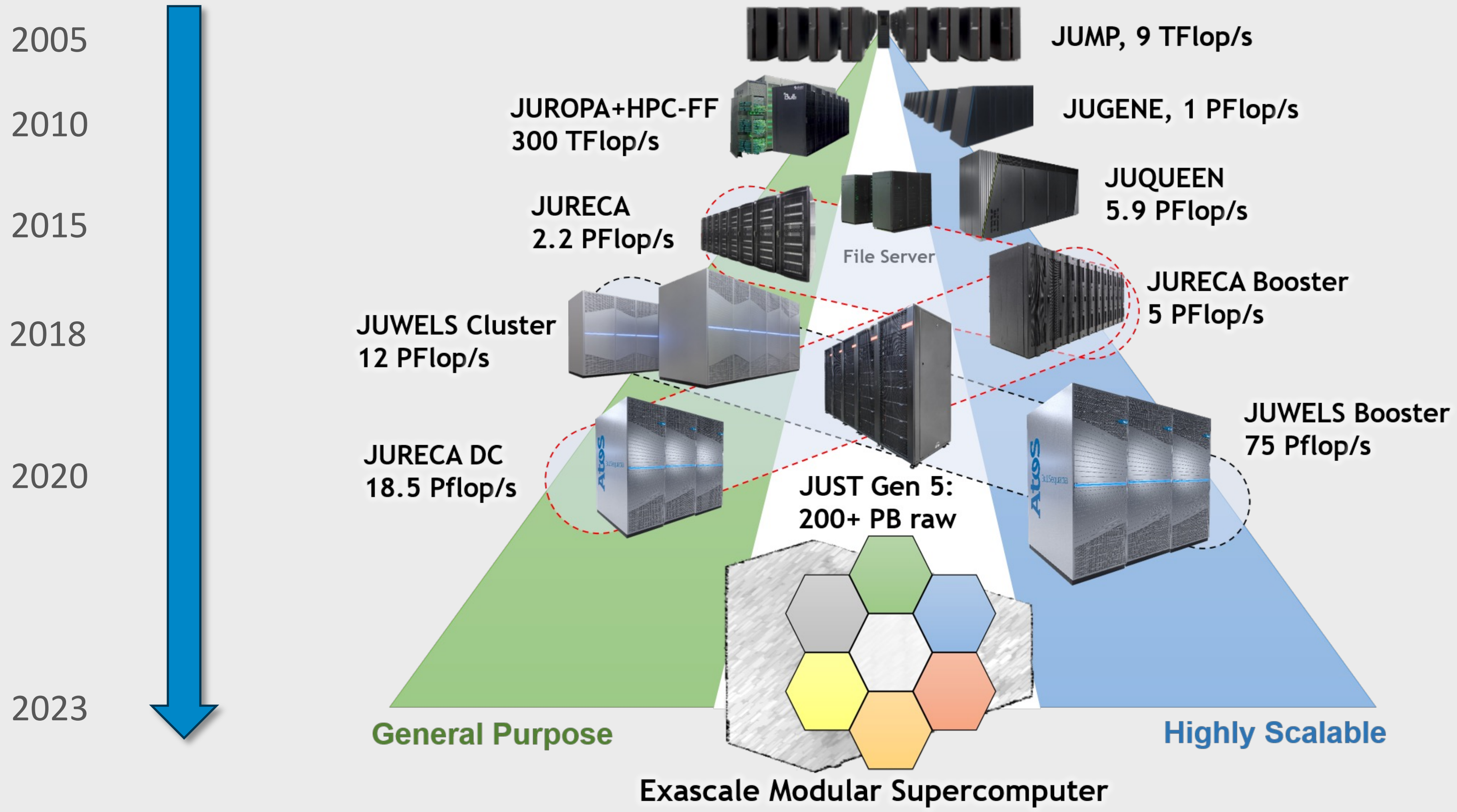
Bridging AI and HPC in the Center of Excellence RAISE

10.01.2024

eFlow4HPC Workshop, Barcelona Supercomputing Center

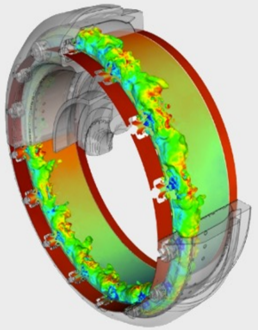
Andreas Lintermann
Jülich Supercomputing Centre, Forschungszentrum Jülich GmbH

The way to Exascale: Hardware strategy at JSC



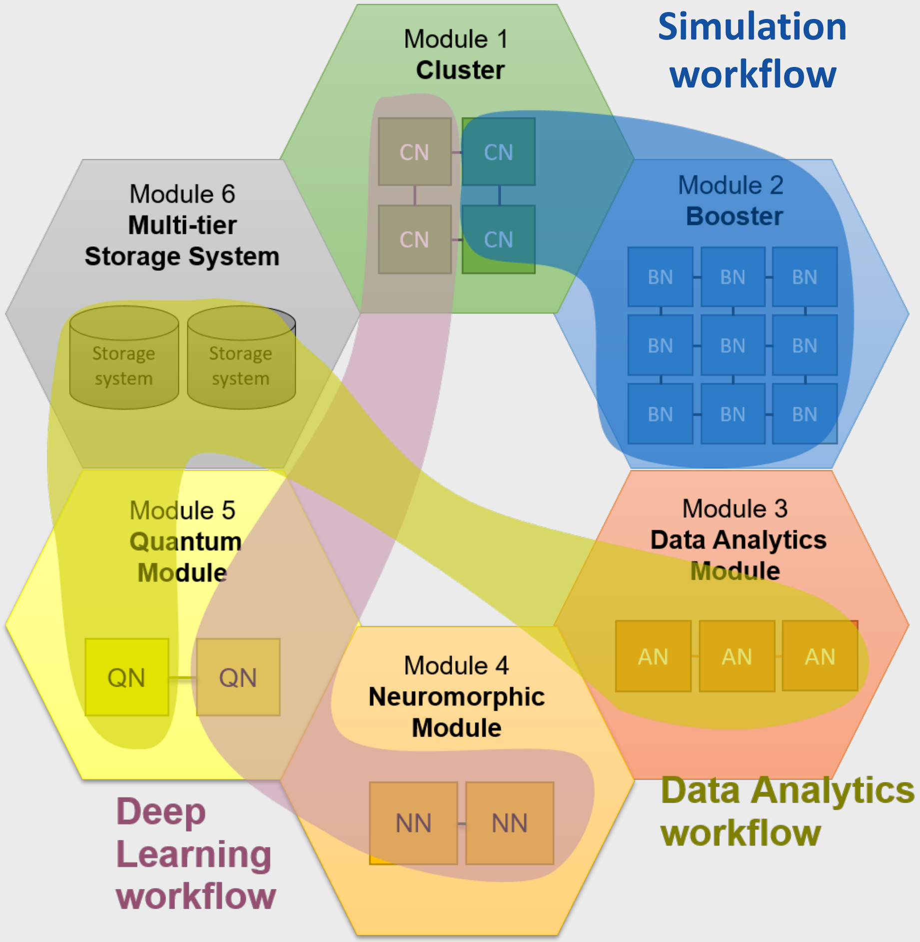
The way to Exascale: Modular supercomputing

Complex hardware



Complex task

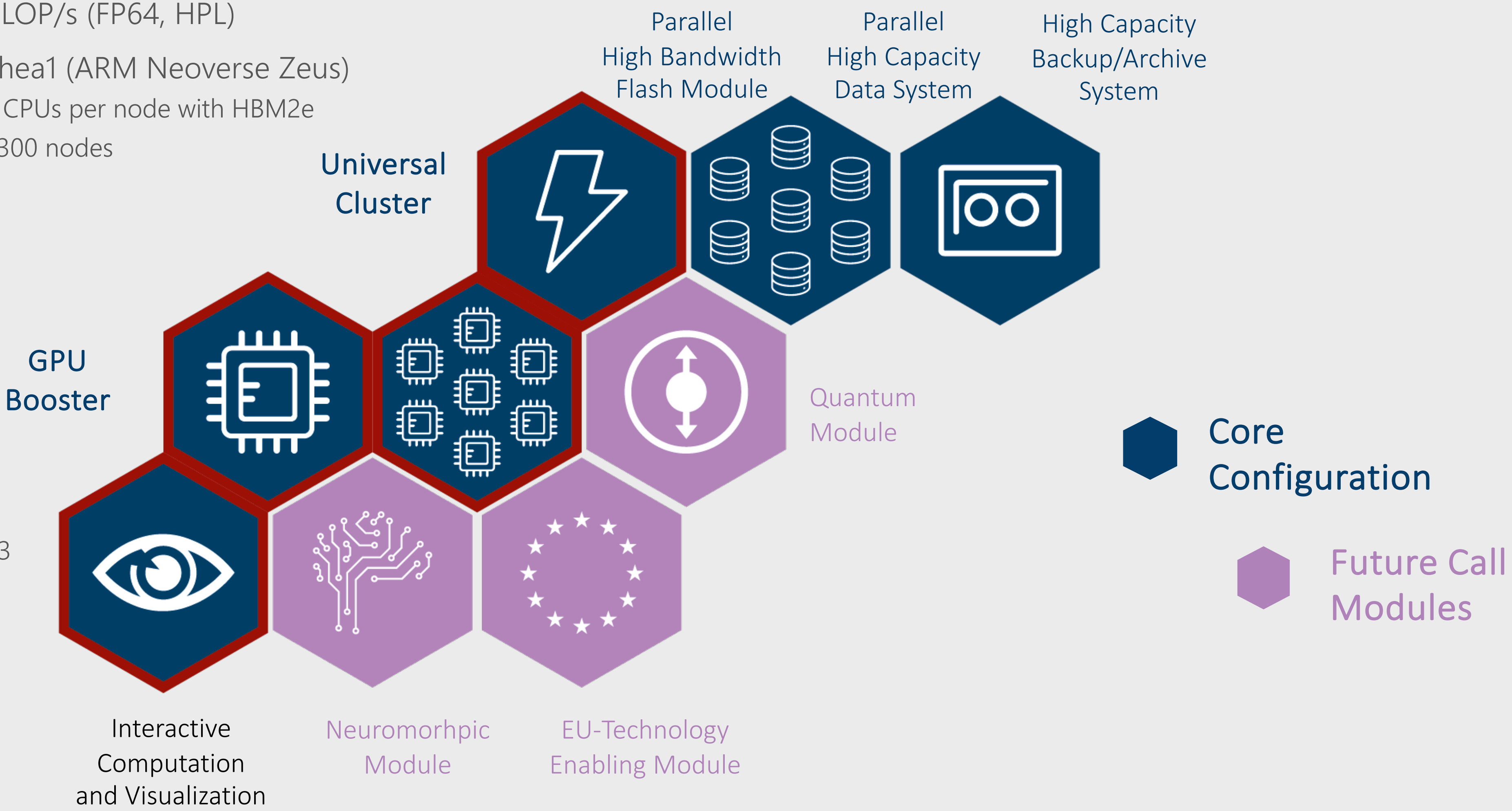
- Find the most suitable hardware for a specific task
- Cost-effective scaling
- Effective resource-sharing
- Match application diversity
- Enable intertwined AI- and HPC-workflows



JUPITER: The first European Exascale computer

- >5 PetaFLOP/s (FP64, HPL)
- SiPearl Rhea1 (ARM Neoverse Zeus)
  - 2 x CPUs per node with HBM2e
  - >1,300 nodes

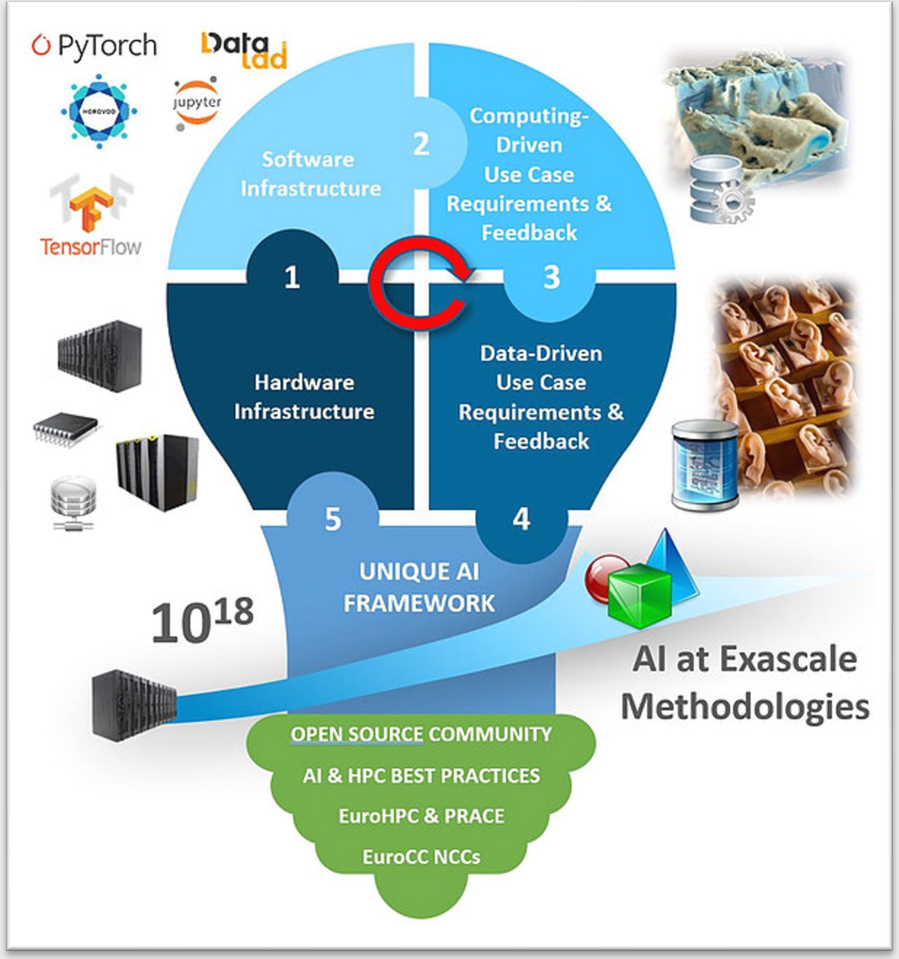
- 1 ExaFLOP/s (FP64, HPL)
- NVIDIA Grace-Hopper
  - 4 x chips per compute node
  - 72 cores per Grace CPU
  - Hopper H100 GPU with HBM3
  - ~6,000 nodes



CoE RAISE[1]'s major objectives

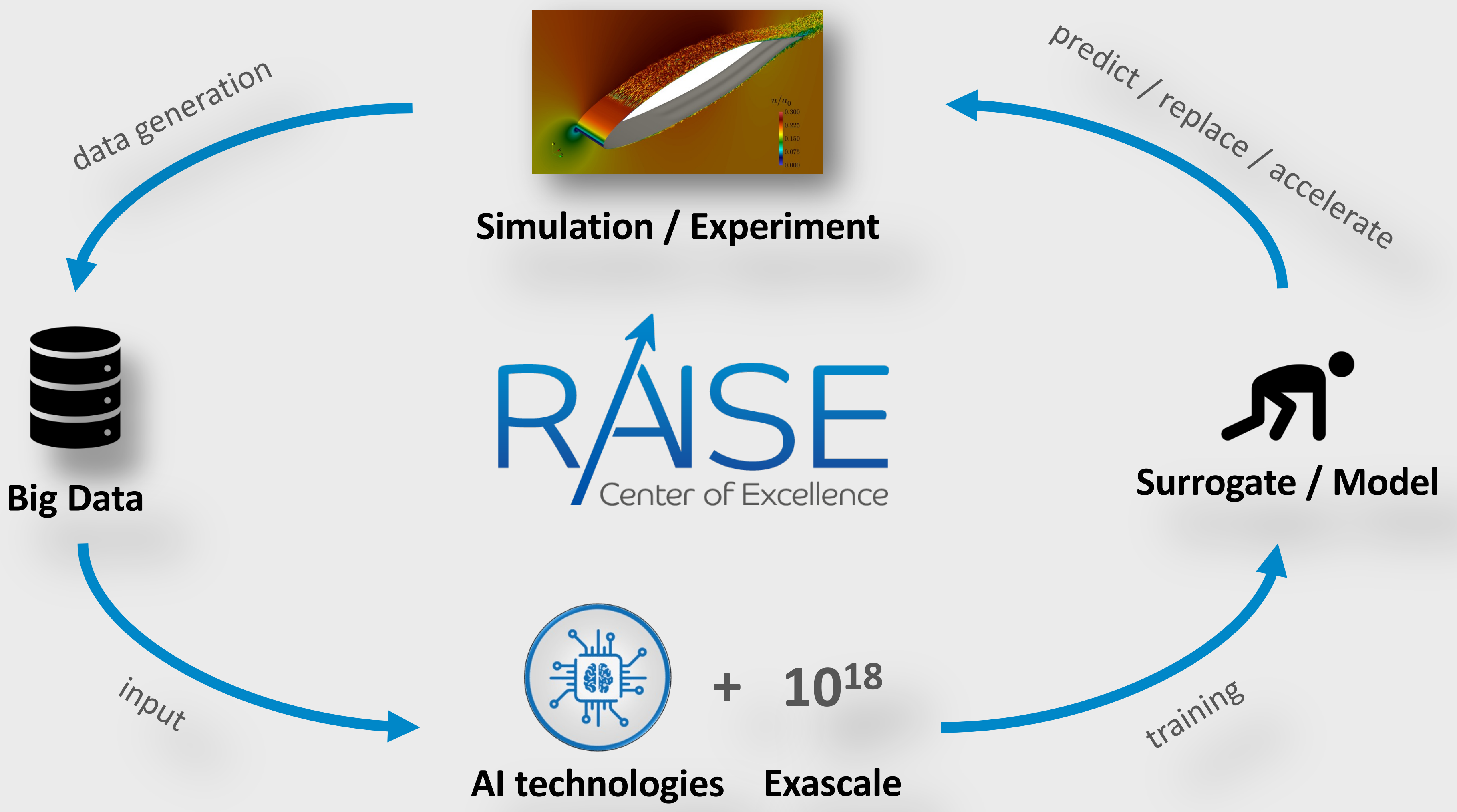
RAISE Center of Excellence

- Development of AI methods towards Exascale
- Connect
  - hardware infrastructure,
  - software infrastructure,
  - compute-driven use cases,
  - and data-driven use cases
- Reach out to other CoEs and EU projects
- Business development
- Create a **Unique AI Framework (UAIF)** for academia and industry

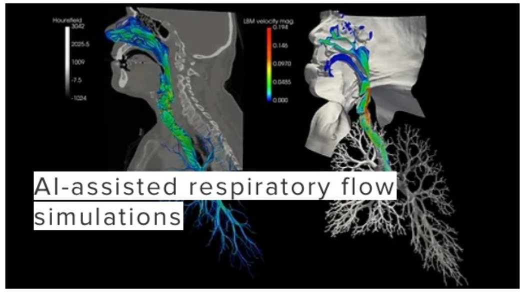


[1] https://www/coe-raise.eu

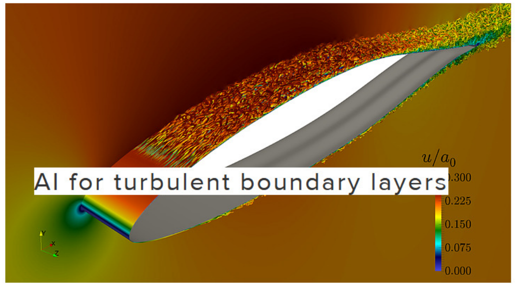
CoE RAISE: Full loop implementations



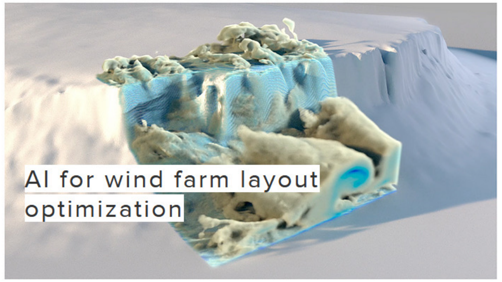
CoE RAISE Use-cases[2]



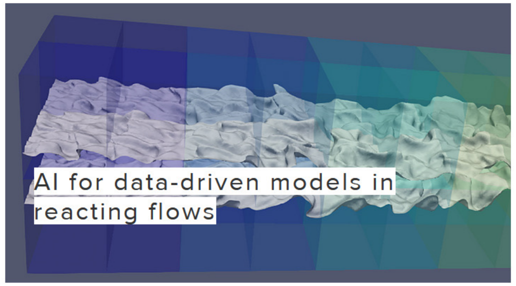
AI-assisted respiratory flow simulations



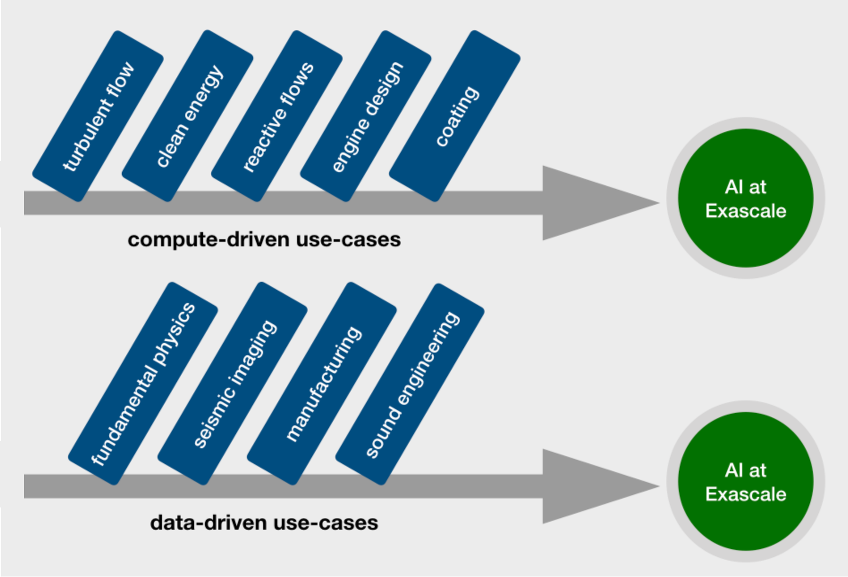
AI for turbulent boundary layers

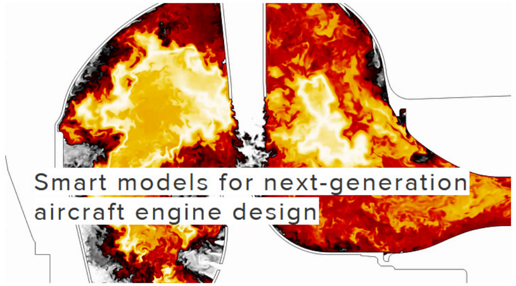


AI for wind farm layout optimization

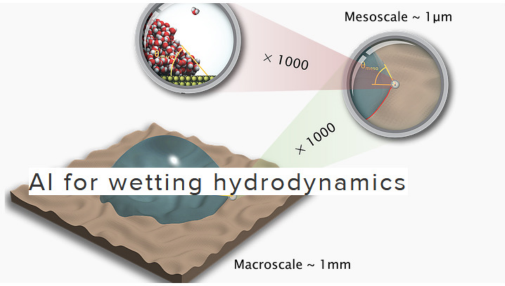


AI for data-driven models in reacting flows

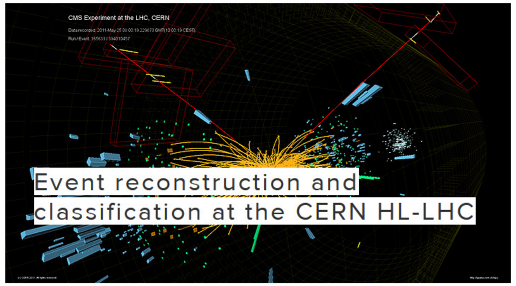




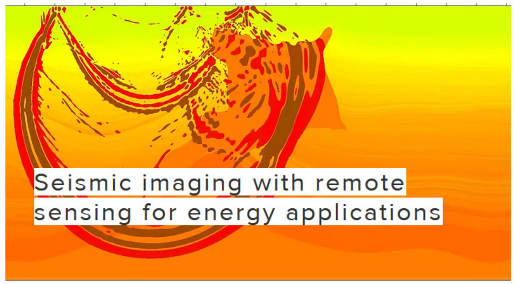
Smart models for next-generation aircraft engine design



AI for wetting hydrodynamics

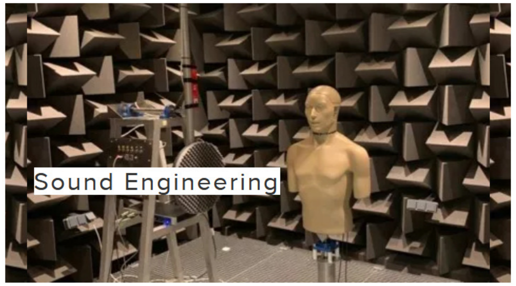


Event reconstruction and classification at the CERN HL-LHC



Seismic imaging with remote sensing for energy applications

Defect-free metal additive manufacturing



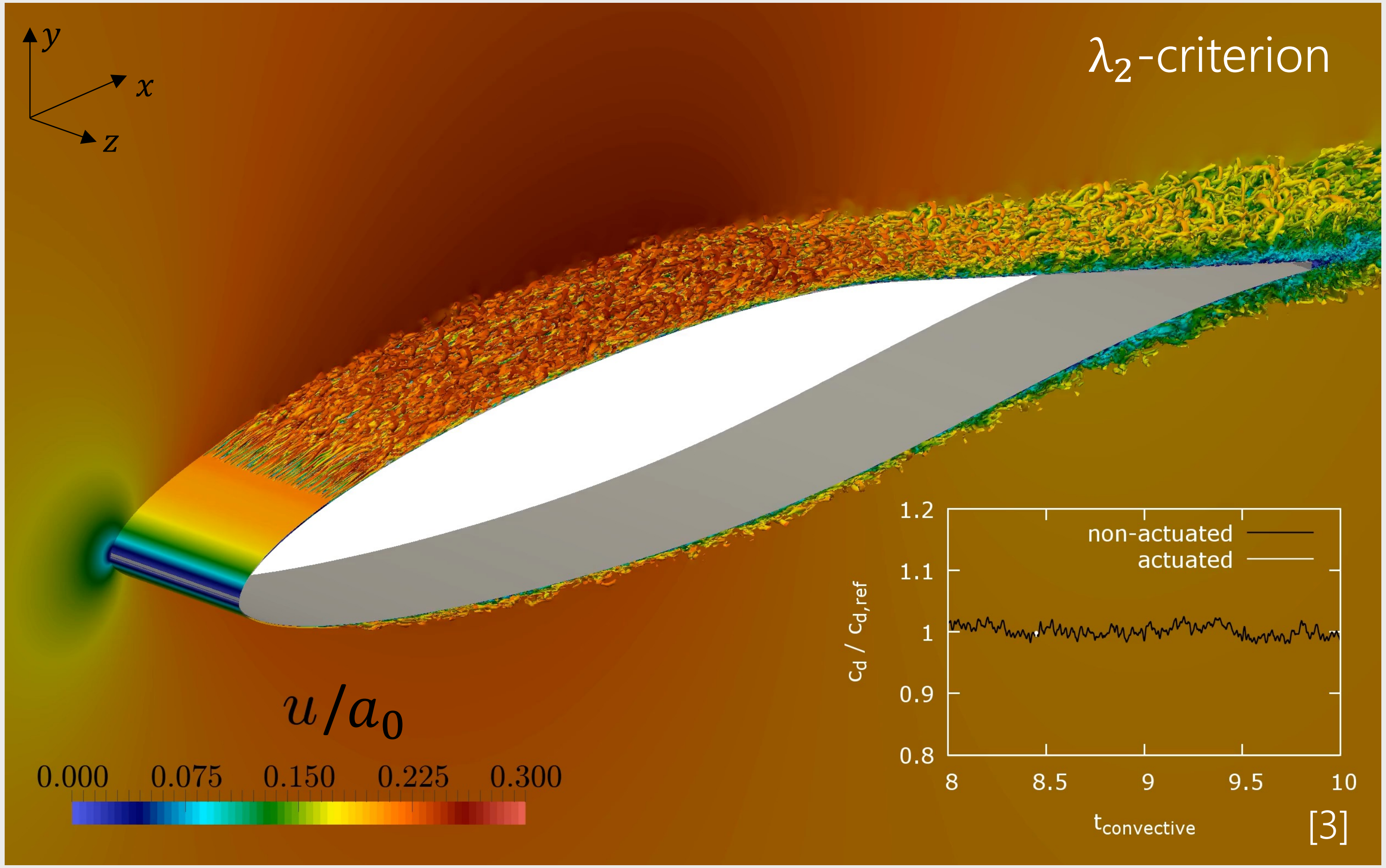
Sound Engineering

[2] https://www.coe-raise.eu/use-cases

AI for Turbulent Boundary Layer Flows

RAISE Center of Excellence

- Active Drag Reduction with spanwise traveling transversal surface waves
- Reduce energy consumption and emissions
- Short/middle-term vision
  - Understand the mechanism of this method, reduce simulation costs, and optimize the design choices
- Long-term vision
  - Application in airplanes in cruise flight, highspeed trains, etc.
- Simulation-based analysis requires HPC resources



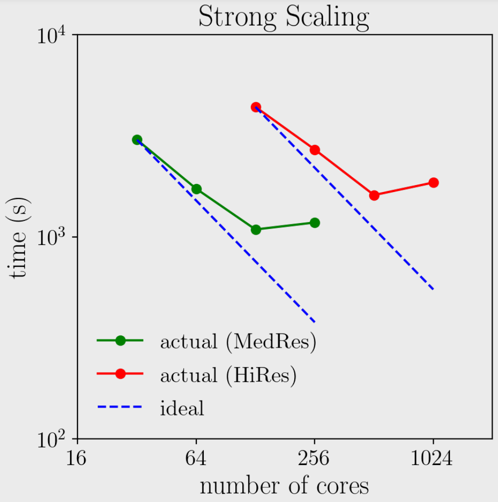
[3] https://doi.org/10.1016/j.ijheatfluidflow.2020.108770l

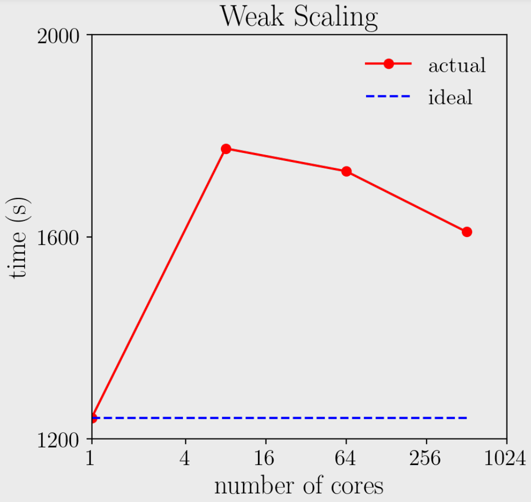
AI for Wetting Hydrodynamics

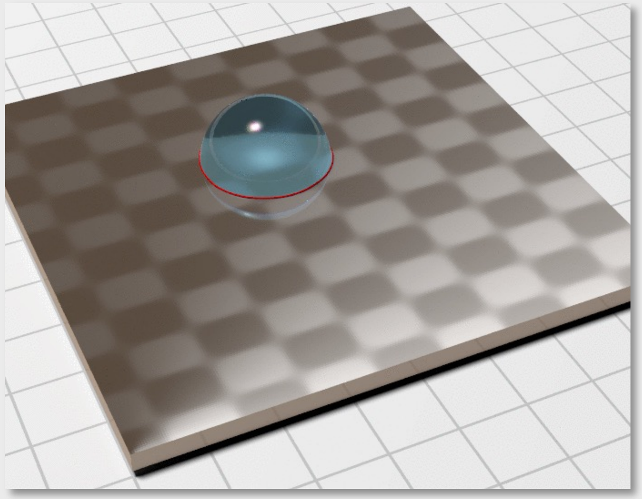
- DNS data are postprocessed
- Use of the Fourier Neural Operator (FNO)

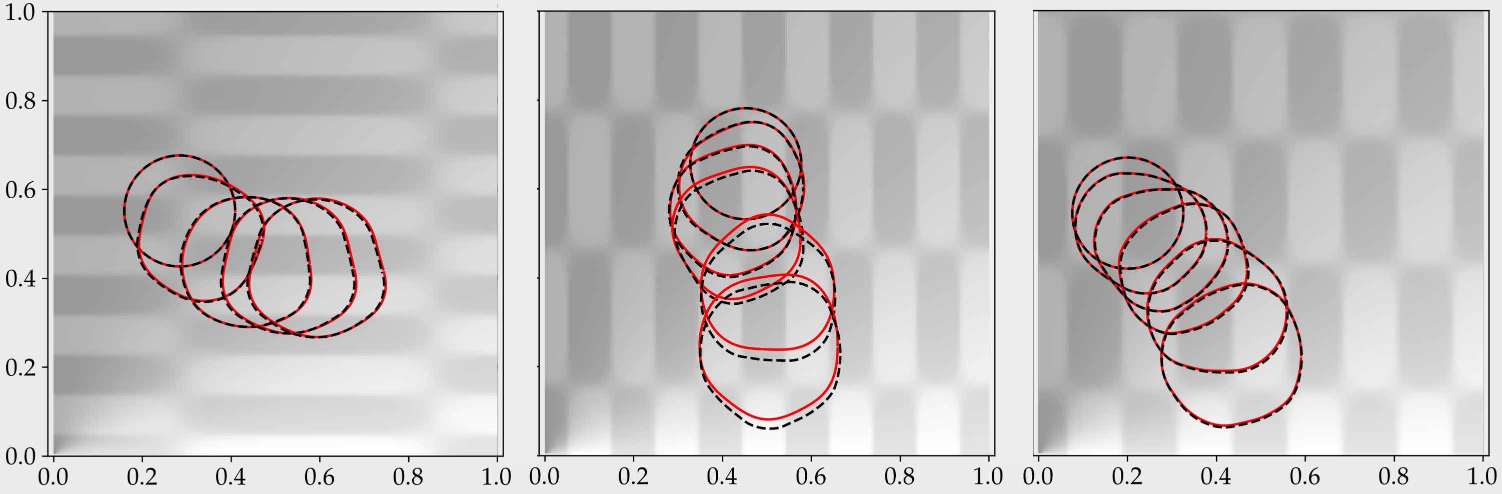
Data-driven surrogate model for the contact line velocity

- **Basilisk:** C-based CFD code (GNU)
  - AMR, VoF, multigrid, MPI + OpenMP





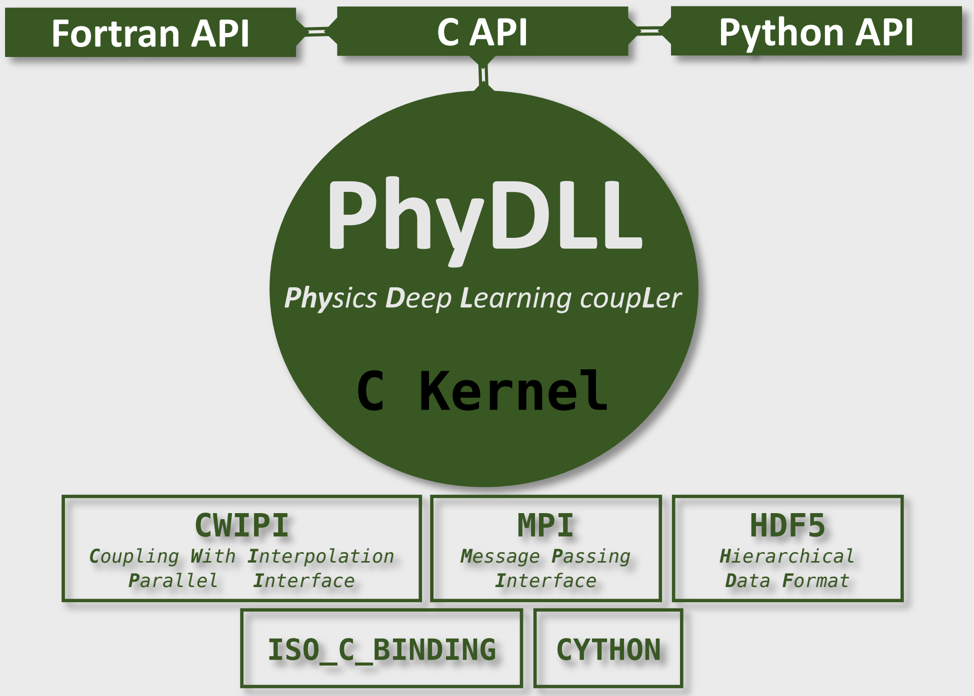




Coupling AI and Simulation Tools

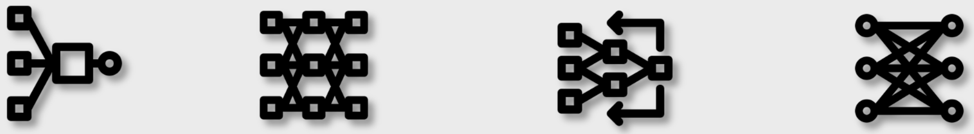
PhyDLL (Physic Deep Learning CoupLer)[4]

- Open-source high-performance coupling library
- Couples massively-parallel physical solvers to distributed Deep Learning inferences
- Kernel written in C
- C, Fortran, and Python API



[4] GitLab: https://gitlab.com/cerfacs/phydll Website: https://phydll.readthedocs.io

Hyperparameters in Machine Learning Models

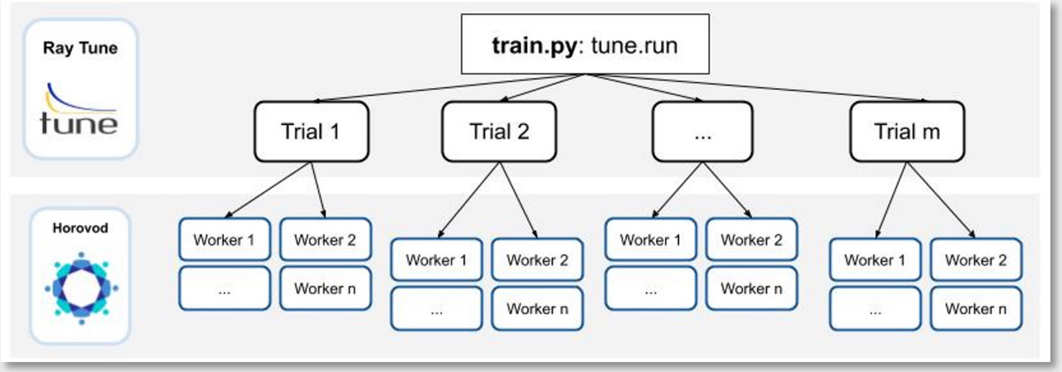


- Architectural parameters
  - Type of model (neural network, SVM?)
  - Number and kind of layers (convolutional, dense, dropout?)
  - Activation functions (ReLu, sigmoid?)
  - Weight initialization and regularization
- Optimizer parameters
  - Optimizer type (SGD, Adam?)
  - Batch size
  - Learning rate

Training ResNet18 on the cifar-10 dataset

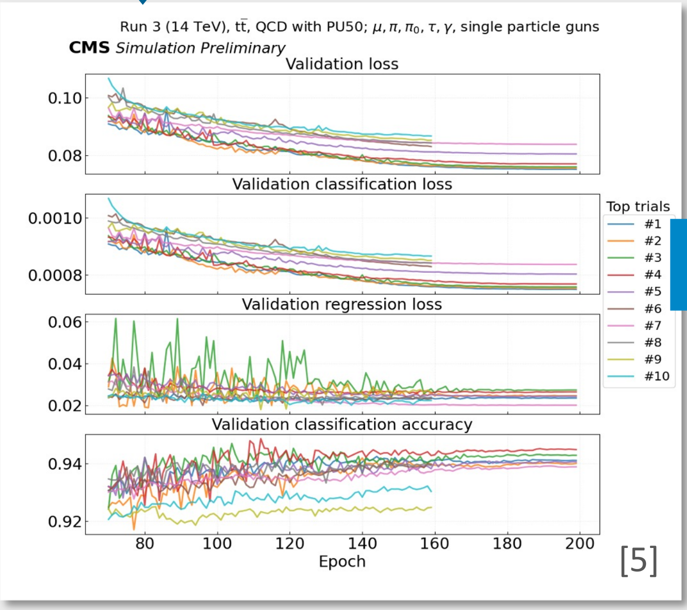
Hyperparameter Optimization (HPO) in High Energy Physics

Distributed HPO



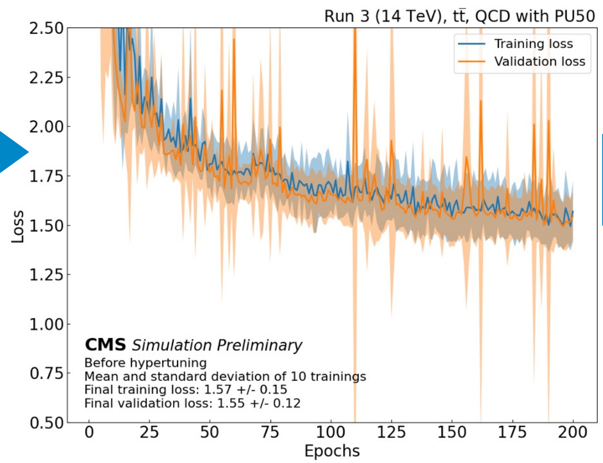
Model selection

- Using ASHA + Bayesian Optimization
- Scalable up to hundreds of GPUs
- Mean validation loss decreased by ~44% giving a significant performance improvement

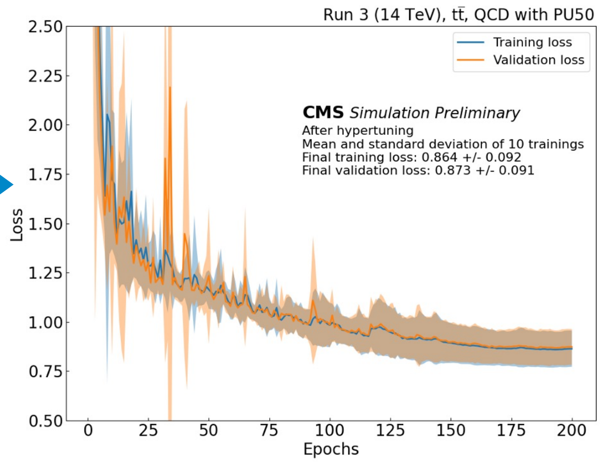


[5]

Assess learning variability



Better learning

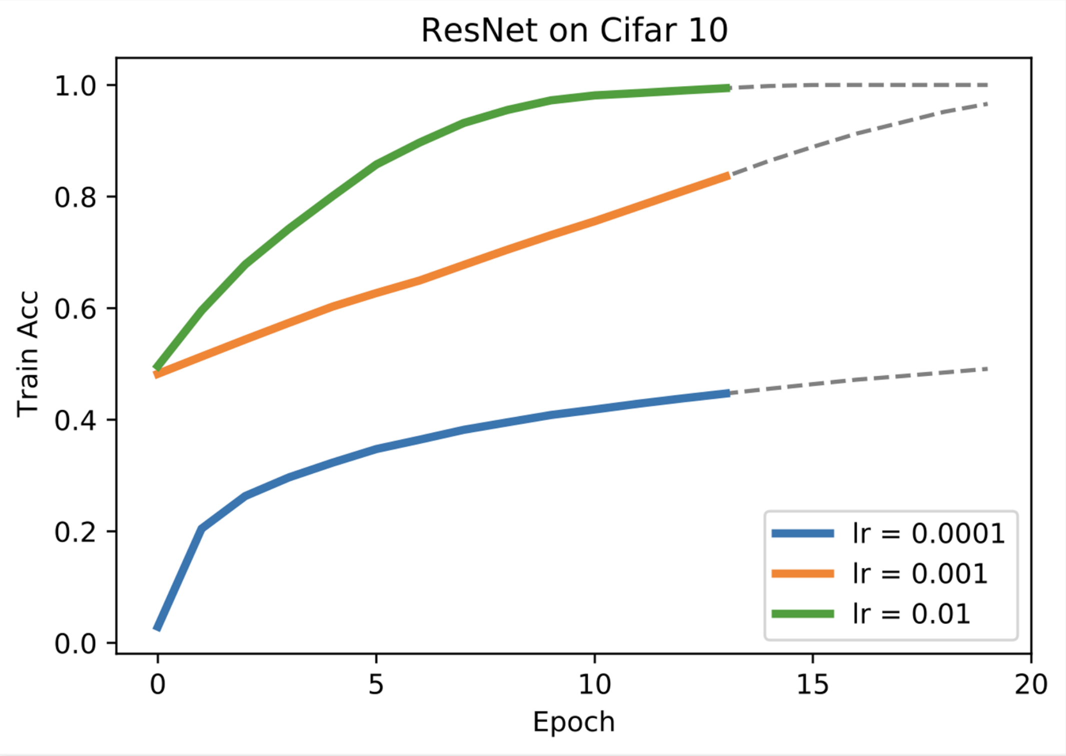


[7]

[5] https://doi.org/10.1088/1742-6596/2438/1/012092

Quantum-Assisted HPO

- Hybrid workflow:
  - Train different networks with different hyperparameters partially on JURECA-DC-GPU
  - Transfer the learning curves to JUPSI and perform extrapolation with regression methods
  - Fully train only the most promising models on JURECA-DC-GPU
- Python Libraries:
  - D-Wave-Ocean SDK
  - PyTorch



From SVR to QSVR: Formulation of the SVR

RAISE Center of Excellence

- Optimize the Lagrange dual form of an SVR problem:

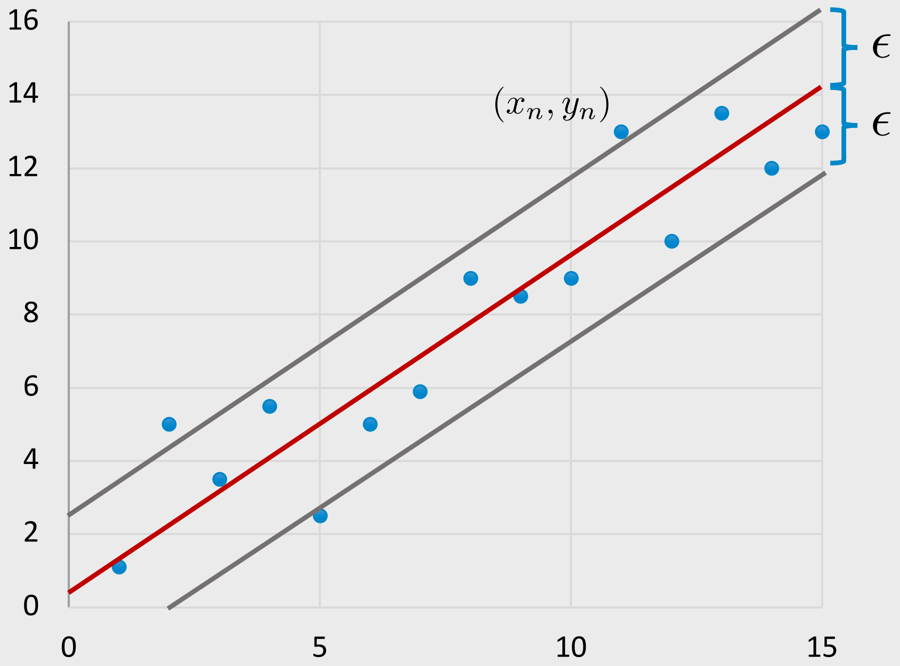
$$L(\boldsymbol{\alpha}, \hat{\boldsymbol{\alpha}}) = \frac{1}{2} \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} (\alpha_n - \hat{\alpha}_n)(\alpha_m - \hat{\alpha}_m) k(\boldsymbol{x}_n, \boldsymbol{x}_m) - \epsilon \sum_{n=0}^{N-1} (\alpha_n + \hat{\alpha}_n) + \sum_{n=0}^{N-1} (\alpha_n - \hat{\alpha}_n) y_n$$

multipliers (otimized for)

kernel function: RBF for non-linearity

$$+\xi \left(\sum_{n=0}^{N-1} (\alpha_n - \hat{\alpha}_n) \right)^2 + \beta \left(\sum_{n=0}^{N-1} \alpha_n \hat{\alpha}_n \right)$$

from contraints



- Quadratic Unconstrained Binary Optimization:

$$E = \sum_{i \leq j} a_i Q_{ij} a_j, \quad a_{i,j} \in \{0, 1\}$$

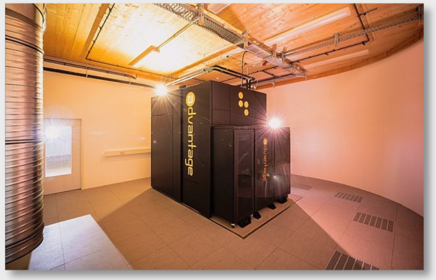
Machine Learning on a Quantum Annealer (QA)

- JUPSI at JSC features 5,615 qubits
- Quantum-Classical Workflows
  - Run trials (with different) hyperparameters on classical GPUs
  - Early stopping of “bad” trials by extrapolating the learning curves with Q-SVR

JUREA-DC-GPU

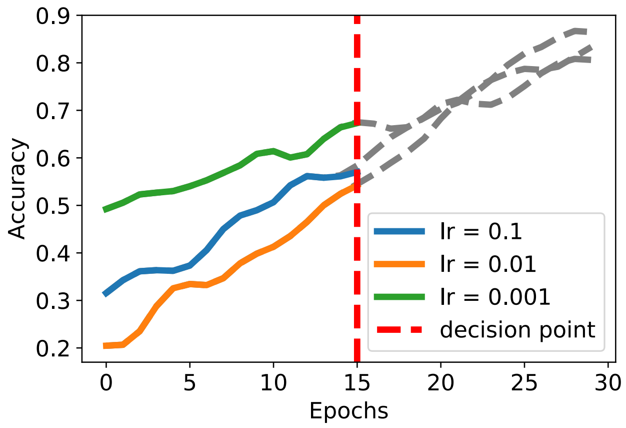
Hybrid

approach

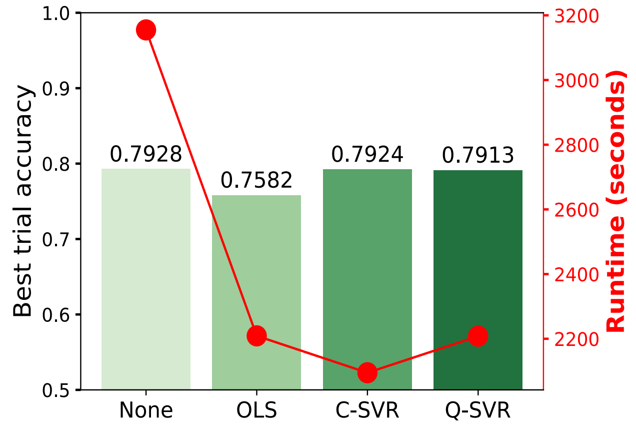


JUPSI

- **Results:**
  - Q-SVR slightly worse than Classical SVR (C-SVR)
- Why Q–SVR?
  - C-SVR complexity $O(n^2)$ - $O(n^3)$
  - Q-SVR complexity <u>does not</u> increase with problem size
  - Quantum speedup possible in the future



Learning curve extrapolation

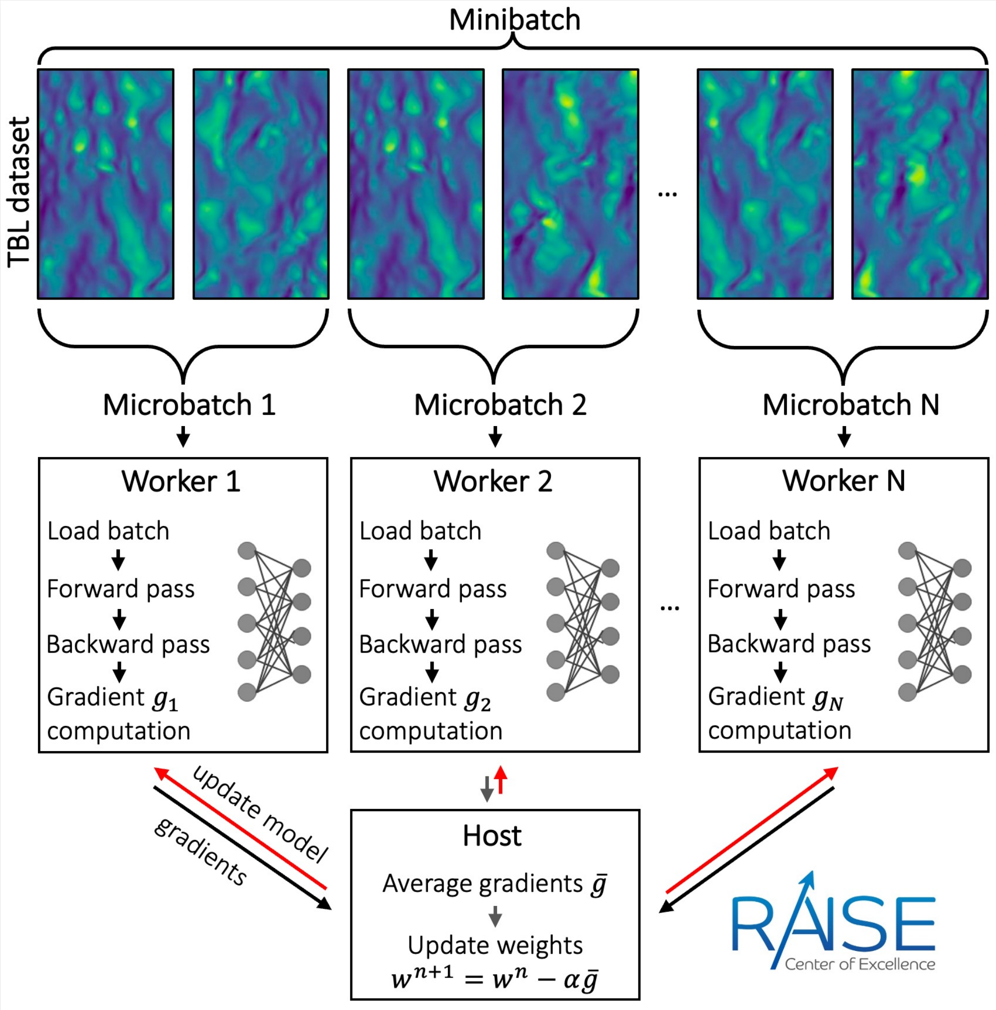


Early stopping extrapolation methods

Parallel training on HPC systems

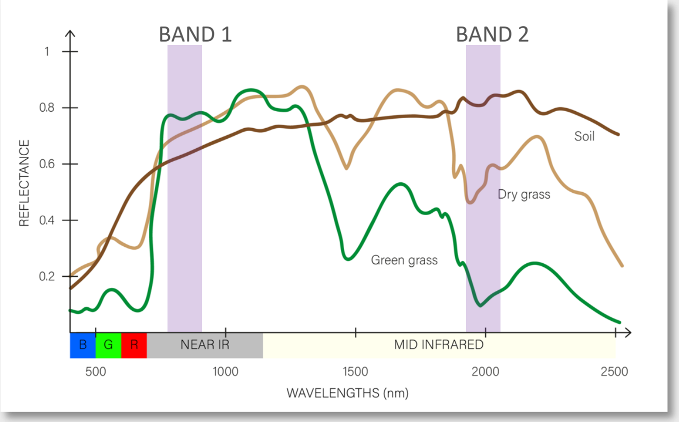
- Parallelization strategies for HPC training
  - Data Distributed Parallel with, e.g., PyTorch-DDP, Horovod, DeepSpeed
  - Model-parallelism using distributed architectures, gradient checkpointing

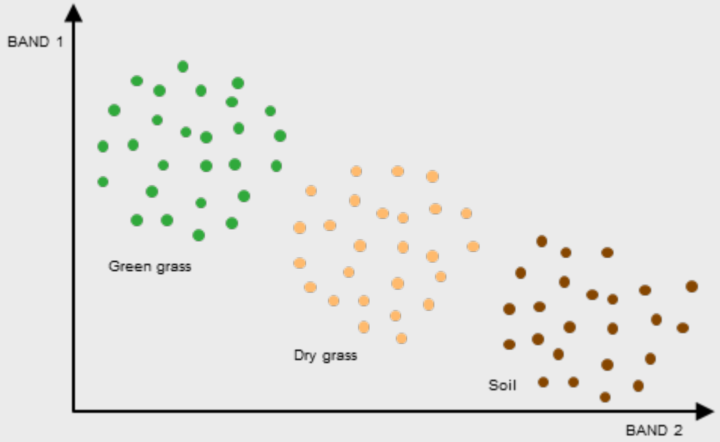
- Benchmarking on available systems (also on HPC prototypes as Exascale blueprints)

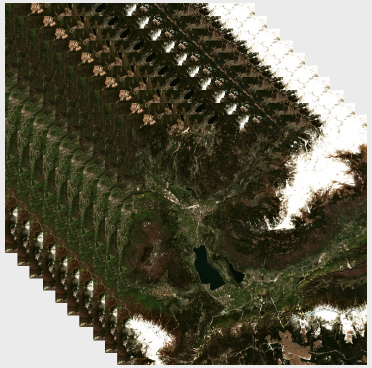


AI in Remote Sensing

- Aim: Produce a consistent multi-temporal land cover map
- Area: Trentino, Italy
- Data:
  - Multispectral information from Sentinel 2 [6]
  - Upsampling of bands to 10m resolution
  - 90,000 samples for training, prediction on ~ 120,000,000 pixels
  - Years: 2018 – 2020 with 15 acquisitions per year



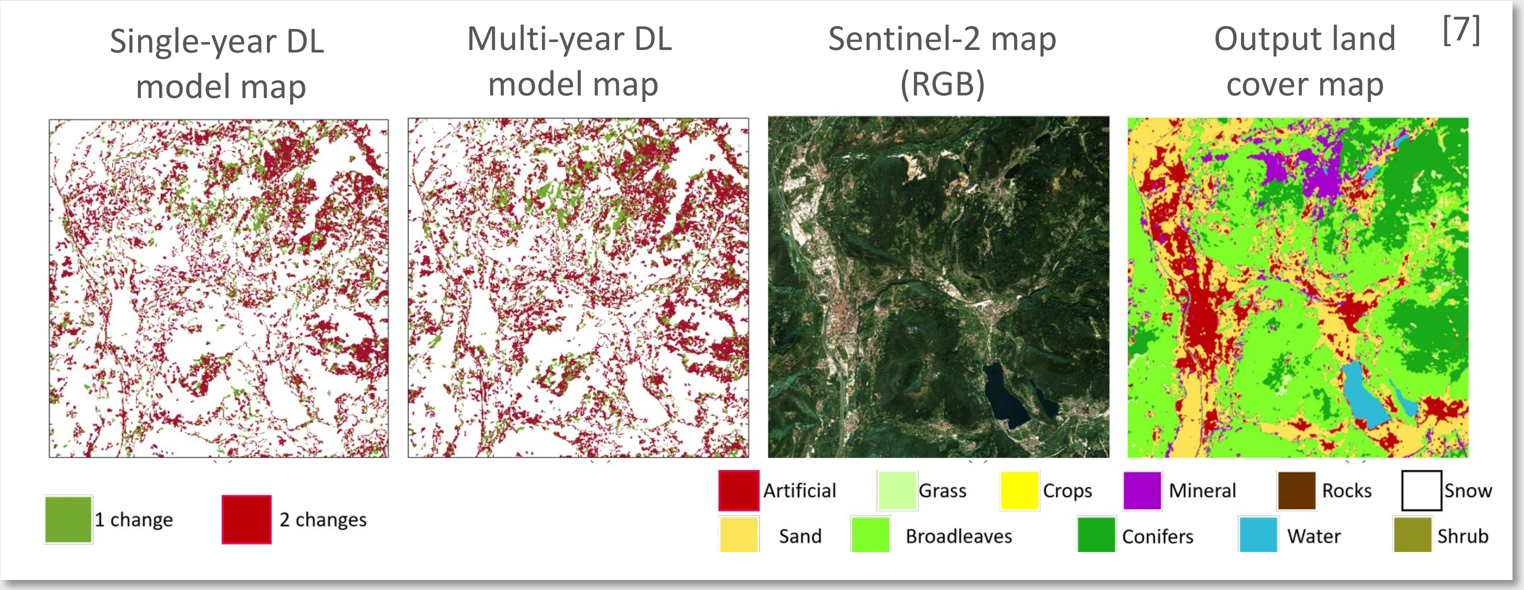


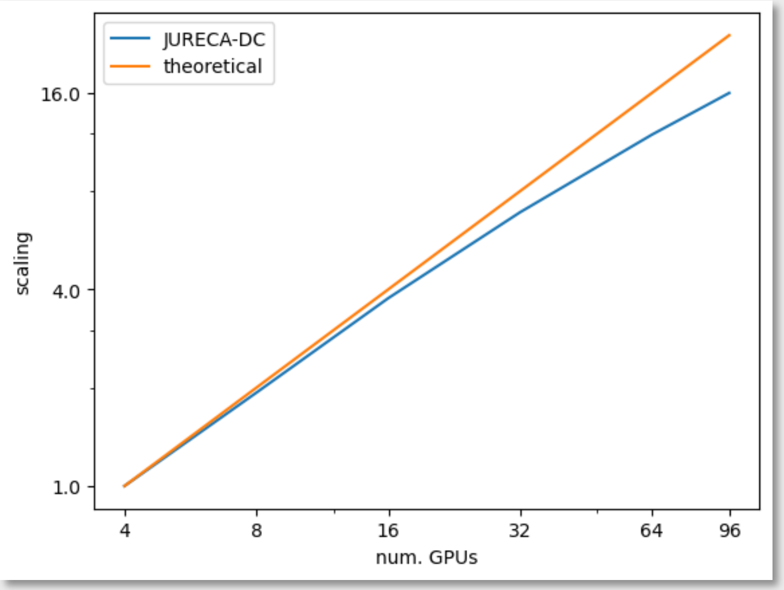


Data cube of satellite images

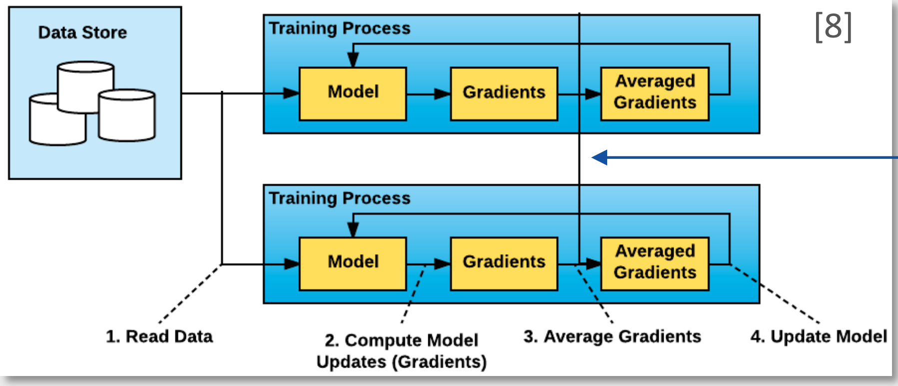
[6] Sentinel 2 - Copernicus https://sentinels.copernicus.eu/

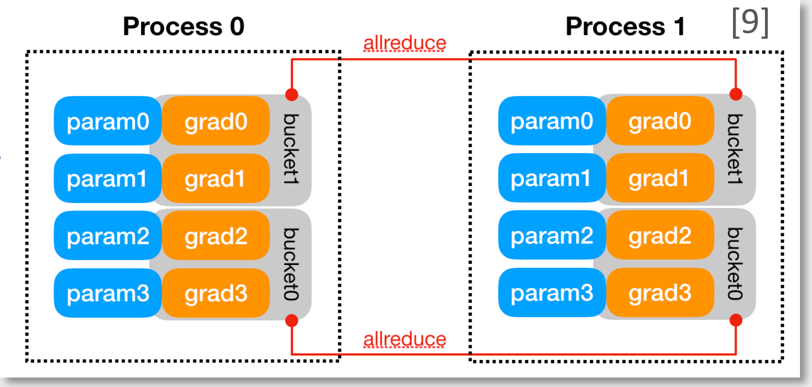
Distributed Deep Learning in Remote Sensing





Deployment on large datasets requires scaling up!

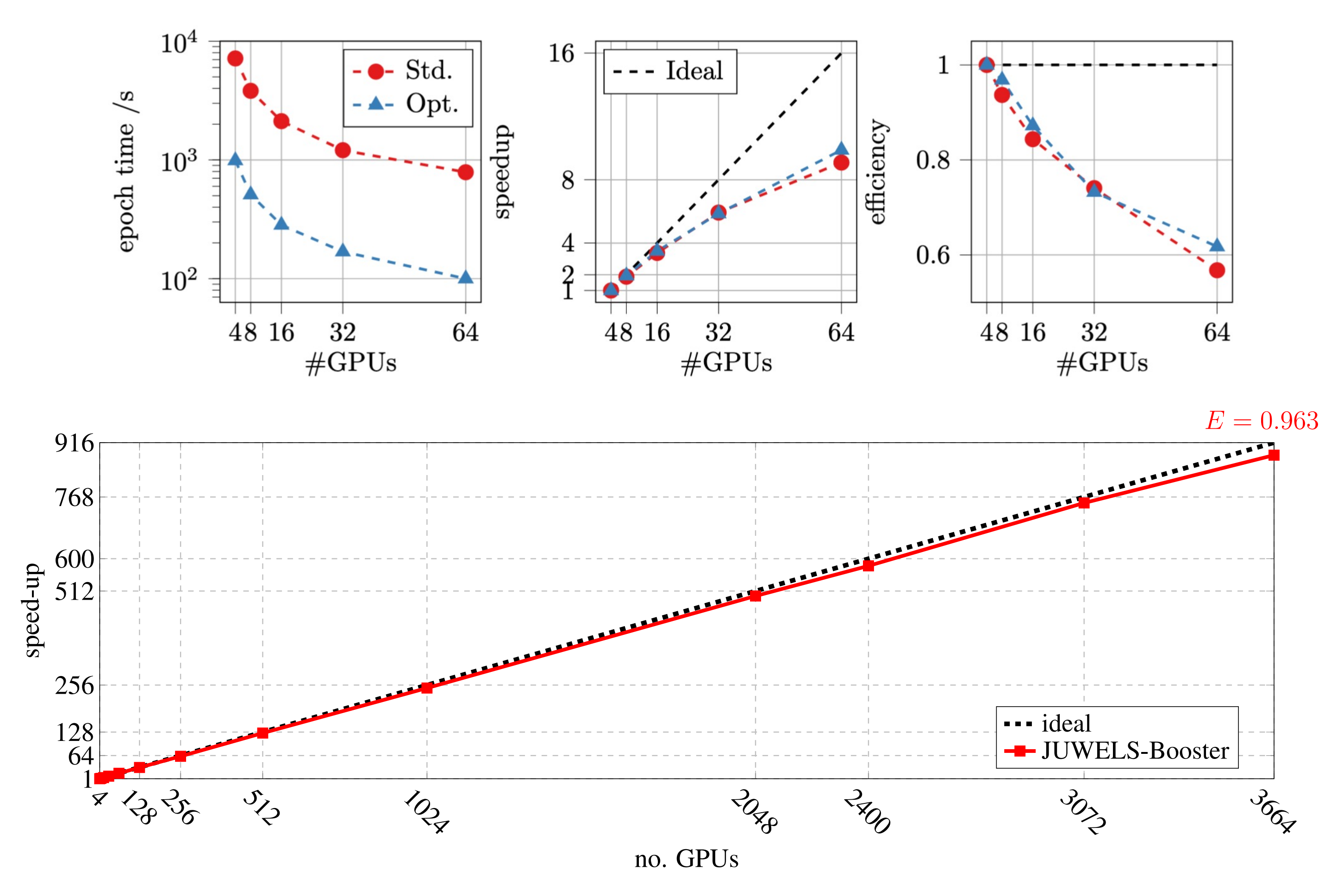




[7] https://doi.org/10.1109/IGARSS46834.2022.9883655
[8] https://pytorch.org/tutorials/intermediate/ddp_tutorial.html
[9] http://www.idris.fr/eng/ia/apprentissage-distribue-eng.html

Parallel training on HPC systems

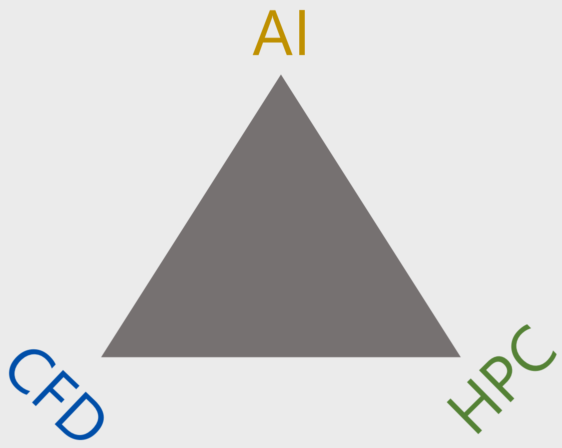
- Improvements by 90%
  - Computation:
    - Automatic Mixed Precision (AMP)
    - cuDNN Autotuner
    - Adaptive Summation Algorithm
    - Gradient checkpointing
  - Communication
    - Gradient accumulation
    - Automated skipping of AllReduce
    - MPI with CUDA awareness
  - I/O
    - Multi-process dataloader (CPU or GPU with direct access - DALI)
- Horovod towards Exascale (CFD)



CoE RAISE Tools: AI4HPC [10]

- AI4HPC contains
  - Pre-processing routines
    - Can deal with irregular data shapes as input
  - ML models for CFD
    - Autoencoders, super-resolution, etc.
  - HPC optimizations
    - Up to 10x speed-up (compared to the standard PyTorch implementation)
    - Tested on JURECA-DC, JUWELS-BOOSTER, LUMI, CTE-AMD, PIZ-DAINT
  - Post-processing routines
    - Data analysis via Jupyter notebooks
  - Benchmarking suite
  - Hyperparameter Optimization (HPO) suite

AI4HPC:
open-source library to train AI models with CFD datasets on HPC systems



[10] https://ai4hpc.readthedocs.io

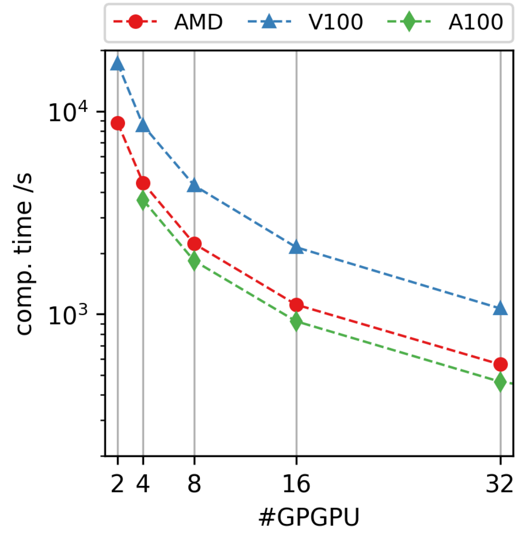
CoE RAISE Tools: AI4HPC Performance

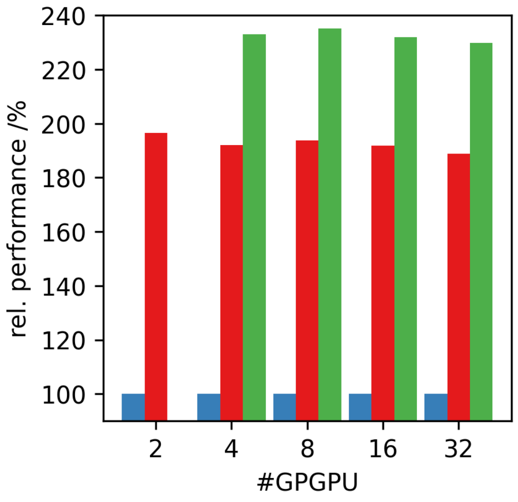
RAISE Center of Excellence

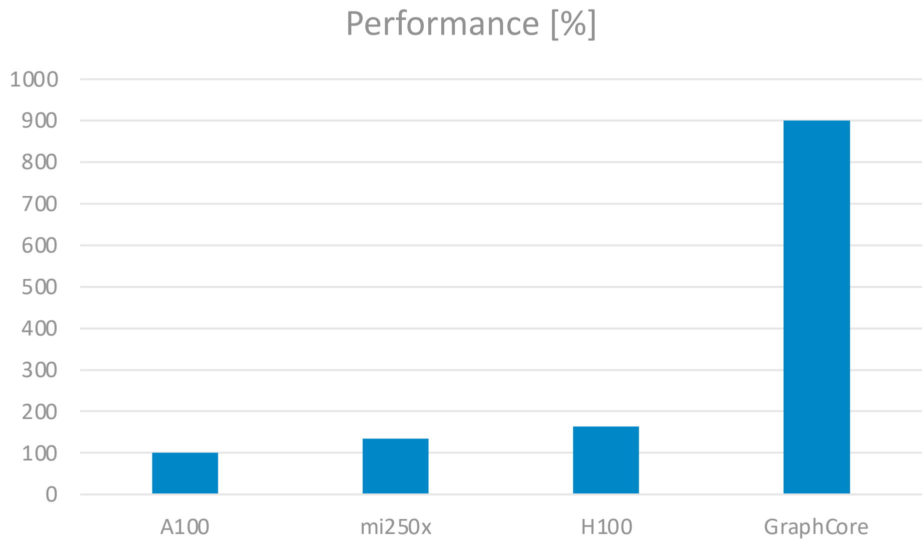
GPU Scalability

- AMD MI100 (CTE-AMD)
- NVIDIA V100 (DEEP-EST)
- NVIDIA A100 (JUWELS)

	NVIDIA A100*	AMD MI250x	NVIDIA H100	GraphCore IPU [11],**
System	JUWELS	LUMI	JURECA-DC	JURECA-DC
Epoch time [s]	54	40	33	6
Performance [%]	100	135	163	900





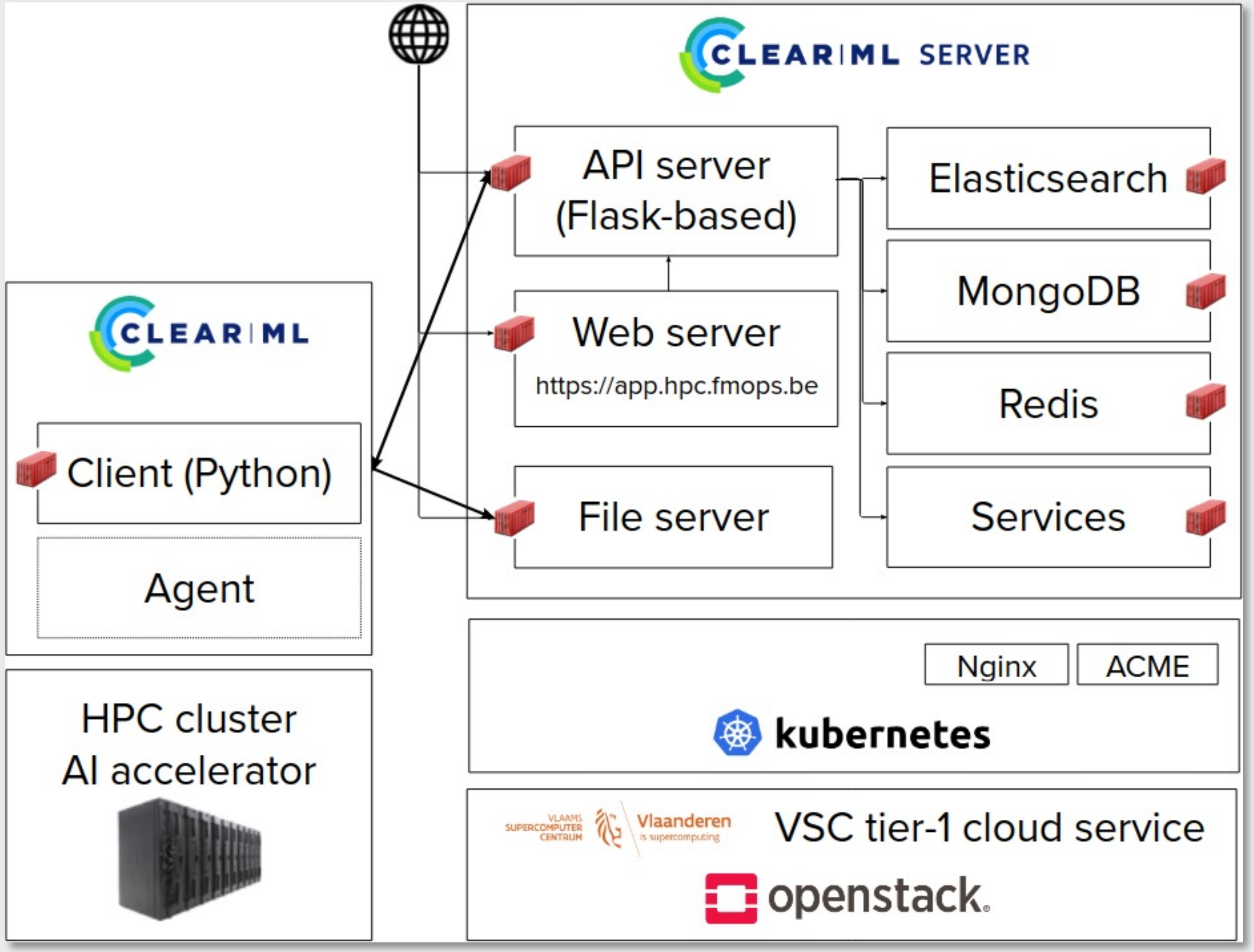


[11] https://www.graphcore.ai

* Base case
** Test was done for smaller test case than the others due to compatibility issues

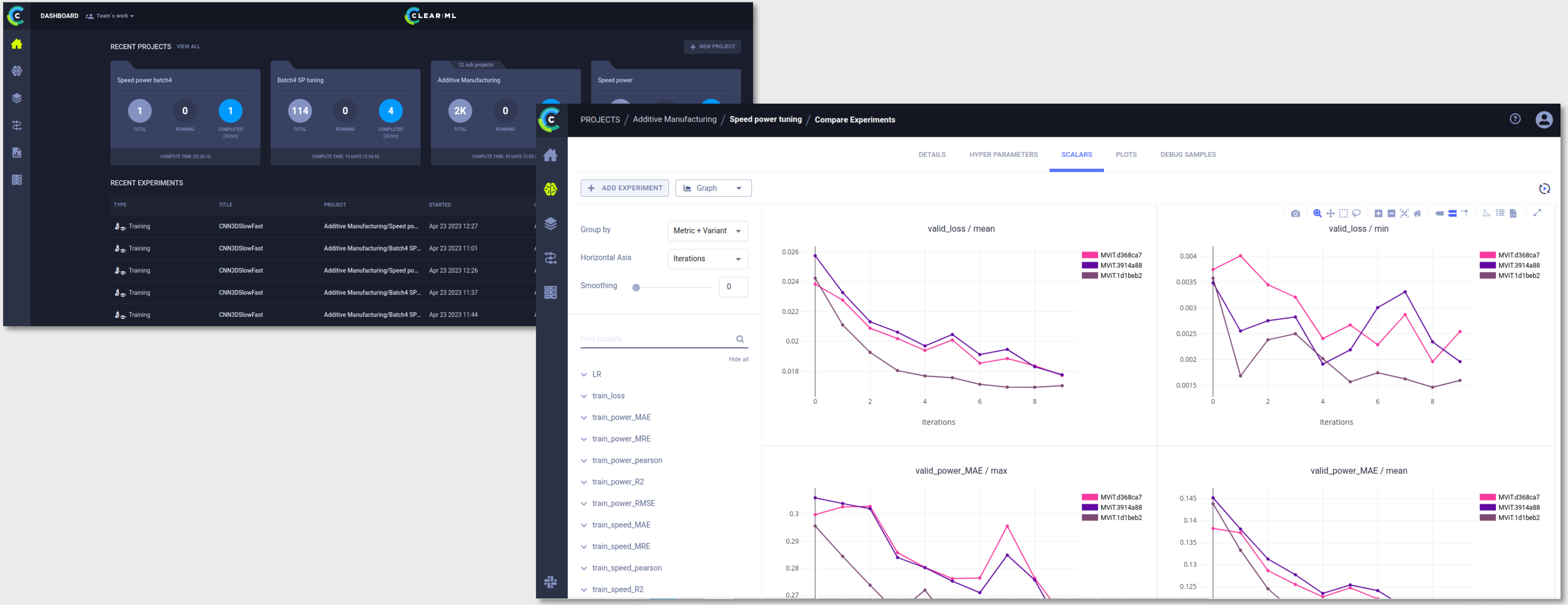
ClearML Server in an HPC Environment

- ClearML as MLOps toolchain
- Experiment with HPC
- Deployed on an OpenStack cloud platform at an HPC center connected to GEANT
- AI training jobs run on HPC systems and report to ClearML



Model Comparison in ClearML‘s Webinterface

RAISE
Center of Excellence



Load AI Modules, Environments, and Containers (LAMEC)

- Modules vary heavily between different HPC systems
- AI developers spend 2-3 days per months setting up the right environment on HPC systems

Goal: simplify setup of components

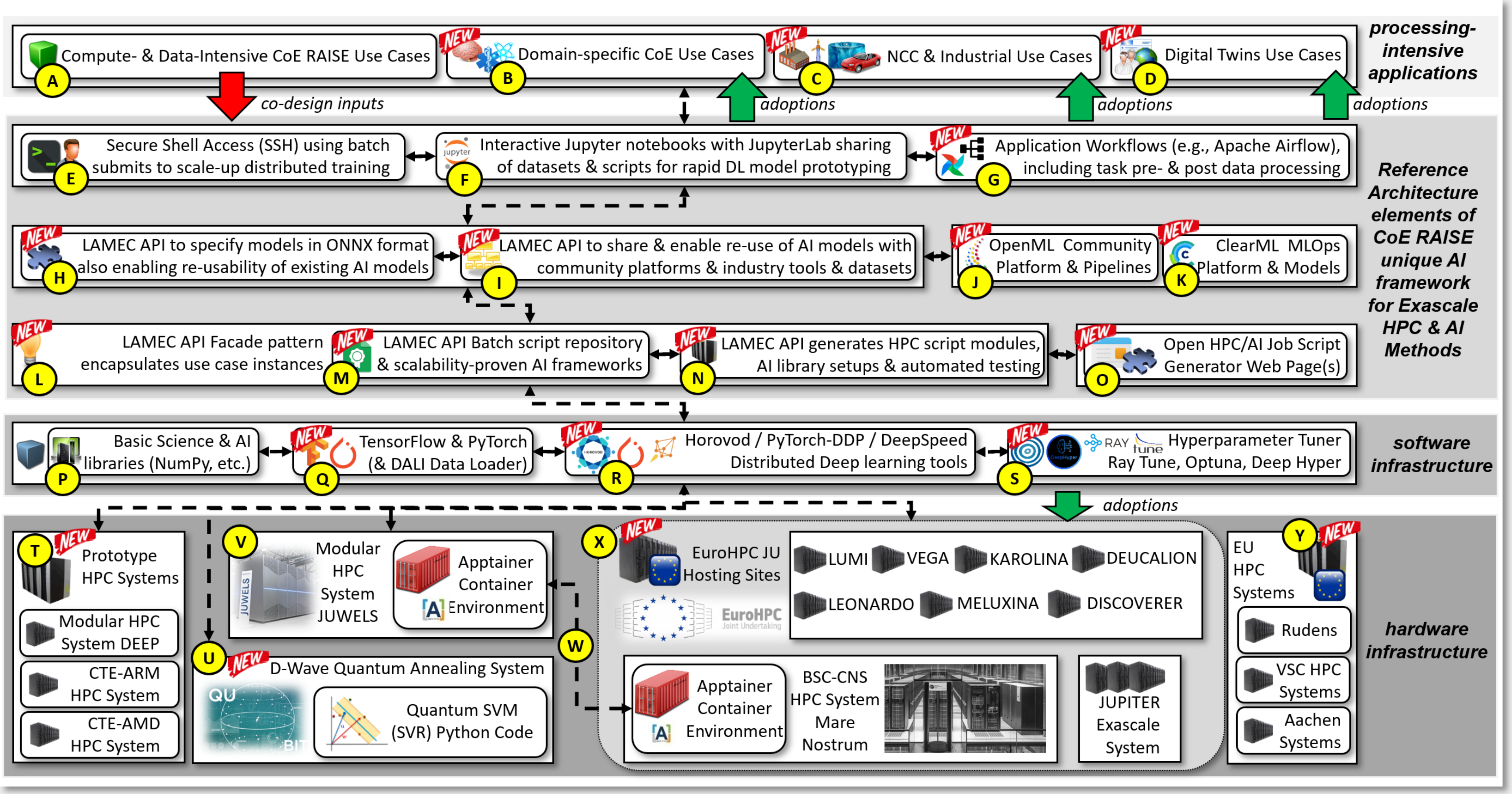
- LAMEC [12,13] is an automatic job script generator selecting the right module setup

Deep_DDP	important bug fix	3 months ago
Deep_DeepSpeed	Deepspeed in Deep	6 months ago
Deep_HeAT	Jureca additions	5 months ago
Deep_Horovod	Deep modifications for Horovod and fex bu...	6 months ago
Deep_TensorFlow	initial TF push	5 months ago
HELPER_Scripts	fix tqdm bug	4 months ago
Jureca_DDP	latest fixes	1 month ago
Jureca_DeepSpeed	latest fixes	1 month ago
Jureca_Graphcore	added Graphcore dir and fixed lrank in CASES	2 months ago
Jureca_HeAT	latest fixes	1 month ago
Jureca_Horovod	latest fixes	1 month ago
Jureca_LibTorch	initial libtorch push	1 month ago
Jureca_RayTune	Update Jureca_RayTune/create_jureca_env.sh	3 months ago
Juwels_DDP	Update README.md	3 months ago
Juwels_Turbulence	merge	9 months ago
PARAMETER_TUNING	Update PARAMETER_TUNING/Autoencoder/...	3 months ago

```
#!/usr/bin/env bash

# Slurm job configuration
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=4
#SBATCH --cpus-per-gpu=20
#SBATCH --account=hai_so2sat
#SBATCH --output=output.out
#SBATCH --error=error.er
#SBATCH --time=6:00:00
#SBATCH --job-name=BENTF2
#SBATCH --gres=gpu:1 --partition=booster

#load modules
ml Stages/2020  GCC/9.3.0  OpenMPI/4.1.0rc1
ml Horovod/0.20.3-Python-3.8.5
ml TensorFlow/2.3.1-Python-3.8.5
#activate my virtualenv
#source /p/project/joaiml/remote_sensing/rocco_sedona/ben_TF2/scripts/env_tf2_juwels_booster/bin/activate

#export relevant env variables
#export CUDA_VISIBLE_DEVICES="0,1,2,3"

#run Python program
srun --cpu-bind=none python -u train_hvd_keras_aug.py
```


[12] UAIF https://gitlab.jsc.fz-juelich.de/CoE-RAISE/FZJ/ai4hpc
[13] LAMEC https://gitlab.jsc.fz-juelich.de/CoE-RAISE/FZJ/lamec-oa

UAIF: The Unique AI Framework



drive. enable. innovate.

RAISE
Center of Excellence

The CoE RAISE project has received funding from
the European Union's Horizon 2020 –
Research and Innovation Framework Programme
H2020-INFRAEDI-2019-1 under grant agreement no. 951733

Follow us: