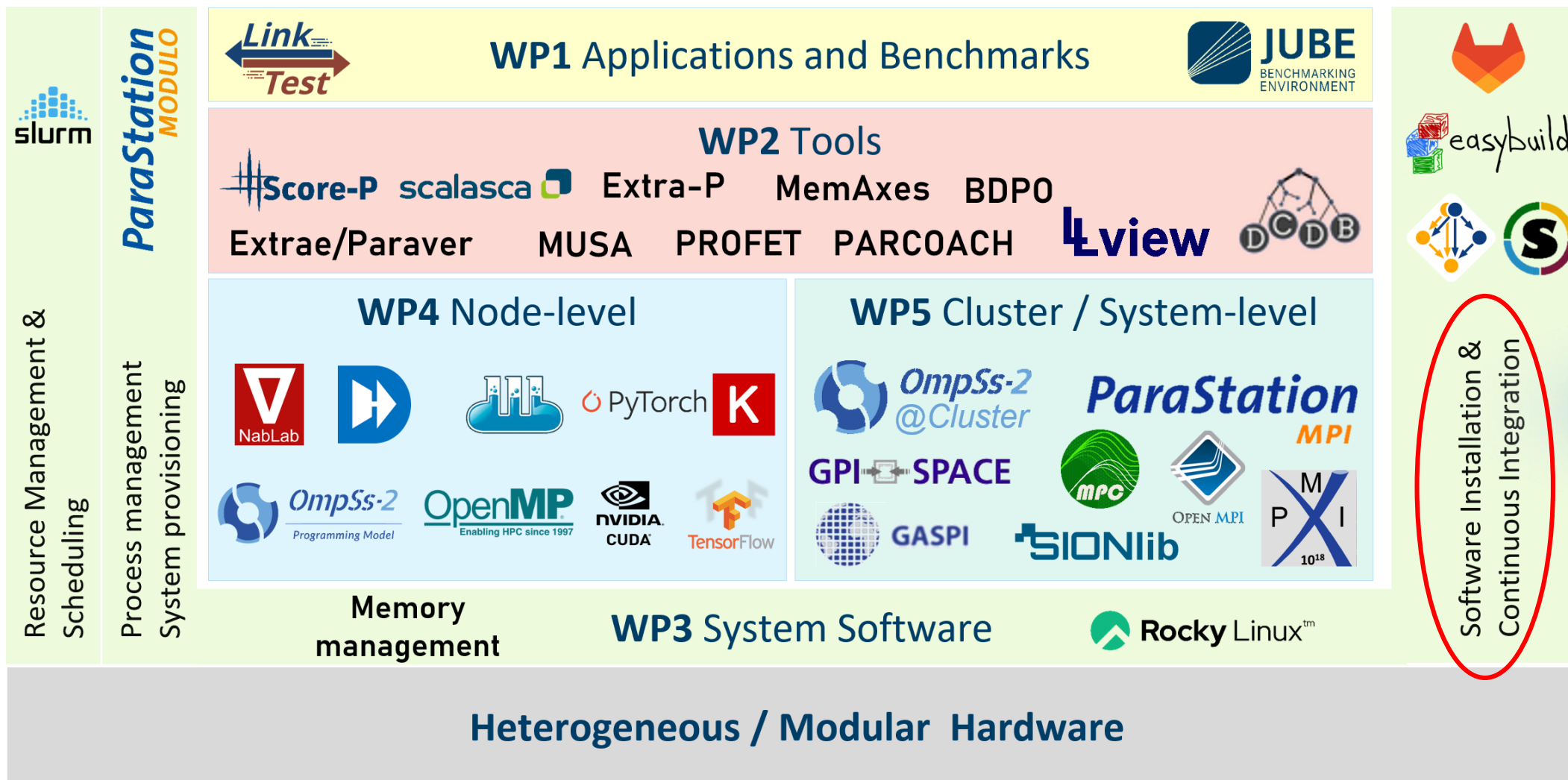# The DEEP-SEA Project

**Hans-Christian Hoppe, Estela Suarez**

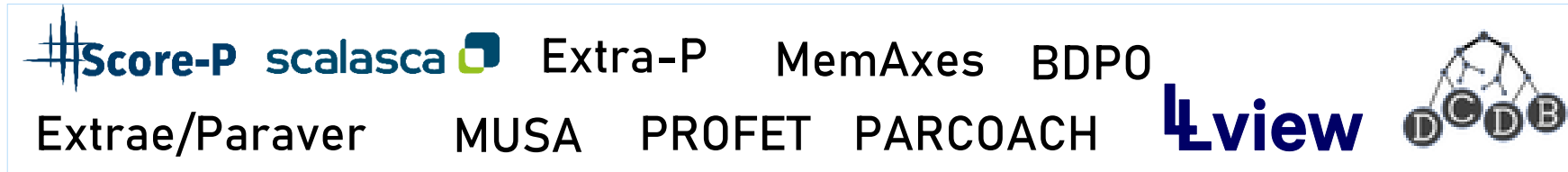**Final SEA Projects Workshop, LRZ Garching, January 16, 2024**

# DEEP-SEA SW Stack



Public release at https://gitlab.jsc.fz-juelich.de/deep-sea/wp3/software/easybuild-repository-deep-sea

# DEEP-SEA: Optimisation Cycles

Bewildering variety of SW tools available to HPC SW developers for analysis and optimisation – in DEEP-SEA alone, these:



Optimisation cycles encapsulate (complex) tool workflows for *specific purposes*

- – For instance, assessing load balance or optimising energy use

They guide SW developers and make it easier to achieve specific goals

- – Rather then using specific tools

You'll hear much more about Optimisation Cycles in the 13:30 presentation

# DEEP-SEA & Heterogeneous Memory

- Examples…
  - DDR DRAM
  - Scratchpad (Embedded systems-on-chip, GPUs)
  - High bandwidth memory (Intel Xeon Phi, GPUs)
  - Byte addressable non-volatile memory (HP's Machine, Intel Optane)
  - Compute Express Link (CXL): high-speed interface to accelerators and memory modules
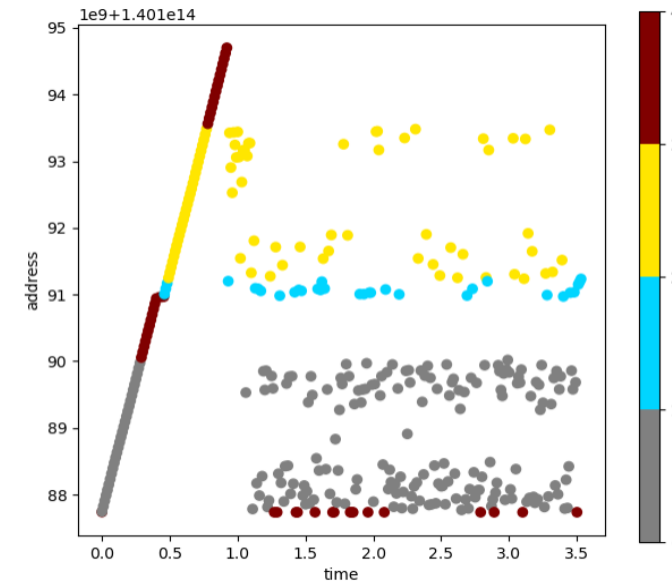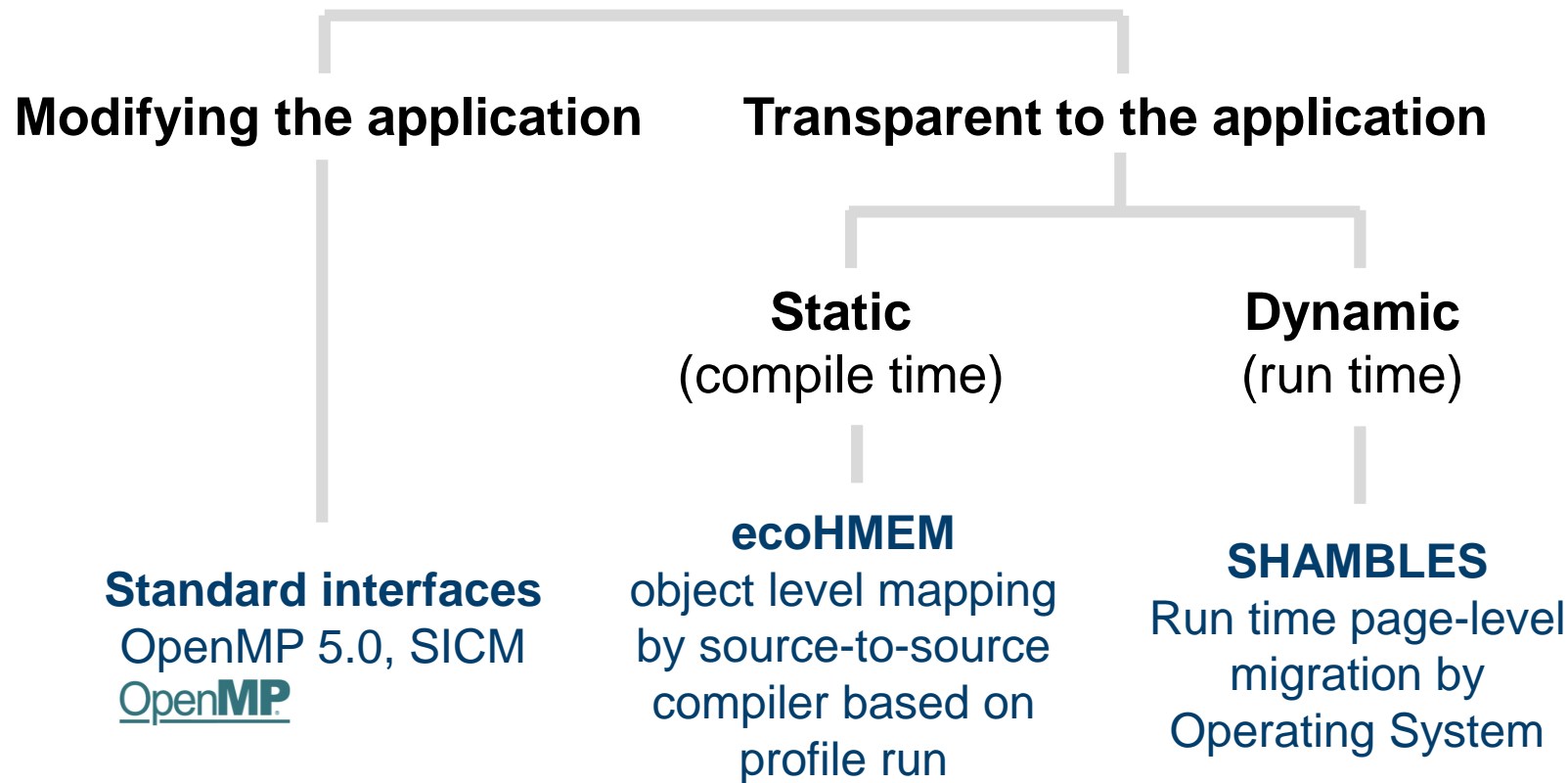


[1]



**Memory hierarchy**
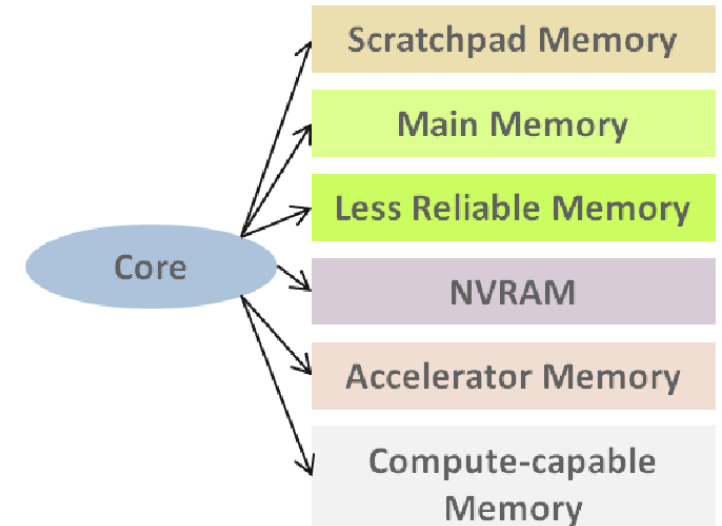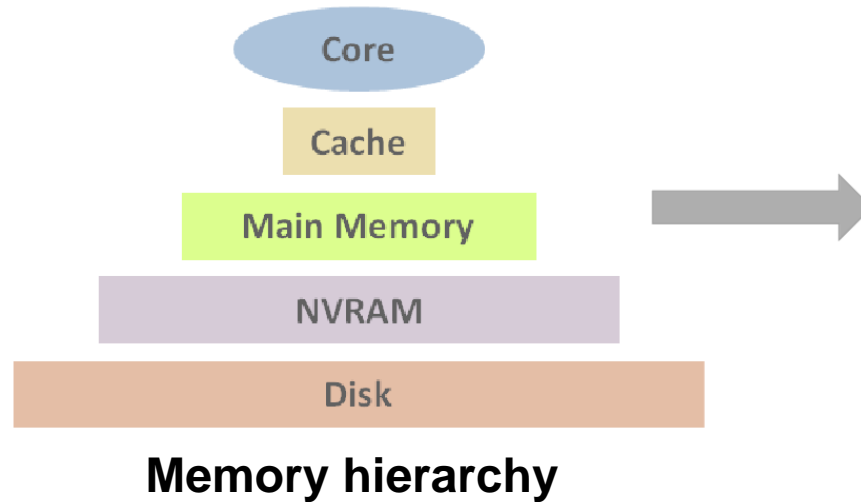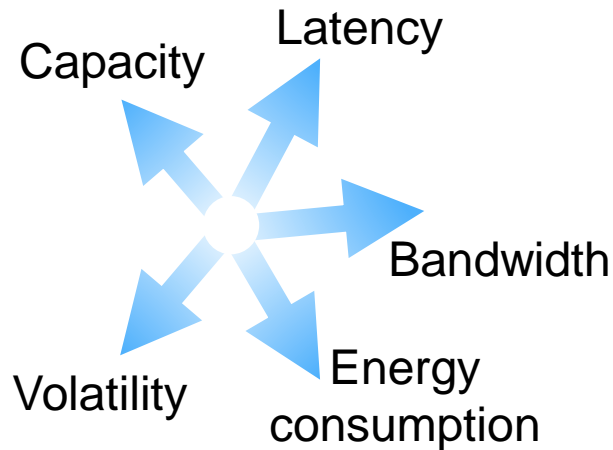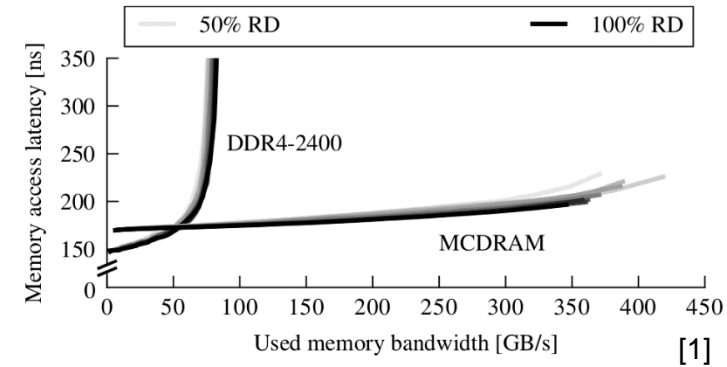
**Explicitly managed**

[1] Milan Radulovic et al. PROFET: Modeling System Performance and Energy Without Simulating the CPU. ACM SIGMETRICS 2019

# DEEP-SEA & Memory Tools

– How much, if any, do the applications need to be modified?

– Which layer manages the memory? When?

– How much can the applications benefit?

**Modifying the application**     **Transparent to the application**

**Static**
(compile time)

**Dynamic**
(run time)

**Standard interfaces**
OpenMP 5.0, SICM

**Open**MP

**ecoHMEM**
object level mapping
by source-to-source
compiler based on
profile run

**SHAMBLES**
Run time page-level
migration by
Operating System



**SHAMBLES scatter plot
example for sparse kernel**

# DEEP-SEA & Malleability
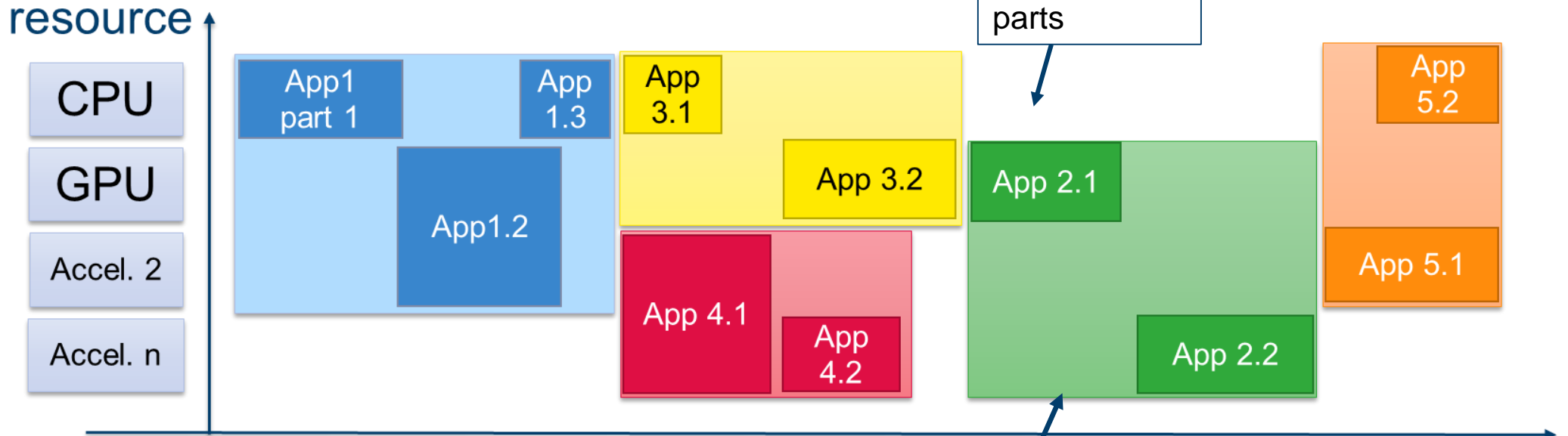


- Usual HPC workload resource reservation (constant # cores or nodes over time)
- Actual use of resources varies over time (yellow curve)
- Workload is able to use more resources in certain phases (arrow)
- Ideal resource allocation for the workload
- Malleable applications
  - Release resources not required
  - Acquire more resources if advantageous
- Resource management systems & fabrics must support this
- Change in # of nodes do require data redistribution in the workload
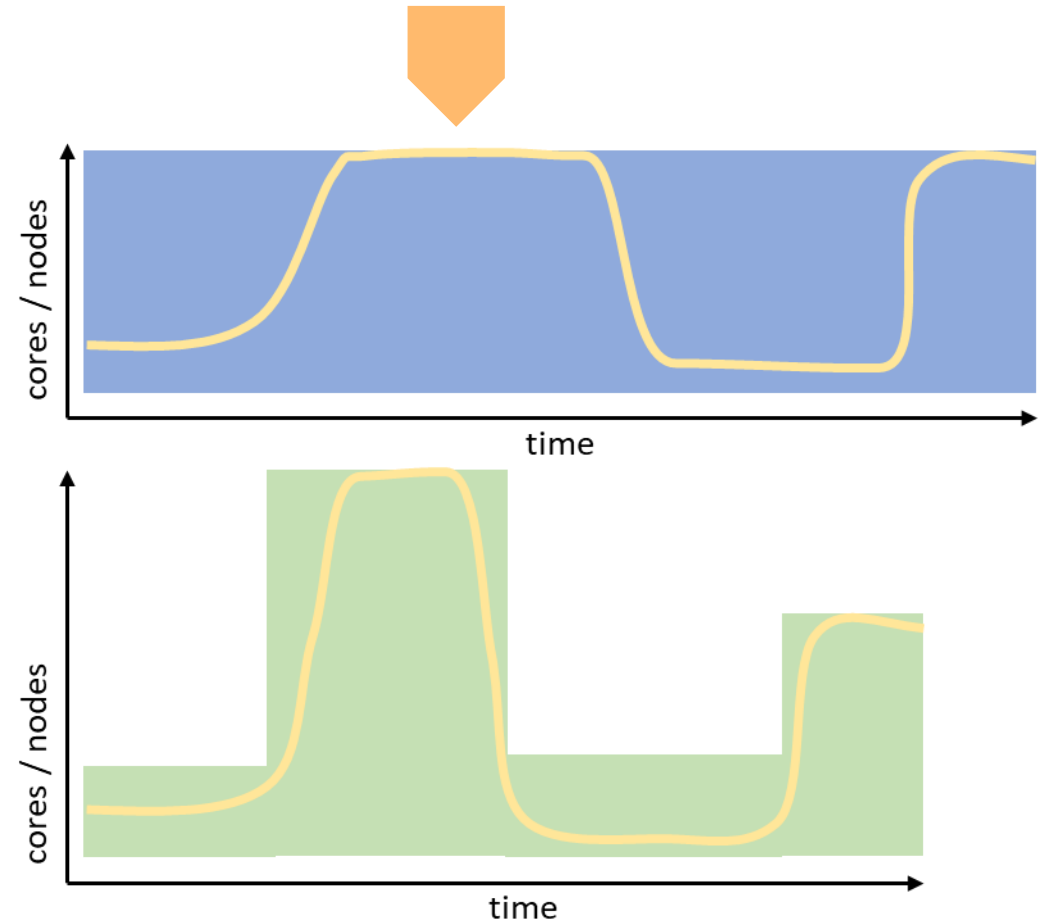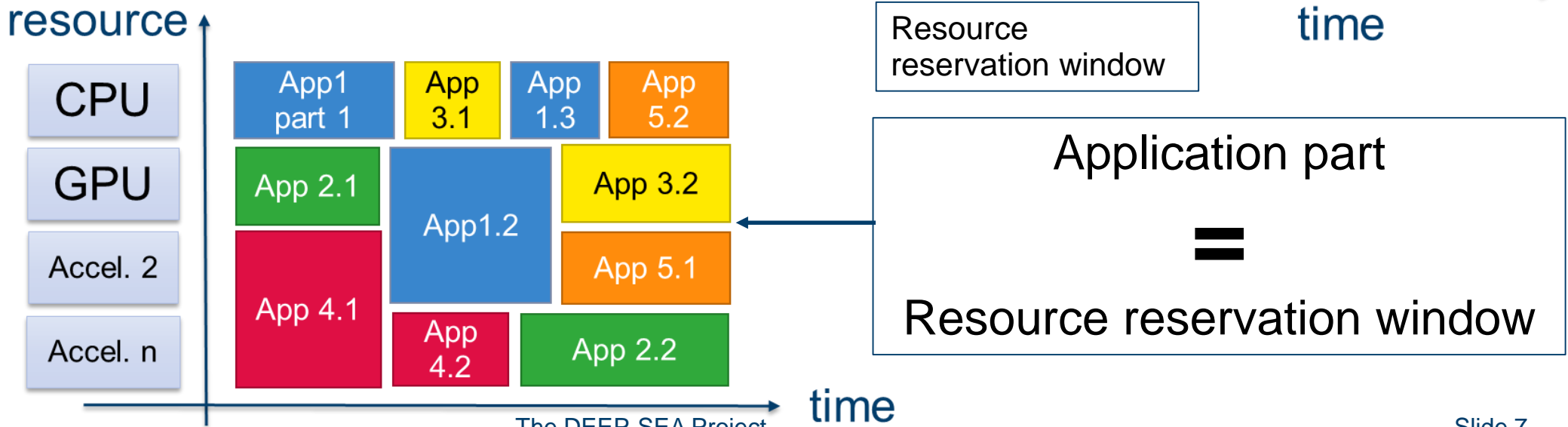
# DEEP-SEA & Scheduling



The DEEP-SEA Project
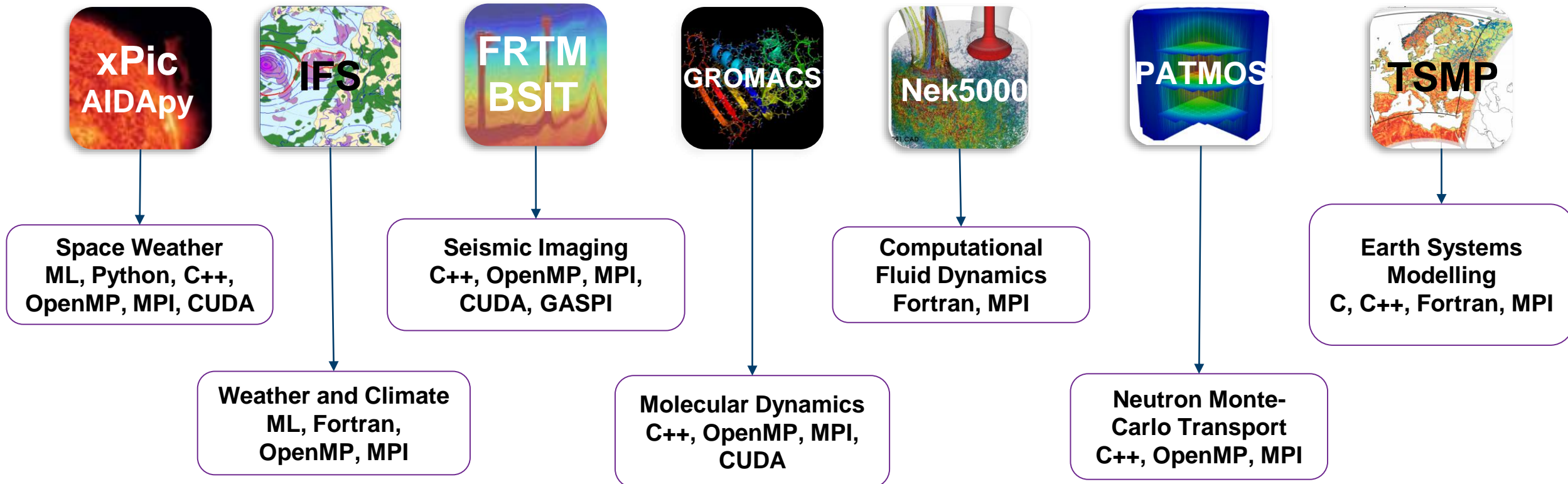
# DEEP-SEA Co-Design Applications



**xPic** / **AIDApy**

Space Weather
ML, Python, C++,
OpenMP, MPI, CUDA

**IFS**

Weather and Climate
ML, Fortran,
OpenMP, MPI

**FRTM BSIT**

Seismic Imaging
C++, OpenMP, MPI,
CUDA, GASPI

**GROMACS**

Molecular Dynamics
C++, OpenMP, MPI,
CUDA

**Nek5000**

Computational
Fluid Dynamics
Fortran, MPI

**PATMOS**

Neutron Monte-
Carlo Transport
C++, OpenMP, MPI

**TSMP**

Earth Systems
Modelling
C, C++, Fortran, MPI

# Acknowledgements

The DEEP-SEA Project has received funding from the European High-Performance Computing Joint Undertaking (EuroHPC JU) under grant agreement n° 955776.  The EuroHPC JU  receives support from the European Union's Horizon 2020 research and innovation programme and Germany, France, Spain, Greece, Belgium, Sweden, and Switzerland.

SPONSORED BY THE

The DEEP-SEA Project