The DEEP/-ER architecture: a modular approach to extreme-scale computing

Estela Suarez
Jülich Supercomputing Centre (JSC)
Germany

06.07.2017

The research leading to these results has received funding from the European Community (*FP7/2007-2013 and H2020*) under *Grant Agreements #*287530, #610476, and #754304

The DEEP projects
DEEP, DEEP-ER and DEEP-EST

EU-Exascale
27 partners
Total budge
EU-funding:
Nov 2011 –

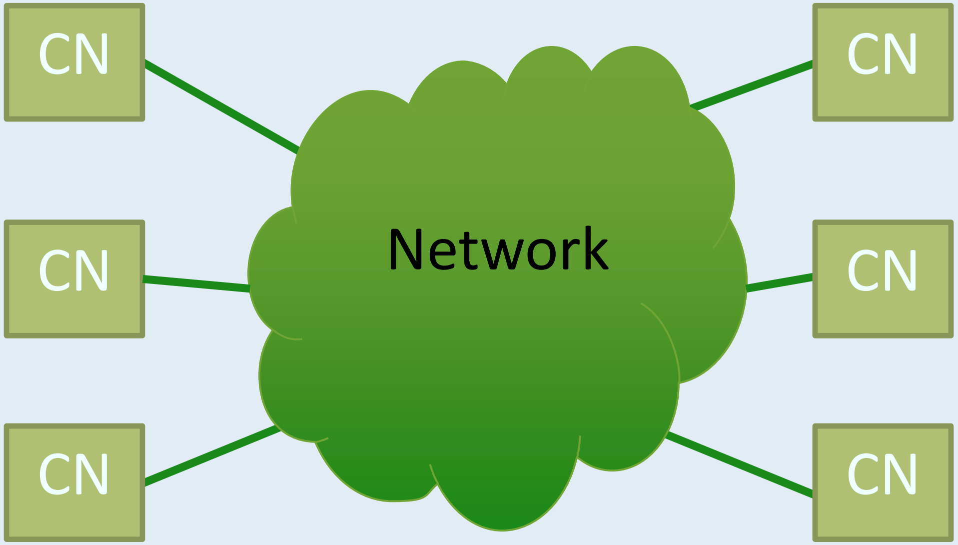
Both combi
-Hardware
-Software
-Application
in a strong c

DEEP:
Cluster-Booster Architecture + software environment

DEEP-ER:
I/O + resiliency

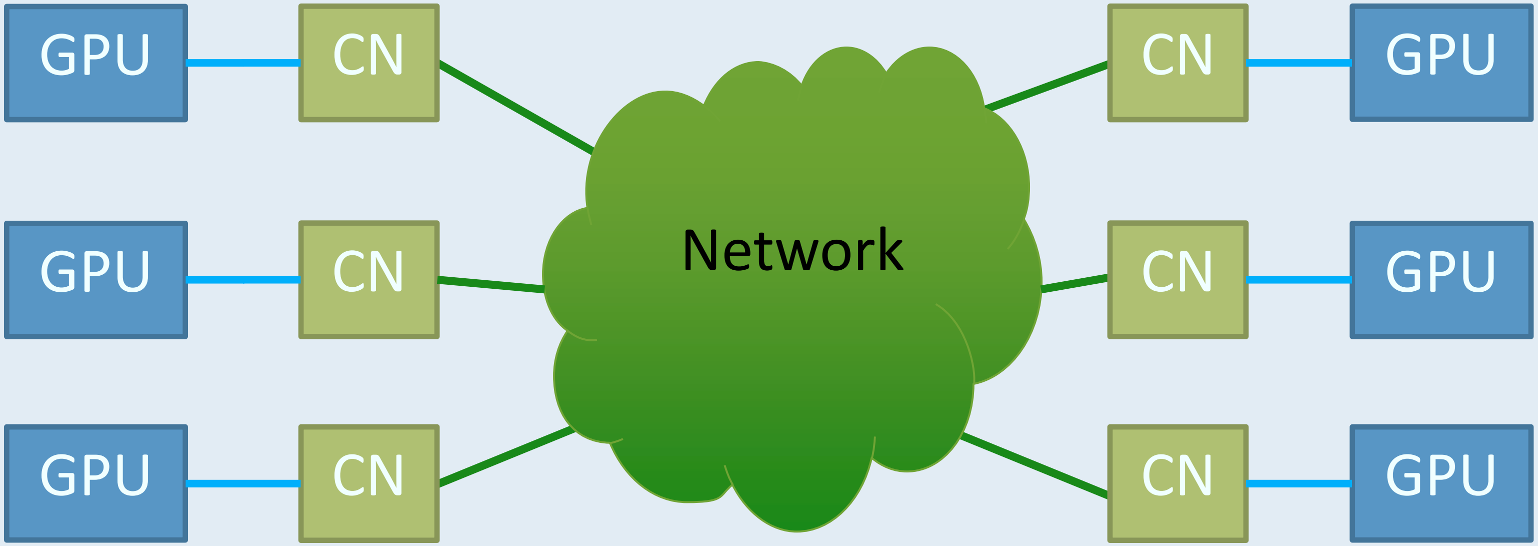
DEEP-EST:
Modular Supercomputing

Homogeneous cluster



- Cluster Nodes: **general purpose** (multi-core) processor technology
  - Same processor characteristics in all nodes
- Single high-speed network connecting them all
- Good concept but limited efficiency for selected HPC applications

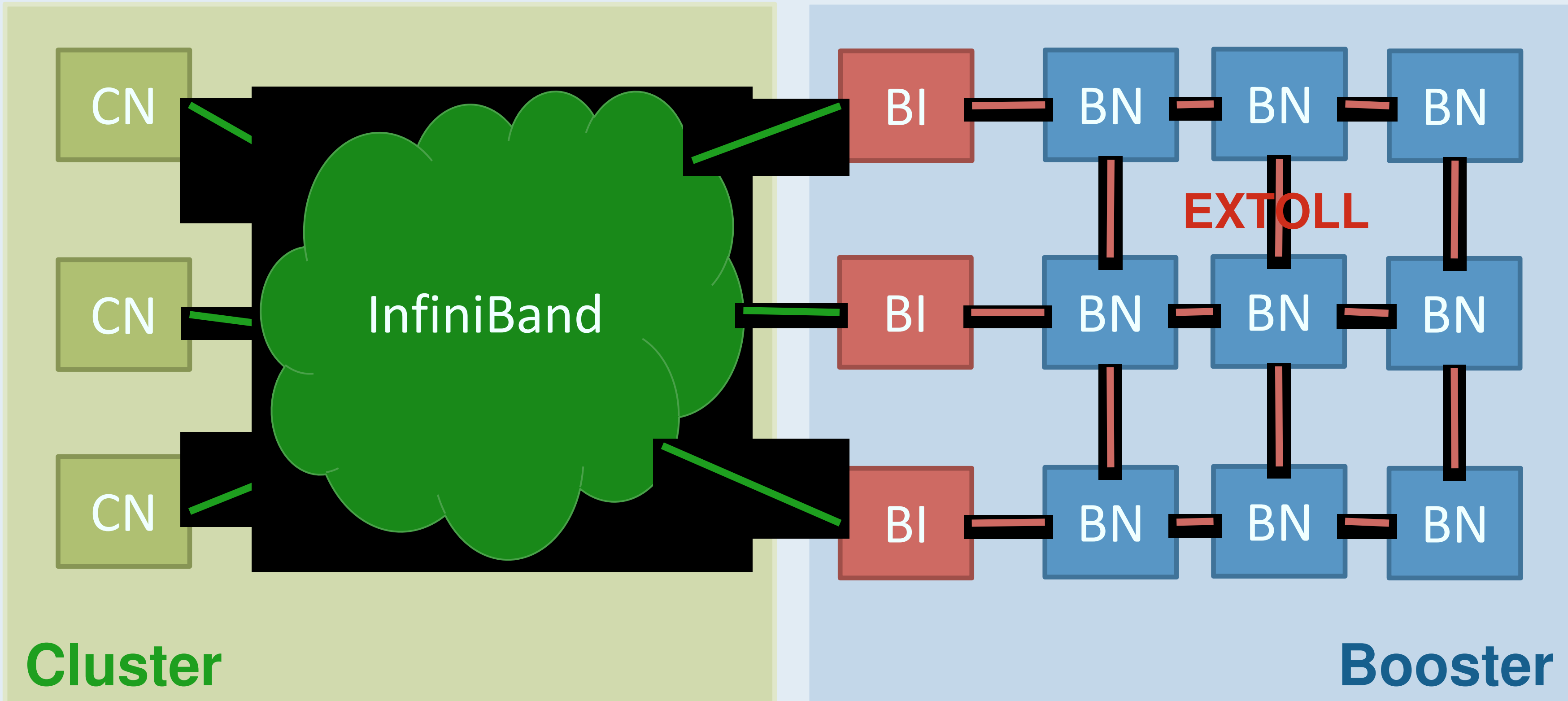
“Standard” heterogeneity



- Flat topology
- Simple management of resources

- Static assignment of accelerators to CPUs
- Accelerators cannot act autonomously

Cluster-Booster architecture

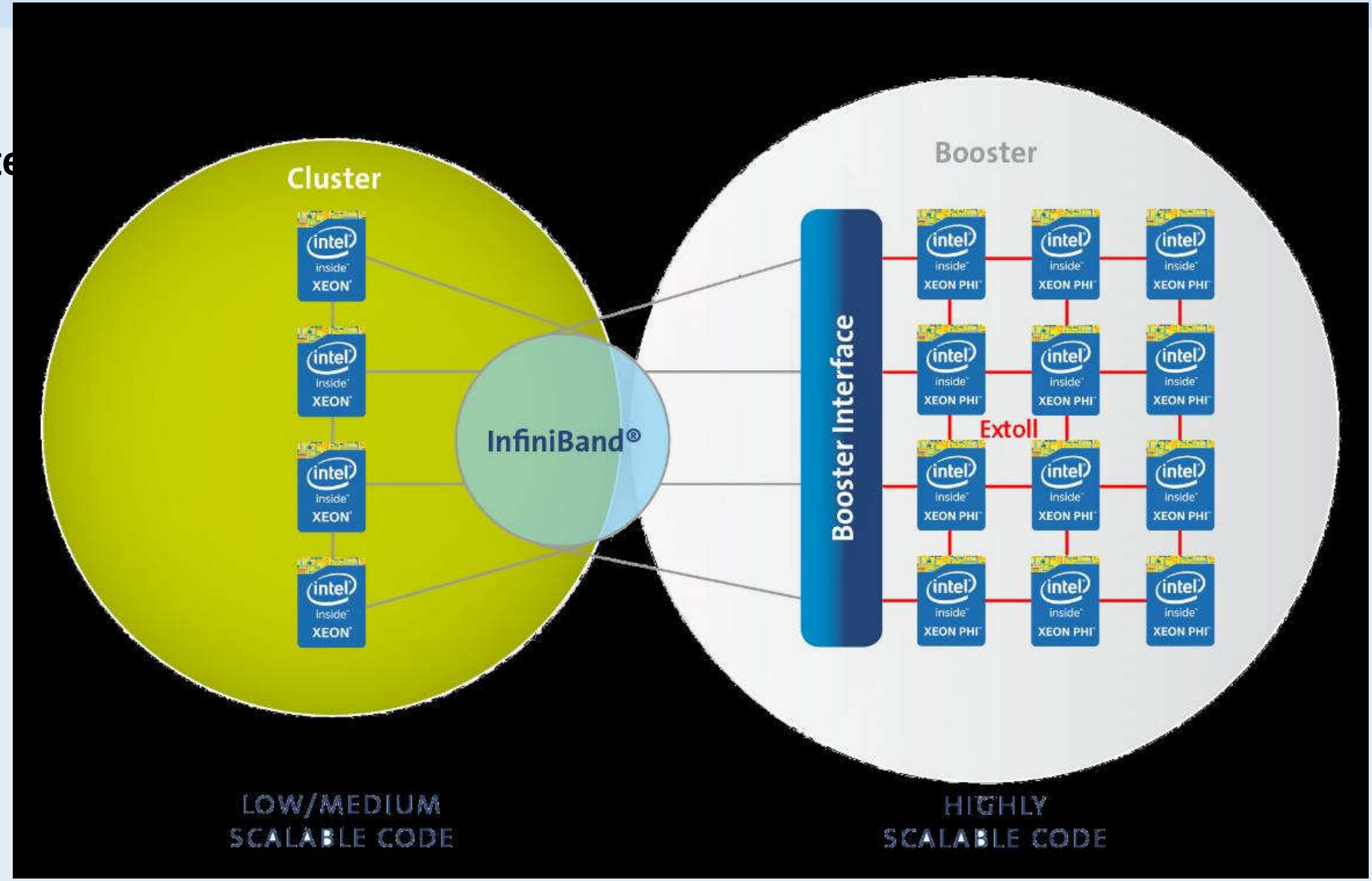


- Flexible assignment of resources (CPUs, accelerators)
- Direct communication between accelerators
- “Offload” of large and complex parts of applications

DEEP Architecture

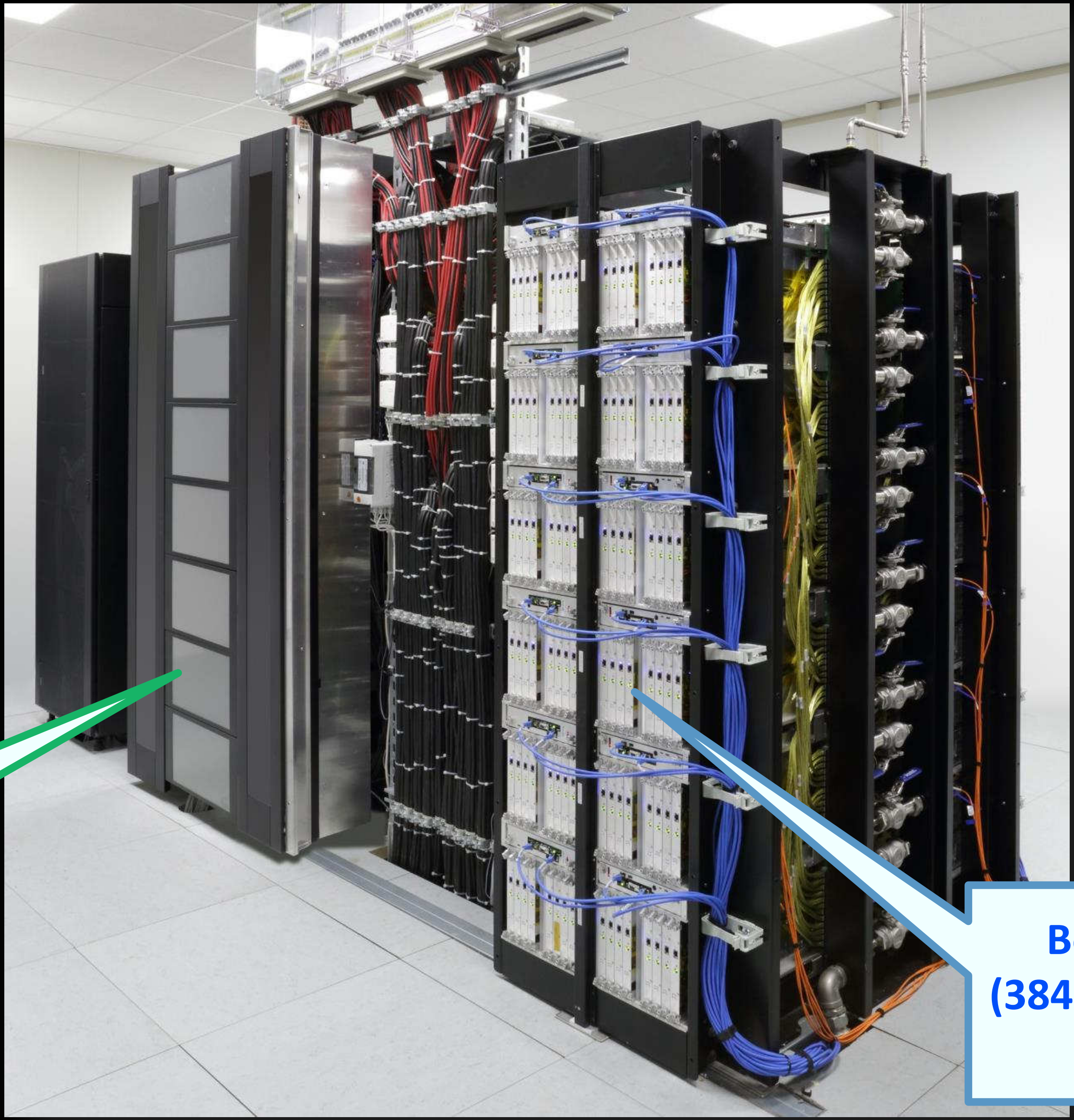
Inte

PhiTM



DEEP Prototype

- Installed at JSC
- 1,5 racks
- 500 TFlop/s peak perf.
- 3.5 GFlop/s/W
- Water cooled

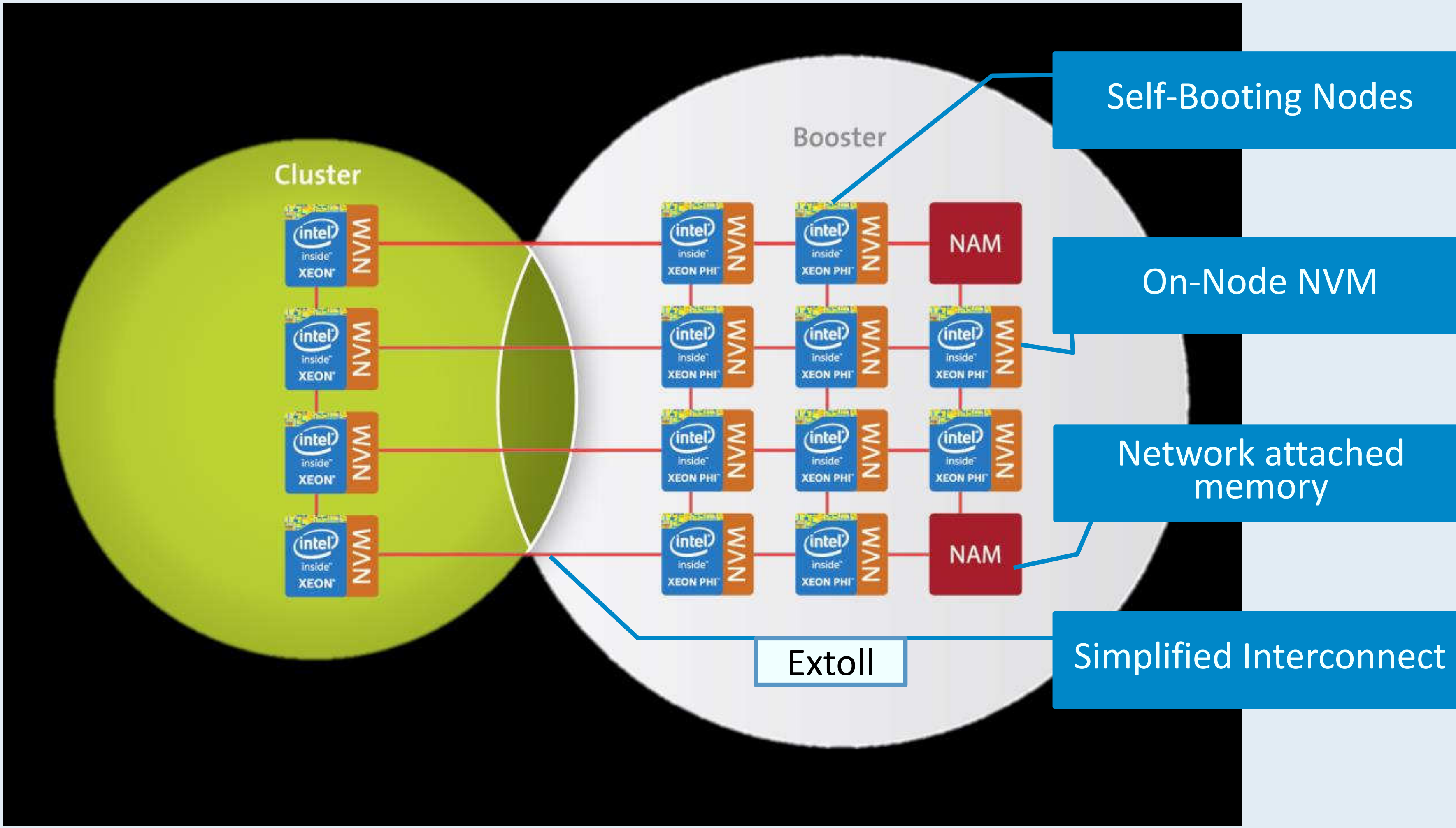


Cluster (128 Xeon)

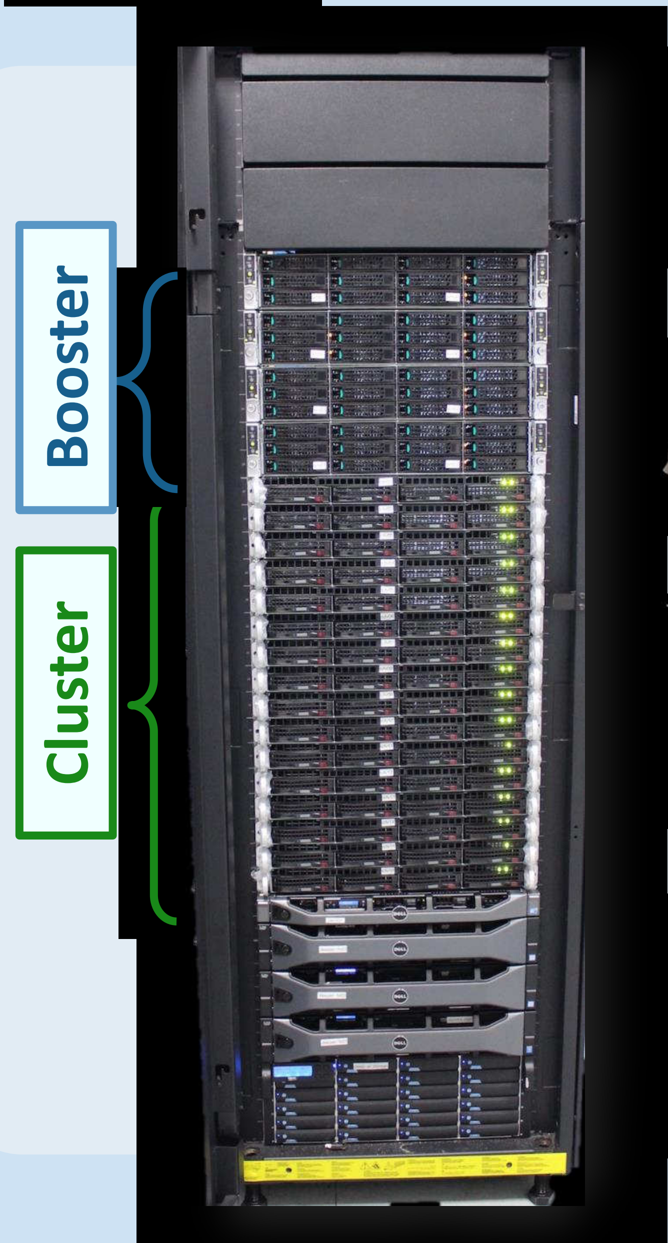
Booster (384 Xeon Phi KNC)

DEEP-ER Architecture

Innovation

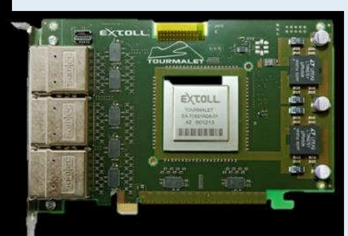


DEEP-ER prototype



Intel Xeon Phi (KNL)

NVMe



EXTOLL Tourmalet

NAM

Booster

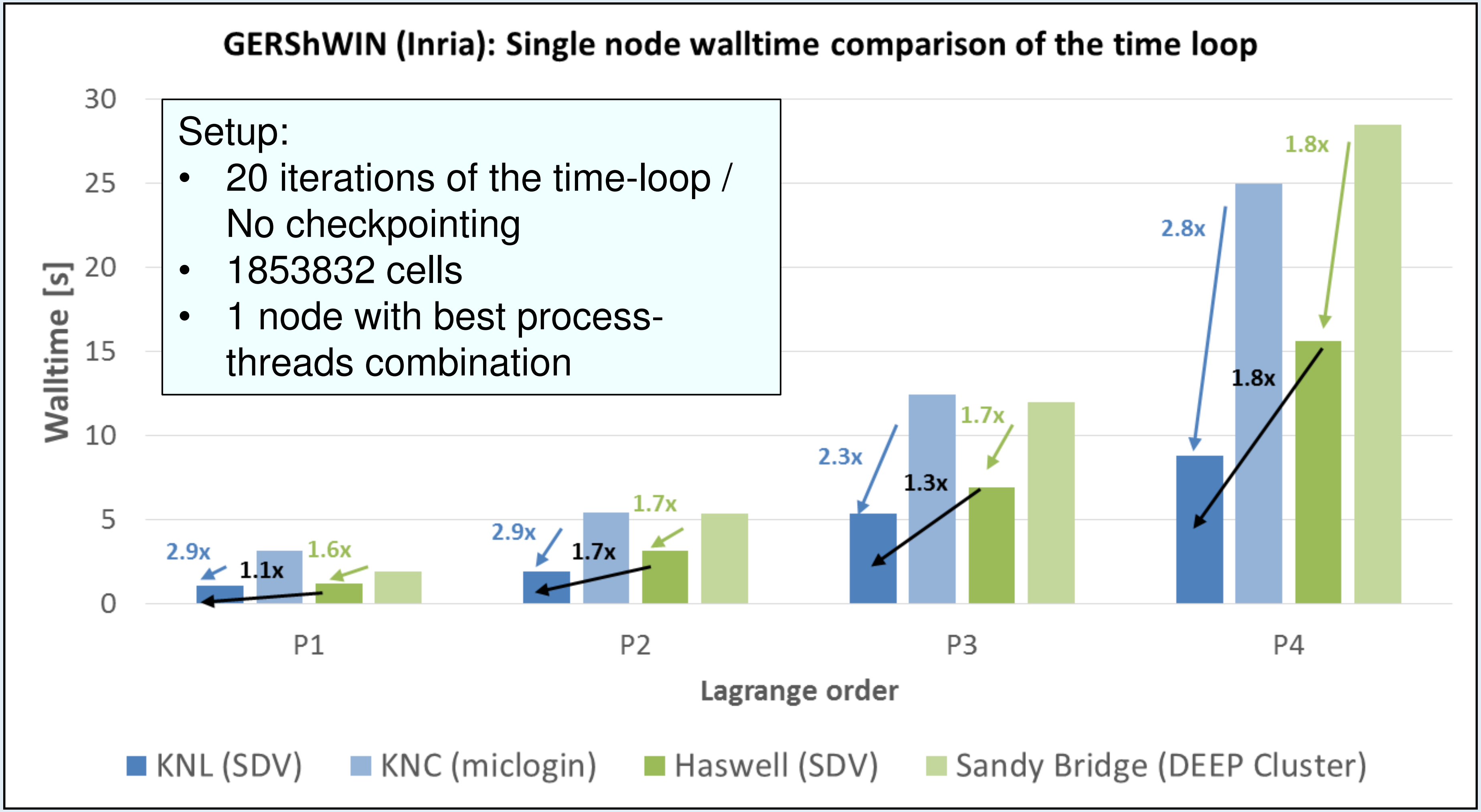
- 8 Intel Xeon Phi (KNL) 7210X nodes (16+96GB)
- 400 GB NVMe
- EXTOLL Tourmalet (ASIC) 100 Gb/s per link
- 2x NAM devices

Cluster

- 16 dual-socket Intel Xeon E5-2680v3 (Haswell)
- 128 GB DRAM
- 400 GB NVMe
- EXTOLL Tourmalet

DEEP vs. DEEP-ER

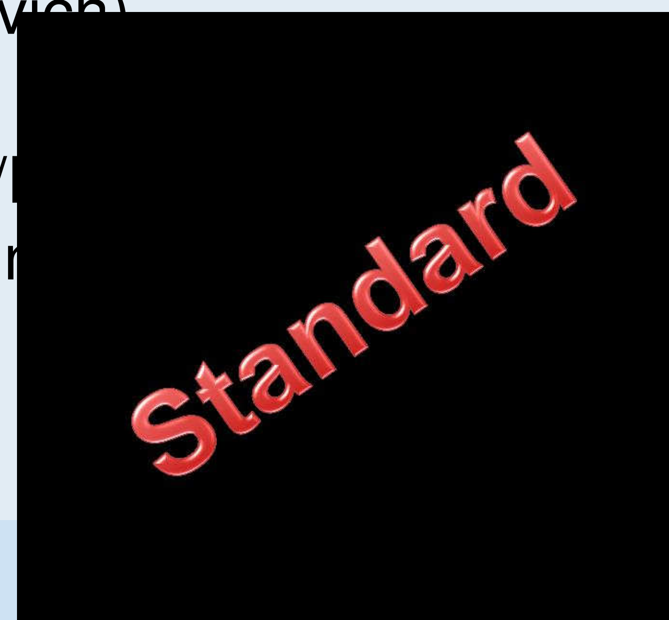
Application performance comparison



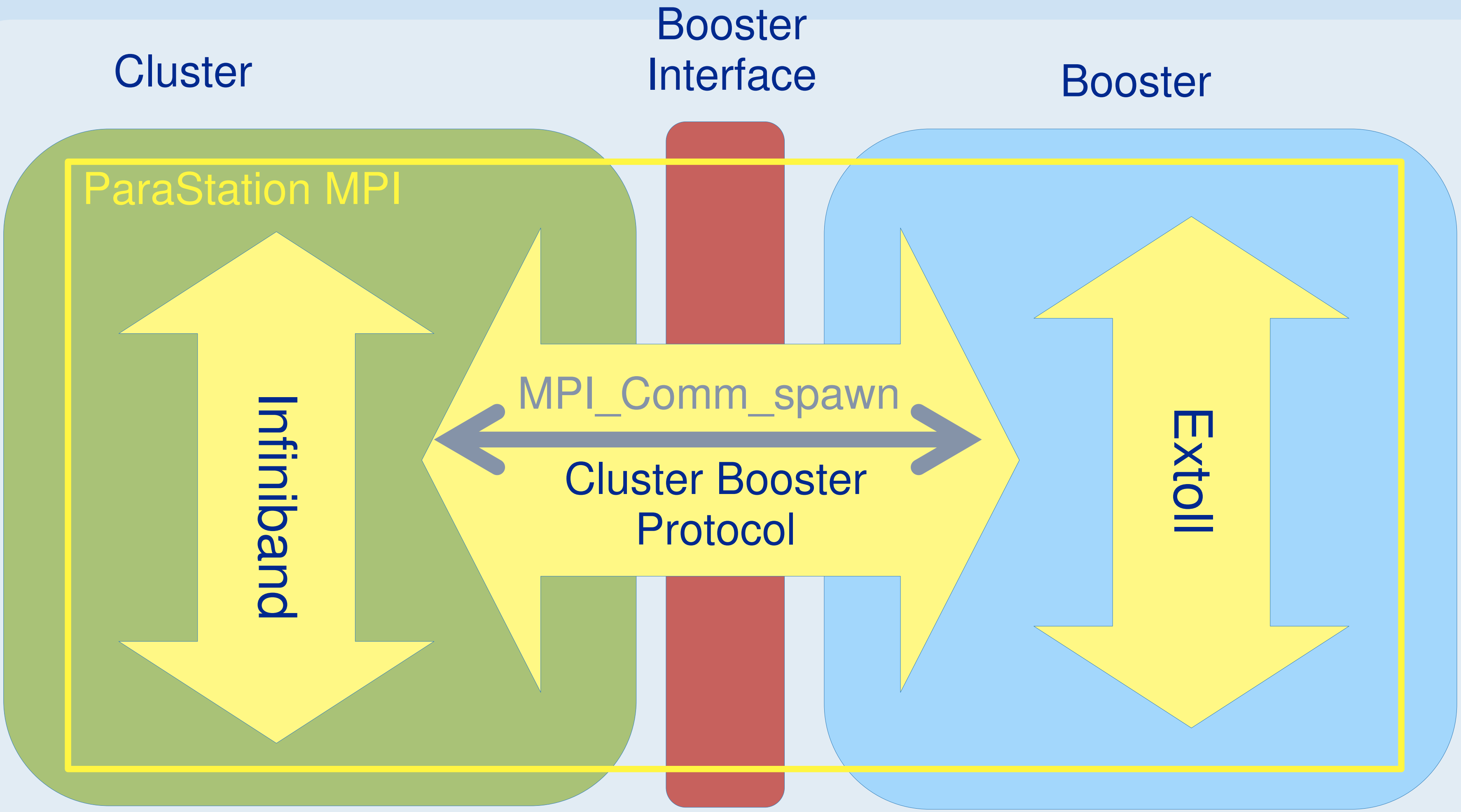
SOFTWARE ENVIRONMENT

Software environment

- **Scheduler**: Torque/Maui → future moving to SLURM
- **Filesystem**: BeeGFS
- **Compilers**: Intel, gcc, PGI
- **Debuggers**: Intel Inspector (threading, memory), TotalView (source code, memory debugger)
- **Programming**: ParaStation MPI [redacted] OpenMP, OmpSs
- **Performance analysis tools**: Extrae/[redacted] Scalasca, Intel Advisor, Intel, VTu[redacted]
- **Libraries**: SIONlib, SCR, E10, HDF5, [redacted] netcdf, PETSc …



Programming environment



OmpSs on top of MPI provides pragmas to ease the offload process

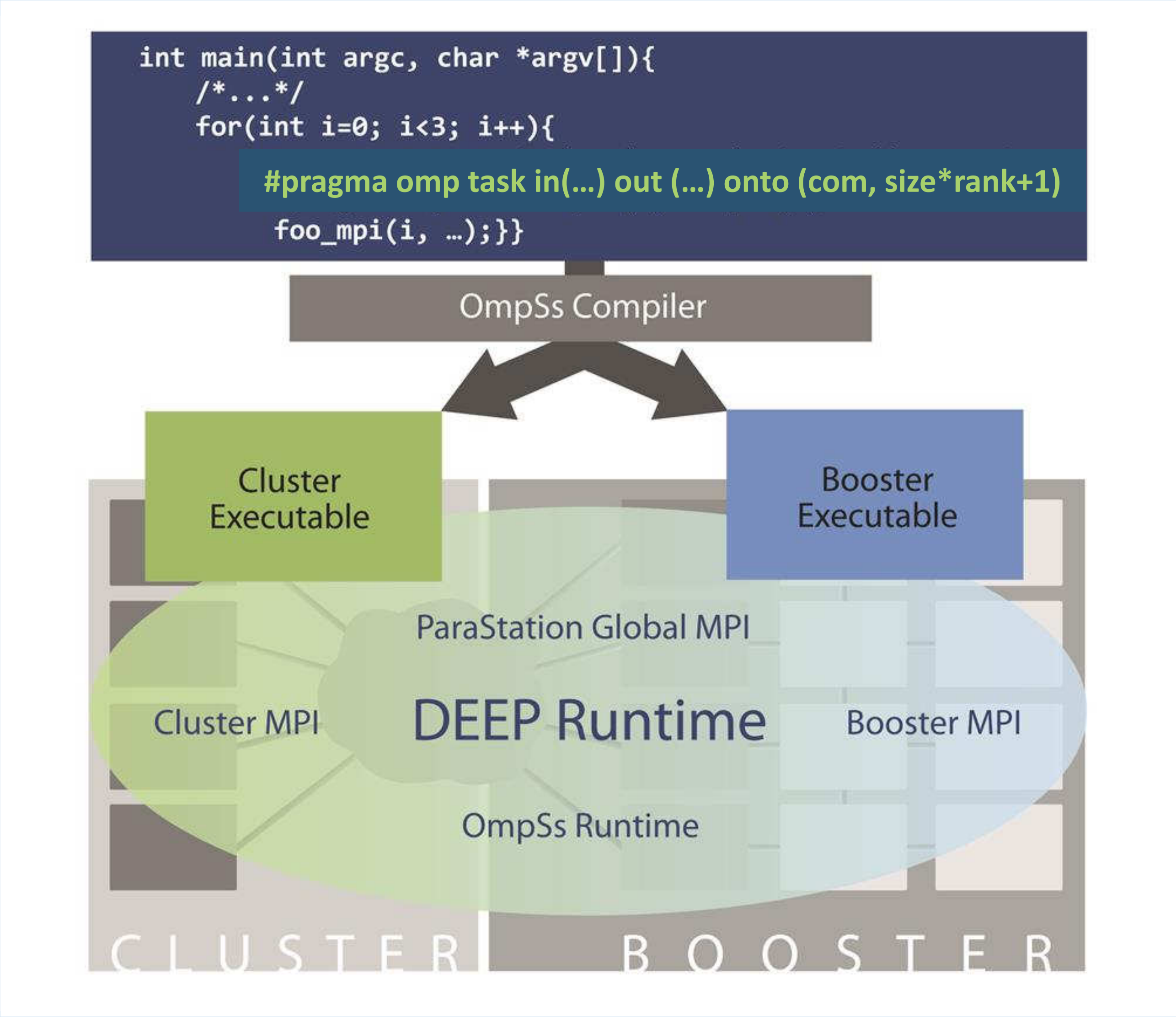
Application running on DEEP

Source code

Compiler

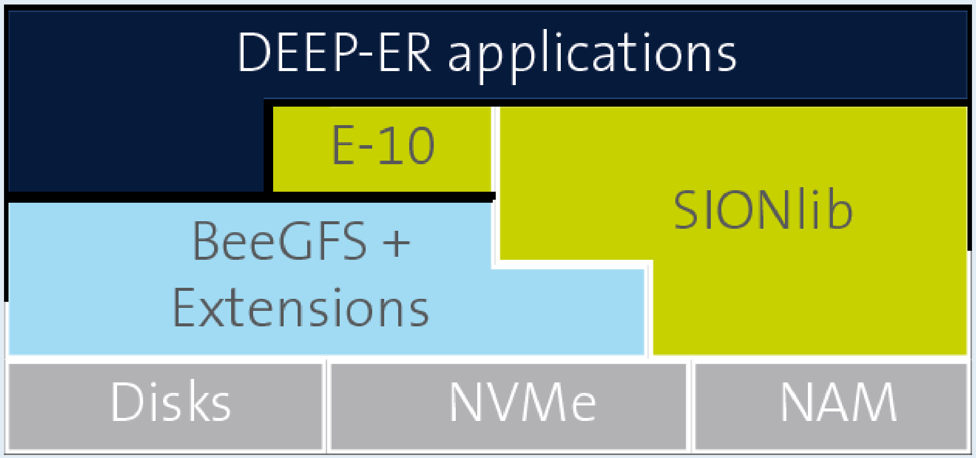
Application binaries

DEEP Runtime



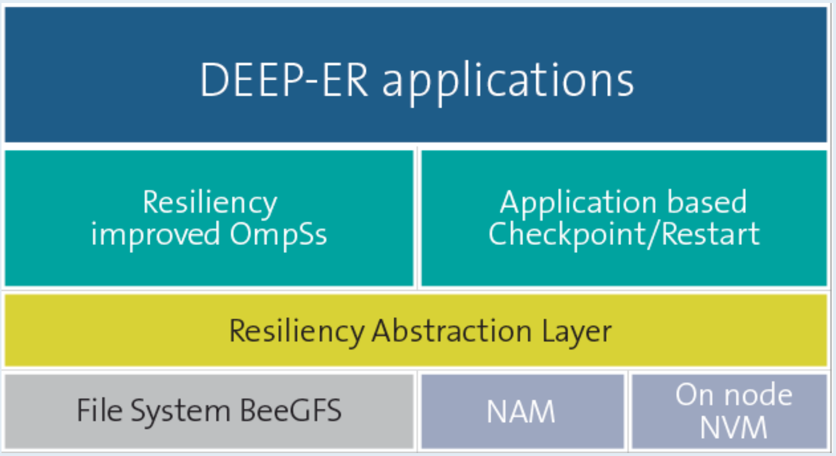
DEEP-ER I/O and resiliency

- I/O Software architecture



  - **BeeGFS** (parallel FS)
  - **SIONlib** (I/O concentrator)
  - **Exascale10** (collective I/O)

- Resiliency SW architecture



  - **SCR** (checkpointing handling)
  - **ParaStation MPI** (process CP)
  - **OmpSs** (task checkpointing)

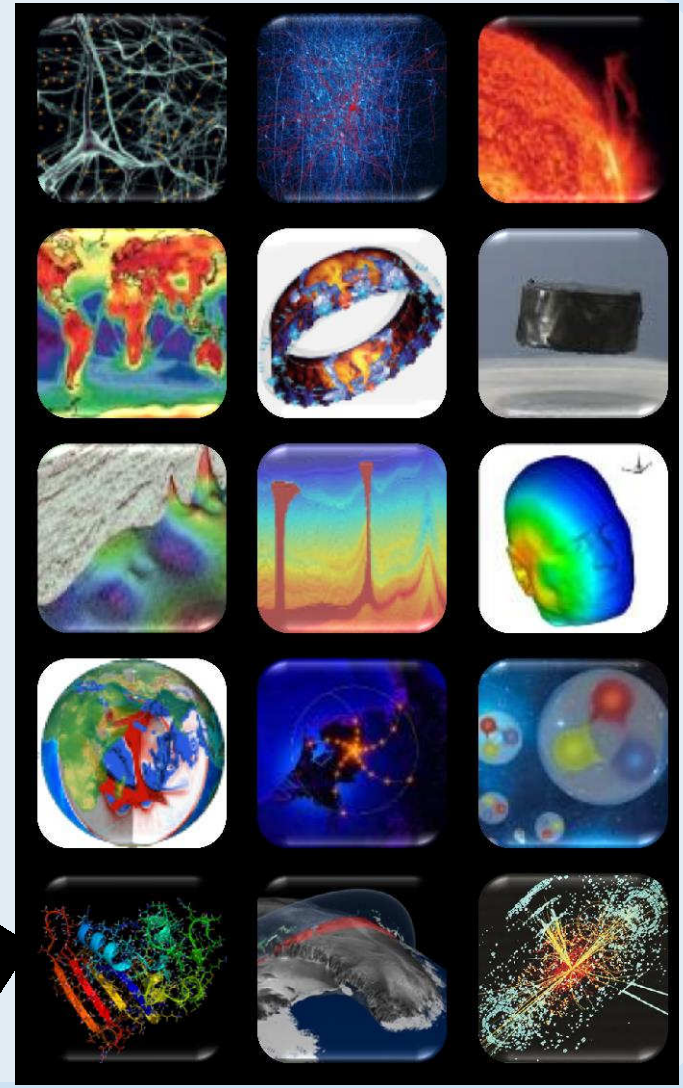
Combination of SW packages provides new functionality and exploits HW

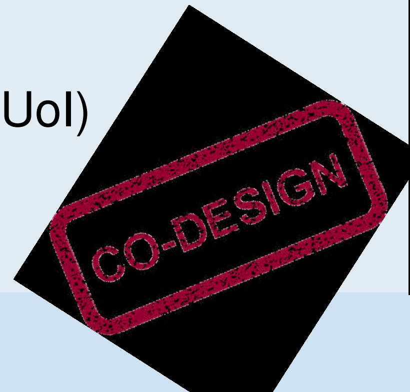
APPLICATIONS

Application-driven approach

DEEP projects applications (15):

- Brain simulation (EPFL + NMBU)
- Space weather simulation (KULeuven)
- Climate simulation (Cyprus Institute)
- Computational fluid engineering (CERFACS)
- High temperature superconductivity (CINECA)
- Seismic imaging (CGG + BSC)
- Human exposure to electromagnetic fields (INRIA)
- Geoscience (LRZ)
- Radio astronomy (Astron)
- Lattice QCD (University of Regensburg)
- Molecular dynamics (NCSA)
- Data analytics in Earth Science (UoI)
- High Energy Physics (CERN)





Architecture advantages

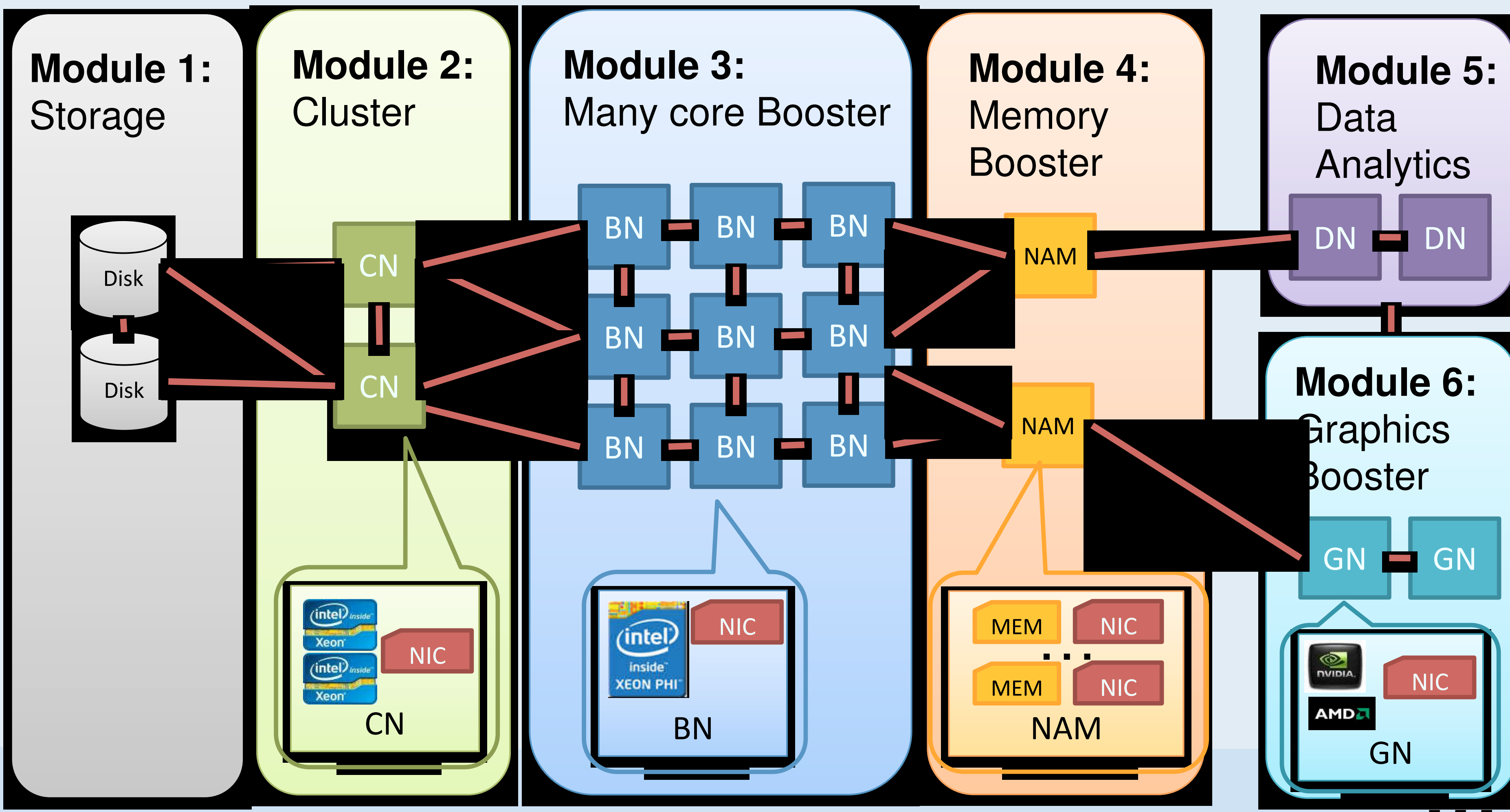
- **Full user flexibility** – many different use modes
  - Dynamic ratio of processors/coprocessors
  - Use Booster as pool of accelerators (globally shared)
  - Discrete use of the Booster
  - Discrete use + I/O offload
  - Specialized symmetric mode
- **More efficient use of system resources**
  - Only resources really needed are blocked by applications
  - Dynamic allocation further increases system utilization
- Better **I/O performance and resiliency**

DEEP-EST and JURECA

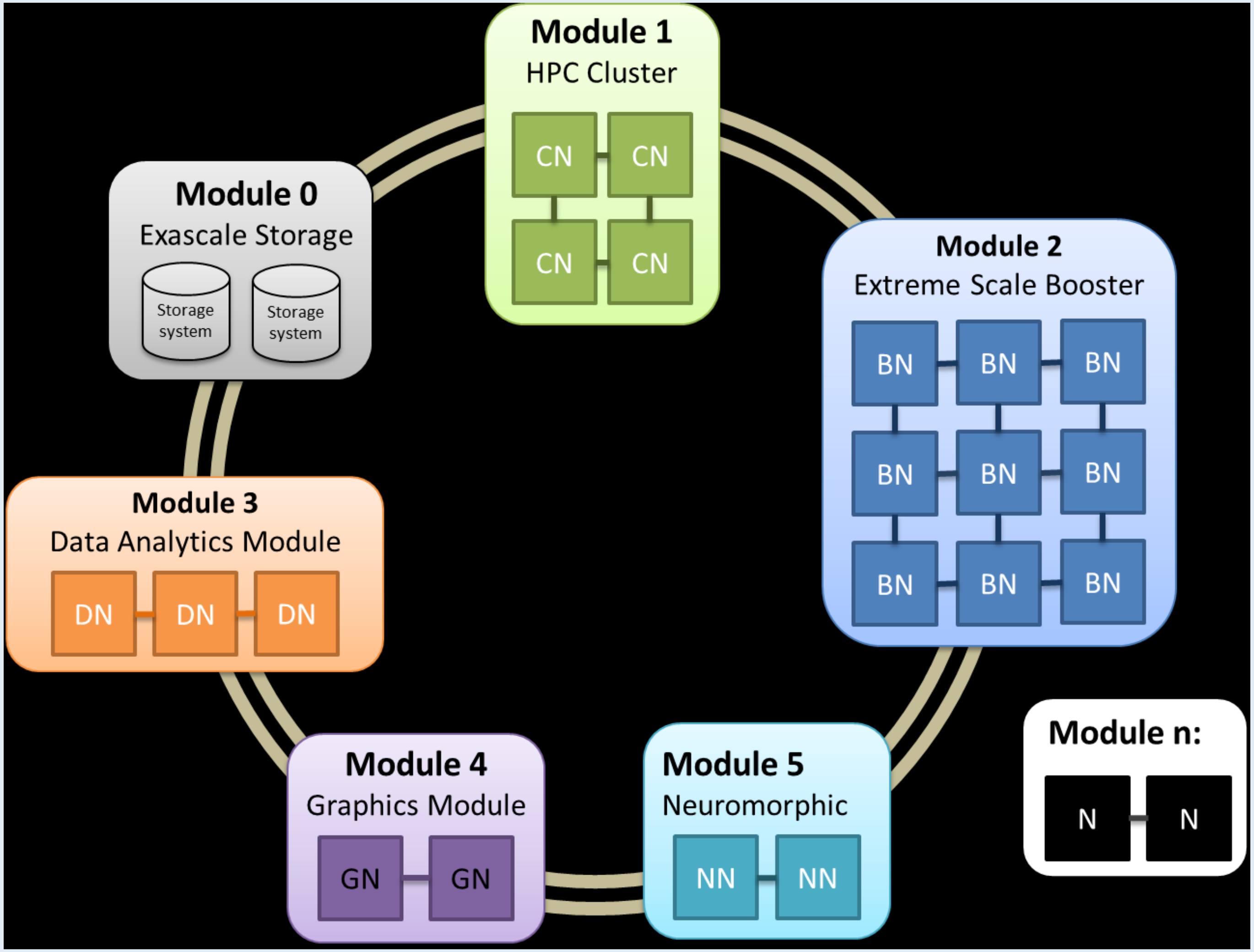
MODULAR SUPERCOMPUTING ARCHITECTURE

Modular Supercomputing

Generalization of the Cluster-Booster concept

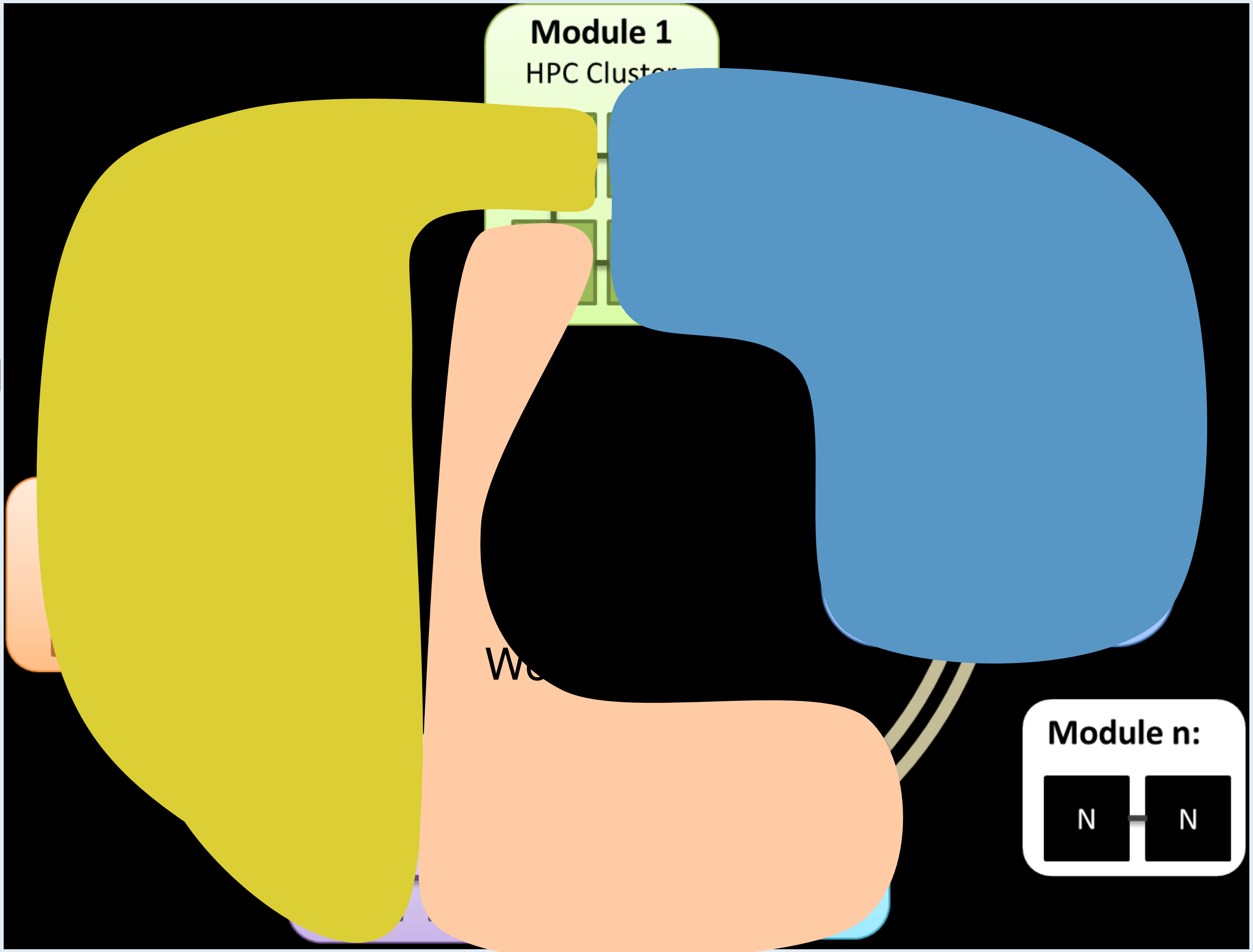


Modular Supercomputing

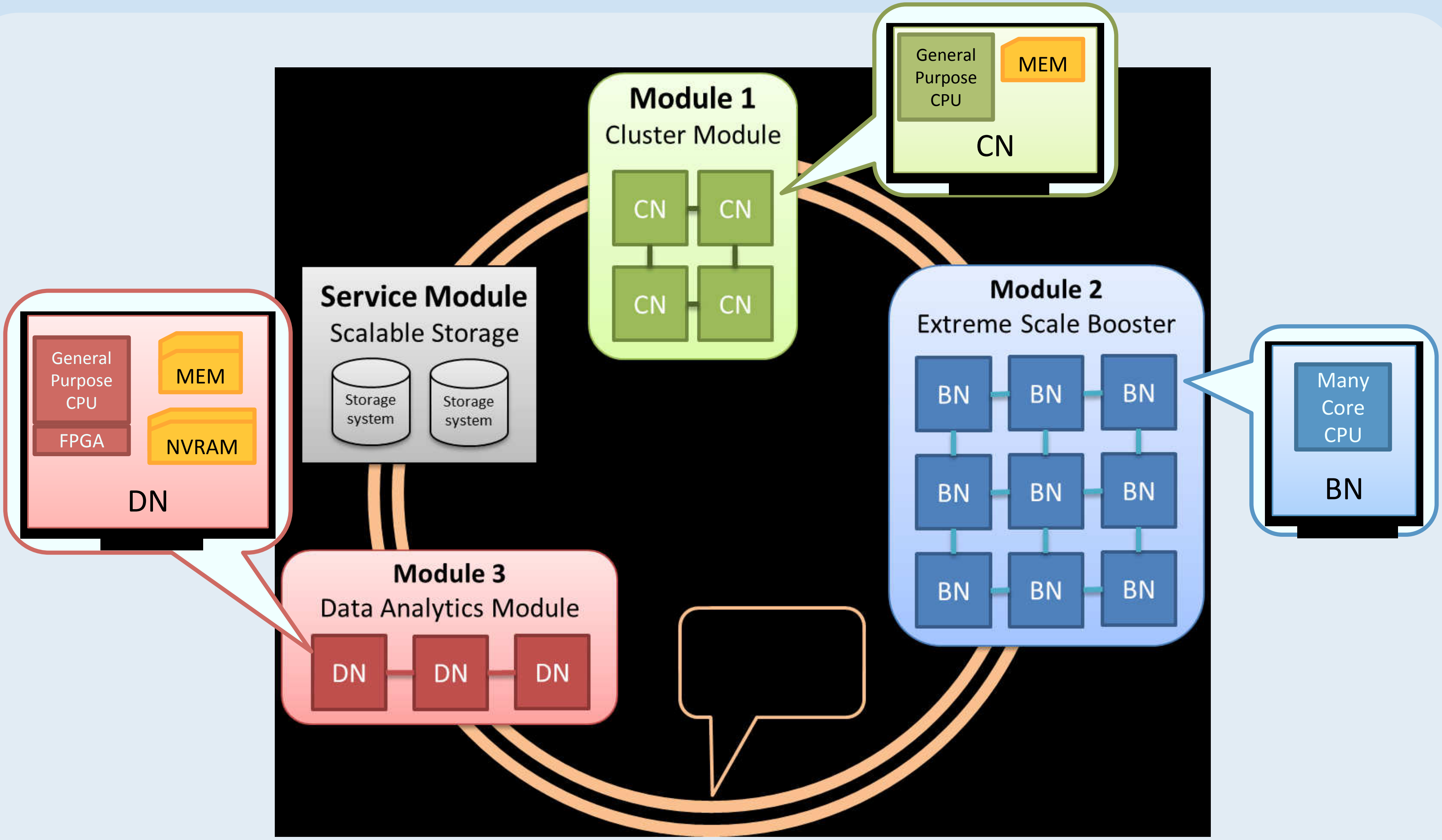


Modular Supercomputing

Workload

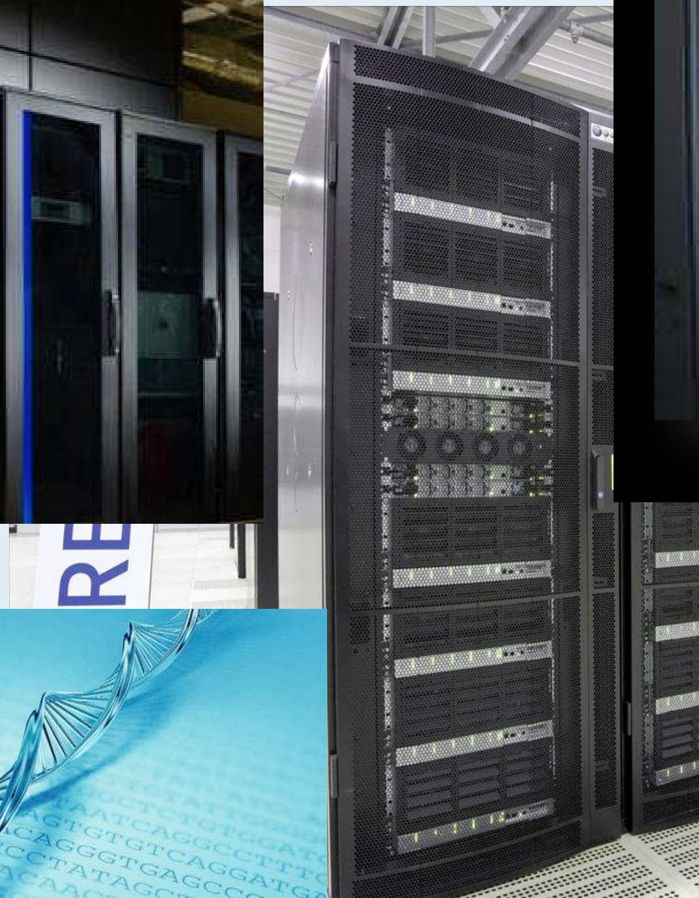


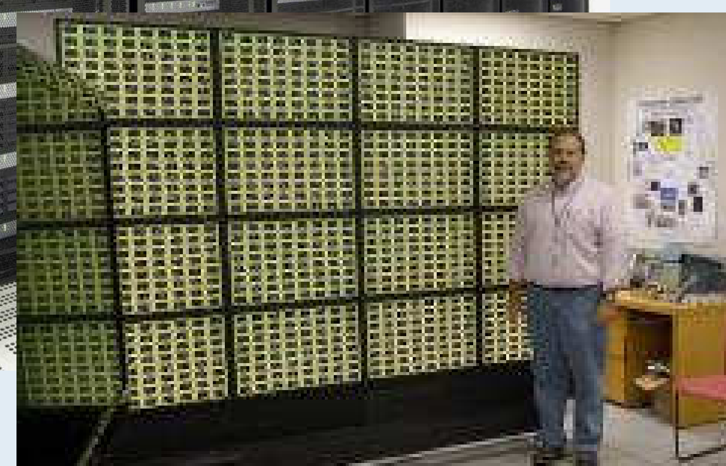
DEEP-EST prototype



DEEP Projects

Going production





Summary

The DEEP projects bring a new view to heterogeneity

- Modular Supercomputing architecture
- Software environment fully supporting system design
- Programming environment based on standards
- Hardware, software and applications jointly developed
- Strongly co-design driven
- Cluster + Booster going in production: JURECA system

Next step: DEEP-EST

- Three modules
- Address HPDA + HPC

Want to try out? →

www.deep-projects.eu
@DEEPprojects
pmt@deep-projects.eu