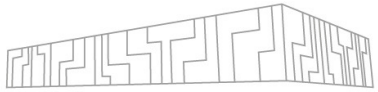
VSB TECHNICAL UNIVERSITY OF OSTRAVA | IT4INNOVATIONS NATIONAL SUPERCOMPUTING CENTER

KAROLINA | SINGULARITY

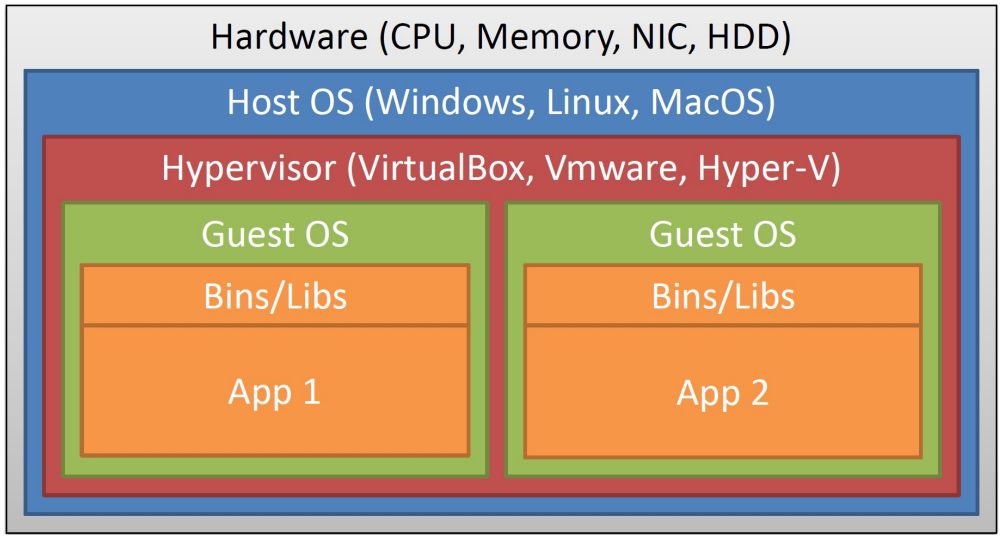
EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research,
Development and Education

MINISTRY OF EDUCATION, YOUTH AND SPORTS

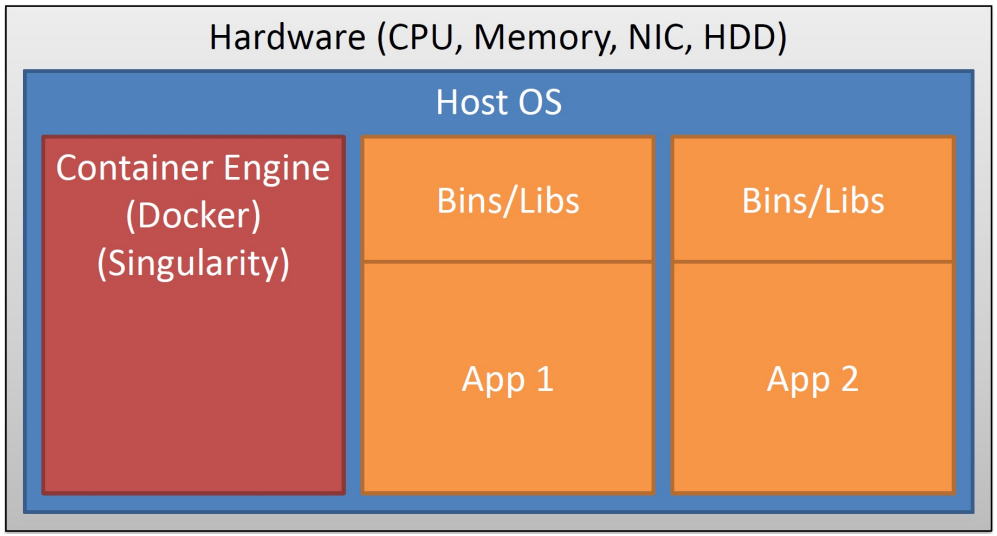
POSSIBILITIES TO RUN OTHER OS ON HPC CLUSTER



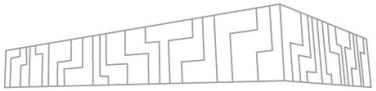
Virtual machines



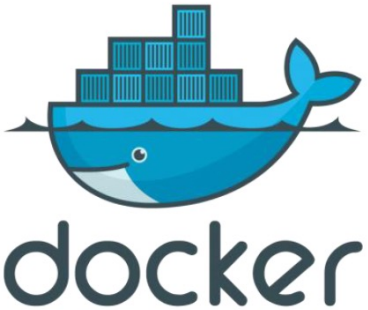
Containers



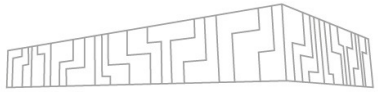
CONTAINERS



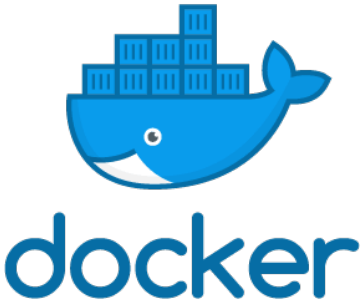
- Docker
  - Most widely used containerization tool
  - Image-based deployment model
  - Allow access to the host's root filesystem
  - User can gain root access to a host's filesystem
- Singularity
  - Permissions inside a container are the same as those outside of the container
  - User can access their files stored outside of the container
  - Directly supports MPI
  - Integration with schedulers, GPU, InfiniBand…
  - More at https://singularity.hpcng.org, https://sylabs.io



CONTAINERS IN HPC

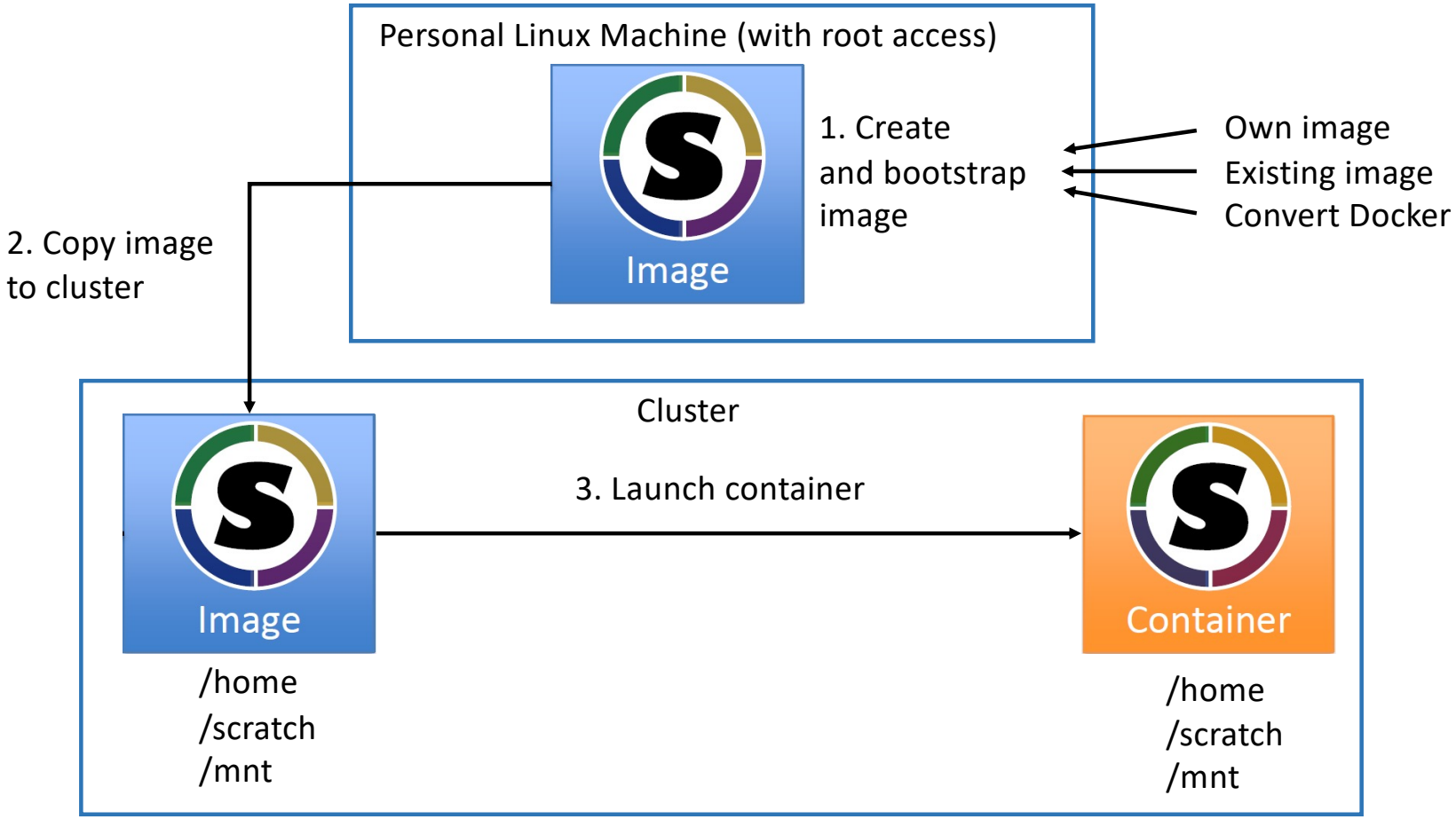


- Can I run Docker on HPC?
- **No** (security reasons).



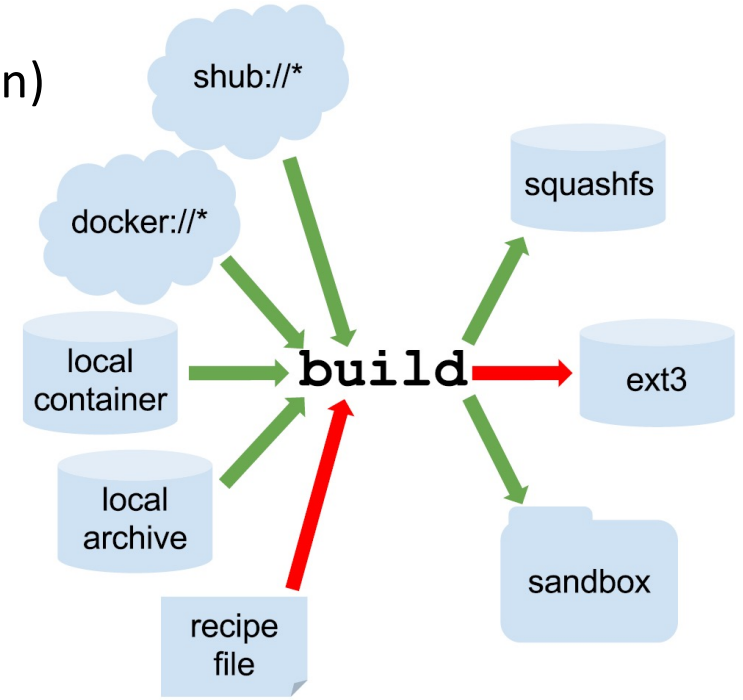
- How can I run container on HPC?
- Use **Singularity.**
- Can Singularity utilize Docker containers? **Yes.**

SINGULARITY WORKFLOW



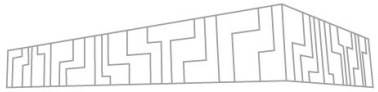
CREATING SINGULARITY IMAGE

- Types of Image
  - Compressed read-only squashfs filesystem image
  - Writable ext3 filesystem image (--writable option)
  - Writable (ch)root directory image (--sandbox option)
- Image can be created from:
  - docker://
  - shub://
  - existing container
  - directory
  - archive
  - bootstrap file




```
# singularity build ubuntu.img docker://ubuntu:latest
```

RUNNING SINGULARITY IMAGE

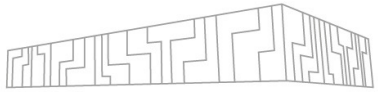


- $ singularity shell <image>
  - Start a container and invoke interactive shell
- $ singularity run <image>
  - Start a container and exec runscript inside container
- $ singularity exec <image> <command>
  - Start a container and exec command inside container
- $ singularity run -app <app_name> <image>
  - Start a container and exec apprun script inside container

GPU SUPPORT (NVIDIA CUDA)

- https://sylabs.io/guides/3.5/user-guide/gpu.html
  - Use the --nv flag to run a CUDA application inside

ACCESSING THE KAROLINA CLUSTER



- Cluster is accessed by the SSH protocol via login nodes at the address karolina.it4i.cz

Login address	Port	Protocol	Login node
karolina.it4i.cz	22	SSH	round-robin DNS record for login[1-4]
login1.karolina.it4i.cz	22	SSH	login1
login2.karolina.it4i.cz	22	SSH	login2
login3.karolina.it4i.cz	22	SSH	login3
login4.karolina.it4i.cz	22	SSH	login4

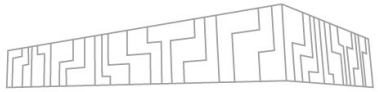
- Example:

```
$ ssh -i /path/to/id_rsa user1234@login1.karolina.it4i.cz
```

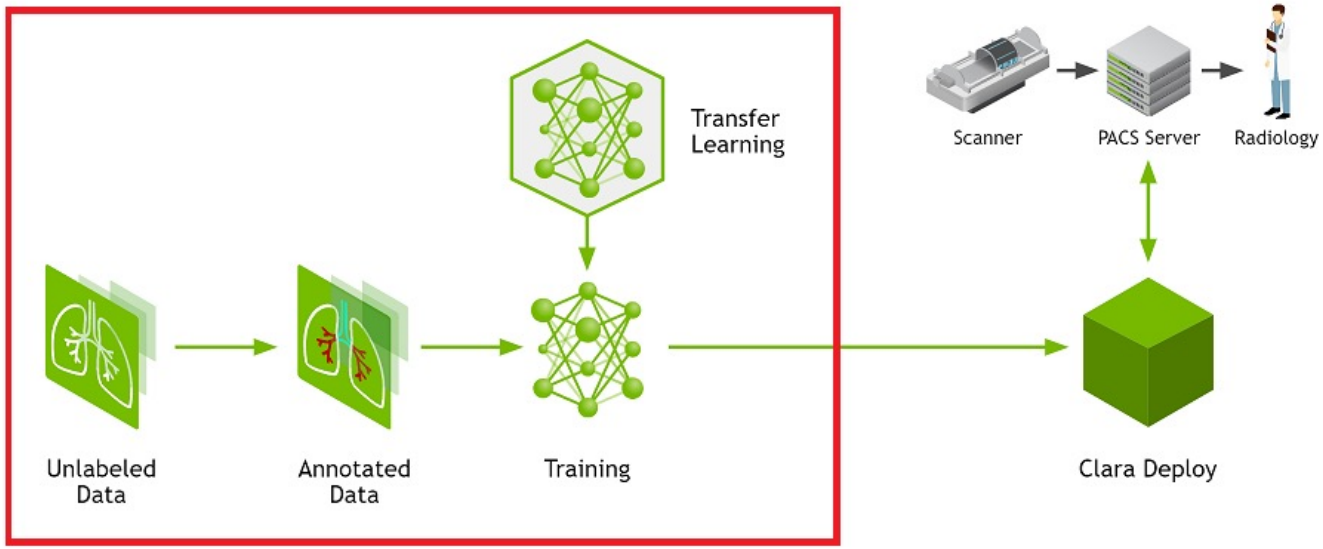
SINGULARITY & KAROLINA

- $ ml Singularity

CLARA TRAIN SDK CONTAINER



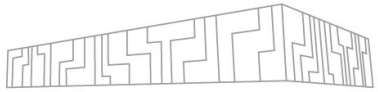
- Clara is an application framework which includes **AI-Assisted Annotation**
- It makes any medical viewer AI capable



- Clara allows to cooperate to researchers, clinicians and data scientists
- Pull command:

```
$ docker pull nvcr.io/nvidia/clara-train-sdk:v3.1.01
```

CLARA SINGULARITY IMAGE



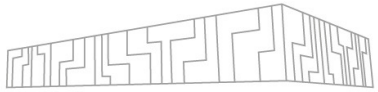
- How to run container using Singularity?
- First, convert Clara-Train **docker image to singularity**

NVIDIA CLARA

- To **build singularity container** use the command below

```
$ singularity build clara-train-sdk.simg docker://nvcr.io/nvidia/clara-train-sdk:v3.1.01
```

CLARA SINGULARITY IMAGE



- Allocation of computational resources

```
[user1234@login1.karolina ~]$ qsub -q qviz -l select=1:ncpus=24 -l walltime=02:00:00 -I -A PROJECT-XX-XX
qsub: waiting for job 477990.isrv1 to start
qsub: job 477990.isrv1 ready
[user1234@vizserv1.karolina ~]$
```

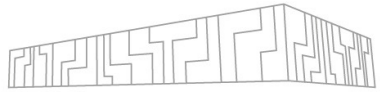
- Executing the image
  - Pre-load Singularity as a module first

```
[user1234@vizserv1.karolina ~]$ ml Singularity
```

  - Run the Singularity with Clara image

```
[user1234@vizserv1.karolina ClaraTrain]$ singularity exec -B /path_to_project/PROJECT-XX-XX/ClaraTrain:/workspace
-B /apps/all/CUDAcore/11.0.2/:/usr/local/cuda --nv clara-train-sdk.simg /bin/bash

MOFED version '5.1-2.5.8' not available in this container.
No matching alternate version found.
Singularity>
```

RUNNING CLARA SERVER



- To start inference server use the following command

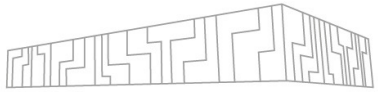
```
Singularity> export TZ="UTC"
```

- Run inference server

```
Singularity> start_aas.sh --workspace /workspace &
```

```
[1] 21616
Singularity> /workspace/aiaa-launch-config.json

ENGINE:: engine=TRITON
TRITON:: Backend is enabled

TRITON:: triton_ip=localhost
TRITON:: Will setup TRITON Server on localhost

TRITON:: triton_http_port=8000
TRITON:: triton_grpc_port=8001
TRITON:: triton_metrics_port=8002
TRITON:: triton_proto=http
TRITON:: triton_shmem=no
TRITON:: triton_model_path=/workspace/triton_models
TRITON:: triton_verbose=false
TRITON:: triton_log=/workspace/logs/0/triton.log
TRITON:: triton_start_timeout=120
TRITON:: triton_model_timeout=30

TRITON is already stopped
TRITON:: Waiting 1 seconds to fully up...
TRITON:: Waiting 2 seconds to fully up...
TRITON:: Waiting 3 seconds to fully up...
TRITON:: Waiting 4 seconds to fully up...
TRITON:: Server started with pid: 21643

AIAA:: aiaa_log_file=/workspace/logs/0/aiaa.log
AIAA:: aiaa_log_dir=/workspace/logs/0
AIAA:: aiaa_workspace=/workspace
AIAA:: aiaa_ssl=false
```

TUNNELING FROM LOGIN NODE

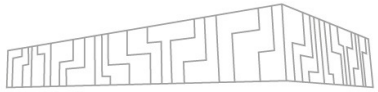


- Create port forwarding from **login node** to **local computer**
  - Established from local computer

```
$ ssh -X -i /path/to/id_rsa -TN -f user1234@login1.karolina.it4i.cz -L 6600:localhost:6002
```

- free_local_port : localhost : free_port_on_login

TUNNELLING FROM COMPUTE NODE



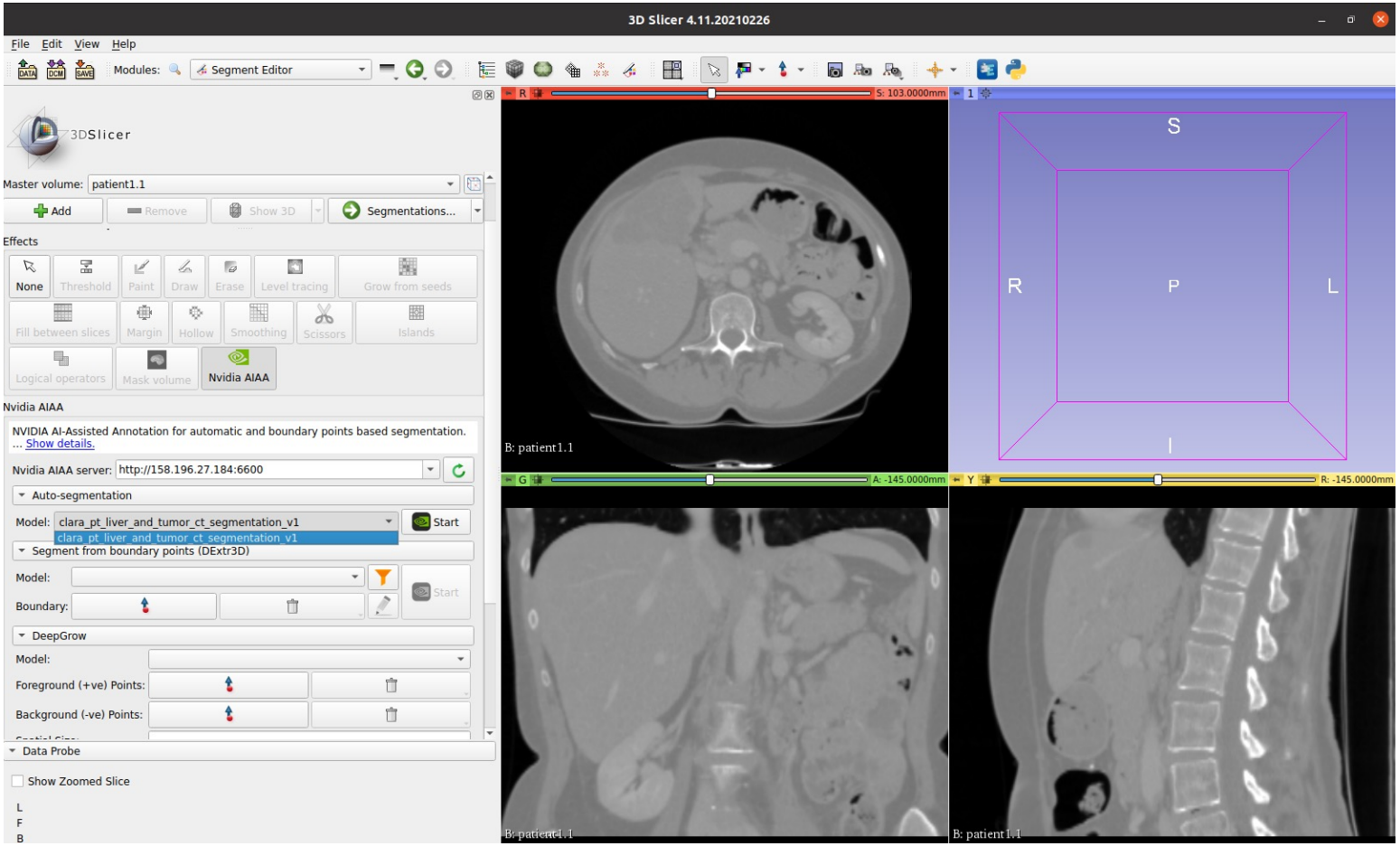
- Create port forwarding from **compute node** to **login node**
  - First, connect to login node
  - Next, use the command below to provide tunnelling

```
[user1234@login1.karolina ~]$ ssh -X -TN -f user1234@vizserv1.karolina.it4i.cz -L 6002:localhost:5000
```

- free_local_port : localhost : port_on_compute_node_where_clara_transmits

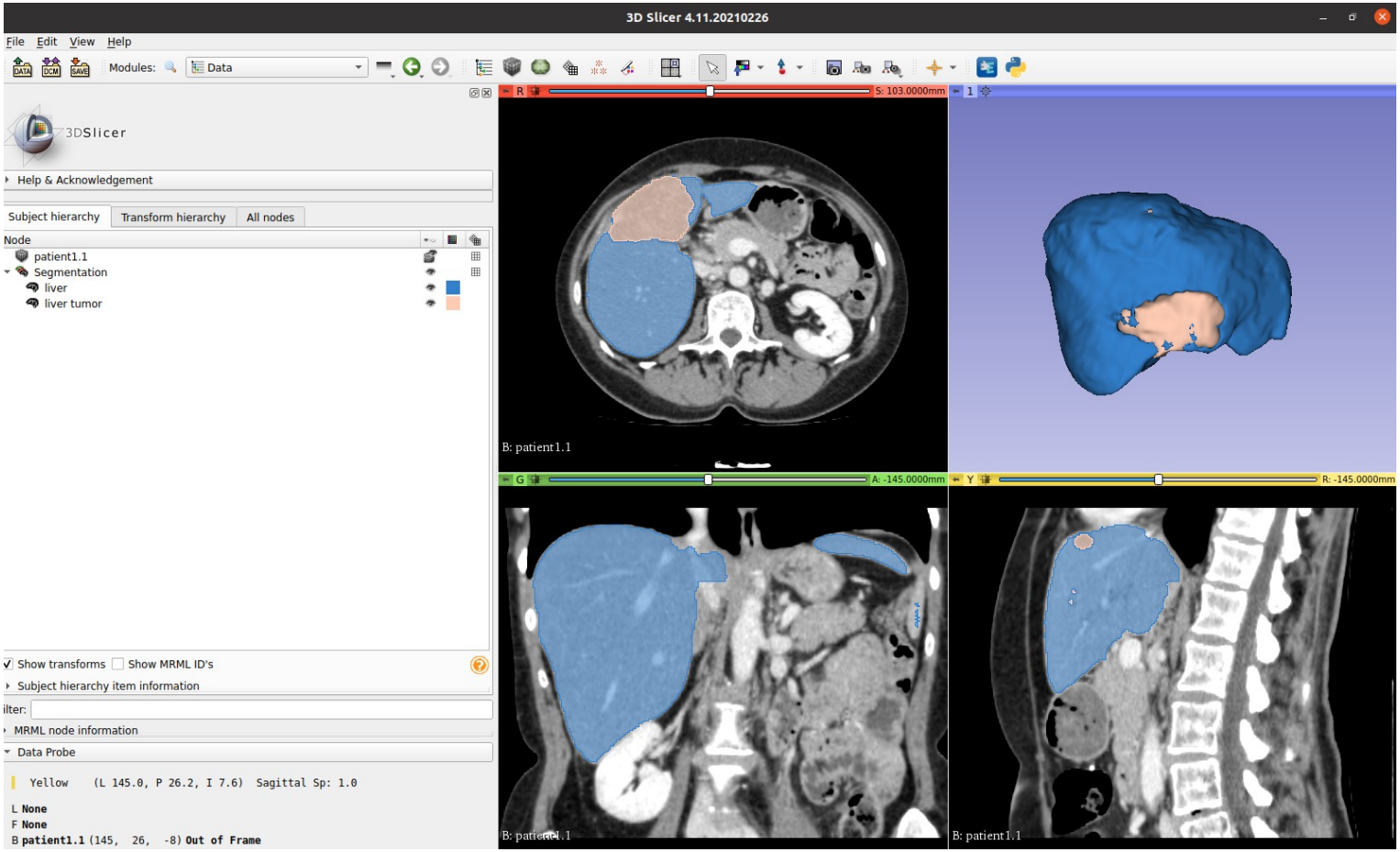
3D SLICER

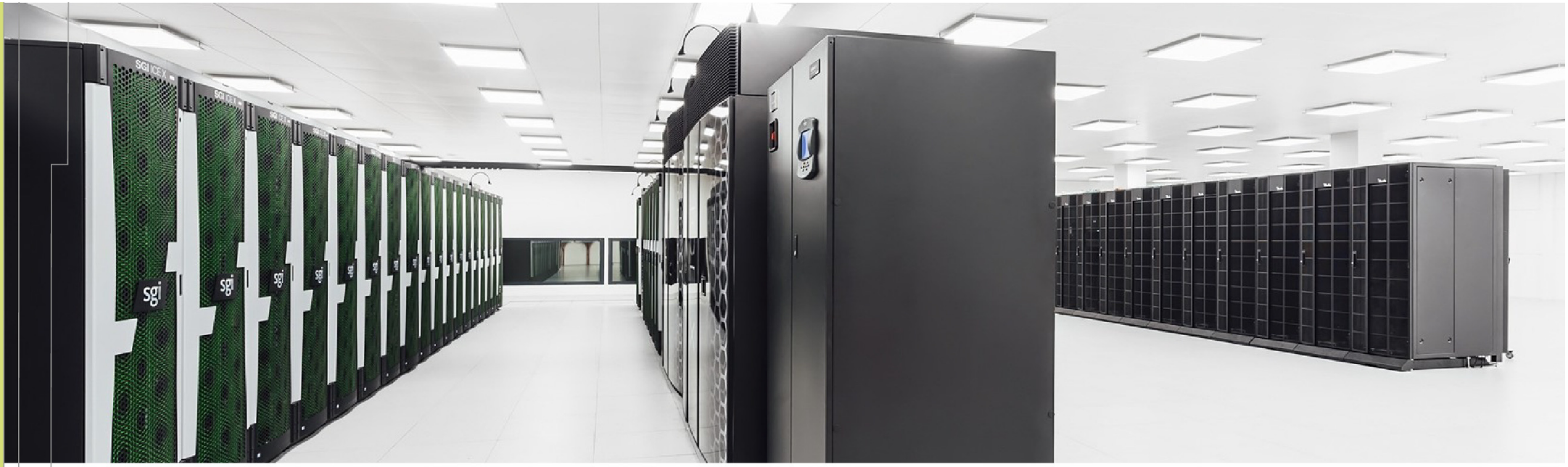
- Medical image processing frontend that utilizes inference capabilities of Clara server as a remote client



3D SLICER

- Medical image processing frontend that utilizes inference capabilities of Clara server as a remote client





Petr Strakoš
petr.strakos@vsb.cz

IT4Innovations National Supercomputing Center
VSB – Technical University of Ostrava
Studentská 6231/1B
708 00 Ostrava-Poruba, Czech Republic
www.it4i.cz

VSB TECHNICAL UNIVERSITY OF OSTRAVA | IT4INNOVATIONS NATIONAL SUPERCOMPUTING CENTER

EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research,
Development and Education

MINISTRY OF EDUCATION, YOUTH AND SPORTS