WRF-Var System Overview

WRF-Var Tutorial

February 2 - 4, 2008

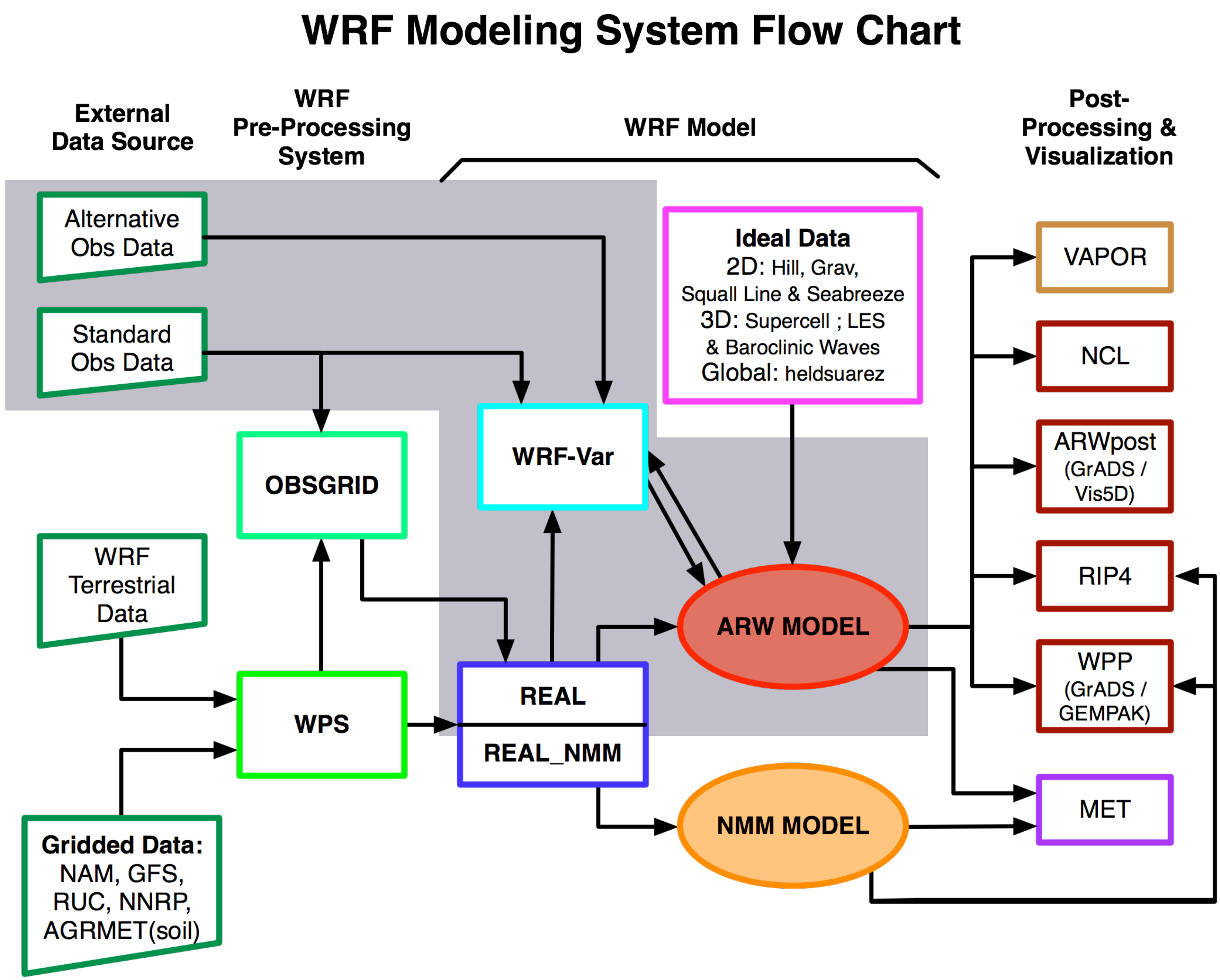
Xin Zhang and Syed RH Rizvi

Michael Duda

Outline

- WRF-Var in the WRF Modeling System
- WRF-Var Software
- WRF-Var Implementation

WRF-Var in the WRF Modeling System



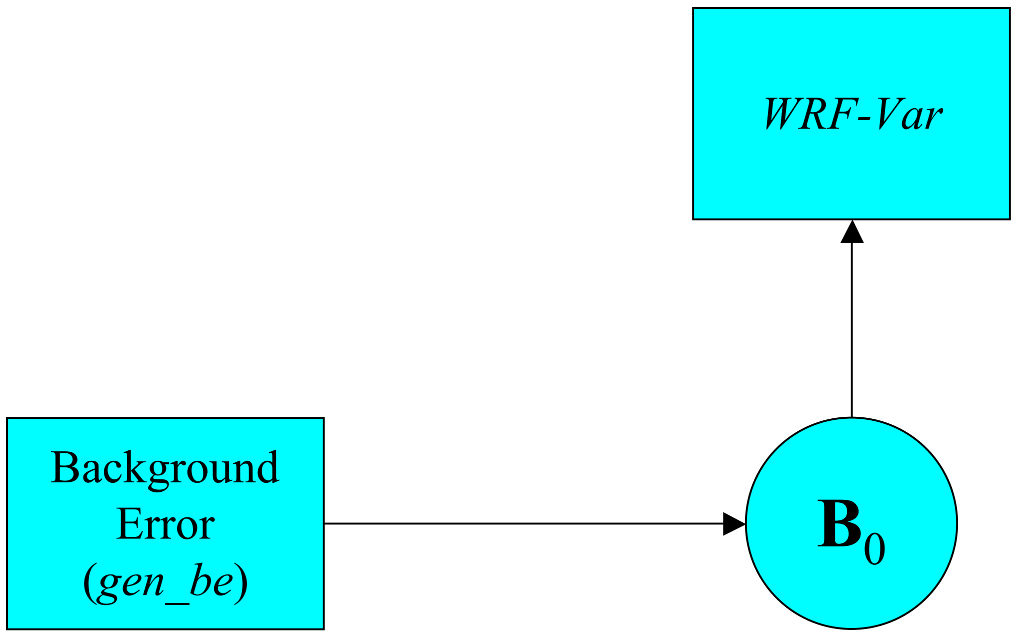
WRF-Var in the WRF Modeling System

WRF-Var

Blue --> Supported by WRF-Var Team

WRF-Var in the WRF Modeling System

1. Prepare BE data



Blue --> Supported by WRF-Var Team

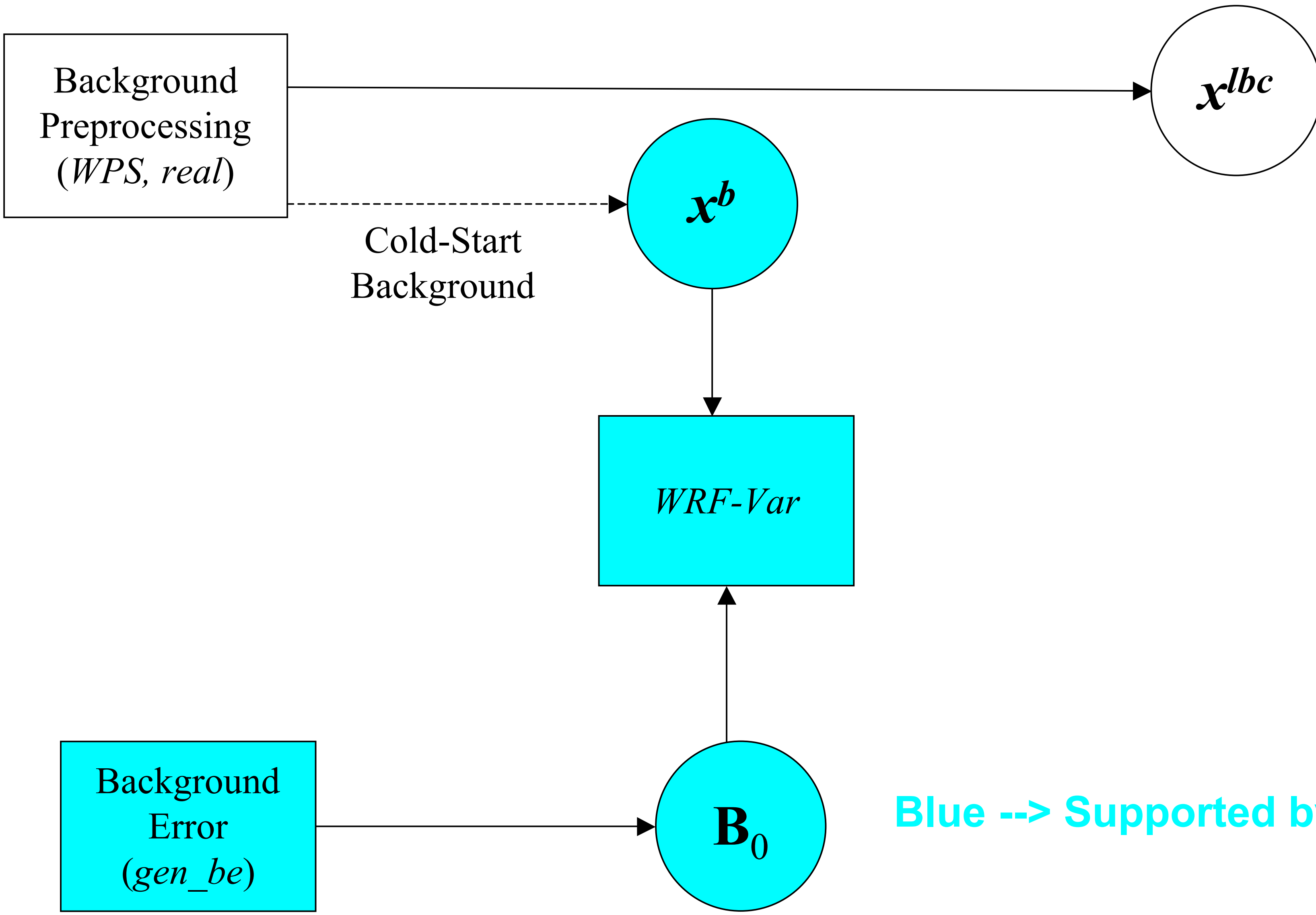
Prepare BE statistics

$$J(\mathbf{x}) = (\mathbf{x} - \mathbf{x}^b)^T \mathbf{B}^{-1} (\mathbf{x} - \mathbf{x}^b) + (\mathbf{y}^o - \mathbf{H}\mathbf{x})^T \mathbf{R}^{-1} (\mathbf{y}^o - \mathbf{H}\mathbf{x})$$

- For initial testing, the default background error statistics may be used
- Ultimately, these should be specific to the particular model domain (and season)
- See “WRF-Var Background Error Estimation” (Syed Rizvi)

WRF-Var in the WRF Modeling System

2. Prepare background (WPS and real)



Blue --> Supported by WRF-Var Team

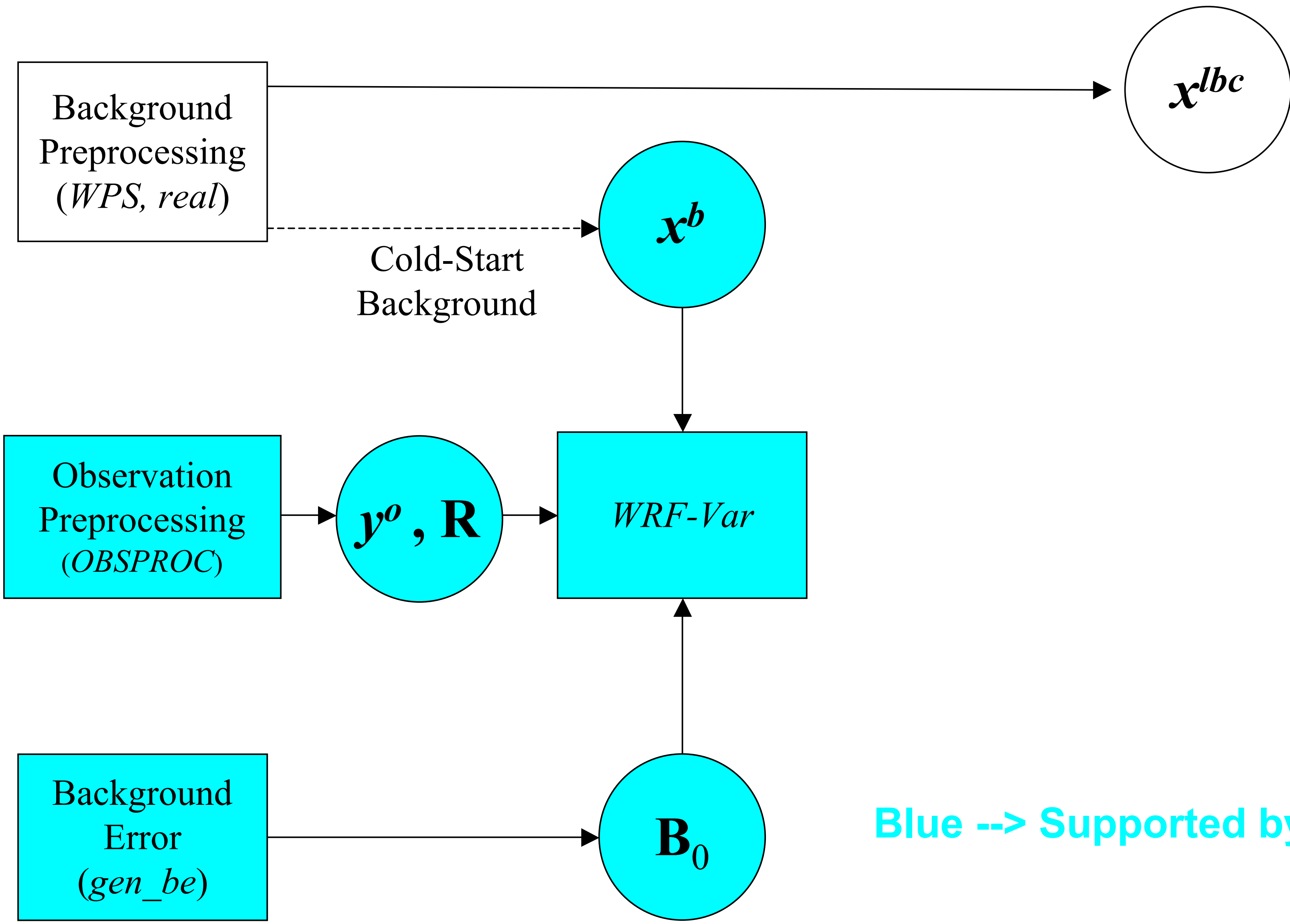
Prepare background

$$J(x) = (x - \mathbf{x}^b)^T B^{-1} (x - \mathbf{x}^b) + (y^o - Hx)^T R^{-1} (y^o - Hx)$$

- In “cold-start” mode: accomplished by running the WPS and *real* programs
  - The background is essentially the wrfinput_d01 file
- In “cycling” mode: the output of the WRF model
  - WRF can output wrfinput-formatted files used for cycling

WRF-Var in the WRF Modeling System

3. Prepare observations (run OBSPROC)



Blue --> Supported by WRF-Var Team

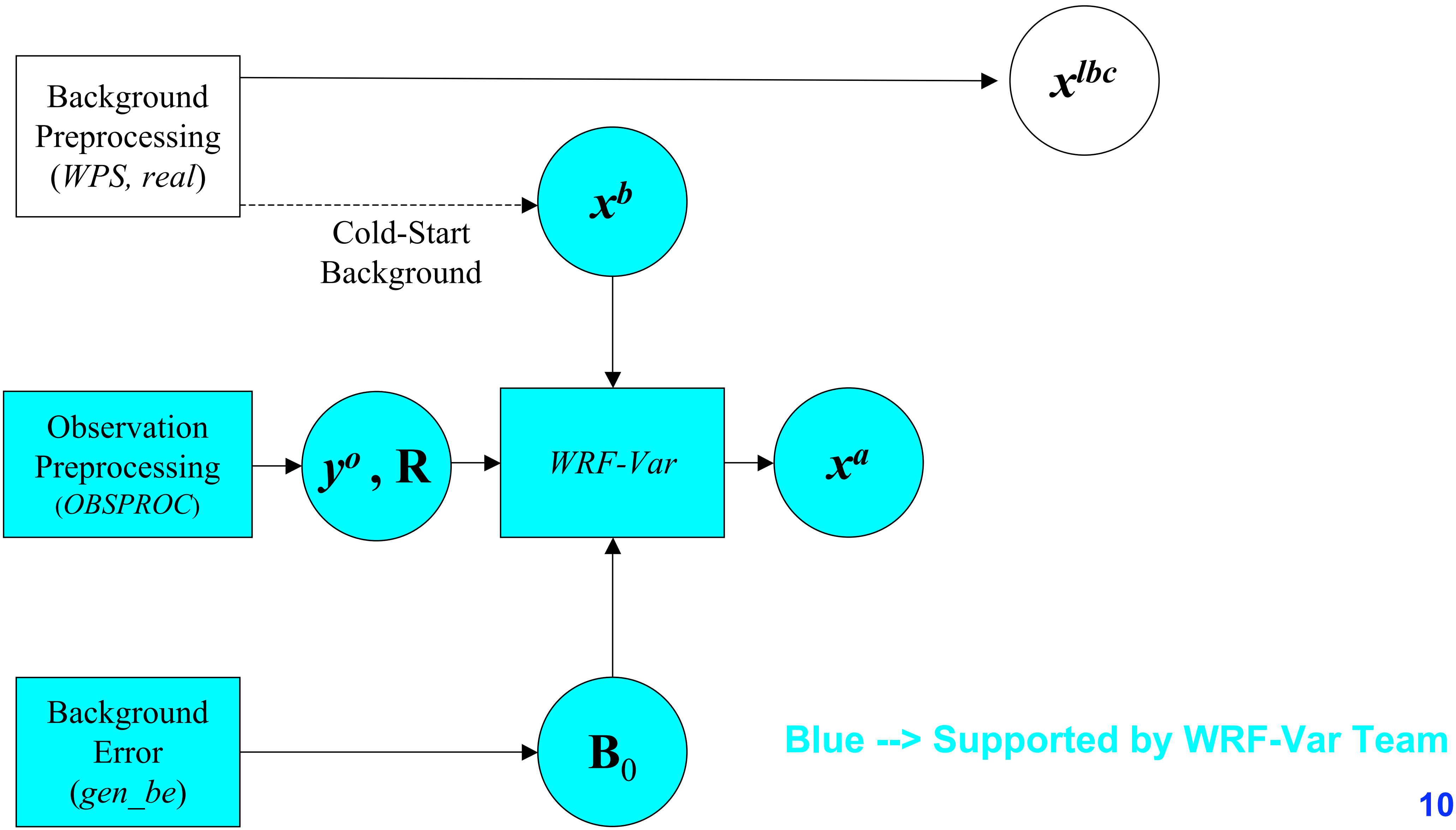
Prepare observations (y^0)

$$J(x) = (\mathbf{x} - \mathbf{x}^b)^T \mathbf{B}^{-1} (\mathbf{x} - \mathbf{x}^b) + (\mathbf{y}^o - \mathbf{Hx})^T \mathbf{R}^{-1} (\mathbf{y}^o - \mathbf{Hx})$$

- Observation input for WRF-Var is supplied through observation preprocessor, OBSPROC
- Observation error covariance also provided by OBSPROC (**R** is a diagonal matrix)
- Separate input file (ASCII) for Radar, both reflectivity and radial velocity.
- See “Observation Pre-Processing” (Yong-Run Guo)

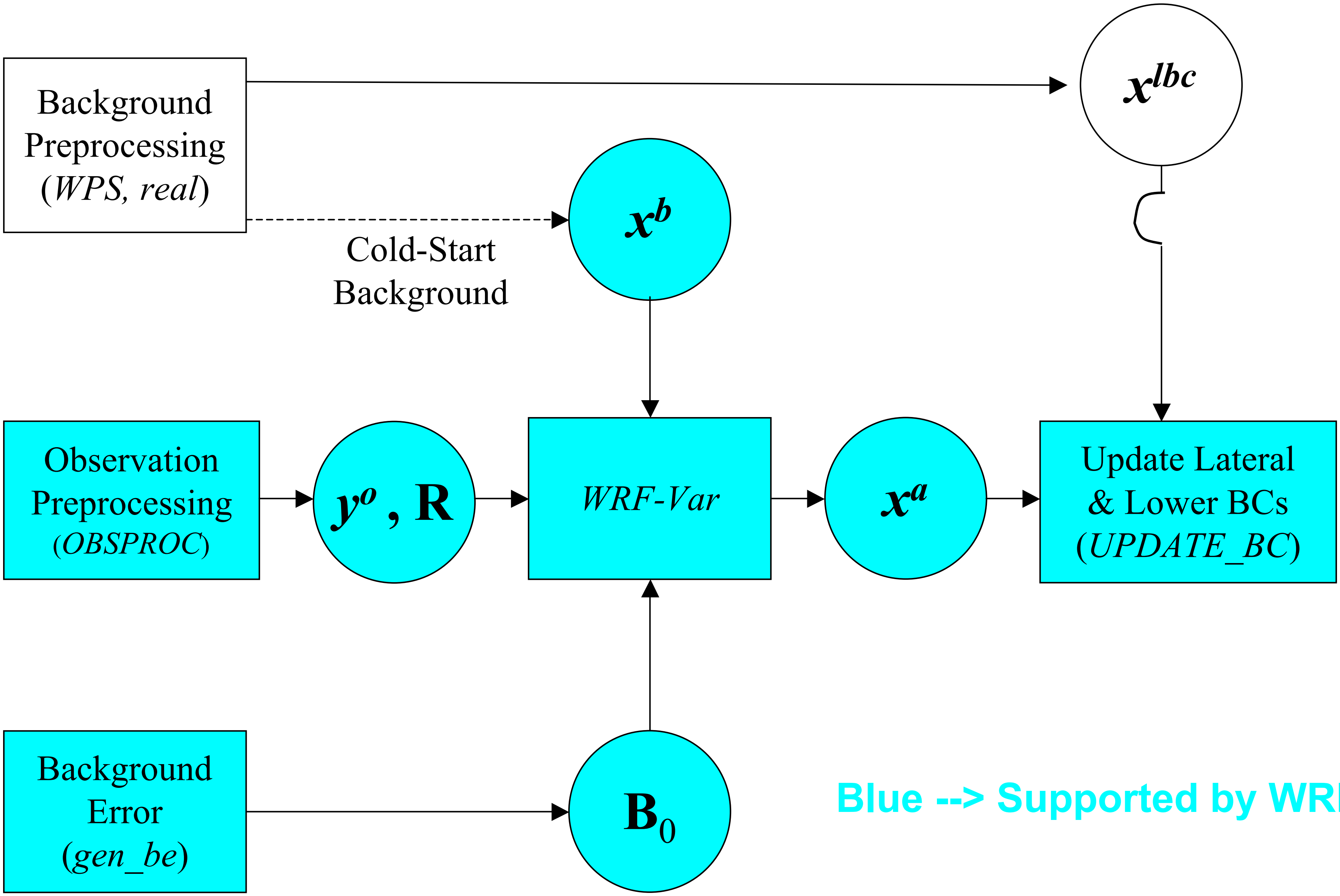
WRF-Var in the WRF Modeling System

4. Run WRF-Var



WRF-Var in the WRF Modeling System

5. Update boundary conditions (UPDATE_BC)



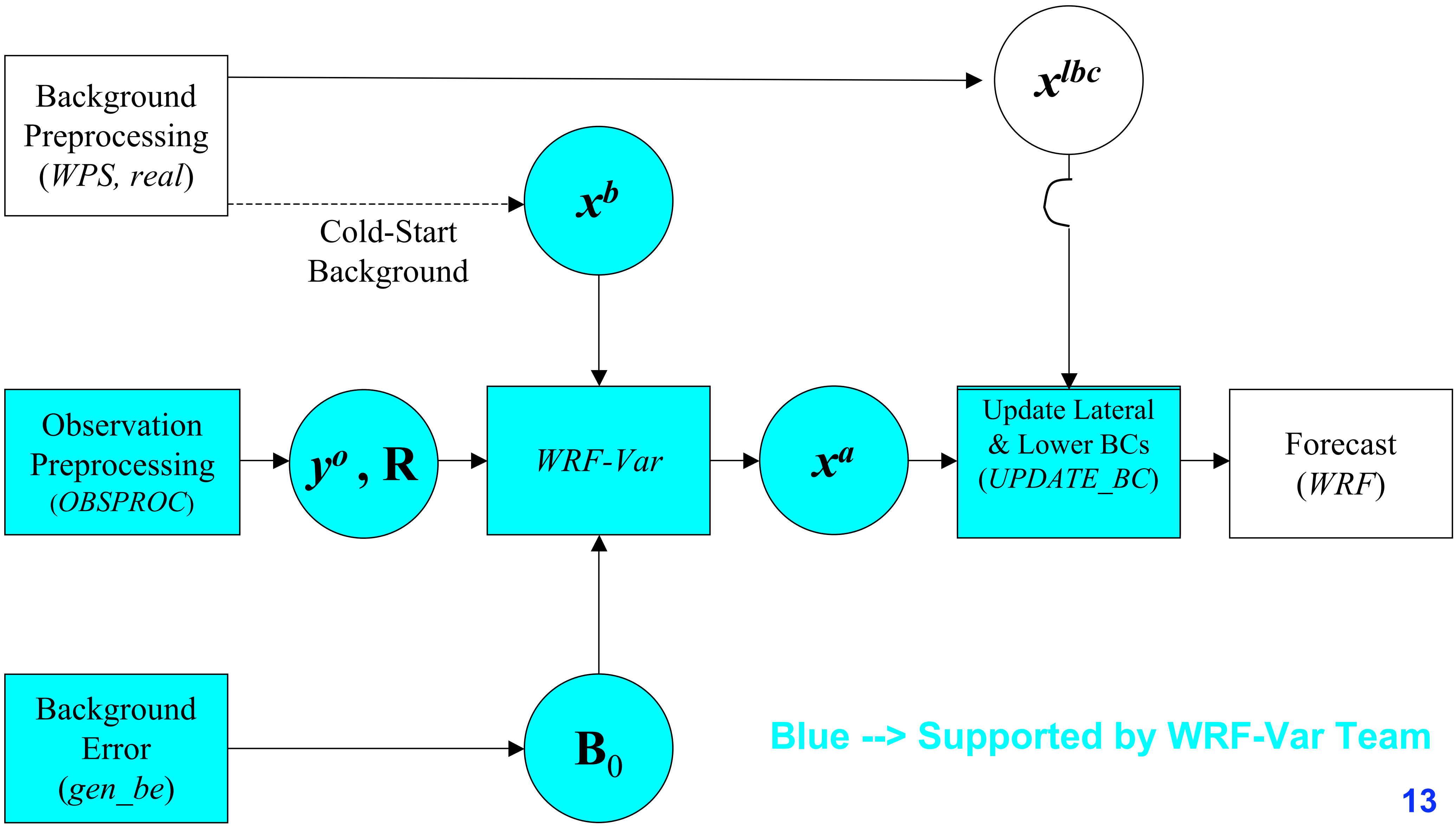
Blue --> Supported by WRF-Var Team

Update boundary conditions

- After creating an analysis, $\mathbf{x}^a$, we have changed the initial conditions for the model
  - However, tendencies in wrfbdy_d01 (and possibly wrflbdy) file are valid for background, $\mathbf{x}^b$
- The update_bc program adjusts these tendencies based on the difference $\mathbf{x}^a - \mathbf{x}^b$
- Of course, if $\mathbf{x}^a$ was produced for reasons other than running WRF, there is probably not a need to update boundary conditions

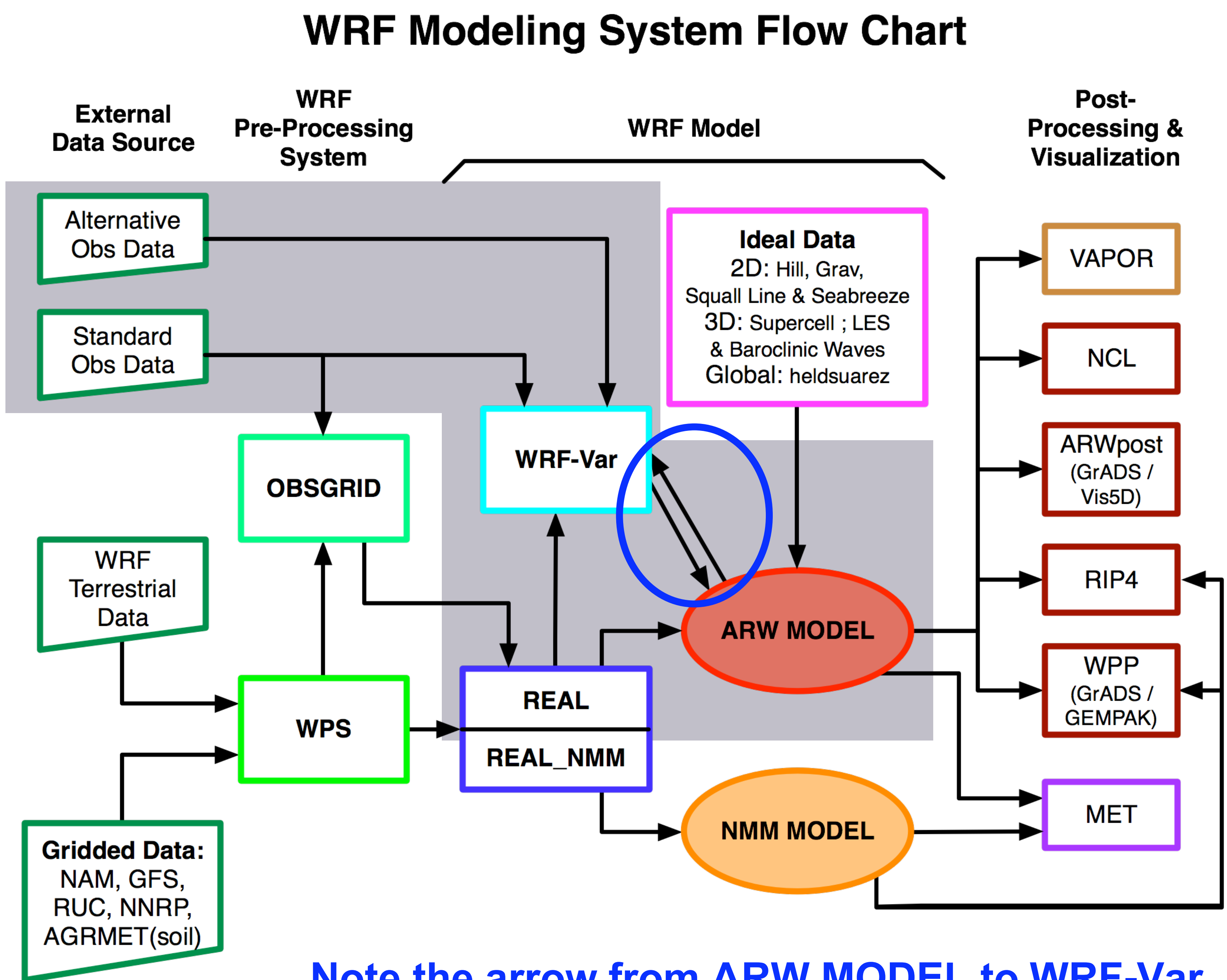
WRF-Var in the WRF Modeling System

6a. Run forecast (cold-start mode)



Blue --> Supported by WRF-Var Team

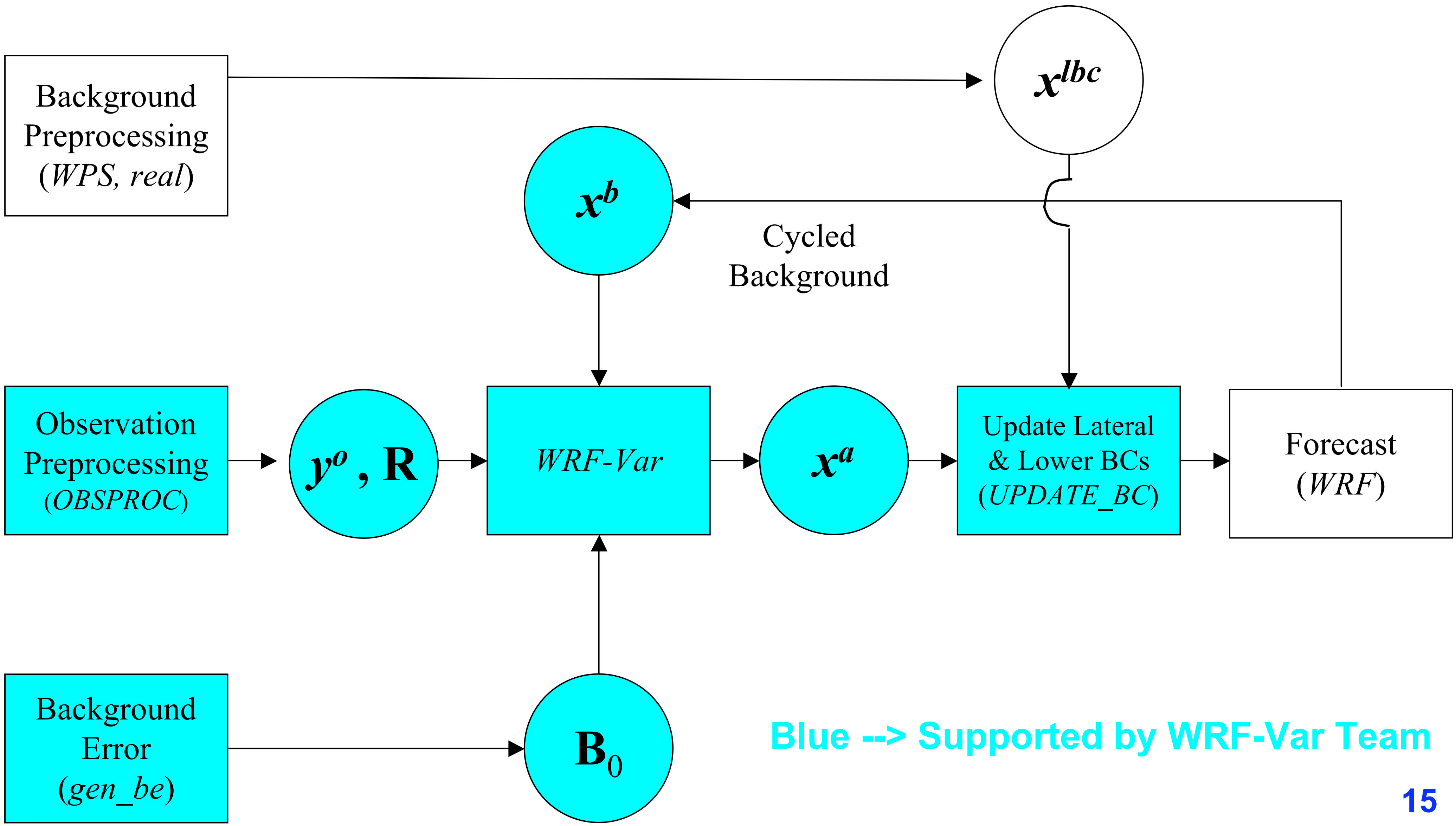
WRF-Var in the WRF Modeling System



Note the arrow from ARW MODEL to WRF-Var

WRF-Var in the WRF Modeling System

6b. Run forecast (cycling mode)



Blue --> Supported by WRF-Var Team

Background Error (BE) for WRF-Var

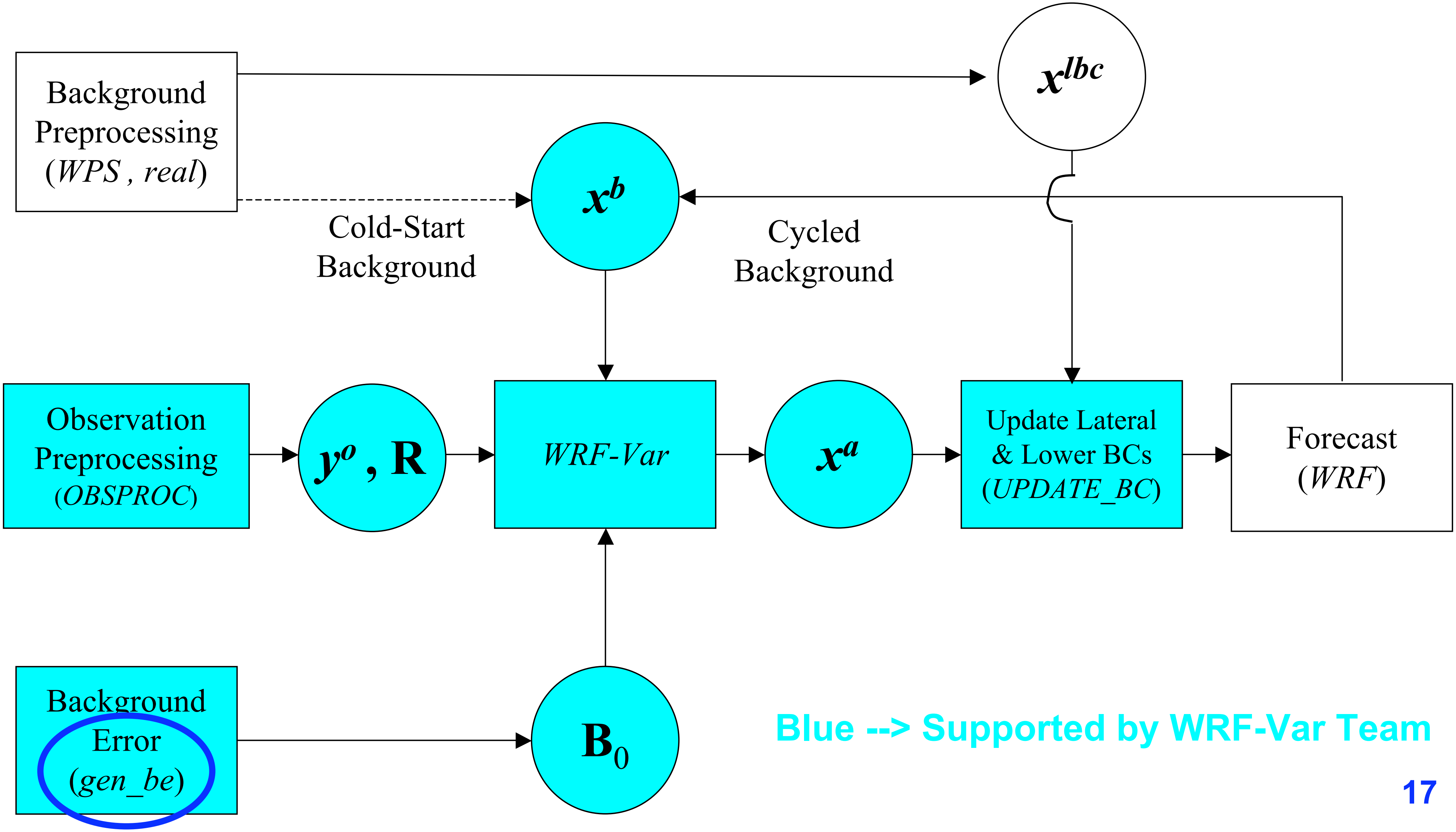
The number 1 question from WRF-Var users is
"What background error are best for my application?"

- Create your own once you have run your system for a few weeks to a month
- Implement, tune, and iterate

A new utility *gen_be* has been developed at NCAR to calculate BEs

WRF-Var in the WRF Modeling System

7. WRF-Var/WRF Ultimate Configuration



Blue --> Supported by WRF-Var Team

WRF-Var Software

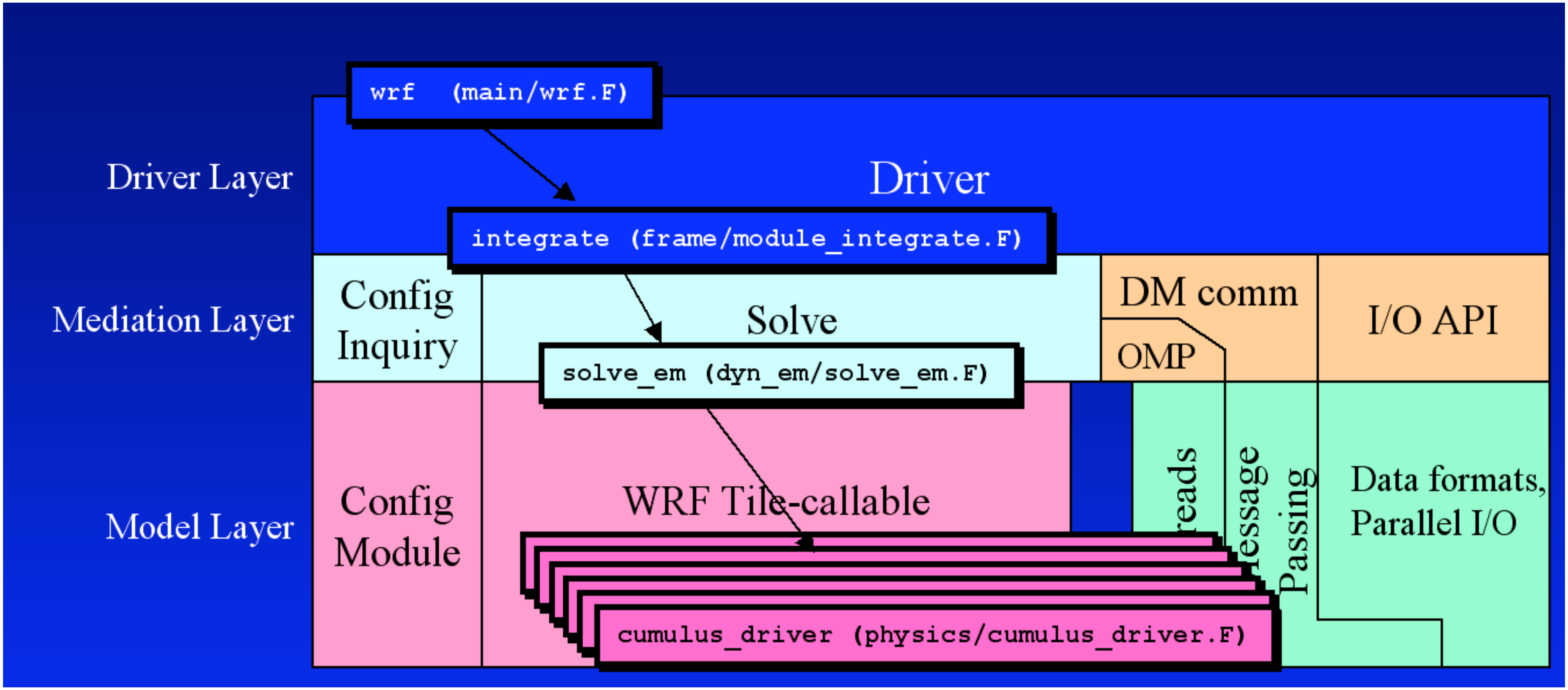
Supported Platforms and Compilers

Generally speaking, the following should work "out of the box" :

- IBM (AIX/xlf)
- PC (Linux/PGI, Linux/ifort, Linux/g95)
- Macintosh (OS X/PGI, OS X/g95)
- SGI (IRIX/ifort) *(tested by 3rd party)*

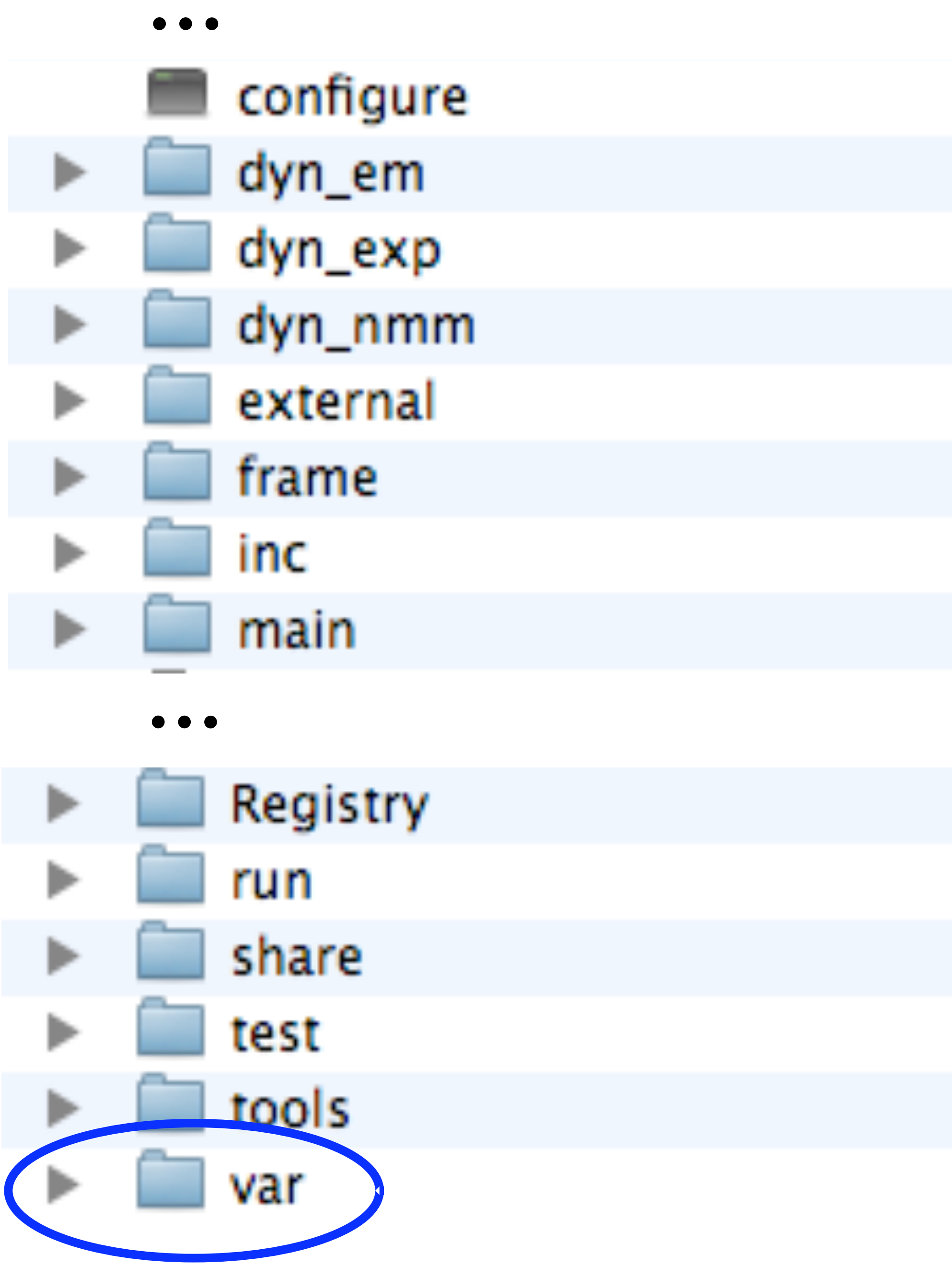
Only serial and dmpar are supported!

Use of the WRF Software Framework



WRF → **WRF-Var**
Integrate → **da_wrfvar_interface**
Solve_em → **da_solve**
Cumulus_driver → **obs. (da_ships) or da_minimisation etc.**

WRF-Var Code Organization



Besides the directories for WRF, the WRFDA tar file contains a "var" directory, which holds all of the WRF-Var code

WRF-Var Code Organization

- var
  - convertor
  - da
  - gen_be
  - graphics
  - Makefile
  - obsproc
  - scripts
  - test

- da
  - da_airep
  - da_airsr
  - da_bogus
  - da_buoy
  - da_control
  - da_define_structures
  - …
  - da_metar
  - da_minimisation
  - da_mtgirs
  - da_obs
  - …
  - da_setup_structures
  - da_ships
  - da_sound

Generally, each subdirectory of “da” contains a Fortran module with the same name

WRF-Var Code Organization

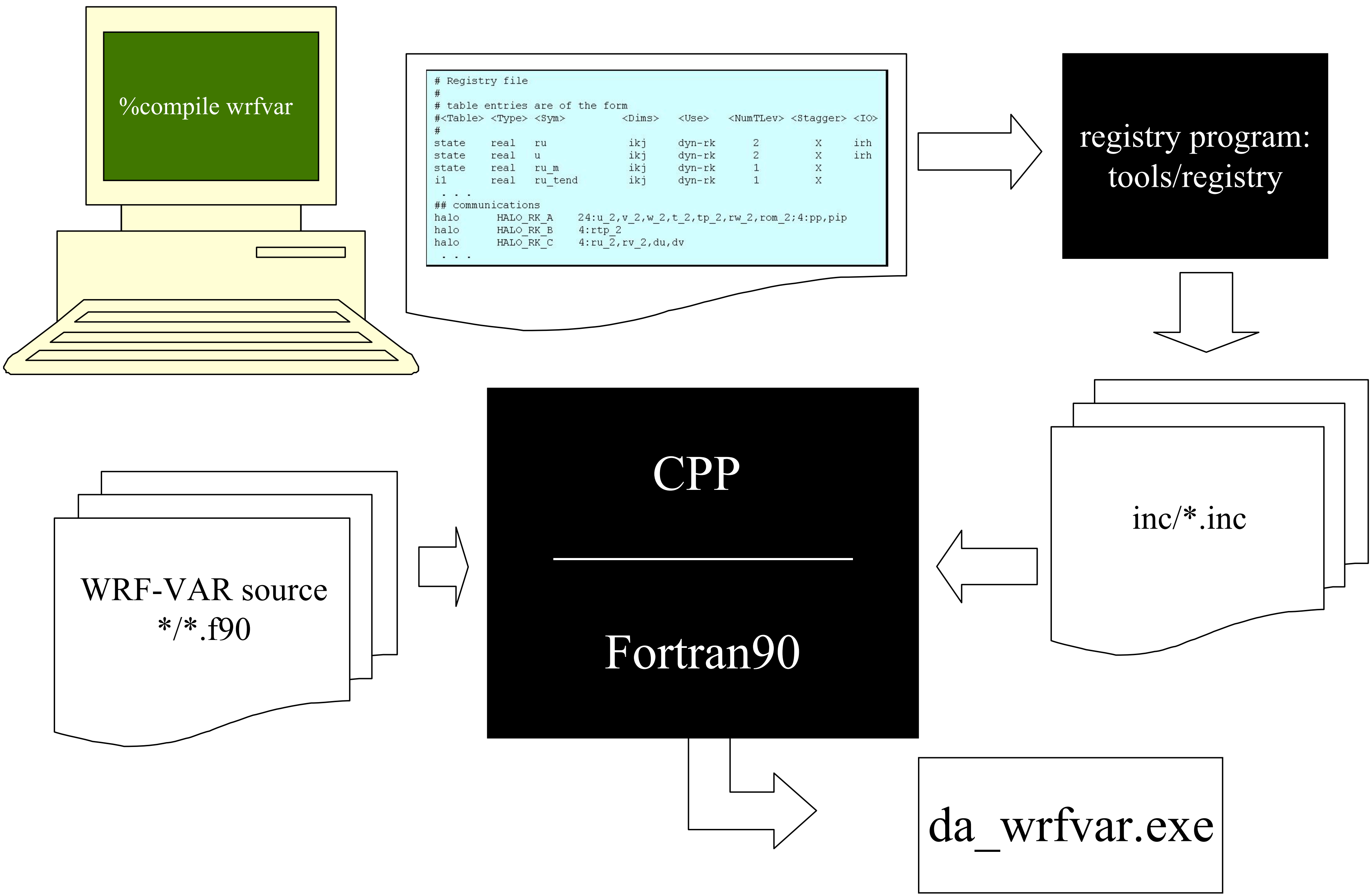
- da_metar
  - da_ao_stats_metar.inc
  - da_calculate_grady_metar.inc
  - da_check_max_iv_metar.inc
  - da_get_innov_vector_metar.inc
  - da_jo_and_grady_metar.inc
  - da_metar.f90
  - da_oi_stats_metar.inc
  - da_print_stats_metar.inc
  - da_residual_metar.inc
  - da_transform_xtoy_metar_adj.inc
  - da_transform_xtoy_metar.inc

- **da_metar.f90 contains a Fortran module**
- **Each .inc file corresponds to a subroutine within the module**

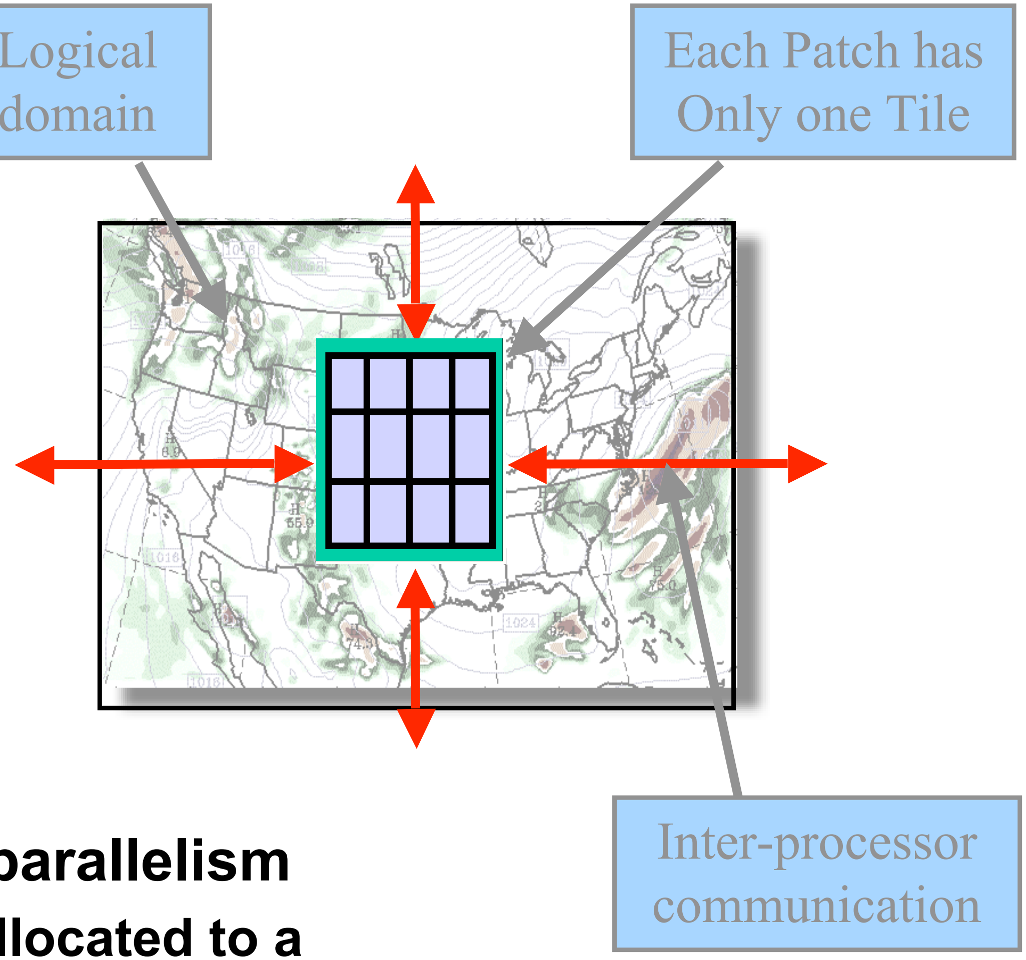
WRF-Var Registry

- "Active data-dictionary" for managing WRF-Var data structures
  - Database describing attributes of model state, intermediate, and configuration data
    - Dimensionality, number of time levels, staggering
    - Association with physics
    - I/O classification (history, initial, restart, boundary)
    - Communication points and patterns
    - Configuration lists (e.g. namelists)
  - Program for auto-generating sections of WRF from database:
    - Argument lists for driver layer/mediation layer interfaces
    - Interprocessor communications: Halo and periodic boundary updates, transposes
    - Code for defining and managing run-time configuration information
- Automates time consuming, repetitive, error-prone programming
- Insulates programmers and code from package dependencies
- Allow rapid development
- Documents the data

Registry Mechanics



Parallelism in WRF-Var: MPI Decomposition



Single version of code for efficient execution on:

- **Distributed-memory**
- **Vector and microprocessors**

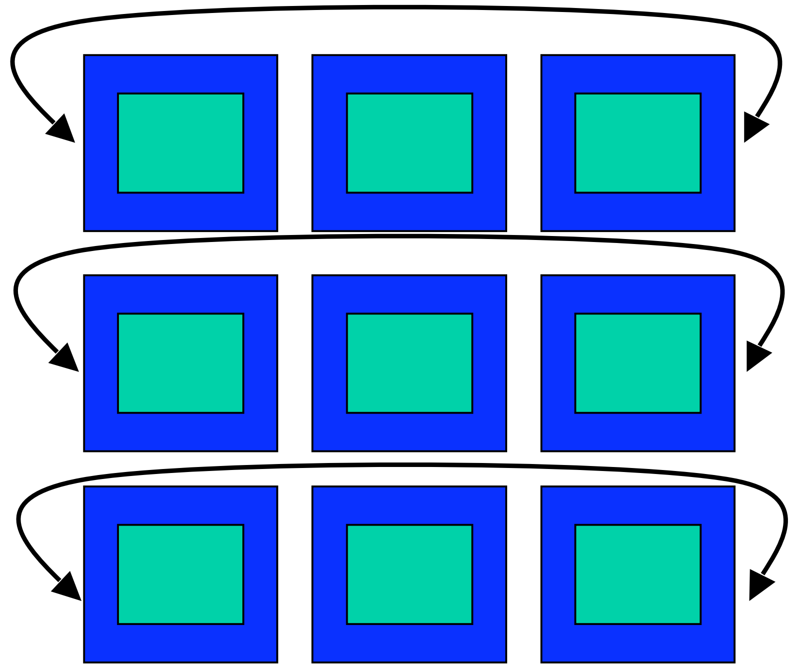
Model domain is decomposed for parallelism

Patch: **section of model domain allocated to a distributed memory node**

Tile: **same as patch in WRF-Var**

Distributed Memory Communications

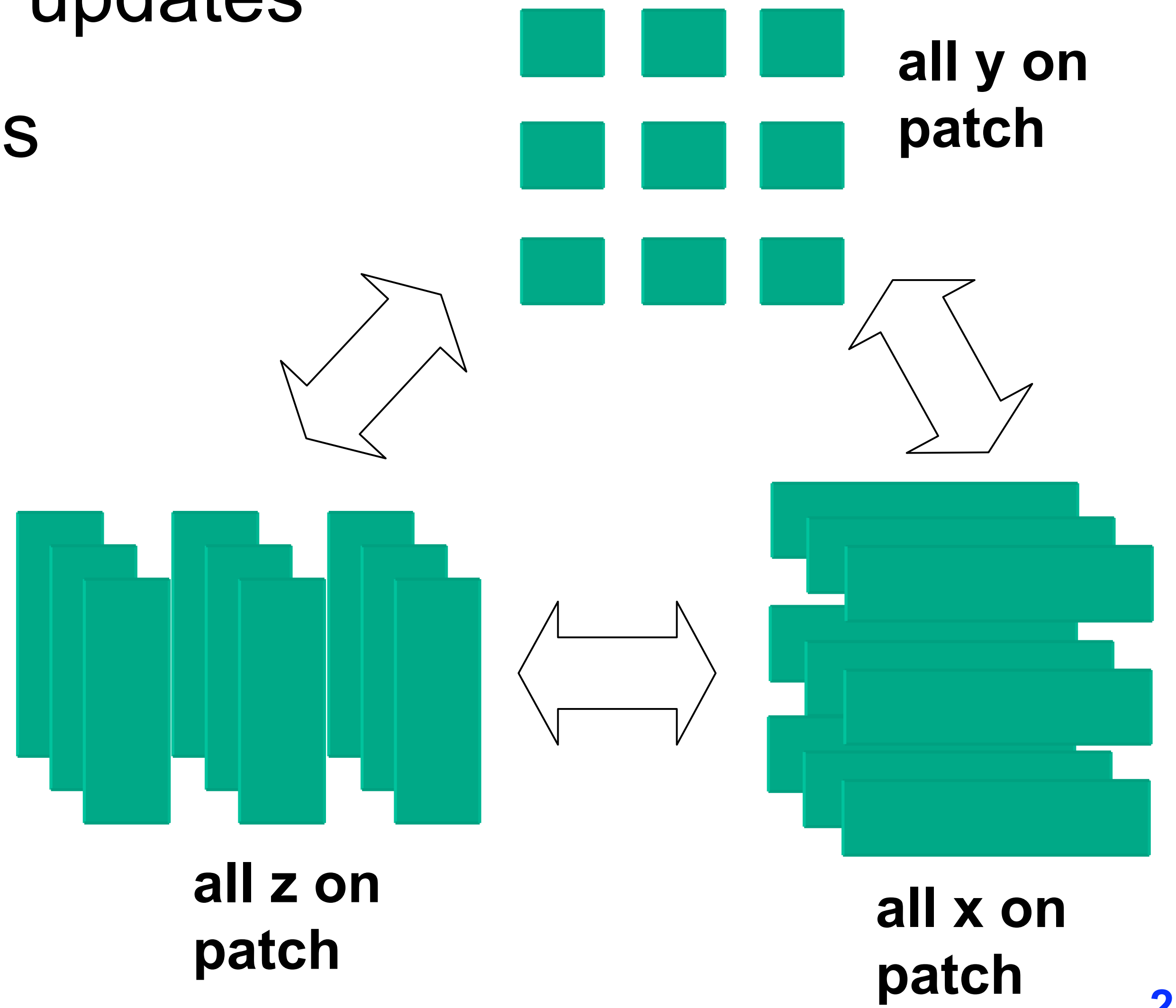
- Halo updates
- Periodic boundary updates (only needed for global 3dvar)



Distributed Memory Communications

- Halo updates
- Periodic boundary updates
- Parallel transposes
- “nproc_x = 1”

  (for global option)



WRF-Var Implementation

WRF-Var Formulation

- WRF-Var actually uses an incremental formulation of the 3DVAR problem

$$J(\mathbf{x}) = (\mathbf{x} - \mathbf{x}^b)^T\mathbf{B}^{-1}(\mathbf{x} - \mathbf{x}^b) + (\mathbf{y}^o - H(\mathbf{x}))^T\mathbf{R}^{-1}(\mathbf{y}^o - H(\mathbf{x}))$$

- Define the increment $\mathbf{x'} = \mathbf{x} - \mathbf{x}^b$
- Also, if $\mathbf{x'}$ is small, $H(\mathbf{x}) = H(\mathbf{x}^b + \mathbf{x'}) \approx H(\mathbf{x}^b) + \mathbf{Hx'}$ where $\mathbf{H}$ is the linearization of H
- Then, the problem becomes

$$J(\mathbf{x'}) = (\mathbf{x'})^T\mathbf{B}^{-1}(\mathbf{x'}) + (\mathbf{y}^{o\prime} - \mathbf{Hx'})^T\mathbf{R}^{-1}(\mathbf{y}^{o\prime} - \mathbf{Hx'})$$

with $\mathbf{y}^{o\prime} = \mathbf{y}^o - H(\mathbf{x}^b)$

WRF-Var Formulation

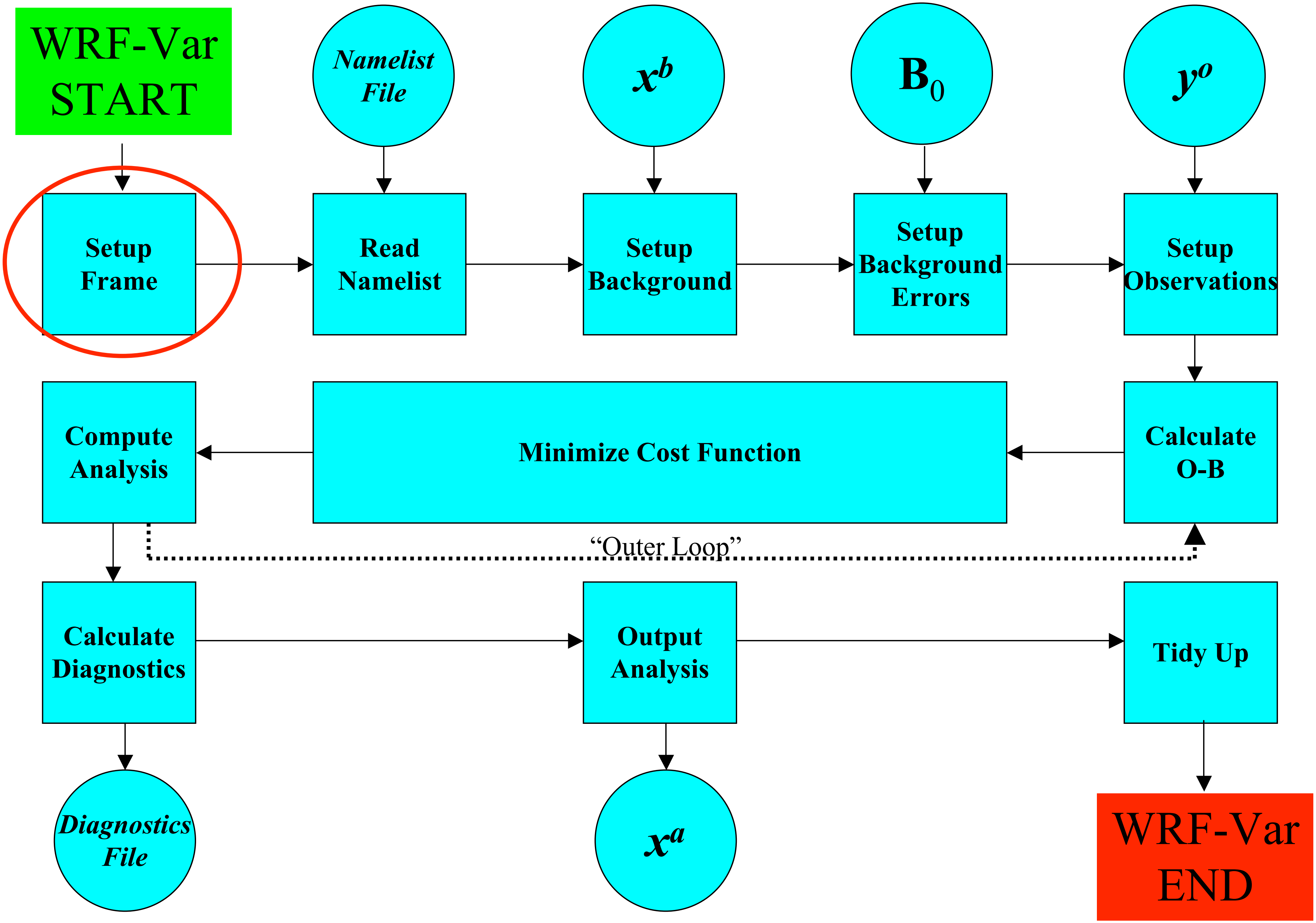
- Next, define the *control variable transform* **U** such that $\mathbf{x}' = \mathbf{U}\mathbf{v}$.
  - **v** is the analysis increment in control variable space
  - **B** is approximated by $\mathbf{U}\mathbf{U}^T$
- WRF-Var actually minimizes

$$J(\mathbf{v}) = \mathbf{v}^T\mathbf{v} + (\mathbf{y}^{o\prime} - \mathbf{H}\mathbf{U}\mathbf{v})^T\mathbf{R}^{-1}(\mathbf{y}^{o\prime} - \mathbf{H}\mathbf{U}\mathbf{v})$$

- After minimization, the analysis, $\mathbf{x}^a$, is given by

$$\mathbf{x}^a = \mathbf{x}^b + \mathbf{U}\mathbf{v}$$

WRF-Var

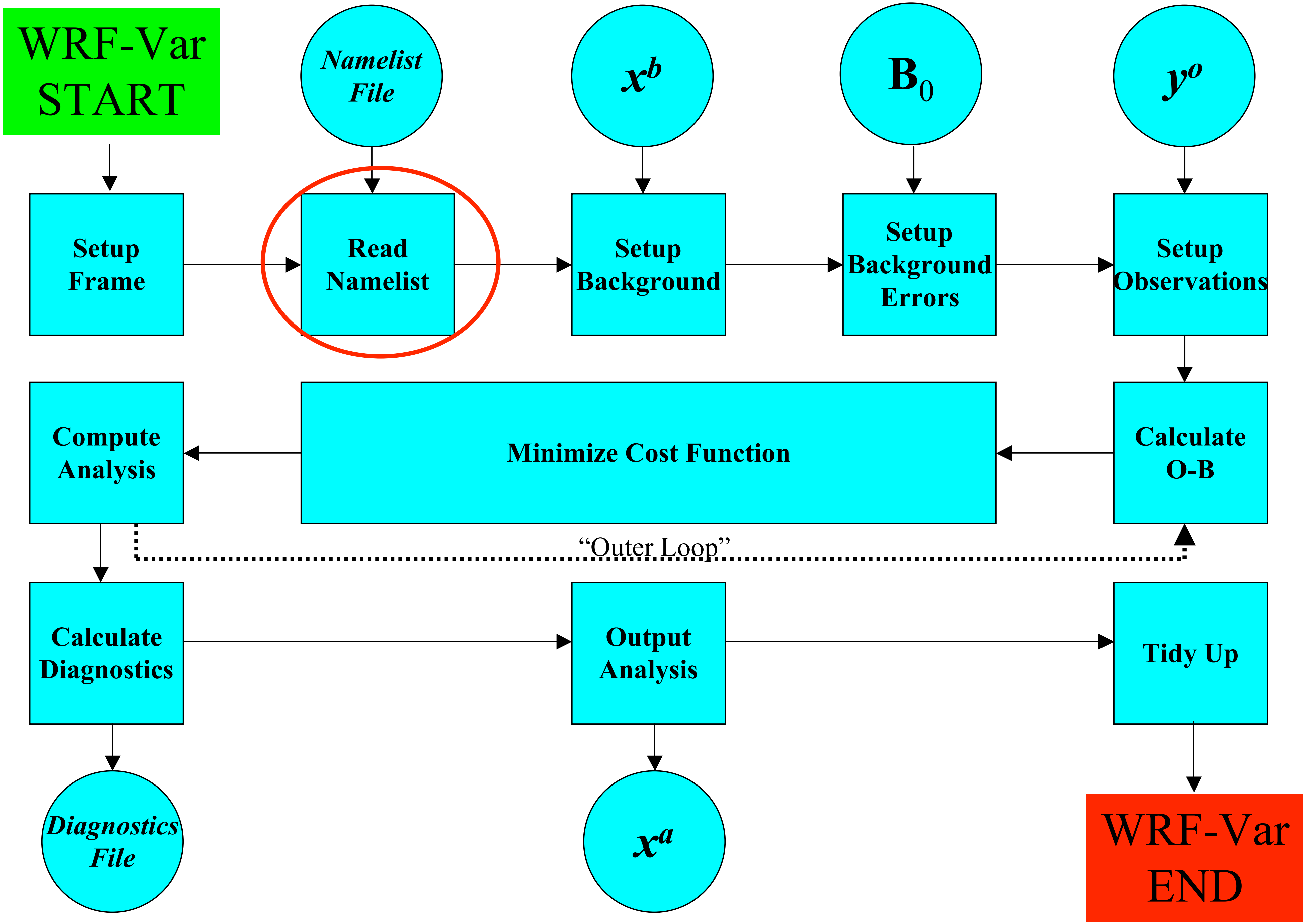


WRF-Var START
Namelist File
x^b
B_0
y^o
Setup Frame
Read Namelist
Setup Background
Setup Background Errors
Setup Observations
Compute Analysis
Minimize Cost Function
Calculate O-B
"Outer Loop"
Calculate Diagnostics
Output Analysis
Tidy Up
Diagnostics File
x^a
WRF-Var END

Setup Frame

- Reads grid dimensions from “namelist.input” file.
- Use WRF framework’s distributed memory capability to initialize tile, memory, patch dimensions, etc.

WRF-Var


WRF-Var START
Namelist File
x^b
B_0
y^o
Setup Frame
Read Namelist
Setup Background
Setup Background Errors
Setup Observations
Compute Analysis
Minimize Cost Function
Calculate O-B
"Outer Loop"
Calculate Diagnostics
Output Analysis
Tidy Up
Diagnostics File
x^a
WRF-Var END

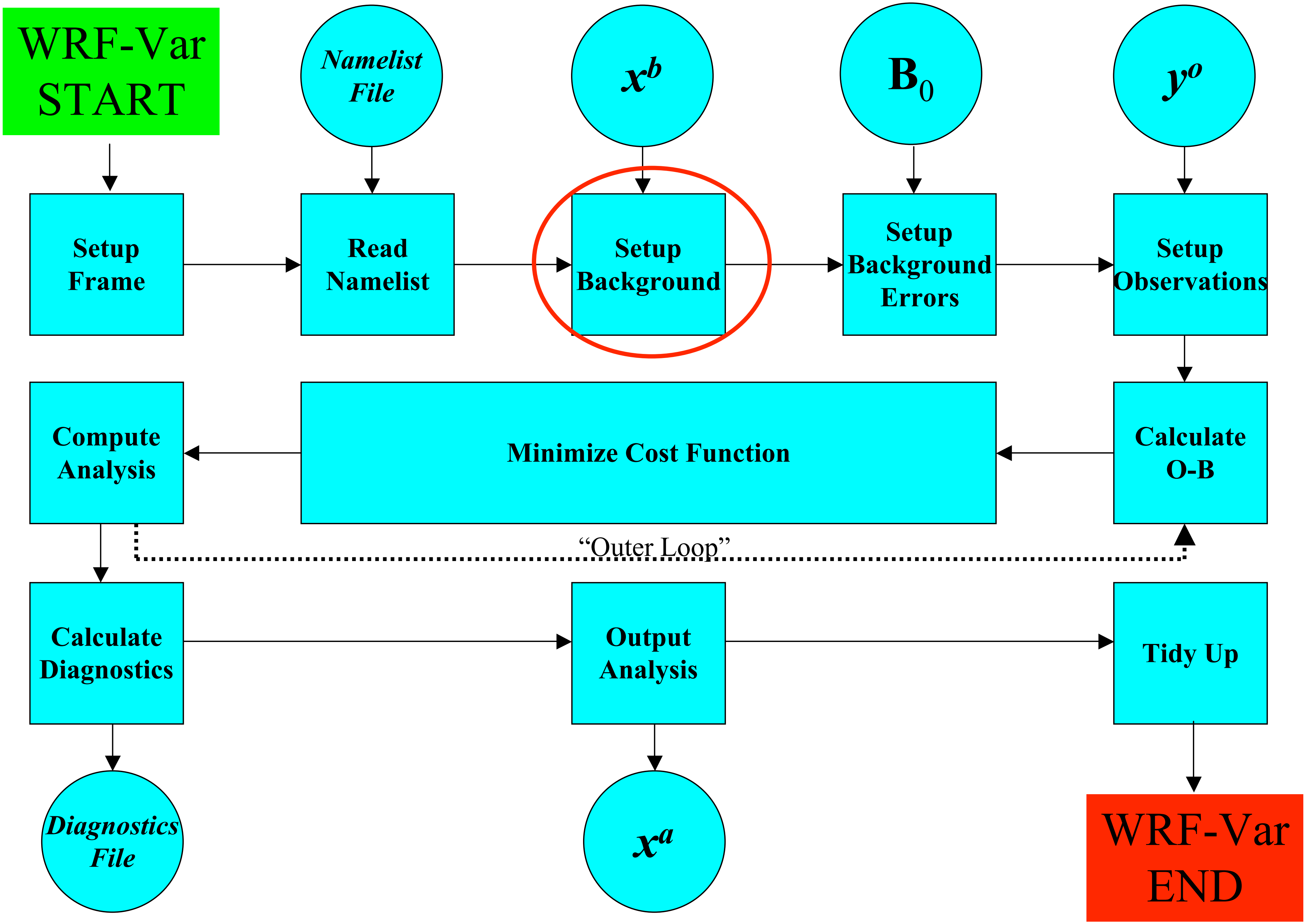
Read Namelist

- Reads WRF-Var data assimilation options from “namelist.input” file.
- Performs consistency checks between namelist options.

namelist.input

```
&wrfvar1
var4d=true,
multi_inc=0,
var4d_coupling=2,
global=false,
/
&wrfvar2
/
&wrfvar3
ob_format=2,
num_fgat_time=7,
/
&wrfvar4
use_synopobs=true,
use_shipsobs=true,
use_metarobs=true,
use_soundobs=true,
use_pilotobs=true,
use_airepobs=true,
use_geoamvobs=true,
use_polaramvobs=true,
use_bogusobs=true,
use_buoyobs=true,
use_profilerobs=true,
use_satemobs=true,
use_gpspwobs=true,
use_gpsrefobs=true,
use_qscatobs=true,
use_radarobs=false,
use_radar_rv=false,
use_radar_rf=false,
/
&wrfvar5
check_max_iv=true,
/
&wrfvar6
ntmax=100,
/
&wrfvar7
/
&wrfvar8
/
&wrfvar9
/
&wrfvar10
/
&wrfvar11
cv_options_hum=1,
check_rh=2,
seed_array1=2007081421,
seed_array2=2007081421,
/
&wrfvar12
/
&wrfvar13
/
&wrfvar14
/
&wrfvar15
/
&wrfvar16
/
&wrfvar17
analysis_type="3D-VAR",
/
&wrfvar18
analysis_date="2007-08-
14_21:00:00.0000",
/
&wrfvar19
/
&wrfvar20
/
&wrfvar21
time_window_min="2007-08-
14_21:00:00.0000",
/
&wrfvar22
time_window_max="2007-08-
15_03:00:00.0000",
/
&wrfvar23
jcdfi_use=false,
jcdfi_io=false,
jcdfi_tauc=21600,
jcdfi_gama=0.1,
jcdfi_error_wind=3.0,
jcdfi_error_t=1.0,
jcdfi_error_q=0.001,
jcdfi_error_mu=1000.,
/
&time_control
run_hours=27,
run_minutes=0,
run_seconds=0,
start_year=2007,
start_month=08,
start_day=14,
start_hour=21,
start_minute=00,
start_second=00,
end_year=2007,
end_month=08,
end_day=15,
end_hour=03,
end_minute=00,
end_second=00,
interval_seconds=21600,
input_from_file=true,
frames_per_outfile=1,
debug_level=0,
history_interval_s=10800,
inputout_interval_s=3600,
auxinput3_interval_s=3600,
inputout_begin_h=6,
inputout_begin_m=0,
inputout_begin_s=0,
inputout_end_h=6,
inputout_end_m=0,
inputout_end_s=0,
input_outname="wrfinput_d<domain
>_<date>",
write_input=true,
/
&fdda
/
&domains
time_step=240,
e_we=123,
e_sn=111,
e_vert=27,
num_metgrid_levels=27,
p_top_requested=5000,
interp_type=1,
t_extrap_type=1,
dx=45000,
dy=45000,
i_parent_start=0,
j_parent_start=0,
smooth_option=0,
nproc_x=0,
eta_levels= 1.000, 0.990, 0.978,
0.964, 0.946, 0.922, 0.894, 0.860,
0.817, 0.766, 0.707, 0.644, 0.576,
0.507, 0.444, 0.380, 0.324, 0.273,
0.228, 0.188, 0.152, 0.121, 0.093,
0.069, 0.048, 0.029, 0.014, 0.000,
/
&physics
mp_physics=3,
ra_lw_physics=1,
ra_sw_physics=1,
radt=45,
sf_sfclay_physics=1,
sf_surface_physics=1,
bl_pbl_physics=1,
cu_physics=1,
cudt=1,
num_soil_layers=5,
mp_zero_out=0,
co2tf=0,
/
&dynamics
w_damping=1,
diff_opt=1,
km_opt=1,
dampcoef=0.0,
time_step_sound=4,
base_temp=290.0,
/
&bdy_control
specified=true,
real_data_init_type=3,
/
&grib2
/
```

WRF-Var

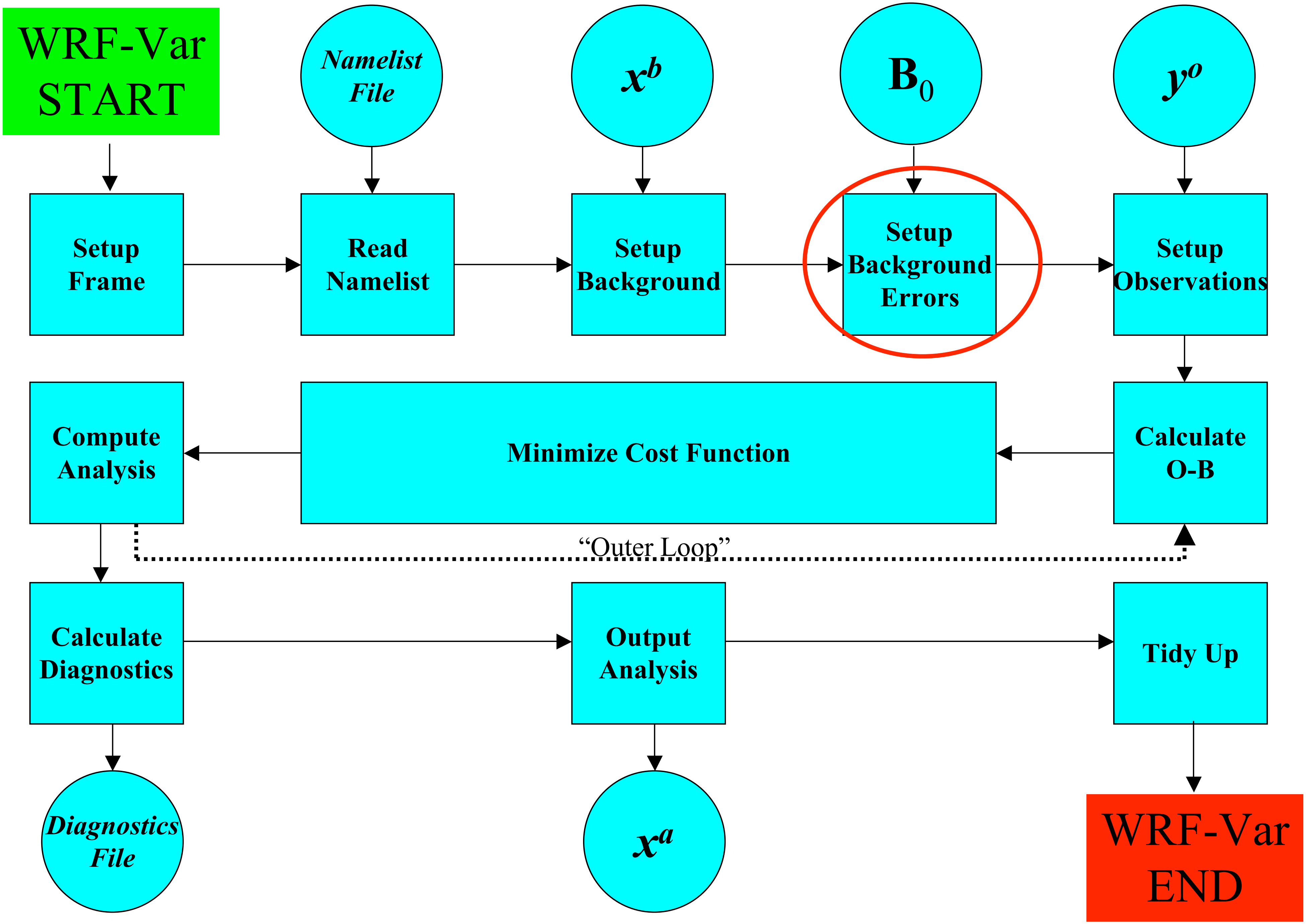


WRF-Var START
Namelist File
x^b
$\mathbf{B}_0$
y^o
Setup Frame
Read Namelist
Setup Background
Setup Background Errors
Setup Observations
Compute Analysis
Minimize Cost Function
Calculate O-B
"Outer Loop"
Calculate Diagnostics
Output Analysis
Tidy Up
Diagnostics File
x^a
WRF-Var END

Setup Background (First-guess)

- Reads in the first-guess field.
- Format depends on namelist option : "fg_format" ; 1= WRF, etc.
- Extracts necessary fields.
- Creates background FORTRAN 90 derived data type "xb" e.g. xb % mix, xb % u(:,:,:), ….

WRF-Var

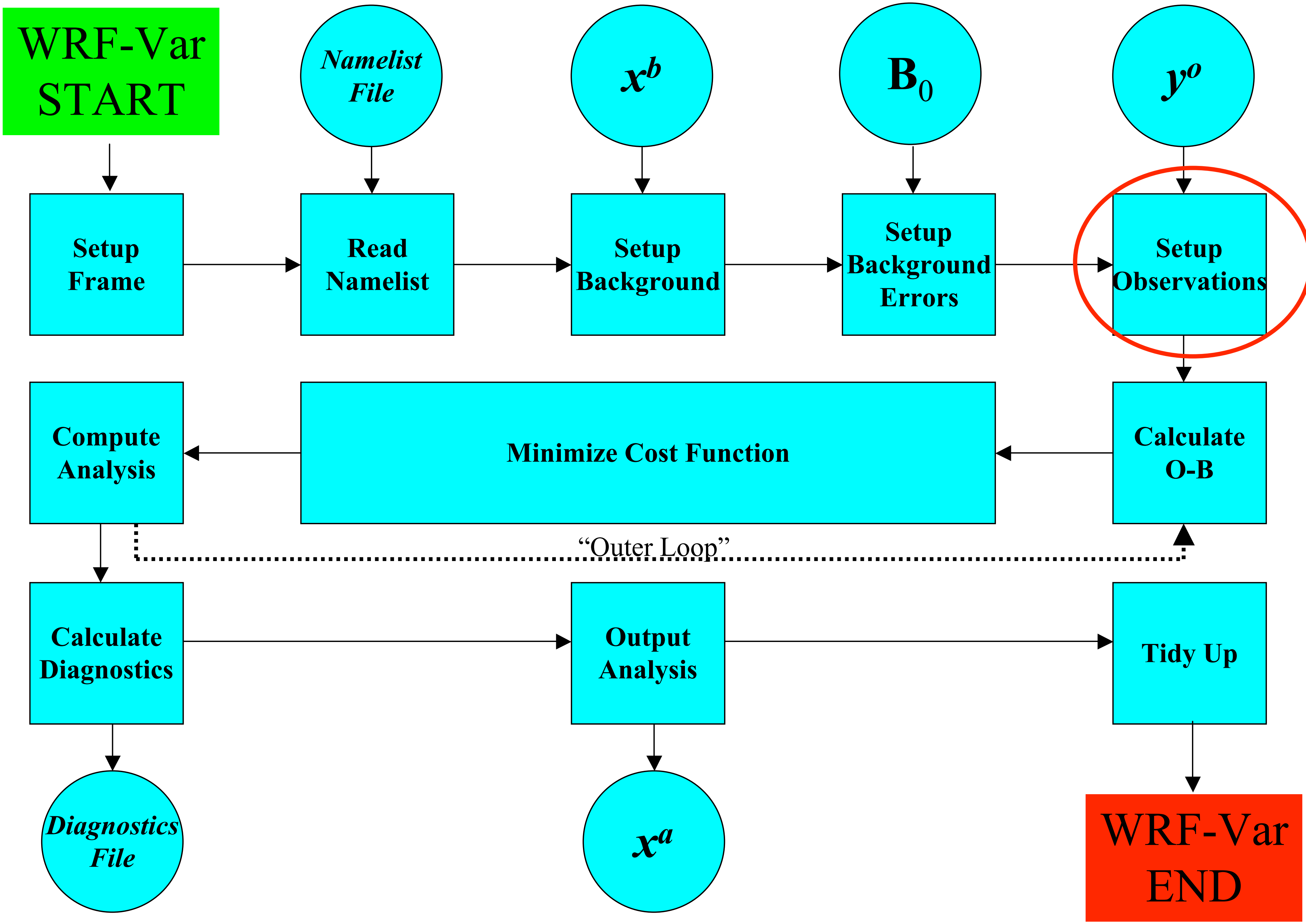


WRF-Var START
Namelist File
x^b
B_0
y^o
Setup Frame
Read Namelist
Setup Background
Setup Background Errors
Setup Observations
Compute Analysis
Minimize Cost Function
Calculate O-B
"Outer Loop"
Calculate Diagnostics
Output Analysis
Tidy Up
Diagnostics File
x^a
WRF-Var END

Setup Background Errors (BE)

- Reads in background error statistics.
- Extracts necessary quantities – eigenvectors, eigenvalues, lengthscales, regression coefficients, etc (see “WRF-Var Background Error Estimation”).
- Creates background error FORTRAN 90 derived data type “be” e.g. be % v1 % evec(:,:), be % v2 % eval(:), etc, ….

WRF-Var



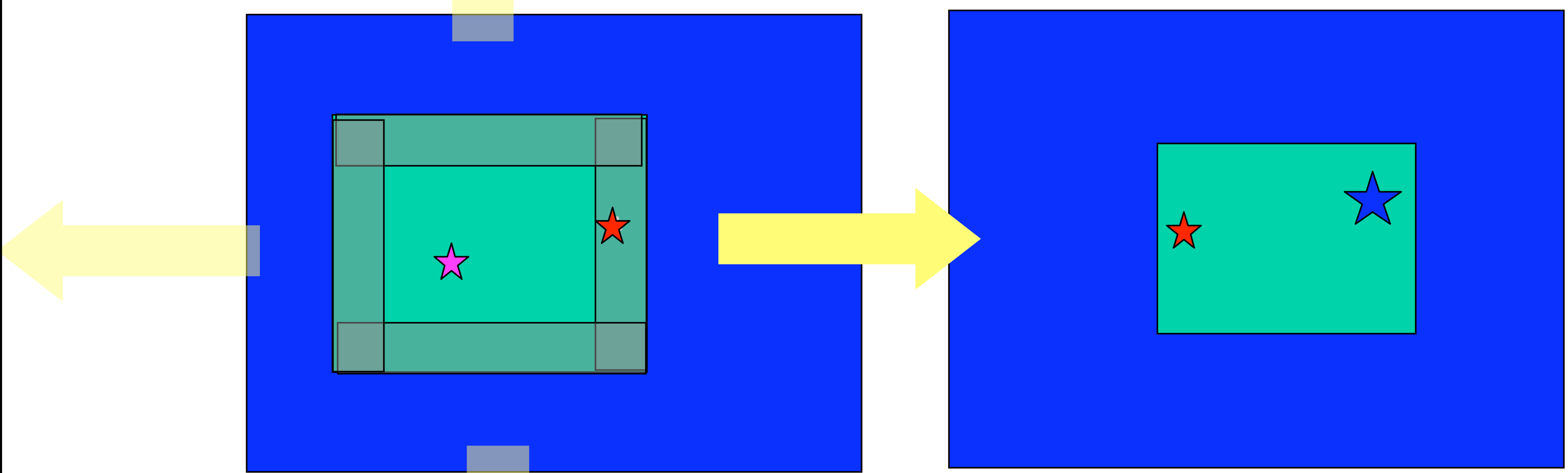
WRF-Var START
Namelist File
x^b
$\mathbf{B}_0$
y^o
Setup Frame
Read Namelist
Setup Background
Setup Background Errors
Setup Observations
Compute Analysis
Minimize Cost Function
Calculate O-B
"Outer Loop"
Calculate Diagnostics
Output Analysis
Tidy Up
Diagnostics File
x^a
WRF-Var END

Setup Observations

- Reads in observations.
- Format depends on namelist variable “ob_format”
  1 = BUFR, 2 = ASCII “WRF-Var” format.
- Creates observation FORTRAN 90 derived data type “ob” e.g. ob % metar(:), ob % sound(:) % u(:), etc, ….
- Identifies Obs outside/inside the domain

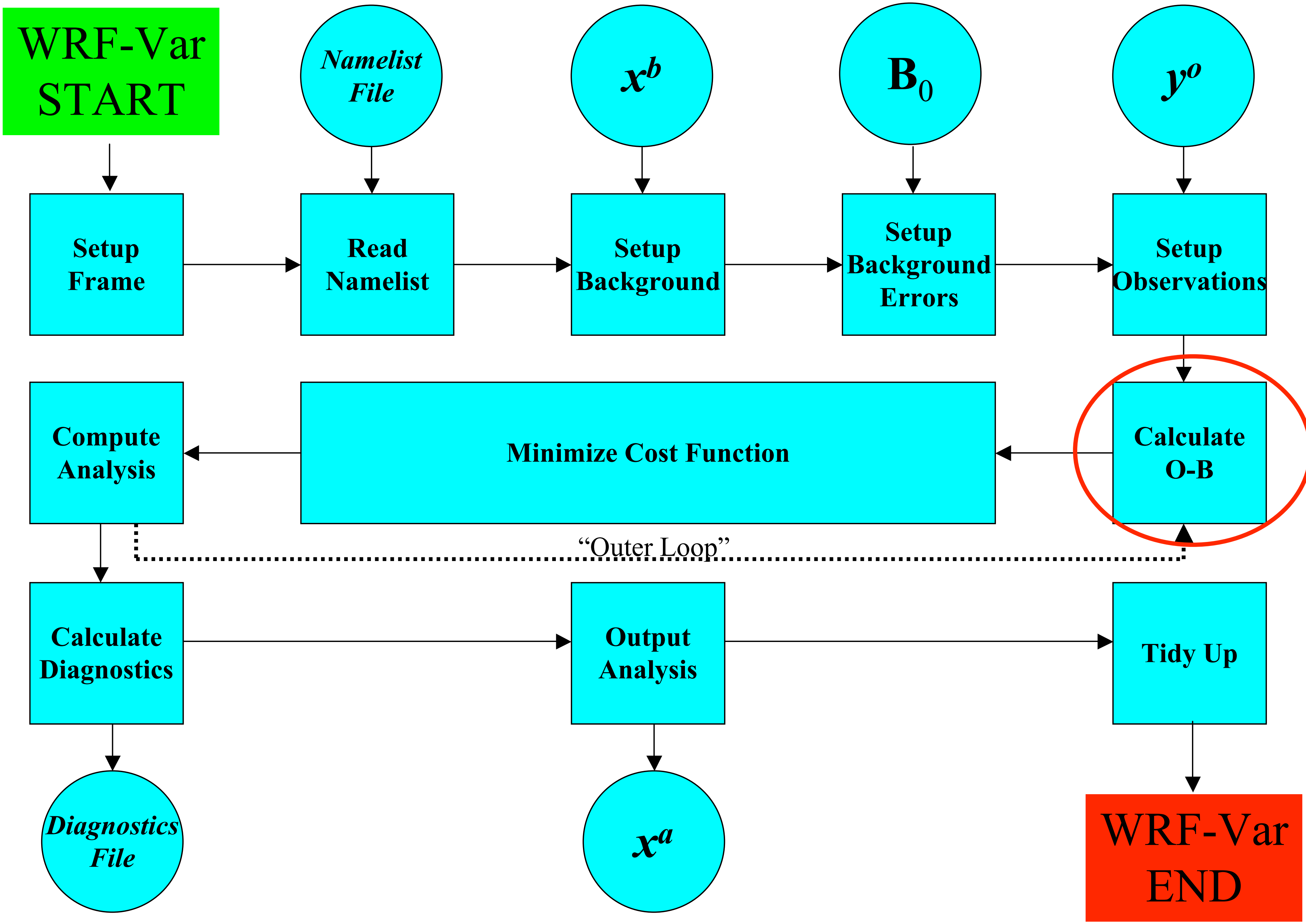
Observations in Distributed Memory

- **Halo Region Observation**
- **For global option obs. on East and West boundaries are duplicated**



Obs. on one processor's halo **Obs. on neighboring processor**

WRF-Var



WRF-Var START
Namelist File
x^b
$\mathbf{B}_0$
y^o
Setup Frame
Read Namelist
Setup Background
Setup Background Errors
Setup Observations
Compute Analysis
Minimize Cost Function
Calculate O-B
"Outer Loop"
Calculate Diagnostics
Output Analysis
Tidy Up
Diagnostics File
x^a
WRF-Var END

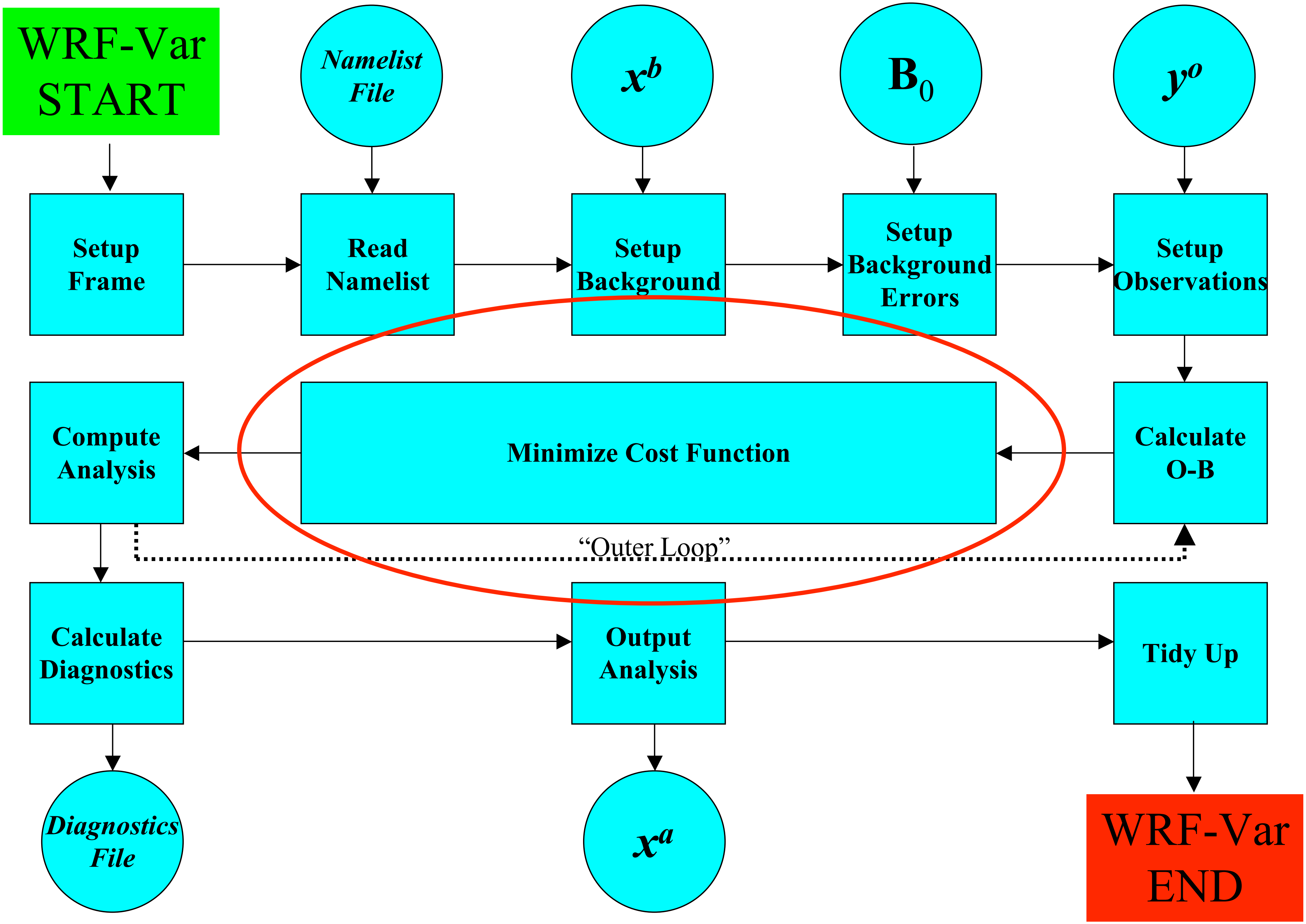
Calculate Innovation Vector (O-B)

- Calculates “model equivalent” B of observation O through interpolation and change of variable.

- Computes observation minus first guess (O-B) value.

- Creates innovation vector FORTRAN 90 derived data type “iv” e.g. iv % metar(:), iv % qscat(:) % u, iv % sound(:) % u(:), etc ….

WRF-Var



WRF-Var START
Namelist File
x^b
B_0
y^o
Setup Frame
Read Namelist
Setup Background
Setup Background Errors
Setup Observations
Compute Analysis
Minimize Cost Function
Calculate O-B
"Outer Loop"
Calculate Diagnostics
Output Analysis
Tidy Up
Diagnostics File
x^a
WRF-Var END

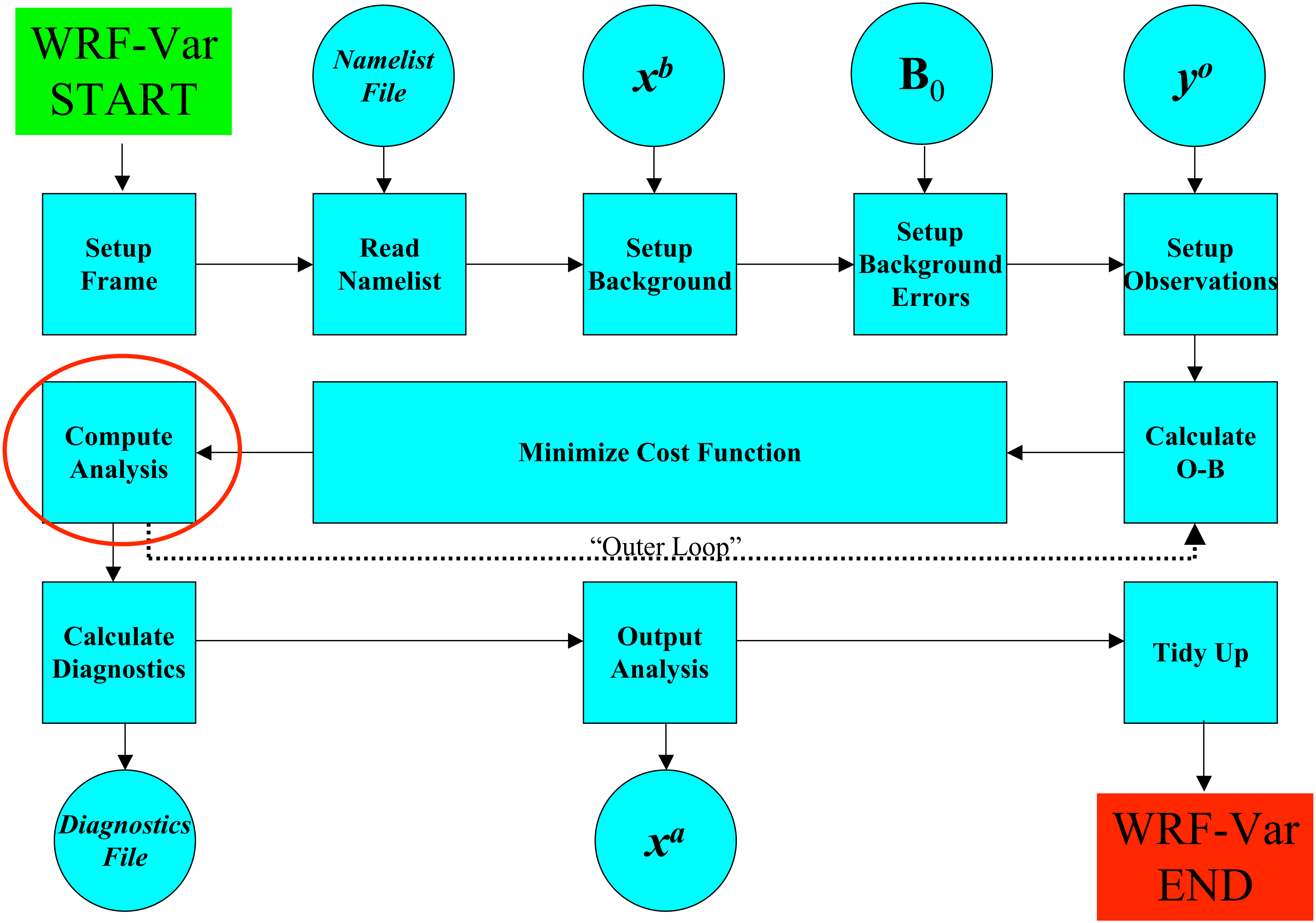
Minimize Cost Function

Use conjugate gradient method

- (a) Initializes analysis increments to zero.
- (b) Computes cost function (if desired).
- (c) Computes gradient of cost function.
- (d) Uses cost function and gradient to calculate new value of analysis control variable, **v**

Iterate (b) to (d)

WRF-Var


WRF-Var START
Namelist File
x^b
B_0
y^o
Setup Frame
Read Namelist
Setup Background
Setup Background Errors
Setup Observations
Compute Analysis
Minimize Cost Function
Calculate O-B
"Outer Loop"
Calculate Diagnostics
Output Analysis
Tidy Up
Diagnostics File
x^a
WRF-Var END

Compute Analysis

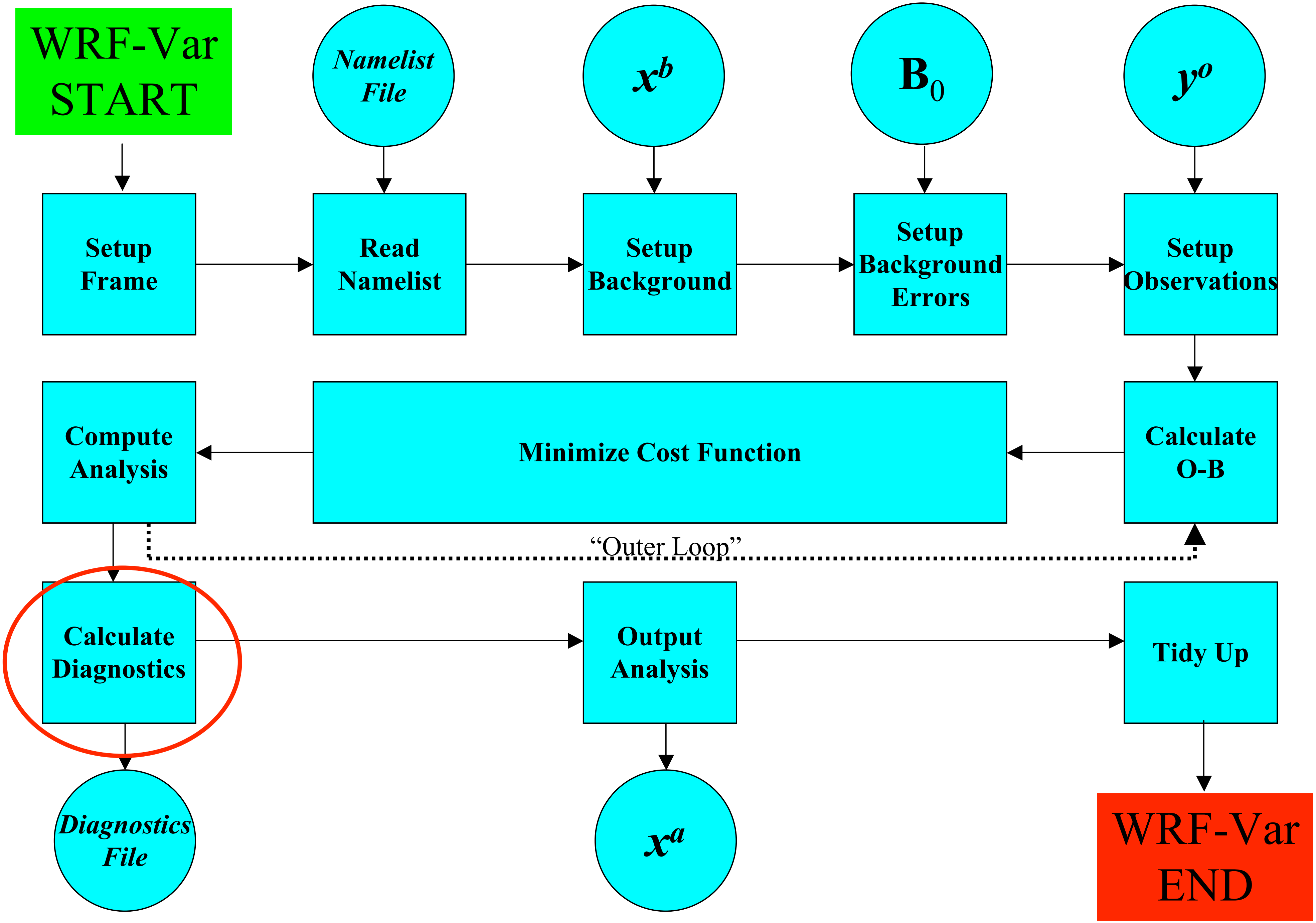
- Once WRF-Var has found a converged control variable, convert control variable to model space analysis increments

- Calculate:

  analysis = first-guess + analysis increment

- Performs consistency checks, e.g., remove negative humidity etc.

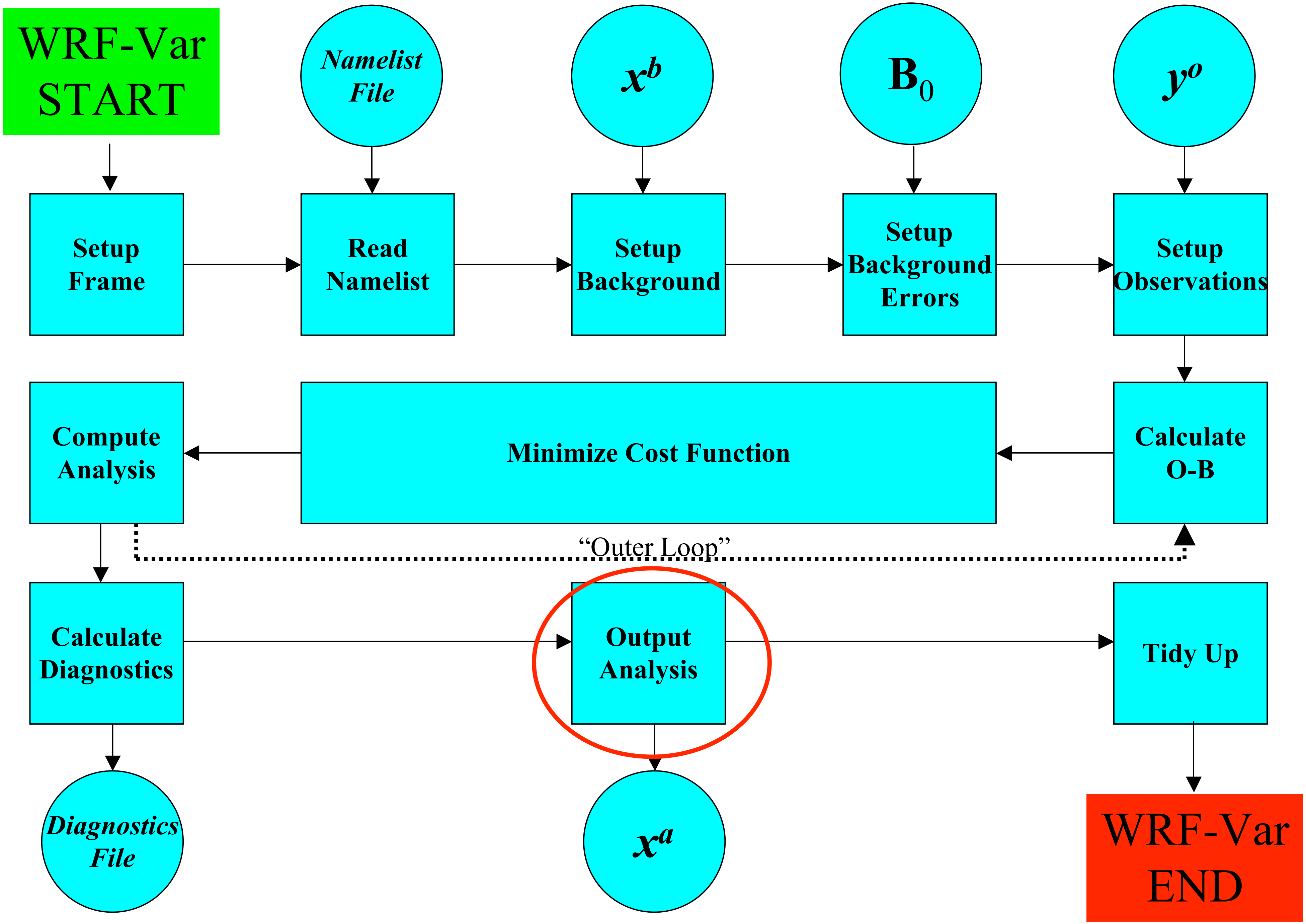
WRF-Var


WRF-Var START
Namelist File
x^b
B_0
y^o
Setup Frame
Read Namelist
Setup Background
Setup Background Errors
Setup Observations
Compute Analysis
Minimize Cost Function
Calculate O-B
"Outer Loop"
Calculate Diagnostics
Output Analysis
Tidy Up
Diagnostics File
x^a
WRF-Var END

Compute Diagnostics

- Compute O-B, O-A statistics for all observation types and variables.
- Compute A-B (analysis increment) statistics for all model variables and levels.
- Statistics include minimum, maximum (and their locations), mean and standard deviation.
- Compute “specialist diagnostics” for error tuning (fort.45, fort.46, fort.47, fort.50 etc.).

WRF-Var



WRF-Var START
Namelist File
x^b
B_0
y^o
Setup Frame
Read Namelist
Setup Background
Setup Background Errors
Setup Observations
Compute Analysis
Minimize Cost Function
Calculate O-B
"Outer Loop"
Calculate Diagnostics
Output Analysis
Tidy Up
Diagnostics File
x^a
WRF-Var END

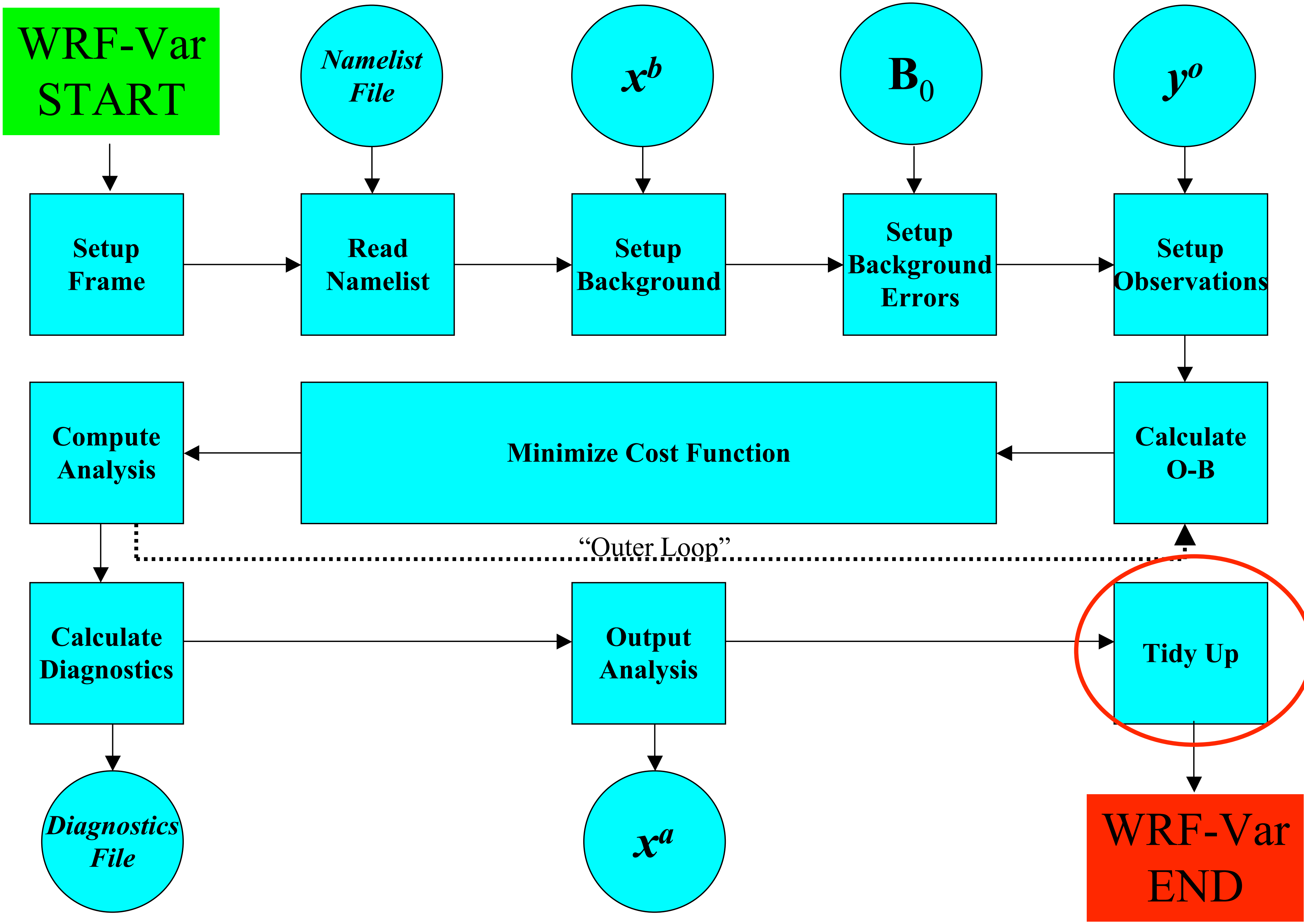
Output Analysis

- Outputs analysis in native model format. Choice is made through namelist option "fg_format"

  1 = WRF, etc.

- Also output analysis increments (for diagnostic purposes) in native model format. Switch off by setting WRITE_INCREMENTS = .FALSE. in namelist.input.

WRF-Var


WRF-Var START
Namelist File
x^b
B_0
y^o
Setup Frame
Read Namelist
Setup Background
Setup Background Errors
Setup Observations
Compute Analysis
Minimize Cost Function
Calculate O-B
"Outer Loop"
Calculate Diagnostics
Output Analysis
Tidy Up
Diagnostics File
x^a
WRF-Var END

Tidy Up

- Deallocate dynamically-allocated arrays, structures, etc.
- Timing information.
- Clean end to WRF-Var.

Procedure for Adding New Observations

- **Edit** da_define_structures.f90 **to add new data type.**
- **Make new observation sub-directory under** “var/da”**.**
- **Develop desired programs like getting innovation vector, forward observation operator, tangent linear and its adjoint, gradient & cost function etc. in this new sub-directory.**
- **Input observation (update** da_obs**).**
- **Sometimes it might be needed to add certain grid arrays in** Registry.wrfvar**.**
- **Link into minimization package (**da_minimisation.f90**)**

Learning to Use WRF-Var

- **Run through the Online WRF-Var Tutorial available at:**

  http://www.mmm.ucar.edu/wrf/users/docs/user_guide_V3/users_guide_chap6.htm

- **If still confused, ask questions via: wrfhelp@ucar.edu**