Dr2xml Training

20 mai 2022

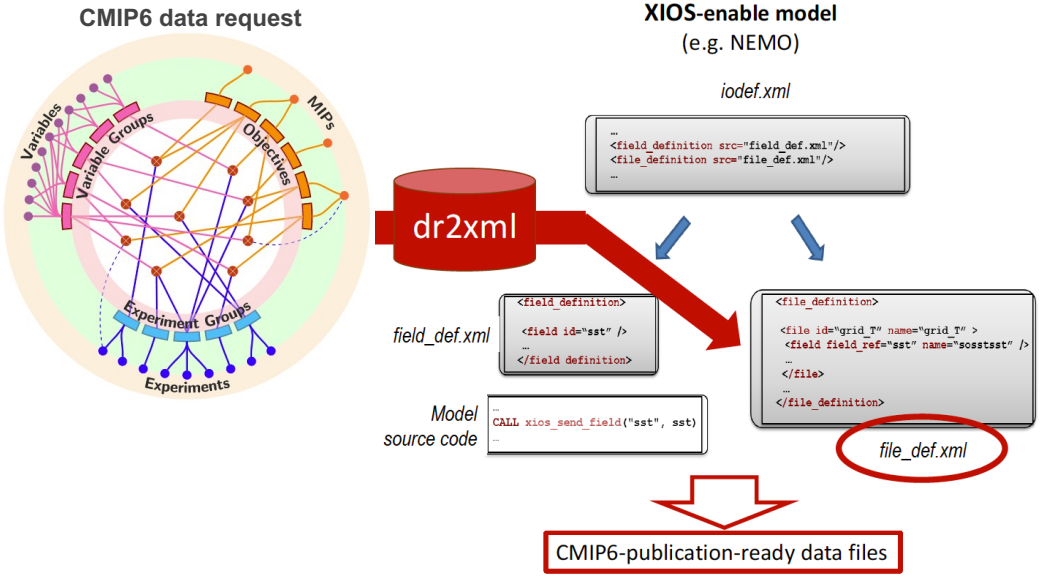
Gaëlle Rigoudy, CNRM, Meteo-France
Marie-Pierre Moine, Cerfacs

Outline

1. Introduction
   - Dr2xml, what's this?
   - Brief history
   - The CMIP6 Data Request
2. General features
   - Utility
   - Cautions
   - Simple functional scheme
   - The ping file
3. Usage
   - Installation
   - Configuration
   - Execution
   - Verification
4. Functionalities
   - Basic functions
   - Customisation
   - Extended usage

a) dr2xml, what's this?

- Python tool
- XIOS file-def XML writer
  - Automatic NetCDF file handling (naming, time-splitting, metadata, append write…)
  - fields and attributes (« variable » in XIOS vocab) in file
  - Automatic implementation of XIOS spatial & temporal filters
- Useful for :
  - XIOS-enabled models (output management)
  - large number of fields to output
  - standard data (format and content)
  - adding a lot of mandatory attributes in the netCDF output files
- CMIP6 production
  - Systematic production of CMIP6 compliant CF-netCDF files
  - Satisfying the CMIP6 Data Request

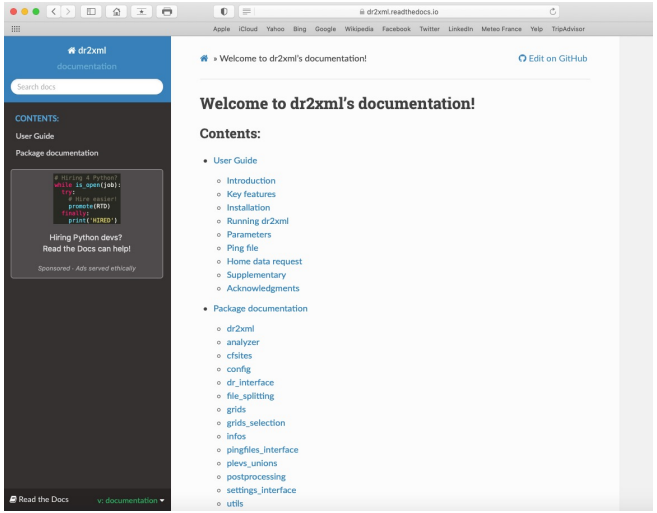


b) brief history

- Born with CMIP6 (Data Request)
- Development started in sept. 2016 (S. Sénési, CNRM + M.P Moine, Cerfacs)
- Now developed & maintained by G. Rigoudy, CNRM
- Developed in close collaboration with the XIOS-dev Team (a lot of functionalities introduced in XIOS for CMIP6/dr2xml needs)

- Used at IPSL and CNRM-CERFACS for climate models doing CMIP6
  - At CNRM-CERFACS: embedded in the modelling workflow (ECLIS)
  - At IPSL : used upstream of the modelling workflow (LibIGCM)

- All along the dev period, phasing with:
  - The CMIP6 Data Request versions
  - The CMIP6 Controlled vocabulary
- Stabilized version in July 2018 for CMIP6 (v1.13)
- Last evolutions:
  - Code modularization
  - New functionalities
  - Python 3 - PEP8



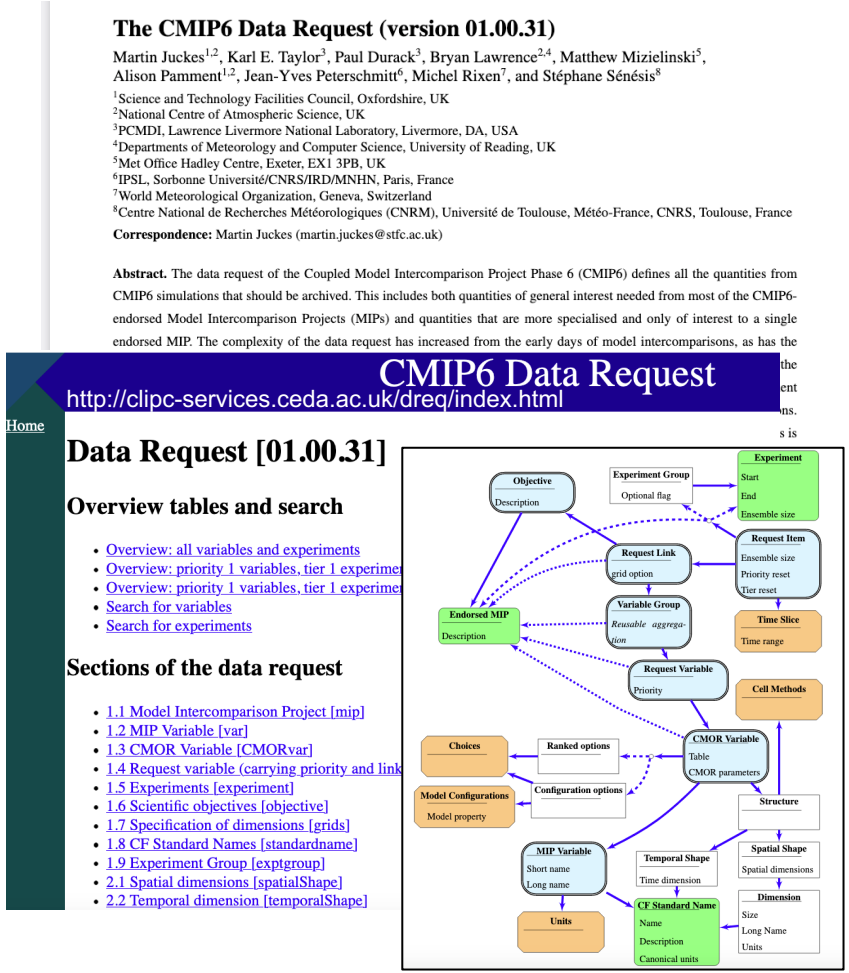
- Sources on Github: https://github.com/rigoudyg/dr2xml
- ReadtheDocs documentation : https://dr2xml.readthedocs.io/en/documentation/ *(not yet finalized)*

c) the CMIP6 Data Request

- Big picture :
  - Developed for CMIP6 by Martin Juckes since 2016 to…
  - Meet the challenge of model/MIP objectives/experiment design complexity and exposing number/diversity of diagnosis requested by each MIP
  - Fully enable the intercomparison :

« The thousands of diagnostics generated at each centre from hundreds of simulations should be produced and documented in a consistent manner to facilitate meaningful comparisons across models. Hence, for each experiment the MIPs have requested specific output to be archived and shared via the Earth System Grid Federation (ESGF), and the CMIP6 organisers have imposed requirements on file format and metadata »

- Concretely :
  - Is a data base with:
    - a python API (facilitates automated interrogation of the data base)
    - a browsable html interface
  - That gives, for each CMIP6 simulation:
    - the variables that should be output (according to the selected priority)
    - on which grid/domain/levels
    - over which time period
    - at which frequency
    - with which netCDF attributes...
  - CMIP6 DR python API is used by dr2xml



The CMIP6 Data Request (version 01.00.31)

Martin Juckes[1,2], Karl E. Taylor[3], Paul Durack[3], Bryan Lawrence[2,4], Matthew Mizielinski[5], Alison Pamment[1,2], Jean-Yves Peterschmitt[6], Michel Rixen[7], and Stéphane Sénésis[8]

[1]Science and Technology Facilities Council, Oxfordshire, UK
[2]National Centre of Atmospheric Science, UK
[3]PCMDI, Lawrence Livermore National Laboratory, Livermore, DA, USA
[4]Departments of Meteorology and Computer Science, University of Reading, UK
[5]Met Office Hadley Centre, Exeter, EX1 3PB, UK
[6]IPSL, Sorbonne Université/CNRS/IRD/MNHN, Paris, France
[7]World Meteorological Organization, Geneva, Switzerland
[8]Centre National de Recherches Météorologiques (CNRM), Université de Toulouse, Météo-France, CNRS, Toulouse, France

Correspondence: Martin Juckes (martin.juckes@stfc.ac.uk)

Abstract. The data request of the Coupled Model Intercomparison Project Phase 6 (CMIP6) defines all the quantities from CMIP6 simulations that should be archived. This includes both quantities of general interest needed from most of the CMIP6-endorsed Model Intercomparison Projects (MIPs) and quantities that are more specialised and only of interest to a single endorsed MIP. The complexity of the data request has increased from the early days of model intercomparisons, as has the

CMIP6 Data Request
http://clipc-services.ceda.ac.uk/dreq/index.html

Home

Data Request [01.00.31]

Overview tables and search

- Overview: all variables and experiments
- Overview: priority 1 variables, tier 1 experime
- Overview: priority 1 variables, tier 1 experime
- Search for variables
- Search for experiments

Sections of the data request

- 1.1 Model Intercomparison Project [mip]
- 1.2 MIP Variable [var]
- 1.3 CMOR Variable [CMORvar]
- 1.4 Request variable (carrying priority and link
- 1.5 Experiments [experiment]
- 1.6 Scientific objectives [objective]
- 1.7 Specification of dimensions [grids]
- 1.8 CF Standard Names [standardname]
- 1.9 Experiment Group [exptgroup]
- 2.1 Spatial dimensions [spatialShape]
- 2.2 Temporal dimension [temporalShape]

a) dr2xml utility

- Management of complex and various outputs, supporting:
  - all physical field shapes *(1D, 2D, 3D, 4D, time-varying or constant)*
  - all output frequencies *(yearly, monthly, daily, 6-hourly, 3-hourly, 1-hourly, sub-hour)*
  - choice of sampling period
  - on the fly diagnostic computation *(ex. zonal means, interpolation on pressure levels, on observation sites…)*
  - allows multivariate diagnostics (computing a diagnostics depending on 2 or more model native variables)
- Avoids time-consuming post-processing steps:
  - no need to use CMOR
  - nor any other offline post-processing steps (even for diagnostic computation)
  - formatting/standardisation directly ensured (file names, global and local attributes, temporal axis)
- Reduced risk of errors
  - “all included” and integrated ~~post-~~processing
  - one tool does all : XIOS !
  - homogeneity, coherence, reliability, robustness

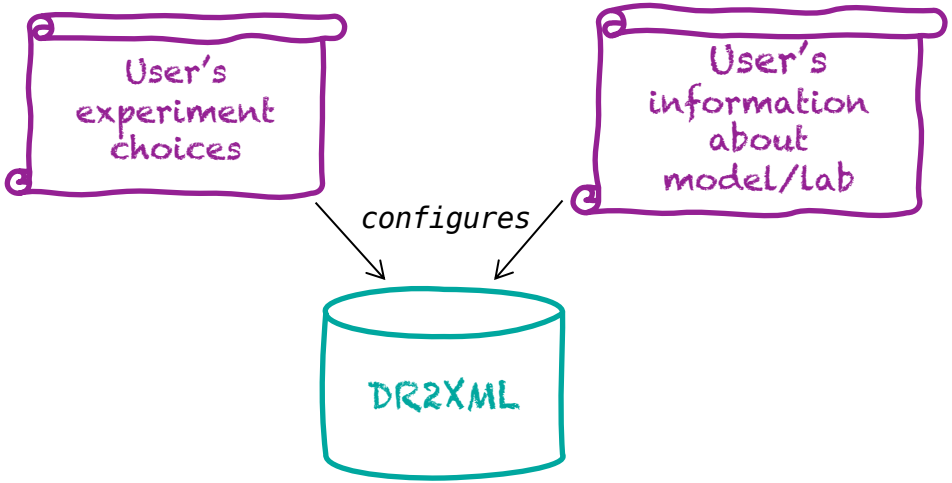
b) cautions

- One shot ! which means…
  - no safety net once the production started
  - all requested variables *and* variables you need must be there !

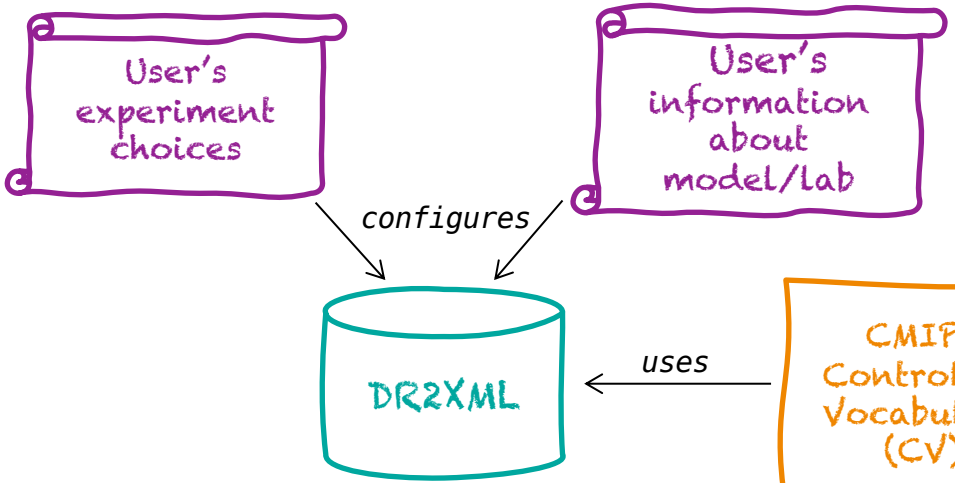
- Before running the whole simulation:
  - Dr2xml configuration must be carefully checked (a verbose log file enables to visualize the planned output variables)
  - It is highly recommended to perform a short test run and control/validate the output netCDF files, for e.g. with NCTIME and potentially PrePARE (if the simulation is a CMIP6 one)

c) simple functional scheme

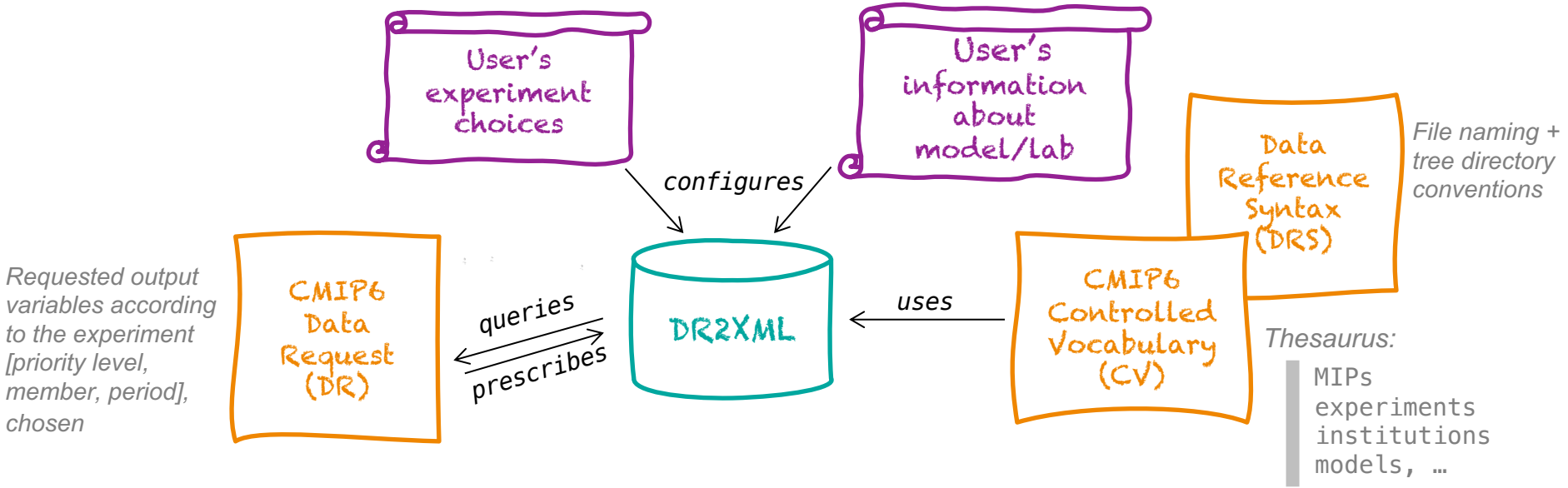
c) simple functional scheme



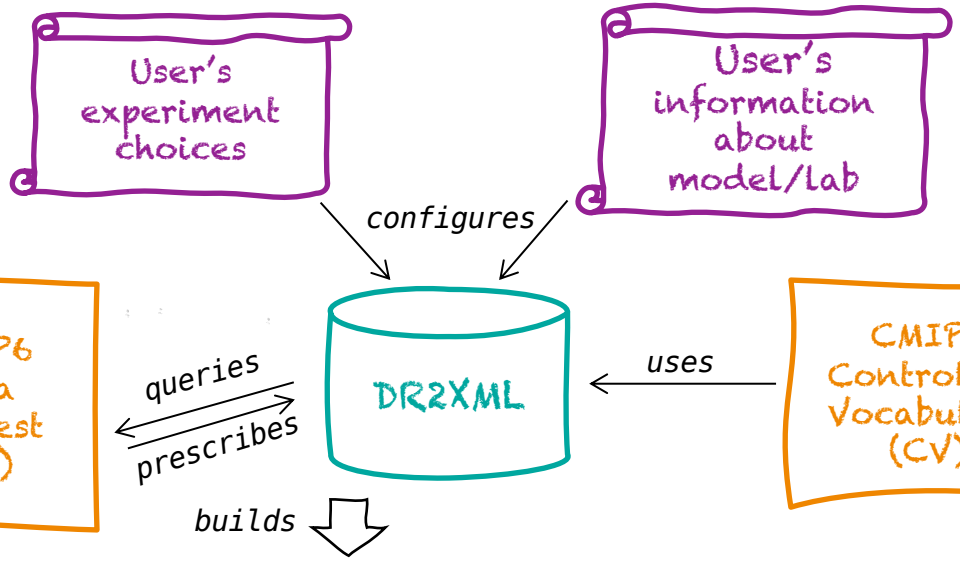
c) simple functional scheme



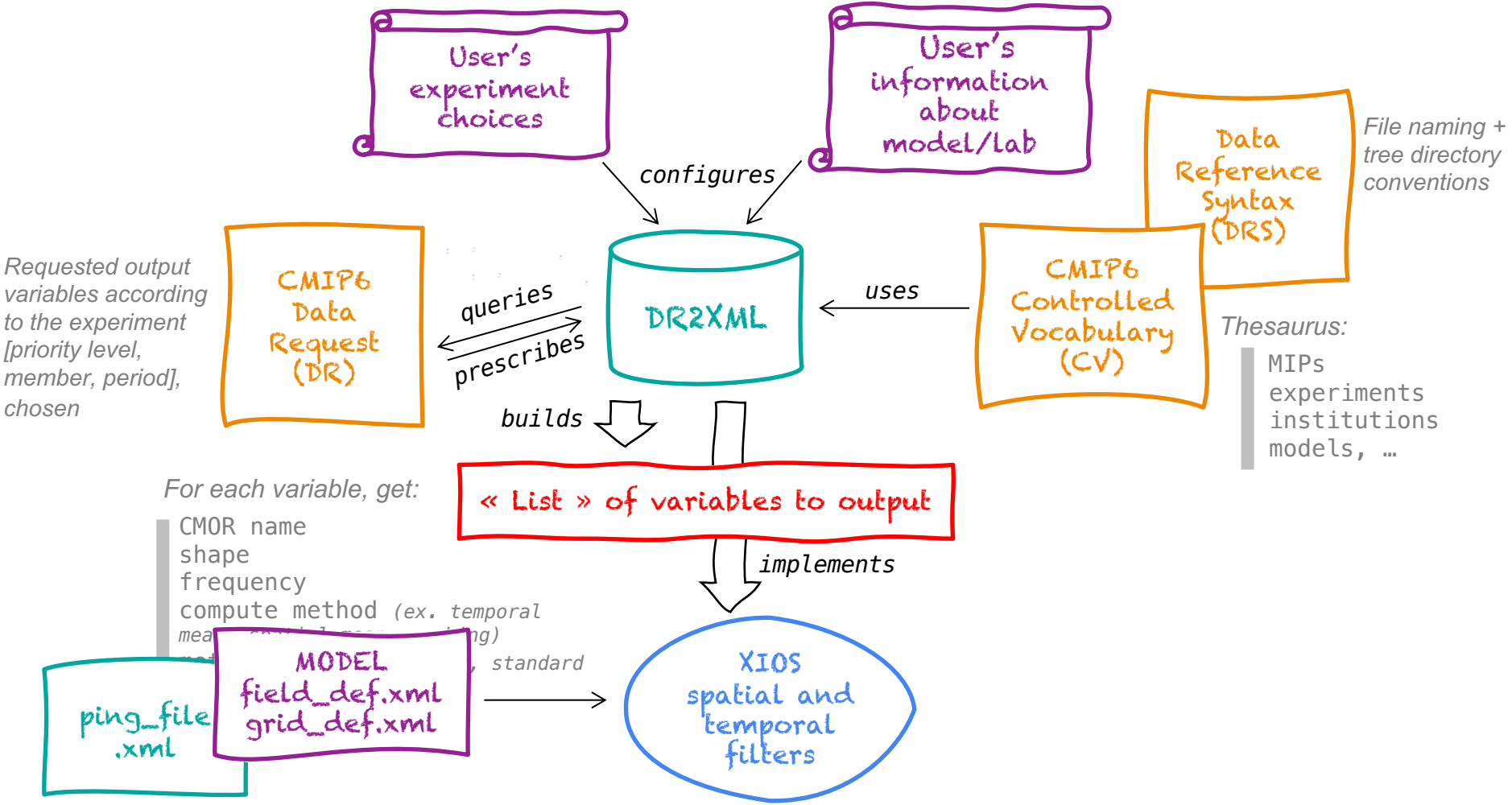
c) simple functional scheme



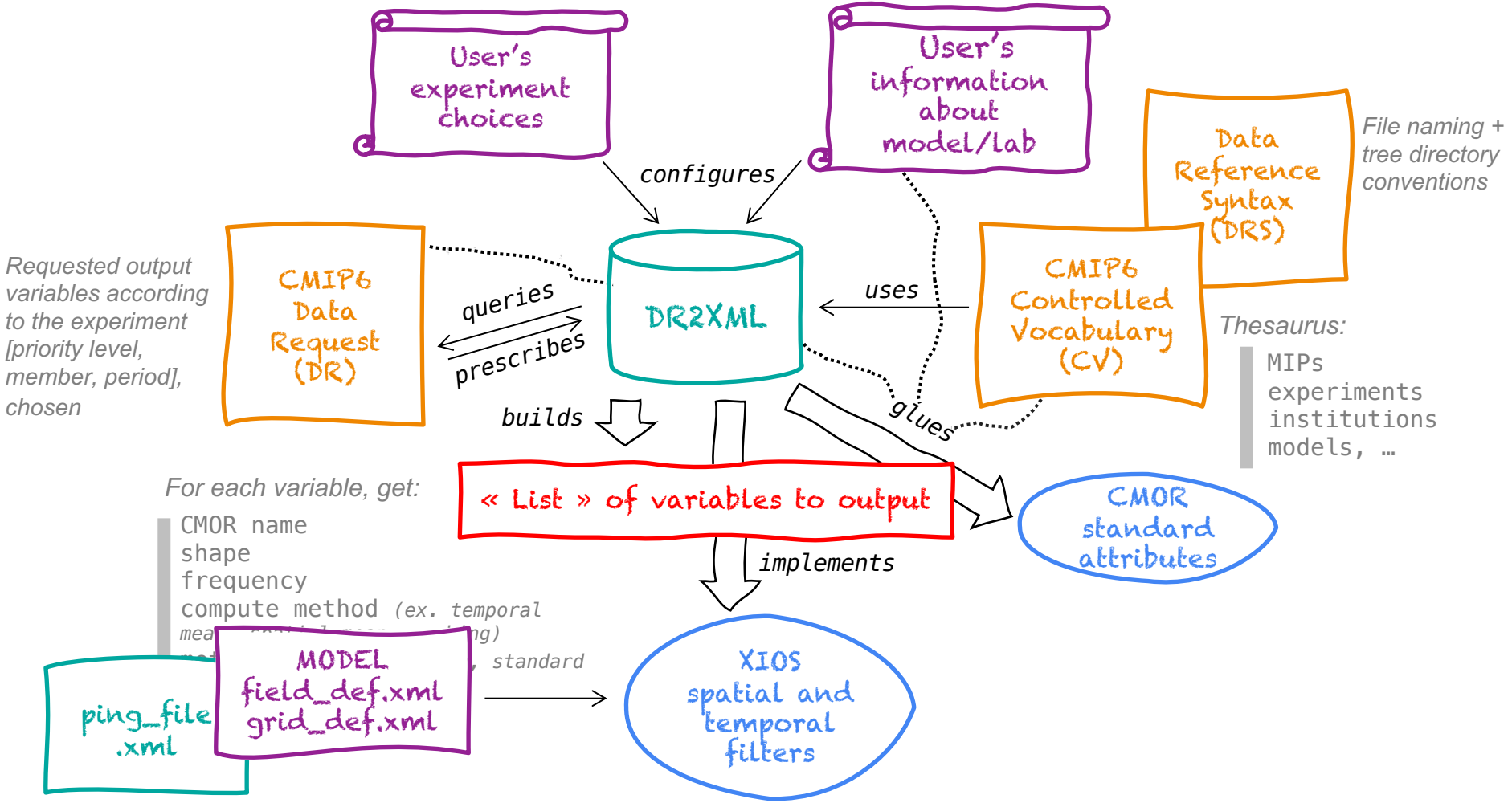
c) simple functional scheme



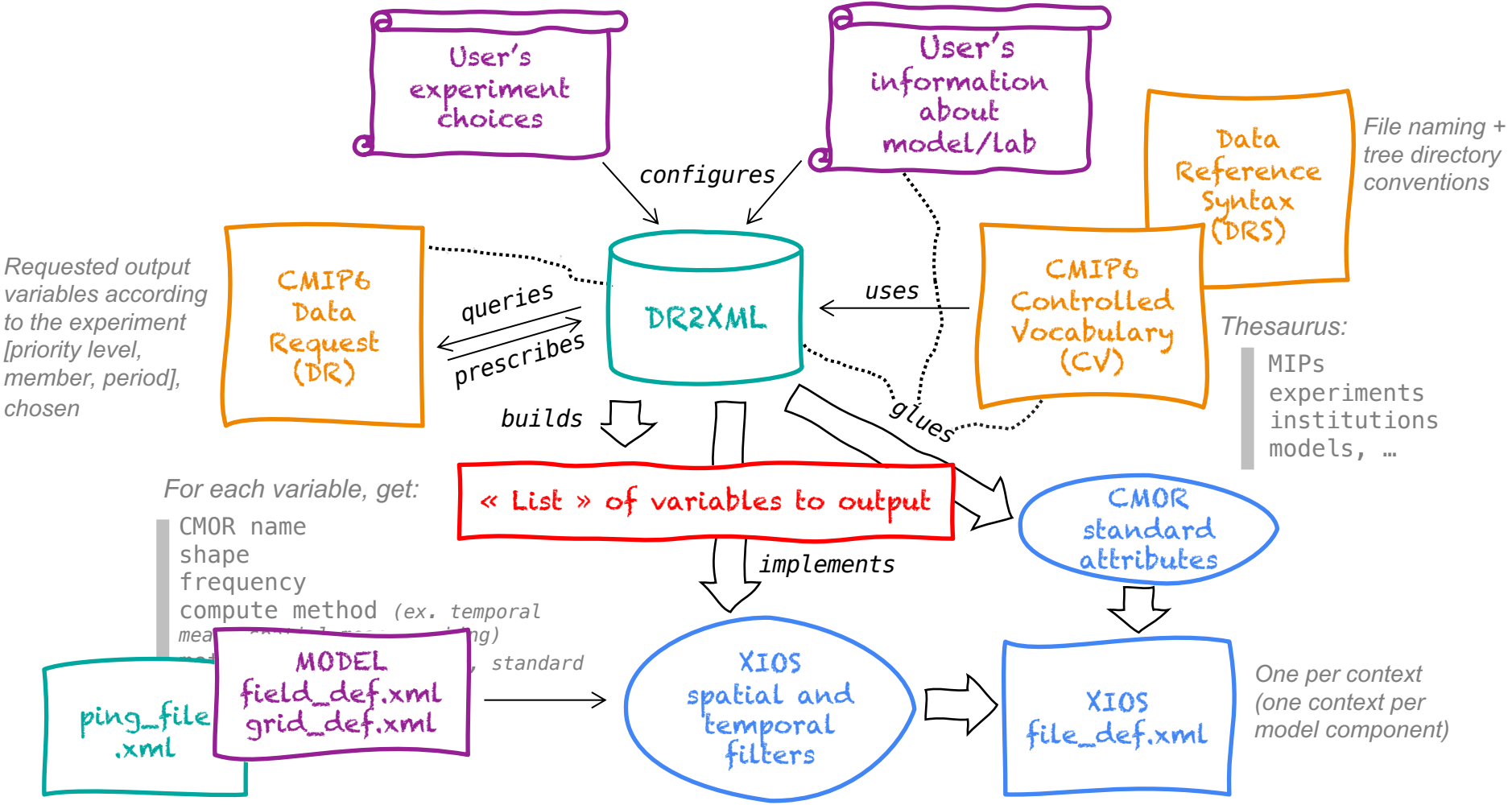
c) simple functional scheme



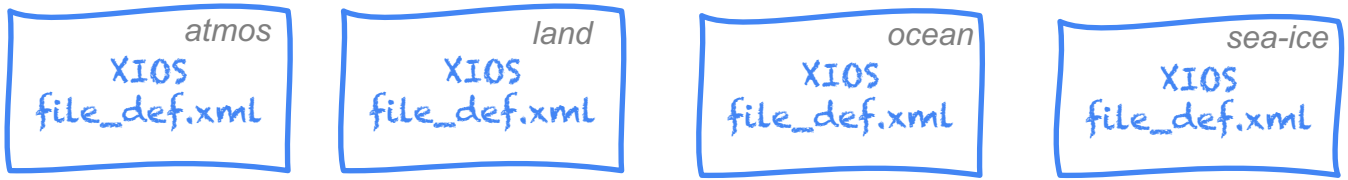
c) simple functional scheme



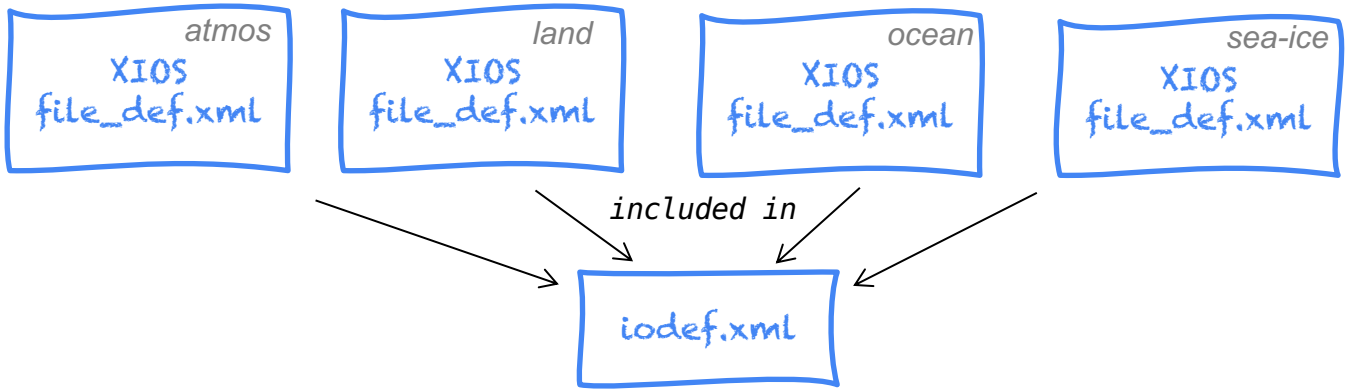
c) simple functional scheme



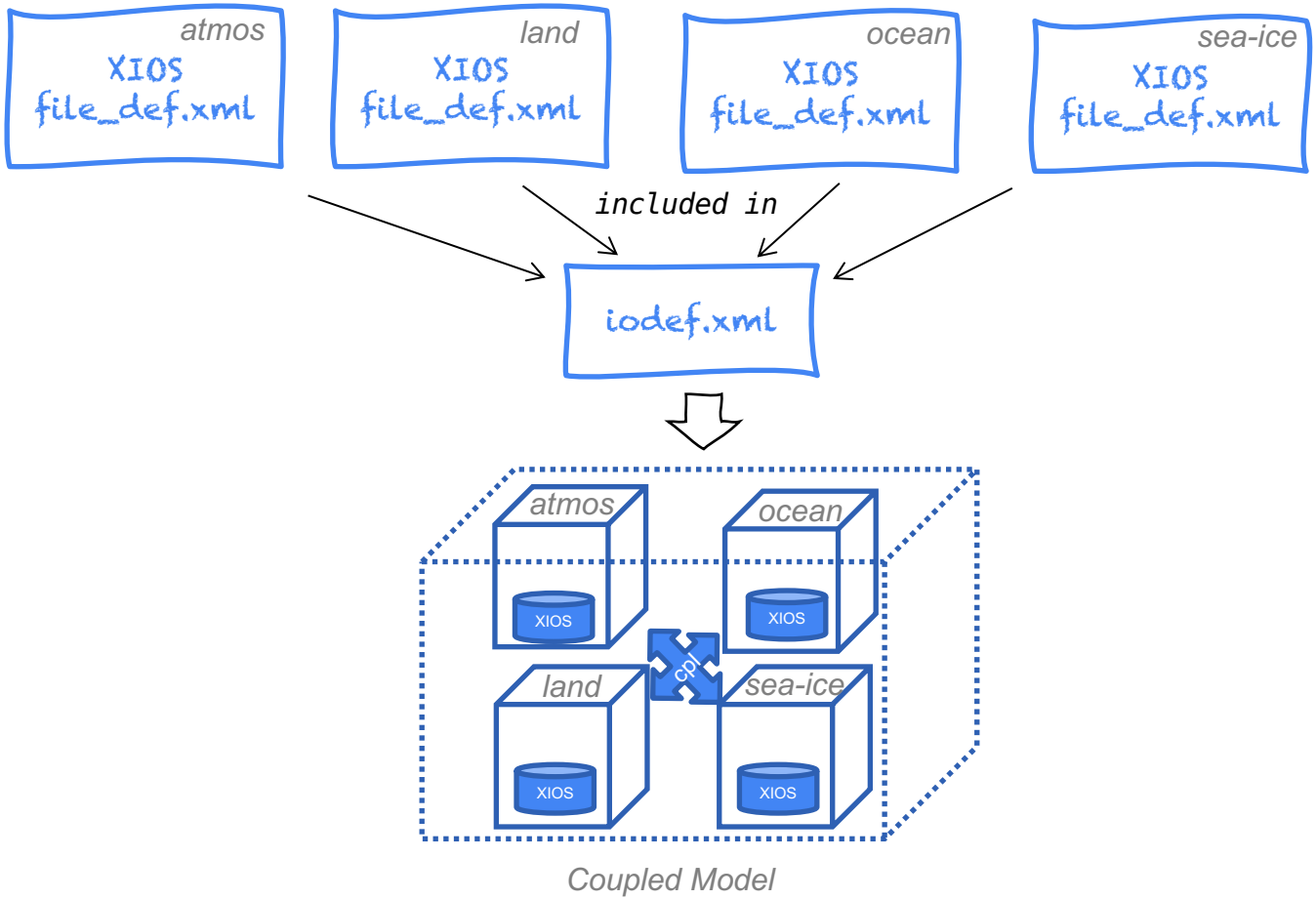
c) simple functional scheme (cont.)



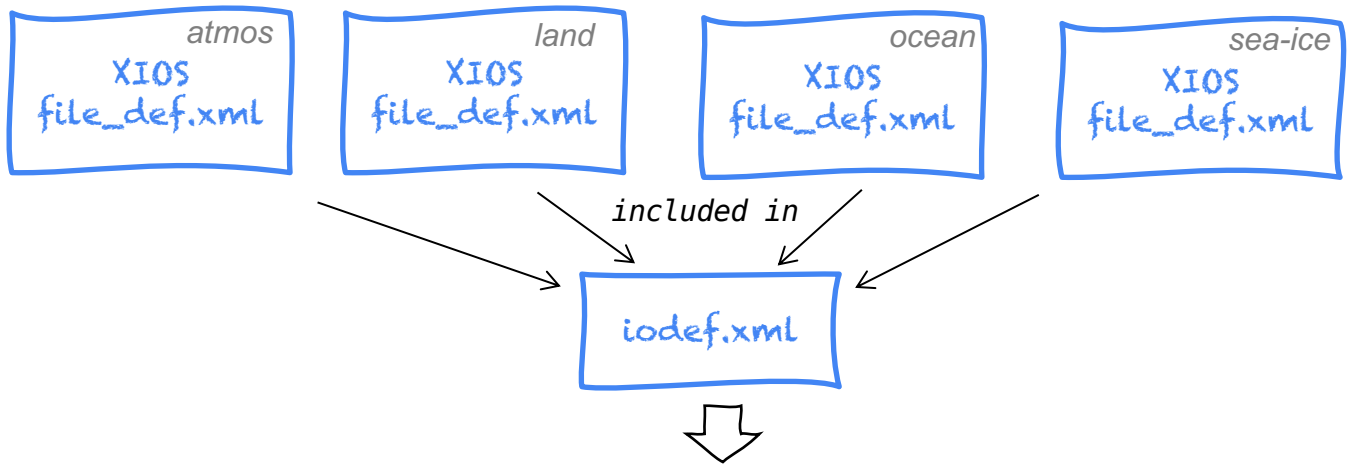
c) simple functional scheme (cont.)



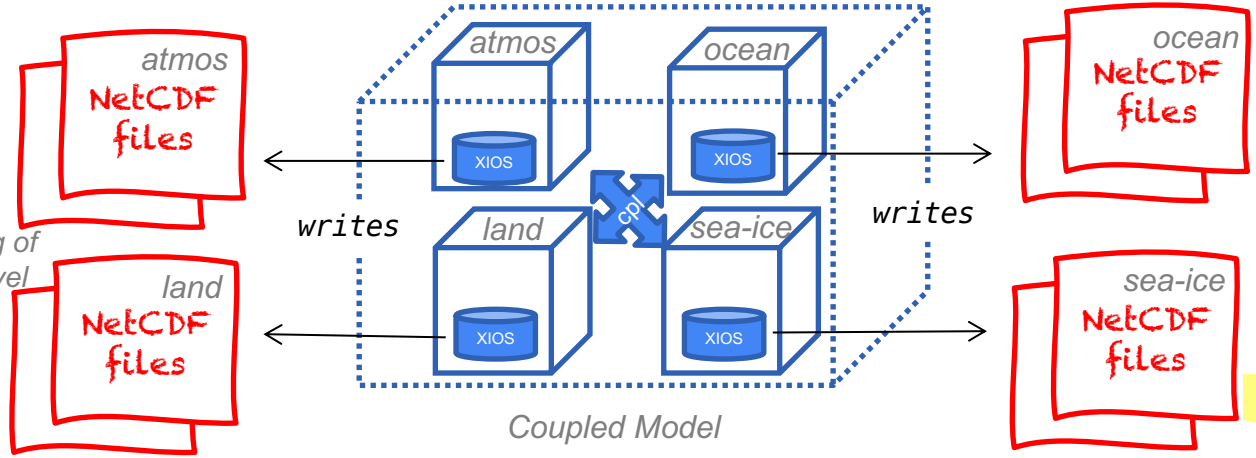
c) simple functional scheme (cont.)



c) simple functional scheme (cont.)



- *one file per physical field*
- *append write (-> split_freq)*
- *Simultaneous writing of netCDF files (2nd level XIOS servers)*

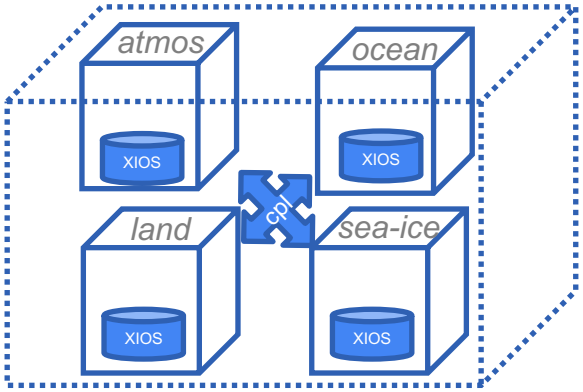


Ready to be published on ESGF.

d) the ping files

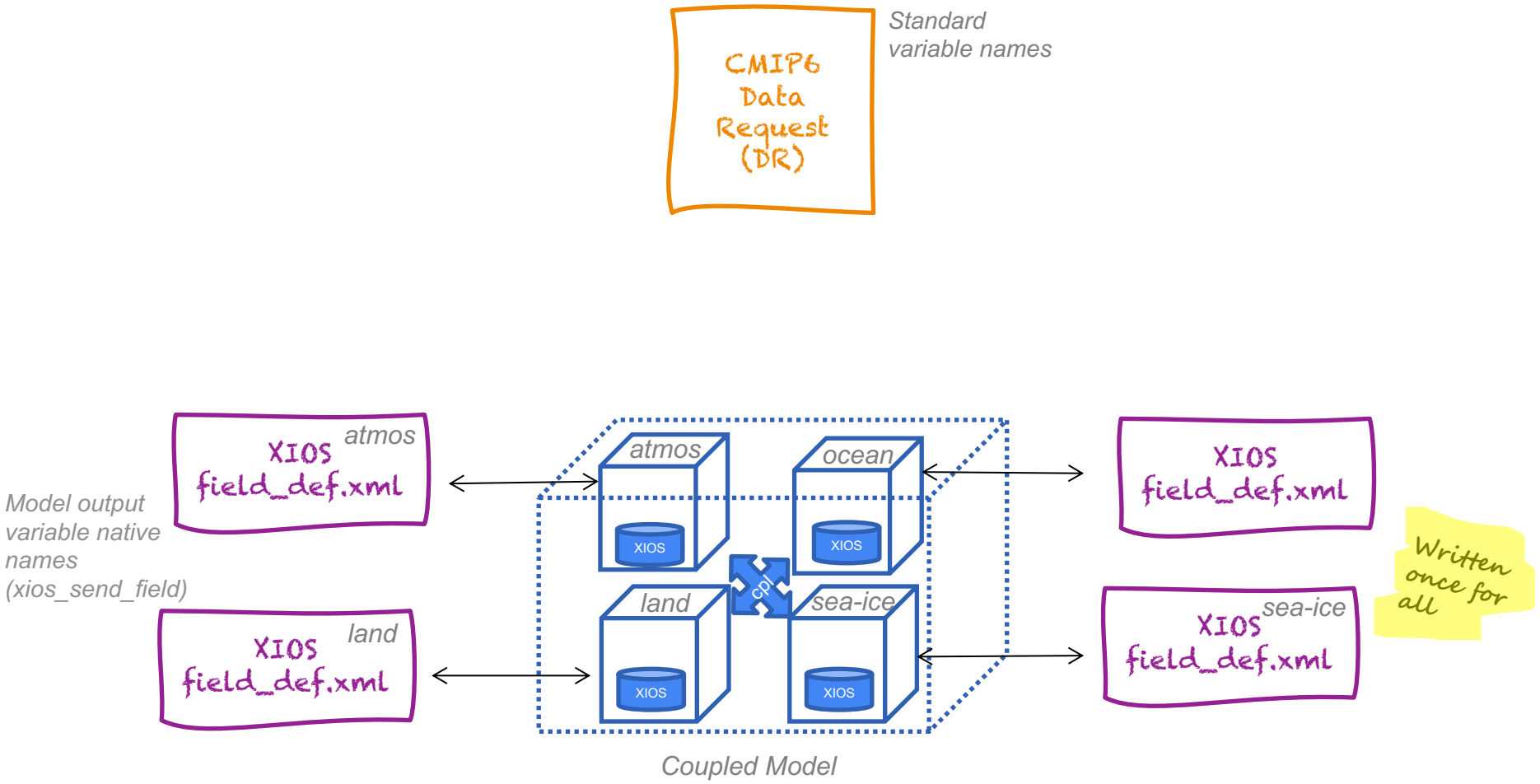
CMIP6 Data Request (DR)

Standard variable names

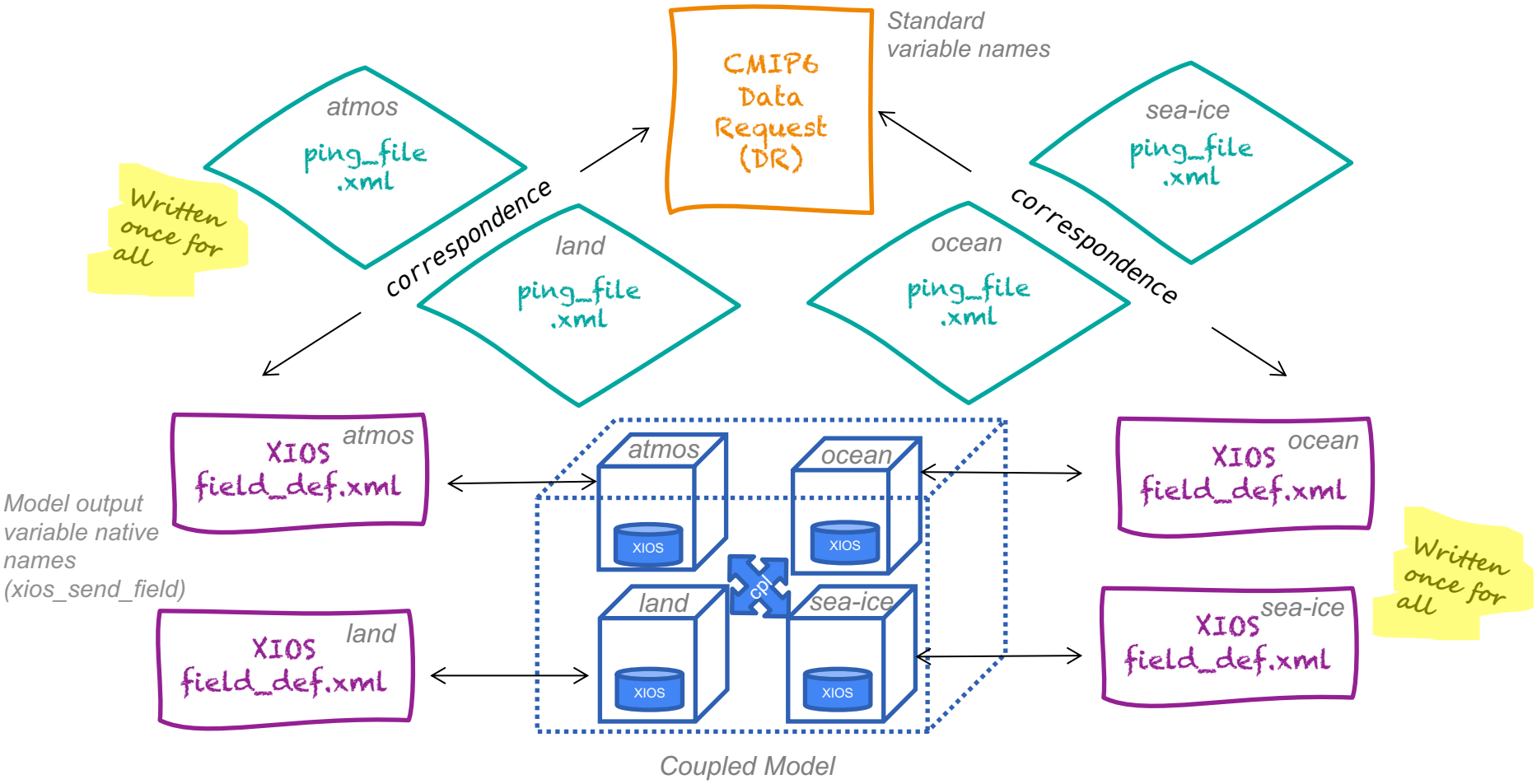


Coupled Model

d) the ping files



d) the ping files



e) “sos_Omon” example (DR info.)

1.3 CMOR Variable: [sos] Sea Surface Salinity

- label : sos
- vid : [var] sos [74a9891bcab2667dbcb66574c6370c86]
- title : Sea Surface Salinity
- defaultPriority : 1
- modeling_realm : ocean
- type : real
- 2.3 Dimensions and related information [stid]: Temporal mean, Global field (single level) [XY-na] [amse-tmn] [str-a098]
- processing : Report on native horizontal grid as well as on a spherical latitude/longitude grid.
- frequency : mon
- rowIndex : 22
- mipTable : Omon
- description : Sea water salinity is the salt content of sea water, often on the Practical Salinity Scale of 1978. However, the unqualified term 'salinity' is generic and does not necessarily imply any particular method of calculation. The units of salinity are dimensionless and the units attribute should normally be given as 1e-3 or 0.001 i.e. parts per thousand.
- […]

1.2 MIP Variable: [sos] Sea Surface Salinity

- label : sos
- provmip : [mip] OMIP [OMIP]
- prov : OMIP.day
- units : 0.001
- title : Sea Surface Salinity
- uid : 74a9891bcab2667dbcb66574c6370c86
- procnote : ('',)
- procComment :
- 1.8 CF Standard Names [sn]: SeaSurfaceSalinity
- description : […]

1.8 CF Standard Names: [SeaSurfaceSalinity] Sea Surface Salinity

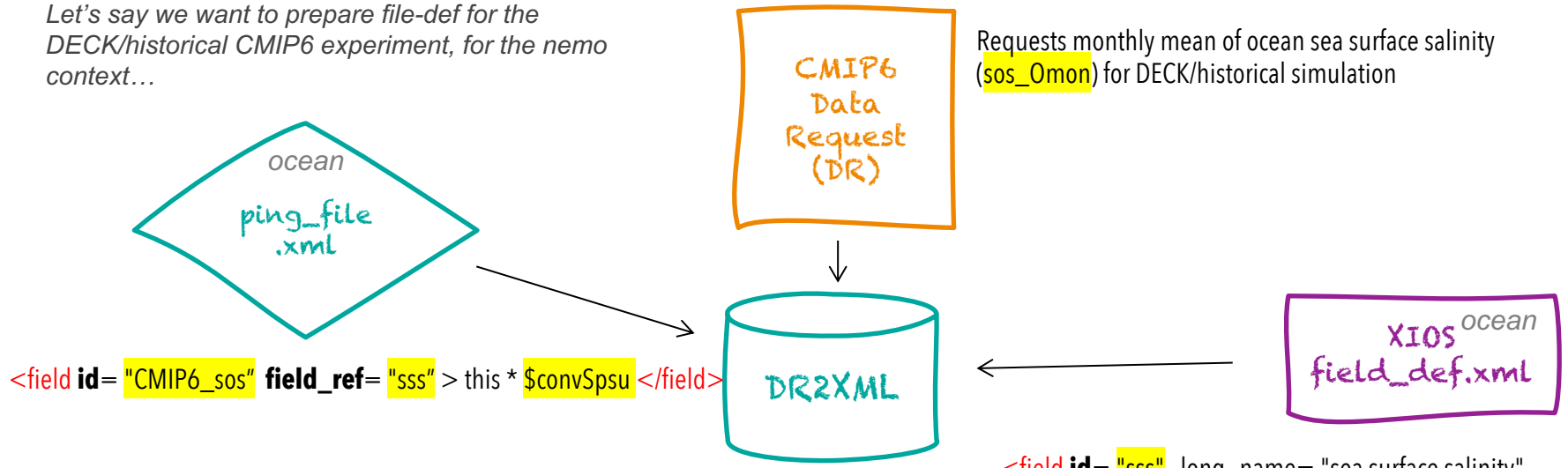
- units : 1e-3
- label : SeaSurfaceSalinity
- title : Sea Surface Salinity
- uid : sea_surface_salinity
- description : […]

2.3 Dimensions and related information: [str-a098] Temporal mean, Global field (single level) [XY-na] [amse-tmn]

- label : str-a098
- spid : [spatialShape] XY-na [a656047a-8883-11e5-b571-ac72891c3257]
- title : Temporal mean, Global field (single level) [XY-na] [amse-tmn]
- cell_methods : area: mean where sea time: mean
- cell_measures : area: areacello
- description : For time mean fields, it may be useful to add information about the sampling interval in the cell_methods string. The syntax is to append, in brackets, 'interval: *amount* *units*', for example 'area: time: mean (interval: 1 hr)'. The units must be valid UDUNITS, e.g. day or hr.
- […]

e) "sos_Omon" example (ping file)

Let's say we want to prepare file-def for the DECK/historical CMIP6 experiment, for the nemo context…




```
<file id= "sos_Omon_gn" name= "sos_Omon_CNRM-CM6-1_historical_r1i1p1f1_gn_%start_date%-%end_date%"
output_freq= "1mo" append="true" […] >
    […]
    <field name= "sos" field_ref= "CMIP6_sos_average" operation="average" detect_missing_value="True" default_value="1.e+20" prec="4"
    cell_methods="area: mean where sea time: mean" cell_methods_mode="overwrite"> […] </field>
    […]
    <field id= "CMIP6_sos_average" field_ref ="CMIP6_sos" operation= "average" />
</file>
```

e) “sos_Omon” example (cont.)

dr2xml_nemo.xml

```
<file id="sos_Omon_gn" name="sos_Omon_CNRM-CM6-1_historical_r1i1p1f1_gn_%start_date%-%end_date%" output_freq="1mo" append="
true" output_level="10" compression_level="4" split_freq="500y" split_freq_format="%y%mo" split_last_date="2014-- 00:00:00"
time_units="days" time_counter_name="time" time_counter="exclusive" time_stamp_name="creation_date" time_stamp_format="
%Y-%m-%dT%H:%M:%SZ" uuid_name="tracking_id" uuid_format="hdl:None/%uuid%" convention_str="CF-1.7 CMIP-6.2">
    <variable name="activity_id" type="string"> CMIP </variable>
    <variable name="data_specs_version" type="string"> 01.00.32 </variable>
    <variable name="dr2xml_version" type="string"> 2.2 </variable>
    <variable name="experiment_id" type="string"> historical </variable>
    <variable name="description" type="string"> CMIP6 historical </variable>
    <variable name="title" type="string"> CMIP6 historical </variable>
    <variable name="experiment" type="string"> all-forcing simulation of the recent past </variable>
    <variable name="external_variables"
    <variable name="forcing_index" type
    <variable name="frequency" type="st
    <variable name="further_info_url" t
    CMIP6.CNRM-CERFACS.CNRM-CM6-1.histo
    <variable name="grid" type="string"
    <variable name="grid_label" type="s
    <variable name="nominal_resolution"
    <variable name="history" type="stri
    <variable name="initialization_inde
    <variable name="institution_id" typ
    <variable name="institution" type="
    CERFACS (Centre Europeen de Recherc
    <variable name="license" type="stri
    Attribution-NonCommercial-ShareAlik
    https://pcmdi.llnl.gov/CMIP6/TermsO
    proper acknowledgment. Further info
    further_info_url (recorded as a glo
    warranty, either express or implied
    particular purpose. All liabilities
    negligence) are excluded to the ful
    <variable name="mip_era" type="stri
    <variable name="parent_experiment_i
    <variable name="parent_mip_era" typ
    <variable name="parent_activity_id"
    <variable name="parent_source_id" t
    <variable name="parent_time_units"
    <variable name="parent_variant_labe
    <variable name="branch_method" type
    <variable name="branch_time_in_chil
    <variable name="physics_index" type
    <variable name="product" type="stri
    <variable name="realization_index"
    <variable name="realm" type="string"> ocean </variable>
    <variable name="source" type="string"> CNRM-CM6-1 (2017):
aerosol: prescribed monthly fields computed by TACTIC_v2 scheme
atmos: Arpege 6.3 (T127; Gaussian Reduced with 24572 grid points in total distributed over 128 latitude circles (with 256 grid points
per latitude circle between 30degN and 30degS reducing to 20 grid points per latitude circle at 88.9degN and 88.9degS); 91 levels;
top level 78.4 km)
atmosChem: OZL_v2
land: Surfex 8.0c
ocean: Nemo 3.6 (eORCA1, tripolar primarily 1deg; 362 x 294 longitude/latitude; 75 levels; top grid cell 0-1 m)
seaIce: Gelato 6.1 </variable>
```

```
    <variable name="source_id" type="string"> CNRM-CM6-1 </variable>
    <variable name="source_type" type="string"> AOGCM </variable>
    <variable name="sub_experiment_id" type="string"> none </variable>
    <variable name="sub_experiment" type="string"> none </variable>
    <variable name="table_id" type="string"> Omon </variable>
    <variable name="title" type="string"> CNRM-CM6-1 model output prepared for CMIP6 / CMIP historical </variable>
    <variable name="variable_id" type="string"> sos </variable>
    <variable name="variant_info" type="string"> none </variable>
    <variable name="variant_label" type="string"> r1i1p1f1 </variable>
    <field name="sos" field_ref="CMIP6_sos_average" operation="average" detect_missing_value="True" default_value="1.e+20"
    prec="4" cell_methods="area: mean where sea time: mean" cell_methods_mode="overwrite">
        <variable name="standard_name" type="string"> sea_surface_salinity </variable>
        <variable name="description" type="string"> Sea water salinity is the salt content of sea water, often on the
        Practical Salinity Scale of 1978. However, the unqualified term 'salinity' is generic and does not necessarily imply
        any particular method of calculation. The units of salinity are dimensionless and the units attribute should normally
        be given as 1e-3 or 0.001 i.e. parts per thousand. </variable>
        <variable name="long_name" type="string"> Sea Surface Salinity </variable>
        <variable name="history" type="string"> none </variable>
        <variable name="units" type="string"> 0.001 </variable>
        <variable name="cell_methods" type="string"> area: mean where sea time: mean </variable>
        <variable name="cell_measures" type="string"> area: areacello </variable>
        <variable name="interval_operation" type="string"> 1800 s </variable>
    </field>
</file>
```

ocean

XIOS
file_def.xml

CNRM-CM6 coupled model

- *Arpege/Surfex: 28 758 lines !*
- *Nemo/Gelato: 14 840 lines !*
- *Trip: 383 lines*

a) installation

- Dr2xml (+ XlsXwritter, six)

```
$ git clone https://github.com/rigoudyg/dr2xml.git
```

- dreqPy (the CMIP6 Data Request)

```
$ pip install –upgrade [--user]  dreqPy==01.00.32
```

or :

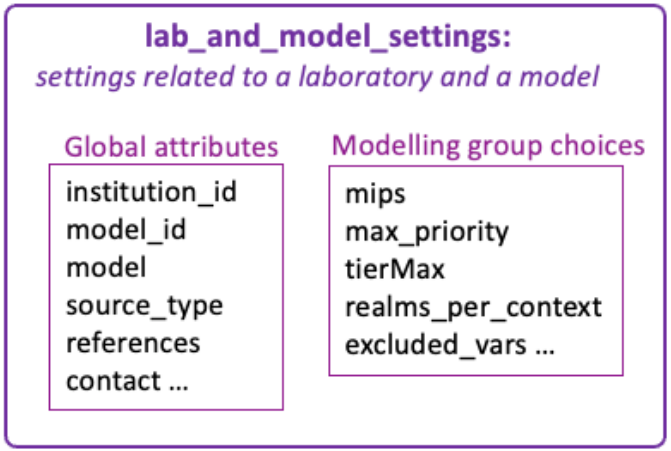
```
$ svn co http://proj.badc.rl.ac.uk/svn/exarch/CMIP6dreq/tags/01.00.32
```

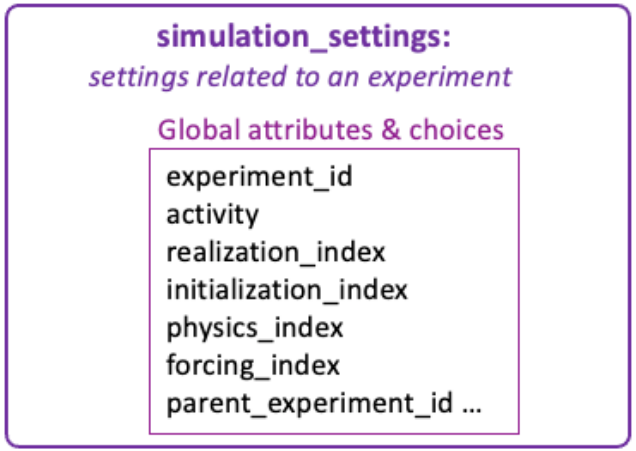
- CMIP6_CVs

```
$ git clone https://github.com/WCRP-CMIP/CMIP6_CVs
```

b) configuration

- 2 files of dr2xml settings (python dictionaries)





- a set of model related XML files for XIOS : context, domains, field_defs
- additional XML files, the so-called "ping_files" (one per context)

c) execution

```
from dr2xml import generate_file_defs # ------> Import dr2xml
my_cvs_path='/home/user/CMIP6_CVs/' # --------> Path to CMIP6 CVs (json files)
my_outdir='/home/user/dr2xml_output/' # ------> Path for dr2xml output
generate_file_defs(lab_and_model_settings, simulation_settings,
                   year="2000", enddate="2100",
                   context='nemo',  pingfiles='nemo_pingfile.xml',
                   printout=True, cvs_path=my_cvpath,
                   dummies='skip', dirname=my_outdir)
```

nemo.xml
(file_def)

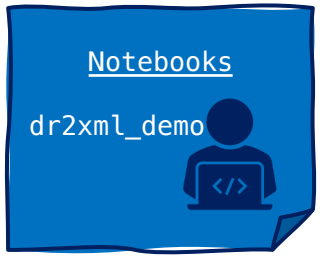
d) verification

```
Skipped variables (i.e. whose alias is not present in the pingfile):
>>> TABLE:             Ofx 03/09 ----> sftof(1) ugrid(1) volcello(1)
>>> TABLE:           SImon 01/20 ----> sirdgconc(1)
>>> TABLE:            Omon 03/37 ----> msftmz(1) msftmzmpa(1) vsf(1)
```

```
Some Statistics on actually written variables per frequency+shape...

                           EmonZ P1   1 :  ['sltbasin']
                            Omon P1   1 :  ['hfbasin']
                            Omon P2   2 :  ['htovgyre', 'htovovrt']
      mon     YB-na    -------- ---   4

                             Omon P1  10 :  ['bigthetao', 'so', 'thetao', 'thkcello',
                                             'umo', 'uo', 'vmo', 'vo', 'wmo', 'wo']
      mon      XY-O    -------- ---  10


                          PrimOday P1   1 :  [u'so']
      day      XY-O    -------- ---   1
```

```
Some Statistics on actually written variables per variable...

------------------------
--- VARNAME:  bigthetao : Sea Water Convervative Temperature
------------------------
             *  mon_Omon_XY-O_1

-----------------
--- VARNAME:  so : Sea Water Salinity
-----------------
             *  mon_Omon_XY-O_1
             *  day_PrimOday_XY-O_1.0
```

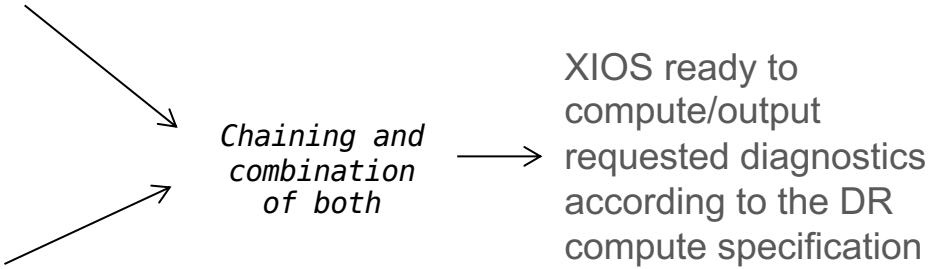
- 2 user-friendly views in dr2xml log file
- …That does not exempt from looking carefully at the generated file-def !



a) basic functions

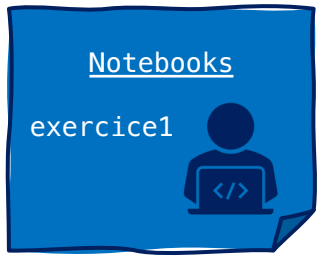
"basic functions" : **ordinated** and **optimised** combination of XIOS filters automatically implemented according to the specifications of the data request.

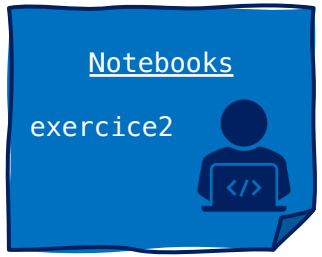
- Spatial:
  - remapping on different grids
  - regridding on pressure levels
  - regridding on height levels *(new !)*
  - interpolation at observation sites
  - meridional/zonal means
- Temporal:
  - sampling period
  - time mean / time point
  - diurnal cycle



b) customisation

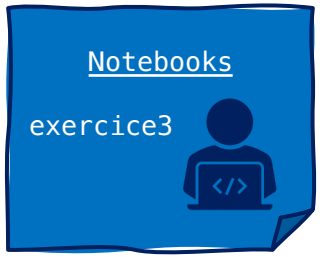
- Filtering options :
  - curation : can filter out variables that are requested twice
  - selection :
    - filtering on experiment, on simulated year, or both
    - choosing the maximum priority level of variables to output
  - exclusion/inclusion : user can also provide lists of excluded/included…
    - variables (var)
    - tables (tbl)
    - pairs (var,tbl)
    - spatial shapes
  - metadata filtering : user can take the control on attributes to write in the NetCDF files



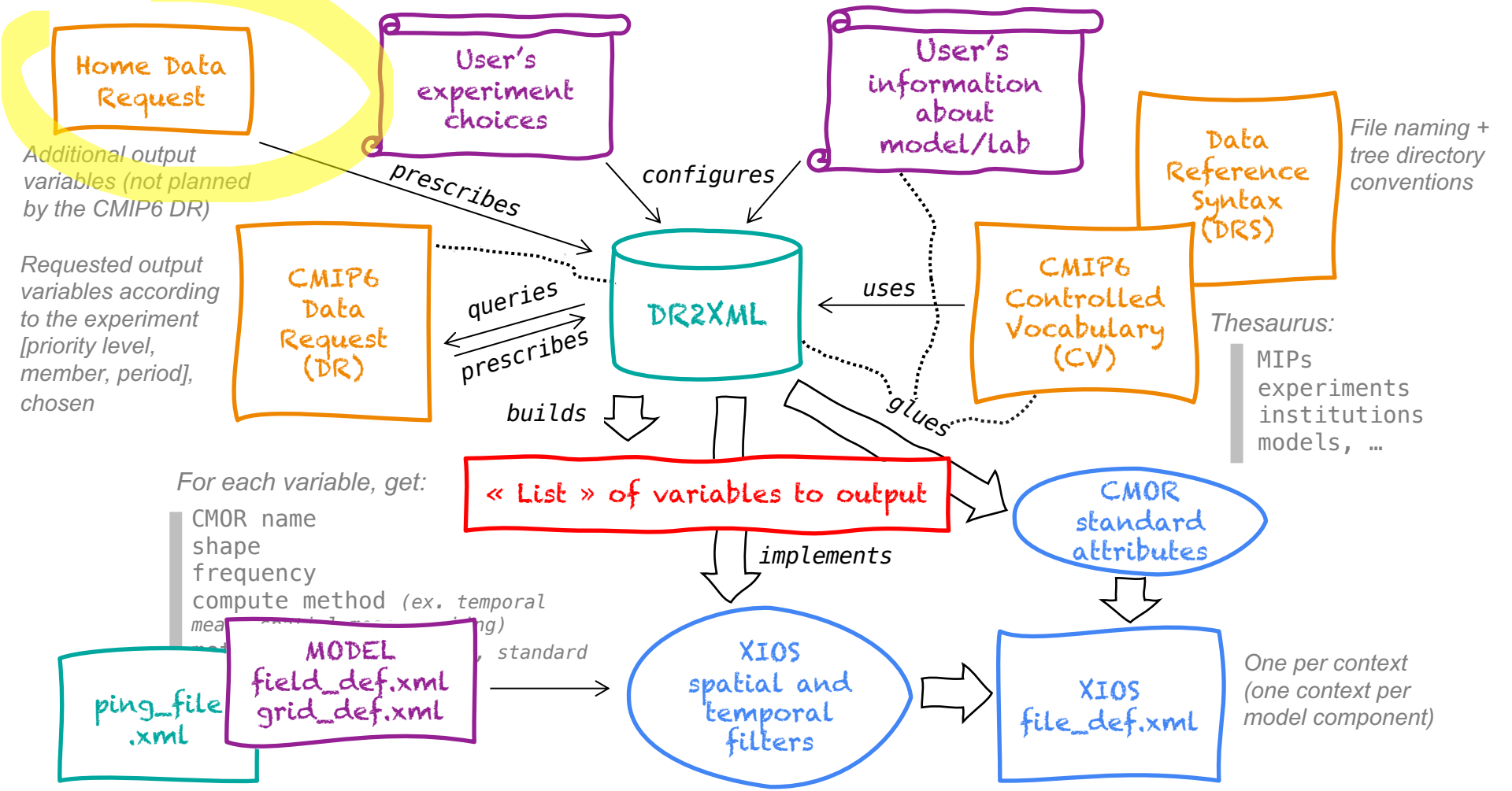


b) customisation (cont.)

- Add methods :
  - dr2xml allows **to output additional variables** = the so-called "home data request"
    - "cmor" variables : to output a variable that is defined in the CMIP6 Data Request but not requested for this experiment
    - "extra" variables : to output non 'cmor' variables but with all standards attributes ; variables are defined through additional table(s)
    - "perso" variables : to quicky output a model native variable, without caring about naming nor attributes *('perso' var. don't need to be defined in the ping file)*
    - "dev" variables : a flexible way to output variables with a minimal set of attributes, possibly on a different grid, customised on the fly ; useful during the model development phase *(dev var don't need to be defined in the field_def xml nor in the ping file)*
  - The "home data request" was used during CMIP6 production as a "safety net" to compensate DR potential lacks



b) "home data request"

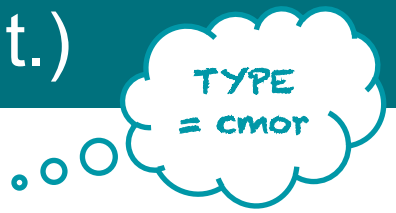


b) “home data request”

- The “home data request” , brief user guide:
  - a simple text file
  - 3 header lines
  - one line per home variable
  - 9 [+5] columns, one per home variable parameter/attribute
  - parameter values are separated by ';'
  - no empty trailing lines

```
#-----------------------------------------------------------------------------------------------------------------------
#TYPE;    VARNAME;     REALM;     FREQUENCY;     TABLE;                TEMPORAL_SHP;     SPATIAL_SHP;     EXPNAME;     MIP
#-----------------------------------------------------------------------------------------------------------------------
cmor;     huss;        atmos;     day;           CMIP6_day;            time-mean;        XY-na;           ANY;         ANY
extra;    wap;         atmos;     6hrPt;         TRAINING_6hrPtExe3;   ANY;              ANY;             ANY;         ANY
perso;    pblh;        atmos;     mon;           NONE;                 time-mean;        XY-na;           ANY;         ANY
#

#-----------------------------------------------------------------------------------------------------------------------
#TYPE;    VARNAME;     REALM;     FREQUENCY;     TABLE;                TEMPORAL_SHP;     SPATIAL_SHP;     EXPNAME;     MIP   [...]
#-----------------------------------------------------------------------------------------------------------------------
dev;      wg_patch1;   atmos;     day;           CNRM_day;             time-mean;        XY-na;           ANY;         ANY;
#

                                  ---------------------------------------------------------------------------------------
                                  UNITS;   LONG_NAME;                                STD_NAME;          SOURCE_GRID; TARGET_H_GRID
[...]                             ---------------------------------------------------------------------------------------
                                  m3 m-3;  Soil Liquid Water Content By Patch;  soil_liquid_water;  NATURE;    NATURE_grid
```

b) “home data request” (cont.)



Example 1 : adding a CMIP6 variable with all of its CMOR standard attributes

```
-------------------------------------------------------------------------------------------------------------------
TYPE;     VARNAME;     REALM;     FREQUENCY;     TABLE;          TEMPORAL_SHP;     SPATIAL_SHP;     EXPNAME;     MIP
-------------------------------------------------------------------------------------------------------------------
cmor;     sos;         ocean;     day;           CMIP6_Oday;     time-mean;        XY-na;           ANY;         ANY
```

Official CMIP6 varname…

…defined for this realm/context

…at this frequency

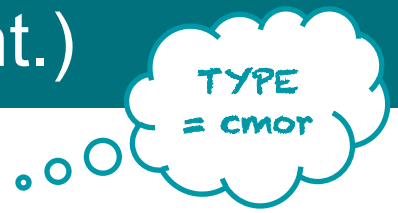
…in this CMIP6 table

…with this temploral shape

…and with this spatial shape

To check consistency of user’s demand with respect to the chosen (VARNAME, TABLE)

b) "home data request" (cont.)



Example 1 : addir

```
--------------------
TYPE;     VARNAME;
--------------------
cmor;     sos;
```

CMI6_Oday.json

a CMIP6/CMOR official table (json file) containing several variables.

```
"Header": {
    "data_specs_version": "01.00.30",
    "cmor_version": "3.4",
    "table_id": "Table Oday",
    "realm": "ocnBgchem",
    "table_date": "09 May 2019",
    "missing_value": "1e20",
    "int_missing_value": "-999",
    "product": "model-output",
    "approx_interval": "1.00000",
    "generic_levels": "olevel",
    "mip_era": "CMIP6",
    "Conventions": "CF-1.7 CMIP-6.2"
},
"variable_entry": {
    "sos": {
        "frequency": "day",
        "modeling_realm": "ocean",
        "standard_name": "sea_surface_salinity",
        "units": "0.001",
        "cell_methods": "area: mean where sea time: mean",
        "cell_measures": "area: areacello",
        "long_name": "Sea Surface Salinity",
        "comment": "Sea water salinity is the salt content of sea
          water, often on the Practical Salinity Scale of 1978 […]",
        "dimensions": "longitude latitude time",
        "out_name": "sos",
        "type": "real",
        "positive": "",
        "valid_min": "",
        "valid_max": "",
        "ok_min_mean_abs": "",
        "ok_max_mean_abs": ""
    },
```

Most of this information is mirrored in the CMIP6 Data Request

```
--------------------------
;      EXPNAME;        MIP
--------------------------
       ANY;            ANY
```

b) “home data request” (cont.)

TYPE = cmor

Example 1 : adding a CMIP6 variable with all of its CMOR standard attributes

```
-------------------------------------------------------------------------------------------------------------------
TYPE;    VARNAME;    REALM;    FREQUENCY;    TABLE;          TEMPORAL_SHP;    SPATIAL_SHP;    EXPNAME;    MIP
-------------------------------------------------------------------------------------------------------------------
cmor;    sos;        ocean;    day;          CMIP6_Oday;     time-mean;       XY-na;          ANY;        ANY
```

- sos; — official CMIP6 varname…
- ocean; — …defined for this model component
- day; — …at this frequency
- CMIP6_Oday; — …in this CMIP6 table
- time-mean; — …with this temporal shape
- XY-na; — …and with this spatial shape

To check consistency of user’s demand with respect to the chosen (VARNAME, TABLE)

‘sos_Oday’ is well taken into account by dr2xml :

```
>>> TABLE          Oday 01 ----> sos(1)
```

b) "home data request" (cont.)

TYPE = extra

Example 2 : adding all variables included in a user-defined external Table with CMOR-like attributes

PRIMAVERA_Prim6hrPt.json

a user-defined external table (json file) containing several additional variables.

```
"Header": {
        "data_specs_version": "01.00.13",
        "table_id": "Table Prim6hrPt",
        "realm": "land atmos ocean",
        "cmor_version": "3.2",
        "table_date": "12 July 2017",
        "missing_value": "1e20",
        "product": "model-output",
        "approx_interval": "0.250000",
        "generic_levels": "",
        "mip_era": "PRIMAVERA",
        "Conventions": "CF-1.7 CMIP-6.0"
    },
    "variable_entry": {
        "thetapv2": {
            "frequency": "6hr",
            "modeling_realm": "atmos",
            "standard_name": "theta_on_pv2_surface",
            "units": "K",
            "cell_methods": "time: point",
            "cell_measures": "area: areacella",
            "long_name": "Theta on PV +/- 2 Surface",
            "comment": "",
            "dimensions": "longitude latitude time1",
            "out_name": "thetapv2",
            "type": "real",
            "positive": "",
            "valid_min": "",
            "valid_max": "",
            "ok_min_mean_abs": "",
            "ok_max_mean_abs": "",
            "primavera_priority": "1"
        },
```

CMIP6 (CMOR) like attributes used by dr2xml to build XIOS instructions (output_freq, operation, …) and to set XIOS variables (i.e. netCDF attributes)

b) "home data request" (cont.)

TYPE = extra

Example 2 : adding all variables included in a user-defined external Table with CMOR-like attributes

```
TYPE;    VARNAME;   REALM;    FREQUENCY;    TABLE;                 TEMPORAL_SHP;  SPATIAL_SHP;  EXPNAME;    MIP
-------------------------------------------------------------------------------------------------------------------
extra;   ANY;       ANY;      ANY;          PRIMAVERA_Prim6hrPt;   ANY;           ANY;          ANY;        HighResMIP;
```

means we want to output all the variables defined in the external Table…

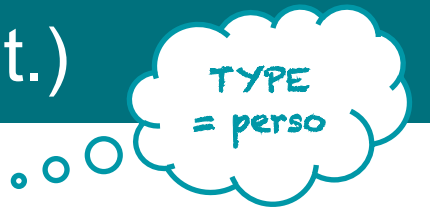
the user-defined external table containing several additional variables

… but only if the experiment is an HighResMIP one

All variables defined as a *variable_entry* in `PRIMAVERA_Prim6hrPt.json` are added by dr2xml (*provided they are defined in the ping file*) :

```
>>> TABLE       Prim6hrPt 07 ----> ps(1.0) hus1000(1.0) thetapv2(1.0)
                                    clt(1.0) va1000(1.0) ua1000(1.0) tslsi(1.0)
```

b) “home data request” (cont.)



Example 3 : adding a non-CMIP6 (model native) variable without any specific attributes. The variable has not to be defined in field_def XML file but not in the ping file (keep the original name)

TYPE;	VARNAME;	REALM;	FREQUENCY;	TABLE;	TEMPORAL_SHP;	SPATIAL_SHP;	EXPNAME;	MIP
perso;	sst;	ocean;	day;	CNRM_cnrmDay;	time-mean;	XY-na;	ANY;	ANY

native variable name in the model (not a CMIP6/CMOR name) ; no need to be defined in the ping file.

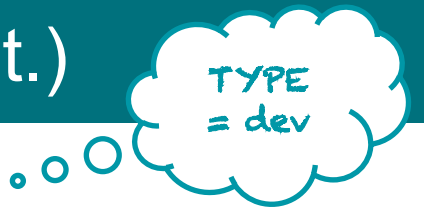
sent by this model component

no json table associated in this case ; only here to set the <table> facet in the file name

‘sst’ is well taken into account by dr2xml (*without being defined in the ping file*, only in the field_def XML) :

```
>>> TABLE            cnrmDay 01 --> sst(1)
```

b) “home data request” (cont.)



Example 4 : adding a non-CMIP6 (model native) variable, defining “on the fly” the target grid and its attributes (units, standard name…). Avoid to have this variable defined in XIOS field_def XML file and/or to write an extra json table. Can be seen as an hybrid of the ‘perso’ and ‘extra’ type (for devel. purpose only)

```
#-------------------------------------------------------------------------------------------------------------------------------------------------------
#TYPE; VARNAME; REALM;  FREQUENCY; TABLE;   TEMPORAL_SHP; SPATIAL_SHP; EXPNAME; MIP;  UNITS; LONG_NAME; STD_NAME;  SOURCE_GRID; TARGET_H_GRID
#-------------------------------------------------------------------------------------------------------------------------------------------------------
dev;  hmv4;  ocean;  day;  CNRM_day; time-mean;  XY-perso;  ANY;  ANY;  hmv4_units; hmv4 name; hmv4_name; complete_klev; FULL
```

like the “perso” case

means a 3D field with a user defined shape :

- an **horizontal domain** eventually to be remapped on TARGET_H_GRID (otherwise keep hmv4 on the SOURCE_GRID domain)
- a **customised vertical axis** given by an additional dictionary in the simulation settings

attributes of hmv4

means domain regridding : “complete” → “FULL”

```
sset['perso_sdims_description'] = {
    'hmv4':{
        'my_dimension':{
            'stdname': 'altitude',
            'long_name': 'my_long_name',
            'positive': True,
            'requested': '',
            'value': '30 50 60',
            'requested': '30 50 60',
            'units': 'my_units',
            'type': 'float',
            'axis': 'Z' } } }
```

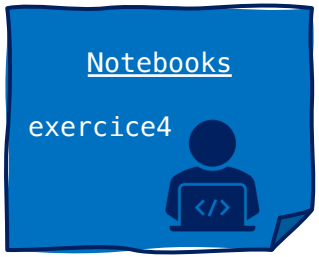
‘hmv4’ is well taken into account by dr2xml (without being defined in any XML file):

```
>>> TABLE                day 01 --> hmv4 (1)
```

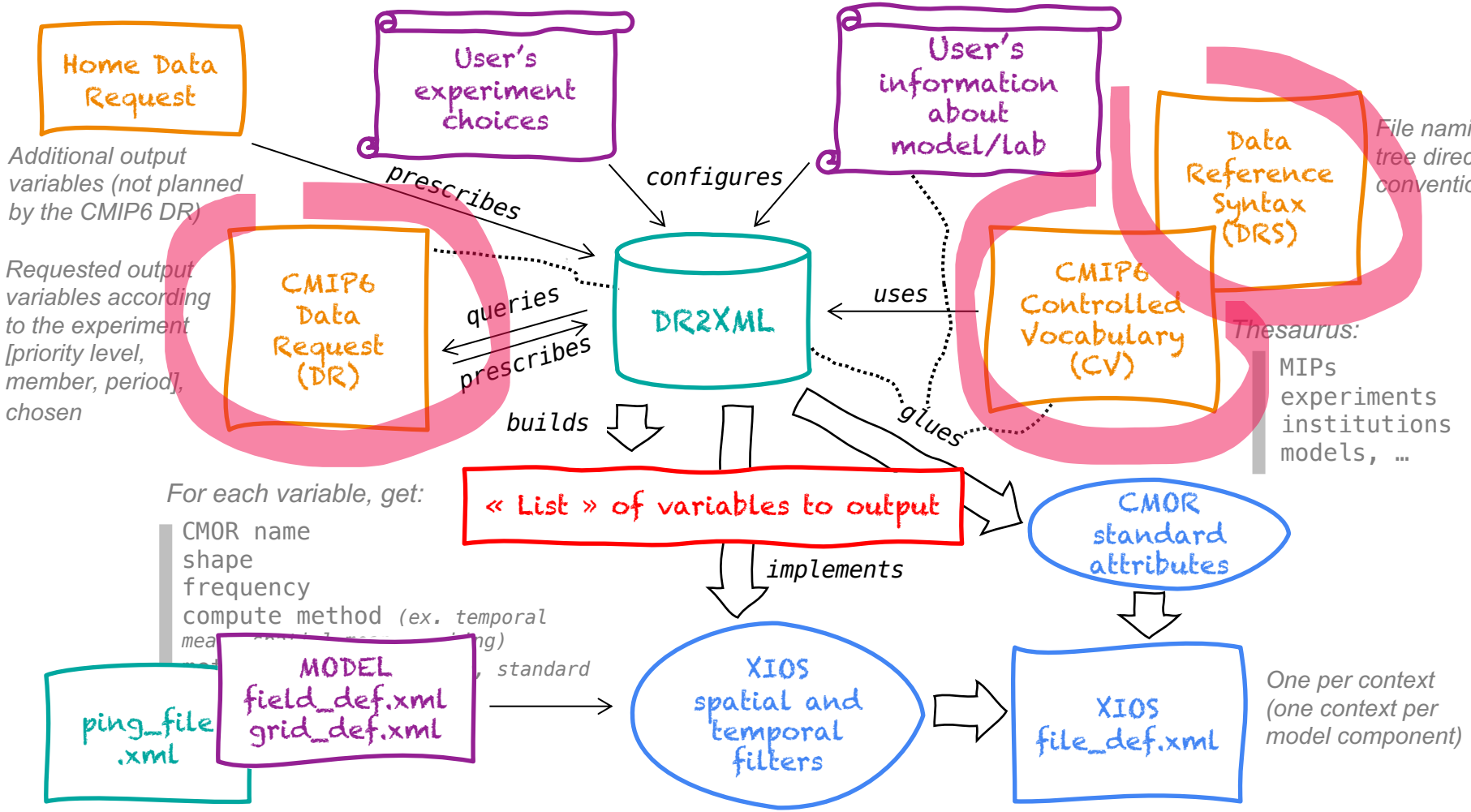
c) extended usage

- Dr2xml for daily research production
  - discipline : Adopt the good practices using the CMOR/CMIP6 standards, even for non CMIP6 production
  - flexibility : The user can free somewhat or a great from the DR, choosing to :
    - make a new simulation with the same output variables as in a given CMIP6 experiment
    - ignore all of from the CMIP6 DR and only specify its own outputs via the "home data request"

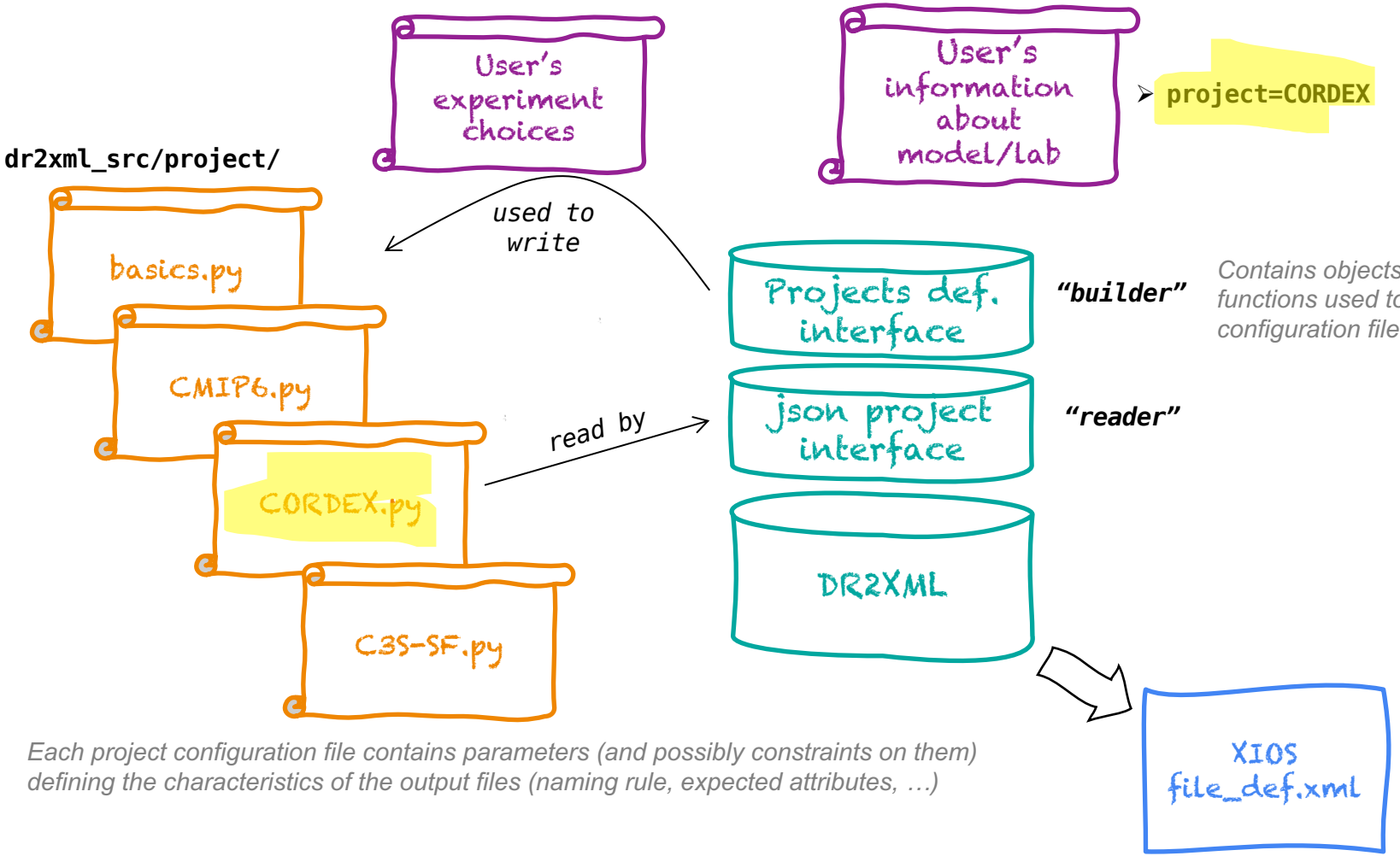
- Adaptation to other projects (like CORDEX)
  - new functionalities implemented *(e.g. interpolation to altitude level)*
  - code cleaning to remove all CMIP6 hard-coded elements
  - code restructured to externalise everything that depends on the project
  - a new interface to define projects *(CMIP6, CORDEX, etc.)*
  - A new interface to avoid calling the CMIP6 Data Request



c) Adaptation to other projects



c) Adaptation to other projects



Each project configuration file contains parameters (and possibly constraints on them) defining the characteristics of the output files (naming rule, expected attributes, …)

summary

- DR-dr2xml-XIOS pipeline is designed to facilitate the configuration of XIOS-enabled climate models contributing to CMIP exercises
- Is the best way to conform (as far as we can) to a data request as complex as the CMIP6 one
- Scraps the nightmare of a “by hand” model output configuration
- Dynamic analyse of the simulated period and adaptation of output configuration consequently
- Files written by XIOS configured by dr2xml are CMIP6-compliant
- Avoid the CMORisation step in the data production workflow
- Can be used as well for daily research simulations, benefitting (or not…) from the CMIP6 standards
- Has evolved to be used widely beyond CMIP6 and to integrate any project constraints (ex. CORDEX)

Without CMIP6 and the Data Request dr2xml won’t have existed, even less without XIOS, but now we’ve got it, sounds there is a life (for it) after CMIP6 ! ☺

The End

Any questions ?

Slack channel :

dr2xmlusers.slack.com

Contact :

gaelle.rigoudy@meteo.fr
moine@cerfacs.fr

hands-on (notebooks)

PCform
LOGIN : stageXX
PASSWD: stageXX

- Before starting the hands-on, please **update the dr2xml_training sources**:

```
cd $HOME/dr2xml_training ; git pull
```

- **Launch** `jupyter-notebook`
- Start with the **dr2xml_demo notebook** and next the **exercises (1 to 4)**
- There is nothing to seek by yourself, the solution is in, and the notebook cells are commented
- The objective is that you **execute the cells step by step and understand what is done**
- In order to have all onboard, the content of the two dr2xml setting files are imported in the notebooks and modified on the fly though notebook cells (python dictionary updating) and 'grep' are done on the log files to popup the result in the notebook ; but **you are encouraged to look at the two dr2xml settings files beforehand** to get familiar with their content and navigate into the input/output directories to see what dr2xml uses/produces concretely.

hands-on (notebooks)

PCform

LOGIN : stageXX

PASSWD: stageXX

Dr2xml_demo notebook
Purpose: Training notebook for a first contact with dr2xml: how to configure? how to run?

Exercice 1: Change the general configuration
Objective: Configure so as to run dr2xml for another:
(a) context/realm
(b) CMIP6 experiment
(c) model version

Exercice 2: Customize the outputs (simple way)
Objective: Try various simple methods to customize the ouputs by using the **excluded/included** keys:
(a) exclude one table
(b) exclude pairs of variable/table
(c) include only one variable from one table

Exercice 3: Customize the output (more flexible ways)
Objective: Try a more sophisticated method to customize the ouputs: the so-called **"home_data_request"**
(a) requesting additional CMIP6 standard variables, already existing in a CMIP6 Table
(b) requesting a CMIP6 standard variable but that do not exist in the predefined CMIP6 Table for the frequency we want
(c) requesting a CMIP6 standard variable but that do not exist in the predefined CMIP6 Table for the shape we want
(d) requesting a totally new variable, that is not defined in the CMIP6 standards (2 ways)
(e)-(h) a flexible way to add variables for development purpose

Exercice 4: Run a non CMIP6 experiment
Objective: configure dr2xml for a non-cmip6 experiment