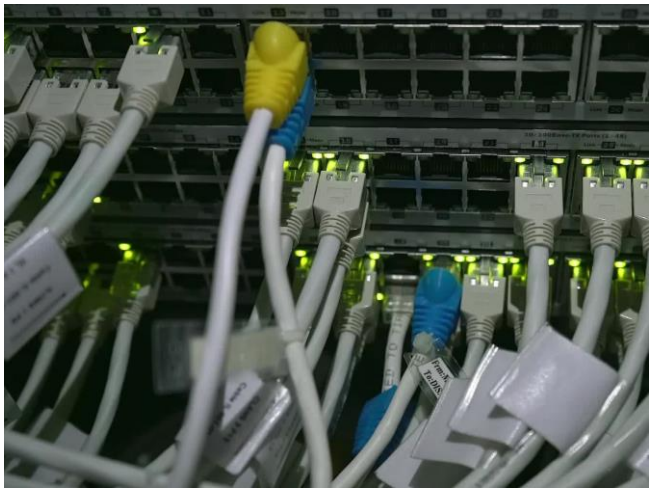
Connecting with SSH



CÉCI HPC Training 2023
juan.cabrera@unamur.be
olivier.mattelaer@uclouvain.be

- What is SSH (protocol)
- Create account (get private key)
- Where is used in CÉCI (context)
- How to install and use (practice)

What is SSH

- Secure Shell protocol
  - Client server model

  - Public key authentication
  - Encrypted communication
  - Encrypted data transfer
  - No password over the network

What is SSH

- Secure Shell protocol
  - Client server model

  - Public key authentication
  - Encrypted communication
  - Encrypted data transfer
  - No password over the network

Authentication

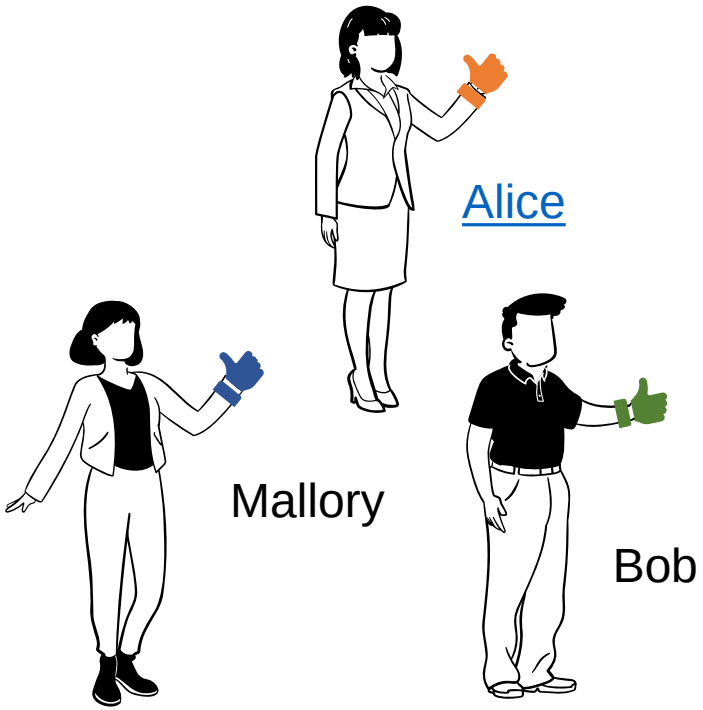
- Private key
  - Identity key
  - Keep in a safe place
  - Encrypted by a passphrase
  - File **Id_<algorithm>**
- Public key
  - Linked to the private key
  - Set in the server you need access
  - Authenticates the user
  - File **id_<algorithm>.pub**

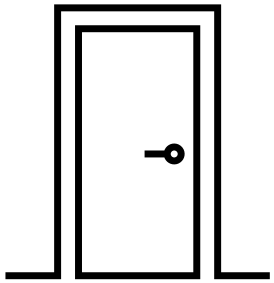
Authentication

- Digital signature
  - **Private key** create a digital signature≈ Thump 
  - **Public key** verify a digital signature ≈ fingerprint 

Authentication

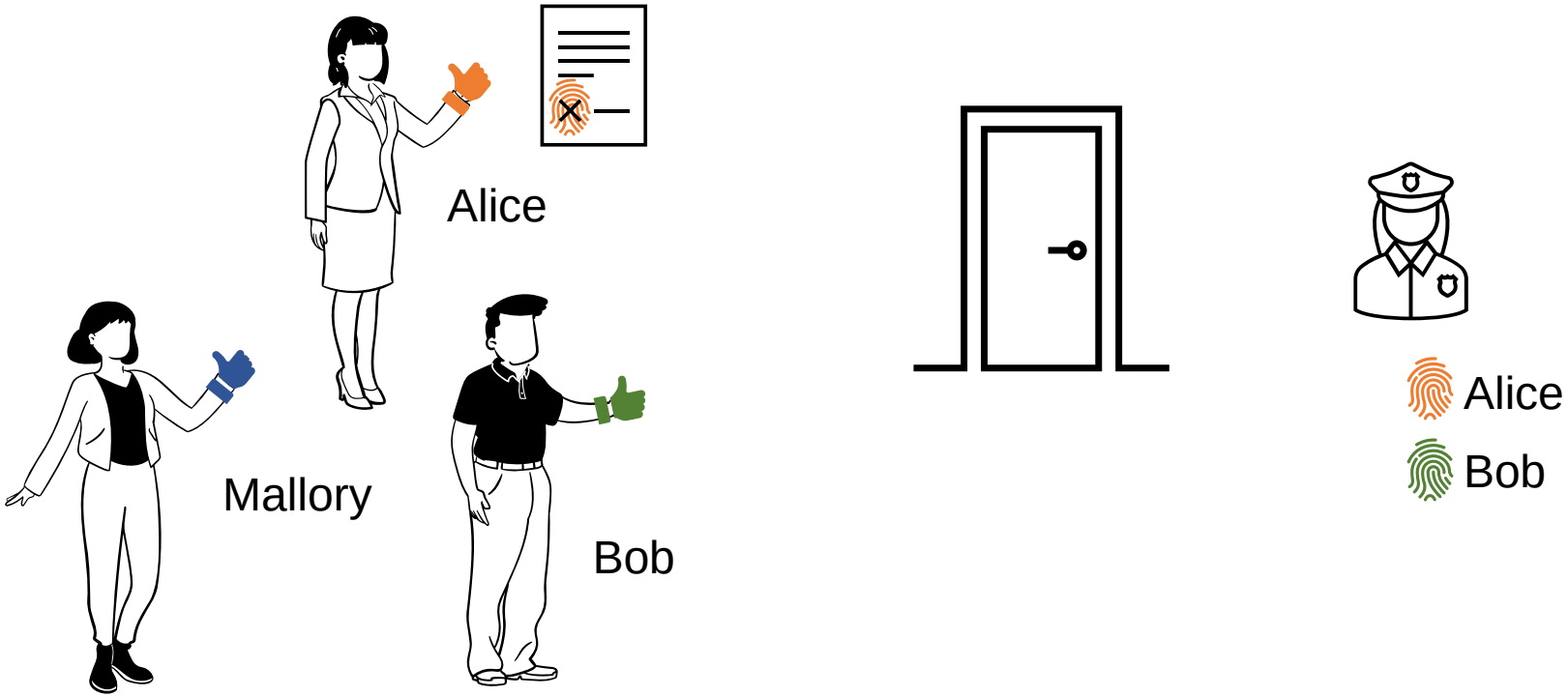
- Digital signature
  - **Private key** create a digital signature≈ Thumb
  - **Public key** verify a digital signature ≈ fingerprint





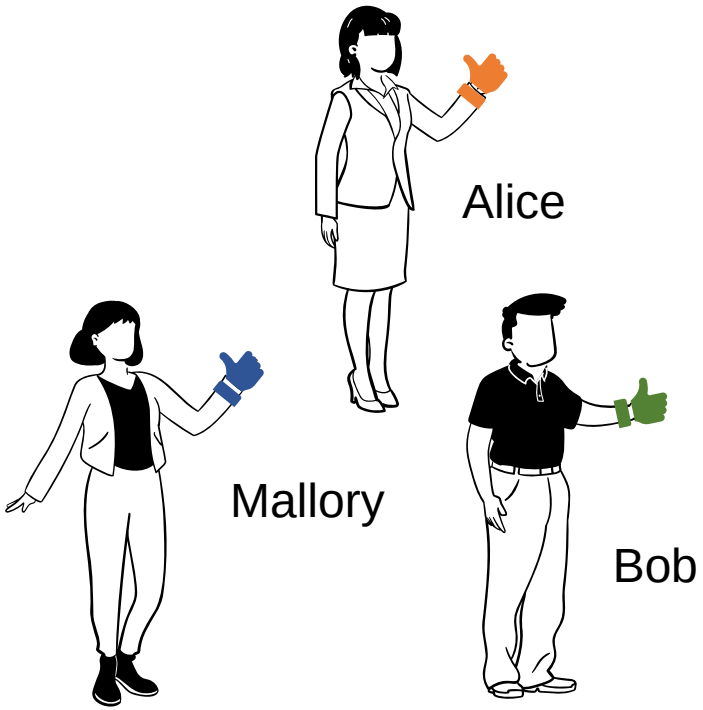
Authentication

- Digital signature
  - **Private key** create a digital signature≈ Thumb
  - **Public key** verify a digital signature ≈ fingerprint



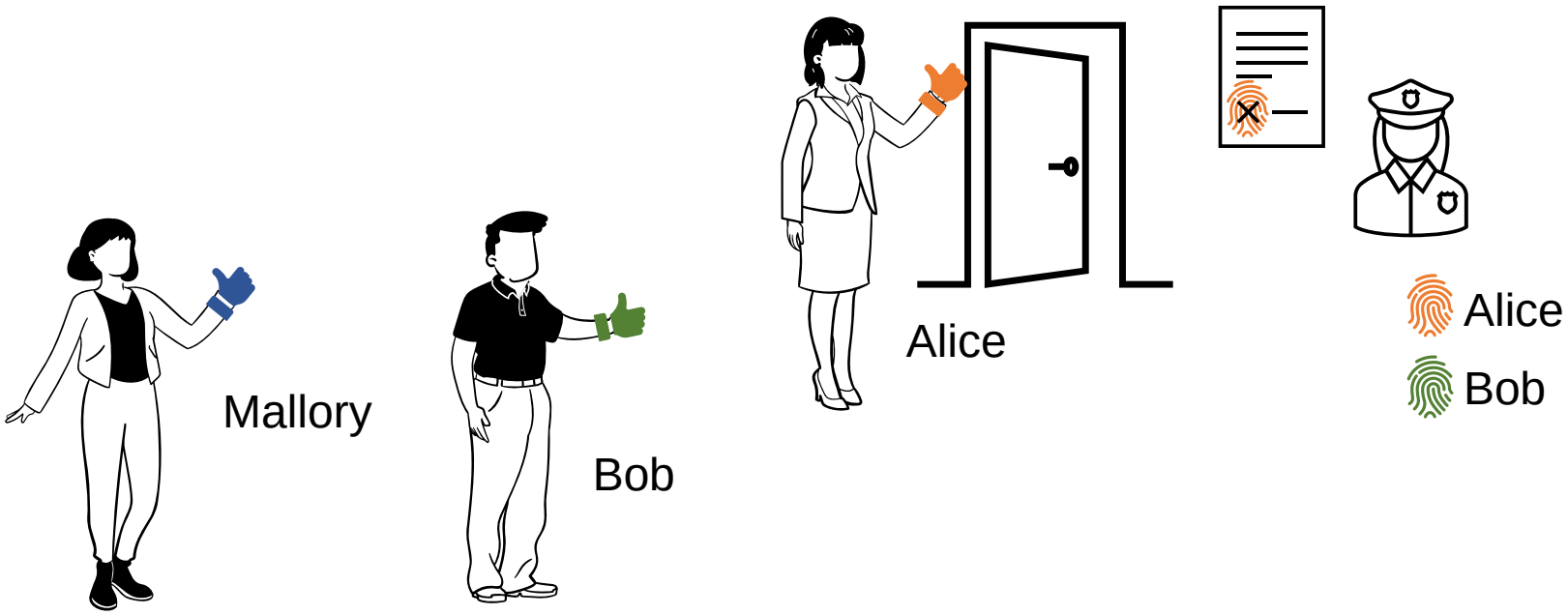
Authentication

- Digital signature
  - **Private key** create a digital signature≈ Thumb
  - **Public key** verify a digital signature ≈ fingerprint



Authentication

- Digital signature
  - **Private key** create a digital signature≈ Thumb
  - **Public key** verify a digital signature ≈ fingerprint

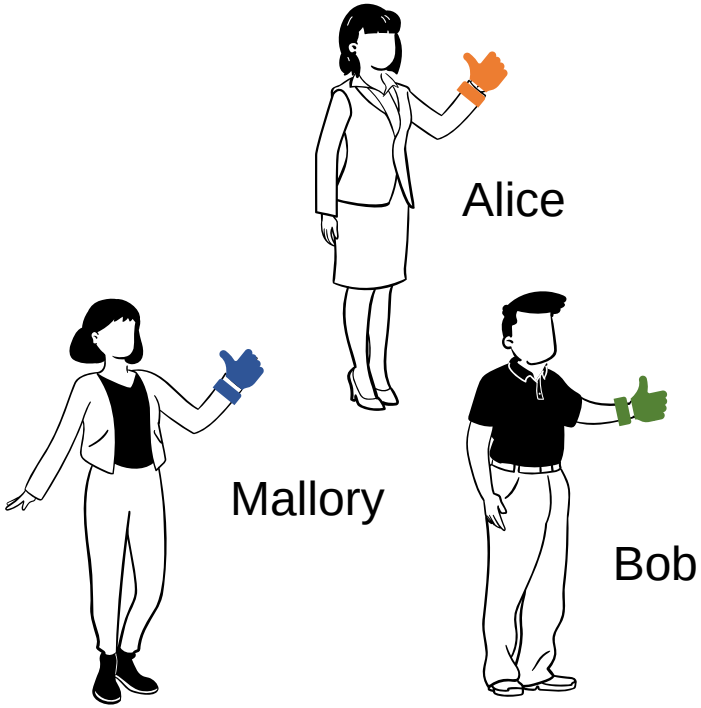


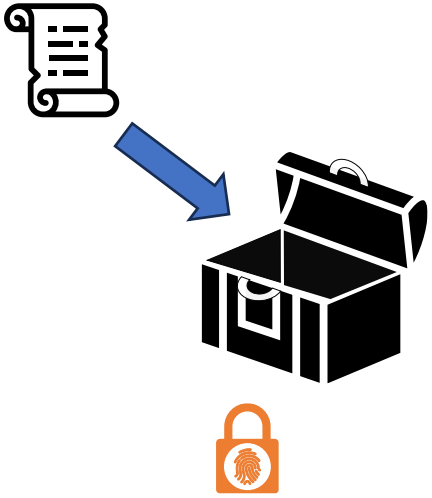
Encryption

- Encrypt/Decrypt data
  - **Private key** Decrypt ≈ Thumb
  - **Public key** Encrypt ≈ Biometric Lock

Encryption

- Encrypt/Decrypt data
  - **Private key** Decrypt ≈ Thumb
  - **Public key** Encrypt ≈ Biometric Lock





Authentication

- Private Key. Keep save

```
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,798194AFB2800B27

KnvjN+KM4NogUADgdVI7GawGEmxJtXl2NKbezDyI8aeUAYxHemgTh
fCeAJkTZ/B23uAWRppVvuPwJtp/AD3cvYxY5jBvSwVlAUdrfOJauegGc
...
...
wT/yGuuRi9xfn6/yY7wTDxeaJg5WRd54oq0jbpTPUQmZWjJ1cuzBNiio
OJkZChE7fLD+C7kvYH0J6u4NiXUWqVheNerl0OnCZuM770gY5P0Q7
-----END RSA PRIVATE KEY-----
```

- Public key. Set in frontends and gateways

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABgQCejTMdLq2r2c7rKGbRF1lae4Z7hUrASpLb5+……
.….+hQErnsEvWdpH+UFLaVFQ6b2GGXoTjh4+yoSX/++Ru4cgLT/+xbhBYRylaN1Ut1Ic=
relog@ceci-relog.segi.ulg.ac.be
```

SSH Connexion

Client

Alice

User Alice in client wants to connect to server Spock

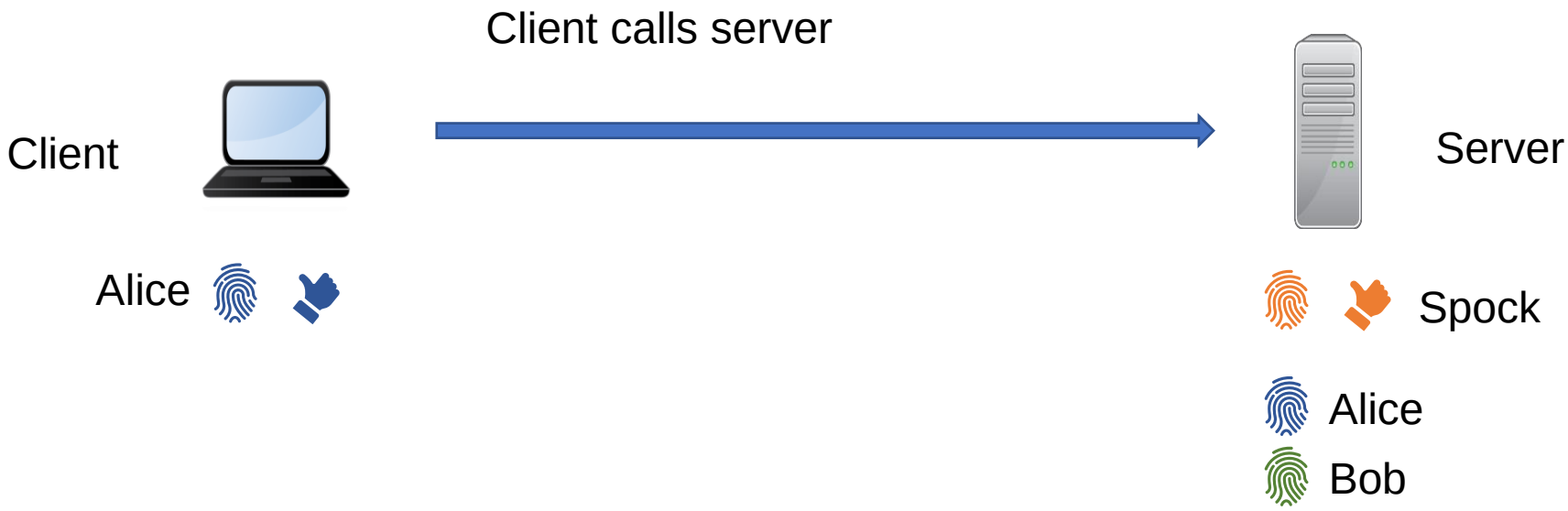
Server

Spock

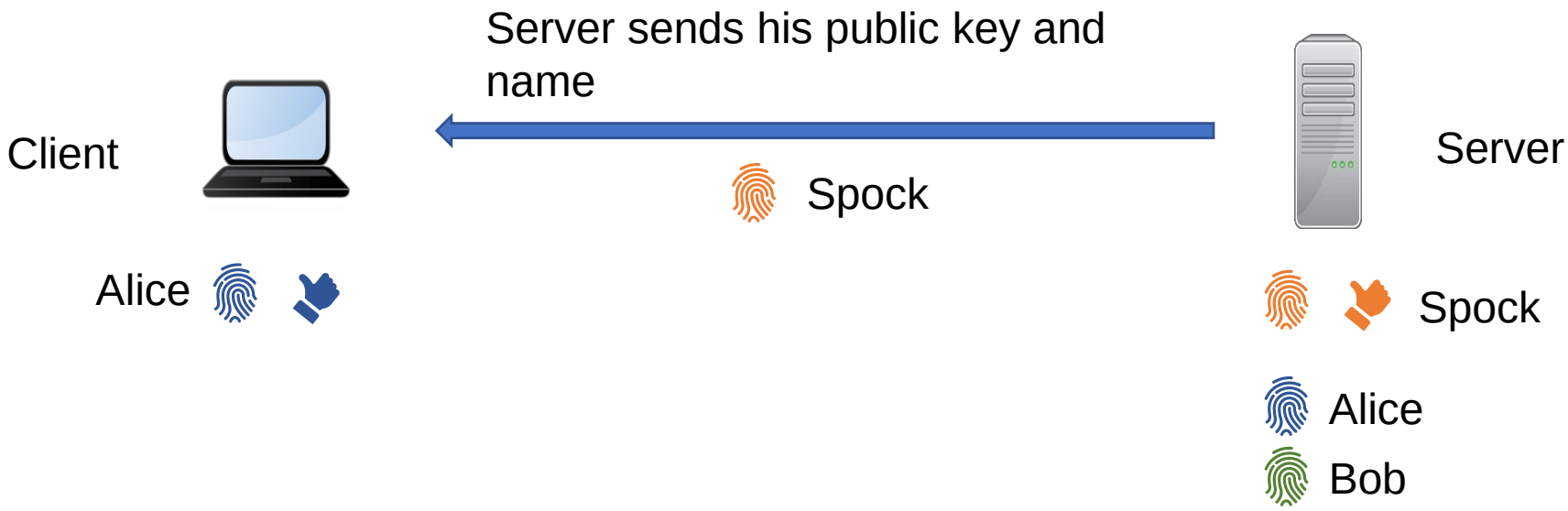
Alice

Bob

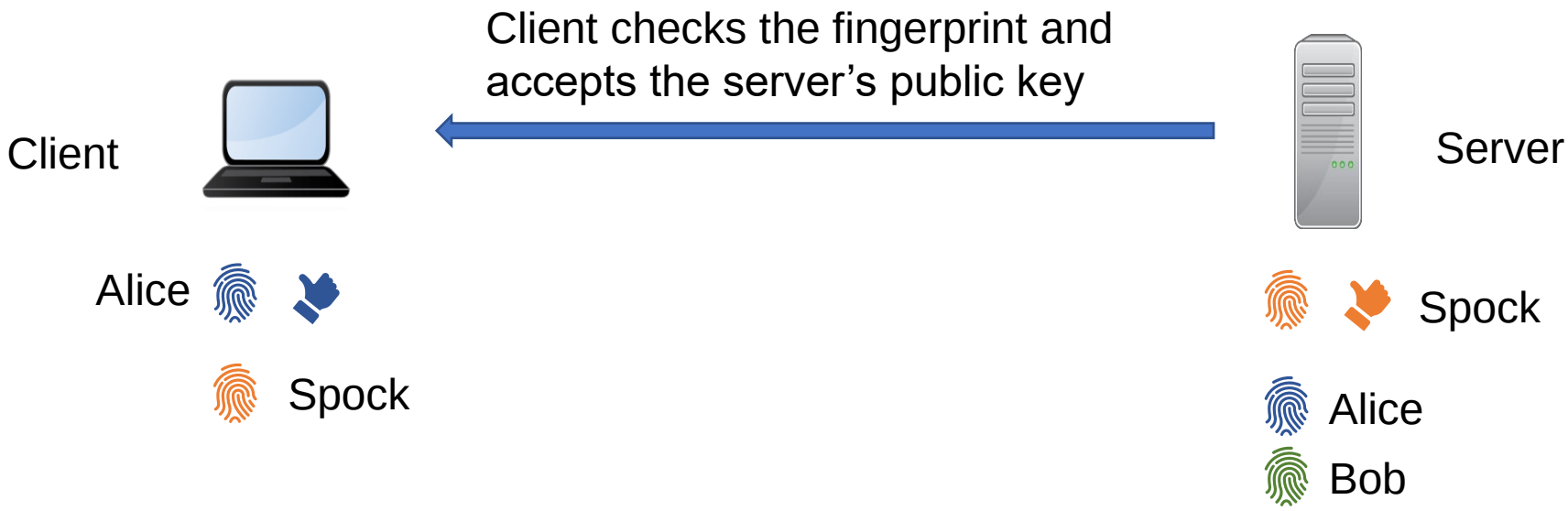
SSH Connexion



SSH Connexion

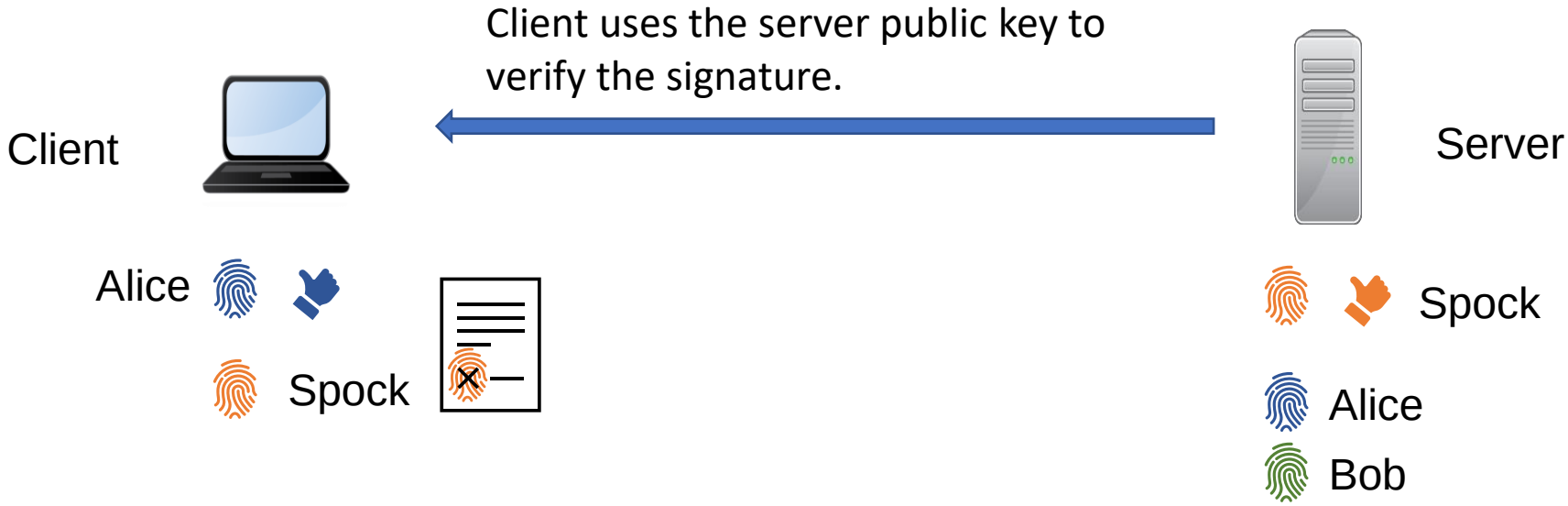


SSH Connexion

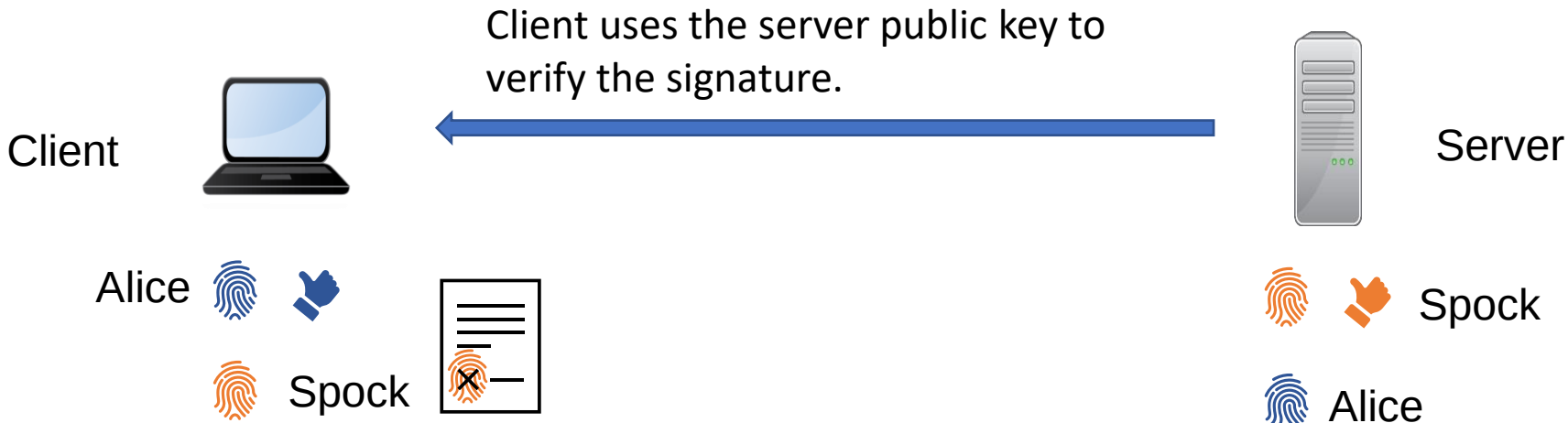


SSH Connexion

SSH Connexion

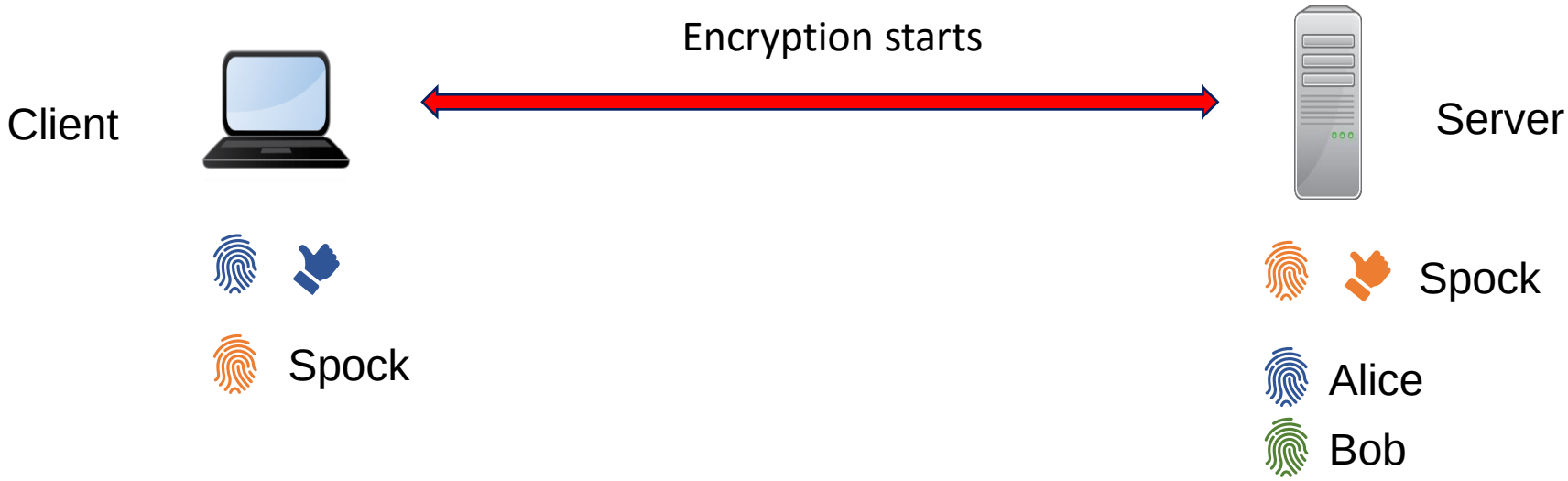


SSH Connexion



- **This proves the server's authenticity**

SSH Connexion



SSH Connexion

Client sends:

- The User name
- The User public key
- A message signed with The User private key

Client

Alice

Spock

Alice

Server

Spock

Alice

Bob

SSH Connexion

Server checks if public key and user are in allowed users
Server uses the user public key to verify the signature.

Client

Alice

Spock

Server

Spock

Alice

Bob

SSH Connexion

Server checks if public key and user are in allowed users
Server uses the user public key to verify the signature.

Client

Server

Alice

Spock

Spock

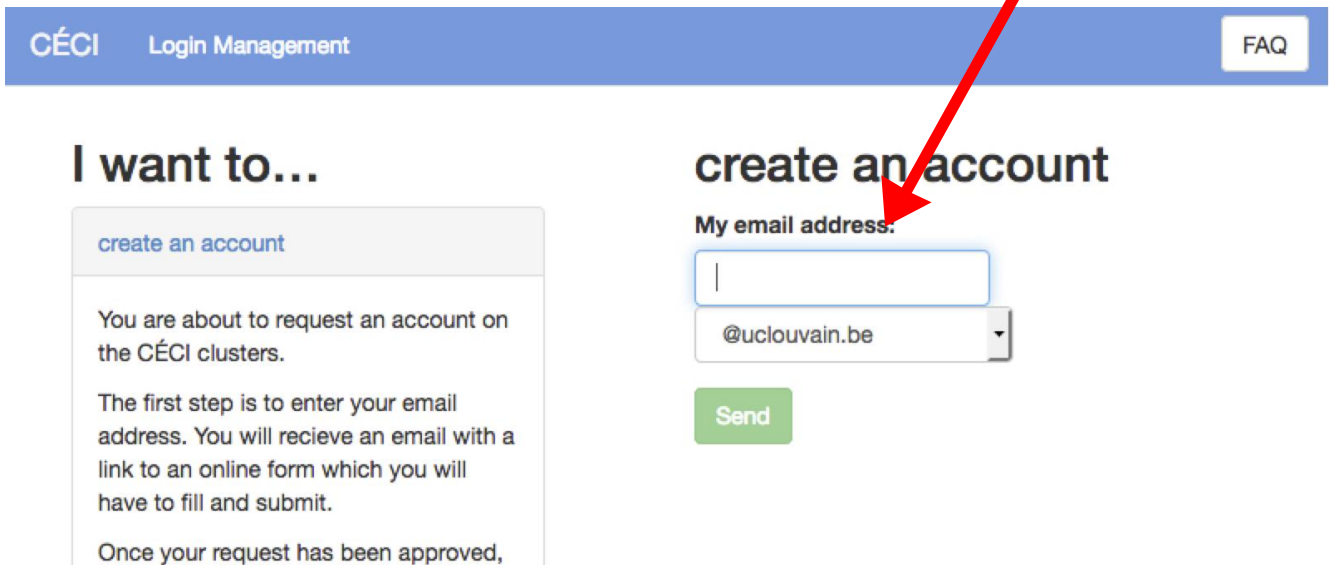
Alice

Bob

- **This proves the user authenticity**

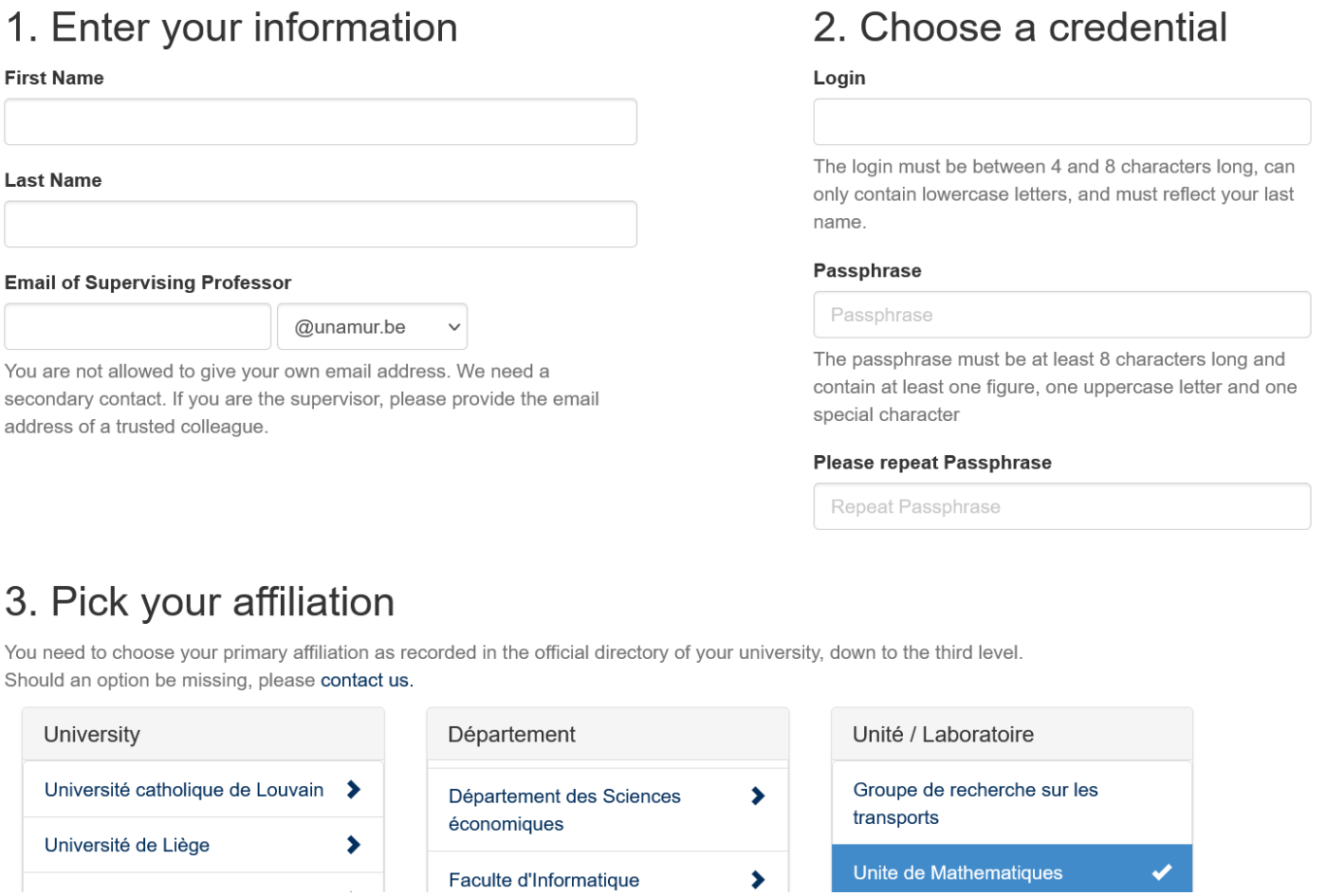
How to get your private key

https://www.ceci-hpc.be



How to get your private key

Click on the link sent to you by email and fill-in the form.



How to get your private key

Click on the link sent to you by email and fill-in the form.

1. Enter your information

First Name

Last Name

Email of Supervising Professor

@unamur.be

You are not allowed to give your own email address. We need a secondary contact. If you are the supervisor, please provide the email address of a trusted colleague.

2. Choose a credential

Login

UNamur users
Use your UNamur eid

The login must be between 4 and 8 characters long, can only contain lowercase letters, and must reflect your last name.

Passphrase

Passphrase

The passphrase must be at least 8 characters long and contain at least one figure, one uppercase letter and one special character

Please repeat Passphrase

Repeat Passphrase

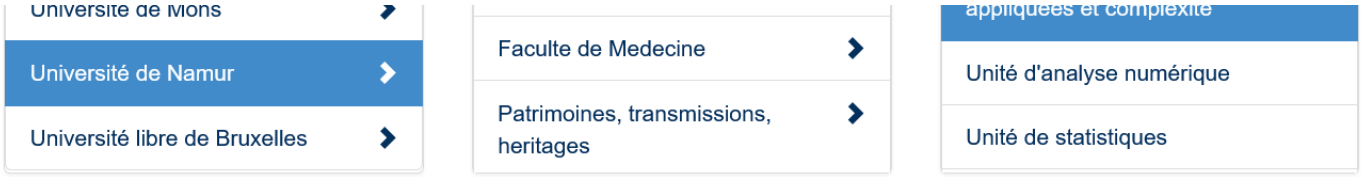
3. Pick your affiliation

You need to choose your primary affiliation as recorded in the official directory of your university, down to the third level.
Should an option be missing, please contact us.

University	Département	Unité / Laboratoire
Université catholique de Louvain	Département des Sciences économiques	Groupe de recherche sur les transports
Université de Liège	Faculte d'Informatique	Unite de Mathematiques

How to get your private key

Agree to the terms and conditions and submit



4. Projects

Projects allow access to supplementary resources, a. o. the Tier-1 cluster. Please make sure you have the authorization from the project owner before choosing one. You will need to know the acronym of the project.

cecisys | Add project

☐ I have read and agree to the terms and conditions.

5. Submit!

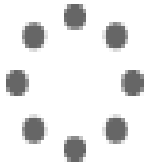
Once the form is completely filled-in, click the 'Send' button. A system administrator will review your request. **Send**

If you want to save the current data and submit it later, click the 'Save' button. **Save**

How to get your private key

Wait ...

A sysadmin is reviewing your information



How to get your private key

Sysadmin confirms the account.

How to get your private key

Sysadmin confirms the account.

Private and **public** key are generated

How to get your private key

Sysadmin confirms the account.

Private and public key are generated

The private key is **encrypted using the passphrase**

How to get your private key

Sysadmin confirms the account.

Private and public key are generated

The private key is encrypted using the passphrase and **sent to you by email**

How to get your private key

Sysadmin confirms the account.

Private and public key are generated

The private key is encrypted using the passphrase and **sent to you by email**

WARNING For security reasons

CÉCI does not keep a copy of your private key.

If you lose your key or passphrase or think it is compromised, you must **renew your CÉCI account** at https://login.ceci-hpc.be

How to get your private key

Sysadmin confirms the account.

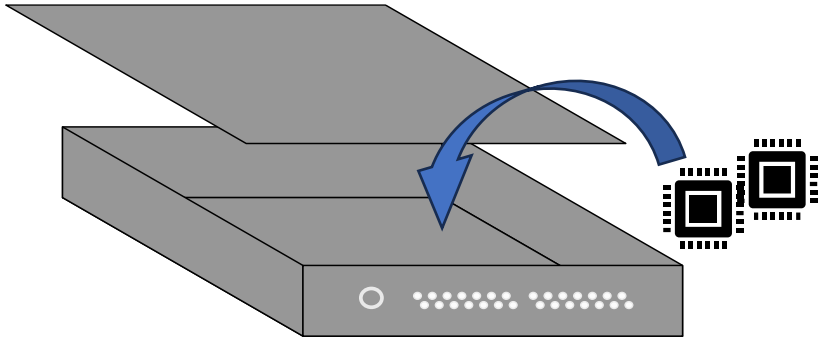
Private and public key are generated

The private key is encrypted using the passphrase and sent to you by email

Your public key is sent to each CÉCI server for authentication

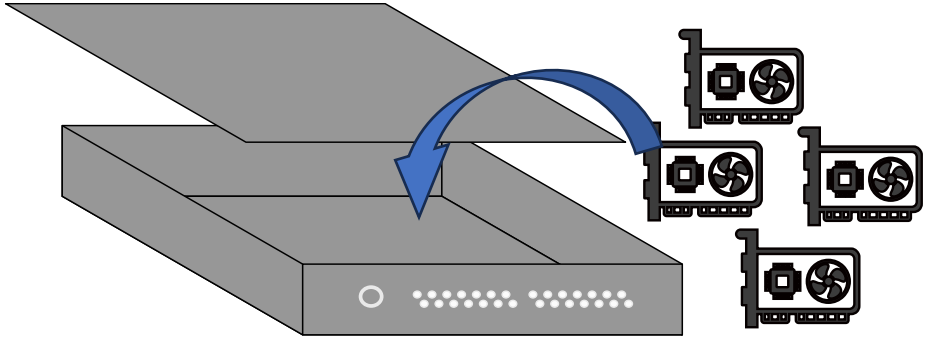
Exercise: Get your private key

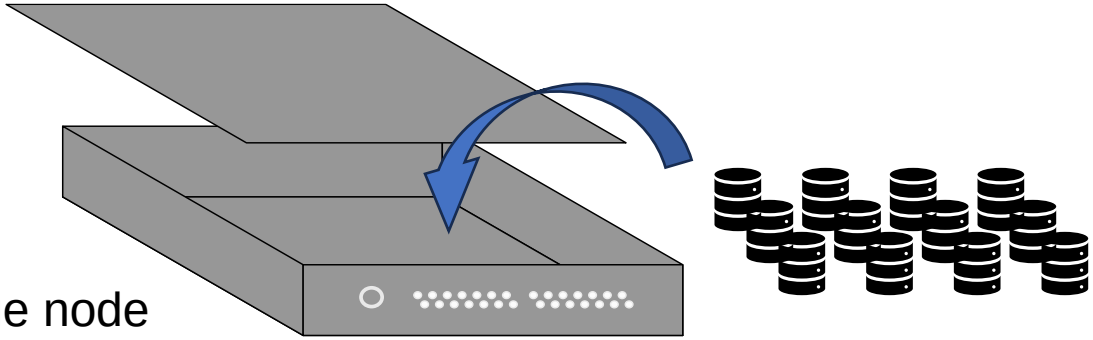
CONTEXT



Compute CPU node
Calculations server

Compute GPU node
Calculations server



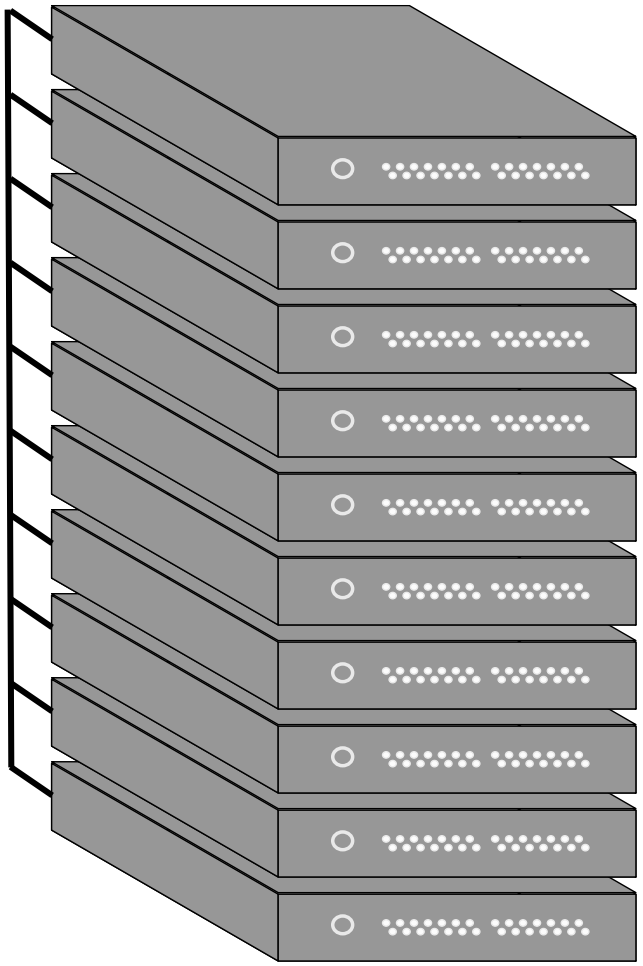


Storage node
File server

CONTEXT

Cluster: stack of nodes

Private network



CONTEXT

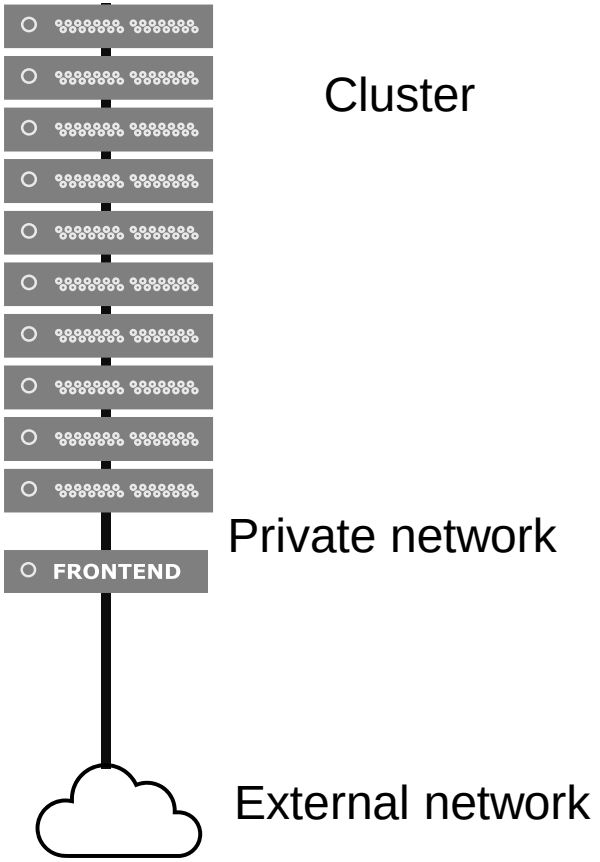
CONTEXT

Frontend

You need to connect with SSH to the **frontend** to :

- **submit jobs** to the compute nodes (**SLURM**)
- **access** your results
- **edit** your files
- **compile (use debug partition)**
- **transfer** your data

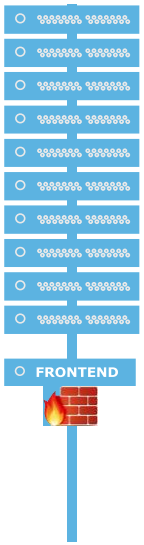
Do not run heavy jobs on the frontend



CÉCI Tier 2

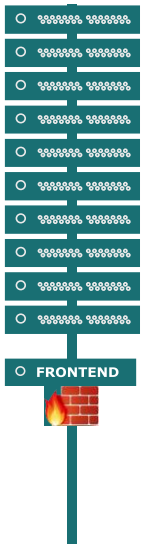
Tier 1

Lemaitre 3

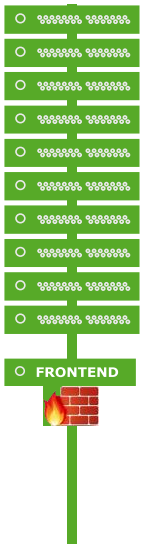


Lyra

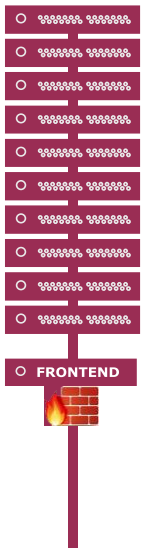
Nic5



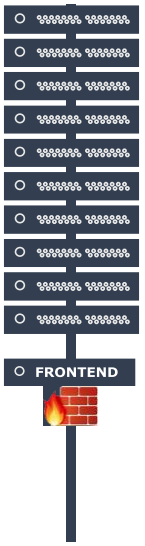
Hercules 2



Dragon 2



Lucia



5 CÉCI clusters

UCLouvain Université

ULB

LIÈGE université

UNIVERSITÉ DE NAMUR

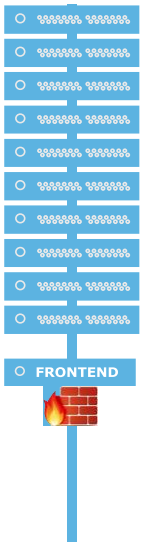
UMONS

Cenaero

CÉCI Tier 2

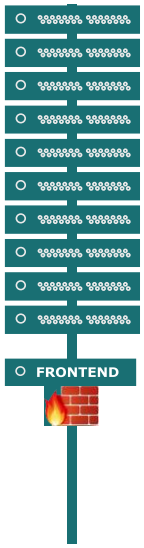
Tier 1

Lemaitre 3

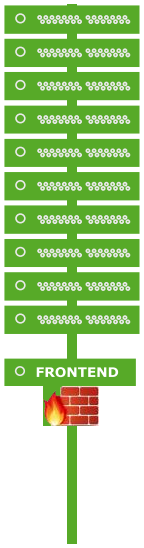


Lyra

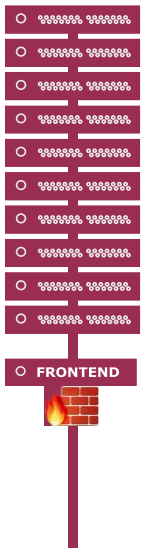
Nic5



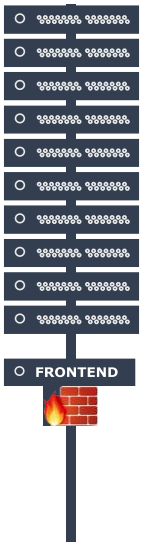
Hercules 2



Dragon 2



Lucia



5 CÉCI clusters

Cenaero Cluster

UCLouvain Université

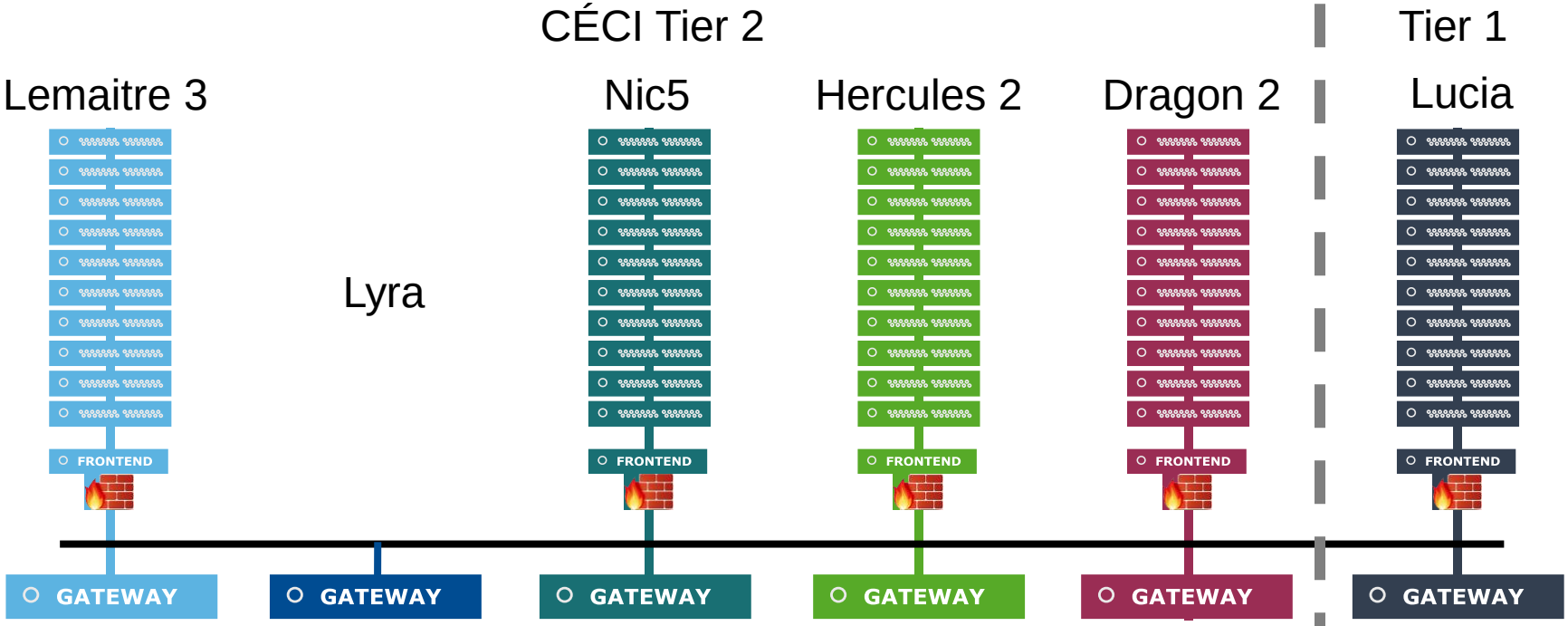
ULB

LIÈGE université

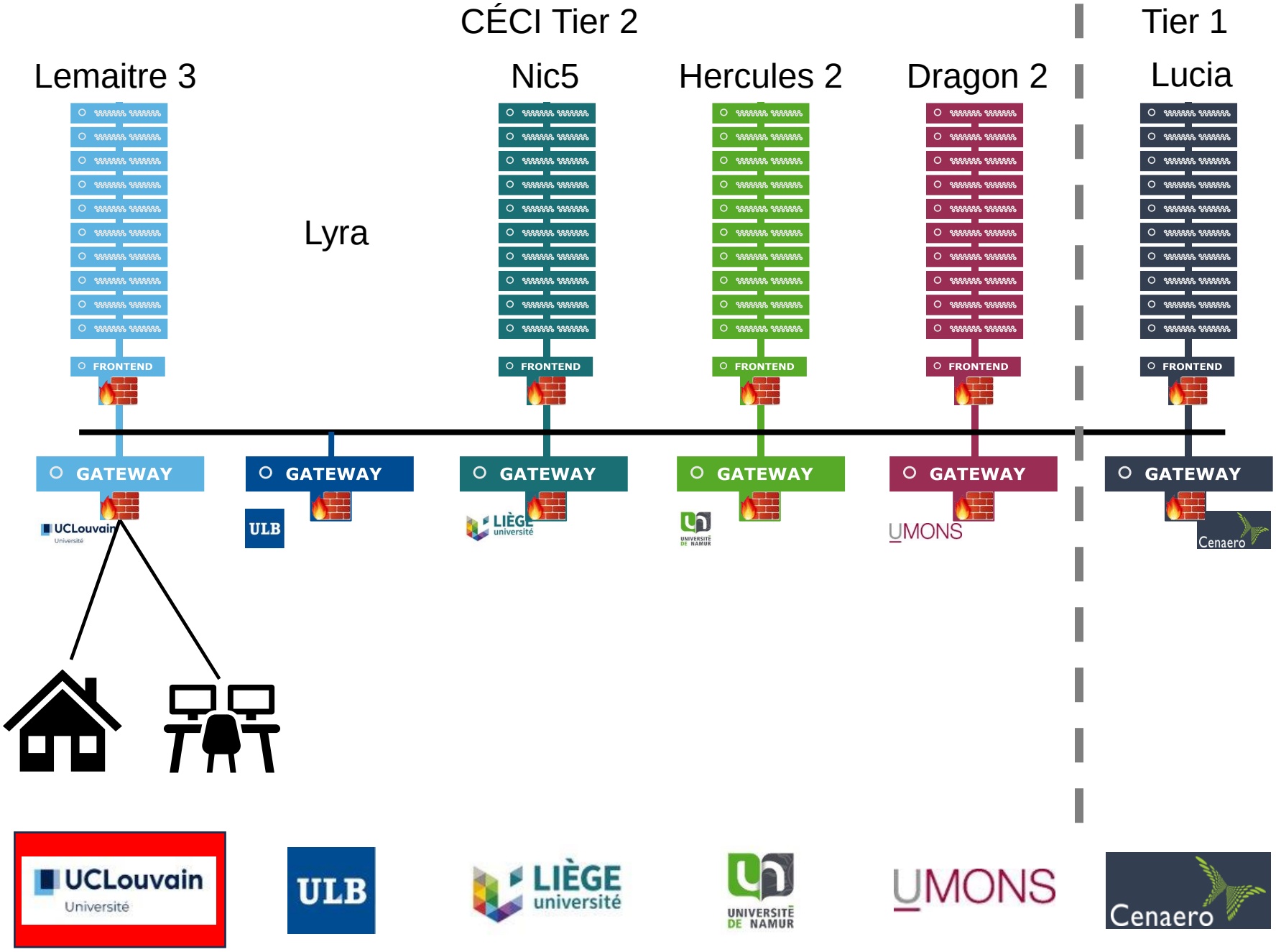
UNIVERSITÉ DE NAMUR

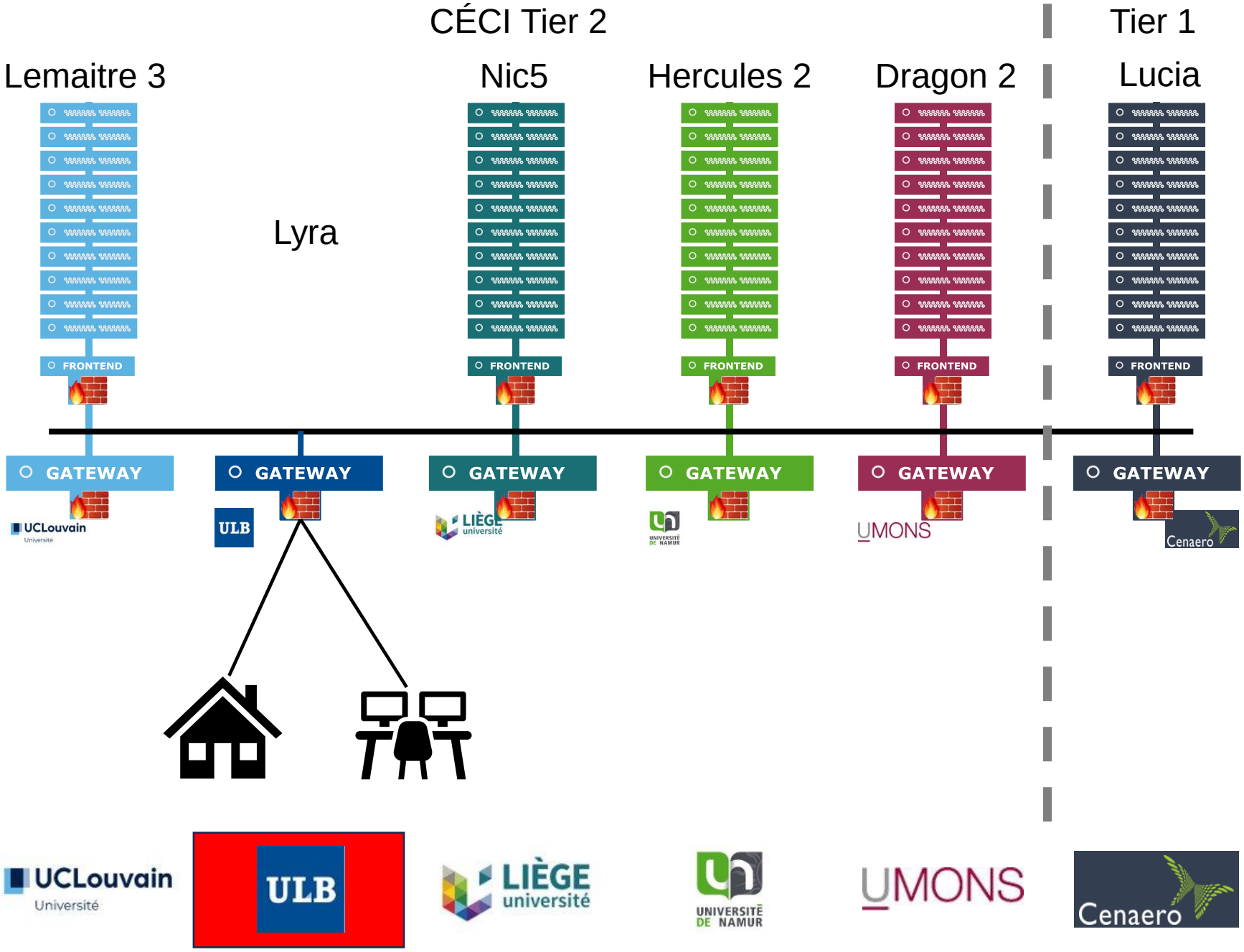
UMONS

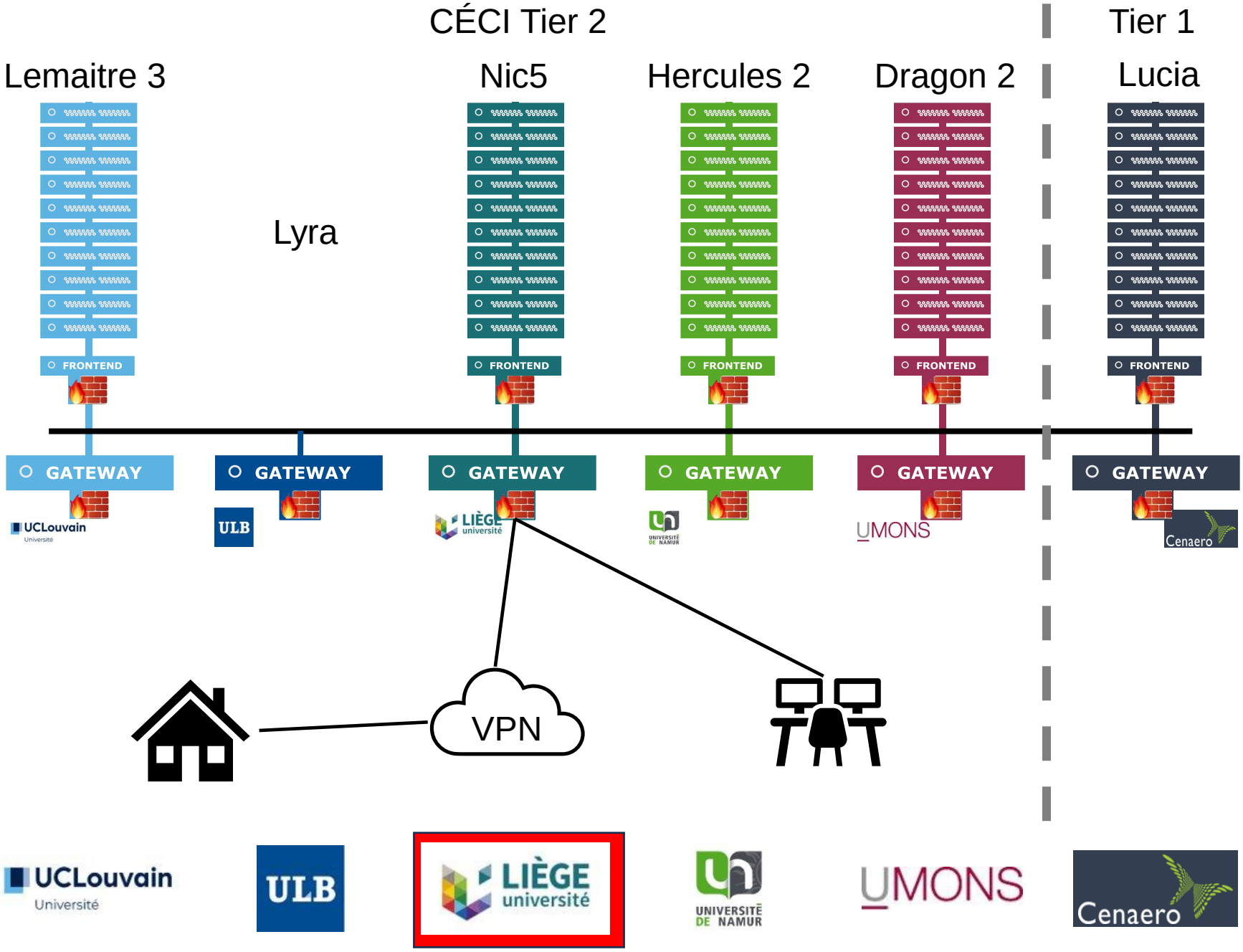
Cenaero

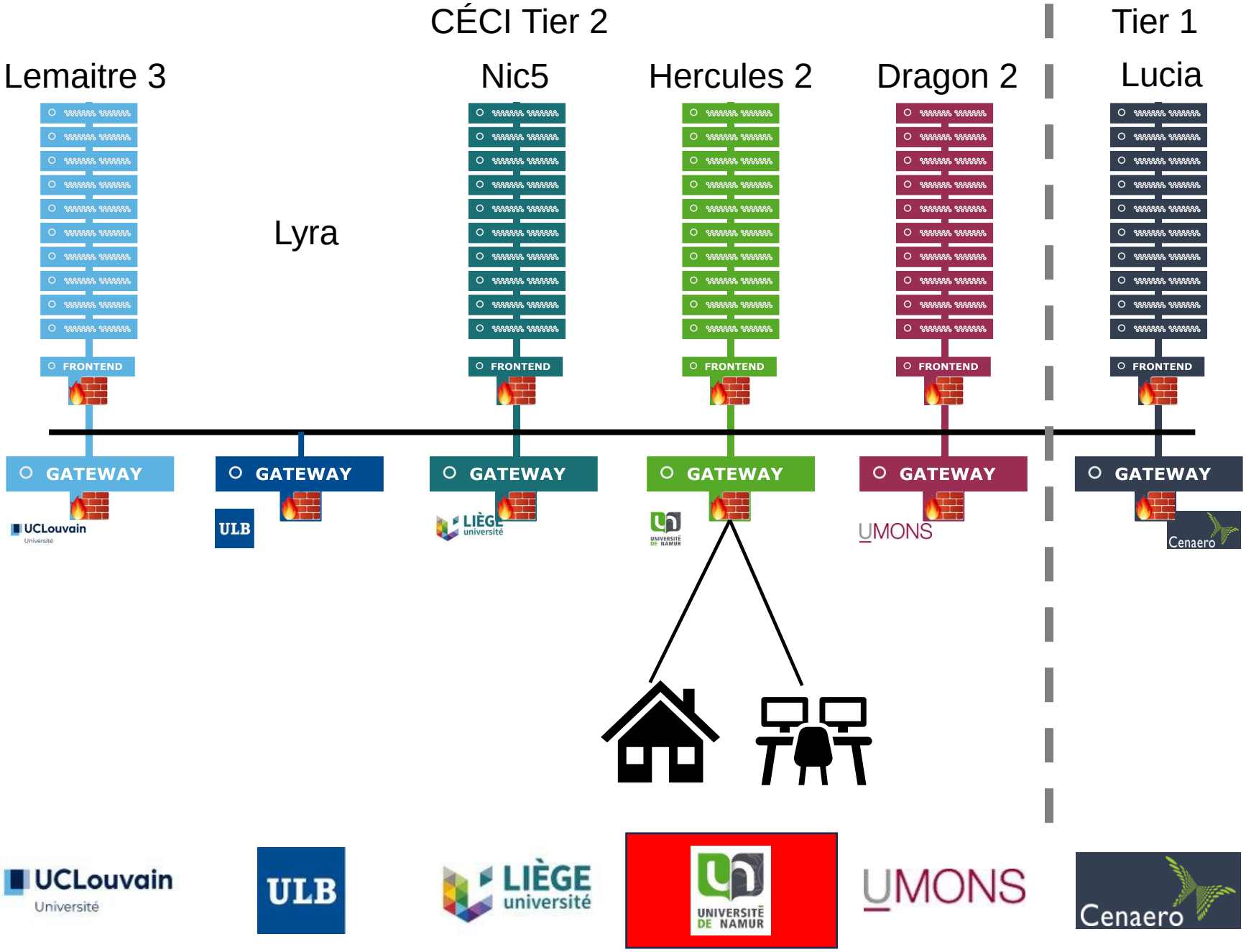


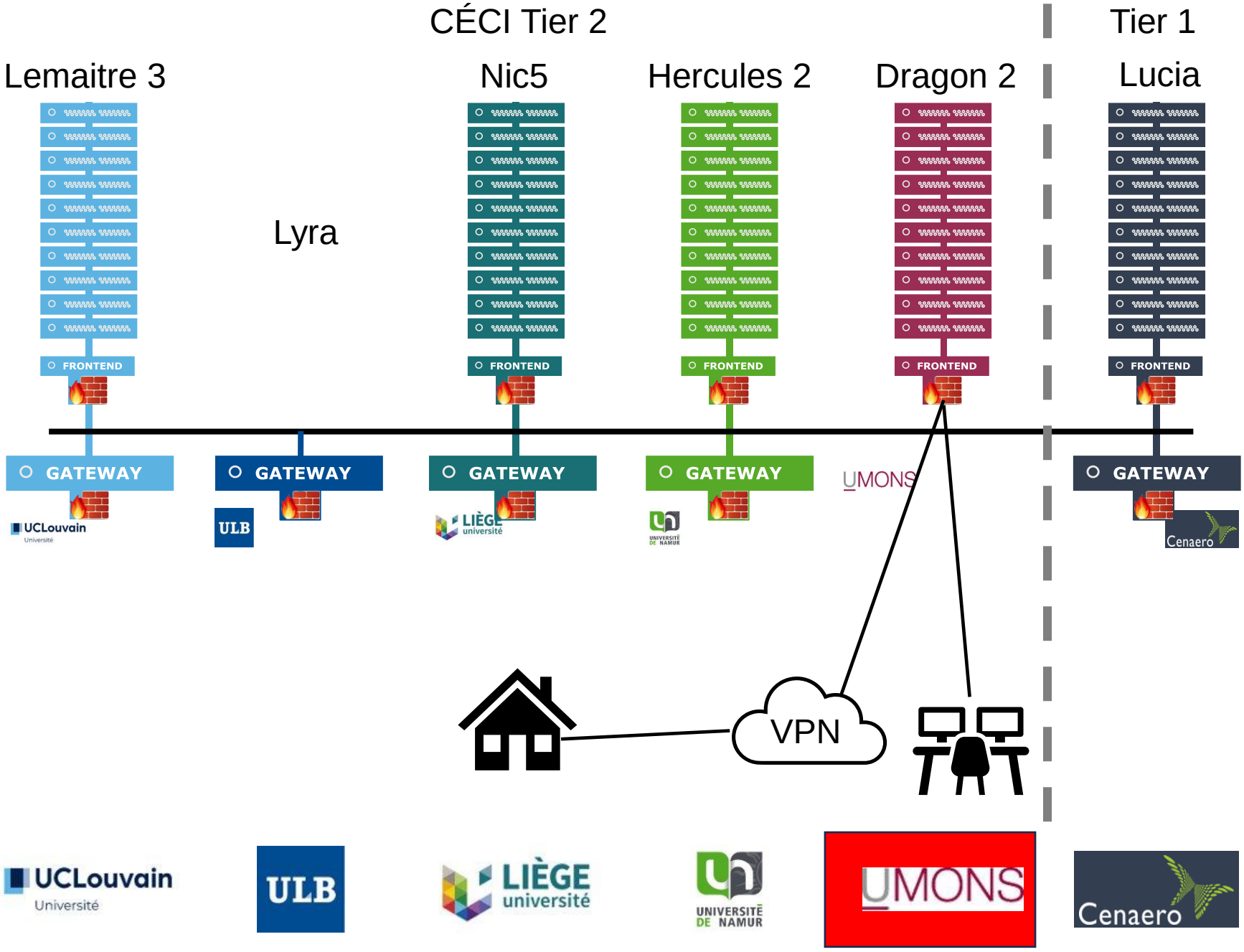
Frontend access only via Gateway
You only need it to connect through it.


CÉCI Tier 2
Tier 1
Lemaitre 3
Nic5
Hercules 2
Dragon 2
Lucia
Lyra
FRONTEND
GATEWAY
UCLouvain
Université
ULB
LIÈGE
université
UNIVERSITÉ
DE NAMUR
UMONS
Cenaero


CÉCI Tier 2
Tier 1
Lemaitre 3
Nic5
Hercules 2
Dragon 2
Lucia
Lyra
FRONTEND
GATEWAY
UCLouvain
Université
ULB
LIÈGE
université
UNIVERSITÉ
DE NAMUR
UMONS
Cenaero


CÉCI Tier 2
Tier 1
Lemaitre 3
Nic5
Hercules 2
Dragon 2
Lucia
Lyra
FRONTEND
GATEWAY
UCLouvain
ULB
LIÈGE université
UMONS
Cenaero
VPN


CÉCI Tier 2
Tier 1
Lemaitre 3
Nic5
Hercules 2
Dragon 2
Lucia
Lyra
FRONTEND
GATEWAY
UCLouvain
Université
ULB
LIÈGE
université
UMONS
Cenaero


CÉCI Tier 2
Tier 1
Lemaitre 3
Nic5
Hercules 2
Dragon 2
Lucia
Lyra
FRONTEND
GATEWAY
UCLouvain
Université
ULB
LIÈGE
université
UMONS
Cenaero
VPN

CONTEXT

- Clusters Frontend address:
  - lemaitre3.cism.ucl.ac.be
  - nic5.uliege.be
  - hercules.ptci.unamur.be
  - dragon2.umons.ac.be
- Gateways addresses
  - gwceci.cism.ucl.ac.be
  - gwceci.ulb.ac.be
  - gwceci.uliege.be
  - gwceci.unamur.be
  - dragon2.umons.ac.be

SSH Tools on Windows

- MobaXterm (only windows)
  - Very easy
  - Connection and file transfer
  - Graphical server
- OpenSSH (also Mac+Linux)
  - Linux like experience/setup
- VSCode (also Mac+Linux)
  - Text editor, connection and file transfer
  - No Graphical server
  - Use OpenSSH
- Putty
  - No file transfer
  - Not easy to support for ssh key

SSH tools on Linux

- OpenSSH
- VScode

Put your private key on your laptop (MacOs/Linux/WSL)

- Save your key id_rsa.ceci file from your e-mail to your home directory
- Open a terminal
- Create the .ssh directory if it does not exist and set permissions

```
mkdir ~/.ssh
chmod 700 ~/.ssh
```

- Move your key to this directory

```
mv id_rsa.ceci  ~/.ssh/.
```

Put your private key on your laptop (MacOs/Lunix/WSL)

- Change the permissions of the identity file so that only you can read it

```
chmod 600 ~/.ssh/id_rsa.ceci
```

- Check the permissions. The follow command :

```
ls -l ~/.ssh/id_rsa.ceci
-rw------- 1 user user 1743 oct 18 06:48 .ssh/id_rsa.ceci
ls -ld .ssh
drwx------ 2 user user 4096 oct 18 06:45 .ssh
```

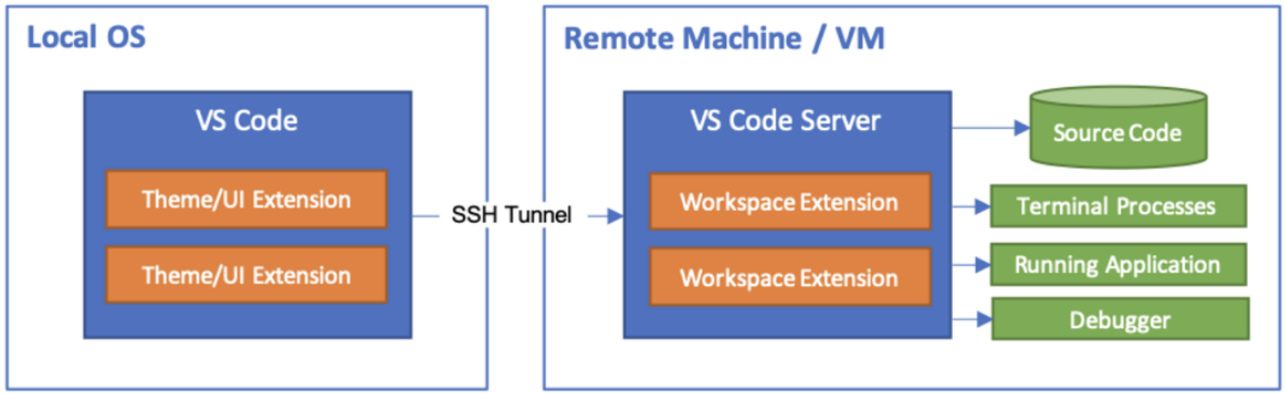
Must output -rw------ and drwx------ permissions

- Create the public key

```
ssh-keygen -y -f ~/.ssh/id_rsa.ceci > ~/.ssh/id_rsa.ceci.pub
```

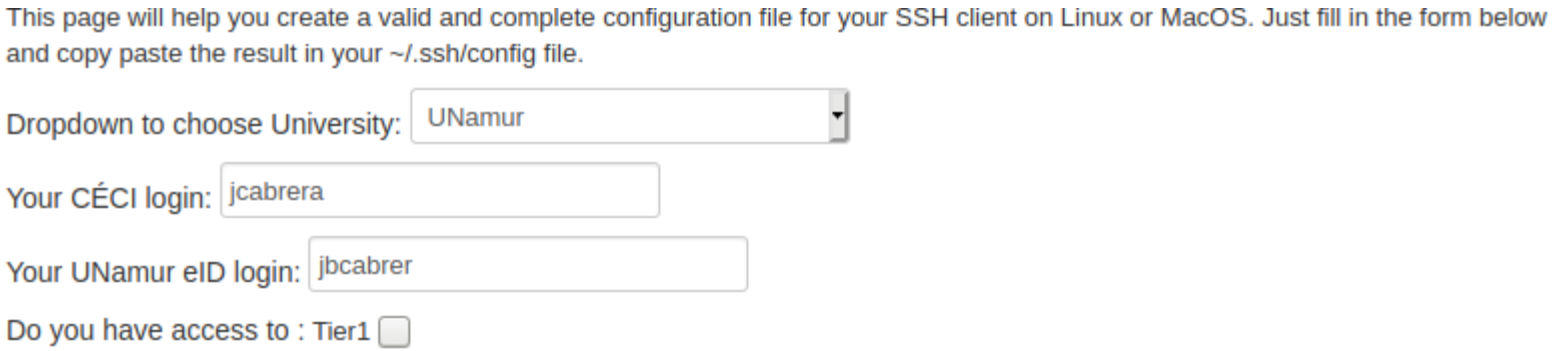
Visual Studio Code

- Install VSC
  - ➡ https://code.visualstudio.com/download
- add ssh extension:
  - ➡ https://code.visualstudio.com/docs/remote/ssh
- See step by step pdf on indico page



Creating your configuration file

- Go to the CÉCI wizard
  http://www.ceci-hpc.be/sshconfig.html
- Chose your university.
- Set your CÉCI and gateway login name.
- Depending on your university, the number of inputs fields will change.
- Tick the field "tier 1" if you have access to Lucia.
  If you are not sure, leave it unchecked.



Creating your configuration file

Copy and paste the result in the **.ssh/config** file

```
# University Gateway ---------------------------------------------------------
Host gwceci
    Hostname hal.unamur.be
    User jbcabrer
    IdentityFile ~/.ssh/id_rsa.ceci

# CÉCI clusters --------------------------------------------------------------
Host lemaitre3 hercules nic5 dragon1 dragon2
    User jcabrera
    ForwardX11 yes
    IdentityFile ~/.ssh/id_rsa.ceci
    ProxyJump gwceci

Host lemaitre3
    Hostname lemaitre3.cism.ucl.ac.be
Host hercules
    Hostname hercules.ptci.unamur.be
Host dragon1
    Hostname dragon1.umons.ac.be
Host dragon2
    Hostname dragon2.umons.ac.be
Host nic5
    Hostname login-nic5.segi.ulg.ac.be
```

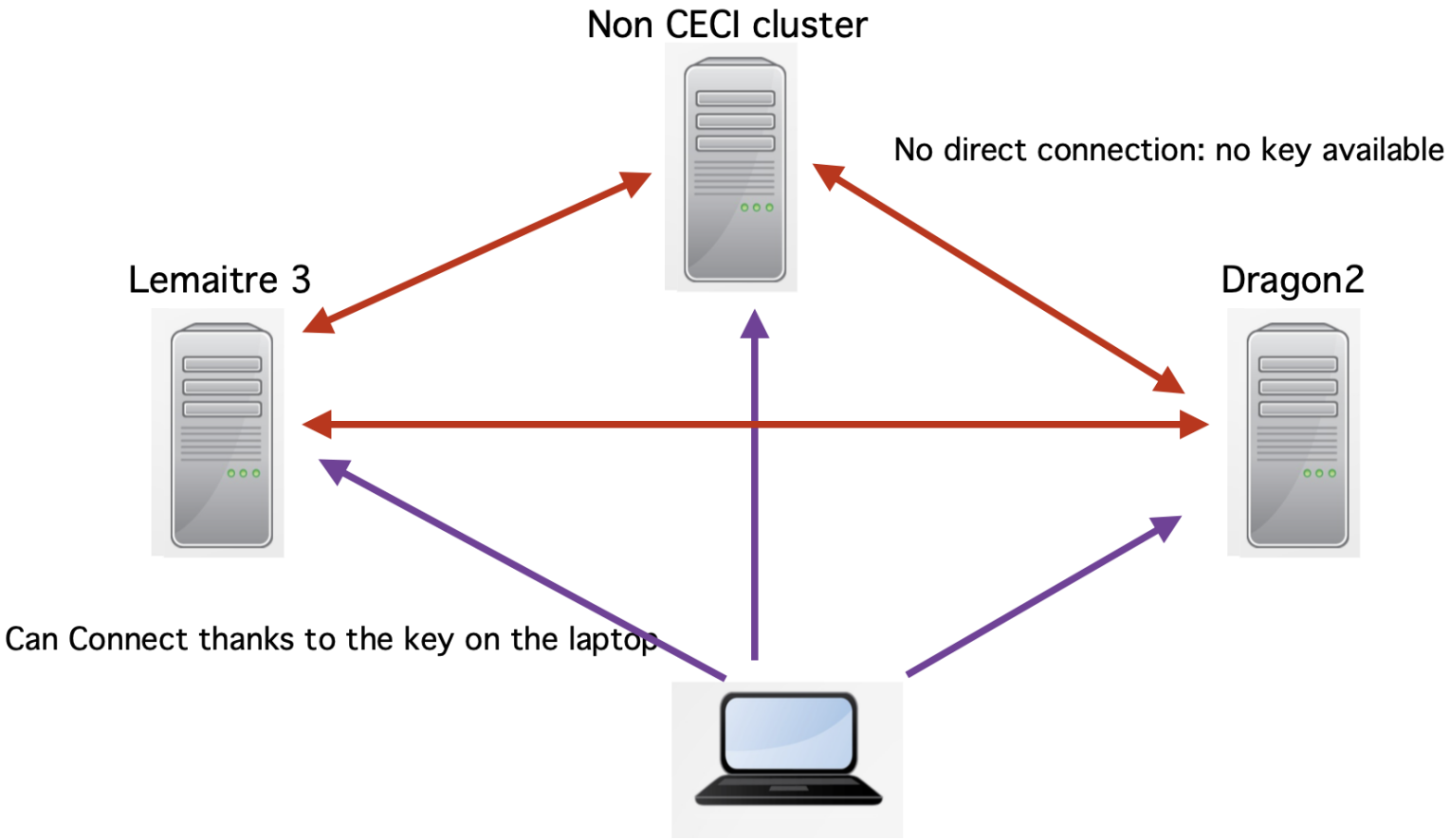
Your gateway server

Common properties to all frontend

Available fronted hosts

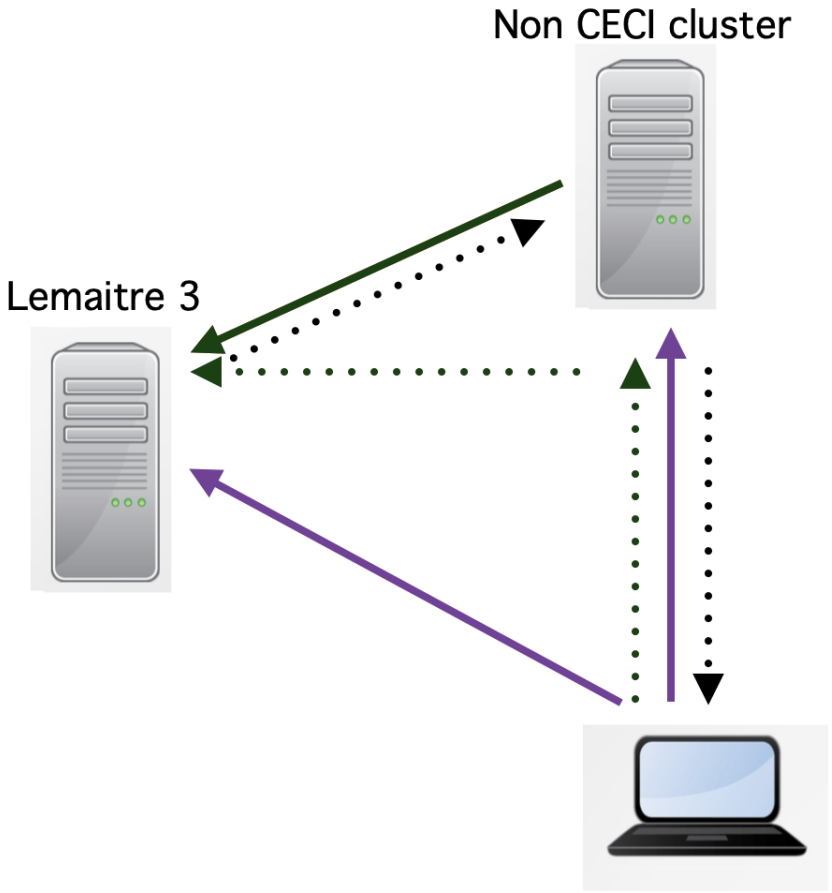
Avoid to propagate your private keys

- Less keys means more security



Avoid to propagate your private keys

- Forward agent send back the ssh request for a key to your laptop



Try to connect

Host ask for a key

Message forward to laptop

Key provided

Connection granted

Agent and Passphrase managers

Use an SSH agent which will remember the passphrase so you do not have to type it in each time you issue the SSH command. Most of the time an ssh-agent starts automatically at login if a password managing software is installed :
Mac OS Keychain, KDE KWallet, Gnome Keyring (Seahorse), etc.

Gnome Keyring loads all private keys in ~/.ssh **which have the corresponding public key**.

In MacOS add in ~/.ssh/config

```
Host *
   UseKeychain yes
   AddKeysToAgent yes
```

Agent and Passphrase managers

Make sure you have an agent running

```
ssh-add -l
Could not open a connection to your authentication agent.
```

```
ssh-add -l
The agent has no identities.
```

If you get "Could not open a connection to your authentication agent." start an agent with

```
eval $(ssh-agent)
```

If you get "The agent has no identities." The agent is already running. Add your key. Your key is decrypted and stored in memory

```
ssh-add ~/.ssh/id_rsa.ceci
Enter passphrase for /home/user/.ssh/id_rsa.ceci:
Identity added: /home/user/.ssh/id_rsa.ceci (/home/user/.ssh/id_rsa.ceci)
```

check the loaded key

```
ssh-add -l
2048 20:6c:8c:cd:e8:e6:9b:4f:8c:9c:d6:8a:eb:37:6d:17 /home/user/.ssh/id_rsa.ceci (RSA)
```

How to connect on mobaxterm

- For windows user only
- Live demo
- Dedicated set of PDF on indico for you to follow step by step
- Demo also available on YouTube:
  - https://youtu.be/o41r0mFaURU

Frequent mistakes

The permissions on your key file are not correct

- **Error**: bad permissions

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@         WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0644 for '/home/user/.ssh/id_rsa.ceci' are too open.
It is recommended that your private key files are NOT accessible by others.
This private key will be ignored.
bad permissions: ignore key: /home/user/.ssh/id_rsa.ceci
user@frontend's password:
it means that Permissions 0644 for '/home/user/.ssh/id_rsa.ceci' are too open.
Change them to 600 as explained in the first section of this document.
```

- **Problem:** Permissions 0644 for '/home/user/.ssh/id_rsa.ceci' are too open.
- **Solution**: Change them to 600 as explained previously

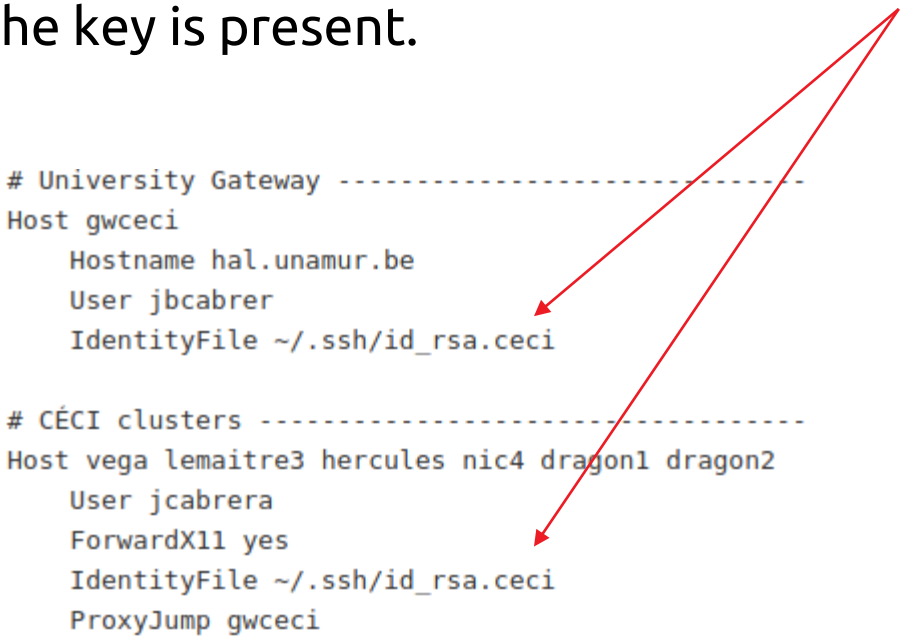
```
chmod 600 ~/.ssh/id_rsa.ceci
```

You did not specify the correct path to your SSH key

- **Error**: you are being asked for a password directly

```
ssh frontend
user@frontend's password:
```

- **Problem**: your SSH client did not use the SSH key.
- **Solution**: Make sure that your .ssh/config is properly configured and the key is present.



```
# University Gateway ------------------------------
Host gwceci
    Hostname hal.unamur.be
    User jbcabrer
    IdentityFile ~/.ssh/id_rsa.ceci

# CÉCI clusters -----------------------------------
Host vega lemaitre3 hercules nic4 dragon1 dragon2
    User jcabrera
    ForwardX11 yes
    IdentityFile ~/.ssh/id_rsa.ceci
    ProxyJump gwceci
```

You used a wrong username or tried to connect before your keys are synchronized

- **Error**: you are being asked for a passphrase, then a password

```
ssh frontend
Enter passphrase for key '/home/user/.ssh/id_rsa.ceci':
user@frontend's password:
```

- **Problem**: the username you are using is not the correct one or you are trying to connect with the new private key while it has not been synchronized to the cluster yet.
- **Solution**: Verify your username or wait ~30 min

```
# University Gateway ------------------------------
Host gwceci
    Hostname hal.unamur.be
    User jbcabrer
    IdentityFile ~/.ssh/id_rsa.ceci

# CÉCI clusters -----------------------------------
Host vega lemaitre3 hercules nic4 dragon1 dragon2
    User jcabrera
    ForwardX11 yes
    IdentityFile ~/.ssh/id_rsa.ceci
    ProxyJump gwceci
```

Troubleshooting

You can use -v, -vv or -vvv to troubleshooting a session

```
ssh frontend -v
OpenSSH_7.6p1 Ubuntu-4ubuntu0.5, OpenSSL 1.0.2n  7 Dec 2017
debug1: Reading configuration data /home/user/.ssh/config
debug1: /home/user/.ssh/config line 4: Applying options for *
debug1: /home/user/.ssh/config line 126: Applying options for hercules
…
debug1: SSH2_MSG_KEXINIT sent
debug1: SSH2_MSG_KEXINIT received
…
debug1: Server host key: ssh-rsa SHA256:GfUSNZEFZg28WRCaxJvDNSCCIhrX1IujNIky29ui7IY
debug1: Host 'gwceci' is known and matches the RSA host key.
debug1: Found key in /home/user/.ssh/known_hosts:33
…
debug1: Offering public key: RSA SHA256:IMDnFOL/9DI4otUnSUJBMxLc0v3jXSHkGUsM4ogi5Us /home/user/.ssh/id_rsa.ceci
debug1: Server accepts key: pkalg rsa-sha2-512 blen 277
debug1: Authentication succeeded (publickey).
Authenticated to gwceci ([YYY.YYY.YYY.YYY]:22).
...
debug1: Server host key: ecdsa-sha2-nistp256 SHA256:SyLaaBe7CuO7Dpa6vJa0vbAUxnYSpl30xaJo5yBF//c
debug1: Host 'frontend' is known and matches the ECDSA host key.
debug1: Found key in /home/user/.ssh/known_hosts:217
…
debug1: Offering public key: RSA SHA256:IMDnFOL/9DI4otUnSUJBMxLc0v3jXSHkGUsM4ogi5Us /home/user/.ssh/id_rsa.ceci
debug1: Server accepts key: pkalg rsa-sha2-512 blen 277
debug1: Authentication succeeded (publickey).
Authenticated to frontend (via proxy).
...
```

Exercise: First connexion

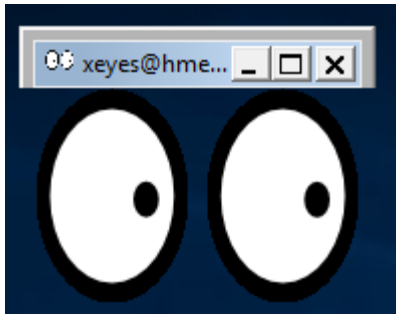
Connect to a cluster with the command

```
ssh frontend
```

where **frontend** is one of the frontend names defined in the configuration file.

The option **ForwarX11** in your configuration file allows you to open a remote window. For this, on **MacOs > 10.7** users need to install xquartz (needs reboot)

Try in **lemaitre3** the command xeyes



Exercise: First connexion

Example

```
ssh user@frontend
The authenticity of host frontend (XXX.XXX.X.XXX)' can't be established.
RSA key fingerprint is 06:54:39:a0:5c:b5:56:b3:29:9e:96:67:a0:4a:c1:ff.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'hmem.cism.ucl.ac.be' (RSA) to the list of known hosts.
```

The FIRST TIME you connect to a frontend, you will be asked to accept the Public Key
Check the key fingerprint from CÉCI web site
http://www.ceci-hpc.be/clusters.html

Exercise: First connexion

Example

```
ssh user@frontend
The authenticity of host frontend (XXX.XXX.X.XXX)' can't be established.
RSA key fingerprint is 06:54:39:a0:5c:b5:56:b3:29:9e:96:67:a0:4a:c1:ff.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'hmem.cism.ucl.ac.be' (RSA) to the list of known hosts.
Enter passphrase for key '/home/user/.ssh/id_rsa.ceci':
```

Now, the frontend public key is stored in your **know_host** file

Enter the **passphrase** you set when you create the account
This will decrypt your private key

Text Editor Option

- Text editor on the cluster
  - Non graphical: Emacs, vi
    - Tutorial on vi, next week
    - Graphical one: gedit , ...
    - Nano (this afternoon)
- Graphical interface running on your laptop
  - Visual Studio Code
  - Mount the file-system

SSH-based file transfer (SCP, rsync, SSHFS)

scp

You can copy files/directories back and forth between computers

Create a temporary directory with dummy files on your computer

```
mkdir -p cours_ssh/scp_test; touch cours_ssh/scp_test/file{1..4}.txt
ssh frontend 'mkdir cours_ssh'
```

Copy the directory to your home directory in one of the frontends and check

```
scp -r cours_ssh/scp_test host:cours_ssh/.
ssh frontend 'ls cours_ssh/scp_test/'
```

Copy it back

```
scp -r frontend:cours_ssh/scp_test cours_ssh/scp_test2
```

Copy between frontends is not permitted. Use $CECITRSF partition

For a copy throw your computer use -3 option

```
scp -r -3 frontend1:cours_ssh/scp_test frontend2:cours_ssh/.
```

rsync

rsync is widely used for backups and mirroring and as an improved copy command for everyday use

Most common usage is to synchronize files with archive option ‘a’, and compress option ‘z’. If you want to get a copy of your hard work you did in the frontend to your laptop:

```
ssh frontend 'mkdir cours_ssh/rsync_test; touch cours_ssh/rsync_test/file{1..4}.txt'
rsync -avz --progress frontend:cours_ssh/rsync_test cours_ssh/.
```

Modify a file at the frontend and synchronize

```
ssh frontend 'echo "Adding hello1 word in $(hostname)" >> coursssh/rsynctest/file4.txt'
rsync -avz --progress frontend:coursssh/rsynctest coursssh/.
```

Modify a file in your computer and prevent Overwrite when synchronize -u

```
echo 'Adding hello in client' > cours_ssh/rsync_test/file3.txt
rsync -avzu --progress frontend:cours_ssh/rsync_test cours_ssh/.
```

Delete a file at the frontend and force delete it in your computer.

```
ssh host rm cours_ssh/rsync_test/file1.txt
rsync -avz --del --progress frontend:cours_ssh/rsync_test cours_ssh/.
```

sshfs

Use SSHFS to mount a remote file system - accessible via SSH

Linux install:

Debian, Ubuntu

```
sudo apt-get install sshfs
```

Fedora/CentOs

```
sudo yum install sshfs
```

MacOS Install:

Install FUSE and SSHFS from https://osxfuse.github.io/

sshfs

Example: Mount your CECIHOME

Create on your computer a repository to mount the CÉCI home

```
mkdir ceci_home
```

Mount the remote CÉCI Home on your computer

```
cluster=frontend
sshfs -o uid=`id -u` -o gid=`id -g` $cluster:$(ssh $cluster 'echo $CECIHOME')/ ceci_home
```

Create a file in the mounted directory

```
echo 'file content' > ceci_home/file_fuse.txt
```

Check the file content in the frontend

```
ssh frontend 'cat $CECIHOME/file_fuse.txt'
```

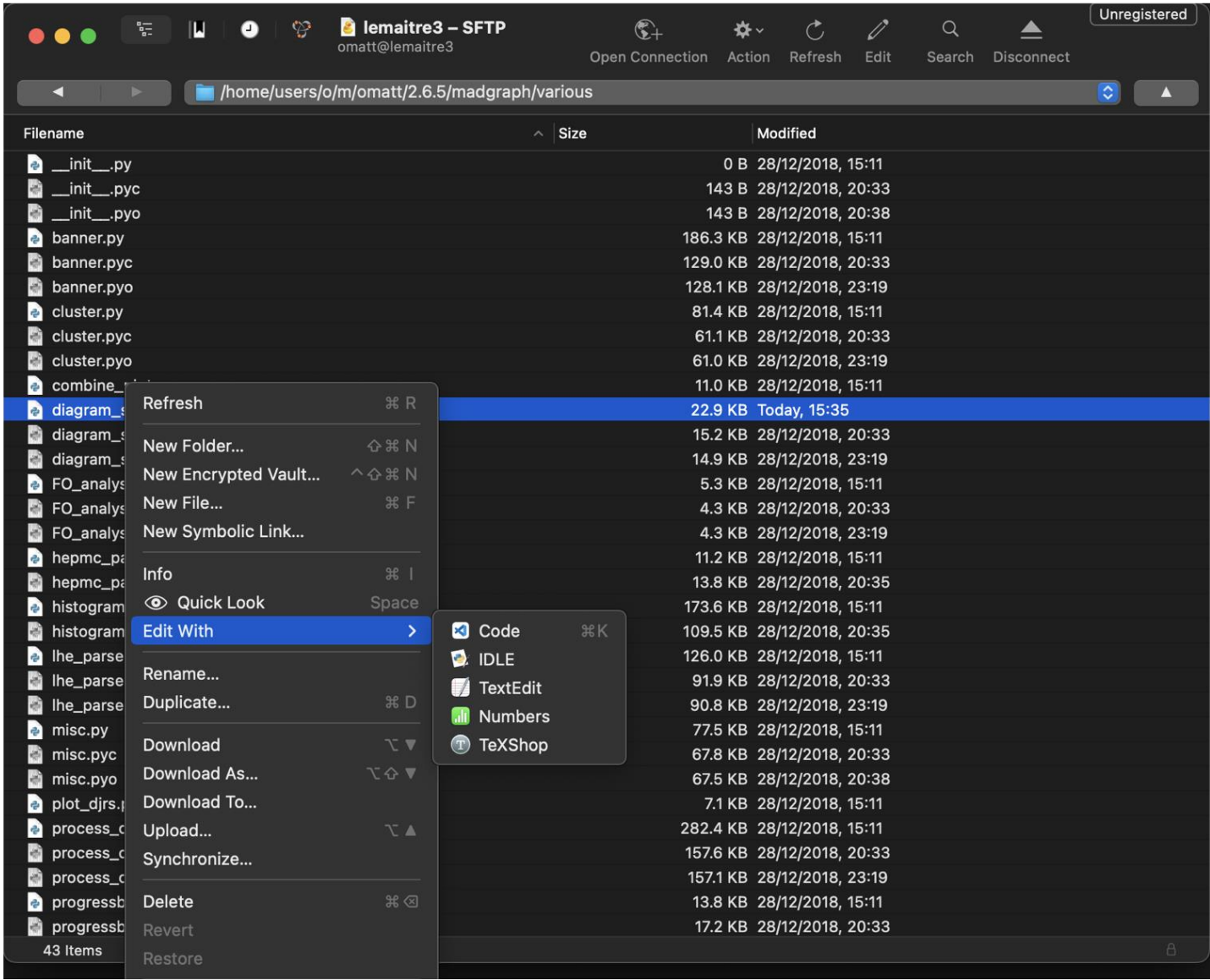
disconnect

```
fusermount -u ceci_home
```

Cyberduck (graphical filesystem)

FTP and text edition

Drag and Drop are working
Rename/ remove/... as well



Conclusion

- The most Important point is to be able to connect
- The frontend is not a place to run production
- Simplify your live with an agent to avoid to type your password all the time
- Keep your private key in safe place a do not share it
- With great power comes great responsibility

References

- OpenSSH Manual Pages
- RSA Cryptography Specifications Version 2.2
- The Secure Shell (SSH) Transport Layer Protocol