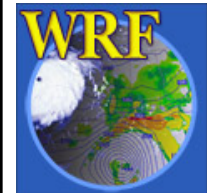


# WRF-Var System Overview

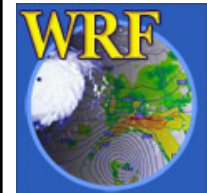
Xin Zhang, Yong-Run Guo,  
Syed R-H Rizvi, and Michael Duda



# Outline

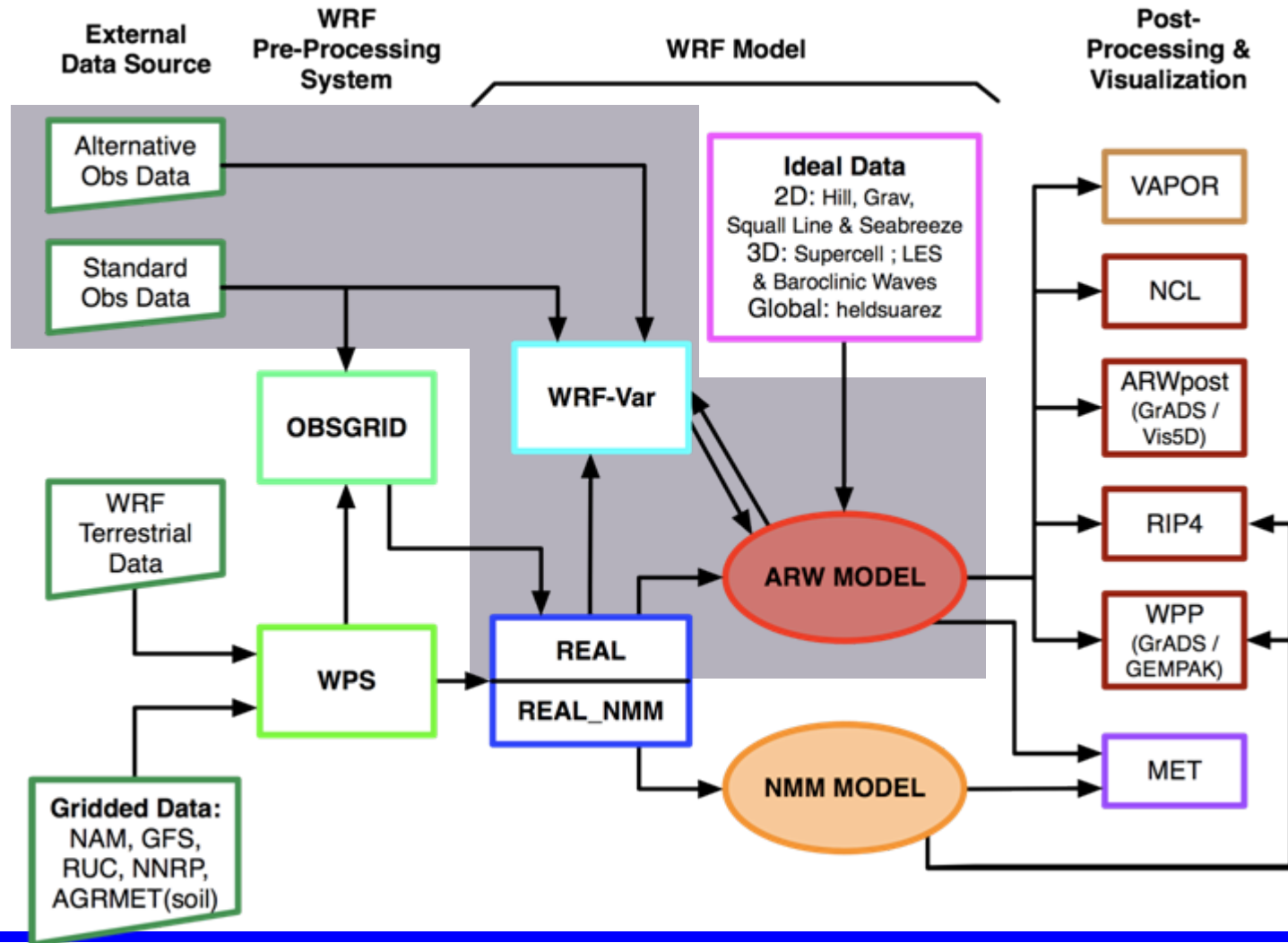
---

1. WRF-Var in the WRF Modeling System
2. WRF-Var Software
3. WRF-Var Implementation



# WRF-Var in the WRF Modeling System

## WRF Modeling System Flow Chart

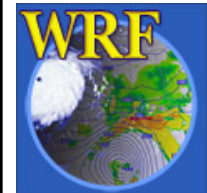


# WRF-Var in the WRF Modeling System

---

*WRF-Var*

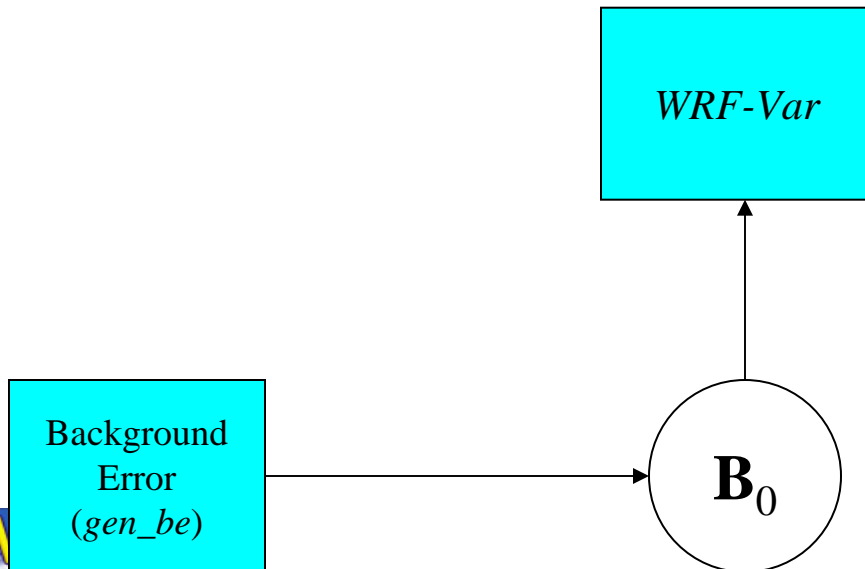
Blue --> Supported by WRF-Var Team



# WRF-Var in the WRF Modeling System

---

## 1. Prepare BE data



Blue --> Supported by WRF-Var Team

# Prepare BE statistics

---

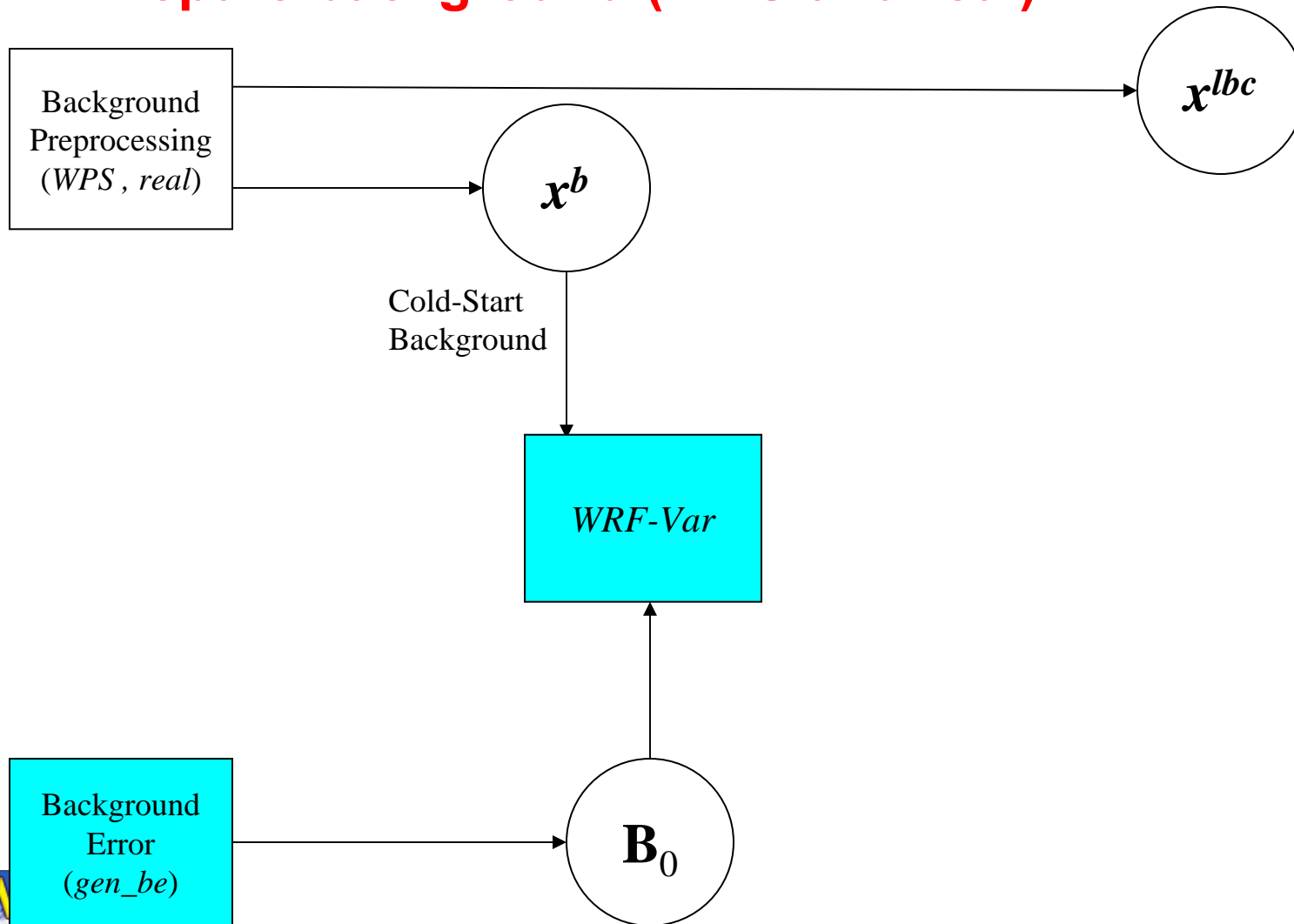
$$J(\mathbf{x}) = (\mathbf{x} - \mathbf{x}^b)^T \mathbf{B}^{-1} (\mathbf{x} - \mathbf{x}^b) + (\mathbf{y}^o - \mathbf{H}\mathbf{x})^T \mathbf{R}^{-1} (\mathbf{y}^o - \mathbf{H}\mathbf{x})$$

- For initial testing , default background error statistics may be used
  - be.dat file (CV option 5) from test case tar file can only be used with the domain from online tutorial
  - be.dat.cv3 (CV option 3) from source code tar file can be used for general test domains
- Ultimately, these should be specific to the particular model domain (and season)
- See lecture “WRF-Var Background Error Estimations”



# WRF-Var in the WRF Modeling System

## 2. Prepare background (WPS and real)



# Prepare background

---

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

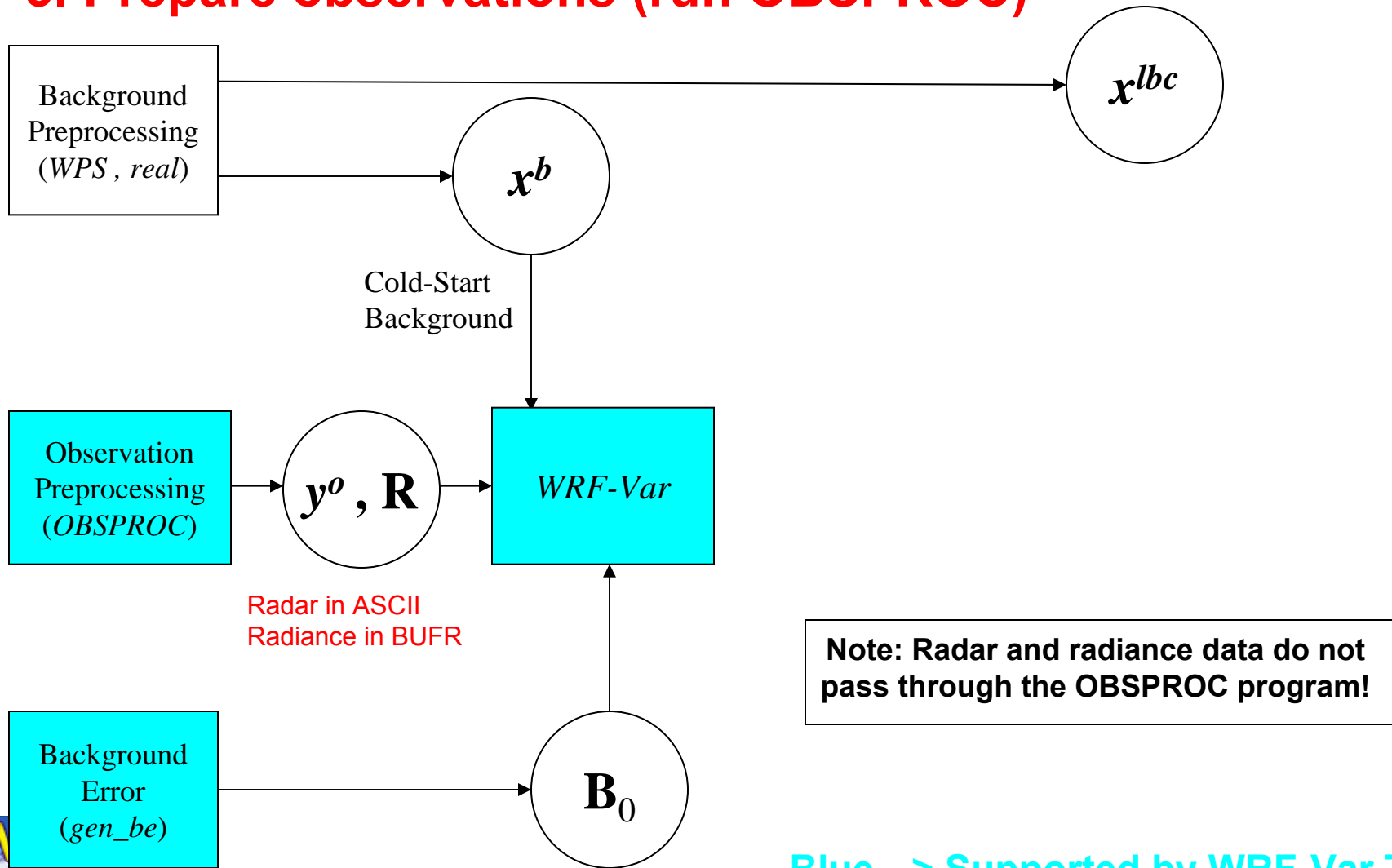
- In “cold-start” mode: accomplished by running the WPS and *real* programs
  - The background is essentially the wrfinput\_d01 file
- In “cycling” mode: the output of the WRF model
  - WRF can output wrfinput-formatted files used for cycling





# WRF-Var in the WRF Modeling System

## 3. Prepare observations (run OBSPROC)



Blue --> Supported by WRF-Var Team

# Prepare observations ( $y^0$ )

---

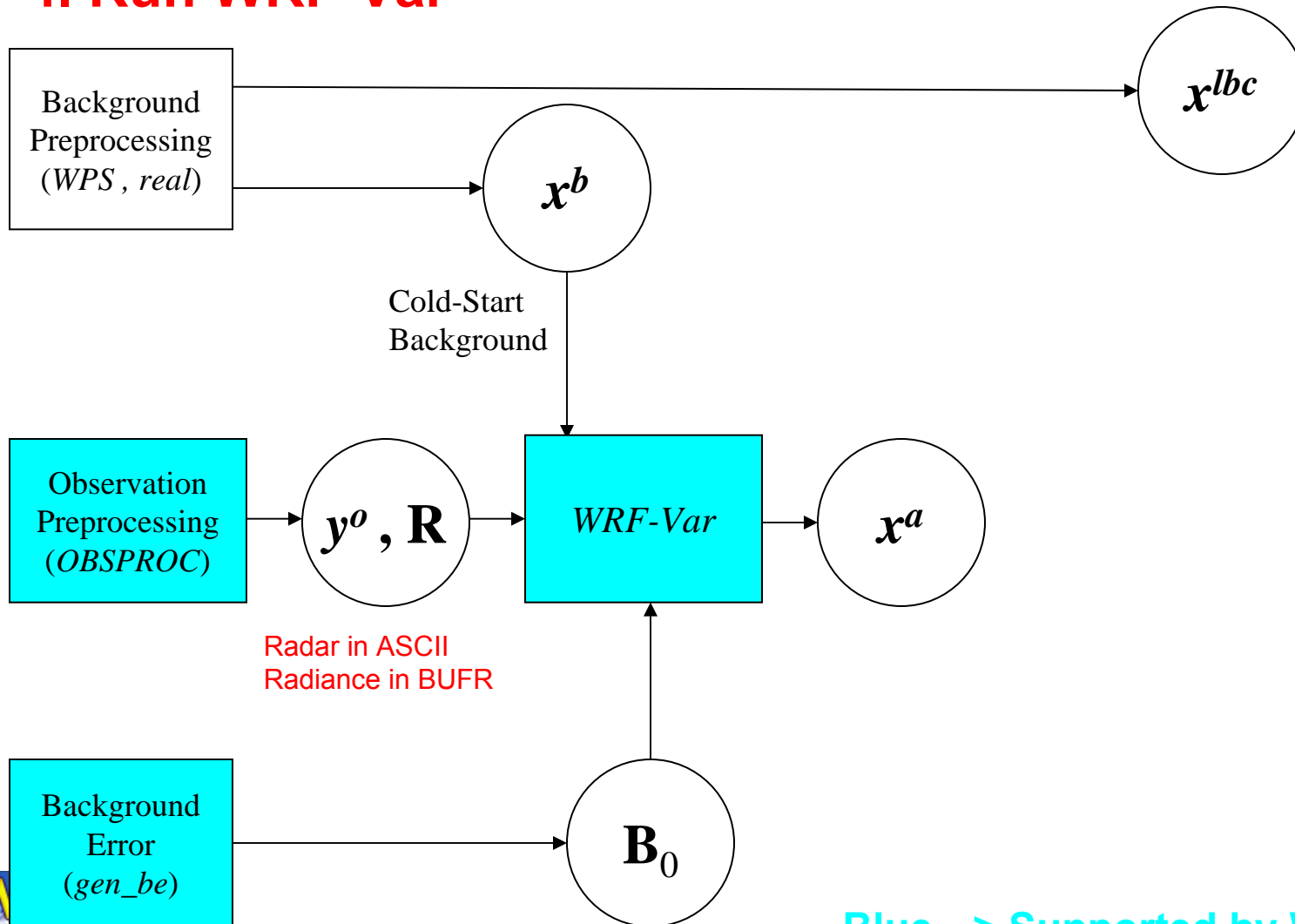
$$J(\mathbf{x}) = (\mathbf{x} - \mathbf{x}^b)^T \mathbf{B}^{-1} (\mathbf{x} - \mathbf{x}^b) + (\mathbf{y}^o - \mathbf{H}\mathbf{x})^T \mathbf{R}^{-1} (\mathbf{y}^o - \mathbf{H}\mathbf{x})$$

- Observation input for WRF-Var is supplied through observation preprocessor, OBSPROC
  - Except radar and satellite radiances
- Observation error covariance also provided by OBSPROC ( $\mathbf{R}$  is a diagonal matrix)
- Separate input file (ASCII) for radar, both reflectivity and radial velocity.
- Separate input file for satellite radiances



# WRF-Var in the WRF Modeling System

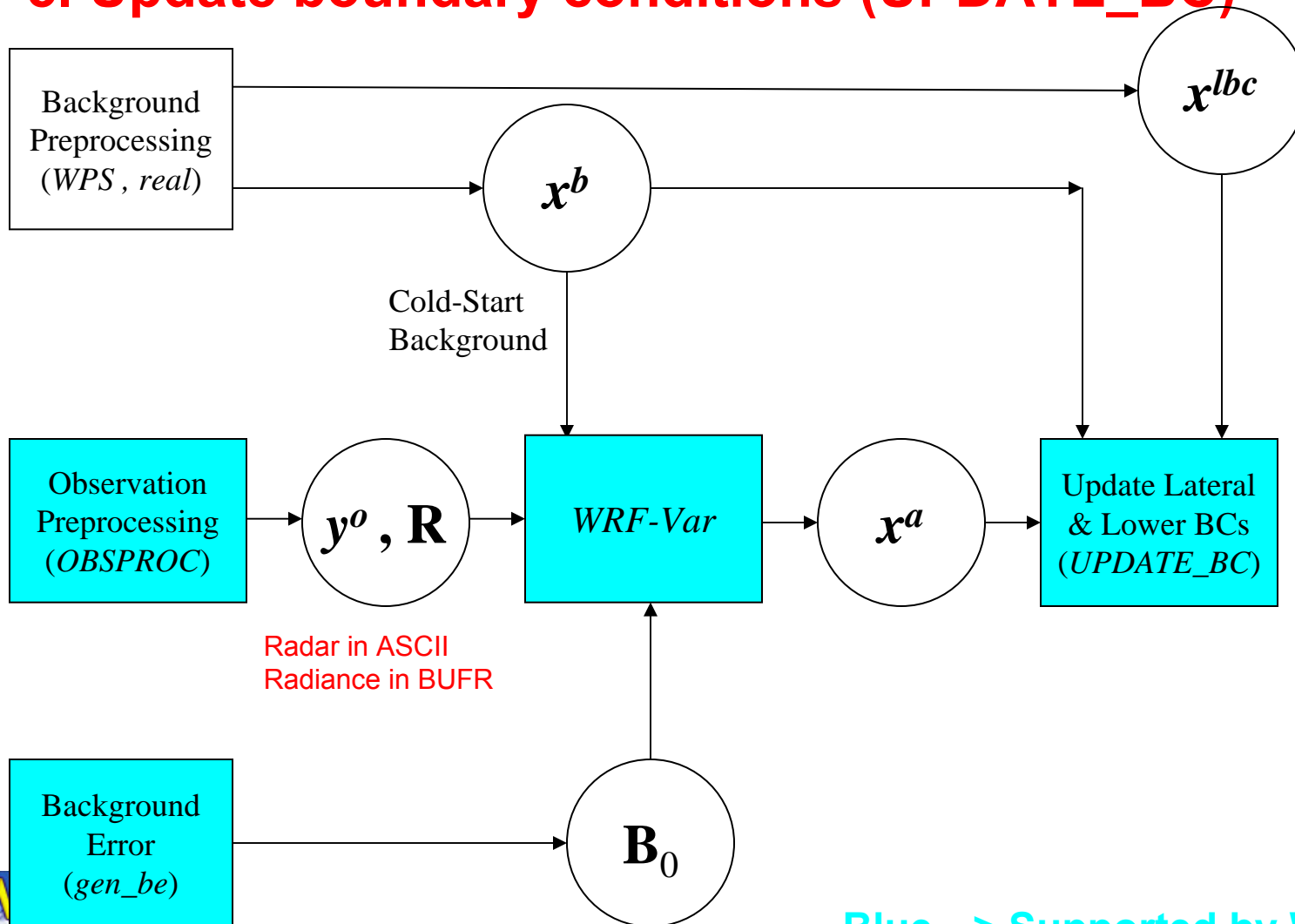
## 4. Run WRF-Var



Blue --> Supported by WRF-Var Team

# WRF-Var in the WRF Modeling System

## 5. Update boundary conditions (UPDATE\_BC)

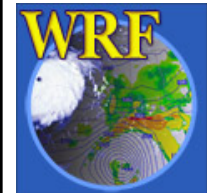


Blue --> Supported by WRF-Var Team

# Update boundary conditions

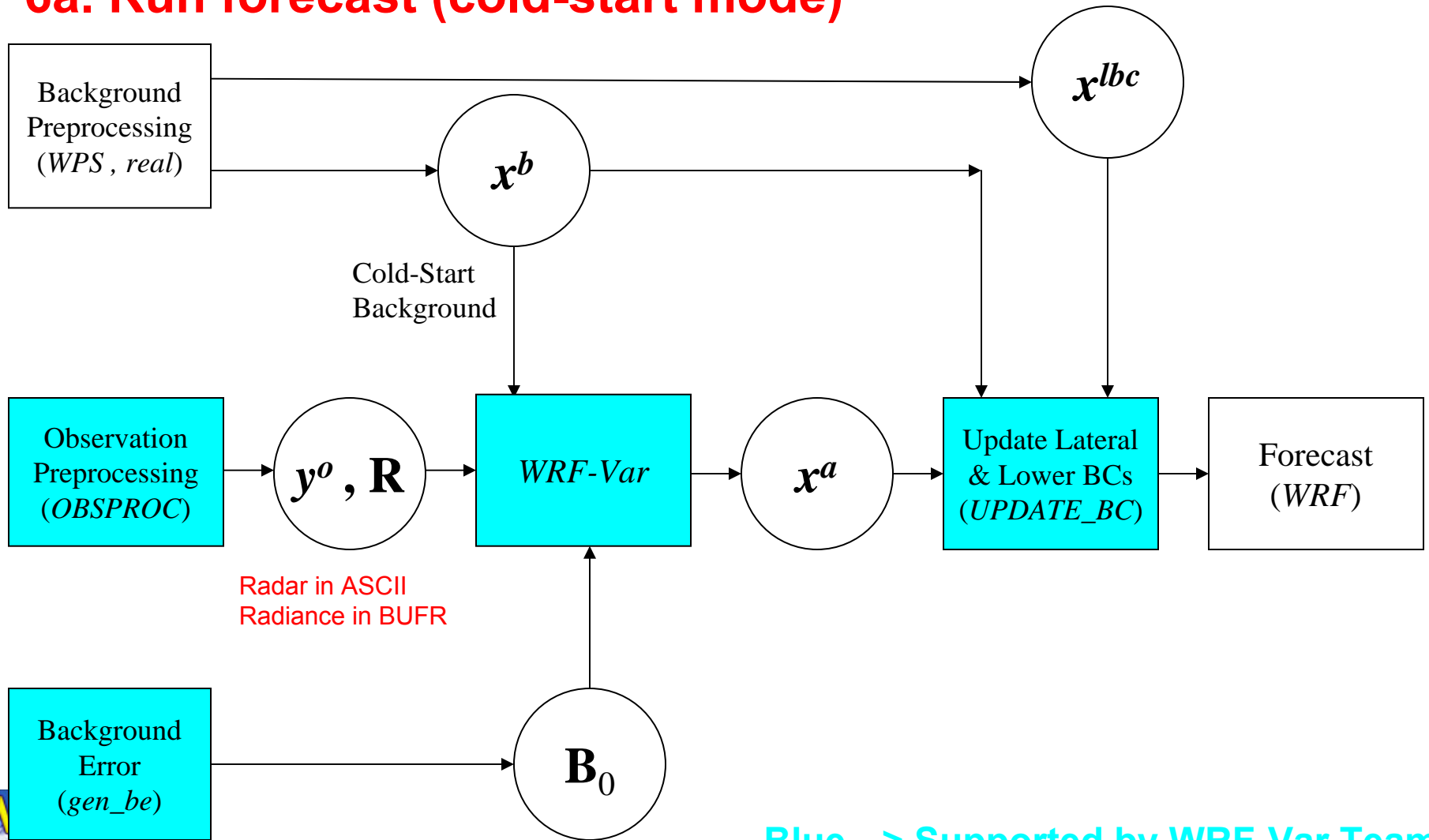
---

- 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

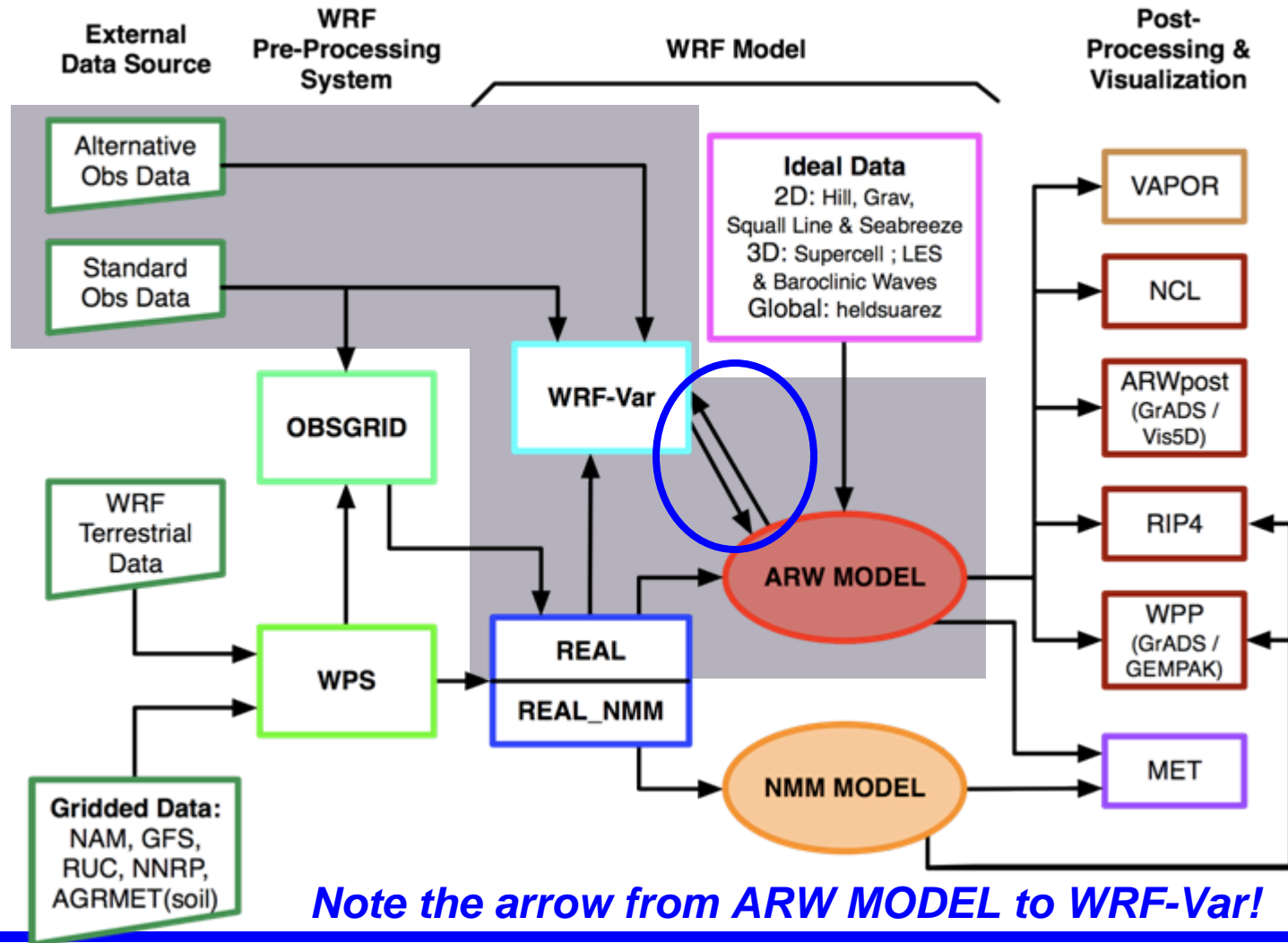
## 6a. Run forecast (cold-start mode)



Blue --> Supported by WRF-Var Team

# WRF-Var in the WRF Modeling System

## WRF Modeling System Flow Chart

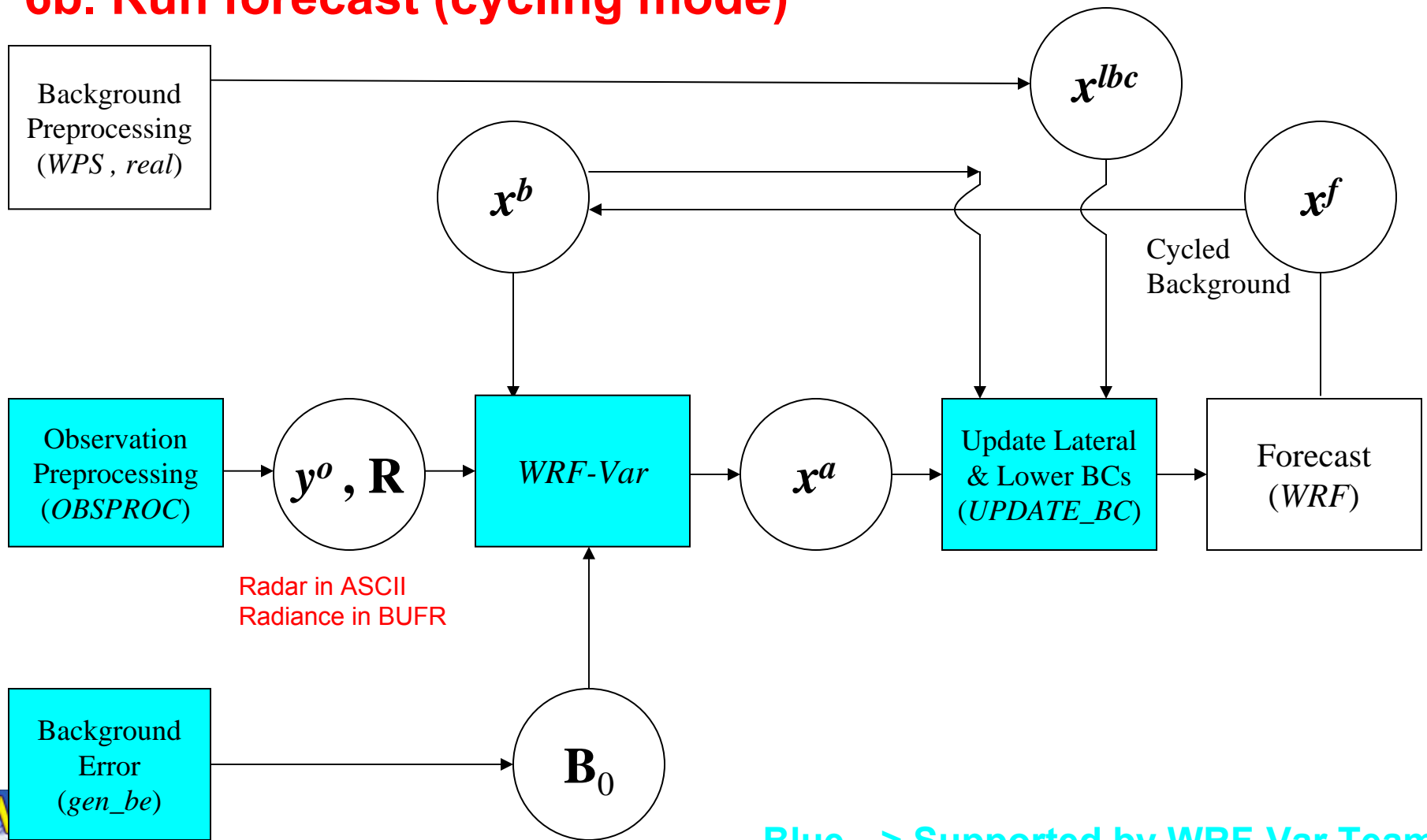


*Note the arrow from ARW MODEL to WRF-Var!*



# WRF-Var in the WRF Modeling System

## 6b. Run forecast (cycling mode)



Blue --> Supported by WRF-Var Team



# Background Error (BE) for WRF-Var

---

One important question from WRF-Var users is

*“What background errors are best for my application?”*

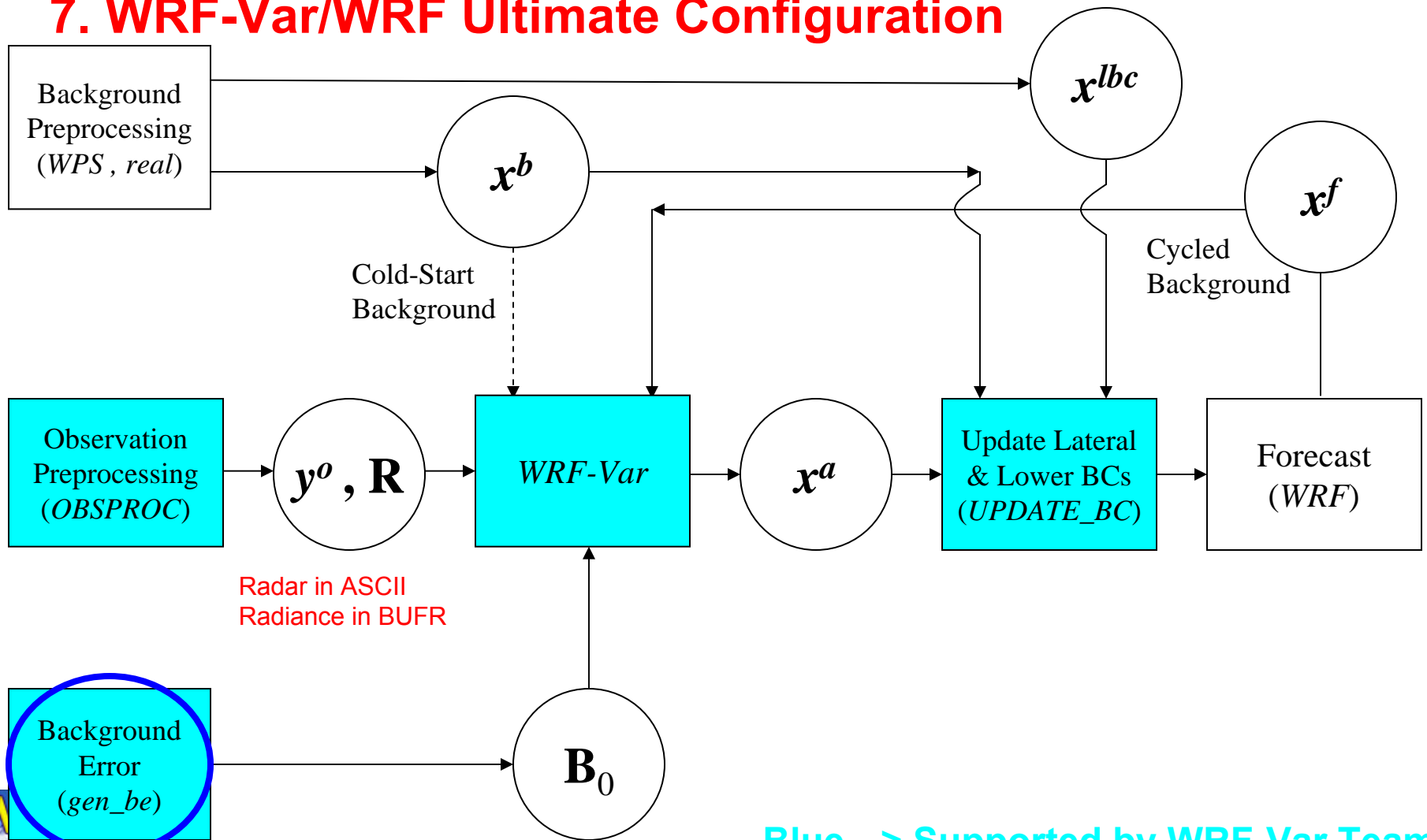
- Create your own once you have run your system for a few weeks to a month
- Implement, tune, and iterate

*The utility [gen\\_be](#) has been developed at NCAR for use in calculating these BEs*



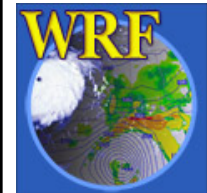
# WRF-Var in the WRF Modeling System

## 7. WRF-Var/WRF Ultimate Configuration



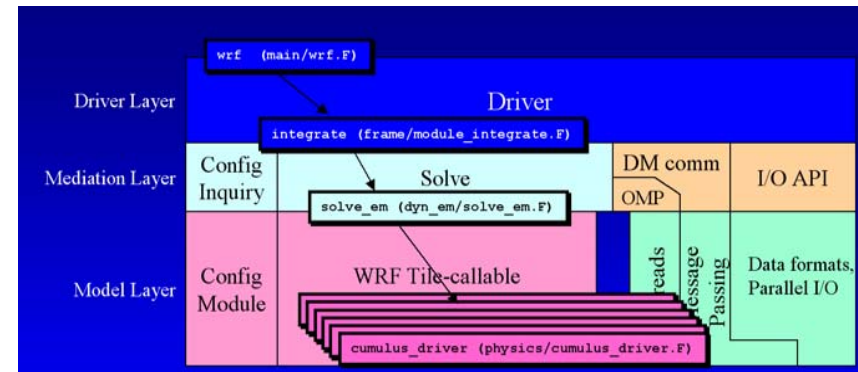
Blue --> Supported by WRF-Var Team

# WRF-Var Software

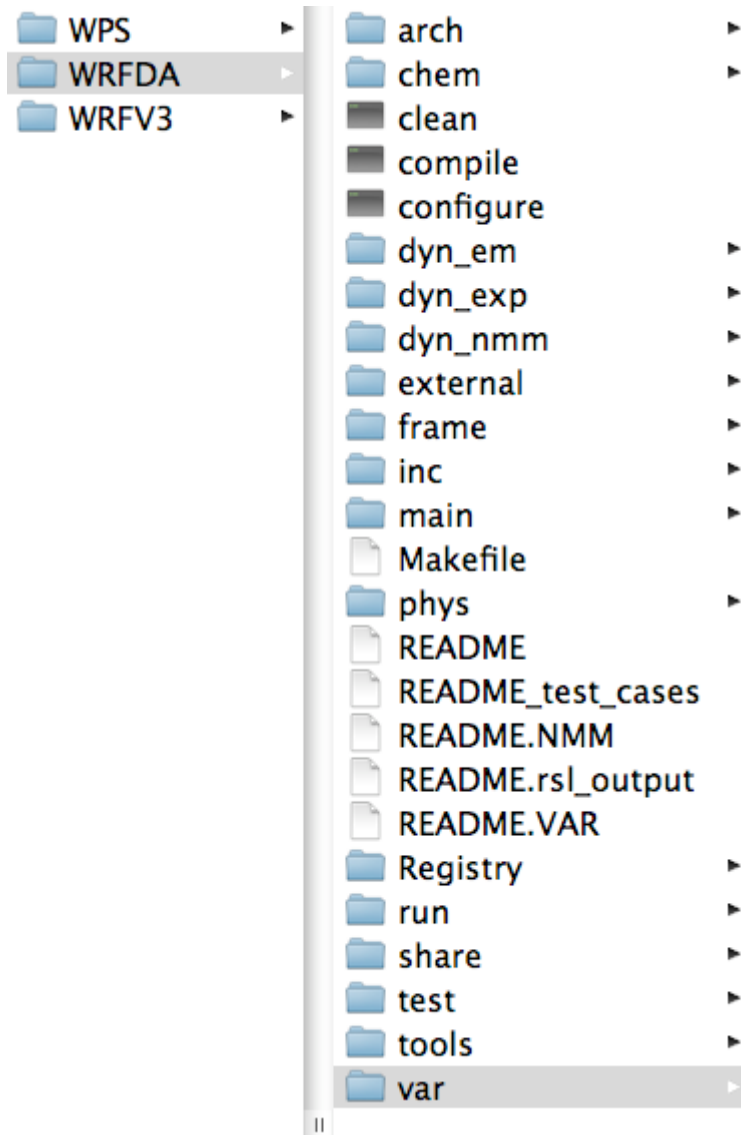


# Use of the WRF Software Framework

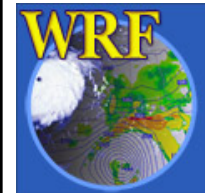
- WRF-Var relies on the WRF Software framework for
  - Distributed memory parallelism (halo exchanges, etc.)
  - Input/Output of first guess and analysis files
  - Parallel transposes
- WRF-Var also uses
  - The WRF Registry mechanism to handle definitions of fields, halos, and transposes
  - The WRF build system (clean, configure, compile)



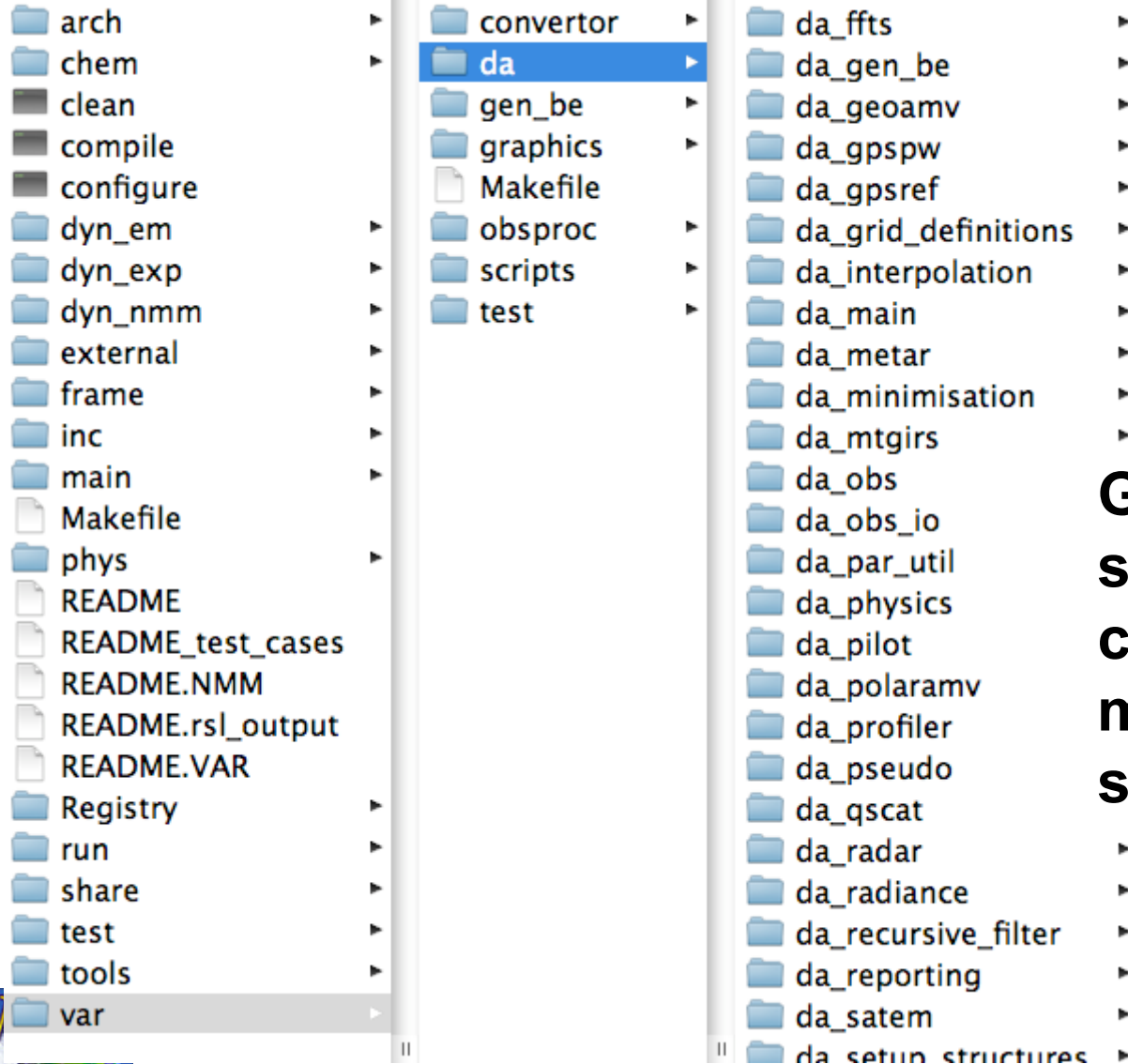
# WRF-Var Code Organization



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

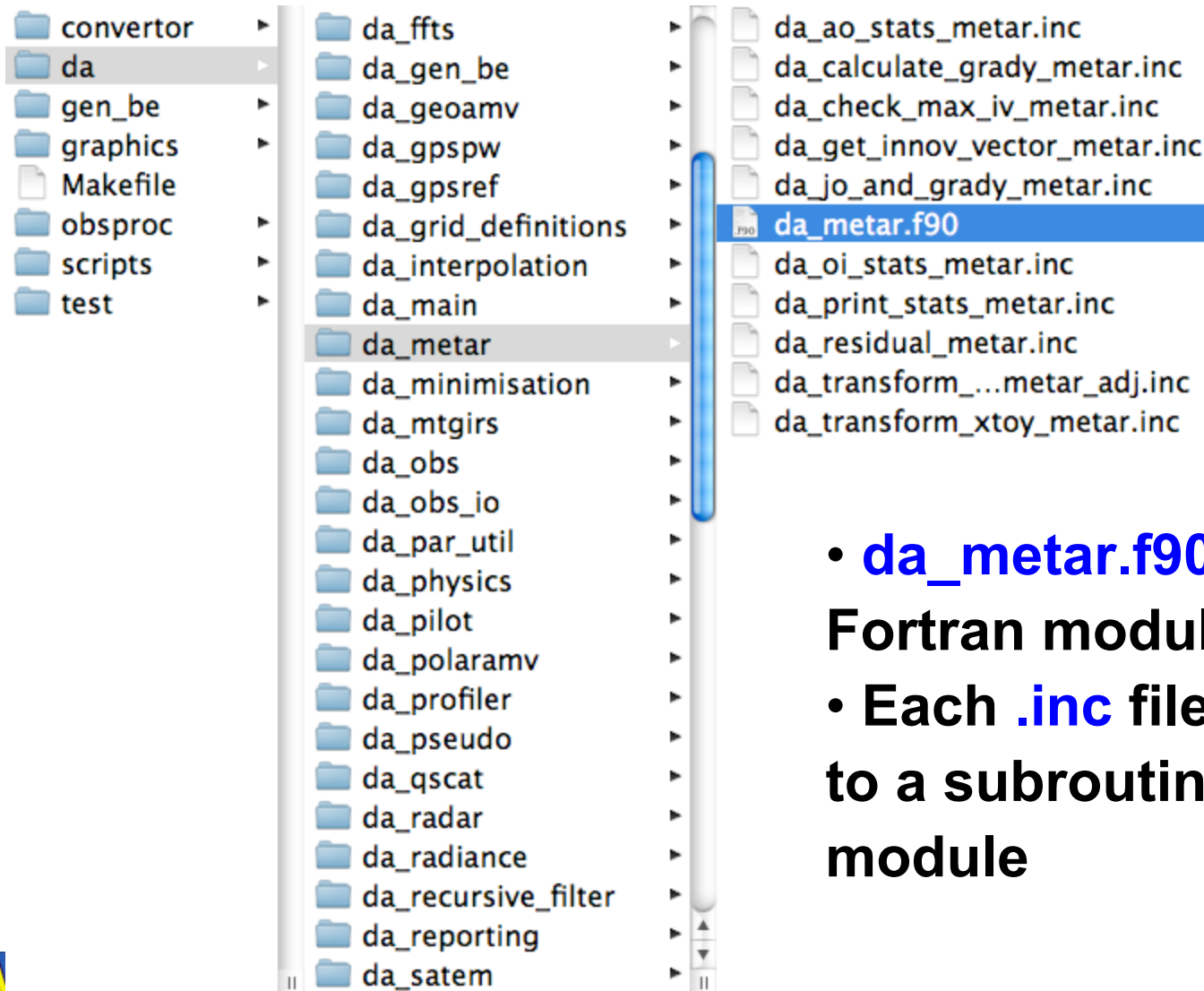


# WRF-Var Code Organization



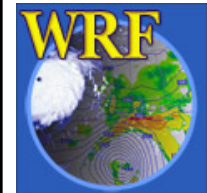
**Generally, each subdirectory of “da” contains a Fortran module with the same name** □

# WRF-Var Code Organization



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

# WRF-Var Implementation





# WRF-Var Formulation

---

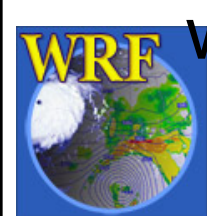
- WRF-Var actually uses an incremental formulation of the 3DVAR problem

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

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

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

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



# WRF-Var Formulation

---

- Next, define the *control variable transform*  $\mathbf{U}$  such that  $\mathbf{x}' = \mathbf{U}\mathbf{v}$ .

- $\mathbf{v}$  is the analysis increment in control variable space
- $\mathbf{B}$  is approximated by  $\mathbf{U}\mathbf{U}^T$

- WRF-Var actually minimizes

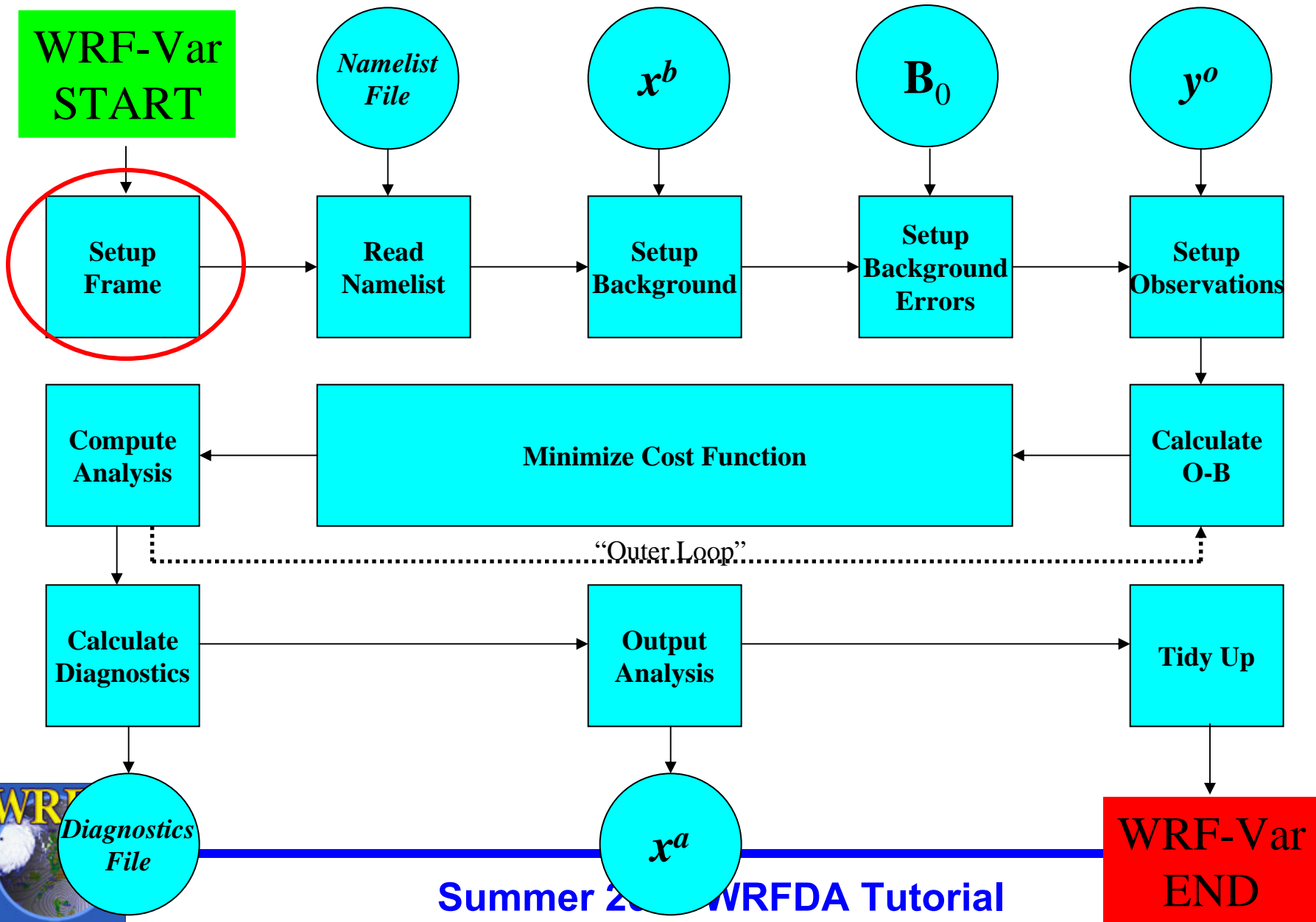
$$J(\mathbf{v}) = \mathbf{v}^T\mathbf{v} + (\mathbf{y}^{o'} - \mathbf{H}\mathbf{U}\mathbf{v})^T\mathbf{R}^{-1}(\mathbf{y}^{o'} - \mathbf{H}\mathbf{U}\mathbf{v})$$

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

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



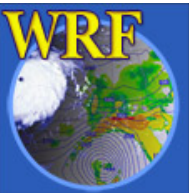
# WRF-Var



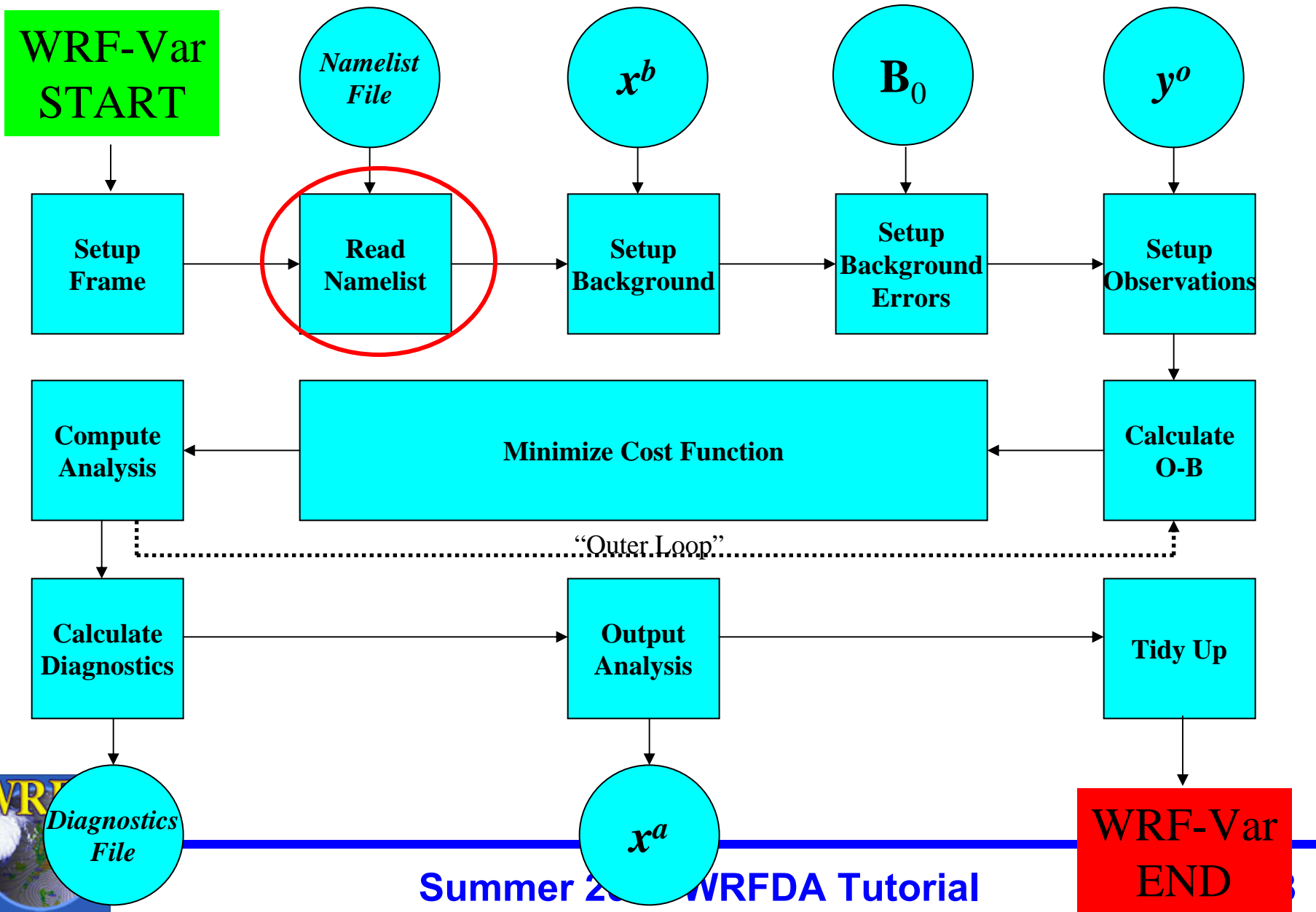
# Setup Frame

---

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



# WRF-Var



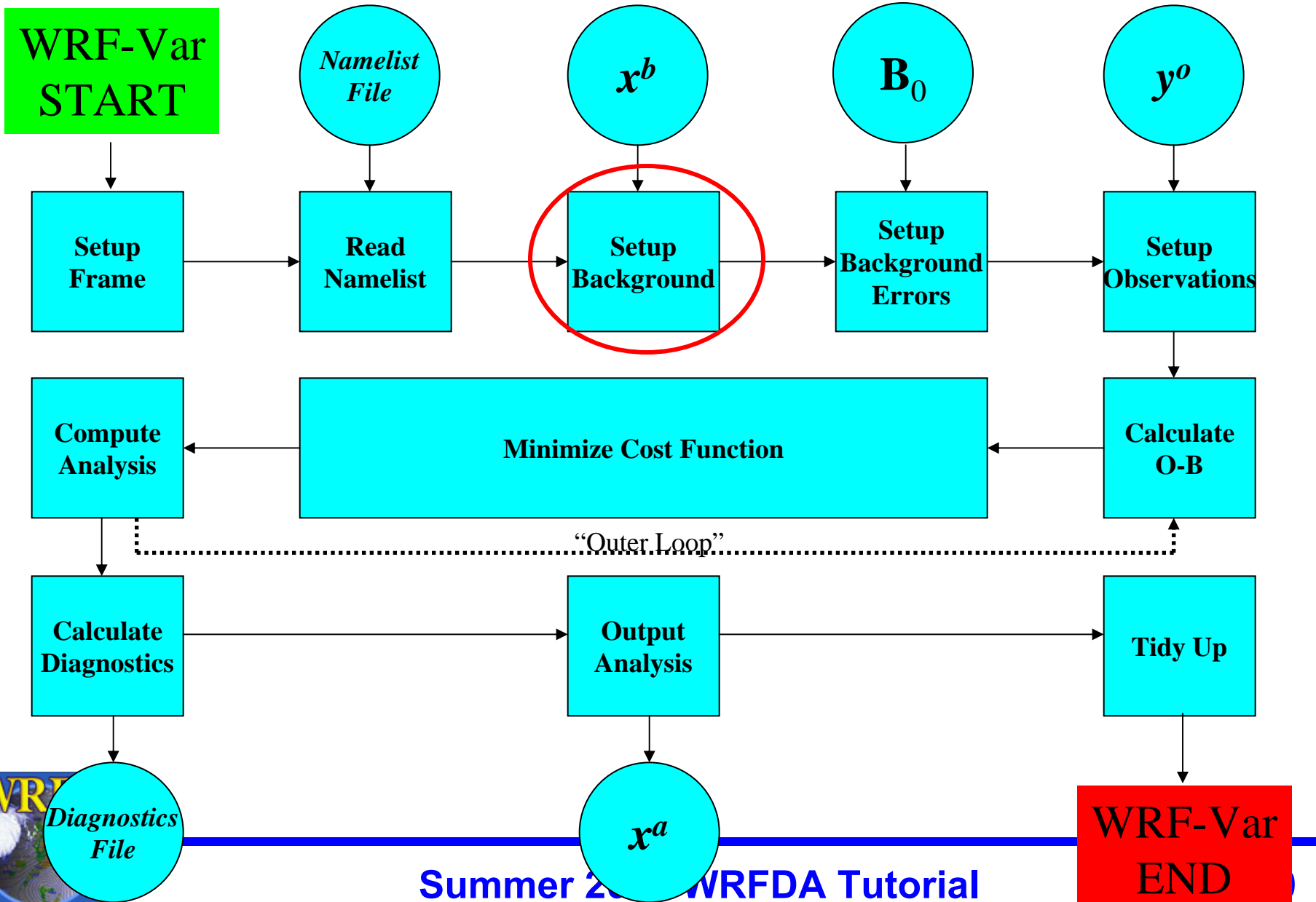
# Read Namelist

---

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



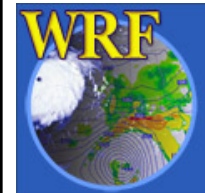
# WRF-Var



# Setup Background (First-guess)

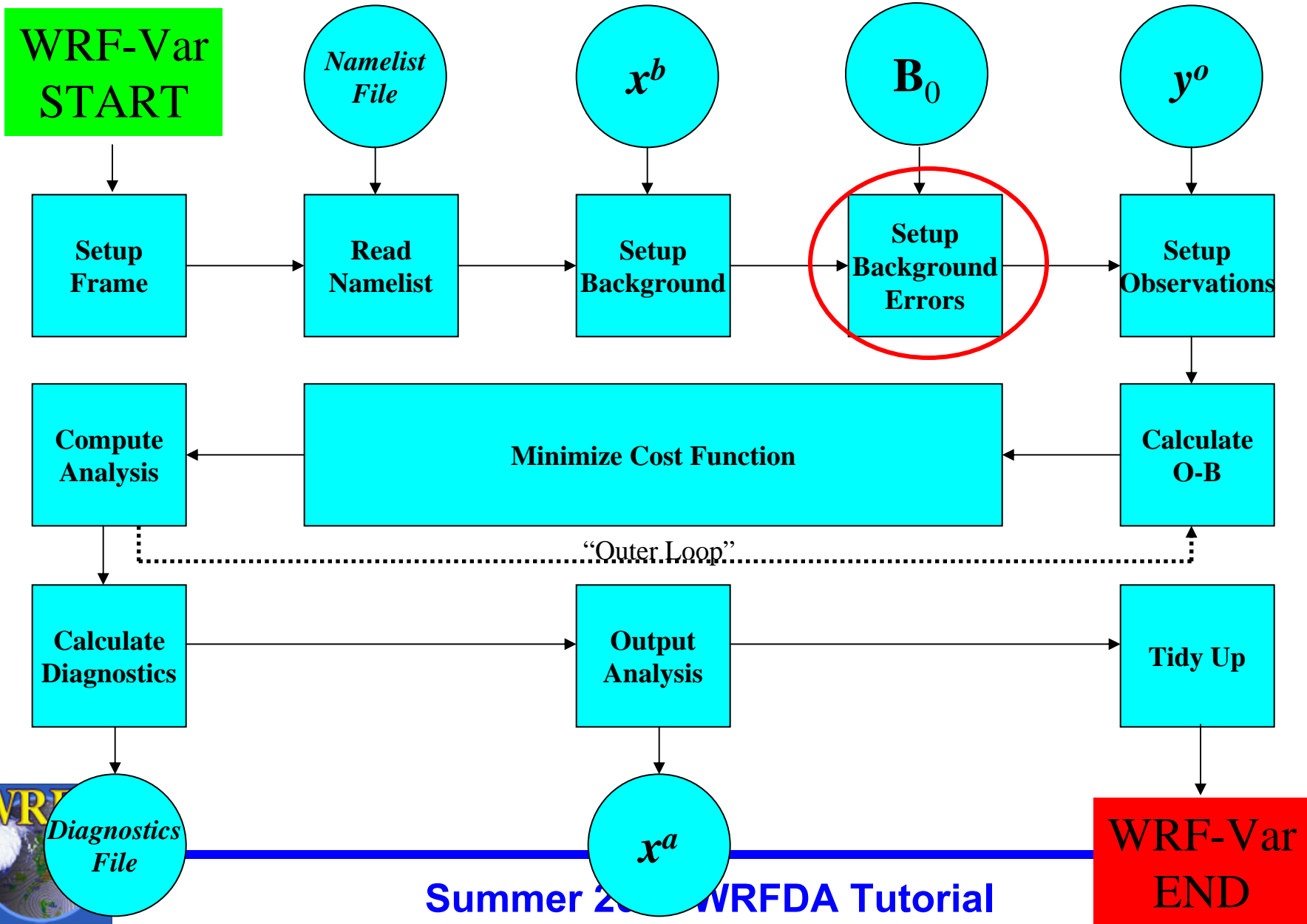
---

- 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



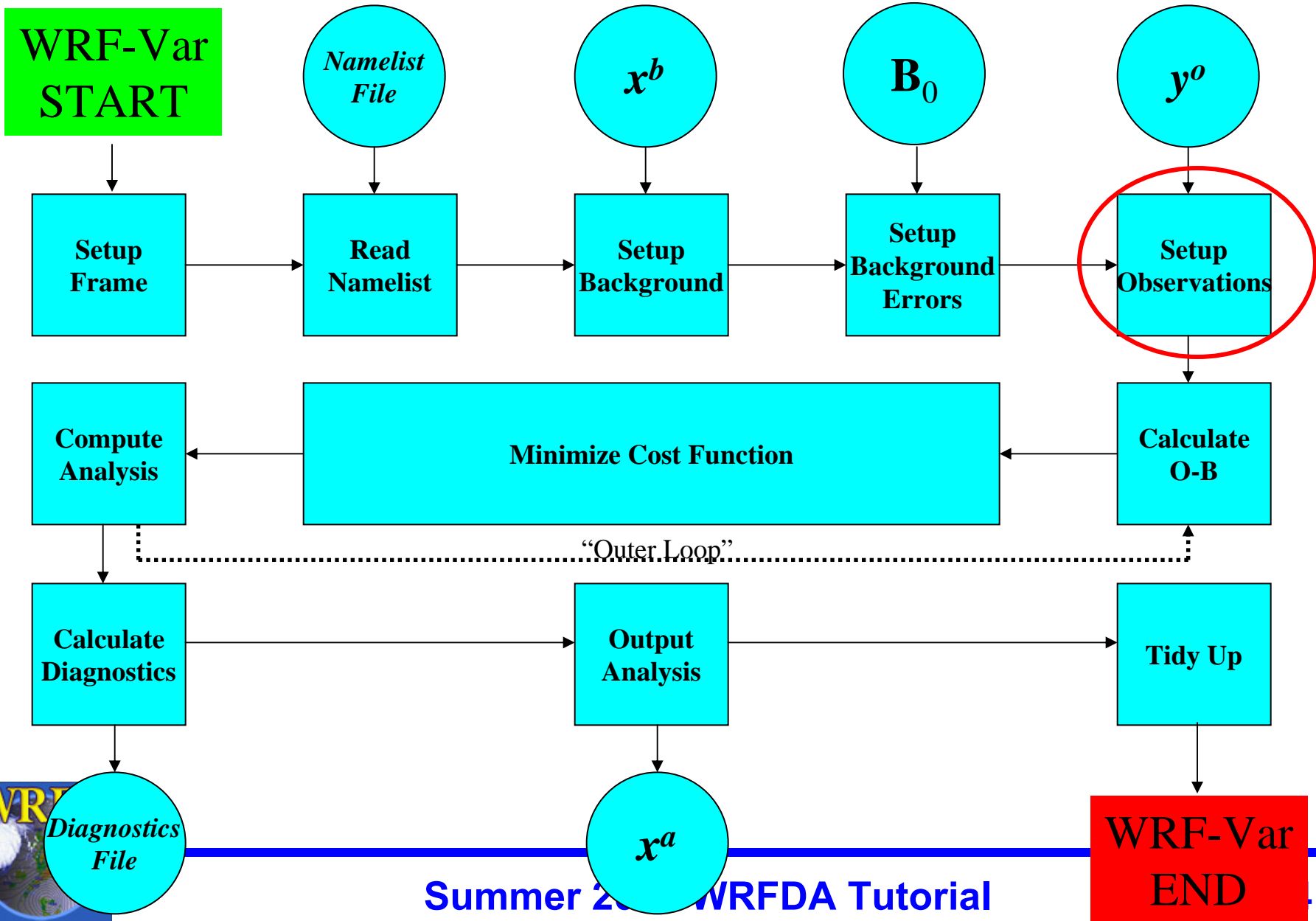
# Setup Background Errors (BE)

---

- 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



# Setup Observations

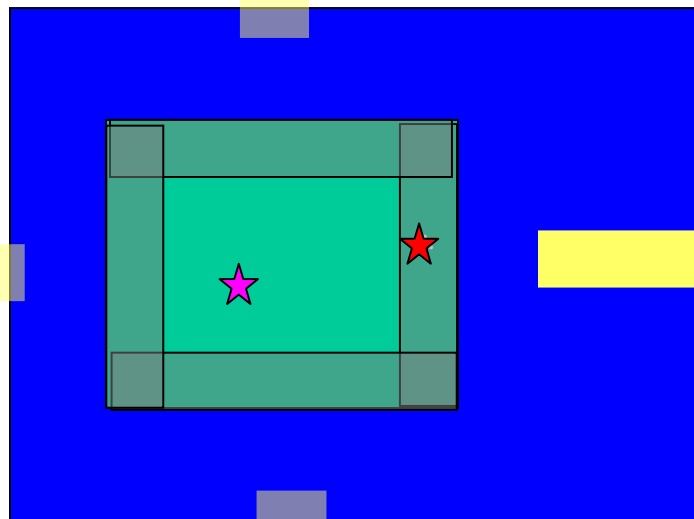
---

- 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

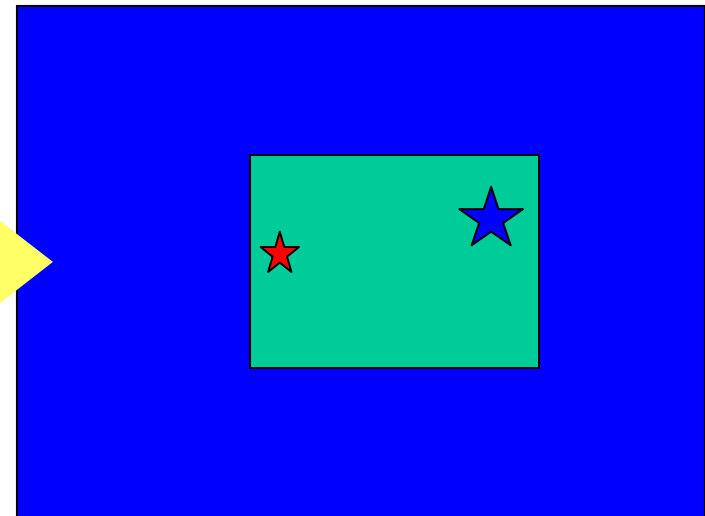


# Observations in Distributed Memory

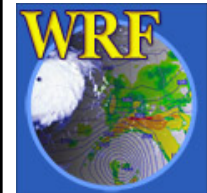
- Halo Region Observation
- For global option obs. on East and West boundaries are duplicated



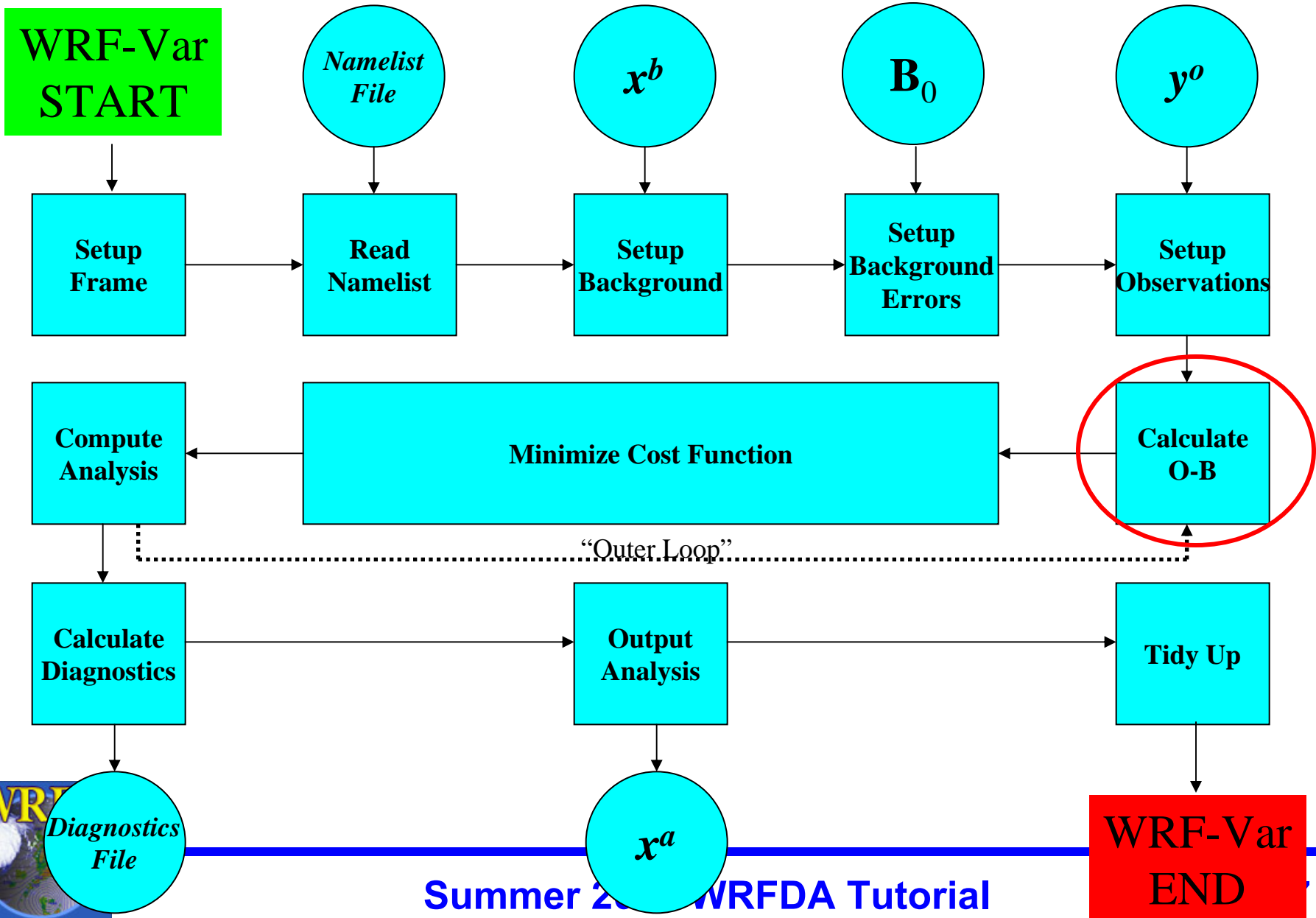
Obs. on one processor's halo



Obs. on neighboring processor



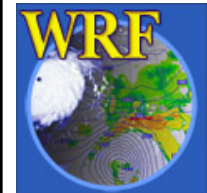
# WRF-Var



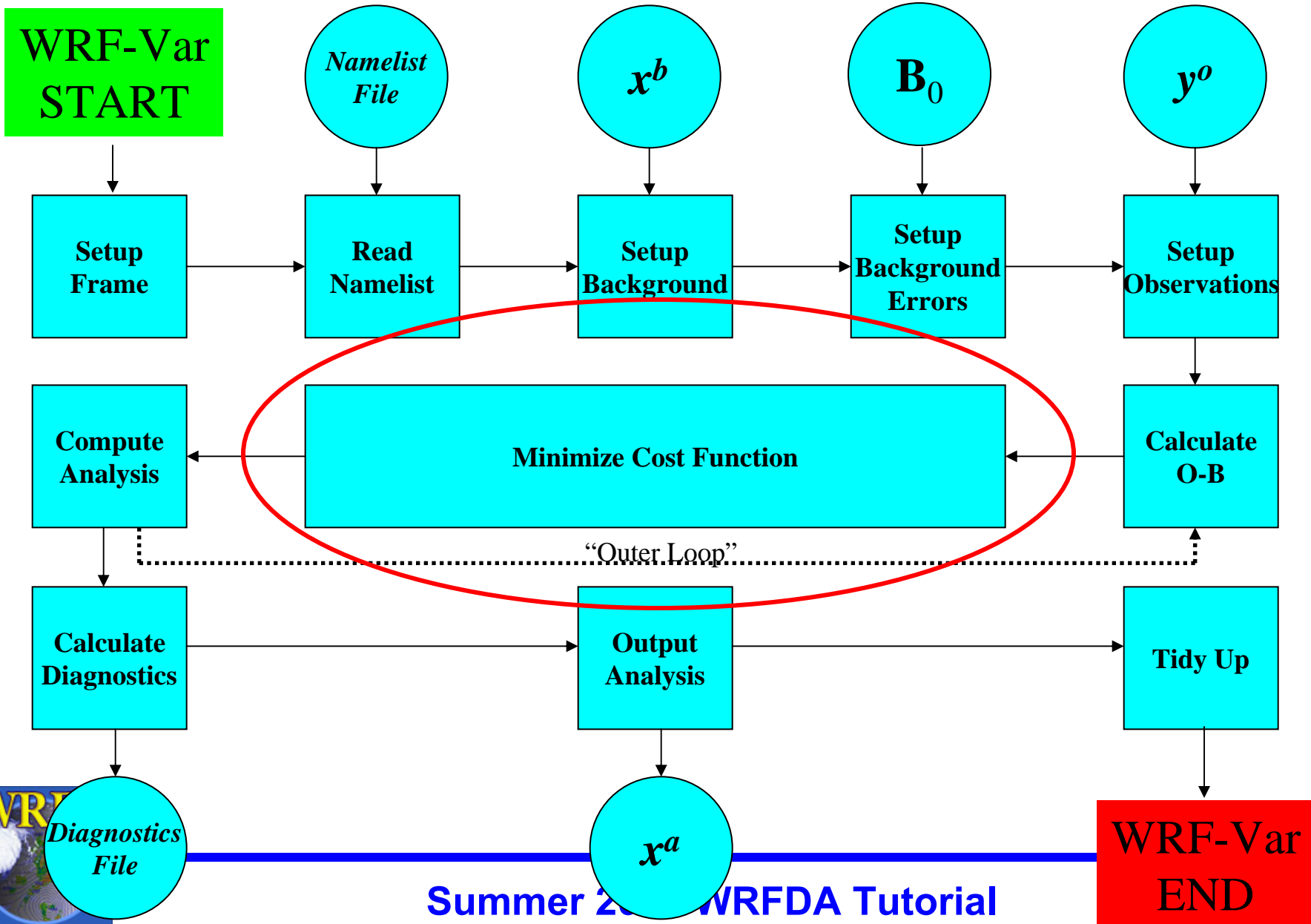
# Calculate Innovation Vector (O-B)

---

- 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





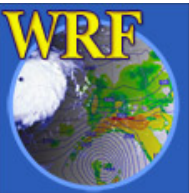
# Minimize Cost Function

---

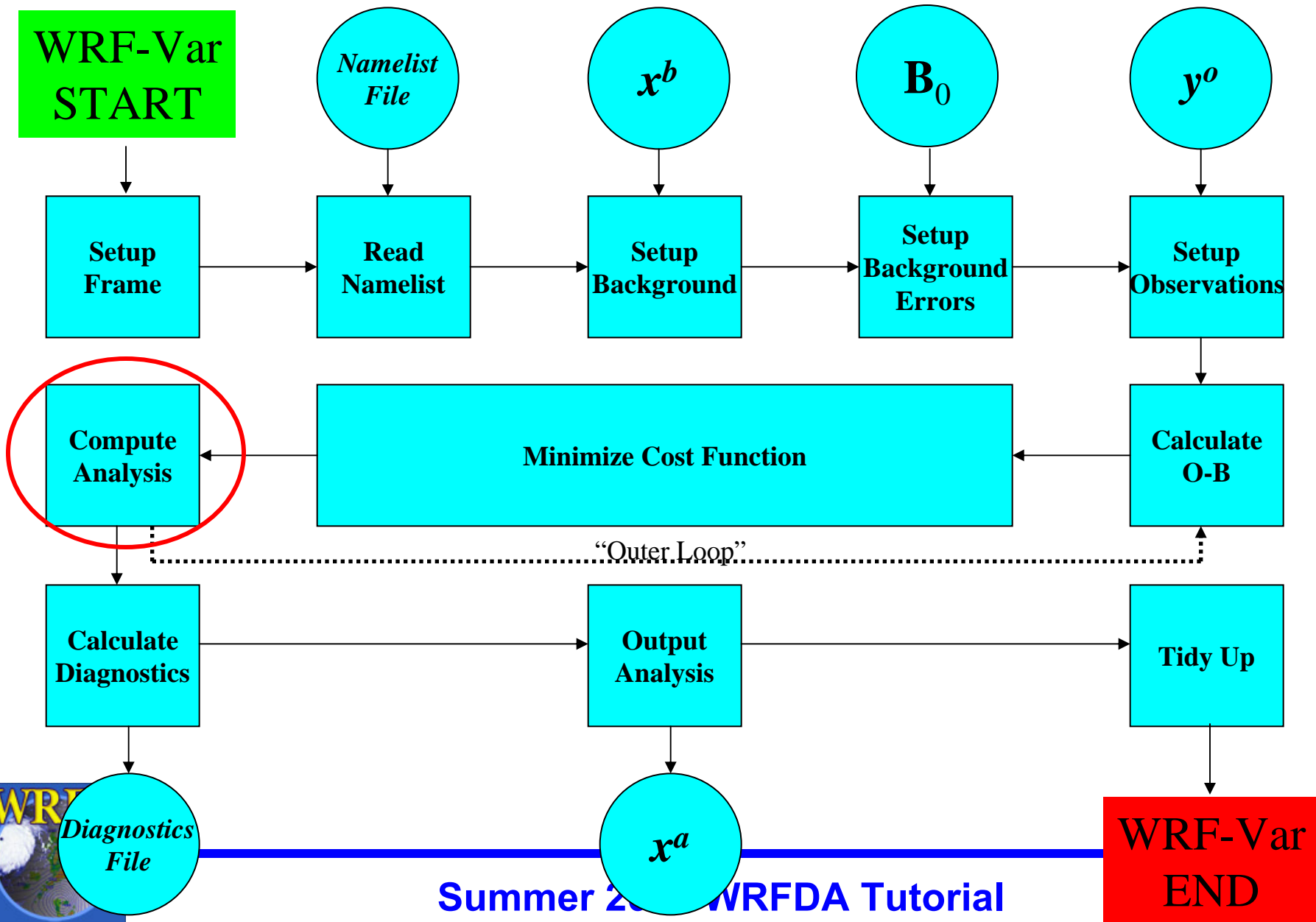
Use conjugate gradient method

- (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,  $\mathbf{v}$

Iterate (b) to (d)



# WRF-Var



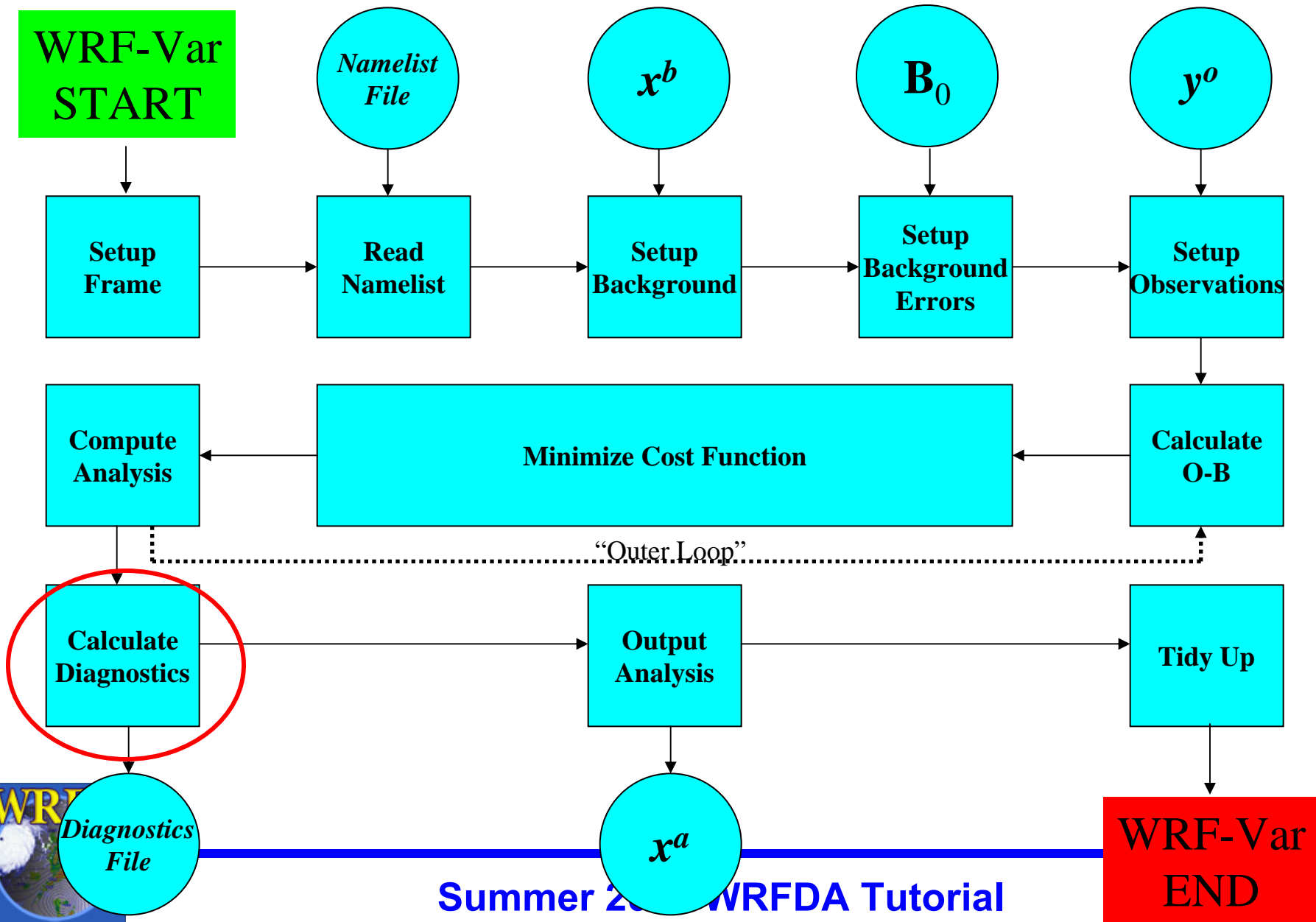
# Compute Analysis

---

- Once WRF-Var has found a converged control variable, convert control variable to model space analysis increments
- Calculate:  
$$\text{analysis} = \text{first-guess} + \text{analysis increment}$$
- Performs consistency checks, e.g., remove negative humidity etc.



# WRF-Var



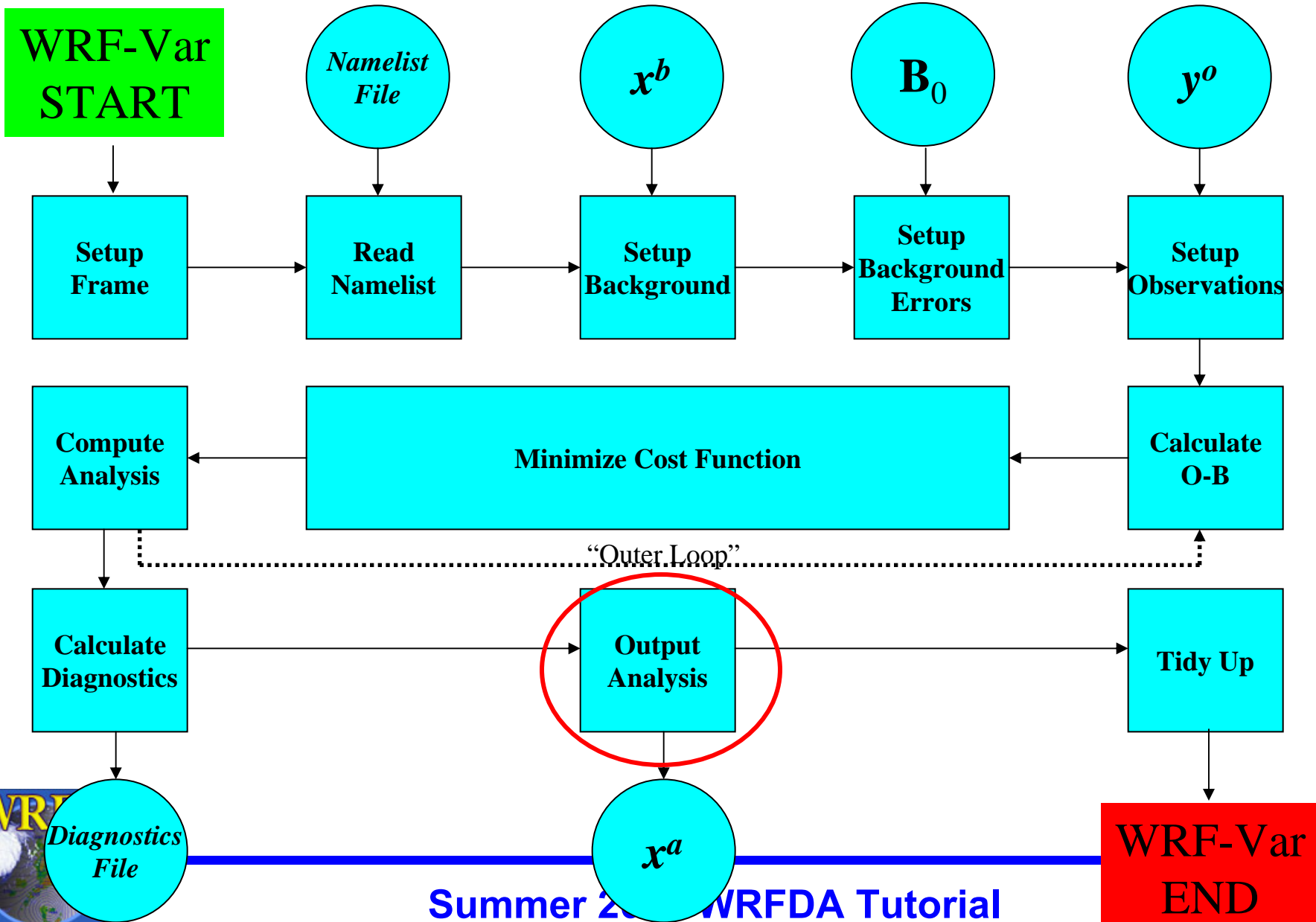
# Compute Diagnostics

---

- 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.



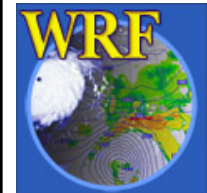
# WRF-Var



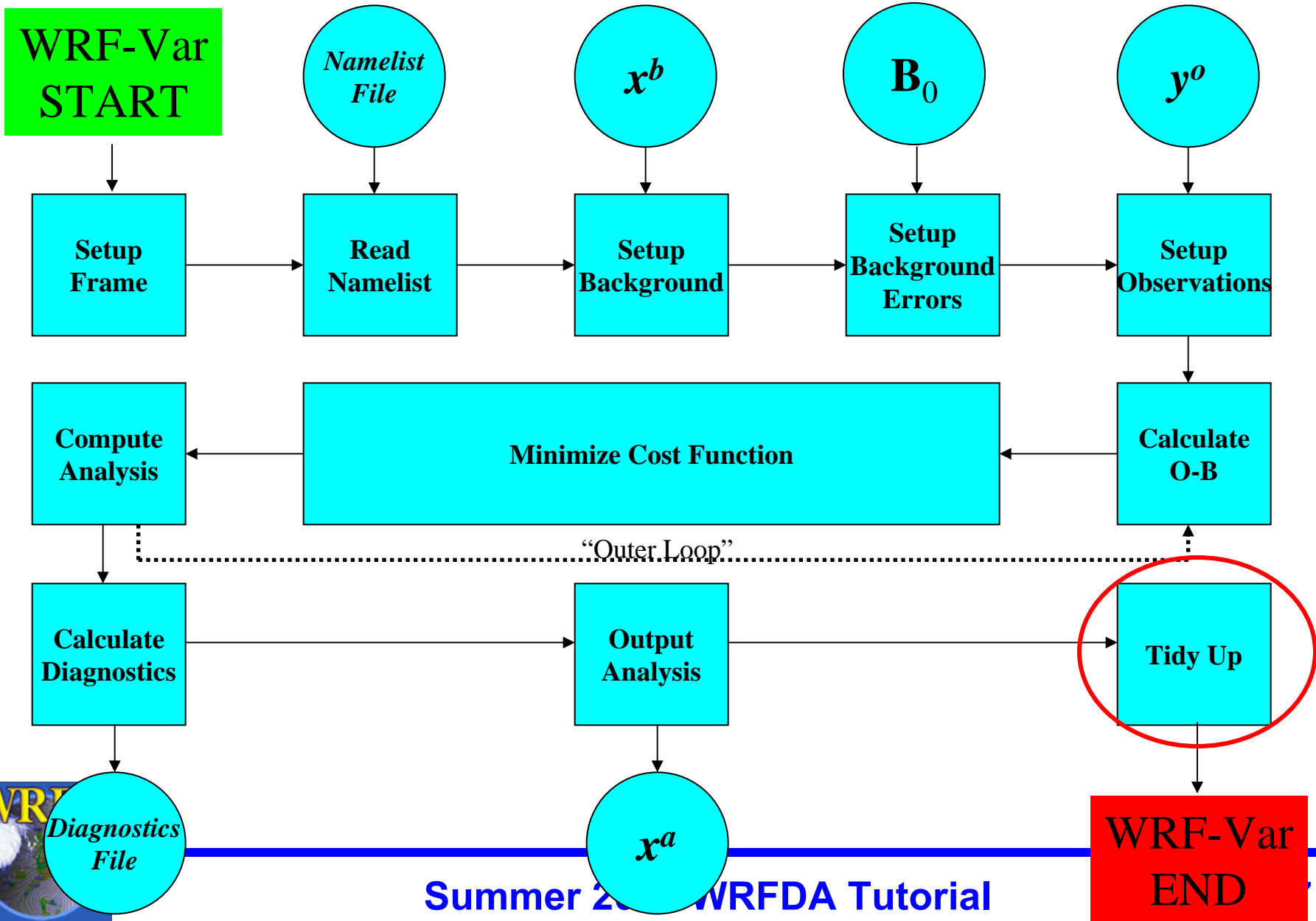
# Output Analysis

---

- Outputs analysis in native model format. Choice is made through namelist option “fg\_format”  
 $1 = \text{WRF, etc.}$
- Also output analysis increments (for diagnostic purposes) in native model format. Switch off by setting `WRITE_INCREMENTS = .FALSE.` in `namelist.input`.



# WRF-Var

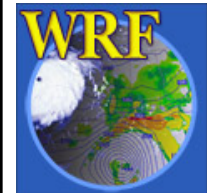




# Tidy Up

---

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



# Online WRFDA Resources

WRFDA has a dedicated page, similar to the ARW Users' page:  
<http://www.mmm.ucar.edu/wrf/users/wrfda/>

**WRFDA USERS PAGE**

Home Analysis System User Support Download Doc / Pub Links Users Forum

Search

**wrf-model.org**  
Public Domain Notice  
Contact WRF Support

### WRF Data Assimilation System Users Page

Welcome to the users home page for the Weather Research and Forecasting (WRF) model data assimilation system (WRFDA). The WRFDA system is in the public domain and is freely available for community use. It is designed to be a flexible, state-of-the-art atmospheric data assimilation system that is portable and efficient on available parallel computing platforms. WRFDA is suitable for use in a broad range of applications across scales ranging from kilometers of regional mesoscale to thousands of kilometers of global scales.

The Mesoscale and Microscale Meteorology Division of NCAR is currently maintaining and supporting a subset of the overall WRF code (Version 3) that includes:

- WRF Software Framework (WSF)
- Advanced Research WRF (ARW) dynamic solver, including one-way, two-way nesting and moving nests, grid and observation nudging
- WRF Pre-Processing System (WPS)
- WRF Data Assimilation System (WRFDA)
- Numerous physics packages contributed by WRF partners and the research community

Other components of the WRF system will be supported for community use in the future, depending on interest and available resources.

*updated Wed, 01 Apr 2009 17:24:21 GMT*

#### ANNOUNCEMENTS

Next WRF tutorial : July 13 - 24, 2009, Boulder, Colorado. Registration not yet open.

[WRF Version 3.1 Release Information](#)

[WRF Version 3.0.1.1 Release:](#)  
August 22, 2008

[WRF Var Version 3.0.1.1 Release:](#)  
August 29, 2008

New 'Known Problems' posts for V3 [WRF](#) (1/6/09) and [WPS](#) (8/4/08)

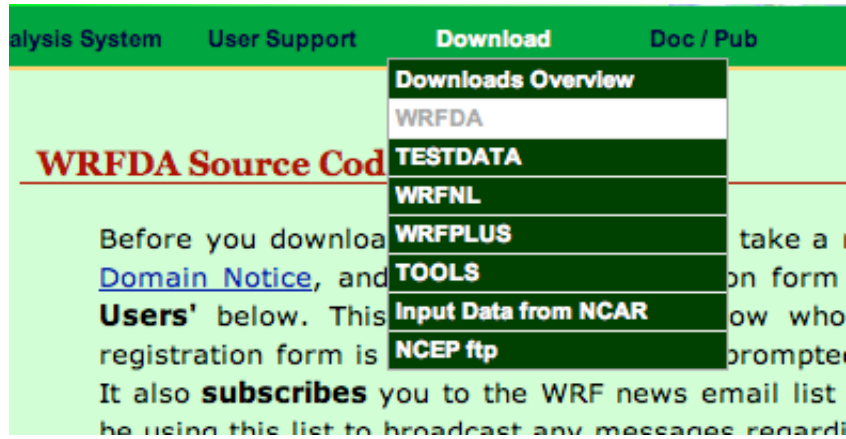
The 9th WRF Users' Workshop was held June 23 - 27, 2008 in Boulder, Colorado. [Workshop Presentations](#) is now online.

[MET 1.1 Release:](#) The Model Evaluation Tools (July, 2008)

[Online Tutorial](#) (updated on Oct 2008)

# Online WRFDA Resources

From the WRFDA page, one can access:

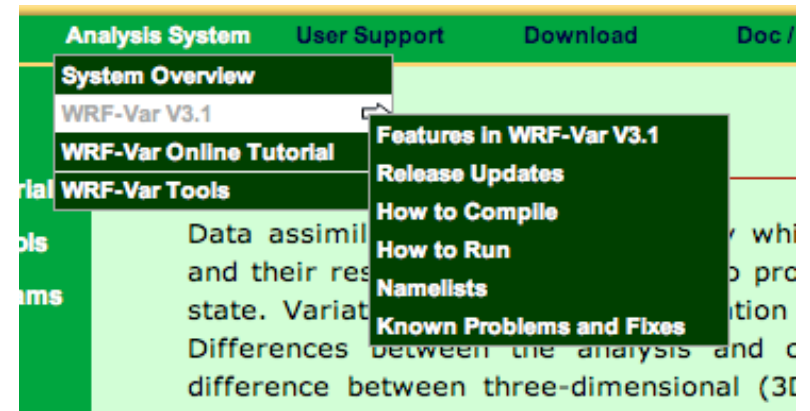


Analysis System User Support **Download** Doc / Pub

- Downloads Overview
- WRFDA
- TESTDATA
- WRFNL
- WRFPLUS
- TOOLS
- Input Data from NCAR
- NCEP ftp

**WRFDA Source Code**

Before you download, please take a moment to read the [Domain Notice](#), and the [Users' Registration Form](#) below. This registration form is required to download the source code. It also **subscribes** you to the WRF news email list by using this list to broadcast any messages regarding



Analysis System User Support **Download** Doc / Pub

- System Overview
- WRF-Var V3.1
- WRF-Var Online Tutorial
- WRF-Var Tools

- Features In WRF-Var V3.1
- Release Updates
- How to Compile
- How to Run
- Namelists
- Known Problems and Fixes

Data assimilation and their respective state. Variational differences between the analysis and observation difference between three-dimensional (3D)



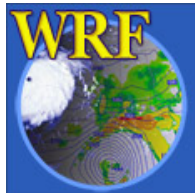
Analysis System User Support **Download** **Doc / Pub** Links Us

- Pubs & Docs Overview
- Tech Notes
- WRFDA User's Guide

**Documents & Publications**

**References:**

Barker, D. M., W. Huang, Y.-R. Guo, and A. Bourgeois, 2003: A Three-Dimensional Variational (3DVAR) Data Assimilation System For Use With MM5. NCAR/TN-453+STR, 68 pp. (Available from UCAR Communications)



# Other WRFDA Resources

---

- **Online WRF-Var Tutorial**

- <http://www.mmm.ucar.edu/wrf/users/wrfda/tutorial.html>

- **WRF-Var chapter of Users' Guide**

- [http://www.mmm.ucar.edu/wrf/users/docs/user\\_guide\\_V3/users\\_guide\\_chap6.htm](http://www.mmm.ucar.edu/wrf/users/docs/user_guide_V3/users_guide_chap6.htm)

- **If further help is needed, try the WRF Users' Forum**

- <http://forum.wrfforum.com/> or **ask questions via**

- [wrfhelp@ucar.edu](mailto:wrfhelp@ucar.edu)



# Questions?

