General code architecture and dataflow in ARPEGE/ALADIN/ALARO/AROME

Yann Seity (CNRM/GMAP)

METEO FRANCE
Toujours un temps d'avance

SUMMARY

- Computations organization

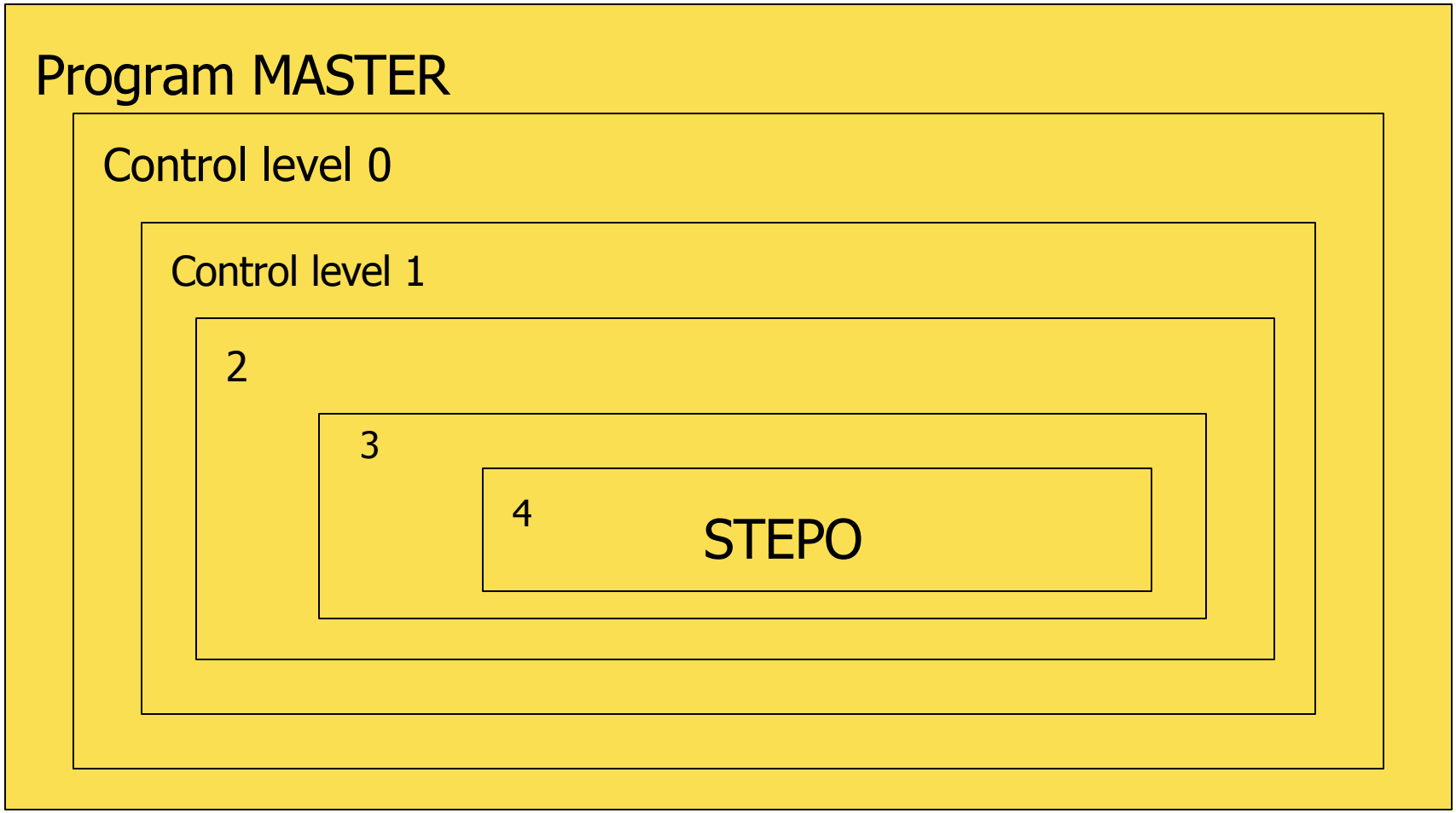
- Data flows (1D, 2D, 3D spectral and gridpoint variables)

Before stating…

- Naming conventions in ALADIN-ARPEGE :

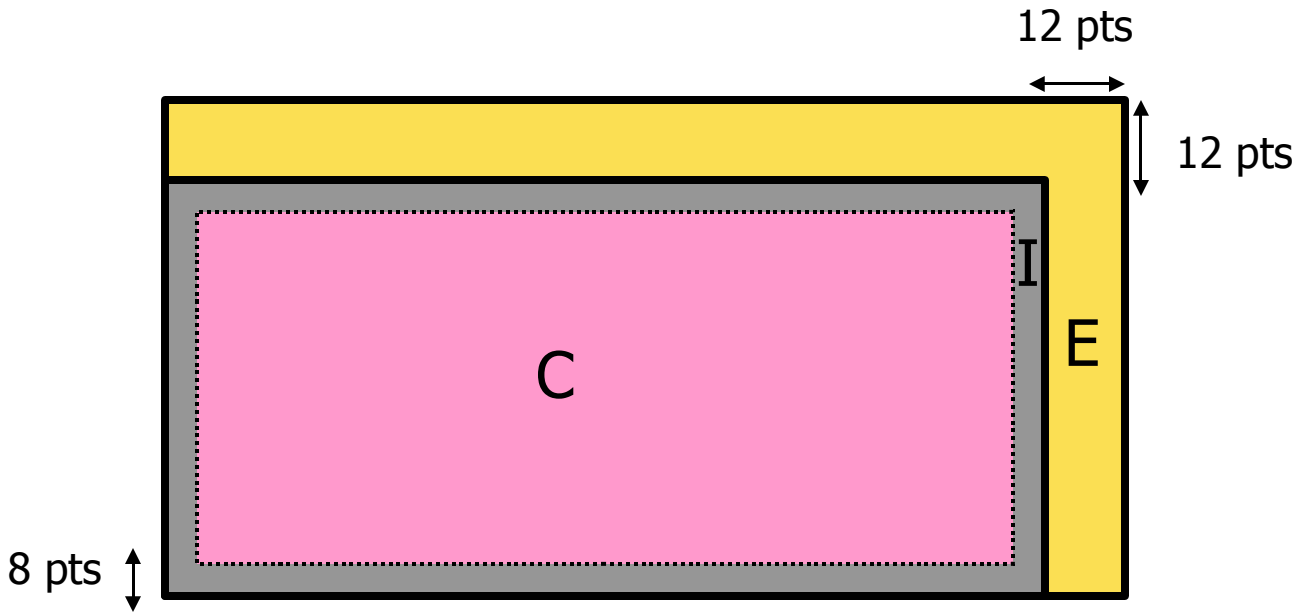
	ARPEGE	ALADIN
Setup routine	SUXXX	SUEXXX
Classical routine	XXXXX	EXXXXX
Module	YOMXXX	YEMXXX
Nameliste	NAMXXX	NAMXXX

General organization



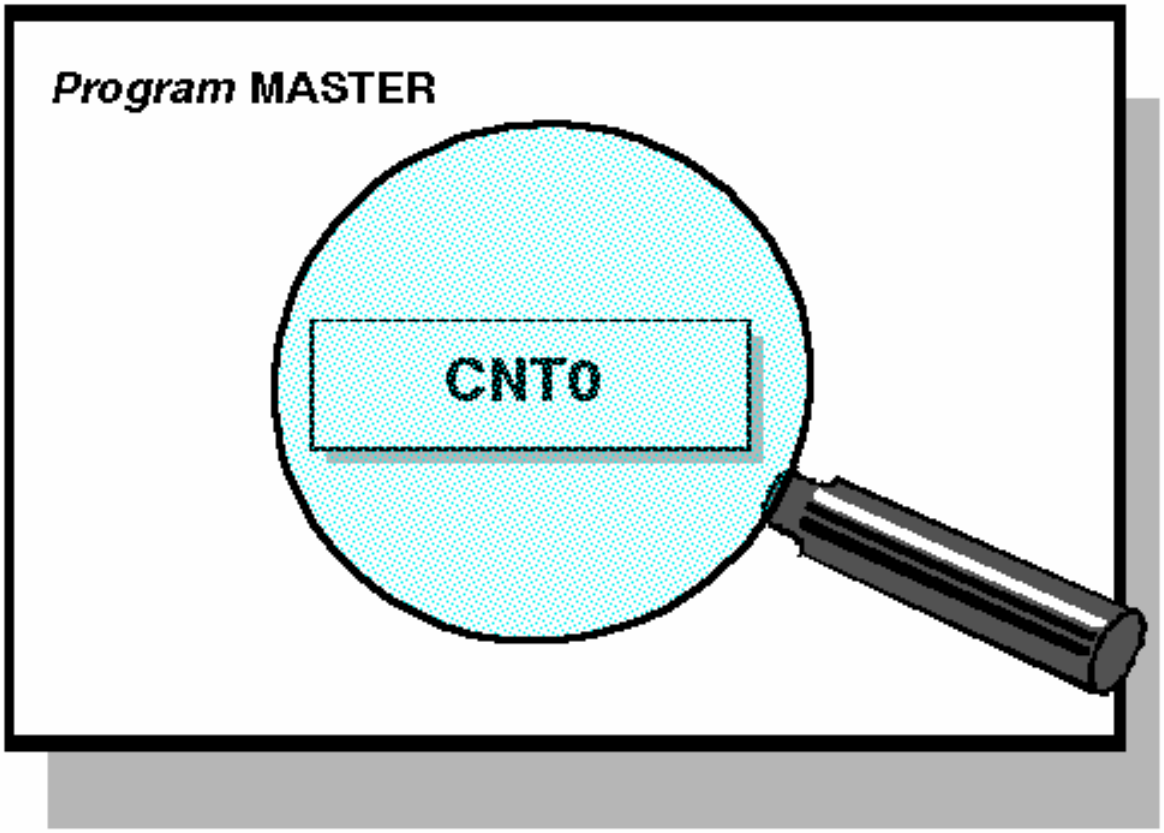
AROME-ALADIN Code in ARPEGE/IFS

- Embedded inside ARP/IFS
- Specific control keys [LELAM and LRPLANE (plane geometry)]
- « E » Rule :



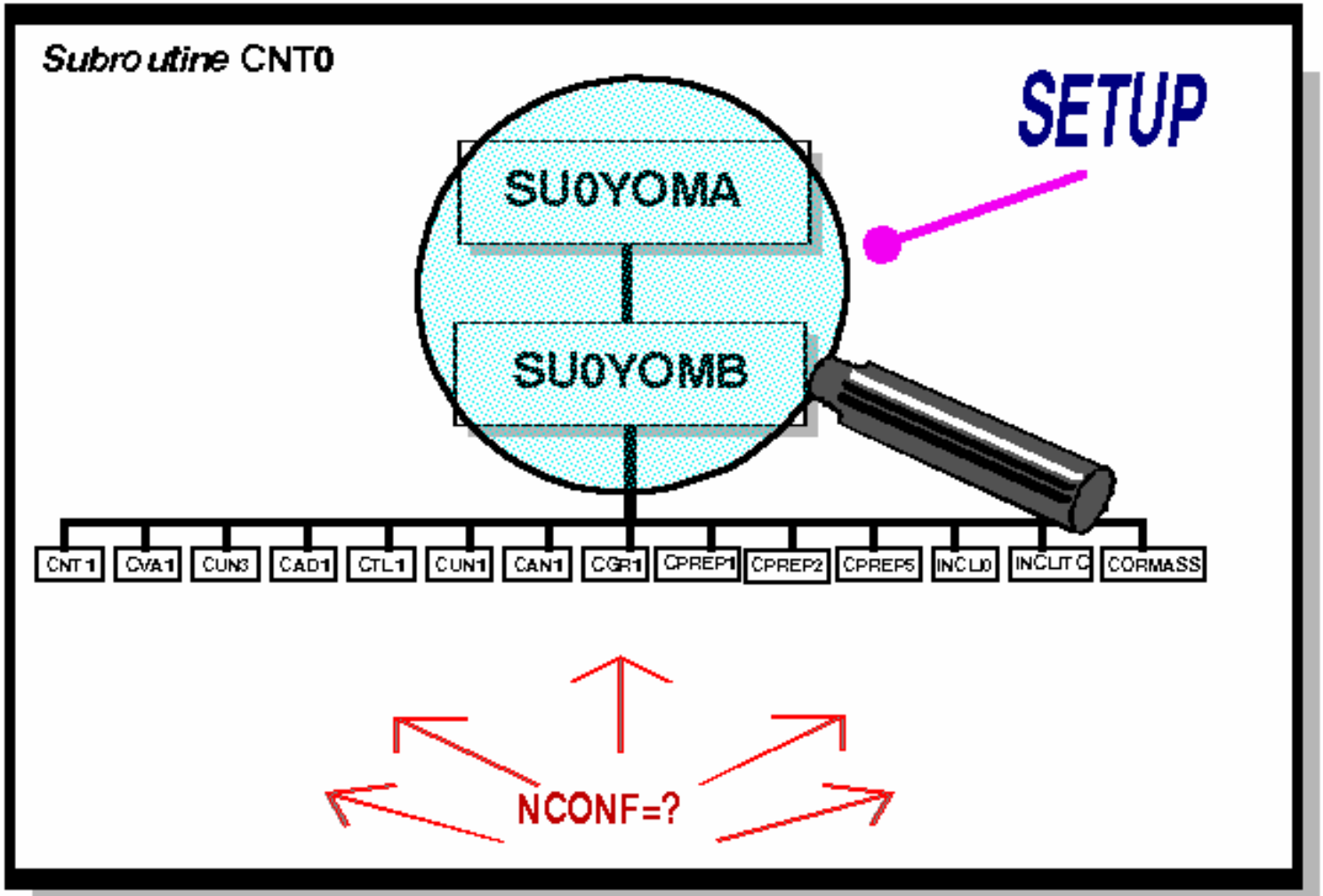
Computations organisation

METEO FRANCE



CONTROL LEVEL 0

METEO FRANCE

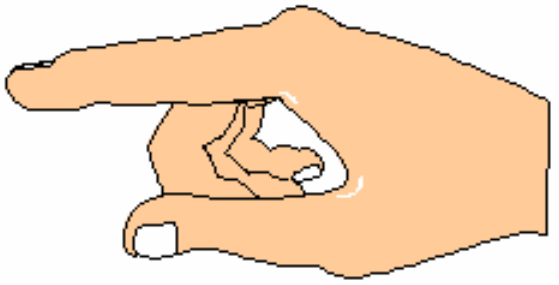


THE SETUP

METEO FRANCE

- To read the parameters controlled by the user
- To initialize all constants (ex : π)
- To allocate and initialize all working arrays

INTERNAL CONSISTANCY OF THE SETUP IS OF PRIME NECESSITY !



SETUP ORGANIZATION

METEO FRANCE

COMMAND LINE

+

DATA FILE HEADER

Subroutine SUOYOMA

SUARG

SU...
SU...
SU...
SU...

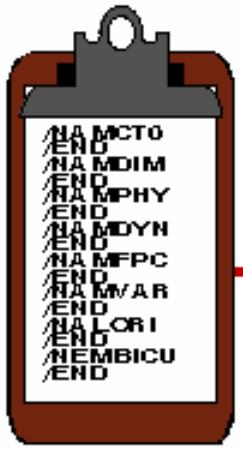
SUALLO

Reads the command line to simplify the incoming setup

Read namelists + command line to initialise scalar variables or arrays dimensionned by *parameters*

Allocate arrays

CENTRAL MEMORY ARRAYS



NAMELISTS FILE

Subroutine SUOYOMB

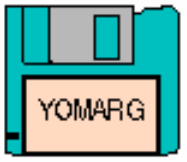
SU...
SU...
SU...

Read more namelists + command line to initialize allocated arrays for the setup

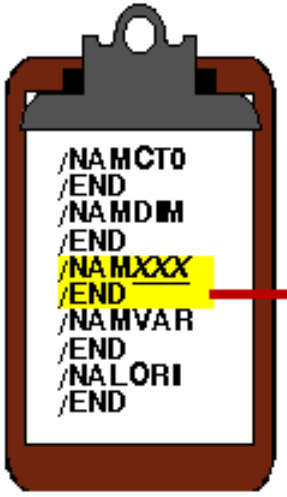
EX : ./AROME.EX -eFCST -c001 -maladin -vmeteo -asli -t60. -fh30

FRAMEWORK OF A SETUP SUBROUTINE

METEO FRANCE



COMMAND LINE ARGUMENTS



NAMELISTS FILE

Subroutine SU*XXX*

INITIALIZE YOM*XXX* WITH DEFAULT VALUES

READ NAMELIST NAM*XXX* IN ORDER TO OVERWRITE (PARTIALLY) YOM*XXX*

CONTROL YOM*XXX* VARIABLES

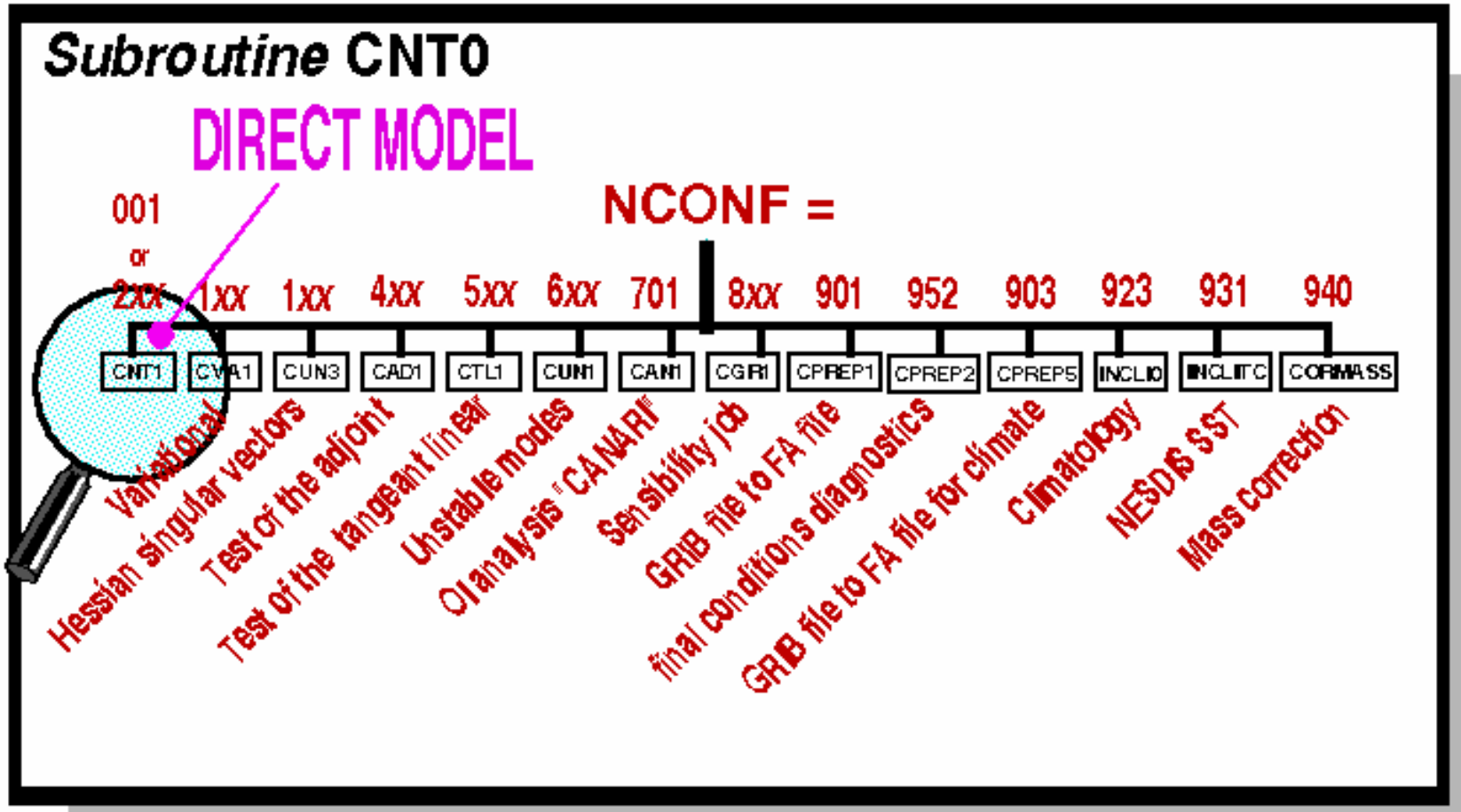
PRINT OUT YOM*XXX* VARIABLES

Taking into account command line arguments and namelists already read

MODULE YOM*XXX*

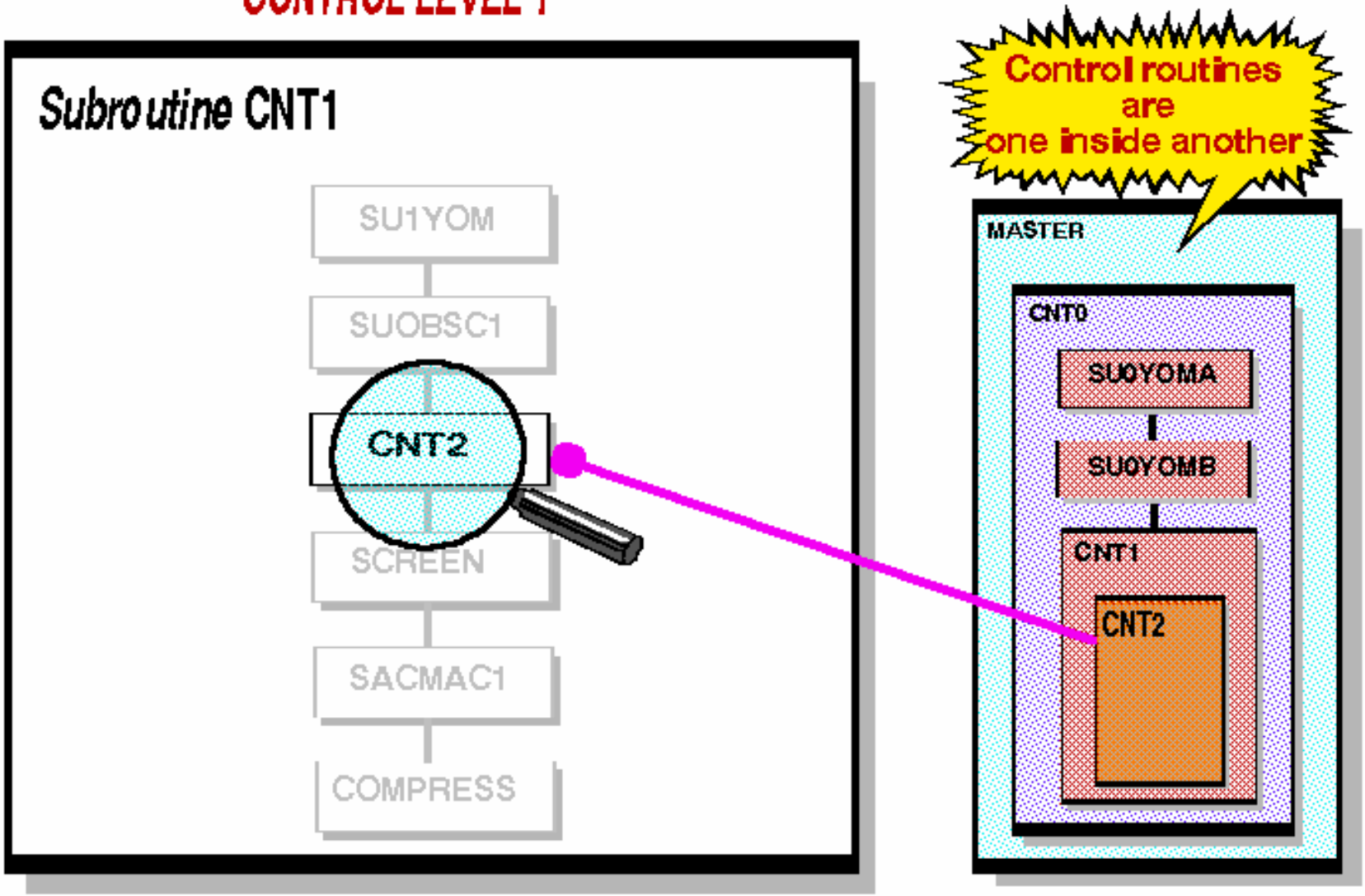
WHICH CONFIGURATION OF WORK ?

METEO FRANCE



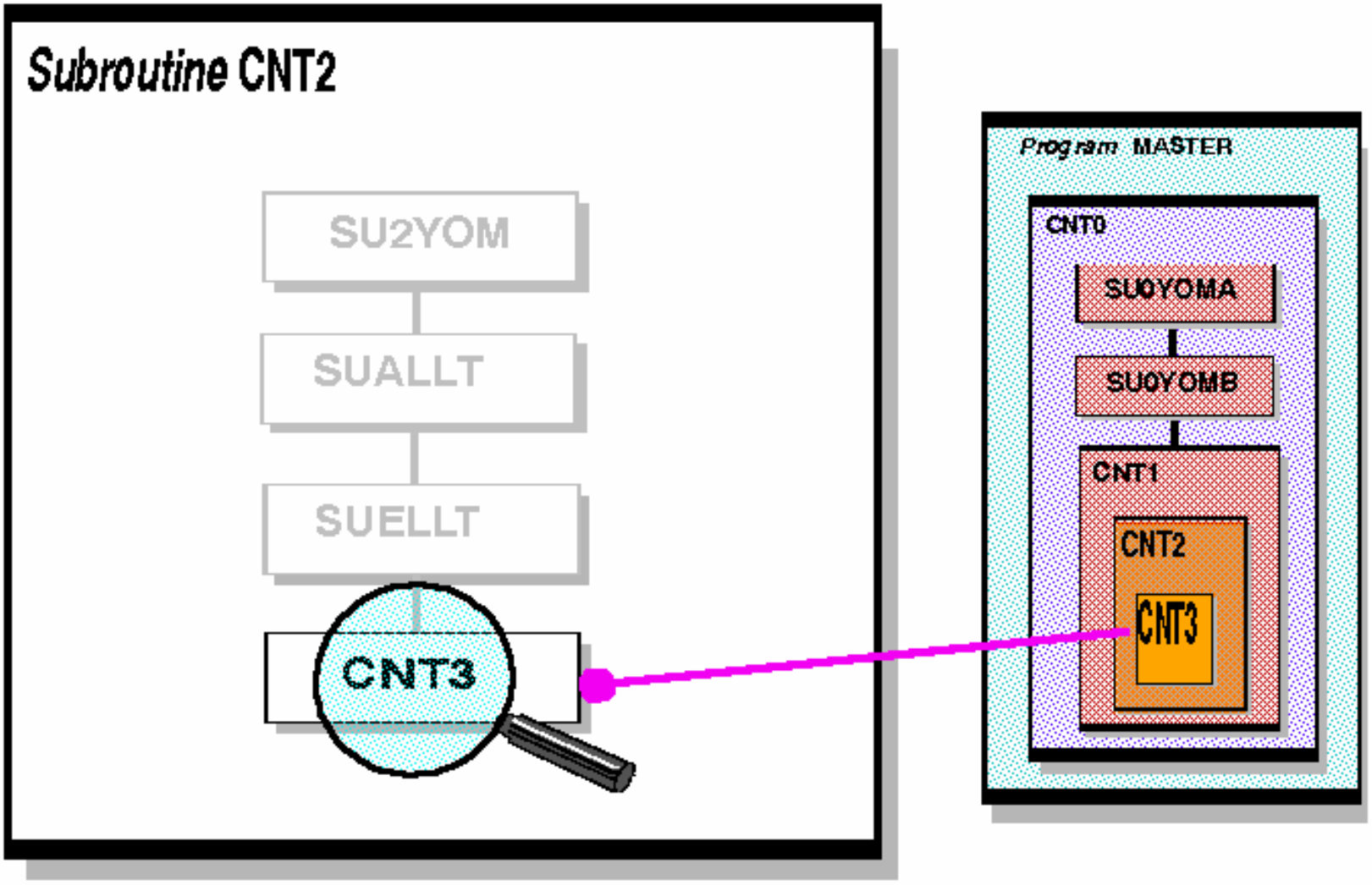
The direct model : CNT1

"CONTROL LEVEL 1"



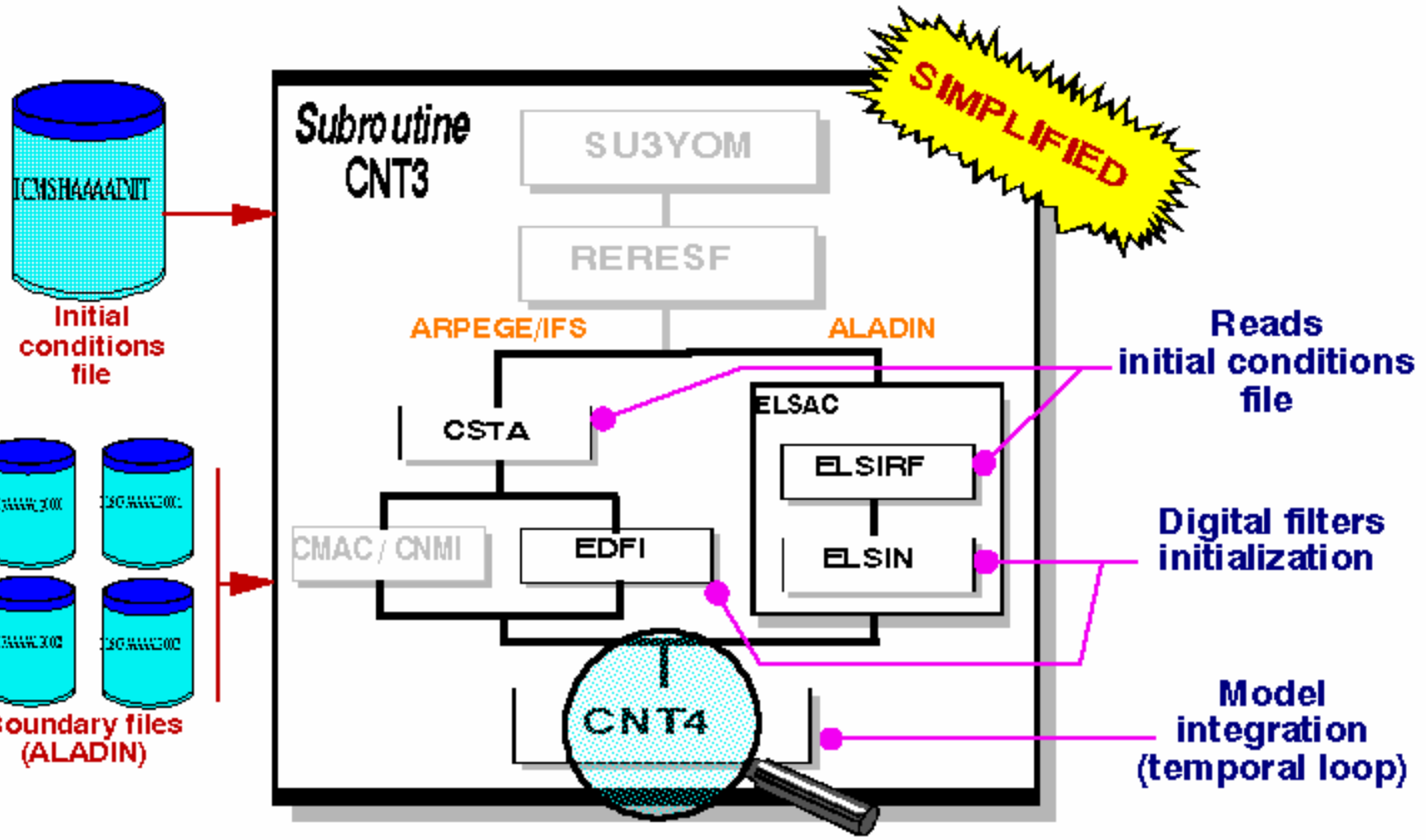
The direct model : CNT2

"CONTROL LEVEL 2"



The direct model : CNT3

"CONTROL LEVEL 3" = READ AND INITIALIZE INITIAL FIELDS



Direct model integration : CNT4

METEO FRANCE

"CONTROL LEVEL 4" = TEMPORAL LOOP

HIGHLY SIMPLIFIED !

Subroutine CNT4

MONIO

Monitoring of input/output events (diagnostics, post-processing, back-ups, etc)

Diagnostics

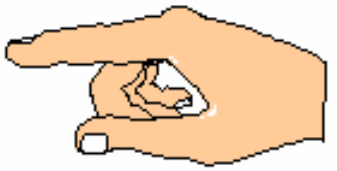
Prepare configuration of STEPO

From START to STOP

TEMPORAL LOOP

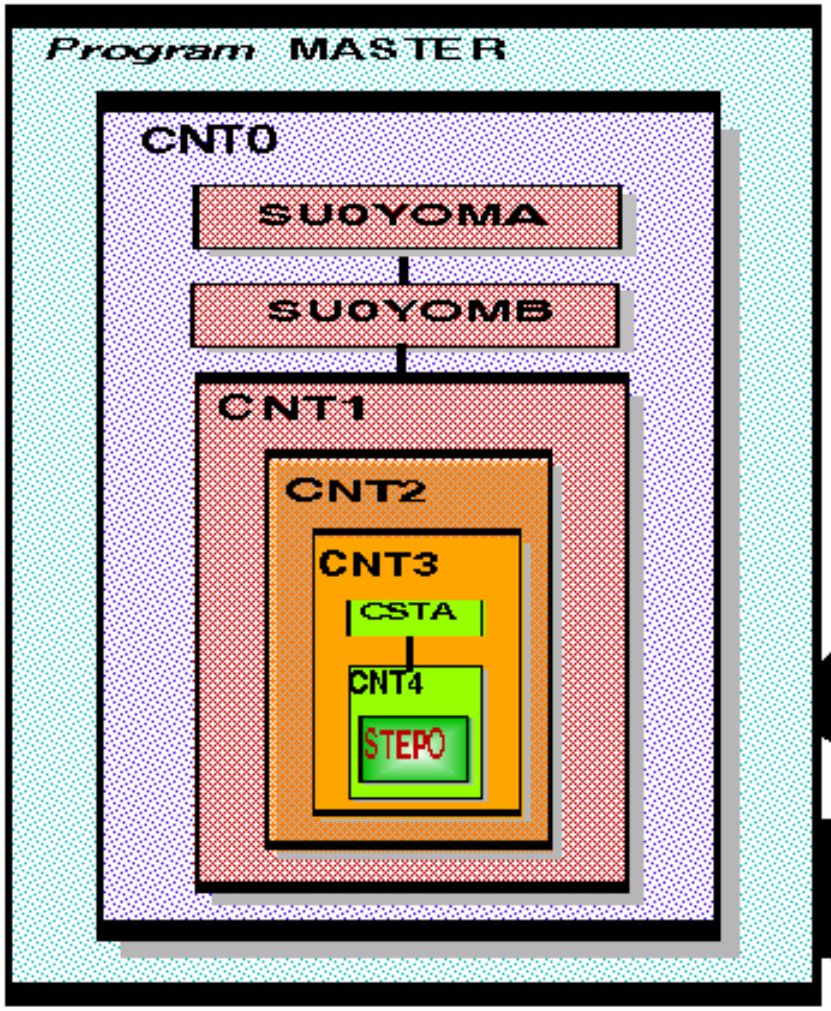
STEPO

STEPO *is the heart of the model*



STEPO

METEO FRANCE



STEPO *IS*
AN ELEMENTARY
STEP
OF THE MODEL

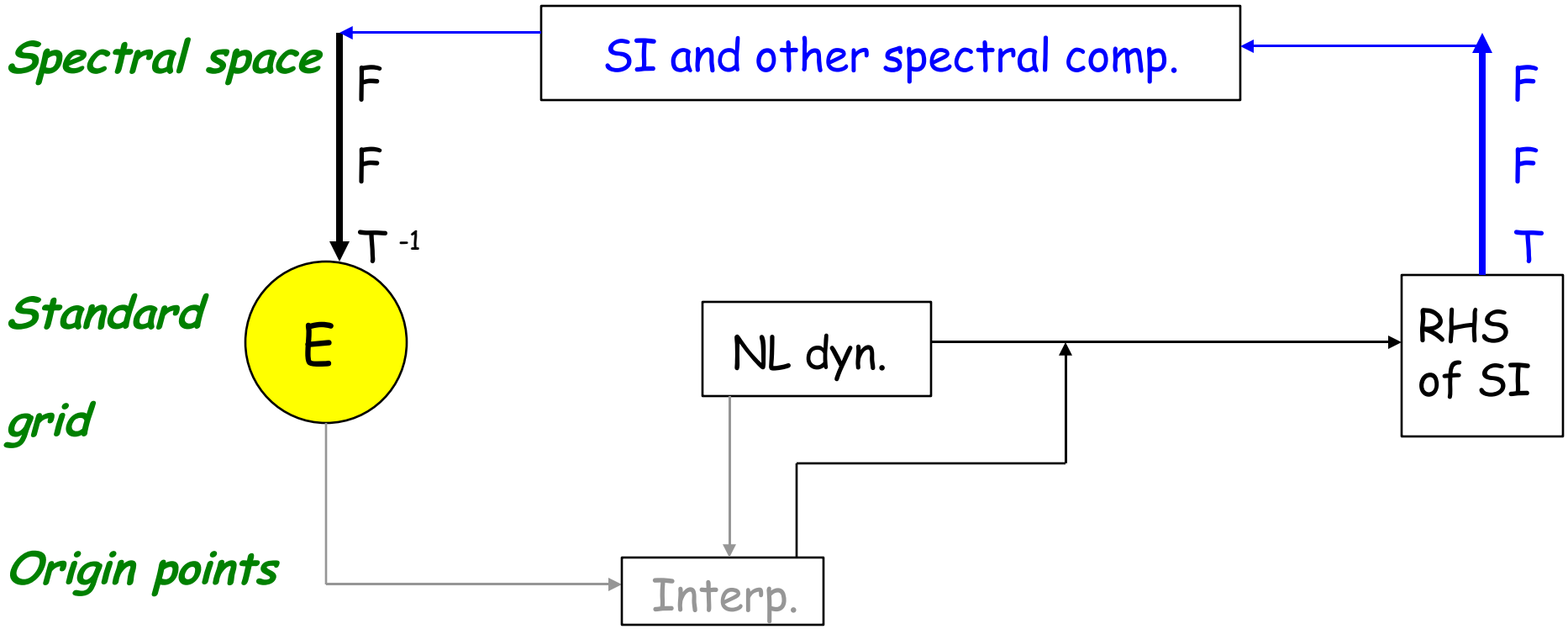
STEPO
IS INVOKED
WITH A
CONFIGURATION STRING
COMPOSED OF
9 CHARACTERS

(Example : Call Stepo ('0AAA00AAA'))

Time step organisation

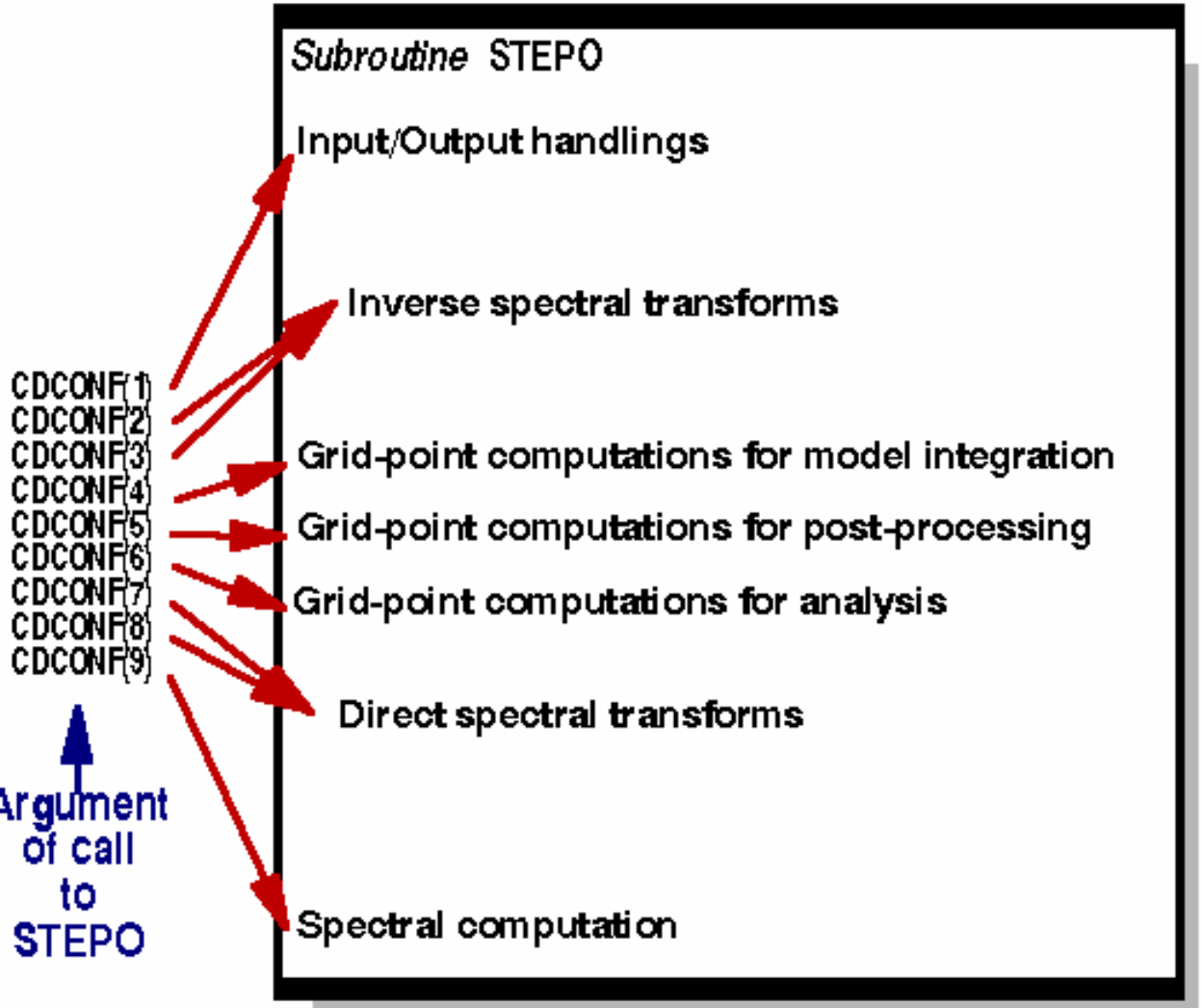
Classical ALADIN/ARPEGE time step :

E : Physics calculations



Organization of STEPO

METEO FRANCE



9 PARTS :

EACH PART IS CONTROLLED BY 1 CHARACTER OF THE CONFIGURATION STRING

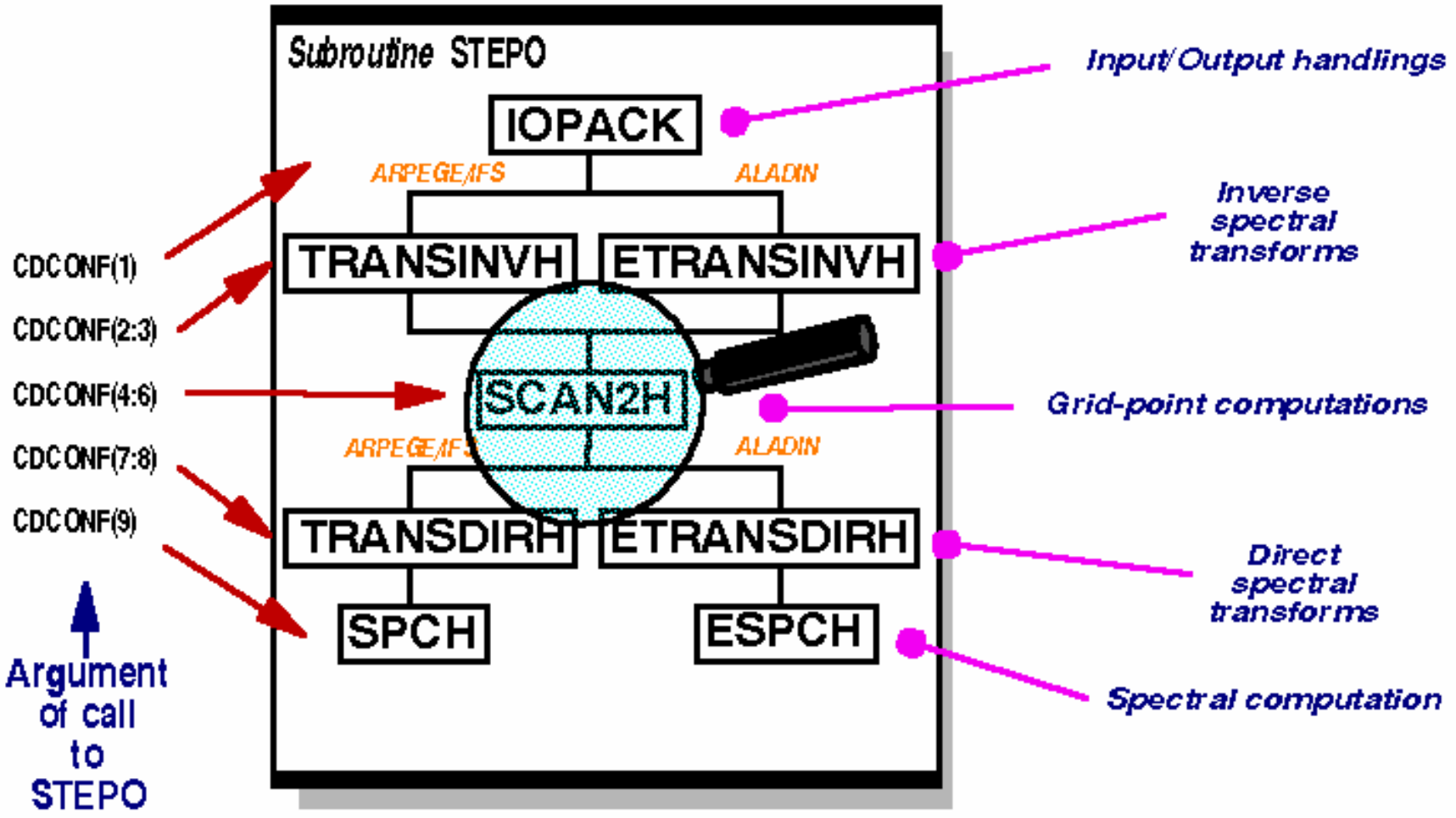
CDCONF(n)='0'
⇔
Do not enter the part n

Example :
a "normal" time step of a forecast is :

CALL STEPO('0AAA00AAA')

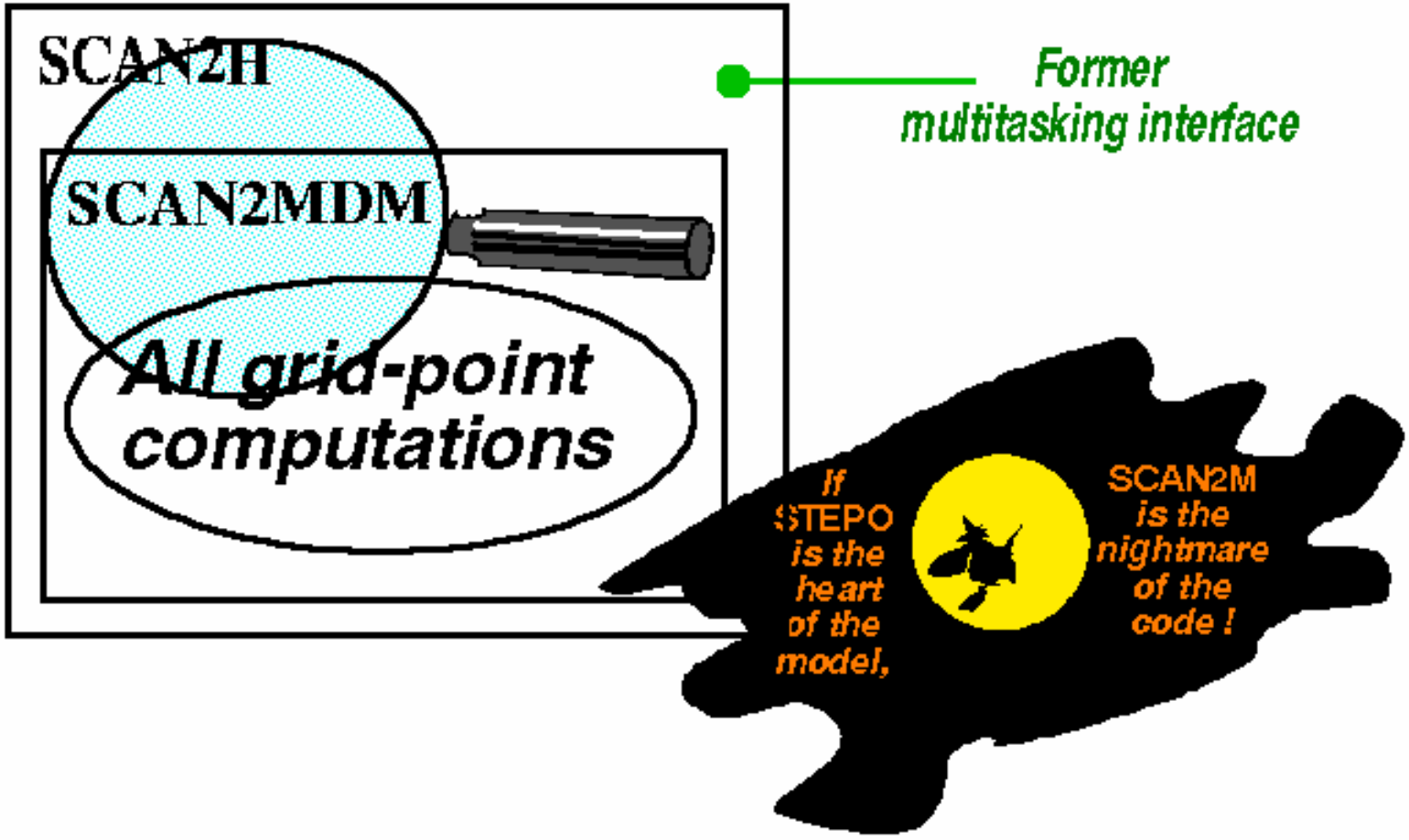
Structure of the code of STEPO

METEO FRANCE



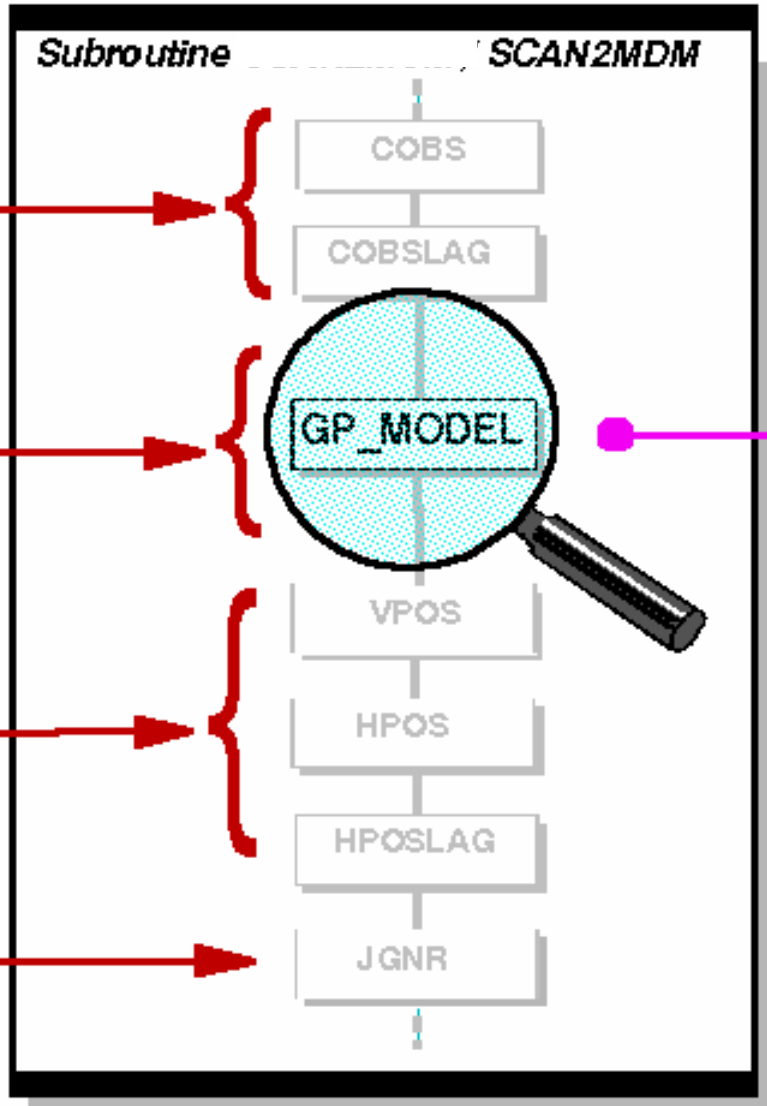
SCAN2H & SCAN2MDM

METEO FRANCE



Gridpoint computations

METEO FRANCE



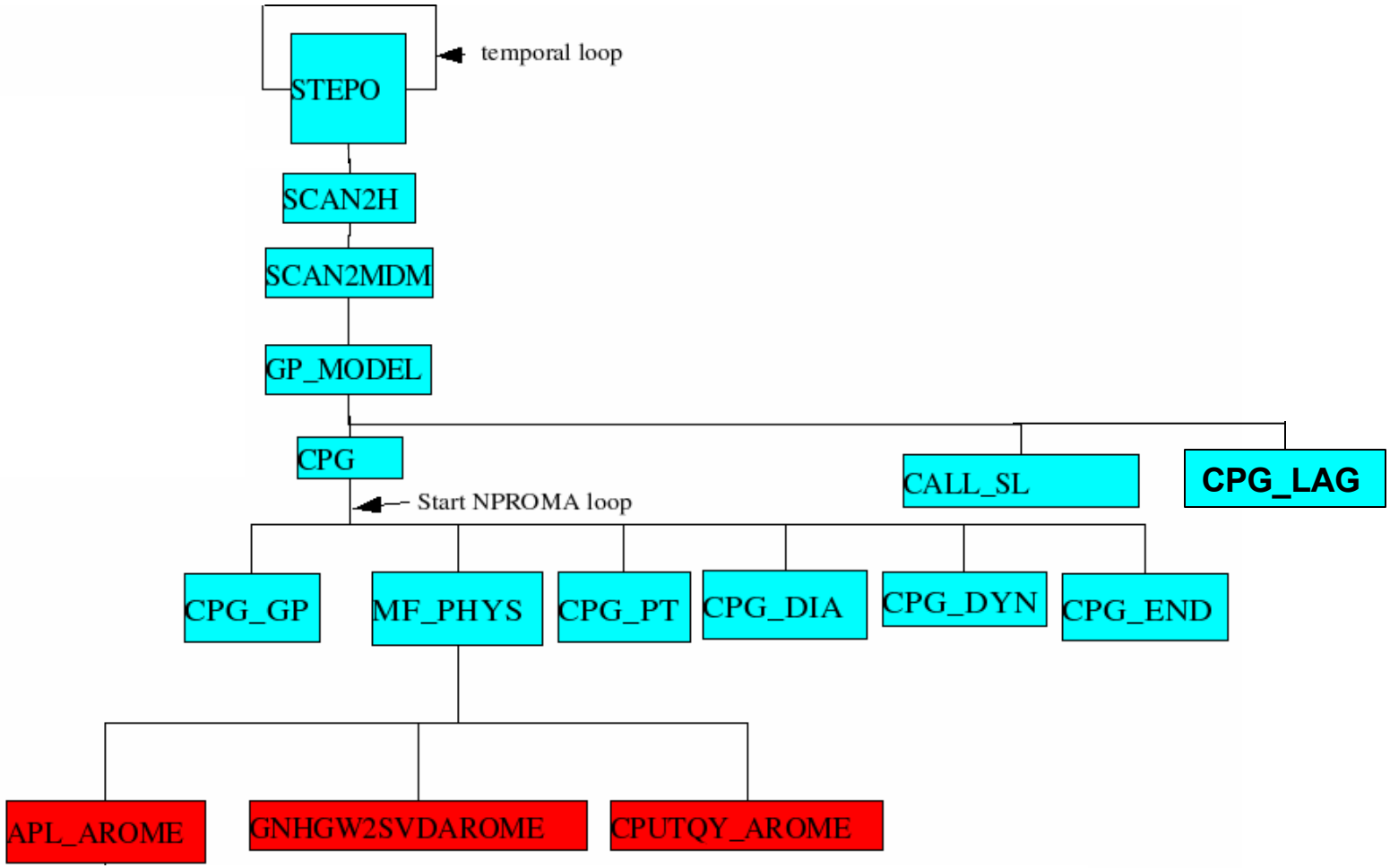
Comparison to observations

Direct model grid-point computations

Post-processing

Cost function JG computations

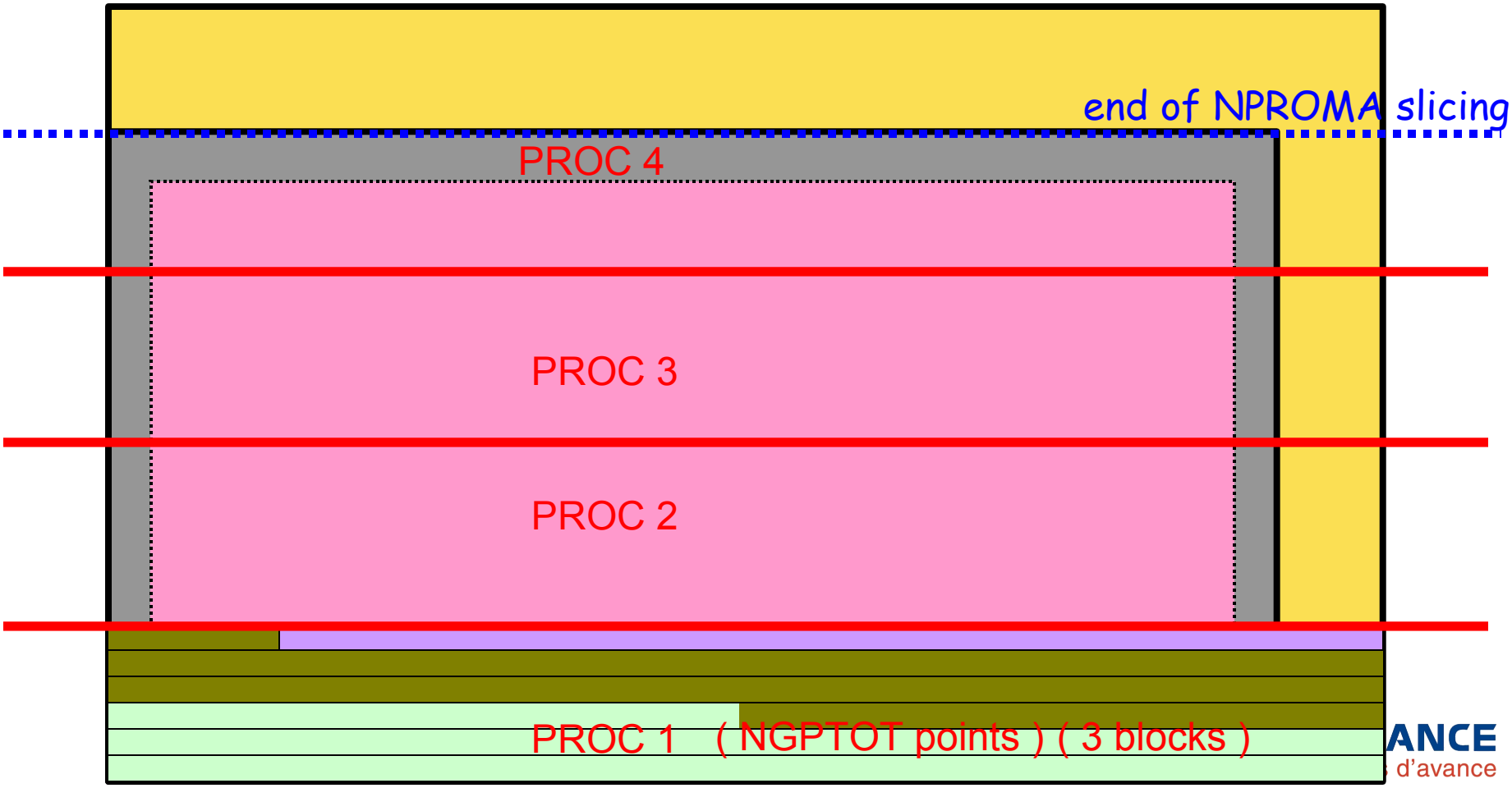
Gridpoint computations



AROME
Case :

NPROMA

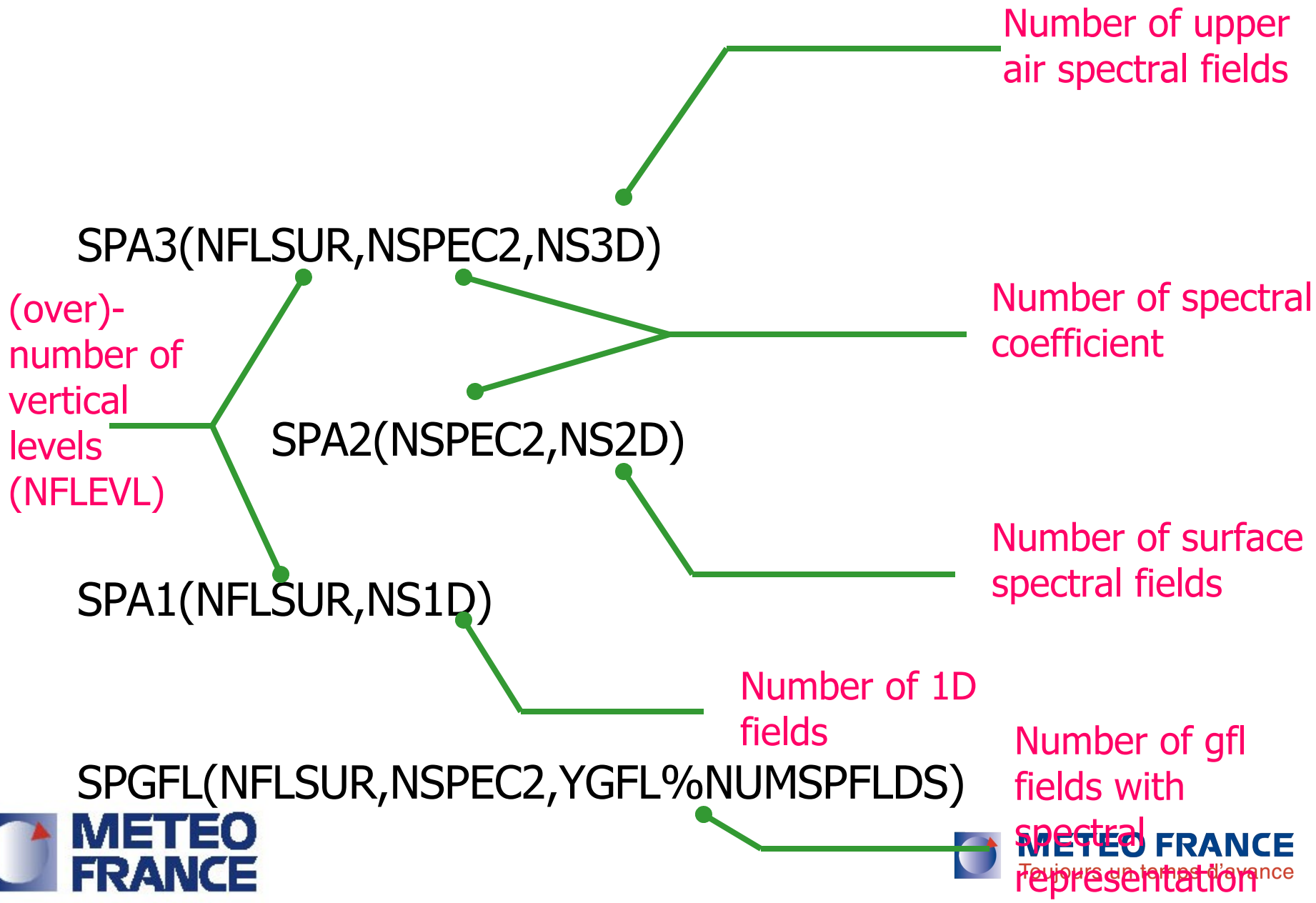
Example in AROME or ALADIN with 4 procs



The data flow

- Spectral arrays
- Grid points arrays
- Data flow

Spectral (distributed) arrays



Spectral arrays are split:

(target <= pointers)

SPA3(:,:,1) <= SPVOR(:,:) *Vorticity*

SPA3(:,:,2) <= SPDIV(:,:) *Divergence*

SPA3(:,:,3) <= SPT(:,:) *Temperature*

SPA2(:,1) <= SPSP(:) *(Ln) surf pressure*

SPA2(:,2) <= SPOR(:) *Surf geopotential*

SPA1(:,1) <= SPUB(:) *mean wind (U)*

SPA1(:,2) <= SPVB(:) *mean wind (V)*

NH variables

Pressure departure

SPA3(:,:,i) <= SPSPD(:,:)

Vertical divergency

SPA3(:,:,j) <= SPSVD(:,:)

… And possibly

SPA3(:,:,k) <= SPGFL(:,:,YQ%MPSP) *Specific moisture*

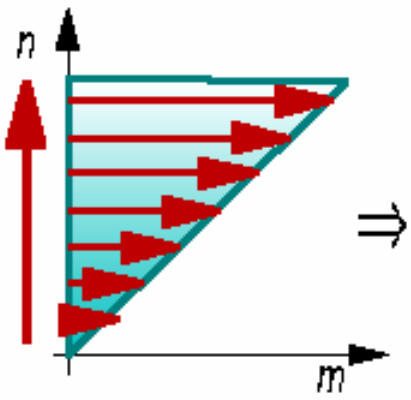
SPA3(:,:,l) <= SPGFL(:,:,YL%MPSP) *liquid water*

SPA3(:,:,m) <= SPGFL(:,:,YI%MPSP) *Ice*

Ordering of spectral coefficients

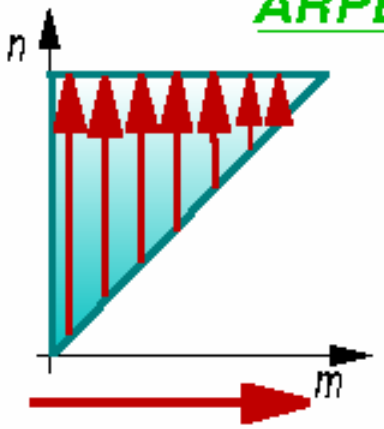
METEO FRANCE

In file ARPEGE :

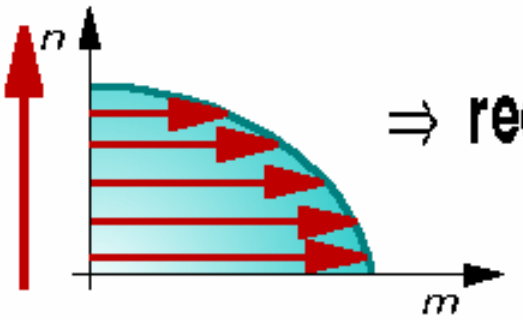


⇒ reordering ⇒

In model ARPEGE/IFS :

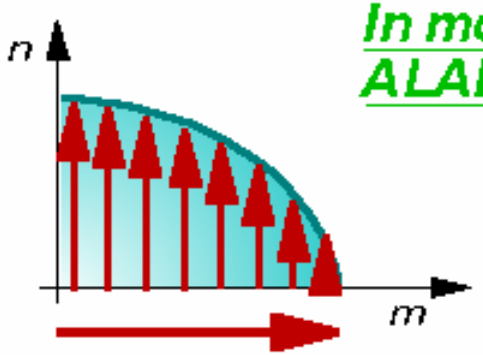


In file ALADIN :



⇒ reordering ⇒

In model ALADIN :



... Initial ordering has been modified in order to enable an easy distribution of data

Grid point arrays

- 2 data structures :
  - GMV : prognostic variables involved in SI
    u,v,T,ps (pd,vd) -> FIXED
  - GFL : other variables such as q, ql, qi, …
    ->
    FLEXIBLE

Grid point arrays

GMV(T1)(nproma,nflevg,nfields,ngpblks) ->3D variables

GMVS(T1)(nproma,nfields,ngpblks) -> 2D variables

Ex : GMV(:,:,YT0%MT,:)

Access to fields by pointers

YT0,YT9,YT1: pointers to t, t-dt, t+dt quantities

« Fields » pointers, ex: MU,MV, MDIV,MVOR, MT,

MSPD, MSVD

Grid point arrays

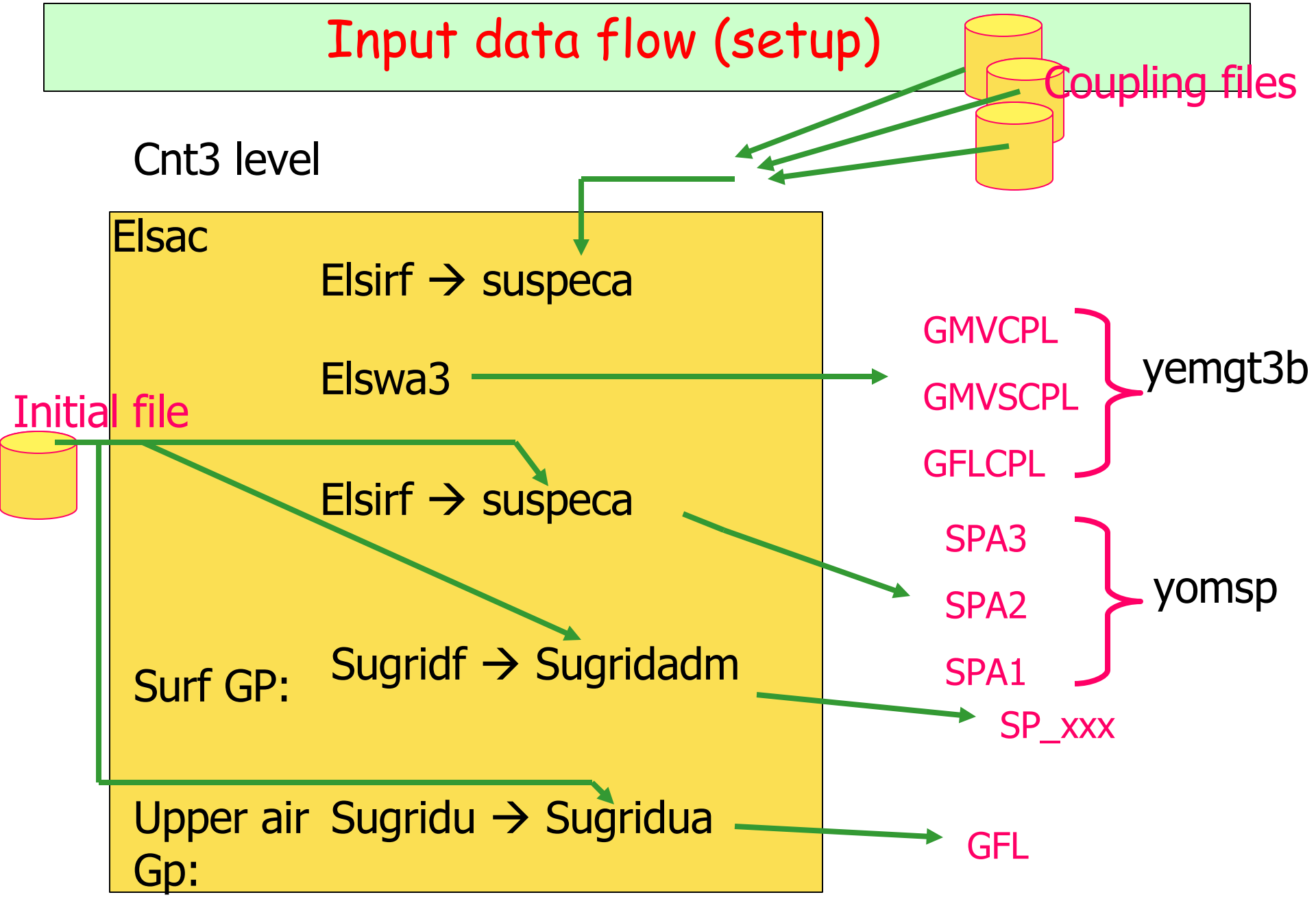
GFL(T1)(nproma,nflevg,nfields,ngpblks)
ex gfl(:,:,YQ%MPL,:)

« fields pointers » : YQ, YI, YL, …

« attributes »:

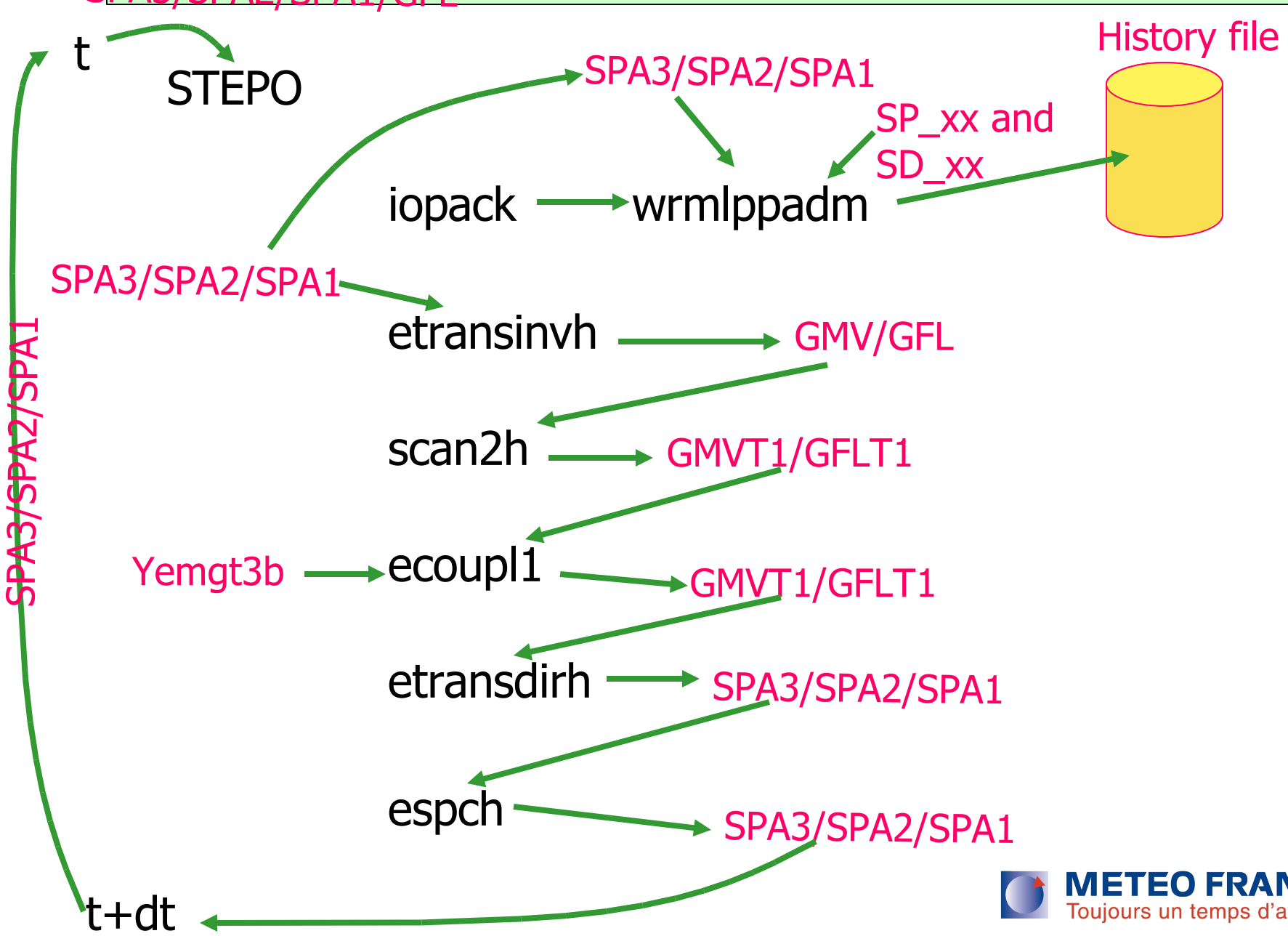
- MP : basic field pointer
- MPL : zonal derivative
- MPM: meridional derivative
- MPSP: basic field spectral space
- LADV (advec or not), LSP (spectral representation or not), LGP (gp or not), NREQIN (required in input or not), LREQOUT (saved in outpout files or not), NCOUPLING (coupled or not) …

Input data flow (setup)



METEO FRANCE
Toujours un temps d'avance

Data flow (stepo loop)



METEO FRANCE
Toujours un temps d'avance