

# Streaming optimized scientific software installations on any Linux distro with EESSI

*PackagingCon'23, Berlin (2023-10-27)*

Kenneth Hoste & Lara Peeters (HPC-UGent)

*kenneth.hoste@ugent.be - 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



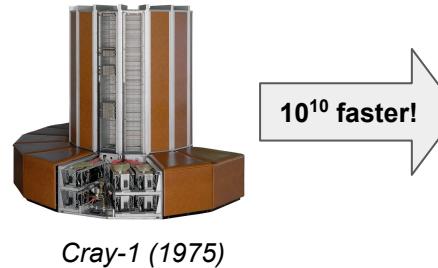
- Hired on the MultiXscale project at Ghent University (Belgium) since May 2023
- Active contributor to **EasyBuild & EESSI**, partner in MultiXscale EuroHPC CoE
- Just getting started, still figuring out the “art” of software packaging

# Context: Scientific Computing & Supercomputers!

- Supercomputers are big, **fast**, expensive
- Used for scientific simulations + much more
- **Multi-tenant:** 1000s of scientists, different profiles



⇒ Wide variety of scientific software required...



Cray-1 (1975)

$10^{10}$  faster!



Frontier @ Oak Ridge Nat. Lab  
(#1 in Top500 of June'2023)

Fastest supercomputers today can perform over  $10^{18}$  64-bit floating-point operations per second (1 exaFLOPS)

- **Performance** is key: faster software means more science gets done (for roughly same cost)



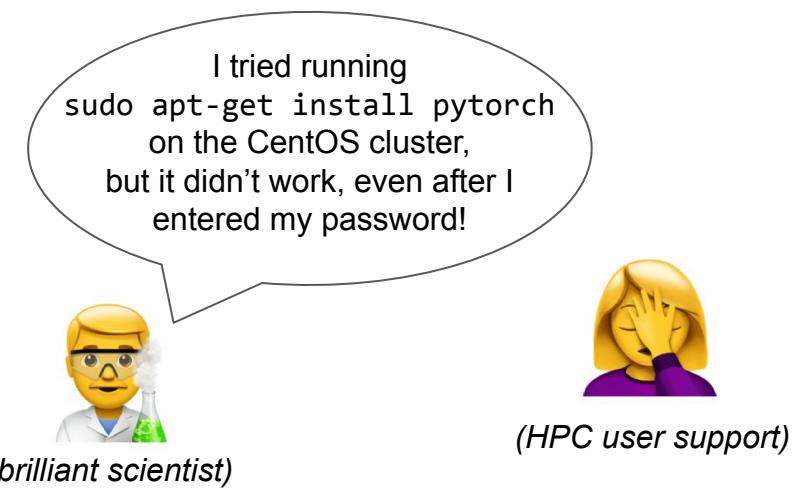
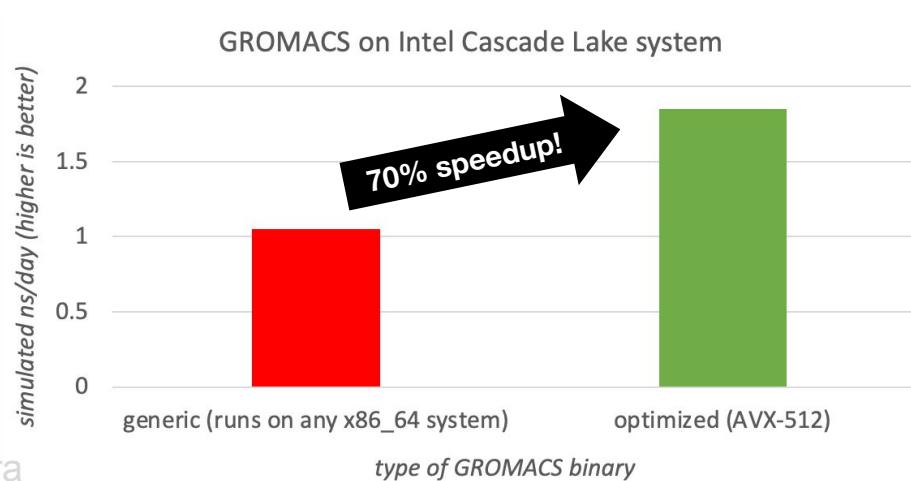
Lara



Supercomputers at Ghent University (Belgium) - [ugent.be/hpc](http://ugent.be/hpc)

# Scientific software is a different breed

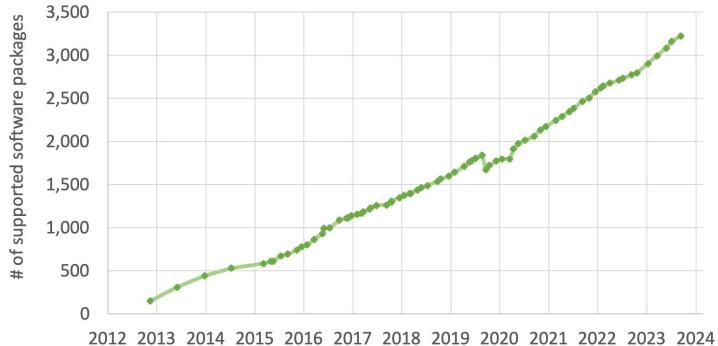
- Scientists are often not trained software engineers (and that's OK)
- They often need help to get the software installed properly (especially on HPC systems)
- **Standard packaging tools do not suffice**, we ~~want~~ need to **build from source**
- Scientific software should be compiled for system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



# EasyBuild (in a nutshell)

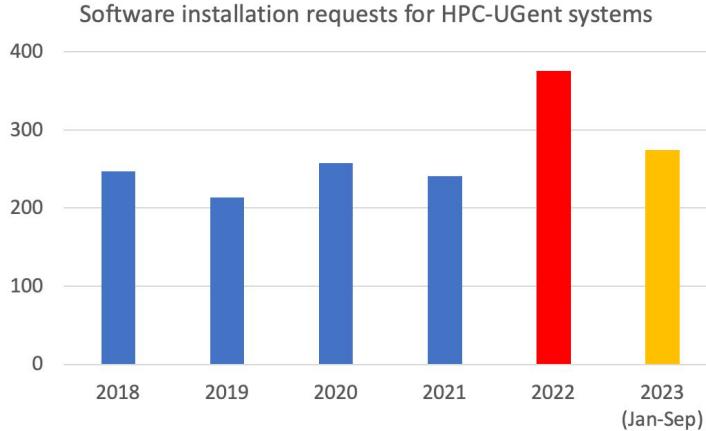
<https://easybuild.io>

- 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 since 2012
- Now a **world-wide community** (> 400 unique contributors, close to 3,000 PRs per year)
- Supports over 3,200 different software projects (excl. versions) + ~2,000 “extensions”



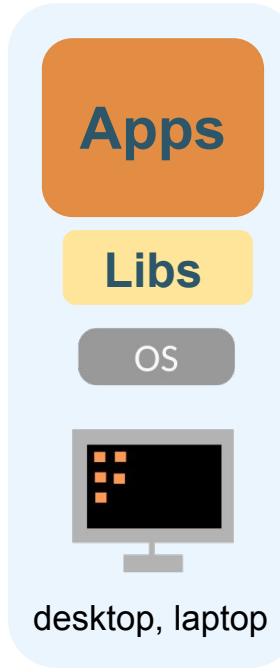
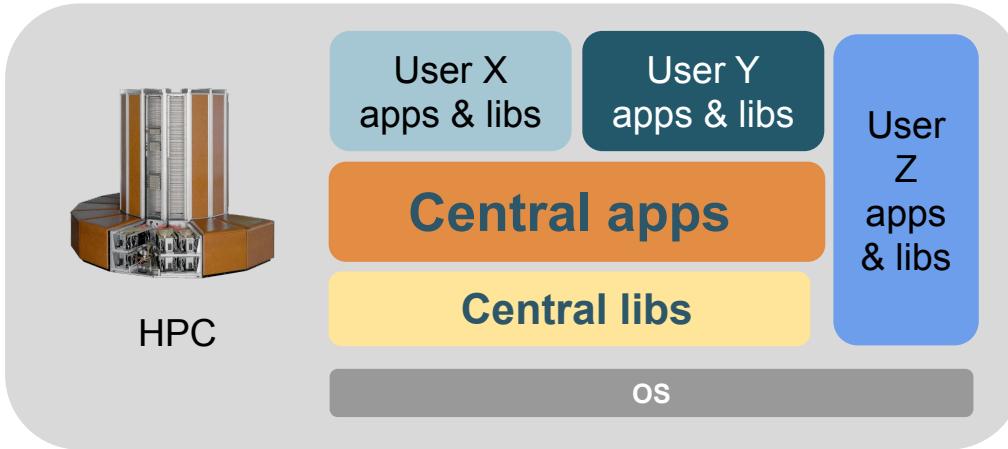
# The changing landscape of scientific computing

- **Explosion of available scientific software** applications (bioinformatics, AI boom, ...)
- 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?)
- In strong contrast: available (wo)manpower in **HPC support teams** is **(still) limited...**

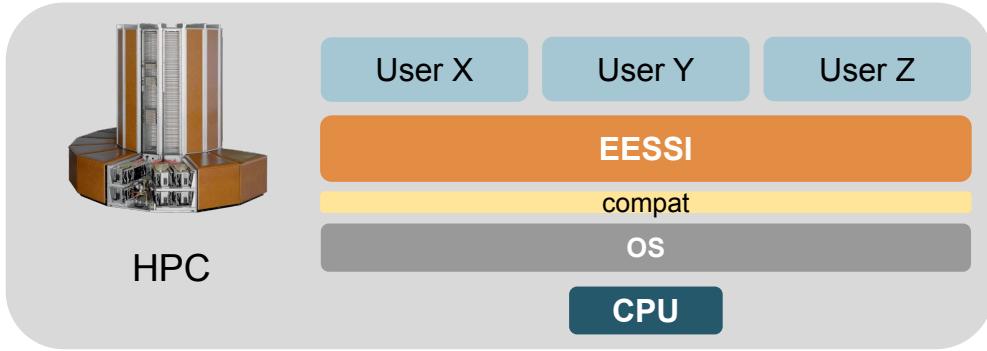
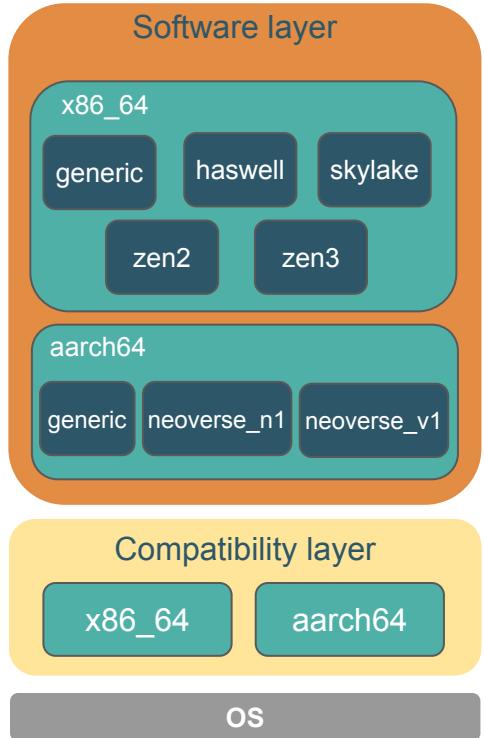


# We need to collaborate more!

- **Too much software** for a single support team to handle
- Different systems (CPU, OS, ...) => different problems
- **EasyBuild is not sufficient anymore...**
- **Duplicate work** across HPC sites and scientists
- **Diverse software stacks** across different platforms



# EESSI to the rescue



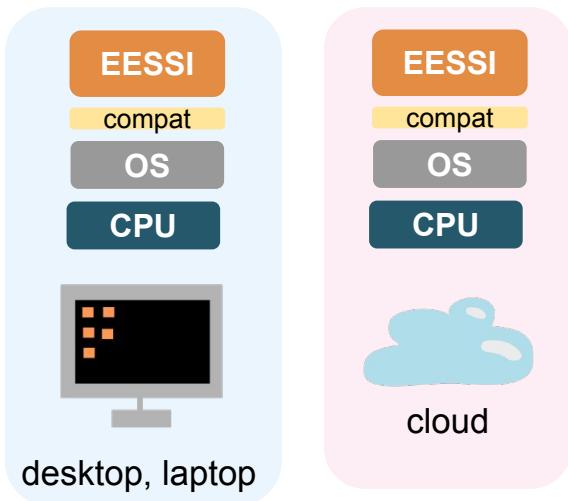
**Shared repository of  
(optimized) scientific  
software installations**

**Same software stack  
everywhere!**

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

[eessi.io/docs](http://eessi.io/docs) (try out the pilot setup!)

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



# High-level overview of EESSI

ReFrame

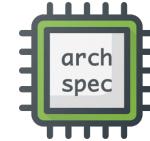
Testing



Host OS provides network & GPU drivers, resource manager (Slurm), ...

## Software layer

Optimized applications + dependencies



## Compatibility layer

Levelling the ground across client OSs



## Filesystem layer

Distribution of the software stack



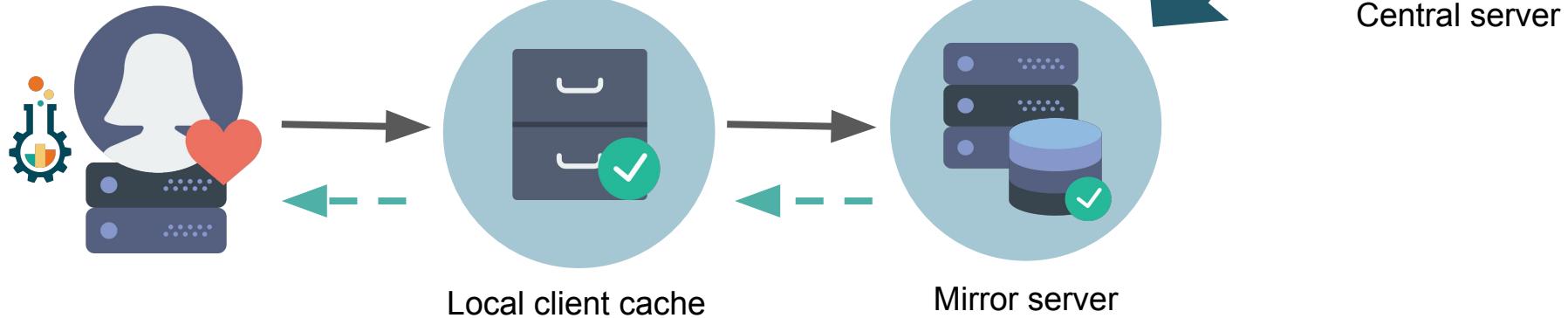
CernVM-FS

Host operating system

- 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 EESSI User Experience

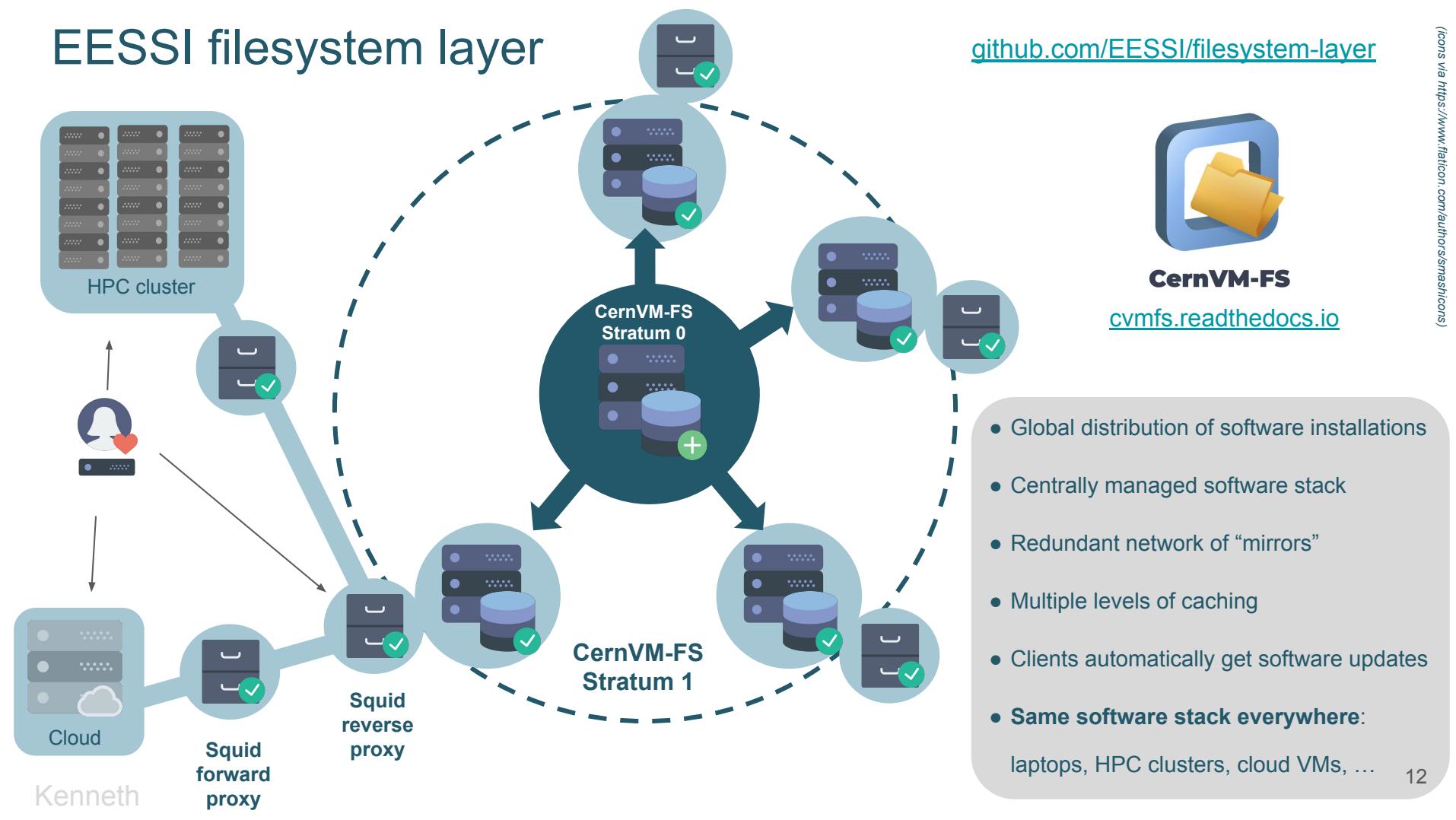
```
$ source /cvmfs/software.eessi.io/latest/init/bash  
[EESSI pilot 2023.06] $ module load GROMACS  
[EESSI pilot 2023.06] $ gmx --help
```



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

# EESSI filesystem layer

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



# EESI compatibility layer

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

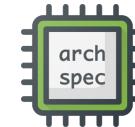
OS

# Software layer

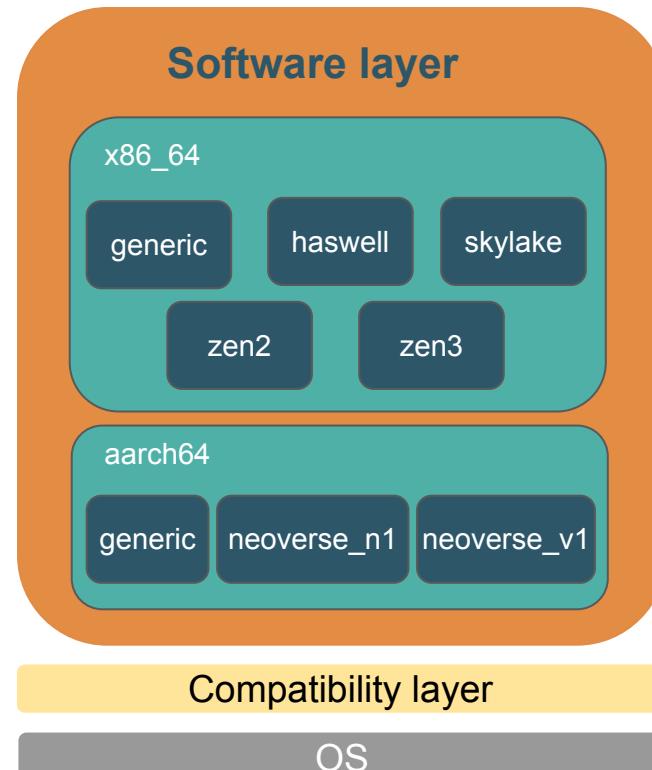
[github.com/EESSI/software-layer](https://github.com/EESSI/software-layer)



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](#)
- **Best subset of software installations for host CPU is automatically selected**



# Software layer (demo)

[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!
archspec says x86_64/intel/skylake_avx512
Using x86_64/intel/skylake_avx512 as software subdirectory
...
Environment set up to use EESSI pilot software stack, have fun!

$ module load R

$ which R
/cvmfs/software.eessi.io/versions/2023.06/software/linux/x86_64/
intel/skylake_avx512/software/R/4.2.1-foss-2022a/bin/R

$ R --version
R version 4.2.1
```

# Software testing in EESSI

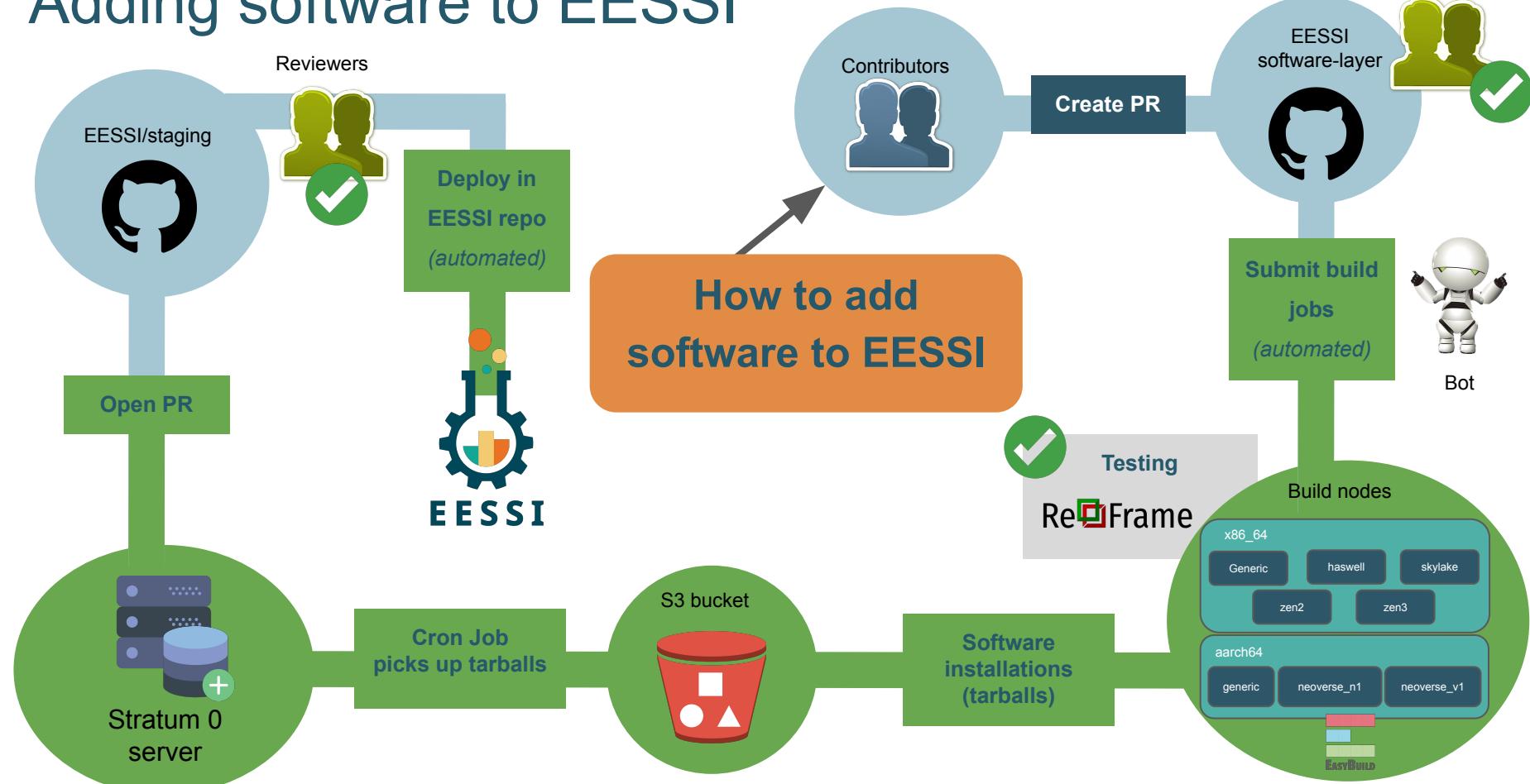


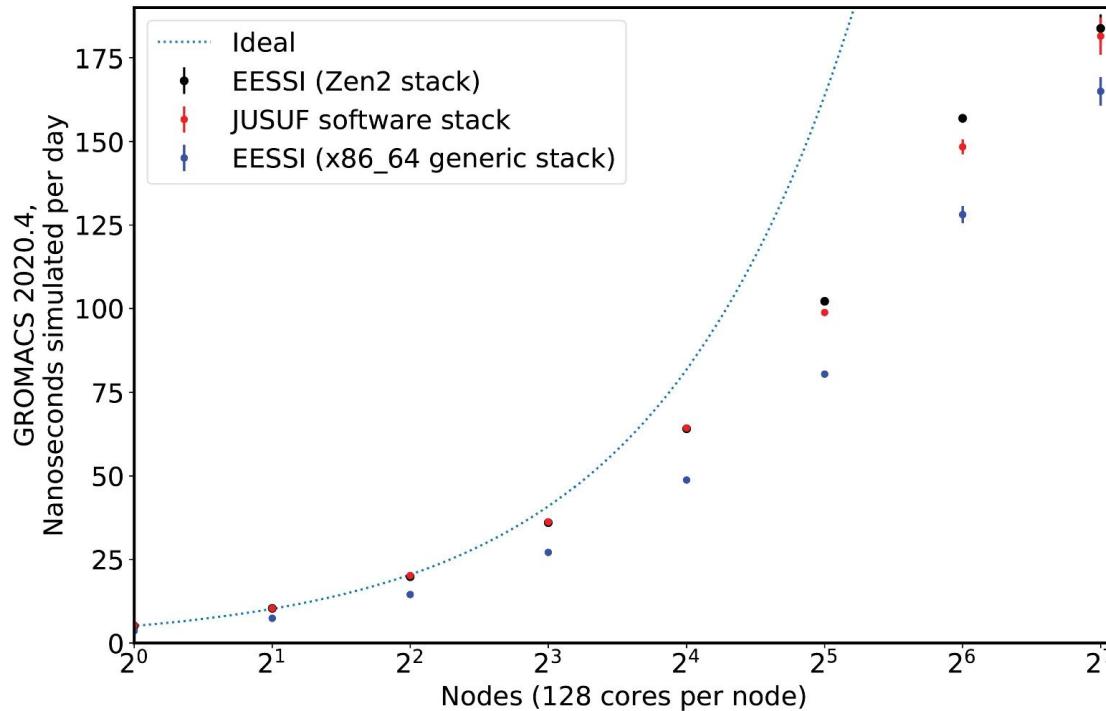
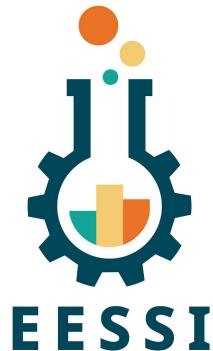
- **Software test suite** is run on build host when building software with EasyBuild
- **Sanity check commands** are run to check that installed software is not horribly broken
- **Portable test suite** based on [ReFrame](#) to evaluate **functional correctness + performance**
  - On host (and OS) that was used to build the software (as part of build procedure)
  - **Also on different host and/or different OS =>** software installations should still work!
- Eventually we also want to do **performance monitoring**
  - Periodic runs of EESSI test suite to catch performance regressions

[github.com/EESSI/test-suite](https://github.com/EESSI/test-suite)

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

# Adding software to EESSI





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

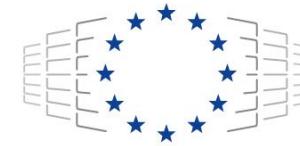
- **On-demand streaming of optimized scientific software installations**
- **Works on any Linux distribution** thanks to EESSI compat layer
- **Uniform software stack** across various systems: laptop, HPC, cloud, ...
- Community-oriented: **let's tackle the challenges we see together!**



# Acknowledgements



Co-funded by  
the European Union



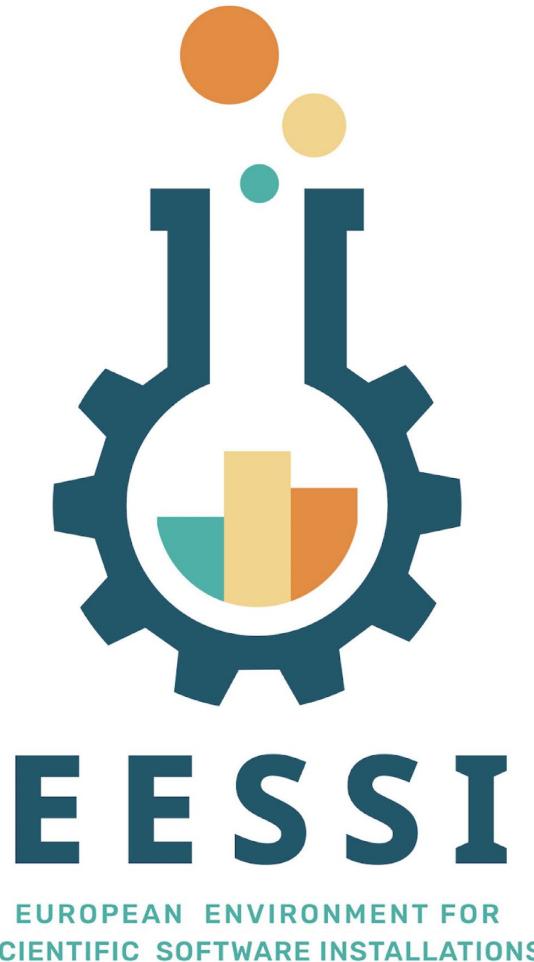
EuroHPC  
Joint Undertaking

- 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.github.io/docs](https://eessi.github.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)

[eessi.io/docs/meetings/2022-09-amsterdam](https://eessi.io/docs/meetings/2022-09-amsterdam)

**Join our mailing list & Slack channel**