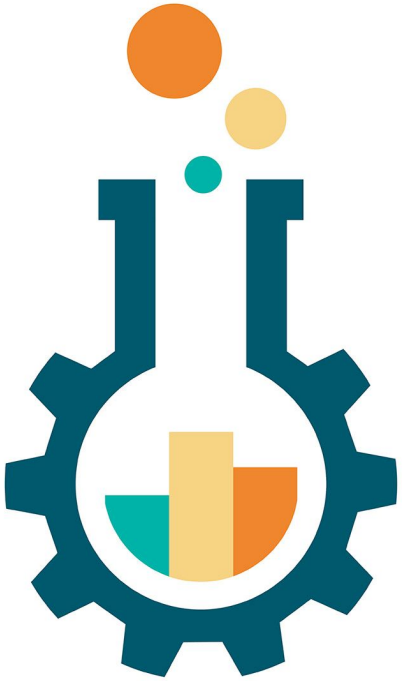
Prepared environment

- This information (and more!) is available at https://hackmd.io/@multixscale/CASTIEL2

- Small AlmaLinux 8 cluster (in the cloud)
  - System will be up until the end of the tutorial (~18:00 CET)

- **You need to create an account!**
  - Signup: https://mokey.cluster.eessi.science/auth/signup
  - Accounts will only be approved for access on the day, so **please record your username/password !**
    - "Reset password" link does **not** work, instead raise any login problem in Slack

- Access via `ssh` or web browser **(pick one and stick to it!)**
  - Shell access: `ssh <username>@cluster.eessi.science`
    - Use login node for hands-on, it has only 16 cores but should be fine to share for small group
  - Via browser: https://cluster.eessi.science
    - Make sure to change default "Time" to 2 hours, increase number of cores to 2 with 6GB RAM



EasyBuild

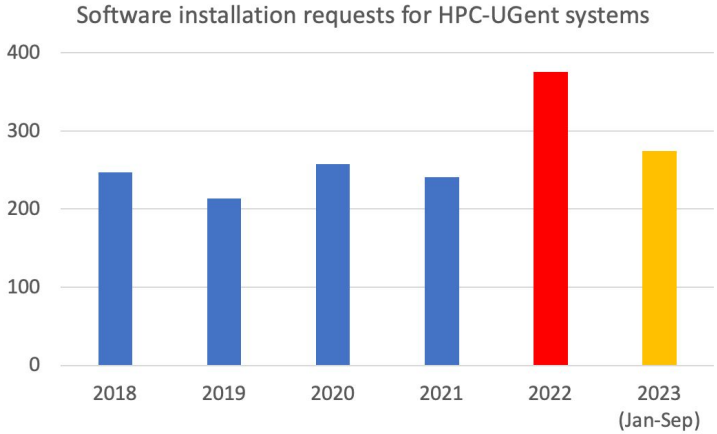
Introduction to EESSI

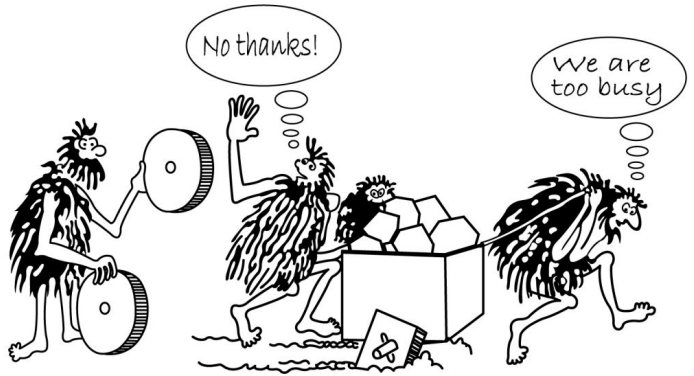
EUROPEAN ENVIRONMENT FOR SCIENTIFIC SOFTWARE INSTALLATIONS

Alan O'Cais
University of Barcelona/CECAM
alan.ocais@cecam.org

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**…

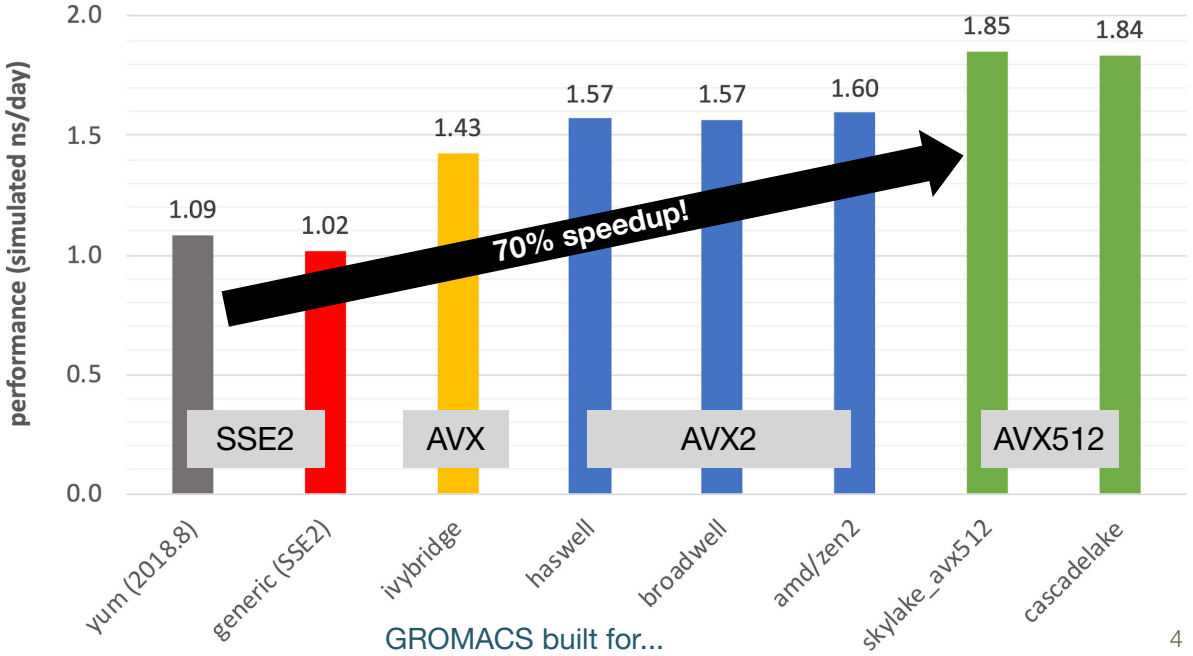




Optimized scientific software installations

- 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**



EESSI in a nutshell

- *European Environment for Scientific Software Installations (EESSI)*
- **Shared repository of (optimized!) scientific software *installations***
- Avoid duplicate work across (HPC) sites 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 (+ WSL, and possibly macOS) and system architecture
  - From laptops and personal workstations to HPC clusters and cloud
  - Support for different CPUs, interconnects, GPUs, etc.
- **Focus on performance, automation, testing, collaboration**

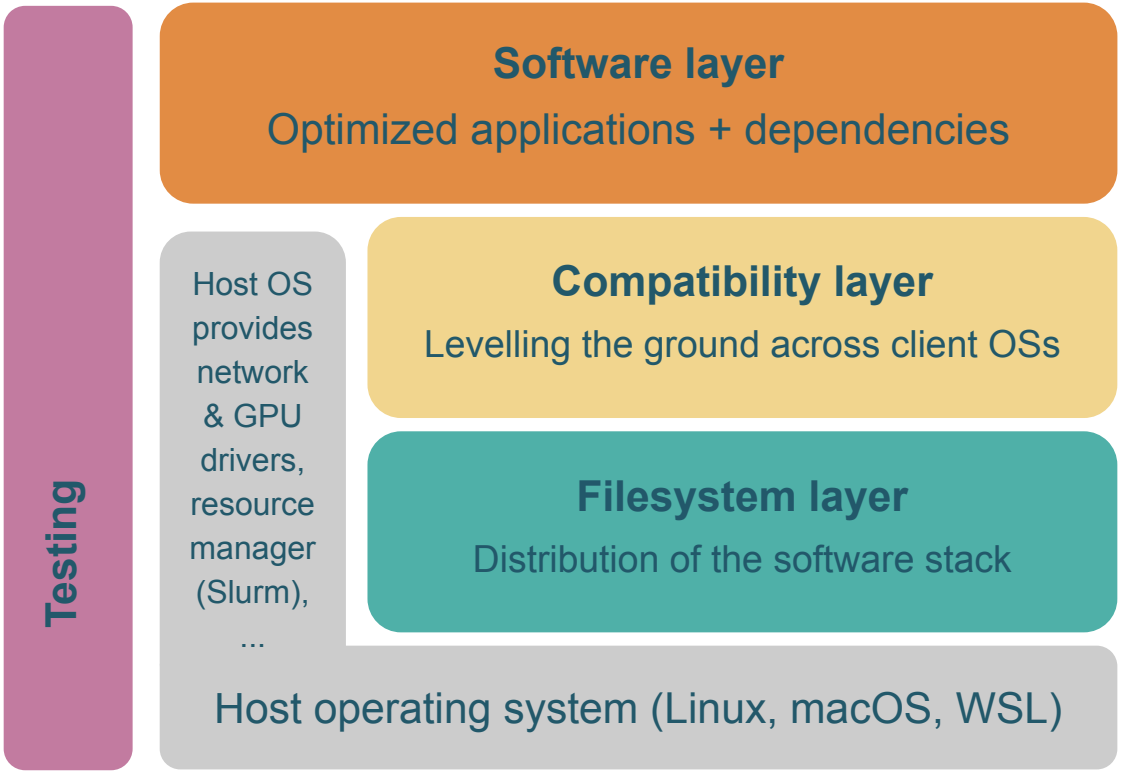
https://www.eessi.io

https://www.eessi.io/docs/

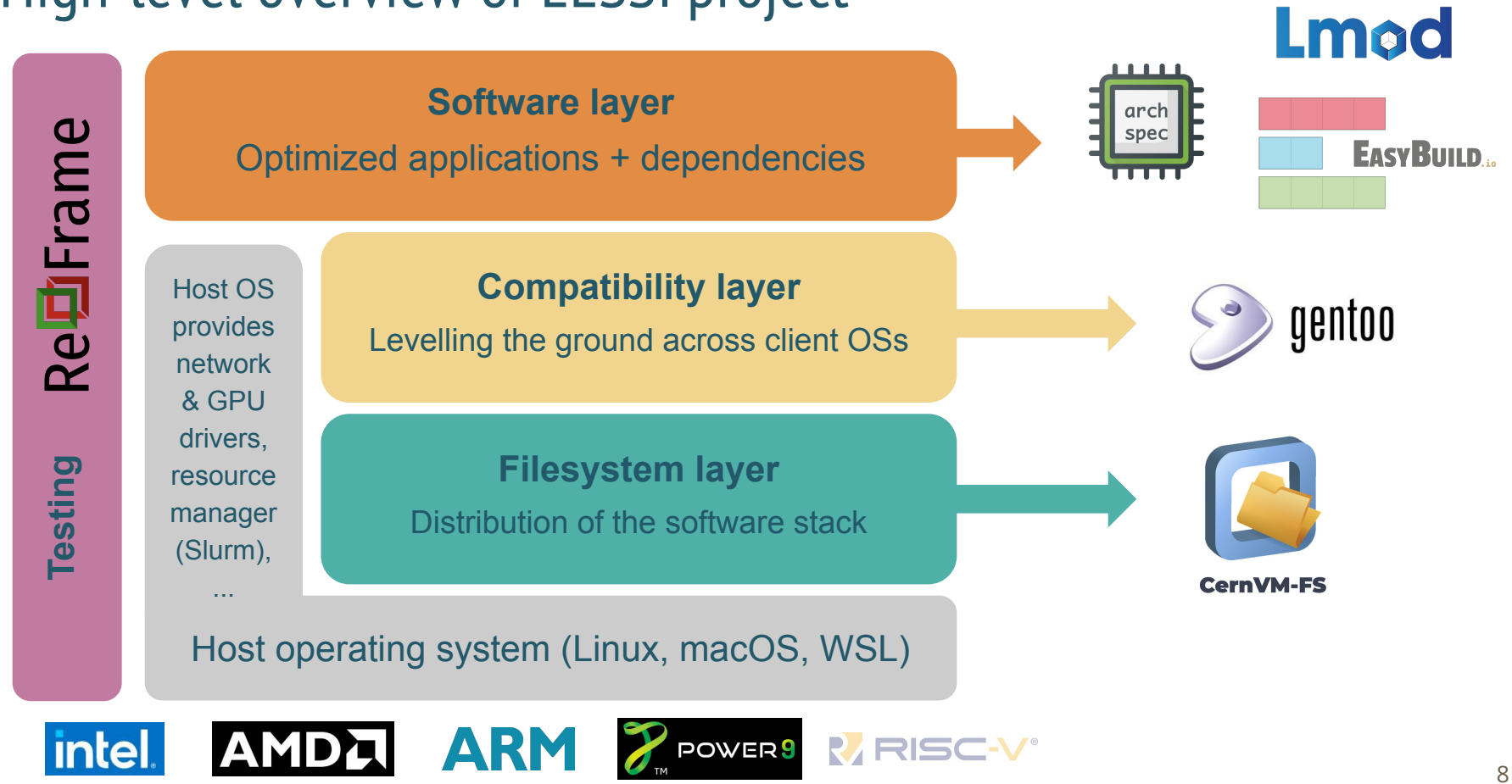
Major goals of EESSI

- **Avoid duplicate work** (for researchers, HPC support teams, …)
  - Tools that automate software installation process (EasyBuild, Spack) are not sufficient
  - Go beyond sharing build recipes => work towards a shared software stack
- Providing a truly **uniform software stack**
  - Use the (exact) same software environment everywhere
  - Without sacrificing performance for “mobility of compute” (like is typically done with containers/conda)
- Facilitate HPC training, development of (scientific) software, …

High-level overview of EESSI project



High-level overview of EESSI project



High-level overview of EESSI project

EESSI

EUROPEAN ENVIRONMENT FOR
SCIENTIFIC SOFTWARE INSTALLATIONS

DEMO

Demo scenario

- Demo: Using an “empty” **Amazon Linux 2023** VM in AWS (**Arm Graviton2**)
  - No CernVM-FS installed, EESSI not available yet, but only takes 2 min.
  - Requires admin rights (`sudo` to install extra packages)
  - Set up EESSI environment by sourcing init script
  - Running EESSI demo script from

https://github.com/EESSI/eessi-demo

Demo: Amazon Linux 2023 Arm VM in AWS (1/2)

- We needed to: https://github.com/EESSI/eessi-demo
  - Install CernVM-FS packages
  - Install EESSI CernVM-FS configuration (`cvmfs-eessi-config*` package)
  - Set up minimal client configuration in `/etc/cvmfs/default.local`
- For production usage (especially large-scale), you should also:
  - Use a squid proxy, next to a local client cache (better start-up performance)
  - Set up your own Stratum-1 mirror server (protection against network disconnects)
  - Also recommended to “be a good citizen” in the EESSI CernVM-FS network

Demo: Amazon Linux 2023 Arm VM in AWS (2/2)

- Once CernVM-FS + EESSI configuration is installed, you're good to go!
- Set up EESSI environment by sourcing the init script, load modules, run.

https://github.com/EESSI/eessi-demo

```
$ ls /cvmfs/pilot.eessi-hpc.org
host_injections latest versions

$ source /cvmfs/pilot.eessi-hpc.org/latest/init/bash
...
Environment set up to use EESSI pilot software stack, have fun!

$ module avail GROMACS TensorFlow OpenFOAM Bioconductor

------ /cvmfs/pilot.eessi-hpc.org/versions/2021.12/software/linux/aarch64/graviton2/modules/all -------

      GROMACS/2020.1-foss-2020a-Python-3.8.2       OpenFOAM/9-foss-2021a                      (D)
      GROMACS/2020.4-foss-2020a-Python-3.8.2 (D)   R-bundle-Bioconductor/3.11-foss-2020a-R-4.0.0
      OpenFOAM/v2006-foss-2020a                    TensorFlow/2.3.1-foss-2020a-Python-3.8.2
      OpenFOAM/8-foss-2020a
```

Demo: Amazon Linux 2023 Arm VM in AWS (2/2)

- Once CernVM-FS + EESSI configuration is installed, you're good to go!

Production version is **`software.eessi.io`** :

`source /cvmfs/software.eessi.io/versions/2023.06/init/bash`

(but doesn't work with all demo examples yet)

```
OpenFOAM/v2006-foss-2020a
OpenFOAM/8-foss-2020a
TensorFlow/2.3.1-foss-2020a-Python-3.8.2
```

EESSI

EUROPEAN ENVIRONMENT FOR
SCIENTIFIC SOFTWARE INSTALLATIONS

Hands-on 1: Running a demo

Hands-on scenario

- On provided infrastructure (**AlmaLinux 8**)
  - EESSI CernVM-FS repository readily available and configured
  - Nodes are pretty low-spec (16 cores, Skylake), front-end has 16 cores...but the examples are also not very demanding
  - Run an EESSI demo script after:

```
git clone https://github.com/EESSI/eessi-demo
```

Try out EESSI yourself using Apptainer!

- Only Apptainer (or Singularity) is required to run the EESSI client container
- **Should work on any Linux distribution**, on Intel/AMD/Arm CPUs
- Detailed instructions available at https://www.eessi.io/docs/pilot

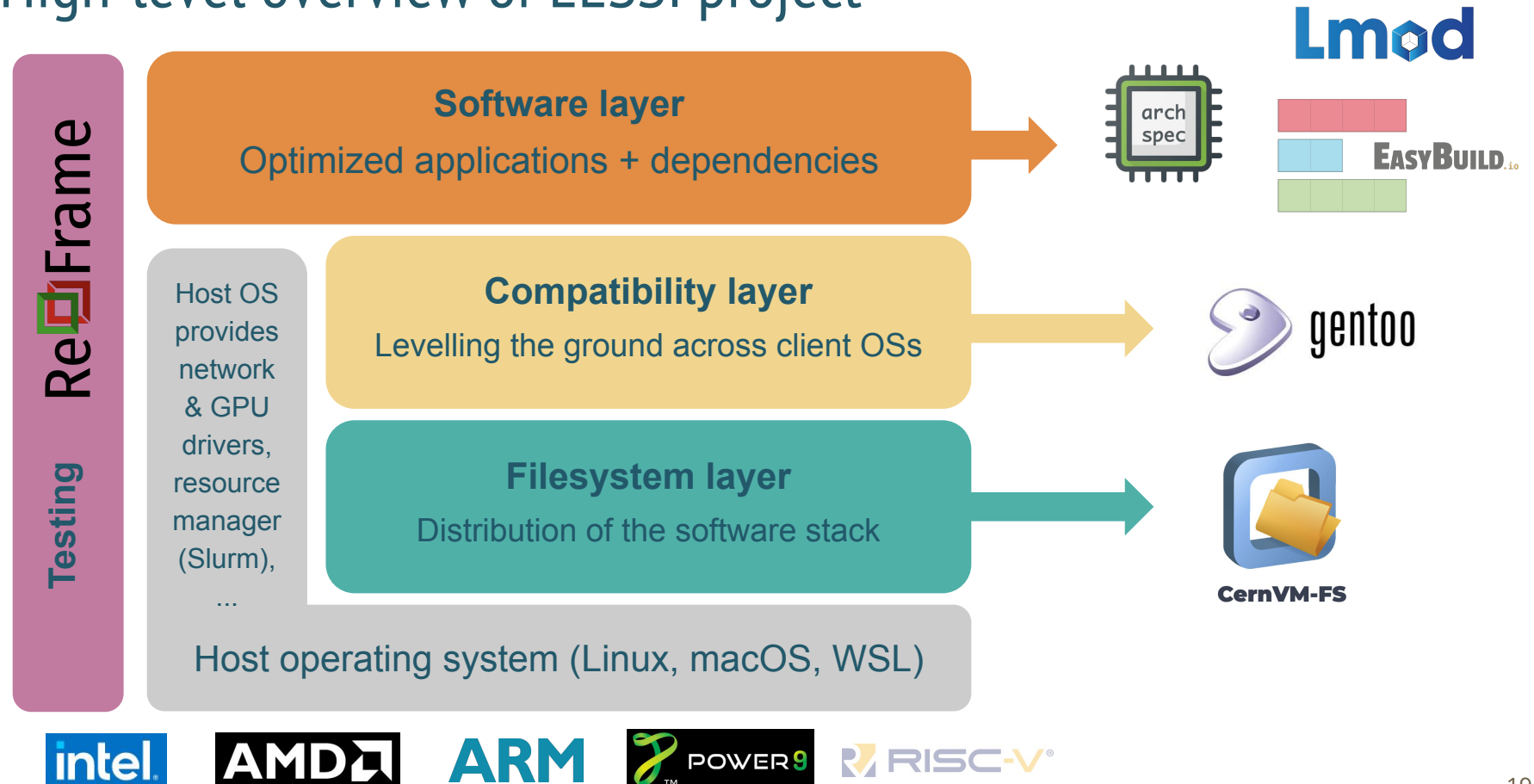
```
$ apptainer shell --fusemount ...
...
Apptainer> ls /cvmfs/software.eessi.io/
host_injections  README.eessi  versions

Apptainer> source /cvmfs/software.eessi.io/versions/2023.06/init/bash

Found EESSI repo @ /cvmfs/software.eessi.io/versions/2023.06!
archdetect says x86_64/generic
Using x86_64/generic as software subdirectory.
Using /cvmfs/software.eessi.io/versions/2023.06/software/linux/x86_64/generic/modules/all as the directory to be
added to MODULEPATH.
Found Lmod configuration file at
/cvmfs/software.eessi.io/versions/2023.06/software/linux/x86_64/generic/.lmod/lmodrc.lua
Initializing Lmod...
Prepending /cvmfs/software.eessi.io/versions/2023.06/software/linux/x86_64/generic/modules/all to $MODULEPATH...
Environment set up to use EESSI (2023.06), have fun!
```

EESSI

EUROPEAN ENVIRONMENT FOR
SCIENTIFIC SOFTWARE INSTALLATIONS

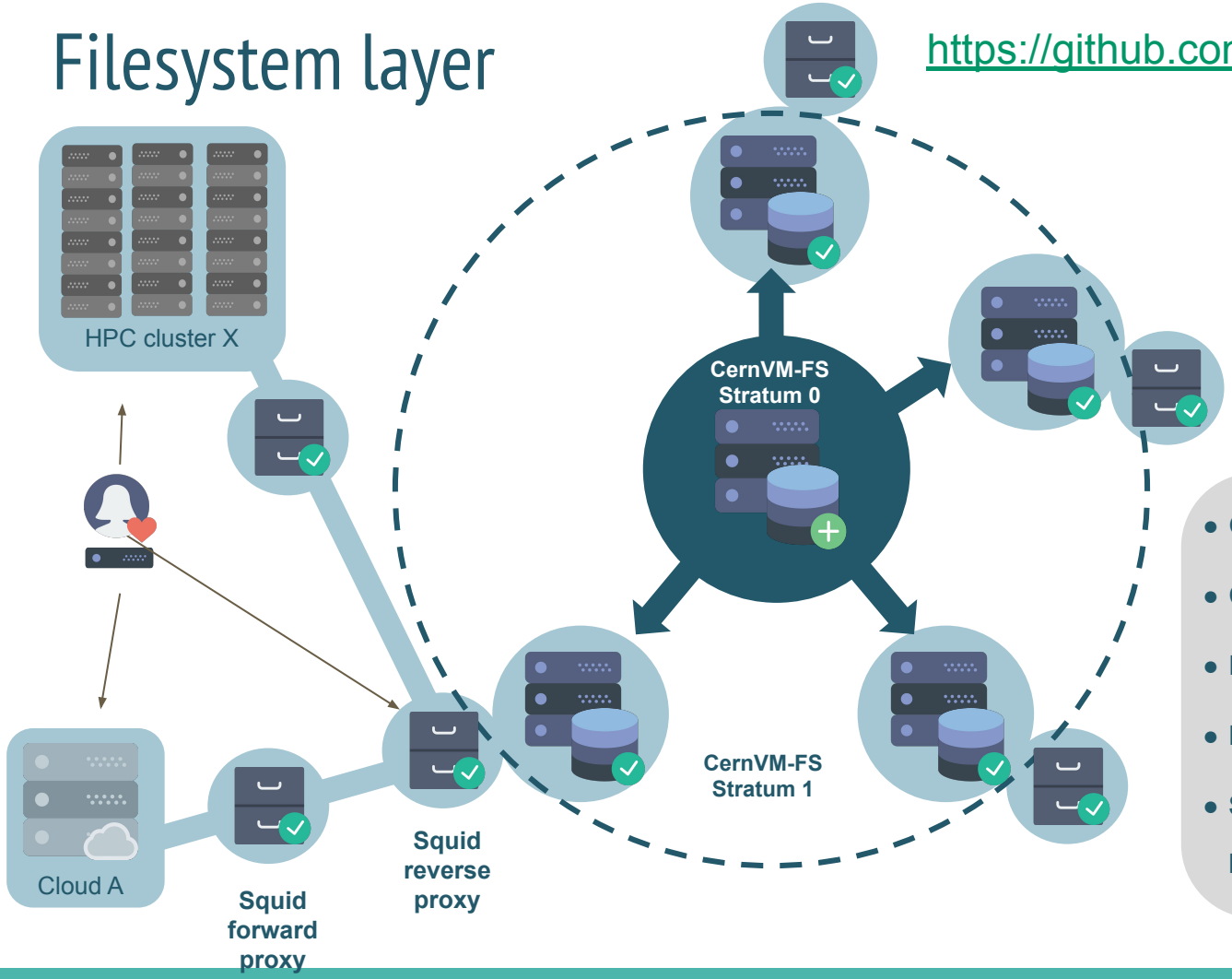
How exactly does it work?

High-level overview of EESSI project



Filesystem layer

https://github.com/EESSI/filesystem-layer



CernVM-FS

https://cvmfs.readthedocs.io

- Global distribution of software installations
- Centrally managed software stack
- Redundant network of “mirrors”
- Multiple levels of caching
- **Same software stack everywhere**:
  laptops, HPC clusters, cloud VMs, …

(icons via https://www.flaticon.com/authors/smashicons)

Compatibility layer

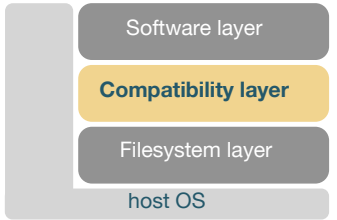
https://github.com/EESSI/compatibility-layer

- **Gentoo Prefix** installation (in `/cvmfs/.../compat/<os>/<arch>/`)
- Set of tools & libraries installed in non-standard location
- Limited to low-level stuff, incl. glibc (no Linux kernel or drivers)
  - Similar to the OS layer in container images
- Only targets a supported processor **family** (`aarch64`, `x86_64`, `riscv64`)
- **Levels the ground for different client operating systems** (Linux distros, later also macOS?)
- Currently in production repository:

  `/cvmfs/software.eessi.io/versions/2023.06/compat/linux/aarch64`

  `/cvmfs/software.eessi.io/versions/2023.06/compat/linux/x86_64`

powered by



Software layer

https://github.com/EESSI/software-layer

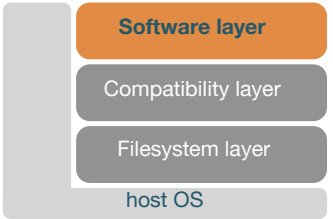
- Provides scientific software applications, libraries, and dependencies
- **Optimized for specific CPU microarchitectures** (Intel Haswell, ...)
  - Separate subdirectory/tree for each (in `/cvmfs/.../software/...`)
- **Leverages libraries** (like glibc) **from compatibility layer** (not from host OS)
- Installed with EasyBuild, incl. environment module files
- Lmod environment modules tool is used to access installations
- **Best subdirectory for host is selected automatically** via archspec (`archdetect` in production version)

EESSI

powered by

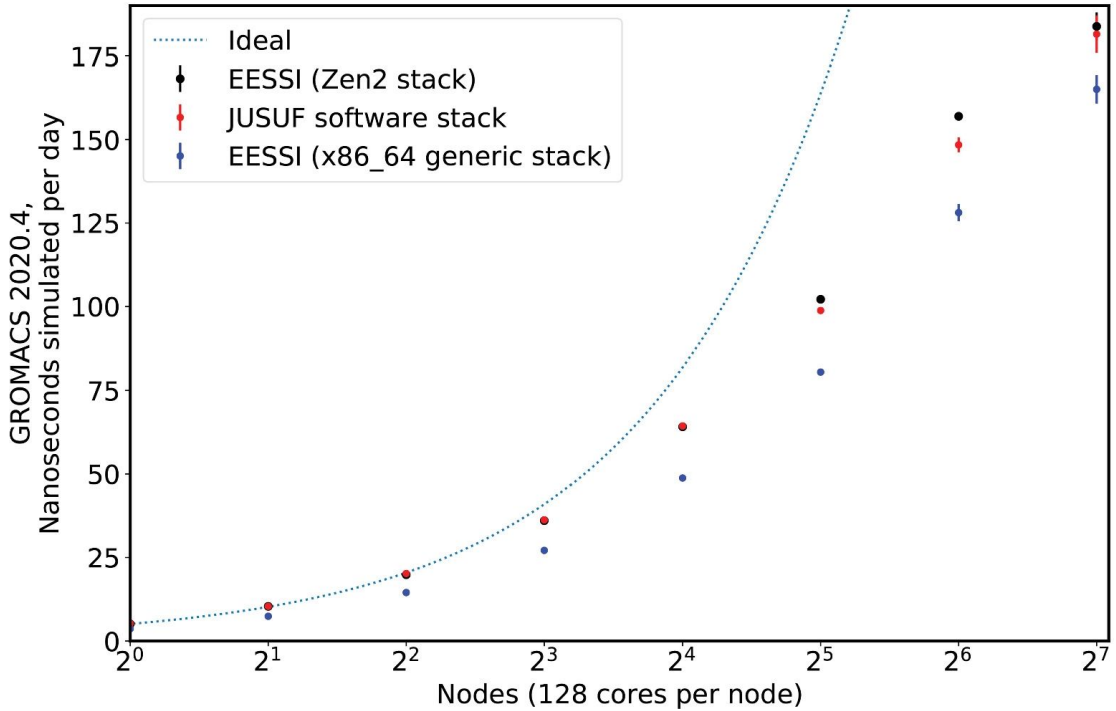
EasyBuild

Lmod



EESSI paper (open access)

doi.org/10.1002/spe.3075



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)

Current status of EESSI

- **Production repository `software.eessi.io` has been released!**
- Ansible playbooks, scripts, docs at `https://github.com/eessi`
- CernVM-FS: Stratum 0 @ Univ. of Groningen + two Stratum 1 servers
- Software (CPU-only): Bioconductor, GROMACS, OpenFOAM, R, TensorFlow, ...
- Hardware targets:
  - `{aarch64,x86_64}/generic`
  - `intel/{haswell,skylake_avx512}, amd/{zen2,zen3}, aarch64/{neoverse_n1,neoverse_v1)`
- NVIDIA GPU support verified and under code review
- Supported by Azure and AWS: sponsored credits to develop necessary infrastructure

Microsoft Azure

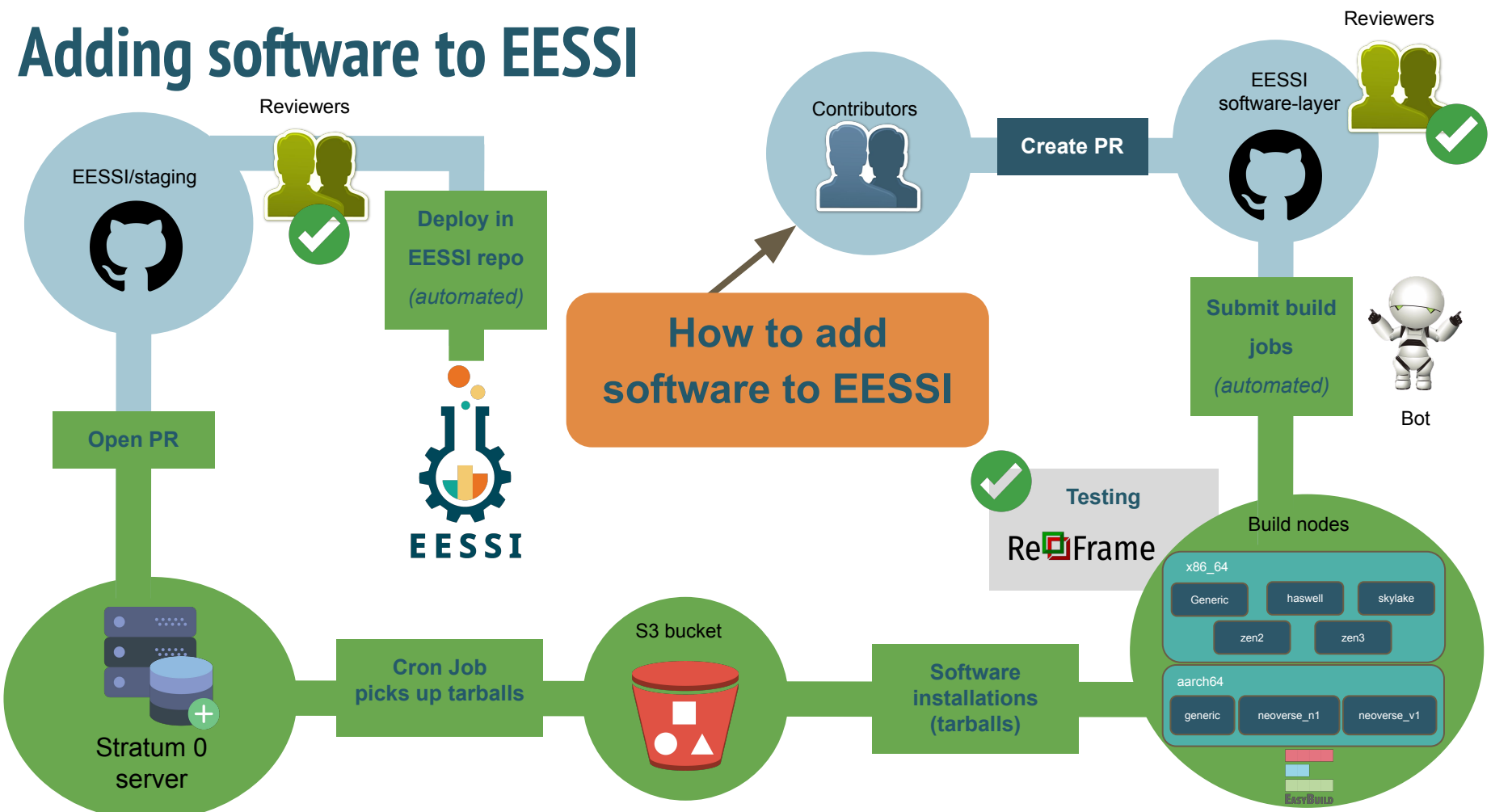
aws

EESSI

EUROPEAN ENVIRONMENT FOR
SCIENTIFIC SOFTWARE INSTALLATIONS

What about adding the software I care about?

Adding software to EESSI



eessi.io/docs/software_layer/adding_software

EESSI use cases

Overview of use cases enabled by EESSI

- A uniform software stack across HPC clusters, clouds, servers, and laptops
- Enable portable workflows
- Can be leveraged in continuous integration (CI) environments
- Significantly facilitates setting up infrastructure for HPC training
- Enhanced collaboration with software developers and application experts

Also discussed in our open-access paper, available via [doi.org/10.1002/spe.3075](doi.org/10.1002/spe.3075)

EESSI provides a uniform software stack

- Main goal: **same software everywhere**: laptop, server, HPC, cloud, …
- Wide variety of systems supported
  - CPUs: `x86_64` (Intel, AMD), `aarch64` (Arm), `riscv64` (soon…)
  - OS: any Linux distribution, Windows via WSL, macOS should be possible too
  - High-speed interconnects (Infiniband), GPUs, etc.
- **Without compromising on software performance**
  - Optimized software installations for specific CPU microarchitectures + auto-detection
  - Large contrast with generic binaries often used in containers
- Facilitates migrating from laptop to HPC, cloud bursting, …

EESSI enables portable workflows

- Portable workflows are significantly easier when relying on EESSI
- They often involve running a broad set of tools, which all need to be available
- Workflows definitions (Snakemake, Nextflow,…) can leverage (or be included in) EESSI
- You can ship your execution environment *inside* your git repository using `direnv`
    - If your users have EESSI and `direnv`, then can start running your workflow after cloning!

EESSI

EUROPEAN ENVIRONMENT FOR
SCIENTIFIC SOFTWARE INSTALLATIONS

Hands-on 2: Shipping a workflow

Hands-on scenario

- On provided infrastructure (AlmaLinux 8)
  - EESSI is available but `direnv` is not
- Clone a workflow that leverages EESSI
  - `git clone https://github.com/EESSI/eessi-nextflow-example`
- Workflow is trivial so can be run via shell
  - Enter repository: `cd eessi-nextflow-example`
  - Setup `direnv`: `source ./.install_direnv.sh`
  - Allow `direnv` in dir: `direnv allow`
  - Run the workflow: `./hello_plus_version.nf`

Leveraging EESSI in CI environments

- EESSI can be used in CI environments like Jenkins, GitHub Actions, …
- We can provide:
  - Different compilers to test your software with
  - Required dependencies for your software
  - Additional tools like ReFrame, performance analysis tools,…
- Other than CernVM-FS, no software installations required
  - Everything that is actually needed is pulled in on-demand by CernVM-FS
- Significantly facilitates also running CI tests in other contexts (laptop, HPC, …)

Leveraging EESSI in CI environment

We have an EESSI GitHub Action that provides EESSI+`direnv`:

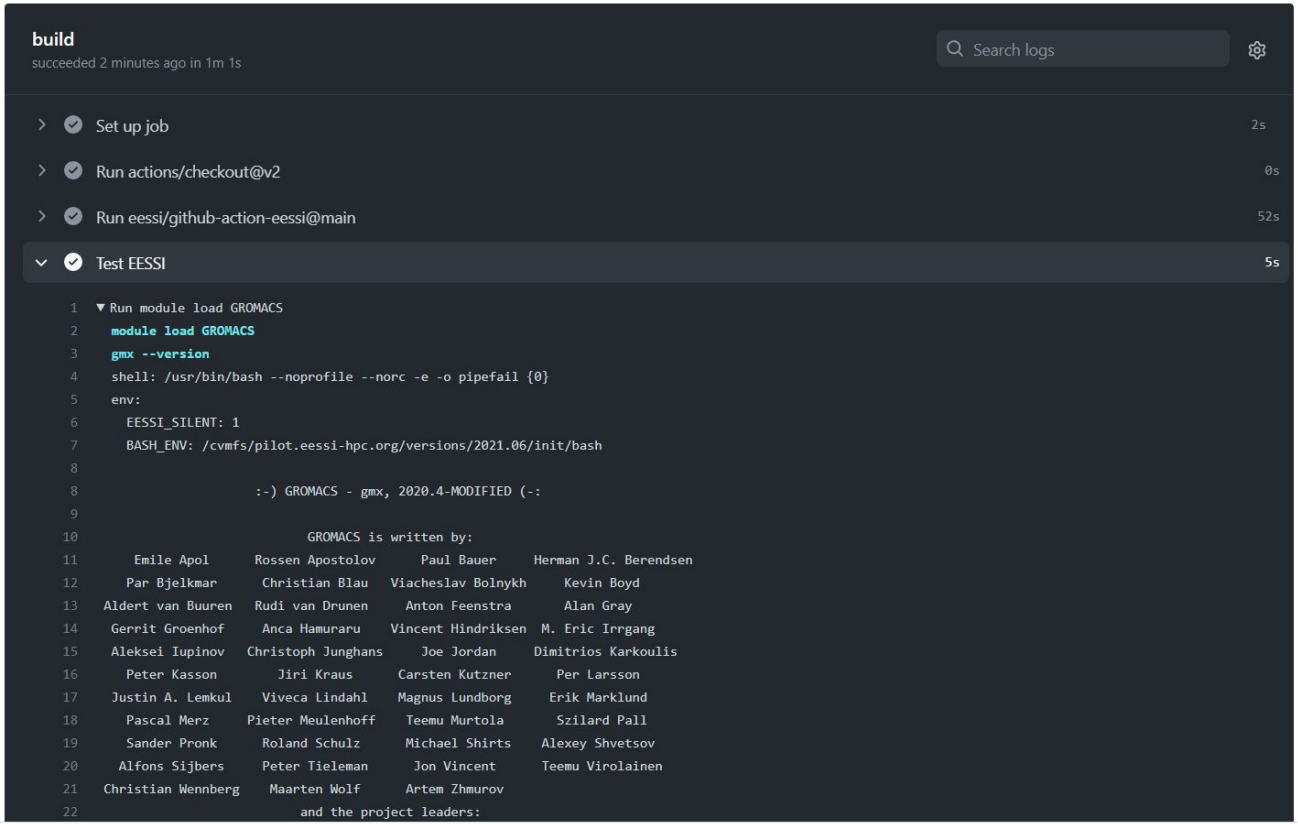
See it in action in the `github-essi-action` repository:

github.com/EESSI/github-action-eessi

github.com/EESSI/github-action-eessi/blob/main/.github/workflows/gromacs-usage.yml

```yaml
name: ubuntu_tensorflow
on: [push, pull_request]
jobs:
 build:
   runs-on: ubuntu-latest
   steps:
   - uses: actions/checkout@v3
   - uses: eessi/github-action-eessi@v3
     with:
       eessi_stack_version: '2023.06'
   - name: Test EESSI
     shell: bash
     run: |
       module load TensorFlow
       python -c 'import tensorflow; print(tensorflow.__version__)'
```

Leveraging EESSI GitHub Action



https://github.com/EESSI/github-action-eessi/actions/runs/3044539257/jobs/4905040409

Facilitate HPC training

- EESSI can significantly reduce effort required to set up infrastructure for HPC training sessions (introductory, software-specific, …)
- Setting up a throwaway Slurm cluster in the cloud is easy via Magic Castle
  - Used Magic Castle today:
    - Cluster was created by editing a single file
    - Automatically configured within 20 minutes
    - Includes support for GPU and fast interconnects (infiniband, EFA)
  - EESSI project uses Magic Castle for the build-and-deploy “bot”
  - There are also commercial alternatives that can/will support EESSI (Azure/AWS)
- EESSI can provide (scientific) software that is required for the training
- Attendees can easily set up the *same* software environment later on their own system(s) by leveraging EESSI

Collaboration with software developers + experts

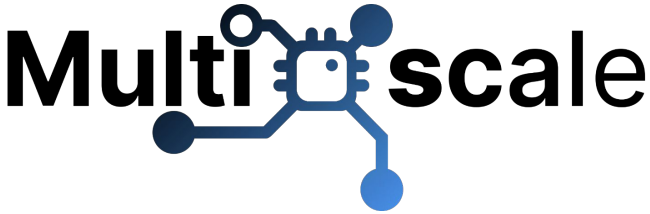
- A central software stack by/for the community opens new doors…
- We can work with software developers/experts to verify the installation
  - Check how installation is configured and built
  - Help to verify whether software is functional for different use cases
  - Show us how to do extensive testing of their software
  - Evaluate performance of the software, enable performance monitoring
  - **“Approved by developers” stamp for major applications included in EESSI**
- Relieve software developers from burden of getting their software installed
  - Remove need to provide pre-built binary packages?
- Developers can also leverage EESSI themselves: dependencies, CI, …

EESSI

EUROPEAN ENVIRONMENT FOR
SCIENTIFIC SOFTWARE INSTALLATIONS

Who pays for this?

The MultiXscale EuroHPC Project



- EuroHPC Centre of Excellence
  - 4 year project, started Q1 2023
- Budget of ~6M EUR (50% EU funding, 50% national funding)
  - Roughly 50% of funding for EESSI-related activities
- Collaboration between EESSI and CECAM (total of 16 partners)
  - EESSI primarily addresses technical aspects
  - CECAM network provides scientific expertise
- Scientific target are 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
- https://www.multixscale.eu

Website: eessi.io

GitHub: github.com/eessi

Documentation: eessi.io/docs

YouTube channel: youtube.com/@eessi_community

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

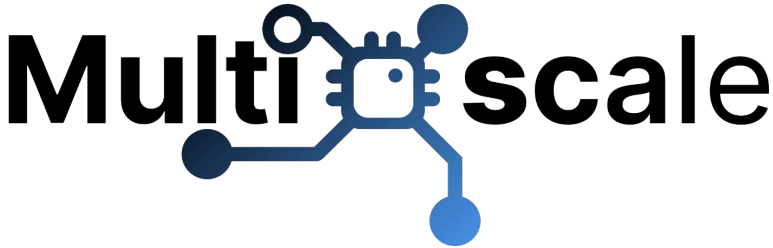
EESSI support portal: gitlab.com/eessi/support

Monthly online meetings (first Thursday, 2pm CEST)

eessi.io/docs/meetings/2022-09-amsterdam

Join our mailing list & Slack channel

Feedback request!



https://form.typeform.com/to/qaXPPdc3

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European High Performance Computing Joint Undertaking (JU) and countries participating in the project. Neither the European Union nor the granting authority can be held responsible for them.