



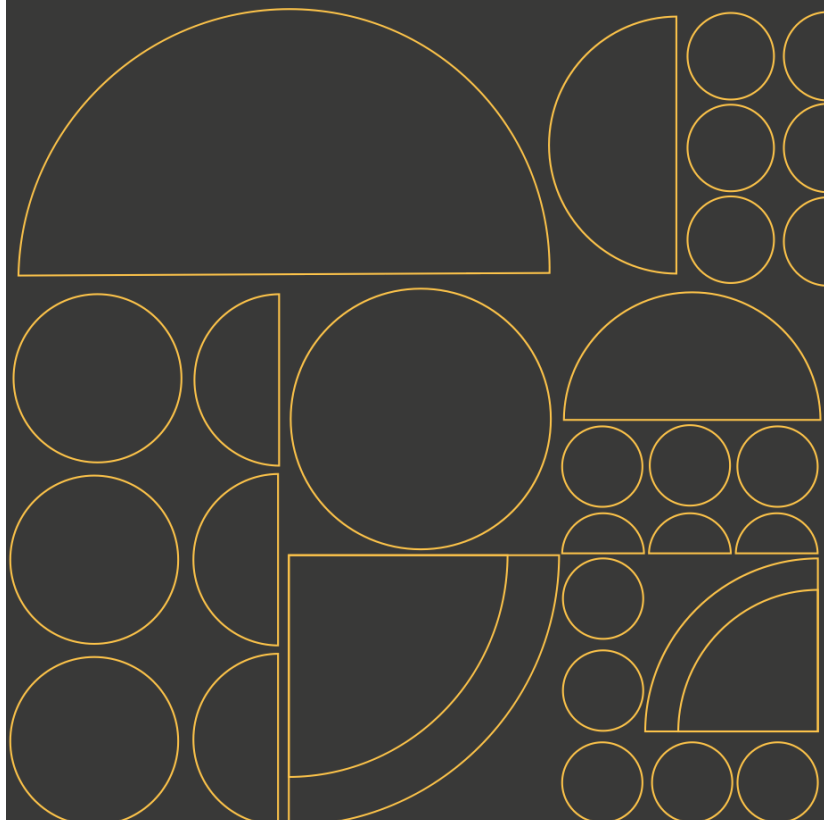
ChEESE

WP4
Simulation ecosystem

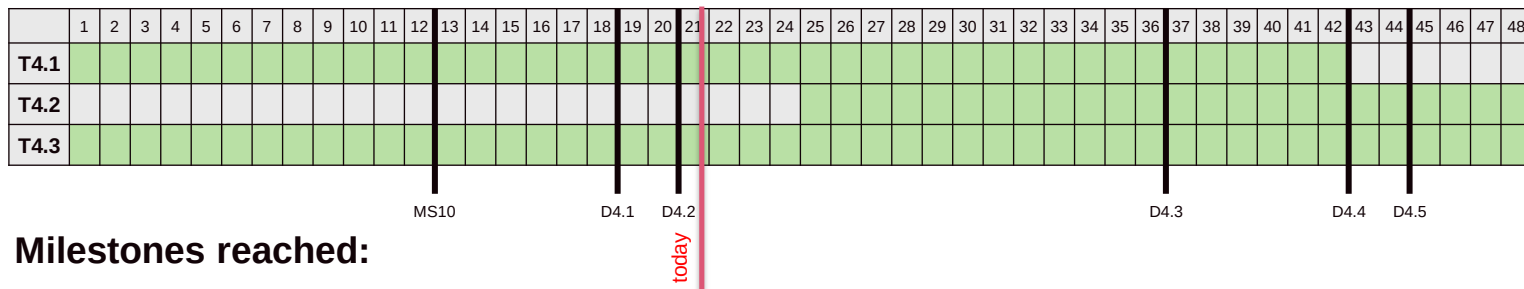
Christoph Niethammer
HLRS



Project funded by EuroHPC under the grant agreement No 101093038.



WP4 status overview



Milestones reached:

MS10	Application deployment	Codes available to users of EuroHPC via automated deployment, documentation, testing and quality checks	HLRS	M12
------	------------------------	---	------	-----



Deliverables submitted:

D4.1	Initial report on workflow setups and workflow auditing	HLRS	M18
D4.2	Plan for development and deployment tool usage for workflows and software	HLRS/ CIN	M20



WP4: Progress update

T4.1 Workflow implementatin and optimization:

- D4.1 „Initial report on workflow setups and workflow auditing“ delivered in time
- All PD/SC workflows drafted
- SC6.e1 analyzed and optimized

T4.3 Development, deployment and quality assessment tools

- D4.2 „Plan for development and deployment tool usage for workflows and software“ delivered in time
- All codes made open access
- ChEESE metadata repository set up by CSIC
 - WP4 aim: link/store codes & workflows



What's next?

T4.1 Workflow implementatin and optimization:

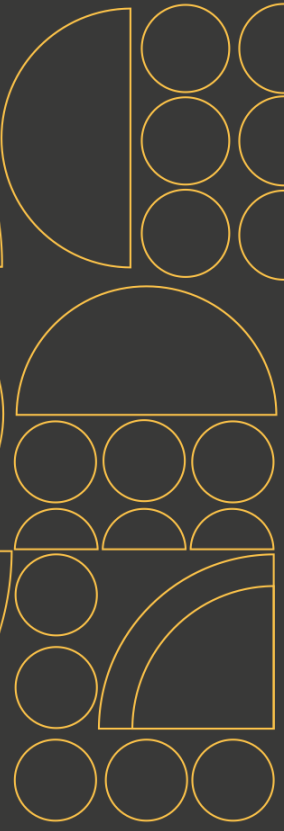
- Continue work on implementations
- Update workflow descriptions
- Analysis and optimization of workflows

T4.3 Development, deployment and quality assessment tools

- Provide Spack/Easybuild package descriptions for codes if not already available
- Link codes and workflow material in ChEESE metadata repository
- Check for possibilities to use CI/CD (also outside CASTIEL2)
- Apply tools to check codes for various programming errors



Key Discussion Items & Other comments



T4.1: Workflow overview for PDs/SCs

No	Name	category	UC	HPC systems	status				
					drafted	details	implemented	analyzed	optimized
PD1	Extreme-scale modeling of seismic hazards				x	x			
	SC1.1 Anthropogenic noise and seismicity, wave propagation at higher frequencies in Rade d'Hyères	capability		Leonardo	x				
	SC1.2 Next generation of earthquake simulators for the Hellenic arc and Iceland	capacity		Leonardo, LUMI	x				
	SC1.3 Physics-based PSHA and urgent UQ for Iceland	capacity	x	MN5	x	x			
PD2	High-resolution seismic inversion and tomography				x	x			
	SC2.1 Seismic tomography of the Adriatic-Ionian region	capability		Leonardo	x				
PD3					x				
	SC3.1 The Global Tsunami Hazard Map	capacity		Leonardo	x				
PD4	Complex multi-source tsunami modeling				x	x			
	SC4.1 Multi-source local high-resolution scenario with emphasis on the 2022 Tonga case	capcability	x	Leonardo	x				
PD5	Ensemble-based volcanic dispersal across multiple scales				x	x			
	SC5.1 Towards the European tephra hazard map	capacity		MN5	x	x			
	SC5.2 Volcanic dispersal at local scale: a reanalysis of the 2021 La Palma case	capacity	x	MN5	x	x			
PD6					x	x			
	SC6.1 Urgent high-resolution, 3D multiphase flow simulation of phreatic eruptions at Vulcano	capability	x	LUMI	x	x	x	x	x
	SC6.2 Long-term probabilistic hazard maps for phreatic eruptions at Vulcano island	capacity		LUMI	x	x			
PD7					x				
	SC7.1 MHD turbulence closure models in planetary cores using AI	capacity		MN5, LUMI, VEGA	x				
PD8					x				
	SC8.1 Magmatic systems and induced landslides: the Etna Case	capability		LUMI	x				
	SC8.2 3D evolution of tectonic deformation as driver of seismic hazard in the Hellenic arc	capacity		LUMI	x				
PD9	Glacial outburst floods				x	x			
	SC9.1 Eastern Skaftá cauldron GLOF simulation, Iceland	capacity		LUMI	x				
	SC9.2 Calving glacier hazard in Greenland	capacity		LUMI	x				

To be continued by PDs/SCs:

Create/update detailed workflow descriptions as implementations progress.

Are you interested in workflow auditing and optimization work?

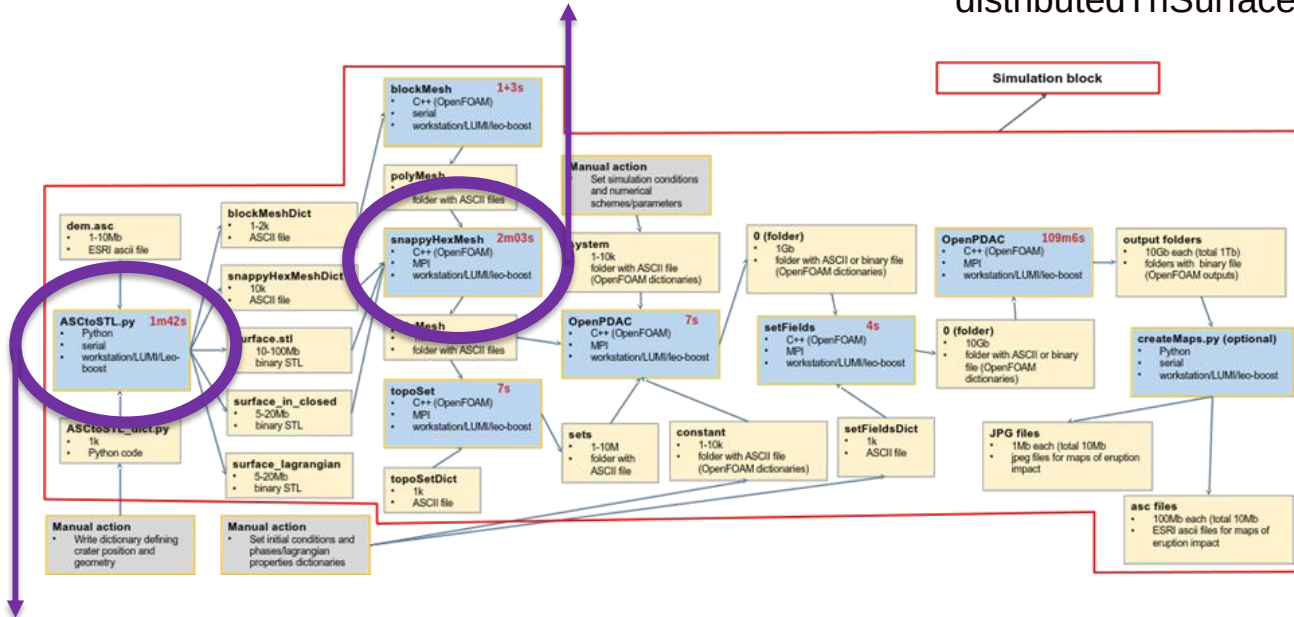
What about resources for workflow audits?

Collaboration needed with WP5 & WP6

T4.1: SC6.1 Auditing and Optimization (small testcase)

Parallel I/O issues due to individual reads from all MPI processes

- Use collated I/O in OpenFoam!
- switching from triSurfaceMesh to distributedTriSurfaceMesh



Serial Python script timing breakdown:

- 40% SciPy (outside scope of ChEESE-2P)
- 60% ChEESE-2P code parts

Optimized parts using NUMBA
→ 26% speedup of this part

T4.3: Development, Deployment and Quality Assessment

1 Development

- Moved ChEESE codes to (openly accesible) development platforms: github, gitlab



2 Deployment

- Ongoing interaction with CASTIEL-2 about the design of the deployment infrastructure
- Provide packages for spack, easybuild,... to allow easy installation / deployment of ChEESE codes
- Use ChEESE metadata repository to ease deployment of codes and workflows



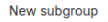
3 Quality Assessment

- Started collecting of EOSC-synergy code quality badges for ChEESE codes
- Identify potential correctness tools to apply to ChEESE codes
- Obtain test coverage for codes



ChEESE metadata repository

 **ChEESE** 

The ChEESE CoE project embarks on its second phase, addressing the multiple challenges posed by geohazards with the help of exascale computers. The project primarily addresses European institutions responsible for operational monitoring networks but also at hardware developers, industrial stakeholders, public institutions and HPC centers.

Recent activity
Last 30 days

Merge requests created
0

Issues created
0

Members added
4

Subgroups and projects

Shared projects

Inactive

 Search (3 character minimum) 

Name  

>  A AUDITS  Owner	 1  0  1 
>  C CODES  Owner	 11  0  1 
>  M MINI-APP  Owner	 7  0  1 
>  S SIMULATION CASES  Owner	 0  15  1 
>  W WORKFLOWS  Owner	 0  9  1 
 R REPOSITORY DOCUMENTATION 	 1 5 days ago

<https://gitlab.com/cheese5311126>

T4.3: Code deployment tool usage plan

Code	Spack	EasyBuild	Container	other
SeisSol	✓	✗	✓	✗
SPECFEM3D	✓	✗	✓	✗
ExaHyPE	✓	✗	✓	✗
Tandem	✓	✗	✓	✗
XSHELLS				
HySEA	✓	✗	✓	✗
FALL3D	✓	✗	✓	✗
OpenPDAC	✗	✓	✓	✗
LaMEM				✓
pTatin3D				
Elmer/Ice	✓		✓	✓ (Nix)

Intended software deployment tools for the ChEES-2P codes showing available (green check, ✓), planned (orange check, ✓), and no (red cross, ✗).



T4.3: Intended application of code quality tools

Compiler:

- Enable, check, and fix Compiler warnings:
e.g., gcc -Wall
<https://gcc.gnu.org/onlinedocs/gcc/Warning-Options.html>
- Use sanitizers for runtime checks:
e.g., -fsanitize=address,leak,undefined,bounds
https://www.osc.edu/resources/getting_started/howto/howto_use_address_sanitizer

Clang-tidy (<https://clang.llvm.org/extra/clang-tidy/>):

- C++ tool that performs static code analysis to identify potential programming errors

MUST (<https://itc.rwth-aachen.de/must/>):

- Tool that checks for a broad range of erroneous usages of MPI during program execution

Code	Programming language	Tool
SeisSol	C++	compiler, clang-tidy
SPECFEM3D	Fortran 2008/C++	compiler, clang-tidy
ExaHyPE	C++	compiler, clang-tidy
Tandem	c++	compiler, clang-tidy
XSHELLS	C++	compiler, clang-tidy
HySEA	C++/CUDA	compiler, clang-tidy
FALL3D	Fortran 2008	compiler
OpenPDAC	C++	compiler, clang-tidy, MUST
LaMEM	C++	compiler, clang-tidy
pTatin3D	C	compiler
Elmer/Ice	Fortran 2008	compiler