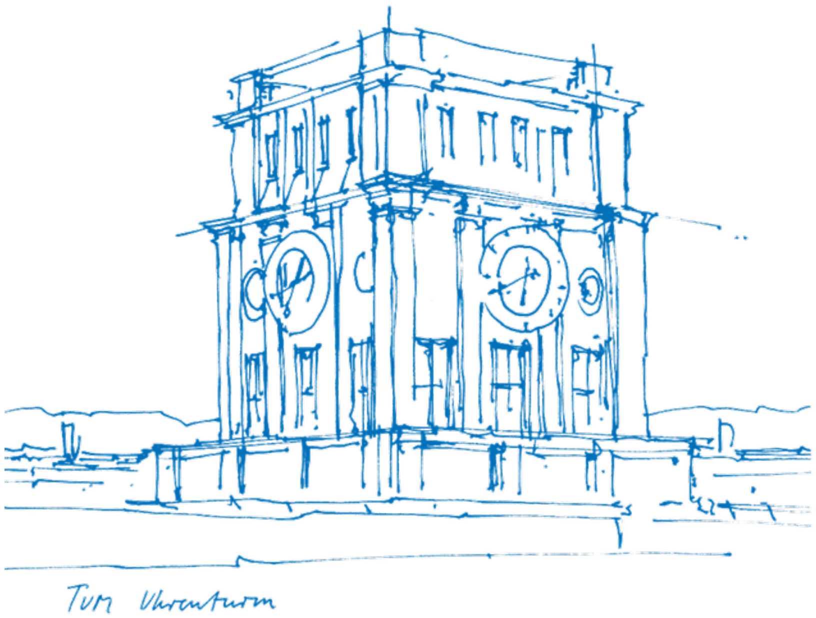
Multi-physics simulations with OpenFOAM through preCICE

Gerasimos Chourdakis, Hans-Joachim Bungartz, Lucia Cheung Yau,
Benjamin Uekermann

Technical University of Munich
Department of Informatics
Chair of Scientific Computing in Computer Science

89th GAMM Annual Meeting
Technical University of Munich
March 22, 2018



Agenda

Part I:

preCICE

Agenda

Part I:

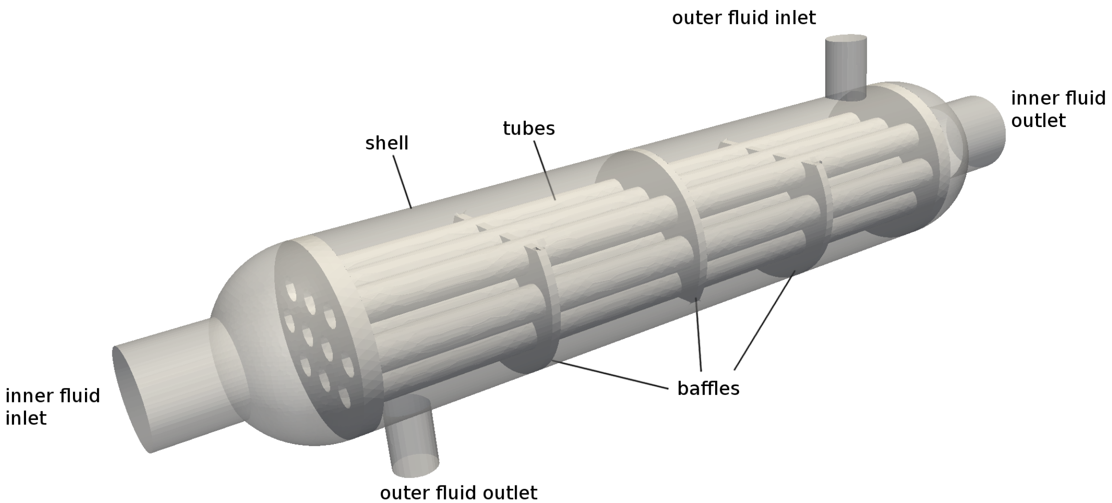
preCICE

Part II:

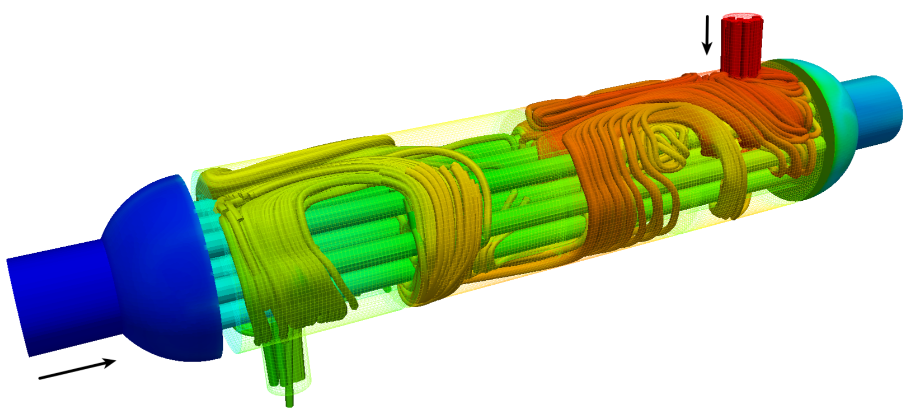
preCICE + OpenFOAM The OpenFOAM Foundation

OpenFOAM

How to simulate this heat exchanger?

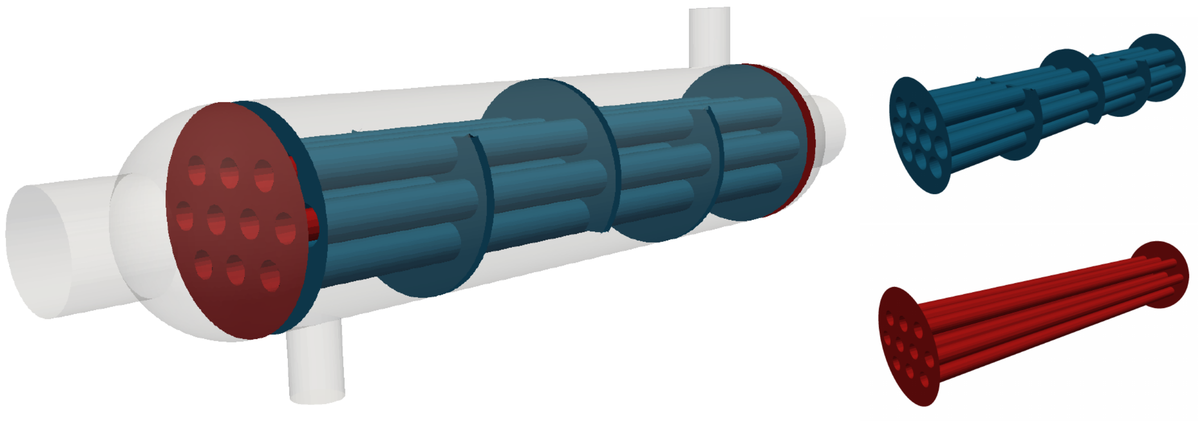


Geometry of a shell-and-tube heat exchanger (Image by L. Cheung Yau, 2016)

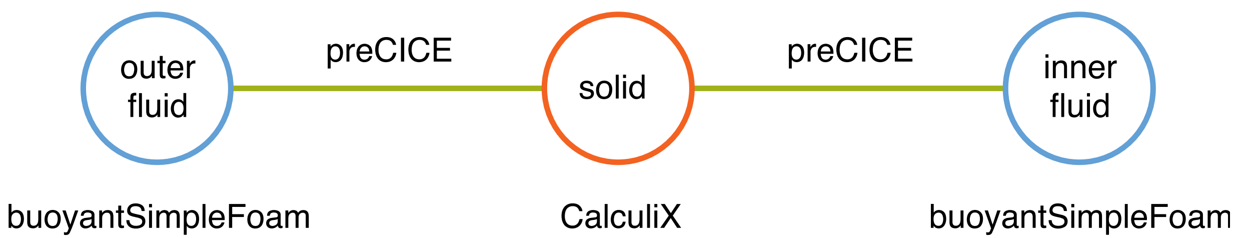


Surface plot and streamlines of the two fluids colored by temperature. Solid not shown.

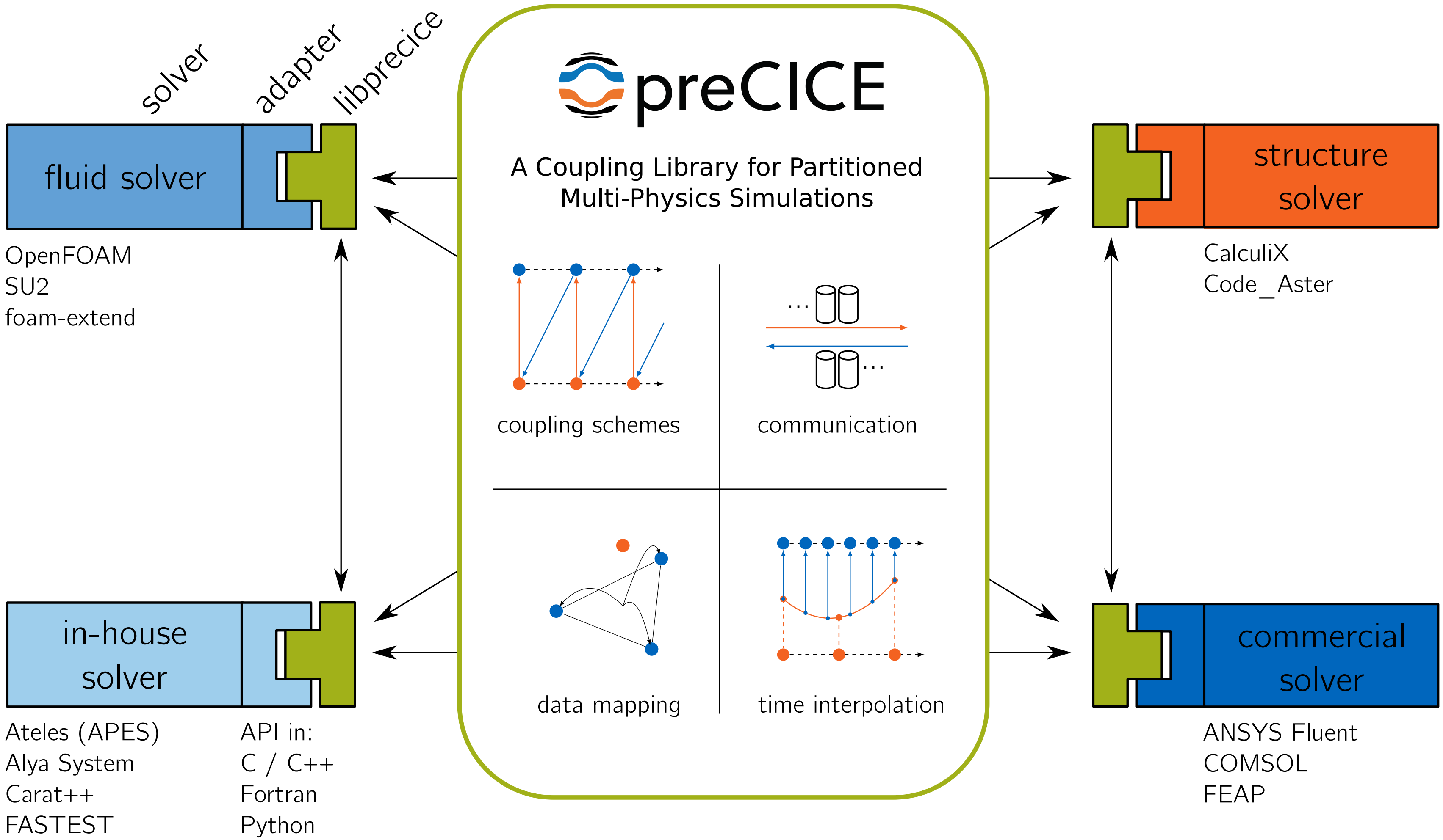
Multiple regions → multiple solvers → preCICE



Coupling interfaces (Image by L. Cheung Yau, 2016)



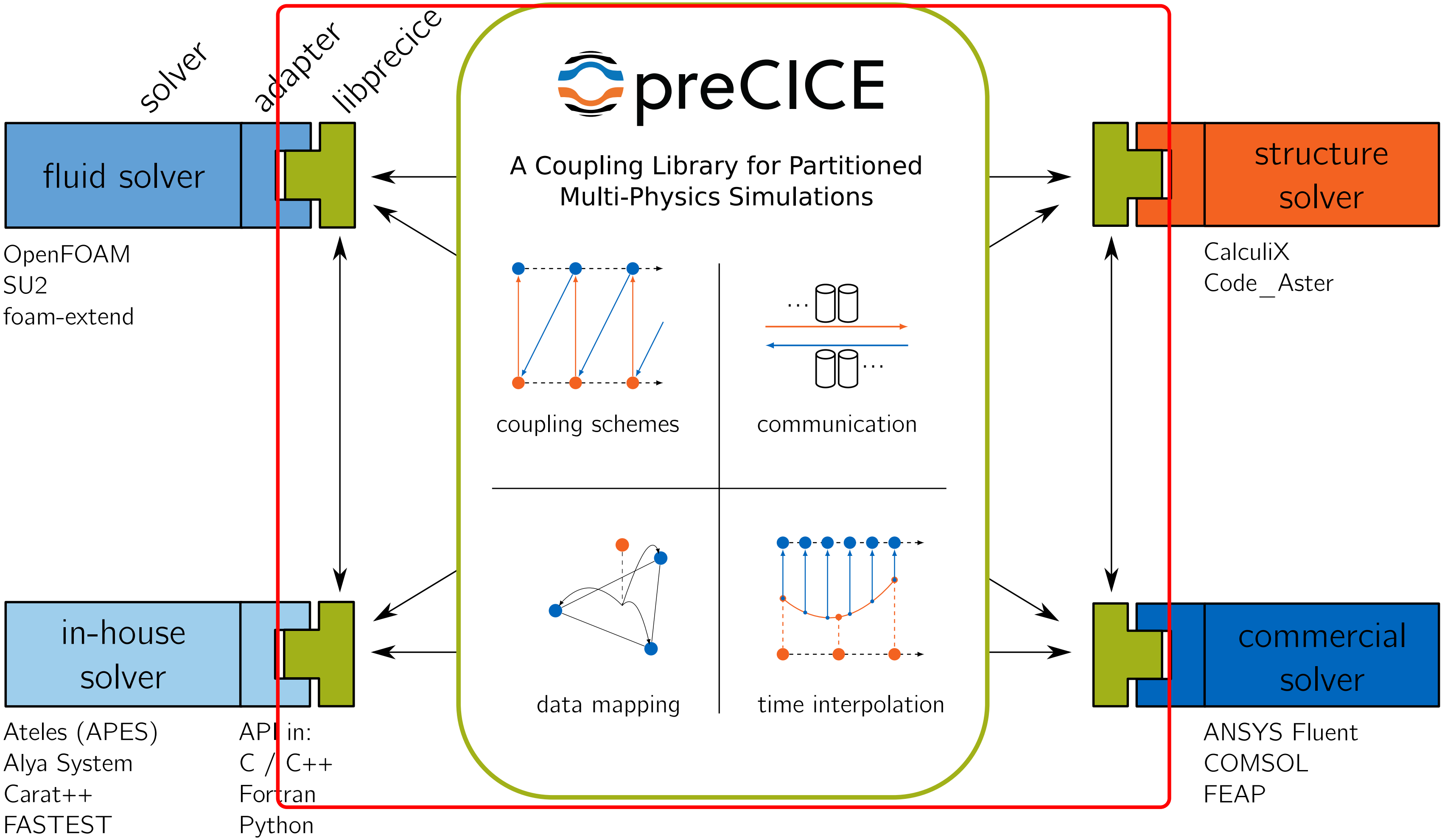
preCICE



www.precice.org

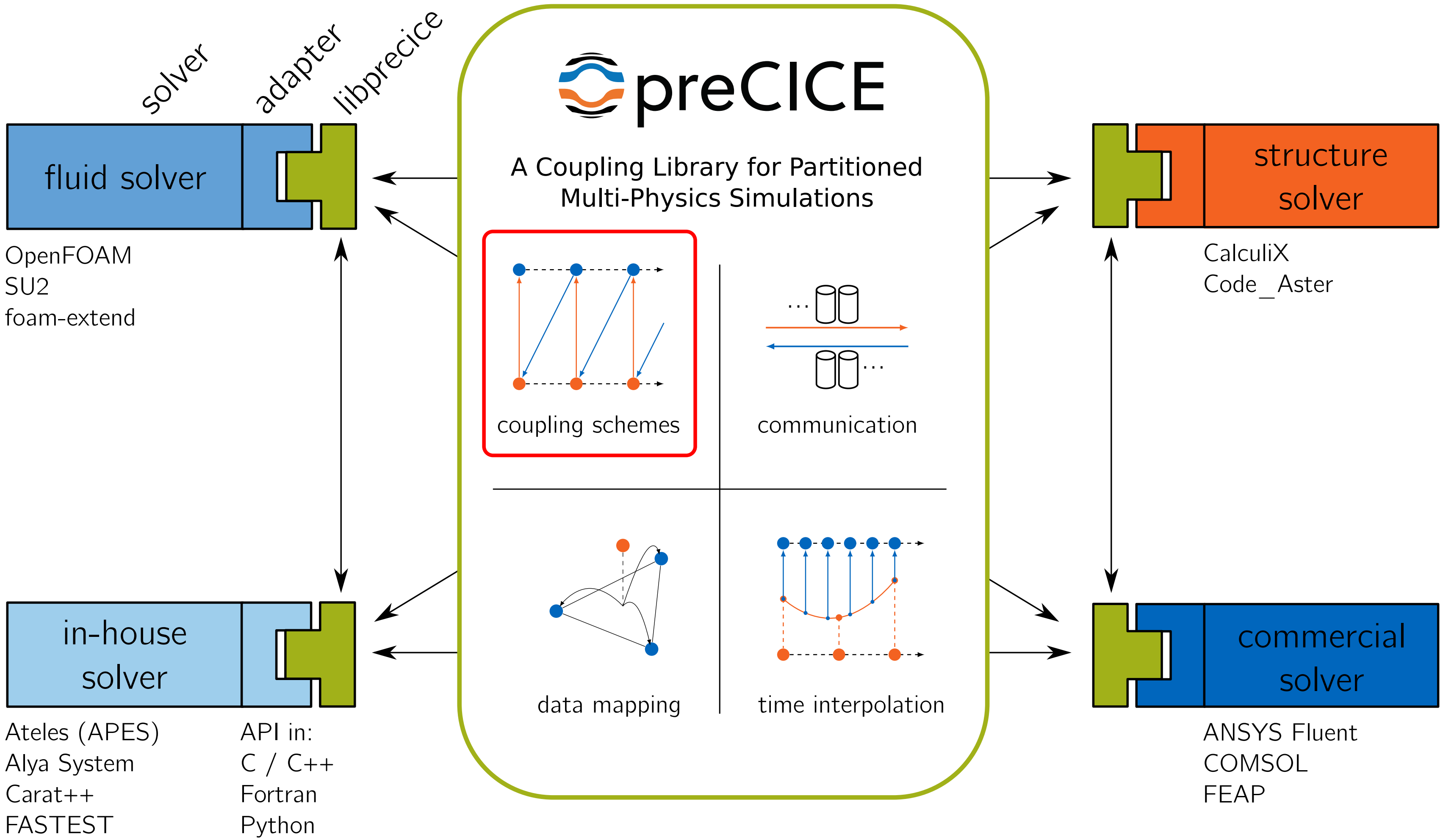
github.com/precice

preVIOUSLY on GAMM2018



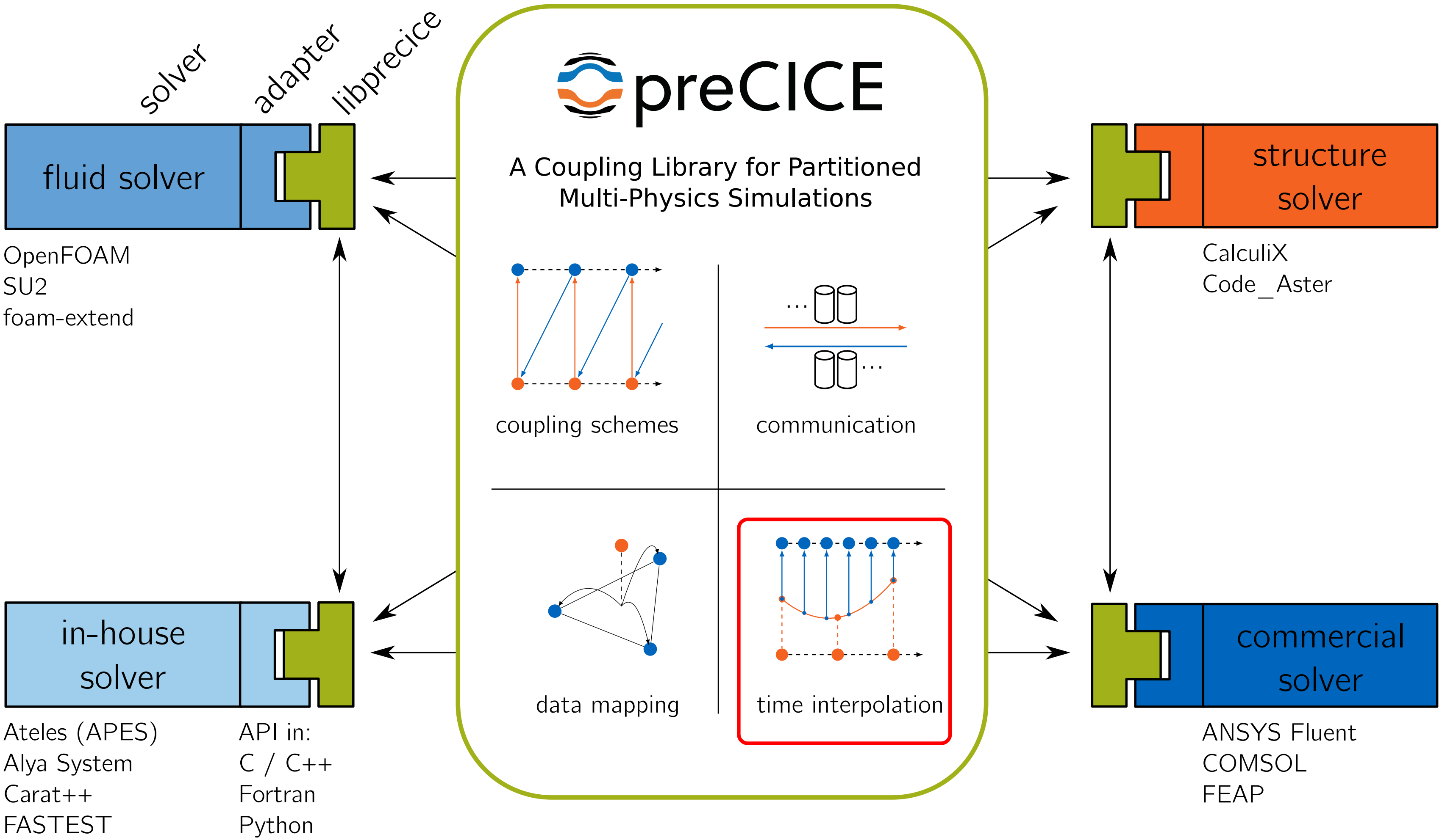
preCICE Coupling Library for Multi-Physics Simulation
Amin Totounferoush, University of Stuttgart
S07.03 Coupled Problems (Tuesday afternoon)

preVIOUSLY on GAMM2018



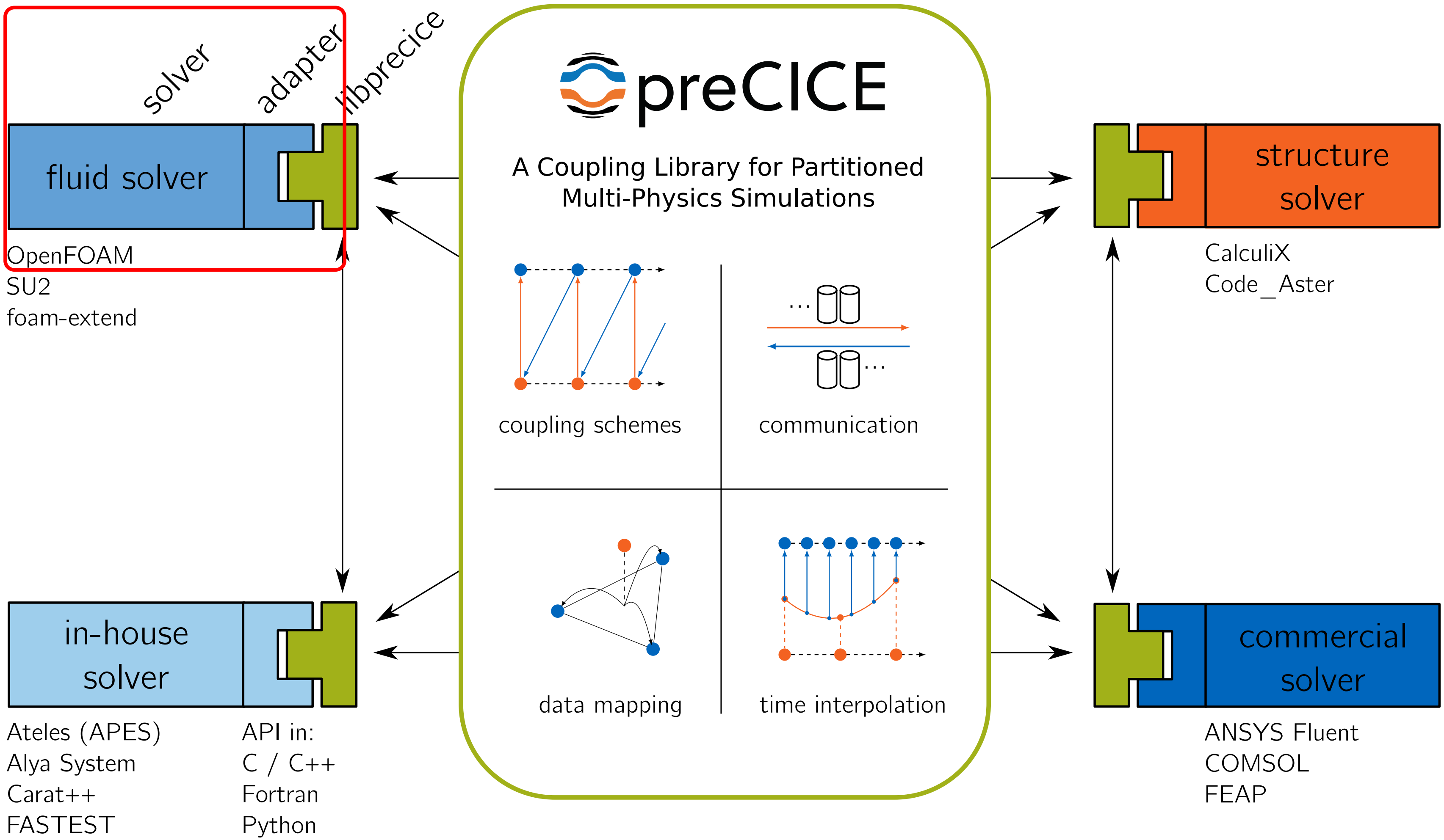
Quasi-Newton – A Universal Approach for Coupled Problems and Optimization
Miriam Mehl, University of Stuttgart
S07.01 Coupled Problems (Tuesday morning)

preVIOUSLY on GAMM2018



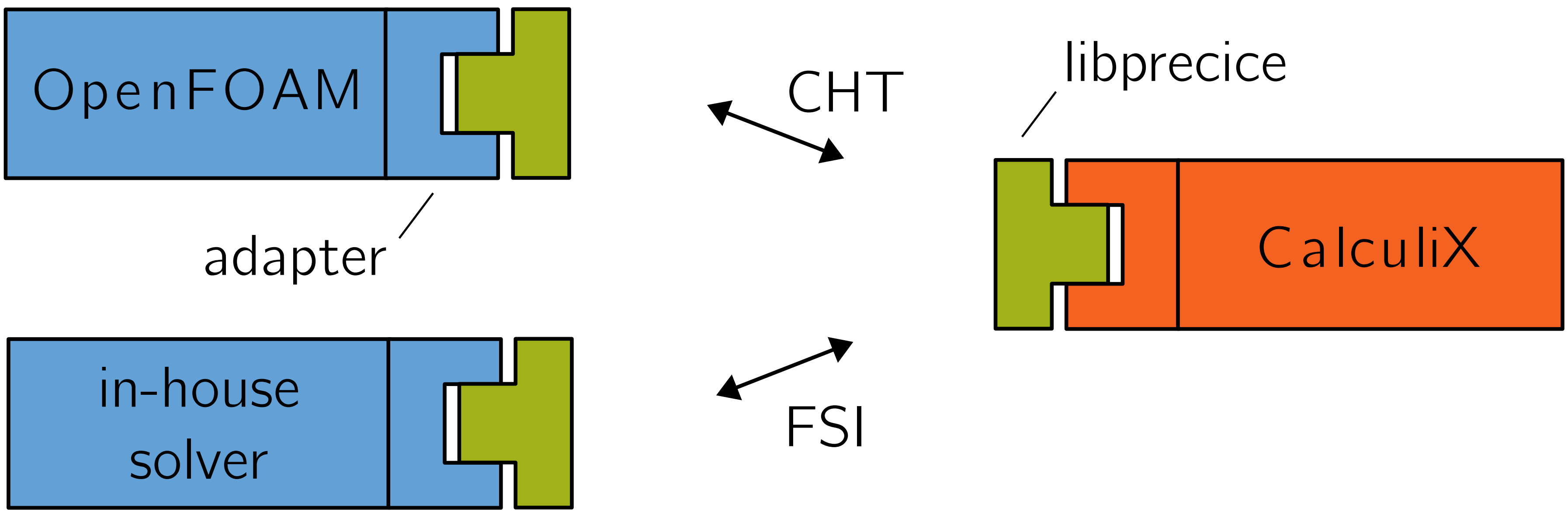
Improving Time Stepping in Partitioned Multi-Physics
Benjamin Rüth, Technical University of Munich
S22.01 Scientific Computing (Tuesday afternoon)

preVIOUSLY on GAMM2018



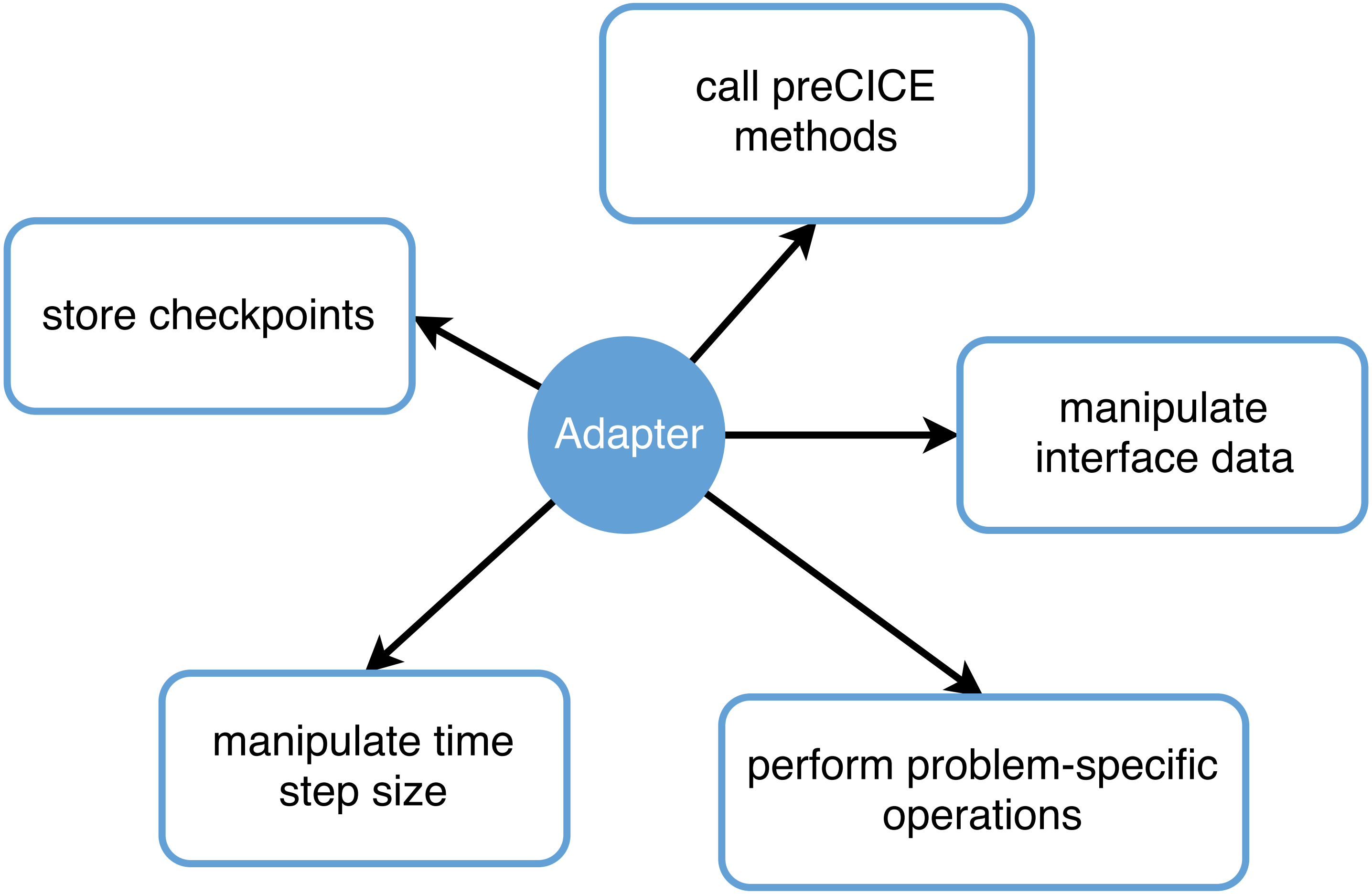
Multi-physics simulations with OpenFOAM through preCICE
Gerasimos Chourdakis, Technical University of Munich
S22.04 Scientific Computing (we are here)

But why preCICE?



- Pure **library** approach $\rightarrow$ flexibility
- Fully parallel, **peer-to-peer** concept $\rightarrow$ scalable and efficient communication
- Sophisticated and robust **quasi-Newton** coupling algorithms
- **Multi-coupling**

The roles of an adapter



Part IIa: previous approach

preCICE + OpenFOAM (The OpenFOAM Foundation) / OpenFOAM

OpenFOAM

Open-source Field Operation And Manipulation

- **Collection** of tools for continuum mechanics (mainly CFD)
- **Framework** for in-house solvers
- **Several variants:** OpenFOAM (openfoam.org), OpenFOAM+ (openfoam.com), foam-extend
- Free (GNU GPL), C++
- Widespread in industry and academia

What if you could **couple your in-house solvers** with OpenFOAM?

Duplicated development effort

OpenFOAM (and family) adapters for preCICE

David Blom,
2015-17 (TU Delft)
FSI, foam-extend

Duplicated development effort

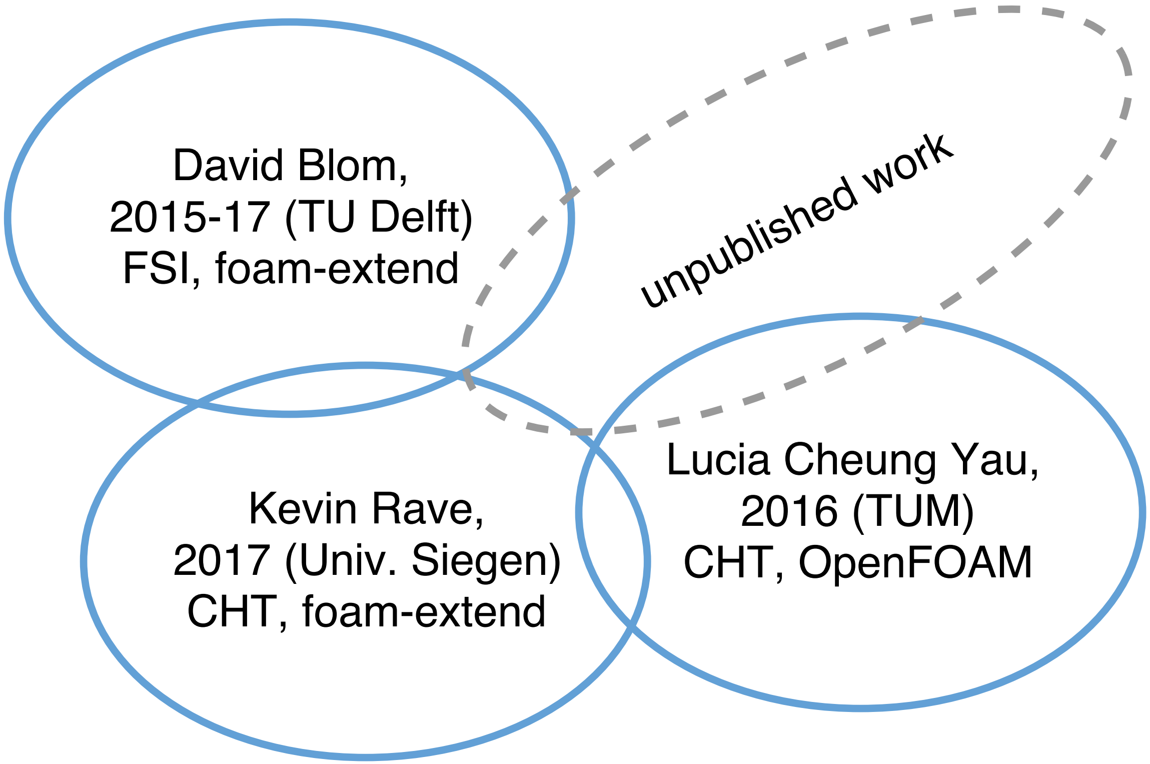
OpenFOAM (and family) adapters for preCICE

David Blom,
2015-17 (TU Delft)
FSI, foam-extend

Lucia Cheung Yau,
2016 (TUM)
CHT, OpenFOAM

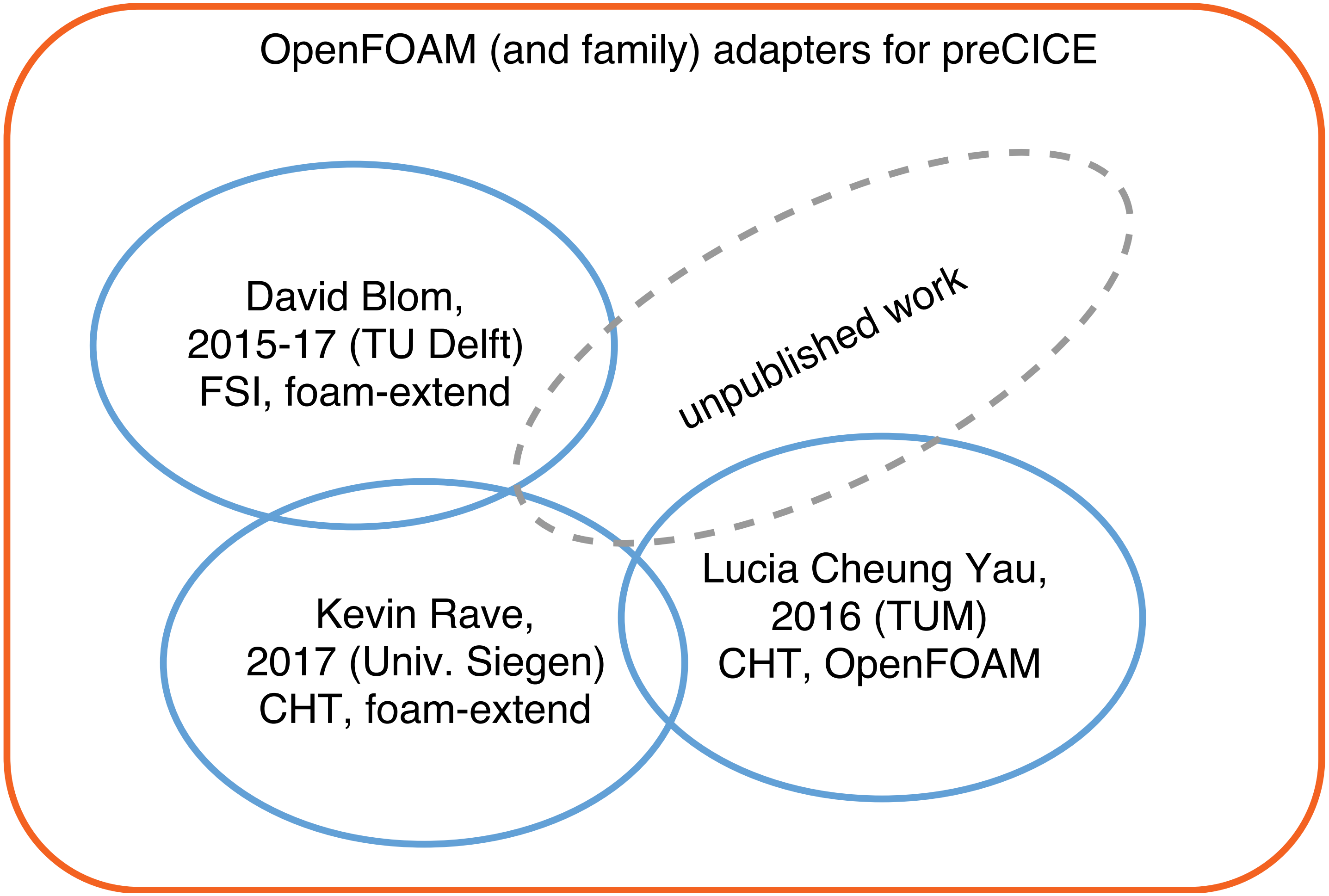
Duplicated development effort

OpenFOAM (and family) adapters for preCICE



All these adapters are **bound to specific solvers**!

Duplicated development effort



All these adapters are **bound to specific solvers**!

→ We need an official, general adapter!

Example of an adapted solver (previous)

```
/* Adapter: Initialize coupling
      calls precice->initialize() */
adapter.initialize();

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

  /* Adapter: Adjust solver time */
  adapter.adjustSolverTimeStep();

  /* Adapter: Write checkpoint */
  if(adapter.isWriteCheckptRequired())
    adapter.writeCheckpoint();

  runTime++;

  /* Adapter: Receive coupling data */
  adapter.readCouplingData();
  /* solve the equations */
  #include "rhoEqn.H"
  while (pimple.loop())
  {
      ...
  }

  /* Adapter: Write in buffers */
  adapter.writeCouplingData();

  /* Adapter: advance the coupling
        calls precice->advnace()  */
  adapter.advance();

  /* Adapter: Read checkpoint */
  if(adapter.isReadCheckptRequired())
      adapter.readCheckpoint();

  if(adapter.isCouplTimeStepComplete())
    runTime.write();
}
```

Before: Working and validated prototypes

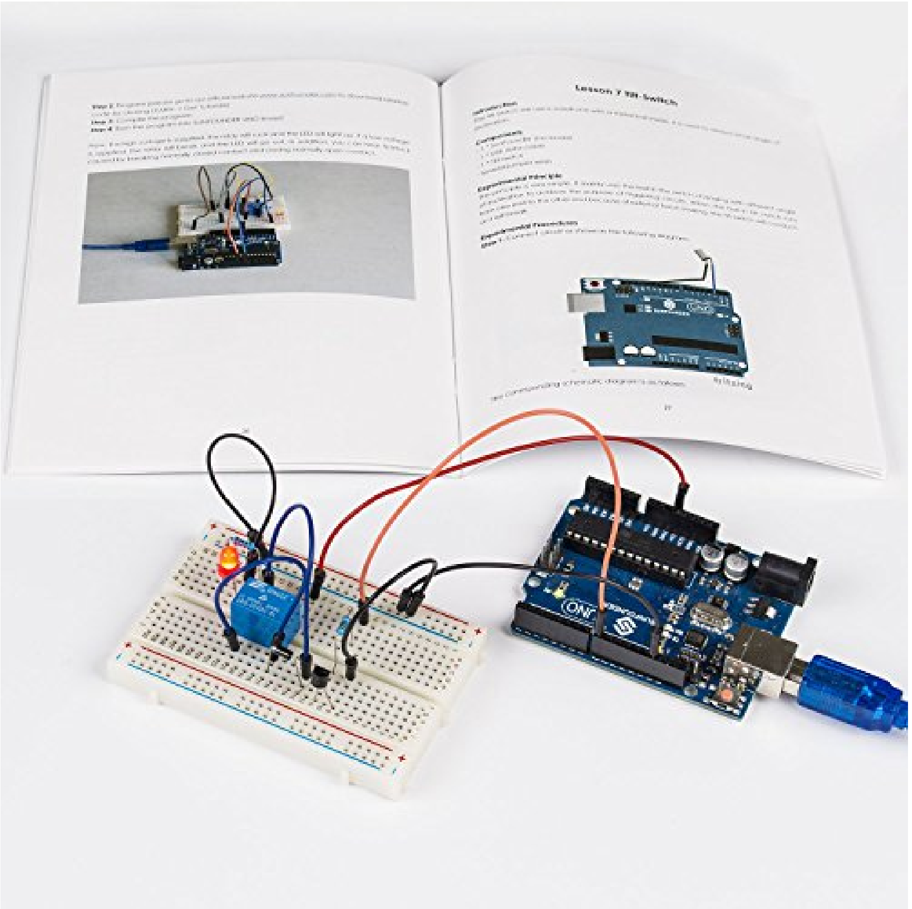


Image from desertcart.ae.

Before: Working and validated prototypes

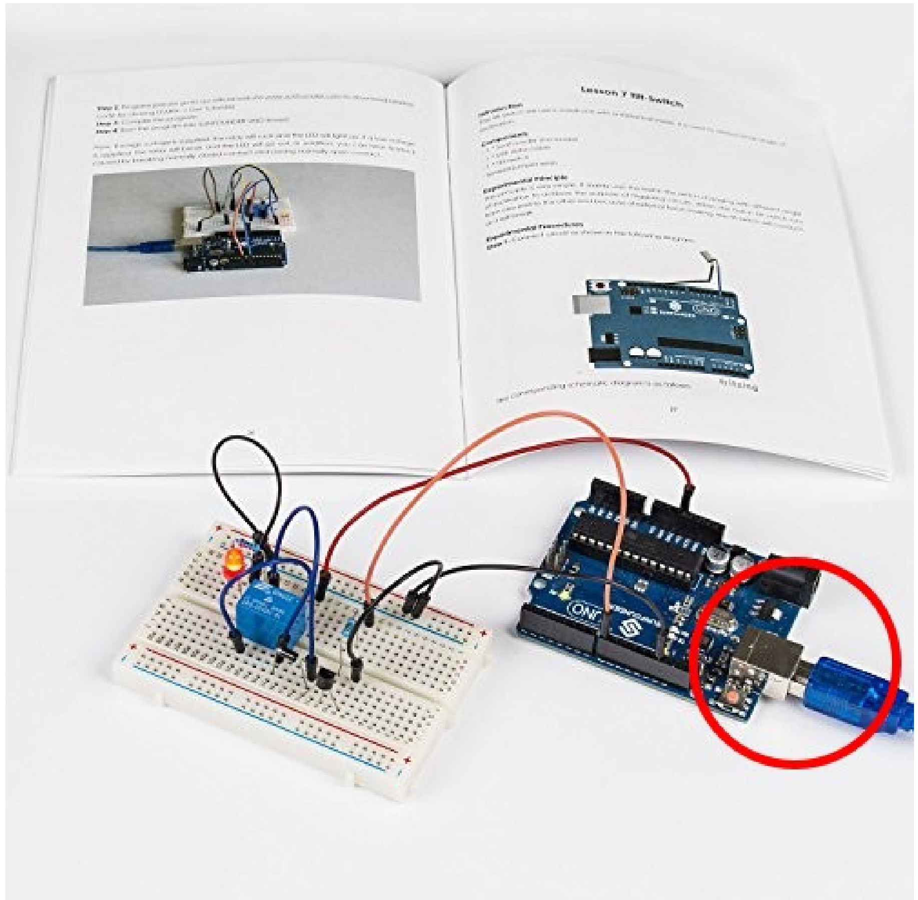


Image from desertcart.ae.

Now: A user-friendly, plug-and-play adapter

KOPPLAD

2x OpenFOAM solvers
1x OpenFOAM adapter
1x preCICE

The human-like figure is a property of ikea.com.

Part IIb: a new, official adapter

preCICE + OpenFOAM (The OpenFOAM Foundation) / OpenFOAM

Making this a function object

OpenFOAM function objects allow to call external code from specific points in every solver.

Several **challenges**:

- No changes in the source allowed
  - Cannot use variables directly
  - Ask the objects' registry
- One adapter for all the solvers and problem types
  - Some parameters are not available
- Only one call to `execute()` at the end
  - We may need to reload a checkpoint at the last timestep...
  - Set the `endTime` to `GREAT` and exit when ready.
- Collaboration with other function objects
  - At the end, call any other `end()` methods explicitly.
- Error handling
  - `read()` degrades errors to warnings
  - Catch them and throw them in `execute`
- One adapter for all the OpenFOAM flavors and versions?
  - E.g. `boundaryField()` and `boundaryFieldRef()`
  - E.g. missing `adjustTimeStep()`
  - How to distribute? Branches/Tags? Preprocessor `ifdef`?
- ...

Making this a function object

OpenFOAM function objects allow to call external code from specific points in every solver.

Several **challenges**:

- No changes in the source allowed
  - Cannot use variables directly
  - Ask the objects' registry
- One adapter for all the solvers and problem types
  - Some parameters are not available
- Only one call to `execute()` at the end
  - We may need to reload a checkpoint at the last timestep...
  - Set the `endTime` to `GREAT` and exit when ready.
- Collaboration with other function objects
  - At the end, call any other `end()` methods explicitly.
- Error handling
  - `read()` degrades errors to warnings
  - Catch them and throw them in `execute`
- One adapter for all the OpenFOAM flavors and versions?
  - E.g. `boundaryField()` and `boundaryFieldRef()`
  - E.g. missing `adjustTimeStep()`
  - How to distribute? Branches/Tags? Preprocessor `ifdef`?
- ...

Several **advantages**:

- No source code changes
- Load at runtime
- (mostly) Solver agnostic

Making this a function object

OpenFOAM function objects allow to call external code from specific points in every solver.

Several **challenges**:

- No changes in the source allowed
  - Cannot use variables directly
  - Ask the objects' registry
- One adapter for all the solvers and problem types
  - Some parameters are not available
- Only one call to `execute()` at the end
  - We may need to reload a checkpoint at the last timestep...
  - Set the `endTime` to `GREAT` and exit when ready.
- Collaboration with other function objects
  - At the end, call any other `end()` methods explicitly.
- Error handling
  - `read()` degrades errors to warnings
  - Catch them and throw them in `execute`
- One adapter for all the OpenFOAM flavors and versions?
  - E.g. `boundaryField()` and `boundaryFieldRef()`
  - E.g. missing `adjustTimeStep()`
  - How to distribute? Branches/Tags? Preprocessor `ifdef`?
- ...

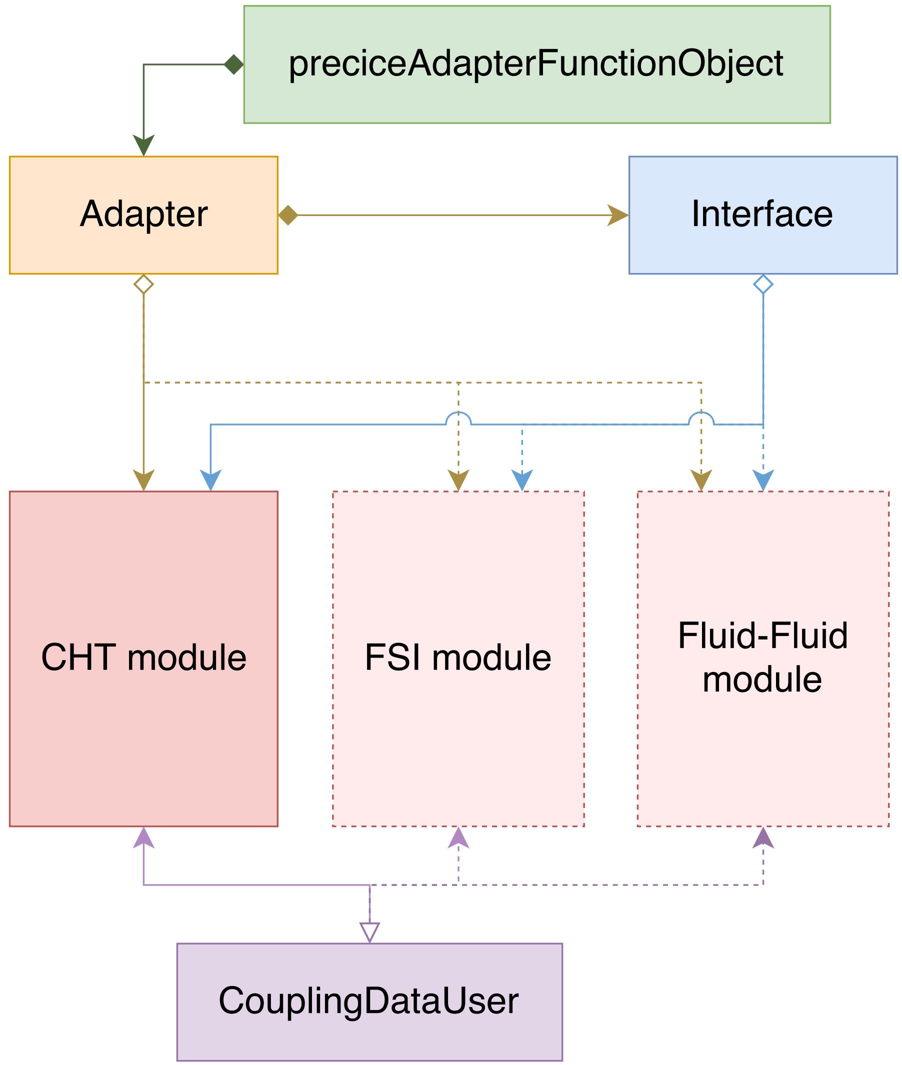
Several **advantages**:

- No source code changes
- Load at runtime
- (mostly) Solver agnostic

However:

- Still ready-to-run only for CHT
- but...

An extensible adapter



How can I use it?

OpenFOAM configuration

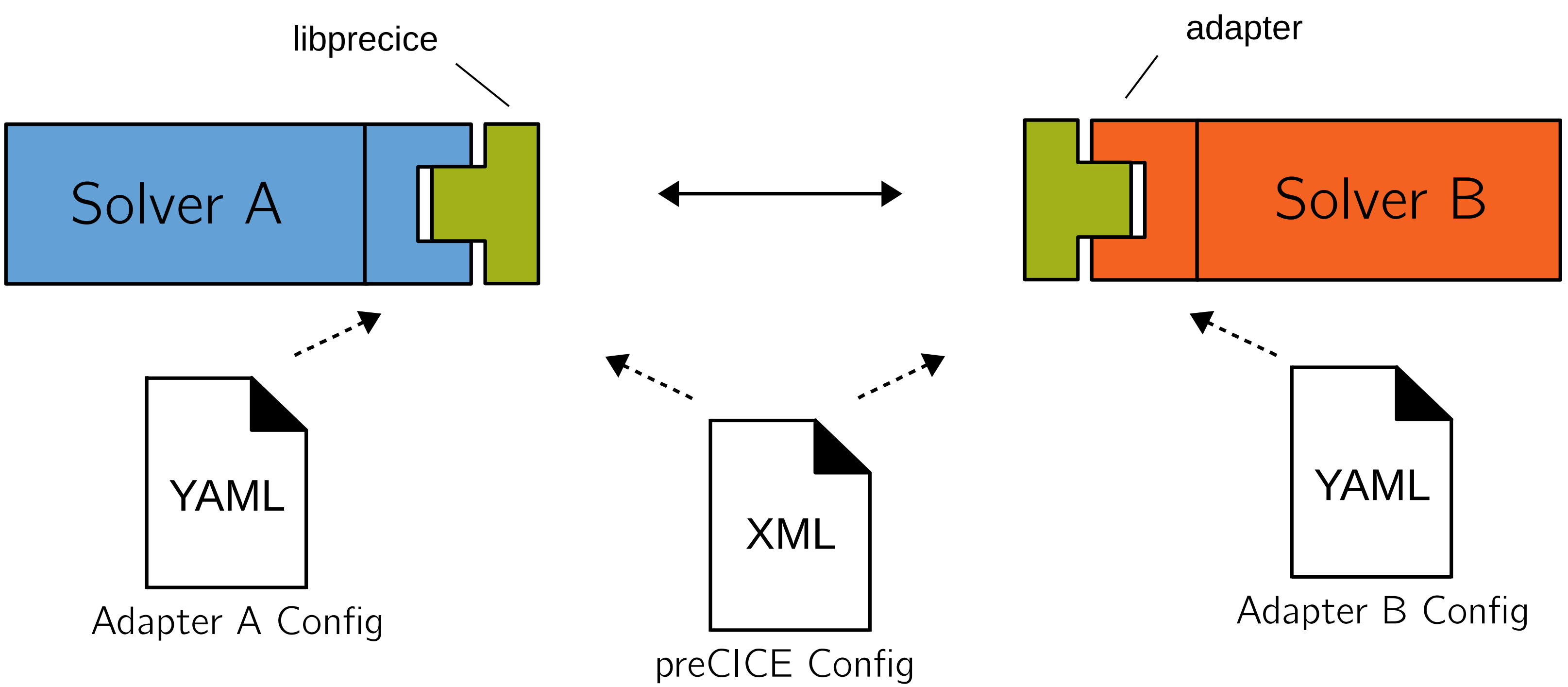
Enable the adapter:

```
// system/controlDict
functions
{
    preCICE_Adapter
    {
        type preciceAdapterFunctionObject;
        libs ("libpreciceAdapterFunctionObject.so");
    }
}
```

Set the appropriate boundary condition types:

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

preCICE & adapter configuration

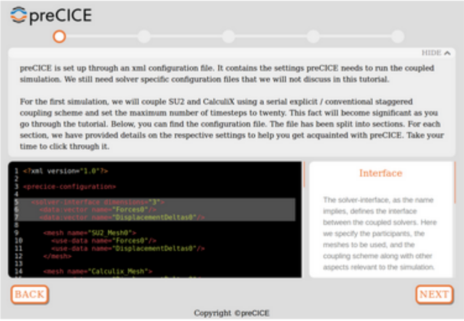
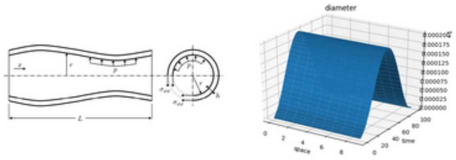
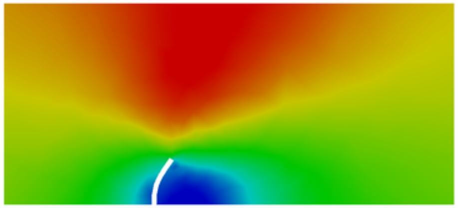


To run the simulation, just execute the solvers as usual.

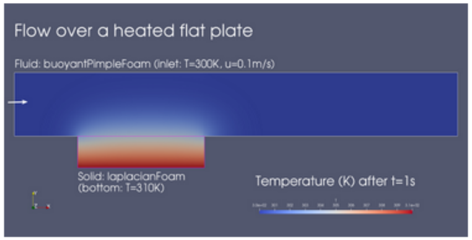
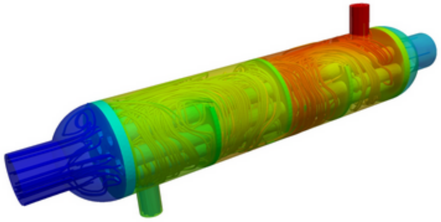
Tutorials

On `www.precice.org/resources`:

Fluid-Structure Interaction

<table>
<tr><td>Web-based tutorial</td><td>1D FSI Example</td><td>FSI with SU2 and CalculiX</td></tr>
<tr><td>Flow in a channel with an elastic flap</td><td>Flow through a deformable tube</td><td>Flow in a channel with an elastic flap</td></tr>
<tr><td>
</td><td>
</td><td></td></tr>
</table>

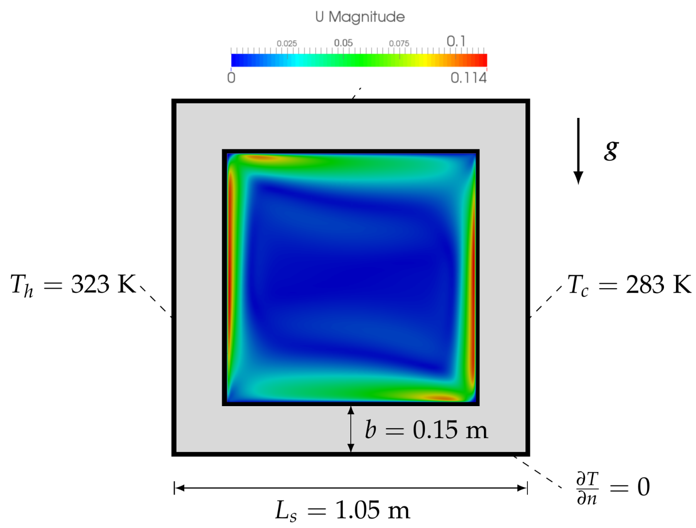
Conjugate Heat Transfer

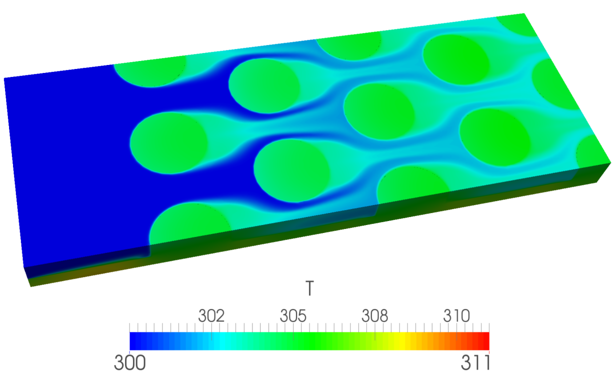
<table>
<tr><td>CHT with OpenFOAM</td><td>CHT with OpenFOAM and CalculiX</td></tr>
<tr><td>Flow above a heated plate</td><td>Shell-and-tube heat exchanger</td></tr>
<tr><td>
</td><td></td></tr>
</table>

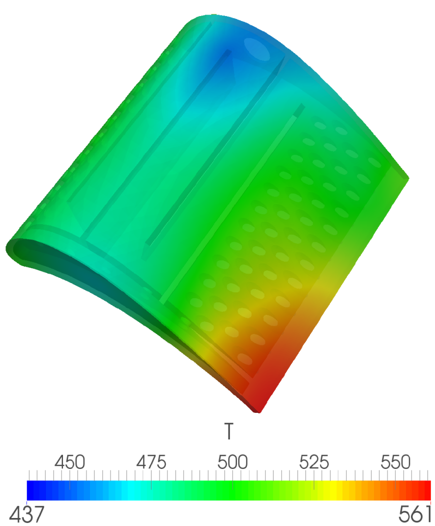
More examples with OpenFOAM

- Natural convection cavity
  - OpenFOAM + CalculiX
  - Transient
  - Robin-Robin serial-implicit coupling, IQN-ILS
- Pin-Fin cooling system
  - OpenFOAM + CalculiX
  - Steady-state
  - Robin-Robin parallel-implicit coupling, IQN-ILS
- Cooling of a turbine blade
  - OpenFOAM + CalculiX (or Code_Aster)
  - Steady-state
  - Robin-Robin parallel-explicit coupling

(simulations by L. Cheung Yau, 2016)

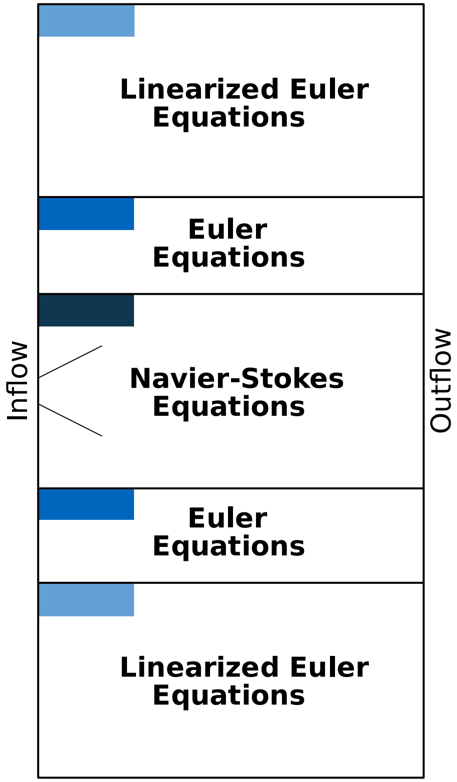


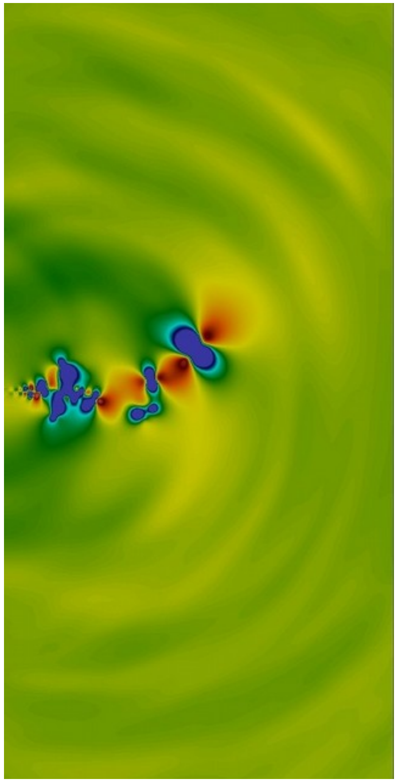




Further possibilities: Multi-fluid coupling

- Besides FSI, many other possible applications of preCICE
- Simulation of a subsonic jet
- Explicit, parallel coupling between three fluid solvers
- Joint work with the University of Siegen





Does it work with “chocolate” OpenFOAM?

Known to work with:

The OpenFOAM Foundation: 4.0 – dev

ESI - OpenCFD: v1706

Currently does not work with:

The OpenFOAM Foundation: $\leq$ 3.0

ESI - OpenCFD: $\leq$ v1606+

foam-extend: any version

Free, documented, and easy to use

preCICE:

- allows to **reuse existing software**,
- is **free software**,
- helps researchers **all over the world**.

Free, documented, and easy to use

preCICE:

- allows to **reuse existing software**,
- is **free software**,
- helps researchers **all over the world**.

we aim to:

- spread it **outside of our groups**
  (see how e.g. OpenFOAM has an impact),
- keep it **well documented** and updated,
- improve its **usability**.

Free, documented, and easy to use

preCICE:

- allows to **reuse existing software**,
- is **free software**,
- helps researchers **all over the world**.

we aim to:

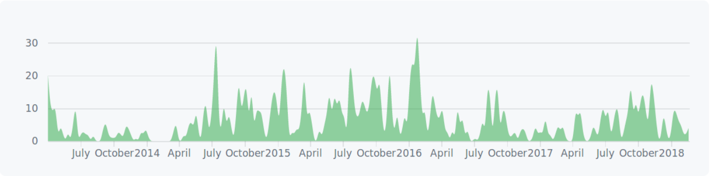
- spread it **outside of our groups**
  (see how e.g. OpenFOAM has an impact),
- keep it **well documented** and updated,
- improve its **usability**.

Remember: Free, quality software means much more than releasing the source code!

Free, documented, and easy to use

preCICE:

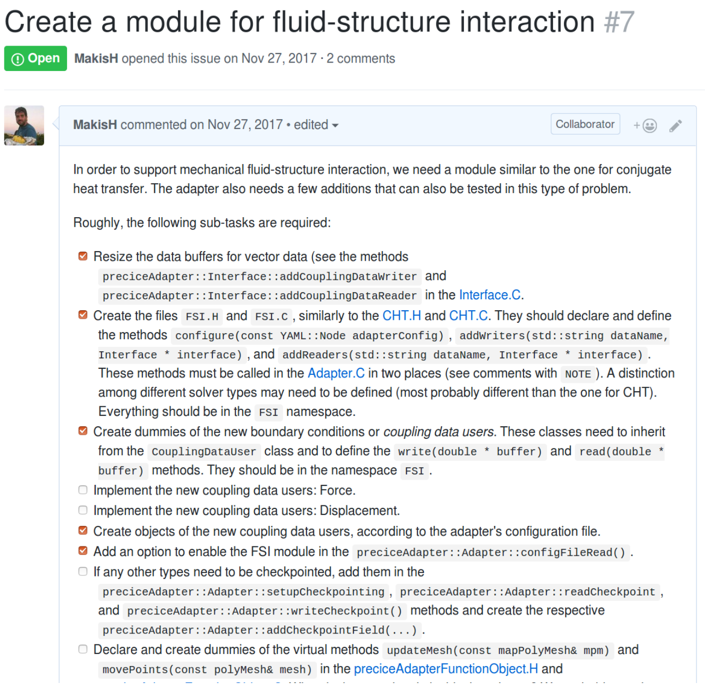
- allows to **reuse existing software**,
- is **free software**,
- helps researchers **all over the world**.



Activity (commits) on preCICE, past 5 years

we aim to:

- spread it **outside of our groups**
  (see how e.g. OpenFOAM has an impact),
- keep it **well documented** and updated,
- improve its **usability**.



Create a module for fluid-structure interaction #7

Open MakisH opened this issue on Nov 27, 2017 · 2 comments

MakisH commented on Nov 27, 2017 • edited
Collaborator

In order to support mechanical fluid-structure interaction, we need a module similar to the one for conjugate heat transfer. The adapter also needs a few additions that can also be tested in this type of problem.

Roughly, the following sub-tasks are required:

- [x] Resize the data buffers for vector data (see the methods `preciceAdapter::Interface::addCouplingDataWriter` and `preciceAdapter::Interface::addCouplingDataReader` in the Interface.C.
- [x] Create the files `FSI.H` and `FSI.C`, similarly to the CHT.H and CHT.C. They should declare and define the methods `configure(const YAML::Node adapterConfig)`, `addWriters(std::string dataName, Interface * interface)`, and `addReaders(std::string dataName, Interface * interface)`. These methods must be called in the Adapter.C in two places (see comments with `NOTE`). A distinction among different solver types may need to be defined (most probably different than the one for CHT). Everything should be in the `FSI` namespace.
- [x] Create dummies of the new boundary conditions or *coupling data users*. These classes need to inherit from the `CouplingDataUser` class and to define the `write(double * buffer)` and `read(double * buffer)` methods. They should be in the namespace `FSI`.
- [ ] Implement the new coupling data users: Force.
- [ ] Implement the new coupling data users: Displacement.
- [x] Create objects of the new coupling data users, according to the adapter's configuration file.
- [x] Add an option to enable the FSI module in the `preciceAdapter::Adapter::configFileRead()`.
- [ ] If any other types need to be checkpointed, add them in the `preciceAdapter::Adapter::setupCheckpointing`, `preciceAdapter::Adapter::readCheckpoint`, and `preciceAdapter::Adapter::writeCheckpoint()` methods and create the respective `preciceAdapter::Adapter::addCheckpointField(...)`.
- [ ] Declare and create dummies of the virtual methods `updateMesh(const mapPolyMesh& mpm)` and `movePoints(const polyMesh& mesh)` in the preciceAdapterFunctionObject.H and

Contribute on GitHub!

Remember: Free, quality software means much more than releasing the source code!

Coming soon in the OpenFOAM adapter:

- Fluid-Structure Interaction Module
- Code improvements and tests
- Support for older OpenFOAM versions

References

preCICE **preCICE – A Fully Parallel Library for Multi-Physics Surface Coupling**
H.-J. Bungartz, B. Gatzhammer, F. Lindner, M. Mehl, K. Scheufele, A. Shukaev, B. Uekermann, 2016
In Computers and Fluids, Volume 141, p. 250—258. Elsevier.

Adapters **Official preCICE Adapters for Standard Open-Source Solvers**
B. Uekermann, H.-J. Bungartz, L. Cheung Yau, G. Chourdakis, A. Rusch, 2017
7th GACM Colloquium on Computational Mechanics for Young Scientists from Academia

Thesis 2 **A general OpenFOAM adapter for the coupling library preCICE**
G. Chourdakis, 2017
Master's thesis, Institut für Informatik, Technische Universität München

Thesis 1 **Conjugate Heat Transfer with the Multiphysics Coupling Library preCICE**
L. Cheung Yau, 2016
Master's thesis, Institut für Informatik, Technische Universität München

FOAM-FSI **Efficient numerical methods for partitioned fluid-structure interaction simulations**
D. Blom, 2017
Dissertation, Delft University of Technology

foam-extend + deal.II **Kopplung von OpenFOAM und deal.II Gleichungslösern mit preCICE zur Simulation multiphysikalischer Probleme**
K. Rave, 2017
Master's thesis, Lehrstuhl für Strömungsmechanik, Universität Siegen

Acknowledgements

The OpenFOAM adapter for preCICE is the result of two Master's Theses. We would like to thank:

DAAD (Germany) for supporting the M.Sc. studies of G. Chourdakis and L. Cheung Yau.

SENACYT (Panama) for supporting the M.Sc. studies of L. Cheung Yau.

SimScale GmbH for supporting the Master's Thesis of L. Cheung Yau.

preCICE itself is an academic project. Information on its funding sources: `www.precice.org/faq/`.

Questions?

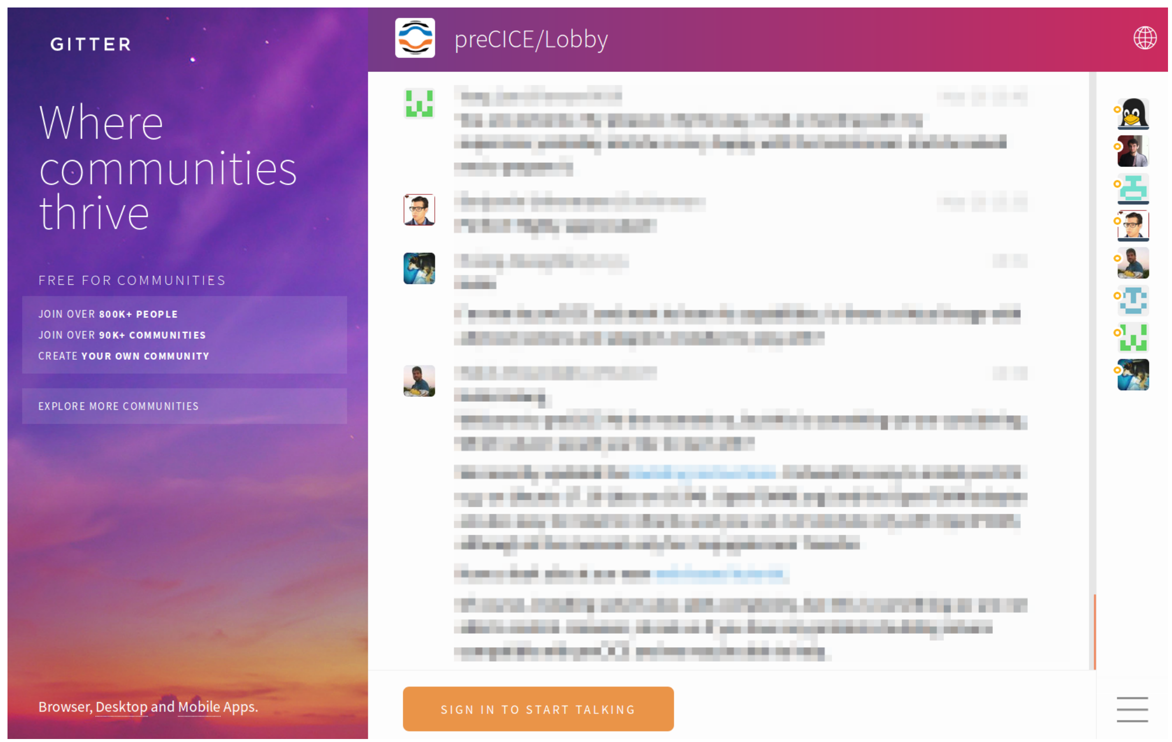
Website: `precice.org`
Source/Wiki: `github.com/precice` ☆

Mailing list: `precice.org/resources`
My e-mail: `gerasimos.chourdakis@tum.de`

Homework:

- Follow a tutorial
- Join our mailing list
- Star on GitHub
- Send us feedback
- Ask me for stickers

New: Ask questions on Gitter (experimental)



Questions?

Website: `precice.org`
Source/Wiki: `github.com/precice` ★

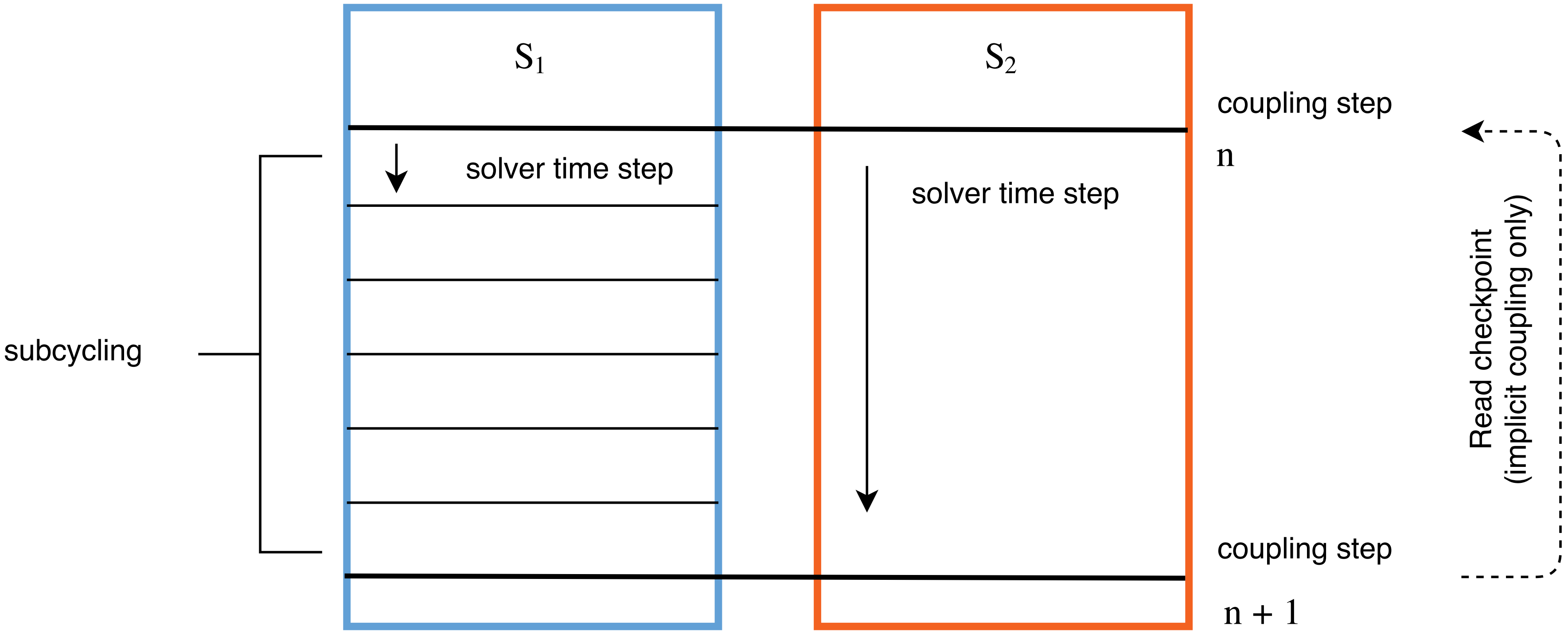
Mailing list: `precice.org/resources`
My e-mail: `gerasimos.chourdakis@tum.de`

Homework:

- Follow a tutorial
- Join our mailing list
- Star on GitHub
- Send us feedback
- Ask me for stickers

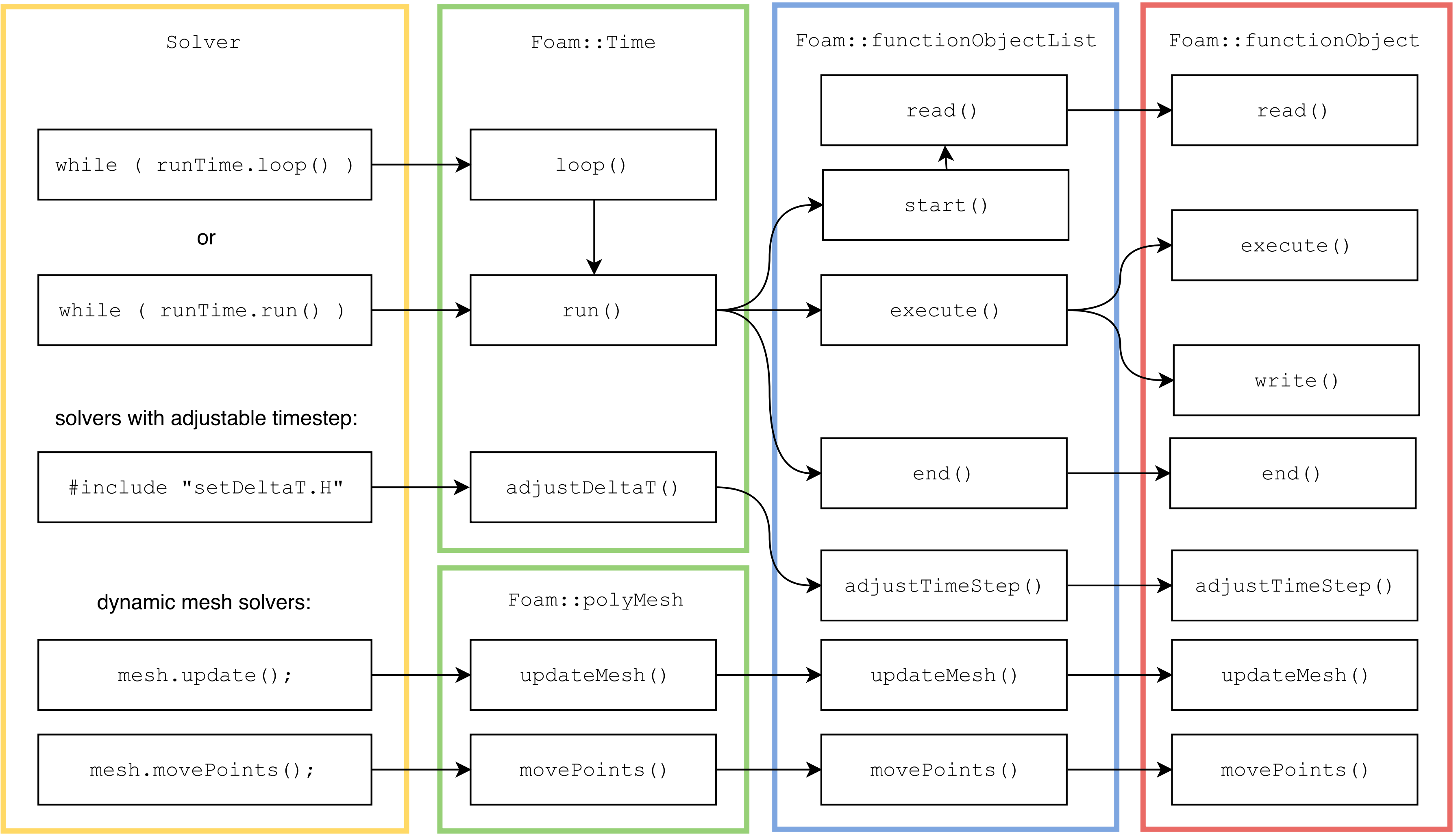
preCICE

Additional slide: Time step sizes



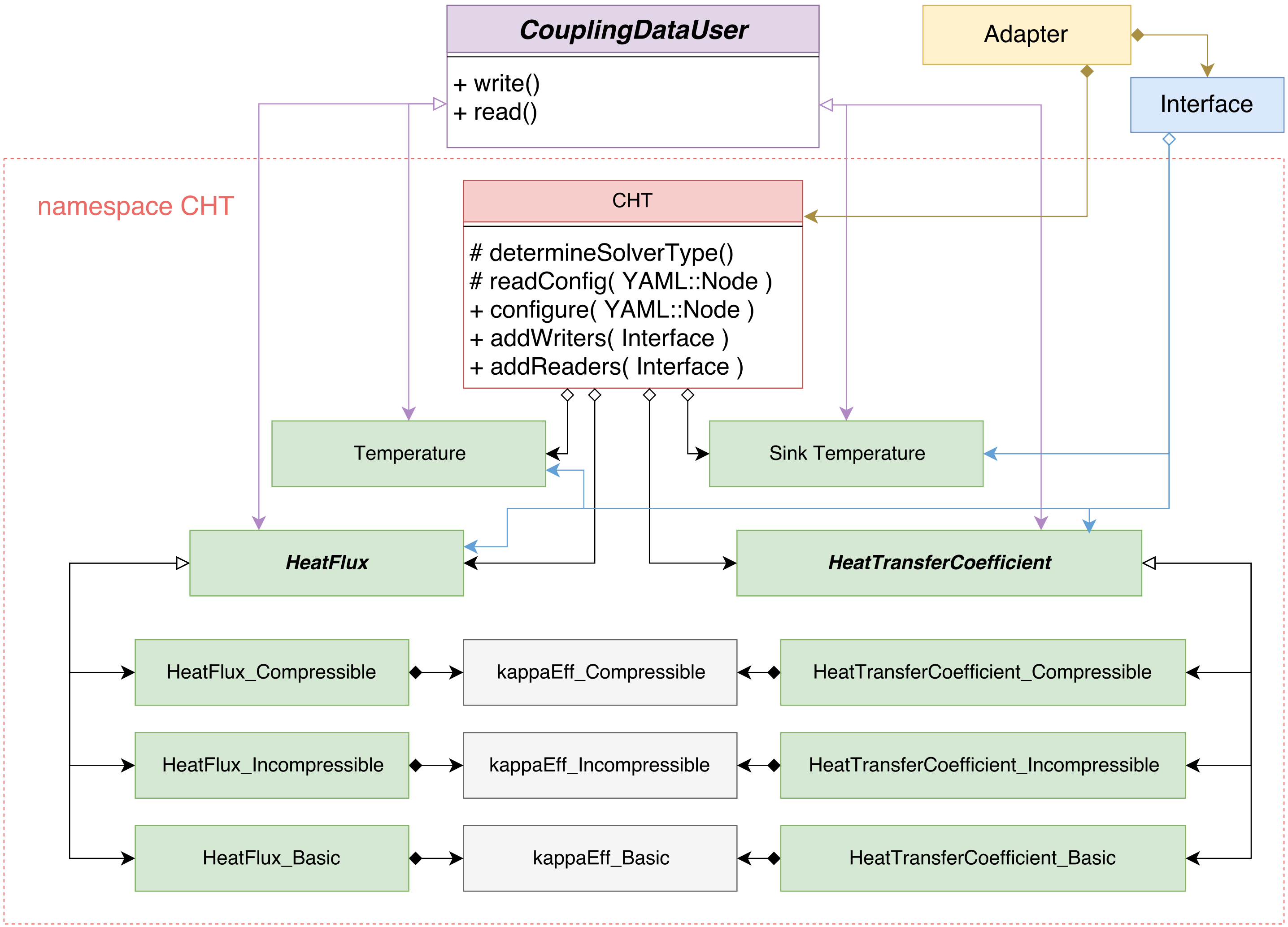
Time step sizes and checkpointing

Additional slide: Function Objects



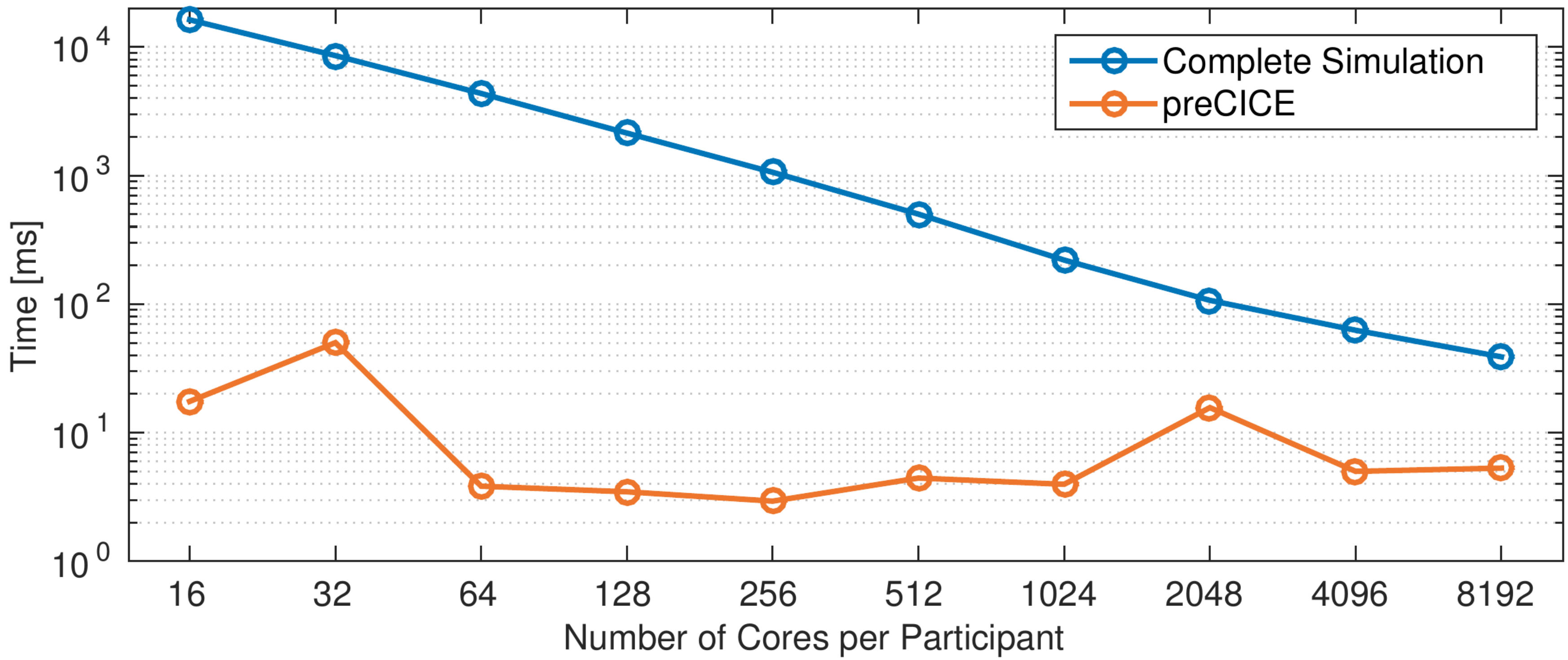
Callbacks in OpenFOAM function objects

Additional slide: The CHT Module



The Conjugate Heat Transfer module

Additional slide: preCICE scaling



Strong scaling of a coupled simulation with two Ateles participants and $5.7 \cdot 10^7$ *dofs*