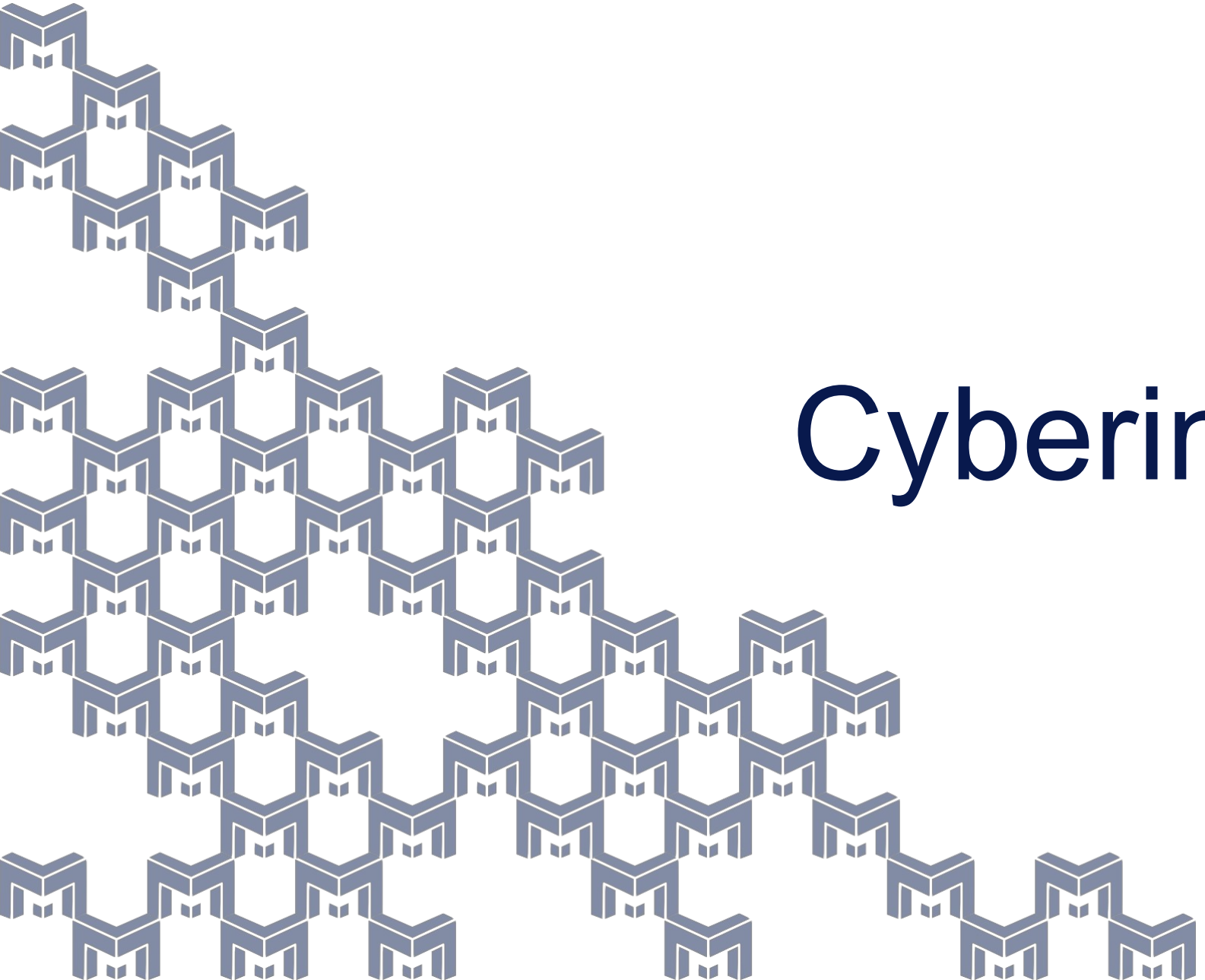
MUSES review: collaboration and cyberinfrastructure

Nico Yunes
Physics Professor and Director of ICASU
University of Illinois Urbana-Champaign

Andrew Manning
Senior Research Scientist
National Center for Supercomputing Applications



Cyberinfrastructure

What does “cyberinfrastructure” (CI) actually mean???

- **Cyberinfrastructure*** consists of systems, data and information management, advanced instruments, visualization environments, and people, **all linked together by software and advanced networks to improve scholarly productivity** and enable knowledge breakthroughs and discoveries not otherwise possible.

Figure 2 shows the distinction between the systems that are elements of cyberinfrastructure and cyberinfrastructure as a system that includes humans as an integral part. The critical issue here is the astronomer (in this case Scott Michael of Indiana University) is able to interactively watching the output of simulations running remotely at the Pittsburgh Supercomputing Center. Data from the simulation are written to the Data Capacitor at IU as simulations are being performed, and **the researcher can either allow simulations to run to completion or cancel jobs when simulations produce nonsensical results. In the latter case the astronomer is able to adjust parameters and restart jobs.** This is an excellent example of cyberinfrastructure in that it involves supercomputers, data storage systems, and visualization systems all linked by networks (in this case the TeraGrid network [16]) and middleware (in this case Globus and the Common TeraGrid Software Stack [17]), **with a researcher as an integral component of the application execution**.

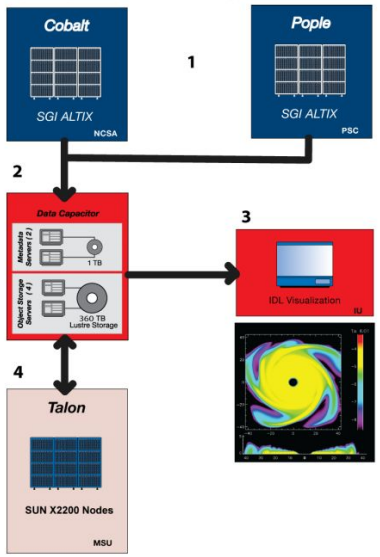
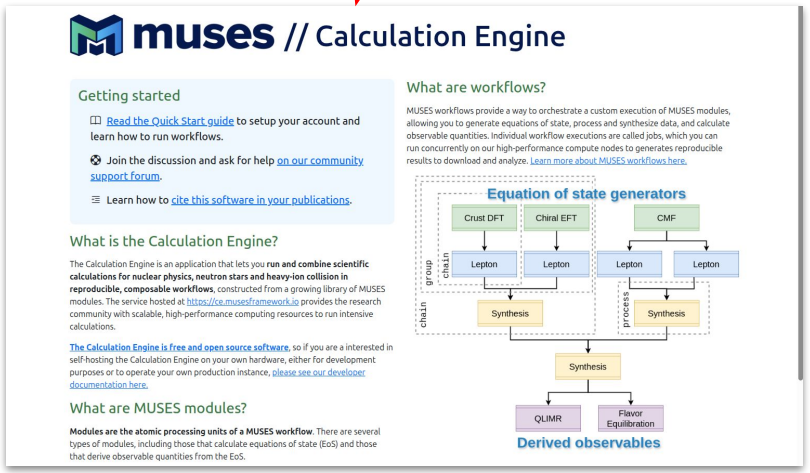


Figure 2. An astronomer (3) views the results of simulations occurring at PSC’s Pople and NCSA’s Cobalt supercomputers (1) as they are being written to IU’s Data Capacitor (2). Further analysis of the data that have been written (2) takes place on MSU’s Talon supercomputer.

* Indiana University.
What is Cyberinfrastructure?
October 2010
DOI:10.1145/1878335.1878347

What does “cyberinfrastructure” (CI) actually mean???

- **Cyberinfrastructure** consists of systems, data and information management, advanced instruments, visualization environments, and people, **all linked together by software and advanced networks to improve scholarly productivity** and enable knowledge breakthroughs and discoveries not otherwise possible.



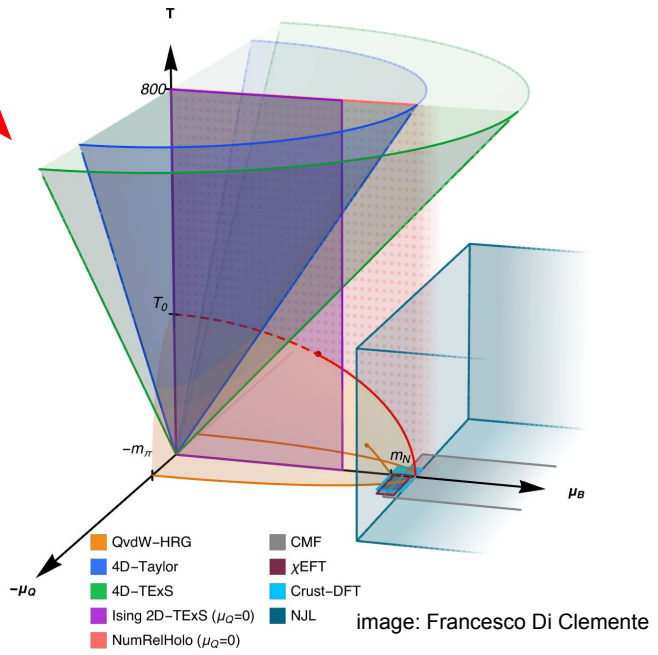


image: Francesco Di Clemente

Look in mirror

Some terminology (i.e. words to use instead of "~~codes~~")

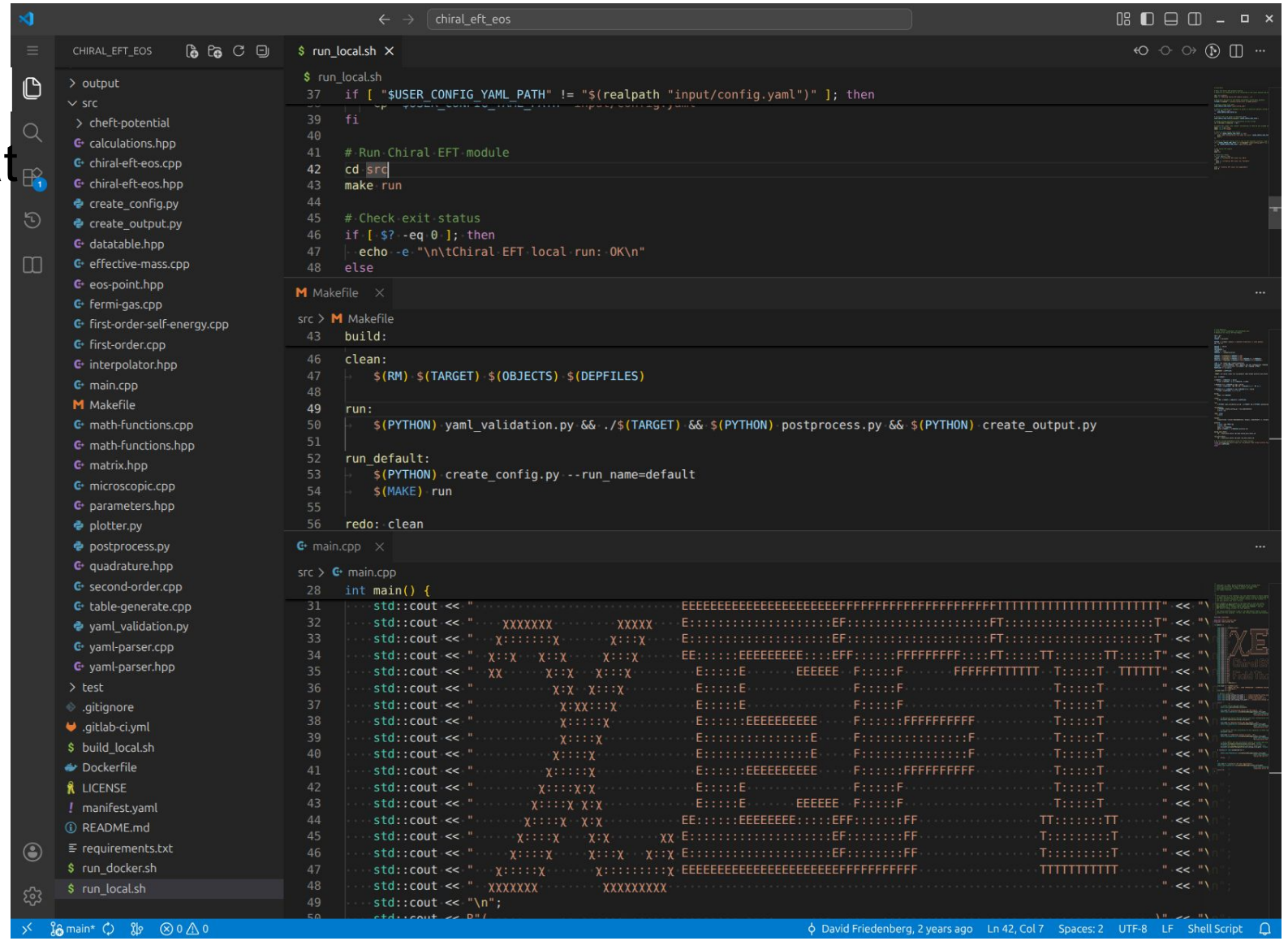
- **Script** - a file whose statements are executed line-by-line to take specific actions or generate particular outputs.
- **Module*** - a **collection of related functions** and classes that are **reusable**. Multiple scripts or other modules may import the same module to reduce redundant code.
- **Library** - a **collection of modules** and scripts with a **well-defined interface** designed to provide a specific set of capabilities.
- **Application** - similar to a **library**, but is executed to provide a service or perform a set of tasks.
- **Package** - a self-contained unit of software that can be installed, typically via a package manager (apt, brew, pip, npm). A package can install a **library** or an **application**.

* The word "module" is a bit overloaded in computer science. A "MUSES module" is actually more of a "library" as defined above; however, we coined "MUSES module" based on the "modular" components that extensibly define the CE capability.

Saying "codes" for software is like saying "letters" to refer to anything composed of words in any format for any purpose. ~ a wise and ancient Chinese and/or Greek philosopher … Confuciates?

Example: Chiral EFT “MUSES module”

There is a top-level **script** that invokes several executables written in Python and C++.

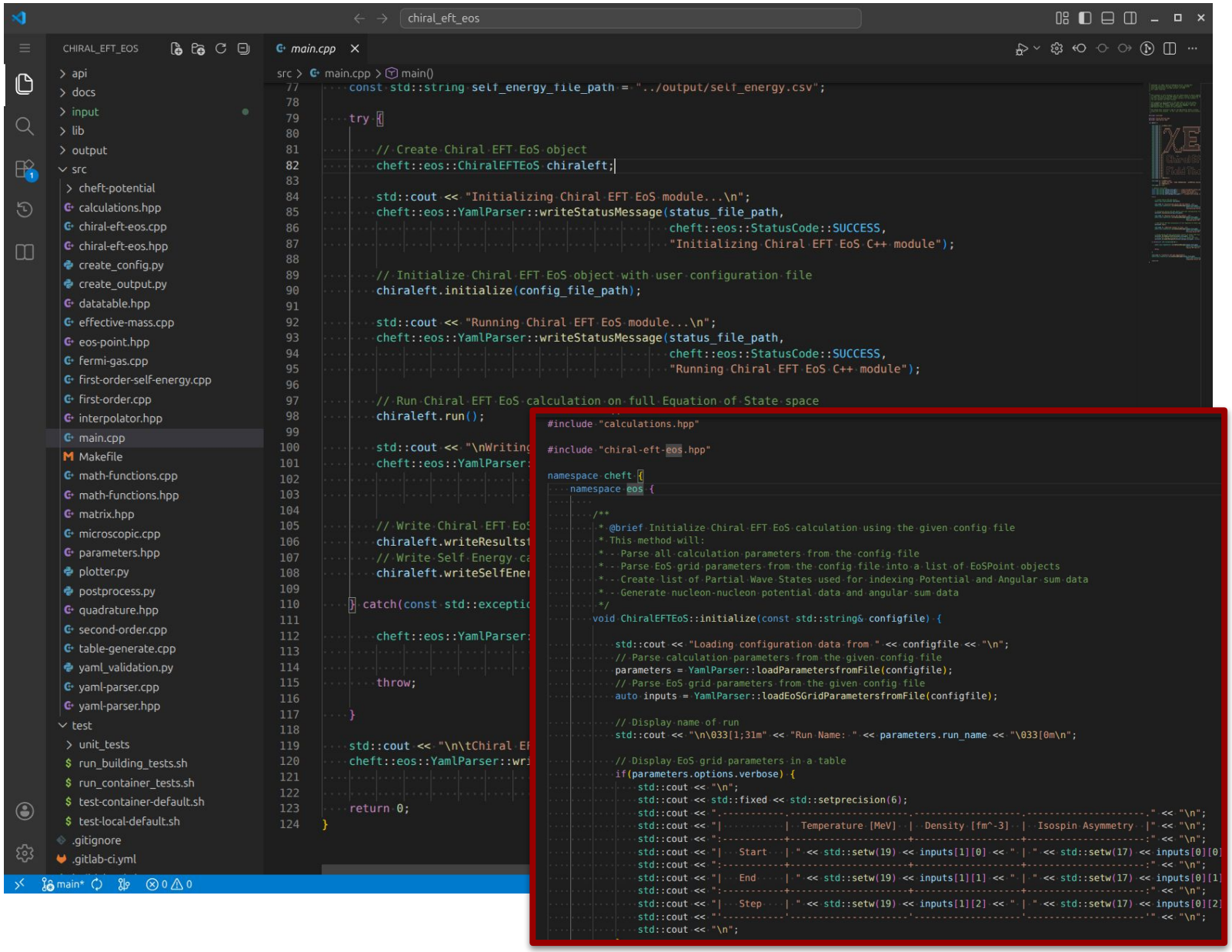


Example: Chiral EFT “MUSES module”

The core of the script is a call to function `main()`, which imports several **modules*** including `chiral-eft-eos.cpp` that define the `ChiralEFTEoS` class.

Together, these modules comprise a **library** that independent software could import for use.

This library could be **packaged** according to a standard like Spack or Python for easy installation on target environments.



* Technically, “C++ modules” comparable to “Python modules” were only introduced in 2020 in C++20, but the header (.h) and implementation (.cpp) files manifest the concept.

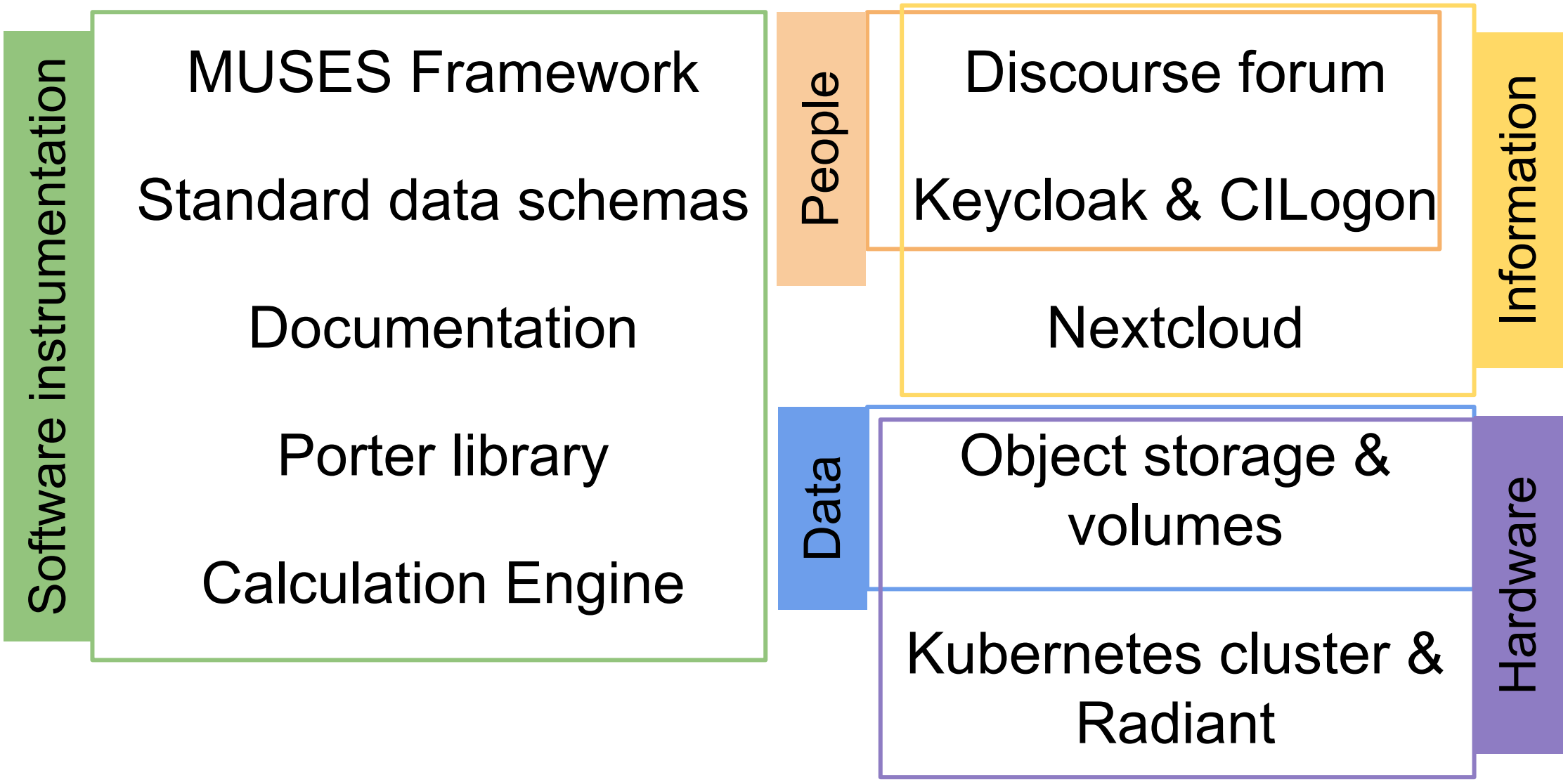
“As modules are a recent addition and compiler vendors were notably slow to develop module support, most third-party libraries are still offered only as headers. However, some popular libraries have implemented module support. These include […] nlohmann-json (a JSON library, as nlohmann.json), […] and various libraries from Boost C++ Libraries (a general purpose C++ library, as boost.*).”

Wikipedia contributors. (2026, May 3). Modules (C++). In Wikipedia, The Free Encyclopedia. Retrieved 13:52, May 7, 2026, from https://en.wikipedia.org/w/index.php?title=Modules_(C%2B%2B)&oldid=1352248815

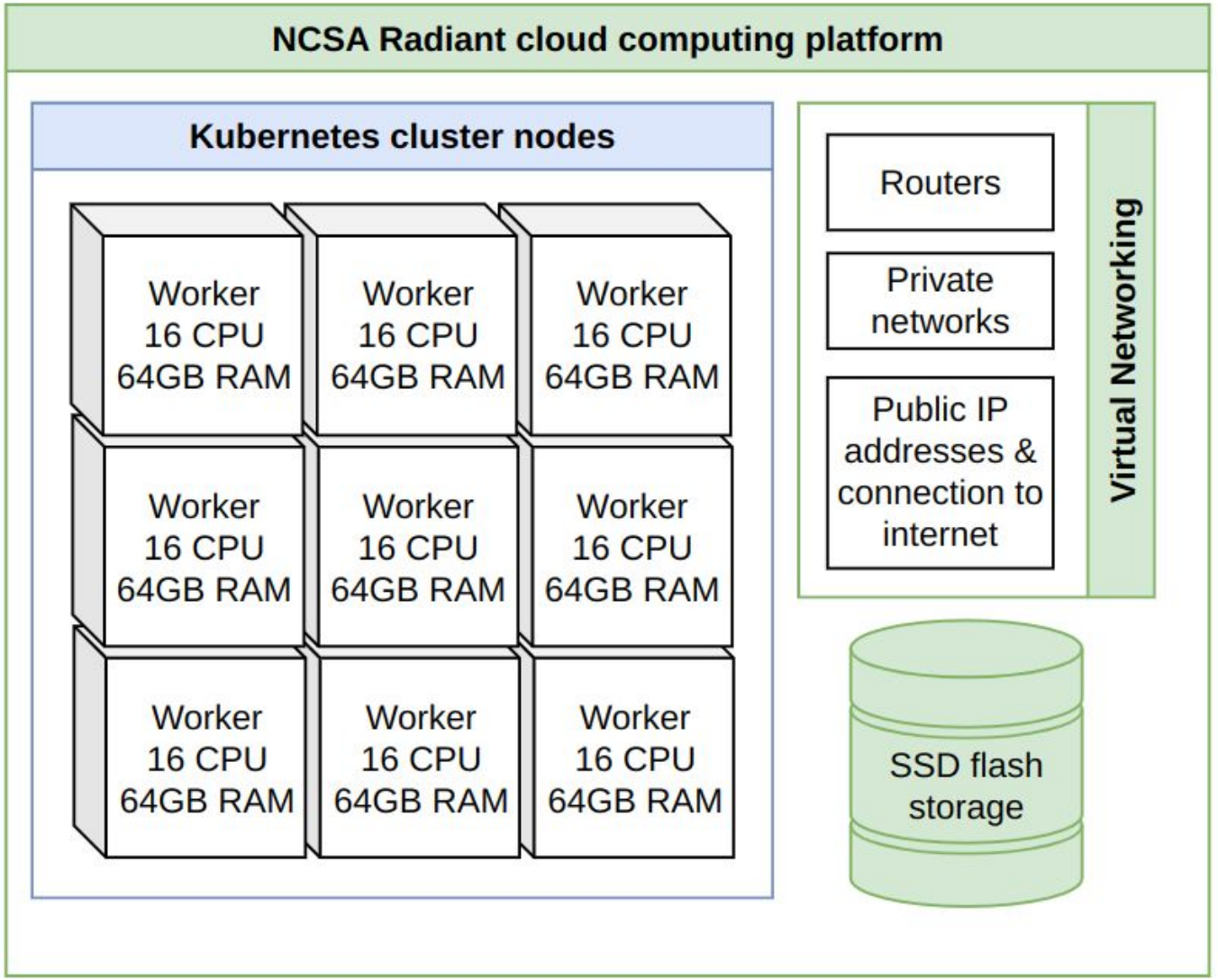
One more term: framework

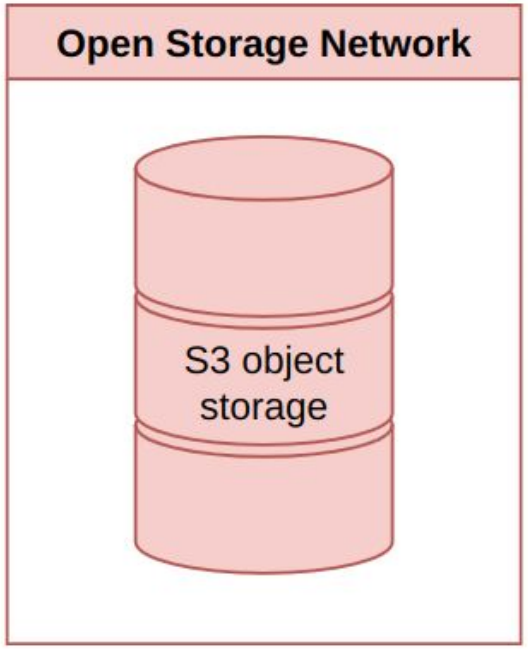
- In computer programming, a **software framework** is “an abstraction in which software, providing generic functionality, can be **selectively changed by additional user-written code**, thus providing application-specific software. It **provides a standard way to build and deploy applications** and is a **universal, reusable software environment** that provides particular functionality as part of a larger software platform to facilitate the development of software applications, products and solutions.”
- The **MUSES framework** defines a standardized way to package otherwise independent software libraries that calculate equations of state and related physical quantities, such that they can be integrated into data processing workflows in a larger application.
- A **MUSES module** is a **library** that has been **packaged** according to the **MUSES framework**.

What comprises MUSES cyberinfrastructure?



Physical & virtual infrastructure





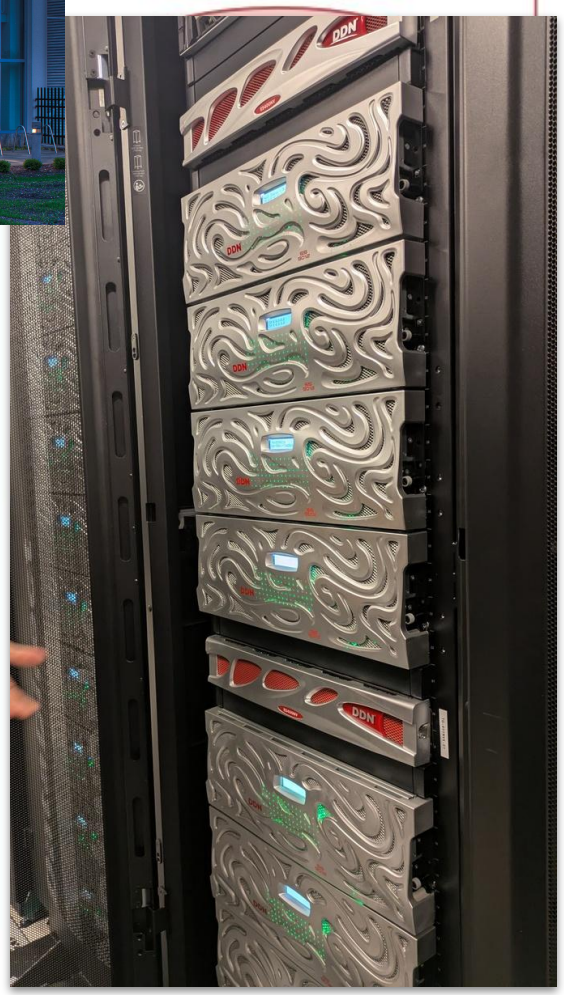
OSN is a bulk storage system provided by a federated network of institutions via NSF ACCESS

Physical & virtual infrastructure

NCSA

NCSA Radiant cloud co

Open Storage Network



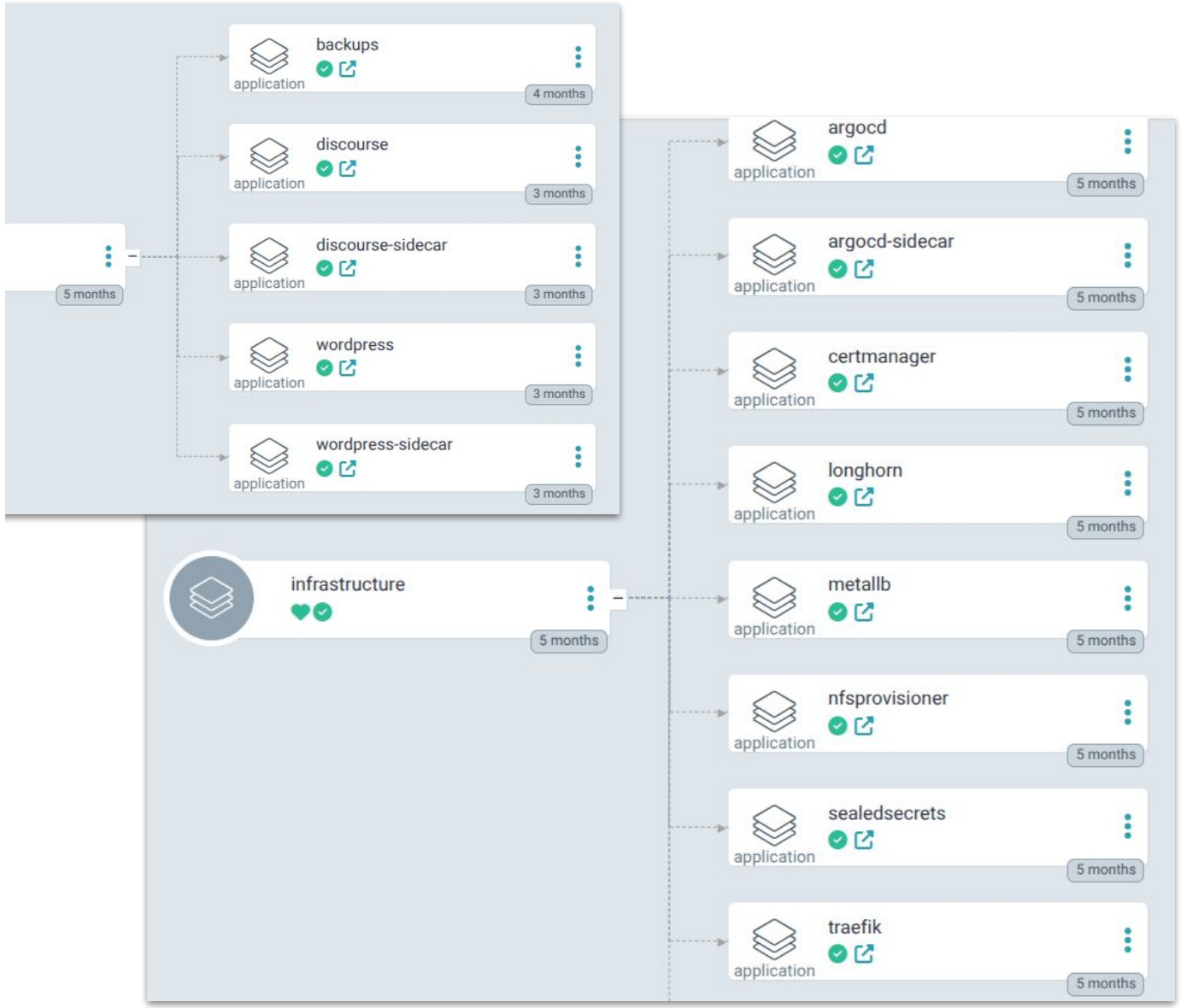
Software infrastructure

Foundational:

- Deployment system
- Secrets management
- Persistent storage provisioner
- Web ingress & TLS certificates
- Identity & access management
- Backup system

Services:

- Cloud file storage
- Discussion forum
- MUSES project website
- JupyterHub
- Calculation Engine



What is the Calculation Engine (CE)?

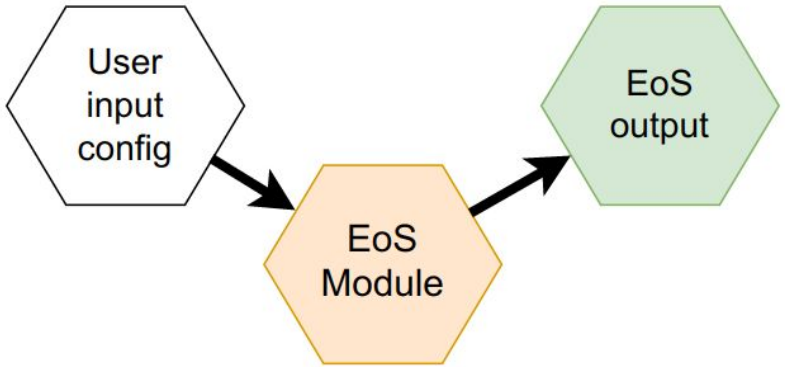
The Calculation Engine is a software **application** that

- manages user-submitted **jobs**
  - a **job** is the execution of one or more modules in a data processing **workflow**.
- manages the movement of data between subsystems
- serves data for download
- enforces access control
- organizes and stores information in a structured database
- tracks data provenance (does it really?)

Workflows

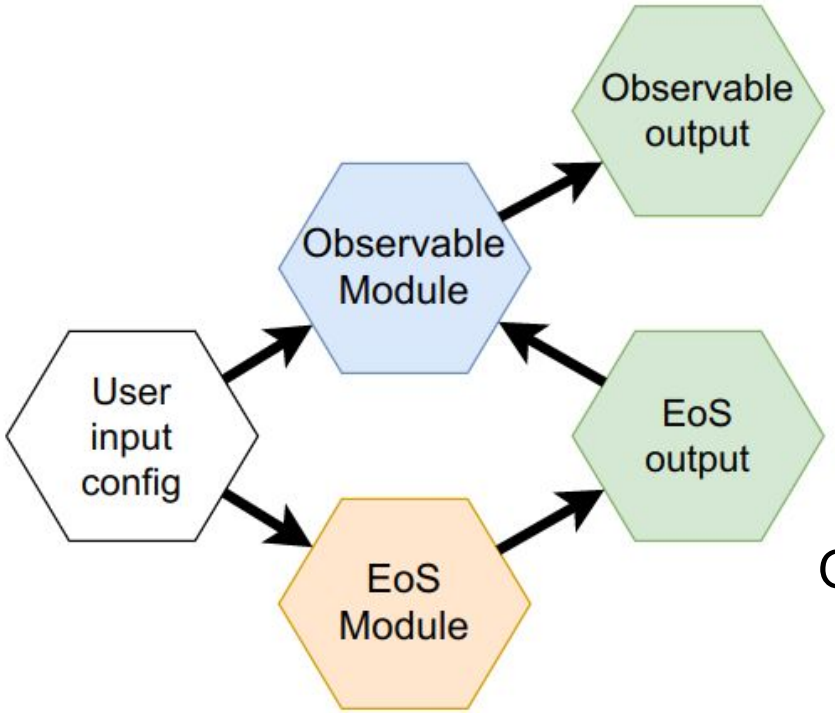
A workflow is a processing sequence of one or more **MUSES modules**, in which output of one module is used as the input to the subsequent modules.

The simplest case is a single module execution such as calculating an equation of state.

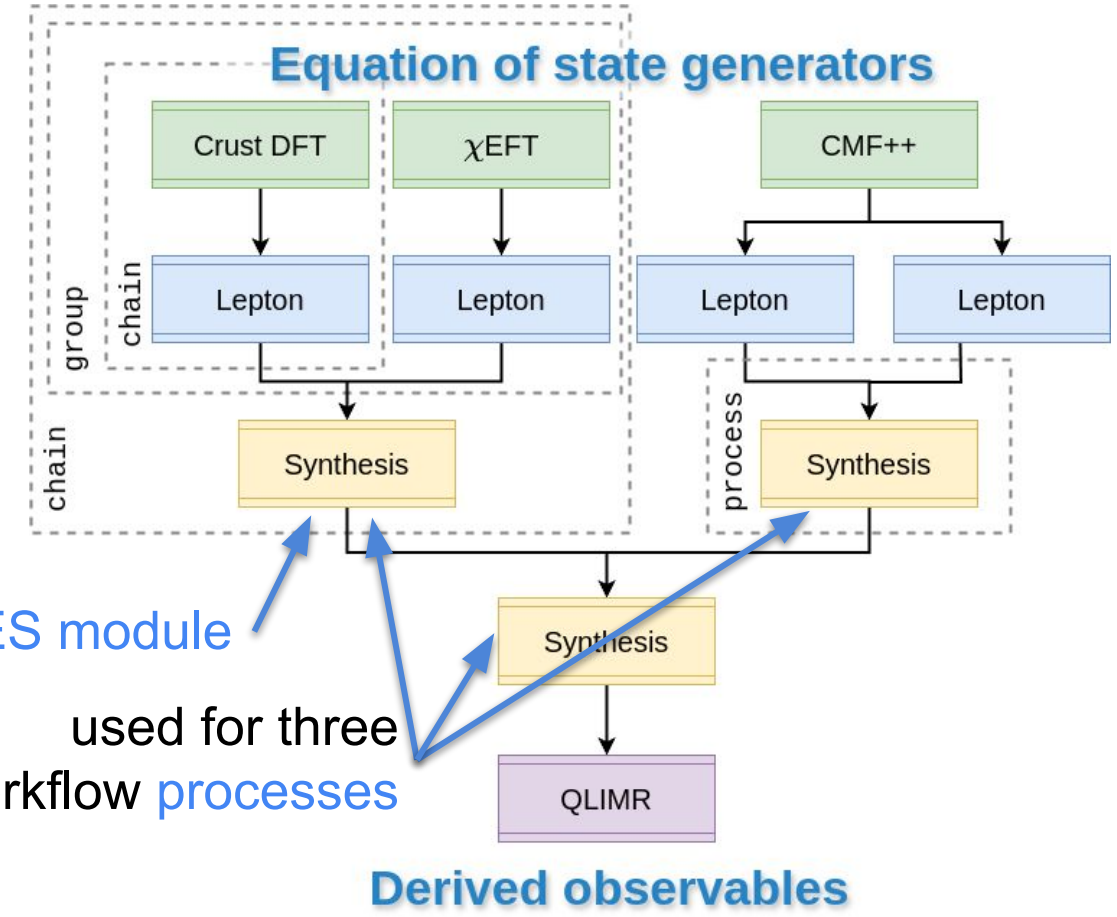


Workflows

The most common multi-module sequence will generate an EoS and then calculate some physical observables using a second module.



More complex workflows are supported as well (arbitrary DAGs limited only in number of processes).



One MUSES module used for three workflow processes

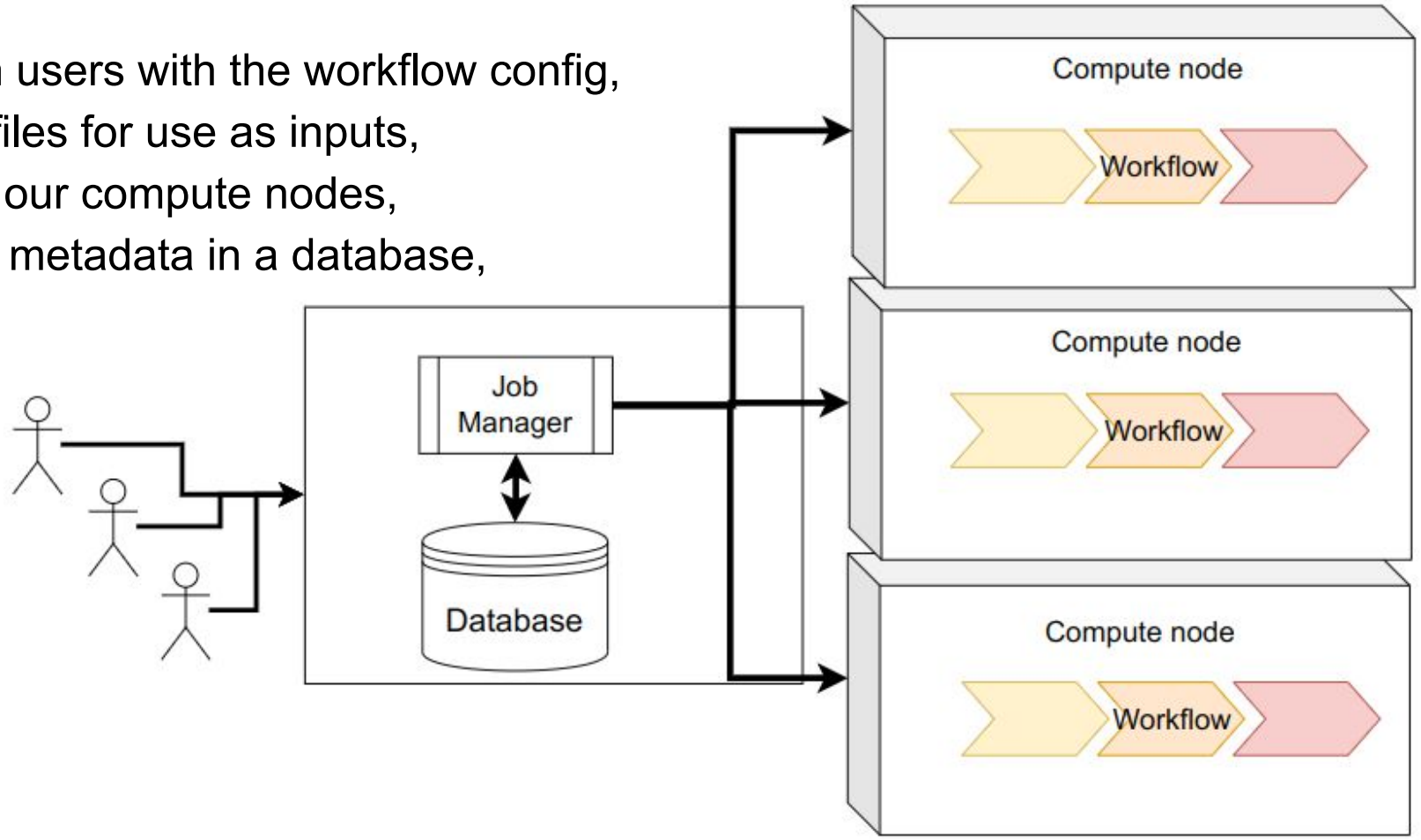
Data flow

From the perspective of data flow, the CE

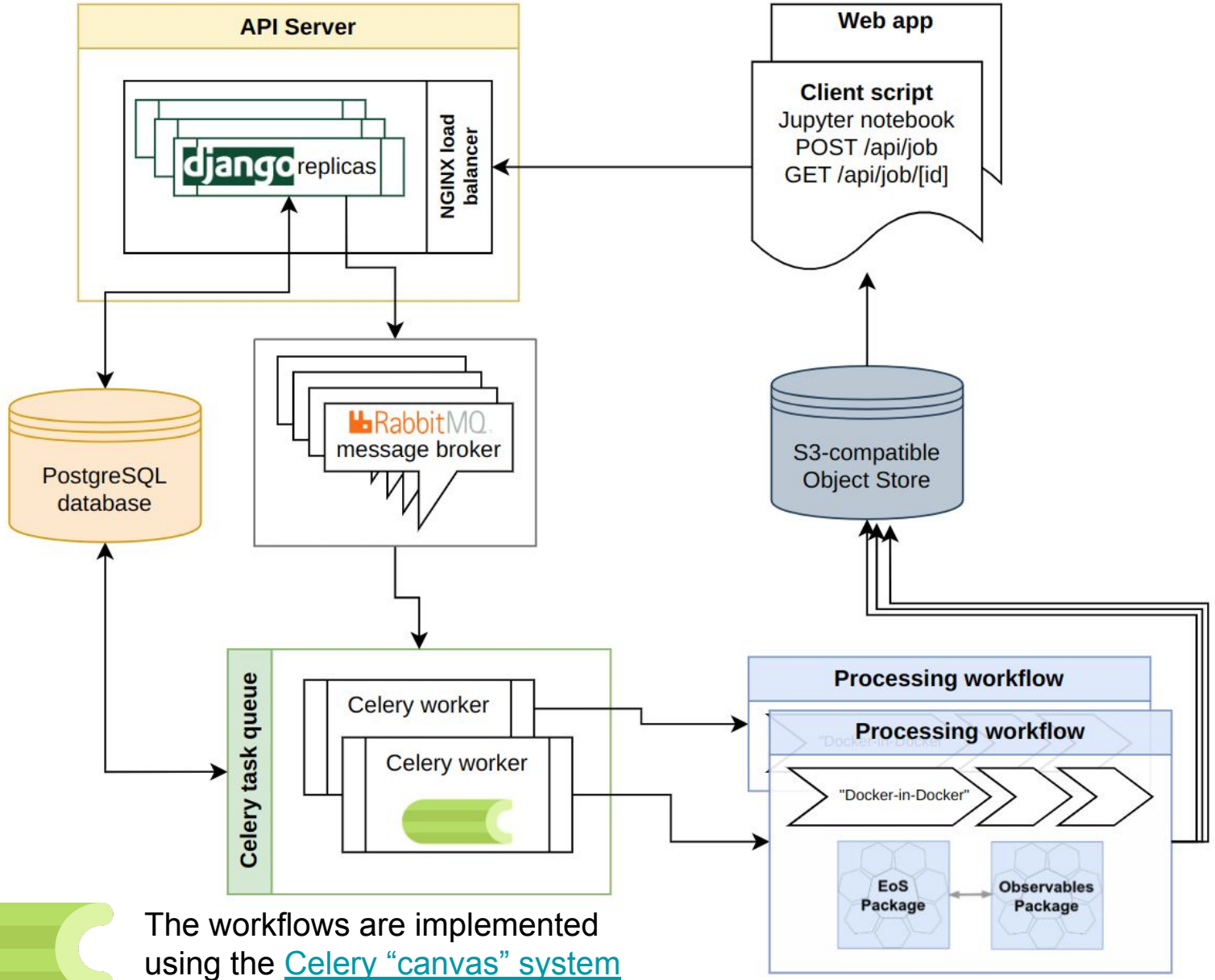
- takes the input requests from users with the workflow config,
- (optionally) stores uploaded files for use as inputs,
- launches processing jobs on our compute nodes,
- keeps track of job status and metadata in a database,

and handles things like

- notifying users when their jobs are complete
- returning output data



Calculation Engine system architecture

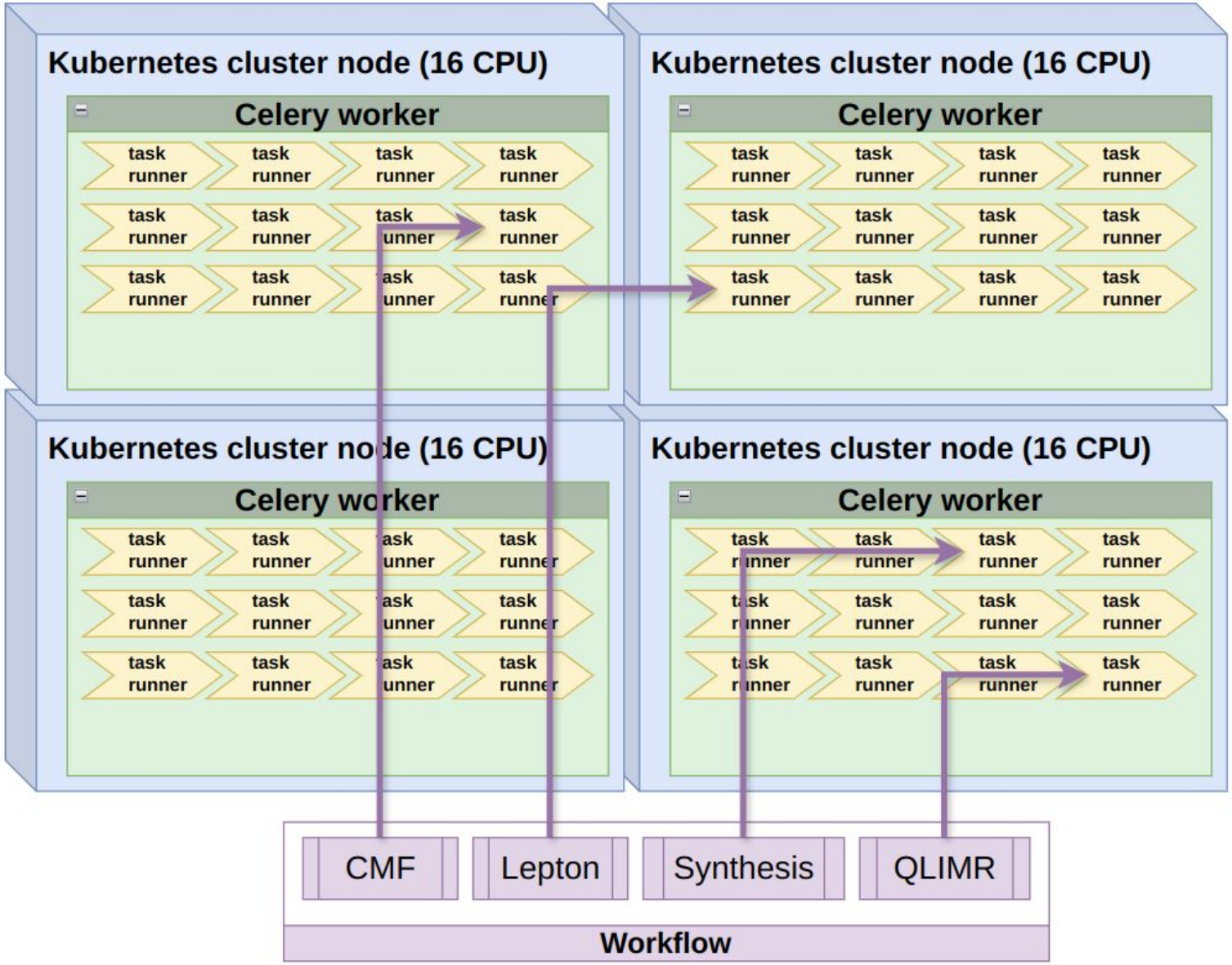


The workflows are implemented using the [Celery “canvas” system]

Workflow management system

Current configuration:

- 10 worker nodes
- 160 cores
- 640 GB memory
- 1 task worker per node
- 12 concurrent tasks per worker
- 120 concurrent tasks
- 12 core limit per worker



Registered MUSES modules

A particular version of the CE has a static set of registered modules available for use in workflows.

These are documented in the version-controlled CE documentation.

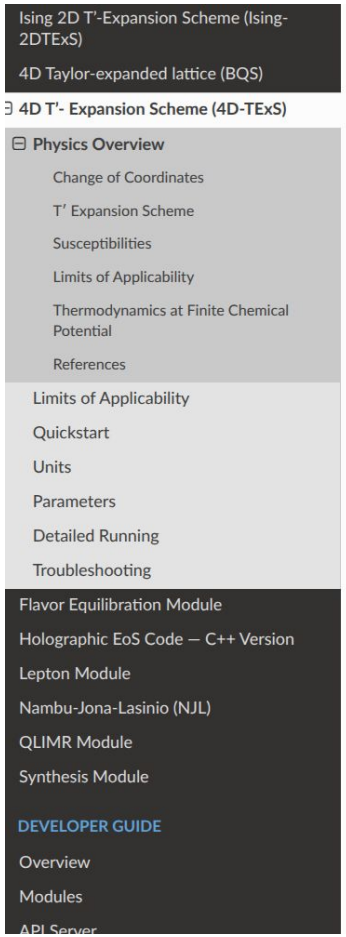
The equation of state modules include:

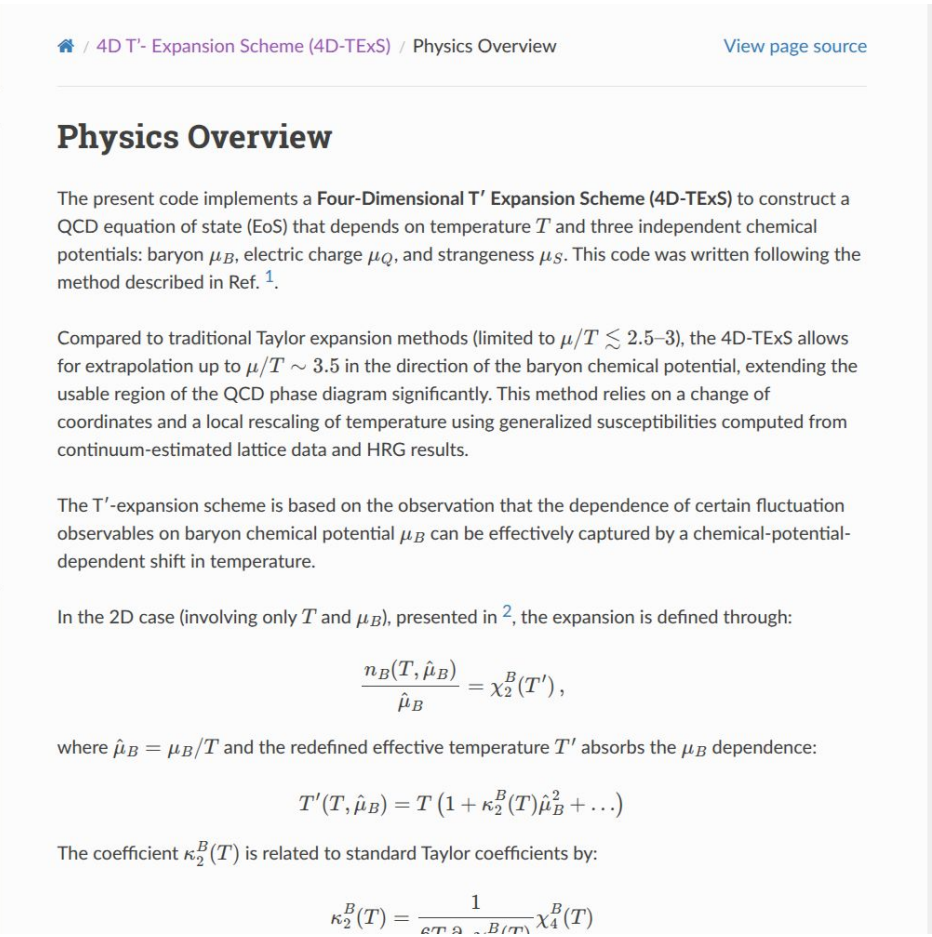
CE v1.10.0

- **Chiral EFT** – `chiral_eft (v1.0.1)`
- **Chiral Mean Field** – `cmf_solver (v1.0.3)`
- **Crust DFT** – `crust_dft (v1.0.3)`
- **Ising 2D T'-Expansion Scheme (Ising-2DTExS)** – `eos_ising_texs_2d (v1.0.2)`
- **4D Taylor-expanded lattice (BQS)** – `eos_taylor_4d (v1.0.2)`
- **4D lattice T-Expansion Scheme (4D-TExS)** – `eos_texs_4d (v1.0.1)`
- **Holographic** – `holographic_eos (v1.3.0)`
- **Lepton** – `lepton (v1.0.3)`
- **NJL** – `njl (v1.0.0)`

The observables modules include:

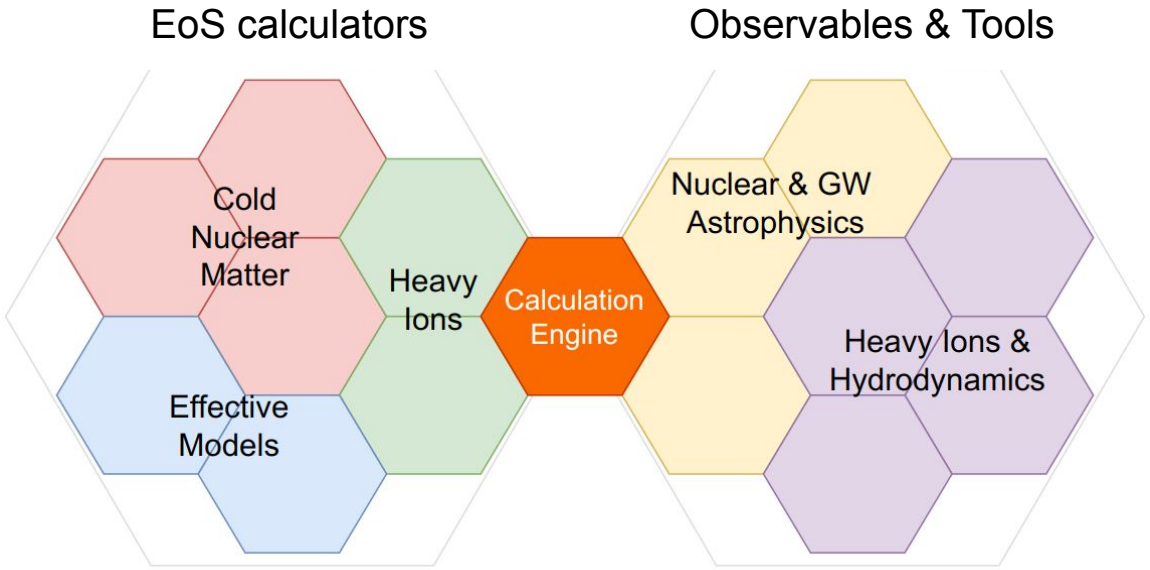
- **Flavor Equilibration** – `flavor_equilibration (v1.0.0)`
- **QLIMR** – `qlimr (v1.0.2)`
- **Synthesis** – `synthesis (v1.0.4)`





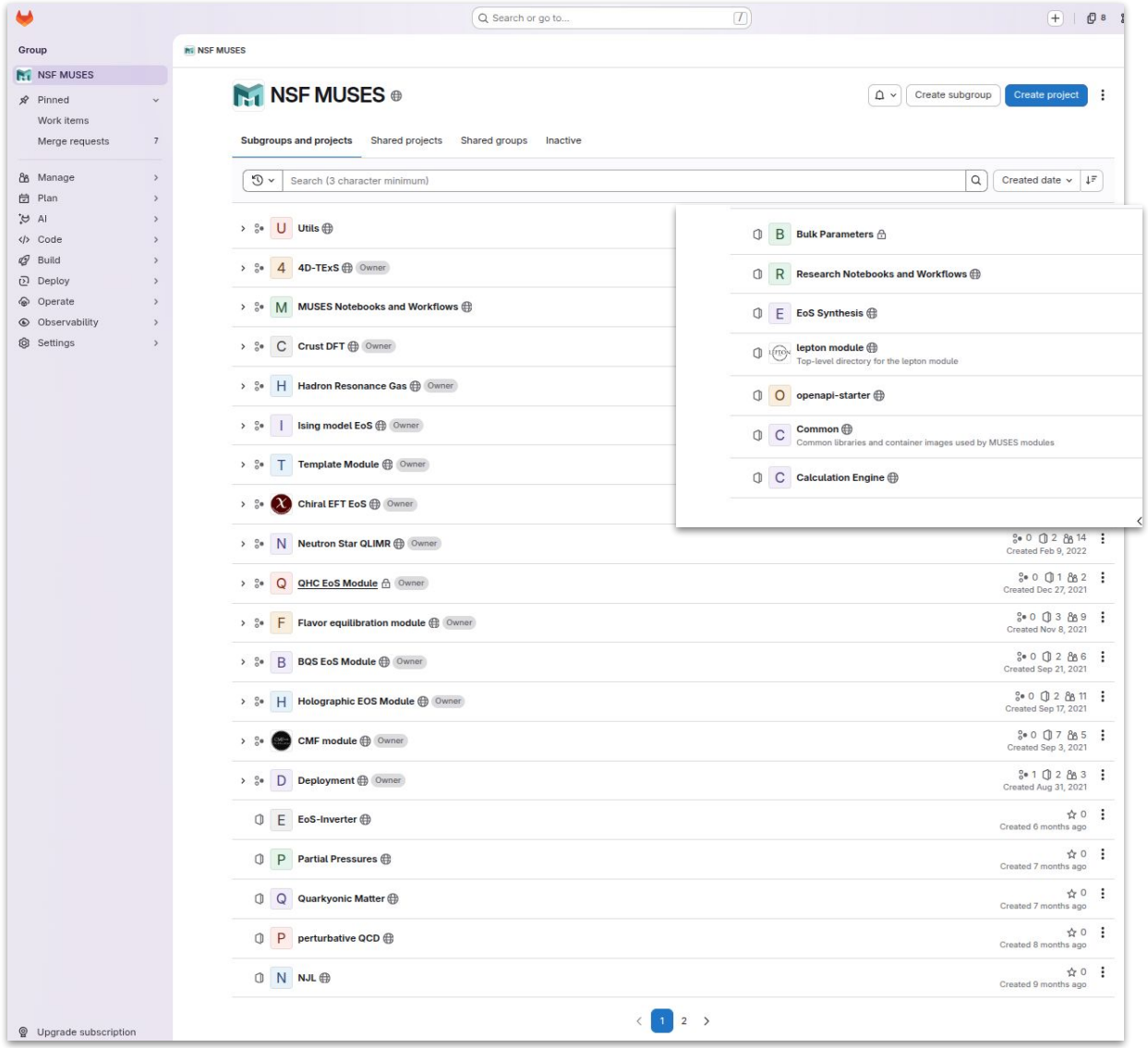
Integration of new modules

Integration is the process of adapting a code module to comply with the MUSES framework we have defined to allow the CE to execute workflows.

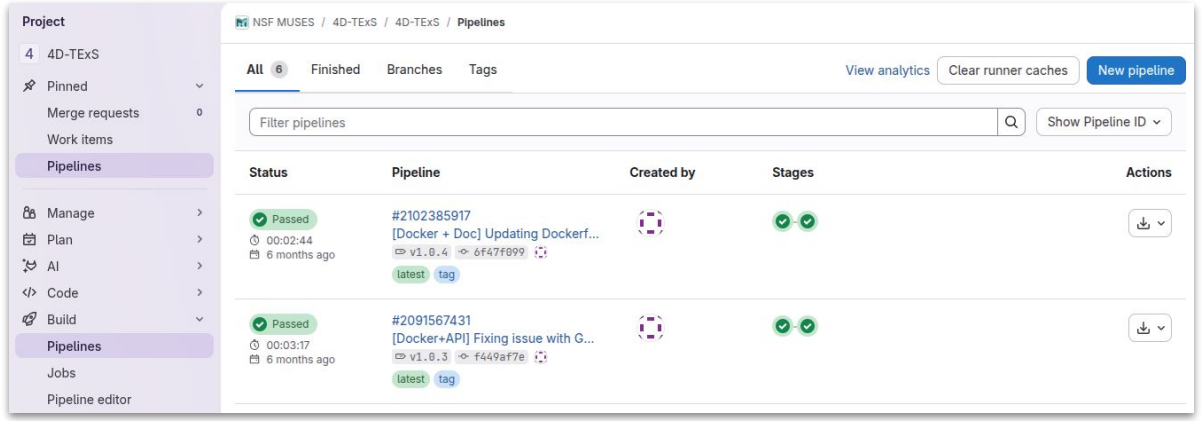


- **Containerization**
  - Modules must build and push container images that the CE can download and execute via Docker.
- **Manifest**
  - Each module defines a so-called **manifest**, declaring information about itself in a standard format for inclusion in the CE config.
- **API specification**
  - The inputs and outputs of a module must be declared in machine-level detail using the OpenAPI standard format

Module software repos and CI/CD



- Module development teams have learned to use GitLab and GitHub CI/CD pipelines to automatically build Docker images and run unit tests on those images
- This method ensures reproducibility and integrity of images with respect to the source code



Resouce sharing and usage limits

The public CE instance is a shared and finite resource. We must have a combination of **technical limits** and **rules of etiquette**.

There are tiers that set the usage limits for each user. Currently these include:

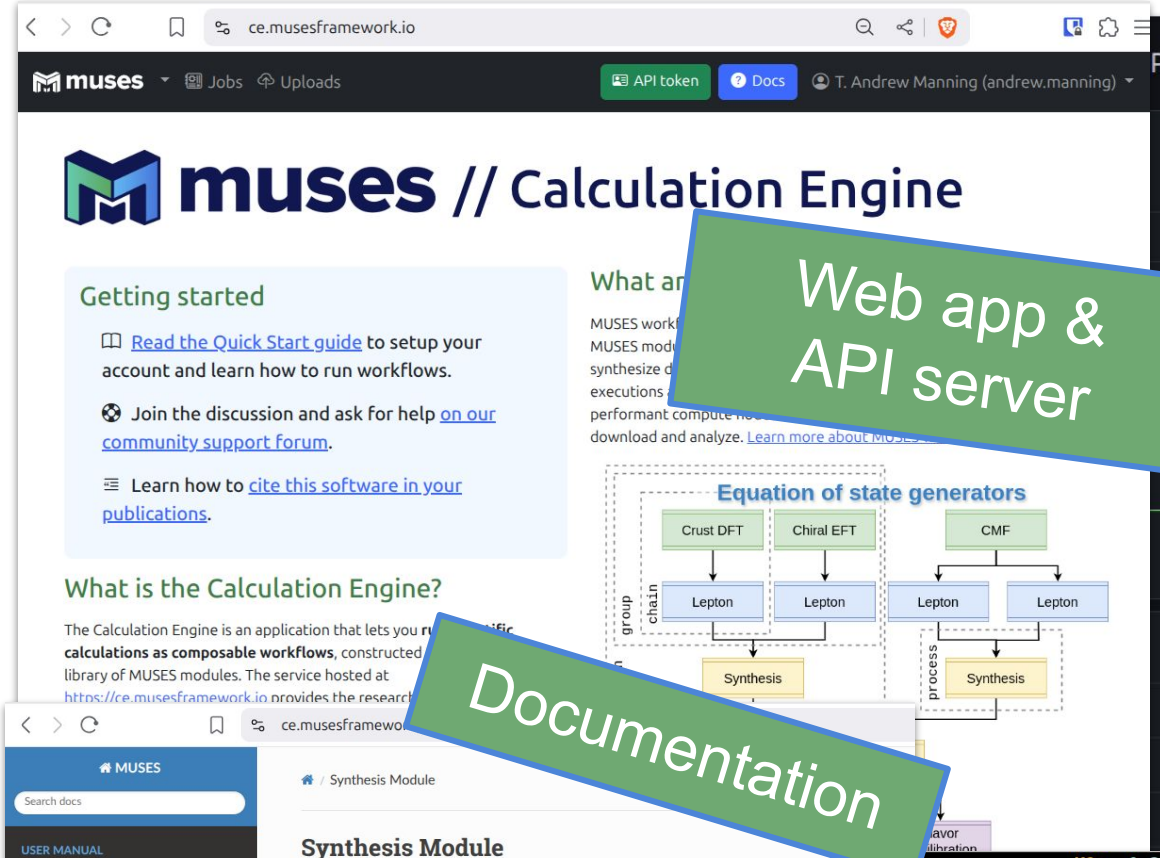
- Number of concurrent jobs
- Number of processes per workflow
- Total size of uploaded files
- Number of saved jobs

Job metadata and output files are deleted after 48 hours unless they are saved.

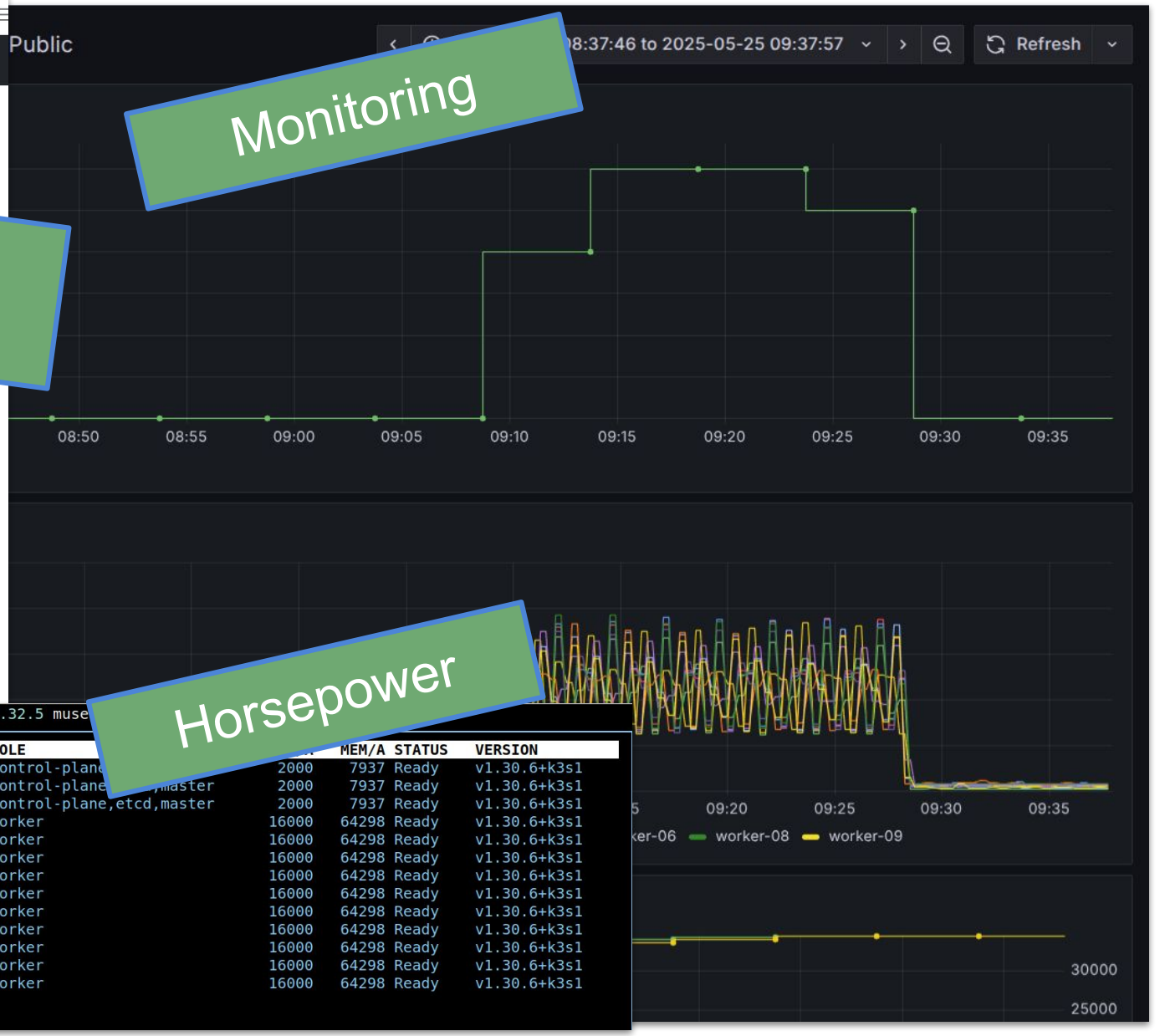
Saved jobs and uploaded files can be marked public for use by anyone.

From May 2025: tada! Public version is online!

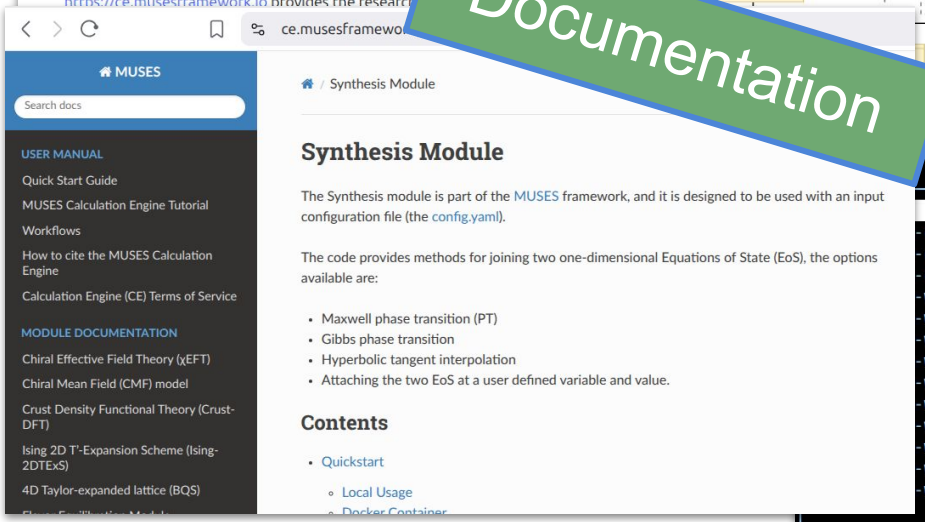
muses



Web app & API server



Monitoring



Documentation

Horsepower

Calculation Engine history

Feb 2025 — v1.0.0 was released

Apr 2025 — Public service announced

May 2025 — Annual MUSES meeting

Jul 2025 — “Gooey” introduced in v1.4.0, forum job notifications in v1.5.0

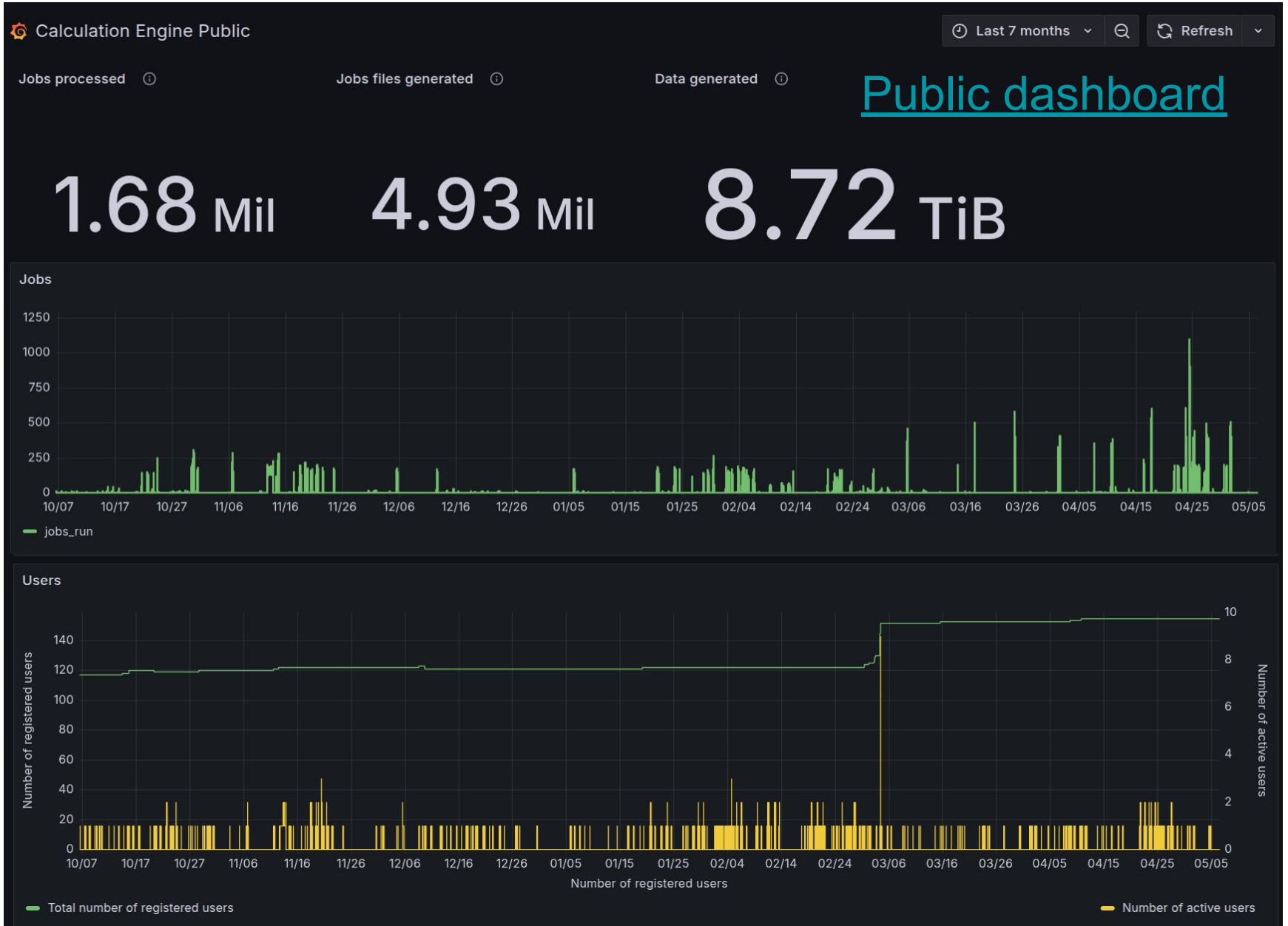
Various small improvements and bug fixes

May 2026

CE: Usage

Intermittent bursts of activity

Typically 1 or 2 simultaneous users



CE: Usage

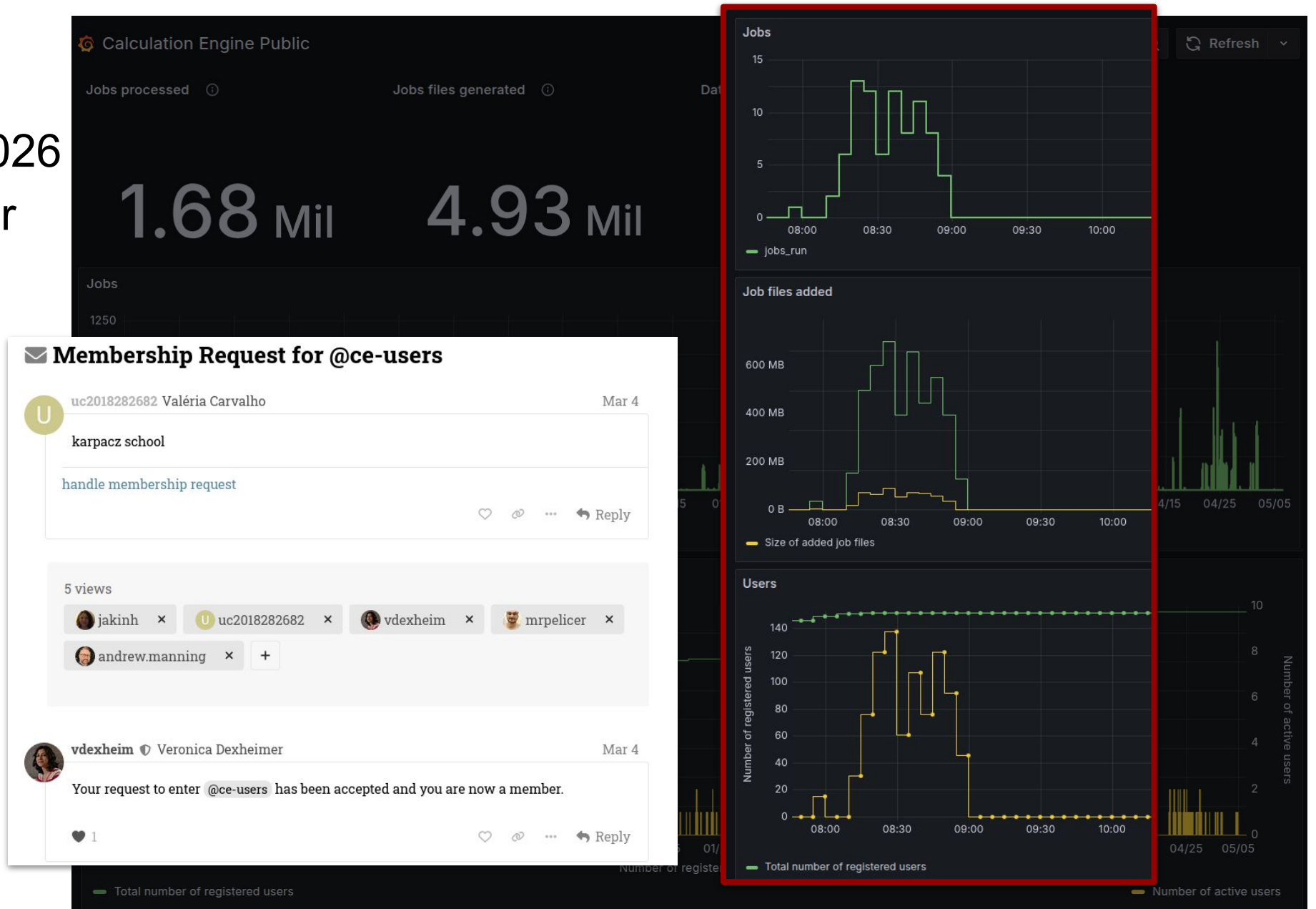
Majority of jobs were run almost a year ago by a “power user” who generously offered to help us debug the system by thrashing it relentlessly.

We will name no names.

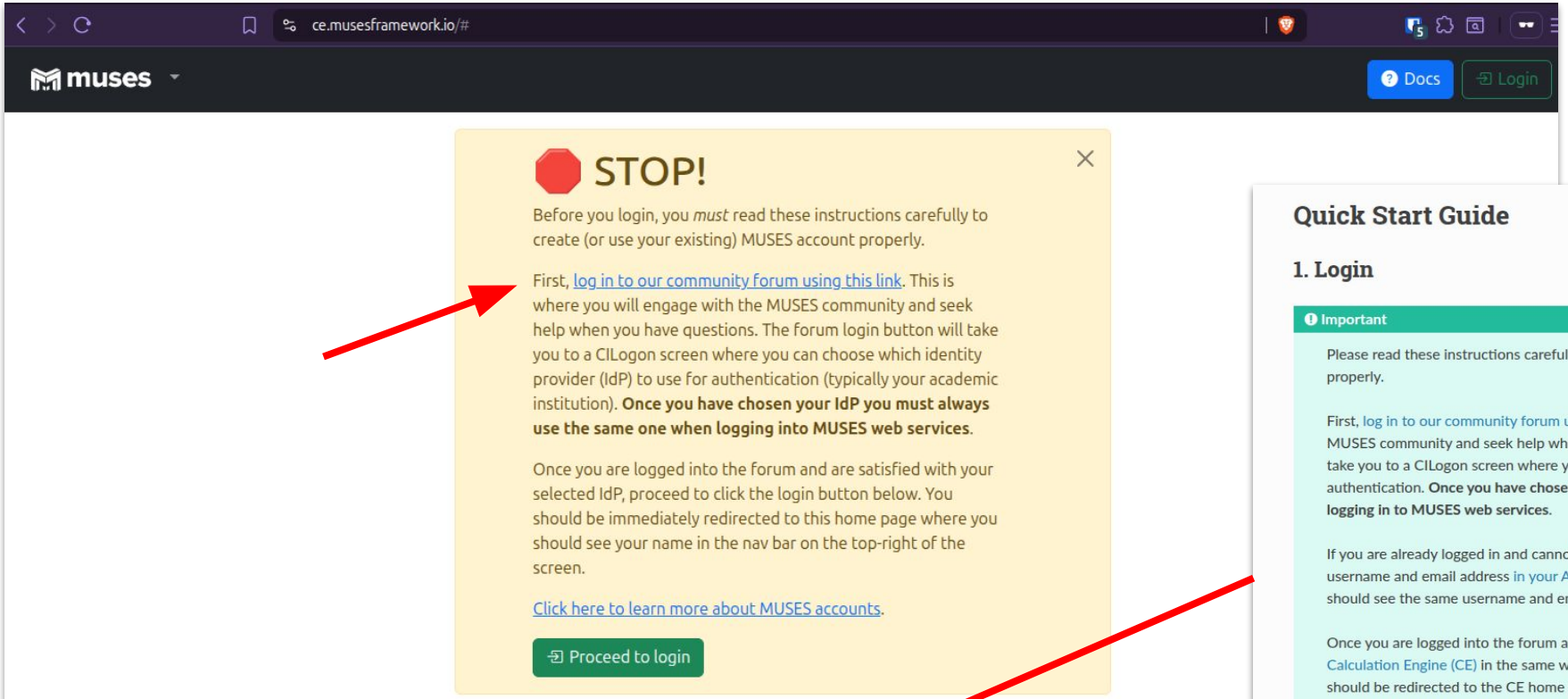
CE: Usage

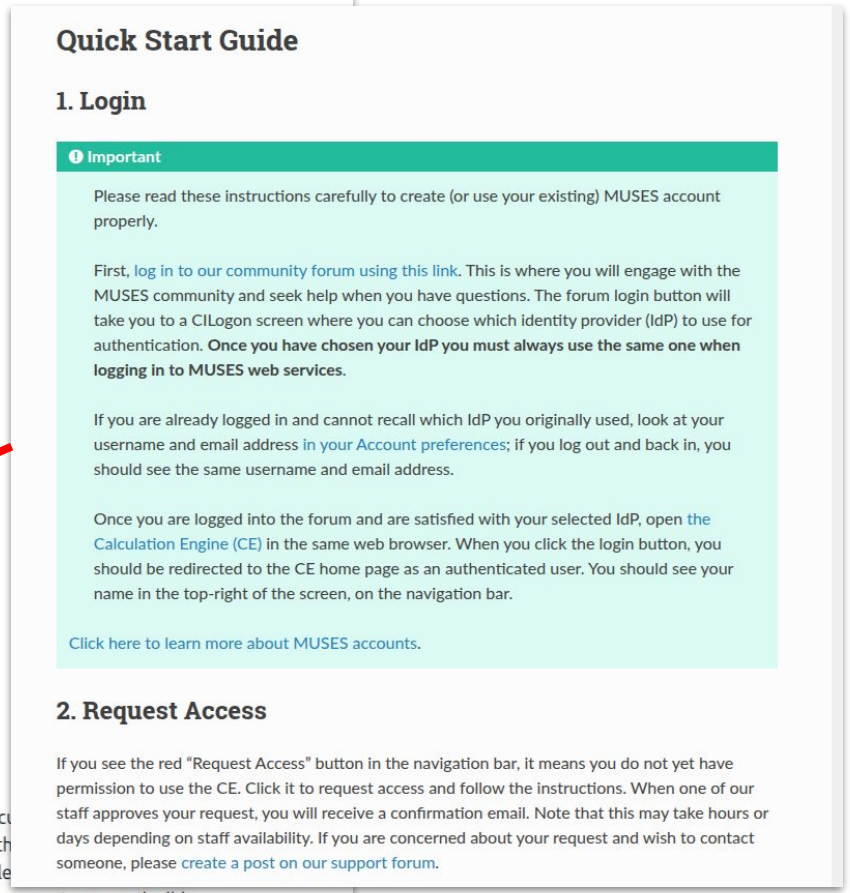
Small spike March 2026 during Karpacz winter school in Theoretical Physics "Multimessenger Astrophysics and Cosmology".

The new user onboarding system worked smoothly!

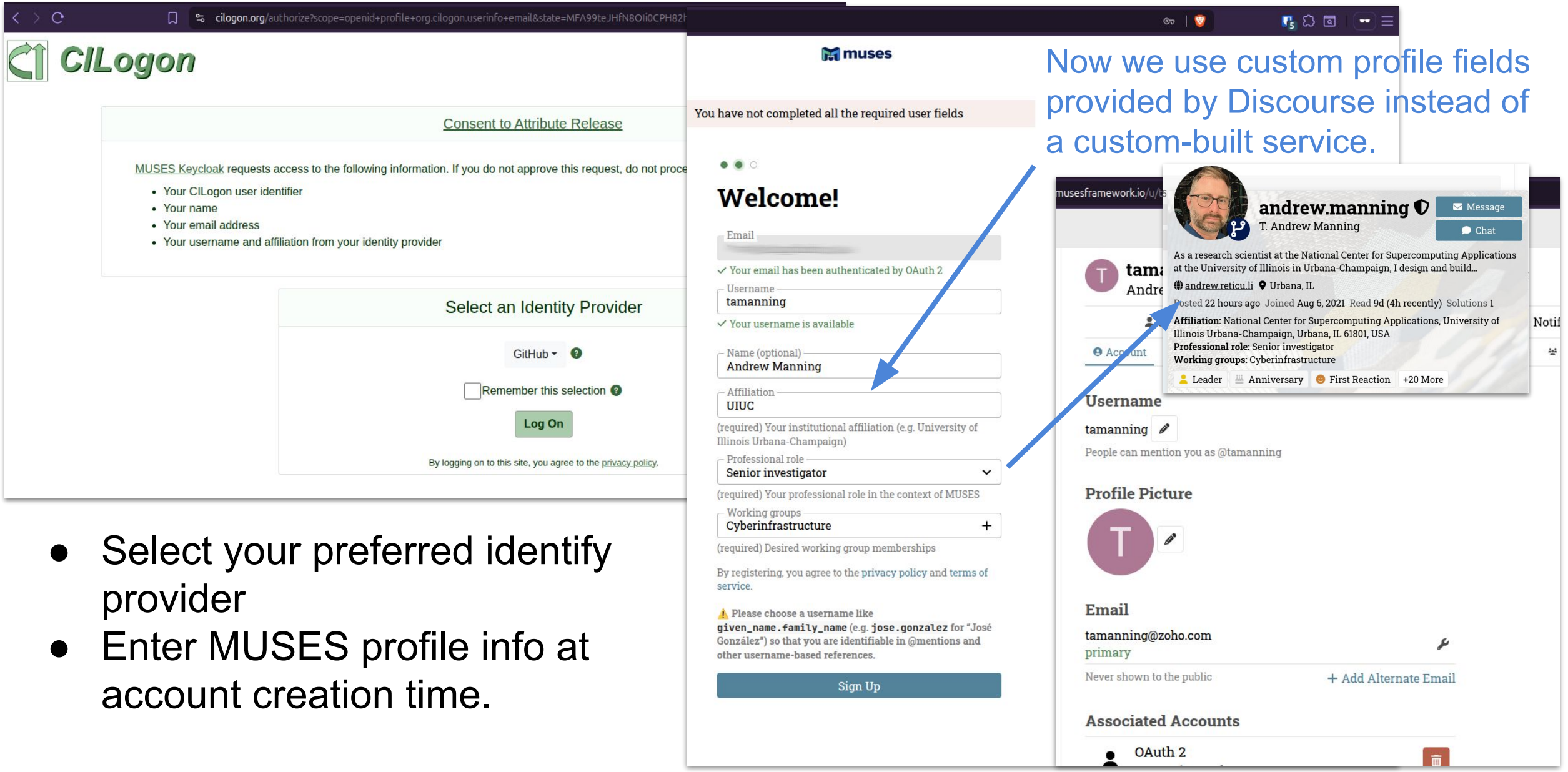


CE: New user onboarding



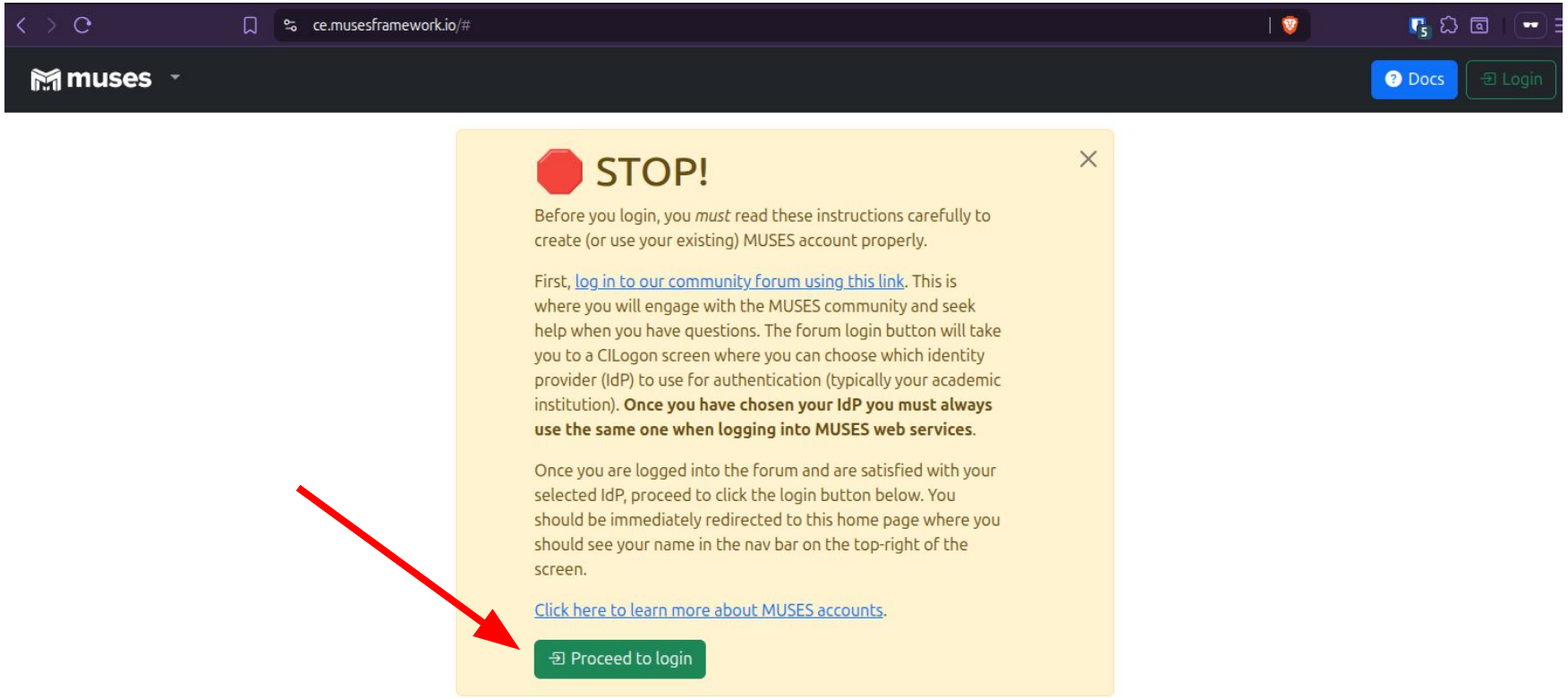


CE: New user onboarding



- Select your preferred identify provider
- Enter MUSES profile info at account creation time.

CE: New user onboarding



CE: New user onboarding

ce.musesframework.io

muses | Gooey | Docs | Request access | Andrew Manning (anaqreon)

muses // Calculation Engine

Getting started

Read the Quick Start guide to setup your account and learn how to run workflows.

Join the discussion and ask for help on our community support forum.

Learn how to cite this software in your publications.

What is the Calculation Engine?

The Calculation Engine is an application that lets you **run and combine scientific calculations for nuclear physics, neutron stars and heavy-ion collision in reproducible, composable workflows**, constructed from a growing library of MUSES modules. The service hosted at https://ce.musesframework.io provides the research community with scalable, high-performance computing resources to run intensive calculations.

The Calculation Engine is free and open source software, so if you are a interested in self-hosting the Calculation Engine on your own hardware, either for development purposes or to operate your own production instance, please see our developer documentation here.

What are MUSES modules?

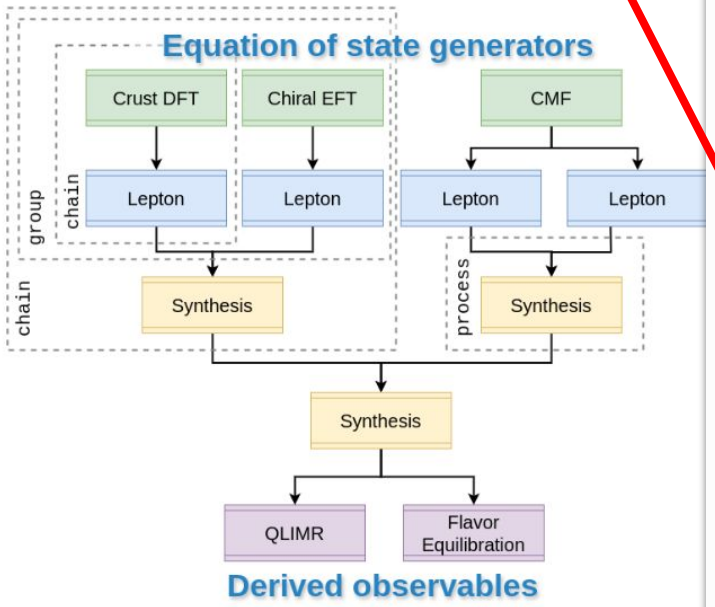
Modules are the atomic processing units of a MUSES workflow. There are several types of modules, including those that calculate equations of state (EoS) and those that derive observable quantities from the EoS.

The equation of state modules include:

- **Chiral EFT** – chiral_eft (v1.0.1)

What are workflows?

MUSES workflows provide a way to orchestrate a custom execution of MUSES modules, allowing you to generate equations of state, process and synthesize data, and calculate observable quantities. Individual workflow executions are called jobs, which you can run concurrently on our high-performance compute nodes to generates reproducible results to download and analyze. Learn more about MUSES workflows here.



What is MUSES?

Quick Start Guide

1. Login

Important

Please read these instructions carefully to create (or use your existing) MUSES account properly.

First, log in to our community forum using this link. This is where you will engage with the MUSES community and seek help when you have questions. The forum login button will take you to a CILogon screen where you can choose which identity provider (IdP) to use for authentication. **Once you have chosen your IdP you must always use the same one when logging in to MUSES web services.**

If you are already logged in and cannot recall which IdP you originally used, look at your username and email address in your Account preferences; if you log out and back in, you should see the same username and email address.

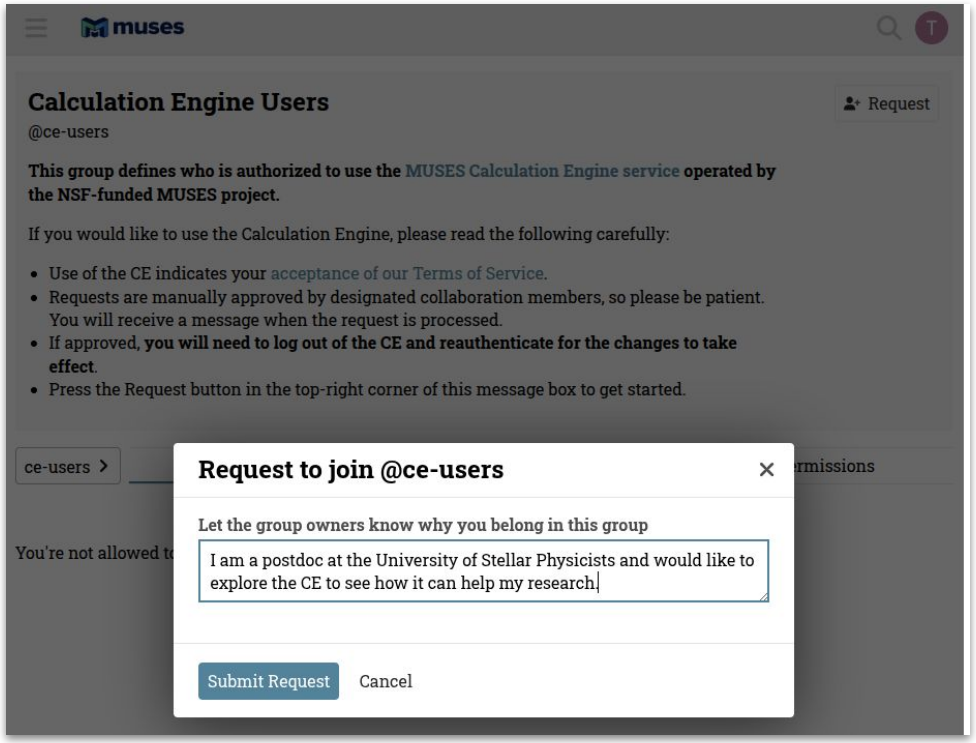
Once you are logged into the forum and are satisfied with your selected IdP, open the Calculation Engine (CE) in the same web browser. When you click the login button, you should be redirected to the CE home page as an authenticated user. You should see your name in the top-right of the screen, on the navigation bar.

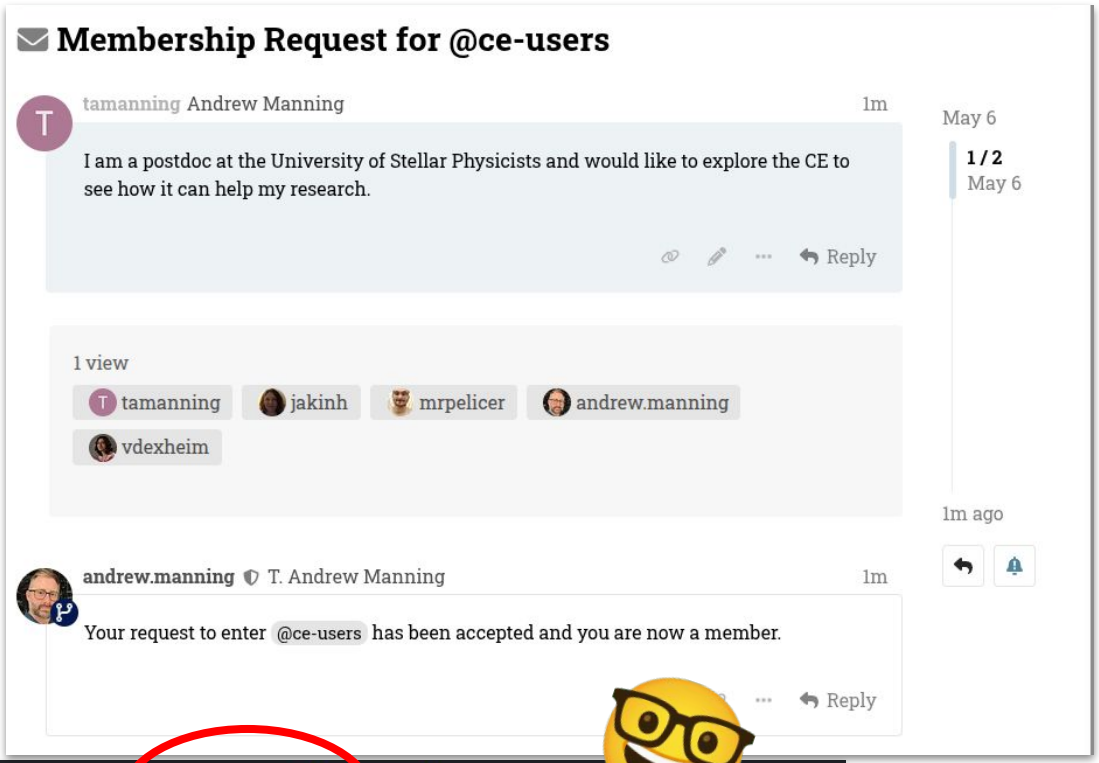
Click here to learn more about MUSES accounts.

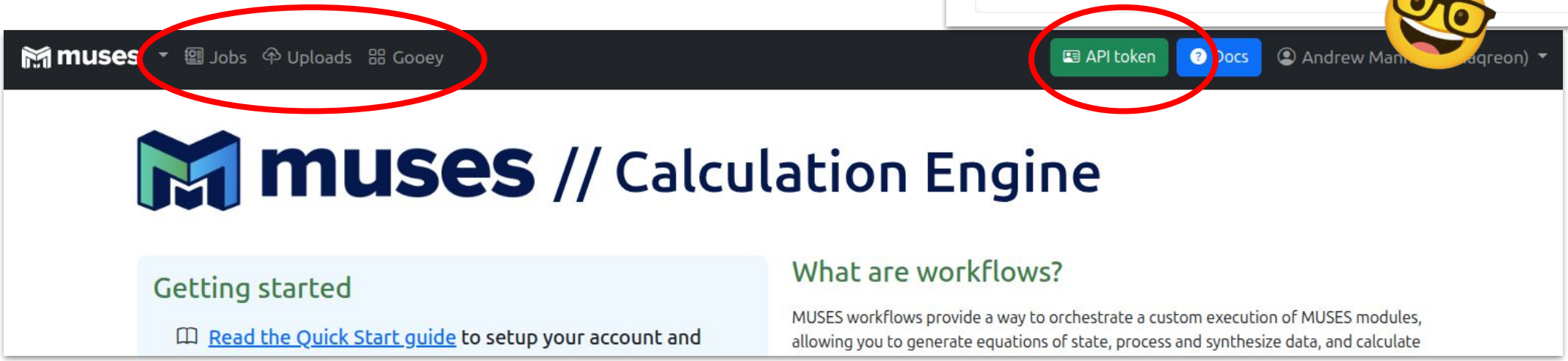
2. Request Access

If you see the red "Request Access" button in the navigation bar, it means you do not yet have permission to use the CE. Click it to request access and follow the instructions. When one of our staff approves your request, you will receive a confirmation email. Note that this may take hours or days depending on staff availability. If you are concerned about your request and wish to contact someone, please create a post on our support forum.

CE: New user onboarding



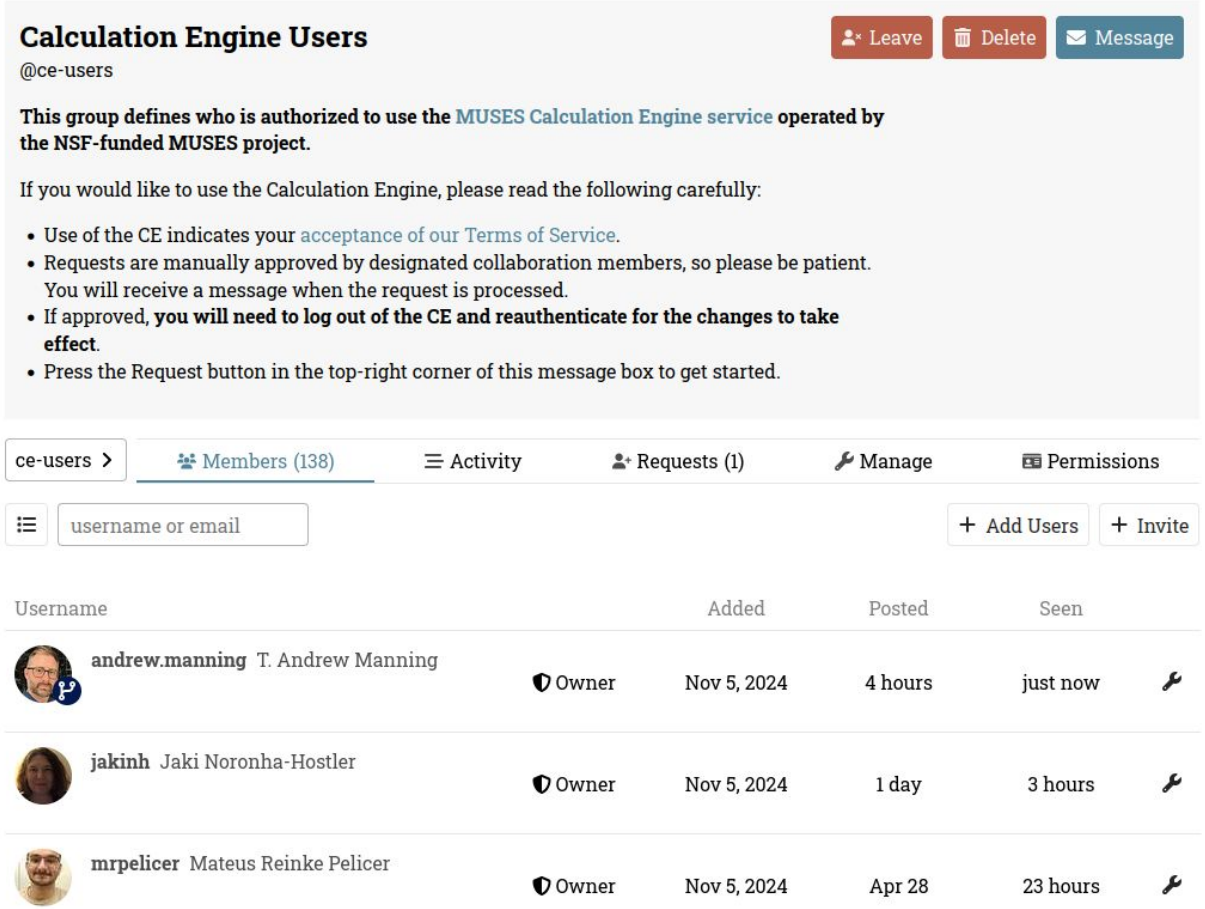




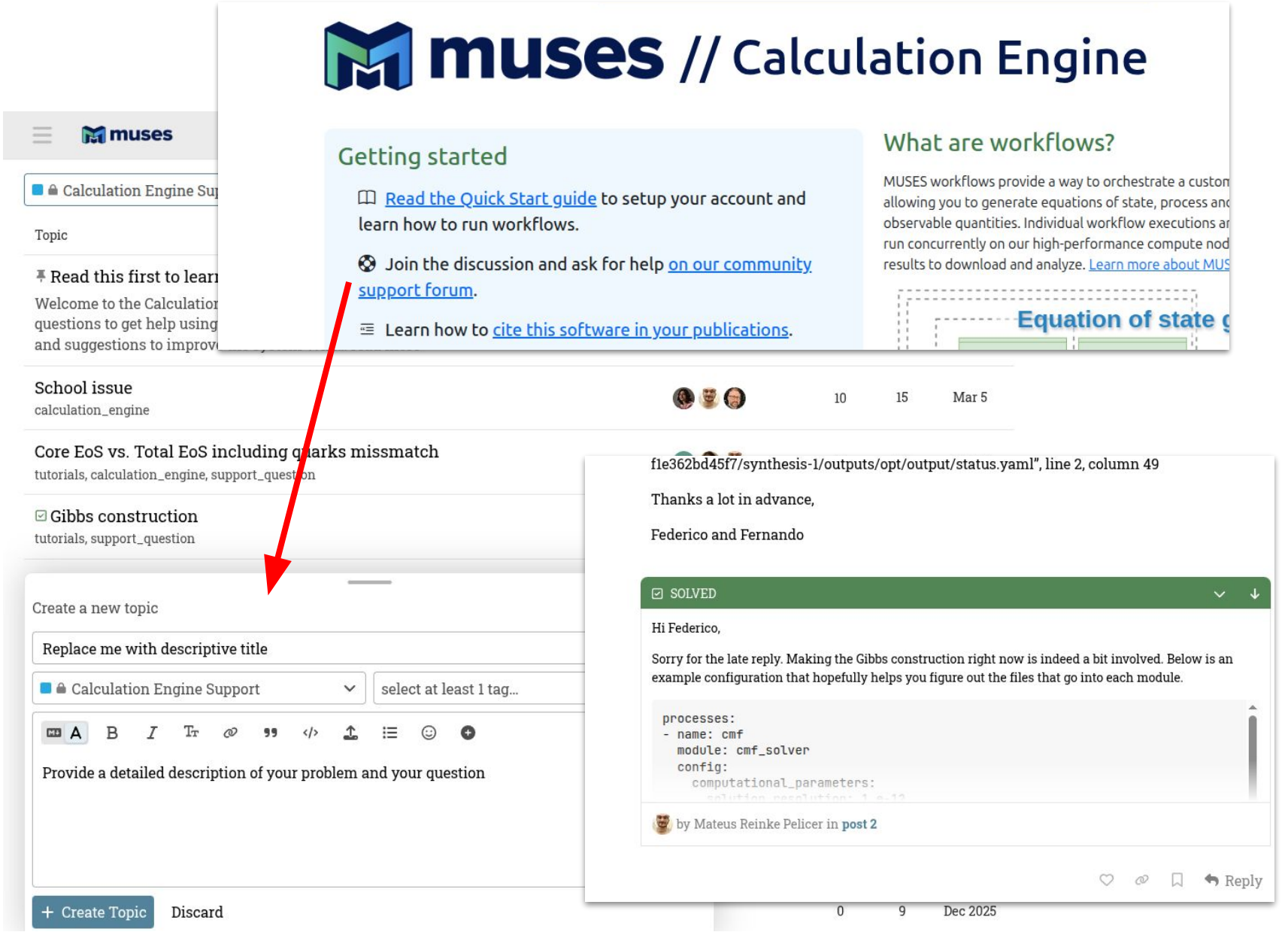
CE: Delegated access control using the forum

Authentication (who you are) is unified with the rest of MUSES web services, where user accounts are defined on our own Keycloak server, which uses the external CILogon service as the identity broker.

Authorization (what you can do) is defined by membership in the ***ce-users*** group on the forum. The owners of this group receive notifications when someone requests access and can approve/reject with the click of a button. *The owner role is dynamic, supporting delegation of responsibility*.



CE: Community-based user support



The forum enables community-based support.

Authorized CE users have access to a support category where they post questions as new topics.

Topics can be marked “solved” with quick links to the answers.

Sustainability of MUSES CI

- Migration of forum, cloud storage, other services to other hosting providers, potentially transferring web domain ownership to new steward organization.
- Options for operating the public CE service
- CE support for custom HPC backend (Globus Compute)
- Implement KEDA (Kubernetes Event-driven Autoscaling) for hyper-efficient use of elastic computing resources (e.g. Jetstream2 via NSF ACCESS)
- CACAO template for researcher-owned CE deployment using NSF ACCESS
- Switch to Apptainer and Podman Compose to align better with FOSS and for better compatibility with HPC environments

Discussion on the future of the CI is Tuesday afternoon