Getting started with scientific clusters

Samuel Fux, Jarunan Panyasantisuk
High Performance Computing Group
Scientific IT Services, ETH Zurich

Outlook

- Introduction

- Accessing the cluster

- Storage and data transfer

- Modules and applications

- Getting help

Outlook

- Introduction

- Accessing the cluster

- Storage and data transfer

- Modules and applications

- Getting help

Intro > What is EULER ?

- EULER stands for **E**rweiterbarer, **U**mweltfreundlicher, **L**eistungsfähiger **E**TH **R**echner (Scalable, environment-friendly, powerful ETH Computer)

- Benefits from 20 years of experience with HPC clusters at ETH
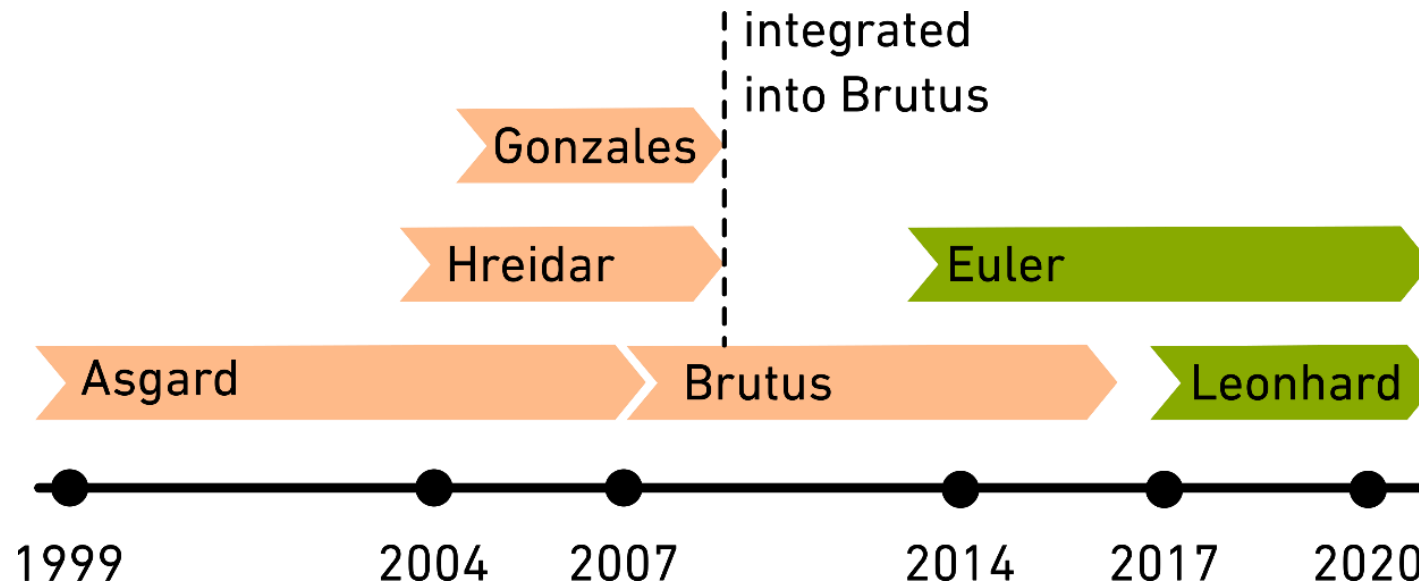
Intro > Shareholder model

Shareholders

- Like its predecessors, Euler has been financed (for the most part) by its users

- So far, over 100 (!) research groups from almost all departments of ETH have invested in Euler

- These so-called **shareholders** receive a share of the cluster's resources (processors, memory, storage) proportional to their investment

Public share

- A small share of Euler financed by IT Services is open to all members of ETH

- The only requirement is a valid ETH account

- These **guest users** can use limited resources

- If someone needs more computing power, they can invest in the cluster and become a shareholder at any time
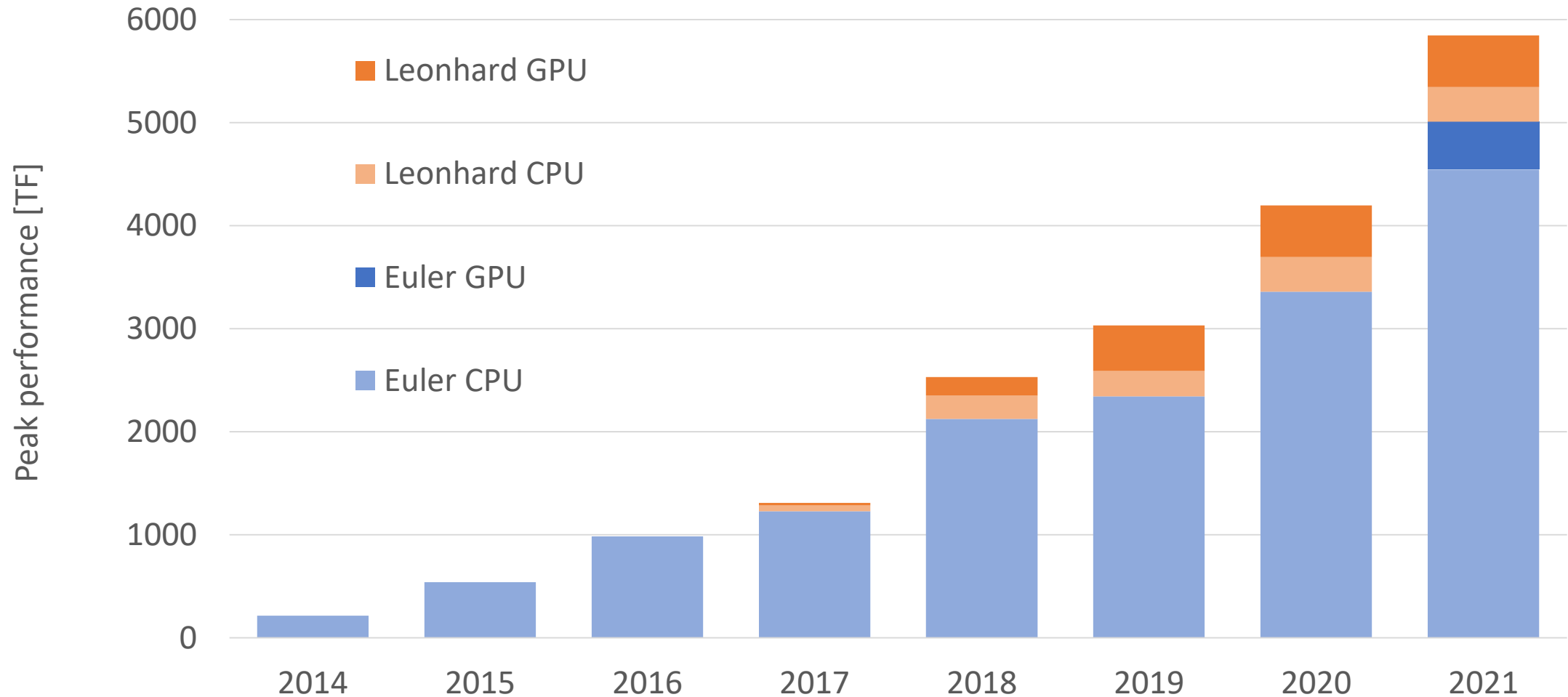
Public share
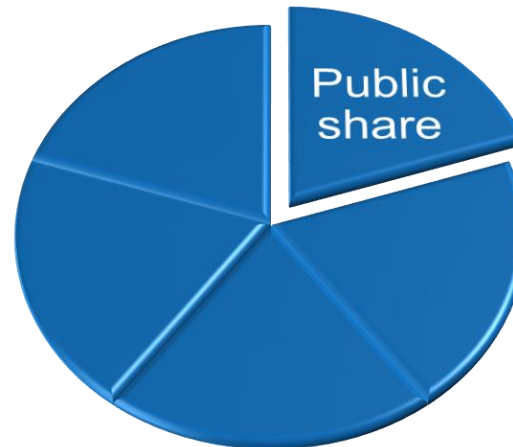
Intro > Euler shares by departments (February 2020)
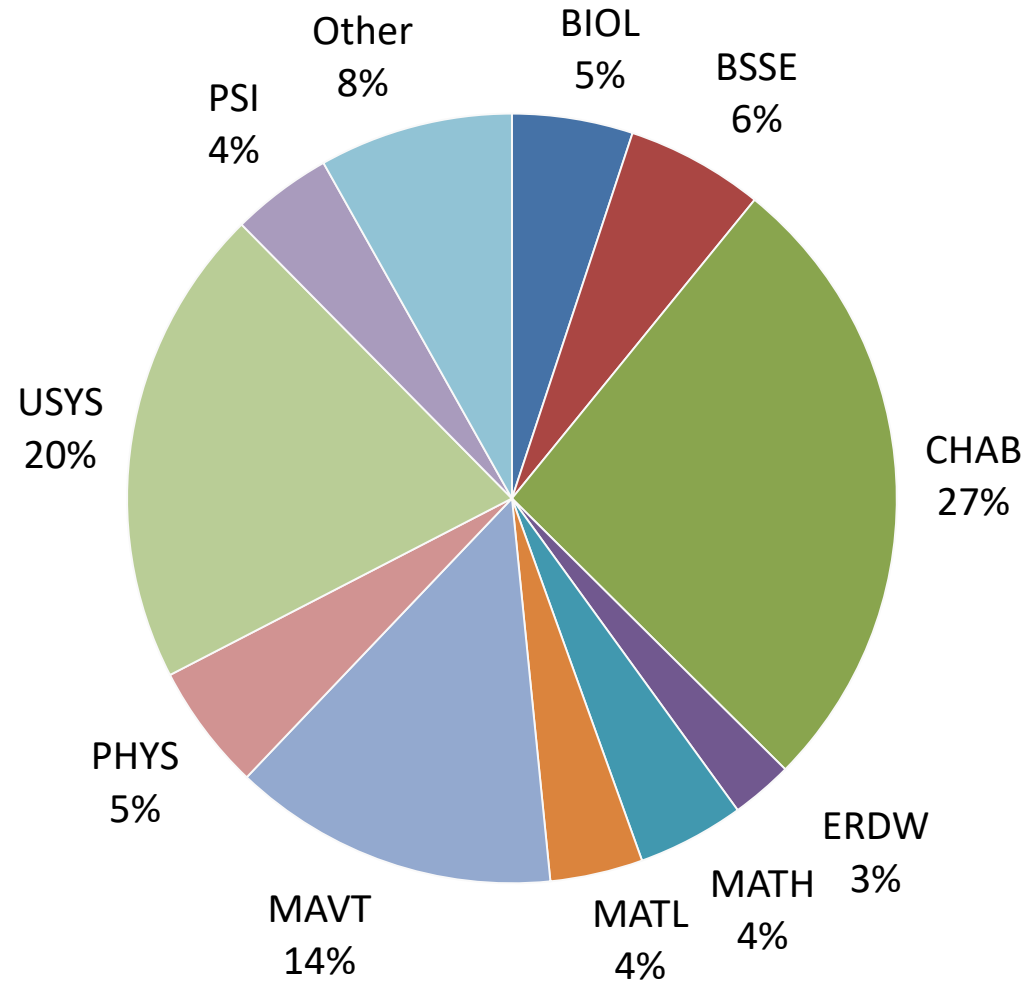
Intro > Cumulated performance of Euler and Leonhard clusters

All values correspond to the state in March of each year
Peak performance in double precision based on CPU/GPU nominal frequency; effective performance is lower

Intro > Euler II (left) & Euler V (right)

Preparation > Linux terminal > Bash

- Bash is a powerful scripting language, which can directly be used in a Linux terminal and is often combined with Linux commands

```
[sfux@eu-login-01 ~]$ for i in *; do if [ -d $i ]; then echo $i; cd $i; du -hs; cd ..; fi; done
bin
52K      .
lib64
36K      .
scripts
192K     .
test
324M     .
testrun
693M     .
[sfux@eu-login-01 ~]$
```

https://tldp.org/LDP/Bash-Beginners-Guide/html/

Preparation > Linux terminal > Basic commands

Command	Explanation
`ls`	Directory listing
`cd`	Change directory
`pwd`	Print working directory
`echo`	Print to standard output
`less, cat, more`	Display content of file to standard output
`cp, mv`	Copy/move a file or directory
`rm`	Remove a file or directory
`mkdir`	Create a directory
`vi, nano, emacs`	Command line text editor
`man`	Show manual for a command in terminal
`grep`	Search for a pattern in a string or file

Preparation > Linux terminal > Basic commands

Command	Explanation
`exit`	Terminate and exit the current terminal
`sort`	Sort a file line by line
`sed`	String manipulation
`awk`	Powerful programming language
`find`	Search for files
`du`	Show disk space used in a directory
`tar`	Create a tar archive with files/directories
`gzip`	Compress files/directories
`top`	Real time view of processes in a computer
`chmod`	Change permissions of file/directory

https://scicomp.ethz.ch/wiki/Linux_command_line

Preparation > Linux permissions

- In Linux, access to files and directories is handled via permissions
  - Read permission (r) - grants permission to read a file or directory
  - Write permission (w) - grants permission to write a file or directory
  - Execute permission (x) - grants permission to execute a file or directory

- There are 3 permission groups
  - User (u) - permissions for the user account that owns the file
  - Group (g) - permissions for the user group that owns the file
  - Other (o) - permissions for all other users except the user account and the user group

```
[sfux@eu-login-29 ~]$ ls -l gurobi.log
-rw-r--r-- 1 sfux sfux-group 800 Sep 17 10:29 gurobi.log
[sfux@eu-login-29 ~]$ ls -ld data
drwxr-xr-x 2 sfux sfux-group 4096 Jan  9  2017 data
```

https://scicomp.ethz.ch/wiki/Linux_permissions

Preparation > Linux permissions

- Another method for representing Unix permissions is an octal (base 8) notation
  - The read bit adds 4 to its total (in binary 100),
  - The write bit adds 2 to its total (in binary 010), and
  - The execute bit adds 1 to its total (in binary 001).

  - Example: `r-x` $\longrightarrow$ 101 $\longrightarrow$ 4+0+1=5

  - (u), (g) and (o) are then combined (755 represents `rwxr-xr-x`)
- Permissions can be changed with the `chmod` command

  - String representation:

    `chmod ugo+rx filename`

  - Number representation:

    `chmod 750 filename`

https://en.wikipedia.org/wiki/File-system_permissions#Numeric_notation

Preparation > Paths > Linux/Mac OS X vs. Windows

- There are differences how operating systems are representing file paths

- Paths on Windows: `C:\Users\Samfux\test`
  - Use backslashes
  - File and directory names are not case sensitive
  - Different drives (`C:`, `D:`, etc.)

- Paths on Linux/Mac OS X: `/cluster/home/sfux/test`
  - Use forward slashes
  - Everything is case sensitive
  - Everything is under the root file system (`/`), no drives

ETHzürich   Scientific IT Services

Preparation > Edit a text file with vim

- Vim is a command line text editor that can be started with the command `vi`. This text editor is useful to edit input files and write scripts directly on the cluster without copying forth and back files

- 2 modes (insert mode, command mode)
  - Type `i` to switch from command mode to insert mode
  - Type `esc` to switch from insert mode to command mode

- Insert mode: You can insert text into a text document

- Command mode: You can type commands after typing `:` (colon) and execute it with the enter key
  - To save a file, type `:w`
  - To close a file, type `:q`
  - To save and close a file, type `:wq`

https://www.tutorialspoint.com/unix/unix-vi-editor.htm

ETH_zürich_   Scientific IT Services

Preparation > Exercise (local shell)

Start a shell/terminal on your local computer (in MobaXterm, click on the button **Local shell**) to carry out the following exercises:

1. Print the current working directory
2. Create a new directory called **test**
3. Change to the new directory
4. Create a text file **my_first_bash_script.sh** with vim with the following content
   ```
   #!/bin/bash
   echo "Good morning $USER"
   hostname
   pwd
   ```
5. Change the permission of the file to **755**
6. Do a directory listing of the **test** directory
7. Execute the file that you have created with the command `./my_first_bash_script.sh`
8. Use the command `grep` to search for the word **morning** in `my_first_bash_script.sh` file
9. Sort the lines in the text file alphabetically
10. Exit the shell

Outlook

- Introduction

- **Accessing the cluster**

- Storage and data transfer

- Modules and applications

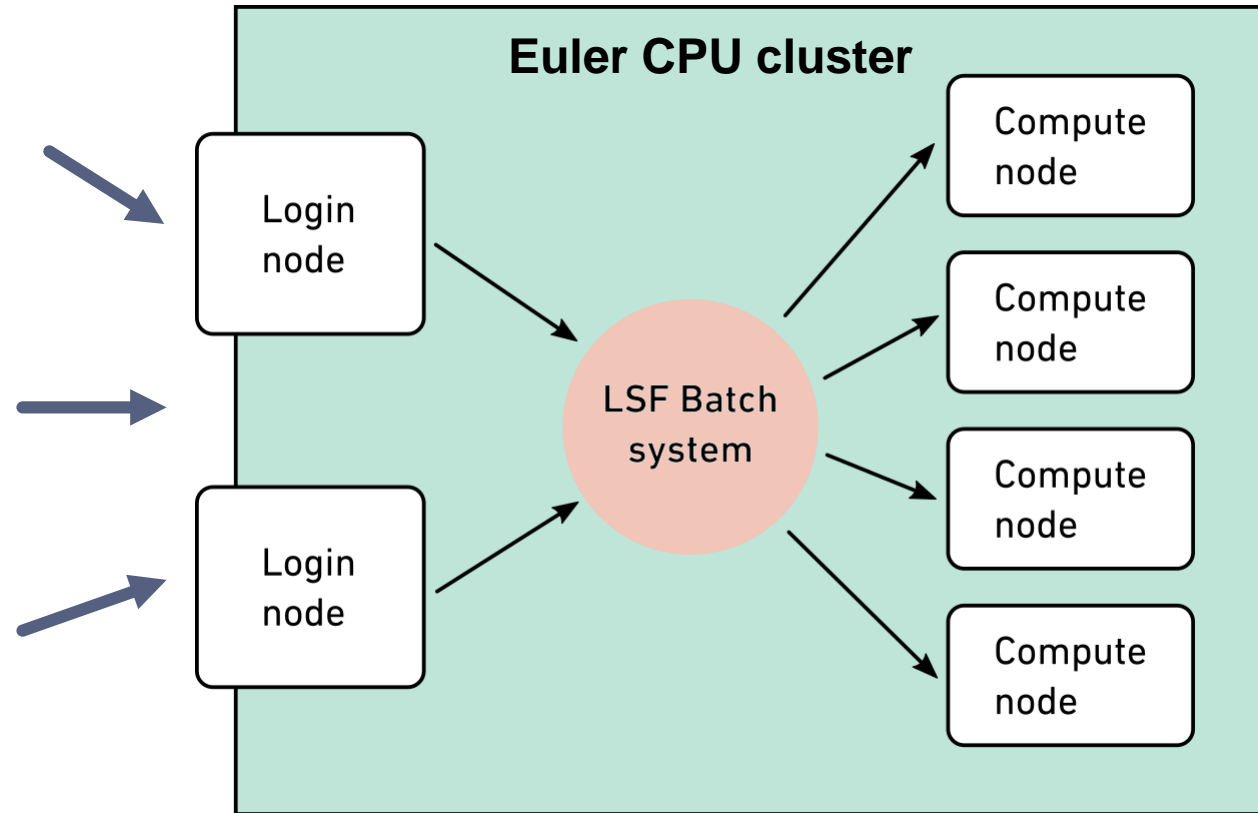- Getting help

Accessing the clusters
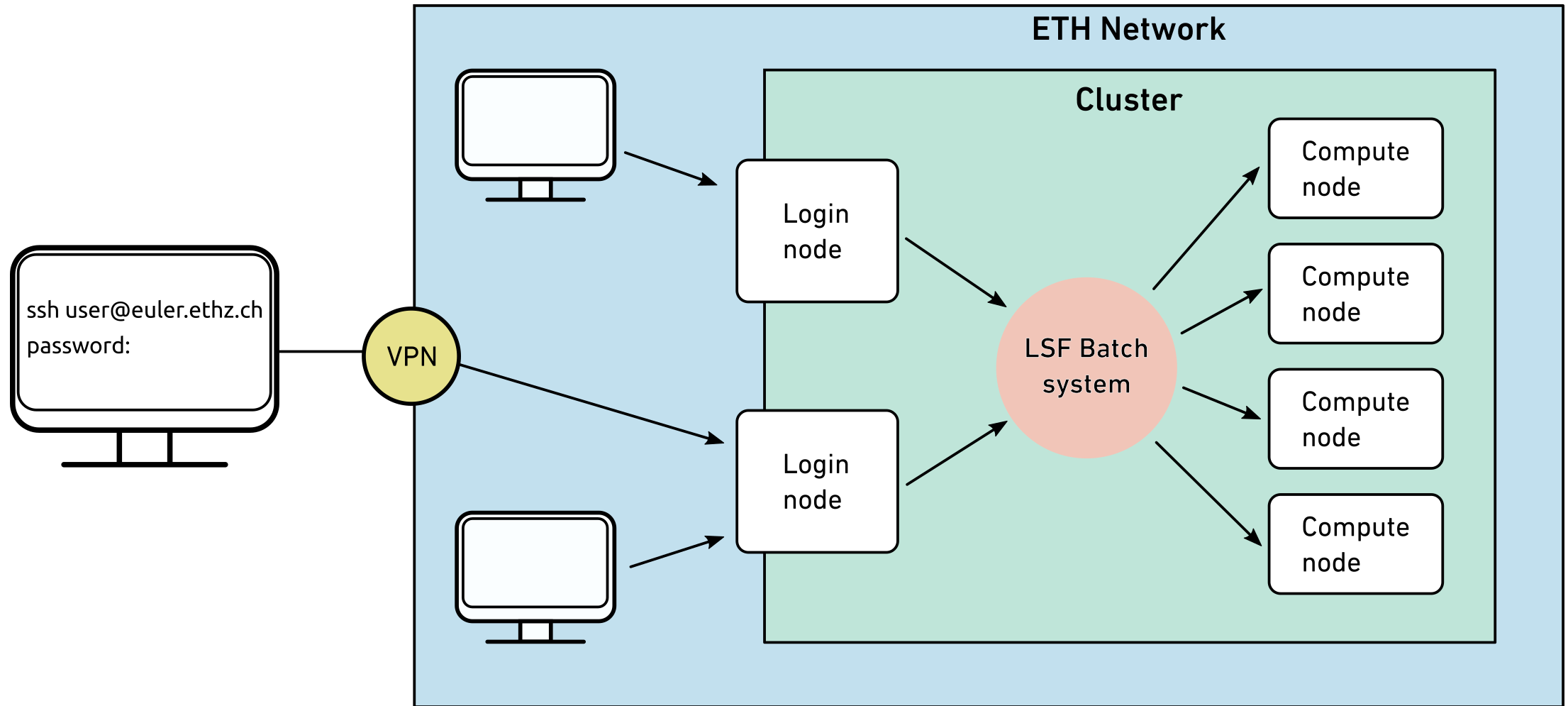
Access > Who can use the cluster > Euler CPU cluster

Shareholders that invested into a share of the clusters resources

External collaboration partners of shareholders

All ETH members (they can use the cluster as guest users with limited resources)
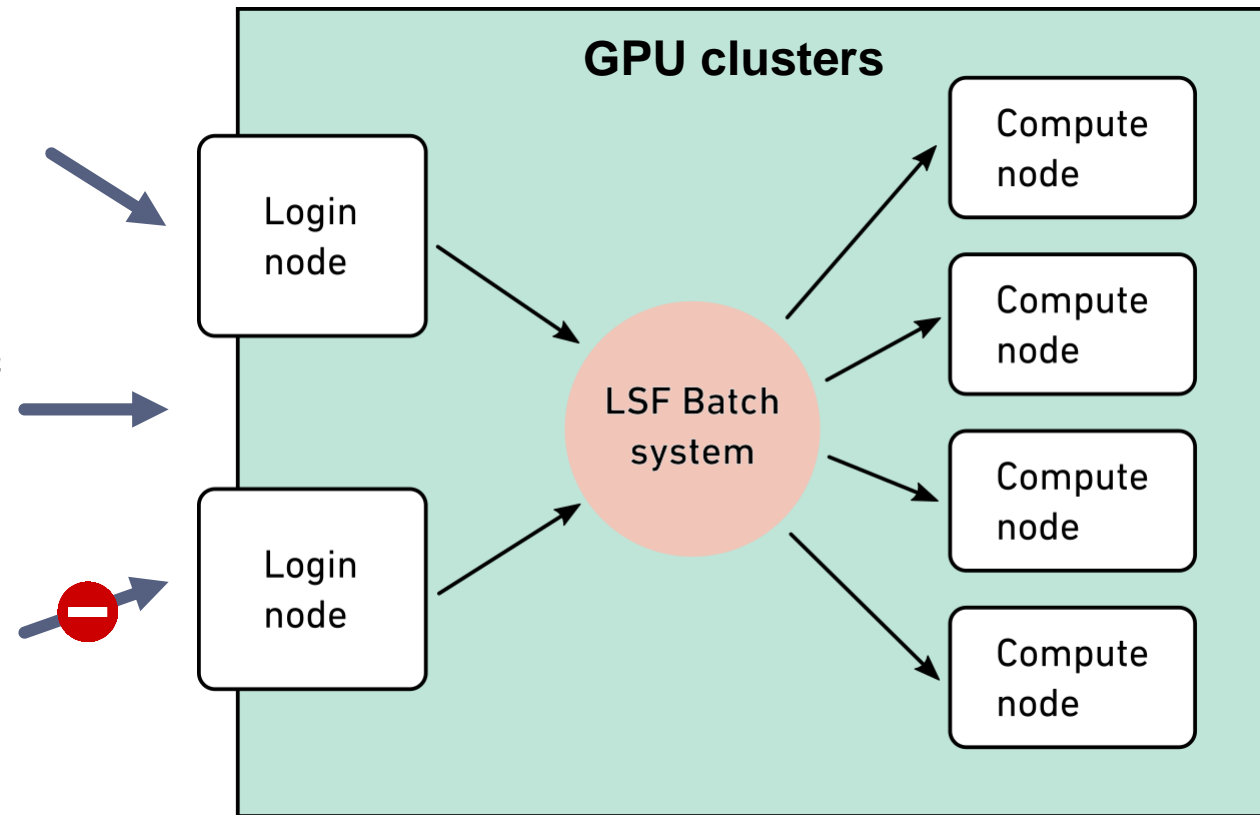
Access > Who can use the cluster > Euler GPU cluster & Leonhard Open

Shareholders that invested into a
share of the clusters resources

External collaboration partners of
shareholders

Guest users cannot access
Leonhard Open

GPU clusters

Login node

Login node

LSF Batch system

Compute node

Compute node

Compute node

Compute node

Scientific IT Services

ETH *zürich*

Access > Prerequisites

- A valid ETH account

- Local computer with an SSH client
  - Linux and macOS contain SSH client as part of the operating system
  - Windows users need to install a third party SSH client
    - MobaXterm (https://mobaxterm.mobatek.net/) is a free open source SSH client that we recommend

- An X11 server for graphical user interface (optional)
  - Linux (https://www.xorg.com)
  - macOS (https://xquartz.org)
  - Windows (included in MobaXterm)

Access > How to access the clusters > ETH members

1. Start your SSH client

2. Use `ssh` command to connect to the login node of Euler or Leonhard Open

   ```
   ssh username@euler.ethz.ch
   ssh username@login.leonhard.ethz.ch
   ```

3. Use your ETH credentials to login

4. First login
   - On first login a verification code is sent to your email address (*username@ethz.ch*)
   - By entering the verification code, your account is created automatically
   - New users must accept the cluster's usage rules

https://scicomp.ethz.ch/wiki/New_account_request_process_for_HPC_clusters

# Access > How to access the clusters > External collaborators

Members of other institutions who collaborate with a research group at ETH may use the clusters for the purpose of said collaboration. To get access:

1. Their ETH counterpart ("sponsor") create an ETH guest account, including e-mail address and VPN service, for them

2. Access the cluster like ETH members

Access > Security > Firewall

From outside, establish
first a VPN connection

Then, connect to the
cluster through SSH
secure protocol
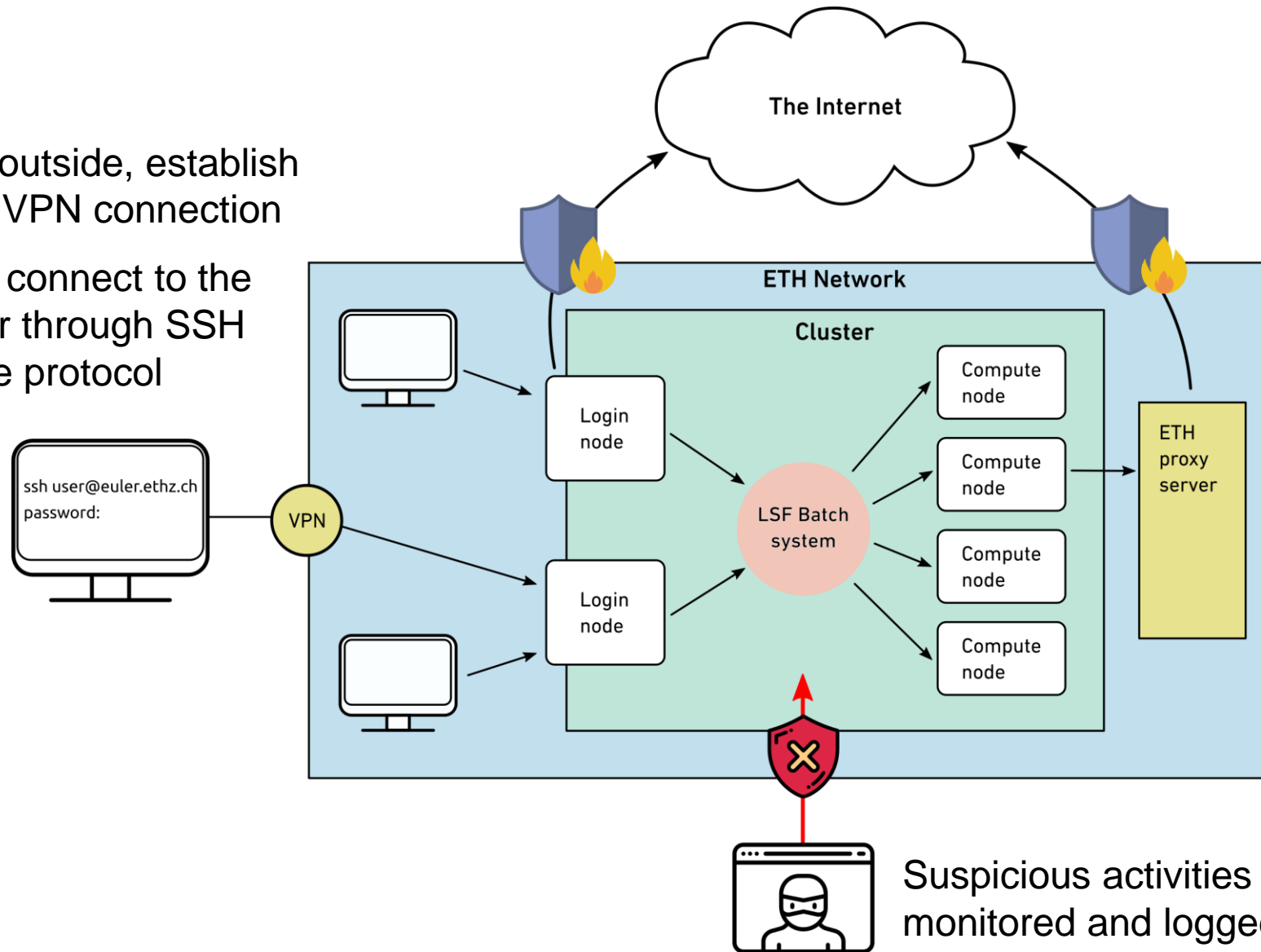
From a compute node to
an external service, use
the ETH proxy

`$ module load eth_proxy`

Suspicious activities are
monitored and logged

Access > Legal compliance

- The HPC clusters are subject to ETH's acceptable use policy for IT resources (Benutzungsordnung für Telematik, BOT, https://rechtssammlung.sp.ethz.ch/Dokumente/203.21en.pdf), in particular:
  - Cluster accounts are **strictly personal**
  - DO NOT share your account (password, ssh keys) with anyone
  - DO NOT use someone else's account, even if they say it's OK
  - If you suspect that someone used your account:
    - change your password at https://password.ethz.ch
    - contact cluster-support@id.ethz.ch
- Consequences
  - In case of abuse, the offender's account may be blocked temporarily or closed
  - System administrators are obliged by law to investigate abusive or illegal activities and report them to the relevant authorities

Access > SSH connection

1. Start your SSH client

2. Use `ssh` command to connect to the login node of Euler or Leonhard Open

   ```
   ssh username@euler.ethz.ch
   ssh username@login.leonhard.ethz.ch
   ```

3. Use your ETH credentials to login

https://scicomp.ethz.ch/wiki/Accessing_the_clusters#SSH

ETH *zürich*   Scientific IT Services

Access > SSH connection > Linux, macOS

```
samfux@bullvalene:~$ ssh sfux@euler.ethz.ch
sfux@euler.ethz.ch's password:
Last login: Fri Sep 13 07:33:57 2019 from bullvalene.ethz.ch

        _____  ____
       / _____    ___    /__/   /
      / ____/  /   /  /  ___     /
     /_____/  /__/  /__/  /__/
    Eidgenoessische Technische Hochschule Zuerich
    Swiss Federal Institute of Technology Zurich
    -------------------------------------------------------------
                            E U L E R   C L U S T E R


                                        https://scicomp.ethz.ch
                                  http://www.smartdesk.ethz.ch
                                    cluster-support@id.ethz.ch


       ===============================================================

[sfux@eu-login-19 ~]$
```
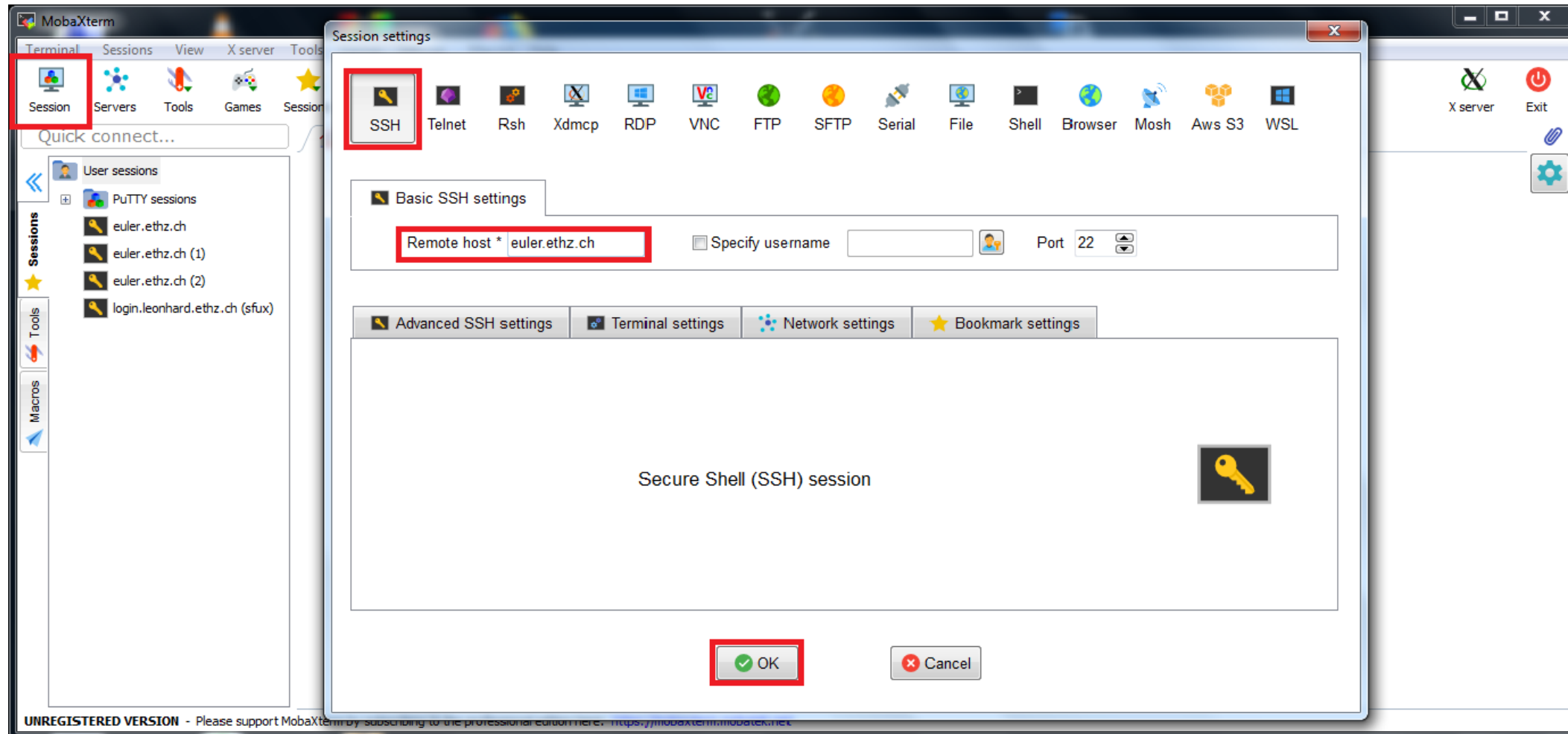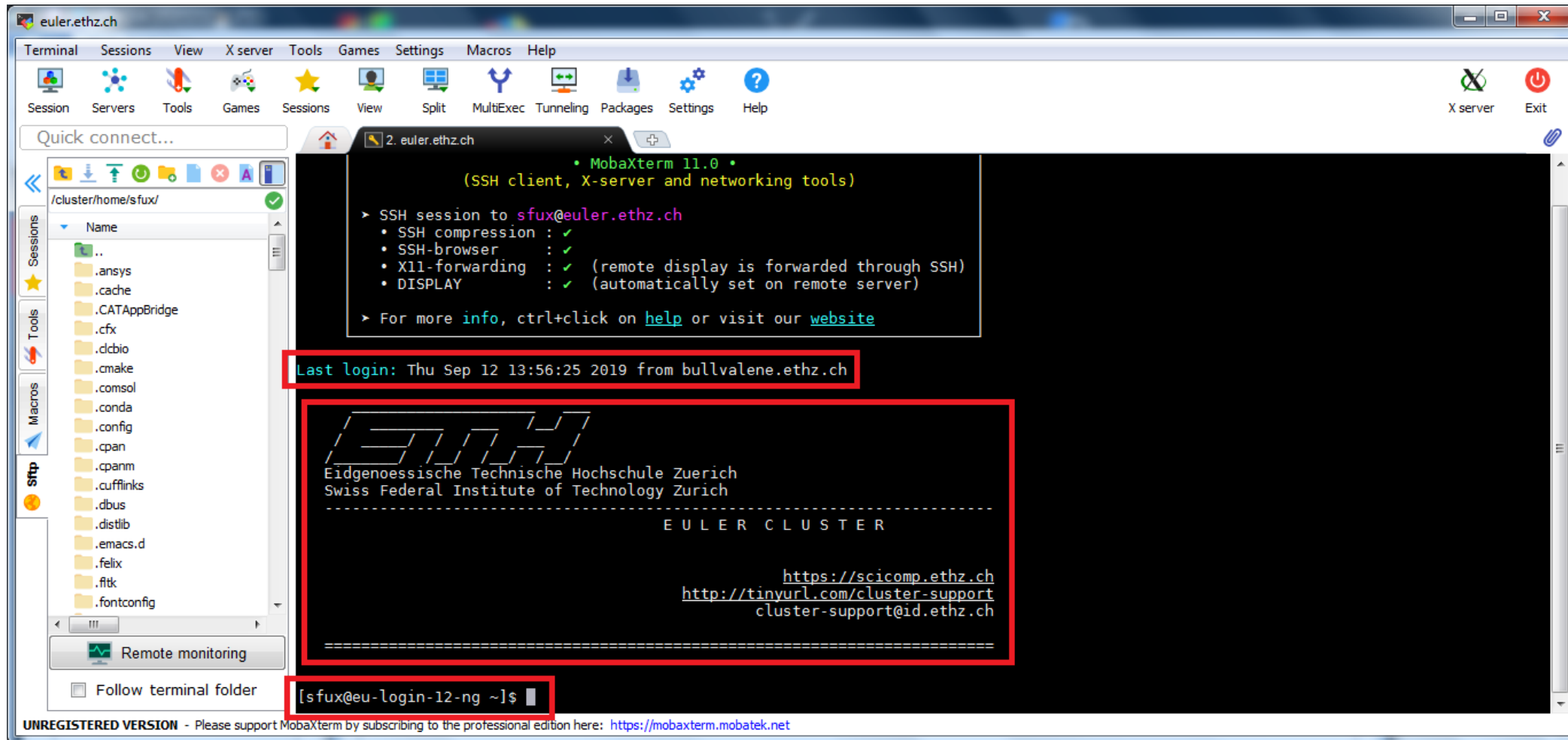
Access > SSH connection > Windows

Access > SSH connection > Windows

Access > Graphical user interface

Our clusters use the X Window System (X11) to display a program's graphical user interface (GUI) on your workstation

X11 server	Linux	macOS	Windows
Installation	Xorg Usually installed	XQuartz Installation needed	Included in MobaXterm

Usage	Use the log-in command
	
	`ssh -Y username@euler.ethz.ch`
	`ssh -Y username@login.leonhard.ethz.ch`
			Automatically enabled in MobaXterm

**ETH**zürich

Access > SSH keys
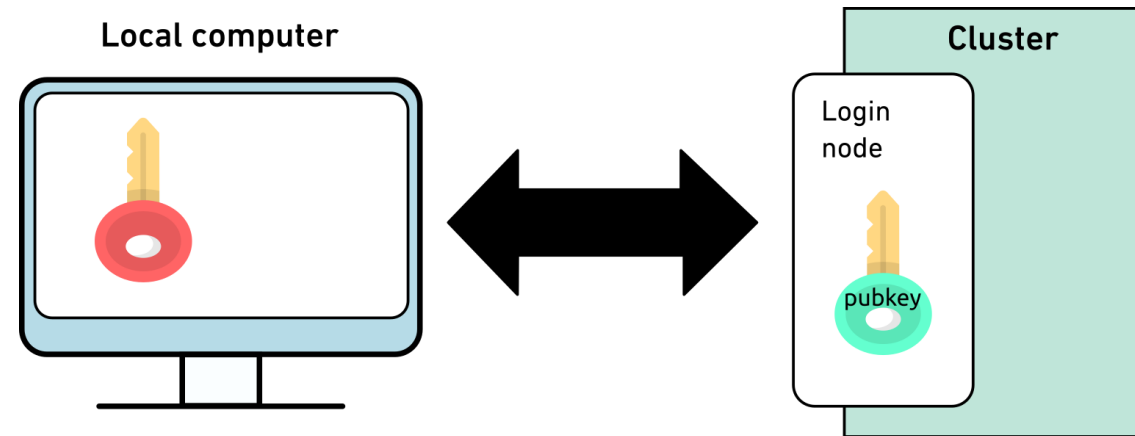
- SSH keys allows passwordless login
  - Useful for file transfers and automated tasks
  - When used properly, SSH keys are much safer than passwords

- SSH keys always come in pairs
  - A private key, stored on your local workstation (and nowhere else!)
  - A public key, stored on the computer(s) you want to connect to

Access > SSH keys
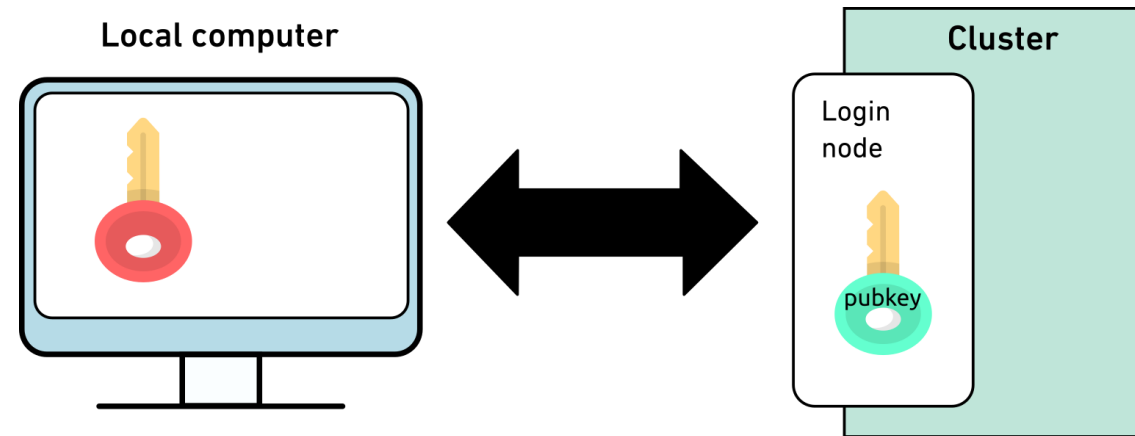
- You can generate as many pairs as you like, e.g., one for each computer you intend to connect to

- Keys should be protected with a passphrase

- SSH key management
  - `ssh-agent` unlocks keys for each terminal
  - `keychain` activates `ssh-agent` in every new terminal

Access > SSH keys > Create your keys

Local computer
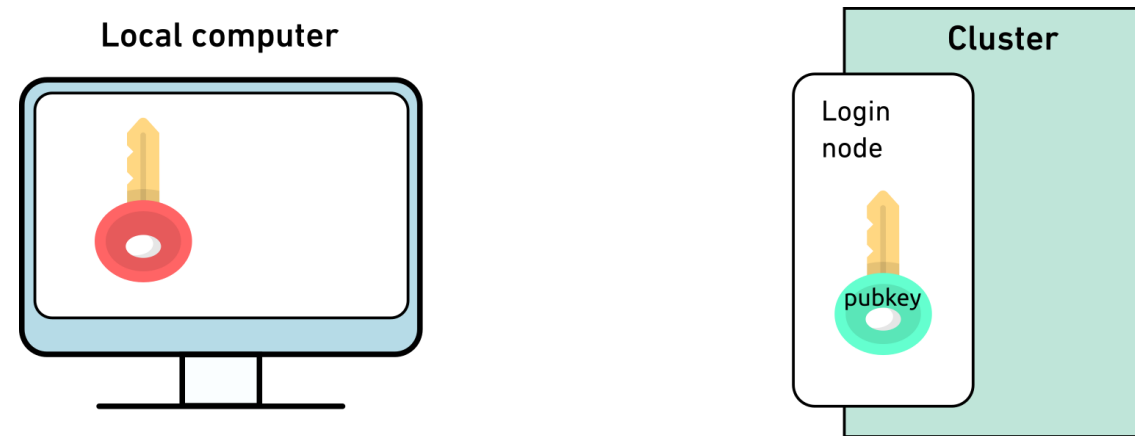
pubkey

Cluster

Login node
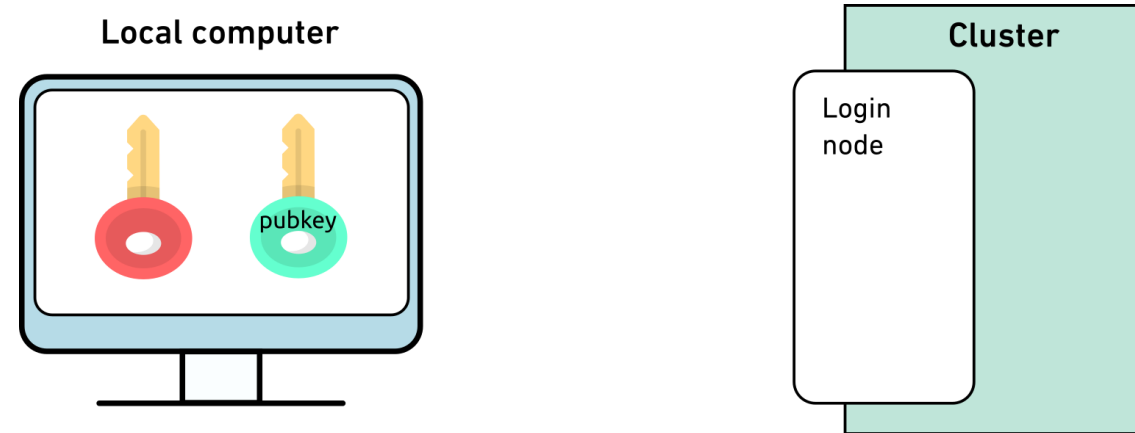
- Verify whether logging in with password works

- Generate a key pair with the ed25519 algorithm for each computer you want to connect to

  ```
  ssh-keygen -t ed25519 -f $HOME/.ssh/id_ed25519_euler
  ssh-keygen -t ed25519 -f $HOME/.ssh/id_ed25519_leonhard
  ```

- Enter a passphrase to protect your SSH keys
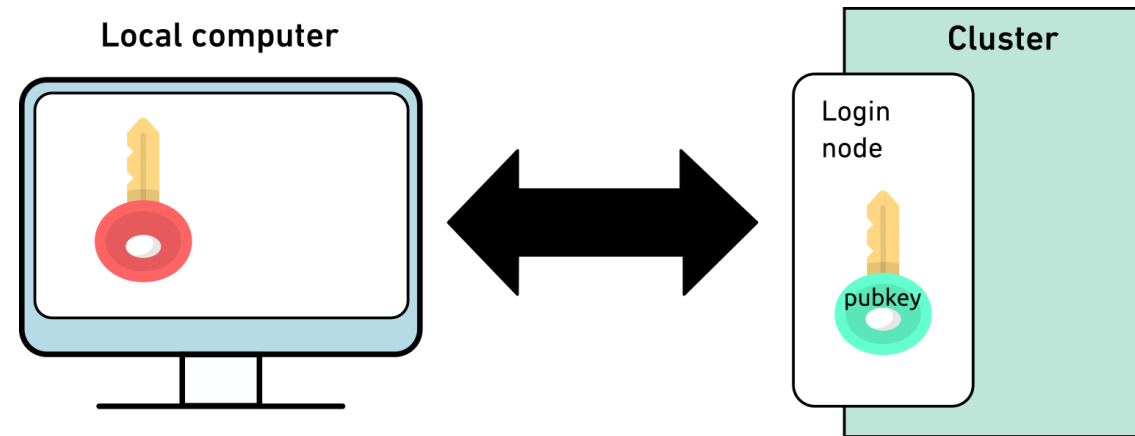
Access > SSH keys > Copy the public key to the cluster


```
ssh-copy-id -i $HOME/.ssh/id_ed25519_euler.pub    username@euler.ethz.ch

ssh-copy-id -i $HOME/.ssh/id_ed25519_leonhard.pub username@login.leonhard.ethz.ch
```

https://scicomp.ethz.ch/wiki/Accessing_the_clusters#SSH_keys

Access > SSH keys > Use keys with non-default names


```
ssh –i $HOME/.ssh/id_ed25519_euler      username@euler.ethz.ch

ssh -i $HOME/.ssh/id_ed25519_Leonhard   username@login.Leonhard.ethz.ch
```

https://scicomp.ethz.ch/wiki/Accessing_the_clusters#How_to_use_keys_with_non-default_names

Access > SSH keys > Use keys with non-default names

- SSH clients can use this option automatically by adding the option `IdentityFile` in your `$HOME/.ssh/config` file, e.g.:

```
Host euler
HostName euler.ethz.ch
User username
IdentityFile ~/.ssh/id_ed25519_euler
```

- Next time you login, you can type

```
$ ssh euler
```

https://scicomp.ethz.ch/wiki/Accessing_the_clusters#How_to_use_keys_with_non-default_names

ETHzürich   Scientific IT Services

Access > Demo: Creating SSH keys

```
$ ssh-keygen -t ed25519 -f $HOME/.ssh/id_ed25519_euler

$ ls -l $HOME/.ssh

$ ssh-copy-id -i $HOME/.ssh/id_ed25519_euler.pub username@euler.ethz.ch

$ ssh -i $HOME/.ssh/id_ed25519_euler username@euler.ethz.ch

$ exit
```

Outlook

- Introduction

- Accessing the cluster

- **Storage and data transfer**

- Modules and applications

- Getting help

Data > Available storage systems
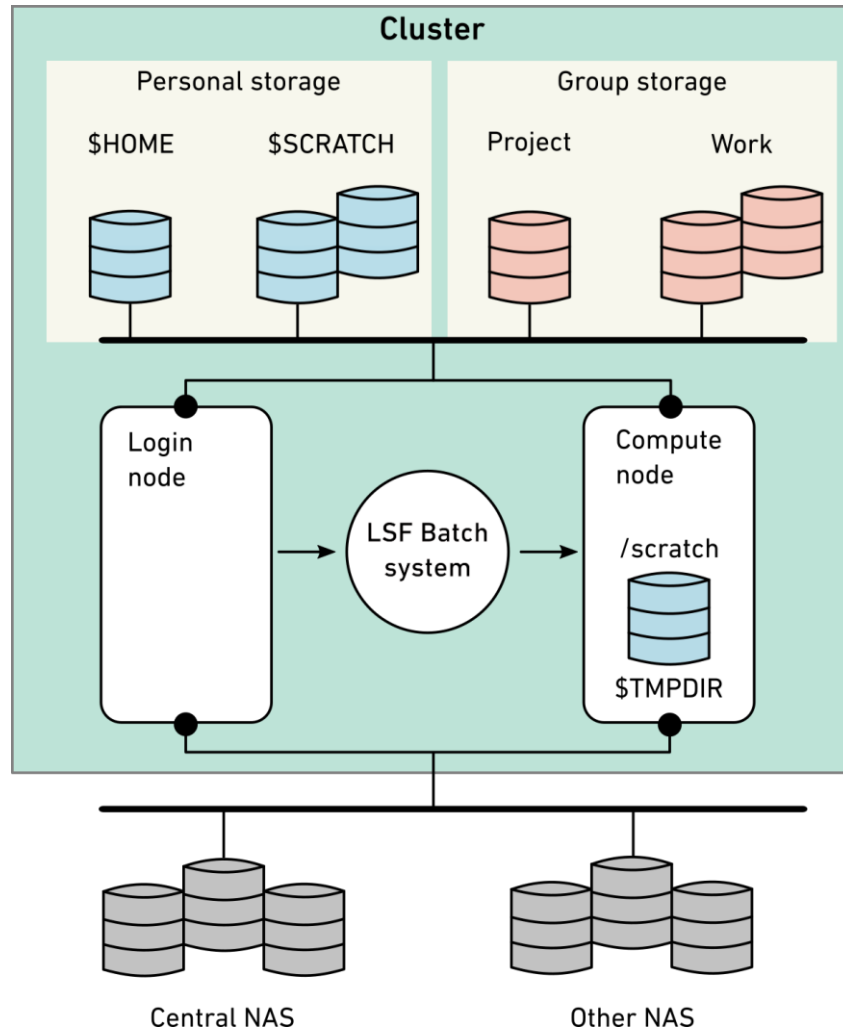
- Cluster wide storage systems
  - Home (personal)
  - Global scratch (personal)
  - Work (group)
  - Project (group)

- Local storage inside the compute node
  - Local scratch

- External storage
  - NAS
  - CDS
  - LTS

Data > Personal storage (every user) > Home

```
$ cd $HOME
$ pwd
/cluster/home/username
```

- Safe, long-term storage for critical data (program source, scripts, etc.)

- Accessible only by the user (owner); other people cannot read its contents

- Disk quota of 16/20 GB and a maximum of 80'000/100'000 files (soft/hard quota). Quota can be checked with the command `lquota`

- Contents saved every hour/day/week using snapshot. Users can access these snapshots in the hidden `.snapshot` directory

https://scicomp.ethz.ch/wiki/Storage_systems#Home

ETH *zürich*   Scientific IT Services

Data > Personal storage (every user) > Global scratch vs. local scratch

Global scratch

```
$ cd $SCRATCH
$ pwd
/cluster/scratch/username
```

- Fast, short-term storage for computations running on the cluster
- Created automatically upon first access
- Visible (mounted) only when accessed
- Strict usage rules; see $SCRATCH/__USAGE_RULES__ for details
- No backup

Local scratch

Local /scratch directory on each compute node (=$TMPDIR)

- Intended for serial, I/O-intensive applications
- Very short life span; data are deleted automatically when the job ends
- Scratch space must be requested by the job (see "batch system" later on)
- No backup

Data > Group storage (only shareholders) > Project vs. Work

Project

`/cluster/`**`project`**`/`*`groupname`*

- Similar to home, but for groups
- Safe, long-term storage for critical data

Work

`/cluster/`**`work`**`/`*`groupname`*

- Similar to global scratch, but without purge
- Fast, short- or medium-term storage for large computations
- Visible (mounted) only when accessed

- Shareholders can buy as much space as they need
- The access rights are managed by the owner
- Quota can be checked with `lquota`
- Backed up multiple times per week

Data > External storage > Central NAS vs. other NAS

Central NAS

Groups who have purchased storage on the central NAS of ETH can access it on our clusters

Other NAS

Groups who are operating their own NAS can export a shared file system via NFS to Euler

The user and group ID's on the NAS needs to be consistent with ETH user names and groups

- The NAS share needs to be mountable via NFSv3 (shares that only support CIFS cannot be mounted on the HPC clusters)

- The NAS share needs to be exported to the subnet of our HPC clusters.
  For central NAS, contact ID Systemdienste and ask them for an NFS export of your NAS share.

- Mounted automatically on our clusters under `/nfs/servername/sharename`

Data > File system comparison

File system	Life span	Snapshot	Backup	Max size	Small files	Large files
`/cluster/home`	Permanent	Yes	Yes	16 GB	+	o
`/cluster/scratch`	2 weeks	-	-	2.5 TB	o	++
`/cluster/project`	4 years	Optional	Yes	Flexible	+	+
`/cluster/work`	4 years	-	Yes	Flexible	o	++
local `/scratch`	Job	-	-	800 GB	++	o
central NAS	Flexible	Yes	Optional	Flexible	+	+

Retention time
Snapshots: up to 3 weeks
Backup: up to 90 days

Data > Copying data from/to the cluster (command line)

Secure copy (`scp`) is most commonly used to transfer files

```
scp [options] source destination
```

Examples: All the following examples need to be run on your local computer

- Upload a file from your workstation to Euler

```
scp local_file username@euler.ethz.ch:/path/to/remotedir
```

- Download a file from Euler to your workstation

```
scp username@euler.ethz.ch:/path/to/remote_file /path/to/localdir
```

- Copy a whole directory

```
scp -r localdir username@euler.ethz.ch:remotedir
```
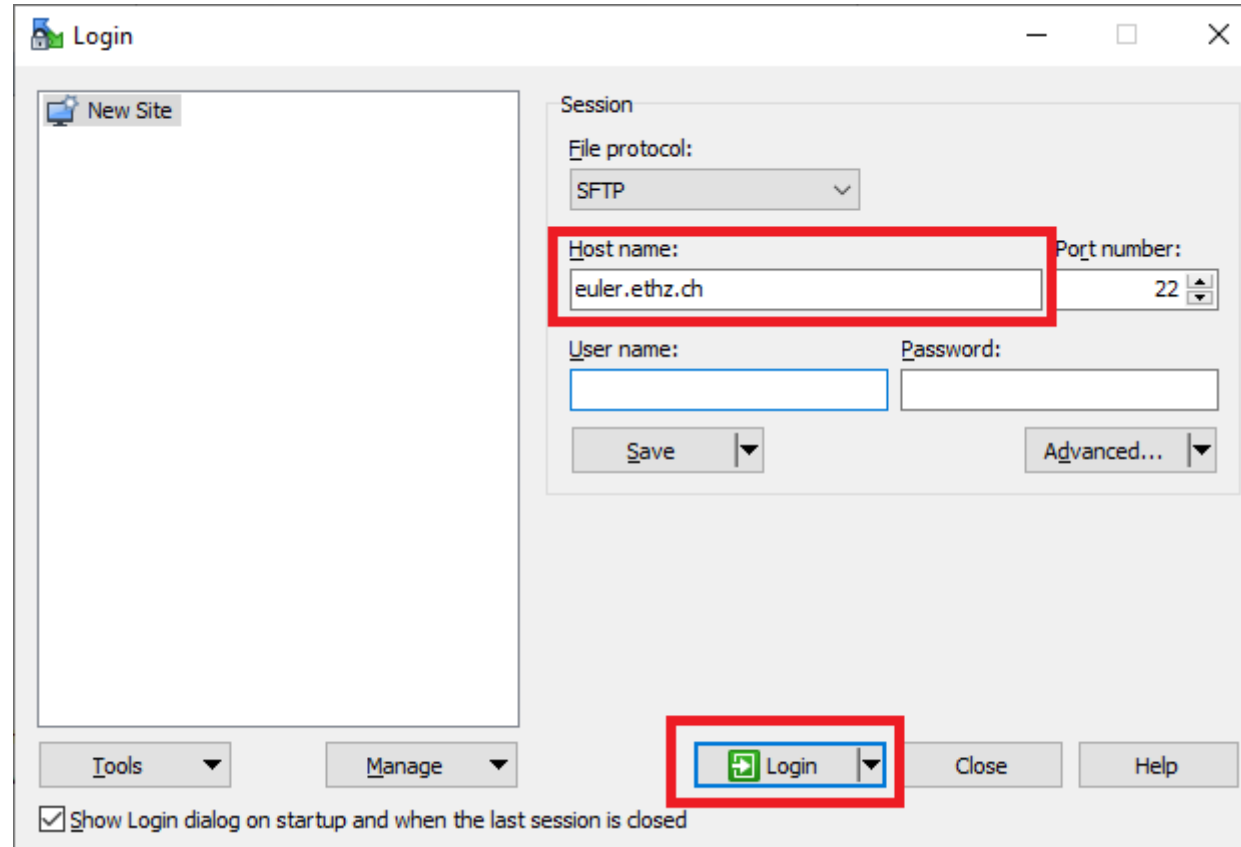
Alternatives to `scp`: `sftp`, `rsync`, `svn`, `git`, `wget`

ETH_zürich_   Scientific IT Services

Data > Copying data from/to the cluster (graphical user interface)

Graphical file transfer programs

Linux	macOS	Windows
FileZilla	FileZilla Cyberduck	WinSCP PSCP FileZilla Cyberduck

Data > Copying data from/to the cluster > WinSCP

Data > Copying data from/to the cluster > WinSCP
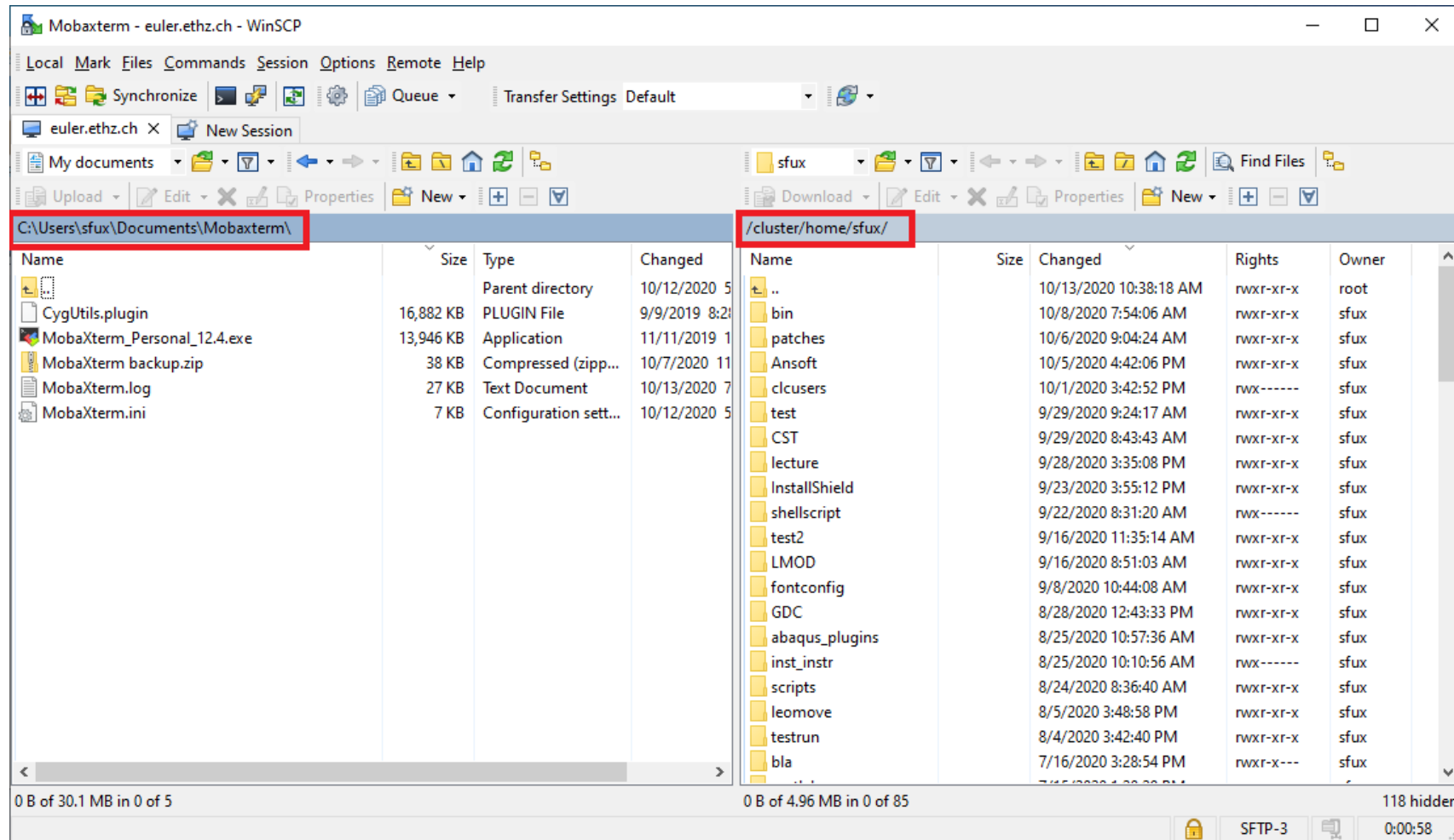
Scientific IT Services

Outlook

- Introduction

- Accessing the cluster

- Storage and data transfer

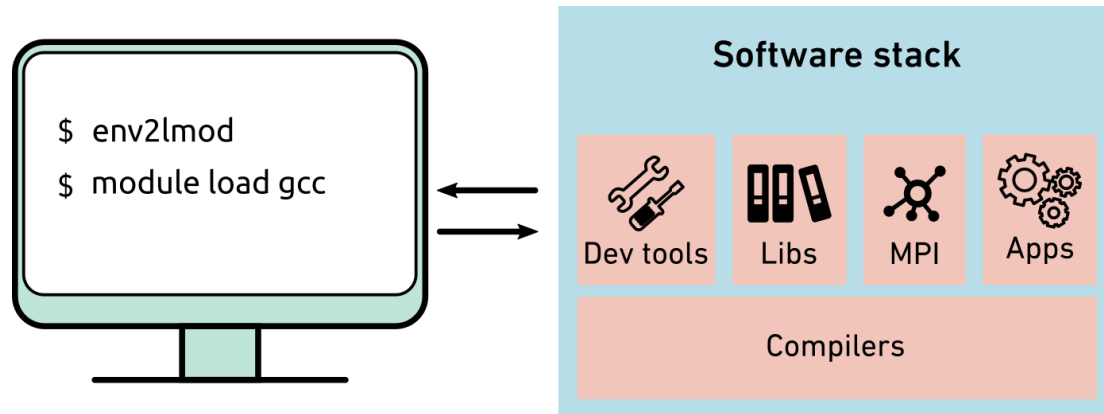- **Modules and applications**

- Getting help

Modules
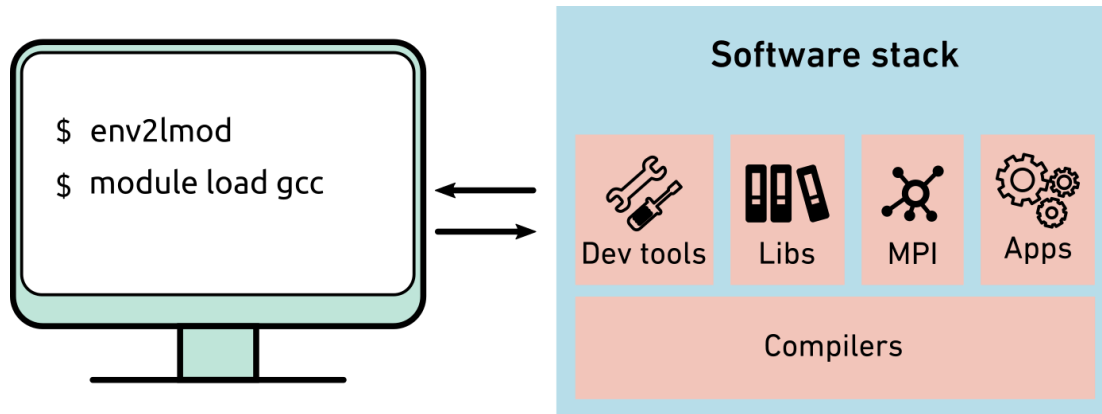
A Modules package is a tool to let users easily configure the computing environment which includes

- Development tools

- Scientific libraries

- Communication libraries (MPI)
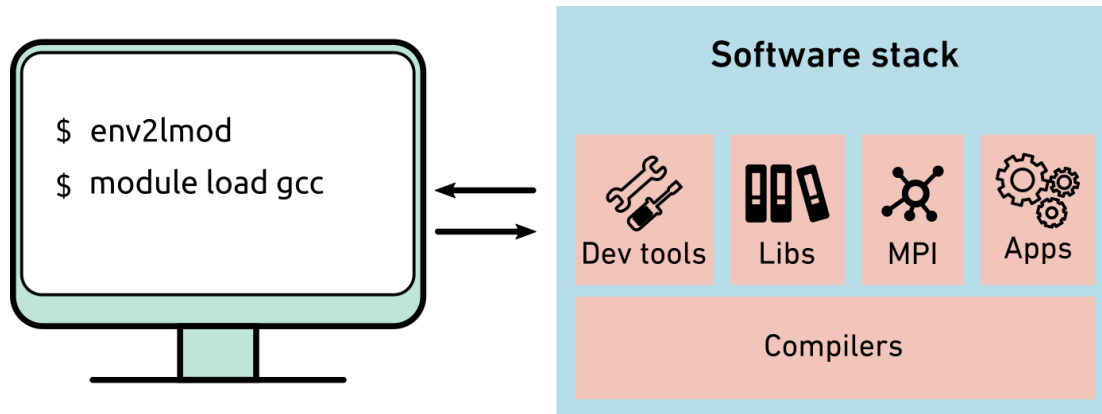
- Third-party applications

Modules

Advantages:

- Automatic configuration
- Different versions of the same software can co-exist and can be selected explicitly
- You can easily try out different tools, switch between versions, to find out which one works best for you

Modules

Software stack

Dev tools Libs MPI Apps

Compilers

```
$  env2lmod
$  module load gcc
```

We employ two Modules packages

- **LMOD Modules** implemented on Euler and Leonhard Open

- **Environment Modules** implemented on Euler

Modules > Commands

Load Python module in GCC/6.3.0 toolchain

```
[jarunanp@eu-login-31 ~]$ module load gcc/6.3.0 python/3.8.5
```

List available python module

```
[jarunanp@eu-login-31 ~]$ module avail python

------------------- /cluster/apps/lmodules/Compiler/gcc/6.3.0 --------------------

python/2.7.14     python/3.6.4     python/3.7.4     python/3.8.5 (D)     python_gpu/3.8.5
```

List all currently loaded modules

```
[jarunanp@eu-login-31 ~]$ module list

Currently Loaded Modules:
  1) StdEnv   2) gcc/6.3.0   3) openblas/0.2.20   4) python/3.8.5
```

Modules > Commands

`module`	get info about module sub-commands
`module avail`	list all modules available on the cluster
`module key` *`keyword`*	list all modules whose description contains *keyword*
`module help` *`name`*	get information about module *name*
`module show` *`name`*	show what module *name* does (<u>without</u> loading it)
`module unload` *`name`*	unload module *name*
`module purge`	unload all modules at once

Modules > Where are applications installed?

Centrally installed in /cluster/apps

Applications that are needed by many users should be installed centrally, like compilers and libraries

- Visible and accessible to all users via modules

- Installed and maintained by cluster support

- Commercial licenses provided by the IT shop of ETH or by research groups

In $HOME

Users can install additional applications in their home directory, but only if the quotas (space: 16 GB, files/directories: 100'000) are not exceeded

- Avoid anaconda installations as they often conflict with the files/directories quota

- For Python and R, packages can easily be installed locally

Modules > Two software stacks on Euler

Old software stack with Env Modules	**New software stack with LMOD Modules**

Only on Euler *Available on Euler and Leonhard Open*

- The old software stack is set as a default.
- The old software stack is provided for reproducibility

- All new software is installed exclusively in the new software stack

To switch from the old to the new software stack

To switch from the new to the old software stack

```
$ env2lmod
```

```
$lmod2env
```

Modules > Two software stacks on Euler

Old software stack with Env Modules

Only on Euler

Applications and tools were installed manually

- Mostly compiled with the standard/system compiler

- Limited implementation of toolchains available

- Only most important libraries available for all compilers/MPI

- Three module categories (legacy/supported/new)

New software stack with LMOD Modules

Available on Euler and Leonhard Open

Installations mostly done with SPACK package manager

- Automatic compilation/installation directly from source code

- Automatic creation of module files (LMOD)

- Predefined build recipes for more than 4'800 applications and libraries

- Large community behind SPACK (LLNL, ORNL, NASA, ETH, EPFL etc.)

https://scicomp.ethz.ch/wiki/New_SPACK_software_stack_on_Euler

Modules > Two software stacks on Euler

Old software stack with Env Modules

Only on Euler

Application pages (for instance R, Python, Matlab etc.) containing installation guides and examples how to submit batch jobs are also valid for the new software stack.

https://scicomp.ethz.ch/wiki/Category:Application

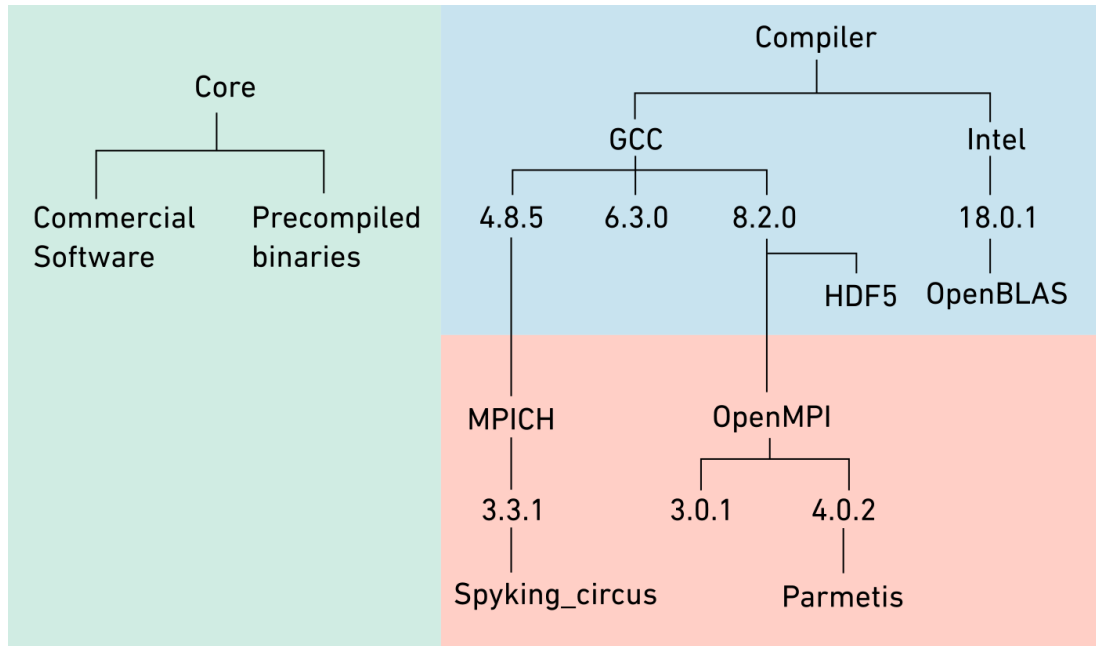**New software stack with LMOD Modules**

Available on Euler and Leonhard Open

Application lists:

https://scicomp.ethz.ch/wiki/Euler_applications_and_libraries

https://scicomp.ethz.ch/wiki/Leonhard_applications_and_libraries

Modules > LMOD modules

Module hierarchy with 3 layers:

- Core layer

```
$ module load comsol/5.6
```
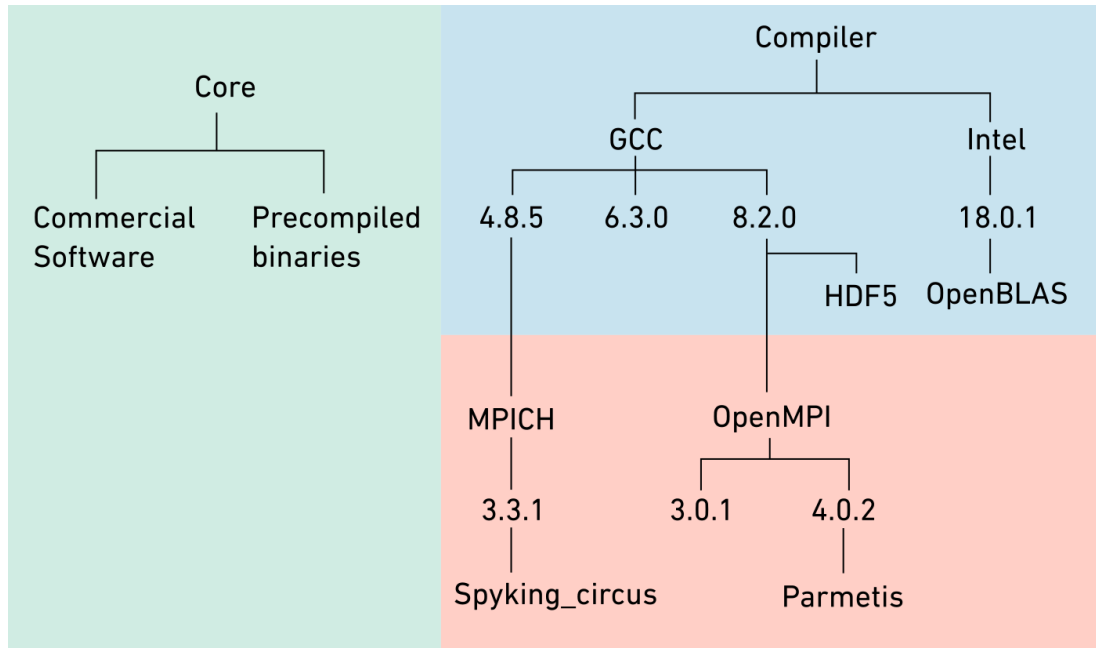
- Compiler layer

```
$ module load gcc/6.3.0 python/3.8.5
```

- MPI layer

```
$ module load gcc/6.3.0 openmpi/3.0.1 openblas
```
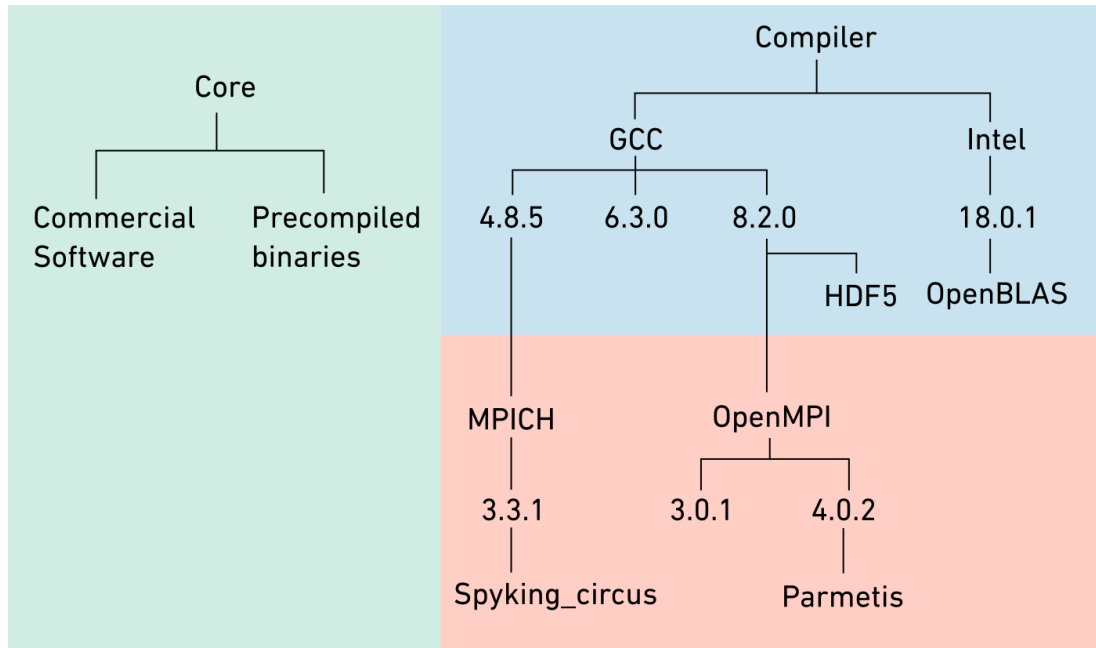
Modules > LMOD modules

The four main toolchains are

1. GCC 4.8.5 (supports C++11 standard)
2. GCC 6.3.0 (supports C++14 standard)
3. GCC 8.2.0 (supports C++17 standard)
4. Intel 18.0.1

- Combined with OpenMPI 3.0.1 or 4.0.2 and OpenBLAS

- For each of the toolchains ~400-500 applications and libraries are available, including many Linux low level libraries

Modules > LMOD modules

- Safety rules to avoid misconfiguration

- Only one compiler/MPI combination can be loaded at the same time

- When changing the compiler or MPI, LMOD will try to reload all currently loaded modules with the new compiler/MPI

Modules > Commercial / Open source applications

Categories	Application lists
Bioinformatics and life sciences	Bamtools, BLAST, Bowtie, CLC Genomics Server, RAxML, Relion, TopHat
Finite element methods	Ansys, Abaqus, FEniCS
Machine learning	PyTorch, Scikit-Learn, TensorFlow, Theano
Multi-physics phenomena	Ansys EDT, COMSOL Multiphysics, STAR-CCM+
Quantum chemistry and molecular dynamics	ADF, Ambertools, Gaussian, Molcas, Octopus, Orca, Qbox, Turbomole
Symbolic, numerical and statistical mathematics	Gurobi, Mathematica, MATLAB, R, Stata
Visualization	Ffmpeg, ParaView, VisIT, VTK

Modules > Software development

Development tools	
Compilers	GCC, Intel
Programming languages	C, C++, Fortran, Go, Java, Julia, Perl, Python, Ruby, Scala
Scientific libraries	Boost, Eigen, FFTW, GMP, GSL, HDF5, MKL, NetCDF, NumPy, OpenBLAS, SciPy
Solvers	PETSc, Gurobi, Hypre, Trilinos
MPI libraries	Open MPI, Intel MPI, MPICH
Build systems	GNU Autotools, Cmake, qmake, make
Version Control	CVS, Git, Mercurial, SVN

Modules > Demo: Python virtual environment

```
$ env2lmod

$ module load gcc/6.3.0 python/3.8.5

$ python -m venv --system-site-packages myenv

$ source myenv/bin/activate

(myenv)$ pip list

(myenv)$ pip install --upgrade pip
```

Outlook

- Introduction

- Accessing the cluster

- Storage and data transfer

- Modules and applications

- **Getting help**

Getting help

- Wiki: https://scicomp.ethz.ch

- Ticket system
  - https://smartdesk.ethz.ch (ETH account authentication)
  - Please describe your problem as accurately as possible

- E-mail
  - cluster-support@id.ethz.ch
  - Please do not send questions to individual members of the team

- Person-to-person (not during COVID 19)
  - Contact us to set up an appointment at your place
  - Visit us at Binzmühlestrasse 130

Dos and don'ts

Dos

- Understand what you are doing

- Ask for help if you don't understand what you are doing

- Optimize your workflow to make it as efficient as possible

- Keep in mind that our clusters are shared by many users

- Choose the file system you want to use carefully

Don'ts

- Don't waste CPU time or disk space

- Don't run applications on the login nodes

- Don't write large amounts of data to standard output

- Don't create millions of small files

- Don't run hundreds of small jobs if the same work can be done in a single job

ETH *zürich*

Questions?