

# Couple OpenFOAM with any other solver using preCICE

Gerasimos Chourdakis et al.

Technical University of Munich

Department of Informatics

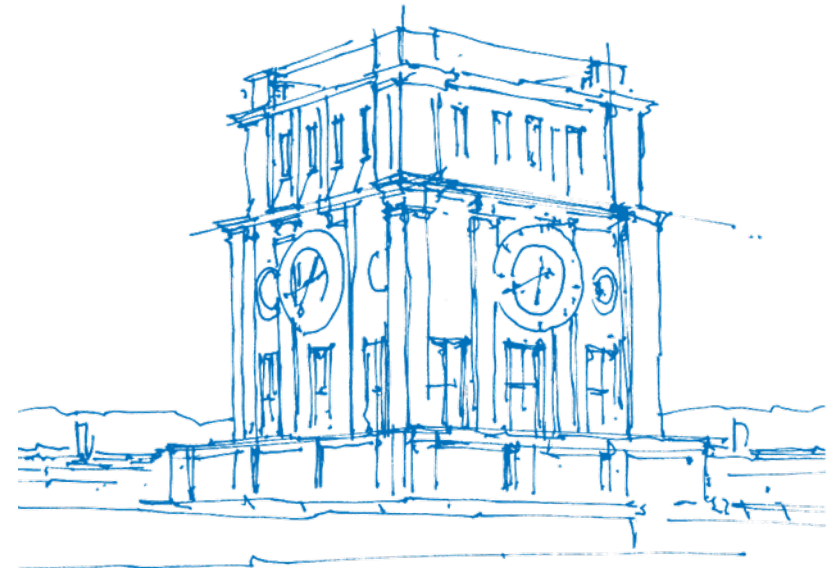
Chair of Scientific Computing in Computer Science

OpenFOAM Workshop

Dutch OpenFOAM Users' Group

TU Delft

February 6, 2018



*TUM Uhrenturm*

# Agenda

## Part I:



# Agenda

Part I:



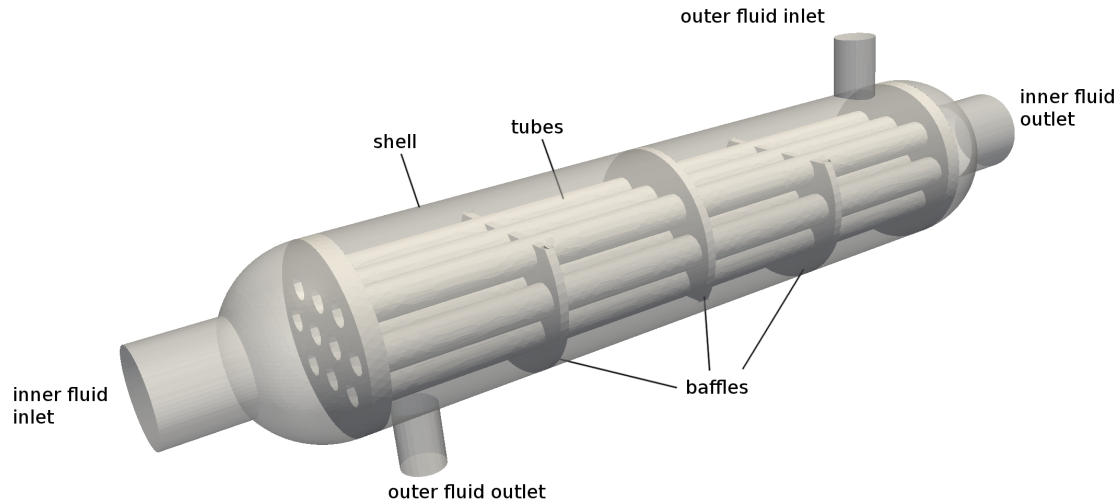
Part II:



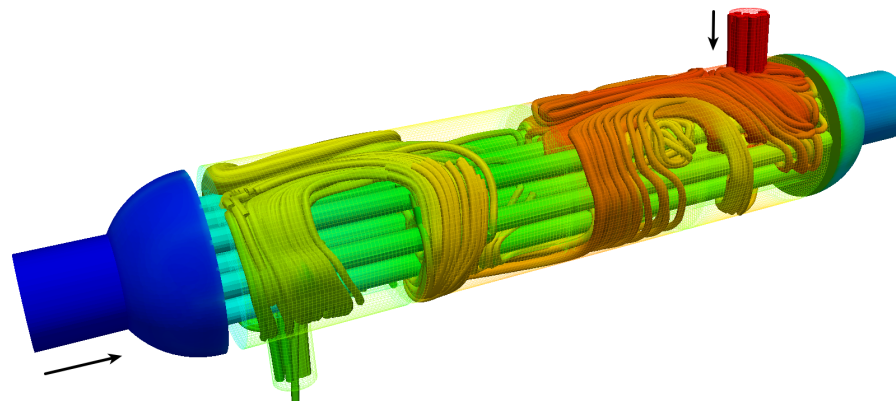
+



# How to simulate this heat exchanger?

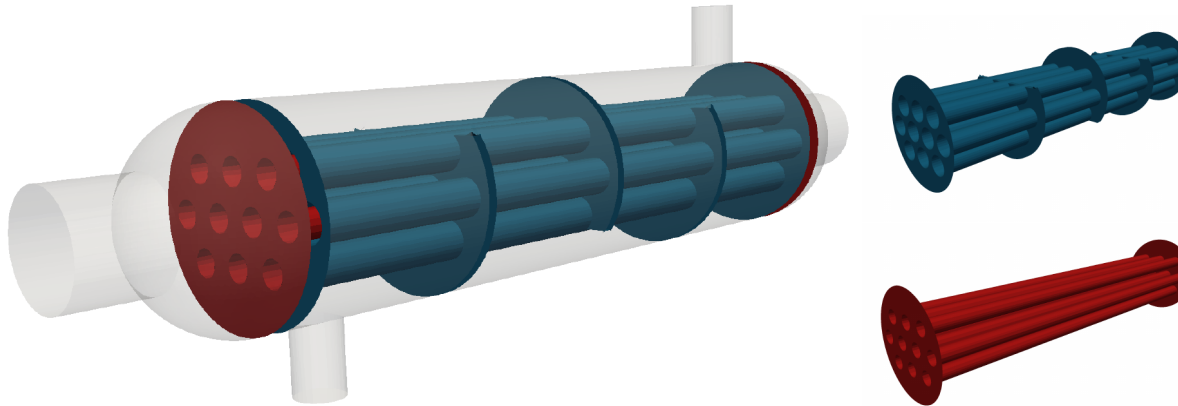


Geometry of a shell-and-tube heat exchanger (Image by L. Cheung Yau, 2016)

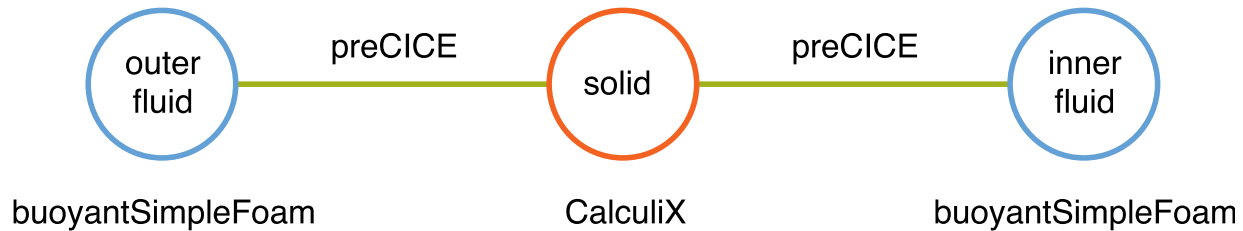


Surface plot and streamlines of the two fluids colored by temperature. Solid not shown.

# A shell-and-tube heat exchanger with preCICE

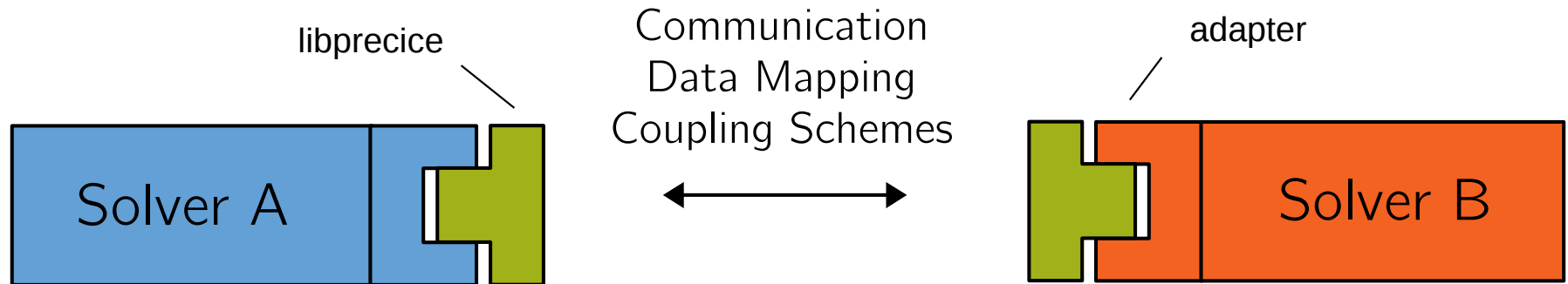


Coupling interfaces (Image by L. Cheung Yau, 2016)



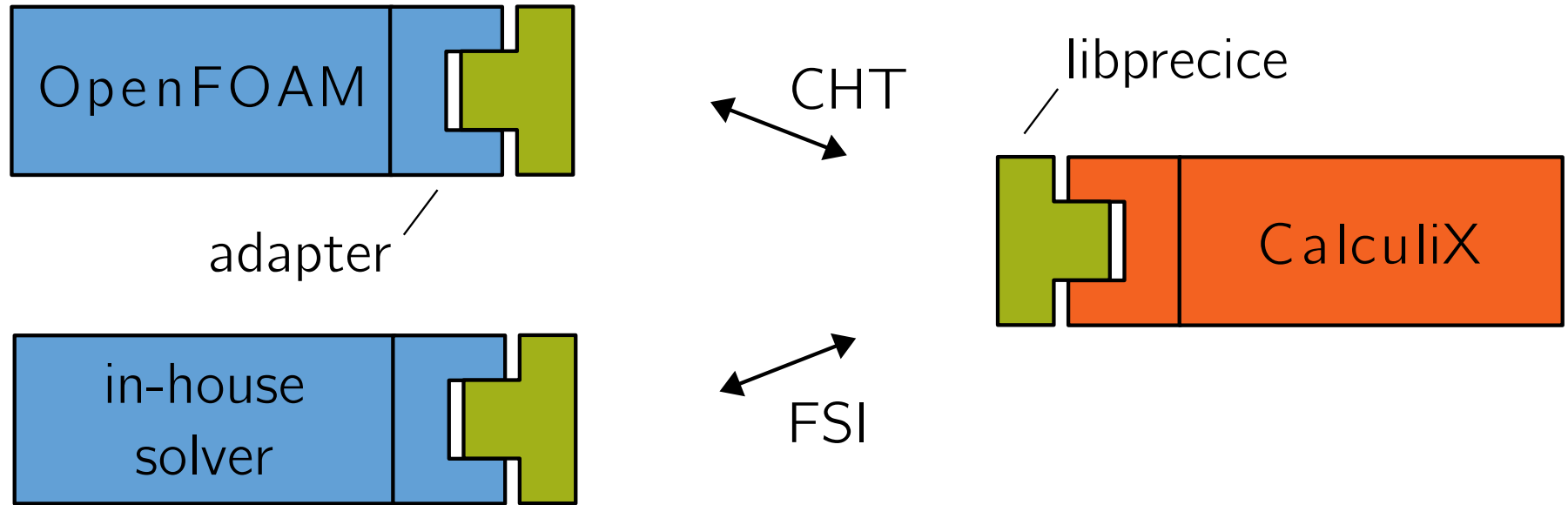
# preCICE

*precise Code Interaction Coupling Environment*



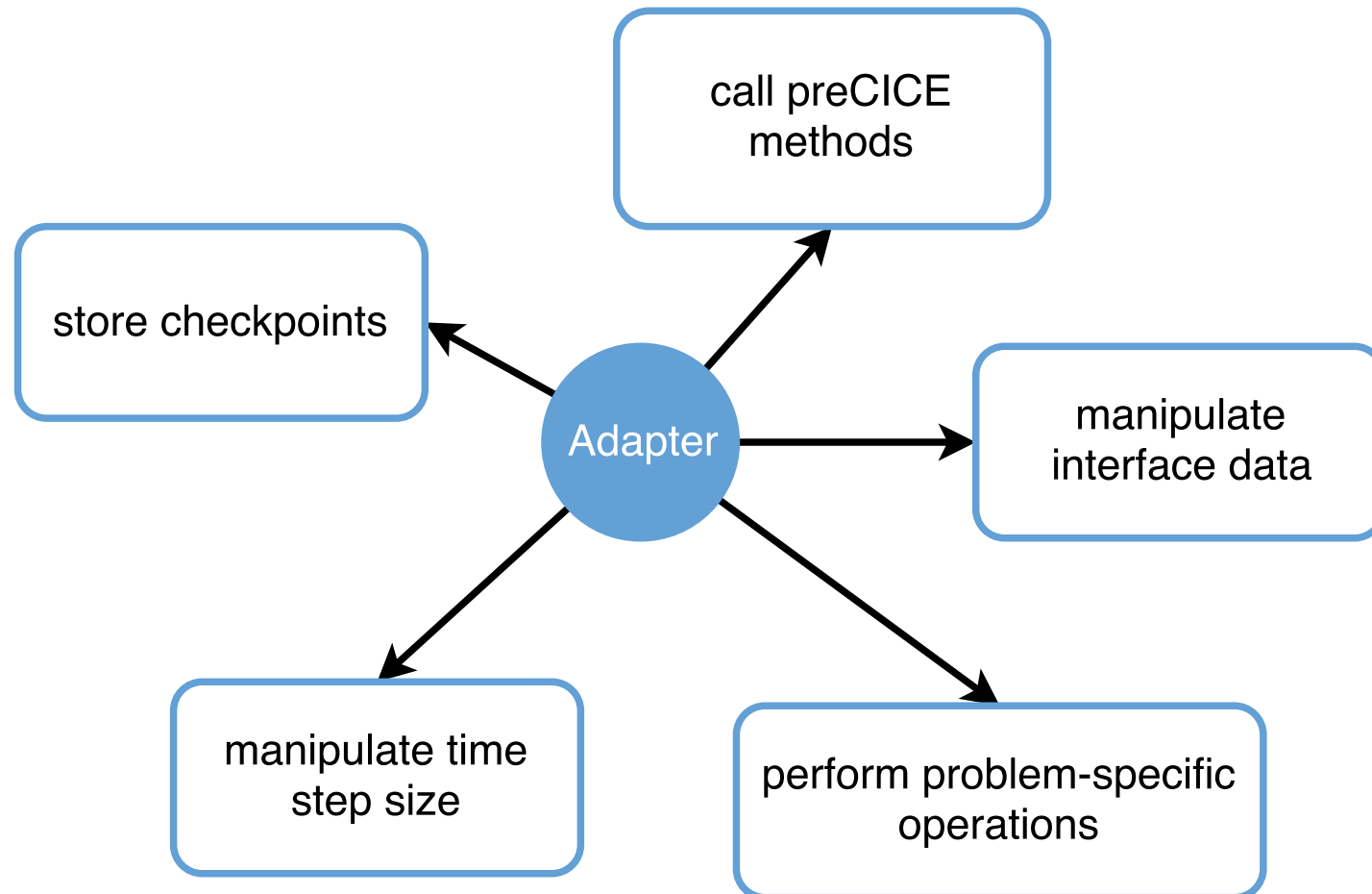
- **Free (GNU LGPL)**, developed at TU Munich & Univ. of Stuttgart.
- **Version 1.0** in November 2017 (10+ years, 3 PhD generations).
- **Official adapters** for CalculiX, Code\_Aster, COMSOL, Fluent, **OpenFOAM**, SU2
- **Third-party adapters** for Ateles, Alya, Carat++, FASTEST, FEAP, **foam-extend**, ...
- **API** in C, **C++**, Fortran, Python

# But why preCICE?



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

# The roles of an adapter



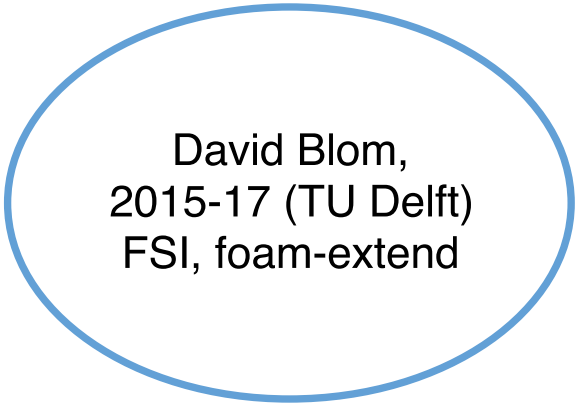


# Part IIa: previous approach



# Duplicated development effort

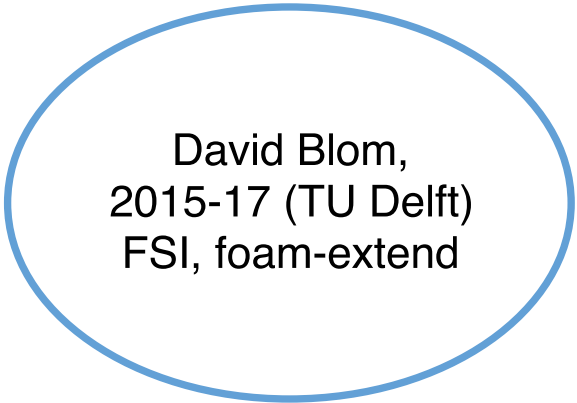
OpenFOAM (and family) adapters for preCICE



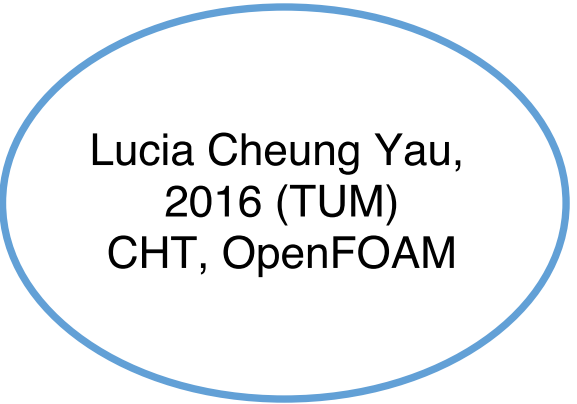
David Blom,  
2015-17 (TU Delft)  
FSI, foam-extend

# Duplicated development effort

OpenFOAM (and family) adapters for preCICE



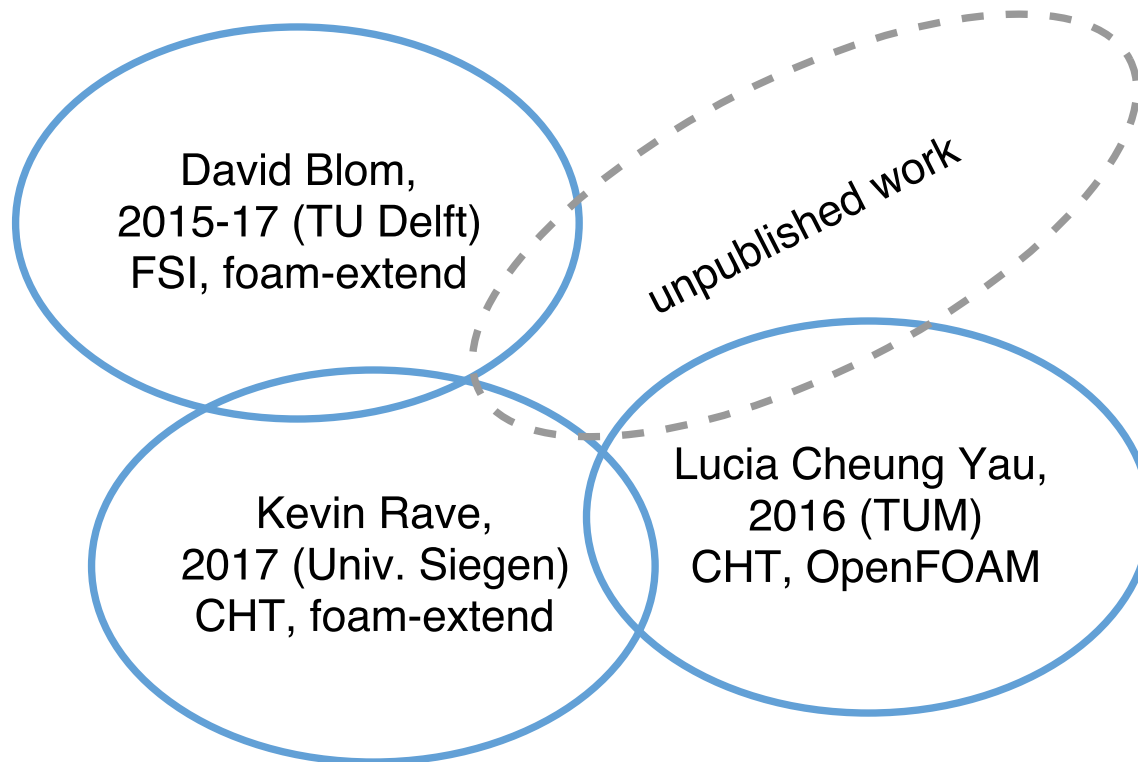
David Blom,  
2015-17 (TU Delft)  
FSI, foam-extend



Lucia Cheung Yau,  
2016 (TUM)  
CHT, OpenFOAM

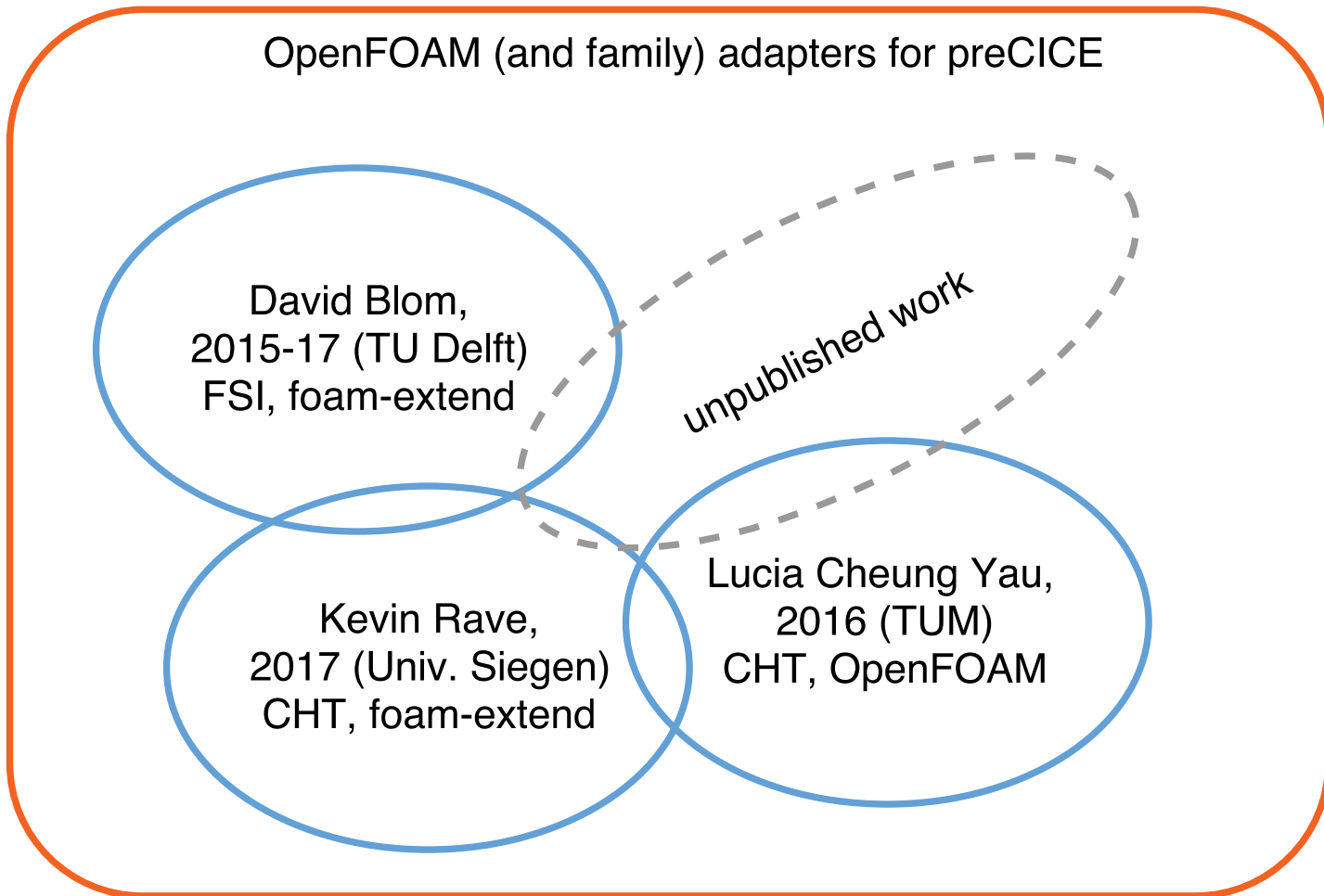
# Duplicated development effort

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)

```
1  /* Adapter: Initialize coupling */
2  adapter.initialize();
3
4  Info<< "\nStarting time loop\n" << endl;
5
6  while (adapter.isCouplingOngoing()) {
7      #include "readTimeControls.H"
8      #include "compressibleCourantNo.H"
9      #include "setDeltaT.H"
10
11     /* Adapter: Adjust solver time */
12     adapter.adjustSolverTimeStep();
13
14     /* Adapter: Write checkpoint */
15     if(adapter.isWriteCheckptRequired())
16         adapter.writeCheckpoint();
17
18     runTime++;
19
20     /* Adapter: Receive coupling data */
21     adapter.readCouplingData();
```

```
22     /* solve the equations */
23     #include "rhoEqn.H"
24     while (pimple.loop())
25     {
26         ...
27     }
28
29     /* Adapter: Write in buffers */
30     adapter.writeCouplingData();
31
32     /* Adapter: advance the coupling */
33     adapter.advance();
34
35     /* Adapter: Read checkpoint */
36     if(adapter.isReadCheckptRequired())
37         adapter.readCheckpoint();
38
39     if(adapter.isCouplTimeStepComplete())
40         runTime.write();
41
42 }
```

# Before: Working and validated prototypes

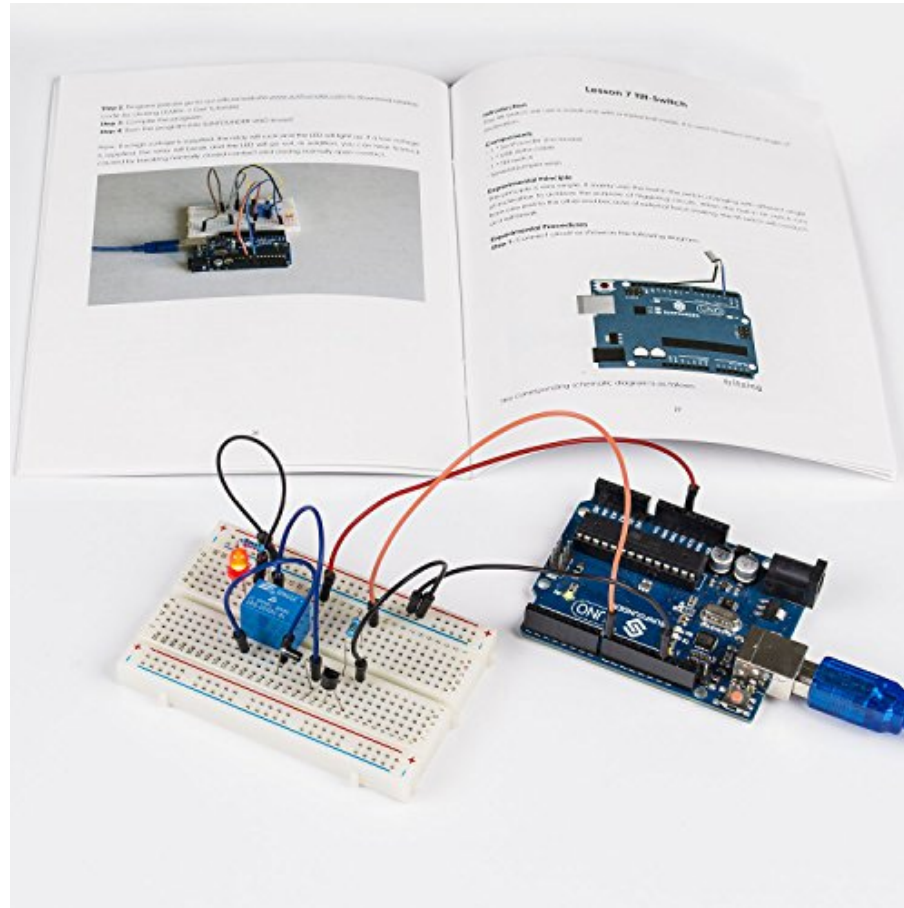


Image from desertcart.ae.

# Before: Working and validated prototypes

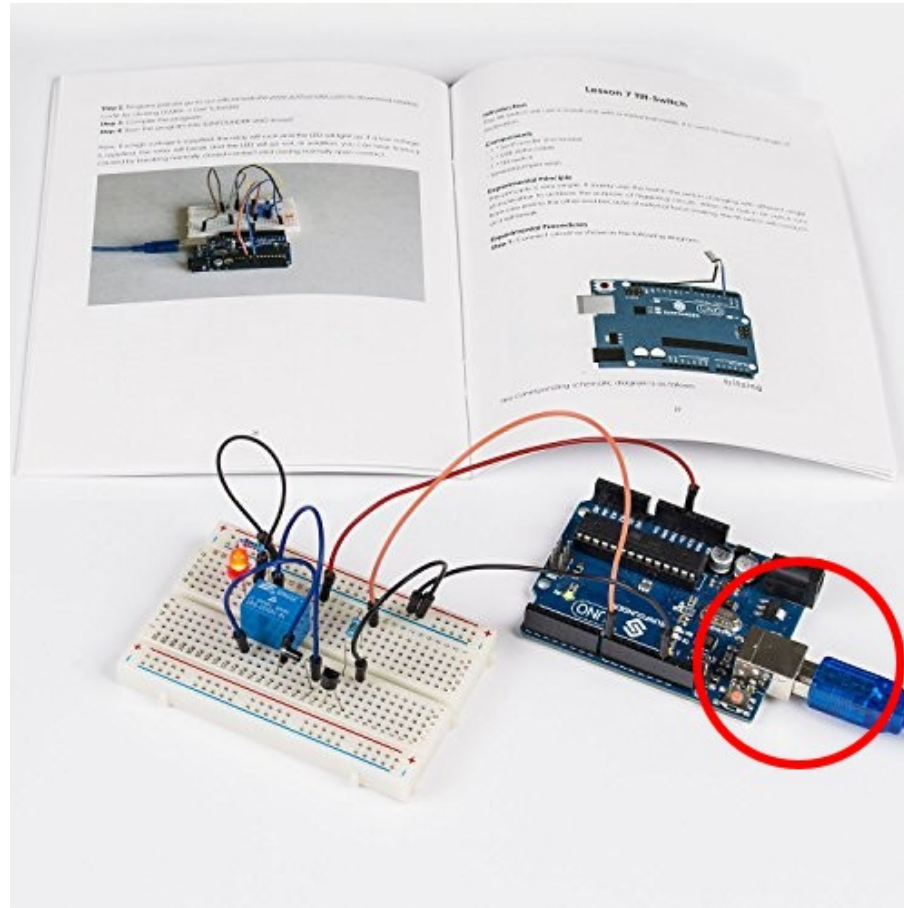
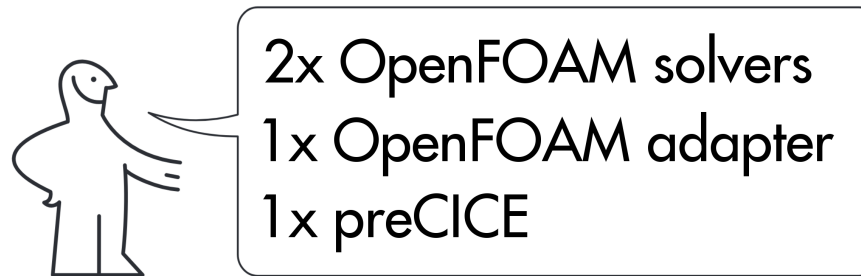


Image from desertcart.ae.



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

# KOPPLAD



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

# Part IIb: a new, official adapter



# Making this a function object

## 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

## 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

## 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`?
- ...

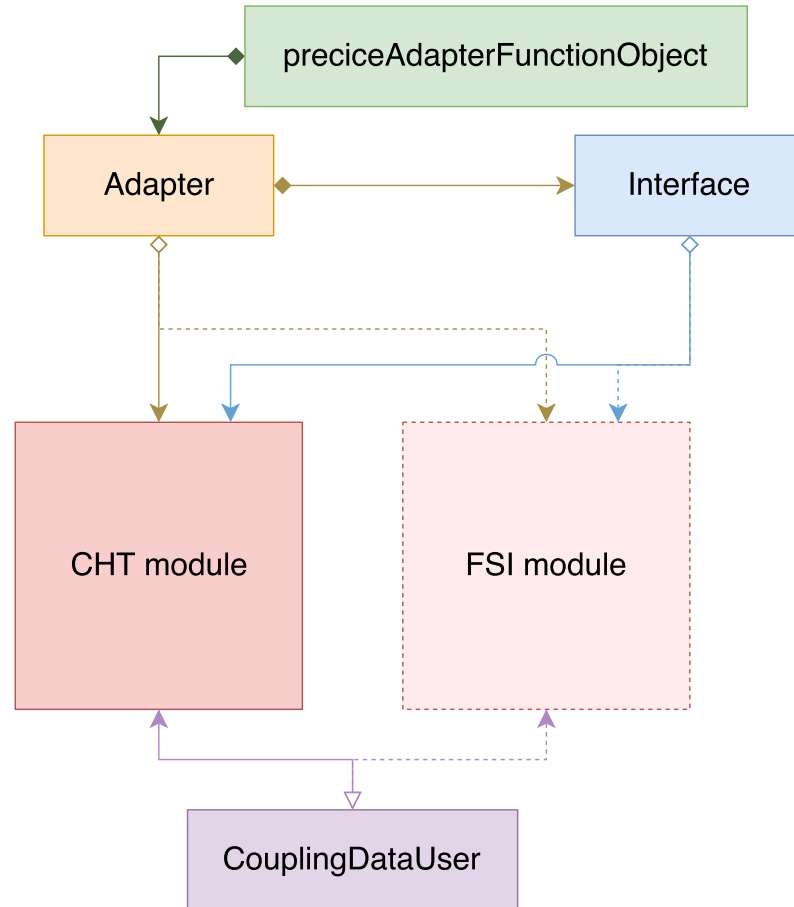
## Several **advantages**:

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

## **However:**

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

# An extensible adapter



OK! I want to use it!

# OpenFOAM configuration

---

```
1 // system/controlDict
2 functions
3 {
4     preCICE_Adapter
5     {
6         type preciceAdapterFunctionObject;
7         libs ("libpreciceAdapterFunctionObject.so");
8     }
9 }
```

---

Set the appropriate boundary condition types:

```
1 // 0/T
2 interface
3 {
4     type          fixedValue;
5     value         uniform 300;
6 }
7
8 // other types: fixedGradient, mixed
```

---



```
1 // system/controlDict
2 functions
3 {
4     preCICE_Adapter
5     {
6         type preciceAdapterFunctionObject;
7         libs ("libpreciceAdapterFunctionObject.so");
8     }
9 }
```

Set the appropriate boundary condition types:

```
1 // 0/T
2 interface
3 {
4     type          fixedValue;
5     value         uniform 300;
6 }
7
8 // other types: fixedGradient, mixed
```

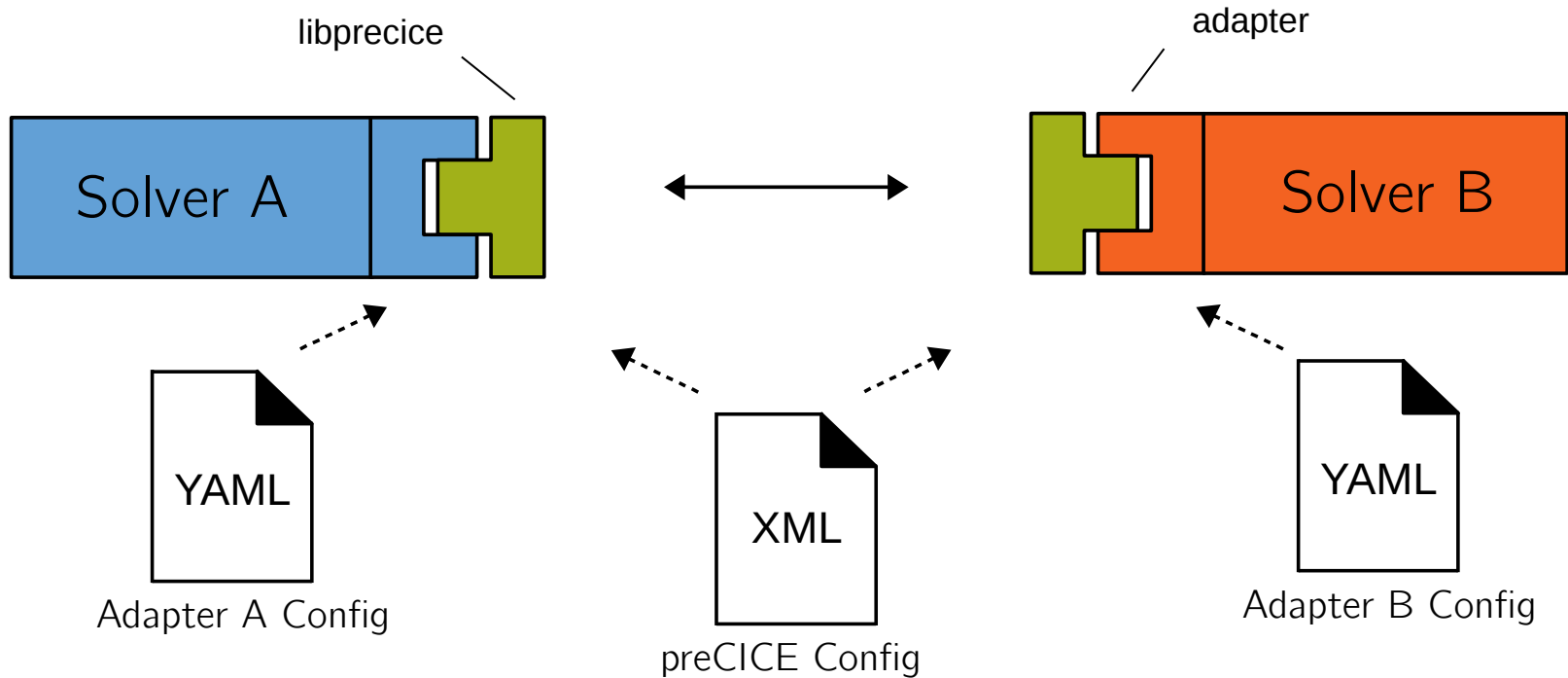
Properties for incompressible solvers:

```
1 // constant/transportProperties
2 rho    rho [ 1 -3 0 0 0 0 0 ] 1;
3 Cp     Cp [ 0 2 -2 -1 0 0 0 ] 5000;
```

Properties for basic solvers:

```
1 // constant/transportProperties
2 k      k [ 1 1 -3 -1 0 0 0 ] 100;
```

# preCICE & adapter configuration



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

## Fluid-Structure Interaction

<a href="#"><u>1D FSI Example</u></a>	<a href="#"><u>FSI with SU2 and CalculiX</u></a>
Flow through a deformable tube	Flow in a channel with an elastic flap

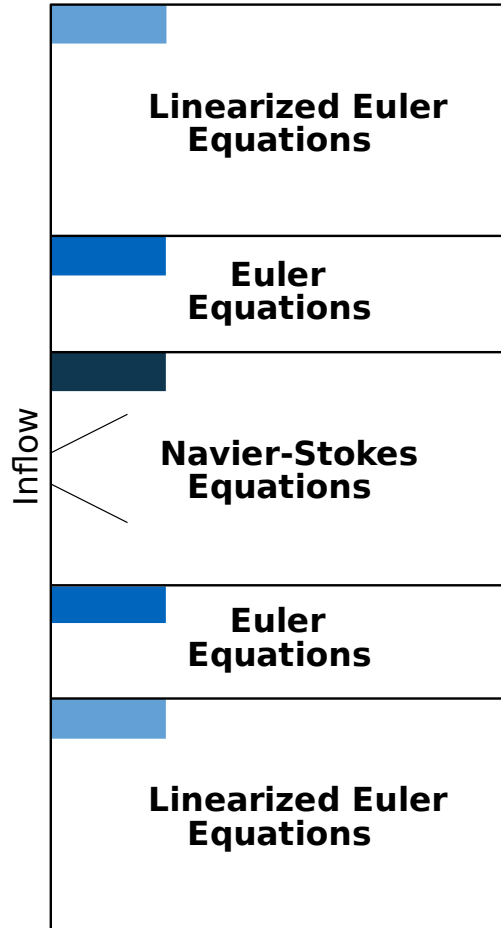
## Conjugate Heat Transfer

<a href="#"><u>CHT with OpenFOAM</u></a>	<a href="#"><u>CHT with OpenFOAM and CalculiX</u></a>
Flow above a heated plate	Shell-and-tube heat exchanger

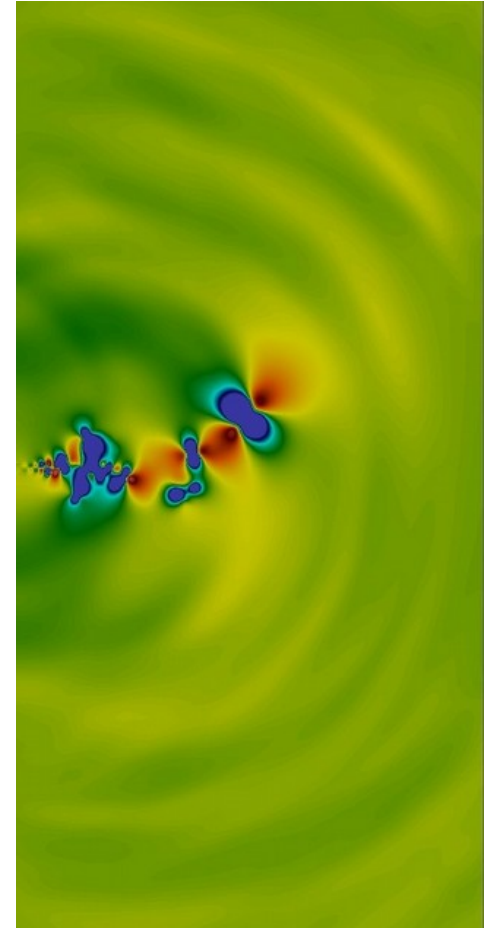
# Application examples



FSI: Aortic bloodflow



Multi-fluid coupling



# 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

Coming soon:

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

## Support OpenFOAM 3.x #8

 MakisH opened this issue on Nov 28, 2017 · 0 comments



MakisH commented on Nov 28, 2017 · edited by floli

Collaborator +😊 ✎ 🗨

While openfoam.org versions 4.x have no changes that affect the adapter in comparison to versions 5.x, there are some differences in older versions.

Versions 3.x are different from 4.x in at least the following points:

- The way that the `boundaryField` and `internalField` are accessed. In the [OpenFOAM 4.0 release notes](#), it is stated:

Robust data handling: new convention for const and non-const reference functions of fields where the non-const function uses `...Ref()`; for example, where `boundaryField()` provides the const reference to the boundary field, `boundaryFieldRef()` provides a non-const reference. for tmp objects, non-const access uses a `ref()` function rather than the `()` dereferencing operator.

See also the [OpenFOAM-dev commit a4e2afa](#) and [this issue](#).

- The class hierarchy for turbulence models, as version 4.0 introduced the templated class `TurbulenceModels`. See the [OpenFOAM-4.x commit 93732c8](#).
- Small changes in the function objects. See the OpenFOAM 4.x commit [91aba2d](#). You may find a function objects code template for your version in the `$FOAM_ETC/` directory. The files [preCiceAdapterFunctionObject.H](#) and [preCiceAdapterFunctionObject.C](#) need to be adjusted.

Contribute on GitHub!

# Questions?

Website: [precice.org](http://precice.org)

Source/Wiki: [github.com/precice](https://github.com/precice) ☆

Mailing list: [precice.org/resources](http://precice.org/resources)

My e-mail: [gerasimos.chourdakis@tum.de](mailto:gerasimos.chourdakis@tum.de)



## Homework:

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



# Questions?

Website: [precice.org](http://precice.org)

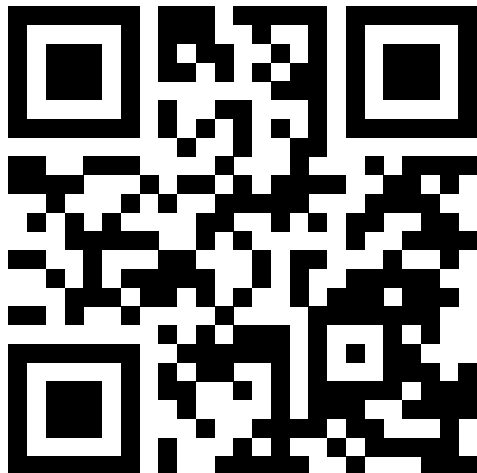
Source/Wiki: [github.com/precice](https://github.com/precice) ☆

Mailing list: [precice.org/resources](http://precice.org/resources)

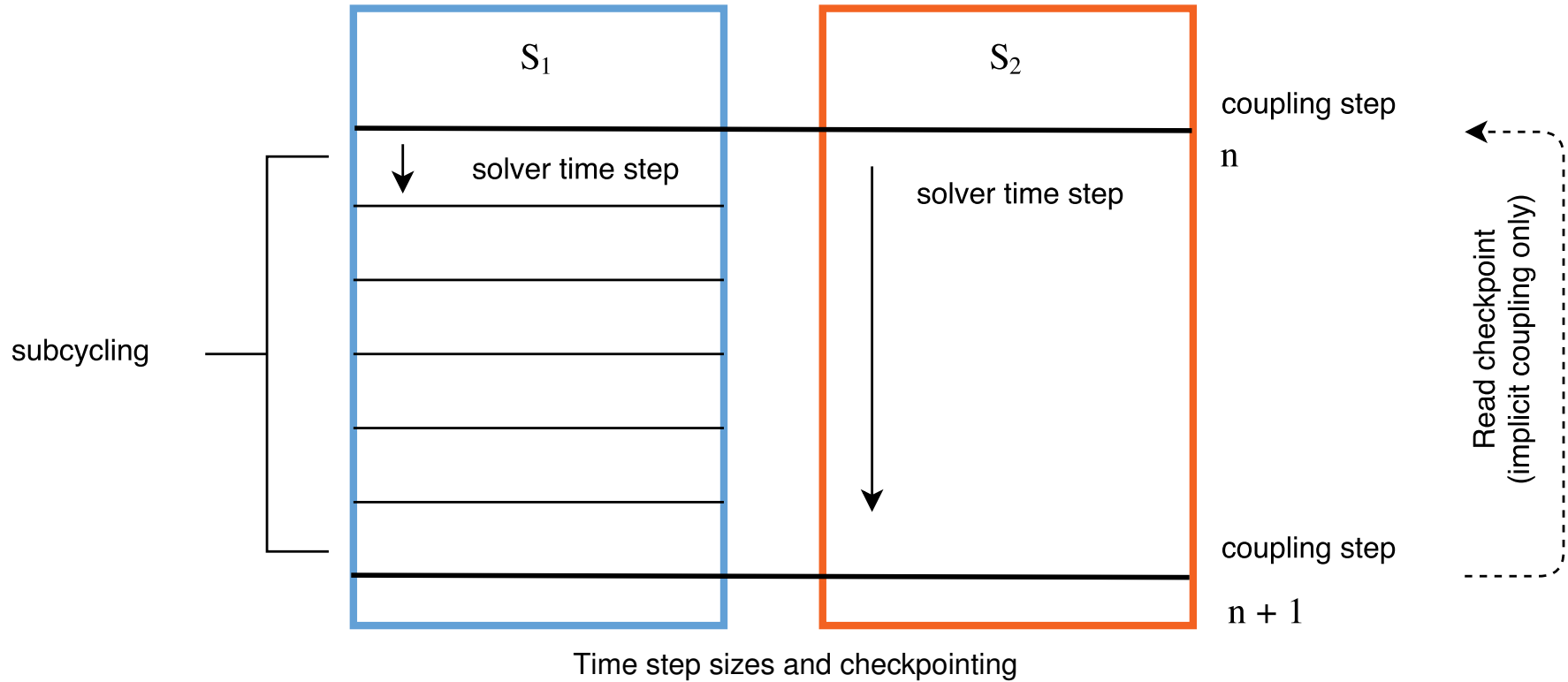
My e-mail: [gerasimos.chourdakis@tum.de](mailto:gerasimos.chourdakis@tum.de)

## Homework:

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

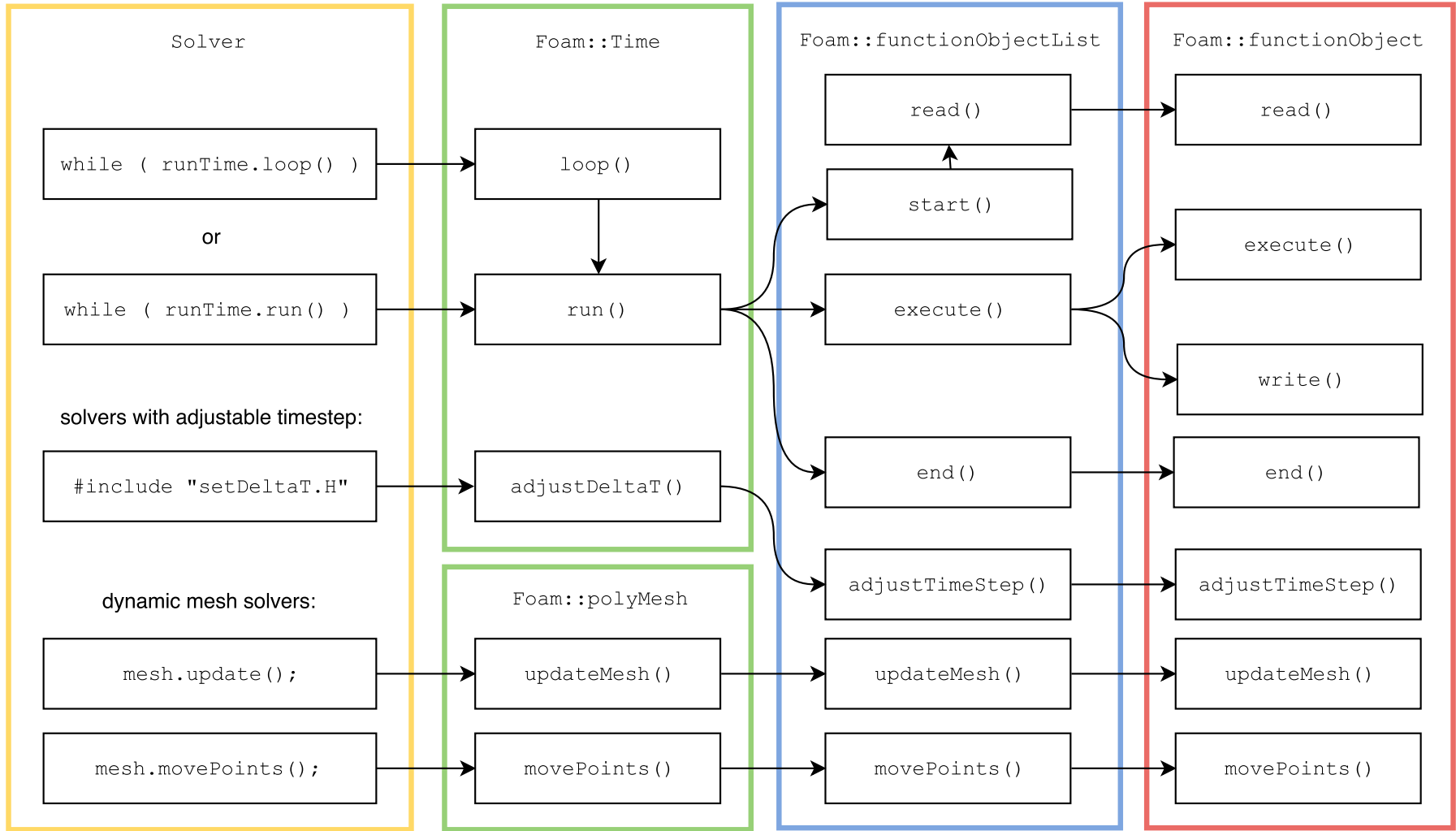


# Additional slide: Time step sizes



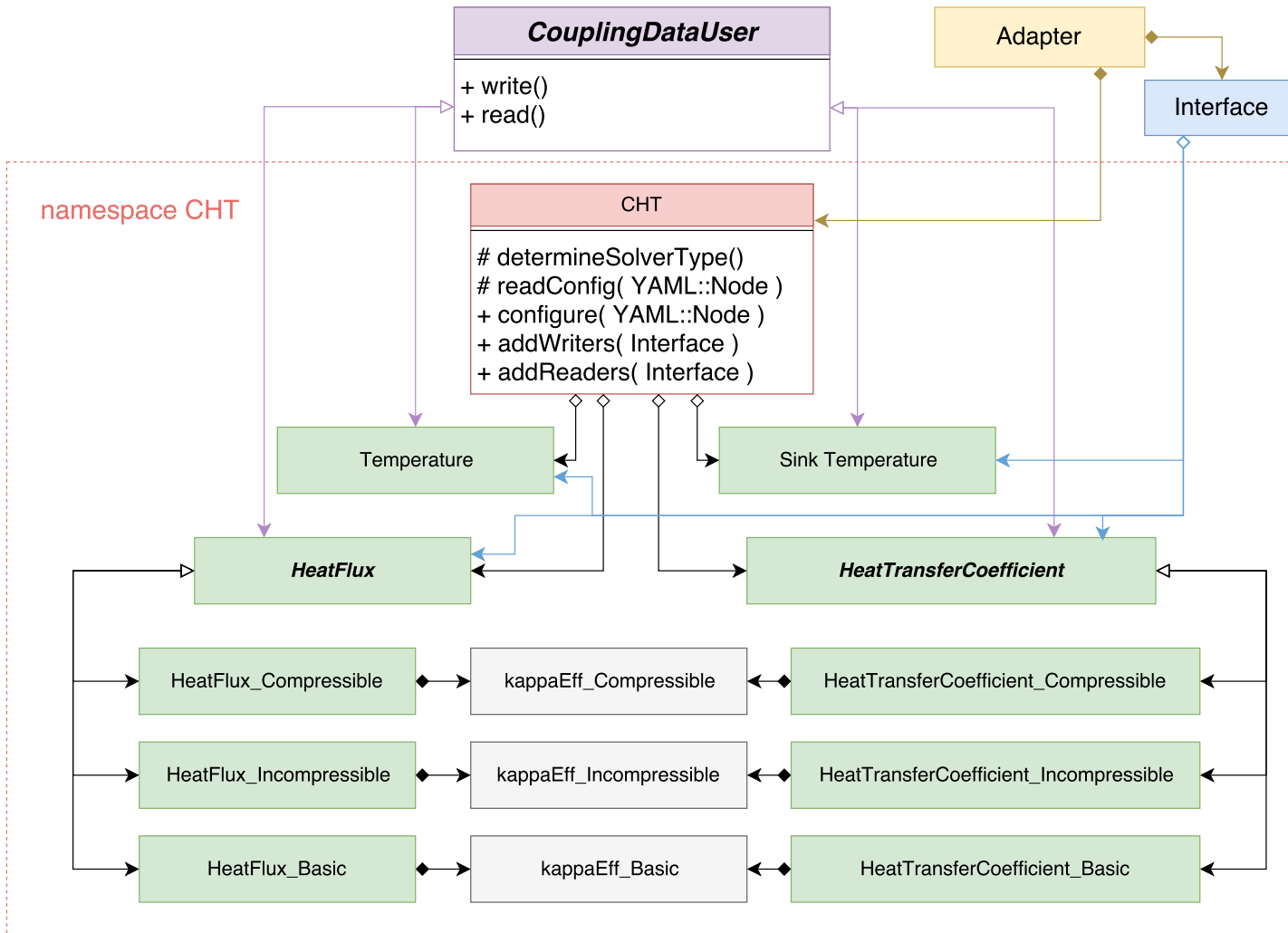


# Additional slide: Function Objects



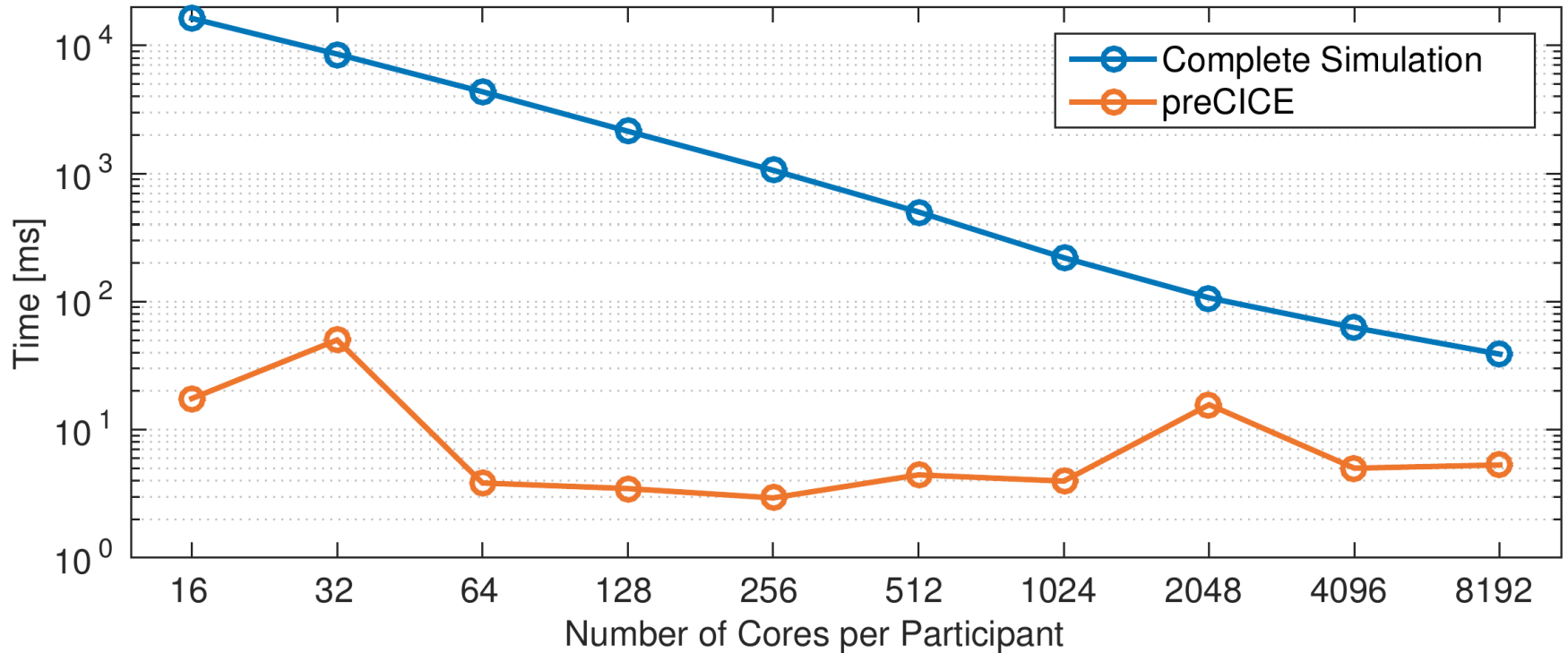
Callbacks in OpenFOAM function objects

# 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*