Technical guide to run CANARI OI in ALADIN

Gergely Bölöni and Alena Trojáková

Outline

Introduction

Ingredients

Prepare observations

Run Canari OI (ALADIN conf. 701)

Installation

Script example

Introduction

CANARI: OI analysis for ARPEGE/ALADIN (conf. 701)

$$\mathbf{x_a} = \mathbf{x_b} + \mathbf{B}\mathbf{H}^{\mathbf{T}}\left(\mathbf{H}\mathbf{B}\mathbf{H}^{\mathbf{T}} + \mathbf{R}\right)^{-1}\left(\mathbf{y} - H\,\mathbf{x_b}\right)$$

- atmospheric analysis

- surface (soil) analysis

Surface analysis:

T_{2m} obs
RH_{2m} obs

T_{soil}, $T_{deepsoil}$ analysis
W_{soil}, $W_{deepsoil}$ analysis

Introduction

CANARI: OI analysis for ARPEGE/ALADIN (conf. 701)

$$\mathbf{x_a} = \mathbf{x_b} + \mathbf{B}\mathbf{H}^{\mathbf{T}}\left(\mathbf{H}\mathbf{B}\mathbf{H}^{\mathbf{T}} + \mathbf{R}\right)^{-1}\left(\mathbf{y} - H\,\mathbf{x_b}\right)$$

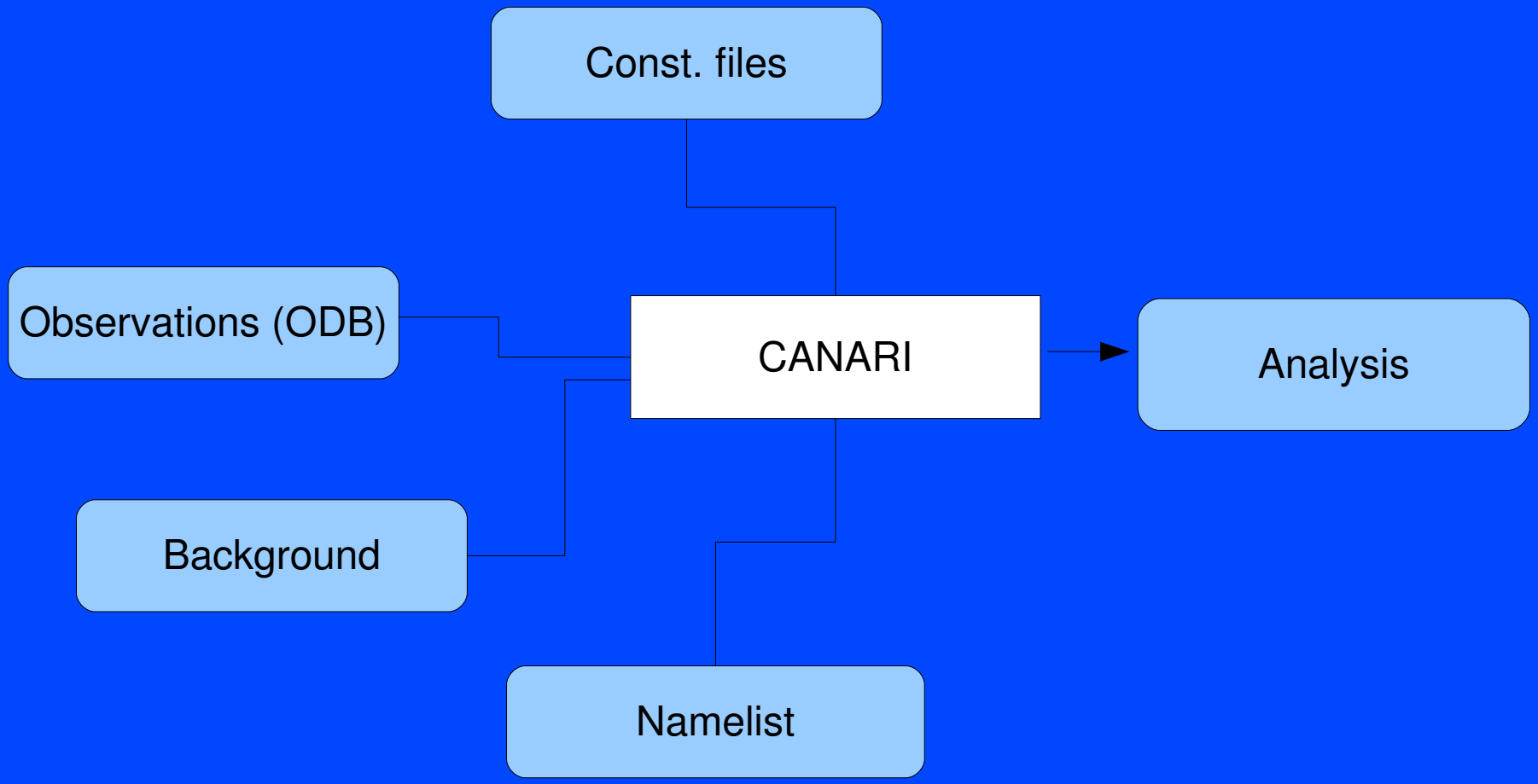
- atmospheric analysis

- surface (soil) analysis

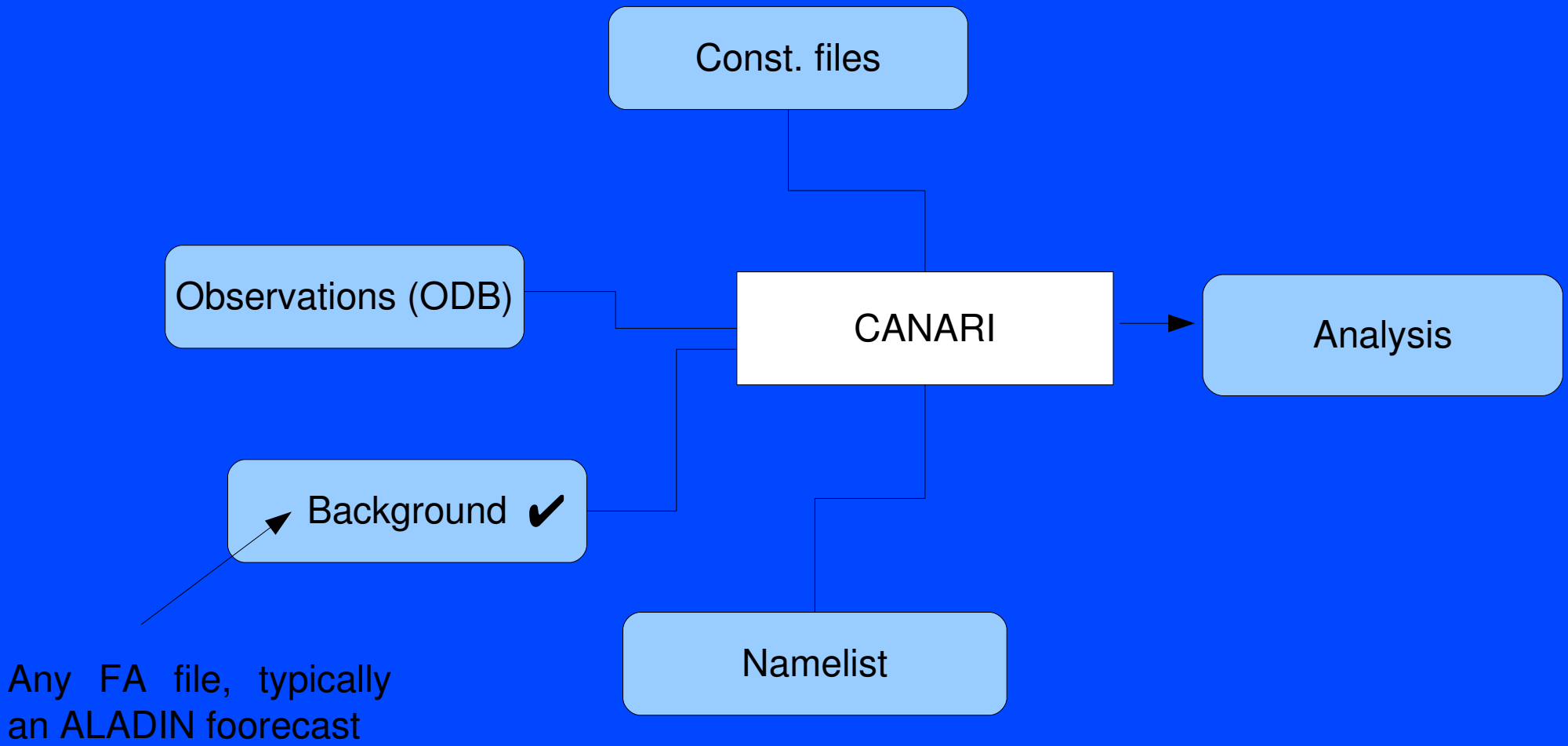
Surface analysis:

T_{2m} obs
RH_{2m} obs
→ T_{soil}, $T_{deepsoil}$ analysis
W_{soil}, $W_{deepsoil}$ analysis

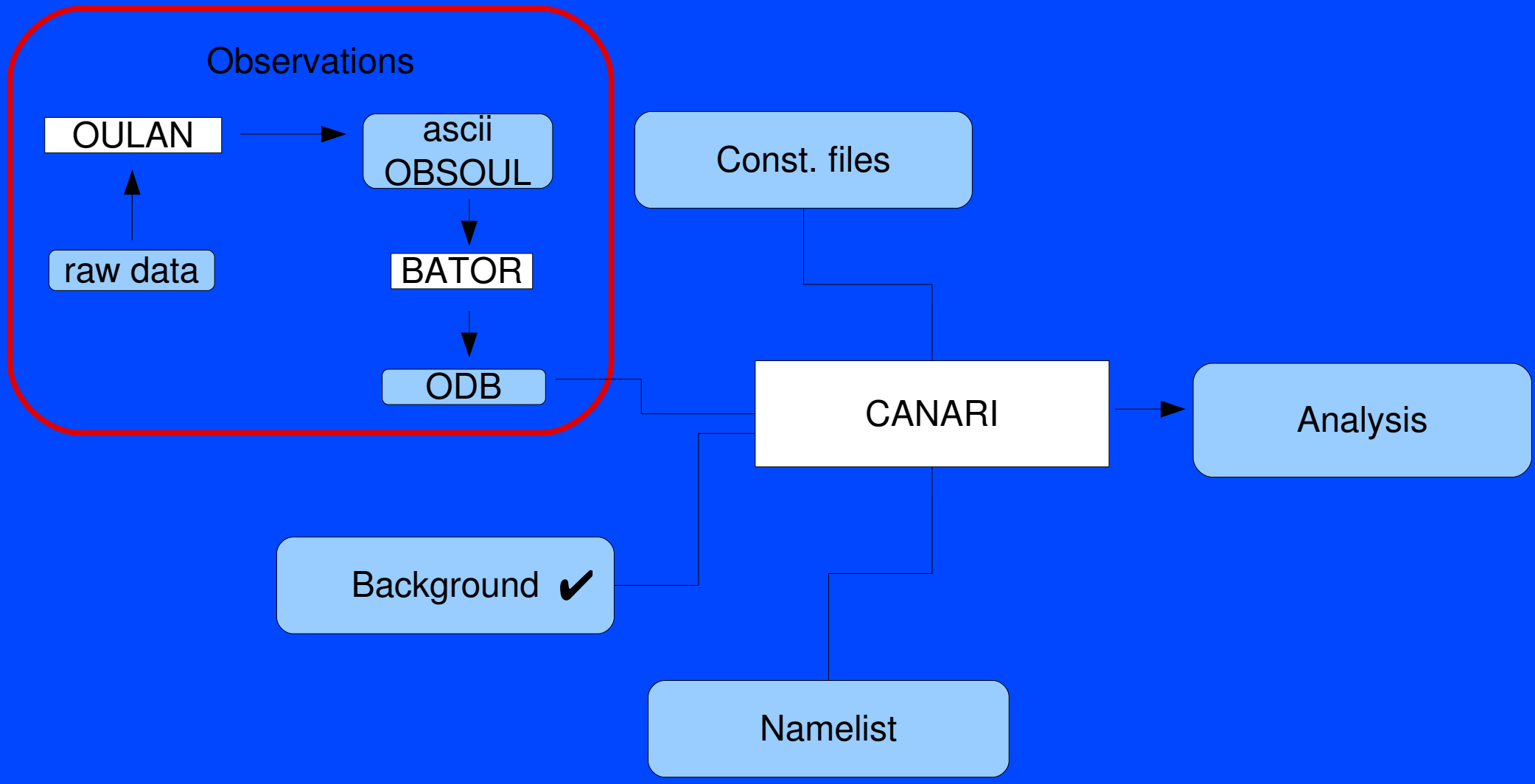
Ingredients



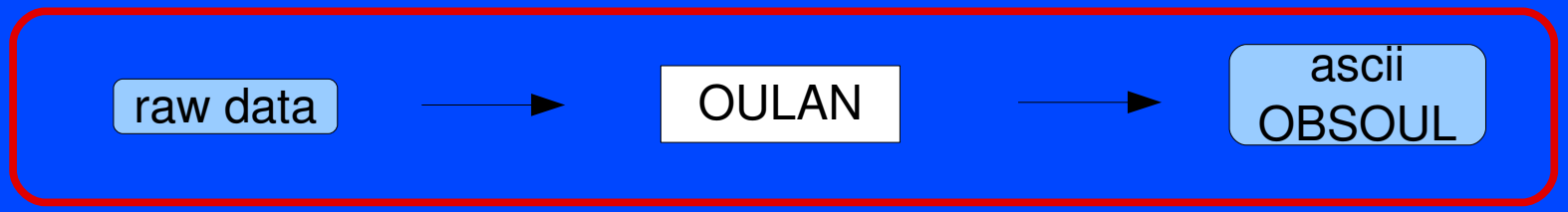
Ingredients



Ingredients



Prepare observations #1



raw data

Input observations stored in your local database (TEMP, SYNOP) usually in many different formats...

OULAN

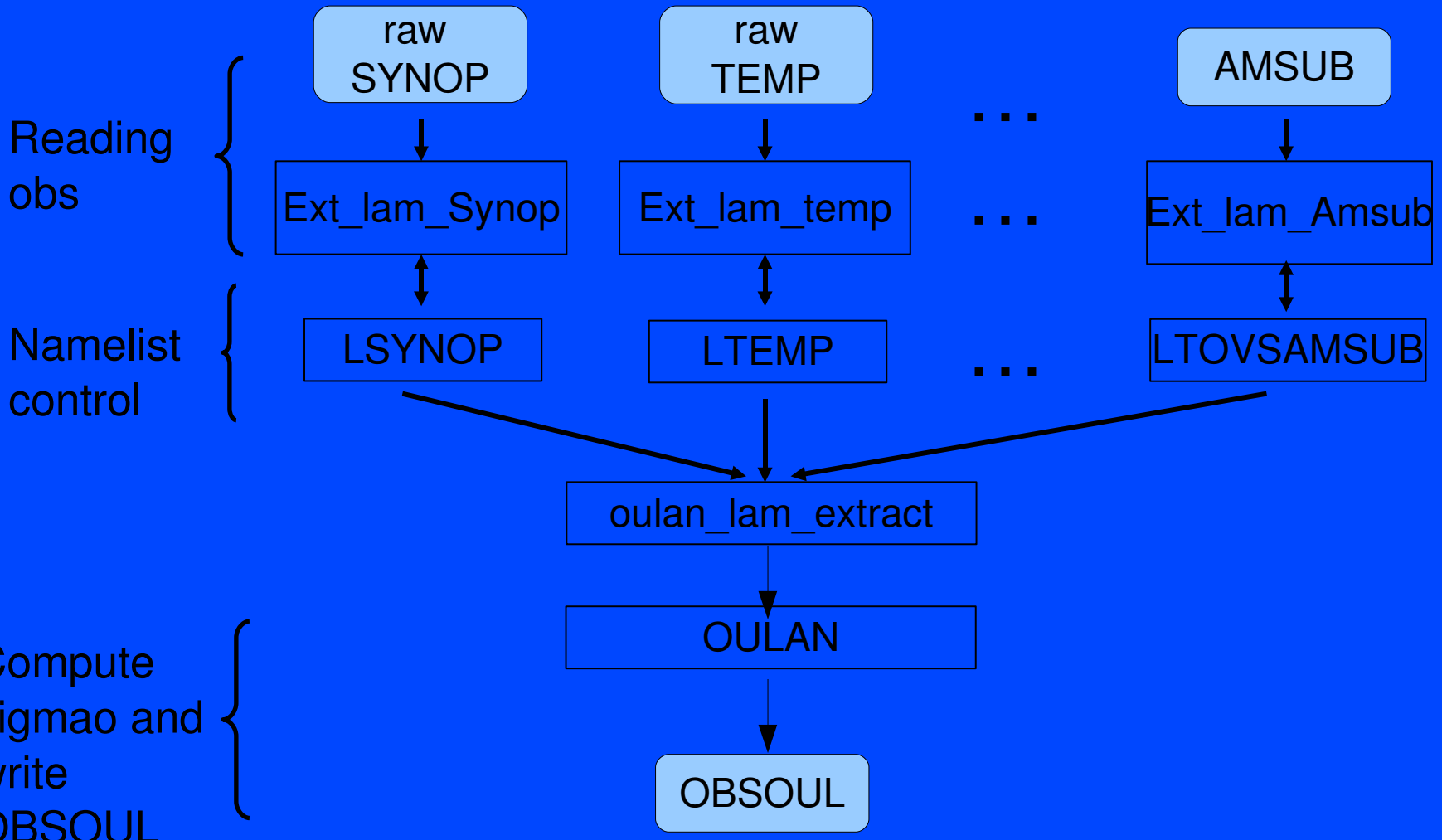
Package of Fortran programs to prepare the needed ascii input for the ODB preparation. It is not only file conversion but a part of the obs errors are also computed here. This program is out of the scope of the ARPEGE/ALADIN/IFS code maintenance!

ascii
OBSOUL

ascii file with a special format. Input of the BATOR program for the ODB preparation.

Prepare observations #1

The structure of OULAN



Prepare observations #1

Example for SYNOP:

```
20041215 12
42 1 10014011    48.10000    19.51667  '12756   '  20041215 120000
153.0000000 6 1111 100000 1 -103290.0000 0.1699999976E+39
0.0000000000E+00 2064 39 101310.0000 0.1699999976E+39
271.2600098 2048 58 101310.0000 0.1699999976E+39 82.00000000
2048 7 101310.0000 0.3211538133E-03 0.2632536227E-02 2048 41
101310.0000 3.000000000 190.0000000 2048 91 101310.0000
0.1699999976E+39 100.0000000 2048
```

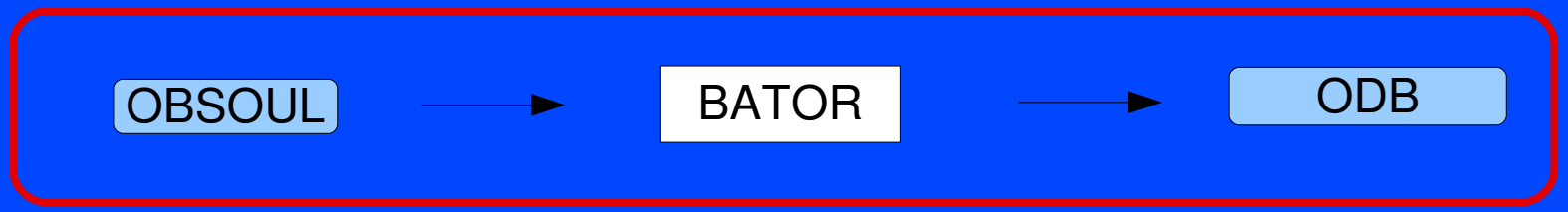
For the **OBSOUL file format** and **OULAN namelist** description see the documentation on the LACE web: http://www.rclace.eu/?page=11

Kertesz.S, 2007: Overview of the observation usage in the ALADIN variational data assimilation system

You can find this document also on 3700a: ~wshop01/Doc/lace_obspp.pdf

```
&NADIRS
  NDATE=20020225,
  NRESO=00,
  ALANZA          =    90.,
  ALASZA          =   -90.,
  ALOOZA          =  -180.,
  ALOEZA          =   180.,
  NDIFFM1         =    30,
  NDIFFP1         =    30,
  NDIFFM2         =   300,
  NDIFFP2         =   259,
  LTOVSAMSUA      = .TRUE.,
  LTOVSAMSUB      = .TRUE.,
  LTOVSHIRS       = .FALSE.,
  LTEMP           = .TRUE.,
  LSYNOP          = .TRUE.,
  LAMDAR          = .TRUE.,
  LEUROPROFIL     = .TRUE.,
  LGEOWIND        = .TRUE.,
  LSATOB          = .FALSE.,
  NINIT           =    0,
  LRH2Q           = .TRUE.,
/END
&NANBOB
  NBTOVSAMSUA     =  80000,
  NBTOVSAMSUB     =  80000,
  NBTOVSHIRS      =   8000,
  NBTEMP          =   1000,
  NBSYNOP         =   4000,
  NBAMDAR         =   9000,
  NBEUROPROFIL    =   8000,
  NBSATOB         =  20000,
  NBGEOWIND       =  20000,
```

Prepare observations #2



OBSOUL

Output of OULAN... ✔

BATOR

Program package to create ECMA ODB data. Part of the ARPEGE/ALADIN/IFS code, maintained regularly cycle-by-cycle. BATOR is used to blacklist data too. Part of the sigmaos are specified in BATOR. Options to read OBSOUL, grib and BUFR data.

ODB

ECMA type of ODB .This is the input for CANARI OI. (A little bit different from the inputs for LAMFLAG/SCREENING/3DVAR! --> no sub-bases needed if ODB_MERGEODB_DIRECT=1)

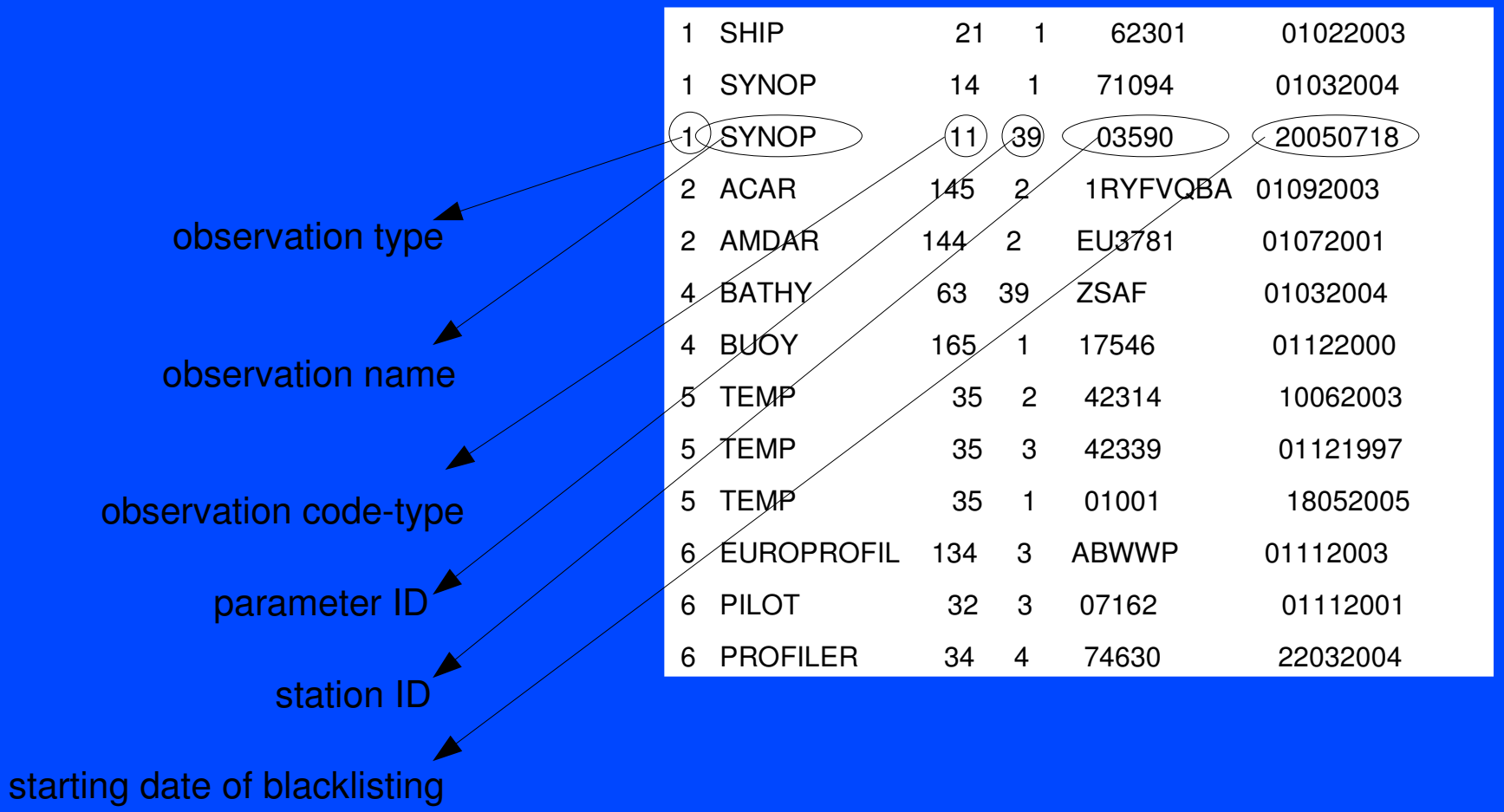
Prepare observations #2

Purpose of BATOR

Blacklisting	Use blacklist files: LISTE_NOIRE_DIAP (static blacklisting of full reports by ID) LISTE_LOC (blacklisting by height and location)
Set sigmao values	Set array “ECTERO” in bator_init.F90
Write an ODB	Set ODB variables

Prepare observations #2

Blacklisting
(example: LISTE_NOIRE_DIAP)



Prepare observations #2

Set sigmao values
(bator_init.F90)

```
ECTERO(NSYNOP,1,1:5,1)=(/Z_VAL,1.4_JPRB,2.0_JPRB,0.1_JPRB,1.5_JPRB /)   ! synop
ECTERO(NSYNOP,2,1:5,1)=(/Z_VAL,1.4_JPRB,3.0_JPRB,0.1_JPRB,1.5_JPRB /)   ! ship
ECTERO(NTEMP,1,3, 1:19) =(/ 2.3_JPRB, 2.3_JPRB, 2.3_JPRB, 2.4_JPRB,   &! temp vent
 & 2.5_JPRB, 2.5_JPRB, 2.8_JPRB, 3.0_JPRB, 3.3_JPRB, 3.6_JPRB, &
 & 3.7_JPRB, 3.8_JPRB, 3.8_JPRB, 3.8_JPRB, 3.8_JPRB, 3.9_JPRB, &
 & 4.1_JPRB, 4.3_JPRB, 4.5_JPRB /)
ECTERO(NTEMP,1,2, 1:19 )=(/ 1.7_JPRB, 1.6_JPRB, 1.5_JPRB, 1.4_JPRB,   &! temp tempe
 & 1.4_JPRB, 1.3_JPRB, 1.3_JPRB, 1.3_JPRB, 1.3_JPRB, 1.4_JPRB, &
 & 1.5_JPRB, 1.5_JPRB, 1.6_JPRB, 1.6_JPRB, 1.6_JPRB, 1.7_JPRB, &
 & 1.8_JPRB, 1.9_JPRB, 2.0_JPRB /)
ECTERO(NTEMP,1,2,1:19) = ECTERO(NTEMP,1,2,1:19) / 1.2_JPRB
ECTERO(NTEMP,1,1, 1:19)=(/  8.0_JPRB,  8.2_JPRB,  8.6_JPRB,          &! temp geop
 & 9.0_JPRB,  9.4_JPRB,  9.9_JPRB, 11.4_JPRB, 12.7_JPRB, &
 & 14.0_JPRB, 16.0_JPRB, 15.7_JPRB, 17.2_JPRB, 20.1_JPRB, &
 & 22.0_JPRB, 24.4_JPRB, 27.0_JPRB, 30.0_JPRB, 31.5_JPRB, &
```

Prepare observations #2

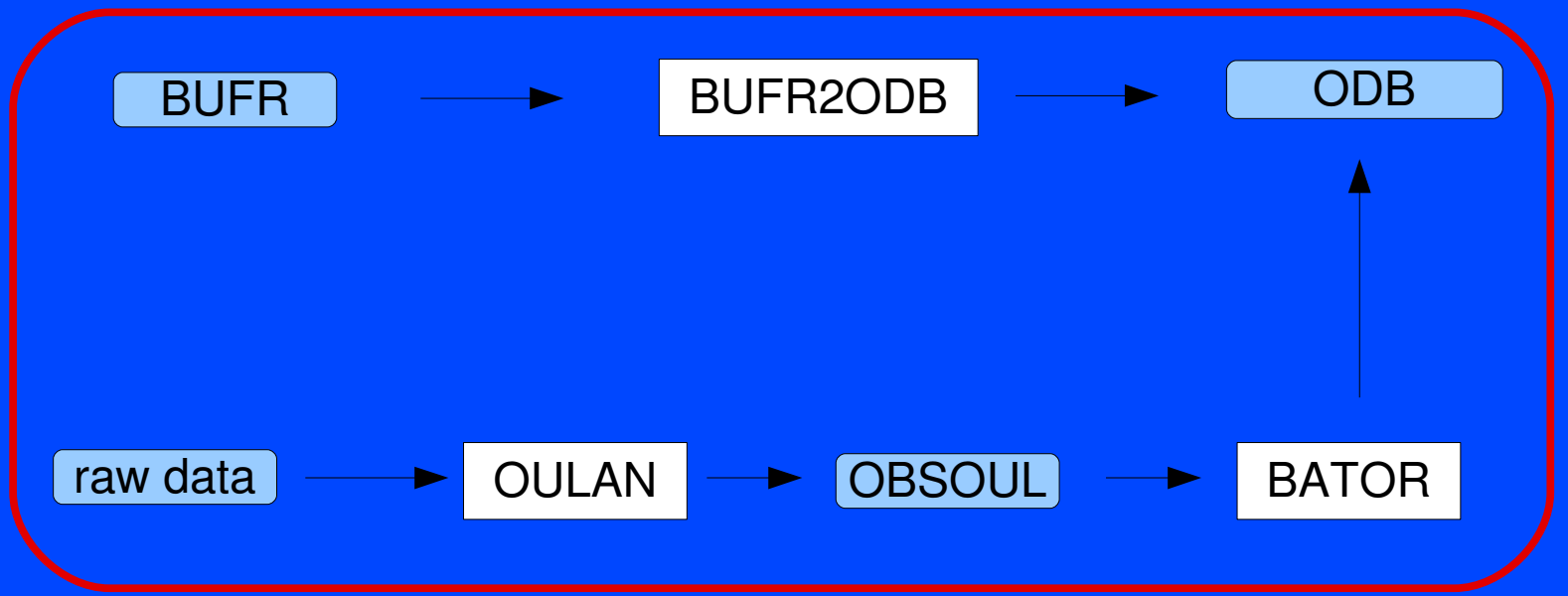
ODB settings for BATOR
(in your script)

```
#--- ODB settings for batodb
ODB_CMA=ECMA
ODB_SRCPATH_ECMA=${d_DB}/ECMA
ODB_DATAPATH_ECMA=${d_DB}/ECMA
ODB_ANALYSIS_DATE=${n_date}
ODB_ANALYSIS_TIME=${n_time}0000
IOASSIGN=${d_DB}/ECMA/IOASSIGN
BATOR_NBPOOL=${NBPROC}
```

ODB_CMA:	ODB type (ECMA--> extended or CCMA --> compressed)
ODB_SRCPATH:	the path for your ODB base
ODB_DATAPATH_ECMA:	the path for your ODB base
ODB_ANALYSIS_DATE:	date of your analysis in form yyyymmdd
ODB_ANALYSIS_TIME:	time of your analysis in form hh0000
IOASSIGN:	the path for your IOASSIGN file (This describes the structure of your ODB)

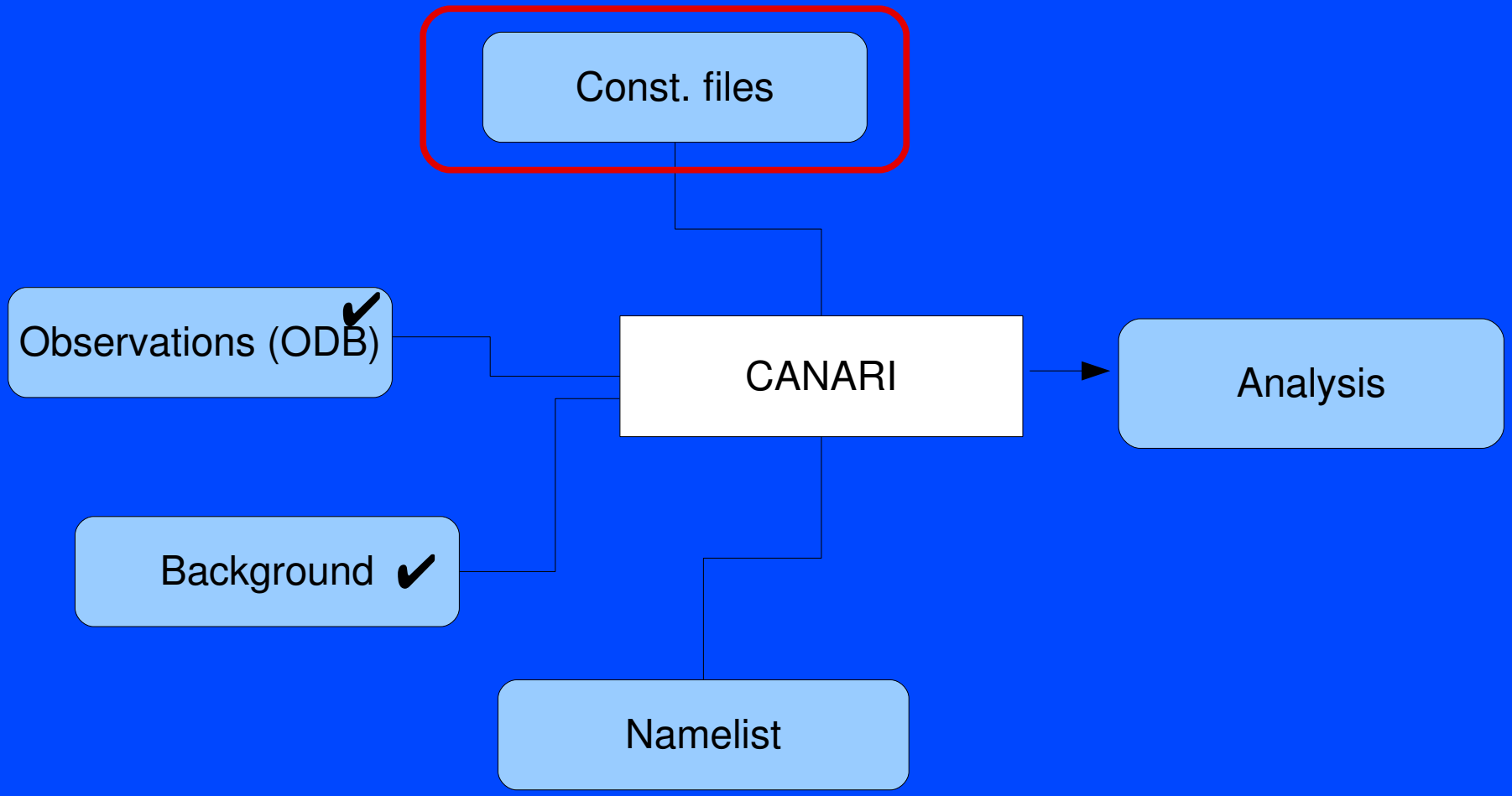
Prepare observations #3

Option of BUFR2ODB instead of OULAN and BATOR

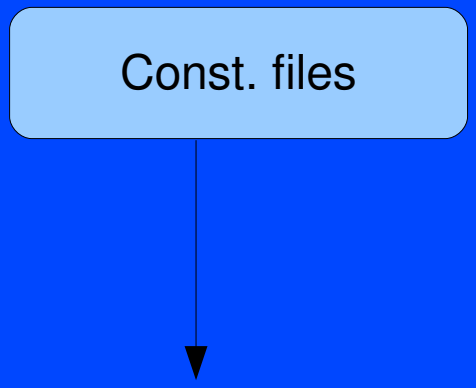


There is not many experience in ALADIN about BUFR2ODB. Probably more in the HARMONIE installations (Norway, Sweden?)

Ingredients



Constant files



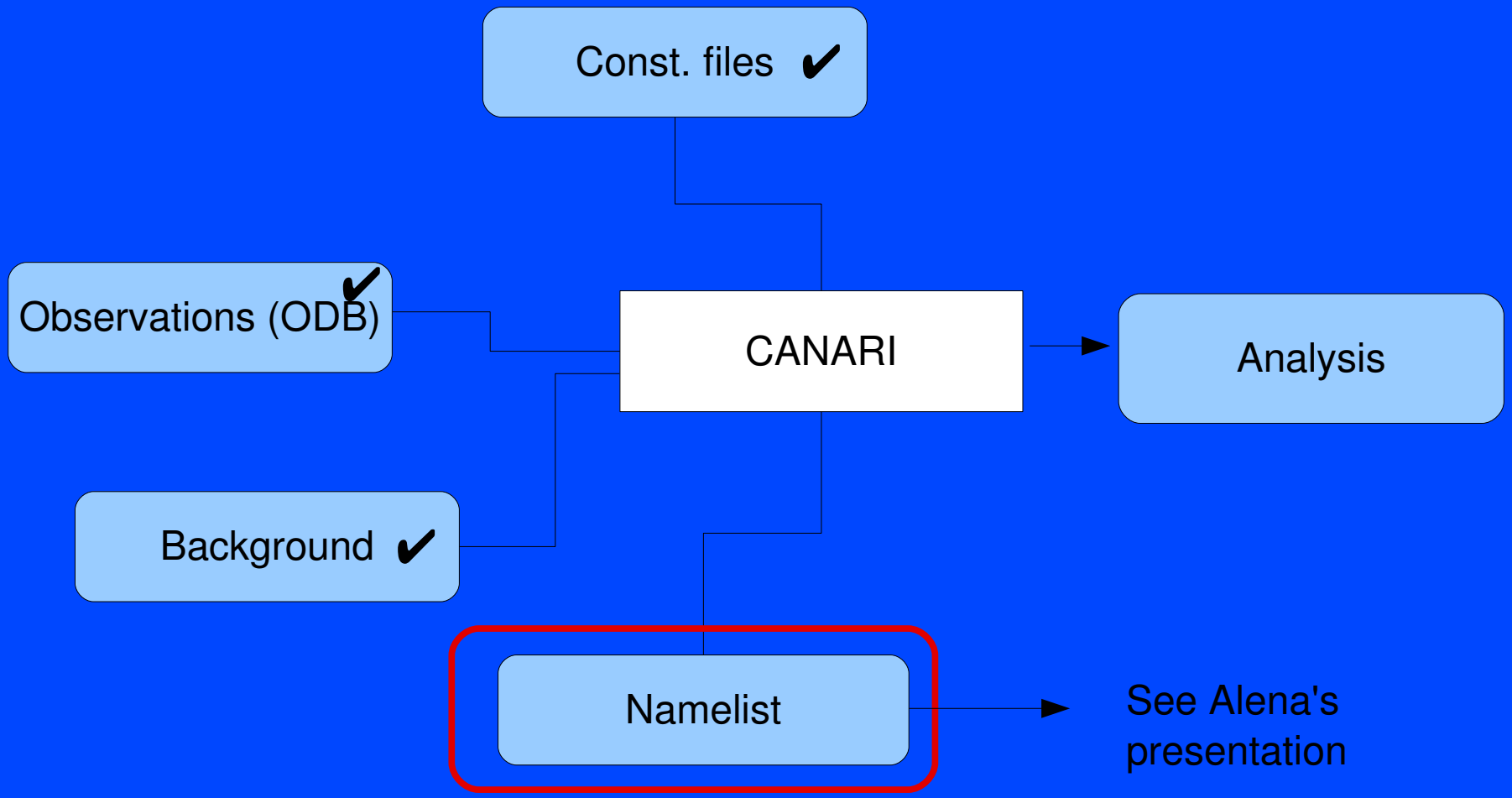
(and some others used in upper-air analysis only)

Climate files of the actual month *mm* and of the month *mm+1* or *mm-1* depending of the date (ICMSHANALCLIM, ICMSHANALCLI2). These are the same files as used in e927 as output climatology. The purpose is relaxation of some analyzed fields towards the climatology (SST for instance).

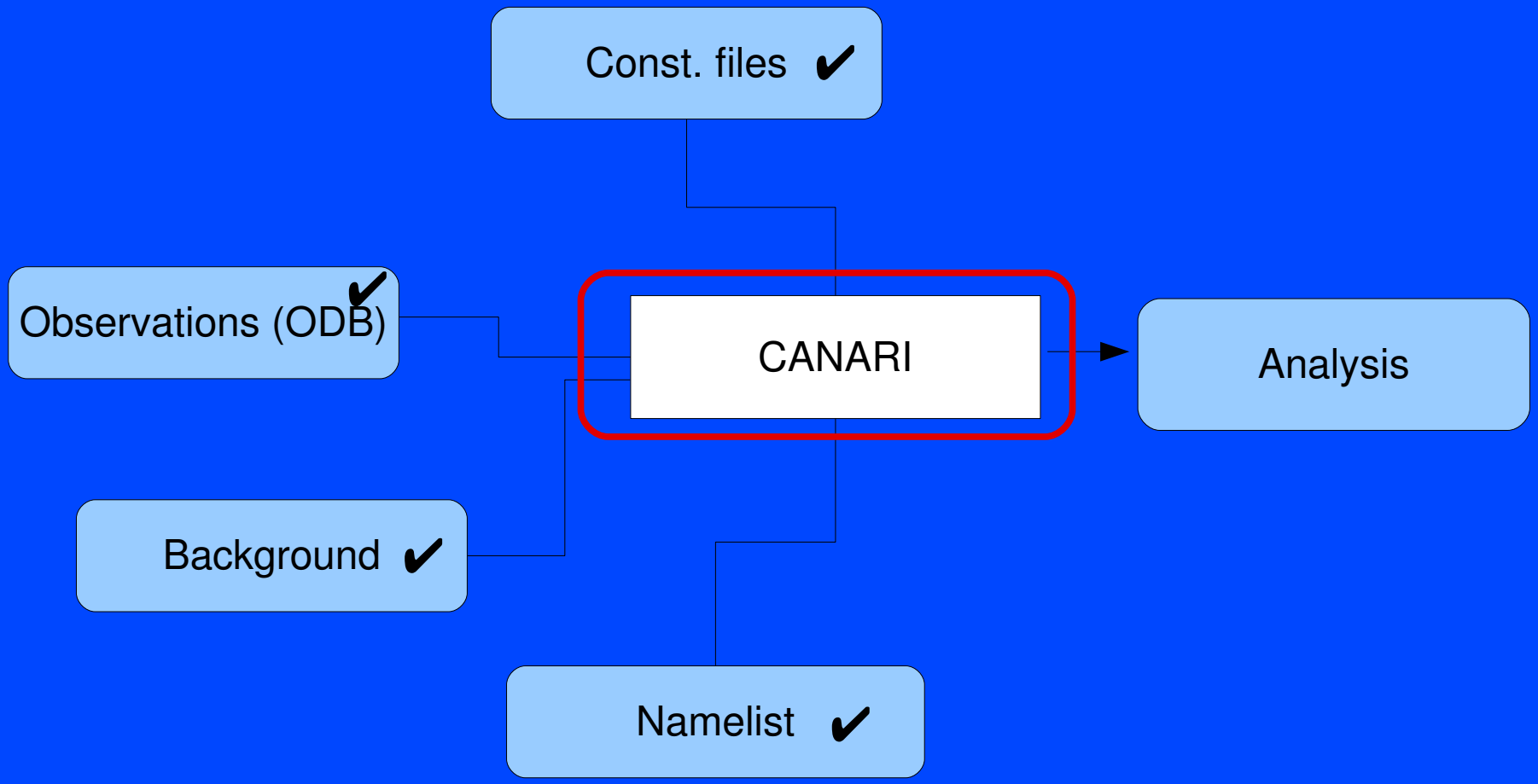
ISBA polynomes to derive soil moisture analysis from the 2m increments (fort.61).

Increment file (ICMSHANALISSE): it contains increments of earlier analyzes. The actual analysis is smoothed with respect to these earlier increments. During the CANARI run it is updated (ICMSHANALLISSEF). Its use and the update is activated by LISSEW=.T. in the namelist NACVEG.

Ingredients



Ingredients



Run CANARI OI

CANARI

Part of the ARPEGE/ALADIN/IFS code, namely the conf. 701. The same executable as for all the other ALADIN configurations. It is named ALDODB in gmkpack.

The command line:

ALDODB -c701 -vmeteo -maladin -eANAL -t1. -ft0 -aeul

-c: configuration (CANARI = 701)
-v: version of the code (always “meteo” for ARPEGE/ALADIN)
-m: LAM or global model (“aladin” or “arpege”)
-e: experiment name (ANAL for instance)
-t: time-step length (“1.” for CANARI)
-f: duration of the integration (“t0” or “h0” for CANARI)
-a: dynamical scheme (does not matter for CANARI Eulerian = “eul”
semi-Lagrangian = “sli”)

Run CANARI OI

How CANARI knows about your ODB?

```
#-- ODB settings
ODB_CMA=ECMA
ODB_SRCPATH_ECMA=${d_DB}/ECMA
ODB_DATAPATH_ECMA=${d_DB}/ECMA
IOASSIGN=${d_DB}/ECMA/IOASSIGN
ODB_MERGEODB_DIRECT=1
```

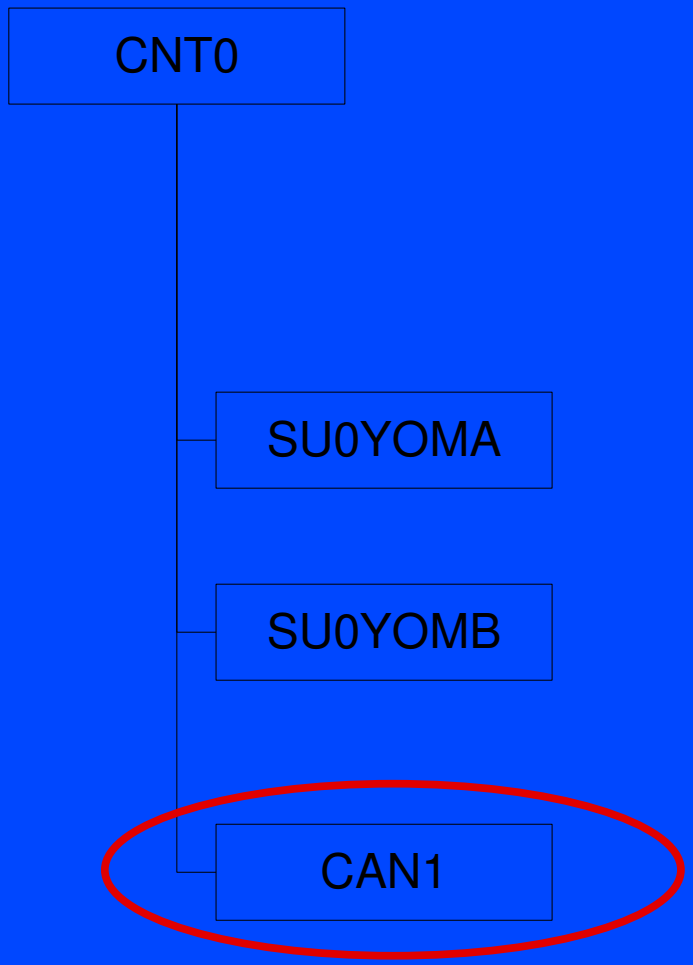
ODB_CMA:	ODB type (ECMA--> extended or CCMA --> compressed)
ODB_SRCPATH:	the path for your ODB base
ODB_DATAPATH_ECMA:	the path for your ODB base
IOASSIGN:	the path for your IOASSIGN file (This describes the structure of your ODB)
ODB_MERGEODB_DIRECT:	merge or not your ODB while running CANARI. (If your ODB was not merged previously, always use “1”.)

Run CANARI OI

How CANARI knows about your other inputs?

```
#-- get the guess
cp ${d_GUESS}/guess ICMSHANALINIT
ln -sf ICMSHANALINIT ELSCFANALALBC000

#-- get the lisseff file
ln -sf ${d_GUESS}/ICMSHANALLISSEF ICMSHANALLISSE

#-- get the climate files
ln -sf ${d_CLIM}/HUNG8kmlin_${mm}     ICMSHANALCLIM
ln -sf ${d_CLIM}/HUNG8kmlin_${mm2}    ICMSHANALCLI2

#-- get the ISBA polynomes
cp ${d_NAM}/POLYNOMES_ISBA      fort.61

#-- get the namelist
cp ${d_NAM}/701.nml        fort.4
```

Run CANARI OI



For the organization of the code please see the documentation on the ALADIN documentation site: http://www.cnrm.meteo.fr/gmapdoc/

Taillefer F. 2002: Optimal Interpolation CANARI (2002)

You can find this document also on 3700a: ~wshop01/Doc/canari_doc_cy25t1ps

→ Control level of CANARI

Installation

OULAN

BATOR

ALDODB
(exe for CANARI)

Use your own tools!

Use gmkpack:

Prepare Makefiles or even simpler compiler tools. You can ask for help from Hungarian and French teams.

gmkpack -r cy30t1 -b main -a -n 01 -l ifort9_2B2 -o x -s -p aldodb/bator/ioassign

Script example

Please login to 3700a and edit ~/Canari/scr/Canari.sh

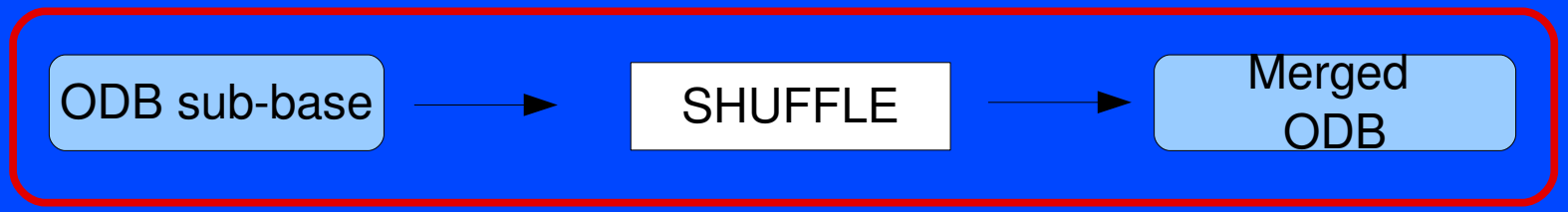
How to login?

user: wshop02, ... ,wshop15
password: 123456

example:
“ssh -X wshop02@3700a”

If you do not have ssh on your PC, please telnet to pc2264 (user: guest, pw: guest) and use ssh from there!

Prepare observations #4



ODB sub-base

Output of BATOR... ✔

SHUFFLE

Program package for ODB modifications. Part of the ARPEGE/ALADIN/IFS code (named as ODBTOOLS), maintained regularly cycle-by-cycle.
(Used also in VAR assimilation to get rid of data rejected by LAMFLAG and QC).

Merged ODB

Merged database of ECMA sub-bases. The ODB tables of the sub-bases are completed. This is the input for CANARI OI (and for LAMFLAG/Screening/3DVAR) (see next slide).

Prepare observations #4

Merged ODB