HPC Training Series

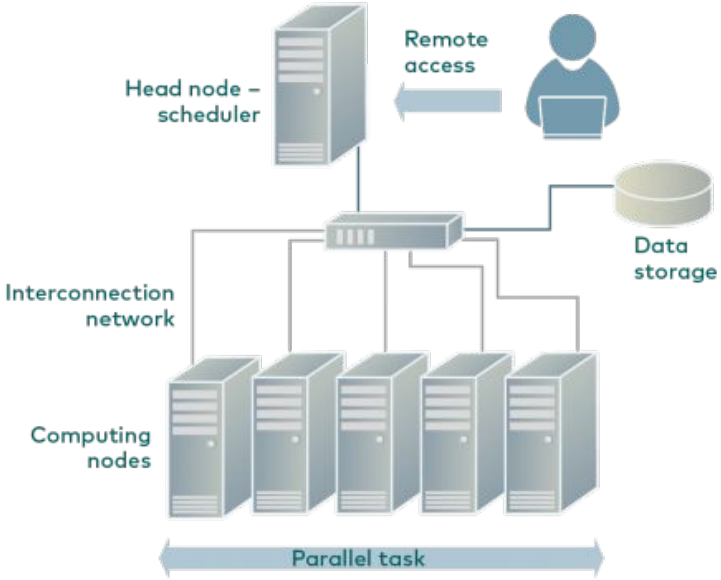
Course 10 "Introduction to Computational Fluid Dynamics and OpenFOAM, using High Performance Computing"

A virtual HPC environment for familiarization with the SLURM job submission system

Nikos Triantafyllis (GRNET)
ntriantafyl@admin.grnet.gr

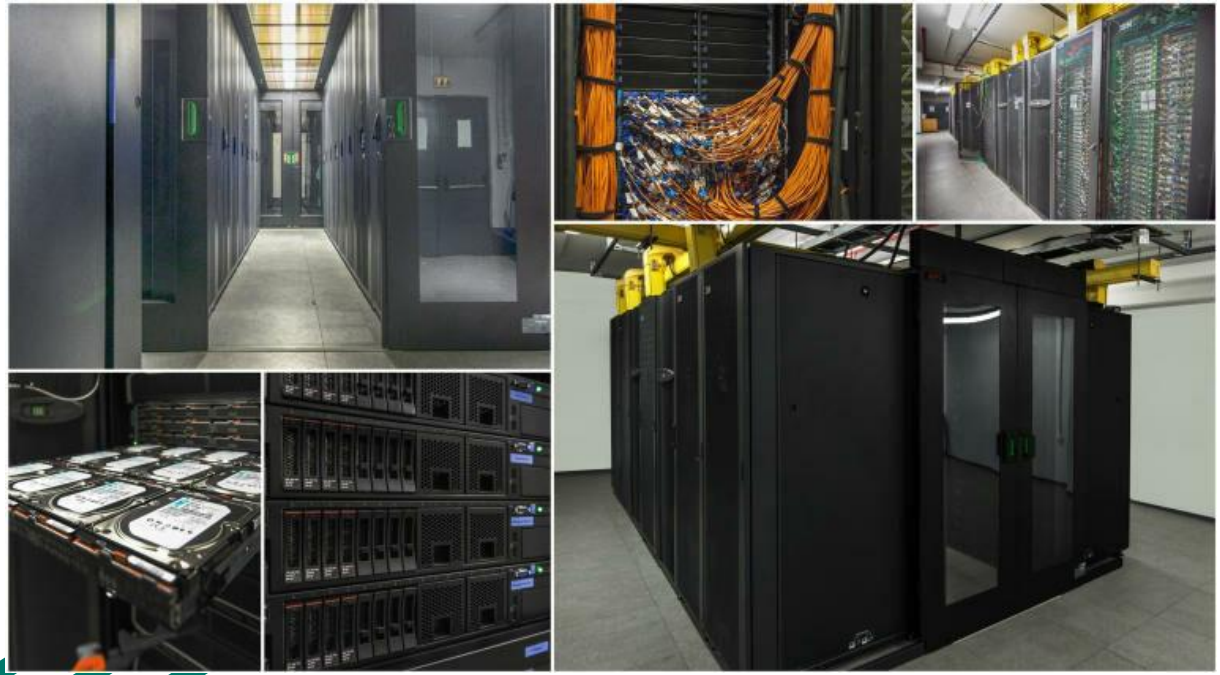
What is HPC?

- High-Performance Computing (HPC) is the ability to perform sophisticated calculations at high speeds.
- An HPC cluster consists of hundreds or thousands of compute servers, so-called nodes. The nodes in each cluster work in parallel with each other.
- HPC solves large problems in science, engineering, or business, that are too complex for a PC. On typical PC it might take e.g. hours, days, weeks to perform the computations, but if you use an HPC Cluster, it might only take minutes, hours, days, respectively.



EURO
Greece

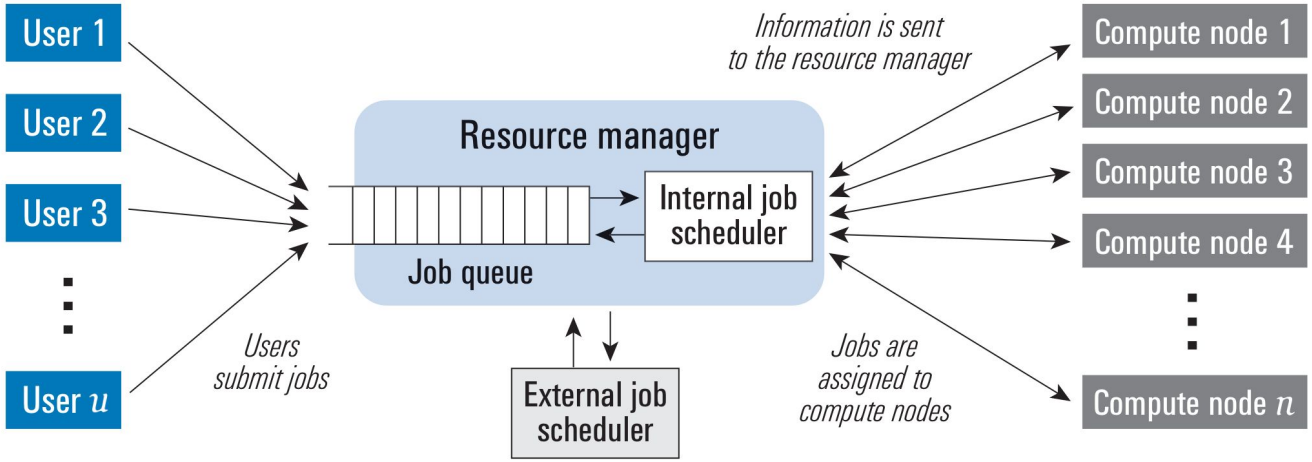
GRNET ARIS HPC Cluster



More info https://www.hpc.grnet.gr

EURO
Greece

SLURM - Resource and Job Management System

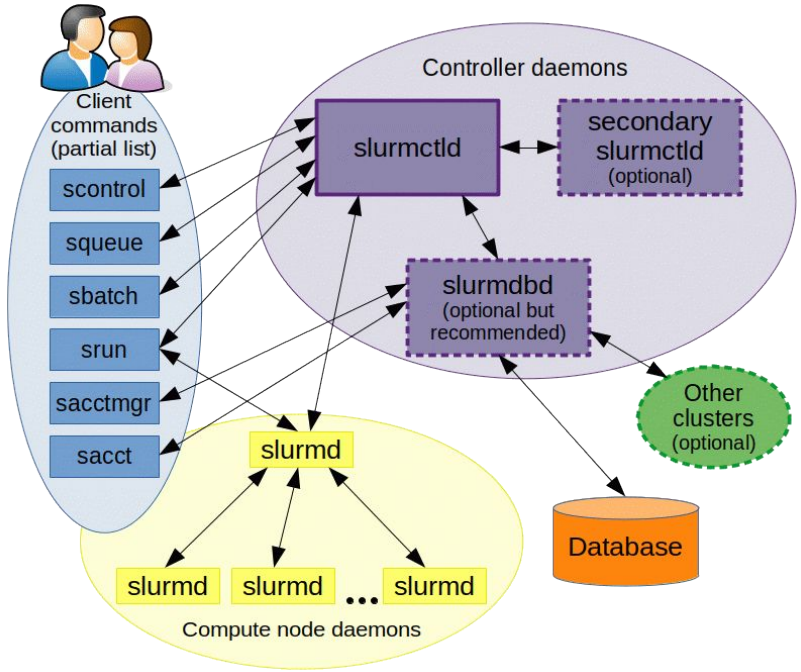


SLURM: software stack that runs on HPC infrastructure and operates resource management, job scheduling and accounting

EURO Greece

Typical HPC/SLURM infrastructure

- User executes SLURM client commands such as job submissions (sbatch) [Blue area]
- SLURM handles the received jobs and orchestrates operations [Purple area]
- SLURM passes user's jobs to compute nodes [Yellow area]
- User receives job's results back to their working dir



EURO Greece

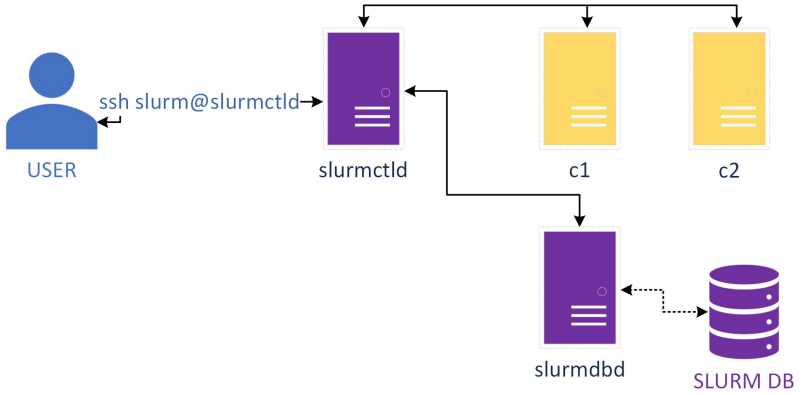
Tutorial

1. In this tutorial you will deploy a typical HPC infrastructure using the SLURM resource manager under containers
2. Submit a simple MPI program, where each process prints a “Hello world” message
3. View example’s output

EURO
Greece

Current HPC/SLURM infrastructure

- 5 containers:
  - 1 MySQL Server instance to store SLURM accounting
  - 1 node as the DB controller
  - 1 login node as the SLURM controller and user's login endpoint
  - 2 compute nodes for calculations
- Each compute node contains 1 CPU of 4 cores
- All 4 nodes operate Debian-based Linux OS



EURO
Greece

Prerequisites

For Windows users

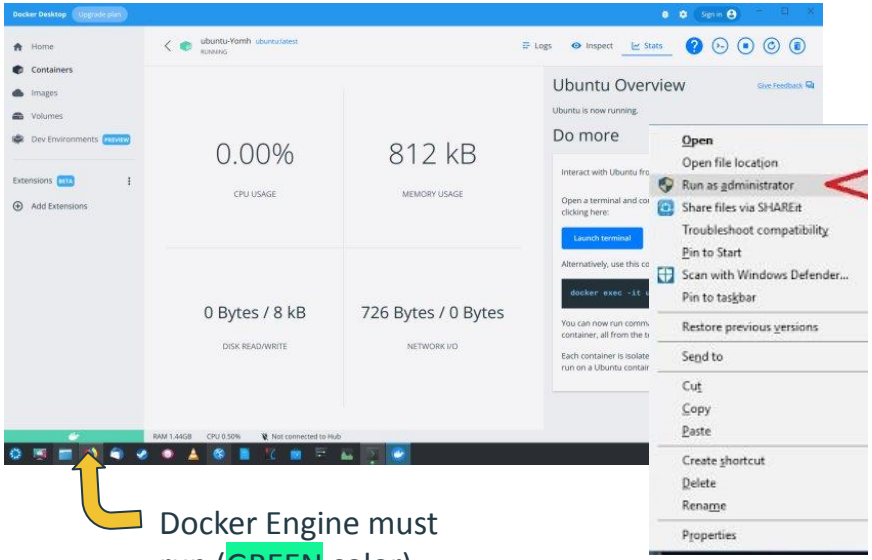
- Download **Docker Desktop** from: https://docs.docker.com/desktop/install/windows-install/
- Follow step-by-step instructions here: https://www.linkedin.com/pulse/step-guide-how-install-docker-windows-1011-shashank-abhishek/
- Download gnuplot: https://sourceforge.net/projects/gnuplot/files/gnuplot/6.0.1/
- Download paraview: https://www.paraview.org/download/

EURO
Greece

Prerequisites

For Windows users

- Use **default** options in installation
- Your PC must be **restarted**
- If docker engine does not start, you might need to close the Docker Desktop and run it in **administration** mode



Docker Engine must run (GREEN color)

EURO Greece

Steps A-Z

For Windows users

1. Make sure that Docker Desktop is **initiated** (GREEN color)
2. **Download** the Docker recipe to setup the virtual infrastructure of SLURM under containers: https://github.com/nikosT/slurm-docker-cluster/archive/refs/heads/openfoam-pull.zip
3. **Extract** content at some folder e.g. C:\...\slurm-docker-cluster-openfoam-pull
4. Open Windows **PowerShell** (in search button type PowerShell)
5. In Windows PowerShell **terminal** type:

   `cd C:\...\slurm-docker-cluster-openfoam-pull`

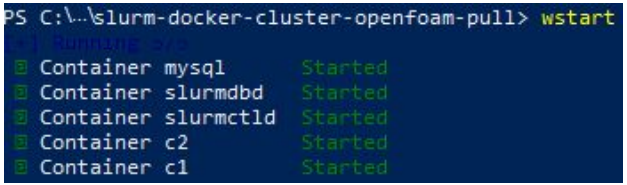
   `powershell -ExecutionPolicy Bypass`

   `. .\alias.ps1` *# load environment*

   `wstart` *# start the virtual cluster (~2.5 GB images' size)*

   When `wstart` is **completed**, you should view this
6. Then, type:

   `ssh slurm@slurmctld` *# access the login node*
7. `cd mpi_hello` *# change dir to the MPI example*
8. `sbatch test.sh` *# submit your first MPI job*
9. `ls` *# view the outputs of your submission*
10. `exit` *# logout from login node*
11. `wstop` *# stop the virtual cluster*



EURO Greece

Steps A-Z

For Linux users

- In terminal type:

```
sudo apt-get install git docker docker.io docker-compose docker-compose-v2 # install docker
sudo apt-get install gnuplot paraview # install visualization s/w
git clone -b openfoam-pull https://github.com/nikosT/slurm-docker-cluster # get docker recipe
cd slurm-docker-cluster # change dir to the appropriate one
chmod -R 777 slurm #set appropriate permissions to the folder
source alias # load environment
wstart # start the virtual cluster (~2.5 GB images' size)
ssh slurm@slurmctld # login control node
cd mpi_hello # change dir to the MPI example
sbatch test.sh # submit your first MPI job
ls # view the outputs of your submission
exit # logout from login node
wstop # stop the virtual cluster
```

EURO Greece

If error occurs

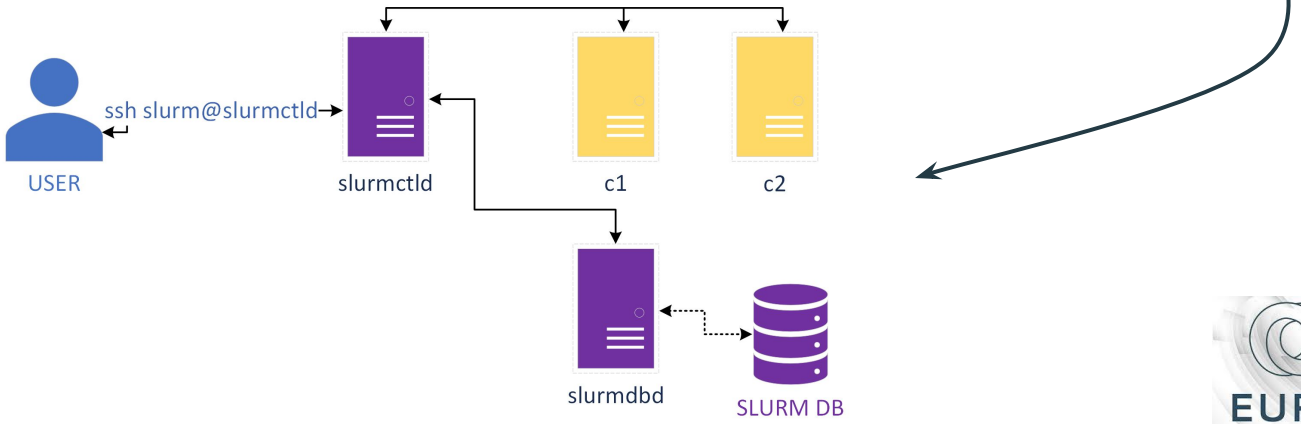
- If an error like this occurs, it is likely due to a previous deployment:

  slurmdbd | ---> Starting the MUNGE Authentication service (munged) …
  slurmdbd | munged: Error: Failed to check keyfile "/etc/munge/munge.key": Permission denied

- You need to remove all associated volumes (check with command: docker volume ls)
  - docker volume rm <volume>
    - etc_munge
    - etc_slurm
    - slurm_jobdir
    - var_lib_mysql
    - var_log_slurm

EURO Greece

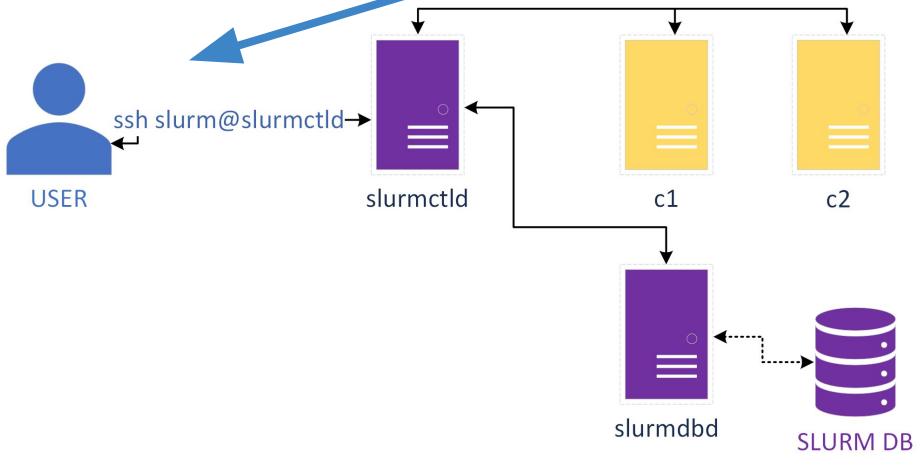
```
Windows PowerShell

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\inter\Downloads\slurm-docker-cluster-openfoam-pull\slurm-docker-cluster-openfoam-pull> . .\alias.ps1
PS C:\Users\inter\Downloads\slurm-docker-cluster-openfoam-pull\slurm-docker-cluster-openfoam-pull> wstart
[+] Running 5/5
 ✔Container mysql      Started                                    0.5s
 ✔Container slurmdbd   Started                                    1.0s
 ✔Container slurmctld  Started                                    1.4s
 ✔Container c1         Started                                    1.9s
 ✔Container c2         Started                                    2.1s
PS C:\Users\inter\Downloads\slurm-docker-cluster-openfoam-pull\slurm-docker-cluster-openfoam-pull> wstatus
NAME        IMAGE                                      COMMAND                  SERVICE     CREATED      STATUS          PORTS
c1          intergallactic/slurm-docker-openfoam:21.08   "/usr/local/bin/dock…"   c1          3 days ago   Up 8 seconds    6818/tcp
c2          intergallactic/slurm-docker-openfoam:21.08   "/usr/local/bin/dock…"   c2          3 days ago   Up 8 seconds    6818/tcp
mysql       mariadb:10.10                              "docker-entrypoint.s…"   mysql       3 days ago   Up 10 seconds   3306/tcp
slurmctld   intergallactic/slurm-docker-openfoam:21.08   "/usr/local/bin/dock…"   slurmctld   3 days ago   Up 9 seconds    6817/tcp
slurmdbd    intergallactic/slurm-docker-openfoam:21.08   "/usr/local/bin/dock…"   slurmdbd    3 days ago   Up 9 seconds    6819/tcp
PS C:\Users\inte  Downloads\slurm-docker-cluster-openfoam-pull\slurm-docker-cluster-openfoam-pull>
PS C:\Users\inter\Downloads\slurm-docker-cluster-openfoam-pull\slurm-docker-cluster-openfoam-pull>
```



EURO
Greece

```
PS C:\Users\inter\Downloads\slurm-docker-cluster-openfoam-pull\slurm-docker-cluster-openfoam-pull> ssh slurm@slurmctld
Connected to HPC facility...
bash-4.4$ ls
dummyCase  mpi_hello
bash-4.4$ cd mpi_hello/
```

Inside login node



EURO Greece

Windows PowerShell

```
bash-4.4$ cat test.sh
#!/bin/bash
#SBATCH --job-name=my_mpi_job        # Job name
#SBATCH --output=my_mpi_job_%j.out  # Output file name (%j expands to jobID)
#SBATCH --error=my_mpi_job_%j.err   # Error file name (%j expands to jobID)
#SBATCH --partition=normal          # Partition name
#SBATCH --nodes=2                    # Number of nodes
#SBATCH --ntasks-per-node=2          # Number of tasks per node
#SBATCH --cpus-per-task=1          # Number of tasks per node
#SBATCH --time=00:10:00              # Time limit (HH:MM:SS)

# needed for docker version
export PSM3_HAL=loopback

# Run MPI application
mpirun -np 4 /home/slurm/mpi_hello/mpi_hello

#srun -n 4 /home/slurm/mpi_hello/mpi_hello
bash-4.4$ sbatch test.sh
Submitted batch job 2
bash-4.4$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
                 2    normal my_mpi_j    slurm  R       0:00      2 c[1-2]
bash-4.4$ ls
mpi_hello  mpi_hello.c  my_mpi_job_1.err  my_mpi_job_1.out  my_mpi_job_2.err  my_mpi_job_2.out  test.sh
bash-4.4$ cat my_mpi_job_1.out
Hello from rank 0 of 4 on c1 (pid: 41)
Hello from rank 1 of 4 on c1 (pid: 42)
Hello from rank 2 of 4 on c2 (pid: 34)
Hello from rank 3 of 4 on c2 (pid: 35)
```

Submit job

Check queue

Job output

EURO Greece

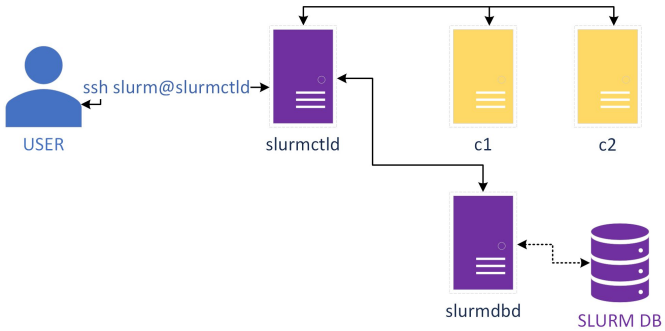
SLURM Useful Commands

- sacct is used to report job accounting information
- sbatch is used to submit a job script for later execution
- scancel is used to cancel a pending or running job
- scontrol is the administrative tool used to view/modify SLURM state
- sinfo reports the state of partitions and nodes managed by SLURM
- squeue reports the state of jobs
- srun usually is executed inside the job script to run apps after job submission

More info https://slurm.schedmd.com

EURO Greece

Exercise



Try accessing resources **interactively** in SLURM:

1. Open 2 terminals, change to the slurm-docker-cluster-openfoam-pull directory and load the environment
2. From both terminals access the login node, then:
3. In Terminal #1, type: `localhost` *(what's the node's name and why?)*
4. In Terminal #1, type: `srun --nodes=1 --time=00:10:00 --pty bash` *(what do you think this command does?)*
5. In Terminal #1, type: `localhost` *(what's the node's name and why?)*
6. In Terminal #2, type: `squeue` *(is there any job running?)*
7. In Terminal #1, type: `exit` *(what happened?)*
8. In Terminal #2, type: `squeue` *(is there any job running?)*
9. In Terminal #1, type: `localhost` *(what's the node's name and why?)*

Try to compile it: `cd mpi_hello; mpicc mpi_hello.c -o mpi_hello`

EURO Greece

EURO Greece

Thanks!

EuroHPC Joint Undertaking

This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 951732. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Germany, Bulgaria, Austria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, Greece, Hungary, Ireland, Italy, Lithuania, Latvia, Poland, Portugal, Romania, Slovenia, Spain, Sweden, United Kingdom, France, Netherlands, Belgium, Luxembourg, Slovakia, Norway, Switzerland, Turkey, Republic of North Macedonia, Iceland, Montenegro