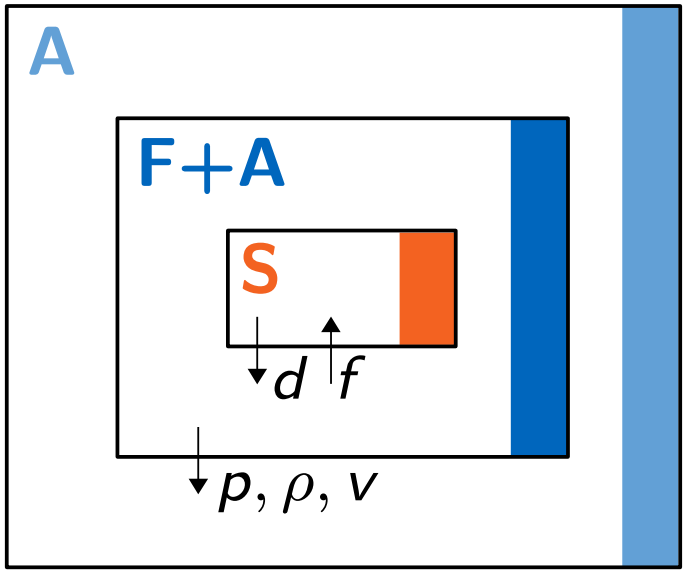
Scalable and Easy-to-use Coupling of Multi-Physics Simulations

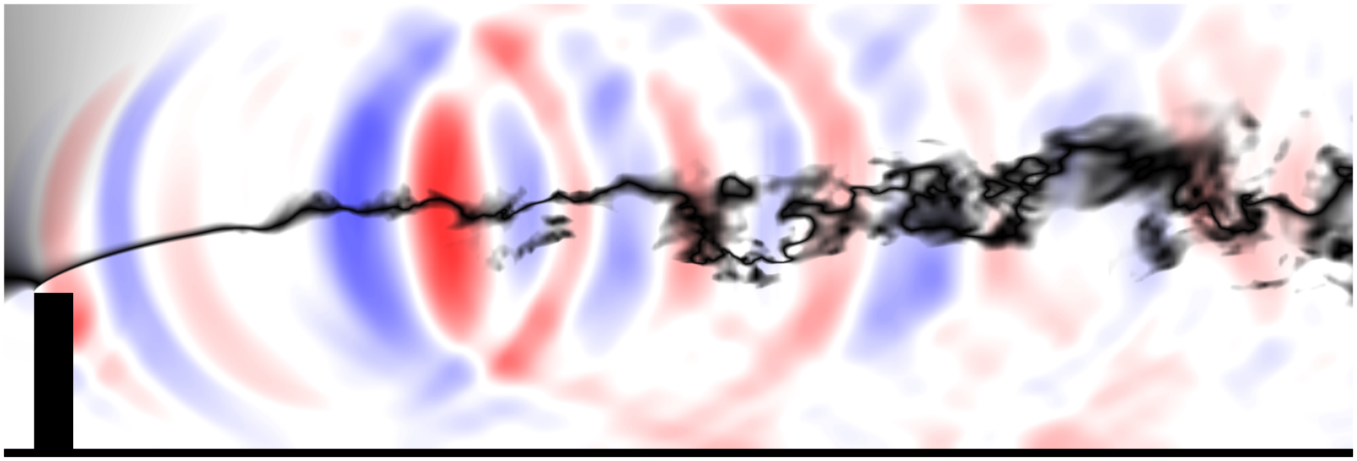
Benjamin Uekermann et al.,

Eindhoven University of Technology

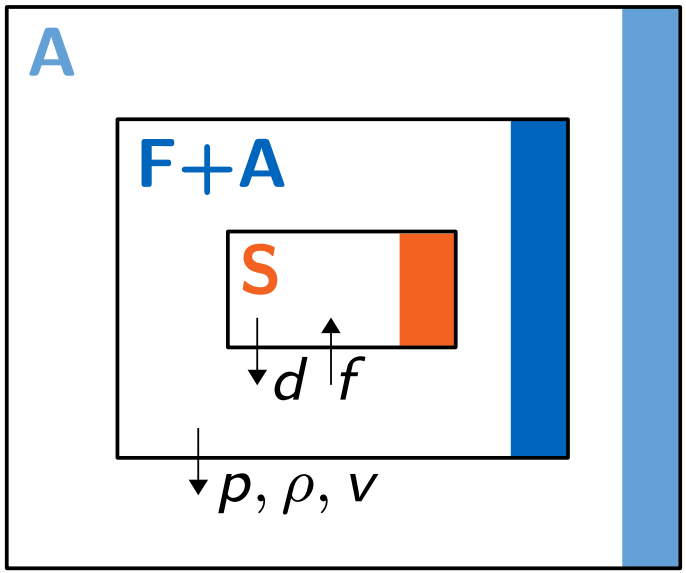
The ExaFSA Challenge



- **S**: FEM structure solver
- **F+A**: Low-order FV solver, resolve geometries, resolve turbulent structures
- **A**: High-order DG solver, linearized Euler, transport phenomena, very large domain



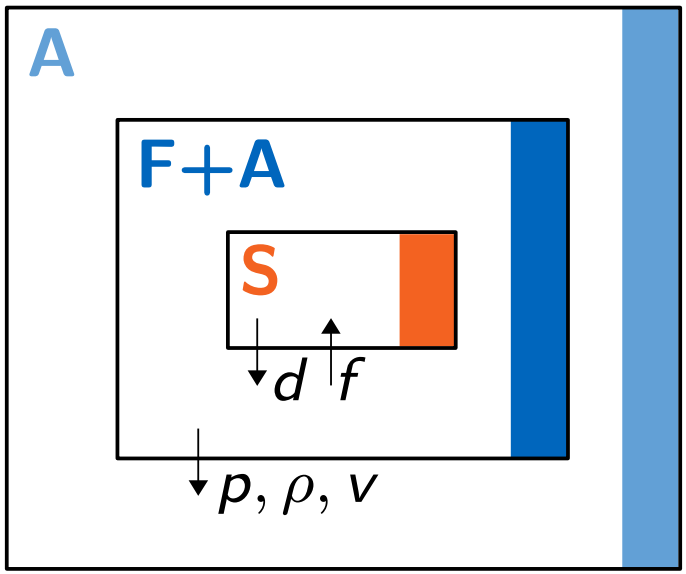
The ExaFSA Challenge



- S: FEM structure solver
- F+A: Low-order FV solver, resolve geometries, resolve turbulent structures
- A: High-order DG solver, linearized Euler, transport phenomena, very large domain

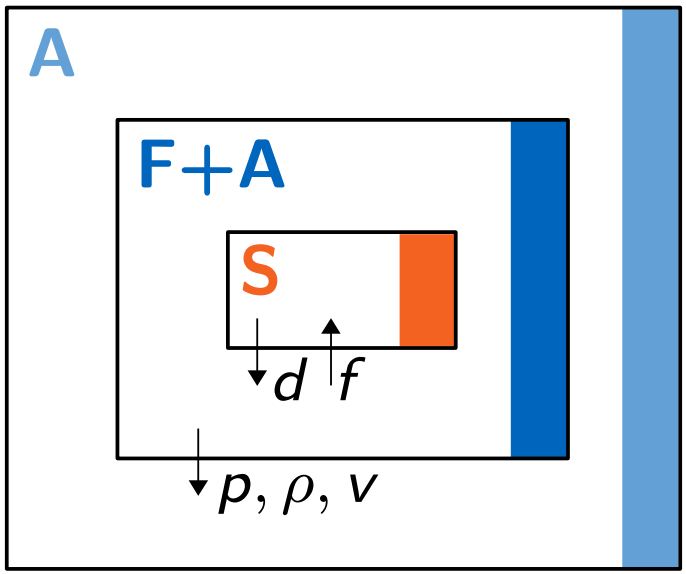
- Taylored numerical methods necessary for each subdomain
  $\Rightarrow$ partitioned approach
- Multi-scale $\Rightarrow$ HPC

The ExaFSA Challenge



- **S**: FEAP or CalculiX
- **F+A**: FASTEST (TU Darmstadt, Thorsten Reimann, Dörte Sternel)
- **A**: Ateles (U Siegen, Verena Krupp, Neda Ebrahimi Pour, Harald Klimach, Sabine Roller)
- **Performance Portability**: Xevolver (Tohoku, Hiro Takizawa, Ryusuke Egawa et al.)

The ExaFSA Challenge



- **S**: FEAP or CalculiX
- **F+A**: FASTEST (TU Darmstadt, Thorsten Reimann, Dörte Sternel)
- **A**: Ateles (U Siegen, Verena Krupp, Neda Ebrahimi Pour, Harald Klimach, Sabine Roller)
- **Performance Portability**: Xevolver (Tohoku, Hiro Takizawa, Ryusuke Egawa et al.)

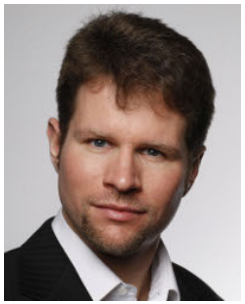
Coupling: preCICE

- . . . has to be scalable and easy-to-use
- 1. **Easy-to-use Coupling**: flexible and minimally-invasive integration into existing legacy codes
- 2. **Scalable Coupling**: don't deteriorate the scalability of the coupled single-physics solvers by the coupling

preCICE Contributors

Miriam Mehl

U Stuttgart



Florian Lindner

U Stuttgart

Amin Totounferoush

U Stuttgart

Kyle Davis

U Stuttgart

Alexander Rusch

ETH Zürich

Hans Bungartz

TUM

Benjamin Rüth

TUM

Gerasimos Chourdakis

TUM

Frédéric Simonis

TUM

Benjamin Uekermann

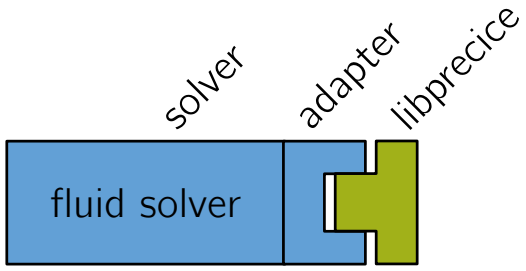
TU/e

Previous and further contributors:

- Bernhard Gatzhammer, Klaudius Scheufele, Lucia Cheung, Alexander Shukaev, Peter Vollmer, Georg Abrams, Alex Trujillo, Dmytro Sashko, David Sommer, David Schneider, Richard Hertrich, Saumitra Joshi, Peter Meisrimel, Derek Risseeuw, Rafal Kulaga, Ishaan Desai ...

1. Easy-to-use Coupling

Easy-to-use: We need a library approach (. . . not a framework)



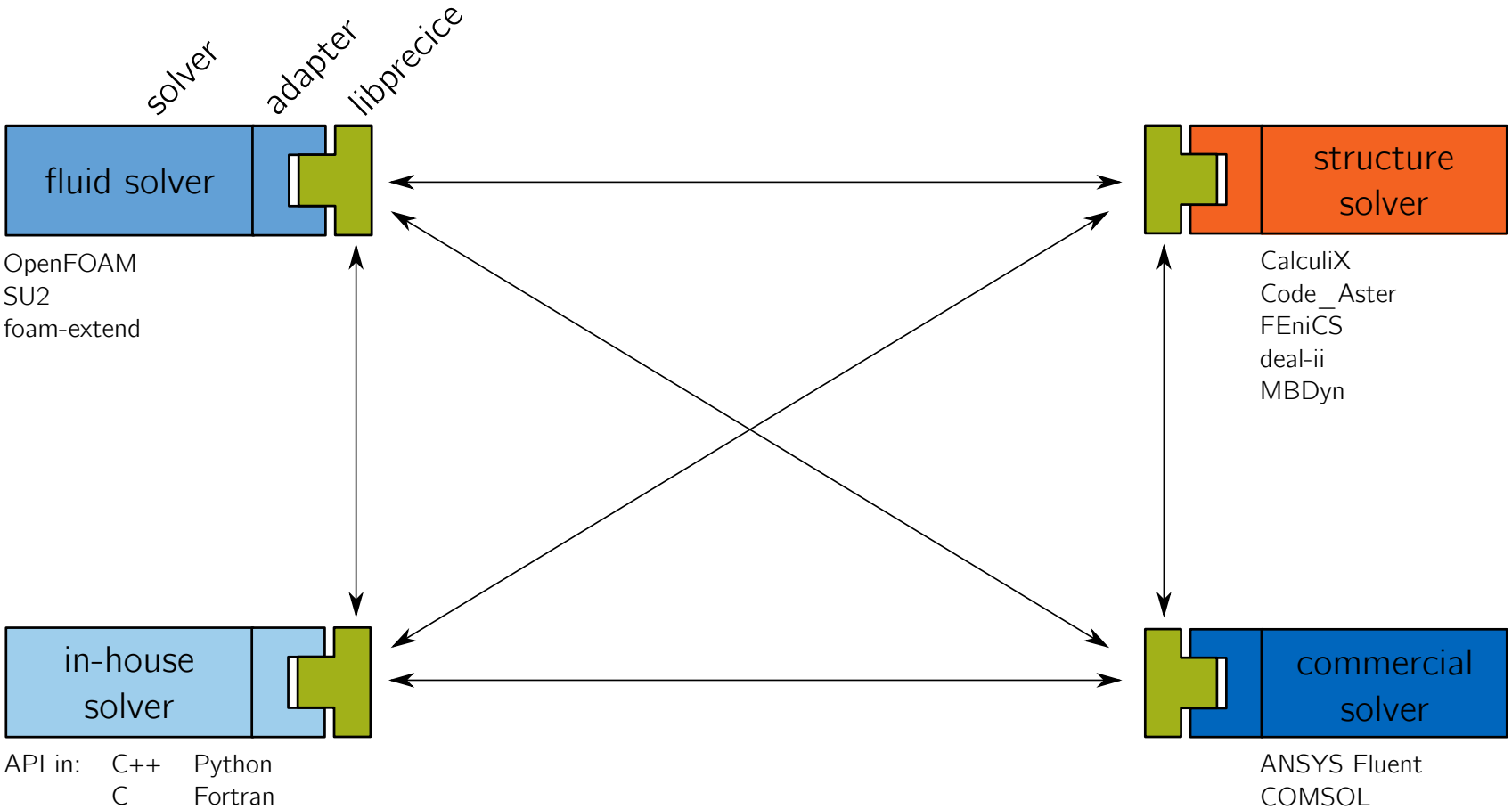
OpenFOAM
SU2
foam-extend

Easy-to-use: We need a library approach (. . . not a framework)

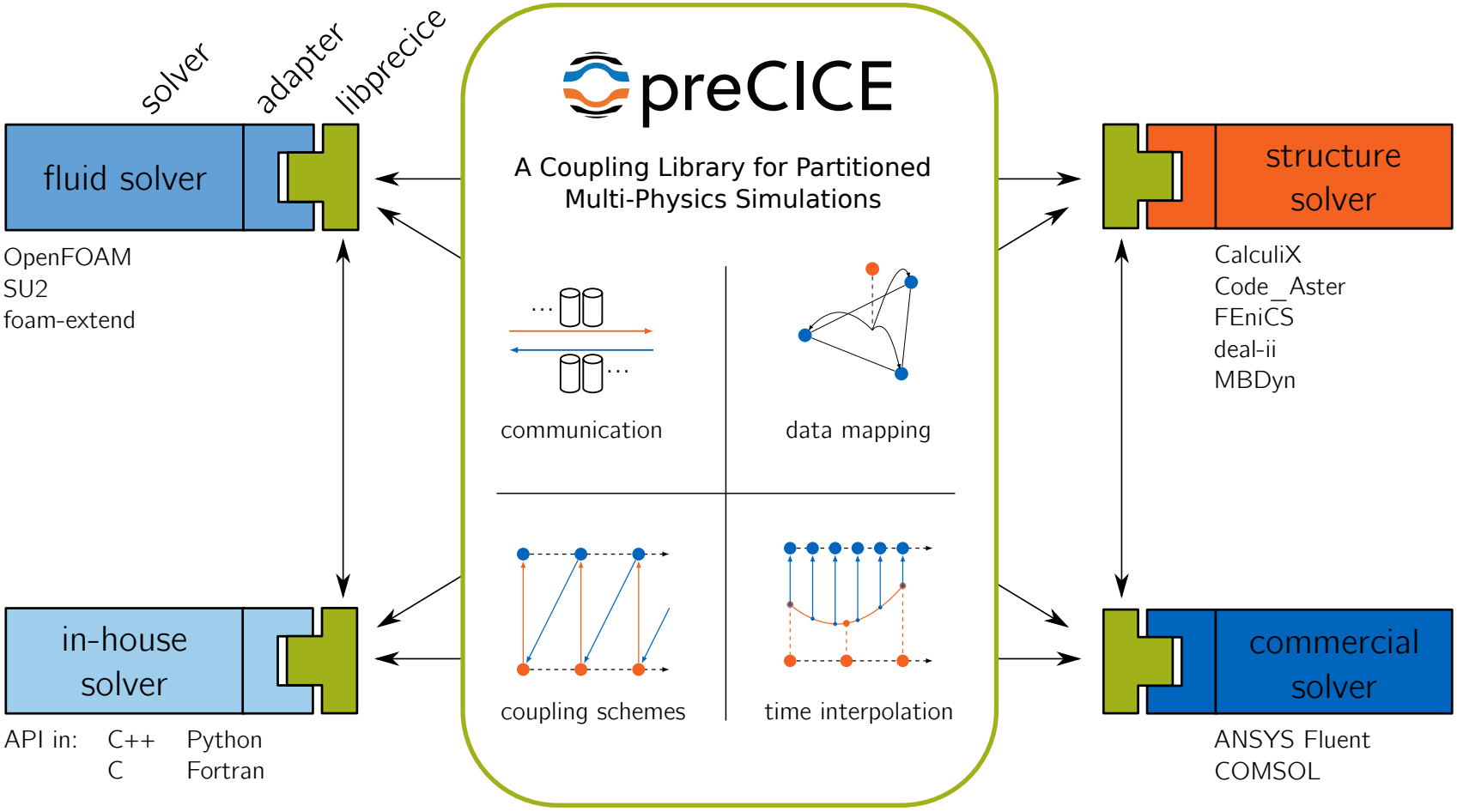
OpenFOAM
SU2
foam-extend

CalculiX
Code_Aster
FEniCS
deal-ii
MBDyn

Easy-to-use: We need a library approach (. . . not a framework)



Easy-to-use: We need a library approach (...not a framework)



Easy-to-use: We need a high-level API

```
1     precice::SolverInterface
2     precice("FluidSolver",rank,size);
3     precice.configure("precice-config.xml");
4     precice.setMeshVertices();
5     precice.initialize();
6
7     while (precice.isCouplingOngoing()) { // main time loop
8         solve();
9
10        precice.writeBlockVectorData();
11        precice.advance();
12        precice.readBlockVectorData();
13
14        endTimeStep(); // e.g. write results, increase time
15    }
16
17    precice.finalize();
```

Timesteps, most arguments and less important methods omitted. Full example in the wiki.

Easy-to-use: We need ready-to-use adapters

→ **Our flagship: The OpenFOAM Adapter**

```
/* Start the solver */

Info<<"\nStarting time loop\n"<< endl;
while (runTime.run()) {
  #include "readTimeControls.H"
  #include "compressibleCourantNo.H"
  #include "setDeltaT.H"

  runTime++;

  /* solve the equations */
  #include "rhoEqn.H"
  while (pimple.loop())
  {
      ...
  }

  runTime.write();
}

/* Finalize */
```

Easy-to-use: We need ready-to-use adapters

→ **Our flagship: The OpenFOAM Adapter**

```
/* Start the solver */

Info<<"\nStarting time loop\n"<< endl;
while (runTime.run()) {
  #include "readTimeControls.H"
  #include "compressibleCourantNo.H"
  #include "setDeltaT.H"

  runTime++;

  /* solve the equations */
  #include "rhoEqn.H"
  while (pimple.loop())
  {
      ...
  }

  runTime.write();
}

/* Finalize */
```

```
// system/controlDict OpenFOAM config file
functions
{
  preCICE_Adapter
  {
    type preciceAdapterFunctionObject;
    libs ("libpreciceAdapterFunctObj.so");
  }
}
```

Easy-to-use: We need ready-to-use adapters

→ Our flagship: The OpenFOAM Adapter

```
/* Start the solver */

Info<<"\nStarting time loop\n"<< endl;
while (runTime.run()) {
  #include "readTimeControls.H"
  #include "compressibleCourantNo.H"
  #include "setDeltaT.H"

  runTime++;

  /* solve the equations */
  #include "rhoEqn.H"
  while (pimple.loop())
  {
      ...
  }

  runTime.write();
}

/* Finalize */
```

```
// system/controlDict OpenFOAM config file
functions
{
  preCICE_Adapter
  {
    type preciceAdapterFunctionObject;
    libs ("libpreciceAdapterFunctObj.so");
  }
}
```

```
// 0/T OpenFOAM config file
interface
{
  type              fixedValue;
  value             uniform 300;
}
// other types: fixedGradient, mixed
```

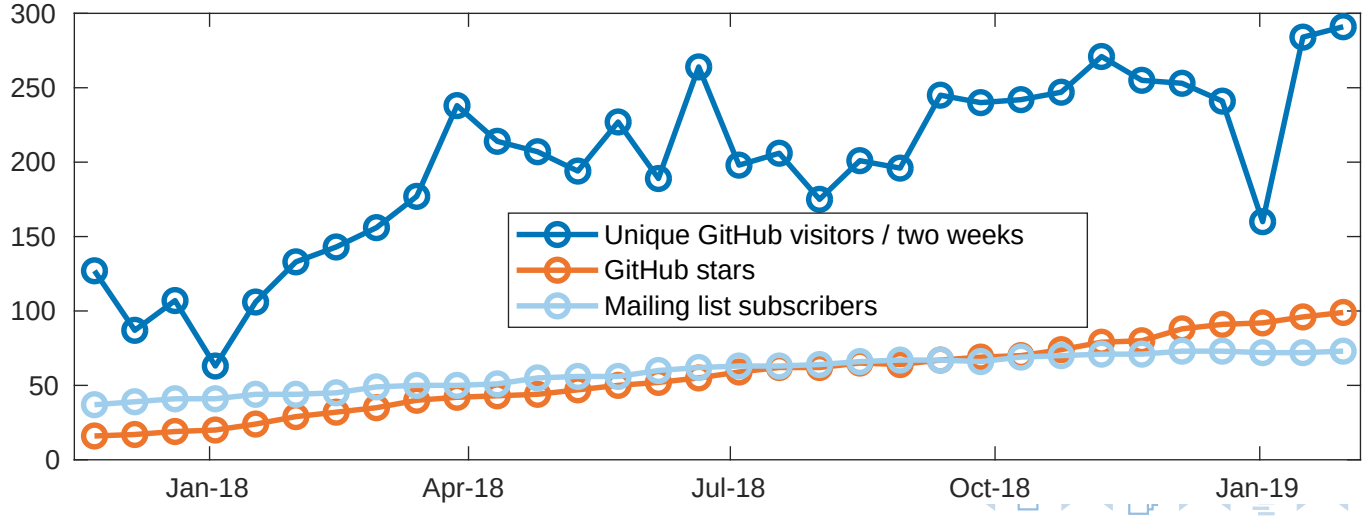
Coupling boundary patches, problem & solver type:
`precice-adapter-config.yml`

It's starting to pay off, our users:

- LSM & STS, U Siegen, Germany
- SC & FNB, TU Darmstadt, Germany
- SCpA, CIRA, Italy
- Cardiothoracic Surgery, UFS, South Africa
- A*STAR, Singapore
- NRG, Petten, The Netherlands
- Aerodynamics & Wind Energy (KITE Power), TU Delft, The Netherlands
- Mechanical and Aeronautical Eng., University of Manchester, UK
- University of Strathclyde, Glasgow, UK
- FAST, KIT, Germany
- AIT, Vienna, Austria
- IAG, University of Stuttgart, Germany
- CTTC UPC, Barcelona, Spain
- Amirkabir U. of Technology, Iran

Upcoming:

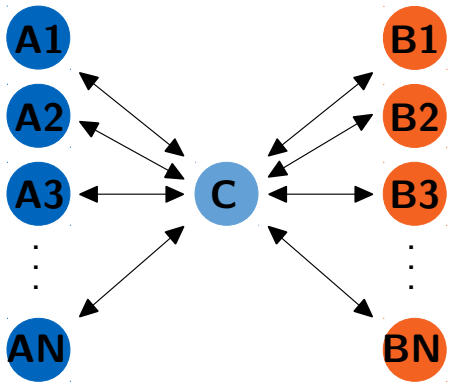
- GRS, Garching, Germany
- MTU Aero Engines, Munich, Germany
- Numerical Analysis, Lund, Sweden
- Helicopter Technology & Astronautics, TUM, Germany
- ATA Engineering Inc., USA
- BITS Pilani, India
- Aviation, MSU Denver, USA



2. Scalable Coupling

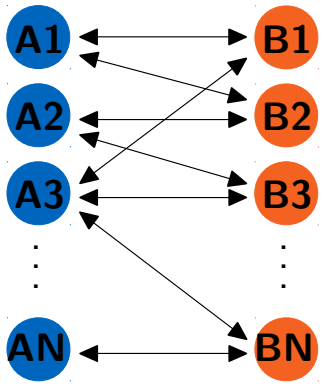
Communication Layout

Server-Based Concept



- Complete communication through central server process
- Interface computations on server (in sequential)
- $\Rightarrow$ Coupling becomes bottleneck for overall simulation already on moderate parallel systems

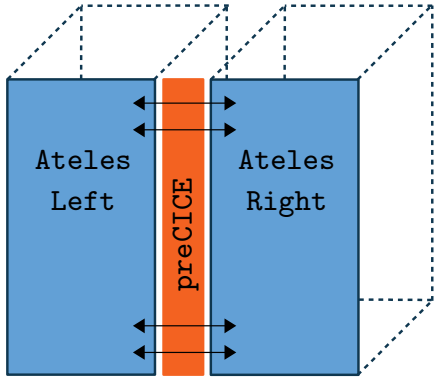
Our Peer-To-Peer Concept

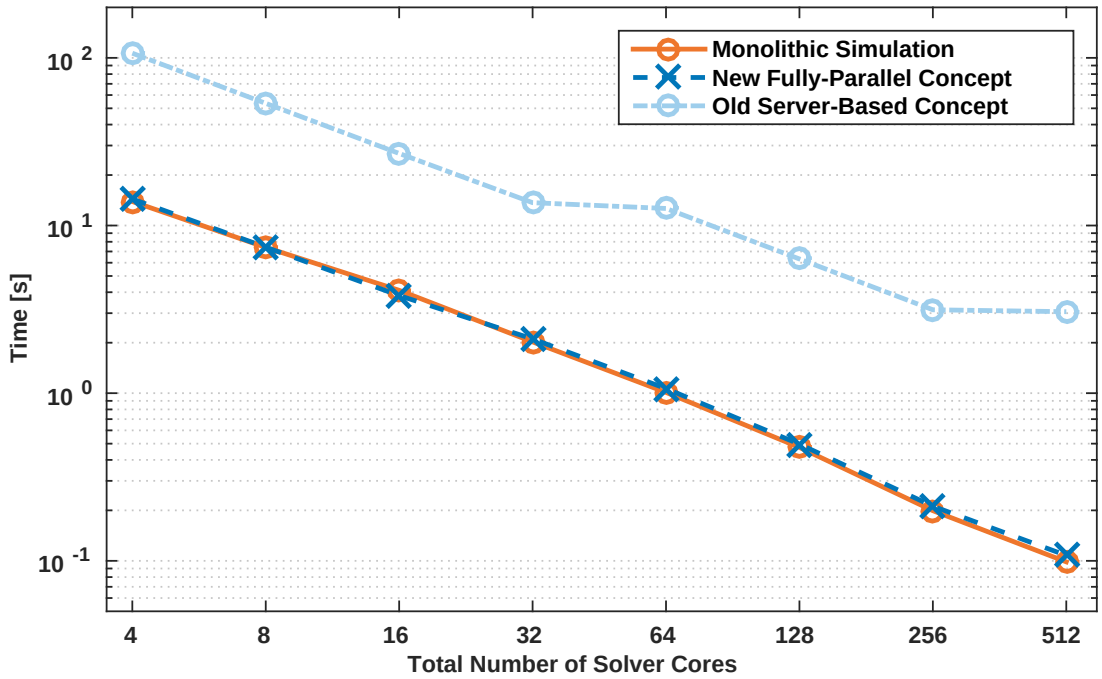


- No central entity
- Interface computations in parallel
- $\Rightarrow$ No scaling issues

Simple Scaling Test

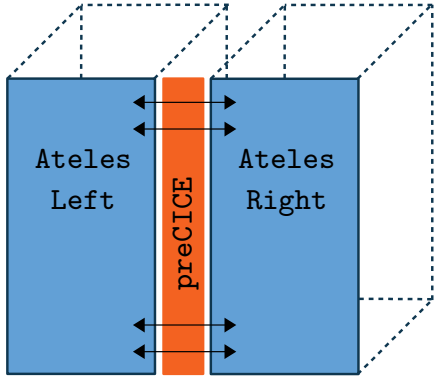
- Travelling density pulse (Euler equations) through artificial coupling interface
- DG solver Ateles (U Siegen), $\mathbf{7.1 \cdot 10^6}$ dofs
- Nearest neighbor mapping and communication

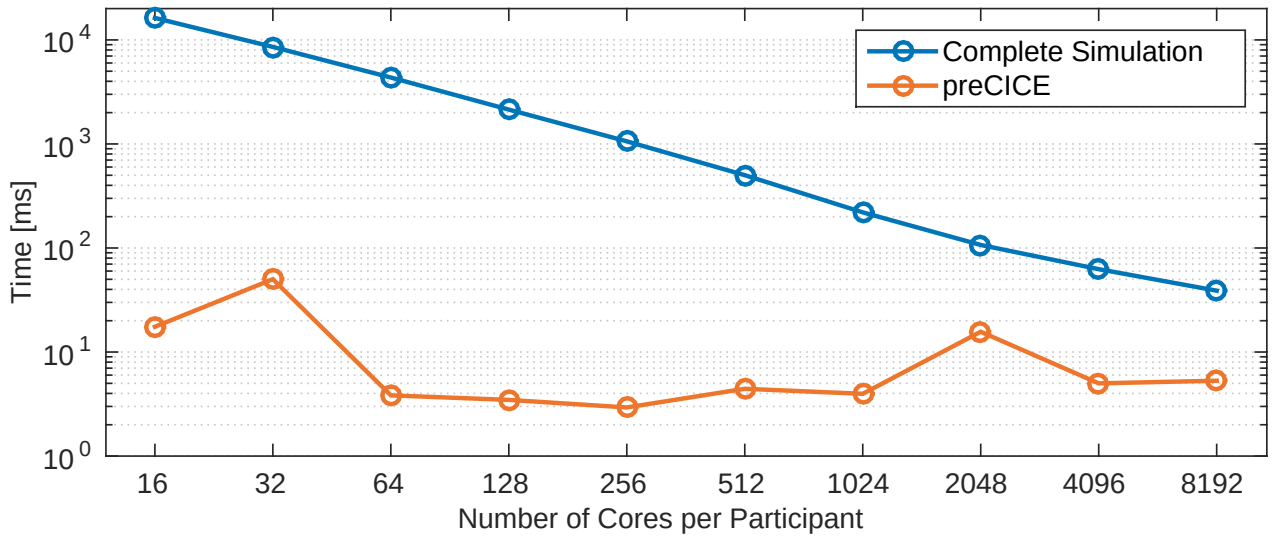




Simple Scaling Test (cont.)

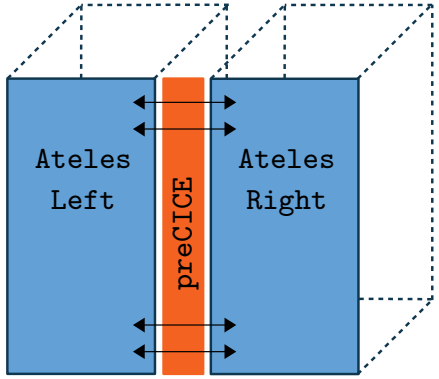
- Travelling density pulse (Euler equations) through artificial coupling interface
- DG solver Ateles (U Siegen), $\mathbf{5.7 \cdot 10^7}$ dofs
- Nearest neighbor mapping and communication

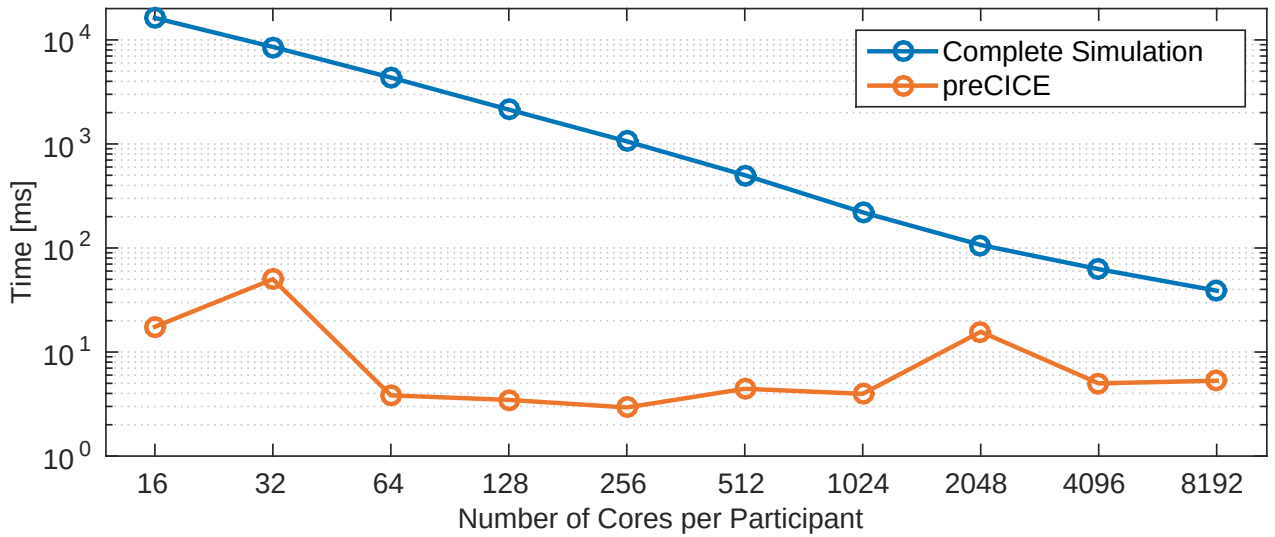




Simple Scaling Test (cont.)

- Travelling density pulse (Euler equations) through artificial coupling interface
- DG solver Ateles (U Siegen), $\mathbf{5.7 \cdot 10^7}$ dofs
- Nearest neighbor mapping and communication

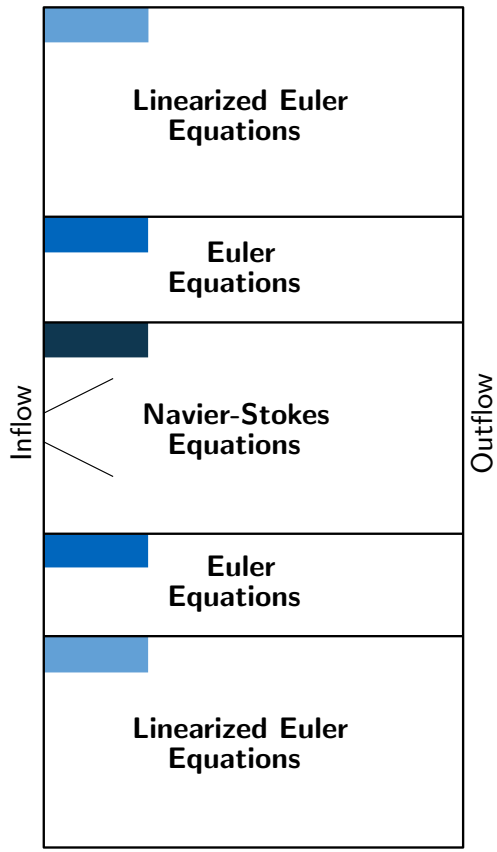


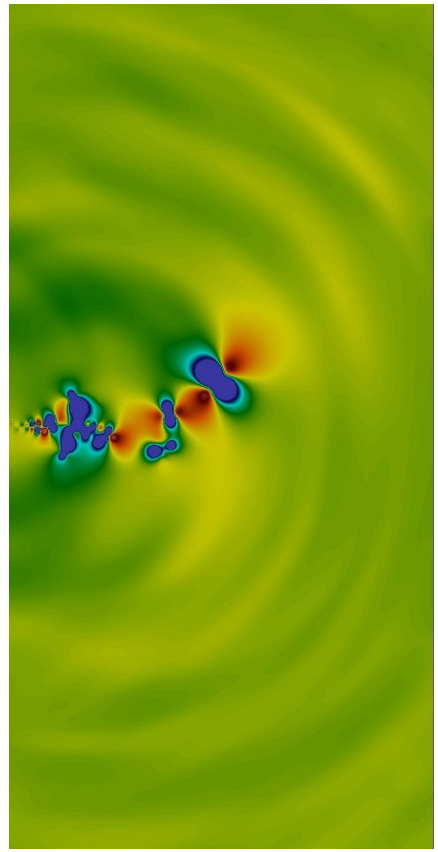


$\Rightarrow$ **Coupling does not deteriorate scalability**

Fluid-Acoustic Interaction

- Sub-sonic jet
- Explicit parallel coupling between three solver, all Ateles
- Linearized Euler 28x cheaper than Navier-Stokes

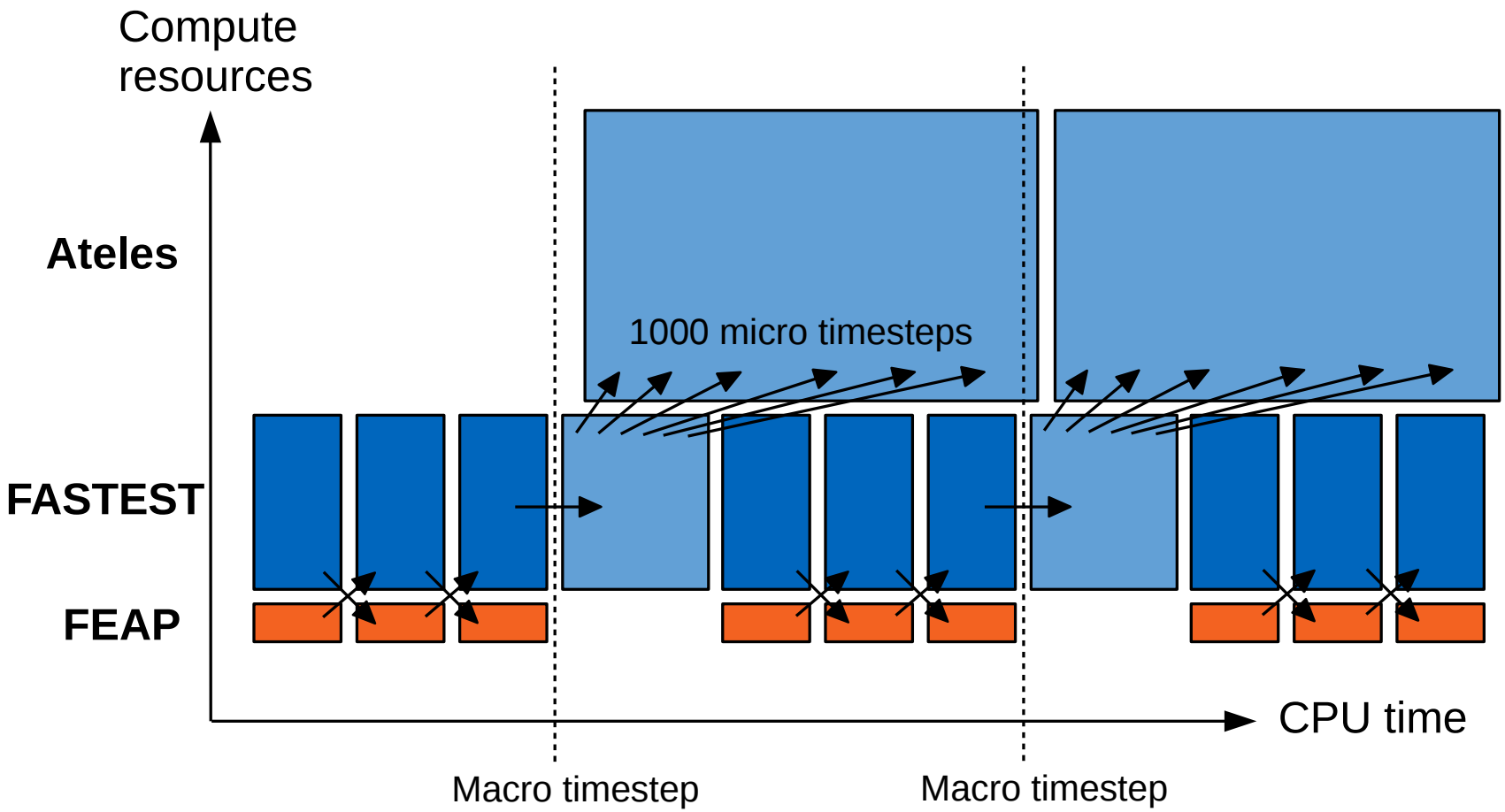




- Joint work with Verena Krupp et al.

Load-Balancing

- Difficult
- We need manual buffering for uni-directional coupling



- Joint work with Thorsten Reimann et al. and Neda Ebrahimi Pour et al.

Summary

- For challenging multiphysics, multiscale problems, we need an **easy-to-use** and **scalable** coupling approach
- **easy-to-use** = minimally-invasive integration in legacy codes
  - library approach (no framework approach)
  - high-level API
  - ready-to-use adapters for community codes
- **scalable** = don't deteriorate scalability of coupled solvers
  - peer-to-peer approach (no server approach)
  - tricks for load-balancing

www.precice.org

github.com/precice

@preCICE_org

www.tue.nl

b.w.uekermann@tue.nl

Figure: Doughnuts contributed by Gerasimos Chourdakis, TUM.