



Vlaanderen
is supercomputing

VSC Tier-1 info session

2026 cutoff #3

The slides will be shared afterwards



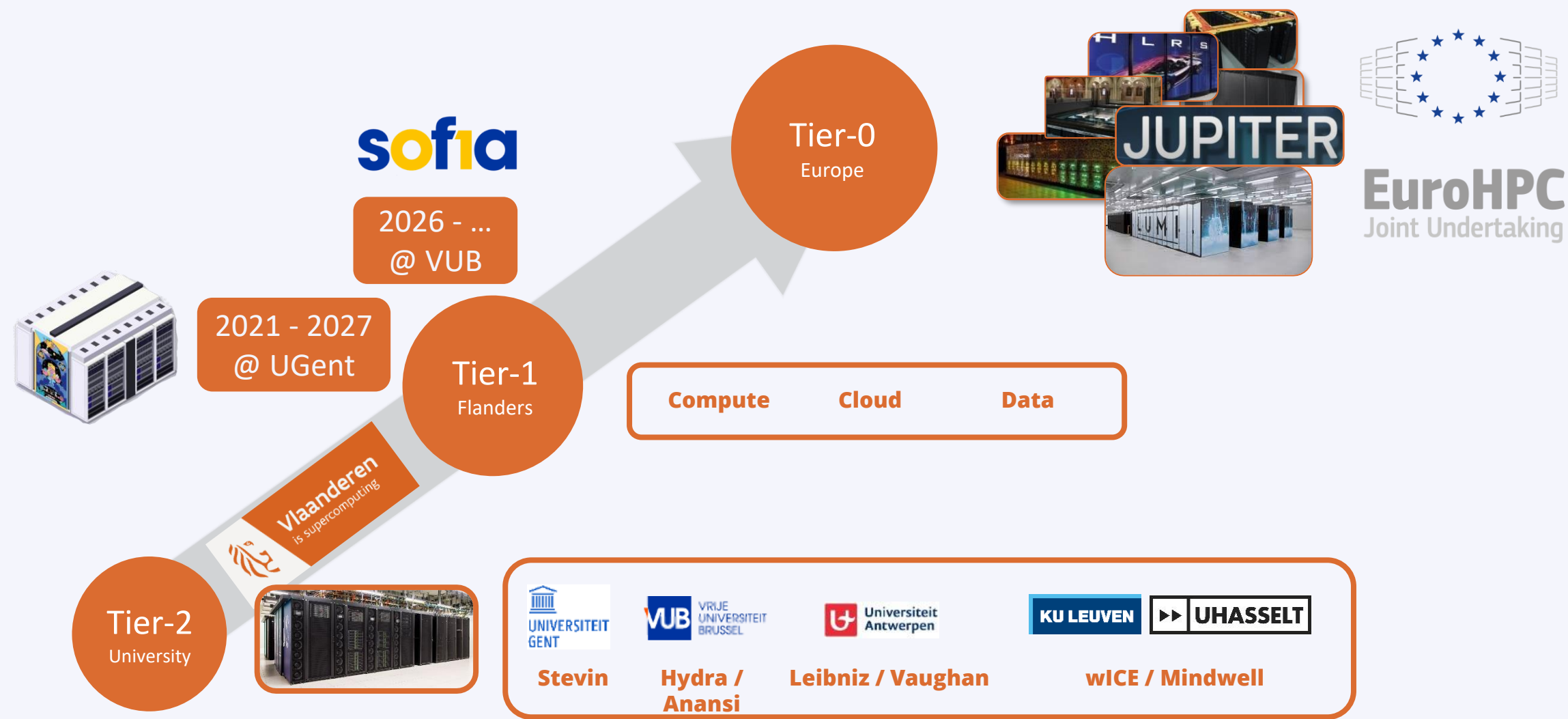
Info session overview

- Overview of VSC Tier-1 services
 - Tier-1 Compute
 - Tier-1 Cloud
 - Tier-1 Data
 - *Bonus slides with additional recommendations*
- 
- Service description & eligibility
- What to watch out for when applying for regular grants

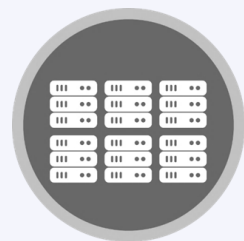
Overview of VSC Tier-1 services



Overview of VSC Tier-1 services



Overview of VSC Tier-1 services



Tier-1 Compute

vscentrum.be/compute

More info: [VscDocumentation](#)



Tier-1 Cloud

vscentrum.be/cloud

More Info: [VscDocumentation](#)



Tier-1 Data

vscentrum.be/data

More info: [VscDocumentation](#)

Starting grant

Starting grant
Regular grant

Starting grant
Regular grant

Regular grant
3 cut-offs per year

Continuous

Collaborative grant



Tier-1 Compute

Eligibility for Tier-1 compute



Flemish University Associations and strategic research centres

Users with HPC experience

The project requires resources not available on Tier-2

or would take too long time to complete on Tier-2

The project's software stack is production-ready

(can run with good performance on Tier-1 infrastructure)

Eligibility for Tier-1 compute

Resource constraints	Starting grant	Regular grant (*)
Core hours	500 K	500 K – 10 M
GPU hours	1 K	1 K – 40 K
Scratch storage	1 TB (100 K files)	2 TB (200 K files)
Duration	8 months	8 months (20% to be used within first 3 months)

(*) If going beyond the limits, additionally motivate why

Steps towards a regular grant

- 1) Decide which Tier-1 machine to go for
(see also the [Hortense](#) and [sofia](#) hardware details)
- 2) Apply for a starting grant (see vscentrum.be/compute)
You may ask one for Hortense and one for sofia
- 3) Use the starting grant to
 - Get to know the Tier-1 hardware
 - Set up and test your software stack
 - Run the requested performance tests
- 4) Pull the latest version of the application form for a regular grant;
check the instructions (see vscentrum.be/compute)

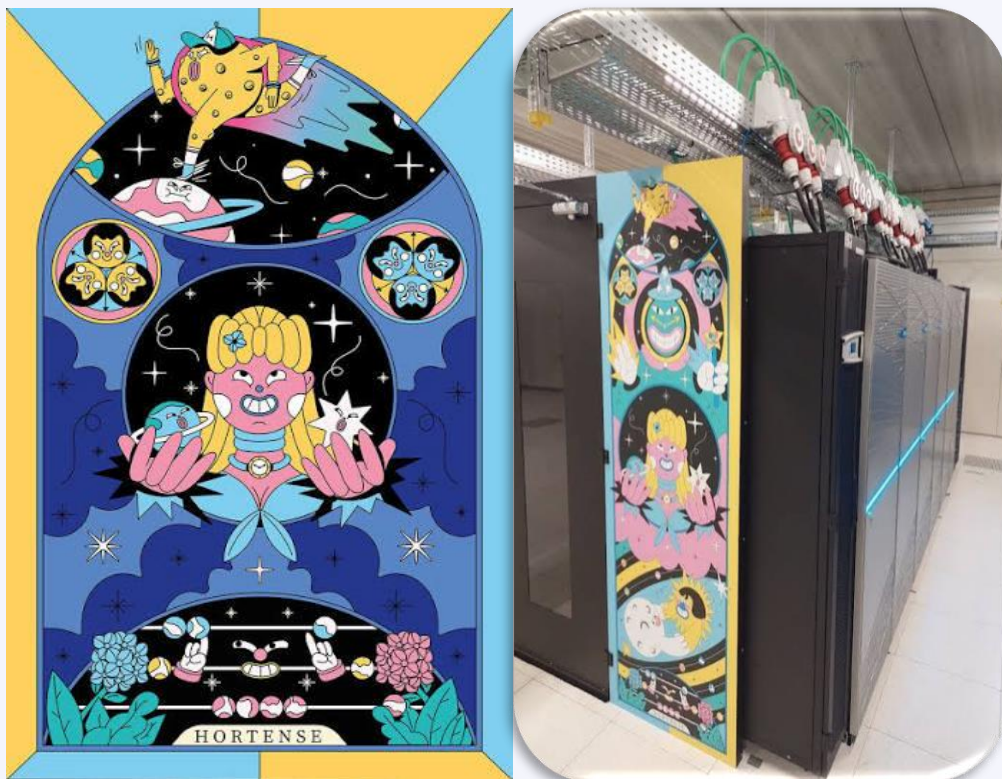
Form and instructions are separate.
Download the instruction PDF!

Steps towards a regular grant

- 5) Fill in the application form timely
- 6) [Optional] ask your [local HPC support](#) to review your draft application
- 7) Submit your application via EasyChair (see vscentrum.be/compute) before the cutoff
- 8) Starting grants: assessed and allocated by the local Tier-1 support team
Regular grants: the Tier-1 allocation board (TAB) reviews all applications
Collaborative grants: evaluated on a rolling basis (no cutoffs) by the VSC MT
See also vscentrum.be/org
- 9) Cutoff + ~3-4 weeks: you receive the decision and feedback
- 10) Cutoff + ~4-5 weeks: start of the 8-month allocation (if granted)

VSC Tier-1 clusters

- **Hortense** operated at UGent
2021 –2027
- Contact: compute@vscentrum.be
- Project follow-up : resapp.hpc.ugent.be



- **sofia** operated at VUB
2026 - ...
- Contact: support@vscentrum.be
- Project follow-up: hub.vscentrum.be



Choosing between *Hortense* and *sofia*

Resource	Hortense	sofia (phase 1)
CPU nodes (thin)	384 nodes with 128 Zen3 cores & 238 GiB RAM	56 nodes with 384 Zen5 cores & 770 GiB RAM
CPU nodes (bigmem)	n/a	16 nodes with 192 Zen5 cores & 1.5 TiB RAM
GPU nodes	20 nodes with 4x A100 80GB	22 nodes with 8x H200 141GB
Choice	• Your code does scale better with 128 core nodes than 384 core nodes • You need large numbers of CPU's • Last phase of a project that was already running on Hortense	• Larger, GPU intensive AI projects • Projects requiring CPUs with large amounts of RAM • Code scales well on the 384 core nodes

Tier-1 compute proposal - Structure

1-2-3 Information on the research project

4 Motivation for using Tier-1 (compared to Tier-2 or Tier-0)

5 Software stack

6 Performance tests

7 Compute time and storage estimates

8 Workflow

9-10 LUMI access and EPICURE support

Sections 1-4

- Introduce the project, supervisor(s) and FRIS/IWETO link
- Be concise and avoid technical jargons.
Tell your scientific story
Refer to the tasks described in the proposal
- [Tip] One proposal per “main” user per cut-off
- Introduce your collaborators and their HPC experience
Tier-2 experience is expected
- Explain why using Tier-1 and not Tier-2
(Not a multiple choice anymore)

Section 5 – Software stack

- Read more about [Scientific Software](#) (modules, EESSI, containers, compilation)
- System-wide stack: via modules or [EESSI](#) (when available)
- Your own stack: Provide links to a public sources/repository
- Available toolchains:
 sofia: 2025a
 Hortense: 2023a, 2023b, 2024a, 2025a, 2025b

Specifically for AI / ML projects:

- Some explanation of code
- Reference to publication describing code

For licensed software, demonstrate that:

- you can use your software/license on Tier-1
- you have sufficient licenses for the largest jobs

Section 6 - Performance tests

Show that the production runs will use the allocated resources efficiently
and
supporting evidence to justify the total amount of requested resources

For every task, set the scene:

- 1) Mention system used for testing (normally same as target system)
- 2) Describe the software setup used for the task (main software packages, important build- & runtime parameters for good performance, ...)
- 3) Use domain/batch/mesh/protein/... sizes similar to production runs
- 4) If applicable, describe how you converted the production run setup into a shorter format suitable for performance testing (max 3-day wall time)

Section 6 - Performance tests

Show that the production runs will use the allocated resources efficiently
and
supporting evidence to justify the total amount of requested resources

Example:

- Task: DFT molecular dynamics with a TZP basis set and OT method for SCF
- Software: CP2K/2025.2-foss-2025a
- Performance parameters: hybrid MPI-OpenMP parallel setup, runtime or MPI env vars., grid cutoffs, ...
- Scaling tests performed on partition X of cluster Y

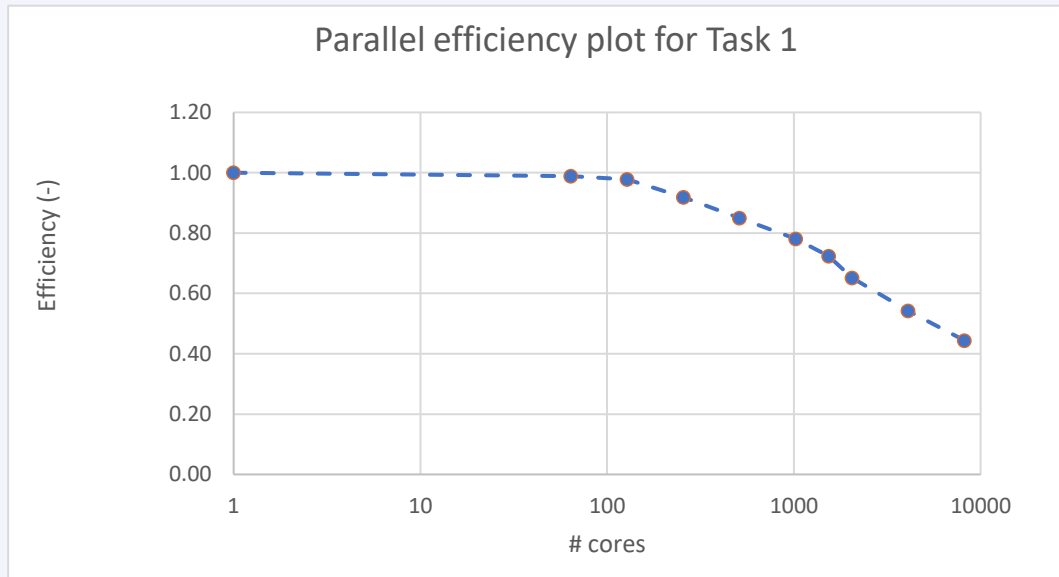
Example AI / ML:

- Task: LLM finetuning using PyTorch
- Software: PyTorch 2.9.1 installed in a venv based on Python/3.13.5-GCCcore-14.3.0 and CUDA/12.9.1
- Performance parameters: data- and model parallelism; batch sizes; ...
- Scaling tests performed on partition X of cluster Y

Section 6 - Performance tests - CPU

Make and describe tables and figures showing parallel efficiency numbers. Start from the most minimalistic resource setup possible, e.g. a single core.

Nr of nodes	Total nr of cores	Wall-time (s)	Speed-up w.r.t. baseline	Efficiency
1	1	200000,0	1,00	1,00
1	64	3160,9	63,27	0,99
1	128	1596,9	125,24	0,98
2	256	850,0	235,29	0,92
4	512	460,0	434,78	0,85
8	1024	250,0	800,00	0,78
12	1536	180,0	1111,11	0,72
16	2048	150,0	1333,33	0,65
32	4096	90,0	2222,22	0,54
64	8192	55,0	3636,36	0,44

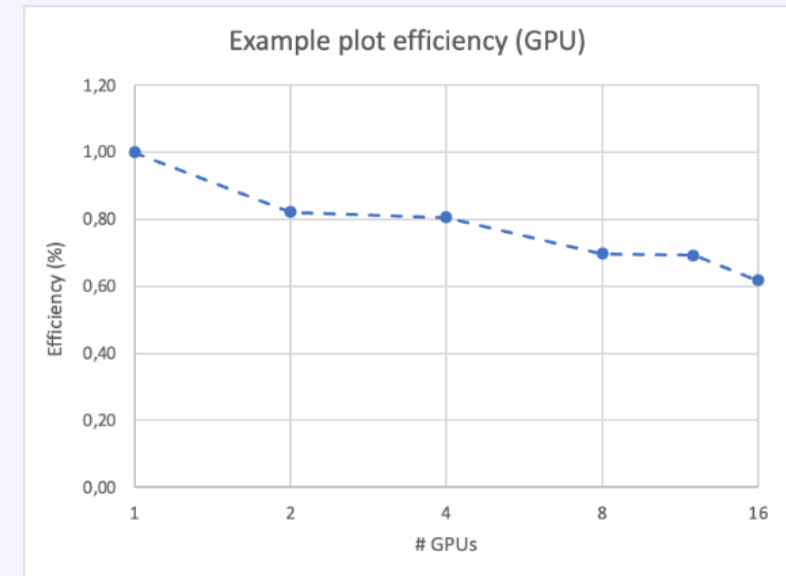


Preferably take the average time of three separate runs

Section 6 - Performance tests - GPU

Make and describe tables and figures showing parallel efficiency numbers. Start from the most minimalistic setup possible, e.g. a single GPU.

Nr of nodes	Total nr of cores	Total nr of GPUs	Wall-time (s)	Speed-up w.r.t. baseline	Efficiency
1	12	1	692	1,00	1,00
1	24	2	422	1,64	0,82
1	48	4	215	3,22	0,80
2	96	8	124	5,57	0,70
3	144	12	83	8,31	0,69
4	192	16	70	9,89	0,62



How do you use CPUs?

Section 6 - Performance tests

Check the guidelines

The benchmark should do similar I/O as in the production runs.
In case of any deviations, describe what effect these will have on performance.

Make sure not to mix strong scaling and weak scaling graphs.

When using AMR codes: describe how dynamic load balancing is addressed and how many AMR levels are activated in the tests.

Section 6 - Performance tests

When different techniques are used, e.g. DFT and CC, make sure to have benchmarks for all of them

Specifically for ML: do batch size scaling at GPU level.

For VASP: find recommendations [in this repository](#).

Section 6 - Performance tests

Contact your [local Tier-2 support team](#) first, or Tier-1 support for specific technical assistance if you see lower parallel efficiencies than expected (i.e. below 60%-70%).

Best approach then depends on the underlying cause:

- Poor load balancing
- Communication overhead
- Process or thread pinning (should e.g. take AMD CPU cache hierarchy into account)
- Memory bandwidth contention
- I/O bottlenecks
- ...

Section 7 - Compute time and storage estimates

Using the results from Section 6, deduce the total amount of compute & storage resources

	Core-hour calculation									Storage volume estimate	
Computational task	Number of such jobs	Wall clock time (in hours) per job	Number of Tier-1 nodes per job	Number of Tier-1 cores per job (per node!)	Memory factor (mem-per-core / default-mem-per-core)	Total core-hours per task	Memory usage (GiB) per node per job	Partition	OpenMP / MPI / hybrid / worker framework	Tier-2 DATA/HOME volume (TiB) + number of files	Tier-1 SCRATCH volume (TiB) + number of files
Task 1 CP2K – molecular dynamics 50 ns runs PBE functional 1 -> 5 water molecules	A 30	B 48	C 5	D 128	E 1	A x B x C x D x E 921 600	250	cpu_milan	Hybrid MPI- OpenMP	0 TiB 0 files	0.2 TiB 5000 files
Task 2 CP2K – molecular dynamics 100 ns runs PBE functional 1 -> 5 water molecules	5	72	10	64	2	460 800	250	cpu_milan	Hybrid MPI- OpenMP	0 TiB 0 files	0.1 TiB 500 files
						1 382 400					0.2 TiB 5000 files

Section 7 - Compute time and storage estimates

Using the results from Section 6, deduce the total amount of compute & storage resources

- For each task, motivate the chosen job size
(e.g. *“X nodes per job because PE of 80%”*)
- Explicitly describe how the job wall times in Table 5 have been calculated
- For GPU jobs, the proportional CPU cores are included
- Double check the math in Table 5 (and the numbers on page 1!)
Required CPU and GPU hours = $\text{sum}(\text{tasks})$
Required storage = $\text{sum}(\text{tasks})$ if concurrent else $\text{max}(\text{tasks})$

Section 8 - Workflow

Show that you have a solid plan to manage the many calculations, data transfers, etc.

Recommendations:

- Use scripts and workflow managers where appropriate
- If you use checkpoints, which ones will be kept
- Also describe your data exit strategy (volume, # of files, transfer tool, timeline, ...)
[Tip] [Globus](#) allows keeping two endpoints in sync
- [Tip] Gantt chart with distribution of planned workload is recommended

Sections 9/10 - LUMI access and EPICURE support

- Flanders has guaranteed time on Tier-0 [LUMI](#).

Eligibility:

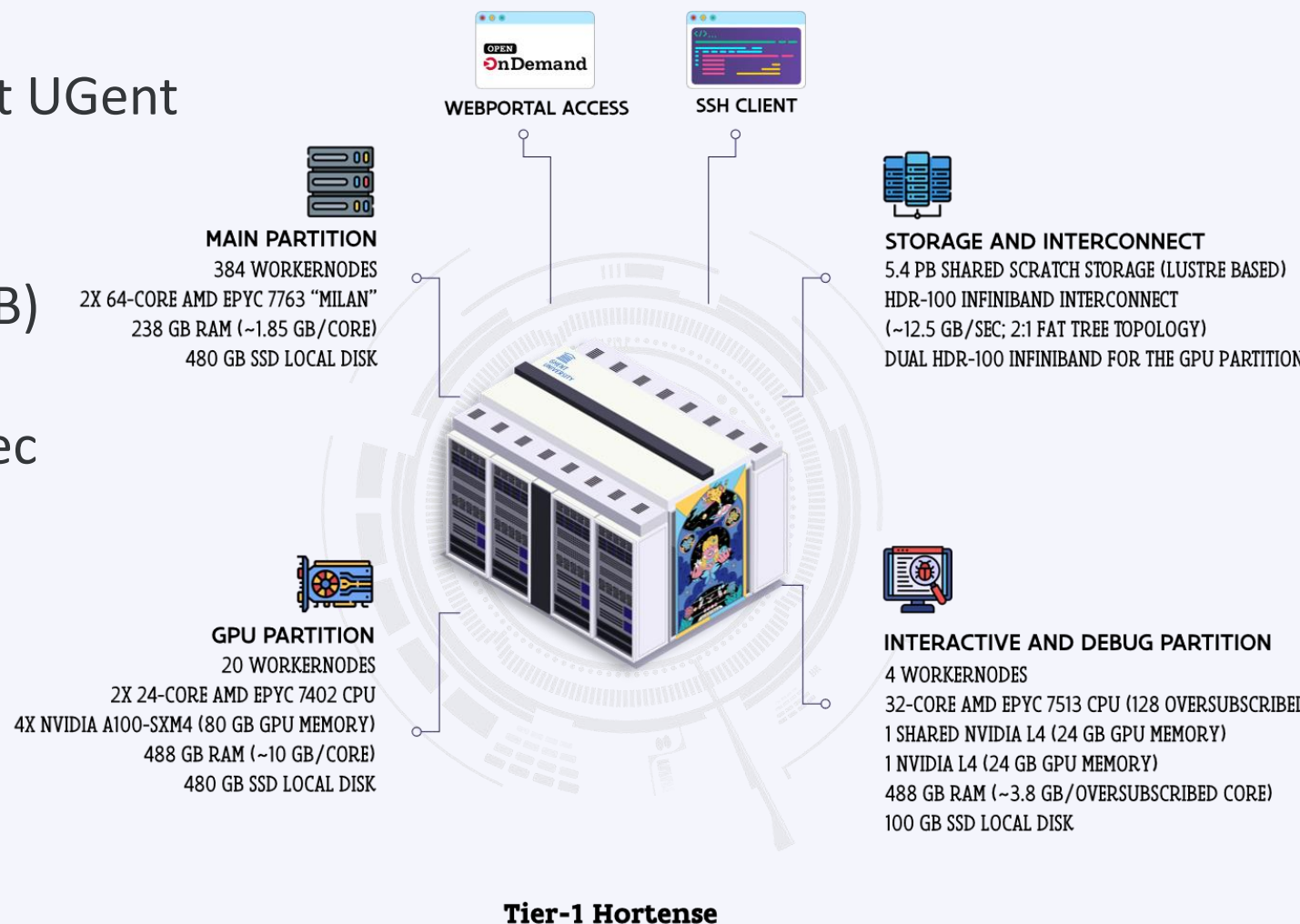
- GPU applications that run on AMDs
 - Need more GPU time on Tier-1 in the future
 - Tick the box, and you will get contacted by the VSC-LUMI support team.
-
- If you develop your own software and you have performance issues, would like to increase scalability or your code needs to be refactored, the VSC Tier-0 support team or the [EPICURE project](#) could help you. Tick the box and we will contact you.



Tier-1 Compute – technical details

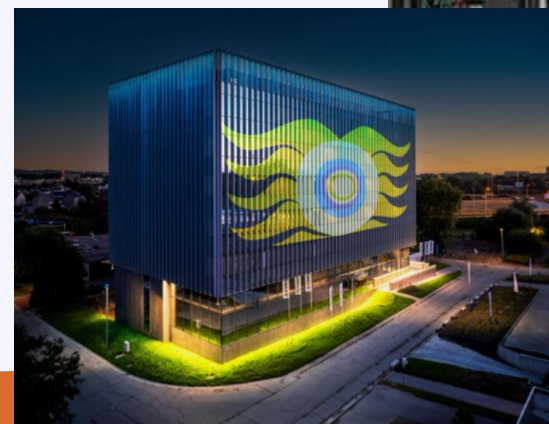
VSC Tier-1 clusters - Hortense

- Hortense (2021 –2027) is operated at UGent
- 384 thin nodes; 49K cores
- 20 GPU nodes (4x NVIDIA A100, 80GB)
- 5 PB storage
- Infiniband interconnect: ~ 12.5 GB/sec
- More technical details on [VSCdocs](https://vscdocs.be).
- Contact: compute@vscentrum.be



VSC Tier-1 clusters - sofia

- **sofia** (2026 -) is managed by VUB
- BRU01 datacenter in Research Park Zellik
- In production since July 2026
- ~20% expansion in 2027
- 56 thin nodes; 21K CPU cores; 2GB/core
- 16 bigmem nodes; 3K CPU cores; 8GB/core
- 22 GPU nodes; 176 GPUs (8x NVIDIA H200; 140 GB)
- 4 vis nodes (2x NVIDIA Ada RTX 5000, 32 GB)
- Infiniband interconnect: 200/400 Gbps
- Contact: support@vscentrum.be



sofia job submission



- No cluster modules: specify a partition
 - **zen5_dense** (384 cores/node, ~2 GiB/core)
 - **zen5_himem** (192 cores/node, ~8 GiB/core)
 - **zen4_h200** (8 H200 GPUs/node, 24 cores/GPU, ~8 GiB/core)
 - **zen5_vis** (interactive use)
- Job resources
 - Fixed CPU-memory per core
 - Fixed 24 cores per GPU
- Native Slurm usage
 - Slurm commands **sbatch**, **squeue**, **scancel**, ...
 - Interactive job **salloc -p <partition>**
 - Job status **slurm_jobinfo <jobid>**
- Clean job environment by default
 - Use **--export** to customize (no **--export=ALL!**)

sofia job submission

- MPI jobs
 - *mympirun* wrapper not supported
 - Use **mpirun** or **srun**
 - Manually control process/thread placement/binding!
- OnDemand portal
 - <https://portal.sofia.vub.be>



sofia storage

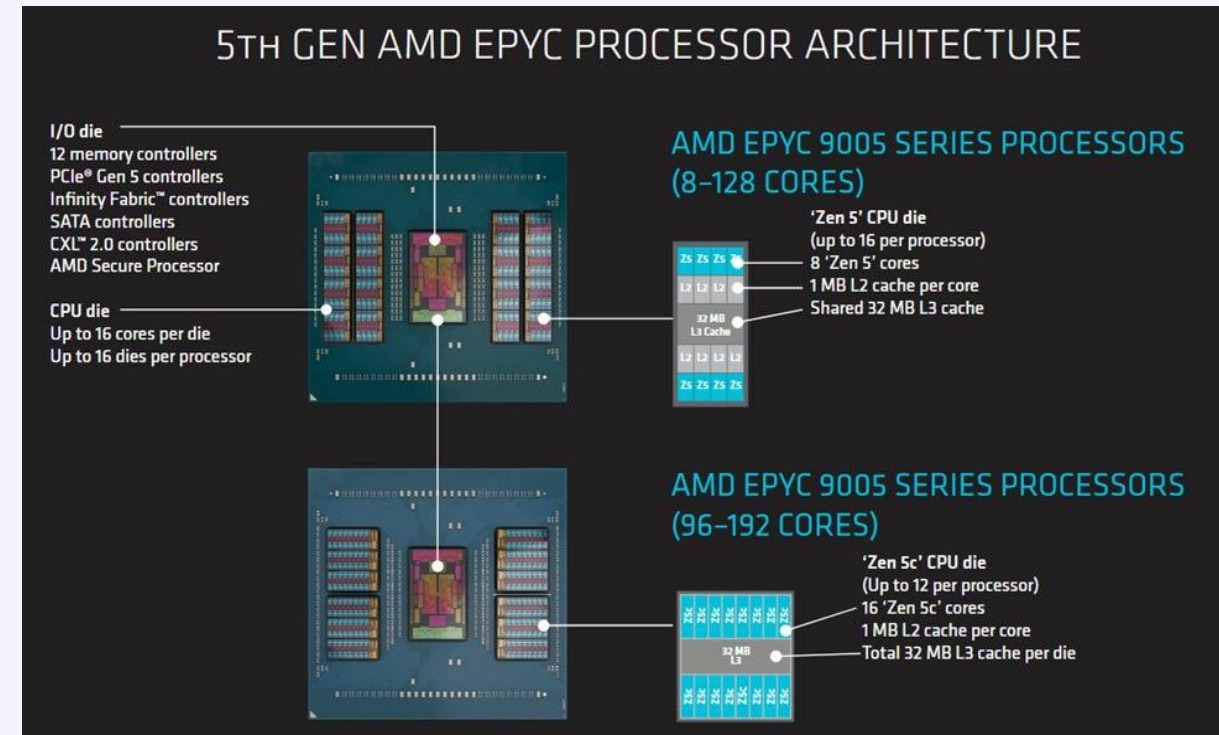


- Parallel filesystem: IBM storage scale (GPFS)
- Home on scratch (50 GB)
 - **sofia** remains accessible when local Tier-2 is down
 - Don't use home for project data
- Data transfer
 - Tier-2 storage (`$VSC_HOME`, `$VSC_DATA`) is not mounted
 - Globus endpoint for **sofia**
 - `rsync/scp`
- Local scratch = private and temporary
 - `/tmp` (`$TMPDIR`, `$VSC_SCRATCH_NODE`)
 - `/dev/shm` (in-memory storage)
- **sofia** access ends shortly after project expiration
 - Retrieve data timely
 - No backups!

sofia CPUs



- CPU microarchitecture
 - Zen4: double-pumped 256-bit AVX-512
 - Zen5: full 512-bit AVX-512
- zen5_dense (zen5c)
 - 2 * 192 cores
 - 2 * 12 CCDs
 - 2 * 4 NUMA nodes
 - 2 * 384 MB L3 cache
- zen5_himem (zen5c) and zen4_h200 (zen4)
 - 2 * 96 cores
 - 2 * 12 CCDs
 - 2 * 4 NUMA nodes
 - 2 * 384 MB L3 cache



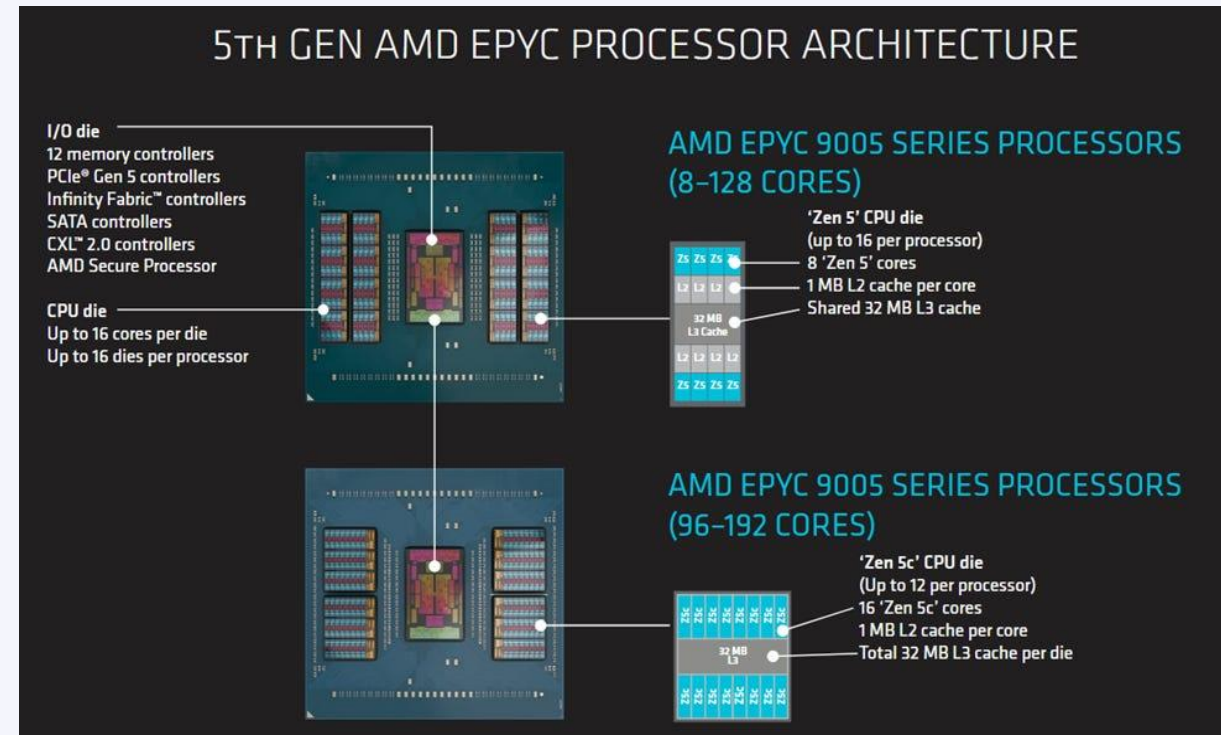
sofia GPUs

- zen4_h200 (NVIDIA Hopper H200)

- 8 GPUs/node
- 141 GB VRAM
- NVLink 4 (~900 GB/s)

- zen5_vis (NVIDIA Ada RTX 5000)

- 2 GPUs/node
- 32 GB VRAM
- Interactive usage (OnDemand)



sofia software



- Software modules
 - Optimized for the different CPU/GPU architectures
- EESSI software stack (TBD)
- Install on your own
 - See VSC training "Getting Scientific Software Installed"
<https://vscentrum.github.io/gssi-training/2025>
 - <https://www.vscentrum.be/events/getting-scientific-software-installed-2>
- Support (software installations, job troubleshooting)
 - Local HPC support teams will be more involved
 - Contact: support@vscentrum.be

sofia benchmarking: weak scaling (HTC)



- Multiple tasks running concurrently on a single node
- Important for **sofia** with 384 cores/node
- Example: weak scaling job script

```
#!/bin/bash
#SBATCH --partition=zen5_dense # 384 cores per node
#SBATCH --nodes=1
#SBATCH --ntasks=48 # 48 tasks
#SBATCH --cpus-per-task=8 # 8 cores per task

for x in $(seq 48); do
    srun --ntasks=1 --exact <my-prog> &
done
wait
```

sofia performance: parallel scaling



- Process/thread placement and binding (pinning)
- Compute limited vs. memory bandwidth (MB) limited
 - MB limited software: OpenFOAM, ANSYS Fluent, ...
 - MB is more important with larger system size
 - MB per core: `zen5_himem` ~ 2 x `zen5_dense`
 - Optimize #cores per process (e.g. 2, 3, 4 cores/process)
- Pure MPI vs. hybrid MPI-threading
- MPI implementation: OpenMPI, Intel-MPI, MPICH
- Consult documentation of your software

sofia performance: MPI



- Example: pure MPI with compact process binding

```
#SBATCH --partition=zen5_dense
#SBATCH --nodes=1 --ntasks-per-node=384 --cpus-per-task=1

export OMP_NUM_THREADS=1
mpirun --map-by slot:PE=1 --report-bindings <mpi-prog>
srun -m block:block --cpu-bind=verbose,cores <mpi-prog>
```

- Example: hybrid MPI-OpenMPI threading with compact process binding

```
#SBATCH --partition=zen5_dense
#SBATCH --nodes=1 --ntasks-per-node=192 --cpus-per-task=2

export OMP_NUM_THREADS=2
mpirun --map-by slot:PE=2 --report-bindings <mpi-prog>
srun -m block:block --cpu-bind=verbose,cores <mpi-prog>
```

sofia performance:

MPI



- Example: pure MPI with compact process binding, skipping every other core (MB limited)

```
#SBATCH --partition=zen5_dense
#SBATCH --nodes=1 --ntasks-per-node=192 --cpus-per-task=2

export OMP_NUM_THREADS=1
mpirun --map-by slot:PE=2 --report-bindings <mpi-prog>
srun -m block:block --cpu-bind=verbose,cores <mpi-prog>
```

sofia performance: multi-threading



- Example: OpenMP with compact thread binding

```
#SBATCH --nodes=1 --ntasks=1 --cpus-per-task=16
```

```
export OMP_NUM_THREADS=16
```

```
export OMP_PLACES=cores
```

```
export OMP_PROC_BIND=close
```

```
# Intel-compiler/LLVM equivalent:
```

```
export KMP_AFFINITY=compact
```

```
<my-prog>
```

sofia performance:

PyTorch multi-GPU



- 1 process per GPU
- Optimize #CPU threads per process (max. 24)
- Try to keep GPUs ~100% busy
 - Pre-load data in memory
 - Load data in parallel
 - Optimize batch size: use all GPU memory
- Use GPU monitoring
 - Weights & Biases (**wandb**)
 - ☐ "Free forever for academic research"
 - ☐ <https://wandb.ai/site/pricing/>

Example: 2 GPUs, 1 task per GPU,
24 threads per task

```
#SBATCH --partition=zen4_h200
#SBATCH --nodes=1
#SBATCH --gpus-per-node=2
#SBATCH --ntasks-per-gpu=1
#SBATCH --cpus-per-task=24
```

The background is a complex collage. On the left, there's a stylized cityscape with arches and buildings. In the center, a line graph with an upward-pointing red arrow is overlaid on a grid. On the right, a detailed molecular structure with blue and red spheres is prominent. A large green leaf is also visible in the lower right quadrant.

Tier-1 Cloud

Eligibility for Tier-1 Cloud



Flemish University Associations & strategic research centres

Users with cloud experience & able to manage VMs

The project requires VMs, web apps, SQL databases, ...

to support specific workflows, postprocessing, analyses, ...

The project requires integration with other VSC services

(Tier-1 Compute and/or Tier-2 Compute and/or Tier-1 Data)

Eligibility for Tier-1 Cloud

Resource constraints	Starting grant	Regular grant
Instance types	UPSv1, CPUv1, GPUv2, GPUv3 (*)	
Flavor sizes (per VM)	≤ 360 GB RAM, ≤ 80 GB disk, ≤ 20 cores, ≤ 1 GPU (*)	
Shared storage (across VMs)	unspecified, presumably ≤ 10 -100 GB	
# of public IPv4 addresses	1	1
Duration	8 months	8 – 24 months

(*) see the [VSCDocumentation](#)

Steps towards a regular grant

- 1) Find the optimal web services you want to serve via Tier-1 Cloud and how these interact with other VSC services
- 2) Apply for a starting grant (see vscentrum.be/cloud)
- 3) Read the latest application form for a regular grant (see vscentrum.be/cloud)
- 4) Use the starting grant to prepare for a regular grant
- 5) [Optional] ask your local HPC support to review your draft application
- 6) Submit the application for a regular grant via EasyChair
- 7) Applications get evaluated on a rolling basis (no cutoffs) by the VSC MT (see vscentrum.be/org)

Sections of the regular grant proposal

1-2-3 Information on the research project

4 Motivation for Tier-1 Cloud

5 Software stack and services (OS choice, databases, orchestrators, ...)

6 Workflow

7 Required resources (processors, memory, storage, connectivity)

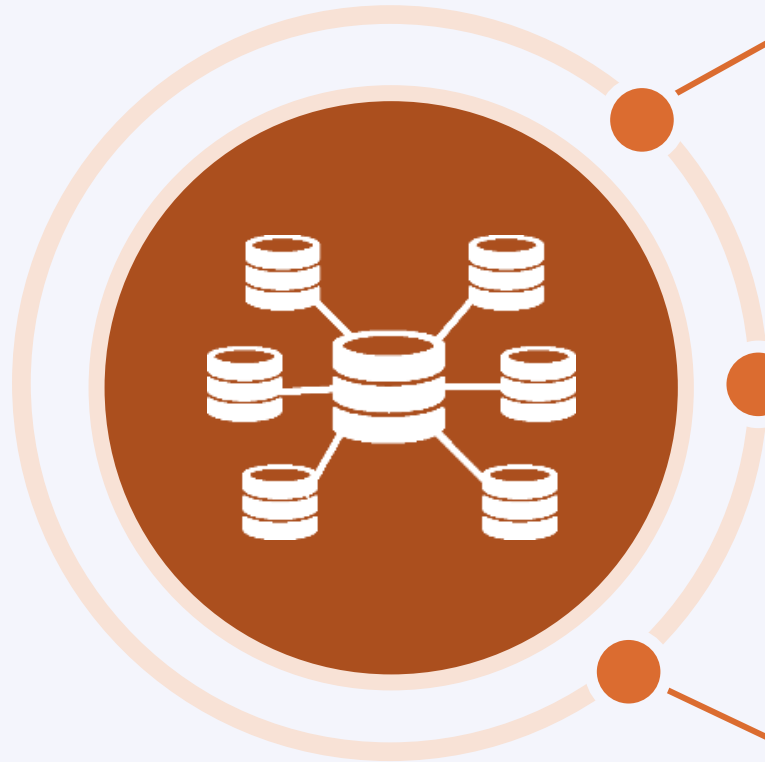
8-9-11 VM and data management (e.g. with respect to security)

10 Exit strategy (which images and data to send where for preservation)

Tier-1 Data



Tier-1 Data – eligibility



Flemish Universities Assoc and strategic research centres

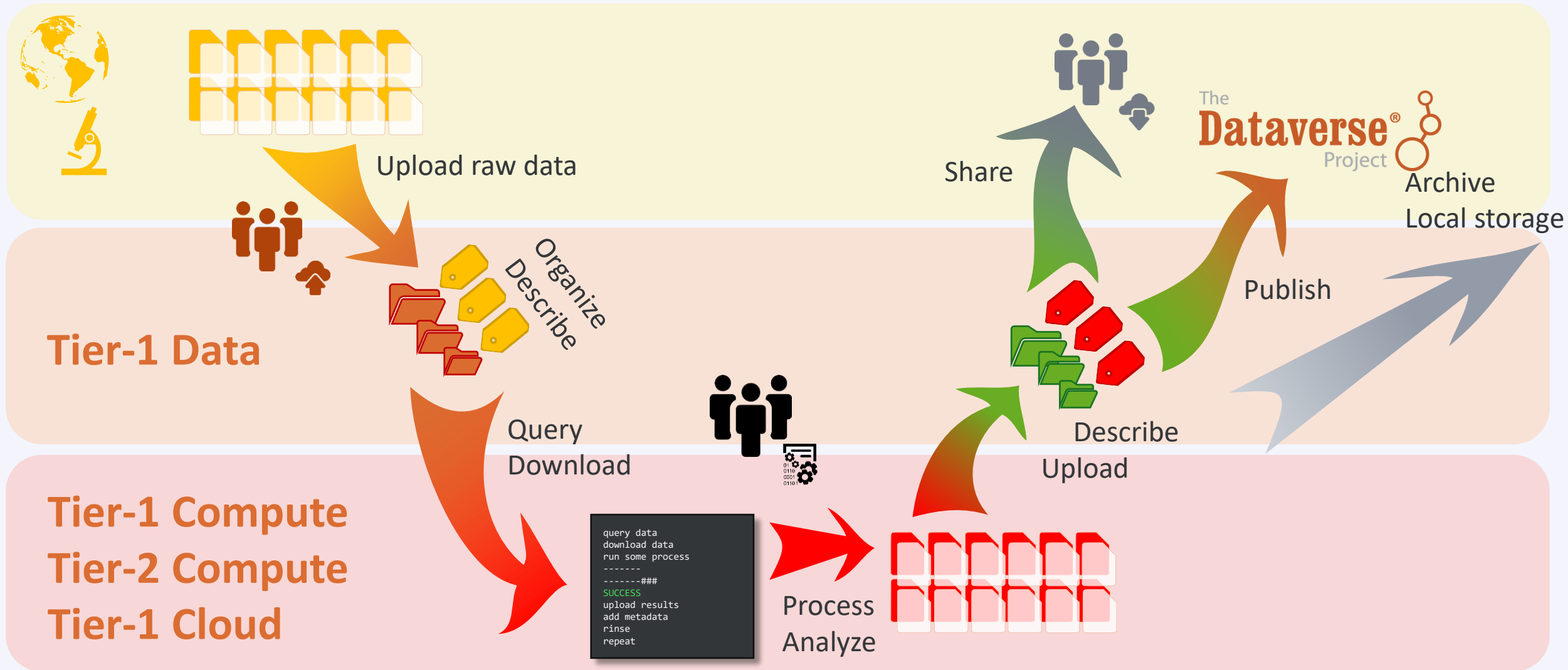
No previous data management experience required

The project requires (large) storage for *active* research data
and (optionally) advanced data management & workflows

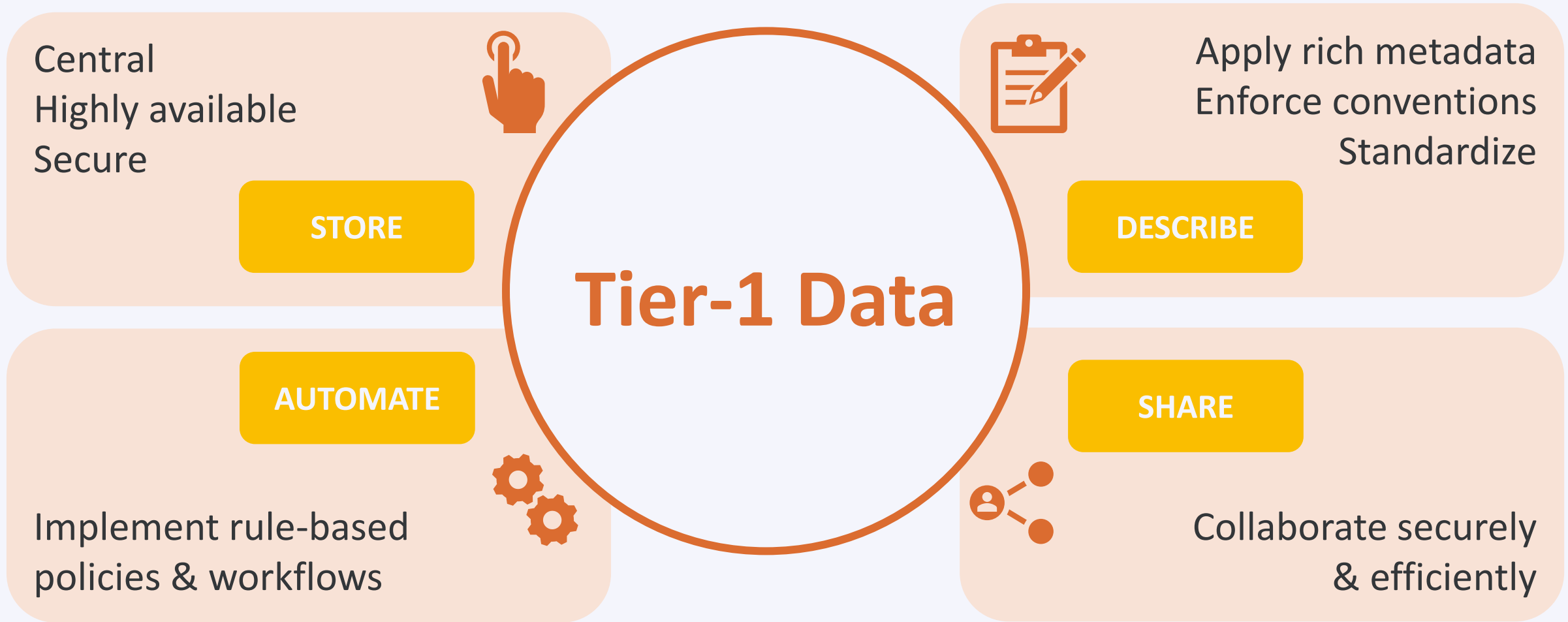
The project requires integration with other VSC services

(Tier-1 Compute and/or Tier-2 Compute and/or Tier-1 Cloud)

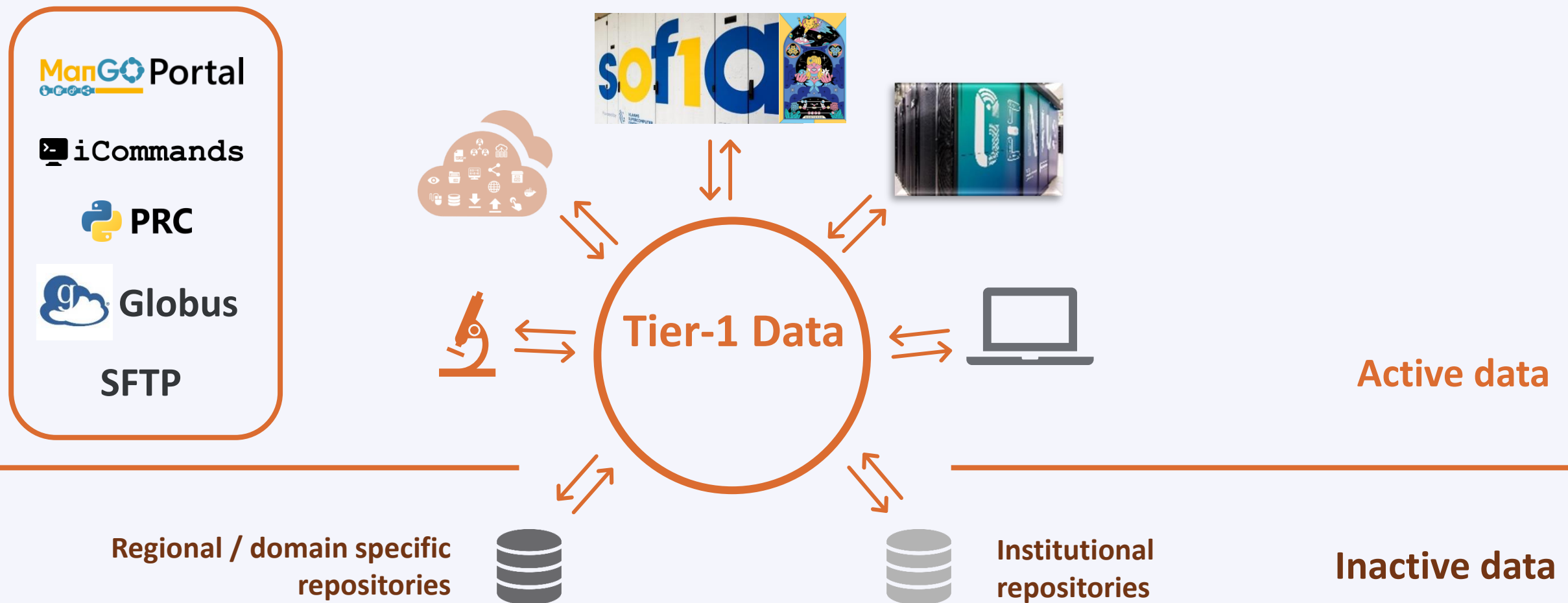
Tier-1 Data – example workflow



Tier-1 Data – more than storage



Tier-1 Data – staging in and out



Tier-1 Data – adding metadata

Example metadata for a collection & data object



ProjectID: FWO99
Title: Thermodynamics of coffee
PI: ProfessorBrown
StartDate: 2026-01-01
EndDate: 2028-12-25
Status: ToArchive



SimulationID: FWO99-sim1
System: Caffeine
JobID: 40000001
System: VSC-Hortense
Software: CP2K
Version: 2026.1
Date: 2026-04-20

Metadata can be added

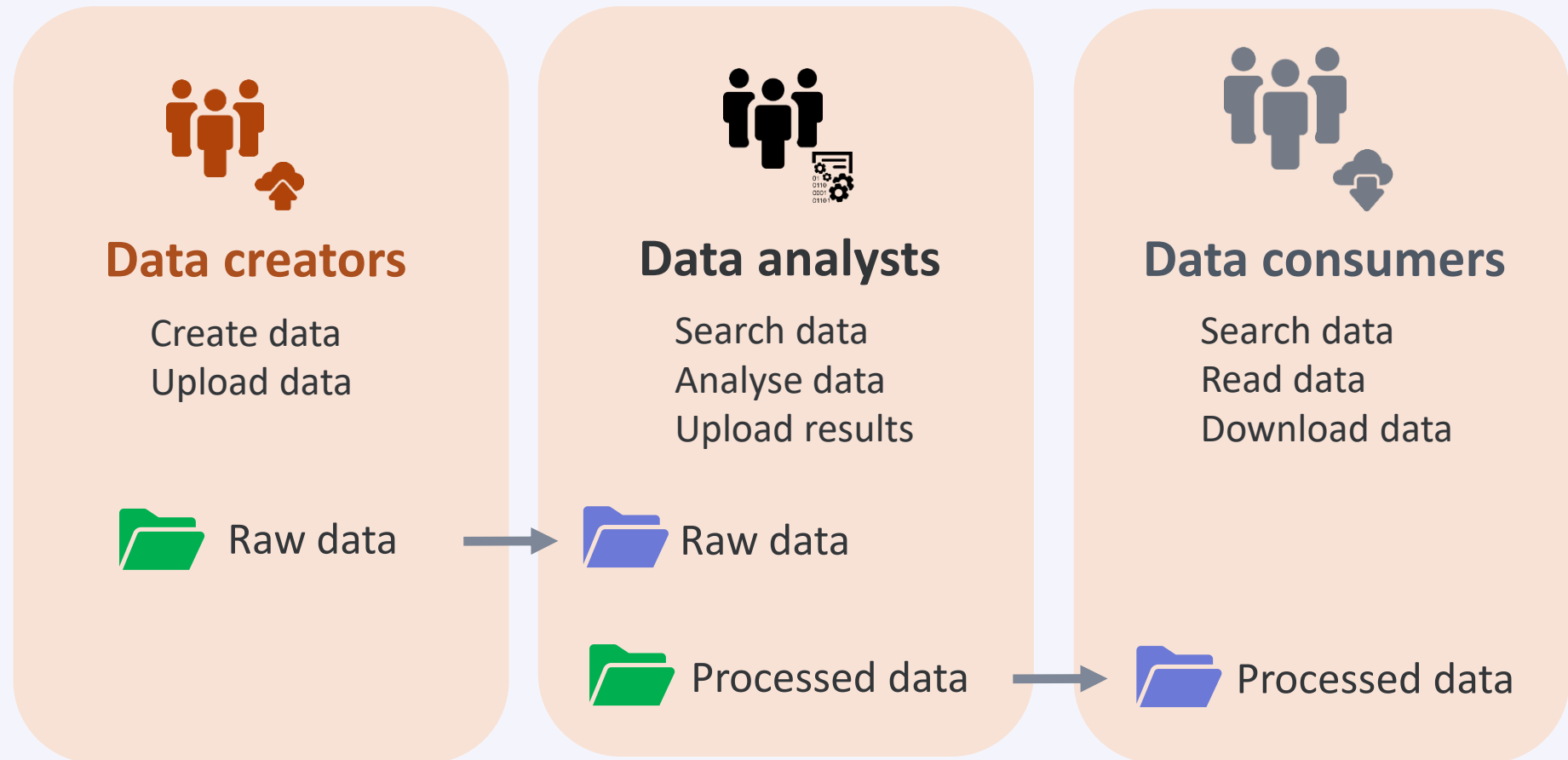
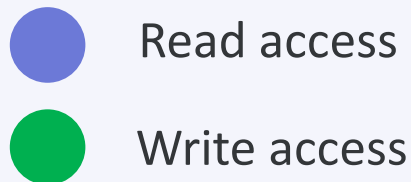
- By hand
- Via predefined schemas
- By automated extraction from file headers, file names, sidecar files, ...

Tier-1 Data – group permissions for collaboration

Example:

Access Levels:

1. Read
2. Write
3. Own



Tier-1 Data – automation

Example:

- Look for all collections with the `Status=ToArchive` as metadata
- Make a read-only copy of the collection
- Add `ArticleID=article1` metadata to it

Implementation:

- Client-side: e.g. own scripts using the iRODS Python client
- Server-side: e.g. actions to take upon file upload

(to be discussed with Tier-1 Data support)

Tier-1 Data – access

Resource constraints	Starting grant	Regular grant
Volume	5 TB	5 TB – 1 PB
Duration	8 months	8 months – 3 years

Steps towards a regular grant

- 1) Decide which datasets and workflows you exploit on Tier-1 Data
- 2) Apply for a starting grant (see vscentrum.be/data)
- 3) Read the latest application form for a regular grant (see vscentrum.be/data)
- 4) Use the starting grant to prepare for a regular grant
- 5) [Optional but recommended] ask data@vscentrum.be to review your draft application
- 6) Submit the application for a regular grant via EasyChair
- 7) Applications get evaluated on a rolling basis (no cutoffs) by the VSC MT (see vscentrum.be/org)

Sections of the regular grant proposal

1-2-3 Information on the research project

4-5 Added value of Tier-1 Data for the project compared to other solutions

6 Data workflow (sources, preprocessing, analyses, permissions, ...)

7-9 Description of the datasets and approach to metadata

10 Exit strategy (which data to send where for preservation)



Collaborative Grant

Collaborative grants

Rules and form on the [VSC website](#)

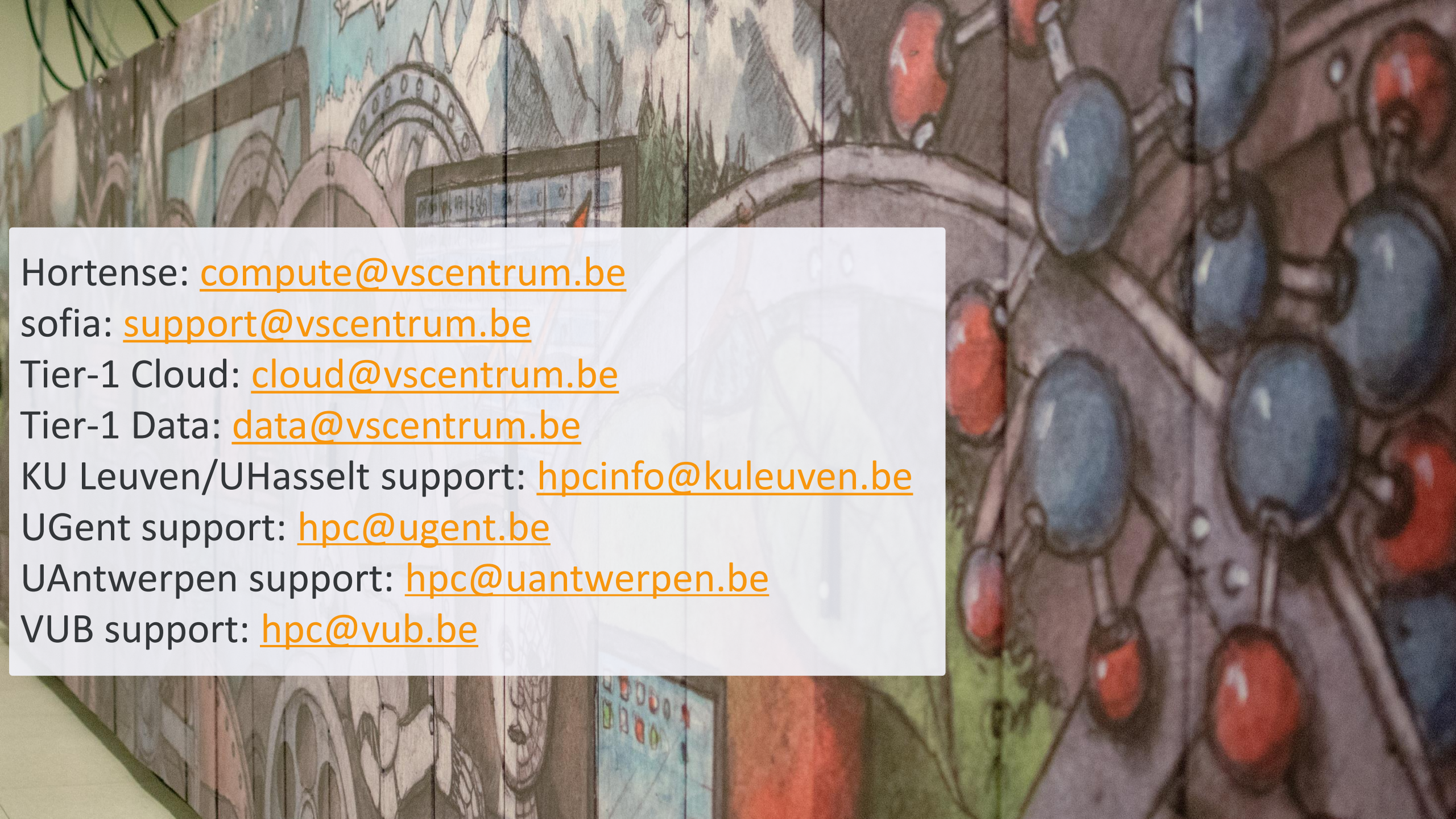
At least 3 research groups from different institutes

Proposal can include Tier-1 Compute, Cloud and Data

Same structure as regular grants

12 month allocation / yearly resubmission

Allocation at any time of the year
Review by the VSC Management Team



Hortense: compute@vscentrum.be

sofia: support@vscentrum.be

Tier-1 Cloud: cloud@vscentrum.be

Tier-1 Data: data@vscentrum.be

KU Leuven/UHasselt support: hpcinfo@kuleuven.be

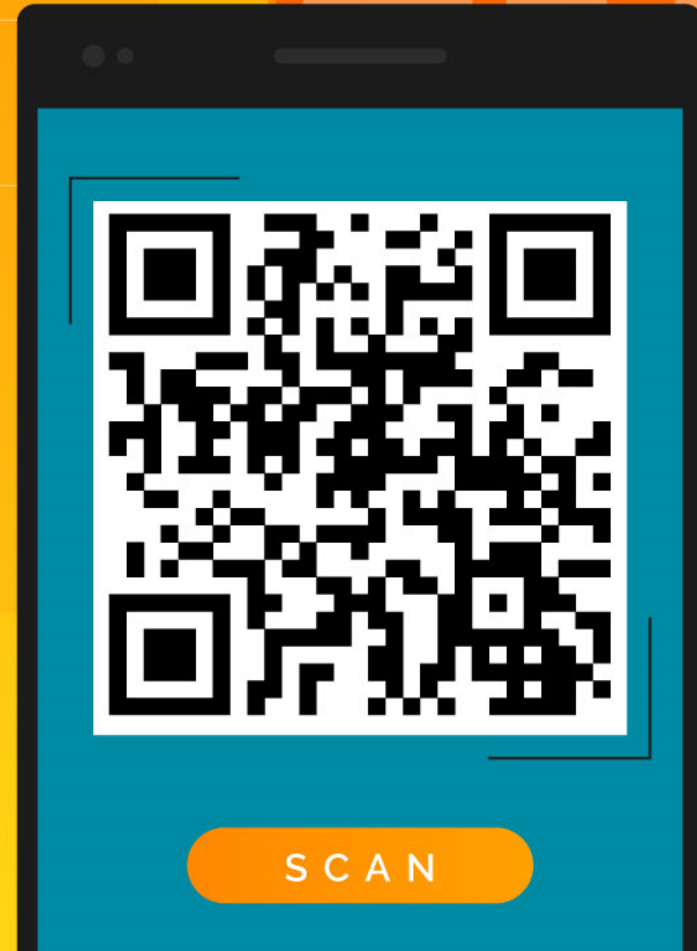
UGent support: hpc@ugent.be

UAntwerpen support: hpc@uantwerpen.be

VUB support: hpc@vub.be

*Stay Connected
to VSC*

Linked  [®]



The background is a collage of three distinct images. On the left, a stylized cityscape with arched buildings and a mountain in the background. In the center, a line graph with an upward-trending orange arrow. On the right, a close-up of a necklace with large blue and red beads.

Addendum – all Tier-1 services

Recommendations for all regular grants

- Clear, consistent and well-structured proposals have a higher chance of getting accepted
- Project descriptions need to be self-contained and be understandable by a broad audience $\Rightarrow$ avoid jargon, URLs, citations, ...
- Each proposal must be self-contained $\Rightarrow$ do not refer to (scaling tests from a) previous proposal

Recommendations for all regular grants

- Address any remarks you received on previous proposals
- Always use the most recent templates from vscentrum.be/{compute,cloud,data}
- Respect the length limits for each section and for the total document
- Provide all requested information
- Verify that the information on the first page matches that of the corresponding sections



Addendum – Tier-1 Compute

Recommendations for Tier-1 Compute regular grants

General

- Each proposal is thoroughly reviewed by 2 board members
- The proposal is scored together with the full board
- As a rule, the 8-month allocation duration is never extended (except in case of major issues with the Tier-1 cluster itself)
⇒ make sure to use all the allocated compute time within these 8 months (the allocation board will verify this)
- Though most Tier-1 projects involve MPI/OpenMP-... parallelized applications, high-throughput computing (HTC) workloads are also accepted on Tier-1

Recommendations for Tier-1 Compute regular grants

Section 5

- Make sure to describe your own local software installations (if any) in sufficient detail (package versions, associated math and communication libraries, ...)
- If selecting an old software version: explain why more recent releases are not an option
- In case of licensed software:
 - Prove your license is valid and allows to install and use the software on Tier-1
 - Be sure the license manager (if any) can be used on Tier-1
 - Be sure you have enough licenses for your largest jobs

Recommendations for Tier-1 Compute regular grants

Section 5 (cont)

When compiling software from source yourself on Hortense or **sofia**:

- If possible, avoid old toolchains (use 2025a or newer)
- Intel compilers: Avoid `-xHost`;

Hortense: use `-march=core-avx2` (or `-mavx2 -fma`)

sofia: `-march=rocketlake`

When using `-xHost`, Intel compilers fall back to SSE4.2 (no AVX or AVX2!)

Avoid using the `buildenv` module!

Recommendations for Tier-1 Compute regular grants

Section 5 (cont)

When compiling software from source yourself on Hortense or **sofia**:

- When linking with Intel MKL, good performance may also require extra care
MKL may fall back to SSE4.2 implementations (no AVX or AVX2!)
- BLAS/LAPACK: Intel MKL (intel/...) and OpenBLAS (foss/...)
are mostly on-par w.r.t. performance
- FFT: some functions in FFTW can be significantly slower than FFTW wrappers in Intel MKL

Recommendations for Tier-1 Compute regular grants

Section 6

- The tables and figures are not sufficient – also summarize your findings in the text and (try to) explain any anomalies
- #SBATCH --exclusive avoids interference from other jobs running on the same nodes (even yours)
- For GPU applications:
 - Take into account the max number of CPU cores per GPU
 - Also mention what the CPU is used for
 - If feasible, also obtain and mention the runtime on e.g. a full CPU node (without GPUs) to prove the relevance of using GPU acceleration
 - Mention relevant parameters such as batch sizes

Recommendations for Tier-1 Compute regular grants

Section 6 (cont)

Scaling tests for HTC workloads (weak scaling):

- First check to what extent a single worker can efficiently use multiple cores, N
- Then run two, four, ... workers concurrently (each possibly using N cores).

Evaluate the parallel efficiency until e.g. an entire node is used

Recommendations for Tier-1 Compute regular grants

Section 7

- Clarify which task(s) will be performed sequentially and which one(s) will be performed concurrently
- The memory factor will only be different from 1 (i.e., > 1) if you need to leave cores idle because of memory requirements or performance reasons (memory bandwidth contention)

Recommendations for Tier-1 Compute regular grants

Section 7 (cont)

- Consult your research project's data management plan (DMP), as it may already contain some of the information needed for this section

Section 8

- Also here the DMP may already contain some of the information for this section
- The Tier-1 Data support team (data@vscentrum.be) can also assist you in choosing appropriate metadata standards

The background is a complex collage. On the left, there's a stylized cityscape with arches and buildings. In the center, a line graph with an upward-pointing red arrow is overlaid on a grid. On the right, a detailed molecular structure with blue and red spheres is visible. A large green leaf is also present in the lower right quadrant.

Addendum – Tier-1 Cloud

Recommendations for Tier-1 Cloud regular grants

Do add a (quantitative) motivation for the requested resources,
especially when you request additional resources in a follow-up project

Starting 2027, yearly usage reports will be requested:

- how many users your project serves
- how frequently the services are used, ...