Dr2xml session

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

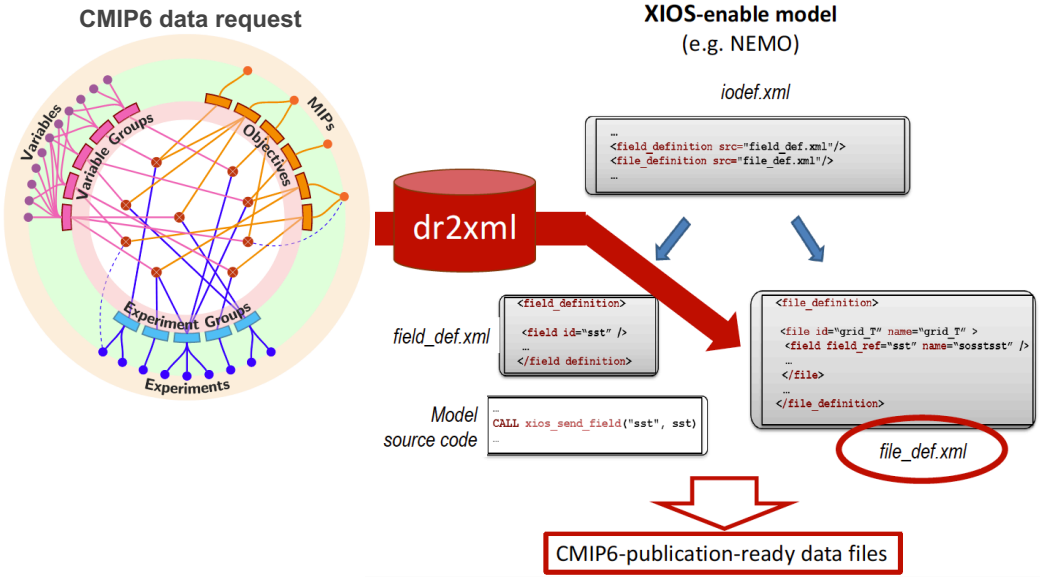
and the XIOS Team !

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
   - Basics functions
   - Customisation
   - Extended usage

a) dr2xml, what's this?

- Python tool
- XIOS file-def XML writer
  - fields and attributes (« variable » in XIOS vocab) in file
  - Automatic implementation of XIOS spatial & temporal filters
  - Automatic NetCDF file handling (naming, time-splitting, metadata, append write…)
- 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

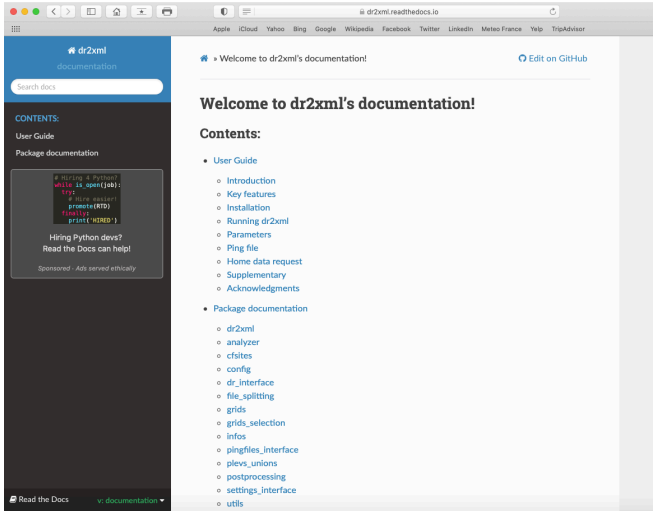


b) brief history

- 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 in 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



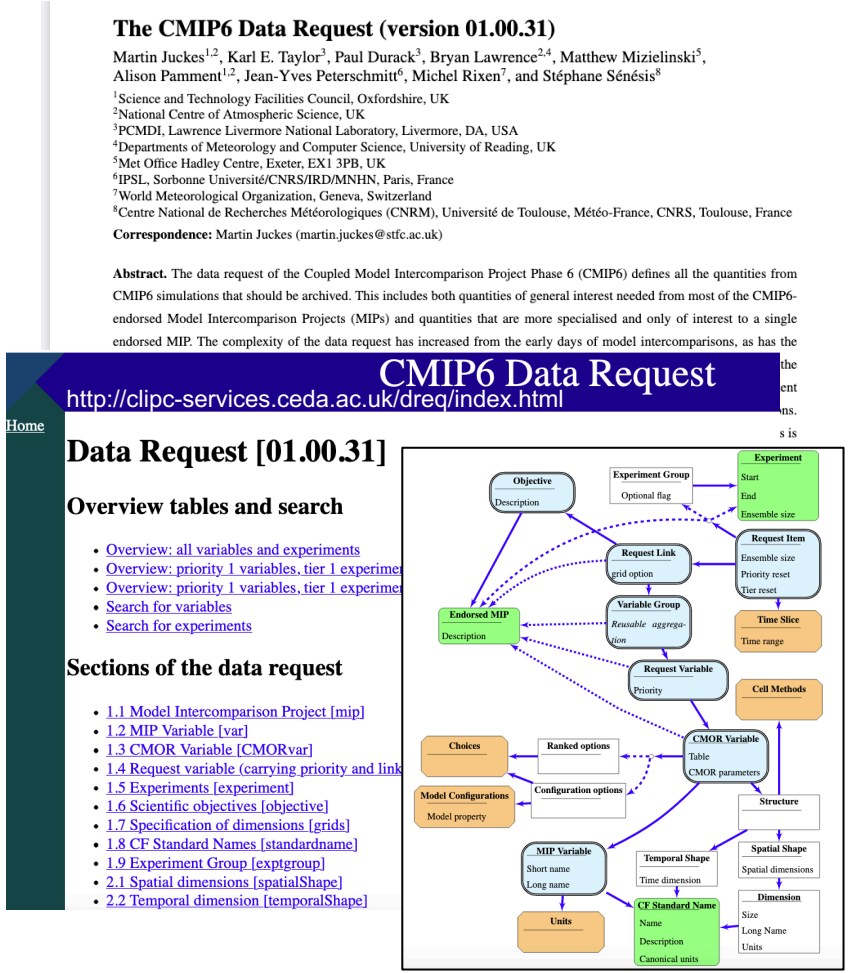
- Sources on Github: https://github.com/rigoudyg/dr2xml
- ReadtheDocs documentation : https://dr2xml.readthedocs.io/en/documentation/ *(not yet finalized)*

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



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

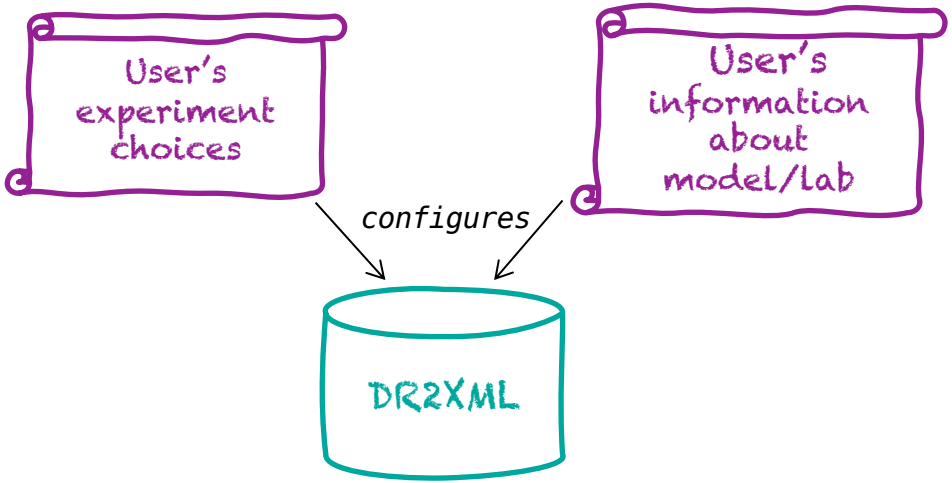
b) cautions

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

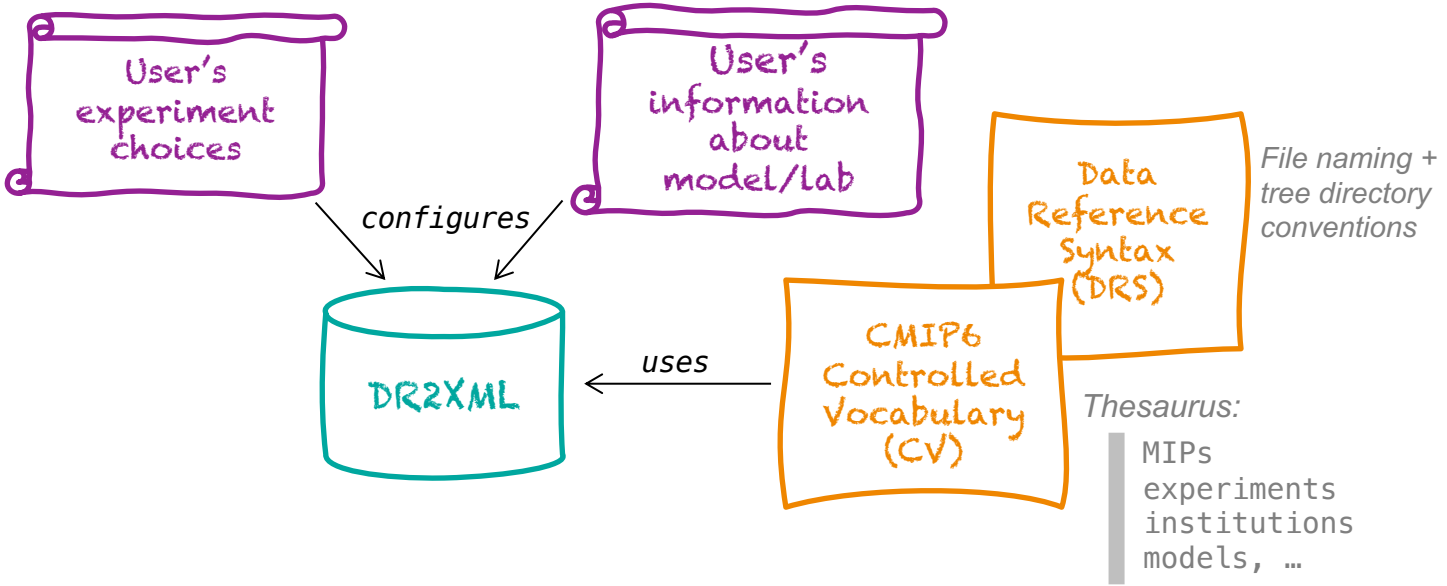
- Before running the whole simulation:
  - Dr2xml configuration must be carefully checked (a verbose log file enables to visualize the planned output variables)
  - It is highly recommended to perform a short test run and control/validate the output netCDF files, for e.g. with NCTIME and potentially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

c) simple functional scheme

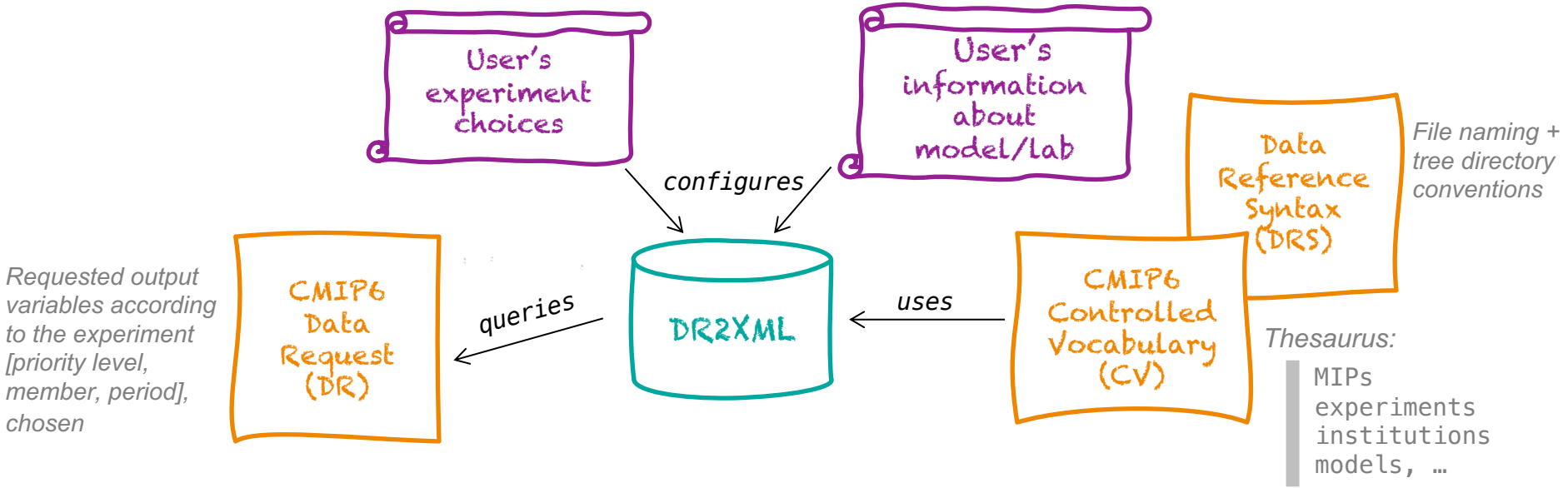
c) simple functional scheme



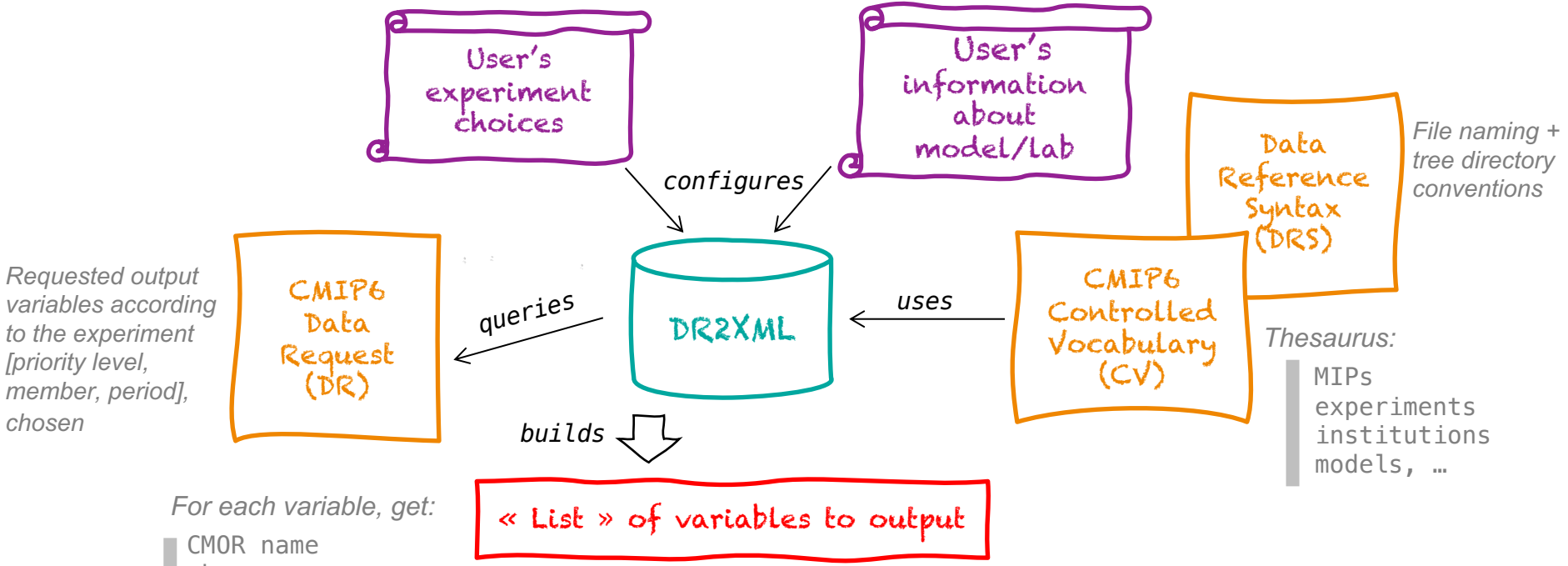
c) simple functional scheme



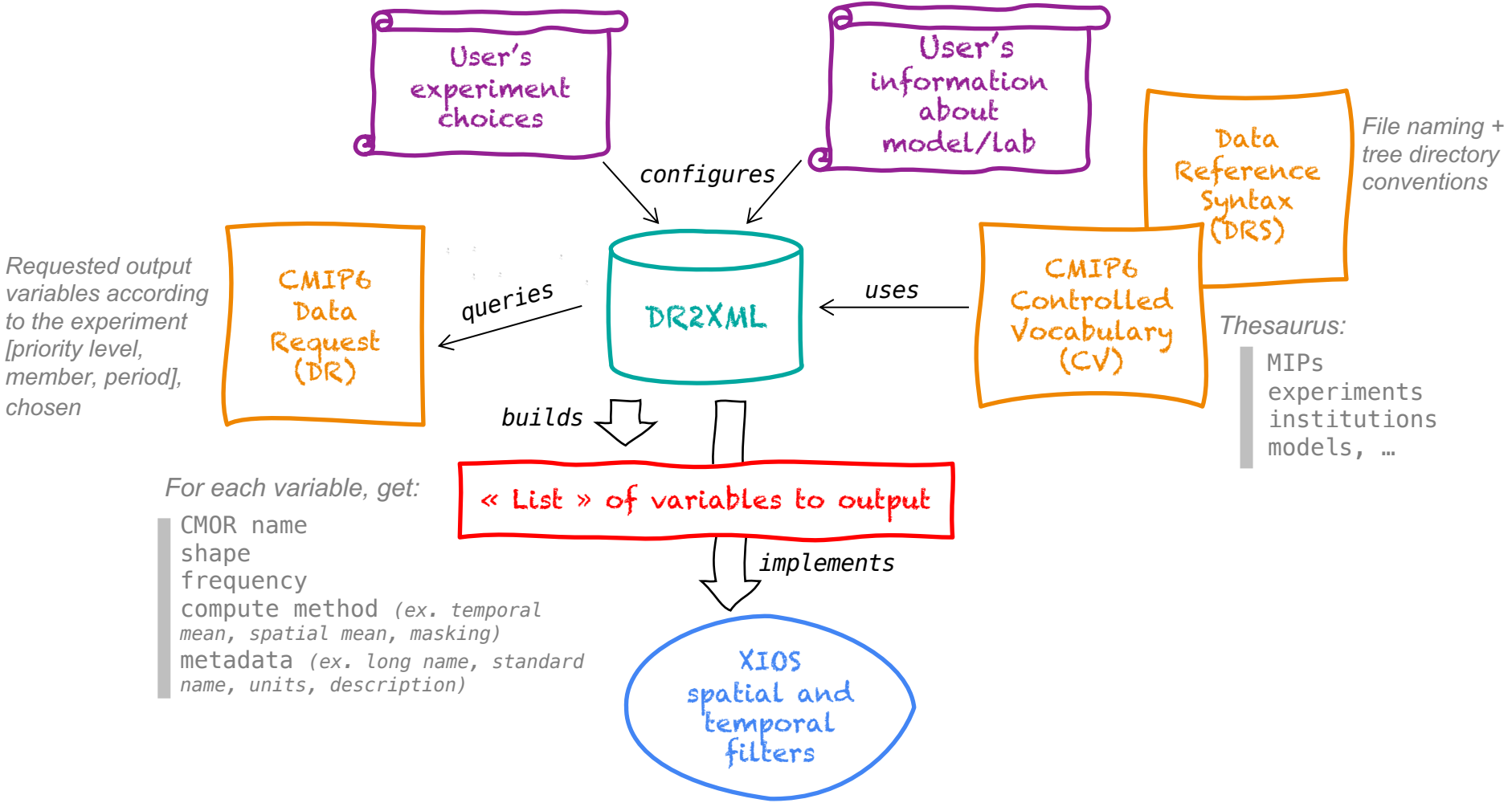
c) simple functional scheme



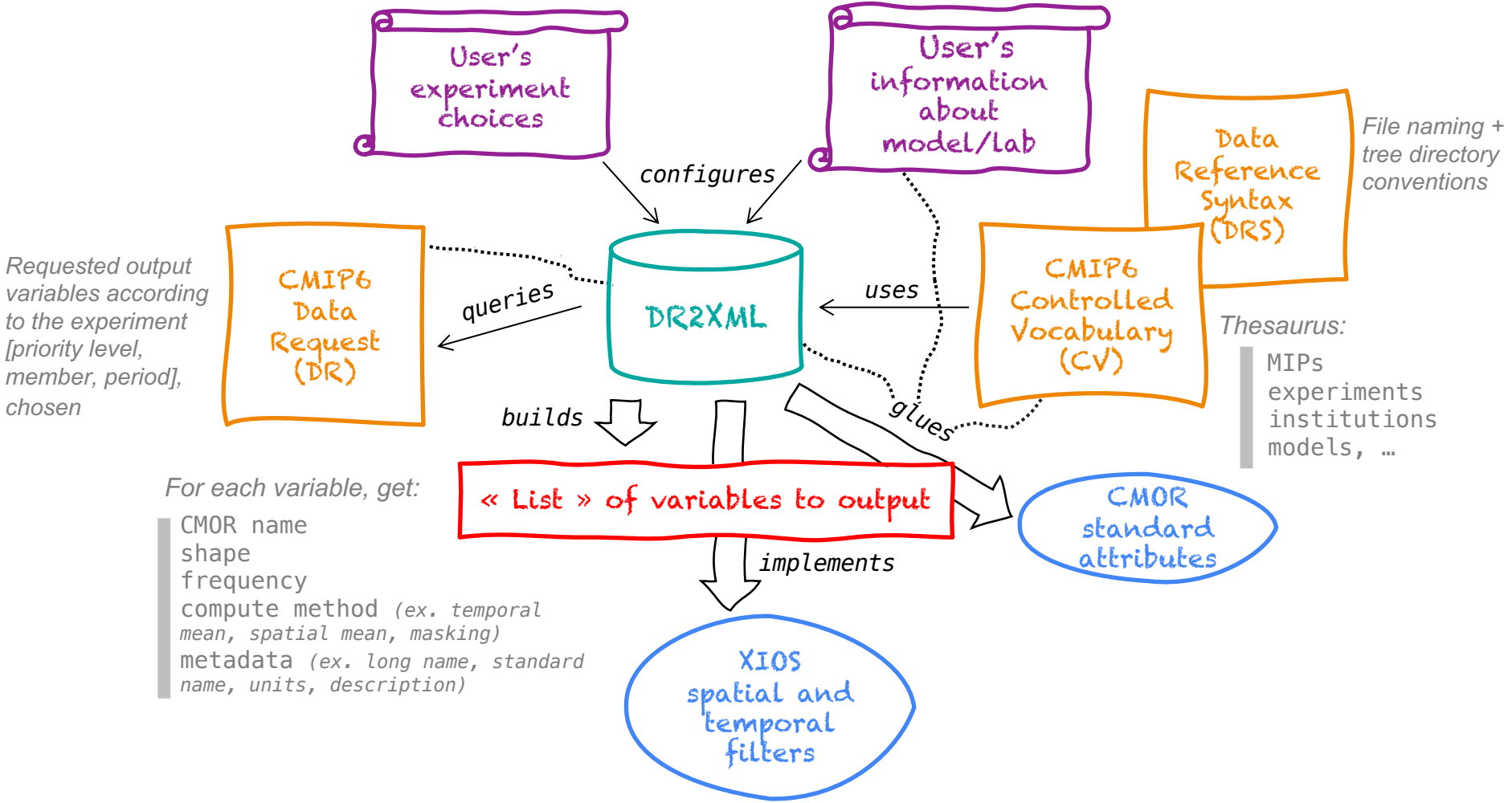
c) simple functional scheme



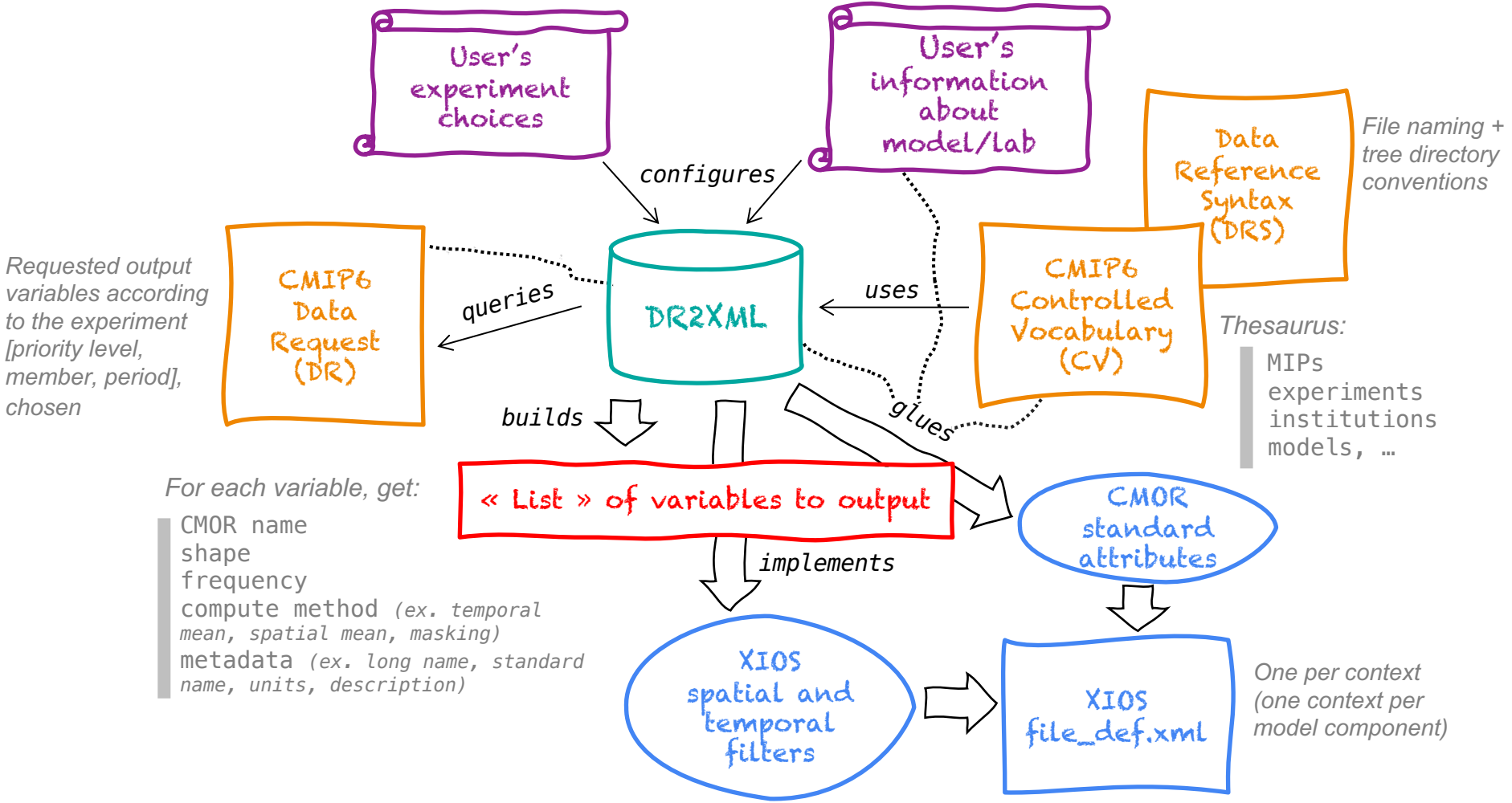
c) simple functional scheme



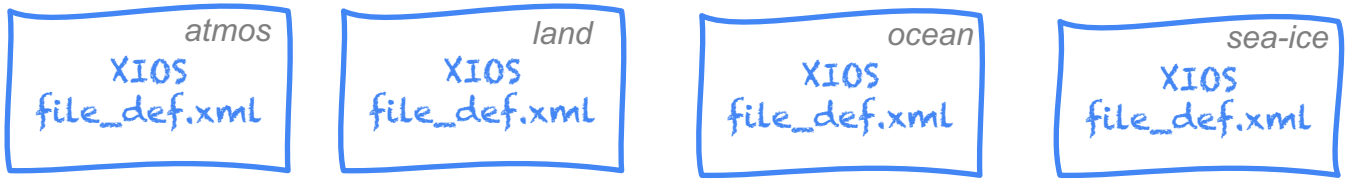
c) simple functional scheme



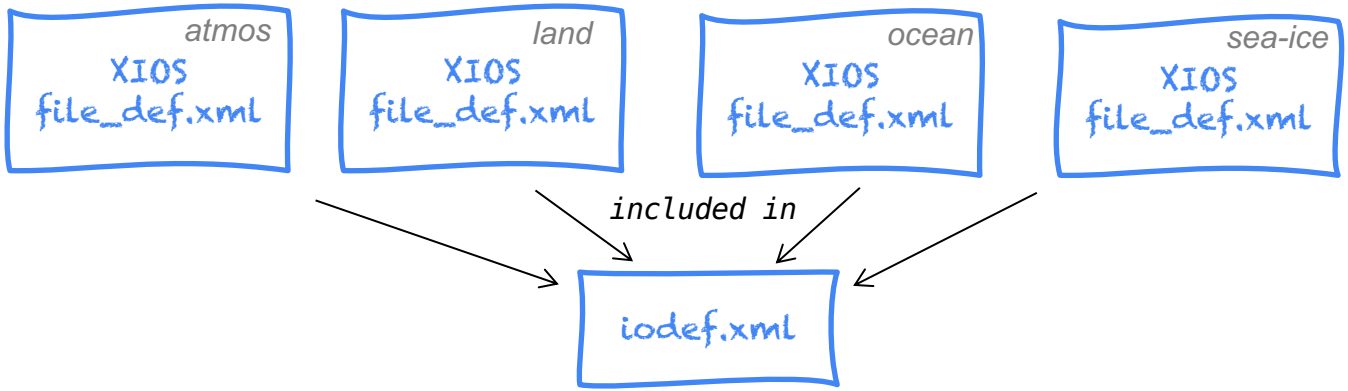
c) simple functional scheme



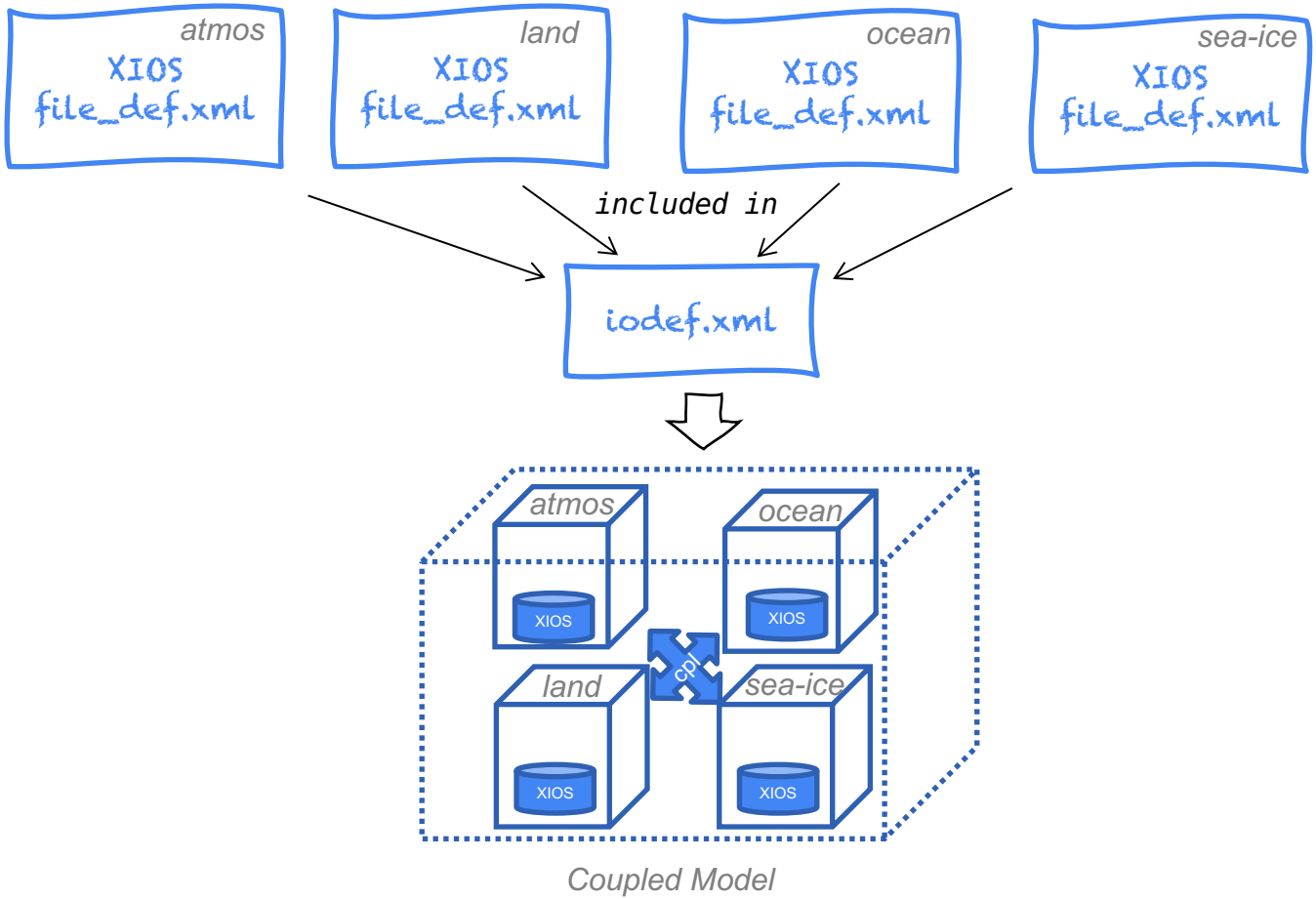
c) simple functional scheme (cont.)



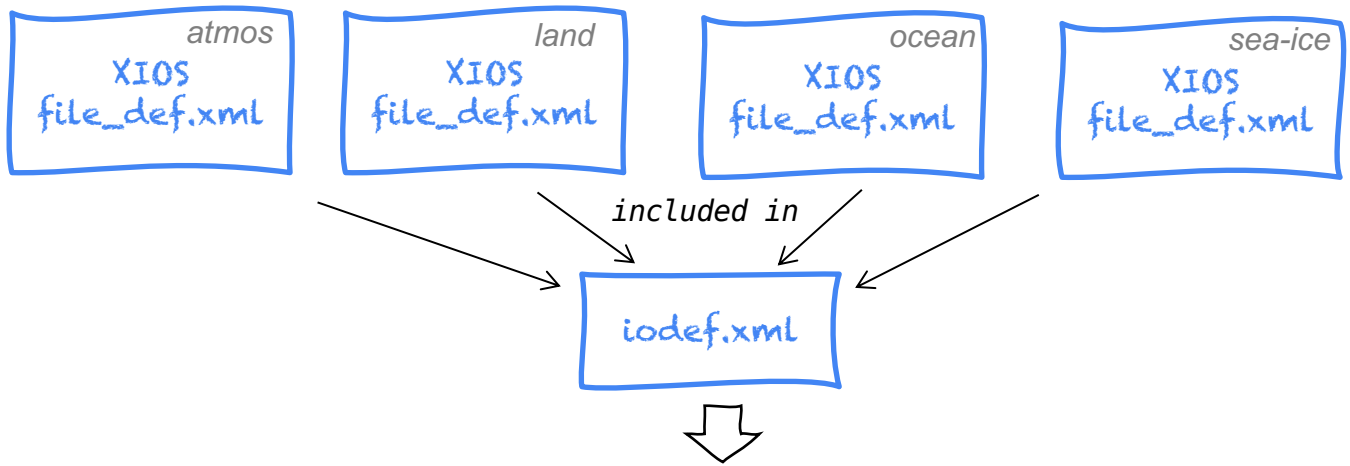
c) simple functional scheme (cont.)



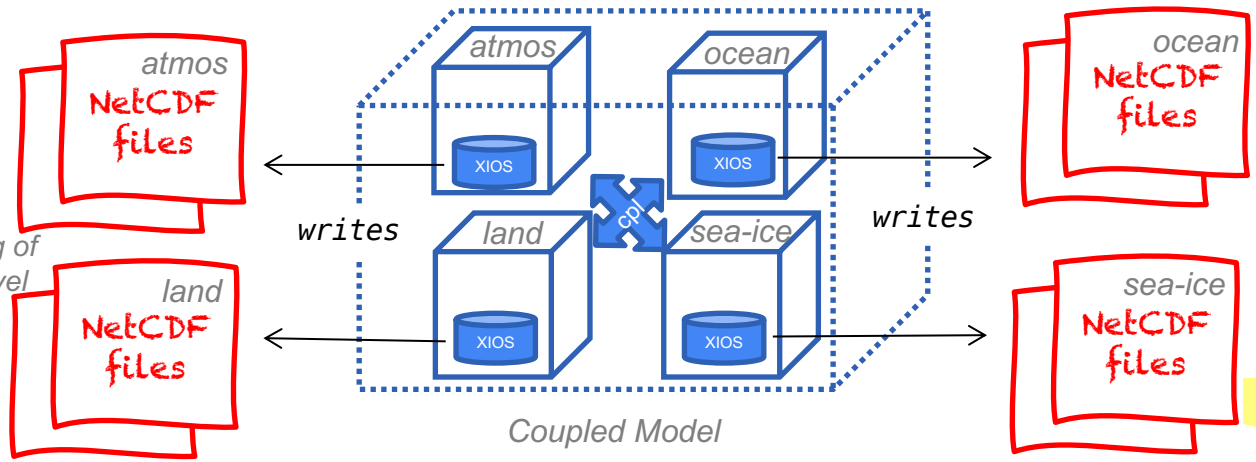
c) simple functional scheme (cont.)



c) simple functional scheme (cont.)



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

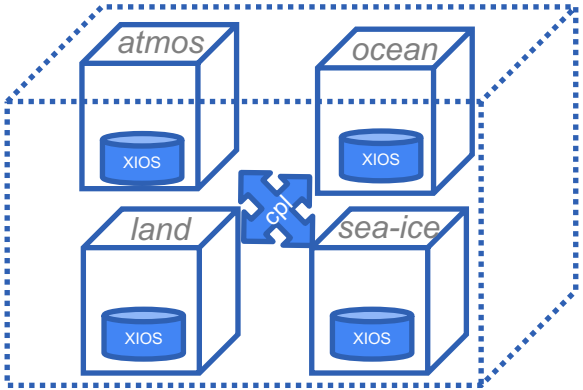


Ready to be published on ESGF.

d) the ping files

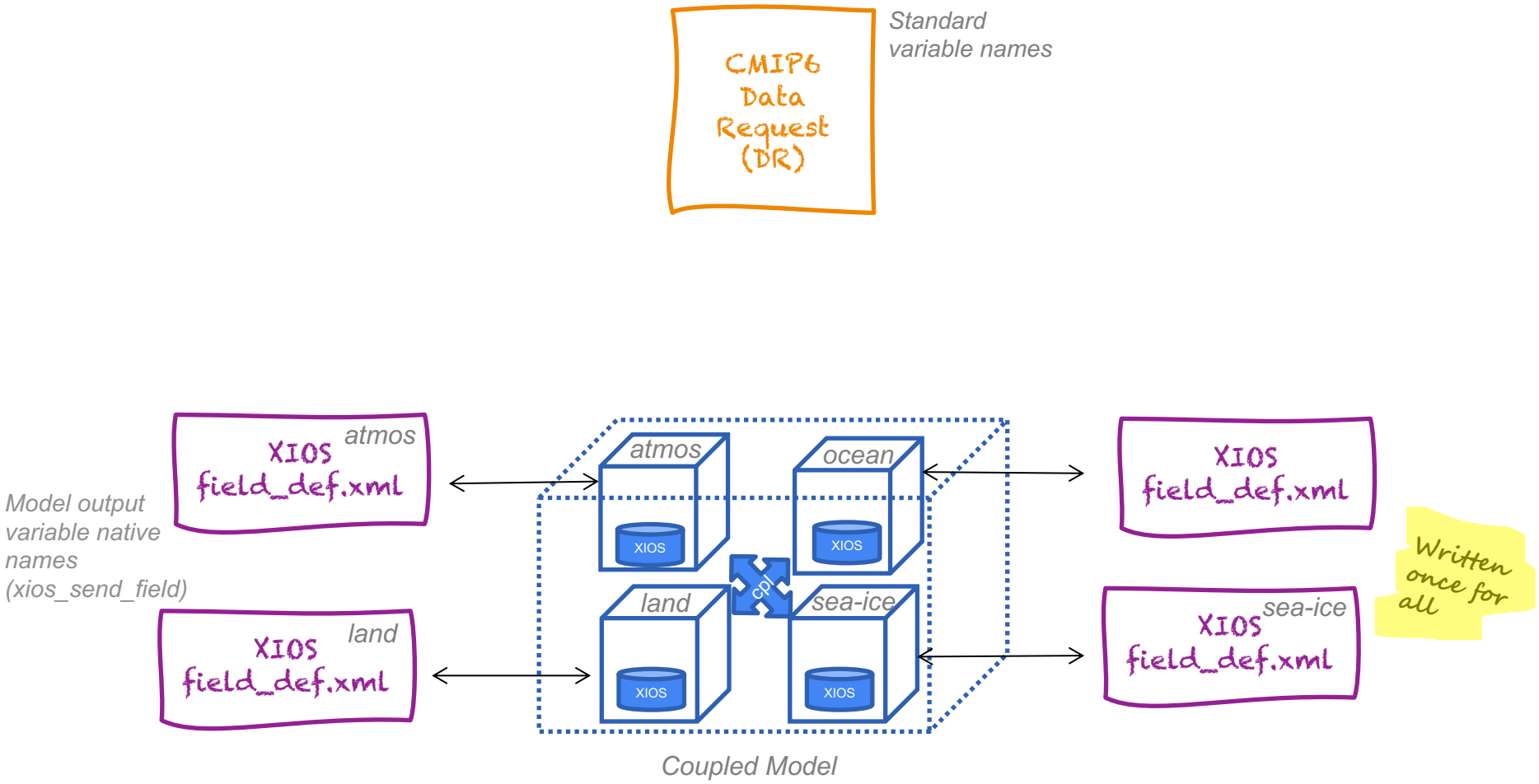
CMIP6 Data Request (DR)

Standard variable names

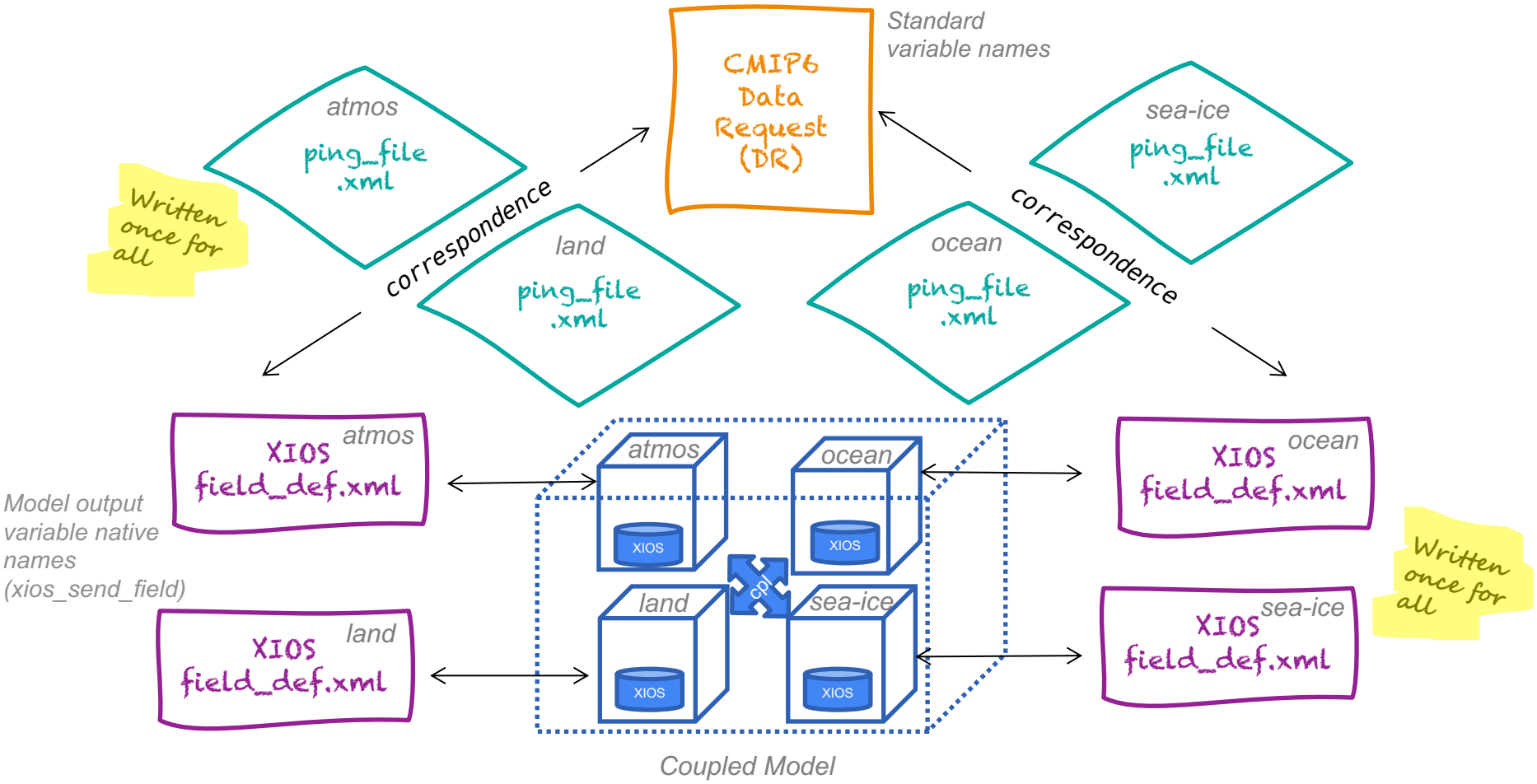


Coupled Model

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

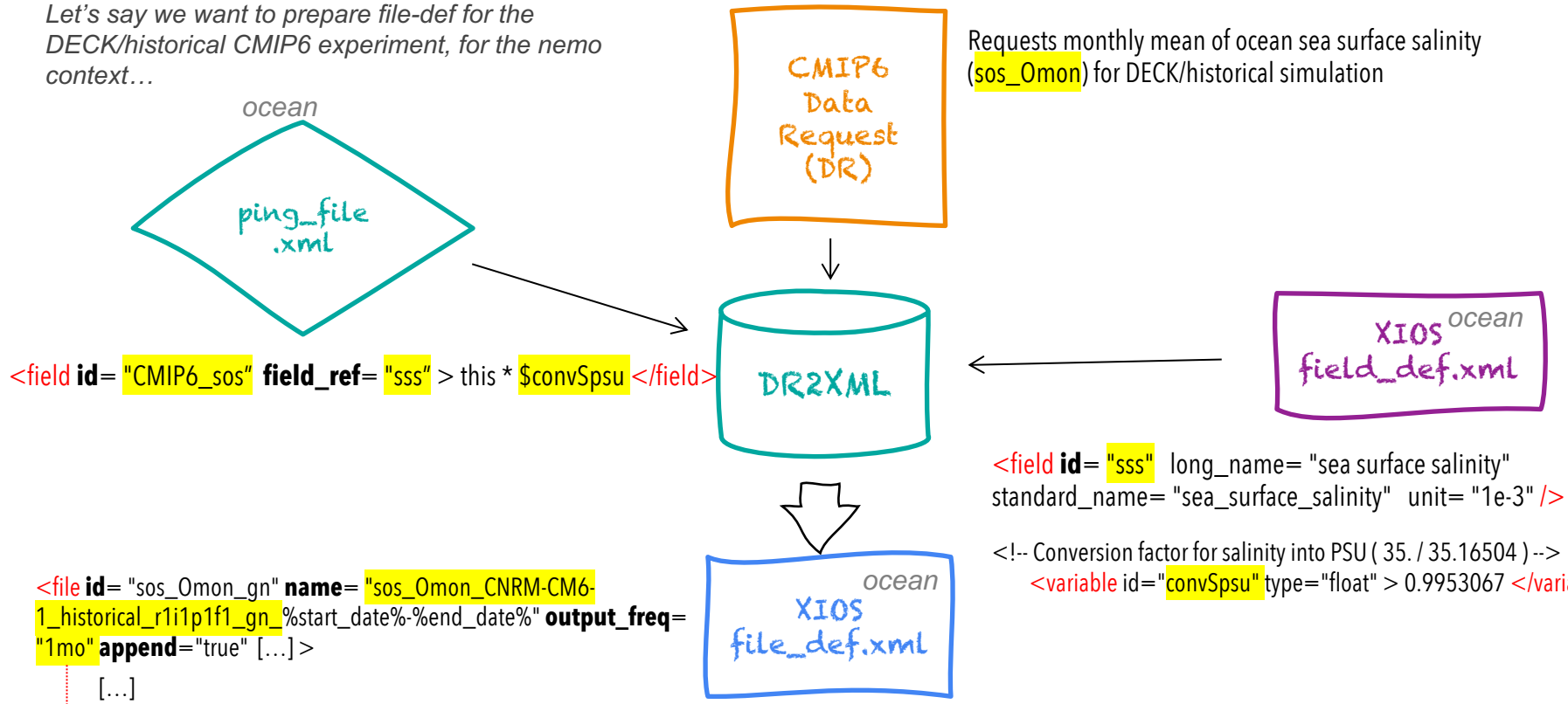
- units : 1e-3
- label : SeaSurfaceSalinity
- title : Sea Surface Salinity
- uid : sea_surface_salinity
- description : […]

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

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



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 :

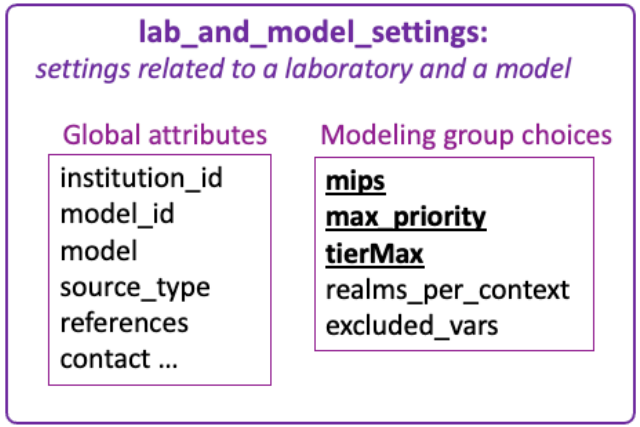
```
$ svn co http://proj.badc.rl.ac.uk/svn/exarch/CMIP6dreq/tags/01.00.32
```

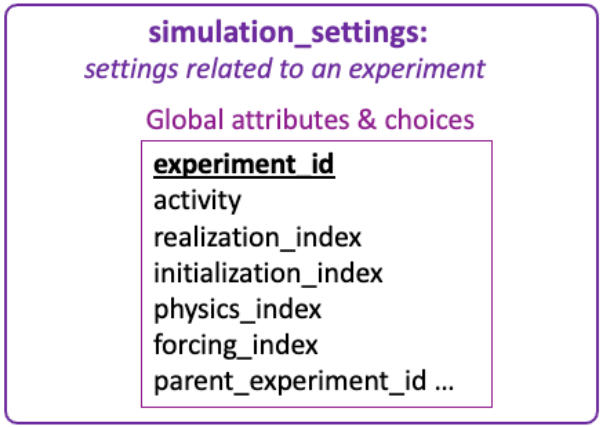
- CMIP6_CVs

```
$ 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 dreqPy import dreq # ------------------> Import Data Request package
dq = dreq.loadDreq()
from dr2xml import generate_file_defs # ------> Import dr2xml
my_cvs_path='/home/user/CMIP6_CVs/' # --------> Path to CMIP6 CVs (json files)
generate_file_defs(dq, lab_and_model_settings, simulation_settings,
                           year=2000, context='nemo, printout=True)
```

```
$ cd DR2XML/dr2xml_training

$ jupyter-notebook

   ➢ dr2xml_training
   ➢ exercice1
```

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
```

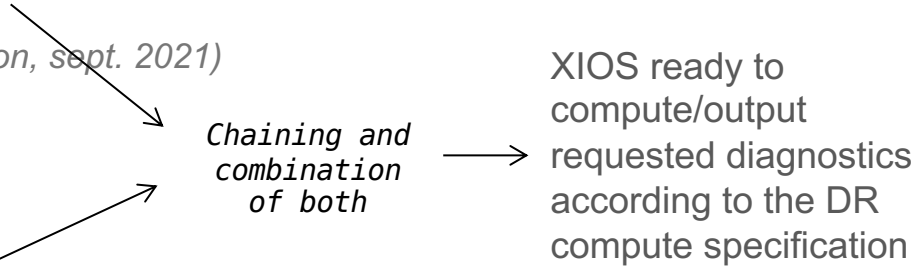
```
Some Statistics on actually written variables per variable...

------------------------
--- 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) basics functions

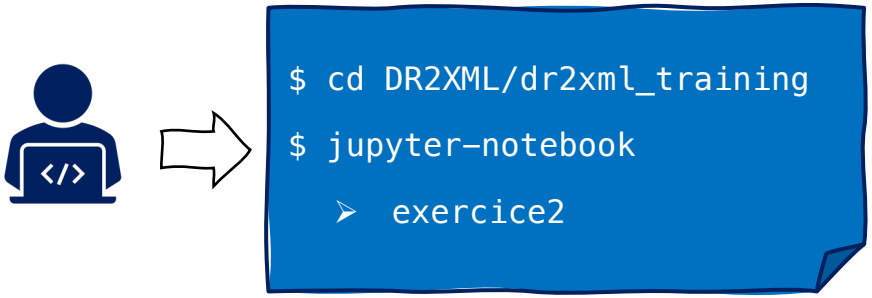
"basics functions" : **ordinated** and **optimised** combination of XIOS filters automatically implemented according to the specification of the data request.

- Spatial:
  - remapping on different grids
  - regridding on pressure levels
  - regridding on height levels *(in next version, sept. 2021)*
  - interpolation at observation sites
  - meridional/zonal means
- Temporal:
  - sampling period
  - time mean / time point
  - diurnal cycle



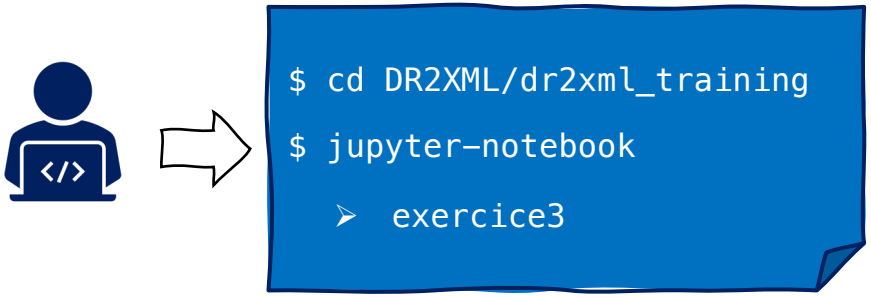
b) customisation

- 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 list 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.)

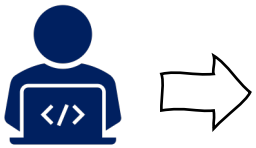
- Add methods :
  - dr2xml allows to output additional variables = the so-called "home data request"
    - "cmor" variables : defined in the Data Request but not requested for this experiment
    - "extra" variables : defined through additional tables and which are not "cmor"
    - "dev" variable : for development purpose, can be used to output variables with a minimal set of arguments
  - The "home data request" was used during CMIP6 production as a "safety net" to compensate DR potential lacks



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 more or less 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)*
  - other needs to be instructed… *(in IS-ENES3 framework)*



```
$ cd DR2XML/dr2xml_training
$ jupyter-notebook
   ➢ exercice4
```

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

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.

Thanks to all for your participation !

Any questions ?

Keep in touch !

xiostraining2021.slack.com

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