

# E E S S I

EUROPEAN ENVIRONMENT FOR  
SCIENTIFIC SOFTWARE INSTALLATIONS

**ISC'24 - Hamburg - 13+15 May 2024**

*Kenneth Hoste & Lara Peeters (Ghent University, Belgium)*

[kenneth.hoste@ugent.be](mailto:kenneth.hoste@ugent.be)

[lara.peeters@ugent.be](mailto:lara.peeters@ugent.be)

# Who are we?

## Kenneth Hoste: Computer scientist from Belgium



- HPC system administrator + user support at Ghent University since Oct 2010
- BDFL of **EasyBuild** - tool to build & install **scientific software on HPC clusters**
- Active contributor to **EESSI**, partner in MultiXscale EuroHPC CoE
- Fan of open source software, beers, and stickers
- [“How To Make Package Managers Cry” talk](#) at FOSDEM’18 (if you haven’t seen it, you should!)

## Lara Peeters: Digital Art Historian from Belgium



- Hired on the MultiXscale project at Ghent University (Belgium) since May 2023
- Active contributor to **EasyBuild & EESSI**, partner in MultiXscale EuroHPC CoE
- Just getting started, still figuring out the “art” of software packaging

# Scientific software is a different breed

- Scientists may not be trained software engineers (and that's OK)
- They often need help to get the software installed properly (especially on HPC systems)
- **Standard packaging tools do not suffice**, we ~~want~~ need to **build from source**
- Scientific software should be compiled for system on which it will be used (performance!)



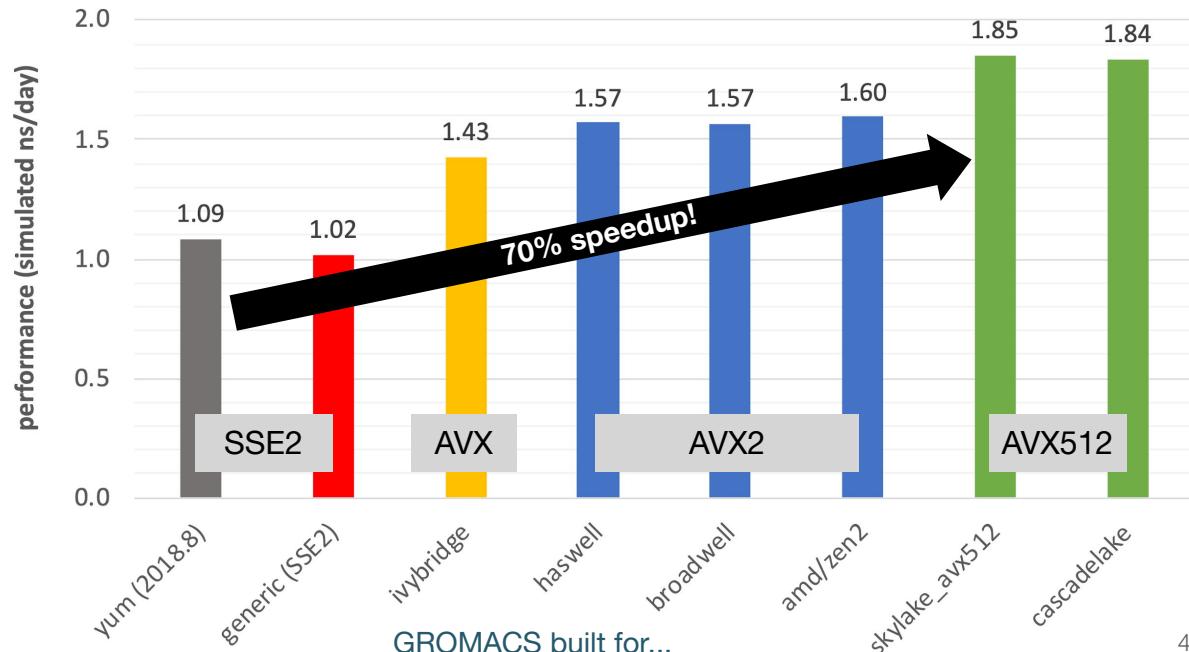
*brilliant scientist*



*HPC user support*

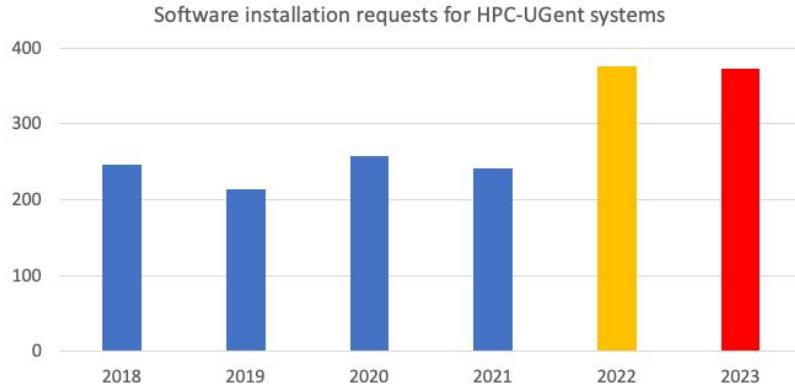
# Optimized scientific software installations

- Software should be optimized for the system it will run on
- Impact on performance is often significant for scientific software
- Example: GROMACS 2020.1  
(PRACE benchmark, Test Case B)
- Metric: (simulated) ns/day,  
higher is better
- Test system: dual-socket  
Intel Xeon Gold 6420  
(Cascade Lake, 2x18 cores)
- **Performance of different  
GROMACS binaries,  
on exact same hardware/OS**



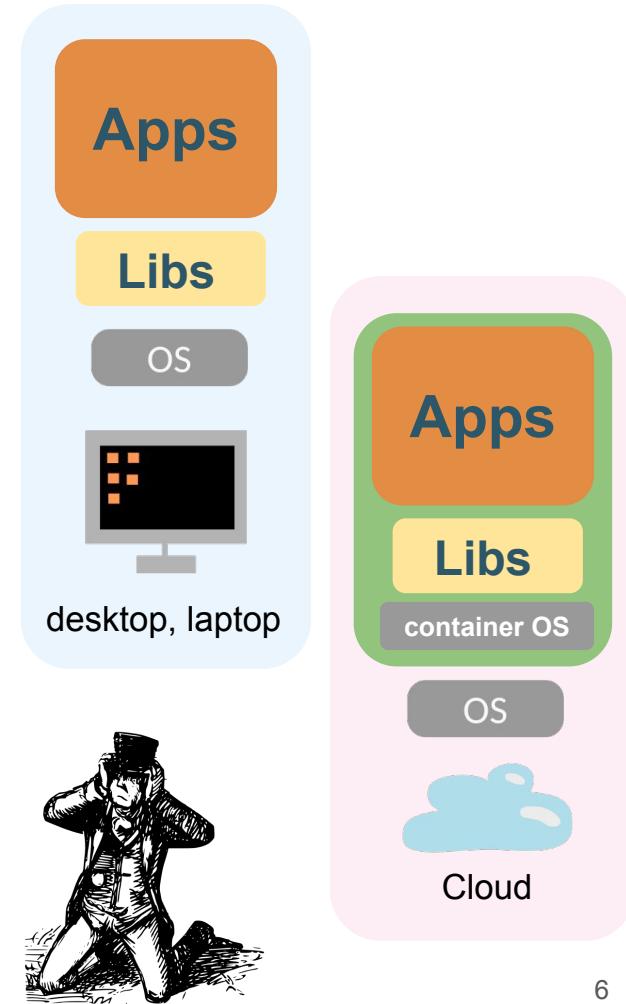
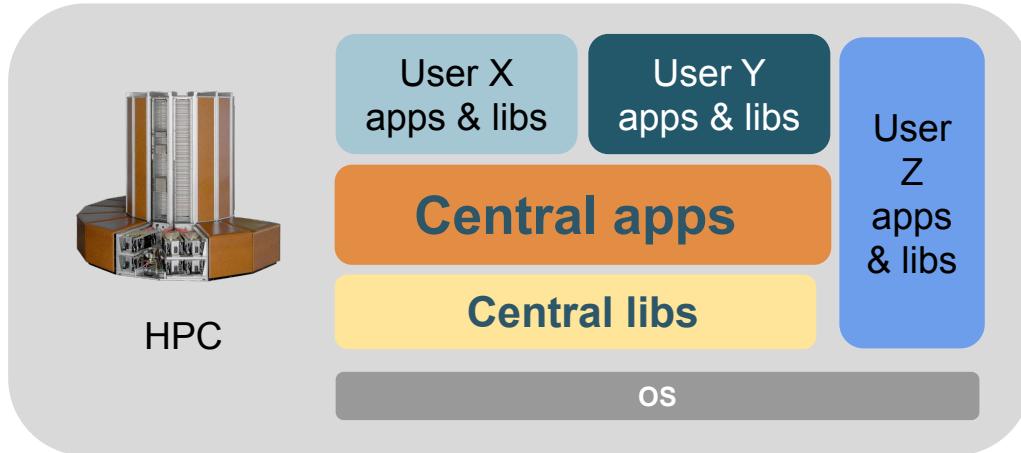
# Landscape of scientific computing is changing

- **Explosion of available scientific software** applications (bioinformatics, AI, ...)
- Increasing interest in **cloud** for scientific computing (flexibility!)
- **Increasing variety in processor (micro)architectures** beyond Intel & AMD:  
Arm is ~~coming~~ already here (see [Fugaku](#), [JUPITER](#), ...), RISC-V is coming (soon?)
- Broader adoption of **accelerated computing**, beyond NVIDIA GPUs
- In strong contrast: available (wo)manpower in HPC support teams is (still) limited...



# We need to collaborate more...

- **Too much software** for a single support team to handle
- Different systems (CPU, GPU, OS, ...) => different problems
- Existing tools (EasyBuild, Spack) are **not sufficient anymore...**
- Lots of **duplicate work** across HPC sites and scientists
- **Diverse software stacks** across different platforms



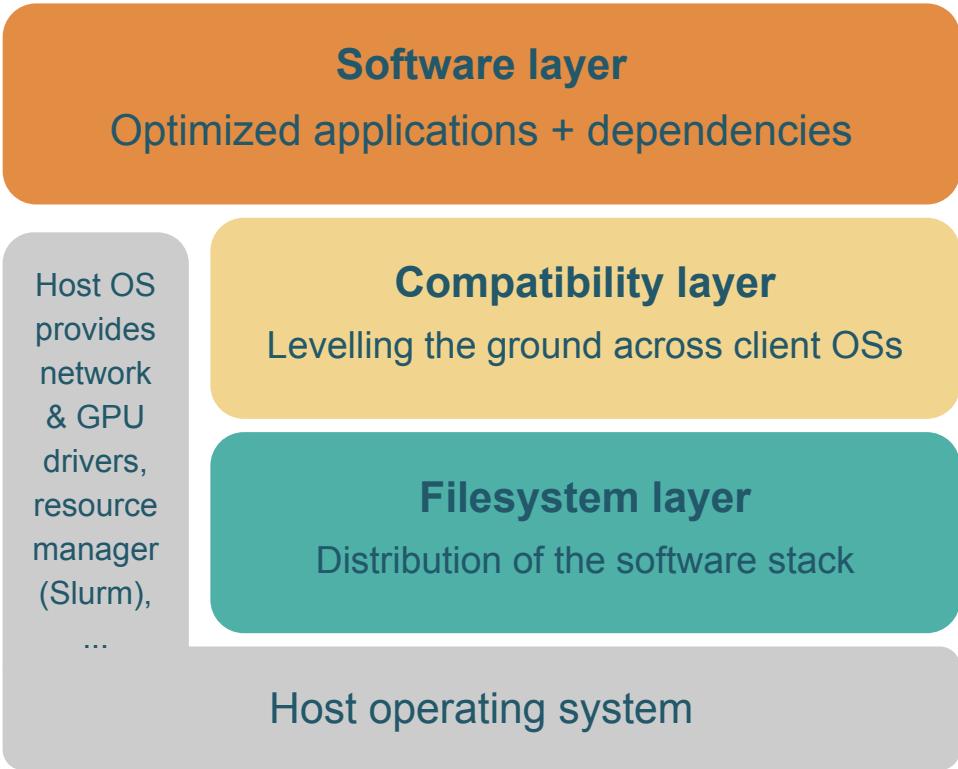
# European Environment for Scientific Software Installations

- Public repository of (optimized!) scientific software installations
- Avoid duplicate work by collaborating on a shared software stack
- Uniform way of providing software to users, regardless of the system they use!
- Should work on any Linux OS (incl. WSL) and system architecture
  - From laptops and personal workstations to HPC clusters and cloud
  - Support for different CPUs (AMD, Intel, Arm, RISC-V), interconnects, GPUs, etc.
- Focus on performance, automation, testing, collaboration



<https://eessi.io>

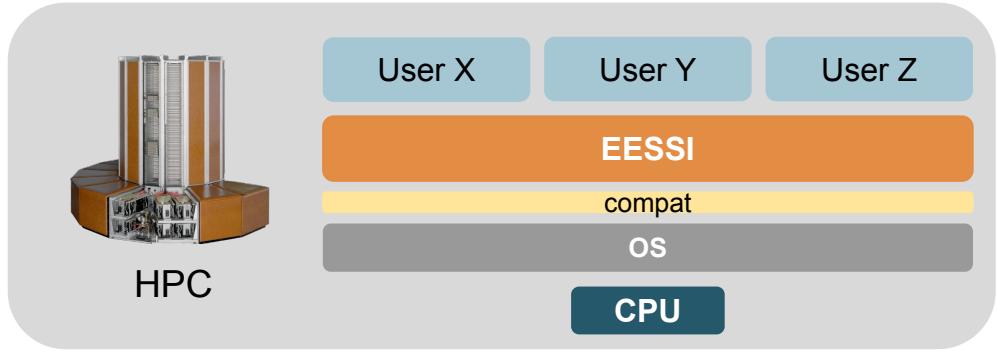
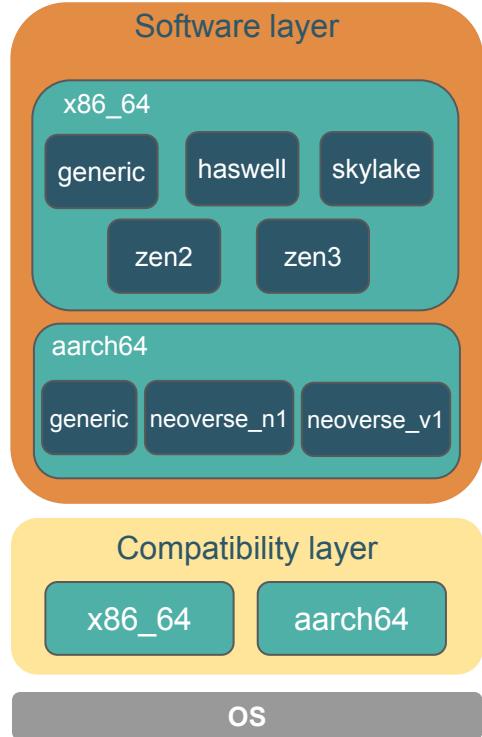
<https://eessi.io/docs>



**ESSI**  
EUROPEAN ENVIRONMENT FOR  
SCIENTIFIC SOFTWARE INSTALLATIONS



# EESSI to the rescue!



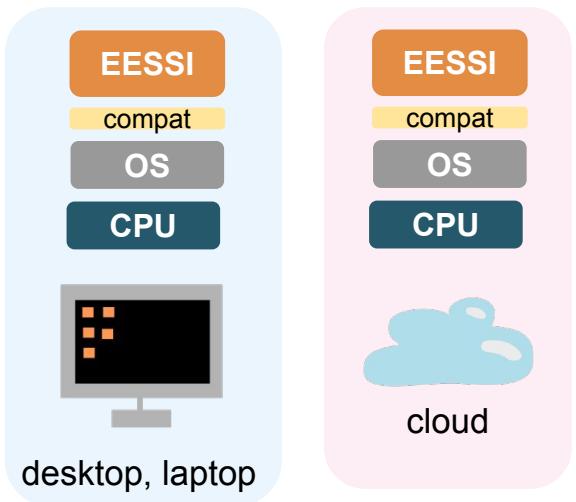
**Shared repository of  
(optimized) scientific  
software installations**

**Same software stack  
everywhere!**

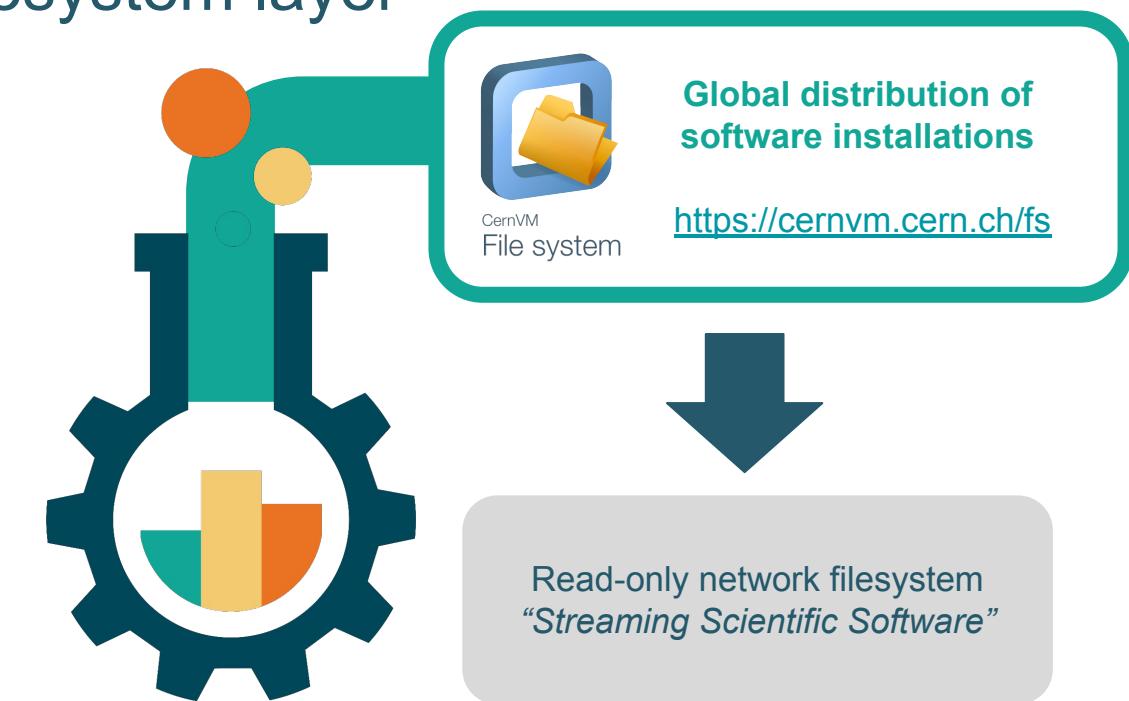
[eessi.io](http://eessi.io)

[eessi.io/docs](http://eessi.io/docs)

[eessi.io/docs/support](http://eessi.io/docs/support)



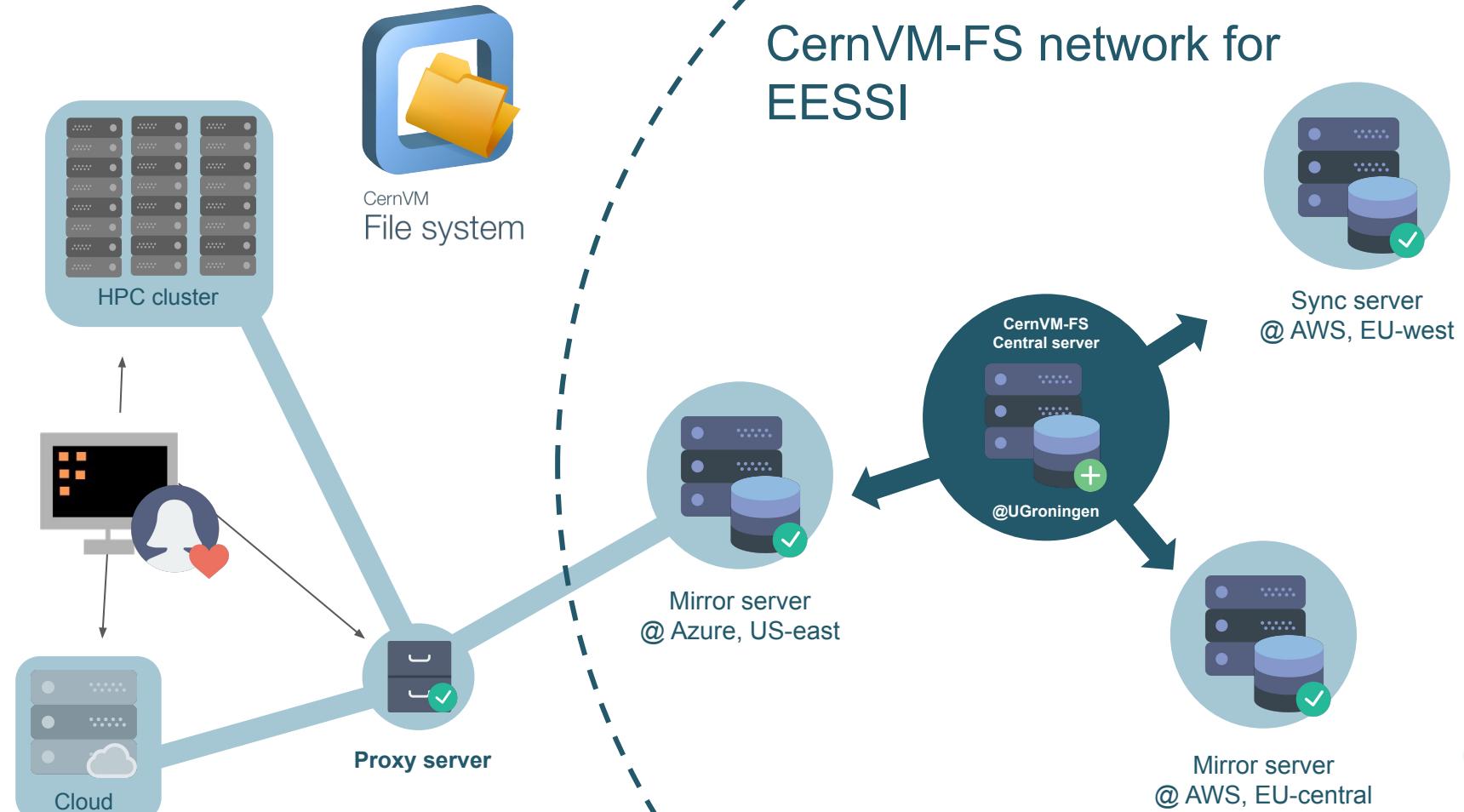
# EESSI ingredients: filesystem layer



**EESSI**

EUROPEAN ENVIRONMENT FOR  
SCIENTIFIC SOFTWARE INSTALLATIONS

# CernVM-FS network for EESSI



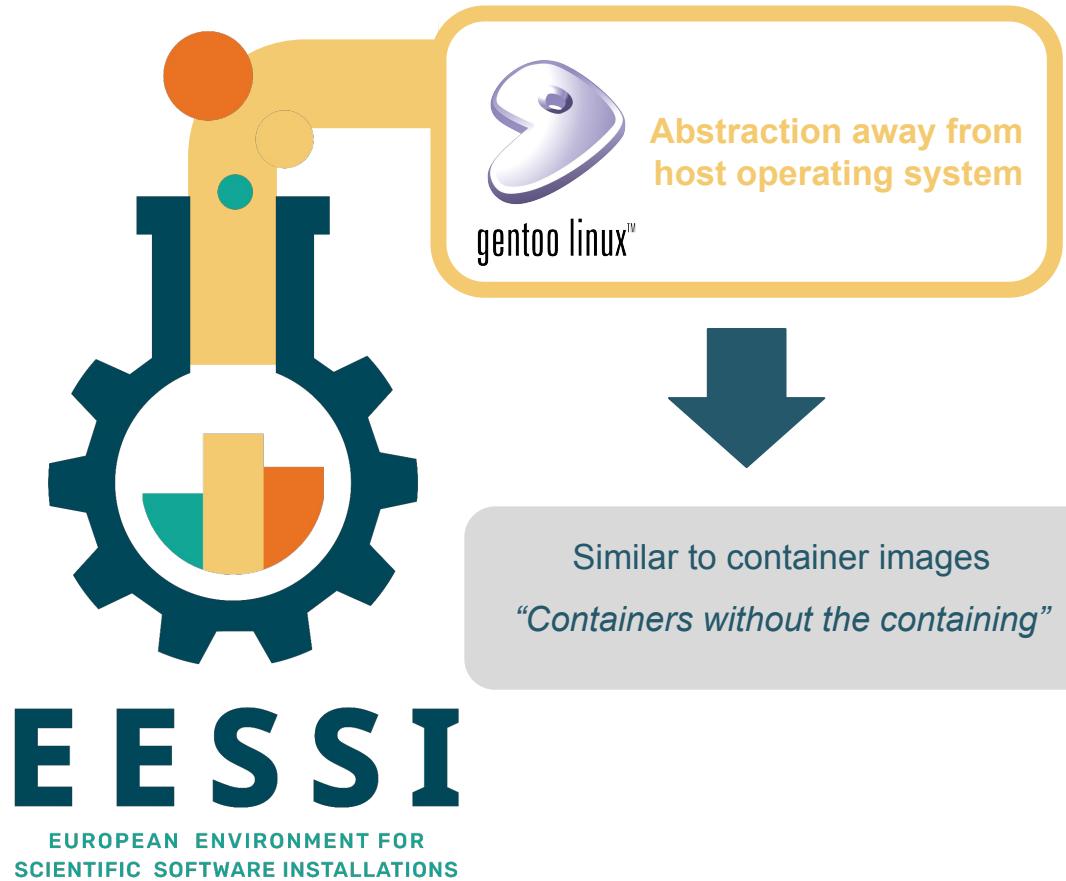
# Tutorial “Best Practices for CernVM-FS in HPC”



- <https://multixscale.github.io/cvmfs-tutorial-hpc-best-practices>
- Held online on 4 Dec 2023 (~3 hours), **recorded & available on YouTube**
- Over 200 registrations, ~125 attending the meeting
- Lecture + hands-on demos
- Topics:
  - Introduction to CernVM-FS + EESSI
  - Configuring CernVM-FS: client, Stratum 1 mirror server, proxy server
  - Troubleshooting problems
  - Benchmarking of start-up performance



# EESSI ingredients: compatibility layer



# EESSI compatibility layer

[github.com/EESSI/compatibility-layer](https://github.com/EESSI/compatibility-layer)



- “Containers without the containing”
- Minimal collection of tools and libraries (incl. glibc, bash, Python, Lmod, ...)
- Built from source per CPU family (x86\_64, aarch64, ...) with [Gentoo Prefix](#)
- Installations included in software layer only link to compat layer (RPATH)
- Ensures compatibility with any client system running Linux

```
$ ls /cvmfs/software.eessi.io/versions/2023.06/compat/linux/aarch64/
bin  etc  lib  lib64  opt  reprod  run  sbin  stage1.log  stage2.log
stage3.log  startprefix  tmp  usr  var
```

Software layer

Compatibility layer

x86\_64

aarch64

OS

# EESSI ingredients: software layer



gentoo linux™

Abstraction away from  
host operating system

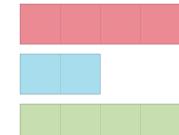
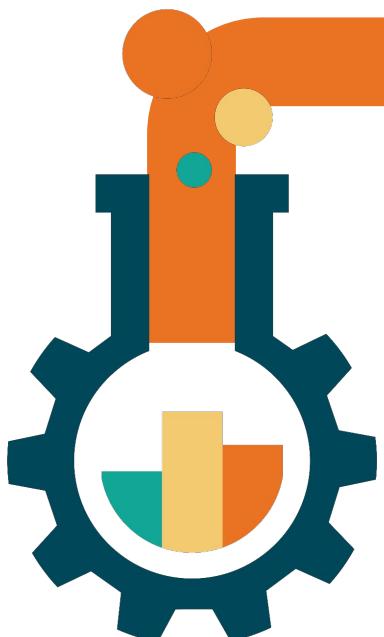


CernVM  
File system

Global distribution of  
software installations

# EESSI

EUROPEAN ENVIRONMENT FOR  
SCIENTIFIC SOFTWARE INSTALLATIONS

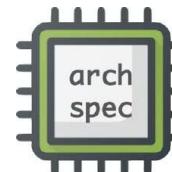


**EASYBUILD**

Optimized software  
installations for specific  
CPU microarchitectures

Intuitive user interface  
module avail,  
module load, ...

**Lmod**

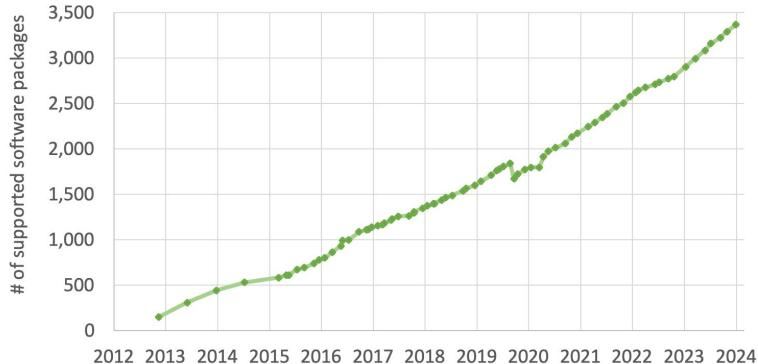


Automatic selection of  
best suited part of  
software stack for  
host CPU microarchitecture

# EasyBuild (in a nutshell)

<https://easybuild.io>

- EasyBuild is a tool to make **installing (scientific) software on HPC systems** easier
- **Building from source** + optimizing for specific CPU is strongly preferred (performance!)
- Created in 2009 by HPC-UGent team, open source (GPLv2) since 2012
- Now a **world-wide community** (> 400 unique contributors, close to 3,000 PRs per year)
- ~3,500 different software projects supported (excl. versions) + ~2,000 “extensions”

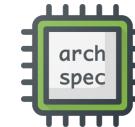


# Software layer

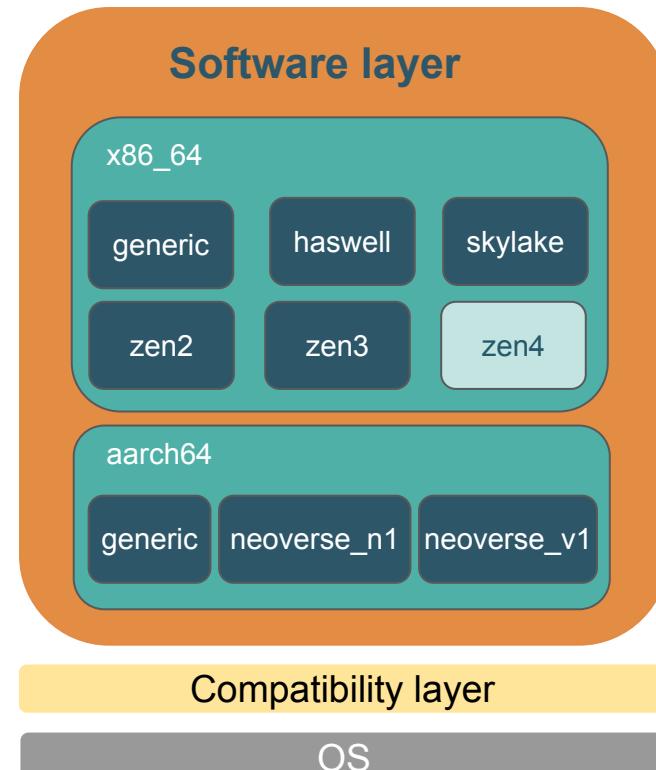
[github.com/EESSI/software-layer](https://github.com/EESSI/software-layer)



Lmod

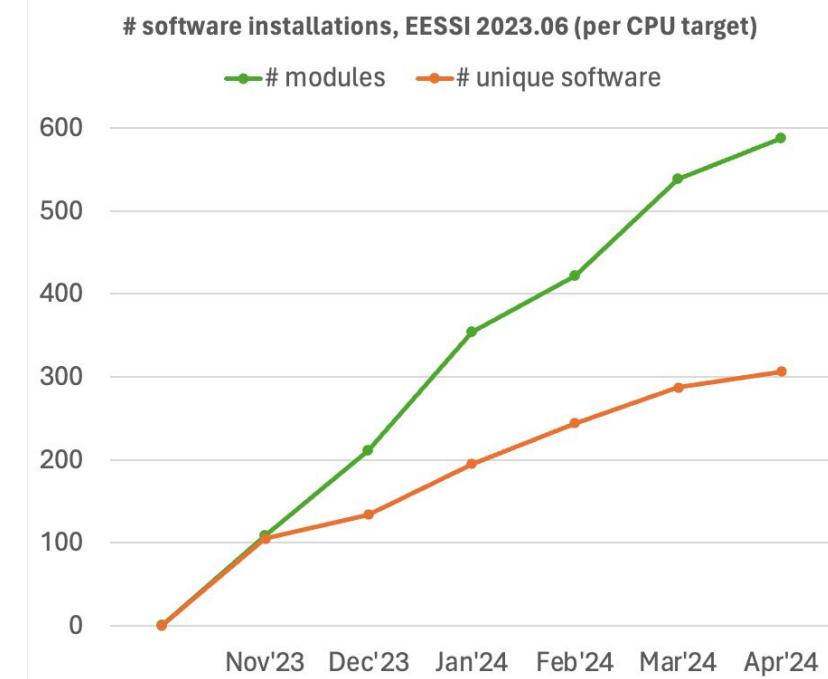


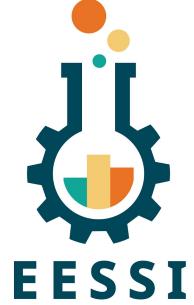
- Installations of scientific software applications
- **Optimized for specific CPU targets**
- Works on any client system running Linux, since we only link to libraries in compat layer
- Built using [EasyBuild](#)
- Environment modules as user interface (via [Lmod](#))
- Detection of host CPU via [archspec](#) (Python) or archdetect (bash)
- **Best subset of software installations for host CPU is automatically selected**



# Overview of installed software

- ~600 software installations available per CPU target, and increasing every day
  - Including ESPResSo, GROMACS, LAMMPS, OpenFOAM, PyTorch, R, QuantumESPRESSO, TensorFlow, WRF
  - [eessi.io/docs/available\\_software](https://eessi.io/docs/available_software) (coming soon!)
- Includes (**but is not limited to!**) applications specific to the MultiXscale CoE
- Focus on recent compiler toolchains: currently targeting foss/2023a and foss/2023b





# Getting access to EESSI

- Native installation of CernVM-FS (*requires admin privileges*)  
[eessi.io/docs/getting\\_access/native\\_installation](https://eessi.io/docs/getting_access/native_installation)
- Using a container (via Apptainer)  
[eessi.io/docs/getting\\_access/eessi\\_container](https://eessi.io/docs/getting_access/eessi_container)
- Via cvmfsexec  
[github.com/cvmfs/cvmfsexec](https://github.com/cvmfs/cvmfsexec)

**To check whether you have access to EESSI:**

```
ls /cvmfs/software.eessi.io
```

# Accessing EESSI via CernVM-FS



```
# Native installation
# Installation commands for RHEL-based distros
# like CentOS, Rocky Linux, Almalinux, Fedora, ...

# install CernVM-FS
sudo yum install -y
https://ecsft.cern.ch/dist/cvmfs/cvmfs-release/cvmfs-release-latest.noarch.rpm
sudo yum install -y cvmfs

# create client configuration file for CernVM-FS
# (no proxy, 10GB local CernVM-FS client cache)
sudo bash -c "echo 'CVMFS_CLIENT_PROFILE=\"single\"' > /etc/cvmfs/default.local"
sudo bash -c "echo 'CVMFS_QUOTA_LIMIT=10000' >> /etc/cvmfs/default.local"

# Make sure that EESSI CernVM-FS repository is accessible
sudo cvmfs_config setup
```

Alternative ways of accessing EESSI are available, via a container image, via cvmfsexec, ...  
[eessi.io/docs/getting\\_access/native\\_installation](https://eessi.io/docs/getting_access/native_installation) - [eessi.io/docs/getting\\_access/eessi\\_container](https://eessi.io/docs/getting_access/eessi_container)

# Using EESSI

[eessi.io/docs/using\\_eessi/eessi\\_demos](https://eessi.io/docs/using_eessi/eessi_demos)



```
/cvmfs/software.eessi.io/versions/2023.06/software
`-- linux
    |-- aarch64
    |   |-- generic
    |   |-- neoverse_n1
    |   `-- neoverse_v1
    '-- x86_64
        |-- amd
        |   |-- zen2
        |   `-- zen3
        |-- generic
        '-- intel
            |-- haswell
            '-- skylake_avx512
                |-- modules
                '-- software
```

```
$ source /cvmfs/software.eessi.io/versions/2023.06/init/bash
Found EESSI pilot repo @
/cvmfs/software.eessi.io/versions/2023.06!
archdetect says x86_64/amd/zen3
Using x86_64/amd/zen3 as software subdirectory
...
Environment set up to use EESSI pilot software stack, have fun!

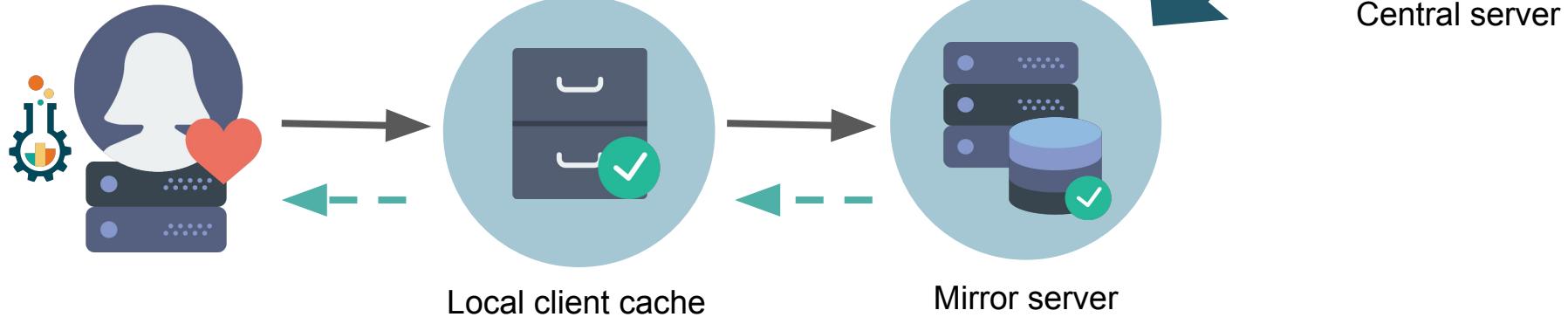
{EESSI 2023.06} $ module load R/4.3.2-gfbf-2023a

{EESSI 2023.06} $ which R
/cvmfs/software.eessi.io/versions/2023.06/software/linux/x86_64/
amd/zen3/software/R/4.3.2-gfbf-2023a/bin/R

{EESSI 2023.06} $ R --version
R version 4.3.2
```

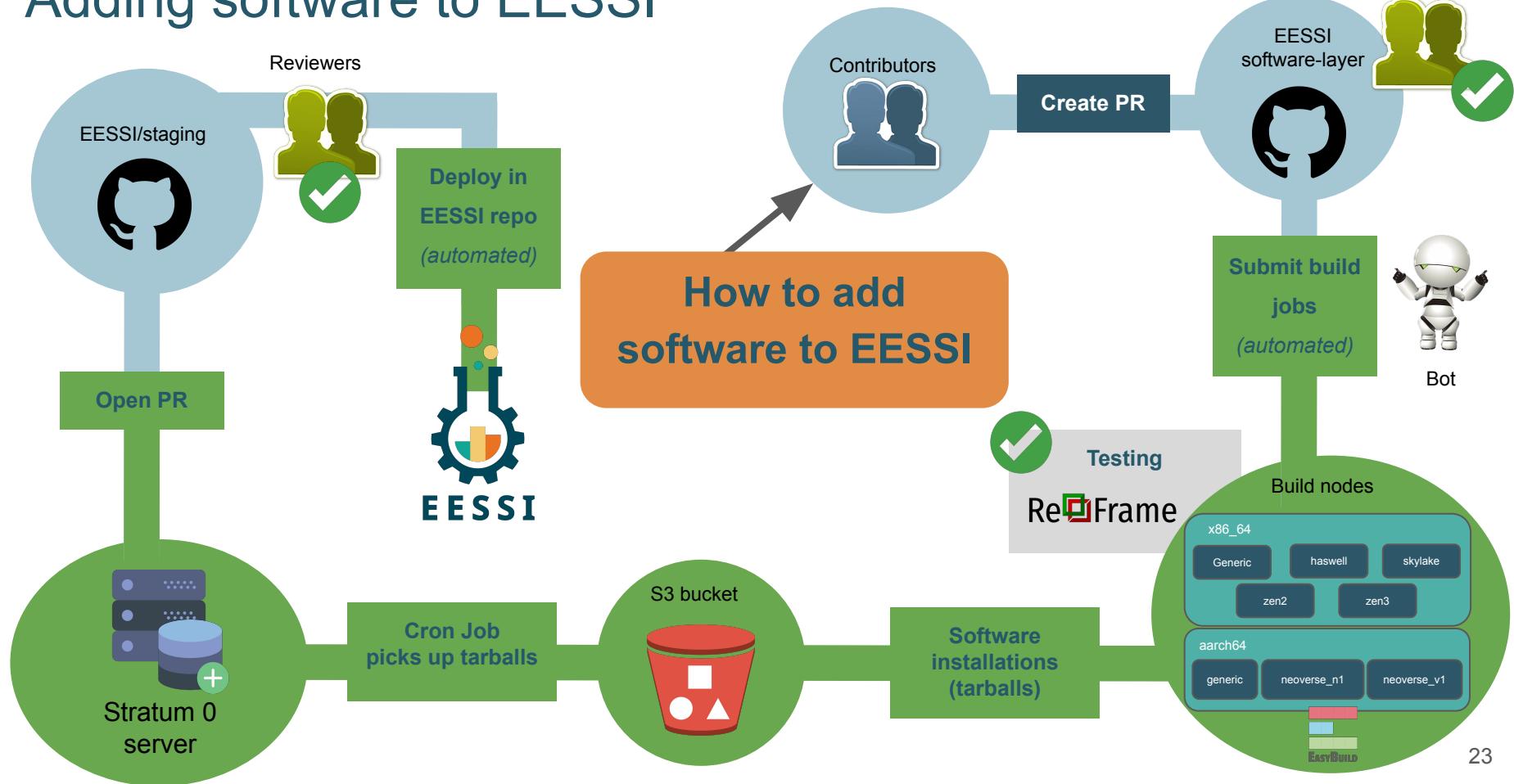
# The EESSI User Experience

```
$ source /cvmfs/software.eessi.io/versions/2023.06/init/bash  
{EESSI 2023.06} $ module load GROMACS/2024.1-foss-2023b  
{EESSI 2023.06} $ gmx mdrun ...
```



EESSI provides **on-demand streaming**  
of (scientific) software (like music, TV-series, ...)

# Adding software to EESSI



# Virtual Slurm clusters via Magic Castle

[https://github.com/ComputeCanada/magic\\_castle](https://github.com/ComputeCanada/magic_castle)



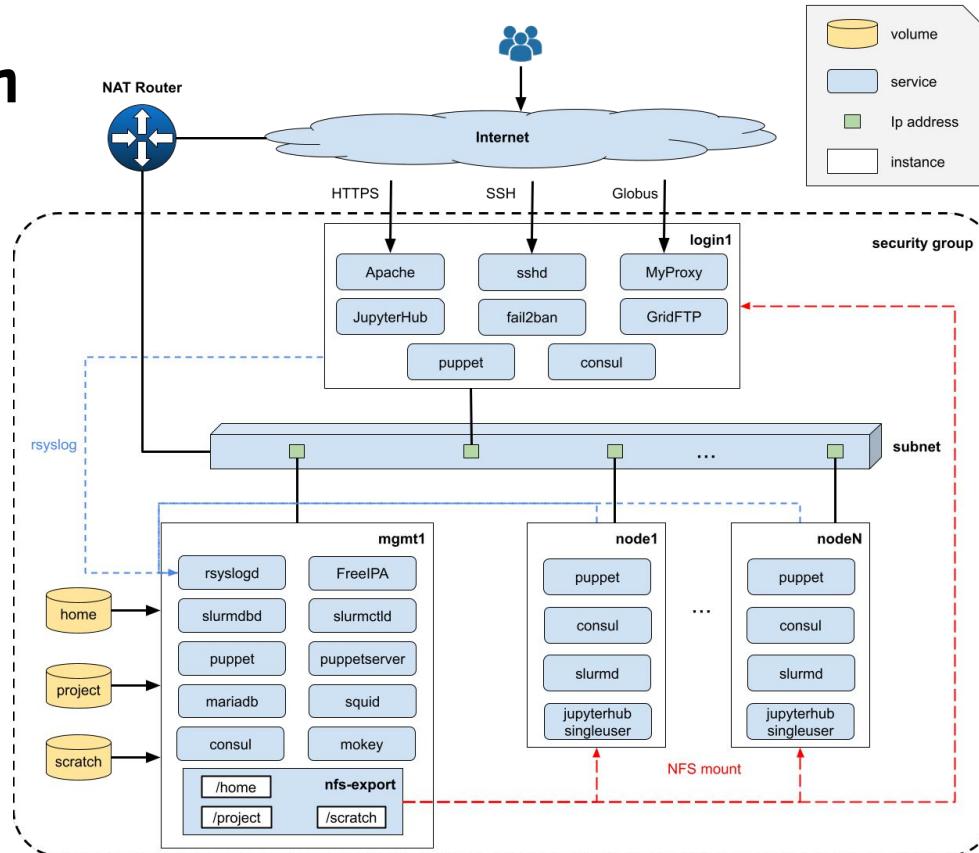
- Slurm clusters were created using Magic Castle (in a cloud agnostic way)
- Used for building all the software installations that are included in EESSI
  - Via build-test-deploy bot: <https://github.com/EESSI/eessi-bot-software-layer>
- Different partitions for different CPU targets (x86\_64/amd/zen3, aarch64/neoverse\_v1, ...)
- Nodes are spun up automatically as jobs appear, and destroyed after ~15min of idle time
- Eventually will also be used for more extensive testing
  - Build & test in EESSI build container (currently Debian 11)
  - Test in other OS (e.g. Rocky Linux), on totally different system/cloud infra

# Virtual Slurm clusters via Magic Castle

[https://github.com/ComputeCanada/magic\\_castle](https://github.com/ComputeCanada/magic_castle)



plan



apply





# NVIDIA GPU support in EESSI

- Initial support for CUDA software is in place in EESSI version 2023.06
- Detailed documentation available at [eessi.io/docs/gpu](https://eessi.io/docs/gpu)
- Problems:
  - 1) We don't know where the NVIDIA GPU driver libraries are in host OS...
  - 2) We can not redistribute the full CUDA installation due to EULA (only runtime libs)
- In EESSI, we provide scripts to deal with both these problems:
  - 1) `link_nvidia_host_libraries.sh` to link GPU driver libraries “into” EESSI (requires write access to `/cvmfs/software.eessi.io/host_injections` variant symlink)
  - 2) `install_cuda_host_injections.sh` installs full CUDA to subdirectory of `/cvmfs/software.eessi.io/host_injections`

# EESSI test suite - Why?

ReFrame  
[eessi.io/docs/test-suite](http://eessi.io/docs/test-suite)



- **Ensure quality of the software installations provided by EESSI**
- Provide more insights into performance of the built software:
  - Not only right after the build, but also over time.
  - Performance aspects cover both single-node and multi-node scaling.
- Indicate system inefficiencies with respect to change in system software at the OS level.
- Provide end-users (scientists) with indicative performance numbers of the software for various hardware architectures (both CPU and accelerators).

# EESSI test suite - What & how?

ReFrame  
[eessi.io/docs/test-suite](http://eessi.io/docs/test-suite)



- A suite of **portable** tests for scientific software, using ReFrame
- EasyBuild already covers unit tests and smoke tests for various applications
- Type of tests to be covered under the test suite:
  - *Integration tests*: Application tests on the build node.
  - *System tests*: Application tests on a completely different system than the build node.
  - *Periodic tests*: Same as system tests, run periodically to check the performance status of the built applications in the central software stack.

# EESSI test suite - Current status

ReFrame  
[eessi.io/docs/test-suite](https://eessi.io/docs/test-suite)



- Version 0.2.0 includes tests for **GROMACS**, **TensorFlow**, **OSU Micro Benchmarks**.
- The test-suite is **integrated in the CI pipeline** used to build and deploy the software.
  - Filtering various tests based on the build (WIP).
- Each software contains **multiple tests** based on the size of the test and scalability.
- The **node configuration is abstracted** in each of the tests, auto-detect output and the config file inputs that the end user provides are used to determine the scales and the type of tests that can be run.
- **Periodic tests** are constantly running in AWS Magic Castle cluster and EuroHPC systems such as Vega apart from site specific systems in Belgium and Netherlands.

# EESSI test suite - What is planned/WIP?

- **More tests** (WIP): LAMMPS, QuantumESPRESSO, CP2K, OpenFOAM, ESPResSo, ...
- Inserting **performance references** based on system arch and configuration.
- **Test dashboard** with results from periodic tests running on multiple systems.
- Integrating more low/mid level application tests that normally act as dependencies rather than end-user applications such as OpenBLAS, MKL etc.
- Further documentation, training sessions, etc.

# EESSI test suite - Contributors



- Test suite is being developed in scope of MultiXscale WP 1 (SURF, UGent, RUG, UB)
- Several external contributors already (Vrije Universiteit Brussel, CECAM, ...)
- See also MultiXscale deliverable D1.2 “Plan for the design of a portable test suite”  
<https://zenodo.org/records/10451718>

More info (installation, configuration, usage) via [eessi.io/docs/test-suite](https://eessi.io/docs/test-suite)

# Problems we have found through software testing (1/2)



## Hang/crash in Open MPI's smcuda Byte Transport Layer (BTL) component

- See <https://gitlab.com/eessi/support/-/issues/41>
- Upstream issue: <https://github.com/open-mpi/ompi/issues/12270>
- Causes hanging or failing tests for FFTW, OpenFOAM, ESPResSo, ...
- Only happens on Arm Neoverse V1 (AWS Graviton 3)
- Fixed by Luke Robinson (AWS), see <https://github.com/open-mpi/ompi/pull/12338>
- Fix will be included in upcoming Open MPI release (v4.1.7+)
- Patch already applied in Open MPI installations included in EESSI 2023.06

# Problems we have found through software testing (2/2)



## Failing tests in GROMACS test suite

- See <https://gitlab.com/eessi/support/-/issues/47>
- Filesystem race when running tests concurrently ([GROMACS PR #4066](#))
- **Bug in SVE support, leading to (very) wrong results for several tests**
  - See <https://gitlab.com/gromacs/gromacs/-/issues/5057>
  - Works fine on A64FX (512-bit SVE), but problem on Graviton 3 + NVIDIA Grace!
  - WIP fix in [https://gitlab.com/gromacs/gromacs/-/merge\\_requests/4299](https://gitlab.com/gromacs/gromacs/-/merge_requests/4299)
  - Will be fixed in upcoming GROMACS release (2024.2?)



# Leveraging EESSI in CI environment

Using EESSI in GitHub Actions is trivial (and works *really* well):

```
name: ubuntu_gromacs
on: [push, pull_request]
jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v2
      - uses: eessi/github-action-eessi@v3
        with:
          eessi_stack_version: '2023.06'
      - name: Test EESSI
        run:
          module load TensorFlow/2.13.0-foss-2023a
          python test_with_tensorflow.py
    shell: bash
```

[github.com/EESSI/github-action-eessi](https://github.com/EESSI/github-action-eessi)



# Leveraging EESSI GitHub Action



Summary

Jobs

ubuntu

succeeded yesterday in 31m 58s

Search logs

Setup EESSI

Checkout repository

Install dependencies

Run module load ESPResSo/4.2.1-foss-2023a

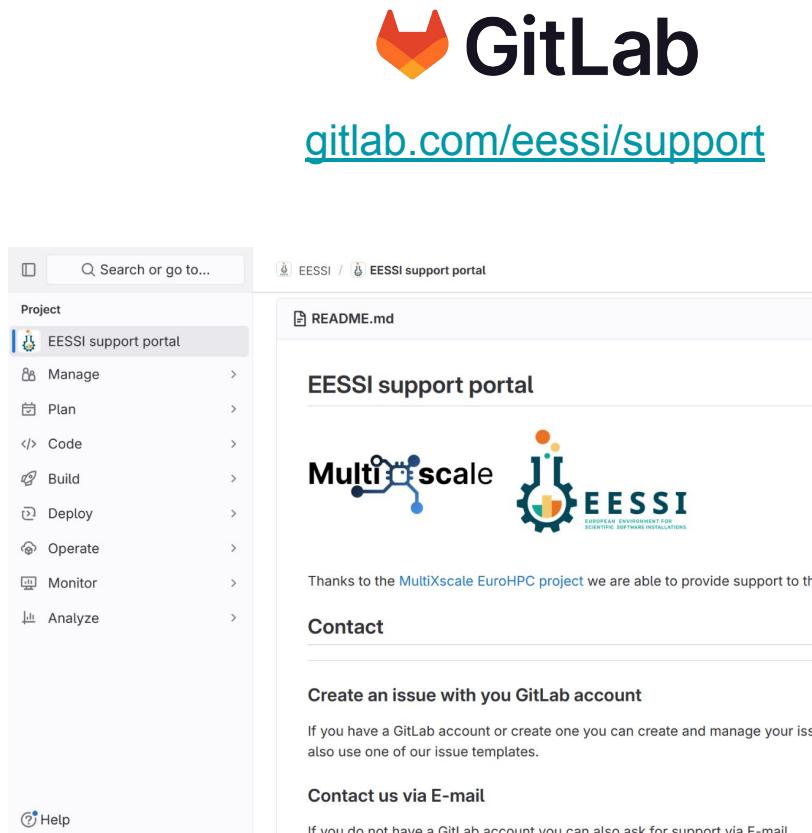
Environment set up to use EESSI (2023.06), have fun!

<https://github.com/pyMBE-dev/pyMBE/blob/main/.github/workflows/testsuite.yml>

<https://github.com/pyMBE-dev/pyMBE/actions/runs/8815523092/job/24197651600>

# Getting support for EESSI

- Via GitLab, or via email: [support@eessi.io](mailto:support@eessi.io)
- Report problems
- Ask questions
- Request software
- Get help with contributing
- Suggest features
- Confidential tickets possible (security issues, ...)



The screenshot shows the GitLab interface for the EESSI support portal. The left sidebar lists project management tasks: Manage, Plan, Code, Build, Deploy, Operate, Monitor, and Analyze. The main content area displays the README.md file, which includes the EESSI support portal logo, the MultiXscale logo, and a message of thanks to the MultiXscale EuroHPC project. It also provides contact information and links for creating issues and contacting via email.

Project

EESSI support portal

- Manage >
- Plan >
- Code >
- Build >
- Deploy >
- Operate >
- Monitor >
- Analyze >

Help

EESSI / EESSI support portal

README.md

## EESSI support portal

MultiXscale EESSI

Thanks to the [MultiXscale EuroHPC project](#) we are able to provide support to the u

### Contact

### Create an issue with your GitLab account

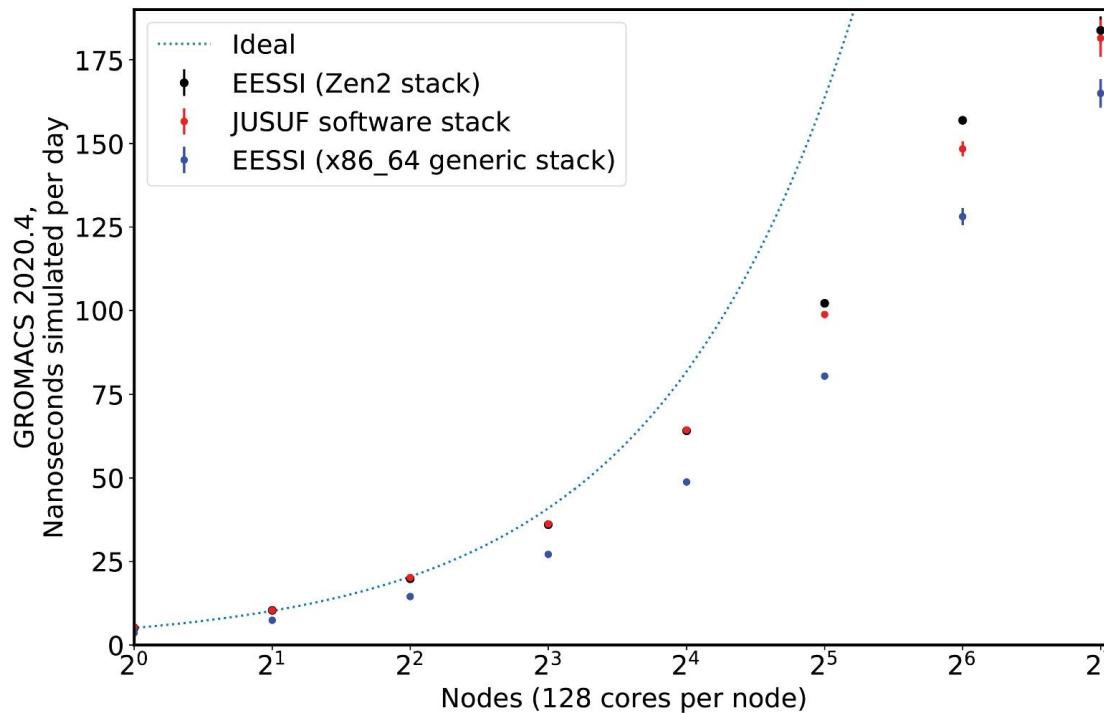
If you have a GitLab account or create one you can create and manage your issue also use one of our issue templates.

### Contact us via E-mail

If you do not have a GitLab account you can also ask for support via E-mail.

Dedicated support team, thanks to EuroHPC Centre-of-Excellence

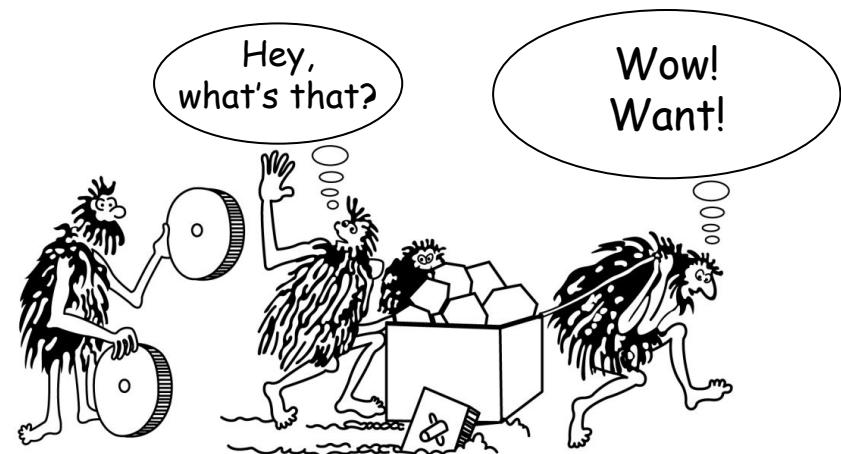




Paper includes **proof-of-concept performance evaluation** compared to system software stack, performed at JUSUF @ JSC using GROMACS 2020.4, up to 16,384 cores (CPU-only)

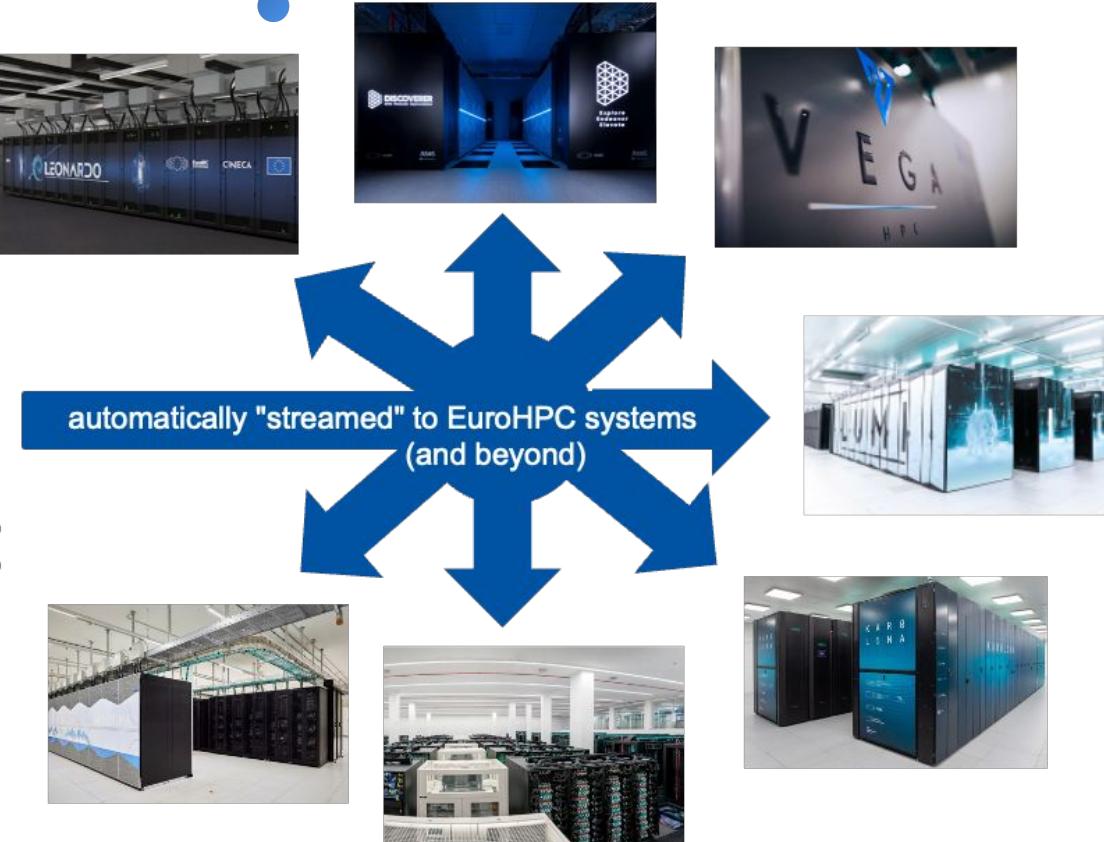
# EESSI in a nutshell

- **On-demand streaming of optimized scientific software installations**
- **Works on any Linux distribution** thanks to EESSI compat layer
- **Uniform software stack** across various systems: laptop, HPC, cloud, ...
- Community-oriented: **let's tackle the challenges we see together!**

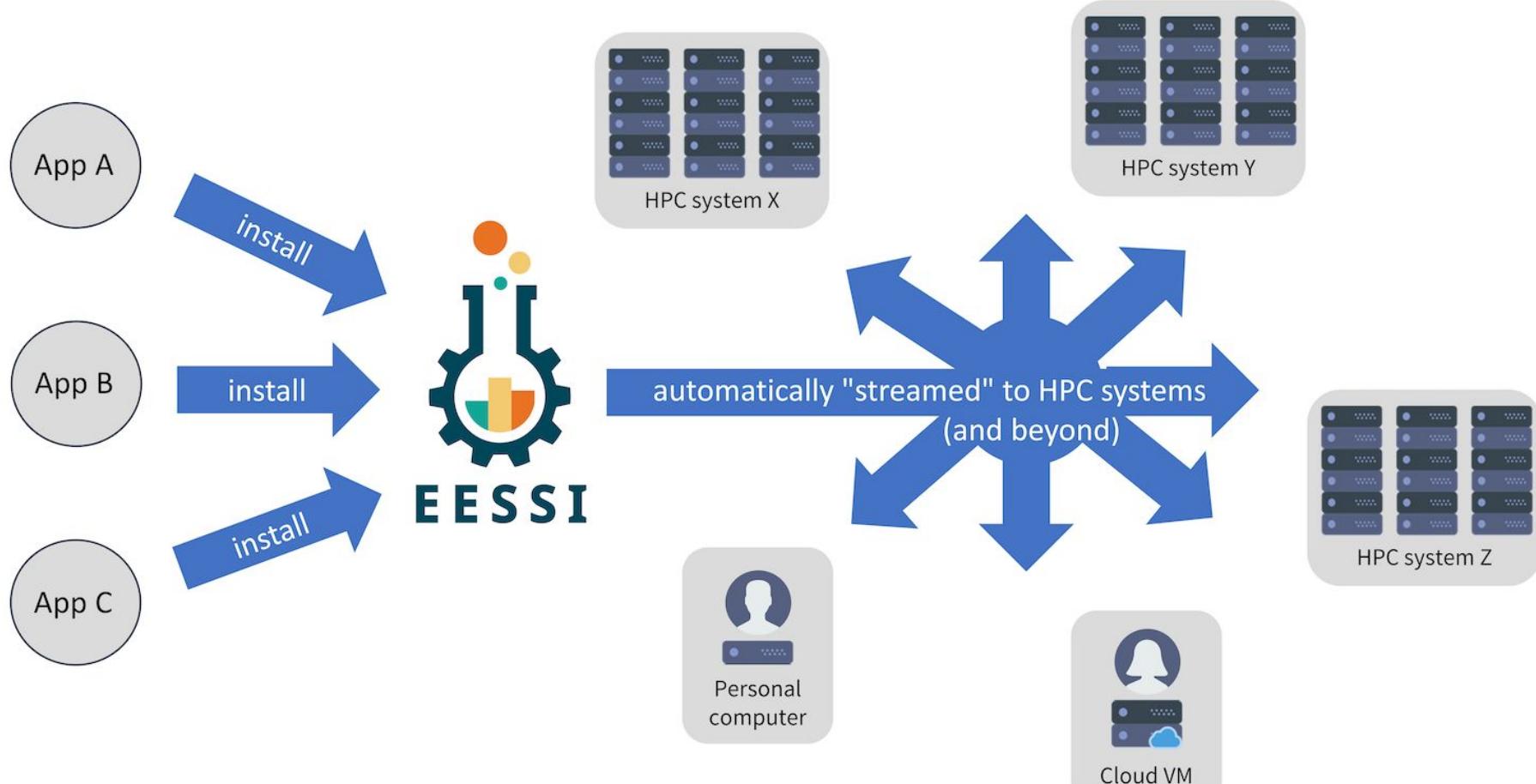


- EuroHPC Centre of Excellence: 4 year project (2023-2026), €6M budget (50% for EESSI)
- Collaboration between EESSI and CECAM: total of 16 partners (academic + industry)
- EESSI focuses on technical aspects: providing a shared stack of scientific software
- Scientific target: multiscale simulations with 3 key use cases
  - Helicopter design and certification for civil transport 
  - Battery applications to support the sustainable energy transition 
  - Ultrasound for non-invasive diagnostics and biomedical applications 

# The role of EESSI in Multi<sup>scale</sup>



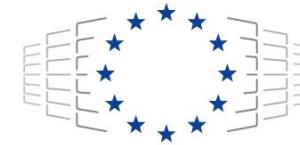
# EESSI as a shared software stack



# Acknowledgements



Co-funded by  
the European Union



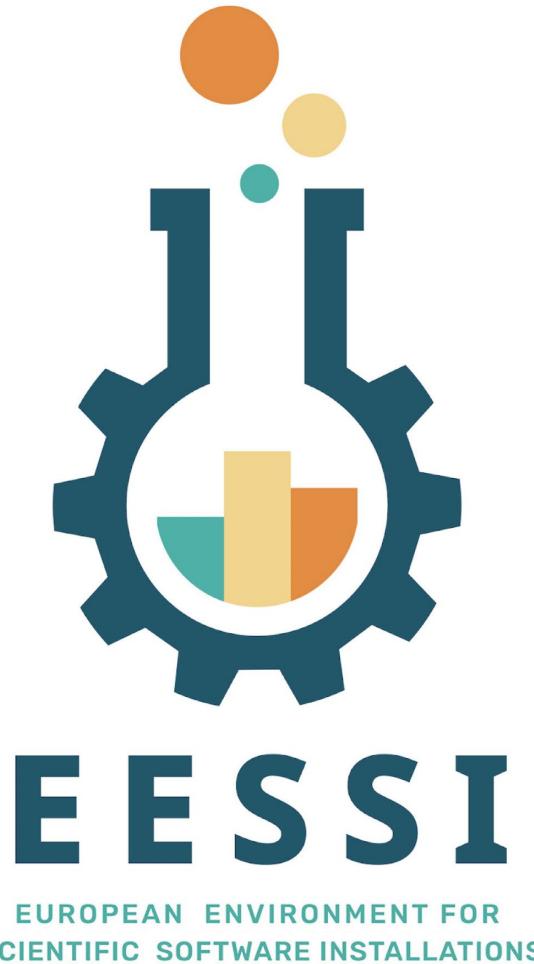
EuroHPC  
Joint Undertaking

- Funded by the European Union. This work has received funding from the European High Performance Computing Joint Undertaking (JU) and countries participating in the project under grant agreement No 101093169.



- Thanks to Amazon Web Services (AWS) and Microsoft Azure for generously sponsoring the EESSI project with cloud credits, feedback, and guidance.





Website: [eessi.io](https://eessi.io)

GitHub: [github.com/eessi](https://github.com/eessi)

Documentation: [eessi.io/docs](https://eessi.io/docs)

YouTube channel: [youtube.com/@eessi\\_community](https://youtube.com/@eessi_community)

Paper (open access): [doi.org/10.1002/spe.3075](https://doi.org/10.1002/spe.3075)

EESSI support portal: [gitlab.com/eessi/support](https://gitlab.com/eessi/support)

Monthly online meetings (first Thursday, 2pm CEST)

**Join our mailing list & Slack channel**