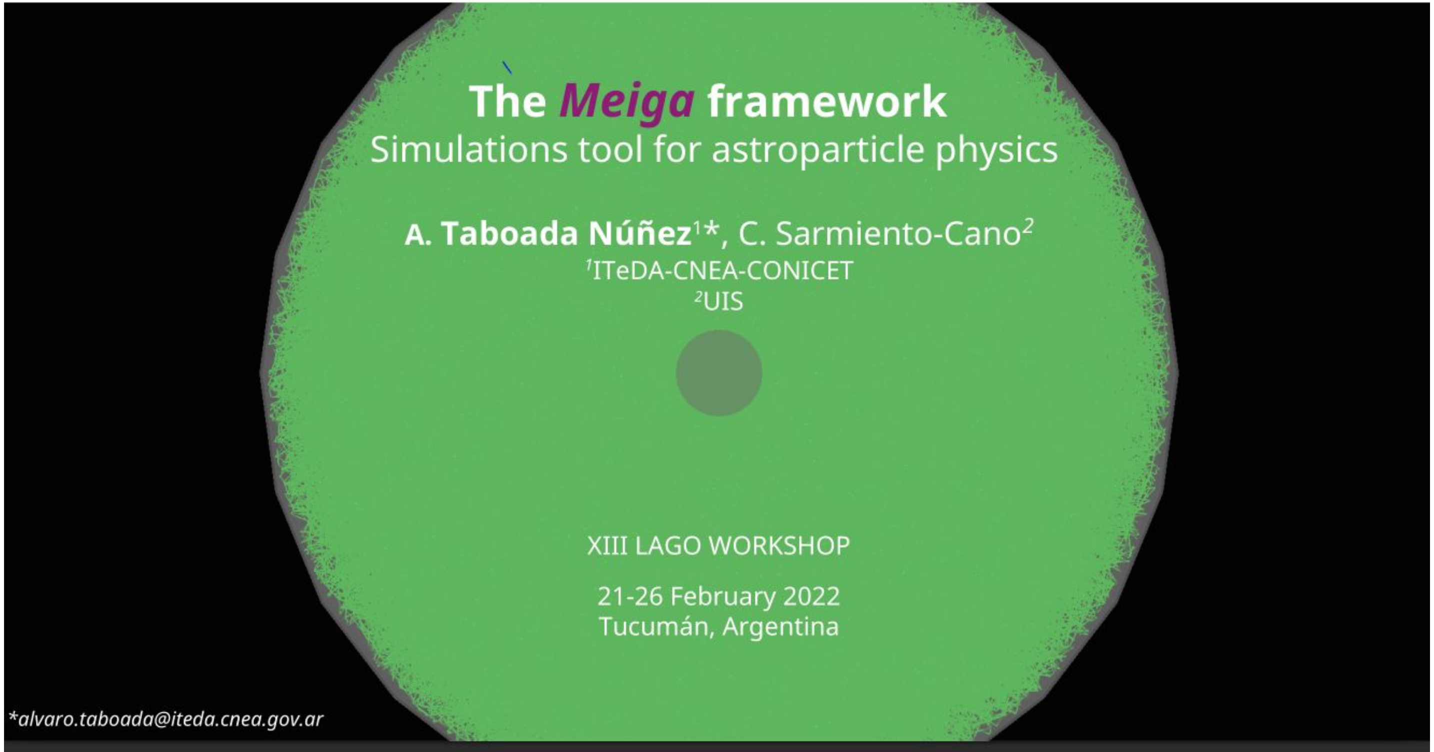
The provider allows the access to the cloud storage server:

```
export ONEPROVIDER="ceta-ciemat-01.datahub.egi.eu"
```

Dockers

```
# Get the docker

docker rmi lagocollaboration/onedatasim-s0:dev

docker pull lagocollaboration/onedatasim-s0:dev

# Run the docker

docker run \
  --privileged \
  -e ONECLIENT_ACCESS_TOKEN=$TOKEN \
  -e ONECLIENT_PROVIDER_HOST=$ONEPROVIDER \
  -it lagocollaboration/onedatasim-s0:dev \
  bash
```

EXTRACT FROM



The *Meiga* framework

Simulations tool for astroparticle physics

A. Taboada Núñez[1]*, C. Sarmiento-Cano[2]
[1]ITeDA-CNEA-CONICET
[2]UIS

XIII LAGO WORKSHOP

21-26 February 2022
Tucumán, Argentina

*alvaro.taboada@iteda.cnea.gov.ar

AND FROM

```
G4Course *MyNewCourse::Construct() {

    G4Course *course = new G4Course();
    course→title(“Geant4 for Beginners. A crash course”);
    course→author(“Hernán Asorey”);
    course→email(“asoreyh@gmail.com”);
    course→description(“a hands-on Geant4 crash course”);
    course→school(“La Conga Physics”);
    course→site(“github.com/asoreyh/geant4-course”);
    course→year(2023);
    course→duration(4*h);
    course→license(“CC0 1.0 Universal”);
    return course;
}
```

geant4

// A toolkit for the simulation of the passage of particles through matter.

// Its areas of application include high energy, nuclear and accelerator physics, as well as studies in medical and space science

```
G4Download("geant4.web.cern.ch/");
G4Docs("geant4.web.cern.ch/docs/");
G4AppDocs("geant4-userdoc.web.cern.ch/UsersGuides/ForApplicationDeveloper/BackupVersions/V10.7/html/index.html");
G4Examples("Check the ${geant4_examples}/ dir for extra fun");
```

// IMPORTANT NOTICE

// I will not spend time showing how to install G4 (and root) →

G4 install (by @asoreyh)

```bash
# While Geant4 have multiple dependencies,
some of them are required and some others are
needed for optional features.

# Check the official installation guide at
https://geant4-userdoc.web.cern.ch/UsersGuides/InstallationGuide/html

# I prepared a bash script for installing the required dependencies,
root and geant4 at ${HOME}/work. (Ubuntu 23.04)
# Warning 1: it will change your .bashrc file.
# Warning 2: it will take time (up to several hours).

$ curl -Lo install-root-geant4.sh
https://www.dropbox.com/s/ej67f1hc88u7w1a/install-root-geant4.sh?dl=1
$ chmod 744 install-root-geant4.sh
$ ./install-root-geant4.sh
```

G4 docker (by @asoreyh)

```
# Docker is a platform designed to help devs
build, share, and run modern apps. We handle the
tedious setup, so you can focus on the code.
# I prepared two Dockerfiles for this course. Follow the instructions
and download them from (look for them at the utils directory):

$ git clone https://github.com/asoreyh/geant4-course.git

# Otherwise, you can pull the docker image from my docker hub:

$ docker pull asoreyh/root:latest    # root version 6.28.04 (2023)
$ docker pull asoreyh/geant4:latest # G4 version 10.07.04 (2022)

# Blank installation. Check the docs!
# There is also a virtual machine built by the Geant4 collaboration
```

G4 docker

(by @asoreyh)

Important note: You will need to follow these steps and provide privileged access to this docker to be able to run the QT Geant4 visualization from docker.

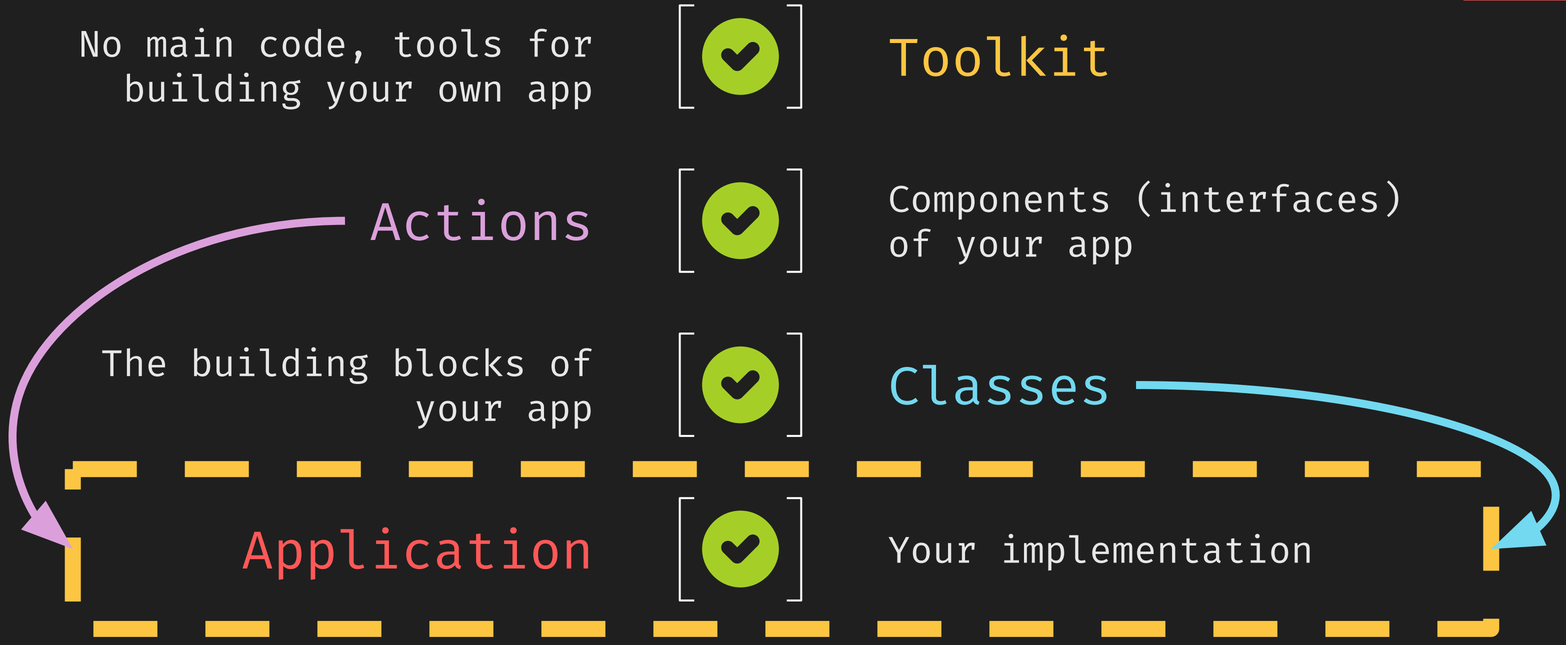
```
# open a terminal and enable local access to xhost:

$ xhost +local:root

# and run the docker (if you don’t download the docker images it will
download them):

$ docker run --privileged -it -e DISPLAY=$DISPLAY -v
/tmp/.X11-unix:/tmp/.X11-unix asoreyh/geant4:10.07.04
```

geant4

```
// A comment about versions.
/*
   By the end of 2022 a new major release, G4 11
   was released (current 11.2.2)

   In this course we will use the latest G4 10,
   G4 10.7.4
*/
```

Geant4 philosophy → Toolkit

G4Help();



No main code, tools for building your own app [✓] Toolkit

Actions [✓] Components (interfaces) of your app

The building blocks of your app [✓] Classes

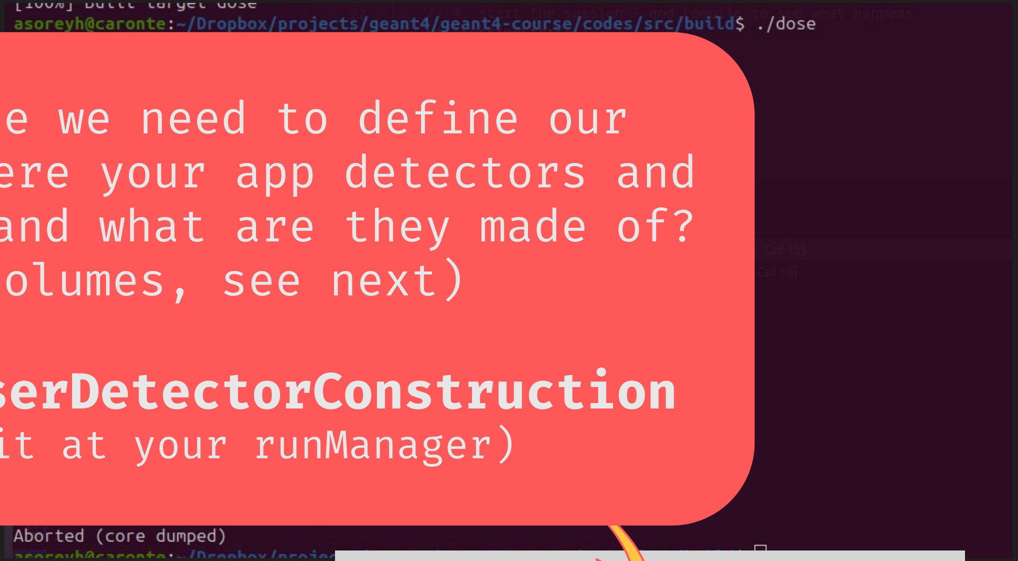
Application [✓] Your implementation

G4RunManager()

```
/*

    This object is the "heart" of any G4 application. It is always mandatory and
    should be defined in your main app.cc code (dose.cc in our example)

    It controls the "flow" of the run

    All the interfaces (G4 toolkit) are defined and provided here:

        * G4VUserDetectorConstruction             ← geometry construction

        * G4VUserPhysicsList                 ← all your physics is here

        * G4VUserActionInitialization             ← actions

            * G4VUserPrimaryGeneratorAction       ← primary particles production

            * G4UserRunAction                     ← optionals…

            * G4UserSteppingAction, …

        * UIManager, VisManager, …

*/
```

First step: the main code (dose.cc)

```
// 0. I/O operations
#include <iostream>
// 1. G4RunManager class
#include "G4RunManager.hh"
// 2. User interface
#include "G
#include
// 3. Vi
#include
#include

int main
      //
      G4
      //
      ru
      //
      G4
      G4UI
      //3. visualization manager
      G4VisManager *visManager = new G4VisExecutive();
      visManager->Initialize();
      // 4. start the session - and compile to see what happens
      ui->SessionStart();
      return 0;
}
```



```
[100%] Built target dose
asoreyh@caronte:~/Dropbox/projects/geant4/geant4-course/codes/src/build$ ./dose
Aborted (core dumped)
asoreyh@caronte:~/Dropbox/projec
```

Before to continue we need to define our "volumes", i.e., where your app detectors and volumes will exist and what are they made of? (always 3 volumes, see next)

Create your G4VUserDetectorConstruction
(and register it at your runManager)

// Clearly we are still not ready for **initialize()** the **runManager** as we need to continue defining our **basic building blocks**

And the new dose.cc

```
// 0. I/O operations
#include <iostream>
// 1. G4RunManager class
#include "G4RunManager.hh"
// 2. User interface
#include "G4UImanager.hh"
#include "G4UIExecutive.hh"
// 3. Visualization
#include "G4VisManager
#include "G4VisExecuti
// once the detector c
#include "construction
int main(G4int argc, ch
   //1. create the G4R
   G4RunManager *runMa
   // once detector is
   // but we are still
   runManager->SetUser
   //5. Initialize the
   // runManager->Init
   //2. create the use
   G4UIExecutive *ui = ne
   G4UImanager *UIManager = G4UImanager::GetUIpointer();
   //3. visualization manager
   G4VisManager *visManager = new G4VisExecutive();
   visManager->Initialize();
   // 4. start the session - and compile to see what happens
   ui->SessionStart();
   return 0;
}
```

It compiles! :)

```
t4-course/codes/src/build$ make
/dose.cc.o
urse/codes/src/dose.cc: In function

urse/codes/src/dose.cc:26:18:
variable]
r::GetUIpointer();

construction.cc.o
```

rk… :/

Before to continue we need to define our “physics”, i.e., what kind of physics our app will implement?

Create your G4VUserPhysicsList
(and register it at your runManager)

Physics, this is why we are here…

// Physics process

Physics processes describe how particles interact with materials.

- electromagnetic
- hadronic
- transportation
- decay
- optical
- photolepton_hadron
- Parameterisation

G4PhysicsListDocs(“https://geant4-userdoc.web.cern.ch/UsersGuides/ForApplicationDeveloper/BackupVersions/V10.7/html/TrackingAndPhysics/physicsProcess.html”);

And the new lines at dose.cc

```
// [...]
// once the physics list is ready, include it
#include "physics.hh"

int main(G4int argc, char** argv) {

// [...]

// once the ph
runManager->Se

// [...]
// after the p
UIManager->App
UIManager->App
// ... draw th
UIManager->App

// [...]
}
```

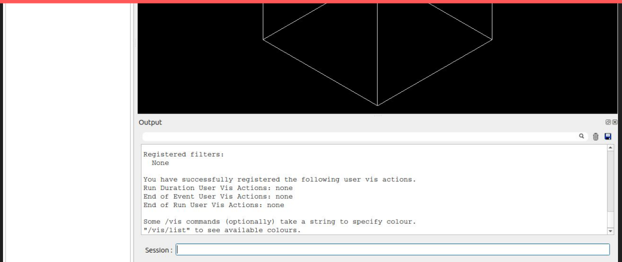
It compiles! :)

```
asoreyh@caronte:~/Dropbox/projects/geant4/geant4-course/codes/src/build$ make
[ 33%] Building CXX object CMakeFiles/dose.dir/dose.cc.o
/home/asoreyh/Dropbox/projects/geant4/geant4-course/codes/src/dose.cc: In function

e.cc:26:18:

uild$ ./dose
```

know)

Ok, we have volumes, materials and physics…
We are almost ready → we need actions!

Create your G4VUserActionInitialization
and G4VUserPrimaryGeneratorAction
(and register them at your runManager)



```
Registered filters:
  None

You have successfully registered the following user vis actions.
Run Duration User Vis Actions: none
End of Event User Vis Actions: none
End of Run User Vis Actions: none

Some /vis commands (optionally) take a string to specify colour.
"/vis/list" to see available colours.
```

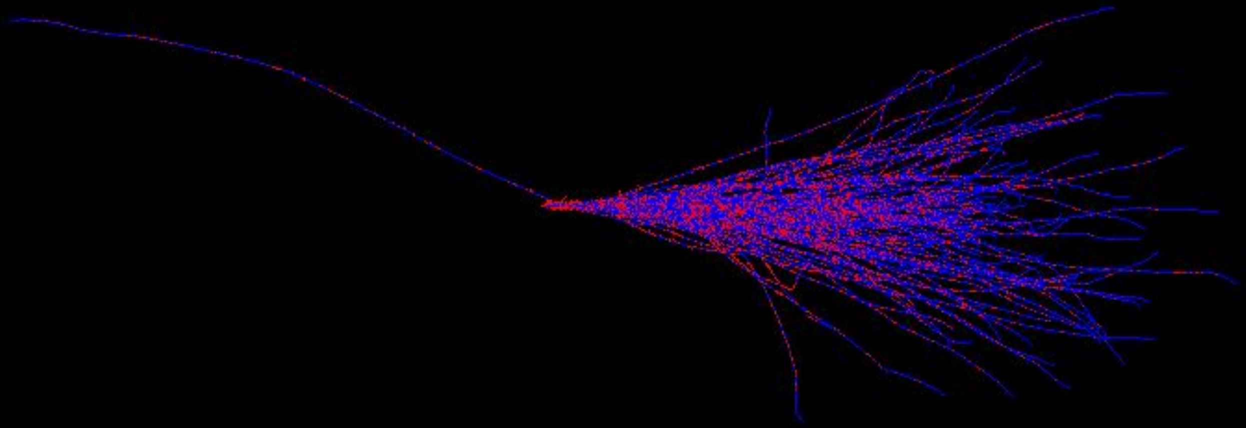
Actions, let's the things evolve

// We need two interfaces

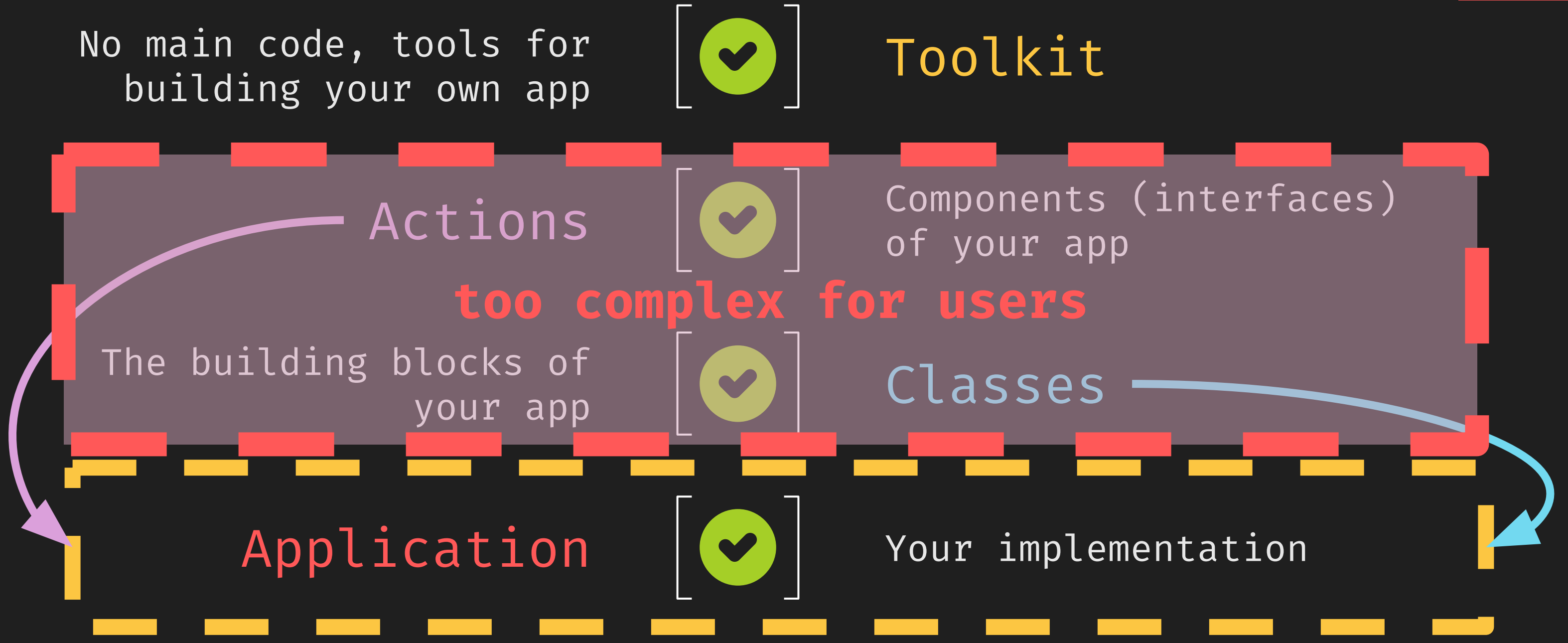
- **G4VUserActionInitialization** is an interface to create and register the **G4VUserPrimaryGeneratorAction** (mandatory) and other user actions
  - Build() ← function

- **G4VUserPrimaryGeneratorAction** is an interface (action!) to describe how the primary particles (injection) should be produced
  - GenerateParticles() ← function
  - Typically, but not always → **G4ParticleGun**
  -

and “that’s it”



Geant4 philosophy→ the problem



No main code, tools for building your own app [✓] Toolkit

Actions [✓] Components (interfaces) of your app

too complex for users

The building blocks of your app [✓] Classes

Application [✓] Your implementation

Meiga, the sorceress

developed by A. Taboada-Núnez et al.

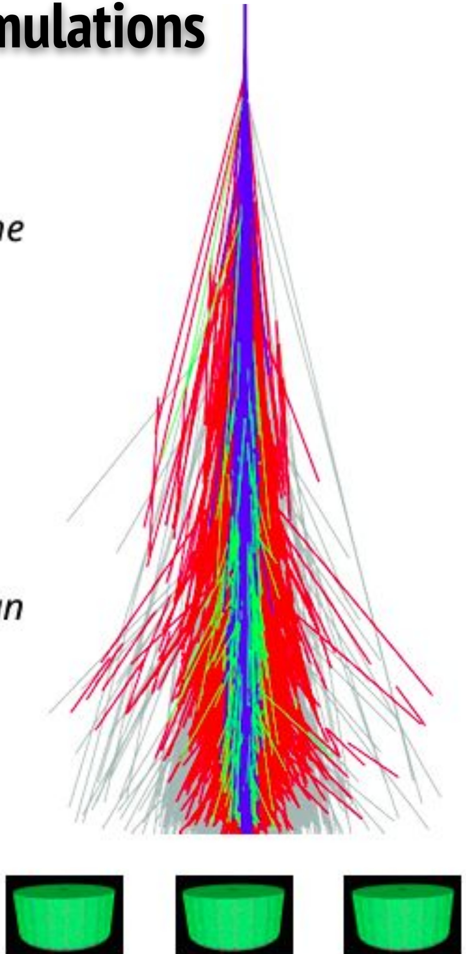
[Download Meiga from GitHub](https://github.com)

MEIGA: modular matter-radiation interaction simulations

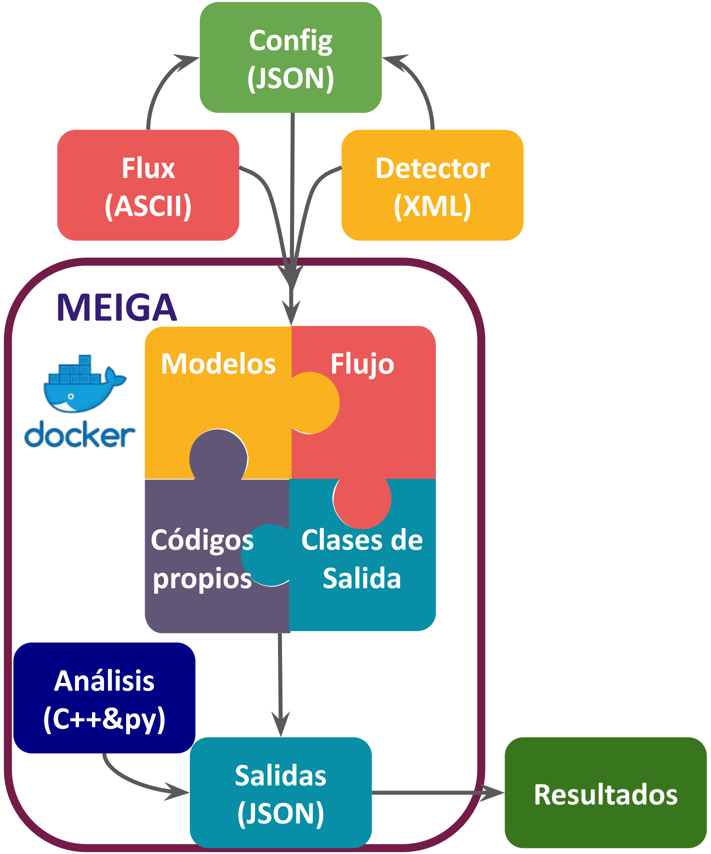
- What is it?
  Meiga *is a collection of C++ classes that integrate the cosmic-ray flux calculation, particle propagation and the detector response in dedicated applications.*

- How?
  The framework is supported by an structure of classes and methods which provide the necessary interfaces between the user and the detector description.

  The applications are written in such a way that users can run different simulation setups by means of configuration files.



MEIGA: modular matter-radiation interaction simulations



Config (JSON)

Flux (ASCII)

Detector (XML)

MEIGA

Modelos

Flujo

Códigos propios

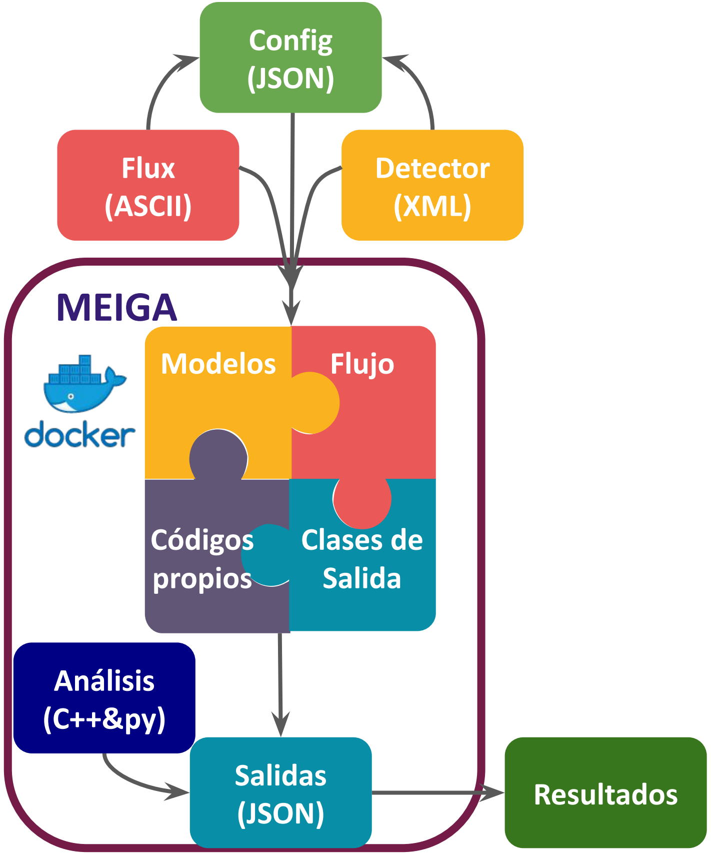
Clases de Salida

Análisis (C++&py)

Salidas (JSON)

Resultados

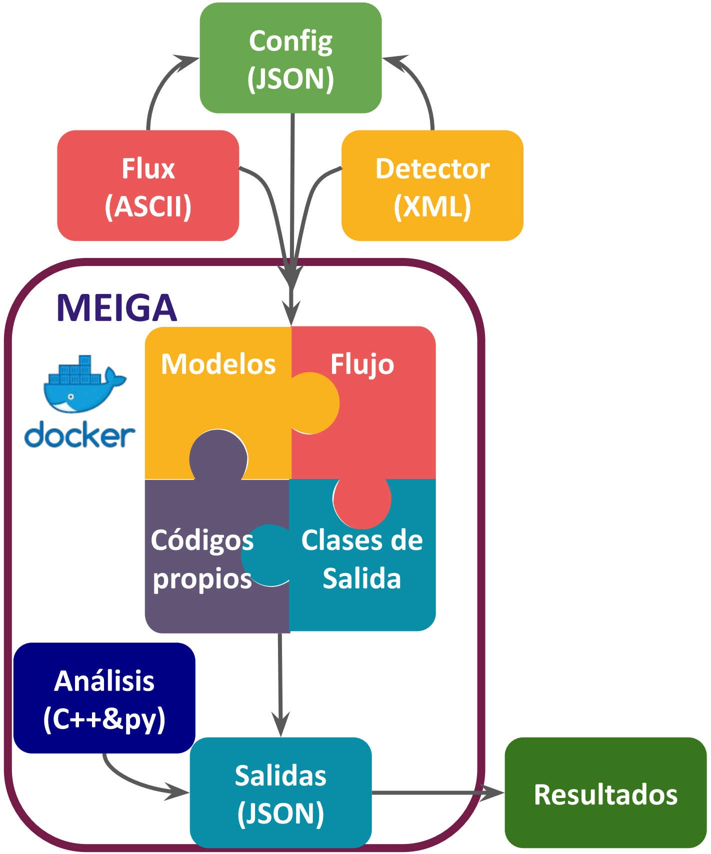
MEIGA: modular matter-radiation interaction simulations



after install root, Geant4 and Meiga dependences, download meiga from GitHub and follow the typical installation sequence:

```
$ git clone git@github.com:ataboadanunez/meiga.git

$ cd meiga

$ mkdir build install

$ cd build

$ cmake -DCMAKE_INSTALL_PREFIX=../install ../src

$ make -j 4; make -j 4

$ make install
```

MEIGA: modular matter-radiation interaction simulations



Config (JSON)

Flux (ASCII)

Detector (XML)

MEIGA

Modelos

Flujo

Códigos propios

Clases de Salida

Análisis (C++&py)

Salidas (JSON)

Resultados

or, use docker (all running dependencies included)

```
$ docker pull asoreyh/meiga:dev
```

of course, you need docker properly installed in your system

asoreyh/meiga:dev

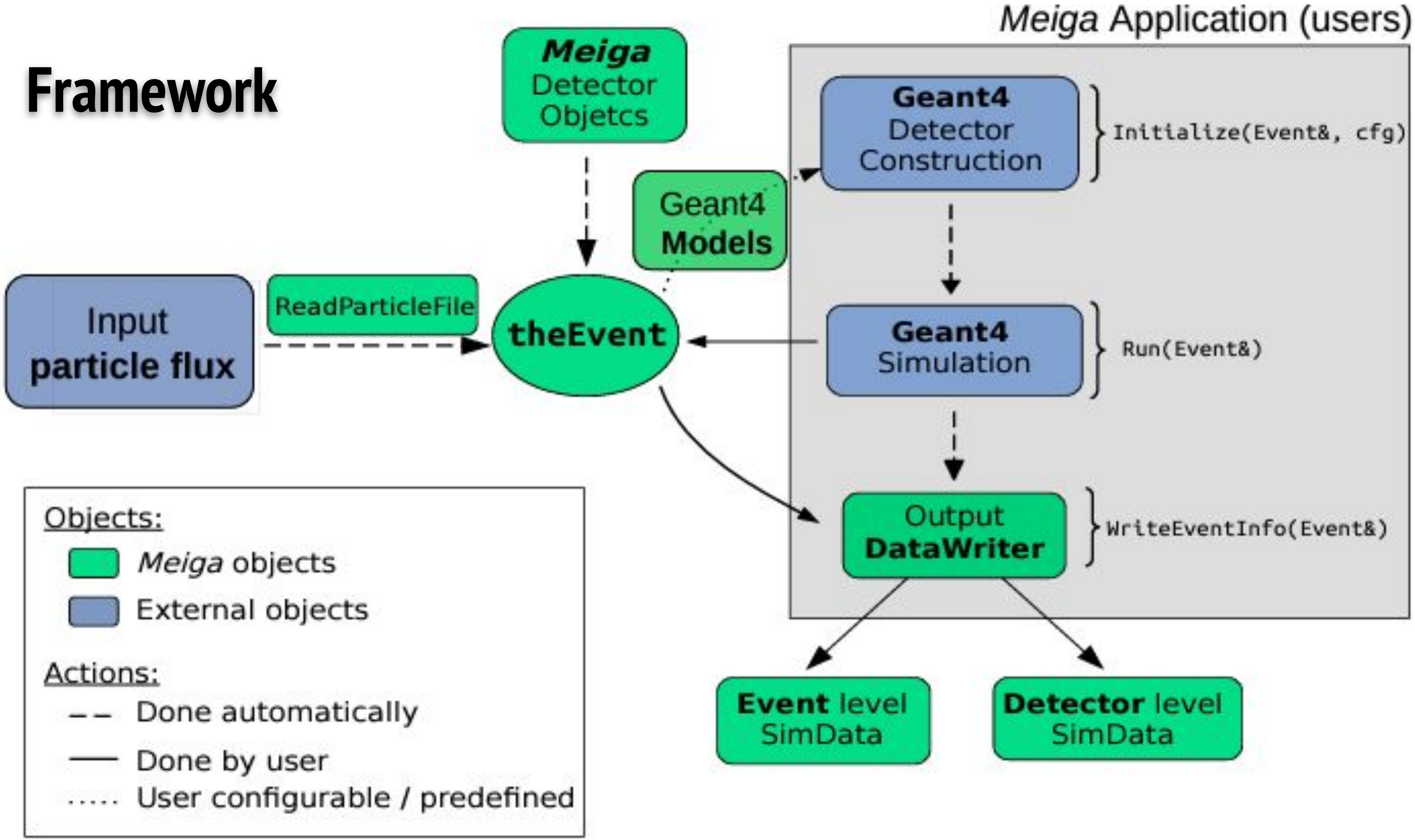
- Ubuntu 22.04.03 LTS
- At /opt dir you will find:
  - root **6.28.06**, from $ `docker pull asoreyh/root:6.28.06`
  - geant4 **10.7.4,** from $ `docker pull asoreyh/geant4:10.7.4`
  - Meiga **2.1.1-dev** commit **39b950**
  - MEGAcmd 1.6.3.2 (for cloud interactions using mega.io CLI)
- coreutils, vim, rsync, screen, python v 3.10.12, …

asoreyh/meiga:dev

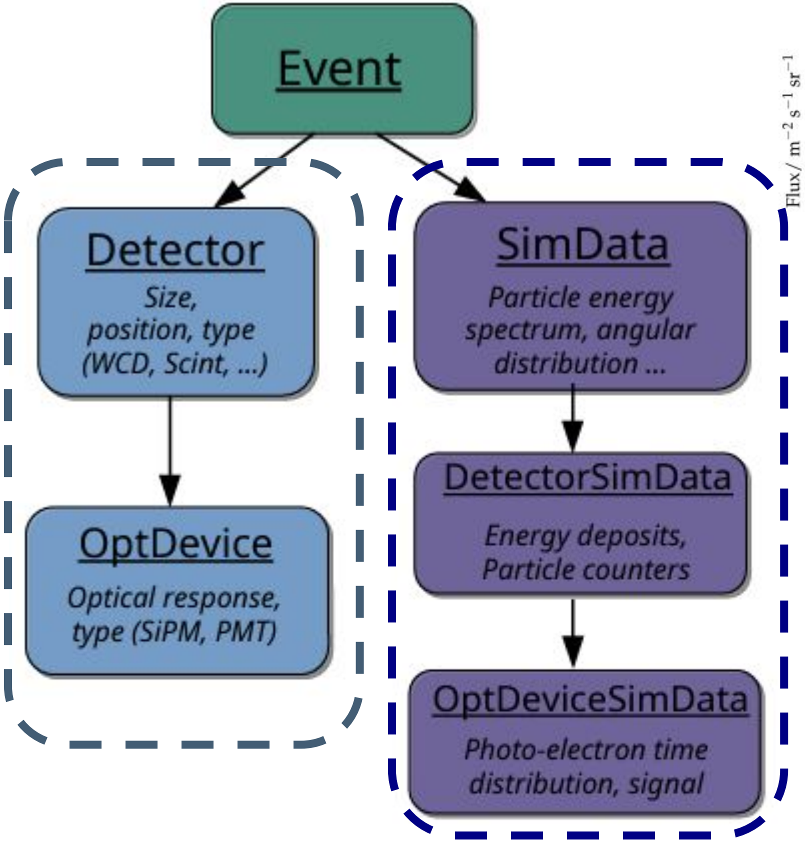
but it will be updated soon...
- Meiga 3 (G4-11 compatible)
- Meiga 2 (G4-10 compatible)
+ numpy, + pandas, + tools

- Ubuntu 22.04.03 LTS
- At /opt dir you will find:
  - root **6.28.06**, from $ `docker pull asoreyh/root:6.28.06`
  - geant4 **10.7.4,** from $ `docker pull asoreyh/geant4:10.7.4`
  - Meiga **2.1.1-dev** commit **39b950**
  - MEGAcmd 1.6.3.2 (for cloud interactions using mega.io CLI)
- coreutils, vim, rsync, screen, python v 3.10.12, …

Framework


Meiga Application (users)
Meiga
Detector
Objetcs
Geant4
Detector
Construction
Initialize(Event&, cfg)
Geant4
Models
Input
particle flux
ReadParticleFile
theEvent
Geant4
Simulation
Run(Event&)
Output
DataWriter
WriteEventInfo(Event&)
Event level
SimData
Detector level
SimData
Objects:
Meiga objects
External objects
Actions:
Done automatically
Done by user
User configurable / predefined

Hierarchy: Event, Detector & SimData



Event

Detector

Size, position, type (WCD, Scint, …)

OptDevice

Optical response, type (SiPM, PMT)

SimData

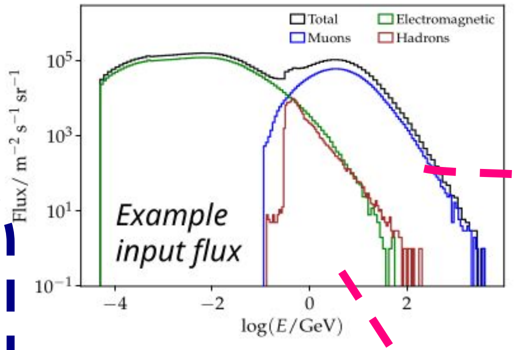
Particle energy spectrum, angular distribution …

DetectorSimData

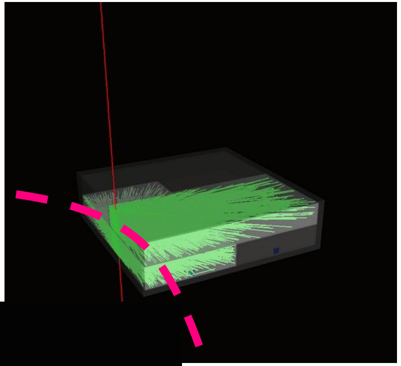
Energy deposits, Particle counters

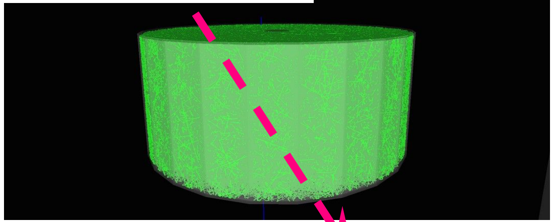
OptDeviceSimData

Photo-electron time distribution, signal

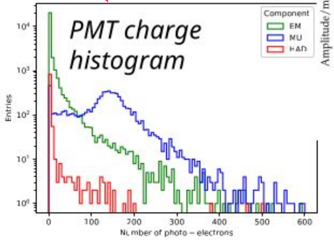


Example input flux

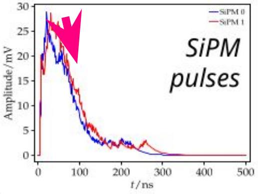




Meiga is supported on a **hierarchical structure** to handle the **detector description** and to access **simulation data** at different levels



PMT charge histogram



SiPM pulses

Applications

Three methods in the *main()* function to manage information at the *Event* object level

```
int main (int arg, char** argv) {
    (…)
    // initialize the Event object
    Event theEvent;
    // create Application class object
    fG4ExSimulator = new G4ExSimulator();
    // set the simulation up using using the configuration file
    fG4ExSimulator->Initialize(theEvent, fCfgFile);
    // run Geant4 simulation
    fG4ExSimulator->RunSimulation(theEvent);
    // write output data
    fG4ExSimulator->WriteEventInfo(theEvent);
    (…)
    return 0;
}
```

Check the examples in the Applications directory

Initialize

Read the config.json file and sets the simulation within this application

```
G4ExSimulator::Initialize(Event& theEvent, string fileName) {
    // JSON parser from boost
    boost::ptree root;
    boost::ptree::read_json(fileName, root);
    // get the particle flux
    fInputFile = root.get<string>("InputFile");
    // read the DetectorList.xml
    fDetectorList = root.get<string>("DetectorList");
    (…)
    // Fills the Event from with the particle flux
    theEvent = ReadParticleFile::EventFileReader(fInputFile);
    // Fills the the Detector Configuration in the Event
    ConfigManager::ReadDetectorList(fDetectorList, theEvent);
}
```

RunSimulation

1. **Checks for particles**

```
SimData& simData = theEvent.GetSimData();
if (!simData.GetTotalNumberOfParticles())
    return false;
```

2. **Construct and sets the Geant4 RunManager**

```
auto fRunManager = G4RunManagerFactory::CreateRunManager();
// Inititalize the DetectorConstruction class
auto fDetectorConstruction = new G4ExDetectorConstruction(theEvent);
fRunManager->SetUserInitialization(fDetectorConstruction);
// Initialize the PhysicsList
fRunManager->SetUserInitialization(fPhysicsList);
// Initialize PrimaryGenerator Action
auto fPrimaryGenerator = new G4ExPrimaryGeneratorAction(theEvent);
fRunManager->SetUserInitialization(fPrimaryGenerator);
```

3. **Loop over particles list or particle beams**

```
for (auto it = simData.GetParticleVector().begin(); it != simData.GetParticleVector().end();
++it) {
    G4ExSimulator::CurrentParticle = *it; // Allows access to primary particle during
simulation process
    fRunManager->BeamOn(1);
}
```

Actions

`G4Actions` is where the user can access data from G4 simulation. To store information in the Meiga classes, just ensure the constructor of the `G4Action` class takes the `Event` object as an argument.

```
// get the pmt object by its ID
auto& pmt = fEvent.GetDetector(detId).GetOptDevice(pmtId);
// check if Cherenkov photon is detected using its energy and the PMT QE
if (!pmt.IsPhotonDetected(energy))
     return continue;
// if we arrive here, photon is detected. Get PMT SimData
auto& pmtSimData = fEvent.GetDetectorSimData(detId).GetOptDeviceSimData(pmtId);
// save photon time
pmtSimData.AddPETimeDistribution(time);
```

Time distribution of photo-electrons in a PMT, from `meiga/src/G4Models/G4MPMTAction.cc`

WriteEventInfo

1. **Retrieve `simData` from the Event**

```
// for accessing Simulated Data at Detector/Event level
SimData& simData = theEvent.GetSimData();
```

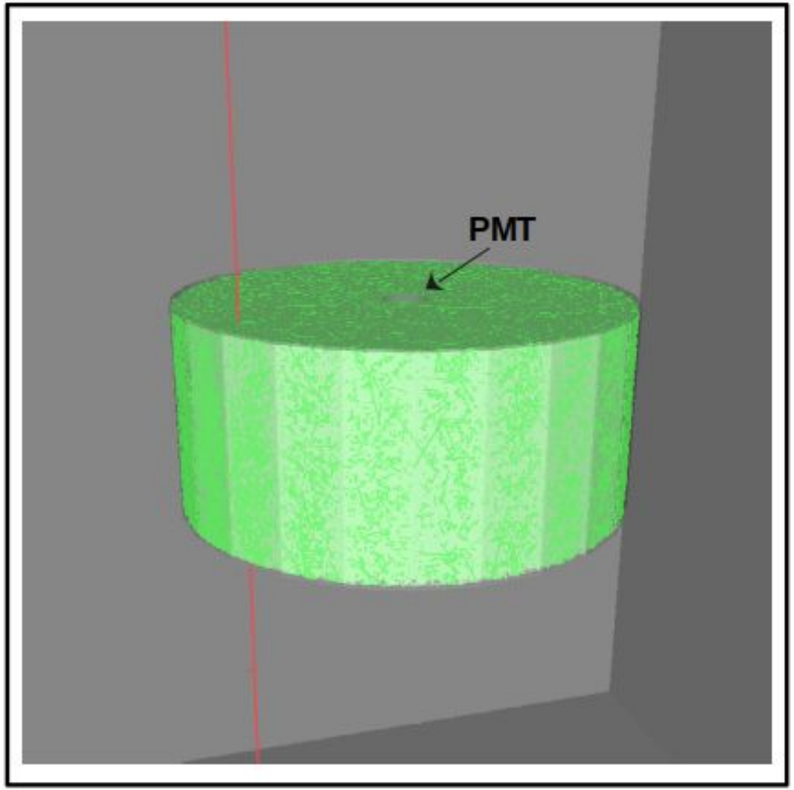
2. **Loops over Detectors and get `DetectorSimData`**

```
// access using the detector ID
for (auto detIt = theEvent.DetectorRange().begin(); … detIt++) {
      int detId = detIt->second.GetId();
      DetectorSimData& detSimData = simData.GetDetectorSimData(detId);

}
```

3. **Gets the PMT SimData for that particular detector**

```
OptDeviceSimData& odSimData = detSimData.GetOptDeviceSimData(odId);

// get the photo-electron time distribution

const auto *peTimeDistribution = odSimData.PETimeDistribution();

// and the number of PE

size_t npe = *(peTimeDistribution)->size()
```

`G4WCDSimulator` : the LAGO detector simulation



A typical LAGO WCD detector

- cylindrical water container of configurable ***r*** radius and ***h*** height
- material: PEHD of configurable ***t*** thickness
- Internal coating: Tyvek
- Hamamatsu R5912 8” PMT, including QE
- Tap water optic characteristics
- Cherenkov production and tracking
- PE production, timing and accounting

The DetectorList.xml

```xml
<?xml version='1.0' encoding='ISO-8859-1'?>

<detectorList>
  <injectionMode type="eCircle">
    <x unit="m"> 0 </x>
    <y unit="m"> 0 </y>
    <z unit="m"> 0 </z>
    <radius unit="m"> 1.5 </radius>
    <height unit="m"> 2 </height>
  </injectionMode>

  <detector id="0" type="eWCD">
    <x unit="cm"> 0.0 </x>
    <y unit="cm"> 0.0 </y>
    <z unit="cm"> 0.0 </z>
    <tankRadius unit="cm"> 98 </tankRadius>
    <tankHeight unit="cm"> 120 </tankHeight>
    <tankThickness unit="mm"> 10 </tankThickness>
  </detector>
</detectorList>
```

eCircle: particles are injected in a circle of radius <radius> and at height <height>.
eHalfSphere: particles are injected over a semi-sphere of radius <radius> and origin <x>, <y>, <z>.
eVertical: particles are injected vertically at a fixed position (given by <x>, <y>, <z>).
eFromFile: injection coordinates are taken from the input file.

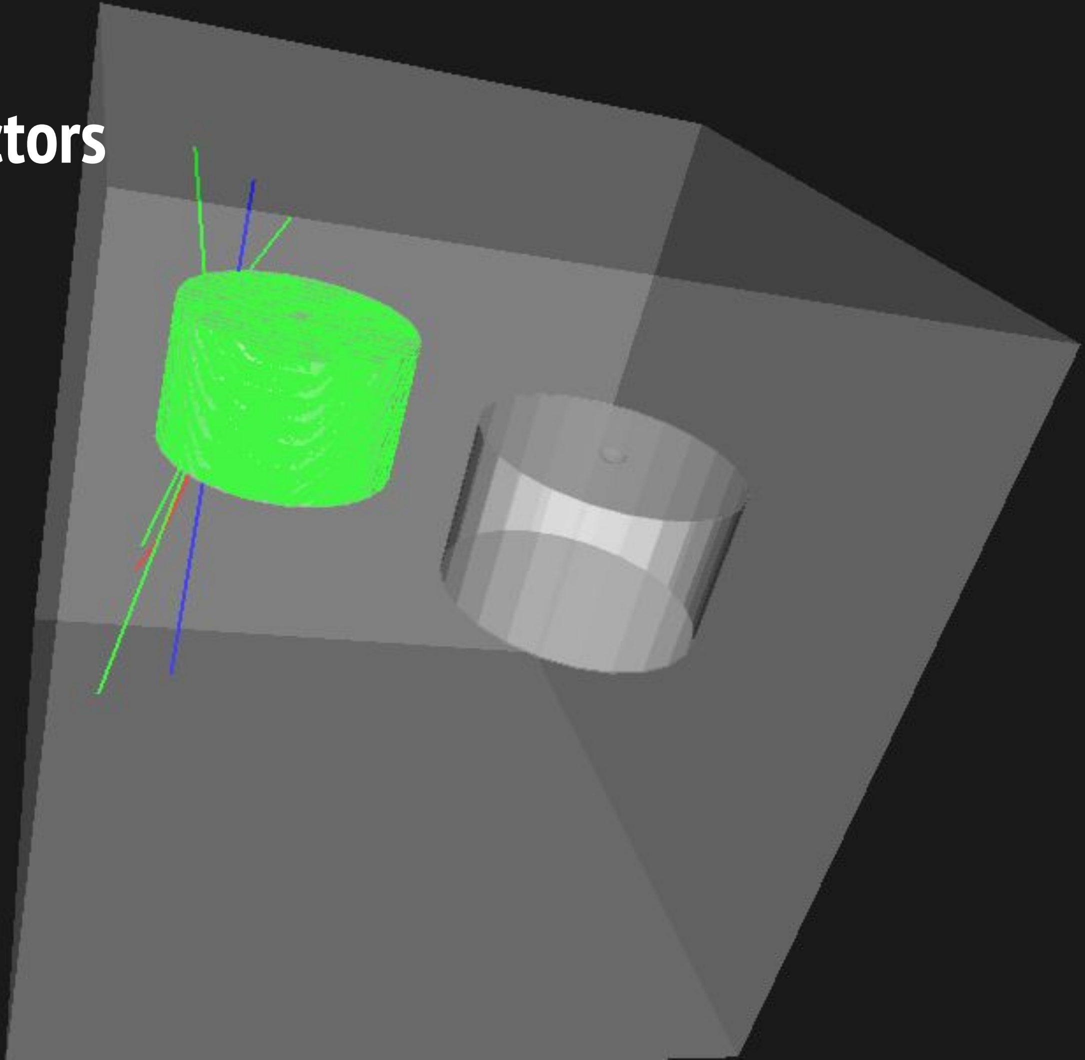
WCD id “0” of radius <tankRadius> and height <tankHeight> and thickness <tankThickness> located at position <x>, <y> and <z>.

Default are used if no magnitude is given.
Defaults can be changed in the DetectorProperties.xml file

repeat this block for each detector in an array.

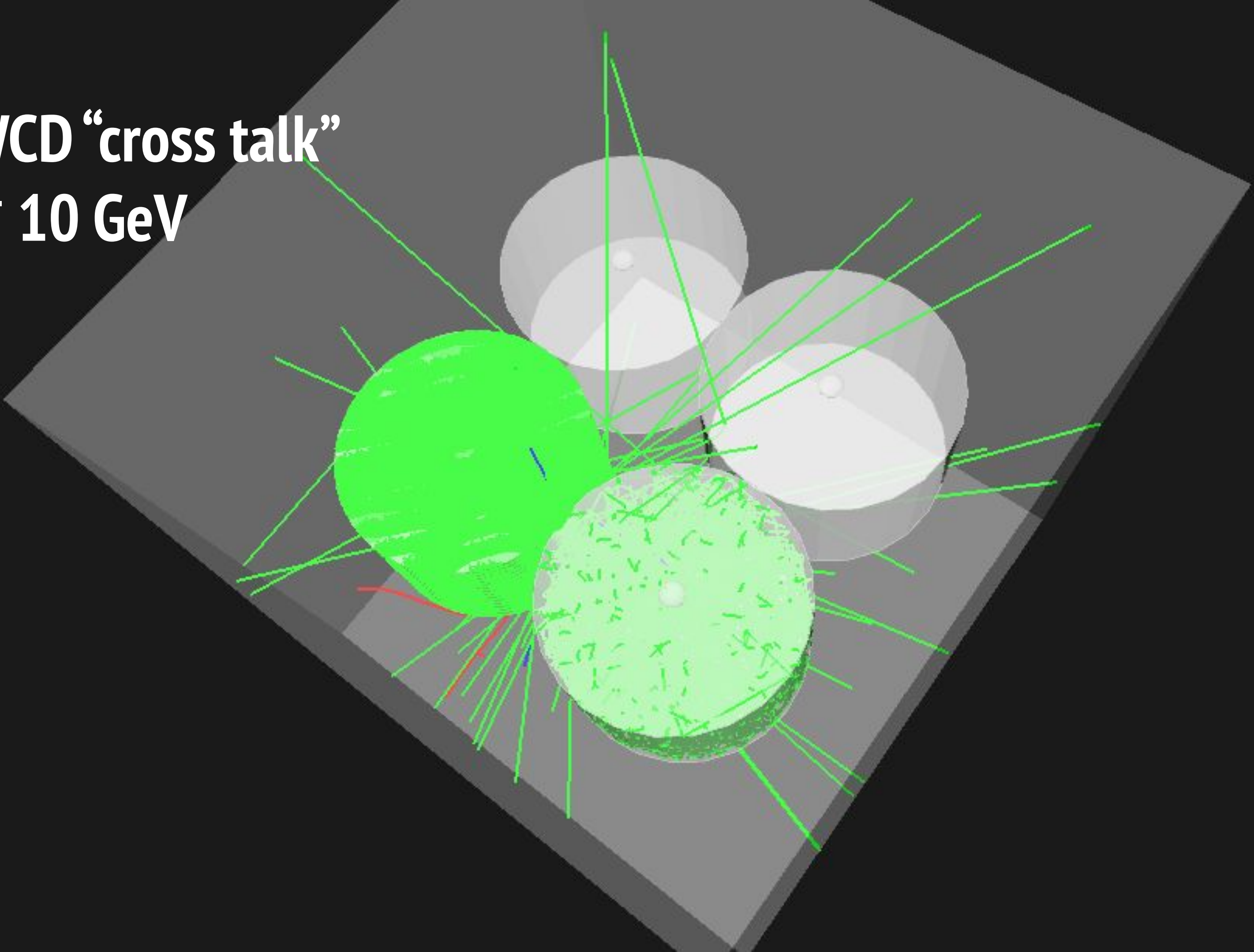
id should be alphanumeric but unique

Multiple detectors



WCD “cross talk”

e$^-$ 10 GeV



The config.json - input

```
"Input" :
{
        "Mode" : "UseARTI",
        "InputFileName" : "../../../src/Documentation/SampleFlux/vertical_muon.txt"
},
```

Mode: can be UseARTI or UseEcoMug for selecting between the two input types
InputFileName: path to input file in case UseARTI is chosen.
InputNParticles: number of muons to be injected in case UseEcoMug is chosen.

If UseARTI is chosen, InputFileName should be an ASCII 12-col ARTI-like:

```
CorsikaId px py pz x y z shower_id prm_id prm_energy prm_theta prm_phi
```

e.g. for a vertical muon+ with p=1 GeV/c in the -z direction (position will depend on injection mode)

```
5 0 0 1 0 0 0 0 0 0 0 0
```

The config.json - output

```
"Output" :
{
        "OutputFile" : "./output.json",
        "CompressOutput" : true,
        "SaveInput" : true,
        "SavePETimeDistribution" : false,
        "SaveEnergy" : true,
        "SaveCharge" : true
},
```

OutputFile: path to the output file.
CompressOutput: enable compression of output file (.gz).
SavePETimeDistribution: save photo-electron time distributions for each injected particle.
SaveComponentsPETimeDistribution: save photo-electron time distributions by particle component (electromagnetic, muons, hadrons).
SaveEnergy: save energy deposits in the detector.
SaveCounts: save particle counters (for scintillator bars).
SaveComponentsEnergy: save energy deposits in the detector by particle component.

The config.json - Detectors

```
"DetectorList" : "./DetectorList.xml",
"DetectorProperties" : "./DetectorProperties.xml",
```

DetectorList: path to the detector list file.
DetectorProperties: path to file with the default detector properties.

The config.json - Simulation

```
"Simulation" :
{
        "SimulationMode" : "eFast",
        "GeoVisOn" : false,
        "TrajVisOn" : false,
        "CheckOverlaps" : false,
        "Verbosity" : 1,
        "RenderFile" : "VRML2FILE",
        "PhysicsName" : "QGSP_BERT_HP"
}
```

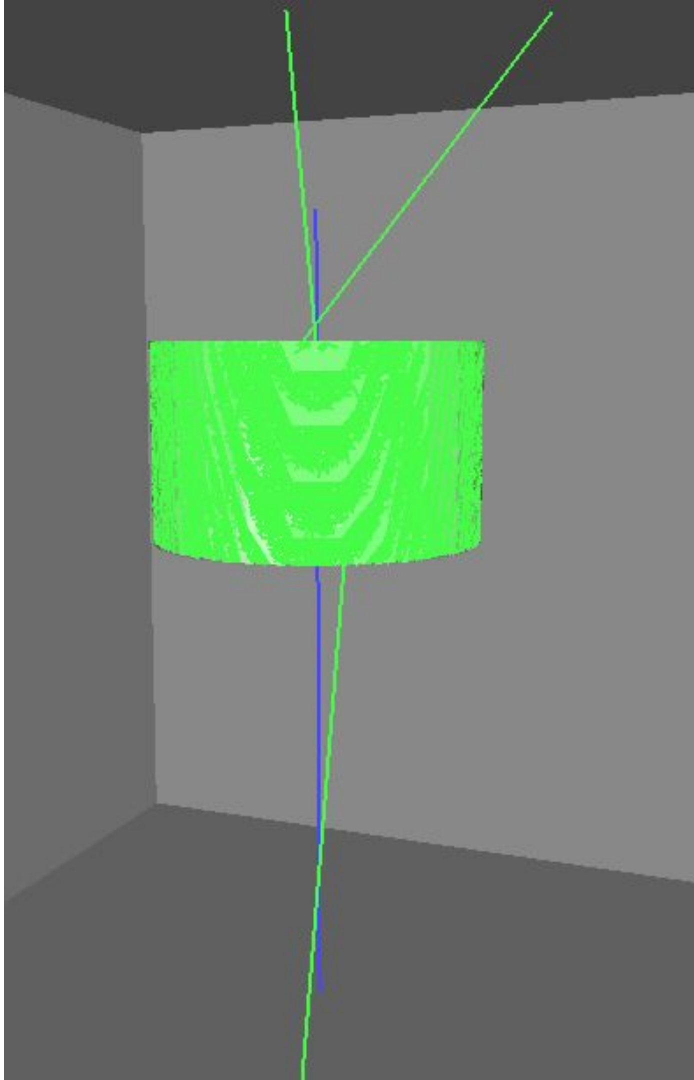
SimulationMode: eFull by default. Some applications allow eFast for fast simulation
GeoVisOn and **TrajVisOn**: if True, a render file with geometry (and particle trajectories) of the detector is generated.
CheckOverlaps: if True detects overlaps between detector volumes.
Verbosity: level of Geant4 verbosity output. Warning, >1 could produce large outputs
RenderFile: type of render file. For now, only .wrl files are allowed.
PhysicsName: name of the physics list.

VEM (Vertical Equivalent Muon)

Vertical (and central) Equivalent Muon

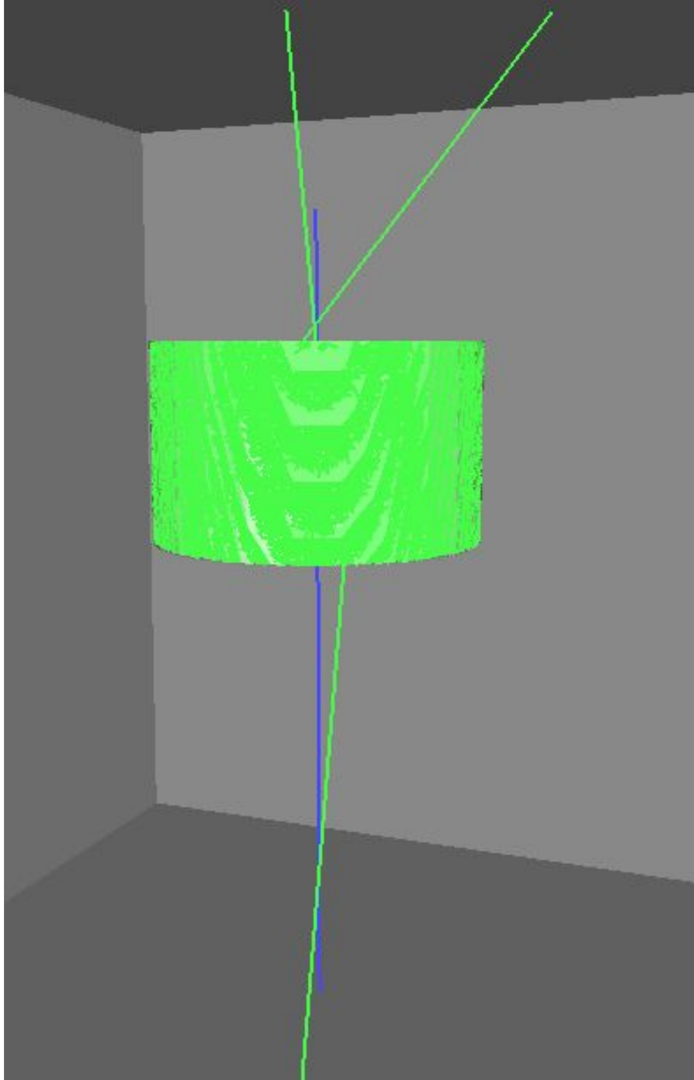
Calibration magnitude:

1 VEM = <tbd> photoelectrons = (2·h) MeV



The output.json(.gz)

```
{
  "Output": {
    "Event_0": {
      "InputFlux": {
        "ID": -13,
        "Position": [0.0, 0.0, 2000.0],
        "Momentum": [0.0, 0.0, -5000.0]
      },
      "Detector_0": {
        "EnergyDeposit": 199.62353614726368,
        "OptDevice_0": {
          "Charge": 670
        }
      }
    }
  },
  "DetectorList": {
    "Detector_0": {
      "ID": 0,
      "Name": "eWCD",
      "Position": [0.0, 0.0,0.0],
      "OptDeviceList": [0]
    }
  }
}
```



Let’s do it together

warning: docker images only include defaults

```
$ docker run -it asoreyh/meiga:dev bash
# cd /opt/meiga/build/Applications/G4WCDSimulator
# ll
total 1052
drwxr-xr-x 3 root root    4096 Nov 12  2023 ./
drwxr-xr-x 8 root root    4096 Nov 12  2023 ../
drwxr-xr-x 3 root root    4096 Nov 12  2023 CMakeFiles/
-rw-r--r-- 1 root root     383 Nov 12  2023 DetectorList.xml
-rwxr-xr-x 1 root root 1033392 Nov 12  2023 G4WCDSimulator*
-rw-r--r-- 1 root root     557 Nov 12  2023 G4WCDSimulator.json
-rw-r--r-- 1 root root   15349 Nov 12  2023 Makefile
-rw-r--r-- 1 root root    2273 Nov 12  2023 cmake_install.cmake
```

Our main files

edit the DetectorList.xml

```xml
<?xml version='1.0' encoding='ISO-8859-1'?>

<detectorList>
  <injectionMode type="eVertical">
    <x unit="m"> 0 </x>
    <y unit="m"> 0 </y>
    <z unit="m"> 2 </z>
    <radius unit="m"> 1.5 </radius>
    <height unit="m"> 2 </height>
  </injectionMode>

  <detector id="0" type="eWCD">
    <x unit="cm"> 0.0 </x>
    <y unit="cm"> 0.0 </y>
    <z unit="cm"> 0.0 </z>
    <tankRadius unit="cm"> 98 </tankRadius>
    <tankHeight unit="cm"> 120 </tankHeight>
    <tankThickness unit="mm"> 10 </tankThickness>
  </detector>
</detectorList>
```

Vertical muon starting at z=2 m

Use your own values

edit the input file

```
# vim muon.txt
```

```
5 0 0 1 0 0 0 0 0 0 0 0
```

1 muon+ with Pz=5 GeV/c
(add one line per secondary for multiple particles)

edit the G4WCDSimulator.json

```
"InputFileName" : "./muon.txt"
(...)
"DetectorList" : "./DetectorList.xml",
(...)
"GeoVisOn" : true,
"TrajVisOn" : true,
(...)
```

our file

our detector

enable vis output

run the simulation (~ 7 s in my notebook)

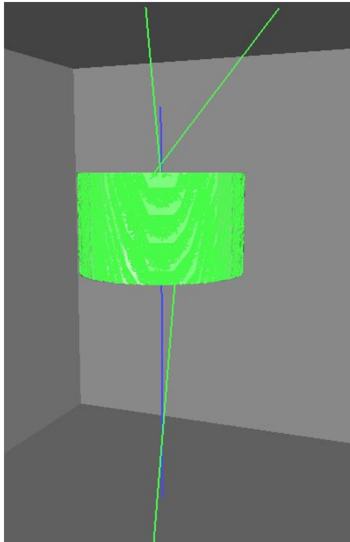
```
#  ./G4WCDSimulator -c G4WCDSimulator.json
# ll

(...)
-rw-r--r-- 1 root root   34403 Nov 17 19:16 g4_00.wrl
-rw-r--r-- 1 root root      24 Nov 17 19:09 muon.txt
-rw-r--r-- 1 root root     177 Nov 17 19:16 output.json.gz
(...)
```

open a new terminal and copy the .wrl file

```
$ asoreyh@caronte:~$ docker ps
CONTAINER ID   IMAGE              COMMAND    CREATED          STATUS          PORTS     NAMES
7b1d505586d8   asoreyh/meiga:dev  "bash"     18 minutes ago   Up 18 minutes             beautiful_albattani

$ docker cp  beautiful_albattani:/opt/meiga/build/Applications/G4WCDSimulator/g4_00.wrl .
Successfully copied 2.05kB to /home/asoreyh/.
$ view3dscene g4_00.wrl
```

if the viewer is not installed (in ubuntu):

```
$ sudo apt install view3dscene
```



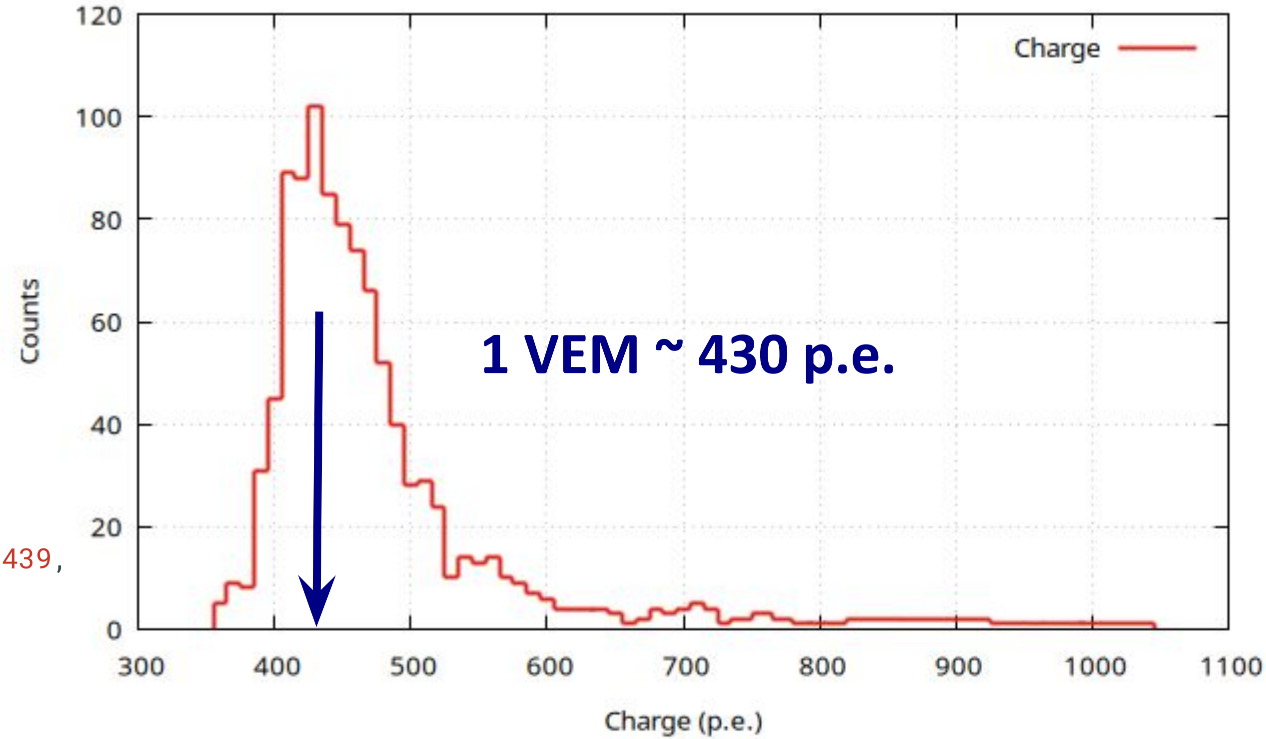
Problem 1: VEM charge distribution

Let’s use, say, N=1000 vertical muons for calculating the VEM charge equivalent. For doing this:

1. Edit the muon.txt accordingly and run the simulation again.
2. Edit the config file for turning off the visualization outputs (Set GeoVision and TrajVision to false), and use the “eFast” mode
3. Run the simulation. It should take ~ 3 minutes
4. From the host terminal, copy the output.json.gz
5. Get the histograms of charge and deposited energy

Get the charge

```
(...)
"Event_3": {
  "InputFlux": {
    "ID": -13,
    "Position": [0.0, 0.0, 2000.0],
    "Momentum": [0.0, 0.0, -5000.0]
  },
  "Detector_0": {
    "EnergyDeposit": 195.0362666897439,
    "OptDevice_0": {
      "Charge": 475
    }
  }
},
(...)
```



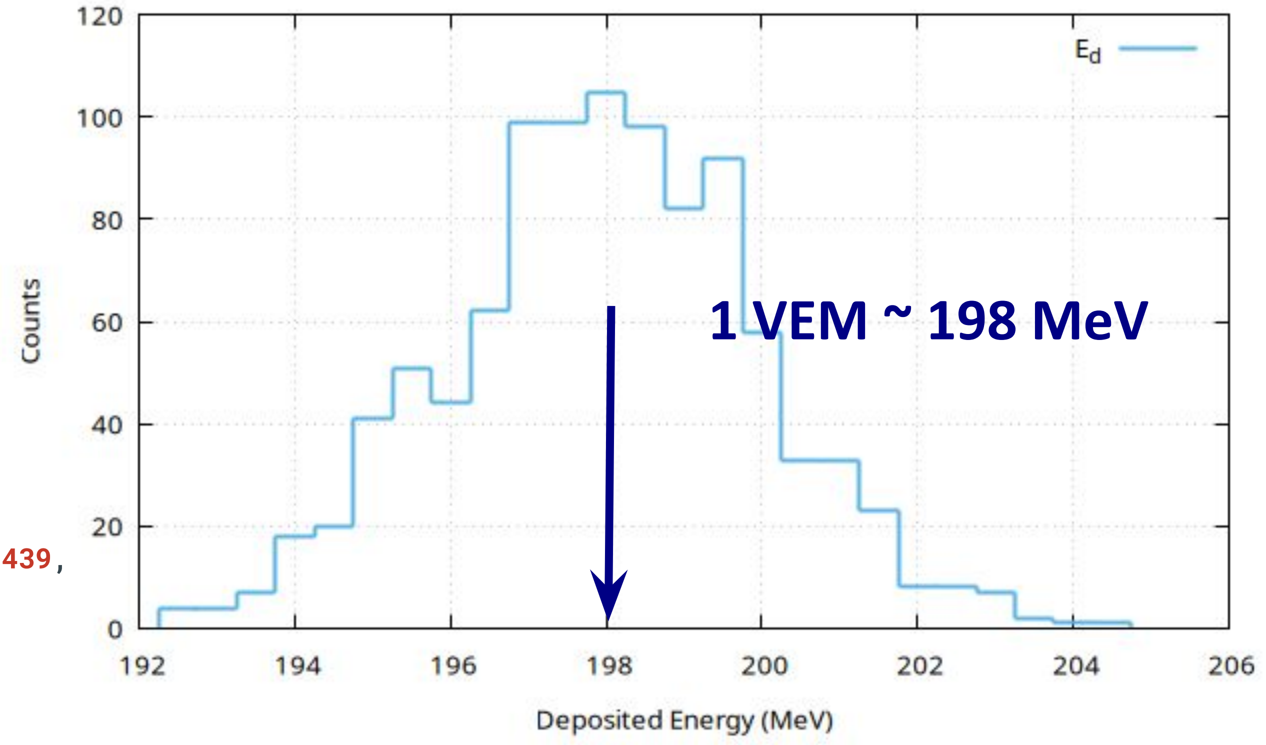
1 VEM ~ 430 p.e.

For example, in your terminal:

```
$ zcat output.json.gz | jq | grep "Charge" | awk '{b=5; print int($2/b)*b}' | sort -n | uniq -c > charge.dat
```

Get E_d

```
(...)
"Event_3": {
  "InputFlux": {
    "ID": -13,
    "Position": [0.0, 0.0, 2000.0],
    "Momentum": [0.0, 0.0, -5000.0]
  },
  "Detector_0": {
    "EnergyDeposit": 195.0362666897439,
    "OptDevice_0": {
      "Charge": 475
    }
  }
},
(...)
```



1 VEM ~ 198 MeV

For example, in your terminal:

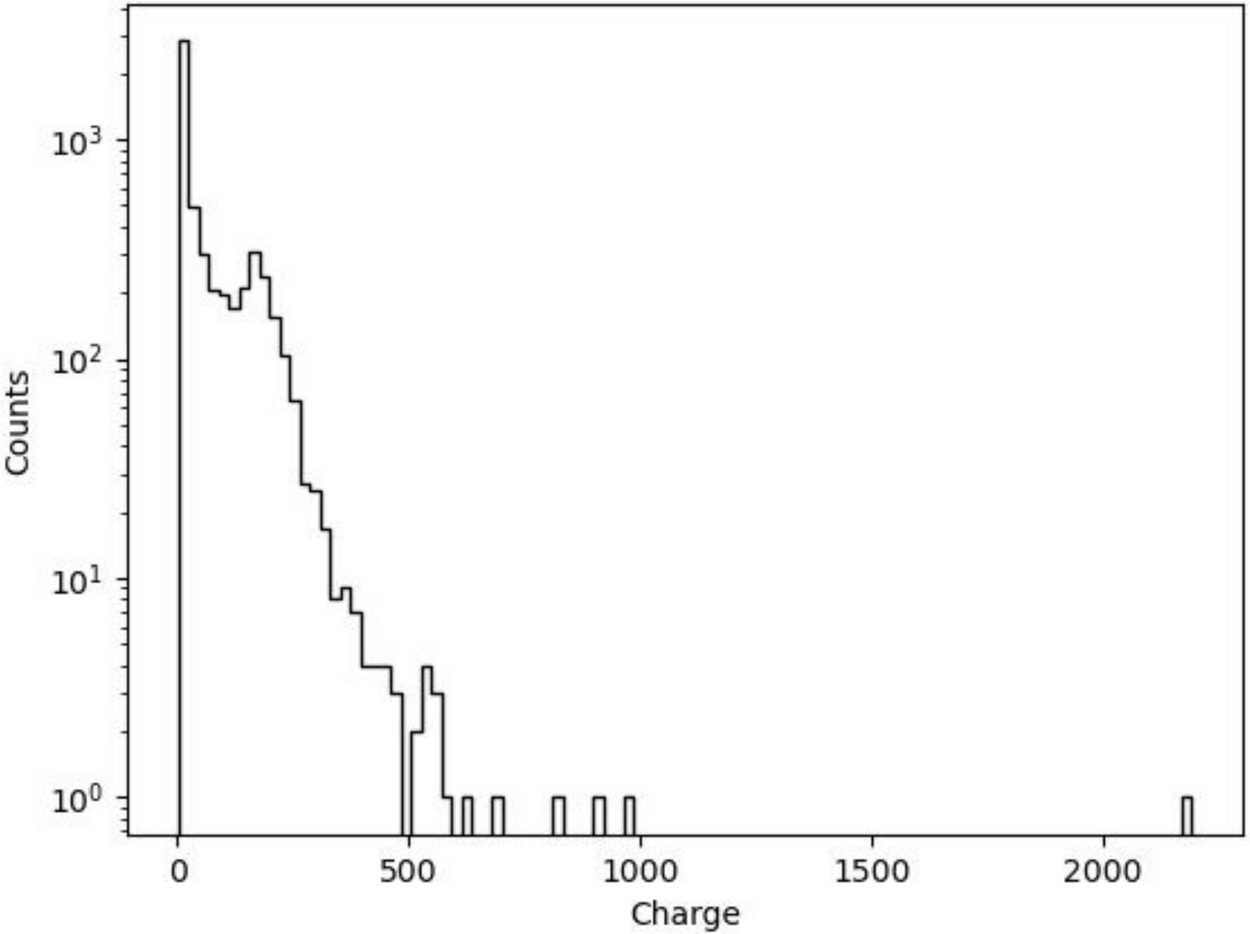
```
$ zcat output.json.gz | jq | grep "EnergyDeposit" | awk '{b=0.5; print int($2/b)*b}' | sort -n |
uniq -c > ed.dat
```

Problem 2: Sample flux (at home)

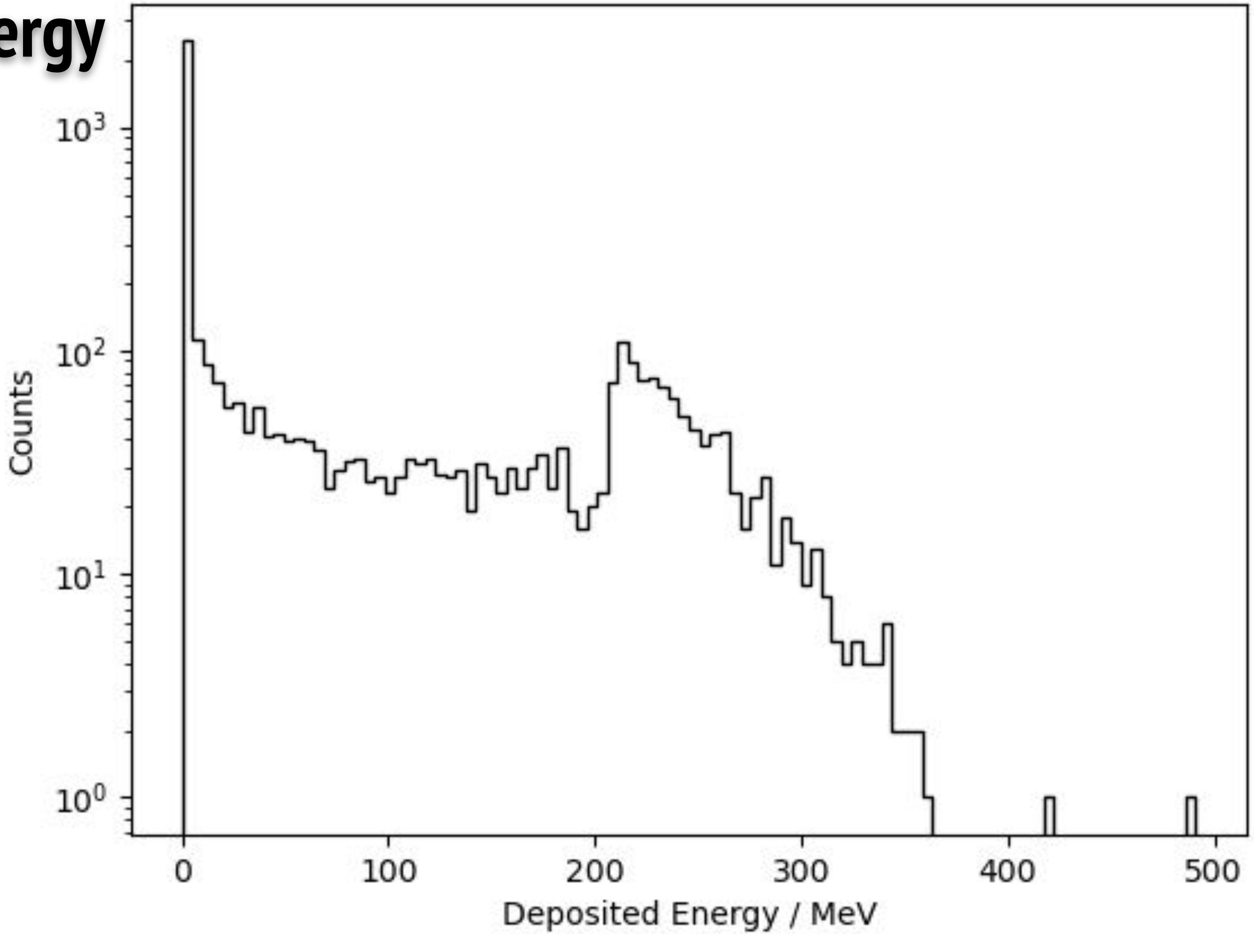
In **`meiga/src/Documentation/SampleFlux/`**, we included a sample for the BGA background calculated using ARTI (~34k part). Let’s see the expected charge histogram:

1. Edit the config file for using this input (be sure “eFast” is selected): ‘/opt/meiga/src/Documentation/SampleFlux/salida_bga_30.shw’
2. Edit the DetectorList.xml for using the “eCircle” mode: radius and height should be at least 2x the tankRadius and tankHeight (the larger the better, but… time)
3. Run the simulation. It should take ~ 6 minutes
4. From the host terminal, copy the output.json.gz
5. Get the histograms of charge and deposited energy

Charge

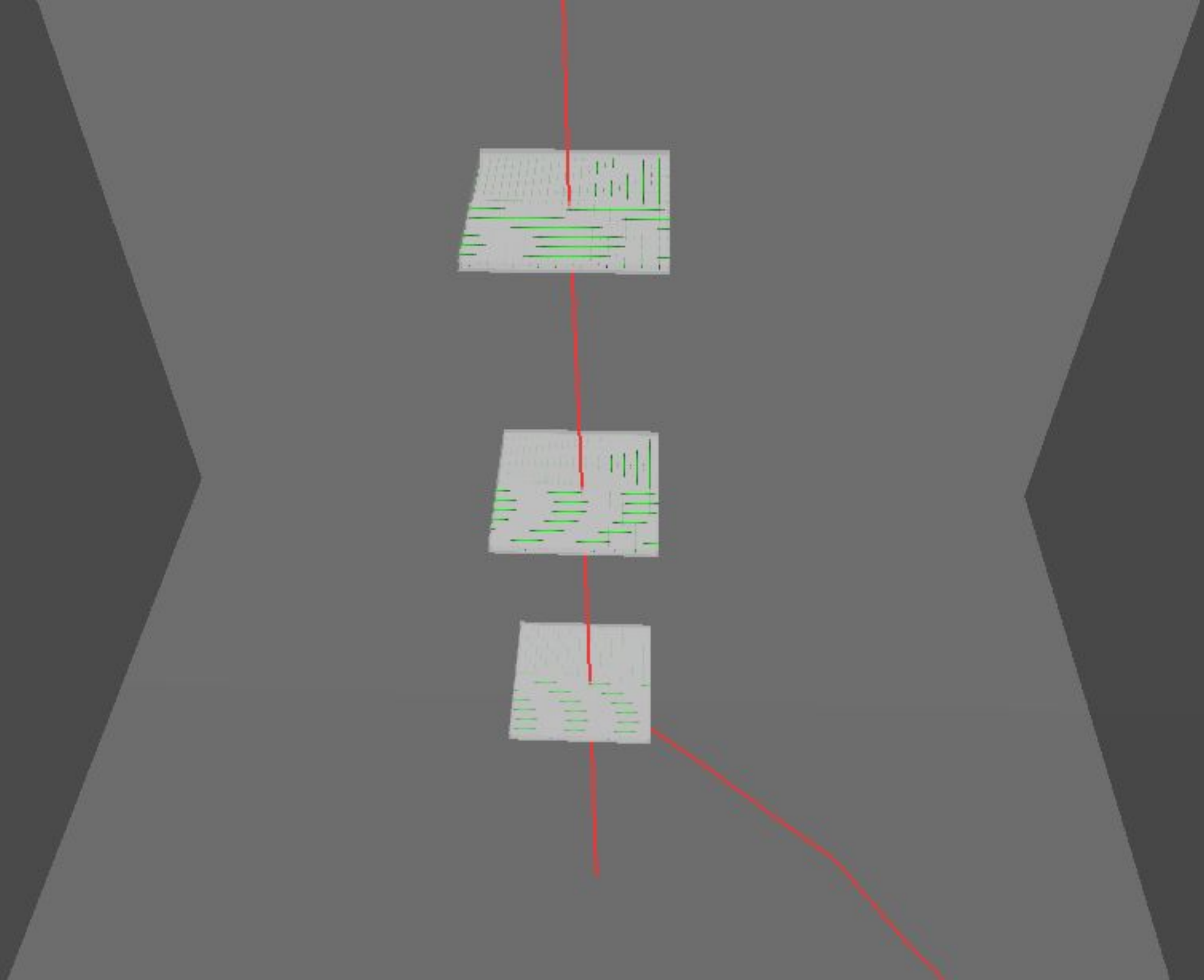


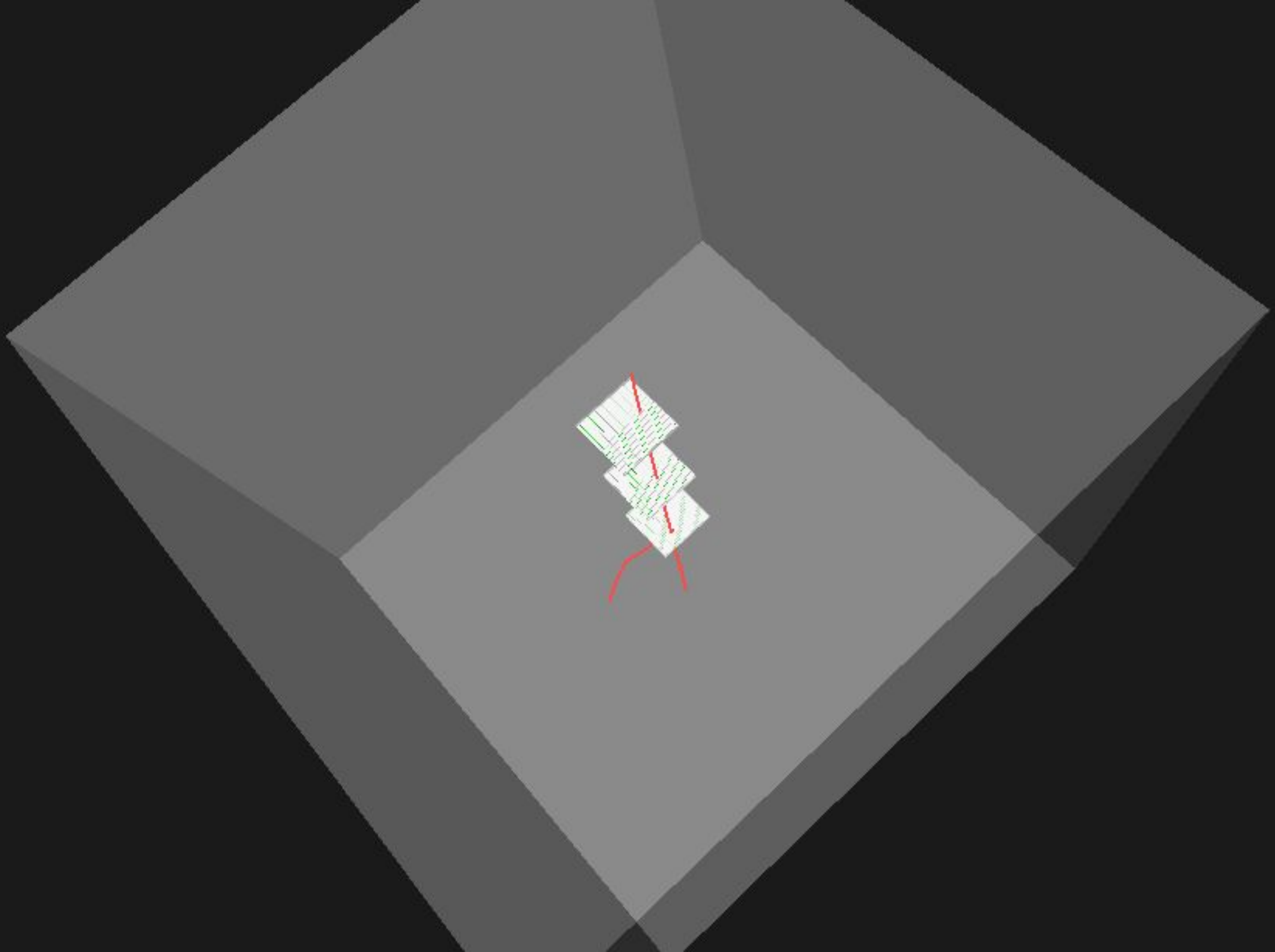
Deposited energy



Problem 3: Explore G4HodoscopeSimulator (at home)

1. Take a look to the files in the directory: what similarities do you observe with G4WCDSimulator? what differences?
2. From what you learnt up to now and obtain a visualization of a muon event detected by the default hodoscope
3. Change the default hodoscope for a Mute2.0-like hodoscope and obtain a new visualization
4. Get the panels hits for a 10k muon flux for Mute2.0






That's all Folks!

GOBIERNO
DE ESPAÑA

MINISTERIO
DE CIENCIA
E INNOVACIÓN

Ciemat
Centro de Investigaciones
Energéticas, Medioambientales
y Tecnológicas

Comisión Nacional
de Energía Atómica

this talk

contact me

LAGO INDICA
EOSC, MEIGA y Muongrafía

Hernán Asorey[1,2] for the LAGO Collaboration

BGA, Colombia, 20/Nov/2024

Thanks!

hernanasorey@cnea.gob.ar