

FISOC: Framework for Ice Sheet – Ocean Coupling

Rupert Gladstone, Ben Galton-Fenzi, David Gwyther, Lenneke Jong



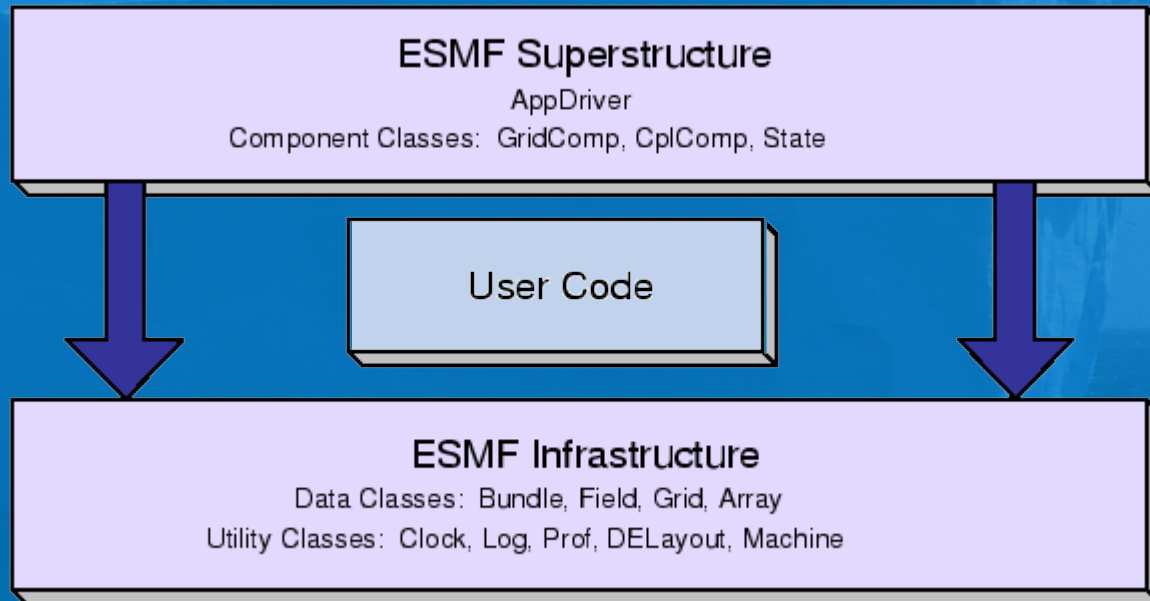
Australian Government
Department of the Environment
Australian Antarctic Division



Contents

- Third party coupling software: Earth System Modelling Framework (ESMF).
- FISOC overview: aims and design ethos.
- FISOC overview: code organisation and structures.
- Elmer/Ice details: build specifics.
- Elmer/Ice details: code modifications.
- Elmer/Ice details: runtime changes and asynchronicity.
- Output visualisation with Paraview.

What is the Earth System Modelling Framework (ESMF)?

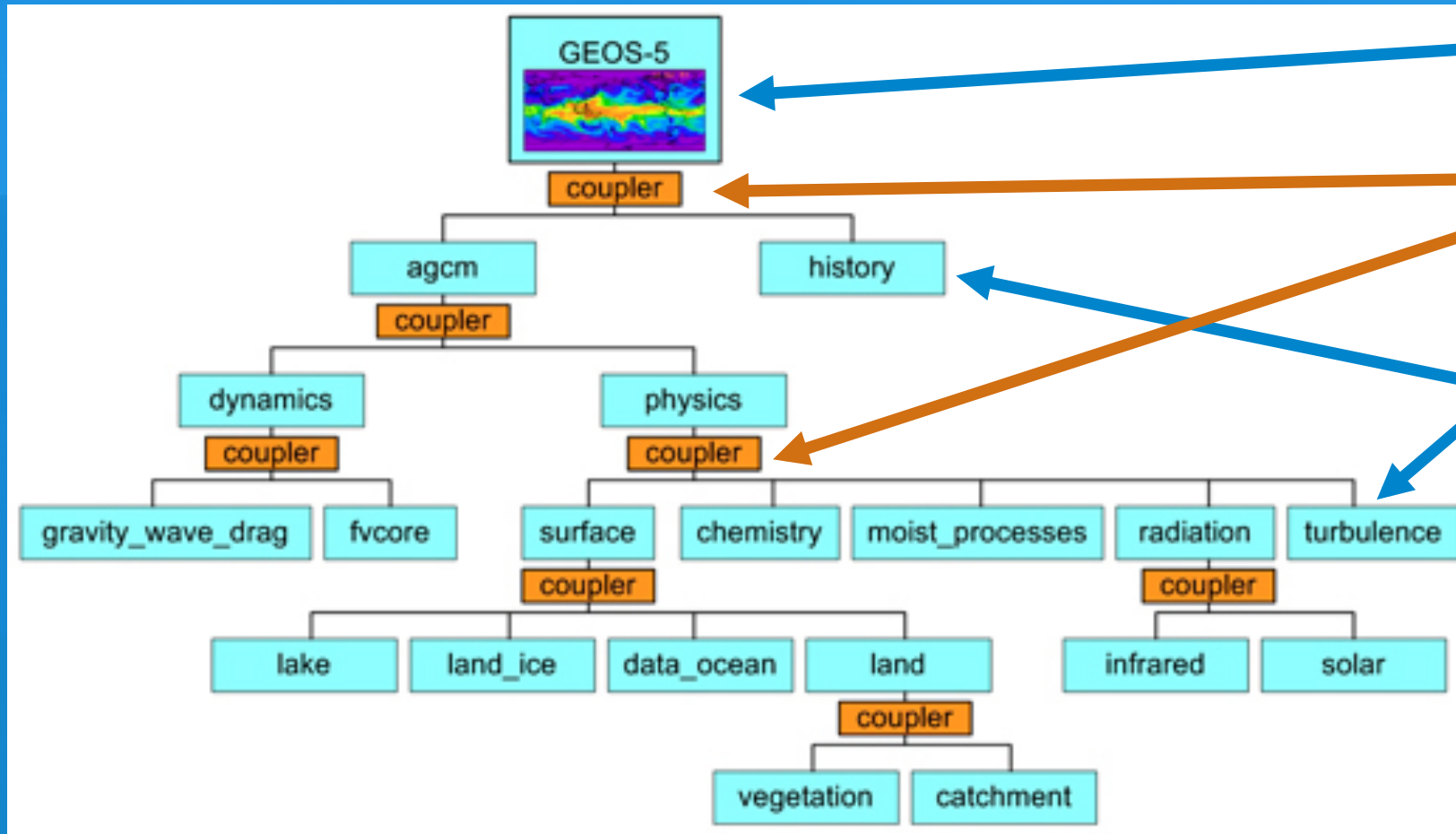


“The *Earth System Modeling Framework (ESMF)* is high-performance, flexible software infrastructure for building and coupling weather, climate, and related Earth science applications.”

Component based architecture, where a “component” is either a (sub) model or a coupler (e.g. Elmer/Ice is a component).

Provides superstructure (e.g. drivers, wrappers) and infrastructure (e.g. fields, grids, clock utilities)

Example ESMF component hierarchy



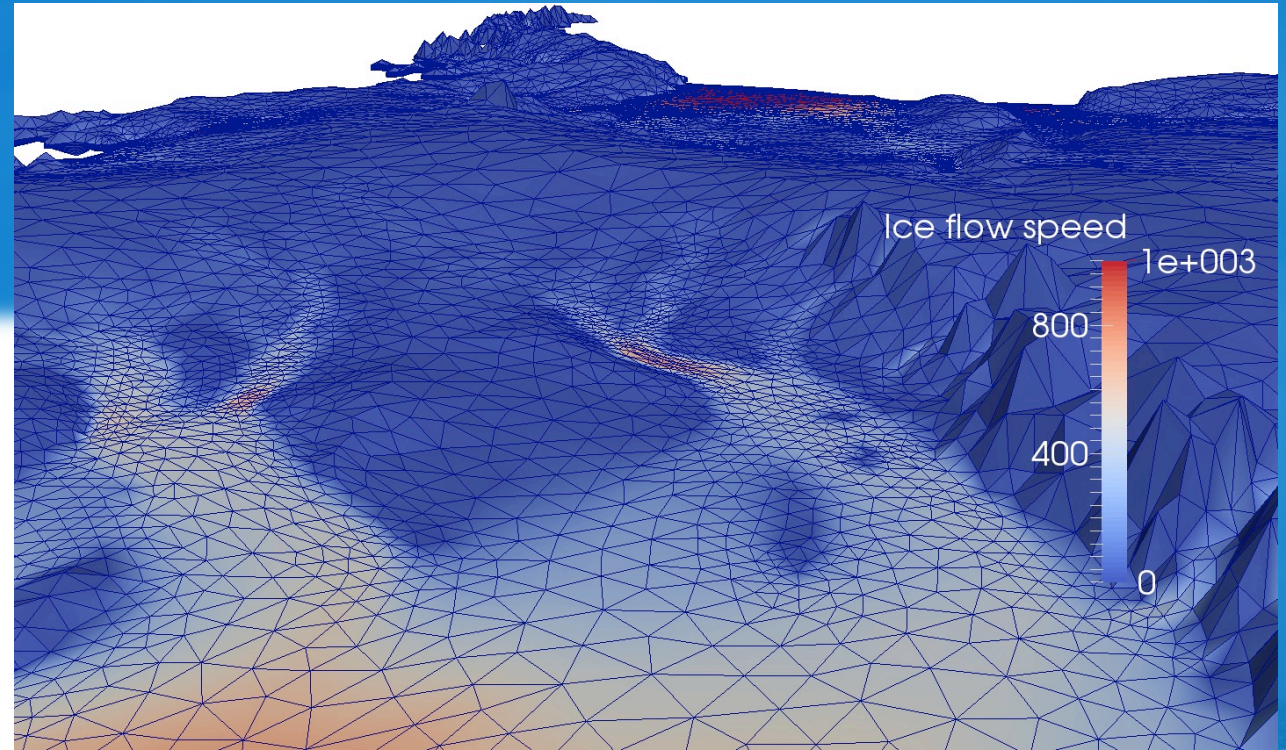
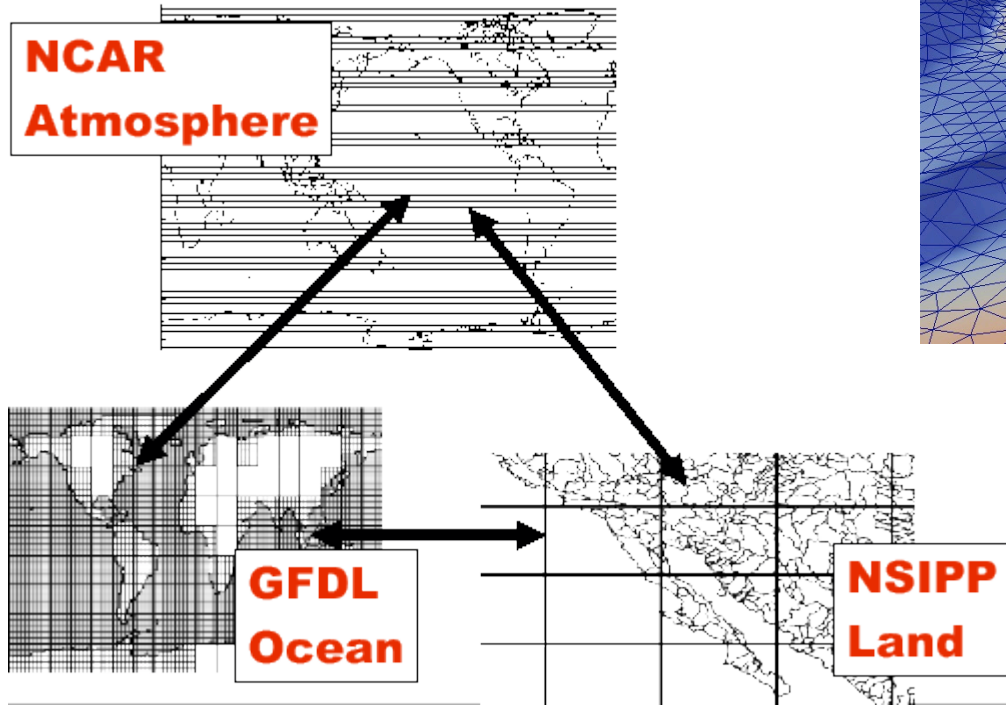
Top level driver

Coupler components exchange data and call regridding routines

Gridded components wrap individual models or sub/models.

Plots/quote from ESMF web site <https://www.earthsystemcog.org/projects/esmf/>

ESMF regridding options include unstructured meshes



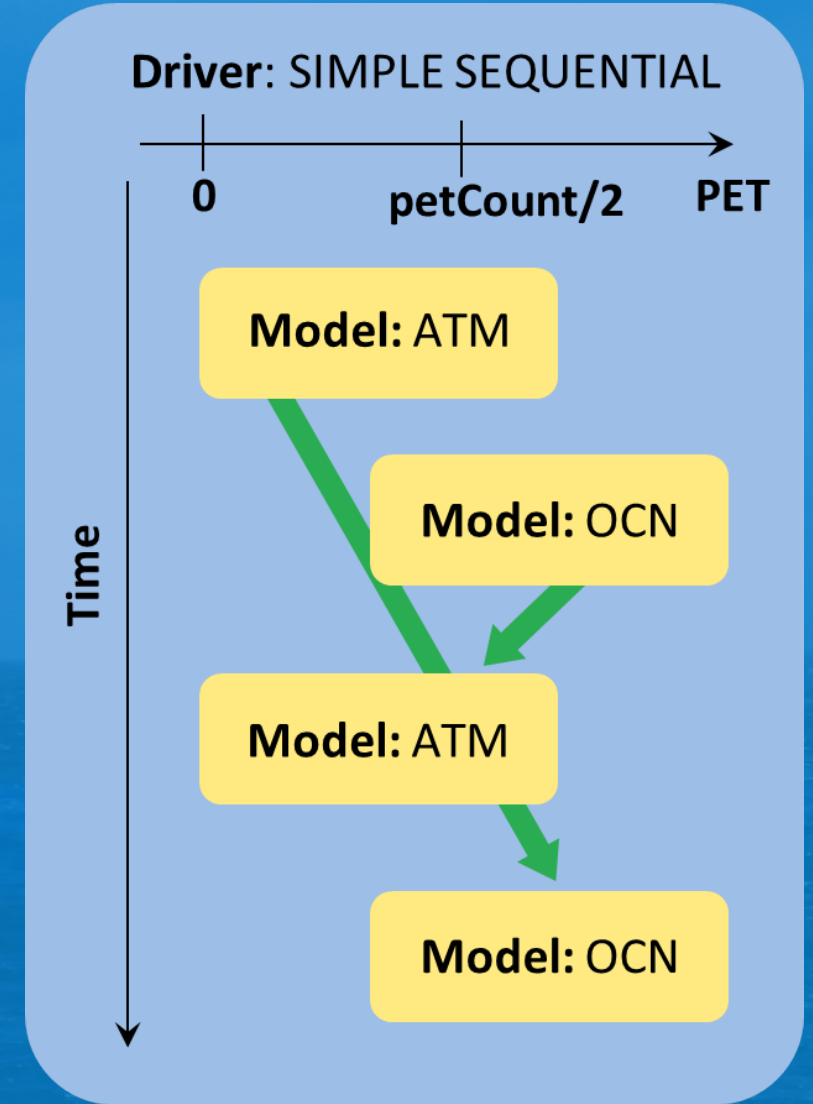
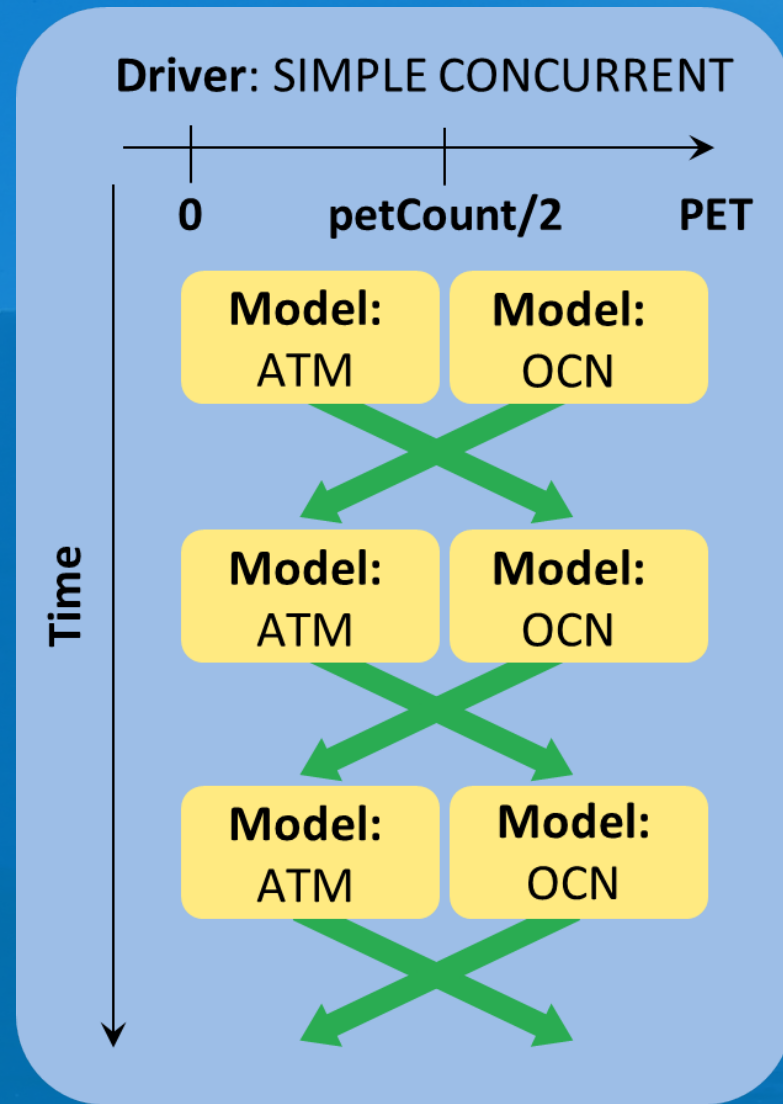
Elmer/Ice example mesh

ESMF example grids

ESMF parallelism options: concurrent vs sequential (combinations and more complex models are possible).

FISOC currently takes a sequential approach on all PETs.

ESMF terminology: Persistent Execution Thread (PET) is an abstraction of the concept of a process



FISOC Concept

Concept: flexible coupling framework to allow Ice Sheet Models (ISMs) and Ocean Models (OMs) to interact at run time

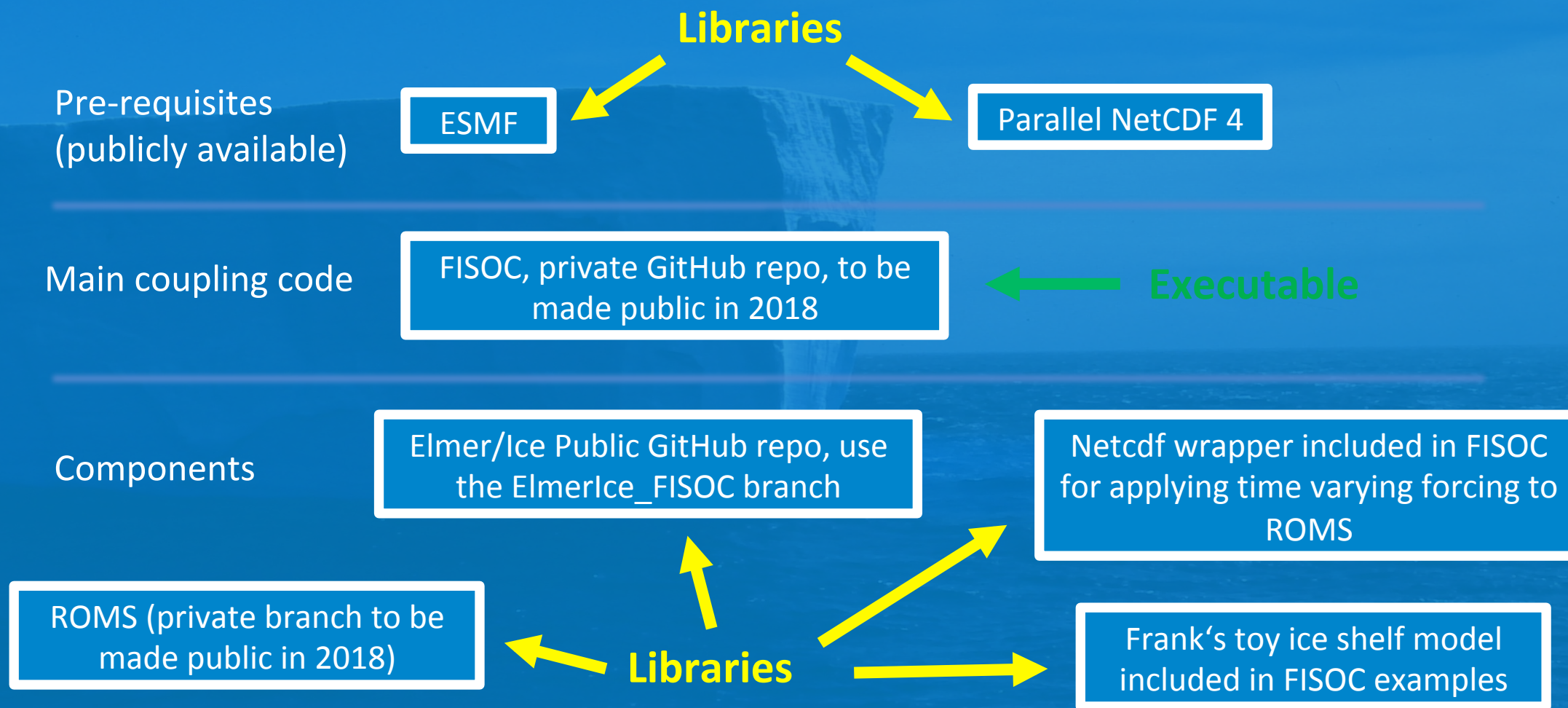
Key features:

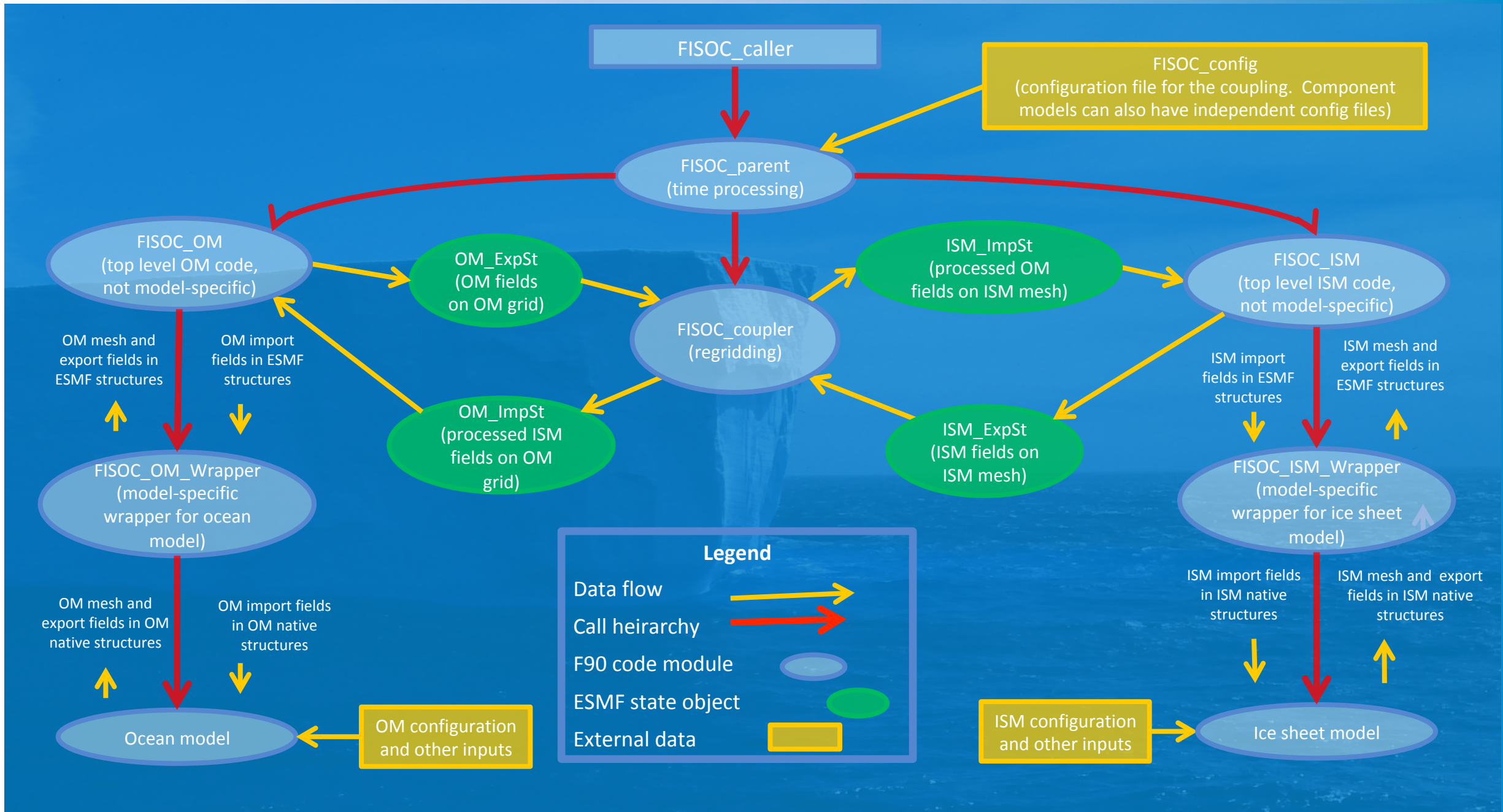
- Regridding capabilities between different types of grids/meshes
- Flexibility: relatively easy to couple in new ISMs or OMs
- Asynchronous coupling options due to differing ice and ocean timescales
- Based on the Earth System Modelling Framework (ESMF)

Participating models during the first phase:

- Regional Ocean Modelling System (ROMS)
- Elmer/Ice (Stokes and more)

FISOC Code organisation





Modifications for the build for FISOC-ESMF

Components must be compiled as libraries. This requires minimal intervention for the ROMS build and none for Elmer/Ice!

FISOC needs to link to Elmer/Ice during compilation. This is straightforward.

Summary: no build issues!

Key points re coding intervention for FISOC/ESMF:

Components must adhere to ESMF structural requirements, i.e. offer an initialise, run and finalise method. This required significant top level code restructuring for Elmer/Ice.

Methods for converting between native data types and ESMF objects must also be provided. These go in an Elmer/Ice specific wrapper within the FISOC code base.

Again, for the Elmer mesh in particular, the coding required was non-trivial.

Summary: code modification for Elmer/Ice to run with FISOC was far from trivial.

Many existing components (including many atmosphere and ocean circulation models) already have some level of ESMF compliance, making their code modifications much simpler than for Elmer/Ice.

```
!-----  
SUBROUTINE ElmerSolver(init)  
  
  INTEGER :: Init  
  
  Initialize = init  
  
  CALL ElmerSolver_init()  
  IF ( Initialize /= 1 ) THEN  
    CALL ElmerSolver_runAll()  
    CALL ElmerSolver_finalize()  
  END IF  
  
END SUBROUTINE ElmerSolver  
!-----
```

The new ElmerSolver subroutine is now simply a wrapper for the init, run and finalize methods implemented by restructuring. The original ElmerSolver code has been split between these methods.

ElmerSolver_init now contains the logic that handles the modes of initialisation.

Full code in the elmerice_FISOC branch in the main github repository. Needs tidying and merging! On my list of things to do...

RunAll wraps the existing method for running the full simulation

The Run method for FISOC is more restrictive and allows calling one timestep at a time

```
!-----  
SUBROUTINE ElmerSolver_runAll()  
  
!-----  
! Here we actually start the simulation ....  
! First go through timeintervals  
!-----  
ExecCommand = ListGetString( CurrentModel % Simulation, &  
    'Control Procedure', GotIt )  
IF ( GotIt ) THEN  
    ControlProcedure = GetProcAddr( ExecCommand )  
    CALL ExecSimulationProc( ControlProcedure, CurrentModel )  
ELSE  
    CALL ExecSimulation( TimeIntervals, CoupledMinIter, &  
        CoupledMaxIter, OutputIntervals, Transient, Scanning)  
END IF  
  
END SUBROUTINE ElmerSolver_runAll  
  
!-----  
SUBROUTINE ElmerSolver_run()  
  
CALL ExecSimulation( 1, CoupledMinIter, &  
    CoupledMaxIter, OutputIntervals, Transient, Scanning)  
  
END SUBROUTINE ElmerSolver_run
```

The Elmer “RUN” command, which effectively enables a continuation simulation at runtime is not yet implemented in the new Elmer/Ice FISOC code.

Example extract from a .sif from one of the standard Elmer/Ice tests

```
Solver 3 :: Reference Norm = Real 0.43380119E-01
Solver 3 :: Reference Norm Tolerance = Real 1E-06
$fprintf( stderr, "TEST CASE 1\n");
RUN
$fprintf( stderr, "END TEST CASE 1: Target NRM=0.43380119E-01,EPS=1E-06\n" );
```

I will tidy my code changes and implement this functionality and run all the tests before requesting to merge the `elmerice_FISOC` branch to the `elmerice` branch in the main Elmer github repository.

Ultimately these code changes should become transparent to FISOC users, and maintain precise and complete backward compatibility

Run time intervention with FISOC is minimal.

The MeltRate variable name is hard coded in FISOC (though I could easily make this an input file option). Here the lower surface variable is called FS lower (this is used in the free surface solver, so the lower surface must have a body id).

The hard coded melt rate variable must match the name of an existing variable in Elmer, such as defined using the “Exported variable” statement in the .sif.

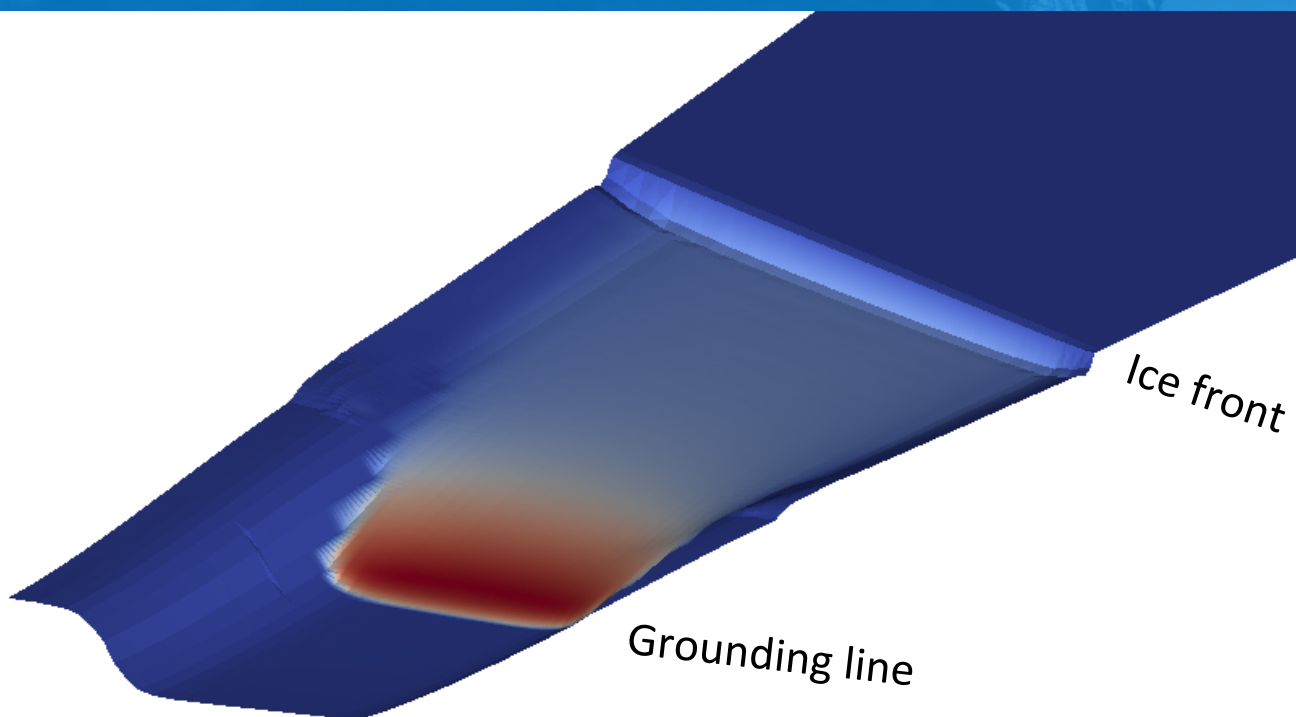
It is straightforward to exchange additional variables through FISOC, e.g. temperature.

```
Body Force 2
  FS lower Accumulation Flux 1 = Real 0.0e0
  FS lower Accumulation Flux 2 = Real 0.0e0
  FS lower Accumulation Flux 3 = Equals meltRate
End
```

```
Exported Variable 1 = FS lower Residual
Exported Variable 1 DOFS = 1
Exported Variable 2 = ReferenceFS lower
Exported Variable 2 DOFS = 1
Exported Variable 3 = meltRate
Exported Variable 3 DOFS = 1
Exported Variable 4 = height
Exported Variable 4 DOFS = 1
Exported Variable 5 = depth
Exported Variable 5 DOFS = 1
End
```

Time stepping is controlled through FISOC and the FISOC input file, which is read at run time

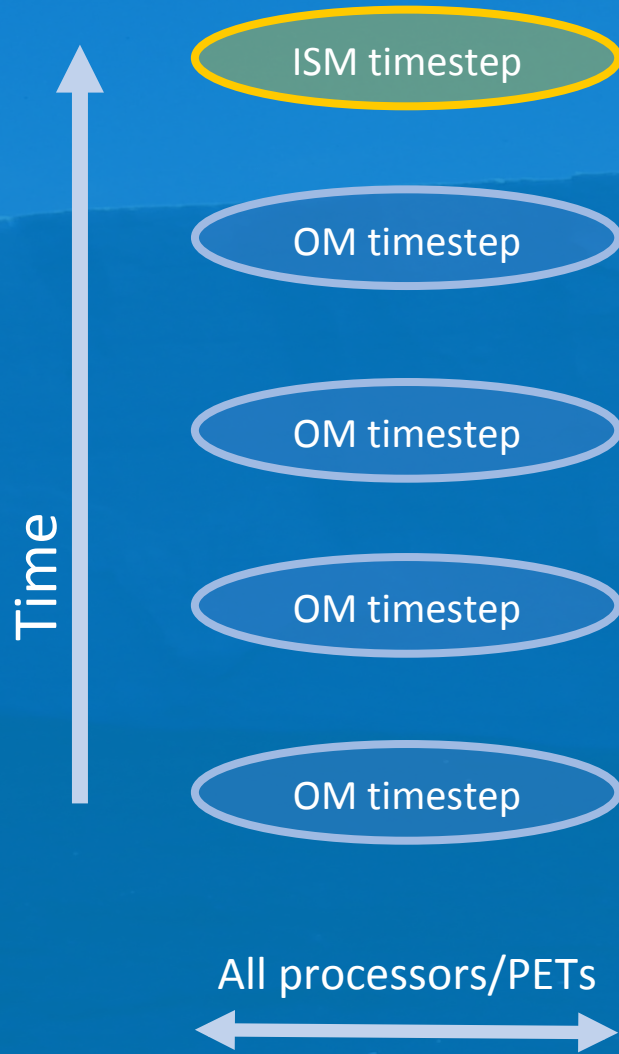
FISOC interrogates the Elmer parameters to check that FISOC and Elmer/Ice both have the same timestep size. FISOC calls Elmer/Ice one timestep at a time. Run length is set in the FISOC input file. Timestep ratio to the ocean is set similarly.



FISOC can pass either geometry change rates or actual ice model geometry to the ocean component.

Left: ice draft change rate is shown in colour, projected on to the underside of the ice shelf. This is from ISOMIP+ ocn3, using FISOC as a wrapper for time varying forcing.

Asynchronous coupling in FISOC



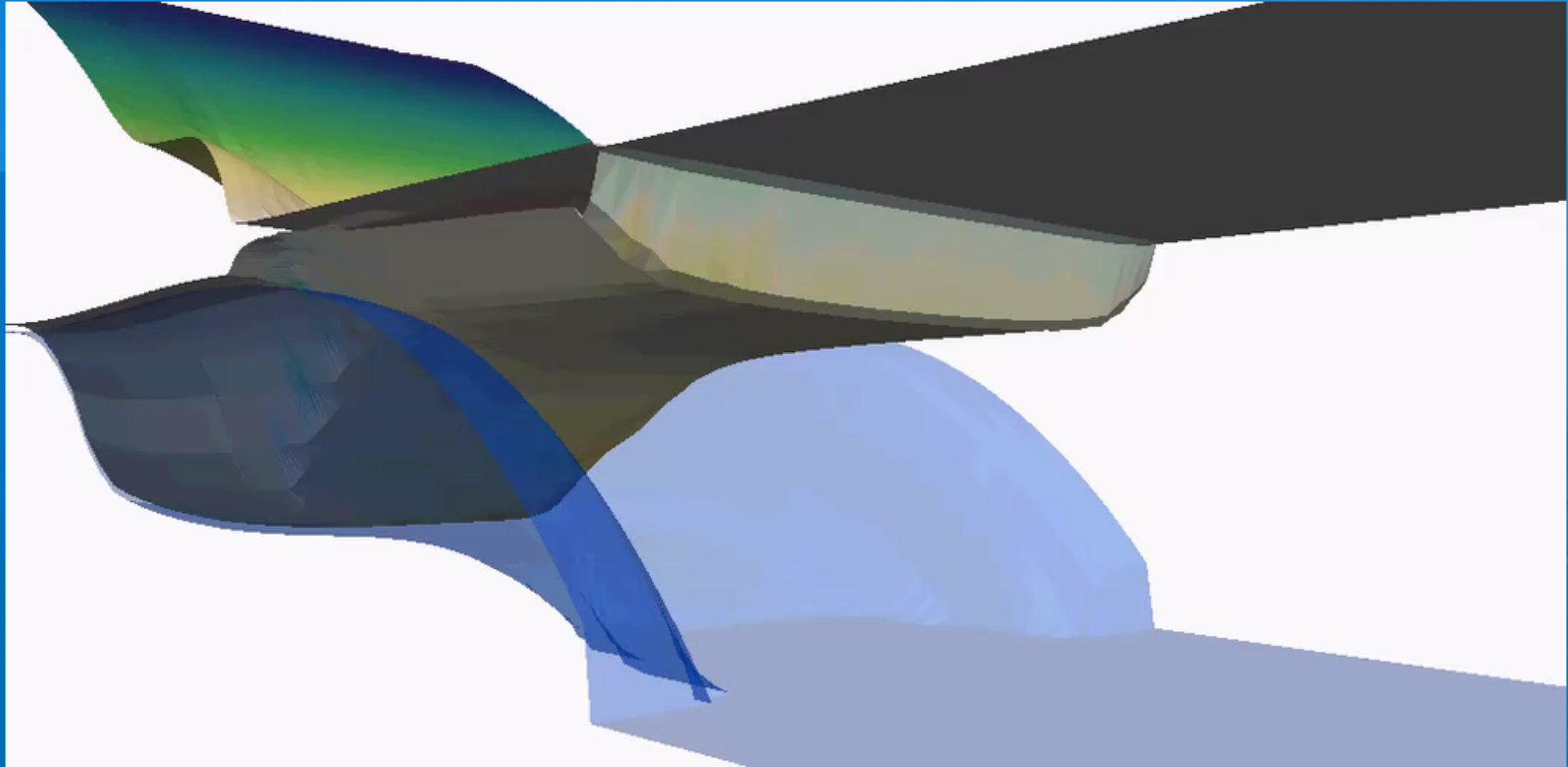
Ocean Model (OM) requires finer temporal resolution.

FISOC assumes the Ice Sheet Model (ISM) timestep is an exact multiple of the OM timestep.

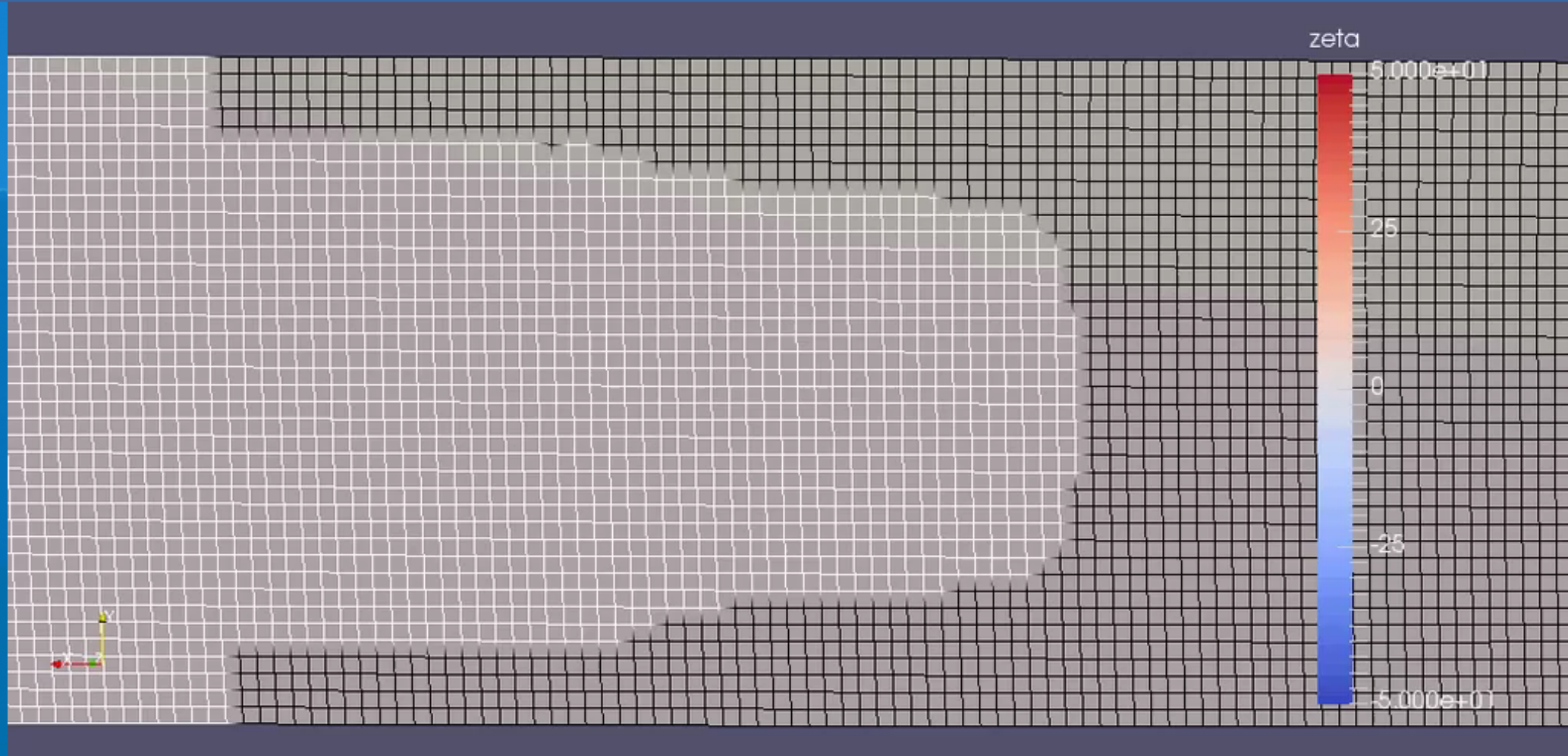
FISOC uses sequential coupling (both components run alternately on all processors).

FISOC can pass geometry rate instead of geometry snapshot to reduce shock to the OM. FISOC calculates the rate (with an optional drift correction) based on the most recent two ISM timesteps. Multiple cavity geometry coupling options are implemented in FISOC.

Paraview could potentially be used for visualising both ocean and ice model outputs together, but some manipulation is needed of ROMS netcdf files for best use within Paraview



FISOC uses the ROMS wetting and drying scheme to implement grounding line migration at run time. This from a very early simulation, with the black/white grid indicating grounded/floating cells.

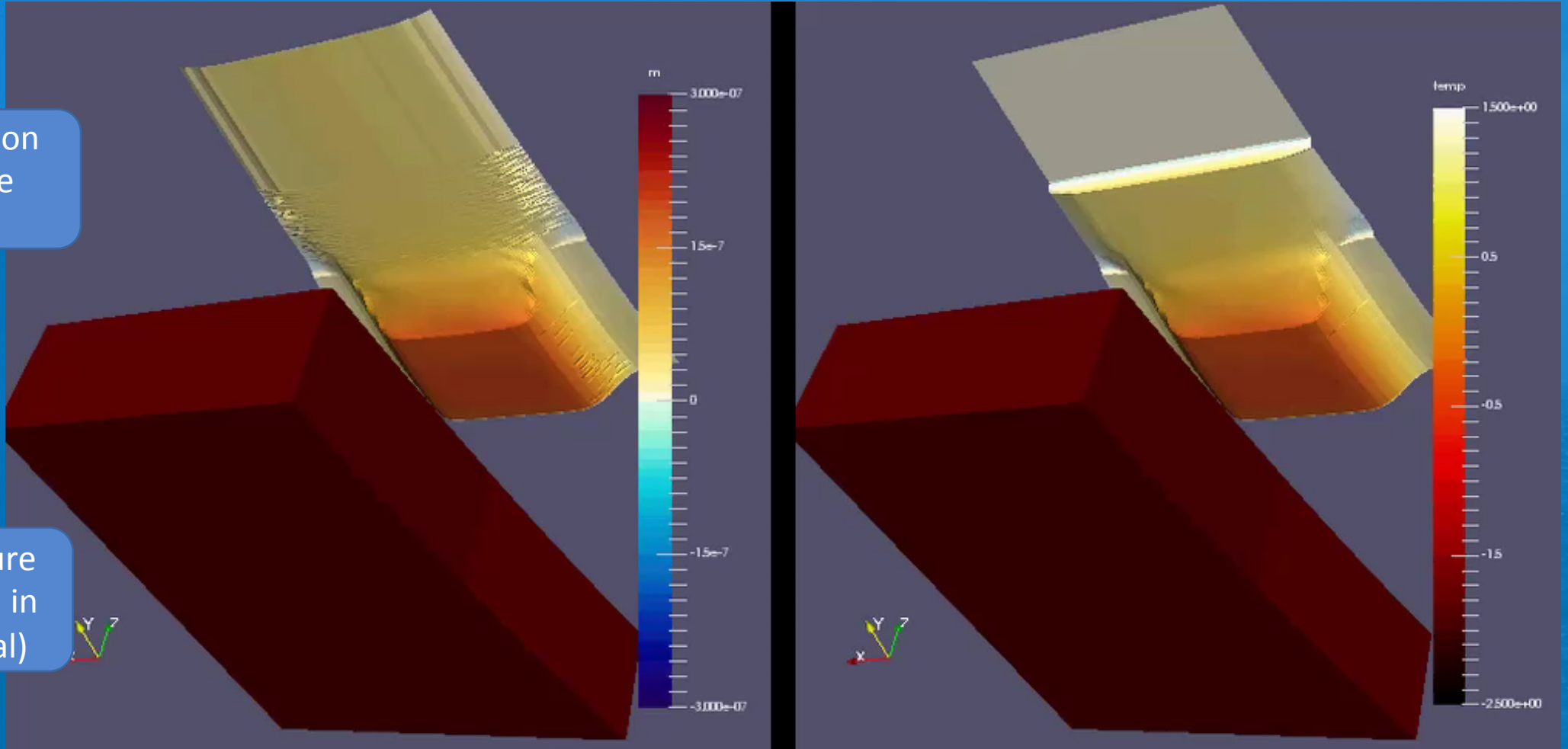


Coupled simulation (using Nacho's MISMIP+ Elmer/Ice setup)

ISOMIP+ setup using FISOC as a wrapper for time-evolving forcing)

Melt rate on lower ice surface

Temperature (not scaled in the vertical)





Power and flexibility vs practicality

FISOC has required significant development time.

There is a new top level calling program containing additional control code and utilities.

Components have undergone a moderate amount of re-engineering.

Advantages include flexibility, ease of use at run time, access to ESMF utilities especially regriding.

The main disadvantages is development time: the MISOMIP simulations with FISOC are not yet complete, though the project started earlier than Nico's Elmer-Nemo project, which I believe has completed the MISOMIP experiments.