



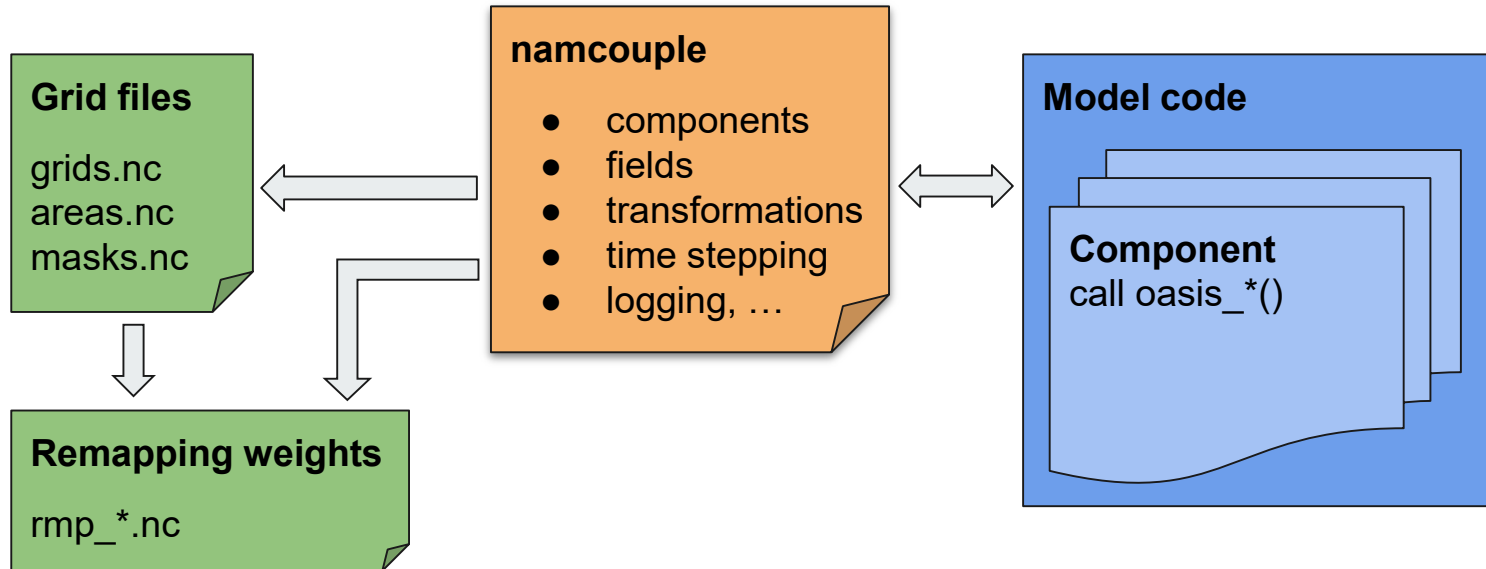
# Ready-to-couple (rdy2cpl)

Automating the Coupling Configuration for EC -Earth4

[github.com/uwefladrich/rdy2cpl](https://github.com/uwefladrich/rdy2cpl)

Uwe Fladrich (SMHI)  
Eric Maisonnave (CERFACS)

# OASIS coupling configuration





# The OASIS namcouple file

Holds configuration information about the coupled model  
(identity of components, routing of coupling fields, dimensions, time stepping, ...)

- Information is partly repeated, at different scopes, partly non-hierarchic
- Non-standard, column-based syntax; hardcoded parser
- Information (at least partly) needed outside the coupler

Alternatives:

- Different format and/or modified content
- Auto-generate namcouple file



# A namcouple data model: Why?

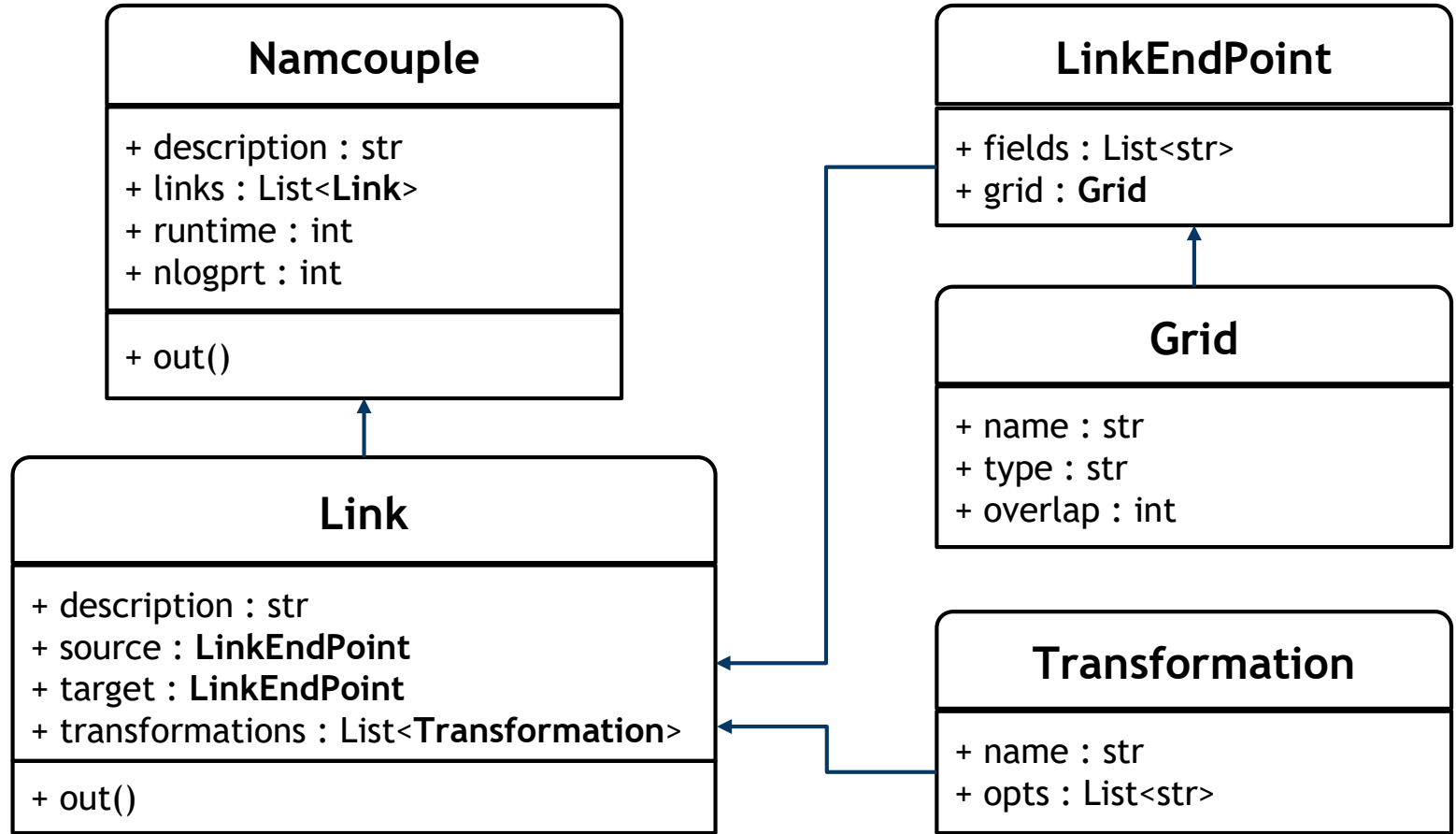
It is not only about the namcouple file format:

- Parsing the file outside OASIS
- Access configuration details
- Build tools around the coupling configuration



## A namcouple data model: How?

- Python data classes
- Concept of *links* and *link end points*
- *Grids* are just rudimentary (“real” grid info handled elsewhere)
- Namcouple objects are “printable” (i.e. namcouple file syntax build-in)



Namcouple data model in rdy2cpl (subset of actual attributes/methods)



# Use case 1

Automatic configuration and  
efficient weight computation

The problem:

namcouple, grid files, and remapping  
weights depend on model configuration

- Manual creation of grid files is prone to errors
- so is providing pre-computed weight files
- On-the-fly computation of remapping files is time consuming
- It is also wasting resources (nodes/cores)



# Auto -conf and efficient weight computation

## Solution

- Read namcouple information from (YAML) file
- Automatically create, set up and run a simplified coupled “model” (one component per distinct link / weight file)
  - Create *reduced* namcouple file
  - Create grid files (using build-in definitions of common grids)
  - Run the “model” until oasis\_end\_def → weight files produced
- Create the *real* namcouple file for the model
- Do all this fully automatic and with optimal resource allocation





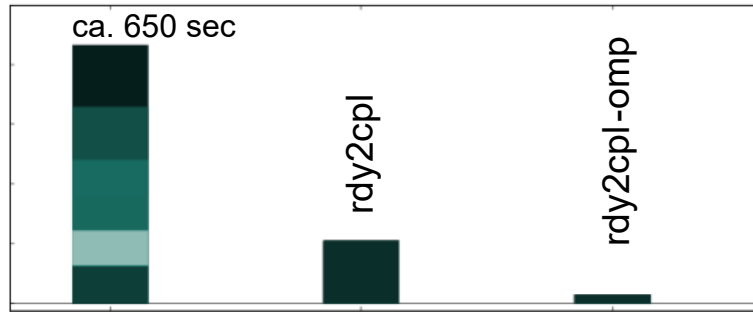
# Auto -conf and efficient weight computation

## Implementation

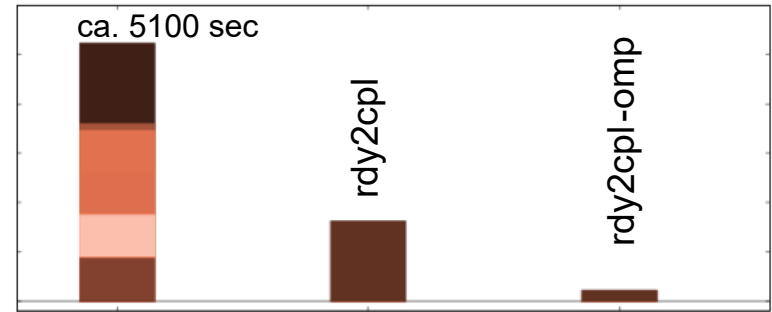
- Create namcouple object from YAML file
- Create grid files from BaseGrid classes (regular, reduced Gaussian, ORCA, etc. available)
- One OASIS component is created for each *distinct* link in the namcouple info
- Fully dynamic pyOASIS coupled “model” is launched
- One MPI process per coupling link / weight file  
*Needs removed MPI barrier in OASIS code!*
- OpenMP is used to utilise one compute node per MPI process
- Model is shot down after `oasis_end_def`
- The only user input needed is namcouple info from YAML file  
(which can be reused by other workflow steps)



## Efficient weight computation



Low-res



High-res

Elapsed times for computation of remapping weights: EGEarth standard, rdy2cp, rdy2cp+OpenMP



## Use case 2

Interpolation testing

- Help model development by testing remapping methods *on the model grids*
- OASIS examples help, but limited to certain grids/interpolations

Goals:

- Use any actual model grid and any valid remapping method
- Get immediate feedback



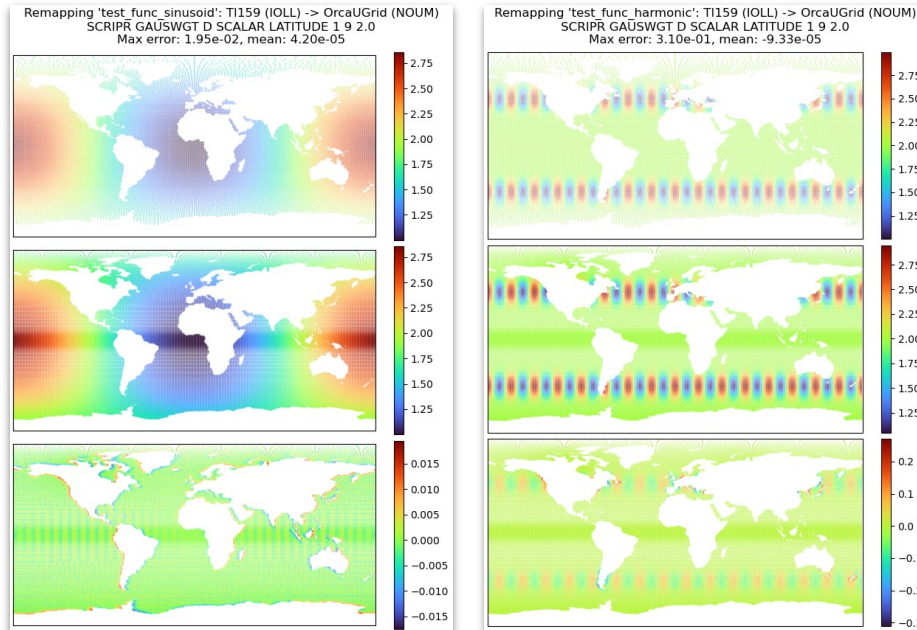
# Remapping/interpolation tests

Using the same automatic setup of a fake “model”, we

- Use test functions from [1] to set up coupling field
- Send one coupling field
- Report error
- Make plots

Note: This tool is model dependent in the sense that it “knows” the model grids.  
(Grid definitions are, however, modular and modelagnostic)

# Remapping/interpolation tests



Example remapping errors for

- sinusoidal (left) and
- harmonic (right)

test functions, using SCRIP GAUSWGT

1. Source field
2. Target field
3. Absolute error

Testing another grid/method is just a small config change ...



# Summary

This work was partly funded by the **IS-ENES3** (Grant Agreement no. 824084) and **ESIWACE2** (823988) projects under the European Union's Horizon 2020 research and innovation programme.



- OASIS namcouple data model
  - Python OO representation of the namcouple file
  - Read from (any) common file format (e.g. YAML)
  - Write in namcouple syntax
  - Provide data model to any Python tool
- Automatic weight computation
  - Flexible (any grid/method)
  - Efficient (MPI, OpenMP)
- Test regridding methods
  - Flexible (any grid/method)
  - Uses exact model configuration
  - Immediate developer feedback, no model run required