



university of
groningen

center for
information technology

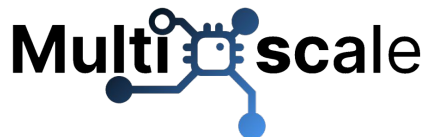


E E S S I

EUROPEAN ENVIRONMENT FOR
SCIENTIFIC SOFTWARE INSTALLATIONS

Keeping it Simple, Keeping it

Updates from 2023-2024



Pedro Santos Neves
SURF Advanced Computing User Day
Utrecht 12/12/2024



EuroHPC
Joint Undertaking

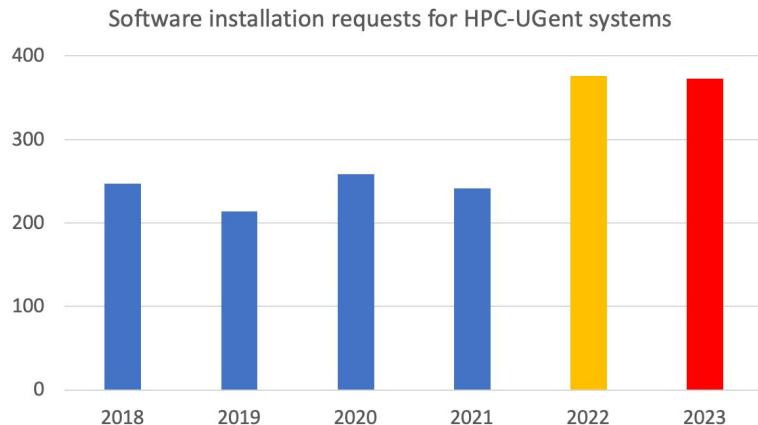
Contents



- Brief introduction to EESSI and MultiXscale
- GPU and RISC-V builds
- Building on top of EESSI - EESSI-extend
- Development repository - dev.eessi.io
- EESSI CI/CD
- Assorted news

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](#), [DEUCALION](#)...), RISC-V is coming (soon?)
- In strong contrast: available (wo)manpower in **HPC support teams is (still) limited...**



Major goals of EESSI



- **Avoid duplicate work** (for researchers, HPC support teams, sysadmins, ...)
 - Tools that automate software installation process (EasyBuild, Spack) are not sufficient anymore
 - 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, ...

EESSI in a nutshell



- European Environment for Scientific Software Installations (EESSI)
- **Shared repository of (optimized!) scientific software installations**
- Uniform way of providing software to users, regardless of the system they use!
- Should work on any Linux OS (+ WSL, macOS via Lima) and system architecture
- From laptops and personal workstations to HPC clusters and cloud
- Support for different CPU (micro)architectures, interconnects, GPUs, etc.
- **Focus on performance, automation, testing, collaboration**

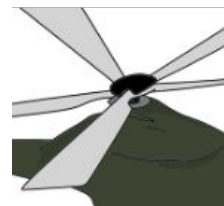
<https://eessi.io>

<https://eessi.io/docs>

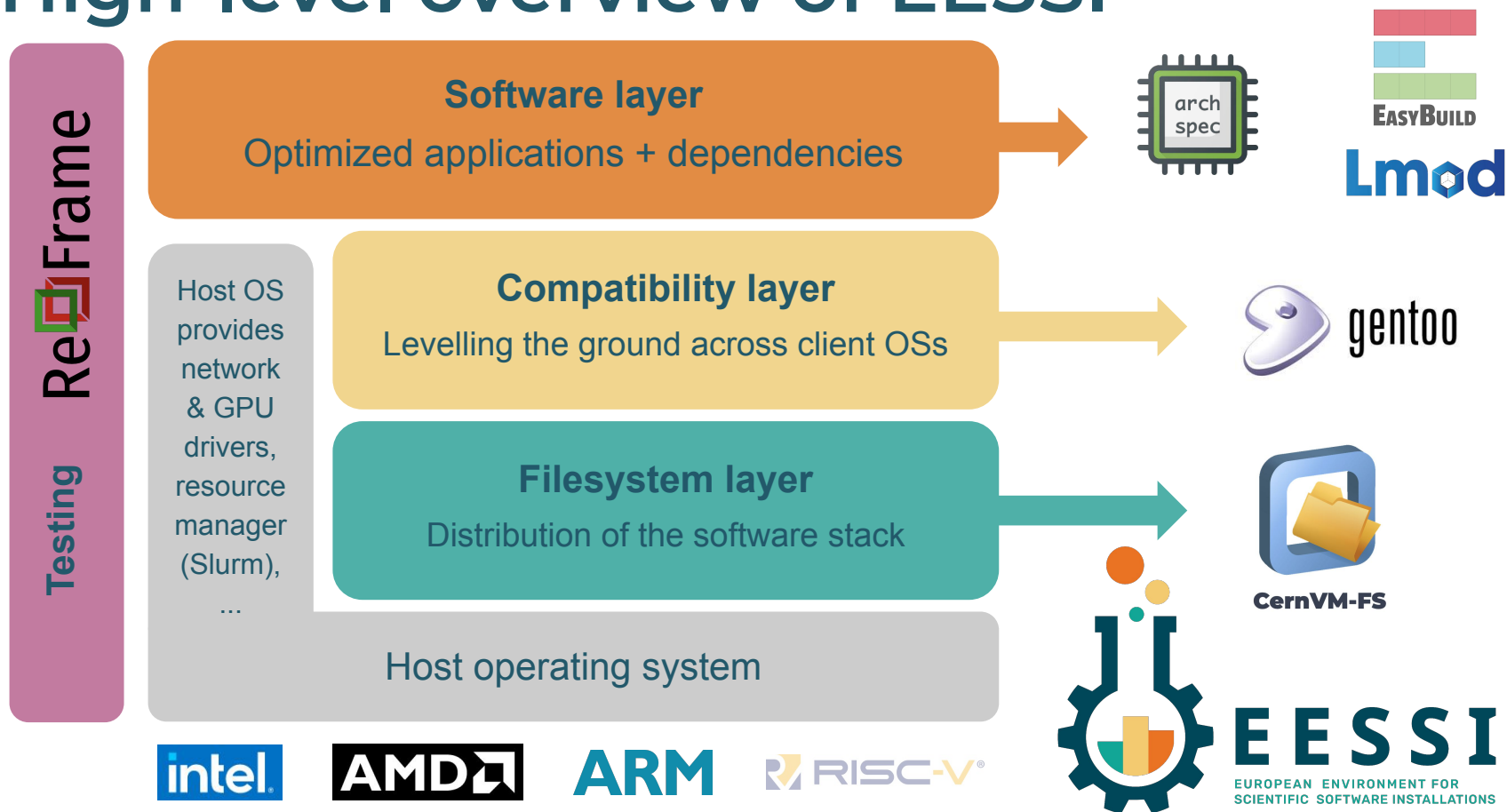
MultiXscale Centre-of-Excellence in a nutshell



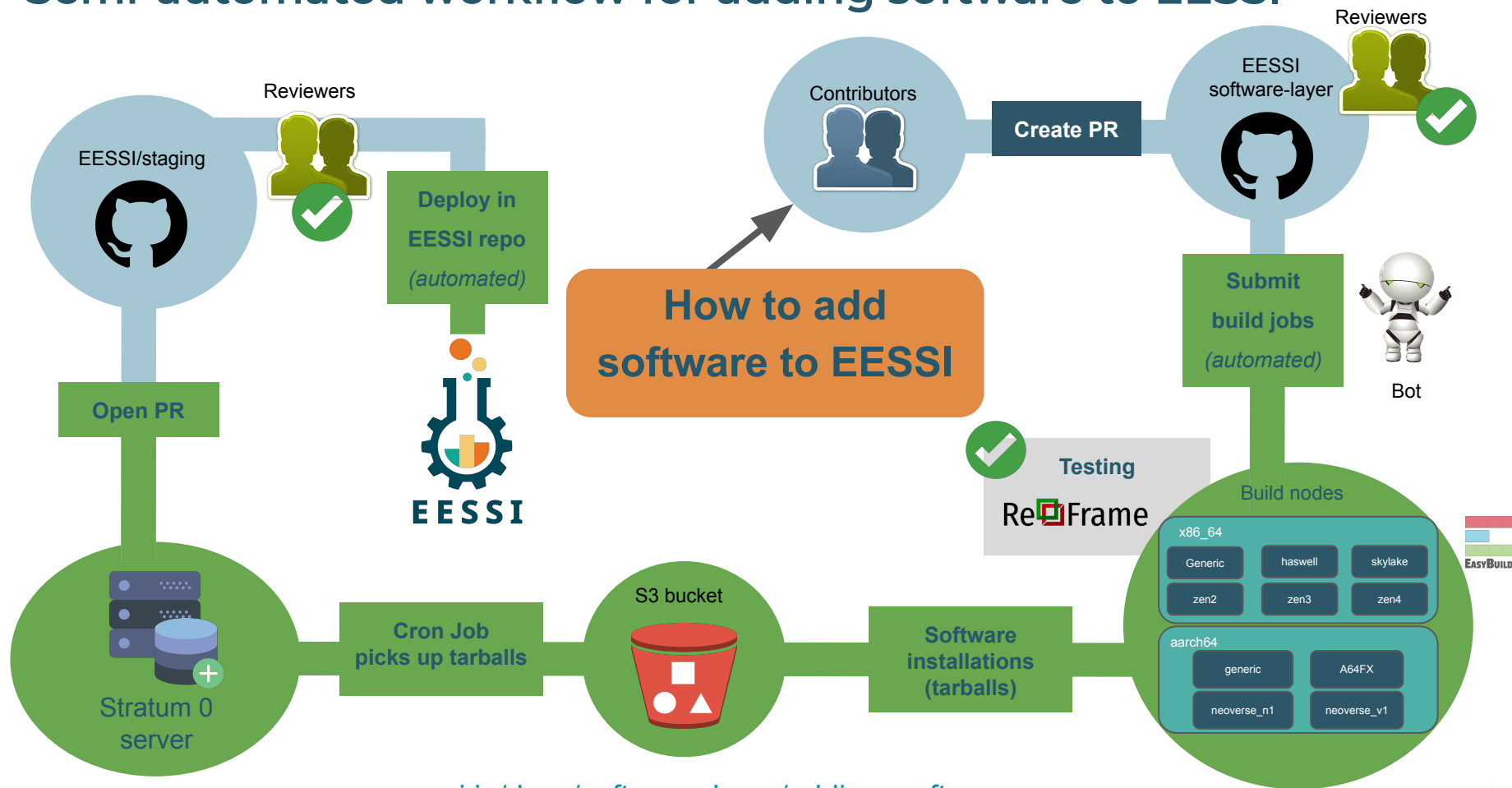
- 4-year project (started in Jan 2023), ~€6M budget
- Collaboration between EESSI and CECAM (total of 16 partners)
 - **EESSI** primarily addresses technical aspects
 - **CECAM** network provides scientific expertise
- Scientific target: multiscale simulations with 3 key use cases
 - Helicopter design and certification for civil transport
 - Battery applications to support the sustainable energy transition
 - Ultrasound for non-invasive diagnostics and biomedical applications
- More info: <https://multixscale.eu>



High-level overview of EESSI



Semi-automated workflow for adding software to EESSI

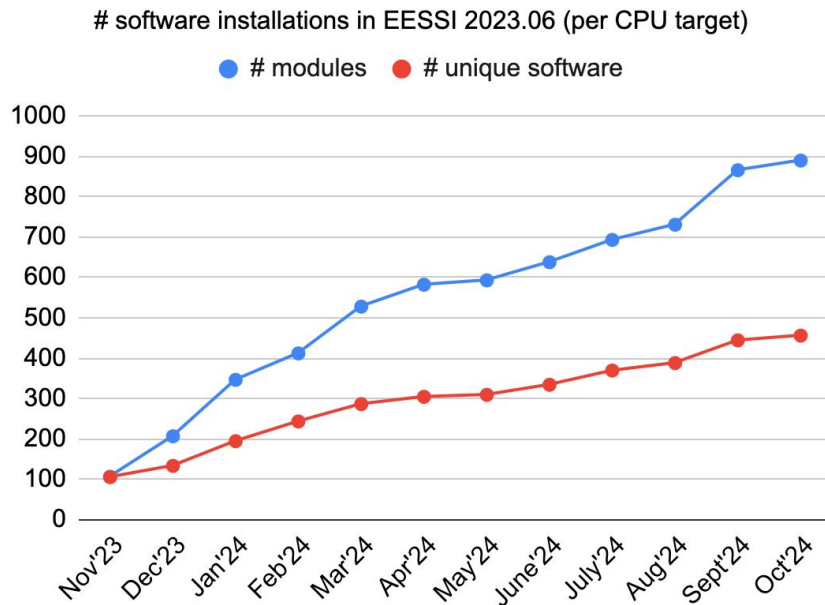


Overview of available software



Currently ~900 software installations available
per CPU target via software.eessi.io CernVM-FS
repository; increasing every day

- Over 450 different software packages
- Excl. extensions: Python packages, R libraries
- Including ESPResSo, GROMACS, LAMMPS, OpenFOAM, PyTorch, R, QuantumESPRESSO, TensorFlow, waLBerla, WRF, ...
- eessi.io/docs/available_software/overview
- Using recent compiler toolchains: currently focusing on `foss/2023a` and `foss/2023b`





Highlights from 2023/2024

GPU and RISC-V support



- Much work in implementing GPU and RISC-V support
- riscv.eessi.io repository available with generic builds
 - <https://www.eessi.io/docs/repositories/riscv.eessi.io/>
- First GPU builds of scientific software on AMD Rome (zen2) with NVIDIA CC 8.0 (A100) - Systems are quite heterogeneous



Building on top of EESSI - EESSI-extend

- **EESSI-extend module**
 - Simple way expand user or site's software stack
 - **module load EESSI-extend/2023.06-easybuild**
 - Automatically build against compatibility layer
 - Other EESSI modules picked up as dependencies
- **Can install in:**
 - User space - `$EESSI_USER_INSTALL`
 - Shared project directory - `$EESSI_PROJECT_INSTALL`
 - Restricted directories by sysadmins- `$EESSI_SITE_INSTALL`

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

Development repository - dev.eessi.io

- Let developers deploy and test their pre-production code
- Similar CernVM-FS repository:
 - `/cvmfs/dev.eessi.io/version/project`
 - Similar infrastructure:
 - GitHub repository to track builds
 - CI on PRs
 - Bot instance that builds, runs tests, ingests software (semi-)automatically

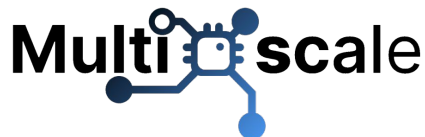
dev.eessi.io - Key differences

software.eessi.io

- Permanent installations
- Only install releases
- Builds for all CPU architectures
- `/cvmfs/software.eessi.io/version/`

dev.eessi.io

- Temporary installations
- Can install from commits
- Can skip some CPU targets
- dev.eessi.io built on top of software.eessi.io
- Subdirectories per project
- `/cvmfs/dev.eessi.io/version/project`



Leveraging EESSI in CI environments



- EESSI can be used in CI environments like:
 - GitHub: github.com/marketplace/actions/eessi
 - GitLab: gitlab.com/explore/catalog/eessi/gitlab-eessi
- EESSI 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 to get access to EESSI, 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 environments



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

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

github.com/EESSI/github-action-eessi

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

```
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



```
build
succeeded 2 minutes ago in 1m 1s
Search logs

> Set up job 2s
> Run actions/checkout@v2 0s
> Run eessi/github-action-eessi@main 52s
▼ Test EESSI 5s
1 Run module load GROMACS
2 module load GROMACS
3 gmx --version
4 shell: /usr/bin/bash --noprofile --norc -e -o pipefail {0}
5 env:
6   EESSI_SILENT: 1
7   BASH_ENV: /cvmfs/pilot.eessi-hpc.org/versions/2021.06/init/bash
8
9   :~) GROMACS - gmx, 2020.4-MODIFIED (~:
10
11      GROMACS is written by:
12      Emile Apol      Rossen Apostolov      Paul Bauer      Herman J.C. Berendsen
13      Par Bjelkmar      Christian Blau      Viacheslav Bolnykh      Kevin Boyd
14      Aldert van Buuren      Rudi van Drunen      Anton Feenstra      Alan Gray
15      Gerrit Groenhof      Anca Hamuraru      Vincent Hindriksen      M. Eric Irrgang
16      Aleksei Iupinov      Christoph Junghans      Joe Jordan      Dimitrios Karkoulis
17      Peter Kasson      Jiri Kraus      Carsten Kutzner      Per Larsson
18      Justin A. Lemkul      Viveca Lindahl      Magnus Lundborg      Erik Marklund
19      Pascal Merz      Pieter Meulenhoff      Teemu Murtola      Szilard Pall
20      Sander Pronk      Roland Schulz      Michael Shirts      Alexey Shvetsov
21      Alfons Sijbers      Peter Tieleman      Jon Vincent      Teemu Virolainen
22      Christian Wennberg      Maarten Wolf      Artem Zhmurov
23      and the project leaders:
```

<https://github.com/EESSI/github-action-eessi/actions/runs/11183032689/job/31090668500>

HPCwire Readers Choice Award

- EESSI wins HPCwire Readers' Choice Award in the [Best HPC Programming Tool or Technology category](#)
- Award presented at SC 24 conference in Atlanta



Thank you for listening! Questions?

News on our blog

<https://www.eessi.io/docs/blog/>



Funded by the European Union, the European High Performance Computing Joint Undertaking (JU) and countries participating in the project under grant agreement No 101093169.



EuroHPC
Joint Undertaking 19