

Dr2xml session

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

and the XIOS Team !

1. Introduction

- Dr2xml, what's this?
- Brief history
- The CMIP6 Data Request

2. General features

- Utility
- Cautions
- Simple functional scheme
- The ping file

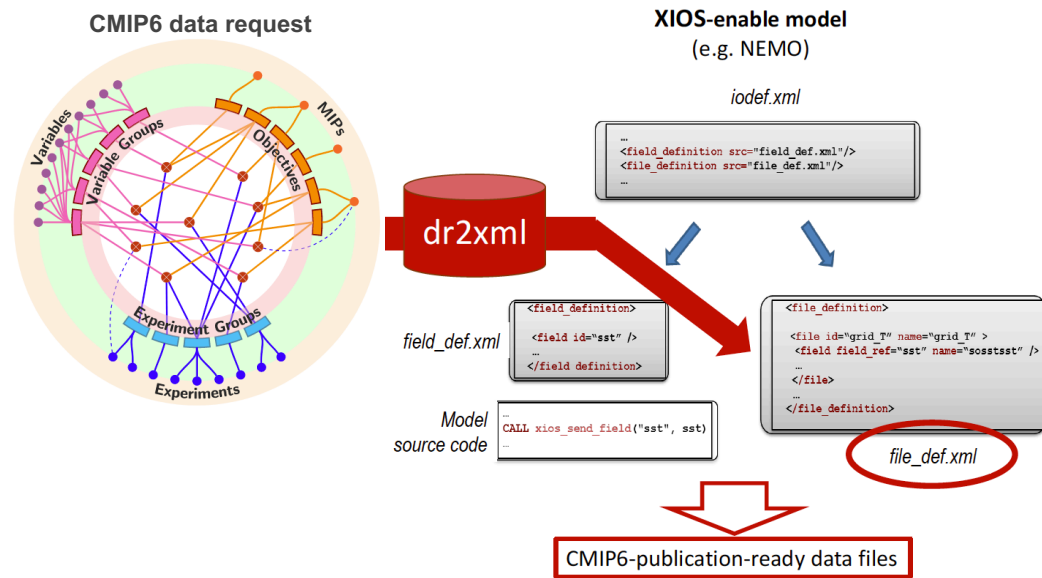
3. Usage

- Installation
- Configuration
- Execution
- Verification

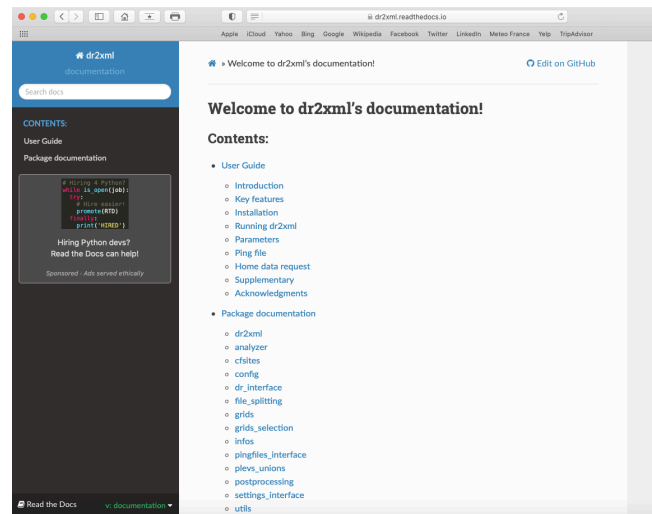
4. Functionalities

- Basics functions
- Customization
- Extended usage

- 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)
 - with a lot of mandatory attributes
- CMIP6 production
 - Systematic production of CMIP6 compliant CF-netCDF files
 - Satisfying the CMIP6 Data Request



- Born with CMIP6 (Data Request)
- Development started in sept. 2016 (S. S n si, 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 ISPL : used upstream of the modelling workflow (LibGCM)
- 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)



- Big picture :
 - Developed for CMIP6 by Martin Jukes 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 Jukes^{1,2}, Karl E. Taylor³, Paul Durack³, Bryan Lawrence^{2,4}, Matthew Mizielinski⁵, Alison Pamment^{1,2}, Jean-Yves Peterschmitt⁶, Michel Rixen⁷, and Stéphane Séné⁸

¹Science and Technology Facilities Council, Oxfordshire, UK

²National Centre of Atmospheric Science, UK

³PCMDI, Lawrence Livermore National Laboratory, Livermore, CA, USA

⁴Departments of Meteorology and Computer Science, University of Reading, UK

⁵Met Office Hadley Centre, Exeter, EX1 3PB, UK

⁶IPSL, Sorbonne Université/CNRS/IRD/MNHN, Paris, France

⁷World Meteorological Organization, Geneva, Switzerland

⁸Centre National de Recherches Météorologiques (CNRM), Université de Toulouse, Météo-France, CNRS, Toulouse, France

Correspondence: Martin Jukes (martin.jukes@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>

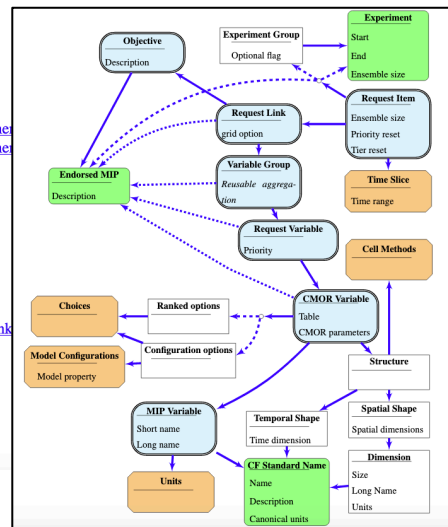
Data Request [01.00.31]

Overview tables and search

- [Overview: all variables and experiments](#)
- [Overview: priority 1 variables, tier 1 experiment](#)
- [Overview: priority 1 variables, tier 1 experiment](#)
- [Search for variables](#)
- [Search for experiments](#)

Sections of the data request

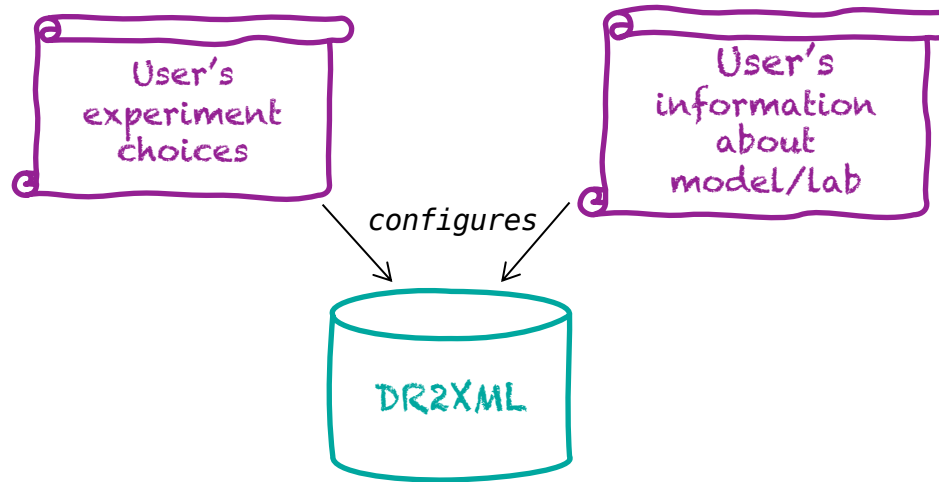
- [1.1 Model Intercomparison Project \[mip\]](#)
- [1.2 MIP Variable \[var\]](#)
- [1.3 CMOR Variable \[CMORvar\]](#)
- [1.4 Request variable \(carrying priority and link\)](#)
- [1.5 Experiments \[experiment\]](#)
- [1.6 Scientific objectives \[objective\]](#)
- [1.7 Specification of dimensions \[grids\]](#)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



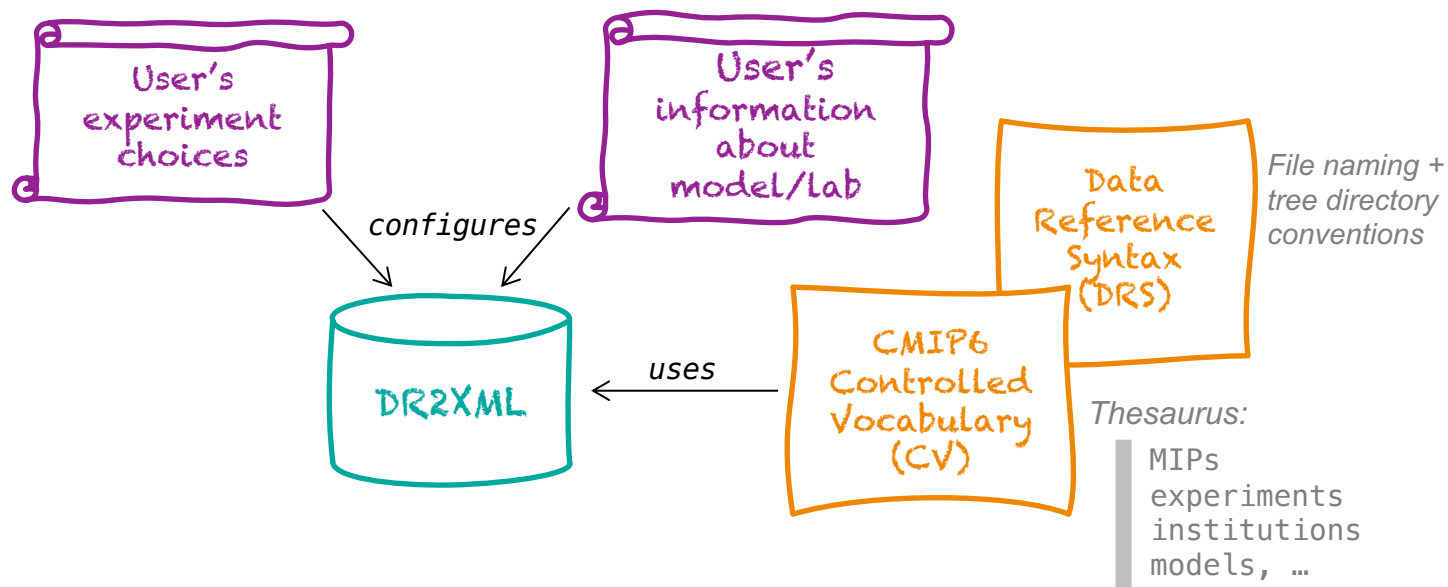
- Management of complex and various outputs, supporting:
 - All physical field shapes (*1D, 2D, 3D, 4D, time-varying or constant*)
 - All output frequencies (*yearly, monthly, daily, 6-hourly, 3-hourly, 1-hourly, sub-hour*)
 - Choice of sampling period
 - On the fly diagnostic computation (*ex. zonal means, interpolation on pressure levels, on observation sites...*)
 - Allow 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

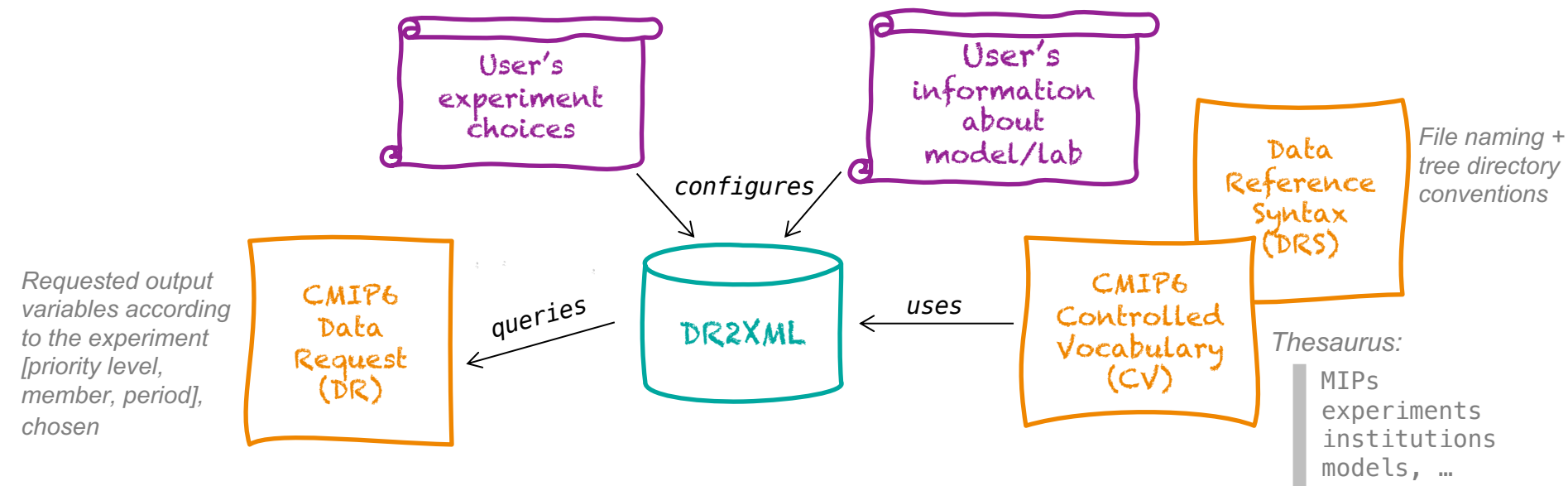
- One shot ! which means...
 - no safety net once the production started
 - => *all requested output* and *output you need* must be there !
 - => Dr2xml configuration must be carefully checked beforehand
 - a verbose log file enables to visualize the planned output variables

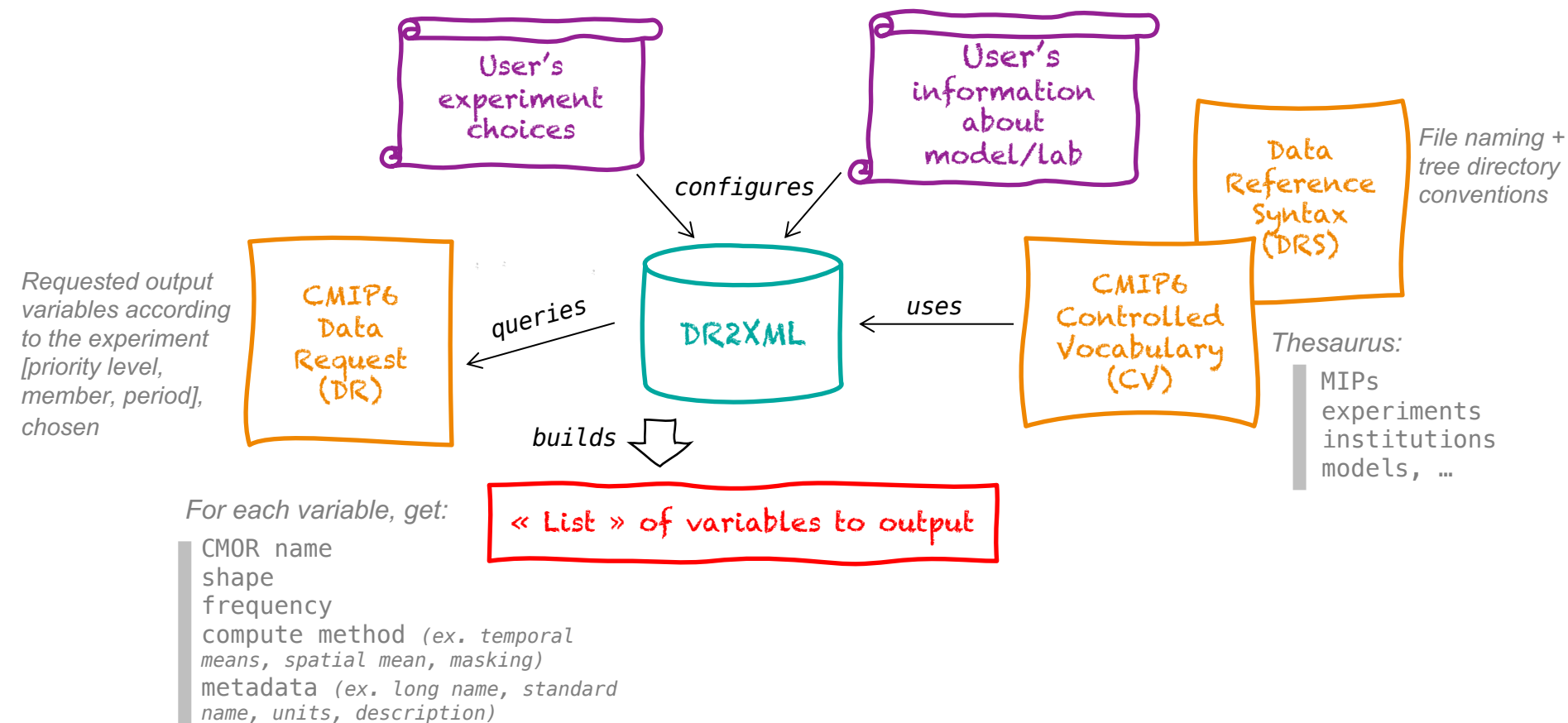




c) simple functional scheme

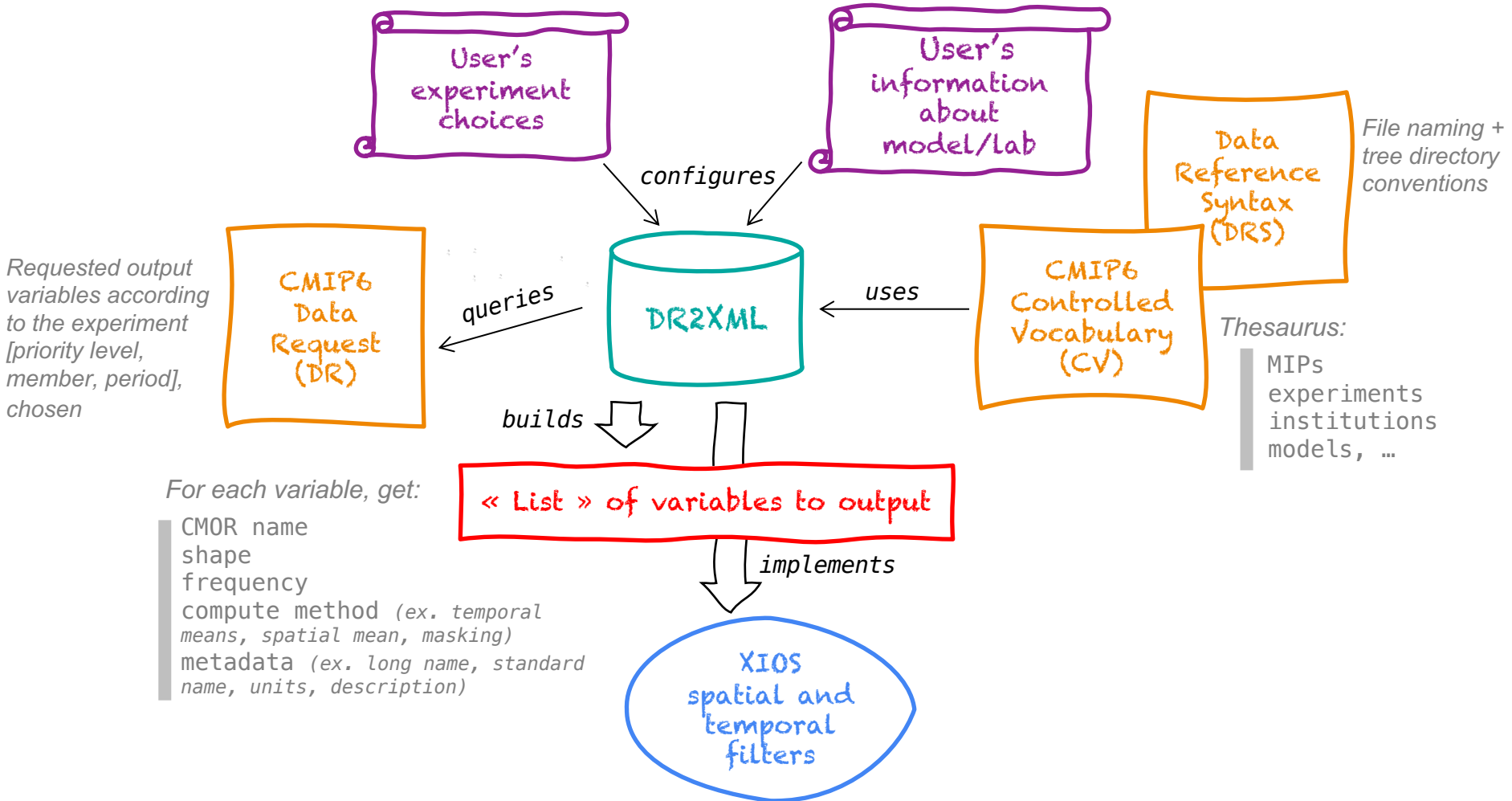






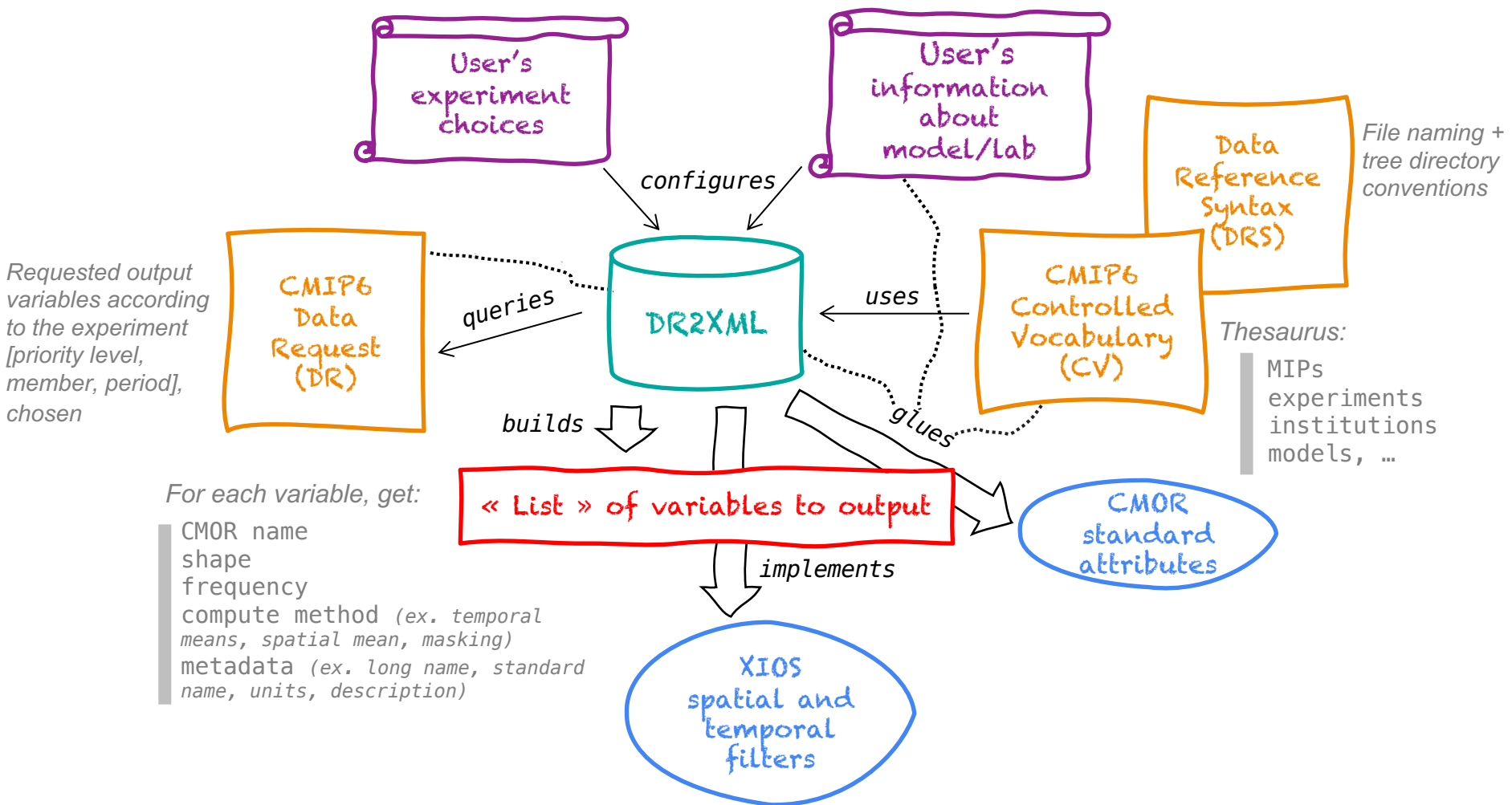
2. General features

c) simple functional scheme



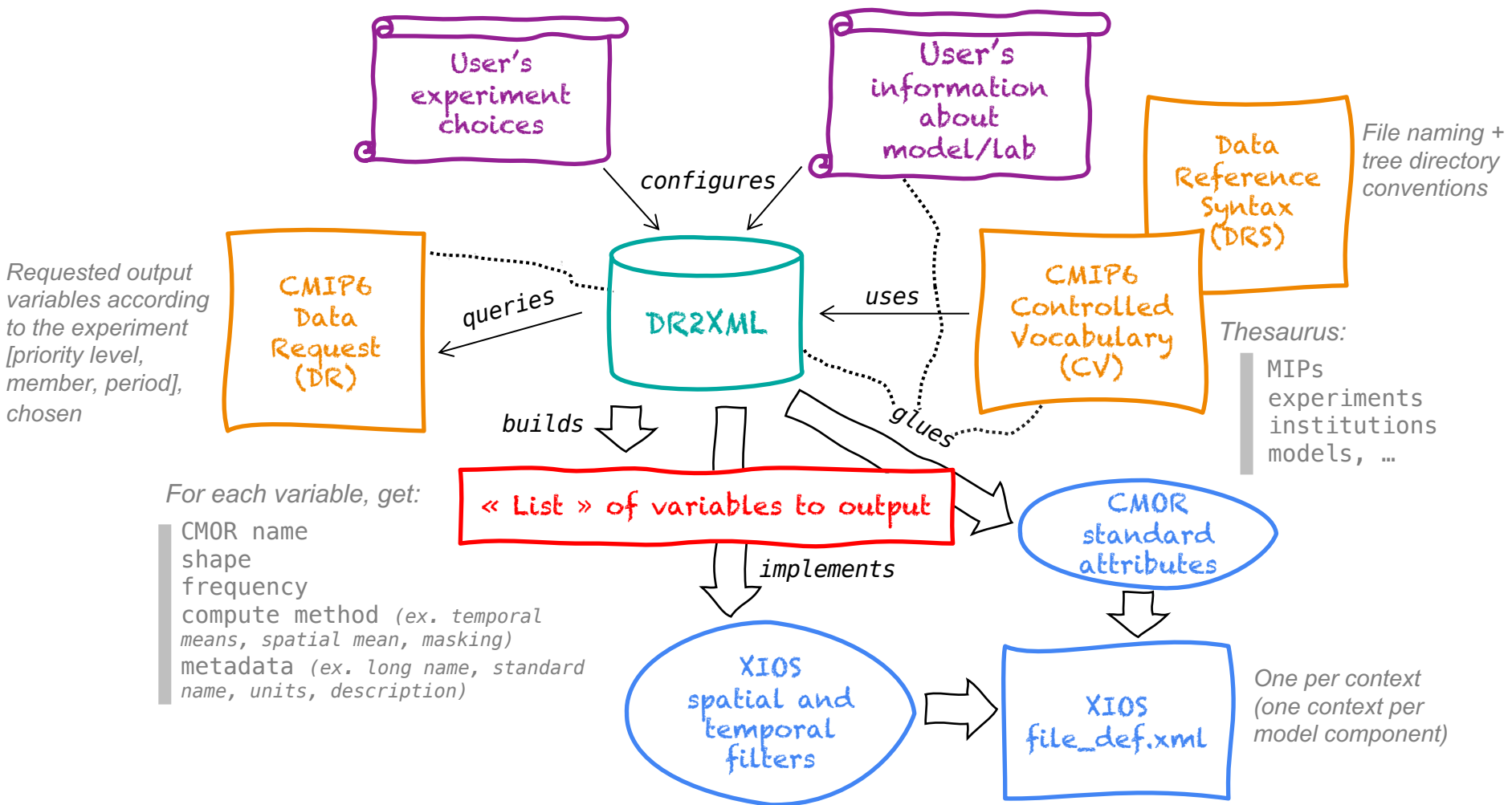
2. General features

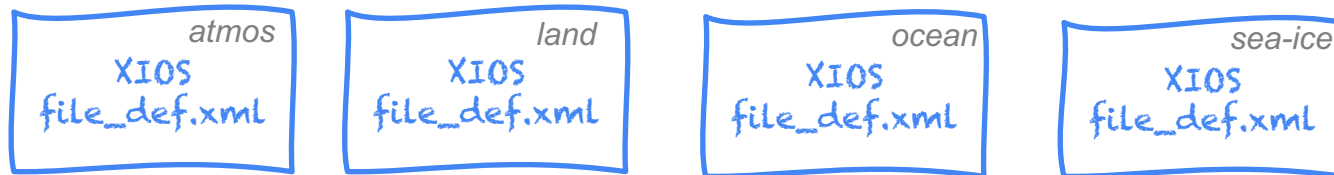
c) simple functional scheme

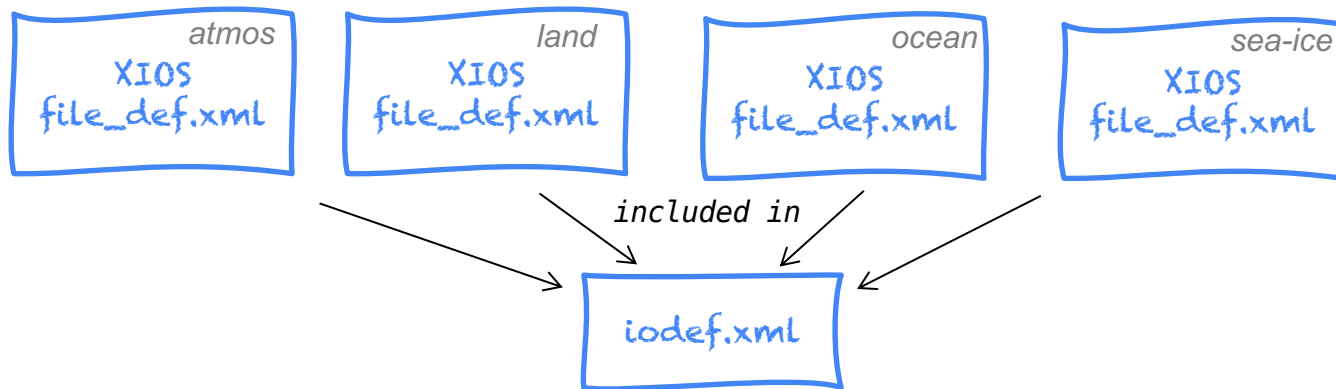


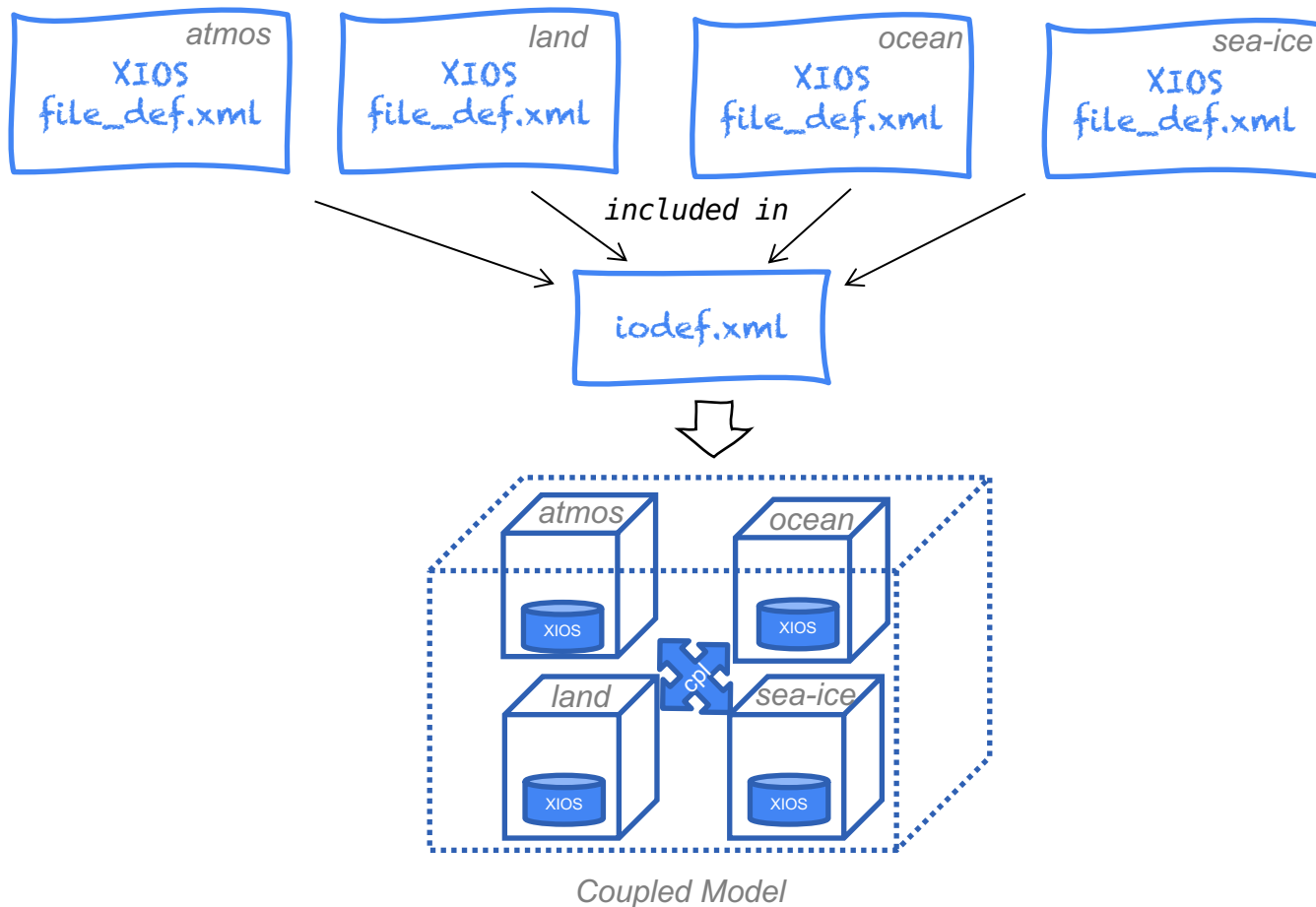
2. General features

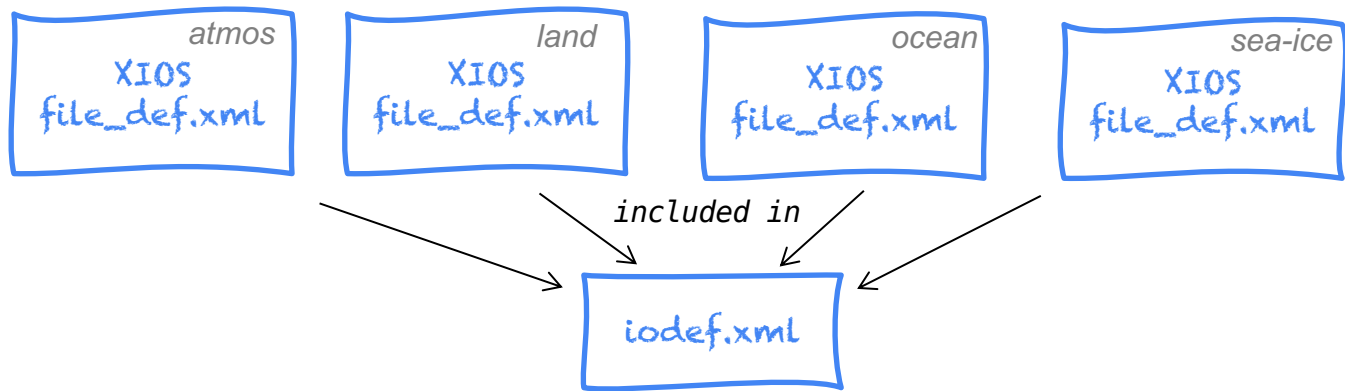
c) simple functional scheme







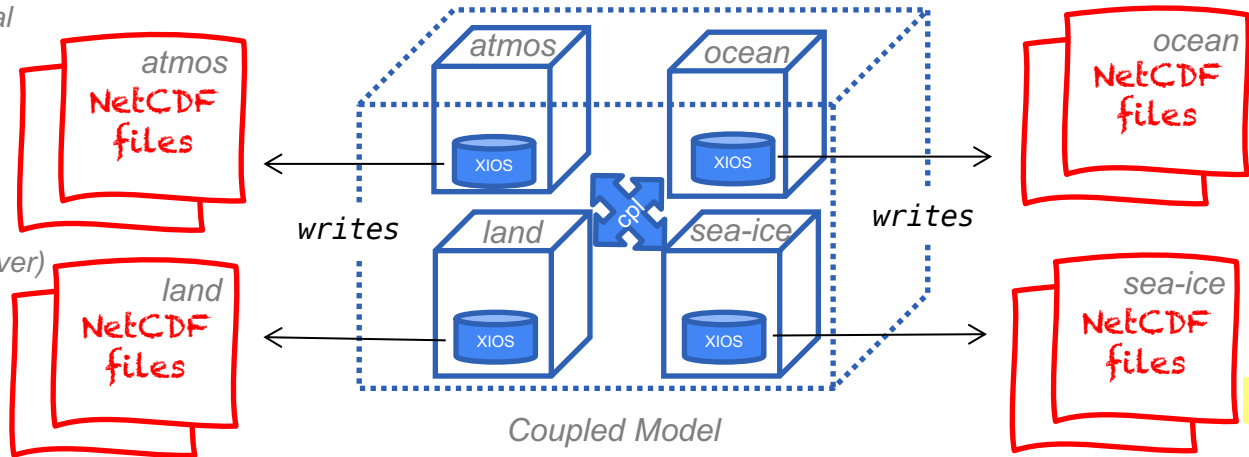




one file per physical field

append write
(-> split_freq)

parallel write
(2nd level XIOS sever)

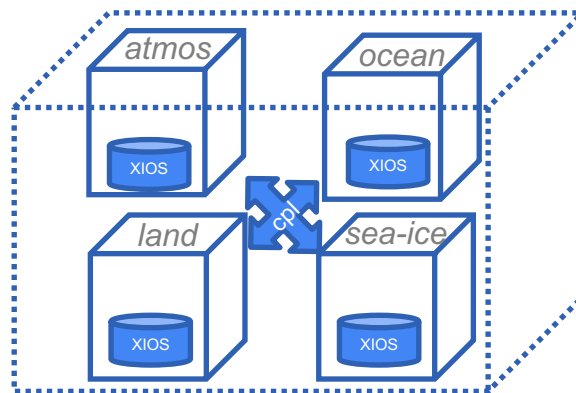


Ready to be published on ESGF.

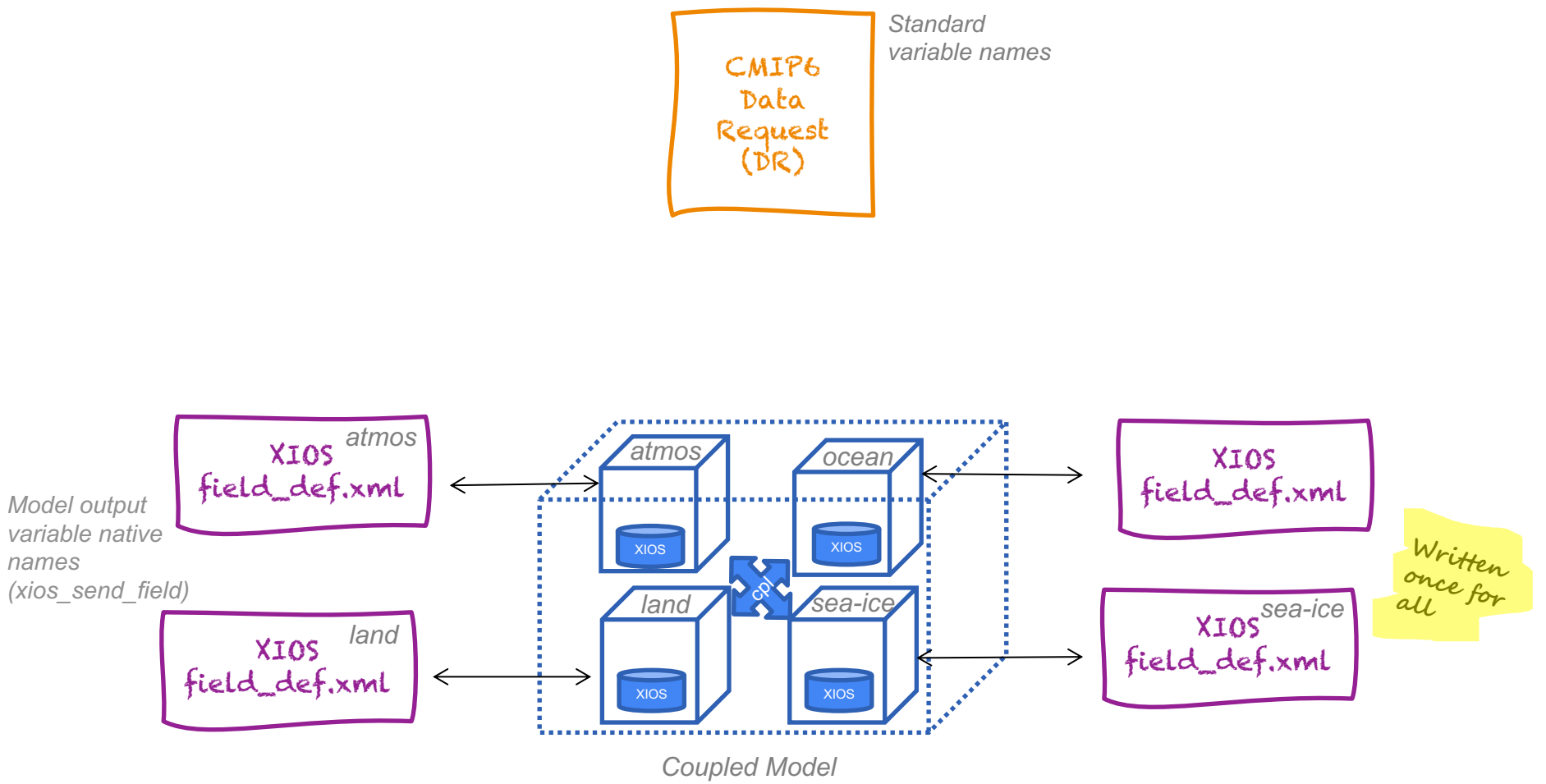
d) the ping files

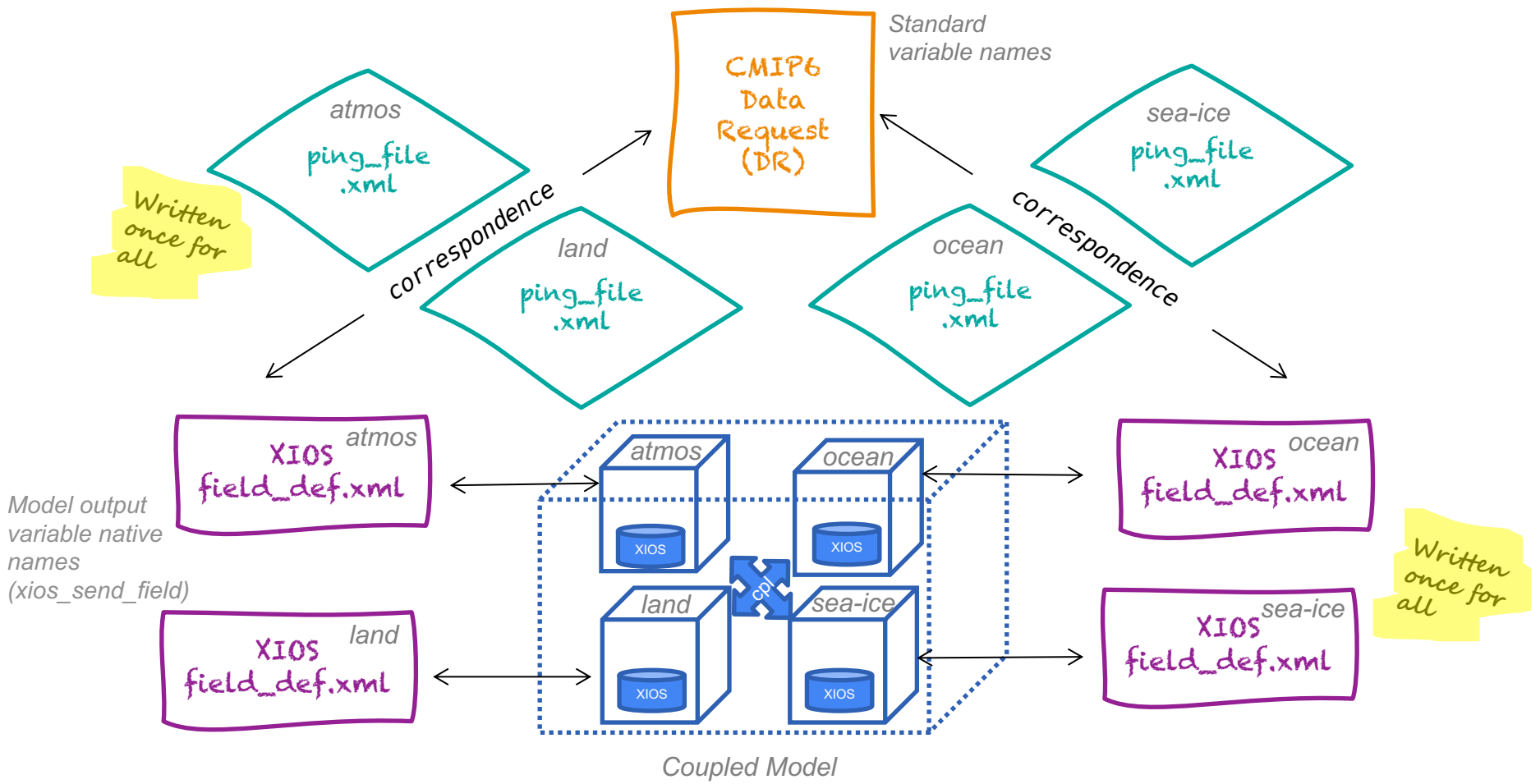


*Standard
variable names*



Coupled Model

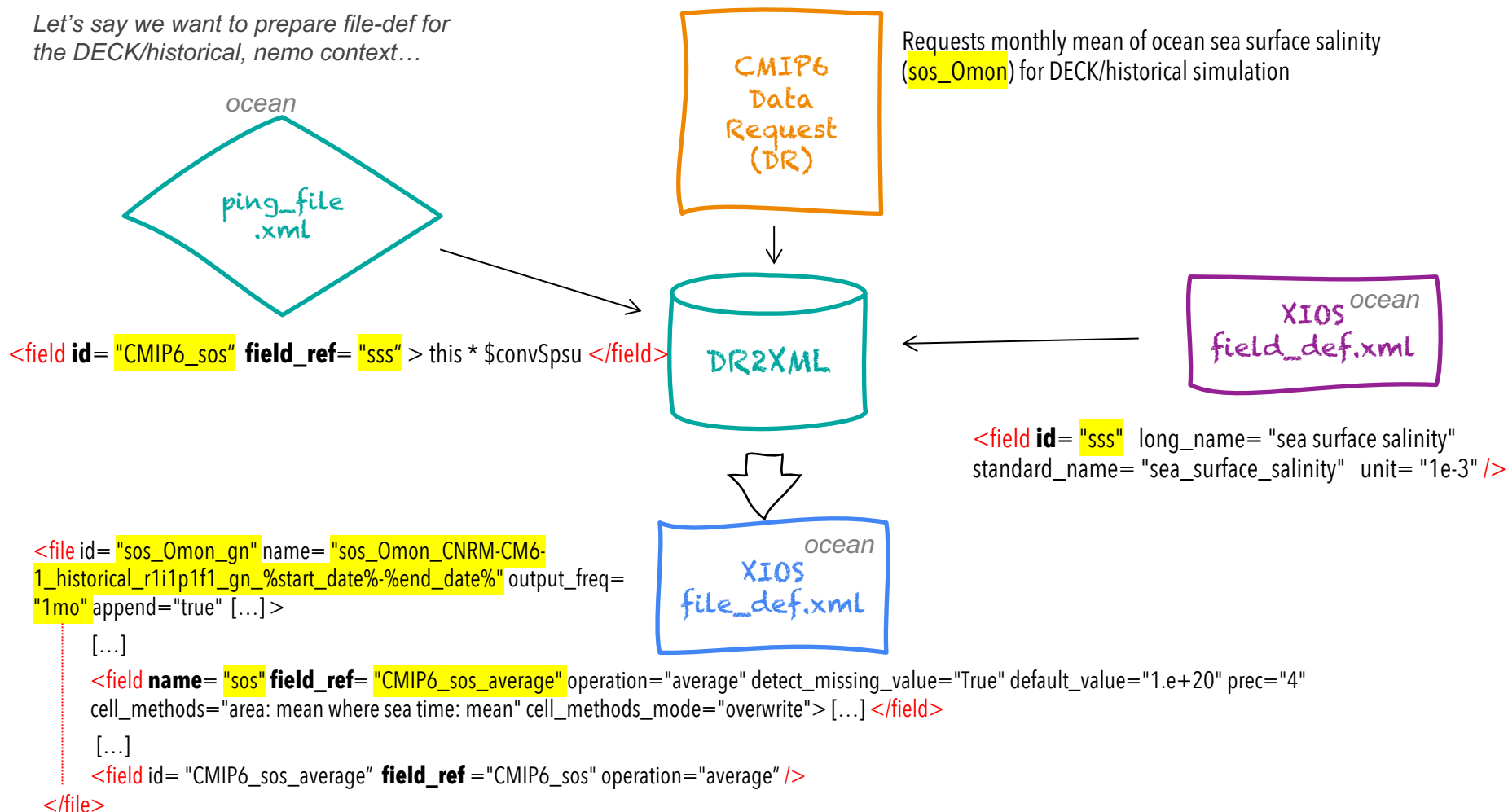




2. General features

e) "sos_Omon" example

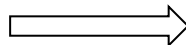
Let's say we want to prepare file-def for the DECK/historical, nemo context...



1.3 CMOR Variable: [sos] Sea Surface Salinity

→ [1.3 CMOR Variable section index](#)

- 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.
- [...]



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

→ [2.3 Dimensions and related information section index](#)

- 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.
- [...]

- `Dr2xml` (+ `xlsxwriter`, `six`)

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

- `dreqPy` (the CMIP6 Data Request)

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

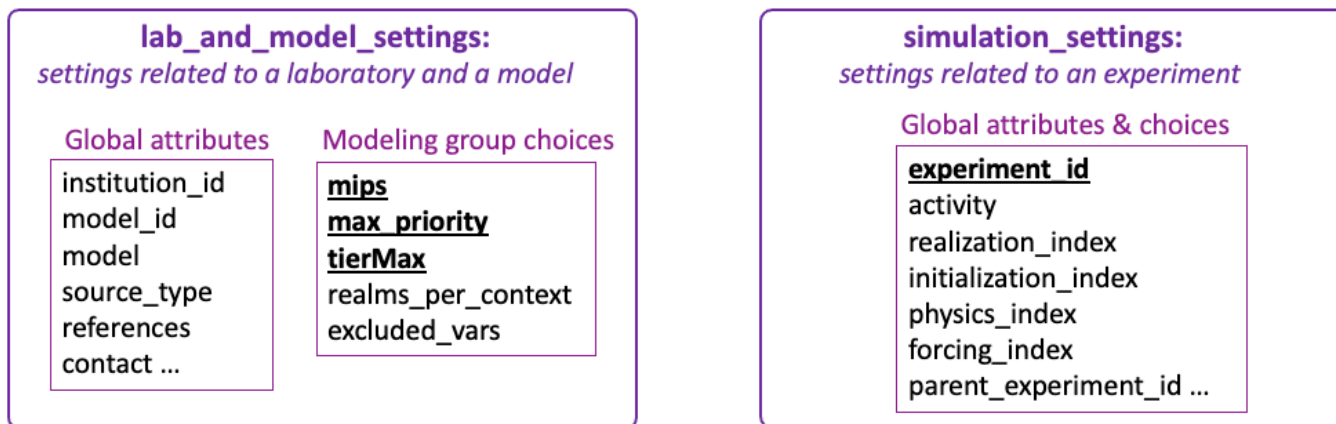
or:

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

- `CMIP6_CVs`

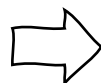
```
$ git clone https://github.com/WCRP-CMIP/CMIP6_CVs
```

- 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)

```
from dreqPy import dreq # -----> Import dataRequest 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_TRAINING/dr2xml_training
$ jupyter-notebook
➤ dr2xml_training
➤ exercice1
```

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-0      ----- ---  10
                                Prim0day P1  1 : [u'so']
day      XY-0      ----- ---  1
```

Some Statistics on actually written variables per variable...

```
-----
--- VARNAME: bigthetao : Sea Water Conservative Temperature
-----
```

```
* mon_Omon_XY-0_1
```

```
-----
--- VARNAME: so : Sea Water Salinity
-----
```

```
* mon_Omon_XY-0_1
* day_Prim0day_XY-0_1.0
```

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

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

- **Spatial:**

- regridding on pressure levels
- regridding on height levels (*in next version, sept. 2021*)
- interpolation at observation sites
- meridional/zonal means
- diurnal cycle

- **Temporal:**

- sampling period
- time mean / time point (=instant)

*Chaining and
combination
of both*

XIOS ready to
compute/output
requested diagnostics
according to the DR
compute specification

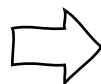
- Filtering options :

- **curation** : can filter out variables that are defined twice or that are too narrow in the DR (can be customised) (???)
- **selection** : basic filtering choosing the maximum priority level to output
- **exclusion** : user can also provide list of excluded...
 - variables (var)
 - tables (tbl)
 - pairs (var,tbl)
 - spatial shapes
- **metadata filtering** : user can take the control on attributes to write in the NetCDF files



```
$ cd DR2XML_TRAINING/dr2xml_training
$ jupyter-notebook
➤ exercice2
```

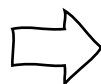
- Add methods :
 - dr2xml allows to output additional variables = the so-called “home data request”
 - “cmor” variables : defined in the Data Request but not requested for this experiment
 - “extra” variables : defined through additional tables and which are not “cmor”
 - “dev” variable : for development 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



```
$ cd DR2XML_TRAINING/dr2xml_training
$ jupyter-notebook
➤ exercice3
```

- 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 its own simulation, but conforming to the DR of a given CMIP6 experiment (→ be sure to have the same output variables)
 - 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_TRAINING/dr2xml_training
$ jupyter-notebook
➤ exercice4
```


- **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
- By dynamically analysing the simulated period (and adapting output configuration consequently), prevents from stopping/restarting the simulation
- 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