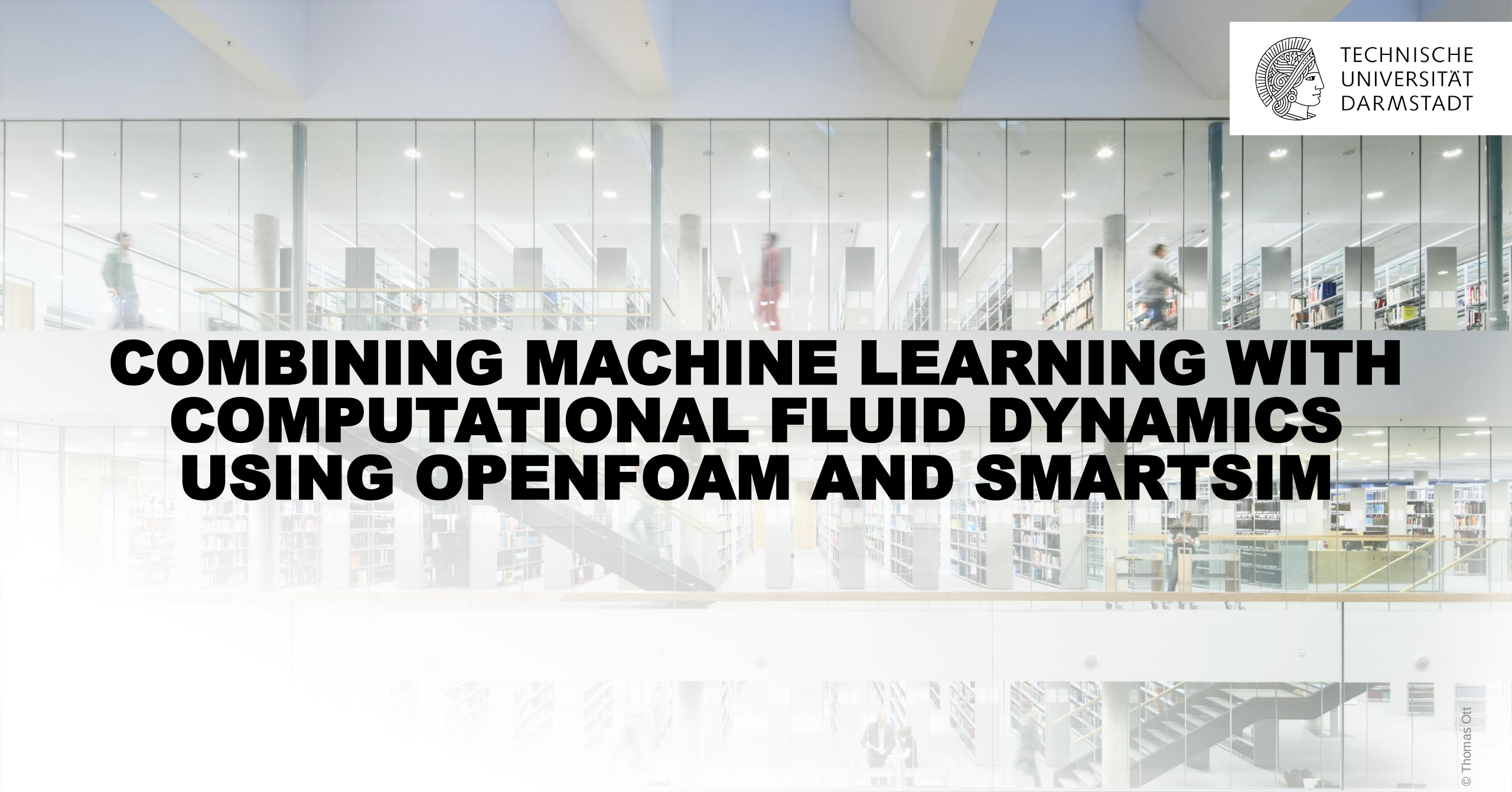


TECHNISCHE UNIVERSITÄT DARMSTADT

# COMBINING MACHINE LEARNING WITH COMPUTATIONAL FLUID DYNAMICS USING OPENFOAM AND SMARTSIM




T. Maric (TU Darmstadt) | A. E. Shao (HPE) | A. Rigazzi (HPE) | M. Ellis (HPE) | E. Fadeli (TU Darmstadt) | Y. Kai (Intel) | A. Weiner (TU Dresden)

# DATA-DRIVEN MODELING SPECIAL INTEREST GROUP

TECHNISCHE UNIVERSITÄT DARMSTADT

Information about the Data-Driven SIG:

https://wiki.openfoam.com/Data_Driven_Modelling_Special_Interest_Group

We organize hackathons on combining ML+CFD in OpenFOAM

https://github.com/OFDataCommittee/OFMLHackathon

## Combining CFD and ML with OpenFOAM and SmartSim

https://github.com/OFDataCommittee/openfoam-smartsim

https://doi.org/10.1007/s11012-024-01797-z

OpenFOAM-SmartSim as an OpenFOAM Module

OpenFOAM + SmartSim OFW18 Training

OpenFOAM

SMARTSIM

# ML+CFD DATA PATTERNS I

TECHNISCHE UNIVERSITÄT DARMSTADT

## Generate Training Data

Simulation → Data

Simulation → Data

...

Simulation → Data

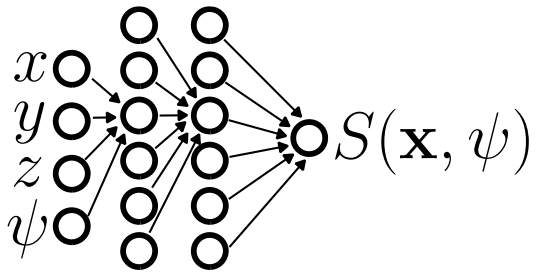
## Train the Model

Data

Data

...

Data




Wait

## Use the Model

$$\partial_t \psi + \nabla \cdot (\boldsymbol{v}\psi)$$
$$-\nabla \cdot (\Gamma \psi) = S(\boldsymbol{x}, \psi)$$

Wait

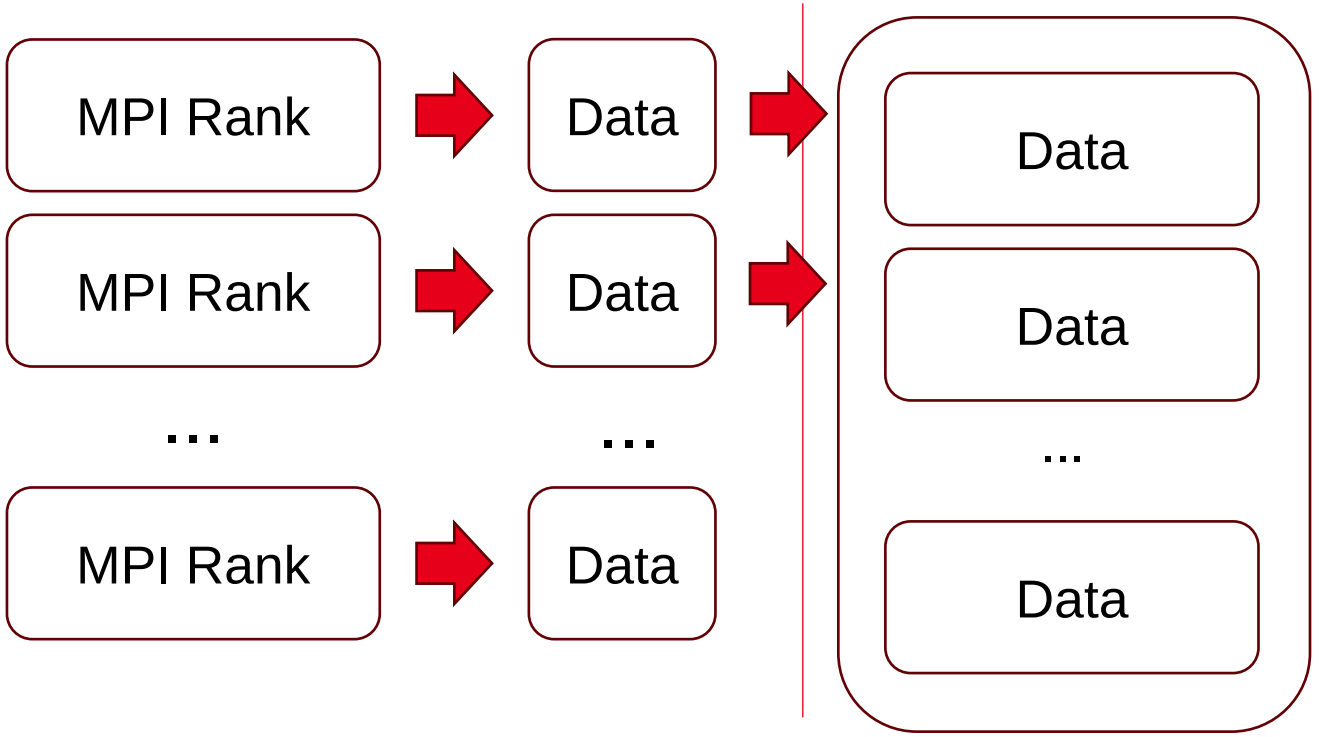
# ML+CFD DATA PATTERNS II

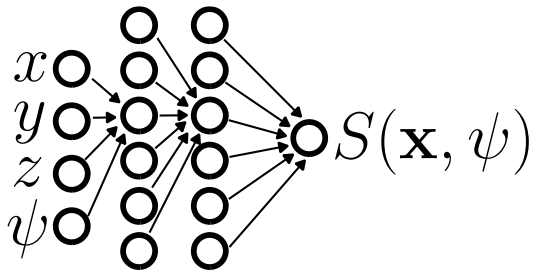
TECHNISCHE UNIVERSITÄT DARMSTADT

Generate Training Data

Train the Model Incrementally

Use the Model



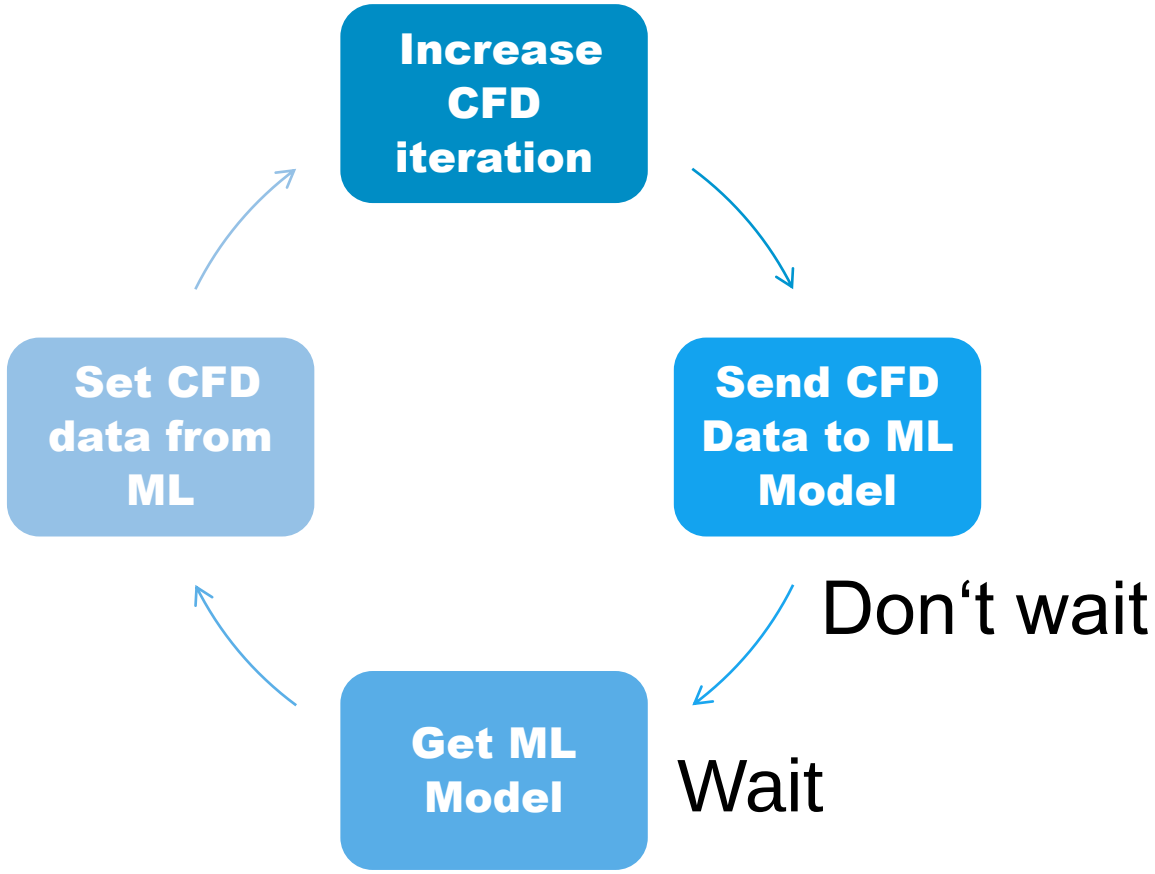

$$\partial_t \psi + \nabla \cdot (\boldsymbol{v}\psi)$$
$$-\nabla \cdot (\Gamma \psi) = S(\boldsymbol{x}, \psi)$$

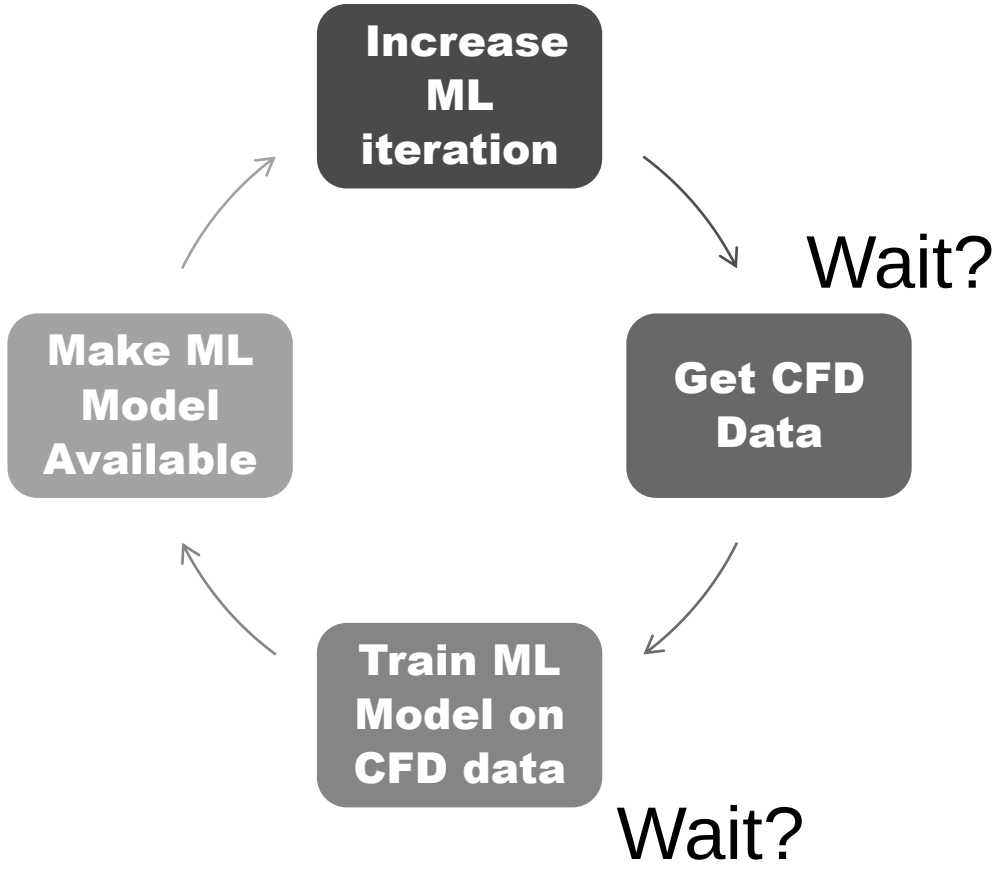
Don‘t wait

Wait

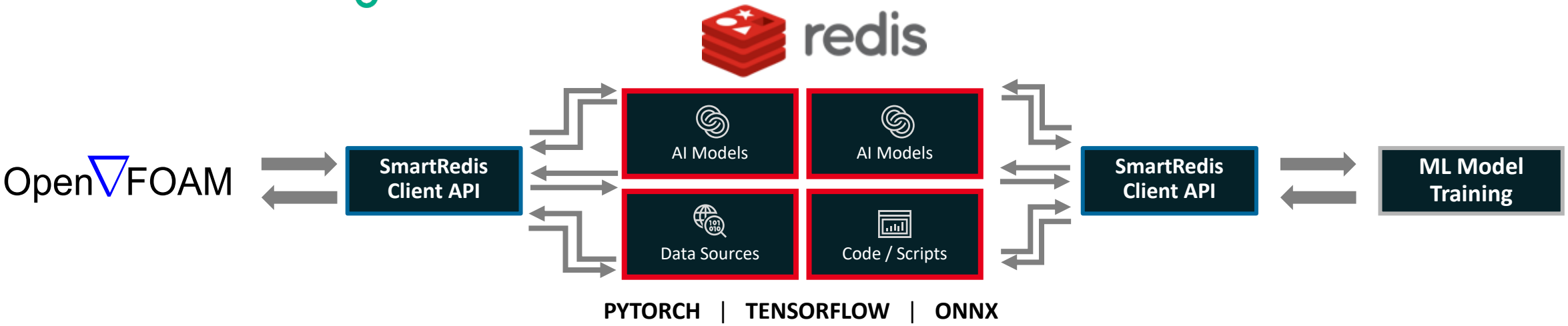
# ML+CFD DATA PATTERNS III

TECHNISCHE UNIVERSITÄT DARMSTADT



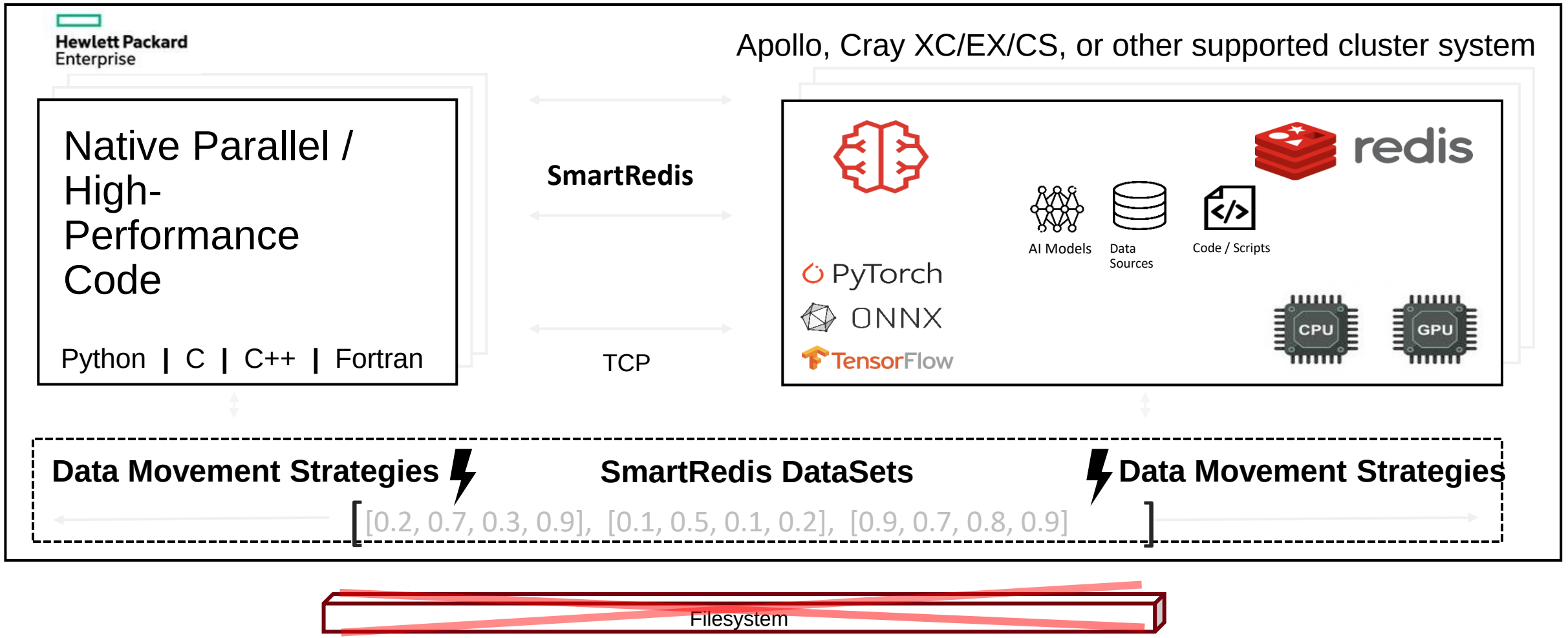
# OPENFOAM + SMARTSIM + ML




- **SmartSim Orchestrator:** implementing the computational workflow.
  - Jupyter Notebook or Python script – straightforward API.
- **SmartRedis Database**: CFD data, trained model, model inference.
  - Straightforward API in C++ (!!) and Python.

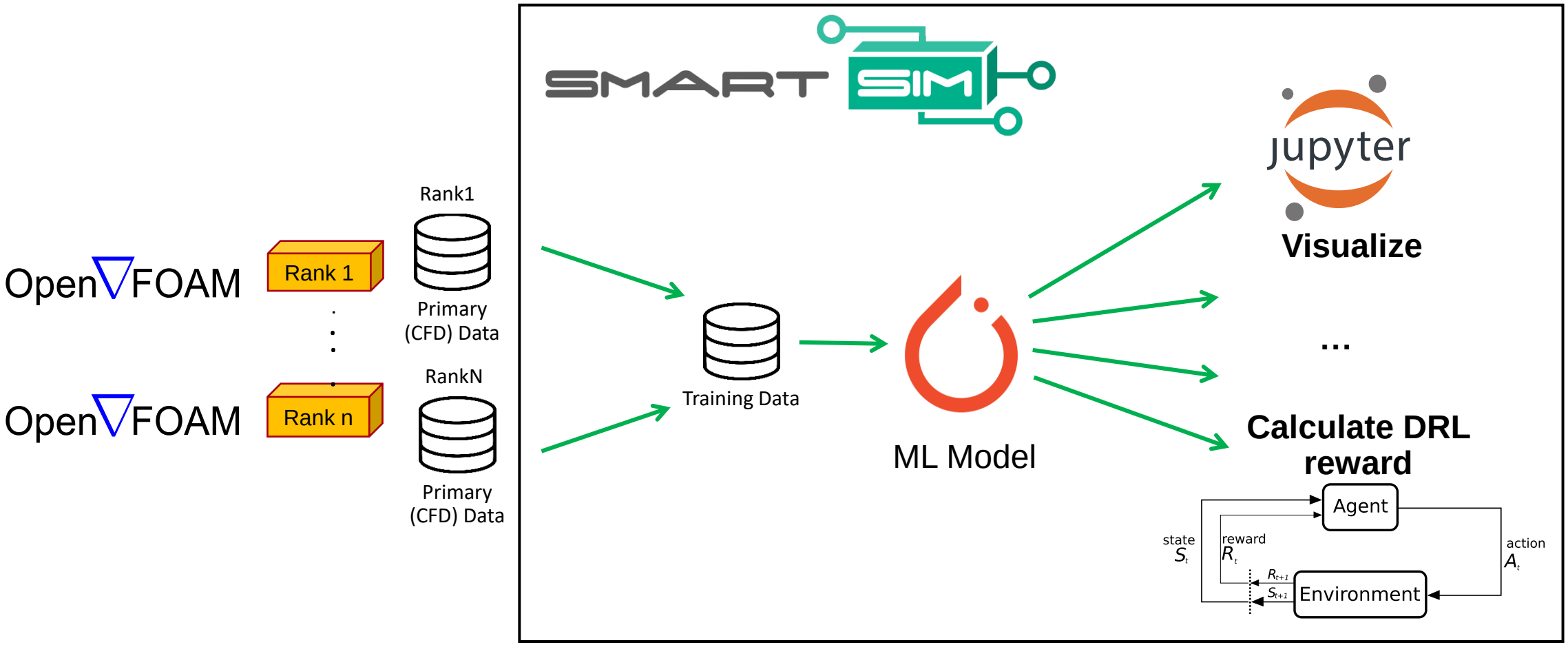
TECHNISCHE UNIVERSITÄT DARMSTADT

# SMART REDIS DATABASE



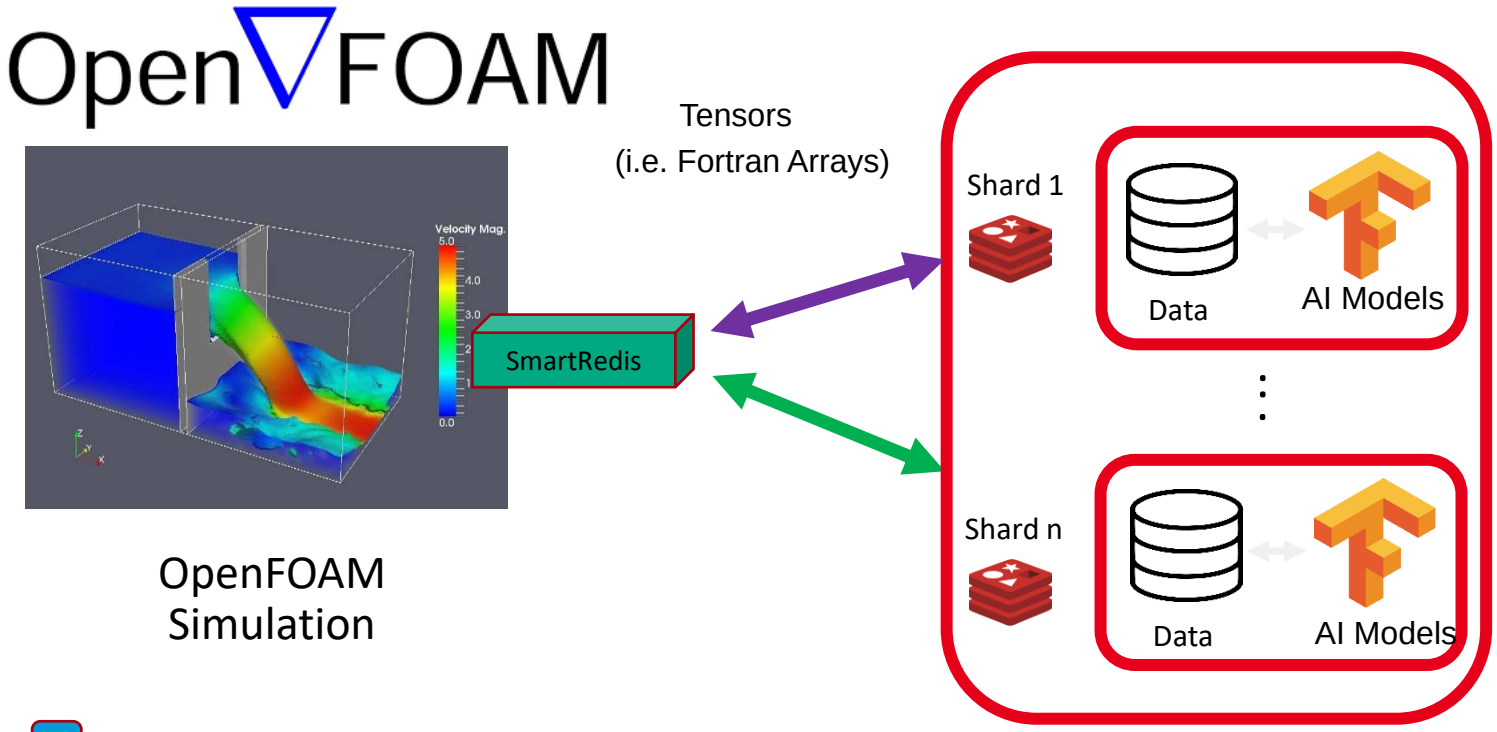
# ONLINE POST-PROCESSING

**On-line Machine Learning** while the simulation is running.



# ML FOR TURBULENCE PREDICTION IN OPENFOAM

- Use a machine-learning model to reduce my time-to-solution.

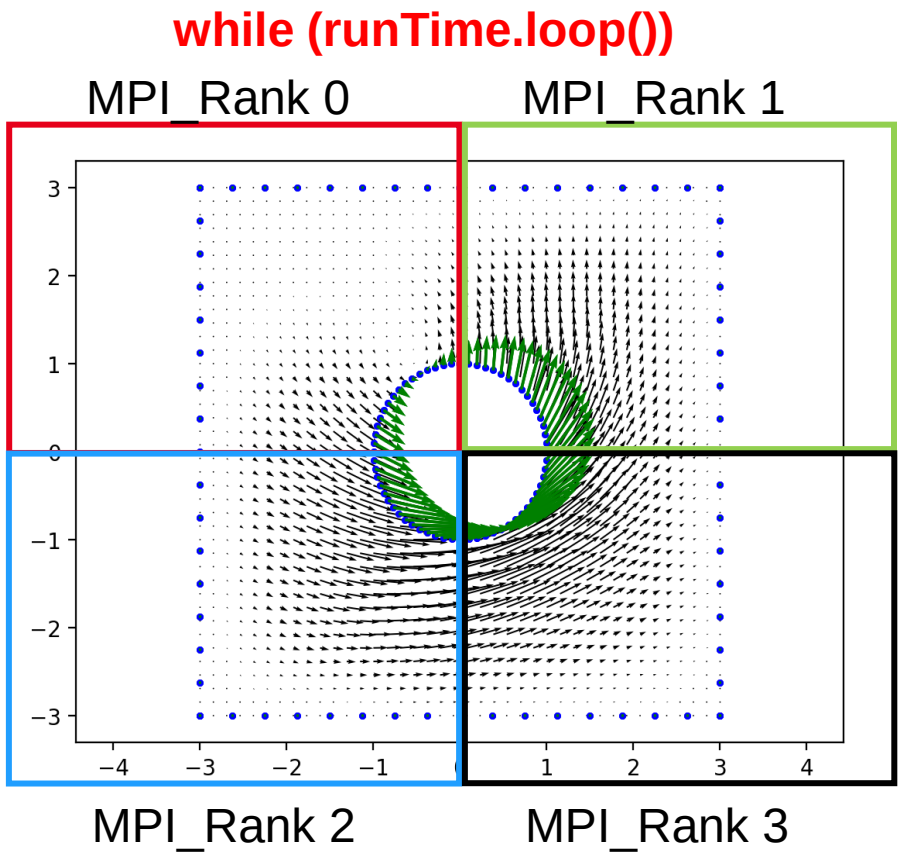



- **Tensorflow model** provides a pre-conditioned state that **reduces** number of solver **iterations**
- **Multi-stage** workflow in **one** Python **script**
  - Mesh decomposition
  - ML training
  - ML inference
  - Solver Integration
  - Analysis
  - Visualization

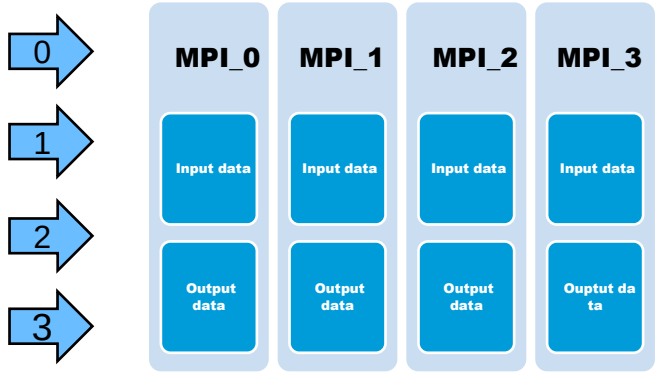
Code repo: https://github.com/CrayLabs/smartsim-openFOAM

# ONLINE ML MESH MOTION

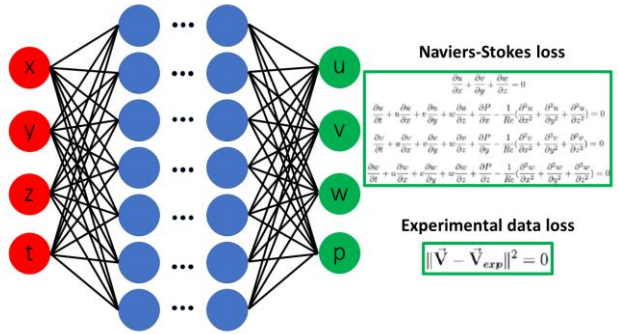
- Use a machine-learning model to approximate mesh-motion displacements.




Training **Data**




- Agglomerates training data.
- Trains on other resources.




Physics-Informed Neural Network
Riccardo Munafò, CC BY-SA 4.0

- **Online training and inference requires synchronization with the CFD algorithm.**

# OPENFOAM FIELDS ARE INTERPRETABLE AS TENSORS

```
template<class T>
class UList
{
    // Private Data

        //- Number of elements in UList
        label size_;

        //- Vector of values of type T
        T* __restrict__ v_;
```

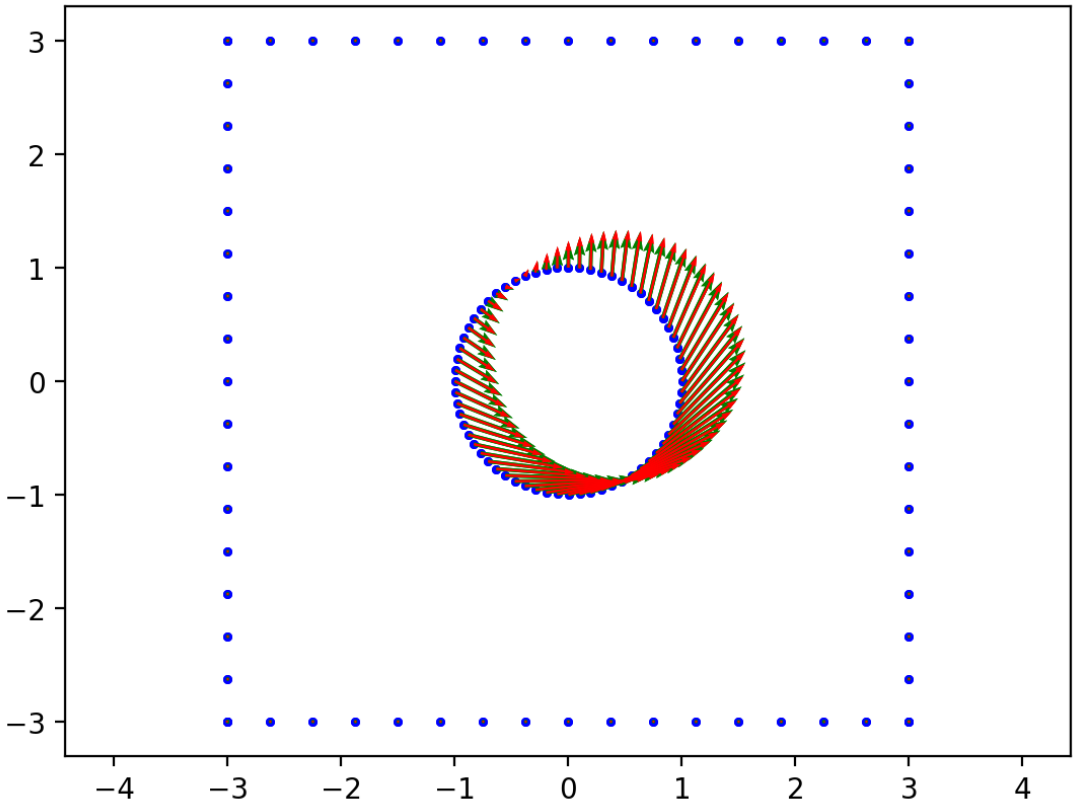
- All OpenFOAM Fields are ULists.
- UList<T> is a wrapper for T*
- Fields provide cdata() to reinterpret them as (void*).
- This is necessary not only for interpreting OpenFOAM fields as SmartRedis tensors, but also tensors in Machine Learning Frameworks.

TECHNISCHE UNIVERSITÄT DARMSTADT

# ONLINE MACHINE LEARNING FOR MESH MOTION

# ONLINE ML MESH MOTION

- **User story:** I want to use a machine-learning model to approximate mesh-motion displacements.




Given Dirichlet (fixed value) conditions for mesh motion on mesh boundary patches $\{\partial\Omega_k\}_k$

- We use displacements $\delta(x,\, t) \in \Omega$

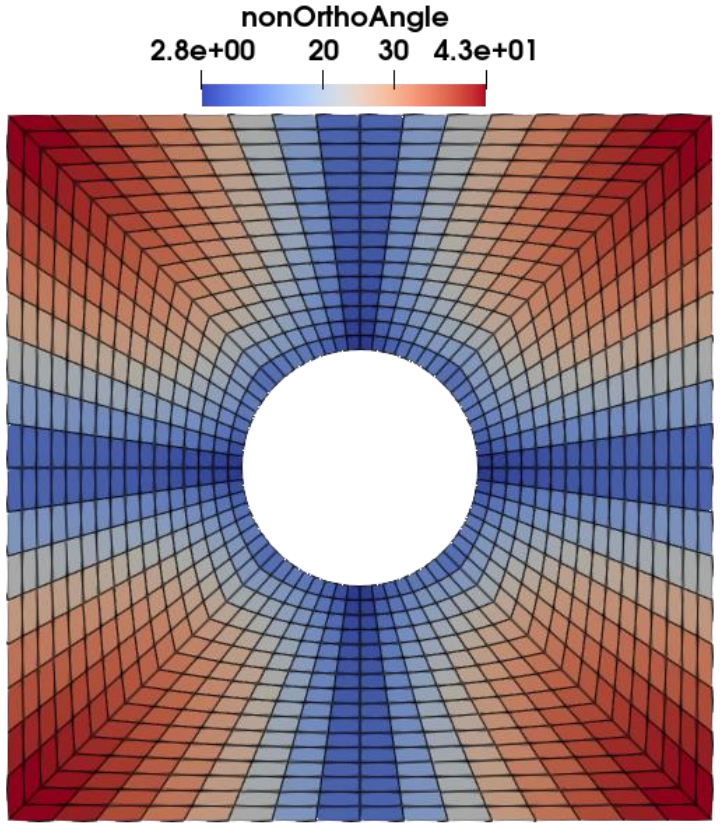
Approximate displacements in the solution domain as

$$\tilde{\delta}(x, t, \theta)$$

with $\theta$ as model parameters.

# PDE-BASED MESH MOTION

- **Why** use a machine-learning model for mesh-motion?




PDE-based mesh motion like the Laplacian mesh motion

$$\nabla \cdot (\lambda \nabla \delta) = 0$$

$$\delta(x,t) = d(x,t),\ \forall x \in \{\partial\Omega_k\}_{\{k \in K\}}$$

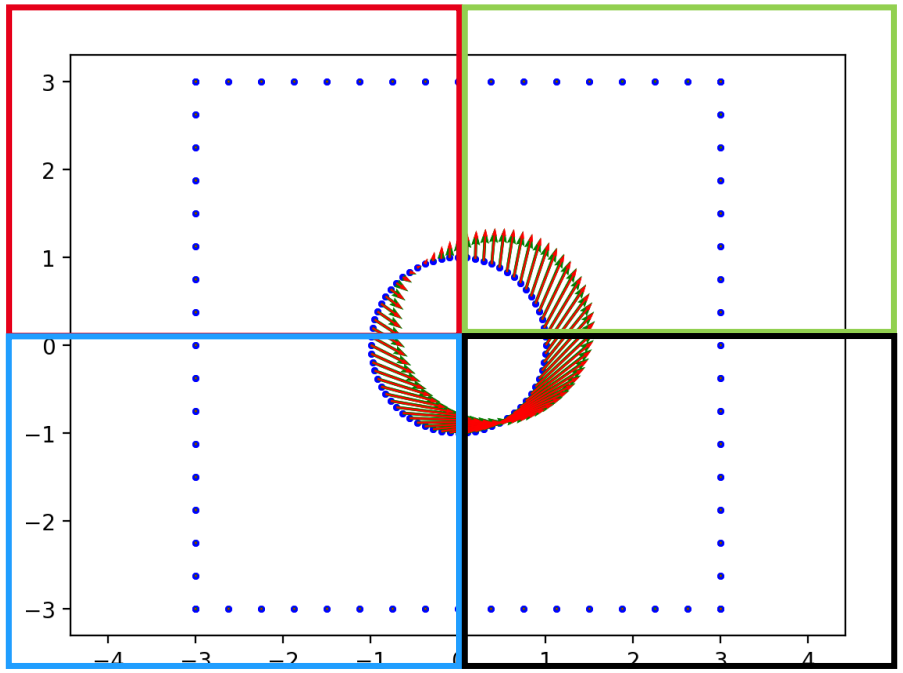
- Approximatively solves the PDE on a deforming mesh with increasingly deteriorating quality (e.g. non-orthogonality).
- Approximation or interpolation-based mesh motion is smoother and can potentially deliver higher mesh quality for stronger deformations.

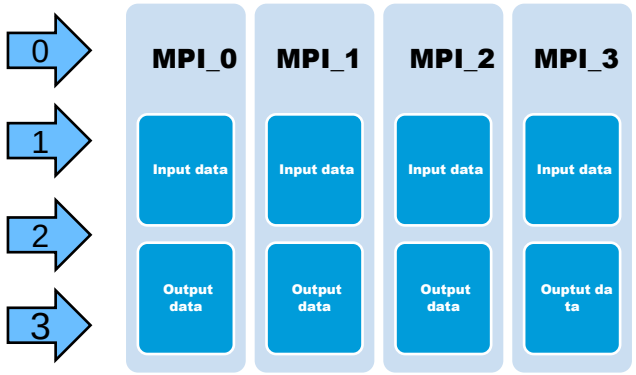
De Boer, A., Van der Schoot, M. S., & Bijl, H. (2007). Mesh deformation based on radial basis function interpolation. *Computers & structures*, *85*(11-14), 784-795.

# OPENFOAM DATA STRUCTURE

OpenFOAM

redis

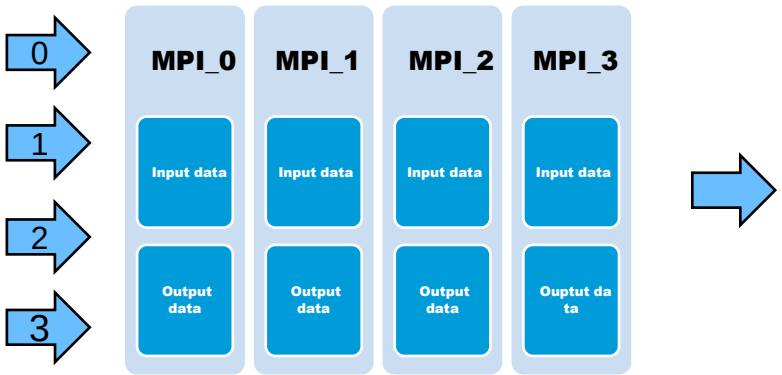



- Each MPI rank sends its own data to increase efficiency.
- We don't need MPI process boundary data.
- We don't need empty patches.
- Boundary field of a pointField stores a list of all boundary patches – zero length for those not on the MPI Rank.
- Input data are datasets:
  - point_patch_timeStep_rank
  - displ_patch_timeStep_rank

**runTime++;**

# DATASETS AND AGGREGATION LISTS

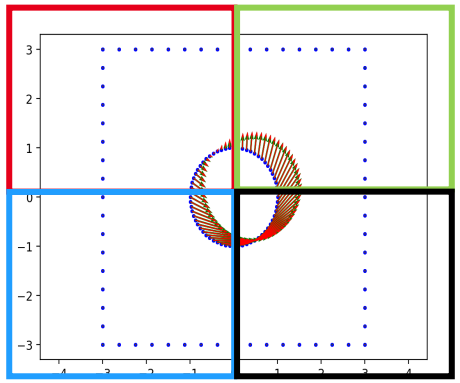



- Input data are datasets:
  - point_**patch**_timeStep_**rank**
  - displ_**patch**_timeStep_**rank**
- **What does this looks like for 30 patches and 50 MPI ranks?**
  - How to check if this data is available in SmartRedis?
- **Dataset: agglomerate over patches**
  - pointsDataset_timeStep_rank
  - displacementsDataSet_timeStep_rank
- **Aggregation list: agglomerate datasets over time steps**
  - If we know MPI_Comm_size and the time index, we know if all the data is available in SmartRedis. **How?**

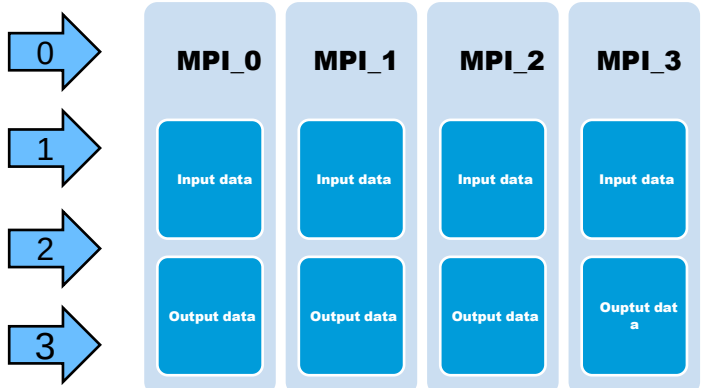
# SYNCHRONISE TIME STEPS

OpenFOAM

runTime++;



redis




SMARTSIM

runTime++;

- **Aggregation list length**
  - **MPI_Comm_size * timeIndex**

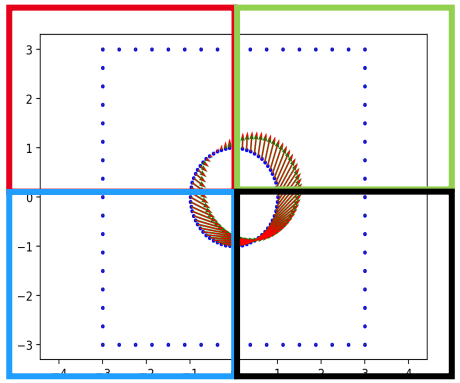
```
client.poll_list_length("displacementsDatasetList",
                        time_index * num_mpi_ranks,
                        10, 1000);
```

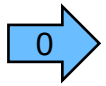
- **Is there anything else?**

# SYNCHRONISE TIME STEPS

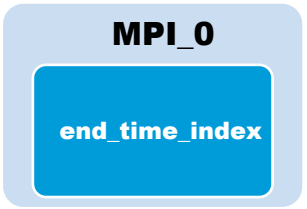
OpenFOAM

runTime++;






redis




SMARTSIM

runTime++;

- **How does SmartSim Python script know that the OpenFOAM simulation has ended?**

```
if client.poll_key("end_time_index", 10, 100):
        print ("End time reached.")
        break
```

TECHNISCHE UNIVERSITÄT DARMSTADT

# ONLINE ML MESH MOTION

- **Let's look at some source code.**