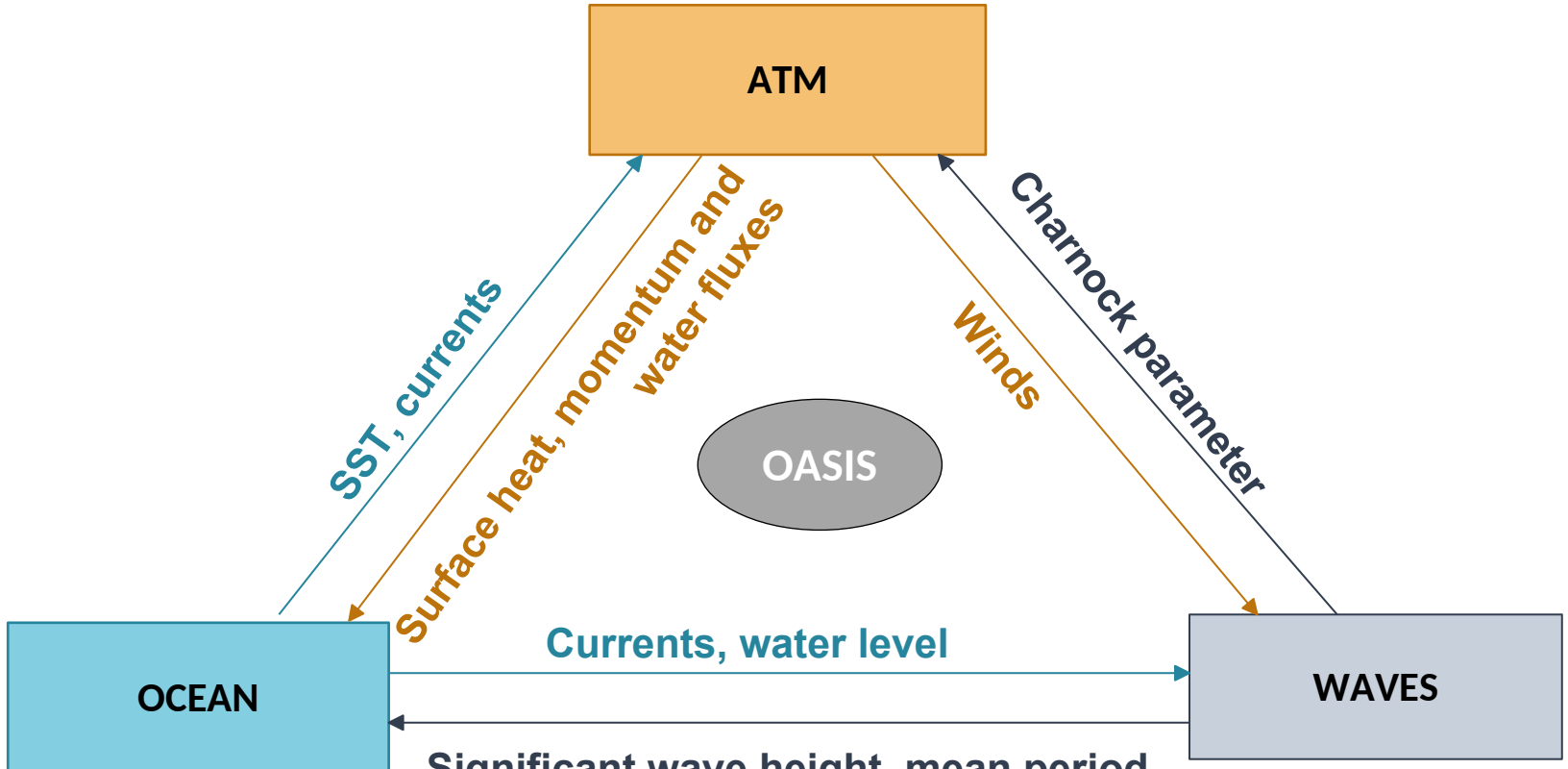
Coupling CROCO with other models using OASIS

Swen Jullien, Gildas Cambon, Mathieu Le Corre, Lionel Renault

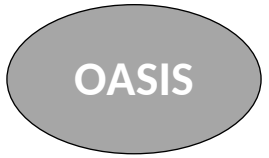
Coupling philosophy



Coupling philosophy

The OASIS-MCT coupler

OASIS-MCT (Ocean-Atmosphere-Sea-Ice-Soil, Model Coupling Toolkit) is a coupler developed at CERFACS, Toulouse, France.



It is a **set of libraries** (not an executable file) providing functions which are called in the models themselves:

- Exchange of variables and time interpolations (PSMILE library)
- Parallel exchanges (MCT library)
- Grid interpolations (SCRIPR library)

It has the **advantage** of being:

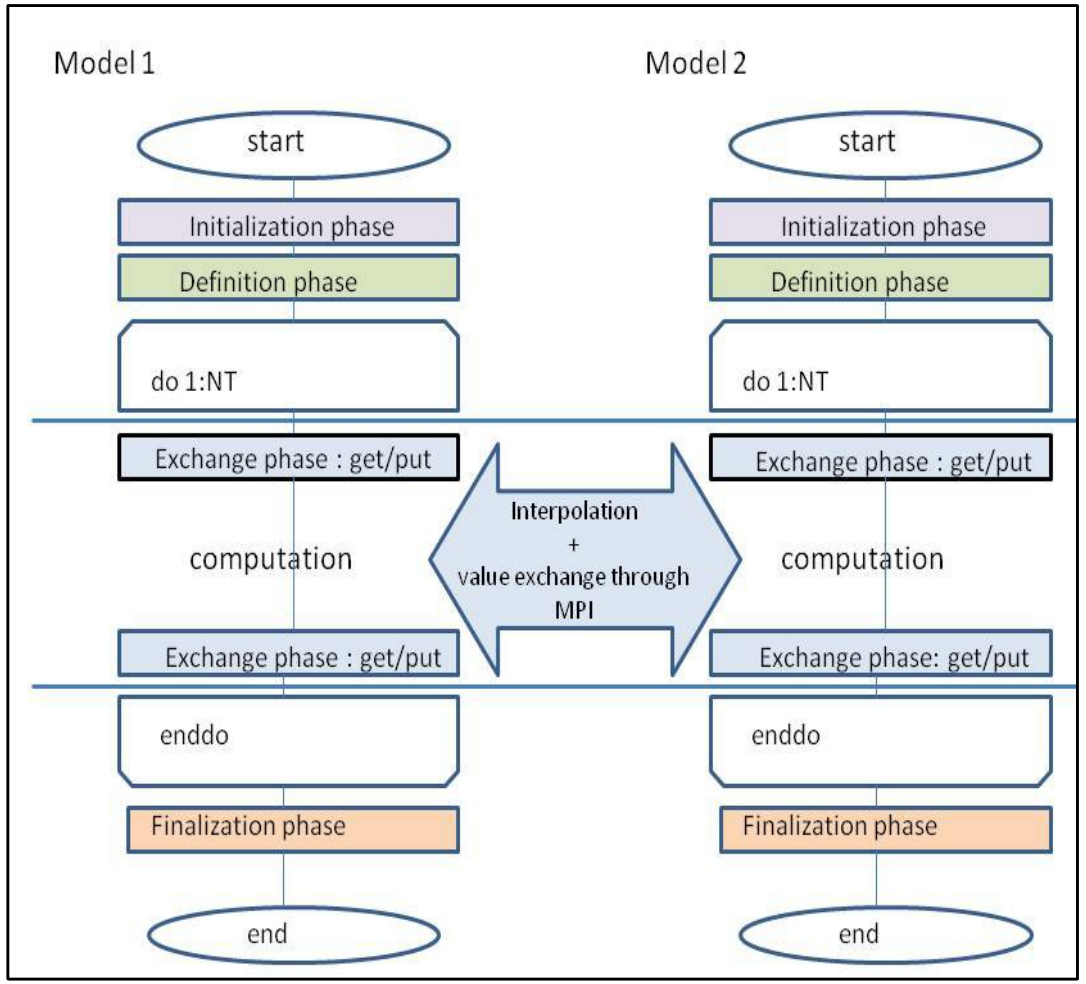
- non-intrusive, easy implementation: only a few calls in the model time stepping, and a few additional routines
- A common interface for a variety of models (e.g. CROCO, NEMO, SURFEX, WW3…)

Coupling philosophy

OASIS-MCT implementation in models

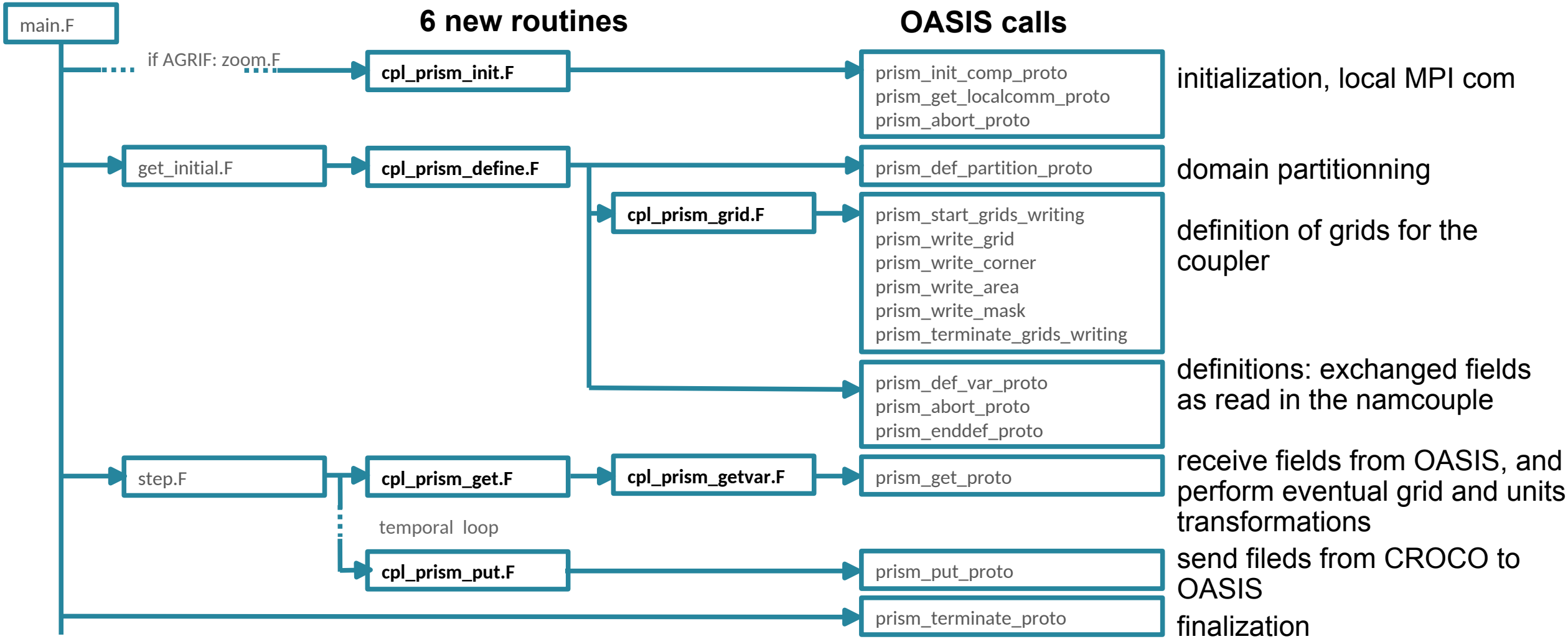
OASIS-MCT implementation calls:

- **Initialization**
  - oasis_init_comp(…)
  - oasis_get_localcomm(..)
- **Definitions**
  - oasis_write_grid(…)
  - oasis_def_partitions(…)
  - oasis_def_var(…)
- **Exchange fields** (witin time stepping)
  - oasis_put(…)
  - oasis_get(…)
- **Finalization**
  - oasis_terminate(…)



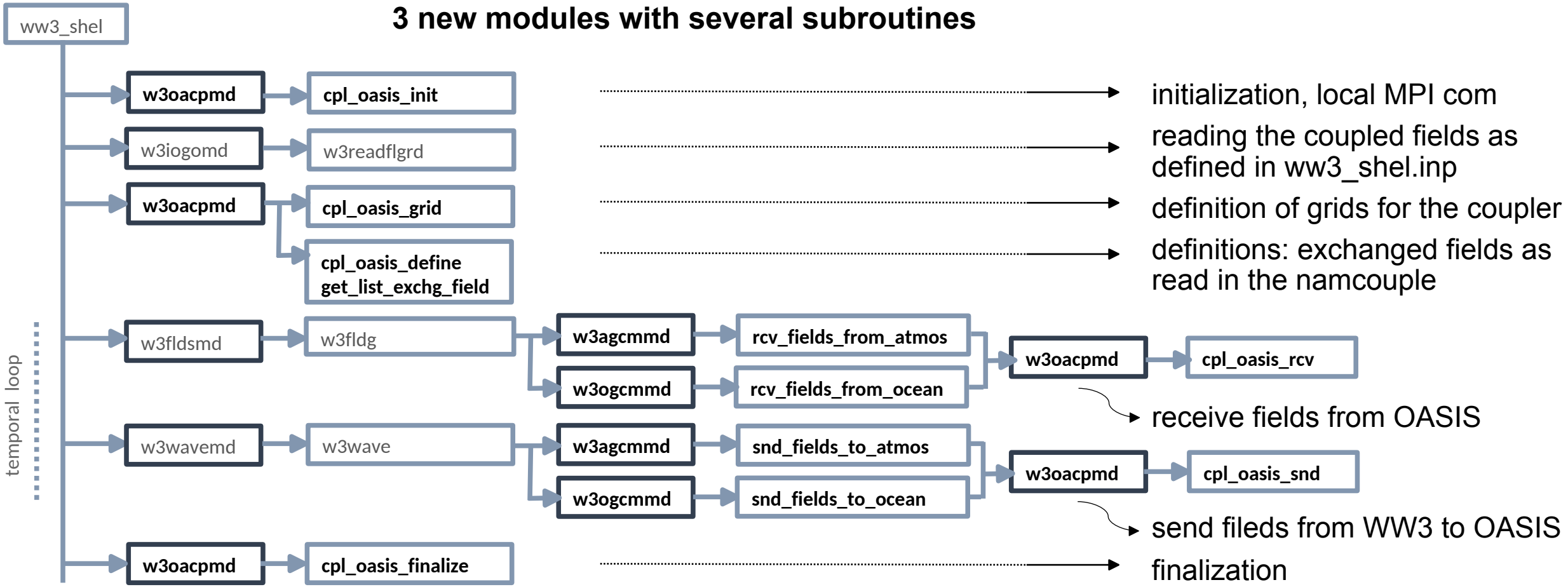
Detailed OASIS implementation

In CROCO



Detailed OASIS implementation

In WW3

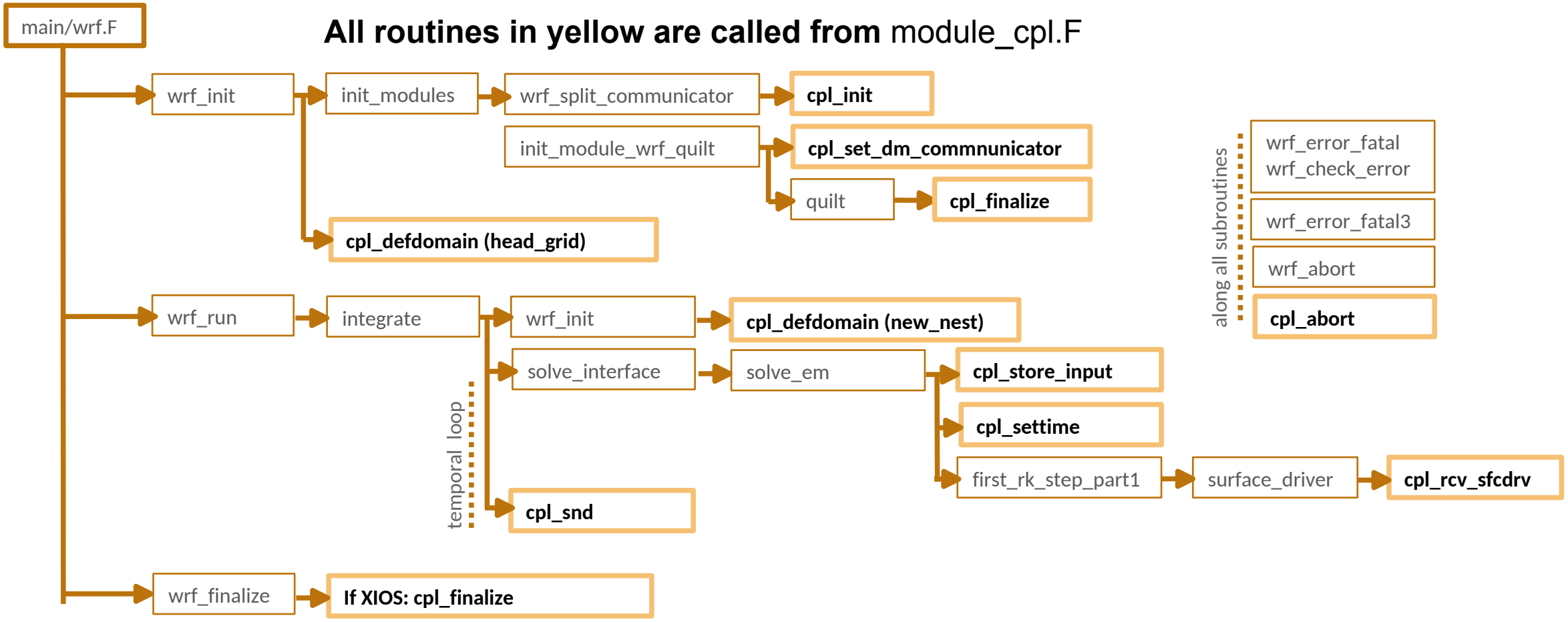


Detailed OASIS implementation

In WRF

2 new modules: frame/module_cpl.F **and** frame/module_cpl_oasis3.F

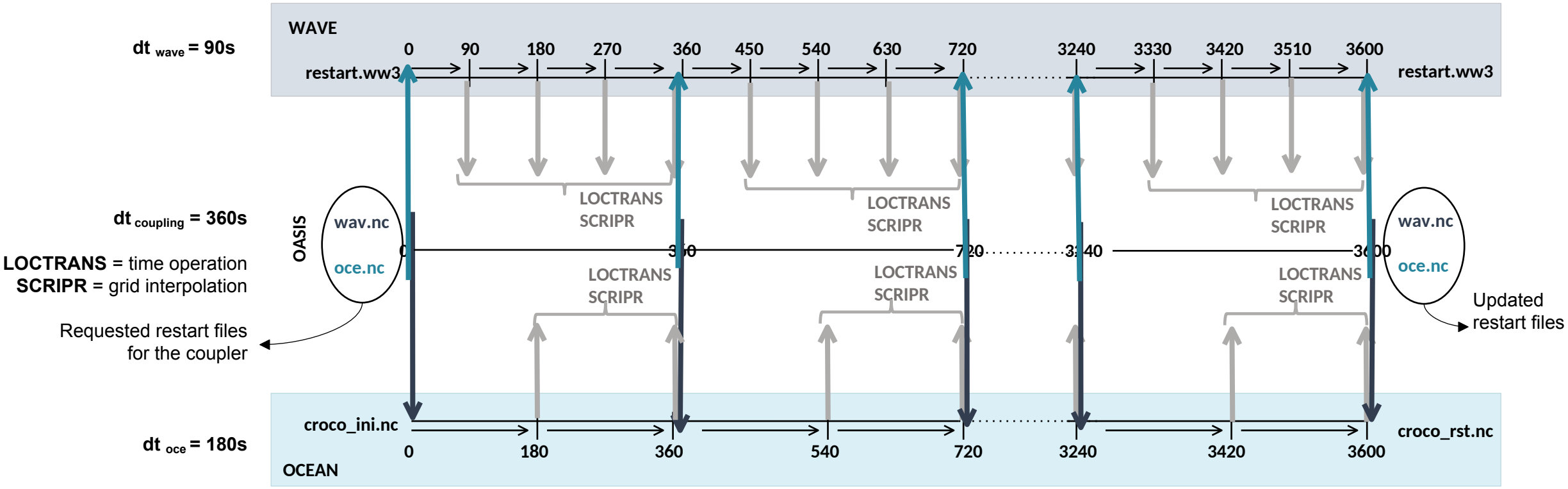
All routines in yellow are called from module_cpl.F



Coupling sequence

Example

Exchange phase is called every time step but the effective exchanges are only performed at the defined coupling frequency



Transformations performed by OASIS-MCT

The OASIS3-MCT coupler can process:

- time transformations (**LOCTRANS**):
  - INSTANT no time transformation, the instantaneous field is transferred
  - ACCUMUL the accumulated field over the coupling period is exchanged
  - AVERAGE the averaged field over the coupling period is transferred
  - T_MIN the minimum value of the field for each source grid point over the coupling period is transferred
  - T_MAX the maximum value of the field for each source grid point over the coupling period is transferred

- 2D spatial interpolations (**SCRIPR**):
  - BILINEAR interpolation based on a local bilinear approximation
  - BICUBIC interpolation based on a local bicubic approximation
  - CONSERV 1st or 2nd order conservative remapping
  - DISTWGT distance weighted nearest-neighbour interpolation (N neighbours)
  - GAUSWGT N nearest-neighbour interpolation weighted by their distance and a Gaussian function

See OASIS manual for more detailed information

OASIS-MCT namelist file

namcouple

The **namcouple** is the namelist file through which you will specify which fields will be coupled.

A first section gathers general settings:

```
# NFIELDS: total number of fields being exchanged
 $NFIELDS
 23
#############################################################
# NBMODEL: number of models and their names (6 characters)
 $NBMODEL
 3  wrfexe  wwatch  crocox
#############################################################
# RUNTIME: total simulated time for the actual run in seconds (<I8)
 $RUNTIME
 86400
#############################################################
# NLOGPRT: debug and time statistics informations printed in log file
$NLOGPRT
 1 1
```

OASIS-MCT namelist file

namcouple

The **namcouple** is the namelist file through which you will specify which fields will be coupled.

A second section provides the information on exchanged fields.
A typical sub-section for one exchanged field looks like:

```
CROCO_SSH WW3__SSH 1 360 1 oce.nc EXPORTED
318 248 318 248 ocnt ww3t LAG=180
R  0  R  0
SCRIPR
DISTWGT LR SCALAR LATLON 1 4
```

Line 1: OASIS name for field in sending model
OASIS name for field in sending model in target model
Unused digit
Coupling period
#of transformations (here 1 interpolation)
Restart file name
Keyword: EXPORTED = field written only at the end of the simulation
EXPOUT = field written at every coupling time step in restart file

Line 2: # of points of the sending and target grids, names of the sending and target grids, LAG=dt of sending model
Line 3: type of grid (**P**eriodical or **R**egional) and # of overlapping points for sending and target models
Line 4: keywords for transformations to perform
Line 5: Parameters for each transformation

Þ A basis of namcouple files can be found in the croco/SCRIPTS/SCRIPTS_COUPLING/OASIS_IN directory.

OASIS-MCT additional files

- Input files:
  - **Restart files**: they have to be build for initialization, and they will be automatically written at the end of the simulation for the next one: oce.nc, atm.nc, wav.nc
  - **Namelist** file: namcouple

- **Grid generated files**: these files are requested for interpolations, they are automatically built at the beginning of the simulation by all models: grids.nc, masks.nc, areas.nc
- **Grid interpolation generated files**: these files are built automatically by OASIS according to the previously cited grid files, and to SCRIPR settings in the namcouple: rmp_ww3t_to_ocnt_DISTWGT.nc, rmp_ocnt_to_ww3t_DISTWGT.nc …

- **Log files**: several log files are produced by OASIS, they should be checked at the end of the simulation or if something goes wrong during the simulation:
  - nout.000000 : OASIS log file
  - crocox.timers_0000, wwatch.timers_0000 : OASIS log file for time statistics
  - debug.root.01, debug.root.02 : log files for the master processor for each model
  - debug.notroot.01 : log files for other processors for each model

Steps to setup a coupled simulation

Summary

(1) Get the source codes

(2) Set-up your configuration architecture and environment. You can use: croco/create_config.bash (all-prod-cpl option)

(3) Compile:
- First compile OASIS
- Then, compile your models in coupled mode (with the same netcdf libraries and compilers)

NB: CROCO can alternatively be compiled automatically when launching the run within the **coupling toolbox**

(4) Perform pre-processing for your different models

(5) Define the namelists and input files:
- OASIS namcouple
- Models' namelists: croco.in, ww3_grid.inp, ww3_shel.inp, namelist.input
- Create restart files for the coupler

(6) Running: launch the models simultaneously, e.g.: mpirun -np 4 wwatch : -np 4 crocox

(7) Outputs: check log and ouptut files

Use can alternatively **use the coupling toolbox** to perform steps 5-6 more easily

Þ If you have problems during your coupled run, check the dimensions of the grids in all grid files (models and OASIS grids and masks files)

Steps to setup a coupled simulation

(1) Get the source codes

Suggested work architecture:

mkdir $HOME/croco
mkdir $HOME/oasis
mkdir $HOME/wrf
mkdir $HOME/ww3
mkdir $HOME/CONFIGS

CROCO
cd $HOME/croco

https://www.croco-ocean.org/croco-project/
tar -zxvf croco-v1.3.tar.gz
tar -zxvf croco_tools-v1.3.tar.gz

Or

git clone https://gitlab.inria.fr/croco-ocean/croco.git
git clone https://gitlab.inria.fr/croco-ocean/croco_tools.git

OASIS
cd $HOME/oasis

git clone https://gitlab.com/cerfacs/oasis3-mct.git

WRF
cd $HOME/wrf

git clone https://github.com/wrf-croco/WRF.git

WPS
git clone https://github.com/wrf-model/WPS.git
git checkout tags/v4.2

WW3
cd $HOME/ww3

git clone https://github.com/NOAA-EMC/WW3

Steps to setup a coupled simulation

(2) Create environment and architecture

You can use the provided script: **croco/create_config.bash** (all-prod-cpl option)

Þ It will create your configuration directory with all the useful scripts

```
cd $HOME/CONFIGS
cp $HOME/croco/croco/create_config.bash .

Edit create_config.bash : paths, settings
./create_config.bash

You should now have YOUR_CONFIG directory
cd $HOME/CONFIGS/ YOUR_CONFIG

Check the myenv_mypath.sh file, eventually edit path if necessary
source myenv_mypath.sh
```

```
# Machine you are working on
# Known machines: Linux DATARMOR IRENE JEANZAY
# -------------------------------------------
MACHINE=”DATARMOR”
# croco source directory
# ---------------------
CROCO_DIR=$HOME/croco/croco
# croco_tools directory
# ---------------------
TOOLS_DIR=$HOME/croco/croco_tools
# Configuration name
# ------------------
MY_CONFIG_NAME=BENGUELA
# Home and Work configuration directories
# ---------------------------------------
MY_CONFIG_HOME=$HOME/CONFIGS
MY_CONFIG_WORK=$SCRATCH/CONFIGS
# Options of your configuration
# -----------------------------
## example for production run architecture and coupling with
external models:
options=( all-prod-cpl )
```

Steps to setup a coupled simulation

(3) Compile

OASIS

cd $HOME/oasis/oasis3-mct/oasis3-mct/util/make_dir
cp $HOME/croco/croco/SCRIPTS/SCRIPTS_COUPLING/OASIS_IN/make.DATARMOR .

Check the paths in make.DATARMOR
In make.inc set the include to the absolute path of your make.DATARMOR:
include $(home)/oasis/oasis3-mct/util/make_dir/make.DATARMOR

Compilation:
make realclean -f TopMakefileOasis3 > oasis_clean.out
make -f TopMakefileOasis3 > oasis_make.out

Steps to setup a coupled simulation

(3) Compile

CROCO

OPTION 1 : “A la mano”

cd $HOME/CONFIGS/YOUR_CONFIG/CROCO_IN

Check and edit :
param.h => grid and MPI settings
cppdefs.h => CPP options **OA_COUPLING or/and OW_COUPLING, and MPI are mandatory**
jobcomp.h => paths, compilers, and libraries

Compile
./jobcomp

Eventually move the executable to another name :
mv croco croco.frc
or
mv croco croco.ow

OPTION 2 : automatic compilation when running

In $HOME/CONFIGS/YOUR_CONFIG/mynamelist.sh

You can set online compilation of CROCO:

Online Compilation
export **ONLINE_COMP=**1

NB! CROCO has to be re-compiled each time the grid or the MPI settings are changed

Steps to setup a coupled simulation

(3) Compile

WRF

OPTION 1 : “A la mano”

```
cd $HOME/wrf/WRF

Classical compilation :
./clean –a
./configure
Check and eventually edit
configure.wrf, then:
./compile em_real
```

OPTION 2 : “A la mano” with some help

```
cd $HOME/wrf/WRF

A few scripts are provided to help you compile : First
set a few environment variables:
cat
$HOME/croco/croco/SCRIPTS/SCRIPTS_COUPLIN
G/SCRIPTS_TOOLBOX/MACHINE/DATARMOR/mye
nv.DATARMOR* myenv.sh
source myenv.sh

Then copy the appropriate configure files :
cp
$HOME/croco/croco/SCRIPTS/SCRIPTS_COUPLIN
G/WRF_IN/CONFIGURE_WRF/DATARMOR/* .

./clean –a
cp configure.wrf.uncoupled configure.wrf
./compile em_real > compile_uncoupled.log

./clean -a
cp configure.wrf.coupled configure.wrf
./compile em_real > compile_coupled.log
```

OPTION 3: using provided scripts

```
cd $HOME/CONFIGS/YOUR_CONFIG

Check your paths in myenv_mypath.sh

cd WRF_IN
qsub DATARMOR.compile.wrf.pbs
```

Steps to setup a coupled simulation

(3) Compile

WPS

OPTION 1 : “A la mano”

```
cd $HOME/wrf/WPS

Classical compilation :
./clean –a
./configure
Check and eventually edit
configure.wps, then:
./compile
```

OPTION 2 : “A la mano” with some help

```
cd $HOME/wrf/WPS

A few scripts are provided to help you compile :
First set a few environment variables:
cat
$HOME/croco/croco/SCRIPTS/SCRIPTS_COUPLI
NG/SCRIPTS_TOOLBOX/MACHINE/DATARMOR/
myenv.DATARMOR* myenv.sh
source myenv.sh

Then copy the appropriate configure files :
cp
$HOME/croco/croco_tools/Coupling_tools/WRF_W
PS/configure.wps.MACHINE .

./clean –a
cp configure.wrf.MACHINE configure.wps
./compile > compile_wps.log
```

Steps to setup a coupled simulation

(3) Compile

WW3

OPTION 1 : “A la mano”

cd $HOME/ww3/model/bin

Check that the $OASISDIR variable correctly refers to your OASIS compile directory
WW3 compilation requests 3 files:

- switch file: examples for coupling with ocean / atmosphere : **switch_OASOCM** / **_OASACM**. Mandatory switches for coupling are: **MPI DIST COU OASIS OASOCM OASACM** and **CRT0 WNT0**
- comp.COMPILER file, e.g. for Intel comp.Intel
- link.COMPILER file : provide useful options and links for compilers

./w3_clean -c
./w3_setup .. -c Intel -s OASACM
./w3_automake
or only useful executables e.g.:
./w3_make ww3_prnc ww3_grid ww3_bounc ww3_strt ww3_shel ww3_ounf

Þ If compilation is successful, executables are in ../exe, they should be moved:
mkdir ../exe_OASACM ; mv ../exe/* ../exe_UNCOUPLED/.

OPTION 2: using provided script

A script to help you compile the various mode is also available:
make_WW3_compil

cd $HOME/CONFIGS/YOUR_CONFIG

Check your paths in myenv_mypath.sh

cd WW3_IN

Edit options in make_WW3_compil
./make_WW3_compil

Steps to setup a coupled simulation

(3) Compile

Tips in case of errors during compilation

In case of strange errors during compilation (e.g. “catastrophic error: could not find …”), try one of these solutions:

- check your home space is not full ;-)
- check your paths to compilers and libraries (especially Netcdf library)
- check that you have the good permissions, and check that your executable files (configure, make…) do are executable
- check that your shell scripts headers are correct or add them if necessary (e.g. for bash: #!/bin/bash)
- try to exit/log out the machine, log in back, clean and restart compilation

In case of ‘segmentation fault’ error:

- try to allocate more memory with “unlimited -s unlimited”
- try to launch the compilation as a job (batch) with more allocated memory

Errors and tips related to netcdf library:

- with netcdf 4.3.3.1: need to add the following compilation flag for all models: -mt_mpi
  the error associated to a missing -mt_mpi flag is of this type: ” /opt/intel//impi/4.1.1.036/intel64/lib/libmpi_mt.so.4: could not read symbols: Bad value “
- with netcdf 4.1.3: do NOT add -mt_mpi flag
- with netcdf4, need to place hdf5 library path in your environment: export LD_LIBRARY_PATH=YOUR_HDF5_DIR/lib:$LD_LIBRARY_PATH
- with netcdf 4, if you use the library splitted in 2: C part and Fortran part, you need to place links to C library before links to Fortran library and need to put both path in this same order in your LD_LIBRARY_PATH

Steps to setup a coupled simulation

(4) Preprocessing

CROCO

cd $HOME/CONFIGS/YOUR_CONFIG/PREPRO/CROCO

Check and edit paths in start.m
Check and edit settings and paths in crocotools_param.m
In matlab for climatological input files :
start
make_grid
make_forcing
make_bry
make_ini

Other example for runnning interannual with clim forcing :

in crocotools_param uncomment:
coads_time=(15.2188:30.4375:350.0313);
coads_cycle=365.25;

start
make_grid
make_forcing
make_QSCAT_clim
make_bry
make_ini

cd $DATAWORK/CONFIGS/YOUR_CONFIG//CROCO_FILES
for year in 2017 2018 ; do for month in 01 02 03 04 05 06 07 08 09 10 11 12 ;
do ln -s croco_frc.nc croco_frc_Y${year}M${month}.nc ; done ; done

for year in 2017 2018 ; do for month in 01 02 03 04 05 06 07 08 09 10 11 12 ;
do ln -s croco_bry.nc croco_bry_Y${year}M${month}.nc ; done ; done

ln -s croco_ini.nc croco_ini_Y2017M01.nc

WRF

cd $HOME/CONFIGS/YOUR_CONFIG/PREPRO/WRF_WPS

Rename and edit configure.namelist.wps_BENGUELA for your own configuration
Check and edit settings and paths in run_wps.bash
Check CPUs in job.wps.pbs and launch it: qsub job.wps.pbs

WW3

Follow WW3 preprocessing tutorial

Or eventually create grid files from croco grid with:
make_ww3_grd_input_files_from_croco_grd.m

Steps to setup a coupled simulation

Summary

(1) Get the source codes

(2) Set-up your configuration architecture and environment. You can use: croco/create_config.bash (all-prod-cpl option)

(3) Compile:
- First compile OASIS
- Then, compile your models in coupled mode (with the same netcdf libraries and compilers)

NB: CROCO can alternatively be compiled automatically when launching the run within the **coupling toolbox**

(4) Perform pre-processing for your different models

(5) Define the namelists and input files:
- OASIS namcouple
- Models' namelists: croco.in, ww3_grid.inp, ww3_shel.inp, namelist.input
- Create restart files for the coupler

(6) Running: launch the models simultaneously, e.g.: mpirun -np 4 wwatch : -np 4 crocox

(7) Outputs: check log and ouptut files

Use can alternatively **use the coupling toolbox** to perform steps 5-6 more easily

Þ If you have problems during your coupled run, check the dimensions of the grids in all grid files (models and OASIS grids and masks files)

Steps to setup a coupled simulation

(5-6) “A la mano”

Steps 5 and 6 : 2 options: « A la mano » or using the coupling toolbox

First option for preparing the run (namelist and launch) : **« A la mano »**

Steps to setup a coupled simulation

(5) Define namelists and input files

You need to prepare restart files for OASIS:

cd $HOME/CONFIGS/YOUR_CONFIG/

Copy useful script: **create_oasis_restart_from_calm_conditions.sh** provided in $HOME/croco/croco/SCRIPTS/SCRIPTS_COUPLING/SCRIPTS_TOOLBOX/ROUTINES/OASIS_SCRIPTS/

Run it for each model with the following arguments:
grid name, restart file name, type of model, list of variables to initialize to 0

Examples :

./create_oasis_restart_from_calm_conditions.sh CROCO_FILES/croco_grd.nc rst_oce.nc croco ”CROCO_SST CROCO_SSH CROCO_NOCE CROCO_EOCE"

./create_oasis_restart_from_calm_conditions.sh WW3_FILES/ww3_201701.nc rst_wav.nc ww3 ” WW3_T0M1 WW3__OHS WW3__DIR WW3_ACHA WW3_TAWX WW3_TAWY WW3_TWOX WW3_TWOY"

./create_oasis_restart_from_calm_conditions.sh WRF_FILES/wrfinput_d01 rst_atm.nc wrf “WRF_d01_EXT_d01_SURF_NET_SOLAR WRF_d01_EXT_d01_EVAP-PRECIP WRF_d01_EXT_d01_SURF_NET_NON-SOLAR WRF_d01_EXT_d01_TAUE WRF_d01_EXT_d01_TAUN WRF_d01_EXT_d01_TAUMOD WRF_d01_EXT_d01_PSFC WRF_d01_EXT_d01_WND_E_01 WRF_d01_EXT_d01_WND_N_01”

Steps to setup a coupled simulation

(5) Define namelists and input files

OASIS
Namelist: namcouple: models, coupled variables…
Input files:rst_oce.nc
rst_atm.nc
rst_wav.nc

CROCO
Namelist: croco.in : **time steps**, input files paths, parameters

Input files:croco_grd.nc
croco_frc.nc
croco_ini.nc or croco_rst.nc
croco_bry.nc or croco_clm.nc

WRF
Namelist: namelist.input : dates, time step, schemes, param…

Input files:wrfinput_d01
wrflowinp_d01
wrfbdy_d01

WW3
Namelists: ww3_grid.inp : **grid** settings, namelist/param options, time steps
ww3_shel.inp : coupled flags, time steps, and fields

```
C F   Water levels
C F   Currents
C F   Winds

$ Type 7 : Coupled fields
20090101 000000  360  20090201 000000
N
T0M1 OHS THM TAW TWO ACHA
SSH CUR WND
```

Input files: mod_def.ww3 (generated by ww3_grid)
restart.ww3 (generated by ww3_strt or previous run)
nest.ww3 (generated by ww3_bounc)
wind.ww3, current.ww3, ice.ww3 : if forcing needed (generated by ww3_prnc)

Steps to setup a coupled simulation

(6) Run

mpirun -np 10 wrfexe : -np 6 wwatch: -np 4 crocox

Steps to setup a coupled simulation

(5-6) Using the coupling toolbox

Steps 5 and 6 : 2 options: « A la mano » or using the coupling toolbox

Second option for preparing the run (namelist and launch) : **with the coupling toolbox**

Steps to setup a coupled simulation

(5-6) Define and run using the coupling toolbox

Coupling toolbox philosophy and workflow:

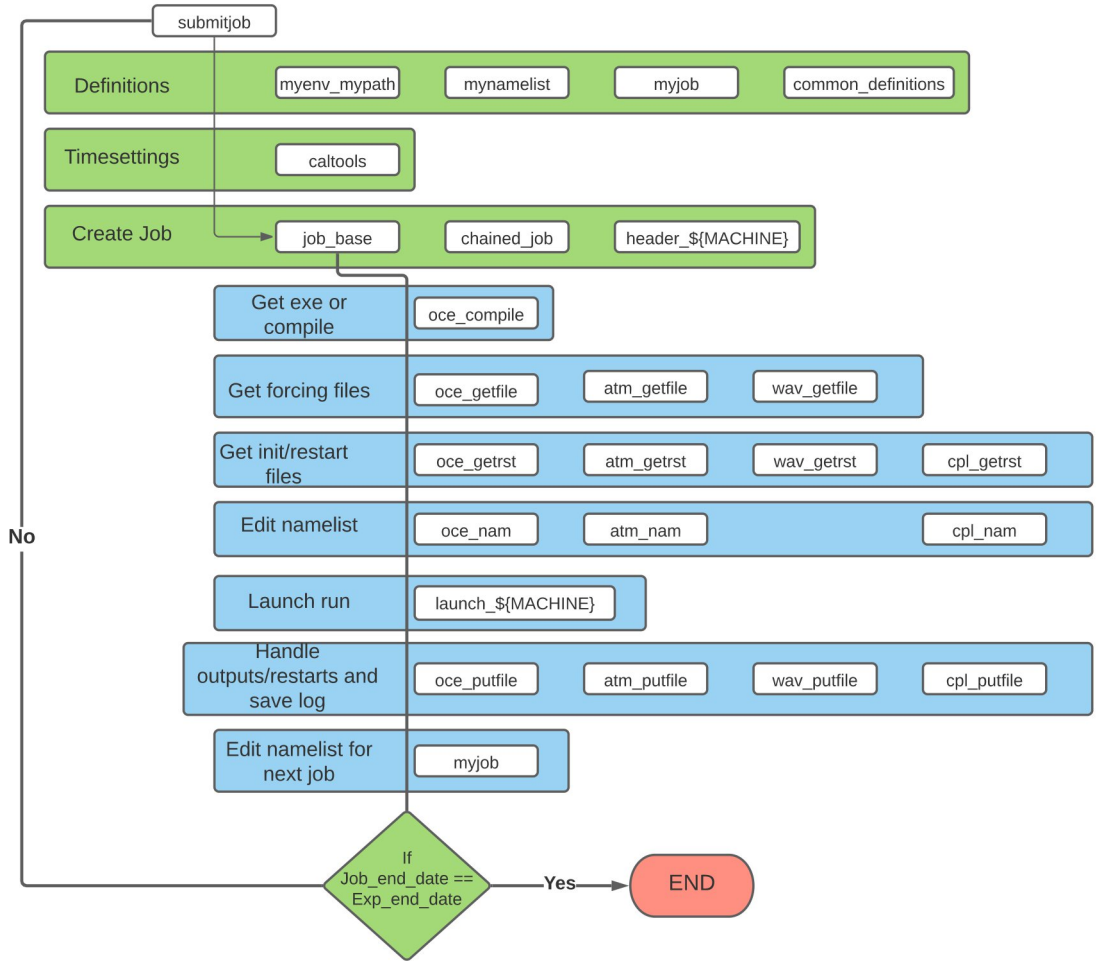
The user edit:

* **myenv_mypath.sh** : environment settings, and paths
* **mynamelist.sh** : settings for the experiment (which models, time stepping, input files…)
* **myjob.sh** : settings for the job (dates notably)

Then the user launch the job with **./submitjob.sh**

<u>The coupling toolbox manages:</u>

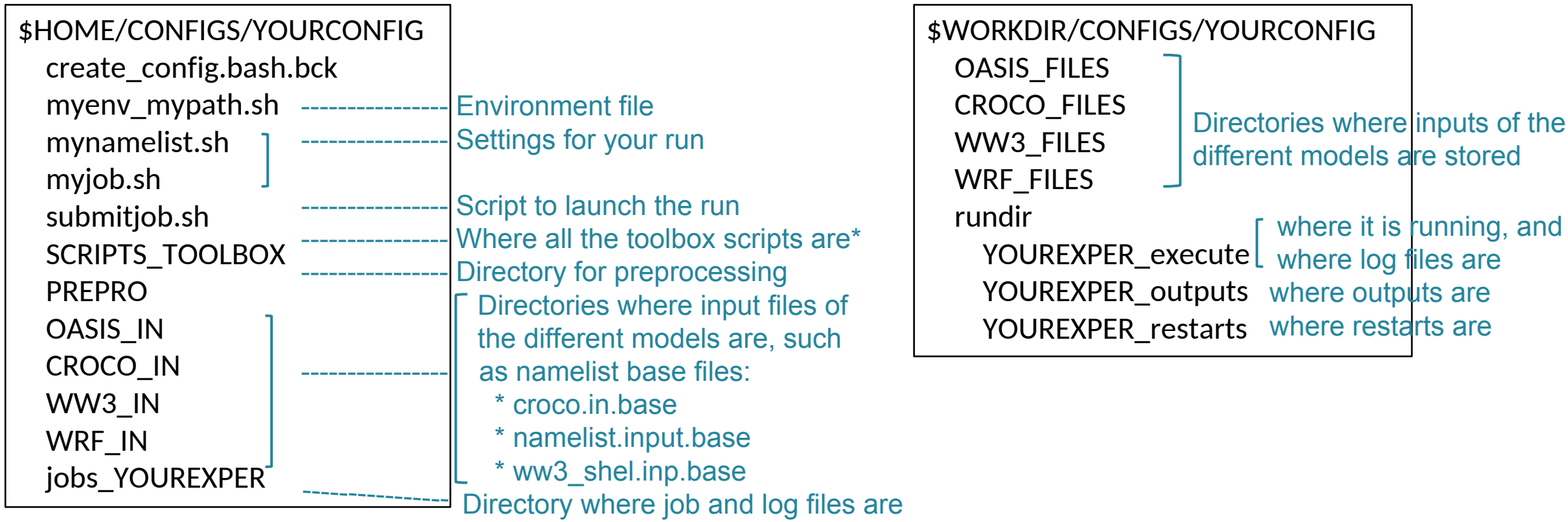
- CROCO compilation if requested
- getting models input files
- preparing OASIS restart files
- editing namelists (for models and OASIS)
- launching the run
- putting output files
- eventually looping for another job



Steps to setup a coupled simulation

(5-6) Define and run using the coupling toolbox

Coupling toolbox: configuration architecture



* More details: https://croco-ocean.gitlabpages.inria.fr/croco_doc/tutos/tutos.16.coupling.tools.html

Steps to setup a coupled simulation

(7) Outputs

Batch log file: YOUREXPER_YYYYMMDD_YYYYMMDD.o*

OASIS

Generated grid files:
- grids.nc
- masks.nc
- areas.nc

NB! If grids.nc or rmp*.nc files exist they will not be re-generated (useful for restart run, but can be source of error…)

Generated grid interpolation files:
- rmp_ww3t_to_ocnt_DISTWGT.nc
- rmp_ocnt_to_ww3t_DISTWGT.nc …

Generated restart files:
- rst_oce.nc, rst_atm.nc, rst_wav.nc
- (overwritten at the end of each simulation)

Logs:
- nout.000000
- crocox.timers_0000, wwatch.timers_0000
- debug.root.01, debug.root.02, …
- debug.notroot.01, debug.notroot.02, …

CROCO

Output files:
- croco_his.nc
- croco_avg.nc
- croco_rst.nc

Logs:
- croco.log
- + eventually standard output redirected to batch log file if LOGFILE cppkey not defined

WW3

Output files:
- out_grd.ww3 => ww3.DATE.nc
- out_pnt.ww3 => ww3.DATE_spec.nc

Logs:
- log.ww3
- output.ww3

WRF

Output files:
- wrfout_d01_DATE
- wrfxtrm_d01_DATE
- wrfrst_d01_DATE

Logs:
- rsl.error.0000
- rsl.out.0000

Steps to setup a coupled simulation

(7) Outputs

In case of error, you should check:

- The job output file: in $CHOME/jobs_YOUREXPER :
  YOUREXPER_YYYYMMDD_YYYYMMDD.o*
- The models' log files: either in
  $C**HOME/jobs_YOUREXPER**/YYYYMMDD_YYYYMMDD or in
  $**CWORK**/rundir/**YOUREXPER_execute**/YYYYMMDD_YYYYMMDD
  croco.log rsl.error.0000 log.ww3
- OASIS log files:
  nout.000000 debug.0?.000000

NB! Usually OASIS errors are in debug.root.0X, and model erros are either in their log, or in the standard output (may be in the batch log if no redirection)

Typical issues are:

- Files not found: check your file names, and location
- Unconsitent dimensions of the grids in the different files: check models grid files, OASIS grids and masks files, OASIS remapping weight files, namelists of models and OASIS (namcouple)
- Unconsistency in exchanged variables: check namcouple
- Model blow up: check the log files, if blow up is due to CFL (unrealistic speed, or segmentation fault) decrease the model time step

NB! If grids.nc or rmp*.nc files exist they will not be re-generated (useful for restart run, but can be source of error…)